

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ

**НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ХАРЬКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ»**

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ
К ЛАБОРАТОРНЫМ РАБОТАМ**

**по курсу «КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ
СРЕДСТВ ИЗМЕРЕНИЙ»**

для студентов специальности 7.091302
«Метрология и измерительная техника»
дневного и заочного обучения

Утверждено
редакционно-издательским
советом университета,
протокол №
от .

Харьков НТУ «ХПИ» 2007

Методичні вказівки до лабораторних робіт з курсу «Комп'ютерне моделювання засобів вимірювання» для студентів спеціальності 7.091302 денного та заочного навчання / За ред. В.М. Балєва, О.Л. Харченко,. – Харків: НТУ «ХПІ», 2007 – ХХ с. – Рос. мовою

Укладачі: В.М. Балєв
О.Л. Харченко

Рецензент О.П. Давиденко

Кафедра «Інформаційно-вимірювальні технології і системи»

СОДЕРЖАНИЕ

| | |
|---|----|
| ВВЕДЕНИЕ | 4 |
| ЛАБОРАТОРНАЯ РАБОТА №1. ОСНОВЫ РАБОТЫ В LABVIEW | 6 |
| ЛАБОРАТОРНАЯ РАБОТА №2. РАБОТА СО СТРУКТУРАМИ В LABVIEW | 15 |
| ЛАБОРАТОРНАЯ РАБОТА №3 ГРАФИЧЕСКОЕ ОТОБРАЖЕНИЕ ИНФОРМАЦИИ В LABVIEW | 23 |

ВВЕДЕНИЕ

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) представляет собой среду графического программирования, которая широко используется в промышленности, образовании и научно-исследовательских лабораториях в качестве стандартного инструмента для сбора данных и управления приборами. LabVIEW - мощная и гибкая программная среда, применяемая для проведения измерений и анализа полученных данных. LabVIEW - многоплатформенная среда: Вы можете использовать ее на компьютерах с операционными системами Windows, MacOS, Linux, Solaris . Персональные компьютеры являются более гибкими инструментами, чем традиционные измерительные приборы, поэтому создание собственной программы на LabVIEW, или виртуального прибора (ВП), является довольно несложным делом, а интуитивно понятный пользовательский интерфейс в среде LabVIEW делает разработку программ и их применение весьма интересным и увлекательным занятием.

Концепция LabVIEW сильно отличается от последовательной природы традиционных языков программирования, предоставляя разработчику легкую в использовании графическую оболочку, которая включает в себя весь набор инструментов, необходимых для сбора данных, их анализа и представления полученных результатов. С помощью графического языка программирования LabVIEW, именуемого G (Джей), вы можете программировать вашу задачу из графической блок-диаграммы, которая компилирует алгоритм в машинный код. Являясь превосходной программной средой для бесчисленных применений в области науки и техники, LabVIEW поможет вам решать задачи различного типа, затрачивая значительно меньше времени и усилий по сравнению с написанием традиционного программного кода

Распространение LabVIEW за пределами лабораторий пошло по всем направлениям: вверх (на борту космических аппаратов), вниз (на подводных лодках) и по горизонтали (от буровых установок в Северном море до промышленных предприятий в Новой Зеландии). С ростом возможностей Интернет сфера применения LabVIEW стала расширяться не только в географическом, но и в виртуальном пространстве. Все большее число разработчиков создает виртуальные приборы, допускающие удаленное управление и наблюдение через Интернет. Измерительные системы на основе виртуальных приборов отличаются своей многофункциональностью, гибкостью и низкой стоимостью как с точки зрения оборудования, так и с точки зрения затрат времени на разработку.

Пожалуй, лучшим способом объяснить причины столь широкого распространения пакета LabVIEW будет обобщение способов его использования. Во всех видах человеческой деятельности существуют области, где не обойтись без определенных видов измерений - очень часто это температурные измерения, например в печах, холодильниках, парниках, технологических помещениях и даже... в кастрюле с супом. Кроме температуры, часто измеряют давление, силу, пространственное смещение, механическое напряжение и т.д. - список огромный! Сейчас персональные компьютеры проникли практически во все сферы жизнедеятельности. LabVIEW ускоряет внедрение компьютера в измерительные системы - и не только потому, что облегчает проведение измерений, он также дает возможность проанализировать измеренные величины, отобразить их на графиках и в отчетах и при желании опубликовать.

ЛАБОРАТОРНАЯ РАБОТА №1

ОСНОВЫ РАБОТЫ В LABVIEW

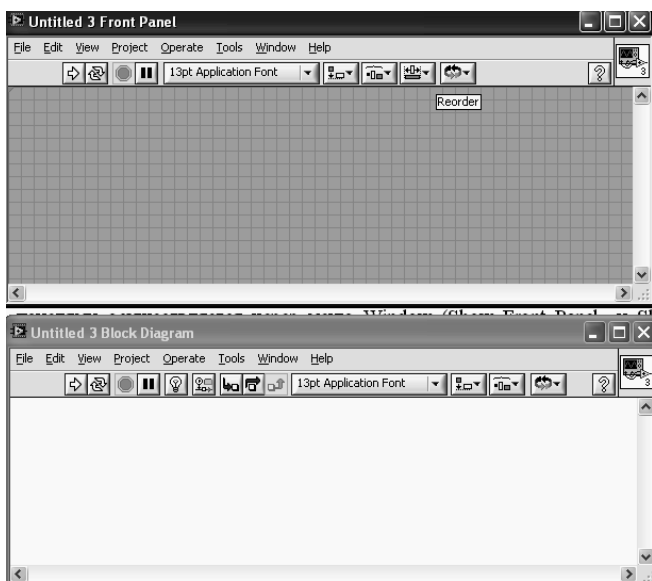
1 Цель работы: Получение практических навыков и использование различных приемов при разработке виртуальных приборов в среде LabView.

2 Описание лабораторного макета: В ходе лабораторной работы используется инструментарий пакета LabView 8.2.

3 Краткие справочные данные

3.1 Передняя панель и диаграмма

Перед разработчиком после попадания в среду LabView и выбора команды



Blank VI появляются две пустые панели, являющиеся главными панелями, в которые как бы закладывается тело программы. Первая из них Front Panel- лицевая (передняя) панель, на которой возможно создавать произвольный виртуальный прибор путем выбора разнообразных деталей из меню Control (появляется при нажатии правой клавиши манипулятора), среди которых большое разнообразие ручек настроек, переключателей, индикаторов, панелей отображения графиков, а также элементов дизайна для улучшения внешнего вида виртуального прибора

[Пейч Л.И., Точилин Д.А., Поллак Б.П., LabView для новичков и специалистов. – М.: Горячая линия – Телеком, 2004. – 384с.]

Вместе с лицевой панелью появляется функциональная панель Block Diagram – панель диаграммы, имеющая то же имя что и передняя панель. На этой панели одновременно с элементами, располагаемыми на передней панели, появляются соответствующие им значки (терминалы). Это небольшие пиктограммы, представляющие собой разъемы для подсоединения к этим приборам других приборов или функций. Соединяя терминалы между собой в разрешенных комбинациях напрямую, либо через функциональные узлы, разработчик получает схему программы. Если соединения сделаны без ошибок, то программа готова к запуску на выполнение.

Панели между собой неравноправны. Основной является передняя панель, с ней можно работать не открывая панели диаграмм. Переключение между панелями осуществляется через меню Window (Show Front Panel и Show Block Diagram), или с клавиатуры клавишами – Ctrl-E.

Запуск программы и работа с ней осуществляется обычно с передней панели, поскольку на ней обычно имеются средства имитации прибора с элементами управления и индикации.


Пошаговая отладка программ проводится на функциональной панели. Здесь


верхнее меню представляет разработчику больше возможностей. "Включив" лампочку из меню и запустив программу, разработчик может наблюдать последовательность выполнения действий, следя за шариками, "бегущими" по проводникам и после остановки шарика видеть результат вычислений каждой операции. Нажав на ПАУЗУ из верхнего меню можно выполнять отладку по шагам, с остановкой после каждого действия. Во время остановки можно переключаться на переднюю панель и изменять состояние органов управления и опять вернуться на панель диаграммы, чтобы выполнить следующий шаг. Это позволяет устранить недоработки которые не были выявлены на этапе построения программы.

3.2 Палитра инструментов


При создании нового или открытии уже созданного виртуального прибора кроме двух панелей на экране можно отобразить панель называемую палитрой инструментов (Tools Palette), если она не отображается, то ее можно активировать из меню View (Tools Palette), или временно ее можно вызвать нажатием правой клавиши мыши при нажатой клавише Shift. Эта панель представляет собой группу кнопок с зависимой фиксацией, в каждый момент времени может быть нажата только одна кнопка.




Кнопка Палец  управляет всеми переключателями, ползунковыми элементами, т.е. имитирует руки, управляющие реальным прибором.


Кнопка Стрелка  выполняет перемещение и позиционирование элементов при их установке на какую-либо панель, выделение одного или нескольких элементов. Кнопкой также пользуются при расширении функциональных возможностей некоторых блоков. Переключение между кнопками Палец и Стрелка на передней панели может осуществляться с помощью клавиши Пробел клавиатуры. Кнопка А


позволяет организовать вывод цифробуквенной информации в элементы лицевой панели или панели диаграммы, а также везде где необходимо написать текст.

Кнопка Катушка  имитирует провод, связывающий между собой детали диаграммы. Переключение между кнопками Катушка и Стрелка на панели диаграммы производится клавишей Пробел.

Кнопка Клавиши  позволяет вызвать свойства выбранного объекта.

Кнопка Рука  обеспечивает быстрое перемещение с помощью мыши видимой части панели в плоскости экрана.








Красная кнопка Стоп  позволяет останавливать работу программы том месте, где необходимо пользователю.


Кнопка Р (Probe)  позволяет в любое место программы вставлять индикаторы для отображения информации о вычисляемом в процессе работы программы параметре.


Кнопка Пипетка  дает возможность копировать цвет при изменении дизайна передней панели.

Кнопка Кисточка  устанавливает цвета рисунков и фона по отдельности.

3.3 Верхнее меню

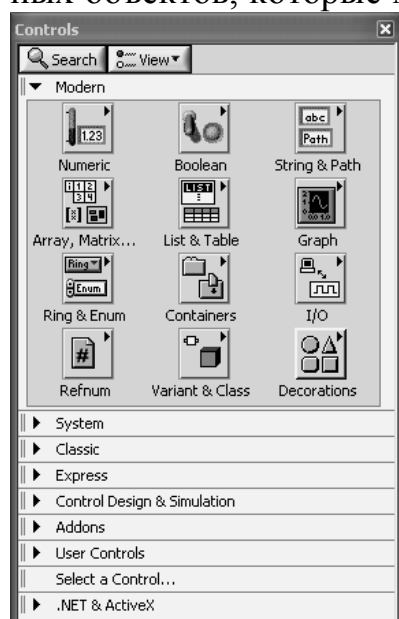
Верхнее меню состоит из строки заголовка, в которой указывается имя файла, строки главного меню сходной с другими приложениями Windows и командной строки, называемой Toolbar. В командной строке расположены управляющие клавиши, часть из которых используется при написании программы, а часть при выполнении программы. Кнопкой с белой стрелкой  (Run - выполнение) позволяет запустить программу на одноразовое выполнение. Если на этапе разработки программы сделана ошибка, то появляется изображение сломанной стрелки , т.е. программа не может быть запущена. Нажав на эту кнопку вызывается окно с сообщением о характере и месте ошибки (ошибок). Кнопкой с двумя стрелками  (Run Continuously - циклическое выполнение) можно запустить программу на циклическое выполнение. Кнопка (Abort Execution – прерывание выполнения)  останавливает работающие программы. Кнопка (Pause – пауза)  приостанавливает работу программы, а также позволяет отлаживать ее по шагам. Кнопка (Highlight Execution – выделение выполнения)  как бы освещает выполнение программы на панели диаграммы, последовательно индицируя результаты операций, что упрощает отладку по шагам. Три следующие кнопки  используются при пошаговой отладке программы.

Кнопки  Align Objects, Distribute Objects, Reorder (выравнивание объектов, распределение объектов, переупорядочивание) позволяет быстро расположить выделенную группу объектов, а также размещать объекты на переднем или заднем плане, как бы позволяя одним объектам затенять другие.

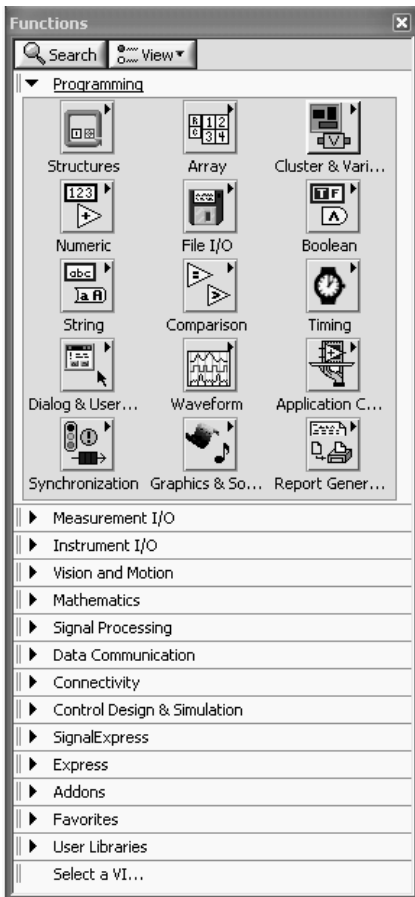
Кнопка  (Text Settings – установки текста) позволяет выбрать шрифт для оформления текстовой информации, его размер, стиль, цвет.

3.4 Палитра Controls (управления) и палитра Functions (функций).

Палитра управления и палитра функций составлены из значков верхнего уровня, представляющих подпалитры, дающие доступ к полному диапазону доступных объектов, которые могут использоваться в создании ВП. К подпалитрам можно обращаться, нажимая на значок верхнего уровня. Подпалитра может также быть преобразована в плавающую палитру, которая остается на вашем экране.



Вы добавляете средство управления и индикаторы к передней панели через палитру Controls. Каждая опция в палитре отображает подпалитру доступного средства управления и индикаторов для выбора. Если палитра Controls - не видна, Вы можете открыть палитру, выбрав Show Controls Palette в меню Windows. Вы можете также вызвать палитру Controls, открыв всплывающее меню на пустой области на передней панели. Вы можете открыть всплывающее меню, щелкнув по пустой области передней панели правой кнопкой мыши. Палитра Controls может быть "пришпилена" к рабочему столу с помощью кнопки в левом углу палитры, либо убрана кнопкой "крестик".

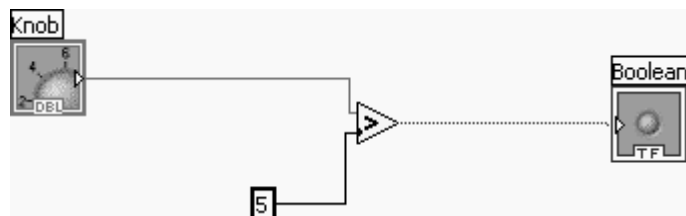
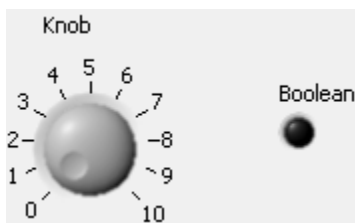


С помощью палитры Functions Вы формируете блок-схему. Каждая опция в палитре отображает подпалитру значков верхнего уровня. Если палитра Functions - не видна, Вы можете вызвать палитру, выбрав Show Functions Palette в меню Windows. Вы можете также открыть палитру Функций, вызвав всплывающее меню на пустой области в окне Diagram. Палитра Функций может быть "пришпилена" к рабочему столу с помощью кнопки в левом углу палитры, либо убрана кнопкой "крестик".

3.4 Создание первого прибора в LabView

Создайте новый виртуальный прибор выбрав команду Blank VI после запуска LabView. Станьте указателем мыши на лицевую панель и вызовите палитру Controls нажатием правой клавиши манипулятора. На вкладке Numeric выберите регулятор Knob, на вкладке Boolean выберите Round Led. Перейдите к окну диаграммы, вызовите палитру Functions и на вкладке Comparison выберите значок больше (Greater?). Выход регулятора подключите к одному из входов значка больше, а на другой вход подключите значение константы

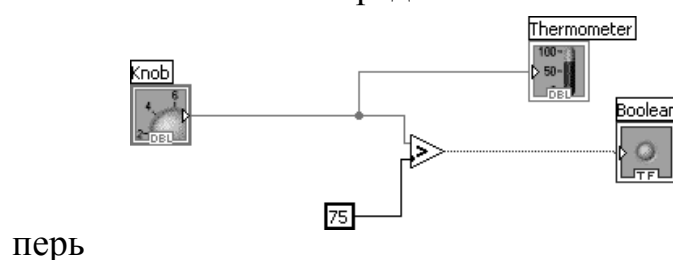
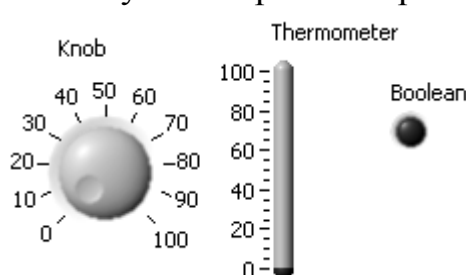
5. Выход значка больше (Greater?) подключите к значку Round Led. У Вас должны были получиться приблизительно такие картинки:



Перейдите к лицевой панели и запустите программу в циклическом режиме. Изменяя состояния регулятора, убедитесь, что светодиод загорается, при значениях более чем 5.

Из меню Numeric лицевой панели добавьте термометр, измените диапазон изменения регулятора Knob, нажав правую клавишу мышки на регуляторе выберите его свойства (Properties -> Scale -> Min=0, Max=100).

На диаграмме подключите вход термометра к регулятору и измените значение константы на входе значка больше (Greater?) на 75. После этих изменений вы должны получить картинки приблизительно похожие на представленные в методичке.



перь

Термометр снова загорается

пустите программу в режиме циклического выполнения, убедитесь, что при изменении регулятора изменяются показания термометра, а при превышении уровня 75 загорается светодиод.

Сохраните полученный виртуальный прибор с именем Termometr.vi.

4. Ход работы

- 1) Получить и ознакомиться с индивидуальным заданием.
- 2) Сформировать требуемые инструменты.
- 3) Отладить программу и представить преподавателю для проверки.
- 5) Составить индивидуальный отчет о проделанной работе, который должен включать в себя:
 - ✓ Вариант и содержание исходного задания.
 - ✓ Текст программы и результаты выполнения программы (лицевая панель и диаграмма.
- 5) Защита отчета проходит индивидуально для каждого студента.

5. Контрольные вопросы

- 1) Из чего состоит виртуальный прибор в LabView ?
- 2) Каким образом осуществляется переход между окнами диаграммы и лицевой панели
- 3) Как вызывается палитра инструментов, палитра Controls и палитра Functions?
- 4) Назначение элементов верхнего меню в LabView.
- 5) Назначение кнопок палитры инструментов.
- 6) Каким образом запустить программу на исполнение?
- 7) Какие средства для облегчения отладки программы имеются в LabView ?

6. Варианты заданий

Задание №1.

1. Создать аналоговый вольтметр, диапазон отображаемых значений от 0 до 20 вольт. Входное напряжение имитировать с помощью произвольных регуляторов с вкладки Numeric меню Controls. В приборе предусмотреть сигнализацию превышения входным напряжением уровня 18 В (красный цвет), и сигнализацию уменьшения напряжения до значений меньше чем 3 В (красный цвет). Нормальные значения напряжения сигнализируются зеленым цветом сигнализаторов.
2. Отладить программу
3. Добавить в программу цифровой индикатор для отображения результата измерения.

Задание №2.

1. Создать аналоговый амперметр, диапазон отображаемых значений от 0 до 5 ампер. Входной ток имитировать с помощью произвольных регуляторов с

вкладки Numeric меню Controls. В приборе предусмотреть сигнализацию превышения входным током уровня 3,5 А (красный цвет) , и сигнализацию уменьшения тока до значений меньше чем 0,5 А (красный цвет). Нормальные значения тока сигнализируются зеленым цветом сигнализаторов.

2. Отладить программу
3. Добавить в программу цифровой индикатор для отображения результата измерения тока.

Задание №3.

1. Создать аналоговый частотомер, диапазон отображаемых значений от 45 до 55 герц. Входное значение частоты имитировать с помощью произвольных регуляторов с вкладки Numeric меню Controls. В приборе предусмотреть сигнализацию превышения частотой уровня 51 Гц (красный цвет) , и сигнализацию уменьшения частоты до значений меньше чем 49,5 Гц (красный цвет). Нормальные значения частоты сигнализируются зеленым цветом сигнализаторов.
2. Отладить программу
3. Добавить в программу цифровой индикатор для отображения результата измерения.

Задание №4.

1. Создать аналоговый фазометр, диапазон отображаемых значений сдвига фазы от -90 до 90 градусов. Входное значение фазы имитировать с помощью произвольных регуляторов с вкладки Numeric меню Controls. В приборе предусмотреть сигнализацию превышения сдвига фазы отрицательного значения -50 градусов (красный цвет), и положительного значения 60 градусов (красный цвет). Нормальные значения фазового сдвига сигнализируются зеленым цветом сигнализаторов.
2. Отладить программу
3. Добавить в программу цифровой индикатор для отображения результата измерения.

Задание №5.

1. Создать аналоговый частотомер, диапазон отображаемых значений от 20 до 15000 герц. Входное значение частоты имитировать с помощью произвольных регуляторов с вкладки Numeric меню Controls. В приборе предусмотреть сигнализацию превышения частотой уровня 12000 Гц (красный цвет) , и сигнализацию уменьшения частоты до значений меньше чем 30,5 Гц (красный цвет). Нормальные значения частоты сигнализируются зеленым цветом сигнализаторов.
2. Отладить программу
3. Добавить в программу цифровой индикатор для отображения результата измерения.

Задание №6.

1. Создать аналоговый ваттметр постоянного тока, диапазон отображаемых значений от 0 до 1000 ватт. Входные значения постоянного напряжения в диапа-

зоне от 0 до 200 В и тока в диапазоне от 0 до 5 ампер имитировать с помощью произвольных регуляторов с вкладки Numeric меню Controls. В приборе предусмотреть сигнализацию превышения мощностью уровня 910 Вт (красный цвет), и сигнализацию уменьшения мощности до значений меньше чем 95 Вт (красный цвет). Нормальные значения мощности сигнализируются зеленым цветом сигнализаторов.

2. Отладить программу
3. Добавить в программу цифровой индикатор для отображения результата измерения.

Задание №7.

1. Создать аналоговый ваттметр переменного тока, диапазон отображаемых значений от 0 до 2,2 киловатт. Входные значения переменного напряжения в диапазоне от 0 до 220 В, тока в диапазоне от 0 до 10 ампер, и сдвига фаз в диапазоне 0 - 20 градусов имитировать с помощью произвольных регуляторов с вкладки Numeric меню Controls. В приборе предусмотреть сигнализацию превышения мощностью уровня 1900 Вт (красный цвет), и сигнализацию уменьшения мощности до значений меньше чем 395 Вт (красный цвет). Нормальные значения мощности сигнализируются зеленым цветом сигнализаторов.
2. Отладить программу
3. Добавить в программу цифровой индикатор для отображения результата измерения.

Задание №8.

1. Создать аналоговый измеритель уровня жидкости в резервуаре с отображением результата измерения на индикаторе Tank, диапазон отображаемых значений от 0 до 15 метров. Входное значение уровня имитировать с помощью произвольных регуляторов с вкладки Numeric меню Controls. В приборе предусмотреть сигнализацию превышения уровня 13,5 метров (красный цвет), и сигнализацию уменьшения уровня до значений меньше чем 3,95 метра (красный цвет). Нормальные значения уровня сигнализируются зеленым цветом сигнализаторов.
2. Отладить программу
3. Добавить в программу цифровой индикатор для отображения результата измерения.

Задание №9.

1. Создать омметр, диапазон отображаемых значений от 0 до 100 килоом. Входные значения сопротивления имитировать с помощью произвольных регуляторов с вкладки Numeric меню Controls. В приборе предусмотреть сигнализацию превышения сопротивлением уровня 95 кОм (красный цвет), и сигнализацию уменьшения сопротивления до значений меньше чем 5,5 кОм (красный цвет). Нормальные значения сопротивления сигнализируются зеленым цветом сигнализаторов.
2. Отладить программу
3. Добавить в программу цифровой индикатор для отображения результата измерения.

Задание №10.

1. Создать цифровой вольтметр, диапазон отображаемых значений от 0 до 500 вольт. Входное напряжение имитировать с помощью произвольных регуляторов с вкладки Numeric меню Controls. В приборе предусмотреть сигнализацию превышения входным напряжением уровня 480 В (красный цвет), и сигнализацию уменьшения напряжения до значений меньше чем 35 В (красный цвет). Нормальные значения напряжения сигнализируются зеленым цветом сигнализаторов.
2. Отладить программу
3. Добавить в программу аналоговый индикатор для отображения результата измерения.

Задание №11.

1. Создать цифровой амперметр, диапазон отображаемых значений от 0 до 20 ампер. Входной ток имитировать с помощью произвольных регуляторов с вкладки Numeric меню Controls. В приборе предусмотреть сигнализацию превышения входным током уровня 15 А (красный цвет) , и сигнализацию уменьшения тока до значений меньше чем 3 А (красный цвет). Нормальные значения тока сигнализируются зеленым цветом сигнализаторов.
2. Отладить программу
3. Добавить в программу аналоговый индикатор для отображения результата измерения тока.

Задание №12.

1. Создать цифровой частотомер, диапазон отображаемых значений от 0 до 20000 герц. Входное значение частоты имитировать с помощью произвольных регуляторов с вкладки Numeric меню Controls. В приборе предусмотреть сигнализацию превышения частотой уровня 14 кГц (красный цвет) , и сигнализацию уменьшения частоты до значений меньше чем 20 Гц (красный цвет). Нормальные значения частоты сигнализируются зеленым цветом сигнализаторов.
2. Отладить программу
3. Добавить в программу аналоговый индикатор для отображения результата измерения.

Задание №13.

1. Создать цифровой фазометр, диапазон отображаемых значений сдвига фазы от -45 до 45 градусов. Входное значение фазы имитировать с помощью произвольных регуляторов с вкладки Numeric меню Controls. В приборе предусмотреть сигнализацию превышения сдвига фазы отрицательного значения -30 градусов (красный цвет), и положительного значения 25 градусов (красный цвет). Нормальные значения фазового сдвига сигнализируются зеленым цветом сигнализаторов.
2. Отладить программу
3. Добавить в программу аналоговый индикатор для отображения результата измерения.

Задание №14.

1. Создать цифровой частотомер, диапазон отображаемых значений от 40 до 60 герц. Входное значение частоты имитировать с помощью произвольных регуляторов с вкладки Numeric меню Controls. В приборе предусмотреть сигнализацию превышения частотой уровня 55 Гц (красный цвет), и сигнализацию уменьшения частоты до значений меньше чем 44 Гц (красный цвет). Нормальные значения частоты сигнализируются зеленым цветом сигнализаторов.
2. Отладить программу
3. Добавить в программу аналоговый индикатор для отображения результата измерения.

Задание №15.

1. Создать цифровой ваттметр постоянного тока, диапазон отображаемых значений от 0 до 2000 ватт. Входные значения постоянного напряжения в диапазоне от 0 до 500 В и тока в диапазоне от 0 до 4 ампер имитировать с помощью произвольных регуляторов с вкладки Numeric меню Controls. В приборе предусмотреть сигнализацию превышения мощностью уровня 1850 Вт (красный цвет), и сигнализацию уменьшения мощности до значений меньше чем 170 Вт (красный цвет). Нормальные значения мощности сигнализируются зеленым цветом сигнализаторов.
2. Отладить программу
3. Добавить в программу аналоговый индикатор для отображения результата измерения.

Задание №16.

1. Создать цифровой ваттметр переменного тока, диапазон отображаемых значений от 0 до 3,5 киловатт. Входные значения переменного напряжения в диапазоне от 0 до 220 В и тока в диапазоне от 0 до 15 ампер имитировать с помощью произвольных регуляторов с вкладки Numeric меню Controls. В приборе предусмотреть сигнализацию превышения мощностью уровня 3250 Вт (красный цвет), и сигнализацию уменьшения мощности до значений меньше чем 750 Вт (красный цвет). Нормальные значения мощности сигнализируются зеленым цветом сигнализаторов.
2. Отладить программу
3. Добавить в программу аналоговый индикатор для отображения результата измерения.

Задание №17.

1. Создать цифровой измеритель уровня жидкости в резервуаре, диапазон отображаемых значений от 0 до 12 метров. Входное значение уровня имитировать с помощью произвольных регуляторов с вкладки Numeric меню Controls. В приборе предусмотреть сигнализацию превышения уровня 11,5 метров (красный цвет), и сигнализацию уменьшения уровня до значений меньше чем 0,95 метра (красный цвет). Нормальные значения уровня сигнализируются зеленым цветом сигнализаторов.
2. Отладить программу

3. Добавить в программу аналоговый индикатор для отображения результата измерения.

Задание №18.

1. Создать омметр, диапазон отображаемых значений от 0 до 200 килоом. Входные значения сопротивления имитировать с помощью произвольных регуляторов с вкладки Numeric меню Controls. В приборе предусмотреть сигнализацию превышения сопротивлением уровня 185 кОм (красный цвет), и сигнализацию уменьшения сопротивления до значений меньше чем 3,3 кОм (красный цвет). Нормальные значения сопротивления сигнализируются зеленым цветом сигнализаторов.
2. Отладить программу
3. Добавить в программу аналоговый индикатор для отображения результата измерения.

ЛАБОРАТОРНАЯ РАБОТА №2

РАБОТА СО СТРУКТУРАМИ В LABVIEW

1. Цель работы: Получение практических навыков написания программ в среде LabView с использованием условных операторов и циклов.

2. Описание лабораторного макета: В ходе лабораторной работы используется инструментарий пакета LabView 8.2.

3. Подготовка к проведению лабораторной работы. Задание на самостоятельную работу

В ходе самостоятельной работы студенты должны ознакомиться со способами организации ветвлений программ и циклов в LabView.

3.1 Структуры в LabView

При разработке блок-схемы программы закладывается алгоритм её работы, порядок производимых операций. И если в текстовых программах последовательность определяется написанным текстом, то в LabView находящиеся рядом, но не связанные между собой, программные блоки, с точки зрения программиста, выполняются одновременно. Поэтому, если важна последовательность операций, то Вы сами должны позаботиться о порядке вычислений и заложить его в структуру программы. В LabView существует раздел, который позволяет регламентировать этот процесс. Это так называемые структуры. Вызов структур осуществляется из функциональной палитры меню Structures. Имеются несколько возможных способов изменения следования программных операций:

Flat Sequence (последовательная структура) – это инструмент состоящий из отдельных страничек (кадров) на каждой из которых можно сформировать часть программы. Последовательность выполнения фрагментов слева направо. При этом ни одна из страничек не пропускается. Процедура выполнения завершается, когда выполнится часть программы расположенная на последней страничке.

Stacked Sequence Structure – структура состоящая из одной или нескольких листочков (кадров) исполняющихся последовательно, один за другим в соответствии с номером. Аналогична по выполнению Flat Sequence, но позволяет существенно сэкономить место на диаграмме.

Case Structure (структура с выбором) – это инструмент похожий на условный оператор текстовых программ. Исполняется код расположенный на страничке True или False в зависимости от некоторого внешнего условия.

For Loop (определенный цикл) - оператор цикла, в котором заранее оговорено количество повторяемых выполнений кода.

While Loop (условный цикл) – оператор цикла, в котором число повторений заранее неизвестно, а имеется дополнительное условие выхода из цикла.

Timed Loop – исполняется часть диаграммы или несколько окон, последовательно в цикле с определенным Вами периодом.

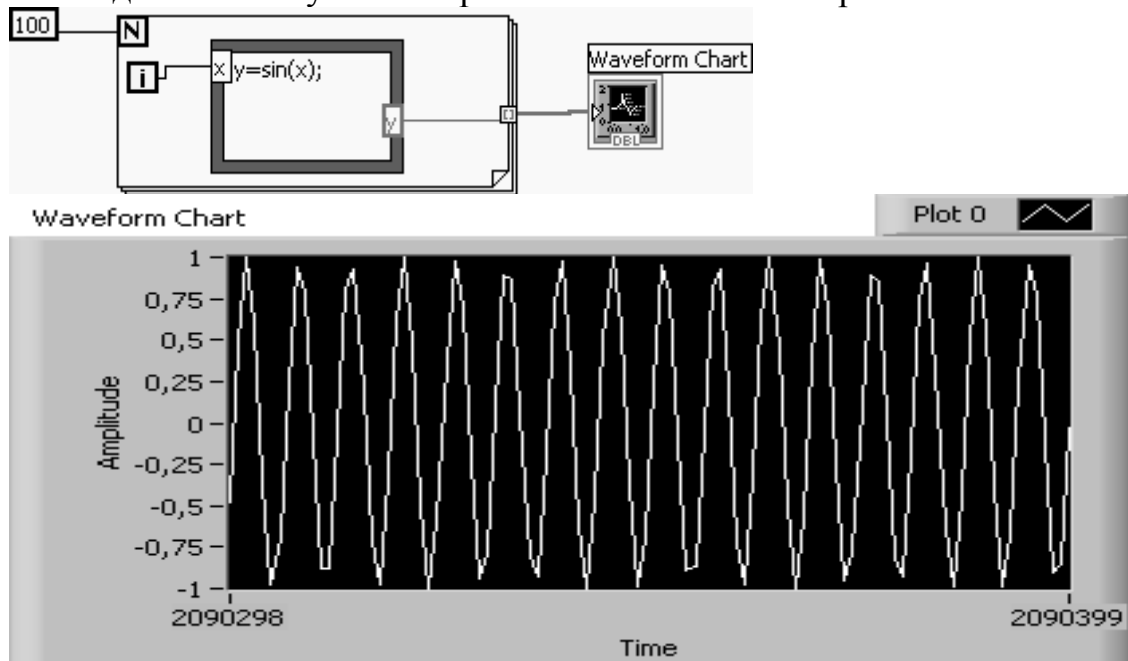
Использование оператора For Loop

Структура **Цикл с фиксированным числом итераций** (For Loop) эквивалентна текстовому оператору `for i = 0 to N - 1 do ...`. При помещении структуры на панель блок-диаграммы ее контур в виде прямоугольника должен быть растянут так, чтобы охватить существующий код программы, который должен выполняться циклически заданное число раз, или так, чтобы позволить разместить в нем новый код программы. Если помещаемая в структуру или перемещаемая внутри структуры функция пересекается с ее границей, то граница автоматически расширяется. Данная опция может быть отключена для этой структуры путем снятия отметки строки **Auto Grow** в контекстном меню структуры или для всего приложения путем снятия отметки строки **Установить структуры с автоматическим расширением** (Place structures with Auto Grow enabled) диалогового окна **Опции** (Options).

Количество циклов может задаваться с помощью константы или элемента управления, подключенных к **терминалу числа итераций** (count terminal) (прямоугольник в левом верхнем углу структуры с буквой N). Текущее число завершенных итераций цикла содержится в **терминале счетчика итераций** (iteration terminal).

Создайте новый пустой виртуальный прибор выбрав команду **Blank VI** после запуска LabView. Разместите на диаграмме цикл For Loop (палитра Controls, меню Structures), в качестве опции для N установите значение 100. Разместите внутри цикла оператор **Formula Node** (палитра Controls, меню Structures) и впишите внутрь выражение $y = \sin(x)$. Вход Formula Node подключите к Loop iteration (i), а выход y к выходу цикла. На лицевой панели разместите **Waveform Chart** и подключите его вход к выходу цикла. Программа рассчитывает 100 значений синусоиды и отображает сигнал на экране **Waveform Chart**.

У Вас должны получиться приблизительно такие картинки:



Запустите программу на выполнение и убедитесь, что на экране появится синусоида.

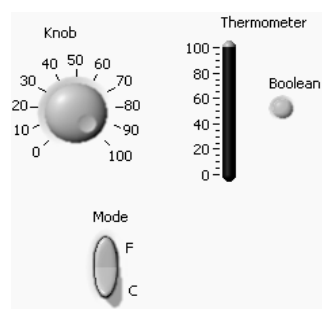
Использование оператора Case Structure

Структура **Вариант** (Case Structure) аналогична операторам case или if then-else в текстовых языках программирования. По умолчанию структура Вариант является логической и имеет два варианта - ИСТИНА (TRUE) и ЛОЖЬ (FALSE), выбираемые с помощью **терминала селектора структуры варианта**. Структура автоматически преобразуется в числовую или строковую при подключении соответственно числовой или строковой переменной к терминалу селектора.

В этом случае структура может иметь практически неограниченное количество вариантов, начиная с нулевого. С помощью строк **Добавить вариант после** (Add Case After) или **Добавить вариант перед** (Add Case Before) можно добавить новый вариант после или до текущего варианта. Одновременно можно наблюдать только один вариант (кадр) структуры. Переход между вариантами производится с помощью **селектора структуры варианта**, расположенного на верхней стороне рамки структуры или контекстного меню структуры. Для использования структуры Вариант необходимо отметить **вариант по умолчанию** (Default).

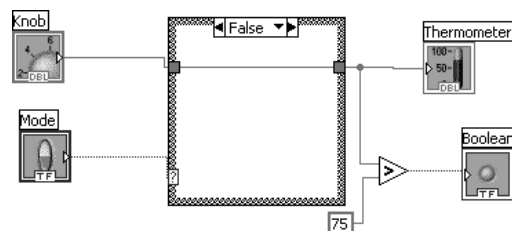
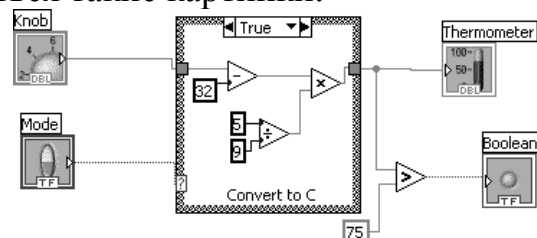
Ввод и вывод данных в структуру Вариант производится с помощью входных и выходных терминалов данных (тоннелей). Создание выходного терминала данных на одной поддиаграмме структуры приводит к его появлению на других поддиаграммах в том же самом месте границы структуры. До подключения данных к выходному терминалу во всех поддиаграммах он сохраняет белый цвет и воспринимается как ошибка создания.

Откройте сохраненный в первой лабораторной работе виртуальный прибор Thermometr.vi. Нашей задачей будет преобразовать его таким образом, чтобы он мог отображать результат в градусах Цельсия и Фаренгейта. Известно, что температура по Цельсию связана с температурой по Фаренгейту соотношением $C=(F-32)*(5/9)$.




Для изменения функциональных возможностей прибора на лицевой панели разместим переключатель Vert Rocker, а на диаграмме добавим Case Structure и селектор подключим к выходу переключателя Vert Rocker а на вход подадим сигнал от регулятора Knob. На вкладке True Case Structure запишем выражение преобразования температуры от Фаренгейта к Цельсию. Выход Case Structure подключим к термометру. На вкладке False вход и выход соединим напрямую, без выполнения действий.

В результате у Вас должны получиться такие картинки:



Запустите программу в режиме циклического выполнения и убедитесь, что при изменении положения переключателя изменяются результаты, отображаемые на термометре.

Откройте на экране одновременно окно лицевой панели и окно диаграммы. В

окне диаграммы нажмите кнопку Highlight Execution  и проследите за прохождением данных по диаграмме. Измените состояние переключателя Vert Rocker и убедитесь, что произошло переключение Case Structure, и выполнение программы происходит по альтернативному пути.

3.4 Использование оператора While Loop

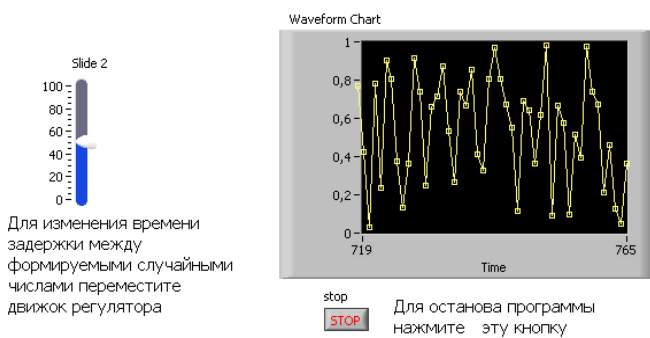
Структура **Цикл по условию** (While Loop) эквивалентна следующему псевдокоду: `do {программа} while {условие}`. Внутри структуры размещаются **терминал счетчика итераций** (iteration terminal) **i** и **терминал условия выхода из цикла** (conditional terminal). Вид структуры с терминалом условия по умолчанию приведен на рисунке

Код программы, размещенный в структуре, выполняется до подачи на терминал условия логической переменной ИСТИНА (TRUE) в структуре на рис. а или ЛОЖЬ (FALSE), в структуре на рис. б. Изменение варианта прекращения выполнения производится с помощью строк **Остановить если истина** (Stop If True) или **Продолжить если истина** (Continue If True) контекстного меню терминала условия. Если терминал условия не подключен к какому-либо выходу, то цикл не будет выполняться. Структура While Loop в отличие от цикла For Loop выполняется всегда хотя бы один раз, так как условие останова проверяется в конце выполнения цикла.



Следует помнить, что условие выхода из цикла нельзя задавать извне цикла. Это бессмысленно, если его значение при входе в цикл было ложно, то цикл выполнится один раз, а если оно было истинно, то цикл будет продолжаться до тех пор, пока его не остановят кнопкой СТОП в командной строке верхнего меню.

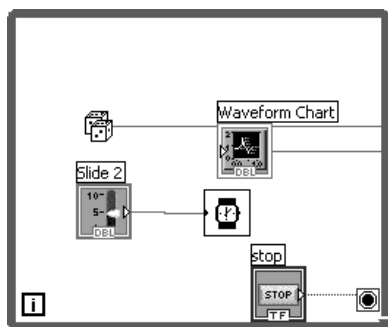
Разработаем прибор для генерации и отображения случайных чисел с использованием структуры While Loop. Для этого

создайте новый пустой виртуальный прибор, выбрав команду Blank VI после запуска LabView. На передней панели разместите регулятор Vertical Pointer Slider (Controls, Modern, Numeric), кнопку останова Stop Button (Controls, Modern, Boolean) и окно для отображения результата генерации числа в графическом виде

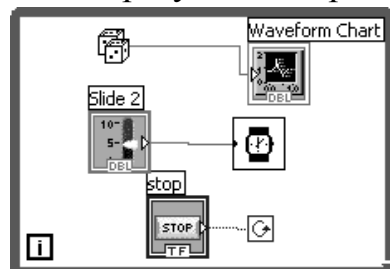


Waveform Chart (Controls, Modern, Graph). Разместите на диаграмме цикл While Loop

(палитра Controls, меню Structures) и внесите в него пиктограммы Vertical Pointer Slider, Waveform Chart, Stop Button. Выход Vertical Pointer Slider подключите к элементу задержки , (Functions, Programming, Timing, Wait(ms)). Stop Button подключите к Loop Condition. На вход Waveform Chart подайте сигналы с генератора случайных чисел  (Functions, Programming, Numeric, Random Number(0-1)).



получится приближенно представленные в мето-



дельно такие картинки как дичке.

Запустите программу на выполнение, убедитесь, что происходит генерация случайных чисел, а останов программы возможен по нажатию кнопки Stop.

Измените свойство Loop Condition на Continue if True. Снова запустите программу и убедитесь, что она заканчивает работу после генерации одного случайного числа.

4. Ход работы

- 1) Получить и ознакомиться с индивидуальным заданием.
- 2) Сформировать требуемые виртуальные инструменты.
- 3) Отладить программу и представить преподавателю для проверки.
- 5) Составить индивидуальный отчет о проделанной работе, который должен включать в себя:
 - ✓ Вариант и содержание исходного задания.
 - ✓ Текст программы и результаты выполнения программы (лицевая панель и диаграмма).
- 5) Защита отчета проходит индивидуально для каждого студента.

5. Контрольные вопросы:

- 1) Какие структуры имеются в LabView для организации ветвления программы?
- 2) В чем состоит основное различие циклов For Loop и While Loop?
- 3) Как задается условие в структуре Case?
- 4) С помощью каких средств оператор может изменять входные данные в программах
- 5) В чем состоит различие структур Flat Sequence и Stacked Sequence Structure.

6. Варианты заданий:

Задание №1.

1. Создать прибор позволяющий производить пересчет количества денег представленного в гривне в доллары и евро по выбору оператора. Входное значение количества денег имитировать с помощью произвольных регуляторов с вкладки Numeric меню Controls.
2. Отладить программу и представить для проверки преподавателю.

Задание №2.

1. Создать прибор позволяющий производить пересчет расстояния представленного в километрах в морские мили и мили США (1 морская миля = 1852 метра, 1 миля США = 1609.3 метра) по выбору оператора. Входное значение расстояния имитировать с помощью произвольных регуляторов с вкладки Numeric меню Controls.
2. Отладить программу и представить для проверки преподавателю.

Задание №3.

1. Создать прибор позволяющий производить пересчет объема нефти представленного в баррелях в галлоны и литры (1 американский галлон для жидкостей = 3.785411784 литра, 1 баррель американский для нефти = 158.9872949 литра)

- по выбору оператора. Входное значение объема нефти имитировать с помощью произвольных регуляторов с вкладки Numeric меню Controls.
2. Отладить программу и представить для проверки преподавателю.

Задание №4.

1. Создать прибор позволяющий производить пересчет времени представленного в часах в сутки и минуты по выбору оператора. Входное значение времени имитировать с помощью произвольных регуляторов с вкладки Numeric меню Controls.
2. Отладить программу и представить для проверки преподавателю.

Задание №5.

1. Создать прибор позволяющий производить пересчет веса представленного в граммах в золотники и унции (1 золотник =4.265754271 грамма, 1 тройская унция =31.1034768 грамма) по выбору оператора. Входное значение веса имитировать с помощью произвольных регуляторов с вкладки Numeric меню Controls.
2. Отладить программу и представить для проверки преподавателю.

Задание №6.

1. Создать прибор позволяющий производить пересчет объема жидкости представленного в немецких квартах в литры и голландские пинты (1 немецкая кварта= 1.145 литра, 1 голландская пинта =0.6063 литра) по выбору оператора. Входное значение объема жидкости имитировать с помощью произвольных регуляторов с вкладки Numeric меню Controls.
2. Отладить программу и представить для проверки преподавателю

Задание №7.

1. Создать прибор позволяющий производить пересчет количества информации представленного в гигабайтах в Терабайты и Мегабайты (1 Гигабайт=1024 Мегабайта, 1 Терабайт=1024 Гигабайта) по выбору оператора. Входное значение количества информации имитировать с помощью произвольных регуляторов с вкладки Numeric меню Controls.
2. Отладить программу и представить для проверки преподавателю.

Задание №8.

1. Создать прибор позволяющий производить пересчет веса представленного в килограммах в фунты и караты (1 фунт русский торговый =409.51242 грамма, 1 карат =0.2 грамма) по выбору оператора. Входное значение веса имитировать с помощью произвольных регуляторов с вкладки Numeric меню Controls.
2. Отладить программу и представить для проверки преподавателю.

Задание №9.

1. Создать прибор позволяющий производить пересчет мощности представленной в лошадиных силах паровых в лошадиные силы международные и ваты(1 лошадиная сила паровая = 9803 ватта, 1 лошадиная сила международная

- =735.499 ватта) по выбору оператора. Входное значение мощности имитировать с помощью произвольных регуляторов с вкладки Numeric меню Controls.
2. Отладить программу и представить для проверки преподавателю

Задание №10.

1. Создать прибор позволяющий производить пересчет давления представленного в килоПаскалях в атмосферы и миллиметры ртутного столба (1 атмосфера = 0.76 метра ртутного столба, 1 атмосфера =101.325 килоПаскаля) по выбору оператора. Входное значение давления имитировать с помощью произвольных регуляторов с вкладки Numeric меню Controls.
2. Отладить программу и представить для проверки преподавателю.

Задание №11.

1. Создать прибор позволяющий производить пересчет урожая пшеницы представленного в пудах в килограммы и центнеры (1 пуд =16.3804964 килограмма, 1 центнер =100 килограмм) по выбору оператора. Входное значение веса урожая имитировать с помощью произвольных регуляторов с вкладки Numeric меню Controls.
2. Отладить программу и представить для проверки преподавателю.

Задание №12.

1. Создать прибор позволяющий производить расчет действующего и амплитудного значения переменного напряжения по известному значению средневыпрямленного напряжения ($K_{\text{формы}}=1.11$, $K_{\text{амплитуды}}=1.41$) по выбору оператора. Входное значение средневыпрямленного напряжения имитировать с помощью произвольных регуляторов с вкладки Numeric меню Controls.
2. Отладить программу и представить для проверки преподавателю.

Задание №13.

1. Создать прибор позволяющий производить пересчет площади представленной в акрах в гектары и десятины казённые (1 акр =0.4046856422 гектара, 1 акр =0.3702693166 десятины казённой)по выбору оператора. Входное значение площади имитировать с помощью произвольных регуляторов с вкладки Numeric меню Controls.
2. Отладить программу и представить для проверки преподавателю.

Задание №14.

1. Создать прибор позволяющий производить пересчет роста человека представленного в метрах в вершки и аршины (1 вершок =4.445 сантиметра, 1 аршин русский =7.112 дециметра) по выбору оператора. Входное значение роста имитировать с помощью произвольных регуляторов с вкладки Numeric меню Controls.
2. Отладить программу и представить для проверки преподавателю.

Задание №15.

1. Создать прибор позволяющий производить пересчет роста человека представленного в футах в метры и дюймы (1 фут = 12 дюймов, 1 фут =30.48 сантиметра) по выбору оператора. Входное значение роста имитировать с помощью

- произвольных регуляторов с вкладки Numeric меню Controls.
2. Отладить программу и представить для проверки преподавателю.

Задание №16.

1. Создать прибор позволяющий производить пересчет количества информации представленного в килобайтах в байты и Мегабайты (1 Мегабайт=1024 килобайта, 1 килобайт=1024 байта) по выбору оператора. Входное значение количества информации имитировать с помощью произвольных регуляторов с вкладки Numeric меню Controls.
2. Отладить программу и представить для проверки преподавателю.

Задание №17.

1. Создать прибор позволяющий производить расчет средневыпрямленного и амплитудного значения переменного тока по известному действующему значению тока ($K_{\text{формы}}=1.11$, $K_{\text{амплитуды}}=1.41$) по выбору оператора. Входное действующее значение тока имитировать с помощью произвольных регуляторов с вкладки Numeric меню Controls.
2. Отладить программу и представить для проверки преподавателю.

Задание №18.

1. Создать прибор позволяющий производить пересчет количества электроэнергии представленной киловатт-часах в киловатт-секунды и ватт-дни по выбору оператора. Входное значение количества электроэнергии имитировать с помощью произвольных регуляторов с вкладки Numeric меню Controls.
2. Отладить программу и представить для проверки преподавателю

ЛАБОРАТОРНАЯ РАБОТА №3

ГРАФИЧЕСКОЕ ОТОБРАЖЕНИЕ ИНФОРМАЦИИ В LABVIEW

1. Цель работы: Изучение практических приемов и особенностей использования различных средств отображения графической информации в LabView.

2. Описание лабораторного макета

В ходе лабораторной работы используется инструментарий пакета LabView 8.2.

3. Подготовка к проведению лабораторной работы. Задание на самостоятельную работу

В ходе самостоятельной работы студенты должны изучить материал «Средства отображения графической информации в LabView». Особое внимание обратить на способы и примеры применения Waveform Chart, X-Y Graph, Mixed Signal Graph.

3.1 Средства отображения графической информации в LabView

Работу инженера по наладке аппаратуры существенно упрощает осциллограф. Объяснение любого материала существенно облегчается, если он сопровождается графическим материалом (картинки, графики, диаграммы и пр.). Разработчики LabView создали мощный аппарат графического представления информации. Они много сделали, чтобы внимание разработчика было направлено на решение собственной задачи, а вывод результатов обеспечивался как бы сам собой. Все сделано настолько удобно, что иногда бывает быстрее и эффективнее вывести любые промежуточные значения на график, чем разобраться, почему программа работает неправильно.

В LabView имеются различные виды графиков: Waveform Chart, Waveform Graph, X-Y Graph, Mixed Signal Graph и другие. Рассмотрим некоторые из них.

Waveform Chart относится к самым простым средствам отображения информации. Пользователю достаточно подключить Waveform Chart к выходу своей программы. Здесь ось X организована очень примитивно, типа самописца, так называемый ленточный график.

Есть другой тип графиков, который выводит сигнал на отображение не поточечно, а только массивом точек. Это график Waveform Graph (график сигнала). Здесь разработчики также облегчили задачу пользователя. Они сами организовали ось X. Имея массив сигнала, пользователь может сразу использовать такой график.

Третий тип графиков X-Y Graph (график функциональной зависимости $y=f(x)$) требует для своей работы создания отдельного массива отображаемого сигнала (Y) и отдельного массива его аргумента (X). Такой график нельзя непосредственно подключать к источнику сигнала, требуется определенная организация массивов X и Y.

Вызов и установка графиков осуществляется на лицевой панели из набора Controls→ Graph.

3.2 Использование Waveform Chart для отображения информации

Waveform Chart – наиболее простой график. Подключенный к любому источнику цифровой информации, он позволяет в реальном масштабе времени отразить

существующее в данный момент числовое значение. Это подобно самопишущему устройству, в котором лента движется с некоторой скоростью по горизонтальной оси, а график отображает положение "пера" на вертикальной оси. Waveform Chart единственный вид графиков, который может отображать информацию поточечно, он может отображать и массивы, но принцип самописца остается.

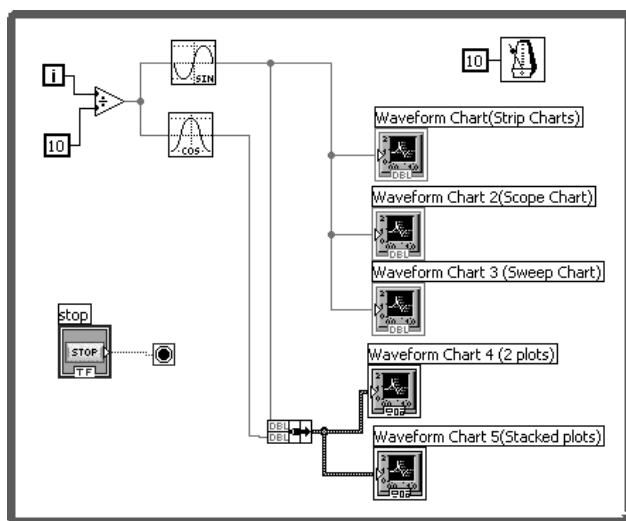
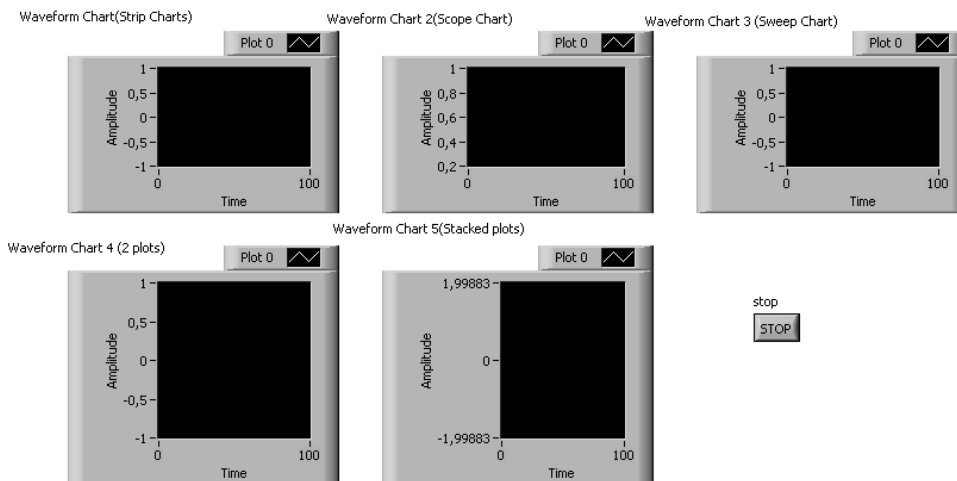
Как и у всех других приборов можно менять размеры графика курсором позиционирования. Однако с точки зрения установления размеров следует различать панель графика (вместе со шкалами, маркерами, метками шкал) и окно графика (с отображаемыми сигналами). Размеры панели и размеры окна меняются отдельно друг от друга. При изменении размера панели размер окна тоже меняется. При этом также изменяют свое положение все установленные атрибуты графика. Но изменение самого окна на другие элементы не влияет.

У графиков можно задавать пределы по осям текстовым курсором. Подведите курсор к минимальному или максимальному пределу, щелкните на нем левой кнопкой мышки, и введите требуемое значение предела. Основная работа с графиком и его атрибутами осуществляется через контекстное меню самого графика и контекстные меню его атрибутов, установленных на передней панели.


График Waveform Chart использует три различных режима отображения данных: панорамирование диаграммы (**strip chart**), временная развертка (**scope chart**) и временная развертка с маркером (**sweep chart**). Режим по умолчанию - панорамирование диаграммы (**strip chart**).

Выбор режима осуществляется щелчком правой клавишей мыши по диаграмме, затем выбором пункта **Advanced>Update Mode** из контекстного меню. Режим панорамирования диаграммы (**strip chart**) представляет собой экран, прокручиваемый слева направо, подобно бумажной ленте. Режимы временная развертка (**scope chart**) и временная развертка с маркером (**sweep chart**) подобны экрану осциллографа и отличаются большей скоростью отображения данных по сравнению с режимом панорамирования диаграммы (**strip chart**). В режиме временная развертка (**scope chart**) по достижении правой границы поле графика очищается и рисование диаграммы начинается с левой границы. Режим временная развертка с маркером (**sweep chart**), в отличие от режима временная развертка (**scope chart**), не очищает поле графика, а текущее значение диаграммы обозначается вертикальной линией - маркером.

Для более детального знакомства с Waveform Chart разработаем новый виртуальный прибор. Создайте новый пустой виртуальный прибор, выбрав команду **Blank VI** после запуска LabView. Разместите на лицевой панели 5 окон с Waveform Chart, выбрав их из меню **Controls-> Graph** и кнопку **Stop Button** (**Controls, Modern, Boolean**). Для первой панели выберите свойство - **strip chart**, для второй - **scope chart**, а для третьей - **sweep chart**. Для пятой панели в свойствах выберите опцию **Stack Plots**. На панели диаграмм разместите цикл **While Loop** и поместите пиктограммы всех пяти Waveform Chart во внутрь цикла.



Кнопку Stop подключите к условию окончания цикла. Для генерации отображаемых сигналов воспользуемся номером итерации i цикла While Loop. Из палитры Functions > Mathematics > Elementary > Trigonometric выберите функции  и  (синус и косинус соответственно). Для замедления

работы цикла вставьте  блок задержки из палитры Functions > Programming > Timing.

На 1,2 и 3 окно Waveform Chart подайте сигнал с выхода формирователя синусоиды, а на вход 4 и 5 окна Waveform Chart подайте сигнал от формирователей синусоиды и косинусоиды пропущенный через функцию Объединение (Bundle) (Functions > Cluster & Variant). В результате разработки программы у Вас должны получиться картинки сходные с представленными на этой странице.

Отладьте программу и запустите ее на выполнение проанализируйте различия в представлении информации в разных режимах работы Waveform Chart.

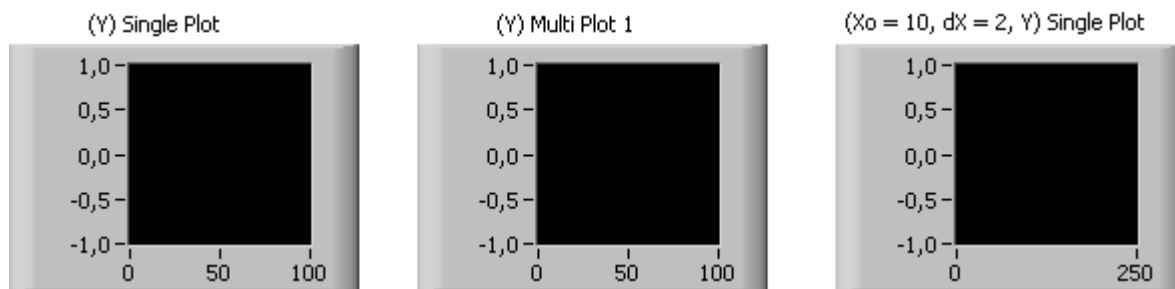
3.3 Использование Waveform Graph и X-Y Graph для отображения информации



График осциллограмм (Waveform Graph) отображает только однозначные функции, такие как $y = f(x)$, с точками, равномерно распределенными по оси X. Двухкоординатный график осциллограмм (X-Y Graph) отображает любой набор точек, будь то равномерно распределенная выборка или нет.

График осциллограмм (Waveform Graph) и двухкоординатный график осциллограмм (X-Y Graph) автоматически поддерживают режим отображения множества осциллограмм.

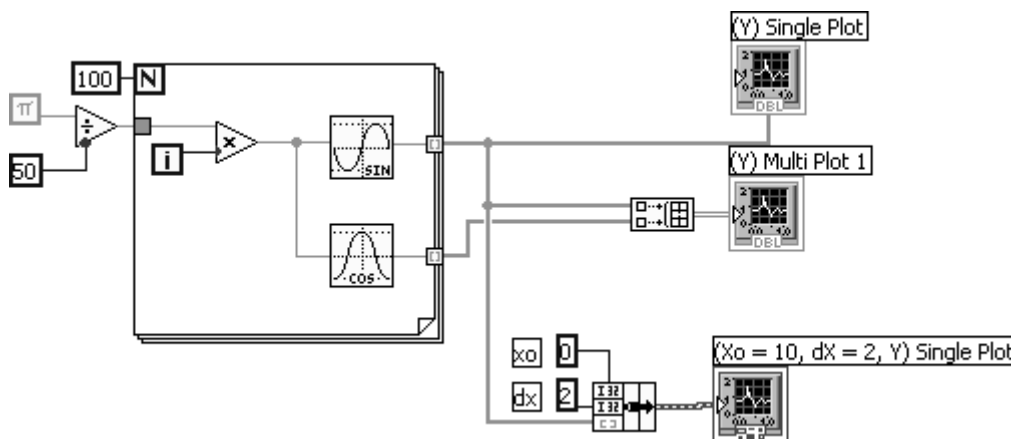
Одиночный график осциллограмм (Waveform Graph) работает с одномерными массивами и представляет данные массива в виде точек на графике, с приращением по оси X равным «1» и началом в $x=0$. Графики также отображают данные кластеров, с установленным начальным значением x , Δx и массивом данных по шкале y .

Рассмотрим пример использования Waveform Graph. Для этого создайте новый виртуальный прибор. На лицевой панели разместите три окна Waveform Graph. На



или диаграмм разместите цикл For Loop внутри которого расположите функции  и  (синус и косинус соответственно), которые в цикле будут воспринимать

номер
цикла
ровать
мент.
цикле
СМОТ-
шагов
в каче-
ного
значе-
поль-



итерации
и форми-
один эле-
Всего в
предусмот-
рите 100
(итераций)
стве вход-
для цикла
ния
зуйте

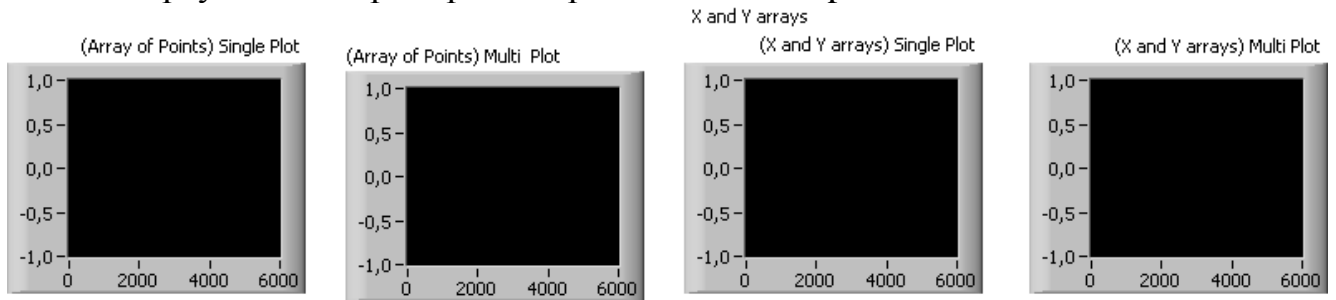
дробную часть (например, одну пятидесятую от числа пи). Таким образом, будут сформированы два массива чисел, описывающих один период синусоиды и косинусоиды, состоящих из ста чисел каждый. Сигнал с выхода формирователя синусоиды подайте на вход первого окна Waveform Graph, на вход второго окна Waveform Graph подайте сигналы с выходов формирователя синусоиды и косинусоиды, предварительно пропустив их через функцию Построение массива (Build Array) (Functions > Programming > Array). На вход третьего окна подайте сигнал с выхода формирователя синусоиды предварительно объединив этот сигнал с помощью функции Объединение Bundle (Functions > Programming > Cluster & Variant) с начальным значением по оси X_0 (например, $X_0=10$) и шагом изменения данных по оси ΔX (например, $\Delta X=2$).

Запустите разработанный виртуальный прибор на выполнение. Проанализируйте изменения в изображении графика в третьем окне если Вы будете изменять значения X_0 и ΔX .



Двухкоординатные графики (X-Y Graph) универсальны, они предназначены для отображения многозначных функций в декартовой системе координат (замкнутые кривые, распределение осциллограммы во времени с переменной временной базой).

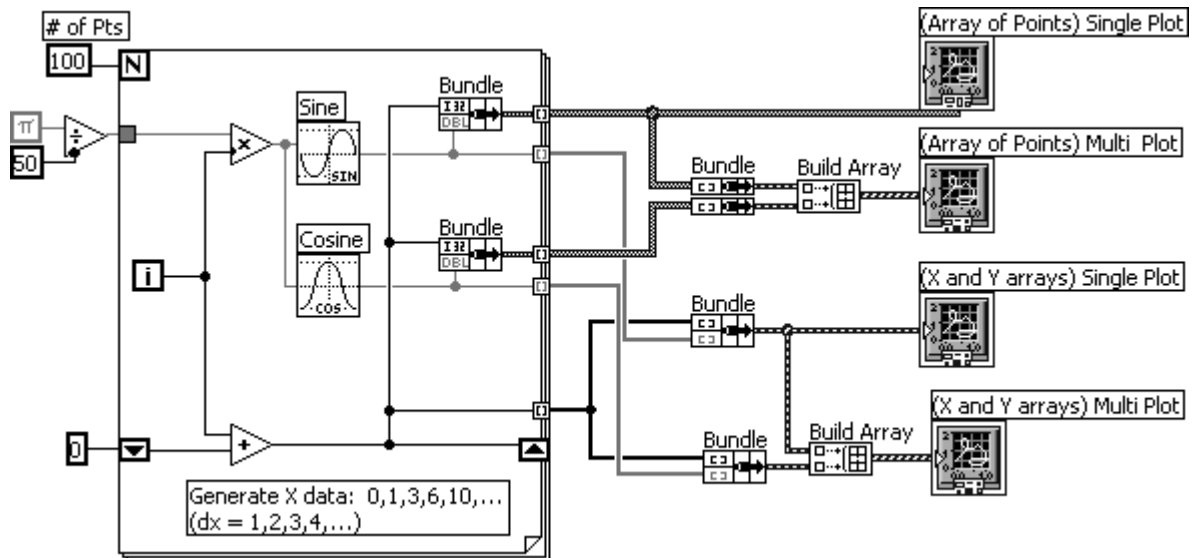
Одиночный двухкоординатный график осциллограмм (XY graph) работает с группами данных, содержащими массивы x и y . Двухкоординатный график осциллограмм (XY graph) также воспринимает массивы точек, где точки - группы данных, содержащие значения по шкалам x и y .

Двукоординатные графики множества осциллограмм работают с массивами осциллограмм, в которых осциллограмма данных - кластер, содержащий массивы значений x и y . Двукоординатные графики множества осциллограмм воспринимают также массивы множества осциллограмм, где каждая осциллограмма представляет собой массив точек. Каждая точка это группа данных, содержащая значения по x и y . В качестве примера рассмотрим способы построения графиков с использованием двукоординатного графика (X-Y Graph), для этого разместите на передней панели нового виртуального прибора четыре окна X-Y Graph.



На панели диаграмм разместите цикл For Loop внутри которого расположите

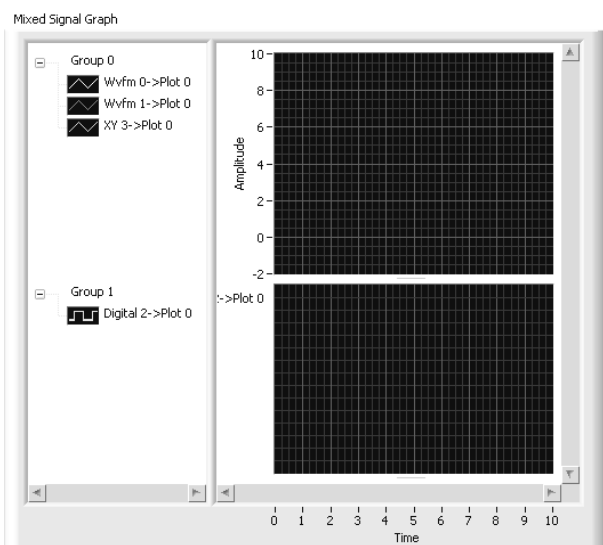
функции  и  (синус и косинус соответственно), которые в цикле будут воспринимать номер итерации цикла и формировать один элемент. Всего в цикле предусмотрите 100 шагов (итераций) в качестве входного для цикла значения используйте дробную часть (например, одну пятидесятую от числа π). Таким образом, будут сформированы два массива чисел, описывающих один период синусоиды и косинусоиды, состоящих из ста чисел каждый. Кроме этого, в цикле сформируем массив чисел с переменным шагом. Для этого установите на левой или правой границе цикла курсор мышки и вызовите контекстное меню, в котором выберите опцию



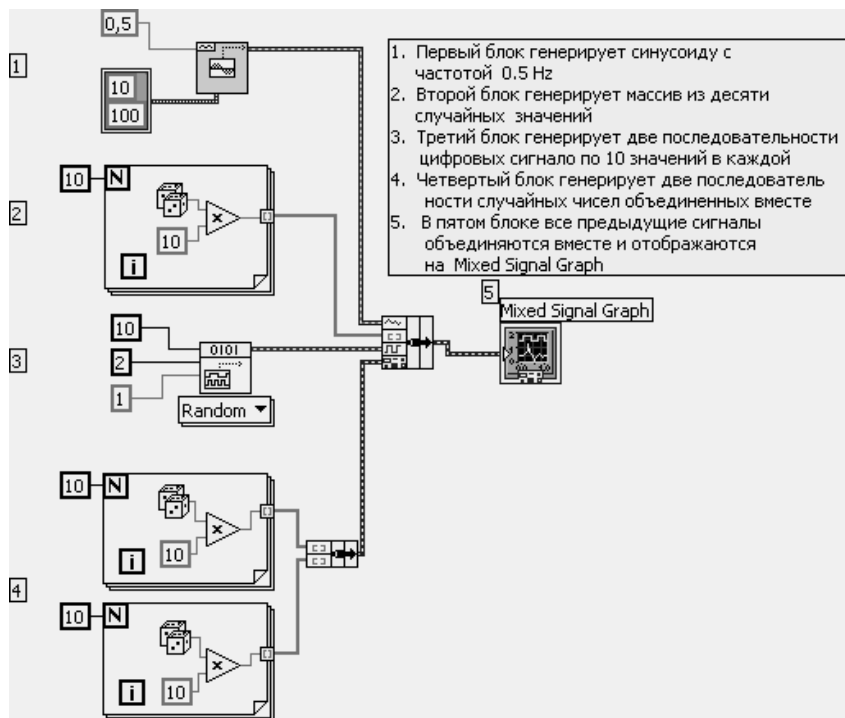
сдвигового регистра Add Shift Register. Подключите все элементы так же, как на представленном рисунке с использованием элементов Build Array и Bundle. Запустите разработанный виртуальный прибор на выполнение.

3.4 Использование Mixed Signal Graph для отображения информации

Mixed Signal Graph позволяет в одном окне выводить как цифровую, так и аналоговую информацию, организованную разнообразными способами. Создайте новый виртуальный прибор, разместите на его лицевой панели Mixed Signal Graph, а на панели диаграмм соберите схему, состоящую из элементов представленных на рисунке ниже.



Кроме известных Вам элементов на схеме используются дополнительно такие элементы. В первом блоке элемент формирования синусоидальных сигналов на вход frequency которого подано значение 0,5 Гц, а на вход sampling info поданы значения: частота формирования отсчетов в секунду 10 и общее количество отсчетов в сгенерированном сигнале 100 (по умолчанию оба значения 1000). Этот



элемент (Sine Wfm) расположен в палитре Functions> Signal Processing> Wfm Generation. В третьем блоке используется элемент генерирования цифровых сигналов (Pattern Generator) расположенный в палитре Functions> Programming> Waveform> Digital Wfm> Pattern Gen. На входах этого элемента установлено количество формируемых отсчетов (10), число формируемых цифровых сигналов (2) и частота формирования сигнала (1, по умолчанию 1000 Гц). Свойство ramp (Pattern Generator) замените на random (будут формироваться случайные цифровые сигналы).

Сформируйте и запустите на выполнение предложенный прибор. Убедитесь, что на графиках представлены все 5 сигналов (три аналоговых - 1, 2 и 4 блоки, и два цифровых сигнала – 3 блок).

При выполнении лабораторной работы используйте возможности системы Help среды LabView.

4. Ход работы

- 1) Создайте виртуальные приборы описанные в третьем разделе.
- 2) Используя контекстные меню графиков измените формы представления

графической информации, цвет, толщину линий и пр.

3) Найдите в контекстном меню пункт позволяющий производить очистку окна отображения информации.

4) Составить индивидуальный отчет о проделанной работе, который должен включать в себя:

- ✓ Лицевую панель разработанных приборов, с отображаемой информацией;
- ✓ Панель диаграмм разработанных приборов

5) Защита отчета проходит индивидуально для каждого студента.

5. Контрольные вопросы:

- 1) Какие средства отображения графической информации имеются в LabView?
- 2) Чем отличается Waveform Chart от Waveform Graph?
- 3) Какие особенности имеются при использовании X-Y Graph?
- 4) Для чего используется Mixed Signal Graph