Resource R7 Setting up the C Development Tool for myRIO

Box 00.2 setting up a lab PC or one's own laptop?

For configuring your own laptop, complete *all* the steps, below.
For configuring a lab PC, complete steps 4 and 5 of Part A, then complete all remaining parts (Part B – Part F).

Box 00.3 myRIO connected?

Parts A, B, and C can be performed *without connecting* your laptop to one of the lab myRIOs. For Parts D, E, and F, a myRIO connection is required.

Do Parts A, B, and C just once, in the order shown.

Resource R7.12 Part A: Setting up the software environment

Follow these instructions to set up the C Development Tool for myRIO. Clicking on hyperlinks opens the appropriate websites in your browser.

1. Download and install LabVIEW 2015 myRIO Toolkit. (2700 Mb)

   For Native Windows 8 or 10: Download myRIOToolkit2015. Mount disk image. Then run `setup.exe`.

   For Native Windows 7: Download from myRIOToolkit2015. You may need a means of mounting the `.iso` disk image. For example, use Virtual CloneDrive to mount the `.iso` disk image files as a virtual CD-ROM drive. Then run `setup.exe`.

   For Virtual Windows 7, 8, or 10 under Parallels: Download myRIOToolkit2015 *under OS X*. From Parallels, Devices › CD/DVD › Connect Image... and mount the disk image.
   Then run `setup.exe`.

2. Install Java. Visit the Java website GetJava to download Java. (17 Mb)
   Use Internet Explorer, not Microsoft Edge.
3. Install the C/C++ Development Tools for NI Linux Real-Time 2014, Eclipse Edition.
   Visit this link Eclipse2014 to download and install Eclipse. (260 Mb)

4. Project templates have been prepared for each of the ME 477 laboratory exercises. Visit the ME 477 website Resources Page to download the `ME477 myRIO support 2018` archive. Remember where you put this archive, but **do not unzip**. (2 Mb)
5. Eclipse uses Launch Configurations to specify how the project will be deployed and run on the myRIO.
   Visit the ME 477 website Resources Page to download the `ME 477 Launch Configurations` archive.
   Unzip into folder `LaunchConfig477` (40 kb). Remember where you put the folder.
6. Add the compiler path to the system environment variables.
   a. Visit the ME 477 website Resources Page.
      `64-bit compiler path` file, select and copy with ctrl + C the contents.
   b. In the Windows Control Panel, select System and Security › System › Advanced system settings to display the `System Properties` dialog box.
   c. Click Environment Variables to display the `Environment Variables` dialog box.
   d. Select `PATH` in the `User variables` group box and click Edit.
      If `PATH` does not exist, click New to create it.
   e. Click New and paste with ctrl + V the compiler path to the end of `Variable value` (separated by the `;` character). Be certain that there are *no extra spaces* in the path.
7. Click OK to close the dialog box and save changes.

Resource R7.13 Part B: Define a connection to the myRIO

Complete the following steps to define a connection in Eclipse from your laptop to the myRIO target.

1. Launch Eclipse, specify a workspace, and click OK to display the *C/C++* perspective (default).
   Two other perspective views, *Remote Systems Explorer* and *Debug*, will also be useful. To make these available, select Window › Open Perspective › Other to display the `Open Perspective` dialog box.
   Then select `Remote Systems Explorer` and click OK to display the *Remote Systems Explorer* perspective. Repeat this process to display

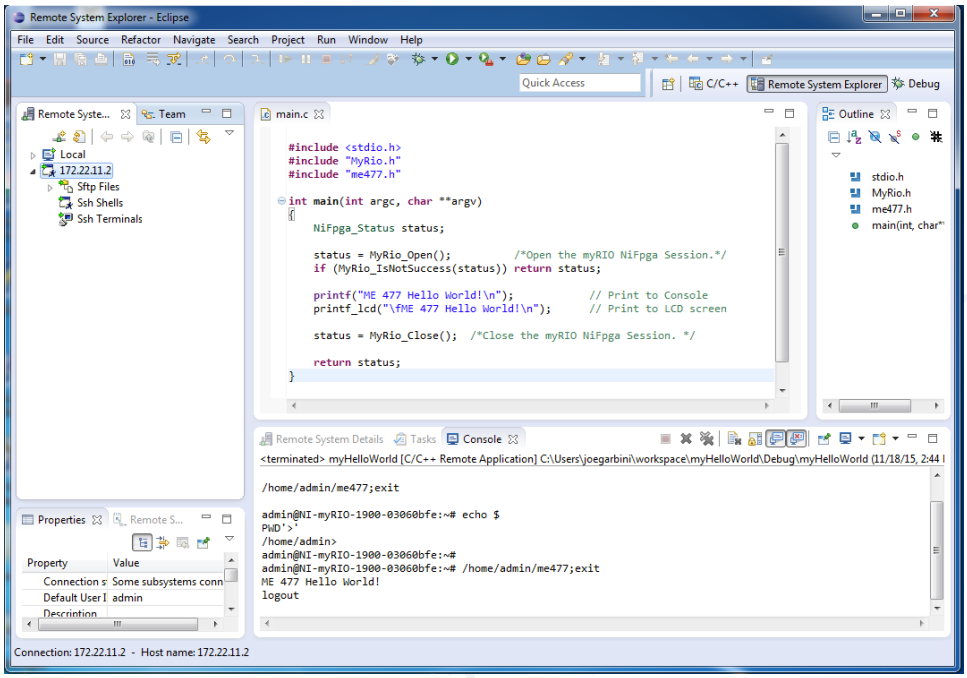

Figure 00.12: `Remote Systems Explorer` with myRIO connection successfully defined.

the *Debug* perspective. Buttons for all three perspectives should appear and can be used at any time to switch perspectives.

2. Open the *Remote Systems Explorer* perspective to display the `Remote Systems` pane at left.
3. Click the `Define a connection to remote system` icon to display the `New Connection` dialog box.
4. Select General > SSH Only.
5. Enter the IP address `172.22.11.2` in the `Host name` textbox and click Finish. Your target displays in the `Remote Systems` tab in the `Remote System Explorer` pane, as shown in Figure 00.12.

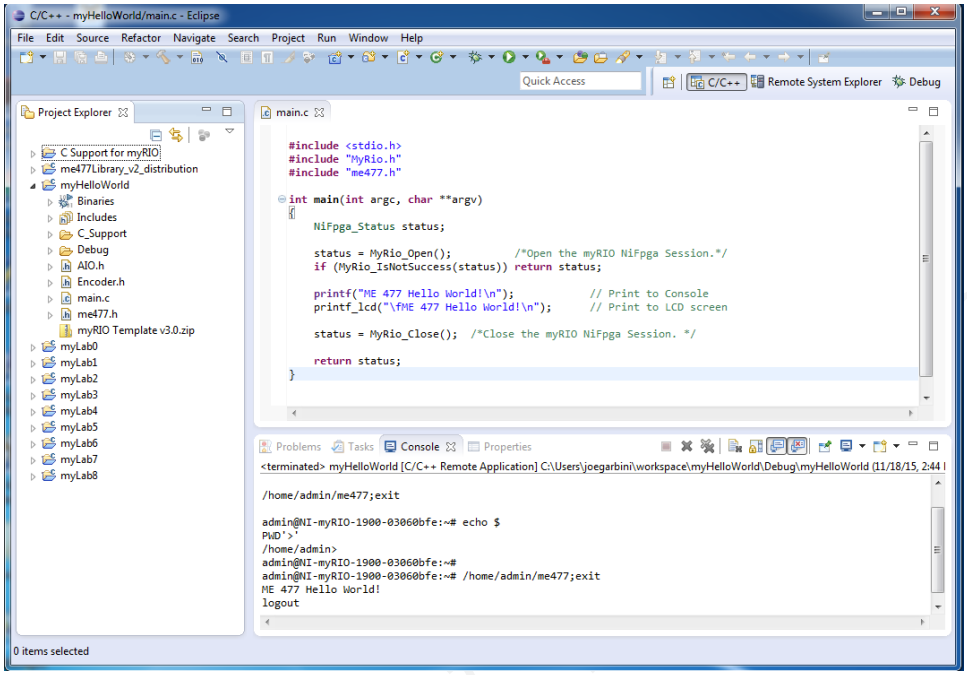

Figure 00.13: ME 477 Project Templates.

Resource R7.14 Part C: Importing C Support and Launch Configurations

Complete the following steps to import C Support and Launch Configurations to Eclipse.

1. From the *C/C++* perspective, select File > Import to display the `Import` dialog box.
2. Select General > Existing Projects into Workspace and click Next to display the `Import Projects` page.
3. Select Select archive file, click Browse and select the `ME 477 C Support for myRIO` zip file downloaded in step 4 of Part A.
4. Ensure that all items are checked and click Finish to import `ME 477 C Support for myRIO`. See Figure 00.13.
5. Build (compile) all projects with menu selection Project > Build All.

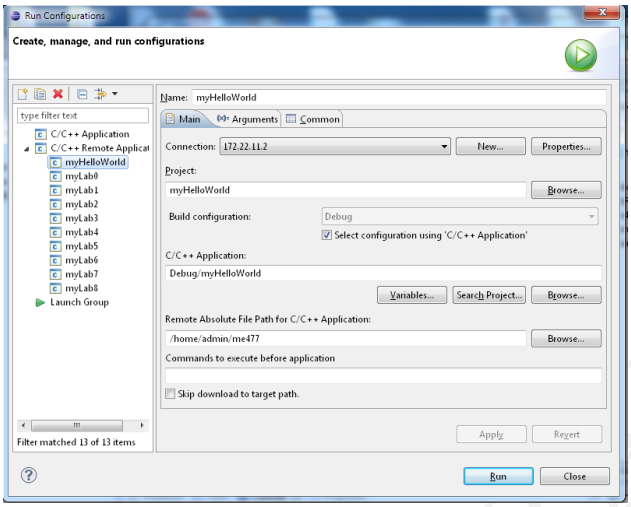

Figure 00.14: Launch Configurations.

6. Again, from the *C/C++* perspective, select File > Import to display the `Import` dialog box.
7. Select menu item Run/Debug > Launch Configurations and click Next to display the `Import Launch Configurations` page.
8. Click Browse and select the `LaunchConfig477` folder that you downloaded in step 5 of Part A.
9. Ensure that all items are checked and click Finish. To check that the import of the Launch Configurations was successful, select menu item Run > Run Configurations... and compare the dialog with Figure 00.14.

Box 00.4 myRIO connected?

Your laptop *must be connected through a USB cable* to one of the myRIOs to perform Parts 4, 5, and 6. Each time you connect, a `myRIO USB Monitor` dialog box will appear indicating myRIO IP Address `172.22.11.2`. *Always* select Do Nothing.

Resource R7.15 Part D: Connect to the myRIO target

Complete the following steps to establish a connection between Eclipse and the myRIO target.

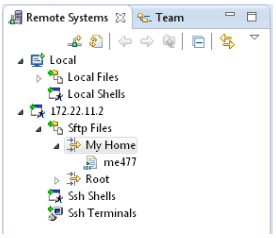

Figure 00.15: the `Remote Systems` tab should appear like this once a connection is established successfully.

1. In the `Remote Systems` pane, right-click the target and select Connect from the shortcut menu to display the `Enter Password` dialog box.
2. Enter the user ID: `admin` and password: <UW: `me477` | SMU: *leave blank*> and click OK.
3. Click OK in the `Info` dialog box.
4. If the `Keyboard Interactive authentication` dialog box appears, leave the password blank, and click OK. As shown in Figure 00.15, green arrow appears on the target icon when the myRIO is connected.

Resource R7.16 Part E: Running the myHelloWorld project

In Parts 5 and 6 you will run and debug a project. Here, the `myHelloWorld` project is used as example.

Eclipse uses a "Run Configuration" to specify how the project will be deployed and run on the myRIO. Run Configurations for ME 477 projects were downloaded in step 5 of Part A.

Complete the following steps to run the `myHelloWorld` example project.

1. In Eclipse, switch to the *C/C++* perspective.
2. You can view and edit the C source code by double clicking on the `myHelloWorld` project in the left pane, and then double clicking on `main.c`.
3. In the `Project Explorer` pane, right-click the `myHelloWorld` project, and select Build Project from the shortcut menu to build the project. Any build errors will be noted in the `Problems` pane.

4. Right-click the `myHelloWorld` project and select Run As › Run Configurations to display the `Run Configurations` dialog box.
5. Select the `myHelloWorld` project in the left pane.
6. Click Run. The project runs on the myRIO target. You can find the result in the `Console` pane, and on the LCD screen.

Resource R7.17 Part F: Debugging the myHelloWorld project

Similarly, Eclipse uses a "Debug Configuration" to specify how the program will be debugged on the myRIO. Once the Debug Configuration for a project is set up, debugging the program requires just a single click.

Complete the following steps to set up the Debug Configuration for the `myHelloWorld` project. These include building, deploying, and debugging the project.

1. In Eclipse, switch to the *C/C++* perspective.
2. In the `Project Explorer` pane, right-click the `myHelloWorld` project and select Debug As › Debug Configurations to display the `Debug Configurations` dialog box.
3. Select the `myHelloWorld` project in the left pane.
4. Click Debug. The project runs on the myRIO target within the debugger. Some warnings may appear in the Console pane. Under normal circumstances, these warnings are not a problem. You can find the debug tools on the toolbar of Eclipse. There will be more about this in the first laboratory exercise.
5. For now, try setting a breakpoint at the `printf()` statement by double-clicking in the margin at left of that statement. A blue dot *with a small checkmark* should appear in the margin. The blue dot indicates that the breakpoint is enabled, and the checkmark indicates that the breakpoint is installed.
   If you `resume` (green arrow) from the beginning of the program, execution should pause at the breakpoint, as shown in Figure 00.16.

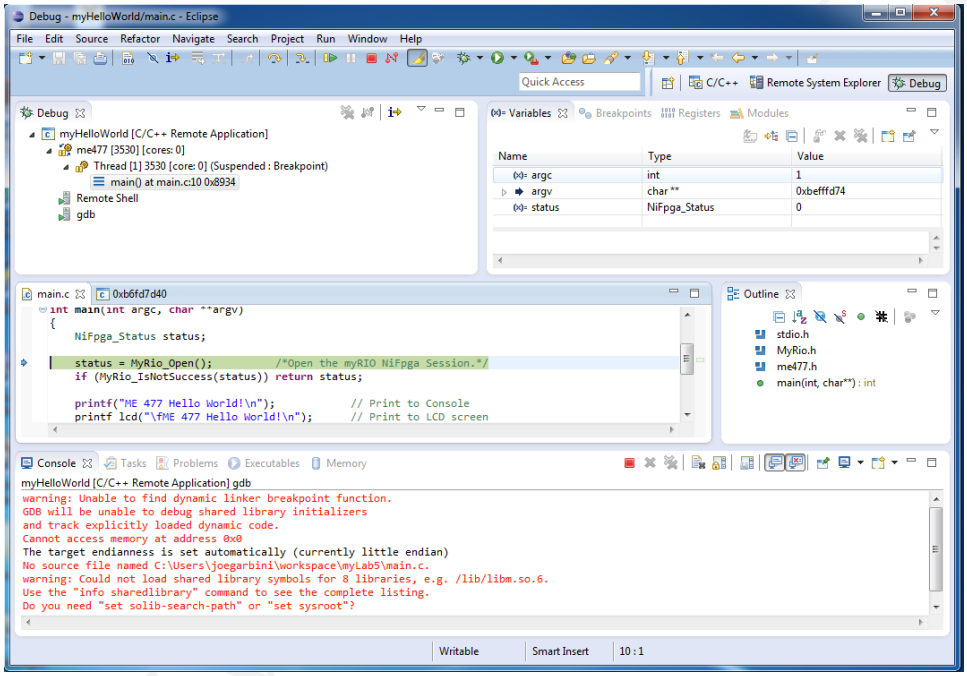

Figure 00.16: debugging and stopped at a breakpoint.