

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
Белорусский национальный технический университет

Кафедра «Робототехнические системы»

А. Р. Околов

А. В. Дрозд

И. И. Гутич

СИСТЕМЫ АВТОМАТИЗИРОВАННОГО КОНТРОЛЯ

Учебно-методическое пособие
для студентов специальности 1-53 01 01 «Автоматизация
технологических процессов и производств»

*Рекомендовано учебно-методическим объединением
по образованию в области автоматизации
технологических процессов, производств и управления*

Минск
БНТУ
2018

УДК 681.51(075.8)
ББК 32.965я7
О-52

Рецензенты:
Кафедра ПОиСОИ БГТУ
(зав. каф. – кандидат технических наук, доцент *М. С. Шмаков*);
профессор *С. Н. Павлович*

Околов, А. Р.
О-52 Системы автоматизированного контроля : учебно-методическое пособие для студентов специальности 1-53 01 01 «Автоматизация технологических процессов и производств» / А. Р. Околов, А. В. Дрозд, И. И. Гутич. – Минск: БНТУ, 2018. – 55 с.
ISBN 978-985-550-797-1.

Данное учебно-методическое пособие направлено на приобретение навыков практического использования систем автоматического проектирования различных компьютерных контрольно-измерительных систем (ККИС) на основе использования языка LabVIEW, освоение студентами основ построения моделей средств измерений, проведение имитационного моделирования процессов измерения и контроля, а также привитие обучаемым навыков работы с объектно-ориентированным программным обеспечением.

Учебно-методическое пособие может быть полезно студентам, инженерам и преподавателям, занимающимся проектированием и эксплуатацией систем автоматизированного контроля и управления технологическими процессами.

УДК 681.51(075.8)
ББК 32.965я7

ISBN 978-985-550-797-1

© Околов А. Р., Дрозд А. В.,
Гутич И. И., 2018
© Белорусский национальный
технический университет, 2018

ВВЕДЕНИЕ

В настоящее время к подготовке инженеров по автоматизации и управлению предъявляются высокие требования в области автоматизации измерений, контроля и испытаний. Одним из обязательных элементов освоения данной предметной области является приобретение навыков практического использования систем автоматического проектирования различных компьютерных контрольно-измерительных систем (ККИС). LabVIEW или Laboratory Virtual Instrument Engineering Workbench (Среда разработки лабораторных виртуальных приборов) представляет собой среду графического программирования, которая широко используется в промышленности, образовании и научно-исследовательских лабораториях в качестве стандартного инструмента для сбора данных и управления приборами.

Преимущества этого языка проявляются при выполнении следующих задач:

- исследования возможности контроля измерительной системой параметров объекта измерений;
- анализа и оптимизации структуры ККИС;
- расчета и анализа статистических и динамических метрологических характеристик (МХ) ККИС и ее компонентов.

Построенные с использованием данного пакета имитационные модели наглядно отражают суть реальных физических процессов, протекающих в объектах исследования.

Сложность выполнения заданий по вариантам возрастает от занятия к занятию. Учебно-методическое пособие построено таким образом, чтобы по его окончании обучаемые имели достаточный уровень знаний и навыков работы с интерфейсом программной среды LabVIEW.

1. ИЗУЧЕНИЕ ОСНОВНЫХ ПОНЯТИЙ ПРОГРАММНОЙ СРЕДЫ LABVIEW И ВИРТУАЛЬНОГО ПРИБОРА (ВП)

1.1. Программная среда LabVIEW

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

Запуск среды программирования LabVIEW осуществляется либо двойным кликом мыши на ярлыке LabVIEW, который находится на рабочем столе, либо из раздела Пуск–Программы – National Instruments LabVIEW. При входе в главное меню LabVIEW пользователю предлагается создание нового виртуального инструмента (**New VI**) или открытие уже существующего (**Open VI**).

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

Разработка VI (ВП) осуществляется на двух панелях, находящихся в двух окнах, – передней (лицевая панель) и функциональной (блок-диаграмма). Лицевая панель – интерфейс пользователя – 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.

После этого на блок-диаграмме ВП осуществляется программирование с использованием палитры Функций (**Functions**), которая включает графическое представление функций для управления объектами на лицевой панели.

Структура панелей одинакова. Основным элементом каждой панели является рабочая область, снабженная горизонтальным и вертикальным скроллингами, в которой и размещаются элементы. Также на панелях имеется верхнее меню и набор управляющих кнопок (рис. 1.1):

- кнопка «стрелка» – пуск выполнения программы. Если в программе имеются ошибки, то данная кнопка расколота на две части;

- кнопка «стрелки в цикле» – запуск программы в циклическом режиме;
- кнопка «красный круг» – остановка выполнения программы;
- кнопка «две вертикальные черты» – пауза в выполнении программы.



Рис. 1.1. Управляющие кнопки

Для обеих панелей доступна панель **Tools Palette** (рис. 1.2), включающая набор управляющих кнопок для изменения режима редактирования. Перечислим некоторые из них:

- кнопка «указательный палец» – служит для изменения позиций выключателей и кнопок, управления значениями цифровых регуляторов, настройки виртуальных осциллографов и др.;
- кнопка «стрелка» – выделение, перемещение объектов, изменение их размера;
- кнопка «А» – открытие и редактирование текстового окна;
- кнопка «катушка» – служит для соединения объектов на функциональной панели;
- кнопка «кисть» – раскрашивание объектов или фона;



Рис. 1.2. Панель Tools

- кнопка «рука» – перемещение рабочей области панели в окне;
- кнопка «пипетка» – выбор текущего цвета из имеющихся на панели;
- кнопка «красный круг» – размещение и снятие точек остановки выполнения программы на функциональной панели;
- кнопка «Р» – размещение на функциональной панели локальных окон для отображения текущих значений данных, передаваемых в ходе выполнения программы.

При активной передней панели становится доступной панель **Controls** (рис. 1.3), она вызывается либо щелчком правой кнопки мыши в рабочем пространстве лицевой панели, либо необходимо выбрать в пункте главного меню **Window / Show Controls Palette**. С ее помощью осуществляется визуальное размещение элементов управления и элементов отображения на лицевой панели ВП. В панели Controls они распределены по отдельным группам по некоторым признакам: числовые, логические, строковые, массивы, диалоговые, ActiveX, Internet и др.



Рис. 1.3. Панель Controls

Рассмотрим основные подпанели панели **Controls**:

- **Numeric** (числовые значения). Состоит из элементов управления и элементов отображения для числовых данных;
- **Boolean** (булевы значения). Состоит из элементов управления и элементов отображения для булевых величин;

- **String & Table** (строковые значения и таблицы). Состоит из элементов управления и элементов отображения для ASCII строк и таблиц;
- **List & Ring** (списки и закольцованные списки). Состоит из элементов управления и элементов отображения для меню, выполненных в форме списков и закольцованных списков;
- **Array & Cluster** (массивы и кластеры). Состоит из элементов управления и элементов отображения для группировки наборов типов данных;
- **Graph** (виртуальные осциллографы). Состоит из элементов отображения для построения графиков данных в графах или диаграммах в реальном масштабе времени;
- **Path & Refnum** (пути и ссылки). Состоит из элементов управления и элементов отображения для путей и ссылок;
- **Decorations** (оформление). Состоит из элементов управления и элементов отображения графических объектов для настройки дисплеев лицевой панели;
- **Select Control** (выбор регулятора). Отображает диалоговое окно для загрузки самодельных элементов управления;
- **User Controls** (средства управления пользователем). Состоит из специальных средств управления, которые формирует сам пользователь;
- **ActiveX** (объекты ActiveX). Состоит из средств управления, позволяющих внедрить объекты ActiveX на лицевую панель;
- **Dialog** (диалоговая панель). Состоит из стандартных объектов для формирования диалога с пользователем;
- **IMAQ Vision** (обработка изображений). Состоит из средств обработки и анализа изображений;
- **Internet Toolkit** (работа с Internet). Состоит из средств управления, располагаемых на передней панели, позволяющих организовывать работу виртуальных инструментов в сети Internet (ftp, электронная почта, telnet, CGI и другие).

После помещения элементов управления или отображения данных на лицевую панель они получают свое графическое отображение (в виде терминала данных) на блок-диаграмме. Символы на терминале соответствуют типу данных терминала. Например, **DBL** – терминал представляет данные в виде вещественных чисел с двойной точностью, **TF** – логический терминал, **I16** – терминал 16-битных целых и др.

- **String** (строковые функции). Состоит из функций для работы со строковыми величинами;
- **Array** (массивы). Состоит из функций для обработки массивов;
- **Cluster** (кластеры). Состоит из функций для обработки кластеров;
- **Comparison** (сравнение). Состоит из функций для сравнения переменных;
- **Time & Dialog** (время и диалог). Состоит из функций для диалоговых окон, синхронизации и обработки ошибок;
- **File I/O** (ввода/вывода файла). Состоит из функций для осуществления операций по вводу/выводу файлов;
- **Instrument I/O** (инструменты ввода/вывода). Состоит из ВП для связи и управления приборами различной архитектуры;
- **Instrument Drivers** (драйверы приборов). Состоит из ВП, способных управлять внешними приборами, осциллоскопами, генераторами и т. д. через последовательный порт или интерфейс GPIB;
- **Data Acquisition** (сбор данных). Состоит из ВП для использования плат сбора данных;
- **Signal Processing** (обработка сигналов). Состоит из ВП для генерации и обработки сигналов;
- **Mathematics** (математические). Состоит из оптимизационных, алгебраических, интегральных, дифференциальных и других функций;
- **Graphics & Sound** (графика и звук). Состоит из ВП для работы с трехмерной графикой, изображениями и звуком;
- **Communication** (связи). Состоит из виртуальных приборов для работы с сетями TCP, DDE и др.;
- **Application Control** (управление приложением). Состоит из ВП, управляющих виртуальными приборами;
- **Advanced** (расширенная). Состоит из разных функций типа функции библиотечного запроса, манипуляции данных и др.;
- **Report Generation** (генерация отчета). Состоит из ВП, используемых для подготовки отчетных документов;
- **Tutorial** (обучающие программы). Состоит из VI, используемых в обучающей программе LabVIEW;
- **User Libraries** (пользовательские библиотеки). С помощью нее организуется быстрый доступ к нужному VI;
- **Select VI** (выбор ВП). Состоит из диалогового окна для внедрения подпрограмм в текущий ВП;

- **IMAQ Vision** (обработка изображений). Состоит из ВП, используемых для обработки и анализа изображений;
- **Image Acquisition** (получение изображения). Состоит из ВП, используемых для получения и обработки изображений;
- **Internet Toolkit** (работа с Internet). Состоит из ВП, используемых для работы в сети Internet (ftp, электронная почта, telnet, CGI и др.);
- **SQL** (SQL запросы). Состоит из ВП, используемых для организации связи с SQL сервером и обработки запросов.

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

Проводники данных между терминалами аналогичны переменным на обычных языках. Данные идут в только одном направлении, с исходного терминала на один или более терминалов адресата. Провода имеют различную толщину и цвет. Синий цвет соответствует целым числам, оранжевый – вещественным числам, зеленый – логическим, лиловый – строковым данным и т. д.

При нажатии правой кнопки мыши на регуляторе/индикаторе (как на передней, так и на функциональной панели) появляется **контекстное меню**, с помощью которого возможно осуществить:

- замену элемента управления (регулятора) на элемент отображения (индикатора) и наоборот (Change to Control, Change to Indicator);
- быстрый поиск терминала на функциональной панели (Find Terminal) и регулятора/индикатора на передней панели (Find Control, Find Indicator);
- демонстрацию или отказ от названия для описания регулятора/индикатора (Show–Label, Show–Caption);
- настройку параметров регулятора/индикатора (Data Operations);
- замену на другой регулятор/индикатор (Replace);
- получение справки по используемой функции (Online Help);
- открытие для функций соответствующих им констант, индикаторов и регуляторов (Create Constant, Create Indicator, Create Control) и др.

1.2. Задания

1.2.1. Задание 1. ВП. Частотный анализ

1. Запустите LabVIEW Пуск / Программы / 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.Ib и дважды щелкните на Frequency Response.VI (Частотная характеристика). Появится лицевая панель ВП Частотный анализ (рис. 1.5).

4. 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.

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

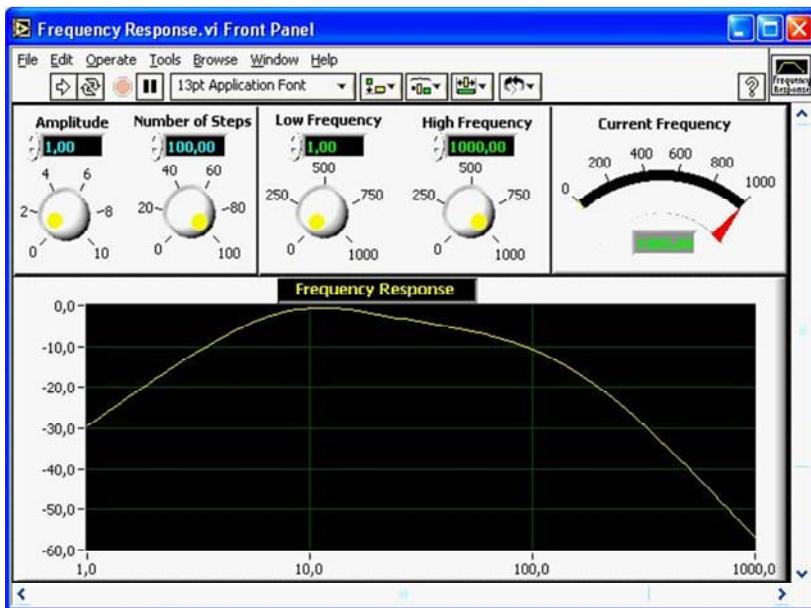


Рис. 1.5. Частотный анализ

Блок-диаграмма

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

7. Откройте окно контекстной справки, выбрав в пункте главного меню Help / Show Context Help. Получить информацию об объекте в окне контекстной справки Context Help можно, наведя на него курсор:

а) поместите инструмент ПЕРЕМЕЩЕНИЕ  над функцией Logarithm Base 10, расположенной под меткой Vode Plot. В окне контекстной справки Context Help появится описание функции;

б) в окне контекстной справки Context Help нажмите кнопку More Help  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. Здесь можно получить подробное описание и других функций;

в) наведите инструмент СОЕДИНЕНИЕ  на поля ввода/вывода данных функции Logarithm Base 10. Соответствующие поля в окне контекстной справки Context Help начнут мигать;

г) переместите инструмент СОЕДИНЕНИЕ на проводник данных. В окне контекстной справки Context Help появится описание типа данных в проводнике.

8. чтобы зафиксировать текущее окно Context Help (Контекстная справка), необходимо либо нажать кнопку , либо выбрать в пункте главного меню Help / Lock Context Help. Когда текущее окно Context Help (Контекстная справка) зафиксировано, то его содержимое не меняется после наведения курсора на другой объект. Для отмены фиксации следует нажать кнопку второй раз.

Контрольные вопросы

1. Из каких основных компонентов состоит ВП?
2. Что понимается под интерфейсом пользователя ВП?
3. Какие палитры доступны для лицевой панели?
4. Какие палитры доступны для блок-диаграммы?
5. Что представляет собой лицевая панель?
6. Каково назначение блок-диаграммы?
7. Из каких подпалитр состоит палитра Controls (Элементы)?
8. Из каких подпалитр состоит палитра Functions (Функции)?
9. На каких панелях осуществляется разработка ВП?
10. Назовите назначение управляющих кнопок на блок-диаграмме.
11. Назовите назначение управляющих кнопок на лицевой панели.
12. Что такое элемент управления и элемент отображения?
13. Назовите основные типы данных.
14. Что такое проводник данных?
15. Каким образом осуществляется вызов контекстной справки?
16. Как можно зафиксировать текущее окно контекстной справки?
17. Назовите назначение контекстного меню.

2. СОЗДАНИЕ, РЕДАКТИРОВАНИЕ И ОТЛАДКА ВИРТУАЛЬНОГО ПРИБОРА (ВП)

2.1. Цель работы

1. Изучить компоненты ВП.
2. Создать ВП (преобразовать °C в °F).
3. Изучить типы данных и проводники данных.
4. Отредактировать ВП.
5. Приобрести практические навыки отладки ВП.

2.2. Методические указания

 Объекты лицевой панели на блок-диаграмме отображаются в виде терминалов данных (графическое изображение прямоугольной формы с буквенно-численными обозначениями). 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.

 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 (Вариант), цикл While (цикл по условию) и т. д. Узлы Add (Сложение) и Subtract (Вычитание) – узлы функций.

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 два 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;
- 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 элементу 01 поражения 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.

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.

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

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.

2.3. Задания

2.3.1. Задание 1. Преобразование °C в °F

Ниже приведена последовательность действий для создания ВП, который будет преобразовывать значение температуры из градусов Цельсия в температуру по Фаренгейту.

Лицевая панель

1. Выберите пункт главного меню File / New / VI, чтобы открыть новую лицевую панель (рис. 2.1).

2. Поместите цифровой элемент управления на лицевую панель. В поле собственной метки элемента управления напечатайте «Град C».

3. Поместите элемент отображения данных на лицевую панель. Он будет использован для отображения значений температуры в °F. В поле собственной метки элемента управления напечатайте «Град F» и щелкните мышью в свободном пространстве лицевой панели или нажмите кнопку Enter.

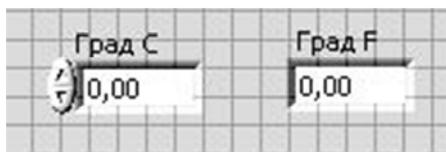


Рис. 2.1. Лицевая панель

На блок-диаграмме LabVIEW создаст терминалы данных, соответствующие элементам управления и отображения. Терминалы данных представляют тип данных соответствующих элементов. Например, терминал данных DBL 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

4. Перейдите на блок-диаграмму, выбрав пункты главного меню Window / Show Diagram.

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

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

7. 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, оно готово для редактирования. Одной константе присвойте значение 1,8, другой 32,0.

8. Соедините объекты блок-диаграммы с помощью инструмента СОЕДИНЕНИЕ.

9. Перейдите на лицевую панель, выбрав в главном меню пункт Window / Show Panel.

10. Сохраните ВП, он будет использоваться позднее.

Запуск ВП

1. Введите число в элемент управления и запустите ВП:

а) для ввода числа в элемент управления следует использовать инструмент УПРАВЛЕНИЕ или инструмент ВВОД ТЕКСТА;

б) нажмите кнопку Run, чтобы запустить ВП;

в) введите несколько разных значений температуры и запустите ВП.

2. Закройте ВП, выбрав пункт главного меню File / Close.

2.3.2. Задание 2. Создание ВП

Создать ВП согласно своему варианту, приведенному в табл. 2.1.

Таблица 2.1

Создание ВП

№	Содержание задания
1	ВП преобразует значение температуры из градусов Цельсия в температуру по шкале Кельвина ($^{\circ}\text{K} = ^{\circ}\text{C} + 273$)
2	ВП преобразует значение температуры из градусов Цельсия в температуру по шкале Реомюра ($^{\circ}\text{R} = ^{\circ}\text{C} \cdot 4/5$)
3	ВП преобразует значение температуры по шкале Кельвина в градусы Цельсия ($^{\circ}\text{C} = ^{\circ}\text{K} - 273$)
4	ВП преобразует значение температуры по шкале Реомюра в градусы Цельсия ($^{\circ}\text{C} = ^{\circ}\text{R} \cdot 5/4$)
5	ВП преобразует значение температуры по Фаренгейту в градусы Цельсия ($^{\circ}\text{F} = 9/5 \cdot ^{\circ}\text{C} + 32$)
6	ВП преобразует значение температуры по шкале Реомюра в температуру по Фаренгейту ($^{\circ}\text{F} = 9/4 \cdot ^{\circ}\text{R} + 32$)

№	Содержание задания
7	ВП преобразует значение температуры из градусов Цельсия в температуру по Реомюру ($^{\circ}R = ^{\circ}C \cdot 4/5$)
8	ВП преобразует значение температуры по Кельвину в температуру по Реомюру ($^{\circ}R = (^{\circ}K - 273) \cdot 4/5$)
9	ВП преобразует по закону Ома значение напряжения и сопротивления в силу тока ($I = U / R$)
10	ВП преобразует значение динамической вязкости μ в кинематическую вязкость ν ($\nu = -\mu / \rho$)
11	ВП преобразует значение напряжения (мВ) и силы тока (мкА) в ватты (Вт) ($P = IU$)
12	ВП преобразует значение напряжения (В) и силы тока (А) в мощность (Вт) ($P = IU$)

Контрольные вопросы

1. Из каких основных компонентов состоит Ваш ВП?
2. Какие узлы на блок-диаграмме ВП Вы знаете?
3. Как отображаются терминалы данных и какие функции они выполняют?
4. Что такое проводник данных?
5. Какие проводники данных Вы знаете?
6. Какие типы данных Вы знаете?
7. Как соединяются объекты проводниками данных?
8. Из каких подпалитр состоит палитра Controls (Элементы)?
9. Из каких подпалитр состоит палитра Functions (Функции)?
10. Как осуществляется пуск разработанного ВП?
11. Назовите назначение управляющих кнопок на блок-диаграмме.
12. Назовите назначение управляющих кнопок на лицевой панели.
13. Что такое элемент управления и элемент отображения?
14. Назовите основные типы данных.
15. Как ввести число в элемент управления?

3. СОЗДАНИЕ ПОДПРОГРАММ ВИРТУАЛЬНОГО ПРИБОРА (ВП)

3.1. Цель работы

1. Изучить подпрограмму ВП.
2. Создать иконку и настроить соединительную панель для возможности использования ВП в качестве подпрограммы ВП.

3.2. Методические указания

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**. Подпрограмма ВП соответствует подпрограмме в текстовых языках программирования. Использование подпрограмм ВП помогает быстро управлять изменениями и отладкой блок-диаграмм.

Любой VI может быть использован как подпрограмма при создании в последующем других виртуальных инструментов. Для объединения нескольких функциональных блоков разрабатываемой блок-диаграммы в подпрограмму достаточно выделить их мышкой на диаграмме, удерживая клавишу Shift, и затем выбрать в верхнем меню пункт **Edit / Create SubVI**. При этом они объединятся в новую подпрограмму с новым значком (иконкой) на функциональной панели. Двойной клик на данном значке позволит вызвать созданную подпрограмму, настроить ее должным образом и сохранить с заданным именем. В последующем данный модуль может быть многократно использован в различных VI.

Каждый виртуальный прибор в правом верхнем углу лицевой панели и в окне блок-диаграммы отображает иконку . 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.

Для редактирования иконки создаваемых VI и подпрограмм достаточно кликнуть правой кнопкой мыши на пиктограмме VI в правом верхнем углу и выбрать пункт Edit Icon... (рис. 3.1).

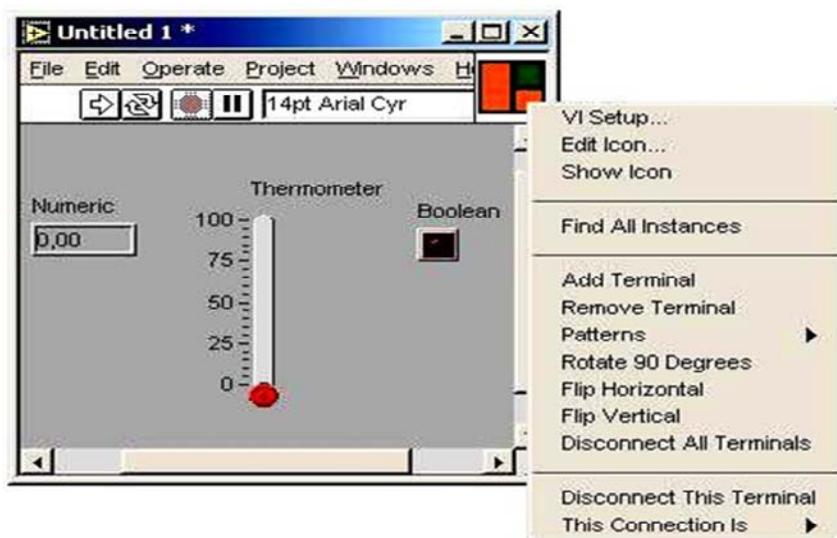


Рис. 3.1. Собственный вариант иконки

С помощью простейших функций графического редактора можно создать собственный вариант иконки. Настройка входов/выходов (терминалов) подпрограмм осуществляется следующим образом. Необходимо нажать правой кнопкой мыши на пиктограмме VI в правом верхнем углу и выбрать пункт Show Connector... (рис. 3.1). При этом пиктограмма разделится на несколько прямоугольников, общий набор и вид которых можно редактировать с помощью всплывающего меню пиктограммы (**добавить/удалить терминал – Add Terminal/Remove Terminal**, поворот на 90 градусов – Rotate 90 Degrees, другой вид – Patterns... и др.). Для того, чтобы сопоставить каждый терминал с определенными данными, необходимо левой кнопкой мыши кликнуть на нужном терминале, а затем – на том элементе управления или отображения на лицевой панели, которой он будет соответствовать. При этом терминал окрасится в цвет, соответствующий типу данных указанного элемента управления или отображения. В результате все терминалы будут связаны с определенными входными или выходными данными.

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** (Функции) 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.

3.3. Задания

3.3.1. Задание 1. ВП. Преобразования °C в °F

В этой работе представлена последовательность действий по созданию иконки и настройке соединительной панели для созданного ВП, который переводит значение измеренной температуры из градусов Цельсия в градусы по Фаренгейту.

Лицевая панель

1. Откройте файл с ранее созданным ВП *Преобразование C в F (начало).vi*.

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 .

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

5. Дважды щелкните по инструменту ПРЯМОУГОЛЬНИК , чтобы обвести область редактирования границей выбранного цвета.

6. Создайте следующую иконку (рис. 3.2):

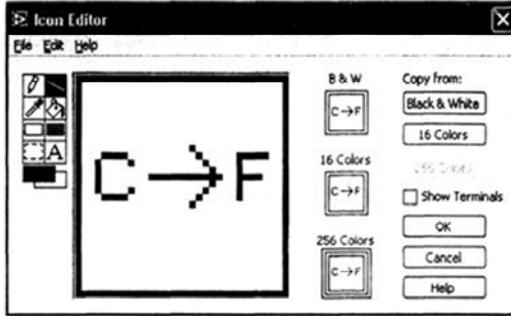


Рис. 3.2. Иконка ВП

- а) введите текст инструментом ВВОД ТЕКСТА ;
- б) напечатайте «С» и «F»;
- в) для выбора размера шрифта дважды щелкните левой кнопкой мыши по инструменту ВВОД ТЕКСТА;
- г) чтобы нарисовать стрелку, воспользуйтесь инструментом

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

д) для передвижения текста и стрелки по полю редактирования иконки используйте инструмент ВЫБОР и стрелки на клавиатуре;

е) в разделе **Copy from** (Копировать из) 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;

ж) в разделе **Copy from** (Копировать из) выберите **16 Colors** и **256 Colors**;

з) 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.

8. Элементам управления и отображения данных назначьте соответственно поля ввода и вывода данных:

а) в пункте главного меню **Help** (Помощь) выберите **Show Context Help** (показать контекстную справку) и выведите на экран окно Context Help (контекстной справки) для просмотра соединений;

б) щелкните левой кнопкой мышки на левом поле соединительной панели. Инструмент УПРАВЛЕНИЕ автоматически поменяется на инструмент СОЕДИНЕНИЕ, а выбранное поле окрасится в черный цвет;

в) щелкните левой кнопкой мыши по элементу **Град С**. Левое поле станет оранжевым и выделится маркером;

г) щелкните курсором по свободному пространству. Маркер исчезнет, и поле окрасится в цвет данных типа соответствующего элемента управления;

д) щелкните левой кнопкой мыши по правому полю соединительной панели и элементу **Град F**. Правое поле станет оранжевым;

е) щелкните курсором по свободному пространству. Оба поля останутся оранжевыми;

ж) 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*.

10. Выберите пункт главного меню **File / Close**. Закройте ВП.

3.3.2. Задание 2. ВП. Термометр

Ниже приведена последовательность действий для создания ВП, который измеряет температуру и отображает значение температуры в градусах Цельсия или температуру по Фаренгейту.

Лицевая панель

1. Создайте элемент отображения данных температуры, как показано ниже на рис. 3.3.

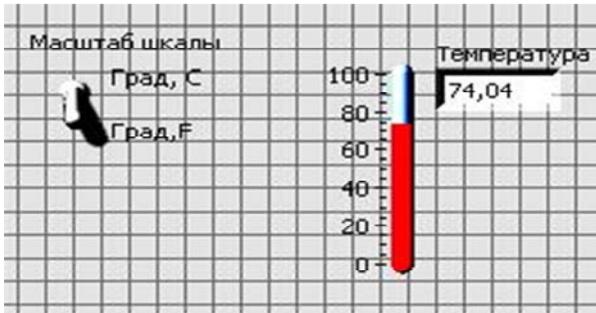


Рис. 3.3. Элемент изображения данных температуры

а) элемент отображения данных, расположенный на палитре Controls в разделе **Numeric** (Числовые элементы) ;

б) напечатайте **Температура** внутри собственной метки и нажмите кнопку Enter на инструментальной панели ;

в) щелкните правой кнопкой мыши по элементу и выберите пункт контекстного меню **Visible Items** (Отображаемые элементы), **Digital Display** (Цифровой индикатор) .

2. Создайте элемент управления в виде вертикального переключателя :

а) выберите вертикальный переключатель, расположенный в палитре Controls раздела **Boolean** (Логические элементы) ;

б) введите имя собственной метки переключателя **Масштаб шкалы** и нажмите кнопку Enter на инструментальной панели;

в) используя инструмент **ВВОД ТЕКСТА**, создайте на лицевой панели свободную метку °C, как показано выше;

г) с помощью инструмента **ВВОД ТЕКСТА** создайте на лицевой панели свободную метку °F, как показано выше.

3. Создайте описание ВП, которое появляется в окне контекстной справки **Context Help** после наведения курсора на иконку ВП:

а) выберите пункт главного меню **File / VI Properties**;

б) выберите пункт **Documentation** (Описание) в разделе **Category** (Категория) из выпадающего меню;

в) в поле ввода текста напечатайте следующее:

Этот ВП измеряет температуру, используя ВП Demo Read Voltage VI.

4. Описание элементов управления и отображения данных, которое появляется в окне контекстной справки **Context Help** после наведения на них курсора:

а) щелкните правой кнопкой мыши по элементу отображения и выберите пункт контекстного меню Description and Tip (Описание и предупреждения);

б) в поле ввода текста напечатайте следующее:

Выводит на экран значения измеренной температуры;

в) введите в поле Tip значение Температура;

г) нажмите кнопку ОК;

д) щелкните правой кнопкой мыши по элементу управления и выберите пункт контекстного меню Description and Tip (Описание и предупреждения);

е) в поле ввода текста напечатайте следующее:

Определяет шкалу (по Фаренгейту или Цельсию), используемую для измерения температуры;

ж) введите в поле Tip значение шкала – °C или °F и нажмите кнопку ОК.

5. Отобразите окно контекстной справки **Context Help**, которое доступно из пункта главного меню Help / Show Context Help.

6. Наведите курсор на один из объектов для просмотра описания их работы в окне **Context Help**.

Блок-диаграмма

7. Перейдите на блок-диаграмму, выбрав **Window / Show Diagram**.

8. Создайте блок-диаграмму, показанную на рис. 3.4.

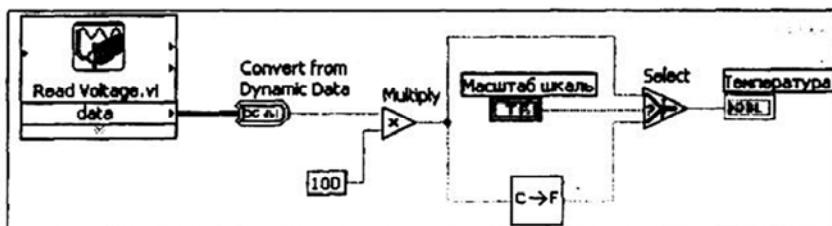


Рис. 3.4. Блок-диаграмма

Поместите на блок-диаграмму ВП Demo Read Voltage VI, расположенный в каталоге d:\netshare\Text\AICiT\for_LabVIEW, который служит для имитации считывания напряжения, пропорционального температуре. Например, если температура составляет 20 °С, то напряжение на выходе датчика будет равно 20 В.

 Поместите на блок-диаграмму ВП Convert from Dynamic Data (преобразовать динамические данные), расположенный в палитре Functions / Express / Signal Manipulation. Этот ВП преобразует динамический тип данных. В конфигурационном диалоговом окне выберите пункт Single Scalar in 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 (Создать константу). Константе присвойте значение «100» и нажмите клавишу <Enter>.

 В палитре Functions (Функций) в разделе Select a VI (Выбор ВП) выберите ВП Преобразование С в F, созданный в разделе 3.3.1. 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 °С или °F в зависимости от состояния переключателя Масштаб шкалы.

Лицевая панель

9. Перейдите на лицевую панель.
10. Нажмите на кнопку непрерывного запуска, показанную слева.
11. Нажмите на кнопку непрерывного запуска еще раз для остановки ВП.

Иконка и соединительная панель

12. Создайте иконку, показанную ниже на рис. 3.5, чтобы использовать ВП в качестве подпрограммы.

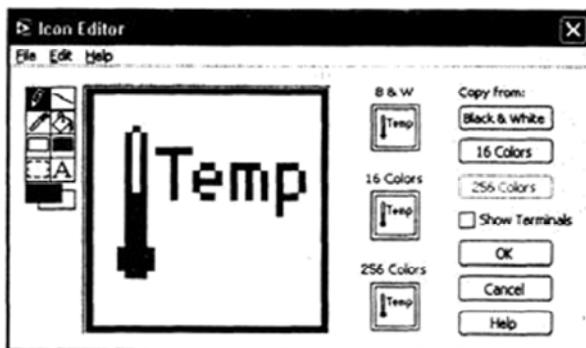


Рис. 3.5. Иконка для использования ВП в качестве подпрограммы

13. 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).

14. Сохраните ВП под именем «Термометр», он будет использоваться позднее.

15. Закройте ВП, выбрав пункт главного меню File / Close.

3.3.3. Задание. Создание ВП

По аналогии с разделом 3.3.2, создать ВП согласно своему варианту (табл. 3.1), используя в качестве подпрограммы прибор, созданный в задании 2.4.

Таблица 3.1

Создание ВП

№	Содержание задания
1	ВП измеряет температуру и отображает значение температуры по шкале Кельвина или в градусах Цельсия
2	ВП измеряет температуру и отображает значение температуры по шкале Реомюра или в градусах Цельсия
3	ВП измеряет температуру и отображает значение температуры в градусах Цельсия или по шкале Кельвина

№	Содержание задания
4	ВП измеряет температуру и отображает значение температуры в градусах Цельсия или по шкале Реомюра
5	ВП измеряет температуру и отображает значение температуры в градусах Цельсия или по шкале Фаренгейта
6	ВП измеряет температуру и отображает значение температуры по Реомюру или по шкале Фаренгейта
7	ВП измеряет температуру и отображает значение температуры в градусах Цельсия или по шкале Реомюра
8	ВП измеряет температуру и отображает значение температуры по Кельвину или по шкале Реомюра
9	ВП измеряет напряжение; на выходе – значение напряжения или силы тока (рассчитанной по закону Ома) в зависимости от состояния переключателя
10	ВП измеряет вязкость и отображает значение динамической вязкости μ или кинематической вязкости ν
11	ВП измеряет напряжение и отображает напряжения (мВ) или мощность (Вт)
12	ВП измеряет напряжение и отображает напряжение (В) или мощность (Вт)

Примечания:

1) для имитации считывания напряжения, пропорционального температуре, в вариантах 1–8 следует использовать ВП Demo Read Voltage VI, расположенный в каталоге d:\netshare\Text\AICiT\for_LabVIEW;

2) в вариантах 9, 11, 12 для имитации считывания напряжения – ВП Demo Read Voltage VI;

3) в варианте 10 для имитации считывания значений вязкости – функцию Random Number (0–1), расположенную в палитре Functions / Numeric. Эта функция будет генерировать случайное число в пределах от 0 до 1.

Контрольные вопросы

1. Из каких основных компонентов состоит Ваш ВП?
2. Что называется иконкой ВП?

3. Как создается иконка ВП?
4. Как отображаются иконки ВП и какие функции они выполняют?
5. Как редактируется иконка ВП?
6. Как вывести на экран значение измеренной температуры?
7. Зачем нужна функция Select?
8. Как осуществляется непрерывный пуск разработанного ВП?

4. МНОГОКРАТНЫЕ ПОВТОРЕНИЯ И ЦИКЛЫ ПРИ СОЗДАНИИ ВИРТУАЛЬНОГО ПРИБОРА (ВП) В СРЕДЕ LABVIEW

4.1. Цель работы

1. Изучение приемов использования терминала выходных данных цикла While (по условию).
2. Измерение температуры раз в секунду в течение одной минуты.
3. Использование сдвиговых регистров и узлов обратной связи для организации доступа к значениям на предыдущих итерациях цикла For (с фиксированным числом итераций).
4. Изучение организации доступа к значениям предыдущих итераций цикла.

4.2. Методические указания

Структуры являются графическим представлением операторов цикла и операторов Case (Вариант), 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), структуру Case (Вариант), структуру Sequence (‘сиквенс) (Последовательность), структуру Event (Событие), а также Formula Node (узел Формулы).

Цикл While (по условию) работает до тех пор, пока не выполнится логическое условие выхода из цикла.

 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.

 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).

Цикл For (с фиксированным числом итераций) выполняет повторяющиеся операции над потоком данных определенное количество раз.

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

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

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 (Добавить сдвиговый регистр).

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.

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 (Узел обратной связи)** в палитре **Structures (Структуры)**. 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.

Структура выбора Case. В структуре выбора Case имеются две или более встроенных блок-схемы. Выбор одной из них, которая будет выполнена, определяется в зависимости от значения, поданного на вход данной структуры. Структура Case включает:

1. Терминал выбора (?). Значение, подаваемое на него, может быть целым, логическим или строковым.

2. Переключатель блок-схем (True \ False \ и т. д.). Позволяет переходить от одной блок-схемы к другой. Содержит по умолчанию два окна True и False. При необходимости количество блок-схем выбора может быть увеличено. Кроме True и False в качестве значений переключателя могут использоваться целые числа или строковые значения.

Формульный блок Formula Node позволяет вводить формулы в обычном виде прямо в блок-схему. Особенно это удобно, когда выражение имеет много переменных и сложный вид. Формулы вводятся как простой текст. При этом создаются терминалы на границе блока (контекстное меню Add Input или Add Output), в которые вписываются имена переменных. Каждое выражение заканчивается разделителем «;».

Структура последовательности Sequence Structure выполняет встроенные в нее блок-схемы последовательно в определенном порядке. Количество встроенных блок-схем определяется числом фреймов данной структуры. Их количество добавляется при помощи контекстного меню – Add Frame After, Add Frame Before. Для передачи значений переменных из фрейма в фрейм используются локальные переменные структуры (контекстное меню – Add Sequence Local variable), создаваемые на границе фрейма. Данные, связанные с такой переменной, доступны во всех последующих фреймах и не доступны в предыдущих.

4.3. Задания

4.3.1. Задание 1. ВП. Измерение температуры во времени

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 создайте лицевую панель, как показано ниже на рис. 4.1.

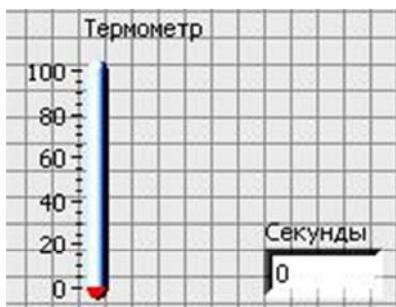


Рис. 4.1. Лицевая панель

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 на рис. 4.2.



Поместите ВП Термометр на блок-диаграмму. Для этого выберите Functions / Select a VI и укажите папку, в которой находится прибор, созданный в задании 3.4.

Щелкните правой кнопкой мыши по полю ввода данных Temp Scale (Шкала температур), 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. Функция Wait Until Next ms Multiple (ждать кратного значения) обеспечивает интервал между итерациями, равный интервалу времени, необходимому для того, чтобы миллисекундный счетчик достиг значения, кратного введенному пользователем. 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).

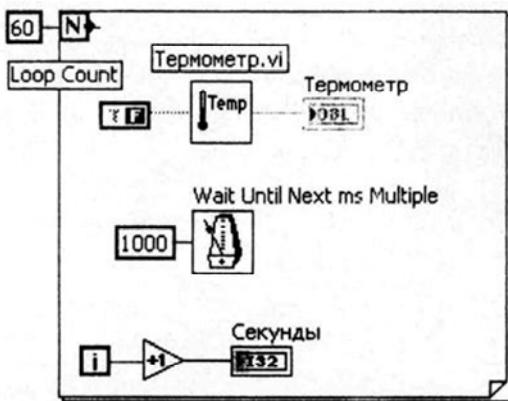


Рис. 4.2. Блок-диаграмма

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.
4. Запустите ВП.
5. Закройте ВП.

4.3.2. Задание 2. Создание ВП

Создать ВП согласно своему варианту (табл. 4.1).

Таблица 4.1

Создание ВП

№	Содержание задания
1	Создайте ВП, генерирующий случайные числа в цикле While. Организуйте выход из цикла по нажатию кнопки на лицевой панели ВП
2	Посчитайте значение выражения $Y = X^2 + Z^3 - XZ + 10$ с помощью блока Formula Node и ВП Formula Express, расположенного в палитре Functions / Arith/Compare (арифметика/сравнение)

№	Содержание задания
3	Создайте ВП, генерирующий 70 случайных чисел в цикле FOR
4	Создайте ВП, который с помощью Formula Node считает значение выражения $Y = x + \cos(x) - 10$ и если $Y \geq 0$, то $Z = \text{sqrt}(Y)$
5	Используя структуру Case, создайте ВП, который считает разность двух чисел; если полученное число ≥ 0 , то вычисляется значение корня, иначе выдается сообщение об ошибке
6	ВП осуществляет поочередное включение индикаторов на лицевой панели; промежутки между включениями индикаторов 2, 3 и 7 с, соответственно. Используйте последовательность Sequence Structure и функцию Time Delay, расположенную в палитре Functions / Time and Dialog
7	Создайте ВП, который измеряет температуру в течение минуты с помощью термометра, созданного в разделе 3.3.2, и считает среднее значение температуры. Подсчет среднего значения осуществить с помощью сдвигового регистра
8	Посчитайте значение выражения $Y = X^5 + \cos^2(Z) - XZ + 10$ с помощью блока Formula Node и ВП Formula Express, расположенного в палитре Functions / Arith/Compare (арифметика/сравнение)
9	Используя структуру Case, создайте ВП, который считает значение выражения $y = ax + 14$, где $a = \text{const}$; если $y \geq 0$, то вычисляется значение корня, иначе выдается сообщение об ошибке
10	Создайте ВП, который генерирует 70 случайных чисел и считает среднее значение. Подсчет среднего значения осуществите с помощью сдвигового регистра
11	Создайте ВП, который с помощью Formula Node считает значение выражения $y = \sin(x)$, если y – положительное число, то $z = y + A$, иначе $z = y - A$
12	ВП осуществляет поочередное включение индикаторов на лицевой панели; промежутки между включениями индикаторов 5, 8 и 12 с, соответственно. Используйте последовательность Sequence Structure и функцию Time Delay, расположенную в палитре Functions / Time and Dialog

Контрольные вопросы

1. Из каких основных компонентов состоит Ваш ВП?
2. Какие приемы использования цикла While Вы знаете?
3. Как измерить температуру с интервалом раз в секунду в течение одной минуты?
4. Как и зачем используются сдвиговые регистры в ВП?
5. Зачем нужны узлы обратной связи в ВП?
6. Как добавить 1 к значению счетчика итераций после завершения выполнения цикла?
7. Из каких подпалитр состоит палитра Controls (Элементы)?
8. Из каких подпалитр состоит палитра Functions (Функции)?
9. Назовите назначение управляющих кнопок на блок-диаграмме.
10. Назовите основные типы данных.

5. РАБОТА С МАССИВАМИ В СРЕДЕ LABVIEW

5.1. Цель работы

1. Изучить типовые приемы создания массива, элементов управления и отображения.
2. Изучить типовые приемы создания массива констант.
3. Изучить функции работы с массивами, полиморфные функции.
4. Освоить приемы работы с массивами.

5.2. Методические указания

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 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.

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.

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 (рис. 5.1). Затем поместить в шаблон массива элемент управления либо отображения данных.

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.

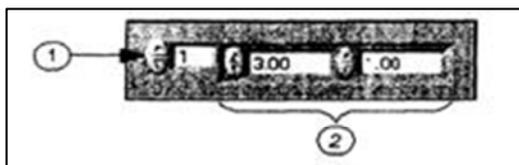


Рис. 5.1. Шаблон массива:

1 – элемент индекса массива; 2 – элементы значений массива

Двумерные массивы. В двумерном (2D) массиве элементы хранятся в виде матрицы. Таким образом, для размещения элемента требуется указание индекса столбца и строки. На рис. 5.2 показан двумерный массив, состоящий из 6 столбцов (длина) и 4 строк (высота). Количество элементов в массиве равно 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 (Добавить размер)**. С этой целью также можно использовать инструмент ПЕРЕМЕЩЕНИЕ. Для этого надо просто изменить размер элемента индекса.



Рис. 5.2. Двумерный массив

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 рис. 5.3 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** (запретить автоиндексацию). Автоиндексацию следует отключать, например, в случае, когда нужно знать только последнее значение.

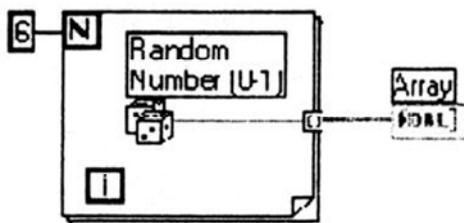


Рис. 5.3. Автоматическая индексация

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** (разрешить автоиндексацию).

Создание двумерных (2D) массивов. Для создания двумерных массивов необходимо использовать два цикла For, один внутри другого. Как показано на рис 5.4, внешний цикл создает элементы массива в строке, а внутренний цикл создает элементы массива в столбце.

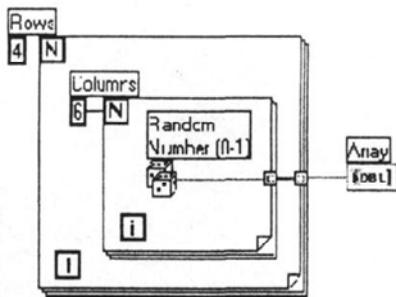


Рис. 5.4. Двумерный массив

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 (Размер массива) – 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 (задать массив) – 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 (создать массив) – 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** (объединение входных данных).

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



Array Subset (подмножество массива) – выдает часть массива, начиная с индекса, поступившего на поле **index**, и длиной, указанной в поле **length** (длина). Например, если подать предыдущий массив на поле ввода функции **Array Subset** значение 2 – на поле **index** и 3 – на поле Подмножество, то

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



Index Array (индекс массива) – 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 полмассива. 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
---	---	---

5.3. Задания

5.3.1. Задание 1. ВП. Работа с массивами

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:

- а) в палитре **Controls / Array & Cluster** выберите шаблон массива;
- б) созданному массиву присвойте имя «Массив случайных чисел»;
- в) поместите внутрь шаблона массива цифровой элемент отображения, расположенный в палитре **Controls / Numeric**;
- г) с помощью инструмента **ПЕРЕМЕЩЕНИЕ** измените размер массива таким образом, чтобы он содержал 10 элементов;
- д) нажмите и удерживайте клавишу **<Ctrl>** и, перемещая элемент «Массив случайных чисел», создайте две его копии;
- е) копиям присвойте имена «Конечный Массив» и «Подмножество Массива»;

ж) создайте три цифровых элемента управления и присвойте им имена «Масштабный коэффициент», «Старт подмножества»;

з) щелкните правой кнопкой мыши по элементам «Старт подмножества» и «Количество элементов подмножества», в контекстном меню выберите пункт Representation, затем пункт 132;

е) значения элементов управления данными пока не изменяйте.

Блок-диаграмма

2. Постройте блок-диаграмму, как показано ниже.

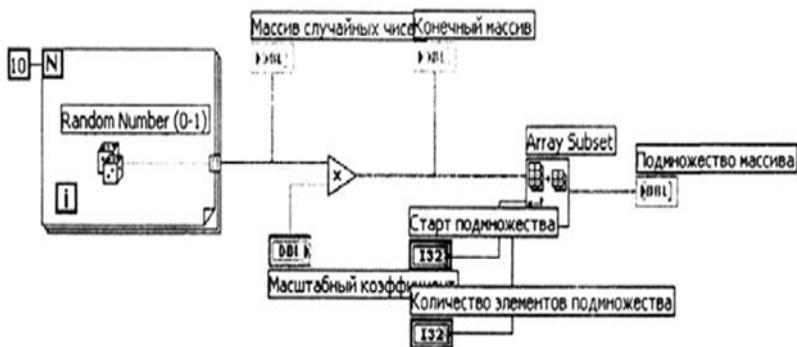


Рис. 5.5. Блок-диаграмма



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».

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.

5.3.2. Задание 2. Создание ВП

Создайте ВП согласно своему варианту (табл. 5.1).

Таблица 5.1

Создание ВП

№	Содержание задания
1	Создайте ВП, который полностью изменяет порядок элементов в массиве, содержащем 10 случайных чисел. Например, элемент массива с индексом 0 становится элементом массива с индексом 9, а элемент массива с индексом 1 становится элементом массива с индексом 8 и т. д. Для изменения порядка данных в массиве следует использовать функцию Reverse ID Array, расположенную на палитре Functions / Array
2	Создайте ВП, который генерирует двумерный массив случайных чисел, содержащий 3 строки и 10 столбцов
3	Создайте ВП, который генерирует одномерный массив, содержащий 80 случайных чисел, и выдает часть массива, начиная с индекса 15 до индекса 60. На лицевую панель вывести массив случайных чисел и полученный массив

№	Содержание задания
4	Создайте ВП, который генерирует одномерный массив случайных чисел до тех пор, пока не нажата кнопка на лицевой панели. На лицевую панель вывести полученный массив и его размерность
5	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}. Используйте функцию Decimate ID Array, расположенную в палитре Functions / Array
6	Создайте ВП, который генерирует одномерный массив случайных чисел и сортирует полученный массив в порядке возрастания. На лицевую панель вывести массив случайных чисел и отсортированный массив. Для сортировки элементов в массиве следует использовать функцию Sort 1D Array, расположенную на палитре Functions / Array
7	Создайте ВП, который генерирует одномерный массив случайных чисел и выводит максимальное значение полученного массива и его порядковый номер. Использовать функцию Array Max & Min, расположенную на палитре Functions / Array
8	Создайте ВП, который генерирует одномерный массив случайных чисел и выводит минимальное значение полученного массива и его порядковый номер. Использовать функцию Array Max & Min, расположенную на палитре Functions / Array
9	Создайте ВП, который генерирует двумерный массив случайных чисел, содержащий 4 строки и 5 столбцов
10	Создайте ВП, который генерирует два одномерных массива случайных чисел и объединяет эти массивы в двумерный массив чисел. На лицевую панель вывести два исходных массива случайных чисел и двумерный массив, состоящий из элементов исходных массивов

№	Содержание задания
11	Создайте ВП, который генерирует двумерный массив случайных чисел размерностью 5×6 и выдает часть этого массива размерностью 4×5 . На лицевую панель вывести исходный массив случайных чисел и полученный массив
12	Создайте ВП, который генерирует двумерный массив случайных чисел и осуществляет транспонирование полученного массива. На лицевую панель вывести массив случайных чисел и транспонированный массив. Для транспонирования массива используйте функцию <code>Transpose 2D Array</code> , расположенную на палитре <code>Functions / Array</code>

Контрольные вопросы

1. Из каких основных компонентов состоит Ваш ВП?
2. Какие типовые приемы создания массива констант Вы знаете?
3. Какие функции работы с массивами Вы знаете?
4. Что такое полиморфные функции?
5. Какие приемы работы с массивами доступны для лицевой панели?
6. Какие палитры доступны для блок-диаграммы?
7. Что Вы знаете о цикле `For`?
8. Как создаются многомерные массивы?
9. Каково назначение элемента «Старт подмножества» в подпрограмме ВП?
10. Как выбирается количество элементов подмножества в подпрограмме ВП?
11. Какая функция генерирует случайное число в пределах от 0 до 1?
12. Какие функции создания массивов Вы знаете?

6. СОЗДАНИЕ КЛАСТЕРОВ ИЗ ЭЛЕМЕНТОВ УПРАВЛЕНИЯ И ОТОБРАЖЕНИЯ ДАННЫХ. РАБОТА С КЛАСТЕРАМИ. МАСШТАБИРОВАНИЕ КЛАСТЕРА

6.1. Цель работы

1. Изучение типовых приемов создания кластеров и функций отображения кластеров.
2. Создание кластеров на лицевой панели.
3. Сборка и демонтаж кластеров с использованием функций обработки кластеров.
4. Создание ВП, использующего полиморфизм в кластерах.
5. Изучение кластеров ошибок.

6.2. Методические указания

Как и массив, кластер является структурой, группирующей данные. Однако в отличие от массива кластер может группировать данные различных типов (логические, числовые и т. д.). 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.

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 (рис. 6.1).

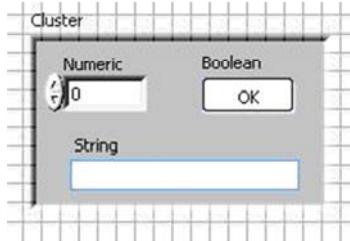


Рис. 6.1. Кластер

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 т. д. 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 на рис 6.2.

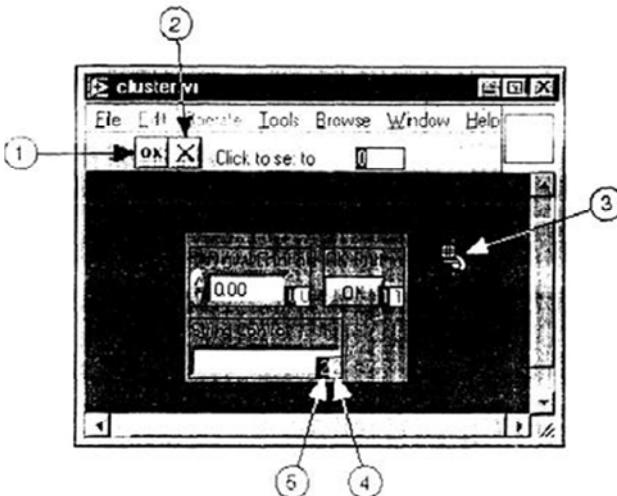


Рис. 6.2. Панель инструментов и кластер

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 **ОК (подтвердить)** на панели инструментов.

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.

Функции работы с кластерами. Для создания и управления кластерами используются функции, расположенные на палитре **Functions / Cluster**. Функции **Bundle (Связать)** и **Bundle by Name (Связать по названию)** используются для сборки и управления кластерами. Функции **Unbundle (Разделить)** и **Unbundle by Name (Разделить по названию)** 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 (Инструменты кластеров)**. 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.

Иногда удобно поменять массивы на кластеры и наоборот, поскольку LabVIEW включает в себя намного больше функций, работающих с массивами, чем с кластерами. Для преобразования кластера в массив служит функция **Кластер в массив (Cluster to Array)**. Обратная операция осуществляется с помощью функции **Массив в кластер (Array to Cluster)**. Функция **Кластер в массив** конвертирует кластер с количеством элементов N одного типа данных в массив с количеством элементов N того же типа данных. Индекс массива соответствует порядковому номеру в кластере (т. е. нулевой элемент кластера становится значением массива с индексом 0). Следует обратить внимание, что при использовании этой функции все элементы в кластере должны быть одного типа.

Функция **Массив в кластер** преобразует одномерный массив с числом элементов N в кластер с числом элементов N того же типа данных. Для включения этой функции необходимо щелкнуть правой

кнопкой мыши по терминалу **Массив в Кластер** и выбрать опцию **Размер кластера (Cluster Size)** для установления размера выходного кластера, поскольку кластеры, в отличие от массивов, не устанавливают свой размер автоматически. Размер кластера по умолчанию равен 9. Если массив имеет меньшее количество элементов, чем это определено размером кластера, LabVIEW автоматически создаст дополнительные элементы кластера со значениями по умолчанию для типа данных кластера. Однако, если количество элементов входного массива больше величины, установленной в окне размера кластера, то проводник блок-диаграммы, идущий к выходному кластеру, будет разорванным, пока не будет отрегулирован его размер.

6.3. Задания

6.3.1. Задание 1. ВП. Работа с кластерами

Лицевая панель

1. Откройте новый ВП и создайте лицевую панель, как показано на рис. 6.3.

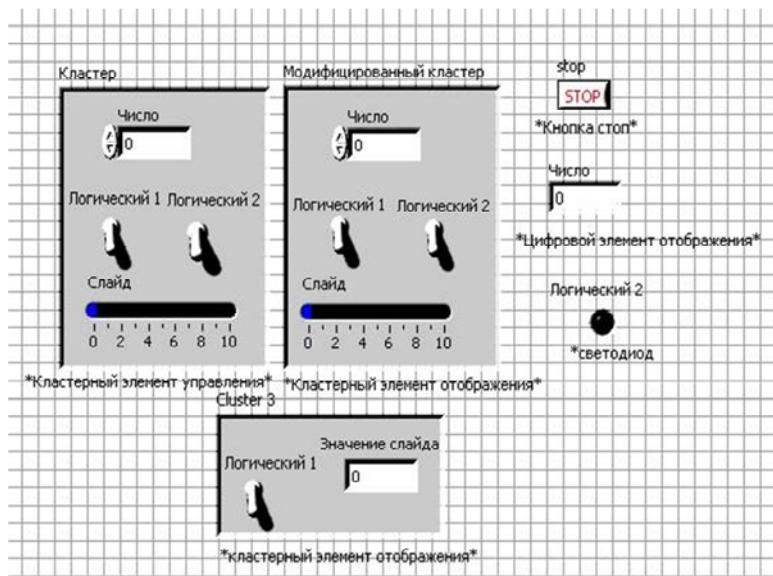


Рис. 6.3. Лицевая панель

а) поместите на лицевую панель кнопку «Стоп» и круглый светодиод;

б) из палитры Controls / Array & Cluster выберите шаблон кластера;

в) объекты лицевой панели, показанные на иллюстрации, поместите в шаблон кластера;

г) 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;

д) повторите пункт г) для создания элемента «Маленький кластер». Измените его, как показано на рис. 6.3.

2. Проверьте порядковые номера элементов в кластерах «Кластер» и «Маленький кластер». Порядковые номера элементов кластеров «Модифицированный кластер» и «Кластер» должны совпадать:

а) щелкните правой кнопкой мыши по границе шаблона каждого кластера, из контекстного меню выберите пункт Reorder Controls in Cluster;

б) порядковые номера элементов установите, как показано ниже на рис. 6.4.

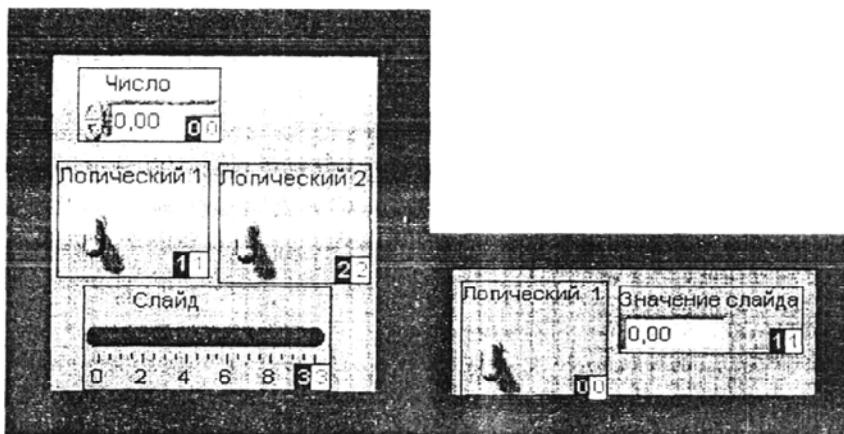


Рис. 6.4. Лицевая панель

Блок-диаграмма

3. Создайте блок-диаграмму, как показано на рис. 6.5.



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 добления полей ввода данных.

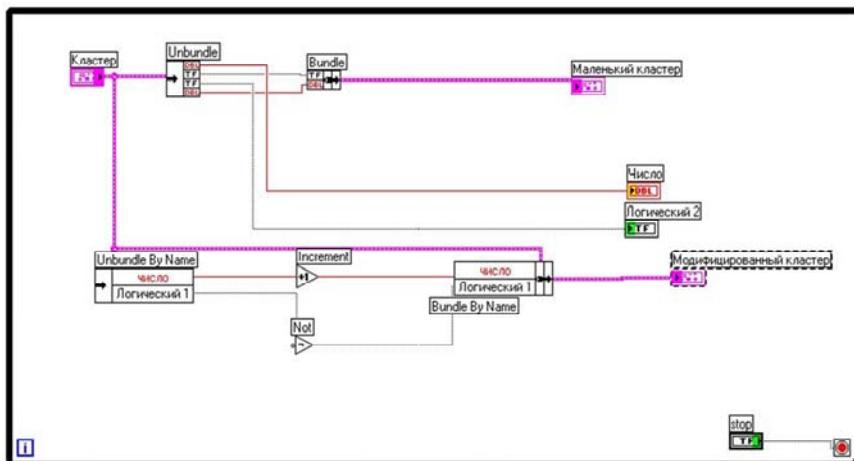


Рис. 6.5. Блок-диаграмма



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

Контрольные вопросы

1. Из каких основных компонентов состоит Ваш ВП?
2. Что понимается под термином Кластер?
3. Какие типовые приемы создания кластеров Вы знаете?
4. Какие функции отображения кластеров Вы знаете?
5. Как создать кластер на лицевой панели?
6. Как собираются и демонтируются кластеры?
7. Какие функции обработки кластеров Вы знаете?
8. Что такое полиморфизм в кластерах?
9. Как создать модифицированный кластер?
10. Каково основное отличие кластера от массива?
11. Каков порядок размещения элементов в кластере?
12. Как посмотреть и изменить порядковый номер объекта, помещенного в кластер?
13. Как изменить количество полей ввода/вывода в кластере?
14. Как устанавливают размер кластера?
15. Каков размер кластера по умолчанию?
16. Как создать и переименовать копию элемента Кластер?

СПИСОК ЛИТЕРАТУРЫ

1. Трефис, Дж. LabVIEW для всех / Дж. Тревис; пер. с англ. Н. А. Клушин. – Москва: ДМК Пресс; ПриборКомплект, 2005. – 544 с.
2. Использование виртуальных инструментов LabVIEW / под ред. К. С. Демирчяна и В. Г. Миронова. – Москва: Солон-Р; Радио и связь, Горячая линия. – Телеком, 1999. – 268 с.
3. Учебный курс LabVIEW Express. Базовый курс 1. – Москва: National Instruments Corporation, 2003. – 354 с. – Режим доступа: ni.com; WWW.labview.ru.
4. Комментарии к LabVIEW 7.0. National Instruments Corporation. Режим доступа: ni.com; WWW.labview.ru.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1. Изучение основных понятий программной среды labview и виртуального прибора (ВП)	4
2. Создание, редактирование и отладка виртуального прибора (ВП)	14
3. Создание подпрограмм виртуального прибора (ВП)	20
4. Многократные повторения и циклы при создании виртуального прибора (ВП) в среде LabVIEW	30
5. Работа с массивами в среде LabVIEW	37
6. Создание кластеров из элементов управления и отображения данных. Работа с кластерами. Масштабирование кластера	47
СПИСОК ЛИТЕРАТУРЫ	54

Учебное издание

ОКОЛОВ Андрей Ромуальдович
ДРОЗД Анастасия Витальевна
ГУТИЧ Ирина Ивановна

**СИСТЕМЫ
АВТОМАТИЗИРОВАННОГО
КОНТРОЛЯ**

Учебно-методическое пособие
для студентов специальности 1-53 01 01 «Автоматизация
технологических процессов и производств»

Редактор *Ю. В. Ходочинская*
Компьютерная верстка *Е. А. Беспанской*

Подписано в печать 02.10.2018. Формат 60×84 ¹/₁₆. Бумага офсетная. Ризография.
Усл. печ. л. 3,26. Уч.-изд. л. 2,55. Тираж 100. Заказ 311.

Издатель и полиграфическое исполнение: Белорусский национальный технический университет.
Свидетельство о государственной регистрации издателя, изготовителя, распространителя
печатных изданий № 1/173 от 12.02.2014. Пр. Независимости, 65. 220013, г. Минск.