LabVIEW Real-Time Control on CompactRIO

Technical Manual for Compact RIO Demostration Kit

Version 1.1.0, September, 2014

Owner Contact Information

Name: Deborah Yagow, LabVIEW RIO PME

Email: Deborah.yagow@ni.com

Address: x36554

NATIONAL INSTRUMENTS™

Table of Contents

REQUIRED HARDWARE **2**
SUPPORTED DEMONSTRATION KIT VERSIONS 2
DEMONSTRATION KIT COMPONENTS USED 2
REQUIRED HARDWARE COMPONENTS 2
REQUIRED SOFTWARE **2**
DESCRIPTION **3**
GOAL OF DEMONSTRATION 3
KEY CONCEPTS DEMONSTRATED 3
SCENARIO 3
APPLICATION DIAGRAM 4
THEORY OF OPERATION 4
SET-UP PROCEDURE **5**
DEMO PROCEDURE/SCRIPT **7**
SHUTDOWN PROCEDURE **10**
TROUBLESHOOTING STEPS **10**
ERROR DEPLOYING PROJECT – SCANNED I/O BUS NOT IN REQUIRED I/O MODE 11

To download a copy of this manual and the latest version of the Demo code referenced in the manual, please visit: CompactRIO Demonstration Kit NI Talk Page

Required Hardware

Supported Demonstration Kit Versions

- CompactRIO Demonstration Kit 1.0

Demonstration Kit Components Used

- NI 9401 8 Ch, 5V/TTL High-Speed Bidirectional Digital I/O Module
- NI 9411 Digital Input Module
- NI 9381 Multifunction AI, AO, DIO Module
- NI 9219 24-Bit Universal Analog Input Module

Required Hardware Components

- Region-Specific Power Cable
- Ethernet Cable
- Four short wires

Required Software

- LabVIEW 2014 or Newer
- LabVIEW Real-Time 2014 or Newer
- LabVIEW PID and Fuzzy Logic Toolkit
- NI-RIO 14.0.1 or Newer

DESCRIPTION

The LabVIEW Real-Time Control on CompactRIO (RIO Scan Interface) demo implements deterministic, software-based control of a simulated plant. This sample project is based on the Simple State Machine and Queued Message Handler templates.

GOAL OF DEMONSTRATION

This sample project uses the RIO Scan Interface (RSI), highlighting an alternative to programming with the LabVIEW FPGA Module, to perform I/O on the FPGA. The RSI programming mode is available on most CompactRIO and some older Single-Board RIO devices and is sufficient for applications that require single-point access to I/O at rates of a few hundred Hertz.

Additional benefits to emphasize to the customer include:

- Deterministic, software-based control, executing parallel control algorithms that run at different rates.
- A User Interface VI that can disconnect from the device and reconnect at any time without affecting the control loop. The user interface VI interacts with the CompactRIO and displays data.
- Error handling throughout the application that reports and logs all errors from the CompactRIO device, shutting down on any critical error.

KEY CONCEPTS DEMONSTRATED

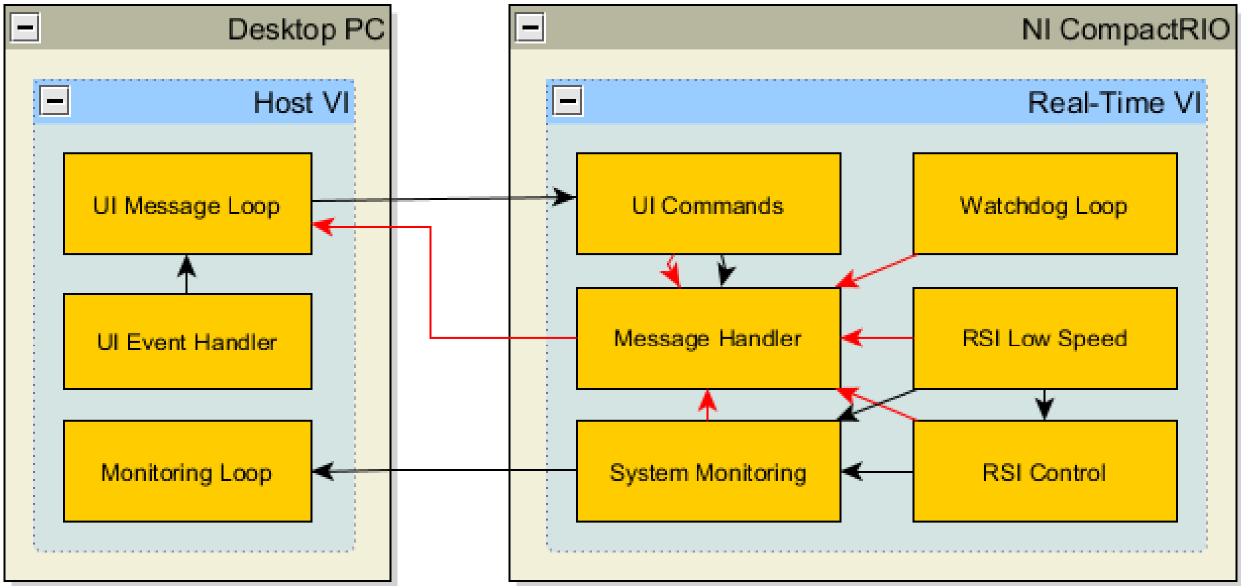
- RIO Scan Interface
- Deterministic Control
- Network Communication
- State Machine
- Queued Message Handler
- Built-in PID libraries

SCENARIO

Customer applications often require an RTOS for determinism and reliability, or they are looking to implement standalone functionality with the ability to shutdown or place I/O in a safe state on its own. This application shows variable control rate tasks running in parallel loops on the real-time controller, low-speed for temperature monitoring, and high-speed for PID control. For this demo, we are simulating a plant through a loopback test, but the customer’s plant could be wired to the terminal block as described in step two of the Set-Up Procedure.

A Real-Time Watchdog loop is in place to set I/O at a safe state, ensuring the application continues to run at the chosen watchdog “pet” rate. Finally, if the user chooses to view the application that is running headlessly, they can run a User Interface VI on the Desktop to connect to the CompactRIO to display I/O and system status data.

Application Diagram



Note: Red arrows are for error handling

Theory of Operation

This sample project consists of nine parallel loops across two execution targets, the Desktop and the CompactRIO. The following loops run in parallel on the desktop computer:

- Handling events from the user interface (UI Main.vi - Event Handling Loop)—Produces messages to the UI Message Loop based on front panel events.
- Handling messages from the user interface and other loops on the desktop computer (UI Main.vi - UI Message Loop)
- Displaying messages and data from the CompactRIO device (UI Main.vi - Monitoring Loop)—Displays the latest values of information acquired from RT Loop - System Health and Monitoring.vi.

The following loops run in parallel on the real-time controller:

- Controlling the plant (RT Loop - RSI Control.vi)—Reads from the NI 9381 inputs, applies a control algorithm to each channel, and writes to the NI 9381 outputs. The control algorithm uses setpoint data and PID gains sent from the user interface.
- Monitoring temperature and stopping the control loop (RT Loop - RSI Low Speed.vi)—Reads RTD and TC temperature data from the NI 9219 and a SW0 DI line from the NI 9411 and determine whether to stop the control loop and write to LED DO lines on the NI 9401.
- Handling commands from the user interface (RT Loop - UI Commands.vi)—Reads commands that are sent from UI Main.vi on the development computer and produces the appropriate messages.
- Handling messages from all loops on the real-time controller (RT Main.vi - Message Handling Loop)—Consumes messages from all loops that run on the real-time controller.
- Ensuring the RT controller remains responsive (RT Loop - Watchdog.vi)—Pets the watchdog, ensuring the RT controller remains responsive.

- Monitoring diagnostic information from the real-time controller (RT Loop - System Health and Monitoring.vi)—Monitors CPU and memory usage of the real-time controller and information about the RSI. This information is written to network-published shared variables and appears on the **Data Monitoring** and **System Monitoring** tabs of UI Main.vi.

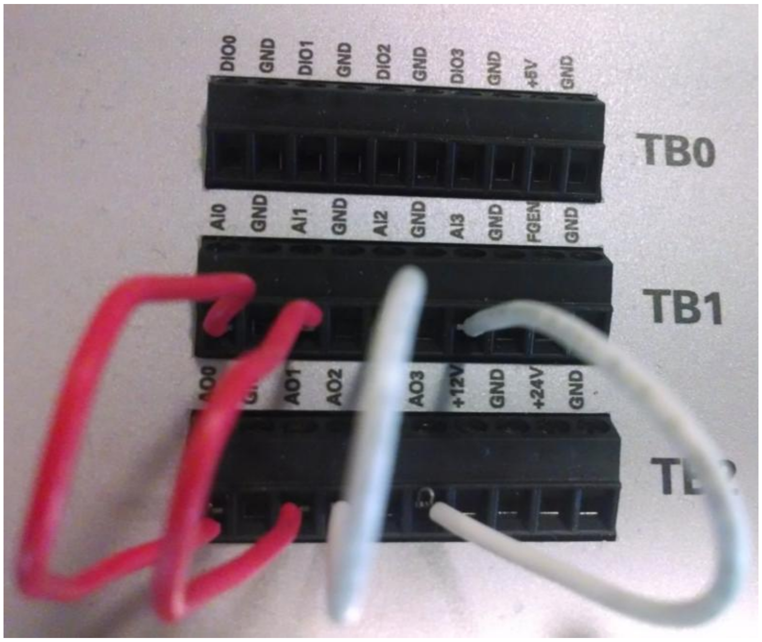
No LabVIEW code runs on the FPGA. Instead, the RSI runs on the FPGA, scans the C Series modules for input and output changes, and passes updated values to the NI Scan Engine. The NI Scan Engine, which is running on the real-time controller, updates the I/O variables with data from the RSI. This update provides single-point data from C Series I/O to both the real-time VI and the user interface.

Additional Reading on the RSI:

- The RIO Scan Interface Under the Hood
- Using NI CompactRIO Scan Mode with NI LabVIEW Software

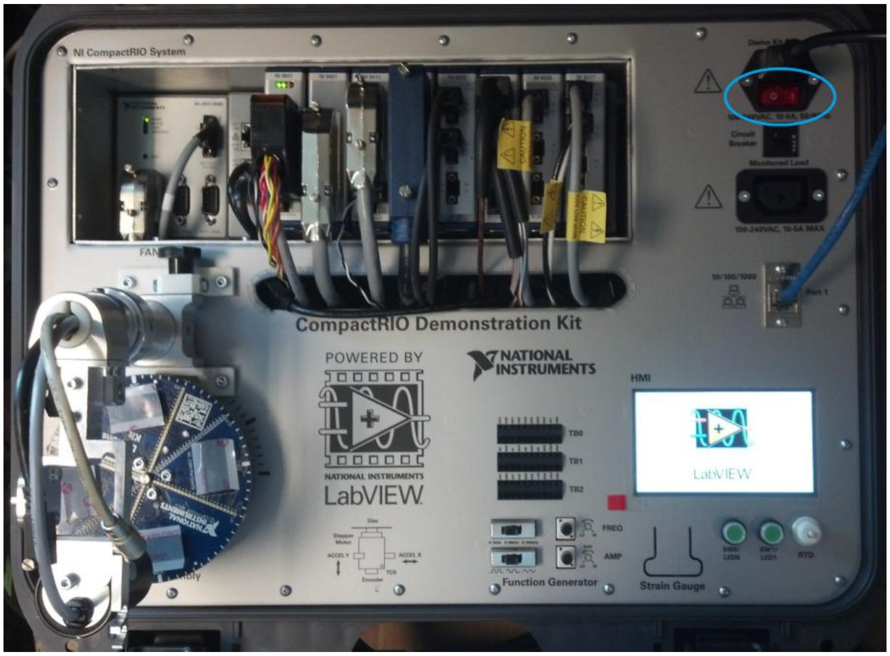
Set-up Procedure

1. Ensure that all required software is installed. (see Required Software)
2. Create the following hardware I connections for basic loopback, OR insert a physical "plant" here.



TB 1	TB2
AO0	AI0
AO1	AI1
AO2	AI2
AO3	AI3

3. Using region specific power cable, plug in and turn on switch located in upper right hand corner of Demo Kit.

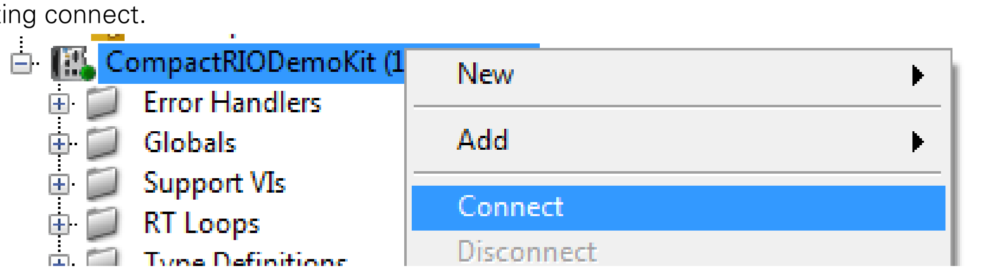


4. Connect an Ethernet cable directly from your computer to Port 1 on the CompactRIO Demo Kit, or connect Port 1 on the CompactRIO Demo Kit to your network infrastructure.

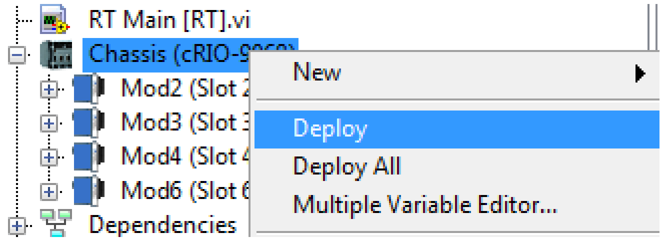
5. Create an instance of **LabVIEW Real-Time Control on CompactRIO** demonstration by selecting it in the **Create Project** dialogue in LabVIEW, under **CompactRIO Demonstration Kit**.

6. In the new project instance, right-click on the CompactRIODemoKit, select properties and configure the IP address to match that of your specific cRIO Demo Kit.

7. Connect to the target by right clicking the CompactRIODemoKit item in the project and selecting connect.



8. Right-click on the chassis item under the CompactRIODemoKit item in the project and choose Deploy.

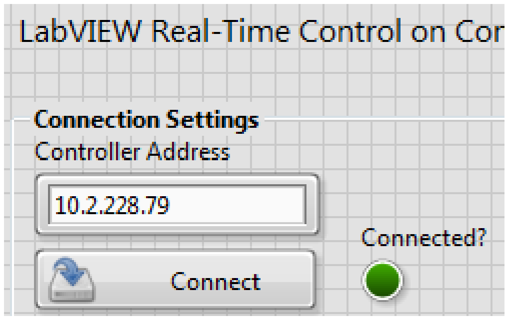


9. Open **RT Main.vi** under the CompactRIODemoKit and **UI Main.vi** under My Computer.

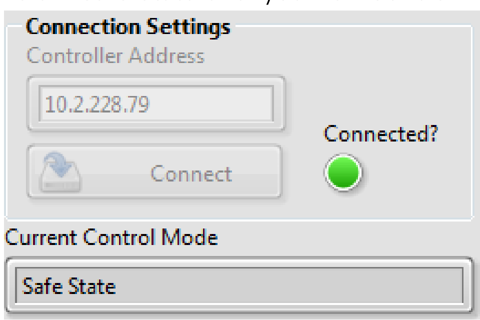
DEMO PROCEDURE/SCRIPT

1. Run **RT Main.vi** and **UI Main.vi**

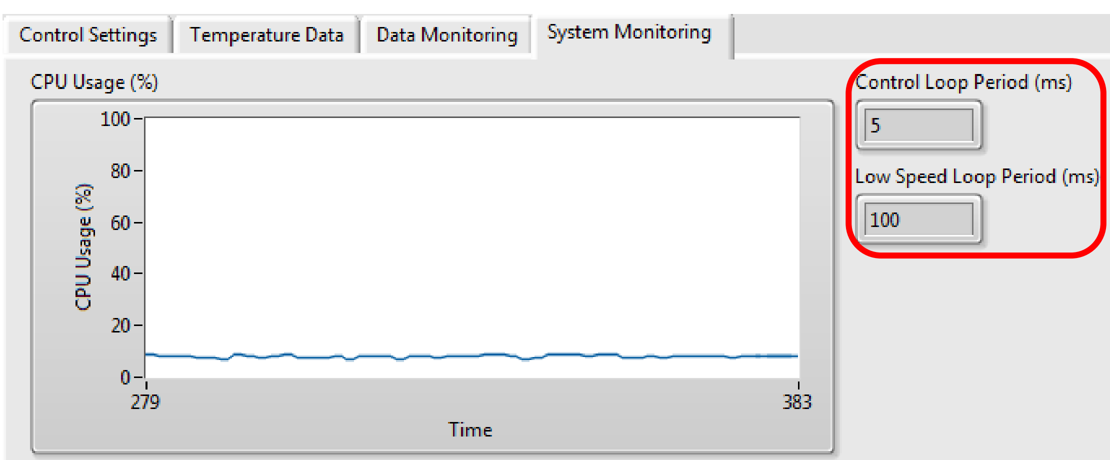
2. On the UI Main.VI front panel, enter in the IP Address of the CompactRIO into the **Controller Address** string control.

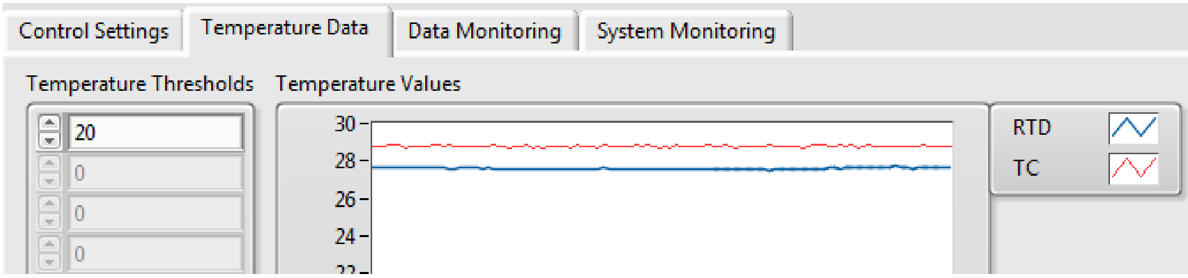


3. Press the **Connect** Boolean control to initialize the network streams and network published shared variables. The **Connected?** Boolean indicator should light up true and Connect Button will be disabled. The target will wait in safe state until you Run Control Mode.



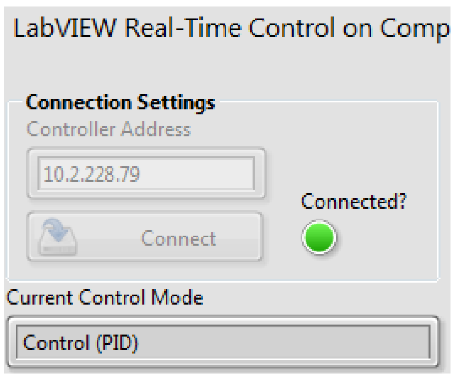
The cRIO is now monitoring the CPU Usage, Free Physical Memory, Control Loop Period, Low Speed Loop Period, ambient Temperature, and Control Input and Output Values. You can view these parameters by switching the tab control to **Temperature Data, Data Monitoring,** or **System Monitoring**. Emphasize the difference in loop period for our parallel tasks.

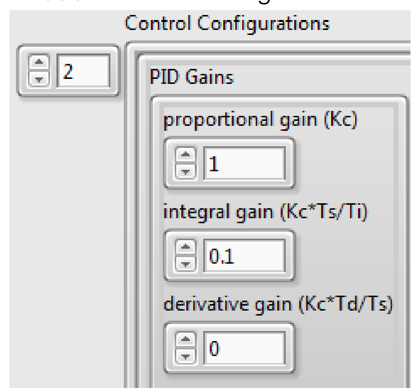




Note: you can set thresholds on the temperature to light up the **Any Above Limit?** Boolean and LED 1 on the kit for alarm functionality.

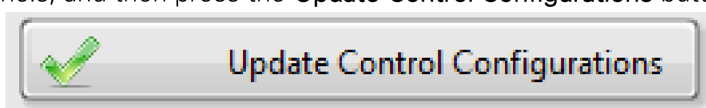
4. Click **Run Control** on the Control Settings tab to start the control loop. Increment the Channel **Setpoints** to desired values. Note the **Current Control Mode** indicator changes.



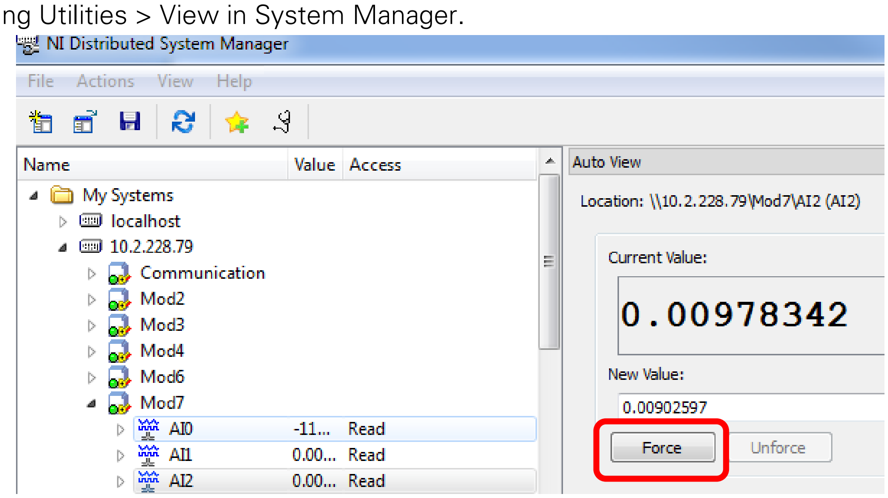


Note: The default PID Gains should be as follows: K_c=1, K_i=0.1, K_D=0. You can change this in the Initialize case of Functional Global - Control Configurations.vi

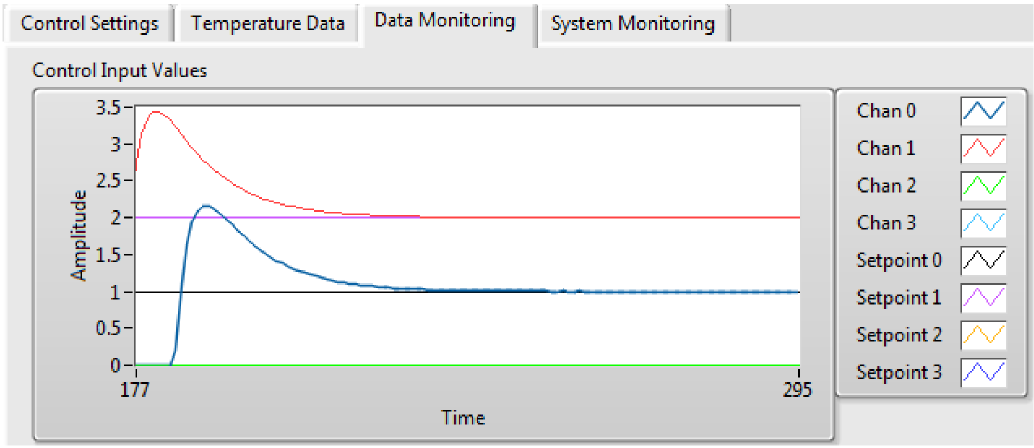
You need to move to the next element in the Control Configurations Array to set new values for each of the 4 channels, and then press the **Update Control Configurations** button.



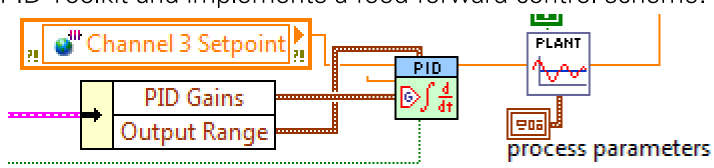
5. If you would like to set manual values for the control inputs, you can force IO values through the NI Distributed System Manager, available by right-clicking the CompactRIO target in the project and choosing Utilities > View in System Manager.



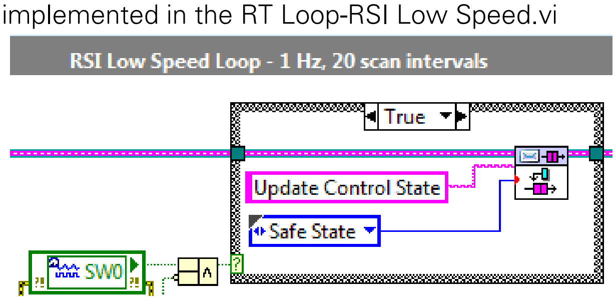
6. View the outputs on the **Data Monitoring** tab. All channels have a default setpoint of 0 to not clutter up the graph for the initial demo. Adjust the values on the Control Settings Tab to see up to all 4 channels run control.



7. The simulated plant and PID control takes place in the **RT Loop-RSI Control.vi**. The **Plant System.vi** was taken and modified from the General PID Simulator.lvproj Shipping Example installed with the PID Toolkit and implements a feed-forward control scheme.

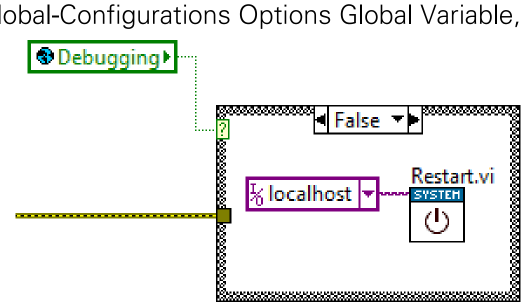


8. Press SW0 on the demo box or force it in Distributed System Manager to manually put the I/O into safe state. This is implemented in the RT Loop-RSI Low Speed.vi



9. If there are any errors in the application, an error log is kept at c:\logs\log.txt. This default logging location can be changed by modifying the Global-Configurations Options Global Variable. Use the Network Browser or follow [these instructions] to use WebDAV read the file off the cRIO.

10. If you wish to build the real-time side into an executable, make sure to set the Debugging Boolean to False in the Global-Configurations Options Global Variable, so the target restarts.

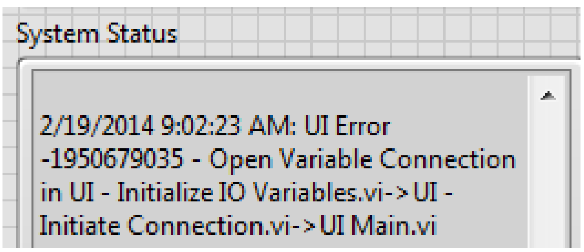


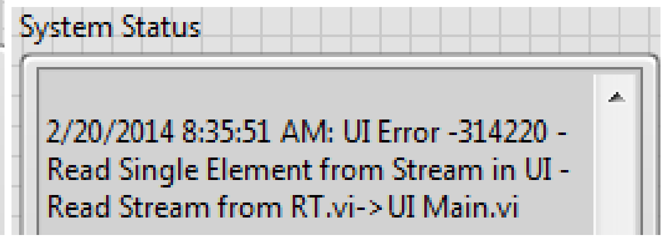
SHUTDOWN PROCEDURE

1. Stop the UI Maini.VI by pressing the **Switch Target to Safe State** button, if it isn't already in Safe State, and then **Exit**. Press the abort button on the RT Main.VI if you no-longer want it to run in interactive mode.
2. Switch off the main power switch.

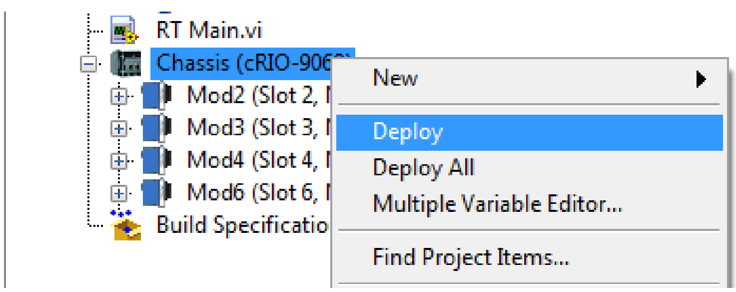
TROUBLESHOOTING STEPS

ERROR UPON RUNNING THE UI MAIN.VI:





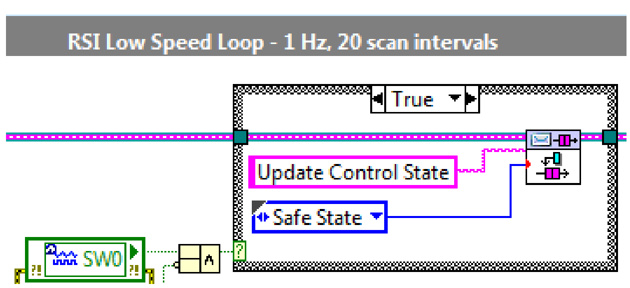
Variable Error 1950679035 or Network Streams error 314220 could mean that one of your modules isn't deployed or your controller is not connected. Check that your Demo Kit is connected in MAX and then right-click in the project on the **Chassis** and select **Deploy** to ensure your modules are deployed.



If it still doesn't work, reset the controller and close and reopen the project.

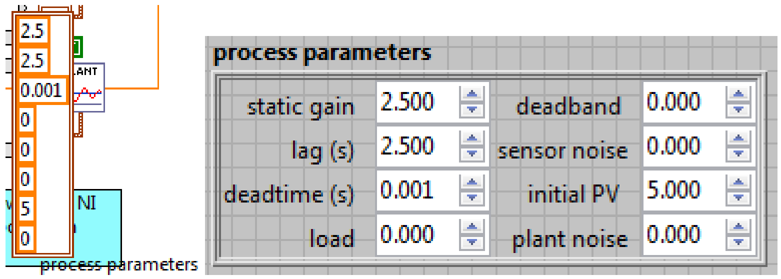
CURRENT CONTROL MODE KEEPS CHANGING TO 'SAFE STATE'

Check that your SW0 is not compressed; this is the manual force to safe state. As seen in the RSI Low Speed.vi. Use Distributed System Manager to Force it to False if necessary.



Control Parameters Seem "Off" or You Want to Add Noise to the System

Make sure you've pressed **Update Control Configurations** on the UI Main.vi Front panel and that each of the channel PID Gains have appropriate values. To tweak Plant System.vi, double click on the **process parameters** input cluster, or open the subVI and manipulate the values on the Front Panel.



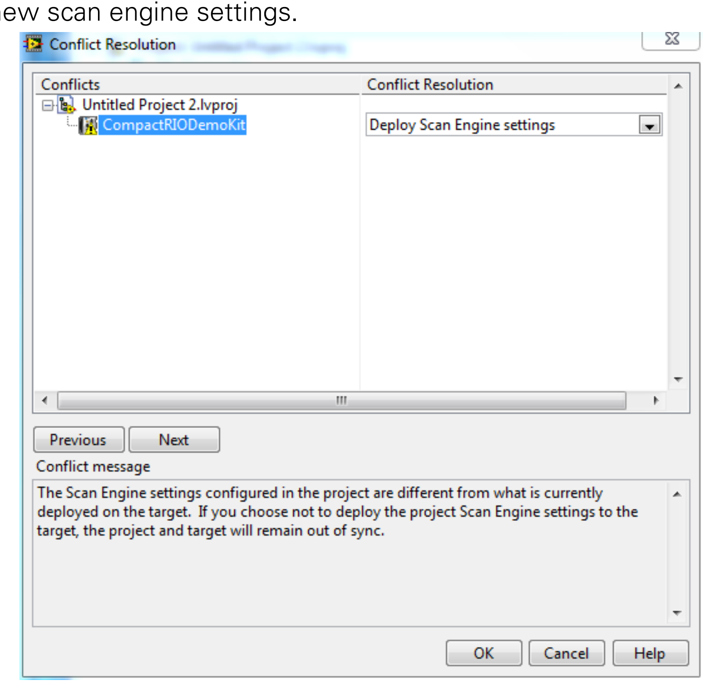
Error Deploying Project – Scanned I/O Bus Not in Required I/O Mode

LabVIEW: The operation cannot be completed because one of the scanned I/O buses is not in the required I/O mode.

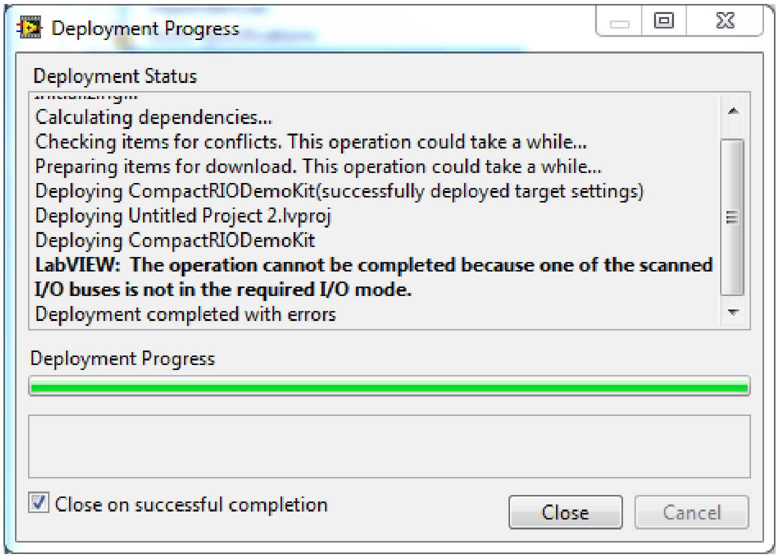
This error occurs when a previous demonstration or project used the scan engine with a Softmotion Axis. Many of the demos, templates and sample projects have a Softmotion Axis included by default, so this error will occur with some frequency. This issue can also manifest with error -77077 "*NI SoftMotion: The requested operation is not possible when the NI Scan Engine is in Active mode*."

The error stems from the fact that the scan engine needs to be placed into configuration mode in order to deploy the new scan engine settings, but the Softmotion Axis is deployed and Active, preventing the engine from changing states. The solution is to use the distributed system manager to manually set the engine to configuration mode, and then redeploy the project.

When you deploy a project and encounter this error, often you will first see a *Conflict Resolution* dialogue asking you to deploy new scan engine settings.



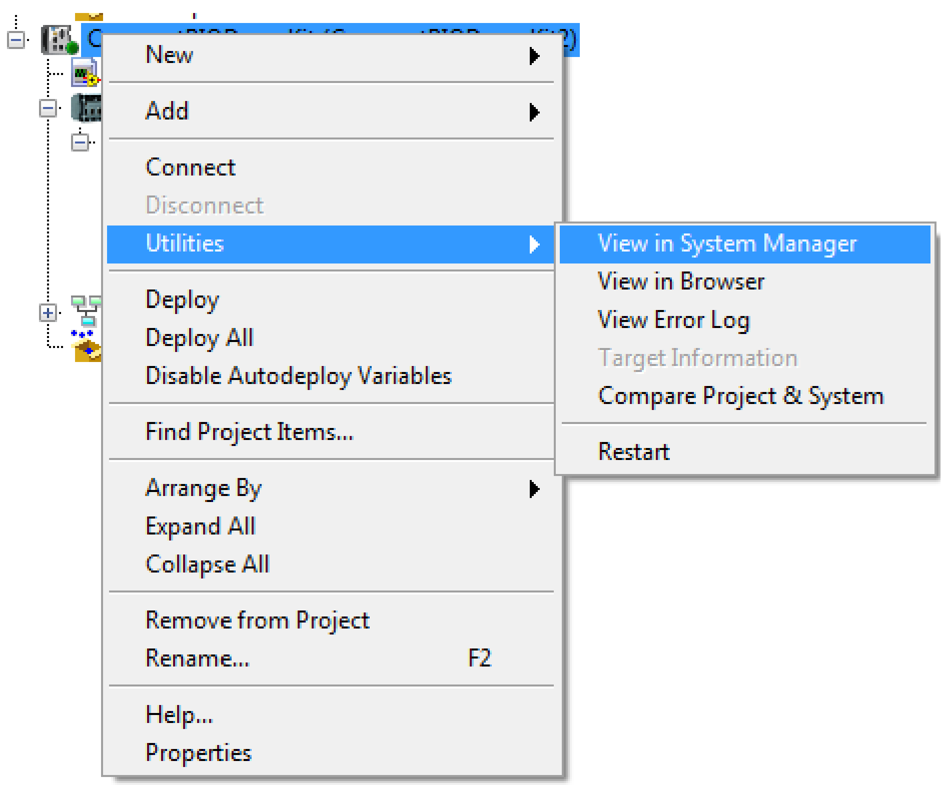
However, the deployment will fail with the following error:



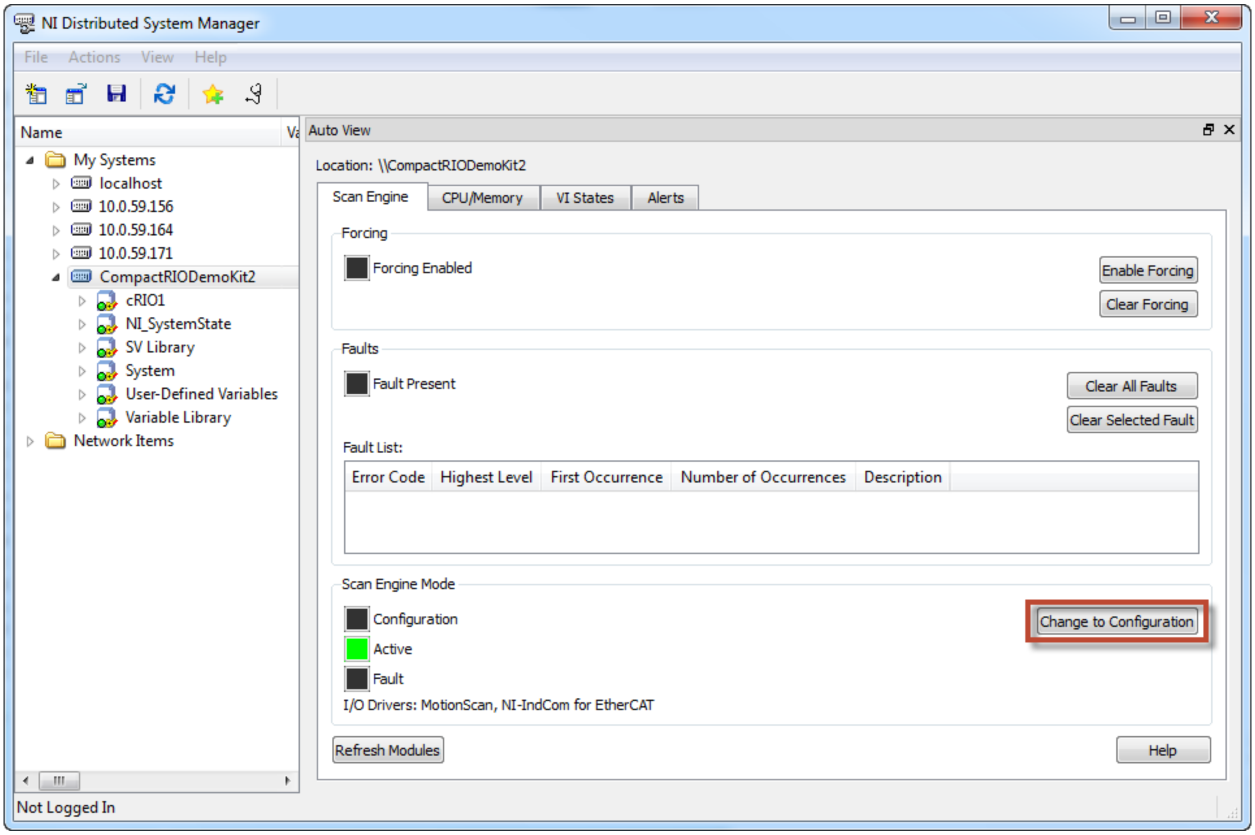
To fix this error and deploy the project, follow these steps.

1. Close the *Deployment Progress* dialogue box.

2. Open the Distributed System Manager by right clicking the CompactRIO target in the project and choosing Utilities >> View in System Manager.



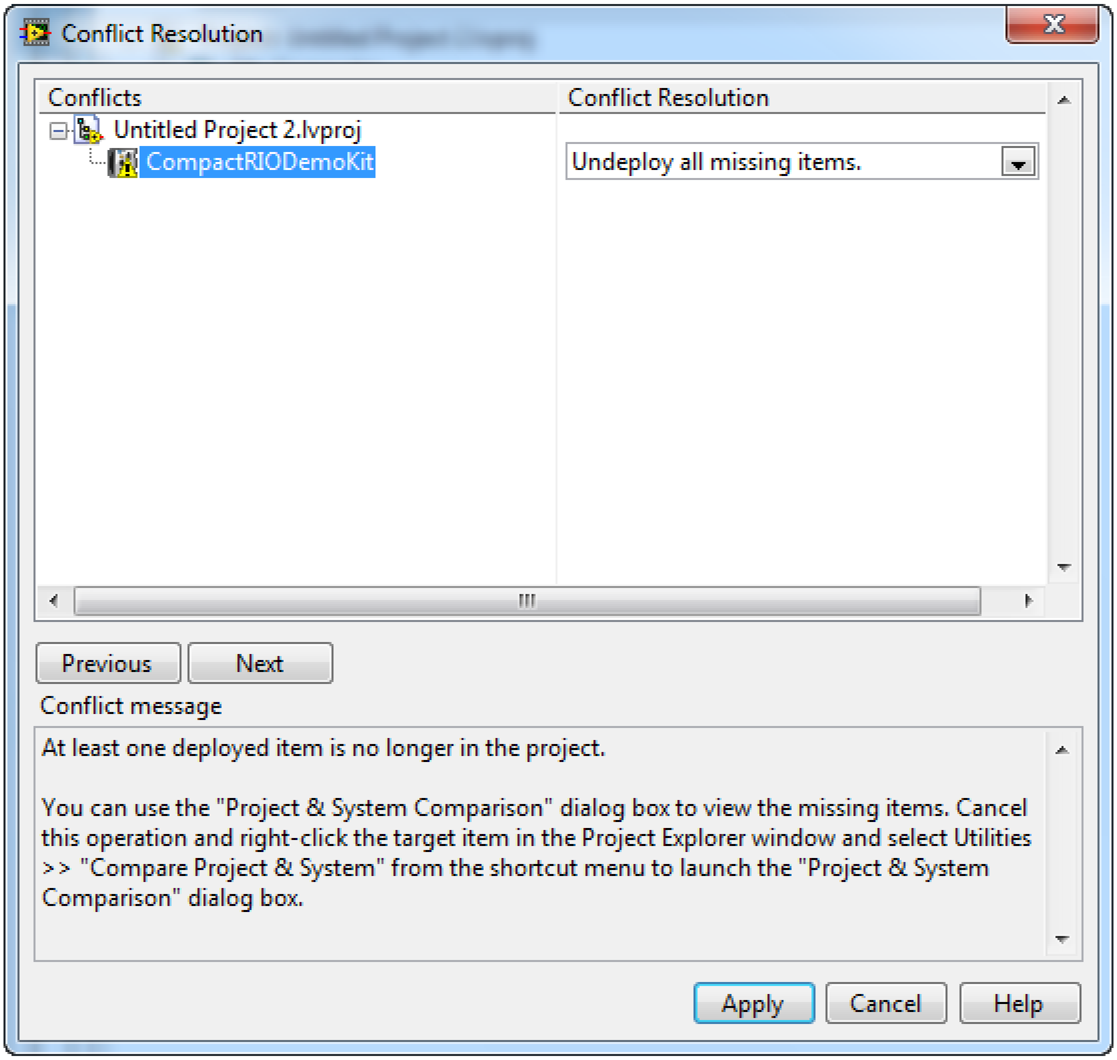
3. In the Distributed System Manager, ensure that the CompactRIO Demonstration Kit is selected and that the *Scan Engine* tab is open. Press the **Change to Configuration** button to force the scan engine into configuration mode.



4. When you confirm that the Scan Engine Mode has changed to Configuration mode, close Distributed System Manager.

5. In the project, re-deploy the project or run the RT VI.

6. You will see another *Conflict Resolution* dialogue box. Press **Apply** in this dialogue.



7. Another *Conflict Resolution* dialogue box will appear asking you to switch the Scan Engine Mode to Active Mode. Choose **Apply** again.

8. Your project should successfully deploy and any Softmotion Axis should be correctly removed. If you have additional problems try rebooting the system and follow this guide again.