Version 15.11

Hands-On Workshop

CompactRIO – Programming With LabVIEW Real-Time

Please do not remove this manual

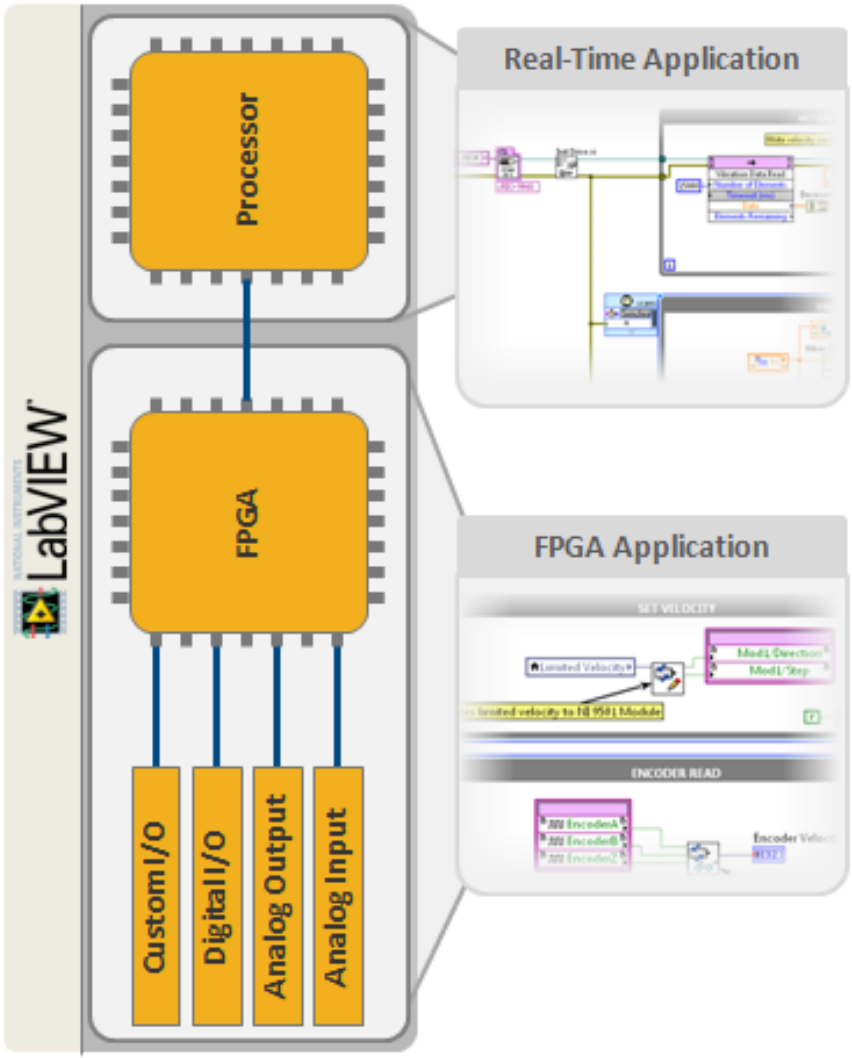
You will be sent an email which enables you to download the presentations and manual

uk.ni.com
ireland.ni.com

NATIONAL INSTRUMENTS

Hands-On: CompactRIO

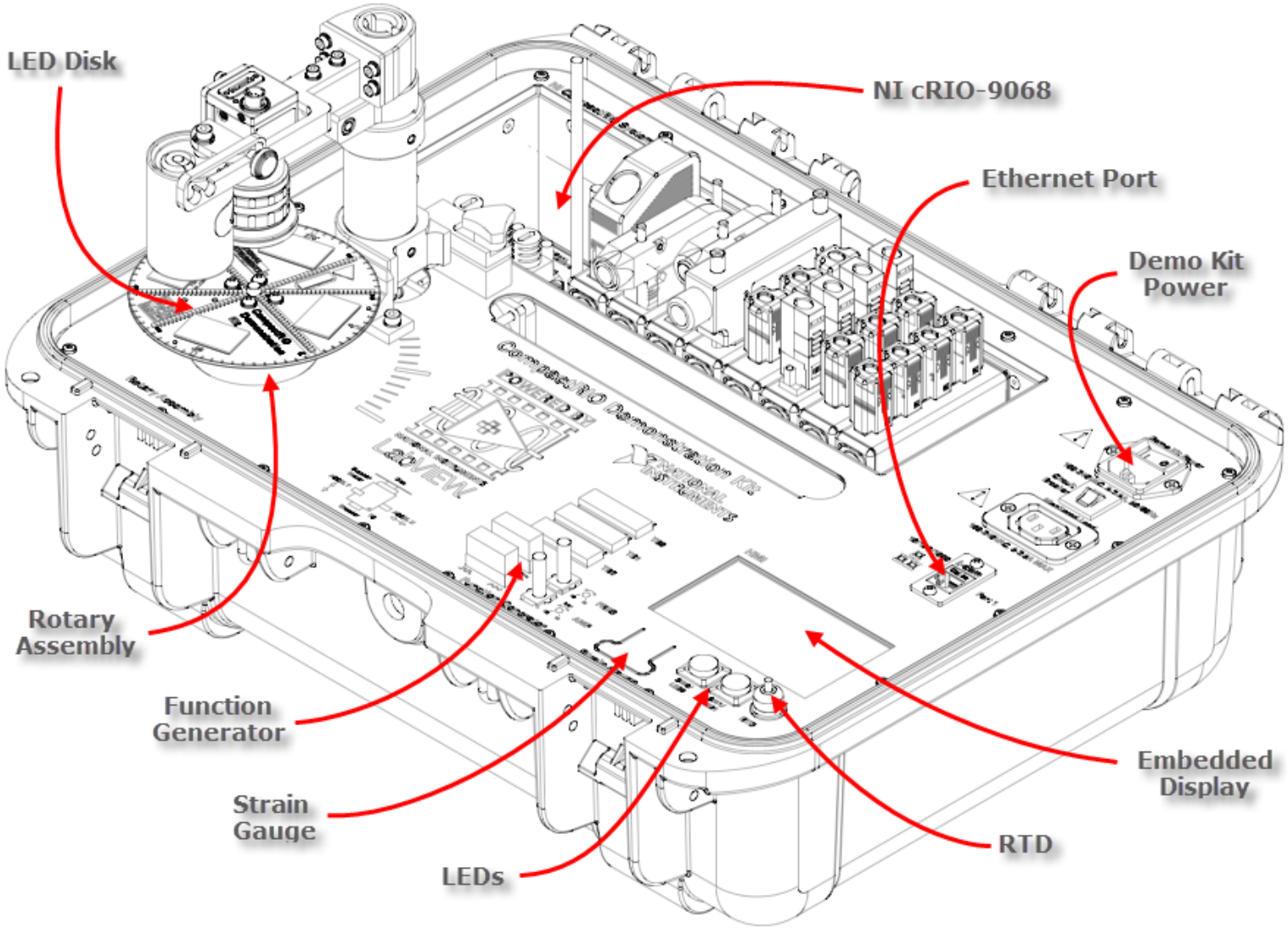
Contents



- Real-time OS
- Application software
- Networking and peripheral I/O drives
- DMA, interrupt, and bus control drivers

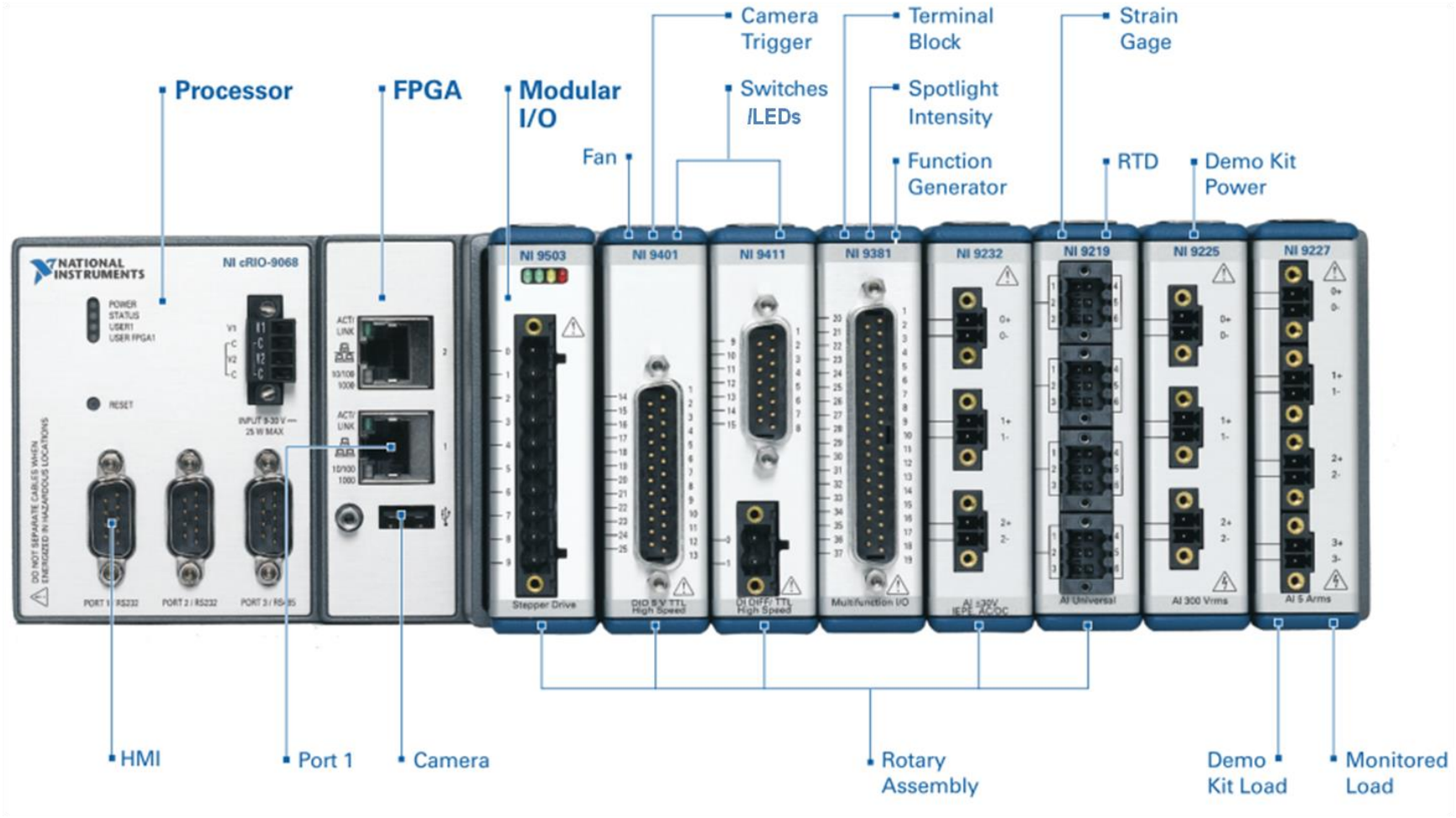
- Application IP
- Control IP
- Digital Signal Processing IP
- Specialized I/O drivers and interface
- DMA controller

NI CompactRIO DEMONSTRATION KIT



NI CompactRIO DEMONSTRATION KIT CONNECTION GUIDE

Built on the NI LabVIEW RIO Architecture

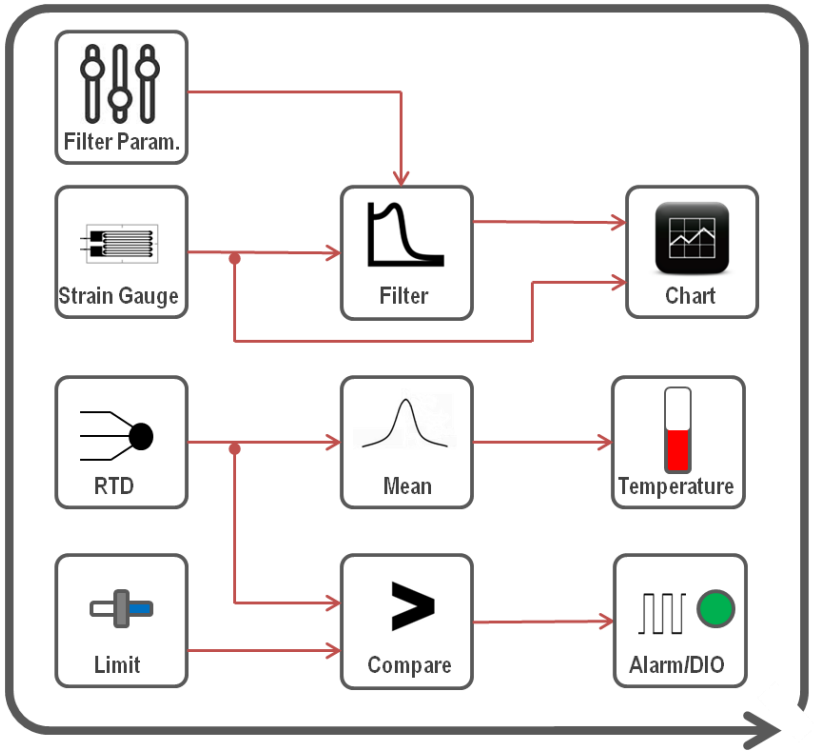


EXERCISE: TEMPERATURE AND STRAIN MONITORING

Goals

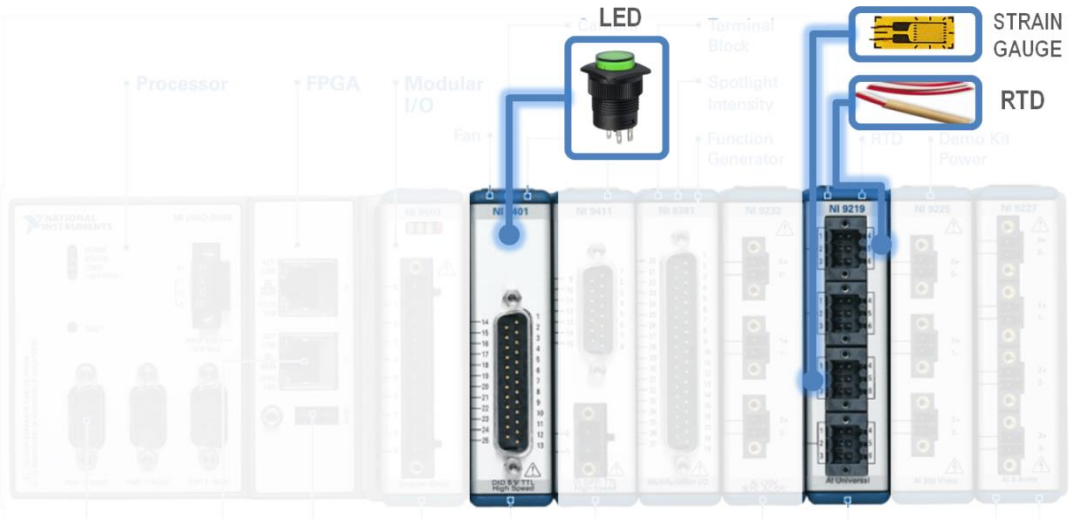
- Use ***LabVIEW*** system design software and the ***CompactRIO*** system configured in ***Scan Mode*** to interface with the sensors located in the ***CompactRIO Demonstration Kit***
- Use the ***LabVIEW Signal Processing*** library to obtain meaningful data from the sensors and display that information on a user interface

Part A—APPLICATION DESCRIPTION



In this exercise, use ***LabVIEW*** and the ***CompactRIO system*** configured in ***Scan Mode*** to acquire data from an ***RTD*** and a ***strain gauge***. Use the ***Signal Processing*** library to filter the strain gauge signal and obtain the mean of the RTD signal. Implement a ***limit-based alarm*** system for the RTD signal. Show the results of the analysis in a user interface.

CompactRIO System



NI Scan Mode

This feature gives you easy access to signals wired to measurement modules connected to the ***FPGA*** included in the ***CompactRIO system***.

In this exercise, use the measurement modules inserted in ***slots 2*** *and* ***6*** of the ***CompactRIO system***.

Inputs

Slot 6: **NI 9219**—*24-Bit Universal Analog Input (4 Diff, 100 S/s/ch)*
Analog Input 0 (AI0)→RTD (3-Wire Pt100-TCR3851)
Analog Input 2 (AI2)→Strain Gauge (Quarter Bridge/350 Ω)

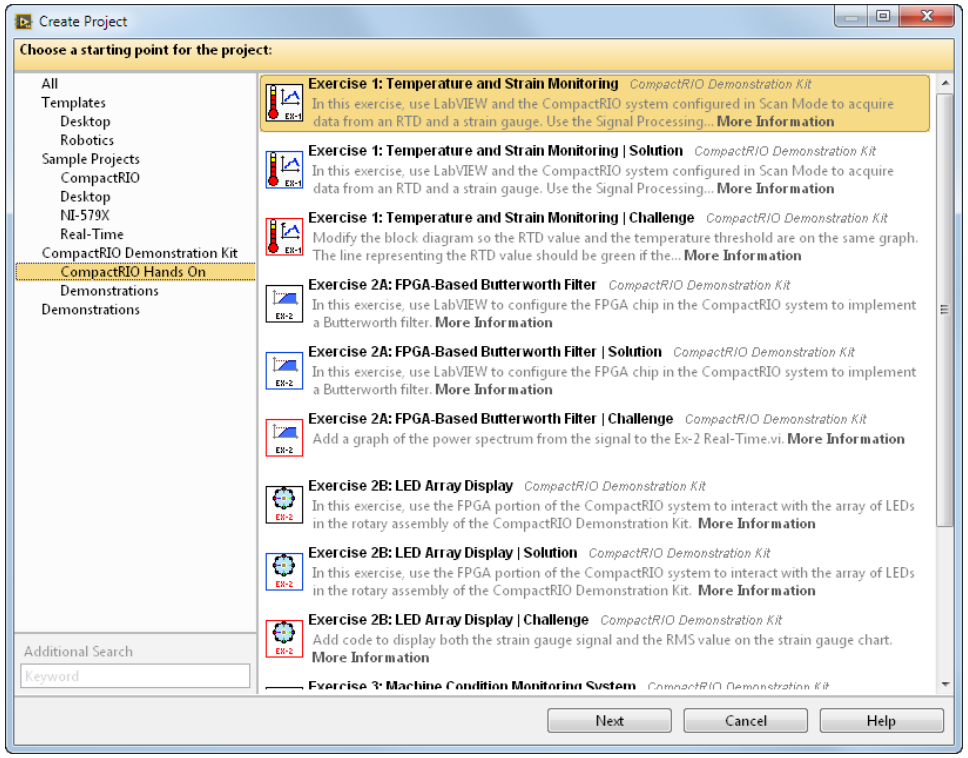
Outputs

Slot 2: NI 9401—*8 Ch, 5 V/TTL High-Speed Bidirectional Digital I/O Module*
Digital I/O 5 (DIO5)→LED1

Part B—CODE IMPLEMENTATION

1. Create an Instance of the Exercise 1: Temperature and Strain Monitoring Sample Project

Launch LabVIEW and create a new LabVIEW project from the Exercise 1: Temperature and Strain Monitoring sample project.



DETAILED INSTRUCTIONS

In this exercise, use an existing ***LabVIEW sample project*** as the starting point of the application.

- ***Launch LabVIEW***
  Launch LabVIEW by navigating to ***Start » All Programs » National Instruments » LabVIEW <Year>» LabVIEW.***

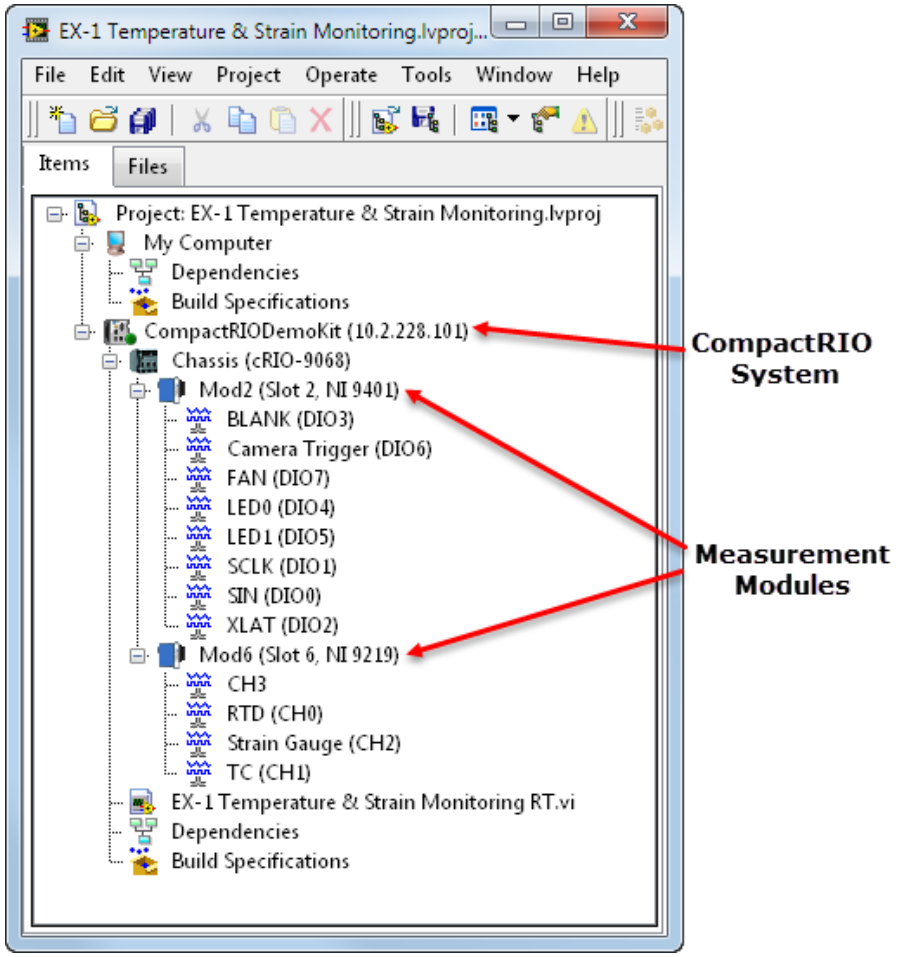
- ***Create an Instance of the Existing LabVIEW Project***
  Click on the ***Create Project*** button on the right side of the main screen of ***LabVIEW*** and create an instance of the **Exercise 1: Temperature and Strain Monitoring** sample project located at:

CompactRIO Demonstration Kit>> CompactRIO Hands On

Click **Next** to customize the **Project Name** and **Project Root** if desired. Finally, click **Finish** to create the project.

2. Explore the LabVIEW project.

Explore the project and expand the hierarchy on the project to display all measurement modules and channels to be used in the exercise.



DETAILED INSTRUCTIONS

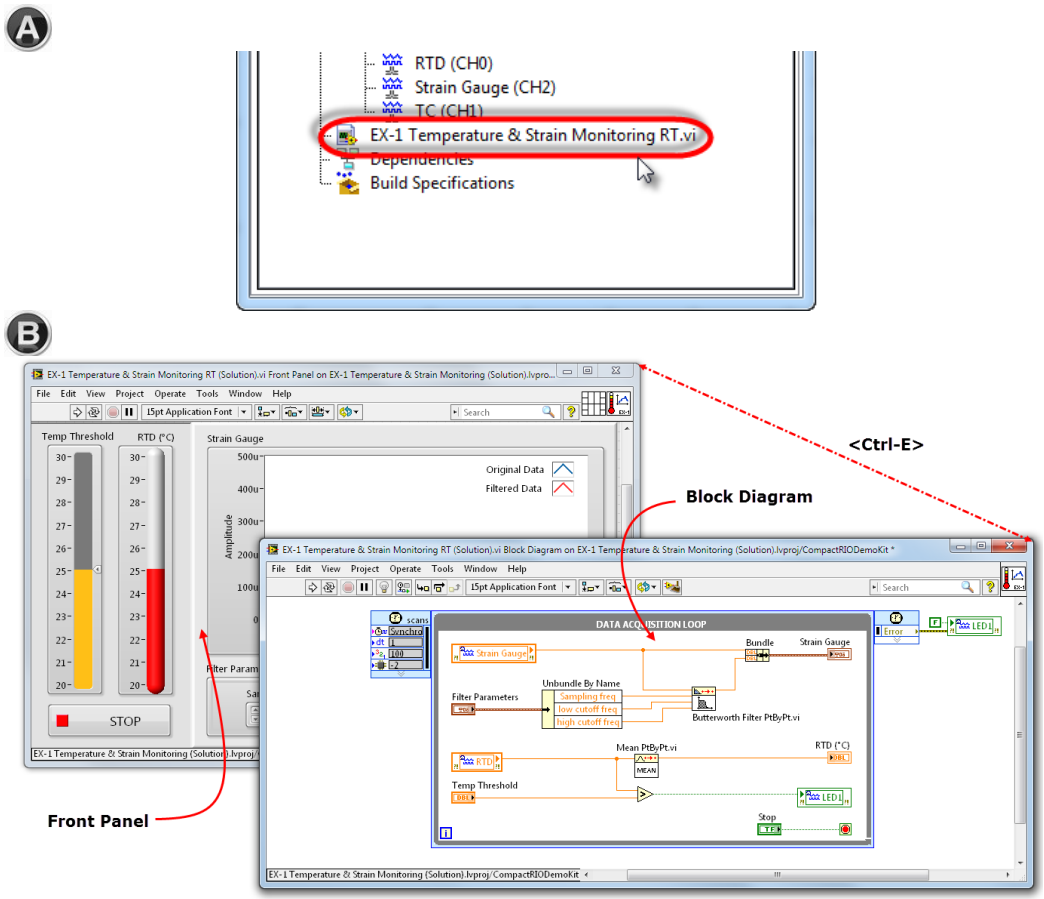
- ***Expand of the Exercise 1: Temperature and Strain Monitoring sample project***
  This project contains the ***CompactRIO system*** and ***measurement modules*** for this exercise.

  To expand the hierarchy and reveal the contents of the project, click the "**+**" boxes next to each part of the CompactRIO system and measurement modules listed.

  The inputs and outputs for this exercise have already been named and are listed under their corresponding modules.

3. Open the EX-1 Temperature & Strain RT.vi file.

This file contains the user interface and the block diagram with the code that you need to complete the temperature control exercise.



DETAILED INSTRUCTIONS

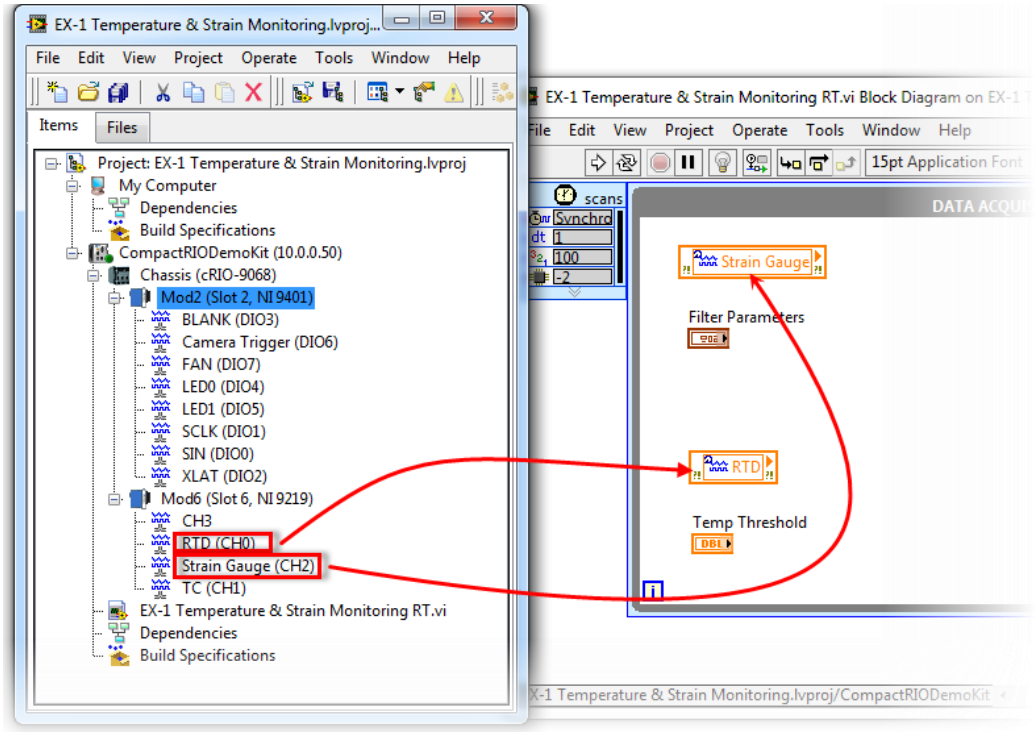
- ***Open the EX-1 Temperature & Strain Monitoring RT.vi File***
  Double-click on the ***EX-1 Temperature & Strain Monitoring RT.vi*** file located at the bottom of the LabVIEW Project Explorer window as shown in ***Figure A***.

- ***Display the Block Diagram***
  On the menu bar, click on ***Window » Show Block Diagram*** or press ***<Ctrl-E>*** to switch between the ***front panel*** and the ***block diagram*** as shown in ***Figure B***.

  Complete the code by inserting the inputs and outputs from the measurement modules into the ***While Loop*** labeled ***Data Acquisition***. Using a ***Butterworth Filter*** function and a ***Mean function***, monitor the ***RTD*** and the ***Strain Gauge***.

4. Add the inputs of the monitoring application to the block diagram.

Place the RTD and Strain Gauge I/O channels onto the block diagram.



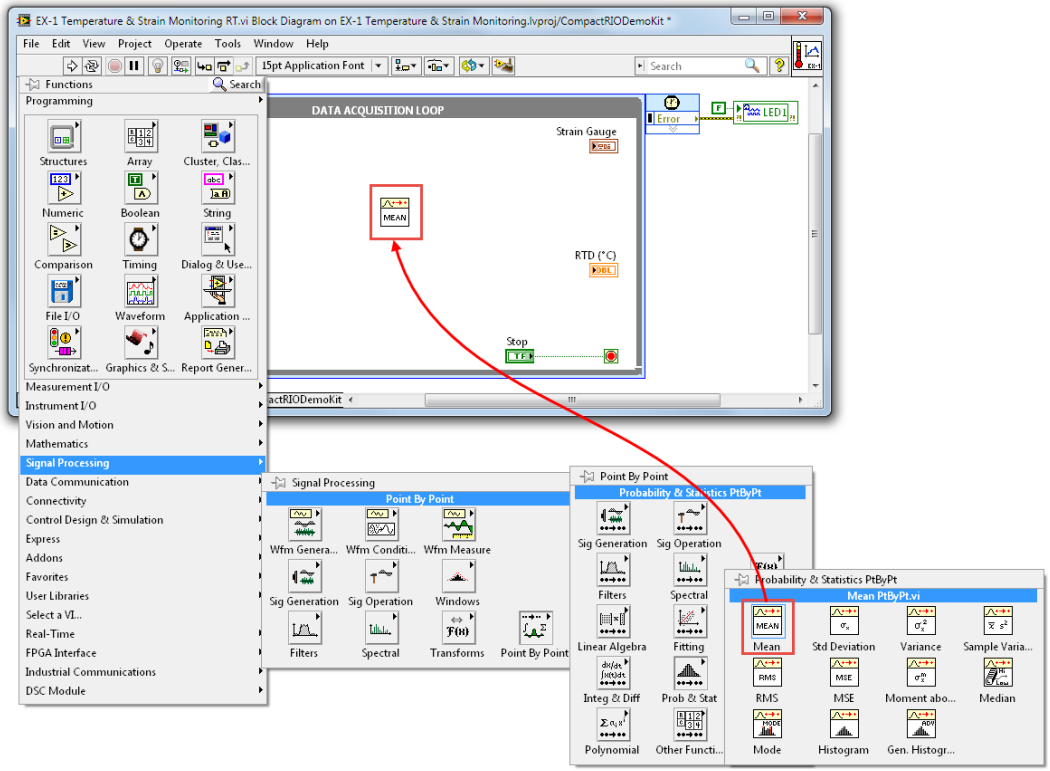
DETAILED INSTRUCTIONS

- ***Add the RTD I/O Node to the Block Diagram***
  From the LabVIEW Project Explorer window, drag and drop the channel labeled ***RTD (CH0)*** under ***Mod6 (Slot6, NI 9219)*** onto the block diagram inside the ***Data Acquisition loop***.

- ***Add the Strain Gauge I/O Node to the Block Diagram***
  From the LabVIEW Project Explorer window, drag and drop the channel labeled ***Strain Gauge (CH2)*** under ***Mod 6 (Slot6, NI 9219)*** onto the block diagram inside the ***Data Acquisition loop***.

5. Implement the code for calculating the mean of RTD measurements.

Insert the Mean Point-by-Point.vi function onto the block diagram.



DETAILED INSTRUCTIONS

- ***Insert the Mean Point-by-Point.vi Function on the Block Diagram***
  Place the ***Mean Point-by-Point.vi*** inside the **Data Acquisition loop** by right-clicking on the block diagram to open the ***Functions Palette***.

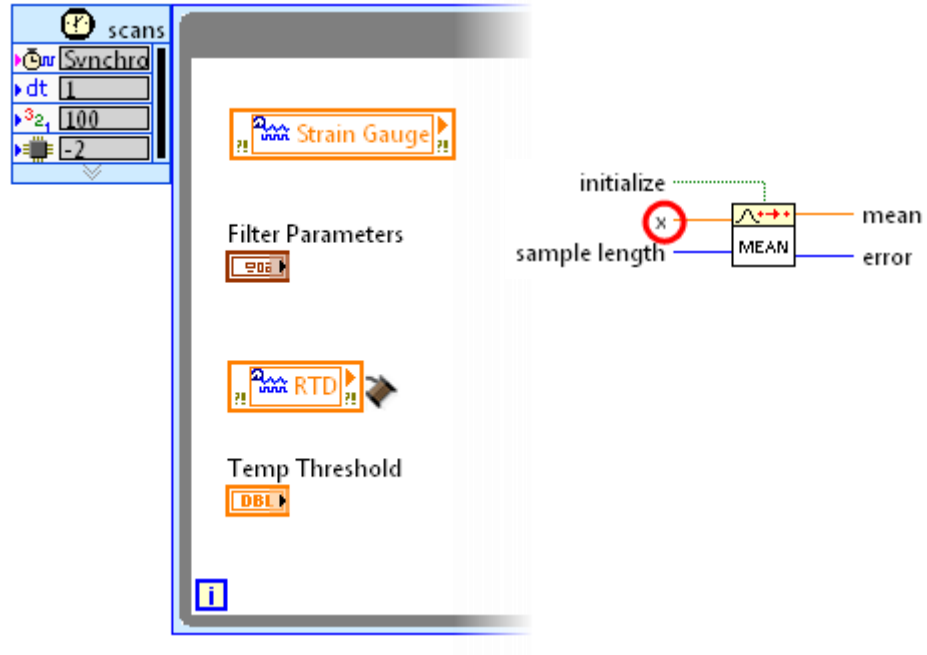
  Navigate to this function by clicking to ***Signal Processing » Point by Point » Prob & Stat » Mean PtByPt.vi***.

  Point-by-point analysis is a method of continuous data analysis in which analysis occurs for each data point, point by point. In point-by-point analysis, the input-analysis-output process takes place continuously in ***real time***.

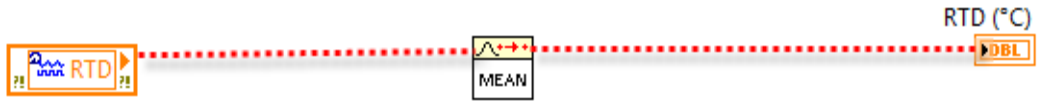
6. Wire the inputs and outputs of the MeanPtByPt.vi function.

Wire the RTD I/O terminal to the "x" input terminal of the MeanPtByPt.vi function and wire the output of the MeanPtByPt.vi function to the RTD (°C) indicator.

A



B



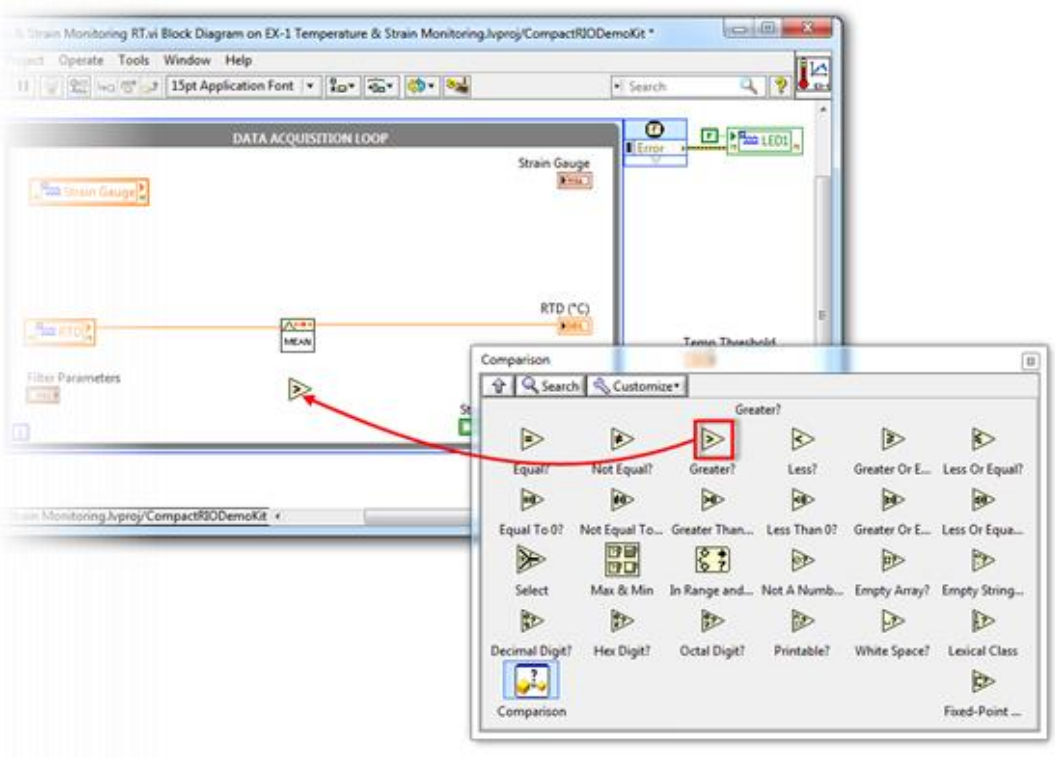
DETAILED INSTRUCTIONS

- ***Wire the RTD I/O Terminal to the "x" Terminal of the Mean PtByPt. vi***
  Move the cursor over the triangle in the upper-left corner of the ***RTD I/O terminal***. The cursor should become a spool; this indicates the cursor is in wiring mode. See ***Figure A***. Click on the terminal and move the cursor to the ***"x"*** input terminal of the ***Mean PtByPt.vi*** and click. A wire now connects the ***RTD I/O terminal*** to the ***Mean PtByPt*** function.

- ***Wire the Output of the Mean PtByPt.vi to the RTD (°C) Indicator***
  Create a wire that connects the Mean output terminal of the ***Mean PtByPt.vi*** function to the ***RTD (°C)*** indicator. The inputs and outputs should look like ***Figure B***.

7. Implement a temperature threshold for the system.

Add logic to the code to determine when the temperature is greater than a user-defined threshold.



DETAILED INSTRUCTIONS

- ***Add the Greater? Function to the Block Diagram***
  Place the ***Greater?*** function on the block diagram by opening the ***Functions Palette*** and navigating to ***Comparison » Greater?***.

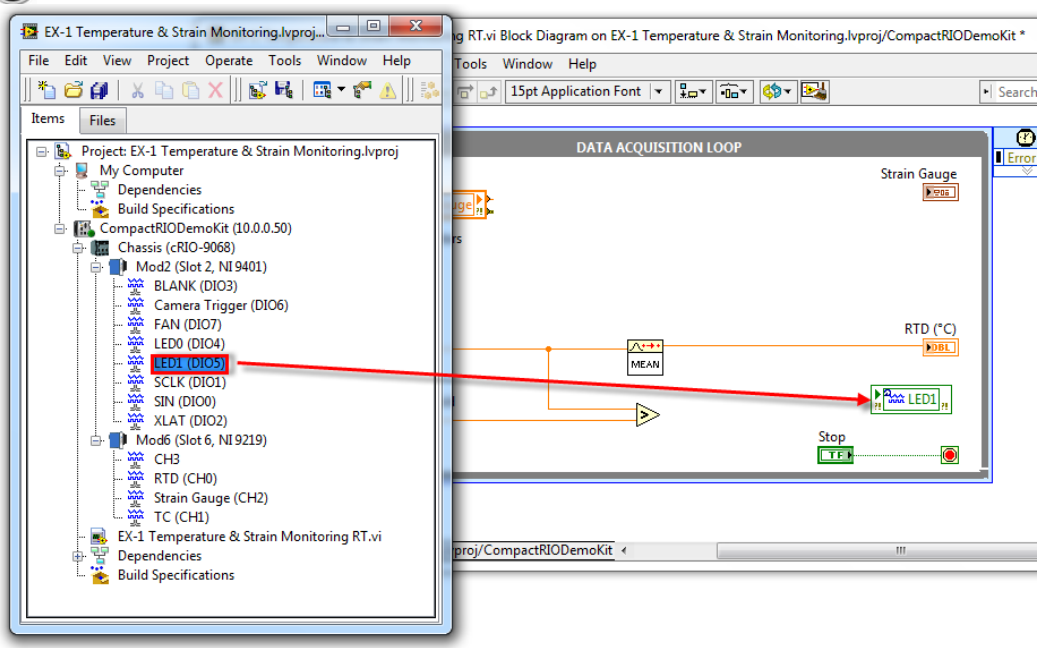
  This function helps you implement the ***limit-based alarm system***. You can now compare the current RTD value against a user-defined limit to fire up an alarm.

8. Complete wiring the temperature threshold.

Wire the output terminals of the RTD I/O terminal and the Temperature Threshold control to the Greater? Function. Also create an instance of the LED indicator that illuminates when the RTD I/O output is greater than the temperature threshold.

A

B



DETAILED INSTRUCTIONS

- ***Wire Inputs to the Greater? Function***
  Wire the output of the ***RTD I/O terminal*** to the first input terminal of the ***Greater?*** function by creating a branch from the wire connecting the ***RTD I/O terminal*** and the ***Mean PtByPt.vi***. Click on the wire with the cursor in wiring mode and move the cursor to the first input terminal of the ***Greater?*** function.

  Wire the output of the ***Temp Threshold control*** to the second input terminal of the ***Greater?*** function as shown in ***Figure A***.

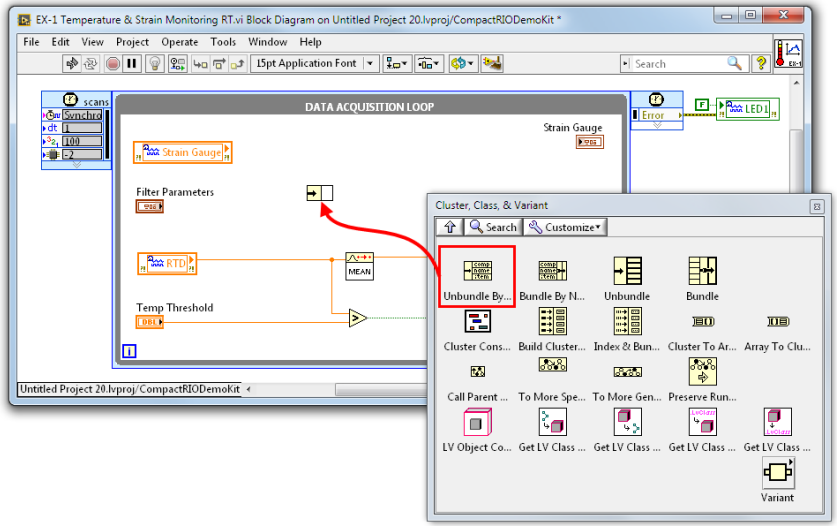
- ***Create an Instance of the LED Indicator***
  Create an instance of the ***LED*** indicator by dragging and dropping the ***LED1 (DIO5)*** from under ***Mod 2 (Slot2, NI 9401)*** in the LabVIEW Project Explorer into the ***Data Acquisition loop*** as shown in ***Figure B.***

  Wire the output of the ***Greater?*** function to the input terminal of the ***LED1 I/O terminal***.

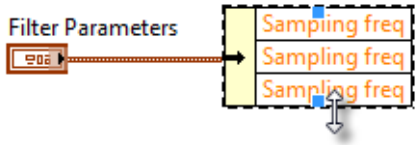
9. Unbundle the values of the filter parameters cluster.

Unbundle the values of the filter parameters cluster to access the individual frequency values within the cluster.

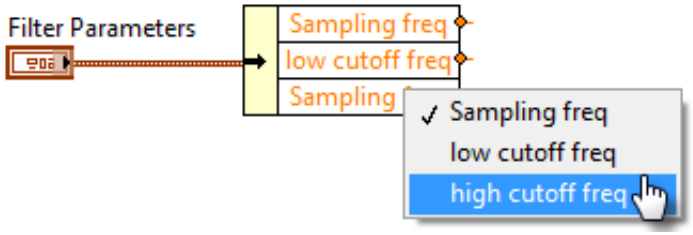
A



B



C



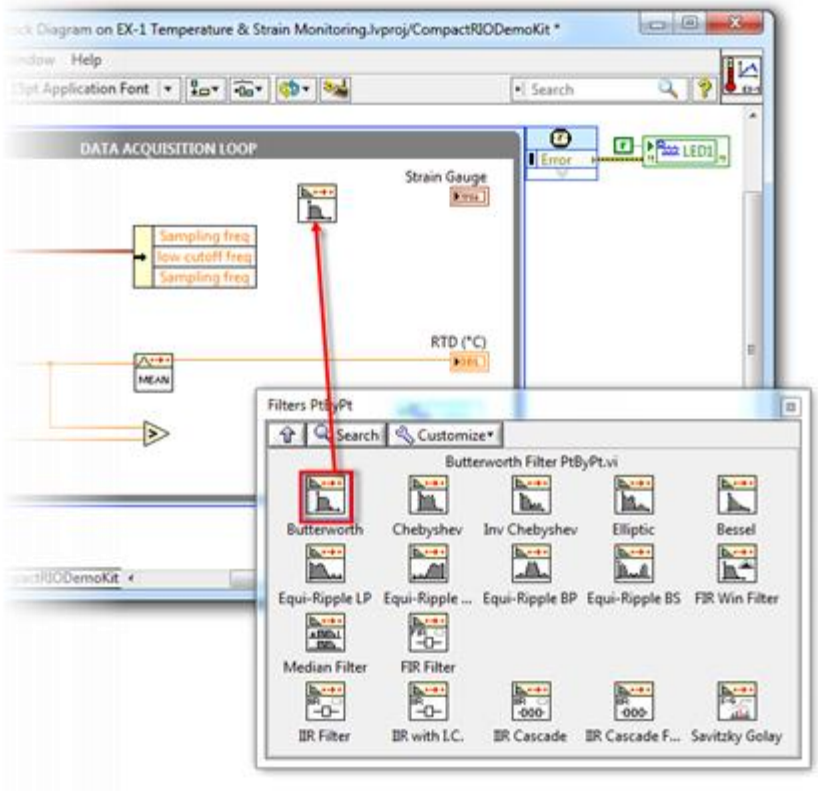
DETAILED INSTRUCTIONS

- ***Use the Unbundle By Name Function to Access the Frequency Values***
  Place an ***Unbundle By Name*** function onto the block diagram by opening the ***Functions Palette*** and navigating to ***Cluster, Class, & Variant » Unbundle By Name*** as shown in ***Figure A***.

  Wire the output from the ***Filter Parameters*** control to the input terminal of the ***Unbundle By Name*** function as shown in ***Figure B***. Drag the bottom resizing tool on the ***Unbundle By Name*** function down to reveal the three boxes labeled ***"Sampling freq"*** as shown in ***Figure B***. Select the frequency that each box represents as shown in ***Figure C.***

10. Implement a Butterworth filter to filter the strain gauge value.

Insert the Butterworth Filter PtByPt.vi onto the block diagram to provide filtering for the strain gauge.



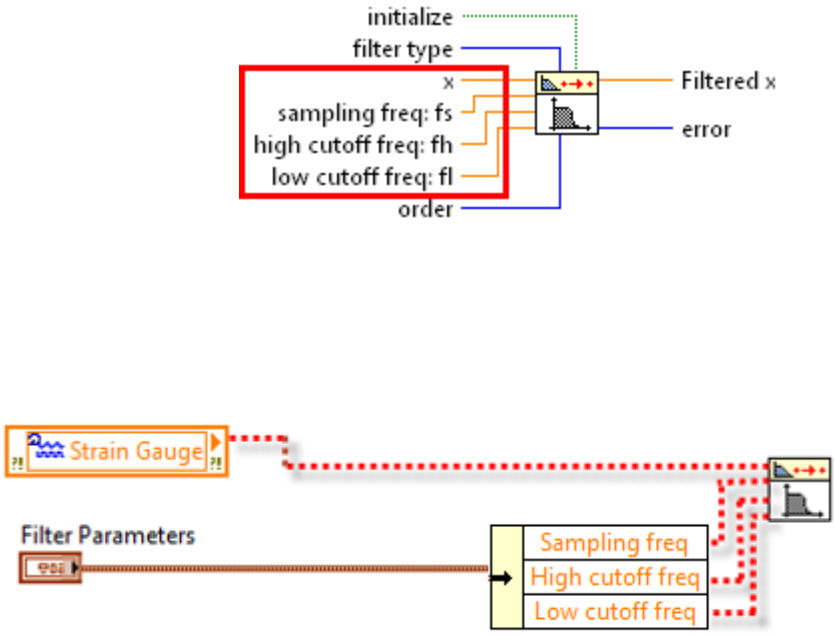
DETAILED INSTRUCTIONS

- ***Insert the Butterworth Filter PtByPt.vi Onto the Block Diagram***
  Place the ***Butterworth Filter PtByPt.vi*** onto the block diagram by opening the ***Functions Palette*** and navigating to ***Signal Processing » Point By Point » Filters » Butterworth Filter PtByPt.vi***.

  Wire the output of the ***Strain Gauge I/O*** terminal to the first input terminal, or the **"x"** terminal, of the ***Butterworth Filter*** function.

11. Wire the filter parameters and the strain gauge I/O terminal to the Butterworth Filter PtByPt function.

Bundle the raw Strain Gauge Value and the output of the Butterworth Filter PtByPt function together and display on the same graph on the front panel.



DETAILED INSTRUCTIONS

- ***Wire the Outputs of the Unbundle By Name Function and the Strain Gauge I/O Terminal to the Butterworth Filter PtByPt Function***

  Wire the output of the ***Strain Gauge I/O*** terminal to the first input terminal, or the **"x"** terminal, of the ***Butterworth Filter*** function.

  Wire the ***Sampling freq terminal*** from the ***Unbundle By Name*** function to the second input terminal of the ***Butterworth Filter*** function.

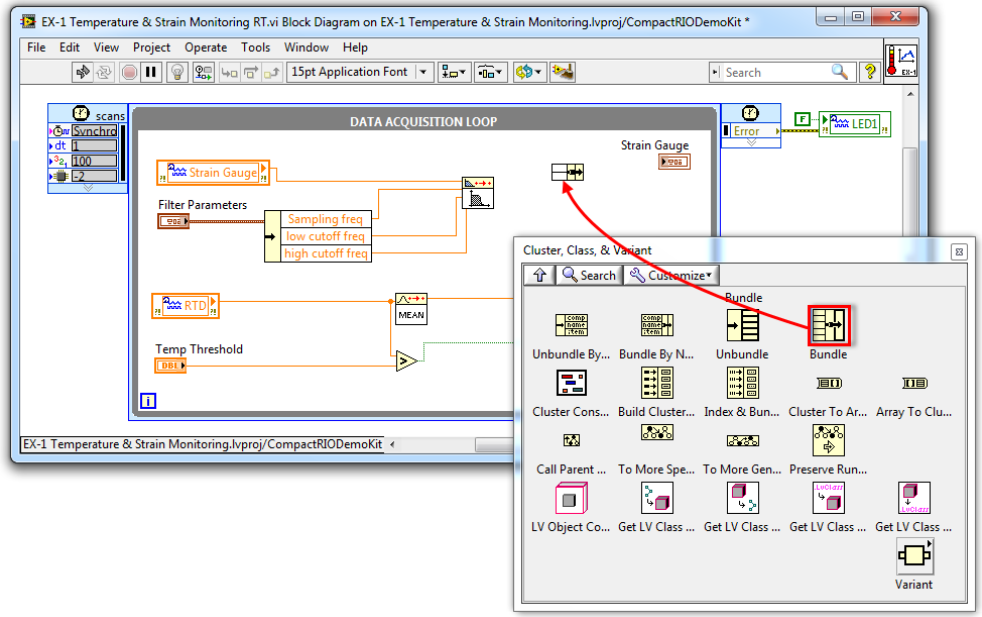
  Wire the low ***cutoff freq terminal*** from the ***Unbundle By Name*** function to the third input terminal of the ***Butterworth Filter*** function.

  Finally, wire the ***high cutoff freq terminal*** of the ***Unbundle By Name*** function to the last input terminal of the ***Butterworth Filter*** function.

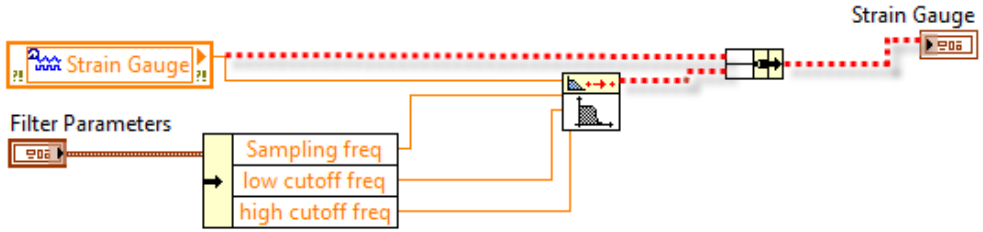
12. Display the raw strain gauge value and the filtered strain gauge value on the front panel.

Bundle the raw Strain Gauge Value and the output of the Butterworth Filter PtByPt function together and display on the same graph on the front panel.

A



B



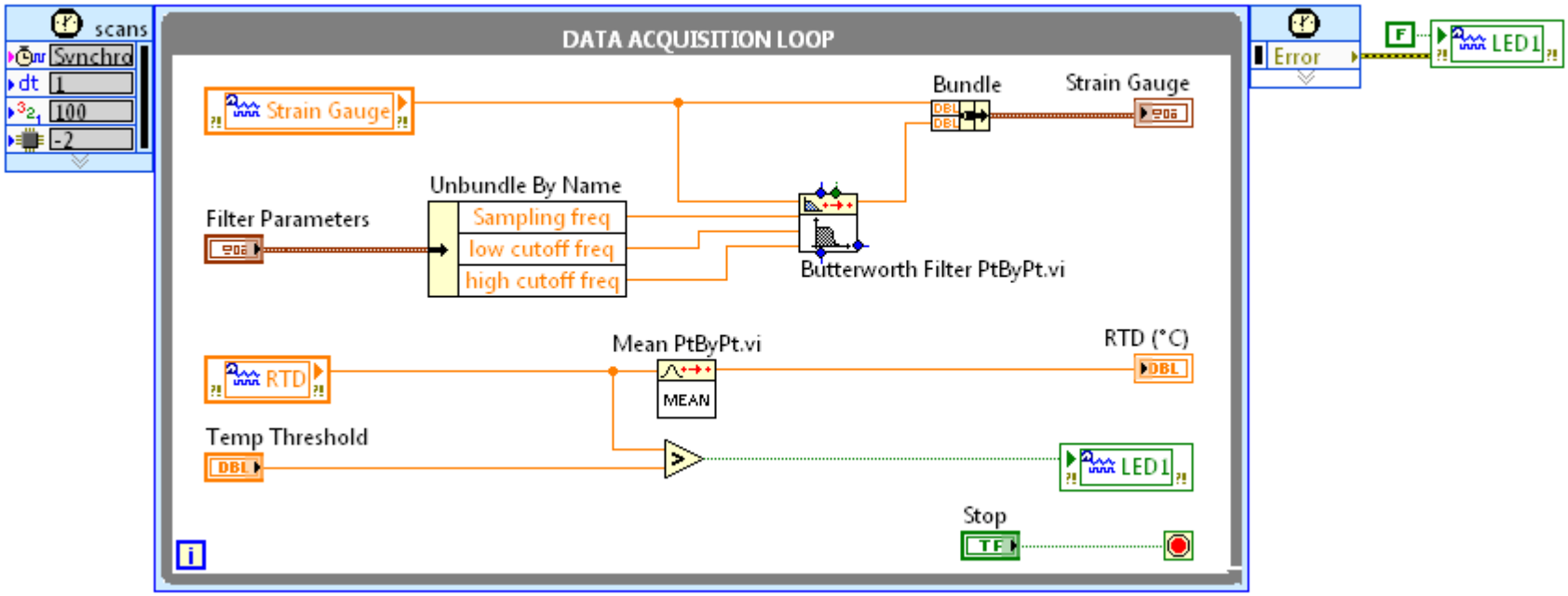
DETAILED INSTRUCTIONS

- ***Bundle the Raw Strain Gauge Value and the Filter Stain Gauge Value***
  Place a ***Bundle*** function on the block diagram by opening the ***Functions Palette*** and navigating to ***Cluster, Class, & Variant » Bundle***. Wire the output from the ***Strain Gauge I/O terminal*** to the first input terminal of the ***Bundle*** function. Wire the output of the ***Butterworth Filter PtByPt.vi*** to the second input terminal of the ***Bundle*** function as shown in ***Figure A.***

- ***Display the Strain Gauge Values on the Same Graph***
  Wire the output of the ***Bundle*** function to the ***Strain Gauge cluster indicator*** to display the values on the same graph. The wiring should be similar to the wiring in ***Figure B***.

13. The completed block diagram should look like the image below.

Be sure to save your work by pressing <Ctrl-S> or clicking File » Save.

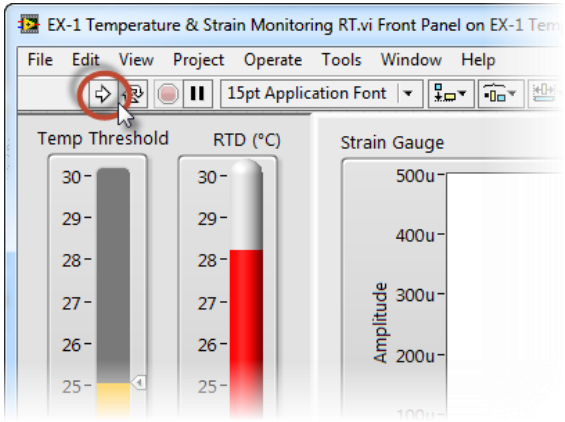


Part C—RUN THE APPLICATION

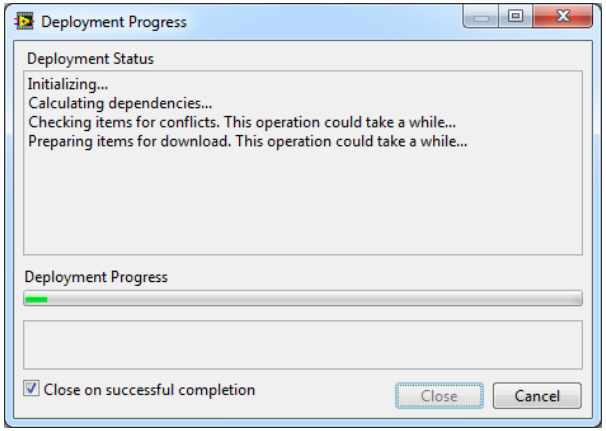
14. Run the application.

Run the EX-1 Temperature & Strain Monitoring RT.vi by clicking the run arrow to deploy the code to the CompactRIO system.

A



B



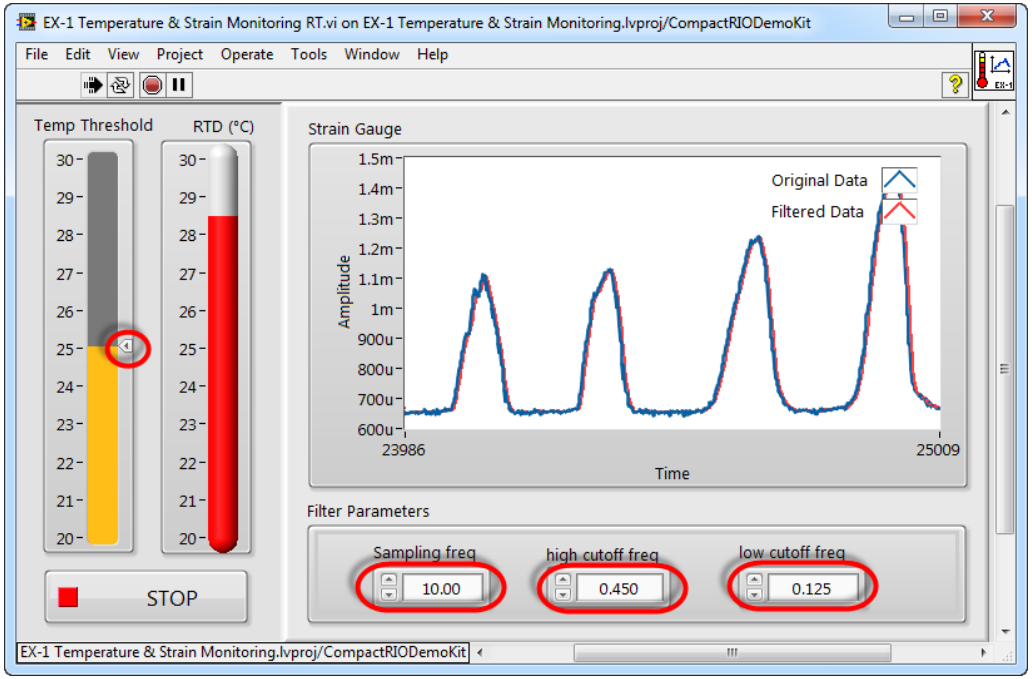
DETAILED INSTRUCTIONS

- ***Run the VI***
  Switch back to the front panel **<Ctrl-E>** and run the program by clicking on the **arrow button** at the top of the front panel as shown in ***Figure A***. Save the VI when prompted.

- ***Deploy the VI***
  Running the VI deploys the code through the ***Ethernet cable*** to the real-time controller of the ***CompactRIO system*** as shown in ***Figure B***. This initializes an interactive execution of the code in which the code is executed on the CompactRIO system while you monitor and control inputs and outputs from the user interface on the development machine.

15. Modify the Temp Threshold and Filter Parameters values to interact with the system.

Change the values of the Temp Threshold and Filter Parameters on the front panel to interact with the application.



DETAILED INSTRUCTIONS

- ***Change the Value of the Temp Threshold on the Front Panel***
  Slide the ***Temp Threshold control*** to change the value of the temperature threshold for the system.

- ***Change the Values of the Filter Parameters on the Front Panel***
  Adjust the values of the different ***Filter Parameter values*** to see how the filtering of the strain gauge measurement changes.

16. Stop the application.

Stop the application of the code by pressing the Stop button on the front panel of the EX-1 Temperature & Strain Monitoring RT.vi.

Part D—CHALLENGE

17. Modify the code to display temperature readings on a graph.

Modify the block diagram so the RTD value and the temperature threshold are on the same graph. The line representing the RTD value should be green if the value is below the temperature threshold and red if the value is above the temperature threshold.

<END OF EXERCISE 1: TEMPERATURE AND STRAIN MONITORING>

ADDITIONAL RESOURCES

BUILD YOUR OWN EMBEDDED SYSTEM WORKSHOP

In the NI Build Your Own Embedded System hands-on workshop, focus on extending your NI LabVIEW skills into FPGA-based embedded design using NI reconfigurable I/O (RIO) hardware.

Purchase the LabVIEW RIO Evaluation kit through the registration process and attend the workshop to receive the following:

- 90-day evaluation of LabVIEW and the LabVIEW FPGA and LabVIEW Real-Time modules
- Board-level NI RIO evaluation hardware device and daughterboard for easy I/O interfacing
- Introduction to the LabVIEW RIO architecture with a qualified NI instructor
- Hands-on experience building your first FPGA-based RIO embedded system with the evaluation kit

To find an event in your area, visit ***ni.com/byoes***.

LabVIEW RIO EVALUATION KIT

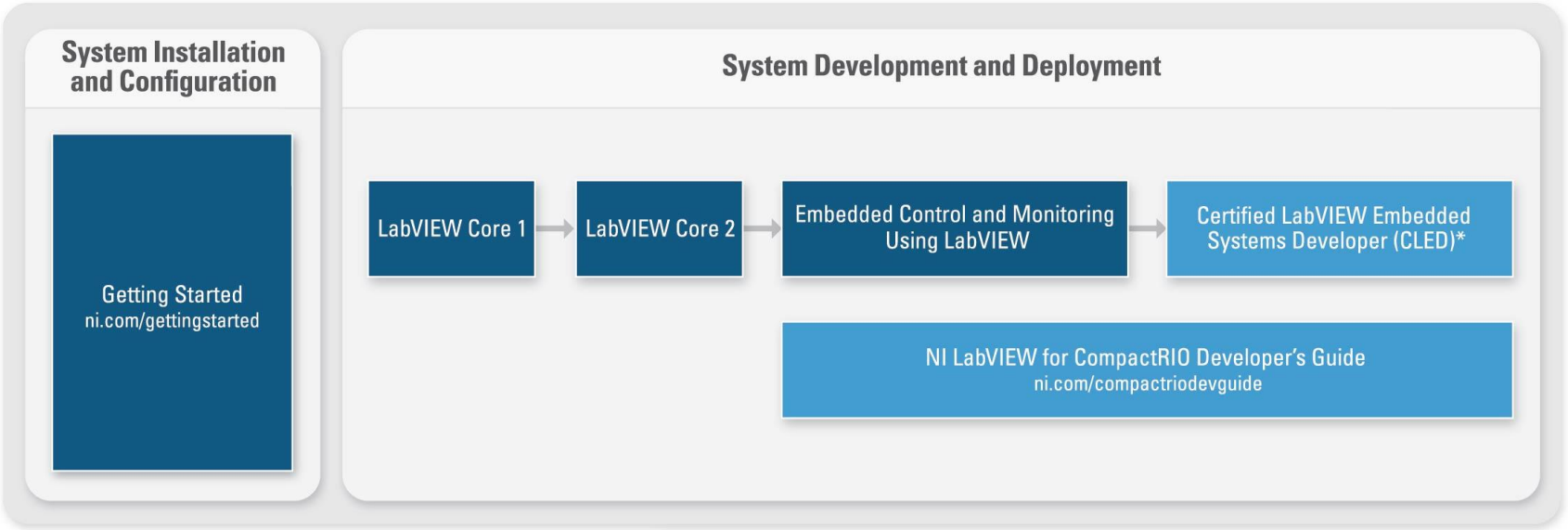
NATIONAL INSTRUMENTS LabVIEW™

Using the evaluation kit, develop an embedded system with the LabVIEW RIO architecture. LabVIEW system design software helps you program NI RIO hardware, which includes a real-time processor, FPGA, and I/O. NI CompactRIO hardware uses this same architecture for prototyping through deployment with a flexible array of configuration, expansion, and NI C Series module I/O options.

The kit includes an extended evaluation of the LabVIEW FPGA and LabVIEW Real-Time modules; an NI RIO evaluation device; a daughterboard for easy I/O interfacing; a step-by-step tutorial; and numerous fully documented, ready-to-run examples of common embedded tasks implemented in LabVIEW.

To learn more, visit ***ni.com/rioeval***.

NI LabVIEW Real-Time and NI LabVIEW FPGA RECOMMENDED RESOURCES AND TRAINING OPTIONS



* A CLD or higher is required before attempting the CLED exam