

# CompactRIO Hands-On Tutorial- Basics II

Publish Date: Oct 29, 2007

## Overview

This hands-on session is an introduction to intermediate level concepts and methods of programming CompactRIO. You will create a cRIO application for embedded motor control, using methods of synchronization and data scaling. This application will also show the flexibility of the cRIO system, creating Pulse Width Modulation and Quad Encoder functionality in the FPGA. This session is an opportunity to work with a more advanced application and explore solutions to some common industrial challenges.

[Download the LabVIEW Example Code for this Hands-on Tutorial](#)

## Table of Contents

- 1. [Hands-On Summary](#)
- 2. [Demo Requirements](#)
- 3. [Exercise 1: Developing the FPGA Application](#)
- 4. [Exercise 2: Developing the Host VI for CompactRIO](#)
- 5. [Appendix-Hardware Setup and Software configuration](#)

### 1. Hands-On Summary

- 1. Review the CompactRIO Product Line.
  - 2. Create an FPGA VI to read and write to CompactRIO I/O and implement PID control operations.
- Create an RT VI to scale and display data, and to provide supervisory control over the FPGA control application

### 2. Demo Requirements

| Hardware  | Software  |
|---|---|
| <ul style="list-style-type: none"><li>NI FPGA 7831R, cRIO-9151 Expansion and SH68-C68-S Cable <b>OR</b></li></ul> | <ul style="list-style-type: none"><li>LabVIEW Full Dev System for Windows (Eng) w/1 Yr Standard Service (776670-09)</li></ul> |
| <ul style="list-style-type: none"><li>NI cRIO-9002 and cRIO-9102 Chassis</li></ul>                                | <ul style="list-style-type: none"><li>LabVIEW Reconfigurable I/O Software Development Kit (779149-03)</li></ul>               |
| <ul style="list-style-type: none"><li>NI cRIO-9211, 9263, 9411, 9474 (AI, AO, DI, DO)</li></ul>                   |   |
| <ul style="list-style-type: none"><li>Temperature Chamber: Light, Fan and Thermocouple</li></ul>                  |   |

Table 1. Software and Hardware Requirements

### 3. Exercise 1: Developing the FPGA Application

- 1. Launch **LabVIEW**.
- 2. In the **Getting Started** screen for **LabVIEW**, select **Empty Project** under the **New** section. The **Project Explorer** window appears.

**Figure 1.** Creating a New Project.

3. Select **File»Save Project** and enter **cRIO Thermostat** in the **Name the Project** text box after selecting the appropriate directory for the project.
4. In the **Project Explorer** window, **Right-click** on **Project: cRIO Thermostat.lvproj** and select **New»Targets and Devices...** The **Add Targets and Devices** window will appear.

**Figure 2.** Selecting a Target for the Project.

5. While **Existing target or device** is selected, select the appropriate type of device.

*PXI R Series Embedded Expansion System*

- Select **Real-Time PXI** and then the appropriate system that is listed on the network.
- The target will show up in the project on the same hierarchy level as **My Computer** with the name **<Target's Name> (IP Address)**, where **<Target's Name>** is the actual name of the device that is given to it and **(IP Address)** is the IP Address of the particular device.
- **Right-click** on the target's name and select **New»Targets and Devices...** The **Add Targets and Devices** window will appear.

**Figure 3.** Selecting FPGA Target for PXI System.

- Select the **+** next to **FPGA Target** and the R Series board should appear.

**Figure 4.** Selecting R Series Board for PXI System.

- In the **Project Explorer** window, **Right-click** on the newly added **FPGA Target** and select **New»R Series Expansion Chassis**. Specify the **Name** of the chassis and the proper **Location** of the **Connector** to the R Series board for the chassis. Ensure that the **Discover C Series Modules** option is selected and select **OK**.

**Figure 5.** Adding Expansion Chassis to Project.

- Ensure that all of the C Series Modules appear to be in the appropriate slots in the **Project Explorer** window when compared to the actual chassis. The **Project Explorer** window should be configured for the chassis.

**Figure 6.** R Series Expansion Chassis Configured in Project.

*CompactRIO Embedded System*

- While the **Add Targets or Devices** window for adding to the project is open, select **Real-Time CompactRIO** and then the appropriate device listed on the network.
- After selecting **OK**, the C Series Modules will be detected and the CompactRIO should show up in the project on the same hierarchy level as **My Computer**.

**Figure 7.** cRIO Chassis Configured in Project.

***Develop an FPGA VI***

1. In the Project Explorer window, Right-click on the FPGA Target for your device and select New»VI.

## Beginning Format of the Application

1. CompactRIO FPGA VIs can use a common format, to ensure readability and completeness of the application. For basic synchronization of cRIO module I/O with RT, a three-frame flat sequence structure can be used as a template. Surround that flat sequence structure with a while loop.

**Figure 8.** Basic Format with Sequence Structure and While Loop.

2. In the first frame we will perform the timing and/or triggering functions. Place a Loop Timer VI, and configure it as a 32 bit Tick Timer.
3. In the third frame we will perform synchronization with the host, in this case by generating an interrupt for the controller. Create a control for the Wait Until Cleared terminal. This will allow us to run the FPGA VI either with or without a controller to enforce synchronization.
4. The second frame contains Control and Module I/O. In this case the Module I/O will be handled in separate loops which will run in parallel with this Communications and Control Loop. We'll add the Control VI later in the exercise.
5. Now we'll create the VI for interfacing to the cRIO-9411 Digital Input Module and performing Quad Encoder functions. This needs to run separate from the other loops, to allow the Quad Encoder inputs to run at the fastest possible speed. From the folder on the desktop, place the Encoder Position and Velocity (FPGA) VI on your diagram, outside of the existing loop. Right Click on the VI terminals and create controls and indicators.
6. Enclose the Encoder Position and Velocity (FPGA) VI and its controls and indicators in a new While Loop, with a constant value wired to the conditional terminal.

**Figure 9.** Encoder Position and Velocity Loop.

7. Open the Encoder Position and Velocity (FPGA) VI. Look at the Block Diagram. Note that there are several broken wires. These wires are broken because the Digital Input VI is not configured for proper input channels. **Left-click** on the **I/O node**, and configure the two channels for **Mod3/DI0** and **Mod3/DI1**. Save and close this VI.
8. We have another Module I/O function to place in an independent loop, to perform the PWM function through the cRIO-9474 Digital Output module. This PWM output will drive the motor directly. The cRIO-9474 can switch an output signal with a propagation delay of less than 1us, and can drive up to 1 Amp of current per channel. From the folder on the desktop, place the Pulse Width Modulation (FPGA) VI on your diagram, outside of the existing loop. Enclose the VI in a new While Loop, with a constant value wired to the conditional terminal.
9. Create a control for the PWM Period (Ticks) terminal, and leave the PWM High Time terminal unwired for now.

**Figure 10.** PWM Loop.

10. We have our Input Quad Encoder and Output PWM configured, so we will now return to the control frame of the Communications loop. From the palette, select the **PID (FPGA)** Express VI from Functions>>FPGA Math and Analysis>>Control. Select **OK** to close the dialog box. Create Controls and Indicators as shown in **Figure 11**.

**Figure 11.** PID setup.

11. Create a local variable of the Velocity (Counts/Interval) indicator and wire that to the process variable input of the PID. Create a local Variable from the PWM High Time (Ticks) indicator, and wire that to the input of the same name on the PWM VI. We now have input, output, and control in three separate parallel loops.

**Figure 12.** Parallel Loop Control.

12. Check your VI against the illustration below. If everything looks right, click the run arrow on the VI. This will begin the LV FPGA build process. Accept the default rate settings and begin compile. The process should take about 10 minutes.

Figure 13. Completed FPGA VI.

13. In order to continue work while the compile runs, click the Disconnect button on the Compile monitor. You can now continue LabVIEW development while the compile task runs.

**4. Exercise 2: Developing the Host VI for CompactRIO**

Use the host VI to communicate with the FPGA VI. You can run the host VI on an RT target or on a Windows PC. Complete the following steps to develop the host VI for the CompactRIO thermostat.

*R Series Expansion System*

1. In the **Project Explorer** window, **Right-click** on the PXI target and select **New»VI**.
- Reconfigurable Embedded System
1. In the **Project Explorer** window, **Right-click** on the cRIO chassis RT controller and select **New»VI**.
  2. **Right-click** on the block diagram and navigate to the **All Functions»FPGA Interface Palette**.
  3. Place the Open FPGA VI Reference function on the block diagram. Use this function to open a reference to the VI you downloaded to the FPGA device. Every host VI must use the Open FPGA VI Reference function.
  4. Right-click the function and select **Select VI...** from the shortcut menu. The **Select Target VI** dialog box appears.
  5. Browse to the **Motor Control (FPGA) VI** and click the **OK** button.
  6. Right-click the Open FPGA VI Reference function and select **Show Resource Input**. This creates an input terminal to programmatically specify what VISA resource to open the FPGA VI on.
  7. Right-click the **Resource Name** input of the Open FPGA Reference function and select **Create»Control** from the shortcut menu.

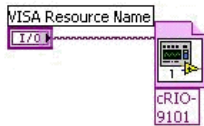


Figure 14. Open FPGA Reference.

8. Place a While Loop on the block diagram. Right-click the conditional terminal and select **Create»Control** from the shortcut menu.
9. Place the Close FPGA VI Reference function outside of the While Loop. Every host VI must use the Close FPGA VI Reference function.

**Figure 15.** Open and Close FPGA Reference.

10. Place two **Read/Write Control** functions inside the While Loop.
11. Wire the **FPGA VI Reference Out** output of the Open FPGA VI Reference function to the **FPGA VI Reference** input of the first Read/Write Control function.
12. Wire the **FPGA VI Reference Out** output of the first Read/Write Control function to the **FPGA VI Reference** input of the second Read/Write Control function.
13. Wire the **FPGA VI Reference Out** output of the second Read/Write Control function to the **FPGA VI Reference** input of the Close FPGA VI Reference function.



**Figure 16.** FPGA VI Reference Wired.

14. Wire the **error out** output of the Open FPGA VI Reference function to the **error in** input of the first Read/Write Control function.
15. Wire the **error out** output of the first Read/Write Control function to the **error in** input of the second Read/Write Control function.
16. Wire the **error out** output of the second Read/Write Control function to the **error in** input of the Close FPGA VI Reference function.

**Figure 17.** Error Terminals Wired.

17. Expand the first **Read/Write Control** to show four **Unselected** controls.

18. **Left-click** on the top **Unselected** control and choose **Position (Counts)**. Repeat this for the remaining **Unselected** controls and change the names to **Velocity (Counts/Interval)**, **PWM High Time (Ticks)**, and **Actual Loop Rate (Ticks)**, respectively.

**Figure 18.** First Read/Write Control Configured.

19. Expand the second **Read/Write Control** to show three **Unselected** controls.

20. **Left-click** on the top **Unselected** control and choose **Loop Time (Ticks)**. Repeat this for the remaining **Unselected** controls and change the names to **Velocity Set Point (Counts/Interval)** and **Velocity Time Interval (Ticks)**, respectively.

**Figure 19.** Second Read/Write Control Configured.

21. From the folder on the desktop, place the **Scale Inputs (RT Host) VI** on your diagram, inside of the existing loop. Also place the **Scale Outputs (RT Host) VI** on your diagram.

22. Connect the outputs of the first Read/Write Control function to the corresponding inputs of the **Scale Inputs (RT Host) VI**.

23. Connect the **Loop Time (Ticks)** output of the Scale Outputs (RT Host) VI to the **Loop Time (Ticks)** and **Velocity Time Interval (Ticks)** inputs of the second Read/Write Control function.

24. Connect the **Velocity Setpoint (Counts/Interval)** output of the Scale Outputs (RT Host) VI to the corresponding input of the second Read/Write Control function.

**Figure 20.** Scale Inputs and Outputs Configured.

25. Right-click the **Loop Rate (Hz)** input of the Scale Inputs (RT Host) VI, and choose **Create»Control**.
26. Also connect the **Loop Rate (Hz)** input of the Scale Outputs (RT Host) VI to the control created in the above step.

**Figure 21.** Loop Rate Control Configured.

27. Right-click on the PWM Output (%) output of the Scale Inputs (RT Host) VI, and choose **Create»Indicator**. Repeat this for the Actual Rate (Hz) and Relative Position (Counts) outputs of the Scaled Outputs (RT Host) VI.

28. Wire the **Velocity (Counts/sec)** output of the Scale Inputs (RT Host) VI to the corresponding input of the Scale Outputs (RT Host) VI.

**Figure 22.** Scale Inputs Indicators Configured.

29. Right-click the **Set Point (Counts/sec)** input of the Scale Outputs (RT Host) VI, and choose **Create»Control**.
30. Right-click on the **Motor Plot** output of the Scale Outputs (RT Host) VI, and choose **Create»Indicator**.
31. Place a **Wait Until Next ms Multiple VI** in the loop from the Tim & Dialog function palette.
32. Right-click the **millisecond multiple** input of the Wait Until Next ms Multiple VI, and choose **Create»Control**. Label this control "Loop Rate."

**Figure 23.** Completed Host VI.

33. When completed the block diagram should look similar to the block diagram in the figure above.

Testing the Complete Application

34. Run the completed RT application. Adjust the velocity setpoint and observe the reaction of the system. Note: in order to limit sudden current spikes to the motor, set the velocity to zero before starting or stopping the application.

## 5. Appendix-Hardware Setup and Software configuration

### Hardware Setup Expansion System

1. Connect the cRIO-9151 Expansion chassis to the FPGA 7831\7811R device via the SH68-C68-S cable. Connect the cable to an available digital port on the 7831\7811R.
2. Afterwards, examine the CompactRIO modules and installed them into the chassis according to the following layout:

**Slot 1:** cRIO-9211 4-Channel Thermocouple Input Module

**Slot 2:** cRIO-9263 4-Channel Analog Output Module

**Slot 3:** cRIO-9411 6-Channel Digital Input Module

**Slot 4:** cRIO-9474 8-Channel Digital Output Module

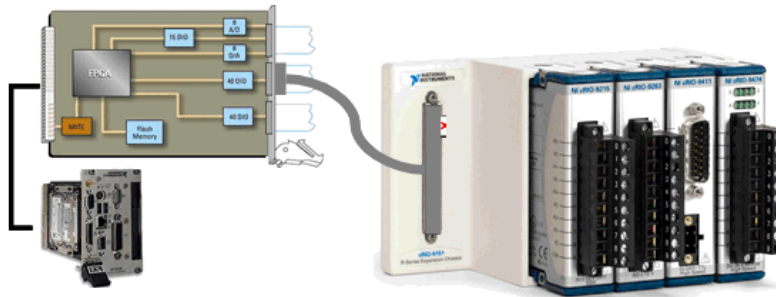


Figure 1. CompactRIO Expansion System

### Reconfigurable Embedded System

1. Install the cRIO-9002 Embedded Real-Time controller onto the controller slot of the cRIO-9101\9103 Reconfigurable chassis.
  2. Afterwards, examine the CompactRIO modules and installed them into the chassis according to the following layout:
- Slot 1:** cRIO-9215 4-Channel Analog Input Module
- Slot 2:** cRIO-9263 4-Channel Analog Output Module
- Slot 3:** cRIO-9411 6-Channel Digital Input Module
- Slot 4:** cRIO-9474 8-Channel Digital Output Module
3. Power the Controller by connecting an 11-30 Volt DC supply to the appropriate terminals.
  4. Connect the controller directly to the Host PC\PXI Chassis via a crossover Ethernet cable or through a LAN setup with a standard Ethernet cable.



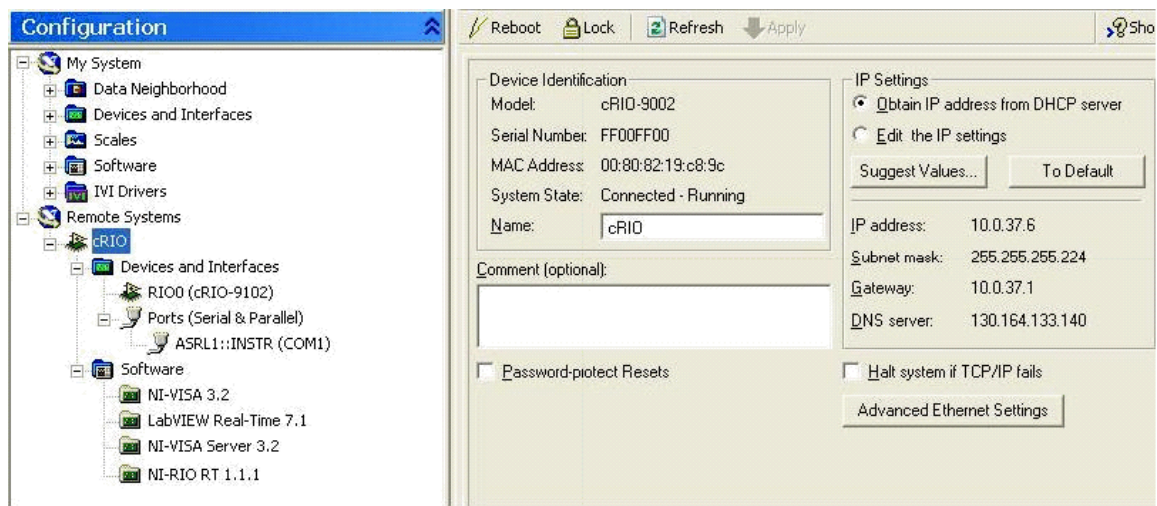
Figure 2. CompactRIO Embedded System

### Software Configuration of CompactRIO

1. Launch Measurement and Automation eXplorer (MAX)

### CompactRIO Embedded System

2. Expand **Remote Systems** to view available Real-Time controllers. You should see the CompactRIO embedded system listed as a target.
3. Configure the Network Settings if needed and install any necessary **Software**. The entry in MAX should appear as follows:



#### CompactRIO Expansion System

- Expand **Devices and Interfaces** under My System to view available Reconfigurable I/O devices. You should see the PXI/PCI-78xxR board listed as a device.
- Verify the **Software** on your system is installed as follows:

