Некоммерческое акционерное общество

АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

Кафедра автоматизации и управления

ГРАФИЧЕСКОЕ ПРОГРАММИРОВАНИЕ В ЗАДАЧАХ ИЗМЕРЕНИЯ И АВТОМАТИЗАЦИИ

Методические указания по выполнению лабораторных работ
для студентов специальности
5В070200 – Автоматизация и управление

Алматы 2018

СОСТАВИТЕЛЬ: С.Г. Хан. Графическое программирование в задачах измерения и автоматизации. Методические указания к выполнению лабораторных работ для студентов специальности 5В070200 – Автоматизация и управление. - Алматы: АУЭС, 2018. - 94 с.

Методические указания содержат описания к 14 лабораторным работам и посвящены изучению среды графического программирования LabView для разработки виртуальных измерительных приборов в задачах измерений и автоматизации.

Методические указания используются студентами 2 курса специальности «Автоматизация и управление» при выполнении лабораторных работ по дисциплине «Графическое программирование в задачах измерения и автоматизации», описывают организацию среды графического программирования LabView.

Ил. 102, библиогр.- 5 назв.

Рецензент: доцент кафедры ТКСиС Ю.М. Гармашова

Печатается по плану издания некоммерческого акционерного общества «Алматинский университет энергетики и связи» на 2018 г.

©НАО «Алматинский университет энергетики и связи», 2018 г.

Содержание

Введение ……..….. 4
1 Лабораторная работа №1. Структурные элементы программной среды LabView……..... 5
2 Лабораторная работа №2. Подпрограммы в среде графического программирования LabView…… 14
3 Лабораторная работа №3. Многократные повторения и циклы….. 22
4 Лабораторная работа №4. Организация доступа к значениям предыдущих итераций цикла…….. 29
5 Лабораторная работа №5. Работа с массивами……..... 32
6 Лабораторная работа №6. Программирование кластеров……..... 38
7 Лабораторная работа №7. Основы графического отображения данных……..... 45
8 Лабораторная работа №8. Графическое отображение данных при разработке виртуальных приборов……..... 52
9 Лабораторная работа №9. Работа со структурами. Принятие решений в ВП…….. 55
10 Лабораторная работа №10. Работа со структурами. Узел формул……. 65
11 Лабораторная работа №11. Работа с последовательной структурой «Sequence Structure»…….. 70
12 Лабораторная работа №12. Локальные и глобальные переменные.. 76
13 Лабораторная работа №13. Функции обработки строк……..... 82
14 Лабораторная работа №14. Файловый ввод/ вывод…..……. 86
Список литературы…….. 94

Введение

Дисциплина «Графическое программирование в задачах измерения и автоматизации» изучается студентами специальности «Автоматизация и управление» на 2 курсе в пакете базовых дисциплин (компонент по выбору), объем 2 кредита. Знания материала данной дисциплины для будущих специалистов (бакалавров) технического профиля, связанных с разработкой или обслуживанием различных контрольно-измерительных приборов и средств автоматизации, на наш взгляд является обязательными.

Предлагаемые методические указания к выполнению лабораторных работ содержат четырнадцать работ. В соответствии с рабочей программой объем лабораторных часов составляет 30 часов. Тематика лабораторных работ посвящена практическому изучению среды графического программирования LabView для разработки виртуальных измерительных приборов в задачах измерений и автоматизации.

Программа, написанная в среде LabView, называется виртуальным прибором (ВП). ВП симулирует реальные физические приборы, например, вольтметр, мультиметр, генератор сигналов. LabView содержит полный набор инструментов для сбора, анализа, представления и хранения данных. С помощью LabView можно разрабатывать программно-аппаратные комплексы для тестирования, измерения, ввода данных, анализа и управления внешним оборудованием [1,2].

В LabView интерфейс пользователя – фронт (лицевая)-панель (Front panel) – создается с помощью элементов управления (элементы ввода: переключатели, кнопки, движки, ползунки и др.) и элементов отображения (визуализации: графики, светодиоды, цифровые табло и др.). После этого код программы (программирование) создается на блок-диаграмме ВП (Block diagram) с помощью блоков-функций для управления объектами на фронт панели [3,4].

Список необходимой литературы приведен в конце методических указаний.

Отчеты по лабораторным работам должны быть выполнены и оформлены в соответствии с требованиями фирменного стандарта Алматинского университета энергетики и связи СТ НАО 56023-1910-04-2014 «Учебно-методические и учебные работы. Общие требования к построению, изложению, оформлению и содержанию учебно-методических и учебных работ».

1 Лабораторная работа №1. Структурные элементы программной среды LabView

Цель работы: изучение структурных элементов, типов данных; выполнение арифметических, логических и строковых операций в LabView.

1.1 Задание к лабораторной работе

Задание 1. Создать виртуальный прибор (ВП) для выполнения простейших арифметических операций с визуализацией на индикаторе виртуального прибора.

Задание 2. Создать ВП для выполнения логических операций с визуализацией на LED-индикаторе.

Задание 3. Создать ВП для выполнения операций со строками.

1.2 Порядок выполнения работы

1.2.1 Для выполнения лабораторной работы необходимо зарегистрироваться в системе:

- загрузить файл Base_LabView, появится окно выбора лабораторных работ (рисунок 1.1), в главном меню LabView нажать кнопку «RUN»;

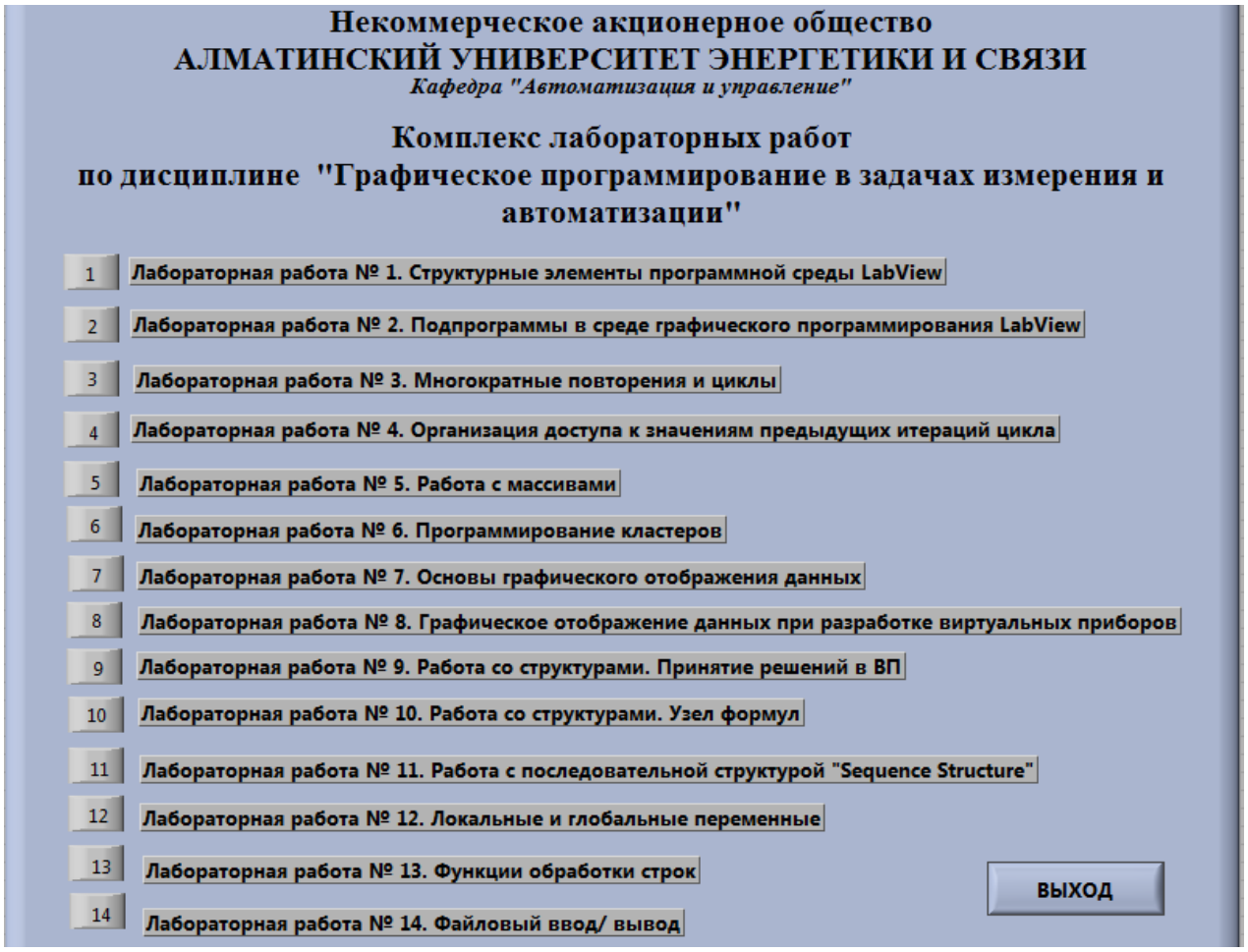


Рисунок 1.1 – Главное меню программы

– выбрать лабораторную работу «Структурные элементы программной среды LabView»;

– появится окно регистрации студента (рисунок 1.2): ввести свои Фамилию, Имя; ввести номер группы; нажать кнопку «Далее».

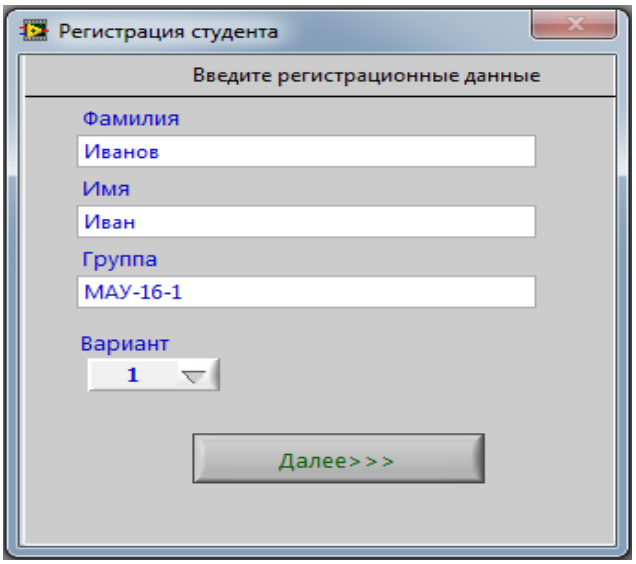


Рисунок 1.2 – Окно регистрации студента

1.2.2 В появившемся окне выбрать в главном меню «File - New VI» - «Новый виртуальный прибор (ВП)» [5]. Для одновременной визуализации двух панелей на экране нажмите «Ctrl-T» (рисунок 1.3):

- «Front panel» - «Фронт-панель» - интерфейс пользователя;

- «Block-diagram» - «Блок-диаграмма» - интерфейс кода программы в LabView.

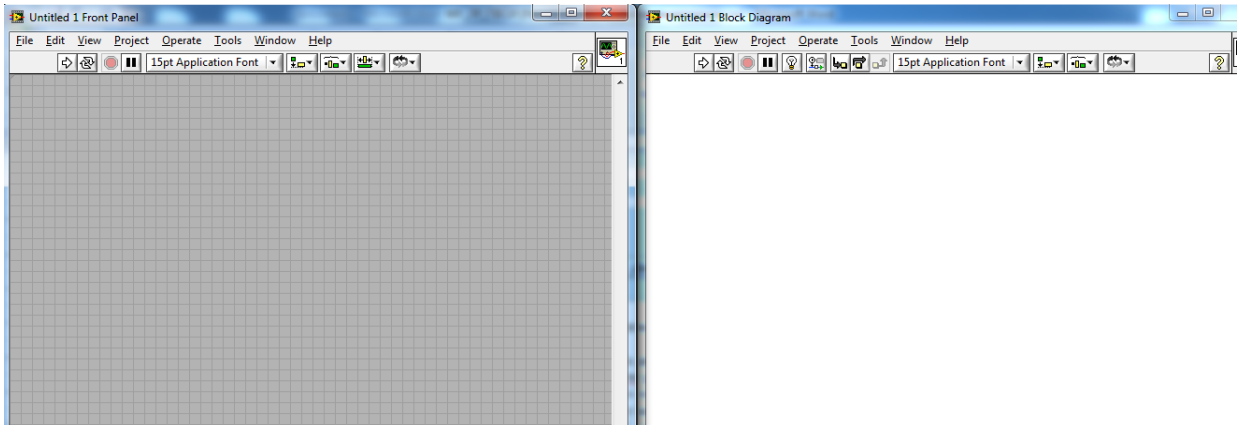


Рисунок 1.3

1.2.3 При создании ВП в LabView рекомендуется начать с организации интерфейса пользователя, помещая на «Front panel» элементы управления из

меню «Controls», для чего нажать в любом месте «Front-panel» на правую кнопку мыши.

1.2.4 Задание 1. Создать ВП для выполнения простейших арифметических операций с визуализацией на индикаторе виртуального прибора.

1.2.4.1 Из меню «Controls» - «Numeric Controls» выбрать и перетащить на интерфейс пользователя (рисунок 1.4) следующие элементы:

- числовой элемент 1 «Num Ctrl» для ввода числовых данных;
- числовой элемент 2 «Num Ctrl» для ввода числовых данных (можно скопировать предыдущий элемент).

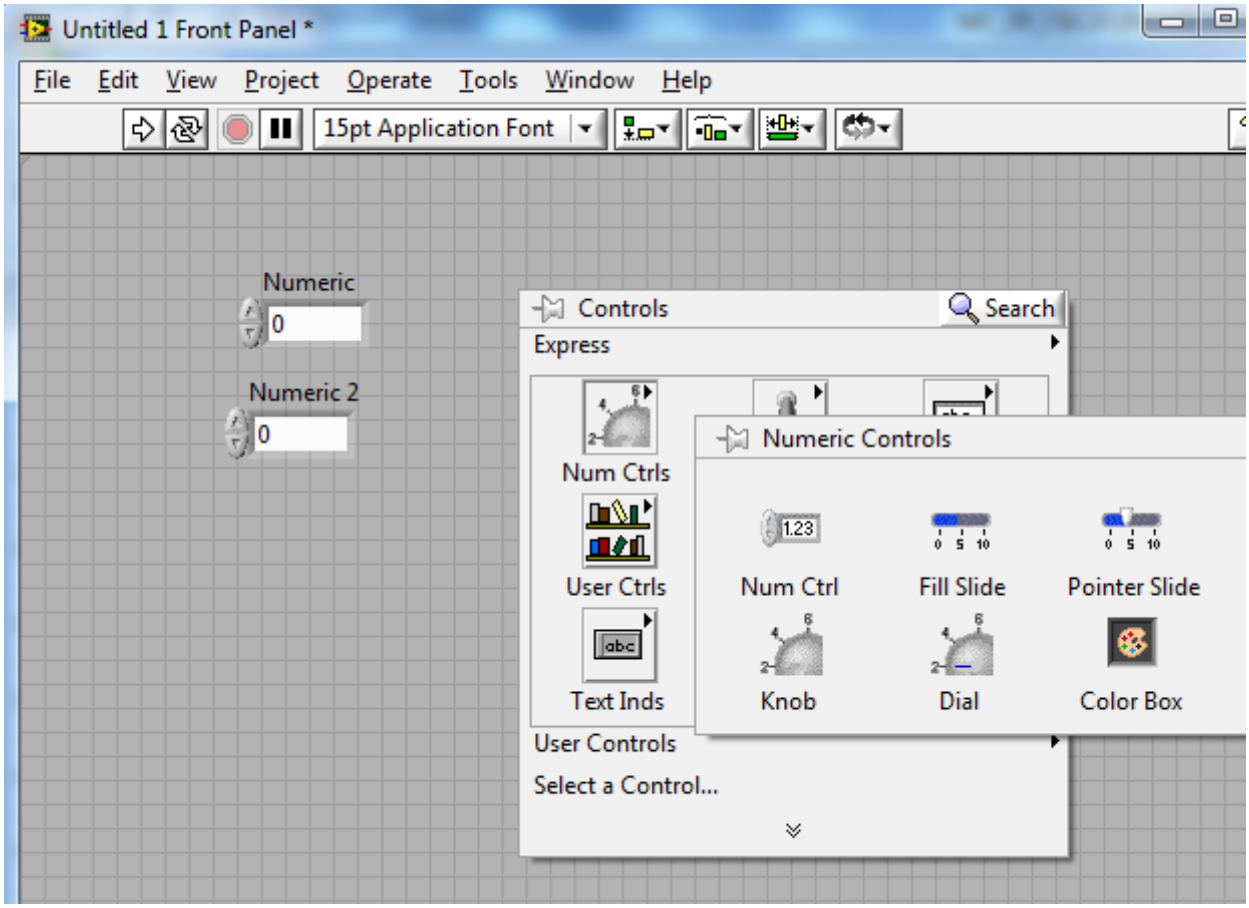


Рисунок 1.4

1.2.4.2 Из меню «Controls» - «Numeric Indicators» выбрать и перетащить на интерфейс пользователя (рисунок 1.5):

- числовой элемент 3 «Num Ind» для вывода числовых данных;
- элемент-индикатор «Gauge» для вывода числовых данных;
- элемент-индикатор «Meter» для вывода числовых данных;

1.2.4.3 При выборе элементов управления на интерфейсе пользователя одновременно появляются их значки на блок-диаграмме. Числовые элементы в Labview отмечаются оранжевым цветом, таким же, как и линии соединения блоков «Numeric» – потоки данных.

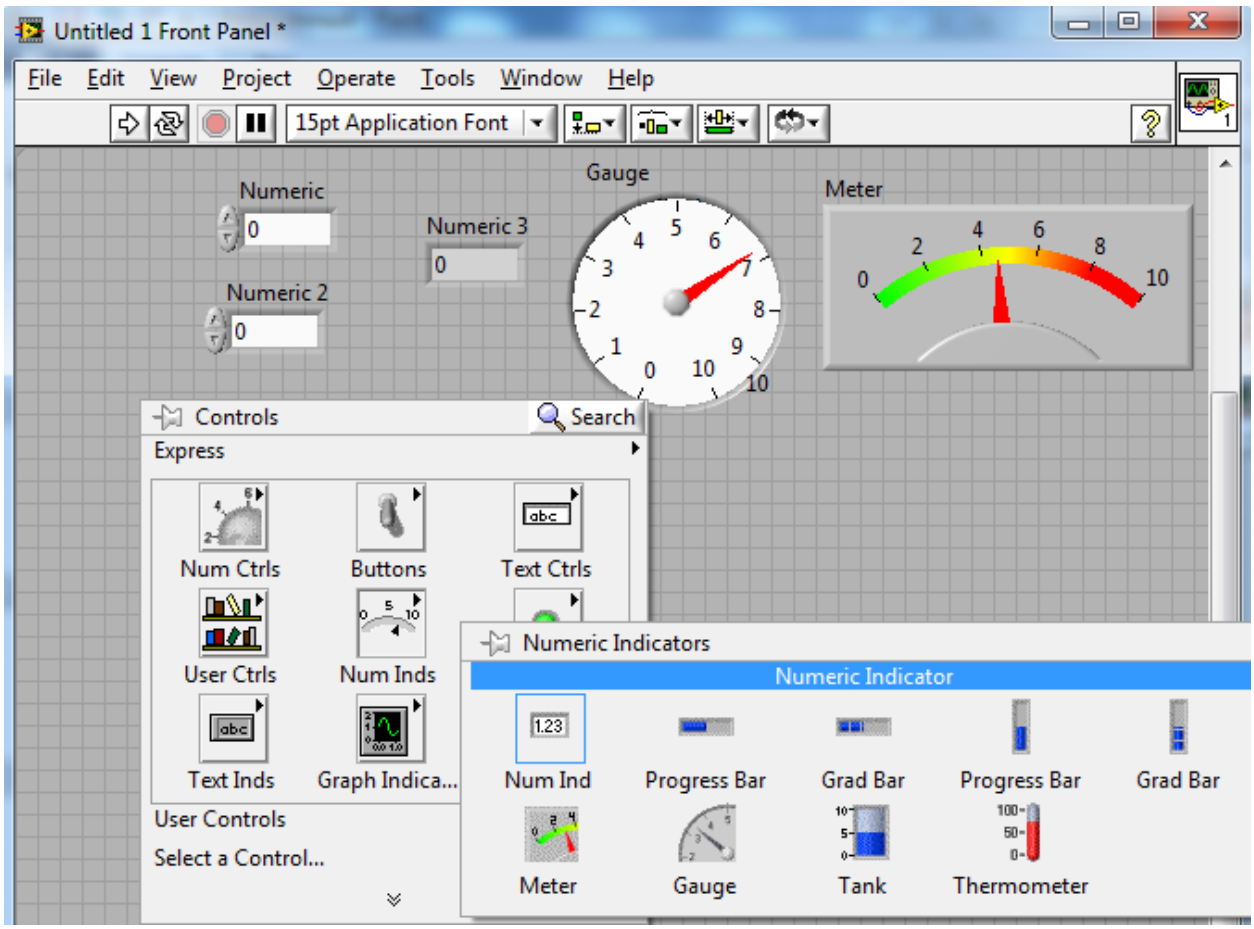


Рисунок 1.5

На блок-диаграмме собирается код программы (рисунок 1.6):

- из меню «Functions-Programming-Numeric» выбрать соответствующие функции арифметических операций, например, функцию сложения – «Add»;

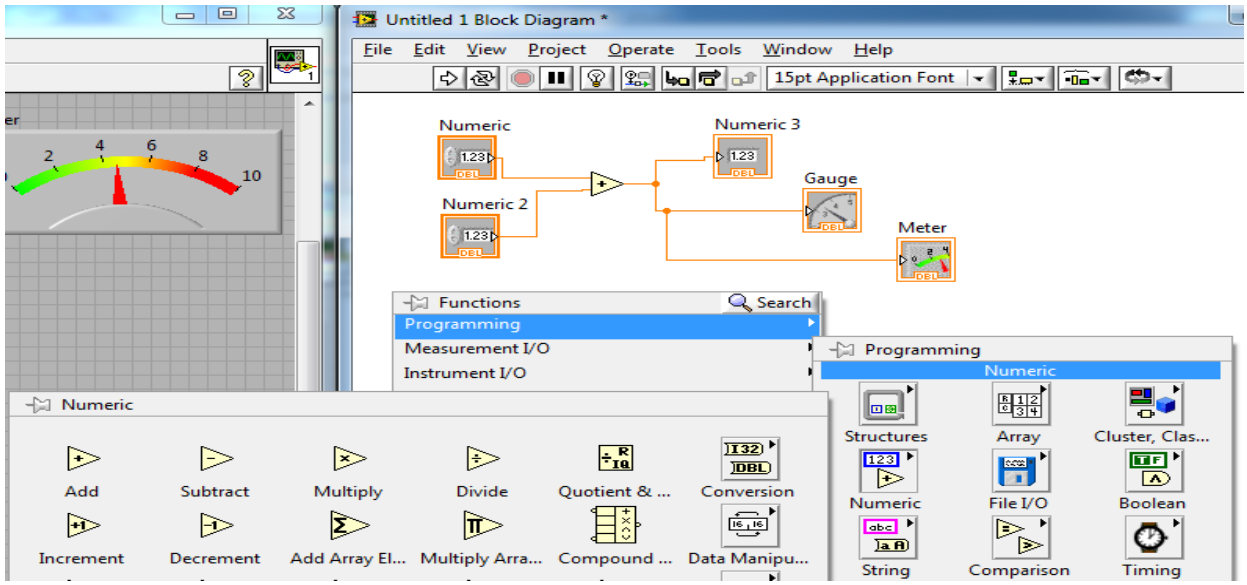


Рисунок 1.6

- соединить блоки потоками данных, при этом, наведя мышь на соответствующий вход или выход блока, появляется катушка-проводник, которая тянет за собой соединительный провод оранжевого цвета.

1.2.4.4 Сохранить программу под именем «Numeric» с расширением «.vi». Запустить программу на выполнение (рисунок 1.7):

- ввести исходные данные в числовые элементы «Numeric» и «Numeric2» с помощью стрелок-переключателей слева от окошка элемента, либо непосредственно ввести число в окошко элемента;

- в главном меню на фронт-панели (или на блок-диаграмме) нажать кнопку «Run», изображенную в виде стрелки и запускающую однократное выполнение задания;

- элементы-индикаторы покажут результат сложения.

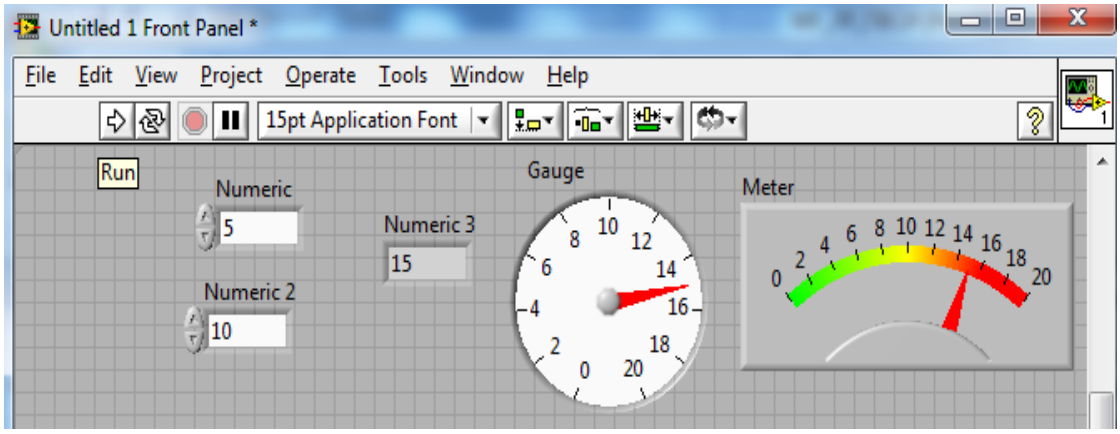


Рисунок 1.7

1.2.5 Задание 2. Создать ВП для выполнения логических операций с визуализацией на LED-индикаторе.

В главном меню LabView выбрать «File - New VI» - новый ВП.

1.2.5.1 Из меню «Controls» - «Buttons & Switches» выбрать и перетащить на интерфейс пользователя следующие элементы (рисунок 1.8):

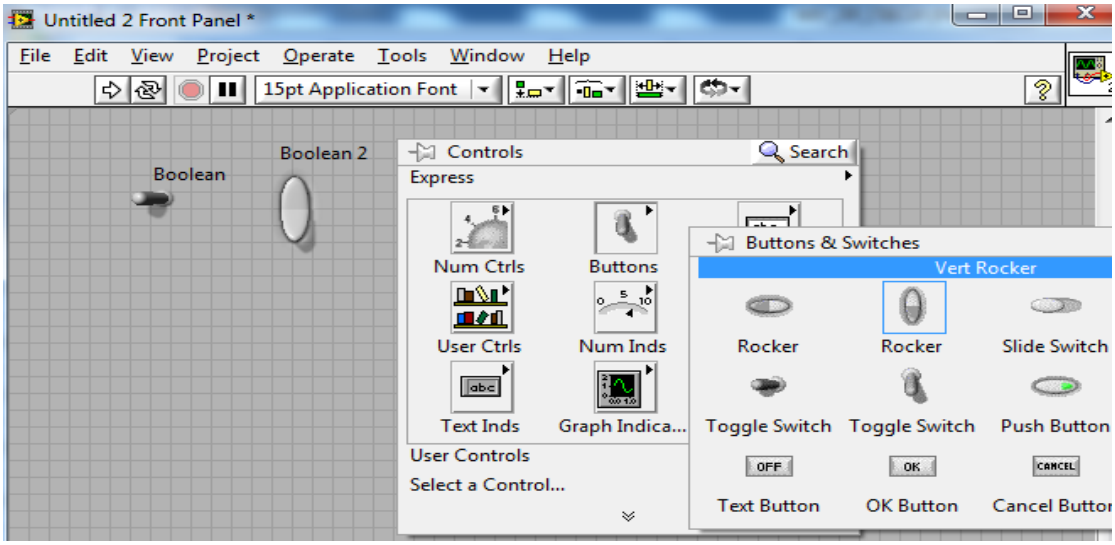


Рисунок 1.8

- логический элемент управления «Toggle Switch» для ввода логических данных Boolean;

- логический элемент управления «Rocker» для ввода логических данных Boolean2;

1.2.5.2 Из меню «Controls» - «Leds» выбрать и перетащить на интерфейс пользователя (рисунок 1.9) логический элемент-индикатор «Square Led» для вывода логических данных Boolean 3.

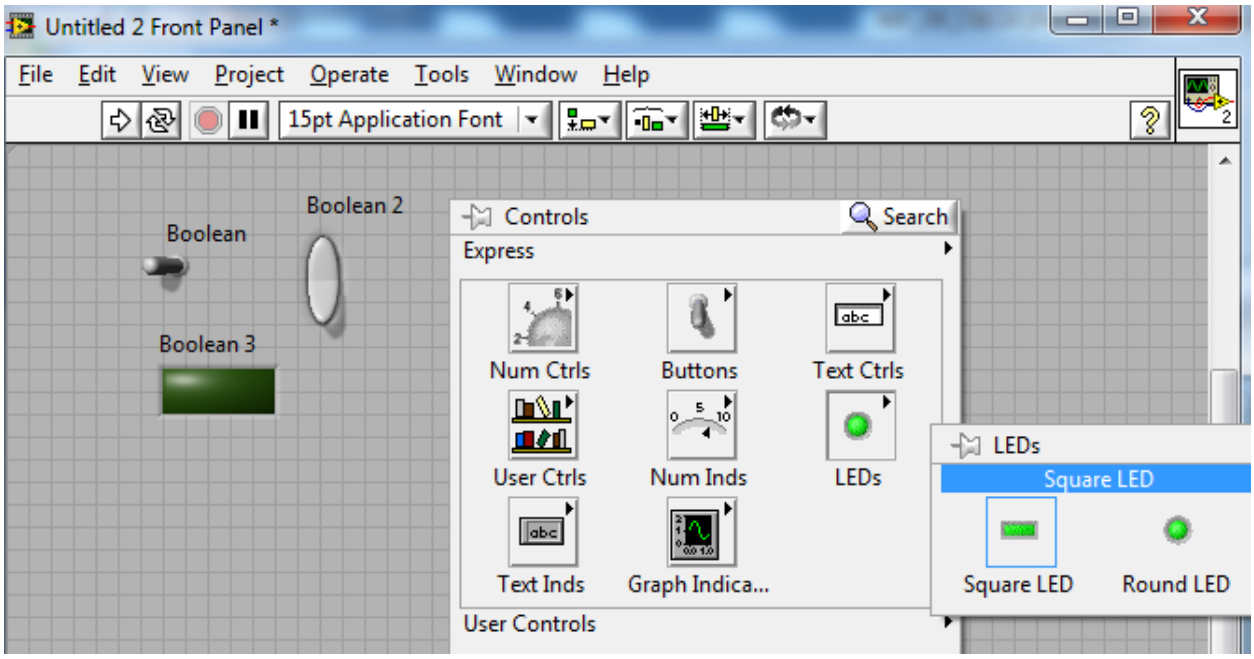


Рисунок 1.9

1.2.5.3 При выборе элементов управления на интерфейсе пользователя одновременно появляются их значки на блок-диаграмме. Логические элементы в Labview отмечаются зеленым цветом, также как и линии соединения блоков «Boolean» – потоки данных.

На блок-диаграмме собирается код программы (рисунок 1.10):

- из меню «Functions-Programming-Boolean» выбрать соответствующие функции логических операций, например, функцию «И» – «And»;

- соединить блоки потоками данных, при этом, наведя мышь на соответствующий вход или выход блока, появляется катушка-проводник, которая тянет за собой соединительный провод зеленого цвета.

1.2.5.4 Сохранить программу под именем «Boolean» с расширением «.vi». Запустить программу на выполнение (рисунок 1.11):

- ввести исходные данные в логические элементы «Boolean» и «Boolean 2», для этого включить (или выключить) эти переключатели;

- в главном меню на фронт-панели (или на блок-диаграмме) нажать кнопку «Running Continuously», изображенную в виде замкнутых двух стрелок и означающую *многократное (циклическое) выполнение*. В данном режиме выполнения задания фронт-панель изменит внешний вид: интерфейс без сетки;

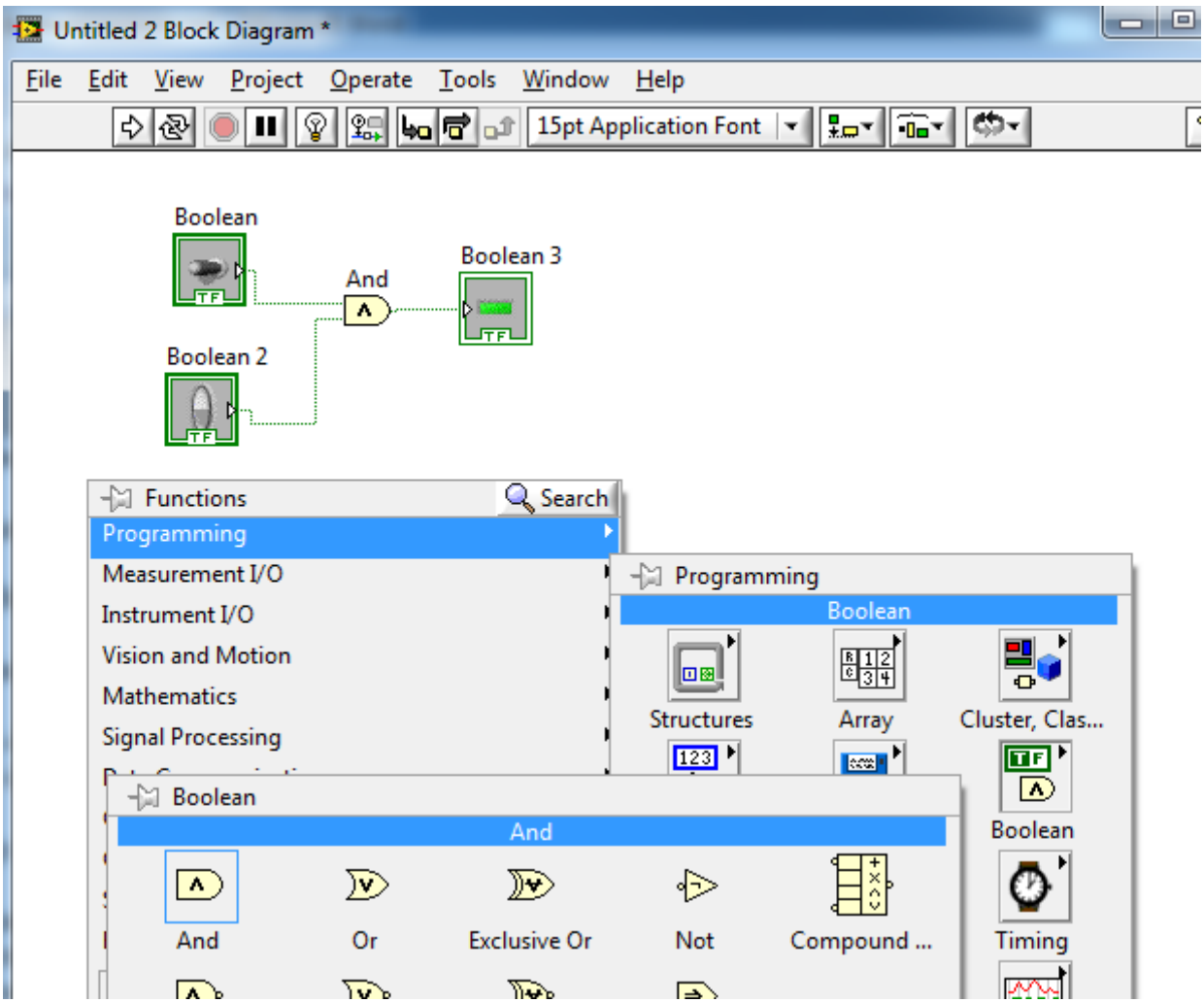


Рисунок 1.10

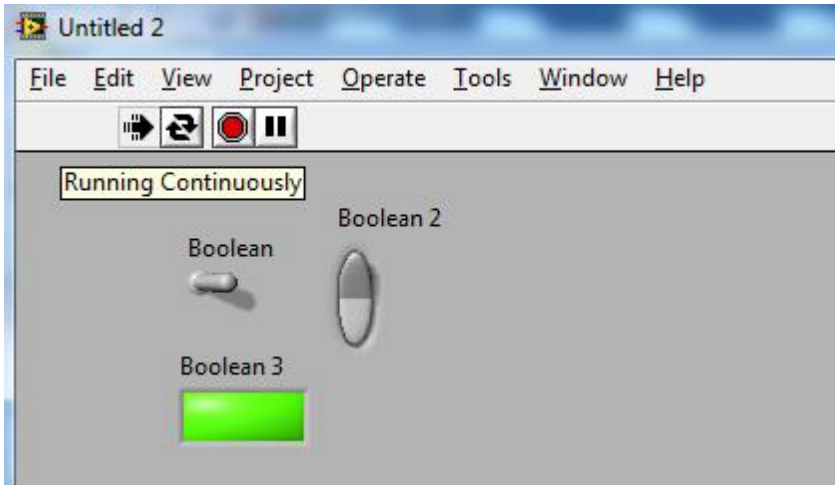


Рисунок 1.11

- элемент-индикатор зажжется при обоих включенных входных элементах (логическая единица – True). Попробовать различные комбинации логики: включение («1») – выключение («0») входных элементов;

- для выхода из циклического режима выполнения задания нажать в главном меню красную кнопку «Abort» - «Останов».

1.2.6 Задание 3. Создать ВП для выполнения операций со строками.

В главном меню LabView выбрать «File - New VI» - новый ВП.

1.2.6.1 Из меню «Contrls» - «Text Contrls» выбрать и перетащить на интерфейс пользователя (рисунок 1.12):

- строковый элемент 1 «String Ctrl» для ввода строковых данных;

- строковый элемент 2 «String Ctrl» для ввода строковых данных (можно скопировать предыдущий элемент).

1.2.6.2 Из меню «Controls» - «Text Indicators» выбрать и перетащить на интерфейс пользователя строковый элемент 3 «String Ind» для вывода строковых данных (рисунок 1.13).

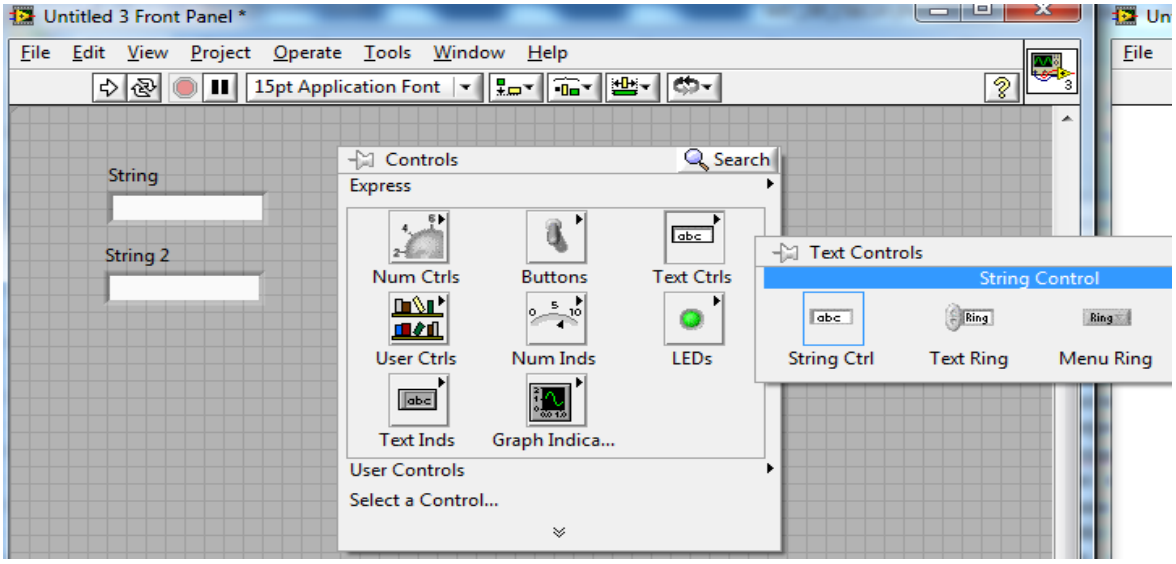


Рисунок 1.12

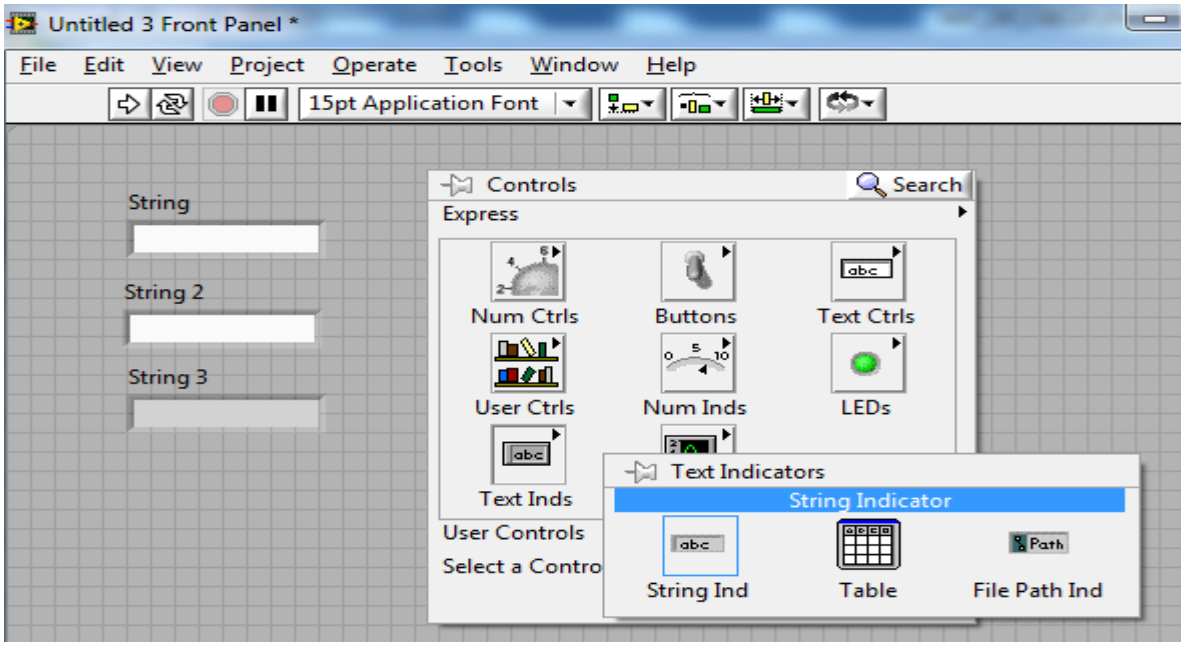


Рисунок 1.13

1.2.6.3 При выборе элементов управления на интерфейсе пользователя одновременно появляются их значки на блок-диаграмме. Строковые элементы в Labview отмечаются малиновым цветом, также как и линии соединения блоков «String» – потоки данных.

На блок-диаграмме собирается код программы (рисунок 1.14):

- из меню «Functions-Programming-String» выбрать соответствующие функции строковых операций, например, функцию связывания (добавления) одной строки с другой – «Concatenate Strings». Для получения более полной информации о любой функции LabView можно вызвать Помощь - Context Help, нажав в главном меню «Help»;

- соединить блоки потоками данных, при этом, наведя мышь на соответствующий вход или выход блока, появляется катушка-проводник, которая тянет за собой соединительный провод малинового цвета.

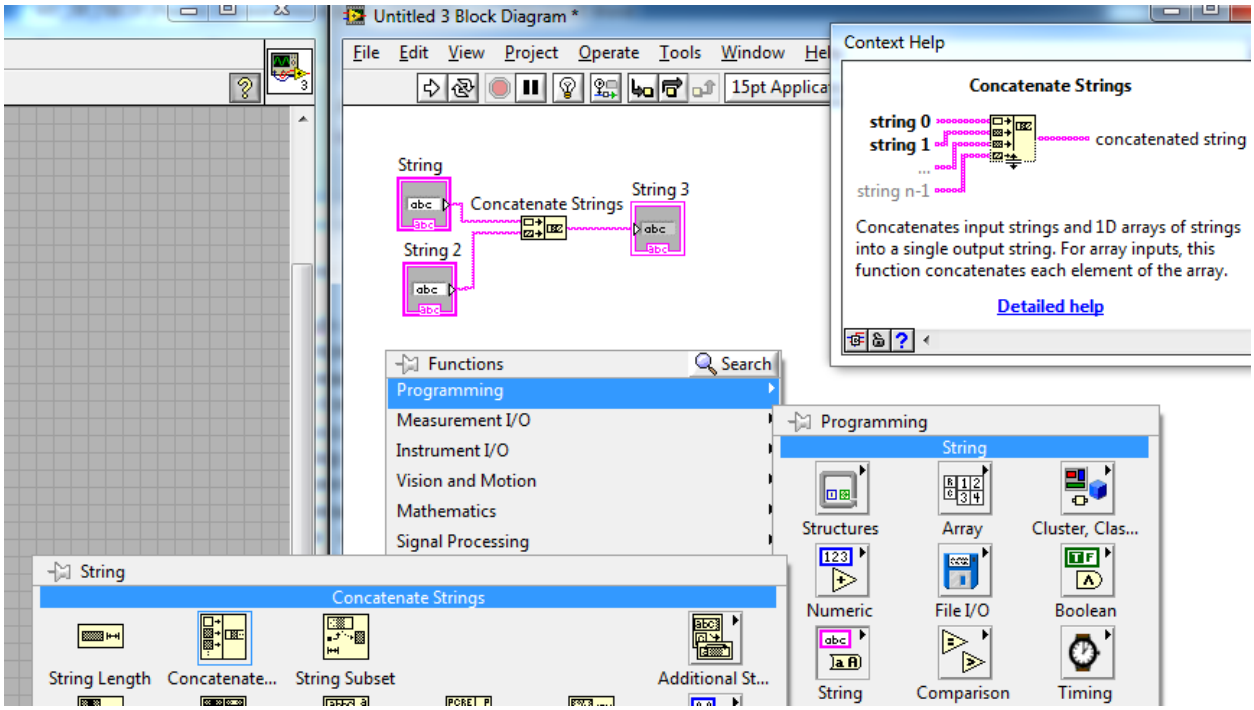


Рисунок 1.14

1.2.6.4 Запустить программу на выполнение (рисунок 1.15):

- ввести исходные данные в строковые элементы «String» и «String 2», непосредственно введя текст в окошко элемента;

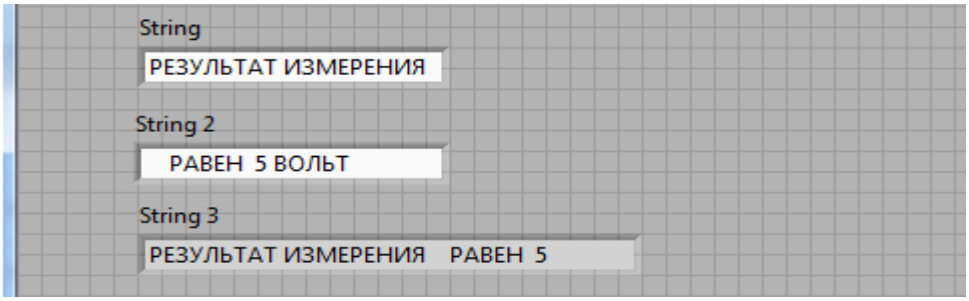


Рисунок 1.15

- в главном меню на фронт-панели (или на блок-диаграмме) нажать кнопку «Run», изображенную в виде стрелки, - однократное выполнение задания;

- элемент-индикатор «String 3» покажет результат связывания двух слов во фразу.

1.2.6.5 Сохранить программу под именем «String» с расширением «.vi».

1.3 Содержание отчета

Отчет должен содержать:

– цель и задание к лабораторной работе;

– screen shots интерфейсов фронт-панелей и блок-диаграмм ВП для всех заданий;

– описание элементов управления и индикации на блок-диаграмме;

– выводы по работе.

1.4 Контрольные вопросы

1.4.1 Какие типы данных есть в LabView?
1.4.2 Как они организованы?
1.4.3 Как запустить выполнение программы?
1.4.4 Какие режимы выполнения программ организованы в LabView?
1.4.5 Отличие главного и контекстного меню LabView.
1.4.6 Функциональное назначение фронт-панели и блок-диаграммы.

2 Лабораторная работа №2. Подпрограммы в среде графического программирования LabView

Цель работы: создание подпрограмм для измерения температуры в среде графического программирования LabView.

2.1 Задание к лабораторной работе

Задание 1. Создать ВП для преобразования градусов Цельсия в градусы Фаренгейта.

Задание 2. Создать подпрограмму ВП «Преобразование градуса Цельсия в градус Фаренгейта».

Задание 3. Создать ВП «Термометр», используя подпрограмму «Преобразование градуса Цельсия в градус Фаренгейта».

Задание 4. Создать подпрограмму ВП «Термометр».

2.2 Порядок выполнения работы

2.2.1 Для выполнения лабораторной работы необходимо зарегистрироваться в системе:

– загрузить файл Base_LabView, появится окно выбора лабораторных работ, в главном меню LabView нажать кнопку «RUN»;

- выбрать лабораторную работу «Подпрограммы в среде графического программирования LabView»;

– появится окно регистрации студента (рисунок 1.2): ввести свои Фамилию, Имя; ввести номер группы; нажать кнопку «Далее».

2.2.2 В появившемся окне выбрать в главном меню «File - New VI» - «Новый виртуальный прибор (ВП)». Для одновременной визуализации двух панелей на экране нажмите «Ctrl-T» (рисунок 1.3):

- «Front panel» - «Фронт-панель» - интерфейс пользователя;

- «Block-diagram» - «Блок-диаграмма» - интерфейс кода программы в LabView.

2.2.3 При создании ВП в LabView рекомендуется начать с организации интерфейса пользователя, помещая на «Front-panel» элементы управления из меню «Controls», для чего нажать в любом месте «Front-panel» на правую кнопку мыши.

2.2.4 Задание 1. Создать ВП для преобразования градусов Цельсия в градусы Фаренгейта.

Алгоритм преобразования градусов Цельсия в градусы Фаренгейта:

- градус Цельсия умножить на коэффициент 1,8;

- полученное произведение сложить с числом 32.

2.2.4.1 В главном меню LabView выбрать «File - New VI» - новый ВП.

На фронт-панели собрать интерфейс пользователя, представленный на рисунке 2.1.

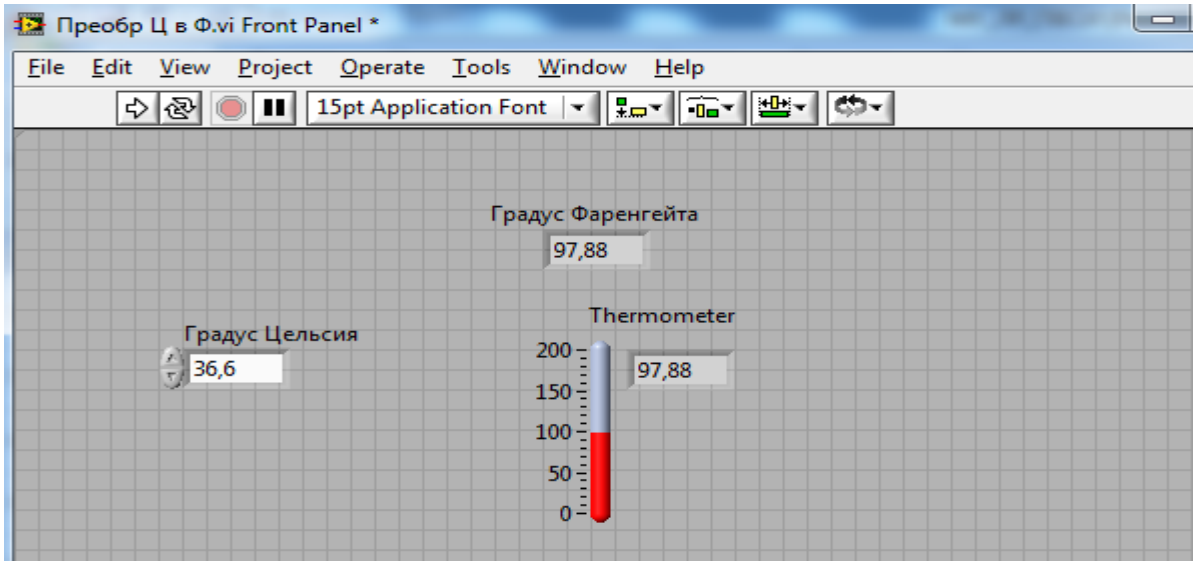


Рисунок 2.1

2.2.4.2 При выборе элементов управления на интерфейсе пользователя одновременно появляются их значки на блок-диаграмме. На блок-диаграмме собрать код программы по алгоритму, описанному в п.2.2.1 и представленному на рисунке 2.2.

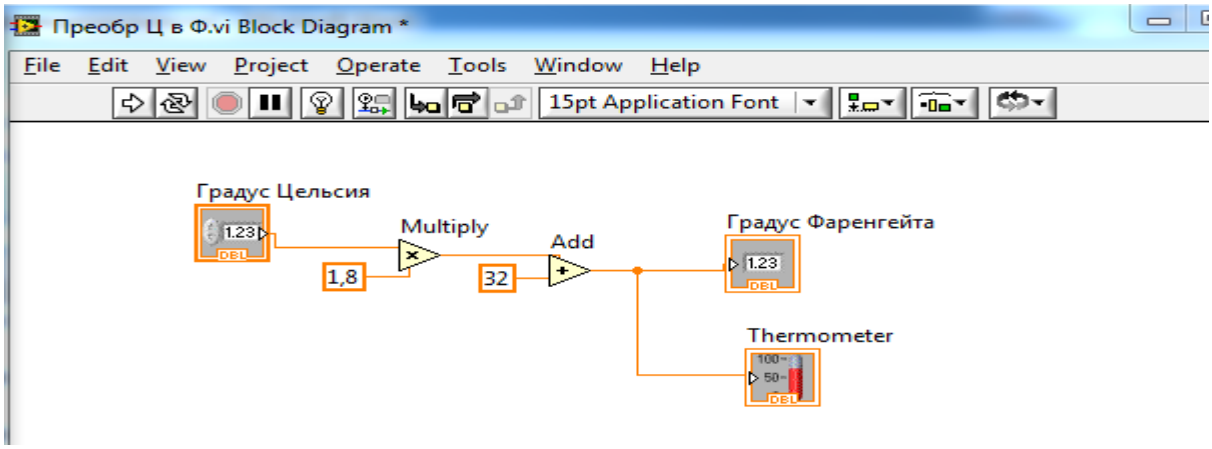


Рисунок 2.2

2.2.4.3 Запустить программу на выполнение:

- ввести исходные данные в числовой элемент «Градус Цельсия» - 36,6 ^{o}C;

- в главном меню на фронт-панели (или на блок-диаграмме) нажать кнопку «Run», изображенную в виде стрелки и задающей однократное выполнение задания;

- элементы-индикаторы покажут градус Фаренгейта, соответствующий 36,6 ^{o}C, в числовом виде - 97,88F (Градус Фаренгейта) и визуализацию на шкале термометра Thermometer (рисунок 2.1).

2.2.5 Задание 2. Создать подпрограмму ВП «Преобразование градуса Цельсия в градус Фаренгейта».

2.2.5.1 Подпрограмма любого ВП должна включать имя подпрограммы (Icon - в LabView), входные и выходные параметры (Connectors - в LabView). Для созданного в задании 1 (п.2.2.1) виртуального прибора разработка подпрограммы заключается в следующем:

- создание иконки подпрограммы (имени подпрограммы);

- организация входных и выходных параметров подпрограммы.

2.2.5.2 Создание иконки подпрограммы – Icon.

Навести мышь на иконку LabView , расположенную в правом верхнем углу фронт-панели. Нажать правую кнопку мыши, выбрать «Edit Icon». Появится окно редактора иконки (рисунок 2.3).

С помощью панели инструментов, расположенной слева от иконки LabView на рисунке 2.3, изменить вид иконки LabView на новое имя Вашей подпрограммы:

- инструментом «Пунктирный прямоугольник» выделить контуры иконки, затем нажатием «Delete» удаляем содержимое иконки;

- инструментом «Белый прямоугольник» (выше «Пунктирного прямоугольника») обозначить границы новой иконки;

- инструментом «Карандаш» или «Буквы - А» вписать имя подпрограммы «Преобразование градуса Цельсия в градус Фаренгейта»,

например «ЦФ»;

- инструментами «Изменение цвета» и «Заливка» закрасить иконку любым цветом (рисунок 2.4).

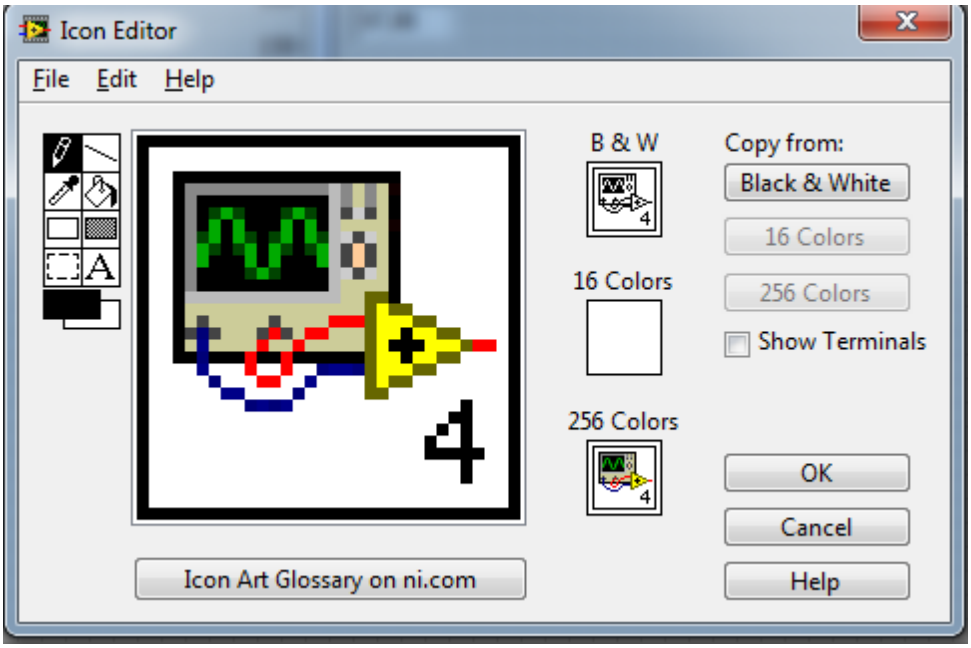


Рисунок 2.3

Готовая иконка должна отразиться в окне «256 color». Для завершения редактирования иконки нажать «Ок».

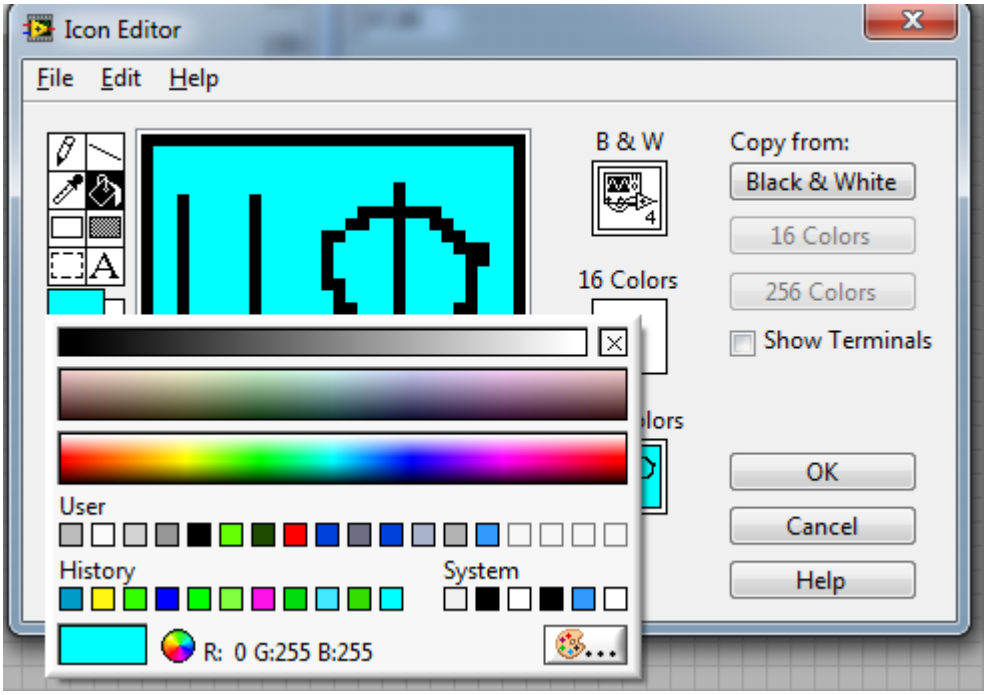


Рисунок 2.4

В правом верхнем углу фронт-панели появится новая иконка - имя подпрограммы «Преобразование градуса Цельсия в градус Фаренгейта».

2.2.5.3 Организация входных и выходных параметров подпрограммы.

Навести мышь на иконку LabView, расположенную в правом верхнем углу фронт-панели. Нажать правую кнопку мыши, выбрать «Show Connector». В окне иконки появится окно разъемов. Нажав еще раз на правую кнопку мышки, в появившемся меню выбрать «Patterns» - «Шаблоны» разъемов (рисунок 2.5) и выбрать тот вид разъемов, соответствующий числу входных и выходных параметров настоящей подпрограммы (один вход – один выход).

Для организации входного параметра подпрограммы щелкнуть мышью на один разъем в иконке «Connector» на фронт-панели, затем щелкнуть на входной элемент «Градус Цельсия», разъем окрасится в оранжевый цвет числового параметра (рисунок 2.6).

Для организации выходного параметра щелкнуть мышью на другой разъем в иконке «Connector», затем – на выходной элемент «Градус Фаренгейта», разъем окрасится в оранжевый цвет с черной рамкой выходного числового параметра .

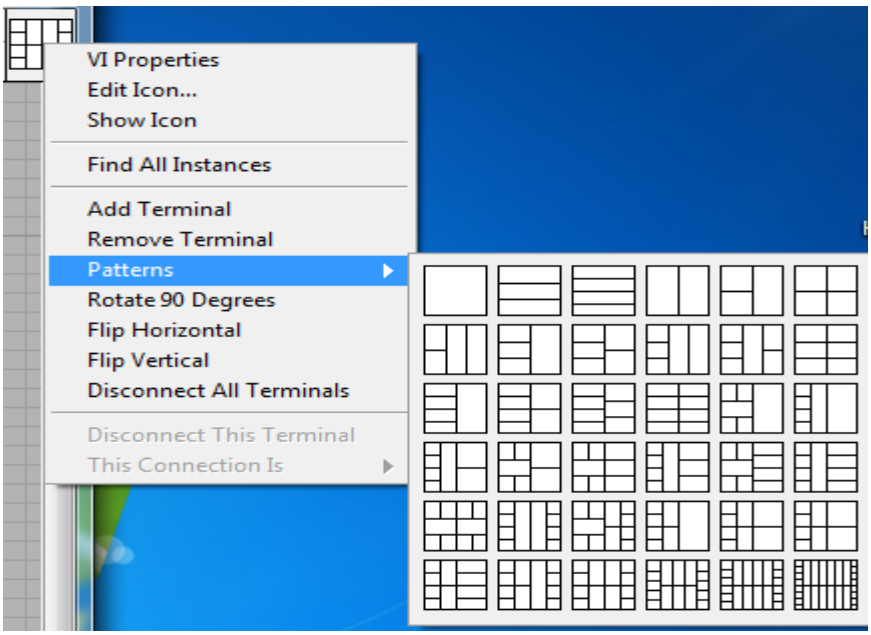


Рисунок 2.5

Таким образом, в правом верхнем углу фронт-панели создана новая иконка - имя подпрограммы «Преобразование градуса Цельсия в градус Фаренгейта» и ее входные и выходные параметры.

2.2.5.4 Сохранить подпрограмму под именем «Преобразование Ц в Ф» с расширением «.vi».

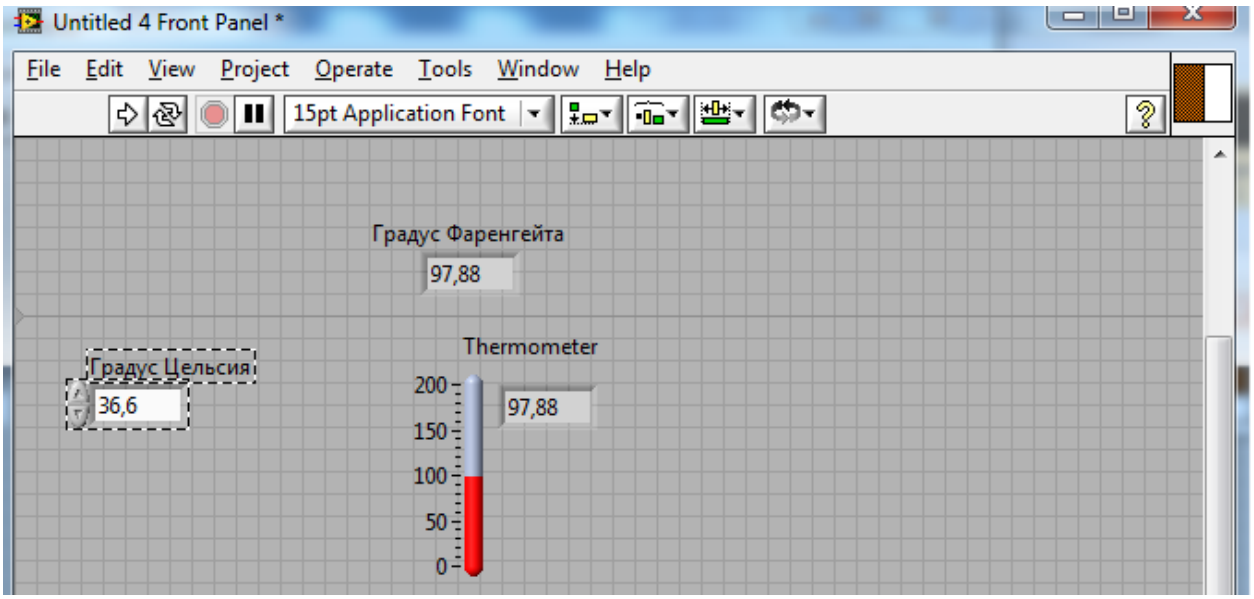


Рисунок 2.6

Таким образом, в правом верхнем углу фронт-панели создана новая иконка - имя подпрограммы «Преобразование градуса Цельсия в градус Фаренгейта» и ее входные и выходные параметры.

2.2.5.4 Сохранить подпрограмму под именем «Преобразование Ц в Ф» с расширением «.vi».

2.2.6 Задание 3. Создать ВП «Термометр», используя подпрограмму «Преобразование градуса Цельсия в градус Фаренгейта».

2.2.6.1 В главном меню LabView выбрать «File - New VI» - новый ВП.

На фронт - панели собрать интерфейс пользователя, представленный на рисунке 2.7:

- «Датчик температуры» - числовой элемент управления «Num Ctrl»;
- «Thermometer» - числовой элемент управления «Thermometer».

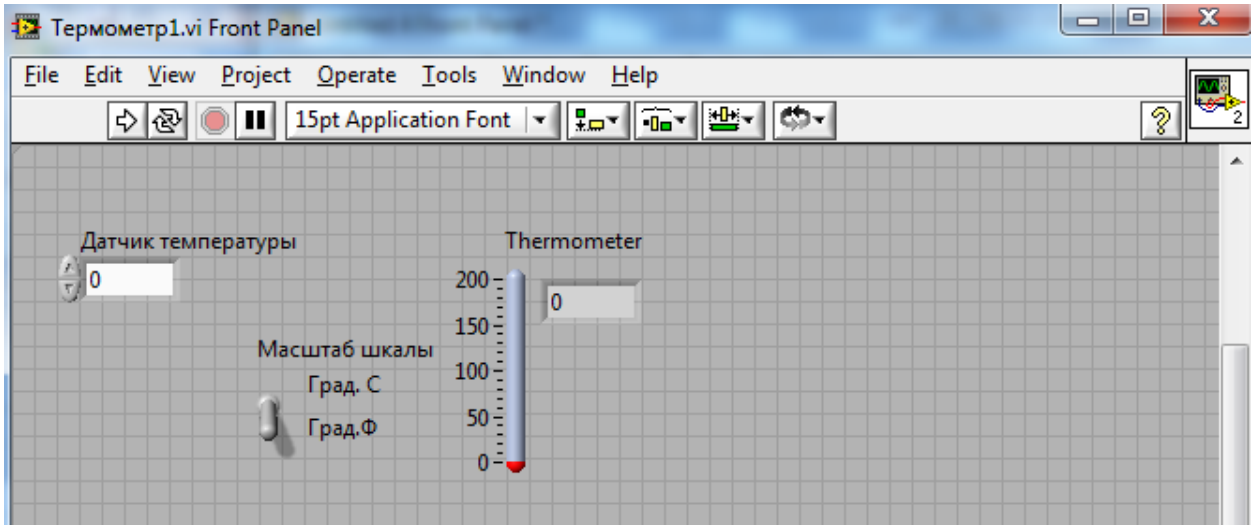


Рисунок 2.7

Подписать положения переключателя «Масштаб шкалы»: градусы Цельсия (Град.С) и градусы Фаренгейта (Град.Ф), дважды щелкнув на панель интерфейса для ввода текста.

ВП «Термометр» измеряет температуру, введенную «Датчиком температуры», и отображает ее значение на «Thermometer» либо в градусах Цельсия, либо в градусах Фаренгейта в зависимости от положения переключателя «Масштаб шкалы».

2.2.6.2 При выборе элементов управления на интерфейсе пользователя их значки появляются на блок-диаграмме. На блок-диаграмме собрать код программы по заданию, описанному в п.2.2.6.1 и представленному на рисунке 2.8.

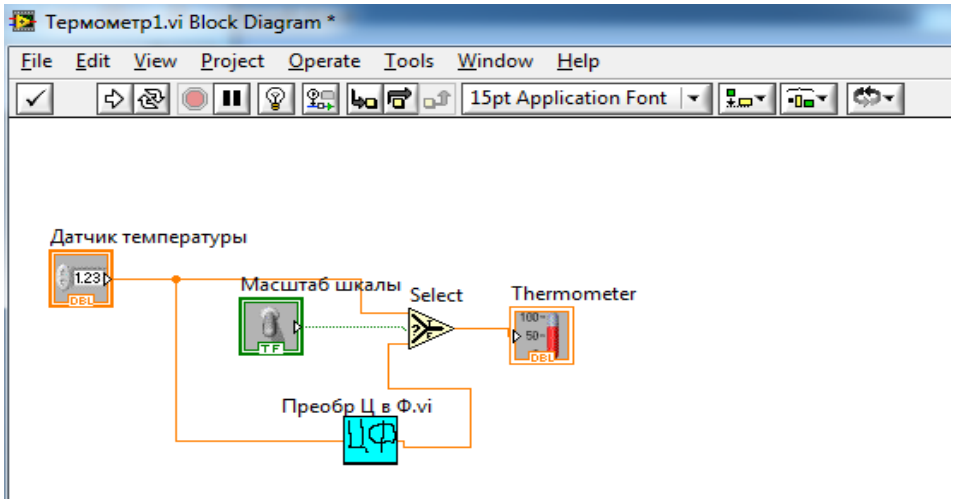


Рисунок 2.8

Для организации выбора масштаба шкалы использовать функцию «Select» из меню «Functions-Programming-Comparison».

Для выбора подпрограммы «Преобразование Ц в Ф.vi» использовать функцию «Select a VI…» из меню «Functions».

2.2.6.3 Запустить программу на выполнение:

- ввести исходные данные в числовой элемент «Датчик температуры» - 36,6 ^{o}C;

- ввести исходные данные в логический элемент «Масштаб шкалы», для этого переключить его в положение «Град.С»;

- в главном меню на фронт-панели (или на блок-диаграмме) нажать кнопку «Running Continuously», изображенную в виде замкнутых двух стрелок и означающую многократное (циклическое) выполнение. В данном режиме выполнения задания фронт-панель изменит внешний вид: интерфейс без сетки;

- элемент-индикатор «Thermometer» покажет измеренную датчиком температуру в градусах Цельсия, равную 36,6 ^{o}C;

- изменить положение логического элемента «Масштаб шкалы», для этого переключить его в положение «Град.Ф»;

- элемент-индикатор «Thermometer» покажет измеренную датчиком температуру в градусах Фаренгейта, соответствующий 36,6 ^{o}C, в числовом виде - 97,88F (рисунок 2.9);

- для выхода из циклического режима выполнения задания нажать в главном меню красную кнопку «Abort» - «Останов».

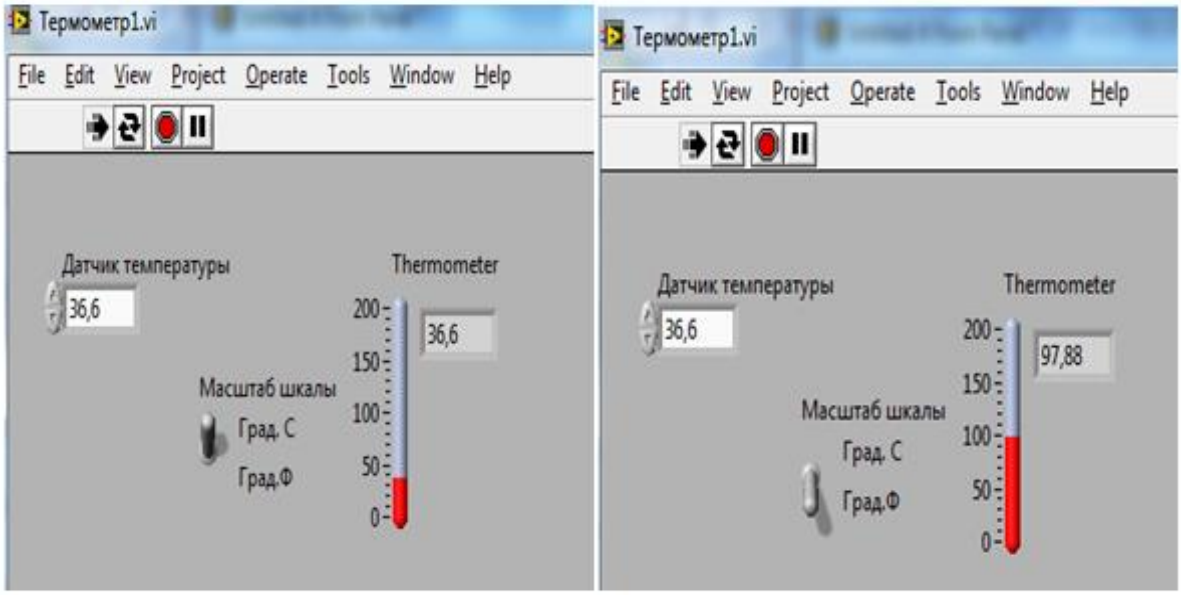


Рисунок 2.9

2.2.7 Задание 4 (самостоятельно). На базе разработанного выше ВП «Термометр» создать подпрограмму ВП «Термометр» по алгоритму, описанному в пунктах (2.2.5.1-2.2.5.3).

2.2.7.1 Сохранить подпрограмму под именем «Термометр» с расширением «.vi».

2.3 Содержание отчета

Отчет должен содержать:

– цель и задание к лабораторной работе;

– screen shots интерфейсов фронт-панелей и блок-диаграмм ВП для всех заданий;

– описание элементов управления и индикации на блок-диаграмме;

– выводы по работе.

2.4 Контрольные вопросы

2.4.1 Как организовано создание имени подпрограммы?

2.4.2 Как организовано создание параметров подпрограммы?

2.4.3 Функция организации выбора масштаба шкалы.

2.4.4 Назначение функции «Select a VI…».

2.4.5 Алгоритм создания иконки подпрограммы.

2.4.6 Описать окно редактора иконки.

3 Лабораторная работа №3. Многократные повторения и циклы

Цель работы: изучить программирование циклов по условию и циклов с фиксированным числом повторений.

3.1 Задание к лабораторной работе

Задание 1. Изучить программирование циклов по условию, для чего создать ВП, который генерирует случайные числа до тех пор, пока одно из них не окажется равным значению, введенному в числовой элемент ввода на фронт панели. На фронт-панели в числовом элементе вывода визуализировать количество итераций, выполненных циклом.

Задание 2. Изучить программирование циклов с фиксированным числом повторений, для чего создать ВП, который измеряет температуру один раз в секунду в течение одной минуты. Использовать ВП «Термометр» и ВП «Преобразование градуса Цельсия в градус Фаренгейта».

Задание 3. Создать ВП, который генерирует случайные числа в цикле «While». Организовать выход из цикла по нажатию кнопки на фронт-панели или по достижению циклом числа итераций, заданного элементом управления на фронт-панели.

3.2 Порядок выполнения лабораторной работы

3.2.1 Для выполнения лабораторной работы необходимо зарегистрироваться в системе:

- загрузить файл Base_LabView, появится окно выбора лабораторных работ, в главном меню LabView нажать кнопку «RUN»;
- выбрать лабораторную работу «Многократные повторения и циклы»;
- появится окно регистрации студента (рисунок 1.2): ввести свои Фамилию, Имя; ввести номер группы; нажать кнопку «Далее».

3.2.2 В появившемся окне выбрать в главном меню «File - New VI» - «Новый виртуальный прибор (ВП)». Для одновременной визуализации двух интерфейсов на экране нажмите «Ctrl-T» (рисунок 1.3).

Появятся две панели LabView:

- «Front-panel» - «Фронт-панель» - интерфейс пользователя;
- «Block-diagram» - «Блок-диаграмма» - интерфейс кода программы в LabView.

3.2.3 При создании ВП в LabView рекомендуется начать с организации интерфейса пользователя, помещая на «Front-panel» элементы управления из меню «Controls», для чего нажать в любом месте «Front-panel» на правую кнопку мыши.

3.2.4 Задание 1. Создать ВП, который генерирует случайные числа до тех пор, пока одно из них не окажется равным значению, введенному в числовой элемент ввода на фронт панели. На фронт панели в числовом

элементе вывода визуализировать количество итераций, выполненных циклом.

3.2.4.1 В главном меню LabView выбрать «File - New VI» - новый ВП.

3.2.4.2 На фронт панели собрать интерфейс задачи, как показано на рисунке 3.1.

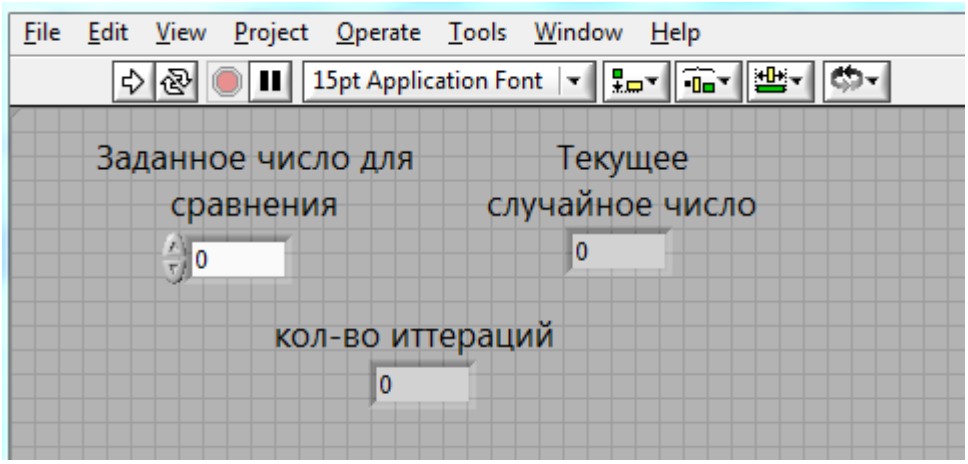


Рисунок 3.1

1) Из палитры «Controls-Numeric» поместить на фронт панель числовой элемент управления «Num Ctrl». Назвать его «Заданное число для сравнения».

2) Из палитры «Controls-Numeric» поместить на фронт панель числовой элемент отображения «Num Ind». Назвать его «Текущее случайное число».

3) Из палитры «Controls-Numeric» поместить на фронт панель еще один числовой элемент отображения «Num Ind». Назвать его «Количество итераций». Он будет показывать номер текущей итерации цикла.

3.2.4.3 Перейти на блок-диаграмму для сборки кода программы. На блок-диаграмме при создании интерфейса пользователя появились блоки числовых элементов управления и отображения «Заданное число для сравнения», «Текущее случайное число», «Количество итераций». Эти элементы по умолчанию заданы как вещественные числа (оранжевый цвет блоков). Элемент «Количество итераций» должен быть задан целым числом. Для изменения типа данных (вещественного на целый тип данных) щелкнуть правой кнопкой мыши на элемент «Количество итераций». Это можно сделать как на блок-диаграмме, так и на фронт панели. Появится контекстное меню элемента (рисунок 3.2). Выбрать «Representation – I32»: тип данных с вещественного двойной точности «DBL» изменится на целый тип «Integer 32 bit» (синий цвет блока).

3.2.4.4 Создайте на блок-диаграмме код программы, как показано на рисунке 3.3.

1) Поместить на блок-диаграмму функцию «Random Number» (Генератор случайных чисел), расположенную в меню «Function - Numeric». Генератор случайных чисел выдает числа в диапазоне от 0 до 1.

2) Для формирования случайных чисел в диапазоне от 0 до 100 поместить на блок-диаграмму функцию «Multiply» (Умножить),

расположенную в меню «Function - Numeric». Один вход этой функции соединить с выходом функции «Random Number». На второй вход функции щелкнуть правой кнопкой мыши, в появившемся контекстном меню выбрать «Create constant». В появившемся окне константы ввести значение 100.

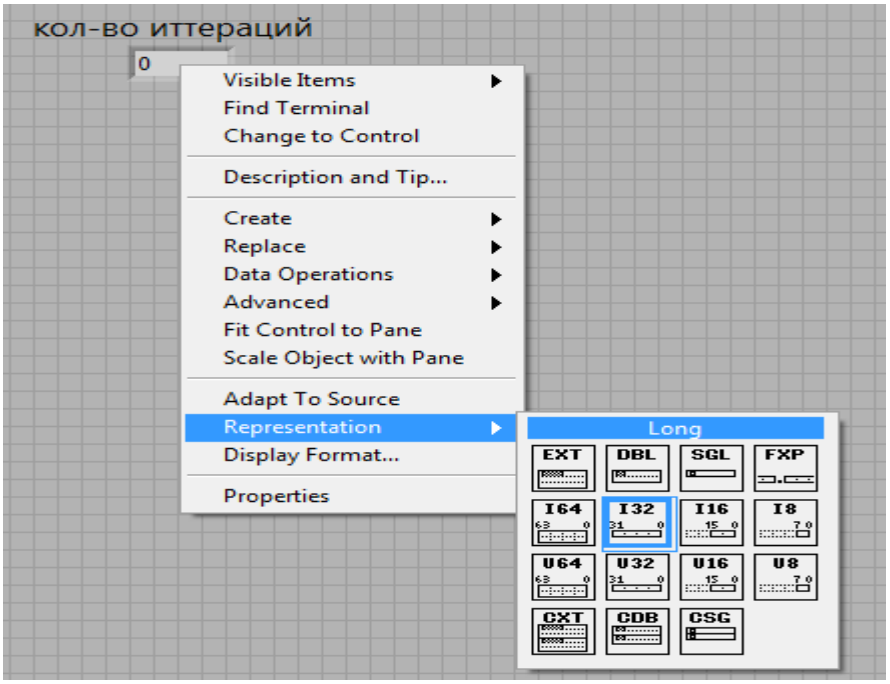


Рисунок 3.2

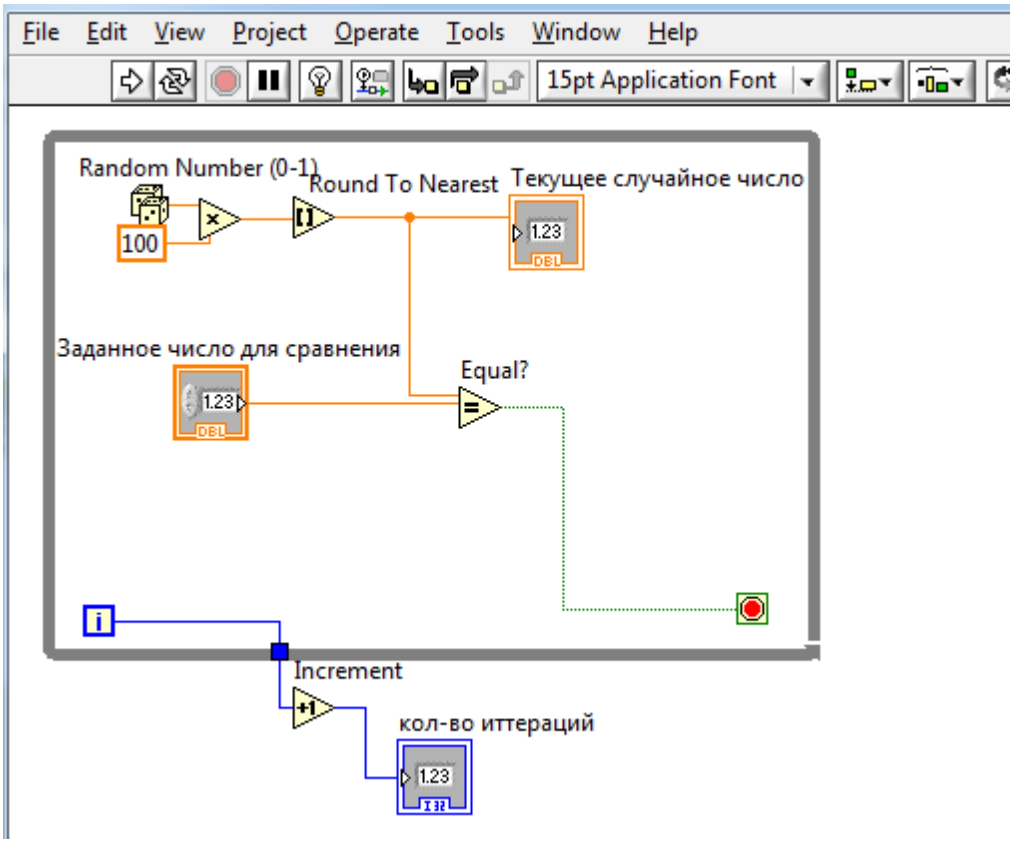


Рисунок 3.3

3) Округлить генерируемое случайное число до ближайшего целого числа. С этой целью поместить на блок-диаграмму функцию «Round To Nearest» (Округление до ближайшего целого), расположенную в меню «Function - Numeric», соединить ее вход с функцией «Умножить». Выход функции «Round To Nearest» со входом элемента «Текущее случайное число» для визуализации генерируемых случайных чисел, округленных до ближайшего целого числа.

4) Для сравнения текущего случайного числа с «Заданное число для сравнения», введенному в числовой элемент ввода на фронт панели поместить на блок-диаграмму функцию «Equal?», расположенную в меню «Function - Comparison». Если значения чисел, поданные на входы этой функции, равны, то функция выдаст значение «True», в противном случае – «False».

5) Все элементы на блок-диаграмме, кроме «Количество итераций», поместить в цикл «While». Для этого поместить на блок-диаграмму цикл «While» из меню «Function - Structures». Цикл «While» содержит два терминала: терминал счетчика итераций и терминал условия выхода, который имеет два возможных состояния: «Continue if True» - продолжить, если Истина, «Stop if True» - остановка, если Истина.

6) Значение счетчика итераций начинается с 0. Таким образом, для отображения количества итераций цикла необходимо увеличить на 1 значение счетчика итераций после завершения выполнения цикла. Для этого поместить на блок-диаграмму функцию «Increment» (Приращение на 1), расположенную в меню «Function - Numeric». Подсоединить функцию «Increment» к элементу «Количество итераций».

3.2.4.5 Сохранить ВП под именем «Подсчет количества итераций».

3.2.4.6 Перейти на фронт-панель. Задать «Заданное число для сравнения» в диапазоне от 0 до 100. Запустить программу на выполнение с помощью однократного «RUN».

3.2.5 Задание 2. Создать ВП, который измеряет температуру один раз в секунду в течение одной минуты. Использовать ВП «Термометр» и ВП «Преобразование градуса Цельсия в градус Фаренгейта».

3.2.5.1 Открыть в главном меню «File - New VI» - новый ВП.

3.2.5.2 На фронт-панели создать интерфейс пользователя, показанный на рисунке 3.4.

1) Из палитры «Controls-Numeric» поместить на фронт-панель два числовых элемента отображения «Num Ind». Назвать их «Входная температура», которая задается генератором случайных чисел, и «Секунды», которые показывают секунды системного времени компьютера за время выполнения задачи.

2) Из палитры «Controls-Numeric» поместить на фронт-панель числовой элемент отображения «Thermometer». Назвать его «Термометр». Он будет показывать измеренное значение температуры.

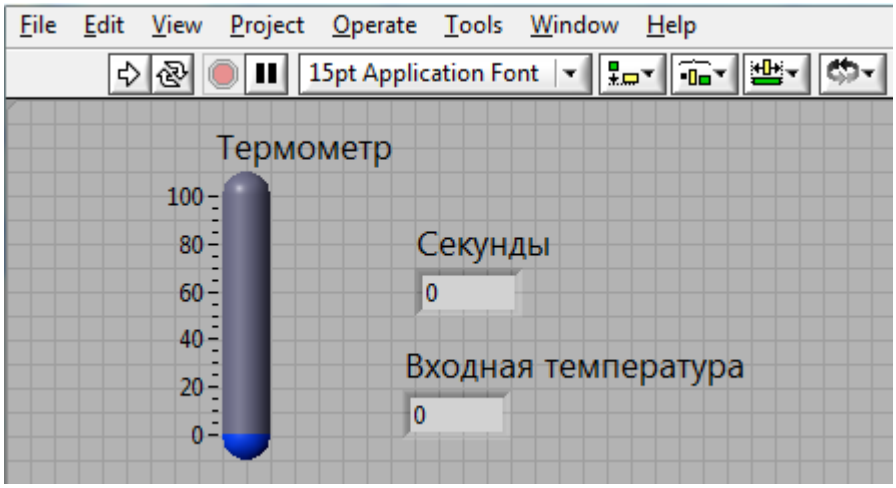


Рисунок 3.4

3.2.5.3 Перейти на блок-диаграмму для сборки кода программы. На блок-диаграмме при создании интерфейса пользователя появились блоки числовых элементов отображения «Термометр», «Секунды», «Входная температура». Эти элементы по умолчанию заданны как вещественные числа (оранжевый цвет блоков). Элемент «Секунды» должен быть задан целым числом. Для изменения типа данных (вещественного на целый тип данных) щелкнуть правой кнопкой мыши на элемент «Секунды». Это можно сделать как на блок-диаграмме, так и на фронт-панели. Появится контекстное меню элемента. Выбрать «Representation – I32»: тип данных с вещественного двойной точности «DBL» изменится на целый тип «Integer 32 bit» (синий цвет блока).

3.2.5.4 Создать на блок-диаграмме код программы, как показано на рисунке 3.5.

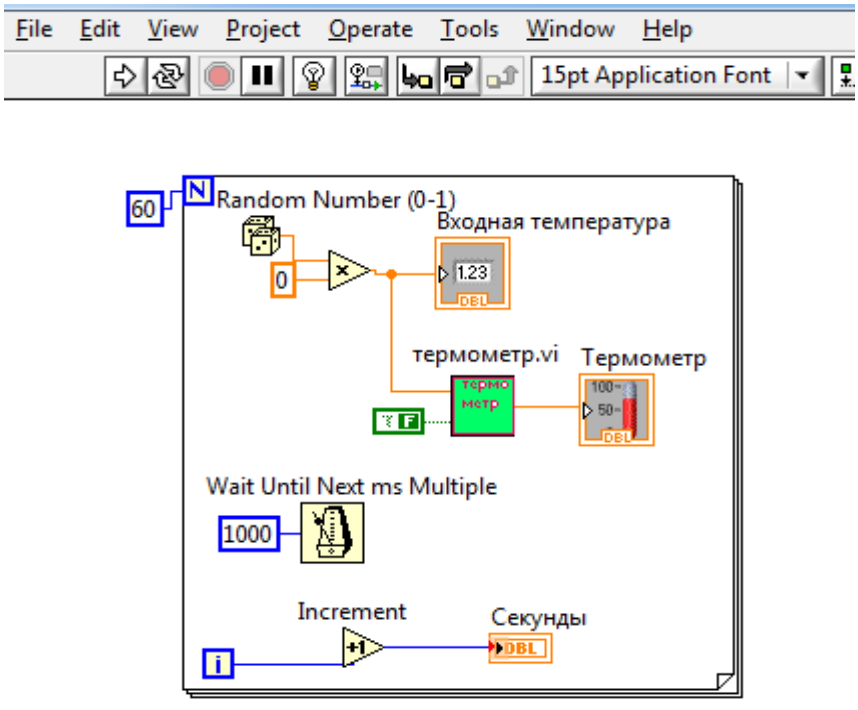


Рисунок 3.5

1) Для задания виртуальной температуры с помощью генератора случайных чисел поместить на блок-диаграмму функцию «Random Number», расположенную в меню «Function - Numeric». Генератор случайных чисел выдает числа в диапазоне от 0 до 1.

2) Для формирования случайных чисел в диапазоне от 0 до 100 поместить на блок-диаграмму функцию «Multiply» (Умножить), расположенную в меню «Function - Numeric». Один вход этой функции соединить с выходом функции «Random Number». На второй вход функции щелкнуть правой кнопкой мыши, в появившемся контекстном меню выбрать «Create – constant». В появившемся окне константы ввести значение 100.

3) Для визуализации входной температуры служит элемент «Входная температура».

4) Для измерения входной температуры поместить на блок-диаграмму ВП «Термометр». Для выбора подпрограммы «Термометр.vi» использовать функцию «Select a VI…» из меню «Functions».

5) Все элементы на блок-диаграмме поместить в цикл «For» с известным числом повторений. Для этого поместить на блок-диаграмму цикл «For» из меню «Function - Structures». Цикл «For» содержит два терминала: терминал максимального числа итераций , расположенный в верхнем левом углу цикла, и терминал счетчика итераций .

Для того, чтобы задать количество итераций выполнения цикла щелкнуть правой кнопкой мыши по терминалу максимального числа итераций и выбрать в контекстном меню «Create constant», задать максимальное число итераций равным 60, что равно 60 секундам или 1 минуте. Согласно заданию, температура измеряется раз в секунду в течение одной минуты.

Значение счетчика итераций начинается с 0. Таким образом, для отображения количества итераций цикла необходимо увеличить на единицу значение счетчика итераций после завершения выполнения цикла. Для этого поместить на блок-диаграмму функцию «Increment» (Приращение на 1), расположенную в меню «Function - Numeric». Подсоединить функцию «Increment» к элементу «Секунды».

Таким образом, цикл «For» генерирует случайные числа (температуру) каждую секунду 60 раз и отображает их в элементах «Входная температура» и «Температура».

6) Для хронометража времени работы ВП используют функции ожидания «Wait Until Next ms Multiple» («Задержка до следующего кратного интервала, мс») или «Wait (ms)» («Задержка, мс»), которые находятся в меню «Function – Time and Dialog».

Функция «Wait (ms)» заставляет ВП ждать некоторое число миллисекунд, прежде чем продолжать выполнять программу.

Функция «Wait Until Next ms Multiple» заставляет ВП ждать, пока значения внутренних часов не будут равны или не превысят кратного количества миллисекунд, поданных на вход функции.

Эти функции похожи, но не одинаковые. Так, задержка «Wait Until Next ms Multiple» будет меньше заданного числа миллисекунд при первой итерации, поскольку зависит от того, сколько времени потребуется для перехода к следующей итерации и возобновления работы ВП. То есть ВП может выйти из синхронизации и замедлиться.

7) Поместить в цикл функцию «Wait Until Next ms Multiple». Щелкнуть правой кнопкой мыши по элементу данной функции и выбрать в контекстном меню «Create constant», присвоить константе значение 1000. Это означает, что каждая итерация цикла будет выполняться с интервалом времени 1000 мс (раз в секунду).

3.2.5.5 Сохранить ВП под именем «Измерение температуры во времени».

3.2.5.6 Прежде, чем запустить программу на выполнение, убедитесь, что на диске загружены ВП «Термометр» и ВП «Преобразование градуса Цельсия в градус Фаренгейта».

Запустить программу на выполнение с помощью однократного «RUN».

3.2.6 Задание 3. Создать ВП, который генерирует случайные числа в цикле «While». Организовать выход из цикла по нажатию кнопки на фронт-панели или по достижению циклом числа итераций, заданного элементом управления на фронт-панели.

Задание выполнить самостоятельно. Сохранить ВП под именем «Цикл While».

3.3 Содержание отчета

Отчет должен содержать:

- цель и задание к лабораторной работе;
- screen shot интерфейсов и блок-диаграмм лабораторной работы для каждого задания;
- описание изучаемых операторов LabView;
- выводы по работе.

3.4 Контрольные вопросы

3.4.1 Какие структуры циклов реализованы в Labview?

3.4.2 Как организовать цикл с известным числом повторений?

3.4.3 Как организовать цикл с неизвестным числом повторений?

3.4.4 Функции ожидания в Labview.

3.4.5 Как определить число итераций, выполненных циклом с неизвестным числом повторений?

3.4.6 Как изменить тип данных числовой переменной?

4 Лабораторная работа №4. Организация доступа к значениям предыдущих итераций цикла

Цель работы: организовать доступ к значениям предыдущих итераций цикла с помощью сдвиговых регистров.

4.1 Задание к лабораторной работе

Задание 1. Создать ВП для организации доступа к данным предыдущих итераций цикла, используя сдвиговые регистры.

Задание 2. Создать ВП с применением стека сдвиговых регистров.

Задание 3. Создать ВП, имитирующий игру в кости.

4.2 Порядок выполнения лабораторной работы

4.2.1 Для выполнения лабораторной работы необходимо зарегистрироваться в системе:

- загрузить файл Base_LabView, появится окно выбора лабораторных работ (рисунок 1.1), в главном меню LabView нажать кнопку «RUN»;
- выбрать лабораторную работу «Организация доступа к значениям предыдущих итераций цикла»;
- появится окно регистрации студента (рисунок 1.2): ввести свои Фамилию, Имя; ввести номер группы; нажать кнопку «Далее».

4.2.2 Задание 1. Создать ВП для организации доступа к данным предыдущих итераций цикла, используя сдвиговые регистры.

4.2.2.1 Открыть «New VI» - новый ВП.

4.2.2.2 На фронт панели создать интерфейс пользователя, показанный на рисунке 4.1.

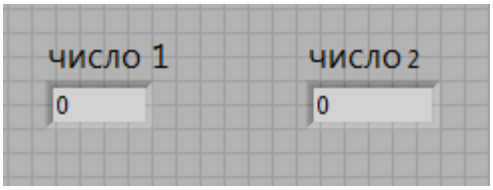


Рисунок 4.1

4.2.2.3 Перейти на блок-диаграмму и создать код программы, как показано на рисунке 4.2.

1) Сдвиговый регистр создается щелчком правой кнопки мыши по границе цикла и выбором из контекстного меню «Add Shift Register».

Сдвиговый регистр выглядит как пара терминалов, расположенных друг против друга на противоположных вертикальных границах цикла в месте щелчка мышки. Правый терминал содержит стрелку вверх, в нем сохраняются данные по завершению текущей итерации. Затем эти данные передаются в левый терминал (со стрелкой вниз) в следующую итерацию цикла. Таким образом, по завершении выполнения цикла последнее значение, записанное в регистр, останется в правом терминале.

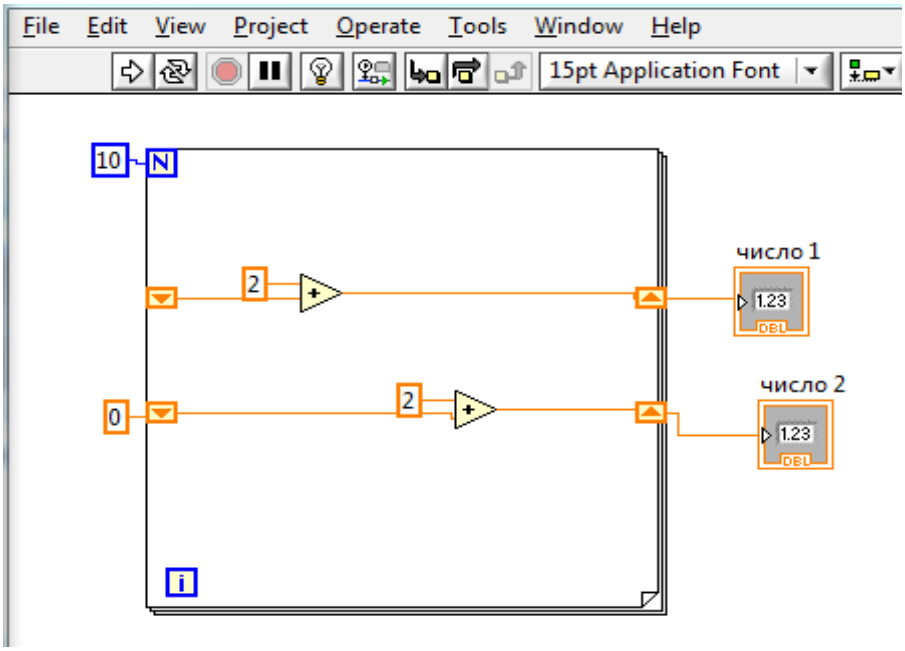


Рисунок 4.2

2) Различают неинициализированный (первый сверху регистр на рисунке 4.2) и инициализированный сдвиговый регистр (второй сверху регистр).

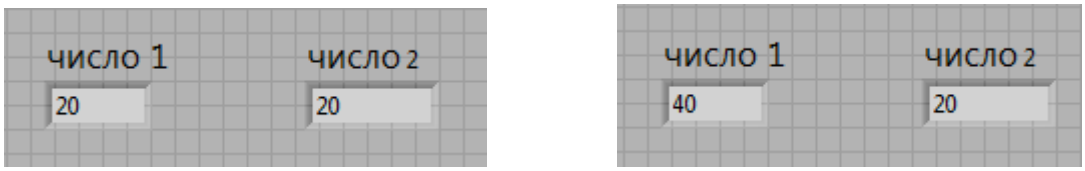
Чтобы инициализировать сдвиговый регистр, необходимо щелкнуть правой кнопкой мыши на его левый терминал, выбрать «Create Constant» и ввести любое значение.

Цикл с инициализированным сдвиговым регистром при повторном запуске ВП использует значение, заданное извне на левый терминал регистра.

Цикл с неинициализированным сдвиговым регистром при повторном запуске ВП использует значение, записанное в регистр во время выполнения последнего цикла.

4.2.2.4 Запустить программу на выполнение с помощью однократного «RUN».

На рисунке 4.3 приведены результаты выполнения программы после первого запуска ВП (а), после второго запуска ВП (б): число 1 показывает результат работы неинициализированного сдвигового регистра; число 2 – инициализированного сдвигового регистра.



а) б)

Рисунок 4.3

4.2.3 Задание 2. Создать ВП с применением стека сдвиговых регистров.

4.2.3.1 Для осуществления доступа к значениям предыдущих итераций цикла создать *стек сдвиговых регистров*. Для этого добавить на фронт-панель пять числовых элементов отображения, как показано на рисунке 4.4.

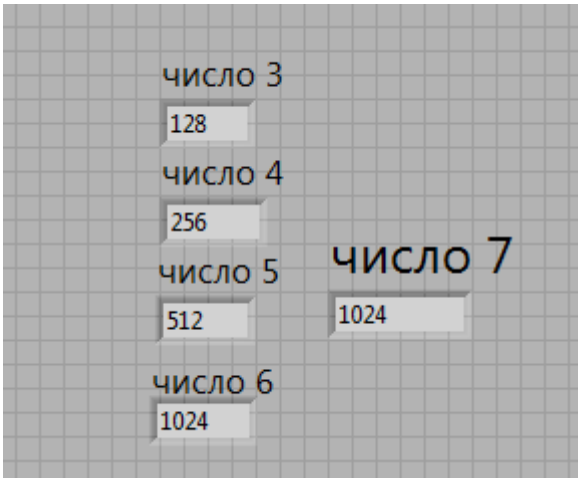


Рисунок 4.4

4.2.3.2 Перейти на блок-диаграмму и создать код программы, как показано на рисунке 4.5.

1) У левого терминала сдвигового регистра создать стек сдвигового регистра. Для этого щелкнуть правой кнопкой мыши по левому терминалу и выбрать пункт контекстного меню «Add Element». Стек содержит столько элементов («Add Element»), сколько значений предыдущих итераций необходимо получить.

2) Таким образом, число 6 и число 7 показывают значение последней итерации, число 5 – значение предпоследней итерации, число 4 – значение, полученное две итерации назад, число 3 - значение, полученное три итерации назад (рисунок 4.4).

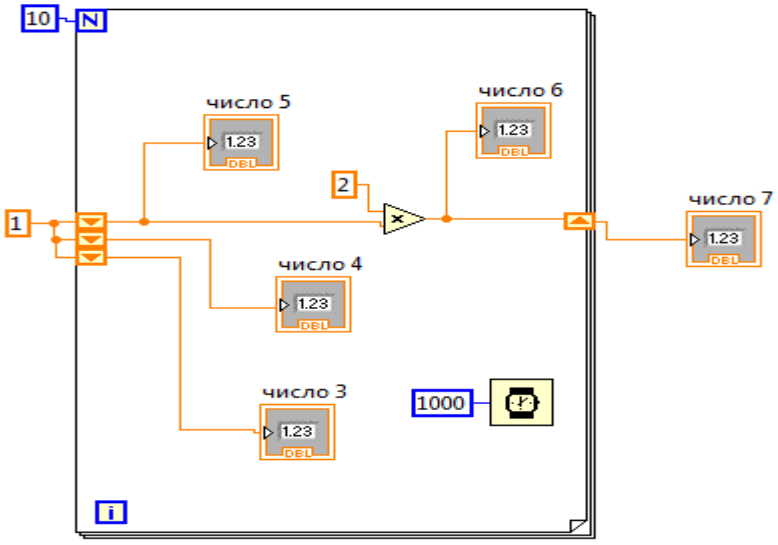


Рисунок 4.5

4.2.3.3 Сохранить ВП под именем «Сдвиговые регистры».

4.2.3.4 Запустить программу на выполнение с помощью однократного «RUN».

4.2.4 Задание 3. Создать ВП, который имитирует игру в кости (возможные значения граней кости 1- 6).

Исходные данные – количество бросков кости; выходные данные – количество раз, когда игральная кость показывает то, или иное значение.

Использовать сдвиговые регистры.

Задание выполнить самостоятельно.

4.3 Содержание отчета

Отчет должен содержать:

- цель и задание к лабораторной работе;
- screen shot интерфейсов и блок-диаграмм лабораторной работы для каждого задания;
- описание изучаемых операторов LabView;
- выводы по работе.

4.4 Контрольные вопросы

4.4.1 С какой целью используются сдвиговые регистры?

4.4.2 Отличие сдвигового регистра от стека сдвиговых регистров.

4.4.3 В чем отличие инициализированного сдвигового регистра от неинициализированного?

4.4.4 Как организовать стек сдвиговых регистров?

4.4.5 Сколько элементов стека сдвигового регистра необходимо для вывода данных, полученных пять итераций назад?

5 Лабораторная работа № 5. Работа с массивами

Цель работы: создание массивов в Labview, знакомство с их функциями обработки.

5.1 Задание к лабораторной работе

Задание 1. Создать ВП, который формирует массив случайных чисел, масштабирует полученный массив и выделяет из него подмножество.

Задание 2. Создать ВП, который накапливает массив значений температур с помощью «Термометр.vi». Размер массива определить с помощью элемента управления с фронт панели. С помощью функции «Initialize Array» создать массив такого же размера, в котором все элементы имеют значение 10. Сложить два массива, найти размер конечного массива и вычислить его среднее значение.

Задание 3. Создать ВП, который создает двумерный массив случайных чисел, содержащий 3 строки и 10 столбцов. Вывести на фронт-панель три одномерных массива, соответствующих строкам исходного двумерного массива.

Задание 4. Создать ВП, который формирует массив из 10 случайных чисел в диапазоне от 0 до 100, затем изменяет порядок элементов в массиве таким образом, чтобы элемент массива с индексом 0 стал элементом с индексом 10, элемент с индексом 1 стал элементом с индексом 9 и т.д.

5.2 Порядок выполнения лабораторной работы

5.2.1 Для выполнения лабораторной работы необходимо зарегистрироваться в системе:

- загрузить файл Base_LabView, появится окно выбора лабораторных работ (рисунок 1.1), в главном меню LabView нажать кнопку «RUN»;
- выбрать лабораторную работу «Работа с массивами»;
- появится окно регистрации студента (рисунок 1.2): ввести свои Фамилию, Имя; ввести номер группы; нажать кнопку «Далее».

5.2.2 Задание 1. Создать ВП, который формирует массив случайных чисел, масштабирует полученный массив и выделяет из него подмножество.

5.2.2.1 В открывшемся меню LabView выбрать «New VI» - новый ВП.

5.2.2.2 На фронт-панели собрать интерфейс задачи, как показано на рисунке 5.1.

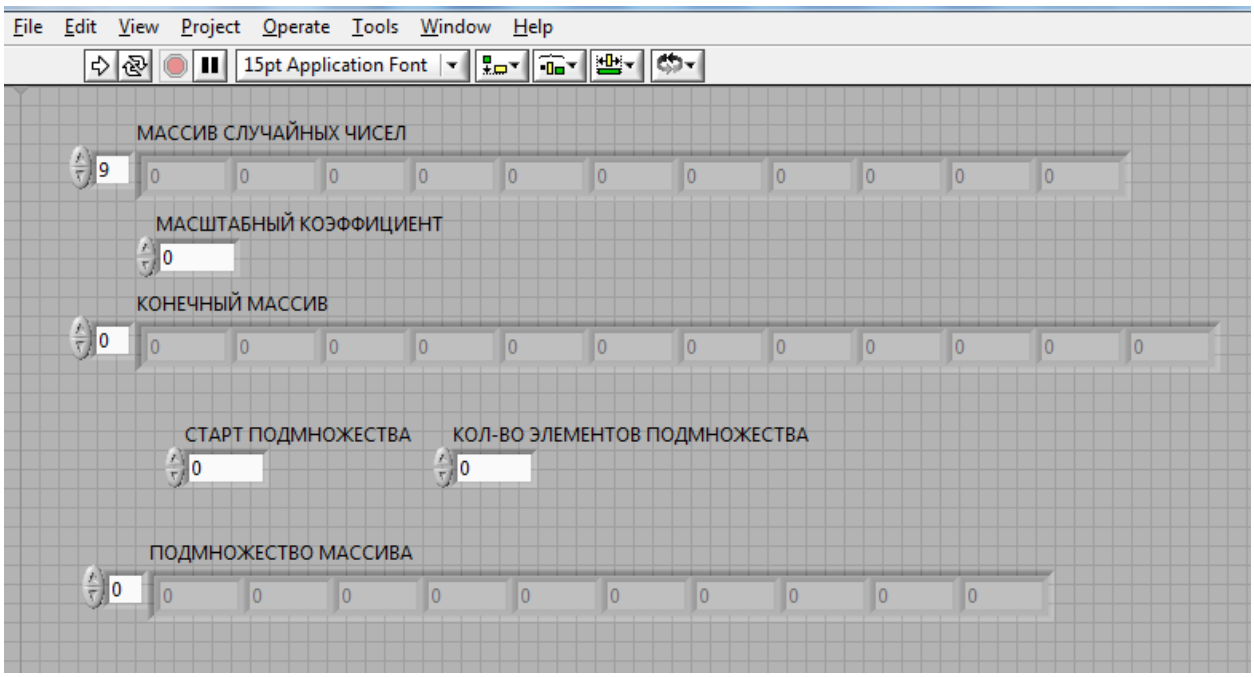


Рисунок 5.1

1) Создание массива элементов управления или индикации данных состоит из двух этапов:

- из меню «Controls – Modern - Array & Cluster» выбрать шаблон массива «Array» и поместить его на фронт-панель;

- из меню «Controls - Numeric» поместить в шаблон массива элемент управления (или индикации).

В шаблон массива можно разместить элементы только одного типа данных. Нельзя создать массив, состоящий из массивов.

У созданного массива слева располагается окно элемента индекса. Справа – окна элементов управления (или индикации) данных. Нумерация элементов массива в LabView всегда начинается с 0.

2) На фронт-панели создать одномерный массив из числовых элементов индикации. Назвать его «Массив случайных чисел».

3) Скопировать этот массив два раза. Переименовать копии массива на «Конечный массив» и «Подмножество массива».

4) На фронт-панель поместить три числовых элемента управления. Назвать их «Масштабный коэффициент», «Старт подмножества», «Количество элементов подмножества».

Тип данных элементов «Старт подмножества», «Количество элементов подмножества» заменить на «I32» через контекстное меню этих элементов и «Representation».

5.2.2.3 Перейти на блок-диаграмму и создать код программы, как показано на рисунке 5.2.

1) Для организации массива случайных чисел поместить генератор случайных чисел «Random Number» внутрь цикла «For». Размерность массива задать максимальным количеством итераций цикла.

2) Масштабирование исходного массива задать умножением «Массива случайных чисел» на «Масштабный коэффициент».

3) Выделение «Подмножества массива» из «Конечного массива»: использовать функцию обработки массива «Array Subset» из меню «Function – Programming - Array».

Функция «Array Subset» выдает часть массива, начиная с индекса, заданного на входе функции «index», и длиной, заданной на входе «length».

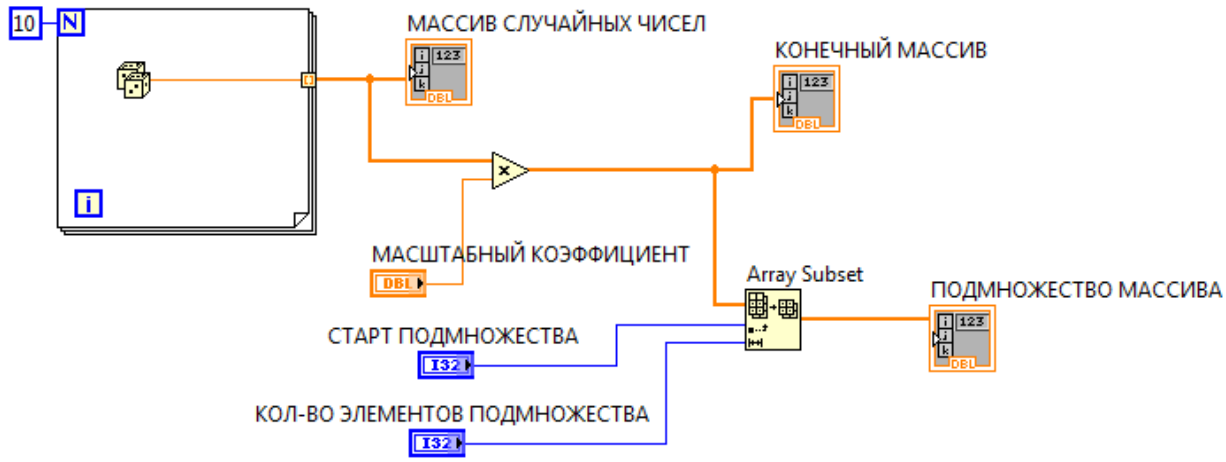


Рисунок 5.2

5.2.2.4 Сохранить ВП под именем «Работа с массивами.vi».

5.2.2.5 Перейти на фронт-панель для ввода исходных данных: «Масштабный коэффициент» = 10; «Старт подмножества» = 2; «Количество элементов подмножества»=5.

5.2.2.6 Запустить программу на выполнение с помощью однократного «RUN».

Результат выполнения программы: в «Массив случайных чисел» отобразится массив из 10 чисел; ВП умножит каждое значение этого массива на масштабный коэффициент 10 и отобразит их в «Конечный массив»; ВП выделит из «Конечного массива» подмножество, начиная с 3-его элемента массива, длиной в 5 элементов и отобразит его в «Подмножество массива».

Проверить результат выполнения программы.

5.2.2.7 Изменить исходные данные в п.5.2.2.5. Запустить программу на выполнение с помощью однократного «RUN».

5.2.3 Функции обработки массива:

- «Array Size» - показывает количество элементов массива каждой размерности;
- «Initialize Array» – создает n-мерный массив, в котором каждый элемент массива инициализирован значением поля данных «element». Размерность задается в поле «dimension size»;
- «Build Array» – объединяет несколько массивов или добавляет элемент в n-мерный массив;
- «Array Subset» – выдает часть массива, начиная с индекса, поступившего на поле «index», и длиной, указанной в поле «length»;
- «Index Array» – выдает элемент, соответствующий индексу, значение которого подается на поле ввода «index». Для двумерного массива: верхнее поле «index» указывает номер строки, нижнее – номер столбца;
- «Полиморфизм»: арифметические операции, расположенные в меню «Function – Numeric», являются полиморфными. Например, можно использовать функцию «Add» для прибавления скалярной величины к массиву или сложения двух массивов.

5.2.4 Задание 2. Выполнить задание самостоятельно.

Создать ВП, который накапливает массив значений температур с помощью «Термометр.vi». Размер массива определить с помощью элемента управления с фронт-панели. С помощью функции «Initialize Array» создать массив такого же размера, в котором все элементы имеют значение 10. Сложить два массива, найти размер конечного массива и вычислить его среднее значение.

На фронт-панель вывести массив значений температур, инициализированный массив, конечный массив и среднее значение.

Сохранить ВП под именем «Среднее массива.vi».

5.2.5 Задание 3. Создать ВП, который создает двумерный массив случайных чисел, содержащий 3 строки и 10 столбцов. Вывести на фронт-

панель три одномерных массива, соответствующих строкам исходного двумерного массива.

5.2.5.1 В открывшемся меню LabView выбрать «New VI» - новый ВП.

5.2.5.2 На фронт-панели собрать интерфейс задачи, как показано на рисунке 5.3.

1) Создание двумерного массива (матрицы) элементов управления или индикации данных состоит из трех этапов:

- из меню «Controls – Modern - Array & Cluster» выбрать шаблон массива «Array» и поместить его на фронт-панель;

- из меню «Controls - Numeric» поместить в шаблон массива элемент управления (или индикации);

- щелкнуть правой кнопкой мыши по элементу индекса полученного одномерного массива и выбрать из контекстного меню «Add Dimension» или использовать инструмент «ПЕРЕМЕЩЕНИЕ» и изменить размер элемента индекса.

2) На фронт-панели создать двумерный массив из числовых элементов индикации, состоящий из трех строк и десяти столбцов. Назвать его «Матрица».

3) Создать одномерный массив. Назвать его «Одномерный массив». Скопировать этот массив два раза. Появятся «Одномерный массив 2» и «Одномерный массив 3».

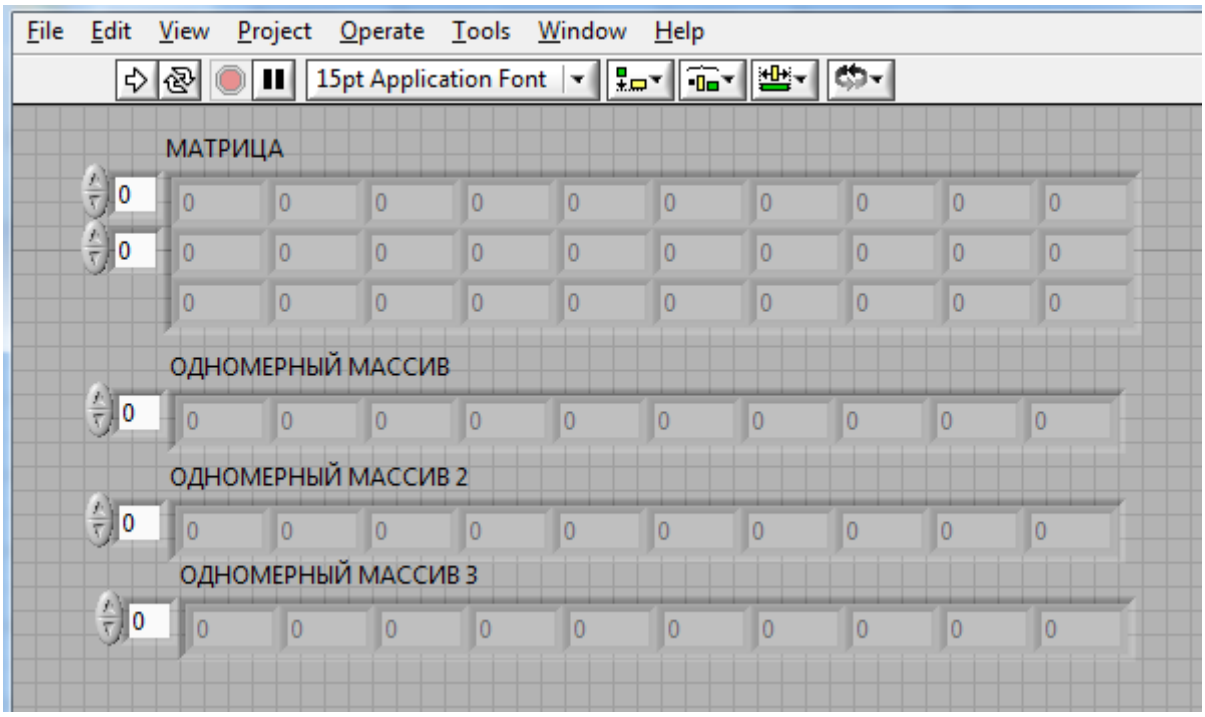


Рисунок 5.3

5.2.5.3 Перейти на блок-диаграмму и создать код программы, как показано на рисунке 5.4.

1) Для организации двумерного массива случайных чисел поместить генератор случайных чисел «Random Number» внутрь вложенных циклов «For». Внешний цикл задает число строк в матрице: задать максимальное количество итераций = 3. Внутренний цикл задает число столбцов в матрице: задать максимальное количество итераций = 10.

2) Для выделения одномерного массива из матрицы использовать функцию «Index Array» (п.5.2.3). При подаче на вход «Array» потока данных матрицы появятся два входа для индекса строки и столбца.

На входе индекса строки «index (row)» создать константу, соответствующую индексу строки, которую необходимо выделить из матрицы. Вход индекса столбца «disabled index (col)» оставить пустым.

5.2.5.4 Сохранить ВП под именем «Матрица.vi».

5.2.5.5 Запустить программу на выполнение с помощью однократного «RUN».

Результат выполнения программы: в «Матрице» отобразится двумерный массив из 3 строк и 10 строк случайных чисел; ВП отобразит строку с индексом «0» в «Одномерный массив»; ВП отобразит строку с индексом «1» в «Одномерный массив 2»; ВП отобразит строку с индексом «2» в «Одномерный массив 3».

Проверить результат выполнения программы.

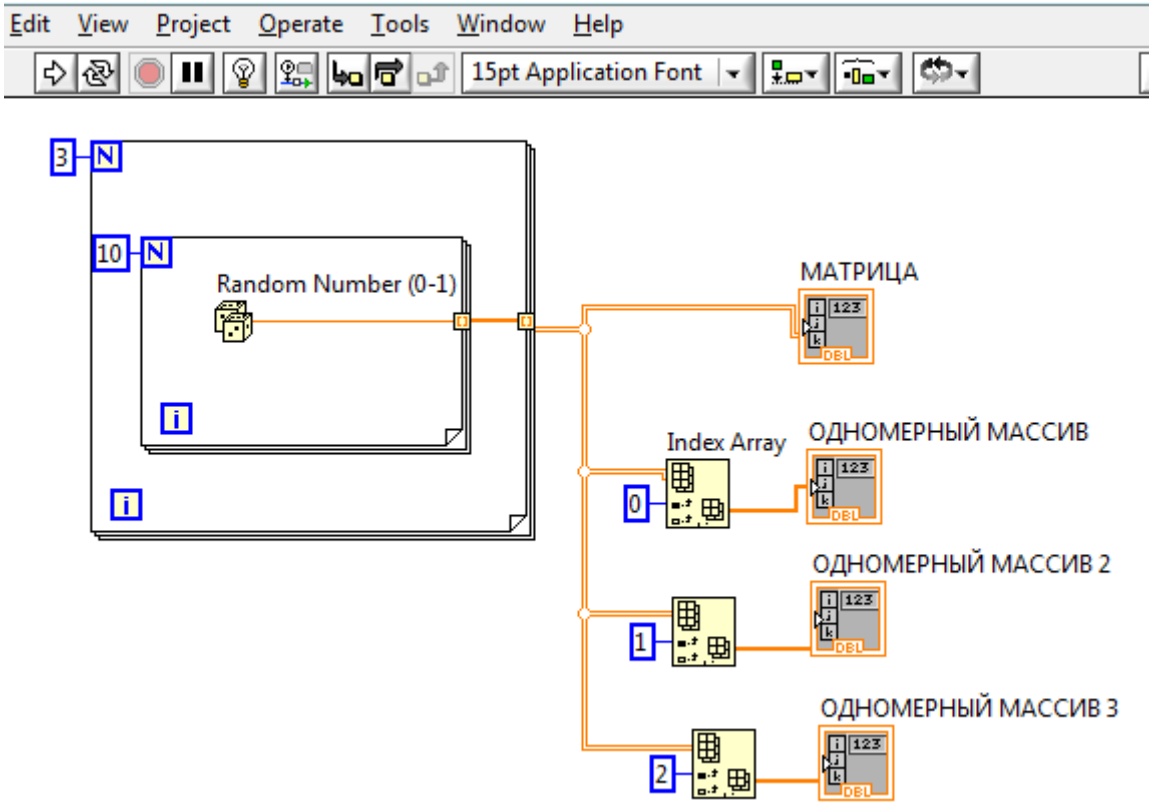


Рисунок 5.4

5.2.6 Задание 4. Выполнить задание самостоятельно.

Создать ВП, который формирует массив из 10 случайных чисел в диапазоне от 0 до 100, затем изменяет порядок элементов в массиве таким образом, чтобы элемент массива с индексом 0 стал элементом с индексом 10, элемент с индексом 1 стал элементом с индексом 9 и т.д.

Использовать функцию «Reverse 1D Array».

Сохранить ВП под именем «Перевернутый массив.vi».

5.3 Содержание отчета

Отчет должен содержать:

- цель и задание к лабораторной работе;
- screen shot интерфейсов и блок-диаграмм лабораторной работы для каждого задания;
- описание изучаемых операторов LabView;
- выводы по работе.

5.4 Контрольные вопросы

5.4.1 Как создать одномерный массивы?
5.4.2 Как индексируются элементы массива?
5.4.3 Как создать двумерный массивы?
5.4.4 Как организовать вложенные циклы?
5.4.5 Функции обработки массива.
5.4.6 Полиморфизм, что это такое?

6 Лабораторная работа № 6. Программирование кластеров

Цель работы: создание кластеров в LabView, знакомство с функциями обработки кластеров.

6.1 Задание к лабораторной работе

Задание 1. Создать ВП, который собирает и разбирает кластеры, используя функции обработки кластеров.

Задание 2. Создать ВП, который полученные с датчиков давления, расхода и температуры исходные данные в единицах напряжения масштабирует по известным коэффициентам и выдает фактические значения физических величин.

Задание 3. Создать ВП, который проверяет значение числового элемента управления во входном кластере большим или равным нулю.

6.2 Порядок выполнения лабораторной работы

6.2.1 Для выполнения лабораторной работы необходимо зарегистрироваться в системе:

- загрузить файл Base_LabView, появится окно выбора лабораторных работ (рисунок 1.1), в главном меню LabView нажать кнопку «RUN»;

- выбрать лабораторную работу «Программирование кластеров»;
- появится окно регистрации студента (рисунок 1.2): ввести свои Фамилию, Имя; ввести номер группы; нажать кнопку «Далее».

6.2.2 Кластеры в LabView, аналогично массивам, – это набор элементов управления или индикации данных. Отличие в том, что кластеры могут объединять элементы разных типов данных. Однако кластер не может одновременно содержать элементы управления и отображения данных.

В кластере, как и в массиве, все элементы упорядочены, но обратиться по индексу к ним нельзя, необходимо сначала разделить их. Для этого предназначены функции обработки кластеров.

6.2.2.1 Функции обработки кластеров.

«Bundle» («Bundle by Name») – сборка и управление кластерами (обращение не по номеру, а по имени элемента кластера).

«Unbundle» («Unbundle by Name») – разборка кластеров (обращение не по номеру, а по имени элемента кластера).

6.2.2.2 Функции взаимозаменяемости массивов и кластеров.

Заменять массивы в кластеры и наоборот бывает полезно, так как LabView имеет больше функций, работающих с массивами, чем с кластерами.

«Cluster to Array» - функция преобразования кластера 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 элементу в кластере. Эта функция не применяется, если в кластере в качестве элемента содержится массив.

«Array to Cluster» - функция преобразования *одномерного* массива с числом элементов N в кластер с числом элементов N *того же типа данных.* Если размерность входного массива превышает величину, установленную в окне размера кластера, то при использовании этой функции щелкнуть правой кнопкой мыши по терминалу «Array to Cluster» и выбрать «Cluster Size» для установления размера выходного кластера. Кластеры, в отличие от массивов, не устанавливают автоматически свой размер. По умолчанию размер кластера равен 9.

6.2.3 Задание 1. Создать ВП, который собирает и разбирает кластеры, используя функции обработки кластеров.

6.2.3.1 В открывшемся меню LabView выбрать «New VI» - новый ВП.

6.2.3.2 На фронт-панели собрать интерфейс задачи, как показано на рисунке 6.1.

1) Создать кластер «Кластер» из элементов управления:

- из меню «Controls – Modern - Array & Cluster» выбрать шаблон массива «Cluster» и поместить его на фронт-панель;

- заполнить *строго в указанной последовательности* шаблон кластера элементами управления: числовой элемент «Num Ctrl» из меню «Controls – Express - Numeric Controls» (назвать «Число»), логический элемент «Toggle

Switch» из меню «Controls – Express - Buttons» (назвать «Логический 1»), логический элемент «Rocker» из меню «Controls – Express - Buttons» (назвать «Логический 2»), числовой элемент «Dial» из меню «Controls – Express - Numeric Controls» (назвать «Шкала»).

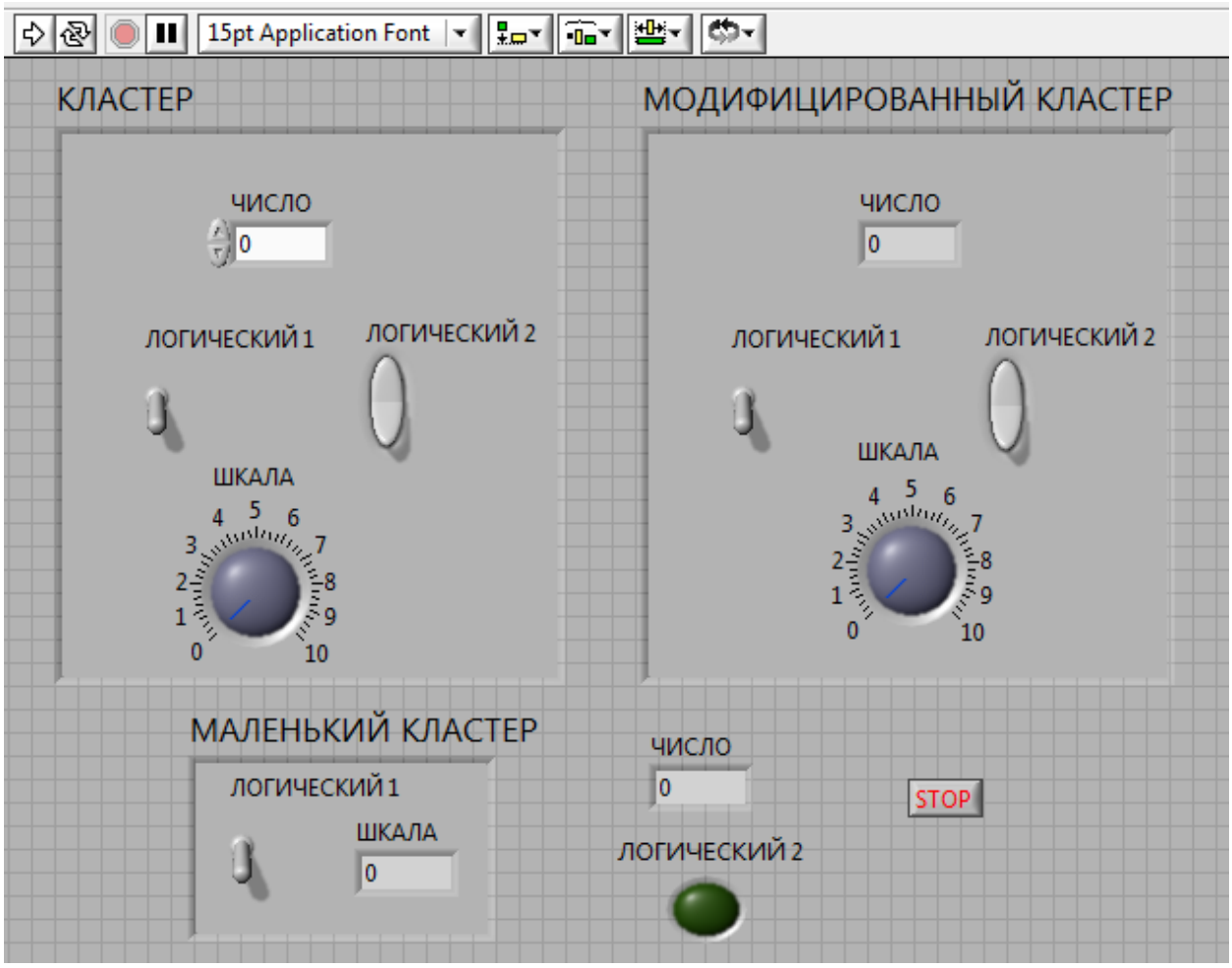


Рисунок 6.1

2) Скопировать «Кластер». Назвать копию «Модифицированный кластер». Щелкнуть правой кнопкой мыши на «Модифицированный кластер», выбрать из контекстного меню «Change to Indicator». Все элементы, входящие в модифицированный кластер, из элементов управления изменятся на элементы индикации данных.

3) Скопировать «Модифицированный кластер». Назвать копию «Маленький кластер». Удалить из маленького кластера элементы «Число» и «Логический 2». Щелкнуть правой кнопкой мыши на элемент «Шкала», выбрать из контекстного меню «Replace – Express – Numeric Indicators» элемент «Num Ind».

4) На фронт-панель поместить:

- числовой элемента индикации «Num Ind» из меню «Controls – Express - Numeric Indicators», назвать его «Число»;

- логический элемент индикации «Round LED» из меню «Controls – Express - LEDs» назвать его «Логический 2»;

- элемент «Stop Button» из меню «Controls – Express - Buttons».

5) Проверить порядковые номера элементов в кластерах «Кластер» и «Модифицированный кластер». Они должны совпадать, поскольку соответствующие элементы, определенные в кластерах одинаковыми порядковыми номерами, должны иметь совместимые типы данных.

Порядок элементов в кластере проверяется так: щелкнуть правой кнопкой мыши по границе шаблона каждого кластера, из контекстного меню выбрать «Reorder controls in cluster» (рисунок 6.2). В белом поле в нижнем правом углу каждого элемента указан текущий порядковый номер элемента, в черном – новый порядковый номер. Для установки другого порядкового номера нужно в поле ввода текста «Click to set to» ввести число и нажать на соответствующий элемент. Для сохранения изменений нажать в главном меню ☑.

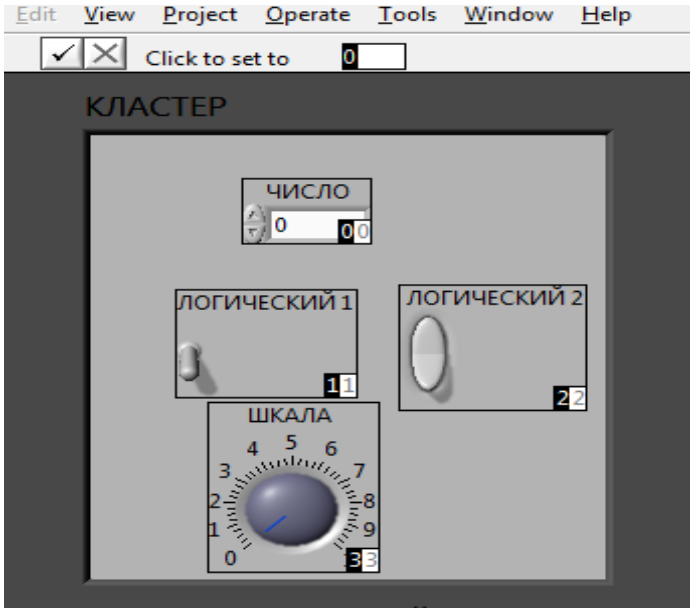


Рисунок 6.2

Первому помещенному в кластер элементу автоматически присваивается номер 0, второму – 1 и так далее. При удалении элемента порядковые номера автоматически изменяются. Порядок элементов в кластере определяет то, как элементы кластера будут распределены по терминалам функций «Bundle» и «Unbundle» на блок-диаграмме.

6.2.3.3 Перейти на блок-диаграмму и создать код программы, как показано на рисунке 3.3.

1) Для выделения элементов из кластера «Кластер» на блок-диаграмму поместить функции разборки кластера «Unbandle» и «Unbandle by Name».

2) Для сборки элементов в кластеры «Модифицированный кластер» и «Маленький кластер» на блок-диаграмму поместить функции сборки кластера «Bandle» и «Bandle by Name».

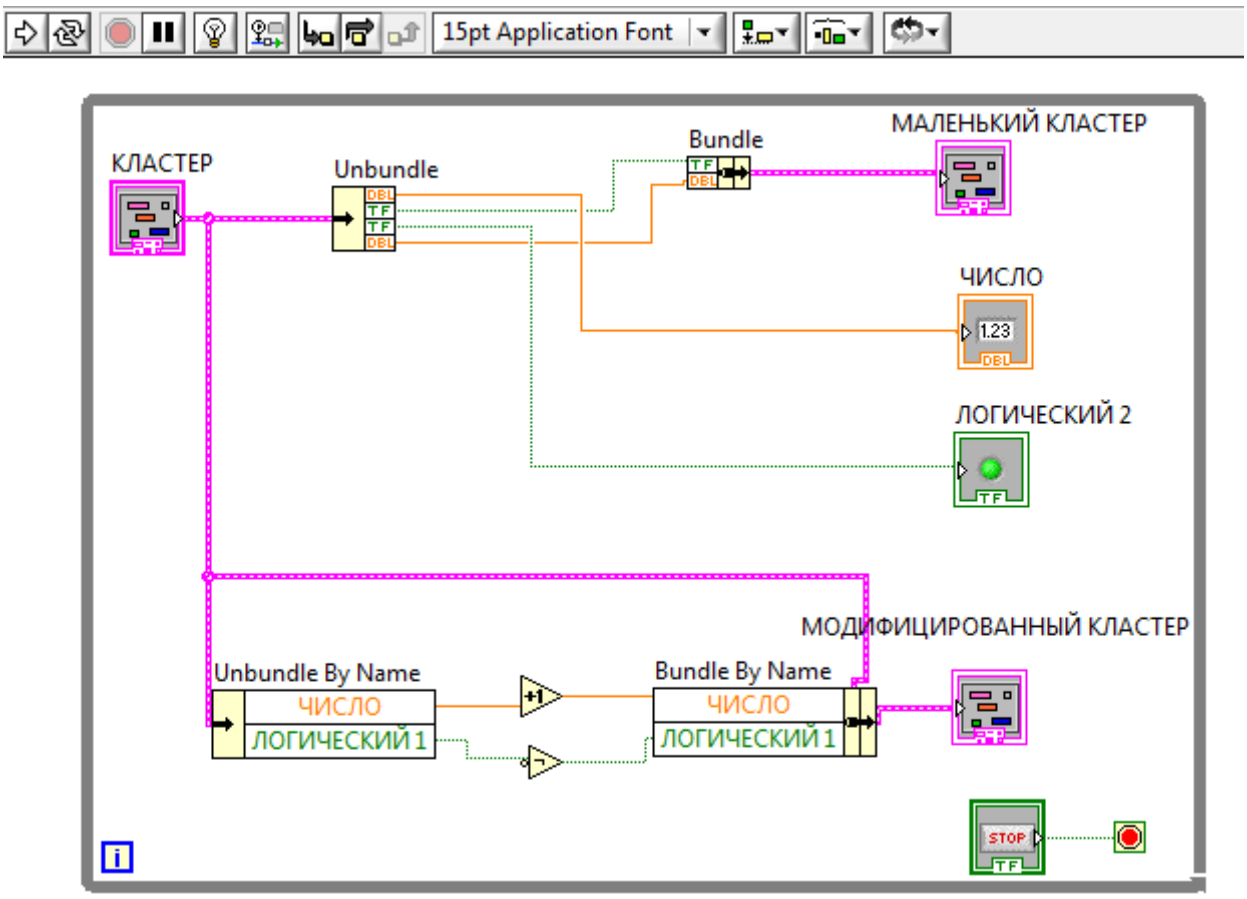


Рисунок 6.3

3) Разобрать «Кластер» для двух задач:

- для сборки двух элементов «Логический 1» и «Шкала» в «Маленький кластер» и индикации вне кластеров элементов «Число» и «Логический 2»;

- для сборки в «Модифицированный кластер» по следующему алгоритму: элемент «Логический 1» отображается инверсно элементу «Логический 1» из кластера «Кластер»; элемент «Число» отображает значение элемента «Число» из кластера «Кластер» плюс единица; элементы «Логический 2» и «Шкала» отображаются без изменений.

4) Поместить все элементы в цикл «While». Организовать выход из цикла по нажатию кнопки «Stop».

6.2.3.4 Сохранить ВП под именем «Работа с кластерами.vi».

6.2.3.5 Перейти на фронт-панель для ввода исходных данных в кластер «Кластер»: ввести значение «Число» = 25; установить переключатели «Логический 1» и «Логический 2» в положение «Вкл», установить значение «Шкала» = 6.

6.2.3.6 Запустить программу на выполнение с помощью однократного «RUN».

Результат выполнения программы:

- в «Модифицированном кластере» отобразится «Число» = 26; «Логический 1» - в положение «Выкл»; «Логический 2» - в положение «Вкл»; «Шкала» - положение 6;

- в «Маленьком кластере» отобразится числовое значение «Шкала» = 6; «Логический 1» - в положении «Вкл»;

- расположенные вне кластеров «Число» = 25; «Логический 2» - светодиод «Свет».

Проверить результат выполнения программы.

6.2.3.7 Изменить исходные данные в п.6.2.3.5. Запустить программу на выполнение с помощью однократного «RUN».

6.2.4 Задание 2. Создать ВП, который полученные с датчиков давления, расхода и температуры исходные данные в единицах напряжения масштабирует по известным коэффициентам и выдает фактические значения физических величин.

6.2.4.1 В открывшемся меню LabView выбрать «New VI» - новый ВП.

6.2.4.2 На фронт-панели собрать интерфейс задачи, как показано на рисунке 6.4.

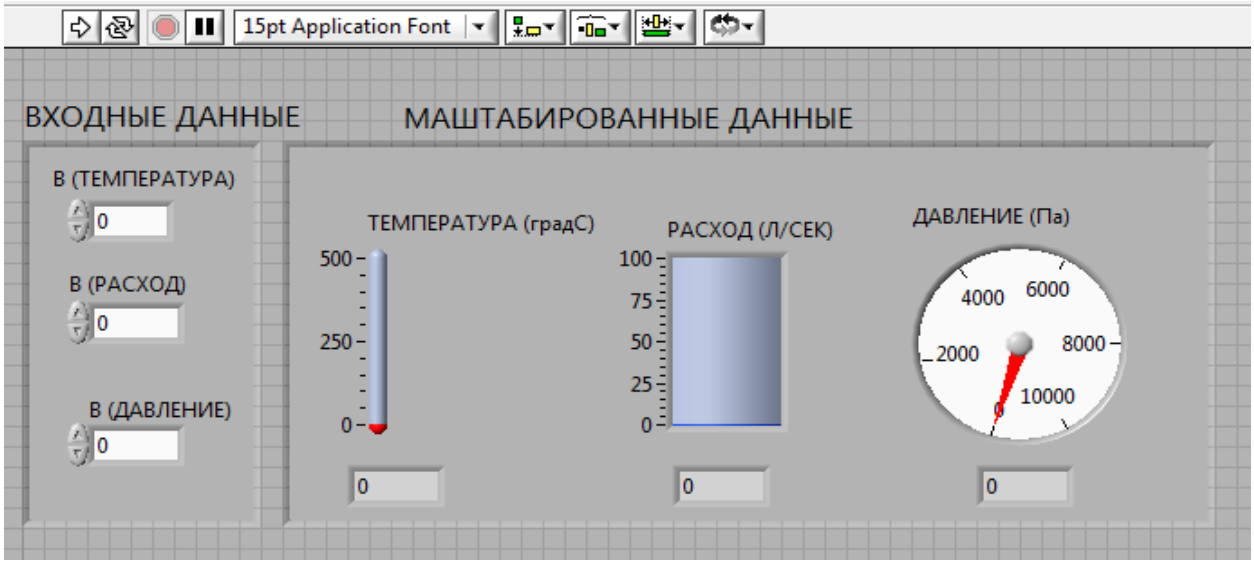


Рисунок 6.4

1) Создать кластер «Входные данные» из элементов управления:

- из меню «Controls – Modern - Array & Cluster» выбрать шаблон массива «Cluster» и поместить его на фронт панель;

- заполнить шаблон кластера элементами управления тремя числовыми элементами «Num Ctrl» из меню «Controls – Express - Numeric Controls». Назвать эти элементы *строго в указанной последовательности:* «Вольт (Температура)», «Вольт (Расход)», «Вольт (Давление)».

2) Создать кластер «Масштабированные данные» из элементов индикации из меню «Controls – Express - Numeric Indicators»:

- «Thermometer» назвать «Температура (град.С)»;

- «Tank» назвать «Расход (л/сек)»;

- «Gauge» назвать «Давление (Па)».

Для получения оцифрованных значений у каждого элемента в кластере «Масштабированные данные» щелкнуть правой кнопкой мыши по элементу, из контекстного меню выбрать «Visible Items – Digital Display», переместить окно «Digital Display», как показано на рисунке 6.4.

6.2.4.3 Перейти на блок-диаграмму и создать код программы, как показано на рисунке 6.5.

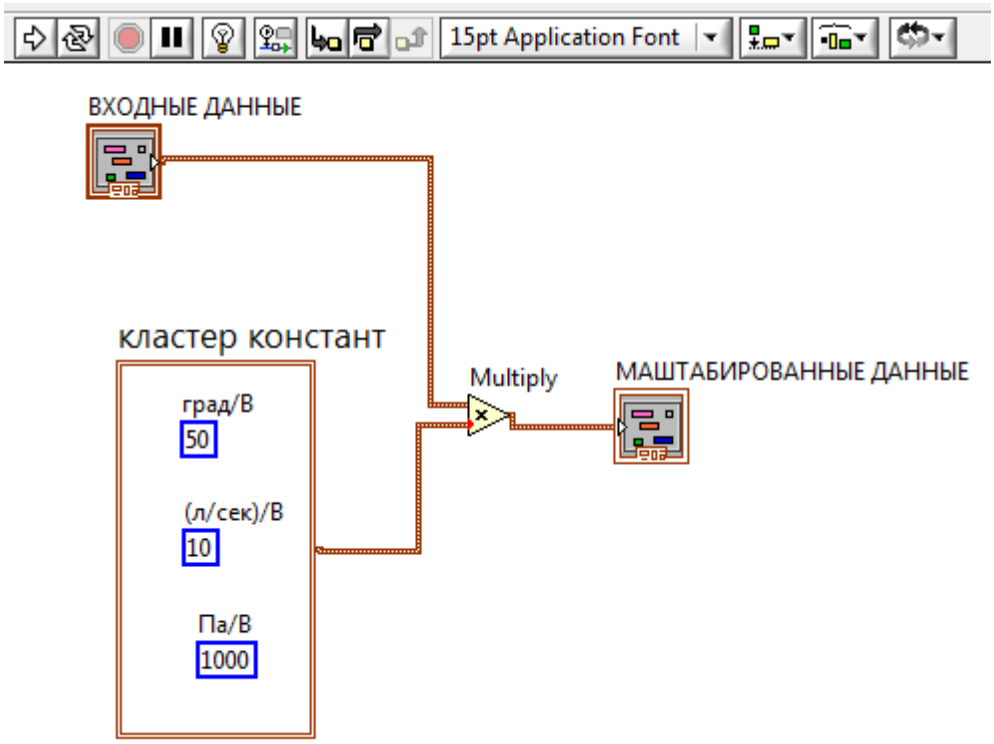


Рисунок 6.5

Для масштабирования кластера «Входные данные» необходимо поместить на блок-диаграмму «Кластер констант» с масштабными коэффициентами, перемножить эти кластеры и вывести результат в кластер «Масштабированные данные».

1) Поместить на блок-диаграмму шаблон «Cluster Const» из меню «Function – Programming – Cluster, Class & Variant».

2) Поместить (инициализировать) в шаблон три числовые константы «Numeric Const» из меню «Function – Programming - Numeric».

3) Первую константу (строго по очередности) назвать «град/В», вторую константу – «(л/сек)/В», третью константу – «Па/В».

4) Задать значения констант - масштабных коэффициентов: для температуры = 50 град.С/Вольт, для расхода = (10 л/сек)/Вольт, для давления = 1000 Па/Вольт.

6.2.4.4 Сохранить ВП под именем «Работа с кластерами.vi».

6.2.4.5 Перейти на фрон-панель для ввода исходных данных в кластер «Входные данные».

6.2.4.6 Запустить программу на выполнение с помощью однократного «RUN».

Проверить результат выполнения программы.

6.2.4.7 Изменить исходные данные в п.6.2.4.3. Запустить программу на выполнение с помощью однократного «RUN».

6.2.5 Задание 3. Создать ВП, в котором входной кластер состоит из четырех числовых элементов. ВП проверяет значение первого числового элемента управления во входном кластере большим или равным нулю.

Если оно меньше нуля, то ВП вычисляет абсолютные величины всех элементов входного кластера. Если значение первого числового элемента больше или равно нулю, то ВП оставит значения всех элементов входного кластера без изменений.

Далее, независимо от значения первого числового элемента, ВП умножает все величины входного кластера на 15 и отображает результаты в выходном кластере.

Для организации операции сравнения использовать функцию «Select» из палитры « Function - Programming - Comparison».

Задание выполнить самостоятельно.

6.3 Содержание отчета

Отчет должен содержать:

- цель и задание к лабораторной работе;
- screen shot интерфейсов и блок-диаграмм лабораторной работы для каждого задания;
- описание изучаемых операторов LabView;
- выводы по работе.

6.4 Контрольные вопросы

6.4.1 Что такое кластер?

6.4.2 Алгоритм создания кластера.

6.4.3 Как создать кластер констант?

6.4.4 Как определить порядок элементов в кластере?

6.4.5 Отличие функций «Bundle» и «Bundle by Name».

6.4.6 Отличие функций «Unbundle» и «Unbundle by Name».

6.4.7 Функции взаимозаменяемости массивов и кластеров.

7 Лабораторная работа №7. Основы графического отображения данных

Цель работы: изучить способы визуализации данных с помощью графика Диаграмм (Waveform Chart), графика Осциллограмм (Waveform Graph), двухкоординатного графика Осциллограмм (XY Graph).

7.1 Задание к лабораторной работе

Задание 1. Создать ВП, который использует три различных режима отображения данных графика Диаграмм (Waveform Chart).

Задание 2. Создать ВП, который одновременно отображает несколько графиков на графике Диаграмм (Waveform Chart).

Задание 3. Создать ВП, который измеряет температуру и отображает результат в виде графика на графике Диаграмм.

Задание 4. Создать ВП, который отображает массив данных на графике Осциллограмм.

Задание 5. Создать ВП, который отображает график окружности на двухкоординатном графике Осциллограмм (XY Graph).

7.2 Порядок выполнения лабораторной работы

7.2.1 Для выполнения лабораторной работы необходимо зарегистрироваться в системе:

- загрузить файл Base_LabView, появится окно выбора лабораторных работ (рисунок 1.1), в главном меню LabView нажать кнопку «RUN»;
- выбрать лабораторную работу «Основы графического отображения данных»;
- появится окно регистрации магистранта: ввести свои Фамилию, Имя; ввести номер группы; нажать кнопку «Далее».

В открывшемся меню LabView выбрать «New - Blank VI» - новый ВП.

7.2.2 Задание 1. Создать ВП, который использует три различных режима отображения данных графика Диаграмм (Waveform Chart).

7.2.2.1 На фронт-панели собрать интерфейс задачи, как показано на рисунке 7.1.

7.2.2.2 Поместить на фронт-панель элемент отображения данных «Chart» из меню «Express – Graf Indicators».

7.2.2.3 Скопировать этот элемент отображения дважды.

7.2.2.4 Заменить название элементов «Waveform Chart» на «Strip chart», «Scope chart», «Sweep chart», которые обозначают название трех режимов отображения данных на графике Диаграмм.

7.2.2.5 На панели управления графиком (Plot Legend) изменить название на «Температура», из контекстного меню (правой кнопкой мыши по Plot Legend) изменить вид линий (Common Plots), цвет линий (Color), стиль линий (Line Style), толщину линий (Line Width).

7.2.2.6 Задать на первом графике режим отображения данных «Strip chart» из контекстного меню (щелкнуть правой кнопкой мыши по экрану графика Диаграмм) «Advanced – Update Mode»; аналогично задать на втором графике режим «Scope chart»; на третьем графике - «Sweep chart».

Режим отображения данных «Strip chart» устанавливается по умолчанию.

7.2.2.7 Поместить на фронт-панель кнопку «Stop».

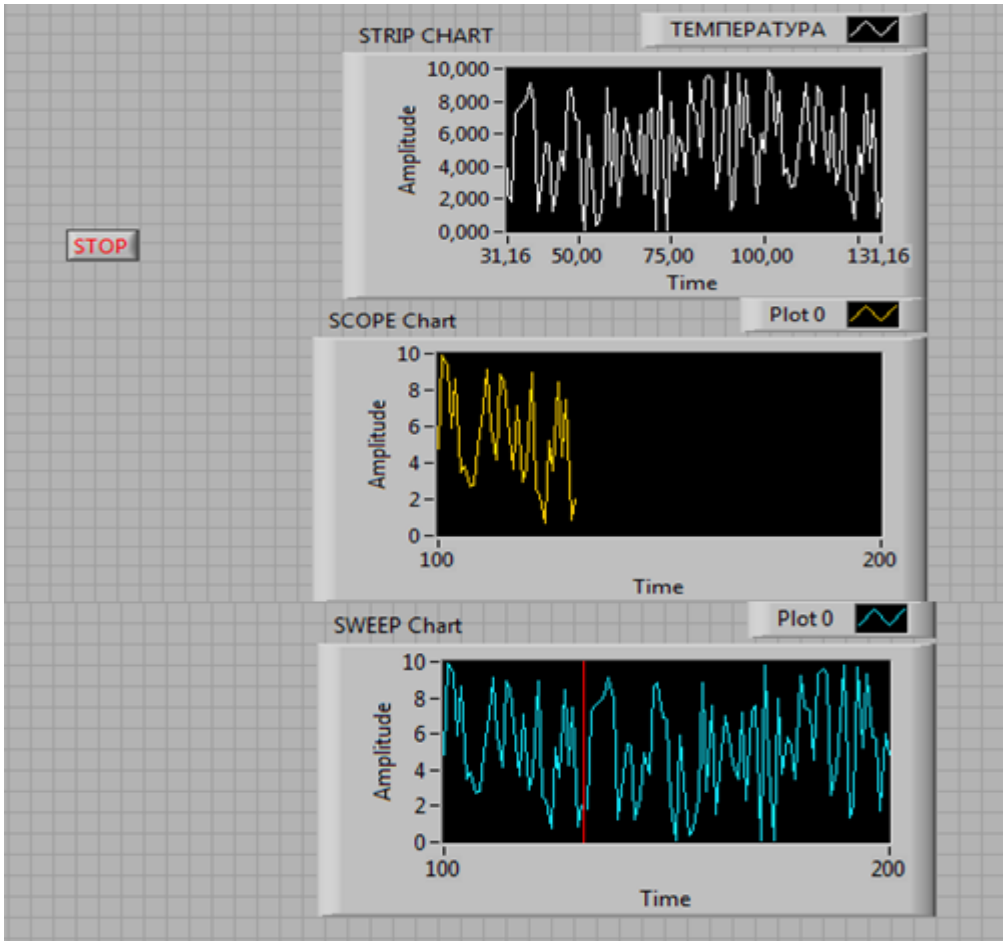


Рисунок 7.1

7.2.2.8 Перейти на блок-диаграмму и создать код программы, как показано на рисунке 7.2.

Поместить все элементы в цикл «While». Организовать выход из цикла по нажатию кнопки «Stop».

7.2.2.9 Сохранить ВП под именем «График Диаграмм.vi».

7.2.2.10 Запустить программу на выполнение с помощью однократного «RUN».

Результат выполнения программы:

- режим «Strip chart» представляет собой экран, прокручиваемый слева направо, подобно бумажной ленте;

- режим «Scope chart»: по достижении графиком правой границы поле экрана очищается, и заполнение диаграммы начинается с левой границы;

- режим «Sweep chart»: вертикальная линия – маркер – отделяет новые данные графика (слева от маркера) от старых данных (справа от маркера).

Режимы «Scope chart» и «Sweep chart» отличаются большей скоростью отображения данных по сравнению с «Strip chart».

7.2.2.11 Проанализировать результат выполнения программы.

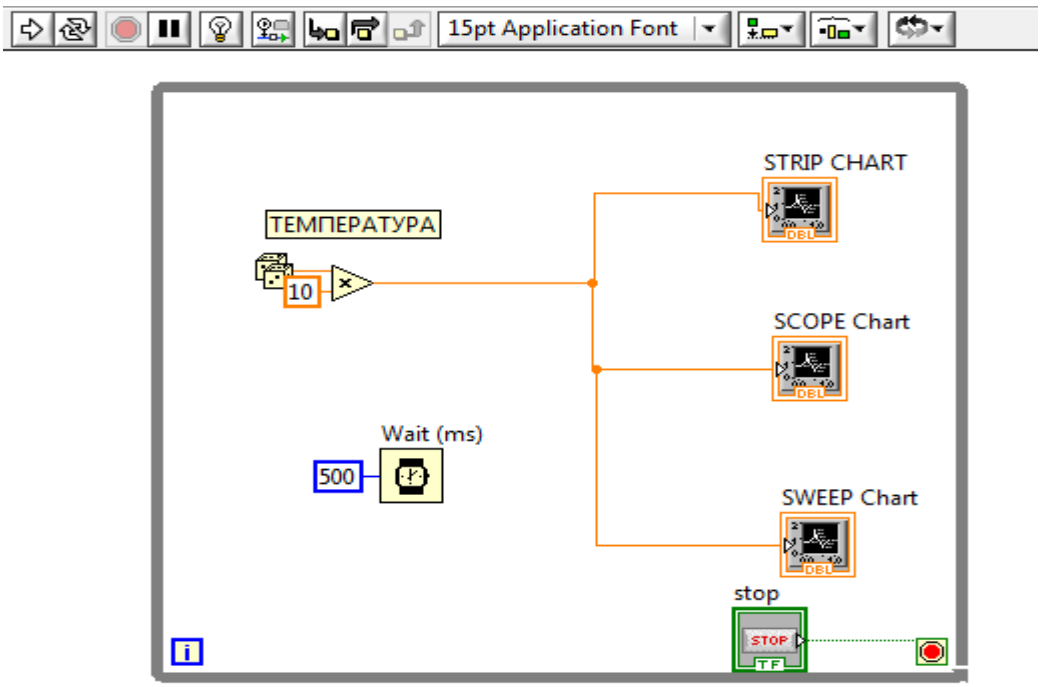


Рисунок 7.2

7.2.3 Задание 2. Выполнить самостоятельно.

Создать ВП, который одновременно отображает несколько графиков на графике Диаграмм (Waveform Chart).

Для объединения отображаемых графиков использовать функцию «Bundle», расположенную в меню «Functions – Programming - Cluster».

7.2.4 Задание 3. Создать ВП, который измеряет температуру и отображает результат в виде графика на графике Диаграмм.

7.2.4.1 Открыть новый бланк ВП, для чего в главном меню выбрать «File – New VI». На фронт-панели собрать интерфейс задачи, на блок - диаграмме собрать код программы, как показано на рисунке 7.3.

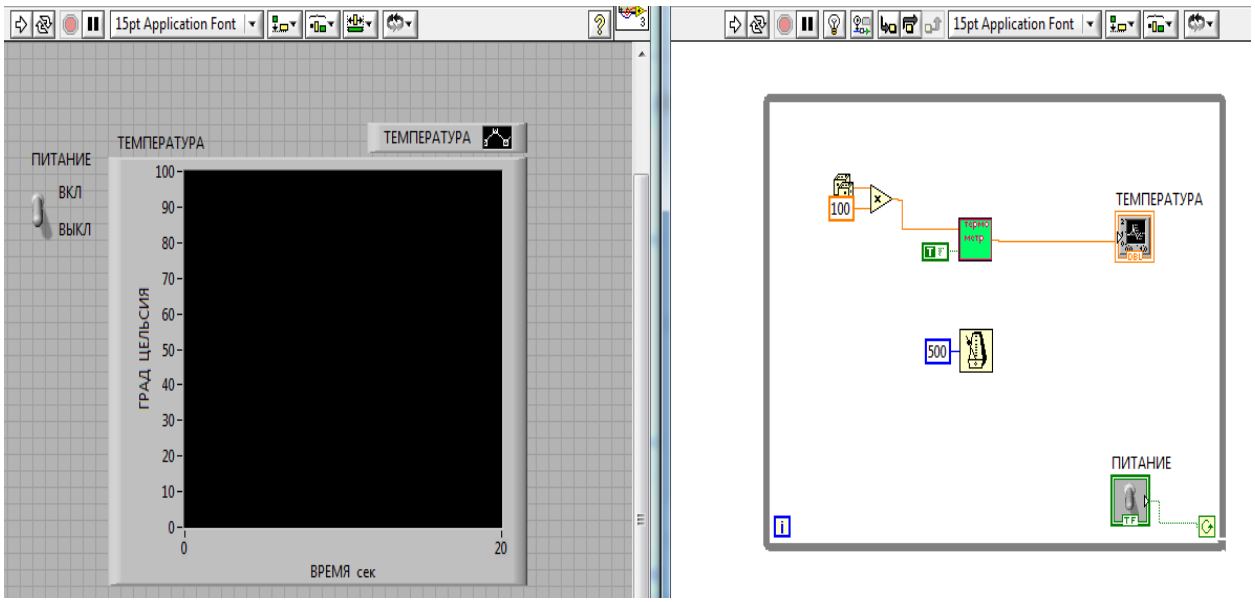


Рисунок 7.3

1) На блок-диаграмме для моделирования измеряемой температуры использовать п/п «Термометр.vi» из меню «Functions – Select a VI…».

2) Измеряемую температуру создать с помощью генератора случайных чисел, значения температуры от 0 до 100 С.

3) Выход из цикла «While» осуществлять включателем «Питание».

7.2.4.2 Сохранить ВП под именем «Мониторинг температуры».

7.2.4.3 Запустить программу на выполнение с помощью однократного «RUN».

7.2.4.4 Проанализировать и записать результат выполнения программы.

7.2.5 Задание 4. Создать ВП, который отображает массив данных на графике Осциллограмм.

Одиночный график Осциллограмм работает с одномерными массивами и представляет данные массива в виде точек на графике с приращением по оси X, равным 1, и началом в точке X=0. Графики также отображают кластеры, с установленным начальным значением X, ΔX и массивом данных по шкале У.

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

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

7.2.5.1 Открыть новый бланк ВП, для чего в главном меню выбрать «File – New VI». На фронт панели собрать интерфейс задачи, на блок - диаграмме собрать код программы, как показано на рисунке 7.4.

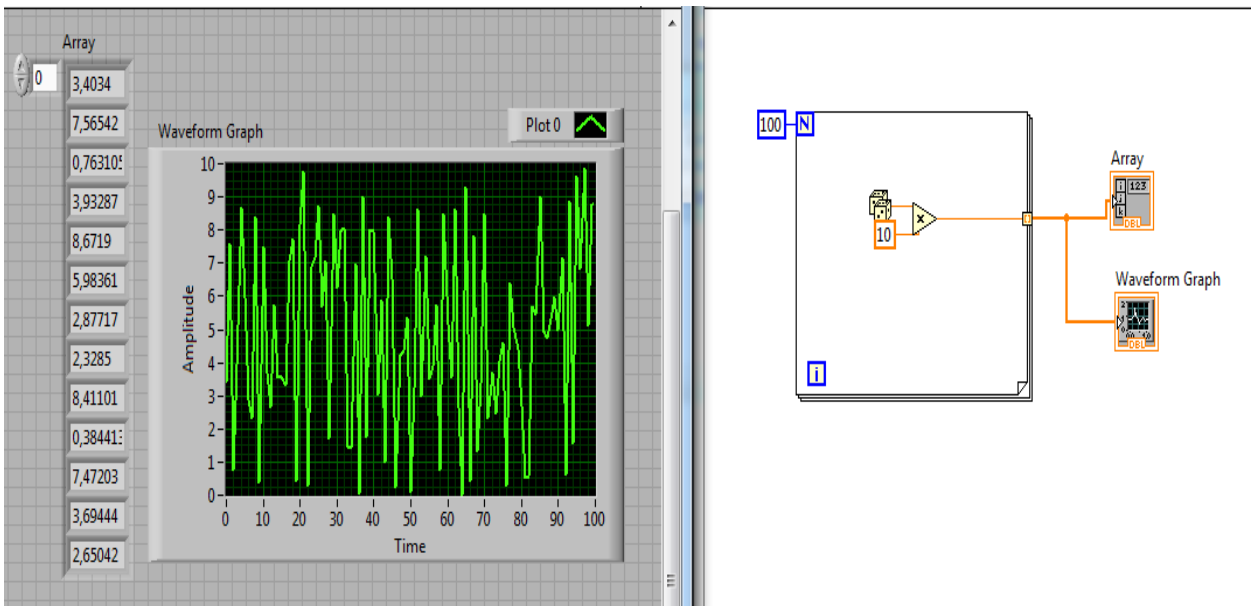


Рисунок 7.4

1) Поместить на фронт-панель шаблон массива, поместить в шаблон массива числовой элемент отображения, метка массива имеет имя «Array».

2) Поместить на фронт-панель график осциллограмм «Graph», расположенный в меню «Controls - Express – Graph Indicators».

3) На блок-диаграмме создать с помощью генератора случайных чисел одномерный массив в цикле «For», количество элементов массива = 100, значения данных массива от 0 до 10.

7.2.5.2 Сохранить ВП под именем «Вывод массива данных на график».

7.2.5.3 Запустить программу на выполнение с помощью однократного «RUN».

7.2.5.4 Проанализировать и записать результат выполнения программы.

7.2.5.5 Изменить блок-диаграмму, как показано на рисунке 7.5.

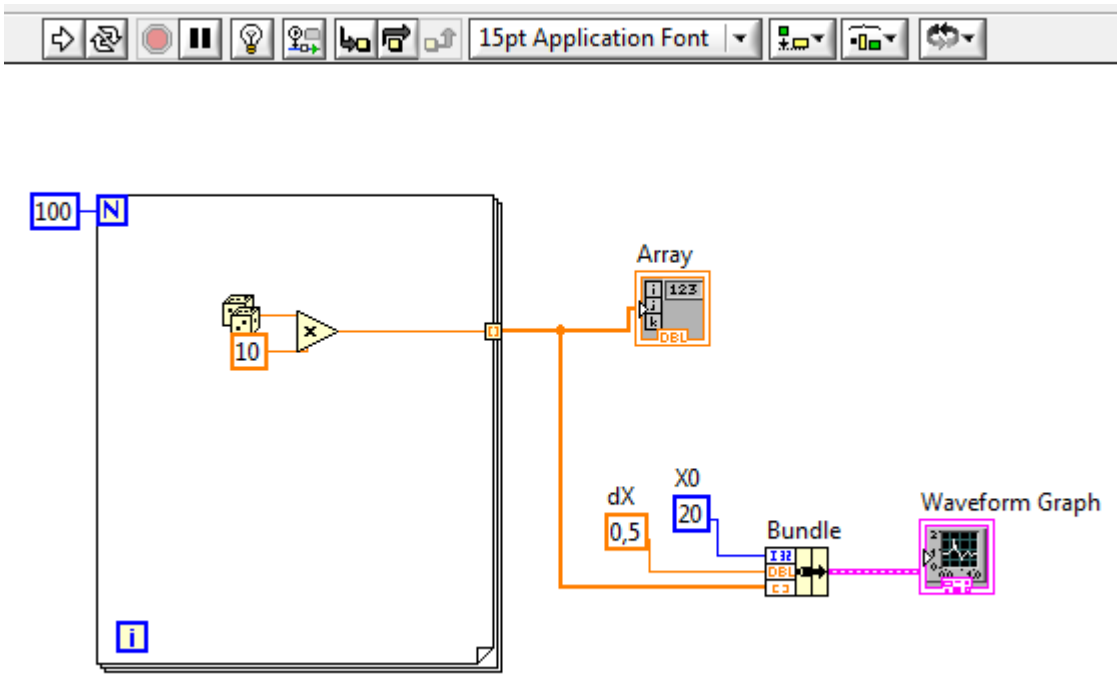


Рисунок 7.5

1) Для изменения точки начала графика X0 и приращения dX поместить на блок-диаграмму функцию «Bundle» из меню «Functions – Programming – Cluster».

2) Растянуть по вертикали функцию «Bundle» на три входа.

3) Поместить на блок-диаграмму константу «Numeric Constant» из меню «Functions – Programming – Numeric», назвать ее «X0»; скопировать эту константу, заменить имя на «dX»; соединить константу «X0» с первым (!) входом «Bundle», константу «dX» - со вторым входом «Bundle».

7.2.5.6 Сохранить ВП под именем «График Осциллограмм».

7.2.5.7 Запустить программу на выполнение с помощью однократного «RUN».

7.2.5.8 Проанализировать и записать результат выполнения программы. Сравнить полученный график с графиком в задании «Вывод массива данных на график».

7.2.6 Задание 5. Создать ВП, который отображает график окружности на двухкоординатном графике Осциллограмм (XY Graph).

Одиночный двухкоординатный график Осциллограмм работает с кластерами, содержащими массивы X и У. Двухкоординатный график Осциллограмм также воспринимает массивы точек, где каждая точка является кластером, содержащим значения по шкалам X и У.

Двухкоординатный график множества Осциллограмм работает с массивами осциллограмм, в которых осциллограмма является кластером, содержащем массивы X и У. Двухкоординатные графики множества Осциллограмм воспринимают также массивы множества осциллограмм, где каждая осциллограмма представляет собой массив точек. Каждая точка – это группа данных, содержащая значения по шкалам X и У.

7.2.6.1 Открыть новый бланк ВП, для чего в главном меню выбрать «File – New VI». На фронт-панели собрать интерфейс задачи, на блок - диаграмме собрать код программы, как показано на рисунке 7.6.

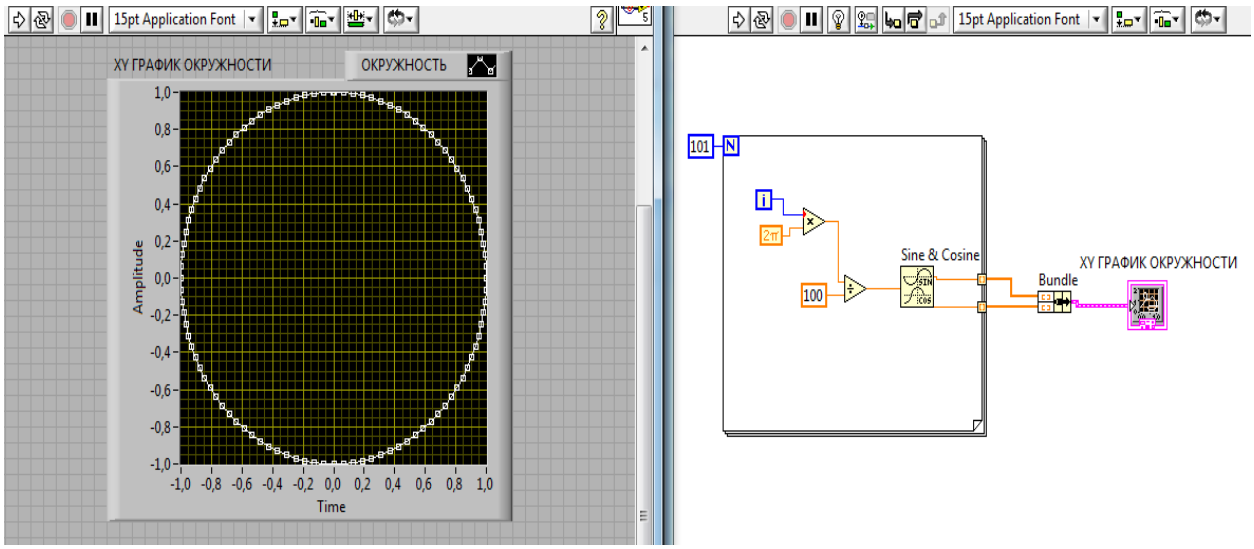


Рисунок 7.6

1) Поместить на фронт панель элемент «XY Graph», расположенный в меню «Controls - Express – Graph Indicators».

2) Изменить метки элемента, как показано на рисунке 7.6.

3) Поместить на блок-диаграмму функцию «Sin & Cos» из меню «Functions – Mathematics – Elementary & Special Functions – Trigonometric Functions»; поместить ее в цикл «For» и создать код программы, как на рисунке 7.6:

а) задать количество итераций цикла = 101;

б) умножить терминал счетчика итераций «i» на константу 2π (2π), расположенную в меню «Functions – Programming - Numeric – Math Constant»;

в) результат умножения разделить на 100 и подать на вход функции «Sin & Cos».

7.2.6.2 Сохранить ВП под именем «График Окружности».

7.2.6.3 Запустить программу на выполнение с помощью однократного «RUN».

7.2.6.4 Проанализировать и записать результат выполнения программы.

7.3 Содержание отчета

Отчет должен содержать:

- цель и задание к лабораторной работе;
- screen shots интерфейса всех заданий;
- разработанные блок-диаграммы ВП;
- описание элементов управления и индикации на блок-диаграмме;
- выводы по работе.

7.4 Контрольные вопросы

7.4.1 Три режима отображения графика Диаграмм.

7.4.2 Отличие отображения данных на графиках Диаграмм, Осциллограмм и двухкоординатном графике Осциллограмм.

7.4.3 Какие функции используются для усреднения четырех последних значений измеряемой величины?

7.4.4 Отличие одиночного графика Осциллограмм от графика множества Осциллограмм.

7.4.5 Отличие одиночного двухкоординатного графика Осциллограмм от двухкоординатного графика множества Осциллограмм.

8 Лабораторная работа №8. Графическое отображение данных при разработке виртуальных приборов

Цель работы: применить способы графического отображения данных при разработке виртуальных приборов.

8.1 Задание к лабораторной работе

Задание 1. Создать ВП, который использует сдвиговые регистры для реализации алгоритма «бегущее среднее».

Задание 2. Создать ВП, который измеряет температуру каждые 0,25 секунды в течение 10 секунд. В процессе измерения ВП в реальном масштабе времени отображает данные на графике Диаграмм и отображает аппроксимацию осциллограммы температуры.

Задание 3. Создать ВП, который измеряет температуру (задание 2) и отображает максимальное и минимальное значения температуры.

Задание 4. Создать ВП, который генерирует двумерный массив случайных чисел и отображает каждую строку на собственном графике.

8.2 Порядок выполнения лабораторной работы

8.2.1 Для выполнения лабораторной работы необходимо зарегистрироваться в системе:

- загрузить файл Base_LabView, появится окно выбора лабораторных работ (рисунок 1.1), в главном меню LabView нажать кнопку «RUN»;
- выбрать лабораторную работу «Графическое отображение данных при разработке виртуальных приборов»;
- появится окно регистрации студента: ввести свои Фамилию, Имя; ввести номер группы; нажать кнопку «Далее».

В открывшемся меню LabView выбрать «New - Blank VI» - новый ВП.

8.2.2 Задание 6. Создать ВП, который использует сдвиговые регистры для реализации алгоритма «бегущее среднее».

Использовать ВП «Мониторинг температуры», усреднить три последних значения измерений температуры и отобразить это среднее значение на диаграмме.

8.2.2.1 Откройте ВП «Мониторинг температуры», созданный в задании 4 данной лабораторной работы. Выберите пункт главного меню «File – Save As» и пересохраните ВП под именем «Расчет средней температуры».

8.2.2.2 На фронт панели собрать интерфейс задачи (рисунок 8.1), на блок - диаграмме собрать код программы, как показано на рисунке 8.2.

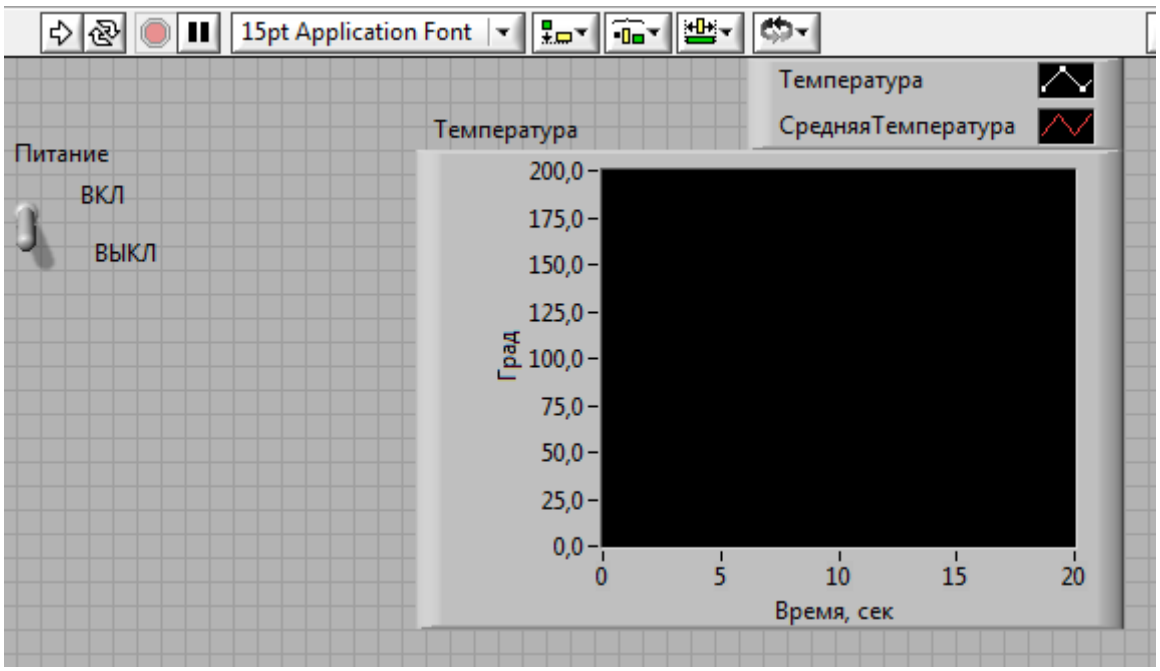


Рисунок 8.1

1) На интерфейсе добавить в «Plot Legend» график средней температуры, растянув вверх «Plot 0» - «Температура». Изменить параметры линии – цвет, тип и т.д.

2) Изменить блок-диаграмму, как показано на рисунке 8.2:

а) добавить сдвиговый регистр и у левого терминала создать стек сдвигового регистра из двух элементов;

б) поместить элемент «Compound Arithmetic» из меню «Functions – Programming - Numeric» для суммирования трех сигналов;

в) усредненное значение сигнала (после деления на три) подать на функцию «Bundle» для одновременного вывода двух графиков на «Waveform Chart».

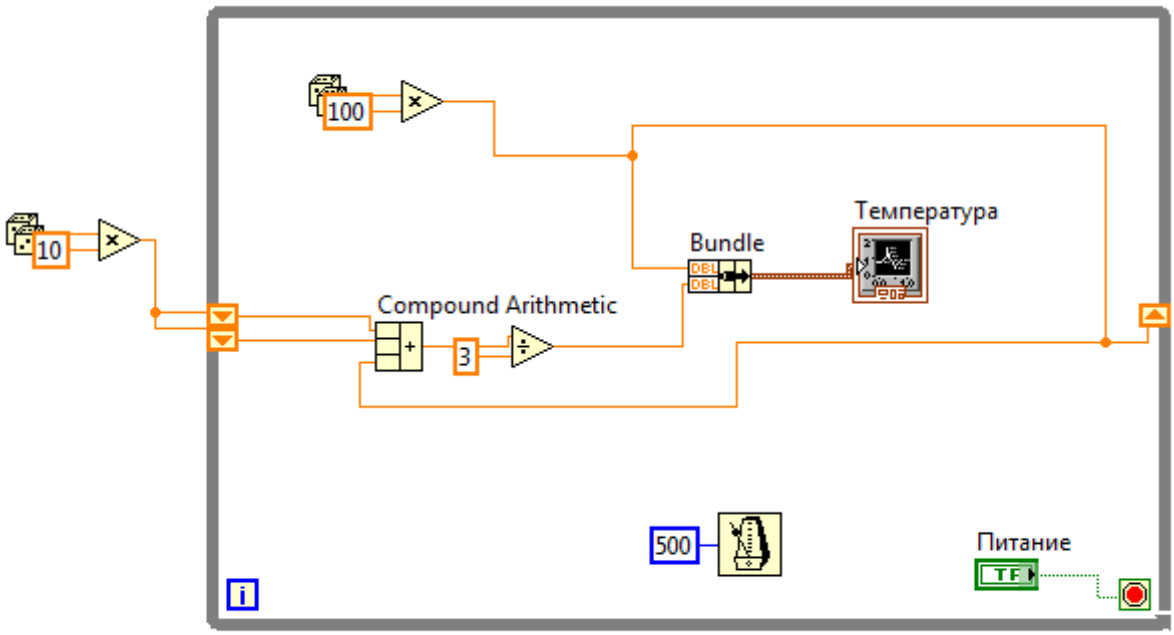


Рисунок 8.2

8.2.2.3 Запустить программу на выполнение с помощью однократного «RUN».

8.2.2.4 Проанализировать и записать результат выполнения программы.

8.2.3 Задание 7. Выполнить самостоятельно.

Создать ВП, который измеряет температуру каждые 0,25 секунды в течение 10 секунд. В процессе измерения ВП в реальном масштабе времени отображает данные на графике Диаграмм. После завершения измерений, ВП выводит данные на график Осциллограмм. Также ВП должен отображать аппроксимацию осциллограммы температуры.

8.2.4 Задание 3. Создать ВП, который измеряет температуру (аналогично заданию 2) и отображает максимальное, минимальное и среднее значения температуры.

Задание выполнить самостоятельно.

8.2.5 Задание 4. Создать ВП, который генерирует двумерный массив (три строки по пять столбцов) случайных чисел и отображает каждую строку на собственном графике. Лицевая панель должна содержать три графика.

Задание выполнить самостоятельно.

8.3 Содержание отчета

Отчет должен содержать:
- цель и задание к лабораторной работе;
- screen shots интерфейса всех заданий;
- разработанные блок-диаграммы ВП;
- описание элементов управления и индикации на блок-диаграмме;
- выводы по работе.

8.4 Контрольные вопросы

8.4.1 Какой элемент отображения данных используется для вывода массивов?

8.4.2 Функции поиска максимального и минимального значений массива.

8.4.3 Как отобразить на одном графике Диаграмм данные нескольких графиков?

8.4.4 Какие функции используются для усреднения четырех последних значений измеряемой величины?

8.4.5 Отображение аппроксимации графика осциллограммы.

9 Лабораторная работа №9. Работа со структурами. Принятие решений в ВП

Цель работы: изучить методы принятия решений в ВП, такие как структура «Case».

9.1 Задание к лабораторной работе

Задание 1. Создать ВП, который реализует три различных *типа* структуры «Case».

Задание 2. Создать ВП, который проверяет входное число на знак, вычисляет его квадратный корень или выдает сообщение об ошибке, если число отрицательное.

Задание 3. Создать ВП, который, используя ВП «Расчет средней температуры», определяет моменты выхода температуры за границы заданного верхнего предела и отображает наибольшее значение температуры.

Задание 4. Создать ВП, который использует структуру «Узел формул» для решения уравнения $Y=A^3+A$; $A= tg(x)^3 +\cos(x)$ и выводит результаты в виде графика.

9.2 Порядок выполнения лабораторной работы

9.2.1 Для выполнения лабораторной работы необходимо зарегистрироваться в системе:

- загрузить файл Base_LabView, появится окно выбора лабораторных работ (рисунок 1.1), в главном меню LabView нажать кнопку «RUN»;
- выбрать лабораторную работу «Работа со структурами. Принятие решений в ВП»;
- появится окно регистрации студента: ввести свои Фамилию, Имя; ввести номер группы; нажать кнопку «Далее».

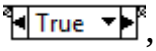
В открывщемся меню LabView выбрать «New - Blank VI» - новый ВП.

9.2.2 Задание 1. Создать ВП, который реализует три различных *типа* структуры «Case».

Структура «Case» аналогична логическим операторам «if … then … else» в текстовых языках программирования.

Структура «Case» имеет два или более *подпрограммы* вариантов. В данный момент времени видна только одна подпрограмма варианта, которая выполняется данной структурой.

Структура «Case» содержит:

- *селектор структуры* 

, расположенный сверху графического изображения Структуры, состоит из указателя значения варианта в центре и стрелок прокрутки для просмотра возможных вариантов по сторонам;
- *терминал селектора варианта* , расположенный на левой границе Структуры, определяет какой вариант структуры будет выполняться. Терминал варианта может быть трех типов: логический, целочисленный и строковый. Если терминал варианта логического типа, то селектор структуры содержит два логических варианта «True» и «False». Если терминал варианта имеет целочисленный или строковый тип, то селектор структуры может содержать до 2^{31}-1 вариантов;
- *контекстное меню структуры Case* (щелчок правой кнопки мыши по границе) содержит функции: добавить вариант до (Add Case Before), после (Add Case After) текущего варианта, дублировать (Duplicate Case), удалить (Delete This Case) вариант, отметить вариант по умолчанию (Make This The Default Case) и другие.

9.2.2.1 На фронт-панели собрать интерфейс задачи, как показано на рисунке 9.1.

9.2.2.2 Перейти на блок-диаграмму и поместить на ней структуру «Case Structure» из меню «Functions – Programming - Structures» (рисунок 9.2).

По умолчанию «Case Structure» - логического типа, что показывает селектор Структуры «True».

1) Соединить все элементы с границей Структуры, как показано на рисунке 9.2. На границе появятся терминалы входа (слева) и терминал выхода

(справа). Терминал выхода будет белого цвета, пока не будет создан код программы варианта «False».

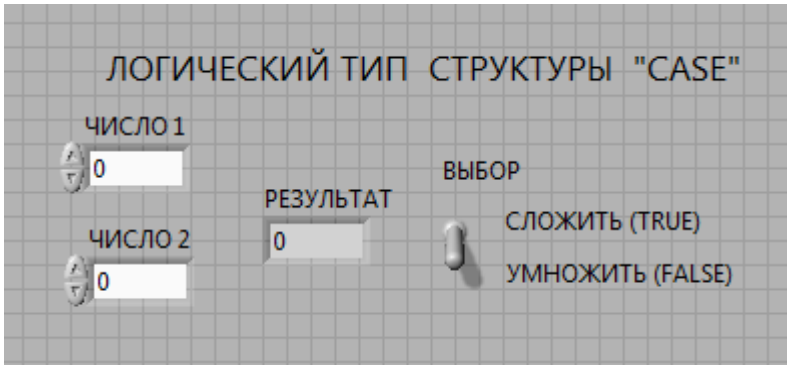


Рисунок 9.1

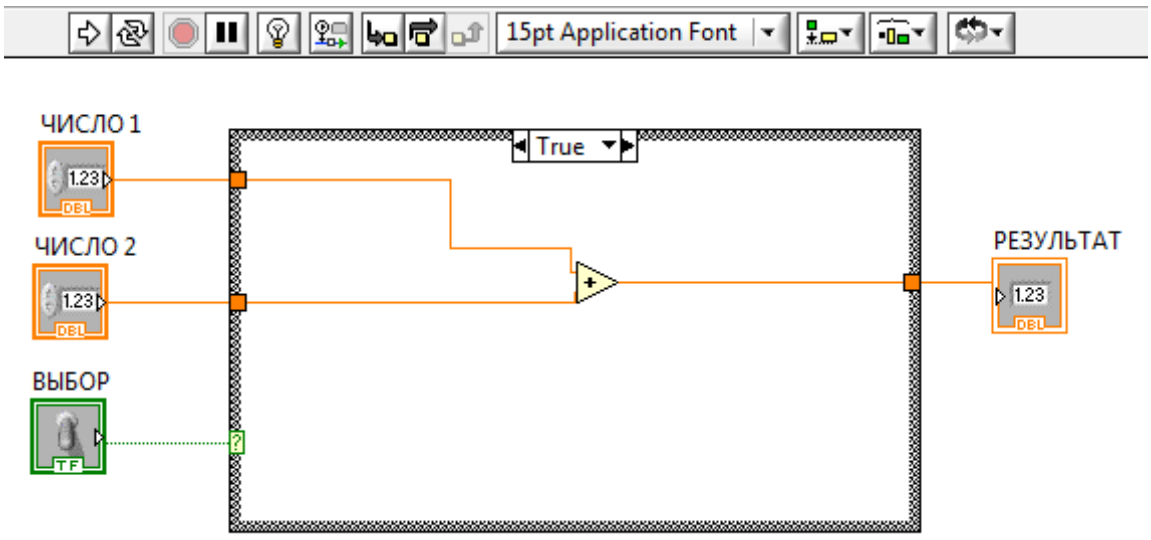


Рисунок 9.2

2) Изменить указатель значения варианта на «False». Создать код программы варианта «False», как показано на рисунке 9.3.

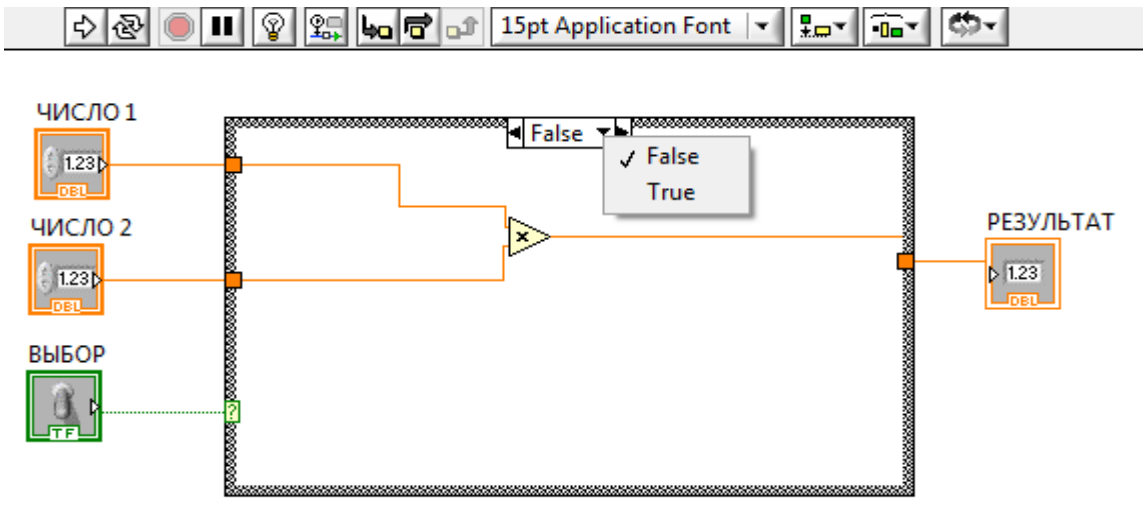


Рисунок 9.3

После соединения функции «Умножить» с терминалом выхода Структуры, терминал окрасится в цвет числового типа данных (оранжевый).

9.2.2.3 Запустить программу на выполнение с помощью «RUN».

9.2.2.4 Проанализировать и записать результат выполнения программы.

9.2.2.5 На той же фронт-панели собрать интерфейс «Числовой тип структуры Case», как показано на рисунке 9.4.

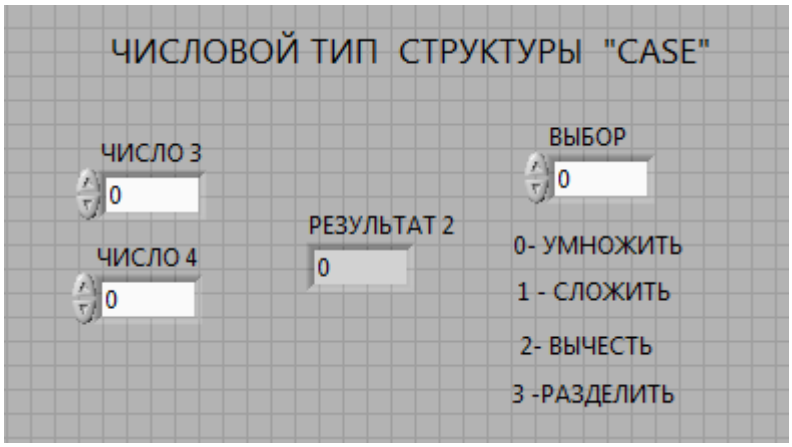


Рисунок 9.4

Элемент «Выбор» – числовой элемент управления «Num ctrl».

9.2.2.6 Перейти на блок-диаграмму и поместить на ней структуру «Case Structure» из меню «Functions – Programming - Structures» (рисунок 9.5).

При подсоединении числового элемента «Выбор» к терминалу селектора варианта, структура «Case Structure» примет числовой тип, что показывает селектор Структуры: задано четыре варианта структуры - 0,1,2,3.

9.2.2.7 Собрать подпрограммы вариантов: «0 - УМНОЖИТЬ», «1 - СЛОЖИТЬ», «2 - ВЫЧЕСТЬ», «3 - РАЗДЕЛИТЬ» в каждом окне структуры, которое создается с помощью функций «Add Case After» или «Add Case Before».

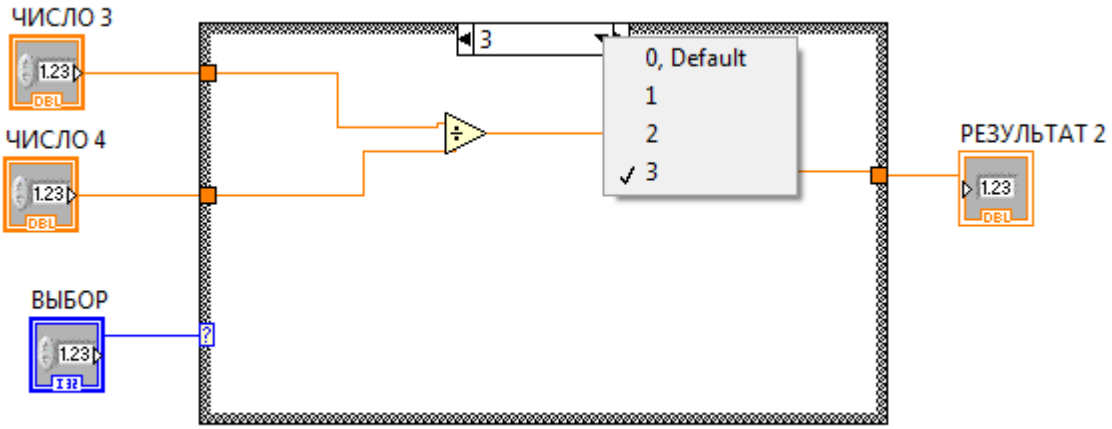


Рисунок 9.5

9.2.2.8 Запустить программу на выполнение с помощью однократного «RUN».

9.2.2.9 Проанализировать и записать результат выполнения программы.

9.2.2.10 На той же фронт-панели собрать интерфейс «Строковый тип структуры Case», как показано на рисунке 9.6.

Элемент «Выбор» – строковый элемент управления «String ctrl».

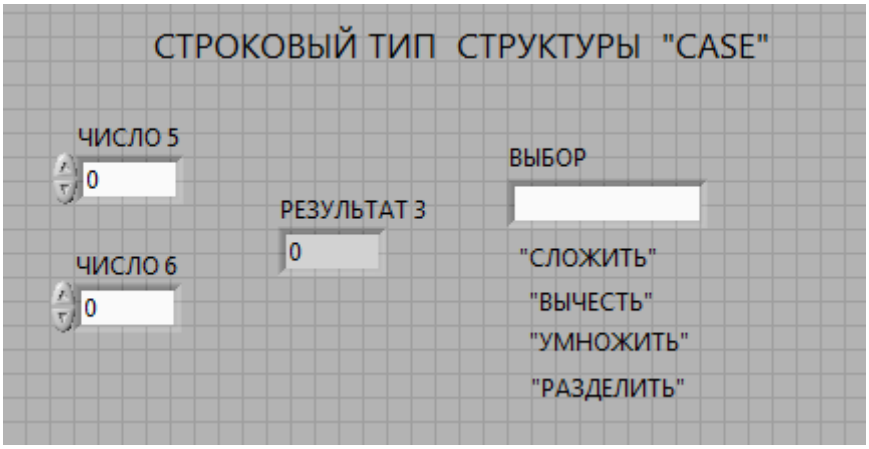


Рисунок 9.6

9.2.2.11 Перейти на блок-диаграмму и поместить на ней структуру «Case Structure» (рисунок 9.7).

При подсоединении строкового элемента «Выбор» к терминалу селектора варианта, структура «Case Structure» примет строковый тип, что показывает селектор Структуры: задано четыре варианта структуры: «”СЛОЖИТЬ”», «”ВЫЧЕСТЬ”», «”УМНОЖИТЬ”», «”РАЗДЕЛИТЬ”».

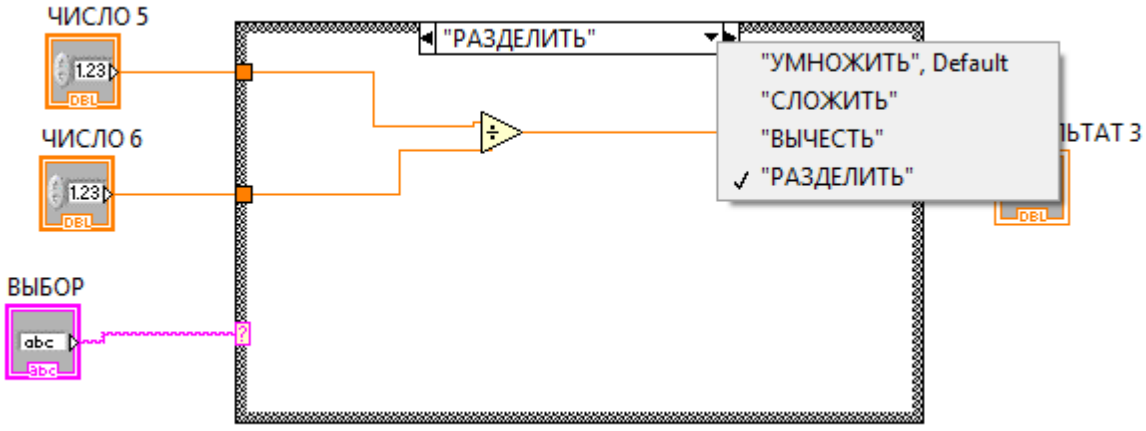


Рисунок 9.7

9.2.22.12 Собрать подпрограммы вариантов в каждом окне структуры.

9.2.2.13 Запустить программу на выполнение с помощью однократного «RUN».

9.2.2.14 Проанализировать и записать результат выполнения программы.

9.2.2.15 На той же фронт-панели собрать интерфейс «Структура по перечислениям», как показано на рисунке 9.8.

Элемент «Выбор» – числовой элемент управления «Menu Ring» из меню «Controls – Text Control». Для задания вариантов выбора щелкнуть правой кнопкой мыши на элемент «Выбор», из контекстного меню выбрать свойства «Properties». В открывшемся окне свойств (рисунок 9.9) открыть закладку «Edit Items». В строки таблицы «Items» вводить названия вариантов, для ввода варианта в конце каждой строки нажать «Insert». По окончании ввода названий вариантов нажать «Ok».

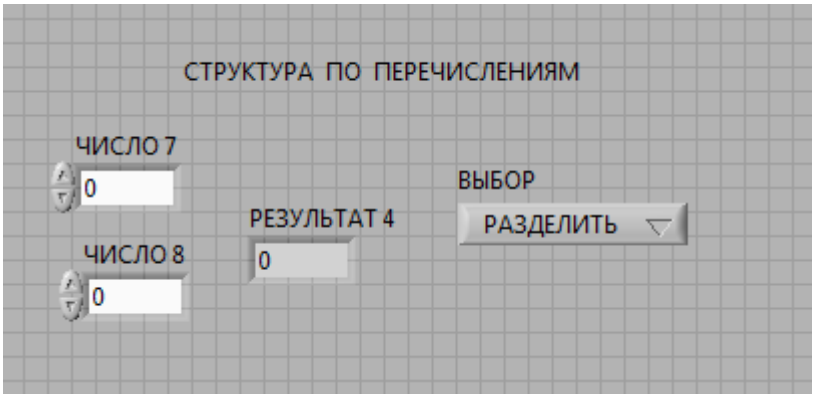


Рисунок 9.8

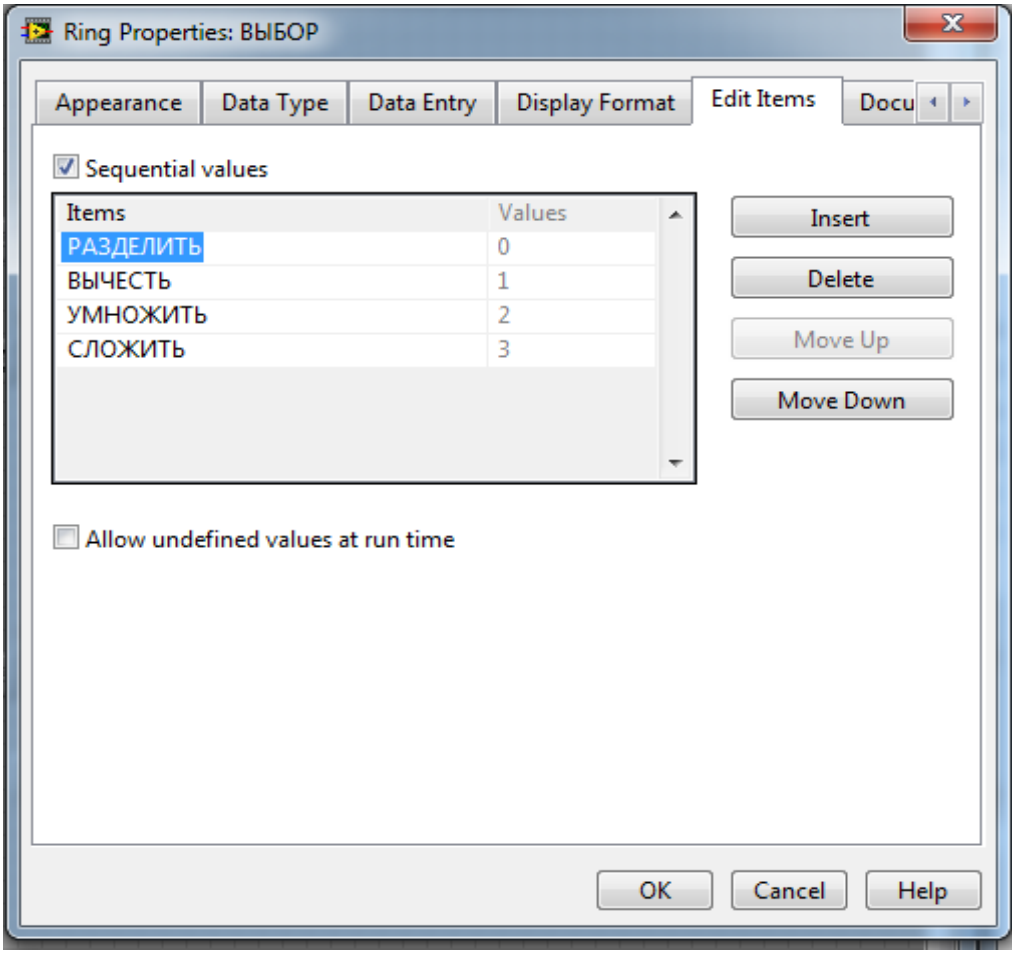


Рисунок 9.9

9.2.2.16 Перейти на блок-диаграмму, собрать код программы, как показано на рисунке 9.10.

9.2.2.17 Собрать подпрограммы вариантов в каждом окне структуры.

9.2.2.18 Запустить программу на выполнение с помощью однократного «RUN».

9.2.2.19 Проанализировать и записать результат выполнения программы.

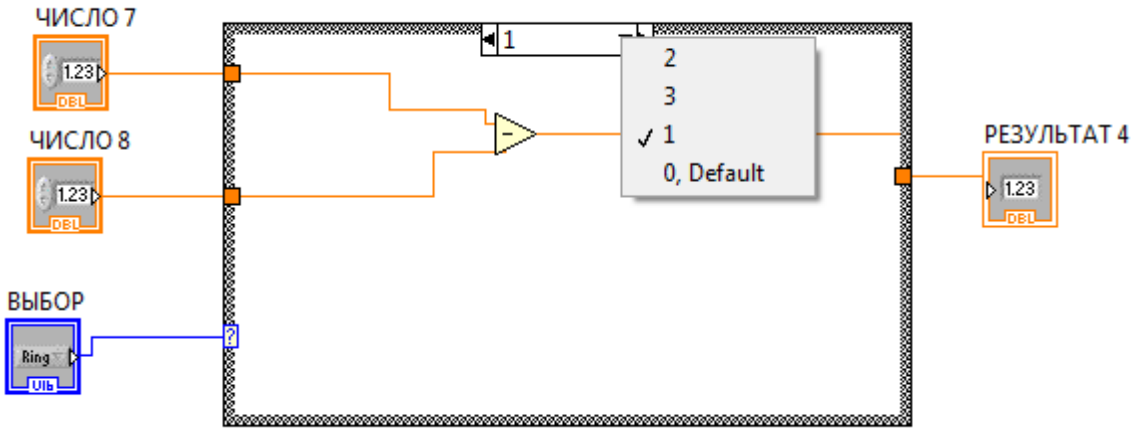


Рисунок 9.10

9.2.3 Открыть новый бланк ВП, для чего в главном меню выбрать «File – New VI».

Задание 2. Создать ВП, который проверяет входное число на знак, вычисляет его квадратный корень или выдает сообщение об ошибке, если число отрицательное.

9.2.3.1 На фронт панели собрать интерфейс задачи, как показано на рисунке 9.11.

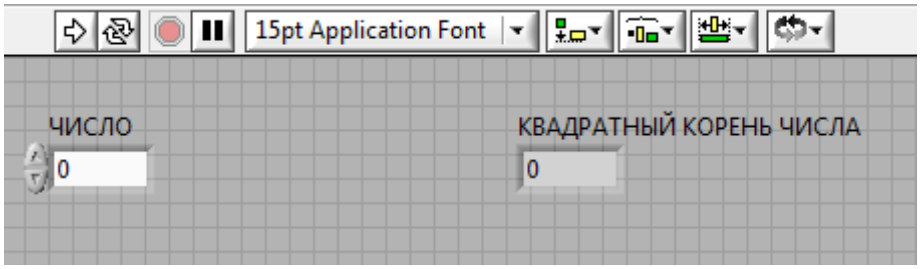


Рисунок 9.11

Элемент «Число» - числовой элемент управления, элемент «Квадратный корень числа» - числовой элемент отображения (индикации).

9.2.3.2 Перейти на блок-диаграмму и создать код программы для вариантов «True» и «False», как показано на рисунках 9.12 и 9.13.

1) В окне варианта «True» для извлечения квадратного корня из входного числа поместить функцию «Square Root» из меню «Functions – Programming - Numeric».

2) Для проверки числа на знак (положительное или отрицательное) поместить функцию «Greater Or Equal To 0» из меню «Functions – Programming - Comparison».

3) В окне варианта «False» для выдачи сообщения пользователю об ошибке (в случае извлечения корня из отрицательного числа) поместить функцию «One Button Dialog» из меню «Functions – Programming – Dialog & User Interface». Далее на входе «Message» этой функции создать константу («Create - Constant») и заполнить ее сообщением «ОШИБКА!!! ОТРИЦАТЕЛЬНОЕ ЧИСЛО».

4) В окне «False» выходной терминал структуры белого цвета. Для исключения ошибки создать числовую константу, например 0.

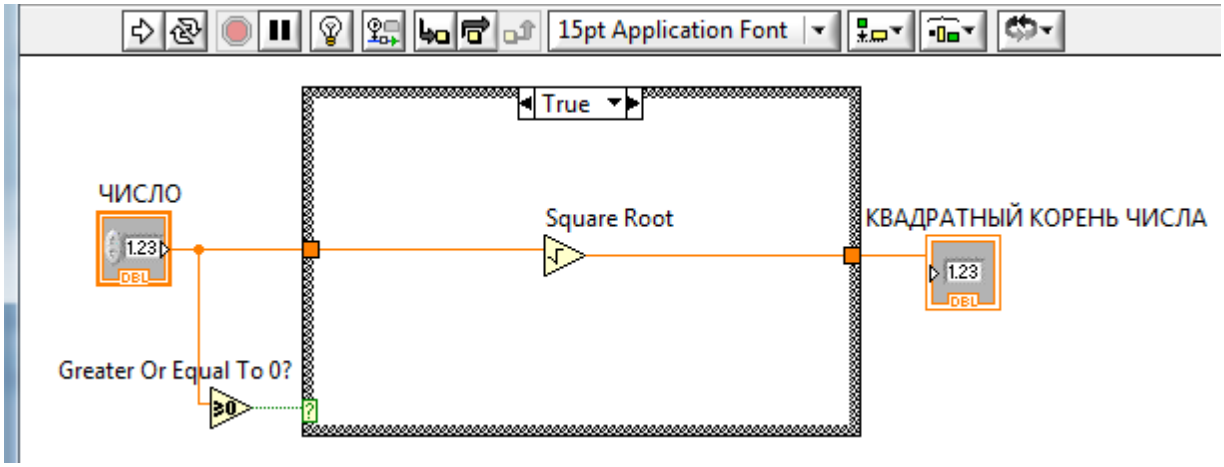


Рисунок 9.12

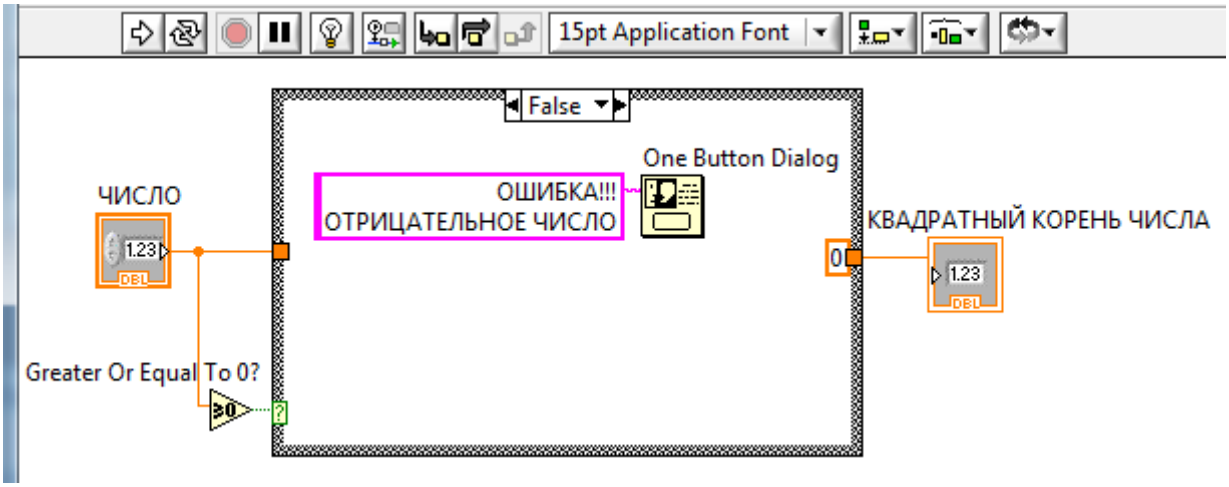


Рисунок 9.13

9.2.3.3 Сохранить созданный ВП под именем «Извлечение квадратного корня».

9.2.3.4 ВНИМАНИЕ! Не запускать программу *циклическим* «RUN».

Запустить программу на выполнение с помощью *однократного* «RUN».

9.2.3.5 Проанализировать и записать результат выполнения программы.

9.2.4 Открыть новый бланк ВП, для чего в главном меню выбрать «File – New VI».

Задание 3. Создать ВП, который, используя ВП «Расчет средней температуры», определяет моменты выхода температуры за границы заданного верхнего предела и отображает наибольшее значение температуры.

9.2.4.1 Открыть ВП «Расчет средней температуры» и пересохранить его под именем «Контроль температуры».

9.2.4.2 Изменить фронт-панель, как показано на рисунке 9.14.

1) Поместить декоративный элемент «Horizontal Smooth Box» из меню «Controls – Modern - Decorations». На нем сгруппировать элементы анализа:

а) переключатель «Анализ» включает и отключает режим анализа;

б) числовой элемент управления «Верхний предел» задает границу измеряемой температуры, при превышении которой зажигается светодиод «Внимание!»;

в) числовой элемент индикации «Верхнее значение» отображает текущее максимальное значение измеряемой температуры.

2) Для цифрового отображения текущих значений измеряемой температуры и средней температуры щелкнуть правой кнопкой мыши по экрану диаграммы, из контекстного меню выбрать «Visible Items – Digital Display».

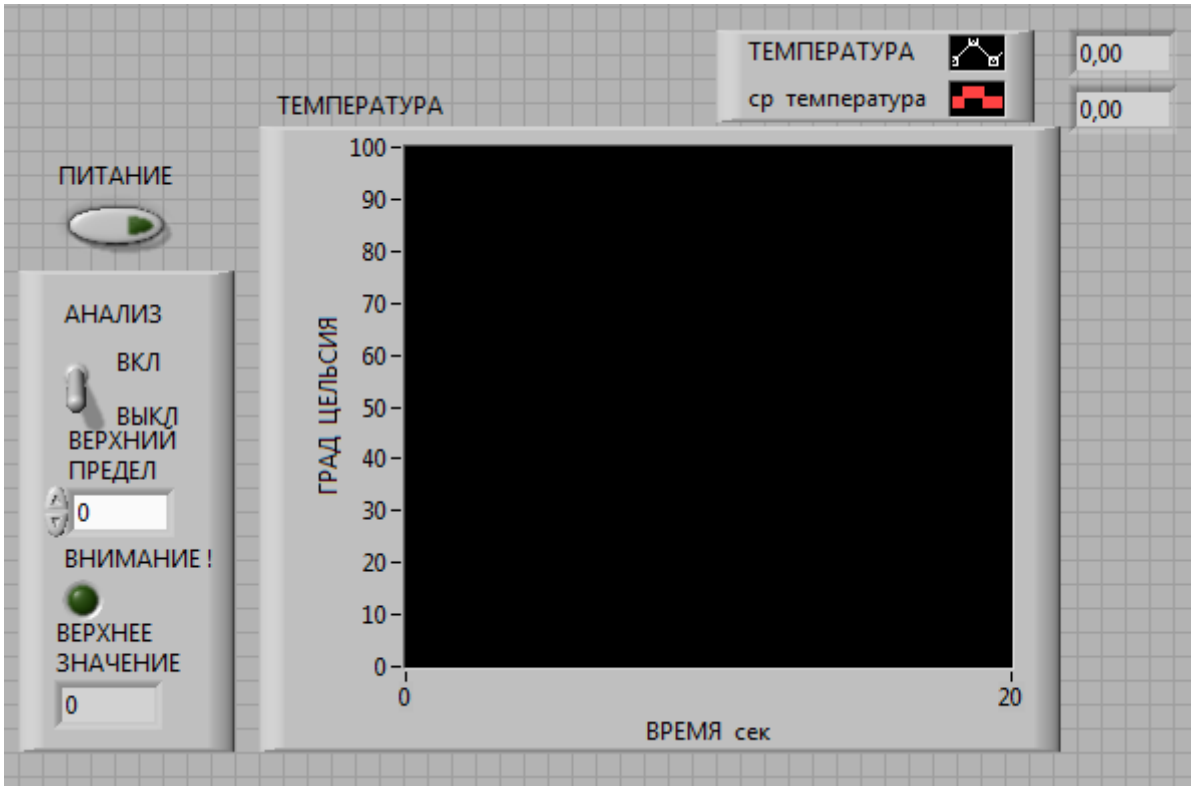


Рисунок 9.14

9.2.4.3 Изменить блок-диаграмму, как показано на рисунке 9.15.

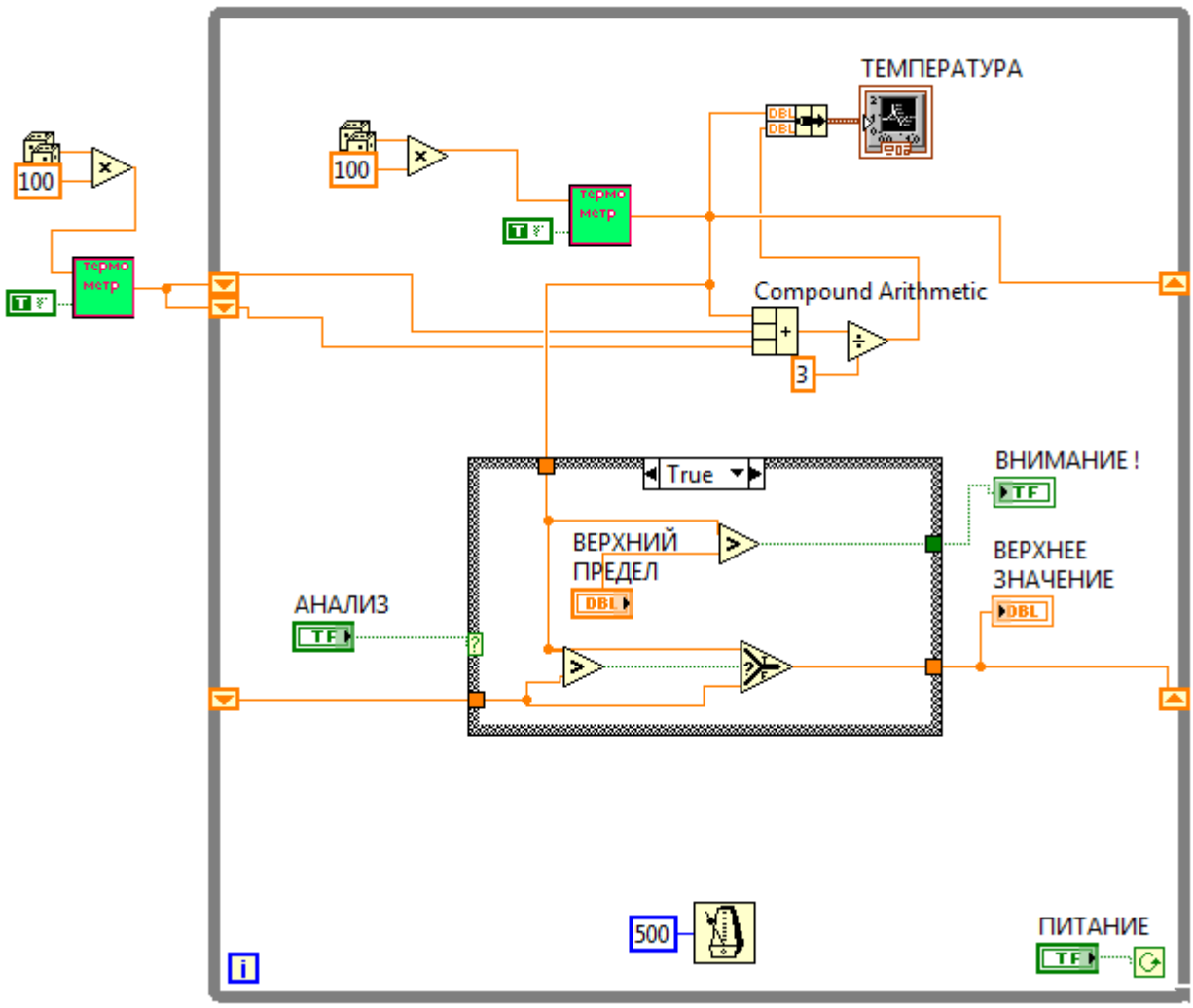


Рисунок 9.15

1) Поместить на блок-диаграмму структуру «Case». При необходимости растянуть цикл «While» так, чтобы внутрь него поместились все элементы.

2) Поместить функцию «Greater?» из меню «Functions – Programming - Comparison». Эта функция возвращает значение «True», если температура превысит величину в элементе «Верхний предел». В противном случае функция возвращает значение «False».

3) Заполнить вариант структуры «False», как показано на рисунке 9.16.

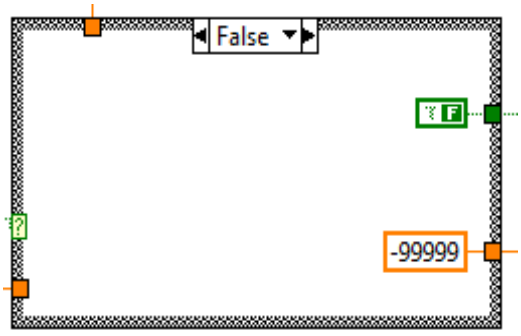


Рисунок 9.16

9.2.4.4 Перейти на фронт-панель; ввести исходные данные: «Верхний предел» = 80; включить элемент «Питание» и «Анализ».

9.2.4.5 Запустить программу на выполнение с помощью однократного «RUN».

9.2.4.6 Проанализировать и записать результат выполнения программы.

9.3 Содержание отчета

Отчет должен содержать:

- цель и задание к лабораторной работе;
- screen shot интерфейса всех заданий;
- разработанные блок-диаграммы ВП;
- описание элементов управления и индикации на блок-диаграмме;
- выводы по работе.

9.4 Контрольные вопросы

9.4.1 Три типа структуры «Case».

9.4.2 Какое количество вариантов можно создать для логического типа структуры «Case»?

9.4.3 Какое количество вариантов можно создать для строкового типа структуры «Case»?

9.4.4 Какое количество вариантов можно создать для числового типа структуры «Case»?

9.4.5 Как создать структуру «Case» по перечислениям?

9.4.6 Как формируются входные и выходные терминалы структуры «Case»?

10 Лабораторная работа №10. Работа со структурами. Узел формул

Цель работы: изучить структуру «Узел формул» и его использование как метод принятия решений в ВП.

10.1 Задание к лабораторной работе

Задание 1. Создать ВП, который использует структуру «Узел формул» для решения уравнения $Y=A^3+A$; $A=\operatorname{tg}(x)^3+\cos(x)$ и выводит результаты в виде графика.

Задание 2. Создать ВП – ограничитель сигнала на уровне ±5, используя узел Формул.

Задание 3. Создать ВП, вычисляющий мгновенные значения сигнала $U(t)=U*\operatorname{Cos}(2\pi ft+\varphi)$ двумя способами:

- используя узел Формул;
- методом графического программирования.

10.2 Порядок выполнения лабораторной работы

10.2.1 Для выполнения лабораторной работы необходимо зарегистрироваться в системе:

- загрузить файл Base_LabView, появится окно выбора лабораторных работ (рисунок 1.1), в главном меню LabView нажать кнопку «RUN»;
- выбрать лабораторную работу «Принятие решений в ВП. Узел формул»;
- появится окно регистрации студента: ввести свои Фамилию, Имя; ввести номер группы; нажать кнопку «Далее».

В открывшемся меню LabView выбрать «New - Blank VI» - новый ВП.

10.2.2 Задание 1. Создать ВП, который использует структуру «Узел формул» для решения уравнения $Y=A^3+A$; $A= tg(x)^3 +\cos(x)$ и выводит результаты в виде графика.

10.2.2.1 Построить фронт-панель, как показано на рисунке 10.1. Элемент «Waveform Chart» переименовать в «График».

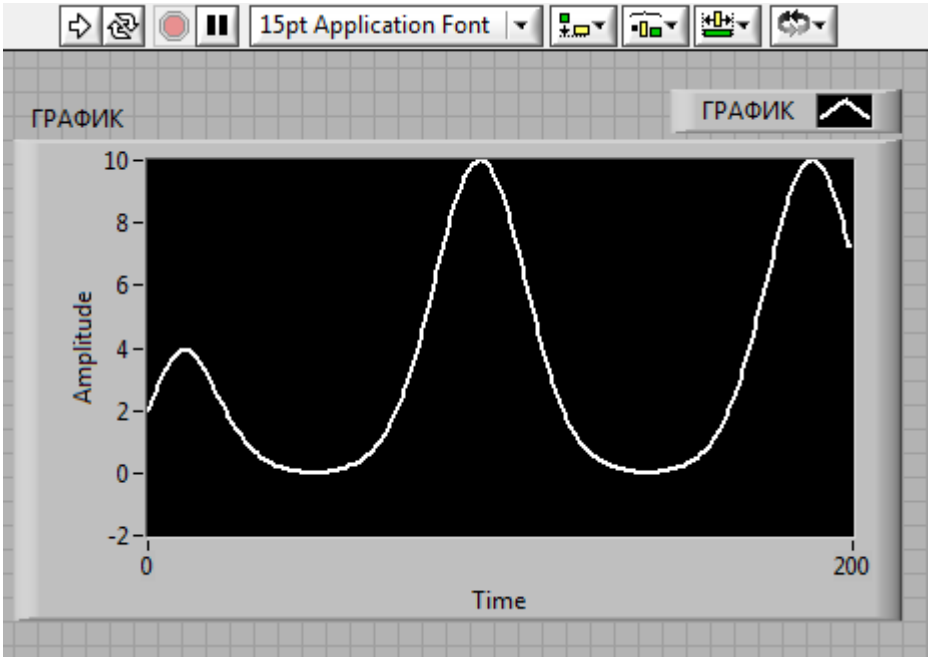


Рисунок 10.1

10.2.2.2 Построить блок-диаграмму, как показано на рисунке 10.2.

1) Поместить на блок-диаграмму элемент «Formula Node» из меню «Functions – Programming - Structure».

2) Создать на левой границе структуры «Formula Node» входной терминал X, щелкнуть правой кнопкой мыши по левой границе структуры и выбрать из контекстного меню «Add Input». В появившееся окно терминала ввести букву X.

3) Создать на правой границе структуры «Formula Node» выходные терминалы для выходной переменной У и промежуточной переменной А, выбрав из контекстного меню «Add Output».

Имя переменной во входных и выходных терминала должны точно соответствовать имени переменной, используемой в уравнении, внесенном в поле структуры «Formula Node».

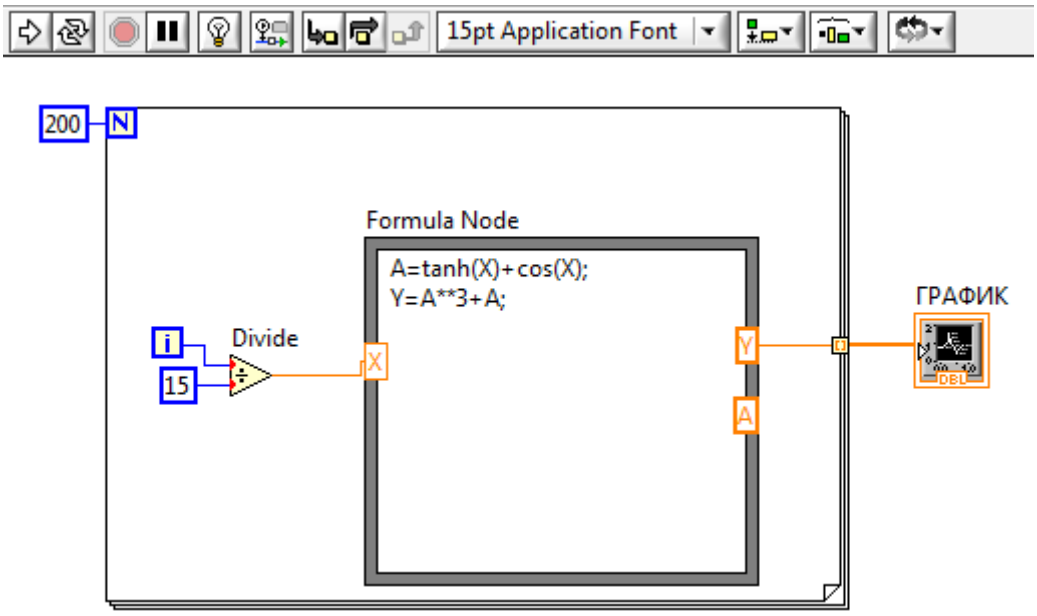


Рисунок 10.2

4) Ввести в поле «Formula Node» выражение заданного арифметического уравнения, используя синтаксис узла Формул:

A=tanh(X) + cos(X);

Y=A**3 + A;

Каждая строка программы завершается точкой с запятой.

Информацию о синтаксисе можно получить в справке «LabView Help».

10.2.2.3 Сохранить ВП под именем «Узел формулы».

10.2.2.4 Запустить программу на выполнение с помощью «RUN».

10.2.2.5 Проанализировать и записать результат выполнения программы.

10.2.3 Задание 2. Создать ВП – ограничитель сигнала на уровне ±5, используя узел Формул. Реализовать сигнал У=Х; если У превысит значение =5, то У=5; если У меньше (-5), то У=-5.

Реализовать программу двумя способами:

- реализовать условный оператор в узле Формул (рисунок 10.3);

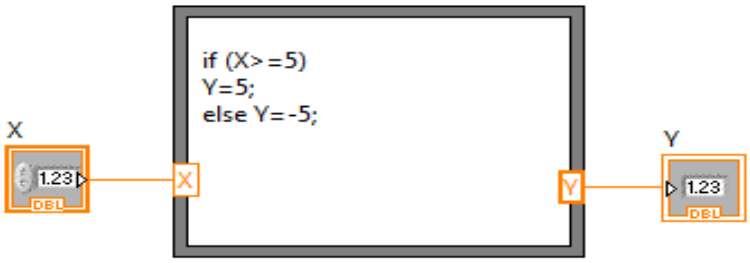


Рисунок 10.3

– реализовать условный оператор обычным способом графического программирования с использованием структуры «Select» (рисунок 10.4).

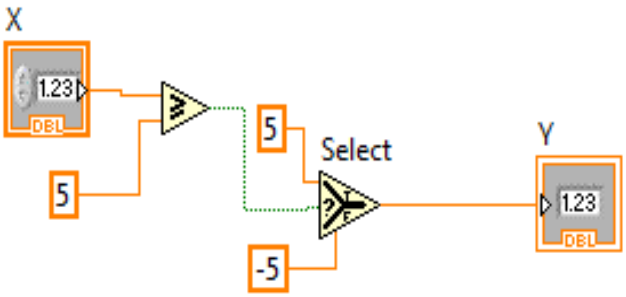


Рисунок 10.4

Задание выполнить самостоятельно.

10.2.4 Задание 3. Создать ВП, вычисляющий мгновенные значения сигнала U(t)=U*Cos(2πft+φ) двумя способами:

– используя узел Формул;

– методом графического программирования.

10.2.4.1 В открывщемся меню LabView выбрать «New - Blank VI» - новый ВП.

10.2.4.2 Построить фронт-панель ВП, как показано на рисунке 10.5 и блок-диаграмму методом графического программирования, как на рисунке 10.6.

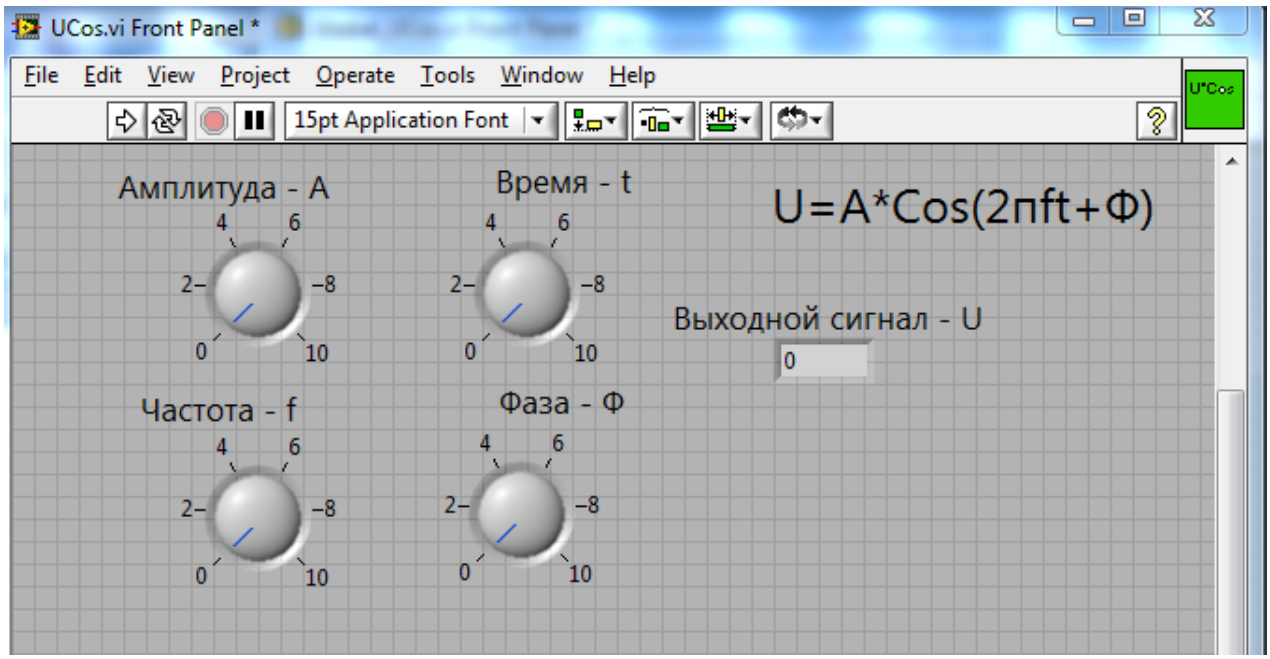


Рисунок 10.5

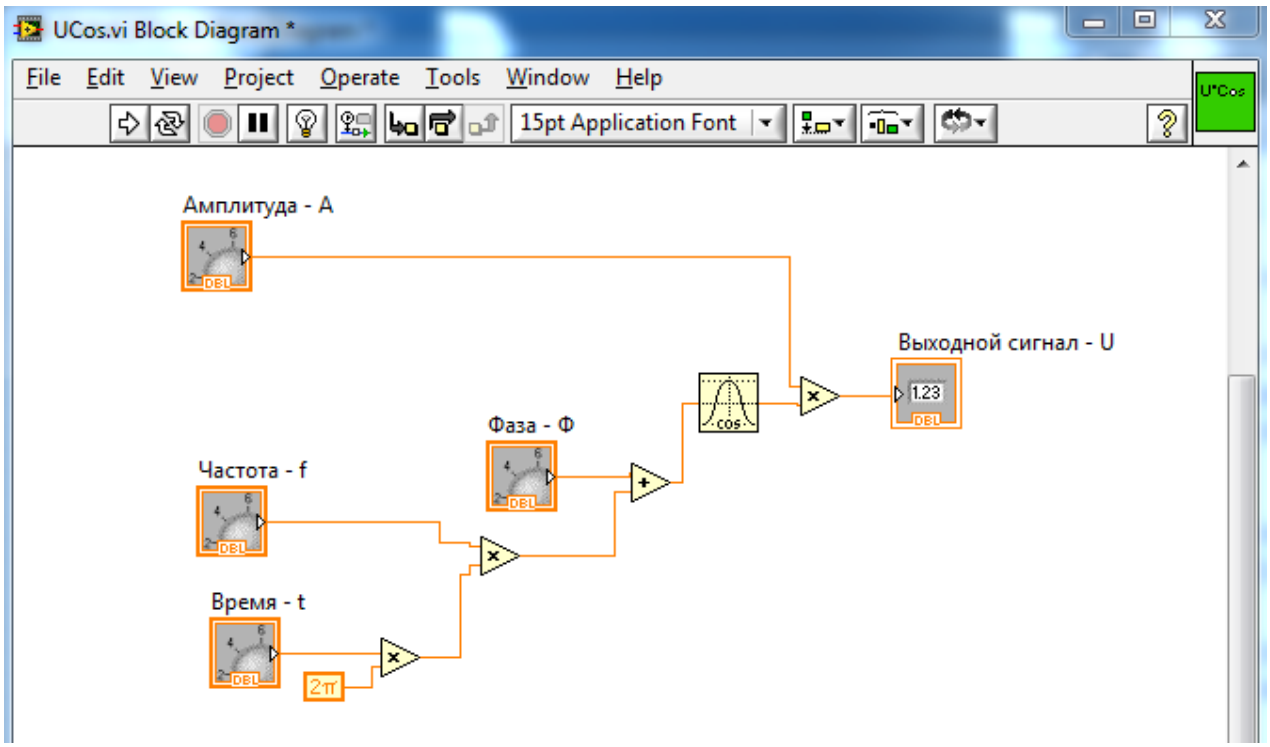


Рисунок 10.6

10.2.4.3 Сохранить ВП под именем «UCos.vi».

10.2.4.5 Создать подпрограмму ВП На базе разработанного выше ВП «UCos.vi» создать подпрограмму ВП «Сигнал напряжения» по алгоритму, описанному в пунктах (2.2.5.1-2.2.5.3) с иконкой .

10.2.4.6 Пересохранить ВП «UCos.vi» под именем «УзелФормул UCos.vi» и заменить блок-диаграмму нового ВП, как показано на рисунке 10.7 с применением узла Формул.

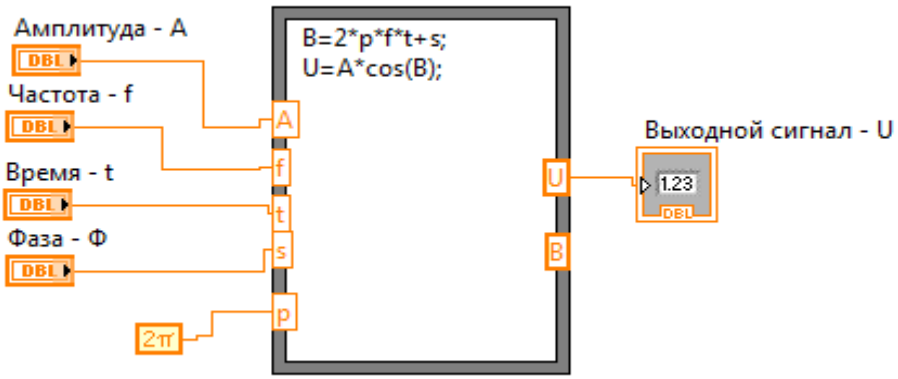


Рисунок 10.7

10.2.4.7 Сравнить результаты выполнения двух ВП: ВП «UCos.vi» и ВП «УзелФормул UCos.vi». Сделать вывод.

10.3 Содержание отчета

Отчет должен содержать:
- цель и задание к лабораторной работе;
- screen shot интерфейса всех заданий;
- разработанные блок-диаграммы ВП;
- описание элементов управления и индикации на блок-диаграмме;
- выводы по работе.

10.4 Контрольные вопросы

10.4.1 Синтаксис структуры «Formula Node».

10.4.2 Как формируются входные терминалы структуры «Formula Node»?

10.4.3 Как формируются выходные терминалы структуры «Formula Node»?

10.4.4 Как реализовать оператор проверки условий в структуре «Formula Node»?

10.4.5 Алгоритм создания подпрограммы ВП.

11 Лабораторная работа №11. Работа с последовательной структурой «Sequence Structure»

Цель работы: изучить структуру последовательного выполнения фрагментов ВП «Sequence Structure».

11.1 Задание к лабораторной работе

Задание 1. Создать ВП, вычисляющий время, занимаемое операцией совпадения заданного числа с числом, сгенерированным счетчиком случайных чисел.

Задание 2. Модифицировать ВП «UCos.vi» с использованием структуры «Sequence Structure».

11.2 Порядок выполнения лабораторной работы

11.2.1 Для выполнения лабораторной работы необходимо зарегистрироваться в системе:

- загрузить файл Base_LabView, появится окно выбора лабораторных работ (рисунок 1.1), в главном меню LabView нажать кнопку «RUN»;
- выбрать лабораторную работу «Работа с последовательной структурой «Sequence Structure»;
- появится окно регистрации студента: ввести свои Фамилию, Имя; ввести номер группы; нажать кнопку «Далее».

В открывщемся меню LabView выбрать «New - Blank VI» - новый ВП.

11.2.2 Последовательная структура «Sequence Structure» - структура типа «Многостраничный блокнот». Последовательность выполнения фрагментов программы определяется номером листа блокнота (от меньшего к большему). Процедура заканчивается, когда выполнится фрагмент программы, находящийся на последнем листе.

11.2.3 Задание 1. Создать ВП, вычисляющий время, занимаемое операцией совпадения заданного числа с числом, сгенерированным счетчиком случайных чисел.

На фронт-панели создать лицевую панель, показанную на рисунке 11.1.

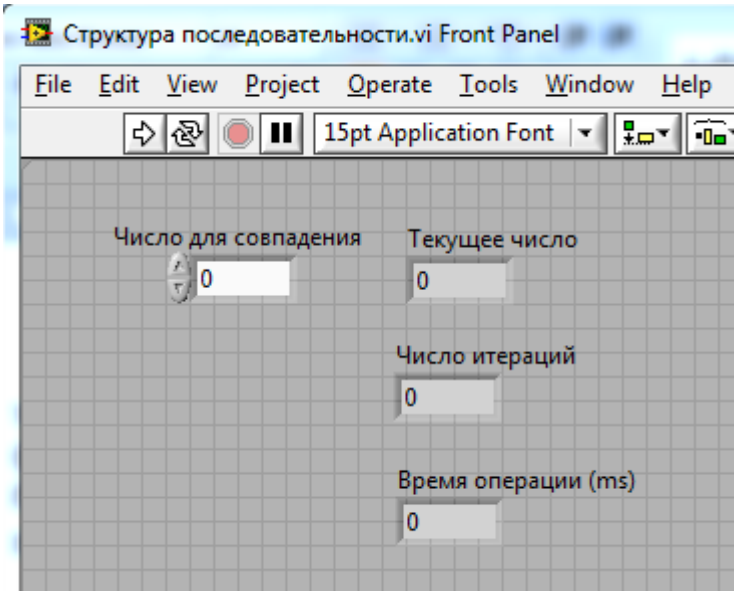


Рисунок 11.1

11.2.3.1 Перейти на блок-диаграмму. Собрать код программы, изображенный на рисунке 11.2, используя цикл «While».

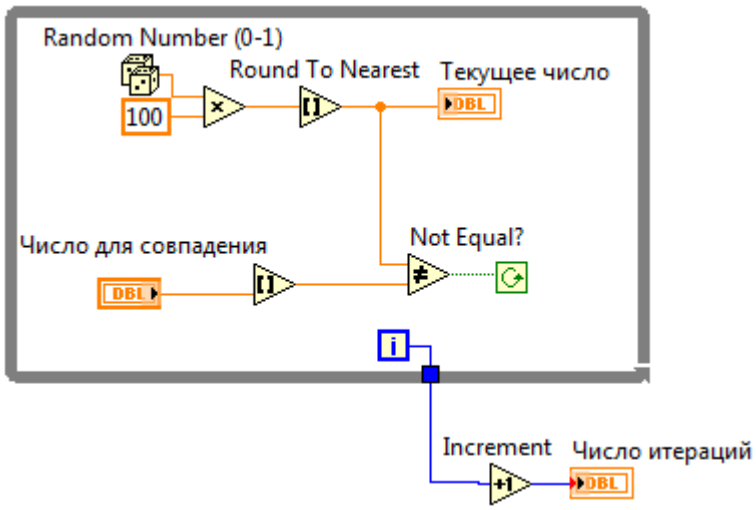


Рисунок 11.2

11.2.3.2 Поместить код программы (рисунок 11.2) в кадр структуры последовательности (рисунок 11.3). Для создания кадра поместить на блок-диаграмму структуру «Stacked Sequence Structure» из палитры «Function – Programming – Structures». Данная структура организовывается аналогично структурам цикла.

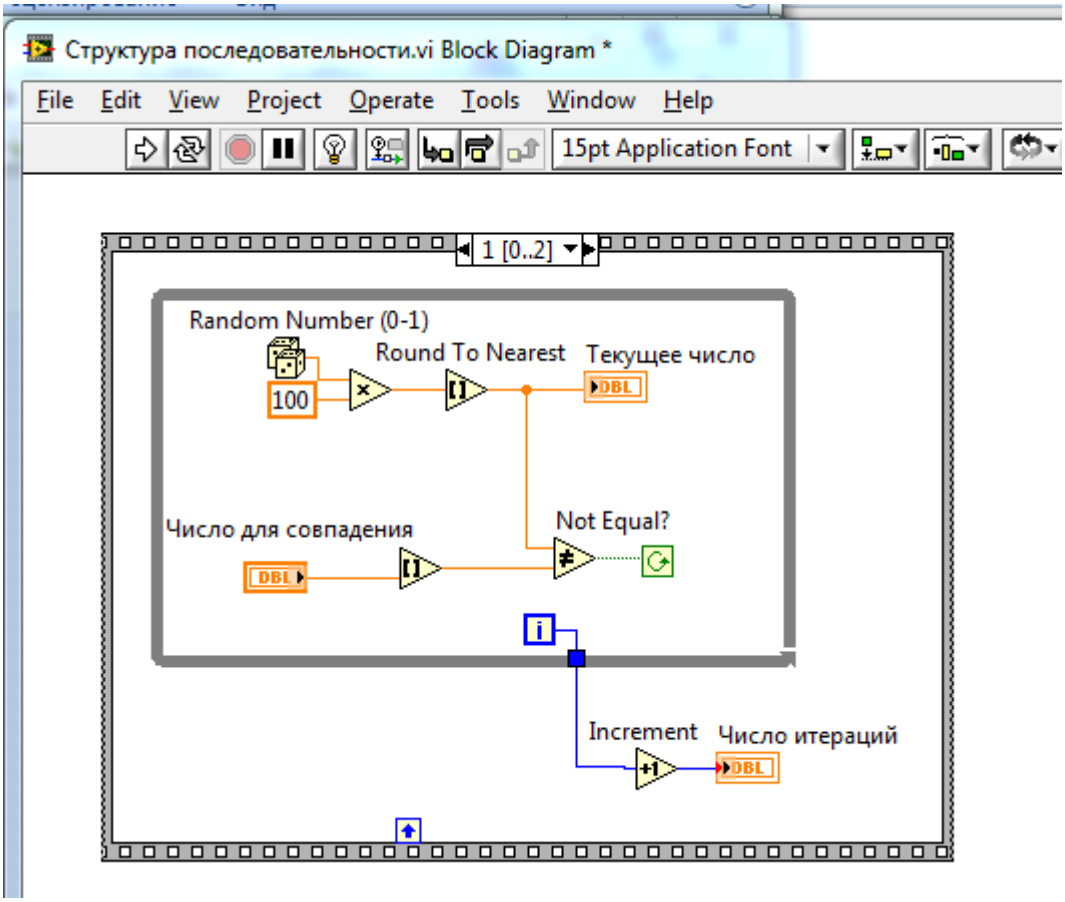


Рисунок 11.3

11.2.3.3 Создать два новых кадра последовательной структуры: один кадр «до» существующего кадра, выбрав пункт «Add Frame Before» из контекстного меню щелчком правой кнопки мыши по границе кадра; и второй кадр – «после», выбрав из контекстного меню пункт «Add Frame After». Контекстное меню содержит опции удаления кадра «Delete This Frame», дублирования кадра «Duplicate Frame» и другие.

В верхнем горизонтальном терминале отображается номер кадра (страницы блокнота) и информация о количестве кадров (в скобках).

11.2.3.4 Записать код программы на нулевом кадре, как показано на рисунке 11.4. Функция «Tick Count (ms)» находится в палитре «Timing», она возвращает показания внутренних часов в миллисекундах.

Создать терминал локальной переменной, щелкнув правой кнопкой мыши по нижней границе кадра и выбрав из контекстного меню пункт «Add Sequence Local». Терминал локальной переменной отобразится в виде пустого квадрата на границе кадра. При подключении его к функции «Tick Count»

внутри квадрата появится стрелка с указанием направления. Такой же терминал локальной переменной появится на кадрах №1 и №2, но со стрелкой в противоположном направлении.

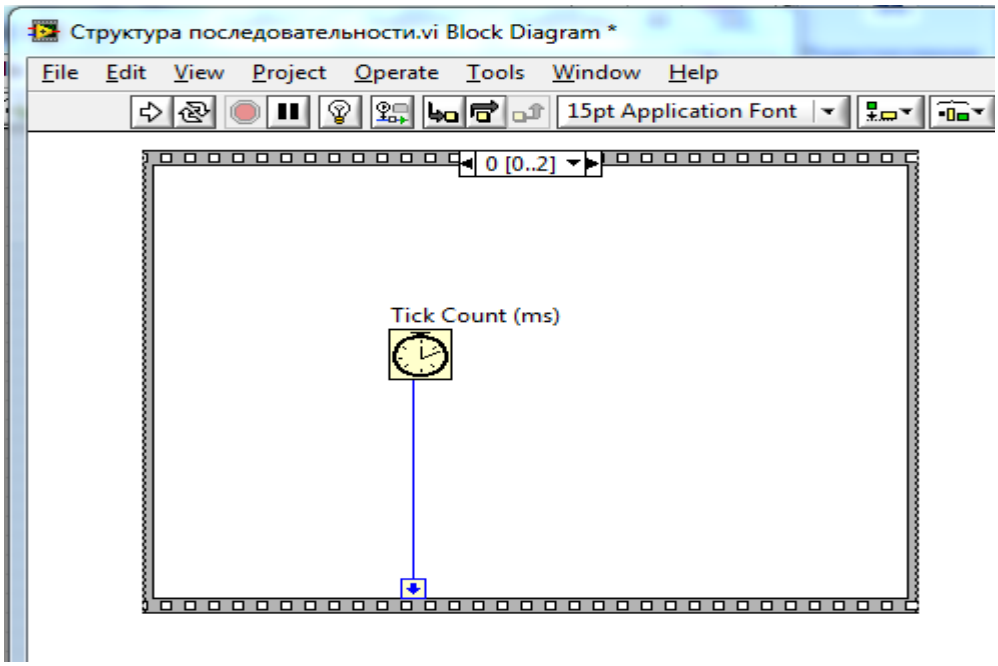


Рисунок 11.4

11.2.3.5 Перейти на кадр №2 и собрать на нем код программы, изображенный на рисунке 11.5.

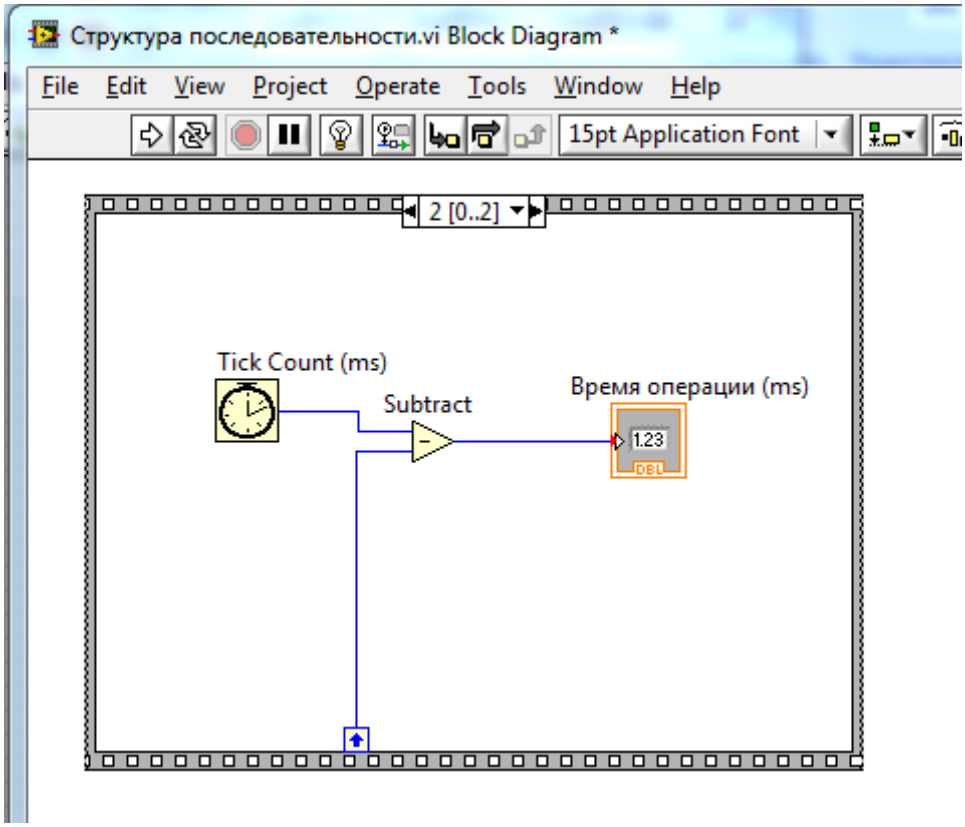


Рисунок 11.5

11.2.3.6 Запустить программу в циклическом режиме «Run Continuously».

При выполнении кадра №0 функция «Tick Count (ms)» возвращает показания внутренних часов в миллисекундах. Это значение передается на терминал локальной переменной, поэтому оно доступно в последующих кадрах. В кадре №1 ВП выполняет цикл по условию до тех пор, пока введенное число не совпадет с числом, выдаваемым датчиком случайных чисел. В кадре №2 функция «Tick Count (ms)» возвращает новые показания внутренних часов в миллисекундах. ВП вычитает старый отсчет времени, полученный в кадре №0, из нового для вычисления времени, за которое найдено равенство чисел в кадре №1.

Результат отображается в окне «Время операции» в миллисекундах.

11.2.3.7 Сохранить ВП под именем «Подсчет времени операции. vi».

11.2.4 Задание 2. Модифицировать ВП «UCos.vi» с использованием структуры «Sequence Structure».

11.2.4.1 В открывшемся меню LabView выбрать «Open» – «Ucos.vi». пересохранить данный ВП, разработанный в п.10.2.4, под именем «Ucos_ Sequence.vi».

11.2.4.2 Поместить код программы в кадр структуры последовательности. Для создания кадра поместить на блок-диаграмму структуру «Stacked Sequence Structure» из палитры «Function – Programming – Structures». Добавить на фронт-панель числовой элемент управления – цифровой регулятор с именем «Коэффициент усиления» (рисунок 11.6), который увеличивает напряжение на выходе в соответствии с установленным на регуляторе значением.

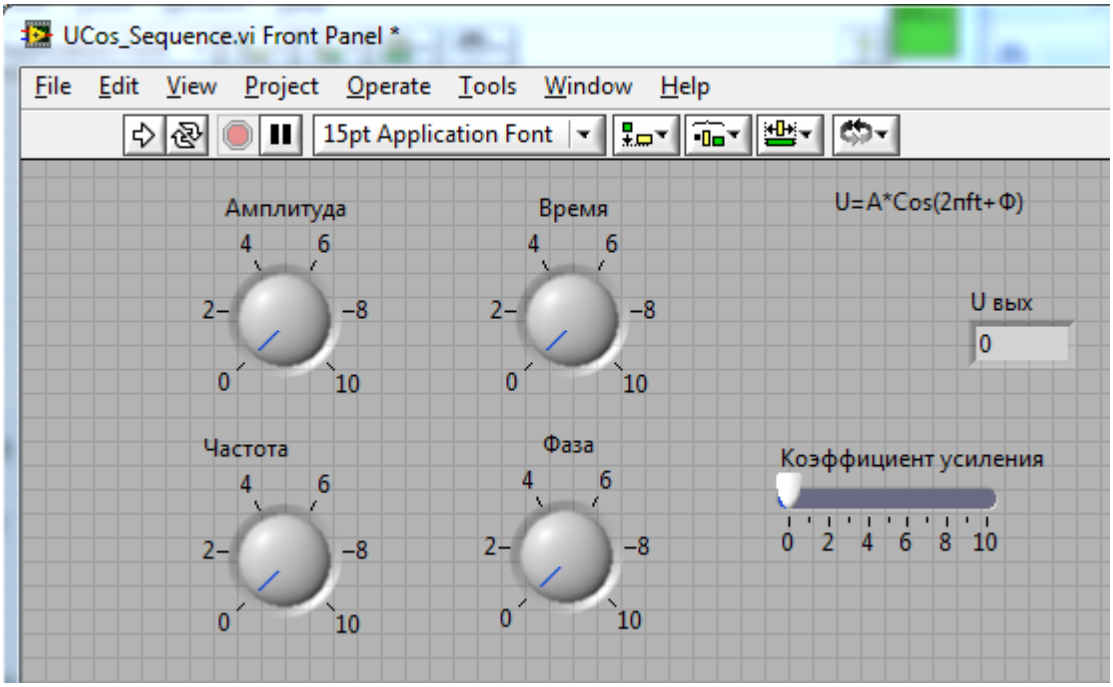


Рисунок 11.6

11.2.4.3 Создать нулевой кадр, используя опцию «Add Frame Before» из контекстного меню структуры. Перетащить элемент «Коэффициент усиления» из кадра №1 на нулевой кадр: передвинуть элемент за пределы кадра №1, затем, переключив номер страницы на нулевой кадр, установить его в кадр №0. Установить на нижней границе кадра терминал локальной переменной, который необходимо соединить с регулятором «Коэффициент усиления» (рисунок 11.7).

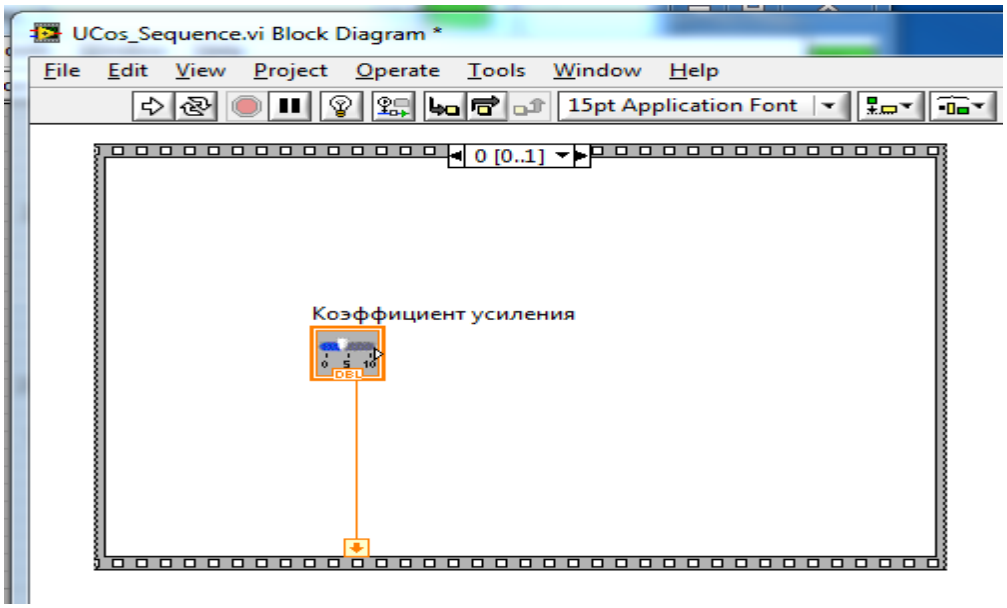


Рисунок 11.7

11.2.4.4 Переключить кадр на №1 и соединить терминал перехода с операцией умножения, вторым сомножителем будет амплитуда сигнала U (рисунок 11.8).

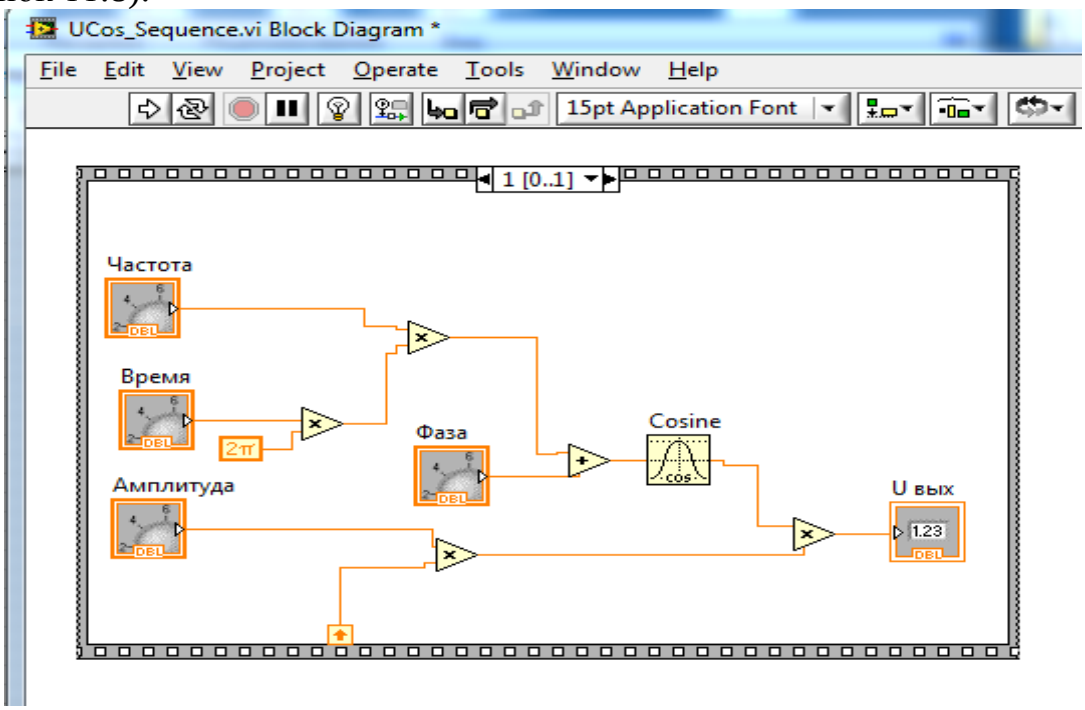


Рисунок 11.8

11.2.4.5 Запустить программу в циклическом режиме «Run Continuously». Наблюдать изменение выходного сигнала при различных значениях коэффициента усиления.

11.2.4.6 Проанализировать и сканировать результат выполнения программы.

11.3 Содержание отчета

Отчет должен содержать:

- цель и задание к лабораторной работе;
- screen shot интерфейса всех заданий;
- разработанные блок-диаграммы ВП;
- описание элементов управления и индикации на блок-диаграмме;
- выводы по работе.

11.4 Контрольные вопросы

11.4.1 Виды структур в LabView.

11.4.2 Организация структуры последовательного выполнения программы.

11.4.3 Назначение функции «Sequence Structure».

11.4.4 Назначение функции «Tick Count (ms)».

11.4.5 Контекстное меню функции «Sequence Structure».

11.4.6 Назначение и создание терминала локальной переменной.

12 Лабораторная работа №12. Локальные и глобальные переменные

Цель работы: изучить программирование и применение локальных и глобальных переменных.

12.1 Задание к лабораторной работе

Задание 1. Использовать локальные переменные для модификации ВП «UCos.vi».

Задание 2. Использовать глобальные переменные для модификации ВП «UCos.vi».

12.2 Порядок выполнения лабораторной работы

12.2.1 Для выполнения лабораторной работы необходимо зарегистрироваться в системе:

- загрузить файл Base_LabView, появится окно выбора лабораторных работ (рисунок 1.1), в главном меню LabView нажать кнопку «RUN»;
- выбрать лабораторную работу «Локальные и глобальные переменные»;

– появится окно регистрации студента: ввести свои Фамилию, Имя; ввести номер группы; нажать кнопку «Далее».

12.2.2 Задание 1. Использовать локальные переменные для модификации ВП «UCos.vi». Цель модификации: показать, что локальные переменные не изменяют суть выполняемой программы.

Понятия «локальный» и «глобальный» соответствуют своим названиям: локальный – это параметр местного значения, принадлежащий одному модулю; а глобальный – межмодульный параметр, принадлежащий основной программе и каждой подпрограмме, в которой этот параметр используется.

Локальные и глобальные переменные, так же, как любой элемент LabView, могут быть переменными управления или индикации. Для этого в контекстном меню выбрать пункт «Change To Write» или «Change To Read».

12.2.2.1 В открывшемся меню LabView выбрать «Open - UCos.vi» - открыть ВП, созданный ранее, для модификации. Пересохранить файл под именем «Local_UCos.vi».

12.2.2.2 На блок-диаграмме заменить все цифровые приборы, за исключением «Амплитуда», их локальными параметрами.

Для этого в контекстном меню каждого цифрового элемента выбрать «Create – Local Variable» (рисунок 12.1).

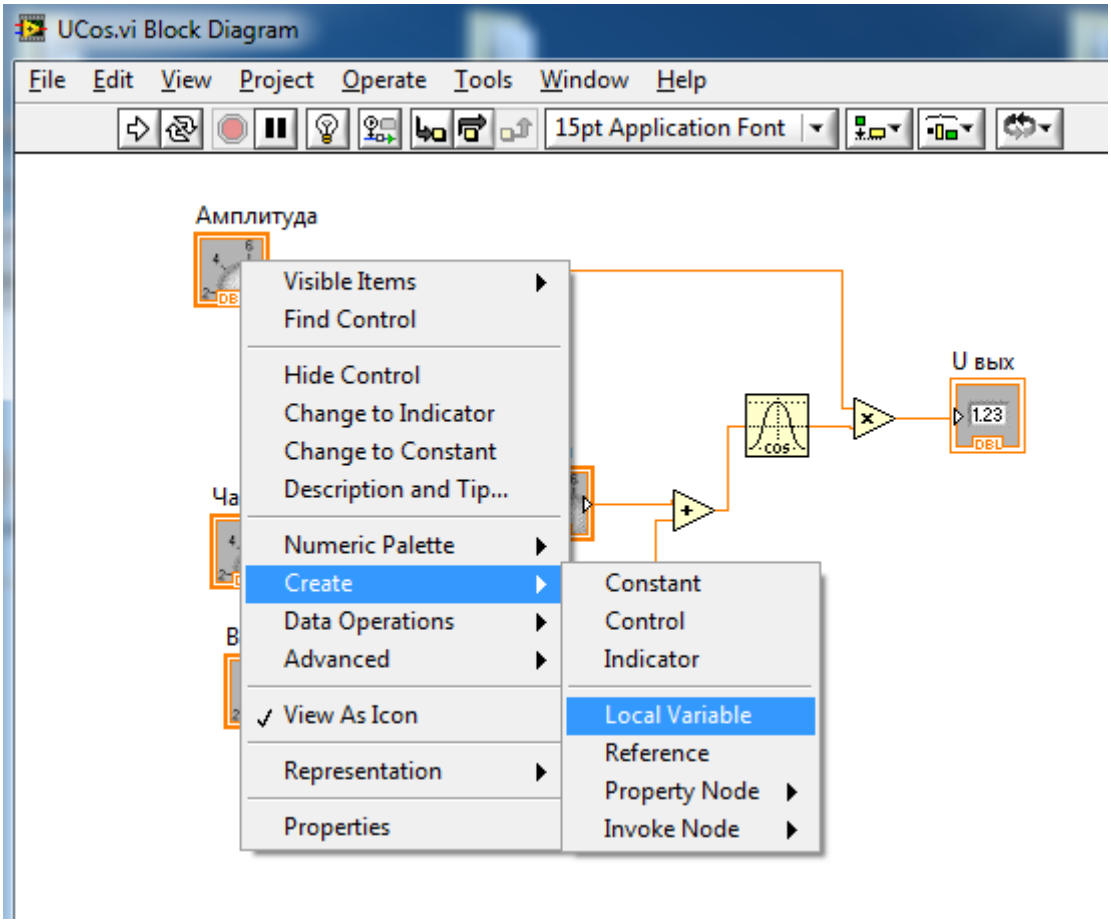


Рисунок 12.1

12.2.2.3 Собрать код программы на блок-диаграмме, как на рисунке 12.2. Заменить цифровые элементы «Частота, Фаза, Время» их локальными переменными, отсоединив их от схемы, не удаляя с фронт-панели и блок-диаграммы.

Таким образом, локальные переменные должны работать, как терминалы цифровых элементов.

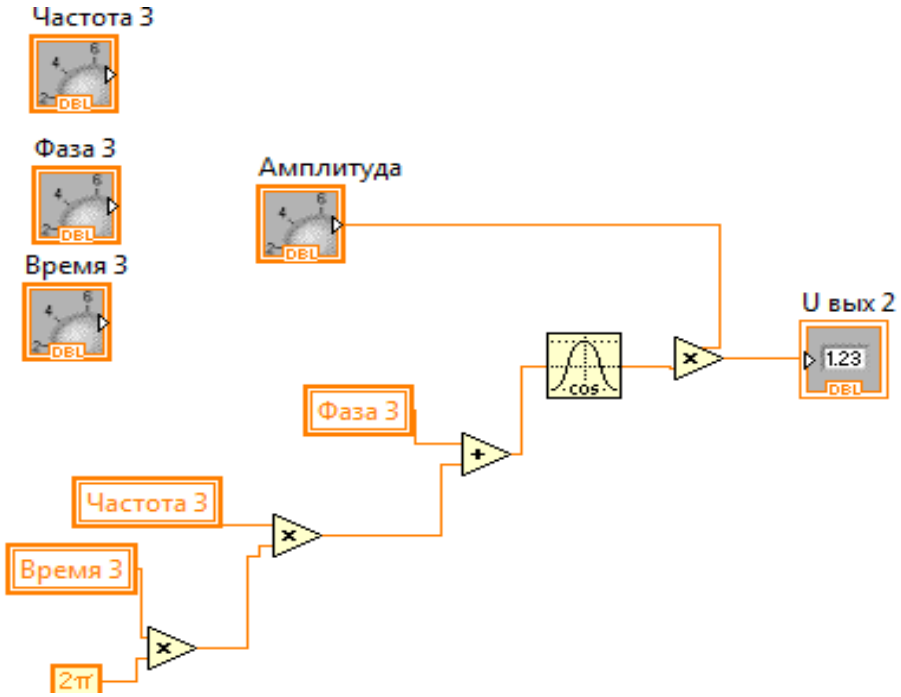


Рисунок 12.2

12.2.2.4 Запустить программу, убедиться, что программа работает так же, как она работала до замены.

Копирование локальной переменной приведет к одновременной установке копии прибора на фронт-панели, что создает неудобство для работы, если требуется тиражировать несколько локальных переменных одного и того же элемента.

Выход из такой ситуации: копировать страницы в структурах, удобнее в структуре Sequence. Локальная переменная устанавливается на страницу структуры, страница дублируется столько раз, сколько нужно копий локальной переменной. Затем копии переносятся в другое место программы, а созданная структура удаляется.

12.2.3 Задание 2. Использовать глобальные переменные для модификации ВП «UCos.vi». Цель модификации: показать, что глобальные переменные не изменяют суть выполняемой программы.

Глобальная переменная – особый вид переменных в LabView. Для них создается специальная дополнительная панель, вызвав ее пиктограмму на блок-диаграмме из меню «Function – Structure – Global Variable» - обезличенный терминал со знаком вопроса.

Установить на пиктограмму курсор и из контекстного меню вызвать «Open Front Panel», на которой собираем все необходимые элементы программы, как обычно делается на фронт-панели. Панель глобальных параметров имеет особый статус: у нее нет собственной блок-диаграммы, ей не нужен коннектор, но есть собственная пиктограмма.

12.2.3.1 В открывшемся меню LabView выбрать «Open – Local Ucos.vi» - открыть ВП, созданный ранее, для модификации. Пересохранить файл под именем «Global_UCos.vi».

12.2.3.2 Создать на блок-диаграмме структуру Sequence, разместив весь код программы в ее кадр (п.11.2.3.2).

12.2.3.3 Создать кадр *ДО* текущего, выбрав из контекстного меню на границе текущего кадра пункт «Add Frame Before». Переместить в новый кадр элемент «Амплитуда», предварительно отсоединив его от функции «Умножить».

Установить пиктограмму глобальной переменной из меню «Function – Structure – Global Variable». Дважды щелкнув на пиктограмму, откроется фронт-панель глобальной переменной под именем Global 1.vi. Создать на ней элемент «Амплитуда» (рисунок 12.3).

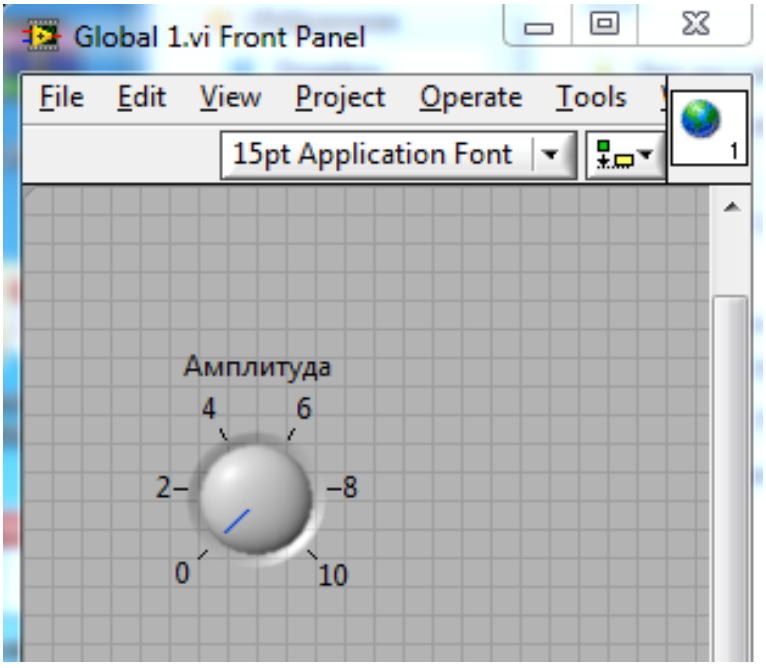


Рисунок 12.4

Закрыть Global 1.vi. дать имя глобальной переменной «Амплитуда», для этого щелкнуть на пиктограмму глобальной переменной и из контекстного меню выбрать Select Item – Амплитуда (рисунок 12.5). Соединить цифровой элемент Амплитуда с глобальной переменной «Амплитуда».

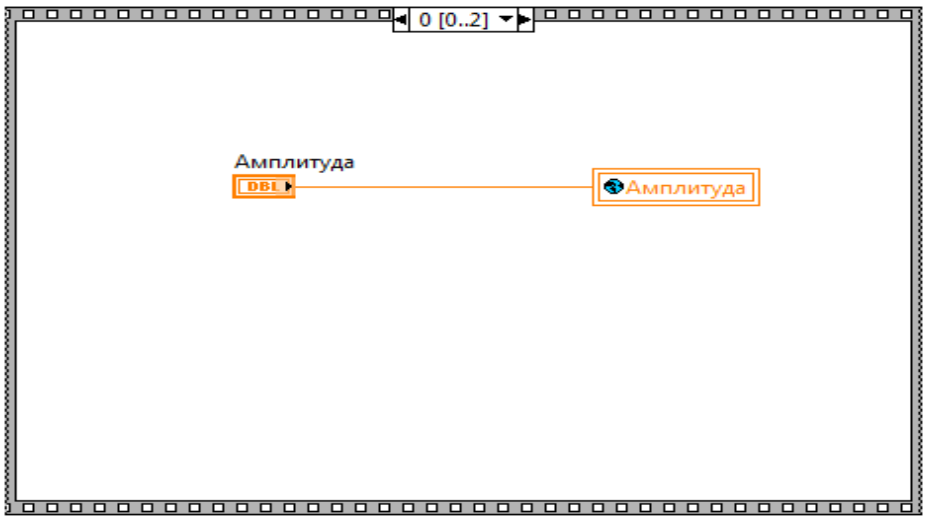


Рисунок 12.5

12.2.3.4 Добавить на фронт-панель цифровой элемент «Коэффициент усиления», как на рисунке 12.6.

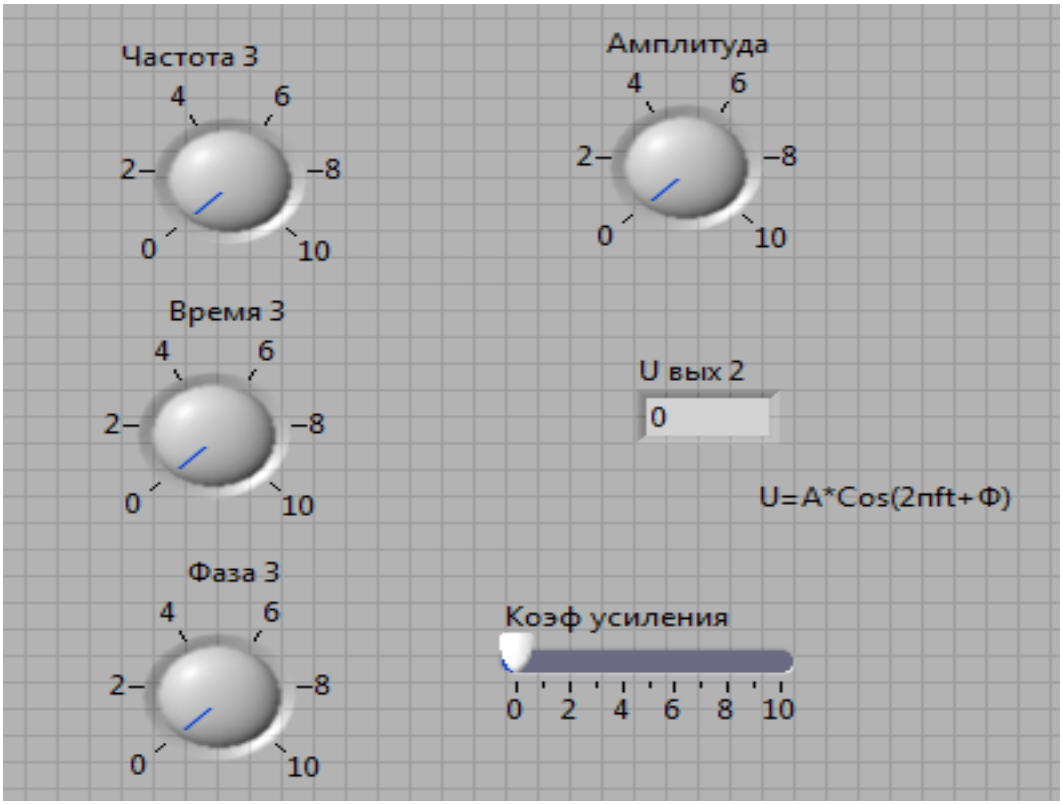


Рисунок 12.6

12.2.3.5 Для усложнения задачи создать подпрограмму «Коэффициент усиления». Для этого открыть бланк «New - Blank VI» - новый ВП. Собрать на нем программу, фронт-панель и блок-диаграмма которой показаны на рисунке 12.7. Использовать глобальную переменную «Амплитуда», созданную в п.12.2.3.3.

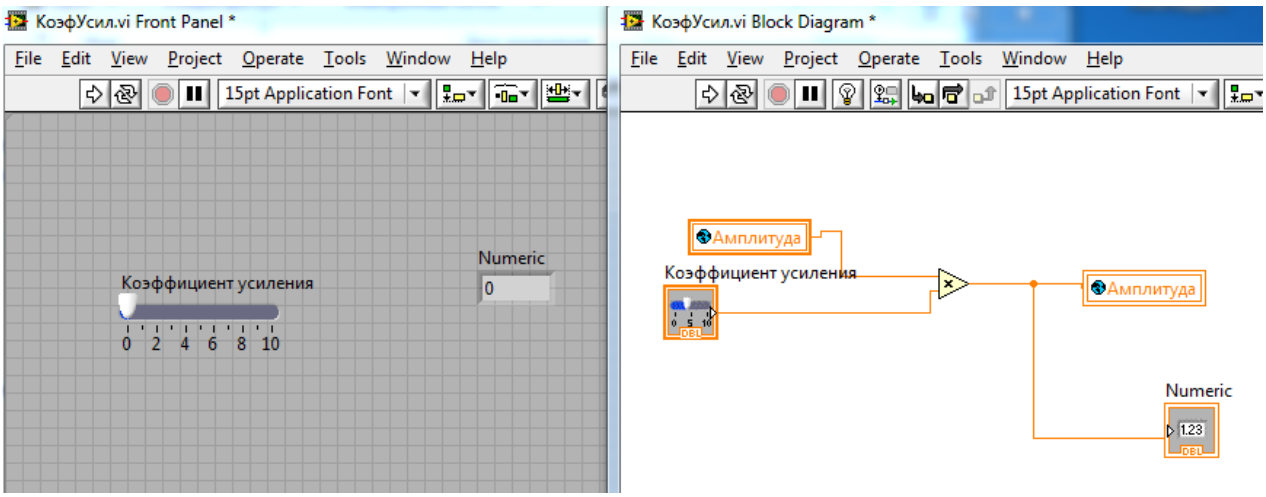


Рисунок 12.7

Создать подпрограмму «Коэффициент усиления» с именем

с одним входом и одним выходом.

12.2.3.6 Создать новый кадр ПОСЛЕ кадра «Амплитуда», выбрав из контекстного меню на границе текущего кадра пункт «Add Frame After». Переместить в новый кадр элемент «Коэффициент усиления» из основного кадра. Создать блок-диаграмму, как на рисунке 12.8.

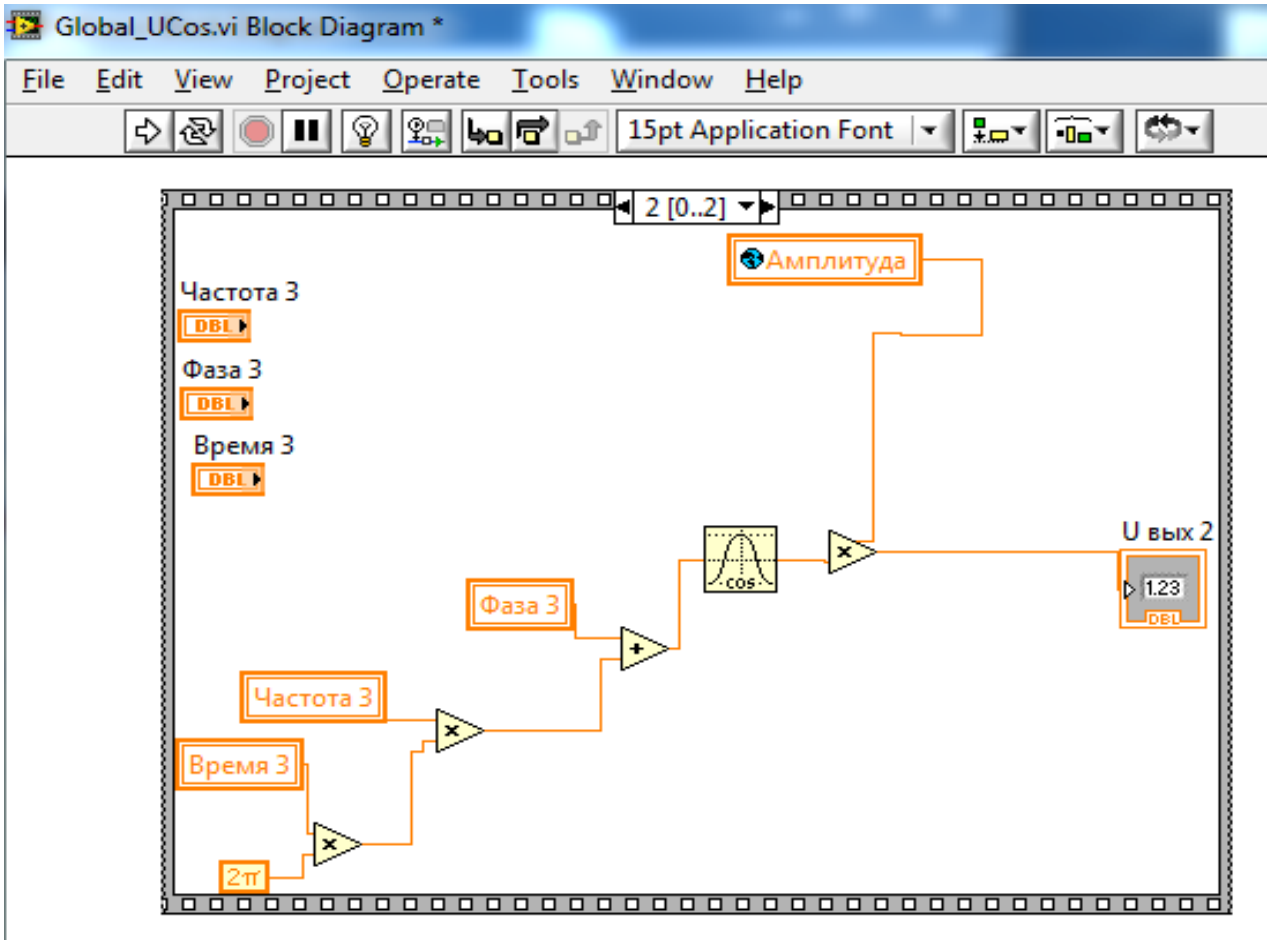


Рисунок 12.8

12.2.3.7 Сохранить файл и запустить на выполнение. Описать порядок выполнения программы с использованием локальных и глобальных переменных, а также подпрограммы «Коэффициент усиления».

12.3 Содержание отчета

Отчет должен содержать:
- цель и задание к лабораторной работе;
- screen shot интерфейса всех заданий;
- разработанные блок-диаграммы ВП;
- описание элементов управления и индикации на блок-диаграмме;
- выводы по работе.

12.4 Контрольные вопросы

12.4.1 Для каких целей используются локальные и глобальные переменные?

12.4.2 Как создать локальную переменную?

12.4.3 Как создать глобальную переменную?

12.4.4 В чем отличие локальной и глобальной переменных?

12.4.5 Как перенести один элемент программы из одного кадра структуры последовательности в другой кадр?

12.4.6 Пояснить алгоритм выполнения программы «Global_UCos.vi».

13 Лабораторная работа №13. Функции обработки строк

Цель работы: изучить создание строковых элементов управления и отображения данных и использование некоторых функций их обработки.

13.1Задание к лабораторной работе

Задание 1. Создать ВП, который преобразует числовые данные в строку и объединяет строку с другими строками в одну.

Задание 2. Создать ВП, который переводит часть строки в числовой формат.

13.2 Порядок выполнения лабораторной работы

13.2.1 Для выполнения лабораторной работы необходимо зарегистрироваться в системе:

- загрузить файл Base_LabView, появится окно выбора лабораторных работ (рисунок 1.1), в главном меню LabView нажать кнопку «RUN»;
- выбрать лабораторную работу «Функции обработки строк»;
- появится окно регистрации студента: ввести свои Фамилию, Имя; ввести номер группы; нажать кнопку «Далее».

В открывшемся меню LabView выбрать «New - Blank VI» - новый ВП.

13.2.2 Задание 1. Создать ВП, который преобразует числовые данные в строку и объединяет строку с другими строками в одну.

13.2.2.1 На фронт-панели собрать интерфейс задачи, как показано на рисунке 13.1.

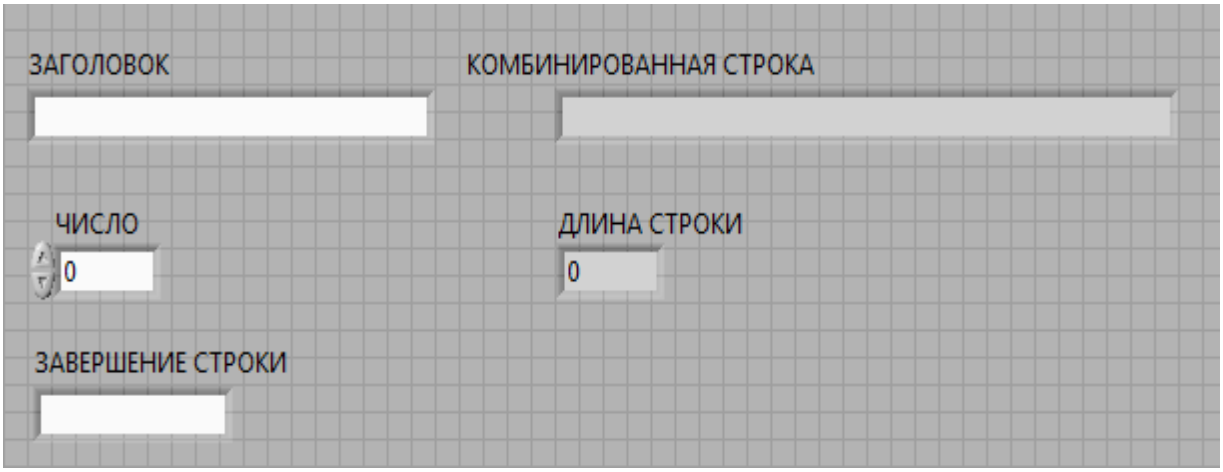


Рисунок 13.1

1) Поместить на фронт-панель два строковых элемента управления «String Ctrl» из меню «Controls – Express – Text Ctrls» под именем «Заголовок» и «Завершение строки»; строковый элемент отображения «String Ind» из меню «Controls – Express – Text Inds» под именем «Комбинированная строка».

2) Поместить числовой элемент управления под именем «Число» и числовой элемент индикации под именем «Длина строки». Изменить тип данных «Длина строки» на целочисленный.

13.2.2.2 Построить блок-диаграмму, как показано на рисунке 13.2.

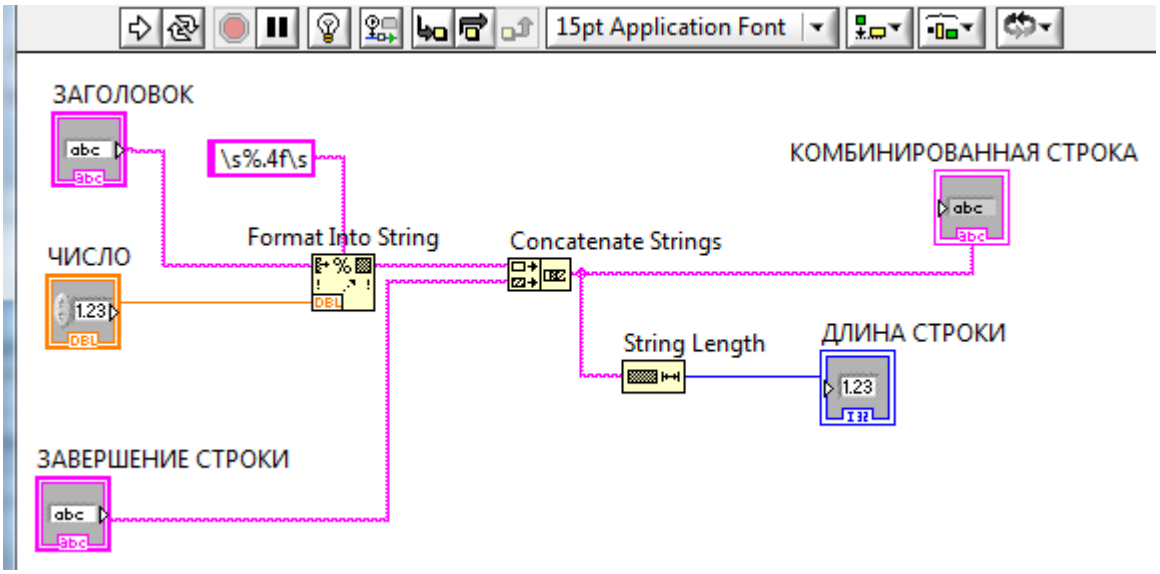


Рисунок 13.2

Функции работы со строками находятся в меню «Functions – Programming - String».

1) Выбрать функцию «Format Into String», которая преобразует параметры любого формата, такие как числовые данные, в строку:

а) щелкнуть правой кнопкой мыши по функции и выбрать из контекстного меню «Edit Format String» для вызова окна редактора;

б) в поле ввода текста «Use specified precision» ввести значение «4», это означает, что элемент «Число» преобразуется в строку с четырьмя знаками после запятой. Нажмите «Ok», LabView создаст формат строки «%.4f»;

в) ввести пробелы с обеих сторон «%.4f» с помощью инструмента «Ввод текста» и нажать клавиши «Shift+Enter». Тем самым на элементе «Комбинированная строка» числовые данные появятся в виде слова (строки) с пробелами с обеих сторон;

г) щелкнуть правой кнопкой мыши по константе «%.4f» и выбрать в контекстном меню пункт «’\’ Codes Display», введенные в константу пробелы заменятся на «\».

2) Выбрать функцию «Concatenate String», которая объединяет входящие в нее строки в одну.

3) Выбрать функцию «String Length», которая выдает значение количества символов в объединенной строке «Комбинированная строка».

13.2.2.3 Сохранить ВП под именем «Комбинированная строка».

13.2.2.4 Перейти на фронт-панель, ввести исходные данные. Например, «Заголовок» = «Результат измерения равен», «Число» = «5», «Завершение строки» = «Вольт».

13.2.2.5 Запустить ВП на выполнение с помощью однократного «RUN».

13.2.2.6 Проанализировать и записать результат выполнения программы.

13.2.2.7 На фронт-панели создать подпрограмму «Комбинированная строка» (иконку и соединительную панель) (п.1.2.8) для использования созданного ВП в качестве подпрограммы в других ВП.

13.2.3 Задание 2. Создать ВП, который переводит часть строки в числовой формат.

13.2.3.1 В главном меню выбрать «New - Blank VI» - новый ВП. На фронт-панели собрать интерфейс задачи, как показано на рисунке 13.3.

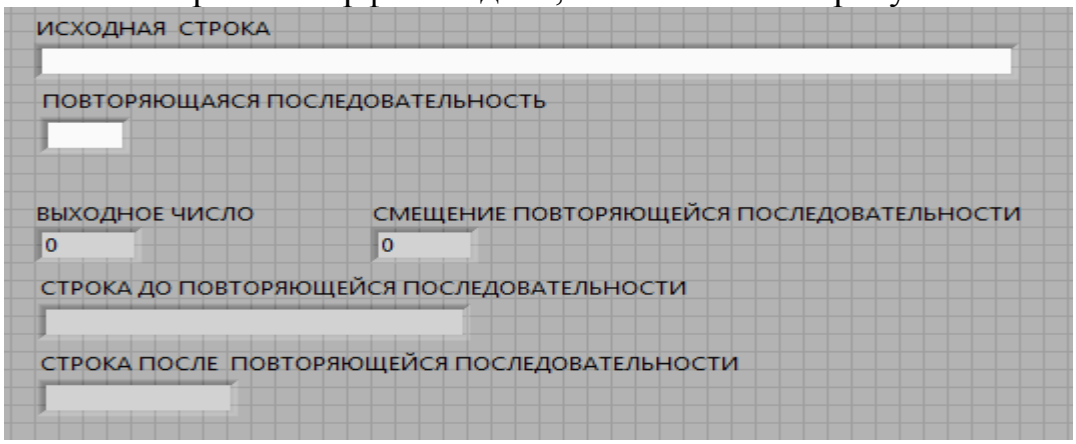


Рисунок 13.3

1) Поместить на фронт-панель два строковых элемента управления «String Ctrl» из меню «Controls – Express – Text Ctrls» под именем «Исходная строка» и «Повторяющаяся последовательность»; строковые элементы отображения «String Ind» из меню «Controls – Express – Text Inds» под именем «Строка до повторяющейся последовательности» и «Строка после повторяющейся последовательности».

2) Поместить числовой элемент отображения под именем «Выходное число» и «Смещение повторяющейся последовательности». Изменить тип данных «Смещение повторяющейся последовательности» на целочисленный.

13.2.3.2 Построить блок-диаграмму, как показано на рисунке 13.4.

Функции работы со строками находятся в меню «Functions – Programming - String».

1) Выбрать функцию «Match Pattern», которая осуществляет поиск «Повторяющейся последовательности» в элементе «Исходная строка»:

а) соединить элемент «Повторяющаяся последовательность» со входом «Regular expression», элемент «Исходная строка» со входом «String»;

б) соединить выходы функции «Before Substring» с «Строка до повторяющейся последовательности», «After Substring» с «Строка после повторяющейся последовательности», «Offset past match» с «Смещение повторяющейся последовательности».

2) Выбрать функцию «Scan from String», которая преобразует строку в числовые данные.

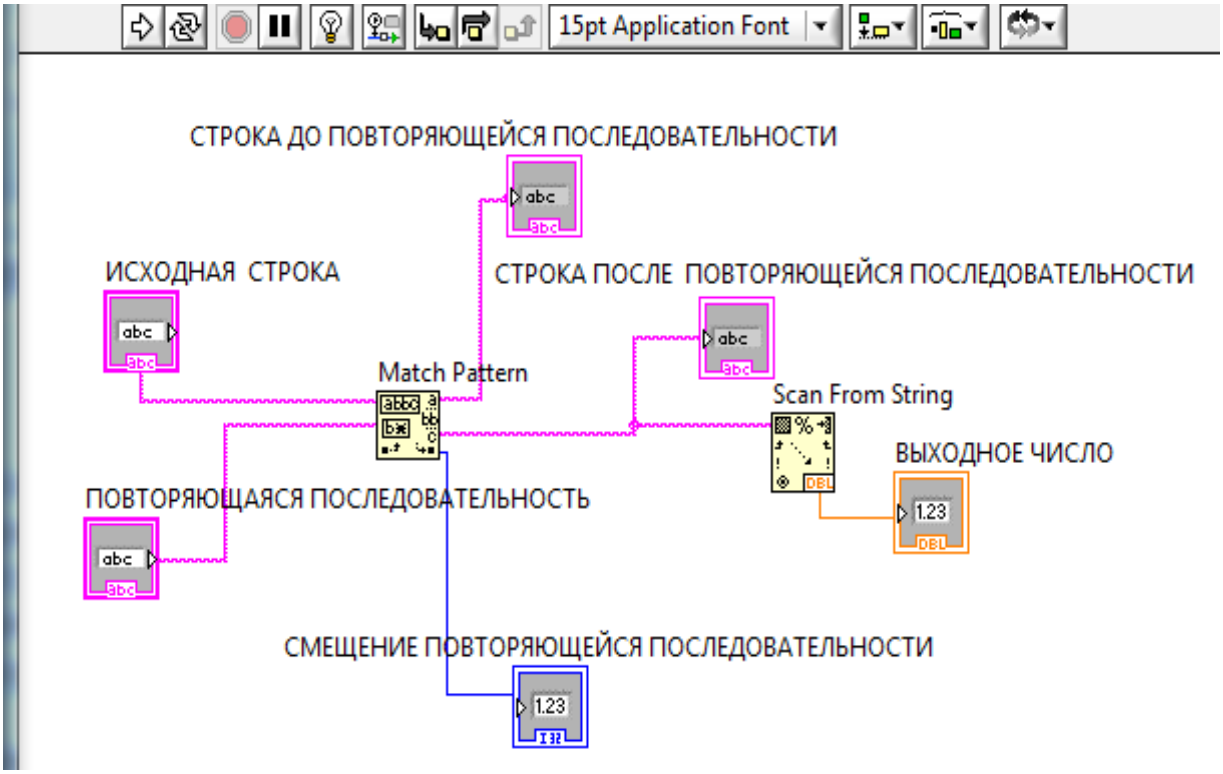


Рисунок 13.4

13.2.2.3 Сохранить ВП под именем «Преобразование строки в число».

13.2.2.4 Перейти на фронт-панель, ввести исходные данные. Например, «Исходная строка» - «Результат измерения в Вольтах = 5 », «Повторяющаяся последовательность» - «=».

13.2.2.5 Запустить ВП на выполнение с помощью однократного «RUN».

13.2.2.6 Проанализировать и сканировать результат выполнения программы.

13.3 Содержание отчета

Отчет должен содержать:

- цель и задание к лабораторной работе;
- screen shot интерфейса всех заданий;
- разработанные блок-диаграммы ВП;
- описание элементов управления и индикации на блок-диаграмме;
- выводы по работе.

13.4 Контрольные вопросы

13.4.1 Как создаются строковые элементы управления и отображения?

13.4.2 Назначение функции «Concatenate String».

13.4.3 Назначение функции «String Lenth».

13.4.4 Назначение функции «Match Pattern».

13.4.5 Как преобразовать числовые данные в строку?

13.4.6 Как преобразовать строку в числовые данные?

13.4.7 Как задать точность представления числовых данных в диалоговом окне редактора?

14 Лабораторная работа №14. Файловый ввод/ вывод

Цель работы: изучить основы файлового ввода/вывода и использование некоторых функций их обработки.

14.1 Задание к лабораторной работе

Задание 1. Создать ВП, который записывает данные в файл и отображает путь к созданному файлу.

Задание 2. Создать ВП, который читает файл, созданный в предыдущем задании, и выводит его содержимое в строковом элементе отображения.

Задание 3. Создать ВП, который записывает данные в файл формата электронной таблицы.

Задание 4. Создать ВП, который записывает данные ВП «Контроль температуры» в файл формата электронной таблицы с регистрацией времени и текущего значения температуры.

14.2 Порядок выполнения лабораторной работы

14.2.1 Для выполнения лабораторной работы необходимо зарегистрироваться в системе:

- загрузить файл Base_LabView, появится окно выбора лабораторных работ (рисунок 1.1), в главном меню LabView нажать кнопку «RUN»;
- выбрать лабораторную работу «Файловый ввод/вывод»;
- появится окно регистрации магистранта: ввести свои Фамилию, Имя; ввести номер группы; нажать кнопку «Далее».

В открывшемся меню LabView выбрать «New - Blank VI» - новый ВП.

14.2.2 Задание 1. Создать ВП, который записывает данные в файл и отображает путь к созданному файлу.

Стандартные операции ввода/вывода данных в/из файла включают последовательность из четырех действий:

- создание/открытие файла - функция «Open/Create/Replace File»;
- операция считывания/записи данных в/из файла – функции «Read File» / «Write File»;
- закрытие файла – функция «Close File»;
- обработка ошибок – подпрограмма обработки ошибок «Simple Error Handler.vi ».

14.2.2.1 На фронт-панели собрать интерфейс задачи, как показано на рисунке 14.1.

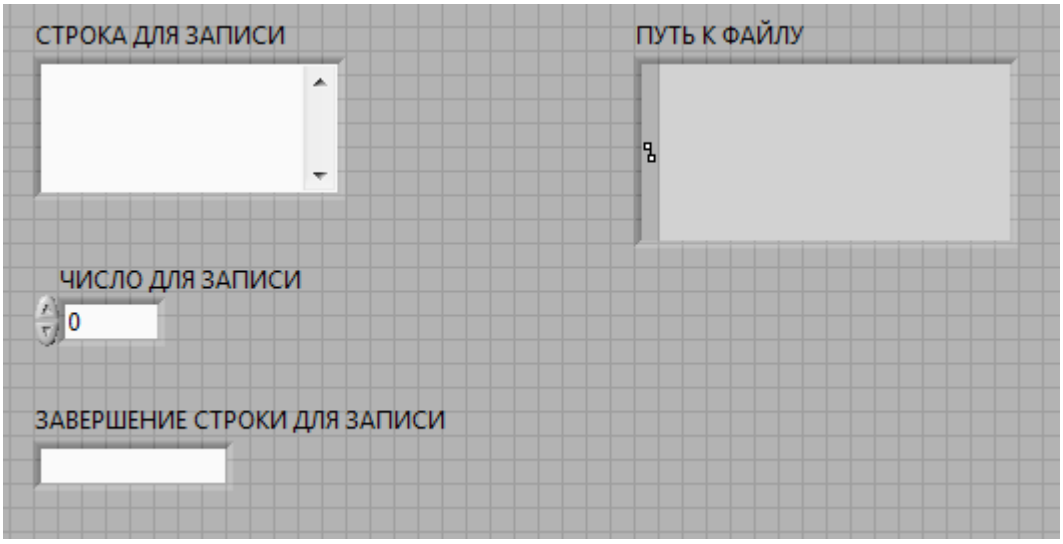


Рисунок 14.1

1) Поместить на фронт-панель два строковых элемента управления «String Ctrl» из меню «Controls – Express – Text Ctrls» под именем «Строка для записи» и «Завершение строки для записи».

2) Поместить числовой элемент управления под именем «Число для записи».

3) Поместить элемент «File Path Indicator» под именем «Путь к файлу» из меню «Controls – Modern – String & Path».

14.2.2.2 Построить блок-диаграмму, как показано на рисунке 14.2.

Функции файлового ввода/вывода находятся в меню «Functions – Programming – File I/O».

1) Поместить функцию «Open/Create/Replace File», которая выводит на экран диалоговое окно для создания файла:

а) щелкнуть правой кнопкой мыши по входу «Prompt» и в контекстном меню выбрать «Create - Constant» для создания константы «Введите имя файла»;

б) щелкнуть правой кнопкой мыши по входу «Operation» и в контекстном меню выбрать «Create - Constant» для выбора из выпадающего меню «Create».

2) Поместить функцию «Write To Text File», которая записывает строку в файл.

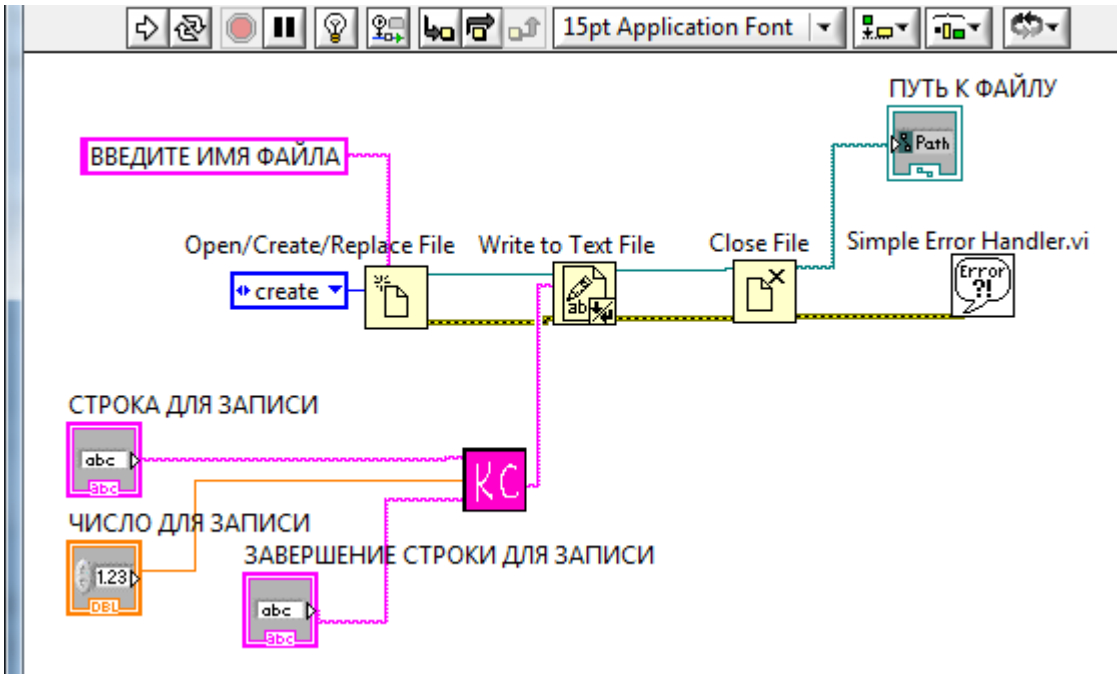


Рисунок 14.2

3) Поместить функцию «Close File», которая предназначена для закрытия файла.

4) Поместить подпрограмму ВП «Simple Error Handler.vi», расположенную в меню «Function – Programming – Dialog & User Interface», для проверки кластера ошибок и вывода диалогового окна при возникновении ошибок.

5) Соединить три первые функции по входам/выходам «refnum» и все четыре функции по входам/выходам «error», как показано на рисунке 14.2.

6) Поместить подпрограмму ВП «Комбинированная строка», созданную в п.13.2.2 и объединяющую три строки в одну, из меню «Functions – Select a VI…».

14.2.2.3 Сохранить ВП под именем «Запись файла».

14.2.2.4 Перейти на фронт-панель, ввести исходные данные. Например, «Строка для записи» - «Результат измерения равен», «Число для записи» - «5», «Завершение строки для записи» - «Вольт».

14.2.2.5 Запустить ВП на выполнение с помощью однократного «RUN».

Появится диалоговое окно «Введите имя файла». Введите в диалоговое окно имя файла с расширением «.txt» и нажмите «Save» или «Ok».

14.2.2.6 Проанализировать и сканировать результат выполнения программы.

14.2.3 В главном меню выбрать «New - Blank VI» - новый ВП.

Задание 2. Создать ВП, который читает файл, созданный в предыдущем задании, и выводит его содержимое в строковом элементе отображения.

14.2.3.1 На фронт-панели собрать интерфейс задачи, как показано на рисунке 14.3.

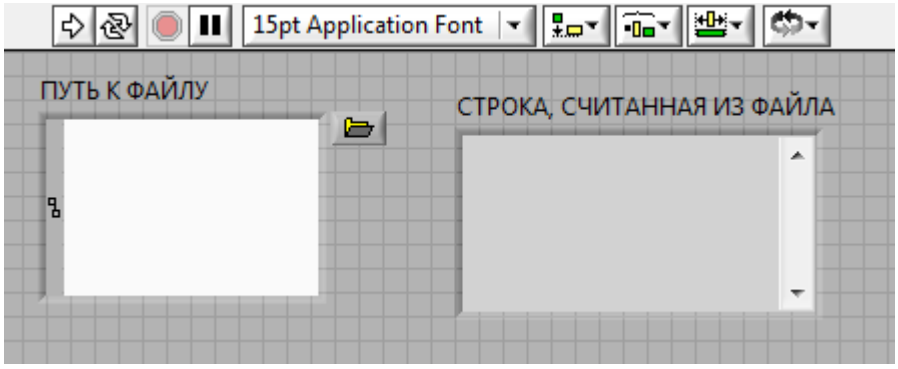


Рисунок 14.3

1) Поместить элемент «File Path Control» под именем «Путь к файлу» из меню «Controls – Modern – String & Path».

2) Поместить на фронт-панель строковый элемент отображения «String Ind» из меню «Controls – Express – Text Inds» под именем «Строка, считанная из файла.

14.2.3.2 Построить блок-диаграмму, как показано на рисунке 14.4.

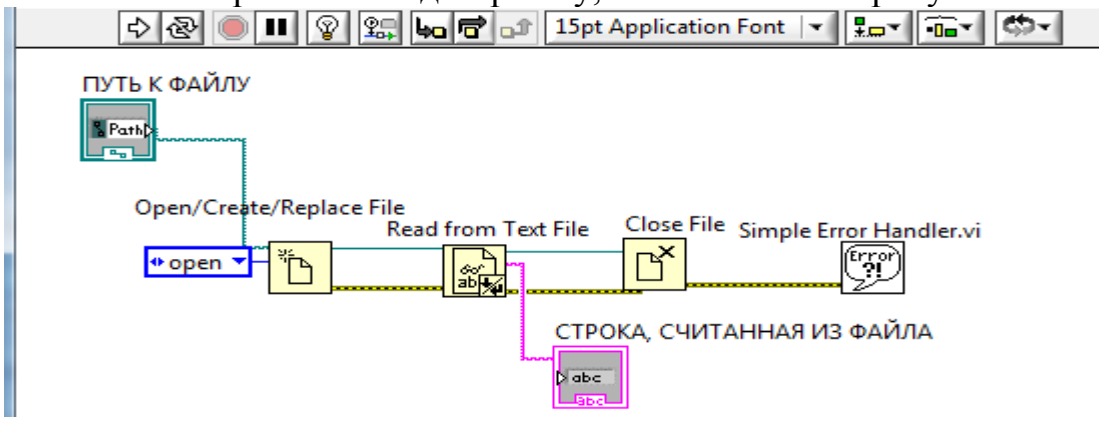


Рисунок 14.4

1) Поместить цепочку функций «Open/Create/Replace File», «Read from Text File», «Close File», ВП «Simple Error Handler.vi», аналогично предыдущему заданию (п.14.2.2.2).

2) Функция «Open/Create/Replace File»:

а) соединить «Путь к файлу» со входом «file path» функции;

б) щелкнуть правой кнопкой мыши по входу «operation» и в контекстном меню выбрать «Create - Constant» для выбора из выпадающего меню «open».

3) Соединить «Строка, считанная из файла» с выходом «text» функции «Read from Text File».

14.2.3.3 Сохранить ВП под именем «Чтение файла».

14.2.3.4 Перейти на фронт-панель, ввести путь к файлу, для чего выбрать файл из раскрывающегося списка, щелкнув на значок папки справа от «Путь к файлу».

14.2.3.5 Запустить ВП на выполнение с помощью однократного «RUN».

14.2.3.6 Проанализировать и сканировать результат выполнения программы.

14.2.4 Задание 3. Создать ВП, который записывает данные в файл формата электронной таблицы.

14.2.4.1 В главном меню выбрать «New - Blank VI» - новый ВП. Перейти на блок-диаграмму. Построить блок-диаграмму, как показано на рисунке 14.6.

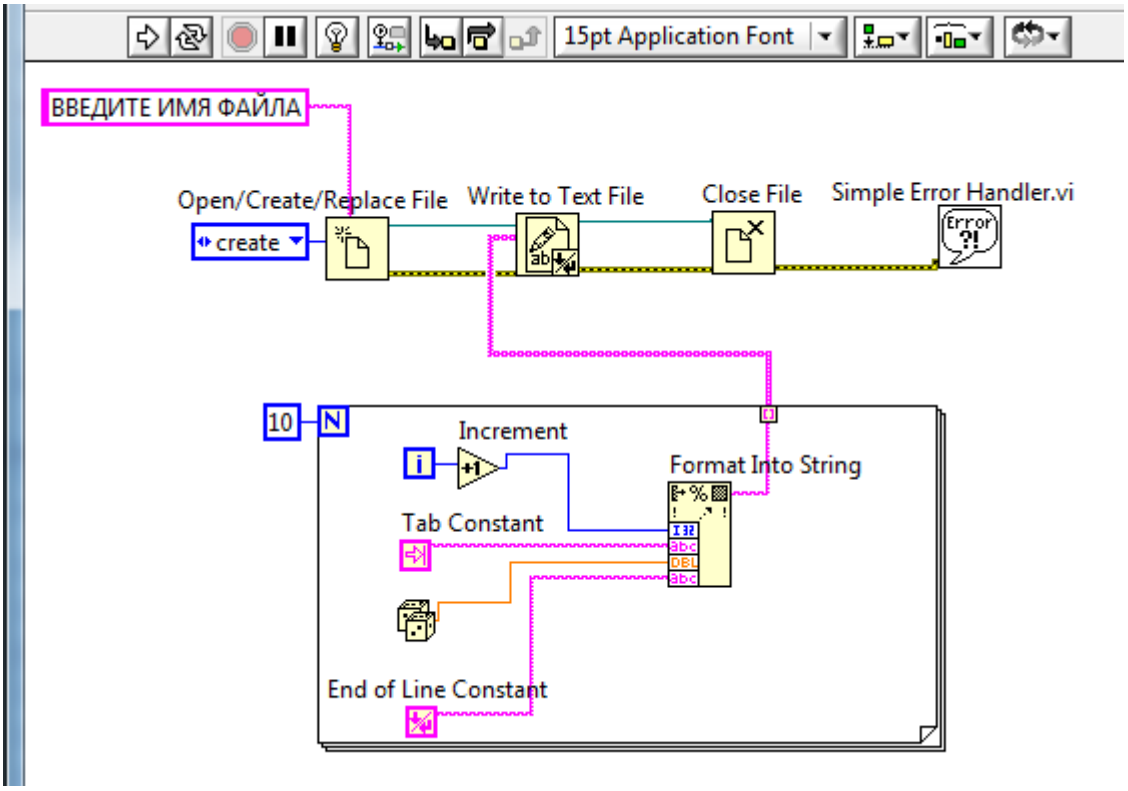


Рисунок 14.6

1) Поместить цепочку функций «Open/Create/Replace File», «Write To Text File», «Close File», ВП «Simple Error Handler.vi», аналогично заданию 1 (п.14.2.2.2).

2) Поместить функцию «Format Into String», которая преобразует значения счетчика итераций и случайное число в строку внутрь цикла «For»:

а) растянуть функцию на четыре входа;

б) на первый вход подать значения счетчика итераций +1, используя функцию «Increment» из меню «Functions – Programming - Numeric», поскольку итерации цикла начинаются с 0;

в) на второй вход подать функцию табуляции «Tab Constant» из меню «Functions – Programming - String» для создания столбцов таблицы;

г) на третий вход подать случайные числа от генератора «Random Number»;

д) на четвертый вход подать функцию конца строки «End of Line Constant» из меню «Functions – Programming - String»;

е) количество итераций цикла N задает количество строк в таблице.

14.2.4.2 Сохранить ВП под именем «Запись таблицы символов».

14.2.4.3 Запустить ВП на выполнение с помощью однократного «RUN».

Появится диалоговое окно «Введите имя файла». Введите в диалоговое окно имя файла с расширением «.xls» и нажмите «Save» или «Ok».

ВП создаст электронную таблицу Excel.

14.2.4.4 Открыть созданный файл Excel, проанализировать и сканировать результат выполнения программы.

14.2.5 Задание 4. Создать ВП, который записывает данные ВП «Контроль температуры» в файл формата электронной таблицы с регистрацией времени и текущего значения температуры.

14.2.5.1 Открыть ВП «Контроль температуры» и сохранить его под именем «Регистратор температуры». Фронт-панель ВП оставить без изменений.

14.2.5.2 Изменить блок-диаграмму, как показано на рисунке 14.7.

1) Растянуть границу цикла вверх, освободив место для следующих функций.

2) Поместить цепочку функций «Open/Create/Replace File», «Write To Text File», «Close File», ВП «Simple Error Handler.vi», аналогично заданию 1 (п.14.2.2.2).

3) Поместить функцию «Get Date/Time String», которая выдает время проведения измерения температуры в строковом режиме, из меню «Functions – Programming - Timing». Щелкнуть правой кнопкой по входу «want seconds?» (нужны секунды) и в контекстном меню выбрать «Create - Constant». Появится логическая константа: изменить ее значение с «False» на значение «True», что означает «включить секунды в строку».

4) Поместить экспресс-ВП «Build Text Express.vi» из меню «Functions – Express - Output». Этот ВП преобразует несколько входных строк в одну строку:

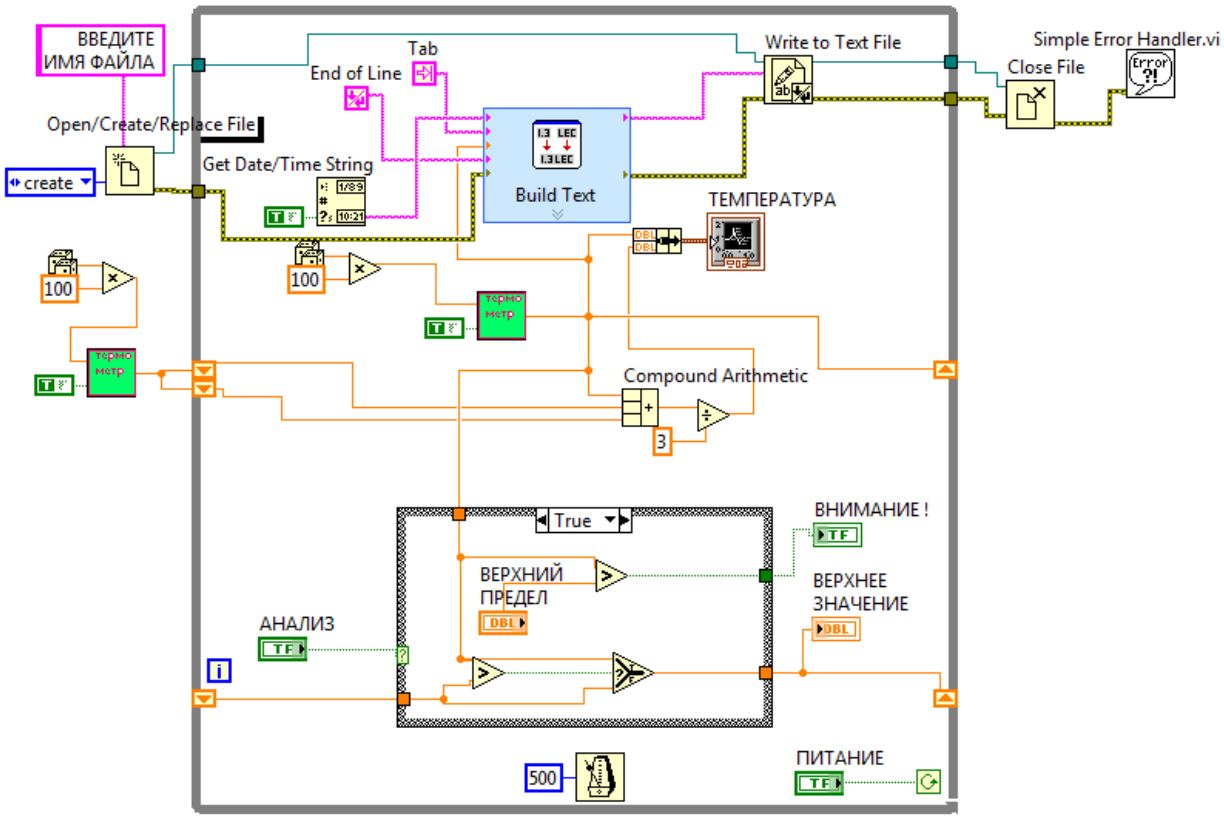


Рисунок 14.7

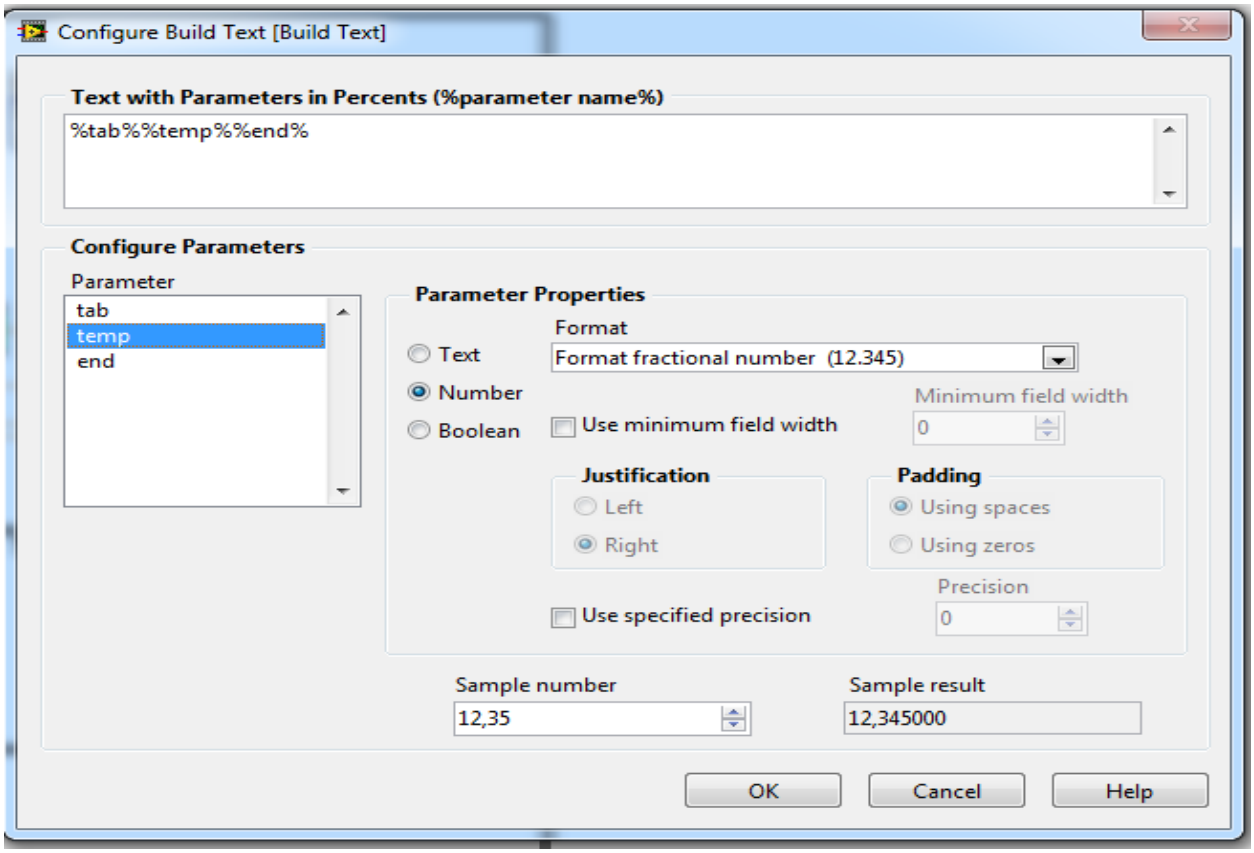


Рисунок 14.8

а) щелкнуть на иконку экспресс-ВП, появится диалоговое окно «Configure Build Text» (рисунок 14.8) для определения трех входных переменных: «tab» - константа табуляции, «temp» - значение входной температуры, «end» - константа конца строки;

б) в поле диалогового окна «Text with Variable in Percents» ввести строку «%tab%%temp%%end%»;

в) в разделе диалогового окна «Configure parameters» выбрать переменную «temp». В разделе «Parameter Properties» отметить пункт «Number» и формат дробного числа «Format fractional number»;

г) нажать кнопку «Ok» для закрытия диалогового окна;

д) на вход «Beginning Text» подать сигнал с выхода функции «Get Date/Time String»;

е) поместить константы «Tab Constant» и «End of Line Constant» из меню «Functions – Programming - String» и подать их соответственно на входы «tab» и «end»;

ж) (по желанию) для экономии места на блок-диаграмме щелкнуть правой кнопкой мыши по ВП «Build Text» и выбрать «View as Icon» из контекстного меню.

14.2.5.3 Запустить ВП на выполнение с помощью однократного «RUN».

Появится диалоговое окно «Введите имя файла». Введите в диалоговое окно имя файла с расширением «.xls» и нажмите «Save» или «Ok».

ВП создаст электронную таблицу Excel.

14.2.5.4 Открыть созданный файл Excel, проанализировать и сканировать результат выполнения программы.

14.3 Содержание отчета

Отчет должен содержать:

- цель и задание к лабораторной работе;
- screen shot интерфейса всех заданий;
- разработанные блок-диаграммы ВП;
- описание элементов управления и индикации на блок-диаграмме;
- выводы по работе.

14.4 Контрольные вопросы

14.4.1 В каком меню расположены функции файлового ввода/вывода?

14.4.2 Последовательность стандартных действий по файловому вводу/выводу данных.

14.4.3 Какая подпрограмма проверяет кластер ошибок?

14.4.4 Какая функция формирует строки таблицы символов?

14.4.5 Какая функция формирует системное время компьютера в строковом формате?

14.4.6 Диалоговое окно экспресс-ВП «Build Text Express.vi».

14.4.7 Что нужно предпринять для экономии места на блок-диаграмме?

Список литературы

1 Батоврин В.К., Бессонов А.С., Мошкин В.В., Папуловский В.Ф. LabView: практикум по основам измерительных технологий. – М.: ДМК Пресс, 2009. – 232 с.

2 Пейч Л.И., Точилин Д.А., Поллак Б.П. LabView для новичков и специалистов. – М.: Горячая линия – Телеком, 2006. – 384 с.

3 Суранов А.Я. LabView 7: справочник по функциям. – М.: ДМК Пресс, 2006. – 512 с.

4 Тревис Дж. LabView для всех. – М.: ДМК Пресс, 2012. – 544 с.

5 Хан С.Г. Проектирование виртуальных средств измерений. Методические указания по выполнению лабораторных работ для магистрантов специальности 6М070200 – Автоматизация и управление. – Алматы: АУЭС, 2017. – 75 с.

Сводный план 2018 г., поз.112

Светлана Гурьевна Хан

ГРАФИЧЕСКОЕ ПРОГРАММИРОВАНИЕ В ЗАДАЧАХ ИЗМЕРЕНИЯ И АВТОМАТИЗАЦИИ

Методические указания по выполнению лабораторных работ
для студентов специальности
5В070200 – Автоматизация и управление

Редактор Н.М. Голева
Специалист по стандартизации Н.К. Молдабекова

Подписано в печать __. __. __.
Тираж 50 экз.
Объем 5,9 уч.-изд. л.

Формат 60х84 1/ 16
Бумага типографская №1
Заказ _____. Цена 3000 тг.

Копировально-множительное бюро
некоммерческого акционерного общества
«Алматинский университет энергетики и связи»
050013, Алматы, Байтурсынова, 126