LabVIEW myRIO Toolkit

2024-04-26

Contents

myRIO Toolkit

June 2015, 373925C-01

The LabVIEW myRIO Toolkit provides you with tools for creating and deploying applications on the myRIO. The myRIO Toolkit contains the following components:

- **myRIO VIs**—The myRIO VIs provide functionality that you use to interface with the myRIO I/O channels and onboard devices.
- **myRIO USB Monitor**—This USB monitor shows basic information about the myRIO that you connect to your computer and provides options for you to work with the myRIO. This USB monitor appears when you use a USB cable to connect the myRIO to your computer.
- **Getting Started with myRIO**—This wizard helps you set up a new myRIO and test the onboard devices. To launch this wizard, launch LabVIEW, click the **Set Up and Explore** link, and select **Launch the Getting Started Wizard**. You can also access this wizard through the **myRIO USB Monitor**.
- **myRIO I/O Monitor**—This dialog box helps you test and monitor data from different I/O channels on the myRIO. To launch this dialog box, launch LabVIEW, click the **Set Up and Explore** link, and select **Launch the I/O Monitor**. You can also access this dialog box through the **myRIO USB Monitor**.
- **Templates and sample projects**—Templates demonstrate useful design patterns and serve as starting points for your applications. Sample projects demonstrate working applications based on the templates. You can customize the templates and sample projects according to the needs of your application. Select **File»Create Project** to display the **Create Project** dialog box, which lists templates and sample projects.
- **Examples**—Example VIs demonstrate common applications that you can create by using the myRIO Toolkit. You can modify an example VI to fit an application, or you can copy and paste from one or more example VIs into a VI that you create. Refer to the `labview\examples\myRIO` directory for example VIs installed with the myRIO Toolkit. You can also access these VIs by

selecting **Help»Find Examples** and then selecting **Toolkits and Modules»myRIO** in the NI Example Finder.

Visit the myRIO Community to find sensor drivers and example code, which requires third-party sensors and actuators. You can also share code and collaborate with other myRIO users on the myRIO Community.

To view related topics, click the **Locate** button, shown at left, in the toolbar at the top of this window. The **LabVIEW Help** highlights this topic in the **Contents** tab so you can navigate the related topics.

Getting Started with the myRIO Toolkit (myRIO Toolkit)

Use the following steps to get started using the myRIO with the myRIO Toolkit for developing your myRIO applications. Select a topic in each step to help you get started.

1. **Install Software and Drivers for Your myRIO Application**
   - LabVIEW myRIO Toolkit Readme
   - LabVIEW myRIO Software Bundle Readme
   - Using myRIO with Other NI Software
2. **Connect and Set Up the myRIO**
   - Connecting myRIO to a Host Computer
   - Using the Getting Started with myRIO Wizard
   - Installing Software on myRIO
   - Using the myRIO I/O Monitor
3. **Learn myRIO Programming**
   - Tutorials on Getting Started with myRIO Programming
   - Concepts in myRIO Programming

4. Create Your myRIO Application
   - Using myRIO Templates and Sample Projects
   - Using myRIO Examples
   - Using myRIO VIs
   - Creating myRIO Applications
   - Using myRIO with Other NI Software

Related Documentation (myRIO Toolkit)

The following documents contain information that you may find helpful as you use the LabVIEW myRIO Toolkit:

- **Getting Started with myRIO Programming**—Use these tutorials to learn about creating your first myRIO application and configuring a wireless network on the myRIO. To access these tutorials, launch LabVIEW, click the **Set Up and Explore** link in LabVIEW, and select **Access Getting Started Tutorials**.
- **myRIO I/O Monitor Help**—Use this help to learn about how to test and monitor data from each I/O channel on the myRIO. You can access this help by clicking the **Help** button in the **myRIO I/O Monitor** dialog box.
- myRIO User Guide and Specifications—Use this guide to learn about the myRIO hardware, specifications, and other helpful resources, such as dimensions, pin assignments, and connectivity information. You must have Adobe Reader installed to view the PDFs. Refer to the Adobe Systems Incorporated website at www.adobe.com to download the latest version of Adobe Reader. Refer to ni.com/manuals for updated documentation resources.
- **LabVIEW myRIO Toolkit Readme**—Use this file to obtain introductory information about the LabVIEW myRIO Toolkit, such as the product overview, system requirements, installation instructions, and known issues. Access this readme by navigating to the `labview\readme` directory and opening `readme_myRIO.html`.
- **LabVIEW myRIO Software Bundle Readme**—Use this file to obtain introductory information about the LabVIEW myRIO Software Bundle, such as

included software, system requirements, installation instructions, and known issues. To access this readme, insert the LabVIEW myRIO Software Bundle DVD 1 and open `readme_myRIOBundle.html`.

- Additional LabVIEW documentation for LabVIEW add-ons such as the Real-Time Module and the FPGA Module.

Error Codes (myRIO Toolkit and roboRIO Toolkit)

The myRIO VIs and the roboRIO VIs can return the following error codes. Refer to the KnowledgeBase for more information about correcting errors in LabVIEW.

Code	Description
−363034	The number of elements to write that you specify in **Values** is out of range. **Values** is a 2D array. The number of elements in each row represents the number of samples you want to write to each analog output channel. Ensure this number is greater than 0 and less than or equal to 10,000.
−363033	An unknown error occurred. Possible reason: the joystick device may not function correctly.
−363032	Unsupported device type. Ensure that you connect a joystick device to the myRIO or the roboRIO.
−363031	Joystick device not found. Ensure that you specify a correct **device ID**.
−363030	Memory allocation failed. Possible reason: one or more programs are causing insufficient system resources. Close some programs or restart the myRIO or the roboRIO.
−363029	The write buffer has overflowed. Specify a longer interval between write operations or set **Wait Until Done?** to TRUE.
−363028	The current personality is incompatible with the peripheral. To change the personality, right-click the myRIO or the roboRIO target in the **Project Explorer** window, select **Switch FPGA Personality**, and select an available personality from the shortcut menu.

−363027	The number of samples to read must be greater than 0 and less than or equal to 10,000.
−363026	Cannot create the timer interrupt with the specified **Timer ID**. Possible reasons: the specified **Timer ID** is out of range; you have already created a timer interrupt with the same timer ID. Specify a **Timer ID** within the range [0, 7] or destroy the created timer interrupt with the same timer ID.
−363025	Cannot register the interrupt with the specified **IRQ Number**. Possible reasons: the specified **IRQ Number** is out of range; you have already registered an interrupt with the same interrupt number. Specify an **IRQ Number** within the range [0, 7] or unregister the interrupt with the same interrupt number.
−363024	Cannot register the interrupt with the specified **Channel Name**. You have already registered an interrupt with the same channel name. Specify another value for **Channel Name** or unregister the interrupt with the same channel name.
−363023	The FPGA reference is already set and cannot be overwritten.
−363022	The input FPGA reference is invalid. Ensure that the input custom FPGA reference is correct for the FPGA target.
−363021	The specified channel has been opened with a different configuration that you cannot change at run time.
−363020	A timeout occurred while waiting for the myRIO or the roboRIO to initialize. Some of the I/O channels may not function correctly.
−363019	The number of scaling constant values provided does not match the number of channels that you specified.
−363018	Cannot write to the channel that you specified because the channel can only be an input.

−363017	The requested frequency is outside the range of supported frequencies and has been coerced to the closest supported frequency within the range.
−363016	Cannot reset the channel output because the channel was set to allow opening for multiple times.
−363015	Cannot open one or more channels that you specified with **allow multiple opens?** set to TRUE because the channel was already opened with **allow multiple opens?** set to FALSE.
−363014	There was an error during the code generation. Reinstall the LabVIEW myRIO Toolkit or the LabVIEW roboRIO Toolkit.
−363013	A timeout occurred during the last I2C read/write operation.
−363012	A No Acknowledge (NAK) bit was received from the slave device after the last data transmission.
−363011	A No Acknowledge (NAK) bit was received from the slave device after the last address transmission.
−363010	There are no channels specified to open. You must specify at least one channel in **channel name**.
−363009	One or more of the channels opened cannot be closed correctly. Check the corresponding open VI to ensure that you opened and/or configured the channels correctly.
−363008	One or more of the channels that you specified were already configured for use as a different I/O type.
−363007	Cannot open one or more of the channels that you specified with **allow multiple opens?** set to FALSE because the channel was already opened.
−363006	An overflow occurred in the encoder counter while the overflow flag was still set.

−363005	An invalid transition occurred in the quadrature encoder. Ensure that you have connected the hardware correctly and the encoder is not turning too fast.
−363004	The number of values to write must match the number of channels that you specify for the write operation.
−363003	One or more of the channels that you specified to read or write are not supported. Check to see if the channels that you specified are available on the myRIO or the roboRIO that you are using.
−363002	One or more of the channels that you specified are invalid. Ensure that all channel names that you enter correspond to the right type of channel and do not contain unnecessary characters or spaces.
−363001	The myRIO or the roboRIO is not configured. You must use the open VIs to open a reference to an I/O channel before using the channel. This error can occur if you attempt to run a myRIO VI or a roboRIO VI within the main application instance or under My Computer without specifying the target version of the myRIO or the roboRIO; or if you attempt to run a myRIO VI or a roboRIO VI on an RT target that is not on a myRIO or a roboRIO.
−363000	The target is unknown. This error occurs when you attempt to run a myRIO VI or a roboRIO VI on an RT target that is not on a myRIO or a roboRIO.

myRIO Hardware Overview (myRIO Toolkit)

Use the following topics to understand basic information about the myRIO hardware:

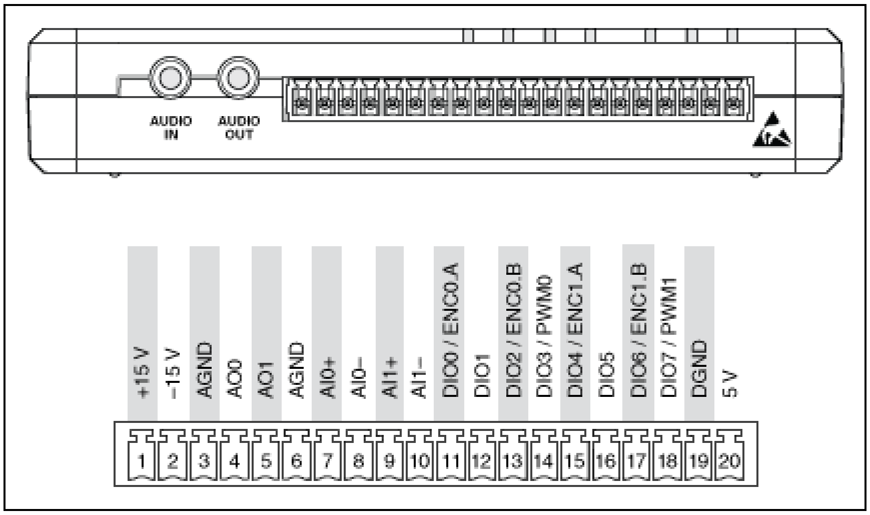
- I/O Connectors
- myRIO-1900 Hardware Block Diagram
- myRIO-1950 Hardware Block Diagram

To view related topics, click the **Locate** button, shown at left, in the toolbar at the top of this window. The **LabVIEW Help** highlights this topic in the **Contents** tab so you can navigate the related topics.

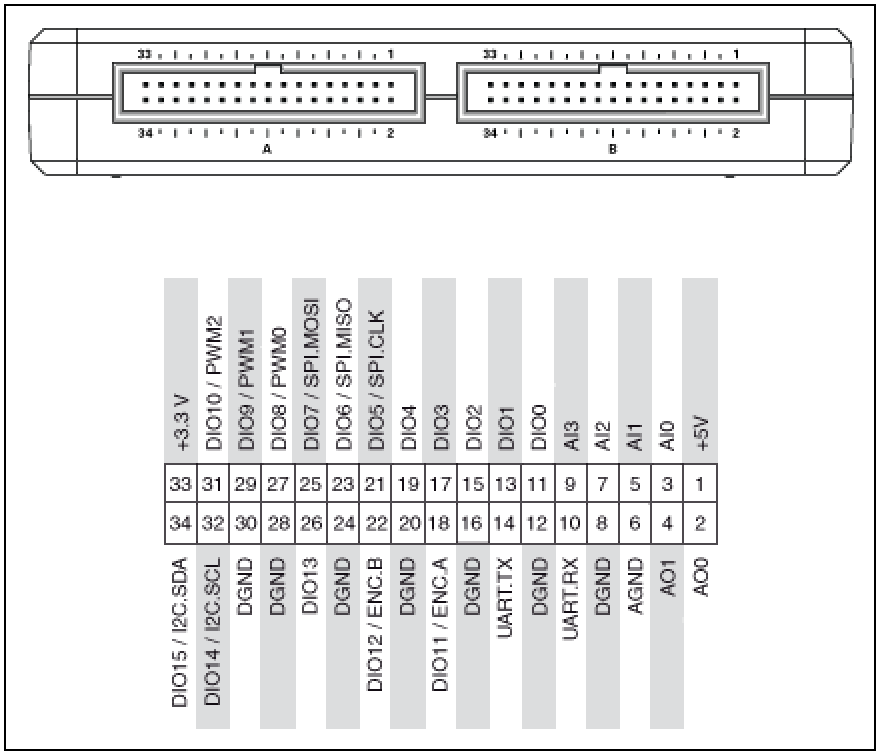
I/O Connectors (myRIO Toolkit)

The following figures show the pinouts of the I/O connectors on the myRIO.

Mini System Port (MSP) Connectors



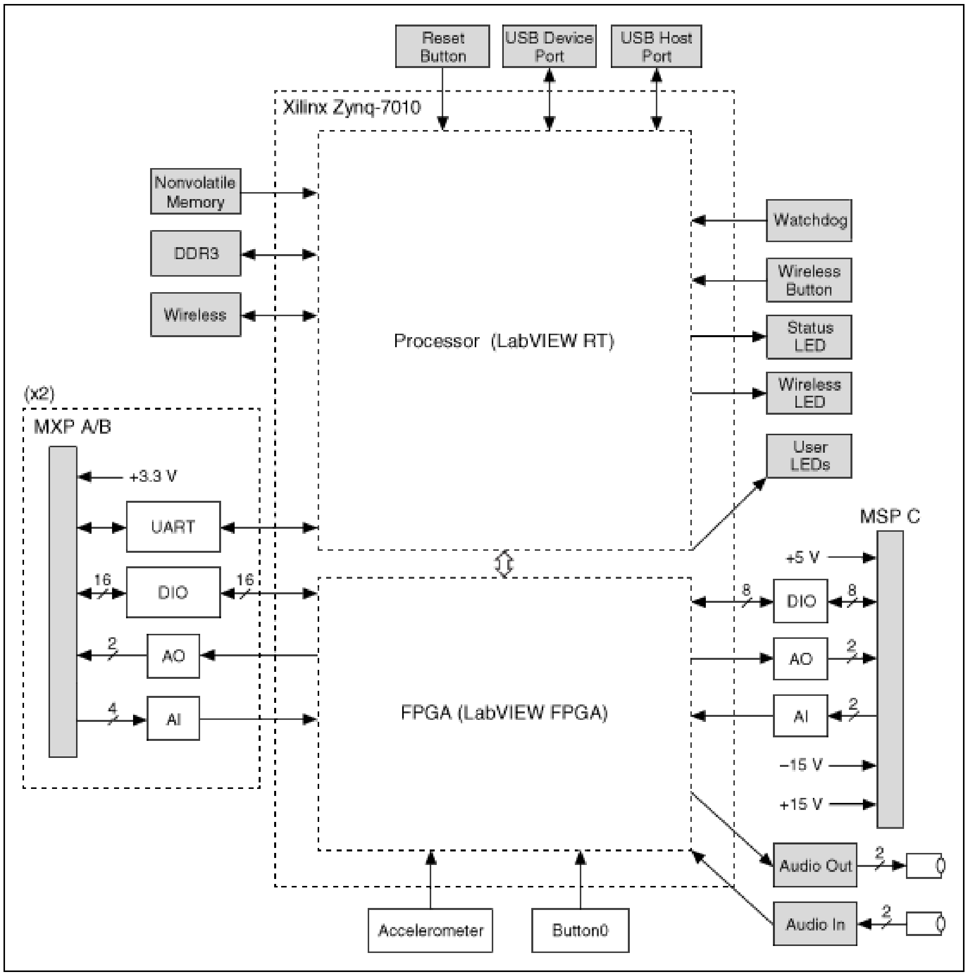
myRIO Expansion Ports (MXP) Connectors



+3.3 V	33	34	DIO15 / I2C.SDA
DIO10 / PWM2	31	32	DIO14 / I2C.SCL
DIO9 / PWM1	29	30	DGND
DIO8 / PWM0	27	28	DGND
DIO7 / SPI.MOSI	25	26	DIO13
DIO6 / SPI.MISO	23	24	DGND
DIO5 / SPI.CLK	21	22	DIO12 / ENC.B
DIO4	19	20	DGND
DIO3	17	18	DIO11 / ENC.A
DIO2	15	16	DGND
DIO1	13	14	UART.TX
DIO0	11	12	DGND
AI3	9	10	UART.RX
AI2	7	8	DGND
AI1	5	6	AGND
AI0	3	4	AO1
+5V	1	2	AO0

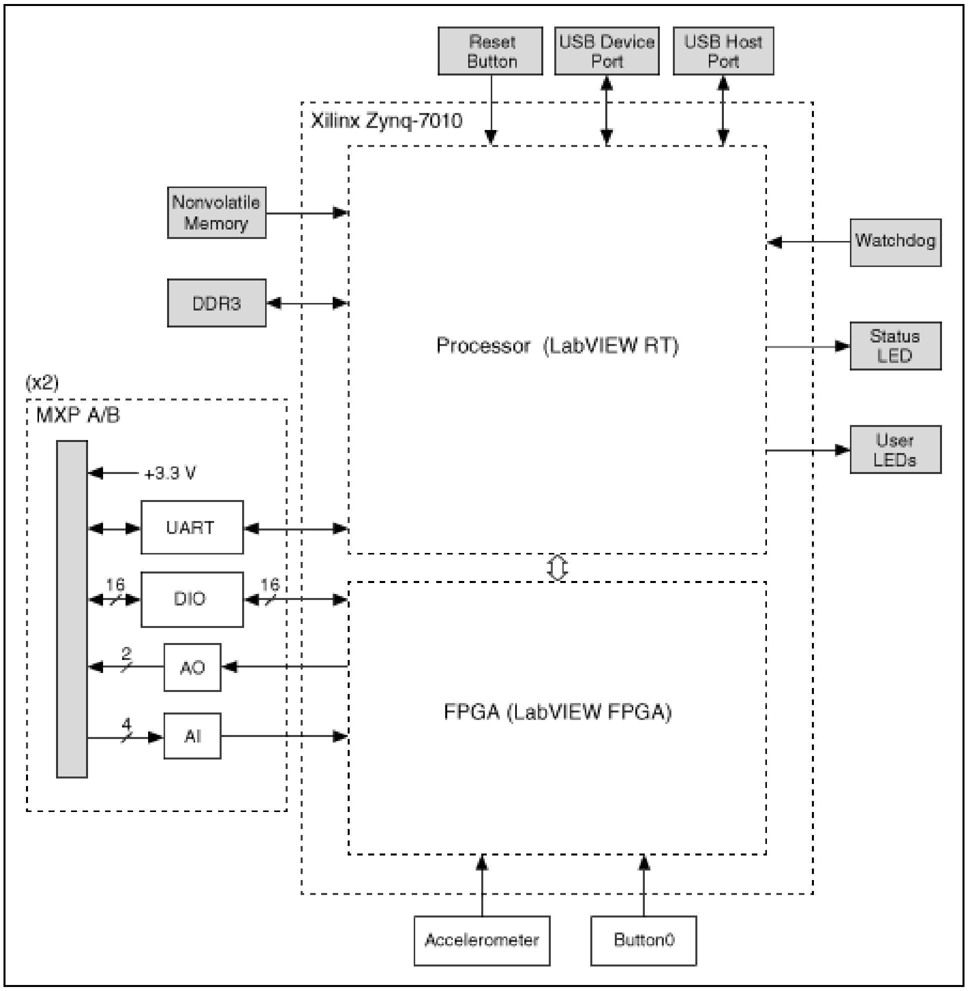
myRIO-1900 Hardware Block Diagram (myRIO Toolkit)

The following figure shows the hardware block diagram of the myRIO-1900.



myRIO-1950 Hardware Block Diagram (myRIO Toolkit)

The following figure shows the hardware block diagram of the myRIO-1950.



Configuring myRIO (myRIO Toolkit)

Use the following topics to learn about connecting the myRIO to a host computer and installing software on the myRIO:

- Connecting myRIO to a Host Computer
- Installing Software on myRIO

To view related topics, click the **Locate** button, shown at left, in the toolbar at the top of this window. The **LabVIEW Help** highlights this topic in the **Contents** tab so you can navigate the related topics.

Connecting myRIO to a Host Computer (myRIO Toolkit)

You can connect the myRIO to a host computer by using one of the following methods:

- **USB device connection**—When you use a USB cable to connect the myRIO from the USB device port to a host computer, the LabVIEW myRIO Toolkit installs a USB driver on the host computer. The USB driver then creates a virtual network interface card and assigns an IP address to the myRIO in the format 172.22.11.**x**.
- **Wireless connection**—When you connect the myRIO to a host computer over a wireless network, you can wirelessly detect the myRIO, deploy applications, and use shared variables to transfer data between the myRIO and the host computer. Before you can configure the settings of the wireless network on the myRIO, you must connect the myRIO to the host computer over USB. To learn about configuring the wireless network on the myRIO, launch LabVIEW, click the **Set Up and Explore** link, and select **Configure WiFi**.

Installing Software on myRIO (myRIO Toolkit)

When you run the **Getting Started with myRIO** wizard, LabVIEW installs the recommended software set on the myRIO. You can also use Measurement & Automation Explorer (MAX) to manually install software on the myRIO.

Complete the following steps to install software on the myRIO using MAX:

1. In MAX, expand **Remote Systems** in the configuration tree and then expand your myRIO target.
2. Right-click **Software** and click **Add/Remove Software** to launch **LabVIEW Real-Time Software Wizard**.
3. Select the recommended software set for the myRIO.
4. Click **Next**.
5. Click **myRIO x**, where **x** matches the version of the LabVIEW myRIO Toolkit.
6. Click **Next** to view a summary of your selection.

7. Click **Next** to start installing the software. When the installation completes, the wizard restarts the myRIO.
8. Click **Finish** to close the wizard.

Related Information

Getting Started with myRIO Wizard

Creating myRIO Applications (myRIO Toolkit)

Use the following topics to learn about the myRIO programming concepts that are helpful for creating a myRIO application:

- 1 Sample versus N Samples Modes
- Choosing between Express VIs and Low Level VIs
- Choosing FPGA Personalities
- Communicating with RT Targets from a Host Computer
- Creating Audio Applications
- Deploying Real-Time Applications
- Generating FPGA Clocks
- Understanding Hysteresis
- Understanding Latency
- Using Callback VIs
- Using the Project Explorer Window

To view related topics, click the **Locate** button, shown at left, in the toolbar at the top of this window. The **LabVIEW Help** highlights this topic in the **Contents** tab so you can navigate the related topics.

1 Sample versus N Samples Modes (myRIO Toolkit)

The LabVIEW myRIO Toolkit provides VIs for you to perform two I/O modes of signal acquisition and generation: 1 sample and **n** samples. Use the following table to learn about these modes.

Note You can perform **n** samples operations only after you install the NI High Throughput Add-On for myRIO. Visit `ni.com/info` and enter the Info Code ex6g5a to learn about the High Throughput Add-On for myRIO.

Operation	I/O Mode	Description	Use Case
Signal Acquisition	**1 Sample**	The myRIO acquires one sample each time for a single channel or multiple channels. This mode uses software-timed acquisition. The sample rate of acquisition depends on the software loop rate on the real-time processor. Various factors can affect this loop rate, such as simultaneous running of multiple programs on the myRIO target.	Acquiring the most recent value or periodically monitoring low frequency signals, such as the temperature.
	N Samples	The myRIO acquires multiple samples at the same time for a single channel or multiple channels. This mode has the following characteristics: - **Hardware-timed acquisition**—The FPGA target on the myRIO has a 40 MHz cloc	Acquiring finite high frequency signals, such as an audio signal.

		k rate, which controls the rate of acquisition. The sample rate depends on the hardware clock, which is faster than a software loop. A hardware clock is more accurate than a software loop. Therefore, you can have accurate control over the time between each sample. ▪ **Buffered acquisition**—FPGA transfers the samples from the FPGA target to an intermediate memory buffer using direct memory access (DMA) before LabVIEW reads these samples on the real-time processor.	
Signal Generation	**1 Sample**	The myRIO generates one sample each time for a single channel or multiple channels. This mode uses software-timed generation. The sample rate of generation depends on the software loop rate on RT side. Various factors can affect this loop rate, such as the simultaneous run	Generating the most recent value or generating low frequency signals. For example, generating a known voltage to stimulate a device.

		ning of multiple programs on the myRIO target.	
	N Samples	The myRIO generates multiple samples at the same time for a single channel or multiple channels. This mode has the following characteristics: - **Hardware-timed generation**—The FPGA target on the myRIO has a 40 MHz clock rate, which controls the rate of generation. The sample rate depends on the hardware clock, which is faster than a software loop. A hardware clock is more accurate than a software loop. Therefore, you can have accurate control over the time between each sample. - **Buffered generation**—The real-time processor takes the samples from LabVIEW and places them in an intermediate memory buffer using DMA before FPGA gets the sa	Generating finite time-varying signals, such as an AC sine wave.

		mples from the buffer.	

After you install the High Throughput Add-On for myRIO, you can refer to `High Frequency Sampling.lvproj` in the `labview\examples\myRIO\High Frequency Sampling` directory for an example of comparing the 1 sample mode with the **n** samples mode using the myRIO Express VIs.

Choosing between Express VIs and Low Level VIs (myRIO Toolkit)

The LabVIEW myRIO Toolkit provides the following types of VIs for creating myRIO applications:

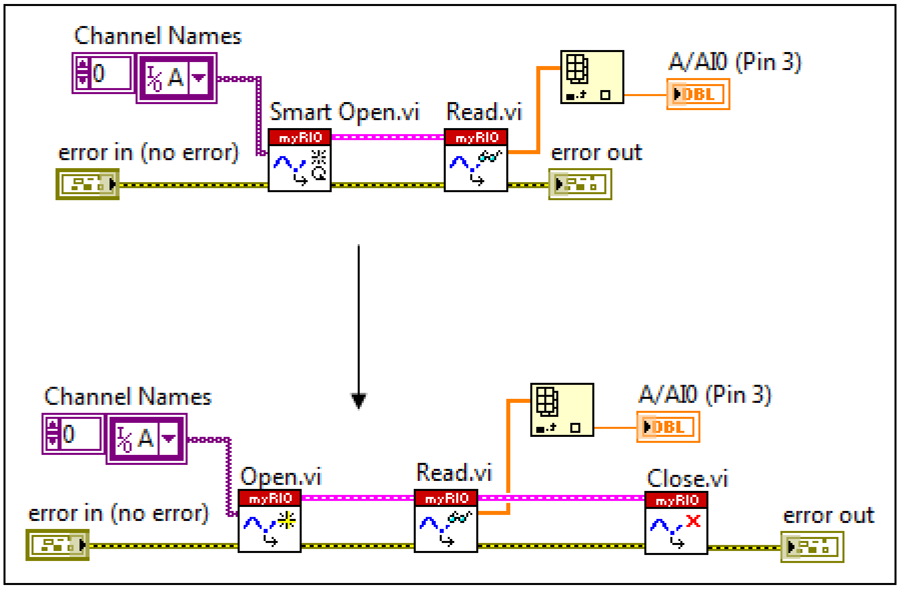
- **Express VIs**—Use the Express VIs to interactively configure the settings for the I/O channels. When you place an Express VI on the block diagram or double-click an Express VI, a configuration dialog box appears. Use this configuration dialog box to configure the Express VI. You can also configure the Express VI by wiring values to the terminals of the Express VI on the block diagram.
- **Low Level VIs**—Use the Low Level VIs to have more control over allocating and releasing I/O channels.

Understanding the Underlying Code of Express VIs

In the configuration dialog box of an Express VI, click the **View Code** tab to view the underlying code of the Express VI. This code consists of the Low Level VIs and other LabVIEW VIs.

The code of the Express VIs uses Smart Open VIs instead of the Open VIs to open references to I/O channels. Smart Open VIs can open an I/O reference the first time they run and then save the reference in memory. The use of Smart Open VIs ensures that the Express VIs can run efficiently in loops because you do not need to open references to I/O channels for each iteration.

You can copy the code of an Express VI to a new block diagram and start programming with the Low Level VIs. After you copy the code, replace the Smart Open VI with an Open VI because the Open VIs open an I/O reference and pass the reference around to all other places where you need to access the I/O channel. You also need to add a Close VI to close I/O references after data operation. The following figure shows an example of modifying the code copied from the Analog Input Express VI.



Related Information

Express VIs

Choosing FPGA Personalities (myRIO Toolkit)

FPGA personalities consist of predefined FPGA bitfiles for you to program with myRIO. The LabVIEW myRIO Toolkit provides the following FPGA personalities.

- **Default**—Supports general I/O, protocols, and interrupt. Use the default personality for control applications.
- **High Throughput**—Supports high-speed analog or digital data access. Use the high-throughput personality for audio signals and projects in need of waveform data.

Note To use the high-throughput personality, you must install the NI High Throughput Add-On for myRIO. Visit `ni.com/info` and enter the Info Code ex6g5a to learn about the High Throughput Add-On for myRIO.

You can choose a personality when you create a myRIO project. With different personalities, you can implement different functionalities and use different channels on the myRIO.

To switch between personalities, right-click a myRIO target in the **Project Explorer** window, select **Switch FPGA Personality**, and choose an available personality from the shortcut menu.

Note You cannot switch between personalities in myRIO projects that include customized FPGA code, such as the myRIO Custom FPGA Project.

The following table lists the functionalities and channel numbers that each personality supports.

Supported Functionalities	Number of Supported Channels			
	Default		High Throughput	
	myRIO-1900	myRIO-1950	myRIO-1900	myRIO-1950
Analog Input (1 Sample)	12	8	12	8
Analog Output (1 Sample)	8	4	8	4
Digital Input	40	32	40	32
Digital Output	40	32	40	32
Button	1	1	1	1
LED	4	4	4	4
Accelerometer	3	3	3	3
PWM	8	6	/	/
Encoder	4	2	/	/
SPI	2	2	/	/
I2C	2	2	/	/
UART	2	2	2	2
Interrupt	8	8	/	/
Analog Input (N Samples)	/	/	1	1

Analog Output (N Samples)	/	/	1	1
Audio Input (N Samples)	/	/	2	/
Audio Output (N Samples)	/	/	2	/
Digital Input (N Samples)	/	/	1	1
Digital Output (N Samples)	/	/	1	1
Input Device (Joystick)	1	1	1	1

Supported Channels

The default FPGA personality supports the following channels of the I/O connectors on the myRIO.

Supported Functionalities	Supported Channels	
	myRIO-1900	myRIO-1950
Analog Input	A/AI0~A/AI3* B/AI0~B/AI3* C/AI0, C/AI1* AudioIn/Left, AudioIn/Right	A/AI0~A/AI3 B/AI0~B/AI3
Analog Output	A/AO0, A/AO1 B/AO0, B/AO1 C/AO0, C/AO1 AudioOut/Left, AudioOut/Right	A/AO0, A/AO1 B/AO0, B/AO1
Digital Input	A/DIO0~A/AI15 B/DIO0~B/DIO15 C/DIO0~C/DIO7	A/DIO0~A/AI15 B/DIO0~B/DIO15
Digital Output	A/DIO0~A/AI15 B/DIO0~B/DIO15 C/DIO0~C/DIO7	A/DIO0~A/AI15 B/DIO0~B/DIO15
Button	Button0	Button0
LED	LED0~3	LED0~3
Accelerometer	X-Axis, Y-Axis, Z-Axis	X-Axis, Y-Axis, Z-Axis

PWM	A/PWM0~A/PWM2 B/PWM0~B/PWM2 C/PWM0, C/PWM1	A/PWM0~A/PWM2 B/PWM0~B/PWM2
Encoder	A/ENC B/ENC C/ENC0, C/ENC1	A/ENC B/ENC
SPI	A/SPI B/SPI	A/SPI B/SPI
I2C	A/I2C B/I2C	A/I2C B/I2C
UART	A/UART B/UART	A/UART B/UART
Interrupt	A/AI0 (Interrupt), A/AI1 (Interrupt), A/DIO0 (Interrupt)~A/DIO3 (Interrupt), Button0 (Interrupt)	A/AI0 (Interrupt), A/AI1 (Interrupt), A/DIO0 (Interrupt)~A/DIO3 (Interrupt), Button0 (Interrupt)
Input Device (Joystick)	USB	USB

* A/, B/, and C/ stand for connector A, B, and C on the myRIO. Refer to the **myRIO User Guide and Specifications** for specifications of the I/O connectors and channels on the myRIO.

The high-throughput FPGA personality supports the following channels of the I/O connectors on the myRIO.

Supported Functionalities	Supported Channels	
	myRIO-1900	myRIO-1950
Analog Input	A/AI0~A/AI3 B/AI0~B/AI3 C/AI0, C/AI1 AudioIn/Left, AudioIn/Right	A/AI0~A/AI3 B/AI0~B/AI3
Analog Output	A/AO0, A/AO1 B/AO0, B/AO1 C/AO0, C/AO1 AudioOut/Left, AudioOut/Right	A/AO0, A/AO1 B/AO0, B/AO1
Analog Input (N Samples)	A/AI0 (N Samples)	A/AI0 (N Samples)
Analog Output (N Samples)	A/AO0 (N Samples)	A/AO0 (N Samples)

Audio Input (N Samples)	AudioIn/Left (N Samples), AudioIn/Right (N Samples)	/
Audio Output (N Samples)	AudioOut/Left (N Samples), AudioOut/Right (N Samples)	/
Digital Input	A/DIO0~A/AI15 B/DIO0~B/DIO15 C/DIO0~C/DIO7	A/DIO0~A/AI15 B/DIO0~B/DIO15
Digital Output	A/DIO0~A/AI15 B/DIO0~B/DIO15 C/DIO0~C/DIO7	A/DIO0~A/AI15 B/DIO0~B/DIO15
Digital Input (N Samples)	A/DI0 (N Samples)	A/DI0 (N Samples)
Digital Output (N Samples)	A/DO0 (N Samples)	A/DO0 (N Samples)
Button	Button0	Button0
LED	LED0~3	LED0~3
Accelerometer	X-Axis, Y-Axis, Z-Axis	X-Axis, Y-Axis, Z-Axis
UART	A/UART B/UART	A/UART B/UART
Input Device (Joystick)	USB	USB

Related Information

myRIO Shipping Personality Reference

myRIO User Guide and Specifications

I/O Connectors

Communicating with RT Targets from a Host Computer (myRIO Toolkit)

Use the following table to learn about the methods that you can use to communicate with myRIO targets from a host computer.

	Network Communication	Front Panel Communication
Description	A host VI runs on the host computer and communicates with the VI running on the RT target using specific network communication programmatic controls.	LabVIEW and the RT Engine execute different parts of the same VI. LabVIEW on the host computer displays the front panel of th

		e VI while the RT Engine executes the block diagram.
Use Case	Customizing the code of the VIs respectively running on the host computer and on the RT target. Controlling the data that the host computer and the RT target send to each other.	Monitoring the VIs running on an RT target.
Example	Refer to the Network Communication sample project for examples of creating different types of network communication systems using the myRIO. Access this sample project by selecting **File»Create Project** in LabVIEW and selecting **Sample Projects»myRIO**.	None.

Related Information

Real-Time System Components

Creating Audio Applications (myRIO Toolkit)

The myRIO-1900 model provides audio input and output channels. You can use the audio channels to perform audio acquisition and playback.

Use the following methods to access the audio input and output channels.

Method	Audio Input		Audio Output	
	Default	High Throughput	Default	High Throughput
myRIO Express VIs	Use the Audio Input Express VI and select **AudioIn/Left** or **AudioIn/Right** channel in the configuration dialog box.	Use the Audio Input Express VI and select **AudioIn/Left (N Samples)** or **AudioIn/Right (N Samples)** channel in the configuration dialog box.	Use the Audio Output Express VI and select **AudioOut/Left** or **AudioOut/Right** channel in the configuration dialog box.	Use the Audio Output Express VI and select **AudioOut/Left (N Samples)** or **AudioOut/Right (N Samples)** channel in the configuration dialog box.

myRIO Low Level VIs	Use the Open VI on the Analog Input 1 Sample palette to open a reference to the audio input channels.	Use the Open VI on the Audio Input N Samples palette to open a reference to the audio input channels.	Use the Open VI on the Analog Output 1 Sample palette to open a reference to the audio output channels.	Use the Open VI on the Audio Output N Samples palette to open a reference to the audio output channels.
FPGA Interface Nodes	Use the Open FPGA VI Reference function to open a reference to the default FPGA personality and wire the output reference to the Read/Write Control function to access the audio input channels.	Use the Open FPGA VI Reference function to open a reference to the high-throughput FPGA personality and wire the output reference to the Read/Write Control function and the Invoke Method function to access the audio input channels.	Use the Open FPGA VI Reference function to open a reference to the default FPGA personality and wire the output reference to the Read/Write Control function to access the audio output channels.	Use the Open FPGA VI Reference function to open a reference to the high-throughput FPGA personality and wire the output reference to the Read/Write Control function and the Invoke Method function to access the audio output channels.

Refer to the Clap Sensor sample project and Voice Recorder sample project for examples of using the audio channels on the myRIO. Access these sample projects by selecting **File»Create Project** in LabVIEW and selecting **Sample Projects»myRIO**.

Note The Voice Recorder sample project is available only after you install the NI High Throughput Add-On for myRIO. Visit `ni.com/info` and enter the Info Code ex6g5a to learn about the High Throughput Add-On for myRIO.

Related Information

myRIO User Guide and Specifications—Refer to this manual for specifications of the audio channels on the myRIO.

Deploying Real-Time Applications (myRIO Toolkit)

After you build a real-time application for the myRIO, you need to deploy the application to the RT target on the myRIO. Use one of the following methods to deploy real-time applications:

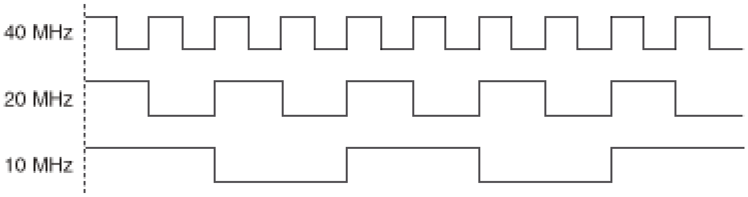
- **Interactive Deployment**—Deploys real-time applications from the **Project Explorer** window. Interactive deployment requires that the RT target stays connected to the LabVIEW development environment. In the **Project Explorer** window, you can run VIs on the RT target by clicking **Run**. LabVIEW compiles the VIs and their dependencies and deploys the application to the RT target.
- **Headless Deployment**—Deploys real-time applications from outside of the LabVIEW development environment, such as C programming. Use this method to programmatically deploy to targets or when you do not have access to LabVIEW.

Related Information

Building and Deploying a Stand-Alone Real-Time Application

Generating FPGA Clocks (myRIO Toolkit)

The FPGA target on the hardware has a clock rate of 40 MHz, which means the clock cycle is 25 ns. To generate a frequency that is less than 40 MHz, you can change a signal on every **x** number of rising edges of the clock signal. The following figure shows an example of changing a 40 MHz clock signal on rising edges to generate 20 MHz and 10 MHz clock signals, respectively.



In the previous figure, you toggle the clock signal on every rising edge to generate the 20 MHz clock signal and toggle the clock signal on every other rising edge to generate the 10 MHz clock signal.

The frequency to generate on the hardware must meet the following requirements:

- The frequency must be divisible by 25 ns because the clock cycle is 25 ns and you count the number of rising edges to make changes. For example, you cannot generate a frequency of 25 MHz. The first achievable frequency below 40 MHz is 20 MHz.
- The frequency must be within the range of approximately 610.35 Hz to 40 MHz. The hardware uses a 16-bit counter that counts from 0 to 65,535. You can calculate the lowest achievable frequency by the equation 1/(25 ns * 65536) ≈ 610.35 Hz.

You can use clock divisors to generate even lower frequencies for different I/O types. You can divide the base frequency by even numbers and use the generated clock to increment the counter. For example, with a clock divisor of 2, the lowest achievable frequency is 1/(50 ns * 65536) ≈ 305.17 Hz.

The following equation calculates frequencies that you can generate for pulse width modulation (PWM) I/O.

$$\mathbf{f}_{\mathrm{PWM}} = \mathbf{f}_{\mathrm{clk}} / (\mathbf{N} * [\mathbf{X} + 1])$$

where

$\mathbf{f}_{\mathrm{PWM}}$ is the desired PWM frequency

$\mathbf{f}_{\mathrm{clk}}$ is the base clock frequency

N is the clock divisor

X is the number of counts before changing the signal

The following equation calculates frequencies that you can generate for serial peripheral interface (SPI) I/O.

$$\mathbf{f}_{\mathrm{SPI}} = \mathbf{f}_{\mathrm{clk}} / (2 * \mathbf{N} * [\mathbf{X} + 1])$$

where

$\mathbf{f}_{\mathrm{SPI}}$ is the desired SPI frequency

$\mathbf{f}_{\mathrm{clk}}$ is the base clock frequency

N is the clock divisor

X is the number of counts before changing the signal

Some Express VIs include a **Validate** button for validating whether these Express VIs can generate the frequency that you specify. If you specify an invalid value for **Frequency**, the Express VIs coerce the specified value to the nearest valid value when you click the **Validate** button.

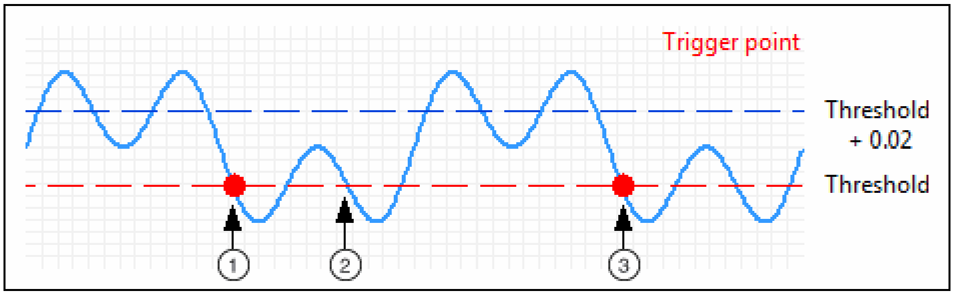
Understanding Hysteresis (myRIO Toolkit)

Hysteresis, also known as window size, adds a window above or below the trigger level. Hysteresis reduces false triggering due to noise or jitter in the signal.

When you use the Register Analog Input Interrupt VI or the Interrupt Express VI to generate analog input interrupts, both of these VIs use a 0.02 V hysteresis to avoid false interrupt registration. The way this hysteresis works depends on whether you register an interrupt on the falling edge or rising edge of the analog input signal, as described in the following table.

Type	Description
Analog Falling Edge	The VI registers an interrupt when the signal star ts or rises above **Threshold** plus hysteresis and then drops below **Threshold**.
Analog Rising Edge	The VI registers an interrupt when the signal star ts below **Threshold** minus hysteresis and then crosses above **Threshold**.

Use the following figure to understand when the VI registers analog-falling-edge interrupts.



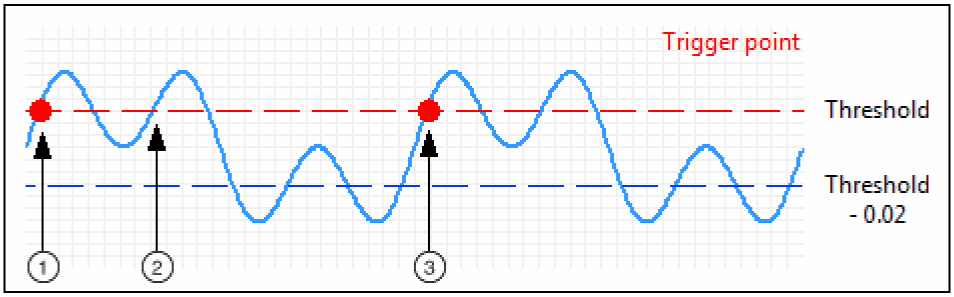
The following list describes important details about the previous figure:

① The signal rises above **Threshold** plus hysteresis. When the signal value drops below **Threshold**, the VI registers an interrupt.

② The VI does not register any interrupt, because the signal stays below **Threshold** plus hysteresis since the VI last registered an interrupt.

③ The signal rises above **Threshold** plus hysteresis. When the signal drops below **Threshold**, the VI registers another interrupt.

Use the following figure to understand when the VI registers analog-rising-edge interrupts.



The following list describes important details about the previous figure:

① The VI registers an interrupt when the signal value rises above **Threshold**.

② The VI does not register any interrupt, because the signal stays above **Threshold** minus hysteresis since the VI last registered an interrupt.

③ The signal drops below **Threshold** minus hysteresis. When the signal crosses above **Threshold**, the VI registers another interrupt.

Related Information

Interrupt Express VI

Register Analog Input Interrupt VI

Understanding Latency (myRIO Toolkit)

Latency refers to the time it takes to complete an operation. When you use the myRIO Express VIs to acquire or generate signals, latency has different meanings.

Latency in Signal Acquisition

Latency in signal acquisition refers to the time that the myRIO needs to transfer the acquired signal to the real-time processor. In the **n** samples mode, latency involves the software or memory transfer that gets the signal into the correct memory and delivers the signal to the algorithm.

When you use the myRIO Express VIs to perform signal acquisition in the **n** samples mode, latency refers to the time it takes to complete the following processes:

- **DMA read operation**—FPGA transfers the signal values from the FPGA buffer to the real-time buffer. DMA is a FIFO-based method of transferring data between an FPGA target and the host.
- **Data copying**—The real-time processor copies the signal values from the real-time buffer to the algorithm.

Note The latency changes when the number of samples you want to read changes. For example, if there is more data to read, data copying takes more time.

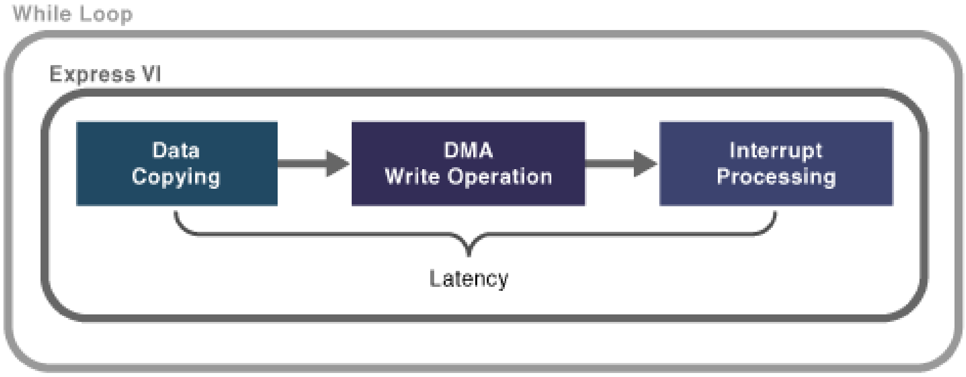
Latency in Signal Generation

Latency in signal generation refers to the time that the myRIO needs to export the acquired signal to another data acquisition (DAQ) device. In the **n** samples mode, latency involves the software or memory transfer that gets the signal from the algorithm and delivers the signal into the correct memory.

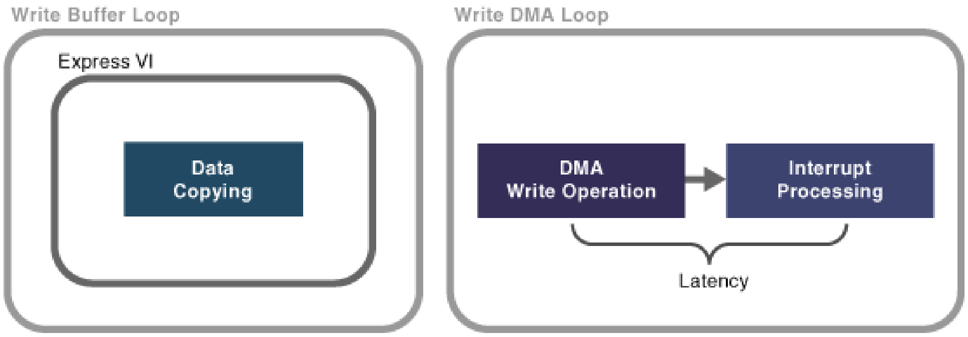
When you use the myRIO Express VIs to perform signal generation in the **n** samples mode, the following processes can result in latency. The actual latency varies depending on whether you place a checkmark in the **Wait until done?** checkbox on these Express VIs.

- **Data copying**—The real-time processor copies the signal values from the algorithm to the real-time buffer.
- **DMA write operation**—The real-time processor transfers the signal values from the real-time buffer to the FPGA side.
- **Interrupt processing**—When the FPGA buffer finishes writing **n** sample values to the pins on the myRIO, FPGA sends an interrupt to the real-time processor. When the real-time processor receives the interrupt, the real-time processor continues to write another **n** sample values to the FPGA buffer.

The following figure shows the latency in signal generation when you place a checkmark in the **Wait until done?** checkbox. Latency refers to the time it takes to complete the data copying, DMA write operation, and interrupt processing.



The following figures show the latency in signal generation when the **Wait until done?** checkbox does not contain a checkmark. In this situation, RT continuously writes signal values from the algorithm to the real-time buffer. Meanwhile, RT uses another loop to transfer the values from the real-time buffer to the FPGA buffer. Thus, latency refers to the time it takes to complete the DMA write operation and interrupt processing. Typically, this latency is 400 µs.



Related Information

Analog Input Express VI

Analog Output Express VI

Audio Input Express VI

Audio Output Express VI

Digital Input Express VI

Digital Output Express VI

Using Callback VIs to Handle Interrupts (myRIO Toolkit)

An interrupt is a signal indicating an event that needs immediate attention. You can use interrupts to alert the target to a high-priority condition that requires the interruption of the current block diagram code that the target is executing. To trigger an interrupt, you can use the Interrupt Express VI or Interrupt Low Level VIs.

You must use callback VIs to handle interrupts. The Callback VI contains code that handles interrupts and runs when the interrupt triggering occurs.

Considerations for Callback VIs

When you create or use a callback VI, take the following factors into consideration:

- Callback VIs are of a time-critical priority, which is the highest priority in block diagram code on RT targets. This means that the priority of callback VIs is higher than that of any Timed Loops on RT targets. To avoid unexpected timing behavior, do not add a Timed Loop to the block diagram of callback VIs.
- You can use callback VIs to unregister or destroy interrupts. When you unregister or destroy an interrupt, the interrupt triggering stops. Destroying an interrupt also releases the resources associated with the interrupt.
- To communicate between a callback VI and a non-callback VI, you must use global variables or function global variables, which help access and pass data among VIs.
- A callback VI can use one or two CPUs, as described in the following table.

Scenario	Use Case	Note
One CPU Used	The block diagram of the callback VI includes only one While Loop. This loop does not contain pieces of code running in parallel.	The non-callback code keeps executing when the callback VI runs.
Two CPUs Used	- The block diagram of the callback VI includes two While Loops. Each CPU executes the code running in one While Loop. - The block diagram code of the callback VI contains one While Loop. This loop contains two pieces of code that execute in parallel.	- The non-callback code stops executing until the callback VI finishes running and frees up one CPU. - Using two CPUs for a long time can cause a disconnection between the host computer and the target, because keeping this connection requires using the CPU of the target. **Note** National Instruments recommends that you use callback VIs to execute short code that handles emergent events.

Creating a Callback VI

To create a callback VI, you must ensure the connector pane of the VI meets the following requirements:

- The connector pane uses the 4 x 2 x 2 x 4 pattern.
- The data type of the top-left terminal is unsigned 8-bit integer.
- The top-left terminal is a recommended terminal.

You can also create callback VIs from templates by completing the following steps:

1. In LabVIEW, select **File»Create Project** to launch the **Create Project** dialog box and create a new myRIO project.
2. In the **Project Explorer** window, right-click the myRIO target, select **Create Callback VI**, and select one of the following options:
   - **IO IRQ**—Creates a callback VI that handles I/O interrupts, such as analog and digital input interrupts.
   - **Timer IRQ**—Creates a callback VI that handles timer interrupts.
3. In the **Specify a name for the callback VI** dialog box, specify a name for the VI and click **Save**.
4. Click **OK** when LabVIEW finishes creating the VI and adding this VI to the myRIO target. You can find the callback VI when you expand the myRIO target.

Opening a Callback VI

In the **Project Explorer** window, you can always double-click a callback VI under the target to open and edit this VI. When you add multiple callback VIs under the target, use one of the following methods to locate the callback VI that a VI uses:

- Double-click the Callback VI reference that you wire to the Register Analog Input Interrupt VI or the Register Digital Input Interrupt VI to open the callback VI that a Low Level VI uses.
- Right-click the Interrupt Express VI and select **Open Callback VI** from the shortcut menu to open the callback VI that the Express VI uses.

Related Information

Building the Connector Pane

Create Project Dialog Box

Global Variables

Interrupt Express VI

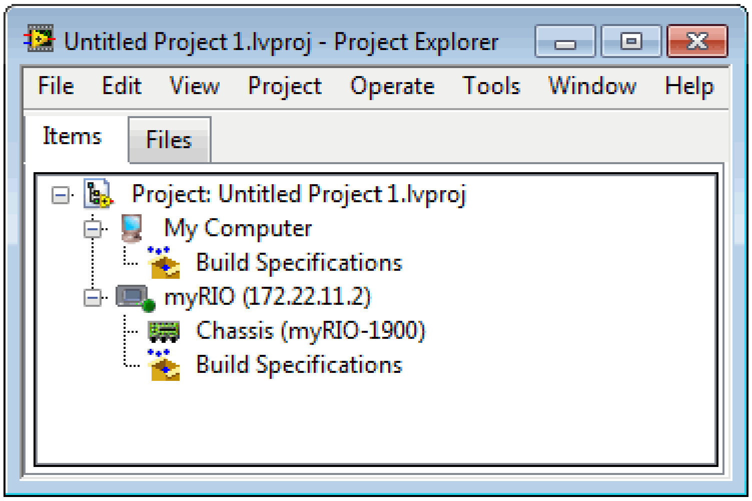
Interrupt VIs

Using the Project Explorer Window (myRIO Toolkit)

Use the **Project Explorer** window to manage LabVIEW projects for the following tasks:

- Managing the targets, VIs, and other support files of a project from one location.
- Connecting the host computer to myRIO targets, setting target properties, and deploying VIs to targets.

The following figure shows the **Project Explorer** window when you add a myRIO target to a LabVIEW project.



The **Project Explorer** window includes the following sections:

- **Project root**—Contains the host computer and the myRIO target you add to the current project. The label on the project root includes the filename for the project. To add more myRIO targets to the project, right-click the project root and select **New»Targets and Devices** from the shortcut menu.
  - **My Computer**—Represents the local or host computer as a target in the project.
    - **Build Specifications**—Includes build configurations for source distributions and other types of builds available in LabVIEW toolkits and modules.

- **RT Target**—Represents the myRIO target you add to the project. VIs and libraries that you add to a myRIO target appear under this target in the **Project Explorer** window.

  To configure the settings of a myRIO target, right-click the target and select **Properties** from the shortcut menu to launch the **myRIO Properties** dialog box. You can also right-click the myRIO target and select different options to accomplish the following tasks: switching between FPGA personalities, creating a callback VI from templates, launching the **Getting Started with myRIO** wizard, and launching the **myRIO I/O Monitor**.

  - **FPGA Target**—Represents the FPGA target on the myRIO. To add the FPGA target under the RT target, right-click the chassis under the myRIO target and select **New»FPGA Target** from the shortcut menu.

    Note Once you add an FPGA target under the chassis, you must use an FPGA bitfile with the myRIO VIs.

  - **Build Specifications**—Includes specifications for building source distributions, stand-alone real-time applications, and zip files. If you have the LabVIEW Professional Development System or the Application Builder installed, right-click **Build Specifications** and select **New»Real-Time Application** from the shortcut menu to create a build specification that defines how to build stand-alone real-time applications.

Related Information

Managing a Project in LabVIEW

Adding RT Targets to a LabVIEW Project

Creating Stand-Alone Real-Time Applications

Real-Time Target

Choosing FPGA Personalities

Using Callback VIs to Handle Interrupts

Getting Started with myRIO Wizard

myRIO I/O Monitor

Using myRIO with Other NI Software (myRIO Toolkit)

Use the following topics to learn about creating applications on the myRIO with the LabVIEW myRIO Toolkit and other NI software:

- Acquiring and Processing Images
- Communicating with Serial Devices Using VISA
- Creating Feedback Control Systems
- Creating Robotics Applications
- Performing Textual Mathematics and Algorithm Design
- Simulating Dynamic Systems
- Visualizing Data on a Mobile Device

To view related topics, click the **Locate** button, shown at left, in the toolbar at the top of this window. The **LabVIEW Help** highlights this topic in the **Contents** tab so you can navigate the related topics.

Acquiring and Processing Images (myRIO Toolkit)

You can connect a USB camera to the USB host port on the myRIO to acquire and process images. You must install the following NI software to create image applications:

- **NI Vision Acquisition Software**—Allows you to acquire images from USB cameras that you connect to the myRIO.
- **NI Vision Development Module**—Helps you develop image processing applications, such as image pattern recognition, color sensing, light sensing, and object tracking.

After you connect a USB camera to the myRIO and install the software, select **Tools»Vision Assistant** in LabVIEW to launch the **Vision Assistant** to start acquiring and processing images.

Visit `ni.com/info` and enter the Info Code exwj7a to learn more about the Vision Acquisition Software and the Vision Development Module.

Communicating with Serial Devices Using VISA (myRIO Toolkit)

You can install NI-VISA to communicate with serial devices that you connect to the myRIO. NI-VISA is the National Instruments implementation of the VISA I/O standard. NI-VISA includes software libraries, interactive utilities, and configuration programs through Measurement & Automation Explorer (MAX) for various development needs.

Related Information

VISA VIs and Functions

Creating Feedback Control Systems (myRIO Toolkit)

You can use the myRIO with the PID VIs to create feedback control systems to control motors, temperature, pressure, and so on. To access the PID VIs, select **Control & Simulation»PID** from the Functions palette in LabVIEW.

Related Information

PID VIs

Functions Palette

Creating Robotics Applications (myRIO Toolkit)

The LabVIEW Robotics Module for myRIO and roboRIO provides you with select functionality from the LabVIEW Robotics Module to use with the myRIO or the roboRIO. You can use the myRIO with the Robotics Module for myRIO and roboRIO to develop and deploy robotics applications, such as steering, path planning, obstacle avoidance, and inverse kinematics.

To install the Robotics Module for myRIO and roboRIO, insert the LabVIEW myRIO Software Bundle DVD 1 and select **Robotics Module for myRIO and roboRIO** from the product list. After you install the Robotics Module for myRIO and roboRIO, you can find the Robotics Algorithms VIs by selecting **myRIO»Robotics Algorithms** from the Functions palette in LabVIEW.

Visit `ni.com/info` and enter the Info Code exb9ig to learn more about the Robotics Module for myRIO and roboRIO.

Related Information

LabVIEW myRIO Software Bundle Readme

Functions Palette

Performing Textual Mathematics and Algorithm Design (myRIO Toolkit)

You can use the LabVIEW MathScript RT Module to develop `.m` files with an interactive command-line interface and deploy `.m` files to the RT target on the myRIO. When you integrate this module with the LabVIEW Control Design and Simulation Module, you can perform textual mathematics and algorithm design in LabVIEW using the `.m` file syntax.

Visit `ni.com/info` and enter the Info Code exccik to learn more about the MathScript RT Module. Visit `ni.com/info` and enter the Info Code exu7rq to learn more about the Control Design and Simulation Module.

Simulating Dynamic Systems (myRIO Toolkit)

You can use the myRIO with the LabVIEW Control Design and Simulation Module to simulate dynamic systems, design sophisticated controllers, and use both classical and state-space approaches to design controllers and estimators.

After you install the Control Design and Simulation Module, you can find the Control Design and Simulation VIs and functions by selecting **Control & Simulation** from the Functions palette in LabVIEW.

Visit `ni.com/info` and enter the Info Code exu7rq to learn more about the Control Design and Simulation Module.

Related Information

Functions Palette

Visualizing Data on a Mobile Device (myRIO Toolkit)

You can install the Data Dashboard for LabVIEW app on an iPad or Android device to visualize the data that you acquire on the myRIO. Use this app to create a custom dashboard to control and monitor your LabVIEW application remotely. You can access data using secure LabVIEW web services and share dashboards with others from your iPad or Android device.

Refer to `Data Dashboard for myRIO.lvproj` in the `labview\examples\myRIO\Data Dashboard` directory for an example of using iPad-specific Data Dashboard with the myRIO.

Visit `ni.com/info` and enter the Info Code ext3du to learn more about Data Dashboard for LabVIEW.

myRIO VIs

June 2015, 373926C-01

Requires: myRIO Toolkit. This topic might not match its corresponding palette in LabVIEW depending on your operating system, licensed product(s), and target.

Use the myRIO VIs to create applications on the myRIO.

Complete the following steps to access the myRIO VIs:

1. Select **File»Create Project** from LabVIEW to display the **Create Project** dialog box.
2. Create a myRIO project by using the **Create Project** dialog box.
3. In the **Project Explorer** window of the myRIO project, right-click the myRIO target and select **New»VI** from the shortcut menu.
4. Select **Window»Show Block Diagram** to view the block diagram of the VI.

5. Select **View»Functions Palette** and navigate to the myRIO palette.

The VIs on this palette can return general LabVIEW error codes and myRIO and roboRIO error codes.

Palette Object	Description
Accelerometer	Reads acceleration values along the X, Y, and Z axes of the accelerometer on the myRIO or the roboRIO.
Analog Input	Reads values from one or more analog input channels on the myRIO or the roboRIO.
Analog Output	Writes values to one or more analog output channels on the myRIO or the roboRIO.
Button	Reads the value from the user button on the myRIO or the roboRIO.
Digital Input	Reads values from one or more digital input channels on the myRIO or the roboRIO.
Digital Output	Writes values to one or more digital output channels on the myRIO or the roboRIO.
Encoder	Reads and decodes signals from an encoder through the encoder channels on the myRIO or the roboRIO. This Express VI reads the number of ticks that the encoder receives since the last counter reset.
I2C	Writes data to or reads data from an Inter-Integrated Circuit (I2C) slave device through the I2C channels on the myRIO or the roboRIO.
Interrupt	Registers analog and digital input interrupts and creates timer interrupts on the myRIO or the roboRIO.
LED	Sets the states of the LEDs on the myRIO or the roboRIO.
PWM	Generates a pulse width modulation (PWM) signal to an external peripheral through the PWM channels on the myRIO or the roboRIO. The roboRIO uses 6 V voltage rail on the PWM port for powering servos and provides 5 V DIO lines for generating PWM signals.

SPI	Writes data to or reads data from a serial peripheral interface (SPI) slave device through the SPI channels on the myRIO or the roboRIO.
UART	Writes data to or reads data from a Universal Asynchronous Receiver/Transmitter (UART) device through the UART channels on the myRIO or the roboRIO. With the roboRIO, you also can use this VI to write data to or read data from an RS-232 device through the RS-232 channel.

Subpalette	Description
Device Management VIs	Use the Device Management VIs to set custom FPGA bitfiles and to reset I/O channels on the myRIO or the roboRIO.
High Throughput FPGA Personality VIs	Use the High Throughput FPGA Personality VIs to create applications on the myRIO with the high-throughput FPGA personality. The myRIO high-throughput FPGA personality supports high-speed analog or digital data access. You can use the high-throughput personality for audio signals and projects in need of waveform data.
Low Level VIs	Use the Low Level VIs to control the I/O channels on the myRIO or the roboRIO.

Accelerometer Express VI

Requires: myRIO Toolkit **or** roboRIO Toolkit

Reads acceleration values along the X, Y, and Z axes of the accelerometer on the myRIO or the roboRIO.

Dialog Box Options

Block Diagram Inputs

Block Diagram Outputs

Dialog Box Options

Parameter	Description

Node name	Specifies the name of this Express VI. You can also double-click the name of this Express VI on the expandable node to edit the name.
X-Axis	Specifies to read the acceleration value along the X axis.
Y-Axis	Specifies to read the acceleration value along the Y axis.
Z-Axis	Specifies to read the acceleration value along the Z axis.
Custom channel name	Specifies a custom name for the axis that you select.
View Code	Displays the underlying code of this Express VI.

Block Diagram Inputs

Parameter	Description
error in (no error)	Describes error conditions that occur before this node runs.

Block Diagram Outputs

Parameter	Description
X-Axis	Returns the acceleration value along the X axis.
Y-Axis	Returns the acceleration value along the Y axis.
Z-Axis	Returns the acceleration value along the Z axis.
error out	Contains error information. This output provides standard error out functionality.

Analog Input Express VI

Requires: myRIO Toolkit **or** roboRIO Toolkit

Reads values from one or more analog input channels on the myRIO or the roboRIO.

This Express VI reads one sample each time with the default FPGA personality on the myRIO. This Express VI reads one sample or multiple samples each time with the high-throughput FPGA personality on the myRIO. Visit `ni.com/info` and enter the Info Code ex6g5a to learn about the myRIO high-throughput FPGA personality.

This Express VI reads one sample each time with the default FPGA personality on the roboRIO. The roboRIO uses a 5 V voltage rail on the ANALOG IN port for powering sensors.

Details

Dialog Box Options

Parameter	Description
I/O mode	(myRIO Toolkit) Specifies to read one sample or multiple samples. The default is Analog input (1 sample). This option is available only when you use the myRIO high-throughput FPGA personality.
Node name	Specifies the name of this Express VI. You can also double-click the name of this Express VI on the expandable node to edit the name.
Channel	Specifies the analog input channel from which to read the values.
Custom channel name	Specifies a custom name for the analog input channel that you select.
Delete Channel	Deletes the analog input channel that you select. (myRIO Toolkit) If you use the myRIO high-throughput FPGA personality, this option is available only when you specify Analog input (1 sample) for **I/O mode**.
Add Channel	Adds a new analog input channel to the channel list. You can add up to 12 analog input channels for the myRIO. You can add up to eight analog input channels for the roboRIO. (myRIO Toolkit) If you use the myRIO high-throughput FPGA personality, this option is available only when you specify Analog input (1 sample) for **I/O mode**.
Sample rate	(myRIO Toolkit) Specifies the sampling frequency of the input signal. Valid values are between 1

	kHz and 50 kHz. If you specify a frequency that is invalid, this Express VI coerces the specified value to the nearest valid value when you click the **Validate** button. This option is available only when you use the myRIO high-throughput FPGA personality and specify Analog input (n samples) for **I/O mode**. ▪ **Frequency value**—Specifies the value of the sampling frequency. The default is 1. ▪ **Frequency unit**—Shows the unit of the sampling frequency. The value is kHz. ▪ **Validate**—Validates whether this Express VI can generate the sampling frequency that you specify. If the specified sampling frequency is not valid, this Express VI coerces the specified value to the nearest valid value.
Samples	(myRIO Toolkit) Specifies the number of samples to read. The default is 1,000. Valid values must be greater than 0 and less than or equal to 10,000. This option is available only when you use the myRIO high-throughput FPGA personality and specify Analog input (n samples) for **I/O mode**.
Latency	(myRIO Toolkit) Displays the latency between two adjacent signal acquisition iterations. Refer to the Details section of this topic for more information about latency. This option is available only when you use the myRIO high-throughput FPGA personality and specify Analog input (n samples) for **I/O mode**.
View Code	Displays the underlying code of this Express VI.
Connection Diagram	Shows the I/O connector pinouts on the myRIO or the roboRIO. The highlighted pinouts represent the channels that you configure.

Block Diagram Inputs

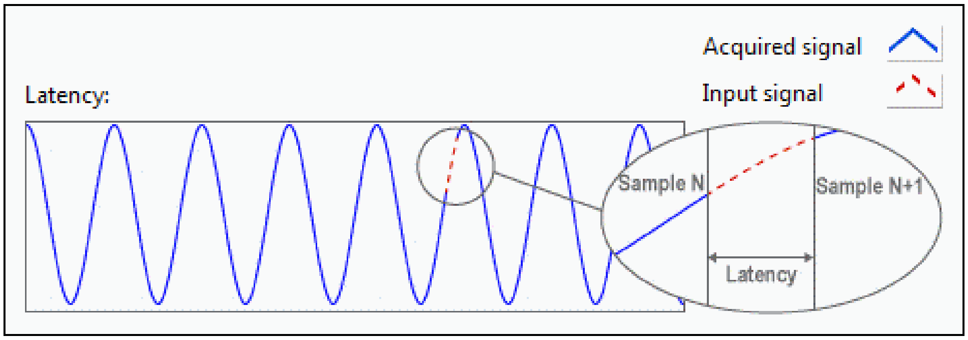
Parameter	Description
error in (no error)	Describes error conditions that occur before this node runs.

Block Diagram Outputs

Parameter	Description
channel name	Returns the value that this Express VI reads from the analog input channel that you select, where **channel name** is the name of the analog input channel.
error out	Contains error information. This output provide s standard error out functionality.

Analog Input Details

(myRIO Toolkit) The following figure demonstrates the latency when you use the Analog Input Express VI with the myRIO high-throughput FPGA personality to perform **n** samples read operations.



In the previous figure, the x-axis represents time and the y-axis represents amplitude. The waveform in blue represents the signal that the myRIO acquires. The red dotted line represents latency. When latency occurs, the myRIO does not acquire any signal. In other words, the time interval between two adjacent signal acquisition iterations is latency.

Related Information

1 Sample versus N Samples Modes (myRIO Toolkit)

Generating FPGA Clocks (myRIO Toolkit)

Generating FPGA Clocks (roboRIO Toolkit)

I/O Connectors (myRIO Toolkit)

I/O Connectors (roboRIO Toolkit)

Latency in N Samples Read and Write Operations (myRIO Toolkit)

Power Supply for Peripheral Devices (roboRIO Toolkit)

Analog Output Express VI

Requires: myRIO Toolkit **or** roboRIO Toolkit

Writes values to one or more analog output channels on the myRIO or the roboRIO.

This Express VI writes one sample each time with the default FPGA personality on the myRIO. This Express VI writes one sample or multiple samples each time with the high-throughput FPGA personality on the myRIO. Visit `ni.com/info` and enter the Info Code ex6g5a to learn about the myRIO high-throughput FPGA personality.

This Express VI writes one sample each time with the default FPGA personality on the roboRIO.

Details

Dialog Box Options
Block Diagram Inputs
Block Diagram Outputs

Dialog Box Options

Parameter	Description
I/O mode	(myRIO Toolkit) Specifies to write one sample or multiple samples. The default is Analog output (1 sample). This option is available only when you use the myRIO high-throughput FPGA personality.

Node name	Specifies the name of this Express VI. You can also double-click the name of this Express VI on the expandable node to edit the name.
Channel	Specifies the analog output channel to which to write a value.
Custom channel name	Specifies a custom name for the analog output channel that you select.
Delete Channel	Deletes the analog output channel that you select. (myRIO Toolkit) If you use the myRIO high-throughput FPGA personality, this option is available only when you specify Analog output (1 sample) for **I/O mode**.
Add Channel	Adds a new analog output channel to the channel list. You can add up to eight analog output channels for the myRIO. You can add up to two analog output channels for the roboRIO. (myRIO Toolkit) If you use the myRIO high-throughput FPGA personality, this option is available only when you specify Analog output (1 sample) for **I/O mode**.
Sample rate	(myRIO Toolkit) Specifies the sampling frequency of the output signal. Valid values are between 1 kHz and 80 kHz. If you specify a frequency that is invalid, this Express VI coerces the specified value to the nearest valid value when you click the **Validate** button. This option is available only when you use the myRIO high-throughput FPGA personality and specify Analog output (n samples) for **I/O mode**. - **Frequency value**—Specifies the value of the sampling frequency. The default is 1 - **Frequency unit**—Shows the unit of the sampling frequency. The value is kHz. - **Validate**—Validates whether this Express VI can generate the sampling frequency that you specify. If the specified sampling frequency is not valid, this Express VI coer

	ces the specified value to the nearest valid value.
Wait until done?	(myRIO Toolkit) Specifies whether this Express VI waits until the write operation completes. If the **Wait until done?** checkbox contains a checkmark, this Express VI waits until the write operation completes. By default, this checkbox does not contain a checkmark. This option is available only when you use the myRIO high-throughput FPGA personality and specify Analog output (n samples) for **I/O mode**.
Latency	(myRIO Toolkit) Displays the latency between two adjacent signal generation iterations. Refer to the Details section of this topic for more information about latency. This option is available only when you use the myRIO high-throughput FPGA personality and specify Analog output (n samples) for **I/O mode**.
View Code	Displays the underlying code of this Express VI.
Connection Diagram	Shows the I/O connector pinouts on the myRIO or the roboRIO. The highlighted pinouts represent the channels that you configure.

Block Diagram Inputs

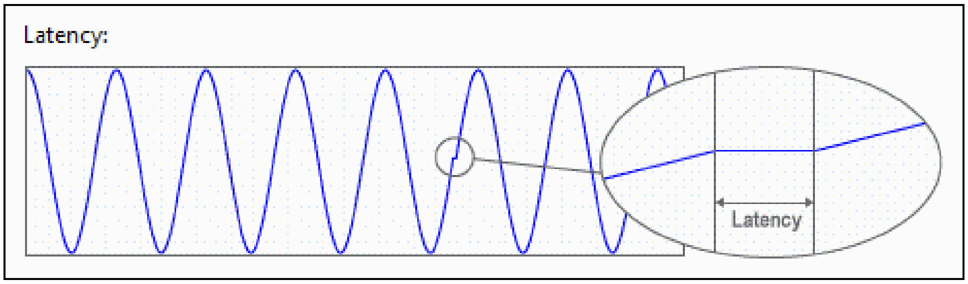
Parameter	Description
channel name	Specifies the value to write to the analog output channel that you select, where **channel name** is the name of the analog output channel.
error in (no error)	Describes error conditions that occur before this node runs.

Block Diagram Outputs

Parameter	Description
error out	Contains error information. This output provides standard error out functionality.

Analog Output Details

(myRIO Toolkit) The following figure demonstrates the latency when you use the Analog Output Express VI with the myRIO high-throughput FPGA personality to perform **n** samples write operations.



In the previous figure, the x-axis represents time and the y-axis represents amplitude. The time interval between two adjacent signal generation iterations is latency. In other words, the myRIO does not export signals when latency occurs.

Related Information

1 Sample versus N Samples Modes (myRIO Toolkit)

Generating FPGA Clocks (myRIO Toolkit)

Generating FPGA Clocks (roboRIO Toolkit)

I/O Connectors (myRIO Toolkit)

I/O Connectors (roboRIO Toolkit)

Latency in N Samples Read and Write Operations (myRIO Toolkit)

Power Supply for Peripheral Devices (roboRIO Toolkit)

Button Express VI

Requires: myRIO Toolkit **or** roboRIO Toolkit

Reads the value from the user button on the myRIO or the roboRIO.

Examples

Dialog Box Options
Block Diagram Inputs
Block Diagram Outputs

Dialog Box Options

Parameter	Description
Node name	Specifies the name of this Express VI. You can also double-click the name of this Express VI on the expandable node to edit the name.
View Code	Displays the underlying code of this Express VI.

Block Diagram Inputs

Parameter	Description
error in (no error)	Describes error conditions that occur before this node runs.

Block Diagram Outputs

Parameter	Description
Value	Returns the value this Express VI reads from the user button on the myRIO or the roboRIO.
error out	Contains error information. This output provides standard error out functionality.

Examples

Refer to the following VIs for examples of using the Button Express VI:

- `labview\examples\myRIO\Up and Down Binary Counter\Up and Down Binary Counter.lvproj`
- `labview\examples\roboRIO\Up and Down Binary Counter\Up and Down Binary Counter.lvproj`

Digital Input Express VI

Requires: myRIO Toolkit **or** roboRIO Toolkit

Reads values from one or more digital input channels on the myRIO or the roboRIO.

This Express VI reads one sample each time with the default FPGA personality on the myRIO. This Express VI reads one sample or multiple samples each time with the high-throughput FPGA personality on the myRIO. Visit `ni.com/info` and enter the Info Code ex6g5a to learn about the myRIO high-throughput FPGA personality.

This Express VI reads one sample each time with the default FPGA personality on the roboRIO. The roboRIO uses a 5 V voltage rail on the DIO port for powering sensors and provides 3.3 V DIO lines for generating digital input signals.

Details

Dialog Box Options

Parameter	Description
I/O mode	(myRIO Toolkit) Specifies to read one sample or multiple samples. The default is Digital input (1 sample). This option is available only when you use the myRIO high-throughput FPGA personalit y.
Node name	Specifies the name of this Express VI. You can al so double-click the name of this Express VI on th e expandable node to edit the name.
Channel	Specifies the digital input channel from which t o read the value. If the channel you select is set as a digital output channel, this Express VI chan ges the channel to a digital input channel before reading the value.
Custom channel name	Specifies a custom name for the digital input ch annel that you select.
Delete Channel	Deletes the digital input channel that you select . (myRIO Toolkit) If you use the myRIO high-thro ughput FPGA personality, this option is availabl e only when you specify Digital input (1 sample) for **I/O mode**.
Add Channel	Adds a new digital input channel to the channel list. You can add up to 12 digital input channels. (myRIO Toolkit) If you use the myRIO high-throu ghput FPGA personality, this option is available

	only when you specify Digital input (1 sample) for **I/O mode**.
Sample rate	(myRIO Toolkit) Specifies the sampling frequency of the input signal. Valid values are between 1 kHz and 8 MHz. If you specify a frequency that is invalid, this Express VI coerces the specified value to the nearest valid value when you click the **Validate** button. This option is available only when you use the myRIO high-throughput FPGA personality and specify Digital input (n samples) for **I/O mode**. - **Frequency value**—Specifies the value of the sampling frequency. The default is 1. - **Frequency unit**—Specifies the unit of the sampling frequency. The default is kHz. - **Validate**—Validates whether this Express VI can generate the sampling frequency that you specify. If the specified sampling frequency is not valid, this Express VI coerces the specified value to the nearest valid value.
Samples	(myRIO Toolkit) Specifies the number of samples to read. The default is 1,000. Valid values must be greater than 0 and less than or equal to 10,000. This option is available only when you use the myRIO high-throughput FPGA personality and specify Digital output (n samples) for **I/O mode**.
Latency	(myRIO Toolkit) Displays the latency between two adjacent signal acquisition iterations. Refer to the Details section of this topic for more information about latency. This option is available only when you use the myRIO high-throughput FPGA personality and specify Digital input (n samples) for **I/O mode**.
View Code	Displays the underlying code of this Express VI.

Connection Diagram	Shows the I/O connector pinouts on the myRIO or the roboRIO. The highlighted pinouts represent the channels that you configure.

Block Diagram Inputs

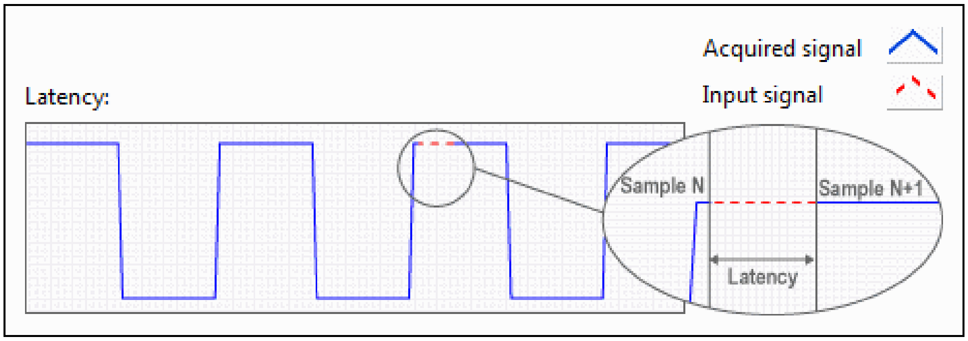
Parameter	Description
error in (no error)	Describes error conditions that occur before this node runs.

Block Diagram Outputs

Parameter	Description
channel name	Returns the value that this Express VI reads from the digital input channel that you select, where **channel name** is the name of the digital input channel.
error out	Contains error information. This output provides standard error out functionality.

Digital Input Details

(myRIO Toolkit) The following figure demonstrates the latency when you use the Digital Input Express VI with the myRIO high-throughput FPGA personality to perform **n** samples read operations.



In the previous figure, the x-axis represents time and the y-axis represents amplitude. The waveform in blue represents the signal that the myRIO acquires. The red dotted line represents latency. When latency occurs, the myRIO does not acquire any signal. In other words, the time interval between two adjacent signal acquisition iterations is latency.

Related Information

1 Sample versus N Samples Modes (myRIO Toolkit)

Generating FPGA Clocks (myRIO Toolkit)

Generating FPGA Clocks (roboRIO Toolkit)

I/O Connectors (myRIO Toolkit)

I/O Connectors (roboRIO Toolkit)

Latency in N Samples Read and Write Operations (myRIO Toolkit)

Power Supply for Peripheral Devices (roboRIO Toolkit)

Digital Output Express VI

Requires: myRIO Toolkit **or** roboRIO Toolkit

Writes values to one or more digital output channels on the myRIO or the roboRIO.

This Express VI writes one sample each time with the default FPGA personality on the myRIO. This Express VI writes one sample or multiple samples each time with the high-throughput FPGA personality on the myRIO. Visit `ni.com/info` and enter the Info Code ex6g5a to learn about the myRIO high-throughput FPGA personality.

This Express VI writes one sample each time with the default FPGA personality on the roboRIO. The roboRIO uses a 5 V voltage rail on the DIO port for powering sensors and provides 3.3 V DIO lines for generating digital output signals.

Details

Dialog Box Options
Block Diagram Inputs
Block Diagram Outputs

Dialog Box Options

Parameter	Description
I/O mode	(myRIO Toolkit) Specifies to write one sample or multiple samples. The default is Digital output (

	1 sample). This option is available only when you use the myRIO high-throughput FPGA personality.
Node name	Specifies the name of this Express VI. You can also double-click the name of this Express VI on the expandable node to edit the name.
Channel	Specifies the digital output channel to which to write a value. If the channel you select is set as a digital input channel, this Express VI changes the channel to a digital output channel before writing the value.
Custom channel name	Specifies a custom name for the digital output channel that you select.
Delete Channel	Deletes the digital output channel that you select. (myRIO Toolkit) If you use the myRIO high-throughput FPGA personality, this option is available only when you specify Digital output (1 sample) for **I/O mode**.
Add Channel	Adds a new digital output channel to the channel list. You can add up to 12 digital output channels. (myRIO Toolkit) If you use the myRIO high-throughput FPGA personality, this option is available only when you specify Digital output (1 sample) for **I/O mode**.
Sample rate	(myRIO Toolkit) Specifies the sampling frequency of the output signal. Valid values are between 1 kHz and 8 MHz. If you specify a frequency that is invalid, this Express VI coerces the specified value to the nearest valid value when you click the **Validate** button. This option is available only when you use the myRIO high-throughput FPGA personality and specify Digital output (n samples) for **I/O mode**. - **Frequency value**—Specifies the value of the sampling frequency. The default is 1. - **Frequency unit**—Specifies the unit of the sampling frequency. The default is kHz.

	▪ **Validate**—Validates whether this Express VI can generate the sampling frequency that you specify. If the specified sampling frequency is not valid, this Express VI coerces the specified value to the nearest valid value.
Wait until done?	(myRIO Toolkit) Specifies whether this Express VI waits until the write operation completes. If the **Wait until done?** checkbox contains a checkmark, this Express VI waits until the write operation completes. By default, this checkbox does not contain a checkmark. This option is available only when you use the myRIO high-throughput FPGA personality and specify Analog output (n samples) for **I/O mode**.
Latency	(myRIO Toolkit) Displays the latency between two adjacent signal generation iterations. Refer to the Details section of this topic for more information about latency. This option is available only when you use the myRIO high-throughput FPGA personality and specify Digital output (n samples) for **I/O mode**.
View Code	Displays the underlying code of this Express VI.
Connection Diagram	Shows the I/O connector pinouts on the myRIO or the roboRIO. The highlighted pinouts represent the channels that you configure.

Block Diagram Inputs

Parameter	Description
channel name	Specifies the value to write to the digital output channel that you select, where **channel name** is the name of the digital output channel.
error in (no error)	Describes error conditions that occur before this node runs.

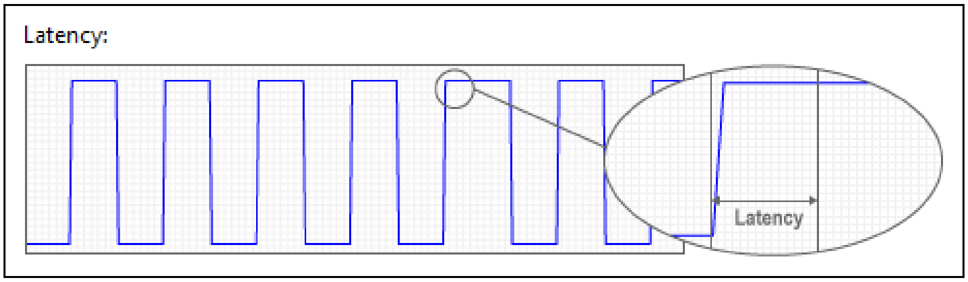
Block Diagram Outputs

Parameter	Description

error out	Contains error information. This output provide s standard error out functionality.

Digital Output Details

(myRIO Toolkit) The following figure demonstrates the latency when you use the Digital Output Express VI with the myRIO high-throughput FPGA personality to perform **n** samples write operations.



In the previous figure, the x-axis represents time and the y-axis represents amplitude. The time interval between two adjacent signal generation iterations is latency. In other words, the myRIO does not export signals when latency occurs.

Related Information

1 Sample versus N Samples Modes (myRIO Toolkit)

Generating FPGA Clocks (myRIO Toolkit)

Generating FPGA Clocks (roboRIO Toolkit)

I/O Connectors (myRIO Toolkit)

I/O Connectors (roboRIO Toolkit)

Latency in N Samples Read and Write Operations (myRIO Toolkit)

Power Supply for Peripheral Devices (roboRIO Toolkit)

Encoder Express VI

Requires: myRIO Toolkit **or** roboRIO Toolkit

Reads and decodes signals from an encoder through the encoder channels on the myRIO or the roboRIO. This Express VI reads the number of ticks that the encoder receives since the last counter reset.

Details

Dialog Box Options
Block Diagram Inputs
Block Diagram Outputs

Dialog Box Options

Parameter	Description
Node name	Specifies the name of this Express VI. You can also double-click the name of this Express VI on the expandable node to edit the name.
Channel	Specifies the encoder channel to read and decode signals from the encoder.
Connections	Shows the myRIO or the roboRIO pins that correspond to the encoder signals.
Encoder output signal type	Specifies the type of output signal from the encoder you use. **Encoder output signal type** contains the following options: - **Quadrature phase signal**—Specifies that the encoder generates two phase signals that are offset by 90 degrees. The count value changes each time there is a falling or rising edge on either of the phases. Most encoders generate quadrature phase signals. You can interpret a quadrature phase signal as Gray code. - **Step and direction signals**—Specifies that the encoder generates a direction signal and a clock signal. The direction signal determines the direction of the encoder. The count value changes on every rising edge of the clock signal.
View Code	Displays the underlying code of this Express VI.
Connection Diagram	Shows the I/O connector pinouts on the myRIO or the roboRIO. The highlighted pinouts represent the channels that you configure.

Block Diagram Inputs

Parameter	Description
Reset Counter	Specifies whether to reset the encoder tick counter to zero. The default is FALSE.
error in (no error)	Describes error conditions that occur before this node runs.

Block Diagram Outputs

Parameter	Description
Counter Value	Returns the number of ticks that this Express VI reads from the encoder since the last counter reset. **Counter Value** must be in the range from -2,147,483,648 to 2,147,483,647.
Counter Direction	Returns the direction of the counter between the last two ticks that the encoder receives.
Overflow?	Returns whether the counter value wraps back to zero. **Overflow?** returns TRUE when the value of the counter goes from the maximum value to the minimum value or from the minimum value to the maximum value. After this Express VI reads the count value once, **Overflow?** changes to FALSE until the counter value wraps back to zero again.
error out	Contains error information. This output provides standard error out functionality.

Encoder Details

Related Information

I/O Connectors (myRIO Toolkit)

I/O Connectors (roboRIO Toolkit)

I2C Express VI

Requires: myRIO Toolkit **or** roboRIO Toolkit

Writes data to or reads data from an Inter-Integrated Circuit (I2C) slave device through the I2C channels on the myRIO or the roboRIO.

The roboRIO uses a 3.3 V voltage rail on the I2C port for powering I2C peripherals and provides 3.3 V DIO lines for generating I2C signals.

Details

Dialog Box Options

Block Diagram Inputs

Block Diagram Outputs

Dialog Box Options

Parameter	Description
Node name	Specifies the name of this Express VI. You can al so double-click the name of this Express VI on th e expandable node to edit the name.
Channel	Specifies the I2C channel to which to write data to or read data from the I2C slave device.
Connections	Shows the myRIO or the roboRIO pins that corre spond to the data line (SDA) and clock line (SCL) .
Mode	Specifies the mode of operation that this Expres s VI uses to communicate with the I2C slave devi ce. **Mode** contains the following options: ▪ **Write**—Specifies that this Express VI wr ites data to the I2C slave device. ▪ **Read**—Specifies that this Express VI rea ds data from the I2C slave device. ▪ **Write/Read**—Specifies that this Expres s VI writes data to the I2C slave device and then reads a specified number of bytes fro m the I2C slave device.
Speed	Specifies the transfer rate of the I2C channel. Th e default is Standard mode (100 kbps).
View Code	Displays the underlying code of this Express VI.

Connection Diagram	Shows the I/O connector pinouts on the myRIO or the roboRIO. The highlighted pinouts represent the channels that you configure.

Block Diagram Inputs

Parameter	Description
Slave Address (7-bit)	Specifies the address of the I2C slave device which this Express VI reads data from or writes data to. You must specify the address in 7 bits. Some I2C devices might have an 8-bit address in which the first 7 bits represent the address and the last bit represents the mode of operation. For this kind of I2C devices, you must specify **Slave Address (7-bit)** using the seven most significant bits.
Bytes to Write	Specifies the data bytes to write to the I2C slave device. This input is available when you set **Mode** to **Write** or **Write/Read**.
Byte Count	Specifies the number of bytes to write to the I2C slave device. This input is available when you set **Mode** to **Read** or **Write/Read**.
error in (no error)	Describes error conditions that occur before this node runs.

Block Diagram Outputs

Parameter	Description
Bytes Read	Returns the data bytes that this Express VI reads from the I2C slave device. This output is available when you set **Mode** to **Read** or **Write/Read**.
error out	Contains error information. This output provides standard error out functionality.

I2C Details

Related Information

I/O Connectors (myRIO Toolkit)

I/O Connectors (roboRIO Toolkit)

Power Supply for Peripheral Devices (roboRIO Toolkit)

DIO Lines (roboRIO Toolkit)

Interrupt Express VI

Requires: myRIO Toolkit **or** roboRIO Toolkit

Registers analog and digital input interrupts and creates timer interrupts on the myRIO or the roboRIO.

Details Examples

Dialog Box Options

Block Diagram Inputs

Block Diagram Outputs

Dialog Box Options

Parameter	Description
I/O mode	Specifies the I/O mode this Express VI uses. The default is Analog input interrupt. Other options are Digital input interrupt and Timer interrupt. **Note** When you specify Analog input interrupt for **I/O mode**, this Express VI uses a 0.02 V hysteresis to avoid false interrupt registration.
Node name	Specifies the name of this Express VI. You can also double-click the name of this Express VI on the expandable node to edit the name.
IRQ number	Specifies the identifier of the interrupt to register. The default is 0. Valid values are within the range [0, 7]. This option is available only when you specify Analog input interrupt or Digital input interrupt for **I/O mode**.

	Note You cannot register an I/O interrupt with the same **IRQ number** as a registered I/O interrupt. However, after you unregister the existing interrupt, you can use the **IRQ number** to register another interrupt.
Channel	Specifies the channel to register and create the interrupt.
Type	Specifies when to register or create an interrupt based on the signal. This option is available only when you specify Analog input interrupt or Digital input interrupt for **I/O mode**. When you specify Analog input interrupt for **I/O mode**, you can specify the following values for **Type**: ▪ **Analog falling edge** (default)—Specifies to register an interrupt on a falling edge of the analog input signal. ▪ **Analog rising edge**—Specifies to register an interrupt on a rising edge of the analog input signal. When you specify Digital input interrupt for **I/O mode**, you can specify the following values for **Type**: ▪ **Digital falling edge** (default)—Specifies to register an interrupt on a falling edge of the digital input signal. ▪ **Digital rising edge**—Specifies to register an interrupt on a rising edge of the digital input signal.

	- **Digital edge**—Specifies to register an interrupt both on a falling edge and on a rising edge of the digital input signal.
Threshold	Specifies the value in volts that the signal must cross for this Express VI to register an interrupt. The default is 2.5. This option is available only when you specify Analog input interrupt for **I/O mode**.
Edge count	Specifies the number of edges of the signal that must occur for this Express VI to register an interrupt. The default is 1. This option is available only when you specify Digital input interrupt for **I/O mode**.
Timer ID	Specifies the identifier of the interrupt to create. The default is 0. Valid values are within the range [0, 7]. This option is available only when you specify Timer interrupt for **I/O mode**. **Note** You cannot create a timer interrupt with the same **Timer ID** as an existing timer interrupt. However, after you destroy the existing interrupt, you can use the **Timer ID** to create another interrupt.
Interval	Specifies the span of time between interrupts in microseconds. The default is 1,000,000. This option is available only when you specify Timer interrupt for **I/O mode**. **Note** Ensure the value of **Interval** is larger than 500. This is because a latency exists between the actual interval and specified interval. Typically, this latency is 5

	00 µs. This value might vary because of jitter.
Callback VI	Specifies the path to the callback VI. When you specify **Callback VI**, you can specify a path either to an existing or new callback VI. If you specify a path to a new callback VI, LabVIEW creates a callback VI with the specified name in the specified directory. **Note** If you create a callback VI that has the same name as another VI that already exists in memory or in the project, LabVIEW returns errors.
Create	Displays a file dialog box where you can create a new callback VI from template.
Browse	Displays a file dialog box where you can select an existing callback VI.
Preview	Displays a preview of when this Express VI registers or creates interrupts in different I/O modes.
View Code	Displays the underlying code of this Express VI.
Connection Diagram	Shows the I/O connector pinouts on the myRIO or the roboRIO. The highlighted pinouts represent the channels that you configure.

Block Diagram Inputs

Parameter	Description
error in (no error)	Describes error conditions that occur before this node runs.

Block Diagram Outputs

Parameter	Description
error out	Contains error information. This output provides standard error out functionality.

Interrupt Details

Related Information

Understanding Hysteresis (myRIO Toolkit)

Understanding Hysteresis (roboRIO Toolkit)

Using Callback VIs (myRIO Toolkit)

Using Callback VIs (roboRIO Toolkit)

Examples

Refer to the following VIs for examples of using the Interrupt Express VI:

- `labview\examples\myRIO\Interrupt Handling\Interrupt Handling.lvproj`
- `labview\examples\roboRIO\Interrupt Handling\Interrupt Handling.lvproj`

LED Express VI

Requires: myRIO Toolkit **or** roboRIO Toolkit

Sets the states of the LEDs on the myRIO or the roboRIO.

Details

Dialog Box Options
Block Diagram Inputs
Block Diagram Outputs

Dialog Box Options

Parameter	Description
Node name	Specifies the name of this Express VI. You can also double-click the name of this Express VI on the expandable node to edit the name.
LED0	Enables state setting for LED0 on the myRIO.
LED1	Enables state setting for LED1 on the myRIO.

LED2	Enables state setting for LED2 on the myRIO.
LED3	Enables state setting for LED3 on the myRIO.
RADIO (Green)	Enables state setting for RADIO (Green) on the roboRIO.
RADIO (Red)	Enables state setting for RADIO (Red) on the roboRIO.
COMM (Green)	Enables state setting for COMM (Green) on the roboRIO.
COMM (Red)	Enables state setting for COMM (Red) on the roboRIO.
MODE (Green)	Enables state setting for MODE (Green) on the roboRIO.
MODE (Red)	Enables state setting for MODE (Red) on the roboRIO.
Custom channel name	Specifies a custom name for the LED that you select.
View Code	Displays the underlying code of this Express VI.

Block Diagram Inputs

Parameter	Description
LED0	Sets the state of LED0 on the myRIO.
LED1	Sets the state of LED1 on the myRIO.
LED2	Sets the state of LED2 on the myRIO.
LED3	Sets the state of LED3 on the myRIO.
RADIO (Green)	Sets the state of RADIO (Green) on the roboRIO.
RADIO (Red)	Sets the state of RADIO (Red) on the roboRIO.
COMM (Green)	Sets the state of COMM (Green) on the roboRIO.
COMM (Red)	Sets the state of COMM (Red) on the roboRIO.
MODE (Green)	Sets the state of MODE (Green) on the roboRIO.
MODE (Red)	Sets the state of MODE (Red) on the roboRIO.
error in (no error)	Describes error conditions that occur before this node runs.

Block Diagram Outputs

Parameter	Description
error out	Contains error information. This output provide s standard error out functionality.

LED Details

You can set either ON or OFF state for the myRIO LEDs.

The roboRIO contains the RADIO, COMM, and MODE LEDs. Every LED associates with two Boolean controls for displaying various colors. For example, the RADIO LED associates with the **RADIO (Green)** control and the **RADIO (Red)** control. The following table shows the configuration options to display a green, red, or orange RADIO LED:

	Green	Red	Orange	Off
RADIO (Green)	TRUE	FALSE	TRUE	FALSE
RADIO (Red)	FALSE	TRUE	TRUE	FALSE

PWM Express VI

Requires: myRIO Toolkit **or** roboRIO Toolkit

Generates a pulse width modulation (PWM) signal to an external peripheral through the PWM channels on the myRIO or the roboRIO.

The roboRIO uses a 6 V voltage rail on the PWM port for powering servos and provides 5 V DIO lines for generating PWM signals.

Details

Dialog Box Options
Block Diagram Inputs
Block Diagram Outputs

Dialog Box Options

Parameter	Description

Node name	Specifies the name of this Express VI. You can also double-click the name of this Express VI on the expandable node to edit the name.
Channel	Specifies the channel to generate the PWM signal.
Frequency	Specifies the frequency settings for the PWM signal. If you specify a frequency that is invalid, this Express VI coerces the specified value to the nearest valid value when you click the **Validate** button. **Frequency** contains the following options: - **Set using input to Express VI**—Specifies to set the frequency by using the **Frequency [Hz]** block diagram input. This option allows you to set the frequency at run time. - **Set constant**—Specifies to use a constant frequency value. - **Frequency value**—Specifies the value of the frequency. The default is 1,000. - **Frequency unit**—Specifies the unit of the frequency. The default is Hz. - **Validate**—Validates whether this Express VI can generate the frequency that you specify. If the specified frequency is not valid, this Express VI coerces the specified value to the nearest valid value.
Duty cycle	Specifies the percentage of time the PWM signal remains high over one PWM cycle. **Duty cycle** contains the following options: - **Set using input to Express VI**—Specifies to set the duty cycle by using the **Duty Cycle** block diagram input. This option allows you to set the duty cycle at run time.

	▪ **Set constant**—Specifies to use a constant duty cycle value. ▪ **Duty cycle value**—Specifies the value of the duty cycle. The default is 0.5.
Output preview	Displays a preview of the output PWM signal.
View Code	Displays the underlying code of this Express VI.
Connection Diagram	Shows the I/O connector pinouts on the myRIO or the roboRIO. The highlighted pinouts represent the channels that you configure.

Block Diagram Inputs

Parameter	Description
Duty Cycle	Specifies the percentage of time the PWM signal remains high over one PWM cycle. Valid values must be within the range [0, 1].
Frequency [Hz]	Specifies the frequency in hertz of the PWM signal.
error in (no error)	Describes error conditions that occur before this node runs.

Block Diagram Outputs

Parameter	Description
error out	Contains error information. This output provides standard error out functionality.

PWM Details

Related Information

I/O Connectors (myRIO Toolkit)

I/O Connectors (roboRIO Toolkit)

Generating FPGA Clocks (myRIO Toolkit)

Generating FPGA Clocks (roboRIO Toolkit)

Power Supply for Peripheral Devices (roboRIO Toolkit)

DIO Lines (roboRIO Toolkit)

SPI Express VI

Requires: myRIO Toolkit **or** roboRIO Toolkit

Writes data to or reads data from a serial peripheral interface (SPI) slave device through the SPI channels on the myRIO or the roboRIO.

In addition to the SPI channels on the MXP port, the roboRIO also has an SPI port. The SPI port contains four chip select (CS) lines to support up to four slave devices. You also can use the CS lines as DIO lines. The roboRIO uses a 3.3 V voltage rail on the SPI port for powering SPI peripherals and provides 3.3 V DIO lines for generating SPI signals.

Details

Dialog Box Options

Parameter	Description
Node name	Specifies the name of this Express VI. You can also double-click the name of this Express VI on the expandable node to edit the name.
Channel	Specifies the SPI channel to which to write data to or read data from an SPI slave device.
Connections	Shows the myRIO or roboRIO pins that correspond to the SPI logic signals.
Mode	Specifies the mode of operation that this Express VI uses to communicate with the SPI slave device. **Mode** contains the following options: - **Write**—Specifies that this Express VI writes data to the SPI slave device. - **Read**—Specifies that this Express VI reads data from the SPI slave device.

	▪ **Write/Read**—Specifies that this Express VI writes data to and reads data from the SPI slave device at the same time.
Frequency	Specifies the frequency of the generated clock signal. If you specify a frequency that is invalid, this Express VI coerces the specified value to the nearest valid value when you click the **Validate** button. ▪ **Frequency value**—Specifies the value of the frequency. The default is 1. ▪ **Frequency unit**—Specifies the unit of the frequency. The default is MHz. ▪ **Validate**—Validates whether this Express VI can generate the frequency that you specify. If the specified frequency is not valid, this Express VI coerces the specified value to the nearest valid value.
Frame length	Specifies the number of bits that make up one SPI transmission frame. The default is 8 bits.
Advanced options	Specifies advanced configuration options for communicating with the SPI slave device. ▪ **Clock phase**—Specifies the clock phase at which the data remains stable in the SPI transmission cycle. The default is Leading, which means the data is stable on the leading edge and changes on the trailing edge. The other option is Trailing, which means the data is stable on the trailing edge and changes on the leading edge. ▪ **Clock polarity**—Specifies the base level of the clock signal and the logic level of the leading and trailing edges. The default is Low, which means the clock signal is low when idling, the leading edge is a rising edge, and the trailing edge is a falling edge. The other option is High, which means the clock signal is high when idling, the lea

	ding edge is a falling edge, and the trailing edge is a rising edge. ▪ **Data direction**—Specifies the order in which the bits in the SPI frame are transmitted. The default is Most significant bit first, which specifies to send the most significant bit first and the least significant bit last. The other option is Least significant bit first, which specifies to send the least significant bit first and the most significant bit last.
View Code	Displays the underlying code of this Express VI.
Connection Diagram	Shows the I/O connector pinouts on the myRIO or the roboRIO. The highlighted pinouts represent the channels that you configure.

Block Diagram Inputs

Parameter	Description
Frames to Write	Specifies the data to write to the SPI slave device. You can write multiple frames to the device at the same time. This input is available when you set **Mode** to **Write** or **Write/Read**.
Frame Count	Specifies the number of frames to read from the SPI slave device. This input is available when you set **Mode** to **Read**.
error in (no error)	Describes error conditions that occur before this node runs.

Block Diagram Outputs

Parameter	Description
Frames Read	Returns the data frames that this Express VI reads from the SPI channel. This output is available when you set **Mode** to **Read** or **Write/Read**.
error out	Contains error information. This output provides standard error out functionality.

SPI Details

Related Information

I/O Connectors (myRIO Toolkit)

I/O Connectors (roboRIO Toolkit)

Generating FPGA Clocks (myRIO Toolkit)

Generating FPGA Clocks(roboRIO Toolkit)

Power Supply for Peripheral Devices (roboRIO Toolkit)

DIO Lines (roboRIO Toolkit)

UART Express VI

Requires: myRIO Toolkit **or** roboRIO Toolkit

Writes data to or reads data from a Universal Asynchronous Receiver/Transmitter (UART) device through the UART channels on the myRIO or the roboRIO.

With the roboRIO, you also can use this VI to write data to or read data from an RS-232 device through the RS-232 channel.

Details

Dialog Box Options
Block Diagram Inputs
Block Diagram Outputs

Dialog Box Options

Parameter	Description
Node name	Specifies the name of this Express VI. You can al so double-click the name of this Express VI on th e expandable node to edit the name.
Channel	Specifies the UART channel on the myRIO or the roboRIO to write data to or read data from the U ART device. With the roboRIO, you also can specifies the RS-

	232 channel to write data to or read data from the RS-232 device.
Connections	Specifies the myRIO or the roboRIO pins that correspond to the receive input line and the transmit output line.
Mode	Specifies the mode of operation for communicating with the UART device. **Mode** contains the following options: ▪ **Write**—Specifies to write data to the UART device. With the roboRIO, you also can specify to write data to the RS-232 device. ▪ **Read**—Specifies to read data from the UART device. With the roboRIO, you also can specify to read data from the RS-232 device. ▪ **Read all available**—Specifies whether to read all available characters from the UART device. With the roboRIO, you also can specify whether to read all available characters from the RS-232 device . The default is FALSE.
Communication settings	Specifies the configuration for communicating with the UART device. With the roboRIO, you also can specify the configuration for communicating with the RS-232 device. **Communication settings** contains the following options: ▪ **Baud rate**—Specifies the baud rate of transmission. The default is 9,600. The maximum baud rate is 230,400 for UART lines and 115,200 for RS-232 lines. ▪ **Data bits**—Specifies the number of bits in the incoming data. The default is 8. ▪ **Parity**—Specifies the parity bits to write or read characters. The default is None.

	▪ **Stop bits**—Specifies the number of stop bits this Express VI uses to indicate the end of a data frame. The default is 1.0.
View Code	Displays the underlying code of this Express VI.
Connection Diagram	Shows the I/O connector pinouts on the myRIO or the roboRIO. The highlighted pinouts represent the channels that you configure.

Block Diagram Inputs

Parameter	Description
Characters to Write	Specifies the characters to write to the UART device. With the roboRIO, you also can specify the characters to write to the RS-232 device. This input is available only when you set **Mode** to **Write**.
Character Count	Specifies the number of characters to read from the UART device. With the roboRIO, you also can specify the number of characters to read from the RS-232 device. This input is available only when you set **Mode** to **Read**.
error in (no error)	Describes error conditions that occur before this node runs.

Block Diagram Outputs

Parameter	Description
Character Count	Returns the number of characters that this Express VI reads from the UART device. With the roboRIO, this output also returns the number of characters that this Express VI reads from the RS-232 device. This output is available when you set **Mode** to **Write**.
Characters Read	Returns the characters that this Express VI reads from the UART device. With the roboRIO, this output also returns the characters that this Express VI reads from the RS-232 device. This output is available when you set **Mode** to **Read**.

error out	Contains error information. This output provides standard error out functionality.

UART Details

The UART lines on the myRIO and roboRIO MXP ports are electrically identical to DIO lines on the MXP port. The UART signals are transistor-transistor logic (TTL) compatible and have the following characteristics:

- Logic low—0 V to 0.8 V
- Logic high—2 V to 5 V

The RS-232 lines on the roboRIO are compliant with TIA/EIA-232-F voltage levels. The following are the valid voltage levels:

- Logic one—-5 V to -15 V
- Logic zero—+5 V to +15 V

Related Information

I/O Connectors (myRIO Toolkit)

I/O Connectors (roboRIO Toolkit)

TTL-Compatible Signals

Serial Port Communication

Serial VIs and Functions

Device Management VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit. This topic might not match its corresponding palette in LabVIEW depending on your operating system, licensed product(s), and target.

Use the Device Management VIs to set custom FPGA bitfiles and to reset I/O channels on the myRIO or the roboRIO.

Palette Object	Description

Reset	Resets the FPGA target and all the I/O channels on the myRIO or the roboRIO.
Set Custom Bitfile	Sets a custom FPGA reference.

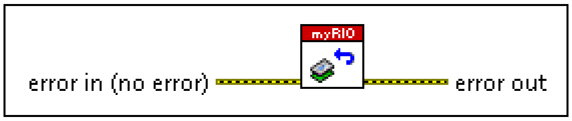
Reset VI

Owning Palette: Device Management VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Resets the FPGA target and all the I/O channels on the myRIO or the roboRIO.

Details Examples



error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

error out contains error information. This output provides standard error out functionality.

Reset Details

This VI resets the FPGA target even when code is running on the FPGA target and no matter whether there are incoming errors. Use this VI only with the Express VIs and run this VI only once at the end of an application.

Examples

Refer to the following VIs for examples of using the Reset VI:

- `labview\examples\myRIO\Data Dashboard\Data Dashboard for myRIO.lvproj`
- `labview\examples\myRIO\Edge Detection and Debouncing\Edge Detection and Debouncing.lvproj`

- `labview\examples\myRIO\Up and Down Binary Counter\Up and Down Binary Counter.lvproj`
- `labview\examples\roboRIO\Edge Detection and Debouncing\Edge Detection and Debouncing.lvproj`
- `labview\examples\roboRIO\Up and Down Binary Counter\Up and Down Binary Counter.lvproj`

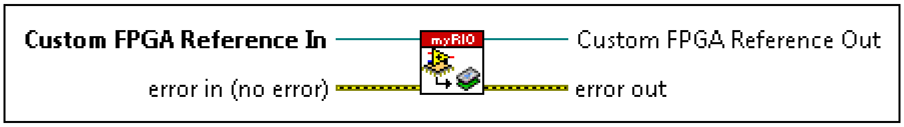
Set Custom Bitfile VI

Owning Palette: Device Management VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Sets a custom FPGA reference.

Details Examples



Custom FPGA Reference In specifies the input custom FPGA reference.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

Custom FPGA Reference Out returns the custom FPGA reference.

error out contains error information. This output provides standard error out functionality.

Set Custom Bitfile Details

You must set a custom FPGA reference before using a custom FPGA bitfile with the myRIO VIs or the roboRIO VIs. Use the Open FPGA VI Reference function to open a reference to the custom FPGA bitfile. Use the Close FPGA VI Reference function to close the reference at the end of an application.

Related Information

Open FPGA VI Reference Function

Close FPGA VI Reference Function

Examples

Refer to the following VIs for examples of using the Set Custom Bitfile VI:

- `labview\examples\myRIO\Customized FPGA\Customized FPGA Signal Generator.lvproj`
- `labview\examples\roboRIO\Customized FPGA\Customized FPGA Signal Generator.lvproj`

Low Level VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit. This topic might not match its corresponding palette in LabVIEW depending on your operating system, licensed product(s), and target.

Use the Low Level VIs to control the I/O channels on the myRIO or the roboRIO.

Subpalette	Description
Accelerometer VIs	Use the Accelerometer VIs to control the onboard accelerometer on the myRIO or the roboRIO.
Analog Input 1 Sample VIs	Use the Analog Input 1 Sample VIs to control the analog input channels on the myRIO or the roboRIO.
Analog Output 1 Sample VIs	Use the Analog Output 1 Sample VIs to control the analog output channels on the myRIO or the roboRIO.
Digital Input/Output 1 Sample VIs	Use the Digital Input/Output 1 Sample VIs to control the digital I/O channels on the myRIO or the roboRIO.
Encoder VIs	Use the Encoder VIs to control the encoder channels on the myRIO or the roboRIO.

I2C VIs	Use the I2C VIs to control the Inter-Integrated Circuit (I2C) channels on the myRIO or the roboRIO.
Interrupt VIs	Use the Interrupt VIs to manage interrupts on the myRIO or the roboRIO. You can register and unregister analog and digital input interrupts. You can also create and destroy timer interrupts.
PWM VIs	Use the PWM VIs to control the pulse width modulation (PWM) channels on the myRIO or the roboRIO.
Relay VIs	Use the Relay VIs to control the relay channels on the roboRIO.
RSL VIs	Use the RSL VIs to control the robot signal light (RSL) channel on the roboRIO.
SPI VIs	Use the SPI VIs to control the serial peripheral interface (SPI) channels on the myRIO or the roboRIO.

This palette also contains the following subpalettes:

- Input Device Control VIs
- Serial VIs and Functions
- Embedded CAN for RIO VIs

Related Information

Choosing between Express VIs and Low Level VIs (myRIO Toolkit)

Choosing between Express VIs and Low Level VIs (roboRIO Toolkit)

Choosing FPGA Personalities (myRIO Toolkit)

Understanding FPGA Personalities (roboRIO Toolkit)

Low Level References (myRIO Toolkit and roboRIO Toolkit)

The low level references are clusters of data that the Low Level VIs use to store and pass configuration data. You must use one of the Open VIs to open a reference to a channel before you use the channel.

Note Do not manually modify the low level references.

Accelerometer Reference

Note This reference is available when you use either the myRIO Toolkit or the roboRIO Toolkit.

myRIO Reference contains the reference to the myRIO or the roboRIO.

- **myRIO Model** specifies the version of the myRIO or the roboRIO.
- **myRIO Hardware Reference** contains reference information about the myRIO or the roboRIO.
  - **Generic FPGA Reference** contains the reference to the FPGA target on the myRIO or the roboRIO.
  - **Allow multiple opens** defines whether to allow opening

the specified channels more than once.

Accelerometer Channels List contains a list of the accelerometer channels to use.

LSB Weight specifies the g-force change represented by a one-digit change in the raw value read from the accelerometer channel.

Analog Input 1 Sample Reference

Note This reference is available when you use either the myRIO Toolkit or the roboRIO Toolkit.

myRIO Reference contains the reference to the myRIO or the roboRIO.

myRIO Model specifies the version of the myRIO or the roboRIO.

myRIO Hardware Reference contains reference information about the myRIO or the roboRIO.

Generic FPGA Reference contains the reference to the FPGA target on the myRIO or the roboRIO.

TF **Allow multiple opens** defines whether to allow opening the specified channels more than once.

AI Channels List contains a list of the analog input channels to use.

AI Scaling Constants List contains a list of scaling constants for converting the raw values returned from the myRIO or the roboRIO into voltage values.

DBL **LSB Weight (Volts)** specifies the voltage change represented by a one-digit change in the raw value read from the analog channel.

DBL **Offset (Volts)** specifies the difference between the actual value read and the expected value.

TF **Signed?** specifies whether to treat the raw value read from the analog channel as a signed value or an unsigned value.

Analog Output 1 Sample Reference

Note This reference is available when you use either the myRIO Toolkit or the roboRIO Toolkit.

myRIO Reference contains the reference to the myRIO or the roboRIO.

myRIO Model specifies the version of the myRIO or the roboRIO.

myRIO Hardware Reference contains reference information about the myRIO or the roboRIO.

Generic FPGA Reference contains the reference to the FPGA target on the myRIO or the roboRIO.

Allow multiple opens defines whether to allow opening the specified channels more than once.

AO Channel List contains a list of the analog output channels to use.

AO Scaling Constant List contains a list of scaling constants for converting the raw values returned from the myRIO or the roboRIO into voltage values.

DBL **LSB Weight (Volts)** specifies the voltage change represented by a one-digit change in the raw value read from the analog channel.

DBL **Offset (Volts)** specifies the difference between the actual value read and the expected value.

TF **Signed?** specifies whether to treat the raw value read from the analog channel as a signed value or an unsigned value.

Digital Input/Output 1 Sample Reference

Note This reference is available when you use either the myRIO Toolkit or the roboRIO Toolkit.

myRIO Reference contains the reference to the myRIO or the roboRIO.

myRIO Model

specifies the version of the myRIO or the roboRIO.

myRIO Hardware Reference

contains reference information about the myRIO or the roboRIO.

Generic FPGA Reference

contains the reference to the FPGA target on the myRIO or the roboRIO.

Allow multiple opens

defines whether to allow opening the specified channels more than once.

DIO Channels List contains a list of the digital I/O channels to use.

Encoder Reference

Note This reference is available when you use either the myRIO Toolkit or the roboRIO Toolkit.

myRIO Reference contains the reference to the myRIO or the roboRIO.

myRIO Model specifies the version of the myRIO or the roboRIO.

myRIO Hardware Reference contains reference information about the myRIO or the roboRIO.

Generic FPGA Reference contains the reference to the FPGA target on the myRIO or the roboRIO.

Allow multiple opens defines whether to allow opening the specified channels

more than once.

Encoder Channels Enum specifies the encoder channel to use.

I2C Reference

Note This reference is available when you use either the myRIO Toolkit or the roboRIO Toolkit.

myRIO Reference contains the reference to the myRIO or the roboRIO.

- **myRIO Model** specifies the version of the myRIO or the roboRIO.
- **myRIO Hardware Reference** contains reference information about the myRIO or the roboRIO.
  - **Generic FPGA Reference** contains the reference to the FPGA target on the myRIO or the roboRIO.
  - **Allow multiple opens** defines

whether to allow opening the specified channels more than once.

I2C Channels Enum specifies the I2C channel to use.

I2C Configuration contains the configuration information of the I2C channel.

I2C Transfer Rate specifies the transfer rate of the I2C channel.

PWM Reference

Note This reference is available when you use either the myRIO Toolkit or the roboRIO Toolkit.

myRIO Reference contains the reference to the myRIO or the roboRIO.

myRIO Model specifies the version of the myRIO or the roboRIO.

myRIO Hardware Reference contains reference information about the myRIO or the roboRIO.

Generic FPGA Reference contains

the reference to the FPGA target on the myRIO or the roboRIO.

TF **Allow multiple opens** defines whether to allow opening the specified channels more than once.

PWM Channels Enum specifies the PWM channel to use.

PWM Configuration contains the configuration information of the PWM channel.

DBL **Frequency** specifies the frequency, in hertz, of the generated PWM signal.

DBL **Duty Cycle** specifies the percentage of time a PWM signal remains high over one PWM cycle.

SPI Reference

Note This reference is available when you use either the myRIO Toolkit or the roboRIO Toolkit.

myRIO Reference contains the reference to the myRIO or the roboRIO.

myRIO Model specifies the version of the myRIO or the roboRIO.

myRIO Hardware Reference contains reference information about the myRIO or the roboRIO.

Generic FPGA Referenc e contains the reference to the FPGA target on the myRIO or the roboRIO.

Allow multiple opens defines whether to allow opening the specified channels

more than once.

SPI Channels Enum specifies the SPI channel to use.

SPI Configuration contains the configuration information of the SPI channel.

Frequency specifies the frequency, in hertz, of the generated SPI clock signal.

Clock Phase specifies the clock phase at which the data remains stable in the SPI transmission cycle.

0	**Leading**—The data is stable on the leading edge, and the data changes on the trailing edge.
1	**Trailing**—The data is stable on the trailing edge, and the data changes on the leading edge.

Clock Polarity specifies the base level of the clock signal and the logic level of the leading and trailing edges.

0	**Low**—The clock signal is low when idling, the leading edge is a rising edge, and the trailing edge is a falling edge.
1	**High**—The clock signal is high when idling, the leading edge is a falling edge, and the trailing edge is a right edge.

Data Direction specifies the order in which the bits in the SPI frame are transmitted.

0	**Most significant bit first**—The most significant bit is sent first an

	d the least significant bit is sent last.
1	**Least significant bit first**—The least significant bit is sent first and the most significant bit is sent last.

Frame Length specifies the number of frames that make up a single SPI transmission frame. **Frame Length** can be a value from 3 to 15, which specifies a frame length of 4 to 16.

Relay Reference

Note This reference is available when you use the roboRIO Toolkit.

myRIO Reference contains the reference to the roboRIO.

myRIO Model specifies the version of the roboRIO.

myRIO Hardware Reference contains reference information about the roboRIO.

Generic FPGA Reference contains the reference to the FPGA target on the roboRIO.

Allow multiple opens defines whether to allow opening the specified channels more than once.

Relay Channels List contains a list of the relay channels to use.

RSL Reference

Note This reference is available when you use the roboRIO Toolkit.

myRIO Reference contains the reference to the roboRIO.

myRIO Model specifies the version of the roboRIO.

myRIO Hardware Reference contains reference information about the roboRIO.

Generic FPGA Reference contains the reference to the FPGA target on the roboRIO.

Allow multiple opens defines whether to allow opening the specified channels more than once.

RSL Channels List contains a list of the RSL channels to use.

Analog Input N Samples Reference

Note This reference is available when you use the myRIO Toolkit with the high-throughput FPGA personality.

myRIO Reference contains the reference to the myRIO.

- **myRIO Model** specifies the version of the myRIO.
- **myRIO Hardware Reference** contains reference information about the myRIO.
  - **Generic FPGA Reference** contains the reference to the FPGA target on the myRIO.
  - **Allow multiple opens** defines whether to allow opening the specified channels

more than once.

AI (N Samples) Channels List contains a list of the analog input channels to use for **n** samples read operations.

AI (N Samples) Scaling Constants List contains a list of scaling constants for converting the raw values returned from the myRIO into voltage values.

LSB Weight (Volts) specifies the voltage change represented by a one-digit change in the raw value read from the analog channel.

Offset (Volts) specifies the difference between the actual value read and the expected value.

Signed? specifies whether to treat the raw value read from the analog channel as a signed value or an unsigned value.

Analog Output N Samples Reference

Note This reference is available when you use the myRIO Toolkit with the high-throughput FPGA personality.

myRIO Reference contains the reference to the myRIO.

myRIO Model specifies the version of the myRIO.

myRIO Hardware Reference contains reference information about the myRIO.

Generic FPGA Reference contains the reference to the FPGA target on the myRIO.

Allow multiple opens defines whether to allow opening the specified channels more than once.

AO (N Samples) Channels List contains a list of the analog output channels to use for **n** samples write operations.

AO (N Samples) Scaling Constants List contains a list of scaling constants for converting the raw values returned from the myRIO into voltage values.

DBL **LSB Weight (Volts)** specifies the voltage change represented by a one-digit change in the raw value read from the analog channel.

DBL **Offset (Volts)** specifies the difference between the actual value read and the expected value.

TF **Signed?** specifies whether to treat the raw value read from the analog channel as a signed value or an unsigned value.

Digital Input N Samples Reference

Note This reference is available when you use the myRIO Toolkit with the high-throughput FPGA personality.

myRIO Reference contains the reference to the myRIO.

myRIO Model specifies the version of the myRIO.

myRIO Hardware Reference contains reference information about the myRIO.

Generic FPGA Reference contains the reference to the FPGA target on the myRIO.

Allow multiple opens defines whether to allow opening the specified channels more than once.

DI (N Samples) Channels List contains a list of the digital input channels to use for **n** samples read operations.

Digital Output N Samples Reference

Note This reference is available when you use the myRIO Toolkit with the high-throughput FPGA personality.

myRIO Reference contains the reference to the myRIO.

myRIO Model specifies the version of the myRIO.

myRIO Hardware Reference contains reference information about the myRIO.

Generic FPGA Reference contains the reference to the FPGA target on the myRIO.

Allow multiple opens defines whether to allow opening the specified channels more than once.

DO (N Samples) Channels List contains a list of the digital output channels to use for **n** samples write operations.

Audio Input N Samples Reference

Note This reference is available when you use the myRIO Toolkit with the high-throughput FPGA personality.

myRIO Reference contains the reference to the myRIO.

- **myRIO Model** specifies the version of the myRIO.
- **myRIO Hardware Reference** contains reference information about the myRIO.
  - **Generic FPGA Reference** contains the reference to the FPGA target on the myRIO.
  - **Allow multiple opens** defines whether to allow opening the specified channels more than once.

AudioIn (N Samples) Channels List contains a list of the audio input channels to use for **n** samples read operations.

AudioIn (N Samples) Scaling Constants List

contains a list of scaling constants for converting the raw values returned from the myRIO into voltage values.

LSB Weight (Volts)

specifies the voltage change represented by a one-digit change in the raw value read from the audio channel.

Offset (Volts)

specifies the difference between the actual value read and the expected value.

Signed? specifies whether to treat the raw value read from the audio channel as a signed value or an unsigned value.

Audio Output N Samples Reference

Note This reference is available when you use the myRIO Toolkit with the high-throughput FPGA personality.

myRIO Reference contains the reference to the myRIO.

myRIO Model

specifies the version of the myRIO.

myRIO Hardware Reference

contains reference

information about the myRIO.

Generic FPGA Reference

contains the reference to the FPGA target on the myRIO.

Allow multiple opens

defines whether to allow opening the specified channels more than once.

AudioOut (N Samples) Channels List

contains a list of the audio output channels to use for **n** samples write operations.

AudioOut (N Samples) Scaling Constants List

contains a list of scaling constants for converting the raw values returned from the myRIO into voltage values.

LSB Weight (Volts)

specifies the voltage change represented by a one-digit change in the raw value read from the audio channel.

DBL **Offset (Volts)** specifies the difference between the actual value read and the expected value.

TF **Signed?** specifies whether to treat the raw value read from the audio channel as a signed value or an unsigned value.

Accelerometer VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit. This topic might not match its corresponding palette in LabVIEW depending on your operating system, licensed product(s), and target.

Use the Accelerometer VIs to control the onboard accelerometer on the myRIO or the roboRIO.

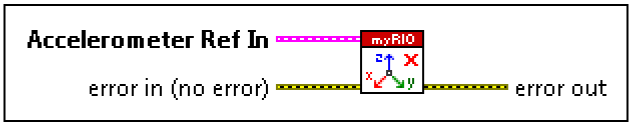
Palette Object	Description
Close	Closes the reference to accelerometer channels.
Open	Opens a reference to one or more accelerometer channels. You must open a reference before you read values from an accelerometer channel.
Read	Reads acceleration values from the onboard acc elerometer. You must open and configure accele rometer channels before reading values from th e accelerometer channels.

Close VI

Owning Palette: Accelerometer VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Closes the reference to accelerometer channels.



Accelerometer Ref In specifies the reference to the accelerometer channels. Use the Open VI to open a reference to the accelerometer channels. Do not modify the **Accelerometer Ref In** values.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

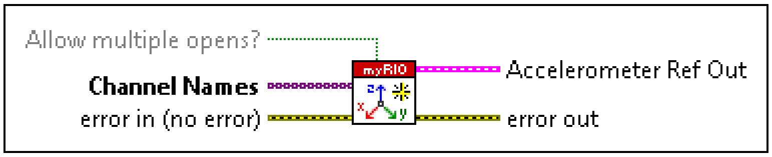
error out contains error information. This output provides standard error out functionality.

Open VI

Owning Palette: Accelerometer VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Opens a reference to one or more accelerometer channels. You must open a reference before you read values from an accelerometer channel.



Allow multiple opens? specifies whether to allow opening the specified channels more than once. The default is FALSE. You must set **Allow multiple opens?** to TRUE if you also use the specified channels in an Express VI.

Channel Names specifies the names of the accelerometer channels to open a reference.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

Accelerometer Ref Out returns reference to the accelerometer channels that you specify.

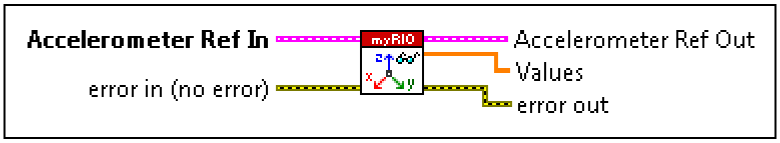
error out contains error information. This output provides standard error out functionality.

Read VI

Owning Palette: Accelerometer VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Reads acceleration values from the onboard accelerometer. You must open and configure accelerometer channels before reading values from the accelerometer channels.



Accelerometer Ref In specifies the reference to the accelerometer channels. Use the Open VI to open a reference to the accelerometer channels. Do not modify the **Accelerometer Ref In** values.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

Accelerometer Ref Out returns the reference to the accelerometer channels that you specify.

Values returns the acceleration values that this VI reads from the accelerometer channels.

error out contains error information. This output provides standard error out functionality.

Analog Input 1 Sample VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit. This topic might not match its corresponding palette in LabVIEW depending on your operating system, licensed product(s), and target.

Use the Analog Input 1 Sample VIs to control the analog input channels on the myRIO or the roboRIO.

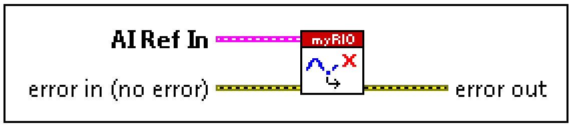
Palette Object	Description
Close	Closes the reference to one or more analog input channels.
Open	Opens a reference to one or more analog input channels. You must open a reference before you read values from an analog input channel.
Read	Reads values from one or more analog input channels.

Close VI

Owning Palette: Analog Input 1 Sample VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Closes the reference to one or more analog input channels.



AI Ref In specifies the reference to the analog input channels. Use the Open VI to open a reference to the analog input channels. Do not modify the **AI Ref In** values.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

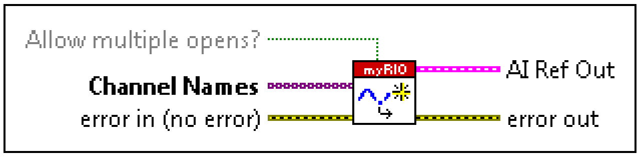
error out contains error information. This output provides standard error out functionality.

Open VI

Owning Palette: Analog Input 1 Sample VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Opens a reference to one or more analog input channels. You must open a reference before you read values from an analog input channel.



Allow multiple opens? specifies whether to allow opening the specified channels more than once. The default is FALSE. You must set **Allow multiple opens?** to TRUE if you also use the specified channels in an Express VI.

Channel Names specifies the names of the analog input channels whose reference you want to open.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

AI Ref Out returns a reference to the analog input channels that you specify.

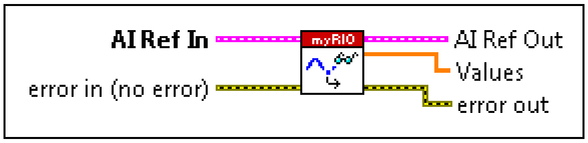
error out contains error information. This output provides standard error out functionality.

Read VI

Owning Palette: Analog Input 1 Sample VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Reads values from one or more analog input channels.



AI Ref In specifies the reference to the analog input channels from which to read values. Use the Open VI to open a reference to the analog input channels. Do not modify the **AI Ref In** values.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

AI Ref Out returns the reference to the analog input channels from which this VI reads values.

Values returns the values in volts that this VI reads from the analog input channels. The order of the values corresponds to the order in which the Open VI opens the analog input channels.

error out contains error information. This output provides standard error out functionality.

Analog Output 1 Sample VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit. This topic might not match its corresponding palette in LabVIEW depending on your operating system, licensed product(s), and target.

Use the Analog Output 1 Sample VIs to control the analog output channels on the myRIO or the roboRIO.

Palette Object	Description
Close	Closes the reference to one or more analog output channels and resets the output voltage to 0 V.
Open	Opens a reference to one or more analog output channels. You must open a reference before you write values to an analog output channel.

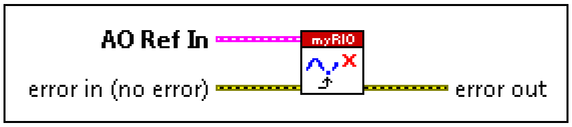
Write	Writes values to one or more analog output cha nnels.

Close VI

Owning Palette: Analog Output 1 Sample VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Closes the reference to one or more analog output channels and resets the output voltage to 0 V.



AO Ref In specifies the reference to the analog output channels. Do not modify the **AO Ref In** values.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

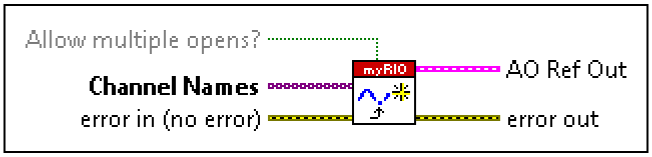
error out contains error information. This output provides standard error out functionality.

Open VI

Owning Palette: Analog Output 1 Sample VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Opens a reference to one or more analog output channels. You must open a reference before you write values to an analog output channel.



TF

Allow multiple opens? specifies whether to allow opening the specified channels more than once. The default is FALSE. You must set **Allow multiple opens?** to TRUE if you also use the specified channels in an Express VI.

[I/O] **Channel Names** specifies the names of the analog output channels to open a reference.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

AO Ref Out returns a reference to the analog output channels that you specify.

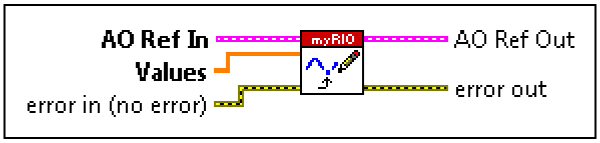
error out contains error information. This output provides standard error out functionality.

Write VI

Owning Palette: Analog Output 1 Sample VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Writes values to one or more analog output channels.



AO Ref In specifies the reference to the analog output channels to which to write values. Use the Open VI to open a reference to the analog output channels. Do not modify the **AO Ref In** values.

[DBL] **Values** specifies the values, in volts, to write to the analog output channels. The order of the values corresponds to the order in which the Open VI opens the analog output channels. You must assign a value for each channel that you specify.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

AO Ref Out returns the reference to the analog output channels to which this VI writes values.

error out contains error information. This output provides standard error out functionality.

Digital Input/Output 1 Sample VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit. This topic might not match its corresponding palette in LabVIEW depending on your operating system, licensed product(s), and target.

Use the Digital Input/Output 1 Sample VIs to control the digital I/O channels on the myRIO or the roboRIO.

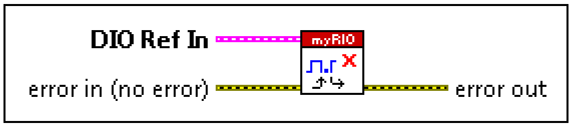
Palette Object	Description
Close	Closes the reference to one or more digital I/O channels, sets the logic levels of all output channels to low, and disables all output channels.
Open	Opens a reference to one or more digital I/O channels. You must open a reference before you read or write values.
Read	Reads the logic levels of one or more digital I/O channels.
Write	Writes logic levels to one or more digital I/O channels.

Close VI

Owning Palette: Digital Input/Output 1 Sample VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Closes the reference to one or more digital I/O channels, sets the logic levels of all output channels to low, and disables all output channels.



DIO Ref In specifies the reference to the digital I/O channels. Use the Open VI to open a reference to the digital I/O channels. Do not modify the **DIO Ref In** values.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

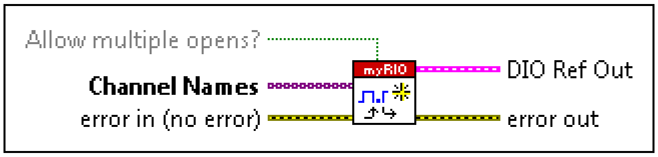
error out contains error information. This output provides standard error out functionality.

Open VI

Owning Palette: Digital Input/Output 1 Sample VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Opens a reference to one or more digital I/O channels. You must open a reference before you read or write values.



Allow multiple opens? specifies whether to allow opening the specified channels more than once. The default is FALSE. You must set **Allow multiple opens?** to TRUE if you also use the specified channels in an Express VI.

Channel Names specifies the names of the digital I/O channels whose reference you want to open.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

DIO Ref Out returns a reference to the digital I/O channels that you specify.

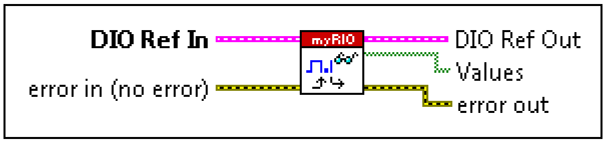
error out contains error information. This output provides standard error out functionality.

Read VI

Owning Palette: Digital Input/Output 1 Sample VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Reads the logic levels of one or more digital I/O channels.



DIO Ref In specifies the reference to the digital I/O channels from which to read logic levels. Use the Open VI to open a reference to the digital I/O channels. Do not modify the **DIO Ref In** values.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

DIO Ref Out returns the reference to the digital I/O channels from which this VI reads logic levels.

Values returns the logic levels that this VI reads from the digital I/O channels. **Values** returns TRUE if the logic level is a high voltage and returns FALSE if the logic level is a low voltage. The direction of a channel changes to input before this VI reads the logic level. The order of the elements in **Values** corresponds to the order in which the Open VI opens the digital I/O channels.

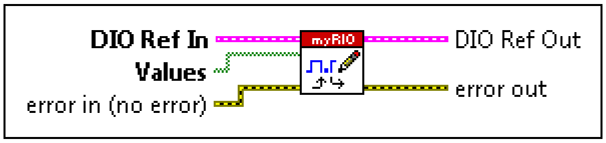
error out contains error information. This output provides standard error out functionality.

Write VI

Owning Palette: Digital Input/Output 1 Sample VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Writes logic levels to one or more digital I/O channels.



DIO Ref In specifies the reference to the digital I/O channels to which to write logic levels. Use the Open VI to open a reference to the digital I/O channels. Do not modify the **DIO Ref In** values.

Values specifies the logic levels to write to the digital I/O channels. Set **Values** to TRUE to write a high voltage and set **Values** to FALSE to write a low voltage. The direction of a channel changes to output before this VI writes the logic level. The order of the elements in **Values** corresponds to the order in which the Open VI opens the digital I/O channels. You must specify a value for each channel that you open.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

DIO Ref Out returns the reference to the digital I/O channels to which this VI writes logic levels.

error out contains error information. This output provides standard error out functionality.

Encoder VIs

Owning Palette: Low Level VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit. This topic might not match its corresponding palette in LabVIEW depending on your operating system, licensed product(s), and target.

Use the Encoder VIs to control the encoder channels on the myRIO or the roboRIO.

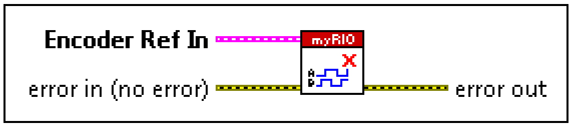
Palette Object	Description
Close	Closes the reference to an encoder channel, dis ables the counter, and resets the counter value t o zero.
Open	Opens a reference to an encoder channel. You m ust open a reference before you use an encoder channel to read and decode signals from an enc oder.
Read	Reads the value of the encoder tick counter, the last direction of the tick counter, and whether th e counter wraps around.
Reset	Resets the encoder tick counter to zero.
Start	Starts the encoder tick counter. The encoder yo u connect to the myRIO or the roboRIO starts to increment or decrement the tick counter values.
Stop	Stops the encoder tick counter. The encoder yo u connect to the myRIO or the roboRIO stops ch anging the tick counter values.

Close VI

Owning Palette: Encoder VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Closes the reference to an encoder channel, disables the counter, and resets the counter value to zero.



Encoder Ref In specifies the reference to the encoder channel. Use the Open VI to open a reference to the encoder channel. Do not modify the **Encoder Ref In** values.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

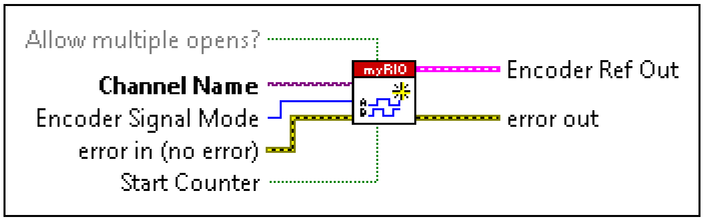
error out contains error information. This output provides standard error out functionality.

Open VI

Owning Palette: Encoder VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Opens a reference to an encoder channel. You must open a reference before you use an encoder channel to read and decode signals from an encoder.



Allow multiple opens? specifies whether to allow opening the specified channel more than once. The default is FALSE. You must set **Allow multiple opens?** to TRUE if you also use the specified channel in an Express VI.

Channel Name specifies the name of the encoder channel whose reference you want to open.

Encoder Signal Mode specifies the type of output signal from the encoder you use.

0	**Quadrature Phase Signal** (default)—Specifies that the encoder generates two phase signals that are offset by 90 degrees. The count value changes each time there is a falling or rising edge on either of the phases. Most encoders generate quadrature phase signals. You can interpret a quadrature phase signal as Gray code.
1	**Step and Direction Signal** —Specifies that the encoder generates a direction signal and a clock signal. The direction signal determines the direction of the encoder. The count value changes on every rising edge of the clock signal.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

Start Counter specifies whether the encoder tick counter starts immediately after the channel is open. The default is TRUE.

Encoder Ref Out returns a reference to the encoder channel that you specify.

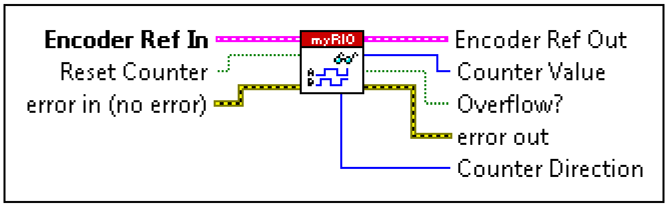
error out contains error information. This output provides standard error out functionality.

Read VI

Owning Palette: Encoder VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Reads the value of the encoder tick counter, the last direction of the tick counter, and whether the counter wraps around.



Encoder Ref In specifies the reference to the encoder channel. Use the Open VI to open a reference to the encoder channel. Do not modify the **Encoder Ref In** values.

TF

Reset Counter specifies whether to reset the tick counter to zero after this VI runs. The default is FALSE.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

Encoder Ref Out returns a reference to the encoder channel that you specify.

I32

Counter Value returns the number of ticks that this VI reads from the encoder since the last counter reset. **Counter Value** must be in the range from -2,147,483,648 to 2,147,483,647.

TF

Overflow? returns whether the counter value wraps around. **Overflow?** returns TRUE when the value of the counter goes from the maximum value to the minimum value or from the minimum value to the maximum value. After the count value is read once, **Overflow?** changes to FALSE until the counter value wraps around again.

error out contains error information. This output provides standard error out functionality.

Counter Direction returns the direction of the counter between the last two ticks that the encoder receives.

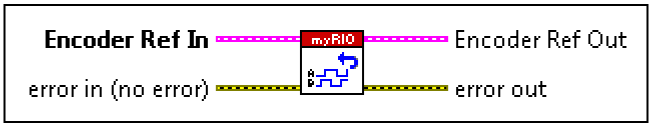
0	**Counting Up**—Returns that the counter is incrementing.
1	**Counting Down**—Returns that the counter is decrementing.

Reset VI

Owning Palette: Encoder VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Resets the encoder tick counter to zero.



Encoder Ref In specifies the reference to the encoder channel for which to reset the counter. Use the Open VI to open a reference to the encoder channel. Do not modify the **Encoder Ref In** values.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

Encoder Ref Out returns the reference to the encoder channel for which you reset the counter.

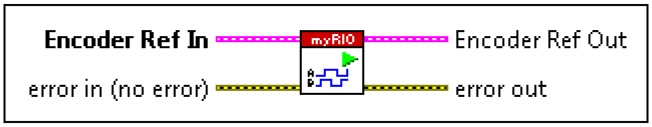
error out contains error information. This output provides standard error out functionality.

Start VI

Owning Palette: Encoder VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Starts the encoder tick counter. The encoder you connect to the myRIO or the roboRIO starts to increment or decrement the tick counter values.



Encoder Ref In specifies the reference to the encoder channel for which to start the counter. Use the Open VI to open a reference to the encoder channel. Do not modify the **Encoder Ref In** values.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

Encoder Ref Out returns a reference to the encoder channel for which you start the counter.

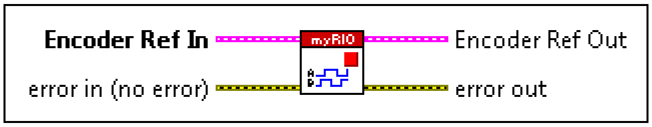
error out contains error information. This output provides standard error out functionality.

Stop VI

Owning Palette: Encoder VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Stops the encoder tick counter. The encoder you connect to the myRIO or the roboRIO stops changing the tick counter values.



Encoder Ref In specifies the reference to the encoder channel for which to stop the counter. Use the Open VI to open a reference to the encoder channel. Do not modify the **Encoder Ref In** values.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

Encoder Ref Out returns a reference to the encoder channel for which you stop the counter.

error out contains error information. This output provides standard error out functionality.

I2C VIs

Owning Palette: Low Level VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit. This topic might not match its corresponding palette in LabVIEW depending on your operating system, licensed product(s), and target.

Use the I2C VIs to control the Inter-Integrated Circuit (I2C) channels on the myRIO or the roboRIO.

Palette Object	Description
Close	Closes the reference to an Inter-Integrated Circuit (I2C) channel. This VI also disables the I2C channel and resets the configuration of the channel.
Configure	Configures the transfer rate of an Inter-Integrated Circuit (I2C) channel based on the input I2C reference.
Open	Opens a reference to an Inter-Integrated Circuit (I2C) channel. You must open a reference before

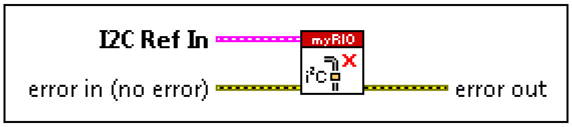
	you use an I2C channel to write data to and read data from an I2C slave device.
Read	Reads a specified number of bytes of data from an Inter-Integrated Circuit (I2C) channel. This VI returns the result when finishing reading all the bytes or when timing out.
Write	Writes data to an Inter-Integrated Circuit (I2C) sl ave device. This VI returns the result when finish ing writing all the bytes or when timing out.
Write Read	Writes data to an Inter-Integrated Circuit (I2C) sl ave device and then reads a specified number of bytes of data from the I2C slave device. This VI r eturns the result when finishing writing or readi ng all the bytes, or when timing out

Close VI

Owning Palette: I2C VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Closes the reference to an Inter-Integrated Circuit (I2C) channel. This VI also disables the I2C channel and resets the configuration of the channel.



I2C Ref In specifies the reference to the I2C channel. Use the Open VI to open a reference to the I2C channel. Do not modify the **I2C Ref In** values.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

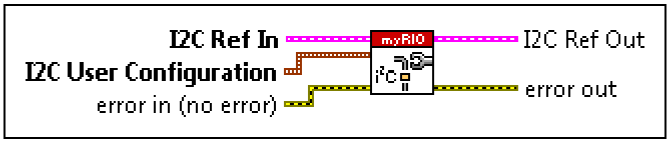
error out contains error information. This output provides standard error out functionality.

Configure VI

Owning Palette: I2C VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Configures the transfer rate of an Inter-Integrated Circuit (I2C) channel based on the input I2C reference.



I2C Ref In specifies the reference to the I2C channel. Use the Open VI to open a reference to the I2C channel. Do not modify the **I2C Ref In** values.

I2C User Configuration specifies the user-configurable properties of the I2C channel.

I2C Transfer Rate specifies the transfer rate of the I2C channel.

0	**Standard Mode (100 kbps)** (default)
1	**Fast Mode (400 kbps)**

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

I2C Ref Out returns the reference to the I2C channel with the transfer rate configuration.

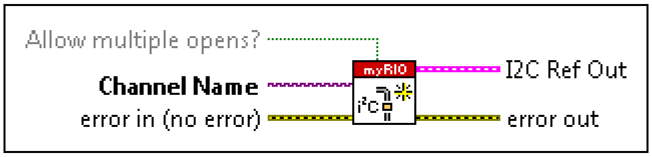
error out contains error information. This output provides standard error out functionality.

Open VI

Owning Palette: I2C VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Opens a reference to an Inter-Integrated Circuit (I2C) channel. You must open a reference before you use an I2C channel to write data to and read data from an I2C slave device.



Allow multiple opens? specifies whether to allow opening the specified channel more than once. The default is FALSE. You must set **Allow multiple opens?** to TRUE if you also use the specified channel in an Express VI.

Channel Name specifies the name of the I2C channel to open a reference.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

I2C Ref Out returns a reference to the I2C channel that you specify. You must use the Configure VI to set the transfer rate of the I2C channel.

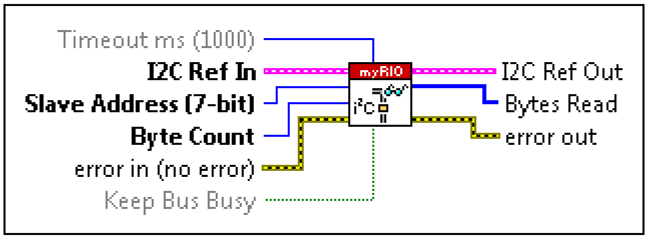
error out contains error information. This output provides standard error out functionality.

Read VI

Owning Palette: I2C VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Reads a specified number of bytes of data from an Inter-Integrated Circuit (I2C) channel. This VI returns the result when finishing reading all the bytes or when timing out.



I32 **Timeout ms** specifies the number of milliseconds this VI waits for receiving a single byte before timing out.

I2C Ref In specifies the reference to the I2C channel. Use the Open VI to open a reference to the I2C channel. Do not modify the **I2C Ref In** values.

U8 **Slave Address (7-bit)** specifies the address of the I2C slave device from which this VI reads data. You must specify the address in 7-bit. Some I2C devices might have a 8-bit address in which the first 7 bits represent the address and the last bit represents the mode of operation. For these kind of I2C devices, you must specify **Slave Address (7-bit)** using the seven most significant bits.

U32 **Byte Count** specifies the number of bytes of data this VI reads from the I2C slave device.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

TF **Keep Bus Busy** specifies whether to keep the I2C channel open so that you can perform additional operations. For example, set **Keep Bus Busy** to TRUE if a slave device requires a read operation followed by a write operation to perform a command. The default is FALSE.

I2C Ref Out returns the reference to the I2C channel that you specify.

Bytes Read returns the data that this VI reads from the I2C slave device.

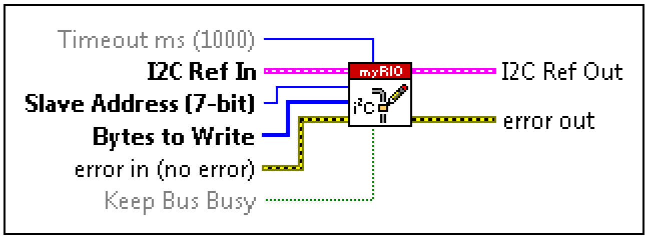
error out contains error information. This output provides standard error out functionality.

Write VI

Owning Palette: I2C VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Writes data to an Inter-Integrated Circuit (I2C) slave device. This VI returns the result when finishing writing all the bytes or when timing out.



Timeout ms specifies the number of milliseconds this VI waits for writing a single byte before timing out.

I2C Ref In specifies the reference to the I2C channel. Use the Open VI to open a reference to the I2C channel. Do not modify the **I2C Ref In** values.

Slave Address (7-bit) specifies the address of the slave device to which this VI writes data. You must specify the address in 7-bit. Some I2C devices might have a 8-bit address in which the first 7 bits represent the address and the last bit represents the mode of operation. For these kind of I2C devices, you must specify

Slave Address (7-bit) using the seven most significant bits.

Bytes to Write specifies the data to write to the I2C slave device.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

Keep Bus Busy specifies whether to keep the I2C channel open so that you can perform additional operations. For example, if a slave device requires a write operation followed by a read operation to perform a command, you must set **Keep Bus Busy** to TRUE. The default is FALSE.

I2C Ref Out returns the reference to the I2C channel that you specify.

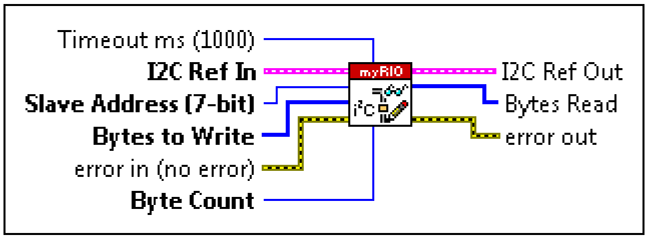
error out contains error information. This output provides standard error out functionality.

Write Read VI

Owning Palette: I2C VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Writes data to an Inter-Integrated Circuit (I2C) slave device and then reads a specified number of bytes of data from the I2C slave device. This VI returns the result when finishing writing or reading all the bytes, or when timing out



I32 **Timeout ms** specifies the number of milliseconds this VI waits for writing or reading a single byte before timing out.

I2C Ref In specifies the reference to the I2C channel. Use the Open VI to open a reference to the I2C channel. Do not modify the **I2C Ref In** values.

U8 **Slave Address (7-bit)** specifies the address of the I2C slave device to which this VI writes data or from which this VI reads data. You must specify the address in 7-bit. Some I2C devices might have a 8-bit address in which the first 7 bits represent the address and the last bit represents the mode of operation. For these kind of I2C devices, you must specify **Slave Address (7-bit)** using the seven most significant bits.

[U8] **Bytes to Write** specifies the data to write to the I2C slave device.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

U32 **Byte Count** specifies the number of bytes of data this VI reads from the I2C slave device.

I2C Ref Out returns the reference to the I2C channel with the transfer rate configuration.

[U8] **Bytes Read** returns the data that this VI reads from the I2C slave device.

error out contains error information. This output provides standard error out functionality.

Input Device Control VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit. This topic might not match its corresponding palette in LabVIEW depending on your operating system, licensed product(s), and target.

Use the Input Device Control VIs to initialize, query, close, and get the state information about the joystick connected to the myRIO or the roboRIO.

Palette Object	Description
Acquire Input Data	Returns data about the joystick connected to th e computer.
Close Input Device	Closes the joystick you specify in **device ID**.
Initialize Joystick	Opens a reference to and initializes a joystick de vice at the index you specify.
Query Input Devices	Obtains information about the joystick connect ed to the computer.

Refer to the `Joystick Monitoring.lvproj` in the `labview\examples\myRIO\Joystick Monitoring` directory for an example of using the Input Device Control VIs with the myRIO.

Refer to the `Joystick Monitoring.lvproj` in the `labview\examples\roboRIO\Joystick Monitoring` directory for an example of using the Input Device Control VIs with the roboRIO.

Interrupt VIs

Owning Palette: Low Level VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit. This topic might not match its corresponding palette in LabVIEW depending on your operating system, licensed product(s), and target.

Use the Interrupt VIs to manage interrupts on the myRIO or the roboRIO. You can register and unregister analog and digital input interrupts. You can also create and destroy timer interrupts.

Palette Object	Description

Callback VI Reference	Maintains a static reference to a callback VI. After you place the Callback VI Reference function on a block diagram, double-click the Callback VI Reference function to display a file dialog box where you can select a callback VI.
Create Timer Interrupt	Creates timer interrupts.
Destroy Timer Interrupt	Destroys timer interrupts. After you destroy an interrupt, LabVIEW stops creating the interrupt and releases the resources associated with the interrupt.
Register Analog Input Interrupt	Registers analog input interrupts.
Register Digital Input Interrupt	Registers digital input interrupts.
Unregister Interrupt	Unregisters analog and digital input interrupts. After you unregister an interrupt, LabVIEW stops registering the interrupt.

Callback VI Reference

Owning Palette: Interrupt VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Maintains a static reference to a callback VI. After you place the Callback VI Reference function on a block diagram, double-click the Callback VI Reference function to display a file dialog box where you can select a callback VI.

You can also right-click the Callback VI Reference function and select **Browse for Path** from the shortcut menu to display the file dialog box. When you replace the Callback VI Reference function with a callback VI in memory, the icon changes to match the VI you selected.
Details

vi reference returns the reference number associated with the callback VI. This output is a strictly typed VI reference, meaning that it identifies the connector pane of the callback VI that you use.

After you replace the Callback VI Reference function with a callback VI, wire the **vi reference** output to the **Callback VI Reference** input of the Create Timer Interrupt VI, Register Analog Input Interrupt VI, or Register Digital Input Interrupt VI.

Callback VI Reference Details

Related Information

Using Callback VIs (myRIO Toolkit)

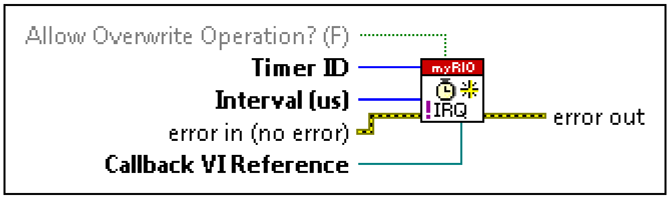
Using Callback VIs (roboRIO Toolkit)

Create Timer Interrupt VI

Owning Palette: Interrupt VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Creates timer interrupts.



TF **Allow Overwrite Operation?** specifies whether to allow creating the specified interrupt more than once. The default is FASLE.

U8 **Timer ID** specifies the identifier of the timer interrupt to create. The default is 0. Valid values are within the range [0, 7]. You cannot create a timer interrupt with the same **Timer ID** as an existing timer interrupt. However, after you destroy the existing interrupt, you can use the **Timer ID** to create another interrupt.

U32 **Interval (µs)** specifies the span of time in microseconds between two adjacent interrupts.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

Callback VI Reference specifies the reference to a callback VI. You must wire the **vi reference** output of the Callback VI Reference to this input.

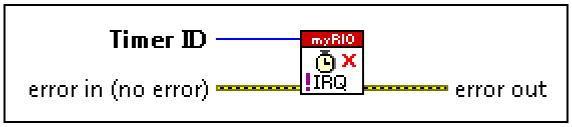
error out contains error information. This output provides standard error out functionality.

Destroy Timer Interrupt VI

Owning Palette: Interrupt VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Destroys timer interrupts. After you destroy an interrupt, LabVIEW stops creating the interrupt and releases the resources associated with the interrupt.



U8 **Timer ID** specifies the identifier of the timer interrupt to destroy. The default is 0. Valid values are within the range [0, 7].

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

error out contains error information. This output provides standard error out functionality.

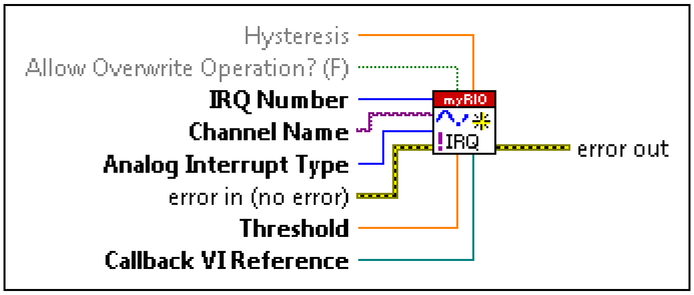
Register Analog Input Interrupt VI

Owning Palette: Interrupt VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Registers analog input interrupts.

Details



DBL

Hysteresis specifies in volts a window above or below **Threshold**. This VI uses hysteresis to prevent from false interrupt registration. The default is 0.02. Valid values are within the range [0, 1]. You do not need to change the value for this input unless you notice a false interrupt registration.

TF

Allow Overwrite Operation? specifies whether to allow registering the specified interrupt more than once. The default is FASLE.

U8

IRQ Number specifies the identifier of the interrupt to register. The default is 0. Valid values are within the range [0, 7]. You cannot register an I/O interrupt with the same **IRQ Number** as a registered I/O interrupt. However, after you unregister the existing interrupt, you can use the **IRQ Number** to register another interrupt.

I/O

Channel Name specifies the name of the analog input channel to open a reference. You must open a reference before this VI registers interrupts in an analog input channel.

U16

Analog Interrupt Type specifies when to register the interrupt based on the analog input signal.

0	**Analog falling edge** (default)—Specifies to register an interrupt on the falling edge of the analog input signal.
1	**Analog rising edge**—Specifies to register an interrupt on the rising edge of the analog input signal.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

Threshold specifies the value in volts that the analog input signal must cross for this VI to register an interrupt.

Callback VI Reference specifies the reference to a callback VI. You must wire the **vi reference** output of the Callback VI Reference to this input.

error out contains error information. This output provides standard error out functionality.

Register Analog Input Interrupt Details

Related Information

Understanding Hysteresis (myRIO Toolkit)

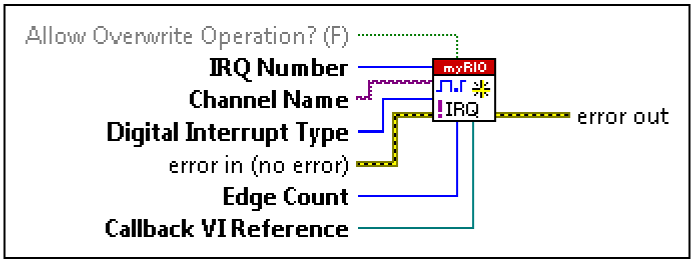
Understanding Hysteresis (roboRIO Toolkit)

Register Digital Input Interrupt VI

Owning Palette: Interrupt VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Registers digital input interrupts.



TF

Allow Overwrite Operation? specifies whether to allow registering the specified interrupt more than once. The default is FASLE.

U8

IRQ Number specifies the identifier of the interrupt to register. The default is 0. Valid values are within the range [0, 7]. You cannot register an I/O interrupt with the same **IRQ Number** as a registered I/O interrupt. However, after you unregister the existing interrupt, you can use the **IRQ Number** to register another interrupt.

I/O

Channel Name specifies the name of the digital input channel to open a reference. You must open a reference before this VI registers interrupts in a digital input channel.

U16

Digital Interrupt Type specifies when to register the interrupt based on the digital input signal.

0	**Digital falling edge** (default)—Specifies to r egister an interrupt on the falling edge of the digital input signal.
1	**Digital rising edge**—Specifies to register an interrupt on the rising

	edge of the digital input signal.
2	**Digital edge**—Specifies to register an interrupt on both the falling edge and rising edge of the digital input signal.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

Edge Count specifies the edge number of the digital input signal that must occur for this VI to register an interrupt.

Callback VI Reference specifies the reference to a callback VI. You must wire the **vi reference** output of the Callback VI Reference to this input.

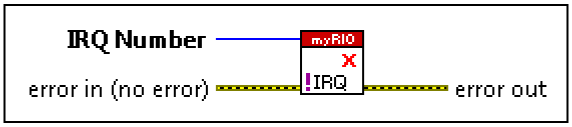
error out contains error information. This output provides standard error out functionality.

Unregister Interrupt VI

Owning Palette: Interrupt VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Unregisters analog and digital input interrupts. After you unregister an interrupt, LabVIEW stops registering the interrupt.



IRQ Number specifies the identifier of the interrupt to unregister. The default is 0. Valid values are within the range [0, 7].

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

error out contains error information. This output provides standard error out functionality.

PWM VIs

Owning Palette: Low Level VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit. This topic might not match its corresponding palette in LabVIEW depending on your operating system, licensed product(s), and target.

Use the PWM VIs to control the pulse width modulation (PWM) channels on the myRIO or the roboRIO.

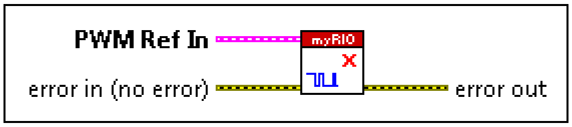
Palette Object	Description
Close	Closes the reference to a pulse width modulation (PWM) channel. This VI also disables the PWM signal and resets the frequency and duty cycle to zero.
Open	Opens a reference to a pulse width modulation (PWM) channel. You must open a reference before you use a PWM channel to generate PWM signals.
Set Duty Cycle	Sets the duty cycle value of a pulse width modulation (PWM) signal.
Set Duty Cycle and Frequency	Sets the duty cycle value and frequency value of a pulse width modulation (PWM) signal.
Set Frequency	Sets the frequency value of a pulse width modulation (PWM) signal.

Close VI

Owning Palette: PWM VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Closes the reference to a pulse width modulation (PWM) channel. This VI also disables the PWM signal and resets the frequency and duty cycle to zero.



PWM Ref In specifies the reference to the PWM channel. Use the Open VI to open a reference to the PWM channel. Do not modify the **PWM Ref In** values.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

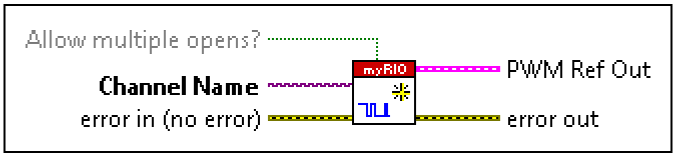
error out contains error information. This output provides standard error out functionality.

Open VI

Owning Palette: PWM VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Opens a reference to a pulse width modulation (PWM) channel. You must open a reference before you use a PWM channel to generate PWM signals.



Allow multiple opens? specifies whether to allow opening the specified channels more than once. The default is FALSE. You must set **Allow multiple opens?** to TRUE if you also use the specified channels in an Express VI.

Channel Name specifies the name of the PWM channel to open a reference.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

PWM Ref Out returns a reference to the PWM channel that you specify.

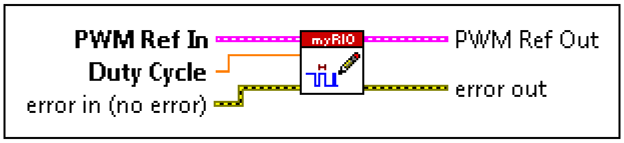
error out contains error information. This output provides standard error out functionality.

Set Duty Cycle VI

Owning Palette: PWM VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Sets the duty cycle value of a pulse width modulation (PWM) signal.



PWM Ref In specifies the reference to the PWM channel. Use the Open VI to open a reference to the PWM channel. Do not modify the **PWM Ref In** values.

Duty Cycle specifies the percentage of time the PWM signal remains high over one PWM cycle. Valid values must be within the range [0, 1].

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

PWM Ref Out returns the reference to the PWM channel that you specify.

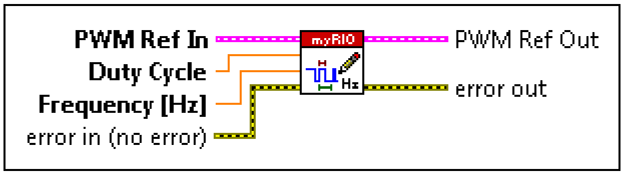
error out contains error information. This output provides standard error out functionality.

Set Duty Cycle and Frequency VI

Owning Palette: PWM VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Sets the duty cycle value and frequency value of a pulse width modulation (PWM) signal.



PWM Ref In specifies the reference to the PWM channel. Use the Open VI to open a reference to the PWM channel. Do not modify the **PWM Ref In** values.

DBL **Duty Cycle** specifies the percentage of time the PWM signal remains high over one PWM cycle. Valid values must be within the range [0, 1].

DBL **Frequency** specify the frequency of the PWM signal in hertz. **Frequency** must be within the range of 40 Hz to 40 kHz.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

PWM Ref Out returns the reference to the PWM channel that you specify.

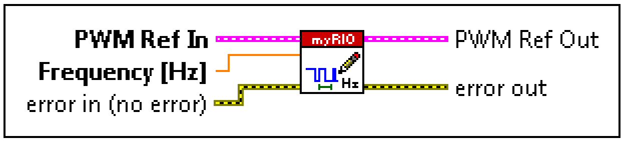
error out contains error information. This output provides standard error out functionality.

Set Frequency VI

Owning Palette: PWM VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Sets the frequency value of a pulse width modulation (PWM) signal.



PWM Ref In specifies the reference to the PWM channel. Use the Open VI to open a reference to the PWM channel. Do not modify the **PWM Ref In** values.

Frequency specify the frequency of the PWM signal in hertz. **Frequency** must be within the range of 40 Hz to 40 kHz.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

PWM Ref Out returns the reference to the PWM channel that you specify.

error out contains error information. This output provides standard error out functionality.

SPI VIs

Owning Palette: Low Level VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit. This topic might not match its corresponding palette in LabVIEW depending on your operating system, licensed product(s), and target.

Use the SPI VIs to control the serial peripheral interface (SPI) channels on the myRIO or the roboRIO.

Palette Object	Description
Close	Closes the reference to a serial peripheral interface (SPI) channel. This VI also disables the SPI channel and resets the configuration of the channel.

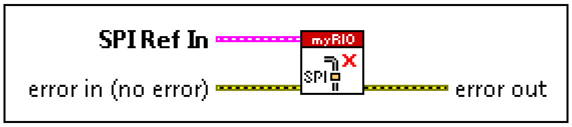
Configure	Configures a serial peripheral interface (SPI) channel based on the input SPI reference and the user configurations that you specify.
Open	Opens a reference to a serial peripheral interface (SPI) channel. You must open a reference before you use an SPI channel to write data to or read data from an SPI slave device.
Read	Reads a specified number of frames from a serial peripheral interface (SPI) channel. This VI returns results when all the frames are read. You use the Configure VI to specify the length of a frame.
Write	Writes data frames to a serial peripheral interface (SPI) channel. This VI returns results when finishing writing all the data frames.
Write Read	Writes and reads data frames through a serial peripheral interface (SPI) channel at the same time. The number of data frames to write equals the number of the data frames to read.

Close VI

Owning Palette: SPI VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Closes the reference to a serial peripheral interface (SPI) channel. This VI also disables the SPI channel and resets the configuration of the channel.



SPI Ref In specifies the reference to the SPI channel. Use the Open VI to open a reference to the SPI channel. Do not modify the **SPI Ref In** values.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

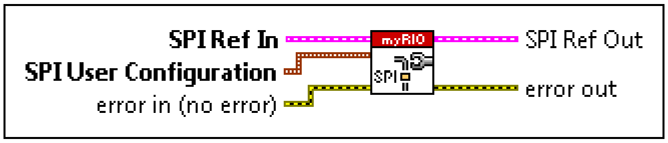
error out contains error information. This output provides standard error out functionality.

Configure VI

Owning Palette: SPI VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Configures a serial peripheral interface (SPI) channel based on the input SPI reference and the user configurations that you specify.



SPI Ref In specifies the reference to the SPI channel. Use the Open VI to open a reference to the SPI channel. Do not modify the **SPI Ref In** values.

SPI User Configuration specifies the user-configurable properties of the SPI channel.

Frequency specifies the frequency, in hertz, of the generated clock signal. **Frequency** must be within the range of 40 Hz to 4 MHz.

Clock Phase specifies the clock phase at which the data remains stable in the SPI transmission cycle.

0	**Leading** (default)— The data i

	s stable on the leading edge, and the data changes on the trailing edge.
1	**Trailing**—The data is stable on the trailing edge, and the data changes on the leading edge.

Clock Polarity

specifies the base level of the clock signal and the logic level of the leading and trailing edges.

0	**Low** (default)—The clock signal is low when idling, the leading edge is a rising edge, and the trailing edge is a falling edge.
1	**High**—The clock signal is high when idlin

	g, the leading edge is a falling edge, and the trailing edge is a right edge.

Data Direction

specifies the order in which the bits in the SPI frame are transmitted.

0	**Most Significant Bit First** (default)—Specifies to send the most significant bit first and the least significant bit last.
1	**Least Significant Bit First**—Specifies to send the least significant bit first and the most significant bit last.

Frame Length

specifies the number of frames that make up a single SPI transmission.

Frame Length can be a value from 3 to 15, which specifies a frame length of 4 to 16. The default is 8.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

SPI Ref Out returns the reference to the SPI channel, with the configurations that you specify.

error out contains error information. This output provides standard error out functionality.

Clock Phase specifies the clock phase at which the data remains stable in the SPI transmission cycle.

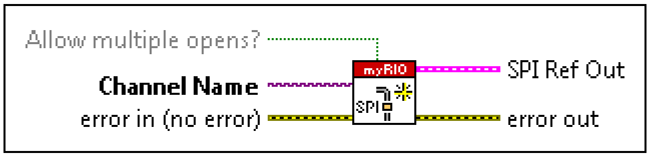
0	**Leading** (default)—The data is stable on the leading edge, and the data changes on the trailing edge.
1	**Trailing**—The data is stable on the trailing edge, and the data changes on the leading edge.

Open VI

Owning Palette: SPI VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Opens a reference to a serial peripheral interface (SPI) channel. You must open a reference before you use an SPI channel to write data to or read data from an SPI slave device.



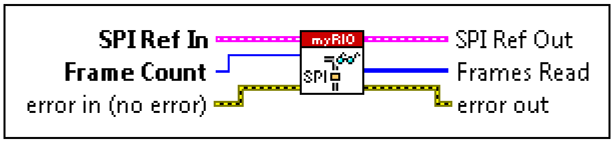
TF	**Allow multiple opens?** specifies whether to allow opening the specified channel more than once. The default is FALSE. You must set **Allow multiple opens?** to TRUE if you also use the specified channel in an Express VI.
I/O	**Channel Name** specifies the name of the SPI channel whose reference you want to open.
	error in describes error conditions that occur before this node runs. This input provides standard error in functionality.
	SPI Ref Out returns the reference to the SPI channel that you specify.
	error out contains error information. This output provides standard error out functionality.

Read VI

Owning Palette: SPI VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Reads a specified number of frames from a serial peripheral interface (SPI) channel. This VI returns results when all the frames are read. You use the Configure VI to specify the length of a frame.



SPI Ref In specifies the reference to the SPI channel. Use the Open VI to open a reference to the SPI channel. Do not modify the **SPI Ref In** values.

Frame Count specifies the number of frames to read from the SPI channel.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

SPI Ref Out returns the reference to the SPI channel that you specify.

Frames Read returns the data frames that this VI reads from the SPI channel.

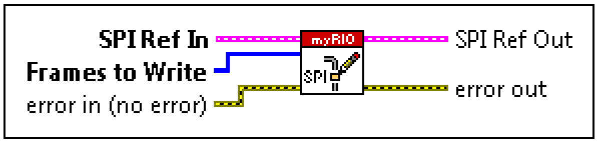
error out contains error information. This output provides standard error out functionality.

Write VI

Owning Palette: SPI VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Writes data frames to a serial peripheral interface (SPI) channel. This VI returns results when finishing writing all the data frames.



SPI Ref In specifies the reference to the SPI channel. Use the Open VI to open a reference to the SPI channel. Do not modify the **SPI Ref In** values.

Frames to Write specifies the data to write to the SPI channel.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

SPI Ref Out returns the reference to the SPI channel that you specify.

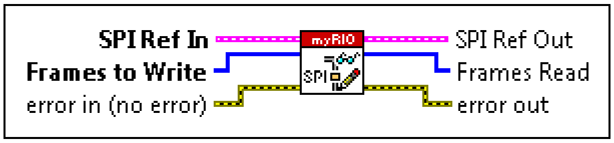
error out contains error information. This output provides standard error out functionality.

Write Read VI

Owning Palette: SPI VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Writes and reads data frames through a serial peripheral interface (SPI) channel at the same time. The number of data frames to write equals the number of the data frames to read.



SPI Ref In specifies the reference to the SPI channel. Use the Open VI to open a reference to the SPI channel. Do not modify the **SPI Ref In** values.

Frames to Write specifies the data to write to the SPI channel.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

SPI Ref Out returns the reference to the SPI channel that you specify.

Frames Read returns the data frames that this VI reads from the SPI channel.

error out contains error information. This output provides standard error out functionality.

High Throughput FPGA Personality VIs

Owning Palette: myRIO VIs

Requires: myRIO Toolkit. This topic might not match its corresponding palette in LabVIEW depending on your operating system, licensed product(s), and target.

Use the High Throughput FPGA Personality VIs to create applications on the myRIO with the high-throughput FPGA personality. The myRIO high-throughput FPGA personality supports high-speed analog or digital data access. You can use the high-throughput personality for audio signals and projects in need of waveform data.

Note The High Throughput FPGA Personality VIs appear on the **myRIO** palette only after you install the NI High Throughput Add-On for myRIO. Visit `ni.com/info` and enter the Info Code ex6g5a to learn about the High Throughput Add-On for myRIO.

Palette Object	Description
Accelerometer	Reads acceleration values along the X, Y, and Z axes of the accelerometer on the myRIO or the roboRIO.
Analog Input	Reads values from one or more analog input channels on the myRIO or the roboRIO.
Analog Output	Writes values to one or more analog output channels on the myRIO or the roboRIO.
Audio Input	Reads values from one or more audio input channels on the myRIO. Use this Express VI to read multiple samples for each channel at one time.
Audio Output	Writes values to one or more audio output channels on the myRIO. Use this Express VI to write multiple samples for each channel at one time.
Button	Reads the value from the user button on the myRIO or the roboRIO.

Digital Input	Reads values from one or more digital input channels on the myRIO or the roboRIO.
Digital Output	Writes values to one or more digital output channels on the myRIO or the roboRIO.
LED	Sets the states of the LEDs on the myRIO or the roboRIO.
UART	Writes data to or reads data from a Universal Asynchronous Receiver/Transmitter (UART) device through the UART channels on the myRIO or the roboRIO. With the roboRIO, you also can use this VI to write data to or read data from an RS-232 device through the RS-232 channel.

Subpalette	Description
Device Management VIs	Use the Device Management VIs to set custom FPGA bitfiles and to reset I/O channels on the myRIO or the roboRIO.
Low Level VIs	Use the Low Level VIs to create applications for different I/O types on the myRIO. The VIs on this palette are compatible with the high-throughput FPGA personality

Accelerometer Express VI

Requires: myRIO Toolkit **or** roboRIO Toolkit

Reads acceleration values along the X, Y, and Z axes of the accelerometer on the myRIO or the roboRIO.

Dialog Box Options

Block Diagram Inputs

Block Diagram Outputs

Dialog Box Options

Parameter	Description
Node name	Specifies the name of this Express VI. You can also double-click the name of this Express VI on the expandable node to edit the name.

X-Axis	Specifies to read the acceleration value along the X axis.
Y-Axis	Specifies to read the acceleration value along the Y axis.
Z-Axis	Specifies to read the acceleration value along the Z axis.
Custom channel name	Specifies a custom name for the axis that you select.
View Code	Displays the underlying code of this Express VI.

Block Diagram Inputs

Parameter	Description
error in (no error)	Describes error conditions that occur before this node runs.

Block Diagram Outputs

Parameter	Description
X-Axis	Returns the acceleration value along the X axis.
Y-Axis	Returns the acceleration value along the Y axis.
Z-Axis	Returns the acceleration value along the Z axis.
error out	Contains error information. This output provides standard error out functionality.

Analog Input Express VI

Requires: myRIO Toolkit **or** roboRIO Toolkit

Reads values from one or more analog input channels on the myRIO or the roboRIO.

This Express VI reads one sample each time with the default FPGA personality on the myRIO. This Express VI reads one sample or multiple samples each time with the high-throughput FPGA personality on the myRIO. Visit `ni.com/info` and enter the Info Code ex6g5a to learn about the myRIO high-throughput FPGA personality.

This Express VI reads one sample each time with the default FPGA personality on the roboRIO. The roboRIO uses a 5 V voltage rail on the ANALOG IN port for powering sensors.

Details

Dialog Box Options

Parameter	Description
I/O mode	(myRIO Toolkit) Specifies to read one sample or multiple samples. The default is Analog input (1 sample). This option is available only when you use the myRIO high-throughput FPGA personality.
Node name	Specifies the name of this Express VI. You can also double-click the name of this Express VI on the expandable node to edit the name.
Channel	Specifies the analog input channel from which to read the values.
Custom channel name	Specifies a custom name for the analog input channel that you select.
Delete Channel	Deletes the analog input channel that you select. (myRIO Toolkit) If you use the myRIO high-throughput FPGA personality, this option is available only when you specify Analog input (1 sample) for **I/O mode**.
Add Channel	Adds a new analog input channel to the channel list. You can add up to 12 analog input channels for the myRIO. You can add up to eight analog input channels for the roboRIO. (myRIO Toolkit) If you use the myRIO high-throughput FPGA personality, this option is available only when you specify Analog input (1 sample) for **I/O mode**.
Sample rate	(myRIO Toolkit) Specifies the sampling frequency of the input signal. Valid values are between 1 kHz and 50 kHz. If you specify a frequency that is invalid, this Express VI coerces the specified value to the nearest valid value when you click the **Validate** button. This option is available only w

	hen you use the myRIO high-throughput FPGA personality and specify Analog input (n samples) for **I/O mode**. ▪ **Frequency value**—Specifies the value of the sampling frequency. The default is 1. ▪ **Frequency unit**—Shows the unit of the sampling frequency. The value is kHz. ▪ **Validate**—Validates whether this Express VI can generate the sampling frequency that you specify. If the specified sampling frequency is not valid, this Express VI coerces the specified value to the nearest valid value.
Samples	(myRIO Toolkit) Specifies the number of samples to read. The default is 1,000. Valid values must be greater than 0 and less than or equal to 10,000. This option is available only when you use the myRIO high-throughput FPGA personality and specify Analog input (n samples) for **I/O mode**.
Latency	(myRIO Toolkit) Displays the latency between two adjacent signal acquisition iterations. Refer to the Details section of this topic for more information about latency. This option is available only when you use the myRIO high-throughput FPGA personality and specify Analog input (n samples) for **I/O mode**.
View Code	Displays the underlying code of this Express VI.
Connection Diagram	Shows the I/O connector pinouts on the myRIO or the roboRIO. The highlighted pinouts represent the channels that you configure.

Block Diagram Inputs

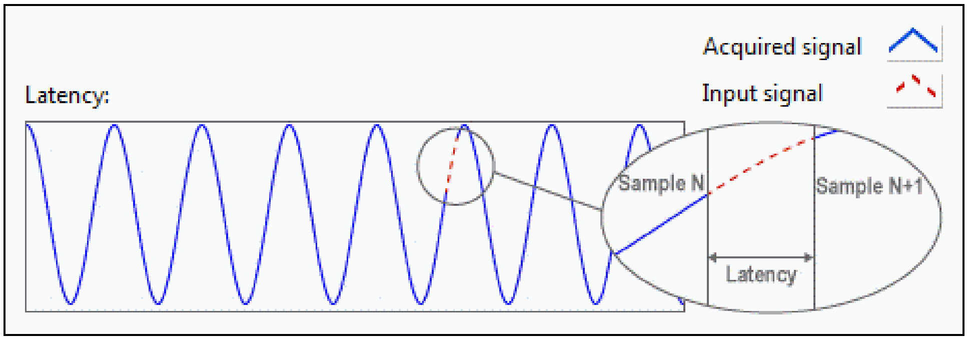
Parameter	Description
error in (no error)	Describes error conditions that occur before this node runs.

Block Diagram Outputs

Parameter	Description
channel name	Returns the value that this Express VI reads from the analog input channel that you select, where **channel name** is the name of the analog input channel.
error out	Contains error information. This output provide s standard error out functionality.

Analog Input Details

(myRIO Toolkit) The following figure demonstrates the latency when you use the Analog Input Express VI with the myRIO high-throughput FPGA personality to perform **n** samples read operations.



In the previous figure, the x-axis represents time and the y-axis represents amplitude. The waveform in blue represents the signal that the myRIO acquires. The red dotted line represents latency. When latency occurs, the myRIO does not acquire any signal. In other words, the time interval between two adjacent signal acquisition iterations is latency.

Related Information

1 Sample versus N Samples Modes (myRIO Toolkit)

Generating FPGA Clocks (myRIO Toolkit)

Generating FPGA Clocks (roboRIO Toolkit)

I/O Connectors (myRIO Toolkit)

I/O Connectors (roboRIO Toolkit)

Latency in N Samples Read and Write Operations (myRIO Toolkit)

Power Supply for Peripheral Devices (roboRIO Toolkit)

Analog Output Express VI

Requires: myRIO Toolkit **or** roboRIO Toolkit

Writes values to one or more analog output channels on the myRIO or the roboRIO.

This Express VI writes one sample each time with the default FPGA personality on the myRIO. This Express VI writes one sample or multiple samples each time with the high-throughput FPGA personality on the myRIO. Visit `ni.com/info` and enter the Info Code ex6g5a to learn about the myRIO high-throughput FPGA personality.

This Express VI writes one sample each time with the default FPGA personality on the roboRIO.

Details

Dialog Box Options

Parameter	Description
I/O mode	(myRIO Toolkit) Specifies to write one sample or multiple samples. The default is Analog output (1 sample). This option is available only when yo u use the myRIO high-throughput FPGA persona lity.
Node name	Specifies the name of this Express VI. You can al so double-click the name of this Express VI on th e expandable node to edit the name.
Channel	Specifies the analog output channel to which to write a value.
Custom channel name	Specifies a custom name for the analog output channel that you select.

Delete Channel	Deletes the analog output channel that you select. (myRIO Toolkit) If you use the myRIO high-throughput FPGA personality, this option is available only when you specify Analog output (1 sample) for **I/O mode**.
Add Channel	Adds a new analog output channel to the channel list. You can add up to eight analog output channels for the myRIO. You can add up to two analog output channels for the roboRIO. (myRIO Toolkit) If you use the myRIO high-throughput FPGA personality, this option is available only when you specify Analog output (1 sample) for **I/O mode**.
Sample rate	(myRIO Toolkit) Specifies the sampling frequency of the output signal. Valid values are between 1 kHz and 80 kHz. If you specify a frequency that is invalid, this Express VI coerces the specified value to the nearest valid value when you click the **Validate** button. This option is available only when you use the myRIO high-throughput FPGA personality and specify Analog output (n samples) for **I/O mode**. - **Frequency value**—Specifies the value of the sampling frequency. The default is 1 - **Frequency unit**—Shows the unit of the sampling frequency. The value is kHz. - **Validate**—Validates whether this Express VI can generate the sampling frequency that you specify. If the specified sampling frequency is not valid, this Express VI coerces the specified value to the nearest valid value.
Wait until done?	(myRIO Toolkit) Specifies whether this Express VI waits until the write operation completes. If the **Wait until done?** checkbox contains a checkmark, this Express VI waits until the write operation completes. By default, this checkbox does not contain a checkmark. This option is available

	only when you use the myRIO high-throughput FPGA personality and specify Analog output (n samples) for **I/O mode**.
Latency	(myRIO Toolkit) Displays the latency between two adjacent signal generation iterations. Refer to the Details section of this topic for more information about latency. This option is available only when you use the myRIO high-throughput FPGA personality and specify Analog output (n samples) for **I/O mode**.
View Code	Displays the underlying code of this Express VI.
Connection Diagram	Shows the I/O connector pinouts on the myRIO or the roboRIO. The highlighted pinouts represent the channels that you configure.

Block Diagram Inputs

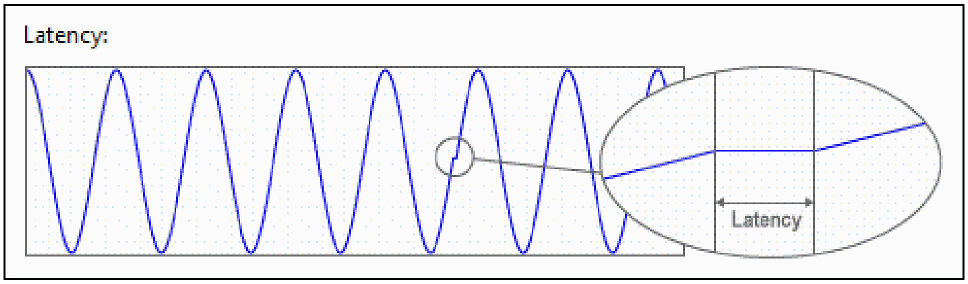
Parameter	Description
channel name	Specifies the value to write to the analog output channel that you select, where **channel name** is the name of the analog output channel.
error in (no error)	Describes error conditions that occur before this node runs.

Block Diagram Outputs

Parameter	Description
error out	Contains error information. This output provides standard error out functionality.

Analog Output Details

(myRIO Toolkit) The following figure demonstrates the latency when you use the Analog Output Express VI with the myRIO high-throughput FPGA personality to perform **n** samples write operations.



In the previous figure, the x-axis represents time and the y-axis represents amplitude. The time interval between two adjacent signal generation iterations is latency. In other words, the myRIO does not export signals when latency occurs.

Related Information

1 Sample versus N Samples Modes (myRIO Toolkit)

Generating FPGA Clocks (myRIO Toolkit)

Generating FPGA Clocks (roboRIO Toolkit)

I/O Connectors (myRIO Toolkit)

I/O Connectors (roboRIO Toolkit)

Latency in N Samples Read and Write Operations (myRIO Toolkit)

Power Supply for Peripheral Devices (roboRIO Toolkit)

Audio Input Express VI

Owning Palette: High Throughput FPGA Personality VIs

Requires: myRIO Toolkit

Reads values from one or more audio input channels on the myRIO. Use this Express VI to read multiple samples for each channel at one time.

Details

Dialog Box Options
Block Diagram Inputs
Block Diagram Outputs

Dialog Box Options

Parameter	Description
Node name	Specifies the name of this Express VI. You can also double-click the name of this Express VI on the expandable node to edit the name.
Channel	Specifies the audio input channel from which to read the value.
Custom channel name	Specifies a custom name for the audio input channel that you select.
Delete Channel	Deletes the audio input channel that you select.
Add channel	Adds a new audio input channel to the channel list. You can add up to two audio input channels.
Sample rate	Specifies the sampling frequency of the input signal. Valid values are between 1 kHz and 44.1 kHz. If you specify a frequency that is invalid, this Express VI coerces the specified value to the nearest valid value when you click the **OK** button on the Express VI. - **Frequency value**—Specifies the value of the sampling frequency. The default is 1,000. - **Frequency unit**—Specifies the unit of the sampling frequency. - **Validate**—Validates whether this VI can generate the sampling frequency that you specify. If the specified sampling frequency is invalid, this Express VI coerces the specified value to the nearest valid value.
Samples	Specifies the number of samples to read. The default is 1,000. Valid values must be greater than 0 and less than or equal to 10,000.
Latency	Displays the latency between two adjacent signal generation iterations. Refer to the Details section of this topic for more information about latency.
View Code	Displays the underlying code of this Express VI.

Connection Diagram	Shows the I/O connector pinouts on the myRIO or the roboRIO. The highlighted pinouts represent the channels that you configure.

Block Diagram Inputs

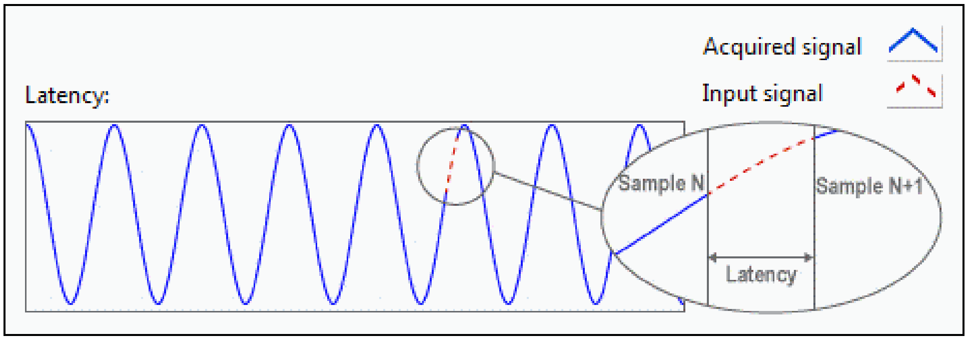
Parameter	Description
error in (no error)	Describes error conditions that occur before this node runs.

Block Diagram Outputs

Parameter	Description
channel name	Returns the value that this Express VI reads from the audio input channel that you select, where **channel name** is the name of the audio input channel.
error out	Contains error information. This output provides standard error out functionality.

Audio Input Details

The following figure demonstrates the latency when you use the Audio Input Express VI to perform **n** samples read operations.



In the previous figure, the x-axis represents time and the y-axis represents amplitude. The waveform in blue represents the signal that the myRIO acquires. The red dotted line represents latency. When latency occurs, the myRIO does not acquire any signal. In other words, the time interval between two adjacent signal acquisition iterations is latency.

Related Information

Generating FPGA Clocks

Latency in N Samples Read and Write Operations

Audio Output Express VI

Owning Palette: High Throughput FPGA Personality VIs

Requires: myRIO Toolkit

Writes values to one or more audio output channels on the myRIO. Use this Express VI to write multiple samples for each channel at one time.

Details

Dialog Box Options

Parameter	Description
Node name	Specifies the name of this Express VI. You can al so double-click the name of this Express VI on th e expandable node to edit the name.
Channel	Specifies the audio output channel to which to write the value.
Custom channel name	Specifies a custom name for the audio output c hannel that you select.
Delete Channel	Deletes the audio output channel that you selec t.
Add channel	Adds a new audio out channel to the channel lis t. You can add up to two analog output channel s.
Sample rate	Specifies the sampling frequency in hertz of the output signal. Valid values are between 1 kHz an d 80 kHz. If you specify a frequency that is invali d, this Express VI coerces the specified value to t

	he nearest valid value when you click the **OK** button on the Express VI. - **Frequency value**—Specifies the value of the sampling frequency. The default is 1,000. - **Frequency unit**—Specifies the unit of the sampling frequency. - **Validate**—Validates whether this Express VI can generate the sampling frequency that you specify. If the specified sampling frequency is invalid, this Express VI coerces the specified value to the nearest valid value.
Wait until done?	Specifies whether this Express VI waits until the write operation completes. If the **Wait until done?** checkbox contains a checkmark, this Express VI waits until the write operation completes. By default, this checkbox does not contain a checkmark.
Latency	Displays the latency between two adjacent signal generation iterations. Refer to the Details section of this topic for more information about latency.
View Code	Displays the underlying code of this Express VI.
Connection Diagram	Shows the I/O connector pinouts on the myRIO or the roboRIO. The highlighted pinouts represent the channels that you configure.

Block Diagram Inputs

Parameter	Description
error in (no error)	Describes error conditions that occur before this node runs.

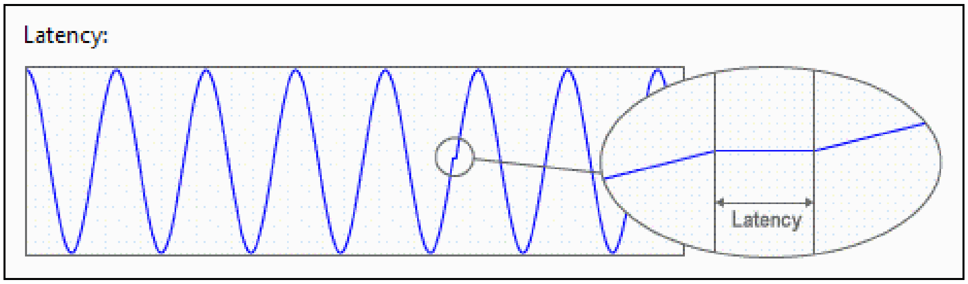
Block Diagram Outputs

Parameter	Description

channel name	Returns the value that this Express VI writes to the audio output channel that you select, where **channel name** is the name of the audio output channel.
error out	Contains error information. This output provides standard error out functionality.

Audio Output Details

The following figure demonstrates the latency when you use the Audio Output Express VI to perform **n** samples write operations.



In the previous figure, the x-axis represents time and the y-axis represents amplitude. The time interval between two adjacent signal generation iterations is latency. In other words, the myRIO does not export signals when latency occurs.

Related Information

Generating FPGA Clocks

Latency in N Samples Read and Write Operations

Button Express VI

Requires: myRIO Toolkit **or** roboRIO Toolkit

Reads the value from the user button on the myRIO or the roboRIO.

Examples

Dialog Box Options
Block Diagram Inputs
Block Diagram Outputs

Dialog Box Options

Parameter	Description
Node name	Specifies the name of this Express VI. You can al so double-click the name of this Express VI on th e expandable node to edit the name.
View Code	Displays the underlying code of this Express VI.

Block Diagram Inputs

Parameter	Description
error in (no error)	Describes error conditions that occur before this node runs.

Block Diagram Outputs

Parameter	Description
Value	Returns the value this Express VI reads from the user button on the myRIO or the roboRIO.
error out	Contains error information. This output provide s standard error out functionality.

Examples

Refer to the following VIs for examples of using the Button Express VI:

- `labview\examples\myRIO\Up and Down Binary Counter\Up and Down Binary Counter.lvproj`
- `labview\examples\roboRIO\Up and Down Binary Counter\Up and Down Binary Counter.lvproj`

Digital Input Express VI

Requires: myRIO Toolkit **or** roboRIO Toolkit

Reads values from one or more digital input channels on the myRIO or the roboRIO.

This Express VI reads one sample each time with the default FPGA personality on the myRIO. This Express VI reads one sample or multiple samples each time with the high-throughput FPGA personality on the myRIO. Visit `ni.com/info` and enter the Info Code ex6g5a to learn about the myRIO high-throughput FPGA personality.

This Express VI reads one sample each time with the default FPGA personality on the roboRIO. The roboRIO uses a 5 V voltage rail on the DIO port for powering sensors and provides 3.3 V DIO lines for generating digital input signals.

Details

Dialog Box Options

Parameter	Description
I/O mode	(myRIO Toolkit) Specifies to read one sample or multiple samples. The default is Digital input (1 sample). This option is available only when you use the myRIO high-throughput FPGA personalit y.
Node name	Specifies the name of this Express VI. You can al so double-click the name of this Express VI on th e expandable node to edit the name.
Channel	Specifies the digital input channel from which t o read the value. If the channel you select is set as a digital output channel, this Express VI chan ges the channel to a digital input channel before reading the value.
Custom channel name	Specifies a custom name for the digital input ch annel that you select.
Delete Channel	Deletes the digital input channel that you select . (myRIO Toolkit) If you use the myRIO high-thro ughput FPGA personality, this option is availabl e only when you specify Digital input (1 sample) for **I/O mode**.
Add Channel	Adds a new digital input channel to the channel list. You can add up to 12 digital input channels. (myRIO Toolkit) If you use the myRIO high-throu ghput FPGA personality, this option is available

	only when you specify Digital input (1 sample) for **I/O mode**.
Sample rate	(myRIO Toolkit) Specifies the sampling frequency of the input signal. Valid values are between 1 kHz and 8 MHz. If you specify a frequency that is invalid, this Express VI coerces the specified value to the nearest valid value when you click the **Validate** button. This option is available only when you use the myRIO high-throughput FPGA personality and specify Digital input (n samples) for **I/O mode**. - **Frequency value**—Specifies the value of the sampling frequency. The default is 1. - **Frequency unit**—Specifies the unit of the sampling frequency. The default is kHz. - **Validate**—Validates whether this Express VI can generate the sampling frequency that you specify. If the specified sampling frequency is not valid, this Express VI coerces the specified value to the nearest valid value.
Samples	(myRIO Toolkit) Specifies the number of samples to read. The default is 1,000. Valid values must be greater than 0 and less than or equal to 10,000. This option is available only when you use the myRIO high-throughput FPGA personality and specify Digital output (n samples) for **I/O mode**.
Latency	(myRIO Toolkit) Displays the latency between two adjacent signal acquisition iterations. Refer to the Details section of this topic for more information about latency. This option is available only when you use the myRIO high-throughput FPGA personality and specify Digital input (n samples) for **I/O mode**.
View Code	Displays the underlying code of this Express VI.

Connection Diagram	Shows the I/O connector pinouts on the myRIO or the roboRIO. The highlighted pinouts represent the channels that you configure.

Block Diagram Inputs

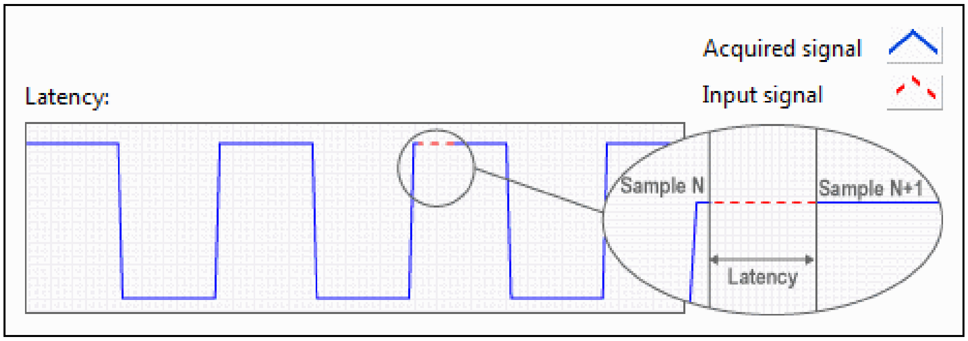
Parameter	Description
error in (no error)	Describes error conditions that occur before this node runs.

Block Diagram Outputs

Parameter	Description
channel name	Returns the value that this Express VI reads from the digital input channel that you select, where **channel name** is the name of the digital input channel.
error out	Contains error information. This output provides standard error out functionality.

Digital Input Details

(myRIO Toolkit) The following figure demonstrates the latency when you use the Digital Input Express VI with the myRIO high-throughput FPGA personality to perform **n** samples read operations.



In the previous figure, the x-axis represents time and the y-axis represents amplitude. The waveform in blue represents the signal that the myRIO acquires. The red dotted line represents latency. When latency occurs, the myRIO does not acquire any signal. In other words, the time interval between two adjacent signal acquisition iterations is latency.

Related Information

1 Sample versus N Samples Modes (myRIO Toolkit)

Generating FPGA Clocks (myRIO Toolkit)

Generating FPGA Clocks (roboRIO Toolkit)

I/O Connectors (myRIO Toolkit)

I/O Connectors (roboRIO Toolkit)

Latency in N Samples Read and Write Operations (myRIO Toolkit)

Power Supply for Peripheral Devices (roboRIO Toolkit)

Digital Output Express VI

Requires: myRIO Toolkit **or** roboRIO Toolkit

Writes values to one or more digital output channels on the myRIO or the roboRIO.

This Express VI writes one sample each time with the default FPGA personality on the myRIO. This Express VI writes one sample or multiple samples each time with the high-throughput FPGA personality on the myRIO. Visit `ni.com/info` and enter the Info Code ex6g5a to learn about the myRIO high-throughput FPGA personality.

This Express VI writes one sample each time with the default FPGA personality on the roboRIO. The roboRIO uses a 5 V voltage rail on the DIO port for powering sensors and provides 3.3 V DIO lines for generating digital output signals.

Details

Dialog Box Options
Block Diagram Inputs
Block Diagram Outputs

Dialog Box Options

Parameter	Description
I/O mode	(myRIO Toolkit) Specifies to write one sample or multiple samples. The default is Digital output (

	1 sample). This option is available only when you use the myRIO high-throughput FPGA personality.
Node name	Specifies the name of this Express VI. You can also double-click the name of this Express VI on the expandable node to edit the name.
Channel	Specifies the digital output channel to which to write a value. If the channel you select is set as a digital input channel, this Express VI changes the channel to a digital output channel before writing the value.
Custom channel name	Specifies a custom name for the digital output channel that you select.
Delete Channel	Deletes the digital output channel that you select. (myRIO Toolkit) If you use the myRIO high-throughput FPGA personality, this option is available only when you specify Digital output (1 sample) for **I/O mode**.
Add Channel	Adds a new digital output channel to the channel list. You can add up to 12 digital output channels. (myRIO Toolkit) If you use the myRIO high-throughput FPGA personality, this option is available only when you specify Digital output (1 sample) for **I/O mode**.
Sample rate	(myRIO Toolkit) Specifies the sampling frequency of the output signal. Valid values are between 1 kHz and 8 MHz. If you specify a frequency that is invalid, this Express VI coerces the specified value to the nearest valid value when you click the **Validate** button. This option is available only when you use the myRIO high-throughput FPGA personality and specify Digital output (n samples) for **I/O mode**. ▪ **Frequency value**—Specifies the value of the sampling frequency. The default is 1. ▪ **Frequency unit**—Specifies the unit of the sampling frequency. The default is kHz.

	- **Validate**—Validates whether this Express VI can generate the sampling frequency that you specify. If the specified sampling frequency is not valid, this Express VI coerces the specified value to the nearest valid value.
Wait until done?	(myRIO Toolkit) Specifies whether this Express VI waits until the write operation completes. If the **Wait until done?** checkbox contains a checkmark, this Express VI waits until the write operation completes. By default, this checkbox does not contain a checkmark. This option is available only when you use the myRIO high-throughput FPGA personality and specify Analog output (n samples) for **I/O mode**.
Latency	(myRIO Toolkit) Displays the latency between two adjacent signal generation iterations. Refer to the Details section of this topic for more information about latency. This option is available only when you use the myRIO high-throughput FPGA personality and specify Digital output (n samples) for **I/O mode**.
View Code	Displays the underlying code of this Express VI.
Connection Diagram	Shows the I/O connector pinouts on the myRIO or the roboRIO. The highlighted pinouts represent the channels that you configure.

Block Diagram Inputs

Parameter	Description
channel name	Specifies the value to write to the digital output channel that you select, where **channel name** is the name of the digital output channel.
error in (no error)	Describes error conditions that occur before this node runs.

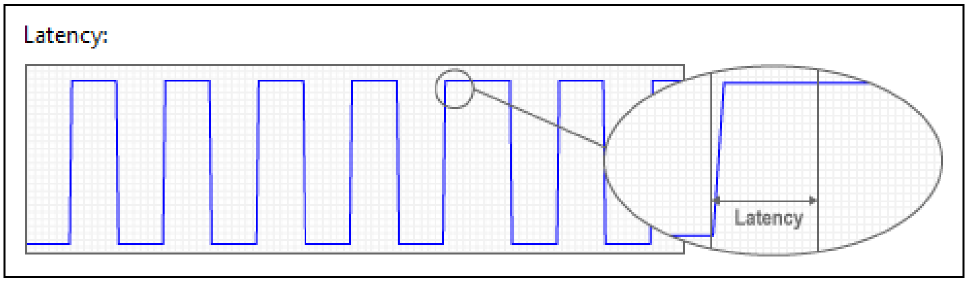
Block Diagram Outputs

Parameter	Description

error out	Contains error information. This output provide s standard error out functionality.

Digital Output Details

(myRIO Toolkit) The following figure demonstrates the latency when you use the Digital Output Express VI with the myRIO high-throughput FPGA personality to perform **n** samples write operations.



In the previous figure, the x-axis represents time and the y-axis represents amplitude. The time interval between two adjacent signal generation iterations is latency. In other words, the myRIO does not export signals when latency occurs.

Related Information

1 Sample versus N Samples Modes (myRIO Toolkit)

Generating FPGA Clocks (myRIO Toolkit)

Generating FPGA Clocks (roboRIO Toolkit)

I/O Connectors (myRIO Toolkit)

I/O Connectors (roboRIO Toolkit)

Latency in N Samples Read and Write Operations (myRIO Toolkit)

Power Supply for Peripheral Devices (roboRIO Toolkit)

LED Express VI

Requires: myRIO Toolkit **or** roboRIO Toolkit

Sets the states of the LEDs on the myRIO or the roboRIO.

Details

Dialog Box Options

Parameter	Description
Node name	Specifies the name of this Express VI. You can also double-click the name of this Express VI on the expandable node to edit the name.
LED0	Enables state setting for LED0 on the myRIO.
LED1	Enables state setting for LED1 on the myRIO.
LED2	Enables state setting for LED2 on the myRIO.
LED3	Enables state setting for LED3 on the myRIO.
RADIO (Green)	Enables state setting for RADIO (Green) on the roboRIO.
RADIO (Red)	Enables state setting for RADIO (Red) on the roboRIO.
COMM (Green)	Enables state setting for COMM (Green) on the roboRIO.
COMM (Red)	Enables state setting for COMM (Red) on the roboRIO.
MODE (Green)	Enables state setting for MODE (Green) on the roboRIO.
MODE (Red)	Enables state setting for MODE (Red) on the roboRIO.
Custom channel name	Specifies a custom name for the LED that you select.
View Code	Displays the underlying code of this Express VI.

Block Diagram Inputs

Parameter	Description
LED0	Sets the state of LED0 on the myRIO.
LED1	Sets the state of LED1 on the myRIO.
LED2	Sets the state of LED2 on the myRIO.

LED3	Sets the state of LED3 on the myRIO.
RADIO (Green)	Sets the state of RADIO (Green) on the roboRIO.
RADIO (Red)	Sets the state of RADIO (Red) on the roboRIO.
COMM (Green)	Sets the state of COMM (Green) on the roboRIO.
COMM (Red)	Sets the state of COMM (Red) on the roboRIO.
MODE (Green)	Sets the state of MODE (Green) on the roboRIO.
MODE (Red)	Sets the state of MODE (Red) on the roboRIO.
error in (no error)	Describes error conditions that occur before this node runs.

Block Diagram Outputs

Parameter	Description
error out	Contains error information. This output provides standard error out functionality.

LED Details

You can set either ON or OFF state for the myRIO LEDs.

The roboRIO contains the RADIO, COMM, and MODE LEDs. Every LED associates with two Boolean controls for displaying various colors. For example, the RADIO LED associates with the **RADIO (Green)** control and the **RADIO (Red)** control. The following table shows the configuration options to display a green, red, or orange RADIO LED:

	Green	Red	Orange	Off
RADIO (Green)	TRUE	FALSE	TRUE	FALSE
RADIO (Red)	FALSE	TRUE	TRUE	FALSE

UART Express VI

Requires: myRIO Toolkit **or** roboRIO Toolkit

Writes data to or reads data from a Universal Asynchronous Receiver/Transmitter (UART) device through the UART channels on the myRIO or the roboRIO.

With the roboRIO, you also can use this VI to write data to or read data from an RS-232 device through the RS-232 channel.

Details

Dialog Box Options
Block Diagram Inputs
Block Diagram Outputs

Dialog Box Options

Parameter	Description
Node name	Specifies the name of this Express VI. You can also double-click the name of this Express VI on the expandable node to edit the name.
Channel	Specifies the UART channel on the myRIO or the roboRIO to write data to or read data from the UART device. With the roboRIO, you also can specifies the RS-232 channel to write data to or read data from the RS-232 device.
Connections	Specifies the myRIO or the roboRIO pins that correspond to the receive input line and the transmit output line.
Mode	Specifies the mode of operation for communicating with the UART device. **Mode** contains the following options: ▪ **Write**—Specifies to write data to the UART device. With the roboRIO, you also can specify to write data to the RS-232 device. ▪ **Read**—Specifies to read data from the UART device. With the roboRIO, you also can specify to read data from the RS-232 device. ▪ **Read all available**—Specifies whether to read all available characters from the UART device. With the roboRIO, you also can specify whether to read all avai

	lable characters from the RS-232 device. The default is FALSE.
Communication settings	Specifies the configuration for communicating with the UART device. With the roboRIO, you also can specify the configuration for communicating with the RS-232 device. **Communication settings** contains the following options: - **Baud rate**—Specifies the baud rate of transmission. The default is 9,600. The maximum baud rate is 230,400 for UART lines and 115,200 for RS-232 lines. - **Data bits**—Specifies the number of bits in the incoming data. The default is 8. - **Parity**—Specifies the parity bits to write or read characters. The default is None. - **Stop bits**—Specifies the number of stop bits this Express VI uses to indicate the end of a data frame. The default is 1.0.
View Code	Displays the underlying code of this Express VI.
Connection Diagram	Shows the I/O connector pinouts on the myRIO or the roboRIO. The highlighted pinouts represent the channels that you configure.

Block Diagram Inputs

Parameter	Description
Characters to Write	Specifies the characters to write to the UART device. With the roboRIO, you also can specify the characters to write to the RS-232 device. This input is available only when you set **Mode** to **Write**.
Character Count	Specifies the number of characters to read from the UART device. With the roboRIO, you also can specify the number of characters to read from the RS-232 device. This input is available only when you set **Mode** to **Read**.

error in (no error)	Describes error conditions that occur before this node runs.

Block Diagram Outputs

Parameter	Description
Character Count	Returns the number of characters that this Express VI reads from the UART device. With the roboRIO, this output also returns the number of characters that this Express VI reads from the RS-232 device. This output is available when you set **Mode** to **Write**.
Characters Read	Returns the characters that this Express VI reads from the UART device. With the roboRIO, this output also returns the characters that this Express VI reads from the RS-232 device. This output is available when you set **Mode** to **Read**.
error out	Contains error information. This output provides standard error out functionality.

UART Details

The UART lines on the myRIO and roboRIO MXP ports are electrically identical to DIO lines on the MXP port. The UART signals are transistor-transistor logic (TTL) compatible and have the following characteristics:

- Logic low—0 V to 0.8 V
- Logic high—2 V to 5 V

The RS-232 lines on the roboRIO are compliant with TIA/EIA-232-F voltage levels. The following are the valid voltage levels:

- Logic one—-5 V to -15 V
- Logic zero—+5 V to +15 V

Related Information

I/O Connectors (myRIO Toolkit)

I/O Connectors (roboRIO Toolkit)

TTL-Compatible Signals

Serial Port Communication

Serial VIs and Functions

Device Management VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit. This topic might not match its corresponding palette in LabVIEW depending on your operating system, licensed product(s), and target.

Use the Device Management VIs to set custom FPGA bitfiles and to reset I/O channels on the myRIO or the roboRIO.

Palette Object	Description
Reset	Resets the FPGA target and all the I/O channels on the myRIO or the roboRIO.
Set Custom Bitfile	Sets a custom FPGA reference.

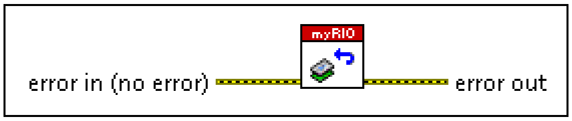
Reset VI

Owning Palette: Device Management VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Resets the FPGA target and all the I/O channels on the myRIO or the roboRIO.

Details Examples



error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

error out contains error information. This output provides standard error out functionality.

Reset Details

This VI resets the FPGA target even when code is running on the FPGA target and no matter whether there are incoming errors. Use this VI only with the Express VIs and run this VI only once at the end of an application.

Examples

Refer to the following VIs for examples of using the Reset VI:

- `labview\examples\myRIO\Data Dashboard\Data Dashboard for myRIO.lvproj`
- `labview\examples\myRIO\Edge Detection and Debouncing\Edge Detection and Debouncing.lvproj`
- `labview\examples\myRIO\Up and Down Binary Counter\Up and Down Binary Counter.lvproj`
- `labview\examples\roboRIO\Edge Detection and Debouncing\Edge Detection and Debouncing.lvproj`
- `labview\examples\roboRIO\Up and Down Binary Counter\Up and Down Binary Counter.lvproj`

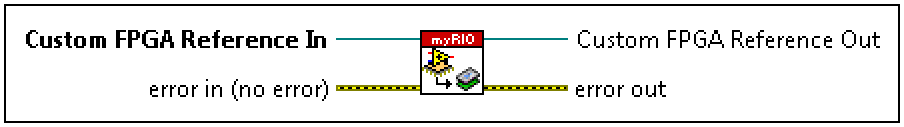
Set Custom Bitfile VI

Owning Palette: Device Management VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Sets a custom FPGA reference.

Details Examples



Custom FPGA Reference In specifies the input custom FPGA reference.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

Custom FPGA Reference Out returns the custom FPGA reference.

error out contains error information. This output provides standard error out functionality.

Set Custom Bitfile Details

You must set a custom FPGA reference before using a custom FPGA bitfile with the myRIO VIs or the roboRIO VIs. Use the Open FPGA VI Reference function to open a reference to the custom FPGA bitfile. Use the Close FPGA VI Reference function to close the reference at the end of an application.

Related Information

Open FPGA VI Reference Function

Close FPGA VI Reference Function

Examples

Refer to the following VIs for examples of using the Set Custom Bitfile VI:

- `labview\examples\myRIO\Customized FPGA\Customized FPGA Signal Generator.lvproj`
- `labview\examples\roboRIO\Customized FPGA\Customized FPGA Signal Generator.lvproj`

Low Level VIs

Owning Palette: High Throughput FPGA Personality VIs

Requires: myRIO Toolkit. This topic might not match its corresponding palette in LabVIEW depending on your operating system, licensed product(s), and target.

Use the Low Level VIs to create applications for different I/O types on the myRIO. The VIs on this palette are compatible with the high-throughput FPGA personality

Subpalette	Description
Accelerometer VIs	Use the Accelerometer VIs to control the onboard accelerometer on the myRIO or the roboRIO.
Analog Input 1 Sample VIs	Use the Analog Input 1 Sample VIs to control the analog input channels on the myRIO or the roboRIO.
Analog Output 1 Sample VIs	Use the Analog Output 1 Sample VIs to control the analog output channels on the myRIO or the roboRIO.
Analog Input N Samples VIs	Use the Analog Input N Samples VIs to perform analog input operations on the myRIO, such as reading multiple samples for each analog input channel at one time.
Analog Output N Samples VIs	Use the Analog Output N Samples VIs to perform analog output operations on the myRIO, such as writing multiple samples for each analog output channel at one time.
Audio Input N Samples VIs	Use the Analog Input VIs to perform audio input operations on the myRIO, such as reading multiple samples for each audio input channel at one time.
Audio Output N Samples VIs	Use the Audio Output N Samples VIs to perform audio output operations on the myRIO, such as writing multiple samples for each audio output channel at one time.
Digital Input/Output 1 Sample VIs	Use the Digital Input/Output 1 Sample VIs to control the digital I/O channels on the myRIO or the roboRIO.
Digital Input N Samples VIs	Use the Digital Input N Samples VIs to perform digital input operations on the myRIO, such as reading multiple samples for each digital input channel at one time.
Digital Output N Samples VIs	Use the Digital Output N Samples VIs to perform digital output operations on the myRIO, such as writing multiple samples for each digital output channel at one time.

This palette also contains the following subpalettes:

- Input Device Control VIs
- Serial VIs and Functions

Related Information

Choosing between Express VIs and Low Level VIs

Accelerometer VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit. This topic might not match its corresponding palette in LabVIEW depending on your operating system, licensed product(s), and target.

Use the Accelerometer VIs to control the onboard accelerometer on the myRIO or the roboRIO.

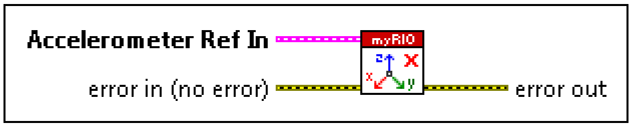
Palette Object	Description
Close	Closes the reference to accelerometer channels.
Open	Opens a reference to one or more accelerometer channels. You must open a reference before you read values from an accelerometer channel.
Read	Reads acceleration values from the onboard accelerometer. You must open and configure accelerometer channels before reading values from the accelerometer channels.

Close VI

Owning Palette: Accelerometer VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Closes the reference to accelerometer channels.



Accelerometer Ref In specifies the reference to the accelerometer channels. Use the Open VI to open a reference to the accelerometer channels. Do not modify the **Accelerometer Ref In** values.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

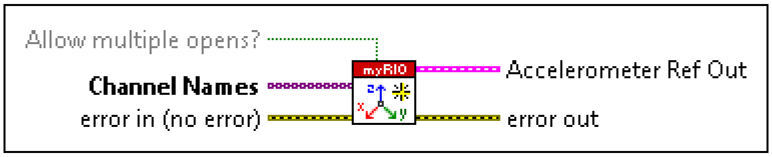
error out contains error information. This output provides standard error out functionality.

Open VI

Owning Palette: Accelerometer VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Opens a reference to one or more accelerometer channels. You must open a reference before you read values from an accelerometer channel.



Allow multiple opens? specifies whether to allow opening the specified channels more than once. The default is FALSE. You must set **Allow multiple opens?** to TRUE if you also use the specified channels in an Express VI.

Channel Names specifies the names of the accelerometer channels to open a reference.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

Accelerometer Ref Out returns reference to the accelerometer channels that you specify.

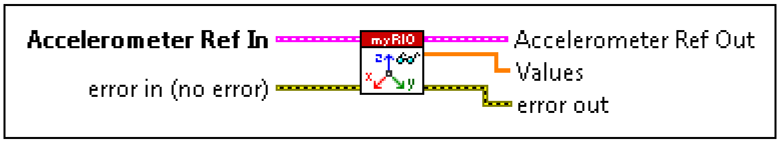
error out contains error information. This output provides standard error out functionality.

Read VI

Owning Palette: Accelerometer VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Reads acceleration values from the onboard accelerometer. You must open and configure accelerometer channels before reading values from the accelerometer channels.



Accelerometer Ref In specifies the reference to the accelerometer channels. Use the Open VI to open a reference to the accelerometer channels. Do not modify the **Accelerometer Ref In** values.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

Accelerometer Ref Out returns the reference to the accelerometer channels that you specify.

Values returns the acceleration values that this VI reads from the accelerometer channels.

error out contains error information. This output provides standard error out functionality.

Analog Input 1 Sample VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit. This topic might not match its corresponding palette in LabVIEW depending on your operating system, licensed product(s), and target.

Use the Analog Input 1 Sample VIs to control the analog input channels on the myRIO or the roboRIO.

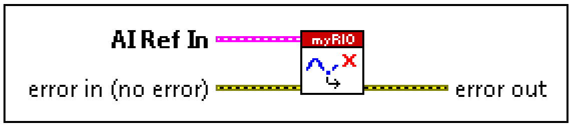
Palette Object	Description
Close	Closes the reference to one or more analog input channels.
Open	Opens a reference to one or more analog input channels. You must open a reference before you read values from an analog input channel.
Read	Reads values from one or more analog input channels.

Close VI

Owning Palette: Analog Input 1 Sample VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Closes the reference to one or more analog input channels.



AI Ref In specifies the reference to the analog input channels. Use the Open VI to open a reference to the analog input channels. Do not modify the **AI Ref In** values.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

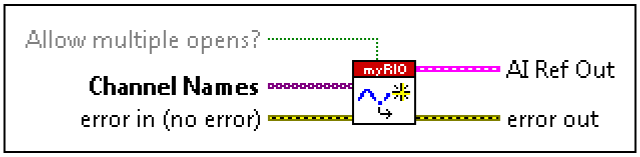
error out contains error information. This output provides standard error out functionality.

Open VI

Owning Palette: Analog Input 1 Sample VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Opens a reference to one or more analog input channels. You must open a reference before you read values from an analog input channel.



Allow multiple opens? specifies whether to allow opening the specified channels more than once. The default is FALSE. You must set **Allow multiple opens?** to TRUE if you also use the specified channels in an Express VI.

Channel Names specifies the names of the analog input channels whose reference you want to open.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

AI Ref Out returns a reference to the analog input channels that you specify.

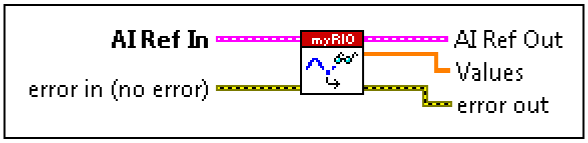
error out contains error information. This output provides standard error out functionality.

Read VI

Owning Palette: Analog Input 1 Sample VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Reads values from one or more analog input channels.



AI Ref In specifies the reference to the analog input channels from which to read values. Use the Open VI to open a reference to the analog input channels. Do not modify the **AI Ref In** values.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

AI Ref Out returns the reference to the analog input channels from which this VI reads values.

Values returns the values in volts that this VI reads from the analog input channels. The order of the values corresponds to the order in which the Open VI opens the analog input channels.

error out contains error information. This output provides standard error out functionality.

Analog Input N Samples VIs

Owning Palette: Low Level VIs

Requires: myRIO Toolkit. This topic might not match its corresponding palette in LabVIEW depending on your operating system, licensed product(s), and target.

Use the Analog Input N Samples VIs to perform analog input operations on the myRIO, such as reading multiple samples for each analog input channel at one time.

Palette Object	Description

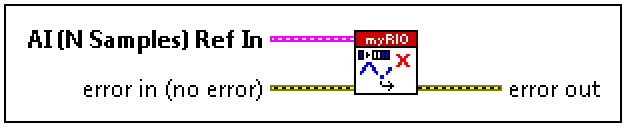
Close	Closes the reference to one or more analog input channels, which you use to perform n samples read operations.
Open	Opens a reference to one or more analog input channels, which you use to perform n samples read operations. You must open a reference before you read values from an analog input channel.
Read	Reads values from one or more analog input channels. Use this VI to read multiple samples for each channel at one time.

Close VI

Owning Palette: Analog Input N Samples VIs

Requires: myRIO Toolkit

Closes the reference to one or more analog input channels, which you use to perform n samples read operations.



AI (N Samples) Ref In specifies the reference to the analog input channels from which to read values. Use the Open VI to open a reference to the analog input channels. Do not modify the **AI (N Samples) Ref In** values.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

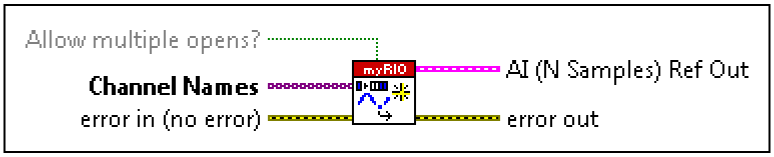
error out contains error information. This output provides standard error out functionality.

Open VI

Owning Palette: Analog Input N Samples VIs

Requires: myRIO Toolkit

Opens a reference to one or more analog input channels, which you use to perform n samples read operations. You must open a reference before you read values from an analog input channel.



Allow multiple opens? specifies whether to allow opening the specified channels more than once. The default is FALSE. You must set **Allow multiple opens?** to TRUE if you also use the specified channels in an Express VI.

Channel Names specifies the names of the analog input channels whose reference you want to open.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

AI (N Samples) Ref Out returns a reference to the analog input channels that you specify.

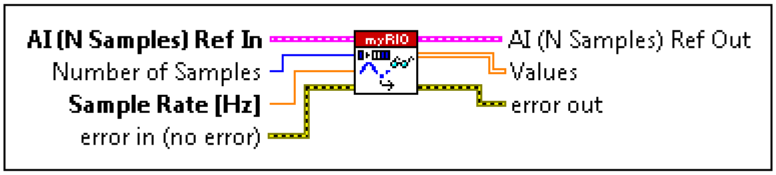
error out contains error information. This output provides standard error out functionality.

Read VI

Owning Palette: Analog Input N Samples VIs

Requires: myRIO Toolkit

Reads values from one or more analog input channels. Use this VI to read multiple samples for each channel at one time.



AI (N Samples) Ref In specifies the reference to the analog input channels from which to read values. Use the Open VI to open a reference to the analog input channels. Do not modify the **AI (N Samples) Ref In** values.

U32 **Number of Samples** specifies the number of samples to read. The default is 1,000. Valid values are between 0 and 10,000.

DBL **Sample Rate** specifies the sampling frequency in hertz of the input signal. The default is 1,000. Valid values are between 1 k and 30 k.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

AI (N Samples) Ref Out returns the reference to the analog input channels from which this VI reads values.

[DBL] **Values** returns the values in volts that this VI reads from the analog input channels. The order of the values corresponds to the order in which the Open VI opens the analog input channels.

error out contains error information. This output provides standard error out functionality.

Analog Output 1 Sample VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit. This topic might not match its corresponding palette in LabVIEW depending on your operating system, licensed product(s), and target.

Use the Analog Output 1 Sample VIs to control the analog output channels on the myRIO or the roboRIO.

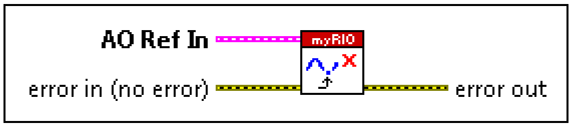
Palette Object	Description
Close	Closes the reference to one or more analog output channels and resets the output voltage to 0 V.
Open	Opens a reference to one or more analog output channels. You must open a reference before you write values to an analog output channel.
Write	Writes values to one or more analog output channels.

Close VI

Owning Palette: Analog Output 1 Sample VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Closes the reference to one or more analog output channels and resets the output voltage to 0 V.



AO Ref In specifies the reference to the analog output channels. Do not modify the **AO Ref In** values.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

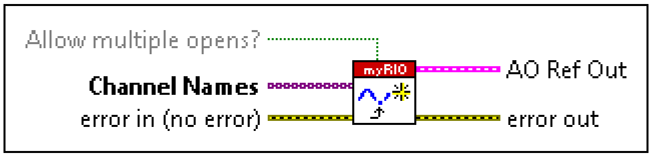
error out contains error information. This output provides standard error out functionality.

Open VI

Owning Palette: Analog Output 1 Sample VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Opens a reference to one or more analog output channels. You must open a reference before you write values to an analog output channel.



Allow multiple opens? specifies whether to allow opening the specified channels more than once. The default is FALSE. You must set **Allow multiple opens?** to TRUE if you also use the specified channels in an Express VI.

Channel Names specifies the names of the analog output channels to open a reference.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

AO Ref Out returns a reference to the analog output channels that you specify.

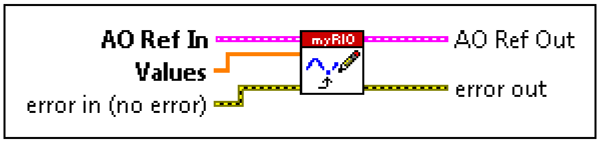
error out contains error information. This output provides standard error out functionality.

Write VI

Owning Palette: Analog Output 1 Sample VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Writes values to one or more analog output channels.



AO Ref In specifies the reference to the analog output channels to which to write values. Use the Open VI to open a reference to the analog output channels. Do not modify the **AO Ref In** values.

Values specifies the values, in volts, to write to the analog output channels. The order of the values corresponds to the order in which the Open VI opens the analog output channels. You must assign a value for each channel that you specify.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

AO Ref Out returns the reference to the analog output channels to which this VI writes values.

error out contains error information. This output provides standard error out functionality.

Analog Output N Samples VIs

Owning Palette: Low Level VIs

Requires: myRIO Toolkit. This topic might not match its corresponding palette in LabVIEW depending on your operating system, licensed product(s), and target.

Use the Analog Output N Samples VIs to perform analog output operations on the myRIO, such as writing multiple samples for each analog output channel at one time.

Palette Object	Description
Close	Closes the reference to one or more analog output channels, which you use to perform n samples write operations.
Open	Opens a reference to one or more analog output channels, which you use to perform n samples write operations. You must open a reference before you write values to an analog output channel.

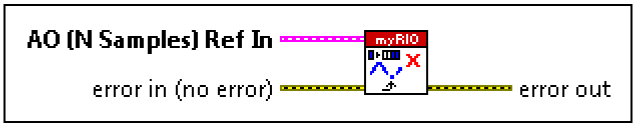
Write	Writes values to one or more analog output channels. Use this VI to write multiple samples for each channel at one time.

Close VI

Owning Palette: Analog Output N Samples VIs

Requires: myRIO Toolkit

Closes the reference to one or more analog output channels, which you use to perform n samples write operations.



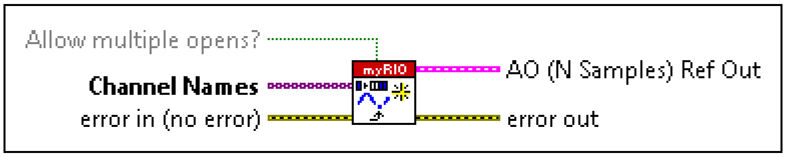
▭	**AO (N Samples) Ref In** specifies the reference to the analog output channels to which to write values. Use the Open VI to open a reference to the analog output channels. Do not modify the **AO (N Samples) Ref In** values.
▭	**error in** describes error conditions that occur before this node runs. This input provides standard error in functionality.
▭	**error out** contains error information. This output provides standard error out functionality.

Open VI

Owning Palette: Analog Output N Samples VIs

Requires: myRIO Toolkit

Opens a reference to one or more analog output channels, which you use to perform n samples write operations. You must open a reference before you write values to an analog output channel.



Allow multiple opens? specifies whether to allow opening the specified channels more than once. The default is FALSE. You must set **Allow multiple opens?** to TRUE if you also use the specified channels in an Express VI.

Channel Names specifies the names of the analog output channels to open a reference.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

AO (N Samples) Ref Out returns a reference to the analog output channels that you specify.

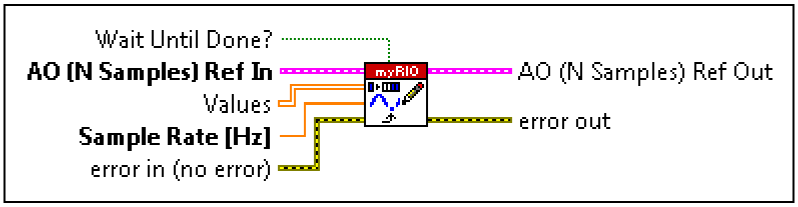
error out contains error information. This output provides standard error out functionality.

Write VI

Owning Palette: Analog Output N Samples VIs

Requires: myRIO Toolkit

Writes values to one or more analog output channels. Use this VI to write multiple samples for each channel at one time.



Wait Until Done? specifies whether to wait until the write operation completes. The default is FALSE. If you set **Wait Until Done?** to TRUE, this VI does not wait until the write operation completes.

AO (N Samples) Ref In specifies the reference to the analog output channels to which to write values. Use the Open VI to open a reference to the analog output channels. Do not modify the **AO (N Samples) Ref In** values.

Values specifies the values in volts to write to the analog output channels. **Values** is a 2D array. The number of elements in each row represents the number of samples to write to each analog output channel. Ensure this number is greater than 0 and less than or equal to 10,000. The order of the values corresponds to the order in which the Open VI opens the analog output channels. You must assign a value for each channel that you specify.

Sample Rate specifies the sampling frequency in hertz of the output signal. Valid values are between 1 k and 80 k.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

AO (N Samples) Ref Out returns the reference to the analog output channels to which this VI writes values.

error out contains error information. This output provides standard error out functionality.

Audio Input N Samples VIs

Owning Palette: Low Level VIs

Requires: myRIO Toolkit. This topic might not match its corresponding palette in LabVIEW depending on your operating system, licensed product(s), and target.

Use the Analog Input VIs to perform audio input operations on the myRIO, such as reading multiple samples for each audio input channel at one time.

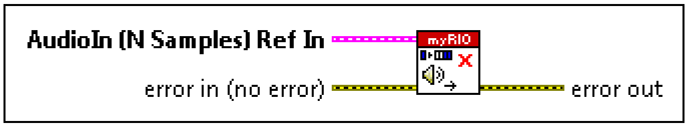
Palette Object	Description
Close	Closes the reference to one or more audio input channels, which you use to perform n samples read operations.
Open	Opens a reference to one or more audio input channels, which you use to perform n samples read operations. You must open a reference before you read values from an audio input channel.
Read	Reads values from one or more audio input channels. Use this VI to read multiple samples for each channel at one time.

Close VI

Owning Palette: Audio Input N Samples VIs

Requires: myRIO Toolkit

Closes the reference to one or more audio input channels, which you use to perform n samples read operations.



AudioIn (N Samples) Ref In specifies the reference to the audio input channels from which to read values. Use the Open VI to open a reference to the audio input channels. Do not modify the **AudioIn (N Samples) Ref In** values.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

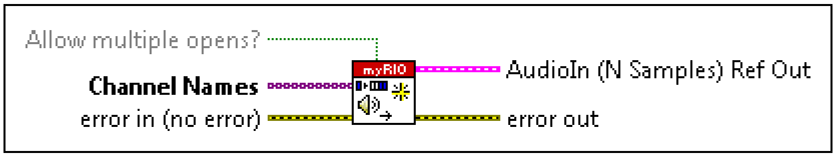
error out contains error information. This output provides standard error out functionality.

Open VI

Owning Palette: Audio Input N Samples VIs

Requires: myRIO Toolkit

Opens a reference to one or more audio input channels, which you use to perform n samples read operations. You must open a reference before you read values from an audio input channel.



Allow multiple opens? specifies whether to allow opening the specified channels more than once. The default is FALSE. You must set **Allow multiple opens?** to TRUE if you also use the specified channels in an Express VI.

Channel Names specifies the names of the audio input channels whose reference you want to open.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

AudioIn (N Samples) Ref Out returns a reference to the audio input channels that you specify.

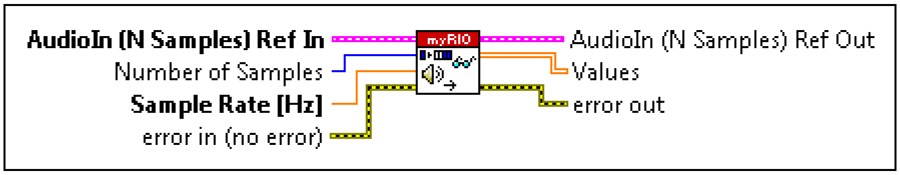
error out contains error information. This output provides standard error out functionality.

Read VI

Owning Palette: Audio Input N Samples VIs

Requires: myRIO Toolkit

Reads values from one or more audio input channels. Use this VI to read multiple samples for each channel at one time.



AudioIn (N Samples) Ref In specifies the reference to the audio input channels from which to read values. Use the Open VI to open a reference to the audio input channels. Do not modify the **AudioIn (N Samples) Ref In** values.

U32 **Number of Samples** specifies the number of samples to read. The default is 1,000. Valid values are between 0 and 10,000.

DBL **Sample Rate** specifies the sampling frequency in hertz of the input signal. The default is 1,000. Valid values are between 1 k and 30 k.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

AudioIn (N Samples) Ref Out returns the reference to the audio input channels from which this VI reads values.

[DBL] **Values** returns the values in volts that this VI reads from the audio input channels. The order of the values corresponds to the order in which the Open VI opens the audio input channels.

error out contains error information. This output provides standard error out functionality.

Audio Output N Samples VIs

Owning Palette: Low Level VIs

Requires: myRIO Toolkit. This topic might not match its corresponding palette in LabVIEW depending on your operating system, licensed product(s), and target.

Use the Audio Output N Samples VIs to perform audio output operations on the myRIO, such as writing multiple samples for each audio output channel at one time.

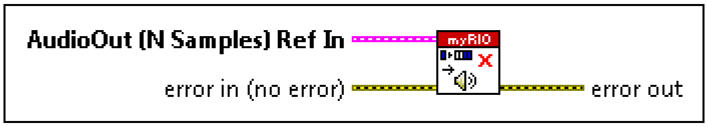
Palette Object	Description
Close	Closes the reference to one or more audio output channels, which you use to perform n samples write operations.
Open	Opens a reference to one or more audio output channels, which you use to perform n samples write operations. You must open a reference before you write values to an audio output channel.
Write	Writes values to one or more audio output channels. Use this VI to write multiple samples for each channel at one time.

Close VI

Owning Palette: Audio Output N Samples VIs

Requires: myRIO Toolkit

Closes the reference to one or more audio output channels, which you use to perform n samples write operations.



AudioOut (N Samples) Ref In specifies the reference to the audio output channels to which to write values. Use the Open VI to open a reference to the audio output channels. Do not modify the **AudioOut (N Samples) Ref In** values.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

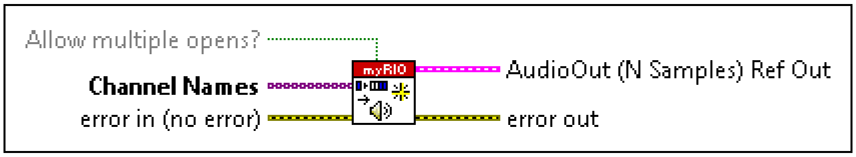
error out contains error information. This output provides standard error out functionality.

Open VI

Owning Palette: Audio Output N Samples VIs

Requires: myRIO Toolkit

Opens a reference to one or more audio output channels, which you use to perform n samples write operations. You must open a reference before you write values to an audio output channel.



Allow multiple opens? specifies whether to allow opening the specified channels more than once. The default is FALSE. You must set **Allow multiple opens?** to TRUE if you also use the specified channels in an Express VI.

Channel Names specifies the names of the audio output channels to open a reference.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

AudioOut (N Samples) Ref Out returns a reference to the audio output channels that you specify.

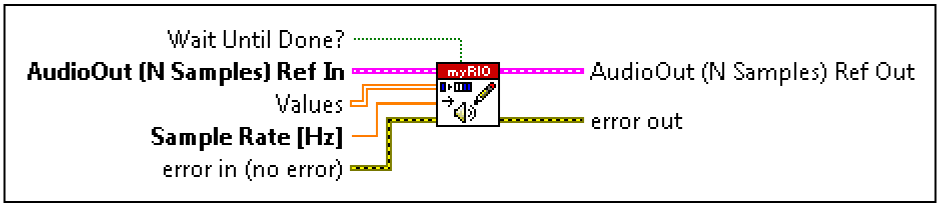
error out contains error information. This output provides standard error out functionality.

Write VI

Owning Palette: Audio Output N Samples VIs

Requires: myRIO Toolkit

Writes values to one or more audio output channels. Use this VI to write multiple samples for each channel at one time.



Wait Until Done? specifies whether to wait until the write operation completes. The default is FALSE. If you set **Wait Until Done?** to TRUE, this VI does not wait until the write operation completes.

AudioOut (N Samples) Ref In specifies the reference to the audio output channels to which to write values. Use the Open VI to open a reference to the audio output channels. Do not modify the **AudioOut (N Samples) Ref In** values.

Values specifies the values in volts to write to the audio output channels. **Values** is a 2D array. The number of elements in each row represents the number of samples to write to each audio output channel. Ensure this number is greater than 0 and less than or equal to 10,000. The order of the values corresponds to the order in which the Open VI opens the audio output channels. You must assign a value for each channel that you specify.

DBL **Sample Rate** specifies the sampling frequency in hertz of the signal. The default is 1,000. Valid values are between 1 k and 80 k.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

AudioOut (N Samples) Ref Out returns the reference to the audio output channels to which this VI writes values.

error out contains error information. This output provides standard error out functionality.

Digital Input/Output 1 Sample VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit. This topic might not match its corresponding palette in LabVIEW depending on your operating system, licensed product(s), and target.

Use the Digital Input/Output 1 Sample VIs to control the digital I/O channels on the myRIO or the roboRIO.

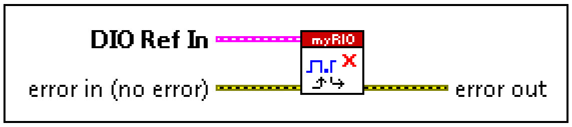
Palette Object	Description
Close	Closes the reference to one or more digital I/O channels, sets the logic levels of all output channels to low, and disables all output channels.
Open	Opens a reference to one or more digital I/O channels. You must open a reference before you read or write values.
Read	Reads the logic levels of one or more digital I/O channels.
Write	Writes logic levels to one or more digital I/O channels.

Close VI

Owning Palette: Digital Input/Output 1 Sample VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Closes the reference to one or more digital I/O channels, sets the logic levels of all output channels to low, and disables all output channels.



DIO Ref In specifies the reference to the digital I/O channels. Use the Open VI to open a reference to the digital I/O channels. Do not modify the **DIO Ref In** values.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

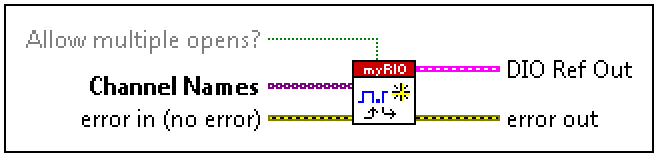
error out contains error information. This output provides standard error out functionality.

Open VI

Owning Palette: Digital Input/Output 1 Sample VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Opens a reference to one or more digital I/O channels. You must open a reference before you read or write values.



Allow multiple opens? specifies whether to allow opening the specified channels more than once. The default is FALSE. You must set **Allow multiple opens?** to TRUE if you also use the specified channels in an Express VI.

Channel Names specifies the names of the digital I/O channels whose reference you want to open.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

DIO Ref Out returns a reference to the digital I/O channels that you specify.

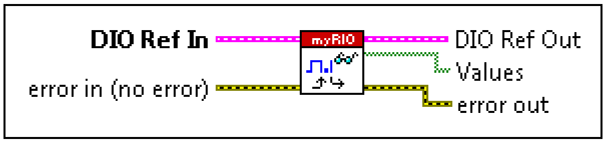
error out contains error information. This output provides standard error out functionality.

Read VI

Owning Palette: Digital Input/Output 1 Sample VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Reads the logic levels of one or more digital I/O channels.



DIO Ref In specifies the reference to the digital I/O channels from which to read logic levels. Use the Open VI to open a reference to the digital I/O channels. Do not modify the **DIO Ref In** values.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

DIO Ref Out returns the reference to the digital I/O channels from which this VI reads logic levels.

[TF] **Values** returns the logic levels that this VI reads from the digital I/O channels. **Values** returns TRUE if the logic level is a high voltage and returns FALSE if the logic level is a low voltage. The direction of a channel changes to input before this VI reads the logic level. The order of the elements in **Values** corresponds to the order in which the Open VI opens the digital I/O channels.

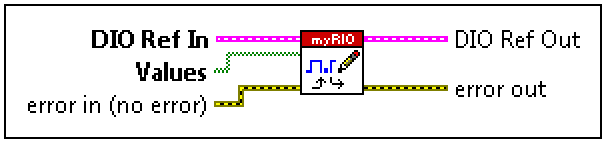
error out contains error information. This output provides standard error out functionality.

Write VI

Owning Palette: Digital Input/Output 1 Sample VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit

Writes logic levels to one or more digital I/O channels.



DIO Ref In specifies the reference to the digital I/O channels to which to write logic levels. Use the Open VI to open a reference to the digital I/O channels. Do not modify the **DIO Ref In** values.

[TF] **Values** specifies the logic levels to write to the digital I/O channels. Set **Values** to TRUE to write a high voltage and set **Values** to FALSE to write a low voltage. The direction of a channel changes to output before this VI writes the logic level. The order of the elements in **Values** corresponds to the order in which the Open VI opens the digital I/O channels. You must specify a value for each channel that you open.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

DIO Ref Out returns the reference to the digital I/O channels to which this VI writes logic levels.

error out contains error information. This output provides standard error out functionality.

Digital Input N Samples VIs

Owning Palette: Low Level VIs

Requires: myRIO Toolkit. This topic might not match its corresponding palette in LabVIEW depending on your operating system, licensed product(s), and target.

Use the Digital Input N Samples VIs to perform digital input operations on the myRIO, such as reading multiple samples for each digital input channel at one time.

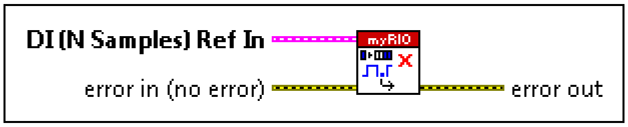
Palette Object	Description
Close	Closes the reference to one or more digital input channels, sets the logic levels of all digital input channels to low, and disables all digital input channels.
Open	Opens a reference to one or more digital input channels, which you use to perform n samples write operations. You must open a reference before you read values from an digital input channel.
Read	Reads values from one or more digital input channels. Use this VI to read multiple samples for each channel at one time.

Close VI

Owning Palette: Digital Input N Samples VIs

Requires: myRIO Toolkit

Closes the reference to one or more digital input channels, sets the logic levels of all digital input channels to low, and disables all digital input channels.



DI (N Samples) Ref In specifies the reference to the digital input channels from which to read values. Use the Open VI to open a reference to the digital input channels. Do not modify the **DI (N Samples) Ref In** values.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

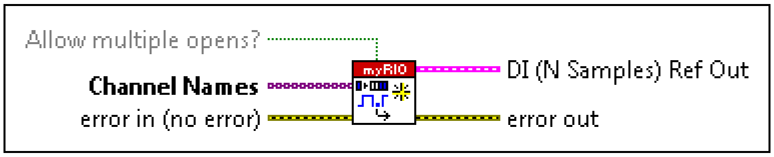
error out contains error information. This output provides standard error out functionality.

Open VI

Owning Palette: Digital Input N Samples VIs

Requires: myRIO Toolkit

Opens a reference to one or more digital input channels, which you use to perform n samples write operations. You must open a reference before you read values from an digital input channel.



Allow multiple opens? specifies whether to allow opening the specified channels more than once. The default is FALSE. You must set **Allow multiple opens?** to TRUE if you also use the specified channels in an Express VI.

[I/O] **Channel Names** specifies the names of the digital input channels to open a reference.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

DI (N Samples) Ref Out returns a reference to the digital input channels that you specify.

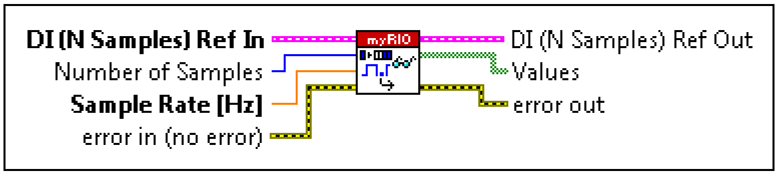
error out contains error information. This output provides standard error out functionality.

Read VI

Owning Palette: Digital Input N Samples VIs

Requires: myRIO Toolkit

Reads values from one or more digital input channels. Use this VI to read multiple samples for each channel at one time.



DI (N Samples) Ref In specifies the reference to the digital input channels from which to read values. Use the Open VI to open a reference to the digital input channels. Do not modify the **DI (N Samples) Ref In** values.

U32 **Number of Samples** specifies the number of samples to read. The default is 1,000. Valid values are between 0 and 10,000.

DBL **Sample Rate** specifies the sampling frequency in hertz of the input signal. The default is 1,000. Valid values are between 1 k and 1 M.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

DI (N Samples) Ref Out returns the reference to the digital input channels from which this VI reads values.

[TF] **Values** returns the values that this VI reads from the digital input channels. **Values** returns TRUE if the logic level is a high voltage and returns FALSE if the logic level is a low voltage. The direction of a channel changes to input before this VI reads the logic level. The order of the values corresponds to the order in which the Open VI opens the digital input channels.

error out contains error information. This output provides standard error out functionality.

Digital Output N Samples VIs

Owning Palette: Low Level VIs

Requires: myRIO Toolkit. This topic might not match its corresponding palette in LabVIEW depending on your operating system, licensed product(s), and target.

Use the Digital Output N Samples VIs to perform digital output operations on the myRIO, such as writing multiple samples for each digital output channel at one time.

Palette Object	Description
Close	Closes the reference to one or more digital output channels, sets the logic levels of the digital output channels to low, and disables digital output channels.
Open	Opens a reference to one or more digital output channels, which you use to perform n samples write operations. You must open a reference before you write values to an digital output channel.

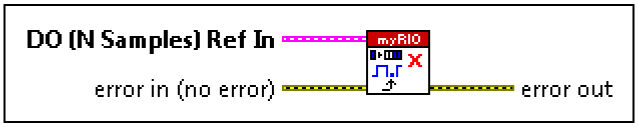
Write	Writes values to one or more digital output channels. Use this VI to write multiple samples for each channel at one time.

Close VI

Owning Palette: Digital Output N Samples VIs

Requires: myRIO Toolkit

Closes the reference to one or more digital output channels, sets the logic levels of the digital output channels to low, and disables digital output channels.



DO (N Samples) Ref In specifies the reference to the digital output channels to which to write values. Use the Open VI to open a reference to the digital output channels. Do not modify the **DO (N Samples) Ref In** values.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

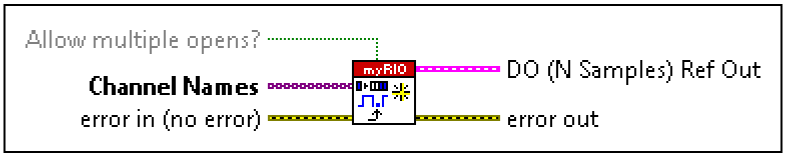
error out contains error information. This output provides standard error out functionality.

Open VI

Owning Palette: Digital Output N Samples VIs

Requires: myRIO Toolkit

Opens a reference to one or more digital output channels, which you use to perform n samples write operations. You must open a reference before you write values to an digital output channel.



TF

Allow multiple opens? specifies whether to allow opening the specified channels more than once. The default is FALSE. You must set **Allow multiple opens?** to TRUE if you also use the specified channels in an Express VI.

[I/O]

Channel Names specifies the names of the digital output channels whose reference you want to open.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

DO (N Samples) Ref Out returns a reference to the digital output channels that you specify.

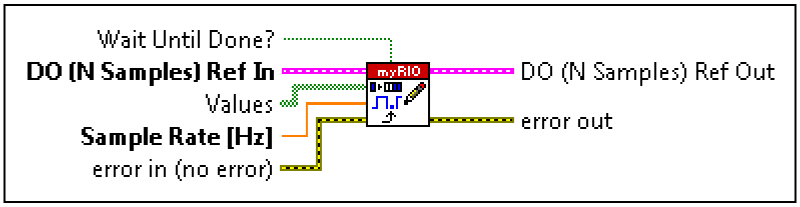
error out contains error information. This output provides standard error out functionality.

Write VI

Owning Palette: Digital Output N Samples VIs

Requires: myRIO Toolkit

Writes values to one or more digital output channels. Use this VI to write multiple samples for each channel at one time.



TF

Wait Until Done? specifies whether to wait until the write operation completes. The default is FALSE. If you set **Wait Until Done?** to TRUE, this VI does not wait until the write operation completes.

DO (N Samples) Ref In specifies the reference to the digital output channels to which to write values. Use the Open VI to open a reference to the digital output channels. Do not modify the **DO (N Samples) Ref In** values.

Values specifies the logic levels to write to the digital output channels. Set **Values** to TRUE to write a high voltage and set **Values** to FALSE to write a low voltage. **Values** is a 2D array. The number of elements in each row represents the number of samples to write to each digital output channel. Ensure this number is greater than 0 and less than or equal to 10,000. The direction of a channel changes to output before this VI writes the logic level. The order of the elements in **Values** corresponds to the order in which the Open VI opens the digital I/O channels. You must specify a value for each channel that you open.

Sample Rate specifies the sampling frequency in hertz of the output signal. The default is 1,000. Valid values are between 1 K and 1 M.

error in describes error conditions that occur before this node runs. This input provides standard error in functionality.

DO (N Samples) Ref Out returns the reference to the digital output channels to which this VI writes values.

error out contains error information. This output provides standard error out functionality.

Input Device Control VIs

Requires: myRIO Toolkit **or** roboRIO Toolkit. This topic might not match its corresponding palette in LabVIEW depending on your operating system, licensed product(s), and target.

Use the Input Device Control VIs to initialize, query, close, and get the state information about the joystick connected to the myRIO or the roboRIO.

Palette Object	Description
Acquire Input Data	Returns data about the joystick connected to th e computer.
Close Input Device	Closes the joystick you specify in **device ID**.
Initialize Joystick	Opens a reference to and initializes a joystick de vice at the index you specify.
Query Input Devices	Obtains information about the joystick connect ed to the computer.

Refer to the `Joystick Monitoring.lvproj` in the `labview\examples\myRIO\Joystick Monitoring` directory for an example of using the Input Device Control VIs with the myRIO.

Refer to the `Joystick Monitoring.lvproj` in the `labview\examples\roboRIO\Joystick Monitoring` directory for an example of using the Input Device Control VIs with the roboRIO.