LabVIEW Real Time - CompactRIO Programming - System Architecture

Author: Matthew Blair

⚠ Under Construction!

System Architecture

Before diving into the intricacies of programming your cRIO system, it is instructive to think about the overall architecture, i.e. what components are present and how they are connected. Figures 1 and 2 present the architecture of our example system in schematic diagrams. Figure 1 shows the hardware layer, the physical components and their physical connections. Figure 2 shows the software layer, the programs that run on each hardware component and the software interfaces between them.

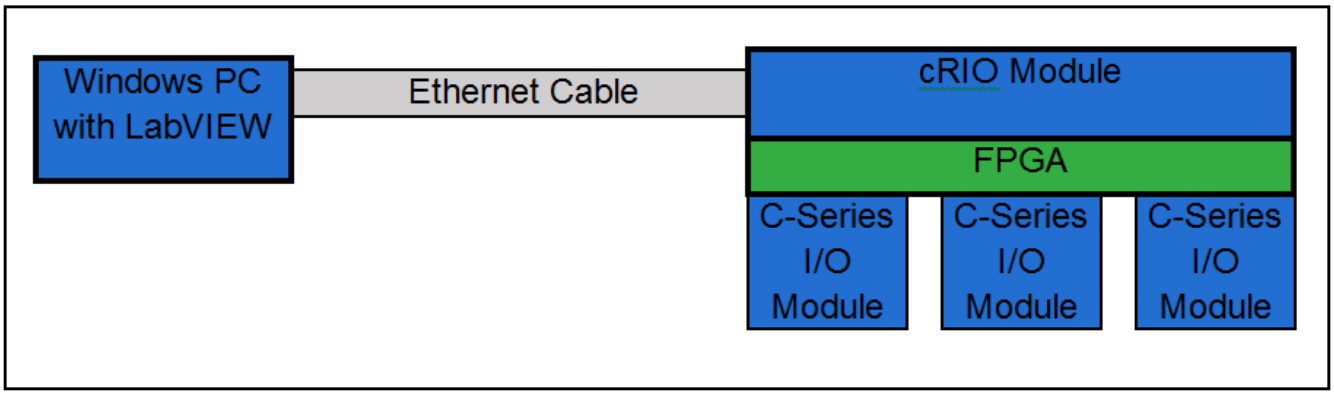

Figure 1: Hardware layer of example system architecture

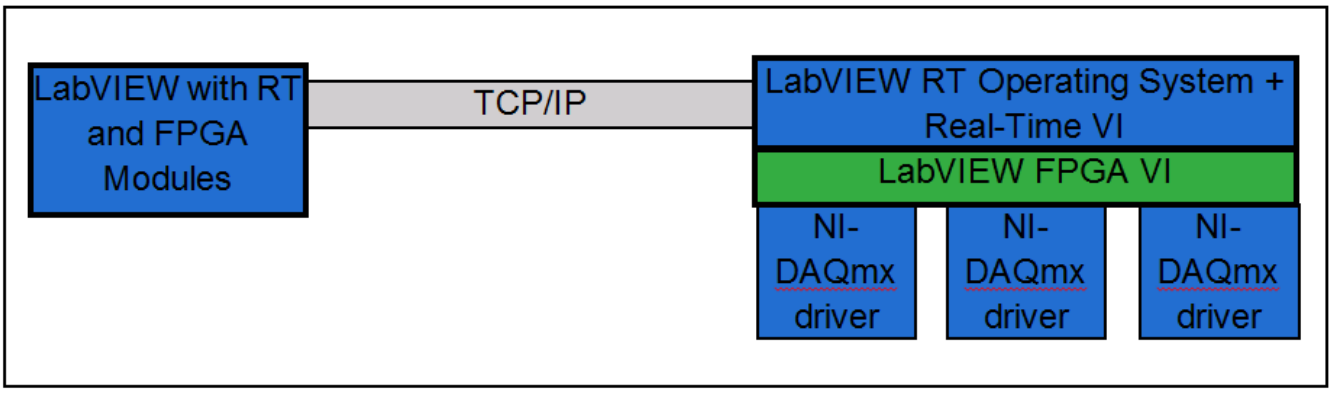

Figure 2: Software layer of example system architecture

The Windows PC on the left of figure 1 is where all of the programming is done. The PC must have LabVIEW installed (this guide assumes LabVIEW 2009 or later) as well as the LabVIEW Real-Time and FPGA modules, two software add-ons that are not part of the standard package. The PC is used to write and compile the VIs that make up your system, which LabVIEW then loads onto the cRIO module through an Ethernet connection. Within the module itself, there is a microprocessor running National Instruments' proprietary Real-Time Operating System (RT OS), this is known as the "Real-Time Host". Also inside the cRIO is the FPGA itself. The FPGA is the only component with access to I/O operations from the installed I/O modules or from the cRIO chassis itself (status LEDs, internal clock, and chassis temperature behave as I/O channels). While the FPGA can be configured to perform almost any desired logic operations and can communicate with the Real-Time Host, it doesn't have access to high-level communication protocols like the Real-Time Host does. Consequently, data acquisition must follow a path like that in figure 3.

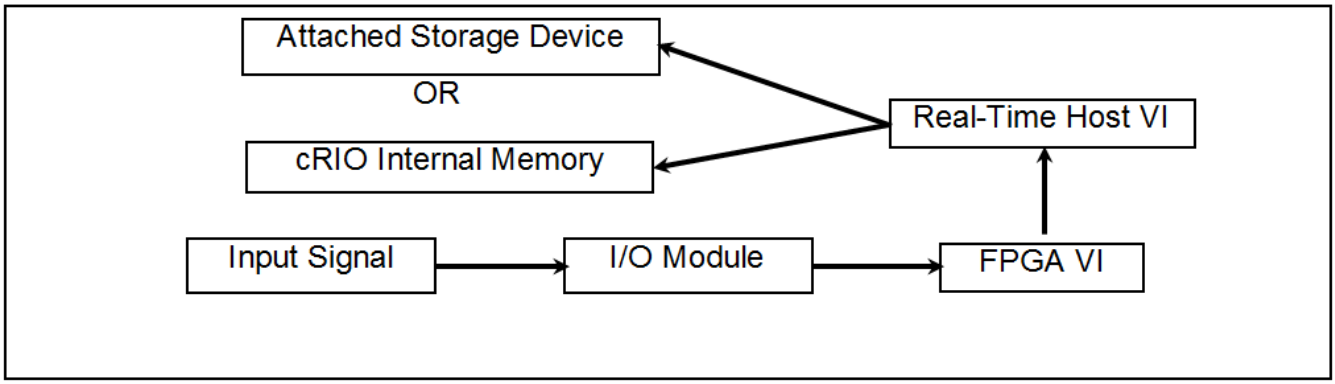

Figure 3: Data flow for data acquisition application

As you may have noticed in figures 2 and 3, there are, in fact, 2 distinct VIs in our example system. One is the FPGA VI, which defines the behavior of the FPGA in the cRIO module. For data acquisition, the FPGA VI performs I/O operations using the installed I/O modules and passes the collected data to the Real-Time Host VI. The FPGA VI can also operate on the I/O data before transferring it. Although the FPGA can only perform limited types of operations, it has the advantage of operating extremely fast, sometimes using as little as 1 clock cycle to manipulate data. The Real-Time Host VI resides on the microprocessor in the cRIO module. This VI receives the output data from the FPGA VI and, after possibly performing additional manipulations, writes the data to a storage device in a format that can be easily accessed from the Windows PC. The Real-Time Host VI can write files using File Transfer Protocol (FTP) to a USB mass storage device (for cRIO models with a USB port), storage devices attached by Ethernet, or the internal memory of the cRIO module.

Next: 2. Hardware Setup