



Design Real Systems, Fast

Что такое NI MyRIO

Учебный прибор разработчика NI MyRIO был создан чтобы студенты могли в течение одного семестра решать "настоящие" инженерные задачи. Он содержит двухъядерный программируемый процессор ARM Cortex-A9 с тактовой частотой 667 МГц. И кастомизируемую программируемую логическую интегральную схему (ПЛИС) Xilinx, которую студенты смогут использовать для начала разработки систем и быстрого решения встающих перед ними проблем разработчика, в компактном, простом и красивом форм-факторе. NI MyRIO содержит программируемый чип Zyng-7010, позволяющий В полную силу использовать возможности LabVIEW, как для приложений реального времени, так и для ПЛИС. Вместо долгих часов, затраченных на отладку кода или разработку пользовательского интерфейса, студенты смогут воспользоваться графическим подходом LabVIEW к программированию И сконцентрироваться разработке, без на



дополнительного давления необходимости изучить среду разработки.

NI MyRIO — реконфигурируемый инструмент, подходящий для многократного использования. С его помощью студенты могут освоить широкий круг инженерных концепций и даже полный цикл разработки. Возможности работы с ПЛИС и реальным временем, а так же встроенный Wi-Fi модуль позволяют запускать приложения удаленно и без подключения к компьютеру. Три разъема (2 порта расширения NI MyRIO (MXP) и один порт NI miniSystems (MSP), идентичный разъему NI myDAQ) передают и получают сигналы от датчиков и электрических схем, используемых студентами в их системах. NI MyRIO содержит в общей сложности 40 цифровых линий ввода/вывода с поддержкой SPI, PWN выхода, входного импульсного датчика, UART и I²C; восемь односторонних аналоговых входов; два дифференциальных аналоговых входа; четыре односторонних аналоговых выхода; и два общих аналоговых выхода, позволяющие подключать бесчисленное количество сенсоров, устройств и программируемых контроллеров системы. Вся необходимая функциональность встроена и предварительно настроена в базовом функционале ПЛИС, что устраняет необходимость в платах расширения или "щитах" для добавления нужных возможностей. В конечном счете, NI MyRIO позволит студентам решать инженерные задачи реального времени прямо сейчас — от радиоуправляемых машин до одиночных медицинских приборов.

Легкий в освоении, и обладающий полной функциональностью в базовой поставке NI MyRIO прост для запуска, а студенты легко могут определять его рабочий статус.

Полностью настроенный ПЛИС внедрен в устройство ещё на стадии производства, так что новички могут перейти к функциональным основам без программирования ПЛИС для нормального функционирования системы. Тем не менее мощь ПЛИС становится очевидной, когда студенты дойдут до работы с ПЛИС и настройки поведения устройства под нужды текущей задачи. С учетом гибких возможностей устройства, студенты могут использовать его в течение всего года, начиная с введения во встраиваемые устройства, закачивая курсами разработки в конце года.

Введение в LabVIEW.

Студенты и преподаватели должны иметь представление о среде LabVIEW до того как приступать к таким задачам, как приложение реального времени или настройка ПЛИС. Для приобретения таких знаний и создано это руководство. Оно предназначено в первую очередь для тех, кто не имеет опыта работы в среде LabVIEW. Тем не менее, оно не является заменой полноценного курса по LabVIEW—здесь не описывается каждое контекстное меню, кнопка или все функции LabVIEW. Если у вас уже есть опыт работы с LabVIEW, вы сможете программировать устройство NI MyRIO. Этот раздел описывает графическую природу и возможности языка программирования LabVIEW. Он также охватывает окно Getting Started и



браузер проектов (Project Explorer) и заканчивается небольшой программой в LabVIEW.

LabVIEW — это графическая среда программирования, где студенты могут быстро разрабатывать приложения и масштабировать их для разных устройств и операционных систем. Сильной стороной LabVIEW является возможность взаимодействия с тысячами приборов и инструментов с помощью сотен встроенных библиотек и виртуальных инструментов, которые помогают вам сократить время на разработку и быстро собирать, анализировать и представлять данные в нужном виде.

Приложения, созданные в LabVIEW имитируют интерфейс настоящих приборов, например мультиметров, осциллографов или генераторов сигналов, поэтому они называются виртуальными инструментами (VI). Каждое приложение LabVIEW имеет переднюю панель, блок-схему и соединительную часть. Передняя панель призвана изображать интерфейс реального прибора, моделируемого в LabVIEW. Программист может гибко настраивать отображение данных, с которыми работает виртуальный инструментах, которые позволяют подключать их к другим приборам. Таким образом, VI могут содержать другие VI, называемые суб-VI, соединенные между собой. Каждый из суб-VI в свою очередь тоже может содержать суб-VI. Такая структура соответствует вызову функций в текстовых языках программирования. И, наконец, блок-схема — это то место, где программист создает собственно код. В отличие от текстовых языков

программирования, таких как C, Java, C++ и Visual Basic, LabVIEW для создания приложений использует иконки вместо строк кода. Благодаря этому ключевому отличию контроль выполнения осуществляется набором правил для потока данных вместо последовательности. Провода, соединяющие VI и иконки на блок-схеме, определяют порядок выполнения кода.

Суммируя, LabVIEW это графический, работающий на потоке данных, ситуационный язык программирования, многоцелевой и мультиплатформенный. Помимо этого он обеспечивает гибкость объектно-ориентированного программирования, многопотоковость и параллельность. Виртуальные инструменты LabVIEW могут использоваться в задачах реального времени и ПЛИС.

Диалог LabVIEW Getting Started

Установите LabVIEW и любые заказанные вами расширения с DVD-диска, идущего в комплекте поставки NI MyRIO. После активации программного обеспечения с помощью Activation Wizard, оно готово к работе.

В Windows 7 вы можете запустить LabVIEW через меню "Пуск", либо отыскав нужную папку, либо воспользовавшись поиском по запросу LabVIEW.

После успешного запуска LabVIEW и загрузки всех необходимых ресурсов, открывается диалог Getting Started.

Если чекбокс "Show on lunch" отмечен, то появится и диалог Set Up and Explore. Для прибора NI MyRIO этот диалог скомпонован таким образом, чтобы отвечать нуждам студентов, работающих с NI MyRIO. С его помощью они смогут запустить Getting Started Wizard для NI MyRIO, почти так же, как это делает автозапуск при подключении устройства по USB, получить доступ к руководствам по началу работы, справочным материалам LabVIEW, и настроить Wi-Fi на устройстве NI MyRIO. Но пока что, закроем этот диалог.

С помощью диалога Getting Started, вы можете создать новые проекты и VI, или выбрать одну из множества опций продолжения работы над уже существующими проектами, или найти подходящий демонстрационный VI с помощью NI Example Finder. Вы можете открыть этот диалог когда угодно с помощью меню LabVIEW»Getting Started Window в любом VI. В нижней части окна доступно ещё несколько опций. Каждая из них ведет к полезной информации, обучающим материалам, упражнениям и другими средствами поддержки ваших изысканий с NI MyRIO.

Create Project	Open Existing
muBID Oustom EP64 Project	Show All
Blank Project	Custam FPGA 6.lvgraj
myBIO Project	02_Button Rising Edge Example.lvproj
Blank VI	Custom FPGA 5.lvproj
	Audio Generation lyproj
	Custom FPGA 4.lvproj
τ	Dustom FPGA Project 3 Iverci
Set Up and Explore Set up and learn how to use NI myRIOI See examples and ge	et inspired Get Support Get answers to your questions!
S LabVIEW News Announcing the 2012 LabVIEW Champion Inductees	

Диалог Getting Started

Частью философии NI MyRIO является создание моста между реальной проблемой при минимально возможном количестве промежуточных проблем. Избегайте опасностей "пустого VI" (или подавленности от вида чистой блок-схемы) находя с его помощью, шаблоны и примеры, или воспользуйтесь помощью интернет-форума или технической поддержки. А если дело доходит до простых систем или задач — многие пользователи или сотрудники NI уже создали что-то, решающее в точности нужную задачу или нечто очень близкое. Изобретать велосипед заново может быть тяжело, но благодаря мощности и гибкости LabVIEW и широкому кругу ученых и инженеров, работающих с ним, вы можете избежать этого.

Браузер проектов LabVIEW

Браузер проектов LabVIEW (Project Explorer) помогает вам контролировать связанные с приложением ресурсы. Они могут включать в себя несколько VI, изображения, текстовые документы, информация о конфигурации, сборке и внедрении. Структура проекта позволяет легко и быстро управлять ресурсами, вы можете распределять ресурсы по нескольким носителям.

Тем, кто уже знаком с другими интегрированными средами разработки (IDE), браузер проектов LabVIEW покажется знакомым, а принцип его использования по большому счету такой же. LabVIEW — это кросс-платформенный язык, с всеобъемлющей поддержкой всевозможных дополнений и библиотек. Вы можете написать большинство VI в то время, как закрепляете их за целевыми устройствами, и легко переназначить их на другие устройства NI, такие как NI MyRIO, поскольку они управляются операционной системой, которая тоже работает с LabVIEW и дополнительными библиотеками. Иерарзия

проекта в браузер проектов LabVIEW не зависит от файлов с исходным кодом на компьютере. Обычно, проект создается в определенной директории, и все сопутствующие файлы с кодом находятся в той же директории без какой-либо систематизации, даже если отображение в браузере проектов хорошо структурировано. Перемещение файла в браузере проектов не затронет сам файл на диске — только его отображение.

Подсказка Целевым может быть любое устройство, способное выполнить код VI.

Воспользуйтесь этой инструкцией для создания нового проекта:

- 1. В диалоге LabVIEW Getting Started, нажмите кнопку Create a New Project.
- 2. Откроется окно Create Project, предлагая несколько опций. Исследуйте все возможные опции либо создав проект с их помощью, либо воспользуйтесь ссылкой *More Information* для интересующего вас шаблона.
- 3. Выберите опцию Blank Project.
- 4. Нажмите кнопку **Finish**. Вы создали новый пустой проект. Откроется окно браузера проектов.

Чтобы сохранить новый проект:

- 1. В браузере проектов LabVIEW выберите пункт File»Save.
- 2. Откроется диалог сохранения, запрашивая имя проекта и директорию для его сохранения. Выберите подходящую директорию, и сохраните проект под именем, позволяющем идентифицировать проект и его назначение.

Теперь браузер проектов открыт и готов к заполнению ресурсами и исходным кодом. Меню браузера проектов содержит набор стандартных опций (File, Edit, View и т.д.) и несколько функций специфичных для LabVIEW. Браузер содержит две вкладки — Items и Files. Вкладка Items отображает сущности, используемые в проекте в иерархическом виде, а вкладка Files показывает файлы на диске, задействованные в проекте.



Браузер проектов LabVIEW.

На вкладке Items вы можете увидеть структуру проекта. Корень проекта, первая сущность в списке, показывает, над каким проектом вы сейчас работаете. Вторым уровнем находятся целевые устройства проекта. В пустом проекте целевым устройством по умолчанию является "мой компьютер". Следующим уровнем являются характеристики сборки целевого устройства. Они включают в себя конфигурации для исходников и других наборов инструментов и модулей, содержащихся в LabVIEW. Вы можете использовать их при внедрении приложений на встраиваемых системах. Хотя этот курс рассматривает простые VI, вам потребуется понимание работы с браузером проектов это потребуется при разработке более сложных проектов. Пока что же наш проект слишком пуст, и не служит никаким целям. Для более подробного ознакомления с программированием в LabVIEW. <u>Не закрывайте проект, чтобы использовать его в дальнейшей работе.</u>

Упражнение 1. Создание виртуального инструмента VI в LabVIEW в Windows

Чтобы познакомиться с графической средой разработки LabVIEW, создайте базовый код, иллюстрирующий некоторые свойства программирования потока данных. В этом упражнении мы создадим простой VI, конвертирующий температуру из шкалы Фаренгейта в шкалу Цельсия.

Для начала надо создать новый VI, с помощью меню File»New VI в браузере проектов LabVIEW.

🛃 Untitled	Project	5.lvproj -	Project Ex	plorer		X
File Edit	View	Project	Operate	Tools	Window	Help
New VI			Ctrl+N		🖪 - f	
New					<u> </u>	
Open			Ctrl+O			
Close			Ctrl+W			
Close Al	l L felsis D					
Close Al	i (this P	roject)				
Save			Ctrl+S			
Save As.						
Save All			Ctrl+Sh	ift+S		
Save All	(this Pr	oject)				
Save for	Previou	is Version				
Kevert						
Create P	roject					
Open Pr	oject					
Save Pro	ject					
Close PI	oject					
Page Se	tup					
Print						
Print Wi	ndow		Ctrl+P			
VI Prope	erties		Ctrl+I			
Recent F	rojects			×		
Recent F	iles			+		
Exit			Ctrl+Q			
					-	
						I 1

Создание нового пустого VI.

Эта операция создаст пустой VI в целевом устройстве Му Computer в проекте, который вы только что создали. Он задержит два окна: переднюю панель и блок-схему.

United Effort Panel on Linksted Project Siliping/myRD-2900	D 3	United L Block Dispan on United Project Shyre(Jmy900-1900		
Ne Edit View Project Operate Tools Wiedow Help	ETTE			
O 🕀 🐷 B. 13pt Application Fort 💌 🚛* 🔂* 🔕*	- Stack - P Hill	🎦 🛛 🔆 🖶 🖬 😨 💱 No 🦉 🕫 Libet Application Fent * 🖬 * 👸 * 👒	🖲 Search 🔍	
		*		
rited Decard T. Agency Weidlinh (2000) a		Links at Propert 1 Approx/900-1008 4		

Передняя панель и блок-схема.

Сохраните VI, выбрав команду **File**»**Save** в любом из двух окон. В диалоге сохранения задайте для VI имя со смыслом. Сохраняйте VI в той же директории, что и проект, к в котором они используются. Вы можете сохранять VI, которые планируете использовать в нескольких проектах, в других директориях. Браузер проектов будет отслеживать их и предупредит об отсутствующем или неправильно расположенном VI.

Теперь, когда вы сохранили VI, вы можете создать код, конвертирующий температуру. Для этого:

- 1. Этот код выполняется непрерывно, и как в текстовых языках программирования, LabVIEW использует зацикливание для обеспечения непрерывной работы кода. Вы можете остановить выполнение либо с помощью кнопки на передней панели, либо воспользовавшись традиционной логикой, чтобы задать точку, где выполнение должно быть остановлено. В этом упражнении используйте цикл While, для постоянного отслеживания температуры по шкале Фаренгейта и непрерывного преобразования в шкалу Цельсия и кнопку на передней панели для остановки.
 - а. Переключитесь на окно блок-схемы, либо стандартными средствами Windows, либо воспользовавшись сочетанием клавиш Ctrl+E (это сочетание позволяет быстро переключаться между передней панелью и блок-схемой).



Подсказка Использование клавиатурных сочетаний позволяет ускорить работу в среде LabVIEW. Кроме того, вы будете выглядеть как настоящий LabVIEW-волшебник!

b. Щелкните правой кнопкой по блок-схеме, чтобы открыть палитру Functions. Она содержит все узлы и VI, используемые при программировании в LabVIEW. Вы можете настроить её отображение, но по умолчанию доступны все дополнения и VI. Ознакомьтесь с расположением всех подпанелей. Выберите пункт Programming»Structures»While Loop и выберите цикл While.



с. Нарисуйте петлю в блок-схеме. Кликните левой кнопкой мыши, растяните окно на нужную площадь и кликните ещё раз для завершения. Вы можете изменять размеры петли, наведя курсор на границу петли, кликнув левой кнопкой мыши, схватив мышью один из восьми синих маркеров и перетянув его на нужное место для изменений размера.

Untitled 1 Block Diagram on Untitled Project 5.lvproj/myRIO-1900 *		
File Edit View Project Operate Tools Window Help		
🕸 🕾 📄 🔢 🤤 🛱 🧤 🗃 🗗 15pt Application Font 🔻 🚛 🙃 🕇 🦗	Search	
		=
<u> </u>		<u> </u>
Untitled Project 5.lvproj/myRIO-1900 <		► a

Блок-схема с циклом While.

d. Переключитесь на окно передней панели и щелкните правой кнопкой мыши для открытия палитры Controls. Она содержит все необходимые элементы для создания внешнего интерфейса и элементов управления, имитирующих реальный инструмент. Выберите пункт Modern»Boolean»Stop Button.

📽 🐵 😻 🚺 15pt Application Font		• Search	् श्व
- A Controls	Q Search		
Modern	• • • • • • • • • • • • • • • • • • •		
Numeric Boolean Numeric Boolean Numeric Boolean Array, Matrix List, Table & Ring & Enum Containers Paring & Enum Containers Variant & Cl Decorations System Classic Express Control Design & Simulation Addons	Control Contro		
Select a Controls			
Robotics	•		
Signal Processing	Image: A state of the state		
Silver	Image: A second seco		
Vision	•		

Выбор кнопки Stop.

е. Выберите кнопку Stop с помощью левого щелчка мышью и перетащите её на нужное место на передней панели. Вы можете отслеживать положение контроллера, который собираетесь расположить по его контуру, отображаемому на передней панели. Поместите кнопку в любом месте на передней панели с помощью левого щелчка.



Передняя панель с кнопкой Stop.

- f. Теперь переключитесь обратно на блок-схему и найдите на ней новую кнопку Stop. Если она находится вне цикла Wihle, перетащите её туда с помощью мыши. При наведение курсора на кнопку подсветится её выходной разъем (зелёный кружочек с правой стороны кнопки). Наведите курсор на кружочек, и он пример вид мотка провода.
- g. Соедините кнопку Stop и иконку состояния выхода из цикла в правом нижнем углу поля цикла, кликнув по разъему кнопки и подтянув провод к разъему состояния выхода из цикла (тень провода будет отображаться при протягивании). Для закрепления, щелкните по разъему состояния выхода из цикла. Между двумя элементами должен появиться зеленый проводок. Это означает два правильно соединенных компонента логического (boolean) типа. Другие типы данных имеют свои цвета.

🔁 U	ntitled (2 Block	: Diagram	on Untitle	d Project 1,	/My Con	nputer *								x
File	Edit	View ক্রি	Project	Operate	Tools W	Vindow	Help plication Fr			- 63	- 1 -1	Search		0	
	ν	G		(<u>*o+</u> *		12000.001	pheadonn		*10 - 1-00			Jearen			<u> </u>
	1	-				_				_		_			
															=
	l	i									stop 🎞	.			
															-
Until	led Proj	ect1/N	ly Compu	ter ∢		_	_	_	_		_	_	_		<u>ار</u> ا

Цикл While с подключенным условием выхода.

- 2. Теперь создадим контроллер для значений по Фаренгейту и индикатор для значений по Цельсию.
 - а. Переключитесь на переднюю панель, и щелкните правой кнопкой мыши для открытия палитры Controls. Выберите пункт Modern»Numeric»Numeric Control.
 - b. Перетащите контроллер на переднюю панель.
 - с. Обратите внимание на маркировку Numeric Control у контроллера. Не очень информативное имя. Переименуйте контроллер, двойным щелком по нему и напечатайте новое имя Fahrenheit.
 - d. На передней панели, щелкните правой кнопкой мыши для открытия палитры Controls. Выберите пункт **Modern»Numeric»Thermometer**. Поместите термометр на переднюю панель.
 - e. Как и для контроллера Fahrenheit, переименуйте индикатор в Celcius.
 - f. Добавим функциональности, разместив на передней панели виртуальный светодиод, загорающийся когда температура падает ниже нуля по Цельсию. Светодиод можно найти в палитре Controls, пункт Modern»Boolean»Round LED.
 - g. Поместите светодиод на переднюю панель.
 - h. Переименуйте светодиод "Below Freezing?"

擾 Un	titled 2 Front Pan	el on Untitled Project 1	/My Compute	er *				
File	ile Edit View Project Operate Tools Window Help							
	🗘 🕹 🔘 ।	15pt Application Fo	int I∎ 🚛 🖛	′िकर ≝र 🔅र	▶ Search	<u>?</u> HIH [∞] 2		
	stop	Celsius						
	STOP	100 -		Below Freezing?				
		80 -		0				
	Cabranhait	50 <u>-</u>						
	Anemet	40 -						
1	50	20 -						
		0=-						
Untitle	ed Project 1/My C	omputer (III III III III III	••••••••••••••••••••••••••••••••••••••		

Передняя панель с контроллером и индикаторами.

- 3. Теперь, когда контроллеры и индикаторы расположены на передней панели, нужно добавить в проект необходимый код и логику для конвертации температуру из шкалы Фаренгейта в шкалу Цельсия.
 - a. Переключитесь на блок-схему и найдите иконки для контроллеров и индикаторов Fahrenheit, Celsius и Below Freezing. Перетащите иконку Fahrenheit в левую часть цикла While, а иконки Celsius и Below Freezing в правую. Оставьте между иконками достаточно места для написания кода.
 - b. Поместите вычитающий модуль на блок-схеме. Щелкните правой кнопкой для открытия палитры Functions и выберите пункт **Programming**»**Numeric**»**Subtract**. Поместите модуль сразу справа от иконки Fahrenheit.
 - с. Соедините иконку Fahrenheit и верхний разъем вычитающего модуля.
 - d. Щелкните правой кнопкой по нижнему разъему вычитающего модуля и добавьте константу, выбрав пункт Create»Constant.
 - е. Задайте для константы значение 32.
 - f. Поместите умножающий модуль на блок-схеме. Щелкните правой кнопкой для открытия палитры Functions и выберите пункт **Programming»Numeric»Multiply**. Поместите модуль сразу справа от вычитающего.
 - g. Соедините выходной разъем вычитающего модуля и верхний разъем умножающего.

- h. Поместите модуль для деления на блок-схеме. Щелкните правой кнопкой для открытия палитры Functions и выберите пункт **Programming**»**Numeric**»**Divide**. Поместите модуль снизу от вычитающего узла.
- i. Щелкните правой кнопкой по верхнему разъему модуля для деления и добавьте константу, выбрав пункт **Create**»**Constant**. Задайте ей значение 5.
- j. Щелкните правой кнопкой по нижнему разъему модуля для деления и добавьте константу, выбрав пункт **Create**»**Constant**. Задайте ей значение 9.
- k. Соедините выходной разъем модуля для деления с нижним разъемом умножающего модуля.
- 1. Соедините выходной разъем умножающего модуля с входным разъемом индикатора Celsius.
- m. Добавьте сравнительный модуль "меньше нуля?" на блок-схему щелкнув правой кнопкой и выбрав в палитре **Programming**»Comparison»Less Than Zero? Поместите модуль справа от умножающего.
- n. Соедините выходной разъем умножающего модуля с модулем "меньше нуля?"
- о. Соедините выходной разъем модуля "меньше нуля?" с индикатором Below Freezing?
- 4. Теперь, когда создан весь необходимый код, запустите VI. Убедитесь, что все соединения выполнены верно, сравнив свой VI с рисунком ниже и проверив, активна ли кнопка Run.
 - а. Переключитесь на переднюю панель (можете воспользоваться сочетанием клавиш Ctrl+E). Убедитесь, что все контроллеры и индикаторы присутствуют на экране.
 - b. Щелкните кнопку Run на приборной панели для запуска программы.
 - с. По умолчанию термометр показывает значения только в промежутке от 0 до 100. Вы можете изменить диапазон, дважды щелкнув по шкале и задав нужные значения. В этом упражнении, вводите значения температуры по шкале Фаренгейта и наблюдайте их преобразования на шкале Цельсия.
 - d. По окончании, прервите выполнение с помощью кнопки Stop.
- 5. Сохраните и закройте VI и проект преобразования температуры.

Выполнив это упражнение, вы освоитесь с графическим подходом к программированию. Кросс-платформенная модель LabVIEW делает разработку приложений для многих поддерживаемых устройств. В следующих упражнениях мы будем использовать ту же среду для создания программ для устройства NI MyRIO, с такой же структурой, контроллерами, индикаторами и функциональностью как в примере, который мы только что рассмотрели. Тем не менее, для начала вам надо установить и сконфигурировать устройство для работы с компьютером на котором ведется разработка. Инструментарий, поставляемы с NI MyRIO помогает сделать этот процесс легким и простым. Следующая глава описывает настройку устройства и мастер Getting Started для устройства NI MyRIO.

VI из упражнения 1:

Exercise 1.vi Front Panel		1 23						
File Edit View Project Operate Tools Window Help								
\$ & ■ Ⅱ	15pt Application Font 🛛 🚛 🐨 🕮 🖓 🕞 🕒 Search 🔍 🤶 🗄	IH 2						
Stop Button	Below Freezing?							
Stop								
Fahrenheit	Celsius							
	100 - 0	======						
	80							
	40-							
	20-							
	0 ²							
•	III	►						



Настройка оборудования: подключения устройства NI MyRIO

Одна из целей разработки NI MyRIO — упростить настройку оборудования. Для достижения этой цели программное обеспечение NI MyRIO имеет собственную утилиту для подключения и конфигурирования, отличную от браузера измерений и автоматизации NI (NI MAX). Вы по-прежнему можете использовать MAX для установки и настройки оборудования и программного обеспечения, если эта среда кажется вам более комфортной. Устройство NI MyRIO имеет приложения USB мониторинга, которое запускается, когда вы подключаете устройство к компьютеру. В ближайших разделе мы изучим как использовать NI MyRIO USB монитор и мастер Getting Started.

NI MyRIO USB монитор

Убедитесь, что вы запитываете NI MyRIO с помощью кабеля, идущего в комплекте поставки. Подключите к устройству кабель USB тип В. Другой конец кабеля подключите к USB-порту компьютера.

Если устройство включено, то без запуска LabVIEW или NI MAX, операционная система распознает устройство NI MyRIO, и установит необходимые драйверы. По завершении установки, в операционной системе Windows откроется окно NI MyRIO USB монитора, показанное на рисунке ниже.



Вы можете выбрать одну из четырех доступных опций:

1. Запустить мастер Getting Started

С его помощью вы сможете быстро узнать функциональный статус устройства NI MyRIO. Он проверяет подключенные устройства NI MyRIO, подключается к выбранному устройству, проверяет актуальность программного обеспечения, в случае необходимости предлагает обновить программного обеспечение, может предложить переименовать устройство, и затем отображает экран, аналогичный передней панели NI MyRIO, который вы можете использовать для наблюдения за функцией акселерометра, включать и выключать встроенные светодиоды и тестировать кнопку с пользовательской настройкой функциональности.

Последний экран мастера имеет две опции:

Запустить мой первый проект (Start my first project now)

Эта опция запускает веб-инструктаж, аналогичный упражнению 2 в этом руководстве.

Запустить LabVIEW (Go Straight to LabVIEW)

Эта опция запускает диалог LabVIEW Getting Started.

2. Перейти к LabVIEW (Go to LabVIEW)

Эта опция запускает диалог LabVIEW Getting Started.

3. Настроить NI MyRIO (Configure NI MyRIO)

Эта опция открывает настроечную утилиту NI MyRIO с веб-интерфейсом.

4. Ничего не делать (Do nothing)

Если LabVIEW уже открыт и проект сконфигурирован, с NI MyRIO в качестве целевого устройства, вы можете использовать эту опцию чтобы закрыть NI MyRIO USB монитор, когда модуль был отключен и вновь подключен к компьютеру.

Запуск мастера Getting Started

В окне USB монитора NI MyRIO, выберите Launch the Getting Started Wizard. Выберите Next и мастер подключится к устройству NI MyRIO, проверит его программное обеспечение и предложит переименовать устройство. Если модуль NI MyRIO не имеет установленного программного обеспечения, мастер автоматически установит последнюю версию ПО. После этого откроется окно диагностики. В нем вы можете просмотреть показания встроенного трехосного акселерометра, проверить назначение кнопки с пользовательской настройкой функциональности и четырех встроенных светодиодов.



Теперь, когда вы установили и настроили устройство NI MyRIO, вы можете создавать VI реального времени, и выполнять их в процессоре (аналогично с Windows VI), а так же ПЛИС VI, для знакомства с возможностями параллельной обработки. Затем, создайте обычный проект, и исследуйте функциональность базовой архитектуры.

Выполнение кода реального времени на устройстве NI MyRIO

Теперь запустите проект NI MyRIO из шаблонов, входящих в LabVIEW 2013 для NI MyRIO. После создания проекта, откройте файл "Main.vi" и внимательно изучите его. затем выполните упражнение по его функциональности.

Код реального времени выполняется на процессоре, встроенном в NI MyRIO. Он может принимать и отправлять данные через ПЛИС с помощью его узлов входа/выхода, входов/выходов с прямым доступом к памяти (DMA) и экспресс-VI, которые используют "сущность" ПЛИС. Входы и выходы разъемов NI MyRIO и порт NIminiSystems взаимодействую с процессором через ПЛИС. Мы подробно рассмотрим ПЛИС позднее, пока что запомните, что NI MyRIO поставляется с модулем ПЛИС, который по умолчанию настроен на прием и передачу данных между разъемами и встроемнными устройствами (кнопки, акселерометр) и процессором, выполняющим код реального времени.

Теперь, когда вы подключили и настроили устройство NI MyRIO, вы можете создать проект для него и начать разработку кода. Использование ПЛИС по умолчанию — это быстрый способ начать и создать простой автономный код или даже код проверки концепции для более сложный проектов, требующих использование ПЛИС. Для кода, требующего кастомизации способа, которым NI MyRIO управляет входом/выходом, требуется собственный ПЛИС-проект, чтобы обеспечить наилучшую производительность кода. Повторимся, что для простых проектов пользовательская настройка ПЛИС не требуется, так что в этой главе мы рассматриваем стандартный ПЛИС-проект.

Упражнение 2: Исследование виртуального инструмента Main.vi

Следуйте этой инструкции, чтобы начать NI MyRIO проект из шаблона:

1. В диалоге LabVIEW Getting Started, выберите кнопку Create Project.



Создание проекта

2. В левой части диалога "Create Project", в дереве шаблонов выберите пункт myRIO.

🕼 Create Project	
Choose a starting point for the proj All Templates Desktop myKID Robotics Sample Projects CompoctIUO Desktop myKID	ect:
myNU Real-Time	Facilitates customizing the myRIO FPGA personality. This template uses code written with the LabVIEW FPGA Module. More Information
Additional Search	
Keyword	Finish Cancel Help

Шаблоны проектов NI MyRIO

- 3. В правой части доступны три опции: стандартный пустой проект (Blank Project), проект myRIO (myRIO Project) и пользовательский myRIO ПЛИС проект (Custom myRIO FPGA Project). Стандартный пустой проект мы рассматривали в упражнении 1.
 - а. Используйте шаблон проекта myRIO для создания проекта с стандартной работой ПЛИС. Он используется в проектах, не требующих расширенной функциональности и дополнительной настройки ПЛИС.
 - b. Пользовательский ПЛИС проект используется для самостоятельной настройки ПЛИС в устройстве NI MyRIO. Например, объединение всех трех разъемов дает восемь цифровых линий ШИМ-входа/выхода. Но, если вам нужно подключить большей таких устройств или сигналов, вы можете настроить ПЛИС таким образом, чтобы поддержку ШИМ имело большее количество линий цифрового входа/выхода. Так же можно поступить с другими протоколами взаимодействия.
- 4. Выберите пункт myRIO Project и нажмите кнопку Next.
- 5. Задайте проекту имя со смыслом, позволяющее идентифицировать его в дальнейшем, и сохраните его в подходящей директории. Убедитесь, что выбрана селективная кнопка **Plugged into USB** и что правильно озаглавленное устройство NI MyRIO отображается в листе справа.

Treate Project	
Configure the new project:	
Project Name	
Untitled Project 5	
Project Root	
C:\Users\mbarrett.AMER\Documents\Sandbox\Untitled Project 5	
File Name Prefix (Optional)	
Modify VI Icons + = = Edit Overlay	
Select Target O Plugged into USB Margaret (172.22.11.2)	
© Connected over WiFi	
Generic Target Specific IP Address or Hostname T	See.
	*
	-
Back Finish Cancel	Help
Curca	

Настройка LabVIEW проекта

6. Если все настроено верно, нажмите кнопку Finish.

Этот шаблон задает NI MyRIO в качестве целевого устройства для проекта LabVIEW. Следует помнить очень важную вещь в концепции целевого устройства — если NI MyRIO выбрано в качестве целевого устройства для VI, то код VI будет выполнятся на нём, даже если передняя панель VI отображается на компьютере. Такой режим работы называется режим интерактивной передней панели (interactive front panel mode) и предназначет только для разработки и отладки. В конечном приложении любые контроллеры и индикаторы с передней панели VI будут недоступны. В конечном итоге, код развертывается на самом устройстве и работает даже без подключения к компьютеру через USB. Вы можете использовать сетевые совместные переменные или какую-либо форму сетевого потока в VI на компьютере для получения и обработки данных с устройства NI MyRIO. Теперь возможности становятся действительно впечатляющими. Даже без перенастройки ПЛИС вы можете собирать данные и принимать решения по управлению прямо в процессоре реального времени устройства NI MyRIO. Пока что, компьютер, на котором ведется разработка VI будем использовать для сохранения и/или анализа данных и даже для отправки управляющих воздействий высокого уровня на устройство NI MyRIO.

В этом упражнении, NI MyRIO привязан к компьютеру, но вы все равно можете захотеть выполнить полученный код на его процессоре. Это значит, вы сможете взаимодействовать с передней панелью VI, выполняющегося на процессоре реального времени, в то время

как движок NI забоится о сетевой передаче данных. Для некоторых задач тестирования и настройки режим интерактивной передней панели вполне адекватен и устраняет необходимость установки более сложного обмена данным между хостом и целевым устройством реального времени.

В браузере проектов LabVIEW теперь отображаются два целевых устройства — "Мой компьютер" и "myRIO-1900 (xxx.xx.xx)". Целевое устройство NI MyRIO уже содержит в проекте VI под названием "Main.vi". Этот VI содержит код для того, чтобы облегчить вам начало работы.



Иерархия проекта LabVIEW

- 1. Двойным щелчком откройте Main.vi из браузера проектов LabVIEW.
- 2. Откроется передняя панель VI. Она должна содержать синусоидальный график и кнопку стоп.
- 3. Перед запуском VI изучите блок-схему. Переключитесь на неё с помощью сочетания клавиш Ctrl+E.
- 4. Структура, окружающая цикл While и имеющая вид трех кадров с фотопленки называется структурой последовательности (Sequence Structure).



Есть два варианта такой структуры — плоская и сферическая. Структура в Main.vi — плоская. Структура последовательности заводит выполняемый код в последовательность. Весь код из каждого кадра (слева направо) должен быть выполнен перед тем, как сможет начаться исполнение следующего кадра. Вы можете использовать *туннели* (tunnels) для передачи данных между кадрами. Использование последовательной структуры в таком ключе — это простейший способ гарантировать выполнение кода нужным образом, хотя вы можете достичь этого используя технику программирования потока данных, без захода на территорию структур последовательности.

Подсказка Тоннели — это маленькие квадратные точки, которые видны на краях циклов, кадров и структур. Вы можете автоматически создавать их с помощью провода, соединяя два разъема, отделенных друг от друга вышеуказанными структурами, или соединить их вручную, кликнув по границе структуры. Туннели передают данные из структуры или цикла только тогда, когда выполнен весь код в структуре.

- Кадр "initialize" будет выполнен первым. Единственное действие, выполняемое в нем — создание кластера для константы ошибки и переход к следующему кадру через туннель.
- b. Кадр "acquire and process data" получает кластер ошибки, и начинает выполнение цикла While, названного "Main Loop". В этом цикле экспресс-VI (синий VI, названный "Accelerometer") используется для получения данных с встроенного акселерометра. Затем данные от осей х, у и z передается через выходные разъемы по оранжевым проводам (тип данных с двойной точностью) и соединяются в кластер, для подачи на индикатор, соответствующий синусоидальному графику на передней панели.

Обратите внимание на VI "wait(ms)" в правом верхнем углу.

Он заставляет код цикла выполняться каждые 10 мсек, если код выполнен менее чем за 10 мсек; в противном случае цикл выполняется настолько

10

быстро, насколько это возможно, без дополнительных 10 мсек ожидания. По существу это дает цикл с частотой 100 Гц (опять-таки, если код в цикле способен выполняться настолько быстро). Как только график получит 100 измерений с акселерометра, вы целиком увидите секунду наблюдений. Как и в большинстве структур с программным управлением временем, неизбежны некоторые дрожания. Для задач, требующих более точного отсчета времени, вы можете улучшить его в LabVIEW с помощью инструментов Timed Loops и программирования ПЛИС.

Вы можете выбрать два условия остановки основного цикла: кнопка Stop, которую вы можете нажать в любое время, чтобы прервать цикл While и продолжить выполнение последовательности, или назначить выход из цикла по срабатыванию ошибки, который можно создать подключив кластер ошибки из экспресс-VI "Accelerometer" к модулю ИЛИ с кнопкой стоп. Срабатывание приведет к остановке цикла While и продолжит выполнение последовательности при возникновении любой ошибки в VI, подключенном к модулю. Вы можете комбинировать ошибки от нескольких VI, чтобы создавать более безопасные протоколы выполнения, что позволяет избежать повреждений устройств, подключенных к системе из-за ошибок в выполнении кода или некорректной информации от датчиков.

- с. Когда цикл While заканчивает выполнение, создается ошибка и передается сквозь структуру в финальный кадр. Этот кадр используется для закрытия всех переменных и ссылок на устройство NI myRIO перед тем, как программа закроется. Когда вы добавите больше функциональности, вы все равно можете использовать этот кадр для закрытия линий ввода/вывода и сохранения или удаления данных.
- 5. Теперь вы изучили структуру кода, и как он ведет себя при исполнении. Переключитесь обратно на переднюю панель, нажав Ctrl+E. Кликните кнопку запуска или нажмите Ctrl+R для запуска этого VI на устройстве NI myRIO.
- 6. После того, как вы успешно внедрили VI на NI myRIO, график начинает отображать данные с акселерометра. Встряхните NI myRIO чтобы отследить изменение показаний акселерометра в реальном времени.
- 7. Нажмите кнопку стоп, чтобы позволить VI выйти из цикла While и завершить выполнение структуры.

8. Оставьте VI и проект открытыми.

Настройка проекта для NI myRIO аналогичная настройке любого другого проекта с целевым устройством реального времени. Вы можете переделать любой проект под реальное время, добавив устройство реального времени в качестве целевого. Вы также можете создавать проекты реального времени из шаблонов по описанному в последней главе процессу. Разбор стандартного виртуального инструмента "Main.vi", его архитектура и функциональности теперь не должны вызывать проблем. Вы сможете расширить функциональность существующего процесса (цикла While) или даже добавить

другие циклы для параллельной обработки данных. Тем не менее, процессор реального времени может обрабатывать параллельно только столько процессов, сколько у него имеется ядер. При наличии большего количества циклов, процессор будет отрабатывать параллельность аналогично другим языкам — с использованием потоков и переключением между задачами при выполнении.

Упражнение 3. Создание кода реального времени для исполнения на устройстве NI myRIO

В этом упражнении мы, с помощью шаблона проекта NI myRIO, напишем в LabVIEW виртуальный инструмент для исследования встроенных функций устройства. В нашем случае, приложение будет использовать кнопку и встроенные светодиоды.

В проекте, оставшемся открытым в предыдущем упражнении, выполните нижеприведенные инструкции для создания VI реального времени. Если вы закрыли проект из прошлого упражнения, повторите его для создания проекта NI myRIO.

- 1. Откройте "Main.vi" под целевым устройством myRIO-1900 (xxx.xx.xx.x) в браузере проектов LabVIEW.
- 2. Откройте блок-схему.
- 3. Увеличьте центральный кадр последовательности и цикл While, расположенный внутри.
 - а. Поместите курсор мыши на нижнюю грань последовательности, для появления синего узла изменения размера. Схватите его и растяните последовательность, обеспечив больше места для иконок.
 - b. Аналогичным образом увеличьте центральный кадр последовательности, растянув его вправо.
 - с. Увеличьте площадь цикла While, потянув за правый нижний угол.
- 4. Создайте триггер нарастающего фронта для детектирования нажатия на кнопку устройства NI myRIO (игнорируя выход, при удержании клавиши).
 - а. Поместите экспресс-VI Button Express VI на блок-схему из палитры NI myRIO.
 - i. Щелкните правой кнопкой мыши и выберите пункт **myRIO»Onboard»Button**. Поместите экспресс-VI внутри цикла While прямо под VI акселерометра.
 - іі. Нажмите **ОК** в диалоге конфигурации оставив стандартные настройки экспресс-VI. Он обрабатывает ссылку на физическое оборудование через стандартные настройки ПЛИС на борту NI myRIO и отслеживает текущее состояние кнопки. Поскольку экспресс-vi находится в теле цикла While, его код выполняется на каждой итерации цикла. Тем не менее, его в его коде имеется функция "smart open", которая позволяет избежать открытия соединения на каждой итерации. Таким образом, после первой итерации кода, экспресс-vi выполняется быстрее, поскольку ему требуется только считывать данные с кнопки.



- b. Соедините выход экспресс-vi кнопки с правой гранью цикла While.
- с. Щелкните правой кнопкой по созданному тоннелю переменной логического типа (Boolean) и выберите **Replace with Shift Register**. Это изменит иконку тоннеля и создаст новую на том же месте с левой стороны цикла While. Два этих тоннеля создают регистр сдвига. Правая часть регистра хранит данные с текущей итерации, в то время как левая с предыдущей итерации. С помощью этого регистра, вы можете сравнить текущее значение переменной кнопки с прошлым значением. (на первой итерации предыдущее значини по умолчанию FALSE).



- d. Поместите модуль Exclusive OR (исключающее ИЛИ) на блок-схему.
 - i. Щелкните правой кнопкой и выберите пункт **Programming»Boolean»Exclusive Or**. Поместите модуль внизу справа от экспресс-vi кнопки.
 - іі. Подключите разъем Value экспресс-vi кнопки с верхним разъемом модуля исключающего ИЛИ.
 - ііі. Подключите регистр сдвига, хранящий значение кнопки с нижним разъемом модуля исключающего ИЛИ.



- е. Разместите на блок-схеме модуль And (И).
 - i. Щелкните правой кнопкой и выберите пункт **Programming»Boolean»And**. Поместите модуль справа от модуля исключающего ИЛИ.
 - іі. Подключите выход экспресс-vi кнопки с верхним разъемом модуля И.
 - ііі. Подключите выход модуля исключающего ИЛИ с нижним разъемом модуля И.



- f. Используя сочетание клавиш *Ctrl+H*, откройте контекстную справку для изучения функционирования модулей **Exclusive OR** и **And**. Теперь, когда создана логика для детектирования нажатия на кнопку устройства NI myRIO, вы можете использовать её в функциональности.
- 5. Создайте цикл мигания светодиодов встроенных светодиодов при нажатии кнопки.
 - а. Поместите константу **True Constant** на блок-схему.
 - i. Щелкните правой кнопкой и выберите пункт **Programming»Boolean»True Constant**. Поместите константу в кадр *Initialize frame*.
 - іі. Подключите константу к левой грани цикла While (она автоматически передается через тоннель в следующий кадр).



- b. Поместите константу False Constant на блок-схему.
 - i. Щелкните правой кнопкой и выберите пункт **Programming»Boolean»False Constant**. Поместите три константы в кадр *Initialize frame* первую прямо под константой **True**, остальные друг под другом.
 - іі. Подключите константы к левой грани цикла While.
- с. Замените тоннели констант регистрами сдвига.
 - i. Щелкните правой кнопкой по каждому тоннелю логических данных только что созданных слева от цикла и выберите Replace with Shift Register.



- d. Каждая из четырех логических констант только что помещенных на блоксхеме используется для отслеживания состояния четырех светодиодов на борту устройства NI myRIO. Этот код — очень простая и прямолинейная реализация сдвига горения светодиода на последующий светодиод. Другой метод использует счетчик и модульную функцию для вычисления остатка от деления отсчета и сравнения к конкретному светодиоду (0, 1, 2,3). Такой подход позволяет использовать только один регистр сдвига, но требует размещения большего количества функций на блок-схеме.
- e. Поместите структуру case structure (выбирающая структура) на блок-схему.
 - i. Щелкните правой кнопкой мыши и выберите пункт **Programming**»Structure»Case Structure. Поместите структуру в цикл While справа внизу от триггера (модулей Exclusive OR и AND).



іі. Соедините выход модуля AND и селектор на структуре.



- ііі. Переключите структуру на состояние False с помощью стрелочек возле индикатора.
- iv. Пустите регистры сдвига через структуру, а затем напрямую к правой грани цикла While. Имейте в виду, что если они изначально не были

подключены к левой грани структуры, автоматический инструмент соединения скорее пустит провода в обход структуры, чем через неё.



- v. Переключите структуру в состояние True.
- vi. Подключите тоннель для каждой логической переменной к тоннелю, проходящему ниже, соединив последний тоннель слева с первым тоннелем справа, и т.д.



- f. Логика работы такова, что когда детектируется восходящий фронт сигнала с кнопки (другими словами — когда кто-то нажимает кнопку), состояние структуры true переключает светодиод на следующий. Загорается следующий светодиод, а предыдущий гаснет. Такое происходит при каждом нажатии на кнопку. А если нажать кнопку при горящем четвертом светодиоде, он погаснет, а загорится первый светодиод. Тем не менее, этот код не будет работать, если не объяснить NI myRIO что логические переменные надо интерпретировать как состояния светодиодов.
- 6. Привяжите созданную логику к состоянию светодиодов.
 - а. Поместите на блок-схему экспресс-vi LED.
 - i. Щелкните правой кнопкой мыши и выберите пункт **myRIO»Onboard»LED**. Поместите экспресс-vi LED в правом нижнем углу цикла While. Нажмите **OK** в диалоге настройки

экспресс-vi для сохранения стандартных настроек. По умолчанию этот экспресс-vi позволяет управлять всеми четырьмя светодиодами.

іі. Подключите каждую из логических переменных созданных ранее в входам LED0, LED1, LED2 и LED3 экспресс-vi.



- 7. Запустите VI нажав кнопку запуска на приборной панели. Проверьте работу кода, нажимая кнопку на NI myRIO и наблюдая за работой светодиодов.
- 8. Чтобы убедиться, что код выполняется на устройстве NI myRIO, а не на компьютере, щелкните правой кнопкой на NI myRIO в проекте LabVIEW и выберите пункт **Disconnect**.
- 9. Извлеките кабель USB из устройства NI myRIO и убедитесь, что код продолжает выполняться.
- 10. Заново подключите USB кабель.
- 11. В проекте LabVIEW выберите Connect.
- 12. Щелкните кнопку стоп на передней панели LabVIEW для остановки кода.

Программируемая логическая интегральная схема

ПЛИС — это кремниевые чипы, работающие в матричной структуре логических блоков, соединенных программируемыми перемычками. Вы можете настраивать эти блоки для

обработки стандартных логических операций и, во многих случаях, более сложной логики. Первые коммерческие ПЛИС были изобретены со-основателями Xilinx Россом Фриманом и Бернардов Фондершмиттом в 1985 году. Главное преимущество ПЛИС состоит в том, что вы можете изменять логику на аппаратном уровне без физических изменений оборудования. Это значит, что вы можете создавать собственную логику для своих систем и реконфигурировать ПЛИС для выполнения логики в работе. Вы программируете "сущность" ПЛИС на программном уровне, а затем воплощаете её в кремнии. В силу природы ПЛИС, отдельные секции ПЛИС на чипе независимы и могут выполняться в истинном параллелизме.

Истинный параллелизм означает, что задачи, выполняемые на ПЛИС, по-настоящему независимы и высоко детерминированы. Детерминизм критически важен в задачах контроля, роботизации и других задачах механотроники (типичная ПЛИС система может быть создана для взаимодействия с цифровым входом длительностью 25 нсек (40 МГц) а иногда и быстрее). Некоторые примеры ПЛИС-приложений LabVIEW включают интеллектуальный сбор данных. контроль на сверх-высоких скоростях. специализированные протоколы взаимодействия, настройка последовательности задач процессора, для сохранения процессорного времени для более сложного анализа, комплексного тактирования и синхронизации и программно-аппаратного тестирования. Благодаря возможности реагировать так быстро, ПЛИС позволяет вам создавать системы и эксперименты промышленного класса без крупных инвестиций и промышленного оборудования.

С модулем LabVIEW FPGA процесс программирования ПЛИС полностью графический и поддерживает полностью автоматическую компиляцию. Вы программируете ПЛИС с использованием модуля LabVIEW FPGA. Затем, когда вы готовы, LabVIEW reнерирует промежуточные VHDL файлы, требуемые компилятором Xilinx, запускает компилятор и передает ему файлы. В результате получается побитовый файл, загружаемый во флэшпамять, и считывающийся в работе. При работе ПЛИС, он считывает файл, и перенастраивается в соответствие с его инструкциями.

Упражнение 4. Исследование NI myRIO ПЛИС

- 1. Откройте диалог LabVIEW Getting Started и выберите Create Project.
- 2. В диалоге выберите стартовую точку для проекта из дерева в левой части. В категориях шаблонов, выберите **myRIO**.
- 3. В правой рамке, выберите **myRIO** Custom FPGA Project. Информацию по настройке проекта можно найти по ссылке *more information*.
- 4. Выберите Next для настройки проекта.
 - а. Задайте проекту подходящее имя, выберите директорию для сохранения и выберите устройство NI myRIO, подключенное по USB.
- 5. Выберите **Finish** для создания проекта.
- 6. Когда браузер проектов LabVIEW загрузит проект, "Мой компьютер" и myRIO-1900 (xxx.xx.xx) появляются как целевые устройства.
- 7. Раскройте целевое устройство NI myRIO и обратите внимание на новое дерево "chassis" (шасси). Такое дерево имеется в устройствах, содержащих ПЛИС, который можно задать целевым. Логически, целевой ПЛИС содержится в дереве шасси. Первоочередная задача ПЛИС — обрабатывать вход/выход на устройстве NI myRIO, так что вы можете обнаружить вход/выход под ПЛИС в дереве. Иерархия ПЛИС построена на папках, так что пользователь может сказать, где физически расположен каждый узел входа/выхода. Два МХР разъема, MSP разъем и встроенный вход/выход имеют каждый уникальную папку. Папки в свою очередь делятся по типу входа/выхода (цифровой/аналоговый) и физические банки. Вы можете перетацить их на ПЛИС-VI для чтения/записи соответствующего разъема. Каждый контроллер или индикатор с передней панели может быть прочитан/записан в VI реального времени.
- 8. Откройте "myRIO-1900 Customized FPGA.vi" для просмотра кода в ПЛИС.
- 9. Исследуйте код.

Целевое устройство ПЛИС имеет таймер 40МГц, настроенный в проекте LabVIEW. Любой VI в этой ветке автоматически становится ПЛИС-VI, и LabVIEW автоматически настраивает функции и тип данных, допустимые в VI. Вы можете создать новый ПЛИС-VI или модифицировать стандартный ПЛИС-VI.

Стандартный ПЛИС-VI подтверждает безопасность обработки входных/выходных данных через ПЛИС и подготовку к передаче в VI реального времени. NI myRIO поставляется со стандартной настройкой ПЛИС, которая обрабатывает все входные и выходные данные через оба МХР-разъема и MSP-разъем. ПЛИС.

Для упрощения архитектуры проектов, используйте стандартную настройку ПЛИС и только программируйте хост-VI реального времени (который выполняется на процессоре NI myRIO) и Windoiws VI (который выполняется на компьютере).

Ресурсы и дальнейшая работа

Наборы аксессуаров

Вы можете заказать наборы аксессуаров для упрощения разработки проектов для устройства NI myRIO. Заказать их можно посетив страничку <u>ni.com/myrio/accessories</u>.

Стартовый набор включает в себя держатель батареи, протоплату (подключаемую в МХР порты), набор проводов, светодиоды, переключатели, динамик, микрофон, двигатель постоянного тока, декодеры, датчик эффекта Холла и пьезоэлемент. Этот набор позволяет студентам быстро начать построение простых цепей и расширять свое понимание цифровых и аналоговых входов и выходов.

Мехатронный набор включает в себя все, что имеется в стартовом наборе, плюс двигатели постоянного тока с декодерами, сенсор естественного освещения, ультразвуковой дальномер, сервоприводы, компас, драйвер постоянного тока (мостовая схема управления), инфракрасный дистанционный датчик, трехосный цифровой гироскоп и трехосный акселерометр. Он предназначен для студентов, стоящих простых роботов и махатронные системы.

Набор для встраиваемых систем включает в себя ЖК-экран, цифровой термометр, цифровой потенциометр, барометр, клавиатуру, матрицу из светодиодов, набор RFID и электронно-программируемое ПЗУ. Он предназначен для студентов, создающих системы, работающие автономно с хост-компьютером.

NI myRIO Project Essentials Guide

NI myRIO Project Essentials Guide представляет собой мультимедийный учебный ресурс для студентов, работающих над проектами любой сложности. Он создан для помощи студентам начать взаимодействие NI myRIO с широким кругом датчиков, приводов и других элементов с помощью LabVIEW. Он рассматривает способы соединения, требования входа/выхода, теоретические основы устройства и программирование более 20 разных устройств, могущих встретится в проектах для NI myRIO.

ni.com/myrio/project-guide

Поддержка языка С в NI myRIO

NI myRIO построена на технологии NI RIO (реконфигурируемый вход/выход), позволяющей вам программировать как процессор, управляющей ОС реального времени, так и ПЛИС. Кроме LabVIEW, процессор NI myRIO может быть запрограммирован на языке С или C++ с помощью стандартных функций ПЛИС.

ni.com/myrio/c-support

Сообщество NI myRIO

Общайтесь с другими пользователями NI myRIO и делитесь проектами и примерами кода.

ni.com/community/myrio

Дальнейшие шаги

National Instruments инвестируют в успех профессоров и студентов, использующих наше оборудование. Теперь, когда вы ознакомились с NI myRIO и потенциалом устройства, следующим шагом станет более подробное знакомство с LabVIEW и системами реального времени. National Instruments предлагает множество курсов для обучения как успешно настраивать оборудование, писать код и внедрять его. Два наиболее легкодоступных это два онлайн ресурса под названием "Learn LabVIEW" и "Learn RIO". Вы можете бесплатно получить доступ к ним, посетив страничку <u>ni.com/students/learn</u>. Оба курса содержат видео-уроки и простые наборы упражнений, позволяющие ускорить обучение. Изучение этих модулей настоятельно рекомендуется всем пользователям для всех пользователей NI myRIO.