Министерство науки и высшего образования
Российской Федерации

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Воронежский государственный технический университет»

Кафедра конструирования и производства радиоаппаратуры

СТРУКТУРА И ВОЗМОЖНОСТИ ПАКЕТА LabVIEW. СОЗДАНИЕ ВИРТУАЛЬНОГО ИЗМЕРИТЕЛЬНОГО СРЕДСТВА

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к выполнению лабораторных работ по дисциплине
«Приемоусилительные и видеотелевизионные системы»
для студентов направления 11.03.03 «Конструирования и
технология электронных средств»
(профиль «Проектирование и технология радиоэлектронных
средств») всех форм обучения

Воронеж 2021

УДК 681.3

Составители:

доктор. техн. наук А. В. Башкиров,
канд. техн. наук И.С. Бобылкин.

Изучения работы возможности пакета LabView: методические указания к выполнению лабораторных работ по дисциплине «Приемоусилительные и видеотелевизионные системы» для студентов направления 11.03.03 «Конструирования и технология электронных средств» (профиль «Проектирование и технология радиоэлектронных средств») всех форм обучения/ ФГБОУ ВО «Воронежский государственный технический университет»; сост.: А. В. Башкиров, И.С. Бобылкин. Воронеж: Изд-во ВГТУ, 2021. 53 с

Методические указания предназначены на изучение и освоение принципов создания виртуальных средств, моделирующих работу процессора вычислительного устройства.

Предназначены для проведения лабораторных работ по дисциплине «Приемоусилительные и видеотелевизионные системы» для студентов 3 курса.

Методические указания подготовлены в электронном виде в текстовом редакторе MS Word 2003 и содержатся в LR3TTI.doc
Ил. 27. Библиогр.: 3 назв.

УДК 681.3
ББК 38.54

Рецензент - О. Ю. Макаров, д-р техн. наук, проф. кафедры конструирования и производства радиоаппаратуры ВГТУ

Издается по решению редакционно-издательского совета Воронежского государственного технического университета

ВВЕДЕНИЕ

Настоящий курс лабораторных работ направлен на изучение и освоение принципов создания виртуальных средств, моделирующих работу процессора вычислительного устройства.

Программная система LabView является удобным средством для проектирования измерительных каналов, приборов, систем. Она обеспечивает построение и моделирование измерительных структур различной сложности. Система имеет библиотеку виртуальных модулей (моделей) измерительных средств, их отдельных блоков и компонентов. Она позволяет пользователю создавать виртуальные измерительные приборы любой сложности и формировать свою библиотеку виртуальных средств (VI). Система обладает удобными средствами редактирования и отладки и обеспечивает работу с реальными измерительными приборами, модулями и сигналами.

Создание виртуального измерительного средства связано с определением его измерительной функции, созданием лицевой панели с органами управления и средствами представления данных, созданием структурной схемы, выполняющей заданную измерительную функцию, редактированием и отладкой работы измерительного устройства. Для этого система поддерживает соответствующие режимы: создание лицевой панели измерительного прибора Panel–, создание структурной схемы и отладка работы Diagram.

Каждый режим имеет свое окно, панель управления и поддерживается библиотекой (*палитрой*) моделей функциональных блоков (*виртуальных модулей*).

Лабораторная работа № 1

Знакомство с LabVIEW

Цель работы:

Получение основных сведений о программно-инструментальной средеLabVIEW.

Задачи работы:

– Создание простейших виртуальных приборов (VI).

– Моделирование простейших вычислительных алгоритмов.

– Структура виртуального прибора LabVIEW

Отчет:

- титульный лист;

– цели и задачи лабораторной работы;

– задание на лабораторную работу;

– результаты выполненной работы.

Справочно-методический материал. Файл LabVIEW – виртуальный инструмент – состоит из двух панелей. Одна из них имитирует переднюю панель реального физического прибора, (рисунок 1), на второй панели (рисунок 2) строится блок-схема виртуального инструмента на языке G.

Обычно при запуске LabVIEW на экране появляются обе панели, расположенные каскадом. Одновременное нажатие клавиш [Ctrl+T] позволяет упорядочить расположение панелей: слева – передняя, справа – блок-схема. В строке заголовка блок-схемы к имени текущего файла добавляется слово «Diagram», что позволяет отличить эту панель от передней. Под этими именами обе панели представлены на панели задач как два самостоятельных окна.

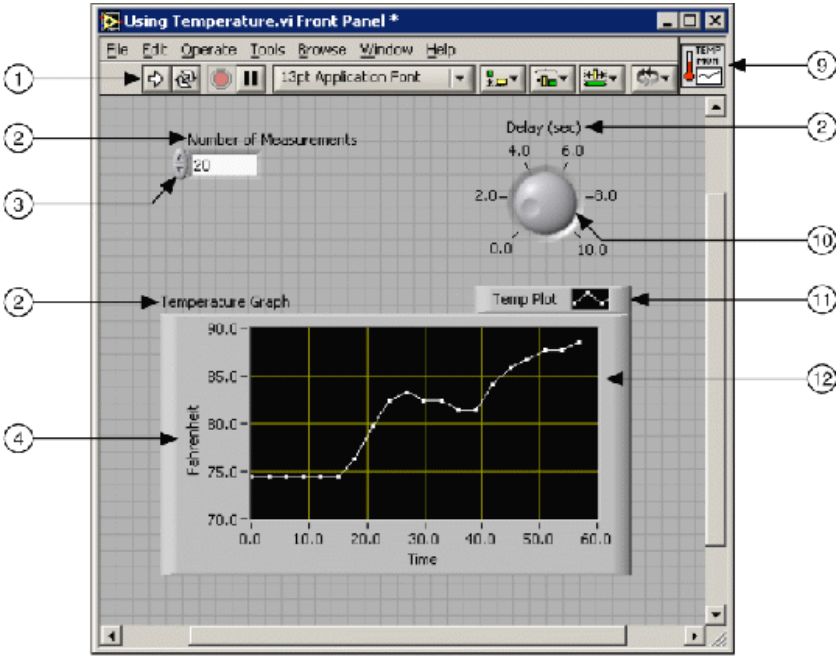


Рис. 1. Передняя панель реального физического прибора

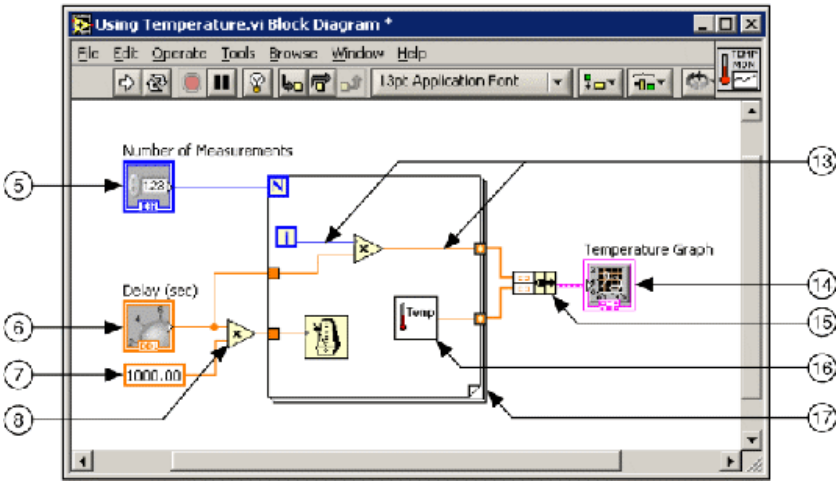


Рис. 2. Блок-схема виртуального инструмента на языке G

Цифрами на рисунках обозначены:

1. Панель инструментов (Toolbar), 2. Ярлык (Label) 3. Цифровойрегулятор (Numeric Control), 4. Ярлык (Label) , 5. Терминалцифровогорегулятора (Numeric Control Terminal), 6. Терминалручки (Knob Terminal), 7. Численнаяконстанта(Numeric Constant), 8. Функцияумножения (Multiply Function), 9. Пиктограмма (Icon), 10. Ручка (Knob Control), 11. Описаниеграфика (Plot Legend), 12. График (XY Graph), 13. Соединение, нитьданных (Wire Data Path), 14. Терминалграфика (XY Graph Terminal), 15. Функцияобъединениявкластер (Bundle Function), 16.Подпрограмма, сабви (SubVI), 17. Цикл for (For Loop Structure)

Работа с главным меню LabVIEW Ниже имени файла расположено главное меню панелей, состоящее из пунктов: *File, Edit, Operate, Project, Windows, Help*. Ниже располагается панель управляющих клавиш. Рассмотрим некоторые команды:

File → Close – закрыть файл. Выбор этой опции с передней панели позволяет закрыть файл виртуального инструмента в целом, а выбор этой же опции на панели блок-схемы убирает только одну панель блок-схемы.

Edit→ Remove Bad Wires – удалить ошибочные соединения проводов

Operate → Run – запуск программы.

Operate → Make Current Values Default – сделать текущие величины (источников, приемников) значениями по умолчанию. Опция позволяет сохранять численные значения, установленные пользователем, до следующего вызова программы. Если пользователь не выбрал эту опцию до сохранения файла, то при следующем вызове этого файла значения источников и приемников будут нулевыми.

Windows → Show Diagram – показать блок-схему (находится на передней панели).

Windows → Show Panel – показать переднюю панель (находится на панели блок-схемы).

Windows → Show Controls Palette – показать набор управляющих элементов Controls передней панели.

Windows → Show Functions Palette – показать набор функций *Functions* (находится на панели блок-схемы).

Windows → Show Tools Palette – показать набор инструментов *Tools* (опция присутствует на обеих панелях).

Windows → Tile Left and Right; Windows → Tile Up and Down – эти опции позволяют располагать панели справа и слева, сверху и снизу соответственно.

Help → Show Help – вызов окна помощи. Используйте эту опцию для вызова описания виртуального инструмента, функции или определения типа провода, а также для определения названия выводов виртуального инструмента. Для этого помещайте конец провода «катушки» на тот вывод инструмента, название которого необходимо узнать, при этом будут мигать изображения обоих выводов – на блок-схеме и в окне помощи.

Help → Online Reference – вызов справочной системы LabVIEW.

Управляющие клавиши

Клавиша *«Run»* («Пуск») располагается в левом верхнем углу обеих панелей. После «нажатия» этой клавиши происходит запуск программы. Остановка выполнения программы должна быть предусмотрена самой программой. На этапе сборки блок-схемы или при наличии в ней ошибок изображение стрелки на клавише разделено на две части. Если «нажать» на клавишу при таком виде стрелки, на экран будет выведен список ошибок (Error List). Список ошибок позволяет найти место каждой ошибки. Для этого выделяют строку с интересующей ошибкой, нажимают кнопку *«Find»*. Место ошибки будет выделено программой.

Клавиша *«Run Continiously»* («Циклический режим») – запуск программы в режиме цикла. Выполнение можно

приостановить клавишей *«Pause»* («Пауза»).

Клавиша *«Abort Execution»* («Стоп») – остановка программы. Клавиши, присутствующие только на панели блок-схемы:

- клавиша *«Highlight Execution»* («Лампочка») - используется для наблюдения прохождения данных по проводам в замедленном и пошаговом режимах. Передача данных от элемента к элементу схемы изображается мультипликацией;
- остальные три клавиши используются для пошагового выполнения программы.

Наборы инструментов

Набор инструментов *Tools* (*Show Tools Palette*) позволяет изменить вид и позиции курсора. Основные позиции (рисунок 1):

- «стрелка» – перемещение, выделение и изменение размеров объектов;
- «катушка» – соединение объектов блок-схемы проводами.
- Активный элемент – конец провода;
- «А» – печать текста с клавиатуры, ввод числовых данных в окна источников и метки объектов;
- «кисть» – раскрашивание объектов и фона. Этот вид курсора не используется для вызова всплывающего меню объектов правой кнопки мыши;
- «рука» – изменяет позиции выключателя и тем самым управляет цифровыми источниками (нажимая на клавиши «больше»или «меньше»), виртуальными осциллографами (нажимая на переключатели и кнопки управления ими) и другими объектами. Также используется для ввода числовых данных.

Набор управляющих элементов *Controls* передней панели (*Show Controls Palette*) позволяет вывести на левую панель контрольно-измерительные приборы, виртуальные осциллографы, кнопки. При помещении курсора на окно

набора в верхней части окна высвечивается название соответствующего поднабора (рисунок1) (например, «*Numeric*» – «Цифровые приборы»). Поднабор открывается нажатием левой кнопки мыши.

Перемещая курсор по элементам поднабора, можно узнать название прибора. Элемент выделяется квадратной рамкой, его название отображается вверху поднабора.

Набор инструментов *Functions* панели блок-схемы (*Show Functions Palette*) содержит функции и виртуальные инструменты (VI), используемые для построения блок-схемы (рисунок 2). Из набора *Functions* вызываются: управляющие структуры (циклы *While, For*), формульный узел (*Formula Node*) –*Structures* поднабор; преобразователи строк – *String* поднабор; приборы преобразования Фурье и линейной алгебры – *Analysis* поднабор и многие другие.

Название функции или виртуального инструмента определяется аналогично набору *Controls*. Наборы *Controls* и *Functions* можно вывести, щелкнув правой кнопкой мыши в любом месте передней панели и панели блок-схемы соответственно.

Элементы в LabVIEW имеют один или несколько выводов (терминалов). Схему терминалов позволяет рассмотреть опция *Show>>Terminals* выпадающего меню объекта на панели блок-схемы.

Все приборы передней панели имеют один вывод, большинство элементов панели блок-схемы (из набора *Functions*) – несколько. Вывод объекта, принимающий данные, будем называть входом. Вывод, который передает данные другим элементам, назовем выходом. Если объект только передает данные, то его называют источником. Его вывод является выходом. Объект, который только принимает данные – приемник. Вывод этого объекта – вход. Термины «источник» –«выход», «приемник» – «вход» однозначно соответствуют друг другу. Выходы элементов панели блок-схемы выделяются утолщенной линией. В любом канале

передачи данных все подключенные к каналу выводы объектов должны быть согласованы:

1) по типу выводов;

2) по типу передаваемых и принимаемых данных.

К каналу связи должен быть подключен только один источник, к одному источнику данных можно подключить неограниченное число приемников. Соединение только одних приемников (или источников) признается программой ошибочным.

Для выделения проводов выбрать пиктограмму с изображением стрелки. Установите стрелочный указатель на участок провода, который необходимо удалить. Нажатием левую кнопку мыши один раз выделяется один сегмент, двойным щелчком выделяется ветвь от узла до элемента, тройным – все разветвленное соединение.

Типы и проводники данных. Создание подпрограмм ВП. 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. Компонентами, определяющими совместимость соединения, являются тип данных элемента управления и/или отображения и тип данных поля ввода/вывода.

Типы данных. В данном курсе используются следующие типы данных:

Numeric (численный тип)

- Floating point — число с плавающей запятой, 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, 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, отображается в виде зеленых терминалов. Логический тип может принимать только два значения: 0 (FALSE) или 1 (TRUE).

String — строковый тип, 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 включают типы данных составляющих элементов и принимают соответствующий им цвет.

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.

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.

Если выбранный объект помещается на блок-диаграмме недалеко от другого объекта, среда LabVIEW показывает пунктирные временные проводники данных, намечающие области возможного соединения. Следует обратить внимание, что при отпускании кнопки мыши LabVIEW автоматически подключает проводник данных к полю ввода/вывода данных, выбранного объекта.

Корректировка параметров автоматического подключения проводников осуществляется через пункты главного меню Tools>>Options>>Block Diagram.

Пример созданрия виртуального прибора

Преобразование °C в °F

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

Лицевая панель

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

Рис.3. Лицевая панель ВП Преобразование температуры

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.

a. Выберите цифровой элемент управления в разделе

палитры Элементов в подразделе Controls»Numeric (Числовые элементы). Для вывода на экран палитры Controls (Элементов) следует щелкнуть правой кнопкой мыши по рабочему пространству лицевой панели.

b. Поместите цифровой элемент управления на лицевую панель.

c. В поле собственной метки элемента управления напечатайте «Град С» и щелкните мышью в свободном пространстве лицевой панели или нажмите кнопку Enter, показанную слева, на инструментальной панели. Если сразу после создания элемента не присвоить имя его собственной метке, то LabVIEW присвоит имя, заданное по умолчанию. Собственная метка в любое время доступна для редактирования, оно производится с помощью инструмента ВВОД ТЕКСТА, показанного слева.

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

5. Выберите цифровой элемент отображения в палитре Элементов в подразделе Controls»Numeric (Числовые элементы).

6. Поместите элемент отображения данных на лицевую панель.

7. 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.

Блок-диаграмма

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

Рис.4. Блок-диаграмма ВП «Преобразование температуры»

9. 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.

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

11. 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.

12. Перейдите на лицевую панель, выбрав в главном меню пункт Window»Show Panel.

13. Сохраните ВП, он будет использоваться позднее.

a. Выберите пункт главного меню File»Save.

b. Укажитекаталог c:\exercises\LV Basics I.

c. В диалоговом окне введите ^ *Преобразование C в F (начало).vi*

d. Нажмите кнопку Save.

Запуск В

Создание подпрограмм ВП

После того как ВП сформирован, создана его иконка и настроена соединительная панель, виртуальный прибор можно использовать как подпрограмму в других ВП.

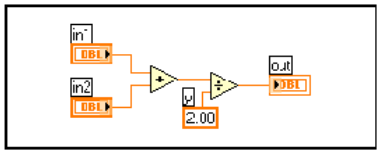
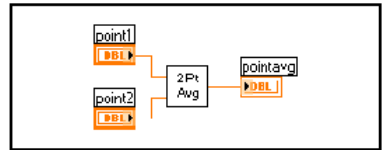
Виртуальный прибор, используемый внутри другого виртуального прибора, называется подпрограммой ВП.

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

Аналоги программного кода и блок-диаграмм

Исходный текст	Вызов подпрограммы
function average (in1,in2,out) {out = (in1 + in2) / 2.0;}	main {ave- age(point1,point2,pointavg);}
	

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 открылось 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.

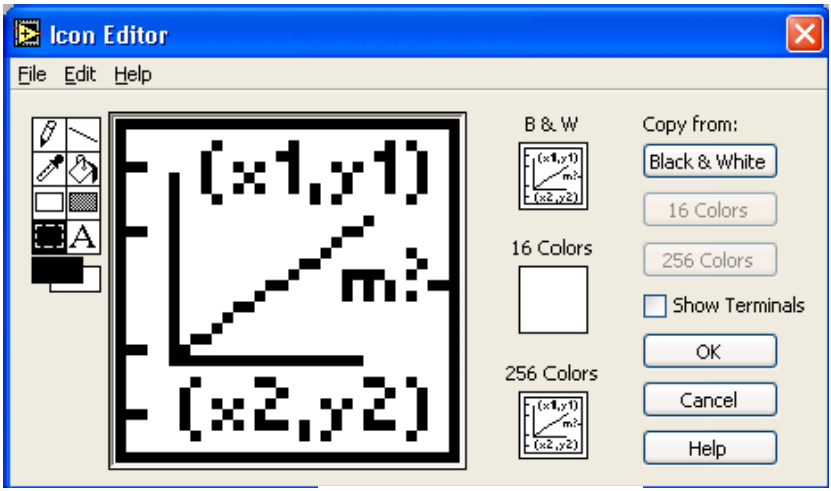


Рис.5. Создание иконки

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.

32 точки.×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

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

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.

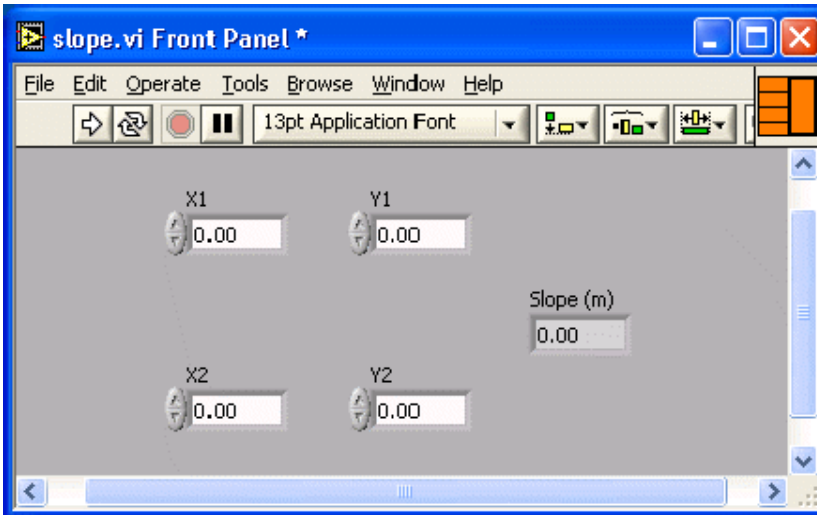


Рис. 6. Соответствие количества полей ввода/вывода от количества отображаемых элементов на соединительной панели

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

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 ввод/вывода данных.

ВП Преобразования °C в °F в виде подпрограммы

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.Появится следующая лицевая панель:

Рис. 7. Лицевая панель ВП

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 , очистите область редактирования иконки.

5. Дважды щелкните по инструменту ПРЯМОУГОЛЬНИК (показан слева), чтобы обвести область редактирования границей выбранного цвета.

6. Создайте следующую иконку:

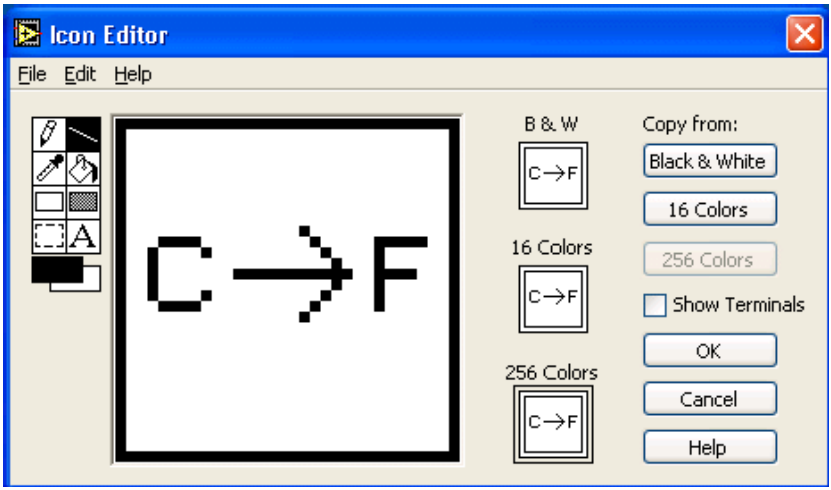


Рис. 8. Вид новой иконки ВП «Преобразование температуры»

a. Введите текст инструментом ВВОД ТЕКСТА, который показан слева.

b. Напечатайте «C» и «F».

c. Для выбора размера шрифта дважды щелкните левой кнопкой мыши по инструменту ВВОД ТЕКСТА.

d. Чтобы нарисовать стрелку, воспользуйтесь инструментом КАРАНДАШ.

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

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 C 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 C. 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.

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 , дважды щелкнуть на нем левой кнопкой мыши.

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

Варианты заданий

1. Создать виртуальный прибор производящий сложение двух целых чисел с использованием панели функций “Mathematics → Numeric”. Обеспечить решение той

же задачи с использованием элемента “Formula Node”. Произвести отладку работы виртуального прибора в циклическом режиме и в режиме однократного запуска.

2. Реализовать вычисления по формуле: z = Ax2+By3. в виде подпрограммы. Создать VI с возможностью задавать значения переменных x и y при помощи элементов «Numeric Control».

3. Реализовать возможность вычисления степени произвольного положительного числа по формуле: : x^y = exp(y * log(x))). Обеспечить возможность задавать значения переменных x и y при помощи элементов «Numeric Control». Вычисление степени реализовать в виде SubVI.

4. Создать виртуальный прибор, отображающий осциллограмму виде фигуры Лиссажу. Обеспечить возможность регулировки частоты исходных сигналов. Сохранить в файле осциллограммы для соотношения частот 1:1, 1:2 и 1:3.

5. Создать виртуальный прибор, имитирующий двухлучевой осциллограф, отображающий гармонический сигнал и прямоугольные импульсы одинаковой частоты. Обеспечить возможность регулировки скважности прямоугольных импульсов.

6. Создать виртуальный прибор, отображающий случайный сигнал с временем выборки 1с. Обеспечить световую сигнализацию превышения некоторого, произвольно задаваемого уровня сигнала.

7. Создать виртуальный прибор, отображающий гармонический сигнал с возможностью наложения шума. Создать элементы управления, позволяющие регулировать амплитуду и частоту гармонического сигнала, а также включение и выключение наложения шумового сигнала.

Лабораторная работа № 2
Многократные повторения и циклы

Цель работы

Получение навыков программирования циклических алгоритмов и сравнение способов их реализации в системе LabView с традиционными языками программирования.

Задачи работы

Получить навыки работы с циклическим алгоритмом For.

Получить навыки работы с циклическим алгоритмом Whil.

Получить навыки работы с оператором выбора Case.

Получить навыки работы с оператором последовательности Sequece.

Получить навыки работы с оператором структуры Event.

Отчет:

- титульный лист;
- цели и задачи лабораторной работы;
- задание на лабораторную работу;
- результаты выполненной работы.

Справочно-методический материал

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.

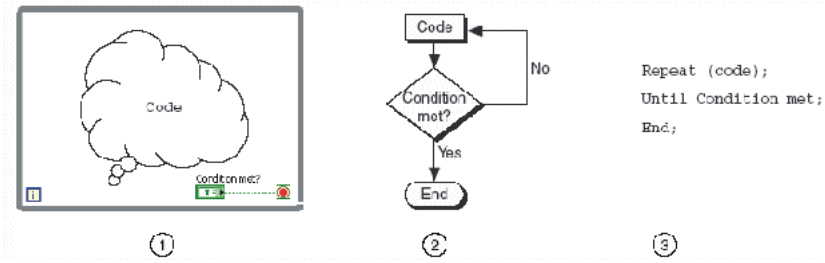
Среда LabVIEW содержит пять структур: Цикл While (по условию), Цикл For (с фиксированным числом итераций), структура Case (Вариант), структура

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).

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. На рисунке 9 представлены 1 – цикл While в среде LabVIEW, 2 – эквивалентную блок-схему работы цикла While, 3 – пример текстового аналога кода работы цикла While.



1. LabVIEW цикл While
2. Блок-схема
3. Текстовый аналог когда

Рис. 9. Представление оператора 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.

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 10, 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.

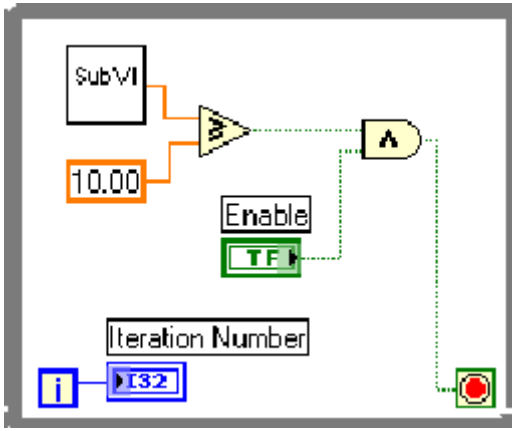


Рис. 10. Блок-диаграмма цикла While «продолжение Если Ложь»

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 (рисунок 11).

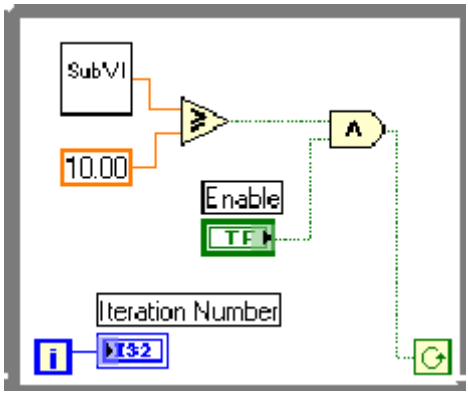


Рис.11. Блок-диаграмма цикла While «продолжение Если Истина» Цикл While выполняется до тех пор, пока выходные данные подпрограммы ВП остаются меньше «10».

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 (рисунок 12) терминал счетчика итераций присоединен к терминалу выхода цикла. Значения из терминала выхода цикла не поступают к элементу отображения номера итерации до завершения цикла While.

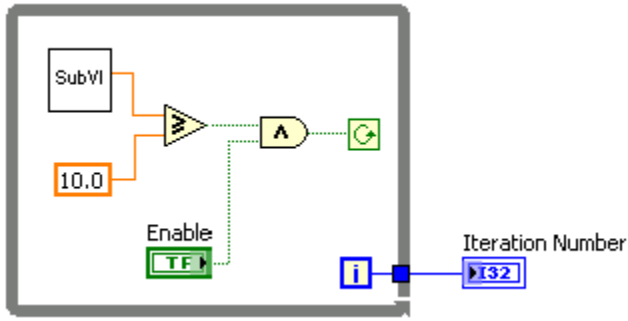


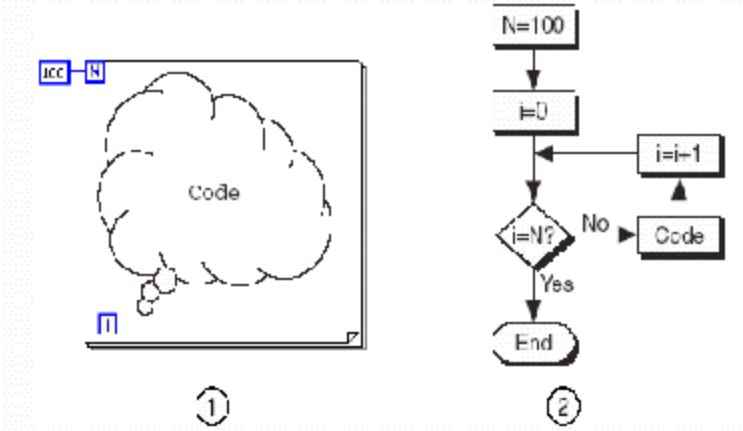
Рис. 12. Блок-диаграмма цикла While со счетчиком итераций на выходе терминала

Лишь последнее значение итерации отображается элементом отображения номера итерации.

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 (рисунок 13) демонстрирует 1 – цикл For в среде LabVIEW, 2 – эквивалентную блок-схему работы цикла For, 3 – пример текстового аналога кода работы цикла For.




```
N=100;
i=0;
Until i=N:
    Repeat (code; i=i+1);
End;
```

3

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

Рис. 13. Представление цикла For

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 14, генерирует случайное число каждую секунду 60 раз и отображает их в элементе отображения данных.

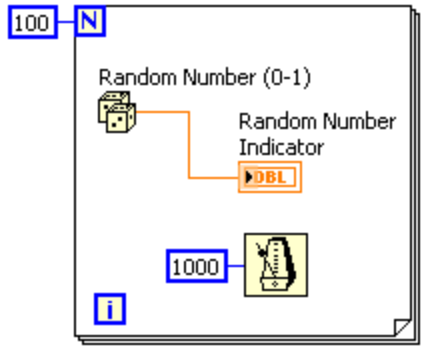


Рис. 14. Блок-диаграмма цикла For с генерацией случайного числа

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. ФункциюWait 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.

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

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.

Для того чтоб 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 рисунке 15 показано использование двух инициализированных сдвиговых регистров.

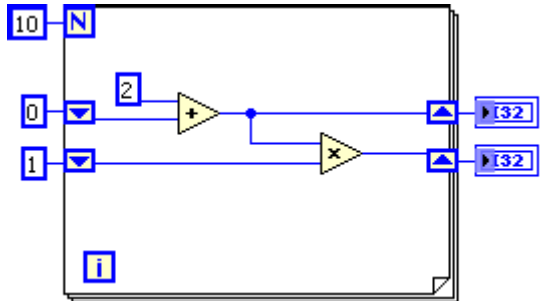


Рис. 15. Блок-диаграмма цикла с двумя сдвиговыми регистрами

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.

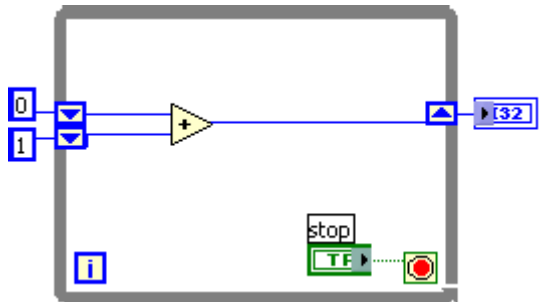


Рис. 16. Блок-диаграмма цикла со стеком сдвиговых регистров

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 сохранят 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.

Пример1:

1. Доступ к данным предыдущих итераций BasicsI. Лицевая панель этого ВП, показанная на рисунке 17, уже создана

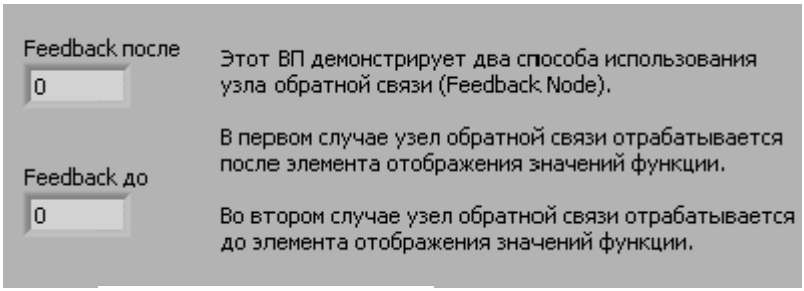


Рис. 17. Блок-диаграмма

2. Отобразите блок-диаграмму, показанную на рис. 18. Разместите лицевую панель и блок-диаграмму на экране так, чтобы они не перекрывали друг друга. Переместите палитры Tools и Functions, если это необходимо.

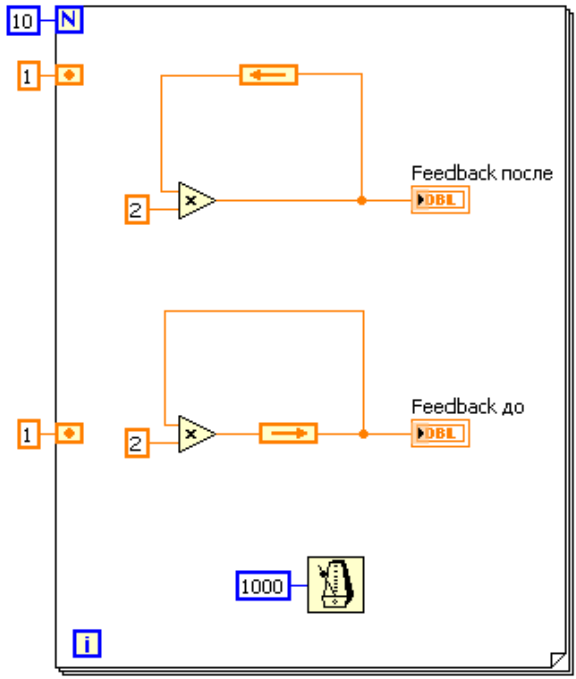


Рис. 18 - блок-диаграмма

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.

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.

3. 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.

4. Замените узел обратной связи сдвиговым регистром, как показано на следующей блок-диаграмме (рис. 19):

a. Выделите нижний узел обратной связи и нажмите клавишу , 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.\

5. Запустите ВП. Обратите внимание, что узел обратной связи и сдвиговый регистр выполняют одинаковые

функции.

6. Не закрывайте ВП, перейдите к выполнению дополнительных упражнений или закройте ВП, не сохраняя его.\

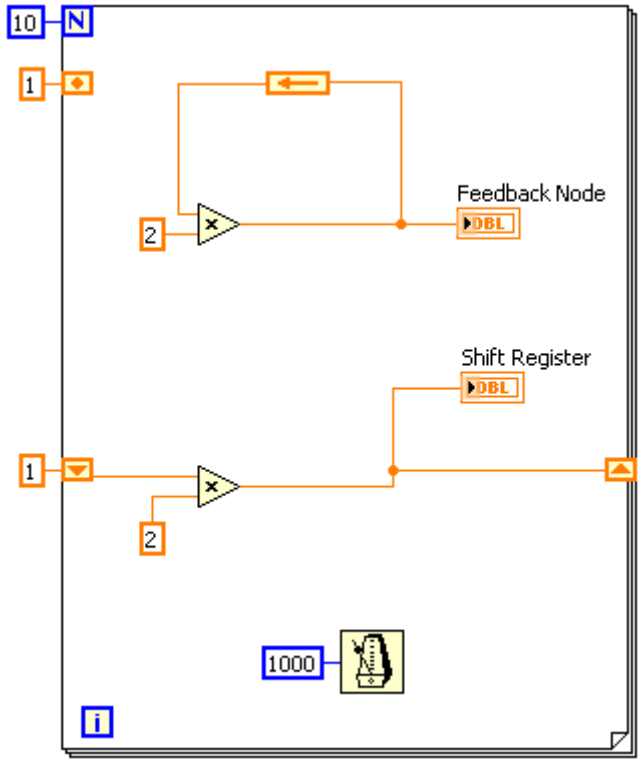


Рис. 19

Варианты заданий

1. Создать виртуальный прибор, генерирующий с интервалом в 1 сек. последовательность чисел Фибоначчи в виде таблицы. Обеспечить возможность очистки значений таблицы в произвольный момент времени.

Используемые компоненты:

Shift register

Add
Build table
Time delay

2. Создать виртуальный прибор, отображающий сумму или произведение двух сигналов (прямоугольного и гармонического) в зависимости от положения тумблера на лицевой панели.

Используемыекомпоненты:
Simulate signal
Formula
Waveform graph
Timedelay
Select
Toggleswitch

3. Создать виртуальный прибор, иллюстрирующий работу логических операций «И» «ИЛИ»«НЕ» на одной лицевой панели.

Используемыекомпоненты:
And
Or
Not
Led
Rocker

4. Создать виртуальный прибор, вычисляющий значение факториала. Реализовать вывод на цифровой индикатор промежуточных значений с интервалом 1сек. Алгоритм вычисления факториала реализовать в виде SubVI

Используемые компоненты:
Forloop
Shift register
Multiply
Increment
Numeric indicator
Timedelay

5. Создать виртуальный прибор, вычисляющий

значение x^y (возведение в степень для целых чисел – циклический алгоритм). Реализовать ввод исходных данных и вывод на цифровой индикатор результата и промежуточных значений с интервалом 1сек. Алгоритм возведения в степень реализовать в виде SubVI

Используемые компоненты:
For loop
Shift register
Multiply
(Decrement)
Numeric indicator
Numeric control
Time delay

6. Реализовать подпрограмму для вычисления суммы: $\sum_{n=0}^{\infty}(-1)^n \frac{x^{2n+1}}{(2n+1)!}$ создать VI с возможностью задания точности вычислений по данной формуле, а также значения переменной *x*.

7. Реализовать подпрограмму для вычисления суммы: $\sum_{k=0}^{\infty}\frac{(-1)^{k+1}}{(2k)!}\left(\frac{x}{3}\right)^{4k}$ с точностью 0.001. Значение *x* передать вкачестве параметра.

Лабораторная работа № 3
Массивы

Цель работы

Получение навыков работы с динамическими массивами с использованием циклических алгоритмов.

Задачи

Освоение технологии применения туннелей и автоиндексации.

Отчет:

- титульный лист;
- цели и задачи лабораторной работы;
- задание на лабораторную работу;
- результаты выполненной работы.

Справочно-методический материал

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.

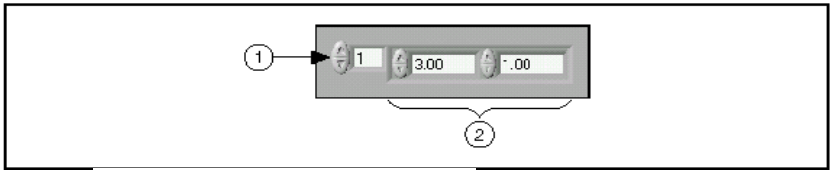
Нельзя создать массив, состоящий из массивов. Все элементы массива упорядочены. Чтобы к ним было легко обращаться, каждому элементу присвоен индекс. Нумерация элементов массива всегда начинается с 0.

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 палитрыControls>>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

Рис.20. Элемент 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 (6.4=24).

Рисунок. 21. Двухмерный массив

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.

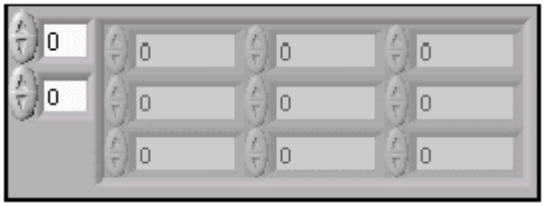


Рис. 22. Неинициализованный двумерный массив

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.

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. Из рисунка 22 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.

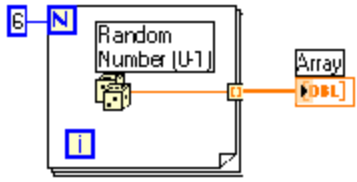


Рис. 23. Автоиндексация при работе циклов

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.

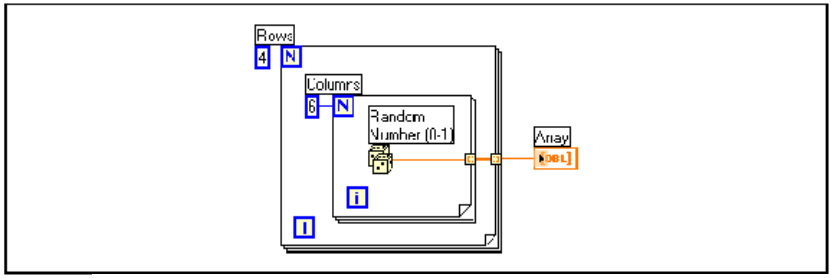


Рис. 24. Создание двумерного массива с помощью двух циклов

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.

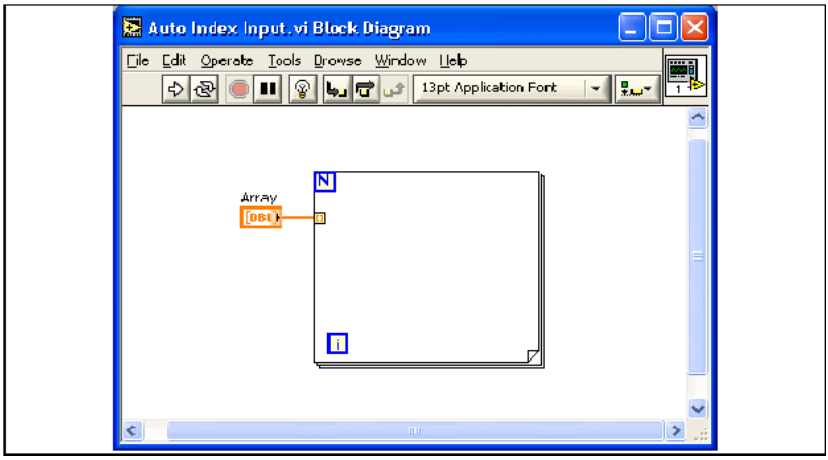


Рис. 25. Задание количества итераций в цикле

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 меньшая 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.

Пример: Работа с массивами

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.

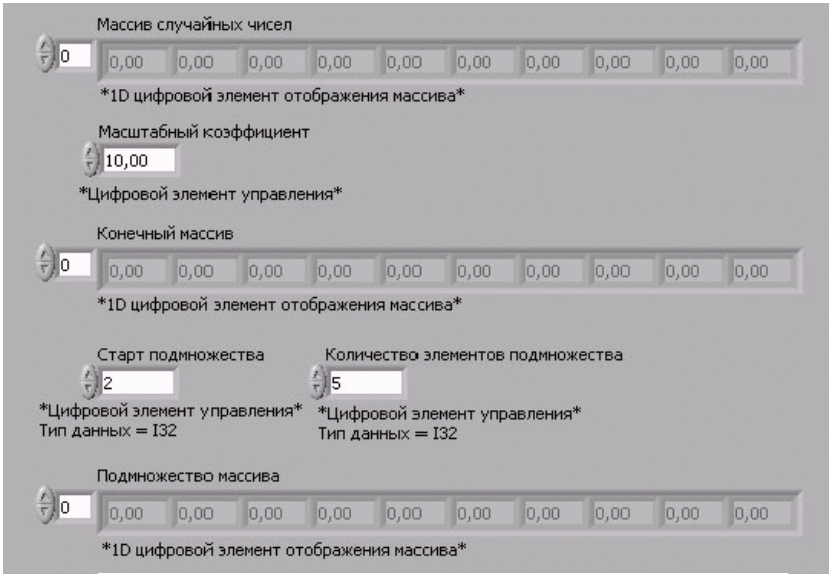


Рис. 26. Лицевая панель ВП «работа с массивами»

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 пунктRepresentation, затем пункт I32.

i. Значения элементов управления данных пока не изменяйте.

2. Постройте блок-диаграмму, как показано ниже.

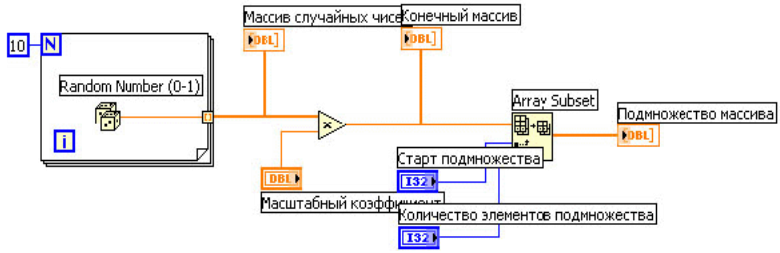


Рис. 27. Блок-диаграмма «Работа с массивами»

 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*

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.

Варианты заданий

1. Создать виртуальный прибор, заполняющий двумерный массив случайными числами в бесконечном цикле. Добавить элемент управления (кнопку) с помощью которого происходит остановка цикла и подсчет среднего арифметического значения элементов массива. Размер массива – произвольный.

2. Реализовать VI для заполнения двумерного массива размером 10 х 10 элементов возрастающими числами от 1 до 100.

3. Присвоить элементам двумерного массива расположенным выше главной диагонали случайные значения.

4. Создать SubVI для подсчета суммы элементов произвольной строки двумерного массива. Продемонстрировать его работу на трех различных массивах.

5. Заполнить элементы одномерного массива из 100 элементов числами Фибоначчи. Вычисления последовательности чисел реализовать в виде подпрограммы.

6. Заполнить четные строки двухмерного массива нулями, нечетные заполнить единицами. Размер массива – произвольный.

7. Подсчитать сумму элементов двумерного массива находящихся выше главной диагонали.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Суранов, А. Я. LabVIEW 7: справочник по функциям [Текст]/А.Я. Суранов. — М. : ДМК Пресс, 2005. — 510 с.
2. Тревис, Д. LabVIEW для всех : пер. с англ. [Текст] / Д. Тревис. — М. : ПриборКомплект : ДМК Пресс, 2005. — 537 с.
3. Евдокимов, Ю. К. Labview для радиоинженера: от виртуальной модели до реального прибора [Текст]: учеб. пособие для вузов/ Ю.К. Евдокимов. — М. : ДМК Пресс, 2007. — 400 с.

СТРУКТУРА И ВОЗМОЖНОСТИ ПАКЕТА LabVIEW. СОЗДАНИЕ ВИРТУАЛЬНОГО ИЗМЕРИТЕЛЬНОГО СРЕДСТВА

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к выполнению лабораторных работ по дисциплине «Приемоусилительные и видеотелевизионные системы» для студентов направления 11.03.03 «Конструирования и технология электронных средств» (профиль «Проектирование и технология радиоэлектронных средств») всех форм обучения

Составители:
доктор. техн. наук А. В. Башкиров,
канд. техн. наук И.С. Бобылкин.

Компьютерный набор И.С. Бобылкин.

Подписано к изданию _____.
Уч.-изд. л. _____.

ФГБОУ ВО «Воронежский государственный технический университет»
394026 Воронеж, Московский просп., 14