C Support for NI myRIO 1.0 User Guide

C Support for NI myRIO is designed for users who want to program the NI myRIO using the C programming language or a programming language other than LabVIEW.

The C Support for NI myRIO includes the following components:

- C Support—Contains headers, source files, and binaries for calling the NI FPGA interface and NI-VISA interface on the NI myRIO target.
- Examples—Contains examples files that invoke the C Support to use the resources on the NI myRIO target, which include accelerometer, AIO, DIO, encoder, I^{2}C, PWM, SPI, and UART.
- Template—Contains a blank project configured with required NI myRIO settings. The template provides an easy starting point for using the C support.

Requirements

- One of the following operating systems:
  - Windows 8 (32-bit and 64-bit)
  - Windows 7 (32-bit and 64-bit)
  - Windows Vista (32-bit and 64-bit)
  - Windows XP Pro (Service Pack 3)
  - Windows Server 2003 R2 (32-bit)
  - Windows Server 2008 R2 (64-bit)
- LabVIEW for myRIO Module, which provides drivers and software for setting up the NI myRIO target.
- Java x86 Runtime, which is required for Eclipse for NI Linux Real-Time.
- Eclipse for NI Linux Real-Time, which provides both the Integrated Development Environment (IDE) and compilation tools.

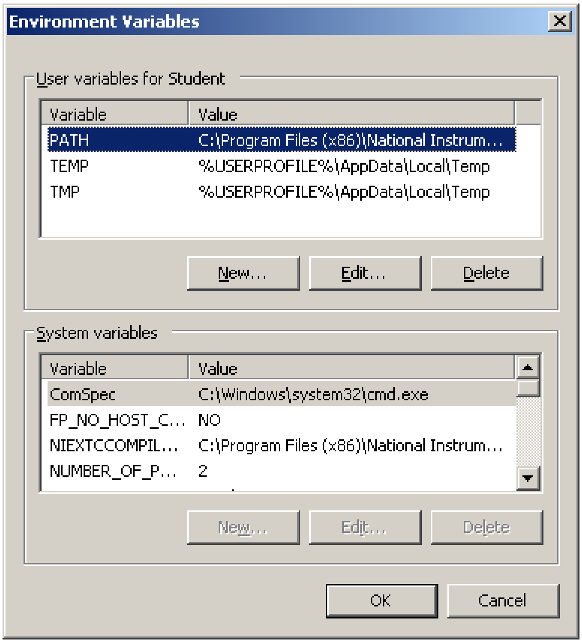
Installing and Configuring

You must perform the following tasks before developing an NI myRIO application.

1. Set up software environment on the host computer
2. Configure the NI myRIO target
3. Import the C Support for NI myRIO to Eclipse
4. Install FPGA bitfiles on the NI myRIO target

Setting up Software Environment

1. Install LabVIEW for myRIO Module.
2. Install Java x86 Runtime. Download Java from *http://www.java.com/getjava*.
3. Install Eclipse for NI Linux Real-Time. Download Eclipse for NI Linux Real-Time by navigating to *ni.com/info* and entering the info code `ex2cii`.
4. Add the compiler path to the system environment variables.
    a. In Control Panel, switch to the **Large icons** or **Small icons** view, navigate to **System»Advanced system settings**, and click **Environment Variables**. The **Environment Variables** dialog box appears.



Variable	Value
PATH	C:\Program Files (x86)\National Instrum...
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp

Variable	Value
ComSpec	C:\Windows\system32\cmd.exe
FP_NO_HOST_C...	NO
NIEXTCCOMPIL...	C:\Program Files (x86)\National Instrum...
NUMBER_OF_P...	2

    b. Select **PATH** in **User Variables** and click **Edit**. Click **New** to create one if **PATH** does not exist.
    c. Append the compiler path to the **Variable value**. The separator between paths is semicolon.

    64-bit Windows: `C:\Program Files (x86)\National Instruments\Eclipse\toolchain\gcc-4.4-arm\i386\bin`

    32-bit Windows: `C:\Program Files\National Instruments\Eclipse\toolchain\gcc-4.4-arm\i386\bin`

Configuring the NI myRIO Target

Complete the following steps to use the **Getting Started with NI myRIO** wizard to install software on the NI myRIO target:

1. Connect the NI myRIO to the host computer using a USB cable. The **NI myRIO USB Monitor** appears after the driver is installed.
2. Take note of the IP address of the NI myRIO target.
3. Click **Launch the Getting Started Wizard** and go through the wizard. The wizard installs all required software to the NI myRIO target and tests the onboard sensors.

After you install the software on the NI myRIO, use **NI Web-based Configuration & Monitoring** to enable SSH on the NI myRIO target by completing the following steps:

1. Navigate to `http://<IP Address>/` in the web browser on the host computer.
2. Select **Enable Secure Shell Server (sshd)** in the **Startup Settings** section.
3. Click **Save** above the **System Settings** section.
4. Click **Restart** on the upper right of the page to restart the NI myRIO target.

Importing C Support for NI myRIO to Eclipse

Complete the following steps to import the C Support for NI myRIO files to Eclipse:

1. Launch Eclipse, and select your default workspace.
2. In Eclipse menu, click **File»Import**. The **Import** dialog box appears.
3. On the **Select** step, select **Existing Projects into Workspace**. Click **Next**.
4. On the **Import Projects** step, select **Select archive file**. Click **Browse** and select the `C_Support_for_NI_myRIO_v1.0.zip` file that you downloaded.
5. Now, a few items are added to the **Project** list. Keep all items checked and click **Finish**.

Note "C Support for NI myRIO" is the base library and other projects' dependency. Projects that begin with "myRIO Example" are examples. "myRIO Template" is for creating your custom project.

Note Projects that already exist in the workspace directory appear grayed. If you imported previously, you need to rename or delete those projects from the disk before you import again.

Installing FPGA Bitfiles

After importing the C Support for NI myRIO to Eclipse, create a connection to install the FPGA bitfiles on to the NI myRIO target.

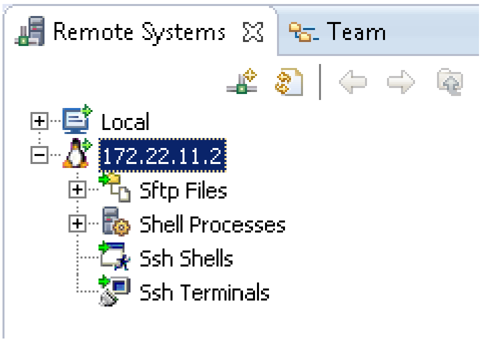
Complete the following steps to create a connection to the NI myRIO target:

1. In Eclipse menu, click **Window»Open Perspective»Other**. Select **Remote System Explorer** and click **OK**.
2. In the Remote Systems pane toolbar, click **Define a new connection to a remote system**. The **New Connection** dialog box appears.
3. On the **Select Remote System Type** step, select **Linux** and click **Next**.

4. On the **Remote Linux System Configure** step, enter the **Host name** as `<IP Address>` (get from the **NI myRIO USB Monitor**, usually `172.22.11.2`) and click **Next**.
5. On the **Files** step, select **ssh.files** and click **Next**.
6. On the **Processes** step, select **processes.shell.linux** and click **Next**.
7. On the **Shells** step, select **ssh.shells** and click **Next**.
8. On the **Ssh Terminals** step, click **Finish**. Your new remote system now shows in the **Remote Systems** pane.

Complete the following steps to connect to the NI myRIO target:

1. In the **Remote Systems** pane, right-click the target and select **Connect**. The **Enter Password** dialog box appears.
2. Enter the password for the NI myRIO and click **OK**. By default, the user ID is “admin” and there is no password.
3. Click **OK** on the **Info** dialog box.
4. The **Keyboard Interactive authentication for...** dialog box appears. Enter the same authentication as step 2 and click **OK**.
5. If the NI myRIO target is connected, a green arrow shows on the target icon.



Complete the following steps to install the FPGA bitfiles on the NI myRIO target:

1. In the **Remote Systems** pane toolbar, expand **Sftp Files** and select **New»Folder** on the **Root** shortcut menu. The **New Folder** dialog box appears.
2. Enter the folder name `/var/local/natinst/bitfiles` and click **Finish**. Ignore the error if the folder already exists.
3. In the **Remote Systems** pane, navigate to the newly created folder. Select **Export From Project...** in the **bitfiles** folder shortcut menu. The **Export** dialog box appears.
4. Expand the C Support for NI myRIO, and click **source**. If your hardware is NI myRIO-1900, check **NiFpga_MyRio1900Fpga10.lvbitx**. If your hardware is NI myRIO-1950, check **NiFpga_MyRio1950Fpga10.lvbitx**. Click **Finish**.
5. If succeeded, you see the **NiFpga_MyRio1900Fpga10.lvbitx** or **NiFpga_MyRio1950Fpga10.lvbitx** file in the **bitfiles** folder.

Using C Support for NI myRIO with Eclipse for NI Linux Real-Time

The C Support for NI myRIO contains the following components:

- Examples you can build and deploy to the NI myRIO.
- The template project you can use to build and deploy your own application.

Examples and the template project are imported to Eclipse when you import the C Support for NI myRIO to Eclipse.

Building and Deploying an Example Project

Complete the following steps to build and deploy an example project. See the Examples Overview section of this document for more information about examples.

1. Open Eclipse in the C/C++ perspective. If Eclipse is not in the C/C++ perspective, click **Window»Open Perspective»Other** in Eclipse menu, select **C/C++ (default)**, and click **OK**.
2. In the Project Explorer pane, right-click the example project and select **Build Project**. Wait until the build finishes.
3. To run the example, right-click the project and select **Run As»Run Configurations**.
4. In the **Run Configurations** dialog box, expand **C/C++ Remote Application** and select the project to run. On the right of the pane, click **Search Project** under the **C/C++ Application** and select the built binary. Click **OK**. Click **Apply** and click **Run**.
5. Now, the example runs on the NI myRIO target and the result displays on the **Console** window.
6. To debug the example, right-click the project and select **Debug As»Debug Configurations**.
7. In the **Debug Configurations** dialog box, expand **C/C++ Remote Application** and select the project to debug. On the right of the pane, click **Search Project** under the **C/C++ Application** and select the built binary. Click **OK**. Then, click **Apply** and click **Debug**.
8. Now, the example runs on the NI myRIO target with a debugger. The debug tools are on the toolbar.

Note The project's default configuration builds the **Debug** version for debugging. You can switch to release by right-clicking the project and selecting **Build Configurations»Set Active»Release**.

Building and Deploying Your First Project

Complete the following steps to build and deploy your first project using the template project:

1. Open Eclipse in the C/C++ perspective. If Eclipse is not in the C/C++ perspective, click **Window»Open Perspective»Other** in Eclipse menu, select **C/C++ (default)**, and click **OK**.
2. In the **Project Explorer** pane, right-click the **myRIO Template** project and select **Rename**. Enter a meaningful name and click **OK**.

Note You must always rename the project. Otherwise you cannot import the "myRIO Template" again.

3. Add your source files to the project. Code in myRIO example projects is designed for reuse. You can copy source files from the example project to use in your application.
4. In the **Project Explorer** pane, right-click the project and select **Build Project**. Wait until the build finishes.
5. To run the project, right-click the project and select **Run As»Run Configurations**.
6. In the **Run Configurations** dialog box, expand **C/C++ Remote Application** and select the project to run. On the right of the pane, rename the configuration to match the project name. Click **Search Project** under the **C/C++ Application** and select the built binary. Click **OK**. Click **Apply** and click **Run**.
7. Now, the project runs on the NI myRIO target and the result displays on the **Console** window.
8. To debug the project, right-click the project and select **Debug As»Debug Configurations**.
9. In the **Debug Configurations** dialog box, expand **C/C++ Remote Application** and select the project to debug. On the right of the pane, rename the configuration to match the project name (if not renamed yet). Click **Search Project** under **the C/C++ Application** and select the built binary. Click **OK**. Then, click **Apply** and click **Debug**.
10. Now, the project runs on the NI myRIO target with a debugger. The debug tools are on the toolbar.

Note The project's default configuration builds the **Debug** version for debugging. You can switch to release by right-clicking the project and selecting **Build Configurations»Set Active»Release**.

Creating a New Project

The “C Support for NI myRIO” project also includes a template project archive that you can use to start your additional projects for the NI myRIO. The template project archive is included when you import the “C Support for NI myRIO” project. Complete the following steps to create a new project:

1. Open Eclipse in the C/C++ perspective. If Eclipse is not in the C/C++ perspective, click **Window»Open Perspective»Other** in Eclipse menu, select **C/C++ (default)**, and click OK.
2. In Eclipse menu, click **File»Import**. The **Import** dialog box appears.
3. On the **Select** step, select **Existing Projects into Workspace**. Click **Next**.
4. On the **Import Projects** step, select **Select archive file**. Click **Browse** and browse to `<WORKSPACE>\C Support for NI myRIO\template project` and select the **myRIO Template v1.0.zip** file.

Note `<WORKSPACE>` varies by machine. The path is set when you launch Eclipse.

5. Now, the myRIO Template item is added to the **Project** list. Click **Finish**.

Note Projects that already exist in the workspace directory appear grayed. If you imported previously, you need to rename or delete those projects from the disk before you import again.

Complete the Building and Deploying Your First Project section of this document to use this template project with the NI myRIO.

Examples Overview

National Instruments provides the following eight examples for using the NI myRIO. Refer to the `main.c` file in each example for more information.

Accelerometer

Demonstrates using the onboard accelerometer. This example reads the acceleration in the three directions and prints the values to the console.

AIO

Demonstrates using the analog input and output (AIO). This example reads initial values of two analog input channels from connector A and writes the sum of the read values on connector B. This example also prints the values to the console.

DIO

Demonstrates using the digital input and output (DIO). This example reads initial values of two digital input channels from connector A and writes the Boolean AND of the read values on connector B. This example also prints the values to the console.

Encoder

Demonstrates using the encoder. This example reads a step and direction signal from the encoder on connector B and prints the values to the console.

I2C

Demonstrates using the I²C. This example reads the temperature from a connected TMP102 digital temperature sensor and writes the response to the console.

PWM

Demonstrates using pulse-width modulation (PWM). This example generates a PWM signal from PWM 0 on connector A.

SPI

Demonstrates using the sperial peripheral interface bus (SPI). This example writes a message to the SPI bus and then prints any returned bytes to the console.

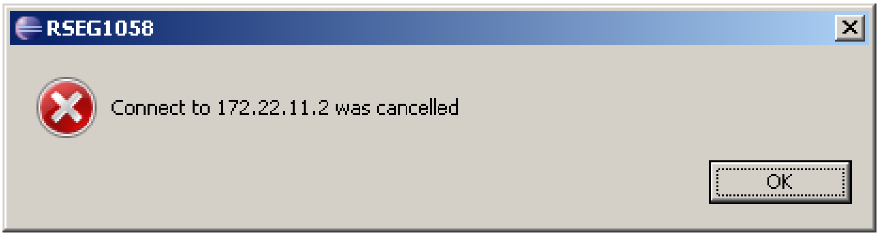
UART

Demonstrates using the universal asynchronous receiver/transmitter (UART). This example writes a character to the UART bus and then prints any returned character to the console.

Known Issues

Connection Cancelled

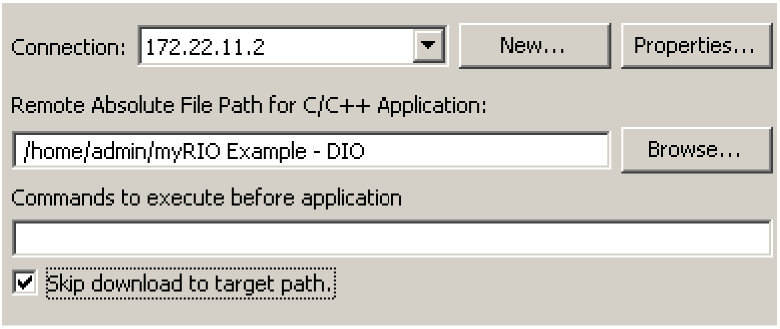
You might find the connection to the NI myRIO target is sometimes cancelled when you start to run or debug your project in Eclipse.



Connect to 172.22.11.2 was cancelled

The connection in Eclipse is not stable when downloading your application to the target. Dismiss the message and the connection will be reestablished when downloading the next time. Or you can manually reestablish the connection in the **Remote System Explorer** perspective.

A way to minimize the problem is to select **Skip download to target path** in the **Run Configurations** or **Debug Configurations** dialog box. However, you must then manually transfer the built executable to the NI myRIO using SFTP.


Connection: 172.22.11.2

Remote Absolute File Path for C/C++ Application: /home/admin/myRIO Example - DIO

Commands to execute before application:

[✓] Skip download to target path.

How to Use the NI myRIO Target with LabVIEW

To use the myRIO target with LabVIEW, you need to install LabVIEW and the LabVIEW Real-Time Module to ensure required components are installed. You then reinstall the LabVIEW for myRIO Module.

Related Information

NI myRIO Shipping Personality Reference—Contains information about the FPGA bitfiles included with the C Support for NI myRIO.