Analytics and Machine Learning

Technical Manual for Compact RIO Demostration Kit

Version 1.0.0, July 2017

Owner Contact Information

Name: Matthew Bollom

Email: matthew.bollom@ni.com

Address: 11500 N. Mopac Expy

Austin, TX 78759

NATIONAL INSTRUMENTS

Table of Contents

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

- 1.0

Demonstration Kit Components Used

- NI 9503
- NI 9401
- NI 9411
- NI 9232
- NI 9219
- Rotary disk assembly

Required Hardware Components

- Region Specific Power Cable
- Ethernet Cable

Required Software

- LabVIEW 2017 or newer
- LabVIEW FPGA
- LabVIEW Real-Time
- LabVIEW Analytics and Machine Learning Toolkit 2017

DESCRIPTION

This demo showcases the ability of a CompactRIO to deploy machine learning models for predictive analytics.

GOAL OF DEMONSTRATION

This application shows how to efficiently implement a predictive analytics application using machine learning with the typical machine learning process. Sensory data is acquired, features extracted, and then deployed in a machine learning model to detect anomalies from baseline data and to classify the data into specific fault conditions. This demonstration will show how to train and deploy these machine learning models.

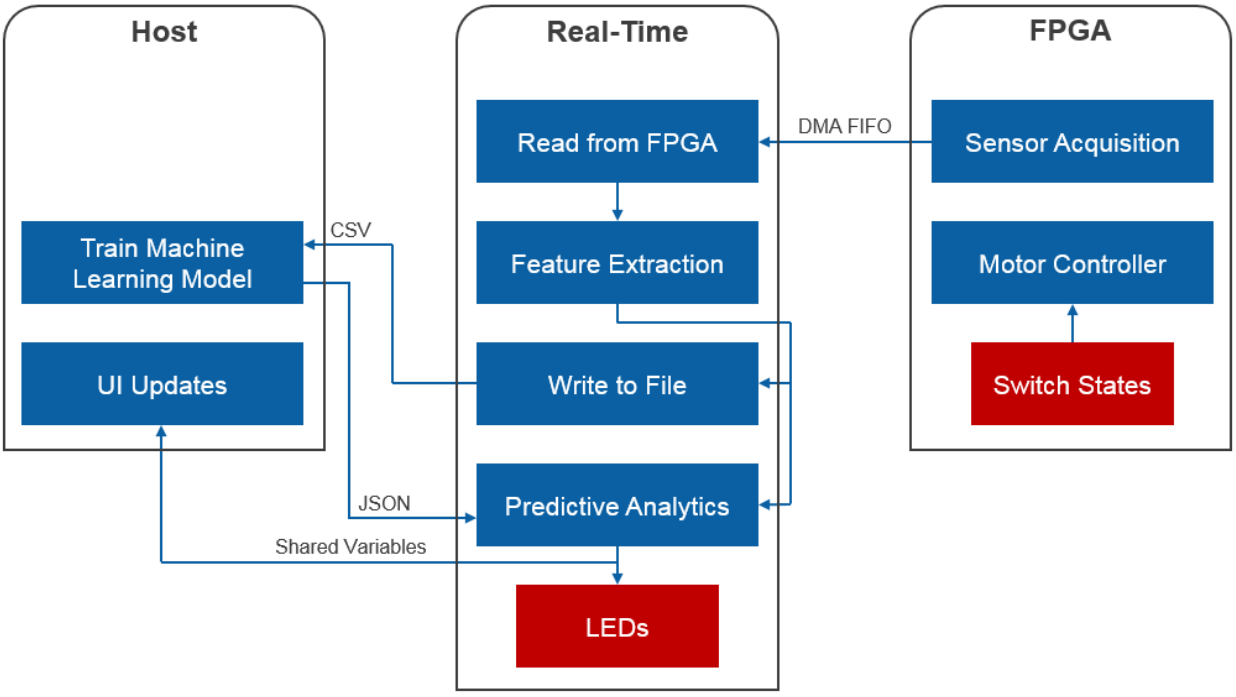
KEY CONCEPTS DEMONSTRATED

- Machine learning workflow and algorithms
- Communication between FPGA, Real-Time, and host
- Vibration measurements
- File I/O
- Motion control

SCENARIO

This demonstration showcases a monitoring application with predictive analytics in which we want to receive early warning of fault conditions and anomalies for. In this case, the rotary disk assembly serves as our equipment and vibration and electrical signals are monitoring the assembly. When the equipment begins to deviate from the baseline behavior, we want to be notified of the problem so we can respond appropriately. To configure this warning system, we will use machine learning to efficiently learn the normal state of the equipment and to configure the warnings automatically.

Application Diagram

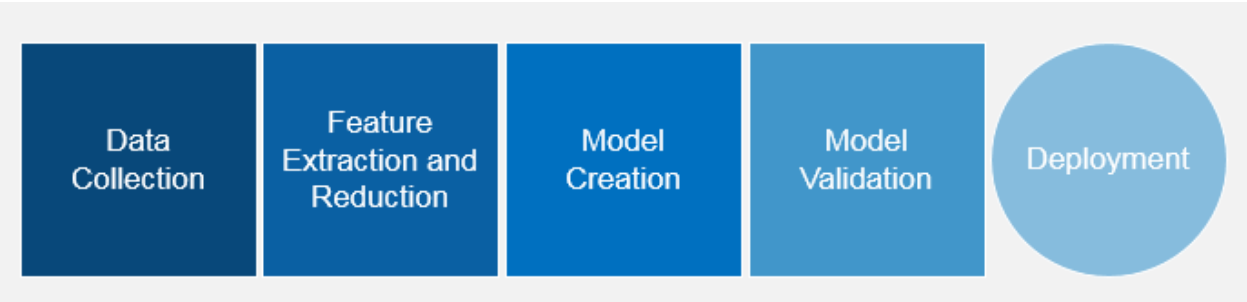


Theory of Operation

The rotary assembly serves as our asset in this demonstration. The rotary disk spins at a steady speed (approximately 6 RPS) and the speed is modulated to create different operating conditions that represent anomalies. Sensory data from two accelerometers mounted to the bottom of the rotary assembly, the current draw by the two motor phases, the motor voltage supply, and the thermocouple attached to the motor is continuously collected and processed (if the motor is working harder, we should see increased vibration and current draw). Rather than trying to implement rules to detect various states of the asset, we will develop two machine learning models:

- Anomaly detection: an unsupervised learning method which develops a baseline model based on a set of data and can provide a health index in which you can compare against a threshold to detect anomalies.
- Classification: a supervised learning method in which you provide data that has been labeled so you can predict the label of a future piece of data. This is useful for performing diagnostics on data.

The creation of this model and demo workflow follows this process:



Data Collection: the raw data collection is handled by the FPGA. Waveform data from the accelerometers and capture of the motor current waveforms is acquired and sent to the real-time processor via a DMA FIFO.

Feature Extraction and Reduction: the raw data is then processed in real time to extract features, or data that describes the waveform. While machine learning can be executed on waveform data, features are usually preferred because it describes the data and reduces the amount of data computation that is required in subsequent steps. In this demonstration, standard features for the sensors have been selected:

- Accelerometer: RMS, Peak, Derived Peak, Peak-Peak, Crest Factor
- Motor current: RMS, Peak
- Voltage supply: RMS, Peak
- Thermocouple: temperature

A total of 17 features are calculated for each one second block of data. While the amount of data has been significantly reduced, not all features may provide value for analytics and are typically processed using a feature reduction algorithm that combines features together to reduce the number of dimensions of the data. For the classification model in this demo, the features are reduced from 17 dimensions to 2 dimensions.

Model Creation: after the features are processed, the data goes through a training process to develop a model. While the process has been automated in this demonstration, this process usually requires you to select a specific algorithm and to train this algorithm on your test data. This step usually requires heavy compute power and should be done on a system with plenty of resources available.

Model Validation: after a model has been trained, you need to test it against other data to ensure it addresses the application problem. If it doesn't, go back to the model creation step and iterate.

Deployment: when a model has sufficient performance on the data, this model can be deployed to an embedded target for immediate detection of fault conditions.

In this demonstration, the data acquisition and feature extraction processes happen continuously. Training data is collected on demand from the user and is saved to a CSV file on the cRIO before being transferred to the host for training. After training, the models are deployed to the cRIO where they are deployed against new data for immediate detection of anomalies and to diagnose the equipment state.

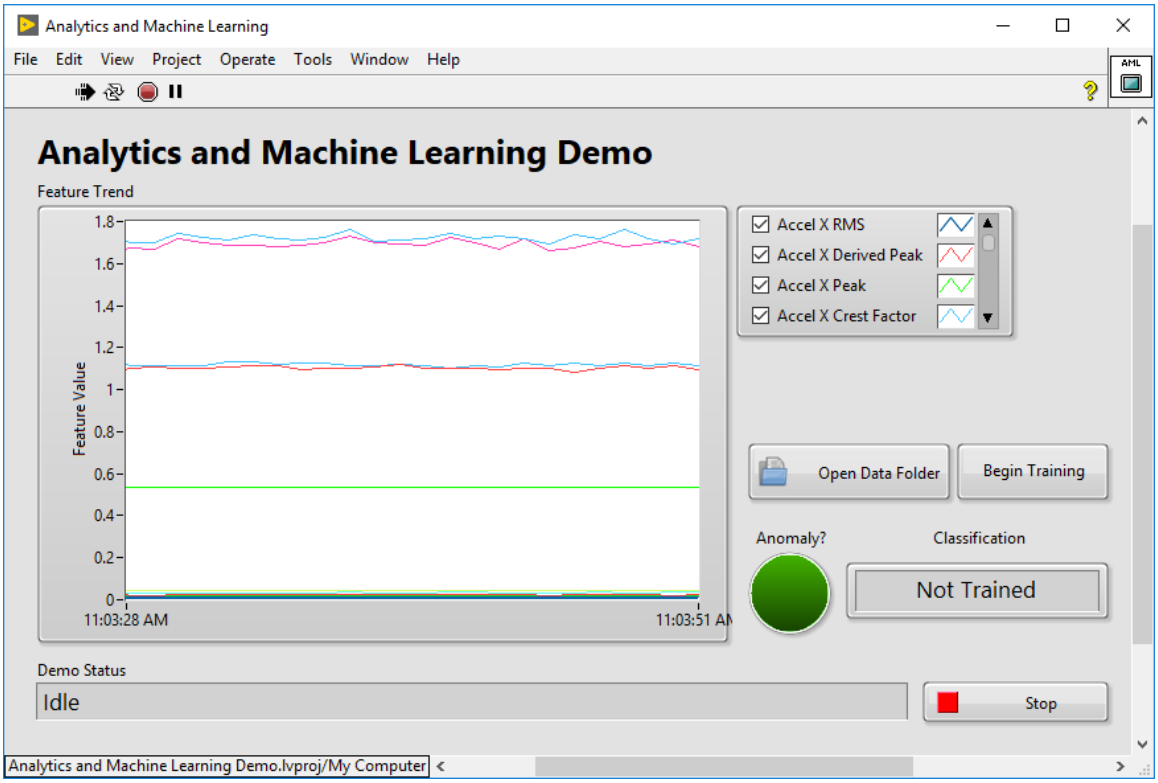
SET-UP PROCEDURE

1. Ensure that all required software is installed. (see Required Software)
2. Create any hardware connections required.
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. Ensure that both SW0 and SW1 are not depressed (they should be in the UP position).
6. Creat/e an instance of the Analytics and Machine Learning demonstration by selecting it in the **Create Project** dialogue in LabVIEW.
7. 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.
8. Connect to the target by right clicking the CompactRIODemoKit item in the project and selecting connect.
9. Right-click on the chassis item under the CompactRIODemoKit item in the project and choose Deploy.

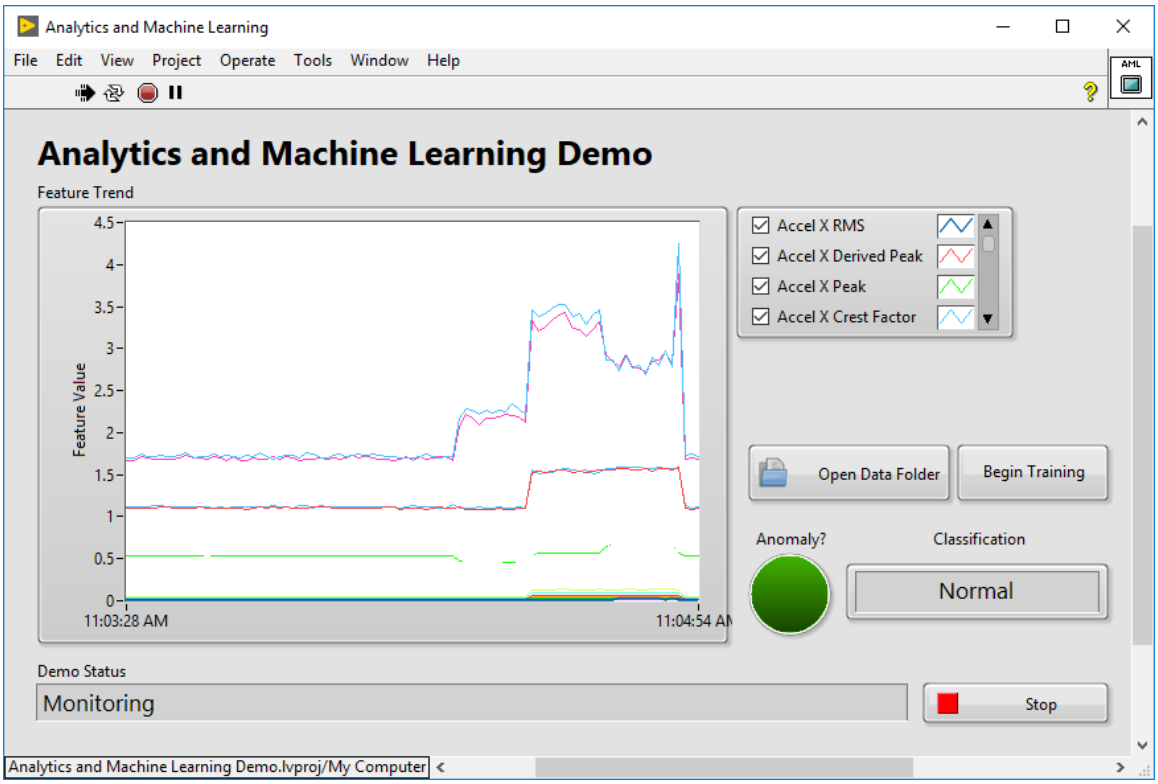
DEMO PROCEDURE/SCRIPT

1. Connect any required hardware and connect to the CompactRIODemoKit in the LabVIEW project.
2. Open **AML [RT].vi** and run the VI. The disk will begin to spin at 6 revolutions per second (RPS).
3. Open **AML [Host].vi** and run the VI.
4. Explain to the customer that the rotary assembly represents a piece of equipment (such as a motor) running at steady state. We want to monitor this equipment to receive warnings when the equipment is not running as it normally is and to receive a diagnosis of any faults that occur. Highlight the sensors that are used as described in the theory of operation.
5. By this time, data should be appearing in the trend viewer on the host VI. Highlight that the raw waveform data that was acquired by the FPGA was processed into 17 different features. Because the motor is running at steady state, the feature trends are relatively constant. (NOTE: The default configuration of the chart hides the data for Vsup and motor temperature because they are much larger than the rest of the values. If you want to show these values, scroll to the bottom of the legend and enable those plots.)



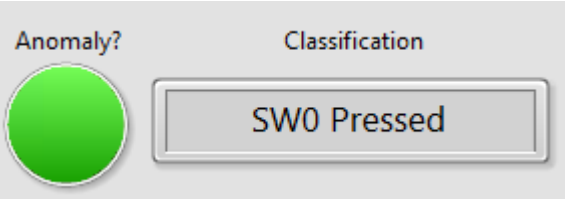
6. Now that we have features, we need to train a model. Click **Begin Training**. This will send a command to the target to begin storing data in a file that we can then retrieve and train our model against. Explain to the customer that the training process will cycle through each of the states of the motor to give us the data we need to train a classification model for future diagnosis of the motor state. Each state will be enabled for 10 seconds. The motor will sound

different for each state and you will see visible difference in the feature trend.



7. Click **Open Data Folder** to open Windows Explorer to the folder where the training data is stored in a CSV file. Open the file in Excel and walk the customer through the contents of the file – each row is a data point while the last column is the label (there are 4 labels, one to represent each state). This data is the same that we saw on the feature trend in the UI.
8. Explain that behind the scenes, this data file was processed and two types of machine learning training occurred:
    a. Anomaly detection: the baseline data (data with label 0) is extracted and trained against an anomaly detection model. This model will output a health value, which we can threshold to determine of the asset is running as we expect.
    b. Classification: all the data goes through a classification algorithm to allow us to diagnose future data. The data is first normalized and the number of features reduced using some machine learning algorithms. The classification model is then trained. The output of the classification model is a predicted label.
9. Highlight that the training was done on the host – training typically requires more processing power than is available on an embedded device. In a real-world scenario, you would typically have several months of data and use a server-grade system to process and train the data.
10. After the model has been trained, it is transferred back to the cRIO for deployment. The model parameters have been stored in a text file for ease of transport – these are visible in the json files next to data.csv.
11. Now that the model has been deployed, let's test the model. Interject a fault into the system by pushing SW0. The following will happen:
    a. The motor speed will be modulated from 5 RPS to 6 RPS, so it will sound different.
    b. The feature trend should change because the speed is now changing.

c. The anomaly indicator should light up with the classification reading "SW0 Pressed". While this diagnosis is demo-specific, a real-world pump might have diagnoses of "cavitation", "misalignment", "low flow", etc.
d. LED0 will light up, indicating that a fault has been detected in the hardware.



12. Review what just happened with the audience – we interjected a fault in the system and based on the sensory data we could detect that the motor was not running in its expected state. Because we had data for the different motor states, we also know exactly what is causing the problem. Without the diagnosis, we would typically need to send someone out to inspect the motor.
13. Press SW0 again so it returns to the upright position. The motor should return to the steady speed and the anomaly will disappear.
14. Press SW1. This time, the speed is modulated from 6 RPS to 7 RPS. An anomaly will be detected and classified as "SW1 Pressed" with LED1 on.
15. Press SW0 so both SW0 and SW1 are pressed. The speed is modulated from 5 RPS to 7 RPS and the diagnosis should be "Both Switches Pressed".
16. After the customer understands how this is working, ensure both switches are up and stop **AML [Host].vi** and disconnect the cRIODemoKit in the project. Remove the network cable. The motor should continue running. Explain to the customer that the model we trained has been deployed to the cRIO and is executing there. Press the switches and show how the LEDs still light up as we expect, showing that the machine learning model is executing on the target. Highlight a few use cases for this – we could change application behavior based on fault conditions, alert nearby employees with an indicator, and so on.

SHUTDOWN PROCEDURE

1. Reconnect the network cable and connect to the cRIODemoKit in the project.
2. Click **Stop** on both AML [Host].vi (if still running) and AML [RT].vi. The disk will stop spinning.
3. Switch off the main power switch and disconnect the network from the demo kit.

TROUBLESHOOTING STEPS

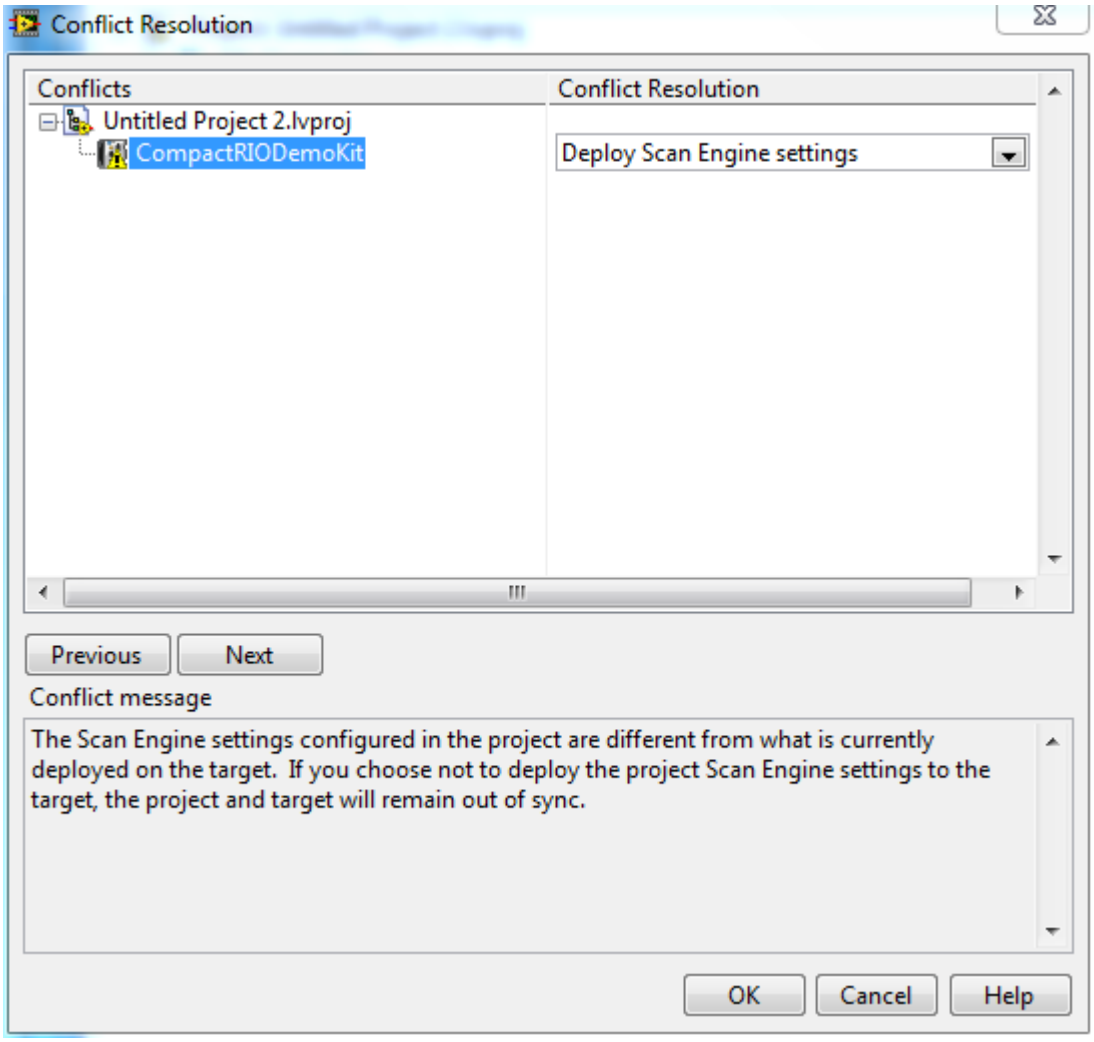
ERROR DEPLOYING PROJECT – SCANNED I/O BUS NOT IN REQUIRED I/O MODE

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

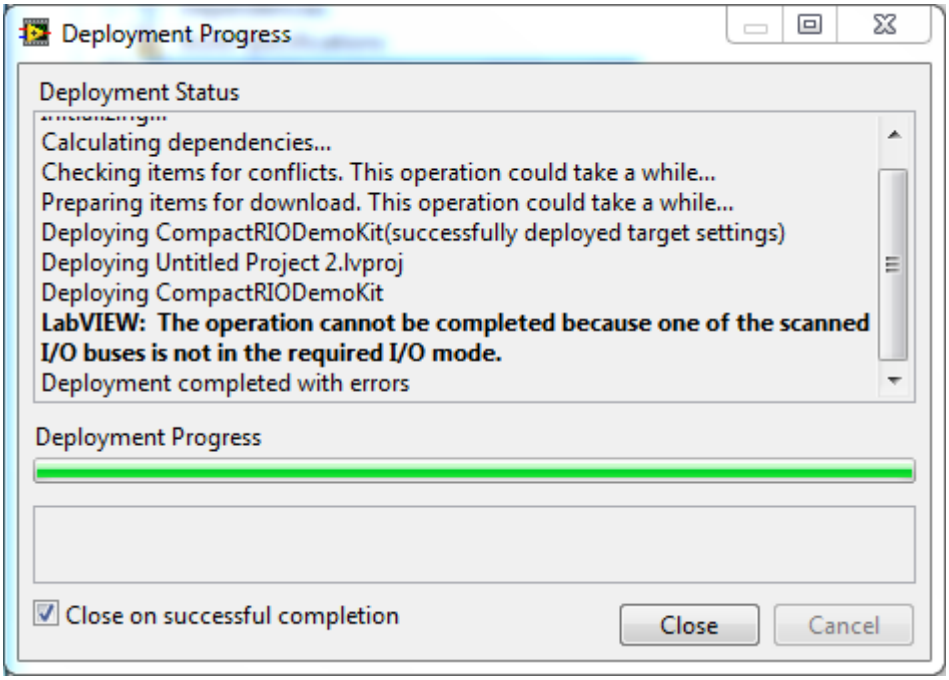
This error occurs when a previous demonstration or project used the scan engine with a Softmotion Axis. Many of the demos, templates and sample projects have a Softmotion Axis included by default, so this error will occur with some frequency. This issue can also manifest with error -77077 "*NI SoftMotion: 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.



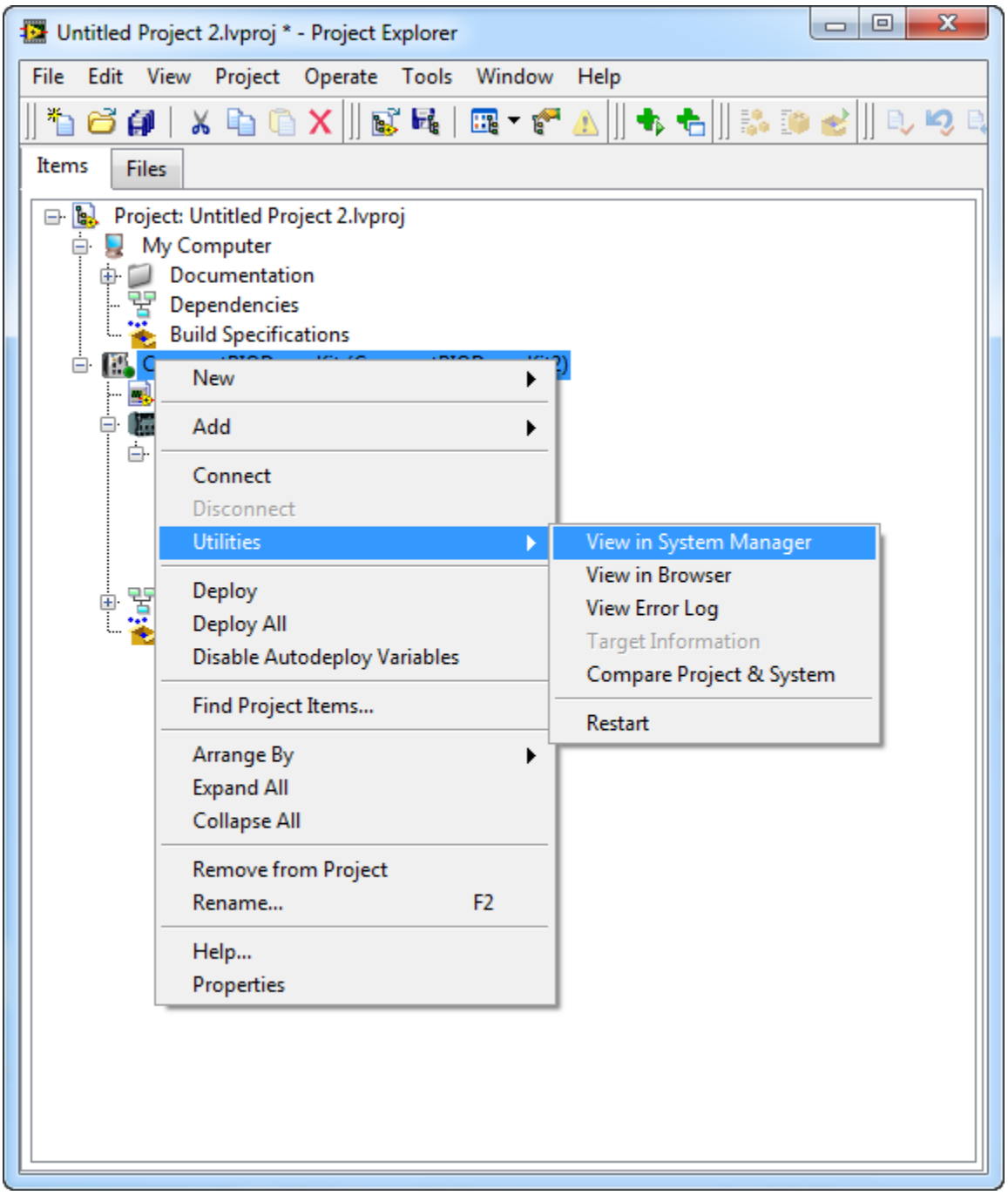
However, the deployment will fail with the following error.



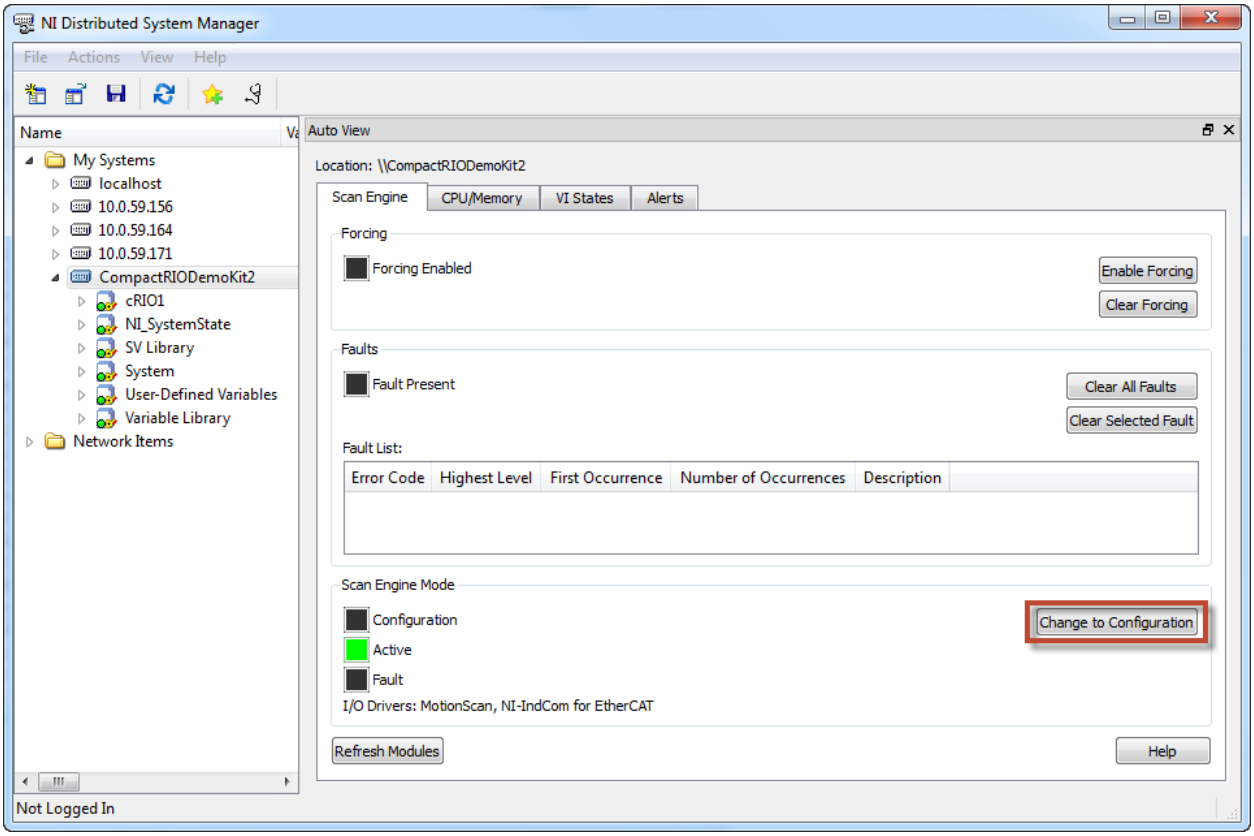
To fix this error and deploy the project, follow these steps.

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**.



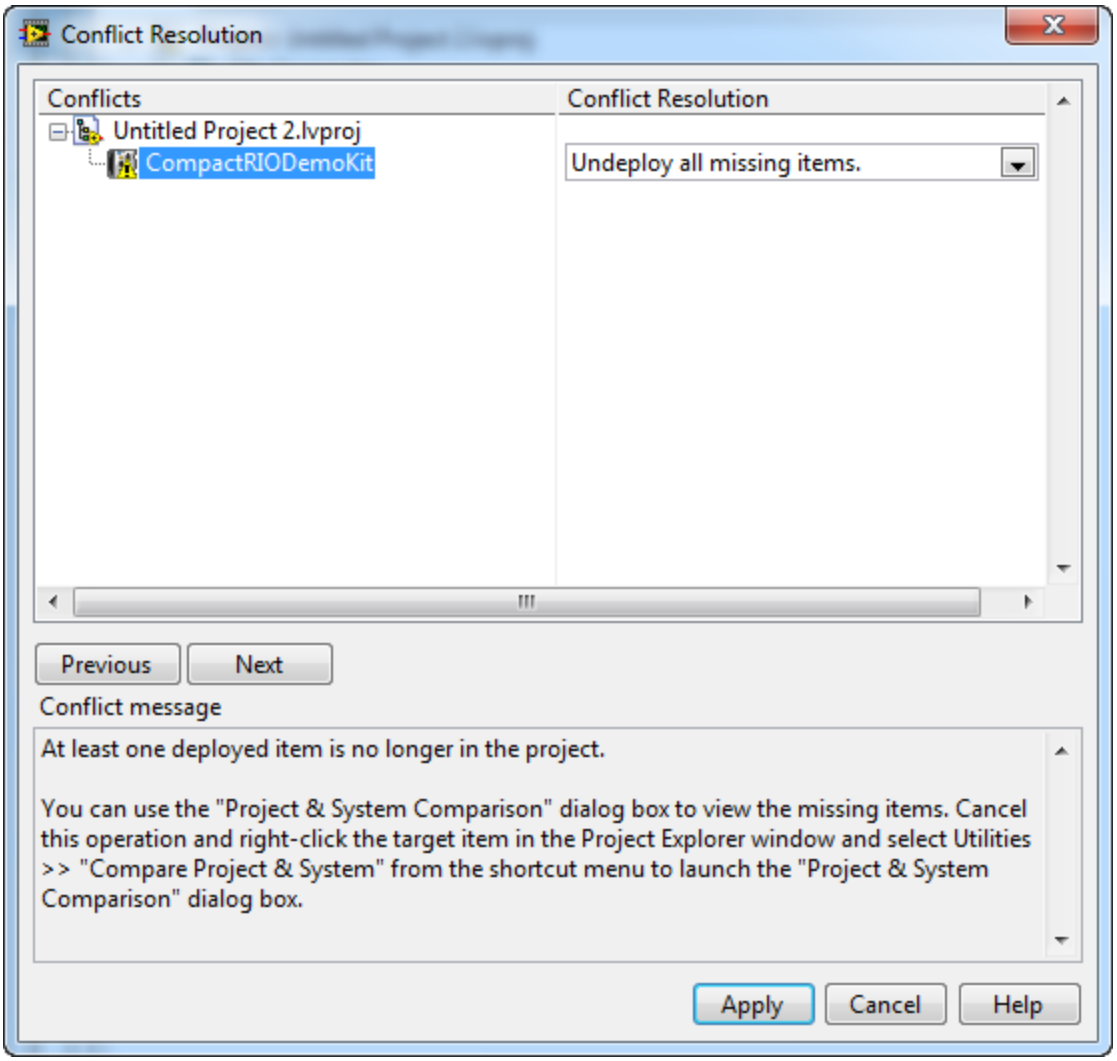
3. In the Distributed System Manager, ensure that the CompactRIO Demonstration Kit is selected and that the *Scan Engine* tab is open. Press the **Change to Configuration** button to force the scan engine into configuration mode.



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.