

УДК 53087

Лабораторная работа № 8

**Введение в LABVIEW**

**Александр Юрьевич Гольцов  
Максим Викторович Путилин**

**Кафедра прикладной физики  
ФПФЭ МФТИ  
2008 г.**

## **Содержание**

Введение.....	4
<b>Раздел 1.</b> Арифметические функции.....	6
<b>Раздел 2.</b> Циклы и условные операторы.....	15
<b>Раздел 3.</b> Функции работы с массивами.....	29
<b>Раздел 4.</b> Графическое представление данных.....	39
<b>Раздел 5.</b> Запись и считывание файлов.....	42
<b>Раздел 6.</b> Экспресс – виртуальные приборы.....	48
Список литературы.....	52

## Введение

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) – среда прикладного графического программирования, используемая в качестве стандартного инструмента для проведения измерений, анализа их данных, последующего управления приборами и исследуемыми объектами.

Сфера применения LabVIEW непрерывно развивается. В образовании она включает лабораторные практикумы по электротехнике, механике, физике. В инженерной практике – объекты промышленности, транспорта, в том числе воздушного, подводного и надводного флотов, космические аппараты. В науке – такие передовые центры как CERN (в Европе), Lawrence Livermore, Batelle, Sandia, Oak Ridge (США).

Динамично развивается и собственно среда LabVIEW. Первая ее версия (LabVIEW 1) была разработана корпорацией National Instruments (NI) в 1986 году в результате поисков путей сокращения времени программирования измерительных приборов. Версии со второй по шестую, каждая последующая из которых существенно расширяла возможности предыдущей версии по обмену данных с измерительными приборами и работе с другими программными средствами, появлялись соответственно в 1990, 1992, 1993, 1996 и 2000 годах. В 2003 году была создана корпорацией NI седьмая версия (LabVIEW 7). Главные ее особенности – введение раскрывающихся узлов, что расширяет возможности диалогового режима, и применение драйверов нового поколения NI-DAQmx, в результате чего существенно упрощается процедура сбора и отображения данных и ускоряется быстроедействие операций аналогового ввода-вывода. Следом за ней была анонсирована версия LabVIEW 7.1, имеющая ряд дополнительных функциональных возможностей. Последняя – восьмая версия (LabVIEW 8) была создана корпорацией NI в 2006 году.

В рамках этого курса лабораторных работ обучение программированию в среде LabVIEW будет проходить на версии LabVIEW 7.1.

Целью учебного курса является обучение:

- использованию среды LabVIEW для создания приложений;
- технике редактирования и отладки приложений;

- пониманию назначения лицевой панели, блок-диаграммы, соединительных панелей и иконок;
- использованию встроенных подпрограмм ВП и функций;
- созданию и сохранению ВП, для его последующего использования в качестве подпрограммы ВП.

В разделе 1 рассмотрены числовые и логические функции, создание и запуск виртуального прибора (ВП). В конце будет предложен ряд заданий для самостоятельного выполнения.

В разделе 2 рассмотрены циклы While, For, сдвиговые регистры и структура выбора Case.

В разделе 3 рассмотрены возможности создания массивов и основные функции работы с массивами.

В разделе 4 рассмотрены функции графического представления данных.

В разделе 5 рассмотрены функции, выполняющие файловые операции записи и считывания данных.

В разделе 6 рассмотрены экспресс-ВП и их основные функции.

## Раздел 1. Арифметические функции

---

### Создание виртуального прибора «Преобразование °C в °F»

Виртуальным прибором (ВП) называется программа, написанная в среде LabVIEW. ВП состоит из четырех основных компонентов – лицевой панели, блок-диаграммы, иконки и соединительной панели.

Лицевая панель – это интерфейс пользователя ВП. Она создается с использованием палитры **Элементов (Controls)**. Эти элементы могут быть либо средствами ввода данных – элементы **Управления**, либо средствами отображения данных – элементы **Отображения**. Элементы **Управления** – кнопки, переключатели, ползунки и другие элементы ввода. Элементы **Отображения** – графики, цифровые табло, светодиоды и т.д. 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**.

Ниже приведена последовательность действий для создания ВП, который будет преобразовывать значение температуры из градусов Цельсия в градусы Фаренгейта.

#### Лицевая панель

1. Выберите пункт главного меню **File»New»VI**, чтобы открыть новую лицевую панель.



2. (Дополнительно) Выбрать пункт главного меню **Window»Tile Left and Right** для вывода на экран

рядом друг с другом лицевой панели и блок-диаграммы.

3. Создайте цифровой элемент управления. Он будет использован для ввода значений температуры в °С.

- a. Выберите цифровой элемент управления в разделе палитры **Элементов** в подразделе **Controls»Numeric** (Числовые элементы). Для вывода на экран палитры **Controls** (Элементов) следует щелкнуть правой кнопкой мыши по рабочему пространству лицевой панели.



- b. Поместите цифровой элемент управления на лицевую панель.



- c. 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.



4. Создайте цифровой элемент отображения данных. Он будет использован для отображения значений температуры в °F.



- a. Выберите цифровой элемент отображения в палитре **Элементов** в подразделе **Controls»Numeric** (Числовые элементы).
- b. Поместите элемент отображения данных на лицевую панель.
- c. В поле собственной метки элемента управления напечатайте «Град F» и щелкните мышью в свободном пространстве лицевой панели или нажмите кнопку **Enter**.



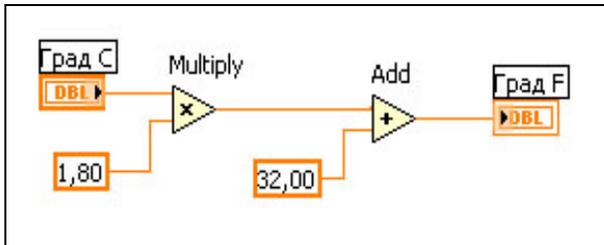
На блок-диаграмме LabVIEW создаст терминалы данных, соответствующие элементам управления и отображения. Терминалы данных представляют тип данных соответствующих элементов. Например, терминал данных DBL, показанный слева, представляет тип числовых данных двойной точности с плавающей запятой.



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

## Блок-диаграмма

5. Перейдите на блок-диаграмму, выбрав пункты главного меню **Window» Show Diagram**.



6. Выберите функцию **Multiply** (Умножение) из палитры **Функций** в разделе **Functions»Numeric** (Арифметические функции). Поместите ее на блок-диаграмму.



7. Выберите функцию **Add** (Сложение) из палитры **Функций** в разделе **Functions»Numeric** (Арифметические функции). Поместите ее на блок-диаграмму.



8. Выберите числовую константу из палитры **Функций** в разделе **Functions»Numeric** (Арифметические функции). Поместите 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 32,0.
- Если значение в константу не введено сразу после ее размещения на блок-диаграмме, следует использовать инструмент ВВОД ТЕКСТА.



9. Соедините объекты блок-диаграммы с помощью инструмента СОЕДИНЕНИЕ, показанного слева.
10. Перейдите на лицевую панель, выбрав в главном меню пункт **Window»Show Panel**.
11. Сохраните ВП, он будет использоваться позднее.
  - a. Выберите пункт главного меню **File»Save**.
  - b. Укажите каталог *C:\Введение в LabVIEW*.
  - c. В диалоговом окне введите *Преобразование C в F (начало).vi*
  - d. Нажмите кнопку **Save**.

## Запуск ВП



12. Введите число в элемент управления и запустите ВП.
  - a. Для ввода числа в элемент управления следует использовать инструмент УПРАВЛЕНИЕ, показанный слева, или инструмент ВВОД ТЕКСТА.
  - b. Нажмите кнопку **Run**, показанную слева, чтобы запустить ВП.
  - c. Введите несколько разных значений температуры и запустите ВП.

13. Закройте ВП, выбрав пункт главного меню **File»Close**.

### Создать иконку и настроить соединительную панель для возможности использования ВП в качестве подпрограммы ВП

Представлена последовательность действий по созданию иконки и настройке соединительной панели для созданного ВП, который переводит значение измеренной температуры из °C в °F.

#### Лицевая панель

1. Выберите пункт главного меню **File»Open**, укажите папку *C:\Введение в LabVIEW* и выберите файл *Преобразование C в F (начало).vi*

Если закрыты все ВП, следует нажать кнопку **Open VI** (Открыть ВП) в диалоговом окне **LabVIEW**.

Появится следующая лицевая панель:



#### Иконка и соединительная панель

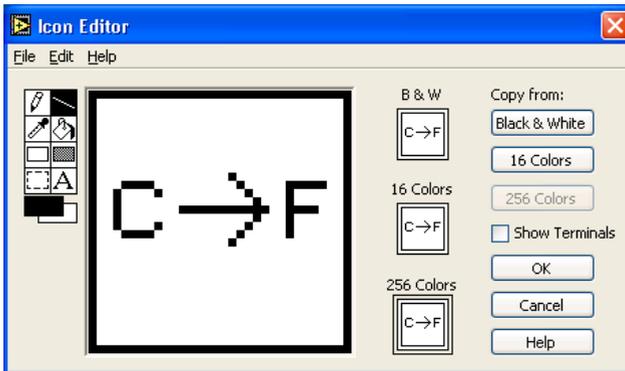
2. Щелкните правой кнопкой мыши по иконке ВП и в контекстном меню выберите пункт **Edit Icon** (Редактирование иконки). Появится диалоговое окно редактора иконки **Icon Editor**.



3. Дважды щелкните правой кнопкой мыши по инструменту **ВЫБОР** (показан слева).
4. Нажав кнопку **<Delete>**, очистите область редактирования иконки.



5. Дважды щелкните по инструменту ПРЯМОУГОЛЬНИК (показан слева), чтобы обвести область редактирования границей выбранного цвета.
6. Создайте следующую иконку:



- a. Введите текст инструментом ВВОД ТЕКСТА, который показан слева.
- b. Напечатайте «C» и «F».
- c. Для выбора размера шрифта дважды щелкните левой кнопкой мыши по инструменту ВВОД ТЕКСТА.



- d. Чтобы нарисовать стрелку, воспользуйтесь инструментом КАРАНДАШ.



**Внимание.** Для рисования вертикальных, горизонтальных и диагональных линий требуется во время рисования нажать и удерживать клавишу <Shift>.

- e. Для передвижения текста и стрелки по полю редактирования иконки используйте инструмент ВЫБОР и стрелки на клавиатуре.
- f. В разделе **Copy from** (Копировать) выберите **B & W** (черно-белую) иконку и **256 Colors** (256-цветный режим) для создания черно-

белой иконки, которую LabVIEW использует в случае отсутствия цветного принтера.

- g. В разделе **Copy from** (Копировать) выберите **16 Colors** и **256 Colors**.
- h. После завершения редактирования иконки нажмите кнопку **ОК** и закройте **Icon Editor**. Новая иконка появится в правом верхнем углу обеих панелей.



- 7. Перейдите на лицевую панель, щелкните правой кнопкой мыши на иконке и выберите пункт **Show Connector** (Показать поля ввода/вывода данных) из контекстного меню. Количество отображаемых LabVIEW полей ввода/вывода данных соответствует количеству элементов на лицевой панели. Например, лицевая панель этого ВП имеет два элемента **Град С** и **Град F** и LabVIEW выводит в соединительной панели два поля, показанные слева.
- 8. Элементам управления и отображения данных назначьте соответственно поля ввода и вывода данных.
  - a. В пункте главного меню **Help** (Помощь) выберите **Show Context Help** (контекстную подсказку) и выведите на экран окно **Context Help** (контекстной справки) для просмотра соединений.
  - b. Щелкните левой кнопкой мышки на левом поле соединительной панели. Инструмент УПРАВЛЕНИЕ автоматически поменяется на инструмент СОЕДИНЕНИЕ, а выбранное поле окрасится в черный цвет.
  - c. Щелкните левой кнопкой мыши по элементу **Град С**. Левое поле станет оранжевым и выделится маркером.
  - d. Щелкните курсором по свободному пространству. Маркер исчезнет, и поле

- окрасится в цвет данных типа соответствующего элемента управления.
- e. Щелкните левой кнопкой мыши по правому полю соединительной панели и элементу **Град F**. Правое поле станет оранжевым.
  - f. Щелкните курсором по свободному пространству. Оба поля останутся оранжевыми.
  - g. Наведите курсор на область полей ввода/вывода данных. Окно **Context Help** (контекстной справки) покажет, что оба поля соответствуют типу данных двойной точности с плавающей запятой.
9. Выберите пункт главного меню **File»Save**. Сохраните ВП под именем *Преобразование C в F.vi*, он будет использоваться позднее.
  10. Выберите пункт главного меню **File»Close**. Закройте ВП.

## Задания для самостоятельного выполнения

### Задание 1.

Рассчитать интенсивность излучения абсолютно черного тела по формуле:  $I = \sigma T^4$ , где  $T$  - температура в [эВ], а  $\sigma = 1,03 \cdot 10^5$  [Вт/см<sup>2</sup>·эВ<sup>4</sup>]. Создать ВП и сохранить под именем *Светимость черного тела.vi*.

### Задание 2.

Рассчитать мощности тормозного излучения, испускаемого из единицы объема полностью ионизованной водородной плазмы по формуле:  $Q[\text{Вт/см}^3 \cdot \text{сек}] = 1,69 \cdot 10^{-32} \cdot n^2 [\text{см}^{-3}] \cdot T[\text{эВ}]^{1/2}$ , где  $n$ ,  $T$  – плотность и температура электронов плазмы соответственно. Создать ВП и сохранить под именем *Q тормозное.vi*.

**Задание 3.**

Вычислить длину волны (в ангстремах) отсечки лазерного излучения в плазме по формуле:

$\lambda = 2\pi/\omega$ ,  $\omega^2 = 4\pi e^2 n_e / m_e$ , где  $e = 4,8 \cdot 10^{-10}$  [СГСЭ],  
 $m_e = 9,1 \cdot 10^{-28}$  [г] – заряд и масса электрона соответственно.

**Конец упражнения**

---

## Раздел 2. Циклы и условные операторы

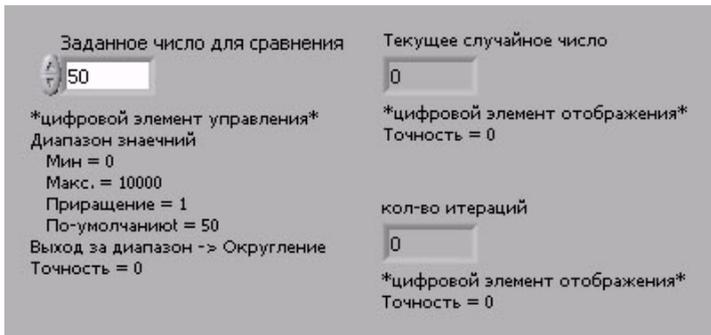
### ВП Подсчет итераций

**Цель:** использование терминала выходных данных цикла **While**

Создайте ВП, который генерирует случайные числа до тех пор, пока одно из них не окажется равным значению, введенному в элемент управления. При этом должно отображаться количество итераций, выполненное циклом.

#### Лицевая панель

1. Откройте новую лицевую панель. Создайте лицевую панель, разместив на ней элементы управления и отображения, как показано ниже на рисунке.



- a. Поместите на лицевую панель числовой элемент управления, находящийся на палитре **Controls»Numeric**. Назовите элемент **Заданное число для сравнения**. Этот элемент задает число, с которым будет проводиться сравнение.
- b. Поместите на лицевую панель числовой элемент отображения, находящийся на палитре **Controls»Numeric**. Назовите элемент **Текущее**

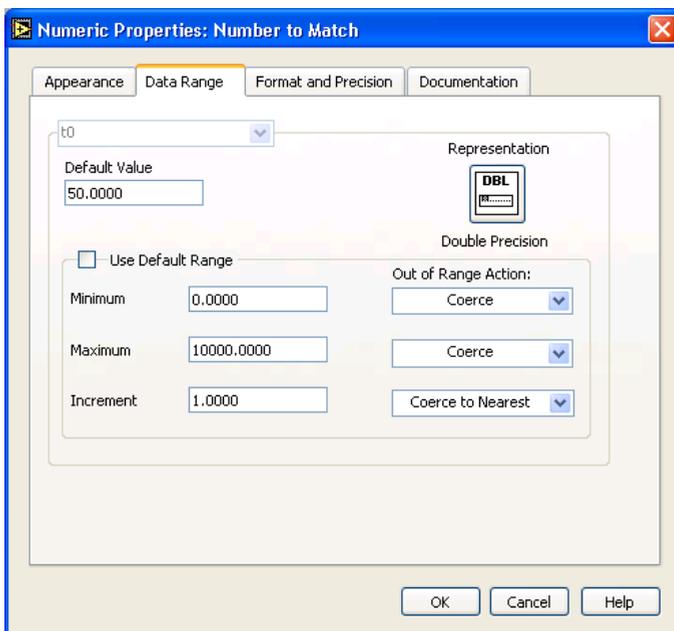
**случайное число.** Этот элемент отображает текущее значение, выданное функцией **Генератор случайных чисел.**

- c. Поместите еще один числовой элемент отображения на лицевую панель. Назовите элемент **Кол-во итераций.** Этот элемент показывает номер текущей итерации.

#### **Установка диапазона данных**

Чтобы значения элемента **Заданное число для сравнения** не выходили за рамки диапазона значений, выдаваемых функцией **Генератор случайных чисел**, следует использовать диалоговое окно **Data Range**. Выполните следующие шаги для настройки диапазона выходных значений элемента **Заданное число для сравнения** от 0 до 10 000 с шагом изменения 1 и значением, по умолчанию равным 50.

2. Щелкните правой кнопкой мыши по элементу **Заданное число для сравнения.** Из контекстного меню выберите пункт **Data Range**. Появится диалоговое окно, показанное ниже.
3. Снимите выделение с пункта **Use Defaults** (использовать значения по умолчанию).
4. Выберите пункты, показанные в этом примере диалогового окна:



- a. Установите **Default Value** (значение по умолчанию) равным 50.
  - b. Установите **Minimum Value** (минимальное значение) равным 0 и выберите **Coerce**.
  - c. Установите **Maximum Value** (максимальное значение) равным 10 000 и выберите **Coerce**.
  - d. Установите **Increment** (значение приращения) равным 1 и выберите **Coerce to Nearest**.
5. Выберите раздел **Format and Precision** (формат и точность).

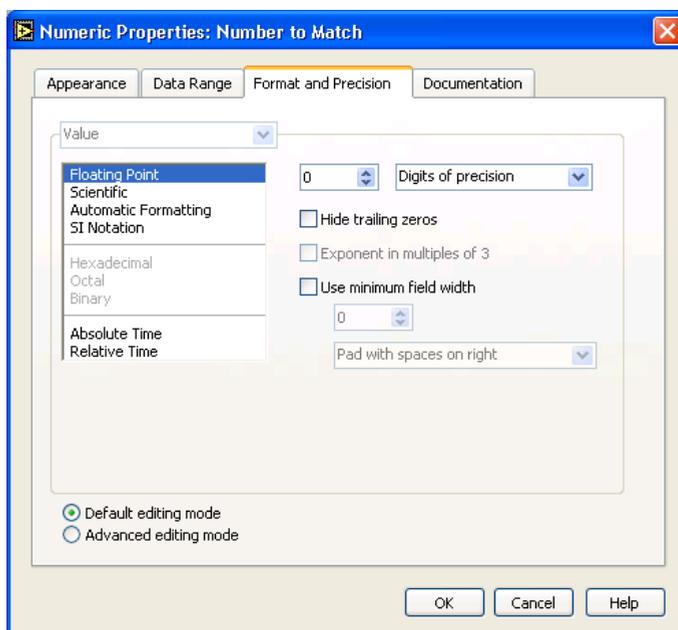
#### Установка количества знаков после запятой

По умолчанию LabVIEW отображает числовые элементы управления и отображения в виде десятичных чисел с точностью до двух знаков после запятой (3,14). С помощью опции **Format&Precision** можно изменить точность и вид представления значений элементов

(научная нотация, инженерная нотация, формат времени).

6. Щелкните правой кнопкой мыши по элементу **Текущее случайное число** и выберите в контекстном меню пункт **Format&Precision**. Появится следующее диалоговое окно **Format&Precision**.

7. Сделайте настройки, показанные ниже.

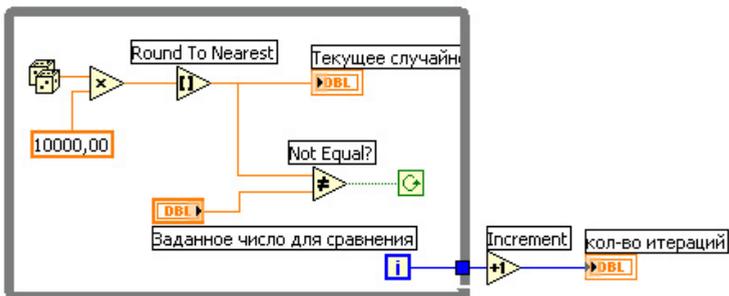


В поле ввода **Digits of Precision** следует ввести значение 0.

8. Повторите шаги 6 и 7 для элементов отображения **Текущее случайное число** и **Кол-во итераций**.

## Блок-диаграмма

9. Создайте блок-диаграмму, как показано на рисунке.



Поместите на блок-диаграмму функцию **Random Number** (Генератор случайных чисел), расположенную на палитре **Функций** в разделе **Function»Numeric**. Эта функция генерирует случайные числа в пределах от 0 до 1.



Поместите на блок-диаграмму функцию **Multiply** (Умножение), расположенную в палитре **Функций** в разделе **Function»Numeric**. Эта функция умножает текущее значение с выхода функции **Random Number** (Генератор функции случайных чисел) на 10 000.

10000.00

Создайте константу. Для этого следует навести курсор на поле ввода данных функции **Multiply** (Умножение), щелкнуть по нему правой кнопкой мыши и выбрать в контекстном меню пункт **Create»Constant**. С помощью инструмента **ВВОД ТЕКСТА** присвойте ей значение 10 000.



Поместите на блок-диаграмму функцию **Round To Nearest** (Округление до ближайшего целого), расположенную в палитре **Функций** в разделе **Function»Numeric**. Эта функция будет округлять полученное в пределах от 0 до 10000 случайное число до ближайшего целого числа.



Поместите на блок-диаграмму функцию **Not Equal? (≠)**, расположенную в палитре **Функций** в разделе **Function»Comparison**. Эта функция предназначена для

сравнения случайного числа с числом, введенным в элемент управления **Заданное число для сравнения**. Если значения не равны, функция выдает значение TRUE.



Поместите на блок-диаграмму цикл **While**, расположенный в палитре **Функций** в разделе **Function»Structures**. Наведите курсор на терминал условия выхода, щелкните по нему правой кнопкой мыши и выберите пункт **Continue if True** (Продолжение если Истина).



Подсоедините терминал счетчика итераций к границе области цикла **While**. На границе цикла появится синий прямоугольник. Терминал выходных данных цикла присоединен к функции приращения. При выполнении цикла счетчик итераций получает приращение, равное 1. После завершения цикла значение счетчика итераций передается на выход через терминал выхода цикла. Вне тела цикла значение счетчика итераций увеличивается на единицу для отображения количества выполненных итераций.



Поместите на блок-диаграмму функцию **Increment** (Приращение на 1), расположенную в палитре **Функций** в разделе **Function»Numeric**. Эта функция добавляет 1 к значению счетчика итераций после завершения выполнения цикла. Следует обратить внимание, что на терминале элемента **Кол-во итераций** имеется серая точка, означающая, что LabVIEW автоматически осуществляет преобразование типа данных счетчика итераций к типу данных терминала элемента **Кол-во итераций**. Подробнее о приведении типов данных можно прочитать в разделе **В "Цикл For (с фиксированным числом итераций)"**.

10. Сохраните ВП под именем *Подсчет итераций.vi*

**Запуск  
ВП**

---

11. Перейдите на лицевую панель и измените значение элемента **Заданное число для сравнения**.
12. Запустите ВП. Измените значение элемента **Заданное число для сравнения** и запустите ВП снова. При этом элемент **Текущее случайное число** обновляется после каждой итерации цикла, потому что его терминал данных расположен внутри тела цикла. Значение же элемента **Кол-во итераций** обновляется после завершения цикла, потому что терминал данных этого элемента расположен вне тела цикла.
-  13. Чтобы посмотреть, как ВП обновляет значения элементов отображения информации, необходимо запустить ВП в режиме анимации. Для этого следует нажать на инструментальной панели кнопку **Highlight Execution**, показанную слева. Режим отладки анимирует поток данных, проходящих по блок-диаграмме. Таким образом, имеется возможность наблюдать изменения значений на каждом этапе их генерации.
14. Измените значение элемента **Заданное число для сравнения** таким образом, чтобы оно с увеличением на 1 выходило за установленный диапазон значений от 0 до 10 000.
15. Запустите ВП. LabVIEW автоматически приведет новое значение к ближайшему значению в указанном диапазоне входных данных элемента.
16. Закройте ВП.

**Конец упражнения**

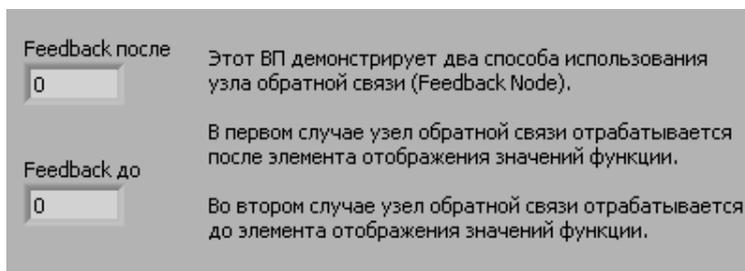
---

## Доступ к данным предыдущих итераций

**Цель:** Использование сдвиговых регистров в цикле For для суммирования элементов массива

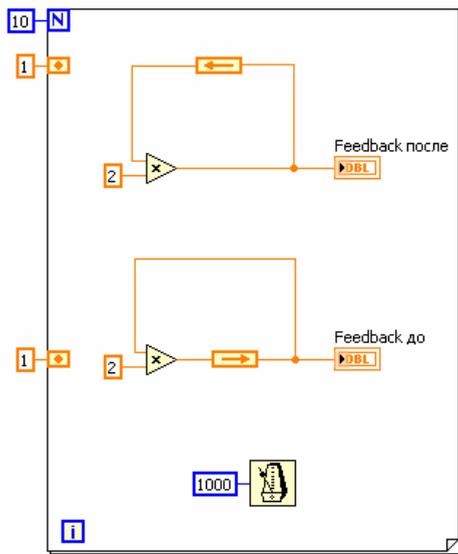
### Лицевая панель

1. Откройте ВП Узел обратной связи VI, находящийся в каталоге *C:\Введение в LabVIEW*. Лицевая панель этого ВП, показанная ниже, уже создана.



### Блок-диаграмма

2. Отобразите блок-диаграмму, показанную ниже. Разместите лицевую панель и блок-диаграмму на экране так, чтобы они не перекрывали друг друга. Переместите палитры **Tools** и **Functions**, если это необходимо.



Единица, соединенная с левым терминалом цикла **For**, инициализирует узел обратной связи. Таймер **Wait Until Next ms Timer** формирует миллисекундную задержку, замедляя процесс выполнения программы. Для замедления выполнения процесса можно также использовать режим **Highlight Execution** (анимации выполнения блок-диаграммы).

На данной блок-диаграмме один и тот же процесс выполняется дважды, при этом узел обратной связи помещен в различных местах соединения.

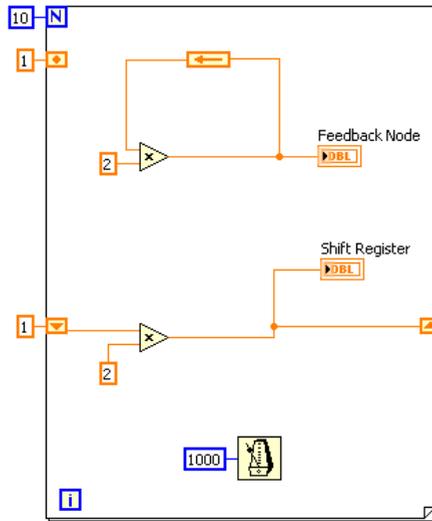
## Запуск ВП

3. Запустите ВП. Программа в верхней части сначала считывает значение узла обратной связи, инициализированного значением 1. Затем это значение передается функции **Multiply** (умножение).

Программа в нижней части сначала считывает значения узла обратной связи, инициализированного значением 1. Затем это значение передается на цифровой элемент отображения. Функция **Multiply** (умножение) не будет выполняться до следующей итерации цикла.



4. Активируйте режим анимации выполнения блок-диаграммы, нажав на показанную слева кнопку **Highlight Execution**. Запустите ВП еще раз для наблюдения порядка выполнения программы. Отключите режим анимации для работы ВП в нормальном режиме.
5. Замените узел обратной связи сдвиговым регистром, как показано на следующей блок-диаграмме:

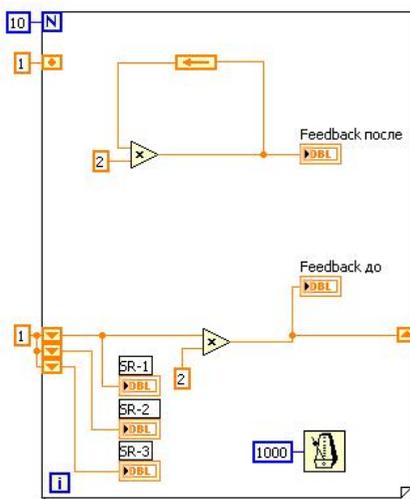


- a. Выделите нижний узел обратной связи и нажмите клавишу <Delete>, чтобы удалить его.

- b. Щелкните правой кнопкой мыши по границе цикла и выберите пункт контекстного меню **Add Shift Register**.
  - c. Инициализируйте сдвиговый регистр значением 1.
  - d. Переименуйте верхний и нижний элементы отображения соответственно **Узел обратной связи** и **Сдвиговый регистр**.
6. Запустите ВП. Обратите внимание, что узел обратной связи и сдвиговый регистр выполняют одинаковые функции.
  7. Не закрывайте ВП, перейдите к выполнению дополнительных упражнений или закройте ВП, не сохраняя его.

### Дополнительно

8. Модифицируйте сдвиговый регистр, чтобы он отображал последние три итерации цикла **For**, как показано на следующей блок-диаграмме:



- a. Измените размер левого сдвигового регистра до трех элементов.
  - b. Инициализируйте все элементы сдвигового регистра значением 1.
  - c. Щелкните правой кнопкой мыши по каждому элементу и выберите пункт контекстного меню **Create»Indicator**. Назовите каждый элемент отображения.
9. Запустите ВП.
  10. Закройте ВП, не сохраняя его.

## Конец упражнения

---

## ВП Извлечение квадратного корня

### Цель: Изучить структуру Case

Выполните следующие шаги для построения ВП, который проверяет входное число на знак, вычисляет его квадратный корень или выдает сообщение об ошибке, если число отрицательное.

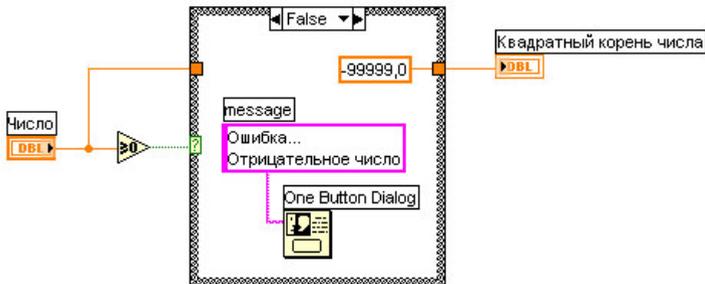
### Лицевая панель

1. Откройте новый ВП и создайте лицевую панель, как показано ниже.



### Блок-диаграмма

2. Создайте блок-диаграмму, показанную ниже на рисунке.



Поместите на блок-диаграмму структуру **Case**, расположенную в палитре **Functions>>Structures**.

Используйте стрелки уменьшения или увеличения селектора структуры для выбора варианта **FALSE**.



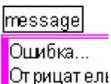
Поместите на блок-диаграмму функцию **Greater or Equal to 0?**, расположенную в палитре **Functions>>Comparison**. Функция возвращает значение **TRUE**, если число больше или равно 0.



Щелкните правой кнопкой мыши по численной константе и в контекстном меню выберите пункт **Format & Precision**. Установите **Digits of Precision** равным 1, выберите вид представления **Floating Point Notation** и нажмите кнопку **OK**.



Поместите на блок-диаграмму **One Button Dialog**, расположенный в палитре **Functions>>Time & Dialog**. Это диалоговое окно будет отображать сообщение «Ошибка... Отрицательное число».

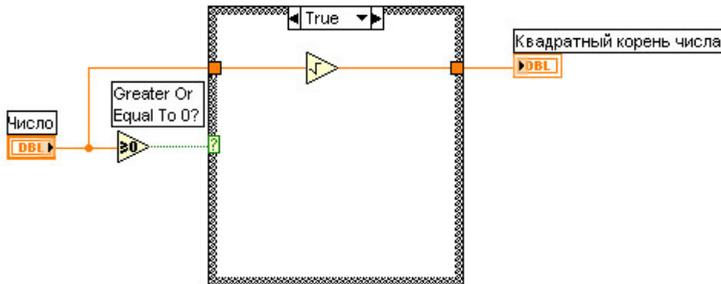


Щелкните правой кнопкой мыши по полю ввода/вывода **message** функции **One Button Dialog** и в контекстном меню выберите пункт **Create>>Constant**. Введите текст «Ошибка... Отрицательное число». Для получения более подробной информации смотрите Урок 9, Строки и файловый ввод/вывод.

3. Выберите вариант **TRUE**.



Поместите функцию **Square Root** на блок-диаграмму, как показано ниже. Функция размещена в палитре **Functions>>Numeric**. Она возвращает квадратный корень входного числа.



4. Сохраните созданный ВП под именем *Извлечение квадратного корня.vi*

## Запуск ВП

5. Отобразите лицевую панель и запустите ВП.  
Если входное значение элемента управления **Число** положительно, то выполняется поддиаграмма варианта TRUE и вычисляется значение квадратного корня. Если значение элемента **Число** является отрицательным, то выполняется поддиаграмма варианта FALSE, которая возвращает  $-99999,0$  и отображает диалоговое окно с сообщением «Ошибка... Отрицательное число».
6. Закройте ВП.

## Конец упражнения

---

## Раздел 3. Функции работы с массивами

---

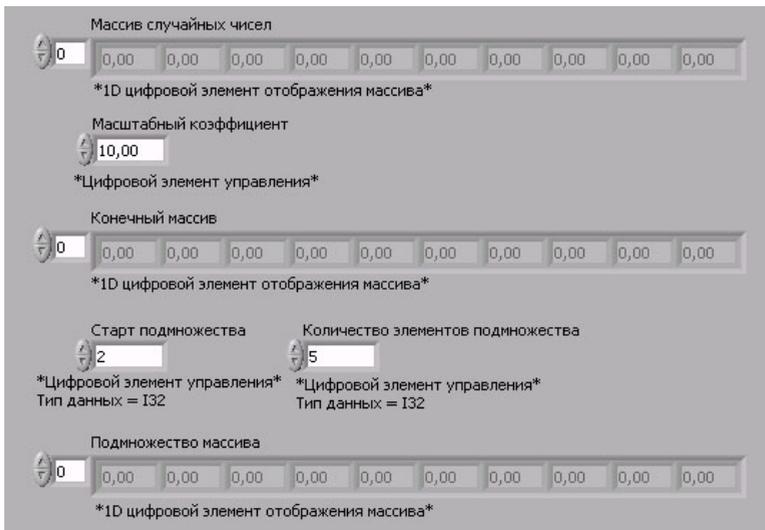
### ВП Подмножество массива

**Цель:** Создание массивов и знакомство с функциями обработки массивов

Выполните следующие шаги для создания ВП, который формирует массив случайных чисел, масштабирует полученный массив и выделяет из него подмножество.

#### Лицевая панель

1. Откройте новый ВП и создайте лицевую панель, как показано ниже.



- a. В палитре **Controls>>Array & Cluster** выберите шаблон массива.
- b. Созданному массиву присвойте имя **Массив случайных чисел**.

- c. Поместите внутрь шаблона массива цифровой элемент отображения, расположенный в палитре **Controls>>Numeric**.
- d. С помощью инструмента ПЕРЕМЕЩЕНИЕ измените размер массива таким образом, чтобы он содержал 10 элементов.
- e. Нажмите и удерживайте клавишу <Ctrl> и, перемещая элемент **Массив случайных чисел**, создать две его копии.
- f. Копиям присвойте имена **Конечный массив** и **Подмножество Массива**.
- g. Создайте три цифровых элемента управления и присвойте им имена **Масштабный коэффициент**, **Старт подмножества** и **Количество элементов подмножества**.
- h. Щелкните правой кнопкой мыши по элементам **Старт подмножества** и **Количество элементов подмножества**, в контекстном меню выберите пункт **Representation**, затем пункт **I32**.
- i. Значения элементов управления данных пока не изменяйте.

## Блок-диаграмма

2. Постройте блок-диаграмму, как показано ниже.



Выберите функцию **Random Number (0-1)**, расположенную в палитре **Functions>>Numeric**. Эта функция будет генерировать случайное число в пределах от 0 до 1.



Выберите цикл **For**, расположенный в палитре **Functions>>Structures**. Этот цикл на терминале выхода накапливает массив из 10 случайных чисел. Терминалу количества итераций присвойте значение 10.



Выберите функцию **Array Subset**, расположенную в палитре **Functions>>Array**. Эта функция выдает подмножество массива, начиная со значения, введенного в элементе **Старт подмножества** и будет содержать количество элементов, указанное в элементе **Количество элементов подмножества**.

3. Сохраните ВП под именем *Подмножество массива.vi*

## Запуск ВП

4. Перейдите на лицевую панель, измените значения элементов управления и запустите ВП.

Цикл **For** совершит 10 итераций. Каждая итерация создаст случайное число и сохранит его в терминале выхода из цикла. В элементе **Массив случайных чисел** отобразится массив из 10 случайных чисел. ВП умножит каждое значение этого массива на число, введенное в элемент управления **Масштабный коэффициент**, для создания массива, отображаемого в индикаторе **Конечный массив**. ВП выделит подмножество из получившегося массива, начиная со значения в элементе **Старт подмножества**, длиной, указанной в элементе **Количество элементов подмножества**, и отобразит это подмножество в индикаторе **Подмножество массива**.

5. Закройте ВП.

## Конец упражнения

---

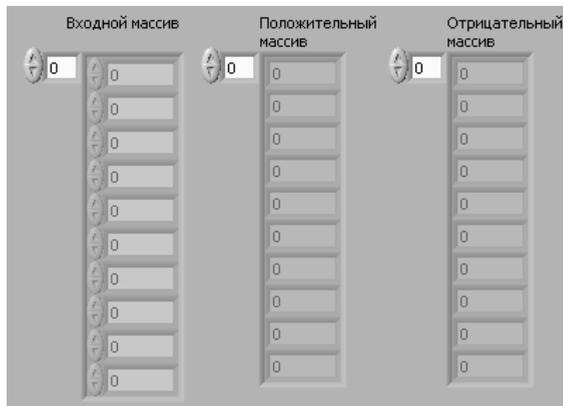
## ВП Сортировка массива

**Цель:** использование цикла For и структуры Case для обработки массива

Ниже приведена последовательность действий для создания ВП, который сортирует входной массив, состоящий из положительных и отрицательных чисел, на два отдельных массива (массив положительных и массив отрицательных чисел) с использованием свойства автоиндексации.

### Лицевая панель

1. Откройте новый ВП и создайте лицевую панель, как показано ниже на рисунке.



- j. В палитре **Controls>>Array & Cluster** выберите шаблон массива.
- k. Созданному массиву присвойте имя **Входной массив**.
- l. Поместите внутрь шаблона массива цифровой элемент отображения, расположенный в палитре **Controls>>Numeric Control**.





Выберите цикл **For**, расположенный в палитре **Functions>>Structures**. Создайте два сдвиговых регистра. Для этого щелкните правой кнопкой мыши по границе цикла и выберите пункт контекстного меню **Add Shift Register**.



Поместите внутрь цикла **For** структуру **Case**, расположенную в палитре **Functions>>Structures**.



Выберите функцию **Build Array**, расположенную в палитре **Functions>>Array**. Эта функция предназначена для построения массива.



Поместите на блок-диаграмму функцию **Greater Than 0**, находящуюся в палитре **Functions>>Comparison**.



Выберите функцию **Initialize Array**, расположенную в палитре **Functions>>Array**. Эта функция создает  $n$ -мерный массив, в котором каждый элемент инициализирован значением поля ввода данных **element**.



Определяет значения элементов массива и его тип.



Определяет размерность массива.

3. Сохраните ВП под именем *Сортировка массива.vi*.
4. Запустите ВП.
5. Закройте ВП.

## Конец упражнения

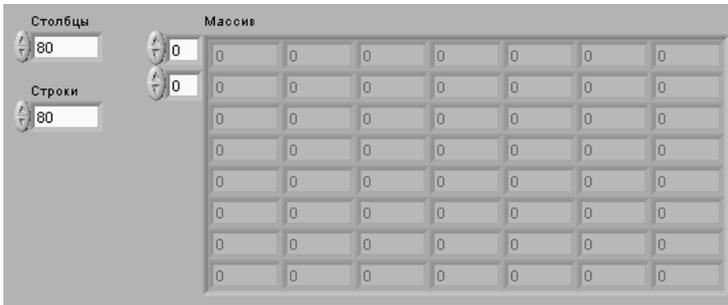
---

### ВП Работа с массивами

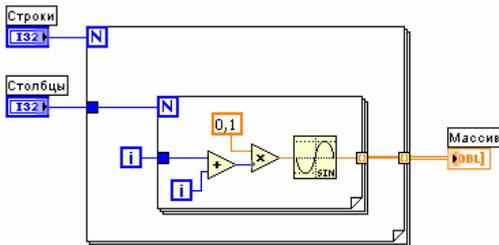
Цель: научиться работать с массивами, используя различные функции LabVIEW

Вам дано задание расширить функции существующего ВП.

Каждую часть этого упражнения начинайте с ВП *Изучение массива.vi*, расположенного в каталоге *C:\Введение в LabVIEW*. Лицевая панель этого ВП показана ниже.



Соответствующая блок-диаграмма показана ниже.



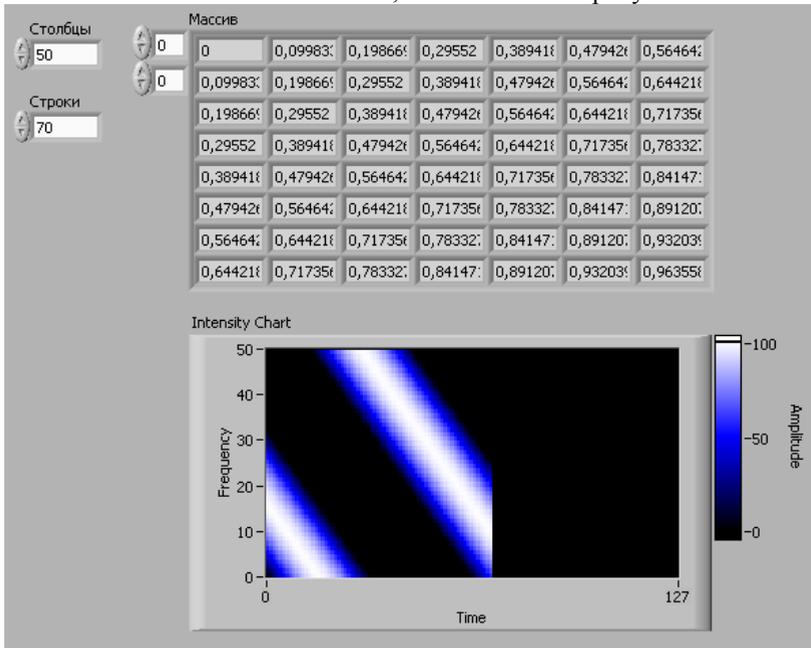
В каждой части этого упражнения дается сценарий. Подробные инструкции для всех частей приведены в конце упражнения.

## Часть 1: реализация

Требуется изменить ВП *Изучение массива* так, чтобы созданный массив вводился в цикл **For**, где каждый элемент умножался бы на 100 и округлялся до ближайшего целого числа. Полученный двумерный массив нужно отобразить на диаграмме интенсивности.

## Лицевая панель

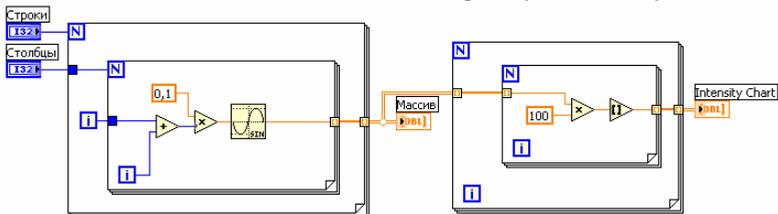
1. Откройте *Изучение массива.vi* из каталога *C:\Введение в LabVIEW*.
2. Сохраните ВП как *Изучение массива (часть 1).vi*.
3. Поместите на лицевую панель диаграмму интенсивности, как показано на рисунке.



4. Откройте блок-диаграмму ВП.

## Блок-диаграмма

5. Создайте блок-диаграмму, показанную ниже.





Выберите цикл **For**, расположенный в палитре **Functions>>Structures**. Поместите второй цикл **For** внутри первого. Соедините терминал индикатора **Массив** с границей внутреннего цикла **For**. В результате будут созданы автоиндексируемые входные туннели в обоих циклах.



Поместите внутрь цикла **For** функцию умножения, соедините индексированный туннель с терминалом  $x$  функции умножения. Щелкните правой кнопкой на терминале  $y$  и выберите в контекстном меню пункт **Create>>Constant**. Введите в константу число 100.



Поместите на блок-диаграмму функцию **Round To Nearest**, соедините выход функции умножения со входом функции **Round To Nearest**.

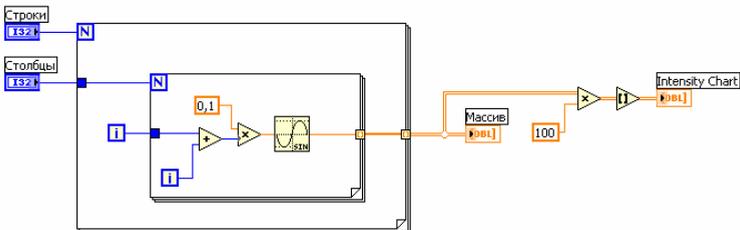
Соедините выход функции **Round To Nearest** с границей внешнего цикла **For**. В результате будут созданы автоиндексируемые выходные туннели в обоих циклах.

6. Перейдите на лицевую панель.
7. Сохраните ВП.
8. Запустите ВП.

## Часть 2: реализация

Требуется модифицировать ВП из части 1 так, чтобы тот же результат достигался без использования вложенных циклов **For**.

1. Откройте *Изучение массива (часть 1).vi*.
2. Сохраните ВП как *Изучение массива (часть 2).vi*.
3. Измените блок-диаграмму, как показано ниже.



4. Сохраните ВП.
5. Перейдите на лицевую панель.
6. Задайте количество строк и столбцов.
7. Запустите ВП.

Обратите внимание, что ВП работает точно так же, как в части 1.

**Конец упражнения**

---

## Раздел 4. Графическое представление данных

### График диаграмм

График Диаграмм (**Waveform Chart**) – специальный элемент отображения данных в виде одного и более графиков. График Диаграмм расположен на палитре **Controls>>Graph**.

График Диаграмм использует три различных режима отображения данных: **strip chart**, **scope chart** и **sweep chart**. Режим по умолчанию – **strip chart**.

Задание режима осуществляется щелчком правой клавишей мыши по диаграмме и выбором пункта **Advanced>>Update Mode** из контекстного меню.

Режим **strip chart** представляет собой экран, прокручиваемый слева направо, подобно бумажной ленте. Режимы **scope chart** и **sweep chart** подобны экрану осциллографа и отличаются большей скоростью отображения данных по сравнению с **strip chart**. В режиме **scope chart** по достижении правой границы поле графика очищается, и заполнение диаграммы начинается с левой границы. Режим **sweep chart**, в отличие от режима **scope chart**, не очищает поле графика, а отделяет новые данные от старых вертикальной линией – маркером.

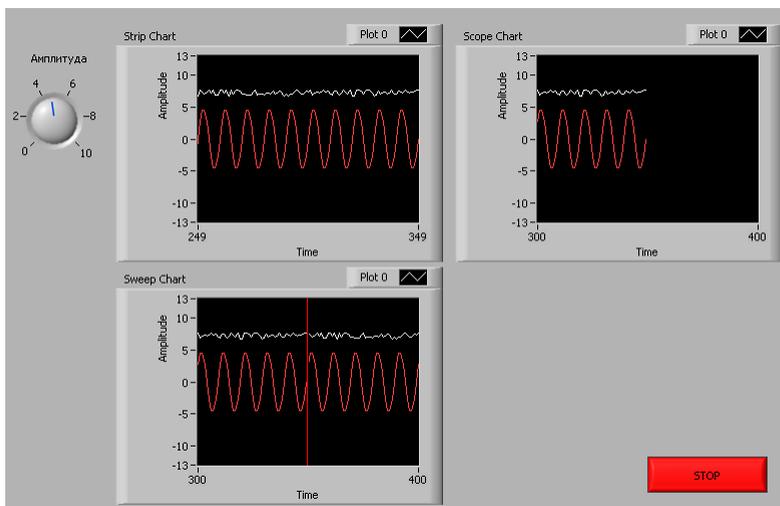
### ВП Отображение данных

#### Цель: знакомство с ленточным графиком **Waveform Chart**

ВП, приведенный в данном упражнении, демонстрирует вывод нескольких кривых на один график, используя режимы вывода данных **Strip Chart**, **Scope Chart** и **Sweep Chart**.

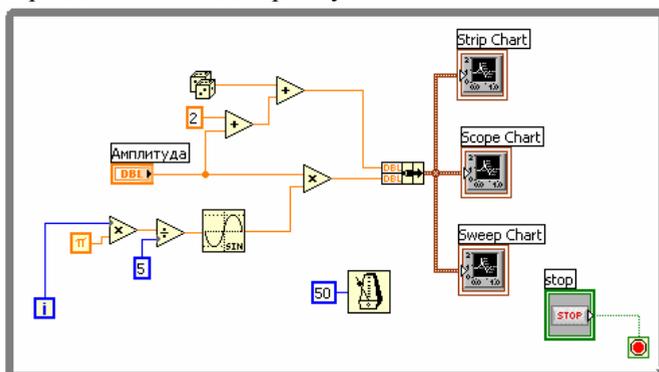
#### Лицевая панель

1. Откройте файл *C:\Введение в LabVIEW\Waveform Chart*.
2. Запустите ВП.

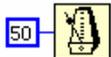


## Блок-диаграмма

Перейдите на блок-диаграмму:



Функция **Bundle** располагается в **Functions>>Cluster**. Эта функция объединяет данные функции синуса и генератора случайных чисел в кластер и выдает на графики **Waveform Chart**.



Осуществляет 50 мс задержку между выдачей данных на графики.

Закройте ВП.

**Конец упражнения**

---

## **Раздел 5. Запись и считывание файлов**

---

### **График осциллограмм и двухкоординатный график осциллограмм**

График осциллограмм (**Waveform Graph**) и двухкоординатный график осциллограмм (**X-Y Graph**) расположены на палитре **Controls>>Graph**. График Осциллограмм отображает только однозначные функции, такие как  $y = f(x)$ , с точками, равномерно распределенными по оси  $X$ . Двухкоординатный график Осциллограмм отображает любой набор точек, будь то равномерно распределенная выборка во времени или нет.

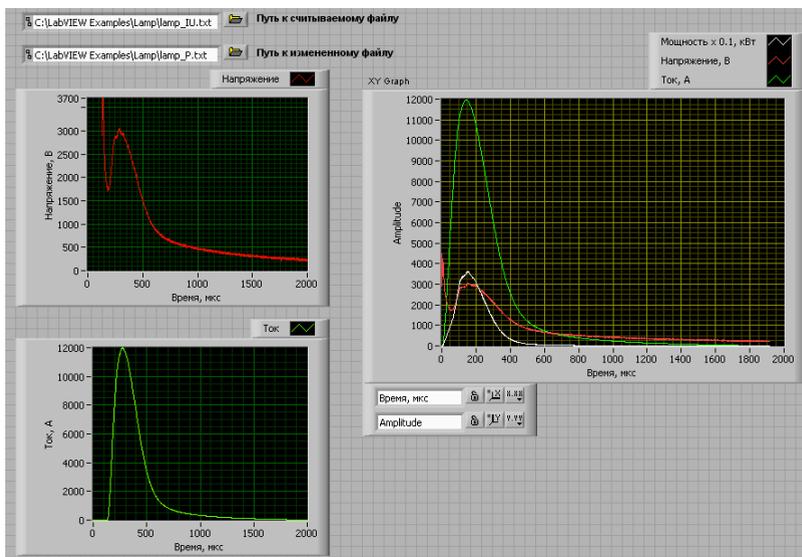
### **ВП Электрические параметры лампы**

**Цель: приобретение навыков по использованию графиков Waveform Graph и XY Graph для отображения нескольких кривых, а также чтение и запись файлов данных**

Ниже приведена последовательность действий для создания ВП, который осуществляет считывание данных тока и напряжения в лампе от времени, вычисляет изменение мощности, отображает их на графике и записывает результат в файл.

#### **Лицевая панель**

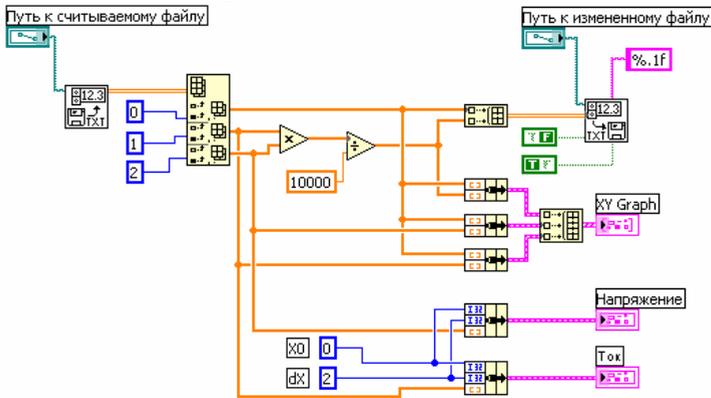
1. Откройте новый ВП и создайте лицевую панель, как показано ниже на рисунке.



- В палитре **Controls>>Graph** выберите **Waveform Graph** и создайте его копию.
- Присвойте одному название **Напряжение**, а другому **Ток**.
- В палитре **Controls>>Graph** выберите **XY Graph**.
- С помощью курсора растяните **Plot Legend**, пока он не отобразит три кривые.
- Каждой кривой присвойте имена **Мощность**, **Напряжение**, **Ток**.
- В палитре **Controls>>String & Path** выберите **File Path Control** и создайте его копию.
- Присвойте одному название **Путь к считываемому файлу**, а другому **Путь к измененному файлу**.

## Блок-диаграмма

2. Создайте блок-диаграмму, показанную ниже на рисунке.



ВП **Read From Spreadsheet File VI**, расположенный в палитре **Functions>>File I/O**, форматирует строку таблицы символов из файла в 2D массив.



Подсоедините элемент **Путь к считываемому файлу** ко входу **file path** ВП **Read From Spreadsheet File** и укажите путь *C:\Введение в LabVIEW \Lamp\lamp IU*.



С ВП **Read From Spreadsheet File** подайте массив на вход **ВП Index Array**. Выделите три 1D массива, первый из которых является разверткой по времени, а два других - ток и напряжение соответственно.



В палитре **Functions>>Cluster** выберите функцию **Bundle**. Эта функция объединяет 1D массив развертки по времени с 1D массивами тока, напряжения и мощности в кластеры.



Выберите функцию **Build Array**, расположенную в палитре **Functions>>Array**. Эта функция объединяет три кластера и передает данные на график отображения **XY Graph**.



Эта функция при работе с **Waveform Graph** позволяет задавать начальное положение первого элемента массива и интервал между элементами по горизонтальной оси, как





Выберите цикл **For**, расположенный в палитре **Functions>>Structures**. Задайте число итераций, равное десяти.



Выберите функцию **Write File**, расположенную в палитре **Functions>>File I/O** и поместите ее внутрь цикла **For**. Эта функция записывает объединенную строку в файл.



Поместите внутрь цикла **For** функцию **Format Into String**, расположенную в палитре **Functions>>String**, и увеличьте число входов до четырех. Выход функции **resulting string** подсоедините ко входу **data** функции **Write File**.



На первый вход подайте значение счетчика итераций цикла. На второй - константу **Tab constant**, расположенную в палитре **Functions>>String**. На третий - значение генератора случайных чисел и на четвертый - константу **End of Line constant**, расположенную в палитре **Functions>>String**.



Поместите на блок-диаграмму подпрограмму ВП **Open/Create/Replace File VI**, расположенную в палитре **Functions>>File I/O**. Этот ВП выводит на экран диалоговое окно для создания файла. Создайте и подсоедините ко входу **function** константу **open or create** и строковую константу «Enter filename» ко входу **prompt**.



В палитре **Functions>>File I/O** выберите функцию **Close File**. Эта функция закрывает файл.

### Обработка ошибок



Подпрограммы ВП и функции низкого уровня содержат информацию об ошибках. Для их обработки используются подпрограммы обработки ошибок, такие как **Simple Error Handler VI** (ВП *Простой обработчик ошибок*), расположенный в палитре **Functions>>Time & Dialog**. Поля ввода **error in** и вывода **error out**

информации об ошибках используются в каждом ВП для обмена информацией об ошибках между ВП.

## Запуск ВП

2. Перейдите на лицевую панель и запустите ВП. Появится диалоговое окно **Enter Filename**. В диалоговом окне введите имя файла *Random num.txt* и нажмите на кнопку **Save** или **OK**.
3. Откройте файл *Random num.txt* одним из текстовых редакторов **Windows**, например Notepad или WordPad.
4. Закройте текстовый редактор и вернитесь в LabVIEW.

**Конец упражнения**

---

## Раздел 6. Экспресс - виртуальные приборы

Экспресс-ВП — узлы функций, которые можно настраивать с помощью диалогового окна. Они используются для выполнения стандартных измерений при минимальных соединениях.

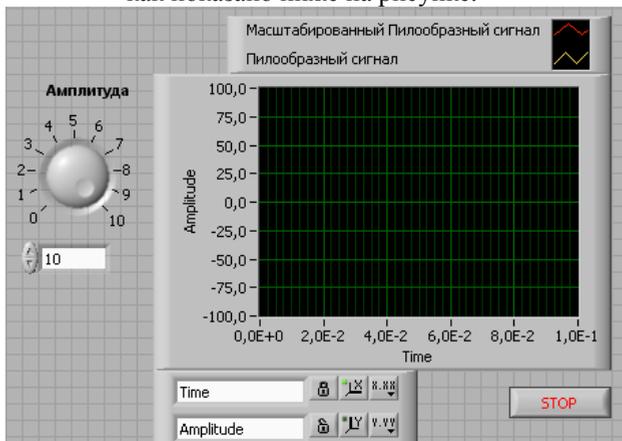
### ВП Масштабирование сигнала

**Цель:** знакомство с экспресс - виртуальными приборами

Ниже приведена последовательность действий для создания ВП, который осуществляет генерацию пилообразного сигнала, линейно преобразует значение его амплитуды и выводит полученный сигнал на графический индикатор **Waveform Graph**.

#### Лицевая панель

1. Откройте новый ВП и создайте лицевую панель, как показано ниже на рисунке.

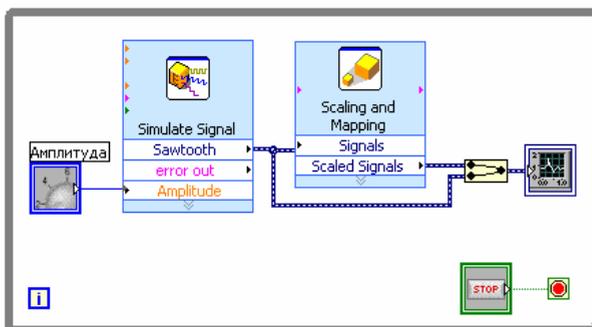


- a. В палитре **Controls>>Graph** выберите **Waveform Graph**.
- b. С помощью курсора растяните **Plot Legend**, пока он не отобразит две кривые.

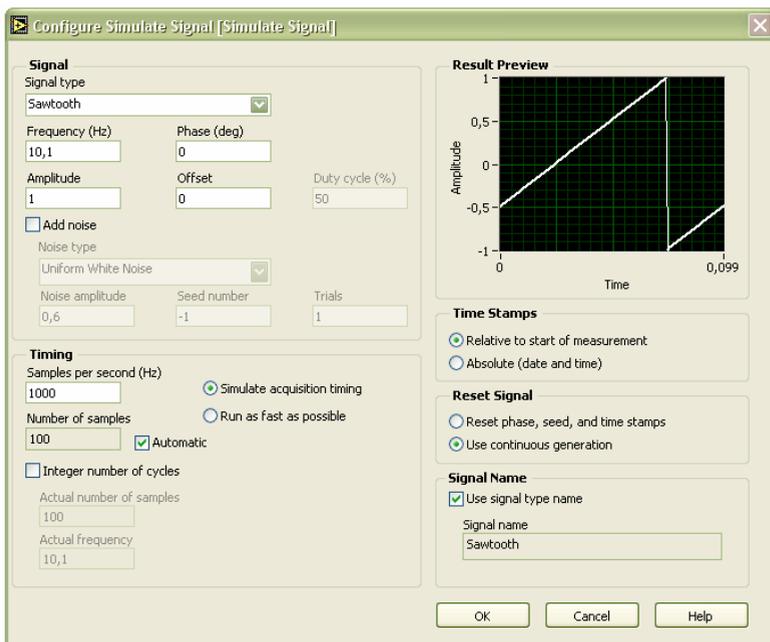
- c. Каждой кривой присвойте имена **Масштабированный Пилообразный сигнал** и **Пилообразный сигнал**.
- d. В контекстном меню графика выберите **Visible Items>>Scale Legend** и отмените привязку к оси Y.
- e. Поместите на лицевую панель элемент управления **Knob**, расположенный в **All Controls>> Numeric**. Присвойте ему название **Амплитуда**.
- f. Отобразите цифровой дисплей элемента управления **Knob** из контекстного меню **Visible Items>>Digital Display**.

## Блок-диаграмма

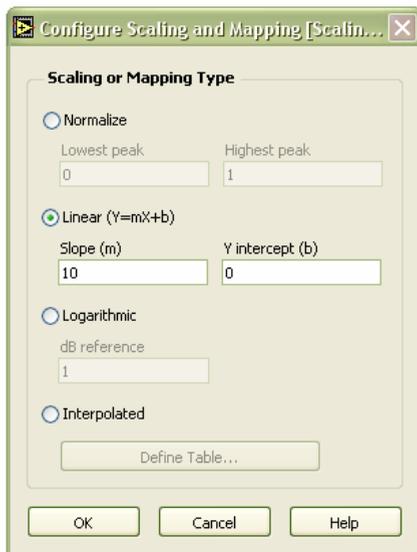
2. Создайте блок-диаграмму, показанную ниже на рисунке.



- a. В палитре **Functions>>Execution Control** выберите **While Loop**.
- b. Внутри цикла из палитры **Functions>>Input** поместите экспресс - ВП **Simulate Signal**.
- c. При помещении экспресс - ВП **Simulate Signal** на блок-диаграмму появляется диалоговое окно.



- d. В разделе **Signal** выберите тип сигнала **Sawtooth**.
- e. В разделе **Timing** выберите **Simulate acquisition timing**.
- f. Нажмите ОК.
- g. Добавьте вход **Amplitude** и подсоедините к нему элемент управления **Knob**.
- h. В палитре **Functions>>Arithmetic & Comparison** выберите экспресс - ВП **Scaling and Mapping**.
- i. При помещении экспресс - ВП **Simulate Signal** на блок-диаграмму появляется диалоговое окно.



- j. Задайте коэффициент  $m = 10$ .
- k. Нажмите ОК.
- l. Соедините выход **Simulate Signal** со входом **Scaling and Mapping**.
- m. Затем соедините выходы обоих экспресс - ВП со входом **Waveform Graph**.

### Запуск ВП

3. Перейдите на лицевую панель.
4. Сохраните ВП.
5. Запустите ВП.

### Конец упражнения

---

## Список литературы

1. *Загидуллин Р.Ш.* LabVIEW в исследования и разработках. – М.: Горячая линия-Телеком, 2005. – 352 с.: ил.
2. *Бутырин П.А., Васьковская Т.А., Каратаев В.В., Материкин С.В.* Автоматизация физических исследований и эксперимента: компьютерные измерения и виртуальные приборы на основе LabVIEW 7/ Под ред. Бутырина П.А. – М.: ДМК-Пресс, 2005. - 264 с.: ил.
3. *Пейч Л.И., Точилин Д.А., Поллак Б.П.* LabVIEW для новичков и специалистов. – М.: Горячая линия–Телеком, 2004. – 384с.: ил.
4. *Суранов А.Я.* LabVIEW 7: Справочник по функциям. – М.: ДМК-Пресс, 2005. – 512 с.