

LabVIEW™ Real-Time 模块

入门指南

本文档通过一系列练习介绍如何开发实时项目和 VI，包括搭建 RT 终端，创建、调试和部署实时应用程序。除了练习之外，本文档还提供创建实时应用程序时所需的 *LabVIEW 帮助* 和其它 Real-Time 模块文档的相关参考信息。

目录

安装和配置 Real-Time 模块	2
安装 Real-Time 模块	2
配置 RT 终端	2
安装 RT 终端软件	4
安装多 CPU 支持	4
查看 LabVIEW 文档	5
LabVIEW 帮助	5
LabVIEW Real-Time 模块发行和升级说明	5
LabVIEW Real-Time 模块自述文件	6
创建实时项目和应用程序	6
使用 Real-Time Project Wizard 创建项目	6
为应用程序选择 RT 终端和主机配置	8
添加 RT 终端至项目	10
在项目浏览器窗口中查看项目和应用程序	10
配置 RT 终端的属性	13
部署和运行 RT 终端上的 VI	14
生成独立的实时应用程序	14
创建实时程序生成规范	15
编辑实时程序生成规范	16
生成和运行独立的实时应用程序	16
监测 RT 终端资源和错误	17
监测 RT 终端资源	17
查看 RT 终端错误	17
调试 RT 终端 VI 和独立应用程序	18
调试 RT 终端 VI	18
调试独立的实时应用程序	18
使用 Real-Time Execution Trace 工具包	19
总结	19
安装 Real-Time 模块和配置 RT 终端	19
使用 Real-Time Project Wizard 创建 RT 项目	19
使用项目浏览器窗口管理和配置项目	20

使用 LabVIEW 应用程序生成器创建独立实时应用程序	20
使用 NI 分布式系统管理器管理 RT 终端资源	20
使用 RT 终端错误对话框查看 RT 终端的错误记录	20
调试 RT 终端 VI 和独立应用程序	21
参考资料	21
Real-Time 模块基本概念	21
Real-Time 模块范例	21

安装和配置 Real-Time 模块

必须将 Real-Time 模块安装在进行开发的计算机上，也就是主机上，并在 RT 终端上配置和安装软件。

使用 NI Measurement & Automation Explorer (MAX) 访问 NI 设备和系统。MAX 可与联网的 RT 终端通信。联网的 RT 终端是和主机位于同一个子网的设备，即远程系统。在主机上安装 Real-Time 模块，然后使用 MAX 配置 RT 终端并在终端上安装 Real-Time 模块和相应的驱动程序。

安装 Real-Time 模块

安装 Real-Time 模块和设备驱动程序之前，必须先在主机上安装 LabVIEW 开发系统。关于 LabVIEW 安装的说明信息，见 *LabVIEW 发行说明*。关于 Real-Time 模块和设备驱动程序安装的说明信息，见软件随附的 *LabVIEW Real-Time 模块发行和升级说明*，或访问 ni.com/manuals 获取。

配置 RT 终端

在 RT 终端上安装软件或驱动程序之前，必须先使用 MAX 配置终端的基本设置。



注 首次配置时，必须将联网 RT 终端连接至运行 MAX 的主机所在的子网。

图 1 显示的是一个 RT 系列 PXI 终端，PXI-8106，配置为自动从 DHCP 服务器自动获取 IP 地址，如图 1 IP 设置部分所示。

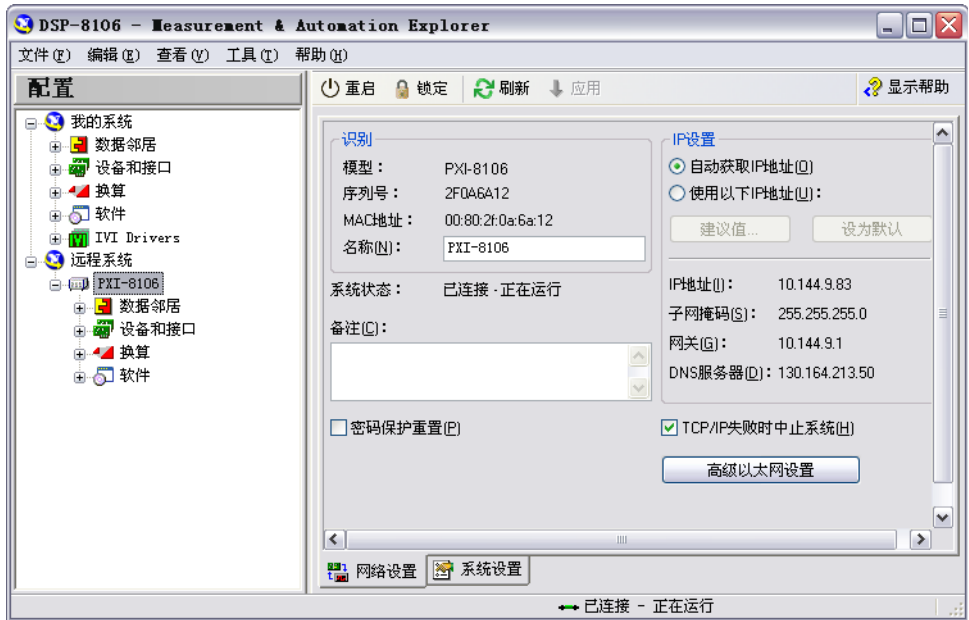


图 1 配置 RT 终端网络设置

关于配置联网 RT 终端的详细信息，请在 MAX 中选择**帮助»MAX 帮助**，打开 MAX 帮助中**目录**栏的 MAX 远程系统帮助部分。关于将台式计算机配置为联网 RT 终端的详细信息，请参考将台式计算机作为 RT 终端与 RT 模块配合使用。上述文档位于目录 labview\manuals\RT_Using_PC_as_RT_Target.pdf。

关于配置 RT 终端，请参考下列资源。

- **Real-Time (RT) Installation/Configuration Troubleshooter** — 访问 ni.com/info 并输入信息代码 rtconfig，查看 *Real-Time (RT) Installation/Configuration Troubleshooter*，其中包含了安装和配置 RT 终端的有用链接和参考信息。
- **LabVIEW Real-Time 终端配置教程** — 参考 MAX 帮助中的 *LabVIEW Real-Time 终端配置教程*，获取配置联网 RT 终端的分步教程。
- **其它和高级设置** — LabVIEW 帮助中的 *Configuring RT Target Settings* 提供了配置 RT 终端的其它设置信息。

安装 RT 终端软件

在主机上使用 MAX 中的 LabVIEW Real-Time 软件向导，在 RT 终端上安装软件和驱动程序。在 MAX 中展开终端，右键单击终端下的**软件**，从快捷菜单中选择**添加 / 删除软件**，打开 LabVIEW Real-Time 软件向导。

图 2 显示了 LabVIEW Real-Time 软件向导，选择在 RT 终端上安装 LabVIEW Real-Time 模块。LabVIEW Real-Time 软件向导显示主机上安装的所有 NI 软件，可从中选择在 RT 终端上安装的软件。

安装多 CPU 支持

如要利用多 CPU 系统的并行处理的优点，使用 MAX 中的 LabVIEW Real-Time 软件向导安装 NI RT Extensions for SMP，如图 2 所示。关于优化应用程序以用于多 CPU 系统的详细信息，请参考 *LabVIEW 帮助* 中目录栏的 *Optimizing RT Applications for Multiple-CPU Systems* 部分。



注

对于单 CPU 系统，不需安装 NI RT Extensions for SMP 即可实现最佳性能。另外，一些应用程序，例如，主要由单点 I/O 构成的应用程序，使用单 CPU，不安装 NI RT Extensions for SMP 时延迟较少。关于优化 RT 应用程序以用于多 CPU 系统的详细信息，请登录 NI 网站 ni.com/info 并输入信息代码 `rtsmc` 查询。



图 2 安装 RT 终端的软件和驱动程序

关于使用 LabVIEW Real-Time 软件向导在 RT 终端上安装软件和驱动程序的详细信息，见 *MAX 帮助* 中的 *添加和删除软件* 部分。

查看 LabVIEW 文档

Real-Time 模块随附有详尽的产品文档，这些文档可帮助您创建在 RT 终端上运行的确定性应用程序。

LabVIEW 帮助

*LabVIEW 帮助*包括下列 Real-Time 模块的专属文档。

- **Real-Time Module Concepts** 一包括创建确定性应用程序的编程概念、应用程序架构和 Real-Time 模块的功能。
- **Real-Time Module How-To** 一包括使用 Real-Time 模块功能的分步指导。
- **Real-Time VIs** 一包括 RT VI、函数和错误代码的参考信息。
- **Unsupported LabVIEW Features** 一包括 Real-Time 模块不支持的 LabVIEW 功能的详细信息。

按照下列步骤，查看 *LabVIEW 帮助* 的 Real-Time 模块部分。

1. 打开 LabVIEW，显示**启动**窗口。在启用窗口，可新建 VI 和实时项目，选择最近打开的 LabVIEW 文件、查找范例和打开 *LabVIEW 帮助*。
2. 选择**帮助**»**搜索 LabVIEW 帮助**，打开 *LabVIEW 帮助*。
3. 展开 *LabVIEW 帮助*中**目录**栏的 **Real-Time Module** 部分。



注

单击**搜索**栏，在半角双引号中输入搜索主题，可在 *LabVIEW 帮助*中搜索特定的主题。

LabVIEW Real-Time 模块发行和升级说明

*LabVIEW Real-Time 模块发行和升级说明*包含安装和配置 Real-Time 模块的信息，以及升级项和新增功能的列表。按照下列步骤查看文档：



注

必须装有 Adobe Reader 6.0.1 或更高版本才能查看或搜索文档的 PDF 版本。

1. 选择**开始**»**所有程序**»**National Instruments**»**LabVIEW**»**LabVIEW 手册**，打开 labview\manuals 目录。该目录下包含 LabVIEW 手册的 PDF 版本。也可从 ni.com/manuals 获取这些文档。
2. 双击 RT_Release_Upgrade_Notes.pdf，打开该文档。
3. *安装 Real-Time 模块*部分是安装 Real-Time 模块的详细信息。
4. *Real-Time 模块新增功能*部分是最新版本 Real-Time 模块的新增功能列表。

LabVIEW Real-Time 模块自述文件

readme_RT.html 文件中包含最新版本 Real-Time 模块中已修正的程序漏洞。按照下列步骤查看该文件：

1. 选择**开始 » 所有程序 » National Instruments » LabVIEW » 自述文件**，打开 labview\readme 目录。该目录下包含 LabVIEW、LabVIEW 模块和附加软件的自述文件。
2. 双击 readme_RT.html，打开 *LabVIEW Real-Time 模块自述文件*。

创建实时项目和应用程序

装好 Real-Time 模块后，必须使用 LabVIEW 项目来创建实时应用程序并使程序在 RT 终端上运行。项目用于组合 LabVIEW 文件和非 LabVIEW 文件、创建程序生成规范以及在终端部署文件。保存项目时，LabVIEW 会创建一个项目文件 (.lvproj)，其中包括对项目中文件的引用、配置信息、生成信息以及部署信息等。

本文档的范例仅使用 Real-Time Project Wizard 创建一个确定性应用程序：模拟数据采集、记录数据至 RT 终端的磁盘，然后在主机上以图表形式显示数据。应用程序由两个 VI 组成：一个是在终端上运行的 RT 终端 VI，模拟确定性数据采集和记录；另一个是在主机上运行的主控 VI，提供用户界面。



完成该练习大约需要 35 分钟。

使用 Real-Time Project Wizard 创建项目

使用 Real-Time Project Wizard 创建新项目，该项目定义 RT 终端并包含相关 VI 和其它应用程序文件。创建项目可使用下列三种架构之一：

- **连续通信架构 (Continuous communication architecture)** — 连续通信架构用于创建数据采集应用程序，该程序可记录数据至磁盘并与主机保持连续通信，或进行连续确定性任务的应用程序。
- **状态机架构 (State machine architecture)** — 状态机架构用于实现复杂的决策算法，通常由状态图或流程图表示。状态机架构可用于实现任何摩尔机 (Moore Machine) 表示的算法。而摩尔型状态机是对状态图中的每一个状态产生一个响应。
- **自定义项目 (Custom Project)** — 自定义项目架构用于添加空白 VI 或导入现有 VI 至主机或 RT 终端。

关于使用 Real-Time Project Wizard 和实时架构的详细信息，请参考 *LabVIEW 帮助*。

按照下列步骤使用 Real-Time Project Wizard 创建一个项目。

1. 在启动窗口的**新建**部分单击**实时项目**，或选择**工具 » RT 模块 » Project Wizard**，打开 Real-Time Project Wizard。
2. 从 **Project type** 下拉菜单中选择 **Continuous communication architecture**。

Real-Time Project Wizard 在右窗格中显示刚才所选架构的示意图，如图 3 所示。

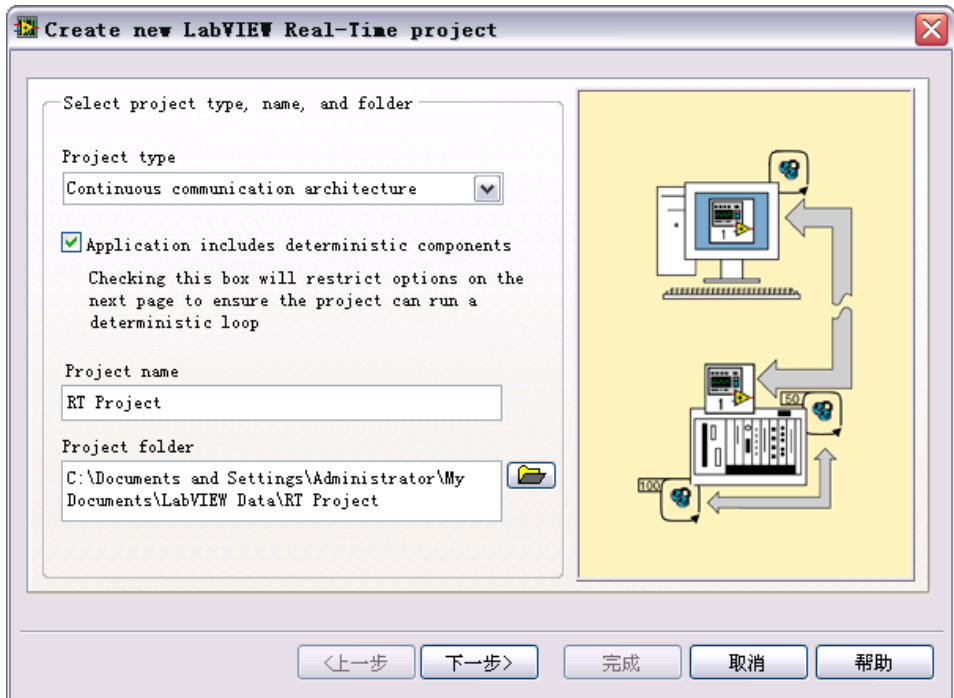


图 3 选择项目类型

3. 勾选 **Application includes deterministic components** 复选框。启用该选项后，Real-Time Project Wizard 将项目的配置选项限制于保证应用程序确定性运行的选项。
4. 在 **Project name** 文本框中输入 RT Project。
5. 在 **Project folder** 文本框中输入在主机上保存项目文件的目录，或单击**浏览**按钮在文件系统中查找保存项目文件的目录。
6. 单击**下一步**按钮，更新设置并继续。

为应用程序选择 RT 终端和主机配置

使用 Real-Time Project Wizard 的 **Customize architecture options** 部分指定应用程序中 VI 的选项。对于连续通信架构而言，Real-Time Project Wizard 创建一个在 RT 终端上运行的 VI，其中包含 LabVIEW 定时循环、控制实时任务的共享变量、用户界面和文件 I/O。

定时循环根据用户指定的循环间隔执行子程序框图或帧。定时循环使用定时源控制子程序框图的定时和执行。关于使用定时循环和其它定时结构的详细信息，请参考 *LabVIEW 帮助*。

共享变量是经过配置的软件实体，用于在 VI 中不能使用连线的位置间传递数据，在 RT 终端上的 VI 之间传递数据，以及在网络中不同终端上的 VI 之间传递数据。创建共享变量后，可在 VI 中使用共享变量节点访问共享变量数据。共享变量节点是程序框图上的对象，用于引用一个共享变量。共享变量节点用于从共享变量读取数据，以及将数据写入共享变量。

Real-Time 模块向共享变量添加了实时 FIFO（先进先出缓存）功能。如启用了共享变量的实时 FIFO 功能，可共享实时数据而不影响 RT 终端上 VI 的确定性。关于使用共享变量确定性共享数据的详细信息，请参考 *LabVIEW 帮助*。

如选择连续通信架构，可为 RT 终端 VI 选择使用包含 1 或 2 个定时循环的应用程序配置。

- **One loop**—RT 终端 VI 使用一个定时循环控制应用程序任务的定时和执行。
- **Two loops**—RT 终端 VI 使用两个优先级不同的定时循环控制应用程序任务的定时和执行。高优先级的定时循环控制实时任务。低优先级的定时循环控制用户界面通信和文件 I/O 任务。

Real-Time Project Wizard 使用主机上的 VI 作为实时应用程序的用户界面，或使用 LabVIEW 远程前面板功能连接至 RT 终端 VI。

- **主控 VI (Host VI)**—使用 LabVIEW 共享变量从 RT 终端 VI 发送用户界面数据。主机上的 VI 获取用户界面数据并将数据显示在前面板上。
- **远程前面板 (Remote Panel)**—使用 LabVIEW 远程前面板查看 RT 终端 VI 的前面板控件。如应用程序要求确定性执行，则不能在配置为只包含一个定时循环的 VI 上使用远程前面板。

按照下列步骤，使用 Real-Time Project Wizard 配置包含两个定时循环的 RT 终端 VI 和主控 VI 用户界面。

1. 选择 **Two loops**，如图 4 所示，创建包含两个定时循环的 RT 终端 VI。
Real-Time Project Wizard 将共享变量添加至 RT 终端 VI，发送控制实时任务的高优先级定时循环中的数据。

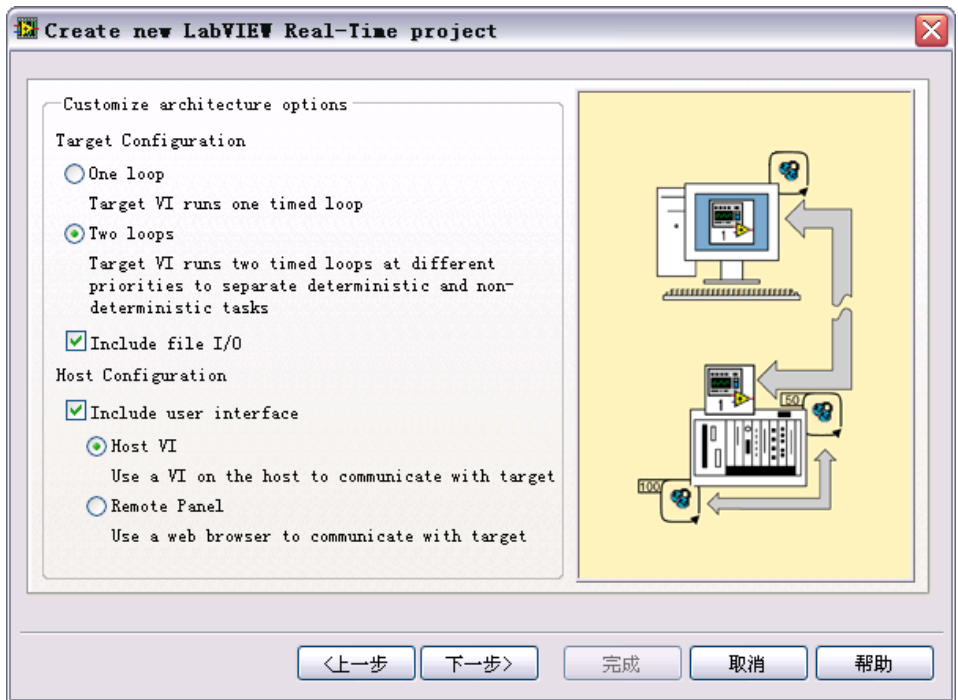


图 4 配置程序架构

2. 勾选 **Include file I/O** 复选框，将 RT 终端 VI 实时循环中采集到的数据记录到 RT 终端的文件中。
启用 **Include file I/O** 选项后，高优先级的定时循环通过共享变量发送采集到的数据至低优先级的定时循环。低优先级的定时循环将数据保存在 RT 终端的文件中。
3. 勾选 **Include user interface** 复选框，将实时应用程序配置为发送用户界面数据至主机的 VI。
4. 选择 **Host VI** 创建一个 VI，该 VI 在主机上运行并显示实时应用程序的前面板。
5. 单击 **下一步** 按钮，更新设置并继续。

添加 RT 终端至项目

使用 Real-Time Project Wizard 的 **Browse targets** 部分，添加 RT 终端至项目。添加至项目的 RT 终端将执行项目中的 RT 终端 VI。

按照下列步骤使用 Real-Time Project Wizard 将 RT 终端添加至项目。



注

添加网络 RT 终端至项目之前，必须先在 MAX 中配置 RT 终端。

1. 单击**浏览**按钮，打开 **Add Targets and Devices** 对话框。使用 **Add Targets and Devices** 对话框选择之前配置的 RT 终端，然后单击**确定**按钮。**Selected target** 文本框中将显示刚才选择的 RT 终端的名称。
2. 单击**下一步**按钮，更新设置并继续。

Real-Time Project Wizard 显示项目和刚才配置的应用程序的预览。预览中将显示项目中的主机、RT 终端和 VI。本教程接下来将介绍项目的组成以及深入学习应用程序 VI。

Real-Time Project Wizard 创建项目、应用程序和支持文件完毕后，可勾选 **Open VIs when finished** 复选框，打开所有项目文件。

3. 单击**完成**按钮退出 Real-Time Project Wizard 并生成实时应用程序所需的新项目、VI 和支持文件。**项目浏览器**窗口将显示刚才创建的新项目。

Real-Time Project Wizard 创建的项目可作为创建实时应用程序的起点。也可在 LabVIEW 中手动创建项目，并将 RT 终端和 VI 添加至项目。关于手动创建和编辑 Real-Time 模块项目的详细信息，见 *LabVIEW 帮助* 中的 *Using LabVIEW Projects with the Real-Time Module* 部分。

在项目浏览器窗口中查看项目和应用程序

项目浏览器窗口用于统一管理 RT 终端、VI 和其它项目支持文件。

项目浏览器窗口有两个选项卡。

- **项目根目录**—包含当前项目的主机和 RT 终端。
 - **我的电脑**—显示项目中的主机。
 - **RT 终端**—显示项目中的 RT 终端。默认情况下，RT 终端名以其在 MAX 中的名称开头。

添加至 RT 终端的 VI 和库显示在**项目浏览器**窗口的终端下。终端下显示的 VI 表示可在该终端上运行该 VI。图 5 显示的是通过 Real-Time Project Wizard 创建的项目 RT Project.lvproj。

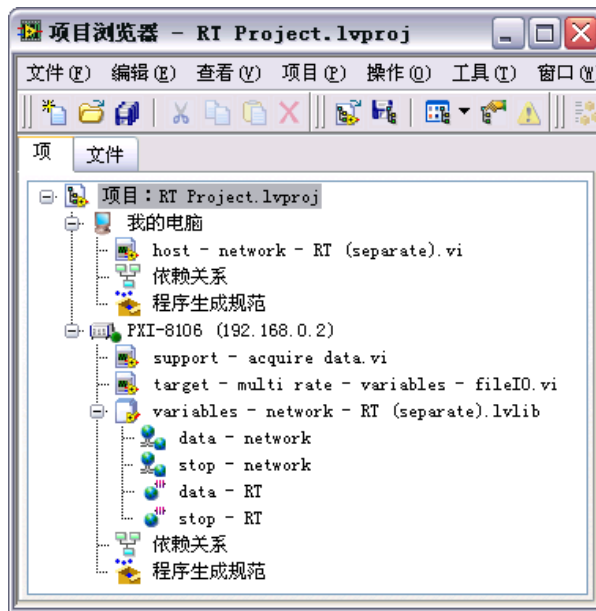


图 5 在项目浏览器窗口中查看实时项目和应用程序

在**我的电脑**下，项目中有用户界面 VI host - network - RT (separate).vi。RT 终端 VI target - multi rate - variables - fileIO.vi 在 RT PXI Target 下显示。在 RT 终端下，还有一个项目库 variables - network - RT (separate).lvlib，其中包含用于与主机通信而创建的共享变量。可使用项目库设置和编辑共享变量的访问权限。关于使用项目库的详细信息，见 *LabVIEW 帮助*。

按照下列步骤，查看通过 Real-Time Project Wizard 在项目中创建的共享变量和 VI。

1. 在**项目浏览器**窗口中，双击 data - network 共享变量，位于 variables - network - RT.lvlib 项目库下，打开**共享变量属性**对话框。

共享变量属性对话框用于定义共享变量的类型和其它设置。使用**共享变量属性**对话框编辑现有共享变量或创建新的共享变量。在**项目浏览器**窗口右键单击**我的电脑**或 RT 终端，从快捷菜单中选择**新建 » 变量**，打开**共享变量属性**对话框。关于创建共享变量的详细信息，请参考 *LabVIEW 帮助*。

2. 在**类别**列表中选择**实时 FIFO**，打开**实时 FIFO**页。请注意**启用实时 FIFO**对话框，启用共享变量的 FIFO。使用**实时 FIFO**页配置共享变量的实时 FIFO，单击**完成**按钮关闭**共享变量属性**对话框。

3. 双击 RT 终端下的 target - multi rate - variables - fileIO.vi, 然后按下 <Ctrl-E> 打开程序框图。

target - multi rate - variables - fileIO.vi 包含两个定时循环, 确定性循环和非确定性循环, 如图 6 所示。确定性循环中有一个模拟数据采集的 VI。可将该 VI 替换为任何自定义数据采集 VI。

确定性循环中有一个共享变量节点, 将采集到的数据写入单进程共享变量 data。然后, 非确定性循环通过共享变量节点读取数据并将数据写入文件 data.bin。非确定性循环使用网络发布共享变量将数据发送至主机。两个定时循环均从共享变量 stop 读取值, 并通过该值判断是否停止执行。

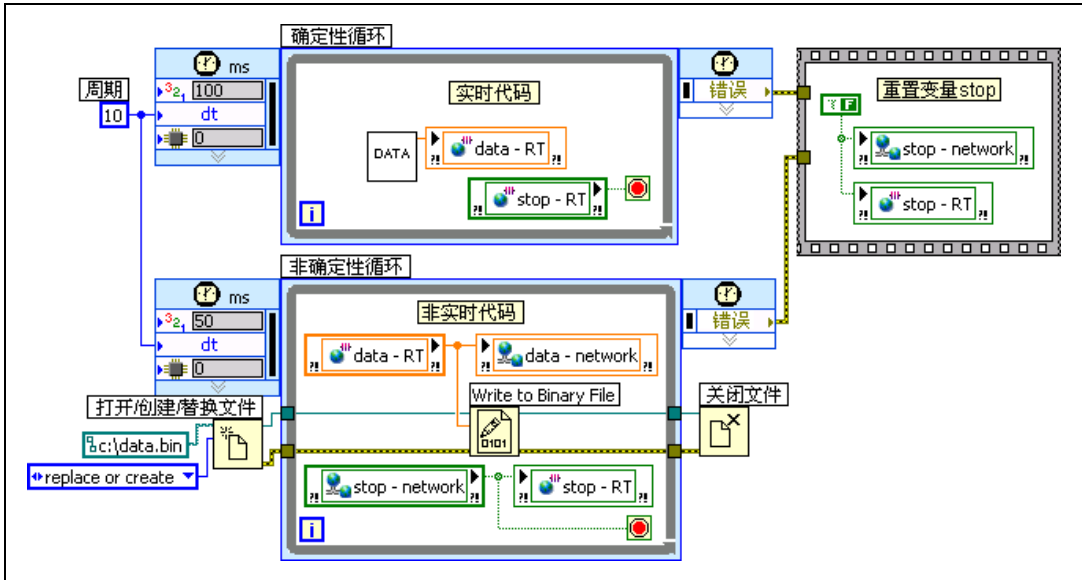


图 6 包含两个定时循环的 RT 终端 VI

4. 双击我的电脑下的 host - network - RT (separate).vi, 然后按 <Ctrl-E> 显示其程序框图。

如图 7 所示, host VI 中有一个 While 循环, 该循环从共享变量 data - network 读取数据, 再将数据发布至前面板的波形图。该 host VI 中有一个共享变量节点, 该节点写入共享变量 stop, 允许用户从主机控制 RT 终端 VI。

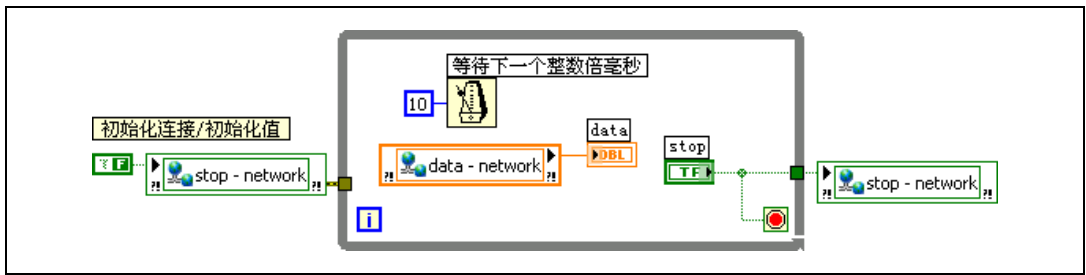


图 7 用户界面主控 VI

5. 注意**项目浏览器**窗口的 RT 终端也包括**依赖关系**和**程序生成规范**。

依赖关系下是 RT 终端运行时所需的子 VI 和 VI。关于在项目中查看依赖关系的详细信息，请参考 *LabVIEW 帮助*。

使用**程序生成规范**创建和管理独立的实时应用程序。关于使用**程序生成规范**创建独立的实时应用程序的详细信息，请参考本文档的[生成独立的实时应用程序](#)部分。

配置 RT 终端的属性

使用 RT 终端的**属性**对话框配置 RT 终端的项目设置。可配置常规、VI 服务器、Web 服务器、访问等其它终端设置。例如，启用 RT 终端上的 VI 服务器功能，本节将介绍相关内容。



完成该练习大约需要 10 分钟。

按照下列步骤，启用 RT 终端的 VI 服务器功能。

1. 在**项目浏览器**窗口右键单击 RT 终端，并从快捷菜单中选择**属性**，打开 RT 终端的**属性**对话框。
2. 从**类别**列表中选择**VI 服务器：配置**，打开**VI 服务器：配置**页。
3. 勾选 **TCP/IP** 复选框，在 RT 终端上启用 VI 服务器。
4. 确保**可访问的服务器资源**部分的所有选项均被勾选，允许 RT 终端上的 VI 和应用程序调用。启用 RT 终端上的 VI 和应用程序调用，使用户可远程打开终端内存中的 VI。
5. 单击**确定**按钮，保存设置并关闭**属性**对话框。
6. 在**项目浏览器**窗口，选择**项目 » 保存项目**，保存 RT 终端和 RT 终端**属性**对话框的设置。这些设置在部署 RT 终端后生效。部署 RT 终端就是将项目的设置应用至终端。



注 部署 RT 终端时，RT 终端**属性**对话框的最新设置将覆盖当前终端的设置。

7. 右键单击 RT 终端，从快捷菜单中选择**部署**，将设置部署至终端。

部署和运行 RT 终端上的 VI

除了将属性应用至 RT 终端外，部署还意味着下载 VI 和相关文件至 RT 终端的内存。可将 VI、库和共享变量部署至可运行或引用这些项的 RT 终端。



注 部署操作不在终端的磁盘上保存 VI，所以终端重启后 VI 就从内存中消失。关于生成和运行独立的实时应用程序（即位于 RT 终端，终端上电后运行）的详细信息，请参考本文档的生成独立的实时应用程序部分。



完成该练习大约需要 10 分钟。

按照下列步骤，部署和运行之前在 RT 终端和主机上创建的实时应用程序。

1. 在**项目浏览器**窗口的 RT 终端下，右键单击 `target - multi rate - variables - fileIO.vi`，从快捷菜单中选择**部署**，下载 VI 至 RT 终端。

LabVIEW 将 VI 部署至 RT 终端，然后打开 VI 的前面板，表示 VI 已在 RT 终端的内存中。关闭前面板将把 VI 从 RT 终端的内存中移除。

2. 单击前面板上的**运行**按钮，在终端上运行 VI。
3. 在**项目浏览器**窗口右键单击 RT 终端，从快捷菜单中选择**断开连接**，关闭与终端的前面板连接，保持 VI 继续运行。
4. 在**项目浏览器**窗口，双击**我的电脑**下的 `host - network - RT (separate).vi`，打开 VI。单击前面板上的**运行**按钮，在主机上运行 VI。

`host - network - RT (separate).vi` 将接收到的 RT 终端数据显示在波形图中。

5. 单击**停止**按钮停止 RT 终端 VI 中定时循环的执行，主机上 VI 的执行也相应停止。

生成独立的实时应用程序

LabVIEW 应用程序生成器和 Real-Time 模块可用于创建独立的实时应用程序。独立的实时应用程序可在 LabVIEW 环境之外运行。可将独立的实时应用程序嵌入 RT 终端并让应用程序在终端上电时自动运行。使用应用程序生成器创建一个程序生成规范，该规范定义如何生成一个独立的实时应用程序。



完成该练习大约需要 20 分钟。

创建实时程序生成规范

程序生成规范定义独立的实时应用程序的 VI、支持文件和设置。必须首先创建程序生成规范，然后使用程序生成规范生成独立的实时应用程序。

按照下列步骤，创建程序生成规范。

1. 在**项目浏览器**窗口，右键单击 RT 终端下的**程序生成规范**，从快捷菜单中选择**新建 » Real-Time Application**，打开 **Real-Time Application Properties** 对话框。
2. 在**程序生成规范名称**文本框中输入 Real-Time Application。程序生成规范的名称将出现在 RT 终端的**程序生成规范**下。
3. 在**Target filename**文本框中输入 rtapp.rtexe，该文件是 RT 终端上独立的实时应用程序可执行文件的文件名。默认情况下，**Target filename**文本框中的文件名是 startup.rtexe。
4. 单击**类别**列表的**源文件**，打开**源文件**页。
5. 在**项目文件**树形目录中，选择 RT 终端下的 target - multi rate - variables - fileIO.vi，单击**启动 VI**列表框旁的右向箭头，将 VI 添加至**启动 VI**列表框。**启动 VI**列表框中的 VI 在启动独立实时应用程序时执行。
6. 单击**确定**按钮，保存设置并关闭 **Real-Time Application Properties** 对话框。程序生成规范的名称将出现在**项目浏览器**窗口 RT 终端的**程序生成规范**下，如图 8 所示。

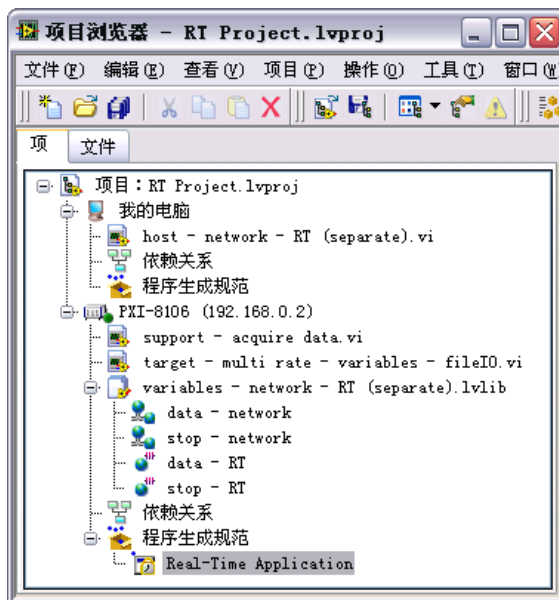


图 8 创建程序生成规范

7. 在**项目浏览器**窗口，选择**项目 » 保存项目**，保存项目和程序生成规范设置。

编辑实时程序生成规范

可使用 **Real-Time Application Properties** 对话框编辑程序生成规范的设置。如改变了程序生成规范，必须保存项目以保存程序生成规范的改动。

按照下列步骤，编辑刚才创建的程序生成规范。

1. 在**项目浏览器**窗口中，右键单击 RT 终端**程序生成规范**下的实时程序生成规范，从快捷菜单中选择**属性**，打开**实时应用程序属性**对话框。
2. 单击**类别**列表中的**高级**，打开**高级**页。
3. 勾选**启用调试**复选框。**启用调试**选项允许用户远程调试使用程序生成规范生成的独立实时应用程序。
4. 单击**确定**按钮，保存设置并关闭 **Real-Time Application Properties** 对话框。
5. 在**项目浏览器**窗口，选择**项目 » 保存项目**，保存程序生成规范。

生成和运行独立的实时应用程序

创建程序生成规范后，可所在 RT 终端上生成和运行独立的实时应用程序。也可将独立实时应用程序设置为 RT 终端上电后自动启动。

按照下列步骤，在 RT 终端上生成和运行独立的实时应用程序。

1. 右键单击程序生成规范，从快捷菜单中选择**生成**，生成独立的实时应用程序。应用程序生成器根据程序生成规范的设置生成一个独立的实时应用程序。生成完毕后，单击**完成**按钮，关闭**生成状态**对话框。
2. 右键单击程序创建规范，在快捷菜单中选择 **Run as startup**。Run as startup 快捷菜单项将应用程序设置为启动时运行的程序，部署应用程序至终端，并提示用户重启 RT 终端。



注

将程序生成规范设置为启动程序后，LabVIEW 将在**项目浏览器**窗口的程序生成规范图标周围显示绿色外框。设置启动应用程序后，从应用程序生成规范生成的应用程序在 RT 终端上电后自动打开。

3. 在**项目浏览器**窗口右键单击包含共享变量的项目库，并禁用 **Autodeploy variables** 选项。如运行主控 VI 时启用 **Autodeploy variables** 选项，LabVIEW 将关闭 RT 终端上的独立应用程序并重新部署共享变量设置。



注

如要使用主控 VI 监控一个已生成的实时应用程序，只需禁用 **Autodeploy variables** 选项。

4. 在**项目浏览器**窗口中，在**我的电脑**下打开并运行 `host.vi`，从采集到的数据和独立实时应用程序发送的数据都显示在图表中。



注

如要继续编辑实时 VI，使其不是在每次 RT 终端上电时都自动运行。右键单击程序生成规范，从快捷菜单中选择 **Unset as startup** 和 **Deploy**，取消应用程序在每次终端上电时启动。也可右键单击项目库，从快捷菜单中选择 **Autodeploy variables** 重新启用自动变量部署。

监测 RT 终端资源和错误

Real-Time 模块提供监测 RT 终端资源和查看终端错误的辅助工具。



完成该练习大约需要 10 分钟。

监测 RT 终端资源

NI 分布式系统管理器显示 RT 终端上运行 VI 的详细信息，并提供终端资源的动态显示，例如，可用内存量和 CPU 占用情况。系统管理器用于停止和启动 RT 终端内存中的 VI，当资源使用超过用户指定的上限后记录产生的警告。



注 使用系统管理器监测终端时必须启用终端上的 VI 服务器。关于启用 RT 终端上 VI 服务器的详细信息，请参考本文档的[配置 RT 终端的属性](#)部分。

按照下列步骤打开系统管理器查看 RT 终端的资源。

1. 在终端上部署和运行之前创建的 RT 终端 VI，`target - multi rate - variables - fileIO.vi`。
2. 在**项目浏览器**窗口中，选择**工具 » 分布式系统管理器**，打开系统管理器。
3. 从**网络项**树形目录中选择 RT 终端。系统管理器将终端信息显示在**自动视图**中。
4. 单击 **CPU/ 内存**选项卡。**自动视图**显示 RT 终端资源的信息。
5. 关闭系统管理器。
6. 关闭前面板将把 VI 从 RT 终端的内存中移除。

关于使用系统管理器监测 RT 终端的详细信息，请参考 *NI 分布式系统管理器帮助*。在系统管理器中，选择**帮助 » 系统管理器概述**，可打开该帮助。

查看 RT 终端错误

RT 终端错误 (RT Target Error) 对话框显示 RT 终端所有会话进程中保存下的错误记录。RT 终端上电即开始一个 RT 会话；RT 终端断电即结束一个 RT 会话。**RT Target Errors** 对话框显示每个 RT 会话的 LabVIEW 错误、实时操作系统错误和故障记录。也可使用 **RT Target Errors** 对话框创建一个错误报表，将报表保存在主机的磁盘上。

按照下列步骤查看 RT 终端会话的错误记录，并创建 RT 会话的报表。

1. 在**项目浏览器**窗口右键单击 RT 终端，并从快捷菜单中选择**工具 » View Error Log**，打开 **RT Target Errors** 对话框。
2. 在 **LabVIEW RT sessions** 中选择当前会话，打开当前会话的错误记录。使用记录调试 RT 终端运行 VI 时发生的错误和故障。
3. 单击 **Create Report** 按钮，找到在主机上保存报表的目录。

4. 在 **File name** 文本框中输入文件名称，单击 **OK** 按钮保存报表。
RT Target Errors 对话框把报表保存为一个文本文件，其中包含选定会话的错误记录。
5. 单击 **Done** 按钮，关闭 **RT Target Errors** 对话框。

调试 RT 终端 VI 和独立应用程序

可使用 LabVIEW 提供的一系列工具调试 RT 终端内存中的 RT 终端 VI 和独立应用程序 (.rtexe)。



完成该练习大约需要 15 分钟。

调试 RT 终端 VI

打开至一个 RT 终端的前面板连接，使用 LabVIEW 调试工具调试终端上运行的 VI。

按照下列步骤打开至 RT 终端上运行 VI 的前面板连接，从主机进行调试。

1. 在终端上部署和运行之前创建的 RT 终端 VI，`target - multi rate - variables - fileIO.vi`。
2. 主机连接至 RT 终端时，使用 LabVIEW 调试工具（高亮显示执行过程、单步调试）步过调试 LabVIEW 代码。关于使用 LabVIEW 调试工具调试 VI 的详细信息，请参考 *LabVIEW 帮助*。
3. 关闭前面板将把 VI 从 RT 终端的内存中移除。
4. 在 **项目浏览器** 窗口右键单击 RT 终端，并从快捷菜单中选择 **Disconnect**，关闭前面板连接。

调试独立的实时应用程序

使用 **调试应用程序或共享库** 对话框调试在 RT 终端上运行的独立的实时应用程序。



注

必须使用 **调试应用程序或共享库** 对话框启用独立实时应用程序的调试，以连接至该应用程序。关于启用调试的详细信息，请参考本文档的 [编辑实时程序生成规范](#) 部分。

按照下列步骤，调试 RT 终端上运行的独立实时应用程序。

1. 右键单击之前创建的独立实时应用程序 **Real-Time Application**，然后从快捷菜单中选择 **Run as startup**，在 RT 终端上运行该应用程序。
2. 在 **项目浏览器** 窗口中，选择 **操作 » 调试应用程序或共享库**，打开 **调试应用程序或共享库** 对话框。
3. 在 **机器名或 IP 地址** 文本框中输入 RT 终端的 IP 地址或名称，单击 **刷新** 按钮，查看 RT 终端内存中独立实时应用程序的列表。

4. 从**应用程序或共享库**下拉菜单中选择 `rtapp.rtexe`，单击 **Connect** 按钮，连接至独立实时应用程序。
LabVIEW 将打启动 VI 的程序框图以待调试。可在程序框图上使用探针、断点和其它 LabVIEW 调试技术以找到应用程序的错误所在。
5. 关闭启动 VI 的程序框图和**调试应用程序或共享库**对话框。
6. 取消独立应用程序在 RT 终端上电时启动的选项，然后部署程序生成规范，将改动应用至启动属性。
7. 重启 RT 终端。

使用 Real-Time Execution Trace 工具包

LabVIEW Real-Time 模块中包含 Real-Time Execution Trace 工具包的 30 天完整功能试用版。Real-Time Execution Trace 工具包用于分析实时应用的定时和执行情况。Real-Time Execution Trace 工具中的 VI 用于获取 RT 终端上 VI 的定时和执行数据，以及应用程序的线程事件。Real-Time Execution Trace 工具包可显示主机上的定时和事件数据，即跟踪会话。在 LabVIEW 中，选择**工具 »RT 模块 »Execution Trace Toolkit**，可打开 Real-Time Execution Trace 工具。关于使用 Real-Time Execution Trace 工具包分析实时应用程序的详细信息，请参考 *LabVIEW 帮助*。

总结

本教程介绍了下列内容。

安装 Real-Time 模块和配置 RT 终端

设置一个 LabVIEW Real-Time 模块系统必须完成三个步骤。

1. 在主机上安装软件—在主控计算机上安装 LabVIEW、Real-Time 模块和 NI 设备驱动程序。关于 LabVIEW 安装的详细信息，请参考 *LabVIEW 发行说明*。关于 Real-Time 模块和设备驱动程序安装的详细信息，请参考 *LabVIEW Real-Time 模块发行和升级说明*。
2. 配置 RT 终端设置—使用 MAX 配置 RT 终端的设置。
3. 在 RT 终端上安装软件—使用 MAX 中的 LabVIEW Real-Time 软件向导在 RT 终端上安装 Real-Time 模块和设备驱动程序。

使用 Real-Time Project Wizard 创建 RT 项目

装好 Real-Time 模块后，必须使用 LabVIEW 项目来创建实时应用程序并使程序在 RT 终端上运行。使用 Real-Time Project Wizard 创建实时项目，该项目定义 RT 终端并包含相关 VI 和其它应用程序文件。单击**启动窗口新建**部分的**实时项目**，或选择**工具 »RT 模块 »Project Wizard**，打开 Real-Time Project Wizard。

也可在 LabVIEW 中手动创建项目，并将 RT 终端和 VI 添加至项目。关于手动创建和编辑 Real-Time 模块项目的详细信息，见 *LabVIEW 帮助*中的 *Using LabVIEW Projects with the Real-Time Module* 部分。

使用项目浏览器窗口管理和配置项目

项目浏览器窗口用于统一管理 RT 终端、VI 和其它项目支持文件。

- **配置 RT 终端属性**—使用 RT 终端的**属性**对话框配置 RT 终端的项目设置。在项目浏览器窗口右键单击 RT 终端，并从快捷菜单中选择**属性**，打开 RT 终端的**属性**对话框。
- **配置和运行 VI**—在项目浏览器窗口中，部署 VI、库和共享变量至可运行或引用这些项的 RT 终端。右键单击 RT 终端下的 VI，从快捷菜单中选择**部署**，将 VI 部署至终端。单击前面板上的**运行**按钮，在终端上运行已部署 VI。

使用 LabVIEW 应用程序生成器创建独立实时应用程序

使用应用程序生成器创建一个程序生成规范，该规范定义如何生成一个独立的实时应用程序。程序生成规范定义独立的实时应用程序的 VI、支持文件和设置。

- **创建程序生成规范**—在项目浏览器中，右键单击 RT 终端下的**程序生成规范**，从快捷菜单中选择**新建 »Real-Time Application**，创建一个新的程序生成规范。也可使用该程序生成规范将独立实时应用程序定义为 RT 终端上电后自动启动。在项目浏览器窗口右键单击程序生成规范，从快捷菜单中选择**Set as startup**，将独立实时应用程序设置为 RT 终端上电后自动打开。
- **生成和运行独立实时应用程序**—右键单击程序生成规范，从快捷菜单中选择**生成**，生成一个独立的实时应用程序。在项目浏览器窗口右键单击程序生成规范，从快捷菜单中选择**Run as Startup**，即可在 RT 终端上运行该独立实时应用程序。

使用 NI 分布式系统管理器管理 RT 终端资源

NI 分布式系统管理器显示 RT 终端上运行 VI 的详细信息，并提供终端资源的动态显示，例如，可用内存量和 CPU 占用情况。在项目浏览器窗口中，选择**工具 » 分布式系统管理器**，打开系统管理器。

使用 RT 终端错误对话框查看 RT 终端的错误记录

RT 终端错误 (RT Target Errors) 对话框显示各个 RT 会话的 LabVIEW 错误、实时操作系统错误和故障记录。在项目浏览器窗口右键单击 RT 终端，并从快捷菜单中选择**工具 »View Error Log**，打开 **RT Target Errors** 对话框。

调试 RT 终端 VI 和独立应用程序

- **调试 RT 终端 VI**—打开至一个 RT 终端的前面板连接，使用 LabVIEW 调试工具调试终端上运行的 VI。在**项目浏览器**窗口右键单击 RT 终端，并从快捷菜单中选择 **Connect**，打开前面板连接。在**项目浏览器**窗口右键单击 RT 终端，并从快捷菜单中选择 **Disconnect**，关闭前面板连接。
- **调试独立的实时应用程序**—使用**调试应用程序或共享库**对话框调试在 RT 终端上运行的独立的实时应用程序。在**项目浏览器**窗口中，选择**操作 » 调试应用程序或共享库**，打开**调试应用程序或共享库**对话框。

参考资料

阅读本教程后，可参考下列资源获取关于 LabVIEW Real-Time 模块的更多信息。

Real-Time 模块基本概念

*LabVIEW 帮助*的 *Real-Time Module Concepts* 部分包含的是概念性内容，涵盖了创建实时应用时所需的编程技巧、应用程序架构和 Real-Time 模块功能。创建确定性的实时应用程序之前，请参考 Real-Time Module Concepts。

Real-Time 模块范例

在 LabVIEW 中选择**帮助 » 查找范例**，打开 NI 范例查找器，浏览或搜索范例 VI。也可通过 `labview\examples\real-time` 目录查看这些范例。Real-Time 模块范例中包含多线程通信、网络通信和共享内存访问等范例。

LabVIEW, National Instruments, NI, ni.com, National Instruments 公司标识, 以及鹰形标识均为 National Instruments Corporation 的商标。关于其它 National Instruments 商标, 请访问 ni.com/trademarks 参考 *Trademark Information*。此处提及的其它产品和公司名称为其各自公司的商标或商业名称。关于 National Instruments 产品和技术专利权, 见软件中的**帮助»专利信息**、光盘上的 patents.txt 文档, 或登录 ni.com/patents 查看 *National Instruments Patent Notice*。