

Setting Up EtherCAT on NI Programmable Automation Controllers

Publish Date: Feb 04, 2014

Overview

This tutorial provides step-by-step instructions to help you set up your EtherCAT system using NI programmable automation controllers (PACs) and NI LabVIEW graphical programming. Follow these guidelines to go from unpacking your NI hardware to programming I/O in LabVIEW.

Table of Contents

- 1. Requirements
- 2. Setting Up the Hardware
- 3. Configuring the Master Controller
- 4. Configuring the Network in LabVIEW
- 5. Programming the NI 9144 FPGA in LabVIEW FPGA
- 6. Related Links

1. Requirements

Hardware

- Master: LabVIEW Real-Time controller with two Ethernet ports (<http://www.ni.com/white-paper/10595/en>), such as CompactRIO, PXI, and NI industrial controller
- Slave: NI 9144 expansion chassis with NI C Series I/O modules
- Ethernet cable and power supplies

Software

Software	Version for Programming in LabVIEW Real-Time Only	Version for Programming in LabVIEW Real-Time and LabVIEW FPGA
LabVIEW (http://ni.com/labview)	8.6 or later	2009 or later
LabVIEW Real-Time Module (http://ni.com/realtime)	8.6 or later	2009 or later
LabVIEW FPGA Module (http://ni.com/fpga)	–	2009 or later
NI-RIO driver software (http://joule.ni.com/nidu/cds/fn/p/q/ni-rio)	3.0.1 or later	3.2 or later
NI-Industrial Communications for EtherCAT driver software (http://joule.ni.com/nidu/cds/fn/p/q/ethercat)	1.0 or later	1.1 or later

2. Setting Up the Hardware

This tutorial uses an NI cRIO-9074 PAC as the EtherCAT master. To set up this hardware system, follow these steps:

- 1. Insert the C Series I/O modules in the appropriate chassis (the CompactRIO master chassis and/or the NI 9144 slave)
- 2. Connect a standard Category 5 or better Ethernet cable from cRIO-9074 Ethernet Port 2 to NI 9144 Ethernet Port 1. You can daisy chain multiple NI 9144 chassis from the controller in this manner.

Note: Port 1 on the cRIO-9074 is on the bottom. Port 1 on the NI 9144 is on the top.

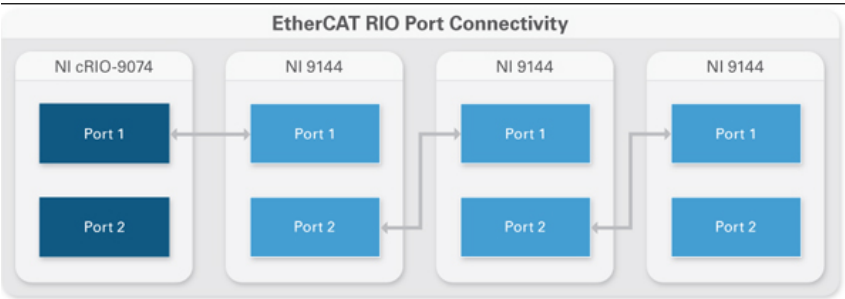


Figure 1. Connecting the Ethernet Ports on the CompactRIO Master Controller and NI 9144 Slave Chassis

- 1. Wire an external power supply to each chassis, which then powers on the hardware.
- 3. Configuring the Master Controller
 - 1. After you have connected the hardware, install the required software on the host computer.
 - 2. Use an Ethernet cable to connect cRIO-9074 Ethernet Port 1 to the same network as your host PC.

3. Launch the NI Measurement & Automation Explorer (MAX) configuration utility from **Start»Programs»National Instruments»Measurement & Automation**. Double-click on **Remote Systems** in the Configuration panel tree on the autodiscover the cRIO-9074 on the network, and click on the CompactRIO controller.
4. Change the name of the master controller and select **Obtain IP address from DHCP server** under **IP Settings**. Click when you are done. (To use a static IP address, see the related links below.)

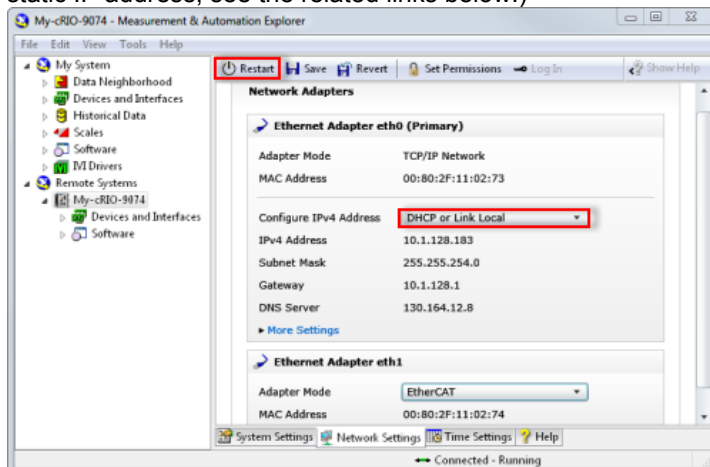


Figure 2. Setting the IP Address for the Real-Time Controller

1. To install the software on the real-time master controller, expand the cRIO-9074 under **Remote Systems**. Right-click and select **Add/Remove Software**.

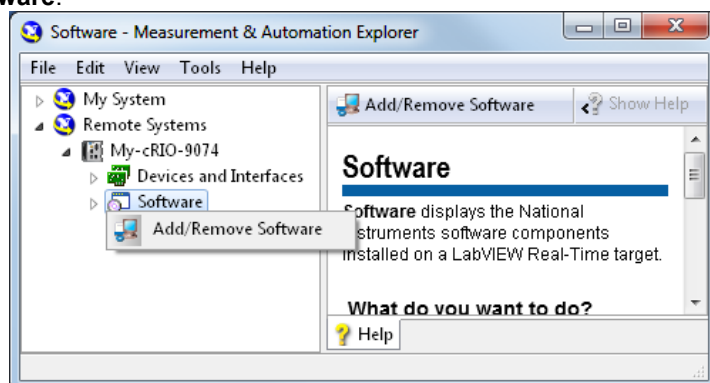


Figure 3. Installing Software on the Real-Time Controller

1. Install the recommended software set—NI-RIO driver software with the NI Scan Engine as well as NI-Industrial Communications for EtherCAT driver software—onto the CompactRIO controller.
2. Once the controller has rebooted, find the CompactRIO controller under **Remote Systems** again. Click on **Advanced Settings** in the lower right-hand corner.
3. Select cRIO-9074 Ethernet Port 2 (the MAC address that is not primary). Then select **EtherCAT** in the pull-down box **Mode** and click **OK**.

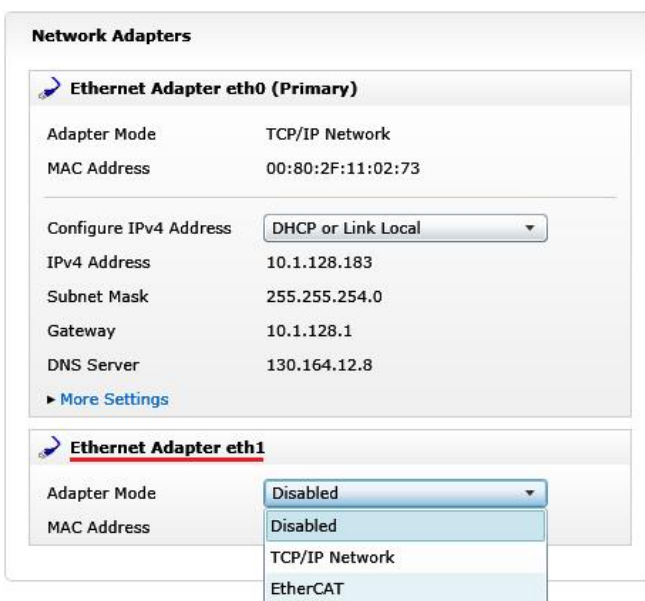


Figure 4. Selecting EtherCAT Mode for the Ethernet Port of the Real-Time Controller

Note: When the controller's Ethernet Port 2 is in EtherCAT mode, you cannot use that port on an Ethernet network.

4. Configuring the Network in LabVIEW

1. Launch LabVIEW from **Start»Programs»National Instruments»LabVIEW X.X»LabVIEW**. Click on **Empty Project**.
2. Right-click on **Project** and select **New»Targets and Devices**.
3. In the **Add Targets and Devices** dialog window, select **Existing target or device** and expand the category **Real-Time CompactRIO** to autodiscover the EtherCAT master on the host PC's subnet. Select the cRIO-9074 and click **OK**.
4. In the LabVIEW Project window, right-click on the cRIO-9074 and select **New»Targets and Devices**.

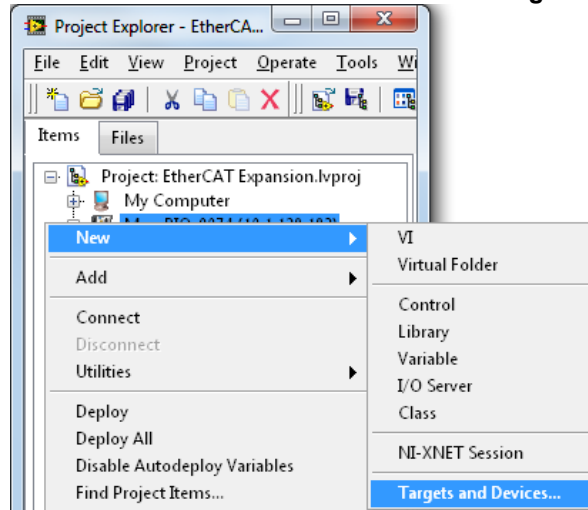


Figure 5. Using LabVIEW to Discover Controllers Connected to the Network

1. In the **Add Targets and Devices** dialog window, select **Existing target or device** and expand the category **EtherCAT Device** to autodiscover the EtherCAT port on the controller.

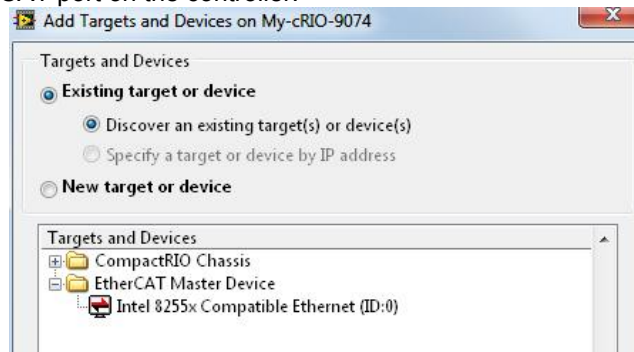


Figure 6. Selecting the EtherCAT Master Port on the Real-Time Controller

1. Select the available EtherCAT master port and click **OK**. The Scan Slaves dialog window appears. The first option Scan Interface autodiscovers any slaves connected to the controller. Click **OK**. The LabVIEW Project now lists each slave cRIO I/O modules, and the physical I/O on each module (called I/O variables).
2. Once all the EtherCAT slaves have been autodiscovered, you may create a VI, or logic program, on the CompactRIO controller. To program the NI 9144 modular slave in LabVIEW Real-Time, click and drag the I/O variables from the Lab Project to the block diagram. With these I/O variables, you can read and write from physical channels in the NI 9144 chassis. When finished, click the **Run button** on the LabVIEW VI program to deploy the program to the real-time controller.

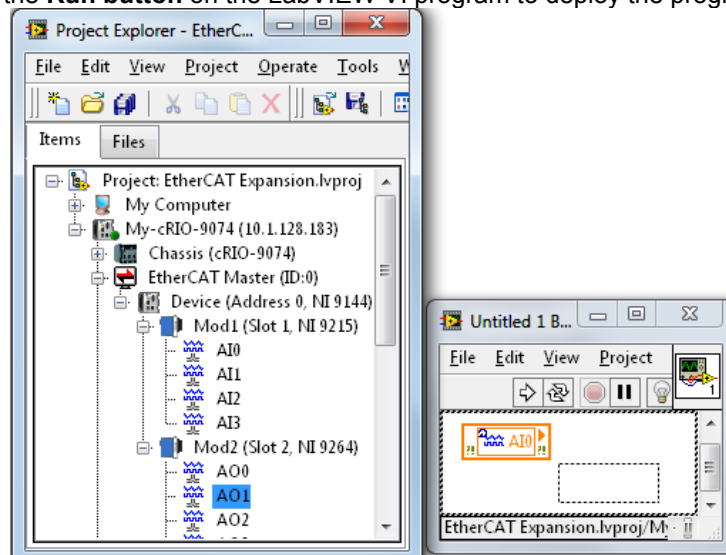


Figure 7. Dragging I/O Variables From the LabVIEW Project to the Block Diagram

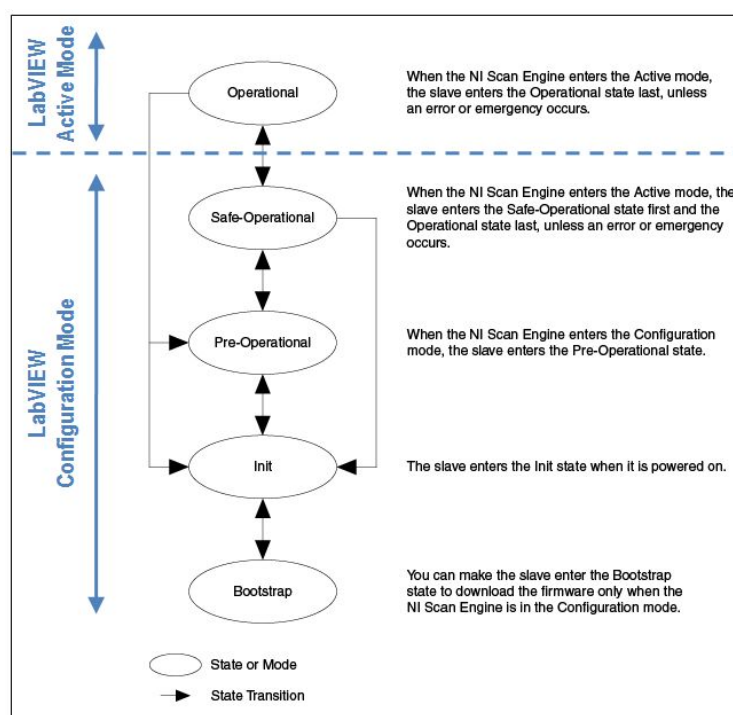
Note: If you use the I/O variables in a Timed Loop, you can synchronize the loop timing source to the NI Scan Engine. When you run the VI, the program automatically deploys onto the CompactRIO master. For more information about the LabVIEW I/O and CompactRIO Scan Mode, see the white paper titled Introduction to CompactRIO Scan Mode (<http://zone.ni.com/devzone/cda/tut/p/id/7338>).

Changing EtherCAT Modes in LabVIEW

LabVIEW streamlines the different EtherCAT states (that is, Operational, Init, Bootstrap, and so on) by grouping them into LabVIEW Scan Engine modes: Configuration Mode and Active Mode. In Active Mode, the LabVIEW program gets real data from the network. In Configuration Mode, the network data is not updated and configuration of the EtherCAT network is enabled, allowing you to add new slave devices and modules and download the NI 9144 EtherCAT slave firmware.

To change the LabVIEW Scan Engine mode, right-click the cRIO-9074 master controller in the LabVIEW project explorer and select **Utilities»Scan Engine Mode»Switch to Active** (or **Switch to Configuration**). The LabVIEW Scan Engine mode can be changed programmatically with the **Set Scan Engine Mode** function, accessed by right-clicking on the LabVIEW bloc and selecting the **Measurement I/O»NI Scan Engine** palette.

You can change an individual slave's EtherCAT state by right-clicking on the slave device in the LabVIEW Project window and selecting **Online Device State**. The slave device's EtherCAT state can be changed almost completely independently from LabVIEW Scan Engine mode, with only two restrictions. One restriction is that you cannot place a slave device into Operational or Safe-Operational state unless the LabVIEW Scan Engine is in an equivalent state (Active Mode). The other restriction is that you cannot place your device into Bootstrap state unless the LabVIEW Scan Engine is in Configuration Mode.



[+] Enlarge Image

Figure 8. How LabVIEW Scan Engine Modes Correspond With EtherCAT States

Adding Third-Party EtherCAT Slaves to NI Master Controllers

To have a third-party slave device show up in the LabVIEW Project window, you must import the slave's device profile XML file before configuring the slave with LabVIEW.

1. Locate the EtherCAT XML Device Description File for your third-party slave. Some companies provide these XML files for download on their websites.
2. Right-click the **EtherCAT master** item and select **Utilities»Import Device Profiles** to display the **Import Device Profiles** dialog box.
3. Click the file **Browse** button to select the device profile(s) in either of the following ways:
 - Select a file and click **Open** to import a device profile.
 - Open a folder and click **Current Folder** to import all device profiles under the folder.
4. After selecting the device profiles, you can see the file import results. The file is imported successfully if Import Result Succeed. Otherwise, the file is not imported due to an invalid format or other reason.
5. (Optional) Repeat step 2 to import other device profiles.
6. Click Close to exit the dialog.
7. You must restart LabVIEW to use the newly imported device profile.

5. Programming the NI 9144 FPGA in LabVIEW FPGA

The program written in LabVIEW Real-Time is deployed to the EtherCAT master, which is responsible for collecting, processing and controlling the I/O in the EtherCAT network. Additionally, each NI 9144 slave chassis contains an embedded field-programmable gate array (FPGA) that can execute custom timing and signal processing to help you create intelligent distributed devices that are synchronized within 100 ns of each other. To take advantage of the programmable FPGA capability you must use LabVIEW, LabVIEW Real-Time, and LabVIEW FPGA 2009 or later.

1. When you add the EtherCAT Master Device to the LabVIEW Project (see step 5 in the previous section), you can either add the NI 9144 slave to program in the NI Scan Interface or LabVIEW FPGA Interface. If you select the former, you can select the program in LabVIEW FPGA by right-clicking on **Device (Address 0, NI 9144)** in the LabVIEW Project window and selecting **New»FPGA Target**.

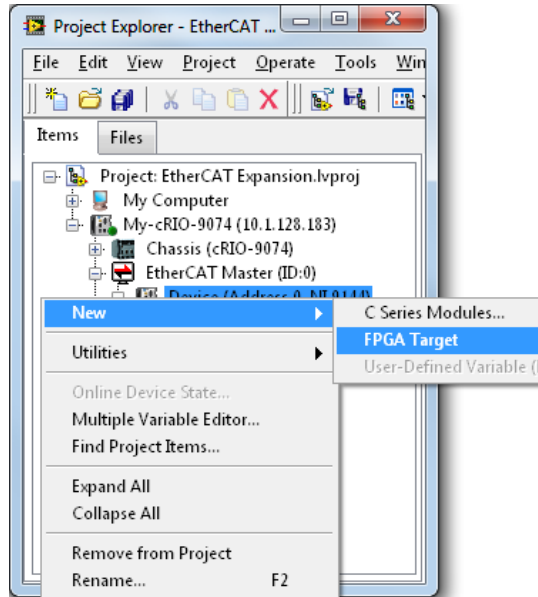


Figure 9. Discovering the FPGA Target on the NI 9144

1. In the LabVIEW Project, the NI 9144 slave device features several FPGA signals. For example, Input Virtual Point and Output Virtual Point are digital timing signals used to synchronize FPGA code between different NI 9144 chassis. To program the module in LabVIEW FPGA Mode, drag and drop the module from **Device (Address 0, NI 9144)** onto the **FPGA Target (NI 9144)**. In the Project Explorer window, you can drag and drop modules between **Device** and the **FPGA Target** to switch between Scan Mode and FPGA Mode.

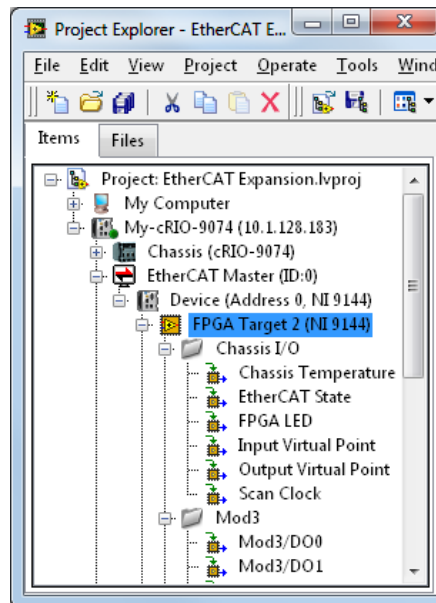


Figure 10. Chassis I/O and Module I/O Listed Under the FPGA Target

1. In the LabVIEW Project Explorer window, right-click on **FPGA Target (NI 9144)** in the LabVIEW Project and select **New»FPGA Target** to create a new FPGA program on the NI 9144. The FPGA I/O channels from modules listed under **FPGA Target (NI 9144)** can be dragged and dropped into the block diagram of this new FPGA VI.
2. To deploy it to the **FPGA Target (NI 9144)**, the VI must be compiled through creating a new **Build Specification»New»Compilation**. After the FPGA VI has compiled successfully, it will be listed under Build Specifications in the Project Explorer and can be downloaded to the **FPGA Target (NI 9144)** by right-clicking on it and selecting **Down**.

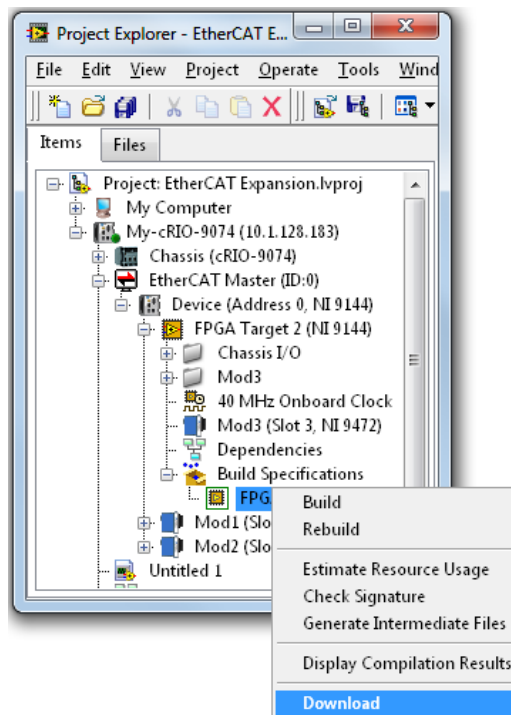


Figure 11. FPGA VI Compiled for Downloading to the NI 9144 FPGA

Differences in the LabVIEW FPGA Programming API

If you are familiar with programming in LabVIEW FPGA for CompactRIO and other NI reconfigurable I/O (RIO) platforms, some differences between programming the master's local FPGA and the slave's FPGA. Since LabVIEW 2009, **user-defined variables** are used to synchronize FPGA data with the Scan Engine. These user-defined I/O variables are also the only method for transferring data between the controller's real-time VI and the NI 9144 expansion chassis' FPGA VI.

FPGA Transfer Methods	Local Chassis	Expansion Chassis
User-defined I/O variables	√	√
FPGA Host Interface	√	X
DMA transfer functions	√	X
FPGA front panel debugging	√	X

Table 1. FPGA Transfer Methods for Local Versus Expansion Chassis

1. Program the FPGA VI by clicking the NI 9144 FPGA I/O nodes from the LabVIEW Project and dragging them onto the block diagram. In this example, a proportional integral derivative (PID) function was used to control analog I/O in the NI 9144 chassis.

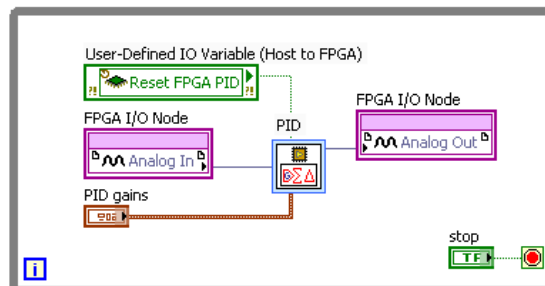


Figure 12. Example FPGA Code on the NI 9144

1. Note the Reset FPGA PID Boolean variable. This is a user-defined I/O variable that allows the real-time VI on the NI master to communicate with the FPGA VI on the NI 9144. To create such a variable, right-click on **Device (Address 0, NI 9144)** in the LabVIEW Project and select **New>User-Defined Variable**. The variable name, data type, and data direction (Host to FPGA or vice versa) may be set from the Properties window. In this case, the master controller is sending info to the NI 9144 slave, so the direction is **Host to FPGA**.

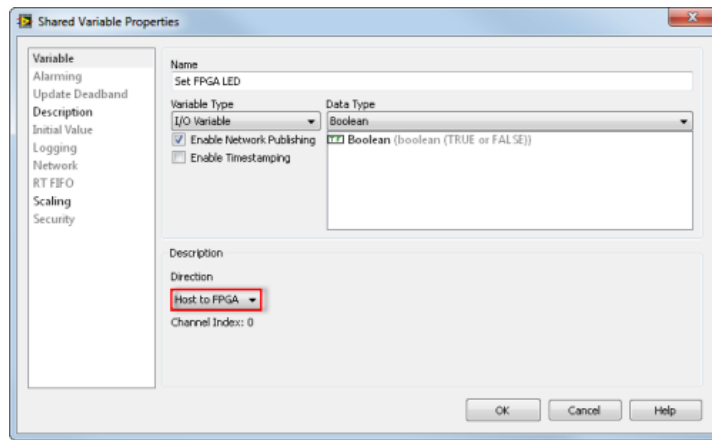


Figure 13. Dialogue Window for Creating a User-Defined I/O Variable

You can now program the user-defined I/O variable in the real-time VI used in this example to reset the PID function runn FPGA VI.

1. To run the FPGA VI, right-click on cRIO-9074 in the LabVIEW Project and select **Deploy All** to deploy the NI 9144 slave device. Put the LabVIEW Scan Engine into Configuration Mode by right-clicking the CompactRIO controller and select **Utilities»Scan Engine Mode»Switch to Configuration** and click the Run button on the FPGA VI. This starts the compile process and downloads the FPGA bitfile to the NI 9144 chassis. After you have successfully downloaded the bitfile to 9144, the FPGA VI begins executing immediately.

Note: You may need to update the firmware on your NI 9144 slave chassis to run custom FPGA code. If you get an error stating that your firmware needs to be updated, follow the instructions in the NI 9144 User Guide (<http://digital.ni.com/manuals.nsf/websearch/863B513DBBACDF54862575F2006FFC75>) under the Updating Your Firmware section.

1. When the compile and download process completes, you need to set the LabVIEW Scan Engine back into Active Mode. To do so, right-click the CompactRIO master controller in the LabVIEW Project window and select **Utilities»Scan Engine Mode»Switch to Active**.

An important note is that there is a limit to the number of user-defined I/O variables that can be created in FPGA Mode. The NI 9144 slave chassis can hold a total of 512 bytes of input data and 512 bytes of output data for both I/O variables in Scan Mode and user-defined I/O variables in FPGA Mode. For example, if you are using four 32-channel modules in Scan Mode and each channel takes up 32 bits of data, then 256 bytes of input data are being used by Scan Mode I/O variables. With the remaining bytes of input data, you can create 64 input user-defined I/O variables (also of 32-bit length) in FPGA Mode.

6. Related Links

See applications, technical resources, and more for EtherCAT RIO (<http://www.ni.com/white-paper/14140/en>)

Browse EtherCAT RIO products (<http://sine.ni.com/nips/cds/view/p/lang/en/nid/210787>)

PRODUCT

Order status and history (<http://www.ni.com/status/>)

Order by part number (<http://sine.ni.com/apps/utf8/nios.store?action=purchase>)

Activate a product (<http://sine.ni.com/myproducts/app/main.xhtml?lang=en>)

Order and payment information (<http://www.ni.com/how-to-buy/>)

SUPPORT

Submit a service request (<https://sine.ni.com/srm/app/myServiceRequests>)

Manuals (<http://www.ni.com/manuals/>)

Drivers (<http://www.ni.com/downloads/drivers/>)

Alliance Partners (<http://www.ni.com/alliance/>)

COMPANY

About National Instruments (<http://www.ni.com/company/>)

Events (<http://www.ni.com/events/>)

Careers (<http://www.ni.com/careers/>)

MISSION

NI equips engineers and scientists with systems that accelerate productivity, innovation, and discovery.

(<http://twitter.com/niglobal>)

(<http://www.facebook.com/NationalInstruments>)

(<http://www.linkedin.com/company/3433?trk=tyah>)

(<http://www.ni.com/rss/>) (<http://www.youtube.com/nationalinstruments>)

Contact Us (<http://www.ni.com/contact-us/>)



(<http://privacy.truste.com/privacy-seal/National-Instruments-Corporation/validation?rid=bc6daa8f-7051-4eea-b7b5-fb24dcd96d95>)

Legal (<http://www.ni.com/legal/>) | © National Instruments. All rights reserved. | **Site map** (<http://www.ni.com/help/map.htm>)