

NI myRIO 入门指南



目录

概述.....	1
0. 开发 NI myRIO 前的准备工作及软件配置.....	2
1. 开发第一个 NI myRIO 项目	10
2. 连接外围 I/O	17
3. NI myRIO 的 WiFi 连接功能	22
4. 图像采集以及视觉算法应用	27
5. 基于 NI myRIO 的控制应用	40
6. 通过智能终端进行远程监控	46
7. 生成上电自启动程序	55
8. NI myRIO 板载 FPGA 资源开发介绍.....	59
附录.....	64

概述

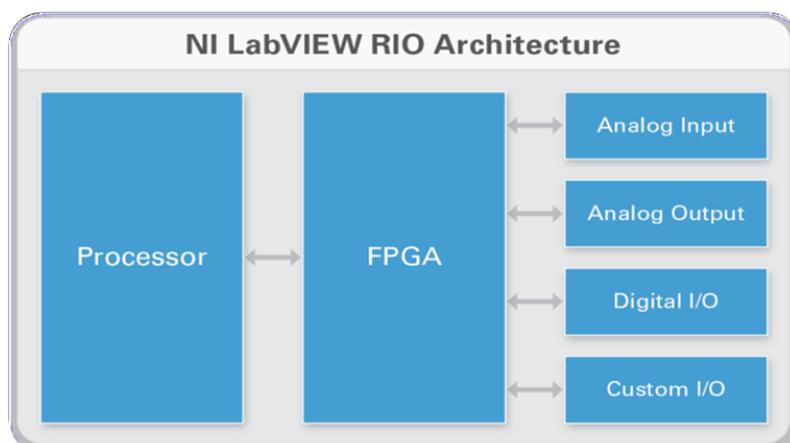
NI myRIO 是 NI 针对教学和学生创新应用而最新推出的嵌入式系统开发平台。NI myRIO 内嵌 Xilinx Zynq 芯片，使学生可以利用双核 ARM Cortex-A9 的实时性能以及 Xilinx FPGA 可定制化 I/O，学习从简单嵌入式系统开发到具有一定复杂度的系统设计。

NI myRIO 作为可重配置、可重使用的教学工具，在产品开发之初即确定了以下重要特点：

- ✓ 易于上手使用：引导性的安装和启动界面可使学生更快地熟悉操作，帮助学生了解众多工程概念，完成设计项目。
- ✓ 编程开发简单：通过实时应用、FPGA、内置 WiFi 功能，学生可以远程部署应用，“无头”（无需远程电脑连接）操作。三个连接端口（两个 MXP 和一个与 NI myDAQ 接口相同的 MSP 端口）负责发送接收来自传感器和电路的信号，以支持学生搭建的系统。
- ✓ 板载资源丰富：共有 40 条数字 I/O 线，支持 SPI、PWM 输出、正交编码器输入、UART 和 I2C，以及 8 个单端模拟输入，2 个差分模拟输入，4 个单端模拟输出和 2 个对地参考模拟输出，方便通过编程控制连接各种传感器及外围设备。
- ✓ 安全性：直流供电，供电范围为 6V~16V，根据学生用户特点增设特别保护电路。
- ✓ 便携性

NI myRIO 上所有这些功能都已经在默认的 FPGA 配置中预设好，能使学生在较短时间内就可以独立开发完成一个完整的嵌入式工程项目应用，特别适合用于控制、机器人、机电一体化、测控等领域的课程设计或学生创新项目。当然，如果有其他方面的嵌入式系统开发应用或者是一些系统级的设计应用，也可以用 NI myRIO（以下简称 myRIO）来实现。

图 1. NI myRIO 基于 NI RIO 架构



本教程将具体地向用户介绍如何从零开始进行基于 NI myRIO-1900 和 LabVIEW 的嵌入式系统开发。

0 开发 NI myRIO 前的准备工作及软件配置

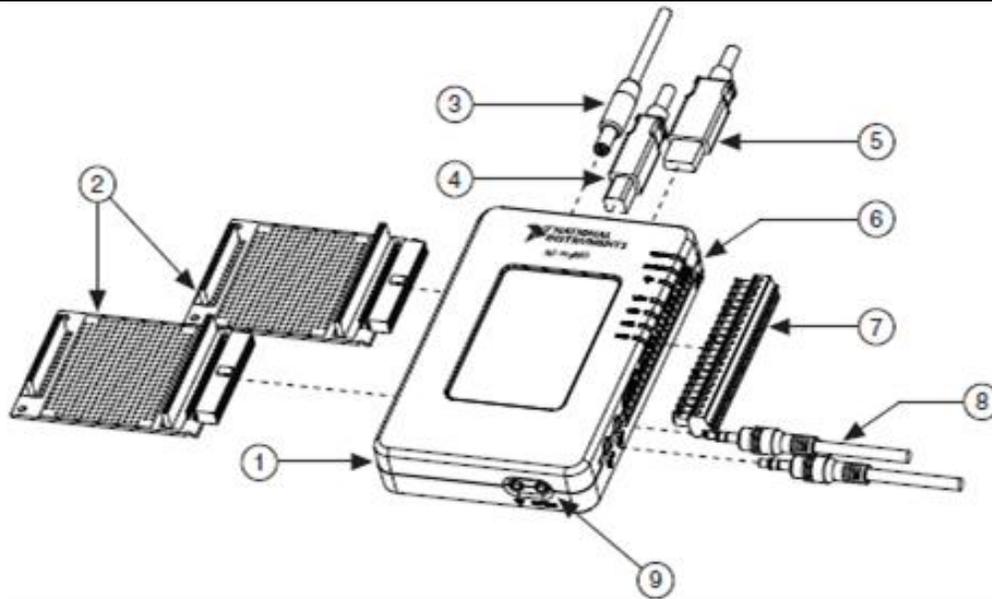
0.1 开发前的准备工作

在使用一个新的 myRIO 之前需要在计算机上安装软件并对其进行配置以做好系统开发的准备。必须安装的软件有：

- LabVIEW
- LabVIEW Real-Time (LabVIEW 实时模块)
- LabVIEW myRIO Module (LabVIEW myRIO 模块)

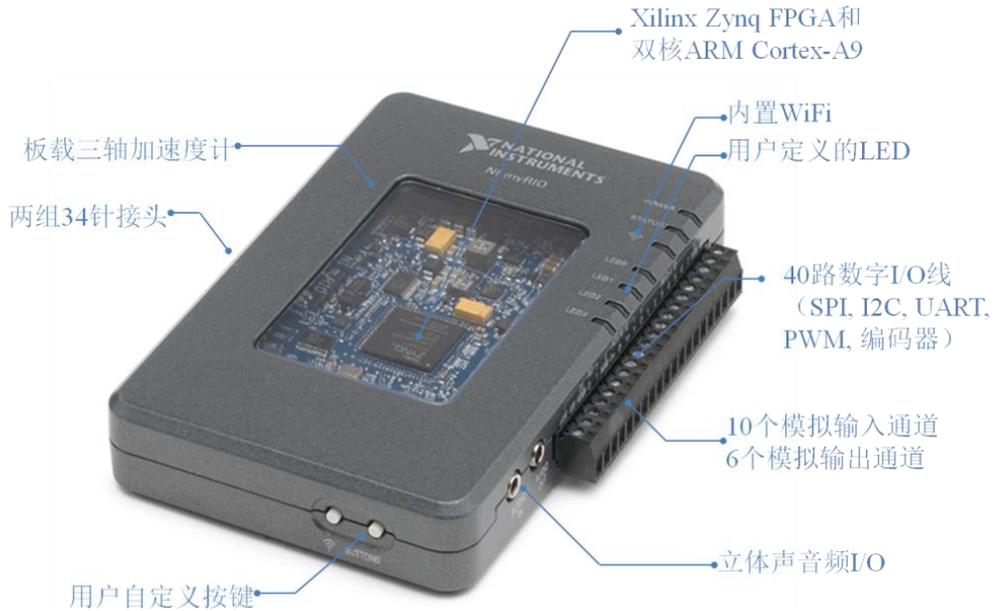
这些软件的安装程序可以在 myRIO 的随盒光盘 NI LabVIEW 2014 myRIO SOFTWARE BUNDLE, DVD1 中找到。

图 0-1. NI myRIO-1900 及其配件



- | | |
|--------------------------------|---------------------------|
| 1 NI myRIO-1900 | 5 USB Host 连接线 (未包含在开发盒中) |
| 2 myRIO 的扩展端口 (MXP) (开发盒中包含一个) | 6 LED 灯 |
| 3 输入电源线 | 7 迷你系统端口 (MSP) 螺旋式接线柱 |
| 4 USB 设备连接线 | 8 音频输入/输出线 (开发盒中包含一条) |
| | 9 按钮 0 |

图 0-2. NI myRIO-1900 实物



安装软件

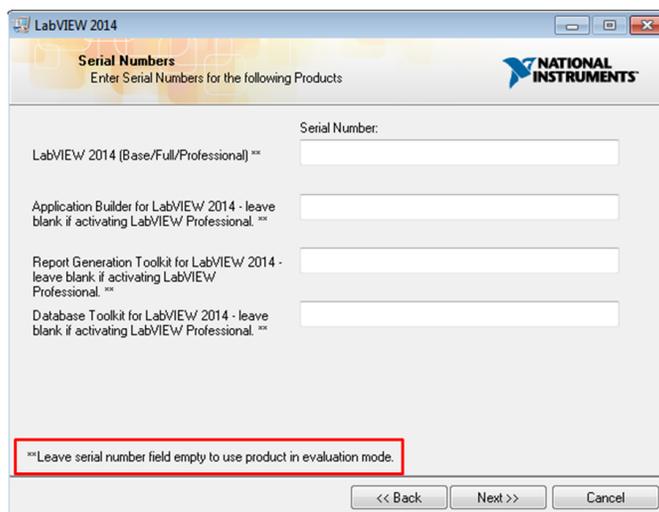
请按以下步骤安装软件：

- 1、将 NI LabVIEW 2014 myRIO SOFTWARE BUNDLE, DVD1 光盘插入电脑光驱。屏幕上会自动弹出 AutoPlay 的对话框，单击 **Open folder to view files** 以查看安装文件。
- 2、双击 **Distribution**，可看到除必须安装的三个软件的文件夹之外，还包括下列文件夹：

目录	说明
Control Design and Simulation	控制设计与仿真模块，用以帮助用户设计控制算法。
FPGA	如果用户需要用到 myRIO 上的 FPGA 资源，并且需要对这部分进行自定义编程，可选用安装。
MathScript RT	如果用户在 LabVIEW 中需要调用 Matlab 编写的 m 文件的脚本，可选用安装。
Vision	视觉开发模块，包含了很多现成的机器识别算法，例如颗粒分析，边缘提取等，以帮助用户在视觉操作时快速实现功能。
VisionAcq	视觉采集模块，当用户需要使用 USB 摄像头与 myRIO 连接以采集视频图像信息时，可选用安装。

- 3、双击 **LabVIEW** 文件夹，在打开的目录中双击 **setup**，按照屏幕提示完成软件安装，此处注意如用户未购买相应软件，仅能安装评估版 LabVIEW，如图 0-3 所示。

图 0-3. 序列号输入窗口



4、以同样的方法再分别安装 **LabVIEW Real-Time** 和 **LabVIEW myRIO Module**，此处注意如用户未购买相应软件，同样仅能安装评估版。

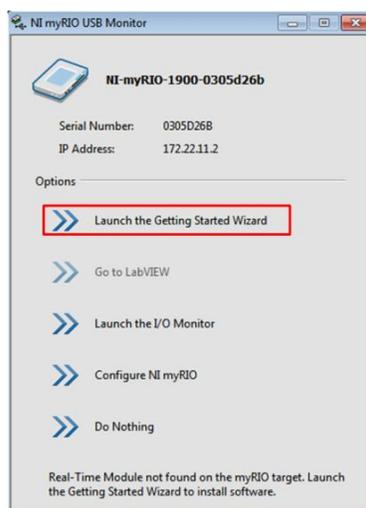
安装好软件之后便可以给 myRIO 插上电源线，并用 USB 线将设备与计算机连接起来。



注意 由于此时 myRIO 的实时处理器上并没有实际安装任何软件，所以右侧 **STATUS** 的 LED 指示灯一直处于红色闪烁状态。

当 myRIO 与计算机连接好后，会自动弹出如图 0-4 所示的启动界面，单击 **Launch the Getting Started Wizard** 对 myRIO 进行相关设置。

图 0-4. NI myRIO USB 启动窗口



选项	说明
Launch the Getting Started Wizard	通过 Getting Started Wizard, 用户可以迅速查看 NI myRIO 的功能状态。向导的功能有: 检查已连接的 NI myRIO, 连接到选中设备, 给 NI myRIO 安装软件或进行软件更新, 为设备重命名, 以及通过一个自检程序测试加速度传感器、板载 LED 以及板载自定义按钮。
Go to LabVIEW	选择此项后直接弹出 LabVIEW Getting Started 窗口。
Configure NI myRIO	选择后打开一个基于网页的 NI myRIO 配置工具。
Do Nothing	你可通过此选项关闭 NI myRIO USB 启动窗口。



注意 如果没有自动弹出 NI myRIO USB Monitor 的启动界面, 可以双击 \Program Files\National Instruments\LabVIEW 2014\resource\myRIO 目录下的 myrioautoplay.exe 打开

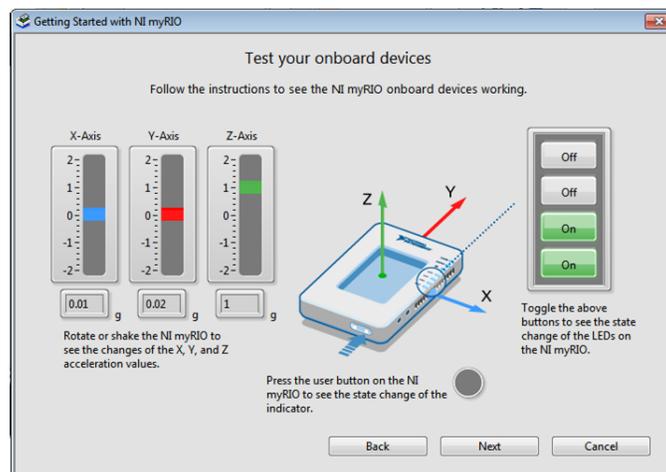
找到已安装的设备之后, 单击 Next, 在下一个界面中可以看到其序列号, 用户也可以修改设备名字, 但之后需要重启 myRIO。再次单击 Next 之后, 会自动将上位机已经安装的相关软件在 myRIO 上创建一套实时操作的副本, 这一过程可能会花费几分钟的时间。由于 myRIO 在安装完软件之后需要重启, 所以启动界面会再次出现, 点击 **Do Nothing** 即可。



注解 myRIO 的 ARM 处理器上运行的是 Linux RT 实时操作系统, 不过一般情况下用户不需要关心底层的操作系统细节, 因为 LabVIEW 实时模块会帮助用户和操作系统打交道, 开发者只需要集中精力实现功能即可。

随后安装向导会提供一个如图 0-5 所示的测试面板使用户可自由测试 myRIO 上的三轴加速度计和 LED 灯的硬件性能。单击 Next 完成安装, 下面就可以在 LabVIEW 中对 myRIO 进行进一步的自定义开发。

图 0-5. NI myRIO USB 启动窗口



0.2 软件配置

本节将介绍 myRIO 的一些深入配置方法，为选择性学习内容，若直接学习下一课内容将不受影响。在上一节中我们使用了 myRIO 现有的配置工具完成了初步的配置，如果需要进一步了解配制方法，可双击打开配置管理软件 NI MAX，在左侧一栏的远程系统中可查看到当前连接的 myRIO 设备。单击打开之后，可在页面右方看到设备的相关信息。在 IP 地址一栏中，以太网地址是指通过 USB 线连接到的网址，无线地址则尚未配置，通过无线方式将 myRIO 与计算机相连接的方法会在后面的课程中继续学习。在此页面中用户可继续自行查阅序列号、操作系统版本号、物理内存等基本信息。



注解 虽然 myRIO 实质上是通过 USB 线实现与计算机相连的，但由于计算机的驱动会将 USB 端口虚拟成网口，所以计算机会将 myRIO 识别成通过网络与其相连的设备。

图 0-6. NI MAX 设备配置管理界面



在左侧设备管理栏中继续展开 myRIO，可看到其设备与接口，如图 0-7 所示。如果在 myRIO 上连接了 USB 摄像头来采集图像时，同样也能在此处查看到 USB 摄像头资源。

图 0-7. NI MAX 设备配置管理界面



1 myRIO 上的 FPGA 资源

2 myRIO 上的 UR 串口

继续展开“软件”，可看到 myRIO 上所安装的软件的信息，此处的软件在上一节中已提到是计算机上所安装软件在实时操作系统下的副本，这些软件副本在主机上分别对应的安装软件可通过“我的系统”—“软件”下拉菜单查看，必须保持实时操作系统下的软件版本与主机的相一致，程序才能正确无误地编译下载至实时操作系统中在 myRIO 上运行。

因此当主机有软件或驱动软件的版本升级时，实时操作系统下的软件副本也需要一起升级。可通过右键单击 myRIO 下的“软件”按钮，**添加/删除软件**，或者直接点击右侧页面顶端的**添加/删除软件**按钮。

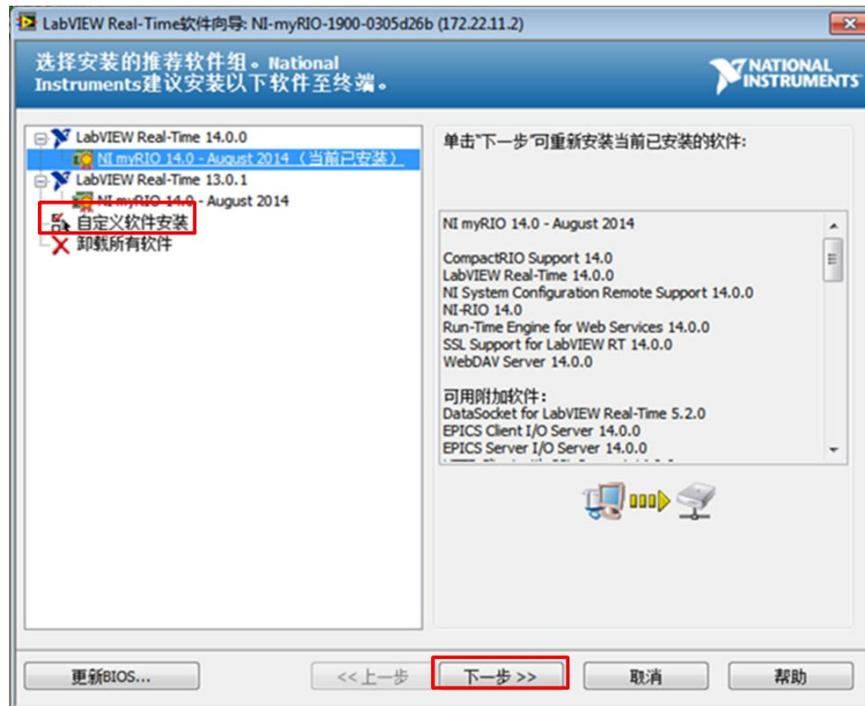
图 0-8. 添加/删除软件方法



注意 如果此处有要求管理员权限，密码默认为空。

在打开的对话框中可以看到当前在 myRIO 上安装的软件版本，单击**自定义软件安装**下一步，在弹出的对话框中选择确定要手动选择安装组件。

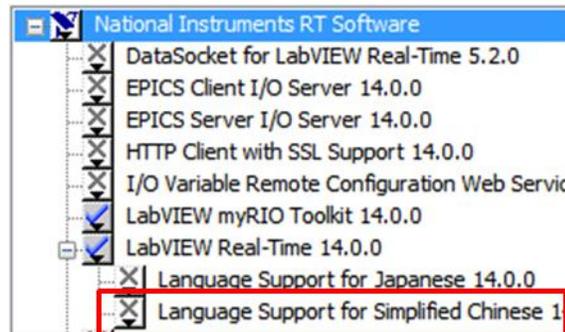
图 0-9. LabVIEW Real-Time 软件向导



在左侧滑动栏中便能看到需安装或卸载的组件，选择需要更新的软件，在右侧**主机中可用版本**中选择更新后的版本单击下一步便能将软件同步更新到 myRIO 上。



注意 如果用户安装的是中文版 LabVIEW 软件，在使用上一节中介绍的安装向导自动在 myRIO 上安装软件后，下载 LabVIEW 程序时系统会提示语言版本不匹配的错误，可以通过在上述自定义软件安装的可选项中选择安装 **Language Support for Simplified Chinese** 来解决此问题，安装完之后还需要回到 NI MAX 设备配置管理界面中的系统设置选项卡里，在**语言环境**的下拉菜单中选择简体中文并单击保存。



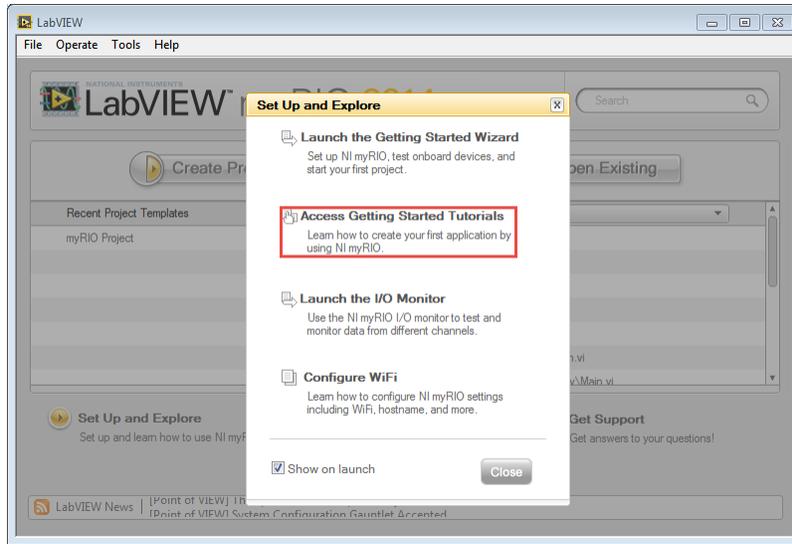


用户除了可以在 NI MAX 中看到 myRIO 的系统信息并对其进行多方面的配置修改之外，还可以通过 NI Network Browser（网络浏览器）来查看所安装的软件。如果没有桌面快捷方式的话可以在**开始 » 所有程序 » National Instruments** 中找到该应用，双击后会打开用户的浏览器，即时搜索网络上连接的所有设备，单击 myRIO 的虚拟 IP 地址打开系统配置窗口，类似于在 NI MAX 中所看到的界面，但目前只能在该浏览器窗口查看所安装的软件信息，配置修改等操作还是需要 NI MAX 中完成。

1 开发第一个 NI myRIO 项目

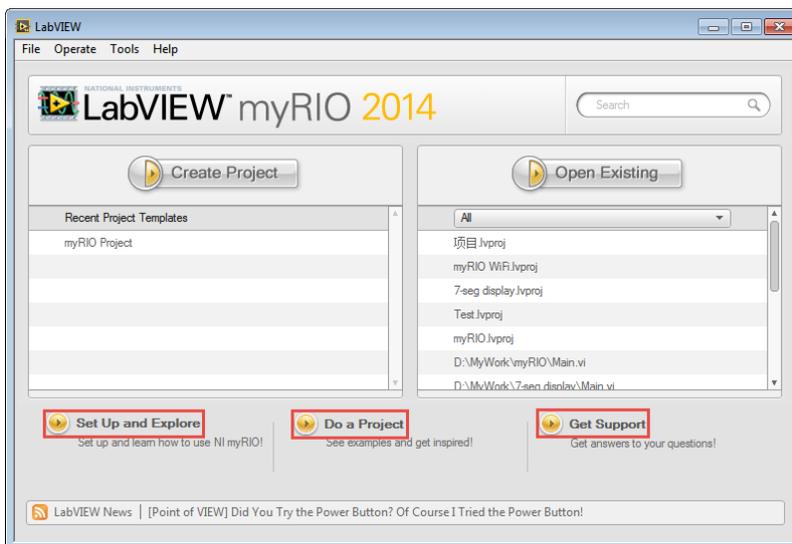
做完前面的准备工作之后便可以打开 LabVIEW 开发第一个 myRIO 项目，在启动时弹出的 Set Up and Explore 对话框中，用户可尝试点击 **Access Getting Started Tutorials**，它会链接到一个 myRIO 项目开发的在线指导，对新手很有帮助。

图 1-1. Set Up and Explore 对话框



此外，可以发现安装了 myRIO 模块之后的 LabVIEW 启动界面上会有更多的帮助链接，除了上述的 Set Up and Explore 还有 **Do a Project** 和 **Get Support**，可以将用户链接至一些指导网页或是论坛，提供更多现成的范例等学习资源。我们首先学习如何针对 ARM 处理器进行开发。

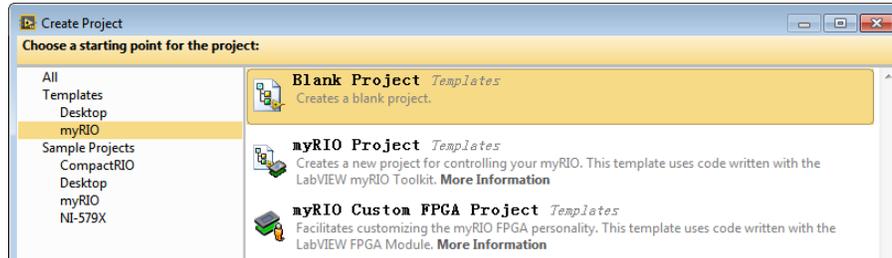
图 1-2. LabVIEW 启动界面



创建工程

在 LabVIEW 启动界面上单击 **Create Project**，会弹出一个对话框，用户可以在左侧看到不同的模板，选择 **Templates » myRIO** 之后会出现相应的一些模板：

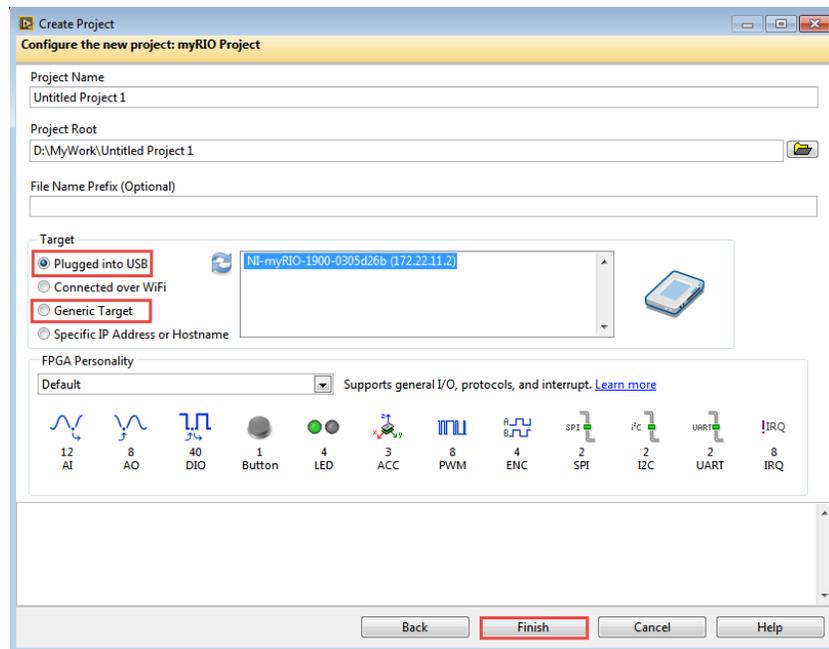
图 1-3. 创建 myRIO 工程模板



目录	说明
	空工程模板。创建一般的工程模板
	myRIO 工程模板。创建针对 myRIO 上 ARM 处理器开发的模板
	myRIO 自定义 FPGA 工程模板。创建同时对 ARM 处理器和 FPGA 编程的模板

选择创建 **myRIO Project**，用户可以自行修改 **Project Name** 和 **Project Root**。在 USB 线连接着 myRIO 和计算机的情况下，在 Target 一栏中会自动搜索到已连接的硬件设备。如果用户当前没有 myRIO，可以在 Target 一栏选择 **Generic Target** 先进行程序的开发，在后面再连接上硬件之后便可以直接运行。单击 **Finish** 完成工程的创建。

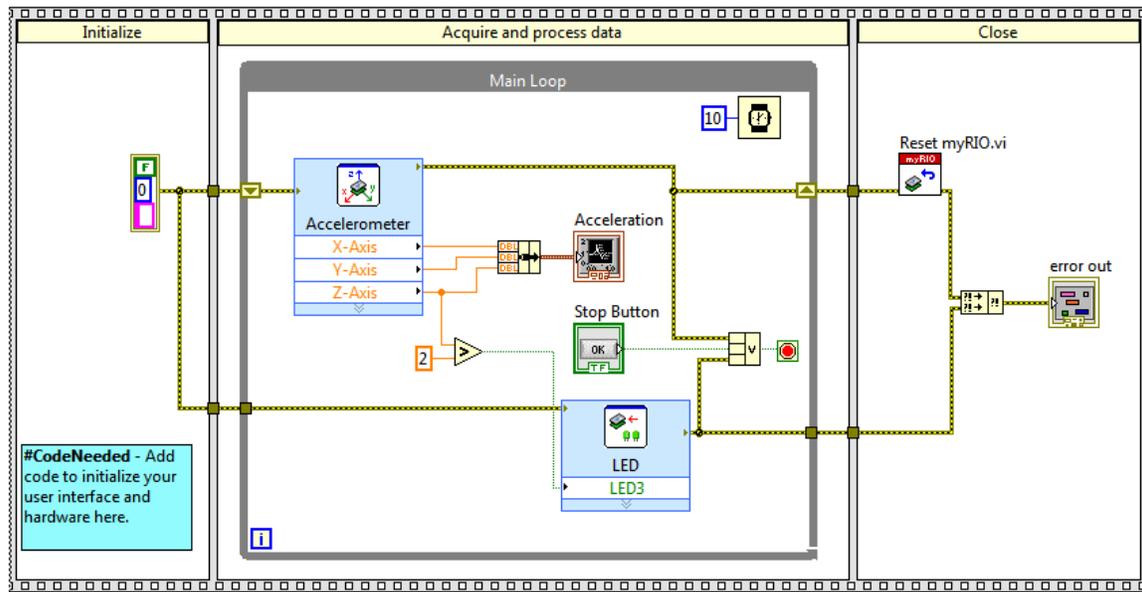
图 1-4. 创建 myRIO 工程模板



在程序自动创建的项目管理器中，用户可以观察到主程序，例如 Main.vi，如果是在 myRIO 这个 Target 下面，那么将来它就会运行在 ARM 处理器上。本工程中的 Main.vi 是所选用模板为用户提供的 一个实例，可直接运行。

双击打开 Main.vi，可以看到其前面板和程序框图。仔细观察可以发现，程序框图中的顺序结构是为了使用户能更清晰地了解其数据流向。整个模板是一个每 10 毫秒执行一次的 while 循环，它从板载加速度传感器上读取 X、Y、Z 轴的加速度数据。

图 1-5. “示例程序”程序框图



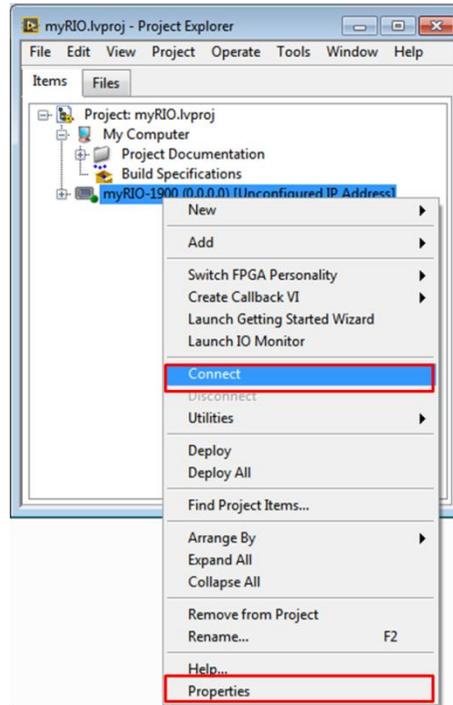
双击打开快速 VI Accelerometer，当三个轴都勾选上时，每次运行循环，会将三个轴的数据都读取，单击左下角的 **View Code** 可以查看底层 VI。在程序框图中右击打开函数选板中的 myRIO 函数选板，则可以看到板载 I/O 资源多个现成的快速 VI 函数。

运行实例

请按以下步骤运行实例程序：

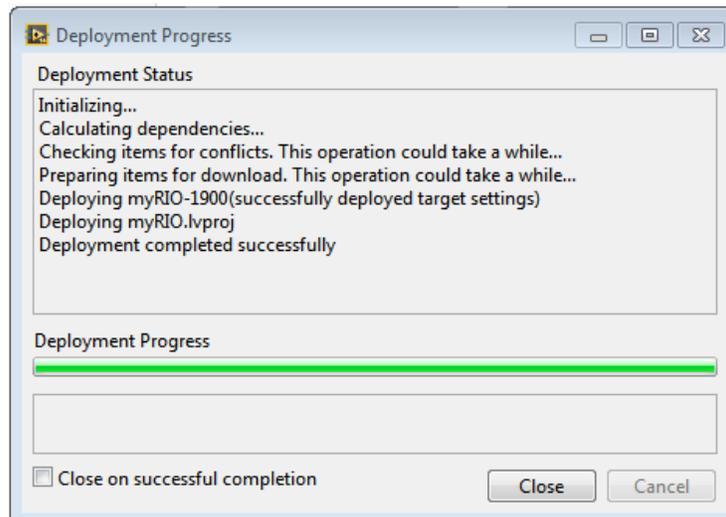
- 1、确保 myRIO 设备已使用 USB 线与计算机相连。
- 2、右键单击项目管理器界面上的 myRIO Target，如果用户在创建工程时已连接 myRIO 设备，则直接在右键菜单中选择 **Connect**。如果用户当时选择了 Generic Target，则需要 **Properties** 的弹出窗中选择 **General » IP Address/DNS Name**，输入 NI MAX 中 myRIO 设备的 IP 地址，保存后再进行 **Connect** 操作。

图 1-6. 连接 myRIO 设备



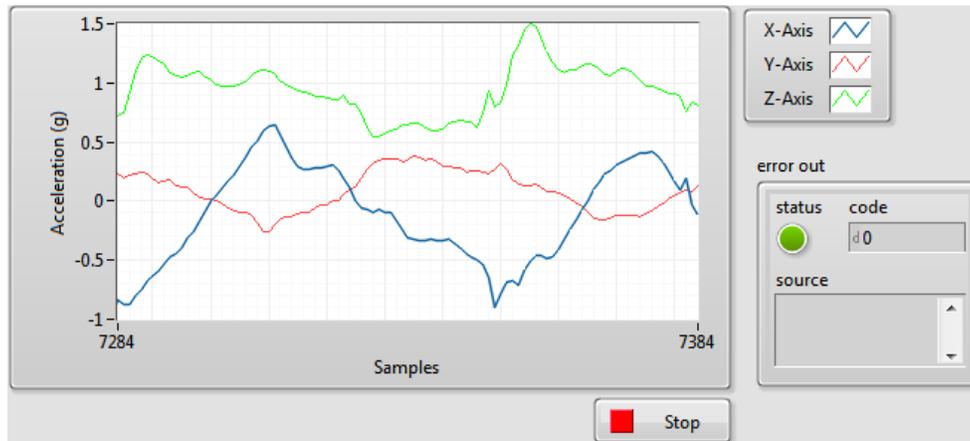
3、只有保证 myRIO Target 与计算机连接上才能编译下载程序，连接成功后单击 **Close**。

图 1-7. 成功连接 myRIO 设备



4、打开 Main.vi 程序，单击运行按钮，用户可以看到程序编译下载至 ARM 处理器上的过程，编译下载完成后单击 **Close**，程序开始运行。用户可通过摇晃摆动 myRIO 来观察图形图表中 X、Y、Z 轴上采集到的加速度数据，单位为 g，其中 Z 轴上有针对自由落体的参考系。

图 1-8. 主程序运行界面



注解 尽管本程序看似与使用数据采集卡（例如 myDAQ）的程序完成了同样的功能，但两者有着本质上的差别。使用数据采集卡的程序，程序本身是在 CPU 上运行的，数据同样也是 CPU 直接显示的。而使用 myRIO 的程序，程序运行在板载芯片上的 ARM 嵌入式处理器中，LabVIEW 底层基于网络的传输机制会自动将数据传至上位机，因此在计算机的 LabVIEW 界面上也能看到显示数据。



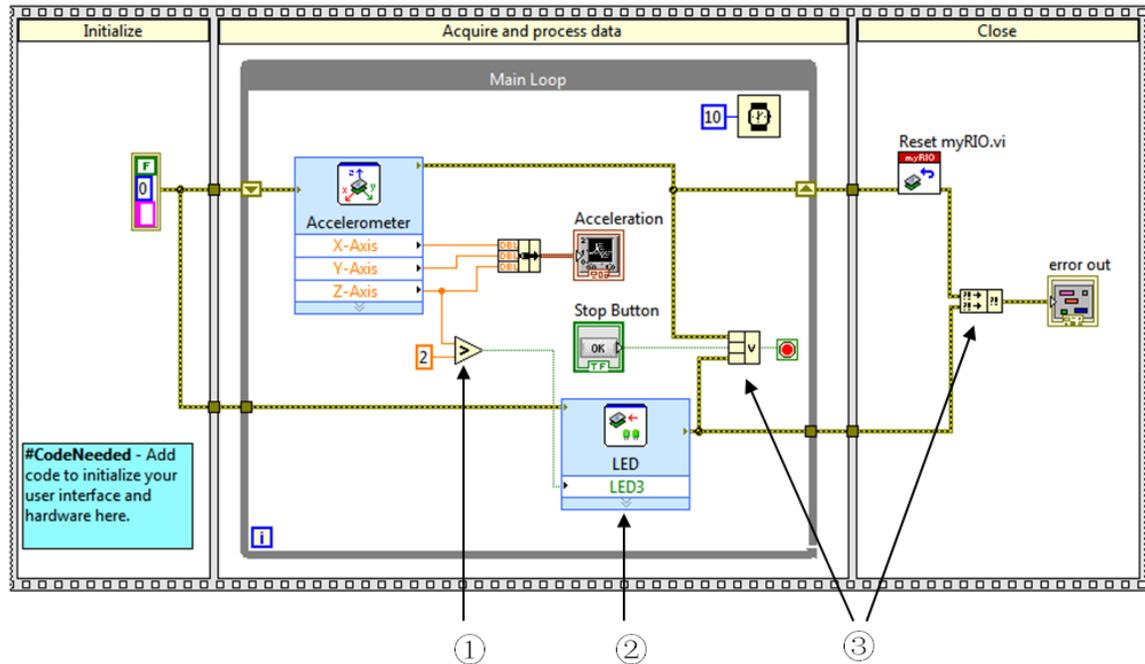
注意 由于上述 LabVIEW 的自动传输机制，程序始终会将需观测的数据从嵌入式处理器中传至计算机上，所以必然会消耗一部分资源。因此绝大部分的嵌入式程序并没有上述示例程序中的显示界面，此种方法一般适用于调试阶段，或必要的的数据监测。

为了能更好地理解程序运行在嵌入式处理器上这一概念，我们可以对示例程序稍加修改，使得修改后的程序能实现，当 Z 轴的加速度大于 2g 时，myRIO 上的一盏 LED 灯被点亮。

修改测试

1、按照下图修改示例程序的程序框图。

图 1-9. 修改后的“示例程序”程序框图



1 大于—将 Z 轴的加速度与 2g 做比较。

2 **LED 快速 VI**—在函数选板上选择 **myRIO»Default»LED**，这是一个使用板载 LED 资源的快速 VI。在快速 VI 的设置界面中，如果仅想使用 LED3，则可将 LED0~2 前的选择框勾去，完成连线后即可实现当 Z 轴加速度大于 2g 时，LED3 被点亮。

3 **复合运算、错误合并**—按照数据流将错误簇连接好。

2、确保 myRIO 与计算机之间已完成 Connect 连接操作，单击运行，下载编译程序。在 Z 轴方向施加加速度，观察 LED3 工作情况。

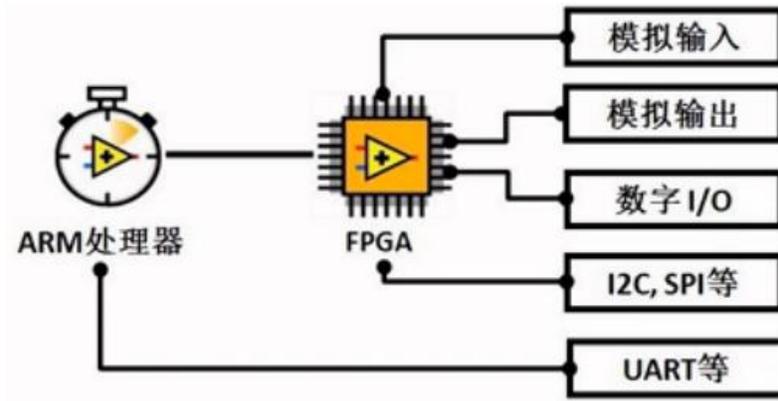
3、断开 myRIO 与计算机之间的 USB 连接线（此时程序并未终止运行），继续在 Z 轴方向施加加速度，观察 LED3 工作情况。

通过上述实验，用户可以发现当断开 myRIO 与计算机之间的连接后，计算机上的 LabVIEW 界面停止更新，但当继续给 myRIO 的 Z 轴方向上施加大于 2g 的加速度时，LED3 依旧会被点亮，就是因为程序是运行在板载嵌入式处理器上的。

在本节中，通过对示例程序的修改，用户可以对 myRIO 上 ARM 处理器的编程有初步的了解和认识。虽然在此过程中并没有直接涉及对 FPGA 资源的开发，但如若用户查看程序中所用快速 VI 的底层代码，便可发现其中已经调用了 FPGA 的接口，即研发人员在开发 myRIO Module 时，已经完善了相关的 FPGA 代码，使得用户在进行 ARM 开发时能更方便快捷。函数选板上现有的 myRIO 驱动函数已经可以满足用户与大多数外围 I/O 进行交

互并实现很多功能，如果用户有特殊需要，例如自定义的信号处理或者实时性要求非常高的控制应用，现有的函数并不能满足要求，那么用户可以考虑自定义 FPGA 上的程序，这部分内容会在后面有所介绍。

图 1-10. myRIO Module 中 ARM 处理器与 FPGA 资源的关系



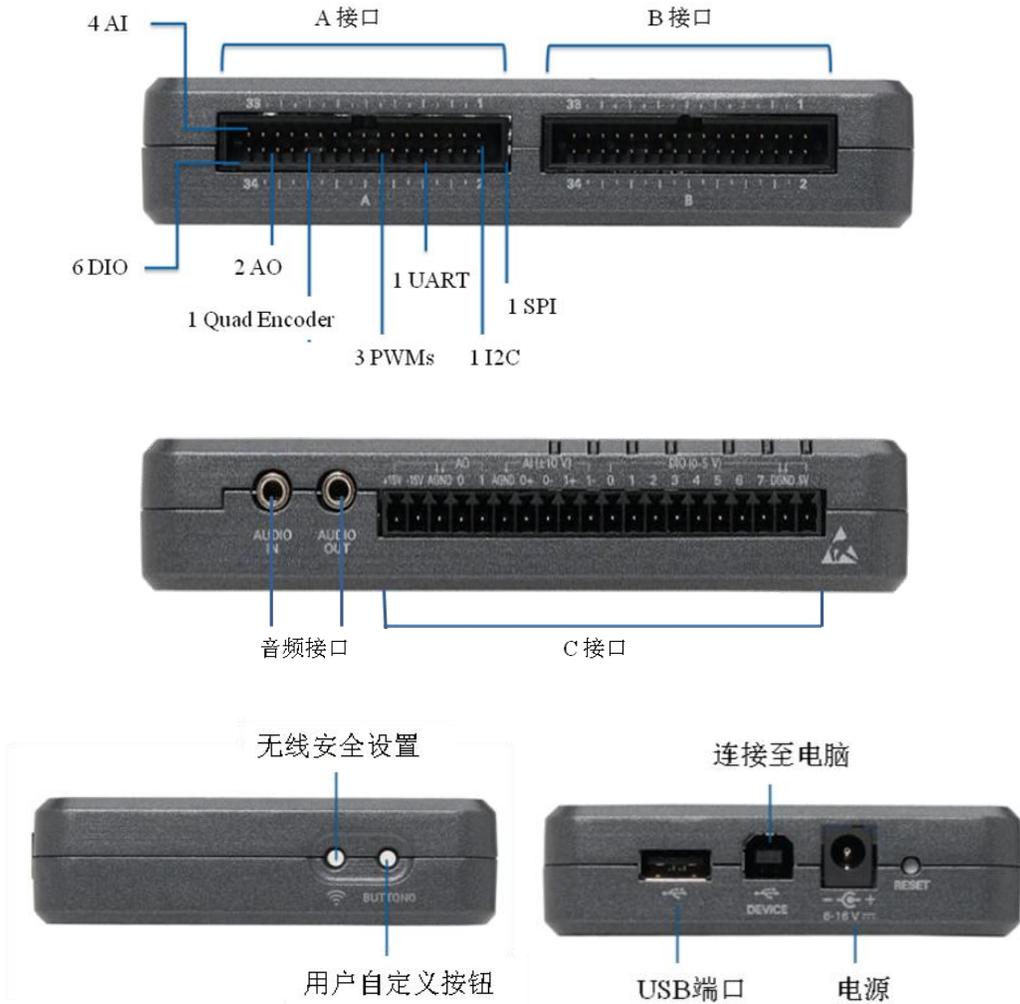
2 连接外围 I/O

上一章中，我们采集的加速度信号来自板载的加速度传感器，用于指示的 LED 灯同样也是板载资源。在本章中则将介绍利用 myRIO 的 I/O 接口来连接用户的传感器和执行机构。

NI myRIO 接口介绍

首先需要了解 myRIO 设备四侧的接口：

图 2-1. myRIO 侧面接口



具体的针脚定义，用户可以通过查阅 myRIO 的 USER GUIDE AND SPECIFICATIONS 来了解。此文件可以通过 NI 官网搜索下载。用户具体连接传感器或者执行机构的方式，可通过在 NI 官网主页搜索 **myRIO Project Essential Guide**，下载该指导书获得参考。

此外，myRIO 拥有丰富的外围配套套件，可以在购买 myRIO 的时候一起选购，这些套件可以帮助用户在完成 myRIO 开发的过程中更加方便快捷。

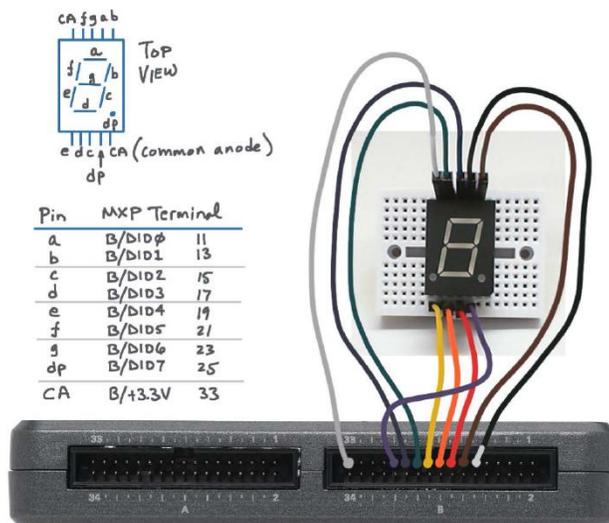
图 2-2. myRIO 的外围套件

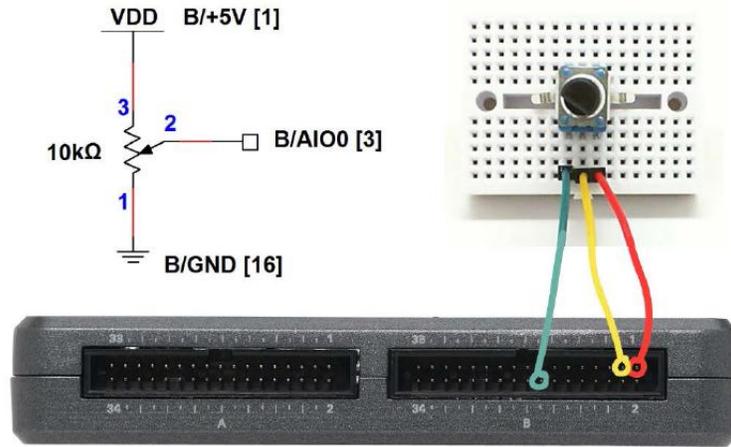


硬件连接

本节中，我们将从**起步套件**里选取两个简单的元器件：七段数码管和电位器，用户可以结合套件中的面包板，也可以利用开发盒中的外围扩展板，将元器件连接到 myRIO 的 B 接口的 I/O 管脚上。具体的连接方法参考 **myRIO Project Essential Guide » Seven-Segment LED Display** 和 **Potentiometer**。

图 2-3. 7 段数码管和电位器与 myRIO 的连接方法





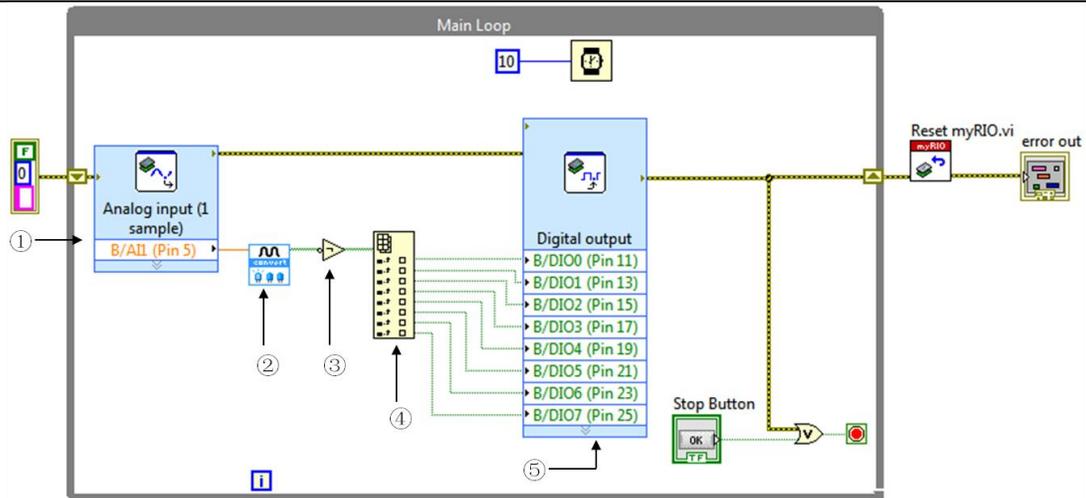
在本实验中，电位器使用 AI1 通道，通过读取 AI1 通道的输入来控制 7 段数码管的显示。给电位器加上 3.3V 的电压，将其输出端接至 AI1 通道，则模拟输入通道 AI1 读取的就是变化的电位信息。从而当用户旋转电位器旋钮时，数码管的显示能逐渐从 0 加至 9。

软件实现

请按以下步骤编写程序：

- 1、与上一章类似地，在 LabVIEW 中新建一个 myRIO Project，工程名可为：**7-seg display**，软件自动生成如上一节中所示的读取三轴加速度的 Main.vi 程序模板。
- 2、对模板做一些修改：去掉顺序结构和不需要的程序部分，按图重新连好程序框图。

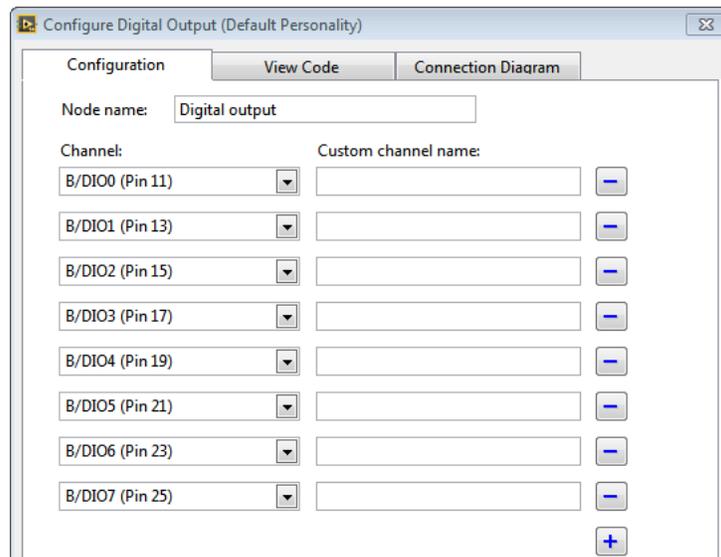
图 2-4. 7-seg display 程序框图



- 1 **模拟量输入**—在函数选板选择 **myRIO » Default » Analog Input**，通过模拟量输入端口读取电位器的输出电压，在配置界面选择输入通道 **B/AI1 (Pin 5)**。
- 2 **子 VI**—把 AI1 读取到的电压转换成数码管对应的显示输出。因为编程稍为繁琐，所以已将这部分程序设计为子 VI 提供给用户直接使用。用户可以将 Solution\7-seg display\AI

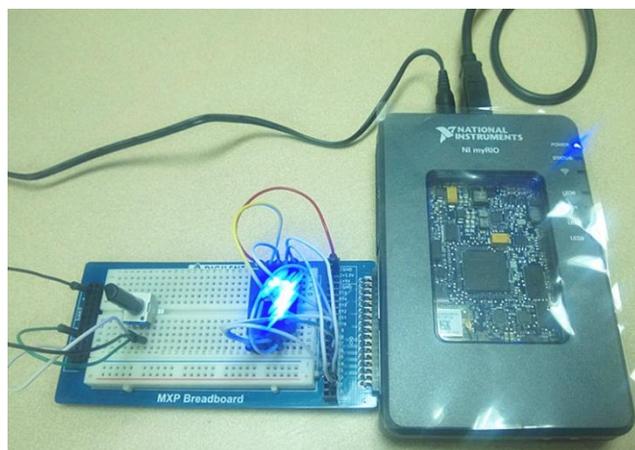
to LED Converter.vi 直接拖入工程，也可以在工程管理器界面右击 myRIO 添加文件。在子 VI 的程序框图中有较为详细的程序注释，程序输出为布尔型的一维数组。

- 3 非门—由于所用为共阳极数码管，即输入为低时有效，因此需要将此处数组取反。
- 4 索引数组—将一维数组里的元素逐个赋予 myRIO 开关量输出端口来控制数码管。
- 5 数字量输出—在函数选板选择 **myRIO » Default » Digital Output**，根据 Project Essential Guide 中的接线方式，实质上是将数码管端口 a~g 分别接到了通道 DIO0~ DIO6 上，所以可以按照图示添加通道，注意此处还有七段数码管的一个点的通道。



3、保存程序，将 myRIO 与计算机相连，单击运行，程序自动编译下载。调节电位器，观察数码管显示情况，断开 USB 连接线，可发现程序仍然在 myRIO 上运行，再一次验证了程序是运行在实时处理器上的。

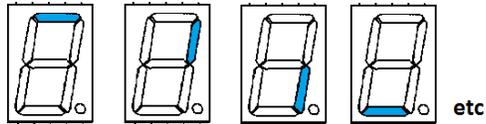
图 2-5. 7 段数码管和电位器的工作实物图



其实目前嵌入式应用中使用数码管的情况并不多了，更多地使用 LCD 液晶屏直接通过 UART 端口与 myRIO 通信，也很方便快捷，而七段数码管会一次性占用 8 个通道，本实验仅做练习使用。

思考

此处用户可以尝试做一个拓展实验，通过修改程序，使其能实现一个跑马灯的效果，即使用数码管的 6 个 LED 灯，能根据电位器的调节以不同速率循环显示，并且允许用户通过前面板按钮改变跑马灯的方向。



3 NI myRIO 的 WiFi 连接功能

myRIO 不仅可以通过 USB 线缆与计算机相连的，还可以通过 WiFi 来实现连接。在本章中，将介绍如何通过 WiFi 来实现两者的连接。

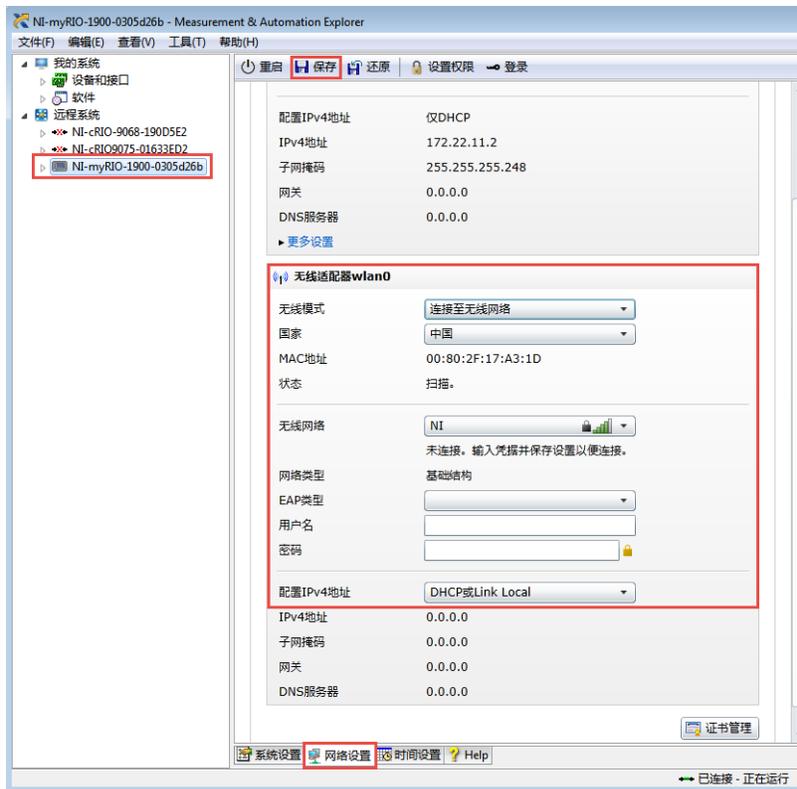
3.1 利用已有网络的 WiFi 连接

配置 WiFi 连接

请按以下步骤完成配置：

- 1、在完成 WiFi 连接配置前还是需要用 USB 线缆连接 myRIO 与计算机。
- 2、打开 NI MAX。在远程系统中找到此时用 USB 线缆连接的 myRIO，单击选中。在右侧配置管理界面中，选择**网络设置**选项卡。在以太网适配器一栏可以看到 myRIO 用 USB 连接时虚拟网口的相关信息。

图 3-1. 对 myRIO 进行网络配置



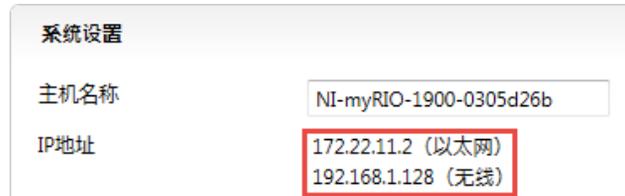
- 3、在无线适配器一栏进行配置。无线模式选择为**连接至无线网络**，国家为**中国**，选择要接入的无线网络后，按需要输入用户名和密码等信息，配置 IPv4 地址可选择为**静态**或是**DHCP 或 Link Local**。点击**保存**后可发现状态变为已连接至所选无线网络，完成配置。



注意 此处 myRIO 必须和上位机接入同一个无线网中。

返回系统设置选项卡，点击刷新后可发现，在 IP 地址一栏，除了因 USB 连接获得的虚拟网口地址（以太网）外，无线地址也已经分配得到。

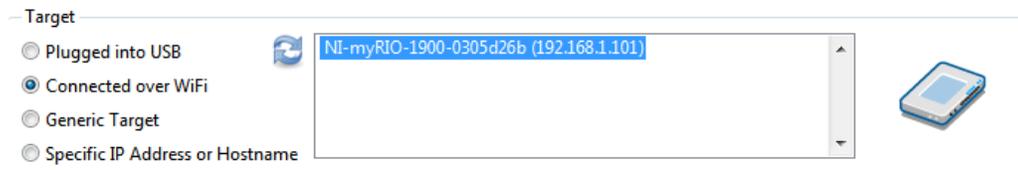
图 3-2. myRIO 的 IP 地址



运行实例

1、打开 LabVIEW，按照第一章中的方法新建一个 myRIO 工程模板。用户可以发现，除了按照之前的方法通过 USB 的方式搜索到目标设备之外，现在还可以通过 WiFi 搜索到。即使用户断开 USB 线缆连接，可发现此时也可以通过 WiFi 搜索到设备。

图 3-3. 通过 WiFi 搜索到 myRIO 设备

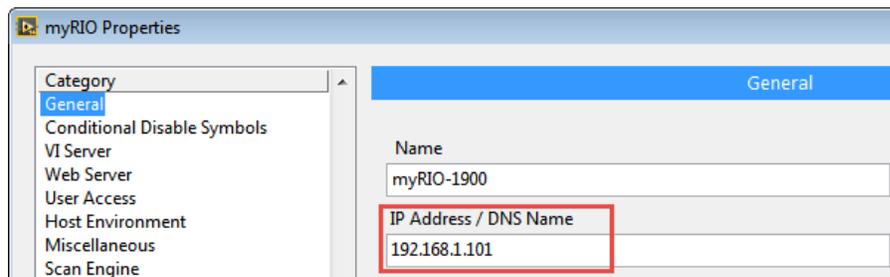


2、创建工程名为 myRIO WiFi，单击 Finish，完成通过 WiFi 连接的 myRIO 工程的创建。



注意 在工程浏览器窗口，设备名后的地址为即为 WiFi 地址。如果在以后的使用中 WiFi 地址有变化，则可以通过右击设备名 » **Properties** » **General** » **IP Address/ DNS Name** 修改。

图 3-4. myRIO 属性菜单界面



3、打开示例程序 Main.vi，点击运行，编译下载。程序能在 ARM 处理器上正常运行。

3.2 配置 myRIO 为热点的 WiFi 连接

上述方法实际上是通过一个外置的无线路由器来实现 WiFi 连接的，而 myRIO 自身还可以被配置为一个 WiFi 热点，上位机和其他智能终端都可以通过其发射的无线网络连接至 myRIO 上，这样就不需要再通过第三方的无线路由器来实现连接，在某些应用中会显得更加便捷，例如车载应用等。

图 3-5. myRIO 实现 WiFi 连接的不同方式



注意 配置 myRIO 为热点的功能需要 **NI-RIO 13.1 或更高版本** 的驱动支持。

升级驱动版本

下面将介绍一下如何升级 NI-RIO 驱动，如果当前驱动版本已经满足要求则不需要进行升级，可跳过此步骤。

1、确保 myRIO 已经连接至计算机，打开 NI MAX，展开**远程系统**下的当前 myRIO 设备，找到**软件**项下的 NI myRIO 驱动，可查看当前驱动的版本号。如需升级，则需要先在计算机本机上升级驱动版本。

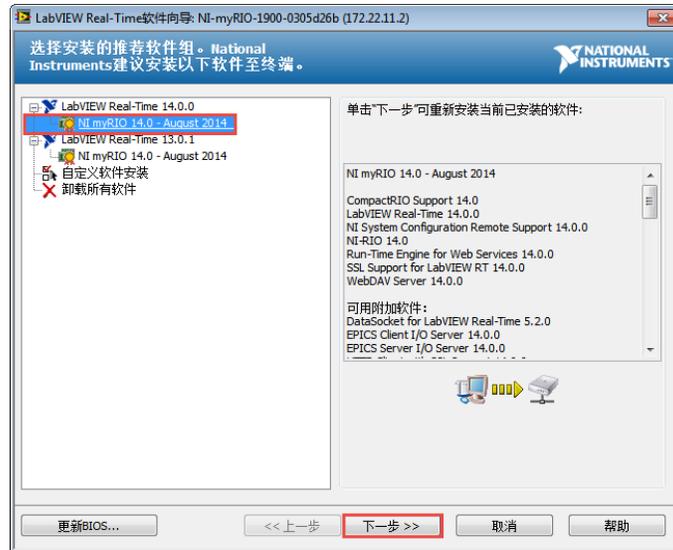
图 3-6. 在 NI MAX 中查看 myRIO 的驱动版本号



2、在 NI 官网上搜索 NI RIO 13.1（或更高版本），找到 [NI-RIO Device Driver](#) 的驱动文件，或更高版本。在计算机本机上下载安装好更新的驱动后，可以在 **NI MAX » 我的系统 » 软件** 中查看到更新后的驱动。

3、在 NI MAX » 远程系统 » myRIO 下右击软件，选择添加/删除软件，在弹出的 LabVIEW Real-Time 软件向导界面，选中新版本的 NI myRIO 驱动，点击下一步。

图 3-7. LabVIEW Real-Time 软件向导界面



注意 点击下一步后的界面是可选的需安装的软件组，如果用户安装的 LabVIEW 是中文版本的话，需要将**软件组附加软件 » LabVIEW RT Add-ons » Language Support for Simplified Chinese** 勾选上。

4、再次点击下一步后将进行软件同步更新，保证 myRIO 系统中的软件模块与上位机中的一致，下载编译时不会发生冲突。

5、安装完毕后点击完成，用户可在远程系统中再次查看驱动版本号。

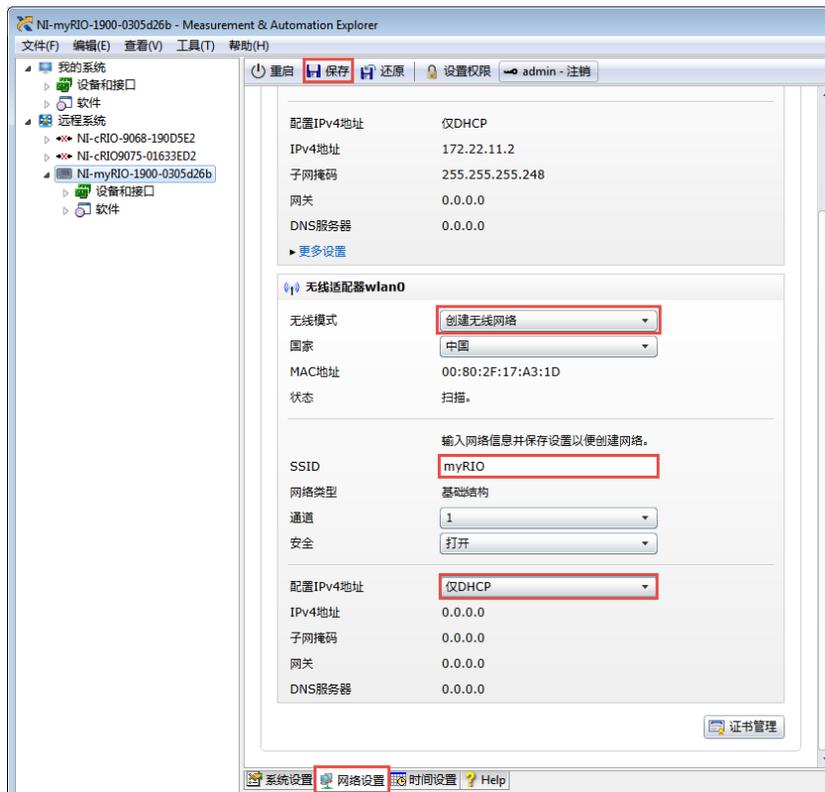
配置接入网络

建议在此暂时先连上 USB 线缆再进行一下操作。

1、同样地，在 NI MAX 中进入目标 myRIO 的**网络设置**选项卡界面，在无线适配器一栏进行更改配置。

2、更改无线模式为**创建无线网络**，即在 myRIO 上创建一个无线网络的接入点。SSID 为创建的无线网络名，可取为 myRIO。在直接接入模式下，需更改配置 Ipv4 地址为**仅 DHCP**。

图 3-8. 对 myRIO 进行网络配置



3、点击保存，可发现状态为正在广播 myRIO，同时会出现新的 Ipv4 地址，这是 myRIO 作为一个无线接入点分配的地址。

此时 myRIO 已工作在无线接入模式下，可以将其理解为一个自定义的热点，第三方设备便可以连接到此无线 AP 上。在装有无线网卡的上位机中，可以直接通过无线网络连接功能，与 myRIO 无线网络进行连接。此后可以再次断开 USB 线缆，与使用第三方无线路由器时类似的，创建 myRIO 模板项目，通过 WiFi 找到目标硬件后，使用示例程序进行验证。

myRIO 的无线连接以及其作为无线 AP 的功能不仅是为了开发方便，更重要的是，利用上述功能，我们可以在开发某些应用时，通过无线设备与 myRIO 通信从而获得其数据状态等信息以及对其进行控制。例如要使 myRIO 通过 WiFi 与其他具有无线网卡的计算机相连，可以通过 LabVIEW 的网络共享变量，Data Socket 技术，TCP 或 UDP 协议等技术方式来实现。而通过拥有无线功能的智能终端进行控制的功能将在后面有所介绍。

4 图像采集以及视觉算法应用

4.1 通过 USB 摄像头采集视频

在前面几章中，主要介绍了 myRIO 通过自带的模拟和数字量 I/O 与外部传感器进行通信的功能。本章将介绍通过 myRIO 上的 USB 端口连接摄像头来采集图像的功能。

安装驱动程序

要实现本章功能，首先需要确保已经在上位机上安装了 **NI Vision Acquisition Software**（视觉采集软件），软件安装之后用户可以在 NI MAX 中 **我的系统 » 软件** 下查看到有 NI-IMAQdx。如果后续需要做开展图像处理工作，建议同时也可以安装 **LabVIEW Vision Development Module**（视觉开发模块），软件安装之后可在 NI MAX 中 **我的系统 » 软件 » LabVIEW 2014** 下查看到 Vision Development Module。以上两个软件的安装程序均可在随盒光盘 DVD1 的 Distribution 目录下找到，其文件夹名分别为 VisionAcq 和 Vision（第 0 章中有相关介绍）。

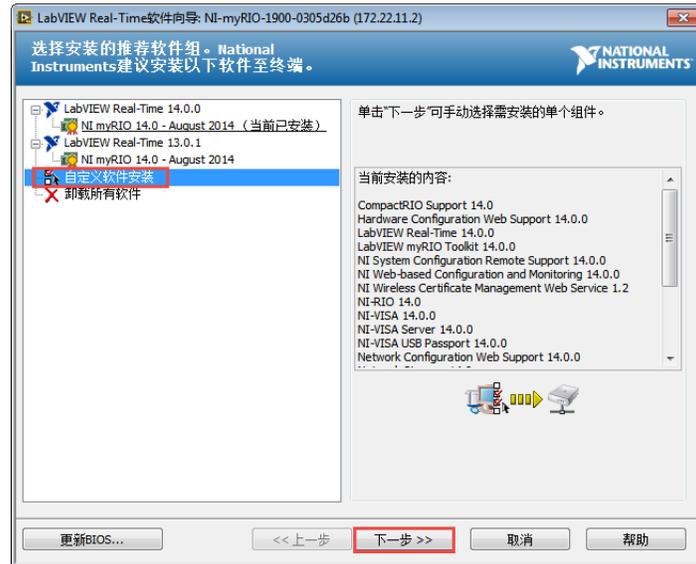
图 4-1. 在 NI MAX 中查看驱动程序



在上位机上安装完软件之后，根据以下步骤在 myRIO 上安装相应驱动。

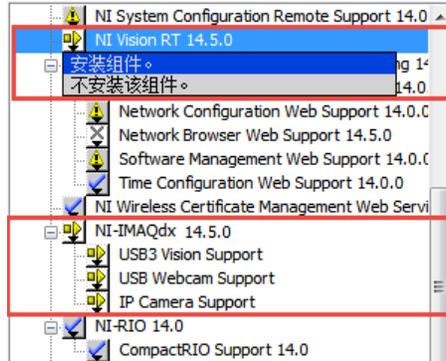
- 1、将 myRIO 通过 USB 线缆或 WiFi 与计算机相连。
- 2、与升级驱动版本类似的，在 NI MAX 中 **远程系统 » myRIO** 下右击 **软件**，选择 **添加/删除软件**，在弹出的 LabVIEW Real-Time 软件向导界面，选择 **自定义软件安装**。

图 4-2. LabVIEW Real-Time 软件向导



3、在选择需安装的组件中分别找到 **NI-IMAQdx** 和 **NI Vision RT** 两个组件模块，右击选择**安装组件**后点击下一步，系统自动将两个模块需要用到的组件和驱动安装到 myRIO 中。

图 4-3. 选择需安装的组件



4、点击更新完成后，将 USB 摄像头通过 myRIO 的 USB 端口相连，便能在 NI MAX 中**远程系统 » myRIO » 设备和接口**下查看到 USB 摄像头，记住设备名，例如本实验中摄像头设备名为“cam0”。选中之后在右侧界面选择 **Acquisition Attributes** 选项卡进行测试，用户可以点击 **Snap** 单帧采集或是 **Grab** 连续采集，还可以通过 **Video Mode** 对参数进行修改，修改参数后需要点击保存当前参数。

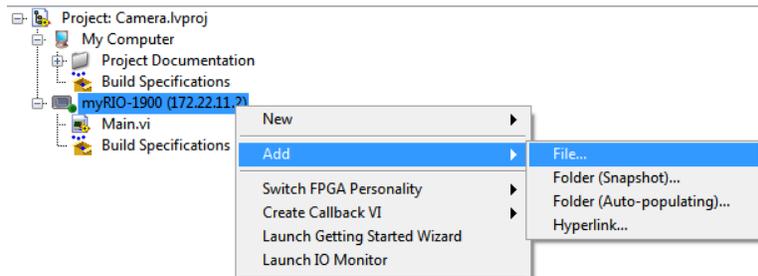
图 4-4. 测试 USB 摄像头



运行实例

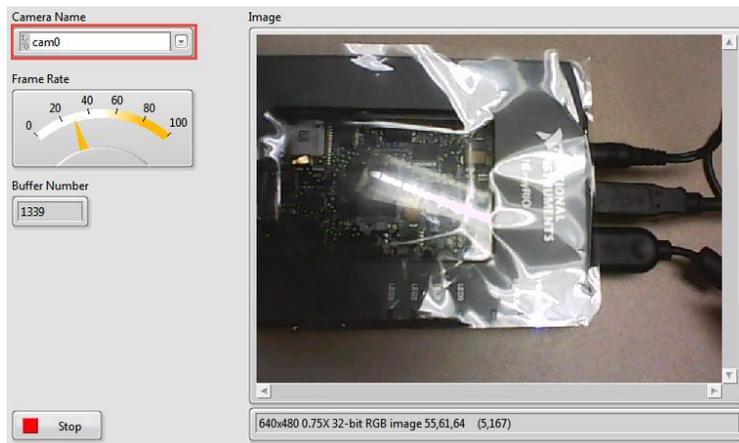
- 1、打开 LabVIEW » Help » Find Examples » Hardware Input and Output » Vision Acquisition » NI-IMAQdx，在此目录下可以找到相关范例。例如查看使用底层 VI 所写的程序，可继续向下打开 **Low-Level » Low-Level Grab.vi**，这是一个连续采集的程序。双击打开后另存，以免修改范例查找器中的示例程序。
- 2、参考之前方法创建一个 myRIO 模板项目。将另存的示例程序添加到 myRIO 的目标下面，通过右击弹出菜单中 **Add » File**，选择对应程序。此时程序位于 myRIO 的目标下，因此它将运行在嵌入式处理器上。

图 4-5. 将示例程序添加至工程中



- 3、双击打开程序，在 Camera Name 中选择我们所使用摄像头在 myRIO 下的设备名。点击运行，开始连续采集图像。

图 4-6. 示例程序运行时的前面板



4、程序停止运行后，用户可尝试在 NI-MAX 中修改 myRIO 所连接摄像头的设置，改变分辨率的值，点击保存后再次运行程序，观察分辨率的变化。

 **注意** 保存修改后，需要切换掉 NI-MAX 选中的摄像头对象，例如点击选中远程系统，否则运行 LabVIEW 程序时会因为 NI-MAX 也在访问摄像头资源而报错。

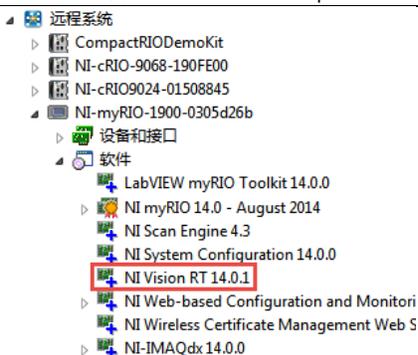
 **注解** 示例程序中的底层函数都可以在**函数面板 » Vision and Motion » NI-IMAQdx** 中找到，利用范例查找器中的现有程序可提高开发者效率。

需要再次强调的是，虽然我们能在上位机界面上看到采集的图像，但程序本身是运行在实时处理器上的，myRIO 通过网络传输机制将图像传输给上位机进行显示。所以即使删除了前面板上的图像显示控件，图像的采集还是继续进行的，因此我们可以在实时处理器上完成对图像的分析处理。

4.2 基于 ARM 的机器视觉应用

机器视觉应用的开发需要使用到在上一个小节中安装的 LabVIEW 的 Vision Development Module。用户可在 NI-MAX » 远程系统 » NI-myRIO » 软件下查看 NI Vision RT 的安装。

图 4-7. 在 NI-MAX 中查看 Vision Development Module 的安装



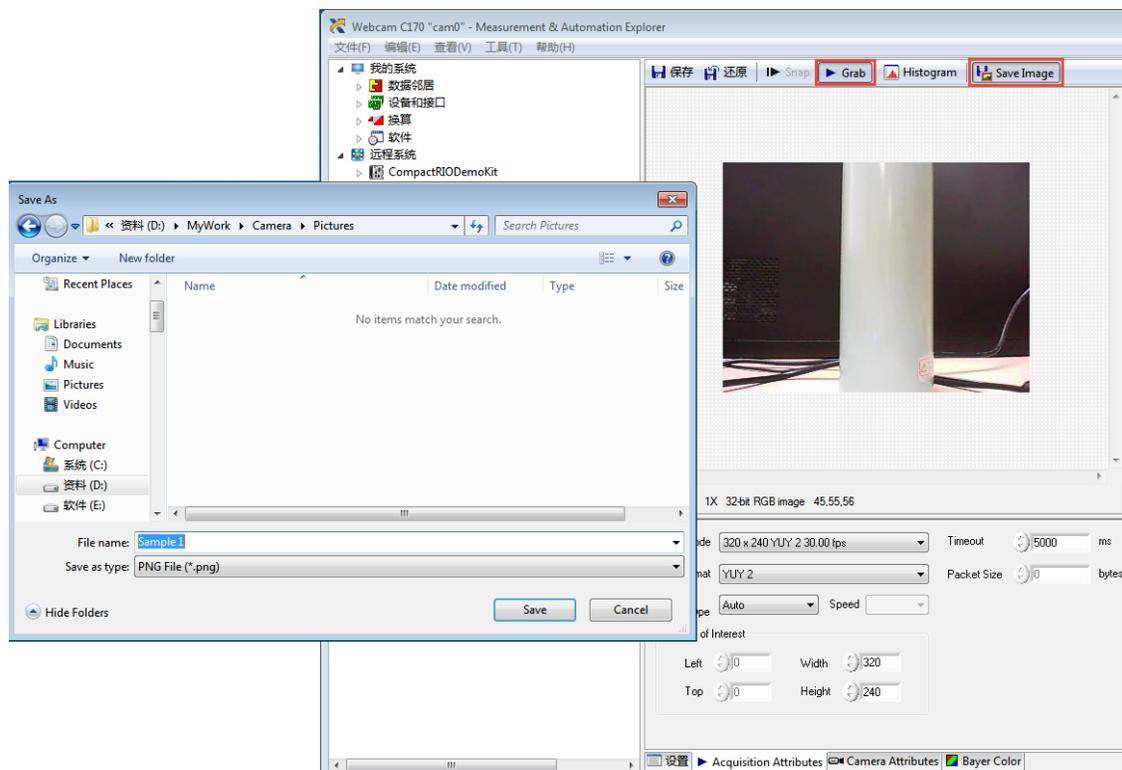
机器视觉的应用的开发可在之前采集的程序基础上，增加机器视觉相关的算法来实现。用户可以在程序框图中打开函数选板» Vision and Motion » Image Processing / Machine Vision 来查看使用这些算法函数。尤其在 Machine Vision 这个选板中，LabVIEW 提供了丰富的功能算法，例如边缘检测，颗粒分析，模式匹配，文字识别等，用户可以通过帮助文档来学习这些算法的应用，也可以打开范例查找器» Toolkits and Modules » Vision 学习相关范例。

实验操作

目的：通过连接到 myRIO 上的 USB 摄像头检测前方的瓶子是否已被拿走，如果被拿走则报警。请按以下步骤进行实验操作：

1、首先进行边缘检测，打开 NI-MAX，找到 myRIO 目标下连接的摄像头。点击 **Grab** 进行连续采集，再分别点击 **Save Image** 保存正常情况下（初始位置以及稍微挪动一点的位置）和非正常情况下（即将拿走以及完全拿走）的图片作为开发时的素材库进行验证。

图 4-8. 保存正常情况下以及非正常情况下的图片

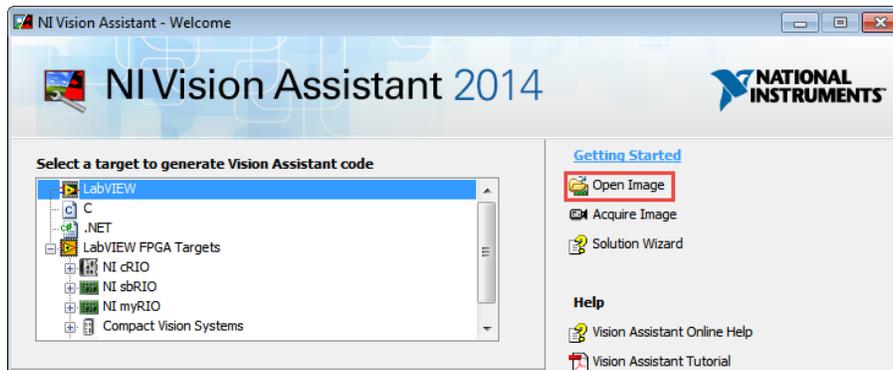




2、采集完所需图片素材后，再次点击 Grab 关闭连续采集，切换掉选中的摄像头。

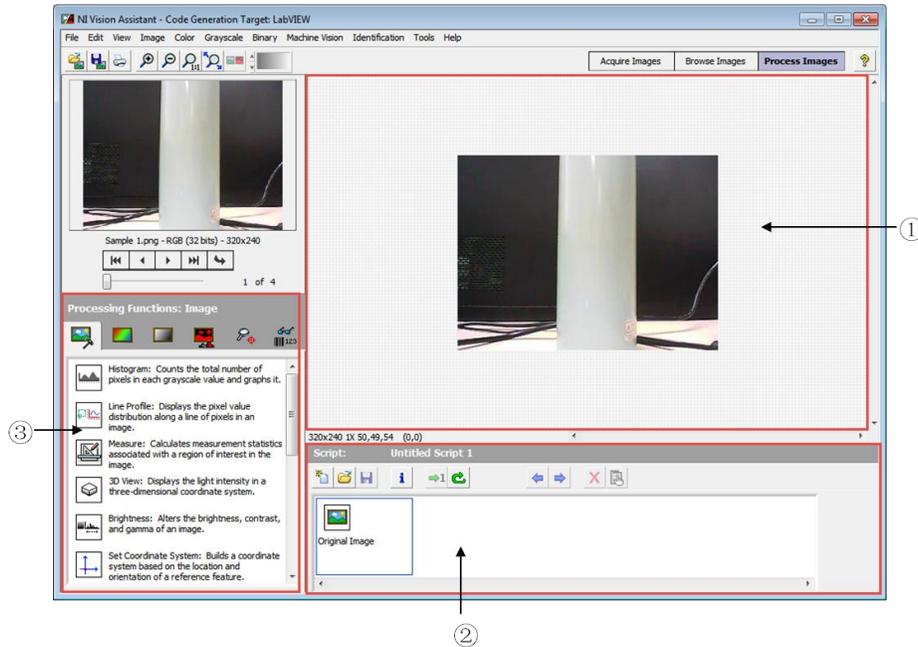
3、Vision Assistant 是 LabVIEW 提供的一个帮助我们高效开发机器视觉应用的工具，可以在不编程的情况下验证相应算法的有效性并将验证步骤转化为代码，因此可以帮助我们加速机器视觉应用的开发。这里，我们打开 Windows 的**开始**菜单，选择 All Programs » National Instruments » Vision » **Vision Assistant**。打开后会弹出一个提示窗口，询问用户待测试图片的来源。因为已经在 NI-MAX 中保存了用于测试的图片，所以直接点击 **Open Image**，打开刚才保存的 4 张（根据用户实际保存图片的数量）不同情况下的图片。

图 4-9. 在 Vision Assistant 中打开测试图片



4、进入 Vision Assistant 主界面，4 帧图片已载入打开。

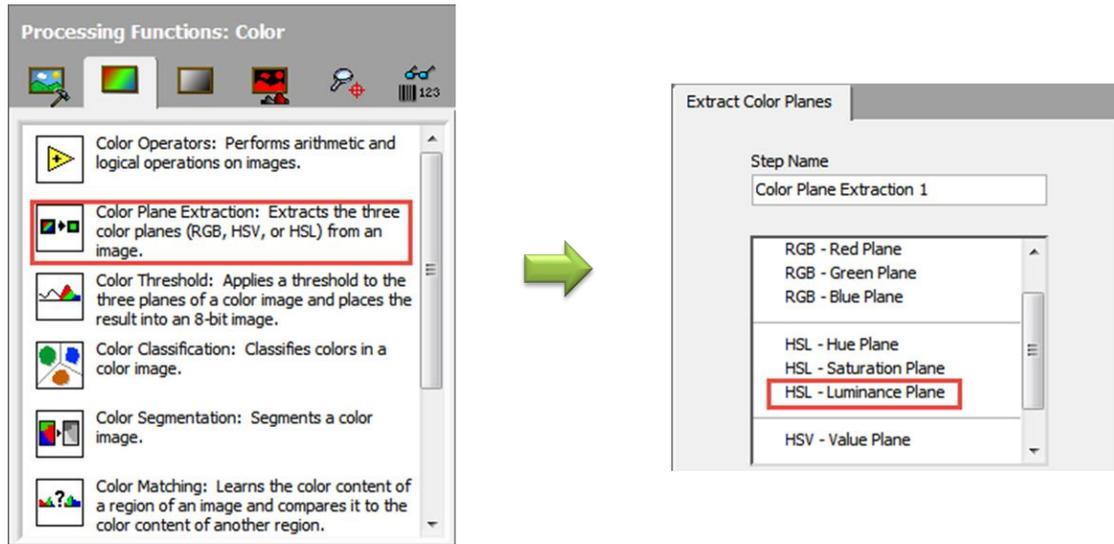
图 4-10. Vision Assistant 主界面



- 1 图片检测区域—显示每一步处理后图像相应的变化。
- 2 **Script 区域**—记录每一次操作步骤。
- 3 **函数选板**—提供了各种图像处理的函数，例如针对色彩处理的函数，针对灰度图像处理的函数，针对二值图像处理的函数等。

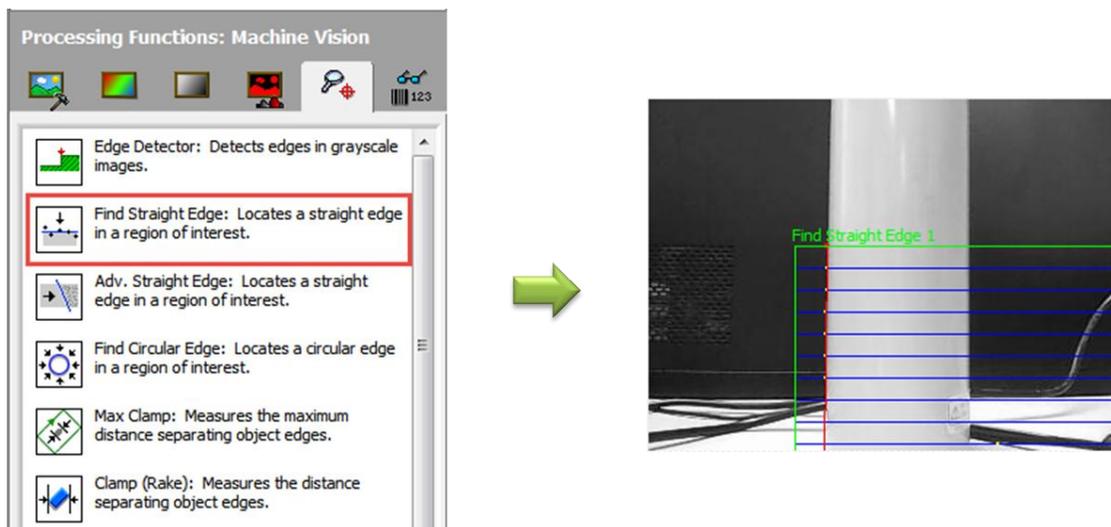
5、检测瓶子是否存在，可通过模式匹配，色彩识别等方法，但因为程序最终需要运行在 ARM 处理器上，所以我们可以选取一种计算量相对较小的图像检测方法，即检测瓶子图像的边沿是否存在，以此作为判断依据。由于边缘检测此类函数只能针对灰度图像进行，因此首先需要提取彩色图片的亮度信息，使用 **Color** 函数选项卡下的 **Color Plane Extraction** 函数，选择 **HSL - Luminance Plane** 后点击 OK 将其变为灰度图像。

图 4-11. 将彩色图片转换为灰度图像



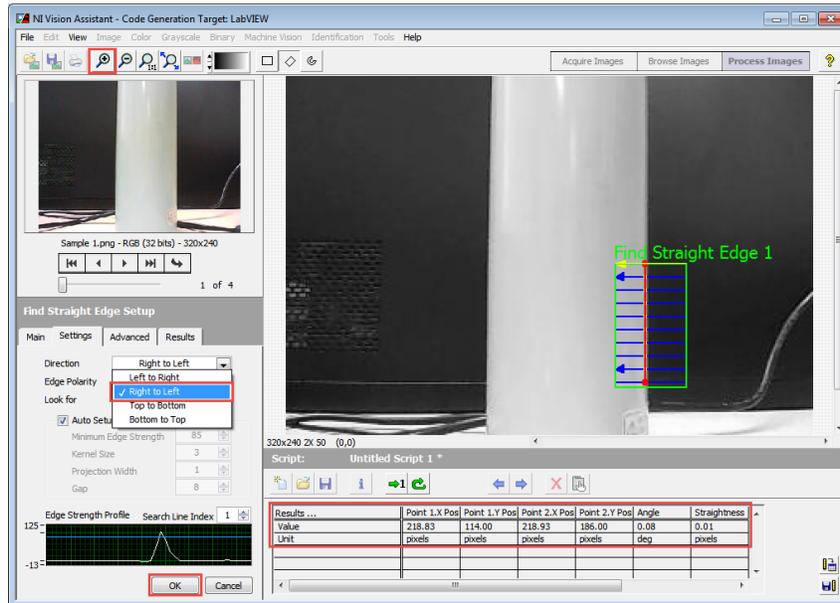
6、使用 **Machine Vision** 函数选项卡下的 **Find Straight Edge** 函数提取瓶身边缘。点击之后图片上出现绿色的 ROI 区域，即用户感兴趣的区域。

图 4-12. 加入边缘检测函数



7、用户可以重新选择 ROI，为方便处理，可先将图片放大。重新选中绘制好 ROI 后，可在左侧一栏中选择检测方向，例如从右往左检测边缘。而在图片下方的一栏中会显示检测结果，包括检测到的直线上两点的左边以及边缘直线的斜率等信息，点击 OK 后完成。

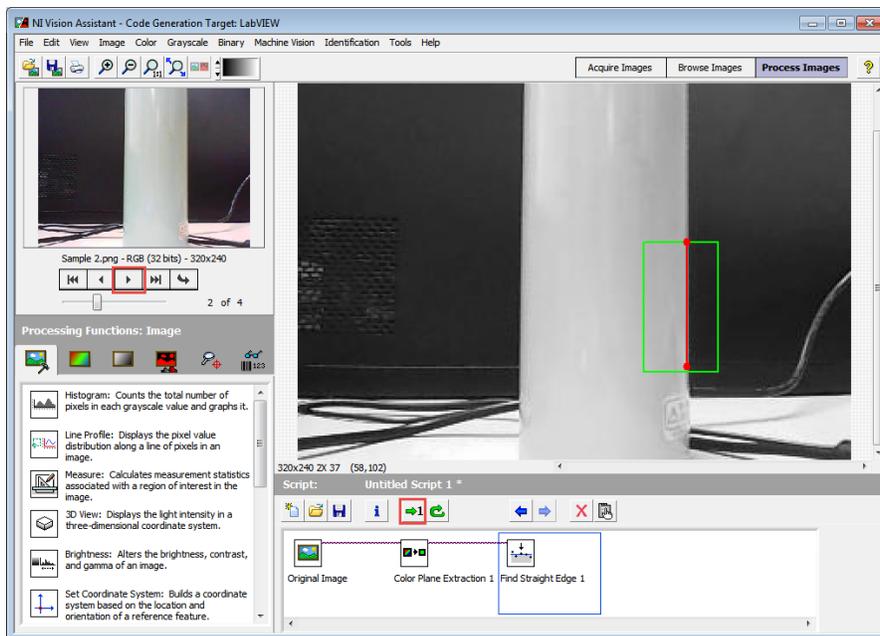
图 4-13. 设置 ROI



注意 ROI 的选取非常重要，如果区域太大，可能在非正常情况下也检测到边缘；如果区域太小，可能导致瓶子稍微有位移就错过需检测边缘。

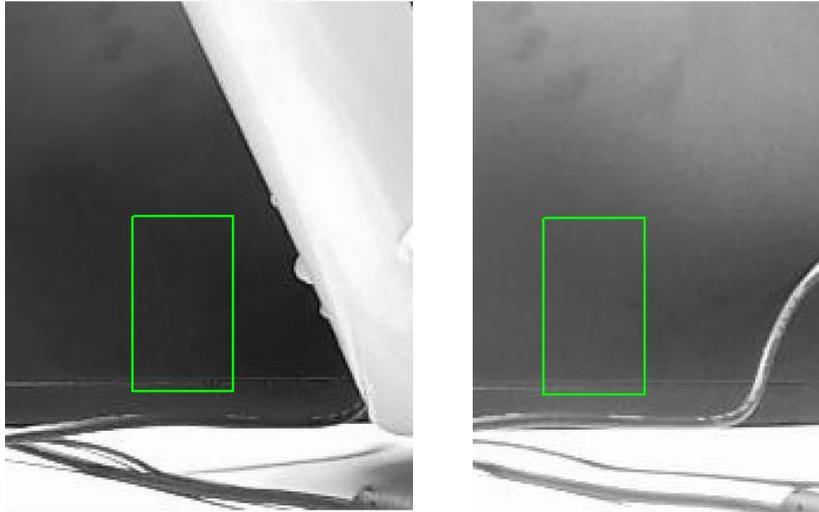
8、判断瓶子是否被拿走即判断是否能检测到边缘，当前已针对一张图片进行了检测，可再检测一下其他的图片以验证是否能正确做出判断：选中第二张图片，在 **Script** 区域点击 **Run Once**，可发现仍然能检测出边缘。

图 4-14. 针对不同情况下的图片进行边缘检测



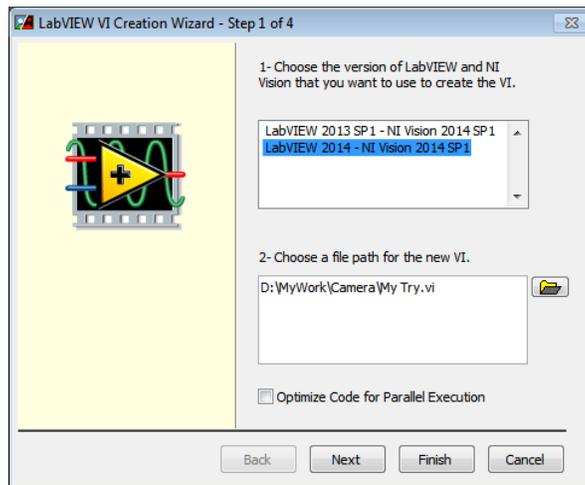
9、对第三张第四张图片也进行类似的操作，可发现在这种非正常情况下检测不到边缘。

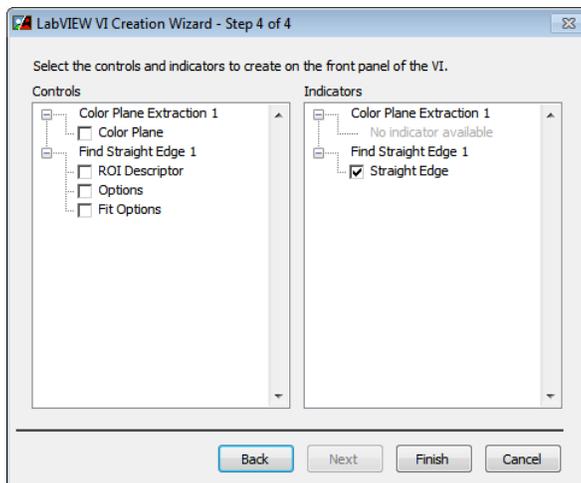
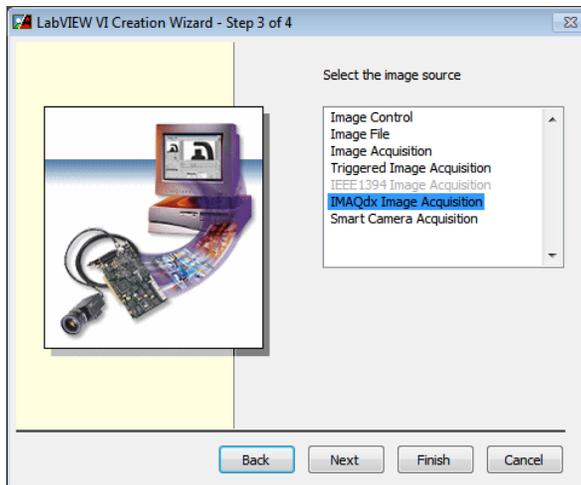
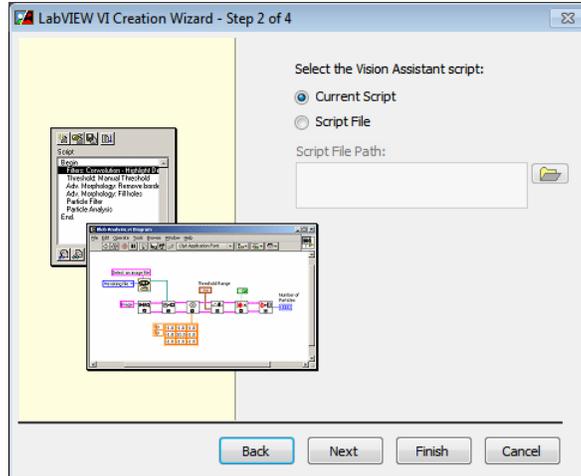
图 4-15. 非正常情况下的图片检测不到边缘



10、完成验证后，可将检测步骤保存以备后用。选择 **File » Save Script** 进行保存。NI Vision Assistant 不仅能帮助我们选择合适的算法和参数还能将一系列操作步骤自动生成 LabVIEW 代码：点击 **Tools » Create LabVIEW VI**，Step1 中选择生成路径时可选择上一小节创建的项目路径；Step2 中选择默认的 **Current Script**；Step3 选择图片来源，选择 **IMAQdx Image Acquisition** 即可；Step4 供用户选择在生成的 LabVIEW 程序中哪些参数作为输入控件，哪些参数作为显示控件，如果不进行选择，将根据 NI Vision Assistant 软件中的设置采用常量作为默认值，由于我们关心其输出值，因此将 **Straight Edge** 勾选上。点击 **Finish** 后将自动生成所需 VI。

图 4-16. 从 NI Vision Assistant 中自动生成 LabVIEW VI



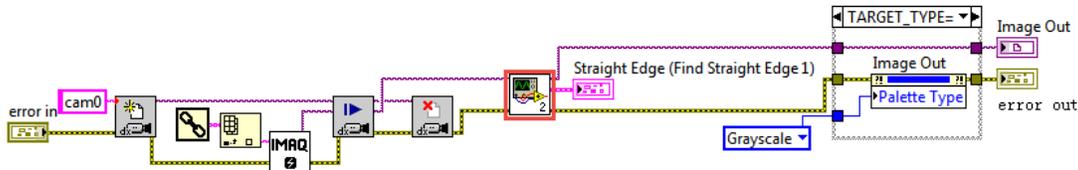


完成边缘检测的相关操作并将操作步骤自动生成 LabVIEW 代码后，打开程序框图查看生成的 VI，可发现程序首先完成了图像采集的工作，从函数 IMAQ ExtractSingleColorPlane（提取亮度信息）开始即是根据 Script 转换而来，通过函数 IMAQ Find Edge（提取边

缘)，最后将图像进行显示。我们需要将此 VI 中图像处理的程序和上一小节中图像采集的程序整合成一个可以运行在 myRIO 上的完整的程序，请按以下步骤进行：

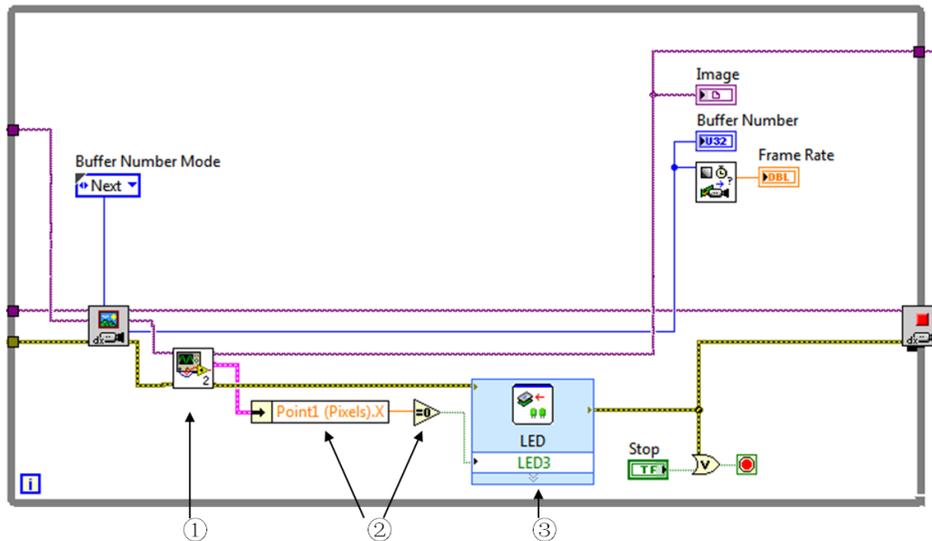
- 1、因为需要图像处理的程序复制到图像采集的程序中去，为使界面简洁可读性强，可将这部分代码写成一个子 VI。选中整个图像处理分析的部分（不包含图像显示部分），点击 **Edit » Create SubVI**，生成子 VI 后双击进入，另存为 Edge Detection.vi。

图 4-17. 生成子 VI 后的 Script 程序



- 2、打开上一小节创建的工程，将子 VI 添加至 myRIO 的 Target 下。
- 3、打开图像采集的 VI，在采集图像的函数后添加相应的处理模块。按住 Ctrl 键和鼠标左键，向右下方拖动画框，扩大 While 循环的区域，将程序另存为 Main 主程序。
- 4、按照下图修改主程序的程序框图。

图 4-18. 主程序的程序框图



- 1 子 VI—将采集到的图像先进入子 VI 进行处理，然后再显示输出。
- 2 按名称解除捆绑、零运算—提取处理结果簇中直线第一个点的 X 坐标信息作为判断依据，为零则表示没有检测到直线，非零则表示检测到边缘。
- 3 LED 快速 VI—零运算输出为真时表示没有检测到边缘即瓶子被拿走，可以利用 myRIO 板载的 LED 进行输出报警。选择函数选板 » myRIO » Default FPGA Personality » LED，勾选 LED3，连接好相关数据线。

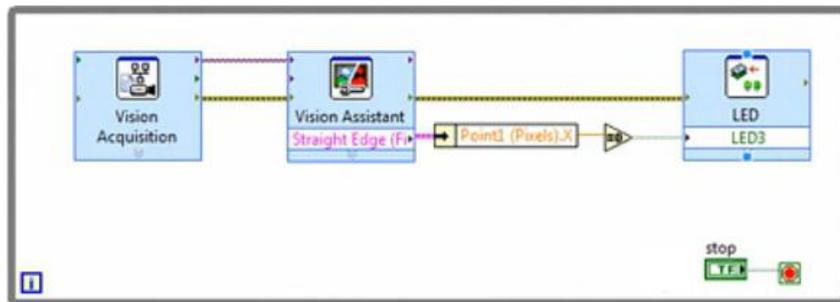
5、保存后点击运行，可发现正常情况下是可以检测到瓶子边缘的，LED3 并未点亮。若把瓶子拿开，则检测不到边缘，LED3 被点亮。实验完成。

事实上由于程序最终是运行在嵌入式的处理器上的，因此没有必要将图像显示控件和计算其每秒处理帧数的显示控件留在界面上，可以删除。当我们将程序下载到 myRIO 上，即使断开和计算机的连接，LED3 也会根据瓶子的位置变化点亮或熄灭，即说明图像的采集和处理分析都是在嵌入式系统的实时处理器上完成的。

在本实验中主要借助了 NI Vision Assistant 的功能，如果用户对 LabVIEW 的程序足够熟悉的话也可以使用 Vision 的函数自行编写程序。

此外上述程序有一个简化的实现方式，在函数选板 » Vision and Motion » Vision Express 中有两个快速 VI: Vision Acquisition 和 Vision Assistant. 图像的采集和处理过程都可以基于上述两个快速 VI 来实现，整个程序框图如下图所示。这个程序虽然实现简单，但其执行效率并没有上一个 Main.vi 高，主要因为在 Main 程序中，声明资源和释放资源都是在 while 循环外执行的，而此程序都则是在 while 循环内执行，因此降低了执行效率，如果对程序执行效率没有很高的要求，则这种快速实现的方式也是可取的。

图 4-19. 简化的图像采集及处理程序

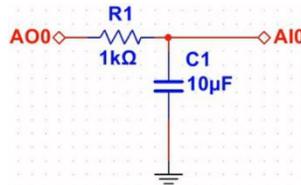


用户可以利用机器视觉来实现很多有趣的应用，有些功能用传感器实现比较方便，但有些功能使用机器视觉来实现会更合适，所以如果将摄像头和传感器结合起来，在 myRIO 上将大有用武之地，你所需要的就仅是创意而已！

5 基于 NI myRIO 的控制应用

myRIO 非常适合进行控制应用，尤其是设备所提供的 PWM 输出和编码器输入通道，使其更能适合一些机电一体化系统的控制，本章我们将探索 myRIO 的控制应用。为简单起见，我们以一个 RC 电路输出电压的控制为例来进行说明。

图 5-1. RC 电路原理图



如果把此 RC 电路当做一个被控对象，AI0 即为反馈信号，可以用 myRIO 的 AI 通道来采集这个电压。AO0 则是控制电压，可以用 myRIO 的 AO 通道来控制。RC 电路是一个典型的一阶线性系统，很多真实系统实际上都可以抽象成一阶线性系统，因此虽然采用的被控对象是一个简单的 RC 电路，但是设计方法都是相通的。

运行仿真程序

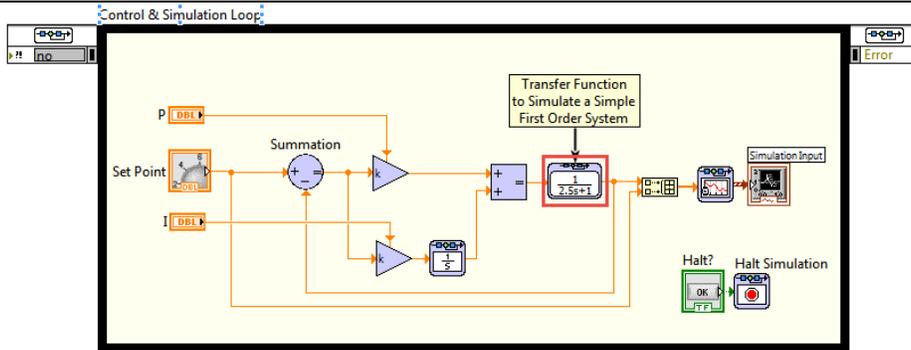
由于该阻容网络电阻和电容的值已知，所以可以用一个一阶模型对其进行建模和仿真。仿真程序 RC Simulate.vi 已经提供，该程序即是对 RC 电路控制的仿真，采用控制 PI 的方法进行，P 和 I 的参数可以从前面板输入，同时也可以在前面板上输入期望的 RC 网络输出值，即期望的被控对象的输出。运行之前需要用户安装光盘中的 Control Design and Simulation 控制设计与仿真模块并复制至 myRIO 中，安装方法与之前的类似不再赘述。



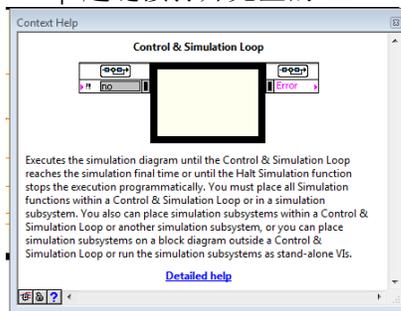
注意 仿真程序在项目中位于 My Computer 目标下，所以它其实并不是运行在 myRIO 上的，而是一个运行在计算机上的仿真程序。

1、双击打开仿真程序。在程序框图中可以看到，此程序是基于 LabVIEW 的控制设计与仿真模块来编写的，所以作为主体的循环是 LabVIEW 中的控制与仿真循环，其中的 PI 控制算法则都是用基本模块搭建的。此外 LabVIEW 中也有 PID 工具包，可以直接用其中的函数来实现这个过程。

图 5-2. 仿真程序的程序框图

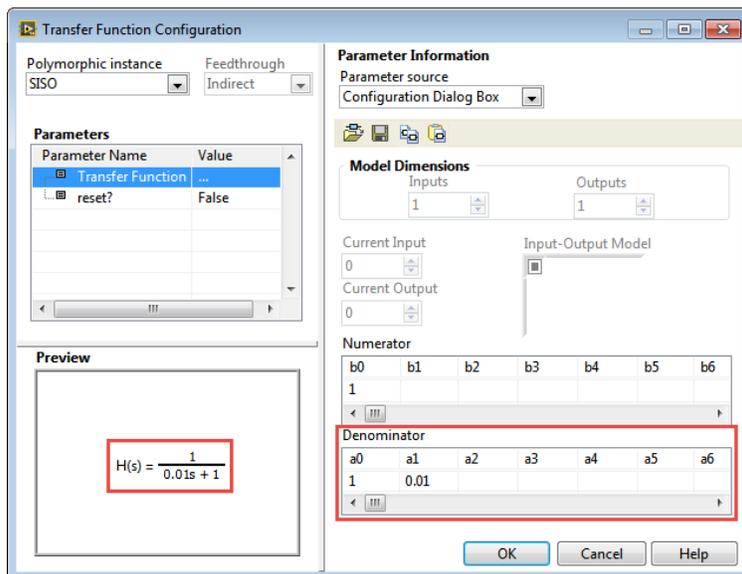


TIP 如果想要了解更多，可以按下<Ctrl-H>打开即使帮助，将鼠标移至感兴趣的函数上方。单击 Detailed Help 超链接打开完整的 LabVIEW 帮助文档。



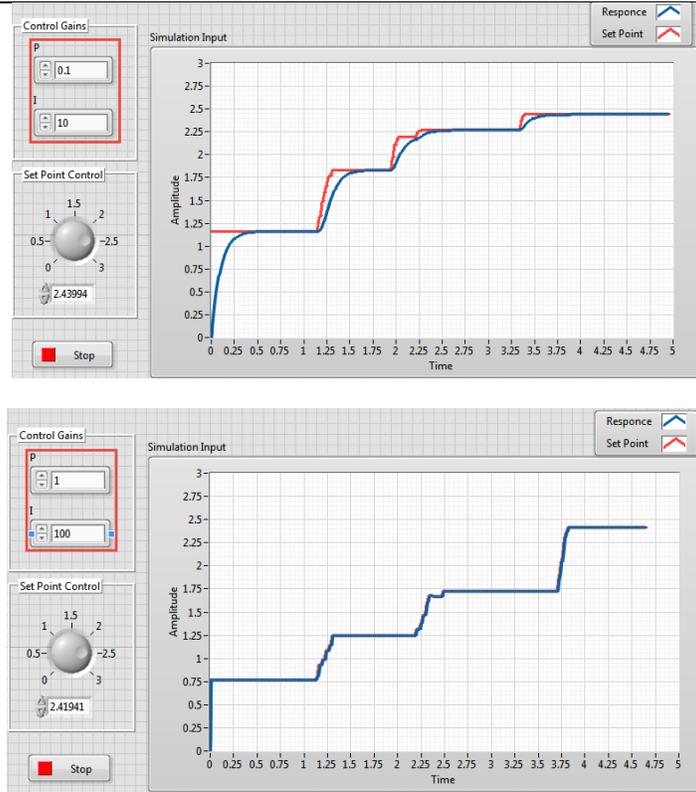
2、在上图所示的仿真系统中，红框中的即是用一个一阶传递函数来表示的仿真模型。双击打开它，在左下角的 Preview 区域中可查看到这个传递函数，我们可以通过参数的设置来修改它。而根据 RC 的实际值则可计算出当前一次项系数为 0.01。

图 5-3. 传递函数的配置框图



3、运行仿真程序。由于当前已经选择了一组比较合适的参数，所以用户可以看到呈现惯性过程的仿真结果。如果希望跟随情况更紧密的话，可以尝试将 P 的值改为 1，I 的值改为 100，结果几乎呈现为一个完全的跟随情况。

图 5-4. 仿真程序在不同 P、I 参数下的仿真结果



实验操作

做完此仿真后，就可以把控制算法部署到 myRIO 的实时控制器上，并用一个真实的 RC 网络替换用于仿真的传递函数模型，同时把软件仿真的接口改成硬件 I/O 的输入和输出。

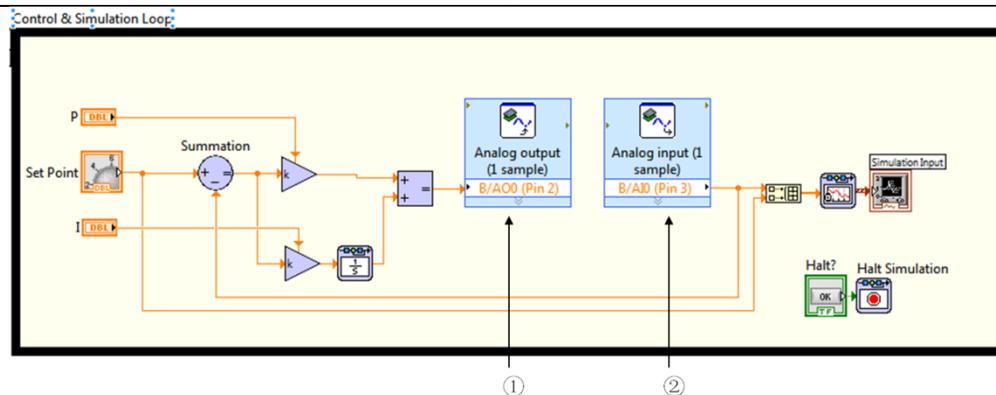
1、根据电路图将真实的阻容电路接到 myRIO 上。通过 myRIO 上 B 接口的 AO0 端口给 RC 网络提供输入信号， AI0 端口测量电容上的电压作为反馈信号。图 5.5 是采用了一个焊接的电路板通过 MXP 接口与 myRIO 链接，当然你也可以在面包板上搭建相应的电路，然后通过杜邦线与 myRIO 的相应端口连接。

图 5-5. 将 RC 电路连接到 myRIO 上 (此处扩展板上的七段数码管和电位器并不使用)



2、在 myRIO 工程管理窗口中，将上位机中用于仿真的控制算法移植到 myRIO 上执行。用 LabVIEW 开发嵌入式应用非常方便的一点是，可直接将 vi 从 My Computer 下方拖到 myRIO 下方即可。拖动之后，可另存文件并将其重命名为 RC Real Control.vi，按照下图修改程序框图。

图 5-6. RC 电路控制程序



- 1 模拟输出—删除传递函数模型，从函数选板 » myRIO » Default FPGA Personality 中拖选出 **Analog output** 快速 VI，在配置窗口的 **Channel** 一项中选择 B/AO0(Pin 2)。
- 2 模拟输入—从上述函数选板中拖选出 **Analog input** 快速 VI，在配置窗口的 Channel 一项中选择 B/AI0(Pin 3)，按图连接好数据流。

3、点击运行，程序下载至实时处理器上执行，改变期望响应值或 P、I 参数，观察响应。

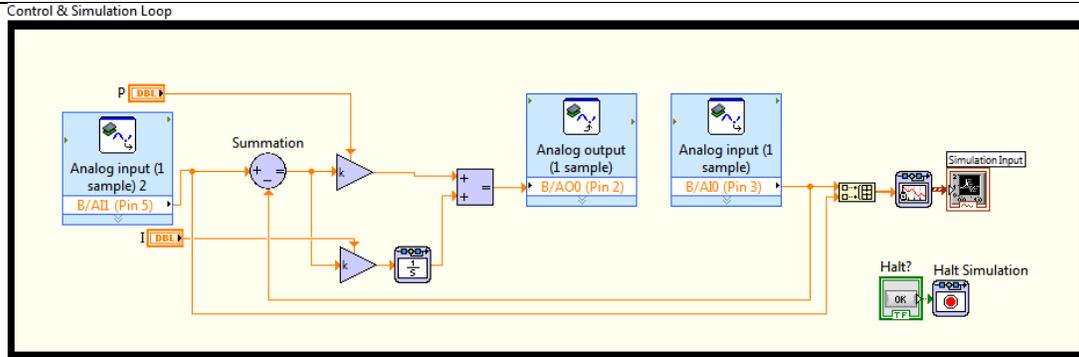
虽然运行结果的显示界面与仿真时完全类似，但此时的控制程序是运行在实时处理器上的，上位机则仅用于将 P、I 参数和期望的响应参数值等信息通过后台的网络传输机制传递给

控制器，并且显示传递回来的响应参数。可以发现在修改 P、I 参数值的过程中，响应结果可能会产生一定的过冲，这和仿真的情况有所不同。

改进实验

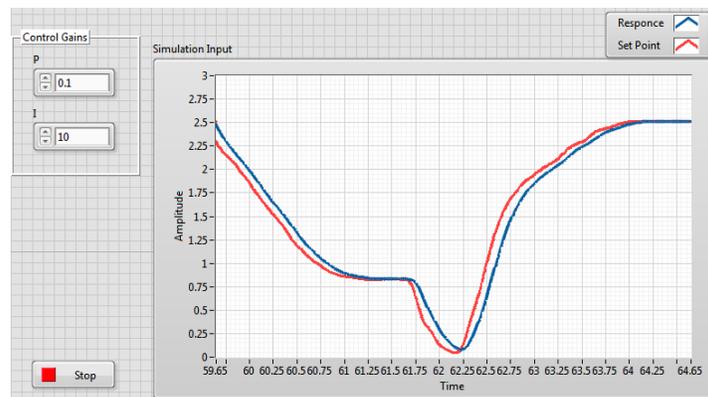
目前期望的阻容网络响应输出值是由上位机程序传递给 myRIO 的，我们还可以直接通过给 myRIO 连接一个电位器赋予期望输出值。我们假设电位器的输出连接在 AI1 通道上，因此只需要将 Set Point 旋钮控件替换成 AI1 通道的模拟输入即可。

图 5-7. 修改后的 RC 电路控制程序



运行程序，当我们图 5.5 中的旋转电位器旋钮，可在显示控件上看到相应的变化和响应。

图 5-8. 使用电位器控制 RC 电路电压输出的响应结果



本实验中，因为我们对被控对象有完全的了解，所以可以对其进行很好的仿真。但在一些实际项目的调试过程中，被控对象的模型很有可能是未知的，因此绝大多数时候控制参数便是靠反复的调试。若是此类情况，则不一定要按照上述经典的步骤（即先建模仿真，再部署控制算法至实时处理器上执行）。可直接用 LabVIEW 中普通的循环来实现控制，如果需要一定的实时性，也可以使用 LabVIEW 中的定时循环，路径为函数选板» Programming » Structures » Timed Structures » Timed Loop，具体信息可以参考帮助文档。

此外，用户在应用过程中可能会经常涉及电机和编码器，相关内容可以参考 NI myRIO Project Essentials Guide 这本书，该书此前已经介绍过，可以在 NI 官网下载到，内容丰富涵盖广。

书中每一个小节的内容都有配套的视频和范例程序，程序同样可以在 **NI 官网**上下载到，而视频资源可在优酷上搜索 myRIO 并且选择专辑即可看到全部视频，视频包括传感器的连接，特殊元器件的连接，对 LabVIEW 范例程序的说明等内容，用户可以仔细搜索利用相关资源完成所需开发。

6 通过智能终端进行远程监控

在前面的章节中介绍了通过 WiFi，myRIO 不仅可以和电脑，还可以和各种智能终端进行通信，从而完成远程监测和控制的功能。本章将探讨如何简单方便地实现该功能。

6.1 通过智能终端监测 myRIO

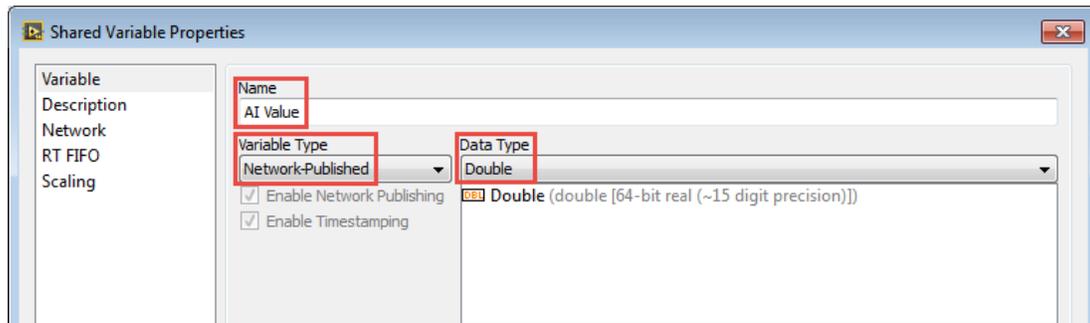
以第二章中电位器控制七段数码管的实验为例，希望可以在远程终端上读取电位器给 AI1 通道的值以及数码管的显示值。首先需要建立 myRIO 和智能终端之间的通信，这可以通过 LabVIEW 的网络共享变量，或是通过 LabVIEW 的 Web Service 等方式来实现。最简单易行的方法即是通过网络共享变量来实现，故以此为例进行具体说明。这里我们以 iPad 为例，介绍如何通过 iPad 远程访问 myRIO 数据。

修改程序

我们需要增加两个网络共享变量来传输数据，使得能在 iPad 上看到 AI1 通道的采集数据，以及经过转换后七段数码管所显示的数值。

- 1、打开第二章中创建的工程项目，右击 myRIO 目标弹出菜单，选择 **New » Variable**，弹出共享变量属性对话框。不妨先建立共享 AI1 通道值的变量，因此 Name 可命名为 **AI Value**，Variable Type 为 **Network Published**，Data Type 为 **Double**，即双精度类型，点击 OK 完成创建。

图 6-1. 共享变量属性对话框



- 2、在工程管理窗口中，选择保存，将新创建的变量保存至变量库中，可以将变量库命名为 7 seg LED。
- 3、打开 Main 程序，可以把新建的共享变量直接拖拽到程序框图中，可发现此时它是一个读取量。由于需要将输入通道的数值赋予此变量，所以右击变量选择 **Access Mode » Write**，连接好数据线后保存。

图 6-2. 转换共享变量的获取模式

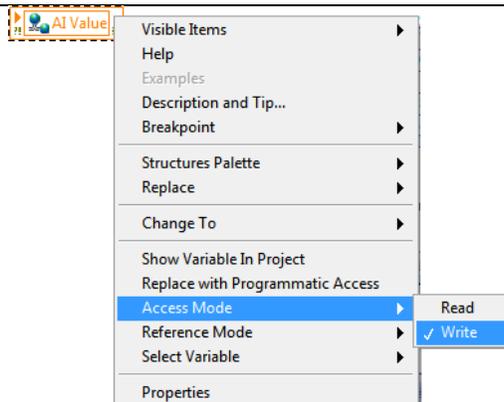
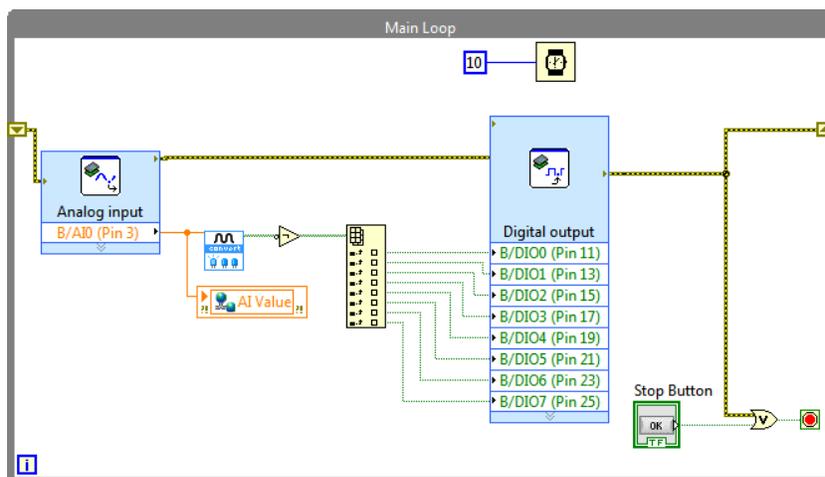


图 6-3. 修改后的程序框图



4、点击运行，可发现共享变量也被部署到 myRIO 上了。

配置终端

完成了部署至 myRIO 上的工作，还需要针对智能终端 iPad 进行相关设置。

1、首先需要为 iPad 安装一个名为 **Data Dashboard for LabVIEW** 的 app，可在 App Store 中搜索到，并查看关于此 app 的相关介绍。

2、下载安装后进入该软件，可看到已经几个已经创建的指导用户使用本软件的 Dashboard。

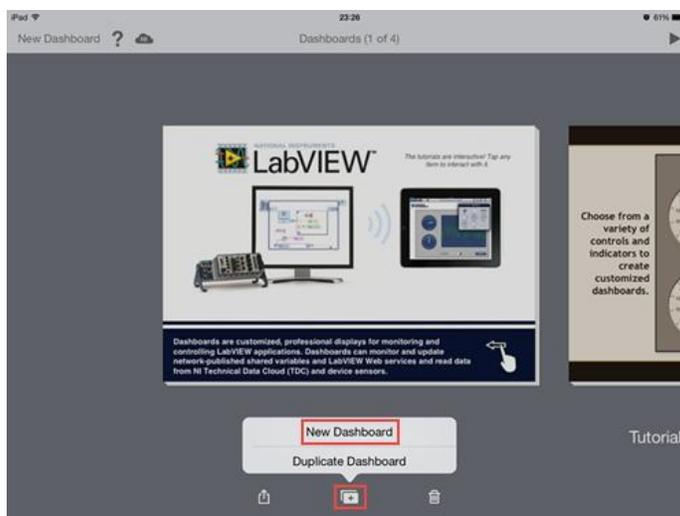
3、连接智能终端和 myRIO。前面介绍过 myRIO 和智能终端的连接有两种实现方式，一种是将智能终端，myRIO 和开发上位机的电脑都连接到同一个无线路由器上，另一种方法则是使 myRIO 作为一个无线 AP，然后将上位机和其他的智能终端都连接到 myRIO 上去。现在我们采用第二种方式，把 iPad 接入 myRIO 的无线网络。

图 6-4. iPad 接入 myRIO 的无线网络



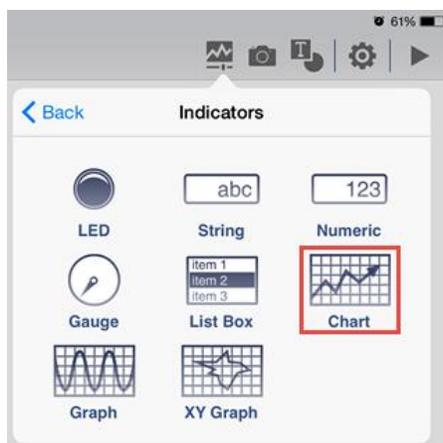
4、进入 Data Dashboard，新建一个 Dashboard，这就相当于一张空白画布，用户可以在其中放置控件和远程系统中的共享变量绑定起来。

图 6-5. 新建一个 Dashboard



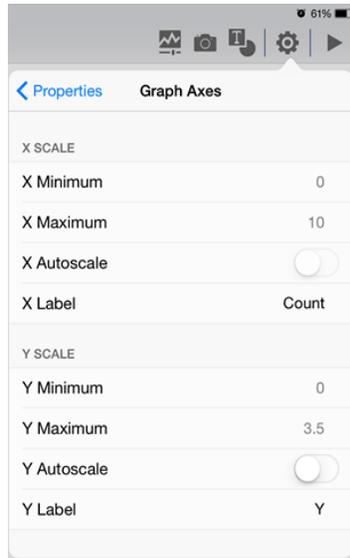
5、使用波形图表来和刚才建立的共享变量 AI Value 进行绑定，将 **Controls and Indicators » Indicators » Chart** 直接拖曳至界面上，调整合适大小。

图 6-6. 创建 Chart 控件



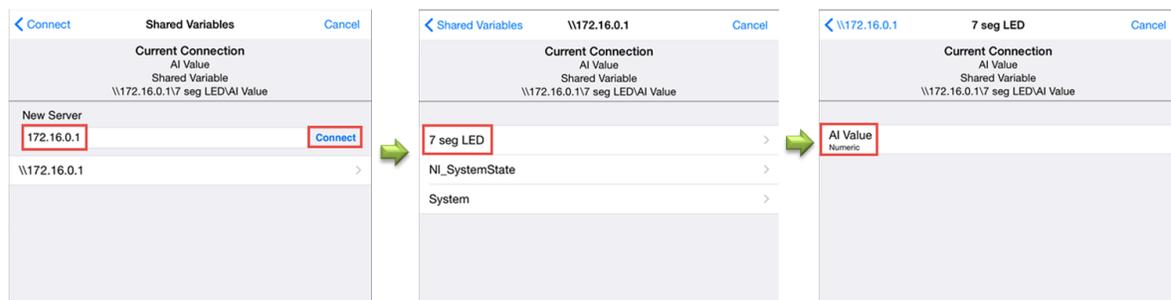
6、更改属性。在选中控件的情况下，点击进入 **Properties » Graph Axes**，可选择关闭 Y 轴的自动调整，将其最大值设置为 **3.5**，因为此端口所提供电压最大值为 3.3V。此外还可以进入 **Properties » Font**，改变字体大小等属性。

图 6-7. 更改控件属性



7、最重要的是需要将控件和共享变量绑定。点击控件下方的 **Connect » Shared Variables**，在 New Server 中输入 myRIO 的 IP 地址，由于此时是将 myRIO 作为无线 AP 使用，所以 IP 地址为 **172.16.0.1**。点击 **Connect** 连接后，看到可供绑定的共享变量库，其中便有刚才创建的 **7 seg LED**，点击选择后，便能看到其中的共享变量，选中 **AI Value** 完成绑定。

图 6-8. 绑定控件和共享变量



8、点击右上角的运行程序（此时 myRIO 上的程序也必须要开始运行），当调节电位器输入（即改变 AI 端口电压）时，可在智能终端（iPad）上读取到相应值的变化。

图 6-9. 智能终端上的监测结果



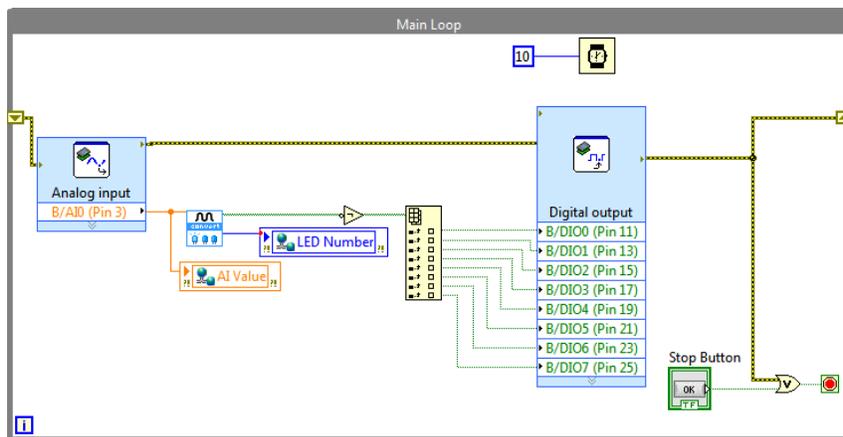
添加共享变量

用户在上述 iPad 远程端实际上看到的是模拟输入通道采集到的电压值，如果还需要显示 LED 的数值，则需要再添加一个共享变量。

1、在库中添加共享变量。右击 7 seg LED 库» New » Variable，将此共享变量命名为 **LED Number**，Variable Type 为 **Network Published**，Data Type 为 **Int8**，即 8 位整形数据。点击 OK 完成创建，保存项目。

2、打开 Main 程序框图，将新创建的共享变量拖入框图中。同样地将获取模式改为写入，连接好数据线后保存程序。

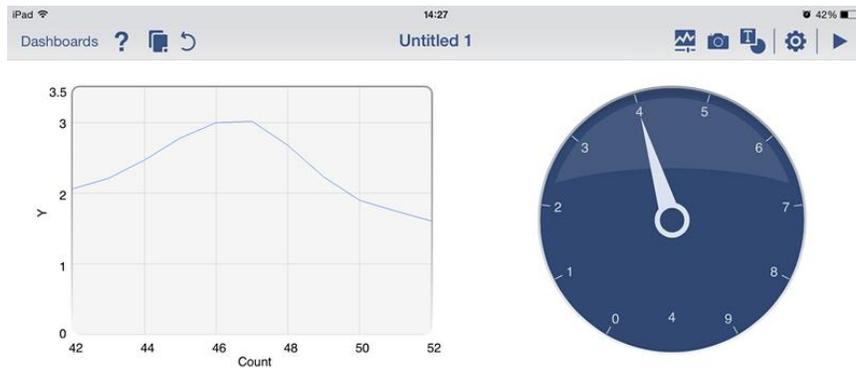
图 6-10. 修改后的程序框图



3、在 iPad 新建 Dashboard 的界面上，添加一个控件和 LED 输出值的共享变量绑定。可以选择 **Controls and Indicators » Indicators » Gauge**，将其拖拽至界面上，调整尺寸和位置。

- 4、在属性设置中根据个人偏好设置显示风格，字体字号等，将显示范围修改为 0~9。
- 5、同样最重要地是将控件与 **7 seg LED** 库中的共享变量 LED Number 进行绑定，步骤与上述类似，注意 iPad 需要连接至 myRIO 的无线网络中且共享变量已被部署至 myRIO 中。
- 6、同时运行 myRIO 和 iPad 上的程序。可发现当再次旋转电位器时，不仅与 myRIO 相连的 7 段数码管的数值发生变化，智能终端上 Dashboard 的界面也会发生相应的变化：左边控件显示的 AI 通道的输入电压值，右边控件显示的是 7 段数码管的值。

图 6-11. 添加共享变量后智能终端上的检测结果



- 7、用户还可以美化 Dashboard 界面，例如增加文字和图形，插入照片，或是改变背景等。
- 8、保存当前 Dashboard。回到管理窗口，Dashboard 修改的界面软件都会自动保存，但是为了方便管理，我们可以将当前 Dashboard 命名为 AI Monitor。

以上实验介绍的是以 iPad 为例和 myRIO 通过 Dashboard 来进行网络共享变量的通信和传输，但实质上这种方法同样适用于 iPhone 和基于 Android 的平板电脑或智能手机，用户也可以在安卓市场或是豌豆荚中，找到 Data Dashboard for LabVIEW 这个应用。

6.2 通过智能终端控制 myRIO

上一小节介绍了通过 Data Dashboard 读取 myRIO 项目中的一些变量，并通过智能终端进行显示。本节将介绍同样通过 Data Dashboard 给 myRIO 项目中的一些参数进行赋值，从而实现控制的过程。

实验操作

仍然以 7 段数码管显示控制的实验为例，项目原本设计是通过手动地调节电位器改变 AI 端口输入电压值从而控制 7 段数码管的显示，现在改为通过智能终端来远程控制参数的变化。

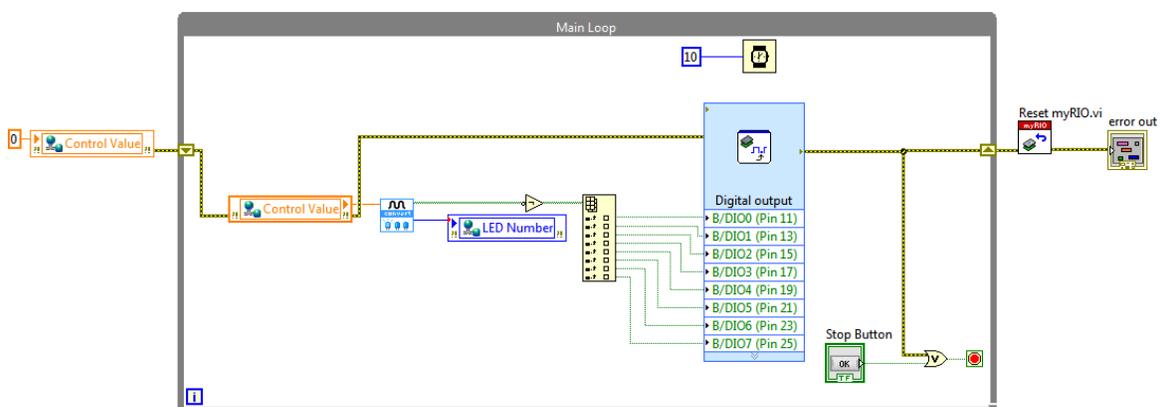
- 1、打开前面的 7-seg display 项目，按照同样的方法，再给项目添加一个数值型的网络共享变量，将其命名为 Control Value。



注解 通过 Dashboard 给网络共享变量赋值，而 myRIO 上运行的程序去读取这个值，便可以实现远程控制机制。

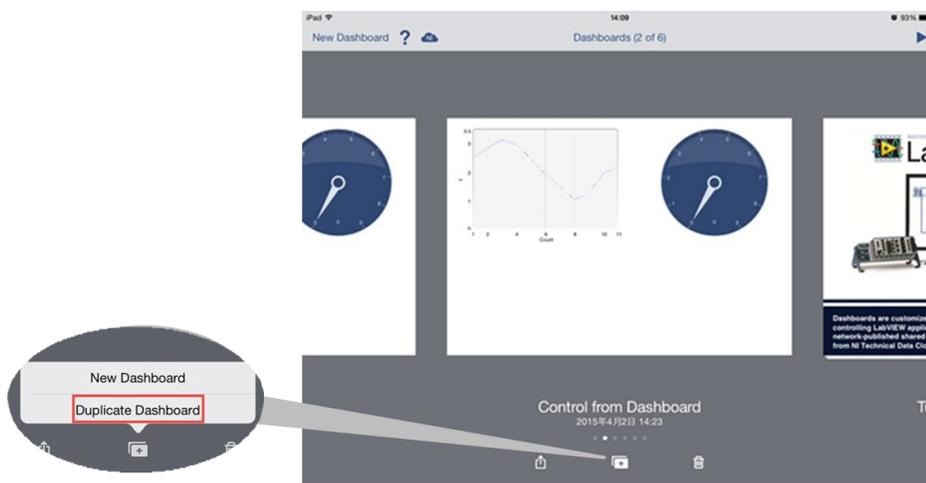
- 2、打开 Main 程序框图，修改程序。由于希望用来自 Data Dashboard 的控件输入来替换通过 AI 端口采集的输入，因此需要删除一部分程序。将 Control Value 拖入框图作为新的输入，智能终端上运行的 Dashboard 将给变量赋值，从而改变程序运行时参数的对应值。
- 3、程序刚启动时，共享变量应有一个默认值，例如 0。摁住 Ctrl 键拖动此变量，复制至循环外，将其获取模式改为写入，并赋常量 0。通过错误簇来控制程序的执行顺序，即先对变量进行赋值，再进入循环，使变量不断接受来自智能终端的赋值，并以此来控制 7 段数码管的显示输出。完成程序的修改。

图 6-12. 修改完成后的程序框图



- 4、保存修改并运行程序。新增的共享变量和修改的程序都会被部署至 myRIO。
- 5、打开 iPad 客户端的 Data Dashboard 软件。点击 **Duplicate Dashboard** 为上一节中保存的 AI Monitor 创建一个副本，重命名为 Control from Dashboard。

图 6-13. 创建 Dashboard 副本



6、进入新的 Dashboard，双击波形图表控件，选择 **Delete** 删除该控件。从 **Controls and Indicators » Controls** 中新增一个控制控件 **Slider**。

图 6-14. 创建 Slider 控件



7、修改控件属性。在 **Properties » Data Range** 中将控件最大值修改为 **3.3**，也可根据个人喜好修改其他属性，包括界面风格和字体字号等。

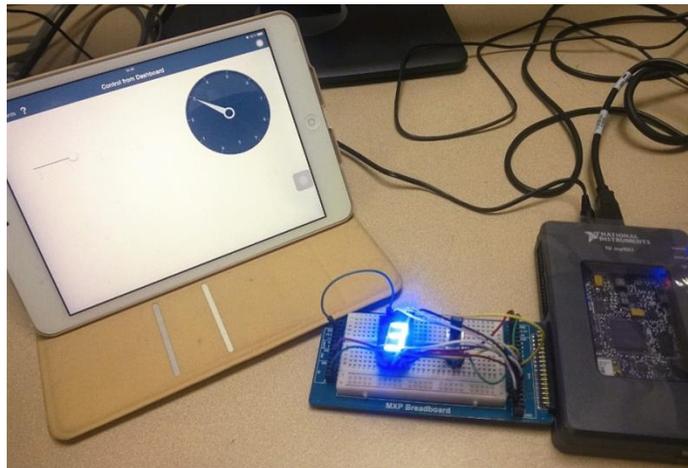
图 6-15. 更改控件属性



8、绑定共享变量。与上一小节中第一部分第 7 步中介绍的方法完全相同，进入 7 seg LED 库中，找到 Control Value 进行绑定。如果在库中没有看到此新建变量，可点击右下角的**刷新**按钮。

9、完成绑定后运行 Dashboard 和上位机程序。当用户通过 iPad 改变输入控件值时，通过 WiFi 所连接的 myRIO 就会被远程控制，7 段数码管的值会随之改变。

图 6-16. 运行结果实物图



10、用户还可以通过云或 E-mail 共享创建好的 Dashboard，以便在其他智能终端上使用。

本章节介绍了通过 Data Dashboard 和 LabVIEW 网络共享变量在智能终端和 myRIO 嵌入式项目有之间进行交互的方法，用户可以在实际项目中加以利用。

7. 生成上电自启动程序

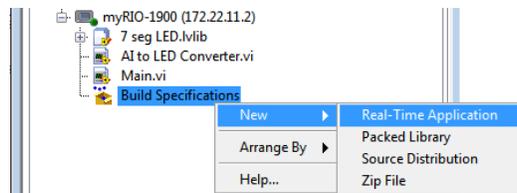
在开发过程中，用户通常会使用 USB 线缆来连接 myRIO 和电脑。当开发完成后用户便可以将整个项目作为一个独立的应用程序部署并存储到 myRIO 的硬盘上，当下一次启动 myRIO 时，不用连接 USB 线缆，应用也将自动运行，这即是上电自启动程序。本章就将介绍生成并部署上电自启动程序到实时操作系统上的具体过程。

生成并部署应用程序

本章仍然以第二章中七段数码管显示的程序为例进行介绍：

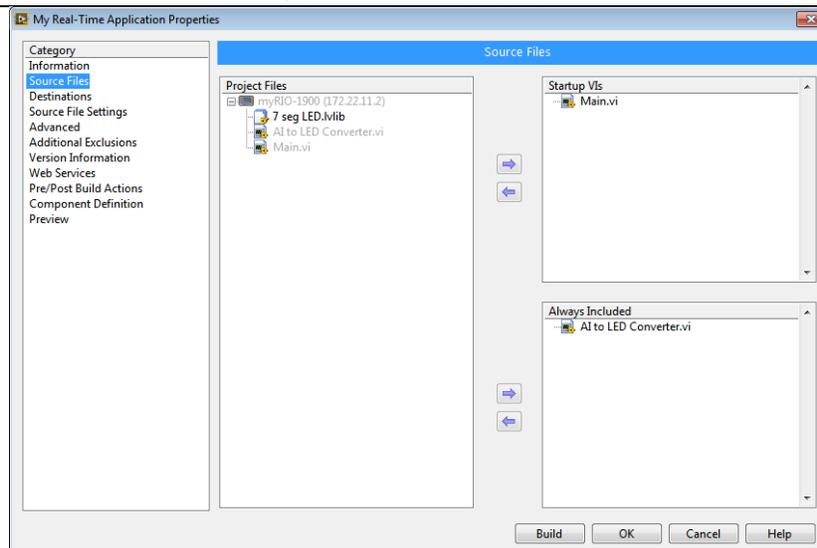
- 1、在工程浏览器窗口中打开 myRIO 目标下的 Main.vi，使用 USB 线缆连接 myRIO 与开发上位机，点击运行，确保程序能在实时操作系统上正常运行。
- 2、回到工程浏览器窗口，右击选择 myRIO 目标下的 **Build Specifications » New » Real-Time Application**。

图 7-1. 创建应用程序



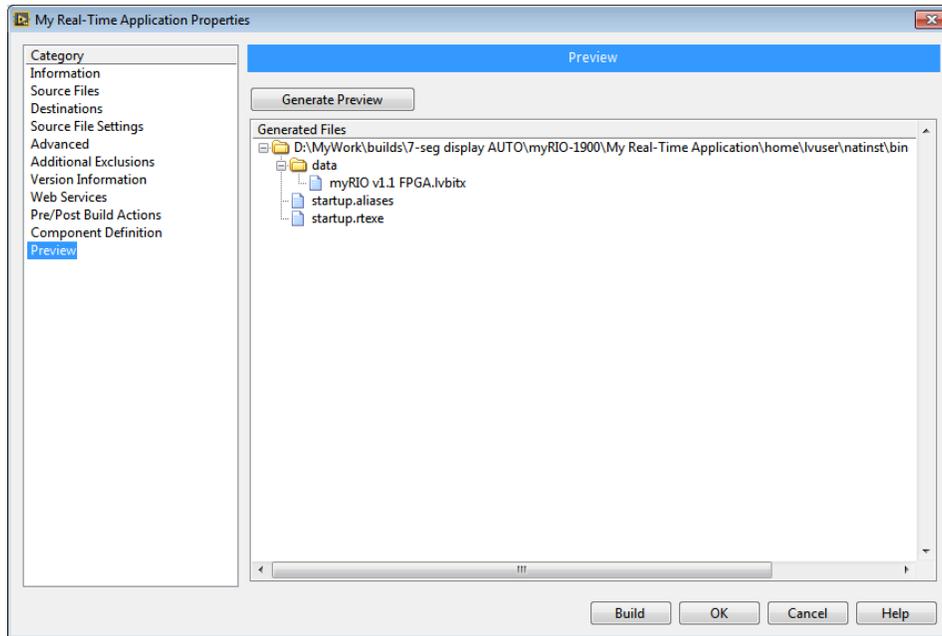
- 3、配置应用程序。在 Information 一项中，Build specification name 可以为“7-seg display”也可以选择为默认名称。下面三项配置都选择默认即可。在 Source Files 一项中，用户的应用程序可能会包含多于一个的 VI，但只会有一个顶层 VI，将其选择为启动 VI，子 VI 可选择为始终包括。目录中其余选项都默认即可，用户可自行查看相关配置信息。

图 7-2. 配置应用程序



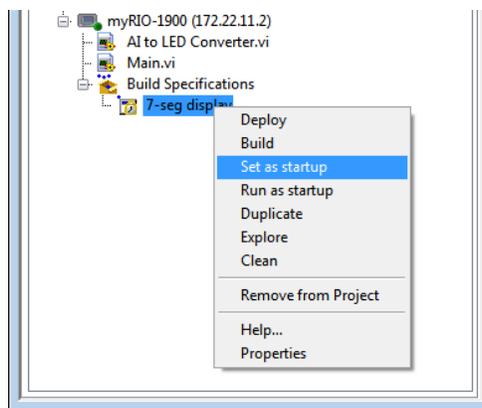
3、选中 **Preview** 一项，点击 **Generate Preview**。如果用户不希望将生成的错误信息写入实时操作系统，可在 **Advanced** 中，将 Copy error code files 勾选掉，再次点击 Generate Preview 即可发现只有必要的应用程序信息将被写入 myRIO。

图 7-3. 只含有必要信息的生成预览



4、点击 **Build** 生成应用程序，完成后即可在 myRIO 目标下看到生成的应用程序 7-seg display。右击该应用程序，选择 **Set as startup**。

图 7-4. 配置应用程序



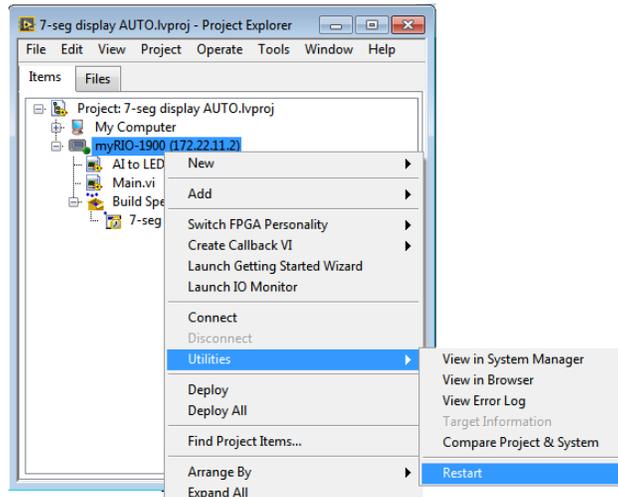
TIP 右击应用程序，选择 Explore，可在本地目录中查看目标文件 startup.rtexe 的具体位置。

5、右击 myRIO 目标下的生成的应用程序，选择 **Deploy** 将程序部署到实时操作系统上。用户可在浏览器中输入 **172.22.11.2/files** 查看部署到 myRIO 上的目标文件，具体路径为 <http://172.22.11.2/files/home/lvuser/natinst/bin/>。

启用/禁用上电启动程序

1、右击 myRIO 目标，选择 **Utilities » Restart**，重启 myRIO 后可看到部署在其上的上电自启动程序将开始自动运行。将来如果完全断电之后再上电，之前部署的程序也会自动运行。

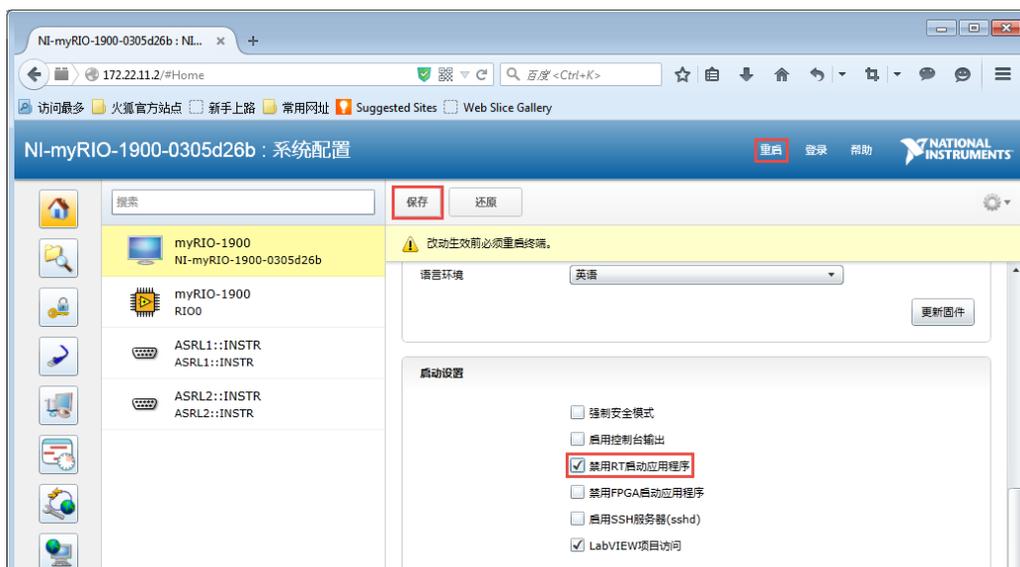
图 7-5. 重启 myRIO 设备



注意 上电自启动程序会在 myRIO 重启完，呈现红色的 STATUS 状态灯熄灭后 10~15 秒开始自动运行。

2、如果用户将来希望禁用上电自启动程序的自动运行，在 myRIO 使用 USB 线缆与计算机相连的情况下，在浏览器地址中输入 172.22.11.2。在启动设置中，勾选**禁用 RT 启动应用程序**，并保存设置。

图 7-6. 通过网页启用/禁用 RT 启动应用程序



3、根据提示重启设备，重启后程序将不再自动运行。

4、用户也可以在 NI MAX 中启用或禁用上电自启动程序。打开 NI MAX，在远程系统中选中 myRIO，在右侧系统设置选项卡中找到启动设置一栏，此时勾选掉禁用 RT 启动应用程序，重启设备又可恢复启用状态。

图 7-7. 通过 NI MAX 启用/禁用 RT 启动应用程序

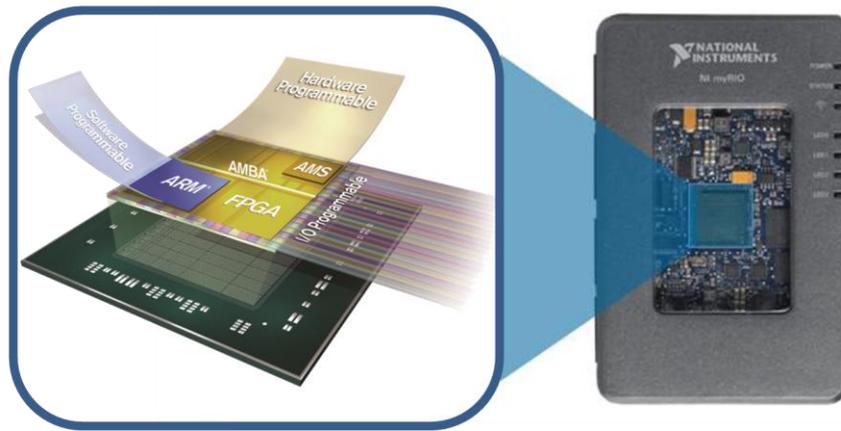


根据以上介绍，用户可按照实际需求在 myRIO 上部署上电自启动程序，并加以运用。

8. NI myRIO 板载 FPGA 资源开发介绍

NI myRIO 内嵌 Xilinx Zynq 芯片，支持 667 MHz 双核 ARM Cortex-A9 可编程处理器和可定制的现场可编程门阵列（FPGA）。设备出厂时已配置好 FPGA，初学者可以直接运行基础功能，无需为 FPGA 编程。在做进一步开发时也支持对 FPGA 自定义，并重新配置 I/O。本章将以 NI myRIO 完成音频信号的提取分析与输出实验为例，介绍对板载 FPGA 资源的开发。

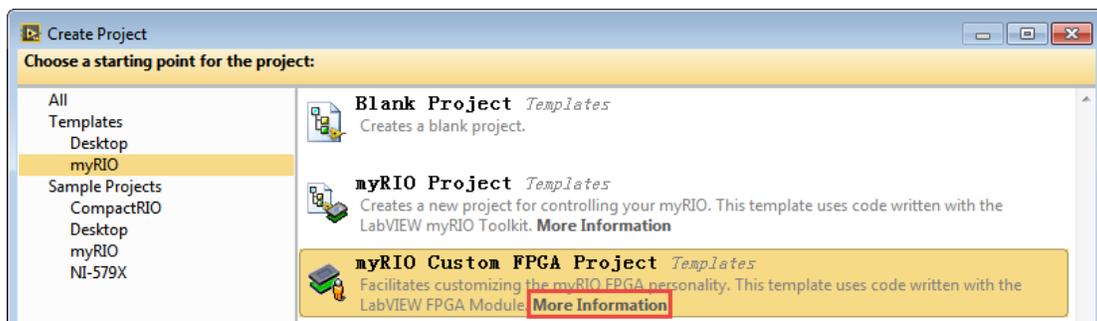
图 8-1. Xilinx Zynq 芯片架构



查看 NI myRIO FPGA 默认的配置信息

- 1、保存所有开启的应用，返回 LabVIEW Getting Started 窗口，选择 **Create Project**。
- 2、在 Create Project 窗口左侧的列表中，选择 Templates 下面的 myRIO。然后在窗口右侧列表中，选择 **myRIO Custom FPGA Project**。说明中的 More Information 链接中讲解了如何设置项目。

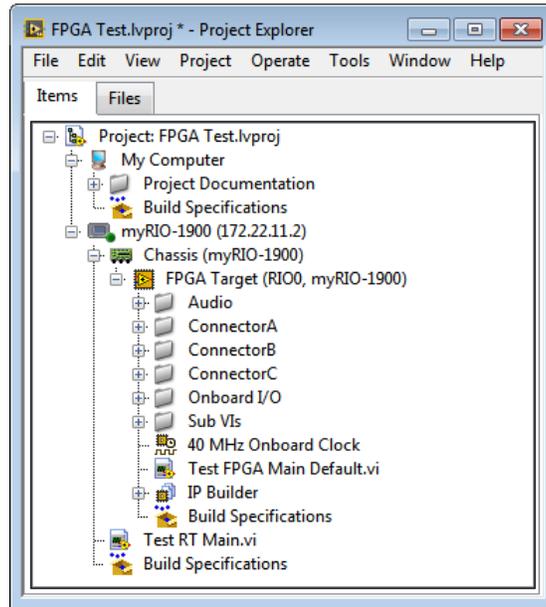
图 8-2. 创建 FPGA 模板项目



3、选择 **Next** 配置项目后点击 **Finish** 完成创建。

4、展开 myRIO 目标后，可查看新出现的“Chassis”树型结构。因为 FPGA 位于 NI myRIO 板上，因此项目中的 FPGA 对象位于 Chassis 下面。

图 8-3. FPGA 模板项目



注解 FPGA 的重要作用 是处理 NI myRIO 上所有 I/O，可以在 Project 中找到这些 I/O。FPGA 下面的 I/O 被组织为文件夹的形式，告诉了用户每一个 I/O 节点在真实物理设备上的对应位置。两个 MXP 接口、一个 MSP 接口、板载 I/O 均有独立文件夹。文件夹再下一级又按照 I/O 类（数字/模拟）和 I/O 的物理分段来划分为子文件夹。可将这些 I/O 项拖入 FPGA VI 中，从而对这些 I/O 进行读/写操作。可以从实时处理上运行的 RT VI 对 FPGA VI 前面板上的输入控件和显示控件进行读写操作。

5、用户可打开 Test FPGA Main Default.vi，查看默认提供的 FPGA 代码。



注解 按照 LabVIEW 项目的配置，FPGA 采用 40MHz 的时钟。创建在 Project 中 FPGA 下的 VI 都会被认为是 FPGA VI，即这些代码的最终执行对象是 FPGA（代码最终将被编译成 bit 流文件对 FPGA 进行配置），因此 LabVIEW 会根据这一特点自动限制 VI 中的函数和数据类型。用户可以选择创建新的 FPGA VI 或修改默认提供的 FPGA VI。

模板中所默认提供的 FPGA VI 主要处理 FPGA I/O 数据，并准备好将这些数据传送到 Real-Time VI（在 ARM 处理器上运行）。默认的 FPGA VI 可以处理 MXP 接口和 MSP 接口的所有输入和输出，包括 PWM，I2C，SPI 和 正交编码器 I/O。

一般情况下，使用默认的 FPGA 配置就可以满足项目需求。为了简化项目开发，只需要开发运行于 ARM 处理器上的 Real-Time Host VI，以及在有需要时开发运行在上位机上的

Windows VI，此时可以通过普通的 myRIO Project 来创建项目。在针对一些特殊需要时，也可以用上述所提到的方法来创建自定义 FPGA 项目，这样就可以利用自定义 FPGA VI 所带来的灵活性、时间确定性、协处理等优势，可参考本章的示例实验。

示例实验

1、使用 USB 线缆连接 myRIO 与开发上位机，使用 myRIO 开发套盒中的音频信号线连接音源（例如 MP3 或智能手机等）和 myRIO 机盒侧面的 AUDIO IN 接口，提取音频信号至板载 AD 电路以待分析。再将音频播放设备（例如音箱或耳机等）的信号线连接至 AUDIO OUT 接口，以播放处理好的左右声道音频信号。

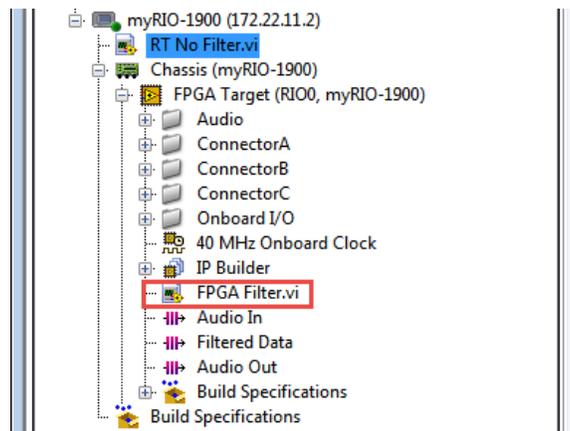
2、本章主要向用户介绍该 FPGA 资源，引导用户体验开发，因此对具体编程开发过程不做强调。在本实验的开发过程中，用户可参考提供的范例程序 FPGA Audio Filter，这是一个使用 myRIO 内嵌的 Xilinx Zynq 芯片上的 PFPGA 模块完成即时声音信号处理的程序。



注解 对于某些需要高确定性控制或大量信号处理的应用而言，更希望将一些算法放在 FPGA 上执行，此时就需要对 FPGA 进行编程重配置，用户可以安装光盘中的 LabVIEW FPGA 模块帮助完成这项工作。此外，在 [NI myRIO 网络社区](#) 中，已有许多编写好的 FPGA 配置程序，可直接使用。

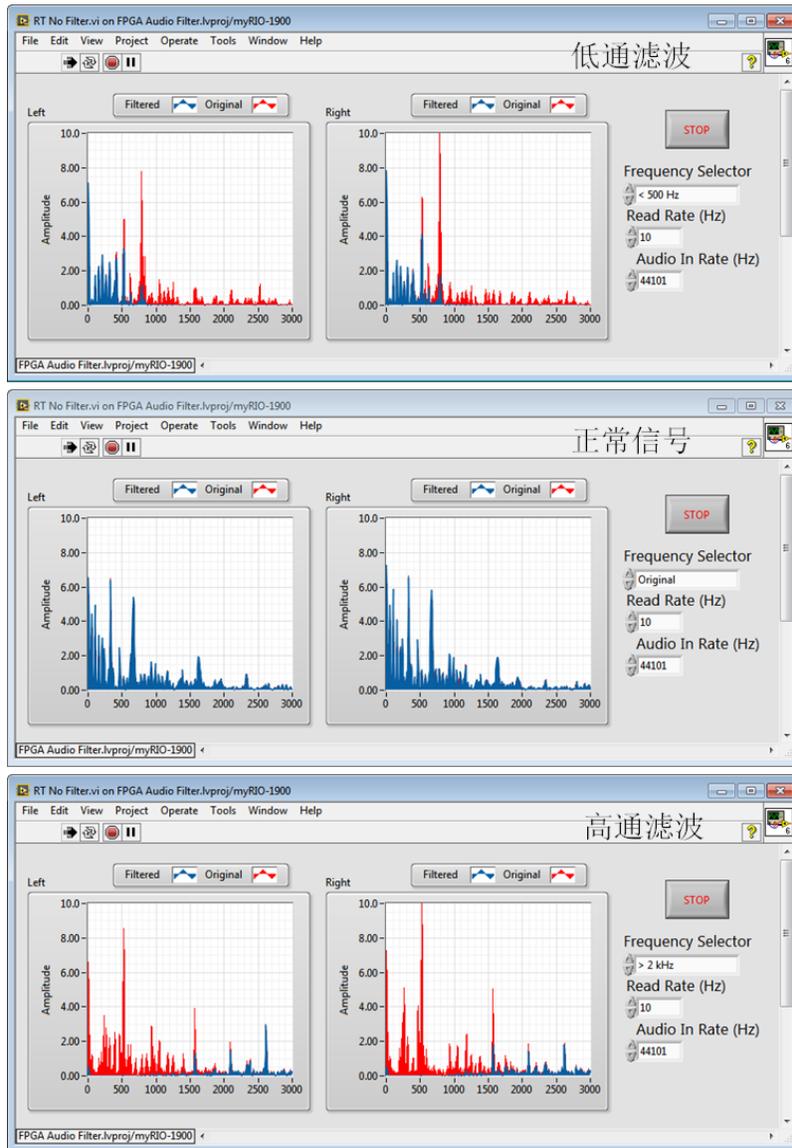
3、打开 FPGA Audio Filter 工程文件，在 FPGA Target 下可找到 FPGA Filter.vi，此程序定义了了在 FPGA 上完成的滤波处理过程，用户可自行查看理解。

图 8-4. 查看实现 FPGA 开发的程序



4、打开 myRIO 目标下运行在实时系统上的程序 RT No Filter.vi，在程序框图中可发现已打开了 FPGA Filter 的 VI 引用，点击 I/O 资源名的下拉菜单，通过 Browse 找到 myRIO 的 FPGA Target Name。

图 8-6. 运行在实时系统上的程序



注意 第一次下载编译此工程时，需先在上位机上编译 FPGA Filter.vi，此程序已被设置为 Run Mode。事实上，如果用户有自定义开发或修改 FPGA Target 下的 VI 时，都需要先编译通过才能在实时操作系统上运行利用 FPGA 资源的程序。

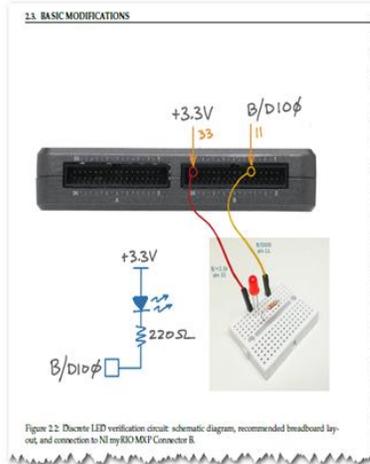
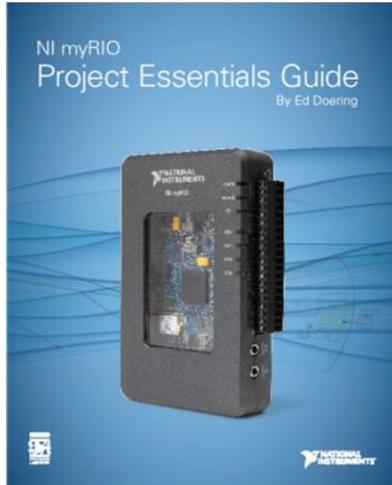
可以发现上述基于 FPGA 进行音频信号处理的程序，不同的滤波参数便能得到不同的输出结果，可以应用于抗噪耳机的开发等实际工程中。

根据以上全部关于 myRIO 的基本功能介绍，用户可以充分发挥想象力，进一步深入挖掘 myRIO 的强大功能，利用其方便快捷地开发更多应用于实际场合的有趣系统。更多的学习参考资源可详见附件。

附录

A 英文参考资料

Rose-Hulman 理工学院的 Ed Doering 教授编写的《NI myRIO Project Essentials Guide》是一本 NI myRIO 的入门实验指导书，通过 16 个实验介绍了与 16 种不同的外围元器件相连接的方法，如电位计、编码器、电机等。该实验指导书可通过 NI 网站[免费下载](#)。



2 Discrete LED

LEDs, or light-emitting diodes, provide simple yet essential visual indicators for system status and error conditions. Figure 21 shows the four types of LEDs included in the SparkFun "LED Mixed Bag (5mm)" kit (<http://www.sparkfun.com/products/9881>).

Learning Objective: In this module you will create a standard interface circuit to verify correct operation of the LED, learn interface circuit design principles and related LabVIEW programming techniques, make some basic modifications to extend your understanding of the interface, and then challenge yourself to design a system that integrates the discrete LED with additional components or devices.

2.1 Component Verification

Follow these steps to verify correct operation of the discrete LED component.

Select these parts:

- Resistor, 220 ohm
- "Basic Red" LED from SparkFun 9881
- Breadboard
- Connecting wires [need details]

Figure 21: Discrete LEDs, from left to right: standard red and green, high-efficiency in various colors, and RGB.

Download the LabVIEW project: Download the project Discrete_LED_demo.lvproj from [need details].

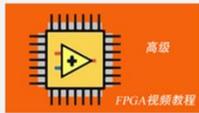
B. 教学视频链接

- NI myRIO 中文教学视频: study.gsdzone.net

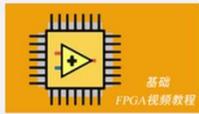
最新课程 / Latest Course study.gsdzone.net



NI myRIO入门教程
NI myRIO是NI针对高校教学应用最新推出的嵌入式创新实验与项目开发平台。NI myRIO基于Xilinx Zynq技术,集成了双核ARM和FPGA,使学生可以通过LabVIEW图形化编...
老师 ttt 免费



LabVIEW FPGA高级教程
现在隆重推出视频讲解教程, FPGA高级教程第一课 深度剖析FPGA之FPGA123 高级教程是一个专辑,计划包括如下内容: 优酷专辑地址 <http://...>
老师 wuxingsoho 免费



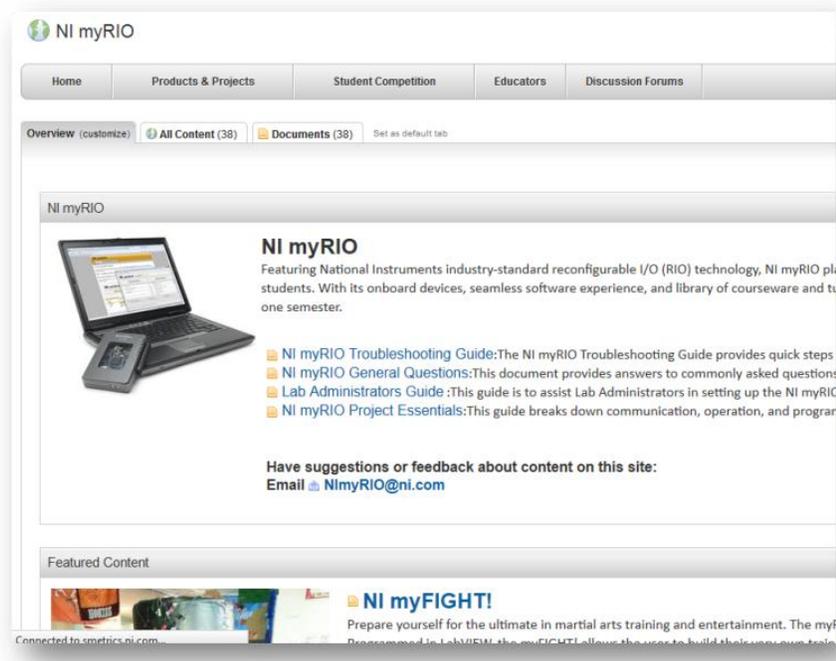
LabVIEW FPGA从入门到精通视频教程
LabVIEW FPGA从入门到精通视频教程 有10课内容,目前已经更新到了第五课
老师 wuxingsoho 免费

- 优酷上的 NI myRIO 视频专辑: soku.com/search_playlist/q_MYRIO



C. 项目范例及讨论

NI myRIO 项目范例资源：ni.com/community/myrio



D. NI myRIO 的 C 语言编程的支持

NI myRIO 基于 NI 可重配置 I/O (RIO) 技术，使您能够同时编程运行实时操作系统的处理器和自定义 FPGA。除了 NI LabVIEW 软件，NI myRIO 处理器也可通过 C 或 C ++ 进行编程（仅限对 ARM 处理器）。更多关于 C 支持的信息请访问

<http://www.ni.com/tutorial/14619/zhs/>。