

LabVIEW Real Time - CompactRIO Programming - Introduction

Author: Matthew Blair

Introduction

1. System Architecture
2. Hardware Setup
3. Software Configuration
4. Project Setup
5. LabVIEW FPGA VI
6. LabVIEW Real-Time VI
7. Troubleshooting



Under Construction!

Introduction

The National Instruments Compact Reconfigurable Input/Output system (CompactRIO or cRIO) is a data acquisition and control device with the capability to operate as a stand-alone system, without a PC. The adaptability and robustness of the CompactRIO system makes it suitable for data acquisition or control in remote locations or situations where it would be otherwise impractical to use a PC-based system. The CompactRIO is designed for use with LabVIEW, and requires the LabVIEW Real-Time and LabVIEW FPGA modules.

This tutorial will explain the steps necessary to develop an example data acquisition system, described below. The tutorial can be completed with or without the specified hardware components; an "offline configuration" mode allows you to simulate most steps without the hardware present. It is assumed that the reader is familiar with programming Virtual Instruments (VIs) in LabVIEW.

For additional information or support with LabVIEW or any National Instruments products, visit www.ni.com/support. National Instruments also has a large developer community that can be accessed at zone.ni.com. No matter what kind of application you are developing, it is almost certain that other LabVIEW developers have attempted something similar, and will be willing to help you.

Example System Specifications

The example data acquisition/control system we will develop makes use of an **NI cRIO-9012 Controller** and an **NI cRIO-9101 Chassis** with **two NI 9201 I/O modules** (analog voltage input) and an **NI 9401 I/O module** (digital input/output). The system operates as a stand-alone application, autonomously collecting analog voltage data through multiplexers that are articulated by the digital I/O module (the multiplexers are not part of this guide). Our goal is therefore simultaneous control and data acquisition: we need to output a combination of digital voltages (the "address" for the multiplexer) and acquire voltage signals as each "address" is given.

[Next: 1. System Architecture](#)