



Введение	5
▼ 1	
Система проектирования LabVIEW: возможности и перспективы.....	8
▼ 2	
Основы проектирования систем на базе виртуальных инструментов.....	11
Практические примеры решения инженерных задач в среде LabVIEW.....	13
Организация циклических алгоритмов в среде LabVIEW.....	24
Обработка сигналов в среде LabVIEW	32
▼ 3	
Конфигурирование системы LabVIEW в Measurement & Automation Explorer.....	48
Конфигурирование модулей DAQ.....	49
Диагностика устройства DAQ в MAX.....	54
Диагностика линий цифрового вывода модулей DAQ	57
Тестирование линий цифрового ввода модуля DAQ.....	60
Тестирование цифровых счетчиков/таймеров устройства DAQ.....	62
Диагностика каналов аналогового ввода/вывода модуля DAQ.....	67
Измерение параметров аналогового сигнала на выходе звуковой карты	74
▼ 4	
Измерение параметров сигналов в системе LabVIEW.....	80
Техника измерения параметров аналоговых сигналов	80
Практические примеры систем измерения параметров непрерывных сигналов.....	87

▼ 5

Цифровой синтез сигналов в системе LabVIEW.....	107
Синтез сигналов с амплитудной модуляцией.....	108
Цифровой синтез сигналов с угловой модуляцией в LabVIEW.....	112
Синтез сигналов с широтно-импульсной модуляцией.....	116
Преобразование непрерывных сигналов в LabVIEW.....	121

▼ 6

Дискретный ввод-вывод в LabVIEW.....	133
--------------------------------------	-----

▼ 7

Периферийные устройства ПК в LabVIEW: звуковая карта.....	155
---	-----

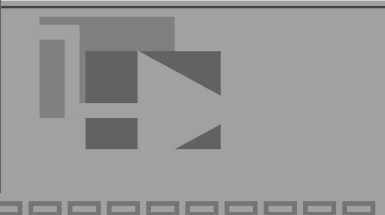
▼ 8

Периферийные устройства ПК в LabVIEW: последовательный порт.....	176
Тестирование последовательного порта с помощью I/O Assistant.....	178
Передача данных в последовательный порт в LabVIEW.....	181
Чтение данных из последовательного порта в LabVIEW.....	183

▼ 9

LabVIEW в сетях TCP/UDP.....	191
Обмен данными по протоколу UDP.....	193
Обмен данными по протоколу TCP.....	203

Введение



Качественное проектирование современных систем управления, автоматизации и измерений требует огромных усилий от инженеров и разработчиков, которые сталкиваются с необходимостью анализа и обработки большого объема информации в процессе разработки даже относительно несложных по структуре систем. Практически любая система управления требует от инженера знаний и практических навыков в смежных областях знаний, таких, например, как математика, электроника, программирование, конструирование и дизайн и т. д. Сложная система взаимосвязей различных компонентов проектируемых систем управления способна существенно усложнить задачи разработки даже для целых коллективов.

Современные технологии проектирования систем управления существенно облегчают жизнь разработчикам. В настоящее время процесс разработки систем в значительной степени упростился благодаря наличию программных систем автоматизации проектирования. Системы автоматизированного проектирования широко используются как при разработке многоцелевых систем автоматизации и управления, так и для решения задач специализированного характера в конкретной предметной области.

Среди систем автоматизированного проектирования особое место занимает пакет программ LabVIEW фирмы National Instruments. LabVIEW – это среда разработки лабораторных виртуальных приборов, в которой используется особый язык графического программирования, существенно упрощающий проектирование систем любой степени сложности для промышленности, образования и научных лабораторий. Среда проектирования LabVIEW во многих случаях является стандартным инструментом для систем сбора данных и управления приборами в научных и заводских лабораториях.

Система проектирования LabVIEW является многоплатформенной и может функционировать на персональных компьютерах с различными операционными системами (Windows, MacOS, Linux и т. д.). Широкая популярность LabVIEW обусловлена наличием интуитивно понятного графического пользовательского интерфейса, который существенно упрощает проектирование систем любого уровня сложности.

Предлагаемая вниманию читателей книга является практическим руководством по проектированию систем управления и измерения в среде LabVIEW. Мате-

риал книги включает несложные и легко повторяемые примеры проектирования часто встречающихся в инженерной практике задач. Виртуальные инструменты, описанные в книге, окажут помощь инженерам-практикам при решении задач измерения, анализа и цифрового синтеза непрерывных и дискретных сигналов, при построении систем сбора данных. Особое внимание в книге уделено аппаратно-программным интерфейсам системы LabVIEW с промышленными многофункциональными измерительными модулями, а также разработке виртуальных инструментов на базе периферийных устройств персональных компьютеров.

Многие современные системы управления и сбора данных состоят из различных подсистем, связанных между собой сетевыми интерфейсами. В книге приводятся несложные примеры построения распределенных систем управления на базе стека протоколов TCP/IP.

Книга будет полезна инженерам-практикам и разработчикам систем управления и сбора данных, а также всем желающим усвоить практические навыки проектирования систем в среде LabVIEW. Материал книги включает несколько глав, краткое описание каждой из которых приводится далее.

Глава 1. Система проектирования LabVIEW: возможности и перспективы. В данной главе рассматриваются архитектура системы LabVIEW и базовые аспекты использования этой системы проектирования при разработке систем управления и измерения.

Глава 2. Основы проектирования систем на базе виртуальных инструментов. Материал этой главы посвящен основам разработки систем на базе графического языка программирования и тесно связанной с этой темой концепции виртуальных инструментов. Приводятся многочисленные практические примеры создания и конфигурирования виртуальных инструментов для анализа и синтеза сигналов, а также рассматриваются основы их математической обработки в среде LabVIEW.

Глава 3. Конфигурирование системы LabVIEW в Measurement & Automation Explorer. В этой главе рассматриваются практические аспекты подготовки к эксплуатации и конфигурирование аппаратных многофункциональных модулей сбора и обработки данных (DAQ) с помощью программы Measurement & Automation Explorer. Значительное внимание уделено созданию и настройке специальных программных модулей (заданий), которые позволяют виртуализировать аппаратные интерфейсы системы управления и существенно тем самым упростить процесс создания виртуальных инструментов. Приводятся многочисленные примеры с детальным описанием и рекомендациями по практическому применению модулей DAQ.

Глава 4. Измерение параметров сигналов в системе LabVIEW. Материал главы описывает базовые аспекты измерения и обработки непрерывных сигналов в системах на базе виртуальных инструментов LabVIEW. Детально рассматриваются принципы цифровой обработки сигналов с помощью аналого-цифровых преобразователей, а также принципы фильтрации сигналов на базе цифровых фильтров. Значительное внимание в главе уделено методам анализа сигналов в частотной и временной области. Детально рассматриваются проблемы шумопо-

давления, приводятся практические решения данной проблемы при измерении сигналов. Теоретический материал главы сопровождается многочисленными примерами разработки виртуальных инструментов, иллюстрирующими изложенные концепции.

Глава 5. Цифровой синтез сигналов в системе LabVIEW. Данная глава посвящена вопросам цифрового синтеза сигналов в системе LabVIEW. Рассматриваются различные виды модуляции непрерывных сигналов для систем управления, телекоммуникаций и связи и их реализация в среде LabVIEW. Теоретические аспекты подкреплены практическими примерами разработки виртуальных инструментов, осуществляющих синтез сигналов.

Глава 6. Дискретный ввод-вывод в LabVIEW. В этой главе рассматриваются теоретические и практические аспекты обработки дискретных сигналов цифровой логики, а также принципы построения интерфейсов цифрового ввода-вывода в системах управления и сбора данных. Приводятся примеры разработки виртуальных инструментов для измерения характеристик импульсных сигналов.

Глава 7. Периферийные устройства ПК в LabVIEW: звуковая карта. Материал главы посвящен созданию систем обработки сигналов на базе интерфейса звуковой карты персонального компьютера. Подробно описываются принципы разработки виртуальных инструментов, в которых используются аналого-цифровой и цифроаналоговый интерфейсы звуковой карты, а также методы анализа аудиосигналов в среде LabVIEW.

Глава 8. Периферийные устройства ПК в LabVIEW: последовательный порт. В данной главе рассматриваются методы обмена данными между приложениями LabVIEW посредством последовательного интерфейса, а также примеры реализации последовательных интерфейсов между виртуальными инструментами LabVIEW и внешними программами.

Глава 9. LabVIEW в сетях TCP/UDP. Здесь рассматриваются основы построения распределенных систем обработки данных на основе стека протоколов TCP/IP. Теоретический материал главы сопровождается практическими примерами создания распределенных сетевых систем управления, в которых показано взаимодействие клиентских и серверных приложений с использованием среды LabVIEW и Microsoft Visual C++.

Система проектирования LabVIEW: возможности и перспективы



Разработка комплексных систем автоматизации и контроля для промышленных и научных целей требует больших усилий от специалистов различного профиля: инженеров-конструкторов, инженеров-системотехников, программистов и др. Современные средства разработки аппаратной и программной части таких систем достаточно развиты, но тем не менее проблемы универсализации и ускорения разработки систем стоят на первом месте.

Система LabVIEW является одним из наиболее развитых программных инструментов для быстрого и качественного проектирования систем управления и сбора данных. Аббревиатура «LabVIEW» расшифровывается как Laboratory Virtual Instrument Engineering Workbench – среда разработки лабораторных виртуальных приборов. LabVIEW является средой программирования, с помощью которой вы можете создавать приложения, используя графическое представление всех элементов алгоритма, что отличает ее от обычных языков программирования, таких как C, C++ или Java, где программируют, используя текст.

Следует отметить, что графический язык проектирования, характерный для системы LabVIEW, не является единственной отличительной особенностью этой среды проектирования. Основное назначение среды проектирования LabVIEW состоит в том, чтобы обеспечить для широкого круга инженеров и исследователей полный цикл разработки системы, начиная от создания алгоритма и заканчивая запуском аппаратно-программной части приложения. Система проектирования LabVIEW позволяет создавать приложения в любой из распространенных операционных систем, таких как Windows, MacOS и Linux.

Для разработчика среда проектирования LabVIEW представляется собой единый аппаратно-программный комплекс, структуру которого можно представить в упрощенном виде так, как показано на рис. 1.1. Программное обеспечение среды LabVIEW может функционировать на персональных компьютерах, ноутбуках и специализированных шасси, например на широко распространенном инструментальном модуле PXI. Аппаратная часть системы LabVIEW может включать, в зависимости от выбранной конфигурации, набор специализированных модулей обработки данных (модули DAQ), работающих либо на шине PCI/PCI-X/USB персональных компьютеров, либо в автономном PXI-модульном инструменте под управлением специализированной операционной системы.

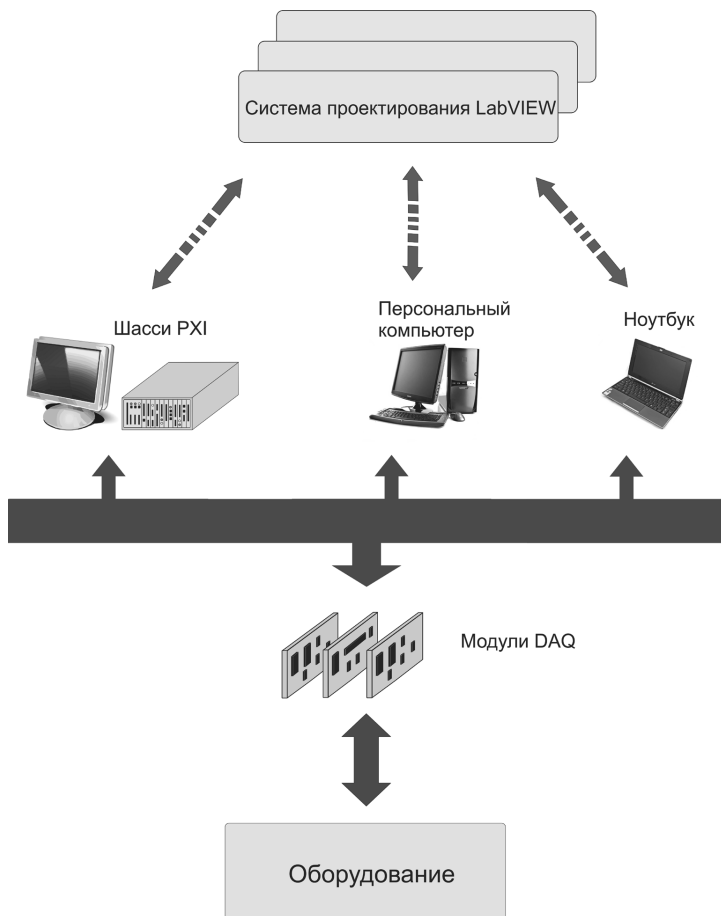


Рис. 1.1

Замечательной особенностью системы LabVIEW является то, что вне зависимости от аппаратной конфигурации для разработчика существует единый аппаратно-программный интерфейс в форме заданий (task), который позволяет манипулировать с любыми аппаратными модулями DAQ как с логическими устройствами. Это позволяет инженеру-разработчику целиком сконцентрироваться на проектировании системы, не вдаваясь в подробности аппаратной реализации системы на низком уровне.

Виртуализация устройств DAQ дает возможность абстрагироваться от аппаратного интерфейса (PCI, PXI или USB), что позволяет при необходимости гибко менять аппаратную конфигурацию разрабатываемой системы управления.

Мощный графический язык программирования LabVIEW помогает существенно улучшить эффективность проектирования. Создание законченного приложения с помощью обычных средств программирования может отнять очень много времени, тогда как с LabVIEW требуется лишь несколько часов, поскольку пакет специально разработан для программирования различных измерений, анализа данных и оформления результатов. Так как LabVIEW имеет гибкий и интуитивно понятный графический интерфейс программирования, он также отлично подходит для моделирования процессов, презентации проектов и создания приложений общего плана.

Для облегчения выполнения повседневных задач разработки в LabVIEW имеется расширенная библиотека функций и готовых к использованию подпрограмм, реализующих большое число типичных задач программирования, которые в обычных языках программирования требуют существенных затрат времени (например, распределение памяти, работа с переменными и массивами и т. д.).

В LabVIEW также содержатся специальные библиотеки виртуальных инструментов для ввода/вывода данных со встраиваемых аппаратных модулей DAQ, для работы со стандартным интерфейсом ввода-вывода (General Purposes Interface Bus, GPIB), управления устройствами через последовательный интерфейс RS-232, программные компоненты для анализа представления и сохранения данных, а также обмена через локальные сети и Интернет. Кроме того, специальная библиотека компонентов для анализа сигналов содержит множество полезных функций для генерирования сигналов и их обработки, цифровые фильтры, функции для статистической обработки сигналов, регрессионного анализа и т. д.

Система LabVIEW позволяет эффективно управлять потоками данных, их представлением и преобразованием. Выходные данные могут быть представлены практически в любой форме, включая графики, диаграммы, таблицы и т. д. Программы, разработанные в LabVIEW, легко импортировать на другие платформы. Приложение, разработанное, например, в MacOS, будет работать в Windows и Linux без каких-либо модификаций. Среда проектирования LabVIEW постоянно совершенствуется, так что уже в ближайшем будущем мы увидим ее новые возможности, особенно в области проектирования беспроводных систем управления и сбора данных.

Для разработки проектов этой книги используется система LabVIEW 2011, хотя все демонстрационные приложения будут работать и в более ранних версиях данной системы. В качестве аппаратного модуля сбора данных была применена плата NI PCI-6221, но все виртуальные инструменты будут работать и с другими PCI- и PXI-устройствами.

Основы проектирования систем на базе виртуальных инструментов



В этой главе мы рассмотрим основы проектирования систем измерения и контроля на базе виртуальных инструментов среды LabVIEW. Материал данной главы имеет практическую направленность – все рассмотренные примеры предназначены для решения часто встречающихся задач обработки, преобразования и анализа сигналов. Кроме того, при разработке примеров виртуальных инструментов обсуждаются базовые принципы реализации программных алгоритмов в LabVIEW: организация разветвленных и циклических вычислений, создание отчетов и визуализация данных.

Все приложения в LabVIEW представляют собой виртуальные инструменты (VI), все параметры которых можно сохранить в файлах с расширением .vi. Система LabVIEW включает множество визуальных и функциональных компонентов, позволяющих эффективно манипулировать с источниками сигналов и потоками данных, а также обеспечивать визуализацию данных на любом этапе функционирования системы и создавать отчеты. Для быстрого и эффективного освоения базовых принципов работы с VI наилучшим методом, с точки зрения автора, являются разработка и анализ практических примеров несложных систем измерения и контроля.

Система LabVIEW предназначена главным образом для создания лабораторных и промышленных систем, работающих в реальном времени. Источниками сигналов и потоков данных в таких системах являются датчики/трансдюсеры, устройства измерения и сетевые интерфейсы, а приемниками сигналов служат, как правило, исполнительные устройства (реле, двигатели и т. д.). Кроме того, практически все системы реального времени содержат устройства визуализации информации (дисплеи, графические индикаторы и т. д.). Все аспекты построения систем реального времени будут рассмотрены в последующих главах.

Для реализации примеров данной главы мы воспользуемся широкими возможностями системы LabVIEW в плане моделирования сигналов и процессов в реальном времени. Источниками сигналов в наших демонстрационных виртуальных инструментах будут компоненты – симуляторы сигналов программной среды LabVIEW, а все данные, полученные в результате функционирования наших виртуальных инструментов, будут отображаться с помощью графических и текстовых индикаторов.

Перед тем как приступить к реализации практических примеров, вкратце рассмотрим некоторые теоретические аспекты разработки систем в среде LabVIEW. Будучи очень мощной и многофункциональной системой проектирования, система LabVIEW предоставляет инженеру очень простой и интуитивно понятный интерфейс разработки, который базируется на нескольких фундаментальных представлениях.

Как уже было сказано, любое приложение в LabVIEW представляет собой виртуальный инструмент (VI). Приложение может включать в себя графические (визуальные) элементы (компоненты) управления и контроля, а также невидимые элементы (компоненты). Невизуальные элементы реализуют какие-либо функции (математические, логические, преобразования и генерации сигналов и т. д.). Визуальные и невидимые компоненты взаимодействуют между собой посредством соединений, через которые проходят потоки данных.

Система LabVIEW функционирует как среда потокового графического программирования, когда выполнение каких-либо операций зависит от наличия/отсутствия данных на входе узла (компонента). Графический язык программирования G, используемый в LabVIEW, основан на архитектуре потоков данных. Последовательность выполнения операторов в таком языке определяется наличием данных на их входах – этот подход отличается от классических языков программирования (C, Паскаля и т. д.), в которых ход выполнения программы управляется инструкциями (операторами). В языке G операторы, не привязанные к данным, выполняются параллельно в произвольном порядке.

Любое приложение в среде LabVIEW манипулирует с определенными данными, поступающими либо от физических (модули DAQ, измерительные инструменты и т. д.), либо от виртуальных (симуляторы сигналов) устройств. Программа в LabVIEW будет выполняться только в том случае, когда определены все данные в узлах. В процессе создания приложения в LabVIEW разработчик задает последовательность операций преобразования потока данных с помощью блок-схемы. На блок-схеме помещаются активные узлы, каждый из которых представляет собой визуальный или невидимый компонент; сами узлы взаимодействуют между собой посредством соединений. При этом соединение не обязательно рассматривается как физический интерфейс, скорее оно показывает направление потока данных от узла-источника к узлу-приемнику. Узел-источник и узел-приемник должны манипулировать одними и теми же типами данных, поэтому этот аспект необходимо учитывать при разработке приложений.

Хотя потоки данных являются основой проектирования приложений в LabVIEW, тем не менее для обеспечения дополнительной гибкости и расширения возможностей программирования в среде LabVIEW применяются методы, характерные для программ, управляемых потоком команд, как это реализовано в классических языках программирования, таких, например, как C/C++.

Очень часто, например, возникает необходимость в разработке LabVIEW-приложения, которое выполнялось бы с определенной цикличностью.

Примером такого приложения может служить генератор сигналов или измерительная система, в каждой из которых определенный алгоритм должен повторять-

ся непрерывно (циклически). В LabVIEW предусмотрены средства для организации циклических процессов, а также для ветвления программы при выполнении определенных условий – мы рассмотрим эти возможности в последующих примерах. Все примеры данной главы реализованы в среде NI LabVIEW 2011.

Практические примеры решения инженерных задач в среде LabVIEW

Для изучения основ проектирования систем в LabVIEW рассмотрим ряд примеров программирования. При изучении основ программирования и разработке демонстрационных виртуальных инструментов в качестве источников сигналов будут использоваться невидимые элементы (симуляторы сигналов) системы LabVIEW. Создание и программирование систем с реальными физическими сигналами рассмотрим в следующих главах этой книги.

В первом примере будет продемонстрировано создание виртуального инструмента, выполняющего обработку сигнала симулятора с последующим выводом результата на виртуальное устройство отображения (графический индикатор).

Пример 1. Предположим, что на вход нашей виртуальной системы поступает сигнал прямоугольной формы в диапазоне частот от 100 до 1000 Гц и амплитудой 1 В. Также предположим, что сигнал на выходе должен иметь треугольную форму, коэффициент амплитуды 0,33 и фазовый сдвиг 180° относительно исходного сигнала. Оба сигнала должны отображаться в форме кривых на одном и том же графике.

Создадим пустой виртуальный инструмент (**Blank VI**) в LabVIEW, выбрав соответствующий пункт меню (показан стрелкой на рис. 2.1).

Система создаст пустой проект, содержащий две панели: переднюю панель (**Front Panel**), которая называется панелью графического интерфейса пользователя; на этой панели можно разместить визуальные элементы управления (кнопки, движки (слайдеры), виртуальные окна и т. д.).

Вторая панель (**Block Diagram**) содержит блок-схему (функциональную схему) нашего приложения; эта панель является, по сути, программным интерфейсом нашего виртуального инструмента.

Разработчик может разместить на панели блок-схемы невидимые элементы, которые выполняют обработку сигналов и данных, а также задать направление потоков данных между компонентами системы с помощью соединений.

При работе с передней панелью нам будут доступны элементы управления на панели **Controls**. Если мы работаем с панелью блок-схемы, то получаем доступ к панели **Functions**, на которой можно выбрать элементы, реализующие какие-либо функции. По умолчанию система отображает панель графического интерфейса на переднем плане, а панель функциональной схемы – на заднем. Если по какой-либо причине панель функциональной схемы недоступна, то перейти к ней можно после выбора опции **Show Block Diagram** в меню Windows на панели интерфейса пользователя.

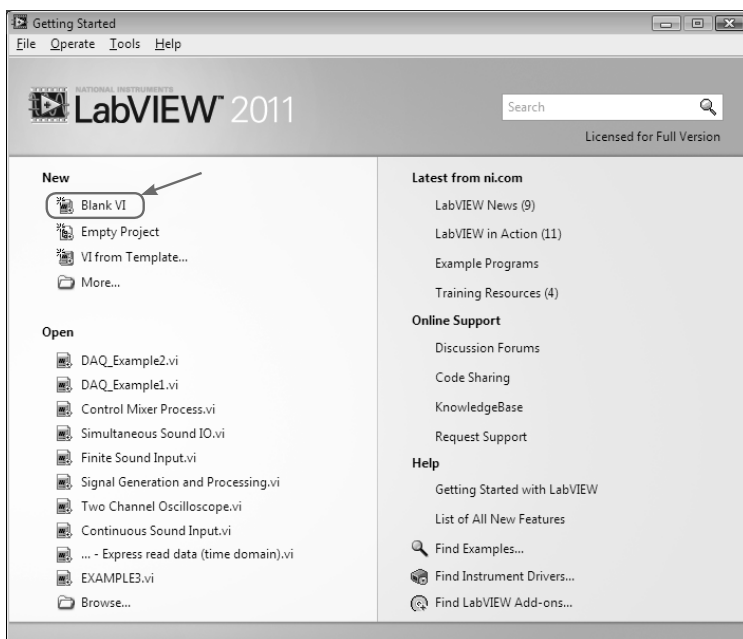


Рис. 2.1

Создание нашей системы начнем с размещения визуальных компонентов на панели интерфейса. Для отображения сигнала мы можем взять один из графических индикаторов на панели визуальных компонентов. На закладке **Silver** выберем компонент **Waveform Graph** (рис. 2.2), затем перейдем на панель интерфейса и разместим там выбранный компонент.

Поскольку мы собираемся регулировать амплитуду и частоту сигнала, то нам понадобятся два элемента регулирования из палитры **Controls** визуальных компонентов. На закладке **Silver** выберем компоненты **Knob** и **Vertical Pointer Slide**, разместим их на панели интерфейса пользователя, после чего сохраним наш виртуальный инструмент в файле Primer.vi (рис. 2.3–2.5).

С помощью кругового регулятора **Knob** мы сможем изменять амплитуду сигнала, а вертикальный слайдер **Vertical Pointer Slide** позволит устанавливать требуемую частоту сигнала. Теперь нам нужно установить параметры элементов управления. Во-первых, заменим заголовок **Knob** на **Amplitude, V**, а **Vertical Pointer Slide** – на **Frequency, Hz**.

Для тестового сигнала положим, что его частота изменяется в диапазоне от 100 до 1000 Гц, а максимальная амплитуда не будет превышать 1 В. В этом случае нам нужно задать требуемые параметры для элементов управления через страницу свойств (опция **Properties** в выпадающем меню). Для перехода на страницу

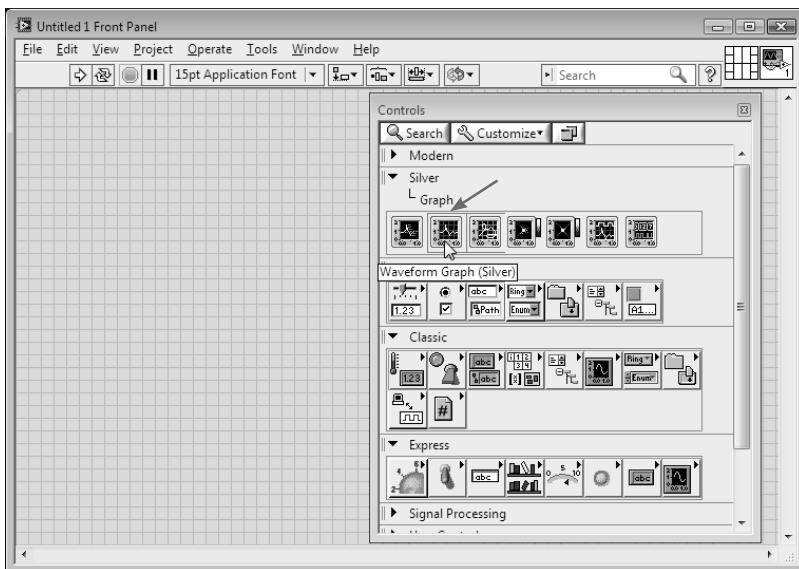


Рис. 2.2

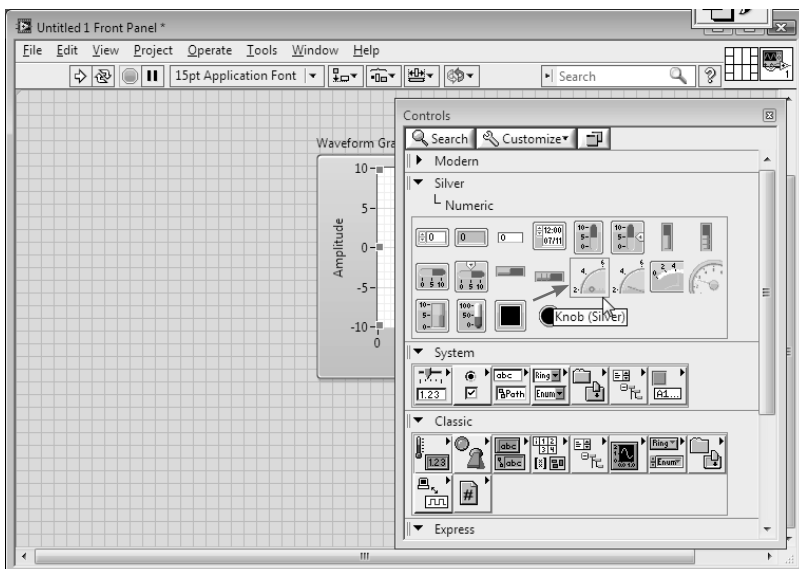


Рис. 2.3

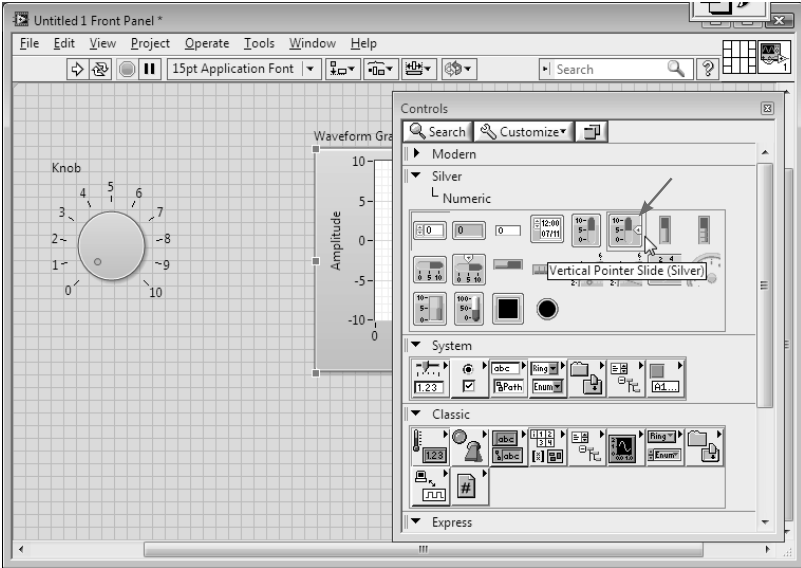


Рис. 2.4

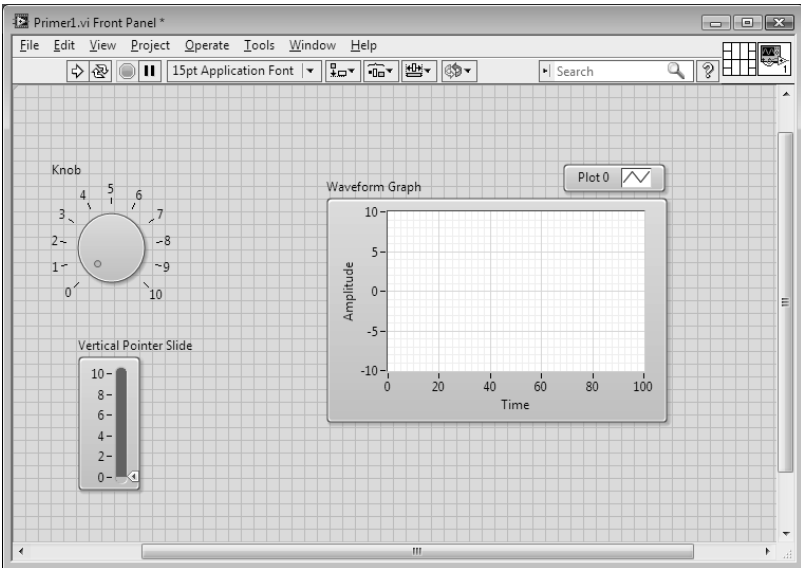


Рис. 2.5

свойств компонента LabVIEW нужно вначале выделить его пунктиром, затем щелкнуть правой кнопкой мыши на изображении компонента.

Для кругового регулятора **Knob** после выбора **Properties** нужно перейти на закладку **Scale** и выбрать максимальное значение регулировки (**Maximum**) равным 1 вместо установленного по умолчанию значения 10 (рис. 2.6).

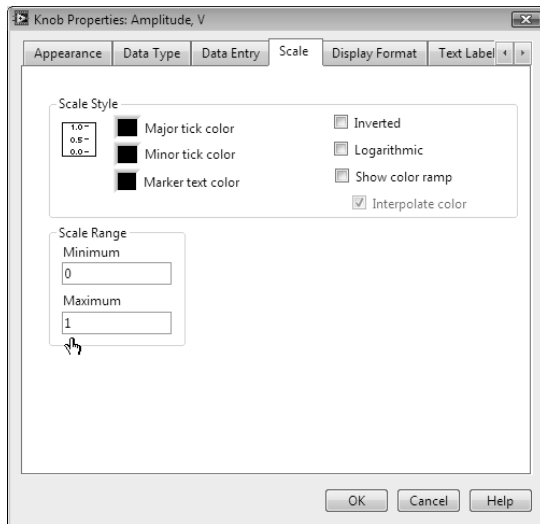


Рис. 2.6

Для вертикального слайдера на закладке свойств установим минимальное значение равным 100, а максимальное – 1000 (рис. 2.7).

Нам нужно, чтобы значение частоты отображалось в цифровой форме, поэтому нажмем правую кнопку мыши на компоненте слайдер и выберем опцию **Visible Items** ⇒ **Digital Display**. Еще один важный момент при установке параметров компонентов в LabVIEW связан с выбором типа данных, с которыми этот компонент может манипулировать.

Для компонента слайдер тип данных по умолчанию задается как число с фиксированной точкой двойной точности (**DBL**). В данном случае есть смысл заменить этот тип данных на целочисленный, поскольку частоту достаточно выбирать с точностью до одного герца. Для изменения типа данных нужно на странице свойств выбрать закладку **Data Type** и установить требуемый тип данных – в данном случае беззнаковое целое **U32** (рис. 2.8).

Подобную операцию выполним и для кругового регулятора амплитуды **Knob**, но здесь мы просто ограничим разрядность отображаемых данных. Сначала выведем цифровой дисплей, а затем зададим тип отображаемых данных как число с двумя значащими разрядами после запятой. После всех модификаций интер-

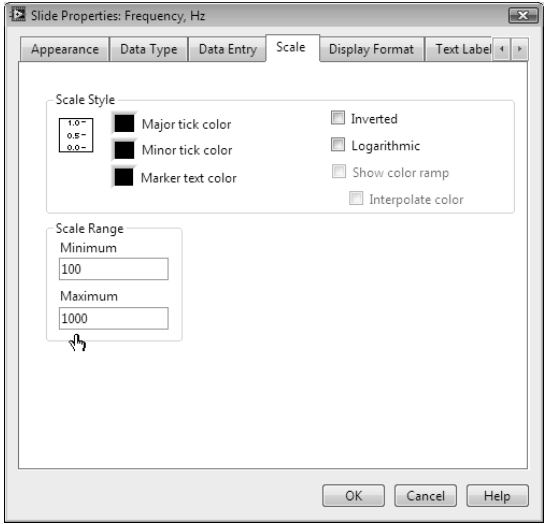


Рис. 2.7

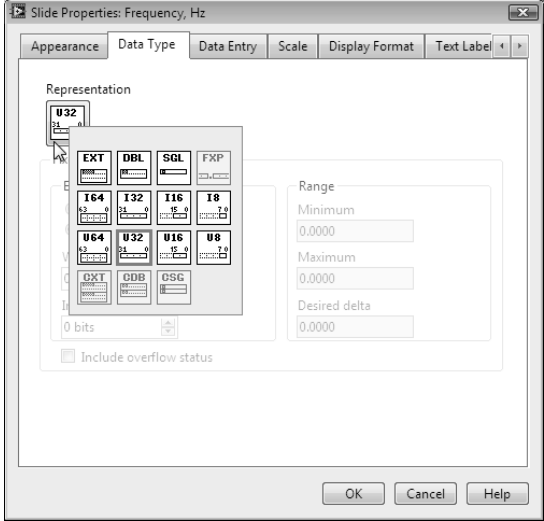


Рис. 2.8

фейс пользователя нашего виртуального инструмента выглядит так, как показано на рис. 2.9.

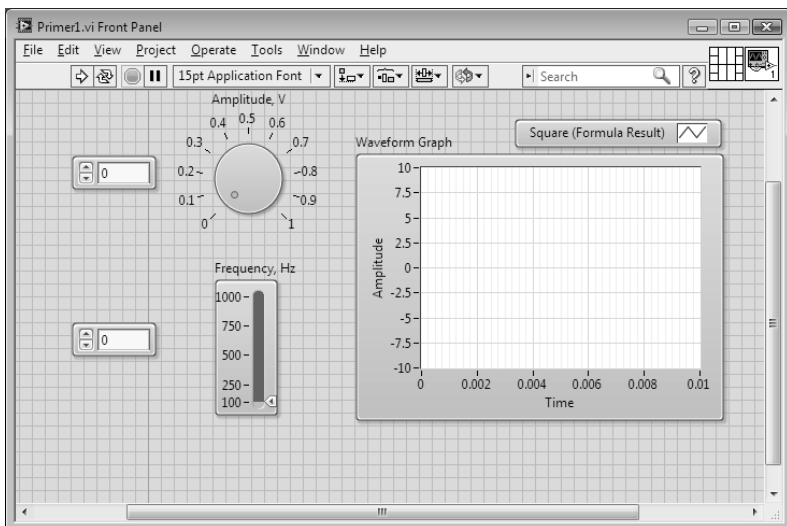


Рис. 2.9

После того как мы создали интерфейс пользователя, приступим к разработке алгоритма, для чего перейдем на панель блок-схемы нашего виртуального инструмента. Как видно по рис. 2.10, на блок-схеме размещены три компонента, пока никак друг с другом не связанные.

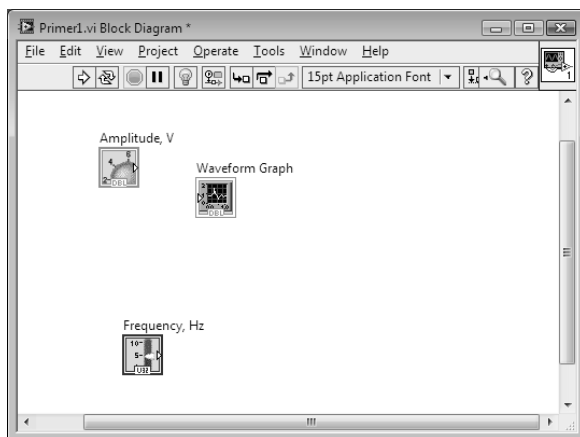


Рис. 2.10

При переходе к блок-схеме виртуального инструмента LabVIEW отображает панель функций **Functions**, содержащую группы невидимых элементов, выполняющих те или иные операции над данными. Большинство из этих элементов не отображаются на панели интерфейса, и для манипуляций с такими элементами необходимо работать в режиме блок-схемы.

Для нашего VI нам понадобятся несколько элементов на панели **Functions**. В качестве источника сигнала будет использоваться элемент **Simulate Signal**, который находится в категории **Functions** \Rightarrow **Express** \Rightarrow **Input**. После помещения этого элемента на блок-схему необходимо присоединить выход регулятора амплитуды сигнала ко входу **Amplitude**, что позволит изменять амплитуду сигнала. По аналогии вход **Frequency** элемента **Simulate Signal** нужно соединить с выходом вертикального слайдера, который задает частоту сигнала.

В самом элементе **Simulate Signal** на панели свойств нужно установить параметры сигнала, как показано на рис. 2.11.

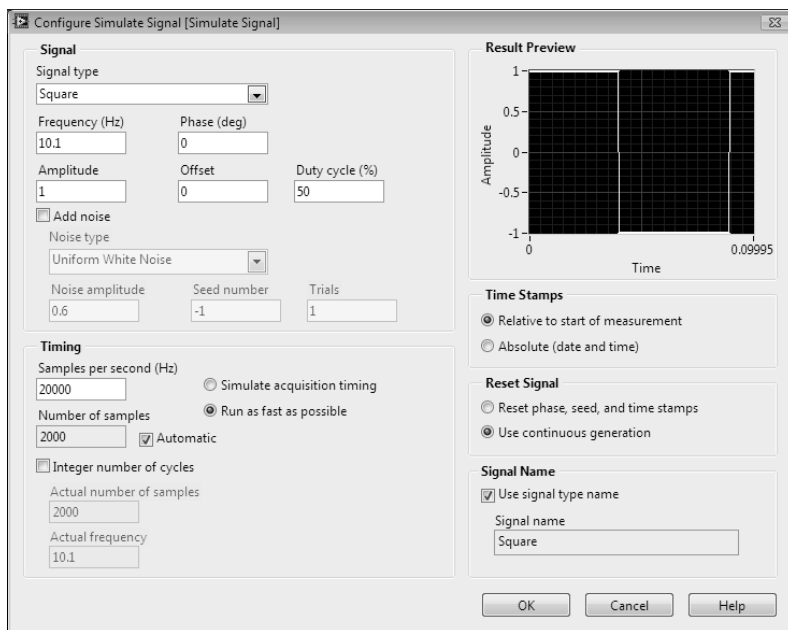


Рис. 2.11

Поскольку мы будем работать с сигналом прямоугольной формы, то параметр **Signal type** установим как **Square**. Далее для параметра **Amplitude** установим значение максимальной амплитуды сигнала равным 1. При установке параметров сигнала следует обратить особое внимание на временные характеристики синтезатора сигналов в LabVIEW.

В среде LabVIEW все аналоговые (непрерывные) сигналы синтезируются из цифровых сэмплов с определенной частотой. Чем выше частота преобразования или, по-другому, частота дискретизации, тем достовернее будет сигнал на выходе. При этом частота дискретизации и частота синтезируемого сигнала связаны теоремой Найквиста, из которой следует, что спектр синтезируемого сигнала не будет содержать побочных составляющих только в том случае, если частота дискретизации в два и более раз выше, чем частота наивысшей гармоники в спектре сигнала.

Например, для точного воспроизведения «чистой» синусоиды с частотой 1000 Гц частота дискретизации должна быть выше 2000 Гц. Для комплексных сигналов картина может быть намного сложнее, поскольку в их состав могут входить высшие гармоники. Так, последовательность прямоугольных импульсов, следующих с частотой 1000 Гц, содержит основную гармонику с частотой 1000 Гц и нечетные гармоники с убывающими амплитудами (3000, 5000, 7000 Гц и т. д.).

В этом случае частота дискретизации должна быть по возможности как можно выше, чтобы избежать искажения синтезируемого сигнала. На практике для такого сигнала можно выбрать частоту дискретизации, например, в два раза выше частоты 11-й гармоники, то есть 22 КГц.

Поскольку мощность гармонических составляющих быстро убывает с ростом частоты, то 11-я и высшие гармоники будут вносить очень малый вклад в спектр синтезируемого сигнала. В данном случае мы выберем частоту дискретизации 20 000 Гц – это значение необходимо указать в поле **Samples per second (Hz)** окна свойств элемента **Simulate Signal** (см. рис. 2.11).

Для преобразования прямоугольного сигнала в сигнал треугольной формы можно применить операцию интегрирования. Для этого на панели **Functions** нужно выбрать функцию **Express** ⇒ **Arithmetic & Comparison** ⇒ **Time Domain Math**. Элемент **Time Domain Math** позволяет выполнить операции дифференцирования и интегрирования входного сигнала. Разместим этот элемент на панели блок-схемы и откроем окно свойств (рис. 2.12).

На закладке **Mathematical Operation** установим отметку на кнопке **Summation (Sum[X])** – таким образом на выходе функционального элемента мы получим интегрированный входной сигнал. Для прямоугольного сигнала с постоянной амплитудой интеграл будет выражен линейной функцией, то есть выходной сигнал будет иметь треугольную форму.

Для преобразования можно использовать еще один элемент из панели **Functions** – речь идет о компоненте **Formula**. Этот элемент находится в той же категории, что и **Time Domain Math (Express ⇒ Arithmetic & Comparison ⇒ Formula)**. С помощью данного элемента можно легко создавать аналитические выражения сигнала посредством формул, получая при этом различные сигналы на выходе этого элемента.

После размещения данного элемента на блок-схеме откроем закладку **Properties** и преобразуем сигнал, поступающий на вход элемента **Formula**, как показано на рис. 2.13.

Как видно из рисунка, выходной сигнал элемента **Formula** определяется формулой следующего вида:

$$(-\text{Input1})/3+\text{Input2}.$$

Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

e-Univers.ru