

ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ
БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ
им. проф. М. А. БОНЧ-БРУЕВИЧА»
(спбгут)

Т. В. Мусаева

МОДЕЛИРОВАНИЕ
ТЕЛЕКОММУНИКАЦИОННЫХ
УСТРОЙСТВ

ГРАФИЧЕСКАЯ СРЕДА
ПРОГРАММИРОВАНИЯ

Практикум

СПб ГУТ)))

САНКТ-ПЕТЕРБУРГ
2018

УДК 004.92(076)
ББК 32.973-018я73
М 91

Рецензент
кандидат технических наук, доцент,
заведующий кафедрой конструирования и производства радиоэлектронных
средств СПбГУТ
И. С. Кирик

*Рекомендован к печати
редакционно-издательским советом СПбГУТ*

Мусаева, Т. В.
М 91 Моделирование телекоммуникационных устройств. Графическая
среда программирования : практикум / Т. В. Мусаева ; СПбГУТ. –
СПб., 2018. – с.

Даны методические рекомендации по выполнению заданий в
программе LabView.

Предназначено для студентов очной, очно-заочной и заочной
форм обучения, а также для обучения с использованием
дистанционных технологий по направлениям подготовки: 09.03.02
«Информационные системы и технологии», 11.03.02
«Инфокоммуникационные технологии и системы связи».

УДК 004.92(076)
ББК 32.973-018я73

© Мусаева Т. В., 2018

© Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Санкт-Петербургский государственный университет
телекоммуникаций им. проф. М. А. Бонч-Бруевича», 2018

СОДЕРЖАНИЕ

Введение	4
1. ИЗУЧЕНИЕ КОМАНД ИНТЕРФЕЙСА И ОСНОВНЫХ ФУНКЦИЙ СРЕДЫ ГРАФИЧЕСКОГО ПРОГРАММИРОВАНИЯ LabView	5
1.1. Команды главного меню	6
1.2. Панели инструментов	6
1.3. Палитра инструментов Tools Palette	8
1.4. Палитра элементов управления и индикаторов Controls Palette	8
1.5. Палитра функций Functions Palette	10
1.6. Соединение элементов	12
2. СОЗДАНИЕ ВИРТУАЛЬНЫХ ПРИБОРОВ	14
2.1. Запуск программы LabView	14
2.2. Создание простейшего виртуального прибора	15
3. ТИПЫ ДАННЫХ, ФУНКЦИИ	19
3.1. Типы данных	19
3.2. Функции математических операций и сравнения	19
<i>Практические задания</i>	21
4. СТРУКТУРЫ, ФУНКЦИИ ГЕНЕРАЦИИ И ОБРАБОТКИ ДАННЫХ	28
4.1. Структуры	28
4.2. Функции генерации и обработки данных	28
<i>Практические задания</i>	29
Заключение	55
Список литературы	56

ВВЕДЕНИЕ

Практикум предназначен для получения и закрепления практических навыков студентами по темам рабочей программы дисциплины. Материал содержит теоретическую справочную информацию, примеры и указания по выполнению заданий в среде графического программирования.

Полученные знания и навыки помогут в дальнейшем изучать последующие дисциплины по учебному плану, в которых предполагается использование современных средств информационных технологий, представленных средой графического проектирования и программирования.

Цель выполнения заданий: ознакомление, приобретение и закрепление практических навыков по применению современных средств информационных технологий для решения задач в профессиональной деятельности.

В результате обучения студент должен:

уметь:

- моделировать виртуальные приборы, осуществляющие измерения;
- настраивать работу виртуальных приборов;
- применять базовые шаблоны и архитектуры в приложении, использовать различные приемы редактирования и отладки;

знать:

- принципы и технологию создания виртуальных устройств;
- методику создания пользовательских интерфейсов;
- приемы редактирования, поиска ошибок и отладки работы устройств;

владеть:

- навыками работы в среде LabView.

Каждая тема содержит теоретическую и практическую часть. Задания могут выполняться в программе LabView версии 2010 и выше. Для выполнения работ требуются знания и навыки, полученные при изучении дисциплины «Информатика и основы программирования». В каждой из тем дается краткая теоретическая информация, необходимая для выполнения практического задания.

Далее описывается краткая методика по созданию виртуального прибора или подприбора с подсказками в виде иллюстраций. В процессе выполнения задания изучается необходимый набор элементов управления и индикаторов, операции, функции, структуры, необходимые для разработки программы.

После получения знаний и навыков по практической части пособия, студенты выполняют индивидуальную работу на произвольную тему. В отчете (индивидуально выполненной работе) студенты должны максимально показать навыки и знания, полученные в процессе изучения возможностей и функций программного средства.

Для самостоятельного выполнения индивидуальной работы могут быть использованы примеры и образцы заданий.

Работа сдается на проверку преподавателю и защищается в указанные сроки в виде электронной версии отчета (созданного виртуального прибора).

1. ИЗУЧЕНИЕ КОМАНД ИНТЕРФЕЙСА И ОСНОВНЫХ ФУНКЦИЙ СРЕДЫ ГРАФИЧЕСКОГО ПРОГРАММИРОВАНИЯ LabView

Все программы, разрабатываемые в среде графического программирования (СГП) LabView, называются виртуальными приборами (Virtual Instruments). Для создания программы, а точнее виртуального прибора, необходимо загрузить программу LabView, после чего одновременно появятся два окна:

- Front Panel (Лицевая панель);
- Block Diagram (Блок-диаграмма).

Далее в методическом пособии будут использованы термины Лицевая панель (ЛП) и Блок-диаграмма.

В окне Лицевой панели (рис. 1.1) разрабатывается внешний вид будущего виртуального прибора. На ней создаются элементы управления и отображения (индикации), которые являются интерактивными средствами ввода и вывода данных виртуального прибора.

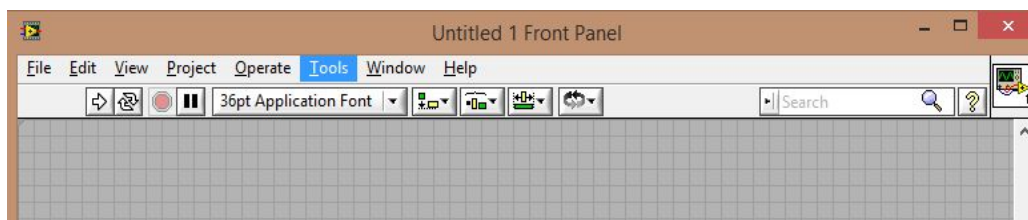


Рис. 1.1

В окне панели Блок-диаграммы (рис. 1.2) создается и записывается код программы.

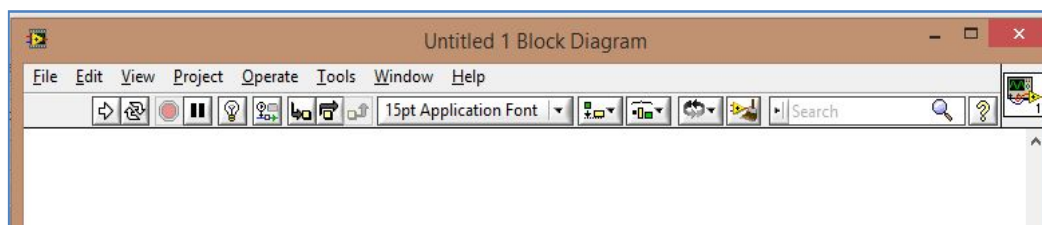


Рис. 1.2

Свободное пространство каждой панели образует рабочую область, снабженную горизонтальной и вертикальной полосами прокрутки. При разработке программ, в рабочей области Лицевой панели размещаются визуальные элементы управления и индикации, формирующие интерфейс пользователя, а в окне Блок-диаграммы составляется код программы. Для одновременного отображения на экране монитора обеих панелей используют

команду Windows ►Title Left and Right. В правом верхнем углу каждого окна находится пиктограмма для закрепления созданной программы в качестве нового виртуального прибора или подприбора.

В процессе разработки программы применяются инструменты трех вспомогательных палитр и команды главного меню.

К вспомогательным палитрам относятся:

- палитра инструментов (Tools Palette), активная на Лицевой панели и в Блок-диаграмме,
- палитра элементов управления и индикации (Controls Palette), активной только в окне Лицевой панели,
- палитра функций (Functions Palette), доступной в окне Блок-диаграммы.

1.1. Команды главного меню

Большинство команд главного меню являются стандартными для Windows-приложений.

Команды главного меню:











- 1) File (Файл) – создание, открытие, закрытие, сохранение, печать и др.;
- 2) Edit (Редактирование, правка) – редактирование панелей, поиск объектов и др.;
- 3) View (Просмотр) – просмотр и вызов панелей элементов, инструментов, функций, просмотр иерархий программ и др.;
- 4) Project (Проект) – создание нового проекта, открытие, сохранение, удаление, информация о проекте и др.;
- 5) Operate (Управление) – запуск и прерывание программ и др.;
- 6) Tools (Инструменты) – управление библиотеками программ и др.;
- 7) Window (Окно) – отображение окон и палитр LabView и др.;
- 8) Help (Справка) – дополнительная информация об элементах и функциях LabView.

1.2. Панели инструментов

Ниже полос главного меню расположены линейки инструментов, содержащие команды, некоторые из которых идентичны Лицевой панели и Блок-диаграммы, а некоторые различны с ними за счет дополнительных кнопок для отладки программ.




В табл. 1.1 представлены команды, расположенные на Лицевой панели инструментов.

Таблица 1.1

Название команды	Назначение	Кнопка
Run	Запуск программы на однократное исполнение	
Run Continuously	Запуск программы на многократное исполнение до принудительной остановки	
Abort Execution	Экстренная остановка выполнения программы	
Pause	Временная пауза выполнения программы	
Text Settings	Форматирование шрифта: текст, размер, стиль, цвет	
Align Objects	Выравнивание объектов по осям (по вертикали, по осям и т. д.)	
Distribute Objects	Выравнивание объектов в пространстве (промежутки, сжатие и т. д.)	
Resize Objects	Приведение к одному размеру многократно используемых объектов Лицевой панели	
Reorder	Применяется при наложении объектов друг на друга	
Context Help	Контекстная справка	






В табл. 1.2 представлены разновидности внешнего вида кнопки команды Run.

Таблица 1.2

Название команды	Назначение	Кнопка
Run	ВП высокого уровня	
Run	ВП работает в качестве подпрограммы	
Run	Ошибка в программе. После нажатия этой кнопки появляется окно Error list, в котором указаны ошибки. Запрещается выполнение программы	

В табл. 1.3 представлены команды панели инструментов Блок-диаграммы.











Таблица 1.3

Название команды	Назначение	Кнопка
Highlight Execution	Просмотр исполнения программы и передачи потока данных	
Step Into	Начало пошагового выполнения цикла программы от узла к узлу	
Step Over	Позволяет пропустить в пошаговом режиме цикл, подпрограмму и т. д	
Step Out	Выход из пошагового выполнения программы.	
Warning	Появляется при потенциальной проблеме в блок-диаграмме, но выполнении программы ВП не запрещается	

1.3. Палитра инструментов Tools Palette

Создавать, изменять и отлаживать ВП можно, используя палитру инструментов Tools Palette. Описание назначения элементов палитры инструментов представлено в табл. 1.4.

Таблица 1.4

Назначение	Вид инструментов
Управление. Для изменения значений элементов управления, ввода данных	
Перемещение, прокрутка экрана без использования полосы прокрутки	
Маркер. Для ввода данных в поле названия элементов, в устройства управления, в структуры при введении данных, в тоннели и т. д.	
Для вызова контекстного меню	
Контрольные точки. Для расстановки контрольных точек на ВП, узлах, проводниках, структурах, для остановки выполнения программы	
Копирование цвета. Для копирования и вставки цвета	
Перемещение, расположение. Для активизации, перемещения, изменения размеров объекта	
Проводник данных. Для соединения элементов управления и индикации между собой, для передачи данных	
Пробник. Установка отладочных индикаторов. Для отладки работы ВП и просмотра промежуточных сигналов	
Инструмент цвета. Для задания цвета элементам, названиям, окнам лицевой панели и Блок-диаграммы	

1.4. Палитра элементов управления и индикаторов Controls Palette

Для размещения элементов управления и отображения данных на Лицевой панели используется палитра элементов управления и отображения (рис. 1.3). Палитра вызывается при нажатии правой кнопкой мыши в рабочем пространстве лицевой панели.

Все элементы управления и отображения на палитре сгруппированы по классификационному признаку:

- Modern – современный стиль элементов управления и отображения;
- System – стиль элементов управления и отображения, соответствующий данной операционной системе;
- Classic – классический стиль (альтернатива стилю Modern) элементов управления и отображения;
- Express – распространенные элементы управления и отображения и т. д.;

Каждый раздел может делиться в свою очередь на подразделы (числовые, строковые индикаторы, кнопки и т. д.).

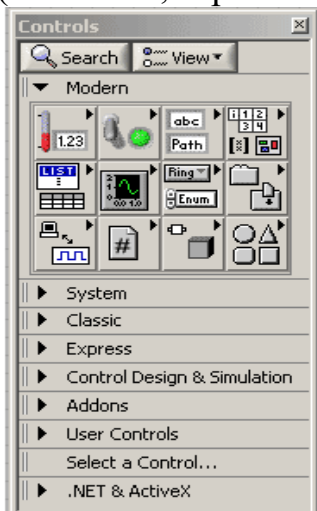


Рис. 1.3

В табл. 1.5 представлен набор библиотек элементов, сгруппированных по признаку типа данных.

Таблица 1.5

Название библиотеки	Состав	Элементы библиотеки
Numeric (числовые значения)	Средства управления и индикаторы для числовых данных.	
Boolean (двоичные (булевы) значения)	Средства управления и индикаторы для двоичных величин	
String (строковые значения)	Средства управления и индикаторы для ASCII строк и таблиц	
List & Ring (списки закольцованных списков)	Средства управления и индикаторы для меню, выполненные в форме списков	
Graph	Индикаторы для построения графика данных в графах или диаграммах в реальном масштабе времени (осциллограф)	
Array & Cluster (массивы и кластеры)	Средства управления и индикаторы для группировки наборов типов данных	
Decorations (оформление)	Графические объекты для настройки дисплеев передней панели	
Control	Диалоговое окно для загрузки самодельных элементов управления	
ActiveX (объекты ActiveX)	Средства управления, позволяющие внедрить объекты ActiveX на переднюю панель	
Path & Refnum (пути и ссылки).	Средства управления и индикаторы для путей и ссылок	
User Controls (средства управления пользователя)	Специальные средства управления, которые формирует сам пользователь	

1.5. Палитра функций Functions Palette

Для разработки исходного теста программы в окне Блок-диаграммы используется палитра функций Functions Palette. Каждая опция в палитре отображает подпалитру значков верхнего уровня. Активация палитры Functions осуществляется с помощью опции Show Functions Palette в главном меню. Палитра Functions может быть либо зафиксирована на рабочем столе с помощью кнопки в левом углу палитры, либо убрана кнопкой «крестик», вызывается при нажатии правой кнопкой мыши в рабочем пространстве Блок-диаграммы.

Группировка содержимого палитры Functions осуществлена по классификационному функциональному признаку назначения элементов (рис. 1.4). На рис. 1.5 представлена подпалитра выбранной функции основной палитры.

Классификационный состав некоторых элементов по функциональному назначению следующий:

- Programming – основные функции, структуры цикла, сравнения, операторы сложения, вычитания, необходимые для создания большинства приложений;
- Instrument I/O – функции для работы с различными устройствами ввода-вывода (COM-порт и др.);
- Mathematics – функции для математических вычислений (решение системы уравнений, решение дифференциальных уравнений и др.);
- Signal Processing – функции, связанные с цифровой обработкой и анализом дискретных сигналов (цифровые фильтры, быстрое преобразование Фурье и др.);
- Express – распространенные функции, связанные как с вычислениями, так и с обработкой данных;
- Select a VI – функции, созданные пользователем и т. д.

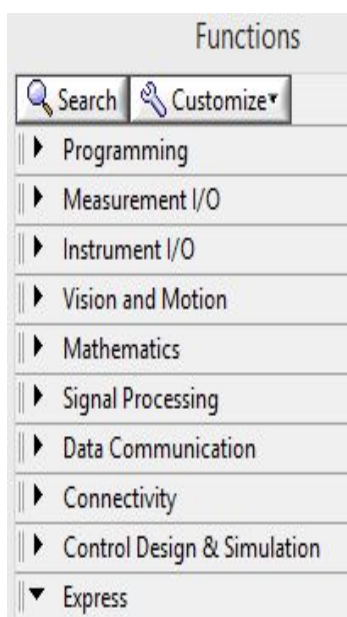


Рис.1. 4

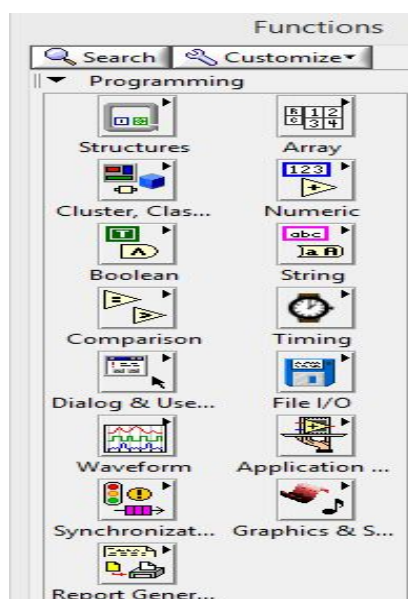


Рис.1. 5

В табл. 1.6 представлено описание подпалитр палитры Programming.

Название библиотеки	Состав	Функции
Structures (структуры)	Набор управляющих структур программы, таких как For Loop, While Loop, Formula Node и др.	
Array (массивы)	Функции для обработки массивов	
Numeric (числовые функции)	Тригонометрические, логарифмические и числовые функции	
Boolean (булевы функции)	Логические функции	
Comparison (сравнения)	Функции для сравнения числовых, строковых значений, булевых переменных	
String (строковые функции)	Функции для работы со строковыми величинами	
Cluster (кластеры)	Функции для обработки кластеров	
Time & Dialog	Функции для диалоговых окон, синхронизации, и обработки ошибок	
File I/O (ввод/вывода файла)	Функции и ВП для ввода/вывода в файл	
Instrument I/O (инструменты ввода/вывода)	ВП для связи и управления приборами по шинам GPIB, VISA	
Communication (связь)	ВП для работы с сетями TCP, DDE, Apple Events и OLE	
Data Acquisition (сбор данных)	ВП для внедрения плат сбора данных	
Analysis (анализ)	ВП для анализа данных	
Tutorial (обучающая программа)	ВП, используемые в обучающей программе LabView	
Advanced (расширенная)	Разные функции типа функции библиотечного запроса, манипуляции данных и т. д.	
VIE	Диалоговое окно для внедрения подпрограмм в текущий ВП	
Instrument Drivers (драйверы приборов)	ВП, способные управлять внешними измерительными приборами (осциллоскопами, генераторами и т. д.) через последовательный порт или интерфейс GPIB	
Application Control (управление приложением)	управляющие виртуальные приборы (ВП), позволяющие запускать ВП на других компьютерах через сеть	
User Libraries	Быстрый доступ к нужному ВП, разработанному пользователем	

Пример вызова группы элементов подпалитры Analysis (анализа) представлен на рис. 1.6.

Пример вызова группы элементов подпалитры All Function ►Numeric представлен на рис. 1.7.

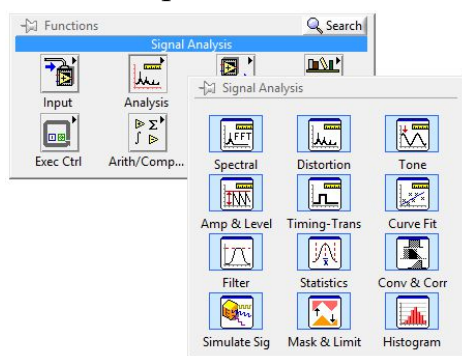


Рис. 1.6

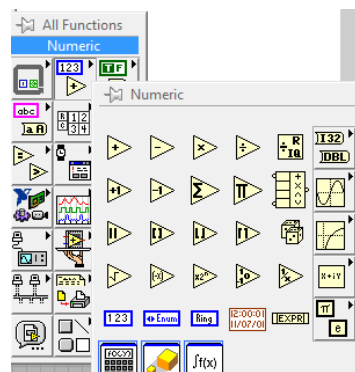


Рис. 1.7

1.6. Соединение элементов

Объекты Лицевой панели в Блок-диаграмме отображаются в виде пиктограмм терминалов данных (рис. 1.8). Терминалы данных обеспечивают обмен данными между Лицевой панелью и Блок-диаграммой.

Различают терминалы данных следующих типов:

- управления и отображения (индикации) данных;
- узлов.

Терминалы управления и отображения относятся к средствам управления и отображения данных на Лицевой панели. Данные, введенные в элементы управления на Лицевой панели, поступают в Блок-диаграмму через терминалы управления.

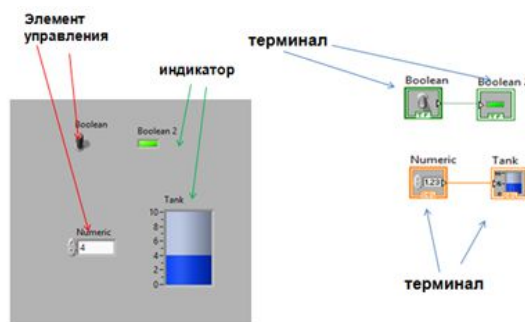


Рис. 1.8

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

Узлы (Node) – это объекты Блок-диаграммы, которые имеют одно или более полей ввода/вывода данных и выполняют алгоритмические операции

виртуального прибора. Они аналогичны операторам, функциям и подпрограммам текстовых языков программирования. Узлы включают в себя функции (functions), подпрограммы (SubVI) и структуры (Structures).

Подпрограмма – виртуальный прибор (ВП), который можно использовать на Блок-диаграмме другого ВП в качестве подпрограммы.

2. СОЗДАНИЕ ВИРТУАЛЬНЫХ ПРИБОРОВ

2.1. Запуск программы LabVIEW

Для начала работы в программе LabVIEW необходимо ее установить на компьютер.

Запустить программу.

Выбрать в диалоговом окне кнопку New (рис. 2.1).

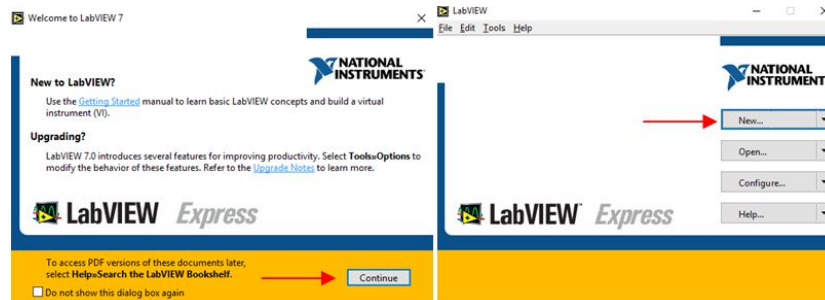


Рис. 2.1

В появившемся окне выбрать Blank VI, нажать ОК (рис. 2.2).

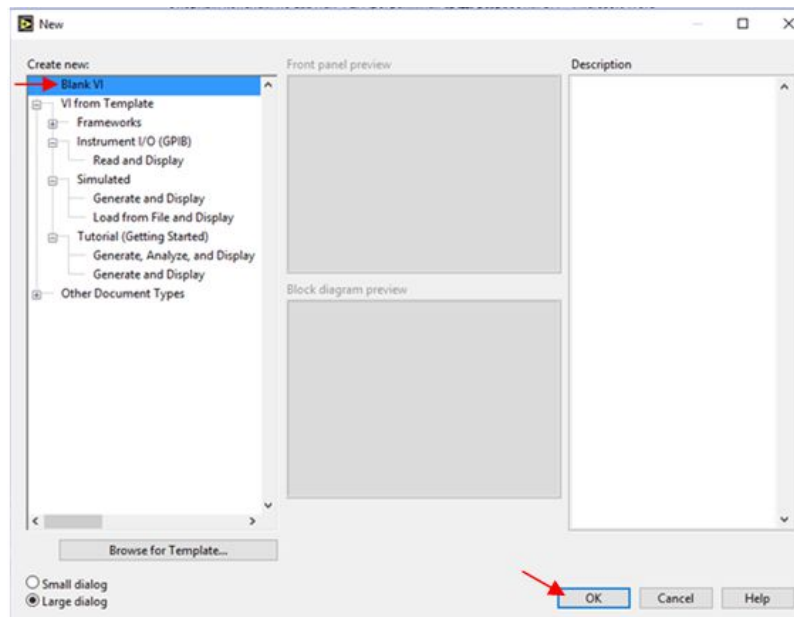


Рис. 2.2

В результате выполненных действий появятся два окна для начала работы и создания виртуального устройства (рис. 2.3).

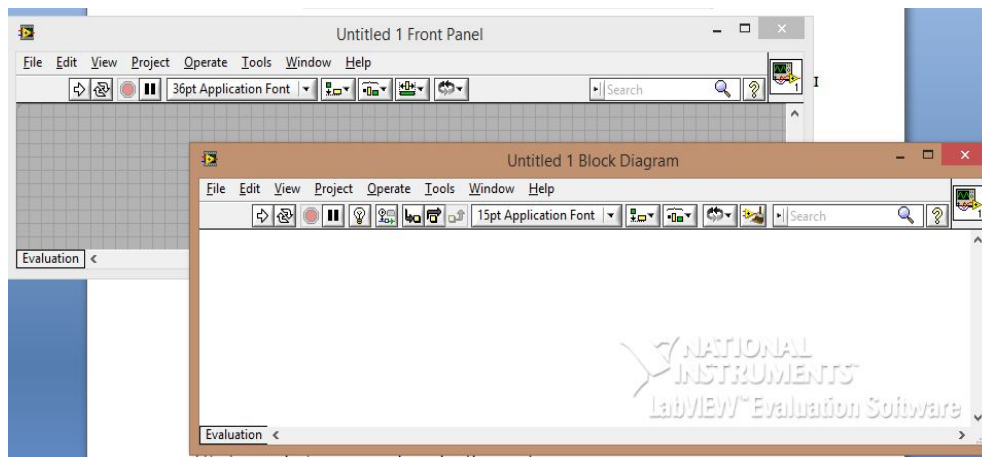


Рис. 2.3

На Лицевую панель устанавливаются элементы управления панели Controls или элементы отображения Indicators (рис. 2.4).

Элементы управления (Controls) – кнопки и другие устройства ввода данных.

Элементы отображения (Indicators) – графики, светодиоды и другие индикаторы.

Элементы управления моделируют устройства ввода данных и передают данные на Блок-диаграмму ВП. Элементы отображения моделируют устройства вывода и отображения данных, которые получает или генерирует блок-диаграмма. В окне Блок-диаграммы разрабатывается исходный код будущего виртуального прибора (рис. 2.5).

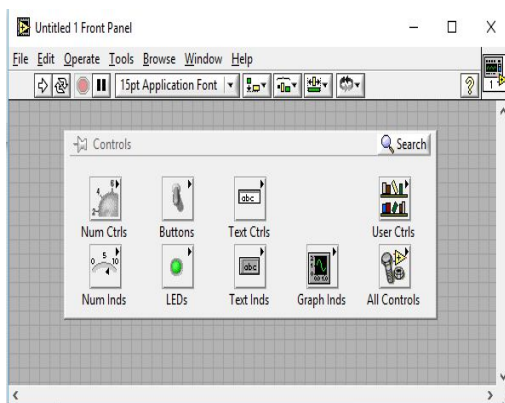


Рис. 2.4

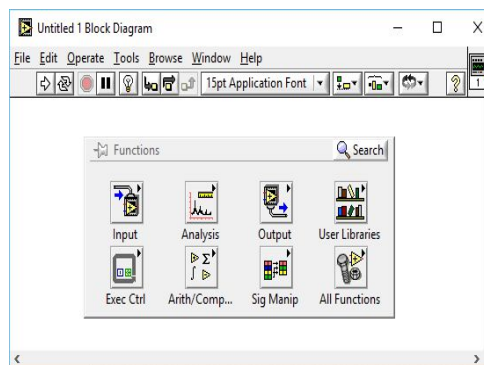


Рис. 2.5

Исходный код LabView представляет собой Блок-диаграмму, где все команды, операторы циклов и сравнения изображаются графическими обозначениями. Блок-диаграмма состоит из узлов, терминалов и проводников данных.

2.2. Создание простейшего виртуального прибора

Загрузить программу.

Активировать панель Controls Palette.

Установить на Лицевую панель (в соответствии с рис. 2.6.) два графических элемента, относящиеся к логическому типу данных.

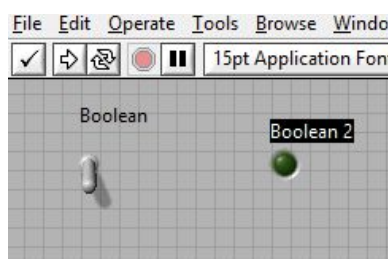


Рис. 2.6

Для этого необходимо выбрать в соответствии с рис. 2.7 элемент управления «Переключатель» и элемент индикации «Лампочка» и перетащить на Лицевую панель, удерживая кнопку мыши. Элементы появятся на Лицевой панели и в окне Блок-диаграммы (рис. 2.8,рис. 2.9).

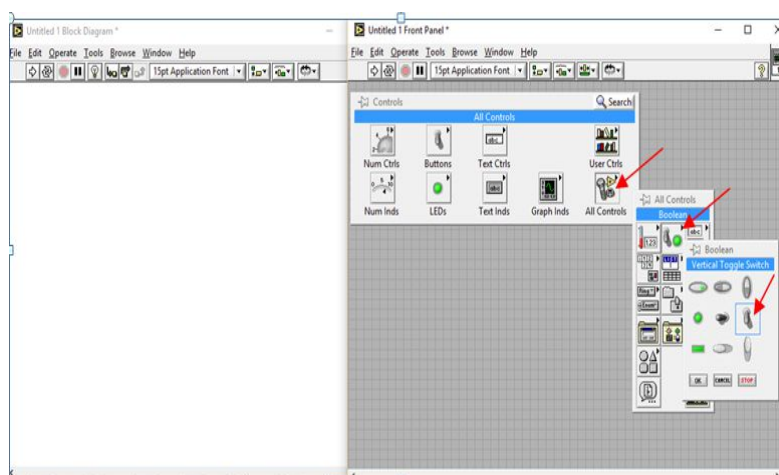


Рис. 2.7

Результат выбора и установки на Лицевую панель элемента управления – одновременное появление в окне Блок-диаграммы – представлен на рис. 2.8, рис. 2.9.

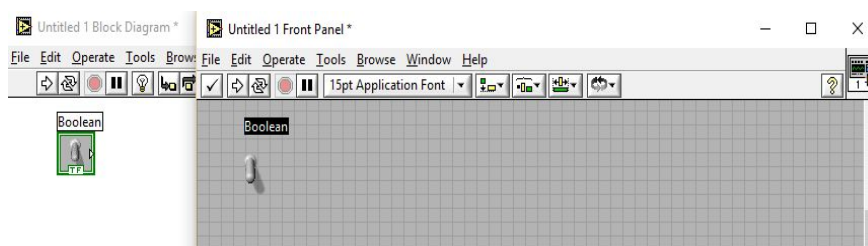


Рис. 2.8

Далее нужно установить второй элемент индикации (рис. 2.9).

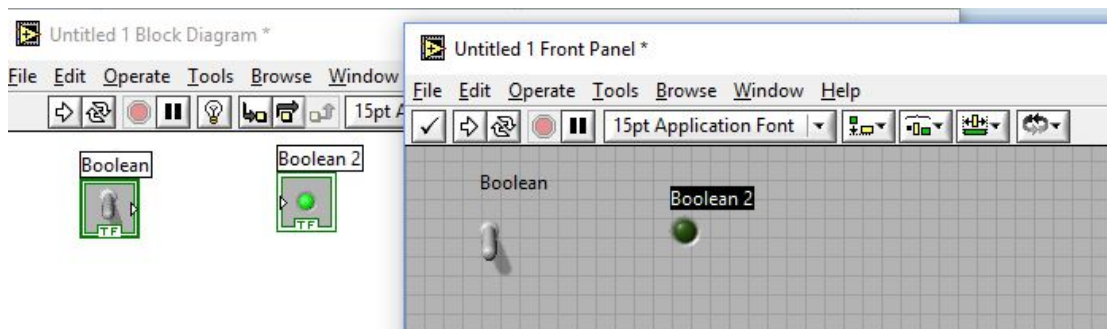


Рис. 2.9

Соединить элемент управления и индикации между собой проводником. Для этого необходимо активировать Tools Palette, выбрать пиктограмму катушки (табл. 1.4) и соединить элементы между собой протягиванием из узла выхода в узел входа.

Соединение двух элементов управления проводниками через узлы выхода и входа, представлено на рис. 2.10.



Рис. 2.10

От выходного узла элемента «Переключатель» осуществляется связь через проводник, т. е. вход, с элементом «Лампочка» (рис. 2.11). Далее необходимо запустить программу на многократное выполнение, выбрав соответствующую пиктограмму, отмеченную красной стрелкой (рис. 2.11).

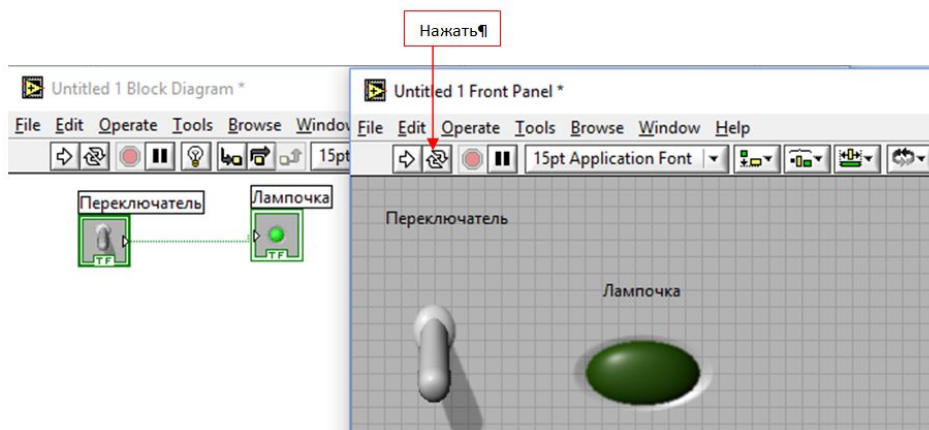


Рис. 2.11

Результат работающей программы созданного виртуального прибора (ВП) представлен на рис. 2.12 и 2.13. При нажатии на переключатель происходит либо включение, либо выключение лампочки.

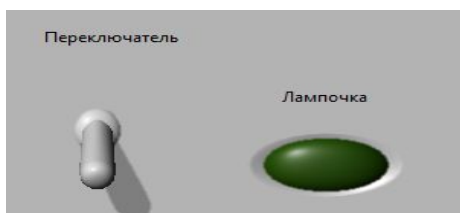


Рис. 2.12

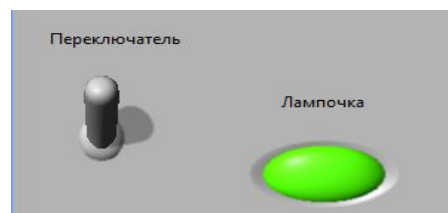


Рис. 2.13

В случае если будет допущена ошибка, программа выдаст соответствующую информацию о ее коде и предложит справку, а также варианты подсказок для ее исправления.

Для компактного расположения элементов Блок-диаграммы, нажать на пиктограмму Clean Up Diagram (рис. 2.14).

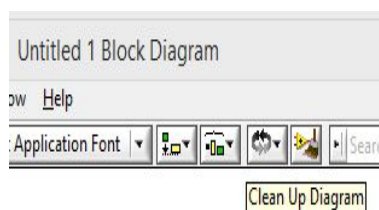


Рис. 2.14

Примечание. Во всех последующих заданиях необходимо применять команду Clean Up Diagram.

Также важно наносить провода прямыми линиями, избегать преломлений и изгибов.

3. ТИПЫ ДАННЫХ, ФУНКЦИИ

3.1. Типы данных

LabView содержит набор разных типов данных, которые представлены в виде графических элементов управления и индикации. Некоторые типы данных являются стандартными и аналогичны типам данных, используемых в других средах программирования, это число, логические данные, строка, массивы, кластеры и др. Также в LabView имеются другие типы данных, которые специфичны, не встречаются в других средах программирования, например, такие как осциллограмма (Waveform), сигнал (Signal), ресурс VISA, измерительный или управляющий канал и т. п.


Типы данных различаются цветом. Узнать тип переменной или константы можно с помощью команды Context Help, значок вопроса в окне Блок-диаграммы, или нажатием CTRL+H и наведением курсора мыши на объект. Таким же методом можно узнать назначение того или иного узла или подпрограммы.

Типы данных представлены на рис. 3.1.

Терминалы и провода в LabView имеют разный цвет и внешний вид, подсказывающие разработчику ВП, как и какие данные обрабатываются Блок-диаграммой.

3.2. Функции математических операций и сравнения

К функциям математических операций относятся: сложение, вычитание, умножение, деление, извлечение квадратного корня, вычисление среднего арифметического значения, случайного числа и т. д.

Элементы математических операций находятся в палитре функций Mathematics в подпалитре Numeric (рис. 3.2). Палитра математических функций вызывается в окне Блок-диаграммы. Здесь же устанавливаются элементы математических операций, например сложение .

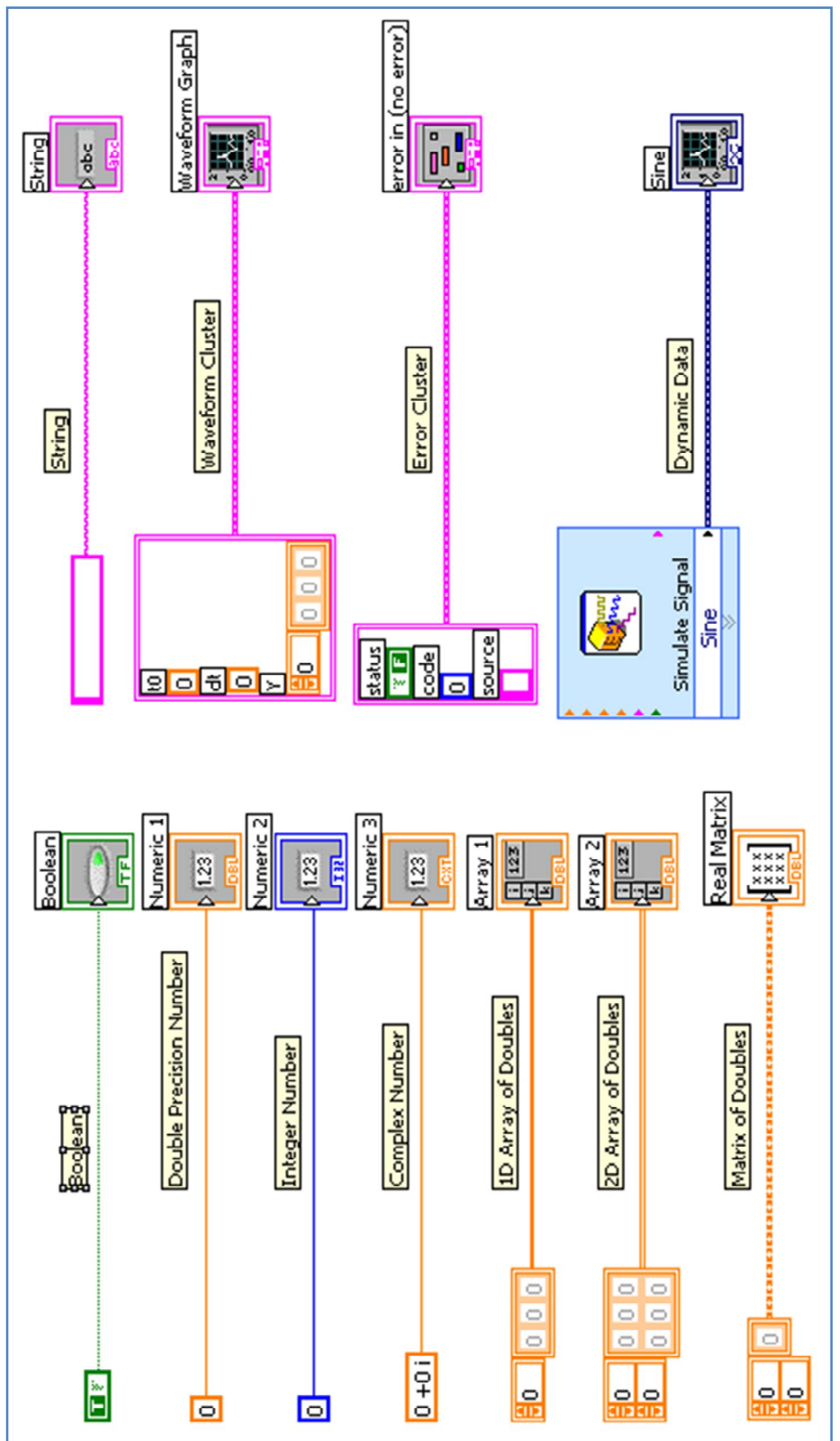


Рис. 3.1

К функциям сравнения относятся: больше, меньше, больше или равно, меньше или равно, равно нулю, не равно нулю и т. д.

Функция сравнения находится в подпалитре Comparison (рис. 1.5, табл. 1.6).

Практические задания

Задание 3.1. Создание ВП «Калькулятор».

Данное задание выполняется аналогично заданию 3.3.

Создать ВП, предварительно установив на Лицевую панель 2 элемента управления Numeric и 4 элемента индикации Numeric. Переименовать их в соответствии с рис. 3.2. Перейти в Блок-диаграмму и вставить математические операции сложения, умножения, деления, вычитания. Объединить все элементы и операции проводами в соответствии с рис. 3.2.

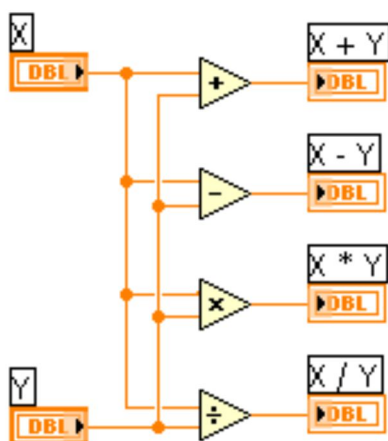


Рис. 3.2

Запустить прибор и проверить его работу. Сохранить файл.

Задание 3.2. Создание ВП «Вычисление среднего арифметического значения».

Создать ВП, предварительно установив на Лицевую панель:

- 3 элемента управления Numeric;
- 1 элемент индикации Numeric, переименовать его в «Average».

Перейти в Блок-диаграмму и вставить математическую функцию Сумматор, которая применяется для операций сложения, деления, умножения при количестве входных данных более чем два значения.

Объединить все элементы и операции проводами в соответствии с рис. 3.3.

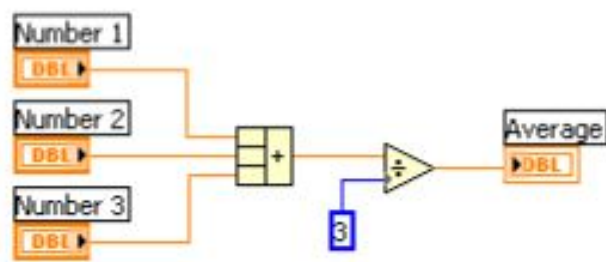


Рис.3.3

Запустить прибор и проверить его работу. Сохранить файл.

Задание 3.3. Создание ВП «Сравнение чисел».

Создать ВП, предварительно установив на Лицевую панель:

- 2 элемента управления Numeric, переименовать их в соответствии с рис. 3.4, щелкнув в поле названия Number 1 и Number 2;
- 1 элемент индикации Boolean (лампочка), и переименовать его в «Equal?».

- Перейти в Блок-диаграмму и вставить операцию сравнения «=». Объединить все элементы и операции проводами в соответствии с рис. 3.5.

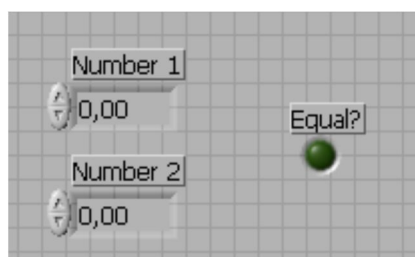


Рис. 3.4

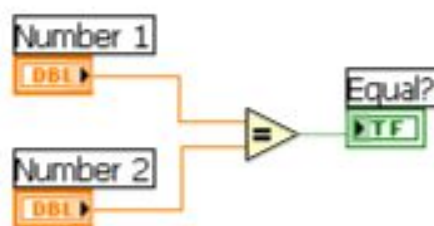


Рис. 3.5


Запустить прибор и проверить его работу. Сохранить файл.

Задание 3.4. Создание ВП «Деление на 0».

Создать ВП, предварительно установив на Лицевую панель:

- 1 элемент управления Numeric, переименовать его в «Делитель»;
- 2 элемента индикации Numeric, переименовать их в «Результат», «Случайное число-делимое»;
- 1 элемент индикации Boolean, переименовать его в «Ошибка».

Перейти в Блок-диаграмму и вставить функцию-элемент случайного

числа , элемент Numeric и ввести число 10, математические операции умножения, деления, сравнения «=0». Объединить все элементы и операции проводами в соответствии с рис. 3.6.

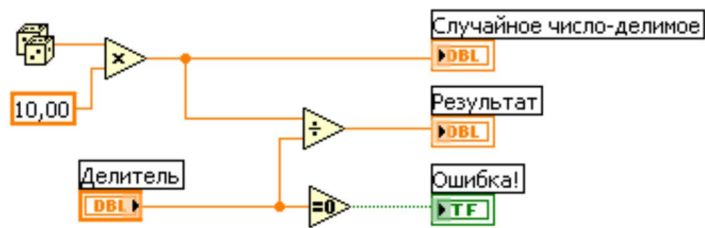


Рис. 3.6

Запустить прибор и проверить его работу. Сохранить файл.

Задание 3.5. Создание ВП «Перевод температуры».

Создать ВП перевода температуры в градусах Цельсия в температуру в градусах по Фаренгейту, в соответствии с рис. 3.7, 3.8.



Рис. 3.7

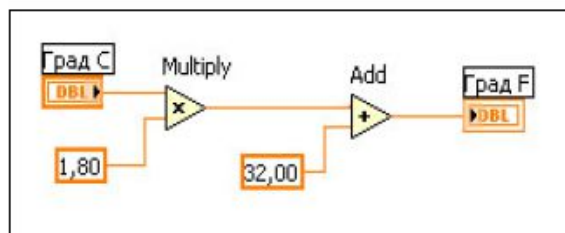


Рис. 3.8

Запустить прибор и проверить его работу. Сохранить файл.

Задание 3.6. Создание ВП «Строки».

Создать ВП, который преобразует число в строку и далее объединит эту строку с другими строками для получения одной строки на выходе. Виртуальный прибор также будет определять длину выходной строки.

Примечание. Функция Format Into String (Преобразовать в строку) преобразует заданное число в строку смешанного формата с четырьмя знаками точности.

Функция Concatenate Strings (Объединение строк) объединяет все входные строки в одну выходную. Для увеличения количества входов растяните иконку с помощью инструмента перемещения («стрелка»).

Функция String Length (Длина строки) возвращает количество символов в строке.

Создать Лицевую панель (рис. 3.9), установив на нее один элемент управления Numeric и один элемент индикации Numeric, три элемента управления String Control и один элемент индикации String Indicator из библиотеки String & Path (рис. 3.10).

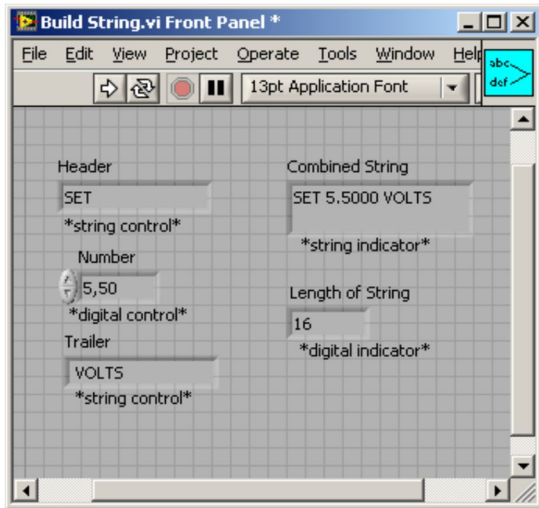


Рис. 3.9

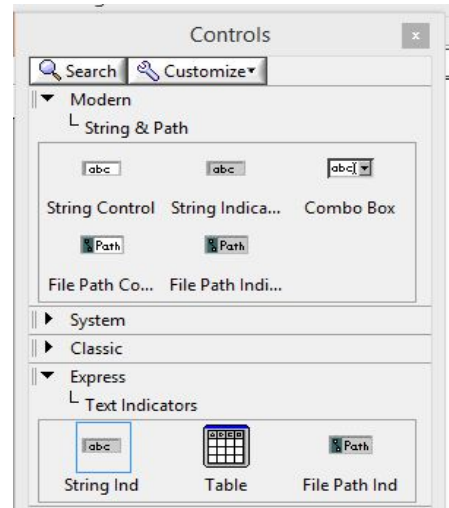


Рис. 3.10

В Блок-диаграмму (рис. 3.11) добавить из библиотеки функций String: Format Into String, Concatenate Strings, String Length (рис. 3.12).

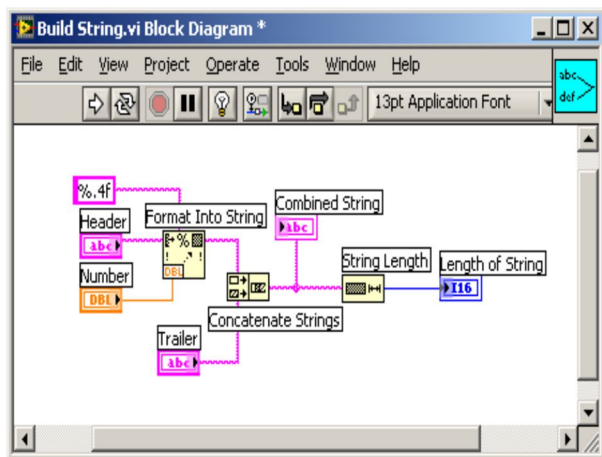


Рис. 3.11

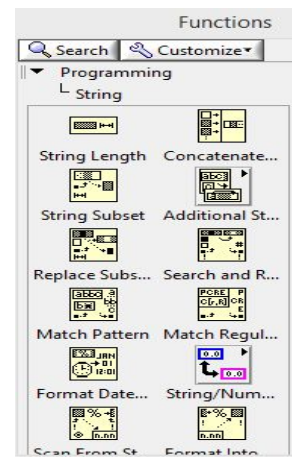


Рис. 3.12

Соединить проводами элементы и функции.

Перейти к лицевой панели и ввести текст «%.4f» в первый строковый элемент управления, в остальные строковые элементы управления ввести любой текст, ввести число в цифровой элемент управления.

Добавить пробелы в конце заголовка и начале строки завершения.

Запустить виртуальный прибор. Сохранить ВП, задав имя Build String.vi.

Задание 3.7. Анализ строк.

Создать ВП, анализирующий данные длиной строки путем выделения подстроки и преобразования числовых символов в этой подстроке в числовое значение.

Создать Лицевую панель (рис. 3.13) и Блок-диаграмму (рис. 3.14).

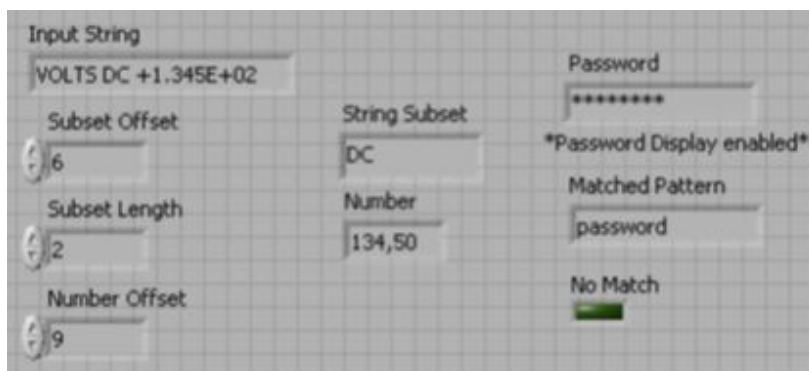


Рис. 3.13

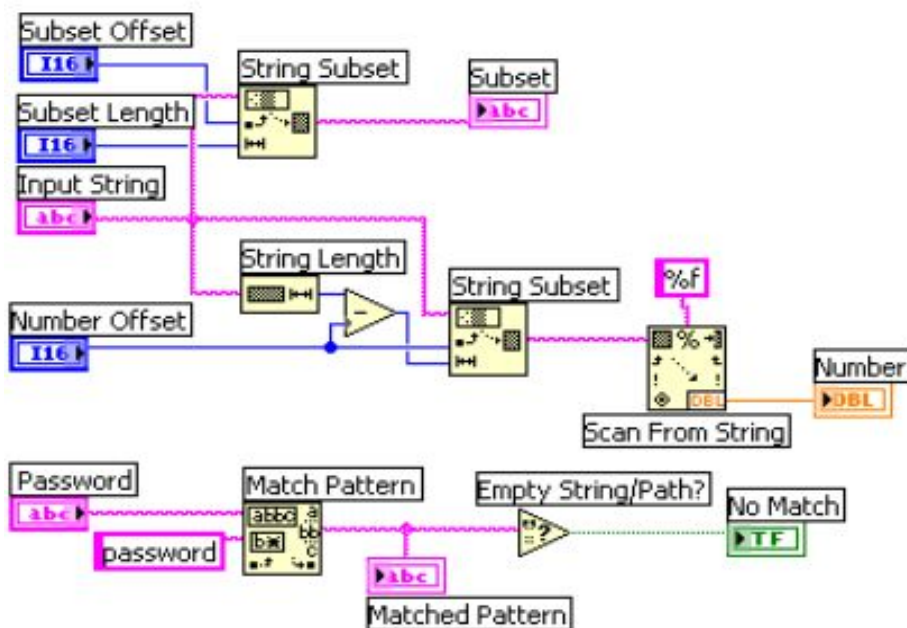


Рис. 3.14

Настроить строку Пароль (Password) так, чтобы она отображала лишь звездочки, путем выбора функции Скрытое отображение (Password Display) в контекстном меню.

Подключить функции, преобразующие и анализирующие строки:

- String Subset (Выделение подстроки) – возвращает из входной строки подстроку заданной длины, начиная с определенного символа;
 - просмотр строки – преобразует строку, содержащую действительные числовые символы (от 0 до 9, +, -, E и период) в число;
 - Match Pattern (Шаблон строки) – сравнивает строку пароля пользователя со строкой заданного шаблона (пароля). Если имеется совпадение, то строка проходит на выход, если же совпадений нет, то функция вернет пустую строку;
 - Empty String/Path? (Пустая строка/путь?) – возвращает логическое значение True, если входная строка или путь пустые, или False в противном случае;
 - String Length (Длина строки) – возвращает число символов в строке.
- Запустить виртуальный прибор с данными, показанными на рис. 3.13.

Числовая часть строки должна быть преобразована в число. Исследовать различные значения элементов управления (строки, как и массивы, индексируются, начиная с 0).

Функция Шаблон строки сравнивает входной пароль с паролем в строке (который в данном случае содержит символы, «password») и возвращает его, если он обнаружен. Если же совпадения не было, функция возвращает пустую строку.

Сохранить ВП под именем Password.vi.

Задание 3.8. Создание ВП «Массивы данных».

Создать ВП с одномерным входным массивом, умножающим пару заданных соседствующих чисел и выдающим на выходе значение произведения полученных данных массива. Для входного массива с элементами 1, 10, 3, 5, 7, 2 будет получен выходной массив с данными 10, 15, 14.

Установить на Лицевую панель один элемент управления Array и один элемент индикации Array из палитры Controls (рис. 3.15). Добавить в Array (управления и индикации) элемент Numeric (рис. 3.16) и растянуть протягиванием элемент Array до 6 ячеек (рис. 3.17).

Примечание. Массив может содержать в себе и другие типы данных. Но совокупность данных содержимого массива может быть только одного типа (рис. 3.17).

В итоге на Лицевой панели будут представлены два массива, один из которых является средством управления, а другой – средством отображения (рис. 3.18).



Рис. 3.15

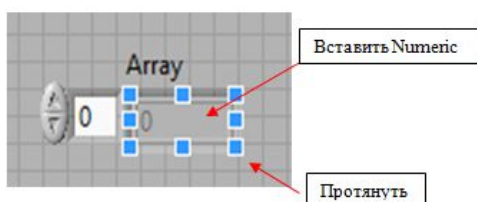


Рис. 3.16

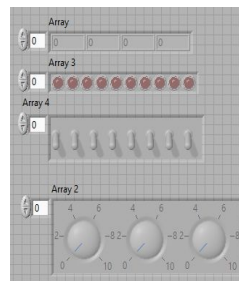


Рис. 3.17

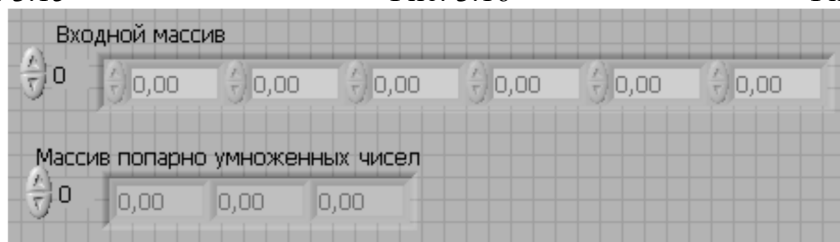


Рис. 3.18

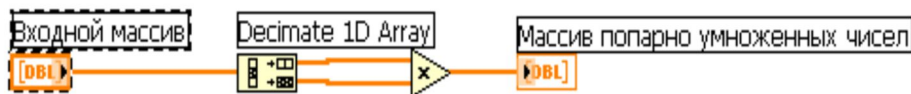


Рис. 3.19

Из палитры функций выбрать Decimate 1D Array и умножение и соединить проводами, как показано на рис. 3.19. Сохранить прибор.

Далее сохранить прибор под новым именем и исследовать другие арифметические операции.

4. СТРУКТУРЫ, ФУНКЦИИ ГЕНЕРАЦИИ И ОБРАБОТКИ ДАННЫХ

4.1. Структуры

Для выполнения операций над потоком данных в LabView с помощью различных структур задается алгоритм работы программы.

К таким структурам относятся:

- Formula Node (узел Формулы);
- MathScript Node (узел Математика);
- цикл For Loop (с фиксированным числом итераций);
- цикл While Loop (по условию);
- структура Case (Вариант);
- структура Stacked Sequence Structure (Стековая последовательность);
- структура Flat Sequence Structure (Открытая последовательность);
- структура Event (Событие);
- структура отключения диаграммы (Diagram Disable Structure);
- структура отключения по условию (Conditional Disable Structure);
- разделяемая переменная (Shared Variable);
- локальная переменная (Local);
- глобальная переменная (Global);
- узел Обратная связь (Feedback Node);
- группа Тактируемые структуры (Timed Structures).

В практической части раздела будут рассмотрены часто используемые, стандартные в программировании виды структур.

4.2. Функции генерации и обработки данных

Применяя различные виды осциллографов, имеющихся в среде графического программирования, которые являются разновидностью типов данных, можно визуализировать сигналы и представлять их в виде графиков функций.

Для установки на Лицевую панель их можно выбрать из палитры Controls Palette ► Modern ► Graph. К ним относятся Waveform Chart, Waveform Graph, XY Graph и другие виды, которые изучаются подробно на лекционных занятиях. В методическом пособии будут рассмотрены перечисленные виды. Каждый из них имеет свой набор характеристик, возможностей и применяется для представления разных видов сигналов. Также имеется палитра функций библиотеки генерации сигналов и шумов, которые могут представлять собой:

- генератор с заданной длительностью сигналов;
- гармонические колебания и шум;
- отрезки синусоидального, импульсного, пилообразного, $\sin(x)/y$, прямоугольного и частотно-модулированного сигналов;

- синусоидальные, треугольные, прямоугольные, пилообразные и произвольные колебания любой длительности;
- равномерный, гауссовский, периодический случайный шум и двоичную последовательность максимальной длины;

- гамма-шум;
- пуассоновский, биномиальный шум;
- шум Бернулли и т. п.

Аналоговые сигналы принимают множество значений, допускаемых их числовым значением Numeric: время, частота, координаты, амплитуда.

Цифровые сигналы принимают множество значений Boolean источники/приемники данных.

Практические задания

Задание 4.1. Создание простейшего ВП с применением структуры «Formula Node».

Загрузить программу. Создать новый виртуальный прибор.

Активировать панель Controls Palette.

Установить на Лицевую панель: 1 элемент управления и 1 элемент индикации Numeric.

В окне Блок-диаграммы (рис. 4.1) установить структуру «Formula Node», которую можно вызвать через контекстное меню нажатием правой клавиши мыши: Functions ► Programming ► Structures ► Formula Node.

Перетащить и растянуть рамку структуры до необходимого размера и в нее вписать логическую формулу с условием (if $x \geq 0$, then $y = \sqrt{x}$, else $y = -99$) в соответствии с рис. 4.1.

Каждая формула пишется в отдельной строке и заканчивается точкой с запятой. Неизвестные записываются в левой части формул.

Затем в формулы нужно внести исходные данные и вывести результаты расчета. Для этого курсор устанавливается правой клавишей мыши на рамке формульного узла и из контекстного меню левой клавишей активируются команды Add Input (добавить вход) для входных величин и Add Output (добавить выход) для выходных величин. В появившиеся рамки вписываются наименования этих величин в соответствии с рис. 4.1.

К входным рамкам проводами подключаются цифровые управляющие элементы, к выходным – индикаторы. Входы и выходы можно устанавливать в любом месте рамки. Наименования в рамках должны быть точно такими же, как в формульном узле. Допускается применение одного и того же наименования для входной и выходной величины. Вид формульного узла показан на рис. 4.1.

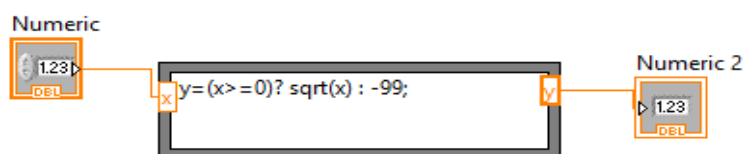


Рис. 4.1

Запустить прибор и проверить его работу. Сохранить файл.

При помощи управляющих элементов задать исходные данные, значения которых можно менять в работающем виртуальном приборе.

Задание 4.2. Создание простейшего ВП с применением «Expression Node».

Загрузить программу. Создать новый виртуальный прибор.

Активировать панель Controls Palette.

Установить на Лицевую панель: 1 элемент управления и 1 элемент индикации Numeric.

Переименовать названия элементов управления и индикации в соответствии с рис. 4.2.

В Блок-диаграмму (рис. 4.2) установить структуру «Expression Node», которая вызывается через контекстное меню: All Functions ► Structures ► Expression Node. Рамку растянуть и в нее вписать формулу вычисления функции: x^2+2x+5

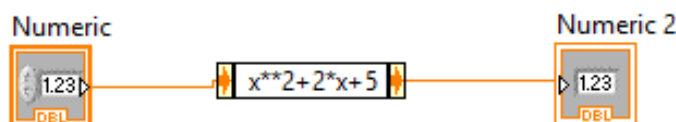


Рис. 4.2

Примечание. Запись « x^{**2} » означает « x в степени 2», запись « $2*x$ » означает « x умноженное на 2».

Запустить прибор и проверить его работу. Сохранить файл.

Задание 4.3. Создание ВП «График функции».

Загрузить программу.

Создать новый виртуальный прибор.

Активировать панель Controls Palette.

Установить на Лицевую панель: 1 элемент управления Numeric и 1 элемент индикации Waveform Chart. Waveform Chart выбрать из палитры: Controls Palette ► Modern ► Graph. Переименовать элементы в соответствии с рис. 4.3.

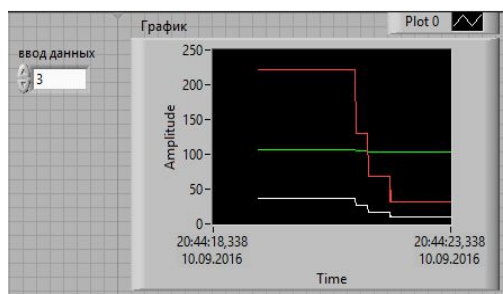


Рис.4.3

Настроить в Waveform Chart данные амплитуды в соответствии с рис. 4.4. Для этого надо щелкнуть на одном из значений шкалы и ввести новые данные. Можно также настроить данные через команду **Свойства**

элемента Waveform Chart, которые можно вызвать через контекстное меню, находясь в пределах графического элемента (рис. 4.4).

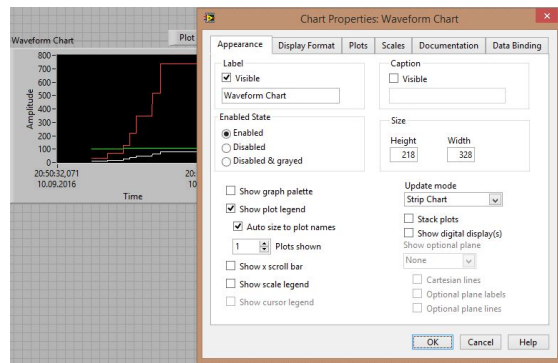



Рис. 4.4

В Блок-диаграмме установить структуру «Formula Node» и вписать формулу построения функции в соответствии с рис. 4.5.

Добавить входы и выходы, ввести в них данные.

Добавить элемент Merge Signal  из палитры Functions ► Express ► Sig Manip ► Signal Manipulation.

Можно добавить или убавить узлы протягиванием .

Соединить проводами выходные данные с узлами Merge Signal.

Соединить Merge Signal с Waveform Chart.

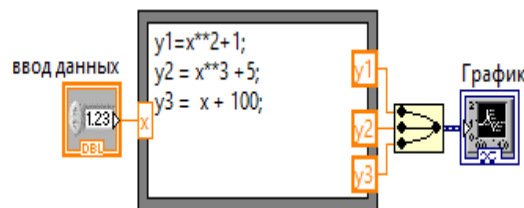


Рис.4.5

Запустить прибор и проверить его работу. Сохранить файл.

Задание 4.4. Создать ВП с применением структуры «Formula Node».

Создать ВП.

Установить на Лицевую панель: 5 элементов управления и 2 элемента индикации Numeric.

Переименовать все элементы управления и индикации.

Установить в Блок-диаграмме структуру Formula Node и внести формулы для построения графиков функций в соответствии с рис. 4.6, 4.7:

$$y1 = x3 + x2 + 5;$$

$$y2 = m * x + b;$$

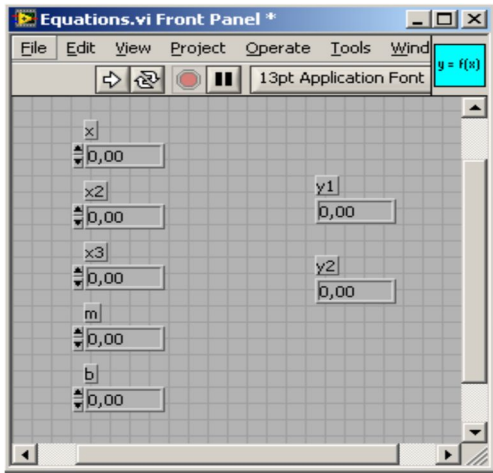


Рис. 4.6

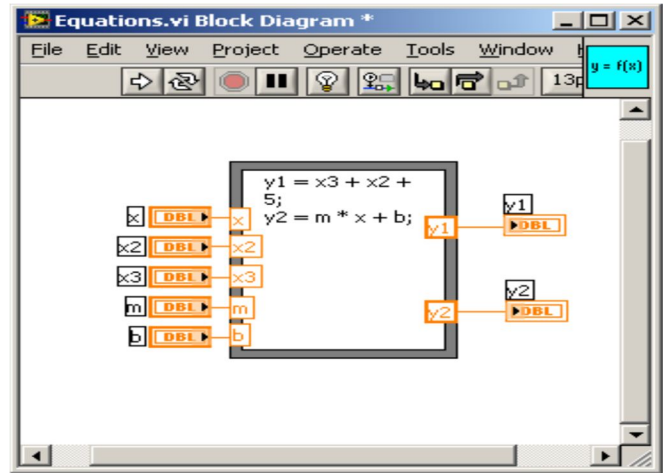


Рис. 4.7

Запустить прибор и проверить его работу. Сохранить файл.

Задание 4.5. Создать ВП «Определение токов в цепи» с использованием структуры «Formula Node».

Задача. Создать ВП «Определение токов в цепи».

Токи в цепи можно рассчитать по законам Ома и Кирхгофа:

$$I_1 = E / (R_1 + (R_2 * R_3 / (R_2 + R_3)));$$

$$I_2 = R_3 * I_1 / (R_2 + R_3);$$

$$I_3 = R_2 * I_1 / (R_2 + R_3);$$

Значения сопротивлений R_1 , R_2 , R_3 и электродвижущей силы E произвольные.

Пример А

Расположить на Лицевой панели: 4 графических элемента управления Numeric и 3 элемента индикации Numeric. Переименовать их названия в соответствии с рис. 4.8.

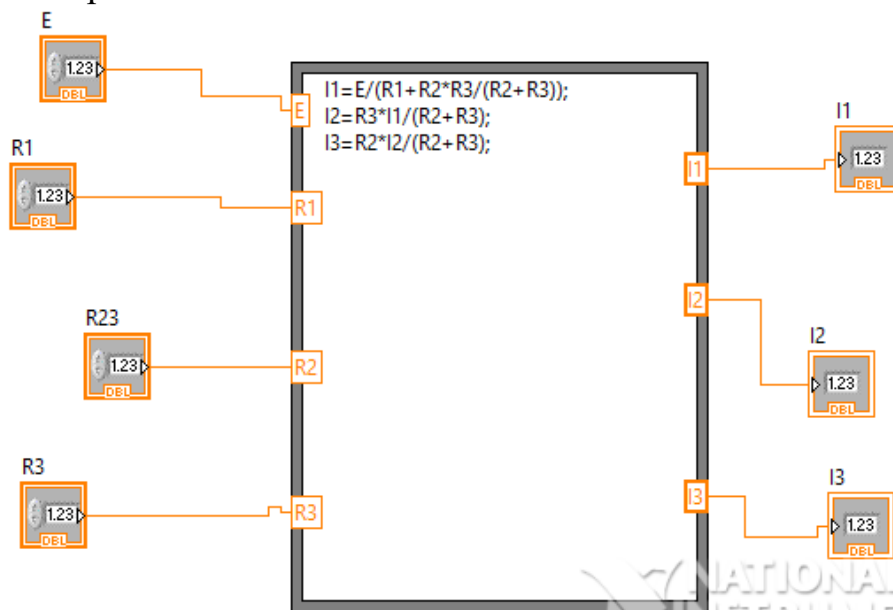


Рис. 4.8

Установить в Блок-диаграмме структуру «Formula Node» и внести формулы в соответствии с рис. 4.8.

Запустить прибор и проверить его работу. Сохранить файл.

Пример Б

Сохранить предыдущий виртуальный прибор под другим именем.

Внести изменения на Лицевой панели, добавив элемент Waveform Chart и удалив индикаторы Numeric. В Блок-диаграмме также внести изменения в соответствии с рис. 4.9. Вывести данные на Waveform Chart.

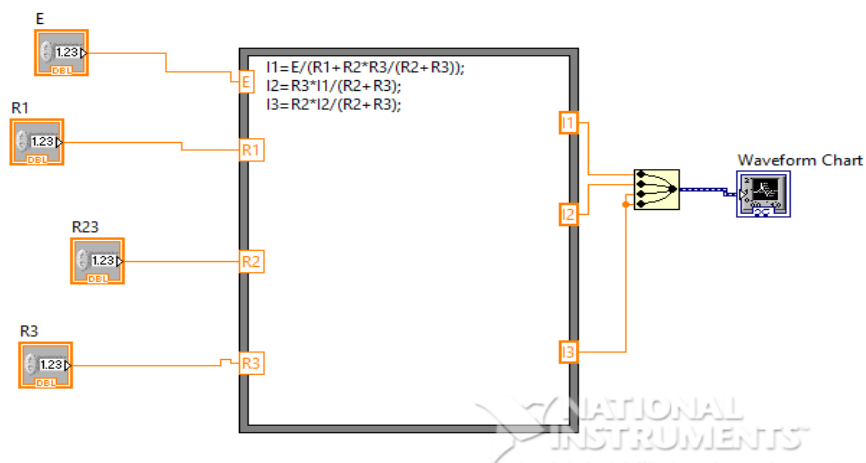


Рис. 4.9

Запустить прибор и проверить его работу. Сохранить файл.

Задание 4.6. Создание ВП «График функции» с применением структур «Formula Node» и «For Loop».

Создать ВП, который использует узел Формула для построения графика функции $y_1 = (\sin(x) + \cos(x))/10$ и $y_2 = \cos(x)$ в соответствии с рис. 4.10, 4.11.

Создать ВП.

Установить на Лицевую панель: 1 элемент Numeric и 1 элемент Waveform Graph.

Переименовать его в График функции.

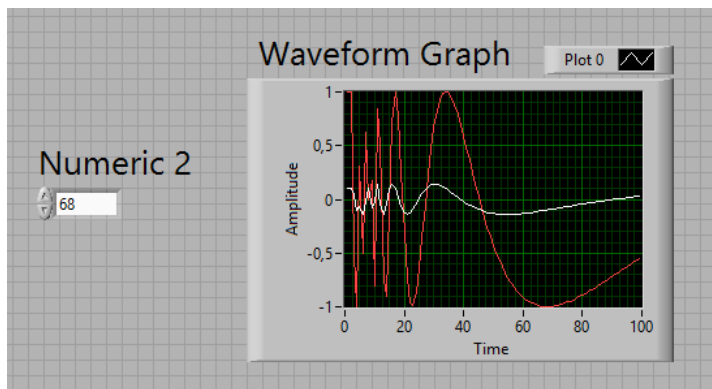


Рис. 4.10

Построить Блок-диаграмму, изображенную на рис. 4.10. В Блок-диаграмме установить структуру «For Loop», вызвав ее через меню: Functions ► Programming ► Structures ► For Loop – и в нее вставить структуру «Formula Node» и записать формулу. Создать тоннели входа и выхода на границах структуры.

Все остальные элементы и операции вставить в соответствии с рис. 4.11.

Объединить все элементы и операции проводами.

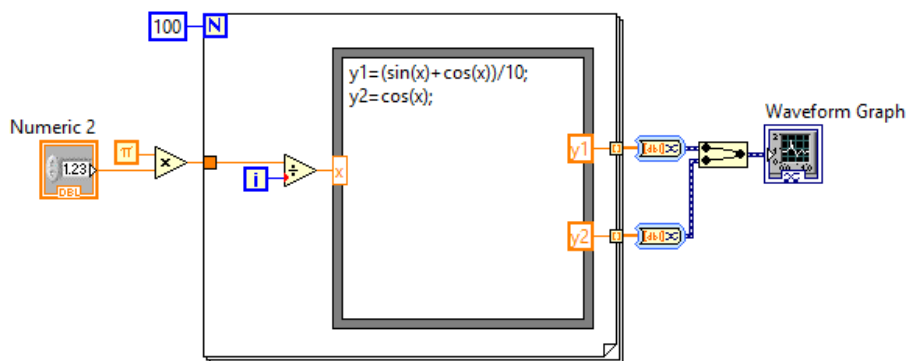


Рис. 4.11

Проверить работу и сохранить.

Задание 4.7. Создание ВП «График функции» с применением структур «Formula Node» и «For Loop».

Создать ВП в соответствии с рис. 4.12 и 4.13.

Установить на Лицевую панель элемент Waveform Chart.

В Блок-диаграмму установить структуру For Loop, в нее вставить структуру Formula Node и создать формулу (рис. 4.13).

Создать тоннели на границах структур.

Все остальные элементы и операции вставить в соответствии с рис. 4.13.

Объединить все элементы и операции проводами.

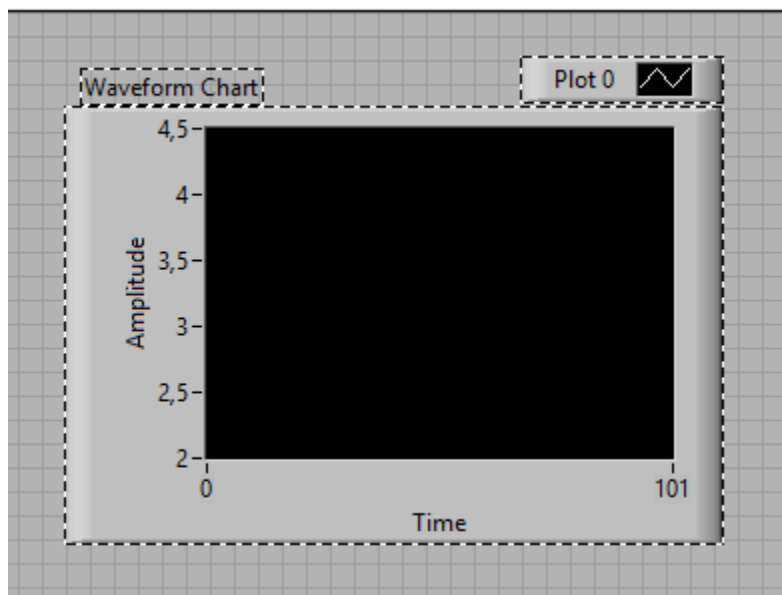


Рис. 4.12

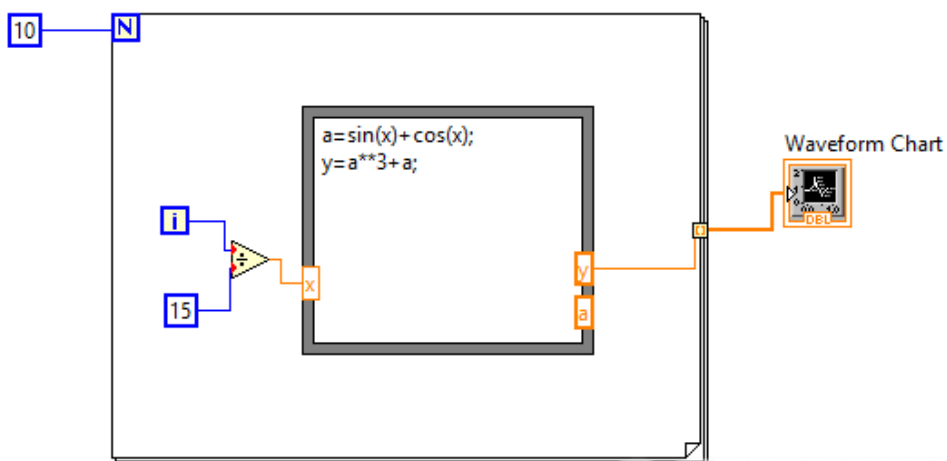


Рис. 4.13

Проверить работу и сохранить.

Задание 4.8. Создание осциллографа с применением структуры «While Loop» и выполнение цикла по условию.

Создать ВП в соответствии с рис. 4.14, 4.15.

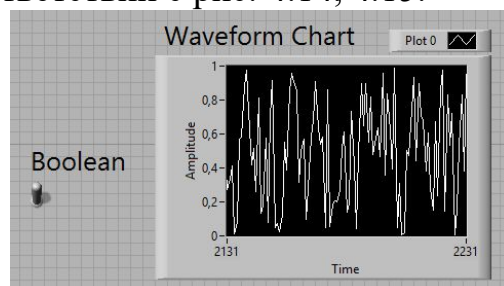


Рис. 4.14

Установить на Лицевую панель:

- 1 элемент индикации Waveform Chart, переименовать его в «График функции»;
- 1 элемент управления Boolean.

В Блок-диаграмме:

- создать структуру «While Loop»;
- установить элемент функции случайное число «Random Number», рис. 4.15;
- объединить все элементы и операции проводами.

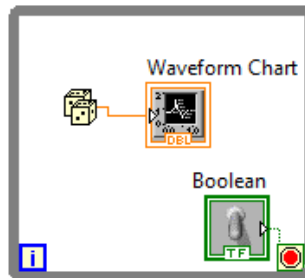


Рис. 4.15

Для изменения значений «Random Number» (случайного числа), используя свойства этого элемента (контекстное меню), добавить варианты элементов, представленных в табл. 4.1.

Таблица 4.1

Далее исследовать все возможные элементы из панели Numeric Palette (рис. 4.16), вызываемые из контекстного меню элемента «Random Number».

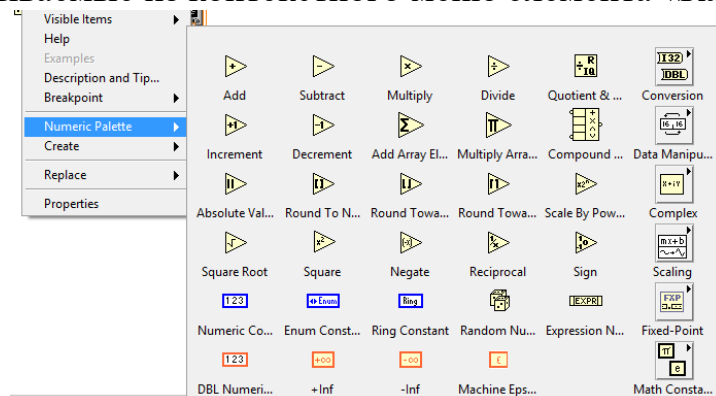


Рис. 4.16

Задание 4.9. Создание ВП «Извлечение квадратного корня» с применением структуры «Case Structure».

Создать ВП, предварительно установив на Лицевую панель:

- 1 элемент управления;
- 1 элемент индикации Numeric, переименовав его в «Square Root Value», рис. 4.17.

Перейти в Блок-диаграмму и вставить структуру Case Structure.

В каждом из состояний False и True установить элементы в соответствии с рис. 4.17–4.19.

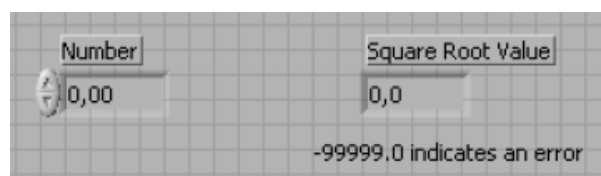


Рис. 4.17

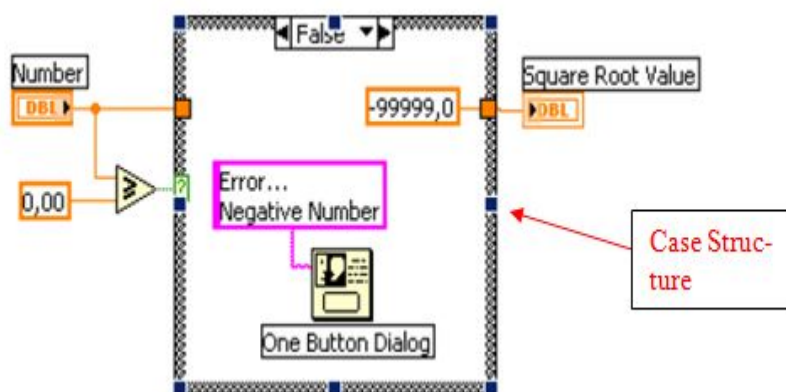



Рис. 4.18

Добавить элемент функции Stop , находящейся в палитре функций Programming ► Application Control, и соединить его с диалоговым элементом. Элемент Stop необходим для завершения процесса работы ВП.

Тоннели входа и выхода создаются автоматически при соединении проводником.

Примечание. Если один из тоннелей будет представлен незакрашенным, то это значит, что в структуре имеются ошибки.

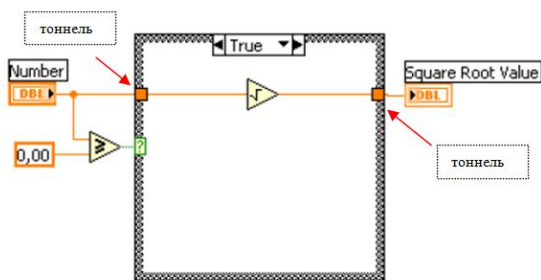


Рис. 4.19

При вводе отрицательного числа должно появиться сообщение об ошибке. Проверить работу прибора и сохранить.

Задание 4.10. Создание ВП, моделирующего температурный монитор, с подключением виртуального подприбора (ВПП).

Для выполнения задания требуется:

- источник напряжения (использовать готовый ВПП Demo Voltage Read.vi);
- создать ВП Термометр и преобразовать его в ВПП;
- создать ВП Температурного монитора.

А. Создание ВП «Термометр» и подключение ВПП «Источник напряжения».

1. Установить на Лицевую панель элемент Термометр (Controls ► Numeric) (рис. 4.20).

2. Изменить название и данные шкалы от 70 до 90 (свойство Scale). Построить Блок-диаграмму, изображенную на рис. 4.21.

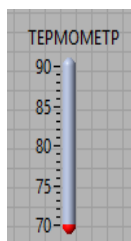


Рис. 4.20

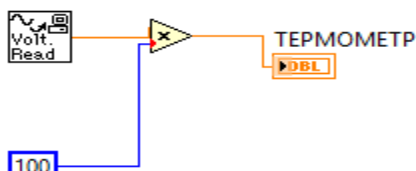


Рис. 4.21

Выполнить следующие условия:

- перетащить на Блок-схему готовый ВПП измерения напряжения Demo Voltage Read.vi. (взять у преподавателя);
- установить элемент умножения и элемент Numeric Constant (добавить в поле 100);
- соединить все элементы.

Посмотреть работу и Блок-схему работы ВПП измерения напряжения Demo Voltage Read.vi.

Проверить работу ВП.

Б. Операция «Назначение терминала элементу управления или индикации».

Для того чтобы ВП использовать в качестве ВПП, передавать и извлекать из него данные, необходимо выполнить операции:

- находясь на ЛП, где расположен элемент Термометр, щелкнуть правой кнопкой мыши на пиктограмме в верхнем левом углу (рис. 4.22);

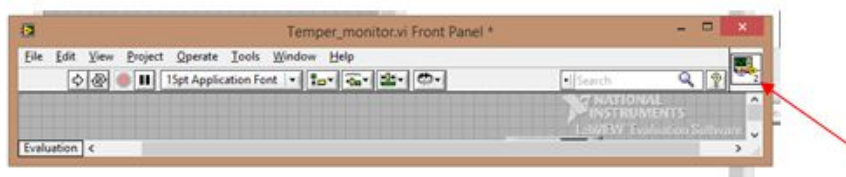


Рис. 4.22

- выбрать Edit Icon, отредактировать внешний вид иконки ВПП, затем выбрать Show Connector (рис. 4.23);

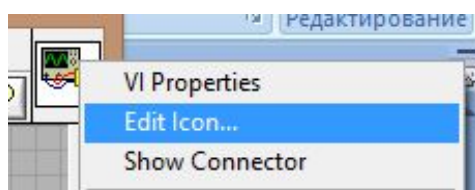


Рис. 4.23

- выбрать Patterns и первый образец (рис. 4.24);

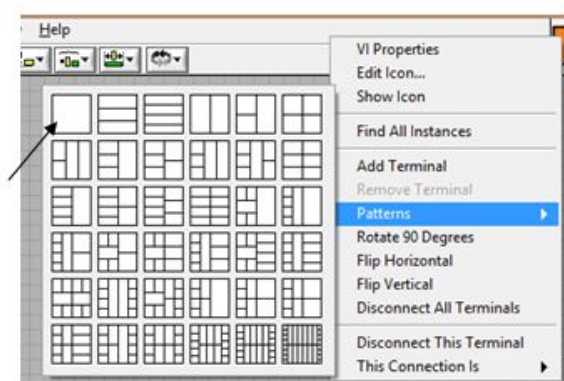


Рис. 4.24

- находясь на ЛП, щелкнуть курсором по терминалу соединительной панели (появится «катушка»);

- щелкнуть по элементу управления Термометр;

- щелкнуть правой клавишей мыши по открытой области ЛП (пунктирная линия исчезнет, а терминал окрасится в цвет, соответствующий типу назначенного элемента);

- Сохранить ВПП (save As..Termometr1).

В. Создание ВП «Температурный монитор».

Поместить на ЛП элементы в соответствии с рис. 4.25.

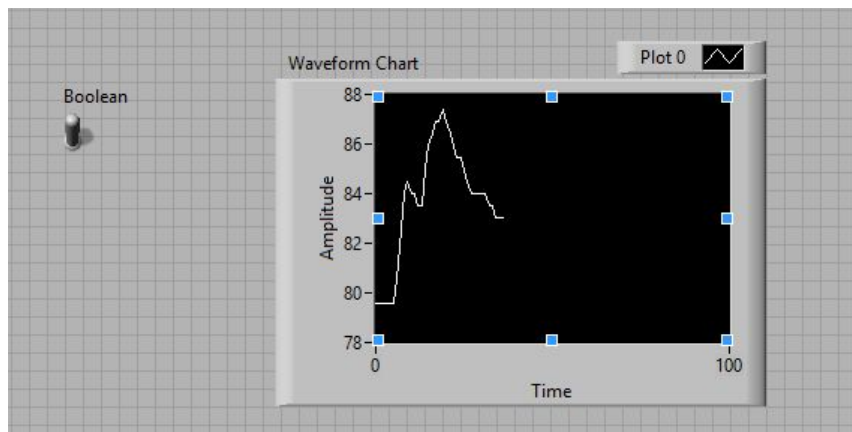


Рис. 4.25

Поместить на Блок-схему ВП структуру While Loop, элемент Numeric Constant (500), функцию таймер Timing.
 Перетащить на блок-схему ВПП «Термометр».
 Соединить элементы в соответствии с рис. .26

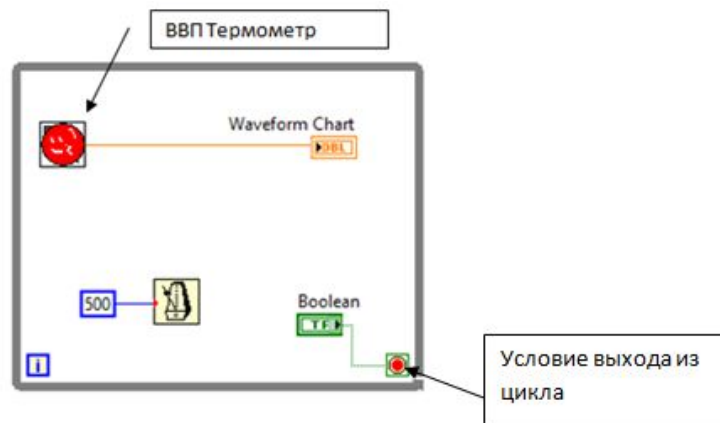


Рис. 4.26

Запустить ВП на выполнение. На графике должен отобразиться процесс изменения температуры.

Задание 4.11. Создание ВП «Регистр» с применением структуры «While Loop».

Создать ВП в соответствии с рис. 4.27, 4.28

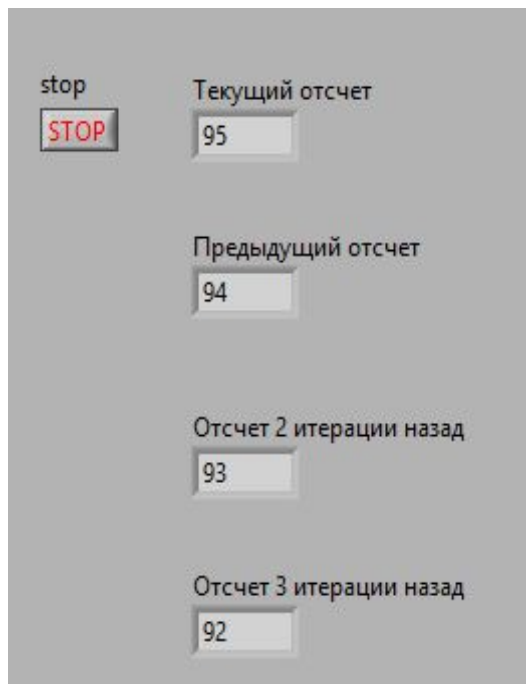


Рис. 4.27

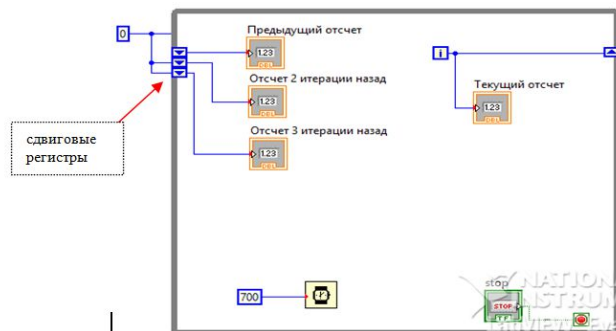


Рис. 4.28

Добавить сдвиговой регистр через контекстное меню Add Shift Register. Добавить к нему элементы Add Element. Соединить с индикаторами. Сохранить ВП.

Задание 4.12. Создать ВП измерения температуры.

Создать ВП в соответствии с рис. 4.29 и 4.30, который производит 50 измерений температуры внутри цикла (одно измерение через каждые 0,25 с) и строит их на развертке осциллограммы. Также он преобразовывает каждое значение в строку, а затем объединяет эту строку с символом табуляции, отсчетом времени и символом конца строки, после чего записывает все полученные данные в файл.

Можно записать данные в файл в виде одной строки за каждую итерацию, но лучше собрать все данные в одной большой строке, используя сдвиговые регистры и функцию Concatenate Strings (Объединение строк), а затем сохранить их в файл. Данные в текстовом редакторе будут выглядеть следующим образом:

78.9 11:34:38

79.0 11:34:39

79.0 11:34:50

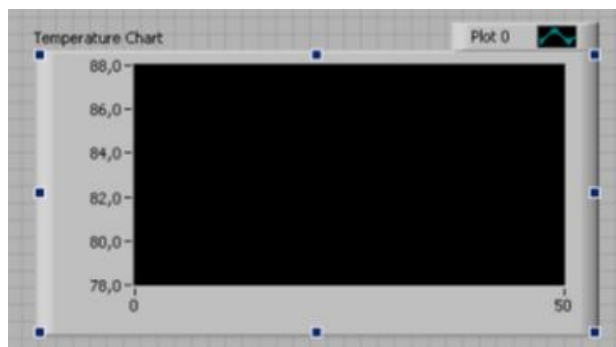


Рис. 4.29

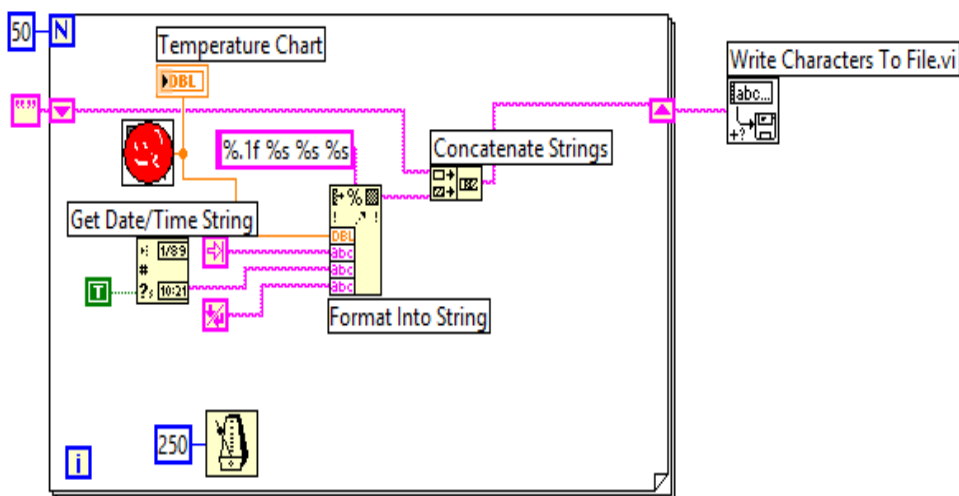


Рис. 4.30

В Блок-диаграмме нужно подключить ВПП (Termometr1), созданный в задании 4.10.

На рис. 4.31, 4.32, 4.33 представлены функциональные элементы, которые нужно подключить к функции Format Into String.

End of Line Constant



Рис. 4.31

Tab Constant



Рис. 4.32

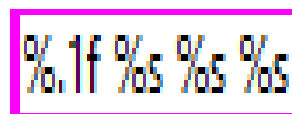


Рис. 4.33

Запустить ВП на выполнение. Проверить работу и сохранить.

Задание 4.13. Построение ВП «График функции» с применением массива данных.

Создать ВП, который будет генерировать массив данных синусоидальной кривой и вычерчивать его на графике осциллограммы Waveform Graph (Графика осциллограммы).

Открыть новый ВП и установить на Лицевую панель элементы в соответствии с рис. 4.34.

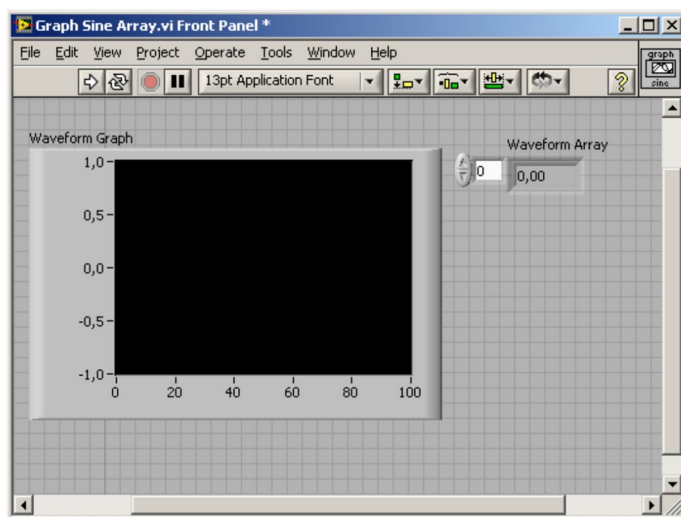


Рис. 4.34

Установить структуру, функции, элементы в Блок-диаграмму в соответствии с рис. 4.35.

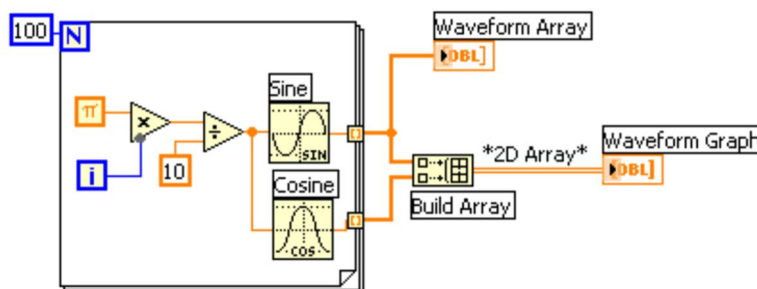


Рис. 4.35

Спрятать панель редактирования графика, вызвав его контекстное меню и изменив опцию Visible Items (Видимые элементы) ► Plot Legend (Панель редактирования графика).

Отключить автоматическую установку масштаба, щелкнув правой кнопкой мыши и выбрав опцию Y Scale (Шкала Y) ► AutoScale Y (Автомасштабирование Y). Изменить пределы отображения оси Y путем их выделения с помощью инструмента ввода текста и указания новых чисел: задать минимальное значение оси Y равным $-1,0$, максимальное – равным $1,0$.

Запустить ВП. Результатом работы будет отображение двух кривых. Сохранить ВП как Graph Sine Array.vi. Данный прибор будет использован в другом задании.

Переставить элемент array с левой стороны цикла и подключить его через тоннель, напрямую к элементу умножения, предварительно удалив элемент π . После запуска ВП отключить его. Вместо элемента π вставить элемент ε и проверить работу прибора и т. д.

Переделать ВП, как показано на рис. 4.36.

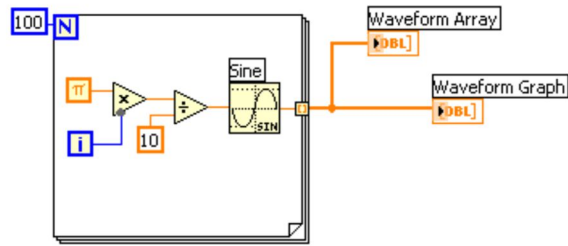


Рис. 4.36

В этом случае наблюдаем одну кривую, так как на вход Waveform Graph (Графика осциллограммы) подается одномерный, а не двумерный массив данных.

Еще раз переделать ВП, как показано на рис. 4.37.

Обратить внимание, что теперь на вход Waveform Graph (График осциллограммы) подается кластер, содержащий начальное значение X, приращение X и массив данных. Найти через поиск функций Bundle (Объединить) и подключить элементы в соответствии с рис. 4.37.

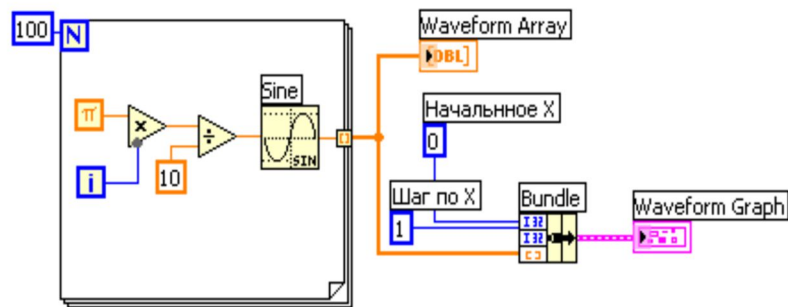


Рис. 4.37

Изменить начальное значение X на 0,5 и начальное значение шага X на 20, а затем снова запустить ВП.

Обратить внимание, что график должен будет показать те же 100 точек с начальным значением 20 и начальным значением X, равным 0,5 (показано на оси X). То есть график не изменится, изменится только ось X. Сохранить как Graph Sine Array_1.vi.

Задание 4.14. Построение графиков с использованием XY Graph (XY Графика).

Создать ВП в соответствии с рис. 4.38, где окружность начерчена на основе данных зависимых массивов X и Y.

Открыть ВП GraphSineArray.vi, сохраненный ранее. Переименовать его в GraphSineArray_2.vi

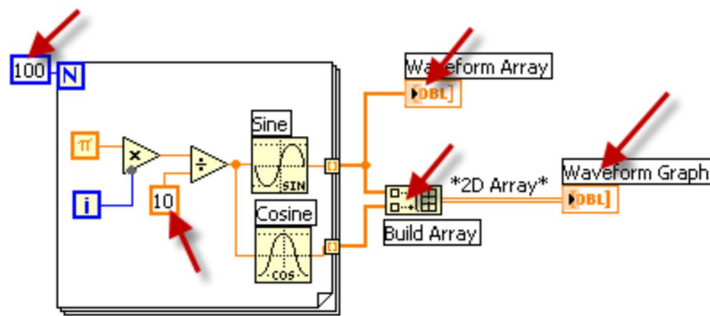


Рис. 4.38

Заменить в нем указанные элементы, чтобы получилась Блок-диаграмма, изображенная на рис. 4.39.

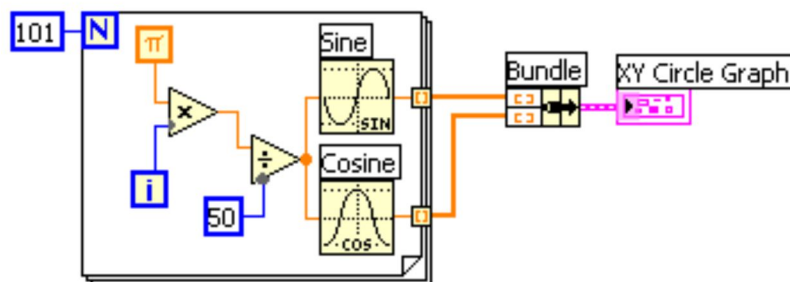


Рис. 4.39

При этом Waveform Graph (График осциллограммы) заменяется на XY Graph (XY График). Используя функцию Bundle (Объединить), вы можете построить массив синуса относительно массива косинуса на двухкоординатном графике, который отображает окружность (рис. 4.40).

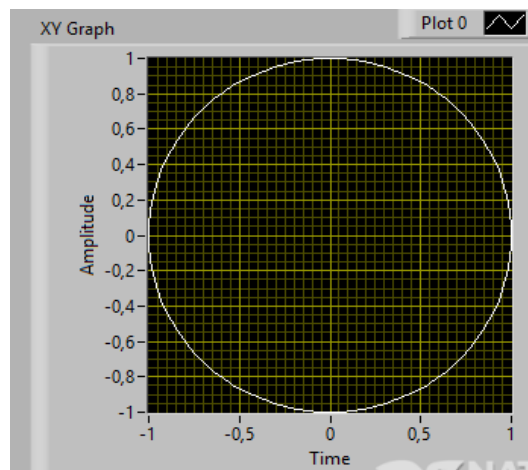


Рис. 4.40

Запустить виртуальный прибор. При желании сохранить его как Graph Circle.vi.

Далее исследовать изменения в графике:

- при значении функции π изменить значение константы на 1, 2, 3, 4, 5, 9, 60, 70, 90;

- при значении функции 2π изменить значение константы на 1, 2, 3, 4, 5, 9, 90, 150;
- при значении функции $\pi/2$ изменить значение константы на 1, 2, 3, 4, 5, 9, 10, 30, 80;
- при значении функции $\ln\pi$ изменить значение константы на 1, -1, 2, 10;
- при значении функции $\ln 10$ изменить значение константы на 1, -1, 2, -2, 4, 10, -10, 60 и т. д.

Задание 4.15. Создание ВП с применением структуры «For Loop».

Создать ВП в соответствии с рис. 4.41. Исследовать порядок потока данных и принцип работы прибора, используя пиктограмму лампочка.

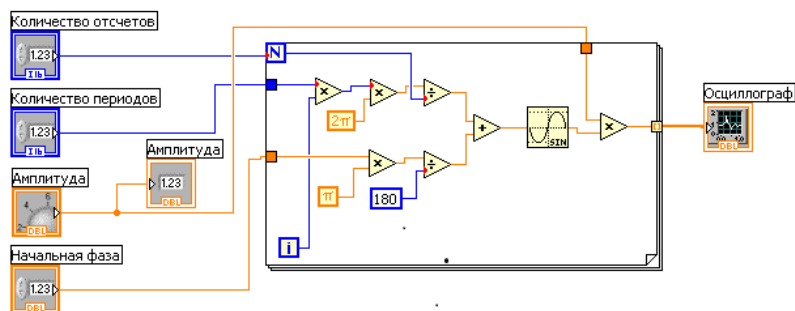


Рис. 4.41

Проверить работу. Сохранить ВП.

Задание 4.16. Запись данных сигналов в таблицу.

Создать (изменить имеющийся) ВП для сохранения данных в новый файл формата ASCII. Можно редактировать этот файл из любого приложения, работающего с текстами или таблицами.

Открыть ВП Graph Sine Array.vi, созданный ранее. Этот ВП генерирует два массива данных и строит их на графике. Необходимо изменить ВП для записи двух массивов в файл, в котором каждый столбец содержит массив данных.

Открыть Блок-диаграмму Graph Sine Array.vi и изменить виртуальный прибор путем добавления элементов, как показано на рис. 4.42

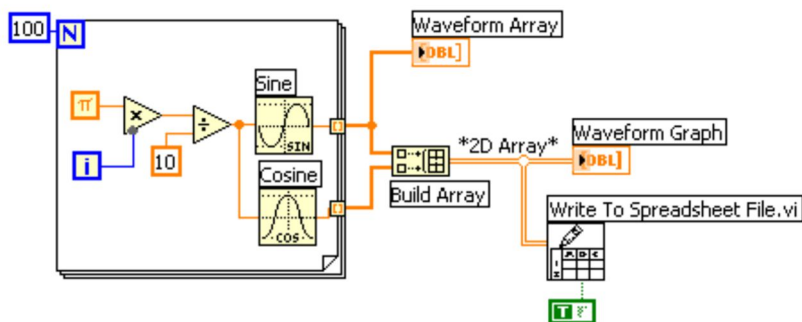


Рис. 4.42

Функция Write To Spreadsheet File (Записать в файл табличного формата, из палитры Programming ►File I/O) преобразует двумерный массив в строку таблицы и записывает ее в файл. Если путь к файлу не установлен, то во время выполнения появится диалоговое окно, где вам предложат выбрать имя файла.

Логическая константа (Boolean Constant) управляет транспонированием двумерного массива перед его записью в файл. Понадобится транспонировать данные, так как массивы данных определяются строками (каждая строка двумерного массива представляет собой массив данных). Так как желательно, чтобы каждый столбец таблицы в файле содержал данные для одной осциллограммы, то двумерный массив сначала должен быть транспонирован.

Вернуться к лицевой панели и запустить ВП. После того как массивы данных будут созданы, появится диалоговое окно, в котором нужно задать имя нового файла, в котором отображены записанные данные (например, Текст_1). В следующем задании данные будут считываться из сохраненного файла.

Сохранить ВП под именем Graph Sine Array_3.vi.

Примечание. Функция Write To Spreadsheet File в разных версиях программы Lab-View может выглядеть и называться иначе.

Задание 4.17. Считывание из файла.

Создать ВП для считывания данных и построения их на графике. Данные будут считываться из файла Текст_1, сохраненного в задании 4.16.

Открыть новый ВП и поместить Wave Form Graph (График осциллограммы) на его Лицевую панель (рис. 4.43). Должна быть включена автоматическая настройка шкалы.

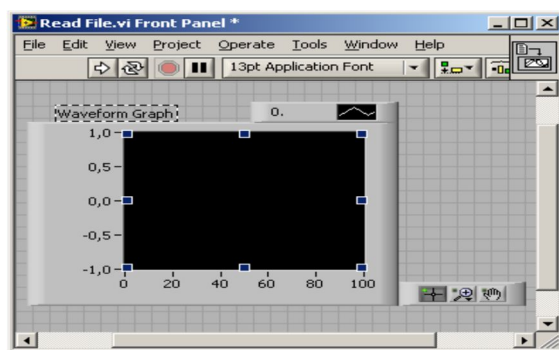


Рис. 4.43

Построить Блок-диаграмму, как показано на рис. 4.44. Применить функцию Read From Spreadsheet File (Считать из файла табличного формата) для ввода данных в прибор и отображения их на графике.

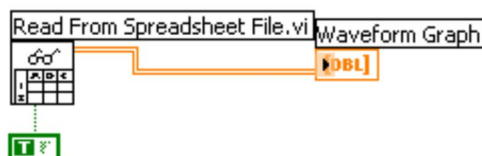


Рис. 4.44

Используя значение True логической константы, транспонировать массив после его считывания. Это необходимо, поскольку графики строят данные построчно, а они были сохранены в файле столбцами.

Запустить ВП. Выбрать файл, созданный при выполнении задания 4.16. ВП считывает данные из файла и построит обе осциллограммы на графике.

Сохранить виртуальный прибор, задав имя Read File.vi.

Задание 4.18. Полиморфизм на примере массивов.

Создать ВП в соответствии с рис. 4.45, который демонстрирует полиморфизм на примере массивов.

Затем осциллограф растянуть в ширину видимой части экрана. Массивы переместить ниже. Отредактировать расположение элементов массивов с помощью пиктограммы, расположенной в направлении красной стрелки (рис. 4.45).

Аналогично отредактировать расположение элементов массивов и осциллографа в окне Блок-диаграммы.

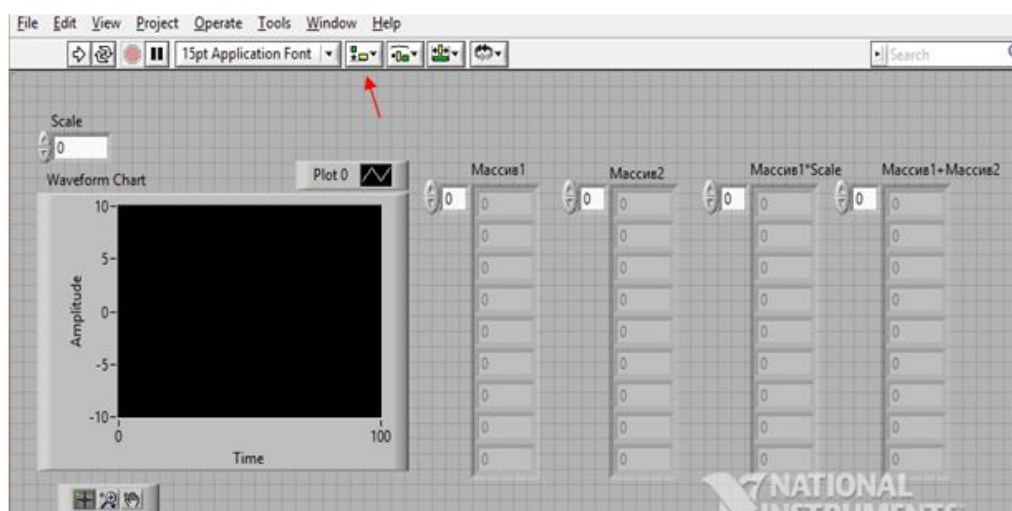


Рис. 4.45

Создать Блок-диаграмму в соответствии с рис. 4.46.

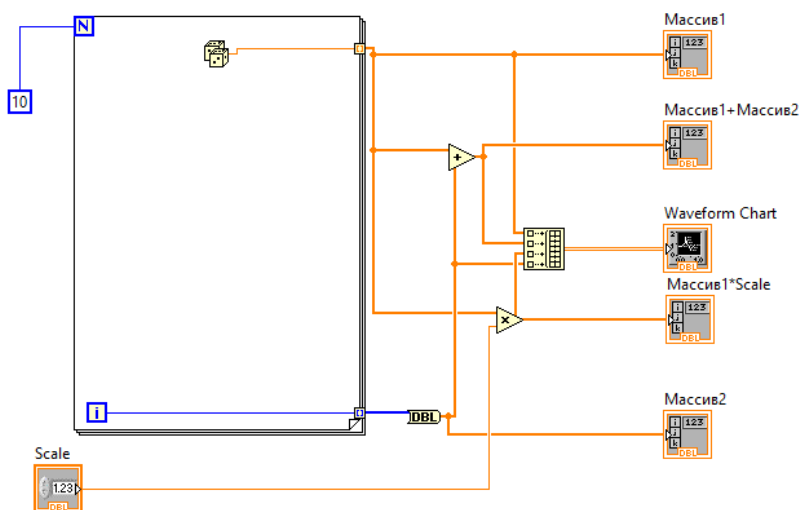


Рис. 4.46

Функцию **DBL** можно найти и выбрать через контекстное меню, установив курсор мыши на проводнике.

Исправить кривые провода с помощью команды контекстного меню Clean Up Wire, предварительно установив курсор на провод (рис. 4.47).

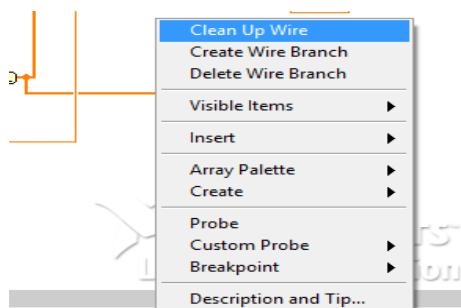


Рис. 4.47

Результат работы ВП показан на рис. 4.48.

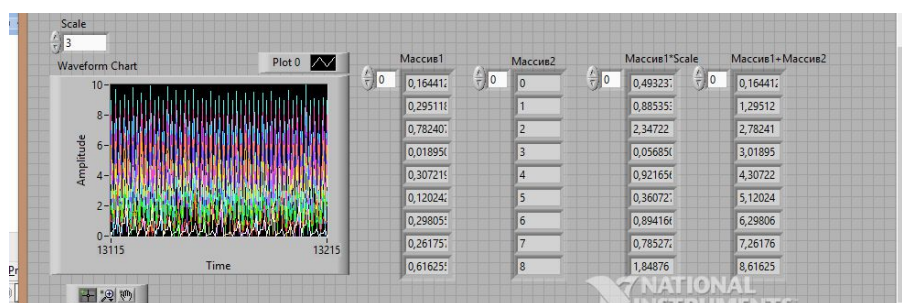


Рис. 4.48

Добавить в блок-схему функцию записи в файл данных осциллографа по образцу предыдущих работ (рис. 4.49) и логическую функцию True для подключения к элементу записи файла.

Запустить ВП, сохранить полученный файл записи данных. Выполнить эту операцию несколько раз, наблюдая за изменением данных. Каждый такт работы ВП должен быть сохранен в отдельном файле.

Заменить в Блок-диаграмме функцию True на False. Проверить работу ВП.

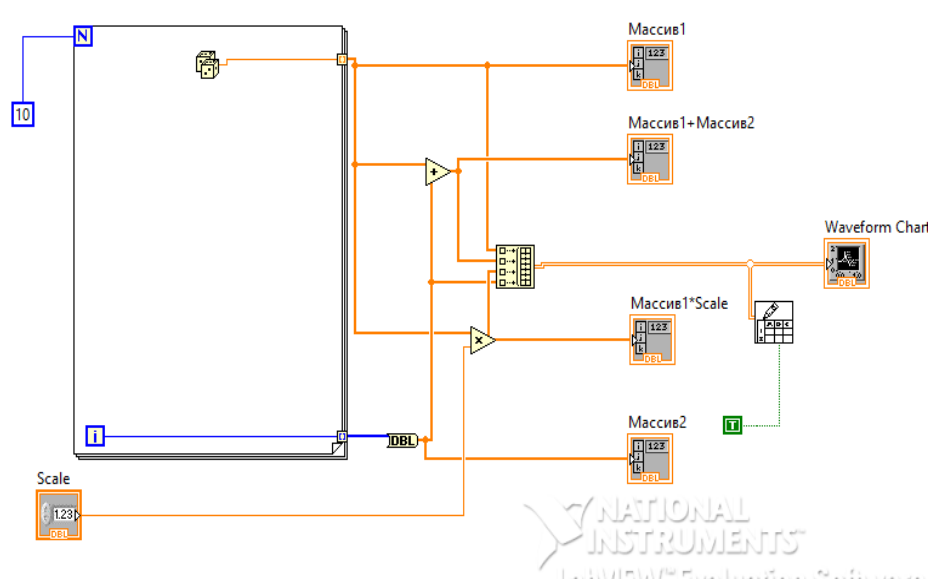


Рис. 4.49

Сохранить этот же файл под новым именем. Добавить функцию чтения данных.

Задание 4.19. Создание ВП «Кластеры данных».

Создать ВП, содержащий элементы, относящиеся к типу данных «кластер». В рассмотренном примере один из элементов будет содержать данные в соответствии с рис. 4.50, которые должны быть отображены в другом. После установки всех элементов на Лицевой панели установить все необходимые функции и связи в соответствии с рис. 4.51.

Открыть новую панель и поместить в нее шаблон Кластер (палитра Массив и кластер). Задать имя Input cluster (Входной кластер). Увеличить шаблон (убедиться, что вы захватили границу кластера, в противном случае ничего не произойдет).

Поместить внутрь шаблона Входной кластер цифровой элемент управления, два логических переключателя и элемент управления строками.

Создать шаблон Output cluster (Выходной кластер) путем копирования входного. Щелкнуть правой кнопкой мыши по объекту в новом кластере (или на границе кластера) и выбрать опцию Заменить на индикатор (Change to Indicator) и сменить ярлык кластера.

Примечание. Эта методика позволяет получить правильный порядок в кластере. Можно создать выходной кластер таким же способом, как и входной, используя элементы отображения вместо элементов управления (необходимо следить за порядком размещения элементов в кластере: он должен совпадать с порядком элементов входного кластера).

Убедиться, что элементы входного и выходного кластеров имеют одинаковый порядок. Это можно сделать, вызвав контекстное меню на

границе каждого кластера и выбрав опцию Изменить порядок элементов кластера.

Если порядки элементов кластеров разные, то сделать один из них аналогичным другому.

Поместить на Лицевой панели прямоугольную кнопку Стоп (Stop Button) из палитры Логические. Обратит внимание, что по умолчанию состояние этой кнопки – False. Это состояние оставить без изменений.

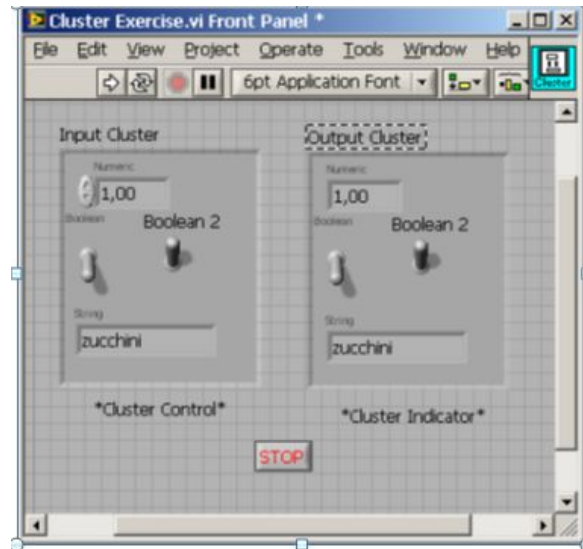


Рис. 4.50

Построить Блок-диаграмму, как показано на рис. 4.51.

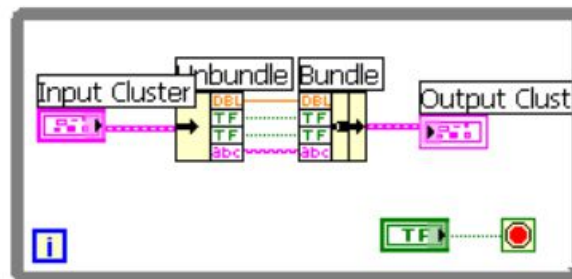


Рис. 4.51

Каждый кластер содержит четыре объекта, но на Блок-диаграмме отображается только по одному терминалу на кластер.

Изменить функцию терминала Условия выхода из цикла по условию так, чтобы она была остановлена, если Истина (это можно сделать вызовом контекстного меню терминала Условия выхода).

Функция Разделить разделяет кластер. После этого происходит доступ к отдельным элементам кластера. Изменить ее размер таким образом, чтобы она имела четыре выхода. При подключении к функции Разделить входной кластер в ее терминалах появятся символы типов данных.

Функция Объединить воссоздает кластер. Изменить ее размер, чтобы она имела четыре входа.

Доступ к функциям Объединить и Разделить можно получить, вызвав контекстное меню на терминале кластера, к которому нужно подключиться,

и выбрав соответствующую функцию в меню Кластерные инструменты (Cluster Tools). Эта функция будет содержать правильное количество вводов и выводов.

Вернуться к лицевой панели и запустить ВП. Ввести различные значения элементов кластера управления. На кластере индикации должны появиться значения, введенные в кластер управления.

Сохранить ВП.

Задание 4.20. Создание ВП «Анализ данных».

В задании будет использоваться ВВП «Термометр», созданный ранее.

Создать ВП в соответствии с рис. 4.52, вставив элементы Waveform Chart и Waveform Graph. Изменить названия, данные шкалы измерений, тип точки и цвет.

ВП должен измерять температуру через каждые 0,25 с в течение 10 с. Во время работы ВП демонстрирует результаты измерений в реальном времени на развертке осциллограммы. После завершения сбора данных ВП показывает данные на графике и вычисляет min, max и среднее значение температуры.

Развертка Температура должна отображать температуру по мере получения данных. После завершения сбора информации о температуре ВП должен вычерчивать данные на Графике температуры.

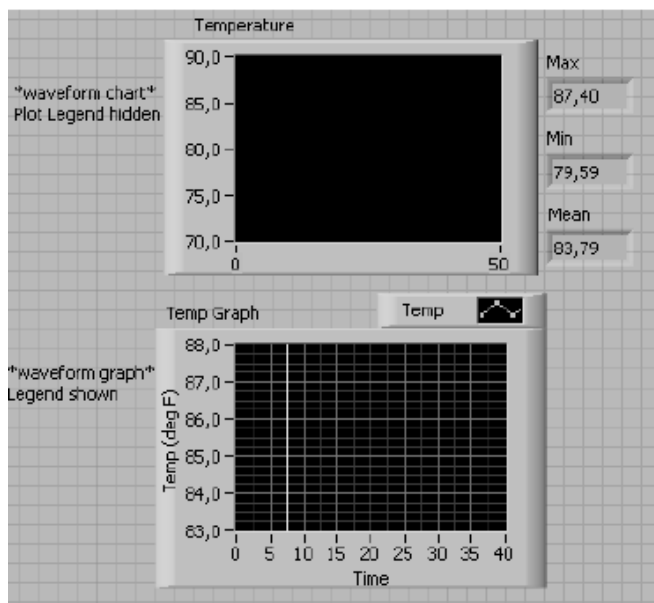


Рис. 4.52

Построить Блок-диаграмму в соответствии с рис. 4.53.

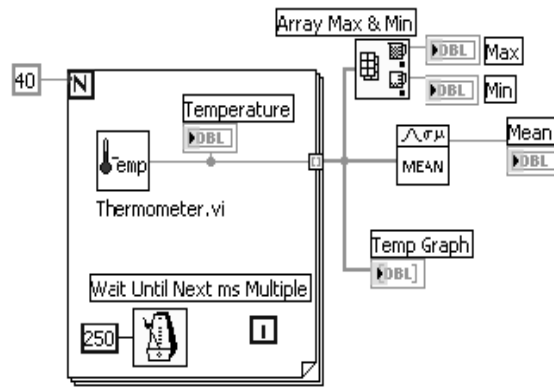


Рис. 4.53

Функцию ВВП Mean.vi для вычисления среднего значения найти через поиск и подключить.

Установить для вычисления максимума и минимума функции Массив (рис. 4.54).

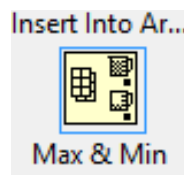


Рис. 4.54

Цикл с фиксированным числом итераций выполняется 40 раз. Функция Задержка до следующего кратного интервала заставляет каждую итерацию выполняться каждые 0,25 с. ВП сохраняет результаты измерений температуры в массиве, созданном на границе цикла с фиксированным значением числа итераций, используя автоиндексирование.

После завершения цикла массив переходит к различным узлам данных. Функция Массив Max&Min возвращает максимальное и минимальное значения температуры, функция Mean.vi – среднее значение.

Запустить ВП.

Используя контекстное меню сделать ее видимой(рис. 4.55), , опцию Видимые элементы ► Панель редактирования шкалы, увеличить график.

Изменить точность значений шкалы Y так, чтобы график отображал числа с тремя знаками после запятой.

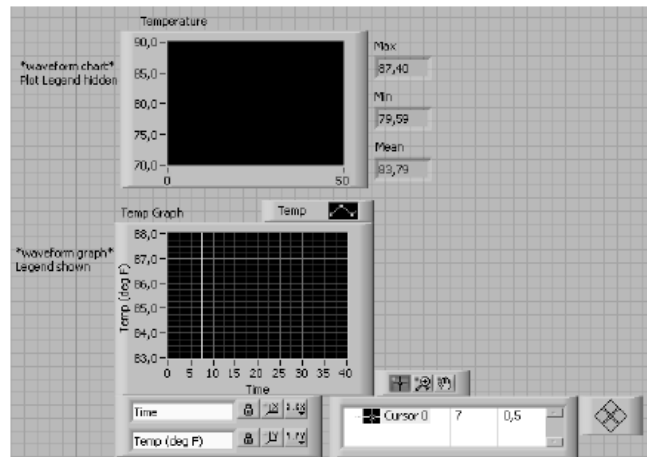


Рис. 4.55

Для перемещения курсора по графику нужно использовать инструмент управления, обращая внимание на то, как изменяются значения X и Y на элементе отображения курсора. Эти значения помогут определить координаты любой точки на графике.

С помощью кнопок управления курсором подвигать курсором в разных направлениях.

Сохранить прибор.

ЗАКЛЮЧЕНИЕ

В практикуме были рассмотрены вопросы, относящиеся к практической части дисциплин «Моделирование телекоммуникационных устройств» и «Информационные системы разработки устройств телекоммуникаций». Теоретическая часть, представленная в виде краткой справочной информации по каждому изучаемому разделу, является необходимой и достаточной для выполнения всех заданий в среде графического программирования LabView. Задания составлены с учетом предварительно полученных знаний и навыков студентов по ранее изученным дисциплинам «Информатика», «Информатика и основы алгоритмизации».

Задания содержат часть основных функций, необходимых для приобретения практических навыков по темам основных разделов изучаемой дисциплины и сопровождаются иллюстрациями.

В пособии рассматриваются и разбираются не все функции, имеющиеся в среде графического программирования, а только необходимый набор типичных функций, операций, что позволяет получить первоначальные знания и практические навыки и в дальнейшем поможет эффективно и качественно выполнять и решать задачи по другим изучаемым дисциплинам. Полученные навыки помогут студентам решать задачи в рамках курсовых, дипломных работ, могут применяться для исследовательской и научной работы.

СПИСОК ЛИТЕРАТУРЫ

1. Жуков, К. Г. Модельное проектирование встраиваемых систем в LabView [Электронный ресурс] : учеб. пособие / К. Г. Жуков. – М. : ДМК Пресс, 2011. – 680 с.
2. Колесов, Ю. Моделирование систем : практикум по компьютерному моделированию [Электронный ресурс] : учеб. пособие / Ю. Колесов, Ю. Сениченков. – СПб. : БХВ-Петербург, 2007. – 352 с.
3. Кознов, Д. В. Основы визуального моделирования [Электронный ресурс] : учеб. пособие / Д. В. Кознов. – М. : БИНОМ. Лаборатория знаний, Интернет-университет информационных технологий (ИНТУИТ), 2008. – 246 с.
4. Тупик, Н. В. Компьютерное моделирование [Электронный ресурс] : учеб. пособие / Н. В. Тупик. – Саратов : Вузовское образование, 2013. – 230 с.
5. Евдокимов, Ю. К. LabView для радиоинженера: от виртуальной модели до реального прибора : практическое руководство для работы в программной среде LabView [Текст] : учеб. пособие для вузов / Ю. К. Евдокимов, В. Р. Линдваль, Г. И. Щербаков. – М. : ДМК Пресс, 2007. – 399 с.
6. Загидулин, Р. Ш. Multisim, LabView и Signal Express. Практика автоматизированного проектирования электронных устройств [Текст] / Р. Ш. Загидулин ; [ред. Т. Н. Лидина]. – М. : Горячая линия-Телеком, 2009. – 366 с.
7. Макаров, Л. М. Моделирование систем : учеб. пособие / Л. М. Макаров ; рец. С. В. Протасеня ; СПбГУТ. – СПб., 2010. – 72 с.
8. Советов, Б. Я. Моделирование систем : учебник для бакалавров / Б. Я. Советов, С. А. Яковлев ; рец. О. С. Чугреев ; СПбЭТУ «ЛЭТИ». – М. : Юрайт, 2013. – 343 с.
9. Федосов, В. П. Цифровая обработка звуковых и вибросигналов в LabView [Текст] : справочник функций системы NI Sound and Vibration LabView / В. П. Федосов. – М. : ДМК, 2010. – 1296 с.

Мусаева Татьяна Вагифовна

**МОДЕЛИРОВАНИЕ
ТЕЛЕКОММУНИКАЦИОННЫХ УСТРОЙСТВ
*ГРАФИЧЕСКАЯ СРЕДА ПРОГРАММИРОВАНИЯ***
Практикум

Редактор *И. И. Щенсяк*
Компьютерная верстка *Н. А. Ефремовой*

План издания 2018 г., п. 113

Подписано к печати .2018
Объем 3,5 усл.-печ. л. Тираж 12 экз. Заказ 881

Редакционно-издательский отдел СПбГУТ
193232 СПб., пр. Большевиков, д. 22

Отпечатано в СПбГУТ