

▼ Урок 1

Введение в LabVIEW	8
А. Инженерная среда программирования LabVIEW	9
В. Что такое виртуальный прибор (ВП)?	9
С. Практическое задание. Пример работы ВП в среде LabVIEW	12
Вопросы и задание	13

▼ Урок 2

Первая программа в LabVIEW	14
А. Элементы управления и отображения ВП	15
В. Пример оформления ВП в среде LabVIEW	18
Упражнение 2.1	19
Блок-диаграмма	22
Запуск ВП	24
Вопросы и задания	25
Вопросы	27

Проверочная работа № 1	27
------------------------------	----

▼ Урок 3

Данные в LabVIEW	28
А. Последовательность обработки данных	29
Режим анимации выполнения ВП	29
В. Типы и проводники данных	30

Типы данных	30
Проводники данных	31
Автоматическое соединение объектов проводниками данных	31
Соединение объектов проводниками данных вручную	32
С. Упражнение 3.1. ВП с данными логического типа	32
Лицевая панель	33
Блок-диаграмма	33
Задания	35

▼ Урок 4

Основные типы алгоритмических структур. Линейный алгоритм. Циклическая структура	36
А. Основные типы алгоритмических структур	37
Палитра Функций	37
В. Линейная алгоритмическая структура	39
С. Алгоритмическая структура «цикл» While	40
Д. Упражнение 4.1. ВП с оператором цикла While	41
Лицевая панель	41
Блок-диаграмма	42
Задания	44

▼ Урок 5

Основные типы алгоритмических структур.	
Алгоритмическая структура «ветвление»	45
А. Алгоритмическая структура «ветвление»	46
Функция Select	46
Упражнение 5.1. ВП Деление чисел	46
В. Алгоритмическая структура «выбор»	48
Терминалы входа и выхода	49
Логическая структура Case	50
Задача	50
С. Упражнение 5.2. ВП Извлечение квадратного корня	51
Лицевая панель	51
Блок-диаграмма	52
Запуск ВП	53
Задание	53

▼ Урок 6

Примеры использования алгоритмической структуры «ветвление». Сравнение чисел.....	54
А. Пример использования алгоритмической структуры «ветвление». Сравнение двух чисел.....	55
Упражнение 6.1. ВП Сравнение действительных чисел.....	55
В. Упражнение 6.2. Зависимость значения одной величины от текущего значения второй величины	59
Вопросы.....	61
Проверочная работа № 2	61

▼ Урок 7

Структура данных массива.....	62
А. Что такое массив?.....	63
В. Объявление массива элементов управления или отображения	64
С. Создание массива элементов управления или отображения с помощью цикла.....	65
Использование автоиндексации для установки значения терминала количества итераций цикла	66
Использование автоиндексации для задания элементов массива	66
Упражнение 7.1. ВП создания числовых массивов	67
Д. Двумерные массивы и вложенные циклы	69
Е. Функции работы с массивами	70
Вопросы.....	73
Проверочная работа № 3	73

▼ Урок 8

Графические возможности.....	74
А. График диаграмм для отображения потока данных.....	75
Упражнение 8.1. ВП для построения графика	76
Упражнение 8.2. Соединение нескольких Графиков диаграмм	78
В. График с постоянным шагом	81

Одиночный график с постоянным шагом.....	81
Упражнение 8.3. График квадратов.....	82
Упражнение 8.4. Отображение нескольких графиков с постоянным шагом на одном поле	83
Задание	85

▼ Урок 9

Проект «Метеостанция»	86
А. Постановка задачи	87
В. Случайные числа	88
С. Упражнение 9.1	90
Вопросы и задание	92

▼ Урок 10

Самые важные алгоритмы в программировании.	
Проект «Теплица»	94
А. Постановка задачи	95
В. Алгоритм поиска среднего значения. Доступ к значениям предыдущей итерации.....	96
Сдвиговые регистры	97
Стек сдвиговых регистров.....	98
Упражнение 10.1.....	98
С. Алгоритм поиска наибольшего значения	99
Упражнение 10.2	101
Д. Проект «Теплица»	102

▼ Урок 11

Создание подпрограмм.....	107
А. Модульный принцип построения программ	108
Иерархический принцип построения программ в LabVIEW	108
В. Модуль Формула.....	109
Узел Формулы.....	109
Упражнение 11.1. ВП Узел Формулы.....	109
С. Подпрограмма ВП.....	112

D. Создание иконки и настройка соединительной панели	113
Создание иконки ВП.....	113
Настройка соединительной панели	116
Выбор и редактирование шаблона соединительной панели.....	117
Привязка полей ввода/вывода данных к элементам лицевой панели	118
Упражнение 11.2. ВП Преобразования °С в °F	118
E. Использование подпрограмм ВП	121
Редактирование подпрограммы ВП	121
Установка значимости полей ввода/вывода данных: обязательные, рекомендуемые и дополнительные (необязательные)	122
Упражнение 11.3. ВП Термометр	122
F. Превращение выделенной секции блок-диаграммы ВП в подпрограмму ВП.....	127
Задания.....	128

▼ Приложение

Примеры оформления лицевых панелей и блок-диаграмм	129
---	-----

▼ Ответы

Ответы к проверочной работе № 1	136
Ответы к проверочной работе № 2.....	138
Ответы к проверочной работе № 3.....	138

Введение в LabVIEW



В первом уроке даются основные представления о программной среде LabVIEW.

На примере практического задания «Сложение и вычитание чисел» представлено, как открыть файл с ВП в среде LabVIEW, рассматриваются элементы ввода и вывода информации, расположенные на лицевой панели. Также показывается, как запустить и остановить работу ВП.

В конце урока есть вопросы по теме занятия.

В этом уроке изложены вопросы:

- А. Инженерная среда программирования LabVIEW.
- В. Что такое виртуальный прибор (ВП)?
- С. Практическое задание. Пример работы ВП в среде LabVIEW.

А. Инженерная среда программирования LabVIEW

Среда LabVIEW совершенствуется уже почти 30 лет.

В соответствии со своим названием LabVIEW – или *Laboratory Virtual Instrument Engineering Workbench* (среда разработки виртуальных приборов) – первоначально использовалась в исследовательских лабораториях. Представьте, что собирается большая группа классных специалистов в разных областях науки: физики, биологи, химики, лингвисты, конечно же, математики. Им необходимо осуществить большой проект, где потребуется сбор сложноструктурированной информации, потом необходимо провести довольно сложную обработку информации. Понятно, что для этого нужен программист и надо выбрать такую среду программирования, которая была бы понятна непрофессионалам в программировании. Желательно, чтобы среда была наглядной – графической, чтобы программы были похожи на схемы, чтобы каждый ученый в группе мог понимать смысл программ, быстро вносить изменения, мог легко анализировать результаты обработки данных. С большой вероятностью, для решения таких крупных проектов будет выбрана инженерная среда графического программирования LabVIEW. Поверьте, что осуществление больших проектов, охватывающих несколько разных областей наук, – это необходимое условие нашего успешного развития сегодня. Время исследователей-одиночек, ученых, «гигантов мыслей» прошло. И теперь успешны проекты, в которых работает команда профессионалов-специалистов: TIAM – Together Each one Achieves More. И можно предположить, что знакомство с основами LabVIEW в средней школе будет полезным занятием для ребят, мечтающих о самых разных профессиях и работе в крупных национальных проектах.

В. Что такое виртуальный прибор (ВП)?

Программа, написанная в среде LabVIEW, называется виртуальным прибором (ВП). Давайте выясним, почему она так называется.

Познакомимся с некоторыми историческими аспектами.

Пропустим период в развитии физических экспериментов, когда ученому приходилось обрабатывать данные вручную. Это был период скрупулезных, долговременных, очень рутинных экспериментов,



не всегда приводящих к результату, период гениальных провидцев и точных в мелочах тружеников.

Появившиеся вычислительные машины одни из первых начали использовать физики для обработки данных своих экспериментов. При изучении того или иного физического явления физики собирали прибор. Он мог быть как совсем простым, так и довольно сложным, включающим множество датчиков, вспомогательных механизмов, огромное число индикаторов: лампочек, экранов и т. д. Показания всех датчиков заносились в вычислительную машину, которая проводила обработку показаний. А потом результаты выводились на экраны, печатались на бумаге графики...

Прибор регистрирует данные, управляет сбором показаний. Программа на компьютере обрабатывает показания прибора, готовит данные для отображения на экранах и других индикаторах прибора.

Но в ходе проведения исследований очень часто необходимо изменять приборы. Надо удалить с приборной панели ненужные кнопки, добавить новое окошко для вывода показаний, да много еще чего изменить.

Подумайте, где легче вносить изменения?

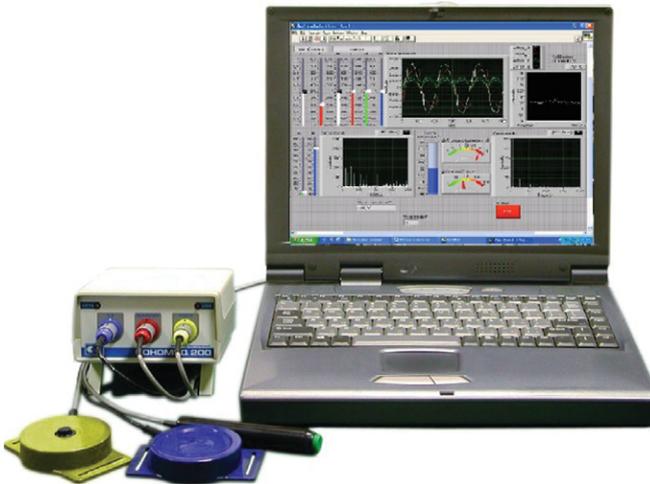
*Здесь?
на железной панели*



*Или здесь?
на экране монитора*



Понятно, что изменение лицевой панели прибора, например добавление еще одной лампочки или ручки управления, – довольно долгое и трудоемкое занятие. А на экране монитора можно легко и быстро менять элементы, добавлять их или убирать, изменять дизайн.



Итак, с развитием компьютерной техники появилась возможность оформления лицевой панели прибора на компьютере. С помощью компьютерной графики можно имитировать процесс включения лампочки, перемещения стрелки по шкале, отслеживать перемещение мыши на мониторе, изменять значения параметров, осуществлять

управление моторами. Поэтому реальная железная или пластмассовая лицевая панель прибора стала ненужной, необязательной. Ее роль выполняет виртуальная лицевая панель, некоторая абстрактная компьютерная модель, ее составляющие существуют только в программном коде и на экране монитора. Пользователь взаимодействует с виртуальной лицевой панелью, как с приборной панелью, «нарисованной» на экране. Наш прибор стал называться виртуальный прибор.

Теперь прибор регистрирует данные. Компьютер управляет сбором показаний, обрабатывает эти показания прибора. Результаты обработки отображаются на экране монитора, сохраняются в памяти компьютера. На компьютере в среде LabVIEW оформляется программа Виртуальный прибор, которая работает с реальными показаниями датчиков прибора, отслеживает реальные изменения параметров, но выводит все результаты эта программа на виртуальную лицевую панель, нарисованную на экране.

Главный вывод: показания датчиков, управление активными устройствами – реальные, элементы лицевой панели и сама лицевая панель – виртуальны.

Данные, с которыми работает ВП, являются реальными, это не симуляция. LabVIEW содержит полный набор инструментов для сбора, анализа, представления, передачи и хранения реально полученных данных.

С. Практическое задание.

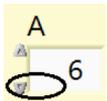
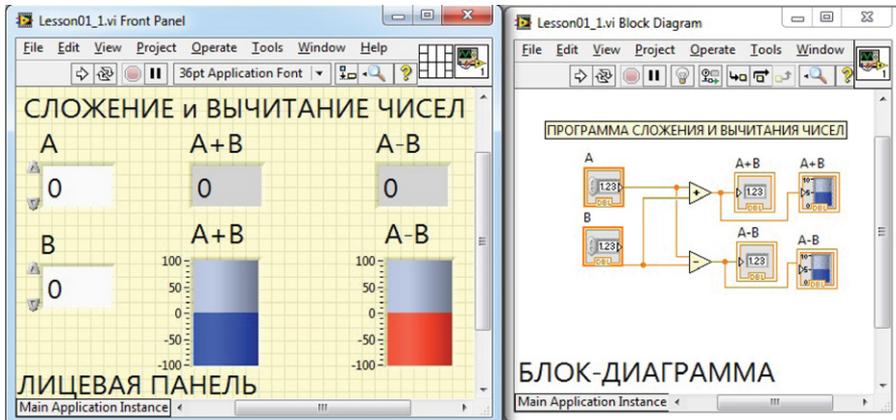
Пример работы ВП в среде LabVIEW

Давайте познакомимся с программной средой LabVIEW на примере тестирования ВП, который будет выполнять сложение и вычитание двух чисел. Откроем файл *Lesson01_1.vi*, расположенный в папке Упражнения.

Лицевая панель – это интерфейс пользователя ВП. Пример лицевой панели представлен слева. Интерфейс, напомним, – это сценарий, способ взаимодействия пользователя с программой. Наш сценарий очень похож на работу с прибором. Мы будем менять значения чисел на лицевой панели и сможем наблюдать, как меняются значения суммы и разности чисел.



Нажмем кнопку многократного запуска ВП, расположенную в панели окна блок-диаграммы или на лицевой панели, чтобы запустить выполнение ВП.



Попробуйте сами изменить значения чисел. Надо либо ввести число непосредственно, например в окошко A, либо с помощью нажатия мышки на кнопки: верхнюю – для увеличения значения, нижнюю – для уменьшения.

Вы заметите, что в некоторых окошках нельзя вносить изменения. Получается, что на лицевой панели есть элементы как для ввода значений пользователем, так и для вывода результатов, рассчитанных программой. Последние окошки недоступны для пользователя.



Нажмите эту кнопку для прекращения работы программы.

Вопросы и задание

1. Приведите примеры приборов: физических и любых других.
2. Как выглядит лицевая панель у приборов?
3. Как можно управлять работой приборами?

Первая программа в LabVIEW



Во втором уроке даются основные представления о программной среде LabVIEW. Мы сделаем с вами нашу первую программу, которая переводит значения температуры в градусах Цельсия в температуру в градусах Фаренгейта.

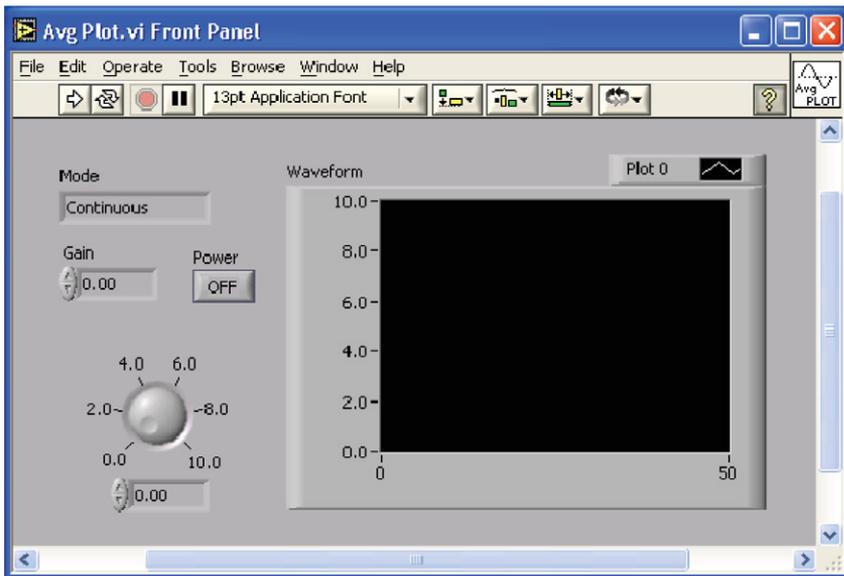
На уроке рассматриваются вопросы по темам «Шкалы и масштабы», «Измерение физических величин», используется формула перевода температуры из градусов Цельсия в градусы Фаренгейта без доказательства. Применяется переместительный закон для суммы и произведения чисел: «От перемены мест слагаемых сумма не меняется». Закрепляются названия значений при вычитании, делении чисел. Закрепляются темы «Арифметические действия с действительными числами: сложение чисел, умножение чисел на число», «Положительные и отрицательные числа», «Целая и дробная части действительного числа».

В этом уроке изложены вопросы:

- A. Элементы управления и отображения ВП.
- B. Пример оформления ВП в среде LabVIEW.

А. Элементы управления и отображения ВП

Лицевая панель – это интерфейс пользователя ВП. Пример лицевой панели представлен ниже.



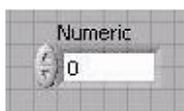
Лицевая панель создается с использованием палитры **Элементы** (Controls), которую можно вызвать, нажав правую кнопку мыши в любом месте лицевой панели.



1 – цифровой элемент управления; 2 – цифровой элемент отображения

Эти элементы могут быть либо средствами ввода данных – элементы **Управления** (Control), либо средствами отображения данных – элементы **Отображения** (Indicator). Никакой принципиальной разницы нет, какой вид имеет элемент **Управления**: в палитре таких элементов **Управления** довольно много. Внешний вид – это лишь

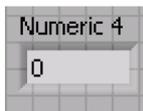
картинка, нарисованная для удобства представления тех или иных реальных кнопок, индикаторов ввода и т. п. Можно выбирать любую, устраивающую вас. Данные, вводимые на лицевой панели ВП, поступают на блок-диаграмму, где ВП производит с ними необходимые операции. Результат вычислений передается на элементы **Отображения** информации на лицевой панели ВП. Картинку этих элементов можно выбрать в панели элементов **Отображения**. Принципиального отличия между картинками нет.



Элементы **Управления** – кнопки, переключатели, ползунки и другие элементы ввода. Элемент имеет собственную метку, в данном случае **Numeric**. Это программное имя терминала. Ее можно изменять.

Выделим отличительные внешние особенности отображения значков элементов **Управления**, расположенных на лицевой панели:

- наличие стрелок для установки значения с левой стороны окошечка отображения данных;
- по умолчанию белый цвет поля в окошечке для отображения данных. Его цвет можно изменять в дальнейшем.



Элементы **Отображения** – графики, цифровые табло, светодиоды и т. д., элементы вывода. Элемент имеет собственную метку, в данном случае **Numeric 4**. Это программное имя терминала. Его можно изменять. Выделим отличительные внешние особенности отображения значков элементов

Управления, расположенных на лицевой панели:

- отсутствие стрелок для установки значения с левой стороны окошечка отображения данных, данные в этом окне с лицевой панели менять нельзя;
- по умолчанию серый цвет поля в окошечке для отображения данных. Цвет можно изменять в дальнейшем.

После помещения элементов **Управления** или **Отображения данных** на лицевую панель они получают свое графическое отображение на блок-диаграмме. Элемент, созданный на лицевой панели, можно удалить и на лицевой панели, и на блок-диаграмме.



Графическое изображение элементов **Управления** представлено слева. Различают два типа изображений: в полном виде (верхний значок) и в виде иконки (нижний маленький значок). Щелкнув правой кнопкой мыши по значку и поменяв параметр **View As Icon**, вы сможете изменить вид иконки. В больших программах в целях экономии места предпочтительнее использовать иконки. Элемент имеет

собственную метку, в данном случае **Numeric**. Это программное имя терминала. Его можно изменять. Выделим отличительные внешние особенности значков элементов **Управления**:

- утолщенная линия контура;
- небольшое острие стрелки с правой стороны границы, которое указывает направление перемещения данных «из терминала».



Графическое изображение элементов **Отображения (Индикаторов)** представлено слева. Различают два типа изображений: в полном виде (верхний значок) и в виде иконки (нижний маленький значок). Каждый элемент имеет собственную метку, в данном случае **Numeric 4**. Это программное имя индикатора. Его можно изменять. Выделим отличительные внешние особенности значков элементов **Отображения**:

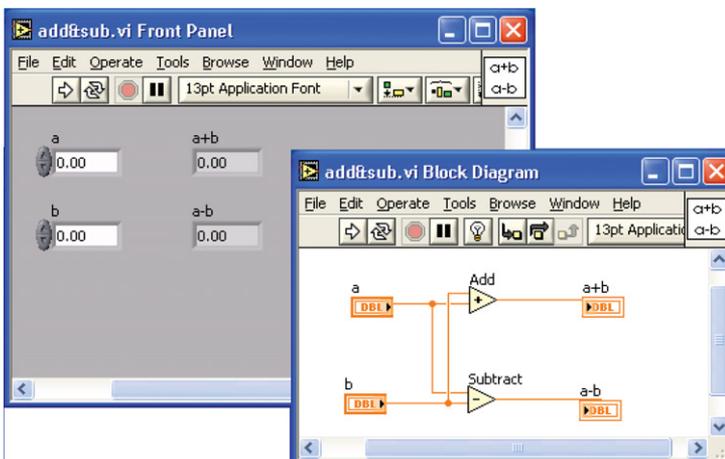
- тонкая линия контура;
- небольшое острие стрелки с левой стороны границы, которое указывает направление перемещения данных «в терминал».

Описанные выше особенности и свойства элементов (терминалов) **Управления** и **Отображения** данных применимы ко всем типам терминалов. С типами терминалов мы познакомимся на другом уроке.

Объекты блок-диаграммы включают графическое отображение элементов лицевой панели, операторов, функций, подпрограмм ВП, констант, структур и проводников данных, по которым производится передача данных между объектами блок-диаграммы.

Следующий пример показывает блок-диаграмму и соответствующую ей лицевую панель ВП нахождения суммы и разности двух чисел.

Этот ВП мы с вами тестировали на уроке 2.



В. Пример оформления ВП в среде LabVIEW



LabVIEW.exe

Давайте теперь приступим к созданию нашего первого ВП, программы в среде LabVIEW, работе, которую мы исследовали на предыдущем уроке. Откроем среду LabVIEW, пиктограмма которой имеет вид, показанный слева.

Среду можно запустить, например, так:

Пуск ⇒ **Программы** ⇒ **National Instruments** ⇒ **LabVIEW 11.0** ⇒ **LabVIEW**. После запуска среды LabVIEW появляется окно:



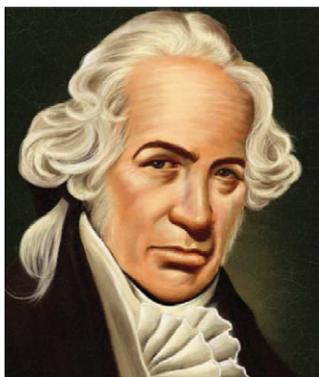
Нам надо открыть пустой ВП, выберем Blank VI.

Упражнение 2.1

Создадим ВП, который переводит значение температуры в градусах Цельсия в значение температуры в градусах Фаренгейта.

Даниель Габриель Фаренгейт (1689–1736) – немецкий физик. В 1714 году сделал ртутный термометр и предложил свою шкалу измерения температуры. На шкале Фаренгейта точка таяния льда равна $+32^{\circ}\text{F}$, а точка кипения воды $+212^{\circ}\text{F}$ (при нормальном атмосферном давлении).

Градус Цельсия назван в честь шведского ученого Андерса Цельсия (1701–1744), предложившего в 1742 году новую шкалу для измерения температуры. Первоначально в ней 0° (нулем) была точка кипения воды, а 100° – температура замерзания воды (точка плавления льда). Позже, уже после смерти Цельсия, его современники и соотечественники – ботаник Карл Линней и астроном Мортен Штремер – использовали эту шкалу в перевернутом виде (за 0° стали принимать температуру таяния льда, а за 100 – кипения воды). В таком виде шкала и используется до нашего времени.



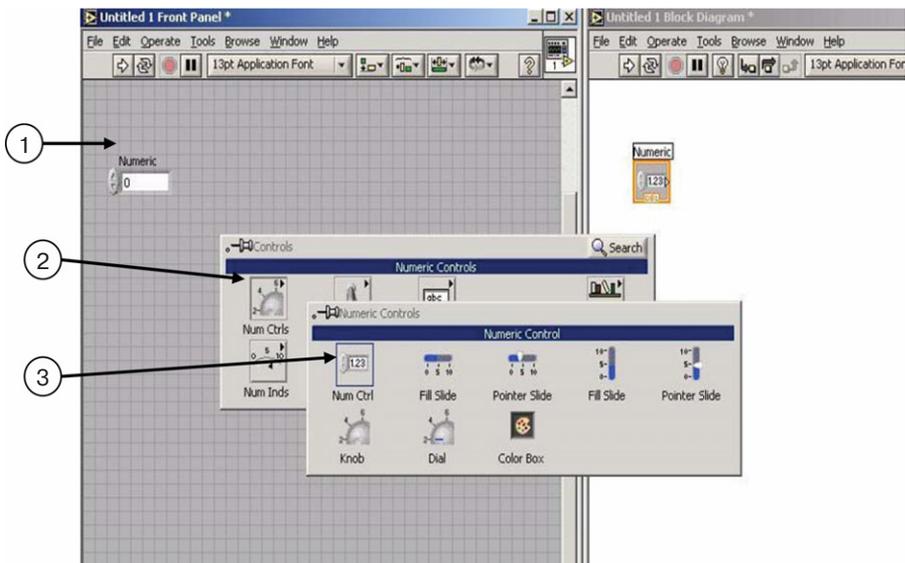
Даниель Габриель
Фаренгейт



Андерс Цельсий

Преобразование температуры из градусов Цельсия в градусы Фаренгейта осуществляется по формуле

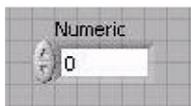
$$T_F = 1,8 \cdot T_C + 32.$$



Лицевая панель:

- 1 – собственная метка; 2 – подраздел палитры Элементов Numeric Controls;
3 – цифровой элемент управления

1. (Дополнительно) Выберем пункт главного меню **Window** ⇒ **Tile Left and Right** для вывода на экран рядом друг с другом лицевой панели и блок-диаграммы. Это удобно для программирования.
2. Создадим цифровой элемент управления. Он будет использован для ввода значений температуры в градусах Цельсия ($^{\circ}\text{C}$).



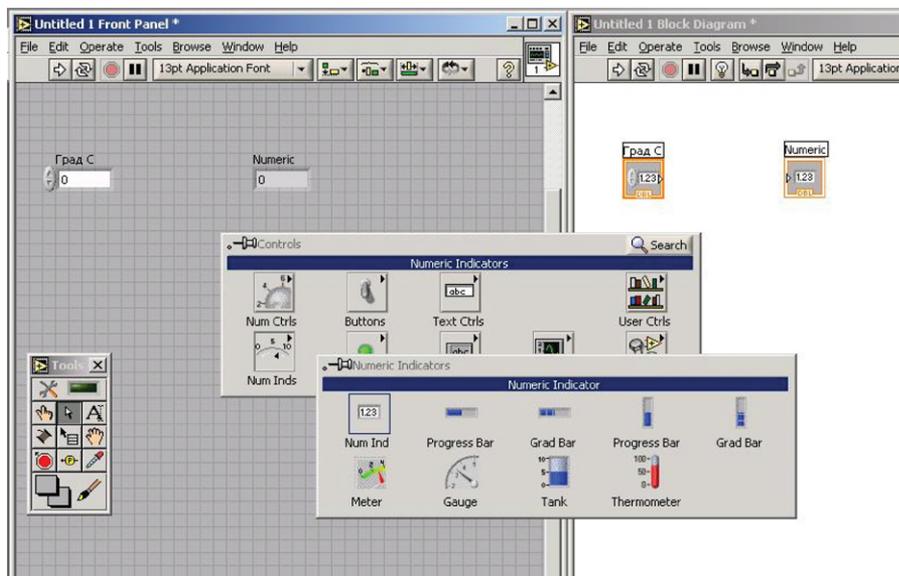
Выберем цифровой элемент управления в разделе палитры **Элементов** в подразделе **Controls** ⇒ **NumCtrl** (Числовые элементы управления). Для вывода на экран палитры **Controls** (Элементов) следует щелкнуть правой кнопкой мыши по рабочему пространству лицевой панели.

Поместите цифровой элемент управления на лицевую панель.



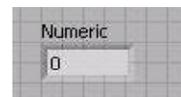
В поле собственной метки элемента управления напечатайте «**Град С**» и щелкните мышью в свободном пространстве лицевой панели или нажмите кнопку **Enter**, показанную слева, на инструментальной панели. Если сразу после создания элемента не присвоить имя его собственной метке, то LabVIEW присвоит имя,

заданное по умолчанию. Собственная метка в любое время доступна для редактирования, оно производится с помощью инструмента ВВОД ТЕКСТА, показанного слева. Этот инструмент можно выбрать в окне **Tools** ⇒ **Tools Palette**.



1 – цифровой элемент управления; 2 – цифровой элемент отображения

Создайте цифровой элемент отображения данных. Он будет использован для отображения значений температуры в градусах Фаренгейта.



Выберите цифровой элемент отображения в палитре **Элементов** в подразделе **Controls** ⇒ **NumInds** (Числовые элементы отображения).

Поместите элемент отображения данных на лицевую панель.

В поле собственной метки элемента управления напечатайте «Град F» и щелкните мышью в свободном пространстве лицевой панели или нажмите кнопку **Enter**.



На блок-диаграмме LabVIEW создаст терминалы данных, соответствующие элементам управления и отображения. Терминал данных представляет тип данных соответствующих элементов.



Внимание. Терминалы данных, соответствующие элементам управления, имеют более широкий обводной контур, по сравнению с терминалами данных, соответствующими элементам отображения.

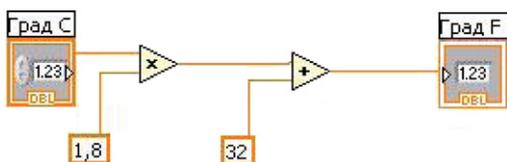
Блок-диаграмма

Перейдем на блок-диаграмму, выбрав пункты главного меню **Window** ⇒ **Show Diagram**.

Преобразование температуры из градусов Цельсия в градусы Фаренгейта осуществляется по формуле

$$T_F = 1,8 \cdot T_C + 32.$$

Надо запрограммировать эту формулу. Блок-диаграмма будет иметь такой вид:



Выберем функцию **Multiply** (Умножение) из палитры **Функций** в разделе **Functions** ⇒ **Numeric** (Арифметические функции). Поместим ее на блок-диаграмму. Для вывода на экран палитры **Functions** (Функций) следует щелкнуть правой кнопкой мыши в рабочем пространстве блок-диаграммы.



Выберем функцию **Add** (Сложение) из палитры **Функций** в разделе **Functions** ⇒ **Numeric** (Арифметические функции). Поместим ее на блок-диаграмму.



Выберем числовую константу из палитры **Функций** в разделе **Functions** ⇒ **Numeric** (Арифметические функции). Поместим две числовые константы на блок-диаграмму. После размещения числовой константы на блок-диаграмме поле ввода ее значений подсвечивается и готово для редактирования. Одной константе присвойте значение 1,8, другой – значение 32.

Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

e-Univers.ru