

# C/C++嵌入式系统设计工具

发布日期：2014年7月10日

## 概览

设计工程师通常使用C等编程语言来开发基于微处理器或微控制器架构的嵌入式系统。C语言的普及催生了用于程序库、编译器和操作系统等工具的大型技术生态系统。现场可编程门阵列（FPGA）是嵌入式领域的新技术，采用硬件描述语言来创建可重配置数字逻辑。为了描述硬件和利用数字电路的高性能并行性，硬件描述语言提供的语法与软件工程师习惯使用的语法截然不同。通过提供包含软件工程师熟悉的编程构造的FPGA开发环境，在设计中使用FPGA嵌入式软件工程师就可以充分发挥FPGA的优势。

使用NI硬件的嵌入式系统设计工程师在构建其系统时可以选择一系列编程语言和工具。阅读本文档，了解如何编程NI嵌入式系统以及下载您需要的软件工具。

## 目录

1. NI嵌入式系统架构介绍
2. 选择NI嵌入式硬件平台
3. NI LabVIEW用于FPGA开发
4. C/C++实时应用程序开发
5. 使用FPGA接口C API
6. NI CompactRIO C/C++开发工具
7. 基于NI PXI或PC的C/C++开发工具

## 1. NI嵌入式系统架构介绍

NI可重配置I/O (RIO)平台包括两个可编程硬件：运行实时操作系统(RTOS)的微处理器和FPGA。这使您可以根据每个任务的重要性，在FPGA和RTOS之间为代码分区。

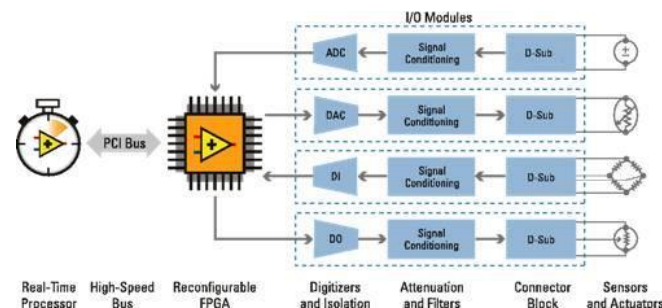


图1. NI嵌入式硬件架构

## NI可重配置硬件

在嵌入式系统设计中引入FPGA，使得无需自定义ASIC设计就可以创建高性能数字逻辑。FPGA的开发通常是采用基于文本的硬件描述语言来实现，如VHDL或Verilog，这些是由数字设计工程师编写的。但是由于这些硬件描述语言的语法过于复杂，要利用FPGA设计的全部功能非常困难。这一点充分体现在嵌入式行业亟需寻找更高层次抽象的FPGA设计，如C-to-gates工具或LabVIEW FPGA的G语言等图形化编程语言。

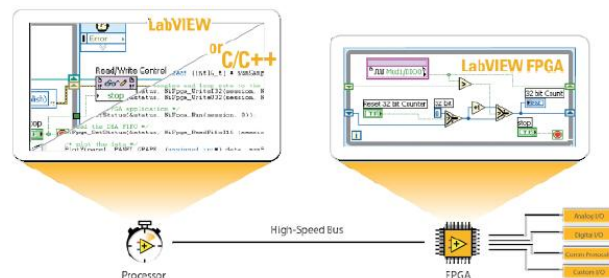


图2. NI嵌入式系统软件架构

## NI嵌入式软件架构

**实时操作系统** - 您可以灵活地选择使用LabVIEW Real-Time、C / C++、文本数学或这些语言的组合来编程嵌入式处理器。您可以从多个软件架构中进行选择，使您既可以集成以往项目的现存源代码，同时还可利用LabVIEW的特性来节省开发时间。

**FPGA** - LabVIEW提供了一种全功能的编程语言，因此如果您熟悉基于文本的传统编程语言，如C，您就可以放心地使用环境，同时利用新硬件平台FPGA的优势。

## 2.选择NI嵌入式硬件平台

NI RIO hardware provides the best off-the-shelf platform to solve any demanding control and monitoring task. This NI platform-based approach gives small teams the confidence to build innovative embedded systems without wasting development time and cost on custom design.

NI RIO硬件提供了最好的现成平台来帮助您完成任何需要的控制和监测任务。NI基于平台的方法使得小团队能够自信地构建创新型嵌入式系统，而且不需要浪费开发时间和成本在自定义设计上。



### NI CompactRIO

NI CompactRIO平台提供了一个高性价比的可重配置嵌入式系统，专为需要高性能和高可靠性的应用而设计。该系统结合了开放嵌入式架构和可热插拔的工业I/O模块，具有体积小、可承受极端条件等优点。该平台是一个封装的嵌入式系统，针对工业环境进行加固。除了CompactRIO平台，NI Single-Board RIO提供了相同但无封装的架构。



### 基于NI PXI或PC

PXI是一种基于PC的坚固平台，适用于测量和自动化系统。PXI结合了PCI的电气总线特性与CompactPCI的模块化及Eurocard机械封装的特性，并增加了专门的同步总线 and 主要软件特性。PXI的高性能、低成本部署平台可用于多种领域，例如制造测试、军事和航空、机器监控、汽车和工业测试。PXI于1997年开发，并在1998年发布，其公开的工业标准由PXI系统联盟（PXISA）所管理。

## 3. NI LabVIEW用于FPGA开发

LabVIEW编程环境显然非常适用于FPGA编程，因为它可清晰地显示并行机制和数据流。使用LabVIEW FPGA模块，擅长C语言的工程师和科学家也可以在他们的设计中利用FPGA，而无需学习硬件设计。此外，FPGA接口C API提供了FPGA和实时应用程序之间的通信层，用于实现I/O控制和数据流。有了这个API，您可以使用LabVIEW图形化工具对FPGA进行编程，并选择LabVIEW或C/C++工具来对系统内的处理器进行编程。

LabVIEW FPGA可帮助您创建自定义测量和控制硬件而无需底层硬件描述语言或板卡级设计。您可以使用此自定义硬件来实现独特的定时和触发程序、超高速控制、数字协议连接、数字信号处理(DS)，P以及许多其他需要高速硬件可靠性和严格确定性的应用。

### LabVIEW FPGA资源

»教程: [FPGA技术介绍: 五大优势](#)

»教程: [LabVIEW FPGA入门](#)

»下载: [评估LabVIEW FPGA模块](#)

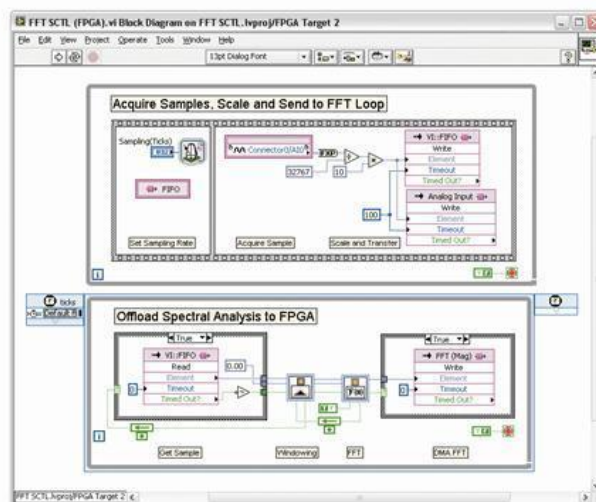


图3. LabVIEW FPGA中定时循环

## 4. C/C++实时应用程序开发

程序员有几种不同的软件架构可供选择，但是他们必须首先将项目需求与最佳的软件设计进行匹配。他们必须考虑软件性能预期等技术需求以及项目截止日期、现有代码重用和团队成员的编程语言技能等业务需求。

例如，系统设计人员必须决定如何最好地利用LabVIEW的生产效率优势来进行实时编程，同时集成之前项目的C/C++代码。一种解决方案是从LabVIEW应用程序中调用C或C++共享库，C或C++共享库使用的是LabVIEW固有的并行线

程调度。设计人员还可以使用内置的 LabVIEW 信号处理函数或结合 LabVIEW MathScript RT 模块使用.m 算法，进一步节省开发精力。另外，程序员可以使用 Eclipse 或任何其他倾向的集成开发环境 (IDE) 来开发、调试和部署完全使用 C 或 C++ 编写的应用程序，同时利用 LabVIEW 的优势来编程 FPGA。

以下是四种基本的设计模式：

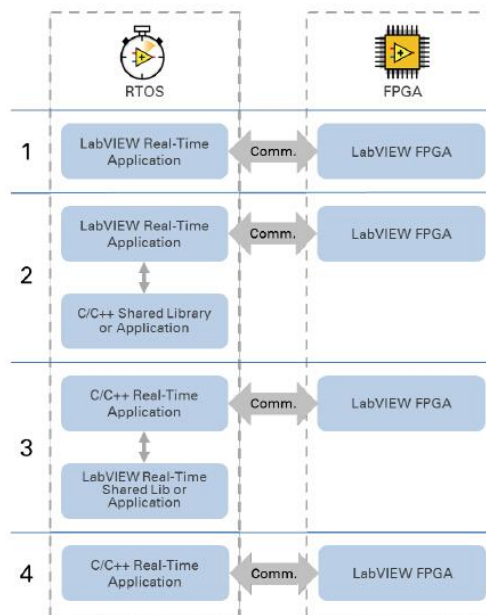


图4. C/C++嵌入式软件设计架构

### 1 LabVIEW用于实时应用程序和FPGA代码

这种方法简化了开发过程，因为程序员可以使用相同的LabVIEW编程环境和语言以及LabVIEW Real-Time和LabVIEW FPGA模块分别编写实时应用程序和FPGA代码。LabVIEW还提供RIO扫描引擎选项，它可使每个I/O模块直接用于LabVIEW Real-Time而无需FPGA编程。

### 2 从LabVIEW调用C/C++库或应用程序

通过使用LabVIEW来编程主实时应用程序，开发人员可利用内置的信号处理函数和网络通信。此外，LabVIEW自动地并行执行独立的代码部分，使程序员可以避免手动管理多个线程。然而，他们可以通过从LabVIEW中调用C/C++共享库或并行运行C/C++应用程序和LabVIEW Real-Time可执行程序来复用现有C或C++代码。有几种方式可以用来集成C/C++

共享库或与外部应用程序进行通信：

- **LabVIEW调用库函数节点** ——使用此函数可以直接从LabVIEW调用C/C++ DLL或Linux共享库函数。配置调用库函数节点，以指定库、函数、参数、返回的节点值、调用规范和回调函数。LabVIEW数据类型可作为输入传输到C/C++库函数或作为输出返回到C/C++库函数。
- **导入共享库向导** ——该工具用于为共享库中的函数生成包装VI的LabVIEW项目库，使得函数更容易使用，无需每次使用时都要配置调用库函数节点。
- **System Exec VI** ——使用此函数来执行或从LabVIEW代码中启动另一个命令行应用程序或Linux脚本文件。您可以在执行命令支持的命令字符串中添加任何参数。只有NI Linux Real-Time提供此功能。
- **本地主机通信** ——并行程序也可以使用本地主机TCP/IP函数来回传输数据，实现彼此之间的通信。

如果要编译NI嵌入式硬件的C/C++代码，请参考以下C/C++开发工具部分，查找使用您硬件的C/C++工具链。

**注意：**有些架构仅通过NI Linux Real-Time支持。并行运行C/C++应用程序和LabVIEW Real-Time应用程序或使用System Exec VI仅在运行NI Linux Real-Time的嵌入式硬件上可支持。

### 3 从C/C++实时应用程序调用LabVIEW函数库

如果您的主实时应用程序是用C或C++编写的，您仍然可以通过使用LabVIEW分析函数进行信号处理，这可通过创建一个LabVIEW共享库，并从C/C++代码调用该库来实现。

»了解如何构建LabVIEW共享库用于外部应用程序

为了避免传输数据和控制FPGA发生冲突，可在LabVIEW或C/C++代码中打开一个FPGA引用。在这种设计模式下，控制FPGA和传输数据到FPGA是由C/C++程序通过第4节描述的FPGA接口C API来进行处理。如果要编译NI嵌入式硬件的C/C++代码，可参考C/C++开发工具部分，找到适合您硬件的C/C++工具链。

### 4 C/C++实时应用程序

您也可以使用Eclipse或任何其他偏好的IDE来开发、调试和部署使用C或C++编写的程序，同时利用LabVIEW的优势来编程FPGA。FPGA接口C API提供了FPGA和实时应用程序之间的通信层，用于实现I/O控制和数据流。这提供了用于访问FPGA输入控件、显示控件、DMA FIFO、中断和数组的接口。

所有的NI嵌入式产品均有兼容的C/C++编译器和工具链。可参考以下C/C++开发工具部分，找到适合您硬件的C/C++

+工具链和开发者教程。

## 5.使用FPGA接口C API

FPGA接口C API使得C / C++开发人员能够更轻松利用LabVIEW FPGA和NI嵌入式硬件来实现嵌入式控制和采集应用。借助FPGA接口C API，开发人员可以使用LabVIEW图形化工具在NI硬件中编程FPGA，还可使用C / C++工具，如Eclipse、Wind River Workbench或GNU Compiler Collection (GCC)工具链来编程实时操作系统。

FPGA接口C API可通过[ni.com/downloads](http://ni.com/downloads)免费下载。该API提供了FPGA和实时应用程序之间的通信层，用于实现I/O控制和数据流。LabVIEW FPGA代码开发结束后，会生成一个头文件，该文件提供了FPGA输入控件、显示控件、DMA FIFO、中断和数组的C引用。FPGA接口C API并没有提供RIO扫描引擎接口。

以下列出了开发过程的每个步骤：

1. 开发和编译LabVIEW FPGA VI，这会产生一个比特文件；
2. 启动FPGA接口C API发生器。这时会根据比特文件生成一个C头文件；
3. 将头文件添加到C/C++应用程序中；
4. 编译并部署应用程序至嵌入式硬件。

## FPGA接口C API资源

» 下载：[FPGA接口C API](#)

» 用户手册：[FPGA接口C API帮助文档](#)

## 6. NI CompactRIO C/C++开发工具

CompactRIO嵌入式系统采用了最新的嵌入式处理器和实时操作系统技术，推出后便受到业界的广泛欢迎。NI RIO系列产品涵盖x86和PowerPC处理器，现在也包含了基于ARM的系统。这些产品可兼容一系列C / C++编译器和工具链。如需确定哪些处理器技术最适合您的嵌入式硬件，请参考这篇[知识库](#)文章或图5。



图5. CompactRIO C/C++开发工具

## 基于ARM和x86的NI Linux Real-Time操作系统

NI持续投资于NI Linux Real-Time这个基于Linux的实时操作系统。该操作系统采用的技术为CompactRIO平台提供了更高层次的开放性，使您可以在LabVIEW代码中访问丰富的Linux应用程序和IP库，Linux应用程序和IP主要来源于庞大的Linux社区。此外，Linux操作系统可更轻松地移植现有C / C++代码，开发人员可以使用Eclipse或偏好的IDE开发、调试和部署代码到CompactRIO硬件上。

Learn More About CompactRIO Devices Based on NI Linux Real-Time 了解更多基于NI Linux Real-Time的CompactRIO设备

- [NI cRIO-9068](#)
- [NI cRIO-903x](#)
- [sbRIO-9651模块系统\(SOM\)](#)

## 必需的软件工具链

1. RTOS——以下工具至少一个
  1. [LabVIEW Real-Time模块](#)
  2. [Eclipse](#)（包含用于ARM编译器的GNU/Linux）
  3. GNU/Linux编译器：
    - 用于ARMv7 Linux的GNU C & C++编译器（Linux操作系统）
    - 用于ARMv7 Linux的GNU C & C++编译器（Windows操作系统）
    - 用于x64 Linux的GNU C & C++编译器（Linux操作系统）
    - 用于x64 Linux的GNU C & C++编译器（Windows操作系统）
2. RTOS-FPGA通信层—[FPGA接口C API](#)（13.0版本或更高版本）
3. FPGA development—all of the following are required: FPGA开发——以下必需全部具备
  1. [LabVIEW](#)（完整版或更高版本）
  2. [LabVIEW FPGA模块](#)
  3. [NI-RIO设备驱动程序](#)（2013版本及更高版本）

## 开发资源

» 教程：[用于NI Linux Real-Time和Eclipse版本的C/C++开发工具入门](#)



- » 教程: [在运行NI Linux Real-Time的CompactRIO上集成C代码与LabVIEW](#)
- » 范例: [用于NI Linux Real-Time和Eclipsed的FPGA接口C API](#)
- » 用户群: [NI Linux Real-Time社区](#)

## PowerPC上的VxWorks实时操作系统

CompactRIO和NI Single-Board RIO的PowerPC版本Wind River VxWorks RTOS。二进制文件可以由编译PowerPC PPC603架构和链接至标准VxWorks库任何工具链创建。NI建议使用Wind River Workbench来开发用于VxWorks的二进制文件,因为它提供了先进的调试功能、功能齐全的开发环境和技术支持。

### 必需的软件工具链

1. RTOS——以下工具至少一个
  1. [LabVIEW FPGA模块](#)
  2. [GCC](#)
  3. [Wind River Workbench](#)
2. RTOS-FPGA通信层—[FPGA接口C API](#)
3. FPGA开发——以下必需全部具备
  1. [LabVIEW \(完整版或更高版本\)](#)
  2. [LabVIEW FPGA模块](#)
  3. [NI-RIO设备驱动程序](#)

### 开发资源

- » 教程: [权威指南: 使用C/C++编程NI VxWorks Real-Time控制器](#)
- » 教程: [开发用于cRIO-901x和其他VxWorks硬件的共享库](#)

## 基于x86的Phar Lap实时操作系统

CompactRIO硬件的x86版本运行的是Phar Lap实时操作系统。NI已经对使用Visual C++ 2003和2008编译器来创建用于该系列嵌入式硬件的C/C++二进制文件进行测试。

### 必需的软件工具链

1. 以下工具至少一个
  1. [LabVIEW FPGA模块](#)
  2. [Visual C++](#)
2. RTOS-FPGA通信层—[FPGA接口C API](#)
3. FPGA开发——以下必需全部具备
  1. [LabVIEW \(完整版或更高版本\)](#)
  2. [LabVIEW FPGA模块](#)
  3. [NI-RIO设备驱动程序](#)

## 7.基于NI PXI或PC的C/C++开发工具

NI PXI硬件不包括内置的FPGA。该平台是完全可配置的,所以您可以针对您的应用程序选择合适的机箱、控制器和模块。您还可以从多个[PXI R系列基于FPGA的模块](#)以及其它数百个[PXI硬件模块](#)中进行选择,以实现系统设计的最大灵活性。

NI提供了一个ANSI C ID——[NI LabWindows™/ CVI](#),以简化基于x86的PXI和基于PC的嵌入式系统开发。LabWindows/CVI中不包含C++编译器或函数库。C++开发人员可以使用他们倾向的IDE开发用于NI PXI的应用程序或PC应用程序。



图7. 基于PXI或PC的C/C++开发工具

LabWindows / CVI提供了集成的驱动程序API、I/O配置助手、测量和分析库以及全面调试和代码优化工具,可帮助您缩短开发时间。

您可以在LabWindows / CVI应用程序中复用现有C源代码以及在远程实时硬件上使用[LabWindows/CVI Real-Time模块](#)进行部署和调试而无需离开IDE。NI提供了适用于所有PXI和PC硬件的高性能和直观驱动程序API以及用于实时应用程序

和FPGA硬件之间的I/O控制和数据传输的FPGA接口C API。

#### 必需的软件工具链

1. RTOS——以下工具至少一个
  1. [LabVIEW](#)和[LabVIEW Real-Time](#)模块
  2. [LabWindows/CVI](#)和[LabWindows/CVI Real-Time](#)模块
2. Windows—[LabVIEW](#) [LabWindows/CVI for ANSI C](#)或其他用于C/C++的IDE

#### 使用PXI R系列FPGA硬件所需的其他软件

1. 处理器-FPGA通信层——[FPGA接口C API](#)
2. FPGA开发——[LabVIEW FPGA](#)模块

#### 开发者资源

- » 产品页面: [了解更多关于LabWindows/CVI](#)
- » 教程: [使用LabWindows/CVI使用R系列FPGA接口主机应用程序](#)
- » 下载: [评估LabWindows/CVI IDE](#)