

Введение

A. LabVIEW и LabVIEW 7

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 данными с измерительными приборами и работе с другими программными средствами, появлялись соответственно в 1990, 1992, 1993, 1996 и 2000 годах. Последняя – седьмая версия (LabVIEW 7) была создана корпорацией NI в 2003 году. 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.

B. Цели учебного курса





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;

- созданию приложений, использующих последовательную связь и GPIB-интерфейс;
- созданию приложений, использующих встроенные DAQ устройства.

Г. Условные обозначения

В учебном курсе использованы следующие условные обозначения:

- » Символ " » " обозначает последовательность вызова пунктов вложенных меню или разделов диалоговых окон. Например, последовательность **File»Page Setup»Options** указывает, что сначала необходимо выбрать меню **File**, затем пункт меню **Page Setup** и далее раздел **Options** в диалоговом окне.
-  Иконка обозначает дополнительную справочную информацию.
-  Иконка обозначает важную информацию для запоминания.
-  Иконка обозначает предостережение от действий, которые могут привести к потере данных или сбоям операционной системы.
-  Иконка обозначает то, что выполняемое упражнение требует наличия встроенных аппаратных средств: GPIB-интерфейса или DAQ устройства.
- Жирный текст** Жирный текст выделяет названия пунктов меню или разделов диалоговых окон и палитр, названия используемых палитр, названия и типы данных элементов на лицевой панели или объектов на блок-диаграмме, численные значения, а также названия используемых ВП, функций и аппаратных средств.
- Курсив* Курсивом обозначены ссылки на дополнительную информацию.
- Ариэль** Текст в этом шрифте обозначает текст или символы, секции исходных кодов и синтаксис управляющих команд, примеры программирования. Этот шрифт также используется для выделения текста, вводимого с клавиатуры.

Занятие 1.

Введение в LabVIEW

На этом занятии представляются основы программной среды LabVIEW.

Чтобы быстро начать работу с LabVIEW, получить основные сведения о технике редактирования и отладки, типах палитр, используемых LabVIEW, «горячих» клавишах и Web-ресурсах – необходимо обратиться к справочнику **LabVIEW Quick Reference Card**.

Для просмотра PDF версии справочника, необходимо в пункте главного меню **Помощь** выбрать раздел книги LabVIEW (**Help»Search the LabVIEW Bookshelf**). Далее нажать клавишу **<Page Down>** и выбрать **LabVIEW Quick Reference Card**.

Изложены вопросы:

- A. Программная среда LabVIEW.
- B. Что такое виртуальный прибор (ВП).
- C. Последовательность обработки данных.
- D. Организация программной среды LabVIEW (окна, меню, инструменты).
- E. Справочная система среды LabVIEW и руководство пользователя.

A. Программная среда LabVIEW

Программа, написанная в среде LabVIEW, называется виртуальным прибором (ВП). ВП симулируют реальные физические приборы, например, осциллограф или мультиметр. LabVIEW содержит полный набор инструментов для сбора, анализа, представления и хранения данных.

В LabVIEW интерфейс пользователя — лицевая панель создается с помощью элементов управления (кнопки, переключатели и др.) и отображения (графики, светодиоды и др.). После этого на блок-диаграмме ВП осуществляется программирование с использованием графических представлений функций для управления объектами на лицевой панели.

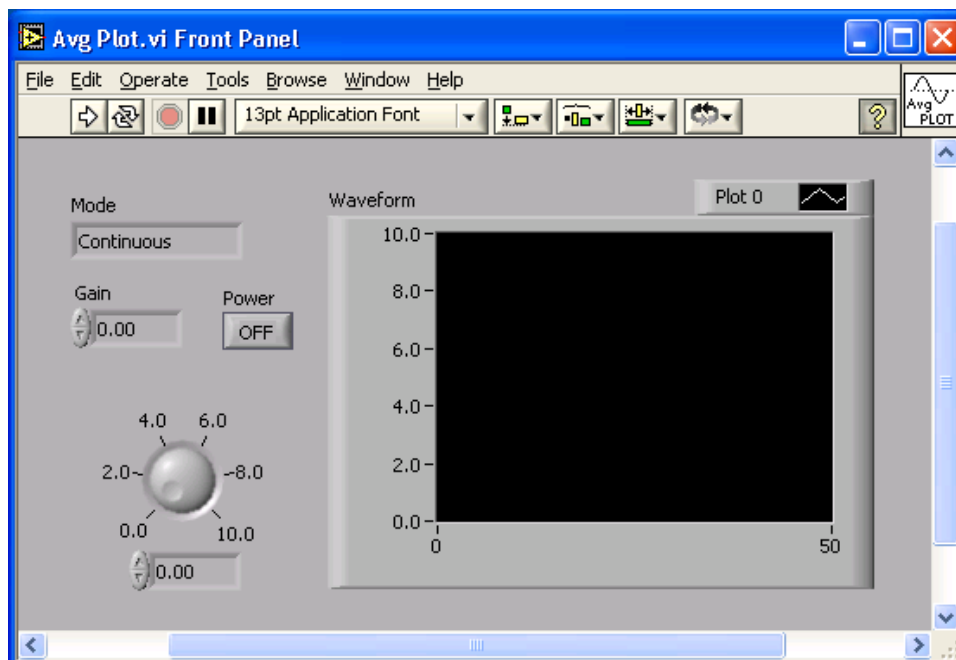
LabVIEW используется для программирования различных **DAQ**-устройств, систем контроля изображения и движения, аппаратных средств, имеющих интерфейсы типа **GPIB**, **VXI**, **PXI**, **RS-232** и **RS-485**. LabVIEW имеет встроенные возможности для работы в компьютерных сетях Интернет, используя LabVIEW Web Server и программные стандарты TCP/IP и Active X.

С помощью программной среды LabVIEW можно разрабатывать программно-аппаратные комплексы для тестирования, измерения, ввода данных, анализа и управления внешним оборудованием. LabVIEW — это 32-х разрядный компилятор, который создает как автономные модули (**.EXE**), так и совместно используемые динамические библиотеки (**.DLL**).

В. Виртуальные приборы (ВП)

ВП состоит из четырех основных компонентов – лицевой панели, блок-диаграммы, иконки и соединительной панели.

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



Лицевая панель 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.



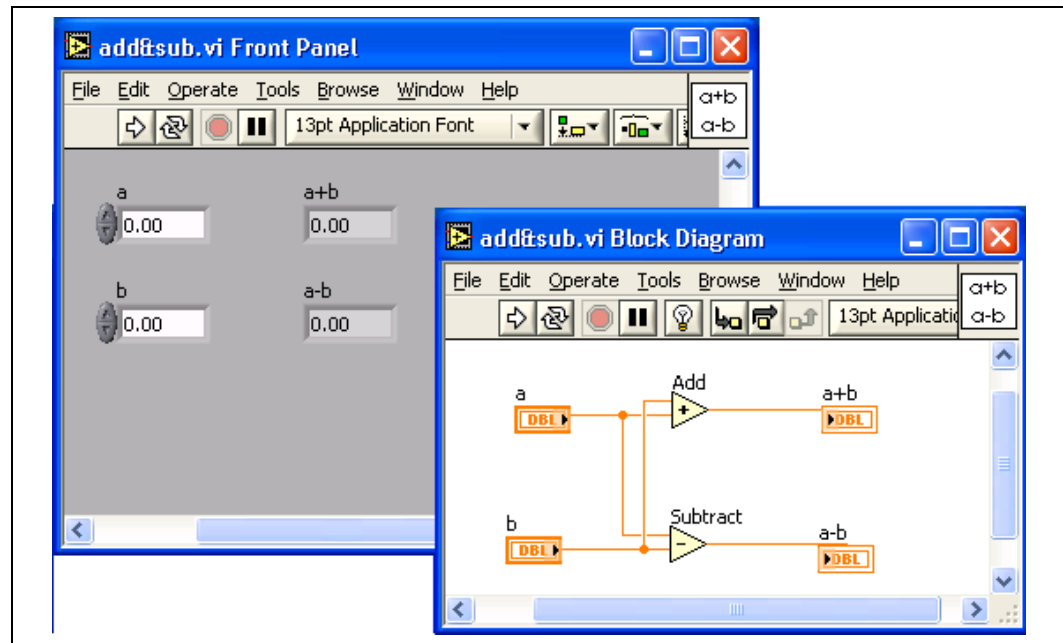
После помещения элементов **Управления** или **Отображения** данных на Лицевую панель, они получают свое графическое отображение на блок-диаграмме. Элемент, созданный на лицевой панели, невозможно удалить на блок-диаграмме. Объекты блок-диаграммы включают графическое отображение элементов лицевой панели, операторов,

функций, подпрограмм ВП, констант, структур и проводников данных, по которым производится передача данных между объектами блок-диаграммы.

Упражнение 1-1. ВП Арифметические операции

Цель: показать блок-диаграмму и соответствующую ей лицевую панель

Требуется построить виртуальный прибор для выполнения операции сложения и вычитания двух чисел a и b .



Для решения этой задачи нужно построить блок-диаграмму и соответствующую ей лицевую панель, представленные на рисунке. Построение ведется следующим образом.

1. Вызываются четыре цифровых элемента на лицевой панели, из которых два управляющих (обозначены a и b) и два индикатора ($a+b$, $a-b$). Их изображения одновременно появляются на блок-диаграмме. Из палитры функций вызываются элементы “сумма” (+) и “разность” (-) по пути Arithmetic & Compare >> Numeric.
2. Соединение элементов осуществляется инструментом “соединение” (катушка). Катушка подводится к элементу; когда элемент начнет мигать, нажимается левая клавиша мыши и появившийся провод подводят к нужному элементу. Когда последний начинает мигать – клавишу нужно отпустить.

3. В окна управляющих элементов вводятся заданные значения a и b (посредством инструмента “ввод текста”).
4. Задача запускается на выполнение кнопкой запуска с панели блок-диаграмм или с лицевой панели. В окнах индикаторов появляются результаты вычислений.



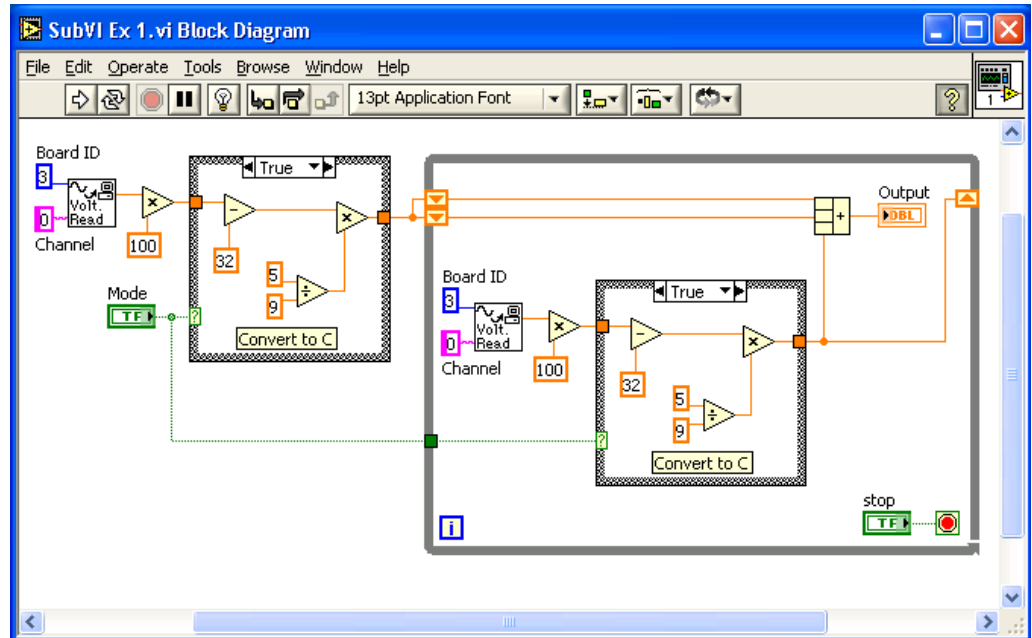
Для использования созданного виртуального прибора внутри другого ВП в качестве подпрограммы, после создания лицевой панели и блок-диаграммы, необходимо оформить иконку и настроить соединительную панель (область полей ввода/вывода данных). Подпрограмма ВП соответствует подпрограмме в текстовых языках программирования. Каждый ВП имеет показанную слева иконку в верхнем правом углу лицевой панели и блок-диаграммы. Иконка – графическое представление ВП. Она может содержать текст и/или рисунок. Если ВП используется в качестве подпрограммы, иконка идентифицирует его на блок-диаграмме другого ВП.



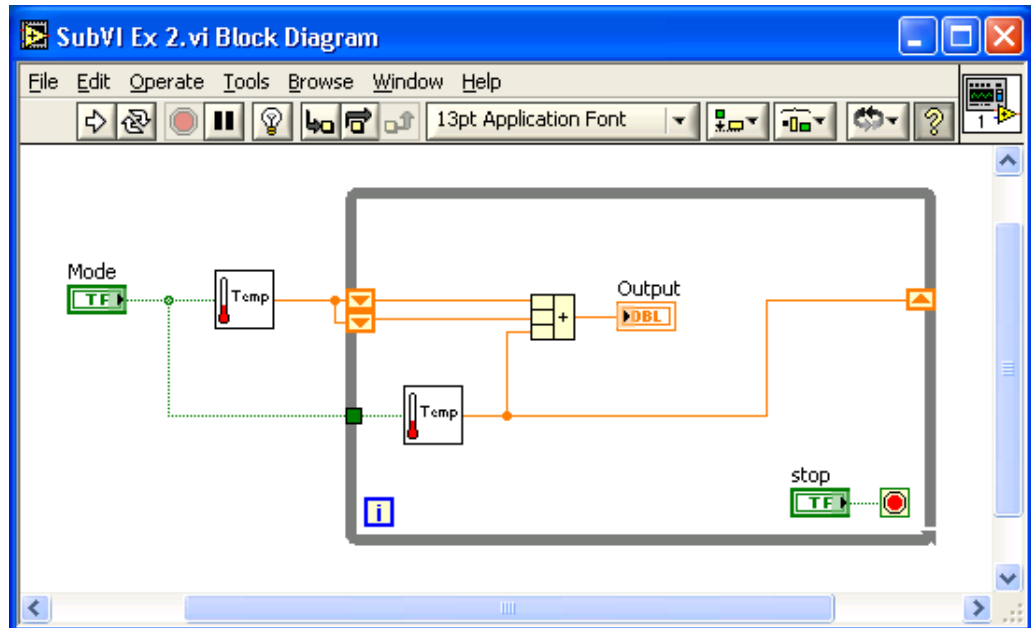
Необходимо также настроить показанную слева соединительную панель (область полей ввода/вывода данных), чтобы использовать ВП в качестве подпрограммы. Соединительная панель – это набор полей, соответствующий элементам ввода/вывода данных этого ВП. Поля ввода/вывода аналогичны списку параметров вызываемой функции в текстовых языках программирования. Область полей ввода/вывода данных позволяет использовать ВП в качестве подпрограммы. ВП получает данные через поля ввода данных и передает их на блок-диаграмму через элементы **Управления** лицевой панели. Результаты отображаются в его полях вывода данных посредством элементов **Отображения** лицевой панели.

Преимущество LabVIEW заключается в иерархической структуре ВП. Созданный виртуальный прибор можно использовать в качестве подпрограммы на блок-диаграмме ВП более высокого уровня. Количество уровней в иерархии не ограничено. Использование подпрограммы ВП помогает быстро изменять и отлаживать блок-диаграмму.

При создании ВП следует обратить внимание на то, что некоторые операции многократно повторяются. Для выполнения таких операций необходимо использовать подпрограммы ВП или циклы. Урок 4 *Многократные повторения и циклы* содержит дополнительную информацию об использовании циклов. Например, приведенная ниже блок-диаграмма содержит две идентичные операции.



Можно создать подпрограмму ВП, которая выполнит эту операцию, и можно вызвать эту подпрограмму дважды. Возможно многократное использование подпрограммы ВП в другом виртуальном приборе. Следующий пример демонстрирует использование **Temperature VI** в качестве подпрограммы на блок-диаграмме.

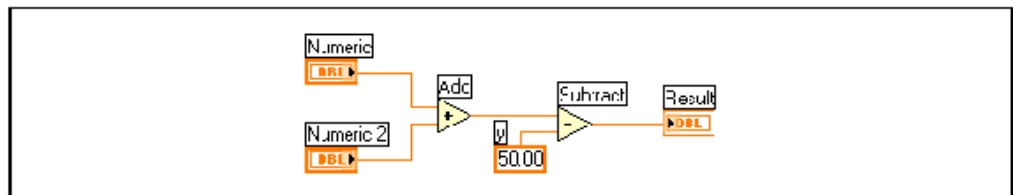


С. Последовательность обработки данных

В Visual Basic, C++, Java и большинстве других текстовых языков программирования порядок выполнения всей программы определяется расположением функций программы.

В среде LabVIEW используется потоковая модель обработки данных. Узлы блок-диаграммы выполняют заложенные в них функции, если данные поступили на все необходимые поля ввода/вывода. По окончании выполнения операции одним узлом результаты операции по проводникам данных передаются следующему узлу и т.д. Другими словами, готовность входных данных определяет последовательность выполнения узлов блок-диаграммы.

В качестве примера можно рассмотреть блок-диаграмму, которая складывает два числа и затем вычитает из получившейся суммы «50.0». В этом случае блок-диаграмма выполняется слева направо не потому, что объекты помещены в этом порядке, а потому, что одно из полей ввода функции **Subtract** (Вычитание) не определено, пока не выполнилась функция **Add** (Сложение) и не передала данные к функции **Subtract** (Вычитание). Не следует забывать, что узел выполняется только тогда, когда определены его поля ввода данных.



В следующем примере рассмотрена последовательность выполнения функций **Add** (Сложение), **Random Number** (Генератор случайных чисел) и **Divide** (Деление). Как видно, в данном случае последовательность выполнения функций не определена, так как поля ввода данных функций **Add** (Сложение) и **Divide** (Деление) инициализируются одновременно, а **Random Number** (Генератор случайных чисел) не имеет полей ввода данных.

Организация программной среды LabVIEW

При запуске LabVIEW появляется следующее диалоговое окно:



Диалоговое окно **LabVIEW** содержит следующие компоненты:

Панель меню со стандартными пунктами, например, **File»Exit**

Набор кнопок для создания и открытия ВП, конфигурации устройств DAQ, поиска полезной информации.

Кнопка **New** - создание нового ВП. Стрелка рядом с кнопкой используется для открытия пустого ВП или открытия диалогового окна **New**.

Кнопка **Open** - открытие созданного ранее ВП. Стрелка рядом с кнопкой предназначена для открытия недавно использовавшегося ВП.

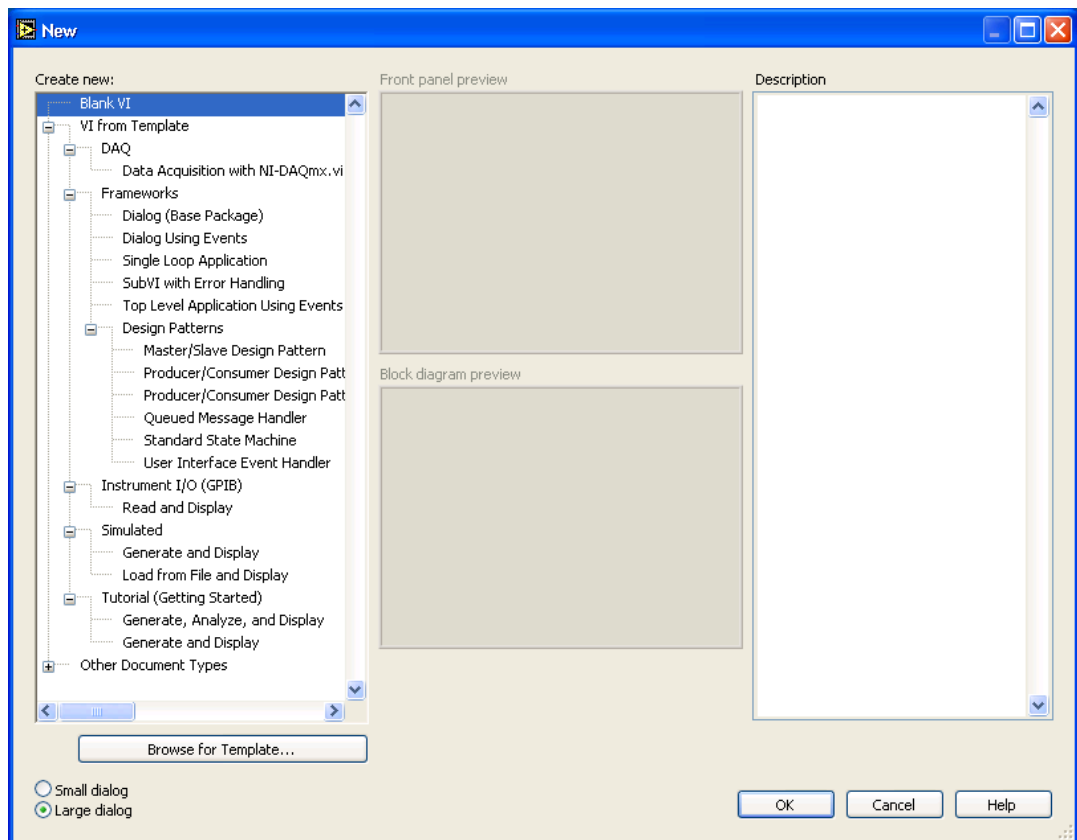
Кнопка **Configure** — настройка устройств DAQ. Стрелка рядом с кнопкой — конфигурация LabVIEW.

Кнопка **Help** — запуск *LabVIEW Help* (встроенной помощи). Стрелка рядом с кнопкой — для выбора опций помощи.

Создание нового ВП.

После нажатия кнопки **New** появляется диалоговое окно **New**. Также для открытия этого окна можно выбрать **File»New**. После выбора шаблона из спискового окна **Create New** (создание нового ВП) лицевая панель будет отображена в секции **Front Panel Preview** (предварительного просмотра лицевой панели), блок-диаграмма будет отображена в секции **Block Diagram Preview** (предварительного просмотра блок-диаграммы), описание шаблона будет отображено в секции **Description**.

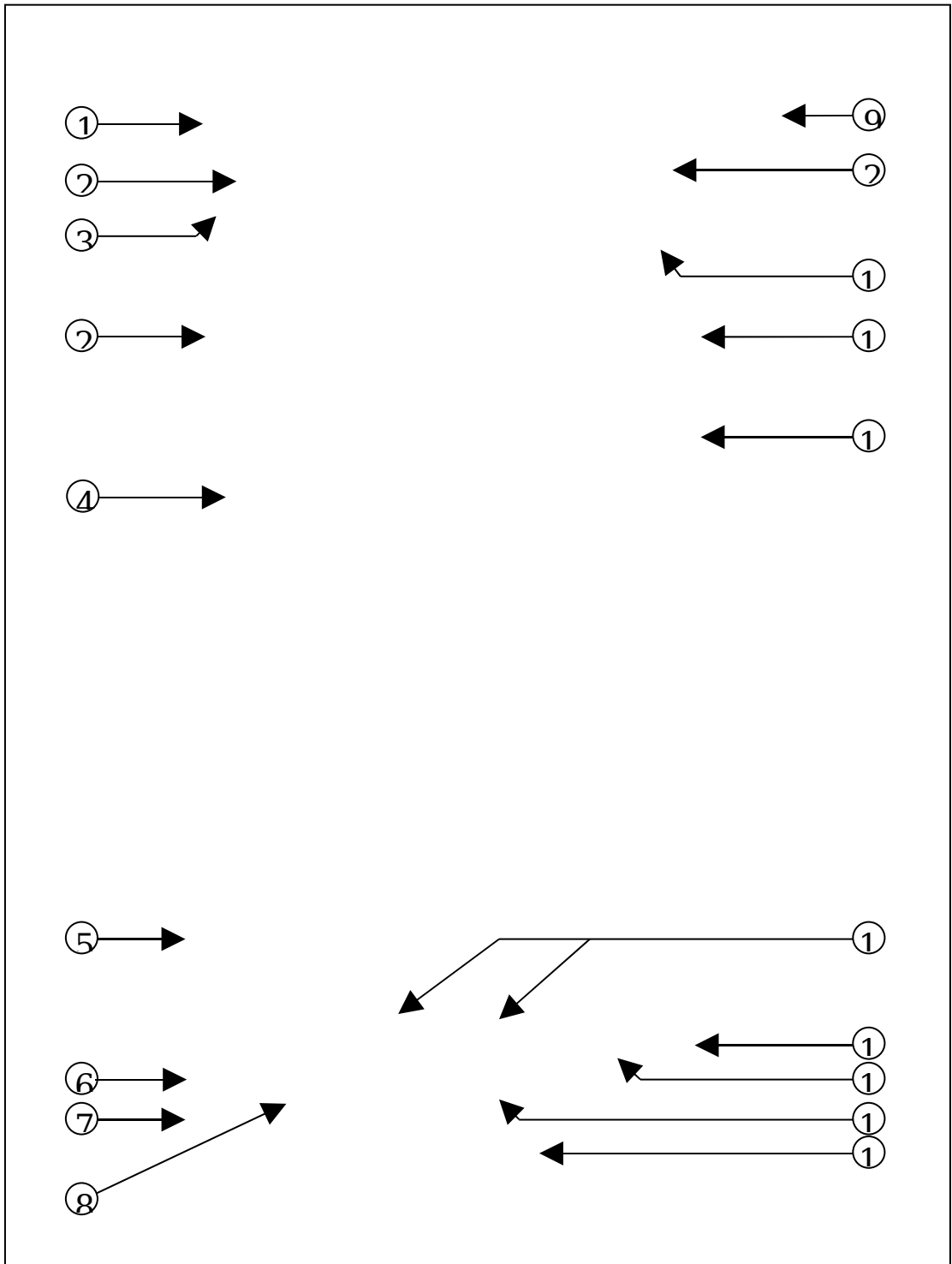
Выберите шаблон и нажмите кнопку **OK** для создания ВП или другого документа LabVIEW. Кнопки <Ctrl-N> используются для отказа от данного диалогового окна и создания пустого ВП.



Лицевая панель и окно блок-диаграммы

После нажатия кнопки **Blank VI** появляется окно лицевой панели. Это одно из двух окон LabVIEW, используемых для создания ВП. Другое окно содержит блок-диаграмму. Следующая иллюстрация демонстрирует лицевую панель и соответствующую ей блок-диаграмму:

- 1.
6. Терминал
данных кнопки
12. Двухкоординатный график
осциллограмм

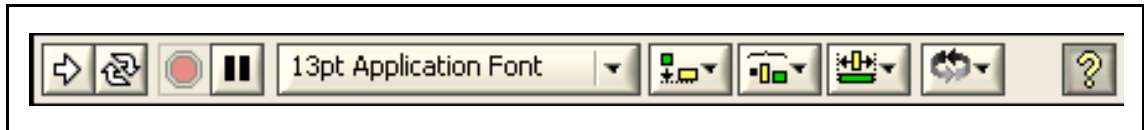


- | | | |
|--------------------------------|-----------------------|--|
| Инструментальная панель | | |
| 2. Собственная метка | 7. Числовая константа | 13. Проводники данных |
| 3. Цифровой элемент управления | 8. Функция Умножение | 14. Терминал данных графика осциллограмм |

- | | | |
|--|----------------------|----------------------------------|
| 4. Свободная метка | 9. Иконка | 15. Функция Объединение (Bundle) |
| 5. Терминал данных цифрового элемента управления | 10. Ручка управления | 16. Подпрограмма ВП |
| | 11. Панель | 17. Цикл с фиксированным |

Инструментальная панель лицевой панели

Инструментальная панель используется для запуска и редактирования ВП. Пример инструментальной панели



показан ниже.



Кнопка запуска **Run** - запускает ВП



Во время работы ВП кнопка **Run** меняет свой вид, как показано слева, если этот виртуальный прибор высокого уровня.



Если ВП работает в качестве подпрограммы, то кнопка **Run** выглядит, как показано слева.



Кнопка **Run** выглядит в виде «сломанной» стрелки, как показано слева, во время создания или редактирования ВП. В таком виде кнопка показывает, что ВП не может быть запущен на выполнение. После нажатия этой кнопки появляется окно **Error list**, в котором перечислены допущенные ошибки.



Кнопка непрерывного запуска **Run Continuously** - ВП выполняется до момента принудительной остановки.



Во время выполнения ВП появляется кнопка **Abort Execution**. Эта кнопка используется для немедленной остановки выполнения ВП.



Примечание. По возможности следует избегать использования кнопки **Abort Execution** для остановки ВП. Следует позволить ВП закончить передачу данных или выполнить остановку программным способом, гарантируя остановку ВП в определенном состоянии. Например, можно установить на лицевой панели кнопку, по нажатию которой ВП останавливается



Кнопка **Pause** приостанавливает выполнение ВП. После нажатия кнопки **Pause** LabVIEW подсвечивает на блок-диаграмме место остановки выполнения. Повторное нажатие – продолжение работы ВП.

13pt Application Font

Text Settings – выпадающее меню установок текста, включая размер, стиль и цвет.



В меню **Align Objects** производится выравнивание объектов по осям (по вертикали, по осям и т.д.).



В меню **Distribute Objects** производится выравнивание объектов в пространстве (промежутки, сжатие и т.д.).



В меню **Resize Objects** производится приведение к одному размеру многократно используемых объектов лицевой панели.



Меню **Reorder** используется при работе с несколькими объектами, которые накладываются друг на друга. Выделив один из объектов с помощью инструмента ПЕРЕМЕЩЕНИЕ, в меню **Reorder** следует выбрать его порядок отображения на лицевой панели.



Кнопка Context Help выводит на экран окно Context Help (контекстной справки)

Инструментальная панель блок-диаграммы

При запуске ВП на блок-диаграмме появляется показанная ниже инструментальная панель:



Кнопка **Highlight Execution** предназначена для просмотра потока данных через блок-диаграмму (режим отладки). Повторное нажатие кнопки отключает этот режим.



Кнопка **Step Into** используется при пошаговом выполнении цикла от узла к узлу, подпрограммы ВП и т.д. При этом узел мигает, обозначая готовность к выполнению.



Кнопка **Step Over** позволяет пропустить в пошаговом режиме цикл, подпрограмму и т.д.



Кнопка **Step Out** позволяет выйти из цикла, подпрограммы и т.д.

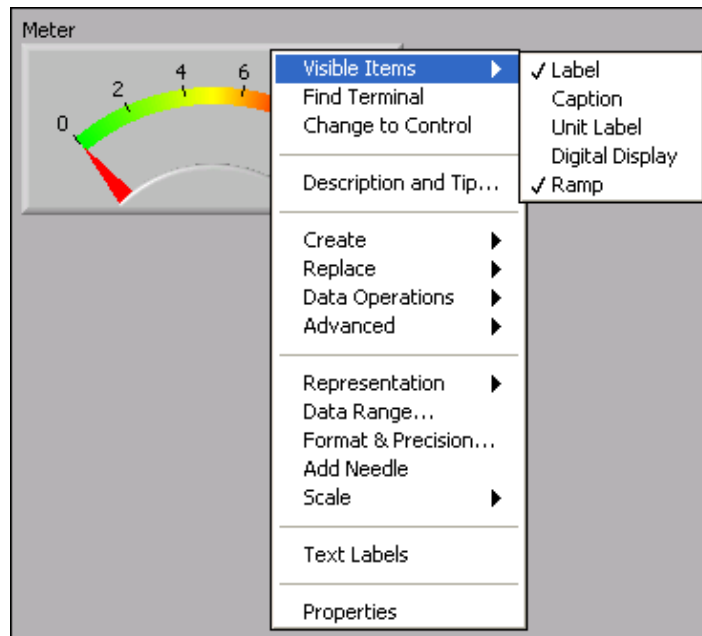
Выход из узла предполагает завершение выполнения этого узла в пошаговом режиме и переход в следующий.



Кнопка **Warning** появляется, когда есть потенциальная проблема с блок-диаграммой, но она не запрещает выполнение ВП. Кнопку **Warning** можно активизировать, войдя в пункт главного меню **Инструменты**, далее – **Опции**, **Отладка (Tools»Options»Debugging)**.

Контекстное меню

Контекстное меню используется наиболее часто. Все объекты LabVIEW, свободное рабочее пространство лицевой панели и блок-диаграммы имеют свои контекстные меню. Контекстное меню используется для изменения поведения объектов блок-диаграммы и лицевой панели. Контекстное меню вызывается щелчком правой кнопкой мыши на объекте, лицевой панели или блок-диаграмме. Пример контекстного меню показан ниже.



Главное меню

Главное меню в верхней части окна ВП содержит пункты общие с другими приложениями, такие как **Open, Save, Copy, Paste**, а также специфические пункты меню LabVIEW. Некоторые пункты главного меню содержат сведения о «горячих» клавишах вызова этих пунктов.



Внимание. Во время выполнения ВП некоторые пункты главного меню недоступны.

Пункт меню **File** используется для открытия, закрытия, сохранения и печати ВП.

Пункт меню **Edit** используется для поиска и внесения изменений в компоненты ВП.

Пункт меню **Operate** используется для запуска, прерывания выполнения и изменения других опций ВП.

Пункт меню **Tools** используется для связи с приборами и **DAQ** устройствами, сравнения ВП, формирования приложений и конфигурации LabVIEW.

Пункт меню **Browse** используется для перемещения по ВП и его иерархии.

Пункт меню **Window** используется для отображения окон LabVIEW и палитр.

Пункт меню **Help** используется для получения информации о палитрах, меню, инструментах, ВП и функциях, для получения пошаговой инструкции использования LabVIEW и информации о компьютерной памяти.

Палитры

LabVIEW имеет три вспомогательные палитры, используемые для создания и выполнения ВП: **Tools Palette** (Палитра Инструментов), **Controls Palette** (Палитра Элементов) и **Functions Palette** (Палитра Функций). Эти палитры можно поместить в любом месте экрана.

Палитра Инструментов

Создавать, редактировать и отлаживать ВП можно с помощью **Tools Palette** (Палитры Инструментов). Палитра Инструментов доступна как на лицевой панели, так и на блок-диаграмме. Термин инструмент подразумевает специальный операционный режим курсора мыши. При выборе определенного инструмента значок курсора изменяется на значок данного инструмента. Палитра Инструментов доступна через пункт главного меню **Window-Show Tools-Palette**. Палитру Инструментов можно

размещать в любой области рабочего пространства блок-диаграммы и лицевой панели.



Примечание. Удерживая нажатой клавишу **<Shift>** и щелкнув правой клавишей мыши, можно вывести на экран временную версию **Tools Palette** (Палитры Инструментов).



Если включен автоматический выбор инструмента, то при наведении курсора на объект лицевой панели или блок-диаграммы LabVIEW автоматически выбирает соответствующий инструмент из палитры **Tools** (Инструментов). Автоматический выбор инструментов включается нажатием на кнопку **Automatic Tool Selection** палитры **Tools** (Инструментов) или нажатием клавиш **<Shift-Tab>**.



Инструмент **УПРАВЛЕНИЕ** используется для изменения значения элементов управления или ввода текста. При наведении курсора на такой элемент как строковый элемент управления, значок инструмента меняется:



Инструмент **ПЕРЕМЕЩЕНИЕ** используется для выбора, перемещения или изменения размеров объектов. При наведении инструмента на объект изменяемого размера значок инструмента меняется:



Инструмент **ВВОД ТЕКСТА** используется для редактирования текста и создания свободных меток. При создании свободных меток значок инструмента меняется:



Инструмент **СОЕДИНЕНИЕ** создает проводники данных, соединяя объекты на блок-диаграмме.



Инструмент **ВЫЗОВ КОНТЕКСТНОГО МЕНЮ** вызывает

контекстное меню соответствующего объекта по щелчку левой кнопки мыши.



Инструмент **БЫСТРАЯ ПРОКРУТКА ЭКРАНА** используется для просмотра окна без использования полосы прокрутки.



Инструмент **ВВОД КОНТРОЛЬНОЙ ТОЧКИ** позволяет расставлять контрольные точки на ВП, функциях, узлах, проводниках данных, структурах и приостанавливать в них выполнение программы.



Инструмент **УСТАНОВКА ОТЛАДОЧНЫХ ИНДИКАТОРОВ** дает возможность исследовать поток данных в проводниках блок-диаграммы. Используется для просмотра промежуточных значений при наличии сомнительных или неожиданных результатов работы ВП.



Инструмент **КОПИРОВАНИЕ ЦВЕТА** предназначен для копирования цвета с последующей вставкой с помощью инструмента **РАСКРАШИВАНИЕ**.



Инструмента **РАСКРАШИВАНИЕ** позволяет изменить цвет объекта. Он также отображает текущий передний план и параметры настройки цвета фона.

Если автоматический выбор инструмента выключен, можно менять инструменты палитры **Tools** (Инструментов) с помощью клавиши **<Tab>**. Для переключения между инструментом **ПЕРЕМЕЩЕНИЕ** и **СОЕДИНЕНИЕ** на блок-диаграмме или между инструментом **ПЕРЕМЕЩЕНИЕ** и **УПРАВЛЕНИЕ** на лицевой панели – достаточно нажать пробел.

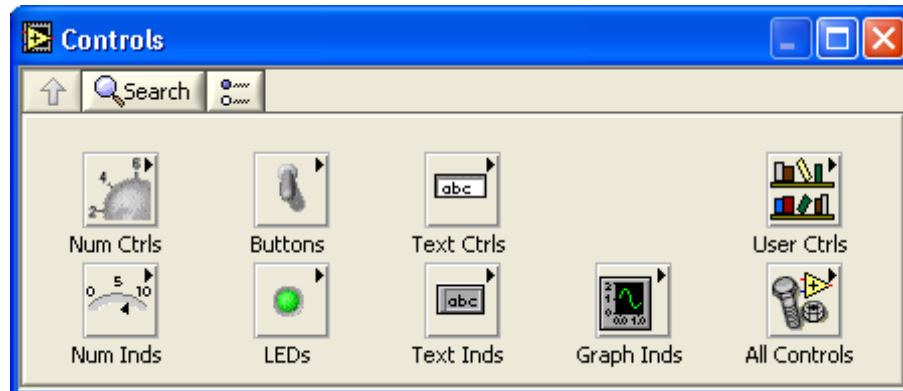
Палитра Элементов и палитра Функций

Палитра **Элементов (Controls)** и палитра **Функций (Functions)** содержат разделы, в которых размещены объекты для создания ВП. При нажатии на значок раздела, на экран выводится окно, содержащее его объекты. Для использования объекта палитры следует щелкнуть на нем мышью и поместить выбранный объект на лицевую панель или блок-диаграмму.

Для перемещения по разделам палитры, выбора элементов, ВП и функций следует использовать кнопки навигации. Для открытия ВП можно также щелкнуть правой кнопкой мыши иконку ВП на палитре и выбрать **Open VI** из контекстного меню.

Палитра Элементов

Палитра **Элементов** используется для размещения элементов управления и отображения на лицевой панели. Она доступна только на лицевой панели. Чтобы отобразить палитру **Элементов**, следует либо выбрать в пункте главного меню **Window>Show Controls-Palette**, либо щелкнуть правой кнопкой мыши в рабочем пространстве лицевой панели. Используя кнопку в верхнем левом углу палитры, можно зафиксировать ее на экране. По умолчанию палитра **Элементов** появляется в экспресс-виде, показанном ниже, и содержит лишь наиболее часто используемые элементы

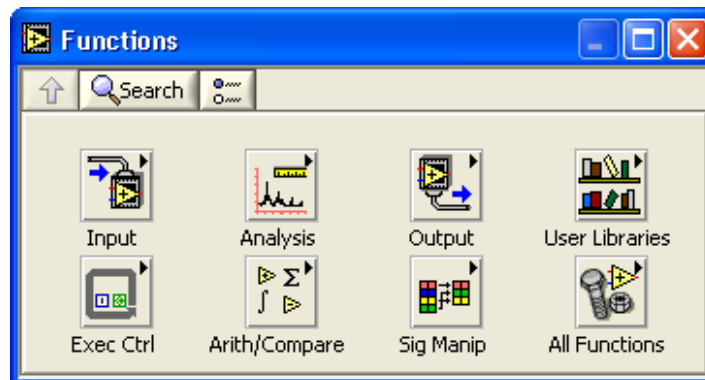


Используя кнопку **All Controls**, находящуюся в правом нижнем углу, можно разместить все элементы на палитре. Такая палитра, показанная ниже, является исходной палитрой **Элементов** для всех упражнений этого курса.



Палитра Функций

Палитра **Функций** используется для создания блок-диаграммы. Она доступна только на блок-диаграмме. Чтобы отобразить палитру **Функций**, следует либо выбрать в пункте главного меню **Window>Show Functions -Palette**, либо щелкнуть правой кнопкой мыши в рабочем пространстве блок-диаграммы. Используя кнопку в верхнем левом углу палитры можно зафиксировать ее на экране. По умолчанию палитра **Функций** появляется в экспресс-виде и отображает экспресс-ВП. Экспресс-ВП — узлы функций, которые можно настраивать с помощью диалогового окна. Они используются для выполнения стандартных измерений при минимальных соединениях.



При нажатии кнопки **Options** (опции), показанной слева, отображается страница **Controls/Functions Palettes** диалогового окна **Options**. Следует заменить **Palette View** на **Advanced**. Такая палитра, показанная ниже, является исходной палитрой **Функций** для всех упражнений этого курса



Поиск объектов на палитрах **Controls** (Элементов) и **Functions** (Функций)

Для быстрого перемещения по разделам палитры **Controls** (Элементов) и палитры **Functions** (Функций) предназначены кнопки, показанные ниже:



Up - перемещает на один уровень вверх в иерархии палитры.

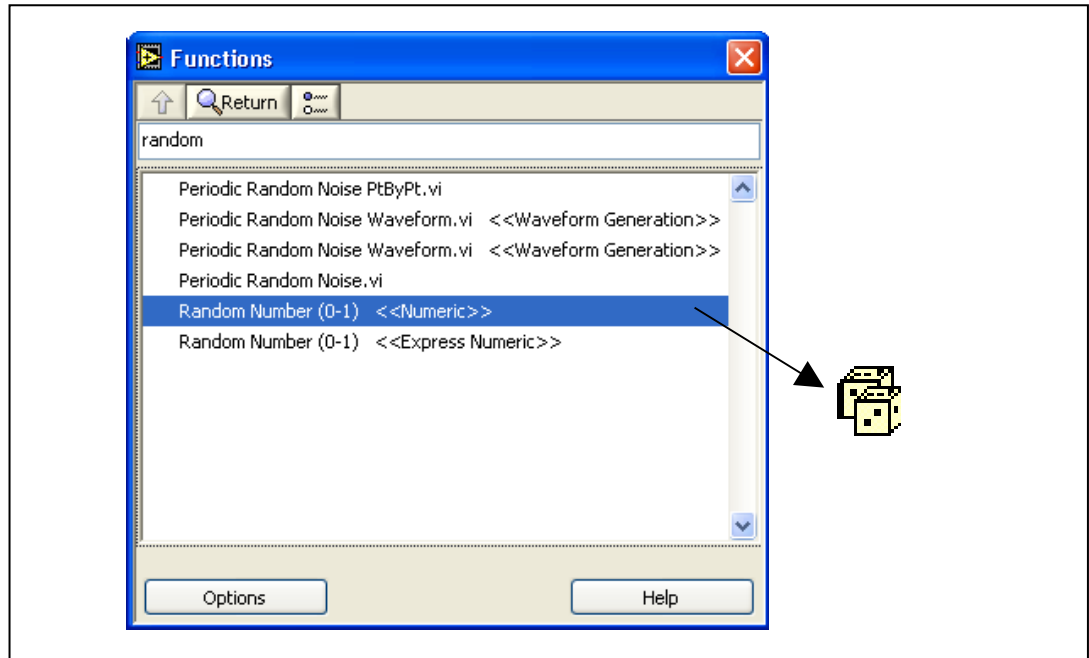


Search - переводит палитры в режим поиска. В этом режиме в палитрах производится поиск узлов, функций и ВП по названию.



Options - после нажатия этой кнопки открывается страница **Controls/Functions Palettes** диалогового окна **Options**, в которой производится настройка внешнего вида палитры.

Например, чтобы найти функцию **Random Number** (Генератор случайных чисел), следует нажать кнопку **Search** на палитре **Functions** (Функций) и ввести в поле ввода текста «Random Number». LabVIEW выводит на экран список узлов и функций, в названии которых встречается введенный текст. Выбрав в результатах поиска искомую функцию, можно перенести ее на блок-диаграмму с помощью мыши.



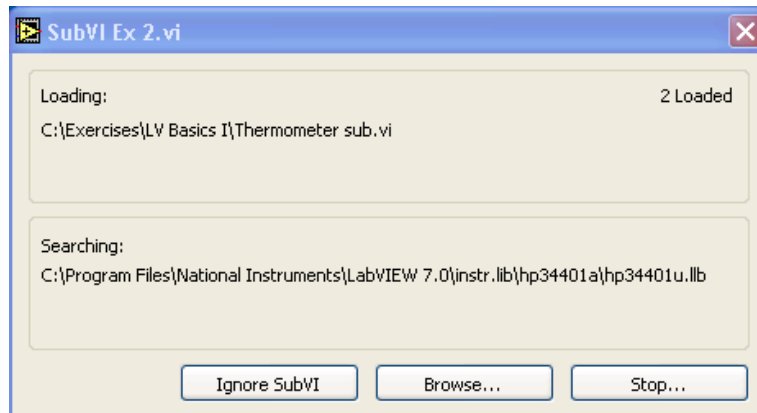
Двойной щелчок кнопкой мыши на искомой функции подсветит ее местоположение на палитре. Нажав кнопку **Up to Owning Palette**, можно посмотреть положение элемента в иерархии палитры.

Загрузка ВП

С помощью пунктов главного меню **File-Open** открывается диалоговое окно, позволяющее выбрать ВП и загрузить его в память компьютера.

Путь к редактируемым ВП – в упражнениях учебного курса **d:\проектирование АСУ\Введение Labview 70\exercises\LV Basics I.**

Ниже представлен вид диалогового окна, появляющегося во время загрузки ВП.



В этом окне перечисляются все подпрограммы выбранного ВП по порядку их загрузки в память. Остановить загрузку можно в любое время, нажав кнопку **Stop**.

Если LabVIEW не может сразу найти подпрограмму, то поиск продолжается по всем директориям, прописанным в пути поиска файлов **Search Path**. Пути поиска файлов можно редактировать, используя пункты меню Инструменты (**Tools** » **Options** » **Paths**). Можно сделать и так, чтобы LabVIEW игнорировал подпрограмму, нажав кнопку **Ignore Sub VI**, или использовать ручной поиск подпрограммы, нажав кнопку **Browse** (Обзор).

Краткое изложение пройденного материала, советы и секреты

ВП состоит из четырех основных компонентов – лицевой панели, блок-диаграммы, иконки и соединительной панели.

Лицевая панель – интерфейс пользователя ВП.

Блок-диаграмма – графический исходный текст программы, состоящий из узлов, терминалов данных и проводников.

Палитра инструментов предназначена для создания и редактирования ВП. Удерживая нажатой клавишу **<Shift>** и щелкнув правой клавишей мыши, можно вывести на экран временную версию Палитры Инструментов.

Палитра **Controls** (Элементов) предназначена для создания интерфейса лицевой панели. Для вывода на экран палитры **Controls** (Элементов) следует щелкнуть правой кнопкой мыши в открытом пространстве лицевой панели.

Палитра **Functions** (Функций) предназначена для создания блок-диаграммы. Для вывода на экран палитры **Functions** (Функций) следует щелкнуть правой кнопкой мыши в открытом пространстве блок-диаграммы.

Все объекты LabVIEW, свободное рабочее пространство лицевой панели и блок-диаграммы имеют свое контекстное меню. Обращение к всплывающему меню производится щелчком правой кнопкой мыши на объекте, лицевой панели или блок-диаграмме.

Занятие 2. Создание ВП

Дано описание встроенной помощи среды LabVIEW.
Представлены основы создания ВП.

Изложены вопросы:

- A. Встроенная помощь среды LabVIEW
- B. Компоненты ВП
- C. Создание ВП

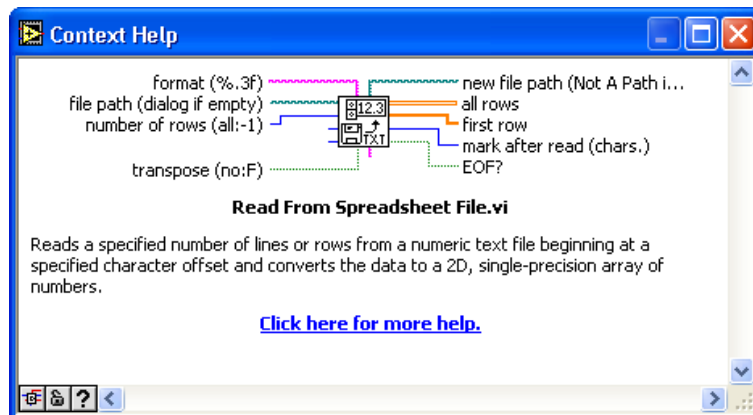
А. Встроенная Помощь среды LabVIEW (LabVIEW Help) и руководство пользователя

Окно контекстной справки **Context Help** помогает при создании и редактировании ВП. Более подробная информация расположена в **LabVIEW Help** (Встроенной Помощи).

Окно контекстной справки

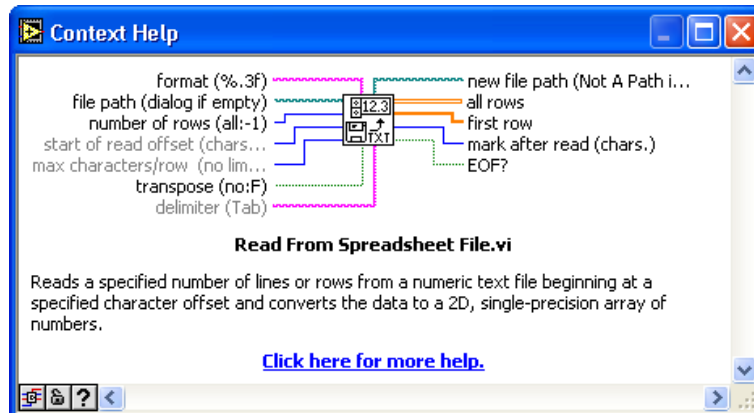
Окно **Context Help** (контекстной справки) выводится на экран из пункта главного меню **Help»Show Context Help** или вводом **<Ctrl-H>** с клавиатуры.

При наведении курсора на объект лицевой панели или блок-диаграммы в окне **Context Help** (контекстной справки) появляются иконки подпрограммы ВП, функции, константы, элементов управления или отображения данных с указанием всех полей ввода/вывода данных. При наведении курсора на опции диалогового окна в окне **Context Help** (контекстной справки) появляется описание этих опций. При этом поля, обязательные для соединения, выделены жирным шрифтом, рекомендуемые для соединения поля представлены обычным шрифтом, а дополнительные (необязательные) поля – выделены серым или вообще не показаны. Ниже приведен пример окна контекстной справки **Context Help**.



Для переключения между кратким и подробным представлением окна контекстной справки следует нажать кнопку **Simple/Detailed Context Help**, расположенную в нижнем левом углу окна **Context Help**. В кратком режиме представлены основные поля ввода/вывода данных. Дополнительные поля не описаны. Подробный режим

показывает все поля ввода/вывода данных, как показано ниже:



Чтобы зафиксировать текущее окно **Context Help** (контекстной справки), необходимо нажать кнопку **Lock Context Help**. Когда текущее окно **Context Help** (контекстной справки) зафиксировано, то его содержимое не меняется после наведения курсора на другой объект. Для отмены фиксации следует нажать кнопку второй раз. Описание опций можно получить из меню **Help**.



Для отображения подробного описания объекта в соответствующем разделе **LabVIEW Help** (Встроенной Помощи) следует нажать кнопку **More Help**.

Встроенная Помощь LabVIEW

Для отображения **LabVIEW Help** (Встроенной Помощи) можно нажать кнопку **More Help** в окне **Context Help** (контекстной справки) и выбрать в пункте главного меню помощь - **Help»VI, Function, & How-To Help** или в окне **Context Help** (контекстной справки) щелкнуть на **Click here for more help**.

Встроенная Помощь LabVIEW содержит детальные описания большинства палитр, меню, инструментов, ВП и функций, включает в себя пошаговую инструкцию использования особенностей LabVIEW и связана с **LabVIEW Tutorial** (руководством пользователя) и технической поддержкой на Web-сайте National Instruments.

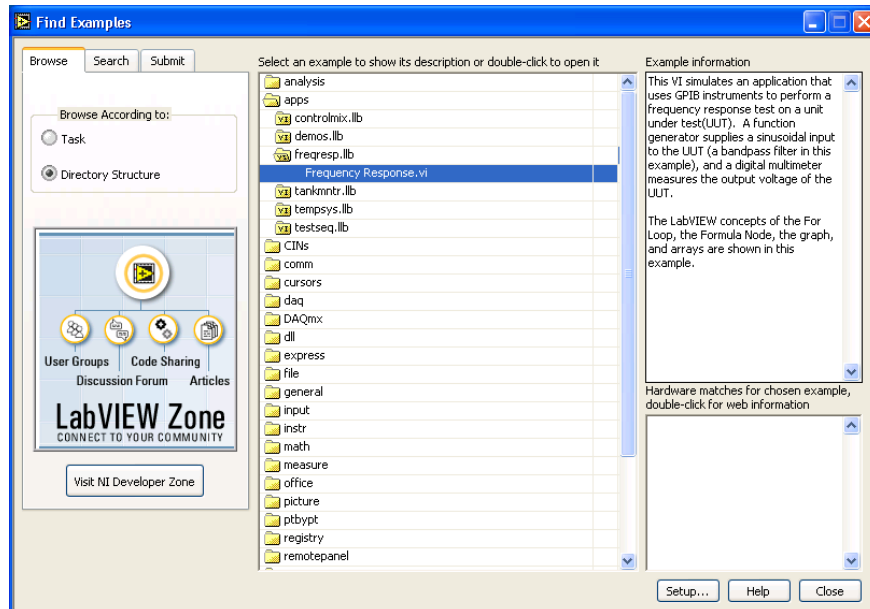
Для ознакомления с встроенной помощью среды LabVIEW предварительно откроем ВП, моделирующий частотную характеристику электронной цепи, а затем используем

справочную систему для получения сведений об элементах данного ВП.

Упражнение 2-1. ВП Частотная характеристика

Цель: Открыть и запустить ВП

1. Запустите LabVIEW (Start - Programs - National Instruments -LabVIEW 7.0 - LabVIEW). Появится диалоговое окно LabVIEW.
2. Выберите Help-Find Examples. На экране появится диалоговое окно поиска примеров ВП, разбитых по категориям.



3. Перейдите на закладку **Browse** (Обзор). Отметьте пункт **Directory Structure**. Выберите **Apps**, **Freqresp.llb**, и дважды щелкните на **Frequency Response VI**. Появится лицевая панель ВП Частотная характеристика.



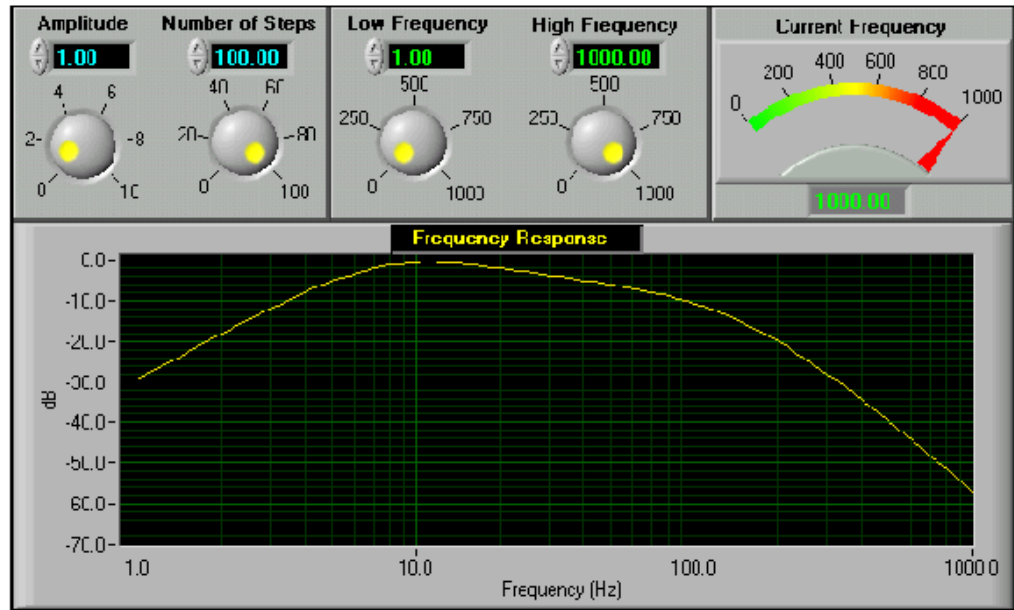
Примечание. Открыть этот ВП можно, нажав кнопку **Open VI** и перейдя в директорию **labview\examples\apps\freqresp.llb\Frequency Response.vi**.

Лицевая панель



4. На инструментальной панели нажмите кнопку **Run**, показанную слева, и запустите ВП. Данный ВП моделирует посылку сигнала к измерительному прибору и регистрацию его отклика. Реакция прибора в частотной

области показана на графике лицевой панели.



5. С помощью инструмента УПРАВЛЕНИЕ измените значение установки амплитуды **Amplitude**. Изменить значение можно либо переместив указатель кнопки в нужное положение, либо используя стрелки изменения значений элемента управления, либо введя число непосредственно в дисплей элемента.



Если число введено непосредственно в дисплей элемента, то необходимо нажать кнопку **Enter**, показанную слева, появившуюся на инструментальной панели. Иначе число не будет введено.

(MacOS and Sun) Нажать кнопку **<Return>**.

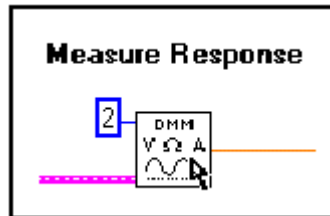
6. Нажать кнопку **Run** и запустить ВП. Изменяя значения других средств управления, находящихся на панели, исследовать работу ВП.

Блок-диаграмма

7. Перейдите на блок-диаграмму. Для этого выберите в главном меню **Window>Show Diagram** или введите **<Ctrl-E>** с клавиатуры.

Блок-диаграмма содержит несколько основных объектов, включая подпрограммы ВП, функции и структуры, о которых пойдет речь позднее.

8. С помощью инструмента УПРАВЛЕНИЕ дважды щелкните по иконке **DMM**.



Эта иконка – графическое представление подпрограммы **Demo Fluke 8840A VI**. После двойного щелчка откроется подпрограмма и на экране появится ее лицевая панель.



Дизайн лицевой панели напоминает мультиметр. Вот почему программы LabVIEW называются виртуальными приборами.

Создавая модульные приложения LabVIEW, можно изменять только части приложения и/или многократно использовать эти части в других приложениях. Например, эта подпрограмма моделирует действие комбинированного прибора **Demo Fluke**, но пользователь может внести в него изменения, чтобы получить новые функции.

9. Выберите в главном меню пункты **File»Close** и закройте **Demo Fluke 8840AVI**.
10. Не закрывайте ВП Частотная характеристика. Этот ВП будет использоваться в упражнении 2-2.

Конец упражнения 2-1

Упражнение 2-2. Использование справочной системы LabVIEW и руководства пользователя

Цель: Использование справочной системы LabVIEW для получения информации об объектах лицевой панели и блок-диаграммы и особенностях их использования

Часть А. Окно контекстной справки

1. ВП Частотная характеристика должен быть открыт. Если нет, откройте его, как описано в упражнении 2-1.
2. Выберите в главном меню пункты **Window>Show Diagram**, чтобы перейти на блок-диаграмму.
3. Выберите в главном меню пункты **Help>Show Context Help** или нажмите клавиши **<Ctrl-H>**, чтобы отобразить окно контекстной справки **Context Help**.
4. Получить информацию об объекте в окне контекстной справки **Context Help** можно, наведя на них курсор.



- a. Поместите инструмент ПЕРЕМЕЩЕНИЕ, показанный слева, над функцией **Logarithm Base 10**, расположенной под меткой **Bode Plot**. В окне контекстной справки **Context Help** появится описание функции.



- b. В окне контекстной справки **Context Help** нажмите кнопку **More Help**, показанную слева, для перехода в соответствующий раздел **LabVIEW Help** (Встроенной Помощи). Можно также щелкнуть по ссылке **Click here for more help** окна контекстной справки **Context Help**.

LabVIEW Help (Встроенная Помощь) содержит подробное описание палитр, меню, инструментов, ВП и функций. Получите подробное описание других функций.



- c. Наведите инструмент СОЕДИНЕНИЕ, показанный слева, на поля ввода/вывода данных функции **Logarithm Base 10**. Соответствующие поля в окне контекстной справки **Context Help** начнут мигать.
- d. Переместите инструмент СОЕДИНЕНИЕ на проводник данных. В окне контекстной справки **Context Help** появится описание типа данных в проводнике.

Часть В. Справочная система LabVIEW

5. Выберите в главном меню пункты **Help»VI, Function, & How-To Help**, чтобы открыть **LabVIEW Help** (Встроенную Помощь среды LabVIEW). **LabVIEW Help** (Встроенная помощь среды LabVIEW) включает в себя пошаговую инструкцию использования особенностей LabVIEW и связана с **LabVIEW Tutorial** (руководством пользователя), PDF версией учебника LabVIEW и технической поддержкой на Web-сайте National Instruments.
6. Использование оглавления справочной системы LabVIEW.
 - a. Нажмите закладку **Index** для вывода оглавления справочной системы LabVIEW.
 - b. В окне ввода текста введите Frequency Response. В оглавлении появятся два близких по смыслу ответа на запрос.
 - c. Просмотрите каждый. Справочная система LabVIEW выведет соответствующий раздел.
 - d. Нажмите закладку **Contents** для вывода содержания раздела справочной системы LabVIEW.
 - e. Снова нажмите закладку **Index**.
 - f. В окне ввода текста напечатайте GPIB (КОП), потому что ВП Частотный анализ – симулятор приложения GPIB.
 - g. Войдите в подраздел **«functions»** для просмотра описания работы функций GPIB в LabVIEW.
7. Выполните поиск по всей справочной системе LabVIEW.
 - a. Нажмите закладку **Search**.
 - b. В окне ввода текста напечатайте displaying frequency. В окне результата поиска нажмите **«Frequency Information Displayed»**.
8. Если на компьютере установлен **«Adobe Acrobat Reader»**, нажмите на закладку **Contents** и откройте PDF версию

руководства пользователя LabVIEW справочной системы LabVIEW.

- a. Выберите папку **Related Documentation** в окне **Contents**. Справа появится соответствующий раздел.
 - b. Нажмите ссылку **LabVIEW User Manual**, чтобы открыть PDF версию руководства пользователя LabVIEW.
 - c. Нажмите кнопку **Back**, чтобы вернуться в раздел **Related Documentation**.
9. Если компьютер подключен к Интернету, обратиться к технической поддержке на *Web-сайте National Instruments*.
- a. Выберите папку **Technical Support Resources** в основном разделе **Contents** (Содержания).
 - b. Войдите в папку и выберите страницу **Technical Support Resources**. Появится раздел ресурсов технической поддержки.
 - c. Выберите ссылку **Technical Support** для перехода в секцию **Технической поддержки** на *ni.com*.
 - d. Вернитесь в раздел **Technical Support Resources**.
 - e. Выберите ссылку **NI Developer Zone** и войдите в **Developer Zone** (Зону Разработчика) National Instruments.
 - f. В окне ввода текста напечатайте **Frequency Response** и нажмите **GO**.
Появившиеся разделы демонстрируют различные решения с помощью продуктов фирмы National Instruments.

Часть С. Книжная полка LabVIEW

10. Если на компьютере установлен Acrobat Reader, выберите в меню **Help» Search the LabVIEW Bookshelf** и перейдите в **LabVIEW Bookshelf** (Книжную Полку LabVIEW).
11. Нажмите ссылку **Search**, появится диалоговое окно **Adobe Acrobat Search**.

12. В поле ввода текста напечатайте «**Frequency Response**» и нажмите кнопку **Search**. На экране появятся все доступные руководства LabVIEW.
13. Дважды щелкните на первом результате поиска. На экране появится точное местоположение «**Frequency Response**» в документе.
14. Снова выведите результат поиска, выбрав пункты меню **Edit»Search»Results**.
15. Просмотрите другие результаты поиска и выйдите из Acrobat Reader.
16. На лицевой панели выберите пункты меню **File»Close** и закройте ВП Частотный анализ. Изменения не сохраняйте.

Конец упражнения 2-2

В. Компоненты ВП

ВП состоит из четырех основных компонентов – лицевой панели, блок-диаграммы, иконки и соединительной панели. Подробная информация о создании иконки и соединительной панели – в уроке 3 «Создание подпрограмм ВП».

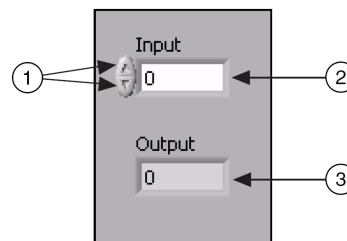
Лицевая панель

На лицевой панели создаются элементы управления и отображения, которые являются интерактивными средствами ввода и вывода данных этого ВП. Элементы **Управления** – кнопки, переключатели и другие устройства ввода данных. Элементы **Отображения** – графики, светодиоды и другие индикаторы. Элементы **Управления** моделируют устройства ввода данных и передают данные на блок-диаграмму ВП. Элементы отображения моделируют устройства вывода и отображения данных, которые получает или генерирует блок-диаграмма.

Для размещения элементов **Управления** и **Отображения** данных на лицевой панели используется палитра **Controls** (Элементов). Палитра **Controls** (Элементов) доступна только с лицевой панели. Для вывода на экран палитры **Controls** (Элементов) следует выбрать пункты главного меню **Window>Show Controls Palette** или щелкнуть правой кнопкой мыши в рабочем пространстве лицевой панели.

Числовые элементы управления и отображения данных

Чаще других используют два числовых объекта – это числовой элемент управления и числовой элемент отображения данных:



1 Кнопки
изменения

2 Цифровой
элемент

3 Цифровой
элемент

значений

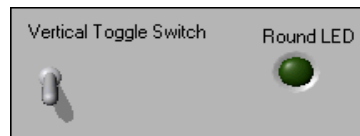
управления

отображения
данных

Ввод или изменение значения элемента управления осуществляется либо с помощью кнопок приращения значений, либо нужное значение просто вводится в элемент с помощью инструмента ВВОД ТЕКСТА, после чего следует нажать кнопку **<Enter>**.

Логические элементы управления и отображения данных

Логические элементы управления и отображения используются для ввода и отображения значения логической переменной (TRUE/FALSE — ИСТИНА/ЛОЖЬ). Логические объекты моделируют выключатели, кнопки и светодиоды. Вертикальный переключатель и круглый светодиод показаны ниже:

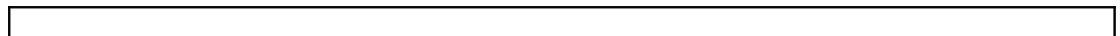


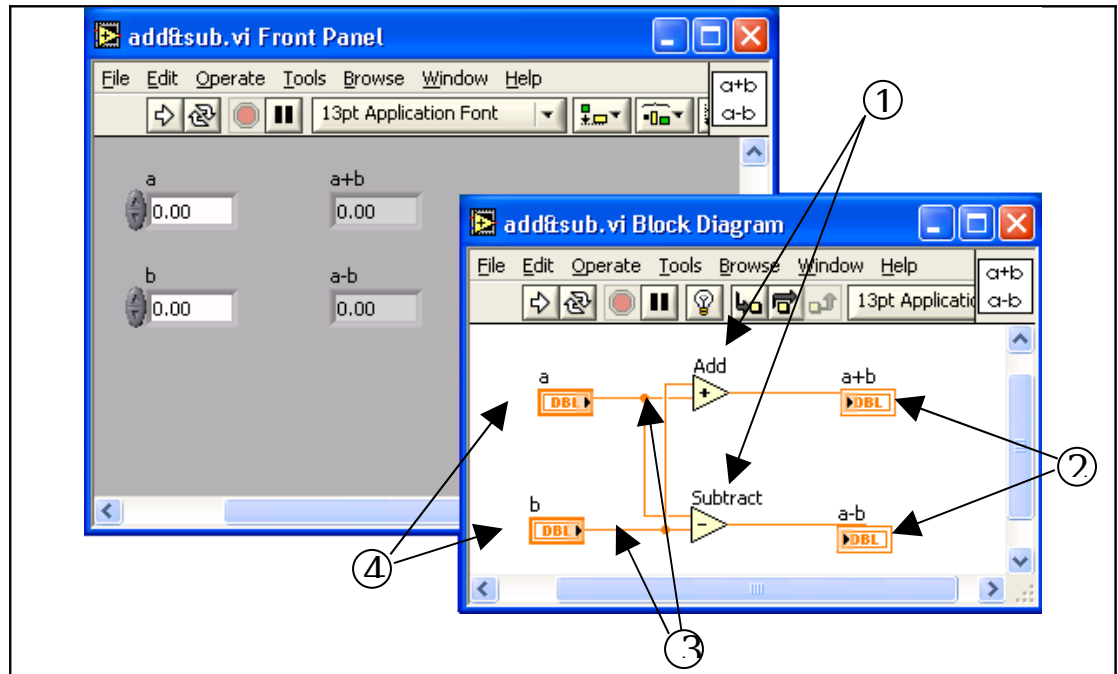
Редактирование элементов управления и отображения данных

Почти все элементы управления и отображения данных можно редактировать, используя их контекстное меню. Для вызова контекстного меню следует щелкнуть правой кнопкой мыши на объекте. Например, для редактирования метки - щелкнуть правой кнопкой мыши на метке.

Блок-диаграмма

Блок-диаграмма состоит из узлов, терминалов и проводников данных, как показано ниже:





1 Узлы	2 Терминалы данных элементов отображения	3 Проводники данных	4 Терминалы данных элементов управления
--------	--	---------------------	---

Терминалы данных



Объекты лицевой панели на блок-диаграмме отображаются в виде терминалов данных (графическое изображение прямоугольной формы с буквенно-численными обозначениями). Буквенно-численное обозначение на терминале данных определяет тип данных, который может использоваться в элементах управления или отображения. Например, DBL-терминал, показанный слева, определяет, что данный элемент управления использует числа двойной точности с плавающей запятой.



Терминал данных может отображаться в виде иконки, показанной слева. Для этого достаточно щелкнуть правой кнопкой мыши в поле терминала данных и выбрать **View as Icon** (отображать в виде иконки) из контекстного меню. Снять метку для отображения в стандартном виде. Отображение терминала данных в стандартном виде позволяет сохранить место на блок-диаграмме.

Терминалы данных обеспечивают обмен данными между лицевой панелью и блок-диаграммой; они подобны переменным и константам текстовых языков

программирования. Различают терминалы данных следующих типов – терминалы элементов управления и отображения данных, терминалы узлов. Терминалы элементов управления и отображения относятся к средствам управления и отображения данных на лицевой панели. Данные, введенные в элементы управления на лицевой панели (а и b на рисунке 2-1), поступают на блок-диаграмму через эти терминалы. Когда функции **Add** (Сложение) и **Subtract** (Вычитание) завершают свои вычисления, то на выходе выдают новое значение данных. Эти значения поступают на терминалы элементов отображения данных и передаются на лицевую панель.



Терминалы данных предыдущей блок-диаграммы принадлежат четырем элементам лицевой панели. Соединительная панель функций **Add** (Сложение) **Subtract** (Вычитание), показанная слева, содержит три поля ввода/вывода данных. Для отображения соединительной панели следует щелкнуть правой кнопкой мыши на функции и в контекстном меню выбрать **Visible Items»Terminals**.

Узлы

Узлы – это объекты на блок-диаграмме, которые имеют одно или более полей ввода/вывода данных и выполняют алгоритмические операции ВП. Они аналогичны операторам, функциям и подпрограммам текстовых языков программирования. Узлы включают в себя функции, подпрограммы ВП и структуры. Подпрограмма ВП – виртуальный прибор, который можно использовать на блок-диаграмме другого ВП в качестве подпрограммы. Структуры – это элементы управления процессом, такие как структура **Case** (Варианта), цикл **While** (цикл по условию) и т.д. Узлы **Add** (Сложение) и **Subtract** (Вычитание), показанные на предыдущей блок-диаграмме, – узлы функций.

Иконки в сравнении с раскрывающимися узлами



ВП и экспресс-ВП могут отображаться в виде иконок или раскрывающихся узлов. Раскрывающиеся узлы представляют собой иконки, окруженные цветным полем. Подпрограммы ВП отображаются на желтом поле, экспресс-ВП – на голубом. Использование иконок, таких как иконка ВП Function Generator, показанная слева, позволяет сохранить место на блок-диаграмме.

Использование раскрывающихся узлов, таких как раскрывающийся узел ВП Function Generator VI, (см. после

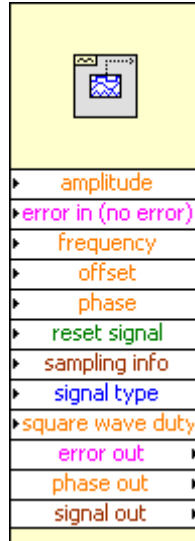
пункта 4), позволяет упростить процесс соединения и документирования блок-диаграммы. По умолчанию подпрограммы ВП отображаются в виде иконок, экспресс-ВП — в виде раскрывающихся узлов.

Для отображения подпрограммы ВП или экспресс-ВП в виде иконок достаточно щелкнуть по ним правой кнопкой мыши и выбрать в контекстном меню пункт **View as Icon**. Для отображения подпрограммы ВП или экспресс-ВП в виде раскрывающихся узлов снимите выделение с этого пункта.

Предусмотрена возможность изменения размеров раскрывающихся узлов с целью облегчения соединений, но при этом на блок-диаграмме расходуется много места. Для изменения размера узла следует выполнить следующее:

1. Установите инструмент ПЕРЕМЕЩЕНИЕ на узел. На верхнем и нижнем основаниях узла появятся метки изменения размера.
2. Переместите курсор на ручку изменения размера для перевода курсора в режим изменения размера.
3. Переместите курсор, удерживая нажатой левую кнопку мыши, до достижения размера, необходимого для размещения дополнительных терминалов.
4. Отпустите левую кнопку мыши.

На следующем примере показан ВП Function Generator в виде раскрывающегося узла с измененным размером.



Внимание. При отображении подпрограммы ВП или Экспресс-ВП в виде раскрывающегося узла, поля ввода/вывода этого узла не отображаются.

С. Создание ВП

Для создания ВП откройте новый ВП или шаблон и сохраните его. После этого можно конструировать лицевую панель и блок-диаграмму.

Открытие ВП и Шаблоны

Диалоговое окно **New** используется для создания различных компонент в среде LabVIEW при построении приложений. Можно начинать с пустого ВП или с шаблона для упрощения программирования. Диалоговое окно **New** содержит следующие компоненты:

- **Create New** — отображает шаблоны, с помощью которых можно создавать ВП или другие документы LabVIEW. Для этого достаточно выбрать шаблон и нажать кнопку ОК.

Blank VI — открывает пустую лицевую панель и блок-диаграмму

VI from Template — открывает лицевую панель и блок-диаграмму, содержащие компоненты для построения различных видов ВП.

Frameworks — открывает лицевую панель и блок-диаграмму, содержащие компоненты для построения ВП, включающих специальные виды выполняемых функций

Instrument I/O — открывает лицевую панель и блок-диаграмму, содержащие компоненты, необходимые для связи с внешними устройствами, подсоединенными к компьютеру

Simulated — открывает лицевую панель и блок-диаграмму, содержащие компоненты, необходимые для моделирования получения данных с устройства

Tutorial (Getting Started) — открывает лицевую панель и блок-диаграмму, содержащие компоненты, необходимые для построения ВП, предназначенных для выполнения упражнений руководства Getting Started

Other Document Types — открывает документы для построения других объектов LabVIEW.

- **Browse for Template** — отображает диалоговое окно **Browse** (обзор), позволяющее указать местоположение ВП, элемента управления или шаблона.
- **Front panel preview** — отображает лицевую панель для выбранного из списка **Create New** шаблона ВП

- **Block diagram preview** — отображает блок-диаграмму для выбранного из списка **Create New** шаблона ВП
- **Description** — отображает описание выбранного из списка **Create New** шаблона ВП, в случае если шаблон имеет описание

Сохранение ВП

Выбрав из пункта главного меню **File** подпункт **Save, Save All** или **Save with Options**, можно сохранить ВП либо как отдельный файл, либо как группу из нескольких ВП в файл библиотеки ВП LabVIEW. Файл библиотеки ВП имеет расширение *.llb. National Instruments рекомендует сохранять ВП в виде отдельных файлов, организованных в каталоги, особенно если над одним и тем же проектом работают несколько разработчиков.

Перемещение ВП между платформами

Предусмотрена возможность портирования ВП из одной платформы в другую, например из MacOS в Windows. На новой платформе LabVIEW автоматически перекомпилирует ВП.



Примечание. Некоторые специфические ВП не переносятся между платформами, такие как DDE (Динамический обмен данными) ВП, ActiveX ВП и AppleEvents.

Для получения дополнительной информации о перемещении ВП между платформами следует выбрать в главном меню пункт **Help»Search the LabVIEW Bookshelf - Porting and Localizing LabVIEW VIs.**

Краткое изложение пройденного материала, советы и секреты

ВП состоит из четырех основных компонентов – лицевой панели, блок-диаграммы, иконки и соединительной панели.

Все объекты LabVIEW, свободное рабочее пространство лицевой панели и блок-диаграммы имеют свое контекстное меню. Обращение к всплывающему меню производится щелчком правой кнопкой мыши на объекте, лицевой панели или блок-диаграмме.

Меню **Help** используется для вывода на экран окна **Context Help** (контекстной справки) и **LabVIEW Help** (Встроенной Помощи среды LabVIEW), которая описывает палитры, меню, инструменты, ВП и функции, включая пошаговую инструкцию использования особенностей LabVIEW.

Выберите меню **Help»Search the LabVIEW Bookshelf** и перейдите в **LabVIEW Bookshelf (Книжную Полку LabVIEW)** для использования PDF версии справочной системы и руководства пользователя LabVIEW.

Дополнительные упражнения

- 2.3. Создайте ВП, который сравнивает два числа. Если первое число больше или равно второму, то должен включаться светодиод.



Совет. Следует использовать функцию **Greater Or Equal?** (Больше или равно?), которая расположена на палитре функций **Functions»Comparison**

Сохраните ВП под именем *Сравнение чисел.vi*

- 2.4. Создайте ВП, который генерирует случайные числа в диапазоне от «0.0» до «10.0» и делит результат на число, введенное на лицевой панели. Если производится попытка деления на ноль, должен включаться светодиод.

Сохраните ВП под именем *Деление на ноль.vi*

Занятие 3. Редактирование ВП

Представлены основы редактирования ВП.

Изложены вопросы:

- A. Типы данных и проводники данных
- B. Редактирование ВП
- C. Отладка ВП

А. Типы и проводники данных

В среде LabVIEW проводники данных используются для соединения многочисленных терминалов данных. Поля ввода/вывода должны быть совместимыми с типами данных, передаваемыми им по проводникам. Например, нельзя соединять поле вывода массива с полем ввода данных численного типа. Кроме того, характер соединения должен быть корректным. Проводники должны быть подсоединены лишь к одному источнику данных и, по крайней мере, к одному полю ввода данных. Например, нельзя соединять 2 элемента отображения.

Типы данных










В данном курсе используются следующие типы данных:

- **Numeric** (численный тип)
 - Floating point** — число с плавающей запятой (**оранжевый цвет**), отображается в виде оранжевых терминалов. Может быть представлено в виде **single** (32 bit), **double** (64-bit) или **extended** (128-bit) **precision** (с одиночной, двойной или расширенной точностью). Число с плавающей запятой может быть комплексным.
 - Integer** — целочисленный тип (**синий цвет**), отображается в виде голубых терминалов. Возможны три представления целых чисел: 8, 16 и 32 бита. Один бит может использоваться для знака числа, если это число является знаковым целым.
- **Boolean** — логический тип (**зеленый цвет**), отображается в виде зеленых терминалов. Логический тип может принимать только два значения: 0 (FALSE) или 1 (TRUE).
- **String** — строковый тип (**розовый цвет**), отображается в виде розовых терминалов. Строковый тип данных содержит текст в ASCII формате.
- **Path** — путь к файлу, отображается в виде терминалов. Путь к файлу близок строковому типу, однако, LabVIEW форматирует его, используя стандартный синтаксис для используемой платформы.
- **Array** — массивы (отличаются увеличенной толщиной соединительных проводников) включают типы данных составляющих элементов и принимают соответствующий им цвет.

- **Cluster** — кластеры включают различные типы данных. Кластерный тип данных отображается коричневым цветом, если все его элементы численные, если же элементы кластера являются данными различных типов, он отображается розовым.
- **Waveform** — сигнальный тип данных является кластером элементов, содержащим данные, начальное значение времени и интервал времени между измерениями.
- **Dynamic** — динамический тип, отображается в виде темно-синих терминалов. Кроме данных сигнала, динамический тип содержит дополнительную информацию, например, название сигнала или дату и время его получения. Большинство экспресс-ВП принимают и/или возвращают данные динамического типа. Данные динамического типа можно направлять к любому элементу отображения или полю ввода, принимающему данные численного, логического или сигнального типа.

Проводники данных

Данные между объектами блок-диаграммы передаются по соединительным линиям – проводникам данных. Проводник данных аналогичен переменным в текстовых языках программирования. Каждый проводник данных имеет единственный источник данных, но может передавать их ко многим ВП и функциям. Проводники данных различаются цветом, стилем и толщиной линии, в зависимости от типа передаваемых данных. Примеры основных типов проводников данных представлены в таблице.

Тип проводника данных	Одно значение	Одномерный (1D) массив	Двумерный (2D) массив	Цвет
Численный				Оранжевый (с плавающей точкой), Голубой (целочисленный)
Логический				Зеленый
Строковые				Розовый

й				
---	--	--	--	--

Автоматическое соединение объектов проводниками данных

В среде LabVIEW объекты соединяются проводниками данных после их помещения на блок-диаграмму. В автоматическом режиме среда LabVIEW подключает те поля ввода/вывода данных, которые наиболее совместимы, несовместимые поля остаются несоединенными.

Соединение объектов проводниками данных вручную

Соединение объектов проводниками данных вручную производится с помощью инструмента СОЕДИНЕНИЕ. После наведения инструмента СОЕДИНЕНИЕ на поле ввода или вывода данных на экране появляется подсказка, которую можно использовать для уточнения места подключения проводника.

Упражнение 3-1. Преобразование °C в °F

Цель: Создать ВП

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



На иллюстрациях, демонстрирующих соединение, стрелка показывает, где произвести щелчок мышью, а число на стрелке – сколько раз необходимо щелкнуть.

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 °C.

е. 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.



f. Поместите цифровой элемент управления на лицевую панель.



г. 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.



а. Выберите цифровой элемент отображения в палитре **Элементов** в подразделе **Controls-Numeric** (Числовые элементы).

б. Поместите элемент отображения данных на лицевую панель.

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



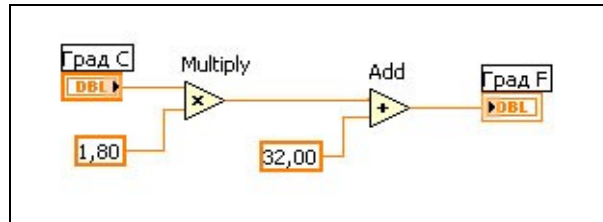
На блок-диаграмме LabVIEW создаст терминалы данных, соответствующие элементам управления и отображения. Терминалы данных представляют тип данных соответствующих элементов. Например, терминал данных DBL, показанный слева, представляет тип числовых данных двойной точности с плавающей запятой.



Внимание. Терминалы данных, соответствующие элементам управления, имеют более широкий обводной контур по сравнению с терминалами данных, соответствующими элементам 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. Для вывода на экран палитры **Functions** (Функций) следует щелкнуть правой кнопкой мыши в рабочем пространстве блок-диаграммы.



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

123

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.

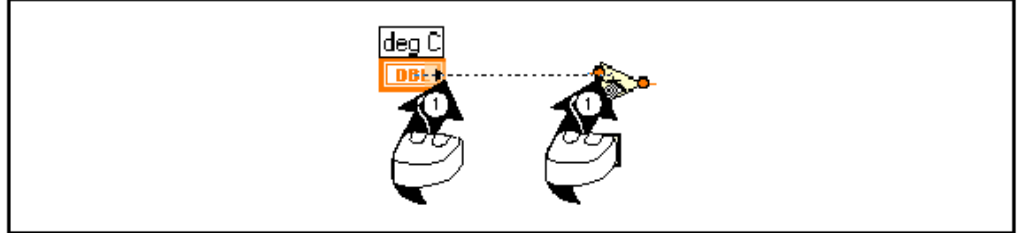
Если значение в константу не введено сразу после ее размещения на блок-диаграмме, следует использовать инструмент ВВОД ТЕКСТА.



9. Соедините объекты блок-диаграммы с помощью инструмента СОЕДИНЕНИЕ, показанного слева:

Для соединения двух объектов на блок-диаграмме инструментом СОЕДИНЕНИЕ следует нажать левой

кнопкой мыши на одном из объектов и, не отпуская кнопку, перевести инструмент на другой объект, как показано на иллюстрации ниже. Последовательность соединения не имеет значения.



Проводник данных можно изгибать, закрепив его щелчком мыши и переместив курсор в перпендикулярном направлении. Нажав пробел, можно переключить направление движения проводника.

Для идентификации полей ввода/вывода данных узла или функции следует щелкнуть на выбранном объекте правой кнопкой мыши и в контекстном меню выбрать пункт **Visible Items-Terminals**. Чтобы вернуться к прежнему виду объекта, следует повторить ту же последовательность действий.

При наведении инструмента СОЕДИНЕНИЕ на поле ввода/вывода данных поле начинает мигать, показывая, что щелчок мыши подключит к нему проводник данных. При этом появляется всплывающая подсказка с указанием имени поля ввода/вывода.

Для отмены начала соединения следует нажать **<Esc>**, либо щелкнуть правой кнопкой мыши или щелкнуть инструментом СОЕДИНЕНИЕ на поле - источник соединения.

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:\exercises\LV Basics I**.



Совет. Все ВП этого курса следует сохранять в каталоге **c:\exercises\LV Basics I**.

- 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**.

Конец упражнения 3-1

В. Редактирование ВП

Существует несколько методов редактирования объектов лицевой панели и блок-диаграммы.

Создание объектов

В дополнение к созданию объектов лицевой панели с помощью палитры **Элементов (Controls)** предусмотрена возможность создания элементов управления и отображения данных, констант по щелчку правой кнопкой мыши на узле. Для этого в контекстном меню следует выбрать пункт **Create**

Constant — создание констант, отображающихся только на блок-диаграмме.

Control — создание элемента управления на лицевой панели ВП.

Indicator — создание элемента отображения данных на лицевой панели.

Выделение объектов

Выделение объектов на лицевой панели и блок-диаграмме производится с помощью инструмента ПЕРЕМЕЩЕНИЕ.

Когда объект выделен, его окружает пунктирная линия. Для выбора нескольких объектов, следует во время их выделения нажать и удерживать клавишу **<Shift>**.

Можно также выделить несколько объектов, щелкнув мышью в свободном пространстве и обведя их курсором.

Перемещение объектов

Перемещение объектов осуществляется при помощи инструмента ПЕРЕМЕЩЕНИЕ. Перемещать объекты можно также при помощи стрелок на клавиатуре. Для перемещения объекта с шагом в несколько пикселей в момент перемещения следует нажать и удерживать клавишу **<Shift>**.

Удаление объектов

Чтобы удалить объект, следует выделить его с помощью инструмента ПЕРЕМЕЩЕНИЕ, после чего нажать на клавиатуре клавишу **<Delete>** или выбрать пункты главного меню **Edit»Clear**.

Отмена и восстановление действий

Если в процессе редактирования ВП была допущена ошибка, можно отменить или восстановить действия, выбрав **Undo** (Отменить) или **Redo** (Восстановить) в пункте главного меню **Edit** (Редактирование).

Копирование объектов

Большинство объектов можно копировать, перемещая выделенный объект и одновременно удерживая клавишу **<Ctrl>**.

После переноса выбранного объекта на новое место, отпускается сначала кнопка мыши, а затем клавиша **<Ctrl>**. В этом месте появляется копия объекта, а первоначальный объект остается на старом месте. Этот процесс называется копированием либо клонированием.

Можно копировать объекты и стандартным способом, выбирая пункты главного меню **Edit>>Copy** и затем **Edit>>Paste**.

Метки объектов

Метки используются для идентификации объектов. Среда LabVIEW имеет два вида меток: свободные и собственные. Собственные метки принадлежат объекту, описывают только его и двигаются вместе с ним. Собственную метку можно перемещать независимо от объекта, но при перемещении объекта метка перемещается вместе с ним. Свободные метки не принадлежат объектам, их можно создавать, перемещать, вращать или удалять независимо. Они используются для описания объектов, ввода комментариев на лицевой панели и блок-диаграмме.

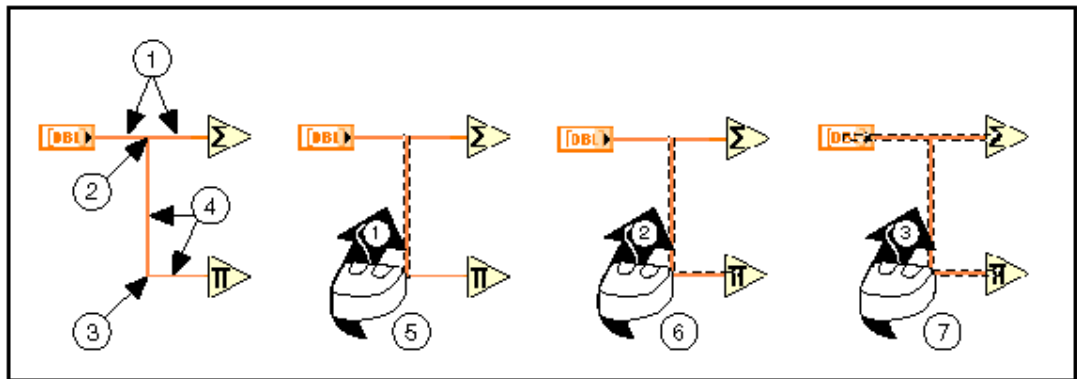
Для создания свободной метки используется инструмент ВВОД ТЕКСТА. Выбрав этот инструмент, необходимо щелкнуть в свободном пространстве одной из панелей и ввести текст. После ввода текста метки поместить курсор в пространство вне метки или нажать кнопку **<Enter>** на инструментальной панели.

Выделение и удаление проводников данных

Сегмент проводника данных – это отдельная горизонтальная

или вертикальная его часть. Место соединения двух сегментов - излом проводника данных. Точка, в которой встречаются два, три или четыре проводника данных называется точкой соединения.

Проводник данных содержит все сегменты между точками соединения, между терминалом данных и точкой соединения, между терминалами данных, если нет точек соединений. Для выделения сегмента используется инструмент ПЕРЕМЕЩЕНИЕ. Двойной щелчок мыши выделяет проводник данных, тройной щелчок - выделяет множество проводников данных:



1 Сегмент

4 Проводник данных

6 Выделенный проводник данных

2 Точка соединения

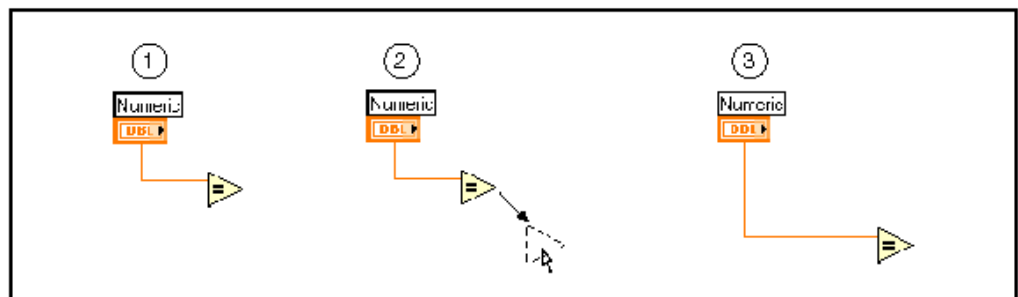
5 Выделенный сегмент

7 Выделенное множество проводников данных

3 Излом проводника данных

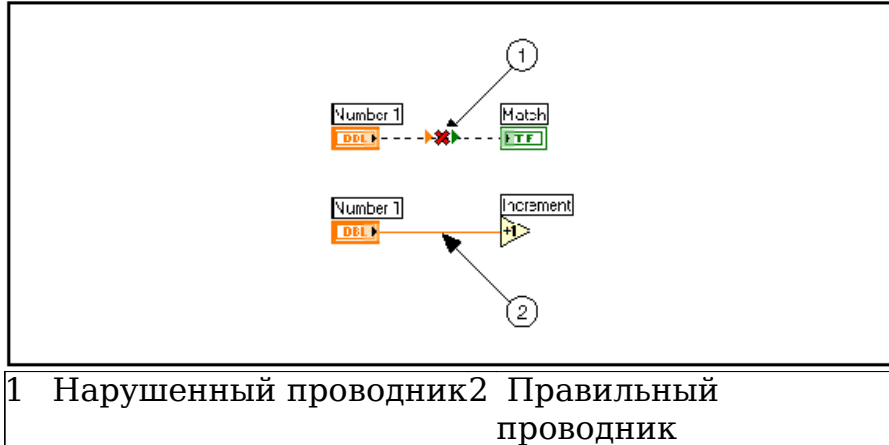
Автомасштабирование проводников данных

Как показано на иллюстрации, перемещение объектов не приводит к нарушению проводника данных.



Разорванные проводники данных

Разорванный проводник данных выглядит, как черная штриховая линия с красным крестом посередине, как показано ниже. Разрыв проводников данных происходит по причинам разного рода. Например, при попытке соединения объектов с несовместимыми типами данных:

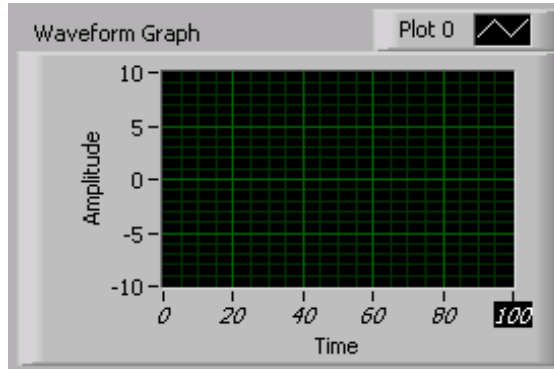


Описание причины разрыва проводника данных появляется в окне всплывающей подсказки после наведения на проводник инструмента СОЕДИНЕНИЕ. Тройной щелчок инструментом ПЕРЕМЕЩЕНИЕ на проводнике и последующее нажатие клавиши **<Delete>** удаляет выделенный проводник. Удаление всех разорванных проводников производится через пункт главного меню **Edit»Remove Broken Wires**.

Редактирование текста (изменение шрифта, стиля и размера)

Выбрав пункт меню **Text Settings** на инструментальной панели, можно изменить шрифт, стиль, размер и провести выравнивание любого текста внутри меток или на дисплеях элементов управления и отображения.

На некоторых элементах управления и отображения данных, текст может быть помещен более чем в одном месте, например оси графиков. В этом случае текст в каждом поле можно изменять независимо. Текст выделяется инструментом ВВОД ТЕКСТА, как показано на рисунке, и на инструментальной панели выбирается пункт меню **Text Settings**.



Изменение размеров объектов



Большинство объектов лицевой панели допускают изменение размеров. Чтобы подготовить объект к изменению размера, необходимо навести на него инструмент ПЕРЕМЕЩЕНИЕ. По углам объекта появляются маркеры, показанные слева. Затем курсор следует установить на один из маркеров и, удерживая нажатой левую кнопку мыши, переместить маркер, размер шрифта при этом не меняется. Промежуточные границы изменяемого размера обозначаются штриховой линией. Когда нужный размер элемента достигнут, кнопку мыши следует отпустить. Удержание клавиши **<Shift>** во время перемещения маркеров сохраняет пропорции объекта.

Можно изменять размеры и объектов блок-диаграммы, таких как структуры и константы.

Выравнивание и распределение объектов в пространстве

Выравнивание группы объектов по оси производится с помощью опций в пункте инструментальной панели **Align Objects**. Для равномерного распределения объектов в пространстве следует воспользоваться пунктом **Distribute Objects**.

Установка порядка размещения объектов, объединение объектов в группу и закрепление местоположения объектов на рабочем пространстве лицевой панели

В случае, когда объекты перекрывают друг друга, можно установить порядок размещения объектов - один впереди другого. Для этого объект следует выделить с помощью инструмента ПЕРЕМЕЩЕНИЕ и в пункте меню **Reorder**

инструментальной панели выбрать необходимые установки: **Move Forward** (Поместить на передний план), **Move Backward** (Поместить на задний план), **Move To Front** (Передвинуть вперед), **Move To Back** (Передвинуть назад).

Для объединения объектов в группу и закрепления их местоположения на рабочем пространстве лицевой панели следует выбрать необходимые установки в пункте меню **Reorder** инструментальной панели: **Group** (Группировать), **Ungroup** (Разгруппировать), **Lock** (Блокировать), **Unlock** (Разблокировать).

Приведение нескольких объектов к одному размеру

Приведение нескольких объектов к одному виду производится с помощью выпадающего меню **Resize Objects** (Изменение размеров объектов). Предусмотрена возможность изменения размера всех выбранных объектов по ширине или высоте до ширины/высоты наименьшего или наибольшего объекта, также имеется возможность задать размер всех выбранных объектов в пикселях.

Отдельные объекты допускают изменения размера лишь по вертикали или горизонтали, например, числовые элементы управления и отображения; некоторые объекты сохраняют пропорции при изменении размера. Например, если среди объектов, выбранных для изменения размера по высоте, присутствует числовая константа, LabVIEW не изменит ее размер, изменив размер остальных объектов, допускающих изменение размера.

Копирование объектов между ВП или между другими приложениями

Копировать и вставлять объекты из одного ВП в другой можно выбором пунктов главного меню **Edit»Copy** и затем **Edit»Paste**. Возможно копирование изображения или текста из других приложений и их использование на лицевой панели или блок-диаграмме. Если оба ВП открыты, можно копировать выбранные объекты, перемещая их с одного ВП на другой.

Окрашивание объектов

Можно изменять цвет большинства объектов ВП, но не всех. Например, терминалы данных и проводники данных блок-диаграммы используют только определенные цвета, соответствующие типу представленных данных.

Изменение цвета объекта или фона рабочего пространства производится с помощью инструмента РАСКРАШИВАНИЕ. Для этого следует щелкнуть правой кнопкой мыши на выбранном элементе или рабочем пространстве любой из панелей. Можно изменить заданные по умолчанию цвета большинства объектов, выбирая пункты меню **Tools-Options** и затем **Colors**.

Можно также сделать объект прозрачным, выбрав **T** в меню **Colors**.

Упражнение 3-2. ВП Редактирование

Цель: Отредактировать ВП

Ниже приведена последовательность действий для изменения существующего ВП Редактирование таким образом, чтобы его лицевая панель выглядела, как показано ниже. Также необходимо соединить объекты на блок-диаграмме для приведения ВП в рабочее состояние.

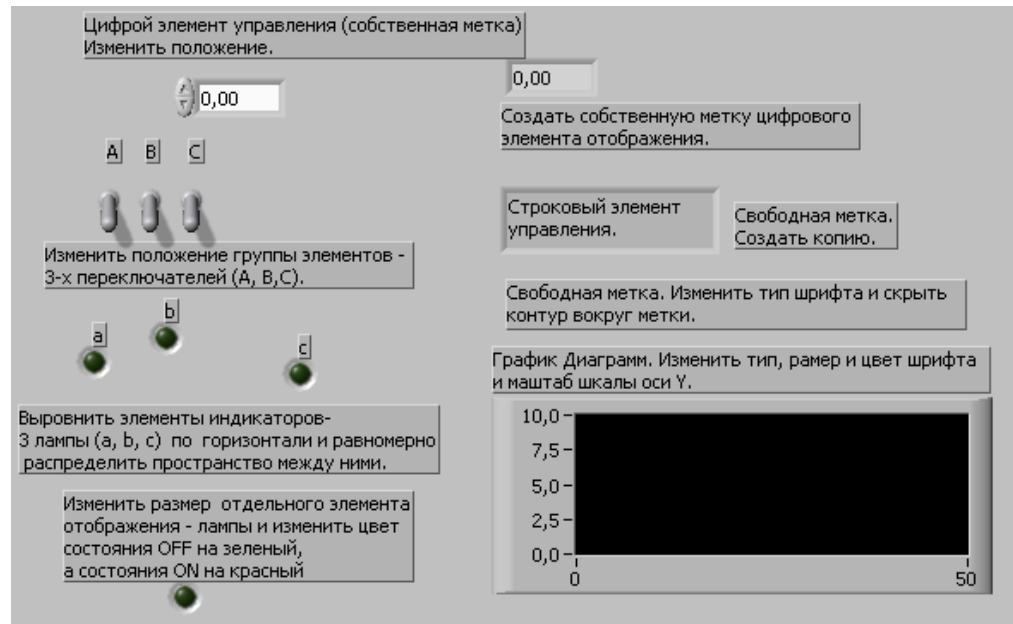


Примечание. В случае совершения ошибки можно воспользоваться пунктом меню **Edit-Undo**

Лицевая панель

1. Выберите пункт главного меню **File>>Open**, укажите папку **c:\exercises\LV basics I** и откройте *Редактирование.vi*

В случае если все ВП закрыты, следует воспользоваться кнопкой **Open VI** диалогового окна **LabVIEW**.



2. Измените положение элемента **Цифровой элемент управления**.



- a. С помощью инструмента **ПЕРЕМЕЩЕНИЕ**, показанного слева, выделите элемент и переместите его на новое место. Собственная метка элемента переместится вместе с ним.

- b. Щелкните левой кнопкой мыши по свободному пространству лицевой панели, чтобы снять выделение с элемента.
 - c. Выделите собственную метку элемента и переместите ее на новое место. Элемент останется неподвижным. Собственную метку элемента можно перемещать независимо от элемента. Собственная метка следует за своим элементом в случае его перемещения.
3. Измените положение трех вертикальных переключателей одновременно.
- a. С помощью инструмента ПЕРЕМЕЩЕНИЕ выделите область лицевой панели, охватив все три переключателя.
 - b. Нажмите мышью на одном из них и переместите на новое место. При этом переместятся все три переключателя.
4. Выверните три светодиода по горизонтали, равномерно распределите пространство между ними, сгруппируйте и закрепите их местоположение на лицевой панели.
- a. С помощью инструмента ПЕРЕМЕЩЕНИЕ выделите область лицевой панели, охватив все три светодиода.
 - b. Чтобы выровнять элементы по горизонтали, в пункте меню инструментальной панели **Align Objects** выберите опцию **Vertical Centers**, показанную слева.
 - c. Чтобы равномерно распределить пространство между элементами, в пункте меню инструментальной панели **Distribute Objects** выберите опцию **Horizontal Centers**, показанную слева.
 - d. Чтобы сгруппировать элементы, в пункте меню инструментальной панели **Reorder** выберите опцию **Group**.
 - e. Чтобы закрепить местоположение элементов на лицевой панели, в пункте меню инструментальной панели **Reorder** выберите опцию **Lock**.
 - f. Теперь попробуйте переместить один из светодиодов. Следует обратить внимание, что светодиоды выделяются как группа и переместить их невозможно, пока их местоположение закреплено.



5. Измените размер отстоящего круглого светодиода.
 - a. Наведите на светодиод инструмент ПЕРЕМЕЩЕНИЕ. По краям светодиода появятся круглые маркеры.
 - b. Захватите курсором маркер и измените размер светодиода. Если при этом нажать и удерживать клавишу **<Shift>**, светодиод будет изменять свои размеры, сохраняя пропорции.
6. Измените цвет этого светодиода.

- a. По умолчанию цвет светодиода в режиме **OFF** является темно-зеленым (значение FALSE). С помощью инструмента УПРАВЛЕНИЕ, показанного слева, переведите светодиод в режим ON. Его цвет изменится на светло-зеленый (значение TRUE).



- b. С помощью инструмента ОКРАШИВАНИЕ, показанного слева, щелкните правой кнопкой мыши по светодиоду для вывода цветовой палитры.
- c. Для режима **ON** установите красный цвет.

7. Для цифрового элемента управления, расположенного в правой верхней части лицевой панели, выведите собственную метку.



- a. С помощью инструмента ВВОД ТЕКСТА, показанного слева, щелкните правой кнопкой мыши на элементе и выберите из контекстного меню пункт **Visible Item-Label**.

Появится маленькое окно ввода текста с курсором в левой части, готовое к вводу текста.

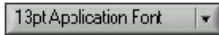
- b. Напечатайте **Цифровой элемент отображения данных**.



- c. Щелкните мышью в пространстве вне метки или нажмите кнопку **Enter**, показанную слева, для завершения ввода текста.

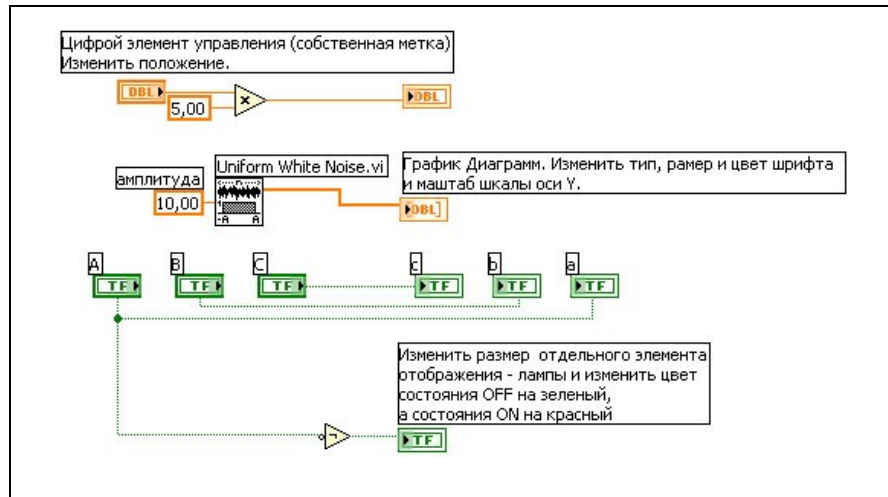
8. Удалите строковый элемент управления.

- a. С помощью инструмента ПЕРЕМЕЩЕНИЕ выделите строковый элемент управления.
 - b. Нажмите клавишу **<Delete>** или выберите пункт главного меню **Edit - Clear**.
9. Размножьте свободную метку.
- a. Нажмите и удерживайте клавишу **<Ctrl>**, с помощью инструмента ПЕРЕМЕЩЕНИЕ передвиньте метку.
 - b. Поместите копию метки на новое место.
10. Измените шрифт свободной метки и скройте поле вокруг нее.
- a. С помощью инструмента ПЕРЕМЕЩЕНИЕ выделите свободную метку.
 - b. Измените текст, выбрав показанный слева пункт **Text Settings** инструментальной панели.
 - c. С помощью инструмента ОКРАШИВАНИЕ щелкните правой кнопкой мыши на метке и в цветовой палитре выберите **T**.
11. Измените текст и цвет оси Y.
- a. С помощью инструмента ВВОД ТЕКСТА выделите **10,0** по оси Y.
 - b. Выбрав пункт инструментальной панели **Text Settings**, измените размер и цвет текста.
12. Двойным щелчком левой кнопкой мыши выделите **0,0** и введите значение **-10,0**, чтобы изменить диапазон оси Y.



Блок-диаграмма

13. Перейдите на блок-диаграмму, выбрав пункт главного меню **Window-Show Diagram**. Соедините терминалы данных блок-диаграммы, как показано ниже:



Функция **Умножение (Multiply)** умножает числовую константу «5.00» на значение, введенное в числовой элемент управления.



Узел **Uniform White Noise VI** генерирует белый шум в диапазоне $[-a:a]$, где a - абсолютное значение амплитуды шума (**amplitude**) «10.00», и передает данные на график осциллограмм.



Функция **Not** (Логическое нет) инвертирует значение логического выключателя **A** и передает значение на светодиод.



14. Для создания константы щелкните правой кнопкой мыши по нижнему полю ввода данных функции **Multiply** (Умножение) и выберите из контекстного меню пункт **Create - Constant**.

Введите значение «5.0» в поле ввода текста и нажмите кнопку **Enter** на инструментальной панели.



15. Используя инструмент СОЕДИНЕНИЕ, соедините проводниками данных объекты блок-диаграммы. Для этого:

Войдите в пункт главного меню **Help»Show Context Help** и выведите на экран окно **Context Help** (контекстной справки). Окно **Context Help** (контекстной справки) поможет определить поля ввода данных, необходимые для соединения. Обязательные для соединения поля выделены жирным текстом, рекомендуемые для соединения

поля обозначены обычным текстом, а дополнительные (необязательные) помечены светло-серым.

Для идентификации полей ввода/вывода данных следует щелкнуть правой кнопкой мыши по терминалу функции, и в контекстном меню выбрать пункт **Visible Items-Terminal**. Когда проводники данных подключены, войдите в пункт контекстного меню **Visible Items - Terminal**, чтобы вернуть терминалу функции прежний вид.

Для добавления ответвления к проводнику данных, щелкните в выбранном месте проводника.

Для отмены начала соединения нажмите клавишу **<Esc>**, щелкните правой кнопкой мыши или нажмите на источник соединения.

16. Выберите пункт главного меню **File»Save**, чтобы сохранить ВП.
17. Перейдите на лицевую панель, выбрав пункт главного меню **Window-Show Panel**.
18. С помощью инструмента УПРАВЛЕНИЕ измените значения элементов управления.
19. Нажмите кнопку **Run** на инструментальной панели, чтобы запустить ВП.
20. Выберите пункт главного меню **File»Close**, чтобы закрыть ВП.

Конец упражнения 3-2

С. Отладка ВП



Если ВП не запускается, это означает, что он не готов к работе. В процессе создания или редактирования ВП кнопка **Run** принимает вид разорванной стрелки, как показано слева. Если после завершения редактирования блок-диаграммы стрелка все еще имеет разорванный вид, то ВП работать не будет.

Поиск ошибок

Нажмите кнопку **Run** или выберите пункт главного меню **Windows»Show Error List**, чтобы вывести на экран окно **Список ошибок**, в котором перечислены все допущенные ошибки. После двойного щелчка левой кнопкой мыши на описании ошибки выделится объект, содержащий эту ошибку.

Режим анимации выполнения ВП



Режим анимации выполнения блок-диаграммы активируется щелчком правой кнопки мыши по кнопке **Highlight Execution**, показанной слева. После нажатия кнопки «лампочка» загорится - режим активирован. Выполнение ВП в этом режиме сопровождается подсветкой движения данных по блок-диаграмме от одного узла к другому. При этом числовые значения передаваемых данных будут отображаться на экране в виде всплывающих окон. Этот режим используется для пошаговой отладки ВП и наблюдения за выполнением блок-диаграммы.



Примечание. Режим анимации замедляет скорость выполнения ВП.

Режим пошаговой отладки ВП

Режим пошаговой отладки ВП используется для просмотра выполнения ВП на блок-диаграмме. Активация пошагового режима осуществляется нажатием кнопок **Step Over** или **Step Into** на инструментальной панели. Чтобы увидеть подсказку, следует поместить курсор поверх кнопок **Step Over, Step Into** или **Step Out**. Подсказка описывает событие, которое последует после нажатия этой кнопки. Пошаговый режим можно использовать и для просмотра выполнения подпрограммы ВП.



При использовании пошагового режима отладки ВП в режиме анимации на иконке подпрограммы ВП появится зеленая стрелка, как показано слева. Зеленая стрелка показывает, что данная подпрограмма ВП в данный момент времени выполняется.

Отладочные индикаторы



Инструмент УСТАНОВКА ОТЛАДОЧНЫХ ИНДИКАТОРОВ, показанный слева, предназначен для проверки промежуточного значения данных в проводнике данных в процессе выполнения ВП. В режиме пошагового выполнения или при остановке в контрольной точке с помощью отладочных индикаторов можно визуализировать значения данных в проводнике или поле ввода/вывода узла данных, если узел уже получил свое значение.

Можно установить несколько локальных отладочных индикаторов для одновременного наблюдения за данными в разных точках блок-диаграммы. Чтобы создать локальный отладочный индикатор, следует щелкнуть правой кнопкой мыши на выбранном проводнике данных и выбрать в контекстном меню пункт **Custom Probe**.

Контрольные точки



Инструмент ВВОД КОНТРОЛЬНОЙ ТОЧКИ, показанный слева, предназначен для размещения контрольных точек в узлах или проводниках данных блок-диаграммы. В месте установки контрольной точки в момент прохождения через нее данных возникает пауза в выполнении программы. Когда ВП приостановил свое выполнение в контрольной точке, LabVIEW подсвечивает узел или проводник данных в месте установки контрольной точки. LabVIEW обводит красной границей узел и блок-диаграмму и отмечает красным маркером проводник данных. После наведения курсора на контрольную точку, черное поле инструмента ВВОД КОНТРОЛЬНОЙ ТОЧКИ становится белым. Для удаления существующей контрольной точки по ней следует щелкнуть инструментом ВВОД КОНТРОЛЬНОЙ ТОЧКИ.

Упражнение 3-3. Отладка ВП

Цель: Приобретение практических навыков отладки ВП

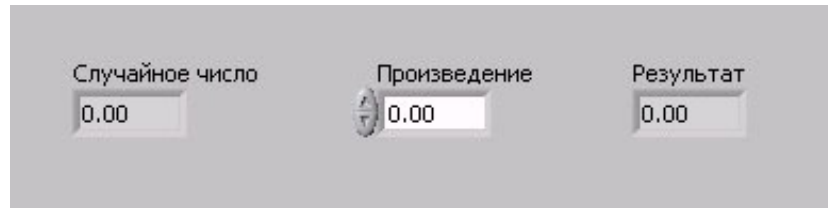
Ниже приведены последовательность действий для загрузки содержащего ошибки ВП, использование режимов анимации и пошаговой отладки для исправления ошибок.

Лицевая панель

1. Выберите пункт главного меню **File-Open** и перейдите в каталог `c:\exercises\LV Basics I`, чтобы открыть *Отладка ВП (Главная).vi*

Если все ВП закрыты, следует воспользоваться кнопкой **Open VI** в диалоговом окне **LabVIEW**.

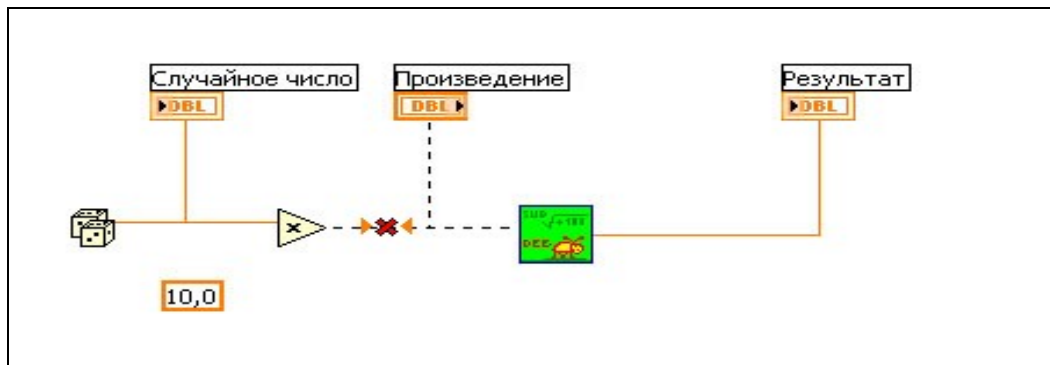
Появится следующая лицевая панель:



Кнопка **Run**, показанная слева, имеет вид разорванной стрелки. Это говорит о том, что ВП не может работать.

Блок-диаграмма

2. Выберите пункт главного меню **Window - Show Diagram**, чтобы перейти на блок-диаграмму.



Функция **Random Number (0-1)** (Генератор случайных чисел) - генерирует случайные числа от 0

до 1.



Функция **Multiply** (Умножение) - умножает случайное число на 10,0.



В этой числовой константе введено значение, на которое умножается случайное число.



ВП *Отладка ВП (Подпрограмма)* - добавляет 100.0 и вычисляет квадратный корень из полученного значения.

3. Найдите все ошибки.
 - a. Нажмите кнопку **Run**. В окне **Error list** (Список ошибок) перечислены все допущенные ошибки.
 - b. Для получения информации об ошибках нажмите на описание каждой из них.
 - c. Нажмите кнопку **Show Error**, чтобы перейти на место текущей ошибки.
 - d. Подробная информация об ошибке находится в окне **Details**.
4. Выберите пункт главного меню **File»Save**, чтобы сохранить ВП.
5. Перейдите на лицевую панель, выбрав пункт главного меню **Window»Show Panel**.

Запуск ВП

6. Несколько раз запустите ВП, нажимая кнопку **Run**.
7. Выберите пункт главного меню **Window-Show Diagram**, чтобы перейти на блок-диаграмму
8. Включите режим анимации выполнения ВП.



a. Нажмите кнопку **Highlight Execution**, показанную слева, чтобы включить режим анимации выполнения.



b. Нажмите кнопку **Step Into**, показанную слева, чтобы включить режим пошаговой отладки. Режим анимации показывает передвижение данных на блок-диаграмме от узла к узлу с помощью подвижных точек на проводниках данных. Узлы мигают, показывая готовность к выполнению.



c. Нажимать кнопку **Step Over**, показанную слева, следует после каждого узла для передвижения по

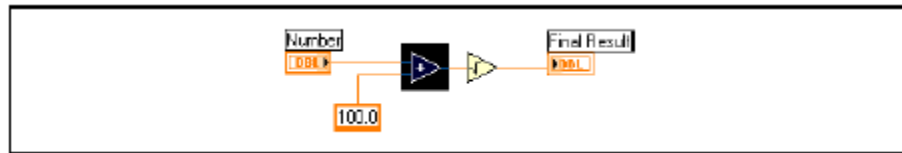
всей блок-диаграмме. Каждый раз после нажатия кнопки **Step Over**, узел выполняется и наступает пауза перед выполнением следующего, готового к выполнению узла. Во время выполнения ВП на лицевой панели отображаются данные. ВП генерирует случайные числа и умножает их на 10,0. Подпрограмма ВП прибавляет 100,0 и вычисляет из результата корень квадратный.



d. Когда начнет мигать контур блок-диаграммы, нажмите кнопку **Step Out**, показанную слева, чтобы выключить режим пошаговой отладки ВП *Отладка ВП (Главная).vi*

9. В режиме пошаговой отладки просмотрите подпрограмму ВП.

- a. Нажмите кнопку **Step Into**, чтобы включить режим пошаговой отладки.
- b. Когда ВП *Отладка ВП (подпрограмма).vi* начнет мигать, нажмите кнопку **Step Into**. Появится следующая блок-диаграмма:



c. Перейдите на блок-диаграмму ВП *Отладка ВП (Главная).vi*, нажав на нее.

Зеленая стрелка на иконке ВП *Отладка ВП (Главная).vi*, показанная слева, указывает на режим пошаговой отладки.



- d. Перейдите на блок-диаграмму ВП *Отладка ВП (подпрограмма) .vi*
- e. Дважды нажмите кнопку **Step Out** для завершения режима пошаговой отладки для подпрограммы ВП. Блок-диаграмма ВП *Отладка ВП (Главная).vi* останется активной.
- f. Нажмите кнопку **Step Out** для завершения режима пошаговой отладки.

10.С помощью отладочных индикаторов посмотрите текущее значение данных в проводнике.

- a. Инструментом **УСТАНОВКА ОТЛАДОЧНЫХ ИНДИКАТОРОВ**, показанным слева, нажмите на любом объекте. Появится следующее окно.



Номер в заголовке окна **Probe** соответствует номеру отладочного индикатора, поставленного на блок-диаграмме.

- b. Снова включите режим пошаговой отладки. Окно **Probe** покажет данные по мере их прохождения через отладочный индикатор.

11. Установите контрольные точки для приостановки выполнения ВП в этом месте.



- a. Инструментом **ВВОД КОНТРОЛЬНОЙ ТОЧКИ** щелкните на узле или проводнике данных. Щелчок по рабочему пространству блок-диаграммы означает остановку выполнения в самом начале выполнения ВП.

- b. Нажмите кнопку **Run**, чтобы запустить ВП. ВП остановится на контрольной точке.



- c. Нажмите кнопку **Continue**, показанную слева, чтобы продолжить выполнение ВП.

- d. Инструментом **ВВОД КОНТРОЛЬНОЙ ТОЧКИ** щелкните по установленной контрольной точке, чтобы удалить ее.

12. Нажмите кнопку **Highlight Execution**, чтобы отключить режим анимации выполнения ВП.

13. Войдите в пункт главного меню **File»Close**, чтобы закрыть ВП и все открытые окна.

Конец упражнения 3-3

Краткое изложение пройденного материала, советы и секреты

Лицевая панель создается с помощью элементов управления и отображения данных, которые являются интерактивными полями ввода и вывода данных ВП соответственно.

Блок-диаграмма состоит из узлов, терминалов данных и проводников.

Кнопка **Search** на палитрах **Controls** (Элементов) и **Functions** (Функций) предназначена для поиска элементов, ВП и функций.

Кнопка **Run** в виде разорванной стрелки появляется на инструментальной панели, когда ВП не готов к работе. После нажатия кнопки **Run** на экран выводится окно **Error list** (Список ошибок), в котором перечислены все ошибки.

Режимы анимации выполнения ВП и пошаговой отладки, отладочные индикаторы и контрольные точки предназначены для отладки ВП в режиме просмотра потока данных на блок-диаграмме.

Советы и секреты

Действие

Часто используемые пункты меню имеют эквивалентные «горячие» клавиши.

Например, чтобы сохранить ВП, необходимо войти в пункт главного меню **File»Save** или нажать клавиши **<Ctrl-S>** на клавиатуре. Основные «горячие» клавиши:

- **<Ctrl-R>** Запуск ВП.
- **<Ctrl-E>** Переключение между лицевой панелью и блок-диаграммой.
- **<Ctrl-H>** Вывод на экран и скрытие окна **Context Help** (контекстной справки).
- **<Ctrl-B>** Удаление всех разорванных проводников данных.
- **<Ctrl-F>** Поиск ВП, глобальных переменных, функций, текста или других объектов, находящихся в памяти ВП.

Для выбора инструментов на палитре **Tools** (Инструментов) используется клавиша **<Tab>**.

На блок-диаграмме, для переключения между инструментом ПЕРЕМЕЩЕНИЕ и инструментом СОЕДИНЕНИЕ, а на лицевой панели – между инструментом ПЕРЕМЕЩЕНИЕ и инструментом УПРАВЛЕНИЕ используется клавиша пробел.

Для быстрого изменения значений элементов управления инструментом УПРАВЛЕНИЕ или ВВОД ТЕКСТА курсор помещается в поле элемента и, удерживая нажатой клавишу **<Shift>**, нажимается одна из стрелок приращения значений.

Можно отключить средства отладки и тем самым уменьшить требования к оперативной памяти. Для этого необходимо выбрать пункт главного меню **File»VI Properties** и войти в раздел **Execution** выпадающего меню, после чего снять метку в **Allow Debugging**.

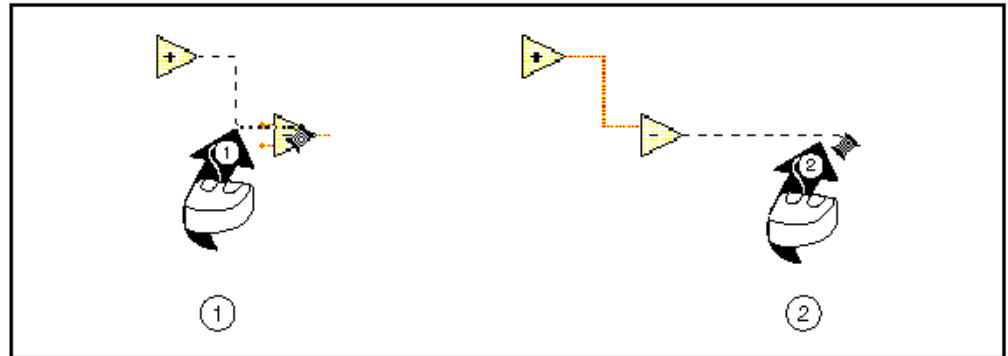
Проводники данных

Выберите пункт главного меню **Help»Show Context Help**, чтобы вывести на экран окно **Context Help** (Контекстной справки). Окно **Context Help** (контекстной справки) поможет определить поля ввода/вывода данных,

обязательные для соединения. Поля ввода/вывода данных, рекомендованные для соединения, представлены обычным текстом, дополнительные (необязательные) поля выделены серым цветом.

Пробел переключает направление движения проводника данных.

Проводник данных можно изогнуть, зафиксировав излом щелчком мыши. Фиксация и разрыв проводника производится двойным щелчком мыши:



1 Фиксация излома проводника	2 Фиксация и разрыв проводника двойным щелчком мыши
------------------------------	---

Перемещение объекта на один пиксель проводится стрелками на клавиатуре. Для быстрого перемещения объекта сразу на несколько пикселей необходимо нажать и удерживать клавишу **<Shift>**.

Для отмены начала соединения следует нажать **<Esc>**, щелкнуть правой кнопкой мыши или вернуться на источник соединения.

После наведения инструмента СОЕДИНЕНИЕ на поле ввода или вывода данных на экране появляется быстрая подсказка, которую можно использовать для уточнения места закрепления проводника.

Отображение соединительной панели терминала объекта осуществляется пунктом контекстного меню **Visible Items»Terminals**.

Редактирование

Ускоренное создание констант, элементов управления и отображения данных:

- Щелкните правой кнопкой мыши на поле ввода или вывода данных функции и выберите **Create»Constant**, **Create»Control** или **Create»Indicator** в контекстном меню.
- Для создания константы с помощью мыши переместите элемент управления или отображения данных с лицевой панели на блок-диаграмму.
- Для создания элемента управления с помощью мыши переместите константу на лицевую панель.

Для копирования объектов нажмите и удерживайте клавишу **<Ctrl>** во время перемещения выделенного объекта инструментом ПЕРЕМЕЩЕНИЕ.

Чтобы ограничить направление движения объекта по горизонтали или по вертикали, во время перемещения нажмите и удерживайте клавишу **<Shift>**.

Чтобы сохранить пропорции во время изменения размера объекта, нажмите и удерживайте клавишу **<Shift>**.

Чтобы изменить размер элемента во время размещения его на лицевой панели, нажмите и, удерживая клавишу **<Ctrl>**, передвиньте маркеры изменения размера.

Для замены узлов выберите пункт **Replace** в контекстном меню.

Чтобы перейти на блок-диаграмму подпрограммы ВП, удерживая нажатой клавишу **<Ctrl>**, дважды щелкните по иконке подпрограммы инструментом ПЕРЕМЕЩЕНИЕ или УПРАВЛЕНИЕ.

Чтобы перейти на лицевую панель подпрограммы ВП, нужно дважды щелкнуть на иконке подпрограммы инструментом ПЕРЕМЕЩЕНИЕ или УПРАВЛЕНИЕ. Другой способ - выберите пункт главного меню **Browse»This VI's SubVIs**.

Для окончания ввода текста следует нажать **<Shift-Enter>**.

Для быстрого добавления варианта в элемент **ring** (Список выбора) и структуру **Case** (Варианта), нажмите клавиши **<Shift-Enter>**. Нажатие клавиш **<Shift-Enter>** добавляет вариант и позиционирует курсор в месте добавления следующего варианта. Подробная информация о работе структуры **Case** (Варианта) представлена в Уроке 8 «Принятие решений в ВП».

Для копирования цвета объекта и передачи этого цвета другому объекту следует воспользоваться инструментом КОПИРОВАНИЕ ЦВЕТА. Щелкните инструментом на объекте, цвет которого необходимо копировать, затем щелкните на объект, которому требуется присвоить выбранный цвет.

Если в процессе редактирования ВП допущена ошибка, следует выбрать пункт главного меню **Edit»Undo**.

Для расширения свободного пространства блок-диаграммы следует нажать клавишу **<Ctrl>** и, удерживая ее, инструментом ПЕРЕМЕЩЕНИЕ на свободном пространстве обвести область расширения.

Занятие 4.

Создание подпрограмм ВП

Представлена последовательность действий по редактированию иконки ВП, а также настройки соединительной панели (области полей ввода/вывода данных), что позволяет использовать виртуальный прибор как подпрограмму в других ВП.

Изложены вопросы:

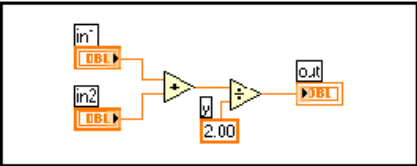
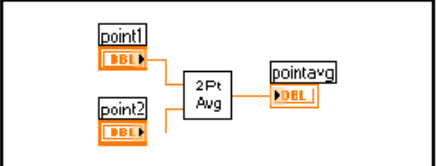
- A. Подпрограмма ВП.
- B. Создание иконки ВП и настройка соединительной панели.
- C. Использование виртуального прибора в качестве подпрограммы ВП.
- D. Преобразование экспресс-ВП в подпрограмму ВП

Е. Превращение выделенной секции блок-диаграммы ВП в подпрограмму ВП.

А. Подпрограммы ВП

После того как ВП сформирован, создана его иконка и настроена соединительная панель, виртуальный прибор можно использовать как подпрограмму в других ВП. Виртуальный прибор, используемый внутри другого виртуального прибора, называется подпрограммой ВП. Подпрограмма ВП соответствует подпрограмме в текстовых языках программирования. Узел подпрограммы ВП соответствует вызову подпрограммы в текстовых языках программирования. Узел – это графическое представление подпрограммы ВП, а не собственно исполняемый код подпрограммы ВП, так же как вызов подпрограммы в текстовых языках программирования не есть сам исполняемый код подпрограммы. Использование подпрограмм ВП помогает быстро управлять изменениями и отладкой блок-диаграмм. Более подробная информация о разработке приложений изложена в курсе **LabVIEW Основы II**.

Для демонстрации аналогии между подпрограммой ВП и подпрограммой текстовых языков программирования ниже представлены текстовый аналог кода и блок-диаграмма:

Исходный текст	Вызов подпрограммы
<pre>function average (in1,in2,out) { out = (in1 + in2) / 2.0; }</pre>	<pre>main { average(point1,point2,pointavg); }</pre>
	

В. Создание иконки ВП и настройка соединительной панели

Следующий шаг после создания блок-диаграммы и формирования лицевой панели ВП – создание иконки ВП и настройка соединительной панели для использования виртуального прибора в качестве подпрограммы ВП.

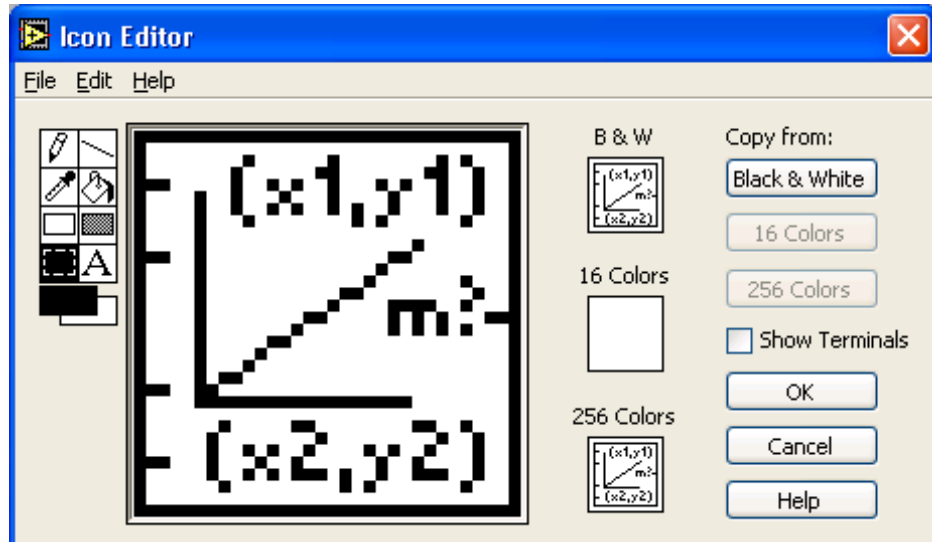
Создание иконки ВП



Каждый виртуальный прибор в правом верхнем углу лицевой панели и в окне блок-диаграммы отображает иконку, показанную слева. Иконка – графическое представление прибора. Она может содержать текст, рисунок или и то и другое одновременно. Если ВП используется в качестве подпрограммы, то иконка идентифицирует его на блок-диаграмме другого ВП.

Установленная по умолчанию иконка ВП содержит номер, который указывает, сколько новых приборов открылись после запуска LabVIEW. Создать собственную иконку, отличную от заданной по умолчанию, можно, щелкнув правой кнопкой мыши по иконке в правом верхнем углу лицевой панели или блок-диаграммы. Затем выбрать пункт **Edit Icon** (Редактирование иконки) из контекстного меню. **Icon Editor** (Редактор иконки) можно также вызвать двойным щелчком левой кнопки мыши в верхнем правом углу одной из панелей. Редактирование иконки доступно также из пункта главного меню **File**, далее **VI Properties** (Свойства ВП), где в диалоговом окне **Category** (Категория) следует выбрать пункт **General** (Общие) и нажать кнопку **Edit Icon** (Редактирование иконки).

Проектирование иконки выполняется в области редактирования, расположенной в центре окна **Icon Editor** (Редактора иконки), при помощи инструментов, расположенных слева от области редактирования. Вид иконки и доступный на блок-диаграмме и в правом верхнем углу обеих панелей размер иконки появляется справа от области редактирования, в соответствующем поле, как показано ниже.



В зависимости от типа монитора, иконка может быть создана для черно-белого, 16-цветного или 256-цветного режима. Для печати, в случае отсутствия цветного принтера, LabVIEW использует черно-белую иконку. По умолчанию установлен 256-цветный режим.

Меню **Edit** (редактирование) используется для вырезания, копирования и вставки картинок из иконки или в нее. При выборе фрагмента иконки для вставки картинки LabVIEW изменяет размер картинки для соответствия размеру выбранной области.

Предусмотрена возможность перемещения графических символов из файловой системы в верхний правый угол лицевой панели или блок-диаграммы. LabVIEW автоматически преобразует изображение в иконку размером 32×32 точки.

Для копирования цветной иконки в черно-белую (или наоборот) достаточно выбрать опцию **Copy from**, находящуюся в правой части диалогового окна **Icon Editor**. Нажать кнопку **OK** для окончательной замены.



Внимание. В случае если сплошная граница вокруг иконки не нарисована, фон иконки будет прозрачным. При выборе иконки на блок-диаграмме маркеры выбора появляются вокруг каждого графического элемента иконки.

Набор инструментов для редактирования иконки расположен в левой части окна **Icon Editor** и выполняет следующие задачи:



Инструмент **КАРАНДАШ** позволяет рисовать или стирать по одной точке.



Инструмент ЛИНИЯ позволяет рисовать прямые линии. Для рисования вертикальных, горизонтальных и диагональных линий необходимо во время рисования нажать и удерживать клавишу **<Shift>**.



Инструмент КОПИРОВАНИЕ ЦВЕТА предназначен для копирования цвета символа в поле редактирования иконки.



Инструмент ЗАПОЛНЕНИЕ ЦВЕТОМ предназначен для заполнения ограниченной области заданным цветом переднего плана.



Инструмент ПРЯМОУГОЛЬНИК выводит в область редактирования прямоугольную границу заданным цветом переднего плана.

Двойной щелчок левой кнопкой мыши на ПРЯМОУГОЛЬНИК обводит иконку рамкой заданным цветом переднего плана.



Инструмент ЗАПОЛНЕННЫЙ ЦВЕТОМ ФОНА ПРЯМОУГОЛЬНИК выводит в область редактирования прямоугольную границу заданным цветом переднего плана, заполненную цветом фона. Двойной щелчок левой кнопкой мыши на ЗАПОЛНЕННОМ ЦВЕТОМ ФОНА ПРЯМОУГОЛЬНИК обводит иконку рамкой цвета символа и заполняет цветом фона.



Инструмент ВЫБОР предназначен для выделения фрагмента иконки, что позволяет вырезать, копировать, перемещать или вносить другие изменения в выделенный фрагмент. Чтобы очистить область редактирования иконки достаточно дважды щелкнуть левой кнопкой мыши на инструменте ВЫБОР и нажать кнопку **<Delete>**.



Инструмент ВВОД ТЕКСТА позволяет вводить текст в область редактирования иконки. Выбор шрифта производится двойным щелчком левой кнопкой мыши на инструменте ВВОД ТЕКСТА. (**Windows**) Доступна опция «**Small Fonts**».



Инструмент ПЕРЕДНИЙ ПЛАН/ФОН отображает цвета фона и переднего плана (символа). При нажатии на каждый прямоугольник появляется палитра выбора цвета.

Опции в правой части **Icon Editor** предназначены для выполнения следующих задач:

Show Terminals - выводит в область редактирования поля ввода/вывода данных.

OK - сохраняет внесенные в иконку изменения

Cancel - закрывает **Icon Editor** без сохранения.

Строка меню в окне **Icon Editor** содержит опции редактирования, такие как **Undo** (Отмена), **Redo** (Повтор), **Cut** (Вырезать), **Copy** (Копировать), **Paste** (Вставить) и **Clear** (Очистить).

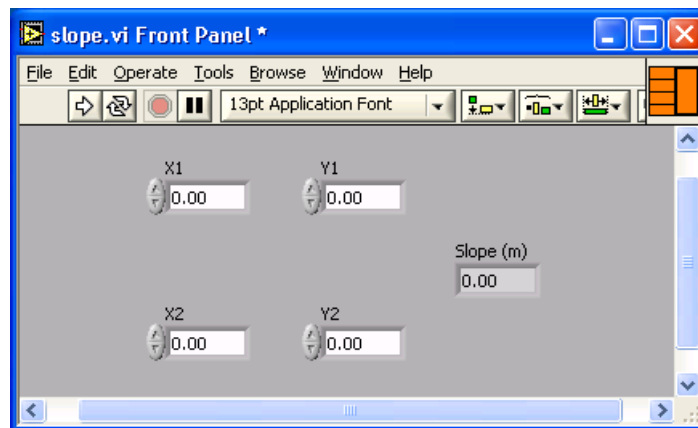
Настройка соединительной панели



Для использования ВП в качестве подпрограммы ВП необходимо настроить соединительную панель, показанную слева.

Соединительная панель является совокупностью полей ввода/вывода данных, соответствующих элементам управления и отображения этого ВП, подобно набору параметров вызова функции в текстовых языках программирования. Соединительная панель определяет поля входных и выходных данных ВП. Таким образом, ВП можно использовать в качестве подпрограммы.

Каждому полю ввода или вывода данных назначается свой элемент лицевой панели. Для редактирования соединительной панели необходимо щелкнуть правой кнопкой мыши на иконке ВП и выбрать из контекстного меню пункт **Show Connector** (Показать поля ввода/вывода данных). Вместо иконки появится соединительная панель, в которой каждый прямоугольник соответствует полю ввода или вывода данных. Количество отображаемых LabVIEW полей ввода/вывода данных соответствует количеству элементов на лицевой панели. Ниже показана лицевая панель, содержащая четыре элемента управления и один элемент отображения. Таким образом, в соединительной панели LabVIEW отображает четыре поля ввода и одно поле вывода данных.



Выбор и редактирование шаблона соединительной панели

Выбор шаблона осуществляется щелчком правой кнопки мыши на соединительной панели и выбором пункта **Patterns** (Шаблон) из контекстного меню. В шаблоне некоторые из полей ввода/вывода данных можно оставить без соединения и задействовать позднее при необходимости. Такая гибкость дает возможность вносить изменения с минимальным отражением на иерархии ВП. Причем не все элементы лицевой панели должны быть обязательно задействованы в соединительной панели.

Задействованные поля выделены цветом, соответствующим типу данных элемента. Максимально возможное количество полей ввода/вывода данных ограничено 28.



Наиболее часто используемый шаблон показан слева. Данный шаблон является стандартным для упрощения соединения. Верхние поля ввода/вывода обычно используются для ссылок, нижние — для обработки ошибок. Подробная информация об обработке ошибок находится в Уроке 6, *Кластеры*



Внимание. Следует избегать необходимости использования более 16 полей ввода/вывода данных. Наличие более 16 полей снижает удобочитаемость.

Предусмотрена возможность изменять пространственное положение полей ввода-вывода соединительной панели с помощью соответствующего пункта контекстного меню: **Flip Horizontal** (отражение по горизонтали), **Flip Vertical** (вертикали) или **Rotate 90 Degrees** (поворот на 90°).

Привязка полей ввода/вывода данных к элементам лицевой панели

После выбора шаблона соединительной панели необходимо каждому полю назначить свой элемент лицевой панели. Для упрощения использования подпрограммы ВП следует поля ввода данных размещать слева, а поля, связанные с элементами отображения, - справа на соединительной панели.

Чтобы назначить поля ввода или вывода данных, следует щелкнуть по выбранному полю левой кнопкой мыши, затем щелкнуть мышью на элементе, который необходимо связать с этим полем, после этого вывести курсор в свободное пространство лицевой панели и снова щелкнуть мышью.

Задействованные поля примут цвет, определенный типом данных соответствующего элемента.

Можно также сначала щелкнуть левой кнопкой мыши по элементу, а потом по полю ввода/вывода данных.



Внимание. Во время назначения полей ввода/вывода данных используется инструмент СОЕДИНЕНИЕ, однако между элементом лицевой панели и соответствующим ему полем проводник не появляется.

Упражнение 4-1. ВП Преобразования °C в °F

Цель: 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:\exercises\LV Basics I** и выберите файл *Преобразование C в F (начало).vi*

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



Совет. Нажатие стрелки рядом с кнопкой **Open VI** (Открыть ВП) в диалоговом окне **LabVIEW** позволит обратиться к недавно использовавшимся ВП, таким как *Преобразование C в F(начало).vi*

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).

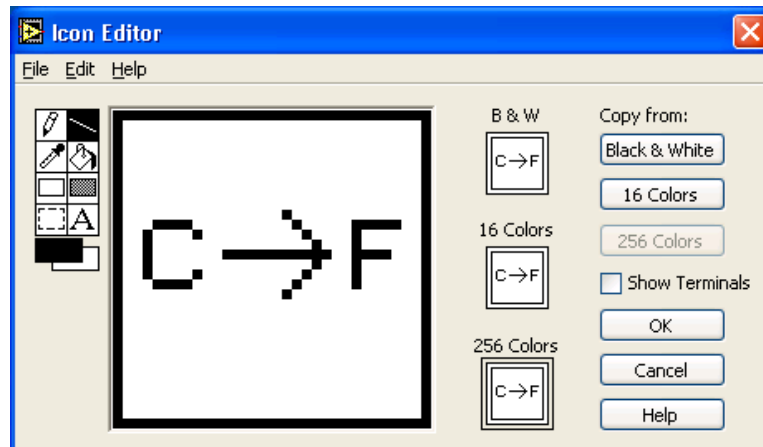
4. Нажав кнопку **<Delete>**, очистите область редактирования иконки.



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 «С» 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 **OK** 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 С в F.vi*, он будет использоваться позднее.
10. Выберите пункт главного меню **File»Close**. Закройте ВП.

Конец упражнения 4-1

С. Использование подпрограмм ВП

После создания ВП, оформления его иконки и настройки соединительной панели ВП может использоваться в качестве подпрограммы. Чтобы поместить подпрограмму ВП на блок-диаграмму, следует выбрать на палитре **Functions** (Функций) подраздел **Select a VI** (Выбор ВП). Указать ВП и перенести его на блок-диаграмму.

Открытый ВП можно поместить на блок-диаграмму другого ВП, переместив на нее иконку этого ВП с помощью инструмента ПЕРЕМЕЩЕНИЕ.

Редактирование подпрограммы ВП

Вызов лицевой панели подпрограммы ВП из блок-диаграммы другого ВП производится двойным щелчком на нем инструментом УПРАВЛЕНИЕ или ПЕРЕМЕЩЕНИЕ. Это же можно сделать с помощью главного меню, выбрав в пункте **Browse** (Обзор) подпункт **This VI's SubVIs** (Подпрограммы этого ВП). Для вызова блок-диаграммы подпрограммы ВП следует, удерживая клавишу **<Ctrl>**, дважды щелкнуть на нем левой кнопкой мыши.

Изменения, внесенные в подпрограмму ВП, доступны вызывающим его программам только после предварительного их сохранения.

Конец упражнения 4-2

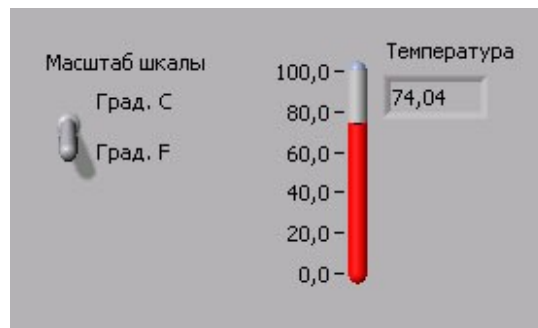
Упражнение 4-3. ВП Термометр гр

Цель: Создание ВП, иконки и настройка соединительной панели для обеспечения возможности использования ВП в качестве подпрограммы

Приведена последовательность действий для создания ВП, который измеряет температуру с помощью температурного датчика сигнальной панели DAQ и отображает значение температуры в градусах Цельсия или в градусах Фаренгейта.

Лицевая панель

1. Откройте ВП **Термометр**, созданный в последнем упражнении, в случае если он не открыт.
2. Создайте элемент отображения данных температуры, как показано ниже:



- a. Выберите элемент отображения данных, расположенный на палитре **Controls** в разделе **Numeric** (Числовые элементы)
 - b. Напечатайте **Температура** внутри собственной метки и нажмите кнопку **Enter** на инструментальной панели.
 - c. Щелкните правой кнопкой мыши по элементу и выберите пункт контекстного меню **Visible Items** (Отображаемые элементы), **Digital Display** (Цифровой индикатор).
3. Создайте элемент управления в виде вертикального переключателя.
 - a. Выберите вертикальный переключатель, расположенный в палитре **Controls** раздела **Boolean** (Логические элементы).

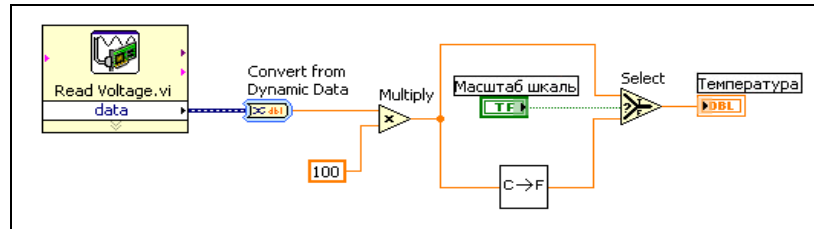


- b. Введите имя собственной метки переключателя **Масштаб шкалы** и нажмите кнопку **Enter** на инструментальной панели.
 - c. Используя инструмент ВВОД ТЕКСТА, создайте на лицевой панели свободную метку °C, как показано выше.
 - d. С помощью инструмента ВВОД ТЕКСТА, создайте на лицевой панели свободную метку °F, как показано выше.
4. Создайте описание ВП, которое появляется в окне контекстной справки **Context Help** после наведения курсора на иконку ВП.
- a. Выберите пункт главного меню **File»VI Properties**.
 - b. Выберите пункт **Documentation** (Описание) в разделе **Category** (Категория) из выпадающего меню.
 - c. В поле ввода текста напечатайте следующее:
Этот ВП измеряет температуру, используя температурный датчик сигнальной панели DAQ.
5. Создайте описание элементов управления и отображения данных, которое появляется в окне контекстной справки **Context Help** после наведения на них курсора.
- a. Щелкните правой кнопкой мыши по элементу отображения и выберите пункт контекстного меню **Description and Tip** (Описание и предупреждения).
 - b. В поле ввода текста напечатайте следующее:
Выводит на экран значения измеренной температуры.
 - c. Введите в поле **Tip** значение **Температура**.
 - d. Нажмите кнопку **ОК**.
 - e. Щелкните правой кнопкой мыши по элементу управления и выберите пункт контекстного меню **Description and Tip** (Описание и предупреждения).
 - f. В поле ввода текста напечатайте следующее:
Определяет шкалу (по Фаренгейту или Цельсию), используемую для измерения температуры.
 - g. Введите в поле **Tip** значение **шкала - °C или °F**.
 - h. Нажмите кнопку **ОК**.

6. Отобразите окно контекстной справки **Context Help**, которое доступно из пункта главного меню **Help»Show Context Help**.
7. Наведите курсор на один из объектов для просмотра описания их работы в окне **Context Help**.

Блок-диаграмма

8. Перейдите на блок-диаграмму, выбрав **Window»Show Diagram**.
9. Создайте блок-диаграмму, показанную ниже:



Внимание. Если аппаратные средства DAQ недоступны, для имитации считывания напряжения следует использовать **(Demo) Read Voltage VI**, расположенный в каталоге **c:\exercises\LV Basics I**.



Поместите на блок-диаграмму ВП **Convert from Dynamic Data**, расположенный в палитре **Functions»Express»Signal Manipulation**. Этот ВП преобразует динамический тип данных. В конфигурационном диалоговом окне выбрать пункт **Single Scalar** из списка **Resulting data type**.



Выберите функцию **Multiply** (Умножение), расположенную в палитре **Functions»Numeric**. Эта функция умножает считанное ВП «**Read Voltage VI**» напряжение на «100.0» для представления температуры в градусах Цельсия.



Щелкните правой кнопкой мыши по полю ввода данных у функции **Multiply** (Умножение) и в контекстном меню выберите пункт **Create»Constant**. Константе присвойте значение «100» и нажмите клавишу **<Enter>**.



В палитре **Functions** (Функций) в разделе **Select a VI** (Выбор ВП) в директории **c:\exercises\LV Basics I** выберите ВП **Преобразование C в F**, созданный в упражнении 3-1. Поместите его на блок-диаграмму. Этот ВП переведет градусы Цельсия в градусы Фаренгейта.



Выберите функцию **Select** (Выбор), расположенную в палитре **Functions»Comparison**. Эта функция выдает значения °C или °F в зависимости от состояния переключателя **Масштаб**

шкалы.

Используйте инструмент ПЕРЕМЕЩЕНИЕ для перемещения объектов в положение, показанное на предыдущей блок-диаграмме, и инструмент СОЕДИНЕНИЕ для их соединения.



Совет. Для идентификации полей ввода и вывода данных узлов следует щелкнуть правой кнопкой мыши на узле и в контекстном меню выбрать пункт **Visible Items»Terminal**.

Лицевая панель

10. Перейдите на лицевую панель.



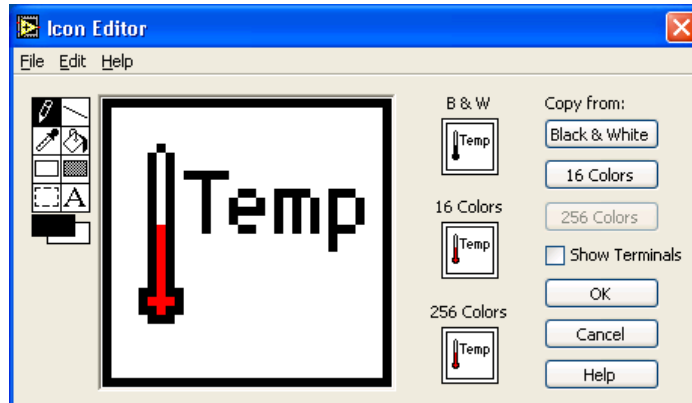
11. Нажмите на кнопку непрерывного запуска, показанную слева.

12. Приложите свой палец к датчику температуры и наблюдайте ее увеличение.

13. Нажмите на кнопку непрерывного запуска еще раз для остановки ВП.

Иконка и соединительная панель

14. Создайте иконку, показанную ниже, чтобы использовать ВП в качестве подпрограммы



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



b. Дважды щелкните по инструменту ВЫБОР (показан слева).

c. Нажмите клавишу **<Delete>**, очистите область редактирования иконки.



d. Дважды щелкните по инструменту ПРЯМОУГОЛЬНИК (показан слева), чтобы обвести область

редактирования границей выбранного цвета.



- e. Чтобы оформить иконку, как показано выше, воспользуйтесь инструментом **КАРАНДАШ**.
- f. Выберите цвет инструмента **ПЕРЕДНИЙ ПЛАН** с помощью инструмента **ЗАПОЛНЕНИЕ ЦВЕТОМ** и раскрасьте термометр в красный цвет.



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

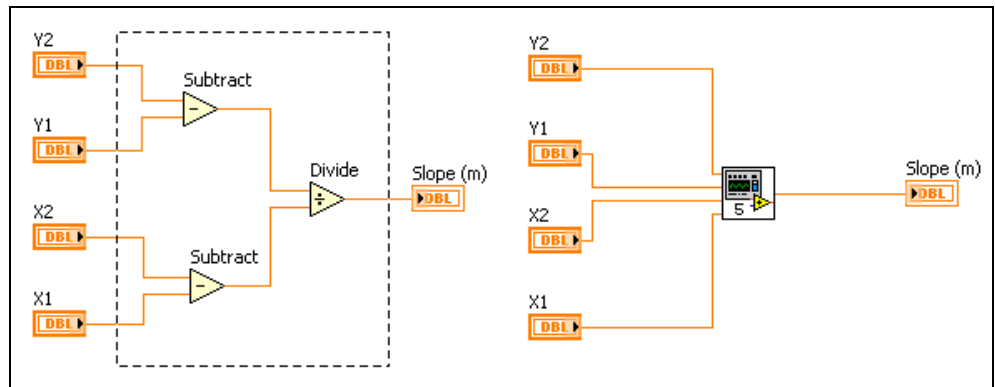


- g. Для выбора размера шрифта дважды щелкните по инструменту **ВВОД ТЕКСТА**.
 - h. Из пункта **Copy from** (Копировать) выберите **B & W** (черно-белую) иконку и **256 Colors** (256-цветный) режим печати для создания черно-белой иконки, которую LabVIEW использует в случае отсутствия цветного принтера.
 - i. После завершения редактирования иконки нажмите кнопку **ОК** и закройте **Icon Editor** (Редактор иконки). Новая иконка появится в правом верхнем углу обеих панелей.
15. Элементам управления и отображения данных поставьте в соответствие поля ввода и вывода данных, щелкнув правой кнопкой мыши по иконке и выбрав пункт контекстного меню **Show Connector** (Показать поля ввода/вывода данных).
- a. Щелкните по левому полю соединительной панели.
 - b. Щелкните по элементу **Масштаб шкалы**. Левое поле ввода данных станет зеленым.
 - c. Щелкните по правому полю соединительной панели.
 - d. Щелкните по элементу **Температура**. Правое поле ввода данных станет оранжевым.
 - e. Щелкните по свободному пространству панели.
16. Сохраните ВП под именем *Термометр.vi* в директории **c:\exercises\LV Basics I**, он будет использоваться позднее.
17. Закройте ВП, выбрав пункт главного меню **File»Close**.

Конец упражнения 4-3

Е. Превращение выделенной секции блок-диаграммы ВП в подпрограмму ВП

Можно упростить блок-диаграмму ВП, создав из часто выполняемых операций подпрограмму ВП. Для этого с помощью инструмента ПЕРЕМЕЩЕНИЕ необходимо выделить интересующую секцию блок-диаграммы и выбрать из пункта главного меню **Edit** (Редактирование) пункт **Create SubVI** (Создать подпрограмму ВП). Выделенная секция сменится иконкой новой подпрограммы ВП. LabVIEW создаст элементы управления и отображения данных для новой подпрограммы ВП и соединит поля ввода/вывода данных с существующими проводниками, как показано на примере ниже:



По умолчанию новая подпрограмма ВП использует шаблон для создания соединительной панели и иконку. Дважды щелкните правой кнопкой мыши по иконке подпрограммы ВП для редактирования соединительной панели и иконки и для сохранения ВП.



Внимание. Нельзя создать подпрограмму ВП из секции с количеством входов и выходов более 28, так как 28 – максимальное количество возможных полей ввода/вывода данных подпрограммы ВП.

Занятие 5.

Многократные повторения и Циклы

Структуры являются графическим представлением операторов цикла и операторов **Case** (Варианта), используемых в текстовых языках программирования. Структуры на блок-диаграмме используются для выполнения повторяющихся операций над потоком данных, операций в определенном порядке и наложения условий на выполнение операций. Среда LabVIEW содержит пять структур: Цикл **While** (по условию), Цикл **For** (с фиксированным числом итераций), структура **Case** (Вариант), структура **Sequence** (Последовательность), структура **Event** (Событие), а также **Formula Node** (узел Формулы).

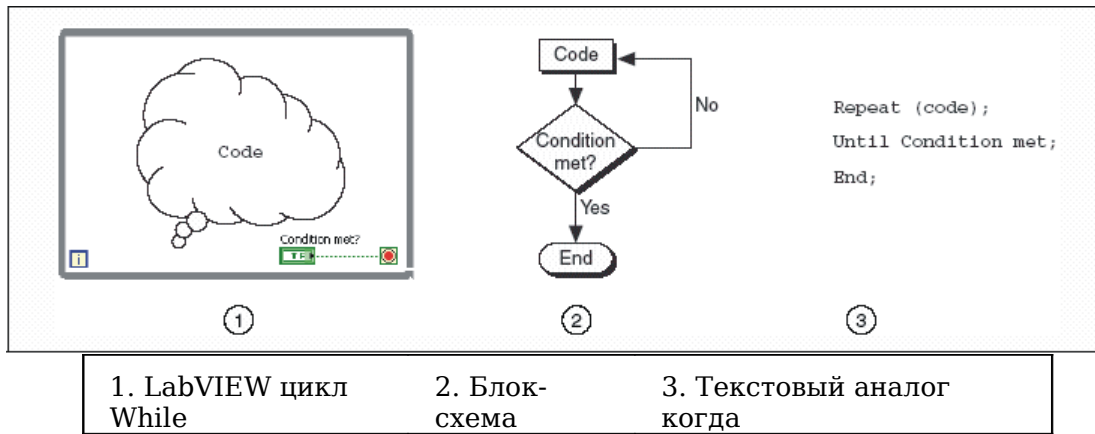
Рассмотрены структуры - Цикл **While** (по условию), Цикл **For** (с фиксированным числом итераций), а также функции, часто используемые с этими структурами, такие как **Shift Register** (сдвиговый регистр) и **Feedback Node** (узел обратной связи).

Изложены вопросы:

- A. Использование цикла **While** (по условию).
- B. Использование цикла **For** (с фиксированным числом итераций).
- C. Организация доступа к значениям предыдущих итераций цикла.

A. Цикл **While** (по Условию)

Цикл **While** (по условию) работает до тех пор, пока не выполнится логическое условие выхода из цикла. Цикл **While** аналогичен циклам **Do** и **Repeat Until**, используемым в текстовых языках программирования. Следующая иллюстрация демонстрирует (1) цикл **While** в среде LabVIEW, (2) эквивалентную блок-схему работы цикла **While**, (3) пример текстового аналога кода работы цикла **While**.



Цикл **While** находится в палитре **Functions»Structures**. После того как цикл выбран в палитре **Functions** (Функций), следует с помощью курсора выделить часть блок-диаграммы, которую необходимо поместить в цикл. После отпускания кнопки мыши, выделенная область блок-диаграммы помещается в тело цикла.

Добавление объектов блок-диаграммы в тело цикла осуществляется помещением или перетаскиванием объектов.



Совет. Цикл **While** выполняется всегда, по крайней мере, 1 раз.

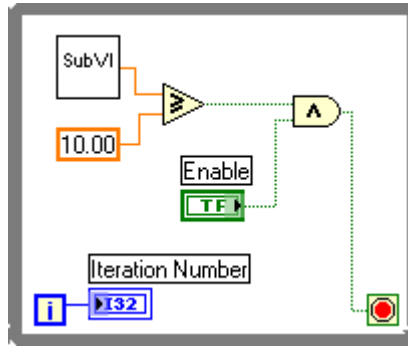


Блок-диаграмма цикла **While** выполняется до тех пор, пока не выполнится условие выхода из цикла. По умолчанию, терминал условия выхода имеет вид, показанный слева. Это значит, что цикл будет выполняться до поступления на терминал условия выхода значения **TRUE**. В этом случае терминал условия выхода называется терминалом **Stop If True** (Остановка если Истина).



Терминал счетчика итераций, показанный слева, содержит значение количества выполненных итераций. Начальное значение терминала всегда равно нулю.

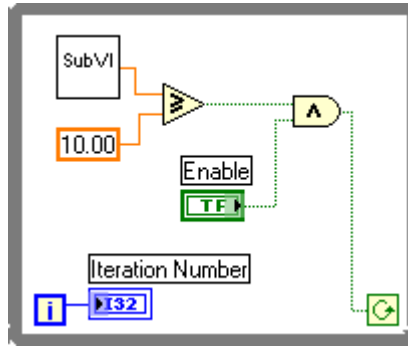
На блок-диаграмме, показанной на рисунке, условие выхода из цикла **While** определяется значением выходного параметра подпрограммы ВП большего или равного 10,00 и состоянием терминала элемента управления **Enable**. Функция **And** (Логическое «И») на выходе выдает значение TRUE, если оба на поля ввода данных функции поступают значения TRUE. В противном случае функция на выходе выдает значение FALSE и работа цикла завершается.



В предыдущем примере велика вероятность получения бесконечно выполняемого цикла. Обычно стремятся получить единственное условие выхода из цикла, нежели одновременное выполнение двух условий.

Предусмотрена возможность изменения условия выхода и соответствующего ему изображения терминала условия выхода. Щелчком правой кнопки мыши по терминалу условия выхода или по границе цикла необходимо вызвать контекстное меню и выбрать пункт **Continue If True** (Продолжение если Истина). Также можно воспользоваться инструментом УПРАВЛЕНИЕ, щелкнув им по терминалу условия выхода. Изображение терминала условия выхода поменяется на показанное слева **Continue If True** (Продолжение Если Истина). В результате условием выхода из цикла становится поступающее на терминал условия выхода значение FALSE, как показано на следующей блок-диаграмме.



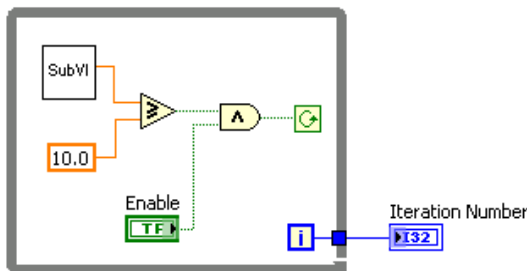


Цикл **While** выполняется до тех пор, пока выходные данные подпрограммы ВП остаются меньше «10».

Терминалы входных/выходных данных цикла

Данные могут поступать в цикл **While** (или выходить из него) через терминалы входных/выходных данных цикла. Терминалы входных/выходных данных цикла передают данные из структур и в структуры. Терминалы входных/выходных данных цикла отображаются в виде сплошных прямоугольников на границе области цикла **While**. Прямоугольник принимает цвет типа данных, передаваемых по терминалу. Данные выходят из цикла по его завершении. В случае если данные поступают в цикл **While** через терминал входных/выходных данных цикла, выполнение цикла начинается при поступлении данных в терминал.

На следующей блок-диаграмме терминал счетчика итераций присоединен к терминалу выхода цикла. Значения из терминала выхода цикла не поступают к элементу отображения номера итерации до завершения цикла **While**.



Лишь последнее значение итерации отображается элементом отображения номера итерации.

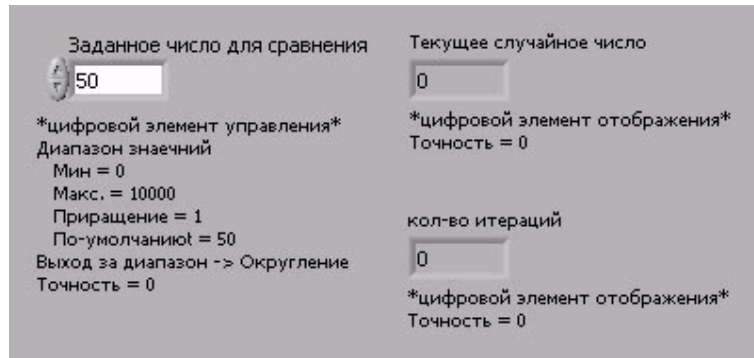
Упражнение 5-1. ВП Подсчет итераций

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.

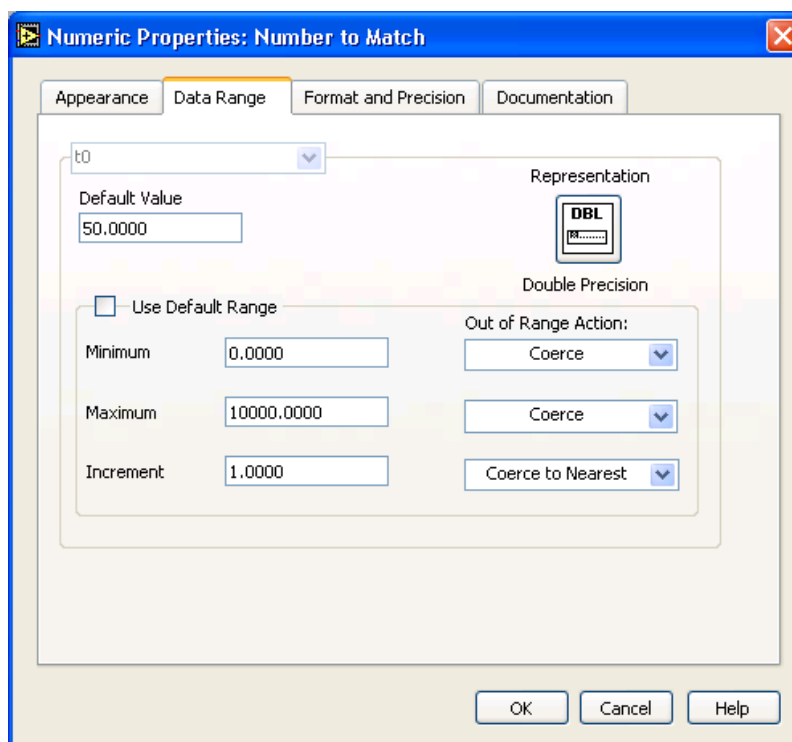


- a. Поместите на лицевую панель числовой элемент управления, находящийся на палитре **Controls»Numeric**. Назовите элемент **Заданное число для сравнения**. Этот элемент задает число, с которым будет проводиться сравнение.
- b. Поместите на лицевую панель числовой элемент отображения, находящийся на 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 100000 с шагом изменения 1 и значением по умолчанию равным 50.

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



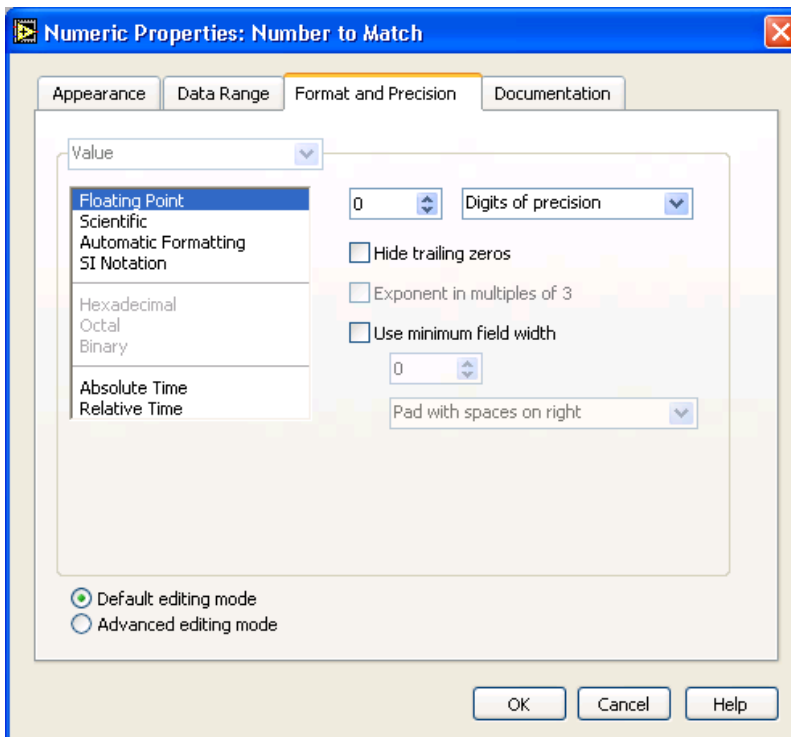
- Установите **Default Value** (значение по умолчанию) равным 50.
- Установите **Minimum Value** (минимальное значение) равным 0 и выберите **Coerce**.

- c. Установите **Maximum Value** (максимальное значение) равным 10000 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.



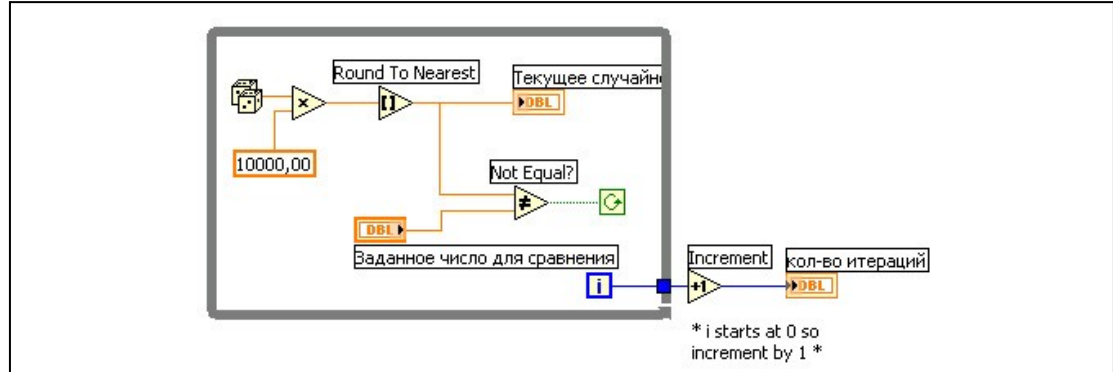
В поле ввода **Digits of Precision** следует ввести значение

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 случайных чисел) на 10000.



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 10000.



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*

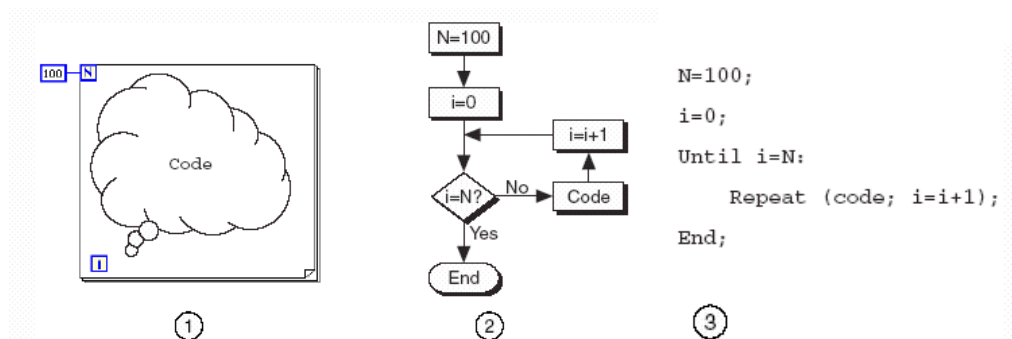
Запуск ВП

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 10000.
15. Запустите ВП. LabVIEW автоматически приведет новое значение к ближайшему значению в указанном диапазоне входных данных элемента.
16. Закройте ВП.

Конец упражнения 5-1

В. Цикл For (с фиксированным числом итераций)

Цикл **For** (с фиксированным числом итераций) выполняет повторяющиеся операции над потоком данных определенное количество раз (по заданию). Следующая иллюстрация демонстрирует (1) цикл **For** в среде LabVIEW, (2) эквивалентную блок-схему работы цикла **For**, (3) пример текстового аналога кода работы цикла **For**.



1. LabVIEW
2. Блок-схема
3. Текстовый аналог кода

N

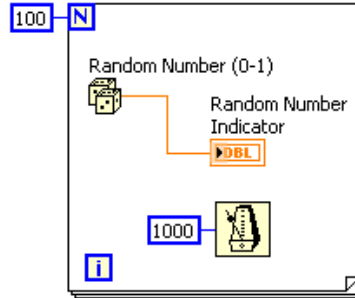
Цикл **For**, расположен в палитре **Функций** в разделе **Functions»Structures**. Значение, присвоенное **терминалу максимального числа итераций N** цикла, показанного слева, определяет максимальное количество повторений операций над потоком данных.

i

Терминал счетчика итераций, показанный слева, содержит значение количества выполненных итераций. Начальное значение счетчика итераций всегда равно 0.

Цикл **For** отличается от цикла **While** тем, что завершает работу, выполнив заданное максимальное число итераций **N**. Цикл **While** завершает работу при выполнении заданного условия выхода из цикла.

Цикл **For**, показанный на рисунке ниже, генерирует случайное число каждую секунду 60 раз и отображает их в элементе отображения данных.

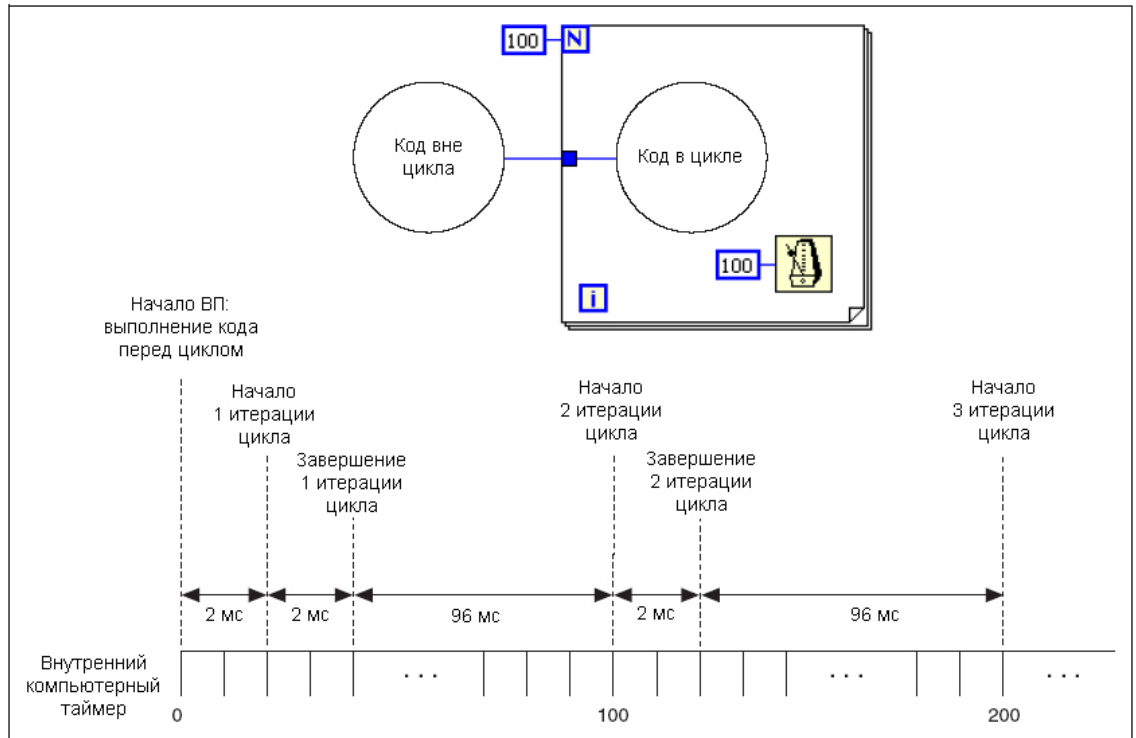


Функции ожидания

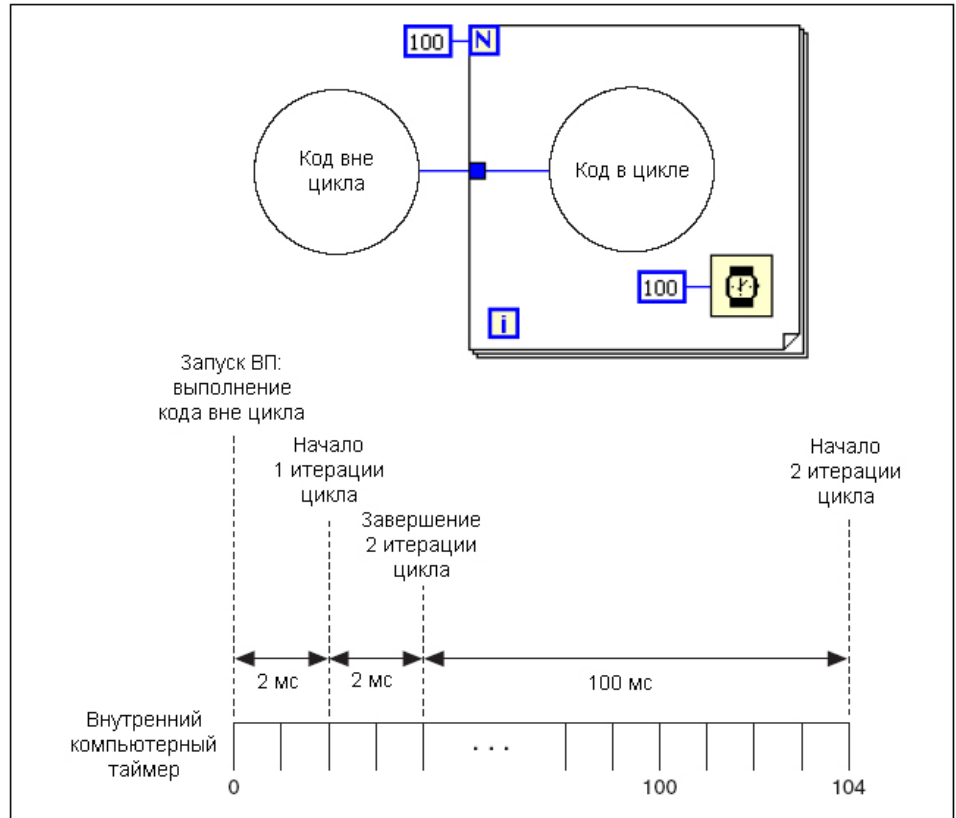


Функция **Wait Until Next ms Multiple**, показанная слева, обеспечивает интервал между итерациями, равный интервалу времени, необходимому для того, чтобы миллисекундный счетчик достиг значения, кратного введенному пользователем. Эта функция используется для синхронизации действий. Функцию **Wait Until Next ms Multiple** вызывают внутри цикла для контроля скорости выполнения цикла.

Функция **Wait Until Next ms Multiple** обеспечивает интервал между итерациями, равный интервалу времени, необходимому внутреннему таймеру компьютера для достижения указанного кратного значения. Существует вероятность, что первый период цикла будет коротким, как показано ниже.



Функция **Wait(ms)**, показанная слева, добавляет время ожидания ко времени выполнения программы, как показано ниже. Это может вызвать затруднения, если время выполнения программы является переменным.



Примечание. ВП **Express Time Delay**, находящийся в палитре **Functions»Express»Execution Control**, подобен функции **Wait(ms)**. Однако он обладает встроенным кластером ошибок. Подробнее о кластерах ошибок можно прочитать в Уроке 6 "Кластеры".

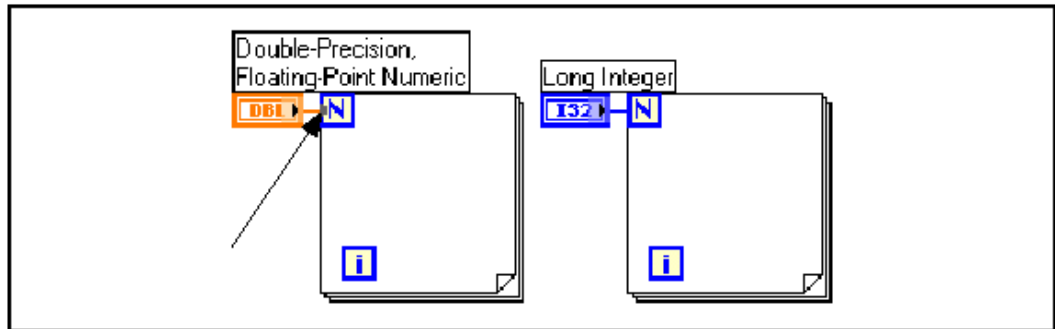
Функции ожидания находятся в палитре **Functions»Time and Dialog**.

Преобразование типов данных

LabVIEW может оперировать с такими типами данных, как целочисленный тип (integer): **byte, word, long**, число с плавающей запятой: **single, double, extended precision**, комплексное число: **single, double, extended precision**. Когда в поле ввода данных функции поступают операнды разных типов, то значение на выходе функции принимает формат данных более широкого диапазона. При этом LabVIEW автоматически осуществляет преобразование типов и в месте соединения проводника с терминалом появляется изображение серой точки.

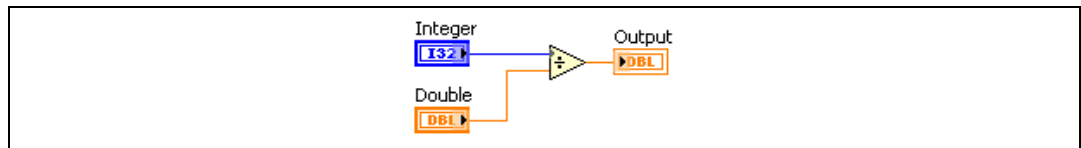
Например, терминал **максимального числа итераций N** цикла **For** имеет **целочисленный тип двойной точности (long integer)**. На него поступают данные в формате числа

двойной точности с плавающей запятой. На терминале числа итераций появляется серая точка, как показано ниже на рисунке.



Если в поля ввода данных функции, работающей с данными одного типа, поступают данные двух разных типов, LabVIEW приводит тип данных одного из терминалов к типу данных другого терминала. LabVIEW выбирает тип данных, занимающий большее количество бит. Если типы эквивалентны по количеству занимаемых бит, LabVIEW предпочитает беззнаковый тип данных.

В следующем примере данные типов I32 (signed 32 bit integer / 32-битный целочисленный со знаком) и DBL (double 64 bit floating-point / 64-битное с плавающей запятой двойной точности) поступают в поля ввода функции деления. LabVIEW на выходе функции формирует данные в представлении DBL, поскольку тип данных I32 занимает меньшее количество бит.



Для изменения типа представления данных на объектах блок-диаграммы необходимо щелкнуть по ним правой кнопкой мыши и из контекстного меню выбрать пункт **Representation**.

Когда LabVIEW проводит преобразование данных из формата числа двойной точности с плавающей запятой в целочисленный формат, то значение $x,5$ округляется до ближайшего целого четного. Например, LabVIEW округляет 2,5 до 2, а 3,5 до 4.

Подробнее о типах данных можно прочитать в тексте занятия 2, "Создание ВП" или в *LabVIEW Help*.

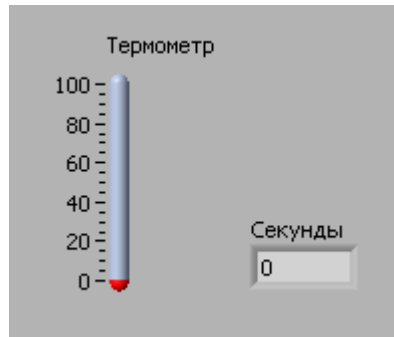
Упражнение 5-2. ВП Измерение температуры во времени

Цель: измерение температуры раз в секунду в течение одной минуты.

Ниже приведена последовательность действий для создания ВП, который использует ВП Термометр для измерения температуры раз в секунду в течение одной минуты.

Лицевая панель

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

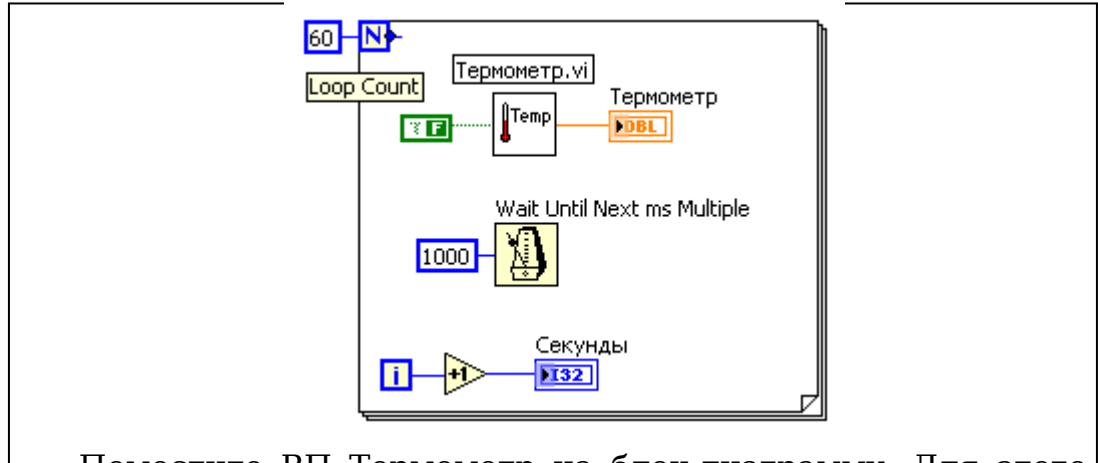


Поместите Термометр, расположенный на палитре **Controls»Numeric**, на блок-диаграмму для отображения на экране измерений температуры.

Поместите на лицевую панель цифровой элемент отображения данных, расположенный в палитре **Controls»Numeric**. Назовите его **Секунды**.

Блок-диаграмма

2. Создайте блок-диаграмму, показанную ниже:



Поместите ВП Термометр на блок-диаграмму. Для этого выберите **Functions»Select a VI** и укажите папку **c:\exercises\LV Basics I**. Этот ВП, снимает показания с устройства DAQ. Если устройства DAQ с датчиком температуры, присоединенным к каналу 0 DAQ платы, недоступны, следует использовать *(Демо) Термометр.vi*

Щелкните правой кнопкой мыши по полю ввода данных **Temp Scale** и в контекстном меню выберите пункт **Create»Constant**. Константе присвойте значение FALSE — для градусов Фаренгейта и TRUE — для градусов Цельсия.



Поместите на блок-диаграмму функцию **Wait Until Next ms Multiple**, находящуюся в палитре **Functions»Time and Dialog**. Щелкните правой кнопкой мыши по полю ввода данных и выберите пункт **Create»Constant**. Созданной константе присвойте значение 1000. Теперь каждая итерация цикла выполняется с интервалом времени 1000 мс (раз в секунду).



Поместите на блок-диаграмму функцию **Increment** (приращение), находящуюся в палитре **Functions»Numeric**. Эта функция добавляет 1 к значению счетчика итераций после завершения выполнения цикла.

3. Сохраните ВП под именем *Измерение температуры во времени.vi* в папке **c:\exercises\LV\Basics I**.
4. Запустите ВП. Снятие первого показания может занять больше одной секунды в случае, если компьютер будет конфигурировать устройство DAQ.
5. Закройте ВП.

Дополнительно

6. Создайте ВП, генерирующий случайные числа в цикле **While**. Организуйте выход из цикла по нажатию кнопки на лицевой панели ВП.
7. Сохраните ВП под именем *Цикл While.vi* в папке **c:\exercises\LV Basics I**.
8. Перейдите к выполнению упражнений повышенной сложности или закройте ВП.

Упражнения повышенной сложности

9. Переделайте ВП Цикл **While** таким образом, чтобы выход из цикла осуществлялся по нажатию кнопки на лицевой панели или по достижению циклом **While** числа итераций, заданного элементом управления на лицевой панели.
10. Сохраните ВП под именем *Цикл While-For.vi* в папке **c:\exercises\LV Basics I**.
11. Закройте ВП.

Конец упражнения 5-2

С. Организация доступа к значениям предыдущих итераций цикла

При работе с циклами зачастую необходим доступ к значениям предыдущих итераций цикла. Например, в случае ВП, измеряющего температуру и отображающего ее на графике, для отображения текущего среднего значения температуры, необходимо использовать значения, полученные в предыдущих итерациях. Есть два пути доступа к этим данным: **Shift Register** (сдвиговый регистр) и **Feedback Node** (узел обратной связи).

Сдвиговые регистры

Сдвиговые регистры используются при работе с циклами для передачи значений от текущей итерации цикла к следующей. Сдвиговые регистры аналогичны статическим переменным в текстовых языках программирования



Сдвиговый регистр выглядит как пара терминалов, показанных слева. Они расположены непосредственно друг против друга на противоположных вертикальных сторонах границы цикла. Правый терминал содержит стрелку «вверх» и сохраняет данные по завершению текущей итерации. LabVIEW передает данные с этого регистра в следующую итерацию цикла. Сдвиговый регистр создается щелчком правой кнопки мыши по границе цикла и выбором из контекстного меню пункта **Add Shift Register**.

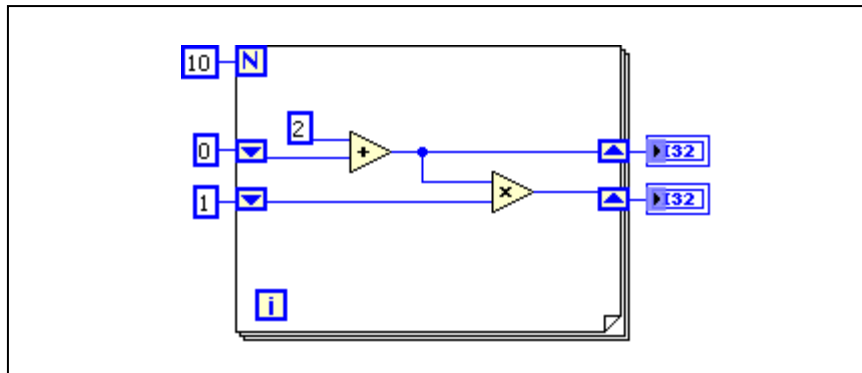
Сдвиговый регистр передает данные любого типа, он автоматически принимает тип первых поступивших на него данных. Данные, передаваемые на терминалы сдвигового регистра, должны быть одного типа.

Чтобы инициализировать сдвиговый регистр, необходимо передать на его левый терминал любое значение извне цикла. Если не инициализировать сдвиговый регистр, он использует значение, записанное в регистр во время последнего выполнения цикла или значение, используемое по умолчанию для данного типа данных, если цикл никогда не выполнялся.

Цикл с неинициализированным сдвиговым регистром используется при неоднократном запуске ВП для присвоения выходному значению сдвигового регистра значения, взятого с последнего выполнения ВП. Чтобы сохранить информацию

о состоянии между последующими запусками ВП, следует оставить вход левого терминала сдвигового регистра не определенным. После завершения выполнения цикла последнее значение, записанное в регистр, останется на правом терминале. При последующей передаче данных из цикла через правый терминал будет передано последнее значение, записанное в регистр.

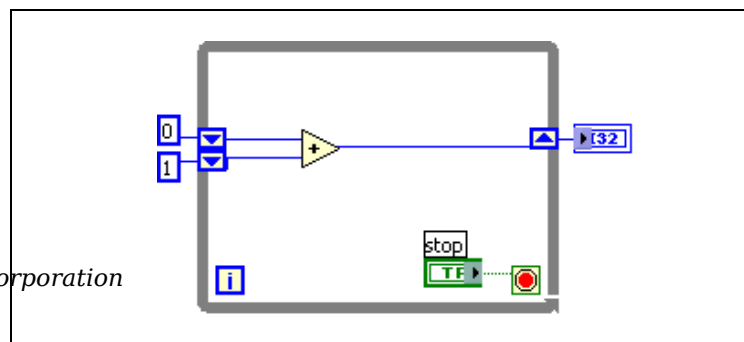
Предусмотрена возможность создания нескольких сдвиговых регистров в одной структуре цикла. Если в одном цикле выполняется несколько операций, следует использовать сдвиговый регистр с несколькими терминалами для хранения данных, полученных в результате выполнения различных операций цикла. На следующем рисунке показано использование двух инициализированных сдвиговых регистров.



Стек сдвиговых регистров.

Для создания стека сдвиговых регистров достаточно щелкнуть правой кнопкой мыши по левому терминалу и выбрать пункт контекстного меню **Add Element**. Стек сдвиговых регистров осуществляет доступ к значениям предыдущих итераций цикла. Стек сдвиговых регистров сохраняет данные предыдущей итерации и передает эти значения к следующей итерации.

Стек сдвиговых регистров может находиться только в левой части цикла, так как правый терминал лишь передает данные из текущей итерации в следующую.



При добавлении еще двух сдвиговых регистров к левому терминалу данные последних трех итераций переносятся на следующую итерацию, при этом значение последней итерации сохраняется в самом верхнем сдвиговом регистре. Второй терминал сохраняет данные, переданные ему с предыдущей итерации, нижний терминал хранит данные, полученные две итерации назад.

Узлы обратной связи



Узел обратной связи, показанный слева, автоматически появляется в циклах **While** или **For** при соединении поля вывода данных подпрограммы ВП, функции или группы подпрограмм ВП и функций с полем ввода данных тех же самых подпрограмм ВП, функций или их групп. Как и сдвиговый регистр, узел обратной связи сохраняет данные любого типа по завершению текущей итерации и передает эти значения в следующую итерацию. Использование узлов обратной связи позволяет избежать большого количества проводников данных и соединений.

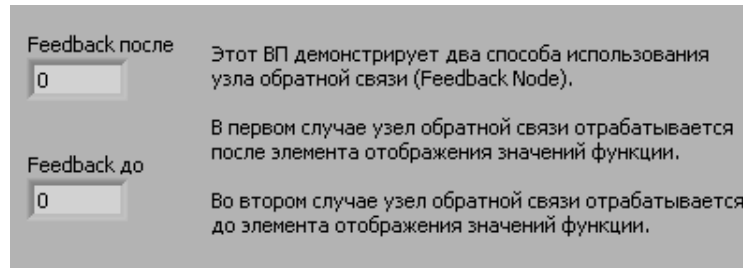
Можно поместить узел обратной связи внутри цикла **While** или **For**, выбрав **Feedback Node** в палитре **Structures**. При помещении узла обратной связи на проводник данных до ответвления, передающего данные на выходной терминал цикла, узел обратной связи передает все значения на выходной терминал цикла. При помещении узла обратной связи на проводник после ответвления, передающего данные на выходной терминал цикла, узел обратной связи передаст все значения обратно на поле ввода данных ВП или функции, а затем передаст последнее значение на выходной терминал цикла. Следующее упражнение содержит пример работы узла обратной связи.

Упражнение 5-3. Доступ к данным предыдущих итераций

Цель: Использование сдвиговых регистров и узлов обратной связи для организации доступа к значениям на предыдущих итерациях цикла For.

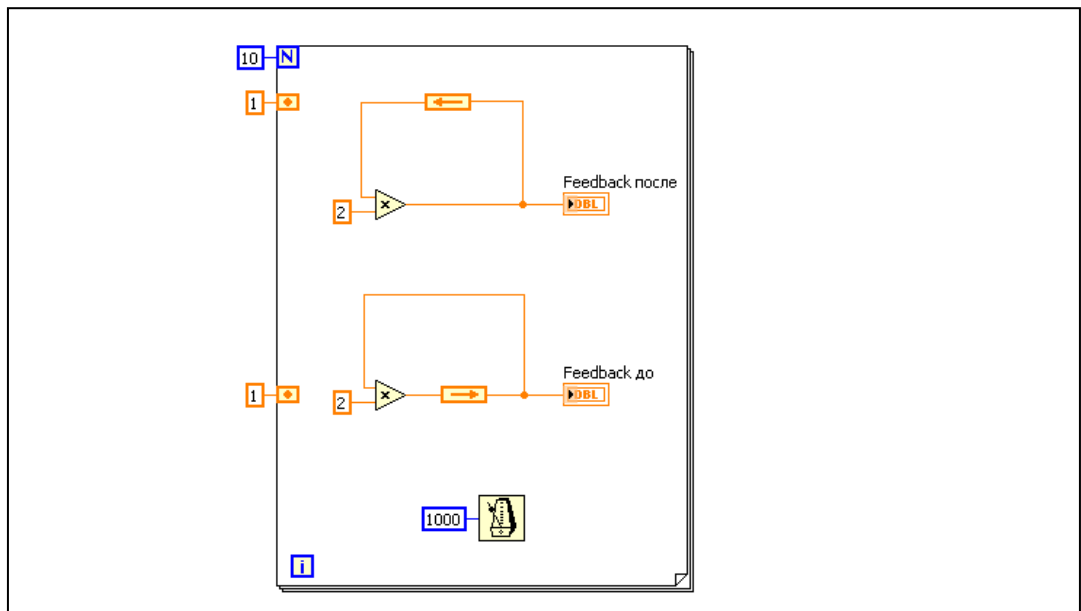
Лицевая панель

1. Откройте ВП Узел обратной связи VI, находящийся в каталоге **c:\exercises\LV Basics I**. 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** замедляет 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

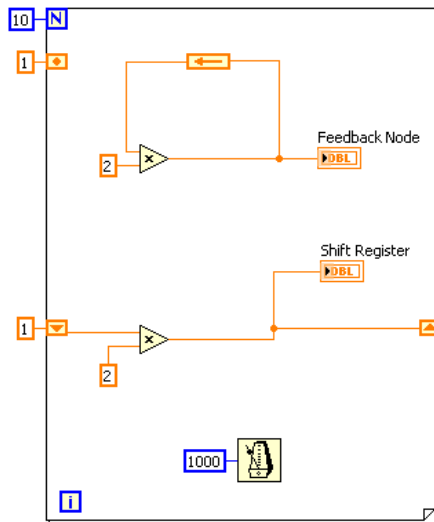
3. Запустите ВП. Программа в верхней части сначала считывает значение узла обратной связи, инициализированного значением 1. Затем это значение передается функции **Multiply** (умножение).

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:

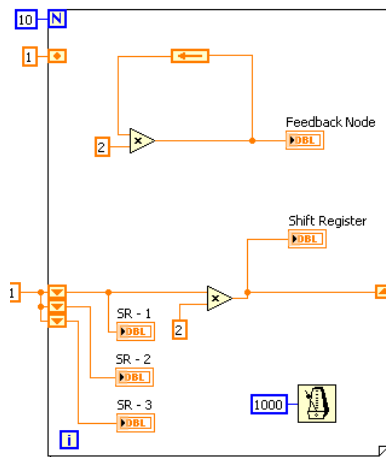




- a. Выделите нижний узел обратной связи и нажмите клавишу <Delete>, чтобы удалить его.
 - b. Щелкните правой кнопкой мыши по границе цикла и выберите пункт контекстного меню **Add Shift Register**.
 - c. Инициализируйте сдвиговый регистр значением 1.
 - d. Переименуйте верхний и нижний элементы отображения 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 5-3

Занятие 6

Массивы

Рассказывается об объединении элементов одного типа данных в массивы.

Изложены вопросы:

- A. Что такое массив.
- B. Создание массивов с помощью цикла.
- C. Использование функций работы с массивами.
- D. Полиморфизм.

А. Массивы

Массивы объединяют элементы одного типа данных. Массив – это набор элементов определенной размерности. Элементами массива называют группу составляющих его объектов. Размерность массива – это совокупность столбцов (длина) и строк (высота), а также глубина массива. Массив может иметь одну и более размерностей, и до $2^{31}-1$ элементов в каждом направлении, насколько позволяет оперативная память.

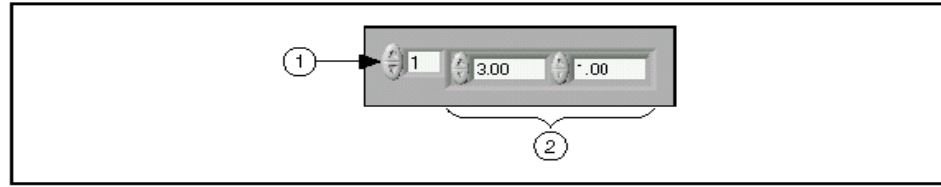
Данные, составляющие массив, могут быть любого типа: целочисленного, логического или строкового. Массив также может содержать элементы графического представления данных и кластеры. Использовать массивы удобно при работе с группами данных одного типа и при накоплении данных после повторяющихся вычислений. Массивы идеально подходят для хранения данных, полученных с графиков, или накопленных во время работы циклов, причем одна итерация цикла создает один элемент массива.

Нельзя создать массив, состоящий из массивов. Однако можно создать массив кластеров, где каждый кластер будет состоять из одного или более массивов. Более подробную информацию по этому вопросу можно найти в Уроке 6, *Кластеры*.

Все элементы массива упорядочены. Чтобы к ним было легко обращаться, каждому элементу присвоен индекс. Нумерация элементов массива всегда начинается с 0. Таким образом, индексы массива находятся в диапазоне от 0 до (n-1), где n – число элементов в массиве. Например, в массиве из девяти планет солнечной системы n=9, следовательно, значение индекса находится в пределах от 0 до 8. Земля является третьей планетой от Солнца, поэтому ее индекс равен 2.

Создание массива элементов управления и отображения

Для создания массива элементов управления или отображения данных, как показано в примере, необходимо выбрать шаблон массива из палитры **Controls>>Array & Cluster** и поместить его на лицевую панель. Затем поместить в шаблон массива элемент управления либо отображения данных. Поместить в шаблон массива запрещенный элемент управления или отображения, например, двухкоординатный график осциллограмм (**XY graph**) не удастся.



1. Элемент индекса массива

2. Элементы значений массива

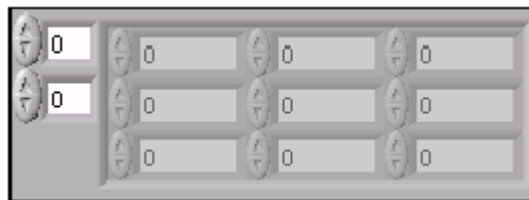
Поместить объект в шаблон массива следует до того, как он будет использоваться на блок-диаграмме. Если этого не сделать, то шаблон массива не будет инициализирован, и использовать массив будет нельзя.

Двумерные массивы

В двумерном (2D) массиве элементы хранятся в виде матрицы. Таким образом, для размещения элемента требуется указание индекса столбца и строки. На иллюстрации показан двумерный массив, состоящий из 6 столбцов (длина) и 4 строк (высота). Количество элементов в массиве = 24 ($6 \times 4 = 24$).



Для увеличения размерности массива необходимо щелкнуть правой кнопкой мыши по элементу индекса и выбрать из контекстного меню пункт **Add Dimension**. С этой целью также можно использовать инструмент ПЕРЕМЕЩЕНИЕ. Для этого надо просто изменить размер элемента индекса. Ниже приведен пример неинициализированного двумерного массива элементов управления.

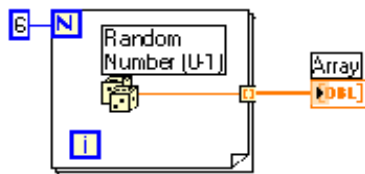


Создание массива констант

Создать массив констант на блок-диаграмме можно, выбрав в палитре **Functions>>Array** шаблон **Array Constant** и поместив в него числовую константу. Массив констант удобно использовать для передачи данных в подпрограммы ВП.

В. Автоматическая индексация

Цикл **For** и цикл **While** могут автоматически накапливать массивы и проводить их индексацию на своих границах. Это свойство называется автоиндексацией. После соединения терминала данных массива с терминалом выхода из цикла каждая итерация цикла создает новый элемент массива. На иллюстрации видно, что проводник данных, соединяющий терминал данных массива с терминалом выхода из цикла стал толще, а сам терминал выхода из цикла окрашен в цвет



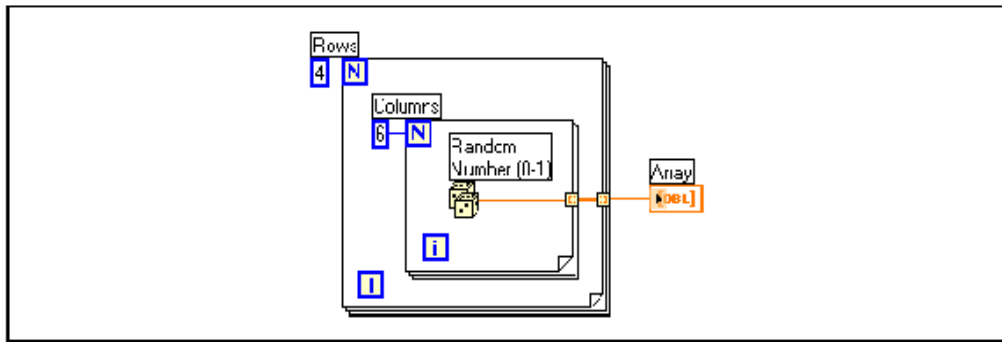
терминала данных массива.

Автоиндексация отключается щелчком правой кнопки мыши по терминалу входа/выхода из цикла и выбором пункта контекстного меню **Disable Indexing**. Автоиндексацию следует отключать, например, в случае, когда нужно знать только последнее значение.

Ввиду того, что цикл **For** часто используется при работе с циклами, для него в LabVIEW автоиндексация включена по умолчанию. Для цикла **While** автоиндексация по умолчанию отключена. Для того, чтобы включить автоиндексацию, необходимо щелкнуть правой кнопкой мыши по терминалу входа/выхода из цикла и выбрать в контекстном меню пункт **Enable Indexing**.

Создание двумерных (2D) массивов

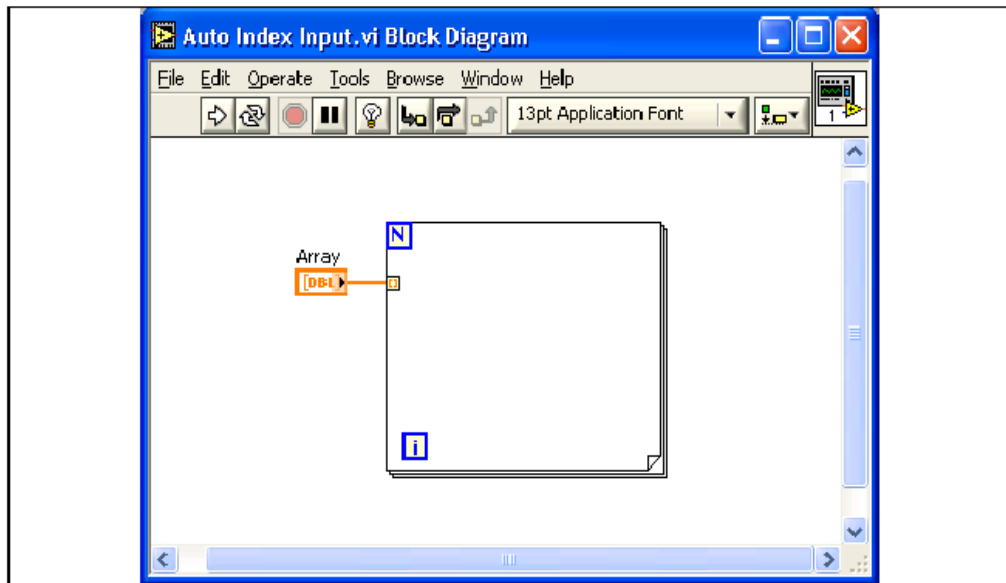
Для создания двумерных массивов необходимо использовать два цикла **For**, один внутри другого. Как показано на иллюстрации, внешний цикл создает элементы массива в строке, а внутренний цикл создает элементы массива в столбце.



Использование автоиндексации для установки значения терминала количества итераций цикла

При включенной автоиндексации массива, подключенного к терминалу входа в цикл **For**, LabVIEW автоматически устанавливает значение терминала количества итераций цикла **N** равным размерности массива. Таким образом, отпадает необходимость задания значения терминалу **N**.

В следующем примере цикл **For** будет выполнен ровно столько раз, сколько элементов в массиве. Как правило, стрелка на кнопке **Run** сломана, если терминал количества итераций цикла не подключен, однако в этом примере стрелка цела, что говорит о возможности запуска ВП.



Если автоиндексация установлена более чем для одного терминала входа в цикл или явно задано значение терминала количества итераций цикла **N**, то значением терминала **N**

станет меньшей из величин. Например, если соединить массив из 10 элементов с терминалом входа в цикл, а значение терминала количества итераций установить равным 15, то цикл выполнит 10 итераций.

С. Функции работы с массивами

Для создания и управления массивами используются функции, расположенные в палитре **Functions>>Array**. Наиболее часто используемые функции работы с массивами включают в себя:



Array Size - показывает количество элементов массива каждой размерности. Если массив *n*-мерный, на выходе функции **Array Size** будет массив из *n* элементов. Например, для приведенного ниже массива функция **Array Size** выдаст значение 3.

7	4	2
---	---	---



Initialize Array - создает *n*-мерный массив, в котором каждый элемент инициализирован значением поля ввода данных **element**. Для увеличения размерности массива достаточно добавить поля ввода данных, растянув узел функции. Например, если для функции **Initialize Array** заданы следующие значения параметров: на поле **element** подается значение 4, а на поле **dimension size** (если оно одно) - значение 3, то на выходе получится массив, показанный ниже.

4	4	4
---	---	---

Build Array - объединяет несколько массивов или добавляет элемент в n-мерный массив. Изменение размера функции увеличивает количество полей ввода данных, что позволяет увеличить количество добавляемых элементов. Например, если объединить два предыдущих массива, то функция **Build Array** выдаст на выходе следующий массив.

7	4	2
4	4	4

Для объединения входных данных в более длинный массив той же размерности, как показано ниже, достаточно щелкнуть правой кнопкой мыши на функции и выбрать из контекстного меню пункт **Concatenate Inputs**.

7	4	2	4	4	4
---	---	---	---	---	---



Array Subset - выдает часть массива, начиная с индекса, поступившего на поле **index**, и длиной, указанной в поле **length**. Например, если подать предыдущий массив на поле ввода функции **Array Subset**, значение 2 - на поле **index** и 3 - на поле **length**, то на поле вывода данных будет следующее подмножество

2	4	4
---	---	---



Index Array - выдает элемент, соответствующий индексу, значение которого подается на поле ввода **index**. Например, при использовании предыдущего массива, функция **Index Array** выдаст значение 2, если на поле ввода данных **index** подать значение 0.

Функцию **Index Array** можно использовать для выделения строки или столбца из двумерного массива и дальнейшего отображения в виде подмассива. Для этого двумерный массив надо подать в поле ввода данных функции. Функция **Index Array** должна иметь два поля **index**. Верхнее поле **index** указывает строку, а нижнее - столбец. Можно задействовать оба поля **index** для выбора отдельного элемента или только одно, для выбора строки или столбца. Например, в поле ввода данных функции подается массив, показанный ниже.

7	4	2
4	4	4

Функция **Index Array** в поле вывода данных выдаст следующий массив в случае, если на поле **index** (строка) подается значение 0.

7	4	2
---	---	---

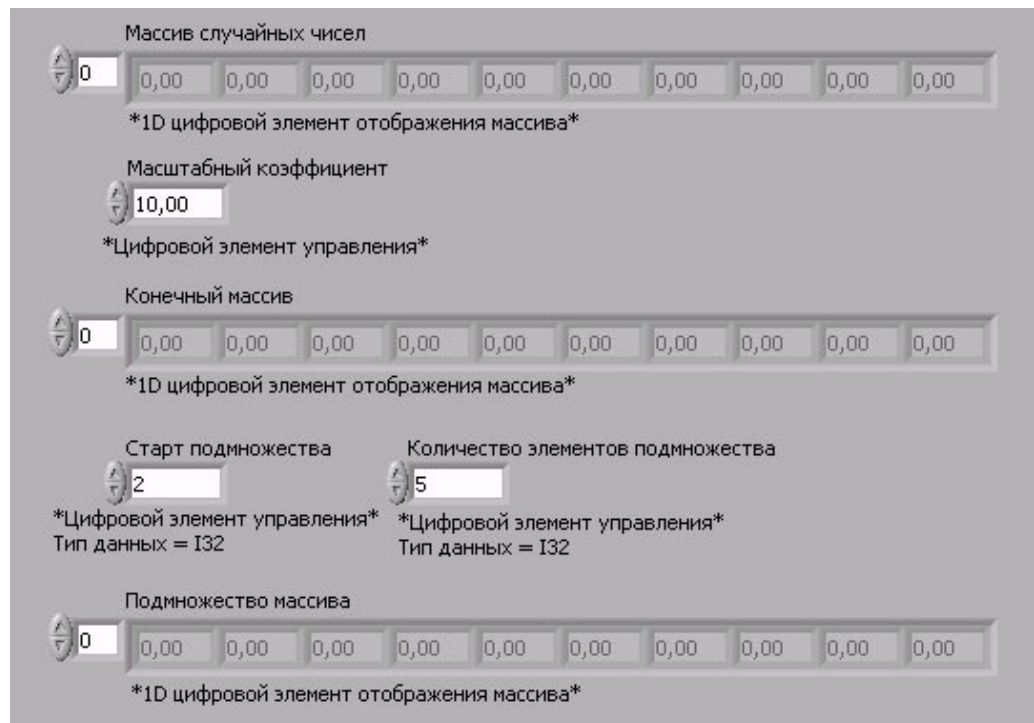
Упражнение 6-1 ВП Работа с массивами

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.

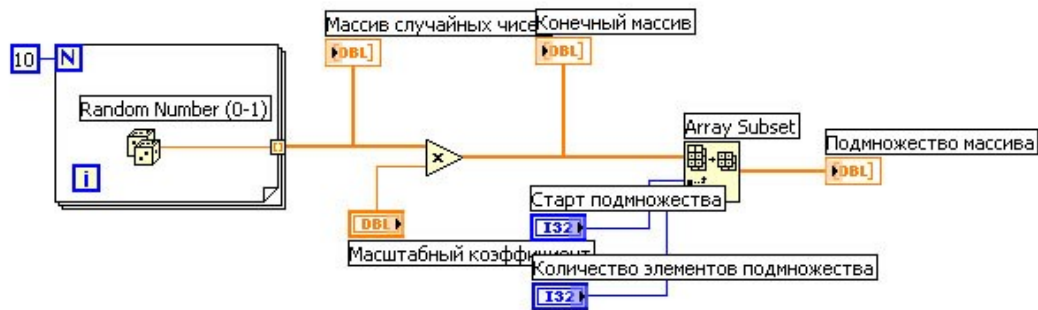


- a. В палитре **Controls>>Array & Cluster** выберите шаблон массива.
- b. Созданному массиву присвойте имя **Массив случайных чисел**.
- c. Поместите внутрь шаблона массива цифровой элемент отображения, расположенный в палитре **Controls>>Numeric**.
- d. С помощью инструмента ПЕРЕМЕЩЕНИЕ измените размер 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 *Работа с массивами.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 6-1

D. Полиморфизм

Арифметические функции, расположенные в палитре **Functions>>Numeric**, являются полиморфными. Это означает, что на поля ввода этих функций могут поступать данные различных типов (скалярные величины, массивы). Например, можно использовать функцию **Add** для прибавления скалярной величины к массиву или сложения двух массивов. Если на одно поле ввода данных функции **Add** подать скалярную величину 2, а другое соединить с массивом, показанным ниже,

1	3	2
---	---	---

то функция прибавит 2 к каждому элементу массива, и массив будет иметь вид:

3	5	4
---	---	---

Если на вход функции **Add** подать два предыдущих массива, функция сложит каждый элемент первого массива с соответствующим элементом второго и выдаст результат в виде массива, показанного ниже.

4	8	6
---	---	---

Если с помощью функции **Add** сложить два массива разной размерности, например, таких, как предыдущий и показанный ниже,

3	1	2	3
---	---	---	---

то функция сложит каждый элемент первого массива с соответствующим элементом второго и выдаст результат в виде массива размерностью меньшей из двух исходных.

7	9	8
---	---	---

С кластерами арифметические функции работают таким же образом. Подробнее о работе с кластерами можно узнать на Уроке 6, *Кластеры*.

Краткое изложение пройденного материала, советы и секреты

Массивы объединяют группу элементов одного типа данных. Данные, составляющие массив, могут быть любого типа: целочисленные, логические или строковые. Массив также может содержать сигнальные данные и кластеры.

Индекс первого элемента массива всегда равен 0. Таким образом, индексы массива находятся в диапазоне от 0 до (n-1), где n – число элементов в массиве.

Поместить объект в шаблон массива следует до того, как использовать этот массив на блок-диаграмме. Иначе на блок-диаграмме шаблон массива не будет инициализирован.

Для создания массива элементов управления или отображения необходимо выбрать его шаблон из палитры **Controls>>Array & Cluster** и поместить его на лицевую панель. Затем поместить в шаблон массива элемент управления либо отображения.

Если массив соединить с терминалом входа цикла **For** или цикла **While**, то при включенной автоиндексации можно читать и обрабатывать каждый элемент массива.

Для создания и управления массивами используются функции, расположенные на палитре **Functions>>Array**.

LabVIEW осуществляет автоиндексацию автоматически. Автоиндексация для цикла **For** включена по умолчанию. Для цикла **While** автоиндексация по умолчанию отключена.

Полиморфизм означает, что поля ввода данных функций могут различаться по структуре данных (скалярные величины, массивы).

Дополнительные упражнения

- 6-2. Создайте ВП, который полностью изменяет порядок элементов в массиве, содержащем 100 случайных чисел. Например, элемент массива с индексом 0 становится элементом массива с индексом 99, а элемент массива с индексом 1 становится элементом массива с индексом 98, и так далее.



Совет. Для изменения порядка данных в массиве следует использовать функцию **Reverse 1D Array**, расположенную на палитре **Functions>>Array**.

Сохраните ВП под именем *Перевернутый массив случайных чисел.vi*

- 6-3. Создайте ВП, который накапливает массив значений температур с помощью *Термометр.vi* (упражнение 3-3). Размер массива определяйте с помощью элемента управления на лицевой панели. С помощью функции **Initialize Array** создайте массив такого же размера, в котором все элементы имеют значение 10. Сложите два массива, найдите размер конечного массива и вычислите его среднее значение. На лицевую панель выведите массив значений температуры, инициализированный массив, конечный массив и среднее значение.

Сохраните ВП под именем *Найти среднее значение.vi*

- 6-4. Создайте ВП, который генерирует двумерный массив случайных чисел, содержащий 3 строки и 10 столбцов. После этого ВП индексирует все строки и выводит осциллограмму каждой из них на отдельном графике. На лицевой панели должно быть три графика.

Сохраните ВП под именем *Выделение 2D массива.vi*



6-5. Создайте ВП, который имитирует бросок игральной кости с гранями 1 - 6 и записывает в массив выпавшие значения после каждого броска. На входе - количество бросков кости, а на выходе - количество выпадений каждой грани. Использовать только один сдвиговый регистр (**shift register**).

Сохраните ВП под именем *Игра в кости.vi*



6-6. Создайте ВП, который генерирует одномерный массив и затем попарно перемножает элементы, начиная с элементов с индексами 0 и 1 и т.д., а затем выводит результаты в массив элементов отображения данных. Например, входной массив имеет значение {1, 23, 10, 5, 7, 11}, а в результате получается массив {23, 50, 77}.



Совет. Используйте функцию **Decimate ID Array**, расположенную в палитре **Functions>>Array**.

Сохраните ВП под именем *Попарное перемножение элементов массива.vi*

Занятие 7

Кластеры

Рассказывается об объединении элементов различных типов данных в кластеры.

Изложены вопросы:

- A. Что такое кластеры.
- B. Использование функций работы с кластерами.
- C. Кластеры ошибок.

А. Кластеры

Кластеры объединяют элементы разных типов данных, подобно пучку проводов телефонного кабеля, где каждый провод представляет собой отдельный элемент кластера. Кластеры играют ту же роль, что и структуры в языках программирования.

Объединение нескольких групп данных в кластер устраняет беспорядок на блок-диаграмме и уменьшает количество полей ввода/вывода данных, необходимых подпрограмме ВП. Максимально возможное количество полей ввода/вывода данных ВП равно 28. Если лицевая панель содержит более 28 элементов, которые необходимо использовать в ВП, можно некоторые из них объединить в кластер и связать кластер с полем ввода/вывода данных. Как и массив, кластер может быть элементом управления или отображения данных, однако кластер не может одновременно содержать элементы управления и отображения данных.

В кластере, как и в массиве, все элементы упорядочены, но обратиться по индексу к ним нельзя, необходимо сначала разделить их. Для этого предназначена функция **Unbundle By Name**, которая обеспечивает доступ к определенным элементам кластера по их имени.

Создание кластеров из элементов управления и отображения данных

Для создания кластеров из элементов управления и отображения данных следует выбрать шаблон кластера на палитре **Controls>>Array & Cluster** и поместить его на лицевую панель. После этого шаблон кластера следует заполнить элементами. Изменить размер кластера можно с помощью курсора.

Ниже показан кластер, содержащий три элемента



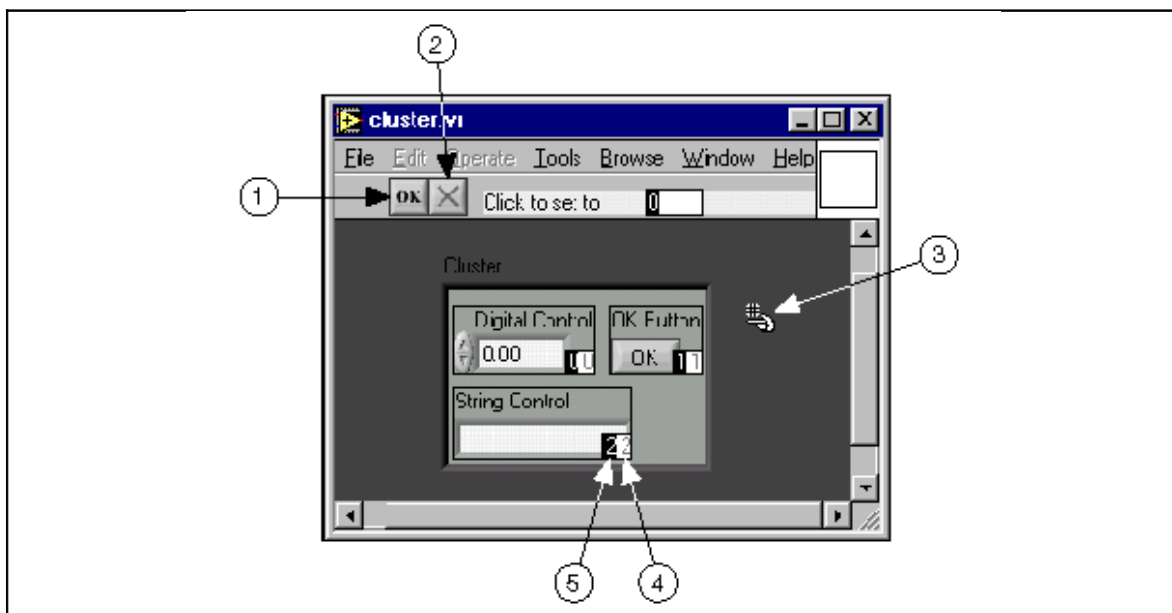
управления.

Порядок элементов в кластере

Каждый элемент кластера имеет свой логический порядковый номер, не связанный с положением элемента в шаблоне. Первому помещенному в кластер элементу автоматически присваивается номер 0, второму элементу – 1 и так далее. При удалении элемента порядковые номера автоматически изменяются.

Порядок элементов в кластере определяет то, как элементы кластера будут распределены по терминалам функций **Bundle** (объединения) и **Unbundle** (разделения) на блок-диаграмме.

Посмотреть и изменить порядковый номер объекта, помещенного в кластер, можно, щелкнув правой кнопкой мыши по краю кластера и выбрав из контекстного меню пункт **Reorder Controls In Cluster**. Панель инструментов и кластер примут вид, показанный ниже на рисунке.



1. Кнопка подтверждения (Confirm button)
2. Кнопка отмены (Cancel button)
3. Курсор определения порядка (Cluster order cursor)
4. Текущий порядковый номер (Current order)
5. Новый порядковый номер (New order)

В белом поле (4) указан текущий порядковый номер элемента, в черном (5) – новый порядковый номер. Для установки порядкового номера элемента нужно в поле ввода текста **Click to set to** ввести число и нажать на элемент. Порядковый номер элемента изменится. При этом корректируются порядковые номера других элементов. Сохранить изменения можно, нажав кнопку **Confirm** на панели инструментов. Вернуть первоначальные установки можно, нажав кнопку **Cancel**.

Соответствующие элементы, определенные в кластерах одинаковыми порядковыми номерами, должны иметь совместимые типы данных. Например, в одном кластере элемент 0 является числовым элементом управления, а элемент 1 – строковым элементом управления. Во втором кластере элемент 0 – числовой элемент отображения данных и элемент 1 – строковый элемент отображения данных. Кластер элементов управления корректно соединится с кластером элементов отображения данных.

Однако если изменить порядковые номера элементов в одном из кластеров, проводник данных между кластерами будет разорван, так как типы данных элементов кластеров не будут соответствовать друг другу.

Создание кластера констант

На блок-диаграмме можно создать кластер констант, выбрав в палитре **Functions>>Cluster** шаблон **Cluster Constant** и поместив в него числовую константу или другой объект данных, логический или строковый.

Если на лицевой панели кластер уже существует, то кластер констант на блок-диаграмме, содержащий те же элементы, можно создать, просто перетащив кластер с лицевой панели на блок-диаграмму или, щелкнув правой кнопкой мыши на кластере, выбрать из контекстного меню пункт **Create>>Constant**.

В. Функции работы с кластерами

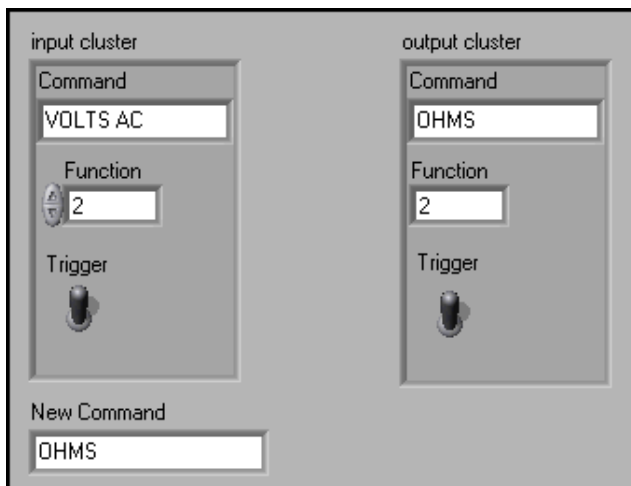
Для создания и управления кластерами используются функции, расположенные на палитре **Functions>>Cluster**. Функции **Bundle** и **Bundle by Name** используются для сборки и управления кластерами. Функции **Unbundle** и **Unbundle by Name** используются для разборки кластеров.

Эти функции также можно вызвать, щелкнув правой кнопкой мыши по терминалу данных кластера и выбрав из контекстного меню подменю **Cluster Tools**. Функции **Bundle** и **Unbundle** автоматически содержат правильное количество полей ввода/вывода данных. Функции **Bundle by Name** и **Unbundle by Name** в полях ввода/вывода данных содержат имя первого элемента кластера.

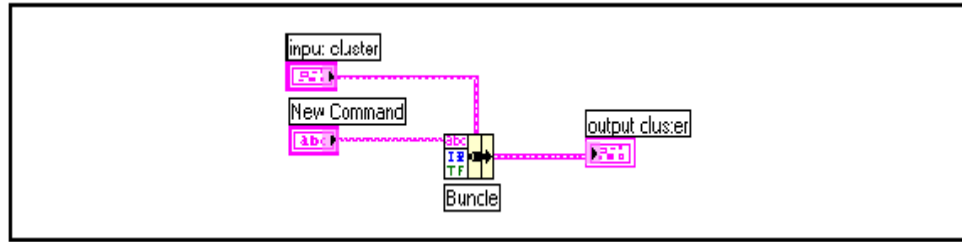
Сборка кластеров

Для сборки отдельных элементов в кластер используется функция **Bundle**. Эта же функция используется для изменения данных в элементе уже существующего кластера. Инструмент ПЕРЕМЕЩЕНИЕ используется для добавления полей ввода данных, для этого также можно щелкнуть правой кнопкой по полю ввода данных и выбрать из контекстного меню пункт **Add Input**. При соединении кластера с полем ввода данных **cluster** количество полей ввода данных функции должно соответствовать количеству элементов во входящем кластере.

На поле ввода данных **cluster** можно подать только одну требующую замены компоненту. Например, ниже показан кластер, имеющий три элемента управления.



Если известен логический порядок элементов, можно использовать функцию **Bundle** для изменения значения элемента **Command**, соединив элементы, как показано ниже.

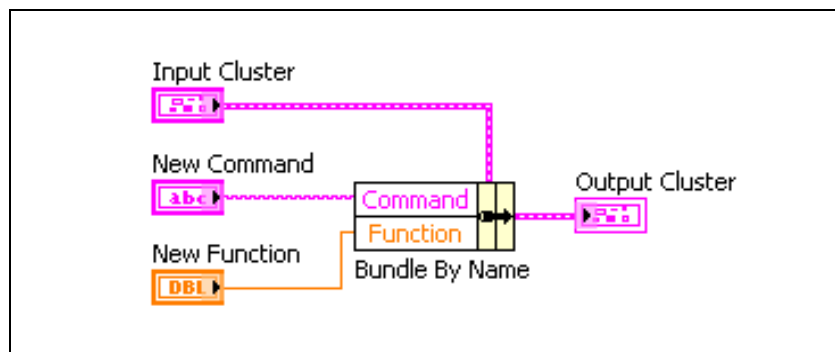


Замена или доступ к элементам кластера

Для замены элемента в уже существующем кластере используется функция **Bundle by Name**. Функция **Bundle by Name** работает так же как функция **Bundle**, но вместо обращения к элементу кластера по его порядковому номеру обращается к нему по его собственной метке (имени). При этом можно получить доступ только к элементам, имеющим собственную метку. Количество полей ввода данных не требует соответствия с количеством элементов в кластере.

С помощью элемента УПРАВЛЕНИЕ можно щелкнуть по полю ввода данных терминала и выбрать желаемый элемент из выпадающего меню. Можно также щелкнуть правой кнопкой мыши по полю ввода данных и выбрать элемент в разделе контекстного меню **Select Item**.

Ниже показано, как можно использовать функцию **Bundle by Name** для изменения значений элементов **Command** и



Function.

Использовать функцию **Bundle by Name** следует при работе со структурами данных, которые могут меняться в процессе работы. Чтобы добавить новый элемент в кластер или изменить порядковый номер элемента, нет необходимости

вновь подключать функцию **Bundle by Name**, так как имя элемента все еще действительно.

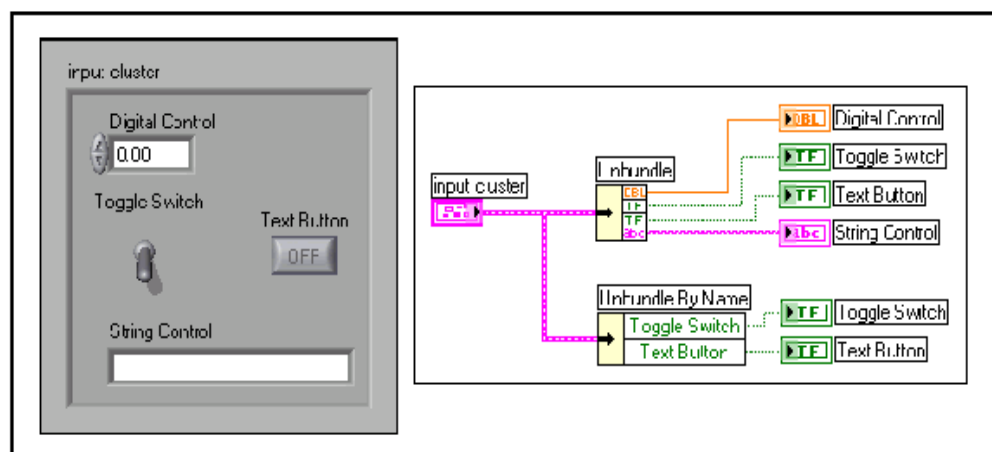
Разделение кластера

Функция **Unbundle** используется для разбиения кластеров на отдельные элементы.

Функция **Unbundle by Name** используется для выделения из кластера элементов по определенному имени. Количество полей вывода данных не зависит от количества элементов в кластере.

С помощью инструмента УПРАВЛЕНИЕ можно щелкнуть по полю вывода данных и выбрать желаемый элемент из контекстного меню. Можно также щелкнуть правой кнопкой мыши по полю вывода данных и выбрать из контекстного меню пункт **Select Item**.

Например, функция **Unbundle**, при использовании кластера, показанного ниже, имеет четыре поля вывода данных, которые соотносятся с четырьмя элементами кластера. Необходимо знать порядок элементов в кластере для корректного сопоставления логического элемента соответствующему вертикальному переключателю в кластере. В этом примере элементы упорядочены сверху вниз, начиная с 0. Если использовать функцию **Unbundle by Name**, то полей вывода данных может быть произвольное количество, и обращаться к отдельным элементам можно в произвольном порядке.

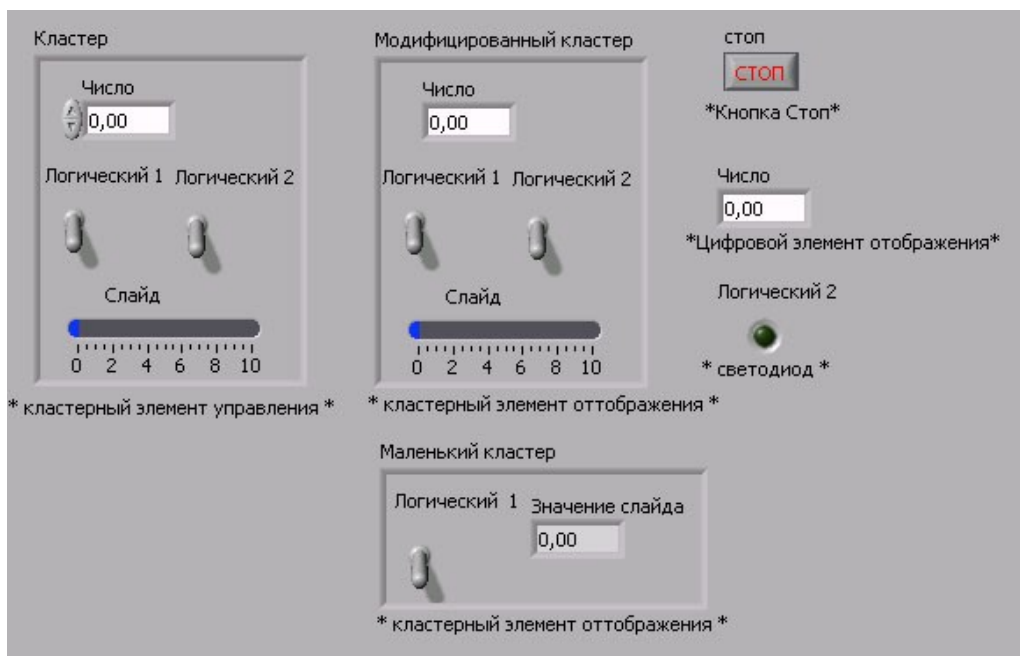


Упражнение 7-1 ВП Работа с кластерами

Цель: Создание кластеров на лицевой панели. Используя функции обработки кластеров, собирать и демонтировать кластеры

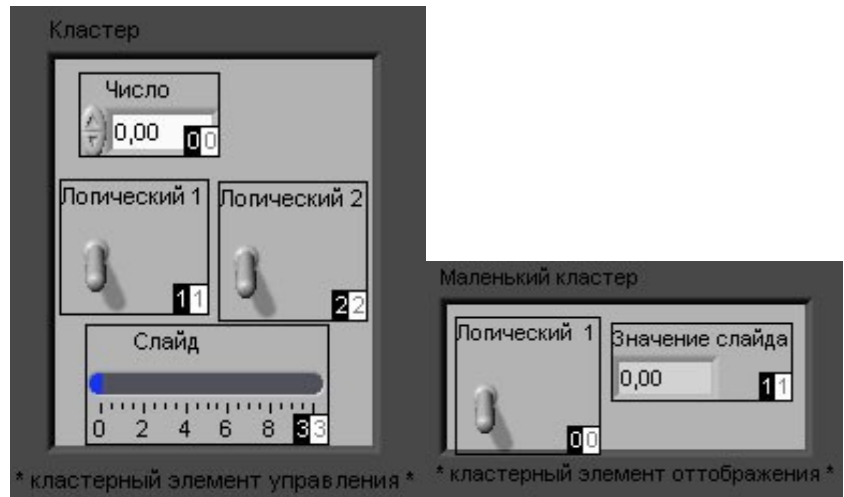
Лицевая панель

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



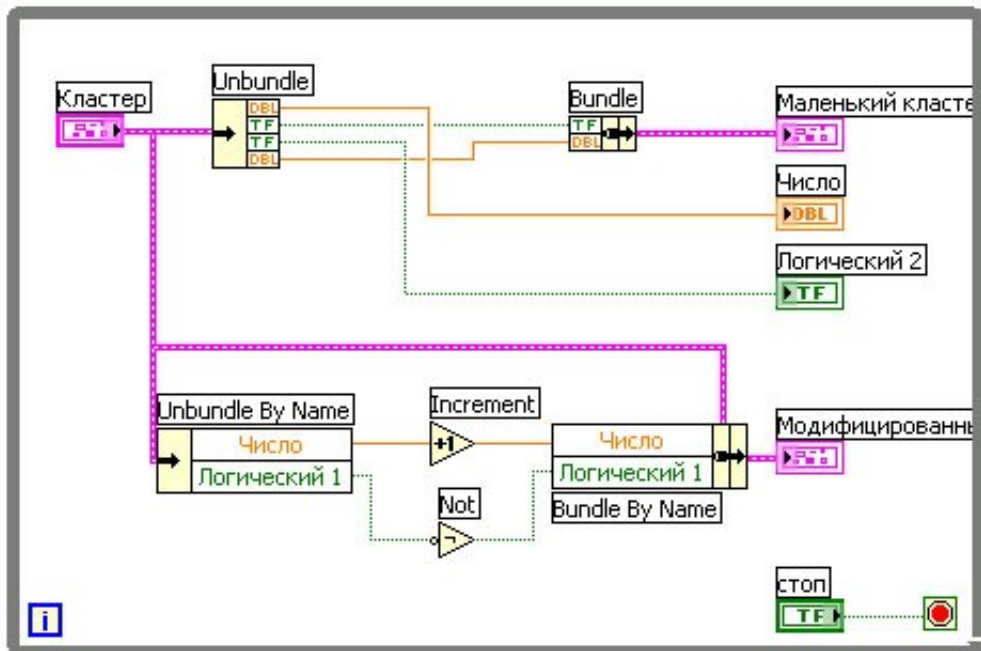
- a. Поместите на лицевую панель кнопку «Стоп» и круглый светодиод.
- b. Из палитры **Controls>>Array & Cluster** выберите шаблон кластера.
- c. Объекты лицевой панели, показанные на иллюстрации, поместите в шаблон кластера.
- d. Создайте и переименуйте копию элемента **Кластер** в **Модифицированный Кластер**. После этого щелкните правой кнопкой мыши по границе шаблона кластера **Модифицированный Кластер** и выберите из контекстного меню пункт **Change to Indicator**.
- e. Повторите пункт d для создания элемента **Маленький кластер**. Измените его, как показано на рисунке.

2. Проверьте порядковые номера элементов в кластерах **Кластер** и **Маленький кластер**. Порядковые номера элементов кластера **Модифицированный кластер** и **Кластер** должны совпадать.
 - a. Щелкните правой кнопкой мыши по границе шаблона каждого кластера, из контекстного меню выберите пункт **Reorder Controls in Cluster**.
 - b. Порядковые номера элементов установите, как показано ниже на рисунке.



Блок-диаграмма

3. Создайте блок-диаграмму, как показано ниже.





Из палитры **Functions>>Cluster** выберите функцию **Unbundle**. Эта функция разъединяет кластер **Кластер**. Измените размер этой функции до четырех полей ввода данных или соедините терминал данных кластера с функцией для автоматического добавления полей ввода данных.



Из палитры **Functions>>Cluster** выберите функцию **Bundle**. Эта функция объединит элементы в кластер **Маленький кластер**.



Из палитры **Functions>>Cluster** выберите функцию **Unbundle by Name**. Эта функция выделит два элемента из кластера **Кластер**. Измените размер функции до двух полей вывода данных. Если имена в полях вывода данных отличаются от показанных на иллюстрации, следует щелкнуть правой кнопкой мыши по имени элемента и в контекстном меню войти в раздел **Select Item**.



Из палитры **Functions>>Numeric** выберите функцию **Increment**. Эта функция добавит 1 к значению элемента **Число**.



Из палитры **Functions>>Boolean** выберите функцию **Not**. Эта функция выдаст логическое отрицание элемента **Логический 1**.



Из палитры **Functions>>Cluster** выберите функцию **Bundle by Name**. Эта функция изменит значения элементов **Число** и **Логический** в кластере **Кластер** и создаст кластер **Модифицированный кластер**. Измените размер этой функции на два поля ввода данных. Если имена в полях вывода данных отличаются от показанных на иллюстрации, следует щелкнуть правой кнопкой мыши по имени элемента и в контекстном меню войти в раздел **Select Item**.

4. Сохраните ВП под именем *Работа с кластерами.vi*

Запуск ВП

5. Перейдите на лицевую панель и запустите ВП.
6. Поменяйте значения элементов в кластере **Кластер** и запустите ВП.
7. Закройте ВП.

Конец упражнения 7-1

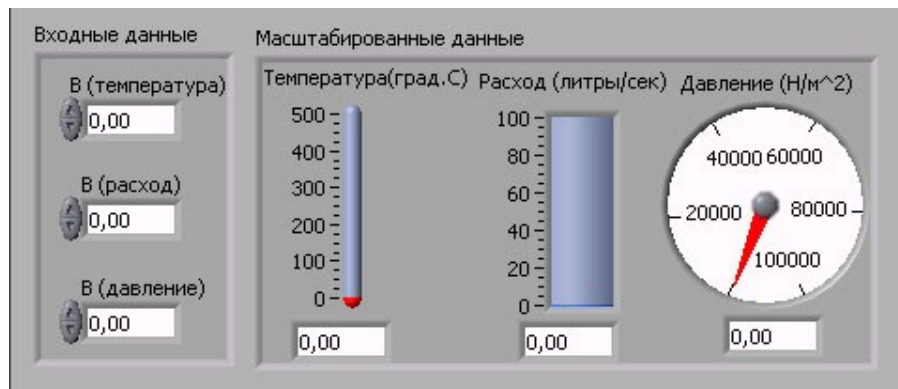
Упражнение 7-2 ВП Масштабирование кластера (дополнительное)

Цель: Создать ВП, использующий полиморфизм в кластерах

Выполните следующие шаги для создания ВП, в котором масштабируются данные, хранящиеся в кластере. Каждый элемент в кластере имеет свой масштабный коэффициент. Предположим, что исходные данные, значения давления, скорости потока и температуры были получены с соответствующих датчиков напряжения. Затем ВП масштабирует эти значения и выдает фактические значения физической величины.

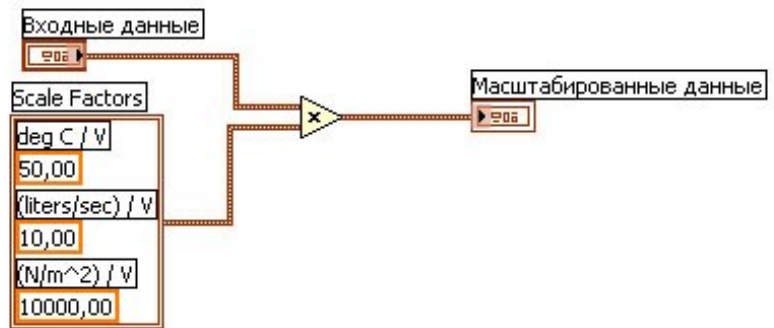
Лицевая панель

1. Откройте ВП *Масштабирование кластера.vi*, расположенный в папке *C:\Exercises\LV Basics I*. Лицевая панель уже подготовлена.
2. Измените значения элементов управления на лицевой панели.



Блок-диаграмма

3. Создайте блок-диаграмму, как показано ниже. Убедитесь, что значения элементов кластера **Входные данные** введены правильно.



4. Сохраните ВП.
5. Перейдите на лицевую панель и запустите ВП.
6. Поменяйте значения элементов управления и снова запустите ВП.
7. Закройте ВП.

Конец упражнения 7-2

Краткое изложение пройденного материала, советы и секреты

Кластеры объединяют элементы разных типов данных. Кластер не может одновременно содержать элементы управления и отображения данных.

Если лицевая панель содержит более 28 элементов, которые необходимо использовать в ВП, то некоторые из них можно объединить в кластер и назначить кластер полем ввода/вывода данных.

Для создания кластеров из элементов управления и отображения данных следует выбрать шаблон кластера на палитре **Controls>>Array & Cluster** и поместить его на лицевую панель. После этого шаблон кластера следует заполнить элементами.

Для создания и управления кластерами используются функции, расположенные на палитре **Functions>>Cluster**.

Проверка ошибок позволяет узнать, в каком месте и почему произошел сбой.

Кластер ошибок содержит элементы **status** (статус), **code** (код) и **source** (источник) ошибки.

Для создания входа и выхода ошибок в подпрограммах ВП используются кластеры ошибок из элементов управления и отображения.

Занятие 8

Отображение данных на диаграммах

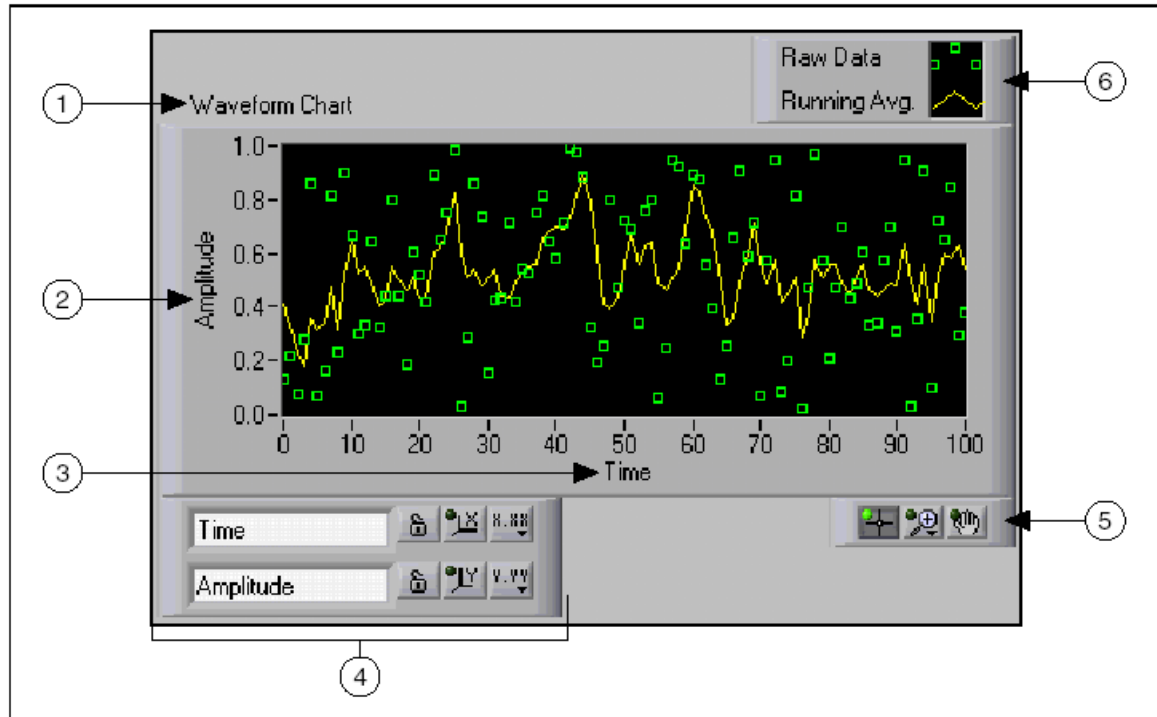
Рассмотрены способы визуализации данных с помощью графика Диаграмм (**Waveform Chart**).

Изложен вопрос:

- A. Использование графика Диаграмм для отображения потока данных.

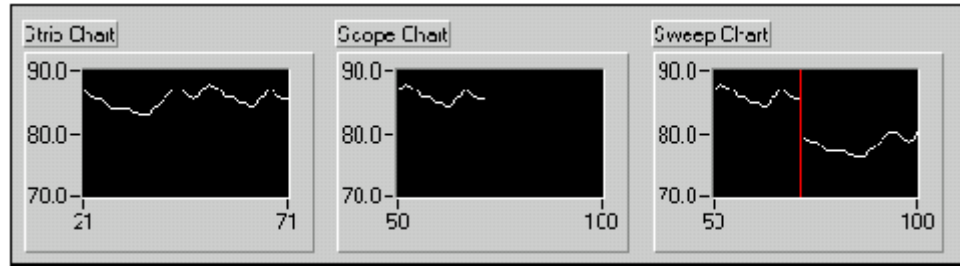
А. График Диаграмм

График Диаграмм (**Waveform Chart**)– специальный элемент отображения данных в виде одного и более графиков. График Диаграмм расположен на палитре **Controls>>Graph**. На рисунке показан пример Графика Диаграмм с двумя графиками: экспериментальные данные и их бегущее среднее значение.



6. Название (Label)
7. Шкала Y (Y-scale)
8. Шкала X (X-scale)
9. Панель управления шкалами (Scale legend)
10. Палитра инструментов для работы с графиком (Graph palette)
11. Панель управления графиком (Plot legend)

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.

Соединение графиков

Для создания диаграмм достаточно соединить поле вывода скалярной величины с терминалом данных графика Диаграмм. В следующем примере тип данных на терминале графика Диаграмм, соответствует входному типу данных.

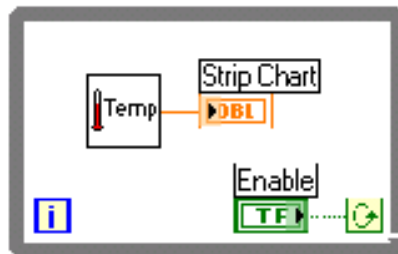
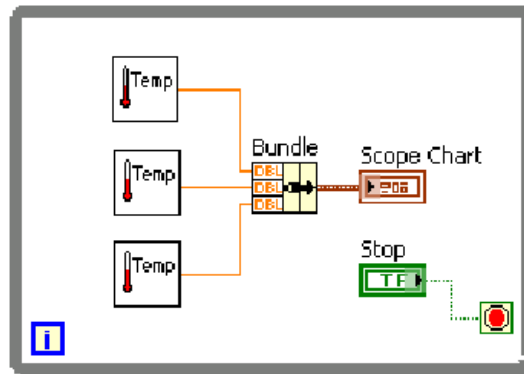


График Диаграмм может отображать несколько графиков. Для объединения отображаемых данных используется функция **Bundle**, расположенная в палитре **Functions>>Cluster**. Например, блок-диаграмма, показанная ниже, с помощью функции **Bundle** объединяет выходные данные трех подпрограмм ВП для последующего отображения на графике Диаграмм.



Терминал данных графика Диаграмм имеет кластерный тип данных в соответствии с полем вывода функции **Bundle**. Для увеличения количества полей ввода данных функции **Bundle** необходимо с помощью инструмента ПЕРЕМЕЩЕНИЕ изменить ее терминала.

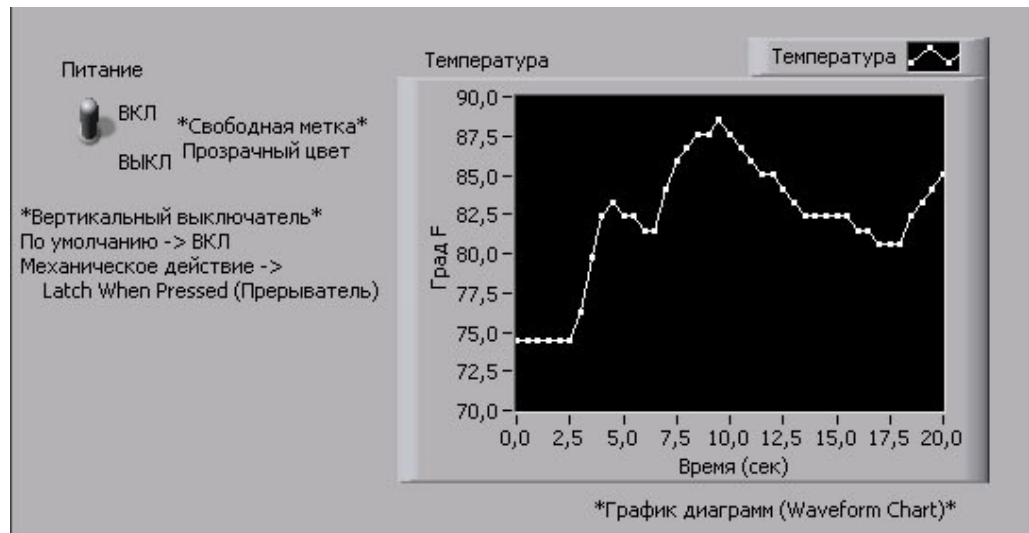
Упражнение 8-1 ВП Мониторинг температуры

Цель: Приобретение навыков по использованию цикла While и графика Диаграмм для получения и отображения данных.

Ниже приведена последовательность действий для создания ВП, который измеряет температуру и отображает результат в виде диаграммы.

Лицевая панель

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



Выберите логический элемент управления – вертикальный переключатель из палитры **Controls>>Boolean** и поместите его на лицевую панель, назвав его **Питание**. Выключатель будет использоваться для остановки выполнения ВП программным способом.

Выберите график Диаграмм (**Waveform Chart**) из палитры **Controls>>Graph** и поместите его на лицевую панель. На графике Диаграмм будет отображаться значение температуры в реальном масштабе времени. Введите текст **Температура** в поле собственной метки графика.



Обратите внимание на то, что на панели управления графиком (**chart legend**) введен текст **Plot 0**. Измените

текст на **Температура** с помощью инструмента ВВОД ТЕКСТА.

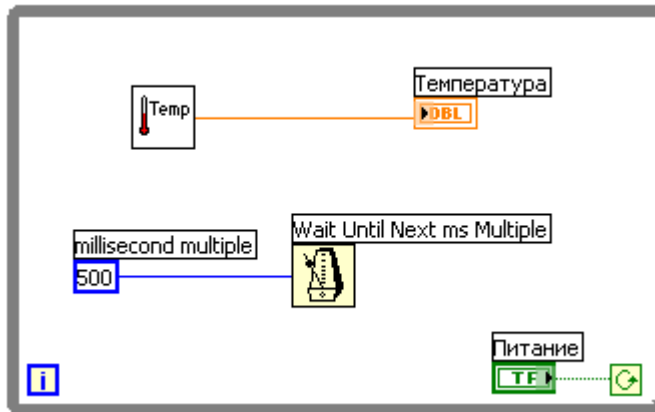
Датчик температуры измеряет комнатную температуру, поэтому с помощью инструмента ВВОД ТЕКСТА измените верхнее значение оси Y с **10** на **90** в °F. Для этого выделите значение **10,0** двойным щелчком инструмента и введите значение **90**.

Измените нижнее значение масштаба оси Y с **-10** на **70** в °F.

Введите названия осей Y – **Град. F**, X – **Время (сек)**.

Блок-диаграмма

13. Перейдите на блок-диаграмму, выбрав пункт главного меню **Window>>Show Diagram**.
14. Поместите терминалы данных в тело цикла **While**, как показано ниже.



Наведите курсор на терминал условия выхода из цикла и нажмите правую кнопку мыши. Выберите пункт контекстного меню **Continue if TRUE**.

15. Соедините элементы, как показано выше.



В палитре **Functions>>Select a VI** выберите ВП *Термометр.vi* (C: \exercises\LV Basics I) и поместите его на блок-диаграмму. ВП на выходе выдает измеренное датчиком значение температуры.



Поместите на блок-диаграмму функцию **Wait Until Next ms Multiple**, размещенную в палитре **Functions>>Time & Dialog**.

500

Щелкните правой кнопкой мышки по полю ввода **millisecond multiple** функции **Wait Until Next ms Multiple** и в контекстном меню выберите пункт **Create>>Constant**. Созданной константе присвойте значение **500**. Теперь каждая итерация цикла будет выполняться с интервалом времени 500 мс (один раз каждые полсекунды).



Примечание Для измерения температуры в °C необходимо поле ввода данных **шкала температуры ВП Термометр** соединить с логической константой, размещенной в палитре **Functions>>Boolean**. Установите значение константы TRUE. Измените значение диапазона оси Y с **70** на **20** и с **32** на **90**.

16. Сохраните ВП под именем *Мониторинг температуры.vi*.

Запуск ВП

17. Перейдите на лицевую панель, нажав клавиши «**Ctrl+E**» или выбрав пункт главного меню **Window>>Show Panel**.

18. С помощью инструмента УПРАВЛЕНИЕ переведите вертикальный переключатель в положение **ВКЛ**.

19. Запустите ВП.

Блок-диаграмма внутри границы цикла **While** выполняется до тех пор, пока значение терминала условия выхода равно TRUE. Например, пока элемент управления **Питание** находится в положение **ВКЛ** (TRUE), ВП **Термометр** осуществляет сбор и отображение текущих значений температуры в виде диаграммы.

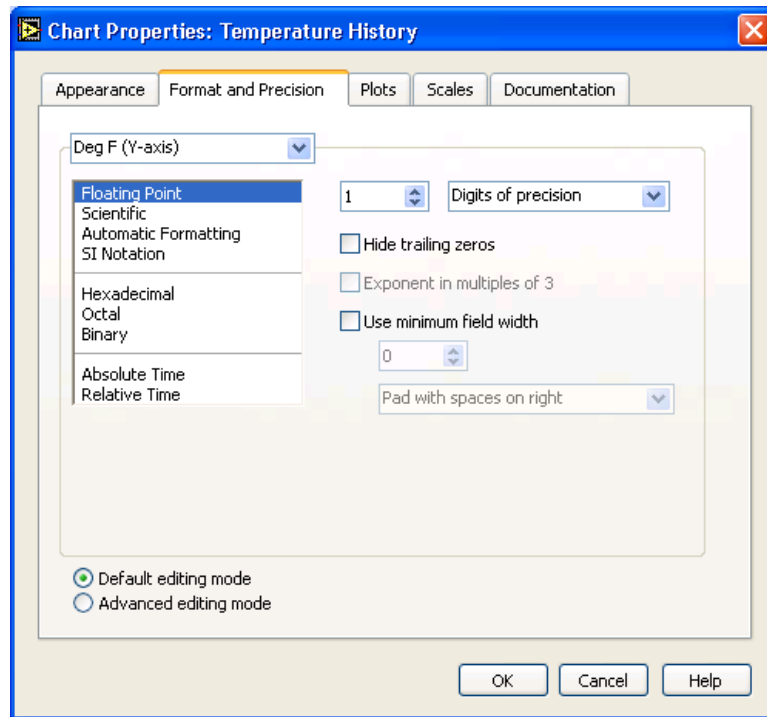
20. Остановите выполнение цикла переводом переключателя **Питание** в положение **ВЫКЛ**. Условие выхода примет значение FALSE и выполнение цикла прекратится.

Лицевая панель

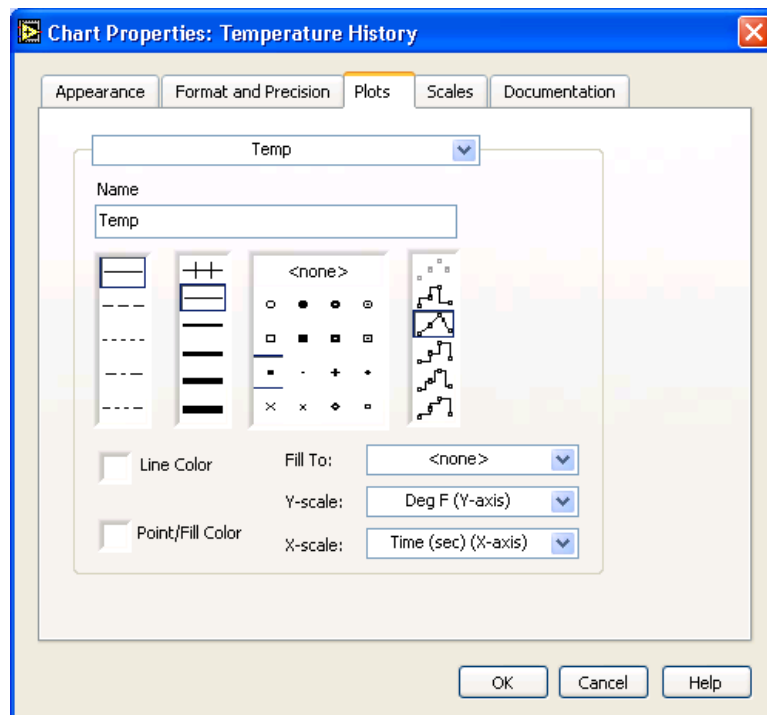
21. Настройте формат и масштаб шкал X и Y:

а. Щелкните правой кнопкой мыши по графику и выберите пункт **Properties**. Появится диалоговое окно **Chart Properties** (свойства графика). Перейдите на закладку **Format and Precision** и установите значение параметра **Digits of precision** (порядок

точности) равное 1.

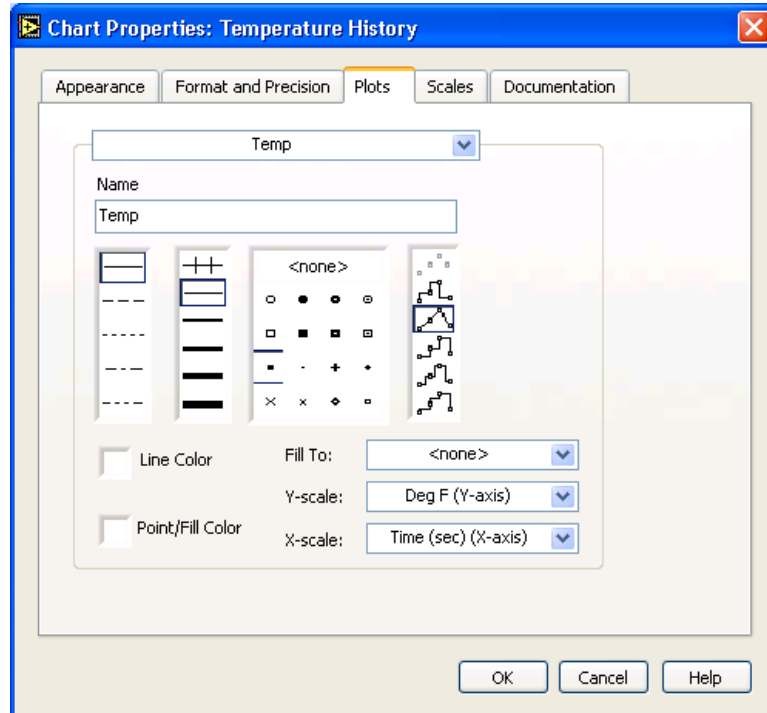


в. Нажмите на закладку **Plots** и просмотрите различные стили оси Y.

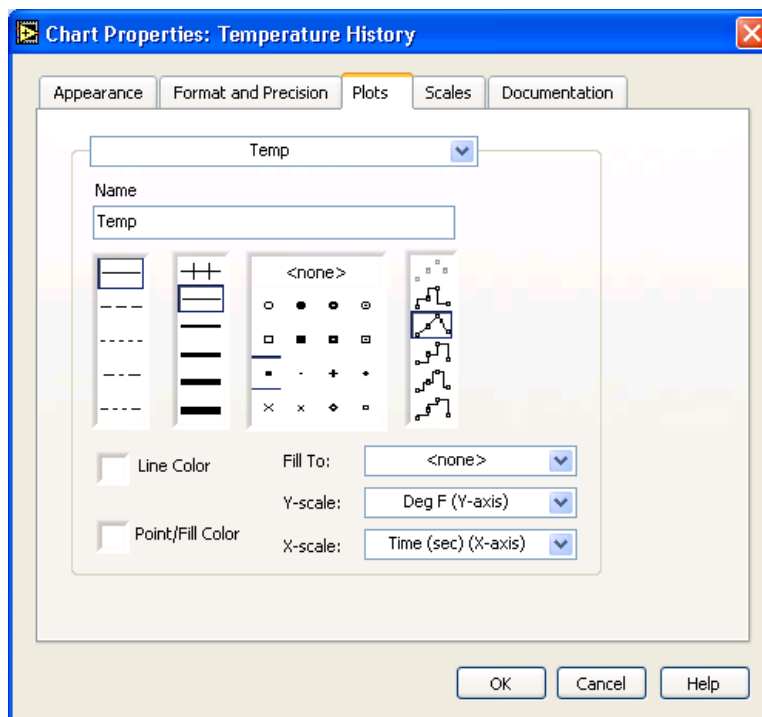


с. Перейдите на закладку **Scales** и выберите пункт

Время (сек) (X-axis) из верхнего меню. Установите параметры, как показано в диалоговом окне ниже. Установите значение параметра **Multiplier** равное **0,5**, так как теперь из-за функции ожидания **Wait** итерации считывания значения температуры происходят через 0,5 секунды.



- d. На закладке **Scales** выберите пункт **Град. F (Y-axis)** из выпадающего меню. Установите параметры, как показано в диалоговом окне ниже.



е. Установив нужные опции, нажмите кнопку **ОК**.

22. Щелкните правой кнопкой мыши по графику и выберите из контекстного меню пункт **Data Operations>>Clear Chart** для очистки буфера и окна отображения графика Диаграмм.



Совет Если ВП выполняется, можно выбрать пункт контекстного меню **Clear Chart**, вызываемого щелчком правой кнопки мыши.

23. В данный момент вертикальному переключателю сопоставлено такое механическое действие, что его необходимо возвращать в начальное положение каждый раз перед запуском ВП. Существует возможность изменить механическое действие переключателя так, чтобы он сам возвращался в исходное положение.

- a. Остановите ВП, если он запущен.
- b. С помощью инструмента УПРАВЛЕНИЕ переведите вертикальный переключатель в положение **ВКЛ**.
- c. Щелкните правой кнопкой мыши по переключателю и выберите из контекстного меню пункт **Data Operations>>Make Current Value Default**. Это делает положение **ВКЛ** положением переключателя по умолчанию.



- d. Щелкните правой кнопкой мыши по переключателю и выберите из контекстного меню пункт **Mechanical Actions>>Latch When Pressed**. Эта установка изменяет значение логического элемента после нажатия левой клавиши мышки и сохраняет его до первого обращения к нему ВП. После обращения ВП значение логического элемента возвращается в исходное положение. Это действие похоже на разрыв цепи и применяется для остановки цикла **While** или для однократного выполнения какой-либо операции.

Запуск ВП

24. Запустите ВП.

25. Инструментом УПРАВЛЕНИЕ нажмите на вертикальный переключатель для остановки процесса считывания температуры. Переключатель перейдет в положение **ВЫКЛ** и, после того как значение попадет на терминал условия выхода из цикла, переключатель вновь перейдет в положение **ВКЛ**.

26. Сохраните ВП. Этот ВП будет использован в упражнении 8-2.

Конец упражнения 8-1

Упражнение 8-2 ВП Расчет средней температуры

Цель: Научиться использовать сдвиговые регистры для реализации алгоритма «бегущее среднее»

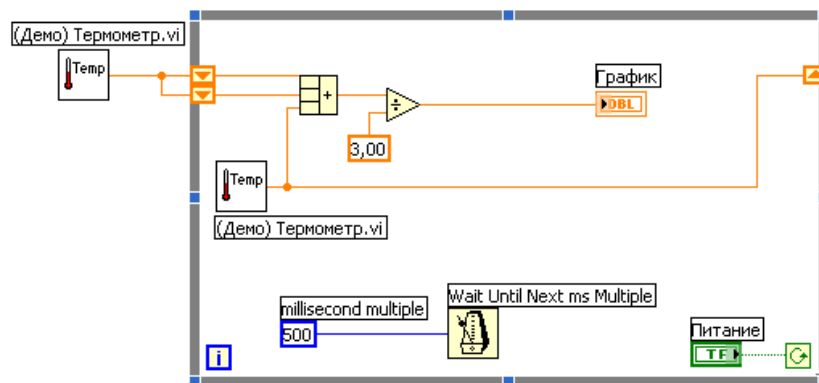
Ниже приведена последовательность действий для изменения ВП **Мониторинг температуры** путем добавления усреднения трех последних измерений температуры и отображения этого среднего значения на диаграмме.

Лицевая панель

1. Откройте ВП **Мониторинг температуры**, созданный в упражнении 7-1.
2. Выберите пункт главного меню **File>>Save As** и переименуйте ВП в *Расчет средней температуры.vi*

Блок-диаграмма

3. Перейдите на блок-диаграмму.
4. Щелкните правой кнопкой мыши по границе цикла и выберите пункт **Add Shift Register** в контекстном меню для создания правого и левого терминалов сдвигового регистра.
5. Щелкните правой кнопкой мыши на левом терминале сдвигового регистра и выберите пункт контекстного меню **Add Element** для добавления дополнительных терминалов к сдвиговому регистру.
6. Измените блок-диаграмму, как показано ниже на рисунке.





Нажав и удерживая клавишу **<Ctrl>**, переместите подпрограмму ВП **Термометр** за границу цикла, чтобы создать его копию.

(Macintosh) Нажмите клавишу **<Option>**. **(Sun)** Нажмите клавишу **<Meta>**. **(Linux)** Нажмите клавишу **<Alt>**.

ВП **Термометр** на выходе выдает одно измеренное датчиком значение температуры и инициализирует левые терминалы сдвигового регистра перед началом выполнения цикла.



Поместите на блок-диаграмму функцию **Compound Arithmetic**, расположенную в палитре **Function>>Numeric**. Эта функция возвращает сумму текущей температуры и двух предыдущих ее значений. С помощью инструмента ПЕРЕМЕЩЕНИЕ измените размеры функции таким образом, чтобы получить три поля ввода данных, показанные слева.



Поместите на блок-диаграмму функцию **Divide**, расположенную в палитре **Function>>Numeric**. Эта функция возвращает среднее значение последних трех измерений температуры.



Щелкните правой кнопкой мыши по полю ввода данных функции **Divide** и выберите пункт **Create>>Constant** контекстного меню. Созданной константе присвойте значение **3**.

7. Сохраните ВП.

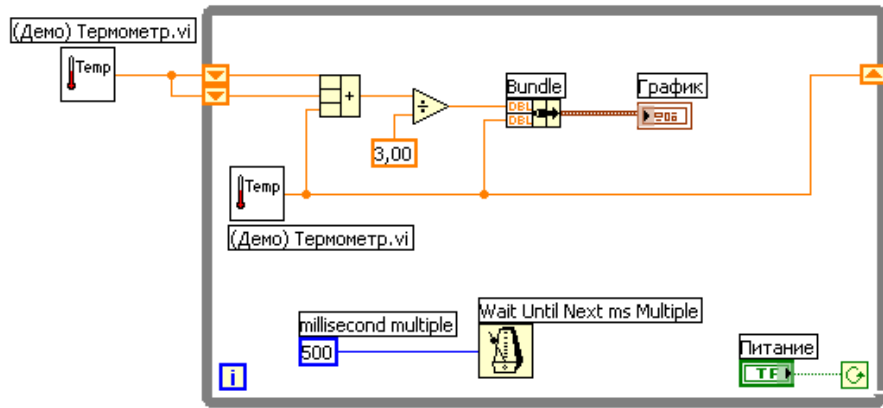
Запуск ВП

8. Запустите ВП.

В каждой итерации происходит измерение температуры **ВП Термометр**. ВП добавляет это значение к двум предыдущим, хранящимся в левых терминалах сдвигового регистра. Для нахождения среднего значения этих измерений их сумма делится на 3. Далее среднее значение выводится на график. Заметьте, что ВП инициализирует сдвиговый регистр значениями температуры.

Блок-диаграмма

9. Измените блок-диаграмму, как показано ниже.



Поместите на блок-диаграмму функцию **Bundle**, расположенную в палитре **Function>>Cluster**, для отображения полученного среднего значения и значения текущего измерения температуры на одном и том же графике Диаграмм. Эта функция связывает значения средней и текущей температуры для вывода на график.

10. Сохраните ВП, так как он будет использоваться в дальнейшем.

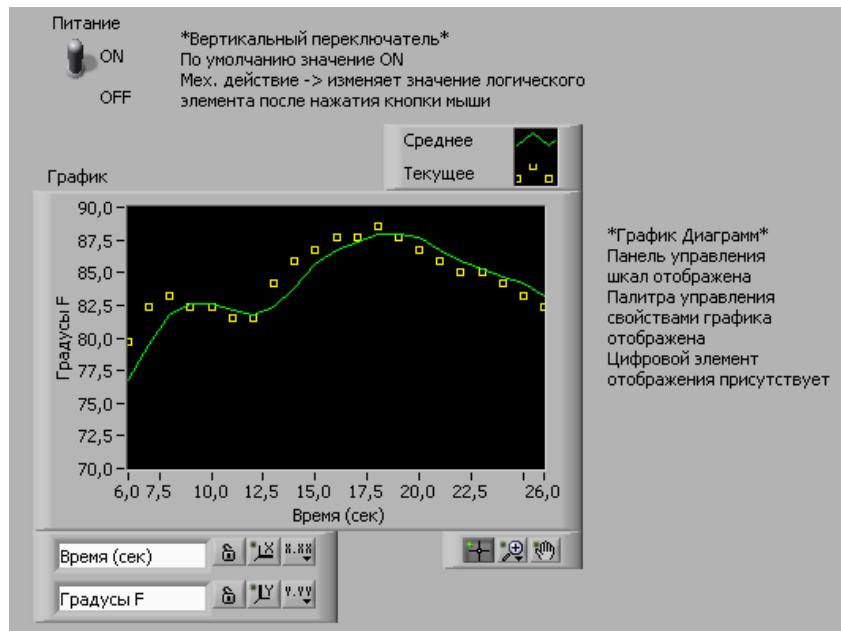
Запуск ВП

11. Запустите ВП. На график Диаграмм выводятся два накладывающихся графика, то есть два графика имеют одну шкалу Y.

12. При желании выполните дополнительную часть упражнения. В противном случае, закройте ВП.

Дополнительная часть

Далее приведена последовательность действий для настройки графика Диаграмм, показанного ниже. На лицевую панель можно вывести панель управления графиком Диаграмм, панель управления масштабом шкалы, палитру элементов управления диаграммой, цифровой элемент отображения, полосу прокрутки и содержимое буфера. По умолчанию, вместе с элементом график Диаграмм отображается панель управления свойствами графика.



13. Настройте ось Y.



- С помощью инструмента ВВОД ТЕКСТА измените значение с **70,0** по оси Y на **75,0**
- С помощью инструмента ВВОД ТЕКСТА измените второе значение на оси Y, напечатав **80,0**. Это число определяет шаг по оси Y.

Например, если следующее значение после **75,0** равно **77,5**, то интервал значений по оси Y составит **2,5**. Если изменить значение **77,5** на **80,0**, то интервал значений по оси Y будет равен **5,0** (**75,0**, **80,0**, **85,0** и так далее).



Примечание Размер графика Диаграмм напрямую связан с масштабом осей. При проблемах с настройкой осей попробуйте увеличить размер графика.

- Щелкните правой кнопкой мыши по полю графика Диаграмм и выберите пункт контекстного меню **Visible Items>>Scale Legend**, чтобы отобразить панель **Scale Legend**, как показано на следующем рисунке. Панель управления масштабом шкалы может быть перемещена в любое место лицевой панели.



- | | |
|------------------------------|-------------------------------|
| 1. Ось X
Lock) | 4. Кнопка блокировки (Scale |
| 2. Оси Y
(Autoscale) | 5. Кнопка автомасштабирования |
| 3. Метки курсоров
Format) | 6. Кнопка формат оси (Scale |

15. Настройте каждую ось, используя панель управления свойствами диаграммы **plot legend**.

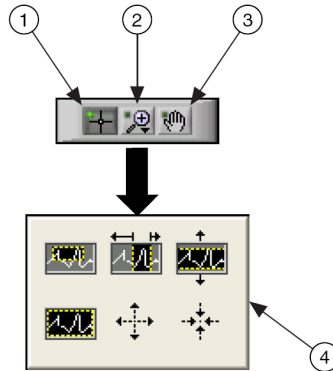
- a. Убедитесь, что кнопка **Autoscale** окрашена в зеленый цвет, а кнопка **Scale Lock** выключена. Таким образом, ось Y автоматически изменяет масштаб вывода на экран текущих значений.
- b. Нажмите кнопку **Scale Format** для изменения формата, точности, режима отображения, масштаба и вида размерной сетки для каждой оси.

16. С помощью панели **plot legend** настройте вид диаграмм, отображаемых на элементе график Диаграмм.

- a. С помощью инструмента ПЕРЕМЕЩЕНИЕ измените размер **plot legend** для отображения двух графиков.
- b. С помощью инструмента ВВОД ТЕКСТА измените метку **Температура** на **Среднее значение** и **Plot 1** на **Текущая температура**. Если текст получается слишком длинным, то следует увеличить размер **plot legend**, используя инструмент ПЕРЕМЕЩЕНИЕ.
- c. Щелкните правой кнопкой мыши по **plot legend** и вызовите контекстное меню с разделами редактирования цвета, фона, представления линий и

точек.

17. Щелкните правой кнопкой мыши по элементу график Диаграмм и выберите пункт контекстного меню **Visible Items>>Graph Palette** для отображения палитры элементов управления графиком **Graph Palette**, как показано ниже на рисунке. Палитра **Graph Palette** может быть перенесена в любую часть лицевой панели.



- 1 Инструмент перемещения курсора (**Cursor Movement**).
- 2 Кнопка масштабирования (**Zoom**).
- 3 Инструмент панорамирования (**Panning tool**).
- 4 Панель управления масштабом шкалы (**Zoom pull-down menu**).

Для изменения масштаба изображения следует использовать кнопку **Zoom**. Для быстрого перемещения по графику следует использовать инструмент **Panning**. Инструмент **Cursor Movement** позволяет перемещать курсор в поле графика.

18. Сохраните и запустите ВП. Во время работы ВП измените настройки графика Диаграммы с помощью панели управления масштабом шкалы **Scale Legend** и палитры элементов управления графиком **Graph Palette**.



Примечание При редактировании меток осей графика Диаграмм следует обратить внимание на то, что размер элемента должен быть больше, чем максимальный размер окна отображения диаграмм, заданный по умолчанию.

19. С помощью инструмента УПРАВЛЕНИЕ переведите

вертикальный выключатель **Питание** в положение **ВЫКЛ**
для завершения работы ВП.

20. Сохраните изменения и закройте ВП.

Конец упражнения 8-2

Краткое изложение пройденного материала, советы и секреты

График Диаграмм (**Waveform Chart**)- это специальный цифровой элемент отображения, предназначенный для отображения одной и более диаграмм.

График Диаграмм имеет три режима отображения данных:

- **strip chart** представляет собой экран, прокручиваемый слева направо подобно бумажной ленте.
- **scope chart**, по достижении правой границы окно диаграммы очищается, и заполнение диаграммы начинается с левой границы.
- **sweep chart**, в отличие от режима **scope chart**, окно диаграммы не очищается, а новые данные отделяются от старых вертикальной линией - маркером..

Для редактирования графика или изменения настроек графика Диаграмм необходимо щелкнуть по нему правой кнопкой мыши.

При выводе данных на диаграмму или график удобно использовать окно контекстной справки **Context Help**.

Дополнительные упражнения

- 8.3. Создайте ВП, строящий два графика: график произвольных чисел и бегущее среднее по четырем точкам на графике Диаграмм в режиме **sweep chart**. Воспользуйтесь следующими советами:

Используйте цикл **For** (N=200) вместо цикла **While**.

Используйте три левых терминала сдвигового регистра для усреднения последних четырех значений.

Для генерации данных используйте функцию **Random Number (0-1)**, расположенную в палитре **Functions>>Numeric**.

Для объединения произвольных чисел с их средним значением для построения на одном графике используйте функцию **Bundle**, расположенную в палитре **Functions>>Cluster**.

Сохраните ВП под именем *Бегущее среднее.vi*

- 8.4. Создайте ВП для непрерывного измерения температуры. Данные выводите на график Диаграмм в режиме **scope chart** с задержкой в 1 секунду. Если температура превысит предельное значение, введенное в соответствующий элемент управления, на лицевой панели должен загореться красный светодиод. График Диаграмм должен отображать текущие и предельные значения измерений температуры. Необходимо предусмотреть возможность изменения предельных значений температуры с помощью элементов отображения.

Сохраните ВП под именем *Предел температуры.vi*

- 8.5. Измените ВП, созданный в упражнении 8-4 так, чтобы он отображал только максимальные и минимальные значения текущих измерений температуры.



Совет Используйте сдвиговый регистр и две функции **Max & Min**, расположенных в палитре **Functions>>Comparison**.

Сохраните ВП под именем файла *Предел*

температуры (Max-Min).vi

Занятие 9

Отображение данных на осциллограммах

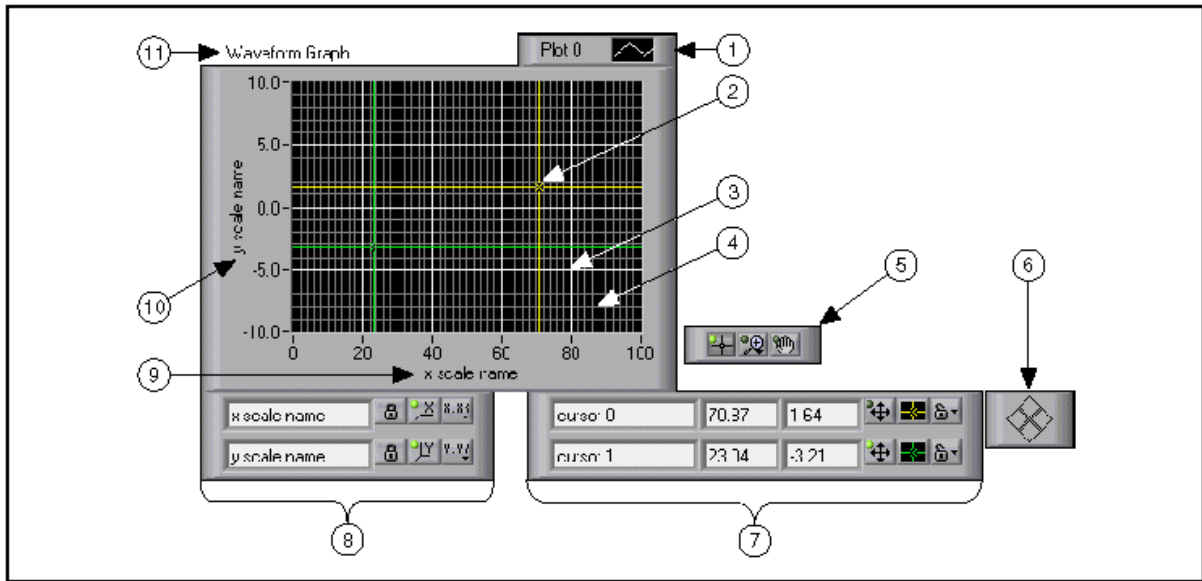
Рассмотрены способы визуализации данных с помощью графика Осциллограмм (**Waveform Graph**), двухкоординатного графика Осциллограмм (**XY graph**) и графика интенсивности (**Intensity Graph**).

Изложены вопросы:

- A. Использование графика Осциллограмм и двухкоординатного графика Осциллограмм для отображения данных.
- B. График интенсивности (дополнительно).

А. График Осциллограмм и двухкоординатный график Осциллограмм

С помощью графиков в виде осциллограмм ВП обычно отображает накопленные в массив данные. На следующей иллюстрации показаны элементы графика.



- | | | |
|--|--|---------------------------------------|
| 1. Панель управления свойствами осциллограмм (Plot legend) | 5. Палитра элементов управления графиком (Graph palette) | 9. Шкала X (X-scale) |
| 2. Курсор (Cursor) | 6. Панель перемещения курсора (Cursor mover) | 10. Шкала Y (Y-scale) |
| 3. Основная размерная сетка (Grid mark) | 7. Панель управления свойствами курсора (Cursor legend) | 11. Собственная метка графика (Label) |
| 4. Дополнительная размерная сетка (Mini-grid mark) | 8. Панель управления шкалой (Scale legend) | |

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.

Для отображения множества осциллограмм необходимо изменить размер панели **Plot legend**. График множества Осциллограмм используется с целью экономии пространства на лицевой панели и для сравнения осциллограмм данных между собой. График Осциллограмм и двухкоординатный график Осциллограмм автоматически поддерживают режим отображения множества осциллограмм.

Одиночный график Осциллограмм

Одиночный график Осциллограмм работает с одномерными массивами и представляет данные массива в виде точек на графике, с приращением по оси X равным **1** и началом в точке $x=0$. Графики также отображают кластеры, с установленным начальным значением x , Δx и массивом данных по шкале y . В качестве примера можно рассмотреть ВП **Waveform Graph VI** в разделе библиотеки примеров (*examples\general\ graphs\gengraph.llb*).

График множества Осциллограмм

График множества Осциллограмм работает с двумерными массивами данных, где каждая строка массива есть одиночная осциллограмма данных и представляет данные массива в виде точек на графике, с приращением по оси X равным **1** и началом в точке $x=0$.

Для представления каждого столбца двумерного массива данных в виде осциллограммы на графике необходимо соединить терминал данных массива с входным терминалом данных графика, затем щелкнуть правой кнопкой мыши по полю графика и выбрать пункт контекстного меню **Transpose Array** (транспонирование массива).

В качестве примера можно рассмотреть график **(Y) Multi Plot 1**, открыв ВП **Waveform Graph VI** из библиотеки примеров (*examples\general\ graphs\gengraph llb*).

Графики множества Осциллограмм отображают кластеры, состоящие из начального значения x , Δx и двумерного массива данных по шкале y . График представляет данные по шкале y в виде точек с приращением Δx по оси x и началом в точке $x=0$. В качестве примера можно рассмотреть график

(X₀, dX, Y) Multi Plot 3, открыв ВП **Waveform Graph VI** из библиотеки примеров (*examples\general\graphs\gengraph llb*).

Графики множества Осциллограмм отображают также и кластеры с установленным начальным значением x , Δx и массивом данных, содержащим кластеры. Каждый кластер содержит массив точек, отображающих данные по шкале Y . Для создания массива кластеров следует использовать функцию **Bundle**, которая объединяет массивы в кластеры. Далее, с помощью функции **Build Array** создается массив кластеров. Можно также использовать функцию **Build Cluster Array**, которая создает массив кластеров с определенными полями ввода данных. В качестве примера можно рассмотреть график **(X₀, dX, Y) Multi Plot 2**, открыв ВП **Waveform Graph VI** из библиотеки примеров (*examples\general\graphs\gengraph llb*).

Одиночные двухкоординатные графики Осциллограмм

Одиночный двухкоординатный график Осциллограмм работает с кластерами, содержащими массивы x и y . Двухкоординатный график Осциллограмм также воспринимает массивы точек, где каждая точка является кластером, содержащим значения по шкалам x и y . В качестве примера можно рассмотреть ВП **XY Graph VI** из библиотеки примеров (*examples\general\graphs\gengraph llb*).

Двухкоординатные графики множества Осциллограмм

Двухкоординатные графики множества Осциллограмм работают с массивами осциллограмм, в которых осциллограмма данных является кластером, содержащим массивы значений x и y . Двухкоординатные графики множества Осциллограмм воспринимают также массивы множества осциллограмм, где каждая осциллограмма представляет собой массив точек. Каждая точка – это группа данных, содержащая значения по x и y . В качестве примера можно рассмотреть ВП **XY Graph VI** из библиотеки примеров (*examples\general\graphs\gengraph llb*).

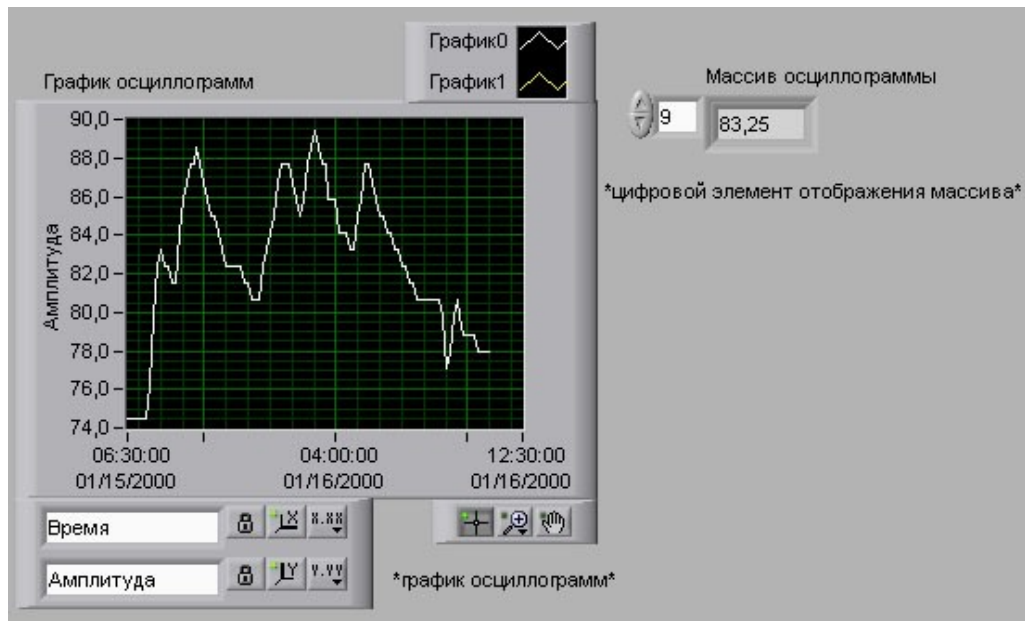
Упражнение 9-1 ВП Вывод массива данных на график

Цель: Создать массив, используя свойство автоиндексации цикла For и вывести данные массива на график Осциллограмм

Ниже приведена последовательность действий по созданию ВП, который отображает массив данных на графике Осциллограмм, и изменению ВП для отображения множества осциллограмм.

Лицевая панель

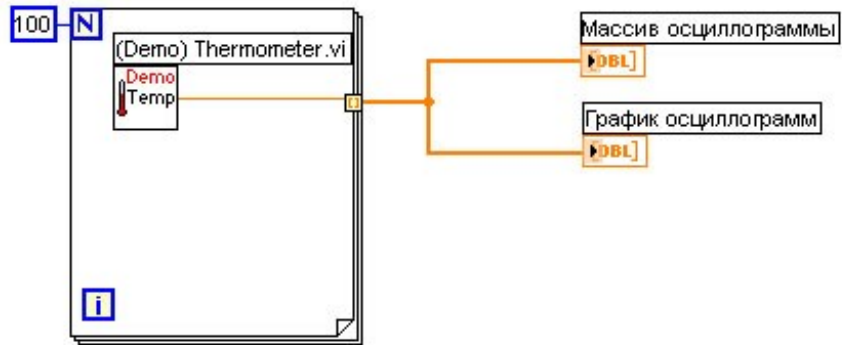
1. Откройте новый ВП и оформите лицевую панель, как показано ниже.



- a. Выберите шаблон массива, расположенный в палитре **Controls>>Array & Cluster**.
- b. Собственной метке массива присвойте имя **Массив осциллограмм**.
- c. Выберите цифровой элемент отображения данных, расположенный в палитре **Controls>>Numeric**, и поместите его в шаблон массива.
- d. Поместите на лицевую панель график Осциллограмм, расположенный в палитре **Controls>>Graph**.

Блок-диаграмма

2. Создайте блок-диаграмму, как показано ниже.



В палитре **Functions**>>**Select a VI** выберите ВП *Термометр.vi* (*c:\exercises\LV Basics I*), который был создан в упражнении 3-2. Поместите выбранный ВП на блок-диаграмму. В этом ВП каждая итерация цикла **For** выдает одно значение температуры.

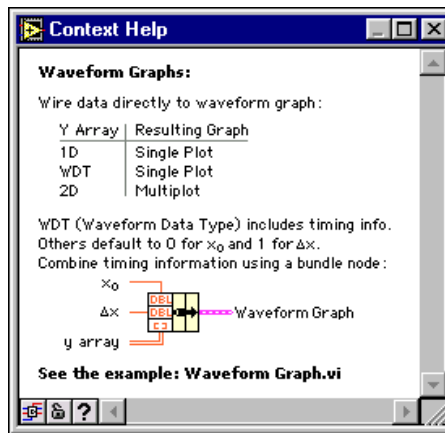


В палитре **Functions**>>**Structures** выберите цикл **For**. На каждой итерации цикла создается значение температуры и передается на терминал выхода из цикла. Терминалу максимального количества итерации цикла присвойте значение **100**.



Совет При передаче данных на диаграмму или график для определения порядка полей ввода данных и т.д. следует использовать окно контекстной справки **Context Help**, чтобы понять, какую функцию следует использовать – **Build Array** или **Bundle**. Для включения окна контекстной справки следует нажать **<Ctrl+N>**. В общем случае, графики Диаграмм следует использовать для вывода на экран скалярных точек, а графики Осциллограмм – массивов данных по *y*. Двухкоординатные графики Осциллограмм следует использовать для отображения массивов значений *x* и *y*.

Например, если на блок-диаграмме навести курсор на терминал данных графика Осциллограмм, в окне **Context Help** появится информация, показанная ниже. Для просмотра примера использования графика Осциллограмм ВП **Waveform Graph** следует выбрать в главном меню **Help**>>**Find Examples** и далее **Fundamentals**>>**Graphs and Charts**. Более подробную информацию о графическом типе данных можно получить в Уроке 10, *Сбор и отображение данных*.



3. Сохраните ВП под именем *Вывод массива данных на график.vi*

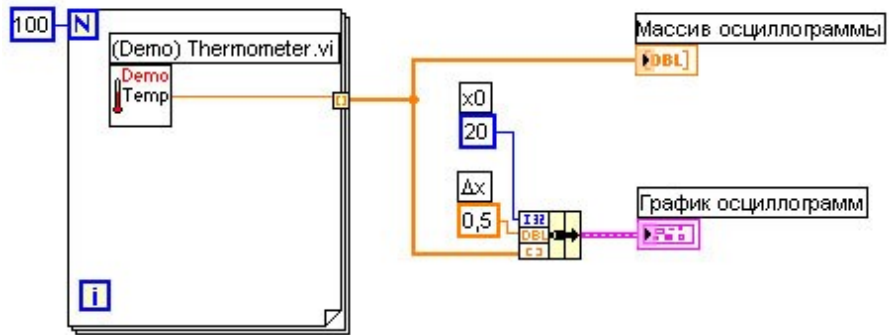
Запуск ВП

4. Перейдите на лицевую панель и запустите ВП. ВП построит осциллограмму данных массива на графике Осциллограмм.
5. Введите любое число в элемент **Массив осциллограммы** для просмотра соответствующего ему элемента массива. Введите число, превосходящее размер формируемого массива данных (100), значения этих элементов неопределенны.
6. С помощью инструмента ПЕРЕМЕЩЕНИЕ измените размер **Массива осциллограмм** для просмотра большего количества элементов. Элемент отображения данных располагает элементы по возрастанию значения индекса, начиная с введенного индекса.

Блок-диаграмма

На блок-диаграмме используются начальные значения x и Δx для графика Осциллограмм, заданные по умолчанию. В случае, когда необходимо установить значения x и Δx , отличные от значений заданных по умолчанию, следует использовать функцию **Bundle**.

3. Измените блок-диаграмму, как показано ниже.



В палитре **Functions>>Cluster** выберите функцию **Bundle**. Эта функция объединяет элементы в одномерный кластер. Элементы состоят из начального значения $x = 20$, $\Delta x = 0,5$ и массива данных y .

Создайте две числовые константы, выбрав их из палитры **Functions>>Numeric**, для установки начального значения x и Δx .

Собственную метку константы Δx назовите **Dx**. С помощью инструмента ВВОД ТЕКСТА выделите символ **D** и в выпадающем меню панели инструментов **Text Settings** выберите шрифт **Symbol**. Символ **D** превратится в Δ .

4. Сохраните ВП.

Запуск ВП

5. Перейдите на лицевую панель и запустите ВП.

График отобразит те же 100 точек с началом в точке 20 и $\Delta x = 0,5$ для каждой точки по оси x . Это соответствует проведению измерений с 20-ти секундной отметки в течение 50 секунд.



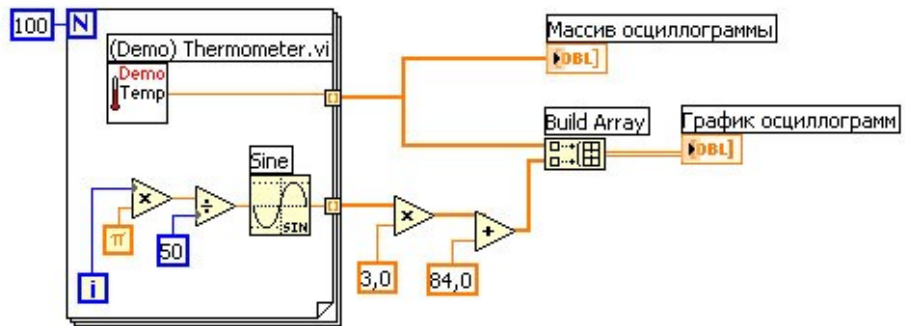
Примечание Изменять начальное значение x и Δx следует только в одном месте - или в функции **Bundle**, или в диалоговом окне свойств графика.

6. При желании, выполните дополнительную часть упражнения. В противном случае закройте ВП.

Дополнительная часть

7. Щелкните правой кнопкой мыши в поле графика Осциллограмм и из контекстного меню выберите пункт **Visible Items>>Graph Palette**. Нажмите кнопку **Zoom** для более детального рассмотрения данных на графике.

- Щелкните правой кнопкой мыши по полю графика и из контекстного меню выберите пункт **Visible Items>>Scale Legend**.
- Перейдите на блок-диаграмму. Выполните следующие пункты для создания графика множества Осциллограмм. Для этого нужно создать двумерный массив элементов типа данных, которые обычно подаются на одиночный график Осциллограмм. Измените блок-диаграмму как показано ниже.



Выберите функцию **Sine**, расположенную в палитре **Functions>>Numeric>>Trigonometric**. Эта функция будет использоваться для формирования массива данных, описывающих один цикл волны синуса.



Выберите функцию **Build Array**, расположенную в палитре **Functions>>Array**. Эта функция создает структуру данных для отображения двух массивов на графике Осциллограмм.



Выберите константу π , расположенную в палитре **Functions>>Numeric>>Additional Numeric Constants**.

- Сохраните ВП.
- Перейдите на лицевую панель и запустите ВП. Два графика отображаются на одном графике Осциллограмм.
- Перейдите на блок-диаграмму.
- Щелкните правой кнопкой мыши по проводнику данных к **Массиву осциллограмм**, из контекстного меню выберите

пункт **Custom Probe>>Graph>>Waveform Graph** для создания графического отладочного индикатора.

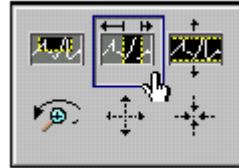
18.Перейдите на лицевую панель и запустите ВП. Отладочный индикатор показывает только массив данных. График синуса отсутствует потому, что отладочный индикатор не был помещен на проводник данных, который связан с синусом.

19.Закройте окно отладочного индикатора.

20.Измените масштаб части графика.



a. Нажмите на кнопку **Zoom** в палитре **graph palette**, показанную слева, для вывода на экран выпадающего меню, показанного ниже.



b. Выберите пункт **Zoom by X Rectangle**, как показано выше.

c. Удерживая кнопку мыши в нажатом состоянии, обведите курсором часть осциллограммы. После отпускания кнопки мыши выделенная область изменит масштаб.

d. Можно также выбрать **Zoom by Y Rectangle** или **Zoom by Selected Area**.



e. Для отмены изменения масштаба следует выбрать **Undo Zoom** из нижнего левого угла выпадающего меню или нажать на кнопку **x-axis** на панели **scale legend**, показанной слева.



21.Используя инструмент ПАНОРАМИРОВАНИЕ, показанный слева, переместите экран графика. Для возвращения экрана графика в его первоначальное положение следует нажать на кнопки **x-axis** и **y-axis** панели редактирования шкалы.



22.С помощью инструмента ПЕРЕМЕЩЕНИЕ КУРСОРА, показанного слева, верните курсор в стандартный режим.

23.Сохраните изменения и закройте ВП.

Конец упражнения 9-1

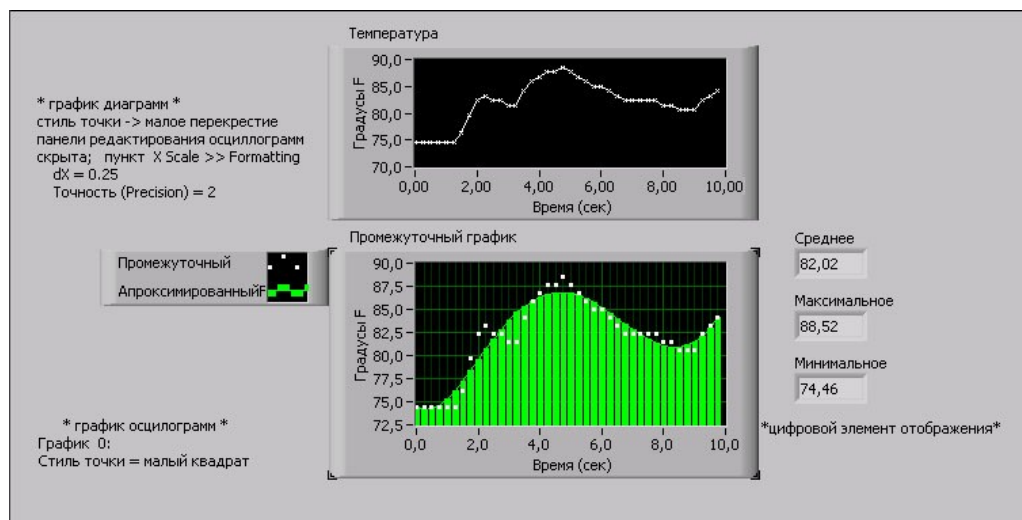
Упражнение 9-2 ВП Температурный анализ

Цель: Отобразить на графике и проанализировать данные

Ниже приведена последовательность действий для создания ВП, который измеряет температуру каждые 0,25 секунды в течение 10 секунд. В процессе измерения ВП в реальном масштабе времени отображает данные на графике Диаграмм. После завершения измерений, ВП выводит данные на график Осциллограмм и рассчитывает минимальную, максимальную и среднюю температуру. Кроме того, ВП отображает аппроксимацию осциллограммы температуры.

Лицевая панель

1. Откройте новый ВП. Используя советы, оформите лицевую панель, как показано ниже на рисунке.



Установите стиль точек на графике диаграмм в виде **x**.

Скройте панель управления свойствами диаграммы.

Щелкните правой кнопкой мыши по графику Диаграмм и из контекстного меню выберите пункт **X Scale>>Formatting**, установите значение **dX** равное **0,25**, а **Digits of Precision** (количество точек после запятой) равное **2**.

С помощью инструмента ПЕРЕМЕЩЕНИЕ измените размер панели **plot legend**.

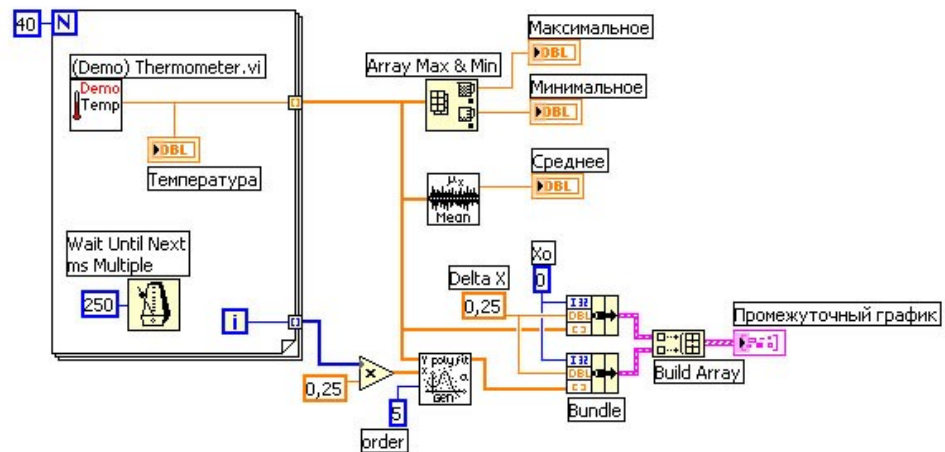
С помощью инструмента ВВОД ТЕКСТА переименуйте **График 0** в **Промежуточный**, а **График 1** - в **Аппроксимированный**.

Установите стиль точек осциллограммы **Промежуточный** в виде маленького квадрата.

Пока не создавайте элементы отображения данных **Среднее**, **Макс.** и **Мин.**

Блок-диаграмма

2. Постройте блок-диаграмму, как показано ниже.



Выберите в палитре **Functions>>Select a VI** и перейдите в директорию *c:\exercises\LV Basics I*. Выберите *Термометр.vi*, созданный в упражнении 3-2, и поместите его на блок-диаграмму. Этот ВП выдает одно измеренное значение температуры.



В палитре **Functions>>Time & Dialog** выберите функцию **Wait Until Next ms Multiple**. С помощью числовой константы на поле ввода функции подайте значение 250, что заставит цикл **For** выполняться каждые 0,25.



В палитре **Functions>>Array** выберите функцию **Array Max & Min**. Эта функция определяет минимум и максимум температуры.



В палитре **Functions>>Mathematics>>Probability and Statistics** выберите ВП **Mean VI**. Этот ВП определяет среднее значение измеренной температуры.

Щелкните правой кнопкой мыши по полю вывода данных **Array Max & Min** и ВП **Mean VI** и выберите в контекстном меню пункт **Create>>Indicator** для создания элементов **Макс.**, **Мин.** и **Среднее**.



В палитре **Functions>>Mathematics>>Curve Fitting** выберите ВП **General Polynomial Fit**. Этот ВП проведет аппроксимацию осциллограммы температуры.



В палитре **Functions>>Cluster** выберите функцию **Bundle**. Нажмите и удерживайте клавишу **<Ctrl>** во время перемещения функции для создания ее копии. Эта функция объединяет элементы в одномерный кластер. Элементы содержат начальное значение $x=0$, $\Delta x=0,25$ и массив значений температуры по y . Значение $\Delta x=0,25$ необходимо для того, чтобы ВП выводил значения температуры на график Осциллограмм каждые 0,25 секунды.



В палитре **Functions>>Array** выберите функцию **Build Array**. Эта функция создает массив кластеров из группы измеренных данных температуры и их аппроксимации.

3. Сохраните ВП под именем *Анализ температуры.vi*

Запуск ВП

4. Перейдите на лицевую панель и запустите ВП.

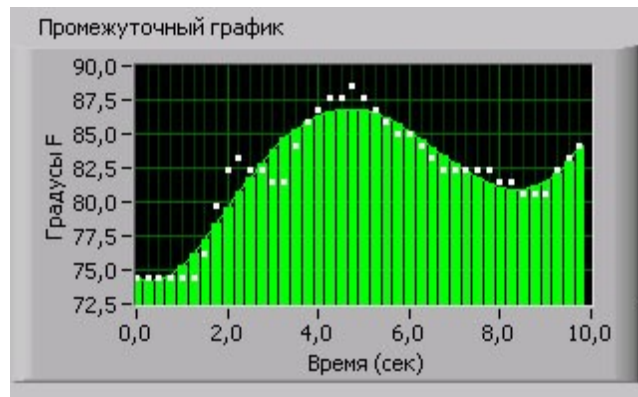
На графике Осциллограмм одновременно появятся осциллограммы данных температуры и их аппроксимации.

5. Поменяйте значения константы порядка аппроксимации на блок-диаграмме и снова запустите ВП.

6. Измените вид представления осциллограмм.

a. Щелкните правой кнопкой мыши по надписи **Промежуточный** на панели **Plot legend** и выберите в контекстном меню **Common Plots>>Scatter Plot** (экспериментальные точки).

b. Щелкните правой кнопкой мыши по надписи **Аппроксимированный** на панели **Plot legend** и в разделе **Bar Plots** контекстного меню выберите вторую иконку в средней строке. Получившийся график осциллограмм должен оказаться подобным изображенному ниже.



7. Сохраните и закройте ВП.

Конец упражнения 9-2

Краткое изложение пройденного материала, советы и секреты

График Осциллограмм (**Waveform Graph**) и двухкоординатный график Осциллограмм (**XY graph**) отображают данные из массивов.

Для редактирования графика или изменения настроек графика Диаграмм необходимо щелкнуть по нему правой кнопкой мыши.

На один график можно вывести более одной осциллограммы с помощью функции **Build Array**, расположенной в палитре **Functions>>Array**. На двухкоординатный график Осциллограмм данные вводятся с помощью функции **Bundle**, расположенной в палитре **Functions>>Cluster**.

Для визуализации трехмерных данных можно воспользоваться графиками и таблицами интенсивности. Третья координата отображается цветом в соответствии с определенной цветовой схемой. Графики и таблицы интенсивности часто применяются в задачах спектрального анализа, визуализации температуры и обработки изображений.

При выводе данных на диаграмму или график удобно использовать окно контекстной справки **Context Help**.

Занятие 10

Принятие решений в ВП

Излагаются методы принятия решений в ВП, такие как функция **Select**, структура **Case**, и узел **Формулы (Formula Node)**. В уроке также описываются возможности узла **Формулы** и ВП **Formula Express VI**.

Изложены вопросы:

- A. Функция **Select** и принятие решений.
- B. Использование структуры **Case**.
- C. Использование узла **Формулы**.

А. Функция **Select** и принятие решений

Каждый ВП до этого места курса выполнялся в порядке, определяемом потоком данных. Однако, бывают случаи, когда по ходу программы должно быть принято решение. Например, если происходит событие А, то необходимо сделать В, а если происходит С - то D.

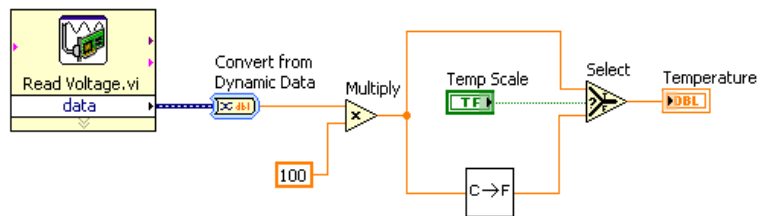
В программах, написанных на текстовых языках программирования, эта задача решается операторами if - else, операторами case, switch и т.д. В LabVIEW реализовано много различных способов принятия решений. Самый простой из них - функция **Select**.

Функция **Select**



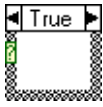
Функция **Select**, расположенная в палитре **Controls>>Comparison**, в зависимости от значения на логическом входе выбирает одно из двух значений. Если на логическом входе будет значение TRUE, то выходе функция выдаст значение, поданное на вход **t**, если же на логическом входе FALSE, то возвращается значение с поля **f**.

Функция **Select** использовалась в упражнении 3-3, **ВП Термометр**, для определения температурной шкалы. Блок-диаграмма упомянутого ВП приведена ниже.

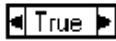


При необходимости принимать более сложные решения может понадобиться структура **Case**.

В. Структура Case.



Структура Case, показанная слева, имеет две или более поддиаграммы вариантов. Только одна поддиаграмма варианта видима в данный момент времени и только одна поддиаграмма варианта работает при выполнении данной структуры. Входное значение терминала селектора структуры определяет, какая поддиаграмма будет выполняться в данный момент времени. Структура **Case** аналогична операторам case или логическим операторам (if...then...else) в текстовых языках программирования.



Селектор структуры **Case**, расположенный сверху графического изображения Структуры, показанный слева, состоит из указателя значения варианта в центре и стрелок прокрутки по сторонам. Эти стрелки используются для просмотра возможных Вариантов.



Значение, подаваемое на терминал селектора варианта, показанный слева, определяет, какая поддиаграмма структуры, или вариант, будет выполняться. Допустимо использовать целочисленный, логический, строковый типы, а также тип перечисления в качестве значения, подаваемого на терминал варианта. Терминал варианта может располагаться в любом месте левой границы структуры **Case**. Если терминал Варианта логического типа, то структура состоит из двух логических вариантов TRUE и FALSE. Если терминал варианта имеет один из следующих типов: целочисленный, строковый или перечисления, то количество вариантов может достигать $2^{31}-1$ вариантов.

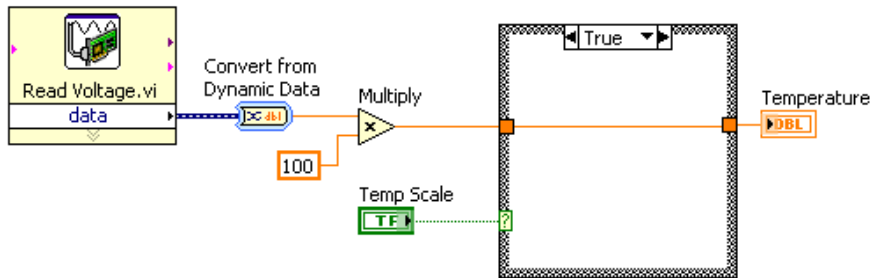
Для использования структуры **Case** необходимо отметить вариант по умолчанию. Вариант по умолчанию или поддиаграмма по умолчанию выполняется, если значение терминала варианта выходит за пределы диапазона или не существуют вариантов для возможных значений терминала варианта.

Щелчок правой кнопки мыши на границе структуры **Case** позволяет добавлять, дублировать, перемещать и удалять варианты (поддиаграммы), а также отмечать

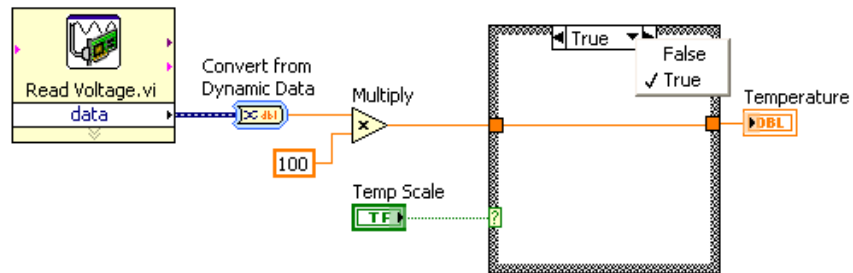
вариант по умолчанию.

Выбор варианта

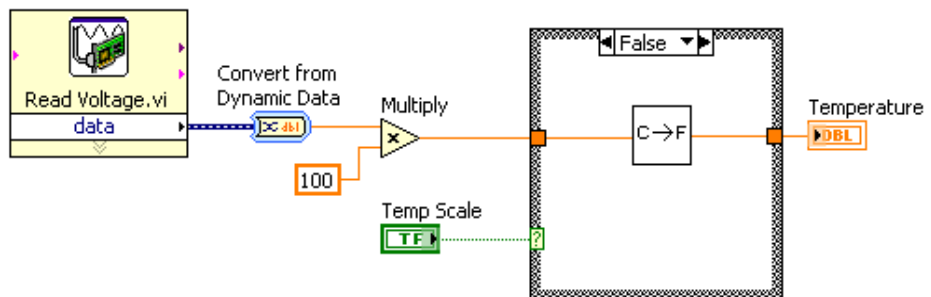
В качестве примера использования структуры **Case** вместо функции **Select** приведена измененная блок-диаграмма **ВП Термометр**. На переднем плане структуры **Case** показан логический вариант TRUE.



Определение варианта осуществляется либо выбором значения на селекторе структуры **Case**, либо вводом значения с помощью инструмента ВВОД ТЕКСТА.



При выборе какого-либо варианта, он появляется на переднем плане, как показано на следующей блок-диаграмме.



Значения селектора варианта должны быть того же типа, что и тип данных, подаваемых на терминал селектора варианта. Значение селектора варианта, окрашенное красным цветом, показывает, что его необходимо удалить или

отредактировать, иначе ВП не будет выполняться. Нельзя подавать числа с плавающей точкой на терминал селектора варианта, так как возможны ошибки округления и возникновение ситуации неопределенности. Если подать число с плавающей точкой на терминал селектора варианта, LabVIEW округлит это значение до ближайшего четного целого. Если число с плавающей точкой введено непосредственно в селектор варианта, то оно окрашивается в красный цвет и должно быть удалено или отредактировано.

Терминалы входа и выхода

Структура **Case** допускает использование входных и выходных терминалов данных. Терминалы входных данных доступны во всех поддиаграммах, но их использование поддиаграммой структуры необязательно. Создание выходного терминала на одной поддиаграмме приводит к его появлению на других поддиаграммах в том же самом месте границы структуры. Если хотя бы в одной поддиаграмме выходной терминал не определен, то поле этого терминала окрашивается в белый цвет, что говорит об ошибке создания структуры. Необходимо определять значения выходных терминалов во всех вариантах (поддиаграммах). Кроме того, выходные терминалы должны иметь значения совместимых типов.

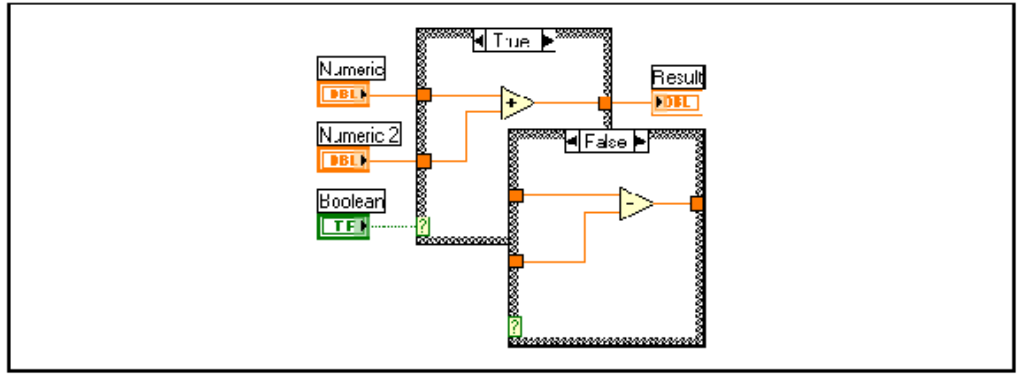
Для определения значения выходного терминала следует правым щелчком мыши по терминалу вызвать контекстное меню и выбрать пункты: **Create>>Constant** или **Create>>Control**.

Примеры

Следующие примеры показывают, как значения входных терминалов структуры **Case** складываются или вычитаются в зависимости от значения терминала варианта.

Логическая структура Case

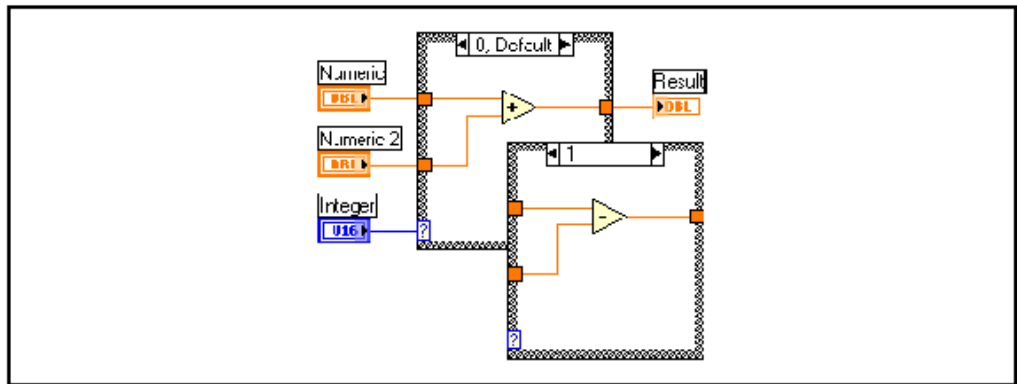
Ниже на рисунке приведен пример логической структуры **Case**. Варианты структуры наложены друг на друга для упрощения иллюстрации.



Если в терминал логического элемента управления, соединенный проводником данных с терминалом селектора варианта, введено значение TRUE, то выполняется сложение; если введено значение FALSE, то выполняется вычитание значений числовых элементов управления.

Целочисленная структура Case

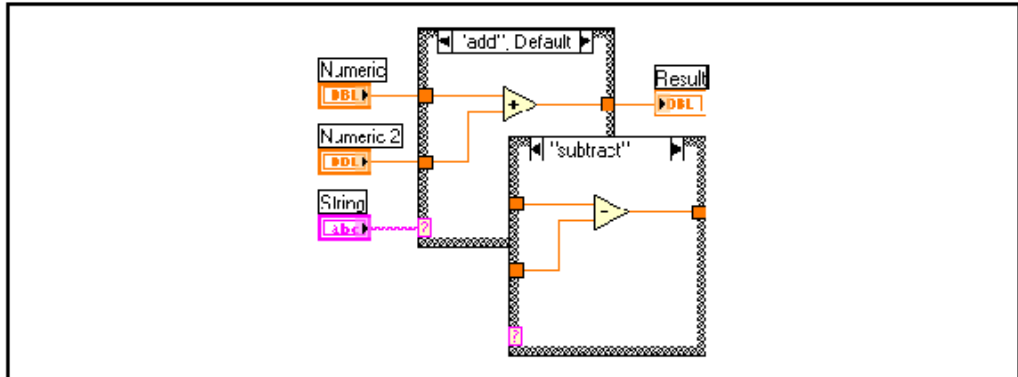
Ниже на рисунке показан пример целочисленной структуры Case.



Терминал **Integer** соответствует элементу управления **ring control** (списка с циклическим перебором значений), расположенного в палитре **Controls>>Ring & Enum**. Если значение элемента управления **ring control** равно **0** (сложить), то ВП складывает числа; если равно **1** (вычесть), то ВП производит вычитание чисел. Если значение элемента управления отлично от **0** (сложить) и **1** (вычесть), то ВП складывает числа, т.к. этот вариант выполняется по умолчанию.

Строковая структура Case

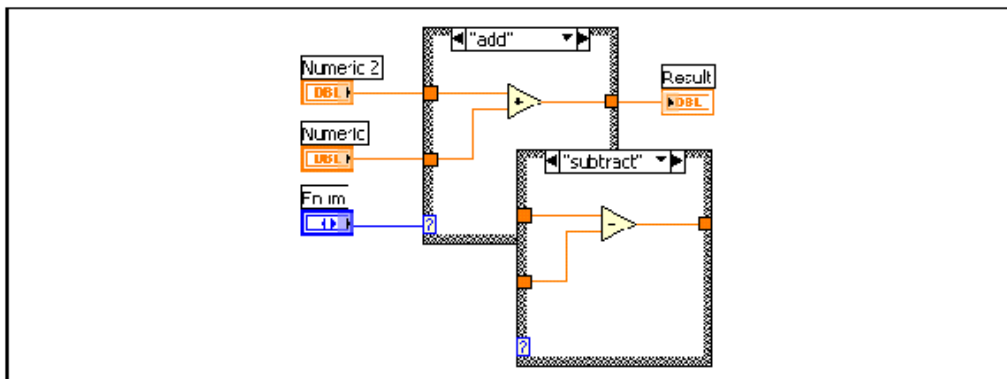
Ниже на рисунке показан пример строковой структуры **Case**.



Если в поле элемента управления введена строка **add**, то ВП производит сложение чисел и вычитает их, если введено значение **subtract**.

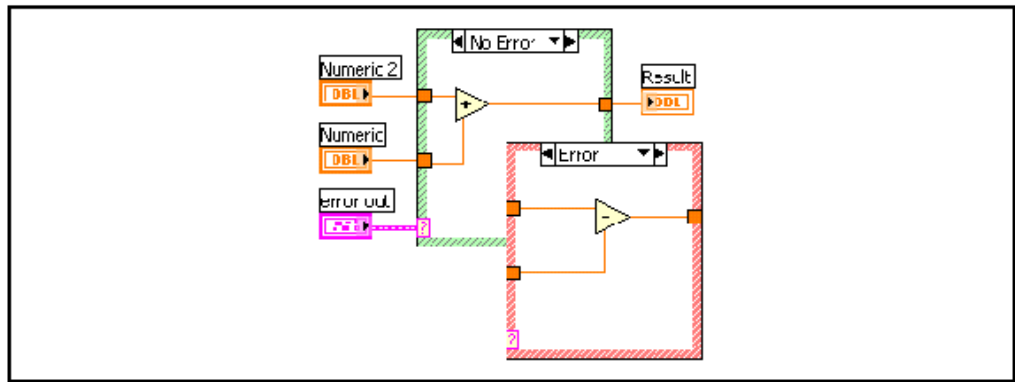
Структура Case по перечислениям

Ниже на рисунке показан пример структуры **Case** по перечислениям.



Структура Case для кластера ошибок

Ниже на рисунке показан пример структуры **Case** для кластера ошибок.



В этом примере на терминал селектора структуры **Case** подается кластер ошибок **error out**. В этом случае есть только два варианта структуры: **Ошибка** и **Нет ошибки**, для которых граница структуры имеет красный и зеленый цвет соответственно. Структура **Case** выполняет вариант, основываясь на информации о наличии ошибки.

Структура **Case** реагирует только на логическую переменную **status** кластера ошибок.

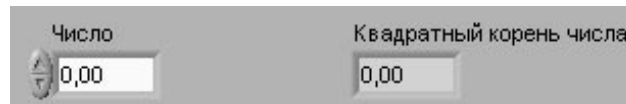
Упражнение 10-1 ВП Извлечение квадратного корня

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.

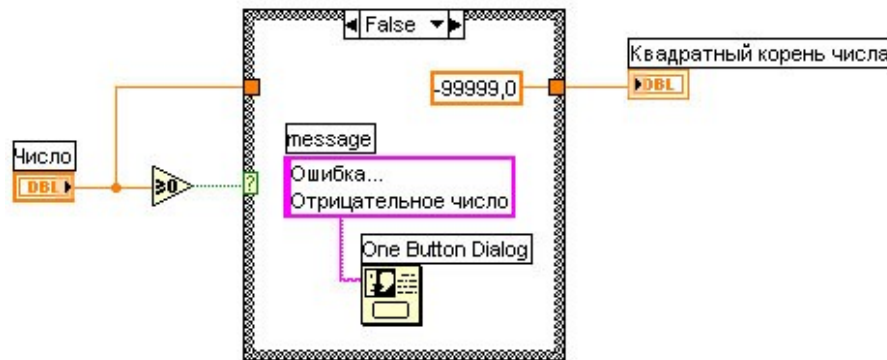
Лицевая панель

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

-99999.0

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».

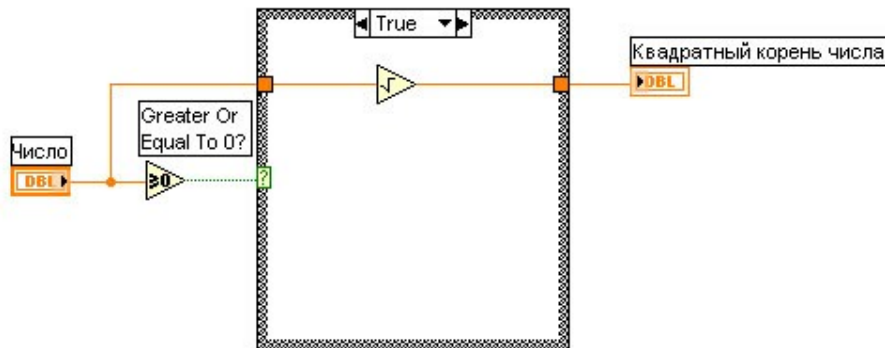
message
Ошибка...
Отрицательное число

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.



Внимание Не запускайте ВП кнопкой непрерывного запуска, так как при определенных обстоятельствах запуск этого ВП в непрерывном режиме может привести к бесконечному циклу.

Если входное значение элемента управления **Число** положительно, то выполнится поддиаграмма варианта

TRUE и вычисляется значение квадратного корня. Если значение элемента **Число** является отрицательным, то выполнится поддиаграмма варианта FALSE, которая возвращает -99999,0 и отображает диалоговое окно с сообщением «Ошибка... Отрицательное число».

6. Закройте ВП.

Конец упражнения 10-1

С. Узел Формулы

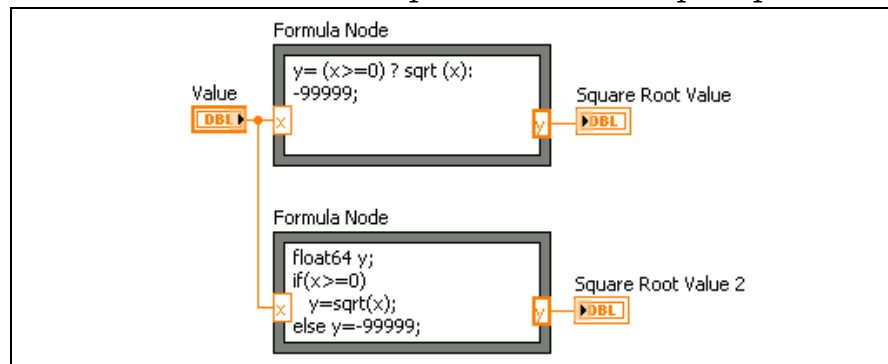
Узел **Формулы (Formula Node)** используется для выполнения математических операций в среде LabVIEW.

Узел Формулы

Узел **Формулы** используется для выполнения математических операций в текстовом виде на блок-диаграмме. Использовать узел **Формулы** удобно, когда выражения имеют много переменных, или они достаточно сложные, а также для уже имеющихся текстовых математических кодов. Для ускорения процесса создания алгоритма можно копировать и вставлять имеющиеся текстовые математические коды в узел **Формулы** вместо их воссоздания на блок-диаграмме.

Создание терминалов входных и выходных данных узла **Формулы** осуществляется щелчком правой кнопки мыши по границе узла. В контекстном меню необходимо выбрать пункты **Add Input** или **Add Output**, а затем ввести переменные для входа и выхода. Далее вводится уравнение в рабочую область структуры. Каждое выражение должно заканчиваться разделителем (;).

Узел **Формулы** может также использоваться для принятия решений. На следующей блок-диаграмме показаны два эквивалентных способа применения операторов **if - then** в



узле **Формулы**.

Узел **Формулы** позволяет производить разнообразные математические операции. Для получения более подробной информации о функциях, операциях и синтаксисе узла **Формулы** используйте справку **LabVIEW Help**.



Замечание ВП **Formula Express**, расположенный в палитре **Functions >> Express >> Arith/Compare**, является, по сути, встроенным в LabVIEW научным калькулятором. Он может

выполнять большинство операций, выполняемых узлом Формулы, однако только по одной операции за раз. Для получения более подробной информации о ВП **Formula Express** используйте справку **LabVIEW Help**.

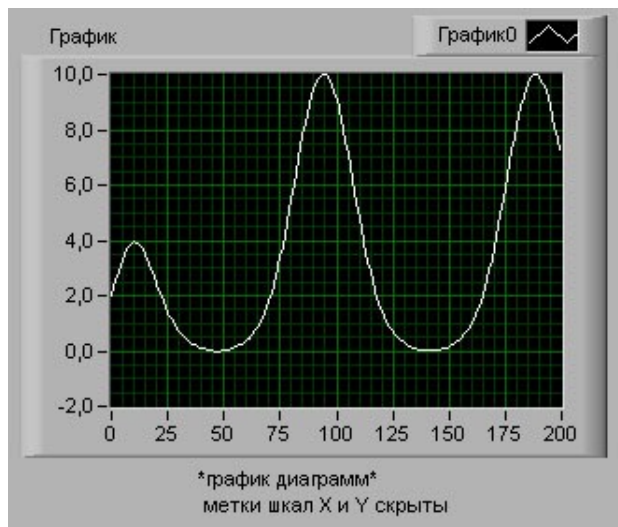
Упражнение 10-3 ВП Узел Формулы

Цель: Изучить работу структуры узел **Формулы**.

Выполните следующие пункты для построения ВП, который использует узел **Формулы** для выполнения нескольких математических операций и выводит результаты в виде графика.

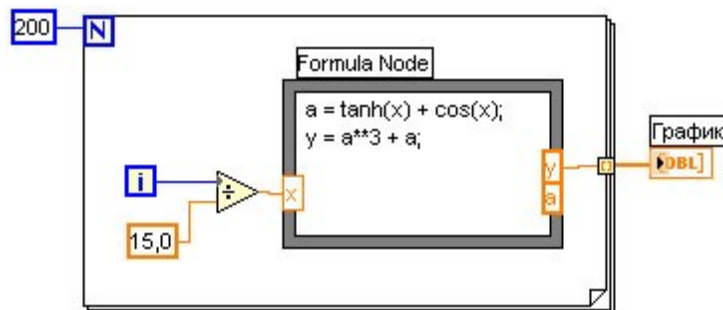
Лицевая панель

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



Блок-диаграмма

2. Постройте блок-диаграмму, показанную на рисунке:



Поместите на блок-диаграмму узел **Формулы (Formula Node)**, расположенный в палитре **Functions>>Structures**.

3. Создайте входной терминал **x**, щелкнув правой кнопкой мыши по левой границе структуры и выбрав пункт **Add Input** из контекстного меню. Введите значение **x** в появившееся окно.
4. Создайте терминалы выходных данных **y** и **a**, щелкнув правой кнопкой мыши по левой границе структуры и выбрав пункт **Add Output** из контекстного меню. Допустимо открытие выходного терминала для промежуточных переменных, например **a**.



Замечание При открытии входных или выходных терминалов имя переменной должно точно соответствовать имени переменной, используемой в арифметическом выражении. Также необходимо учитывать то, что прописные и заглавные буквы различаются.

5. Введите следующие уравнения в узел Формулы (** - это оператор степени). Для получения информации о синтаксисе узла Формулы используйте справку **LabVIEW Help**.

$$a = \tanh(x) + \cos(x);$$

$$y = a^{**3} + a;$$

6. Сохраните ВП под именем *Узел Формулы.vi*

Запуск ВП

7. Отобразите лицевую панель и запустите ВП. На графике отобразится кривая, соответствующая функции $y = f(x)^3 + f(x)$, где $f(x) = \tanh(x) + \cos(x)$.

В каждой итерации ВП делит значения терминала текущей итерации на 15,0. Частное является аргументом **x** узла Формулы, по которому рассчитывается значение функции **y**. Далее ВП строит график массива **y**.

8. Закройте ВП.

Конец упражнения 10-3

Краткое изложение пройденного материала, советы и секреты

Функция **Select** возвращает одно из двух входных значений в зависимости от значения на третьем логическом входе.

Структура **Case** может иметь два или более вариантов (поддиаграмм). Только одна поддиаграмма видна на блок-диаграмме, и только одна поддиаграмма может выполняться в момент времени.

Если терминал варианта логического типа, то структура состоит из двух логических вариантов: TRUE и FALSE. Если терминал варианта имеет типы: целочисленный, строковый или перечисления, то количество вариантов может достигать значений $2^{31}-1$.

Для того чтобы избежать выполнения подпрограммы ВП в случае ошибки необходимо подсоединить входной кластер ошибок к терминалу селектора и поместить весь код подпрограммы ВП в Вариант **Нет ошибки** структуры **Case**.

Использование узла Формулы полезно, когда арифметическое выражение имеет много переменных или достаточно сложное, а также для уже имеющегося текстового математического кода. Каждое выражение должно заканчиваться разделителем (;).

Дополнительные упражнения

- 10- Создайте ВП, который использует узел Формулы для
4. расчета следующих уравнений:

$$y_1 = x^3 + x^2 + 5$$

$$y_2 = mx + b$$

Введите оба уравнения в узел Формулы, используйте разделитель (;) после каждого уравнения.

Сохраните ВП как файл *Уравнения.vi*

- 10- Создайте ВП, функционирующий как калькулятор.
5. Используйте элементы цифрового управления для ввода численных значений и элемент цифрового отображения для вывода результата арифметических операций: сложение, вычитание, деление, умножение, которые ВП выполняет над вводимыми числами. Для выбора арифметической операции рекомендуется использовать элемент управления **slide control**.

Сохраните ВП под именем *Калькулятор.vi*

- 10- Измените ВП *Извлечение квадратного корня.vi*,
6. созданный в упражнении 8-1, так чтобы ВП выполнял все арифметические и логические операции с помощью узла Формулы.

Сохраните ВП под именем *Извлечение квадратного корня 2.vi*.



- 10- Создайте ВП, который имеет два элемента
7. управления с именами **Порог** и **Входной Массив**, а также один элемент отображения **Выходной массив**. **Выходной массив** должен состоять из элементов **Входного Массива**, значения которых превысят значение **Порога**.

Сохраните ВП под именем *Превышение над порогом.vi*

Создайте другой ВП, который генерирует массив случайных чисел в диапазоне от 0 до 1 и использует

ВП *Превышение над порогом.vi* в качестве подпрограммы для создания выходного массива значений элементов превышающих порог 0,5.

Сохраните ВП под именем *Использование ВП Превышение над порогом.vi*

Занятие 11

Строки и файловый ввод/вывод

Строки объединяют последовательности и являются массивами ASCII символов. Подпрограммы ВП работы с файлами обеспечивают ввод/вывод данных в/из файл(а).

Изложены вопросы:

- A. Создание строковых элементов управления и отображения данных.
- B. Использование некоторых функций обработки строк.
- C. Использование файловых функций ввода/вывода.

А. Строки

Строки - это последовательность отображаемых и неотображаемых ASCII символов. Строки обеспечивают независимый от платформы формат обмена данными. Некоторые из наиболее распространенных строковых приложений включают в себя:

Создание простых текстовых сообщений.

Передача числовых данных в приборы в виде строк символов и преобразование строк в числовые данные.

Сохранение числовых данных на диск. Чтобы сохранять числовые данные в виде файла ASCII, необходимо перед записью преобразовать их в строки.

Диалоговые окна инструкций и подсказок.

На лицевой панели строки появляются в виде таблиц, полей ввода текста и меток.

Создание строковых элементов управления и отображения данных

Для работы с текстом и метками используются строковые элементы управления и отображения данных, расположенные в палитре **Controls>>String & Path**. Создание и редактирование текста в строке производится с помощью инструментов УПРАВЛЕНИЕ и ВВОД ТЕКСТА. Для изменения размера строкового объекта на лицевой панели используется инструмент ПЕРЕМЕЩЕНИЕ. Для экономии места на лицевой панели можно использовать полосу прокрутки. Для этого необходимо щелкнуть правой кнопкой мыши по строковому объекту и выбрать в контекстном меню пункт **Visible Items>>Scrollbar**.

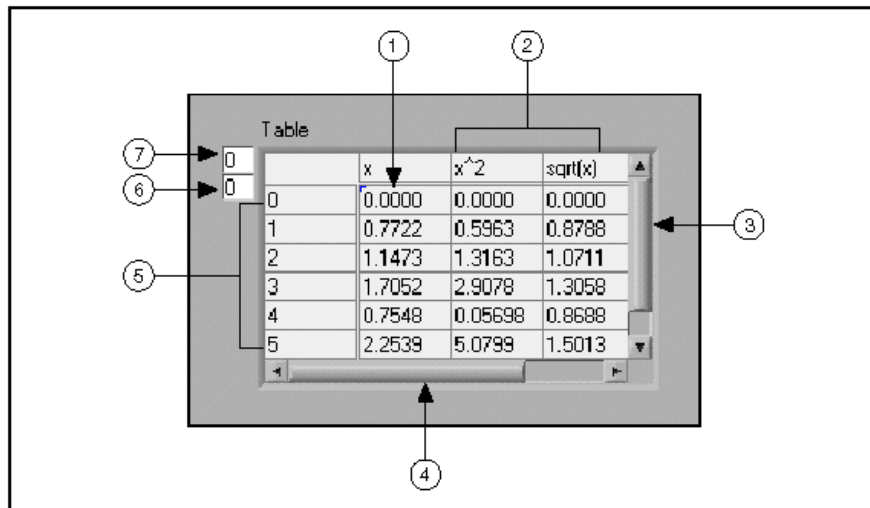
Тип отображения строкового объекта выбирается в его контекстном меню. Типы отображения строки и примеры заполнения поля ввода текста показаны в таблице ниже.

Тип отображения	Описание	Пример текста
Режим стандартного отображения (Normal Display)	Отображает стандартные ASCII коды, используя шрифт элемента управления. Управляющие коды для печати выводятся на экран в виде квадратов.	There are four display types. \ is a backslash

Режим отображения с обратным слэшем непечатаемых управляющих кодов ('\ Codes Display)	Выводит \ для всех непечатаемых управляющих кодов	There\sare\sfour\sdisplay\stypes.\n\\\sis\sa\sbackslash
Режим скрытого отображения текста (Password Display)	Выводит * для всех кодов текстового пространства	***** ***** *****
Режим отображения 16-тиричных ASCII кодов (Hex Display)	Выводит значение ASCII кода для каждого символа	5468 6572 6520 6172 6520 666F 7572 2064 6973 706C 6179 2074 7970 6573 2E0A 5C20 6973 2061 2062 6163 6B73 6C61 7368 2E

Таблицы

Элемент управления Таблица, расположенный в палитре **Controls>>List & Table** предназначен для создания таблиц на лицевой панели. Каждая ячейка находится в строке и столбце таблицы. Поэтому таблица отображает двумерный массив строк. Ниже показана таблица и ее составные части.



- | | |
|------------------------------------|--------------------------|
| 1. Ячейка таблицы | 5. Заголовок строки |
| 2. Заголовок столбца | 6. Индекс по горизонтали |
| 3. Вертикальная полоса прокрутки | 7. Индекс по вертикали |
| 4. Горизонтальная полоса прокрутки | |

Для инициализации значений ячеек таблицы используется инструмент УПРАВЛЕНИЕ или ВВОД ТЕКСТА, с помощью которых достаточно ввести текст в выделенную ячейку.

Таблица – это двумерный массив строк. Таким образом, для использования таблицы в качестве элемента отображения данных, необходимо двумерный массив чисел преобразовать в двумерный массив строк. Заголовки строк и столбцов таблицы, как в таблице символов, автоматически не отображаются. Необходимо создать одномерный массив строк, содержащий заголовки строк и столбцов таблицы.

В. Функции работы со строками

Для редактирования и управления строками на блок-диаграмме следует пользоваться функциями обработки строк, расположенными в палитре **Functions>>String**. Некоторые из функций работы со строками рассмотрены ниже:

String Length - выдает количество символов в строке, включая пробелы. Например, функция **String Length** выдает значение 19 для приведенного ниже текста: *The quick brown fox*

Concatenate Strings - объединяет строки и одномерные массивы строк в отдельную строку. Для увеличения полей ввода данных функции следует изменить ее размер. Например, объединив предыдущую строку со следующим массивом строк,

jumped	over	the	lazy	dog
--------	------	-----	------	-----

функция **Concatenate Strings** на выходе выдает следующую строку: *The quick brown fox jumped over the lazy dog*

String Subset - выдает подстроку определенной длины **length**, начиная со значения **offset** (смещение). Смещение первого элемента в строке равно 0. Например, если на поле ввода данных функции подать предыдущую строку, то функция **String Subset** при **offset**=4 и **length**=5 выдаст значение: *quick*.

Match Pattern - ищет повторяющуюся последовательность, поданную на поле ввода данных **regular expression**, в строке начиная со значения смещения **offset**, и, если находит соответствие, разбивает строку на три подстроки. Если соответствие не найдено, поле вывода данных **match substring** является пустым, а значение поля вывода данных **offset past match** (смещение повторяющейся последовательности в строке) равно -1. Например, на поле **regular expression** (шаблон подстроки) подается значение **:**, а строка на входе *VOLTS DC: +1.22863E+1*.

Функция **Match Pattern** выдаст величины **before substring** (перед подстрокой) *VOLTS DC*, **match**

substring (шаблон подстроки) : и **after substring** (после подстроки) $+1.22863E+1$, а также **offset past match** равный 9.

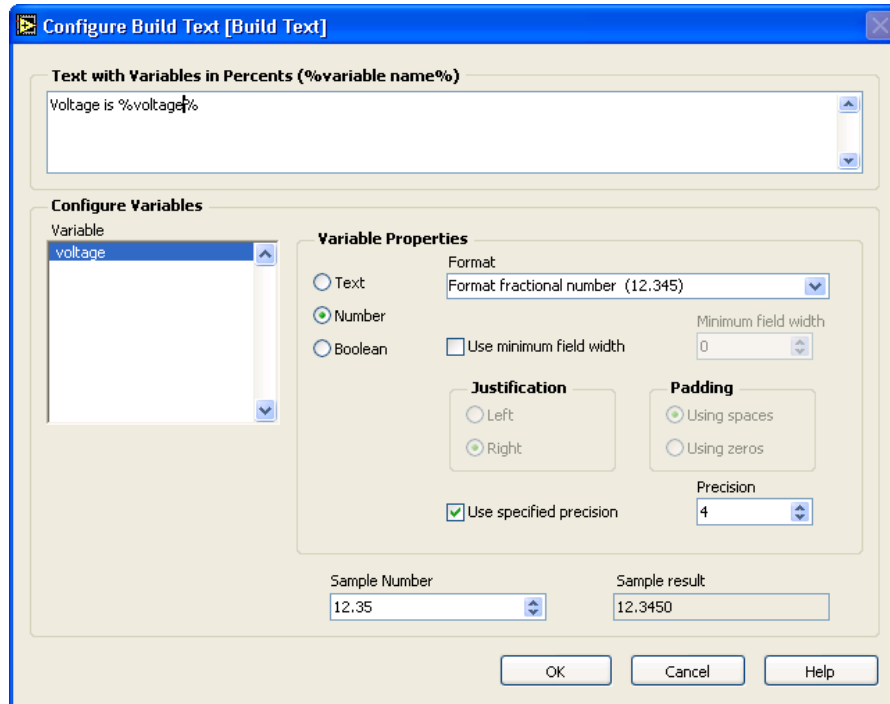
Преобразование числовых данных в строку

Для преобразования числовых данных в строковые используются ВП **Build Text Express** и функция **Format Into String** (конвертирование в строку). Обе эти функции имеют входные и выходные кластеры ошибок. При недостатке места на блок-диаграмме лучше использовать функцию **Format Into String**.

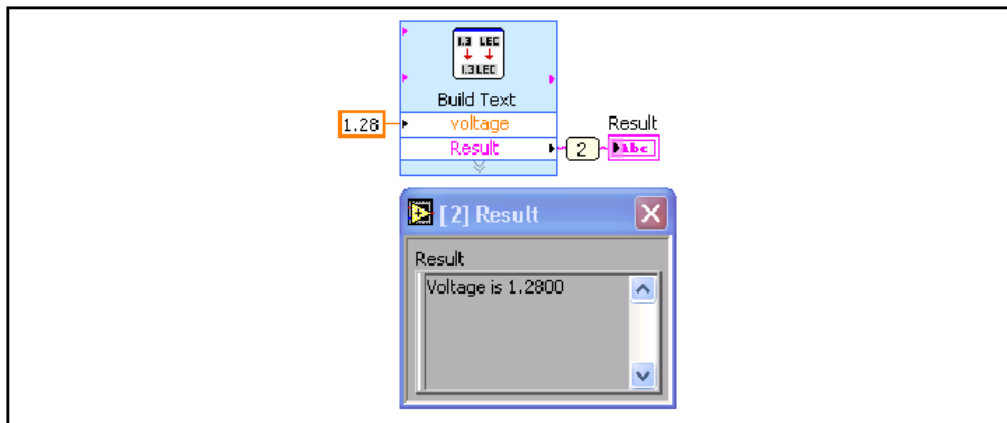
Экспресс-ВП Build Text Express VI

Экспресс-ВП **Build Text**, расположенный в палитре **Functions>>Express>>Output** производит объединение входных строк. Если входные величины имеют не строковый тип данных, то они преобразуются в строку в соответствии с настройками этого экспресс-ВП.

При помещении Экспресс-ВП **Build Text** на блок-диаграмму появляется диалоговое окно настроек **Configure Build Text**. В следующем примере значение напряжения подается на вход экспресс-ВП и преобразуется к формату данных с плавающей запятой с 4-мя числами после запятой. Затем это значение добавляется к концу строки **Voltage is** (Напряжение равно).



При такой настройке экспресс-ВП выглядит следующим образом. Для наблюдения за выходной строкой используется отладочный индикатор. Любые значения, подаваемые на поле ввода данных **Beginning Text**, будут присоединяться к началу текста из диалогового окна настроек.



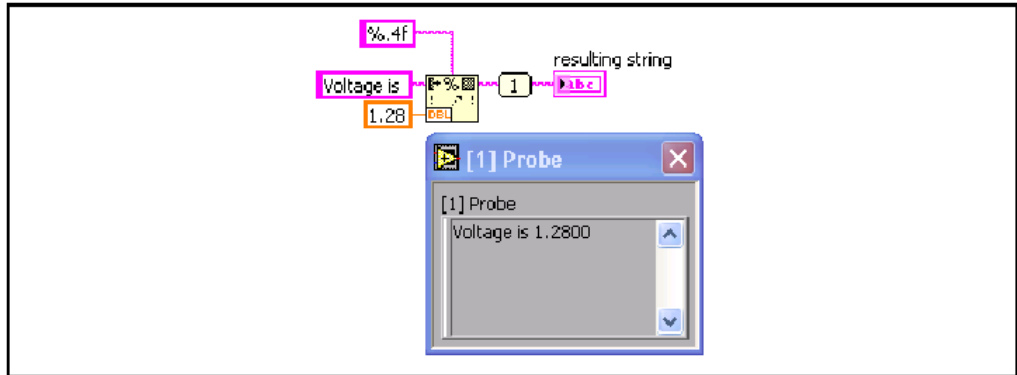
Совет Операция объединения строк называется конкатенацией, т.е. в данном случае при конкатенации две строки сливаются в одну.

Функция **Format Into String**

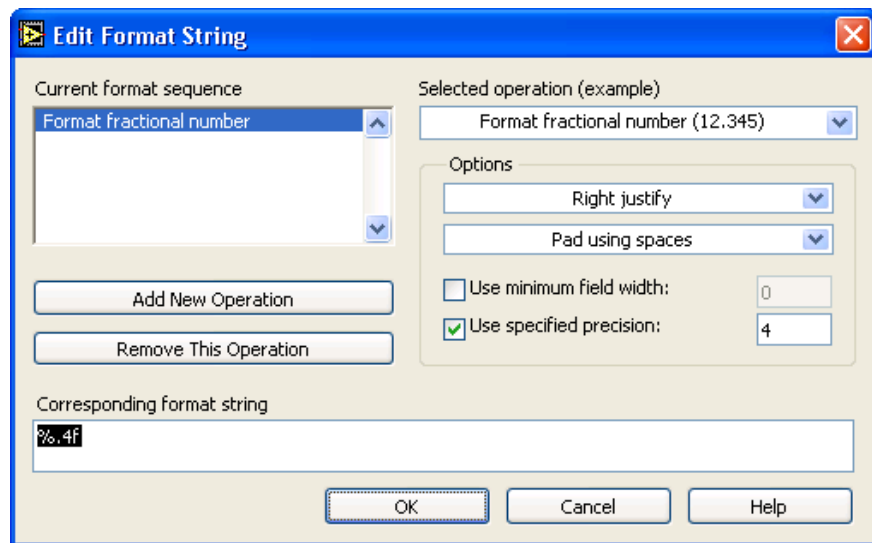
Функция **Format Into String** преобразует параметры любого формата, такие как числовые данные, в строку. Для

увеличения количества параметров следует изменить размер функции.

В приведенном ниже примере функция **Format Into String** выдает указанную на отладочном индикаторе строку при значениях полей **format string** (формате строки) `%.4f`, **input string** (входной строке) *Voltage is* (учитывая пробел в конце), и параметре 1.28.



В формате строки `%` - указывает начало формата строки, `.` - число после точки определяет точность представления числа, `4` - показывает количество знаков после запятой, а `f` - указывает тип данных с плавающей запятой. Для создания и редактирования формата строки следует щелкнуть правой кнопкой мыши по функции и выбрать пункт контекстного меню **Edit Format String**. Рисунок ниже показывает вид диалогового окна **Edit Format String** из предыдущего примера.



Для получения более подробной информации о синтаксисе форматов, следует обратиться к встроенной в LabVIEW справочной информации (**LabVIEW Help**).

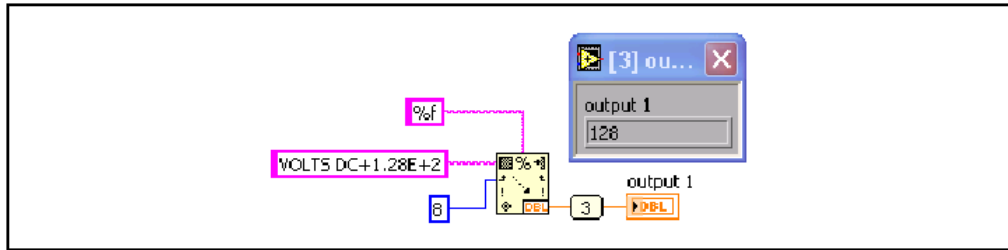
Преобразование строк в числовые данные

Для преобразования строки в числовые данные следует использовать функцию **Scan From String**.

Просмотр и конвертирование строки

Функция **Scan From String** преобразует строку, содержащую допустимые числовые символы, такие как 0-9, +, -, e, E и разделитель ., в данные числового формата. Функция начинает просмотр строки, подаваемой на поле ввода данных **input string** с номера символа, задаваемого на поле **initial search location**. Функция может просматривать входящую строку различных типов данных, таких как числовые или логические данные, основываясь на формате строки. Для увеличения количества полей вывода данных следует изменить размер функции.

Например, при значениях на полях ввода данных **format string** - %f, **initial search location** - 8, **input string** - *VOLTS DC+1.28E+2* функция выдает результат 128.00, как показано ниже.



В формате строки % - указывает начало формата строки, f - указывает тип данных с плавающей запятой. Для создания и редактирования формата строки следует щелкнуть правой кнопкой мыши по функции и выбрать пункт контекстного меню **Edit Scan String**.

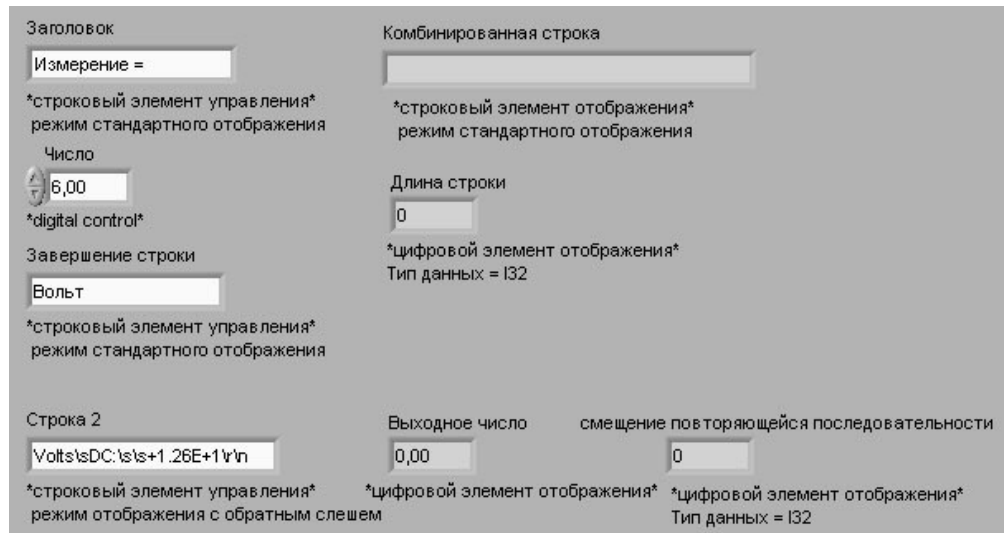
Упражнение 11-1 ВП Компоновка строки

Цель: Приобретение навыков по использованию функций обработки строк.

Ниже приведена последовательность действий для создания ВП, который преобразует числовые данные в строку и объединяет строку с другими строками в одну. Затем после поиска по шаблону полученная часть строки переводится в числовой формат.

Лицевая панель

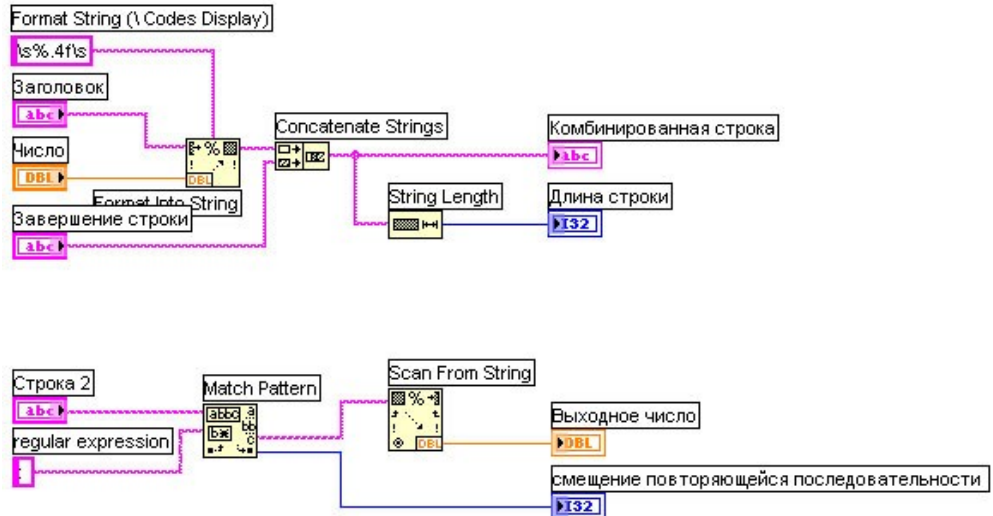
1. Откройте новый ВП и оформите лицевую панель, как показано ниже на рисунке. Воспроизводить комментарии и подписи к элементам не обязательно.



- a. Щелкните правой кнопкой мыши по элементу **Строка 2** и выберите из контекстного меню режим отображения **'\ Codes Display**.
- b. Для элементов **Длина строки** и **Смещение повторяющейся последовательности** установите тип представления данных **I32**.

Блок-диаграмма

2. Постройте блок-диаграмму, как показано ниже:



Выберите функцию **Format Into String**, расположенную в палитре **Functions>>String**. Эта функция преобразует число в строку.

- h. Щелкните правой кнопкой мыши по функции **Format Into String** и выберите пункт **Edit Format String** для вызова соответствующего диалогового окна.
- i. Выделите опцию **Use specified precision** и в поле ввода текста введите значение 4 для преобразования элемента **Число** в строку с четырьмя знаками после запятой.
- j. Нажмите на кнопку **OK**. LabVIEW создаст формат строки `%4f`, используя выбранную опцию.
- k. С помощью инструмента **ВВОД ТЕКСТА**, введите пробел с обеих сторон `%4f` и нажмите клавиши **<Shift+Enter>**. Таким образом, на элементе **Комбинированная строка** числовые данные появятся с пробелами с обеих сторон.
- l. Щелкните правой кнопкой мыши по константе и выберите режим отображения **'\ Codes Display** из контекстного меню. Введенные пробелы заменятся на `\`.



Выберите функцию **Concatenate Strings**, расположенную в палитре **Functions>>String**. Эта функция объединит входящие в нее строки в одну.



Выберите функцию **String Length**, расположенную в палитре **Functions>>String**. Эта функция выдаст

значение количества символов в объединенной строке
Комбинированная строка.



Выберите функцию **Match Pattern**, расположенную в палитре **Functions>>String**. Эта функция осуществляет поиск в элементе **Строка 2** по шаблону : (двоеточие).

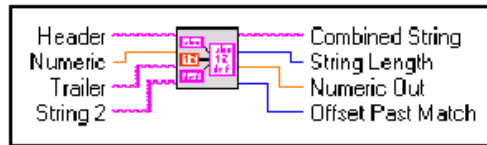
м.Щелкните правой кнопкой мыши по полю **regular expression** и выберите пункт контекстного меню **Create>>Constant**, введите двоеточие и нажмите на клавиши **<Shift+Enter>**.



Выберите функцию **Scan from String**, расположенную в палитре **Functions>>String**. Эта функция преобразует строку после двоеточия в числовые данные.

Иконка ВП и соединительная панель

3. Перейдите на лицевую панель и создайте иконку и соединительную панель для использования созданного ВП в качестве подпрограммы в других ВП. Подробная информация о методах оформления иконки и соединительной панели находится в Уроке 3, *Подпрограммы ВП*.



4. Сохраните ВП под именем *Компоновка строки.vi*. Этот ВП будет использоваться позднее.

Запуск ВП

5. Измените значение элементов на лицевой панели и запустите ВП.

ВП объединит элементы: **Заголовок**, **Число** и **Завершение строки** в строку **Комбинированная строка** и выдаст значение длины строки.

ВП также найдет месторасположение подстроки : в элементе **Строка 2**. При выполнении ВП преобразует строку после двоеточия в число **Выходное число** и

выводит на экран индекс первого элемента после двоеточия в элемент **Смещение повторяющейся последовательности**.

6. Сохраните и закройте ВП.

Конец упражнения 11-1

С. Функции файлового ввода/вывода

Функции файлового ввода/вывода производят файловые операции записи и считывания данных. Функции файлового ввода/вывода расположены в палитре **Functions>>File I/O** и предназначены для:

Открытия и закрытия файла данных.

Считывания и записи данных из/в файл(а).

Считывания и записи данных из/в файл(а) в виде таблицы символов.

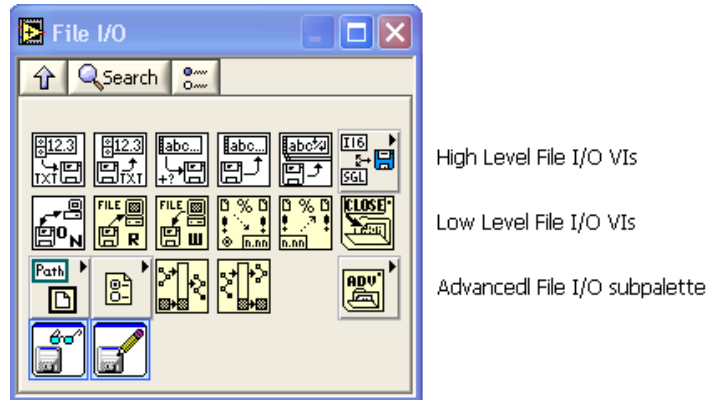
Перемещения и переименования файлов и каталогов.

Изменения характеристик файла.

Создания, изменения и считывания файлов конфигурации.

Функции файлового ввода/вывода

Палитра функций файлового ввода вывода, показанная ниже, разделена на три части: функции высокого уровня (**high level File I/O**), функции низкого уровня (**low level File I/O**) и подпалитра функций расширенных возможностей (**advanced File I/O**).



Функции файлового ввода/вывода высокого уровня

Функции файлового ввода/вывода высокого уровня расположены в верхней строке палитры **Functions>>File I/O**. Они предназначены для выполнения основных операций по вводу/выводу данных. Более подробную информацию

можно получить в разделе Е *Использование файлового ввода/вывода высокого уровня*.

Использование функций файлового ввода/вывода высокого уровня позволяет сократить время и усилия программистов при записи и считывании данных в/из файл(а). Функции файлового ввода/вывода высокого уровня выполняют запись и считывание данных и операции закрытия и открытия файла. При наличии ошибок функции файлового ввода/вывода высокого уровня отображают диалоговое окно с описанием ошибок и предлагают на выбор: продолжить выполнение программы или остановить ее. Однако из-за того, что функции данного класса объединяют весь процесс работы с файлами в один ВП, переделать их под определенную задачу бывает трудно. Для специфических задач следует использовать функции файлового ввода/вывода низкого уровня.

Функции файлового ввода/вывода низкого уровня

Функции файлового ввода/вывода низкого уровня расположены в средней строке палитры **Functions>>File I/O**. Дополнительные функции работы с файлами (**Advanced File I/O**) расположены в палитре **Functions>>File I/O>>Advanced File Functions** и предназначены для управления отдельными операциями над файлами.

Функции файлового ввода/вывода низкого уровня используются для создания нового или обращения к ранее созданному файлу, записи и считывания данных и закрытия файла. Функции низкого уровня работы с файлами поддерживают все операции, необходимые при работе с файлами. Информация о дополнительных возможностях функций работы с файлами изложена в учебном курсе LabVIEW Основы II.

Основы файлового ввода/вывода

Стандартные операции ввода/вывода данных в/из файла состоят из следующей последовательности действий:

1. Создание или открытие файла. Указание месторасположения существующего файла или пути для создания нового файла с помощью диалогового окна LabVIEW. После открытия файл LabVIEW создает ссылку на него. Более подробную информацию о ссылке на файл

можно найти в разделе *Сохранение данных в новом или уже существующем файле*.

2. Произведение операций считывания или записи данных в/из файл(а).
3. Закрытие файла.
4. Обработка ошибок.

Для осуществления основных операций файлового ввода/вывода используются следующие ВП и функции:



Open/Create/Replace File - открывает, перезаписывает существующий файл, или создает новый. Если **file path** (путь размещения файла) не указан, ВП выводит на экран диалоговое окно, в котором можно создать новый или выбрать уже существующий файл.



Read File - считывает данные из файла, определяемого по ссылке **refnum**, и выдает данные на поле вывода **data**, на поле **count** подается значение количества считываемых данных. Считывание данных начинается с места, определяемого элементами **pos mode** и **pos offset**, и зависит от формата файла.



Write File - записывает данные в файл, определяемый по ссылке **refnum**. Запись начинается с места, определяемого полями ввода данных **pos mode** и **pos offset** для файла потока байтовых данных, и указателем конца файла для файла протоколированных данных.



Close File - закрывает указанный в ссылке **refnum** 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.

Во время работы ВП LabVIEW проверяет наличие ошибок в каждом узле. Если LabVIEW не находит ошибок, то узел выполняется нормально. Если LabVIEW обнаруживает ошибку в одном узле, то его выполнение прерывается, а

информация об ошибке передается следующему узлу. Следующий узел поступает так же, и в конце выполнения LabVIEW сообщает об ошибках.

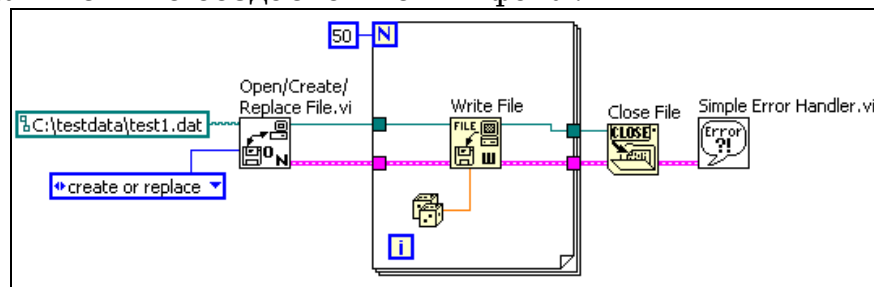
Сохранение данных в новом или уже существующем файле

В файл, созданный (или открытый) с помощью функций файлового ввода/вывода, можно записать данные любого типа. При необходимости доступа к файлу со стороны других приложений или пользователей, следует записывать данные в виде строки ASCII символов. Более полную информацию о записи бинарных файлов и файлов регистрации данных можно найти в учебном курсе LabVIEW Основы II.

Доступ к файлу можно осуществить программным путем или с использованием диалогового окна. Для доступа к файлу с помощью диалогового окна на поле ввода **file path** подпрограммы ВП **Open/Create/Replace File VI** не следует подавать данные. Программный доступ к файлу экономит время. Приведенная ниже таблица демонстрирует организацию путей к файлам.

Платформа	Путь к файлу
Windows	Состоит из имени дисковод, двоеточия, обратного слэша, разделяющего директории, и имени файла. Например, <i>c:\testdata\test1.dat</i> - путь к файлу <i>test1.dat</i> в папке <i>testdata</i>

В приведенном ниже примере показано, как записать строку данных в файл при программном указании пути и имени файла. Если файл уже существует, то он перезаписывается, если нет - то создается новый файл.



Подпрограмма ВП **Open/Create/Replace File VI** открывает файл *test1.dat*. ВП также создает ссылку на файл и кластер ошибок.



Совет Ссылка (**refnum**) является уникальным идентификатором для таких объектов как файл, прибор и сетевое соединение.

При открытии файла, устройства или сетевого соединения LabVIEW создает ссылку на объект. Все операции с открытыми объектами выполняются с использованием ссылок.

Кластер ошибок и ссылка на файл последовательно передаются от узла к узлу. Поскольку узел не может выполняться, пока не определены все его входные поля данных, эти два параметра заставляют узлы работать в определенном порядке. Подпрограмма ВП **Open/Create/Replace File VI** передает ссылку на файл и кластер ошибок функции **Write File**, которая производит запись файла на диск. Функция **Close File** закрывает файл после получения кластера ошибок и ссылки на файл из функции **Write File**.

Подпрограмма ВП **Simple Error Handler VI** проверяет наличие ошибок и выводит информацию о них в диалоговом окне. Если в одном из узлов допущена ошибка, последующие узлы не выполняются, и кластер ошибок передается в подпрограмму ВП **Simple Error Handler VI**.

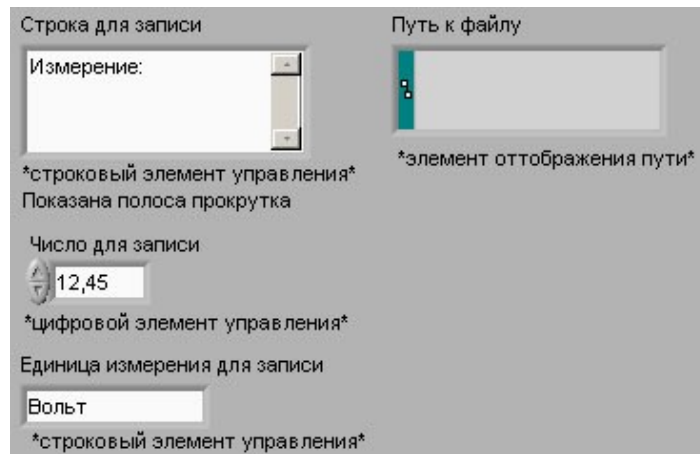
Упражнение 11-2 ВП Запись файла

Цель: Записать данные в файл.

Ниже приведена последовательность действий для создания ВП, который объединяет строку, числовые данные и модуль строки в файл. В упражнении 11-3 будет создан ВП, который считывает и отображает его содержимое.

Лицевая панель

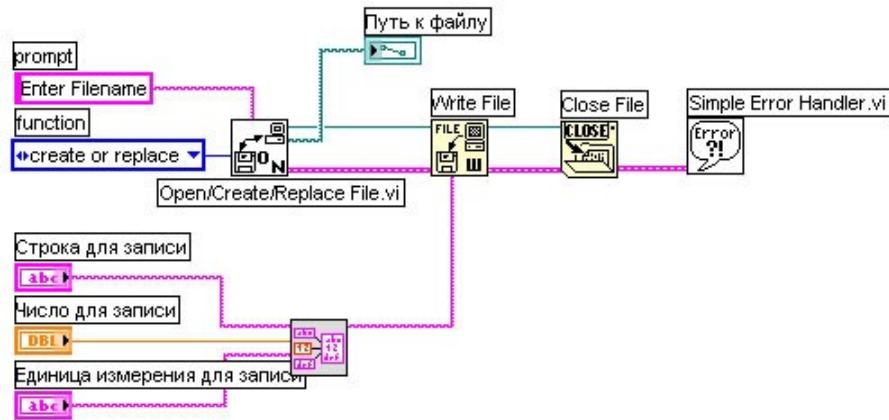
1. Откройте новый ВП и оформите лицевую панель, как показано ниже:



- a. В палитре **Controls>>String & Path** выберите элемент отображения пути. Этот элемент отобразит путь к созданному файлу данных.
- b. Щелкните правой кнопкой мыши по элементу **Строка для записи** и в контекстном меню выберите пункт **Visible Items>>Scrollbar**.

Блок-диаграмма

2. Постройте следующую блок-диаграмму:



Выберите в разделе **Functions>>Select a VI** ВП *Компоновка строки.vi*, созданный в упражнении 9-1, и поместите его на блок-диаграмму. Этот ВП объединяет три строки в одну.



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.

- a. Щелкните правой кнопкой мыши по полю **prompt** и в контекстном меню выберите пункт **Create>>Constant** для создания константы **Введите имя файла**. При запуске ВП появится окно выбора файла, которое будет называться **Введите имя файла**.
- b. Щелкните правой кнопкой мыши по входному полю **function** и в контекстном меню выберите пункт **Create>>Constant**. Для выбора пункта выпадающего меню **create or replace** следует использовать инструмент УПРАВЛЕНИЕ.



Выберите функцию **Write File**, расположенную в палитре **Functions>>File I/O**. Эта функция записывает объединенную строку в файл.



Выберите функцию **Close File**, расположенную в палитре **Functions>>File I/O**. Эта функция закрывает файл.



Выберите подпрограмму ВП **Simple Error Handler VI**, расположенную в палитре **Functions>>Time & Dialog**. Этот ВП проверяет кластер ошибок и выводит диалоговое окно при возникновении ошибки.

3. Сохраните ВП под именем *Запись файла.vi*.

Запуск ВП

4. Поменяйте значения элементов управления на лицевой панели и запустите ВП. Появится диалоговое окно **Введите имя файла**.
5. Введите в диалоговое окно название файла *демофайл.txt* и нажмите на кнопку **Save** или **OK**.
ВП запишет в файл данные из элементов **Строка для записи**, **Число для записи** и **Единица измерения для записи**.
6. Закройте ВП.

Конец упражнения 11-2

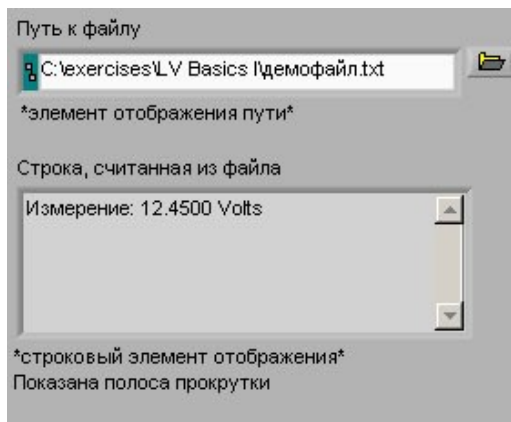
Упражнение 11-3 ВП Чтение файла

Цель: Считать данные из файла.

Ниже приведена последовательность действий для создания ВП, который читает файл, созданный в упражнении 11-2, и выводит данные в строковом элементе отображения.

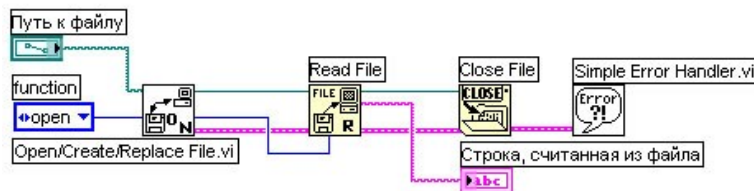
Лицевая панель

7. Откройте новый ВП и создайте лицевую панель, используя элемент управления путем к файлу и строковый элемент отображения в палитре **Controls>>String & Path**.



Блок-диаграмма

8. Постройте следующую блок-диаграмму:



В палитре **Functions>>File I/O** выберите подпрограмму **Open/Create/Replace File VI**. Этот ВП выведет на экран диалоговое окно, которое используется для создания и открытия файла.

- а. Щелкните правой кнопкой мыши по входному полю **prompt** и из контекстного меню выберите **Create>>Constant** для создания константы **Выбрать имя файла**.

- b. Щелкните правой кнопкой мыши по полю **function** и выберите в контекстном меню пункт **Create>>Constant** для создания константы. С помощью инструмента УПРАВЛЕНИЕ выберите пункт выпадающего меню **open**.



В палитре **Functions>>File I/O** выберите функцию **Read File**. Эта функция читает количество байт данных с начала файла, определяемое значением поля **count**.



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



В палитре **Functions>>Time& Dialog** выберите подпрограмму **Simple Error Handler VI**. Этот ВП проверяет кластер ошибок и, в случае появления ошибки, выводит на экран диалоговое окно.

9. Сохраните ВП под именем *Чтение файла.vi*.

Запуск ВП

10. Перейдите на лицевую панель и с помощью инструмента УПРАВЛЕНИЕ выберите кнопку **Browse** (обзор) в элементе управления **Путь к файлу**.
11. Выберите файл *демофайл.txt* и нажмите на кнопку **Open** или **OK**.
12. Запустите ВП. Строка, считанная из файла, отобразится на лицевой панели ВП.
13. При желании выполните часть упражнения повышенной сложности. В противном случае сохраните и закройте ВП.



14. Измените ВП таким образом, чтобы числовые данные распознавались в строке и отображались в цифровом элементе отображения. По окончании сохраните и закройте ВП.



С Для поиска первого числового символа **ответ** используйте функцию **Match Pattern**.

Конец упражнения 11-3

Краткое изложение пройденного материала, советы и секреты

Строки являются последовательностями ASCII кодов. Для создания окон ввода текста и меток используются строковые элементы управления и отображения данных, расположенные в палитре **Controls>>String & Path**.

С помощью полосы прокрутки можно минимизировать пространство, занимаемое строковым объектом. Для этого необходимо щелкнуть правой кнопкой мыши по объекту и выбрать в контекстном меню пункт **Visible Items>>Scrollbar**.

Для редактирования и управления строками на блок-диаграмме используются функции обработки строк, расположенные в палитре **Functions>>String**.

Для преобразования числовых данных в строки используются функция **Format Into String** и экспресс-ВП **Build Text Express VI**.

Для преобразования строк в числовые данные используется функция **Scan From String**.

Для создания или редактирования формата строки необходимо щелкнуть правой кнопкой мыши по функции **Format Into String** или **Scan From String** и выбрать пункт контекстного меню **Edit Format String** или **Edit Scan String**.

Для проведения операций над файлами используются функции файлового ввода/вывода, расположенные в палитре **Functions>>File I/O**.

Функции файлового ввода-вывода низкого уровня расположены в средней строке палитры **Functions>>File I/O**. Дополнительные функции работы с файлами расположены в палитре **Functions>>File I/O>>Advanced File Functions** и предназначены для управления отдельными операциями над файлами.

При осуществлении записи в файл производятся следующие операции: открытие файла, создание или перезапись файла, запись данных и закрытие файла.

Аналогичные операции производятся и при чтении файла: открытие существующего файла, считывание данных и закрытие файлов.

Для получения доступа к файлу с помощью диалогового окна поле **file path** подпрограммы ВП **Open/Create/Replace File VI** следует оставить неподключенным.

Занятие 12

Файловый ввод/вывод высокого уровня

Подпрограммы ВП работы с файлами обеспечивают ввод/вывод данных в/из файл(а).

Изложены вопросы:

- А. Форматирование текстовых файлов для использования в таблице символов.
- В. Использование файлового ввода/вывода высокого уровня.

А. Форматирование строк таблицы СИМВОЛОВ

Для того чтобы записать данные в файл формата электронной таблицы, необходимо переформатировать строковые данные в строку таблицы, содержащую разделители, такие как символ табуляции. Во многих приложениях символ табуляции разделяет столбцы, а символ **end of line** (конец строки) разделяет строки.

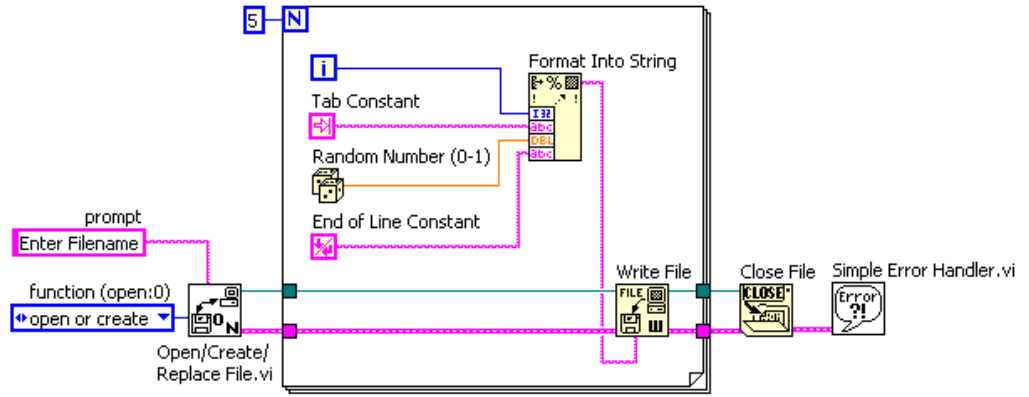


Примечание Для обеспечения совместимости между различными платформами следует использовать константу **end of line constant**, расположенную в палитре **Functions>>String. (Windows)** Константа осуществляет перевод каретки и перевод строки. **(MacOS)** Константа осуществляет перевод каретки. **(UNIX)** Константа осуществляет перевод строки.

Функция **Format Into File** предназначена для форматирования строк, путей к файлам, числовых и логических данных в текст, а также для записи текста в файл. Часто эта функция используется вместо двух операций – форматирования строки с помощью функции **Format Into String** или ВП **Build Text Express VI** и записи результата с помощью функций **Write Characters To File** или **Write File**.

Функция **Format Into File** предназначена для определения порядка, в котором данные записываются в тестовый файл. Однако ее нельзя применять для добавления данных в файл или перезаписи существующего файла. Для этих операций используется функция **Format Into String** совместно с функцией **Write File**. Путь к файлу или ссылку на него можно подать на поле **input file** или оставить это поле без соединения, чтобы указать имя файла в диалоговом окне.

Ниже представлена блок-диаграмма, на которой подпрограмма ВП **Open/Create/Replace File VI** открывает файл. Цикл **For** выполняется пять раз. Функция **Format Into String** преобразует значения счетчика итераций и случайное число в строку. Также указываются символы **Tab constant** (табуляции) и **End of Line Constant** (конца строки) для создания двух столбцов и одной строки таблицы символов. По окончании пяти итераций цикла файл закрывается и ВП проверяет наличие ошибок.



Этот ВП создает следующий текстовый файл, в котором стрелка (→) указывает символ табуляции, а символ параграфа (¶) указывает конец строки:

0→0.798141¶
1→0.659364¶
2→0.581409¶
3→0.526433¶
4→0.171062¶

Можно открыть данный текстовый файл в любом редакторе электронных таблиц для отображения на экране следующей таблицы.

	A	B
1	0	0.798141
2	1	0.659364
3	2	0.581409
4	3	0.526433
5	4	0.171062

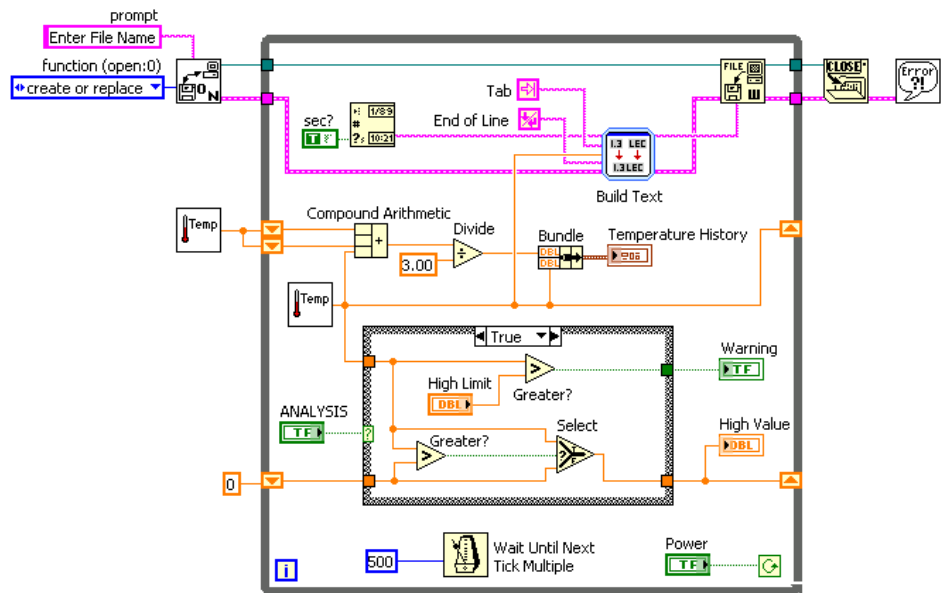
Упражнение 12-1 ВП Регистратор температуры

Цель: Сохранить данные в файл в форме, доступной для редактора электронных таблиц или текстового редактора.

Выполните следующие шаги для сохранения времени и текущего значения температуры в файл.

Блок-диаграмма

1. Откройте ВП *Контроль температуры.vi*, созданный в упражнении 8-2, и сохраните его под именем *Регистратор температуры.vi*. Изменять лицевую панель нет необходимости.
2. Измените блок-диаграмму, как показано ниже:



В палитре **Functions>>File I/O** выберите **Open/Create/Replace File VI**. Этот ВП выводит диалоговое окно для открытия или создания файла.



В палитре **Functions>>Time & Dialog** выберите функцию **Get Date/Time String**. Эта функция выдает время проведения измерения температуры в строковом формате.

Щелкните правой кнопкой мыши по терминалу **want seconds?** (нужны секунды?) и в контекстном меню

выберите **Create>>Constant**. С помощью инструмента УПРАВЛЕНИЕ измените значение логической константы с FALSE на значение TRUE. Это значение константы заставит функцию включить секунды в строку.



Поместите на блок-диаграмму экспресс-ВП **Build Text Express VI**, расположенный в палитре **Functions>>Express>>Output**. Этот экспресс-ВП преобразует несколько входных строк в одну.

- a. При помещении экспресс-ВП **Build Text Express VI** на блок-диаграмму появляется диалоговое окно. Для определения трех переменных введите строку **%tab%temp%end%** в поле **Text with Variables in Percents**. Первая переменная - константа табуляции, вторая - значение температуры, последняя - константа конца строки. Для времени создавать переменную не нужно, оно подается на поле ввода **Beginning Text**.
- b. Выберите переменную **temp** в разделе **Configure variables**. Отметьте пункты **Number** и формат дробного числа **Format fractional number**. Переменные **tab** и **end** настраивать не нужно, оставьте их в состоянии по умолчанию.
- c. После выполнения настроек нажмите на кнопку **OK**.
- d. Для экономии места на блок-диаграмме щелкните правой кнопкой мыши по ВП **Build Text Express VI** и выберите пункт **View as Icon**.



В палитре **Functions>>String** выберите константу **Tab constant** и константу **End of Line constant**.



Поместите на блок-диаграмму функцию **Write File** из палитры **Functions>>File I/O**. Эта функция записывает данные в файл, указанный в ссылке **refnum**.



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



В палитре **Functions>>Time & Dialog** выберите подпрограмму ВП **Simple Error Handler VI**. Эта подпрограмма проверяет кластер ошибок ВП и при наличии ошибок выводит на экран диалоговое окно.

3. Сохраните ВП, он будет использоваться позднее.

Запуск ВП

4. Перейдите на лицевую панель и запустите ВП. Появится диалоговое окно **Enter Filename**.

5. В диалоговом окне введите имя файла *temp.txt* и нажмите на кнопку **Save** или **OK**.

ВП создаст файл с именем *temp.txt*. ВП считывает данные каждые полсекунды и сохраняет значения измеренной температуры и время измерения в файл, пока не нажата кнопка **Power**. Когда ВП заканчивает работу, файл закрывается.

6. Откройте текстовый редактор, например, (**Windows**) Notepad или WordPad, (**MacOS**) SimpleText или (**UNIX**) Text Editor.

7. Откройте файл *temp.txt*. Время измерения отображено в первом столбце, а данные температуры – во втором.

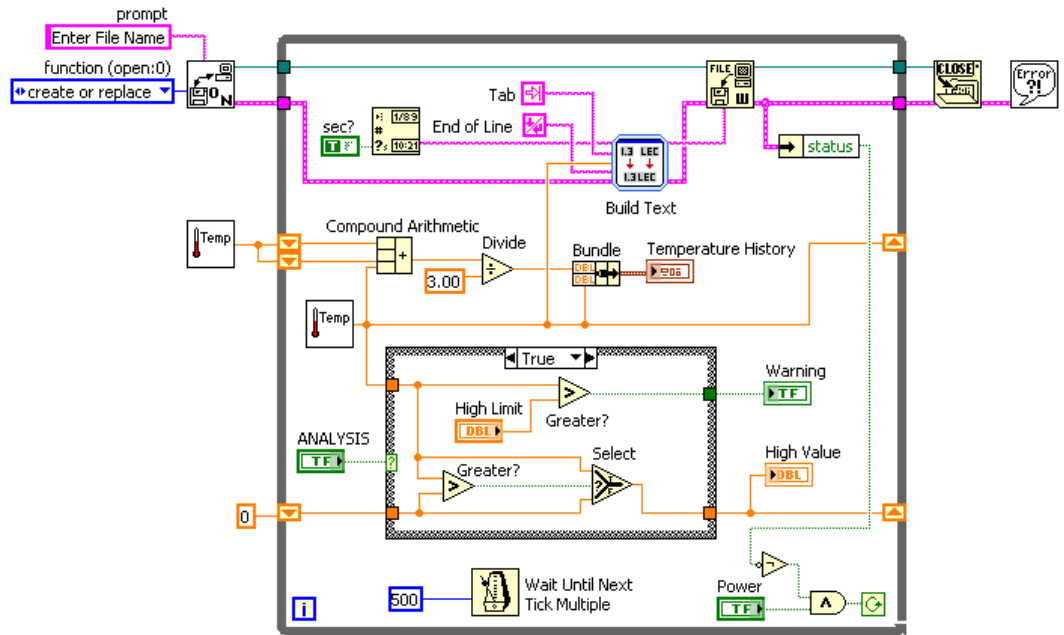
8. Закройте текстовый редактор и вернитесь в LabVIEW.

9. При желании выполните дополнительную часть упражнения. В противном случае закройте ВП.

Дополнительная часть

При использовании системы обработки ошибок в ВП, как только возникает ошибка, цикл **While** должен остановиться. Прodelайте следующие шаги для изменения ВП так, чтобы он прекращал работу не только от кнопки **Power**, но и в случае возникновения ошибки.

10. Измените блок-диаграмму, как показано на рисунке.



В палитре **Functions>>Cluster** выберите функцию **Unbundle by Name**. Эта функция выделяет параметр **status** из кластера ошибок.



В палитре **Functions>>Boolean** выберите функции **Not** и **And**. Функции **Not** и **And** формируют условие выхода из цикла. Пока элемент **Power** имеет значение TRUE, и ошибки отсутствуют, ВП продолжает выполняться.

11. Сохраните и запустите ВП.
12. Проверьте обработку ошибок, удалив проводник ссылки на файл между функцией **Write File** и левой границей цикла. Щелкните правой кнопкой мыши по полю ввода **refnum** и выберите пункт **Create>>Constant**.
13. Снова запустите ВП. ВП должен запросить путь к файлу, а затем остановиться из-за ошибки. Если бы обработка ошибок не была включена в этот ВП, то сообщение об ошибке появилось бы только после остановки ВП пользователем.
14. При желании выполните часть упражнения повышенной сложности. В противном случае закройте ВП, не сохраняя изменений.



15. Замените ВП **Build Text Express VI** функцией **Format Into String**.

16. Запустите ВП.

17. Закройте ВП, не сохраняя изменений.

Конец упражнения 12-1

В. Функции файлового ввода/вывода высокого уровня

Функции файлового ввода/вывода высокого уровня расположены в верхней строке палитры **Functions>>File I/O**. Они предназначены для выполнения действий файлового ввода или вывода данных следующих типов:

Символов в/из текстовых файлов.

Строк из текстовых файлов.

Одномерных или двумерных массивов числовых данных одинарной точности в/из файла электронной таблицы.

Одномерных или двумерных массивов числовых данных одинарной точности или целочисленных 16-разрядных в/из бинарного файла.

Функции файлового ввода/вывода высокого уровня включают в себя:

Write to Spreadsheet File - преобразует 2D или 1D массив числовых данных одинарной точности в текстовую строку и записывает строку в новый или добавляет в уже существующий файл. При этом можно также транспонировать данные. ВП открывает или создает файл перед записью и после всех операций закрывает его. Этот ВП используется для создания текстовых файлов, читаемых большинством текстовых редакторов и редакторов электронных таблиц.

Read From Spreadsheet File - считывает определенное число строк от начального смещения **start of read offset** и преобразует данные в 2D массив числовых данных одинарной точности. ВП открывает файл перед чтением и после всех операций закрывает его. Этот ВП можно использовать для чтения таблицы символов, сохраненной в текстовом формате.

Write Characters to File - записывает строку символов в новый файл или добавляет ее в уже существующий. ВП открывает или создает файл перед записью и после всех операций закрывает его.

Read Characters From File - считывает количество символов **number of characters** от начального смещения

start of read offset. ВП открывает файл перед чтением и после всех операций закрывает его.

Read Lines From File - считывает определенное число строк из текстового или бинарного файла с положения **start of read offset**. ВП открывает файл перед чтением и закрывает его после.

Binary File - читает и записывает файл в бинарном формате. Данные могут быть целочисленного типа или числовыми данными одинарной точности с плавающей точкой.

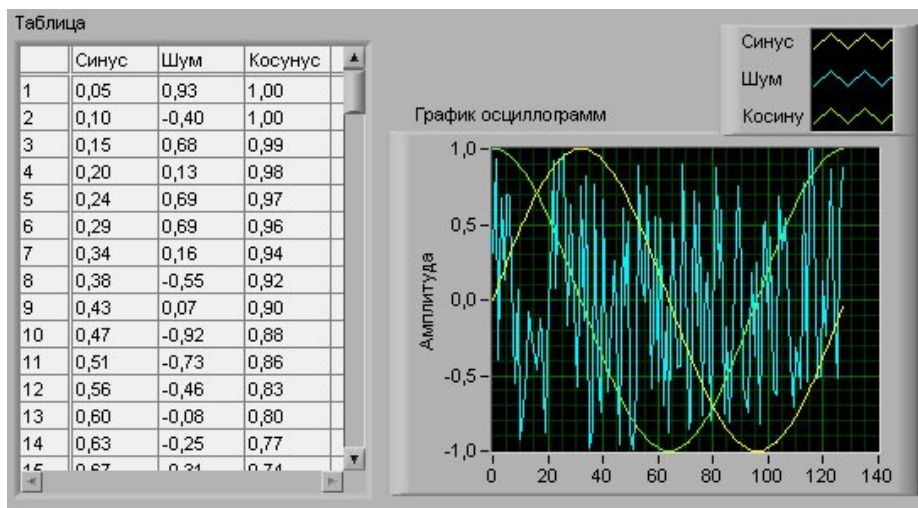
Упражнение 12-2 ВП Запись таблицы СИМВОЛОВ

Цель: Сохранить 2D массив в текстовый файл в виде таблицы так, чтобы он был доступен редактору электронной таблицы.

В упражнении 12-1 строка была отформатирована таким образом, что позиция табулятора разделял столбцы, а символ конца строки разделял строки. Выполните следующие шаги для изучения ВП, который сохраняет числовой массив в файл в формате, доступном для редактора электронных таблиц.

Лицевая панель

1. Откройте ВП *Запись таблицы символов.vi* из папки *C:\Exercises\LV Basics I*. Лицевая панель уже создана.



Запуск ВП

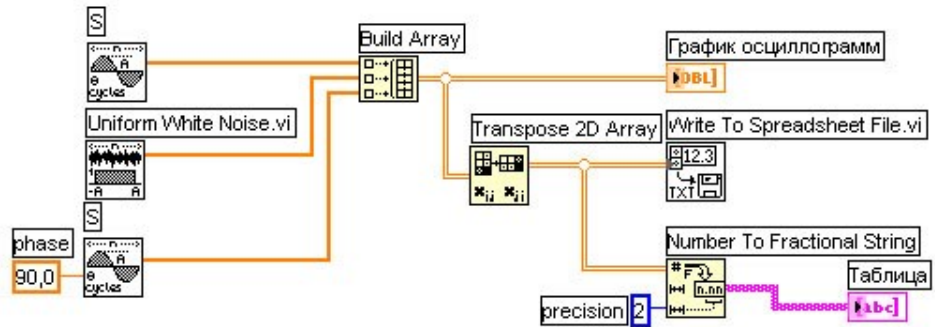
2. Запустите ВП.

ВП генерирует 2D массив из 128 строк и 3 столбцов. Первый столбец содержит данные осциллограммы синуса, второй – осциллограммы шума, третий столбец – данные осциллограммы косинуса. ВП выводит осциллограммы каждого столбца и отображает данные в виде таблицы. После вывода данных на экран появляется диалоговое окно.

3. Введите имя файла *осциллограмма.txt* и нажмите на кнопку **Save** или **OK**. Позднее этот файл будет использоваться.

Блок-диаграмма

4. Перейдите на блок-диаграмму:



ВП **Sine Pattern VI**, расположенный в палитре **Functions>>Analyze>>Signal Processing>>Signal Generation** выдает числовой массив из 128 элементов, содержащий значения синусоидального сигнала. Константа 90,0 определяет фазу гармонического сигнала.



ВП **Uniform White Noise VI**, расположенный в палитре **Functions>>Analyze>>Signal Processing>>Signal Generation** выдает числовой массив из 128 элементов, содержащий значения шумового сигнала.



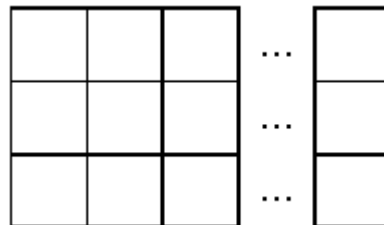
Функция **Build Array**, расположенная в палитре **Functions>>Array**, компонует 2D массив из массивов синуса, шума и косинуса.

Массив синуса

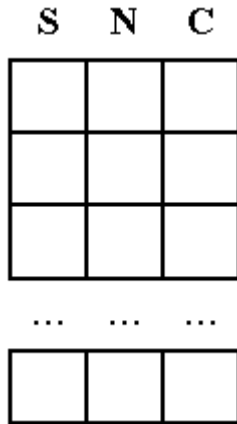
Массив шума

Массив

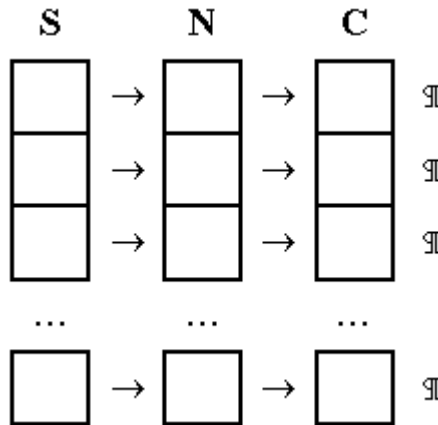
косинуса



Функция **Transpose 2D Array**, расположенная в палитре **Functions>>Array**, перестраивает элементы 2D массива так, что элемент $[i,j]$ становится элементом $[j,i]$, как показано ниже:



ВП **Write To Spreadsheet File VI**, расположенный в палитре **Functions>>File I/O**, форматирует 2D массив в строку таблицы символов и записывает строку в файл. Формат строки показан на рисунке, где стрелка → показывает позицию табулятора, а символ ¶ показывает окончание строки символов.



Функция **Number To Fractional String**, расположенная в палитре **Functions>>String>>String/Number Conversion**, преобразует массив чисел в массив строк, которые отображаются в таблице.

5. Закройте ВП.



Примечание В этом примере сохраняются в файл только три массива. Для добавления большего количества массивов следует увеличить количество полей ввода данных функции **Build Array**.

Дополнительная часть

Откройте файл текстовым редактором или редактором электронных таблиц и просмотрите его содержимое.

6. Откройте текстовый редактор, такой как в **(Windows)** Notepad или WordPad, **(MacOS)** SimpleText, **(UNIX)** Text Editor.
7. Откройте файл *осциллограмма.txt*. Данные синуса представлены в первом столбце, шум – во втором, косинус – в третьем.
8. Выйдите из текстового редактора или редактора электронных таблиц и вернитесь в LabVIEW.

Конец упражнения 12-2

Краткое изложение пройденного материала, советы и секреты

Для проведения операций над файлами используются функции файлового ввода/вывода, расположенные в палитре **Functions>>File I/O**.

Функции файлового ввода/вывода высокого уровня расположены в верхней строке палитры **Functions>>File I/O**. Они предназначены для выполнения стандартных действий по файловому вводу/выводу данных.

Функции файлового ввода-вывода низкого уровня расположены в средней строке палитры **Functions>>File I/O**. Дополнительные функции работы с файлами расположены в палитре **Functions>>File I/O>>Advanced File Functions** и предназначены для управления отдельными операциями над файлами.

При осуществлении записи в файл производятся следующие операции: открытие файла, создание или перезапись файла, запись данных и закрытие файла. Аналогичные операции производятся и при чтении файла: открытие существующего файла, считывание данных и закрытие файлов.

Для получения доступа к файлу с помощью диалогового окна поле **file path** подпрограммы ВП **Open/Create/Replace File VI** следует оставить неподключенным.

Чтобы записать таблицу символов в файл, следует отформатировать строку в строку таблицы символов, которая содержит разделители, такие как табуляция. Для форматирования строки, числовых, логических данных и путей в текстовый файл используется функция **Format Into File**.

Дополнительные упражнения



- 12-4. Постройте ВП, который генерирует 2D массив, состоящий из трех строк и 100 столбцов случайных чисел, помещает данные в таблицу символов и записывает данные таблицы в файл. Добавьте заголовок каждому столбцу. Используйте функции файлового ввода/вывода высокого уровня, расположенные в верхней строке палитры **Functions>>File I/O**.



Совет

Для записи заголовка и числовых данных в тот же файл следует использовать подпрограммы ВП **Write Characters To File VI** и **Write To Spreadsheet File VI** соответственно.

Сохраните ВП под именем *Запись осциллограмм в таблицу.vi*.

- 12-5. Постройте ВП, который преобразует строки таблицы символов, разграниченные позицией табуляции, в строки с разделителями - запятыми. Причем столбцы должны быть разделены запятыми, а строки - символами **end of line**. Строки таблицы символов, разграниченные позицией табуляции, и строки с разделителями-запятыми необходимо отобразить на лицевой панели.



Совет

Следует использовать функцию **Search and Replace String**.

Сохраните ВП под именем *Преобразование таблицы.vi*.

- 12-6. Измените **ВП Регистратор температуры**, созданный в упражнении 12-1, так, чтобы ВП не создавал файл каждый раз при запуске. Добавьте данные в конец уже существующего файла *temp.dat*. Запустите ВП несколько раз и с помощью текстового редактора убедитесь, что ВП добавляет в файл новые значения.



Совет

Следует удалить функцию **Format Into File**, заменив ее функциями **Format Into String** и **Write File**. Для

перемещения текущей метки файла, следует использовать параметры **pos mode** и **pos offset** функции **Write File**.

Сохраните ВП под именем *Регистратор температуры 2.vi*.