

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
**Федеральное государственное автономное образовательное
учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Методические указания
к контрольной работе
по дисциплине "Моделирование в среде Labview"

Направление подготовки 13.03.02 Электроэнергетика и
электротехника Направленность (профиль) «Электроснабжение»
Квалификация выпускника бакалавр

Ставрополь
2026

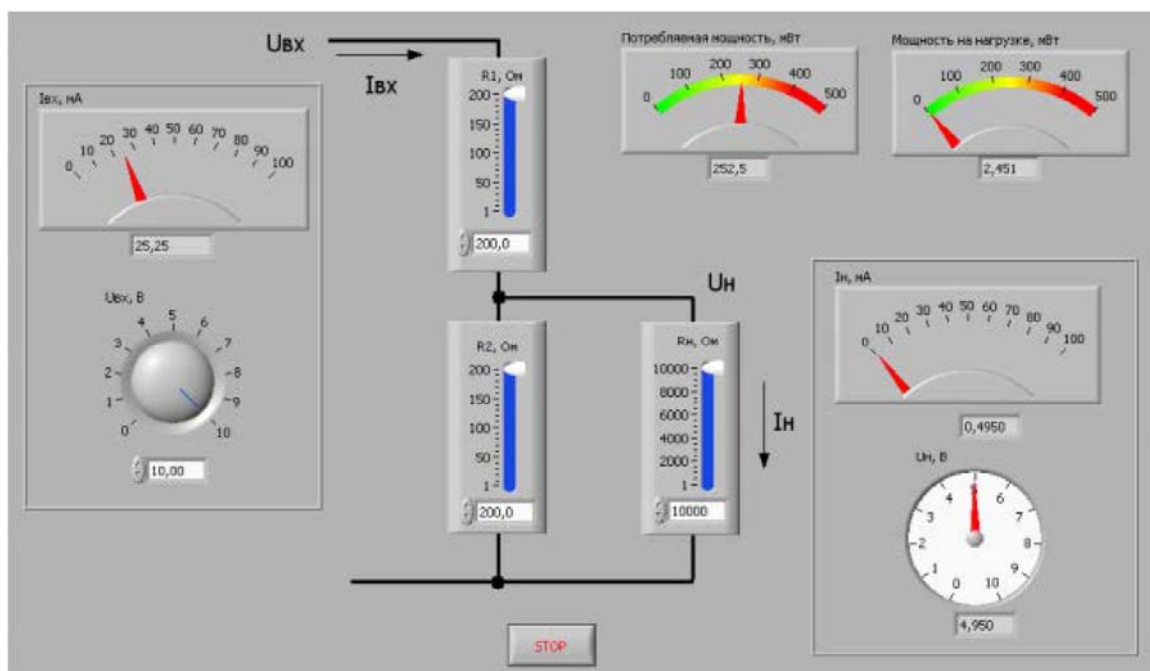
Введение

Создание виртуальных приборов в настоящее время сопровождается процессом разработки современных информационно-измерительных систем в учебном процессе и на производстве. Данный курс работ посвящен проектированию виртуальных приборов в среде LabView. Технология виртуальных приборов находится в постоянном развитии за счет развития компьютерной техники и контрольно-измерительного оборудования. Виртуальные приборы позволяют создавать гибкие и легко адаптируемые к новым условиям системы, которые соответствуют требованиям надежности, точности и производительности. На сегодняшний день наиболее мощным и надежным средством разработки виртуальных приборов является среда графического программирования LabView компании National Instruments, предлагающая наиболее широкий спектр инструментов для работы с различного рода промышленным и контрольно-измерительным оборудованием.

Задание для самостоятельной работы студента.

«Делитель напряжения»

Создайте виртуальный прибор, моделирующий работу делителя напряжения, состоящего из двух резисторов R_1 и R_2 . К выходу делителя подключена нагрузка сопротивлением R_n . Значение сопротивлений, а также входное напряжение делителя $U_{вх}$ можно задавать при помощи элементов управления. Индикаторы показывают входной ток делителя $I_{вх}$, напряжение на нагрузке U_n и ток через нее I_n . Два индикатора показывают мощность, потребляемую делителем $P_{вх} = U_{вх}I_{вх}$ и мощность, передаваемую нагрузке $P_n = U_nI_n$



Указание: Для вычисления токов и напряжений используйте закон Ома

$$I = \frac{U}{R} \quad (1)$$

где I – ток через участок электрической цепи, U – разность потенциалов на концах этого участка, R – электрическое сопротивление этого участка цепи.

Сопротивление двух последовательно соединенных резисторов R_1 и R_2 вычисляется по формуле

$$R = R_1 + R_2 \quad (2)$$

а сопротивление двух резисторов, соединенных параллельно – по формуле

$$R = \frac{R_1 \cdot R_2}{R_1 + R_2} \quad (3)$$

Ограничьте точность отображения значений на цифровых индикаторах четырьмя значащими цифрами (Significant digits), используя контекстное меню Display Format...

Проверку состояния элементов управления организуйте, используя цикл While Loop с задержкой 100 мс.

AB – номер варианта

Амплитуда напряжения выбирается по первой цифре варианта «A».

Угол Alfa выбирается по второй цифре номера варианта.

Таблица 1 – Параметры периодического сигнала тока I

A	U	B	Alfa
0	100	0	30
1	110	1	45
2	120	2	60
3	130	3	75
4	140	4	90
5	150	5	105
6	160	6	120
7	170	7	135
8	180	8	150
9	190	9	165

Сопротивления делителя напряжения:

$R_1 = 1000 \text{ Ом}$, $R_2 = 10 \text{ Ом}$, сопротивление нагрузки $R_H = AB \text{ Ом}$ равняется номеру варианта.

Наименование лекций

1 Лекция 1. «Введение в LabView»

Программная среда LabView. Понятие Виртуального Прибора (ВП). Последовательность обработки данных. Организация программной среды LabView.

Справочная система среды LabView.

2 Лекция №2 «Создание ВП»

- компоненты ВП
- создание ВП
- типы данных и проводники данных
- редактирование ВП
- отладка ВП

3 Лекция №3 «Создание подпрограмм ВП»

Подпрограмма ВП:

- создание иконки ВП и настройка соединительной панели;
- использование виртуального прибора в качестве подпрограммы;
- преобразование экспресс-ВП в подпрограмму ВП;
- превращение выделенной секции блок-диаграммы ВП в подпрограмму

ВП.

4 Лекция №4 «Многократные повторения и циклы. Массивы.»

Использование цикла While (по условию)

Использование цикла For (с фиксированным числом итераций).

Массивы:

- что такое массив;
- создание массивов с помощью цикла;
- использование функций работы с массивами;
- полиморфизм.

5 Лекция №5 «Принятие решений в ВП»

Принятие решений в ВП:

- функция Select и принятие решений;

- использование структуры Case;
- использование узла Формулы.

6 Лекция №6 «Сбор и отображение данных»

Тема 1. Введение и конфигурация

- сбор данных в LabView;
- выполнение операций аналогового ввода;
- запись полученных данных в файл;
- выполнение операций аналогового вывода;
- информация о счетчиках;
- информация о цифровых линиях ввода/вывода.

Тема 2. Введение и конфигурация NI-DAQ:

- Стандартный NI-DAQ;
- NI-DAQmx;
- Настройка аппаратных средств DAQ-устройств.

Тема 2. Measurement & Automation Explorer

- Тестирование компонент DAQ-устройства

Тема 3. Ввод аналогового сигнала с помощью DAQ-устройства:

- ВП Монитор напряжения;
- Измерение среднего значения;
- простая запись данных в файл

Литература

1. Суранов, Александр Яковлевич. LabVIEW 7: справочник по функциям. – М.: ДМК Пресс, 2005. – 510 с.
2. Тревис, Джеффри. LabVIEW для всех: пер. с англ. – М.: ПриборКомплект: ДМК Пресс, 2005. – 537 с.
3. Евдокимов, Юрий Кириллович. Labview для радиоинженера: от виртуальной модели до реального прибора: учебное пособие для вузов. – М.: ДМК Пресс, 2007. – 400 с
4. Учебный курс LabVIEW ОСНОВЫ I. Авторский коллектив: Михеев П.К., Крылова С.И., Лукьянченко В.А., Урюпина Д.С. Учебно-технический центр «Систем автоматизации научных исследований» Международный учебно-научный лазерный центр, Московский государственный университет им, Март 2007. – 365 с.
5. <http://e-lib.kemtipp.ru/uploads/25/eteo156.pdf> - Учебный курс Labview.

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ**
**Федеральное государственное автономное образовательное учреждение
высшего образования**
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Моделирование в среде Labview

**Методические указания к лабораторным работам
(лабораторный практикум)**

Направление подготовки 13.03.02 Электроэнергетика и электротехника
Направленности (профили) «Электроснабжение», Квалификация
выпускника бакалавр
Изучается в 4 семестре

Ставрополь
2026

Оглавление

Введение	5
Лабораторная работа 1	6
Элементы ввода вывода в LabVIEW	6
Лабораторная работа 2	27
Работа с циклами и графический вывод данных ВП Монитор напряжения	27
Лабораторная работа 3	43
Работа с массивами данных, простая запись данных в файл	43
Лабораторная работа 4	59
Измерение среднего значения	59
Лабораторная работа 5	70
ВП Кофейный аппарат	70
Лабораторная работа 6	84
Введение и конфигурация NI-DAQmx, сбор данных	84
Лабораторная работа 7	97
Настройка аппаратных средств DAQ-устройств	97
Лабораторная работа 8	111
Калибровка генерации сигнала средствами DAQ-устройств	111
Лабораторная работа 9	124
Ввод аналогового сигнала с помощью DAQ-устройства его преобразование и передача по цифровому каналу	124
Литература	141
Приложение 1	142

Введение

Настоящее учебное пособие предназначено для проведения лабораторного практикума по дисциплине «Моделирование в среде Labview» у студентов, обучающихся по направлению 13.03.02 «Электроэнергетика и электротехника». Дисциплина «Моделирование в среде Labview» является дисциплиной, призванной формировать обще профессиональные компетенции бакалавров – электроэнергетиков. Лабораторный практикум рассчитан на закрепление и расширение знаний, полученных при изучении наиболее трудоёмких и сложных теоретических разделов дисциплины. Целью выполнения лабораторных работ является изучение инструментария LabView для проектирования приборов, моделирование элементов цифровой электроники, изучение принципы работы электронных измерительных устройств и работа с текстовыми данными и аналоговыми измерительными устройствами с программным управлением и цифровой обработкой данных. Учебное пособие разработано в соответствии с федеральным государственным образовательным стандартом высшего образования, рабочим учебным планом и рабочей программой дисциплины «Моделирование в среде Labview».

ЛАБОРАТОРНАЯ РАБОТА 1

Элементы ввода вывода в LabView

Цель работы: ознакомление с программным пакетом LabView. В связи с этим необходимо решить следующие задачи:

1. Выполнить построение кода согласно заданию.
2. Проверка работоспособности кода вычислением заданных математических действий, индивидуализация блок диаграммы изменением размера шрифтов и цвета элементов.

Теоретическая часть

Основные элементы структуры. LabView, в отличие от языков программирования C, PASCAL или BASIC, использует графический язык программирования, предназначенный для создания программ в форме структурных схем. LabView содержит обширные библиотеки функций и инструментальных средств, предназначенных для создания систем сбора данных и систем автоматизированного управления. LabView также включает стандартные инструментальные средства разработки программ.

Программы в LabView называются виртуальными приборами(VI, Virtual instrument – англ.), так как их вид и функционирование имитируют реальные измерительные приборы. Однако, при этом виртуальные приборы подобны функциям в программах стандартных языков программирования.

Структура виртуального прибора может быть представлена следующими элементами:

- **лицевой панелью** (лицевая панель может содержать кнопки, переключатели, регуляторы и другие органы управления и индикаторы);
- **структурной схемой** (структурная схема представляет собой наглядное представление решения задачи и содержит исходные коды для виртуального прибора.

LabView придерживается концепции модульного программирования. Можно разделить прикладную программу на несколько более простых подпрограмм, а затем создать несколько виртуальных приборов для выполнения каждой подпрограммы и объединить эти виртуальные приборы на общей структурной схеме, выполняющей основную программу. В результате основной виртуальный прибор верхнего уровня будет содержать совокупность суб-приборов (subVI), которые смогут реализовать функции прикладной программы. Многие subVI низкого уровня часто выполняют задачи, общие для нескольких прикладных программ, так что можно разработать специализированный набор subVI, хорошо подходящий для прикладных программ, которые будут созданы в дальнейшем.

Лицевая панель. Лицевая панель виртуального прибора – прежде всего комбинация органов управления и индикаторов. Органы управления моделируют инструментальные устройства ввода данных и передают данные на структурную схему виртуального прибора. Индикаторы моделируют инструментальные устройства вывода, которые отображают данные, собранные или сгенерированные структурной схемой виртуального прибора.

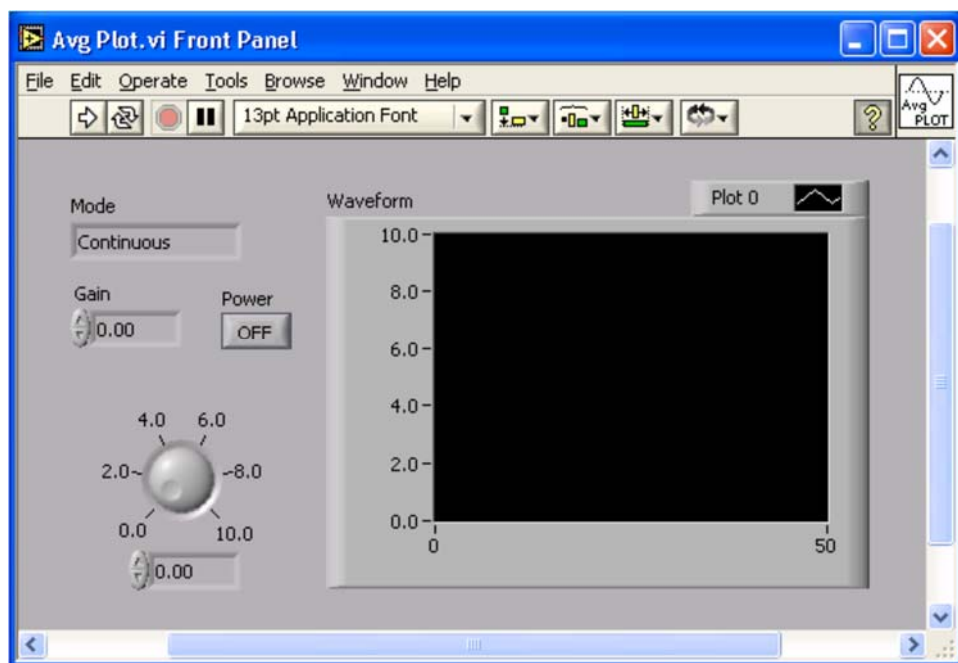


Рисунок 1.1 – Лицевая панель виртуального прибора.

Структурная схема (блок диаграмма). Окно схемы содержит структурную схему виртуального прибора, которая является исходным графическим текстом виртуального прибора в LabView. Структурная схема создается посредством соединения объектов, которые посылают или получают данные, выполняют определенные функции и управляют потоком выполнения.

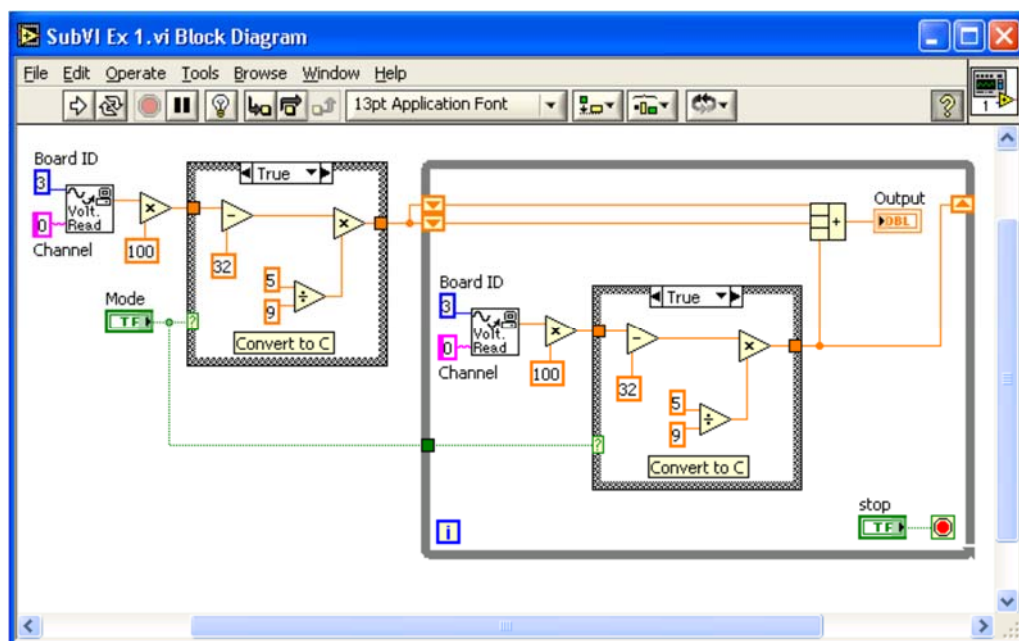


Рисунок 1.2 – Структурная схема (блок диаграмма).

Первичные программные объекты структурной схемы – узлы, терминалы и провода.

При появлении органа управления или индикатора на лицевой панели, LabView помещает соответствующий терминал на структурную схему. Пиктограммы функций также имеют терминалы. Данные, которые вводятся в органы управления, поступают с лицевой панели через терминалы органов управления на структурную схему. Затем данные поступают в функции. Когда функции завершают свои внутренние вычисления, они производят новые значения данных на своих выходных терминалах. Данные поступают на терминалы индикаторов и повторно попадают на лицевую панель, где они и отображаются.

Узлы – элементы выполнения программы. Они аналогичны инструкциям, операторам, функциям и подпрограммам в стандартных языках

программирования. **Функция** – один из типов узлов. LabView имеет обширную библиотеку функций для математических вычислений, сравнения, преобразования, ввода/вывода и так далее. Другой тип узлов – **структура**. Структуры являются графическим представлением циклов и операторов выбора традиционных языков программирования, повторяя блоки инструкций или выполняя их по условию.

LabView имеет также специальные узлы для взаимосвязи с внешними текстовыми программами и для вычислений по текстовым формулам.

Провода – пути данных между терминалами источника и адресата. Нельзя подключить терминал-источник к другому источнику, нужно подключать терминал-адресат к другому терминалу-адресату. Можно подключать один источник к нескольким адресатам. Провода имеют различный вид или цвет, в зависимости от типа данных, которые по ним передаются.

Принцип, который управляет выполнением программы в LabView, называется потоком данных. Запущенный узел выполняется только тогда, когда на всех входах появляются данные; узел выдает данные на все выходные терминалы только тогда, когда он заканчивает выполнение; и данные сразу же поступают от терминала источника на терминал адресата. Поток управления регулируется командами. Поток данных – управляется данными или зависит от данных.

Когда пиктограмма виртуального прибора помещена в схему другого виртуального прибора, первый виртуальный прибор становится subVI, то есть подпрограммой в LabView. Органы управления и индикаторы subVI получают данные от вызывающего виртуального прибора и возвращают их ему же.

Инструменты используются для выполнения определенных функций. Многие из инструментов LabView содержатся в палитре «**Tools**» (таблица 1.1). Изначально при создании нового виртуального прибора либо загрузке существующего на экране появляется окно лицевой панели.

Перейти к окну лицевой панели из окна структурной схемы можно, выбрав в меню Windows>>Show Panel.

Объекты на лицевой панели создаются при выборе их из палитры «Controls»(Windows>>Show Controls Palette).

При создании объекты лицевой панели появляются с прямоугольником метки, в которую сразу же можно ввести текст – название органа управления или индикатора.

Объектное меню вызывается нажатием правой кнопки мыши, когда курсор в виде руки или стрелки находится на объекте.

Созданная метка объекта редактируется меточным инструментом из палитры «Tools».

Таблица 1.1 – Инструменты палитры «Tools» [4]

	Если включен автоматический выбор инструмента, то при наведении курсора на объект лицевой панели или блок-диаграммы LabVIEW автоматически выбирает соответствующий инструмент из палитры Tools (Инструментов). Автоматический выбор инструментов включается нажатием на кнопку Automatic Tool Selection палитры Tools (Инструментов) или нажатием клавиш <Shift-Tab>.[4]
	Инструмент УПРАВЛЕНИЕ используется для изменения значения элементов управления или ввода текста. При наведении курсора на такой элемент как строковый элемент управления, значок инструмента меняется:  [4]
	Инструмент ПЕРЕМЕЩЕНИЕ используется для выбора, перемещения или изменения размеров объектов. При наведении инструмента на объект изменяемого размера значок инструмента меняется: [4] 
	Инструмент ВВОД ТЕКСТА используется для редактирования текста и создания свободных меток. При создании свободных меток значок инструмента меняется: 
	Инструмент СОЕДИНЕНИЕ создает проводники данных, соединяя объекты на блок-диаграмме. [4]
	Инструмент ВЫЗОВ КОНТЕКСТНОГО МЕНЮ вызывает контекстное меню соответствующего объекта по щелчку левой кнопки мыши. [4]
	Инструмент БЫСТРАЯ ПРОКРУТКА ЭКРАНА используется для просмотра окна без использования полосы прокрутки. [4]

	Инструмент ВВОД КОНТРОЛЬНОЙ ТОЧКИ позволяет расставлять контрольные точки на ВП, функциях, узлах, проводниках данных, структурах и приостанавливать в них выполнение программы. [4]
	Инструмент УСТАНОВКА ОТЛАДОЧНЫХ ИНДИКАТОРОВ дает возможность исследовать поток данных в проводниках блок-диаграммы. Используется для просмотра промежуточных значений при наличии сомнительных или неожиданных результатов работы ВП.
	Инструмент КОПИРОВАНИЕ ЦВЕТА предназначен для копирования цвета с последующей вставкой с помощью инструмента РАСКРАШИВАНИЕ . [4]
	Инструмента РАСКРАШИВАНИЕ позволяет изменить цвет объекта. Он также отображает текущий передний план и параметры настройки цвета фона. [4]
	Если автоматический выбор инструмента выключен, можно менять инструменты палитры Tools (Инструментов) с помощью клавиши <Tab> . Для переключения между инструментом ПЕРЕМЕЩЕНИЕ и СОЕДИНЕНИЕ на блок-диаграмме или между инструментом ПЕРЕМЕЩЕНИЕ и УПРАВЛЕНИЕ на лицевой панели – достаточно нажать пробел. [4]

Типы и проводники данных

В среде LabVIEW проводники данных используются для соединения многочисленных терминалов данных. Поля ввода/вывода должны быть совместимыми с типами данных, передаваемыми им по проводникам. Например, нельзя соединять поле вывода массива с полем ввода данных численного типа. Кроме того, характер соединения должен быть корректным. Проводники должны быть подсоединены лишь к одному источнику данных и, по крайней мере, к одному полю ввода данных. Например, нельзя соединять 2 элемента отображения. Компонентами, определяющими совместимость соединения, являются тип данных элемента управления и/или отображения и тип данных поля ввода/вывода.

Типы данных

В данном курсе используются следующие типы данных:

- **Numeric** (численный тип)
- **Floating point** – число с плавающей запятой, отображается в виде оранжевых терминалов. Может быть представлено в виде **single** (32 bit), **double** (64-bit)

или **extended** (128-bit) **precision** (с одиночной, двойной или расширенной точностью). Число с плавающей запятой может быть комплексным. [4]

- **Integer** - целочисленный тип, отображается в виде голубых терминалов. Возможны три представления целых чисел: 8, 16 и 32 бита. Один бит может использоваться для знака числа, если это число является знаковым целым.
- **Boolean** - логический тип, отображается в виде зеленых терминалов. Логический тип может принимать только два значения: 0 (FALSE) или 1 (TRUE).
- **String** - строковый тип, отображается в виде розовых терминалов. Строковый тип данных содержит текст в ASCII формате.
- **Path** - путь к файлу, отображается в виде терминалов. Путь к файлу близок строковому типу, однако, LabVIEW форматирует его, используя стандартный синтаксис для используемой платформы.
- **Array** - массивы включают типы данных составляющих элементов и принимают соответствующий им цвет.
- **Cluster** - кластеры включают различные типы данных. Кластерный тип данных отображается коричневым цветом, если все его элементы численные, если же элементы кластера являются данными различных типов, он отображается розовым.
- **Waveform** - сигнальный тип данных является кластером элементов, содержащим данные, начальное значение времени и интервал времени между измерениями.
- **Dynamic** - динамический тип, отображается в виде темно-синих терминалов. Кроме данных сигнала, динамический тип содержит дополнительную информацию, например, название сигнала или дату и время его получения. Большинство экспресс-ВП принимают и/или возвращают данные динамического типа. Данные динамического типа можно направлять к любому элементу отображения или полю ввода, принимающему данные численного, логического или сигнального типа [4].

Проводники данных

Данные между объектами блок-диаграммы передаются по соединительным линиям – проводникам данных. Проводник данных аналогичен переменным в текстовых языках программирования. Каждый проводник данных имеет единственный источник данных, но может передавать их ко многим ВП и функциям. Проводники данных различаются цветом, стилем и толщиной линии, в зависимости от типа передаваемых данных. Примеры основных типов проводников данных представлены ниже [4].

Тип проводника данных	Одно значение	Одномерный (1D) массив	Двумерный (2D) массив	Цвет
Численный				Оранжевый (с плавающей точкой), Голубой (целочисленный)
Логический				Зеленый
Строковый				Розовый

Автоматическое соединение объектов проводниками данных

В среде LabVIEW объекты соединяются проводниками данных после их помещения на блок-диаграмму. В автоматическом режиме среда LabVIEW подключает те поля ввода/вывода данных, которые наиболее совместимы, несовместимые поля остаются несоединенными.

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

Корректировка параметров автоматического подключения проводников осуществляется через пункты главного меню

Tools>>Options>>Block Diagram [4].

Соединение объектов проводниками данных вручную

Соединение объектов проводниками данных вручную производится с помощью инструмента СОЕДИНЕНИЕ. После наведения инструмента СОЕДИНЕНИЕ на поле ввода или вывода данных на экране появляется подсказка, которую можно использовать для уточнения места подключения проводника.

Узел Формула

Синтаксис узла Формула аналогичен синтаксису текстовых языков программирования. Все переменные, объявленные в блоках программы, ограниченных скобками, доступны только в этих блоках. Все входные терминалы считаются переменными внешнего (самого крайнего) блока (не заключенного в скобки) и не могут быть объявлены еще раз в этом блоке. LabVIEW старается сопоставить переменные, объявленные во внешнем блоке (не заключенном в скобки), с выходным терминалом с тем же именем [4].

оператор (statement):

1. объявление переменной (variable-declaration).
2. присваивание (assignment).
3. составной оператор (compound-statement).
4. условный оператор (conditional-statement)
5. итеративный оператор (iterative-statement)
6. оператор переключения (switch-statement).
7. оператор управления (control-statement).

1. объявление переменной (variable-declaration):

1.1. идентификатор определителя типа (type-specifier identifier), [1]

1.2. идентификатор определителя типа (type-specifier identifier) список индексов массива (array-index-list),

1.3. идентификатор определителя типа (type-spec identifier) =
присваивание (assignment).

1.1. определитель типа (type-specifier):

1.1.1. тип с плавающей точкой (floating-point-type): float, float32,
float64,

1.1.2. целый тип (integer-type): int, int8, int16, int32, uint8, uint16, uint32.

[1]

1.2. список индексов массива (array-index-list):

1.2.1. [целая константа (integer-constant)],

1.2.2. [целая константа (integer-constant)] список индексов массива
(array-index-list) [1]

2. присваивание (assignment):

2.1. выражение (expression),

2.2. левосторонний оператор присваивания (left-hand-side assignment-
operator) присваивание (assignment).

2.1. выражение (expression):

2.1.1. выражение (expression) двоичный оператор (binary-operator)
выражение (expression),

2.1.2. унарный оператор (unary-operator) выражение (expression),

2.1.3. выражение (expression) унарный оператор (unary-operator),

2.1.4. выражение (expression) ? выражение (expression) : выражение
(expression),

2.1.5. (выражение (expression)),

2.1.6. идентификатор (identifier),

2.1.7. константа (constant),

2.1.8. имя функции (function-name) (список аргументов (argumentlist)).

2.2. left-hand-side:

2.2.1. идентификатор (identifier),

2.2.2. идентификатор (identifier) описание массива (array-subscription).

2.2.2. описание массива (arraysubscription):

2.2.2.1. [присваивание (assignment)]

2.2.2.2. [присваивание (assignment)] описание массива (array-subscription). [1]

- оператор присваивания (assignment-operator):

=, +=, -=, *=, /=, >>=, <<=, &=, ^=, |=, %=, **=.

- двоичный оператор (binary-operator):

+, -, *, /, ^, !=, ==, >, <, >=, <=, &&, ||, &, |, %, **.

- унарный оператор (unary-operator):

+, -, !, ++, --, ~.

список аргументов (argument-list):

выражение (expression),

выражение (expression), список аргументов (argument-list).

константа (constant): pi, число (number).

3. составной оператор (compound-statement):

3. { список операторов (statement-list) }.

4. условный оператор (conditional-statement):

4.1. оператор if (if-statement),

4.2. оператор if-else (if-else-statement) [1].

4.1. оператор if (ifstatement):

if (присваивание) оператор (if (assignment) statement),

4.2. оператор if-else (if-else-statement):

if оператор else оператор (if-statement else statement).

5. итеративный оператор (iterative-statement):

5.1. оператор do (do-statement):

do оператор while (присваивание) (do statement while (assignment)).

5.2. оператор for (for-statement):

for ([присваивание] ; [присваивание] ; [присваивание]) оператор (for ([assignment] ; [assignment] ; [assignment]) statement).

5.3. оператор while (while-statement):

while (присваивание) оператор (while (assignment) statement).

6. оператор управления (control-statement):

6.1. прервать (break),

6.2. продолжить (continue).

7. оператор переключения (switch-statement):

switch (присваивание) { список операторов варианта }

(switch (assignment) { case-statement-list })

список операторов варианта (case-statement-list):

оператор варианта (case-statement),

список операторов варианта (case-statement-list) оператор варианта
(case-statement).

оператор варианта (case-statement):

номер варианта (case number): список операторов (statement-list),

по умолчанию (default): список операторов (statement-list).

не цифра (non-digit):

один из символов a~z A~Z.

цифра (digit):

один из символов 0 1 2 3 4 5 6 7 8 9.

ненулевая цифра (non-zero-digit):

один из символов 1 2 3 4 5 6 7 8 9.

двоичная цифра (binary-digit):

один из символов 0 1.

восьмеричная цифра (octal-digit):

0 1 2 3 4 5 6 7.

шестнадцатеричная цифра (hex-digit):

0 1 2 3 4 5 6 7 8 9 a b c d e f A B C D E F.

идентификатор (identifier):

не цифра (non-digit) [не первый символ (non-first-character)].

не первый символ (non-first-character):

не цифра (non-digit) [не первый символ (non-first-character)],
 цифра (digit) [не первый символ (non-first-character)].

число (number):

целая константа (integerconstant)

константа с плавающей точкой (float-constant)

целая константа (integer-constant):

десятичная константа (decimal-constant),

двоичная константа (binary-constant),

восьмеричная константа (octal-constant),

шестнадцатеричная константа (hex-constant).

десятичная константа (decimal-constant):

ненулевая цифра (non-zero-digit) набор цифр (#digit).

двоичная константа (binary-constant):

0b набор двоичных цифр (#binary-digit),

0B набор двоичных цифр (#binary-digit).

восьмеричная константа (octal-constant):

0 набор восьмеричных цифр (#octal-digit).

шестнадцатеричная константа (hex-constant):

0x набор шестнадцатеричных цифр (#hex-digit),

0X набор шестнадцатеричных цифр (#hex-digit).

константа с плавающей точкой (float-constant):

дробь (fraction) экспоненциальная часть (exponent-part),

десятичная константа (decimal-constant) экспоненциальная часть (exponent-part).

дробь (fraction):

набор цифр (#digit) . цифра (digit) набор цифр (#digit).

экспоненциальная часть (exponent-part):

e [знак (sign)] набор цифр (#digit),

E [знак (sign)] набор цифр (#digit).

знак (sign):

+ или −.[1]

Оборудование и материалы

Дисплейный класс, среда графического программирования LabView версии 2009 или выше.

Указания по технике безопасности

К работе допускаются студенты, прошедшие инструктаж по технике безопасности при выполнении работ в лабораториях кафедры «Автоматизированных электроэнергетических систем и электроснабжения», инструкция приведена в Приложении 1. Прохождение инструктажа по технике безопасности фиксируется преподавателем в специальном журнале.

Указания по порядку выполнения работы

Создать виртуальный прибор выполняющий арифметические действия сложения, вычитания и умножения чисел двумя способами

Выполнение работы

Пример создания простого вычислительного устройства

1. Откройте новый виртуальный прибор File>>New VI.
2. На лицевой панели(Front Panel) разместите четыре управляющих элемента для ввода чисел A, B, C, D. Из подпалитры «Numeric Controls» палитры «Controls» выберите цифровой элемент ввода чисел «Numeric Control». Разместите элемент на поле лицевой панели и в появившейся метке введите название переменных.
3. На лицевой панели создайте три цифровых индикатора для

вывода результатов вычисления.

Из подпалитры «Numeric Controls» палитры «Controls» выбрать цифровой элемент ввода чисел «Numeric Indicator».

Разместить элемент на поле лицевой панели и в появившейся метке введите название «A+B», «A-B», «A*B».

4. На данный момент лицевая панель может выглядеть так, как показано на рисунке 1.3.

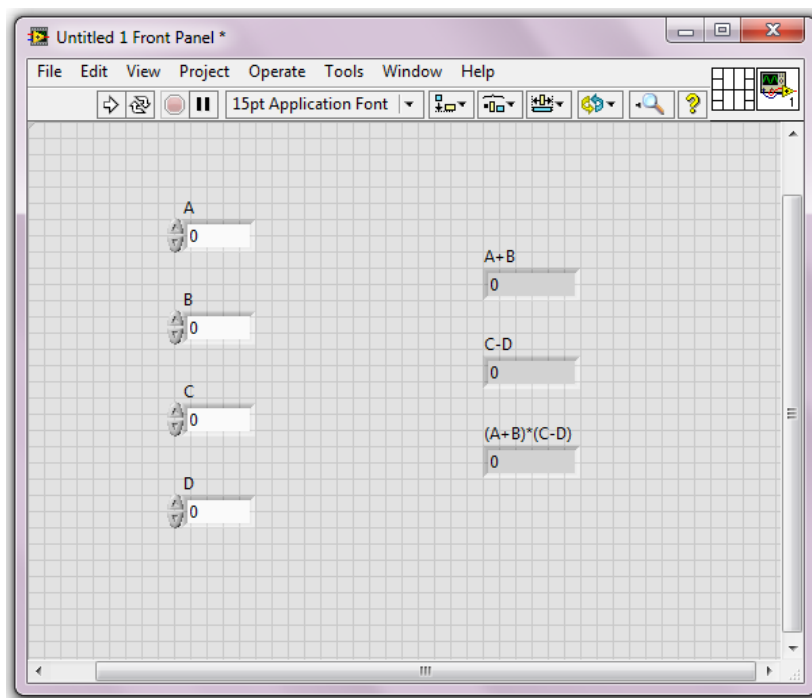


Рисунок 1.3 – Лицевая панель простого вычислительного устройства

5. Переключитесь на структурную схему Windows>>Show Diagram. На структурной схеме размещены терминалы, соответствующие органам управления и индикаторам лицевой панели.

Терминалы имеют те же метки, что и соответствующие им объекты лицевой панели.

6. Разместите на ней объекты управления (Controls) слева, а индикаторы (Indicators) справа аналогично тому, как они расположены на лицевой панели.

7. Выберите функции вычислительного устройства из палитры «Functions» >> «Programming»>> «Numeric»: сложение (Add), вычитание (Subtract), умножение (Multiply) (рисунок 1.4).

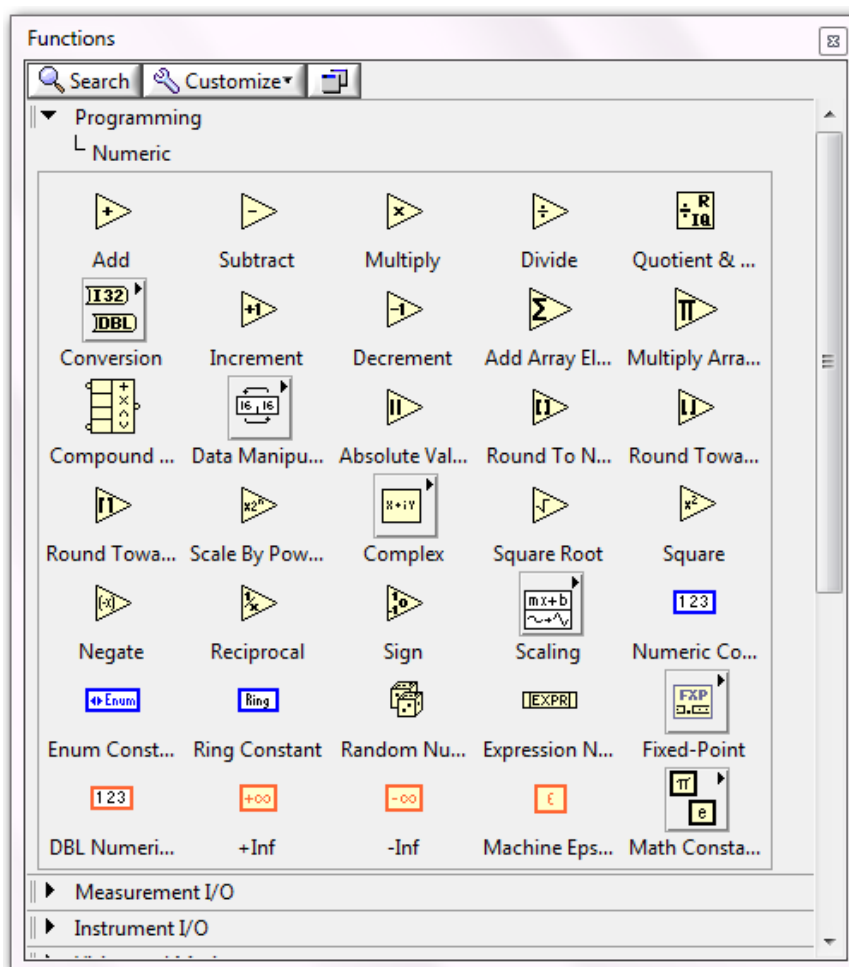


Рисунок 1.4 – Функциональные элементы палитры «Numeric»

8. Из палитры «Tools» выберите «катушку». Соедините между собой терминалы органов управления, функций и индикаторов.

9. На этом монтаж структурной схемы закончен. На данный момент структурная схема прибора может выглядеть, так как показано на рисунке 1.5.

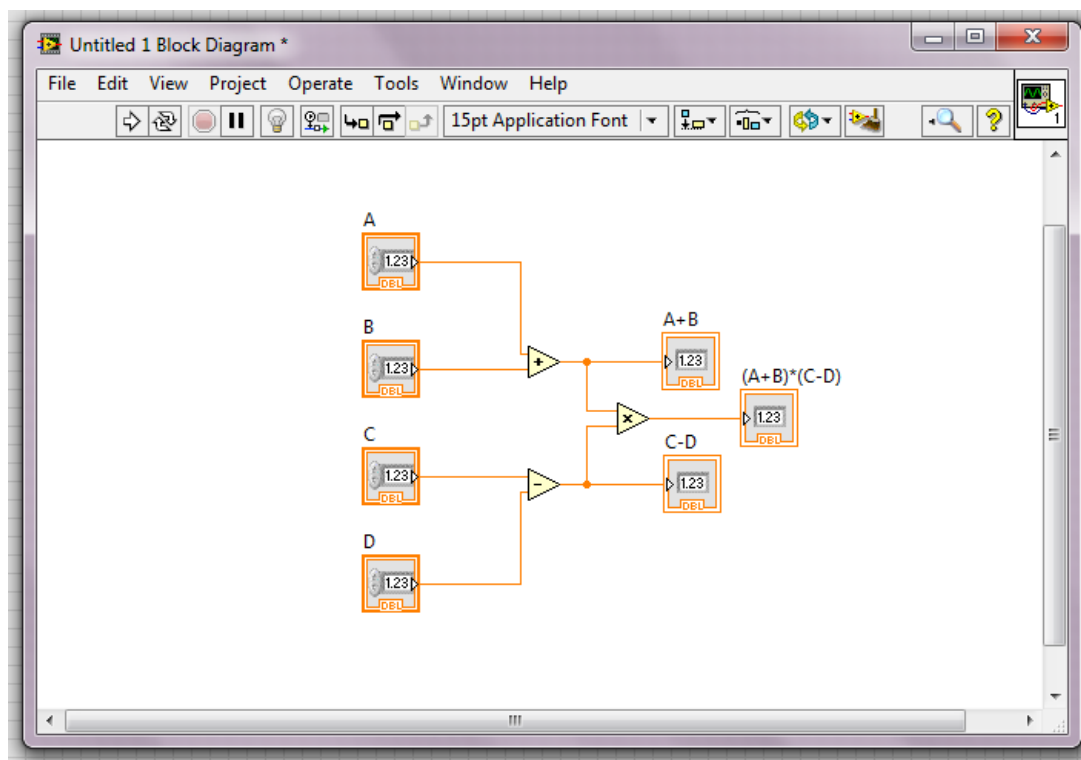


Рисунок 1.5 – Блок диаграмма простого вычислительного устройства

10. Перейдите в окно лицевой панели и запустите виртуальный прибор, нажав кнопку «Run» («запуск») в левом верхнем углу окна.
11. Выполнить те же арифметические вычисления при помощи функции «Formula Node». Переключитесь на структурную схему Windows>>Show Diagram. Выберите функции вычислительного устройства из палитры «Functions» >> «Programming»>> «Structures»: «Formula Node». (Рисунок 1.6)

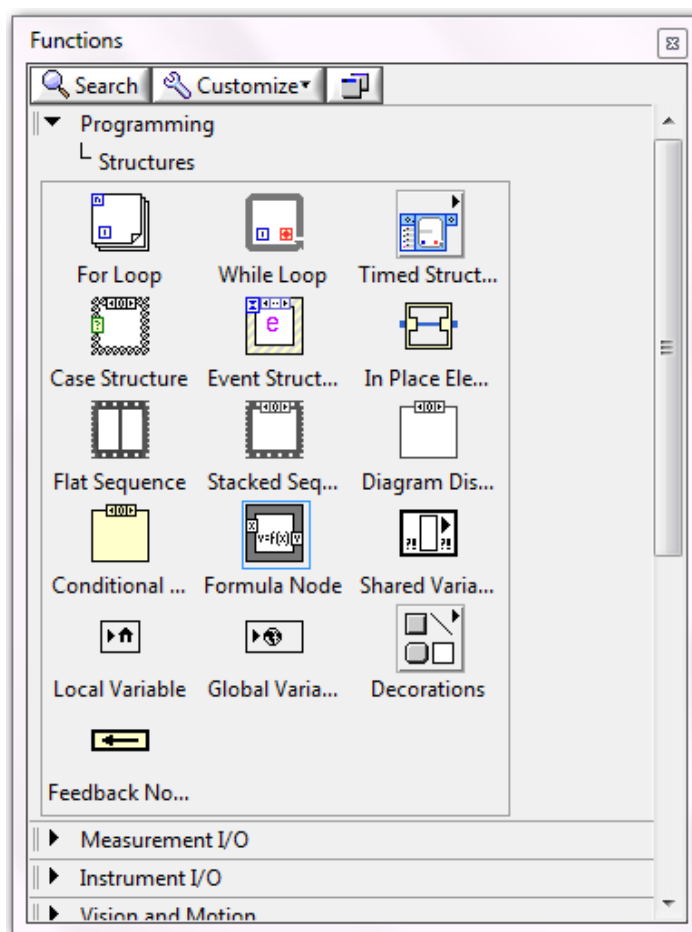


Рисунок 1.6 – Функциональные элементы палитры «Structures»

12. Создать прямоугольную область «Formula Node» и на левой границе, которой необходимо создать четыре входа(A, B, C, D), а на правой три выхода(S1, S2, S3).
13. Из подпалитры «Numeric Controls» палитры «Controls» выбрать цифровой элемент ввода чисел «Numeric Indicator». Создать три индикатора с именами S1, S2, S3.
14. На поле области «Formula Node» написать арифметические операции в виде формул: $S1=A+B$;
 $S2=C-D$;
 $S3=S1*S2$;
15. Из палитры «Tools» выберите «катушку». Соедините между собой четыре управляющих элементов для ввода чисел A, B, C, D и четыре входов (A, B, C, D) области «Formula Node», а так же выходы области «Formula Node» с индикаторами с именами S1, S2, S3.

16. На этом монтаж структурной схемы закончен. На данный момент структурная схема прибора может выглядеть, так как показано на рисунке 1.7.

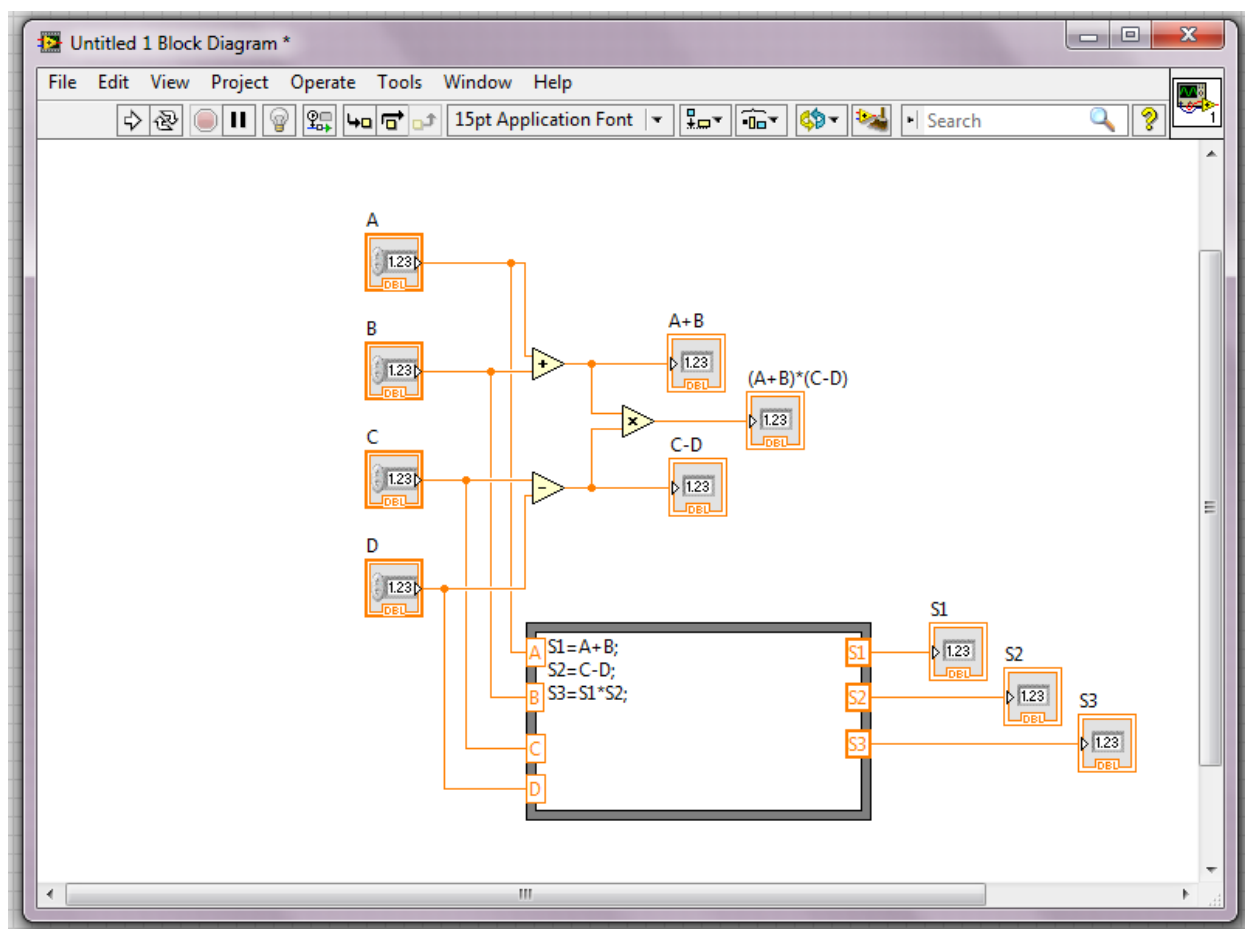


Рисунок 1.7 – Блок диаграмма простого вычислительного устройства

17. Перейдите в окно лицевой панели и запустите виртуальный прибор, нажав кнопку «Run» («запуск») в левом верхнем углу окна

Содержание отчета

Отчет должен содержать:

Титульный лист;

Задание с указанием цели и задач;

Принт-скрины лицевой панели и блок-диаграммы выполненного задания в среде Labview;

Выводы по работе.

Контрольные вопросы

1. Из каких основных компонентов состоит виртуальный прибор?
2. Что называется лицевой панелью?
3. Что называется блок-диаграммой?
4. Для чего предназначена палитра инструментов?
5. Для чего предназначена палитра Controls (Элементов)?
6. Для чего предназначена палитра Functions (Функций)?
7. Как производится обращение к всплывающему меню на свободном рабочем пространстве лицевой панели и блок-диаграммы?

Литература: 1, 2, 3.

Лабораторная работа №2

Работа с циклами и графический вывод данных.

ВП Монитор напряжения

Цель работы: ознакомление с циклами, графическим выводом данных из циклов и возможностями их применения в LabView. В связи с этим необходимо решить следующие задачи:

1. Выполнить построение программы, используя разные виды циклов согласно заданию.

2. Сравнить результаты работы приложения при визуализации решений с использованием разных видов циклов при решении математических задач, индивидуализировать блок диаграмму изменением размера шрифтов и цвета элементов.

Знания, умения и владения, приобретаемые обучающимся в результате освоения тем в рамках освоения компетенций: способностью осуществлять поиск, хранение, обработку и анализ информации из различных источников и баз данных, представлять ее в требуемом формате с использованием информационных, компьютерных и сетевых технологий (ОПК-1); способностью применять соответствующий физико-математический аппарат, методы анализа и моделирования, теоретического и экспериментального исследования при решении профессиональных задач (ОПК-2)

Знать: о связи дисциплины с другими изучаемыми дисциплинами; о роли дисциплины в дальнейшей научной и практической деятельности; о возможностях и преимуществах моделирования в среде LabView; терминологию, основные понятия и определения; технологию графического программирования в среде LabView; типы данных и последовательность их обработки (ОПК-1); общие вопросы построения измерительных систем с использованием: современных компьютерных технологий, принципы решения технических задач; измерений с применением современных аппаратных и программных средств (ОПК-2);

Уметь: формулировать постановку задачи, определять цель (ОПК-1); оптимально выбирать программно-аппаратные средства для решения задач и проблем измерительной техники; разработать алгоритмы измерения и обработки данных (ОПК-2);

Владеть: терминологией, основными понятиями и определениями; технологией графического программирования в среде LabView; навыками обработки различных типов данных; общими вопросами построения измерительных систем с использованием современных компьютерных технологий (ОПК-1); принципами решения технических задач измерений с применением современных аппаратных и программных средств (ОПК-2);

Теоретическая часть

Многократные повторения и Циклы.

Структуры являются графическим представлением операторов цикла и операторов Case (Варианта), используемых в текстовых языках программирования. Структуры на блок-диаграмме используются для выполнения повторяющихся операций над потоком данных, операций в определенном порядке и наложения условий на выполнение операций. Среда LabVIEW содержит пять структур: Цикл While (по условию), Цикл For (с фиксированным числом итераций), структура Case (Вариант), структура Sequence (Последовательность), структура Event (Событие), а также Formula Node (узел Формулы).

Цикл While (по Условию)

Цикл While (по условию) работает до тех пор, пока не выполнится логическое условие выхода из цикла. Цикл While аналогичен циклам Do и Repeat Until, используемым в текстовых языках программирования. Следующая иллюстрация демонстрирует (1) цикл While в среде LabVIEW, (2) эквивалентную блок-схему работы цикла While, (3) пример текстового аналога кода работы цикла While.

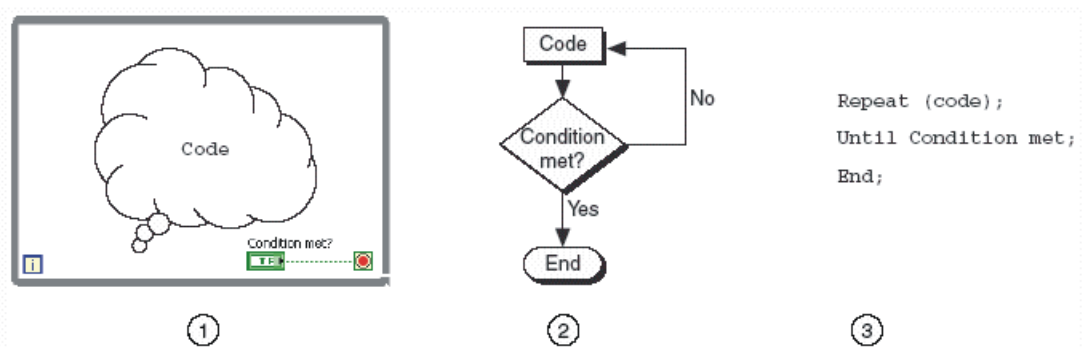


Рисунок 2.1 – 1. LabVIEW цикл While, 2. Блок-схема, 3. Текстовый аналог.

Цикл While находится в палитре Functions»Structures. После того как цикл выбран в палитре Functions (Функций), следует с помощью курсора выделить часть блок-диаграммы, которую необходимо поместить в цикл. После отпускания кнопки мыши, выделенная область блок-диаграммы помещается в тело цикла.

Добавление объектов блок-диаграммы в тело цикла осуществляется помещением или перетаскиванием объектов.



Блок-диаграмма цикла **While** выполняется до тех пор, пока не выполнится условие выхода из цикла. По умолчанию, терминал условия выхода имеет вид, показанный слева. Это значит, что цикл будет выполняться до поступления на терминал условия выхода значения TRUE. В этом случае терминал условия выхода называется терминалом **Stop If True** (Остановка если Истина).



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

На блок-диаграмме, показанной на рисунке, условие выхода из цикла **While** определяется значением выходного параметра подпрограммы ВП большего или равного 10,00 и состоянием терминала элемента управления **Enable**. Функция **And** (Логическое «И») на выходе выдает значение TRUE, если оба на поля ввода данных функции поступают значения TRUE. В

противном случае функция на выходе выдает значение FALSE и работа цикла завершается.

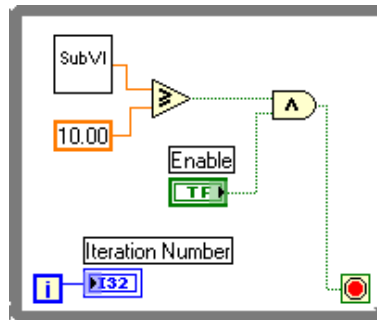


Рисунок 2.2

В предыдущем примере велика вероятность получения бесконечно выполняемого цикла. Обычно стремятся получить единственное условие выхода из цикла, нежели одновременное выполнение двух условий.



Предусмотрена возможность изменения условия выхода и соответствующего ему изображения терминала условия выхода. Щелчком правой кнопки мыши по терминалу условия выхода или по границе цикла необходимо вызвать контекстное меню и выбрать пункт **Continue If True** (Продолжение если Истина). Также можно воспользоваться инструментом УПРАВЛЕНИЕ, щелкнув им по терминалу условия выхода. Изображение терминала условия выхода поменяется на показанное слева **Continue If True** (Продолжение Если Истина). В результате условием выхода из цикла становится поступающее на терминал условия выхода значение FALSE, как показано на следующей блок-диаграмме.

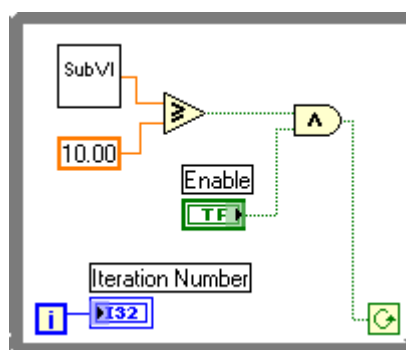


Рисунок 2.3

Цикл **While** выполняется до тех пор, пока выходные данные подпрограммы ВП остаются меньше «10».

Терминалы входных/выходных данных цикла

Данные могут поступать в цикл **While** (или выходить из него) через терминалы входных/выходных данных цикла. Терминалы входных/выходных данных цикла передают данные из структур и в структуры. Терминалы входных/выходных данных цикла отображаются в виде сплошных прямоугольников на границе области цикла **While**. Прямоугольник принимает цвет типа данных, передаваемых по терминалу. Данные выходят из цикла по его завершении. В случае если данные поступают в цикл **While** через терминал входных/выходных данных цикла, выполнение цикла начинается при поступлении данных в терминал.

На следующей блок-диаграмме терминал счетчика итераций присоединен к терминалу выхода цикла. Значения из терминала выхода цикла не поступают к элементу отображения номера итерации до завершения цикла **While**.

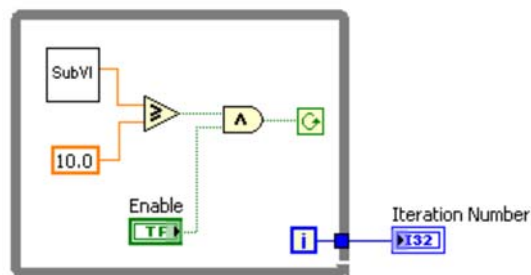


Рисунок 2.4

Лишь последнее значение итерации отображается элементом отображения номера итерации.

Цикл For (с фиксированным числом итераций)

Цикл For (с фиксированным числом итераций) выполняет повторяющиеся операции над потоком данных определенное количество раз (по заданию). Следующие рисунки демонстрируют Рис.2.5 (1) цикл For в

среде LabVIEW, (2) эквивалентную блок-схему работы цикла For, (3) пример текстового аналога кода работы цикла For.

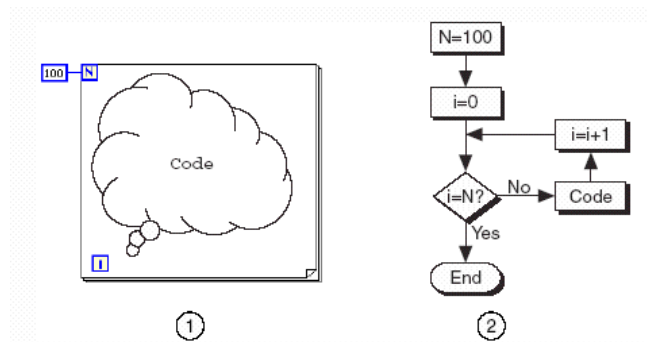


Рисунок 2.5 – (1) цикл For в среде LabVIEW, (2) эквивалентная блок-схема работы цикла For

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

③

Рисунок 2.6 – Пример текстового аналога кода работы цикла For

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

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

Цикл **For** отличается от цикла **While** тем, что завершает работу, выполнив заданное максимальное число итераций **N**. Цикл **While** завершает работу при выполнении заданного условия выхода из цикла.

Функции ожидания



Функция **Wait Until Next ms Multiple**, показанная слева, обеспечивает интервал между итерациями, равный интервалу времени, необходимому для того, чтобы миллисекундный счетчик достиг значения, кратного введенному пользователем. Эта функция используется для синхронизации действий. Функцию **Wait Until Next ms Multiple** вызывают внутри цикла для контроля скорости выполнения цикла.

Функция **Wait Until Next ms Multiple** обеспечивает интервал между итерациями, равный интервалу времени, необходимому внутреннему таймеру компьютера для достижения указанного кратного значения. Существует вероятность, что первый период цикла будет коротким, как показано ниже.

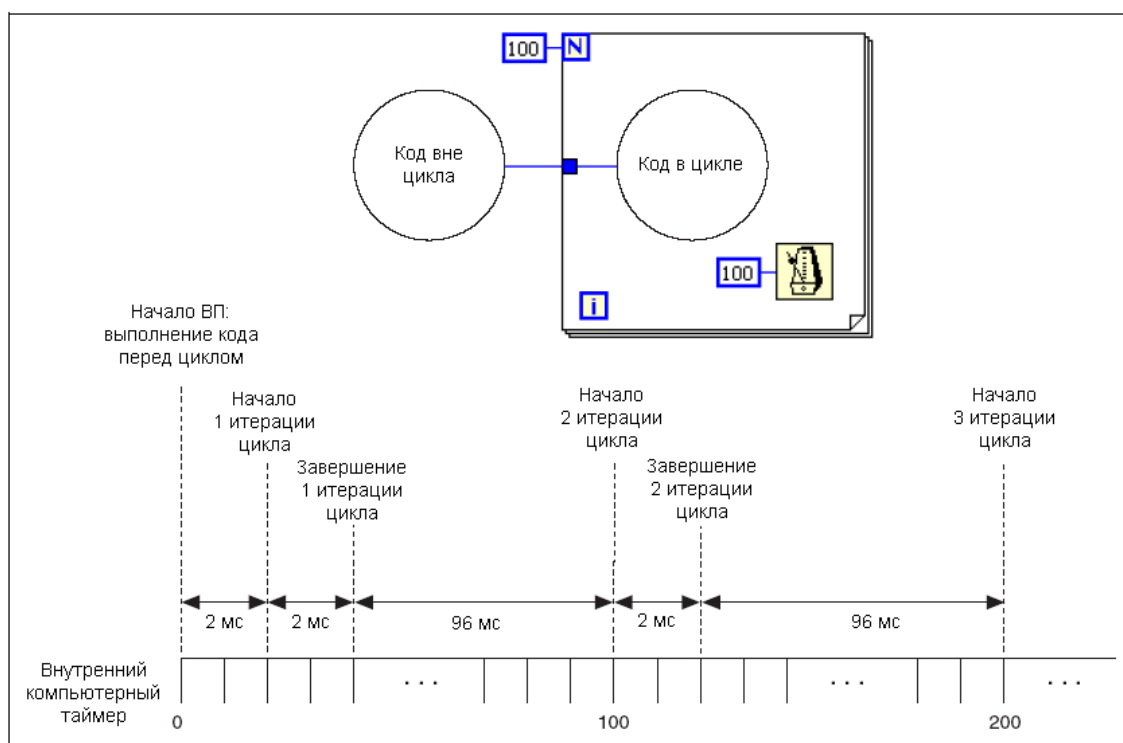


Рисунок 2.6 – Функция **Wait Until Next ms Multiple**



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

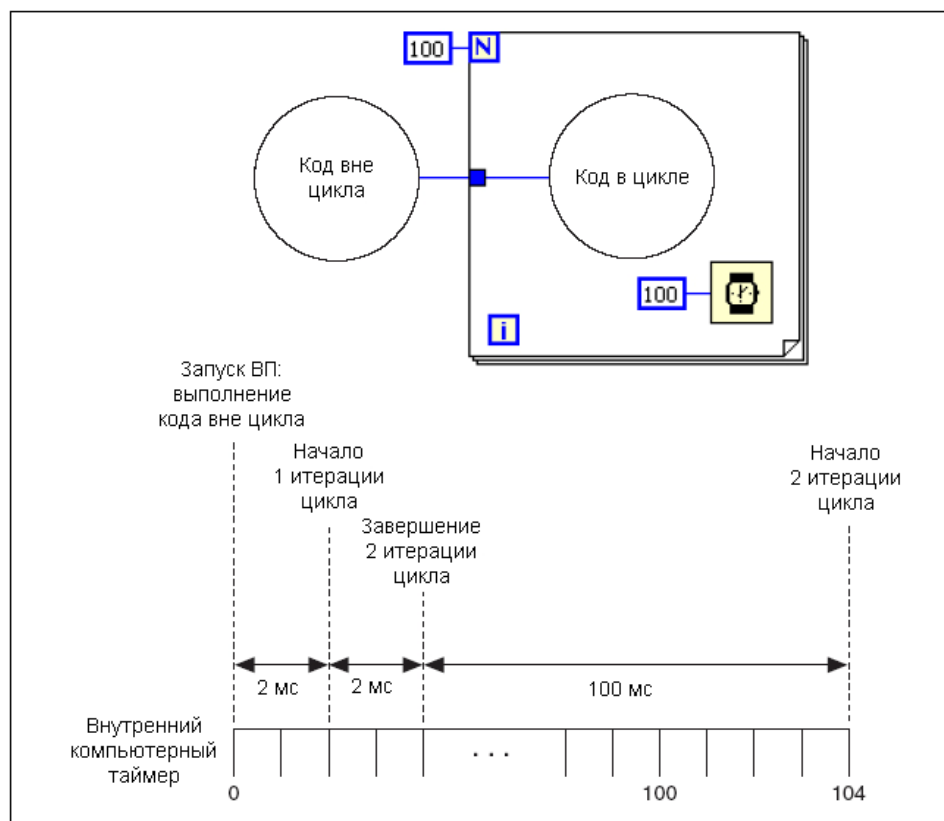


Рисунок 2.7 – Функция **Wait(ms)**

Сдвиговые регистры.

Сдвиговые регистры используются при работе с циклами для передачи значений от текущей итерации цикла к следующей. Сдвиговые регистры аналогичны статическим переменным в текстовых языках программирования

Сдвиговый регистр выглядит как пара терминалов, показанных слева. Они расположены непосредственно друг против друга на противоположных вертикальных сторонах границы цикла. Правый терминал содержит стрелку «вверх» и сохраняет данные по завершению текущей итерации. LabVIEW передает данные с этого регистра в следующую итерацию цикла. Сдвиговый регистр создается щелчком правой кнопки мыши по границе цикла и выбором из контекстного меню пункта **Add Shift Register**.

Сдвиговый регистр передает данные любого типа, он автоматически принимает тип первых поступивших на него данных. Данные, передаваемые на терминалы сдвигового регистра, должны быть одного типа.

Чтобы инициализировать сдвиговый регистр, необходимо передать на его левый терминал любое значение извне цикла. Если не инициализировать сдвиговый регистр, он использует значение, записанное в регистр во время последнего выполнения цикла или значение, используемое по умолчанию для данного типа данных, если цикл никогда не выполнялся.

Цикл с неинициализированным сдвиговым регистром используется при неоднократном запуске ВП для присвоения выходному значению сдвигового регистра значения, взятого с последнего выполнения ВП. Чтобы сохранить информацию о состоянии между последующими запусками ВП, следует оставить вход левого терминала сдвигового регистра не определенным. После завершения выполнения цикла последнее значение, записанное в регистр, останется на правом терминале. При последующей передаче данных из цикла через правый терминал будет передано последнее значение, записанное в регистр.

Предусмотрена возможность создания нескольких сдвиговых регистров в одной структуре цикла. Если в одном цикле выполняется несколько операций, следует использовать сдвиговый регистр с несколькими терминалами для хранения данных, полученных в результате выполнения различных операций цикла. На следующем рисунке показано использование двух инициализированных сдвиговых регистров.

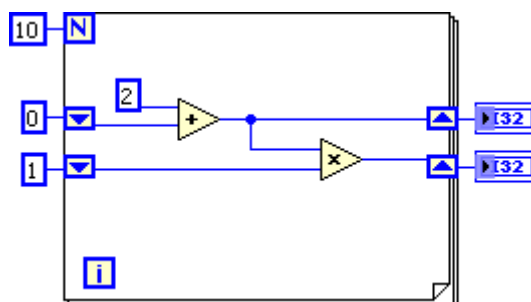


Рисунок 2.7 – Пример выполнения сдвигового регистра

Краткое изложение пройденного материала, советы и секреты.

- Структуры на блок-диаграмме используются для повторяющихся операций над потоками данных, для выполнения операций над данными в определенном порядке и наложения условий выполнения операций.

- Цикл While выполняется до тех пор, пока на его терминал условия выхода не поступит логическое значение выхода из цикла. По умолчанию, цикл While выполняется, пока на терминал условия выхода из цикла не поступит значение TRUE.

- Цикл For выполняется определенное количество раз (по заданию).

- Для создания цикла достаточно с помощью курсора выделить часть блок-диаграммы, которую необходимо поместить в цикл, и отпустить кнопку мыши или перенести эту часть блок-диаграммы в тело цикла с помощью курсора.

- Функция Wait Until Next ms Multiple обеспечивает заданный интервал времени выполнения итераций цикла.

- Функция Wait ожидает определенное время.

- Для передачи данных из одной итерации цикла в другую используются сдвиговые регистры.

- Для создания сдвигового регистра необходимо щелкнуть правой кнопкой мыши по границе цикла и выбрать из контекстного меню пункт Add Shift Register.

- Для передачи значений из нескольких предыдущих итераций цикла в последующую необходимо сдвиговому регистру добавить дополнительные терминалы данных. Для этого необходимо щелкнуть по нему правой кнопкой мыши и выбрать из контекстного меню пункт Add Element.

- Узел обратной связи сохраняет данные любого типа по завершению текущей итерации, передает эти значения в следующую итерацию.

- Узел обратной связи используется для уменьшения количества соединений.

Узел Формулы.

Узел Формулы (Formula Node) используется для выполнения математических операций в среде LabVIEW.

Узел Формулы используется для выполнения математических операций в текстовом виде на блок-диаграмме. Использовать узел Формулы удобно, когда выражения имеют много переменных, или они достаточно сложные, а также для уже имеющихся текстовых математических кодов. Для ускорения процесса создания алгоритма можно копировать и вставлять имеющиеся текстовые математические коды в узел Формулы вместо их воссоздания на блок-диаграмме.

Создание терминалов входных и выходных данных узла Формулы осуществляется щелчком правой кнопки мыши по границе узла. В контекстном меню необходимо выбрать пункты Add Input или Add Output, а затем ввести переменные для входа и выхода. Далее вводится уравнение в рабочую область структуры. Каждое выражение должно заканчиваться разделителем (;).

Узел Формулы может также использоваться для принятия решений. На следующей блок-диаграмме показаны два эквивалентных способа применения операторов if – then в узле Формулы.

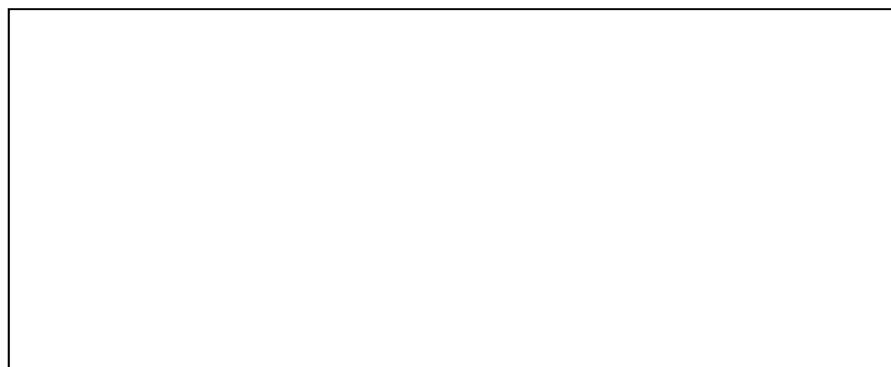


Рисунок 2.8 – Узел Формулы

Узел Формулы позволяет производить разнообразные математические операции. Для получения более подробной информации о функциях, операциях и синтаксисе узла Формулы используйте справку LabVIEW Help

Задание

Необходимо найти значения функции $y=\sin(A*x)+\sin(B*x)$ в диапазоне значений x от 0 до 1000 двумя способами используя структуру «Formula Node» для вычисления значений уравнения и структуры «For Loop» и «While loop» для дискретного изменения переменной x с шагом 1. Необходимо использовать график «Chart» из палитры «Graph Indicator» для отображения одновременно вычисленные значения уравнения в графической форме.

Выполнение работы

1. Откройте новый виртуальный прибор File>>New VI.
2. На лицевой панели(Front Panel) разместите два управляющих элемента для ввода чисел A, B. (**A – первая цифра варианта, B – вторая цифра варианта**) Из подпалитры «Graph Indicator» палитры «Express» выберите график «Chart». Разместите два графических элемента на поле лицевой панели и в появившейся метке введите названия графиков.
4. На данный момент лицевая панель может выглядеть так, как показано на рисунке 2.9.
5. Переключитесь на структурную схему Windows>>Show Diagram. На структурной схеме размещены терминалы, соответствующие органам управления и индикаторам лицевой панели. Терминалы имеют те же метки, что и соответствующие им объекты лицевой панели.
6. Разместите на ней объекты управления(Controls) слева, а индикаторы(Indicators) справа аналогично тому, как они расположены на лицевой панели.
7. Выберите и разместите на поле структуры «For Loop» и «While loop» , а так же в каждом из циклов структуру «Formula Node» (рисунок 2.10).

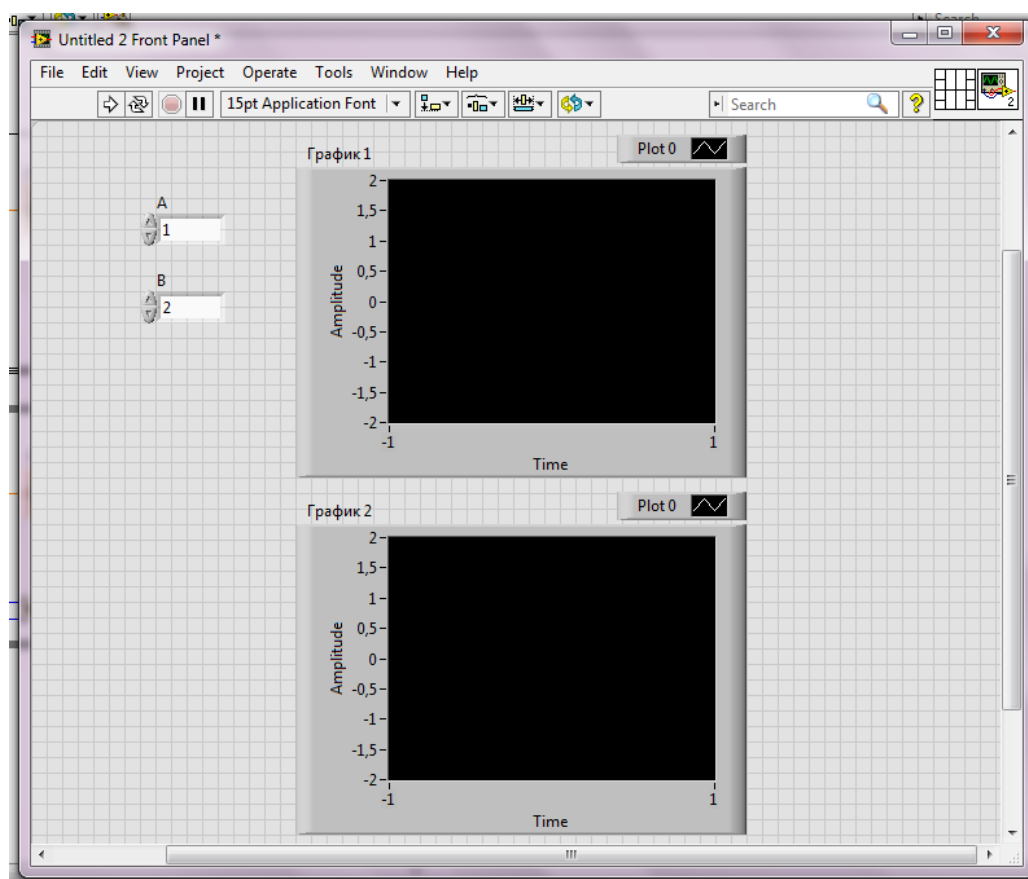


Рисунок 2.9 – Лицевая панель виртуального прибора

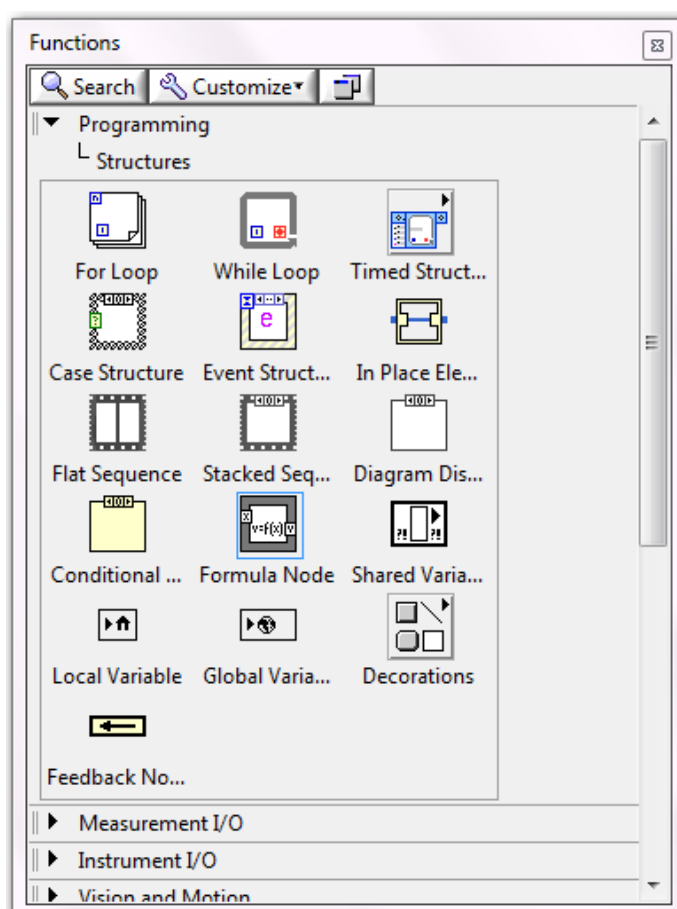


Рисунок 2.10 – Функциональные элементы палитры «Structures»

8. В каждой из структур «Formula Node» циклов «For» и «While» (они идентичны друг другу) необходимо создать входы «A», «B», «x» и выход «y».

«Formula Node» должен содержать арифметическое выражение $y = \sin(A \cdot x) + \sin(B \cdot x)$.

9. После соединения объектов управления с соответствующими вводами «Formula Node» и соответствующих выводов с индикаторами необходимо задать количество итераций $N=1000$ и используя функцию «Equal?» подпалитры «Comparison» палитры «Programming» (рисунок 2.10) построить соответствующее условие остановки цикла «While» как показано на рисунке 2.11.

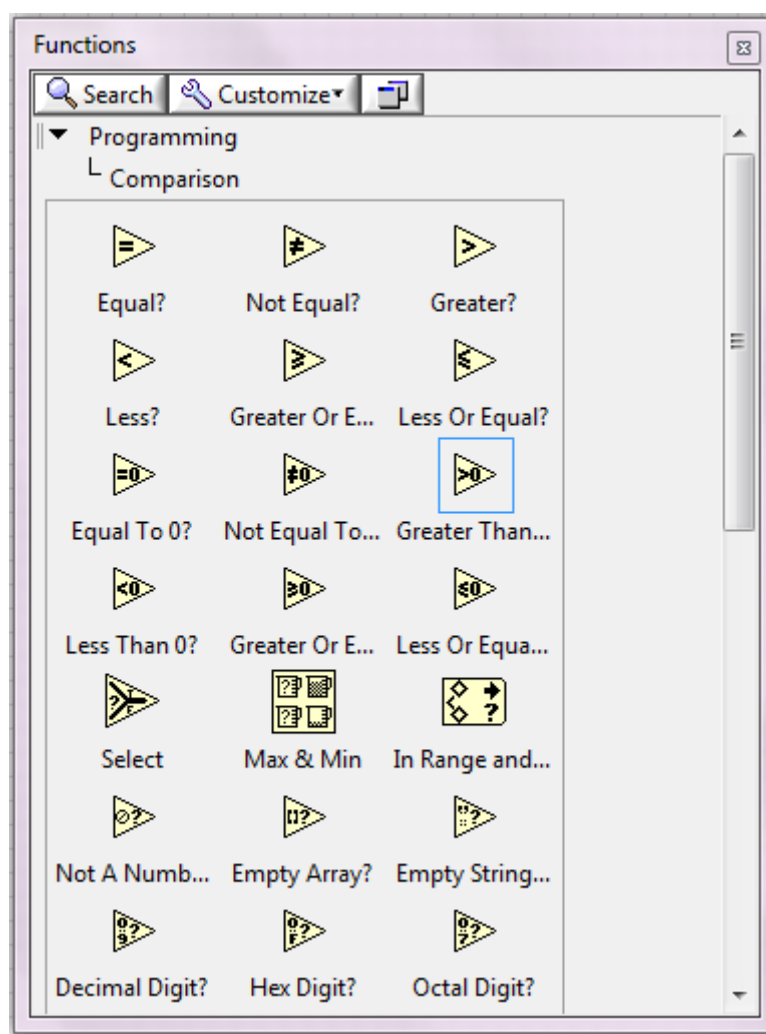


Рисунок 2.10 – Функциональные элементы палитры «Comparison»

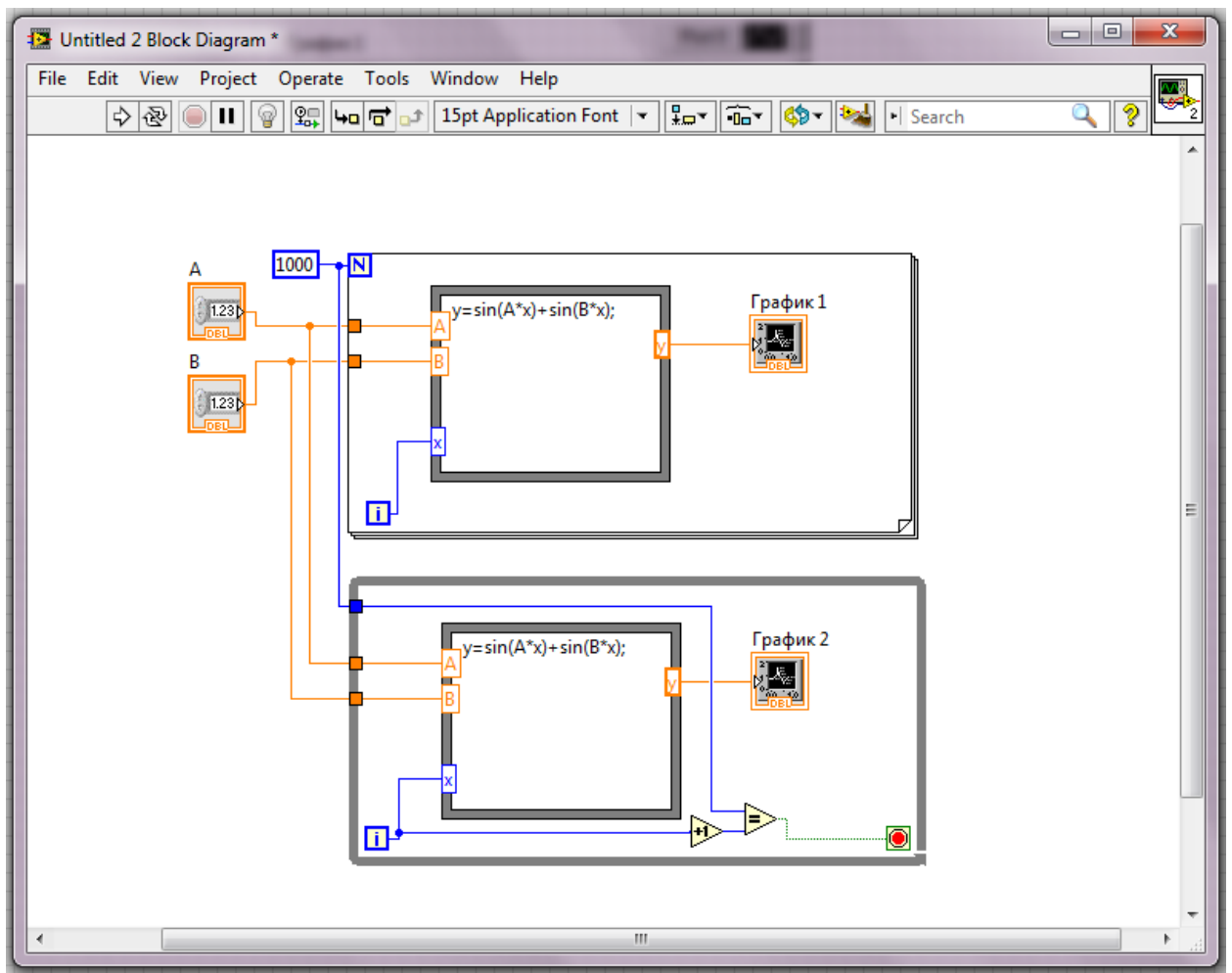


Рисунок 2.11 – Лицевая панель виртуального прибора

10. Перейдите в окно лицевой панели и запустите виртуальный прибор, нажав кнопку «Run» («запуск») в левом верхнем углу окна

Содержание отчета

Отчет должен содержать:

1. Титульный лист;
2. Задание с указанием цели и задач;
3. Принт-скрины лицевой панели и блок-диаграммы выполненного задания в среде Labview;
4. Выводы по работе.

Контрольные вопросы

1. Для чего используются структуры на блок-диаграмме?
2. До возникновения какого события выполняется цикл While?
3. Охарактеризуйте выполнение цикла For?
4. Какие действия необходимо выполнить для создания цикла?
5. Для чего служит функция Wait Until Next ms Multiple?
6. Для чего используются сдвиговые регистры?
7. Как создать сдвиговый регистр?
8. Как передать значения из нескольких предыдущих итераций цикла в последующую при помощи регистра?

Литература: 2, 3, 4.

Лабораторная работа №3

Работа с массивами данных, простая запись данных в файл

Цель работы: ознакомление с технологией записи массивов данных в файл при помощи функционала LabView. В связи с этим необходимо решить следующие задачи:

1. Выполнить построение кода формирующего массивы численных данных согласно задания.
2. Построение кода для отображения данных и выполнения функций работы с массивами, реализовать запись части массива в файл.

Знания, умения и владения, приобретаемые обучающимся в результате освоения тем в рамках освоения компетенций: способностью осуществлять поиск, хранение, обработку и анализ информации из различных источников и баз данных, представлять ее в требуемом формате с использованием информационных, компьютерных и сетевых технологий (ОПК-1); способностью применять соответствующий физико-математический аппарат, методы анализа и моделирования, теоретического и экспериментального исследования при решении профессиональных задач (ОПК-2)

Знать: о связи дисциплины с другими изучаемыми дисциплинами; о роли дисциплины в дальнейшей научной и практической деятельности; о возможностях и преимуществах моделирования в среде LabView; терминологию, основные понятия и определения; технологию графического программирования в среде LabView; типы данных и последовательность их обработки (ОПК-1); общие вопросы построения измерительных систем с использованием: современных компьютерных технологий, принципы решения технических задач; измерений с применением современных аппаратных и программных средств (ОПК-2);

Уметь: формулировать постановку задачи, определять цель (ОПК-1); оптимально выбирать программно-аппаратные средства для решения задач и

проблем измерительной техники; разработать алгоритмы измерения и обработки данных (ОПК-2);

Владеть: терминологией, основными понятиями и определениями; технологией графического программирования в среде LabView; навыками обработки различных типов данных; общими вопросами построения измерительных систем с использованием современных компьютерных технологий (ОПК-1); принципами решения технических задач измерений с применением современных аппаратных и программных средств (ОПК-2);

Теоретическая часть

Массивы.

Массивы объединяют элементы одного типа данных. Массив – это набор элементов определенной размерности. Элементами массива называют группу составляющих его объектов. Размерность массива - это совокупность столбцов (длина) и строк (высота), а также глубина массива. Массив может иметь одну и более размерностей, и до $2^{31}-1$ элементов в каждом направлении, насколько позволяет оперативная память.

Данные, составляющие массив, могут быть любого типа: целочисленного, логического или строкового. Массив также может содержать элементы графического представления данных и кластеры. Использовать массивы удобно при работе с группами данных одного типа и при накоплении данных после повторяющихся вычислений. Массивы идеально подходят для хранения данных, полученных с графиков, или накопленных во время работы циклов, причем одна итерация цикла создает один элемент массива.

Нельзя создать массив, состоящий из массивов. Однако можно создать массив кластеров, где каждый кластер будет состоять из одного или более массивов.

Все элементы массива упорядочены. Чтобы к ним было легко обращаться, каждому элементу присвоен индекс. Нумерация элементов

массива всегда начинается с 0. Таким образом, индексы массива находятся в диапазоне от 0 до $(n-1)$, где n – число элементов в массиве. Например, в массиве из девяти планет солнечной системы $n=9$, следовательно, значение индекса находится в пределах от 0 до 8. Земля является третьей планетой от Солнца, поэтому ее индекс равен 2.

Создание массива элементов управления и отображения.

Для создания массива элементов управления или отображения данных, как показано в примере, необходимо выбрать шаблон массива из палитры Controls>>Array & Cluster и поместить его на лицевую панель. Затем поместить в шаблон массива элемент управления либо отображения данных. Поместить в шаблон массива запрещенный элемент управления или отображения, например, двухкоординатный график осциллограмм (XY graph) не удастся.

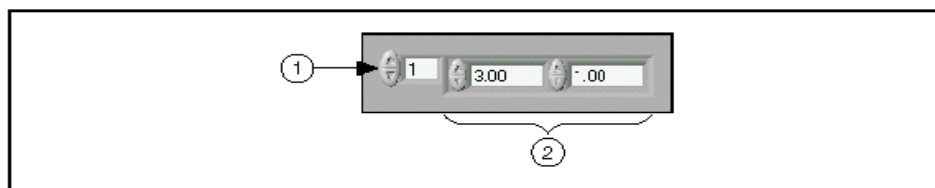


Рисунок 3.1 – Массив: (1) Элемент индекса массива, (2) Элементы значений массива

Поместить объект в шаблон массива следует до того, как он будет использоваться на блок-диаграмме. Если этого не сделать, то шаблон массива не будет инициализирован, и использовать массив будет нельзя.

Двумерные массивы.

В двумерном (2D) массиве элементы хранятся в виде матрицы. Таким образом, для размещения элемента требуется указание индекса столбца и строки. На иллюстрации показан двумерный массив, состоящий из 6 столбцов (длина) и 4 строк (высота). Количество элементов в массиве = 24 ($6 \times 4 = 24$).

		Индекс колонки					
		0	1	2	3	4	5
Индекс строки	0						
	1						
	2						
	3						

Рисунок 3.2 – Пример двумерного массива

Для увеличения размерности массива необходимо щелкнуть правой кнопкой мыши по элементу индекса и выбрать из контекстного меню пункт Add Dimension. С этой целью также можно использовать инструмент ПЕРЕМЕЩЕНИЕ. Для этого надо просто изменить размер элемента индекса. Ниже приведен пример неинициализированного двумерного массива элементов управления.

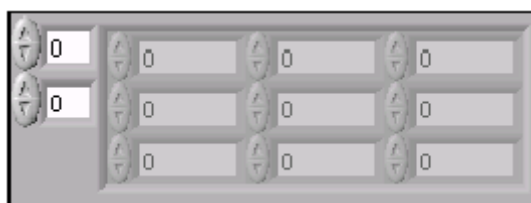


Рисунок 3.2 – Двумерный массив

Создать массив констант на блок-диаграмме можно, выбрав в палитре Functions>>Array шаблон Array Constant и поместив в него числовую константу. Массив констант удобно использовать для передачи данных в подпрограммы ВП.

Автоматическая индексация.

Цикл For и цикл While могут автоматически накапливать массивы и проводить их индексацию на своих границах. Это свойство называется автоиндексацией. После соединения терминала данных массива с терминалом выхода из цикла каждая итерация цикла создает новый элемент массива. На иллюстрации видно, что проводник данных, соединяющий терминал данных массива с терминалом выхода из цикла стал толще, а сам терминал выхода из цикла окрашен в цвет терминала данных массива.



Рисунок 3.3 – Пример создания одномерного массива.

Автоиндексация отключается щелчком правой кнопки мыши по терминалу входа/выхода из цикла и выбором пункта контекстного меню Disable Indexing. Автоиндексацию следует отключать, например, в случае, когда нужно знать только последнее значение.

Ввиду того, что цикл For часто используется при работе с циклами, для него в LabVIEW автоиндексация включена по умолчанию. Для цикла While автоиндексация по умолчанию отключена. Для того, чтобы включить автоиндексацию, необходимо щелкнуть правой кнопкой мыши по терминалу входа/выхода из цикла и выбрать в контекстном меню пункт Enable Indexing.

Создание двумерных (2D) массивов.

Для создания двумерных массивов необходимо использовать два цикла For, один внутри другого. Как показано на иллюстрации, внешний цикл создает элементы массива в строке, а внутренний цикл создает элементы массива в столбце.

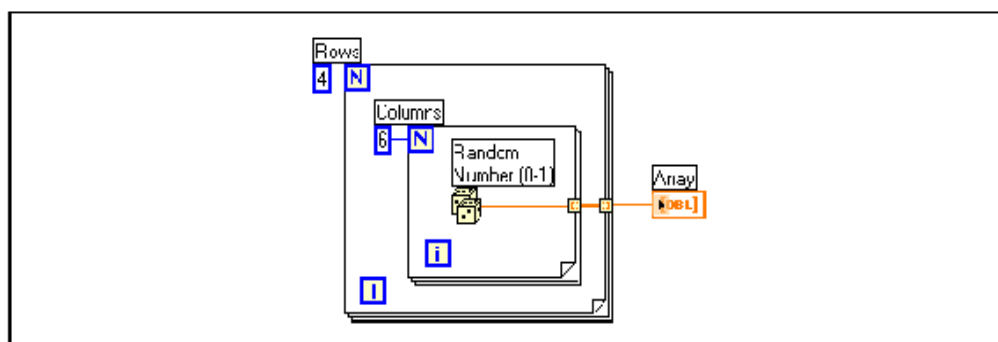


Рисунок 3.4 – Пример создания двумерного массива

Функции работы с массивами.

Для создания и управления массивами используются функции, расположенные в палитре Functions>>Array. Наиболее часто используемые функции работы с массивами включают в себя:



Array Size - показывает количество элементов массива каждой размерности. Если массив n-мерный, на выходе функции **Array Size** будет массив из n элементов. Например, для приведенного ниже массива функция **Array Size** выдаст значение 3.

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



Initialize Array - создает n-мерный массив, в котором каждый элемент инициализирован значением поля ввода данных **element**. Для увеличения размерности массива достаточно добавить поля ввода данных, растянув узел функции. Например, если для функции **Initialize Array** заданы следующие значения параметров: на поле **element** подается значение 4, а на поле **dimension size** (если оно одно) - значение 3, то на выходе получится массив, показанный ниже.

4	4	4
---	---	---

Build Array - объединяет несколько массивов или добавляет элемент в n-мерный массив. Изменение размера функции увеличивает количество полей ввода данных, что позволяет увеличить количество добавляемых элементов. Например, если объединить два предыдущих массива, то функция **Build Array** выдаст на выходе следующий массив.

7	4	2
4	4	4

Для объединения входных данных в более длинный массив той же размерности, как показано ниже, достаточно щелкнуть правой кнопкой мыши на функции и выбрать из контекстного меню пункт **Concatenate Inputs**.

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



Array Subset - выдает часть массива, начиная с индекса, поступившего на поле **index**, и длиной, указанной в поле **length**. Например, если подать предыдущий массив на поле ввода функции **Array Subset**, значение 2 – на поле **index** и 3 – на поле **length**, то на поле вывода данных будет следующее подмножество

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



Index Array - выдает элемент, соответствующий индексу, значение которого подается на поле ввода **index**. Например, при использовании предыдущего массива, функция **Index Array** выдаст значение 2, если на поле ввода данных **index** подать значение 0.

Функцию **Index Array** можно использовать для выделения строки или столбца из двумерного массива и дальнейшего отображения в виде подмассива. Для этого двумерный массив надо подать в поле ввода данных функции. Функция **Index Array** должна иметь два поля **index**. Верхнее поле **index** указывает строку, а нижнее – столбец. Можно задействовать оба поля **index** для выбора отдельного элемента или только одно, для выбора строки или столбца. Например, в поле ввода данных функции подается массив, показанный ниже.

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

	4	4	4
--	---	---	---

Функция **Index Array** в поле вывода данных выдаст следующий массив в случае, если на поле **index** (строка) подается значение 0.

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

Полиморфизм.

Арифметические функции, расположенные в палитре **Functions>>Numeric**, являются полиморфными. Это означает, что на поля ввода этих функций могут поступать данные различных типов (скалярные величины, массивы). Например, можно использовать функцию **Add** для прибавления скалярной величины к массиву или сложения двух массивов. Если на одно поле ввода данных функции **Add** подать скалярную величину 2, а другое соединить с массивом, показанным ниже,

1	3	2
---	---	---

то функция прибавит 2 к каждому элементу массива, и массив будет иметь вид:

3	5	4
---	---	---

Если на вход функции **Add** подать два предыдущих массива, функция сложит каждый элемент первого массива с соответствующим элементом второго и выдаст результат в виде массива, показанного ниже.

4	8	6
---	---	---

Если с помощью функции **Add** сложить два массива разной размерности, например, таких, как предыдущий и показанный ниже,

3	1	2	3
---	---	---	---

то функция сложит каждый элемент первого массива с соответствующим элементом второго и выдаст результат в виде массива размерностью меньшей из двух исходных.

7	9	8
---	---	---

Краткое изложение теоретического материала.

- Массивы объединяют группу элементов одного типа данных. Данные, составляющие массив, могут быть любого типа: целочисленные, логические или строковые. Массив также может содержать сигнальные данные и кластеры.
- Индекс первого элемента массива всегда равен 0. Таким образом, индексы массива находятся в диапазоне от 0 до (n-1), где n – число элементов в массиве.
- Поместить объект в шаблон массива следует до того, как использовать этот массив на блок-диаграмме. Иначе на блок-диаграмме шаблон массива не будет инициализирован.
- Для создания массива элементов управления или отображения необходимо выбрать его шаблон из палитры **Controls>>Array & Cluster** и поместить его на лицевую панель. Затем поместить в шаблон массива элемент управления либо отображения.
- Если массив соединить с терминалом входа цикла **For** или цикла **While**, то при включенной автоиндексации можно читать и обрабатывать каждый элемент массива.
- Для создания и управления массивами используются функции, расположенные на палитре **Functions>>Array**.
- LabVIEW осуществляет автоиндексацию автоматически. Автоиндексация для цикла **For** включена по умолчанию. Для цикла **While** автоиндексация по умолчанию отключена.

- Полиморфизм означает, что поля ввода данных функций могут различаться по структуре данных (скалярные величины, массивы).

График Диаграмм

График Диаграмм (Waveform Chart)– специальный элемент отображения данных в виде одного и более графиков. График Диаграмм расположен на палитре Controls>>Graph. На рисунке показан пример Графика Диаграмм с двумя графиками: экспериментальные данные и их бегущее среднее значение.

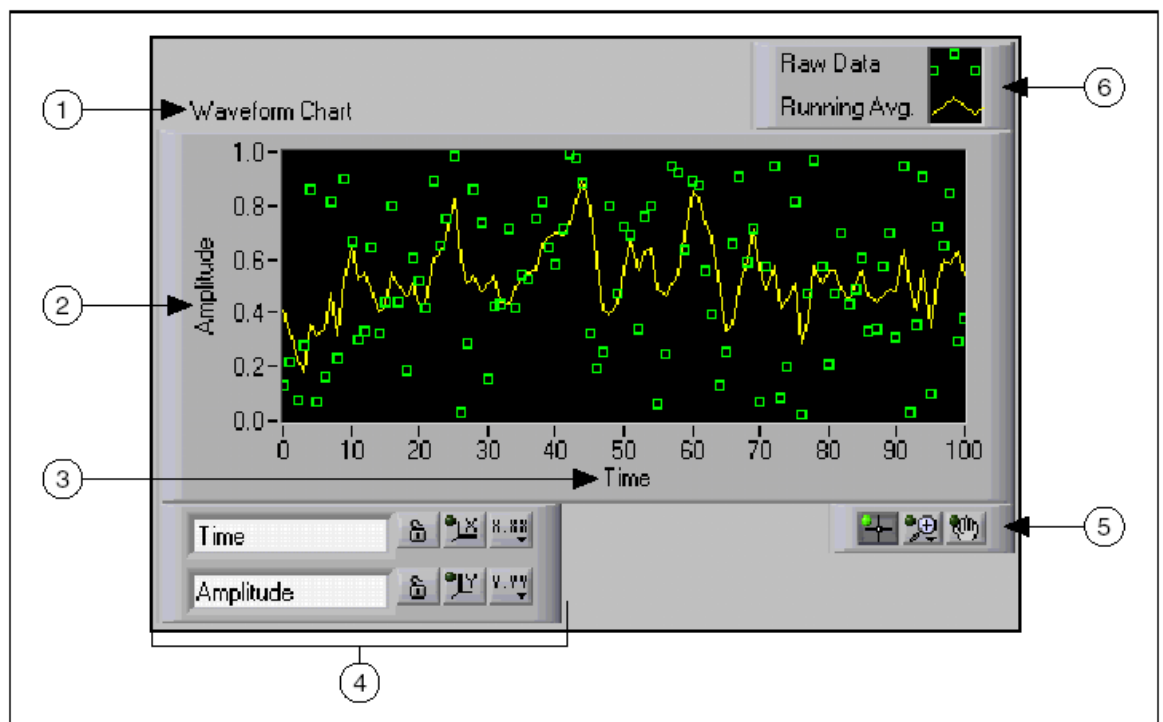


Рисунок 3.5 – График Диаграмм

- (1) Название (Label)
- (2) Шкала Y (Y-scale)
- (3) Шкала X (X-scale)
- (4) Панель управления шкалами (Scale legend)
- (5) Палитра инструментов для работы с графиком (Graph palette)
- (6) Панель управления графиком (Plot legend)

График Диаграмм использует три различных режима отображения данных: strip chart, scope chart и sweep chart. Режим по умолчанию – strip chart.

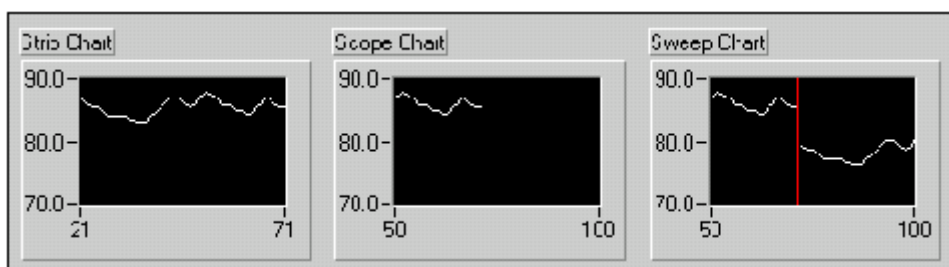


Рисунок 3.6 – Задание режима для графика диаграмм

Задание режима осуществляется щелчком правой клавишей мыши по диаграмме и выбором пункта `Advanced>>Update Mode` из контекстного меню.

Режим strip chart представляет собой экран, прокручиваемый слева направо, подобно бумажной ленте. Режимы scope chart и sweep chart подобны экрану осциллографа и отличаются большей скоростью отображения данных по сравнению с strip chart. В режиме scope chart по достижении правой границы поле графика очищается, и заполнение диаграммы начинается с левой границы. Режим sweep chart, в отличие от режима scope chart, не очищает поле графика, а отделяет новые данные от старых вертикальной линией – маркером.

Соединение графиков.

Для создания диаграмм достаточно соединить поле вывода скалярной величины с терминалом данных графика Диаграмм.

График Диаграмм может отображать несколько графиков. Для объединения отображаемых данных используется функция `Bundle`, расположенная в палитре `Functions>>Cluster`. Например, блок-диаграмма, показанная ниже, с помощью функции `Bundle` объединяет выходные данные трех подпрограмм ВП для последующего отображения на графике Диаграмм.

Задание

Необходимо сгенерировать периодический синусоидальный сигнал с амплитудой «U», фазой «Alfa», частотой «f» согласно номеру варианта по таблице 3.7.

Таблица 3.1 – Параметры периодического сигнала

A	B	U	Alfa	f
0	0	100	30	50
1	1	110	45	60
2	2	120	60	100
3	3	130	75	120
4	4	140	90	145
5	5	150	105	170
6	6	160	120	195
7	7	170	135	220
8	8	180	150	245
9	9	190	165	270

Примечание: A – первая цифра варианта, B – вторая цифра варианта.

Отобразить на графике один период заданного сигнала, найти RMS значение на периоде.

Записать в текстовый файл мгновенные значения половины периода генерируемого сигнала и найти среднее значение на полупериоде сигнала.

Выполнение работы

На рисунке 3.7 отображена лицевая панель виртуального прибора, несущая в себе элементы управления параметрами сигнала: амплитудой, фазой и частотой, а так же элементы отображения сигнала в графическом виде и вычисляемые значения количественной оценки исследуемого сигнала.

Так же на лицевой панели имеется диалоговое окно указывающее название и местоположение файла с записанными в него мгновенными значениями.

Блок диаграмма виртуального прибора (Рисунок 3.8) несет функциональную часть прибора и содержит: элементы ввода и отображения

рабочих переменных, экспресс функцию генерации сигнала, функции анализа и записи мгновенных значений сигнала в текстовый файл.

Результат выполнения лабораторной работы.

Содержание отчета

Отчет оформляется каждым студентом самостоятельно.

Демонстрация работы программы на ЭВМ.

Содержание отчета:

1. Титульный лист.
2. Задание.
4. Лицевая панель и структурная схема прибора, разработанного по пункту задания.
5. Описание всех элементов лицевой панели и структурной схемы прибора.
6. Лицевая панель прибора и структурная схема прибора, разработанного по пункту задания; объяснения принципов его функционирования; распечатка файла с результатами работы программы.

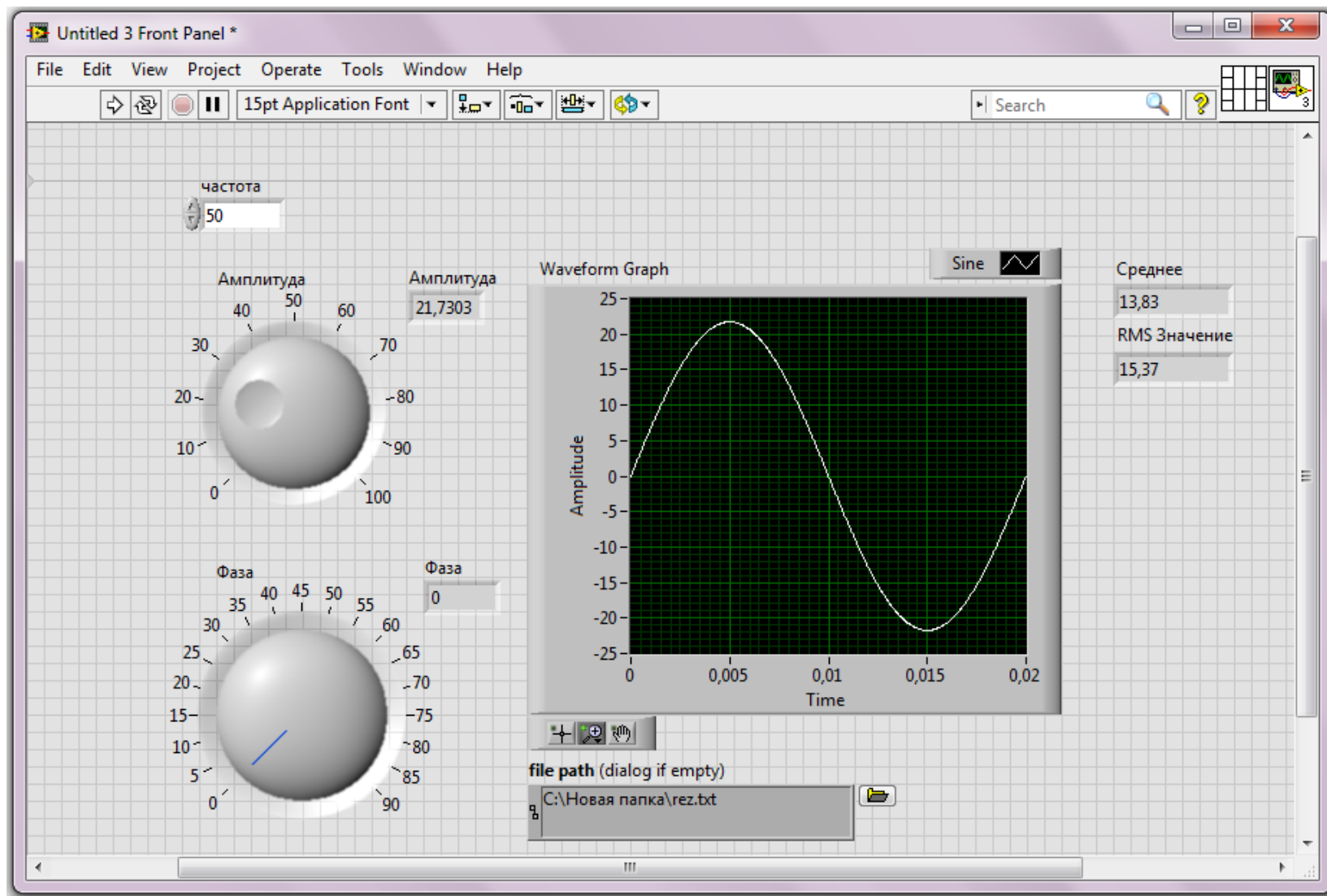


Рисунок 3.7 – Фронтальная панель виртуального прибора

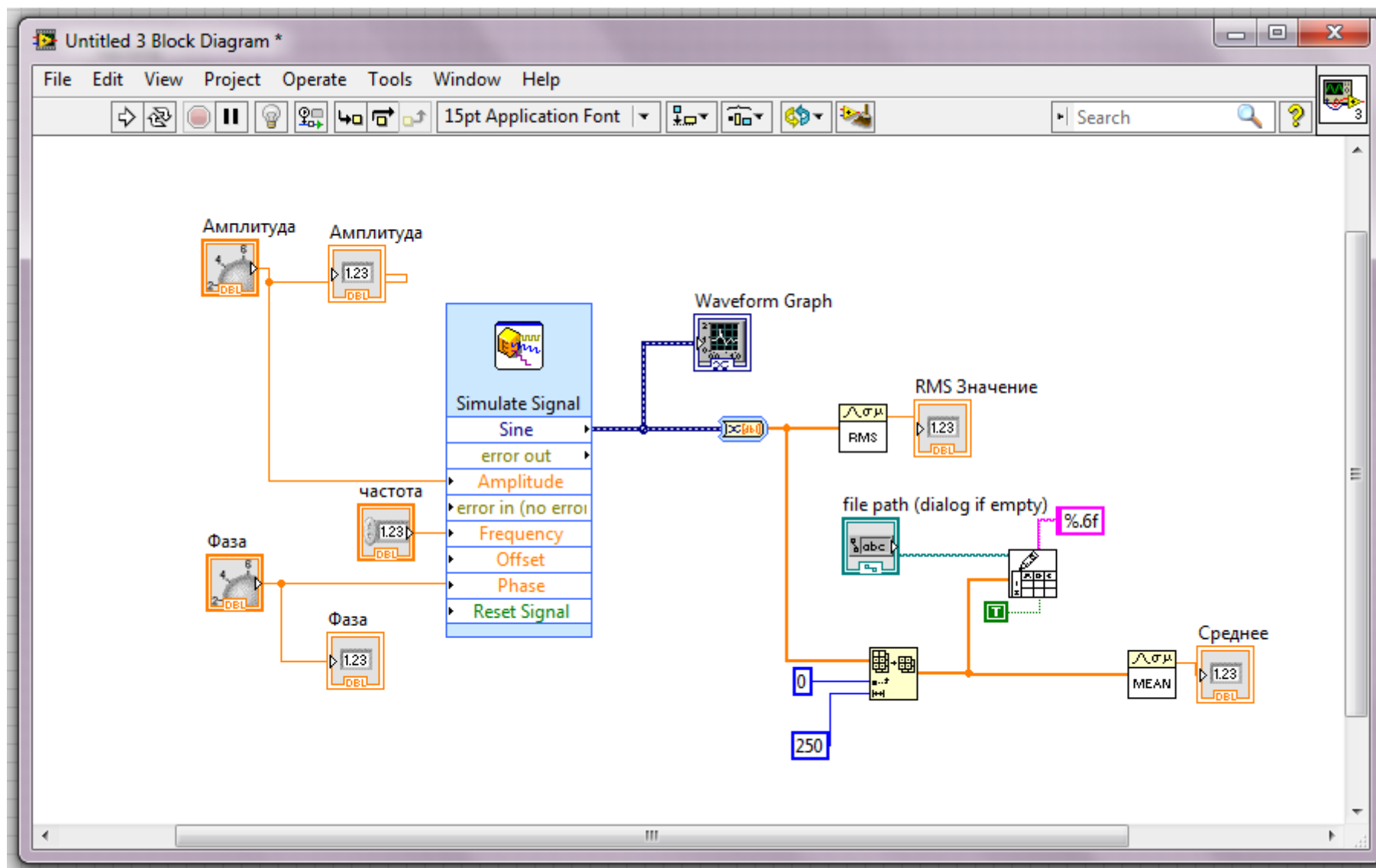


Рисунок 3.8 – Блок диаграмма виртуального прибора

Контрольные вопросы

1. Что такое массив данных?
2. Чему равен индекс первого элемента массива?
3. В какой палитре находится шаблон для создания массива элементов управления или отображения?
4. Что такое автоиндексация в циклах For или While, и как ей управлять?
5. В какой палитре расположены функции использующиеся для создания и управления массивами?
6. LabVIEW осуществляет автоиндексацию автоматически?
7. Для какого из типов циклов автоиндексация включена по умолчанию?
8. Что означает термин полиморфизм в Labview?
9. Что такое график Диаграмм (Waveform Chart)?
10. Какие три режима отображения имеет график Диаграмм?

Литература: 2, 3, 4.

Лабораторная работа №4

Измерение среднего значения

Цель работы: ознакомление с функциями статистической обработки данных в темпе процесса измерения при помощи палитр функций LabView. В связи с этим необходимо решить следующие задачи:

1. Выполнить построение кода формирующего массивы случайных данных с заданным законом распределения согласно заданию.
2. Используя функции статистической обработки данных выполнить поиск требуемых численных оценок среднего значения, и стандартного отклонения.

Знания, умения и владения, приобретаемые обучающимся в результате освоения тем в рамках освоения компетенций: способностью осуществлять поиск, хранение, обработку и анализ информации из различных источников и баз данных, представлять ее в требуемом формате с использованием информационных, компьютерных и сетевых технологий (ОПК-1); способностью применять соответствующий физико-математический аппарат, методы анализа и моделирования, теоретического и экспериментального исследования при решении профессиональных задач (ОПК-2)

Знать: о связи дисциплины с другими изучаемыми дисциплинами; о роли дисциплины в дальнейшей научной и практической деятельности; о возможностях и преимуществах моделирования в среде LabView; терминологию, основные понятия и определения; технологию графического программирования в среде LabView; типы данных и последовательность их обработки (ОПК-1); общие вопросы построения измерительных систем с использованием: современных компьютерных технологий, принципы решения технических задач; измерений с применением современных аппаратных и программных средств (ОПК-2);

Уметь: формулировать постановку задачи, определять цель (ОПК-1); оптимально выбирать программно-аппаратные средства для решения задач и проблем измерительной техники; разработать алгоритмы измерения и обработки данных (ОПК-2);

Владеть: терминологией, основными понятиями и определениями; технологией графического программирования в среде LabView; навыками обработки различных типов данных; общими вопросами построения измерительных систем с использованием современных компьютерных технологий (ОПК-1); принципами решения технических задач измерений с применением современных аппаратных и программных средств (ОПК-2);

Теоретическая часть

Функции генерации сигналов и шумов.

Функции генерации и обработки сигналов, размещенные в подпалитре Обработка сигналов, позволяют формировать как отдельные значения, так и массивы отсчетов сигналов и шумов и производить их обработку с помощью различных операций в частотной и временной областях.

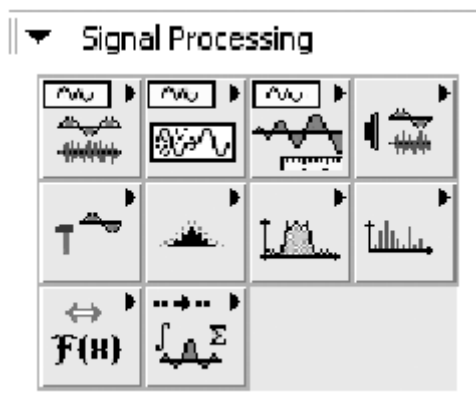


Рисунок. 4.1 – Вид категории функций Обработка (Signal Processing)

ВП из палитры генерации сигналов и шумов (рис. 4.2) используются для формирования детерминированных и случайных сигналов с заданным набором параметров. Первые два ВП в верхнем ряду представляют многофункциональные генераторы сигналов с широким набором

регулируемых параметров. ВП, размещенные во второй и третьей строках, предназначены для генерации наиболее широко применяемых детерминированных периодических сигналов, а ВП, находящиеся в четвертой и пятой строках, служат для расчета шумов с различными законами амплитудного и спектрального распределения.

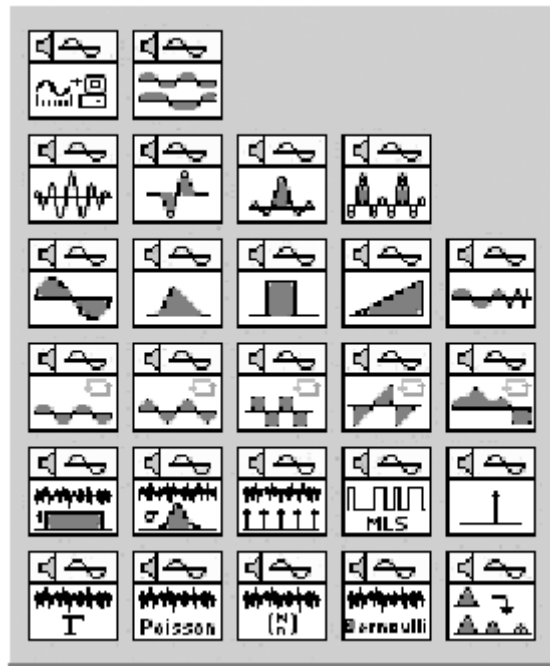


Рисунок 4.2 – Палитра функций генерации сигналов и шумов.

Описание функций генерации сигналов и шумов приведено далее.

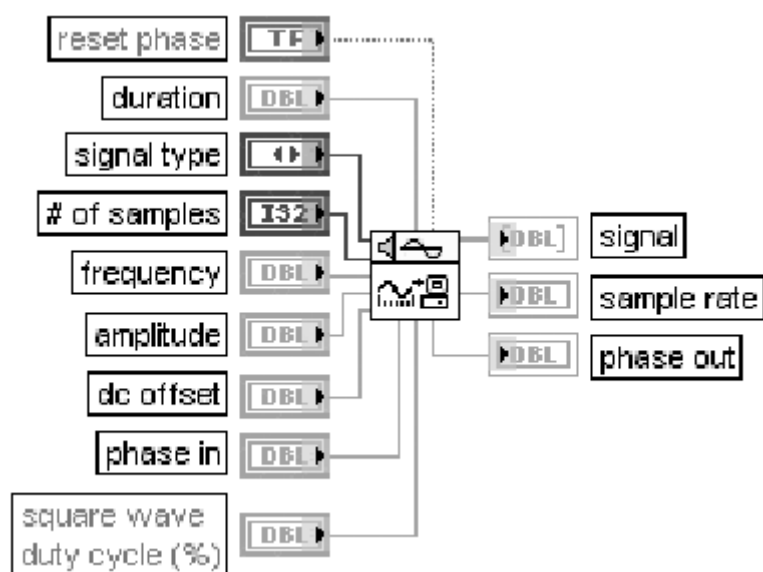


Рисунок 4.3 – Signal Generator by Duration

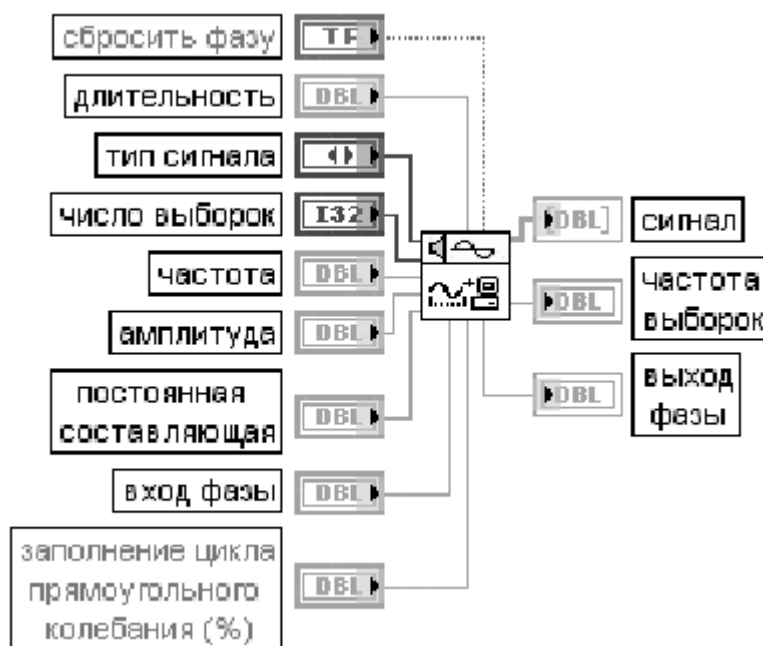


Рисунок 4.4 – Генератор сигналов с заданной длительностью

ВП генерирует сигнал `сигнал (signal)`, имеющий форму, задаваемую на входе `тип сигнала (signal type)`.

Вход *сбросить фазу* (`reset phase`) определяет начальную фазу выходного сигнала. При установке на входе состояния ИСТИНА (по умолчанию) начальная фаза сигнала определяется входом *вход фазы* (`phase in`). В противном случае начальная фаза устанавливается равной значению фазы на выходе *выход фазы* (`phase out`) при последнем выполнении этого ВП.

Вход *длительность* (`duration`) задает время в секундах, равное длительности генерируемого выходного сигнала. По умолчанию значение длительности равно 1,0.

Вход *тип сигнала* (`signal type`) задает следующие типы генерируемого сигнала:

Синусоидальный (`sine`) (по умолчанию);

Косинусоидальный (`cosine`);

Треугольный (`triangle`)

Прямоугольный (square);
Пилообразный (sawtooth);
Линейно нарастающий (increasing ramp);
Линейно спадающий (decreasing ramp).

Вход *число выборок* (# of samples) задает число выборок выходного сигнала. По умолчанию это значение равно 100.

Вход *частота* (frequency) определяет частоту выходного сигнала в герцах. По умолчанию значение частоты равно 10. При задании частоты необходимо учитывать требование выполнения критерия Найквиста: частота $< \text{число выборок} / (2 * \text{длительность})$.

Вход *амплитуда* (amplitude) задает амплитуду выходного сигнала. По умолчанию значение амплитуды равно 1,0.

Вход *постоянное смещение* (dc offset) задает постоянное смещение или значение постоянной составляющей выходного сигнала. По умолчанию значение постоянной составляющей равно 0.

Вход *Вход фазы* (phase in) определяет начальную фазу (в градусах) выходного сигнала при установке входа *сбросить фазу* (reset phase) в состояние ИСТИНА. По умолчанию значение на *входе фазы* равно 0.

Вход *заполнение цикла прямоугольного колебания* (square wave duty cycle) определяет время (в процентах от периода), в течение которого прямоугольный сигнал имеет высокий уровень. ВП использует данный параметр только для прямоугольного сигнала. По умолчанию значение на входе равно 50%.

Выход *сигнал* (signal) представляет сгенерированный массив выборок сигнала.

Выход *частота выборок* (sample rate) отображает частоту дискретизации выходного сигнала. Частота выборок равна отношению числа выборок к длительности длительности.

Выход фазы (phase out) указывает значение фазы (в градусах) последней выборки выходного сигнала

Tones and Nonise – Гармонические колебания и шум.

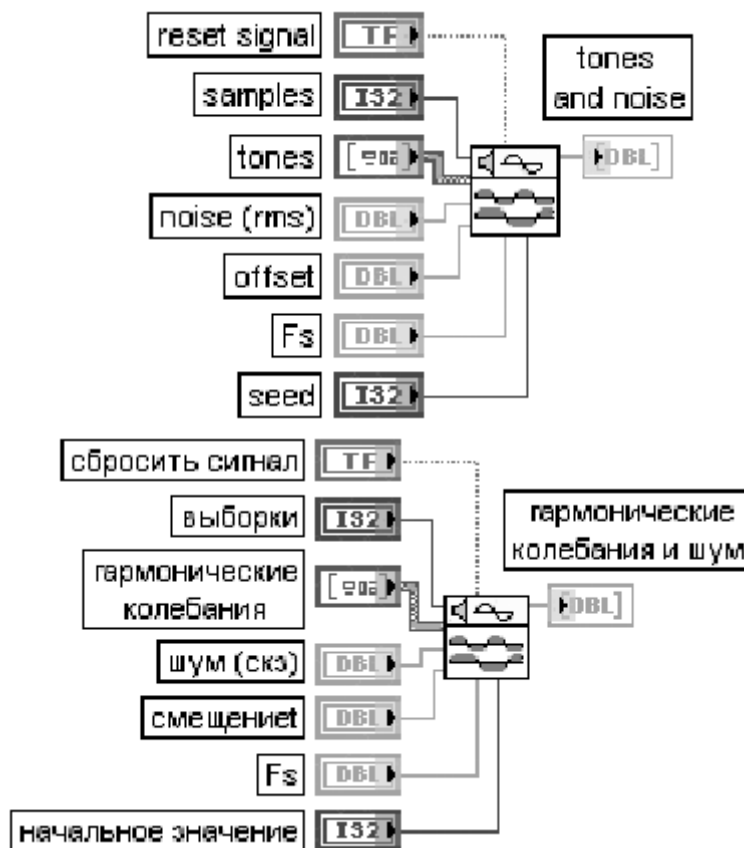


Рисунок 4.5 – Tones and Nonise – Гармонические колебания и шум

ВП генерирует массив, содержащий сумму гармонических колебаний, шум и постоянную составляющую.

Вход сбросить сигнал сбросить сигнал сбросить сигнал сбросить сигнал сбросить сигнал (reset signal) при подаче значения ИСТИНА устанавливает фазу каждого гармонического колебания равной значению фазы (phase) из массива гармонические колебания (tones), значение начального числа при генерации шума равным значению входа начальное значение (seed) и отметку времени равной нулю. По умолчанию состояние входа – ЛОЖЬ.

Вход `samples` определяет число выборок выходного массива гармонические колебания и шум (`tones and noise`). По умолчанию это число равно 1000.

Вход гармонические колебания гармонические колебания гармонические колебания гармонические колебания гармонические колебания (`tones`), являющийся кластером, содержит параметры каждого гармонического колебания:

- частота (`frequency`) определяет частоту синусоидального колебания в герцах;
- амплитуда (`amplitude`) определяет амплитуду синусоидального колебания;
- фаза (`phase`) определяет начальную фазу синусоидального колебания.

По умолчанию фаза равна 0

Вход шум (`noise`) определяет среднеквадратичное значение (с.к.з.) аддитивного гауссовского шума. По умолчанию значение шума равно 0,0.

Вход смещение (`offset`) задает уровень постоянной составляющей сигнала. По умолчанию уровень равен 0,0.

Вход `Fs` определяет частоту выборок в единицах число выборок/с. По умолчанию частота равна 1000.

Установка на входе начальное значение начальное значение начальное значение (`seed`) значения >0 вызывает инициализацию генератора аддитивного шума. По умолчанию значение входа равно -1 . Если начальное значение начальное значение начальное значение меньше или равно 0, то генератор шума не инициализируется и продолжает генерацию шума на основе предыдущих значений.

Выход гармонические колебания и шумы гармонические колебания и шумы гармонические колебания и шумы гармонические колебания и шумы гармонические колебания и шумы (`tones and noise`) отображает сгенерированный выходной массив.

Функции статистической обработки данных

Функции основной палитры статистической обработки данных (рисунок 4.6а) позволяют рассчитать основные статистические параметры наборов данных: среднее, среднеквадратическое отклонение, дисперсию, среднеквадратичное значение, центральные моменты различного порядка, медиану и моду, а также рассчитать значения гистограммы набора данных. Дополнительные подпалитры (рисунок 4.6б–4.6г) содержат функции расчета вероятностей (интегральных распределений) для различных законов распределения и функции расчета квантилей распределений по заданным значениям вероятности, проверки гипотез и дисперсионного анализа.

Ниже приведено описание функций основной палитры статистической обработки данных.

Mean



Среднее

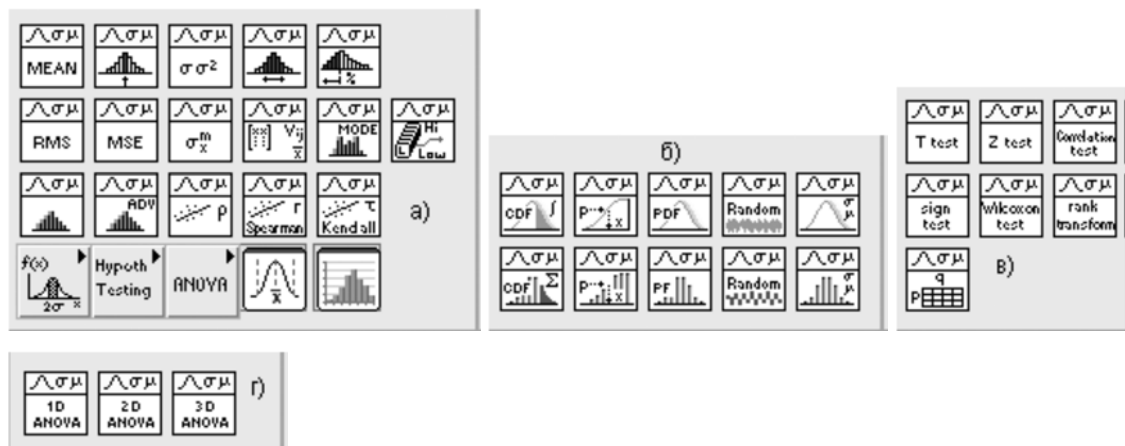


Рисунок 4.6 – Вид основной палитры (а) и дополнительных подпалитр (б) – (г) функций статистической обработки данных

Задание

Необходимо создать виртуальный прибор управляющий конфигурированием, запуском и отображением генерируемых и вычисляемых величин.

Прибор должен генерировать произвольное множество значений по заданному закону распределения, и вычислять среднее значение из множества и стандартное отклонение, а также отображать в графической форме этих данные на графике в масштабе.

Так же необходимо выбрать из таблицы 4.1 согласно варианту исходное значение для генерации произвольного множества и закон распределения, задав их значения на фронтальной панели и на блок-диаграмме реализованного виртуального прибора определить среднее значение и стандартное отклонение для десяти опытов и занести значения в таблицу.

Выполнение работы

На рисунке 4.1 отображена лицевая панель виртуального прибора, несущая в себе элементы управления.

Таблица 4.1 – Варианты значений коэффициентов и типа распределения для генерации исходных последовательностей данных.

№ варианта		исходные значения для генерации		Тип распределения
1-я цифра варианта	2-я цифра варианта	среднее	разброс	
0	0	1	1	Нормальное (Normal)
1	1	2	2	Лапласа (Laplas)
2	2	3	3	Коши (Cauchy)
3	3	4	4	Экспоненциальное (Exp)
4	4	5	5	F-распределение (F)

5	5	6	6	Гамма (Gamma)
6	6	7	7	Парето- распределение (Pareto)
7	7	8	8	Треугольное (Triangular)
8	8	9	9	Формальное (Uniform)
9	9	10	10	Вейбула распределение (Weibull)

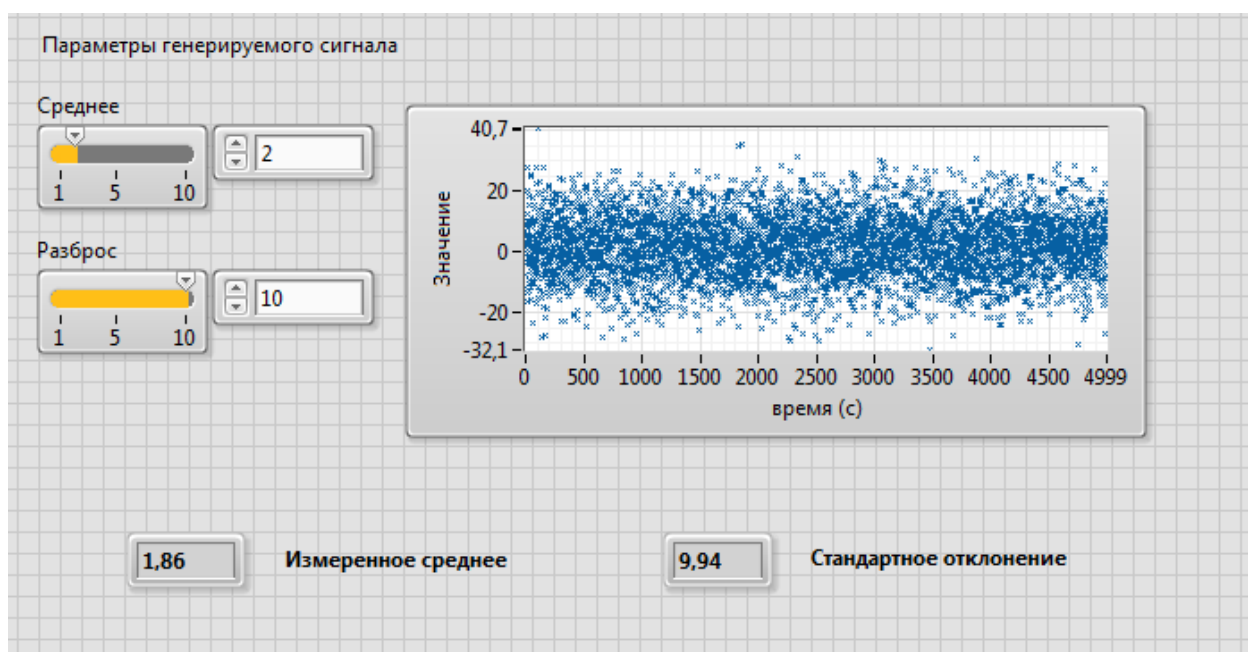


Рисунок 4.1 – Фронтальная панель виртуального прибора

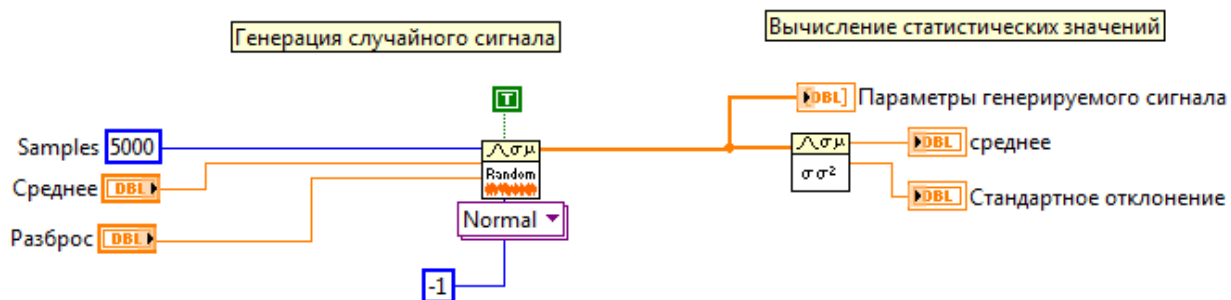


Рисунок 4.2 – Блок-диаграмма виртуального прибора

Результат выполнения работы.

В результате выполнения работы необходимо создать виртуальный прибор по выше предлагаемому заданию и написать отчет о лабораторной работе.

Отче о лабораторной работе должен содержать:

1. Титульный лист;
2. Задание с указанием варианта;
3. Принтскрины фронтальной панели и блок диаграммы виртуального прибора;
4. Значения среднего и стандартного отклонения.

Лабораторная работа №5

ВП «Кофейный аппарат»

Цель работы: освоение технологий работы с логическими переменными и построения систем обработки логики событий при помощи функционала LabView. В связи с этим необходимо решить следующие задачи:

1. Используя логические функции выполнить построение кода повторяющего работу автомата по приготовлению горячих напитков «Кофейного аппарата».

2. Продемонстрировать работу программы по произвольному выбору напитка.

Знания, умения и владения, приобретаемые обучающимся в результате освоения тем в рамках освоения компетенций: способностью осуществлять поиск, хранение, обработку и анализ информации из различных источников и баз данных, представлять ее в требуемом формате с использованием информационных, компьютерных и сетевых технологий (ОПК-1); способностью применять соответствующий физико-математический аппарат, методы анализа и моделирования, теоретического и экспериментального исследования при решении профессиональных задач (ОПК-2)

Знать: о связи дисциплины с другими изучаемыми дисциплинами; о роли дисциплины в дальнейшей научной и практической деятельности; о возможностях и преимуществах моделирования в среде LabView; терминологию, основные понятия и определения; технологию графического программирования в среде LabView; типы данных и последовательность их обработки (ОПК-1); общие вопросы построения измерительных систем с использованием: современных компьютерных технологий, принципы решения технических задач; измерений с применением современных аппаратных и программных средств (ОПК-2);

Уметь: формулировать постановку задачи, определять цель (ОПК-1); оптимально выбирать программно-аппаратные средства для решения задач и проблем измерительной техники; разработать алгоритмы измерения и обработки данных (ОПК-2);

Владеть: терминологией, основными понятиями и определениями; технологией графического программирования в среде LabView; навыками обработки различных типов данных; общими вопросами построения измерительных систем с использованием современных компьютерных технологий (ОПК-1); принципами решения технических задач измерений с применением современных аппаратных и программных средств (ОПК-2);

Теоретическая часть

Числовые функции и функции манипуляции данными.

Числовые функции используются для выполнения арифметических и комплексных операций с числовыми данными, а также для преобразования типов числовых данных. На рис. 2.2 показан вид основной палитры числовых функций (рис. 5.2а) и дополнительных подпалитр Преобразование (Conversion) (рис. 5.2б), Функции манипуляции данными (Data Manipulation) (рис. 5.2в), Комплексные (Complex) (рис. 5.2г) и Математические и научные константы (Math & Scientific Constants) (рис. 5.2д).

Формула (Formula)

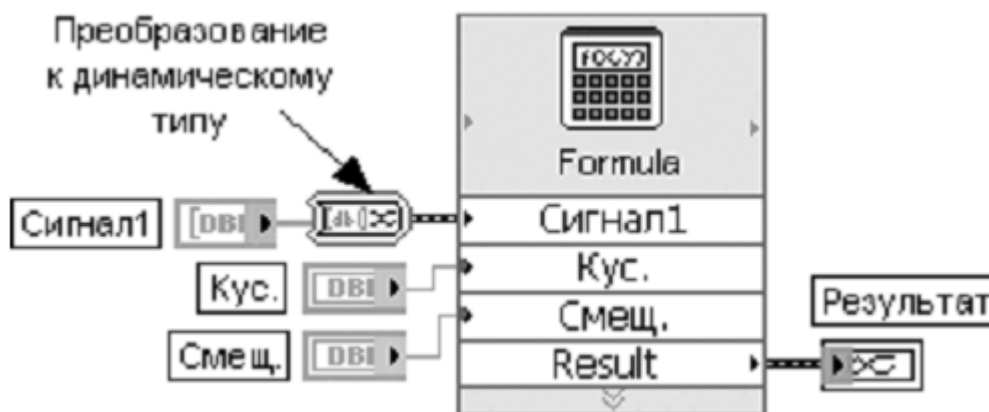


Рисунок 5.1 – Блок-диаграмма варианта подключения Экспресс-ВП

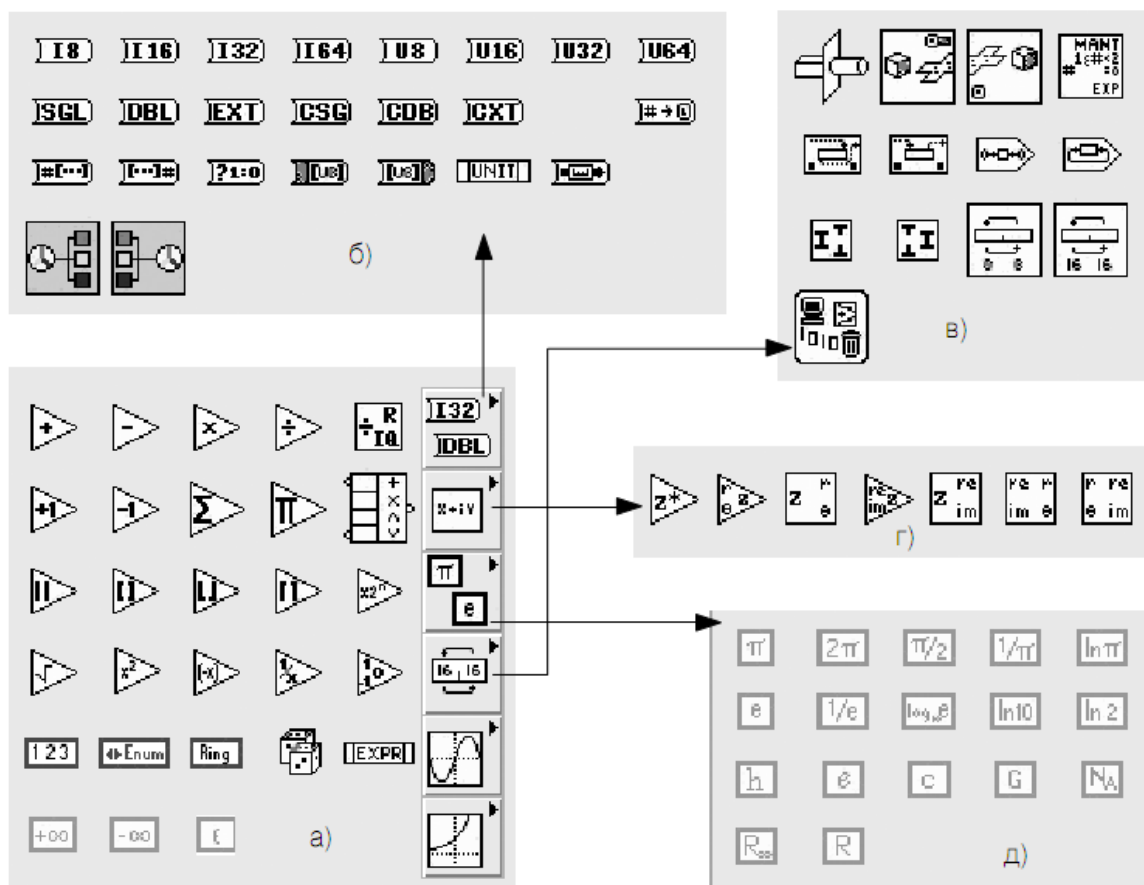


Рисунок 5.2 – Вид основной палитры (а) и дополнительных подпалитр (б)–(д) числовых функций

Экспресс-ВП Формула Формула (Formula) (рис. 5.1) позволяет производить математическую обработку входных данных, обеспечиваемую базовыми научными калькуляторами. Входы и выходы Экспресс-ВП рассчитаны на подключение данных динамического типа. При подключении данных другого типа производится преобразование типов, в том числе и с помощью Экспресс-ВП Преобразование в данные динамического типа (Convert to Dynamic Data) (рисунок 5.1).

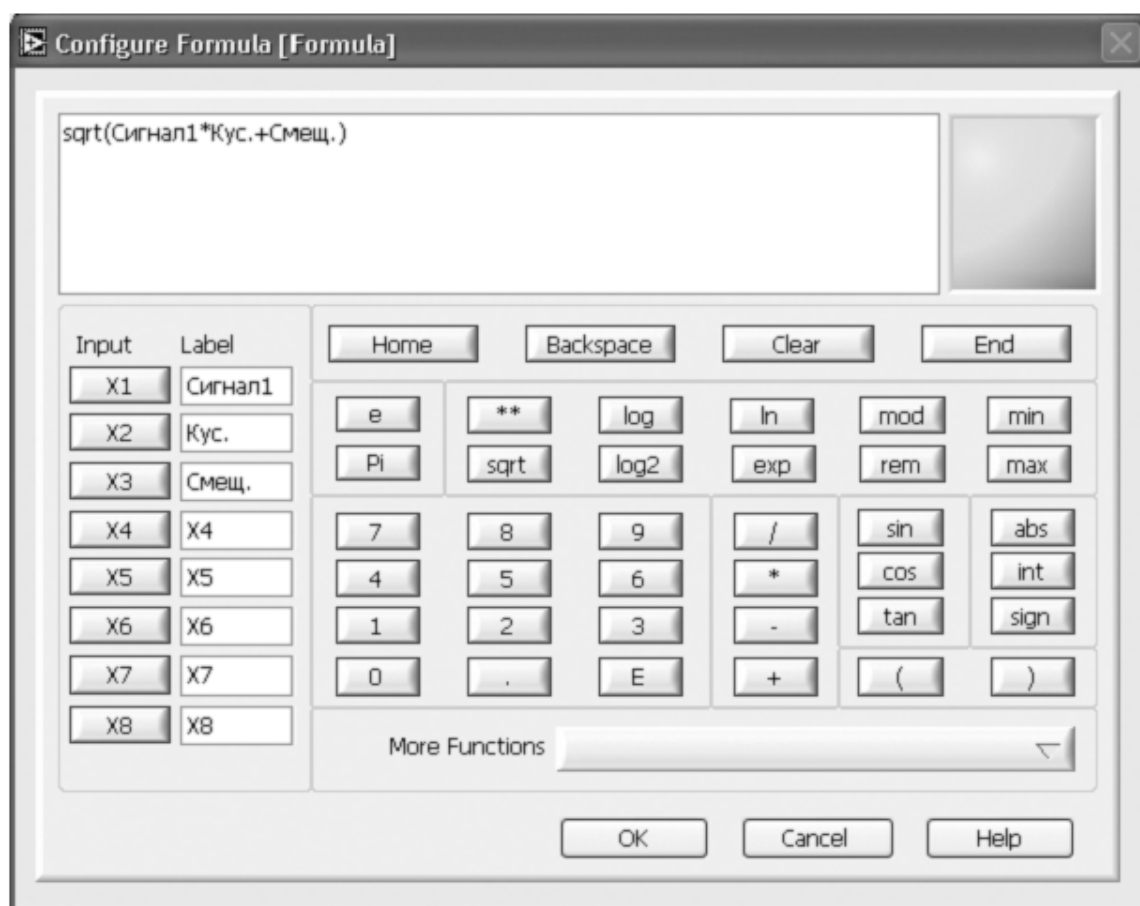


Рисунок 5.3 – Вид диалогового окна конфигурирования Экспресс-ВП Формула (Formula)

В Экспресс-ВП Формула Формула используется функциональность следующих ВП: Узел Выражение (Expression Node), Узел Формула (Formula Node), Оценка узла (Eval Formula Node), Оценка строки формулы (Eval Formula String), Оценка массива значений параметрической функции (Eval Multi-Variable Array).

Логические функции.

Логические функции (рисунок 5.4) используются для выполнения логических операций над значениями как скаляров, так и массивов логических величин. Функции И (And), ИЛИ (Or), Исключающее ИЛИ (Exclusive Or), НЕ (Not), И(НЕ (Not And), ИЛИ(НЕ (Not Or), Исключающее ИЛИ (НЕ (Not Exclusive Or), Исключение (Implies) являются полиморфными. Оба входа этих функций должны иметь логические или числовые значения, причем эти значения могут быть скалярами, массивами или кластерами. При

обработке числовых значений перечисленные функции выполняют побитовую обработку чисел. Ниже в таблице приведены пояснения к набору логических функций.

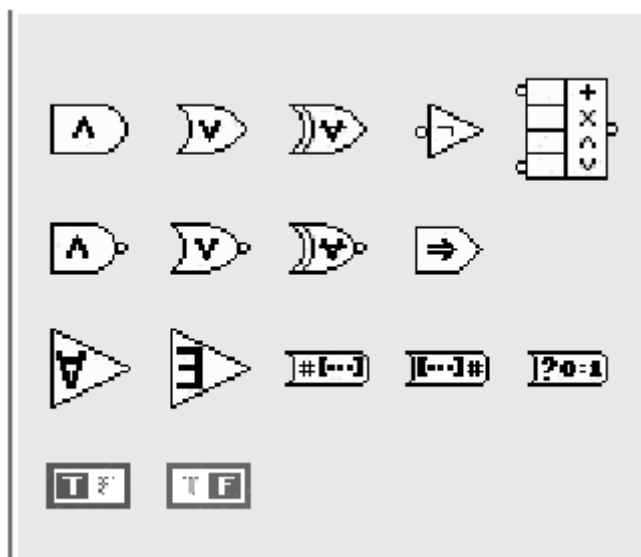


Рисунок 5.4 – Вид палитры логических функций

Логическая функция И (AND)

Возвращает значение ИСТИНА (1) только при подаче на оба входа значений ИСТИНА (1), иначе возвращает значение ЛОЖЬ (0).

На рисунке 5.5а приведены примеры использования данной функции для обработки скаляров логического типа, на рисунке 5.5б – скаляров числового типа, на рисунке 5.5в – массивов числовых значений

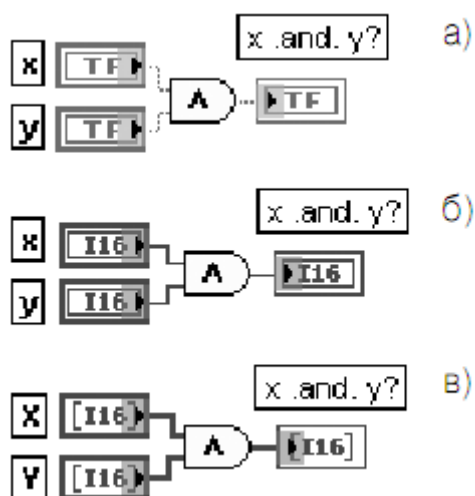
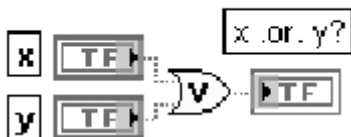
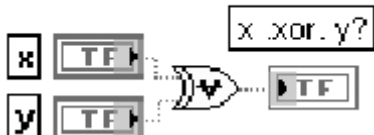
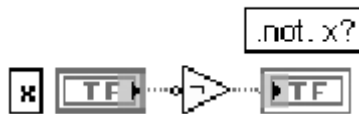
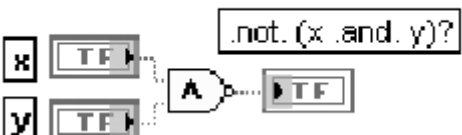
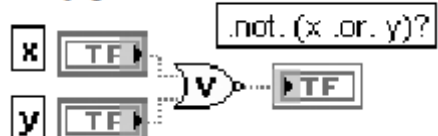
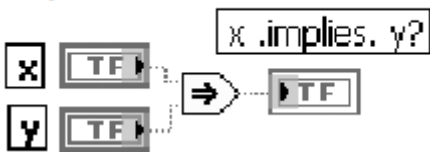
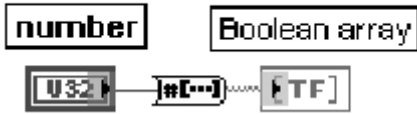
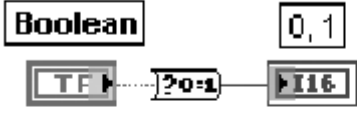
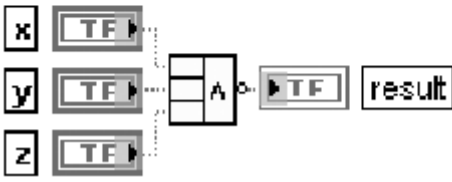


Рисунок 5.5 – Варианты подключения функции И

Таблица 5.1 – Логические функции

Функция	Описание действий
Or 	Логическая функция ИЛИ Логическая функция ИЛИ Логическая функция ИЛИ Логическая функция ИЛИ Возвращает значение ЛОЖЬ (0) только при подаче на оба входа значений ЛОЖЬ (0), иначе возвращает значение ИСТИНА (1)
Exclusive Or 	Логическая функция Иключающее ИЛИ Возвращает значение ЛОЖЬ (0) только при подаче на оба входа значений ЛОЖЬ (0) или значений ИСТИНА (1), иначе возвращает значение ИСТИНА (1)
Not 	Логическая функция НЕ Возвращает значение ЛОЖЬ (0) при подаче на вход значения ИСТИНА (1), и наоборот.
Not And 	Логическая функция И-НЕ Возвращает значение ЛОЖЬ (0) только при подаче на оба входа значений ИСТИНА (1), иначе возвращает значение ИСТИНА (1)
Not Or 	Логическая функция ИЛИ-НЕ Возвращает значение ИСТИНА (1) только при подаче на оба входа значений ЛОЖЬ (0), иначе возвращает значение ЛОЖЬ (0)
Not Exclusive Or 	Логическая функция Иключающее ИЛИ-НЕ Возвращает значение ИСТИНА (1) только при подаче на оба входа значений ЛОЖЬ (0) или значений ИСТИНА (1), иначе

	возвращает значение ЛОЖЬ (0)
Implies 	Исключение Инвертирует x и затем выполняет операцию. Логическое ИЛИ с входом y. Если на входе x значение ИСТИНА и на входе y значение ЛОЖЬ, то функция возвращает ЛОЖЬ. Иначе возвращает ИСТИНА
And Array Elements 	Логическая функция И для элементов Возвращает значение ИСТИНА, если все элементы Логического массива (Boolean array) имеют значение ИСТИНА. Иначе возвращает ЛОЖЬ. Функция воспринимает массивы любого размера.
Or Array Elements 	Логическая функция ИЛИ для элементов массива Возвращает ЛОЖЬ, если все элементы Логического массива (Boolean array) имеют значение ЛОЖЬ. Иначе возвращает ИСТИНА. Функция воспринимает массивы любого размера
Number To Boolean Array 	Число в логический массив Преобразует целое число в логический массив из 8, 16 или 32 элементов в зависимости от числа битов целого числа. Нулевой элемент логического массива соответствует младшему разряду двоичного представления числа.
Boolean Array To Number 	Логический массив в число. Преобразует Логический массив (Boolean array) в 32-битовое целое число без знака, интерпретируя массив как двоичное

	представление целого числа, причем нулевой элемент массива соответствует младшему биту числа.
Boolean To (0,1)  Compound Arithmetic 	Логический массив является одномерным массивом логических значений. Функция исключает часть логического массива, если он превышает заданную длину, и дополняет значениями ЛОЖЬ, если массив короткий Логическое значение в число преобразует логические значения ЛОЖЬ или ИСТИНА в 16-битовое целое число, имеющее значение соответственно 0 или 1. Составная арифметика. Функция позволяет выполнять логические операции И, ИЛИ и Исключающее ИЛИ с произвольным числом числовых или логических величин. Вид операции выбирается с помощью строки Изменить режим (Change Mode) контекстного меню функции. При этом на выходе и на любом входе функции с помощью строки Инвертировать (Invert) того же меню может быть установлена операция инверсии. Операция Исключающее ИЛИ входов более двух выполняется последовательно сначала с парой входов, затем с результатом и следующим входом и т. д.

Задание

Необходимо создать виртуальный прибор управляющий работой кофейного аппарата.

Прибор должен эмитировать работу аппарата по приготовлению горячих напитков в форме интерактивного интерфейса, генерируемого сигналы нагрева воды, приготовления напитка, а также отображать информацию о выборе напитка.

Выполнение работы

На рисунке 5.6 отображена лицевая панель виртуального прибора, несущая в себе элементы общения программы с оператором.

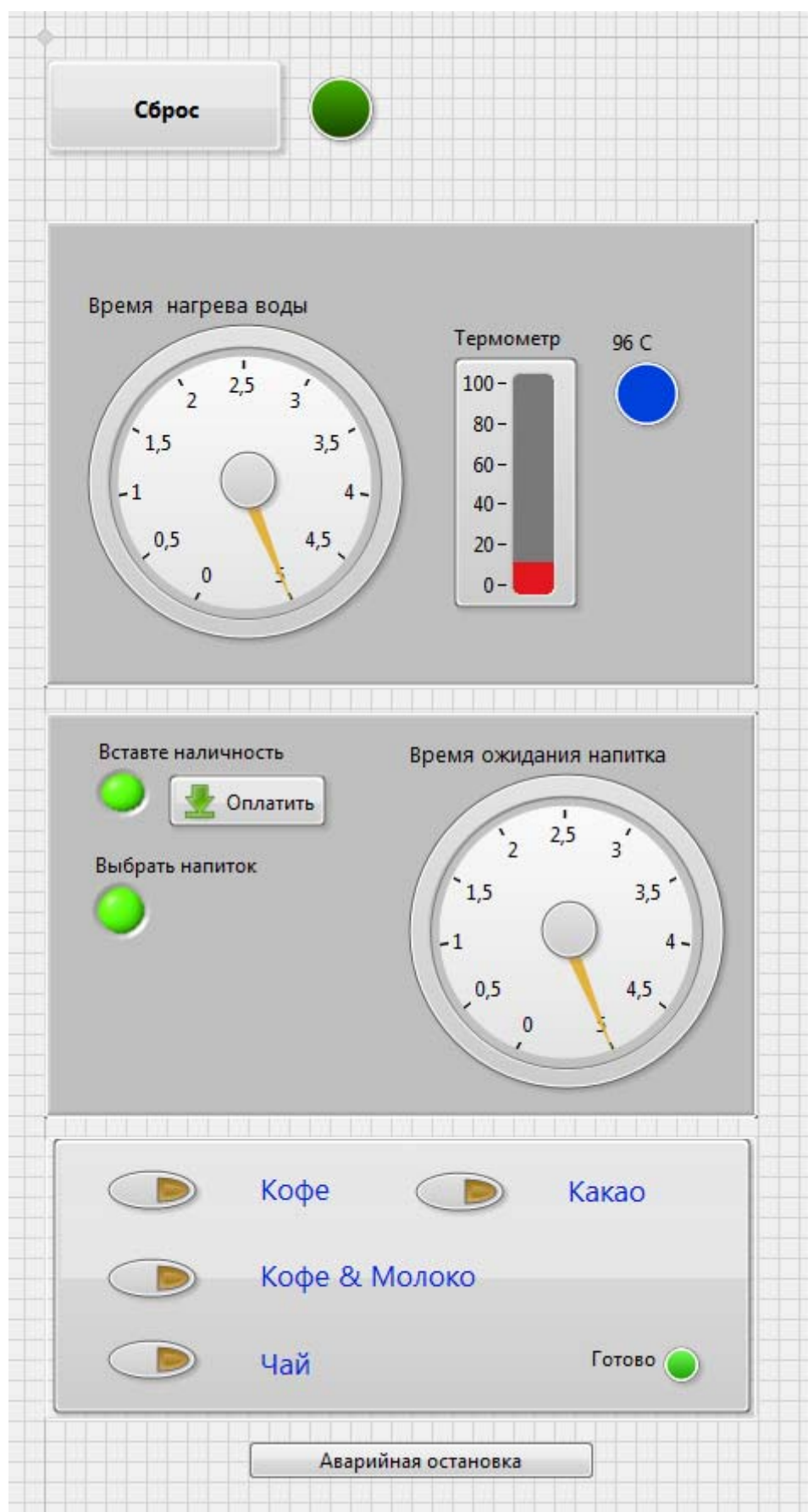


Рисунок 5.6 – Фронтальная панель виртуального инструмента

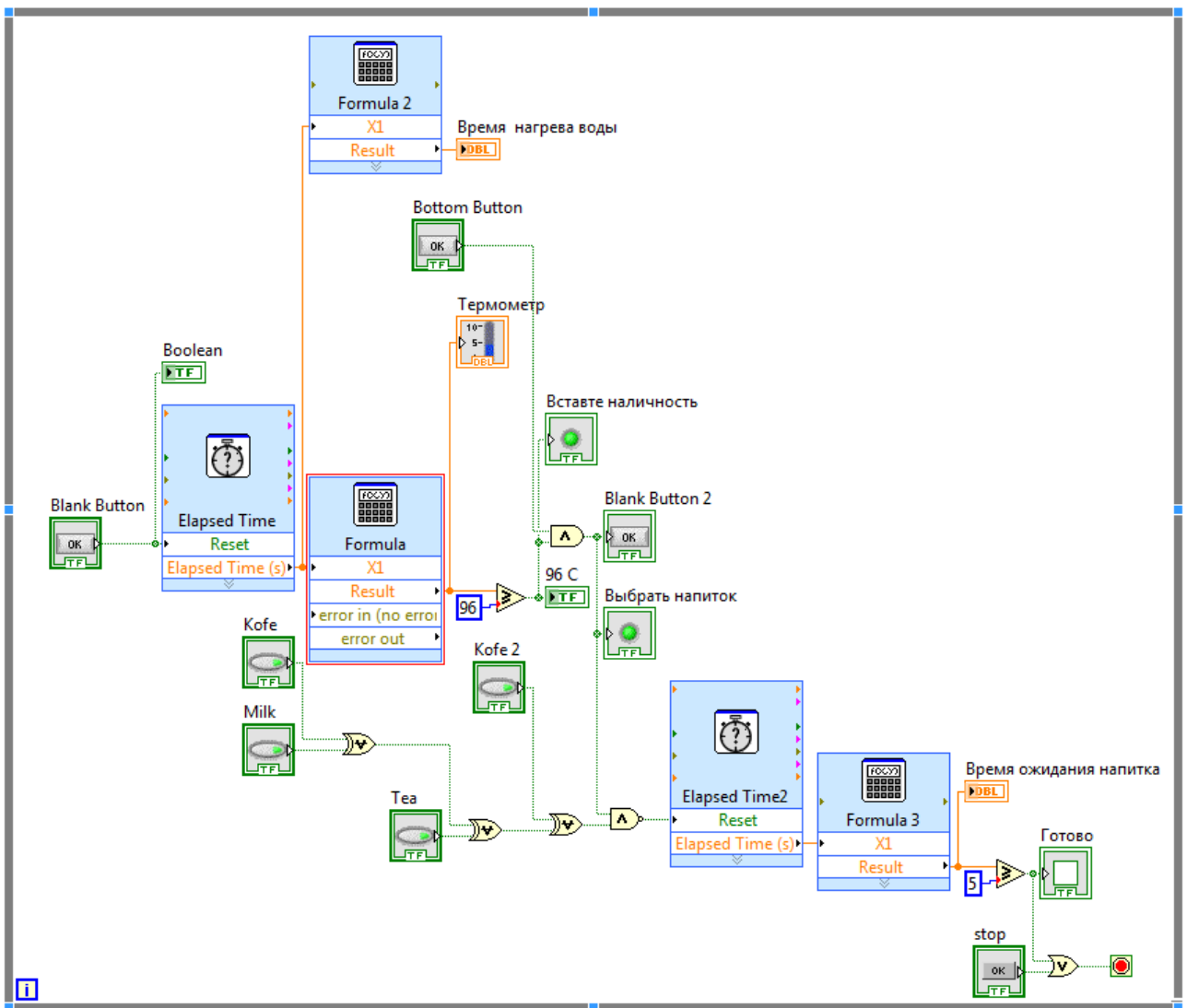


Рисунок 5.7 – Блок-диаграмма виртуального прибора.

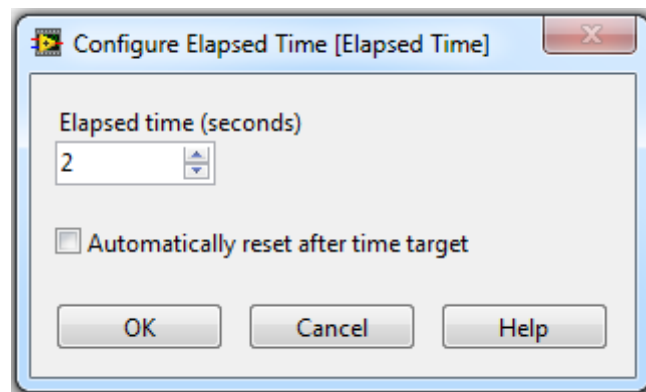


Рисунок 5.8 – Настройка Elapsed Time

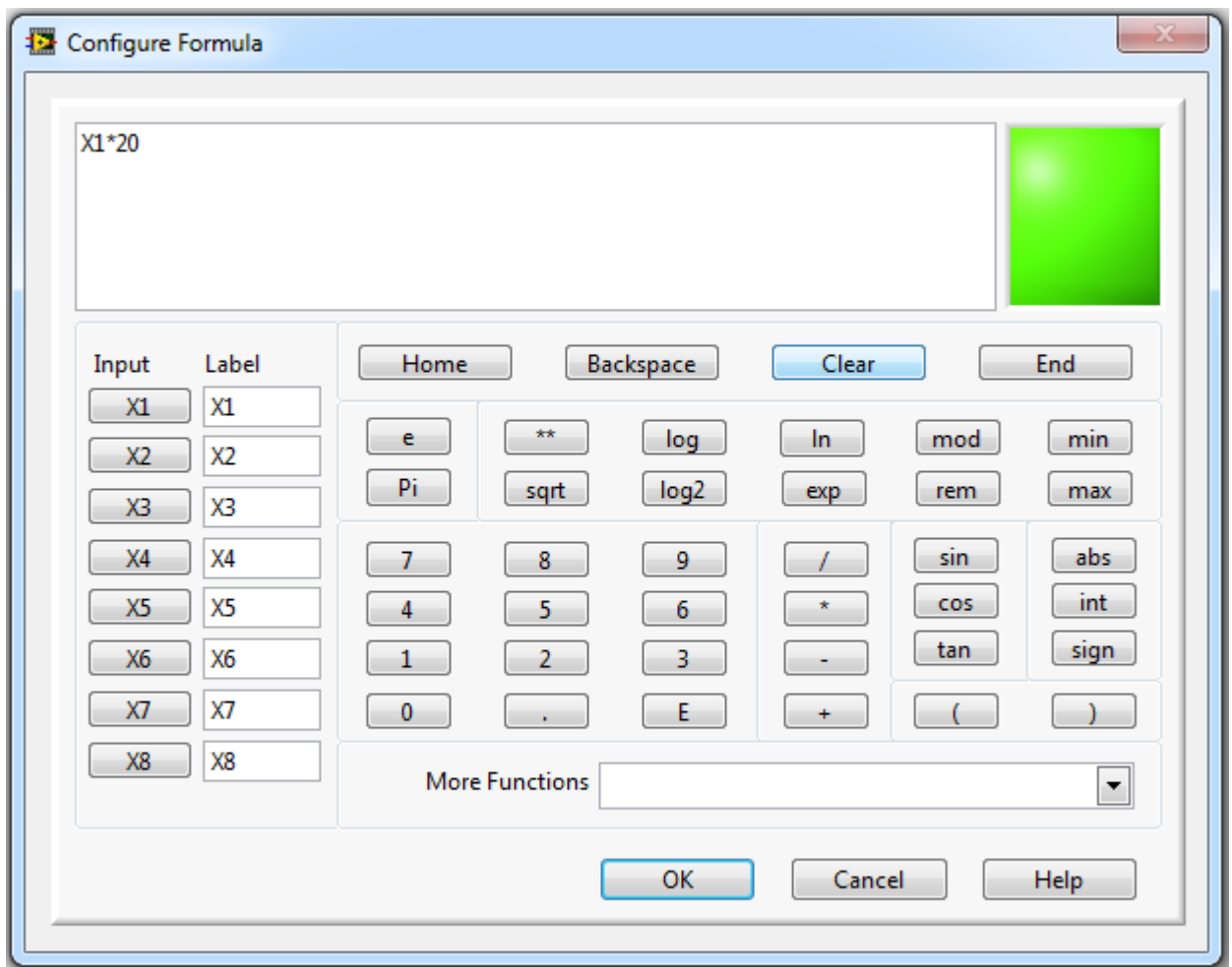


Рисунок 5.9 – Настройка Formula

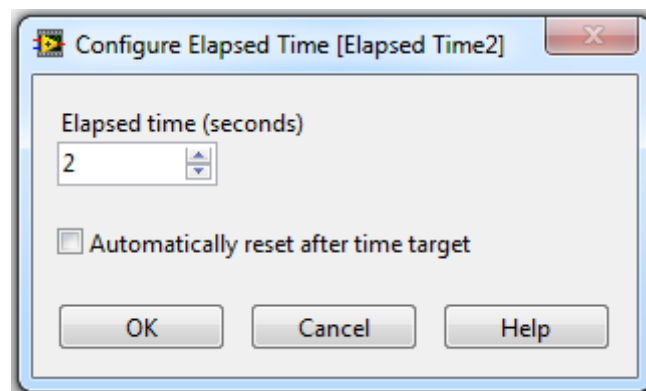


Рисунок 5.10 – Настройка Elapsed Time2

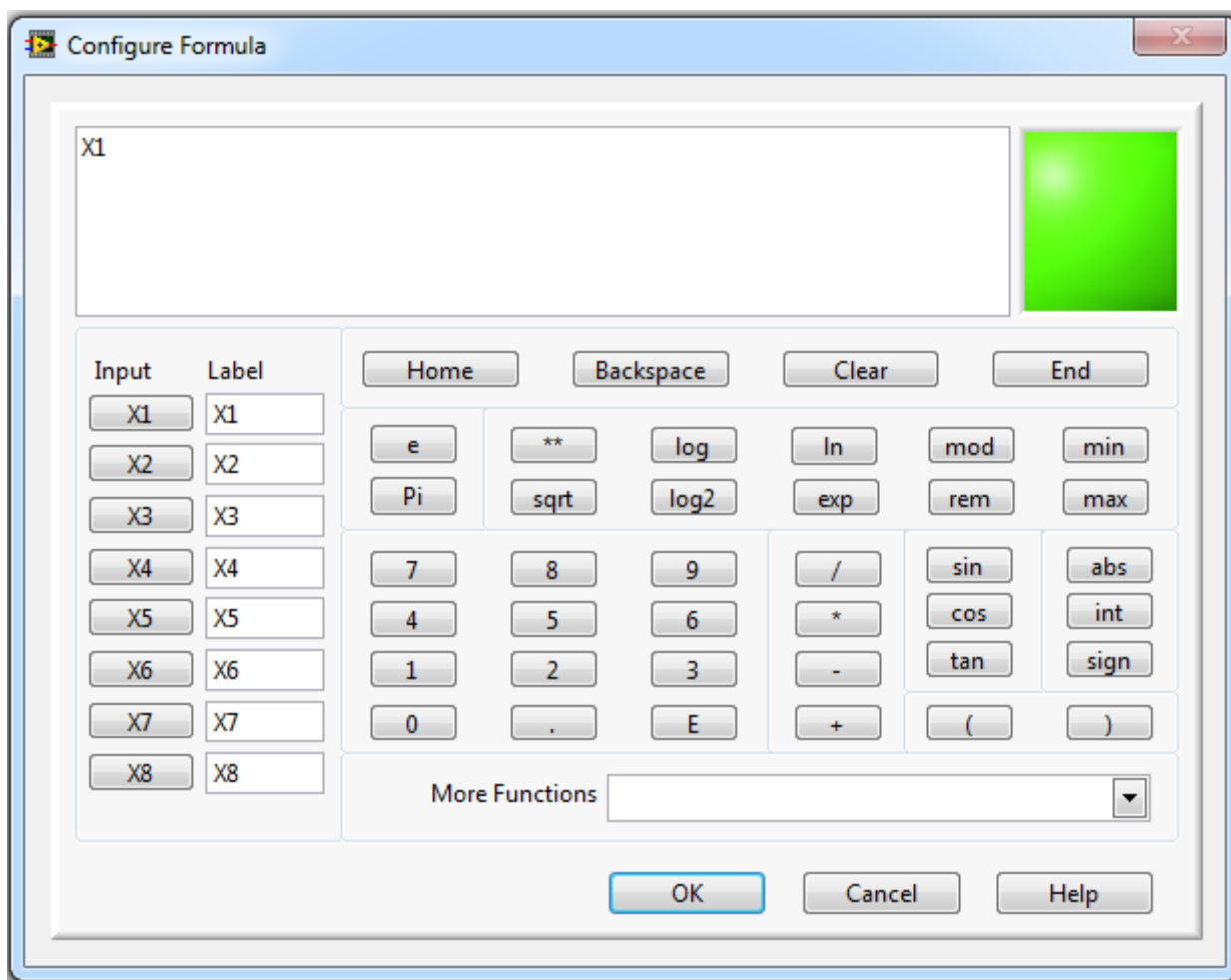


Рисунок 5.11 – Настройка Elapsed Time 1

Результат выполнения работы.

В результате выполнения работы необходимо создать виртуальный прибор по выше предлагаемому заданию и написать отчет о лабораторной работе.

Отче о лабораторной работе должен содержать:

1. Титульный лист;
2. Задание с указанием варианта;
3. Принтскрины фронтальной панели и блок диаграммы виртуального прибора;

Контрольные вопросы

1. Для чего используются логические функции?
2. Что означает выражение «полиморфная функция»?
3. В какой палитре функций находятся логические функции?
4. Приведите примеры использования логической функции И?
5. В чем особенность работы функции Исключающее ИЛИ?
6. Дайте объяснение инвертирующим свойствам некоторых логических функций.
7. Логический массив – что это?
8. В чем состоит настройка функции Elapsed Time?
9. Какую задачу решает функция Elapsed Time в лабораторной работе?
10. В чем состоит настройка функции Formula?

Литература: 2, 3.

Лабораторная работа №6

Введение и конфигурация NI-DAQmx, сбор данных

Цель работы: ознакомление с технологией конфигурирования цифровых измерительных приборов аналогового и цифрового ввода и вывода под управлением графического интерфейса LabView. В связи с этим необходимо решить следующие задачи:

1. Ознакомиться с палитрой функций DAQmx I/O для построения алгоритма управления устройством ввода и вывода аналоговых и цифровых данных.
2. Выполнить эмуляцию физического устройства ввода и вывода аналоговых и цифровых данных в Measurement & Automation Explorer.
3. Построить код для сбора аналоговых данных с заданными параметрами устройства сбора данных.
4. Произвести настройку и опробование программы сбора данных.

Знания, умения и владения, приобретаемые обучающимся в результате освоения тем в рамках освоения компетенций: способностью осуществлять поиск, хранение, обработку и анализ информации из различных источников и баз данных, представлять ее в требуемом формате с использованием информационных, компьютерных и сетевых технологий (ОПК-1); способностью применять соответствующий физико-математический аппарат, методы анализа и моделирования, теоретического и экспериментального исследования при решении профессиональных задач (ОПК-2)

Знать: о связи дисциплины с другими изучаемыми дисциплинами; о роли дисциплины в дальнейшей научной и практической деятельности; о возможностях и преимуществах моделирования в среде LabView; терминологию, основные понятия и определения; технологию графического программирования в среде LabView; типы данных и последовательность их обработки (ОПК-1); общие вопросы построения измерительных систем с

использованием: современных компьютерных технологий, принципы решения технических задач; измерений с применением современных аппаратных и программных средств (ОПК-2);

Уметь: формулировать постановку задачи, определять цель (ОПК-1); оптимально выбирать программно-аппаратные средства для решения задач и проблем измерительной техники; разработать алгоритмы измерения и обработки данных (ОПК-2);

Владеть: терминологией, основными понятиями и определениями; технологией графического программирования в среде LabView; навыками обработки различных типов данных; общими вопросами построения измерительных систем с использованием современных компьютерных технологий (ОПК-1); принципами решения технических задач измерений с применением современных аппаратных и программных средств (ОПК-2);

Теоретическая часть

Введение в сбор данных и конфигурация.

Среда LabVIEW включает в себя набор подпрограмм ВП, позволяющих конфигурировать, собирать и посылать данные на DAQ-устройства. Часто DAQ-устройства могут выполнять разнообразные функции: аналого-цифровое преобразование (A/D), цифро-аналоговое преобразование (D/A), цифровой ввод/вывод (I/O) и управление счетчиком/таймером. Каждое устройство имеет свой набор возможностей и скорость обработки данных. Кроме этого, DAQ-устройства разрабатываются с учетом аппаратной специфики платформ и операционных систем. Для получения дополнительной информации о DAQ-устройствах используйте документ National Instruments Product Catalog на web-сайте ni.com/catalog.

Компоненты DAQ-системы.

На иллюстрации продемонстрированы два варианта компоновки DAQ-системы. В варианте «А» DAQ-устройство встроено в компьютер, а в варианте «В» DAQ-устройство является внешним. С внешним устройством

можно построить DAQ-систему на базе компьютера без доступных слотов расширения, например, с использованием портативных компьютеров. Компьютер и DAQ-модуль связываются между собой через аппаратные интерфейсы, такие как параллельный порт, последовательный порт и сетевые карты (Ethernet). Практически эта система является примером удаленного управления DAQ-устройством.

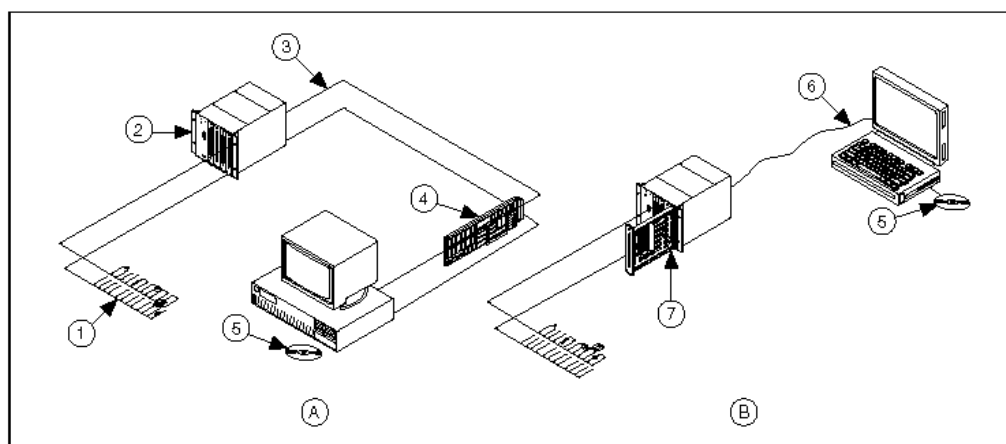


Рисунок 6.1 – Компоненты DAQ-системы: (1) Датчики, (2) Модуль согласования сигналов, (3) Согласованные сигналы, (5) Программное обеспечение, 6. Связь с параллельным портом, 7. Внешний DAQ-модуль.

Основной задачей, решаемой DAQ-системами, является задача измерения или генерации физических сигналов в реальном времени. Перед тем как компьютерная система измерит физический сигнал, датчик или усилитель должен преобразовать физический сигнал в электрический, например, ток или напряжение. Встроенное DAQ-устройство часто рассматривается как полная DAQ-система, хотя практически это только один из компонент системы. В отличие от самостоятельных устройств измерения, не всегда возможно соединение напрямую источника сигналов с встроенным DAQ-устройством. В этих случаях необходимо использовать дополнительные модули согласования сигналов перед тем как DAQ-устройство преобразует их в цифровой формат. Программные средства DAQ-систем включают в себя: сбор данных, анализ данных и представление результатов.

DAQ-устройства производства компании NI поставляются в комплекте с драйверами NI-DAQ. NI-DAQ взаимодействует и управляет измерительными устройствами National Instruments, включая такие DAQ-устройства как многофункциональные устройства ввода-вывода сигналов (MIO) серии E, SCXI модули согласования сигналов и модули переключения сигналов. NI-DAQ является расширенной библиотекой функций, которые можно вызывать из среды создания приложений, например, LabVIEW, для программирования всех возможностей измерительного устройства NI.

Программирование измерительного устройства NI возможно как в программных пакетах National Instruments: LabVIEW, LabWindows/CVI и Measurement Studio, так и в любой среде программирования, поддерживающей вызовы динамических библиотек (DDL) с использованием ANSI C интерфейса. Использование любого программного обеспечения NI существенно уменьшает время, затраченное на создание приложений сбора данных:

- LabVIEW обеспечивает сбор данных с помощью LabVIEW DAQ - комплекса виртуальных приборов для программирования измерительных устройств NI.
- LabWindows/CVI имеет полную встроенную поддержку ANSI C окружения, программирование измерительных устройств NI производится с помощью библиотеки сбора данных LabWindows/CVI Data Acquisition library.
- Инструменты программирования Measurement Studio предназначены для создания тестовых программ и приложений сбора данных в среде Microsoft Visual Studio .NET. Measurement Studio имеет поддержку Visual C#, Visual Basic .NET и Visual C++ .NET

Комплекс разработки приложений сбора данных состоит из среды программирования, MAX и NI-DAQ. MAX является высокоуровневым приложением, которое используется для тестирования и настройки DAQ-устройств. NI-DAQ состоит из следующих программных интерфейсов:

- стандартный NI-DAQ;

- NI-DAQmx;
- NI-SWITCH.

NI-DAQmx является последним поколением драйверов NI-DAQ. Он обладает новыми функциями и инструментами управления измерительными устройствами. NI-DAQmx имеет много новых особенностей и преимуществ по сравнению с предыдущей версией NI-DAQ:

- DAQ Configuration Assistant - помощник настройки DAQ-устройств с помощью нового графического интерфейса позволяет конфигурировать настройки, каналы и задания измерения DAQ-устройств в LabVIEW, LabWindows/CVI и Measurement Studio.
- Увеличилось быстродействие ряда операций, в частности однократного аналогового ввода-вывода, более эффективно организована многозадачность.
- Программный интерфейс для создания DAQ-приложений стал более простым и интуитивно понятным.
- Появились дополнительные возможности программного интерфейса NI-DAQmx для LabVIEW, включая узлы Атрибутов для сбора данных и улучшенную поддержку типов данных waveform для операций аналогового и цифрового ввода-вывода.
- Разработаны единые программные интерфейсы и функциональность для ANSI C, LabWindows/CVI и Measurement Studio, включая интерфейсы .NET и C++.

Настройка аппаратных средств DAQ-устройств.

Перед использованием ВП Сбора Данных необходимо настроить DAQ-устройство, установленное в компьютер.

Windows Configuration Manager хранит информацию обо всем установленном на компьютер аппаратном обеспечении, включая устройства National Instruments DAQ. Если у Вас есть Plug & Play (PnP) устройства,

например, карта МІО серии Е, Windows Configuration Manager автоматически его определит и настроит. Если устройство не поддерживает PnP, необходимо его настроить вручную с помощью раздела Add New Hardware (Установка и удаление устройств) в Control Panel (Панели Управления).

Аппаратную конфигурацию Windows можно проверить с помощью Device Manager (Менеджера Устройств), который находится в Start»Settings»Control Panel»System»Device Manager (Пуск»Настройки»Панель Управления»Система»Диспетчер Устройств). В разделе Data Acquisition Devices отображаются все DAQ-устройства, установленные на компьютер. Дважды щелкните мышью по устройству для отображения диалогового окна с закладками. На закладке General (Общие) приводится общая информация об устройстве. На закладке Resources (Ресурсы) перечислены системные ресурсы, используемые устройством: номера прерываний (IRQ), каналы прямого доступа к памяти (DMA) и базовые адреса ввода-вывода для программно конфигурируемых устройств. На закладке NI-DAQ Information указан тип шины данного DAQ-устройства. Производитель и номер версии драйвера DAQ-устройства приведены на закладке Driver (Драйвер).

Среда LabVIEW устанавливает утилиту конфигурации Measurement & Automation Explorer для детальной настройки параметров конфигурации каналов устройств. Эту утилиту необходимо запускать после установки DAQ-устройства на компьютер. Утилита конфигурации считывает информацию из реестра Windows, записанную Диспетчером устройств (Device Manager), и присваивает логическое имя для каждого DAQ-устройства. По логическому имени среда LabVIEW распознает DAQ-устройство. Запуск конфигурационной утилиты происходит двойным щелчком левой кнопки мыши по ее иконке на рабочем столе операционной системы или выбором пункта главного меню Tools»Measurement & Automation Explorer непосредственно в среде LabVIEW. Начальное окно

конфигурационной утилиты показано ниже на рисунке. Measurement & Automation Explorer также используется для конфигурации устройств стандарта SCXI.

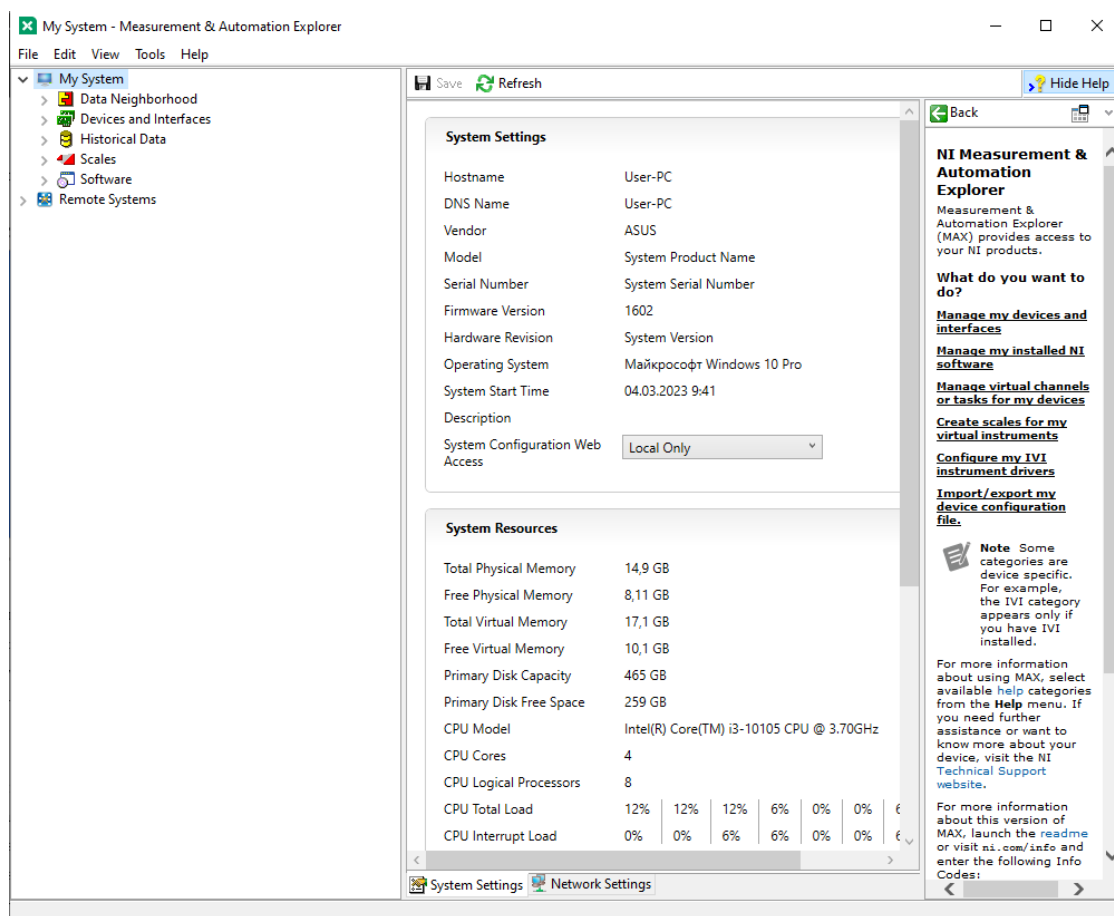


Рисунок 6.2 – Measurement & Automation Explorer.

Конфигурационная утилита определяет все аппаратные средства фирмы National Instruments. Параметры устройства можно также установить с помощью утилит-конфигураций, входящих в комплект поставки устройств. Measurement & Automation Explorer позволяет сохранить логическое имя устройства и параметры конфигураций в реестр Windows.

Настройка каналов и заданий.

В Стандартном NI-DAQ можно создавать виртуальные каналы, которые включают в себя совокупность настроек физического канала DAQ, типа измерений и информацию о нормировке значений. В Стандартном NI-DAQ и NI-DAQ более ранних версий понятие виртуальных каналов

используется в качестве соответствия физических каналов проводимым измерениям. Понятие NI-DAQmx channels аналогично виртуальным каналам Стандартного NI-DAQ.

NI-DAQmx также включает в себя задания, являющихся частью программного интерфейса API. Задание – это совокупность настроек одного или нескольких каналов, синхронизации, временных и других параметров. Задание описывает измерение или генерацию сигналов, которые необходимо выполнить. Каналы, созданные в рамках задания, являются локальными. Каналы, определенные вне заданий, являются глобальными, и могут быть использованы отдельно. Настройка виртуальных каналов является необязательной в Стандартном NI-DAQ и более ранних версиях, но в NI-DAQmx выполнение этой процедуры необходимо для проведения измерений. В Стандартном NI-DAQ конфигурирование виртуальных каналов производилось в MAX. В NI-DAQmx настройка виртуальных каналов возможна как в MAX, так и в самом приложении, причем как в рамках задания, так и отдельно.

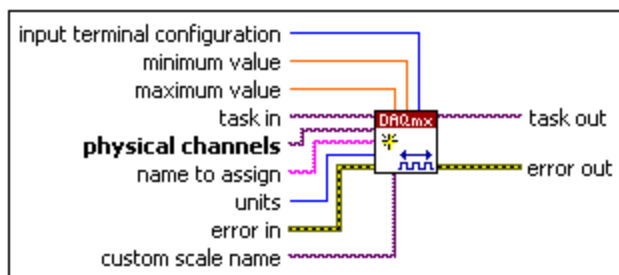
Задание

Необходимо создать в Measurement & Automation Explorer (рисунок 6.3) создать симулируемый прибор NI 9215 (или его аналог) (рисунок 6.4) для измерения сигнала напряжения в одноканальном режиме с частотой 10000 измерений в секунду. Далее необходимо создать виртуальный прибор управляющий конфигурированием, запуском и отображением данных измеренных при помощи симулируемого прибора NI 9215 (или его аналога).

Выполнение работы

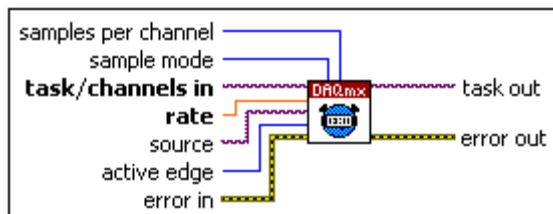
На рисунке 6.5 отображена лицевая панель виртуального прибора, несущая в себе элементы управления настройкой сигналов, настройкой таймингов и отображение полученных данных в графическом виде.

Блок диаграмма виртуального прибора (Рисунок 6.6) несет функциональную часть прибора и содержит:



DAQmx Create Virtual Channel

Элемент создания и настройки виртуального канала;



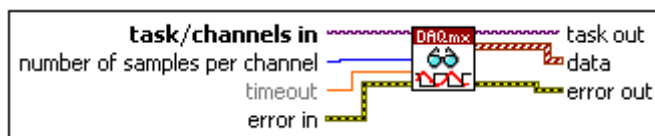
DAQmx Timing

Элемент настройки канала по частоте измерения и созданию буфера данных в случае необходимости;



DAQmx Start Task

Элемент переводит задачу в рабочее состояние, чтобы начать измерения или генерации;



DAQmx Read

Элемент считывает заданное количество образцов из задачи или виртуального канала, которые вы укажете;



DAQmx Clear Task

Этот элемент останавливает задачу, если необходимо, и освобождает все ресурсы. Вы не можете использовать задачу после того, как вы ее остановили.

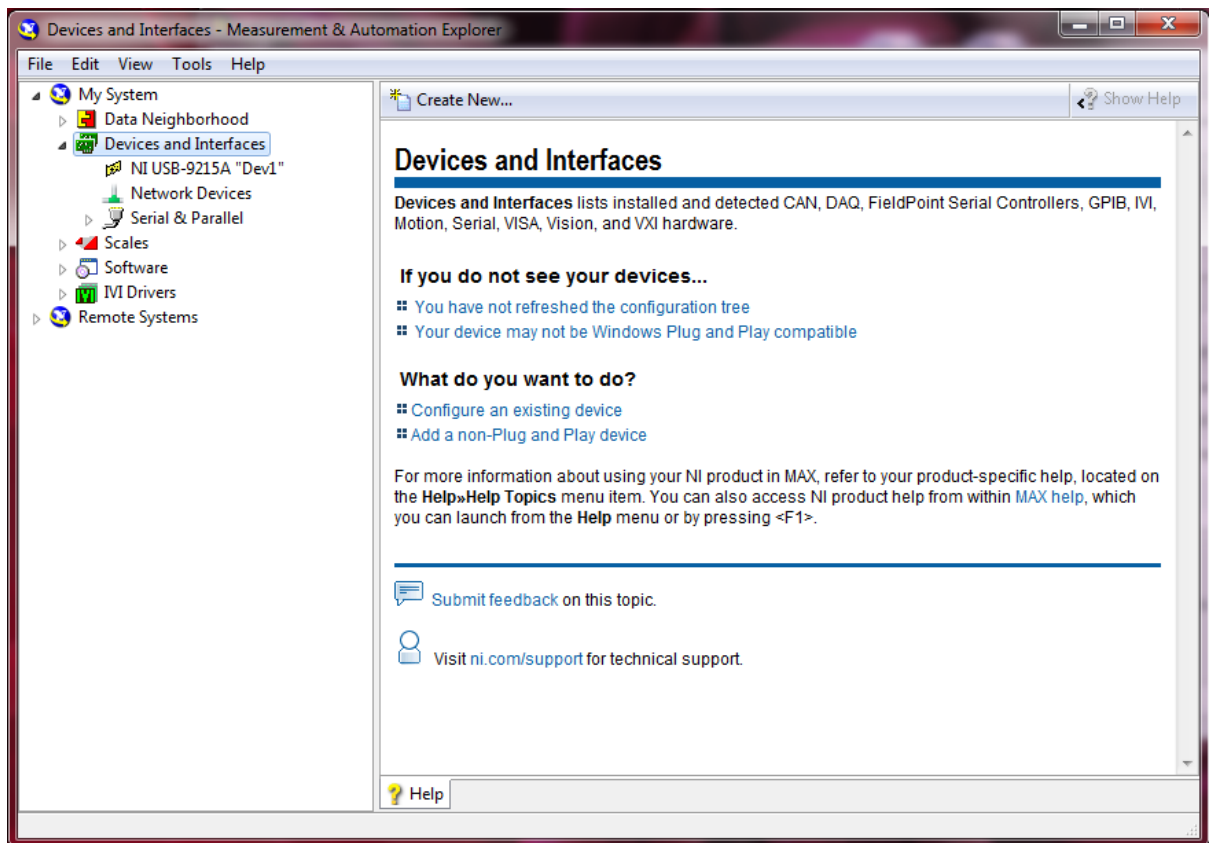


Рисунок 6.3 – панель программы Measurement & Automation Explorer

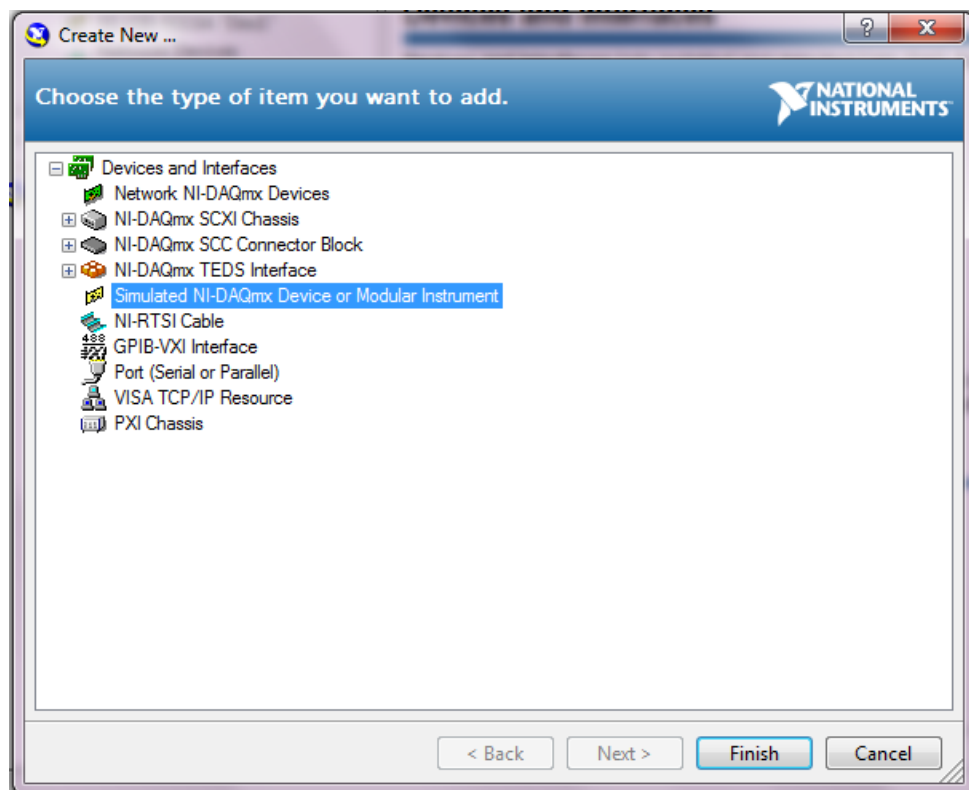


Рисунок 6.4 – создание симулируемого прибора в Measurement & Automation Explorer

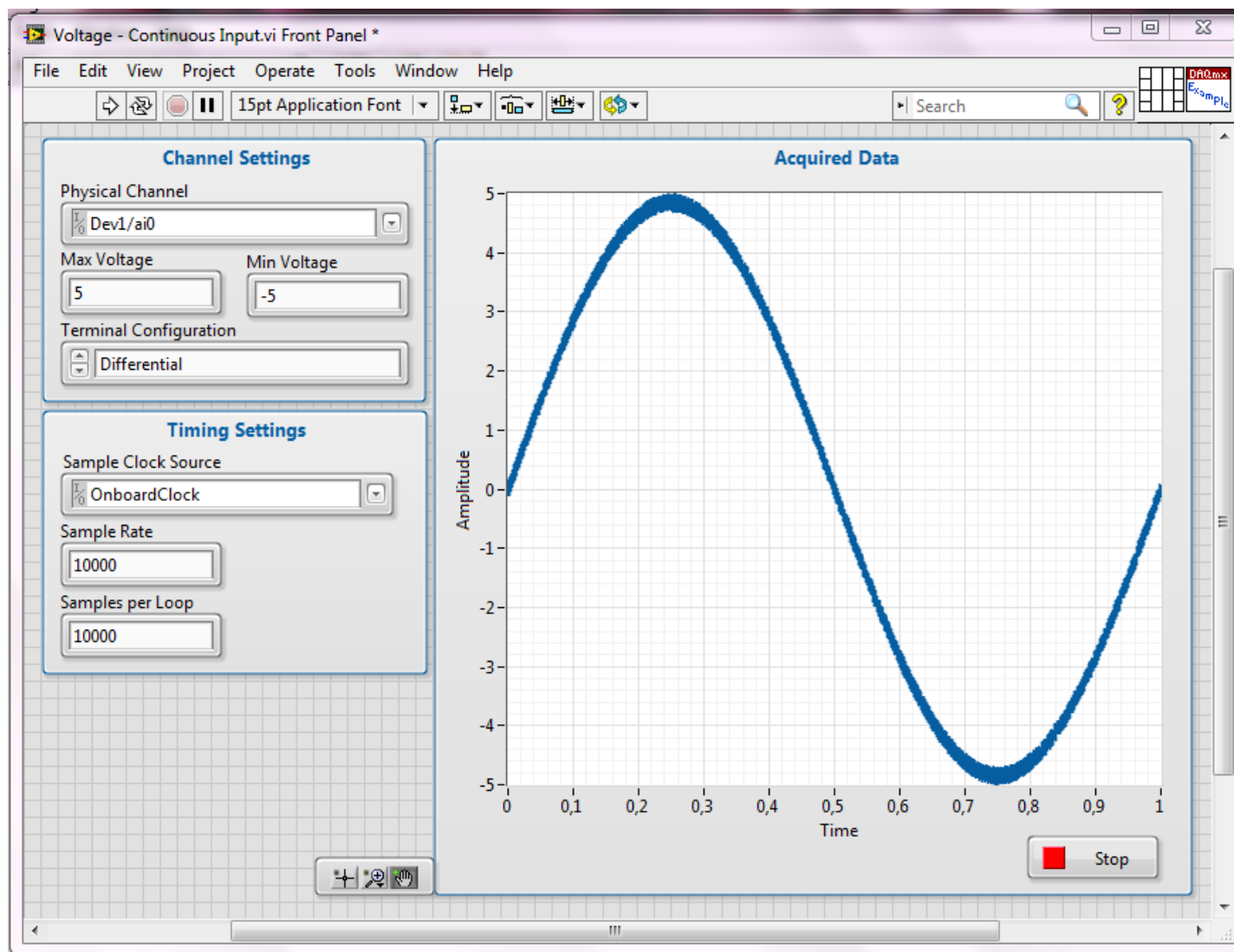


Рисунок 6.5 – Фронтальная панель виртуального прибора

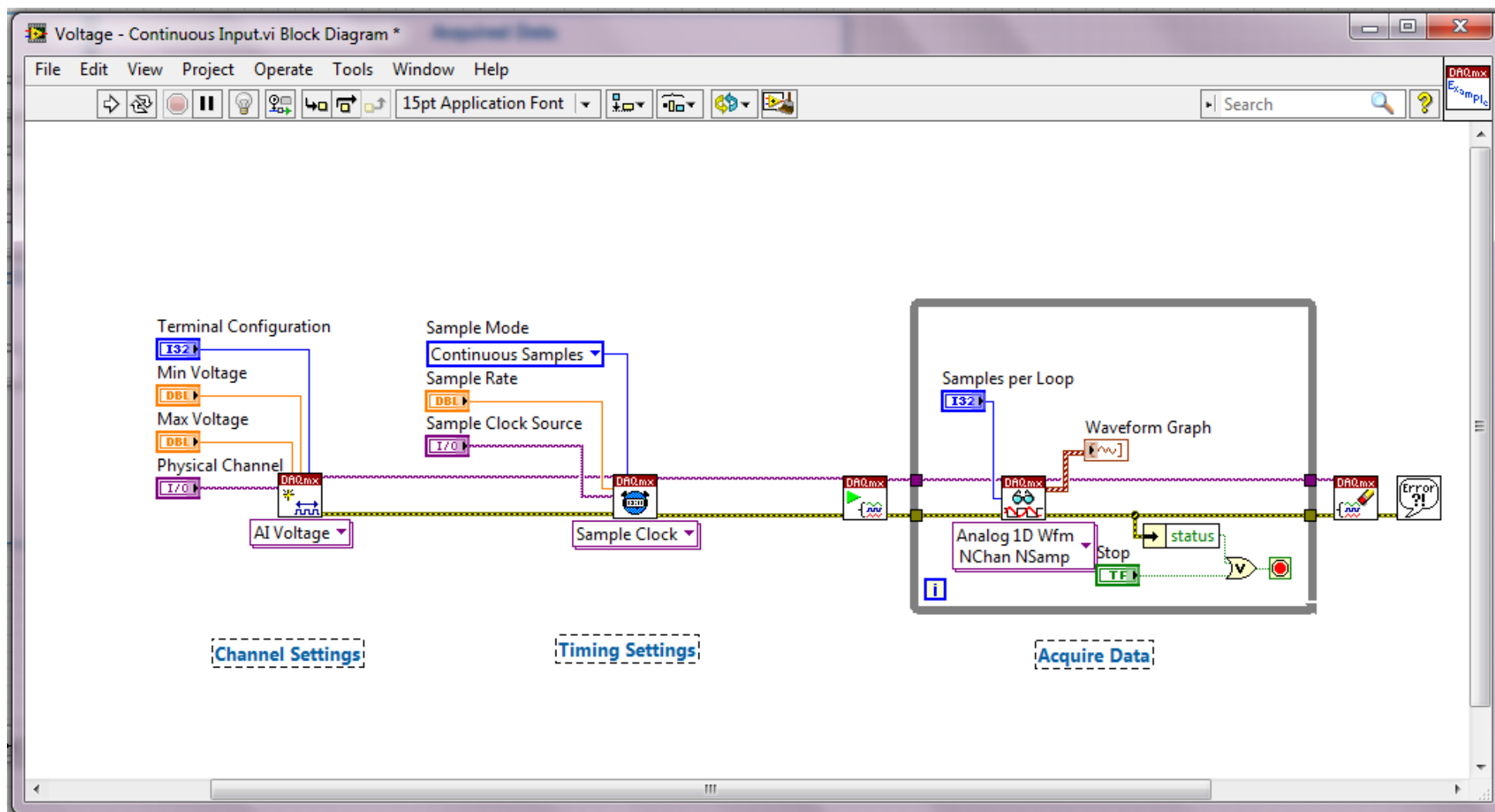


Рисунок 6.6 – Блок диаграмма виртуального прибора.

Результат выполнения лабораторной работы.

Содержание отчета

Отчет оформляется каждым студентом самостоятельно.

Демонстрация работы программы на ЭВМ.

Содержание отчета:

1. Титульный лист.
2. Задание.
4. Лицевая панель и структурная схема прибора, разработанного по пункту задания.
5. Описание всех элементов лицевой панели и структурной схемы прибора.
6. Лицевая панель прибора и структурная схема прибора, разработанного по пункту задания; объяснения принципов его функционирования; распечатка файла с результатами работы программы

Контрольные вопросы

1. Что относится к компонентам DAQ-системы?
2. Какие параметры позволяет задать элемент создания и настройки виртуального канала?
3. Каким элементом задается настройка канала по частоте измерения?
4. Какие задаются параметры настройки при создании буфера данных в случае необходимости?
5. Какие функции выполняются элементом DAQmx Clear Task?
6. Какие есть варианты задания элементов считывания образцов из задачи или виртуального каналы?
7. Опишите последовательность действий для выполнения эмуляции физического устройства ввода в Measurement & Automation Explorer?
8. Зачем нужен кластер ошибок в программах сбора данных?

Литература: 2, 3.

Лабораторная работа №7

Настройка аппаратных средств DAQ-устройств

Цель работы: освоение технологии настройки аппаратных измерительных DAQ-устройств ввода под управлением графического интерфейса LabView. В связи с этим необходимо решить следующие задачи:

1. Выполнить построение кода выполняющего сбор данных двух аналоговых каналов заданного DAQ-устройства.
2. Произвести программную обработку получаемых в реальном времени измеренных данных согласно заданию и отобразить результат при помощи графического интерфейса.

Знания, умения и владения, приобретаемые обучающимся в результате освоения тем в рамках освоения компетенций: способностью осуществлять поиск, хранение, обработку и анализ информации из различных источников и баз данных, представлять ее в требуемом формате с использованием информационных, компьютерных и сетевых технологий (ОПК-1); способностью применять соответствующий физико-математический аппарат, методы анализа и моделирования, теоретического и экспериментального исследования при решении профессиональных задач (ОПК-2)

Знать: о связи дисциплины с другими изучаемыми дисциплинами; о роли дисциплины в дальнейшей научной и практической деятельности; о возможностях и преимуществах моделирования в среде LabView; терминологию, основные понятия и определения; технологию графического программирования в среде LabView; типы данных и последовательность их обработки (ОПК-1); общие вопросы построения измерительных систем с использованием: современных компьютерных технологий, принципы решения технических задач; измерений с применением современных аппаратных и программных средств (ОПК-2);

Уметь: формулировать постановку задачи, определять цель (ОПК-1); оптимально выбирать программно-аппаратные средства для решения задач и проблем измерительной техники; разработать алгоритмы измерения и обработки данных (ОПК-2);

Владеть: терминологией, основными понятиями и определениями; технологией графического программирования в среде LabView; навыками обработки различных типов данных; общими вопросами построения измерительных систем с использованием современных компьютерных технологий (ОПК-1); принципами решения технических задач измерений с применением современных аппаратных и программных средств (ОПК-2);

Теоретическая часть

MAX (только для Windows). Проверка настроек DAQ-устройства.

1) Двойным щелчком левой кнопки мыши по иконке на рабочем столе или выбором пункта главного меню Tools»Measurement & Automation Explorer в среде LabVIEW запустите конфигурационную утилиту. Утилита производит поиск в системе установленных аппаратных средств фирмы National Instruments и отображает найденную информацию.

2) Откройте секцию Devices and Interfaces (Устройства и Интерфейсы) для просмотра установленных устройств фирмы National Instruments. В примере, показанном на рисунке, обнаружены устройства PCI-6024E и PCI-GPIB.

Помимо аппаратных средств, конфигурационная утилита отображает программные средства, установленные в системе. Номер устройства отображен после его названия. ВП Сбора Данных использует номер устройства для определения, какое устройство выполняет операцию сбора данных. MAX также отображает параметры устройства, например, используемые системные ресурсы.

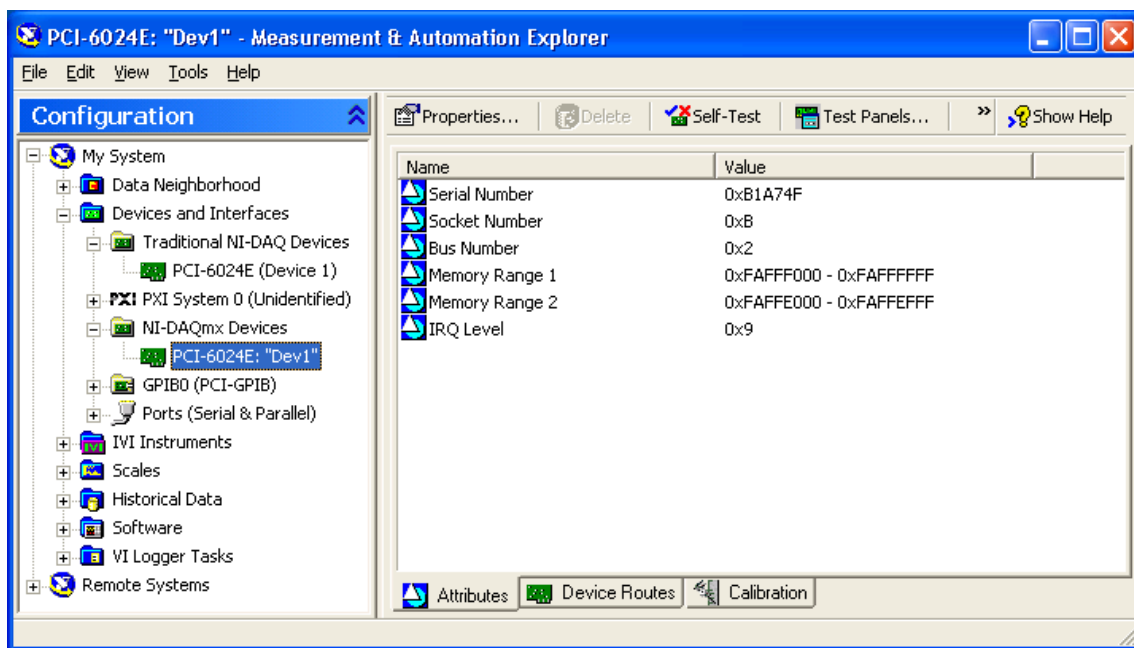


Рисунок 7.1 – Devices and Interfaces (Устройства и Интерфейсы)

Примечание. Возможно у вас установлены другие DAQ-устройства, и некоторые настройки могут различаться. Нажмите на кнопку Show/Hide в верхнем правом углу окна конфигурационной утилиты для скрытия контекстной подсказки и показа информации о DAQ-устройствах

3) На закладке Device Routes (Разводка сигналов устройства) приведена подробная информация о внутренних сигналах устройства, которые могут быть различным образом перенаправлены. Эта панель является мощным инструментом, позволяющим осуществлять синхронизацию внутренних компонент и внешних устройств.

4) На закладке Calibration (Калибровка) указаны даты проведения внешней и внутренней калибровки устройства.

5) Щелкните правой кнопкой мыши по устройству в разделе NI-DAQmx и выберите пункт Self Calibrate (Внутренняя калибровка) для осуществления калибровки DAQ-устройства с помощью высокоточного источника напряжения и обновления калибровочных констант. По окончании калибровки информация о ней появится на закладке Calibration в разделе Self Calibration.

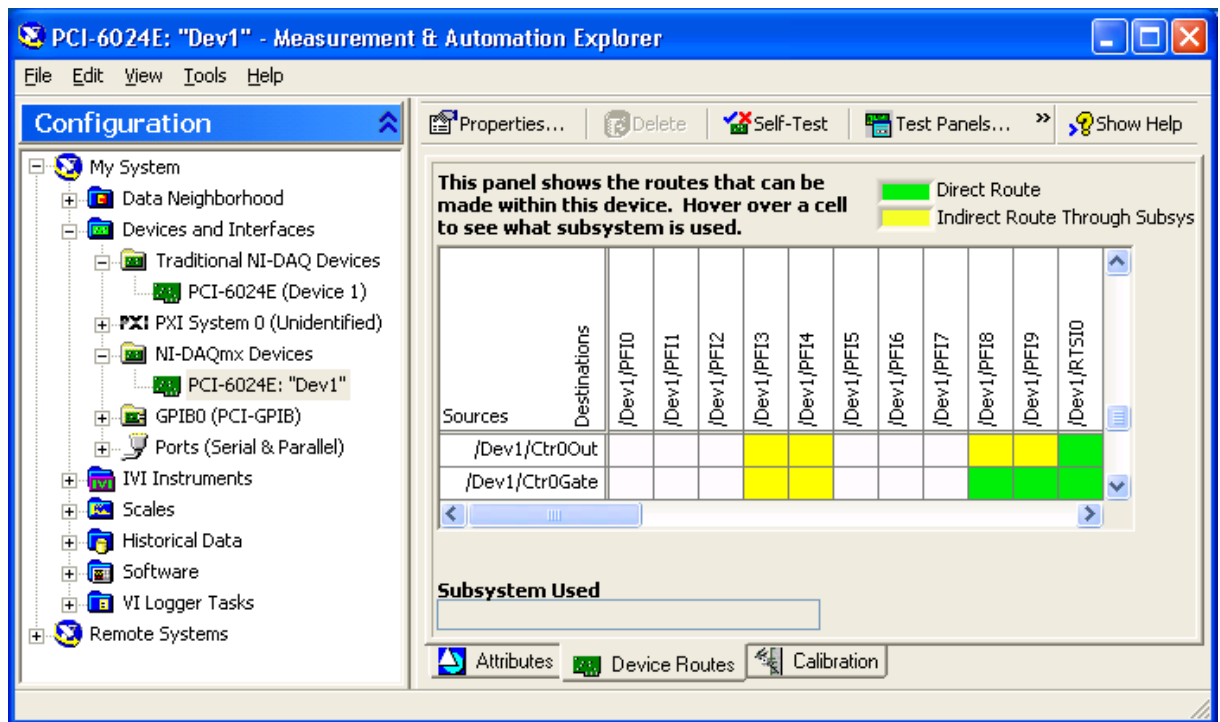


Рисунок 7.2 – Устройства в разделе NI-DAQmx

Тестирование компонент DAQ-устройства

6) Нажмите на кнопку Self-Test для проведения тестирования устройства. Проводится тестирование системных ресурсов, используемых устройством. Устройство должно успешно пройти все тесты, поскольку оно уже настроено.

7) Нажмите на кнопку Test Panel (Тестовая панель) для тестирования отдельных функций DAQ-устройства, таких как аналоговый вход и выход. На экране появится диалоговое окно Test Panel (Тестовая панель).

а) Используйте закладку Analog Input (Аналоговый Вход) для тестирования различных каналов аналогового входа DAQ-устройства. Канал Dev1/ai0 подключен к датчику температуры сигнальной панели. Нажмите на кнопку Start для начала сбора данных. Поместите палец на датчик температуры, в результате напряжение возрастает. Измените положение переключателя Noise (Шум) на On (Включено) и проследите за изменением сигнала на закладке. По завершению нажмите на кнопку Stop.

b) Перейдите на закладку Analog Output (Аналоговый выход) для установки значения постоянного напряжения или гармонического сигнала на одном из аналоговых выходов DAQ-устройства. Установите Output Mode (Режим генерации) в положение Sine Generator (Синусоидальный сигнал) и нажмите на кнопку Start Sine Generator (Генерация гармонического сигнала). LabVIEW генерирует непрерывный синусоидальный сигнал на аналоговом выходе канала «0».

c) На внешней соединительной сигнальной панели DAQ-устройства подключите Аналоговый выход канала «0» к Аналоговому входу канала «1».

d) Откройте закладку Analog Input (Аналоговый вход) и выберите канал Dev1/ai1. Нажмите на кнопку Start для начала сбора данных со входного канала «1». На графике отобразится синусоидальный сигнал, который генерируется на выходе канала «0».

e) Перейдите на закладку Digital I/O (Цифровой ввод/вывод) для проверки работы цифровых линий DAQ-устройства.

f) Настройте цифровые каналы с «0» по «3» в качестве каналов вывода и отметьте или уберите выделение с пунктов Logic Level (Логический уровень). При этом погаснут или загорятся соответствующие светодиоды на сигнальной панели. Светодиоды работают по принципу отрицательной логики: выключенный светодиод соответствует положительному уровню цифрового канала.

g) Нажмите на кнопку Close для закрытия тестовой панели и возврата в главное окно MAX.

8) Перейдите на закладку Counter I/O (Счетчики ввода-вывода) для проверки работы счетчиков и таймеров DAQ-устройства. Для проверки операций счетчик/таймер перевести режим Counter Mode (Режим Счетчика) в положение Simple Event Counting (Простой подсчет событий) и нажать на кнопку Start. Значение счетчика начнет быстро возрастать. Нажмите на кнопку Stop для остановки теста.

9) Закройте MAX, выбрав в меню пункт File»Exit.

Сбор данных в LabVIEW

Подпрограммы сбора данных размещены в палитрах Functions»NI Measurements»Data Acquisition и Functions»NI Measurements»DAQmx - Data Acquisition. В разделе Data Acquisition собраны стандартные виртуальные приборы сбора данных, а в DAQmx - Data Acquisition - ВП для работы с NI-DAQmx.

В палитре DAQmx - Data Acquisition содержатся все необходимые подпрограммы для осуществления операций аналогового и цифрового ввода-вывода и работы со счетчиками/таймерами. Виртуальные приборы собраны таким образом, что большинство задач могут быть решены с их использованием. Можно настроить ВП для выполнения специфического действия с помощью узла Атрибутов. Многие задания, не требующие расширенных возможностей синхронизации, могут быть выполнены с помощью экспресс-ВП DAQmx Assistant. В этом курсе описано, как использовать экспресс-ВП DAQmx Assistant для выполнения операций сбора данных. Для получения дополнительной информации об использовании NI-DAQmx обратитесь к справочному пособию NI-DAQmx или изучите расширенный курс LabVIEW Data Acquisition and Signal Conditioning.

Экспресс-ВП DAQmx Assistant позволяют просто осуществить настройку DAQ-устройства. При добавлении экспресс-ВП DAQmx Assistant на блок-диаграмму появляется диалоговое окно, в котором осуществляется конфигурация задания - провести определенные измерения. В процессе создания локального задания указывается необходимый тип измерения.

Один раз создав задание, информация о нем сохраняется в экспресс-ВП DAQmx Assistant. Впоследствии можно сконфигурировать экспресс-ВП DAQmx Assistant заново, дважды щелкнув по нему мышью.

Используйте операции аналогового ввода для осуществления аналого-цифрового преобразования.

Существует несколько типов измерений входного сигнала: напряжение, температура, деформация, ток, сопротивление или частота.

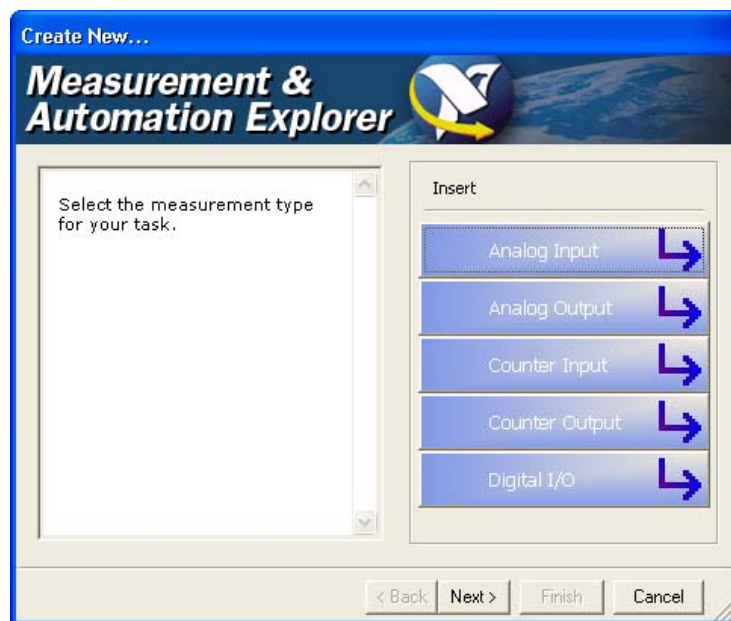


Рисунок 7.3 – Создание задания

Каждый тип измерений имеет собственные параметры, например, величина сопротивления для измерения тока или калибровка датчика деформаций для их измерений.

Установление временного такта выполнения заданий.

При выполнении операций аналогового ввода данных задание может формулироваться по-разному: получение 1 значения, получение n значений или непрерывный сбор данных.

Получение одного значения

Получение одного значения является операцией по вызову. Другими словами NI-DAQmx оцифровывает одно значение с входного канала и немедленно возвращает его величину. Выполнение этой операции не требует наличия буфера и аппаратного контроля временного такта. Например, для периодического контроля уровня жидкости в резервуаре необходимо получать по одному значению. Вы можете подключить датчик, генерирующий разное напряжение в зависимости от уровня жидкости, к одному из каналов DAQ-устройства и производить контроль, периодически оцифровывая по одному значению.

Получение n значений

Один из методов получения n значений - n раз получить по одному значению. Однако получение по одному значению из одного или нескольких каналов снова и снова является неэффективным и занимает много времени. Более того, отсутствует контроль времени между последовательными операциями получения значений. Вместо этого необходимо использовать аппаратное задание временного такта выполнения операций, в процессе которого используется буферизация полученных данных в компьютерной памяти, что приводит к более эффективному процессу сбора данных. С программной точки зрения необходимо включить режим аппаратного задания временного такта выполнения операций и задать частоту оцифровки `sample rate` и ограниченный по времени режим работы `sample mode (finite)`. Возможна оцифровка нескольких значений из одного канала или из нескольких.

С помощью NI-DAQmx можно осуществлять сбор данных с нескольких каналов. Например, необходимо контролировать уровень жидкости в резервуаре и ее температуру. В этом случае нужно иметь два датчика, подключенных к разным каналам DAQ-устройства.

Непрерывный сбор данных

Если необходимо отображать, обрабатывать и производить запись данных по мере их поступления, лучше использовать режим непрерывного сбора данных. Для этого устанавливается режим работы `sample mode (continuous)`.

Синхронизация заданий

Когда устройство, управляемое NI-DAQmx, работает, оно производит операции. Два наиболее часто встречающихся действия - обработка значения и начало сбора данных. Каждое производимое действие NI-DAQmx вызвано чем-либо, либо имеет причину. Такие причины называются синхронным запуском.

Start Trigger

Сигнал, запускающий сбор данных.

Reference Trigger

Сигнал, устанавливающий реперную точку в наборе входных значений. Данные, полученные до этой точки называются pretrigger data. Данные после реперной точки являются posttrigger data.

Краткое изложение теоретического материала

- MAX является основным конфигурационным и тестовым приложением для DAQ-устройств.
- Экспресс-ВП DAQmx Assistant используется для настройки DAQ-устройств и проведения измерений.
- Большинство программ могут использовать экспресс-ВП DAQmx Assistant. В приложениях, использующих расширенные возможности установки временного такта и синхронизации, используются Виртуальные приборы, поставляемые вместе с экспресс-ВП DAQmx Assistant.
- DAQmx Assistant может осуществлять операции аналогового и цифрового ввода-вывода и операции со счетчиками.

Задание

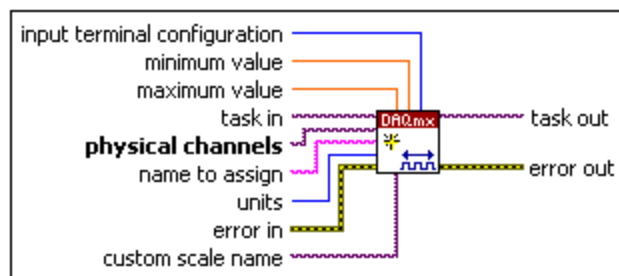
Необходимо создать в Measurement & Automation Explorer (рисунок 7.4) создать симулируемый прибор NI 9215 (или его аналог) (рисунок 7.5) для измерения сигнала напряжения в одноканальном режиме с частотой 10000 измерений в секунду. Далее необходимо создать виртуальный прибор управляющий конфигурированием, запуском и отображением данных измеренных при помощи симулируемого прибора NI 9215 (или его аналога)

Прибор должен вычислять значения активной, полной и реактивной мощностей по измеряемым значениям тока и напряжения, а также отображать осциллограммы этих сигналов на одном графике в масштабе.

Выполнение работы

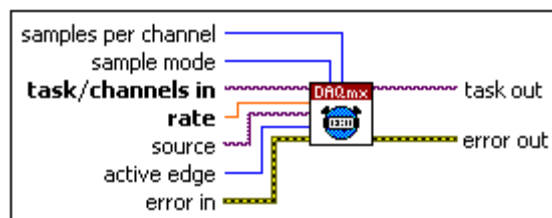
На рисунке 7.6 отображена лицевая панель виртуального прибора, несущая в себе элементы управления настройкой сигналов, настройкой таймингов и отображение полученных данных в графическом виде.

Блок диаграмма виртуального прибора (Рисунок 7.7) несет функциональную часть прибора и содержит:



DAQmx Create Virtual Channel

Элемент создания и настройки виртуального канала;



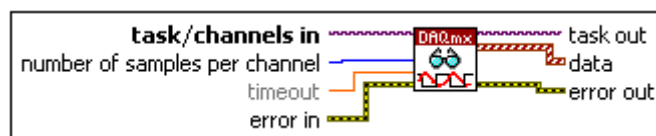
DAQmx Timing

Элемент настройки канала по частоте измерения и созданию буфера данных в случае необходимости;



DAQmx Start Task

Элемент переводит задачу в рабочее состояние, чтобы начать измерения или генерации;



DAQmx Read

Элемент считывает заданное количество образцов из задачи или виртуального каналы, которые вы укажете;



DAQmx Clear Task

Этот элемент останавливает задачу, если необходимо, и освобождает все ресурсы. Вы не можете использовать задачу после того, как вы ее остановили.

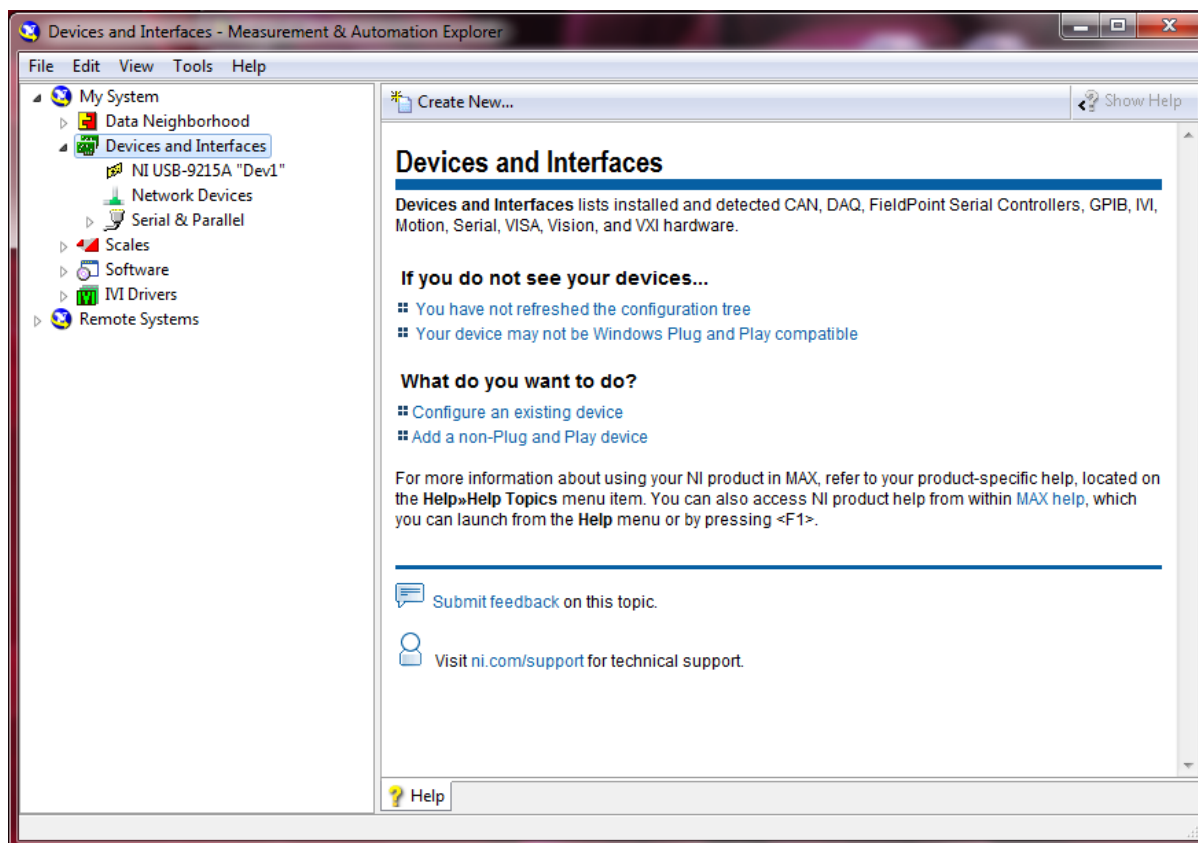


Рисунок 7.4 – панель программы Measurement & Automation Explorer

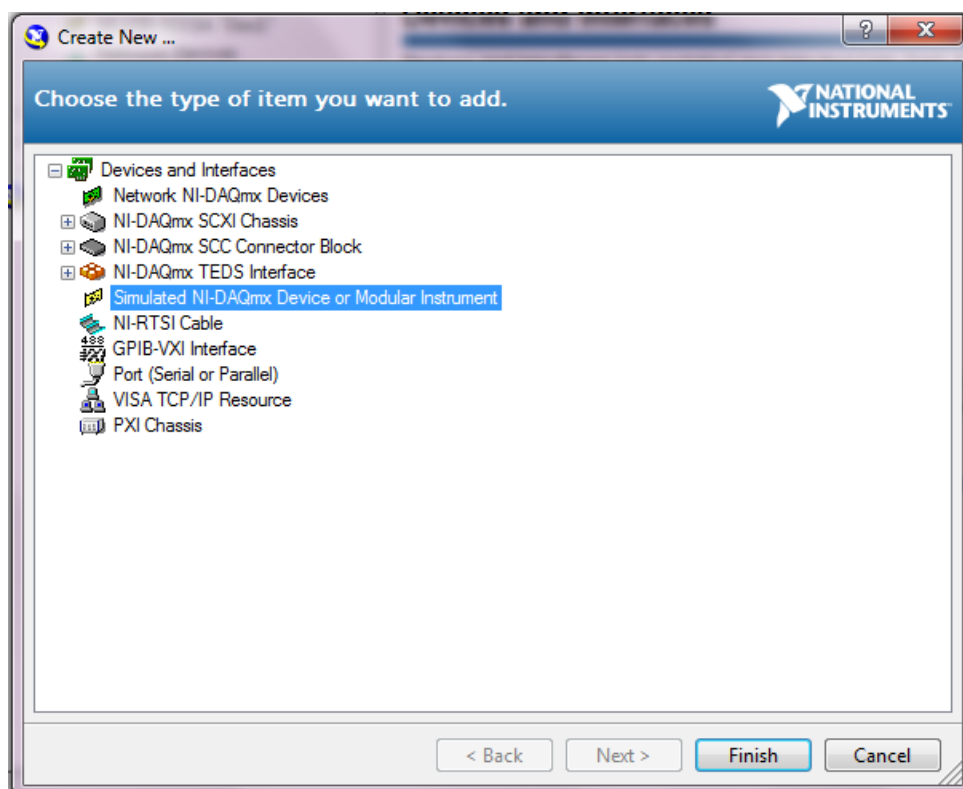


Рисунок 7.5 – создание симулируемого прибора в Measurement & Automation Explorer

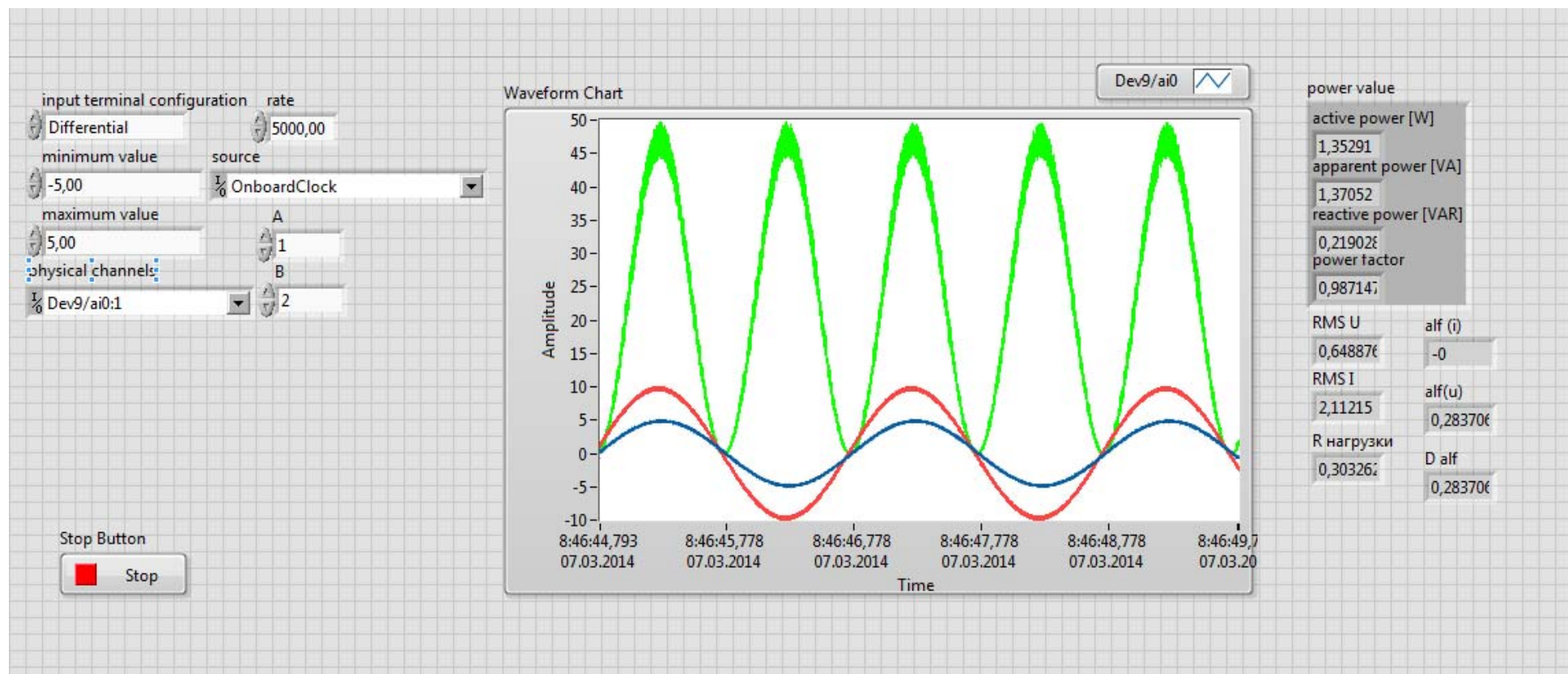


Рисунок 7.6 – Фронтальная панель виртуального прибора

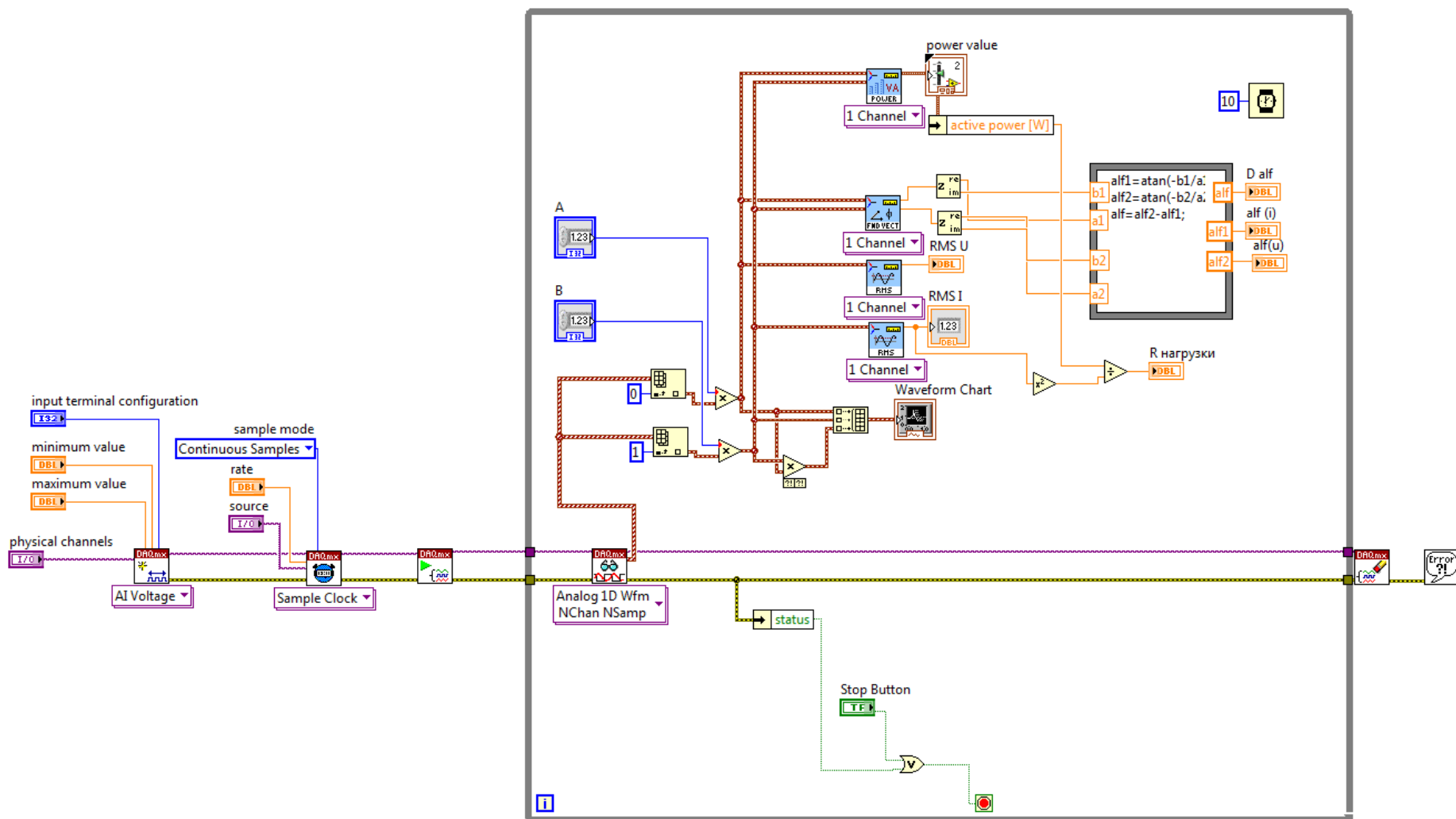


Рисунок 7.7 – Блок диаграмма виртуального прибора

Результат выполнения лабораторной работы.

Содержание отчета

Отчет оформляется каждым студентом самостоятельно.

Демонстрация работы программы на ЭВМ.

Содержание отчета:

1. Титульный лист.
2. Задание.
4. Лицевая панель и структурная схема прибора, разработанного по пункту задания.
5. Описание всех элементов лицевой панели и структурной схемы прибора.
6. Лицевая панель прибора и структурная схема прибора, разработанного по пункту задания; объяснения принципов его функционирования; распечатка файла с результатами работы программы

Контрольные вопросы

1. Какое приложение является тестовым приложением для DAQ-устройств?
2. Как сконфигурировать DAQ-устройства в MAX?
3. Поясните функции DAQmx Assistant
4. Что такое DAQ-устройство?
5. Что такое MAX?
6. Как используют использовать экспресс-ВП DAQmx Assistant?
7. Как используют расширенные возможности установки временного такта и синхронизации?
8. Что означает экспресс-ВП DAQmx Assistant?

Литература: 2, 3, 4.

Лабораторная работа №8

Калибровка генерации сигнала средствами DAQ-устройств

Цель работы: освоение технологии создания эталонных сигналов, настройки аппаратных измерительных DAQ-устройств вывода под управлением графического интерфейса LabView. В связи с этим необходимо решить следующие задачи:

1. Выполнить построение кода, выполняющего вывод данных на аналоговый канал заданного DAQ-устройства.
2. Произвести генерацию эталонного сигнала заданной формы и параметров и вывод в реальном времени полученных данных на канал выхода и отобразить результат при помощи графического интерфейса.

Знания, умения и владения, приобретаемые обучающимся в результате освоения тем в рамках освоения компетенций: способностью осуществлять поиск, хранение, обработку и анализ информации из различных источников и баз данных, представлять ее в требуемом формате с использованием информационных, компьютерных и сетевых технологий (ОПК-1); способностью применять соответствующий физико-математический аппарат, методы анализа и моделирования, теоретического и экспериментального исследования при решении профессиональных задач (ОПК-2)

Знать: о связи дисциплины с другими изучаемыми дисциплинами; о роли дисциплины в дальнейшей научной и практической деятельности; о возможностях и преимуществах моделирования в среде LabView; терминологию, основные понятия и определения; технологию графического программирования в среде LabView; типы данных и последовательность их обработки (ОПК-1); общие вопросы построения измерительных систем с использованием: современных компьютерных технологий, принципы решения технических задач; измерений с применением современных аппаратных и программных средств (ОПК-2);

Уметь: формулировать постановку задачи, определять цель (ОПК-1); оптимально выбирать программно-аппаратные средства для решения задач и проблем измерительной техники; разработать алгоритмы измерения и обработки данных (ОПК-2);

Владеть: терминологией, основными понятиями и определениями; технологией графического программирования в среде LabView; навыками обработки различных типов данных; общими вопросами построения измерительных систем с использованием современных компьютерных технологий (ОПК-1); принципами решения технических задач измерений с применением современных аппаратных и программных средств (ОПК-2);

Теоретическая часть

Выполнение операций аналогового вывода.

Аналоговый выход используется для произведения цифро-аналогового преобразования. Величина на выходе может быть напряжением или током.

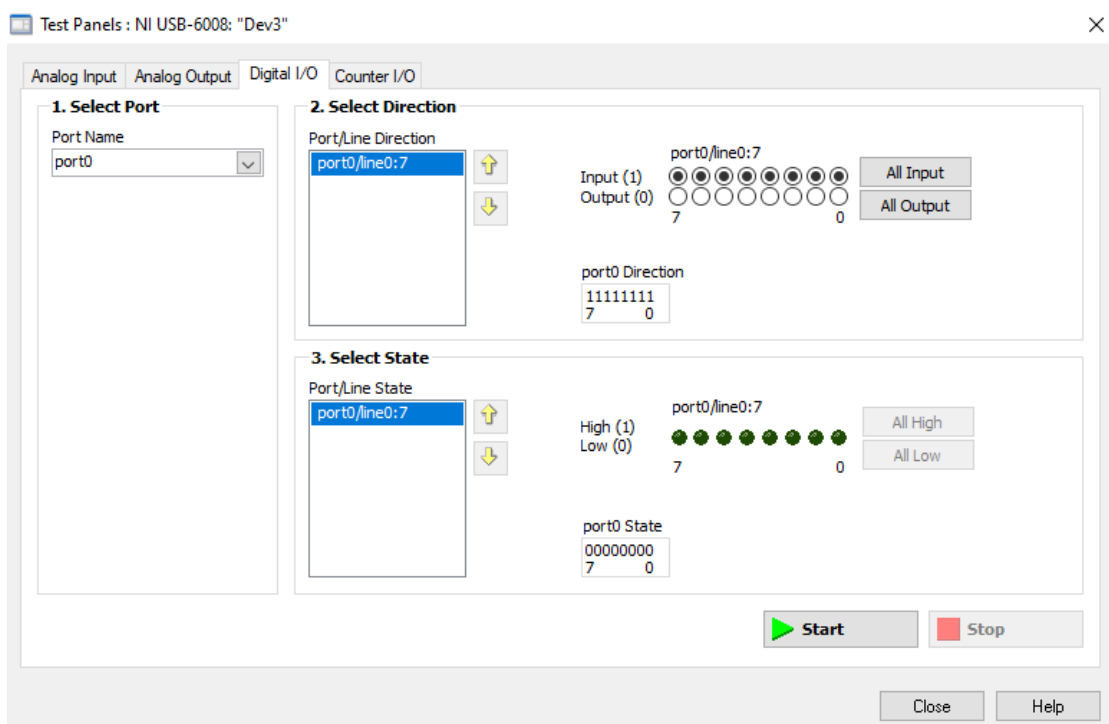


Рисунок 8.1 – Создание выходного канала.

Для осуществления вывода напряжения или тока должно быть установлено соответствующее DAQ-устройство, позволяющее генерировать сигналы заданной формы.

Генерация данных

При выполнении операций аналогового вывода данных задание может формулироваться по-разному: генерация 1 значения, генерация n значений или непрерывная генерация данных.

Генерация одного значения

Генерация одного значения используется в том случае, когда уровень сигнала более важен, чем скорость его обновления. Например, используйте генерацию одного значения, если необходимо сгенерировать постоянный сигнал. В этом случае возможно использование программных средств управления временными задержками.

Выполнение этой операции не требует наличия буфера и аппаратного контроля временного такта. Например, необходимо сгенерировать известное напряжение для эмуляции работы устройства, в этом случае можно использовать метод генерации одного значения.

Генерация n значений

Один из методов генерации n значений - n раз сгенерировать по одному значению. Однако генерация по одному значению из одного или нескольких каналов снова и снова является неэффективной и занимает много времени.

Более того, отсутствует контроль времени между последовательно сгенерированными значениями. Вместо этого необходимо использовать аппаратное задание временного такта выполнения операций, в процессе которого используется буферизация генерируемых данных в компьютерной памяти, что приводит к более эффективному процессу генерации.

Возможно использование программного и аппаратного задания временного такта выполнения операций. В случае программного управления моментом генерации значений, временные задержки определяются программой и операционной системой, а не измерительным устройством. В

случае аппаратного управления временным тактом генерация данных производится по TTL сигналу внутреннего таймера DAQ-устройства. Аппаратный таймер работает намного быстрее программных циклов. Также аппаратный таймер более точен по сравнению с программными циклами.

Примечание. Некоторые устройства не поддерживают аппаратное управление тактом. Обратитесь к руководству пользователя DAQ-устройства для получения подробной информации.

С программной точки зрения необходимо включить режим аппаратного задания временного такта выполнения операций и задать частоту оцифровки sample rate и ограниченный по времени режим работы sample mode (finite). Возможна генерация значений на одном канале или нескольких.

Режим генерации n значений имеет смысл использовать при создании изменяющегося в конечный интервал времени сигнала, например, фрагмента переменного тока.

Непрерывная генерация данных

Непрерывная генерация данных аналогична генерации n значений с отличием в том, что для остановки непрерывной генерации должно произойти какое-то событие. Если необходимо непрерывно генерировать сигнал, для этого устанавливается режим работы sample mode (continuous).

Синхронизация заданий

Когда устройство, управляемое NI-DAQmx, работает, оно производит операции. Два наиболее часто встречающихся действия - задание значения и начало его генерации. Каждое производимое действие NI-DAQmx вызвано чем-либо, либо имеет причину. Такие причины называются синхронным запуском.

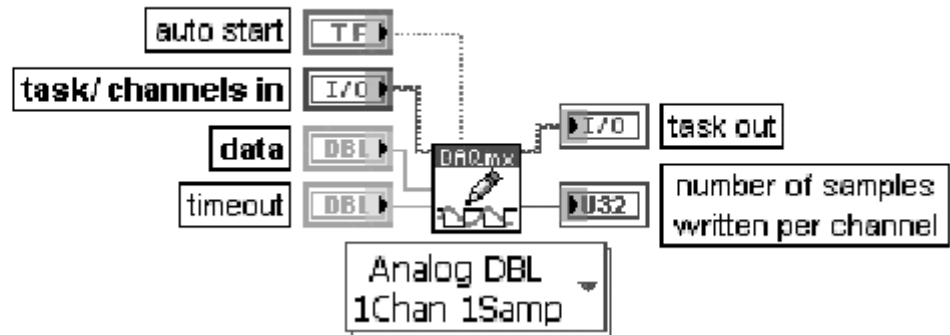
Start Trigger

Сигнал, запускающий генерацию данных.

Reference Trigger

Этот сигнал не поддерживается для аналогового выхода.

DAQmx Write



Записать в DAQmx

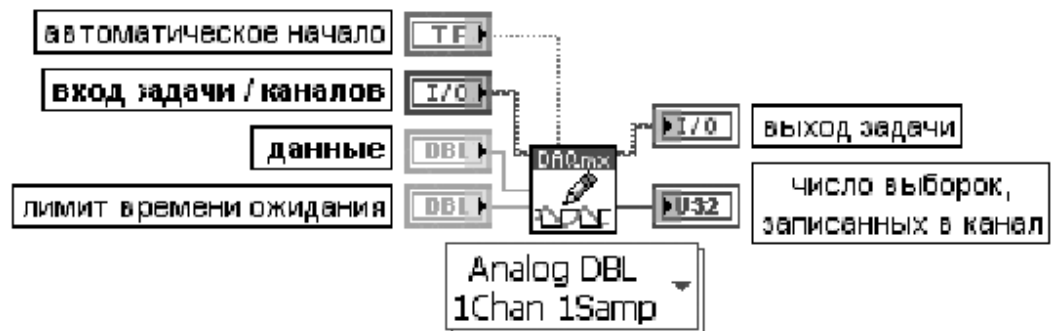


Рисунок 8.2 – Запись в DAQmx

ВП записывает выборки в задачи или каналы, определенные пользователем. Реализация этого полиморфного ВП предусматривает выбор формата записываемых выборок, выбор записи единичной выборки или совокупности выборок, а также выбор записи в один или несколько каналов.

Если задача использует синхронизацию по запросу, по умолчанию ВП Синхронизация D Синхронизация D Синхронизация D Синхронизация D Синхронизация DAQmx (DAQmx Timing) не используется, этот ВП возвращает только после того, как устройство сгенерирует все выборки. Если задача использует типы синхронизации, отличающиеся от синхронизации по запросу, этот ВП возвращает немедленно и не ожидает завершения генерации всех выборок.

Используемое приложение должно определять выполнение задачи, чтобы гарантировать, что устройство сгенерировало все выборки.

Функции установления времени

Функции установления времени (рис. 8.2) позволяют вводить задержки в работу ВП или измерять длительность выполнения отдельных операций, вводить текущее время и дату.

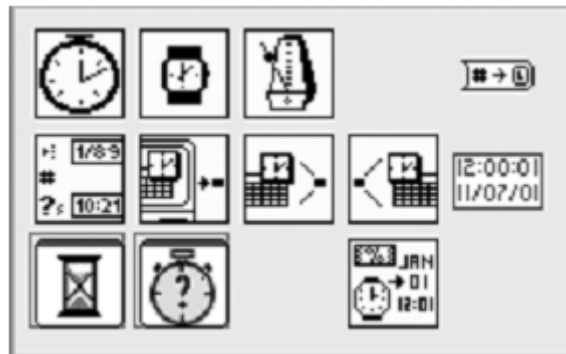


Рисунок 8.2 – Палитра функций установления времени

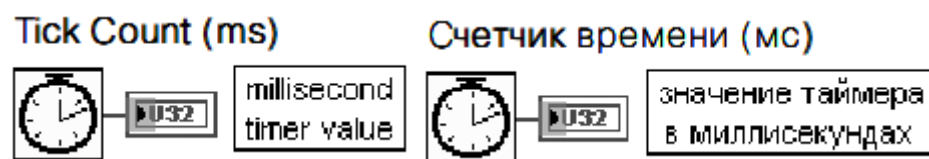


Рисунок 8.3 – Счетчик времени

Функция возвращает значение внутренних часов операционной системы в миллисекундах.

Начальное время отсчета (нуль миллисекунд) является неопределенным. Таким образом, невозможно преобразовать значение времени на выходе функции значение таймера в миллисекундах таймера в миллисекундах таймера в миллисекундах таймера в миллисекундах (millisecond timer value) в реальное время или дату. При использовании функции в операциях сравнения необходимо учитывать, что значение таймера сбрасывается в 0 после значения $2^{32}-1$.

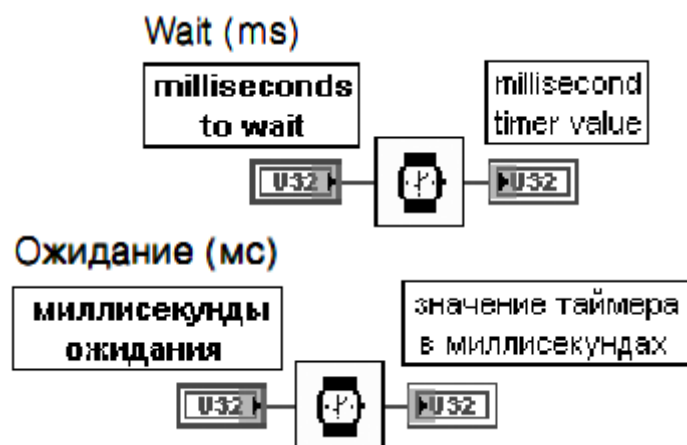


Рисунок 8.4 – Функция ожидание

Функция ожидает заданное число миллисекунд и возвращает значение таймера в миллисекундах. Данная функция выполняет асинхронные системные вызовы, но сама работает синхронно. Следовательно, она не завершит выполнение до истечения заданного времени.



Рисунок 8.5 – Функция Задержка до следующего кратного интервала

Функция заставляет ВП ожидать, пока показания внутренних часов не сравняются или не превысят кратного количества миллисекунд, поданных на вход функции кратность, миллисекунды (millisecond multiple). Данную функцию целесообразно использовать для синхронизации выполняемых операций. Так, например, функцию можно использовать в цикле для

управления скоростью его выполнения. При этом первый цикл может быть короче остальных.

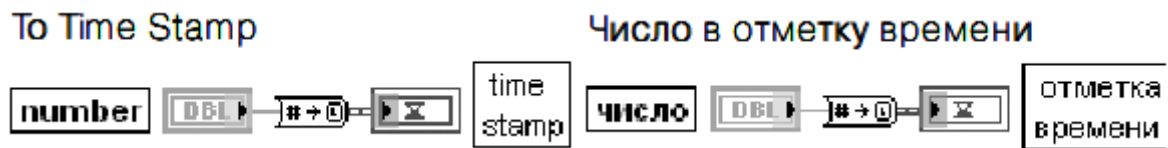


Рисунок 8.6 – Функция преобразует число в отметку времени

Функция преобразует число в отметку времени. Отметка времени представляет не зависящее от временного пояса число полных секунд, прошедших с 0:00 1 января 1904 года по Гринвичу.



Рисунок 8.7 – Функция преобразует значение времени

Функция преобразует значение времени или числовое значение в не зависящее от временного пояса число полных секунд, прошедших с 0:00 1 января 1904 года по Гринвичу, и возвращает кластер из девяти 32-битовых целых чисел со знаком, определяющих значение времени или числовое значение.

Компоненты кластера дата и время дата и время дата и время дата и время дата и время были рассмотрены выше при анализе функции Дату/время в секунды

Задание

Необходимо создать виртуальный прибор управляющий конфигурированием, запуском и отображением генерируемых и вычисляемых величин.

Прибор должен вычислять значения амплитуд сигнала заданной формы и генерировать на выходе подключенного устройства NI USB-6008 (или эмулированного устройства, поддерживающего генерацию аналогового сигнала), а также отображать осциллограмму этого сигнала на графике в масштабе.

Так же необходимо выбрать из таблицы 8.1 согласно варианту значения амплитуды и тип сигнала, задать их значения на фронтальной панели реализованного виртуального прибора.

Выполнение работы

На рисунке 8.2 отображена лицевая панель виртуального прибора, несущая в себе элементы управления и отображения.

Таблица 8.1 – Варианты значений активных и реактивных сопротивлений нагрузки.

A	Амплитуда, В	B	Тип сигнала (Signal type)
0	0,5	0	Sine Wave (Синусоидальный)
1	1	1	Triangle Wave (треугольная форма)
2	1,5	2	Square Wave (Квадратная форма)
3	1,2	3	Sawtooth Wave (Пилообразной формы)
4	0,7	4	Sine Wave (Синусоидальный)
5	0,9	5	Triangle Wave (треугольная форма)
6	0,25	6	Square Wave (Квадратная форма)
7	0,75	7	Sawtooth Wave (Пилообразной формы)
8	1,1	8	Sine Wave (Синусоидальный)
9	1,3	9	Triangle Wave (треугольная форма)

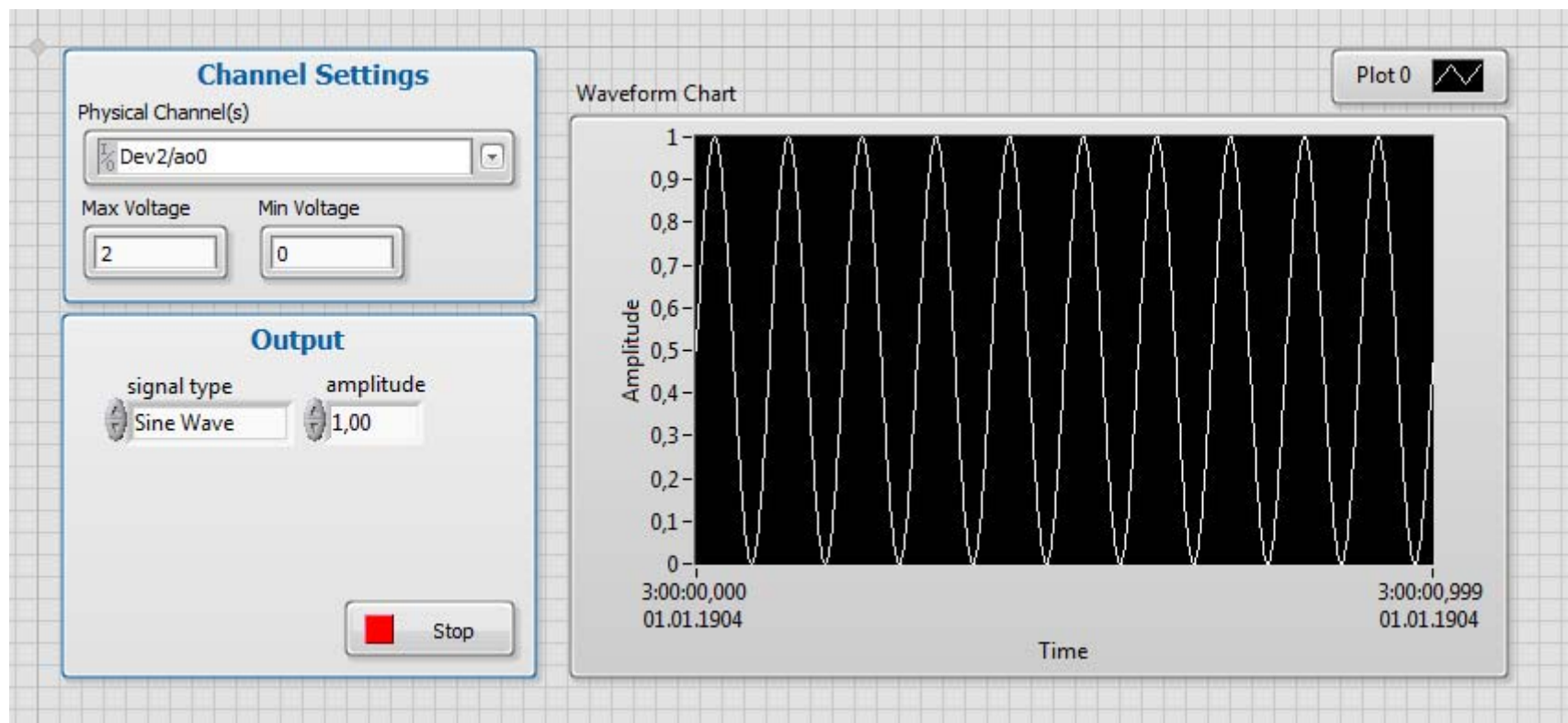


Рисунок 8.1 – Фронтальная панель виртуального инструмента

Результат выполнения работы.

В результате выполнения работы необходимо создать виртуальный прибор по выше предлагаемому заданию и написать отчет о лабораторной работе.

Отчет о лабораторной работе должен содержать:

1. Титульный лист;
2. Задание с указанием варианта;
3. Принтскрины фронтальной панели и блок диаграммы виртуального прибора;
4. Ответ на индивидуально заданный вопрос преподавателя по тематике лабораторной работы.

Контрольные вопросы

1. Какие параметры задаются в настройках каналов вывода аналогового сигнала?
2. Каким образом происходит обработка вероятных ошибок приложения?
3. Какой тип данных отображается на графике осциллограмм?
4. Объясните последовательность обработки событий в цикле с применением функции Задержки до следующего кратного интервала?
5. В какой палитре расположены функции создания сигнала произвольной формы?
6. Как сконфигурировать DAQ-устройства в MAX?
7. Для какого из типов циклов автоиндексация включена по умолчанию?
8. Что означает термин полиморфизм в Labview?
9. Какие три режима отображения имеет график Диаграмм (Waveform Chart)?

Литература: 2, 3, 4.

Лабораторная работа №9

Ввод аналогового сигнала с помощью DAQ-устройства его преобразование и передача по цифровому каналу

Цель работы: освоение технологии одновременной работы с аналоговыми и цифровыми сигналами используя программный и аппаратный функционал LabView. В связи с этим необходимо решить следующие задачи:

1. Ознакомление с техническими параметрами и вариантами подключения каналов устройства NI USB-6008.

1. Выполнить построение кода реализующего возможности измерения аналогового сигнала напряжения с заданными параметрами точности и скорости оцифровки, его преобразование и передачу по цифровой линии с последующим приемом цифровых данных и их декодирования в аналоговый сигнал с условиями согласно заданию.

2. Анализ потерь точности данных в результате преобразования.

Знания, умения и владения, приобретаемые обучающимся в результате освоения тем в рамках освоения компетенций: способностью осуществлять поиск, хранение, обработку и анализ информации из различных источников и баз данных, представлять ее в требуемом формате с использованием информационных, компьютерных и сетевых технологий (ОПК-1); способностью применять соответствующий физико-математический аппарат, методы анализа и моделирования, теоретического и экспериментального исследования при решении профессиональных задач (ОПК-2)

Знать: о связи дисциплины с другими изучаемыми дисциплинами; о роли дисциплины в дальнейшей научной и практической деятельности; о возможностях и преимуществах моделирования в среде LabView; терминологию, основные понятия и определения; технологию графического программирования в среде LabView; типы данных и последовательность их обработки (ОПК-1); общие вопросы построения измерительных систем с

использованием: современных компьютерных технологий, принципы решения технических задач; измерений с применением современных аппаратных и программных средств (ОПК-2);

Уметь: формулировать постановку задачи, определять цель (ОПК-1); оптимально выбирать программно-аппаратные средства для решения задач и проблем измерительной техники; разработать алгоритмы измерения и обработки данных (ОПК-2);

Владеть: терминологией, основными понятиями и определениями; технологией графического программирования в среде LabVIEW; навыками обработки различных типов данных; общими вопросами построения измерительных систем с использованием современных компьютерных технологий (ОПК-1); принципами решения технических задач измерений с применением современных аппаратных и программных средств (ОПК-2);

Теоретическая часть

Выше было отмечено, что основным элементом NI-DAQmx является задача.

Также было отмечено, что задача может быть создана как в программе MAX, так и в LabVIEW. В LabVIEW задача может быть создана несколькими способами:

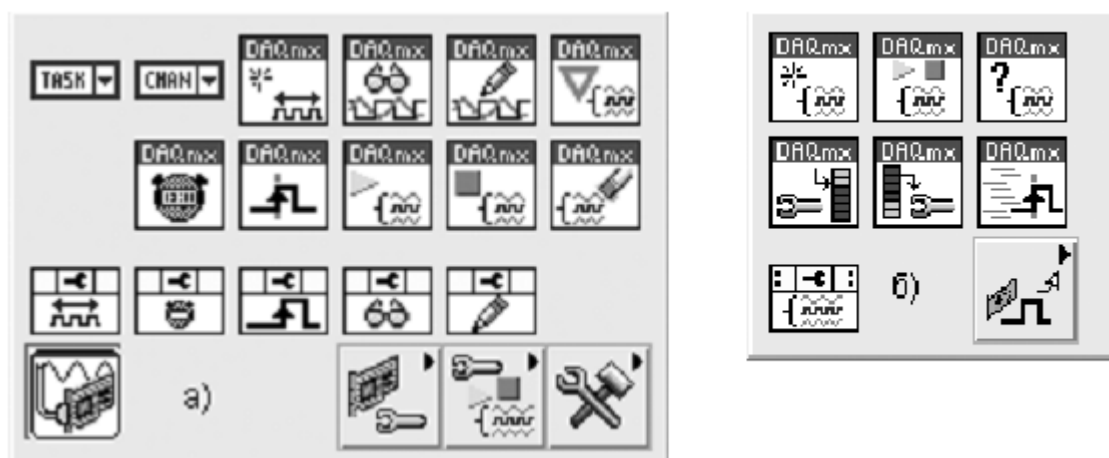


Рисунок 9.1 – Вид основной (а) и одной из дополнительных подпалитр (б) функций сбора данных DAQmx

1) путем размещения элемента управления Имя задачи DAQmx (DAQmx Task Name) на лицевой панели или константы с таким же именем (рис. 9.1а) на блок-диаграмме и выбора пункта Новая задача (New Task) из контекстного меню элемента или константы. При этом LabVIEW загружает данную задачу в память только один раз, даже если константа или элемент управления находятся в структуре цикла;

2) путем размещения Экспресс ВП Помощник DAQ (DAQ Assistant) на блок-диаграмме. Задача, созданная с помощью Экспресс ВП, является локальной по отношению к приложению и не может быть сохранена в MAX для использования в других приложениях. Для использования задачи в других приложениях или для генерации кода Экспресс ВП Помощник DAQ должен быть преобразован в константу Имя задачи DAQmx. Преобразование осуществляется с помощью пункта Преобразовать в константу имени задачи (Convert to Task Name Constant) контекстного меню иконки Экспресс-ВП;

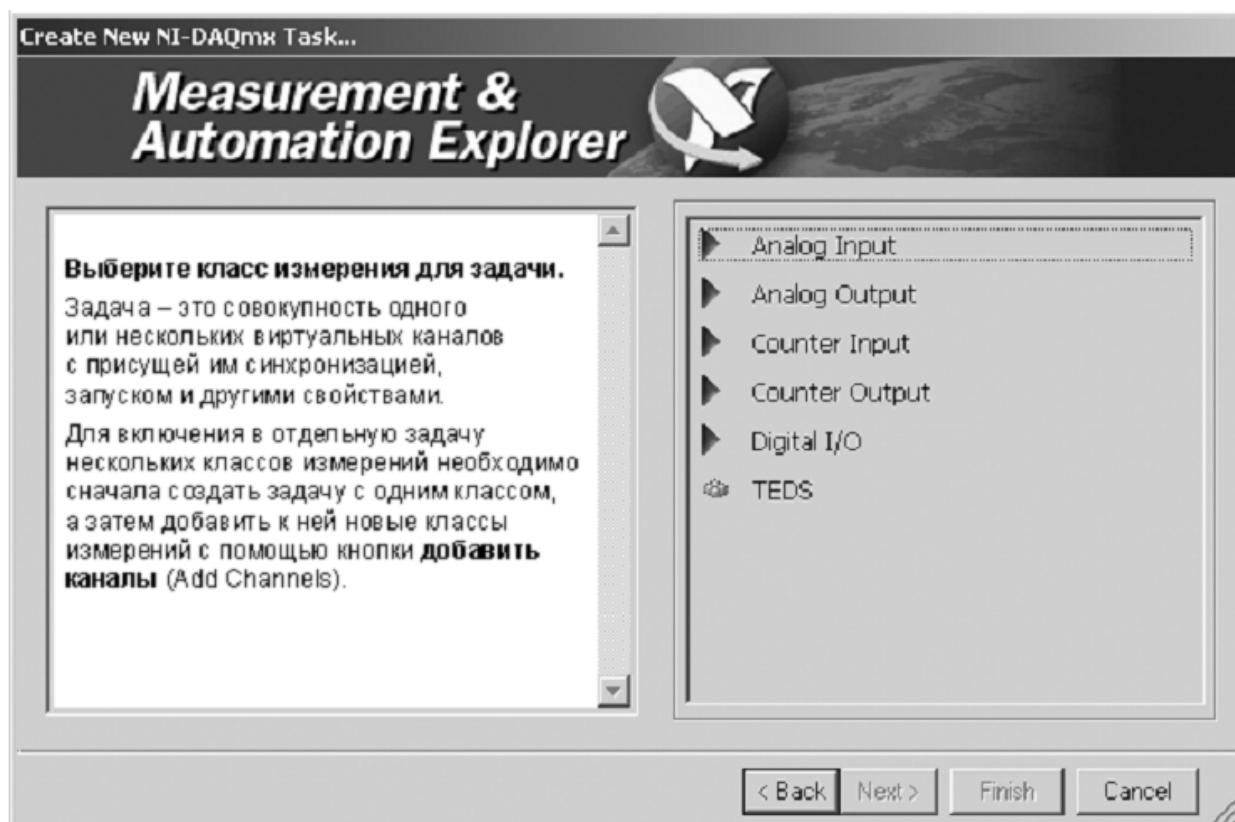


Рисунок 9.2 – Вид диалогового окна создания новой задачи

3) путем выбора строки ВП из шаблона (VI from Template) в окне первоначального запуска и последующего выбора ВП Сбор данных с помощью NI DAQmx (Data Acquisition with NI DAQmx).

При создании задачи как в MAX, так и в LabVIEW открывается диалоговое окно «Создать новую» (Create New). На рисунке 9.2 показан вид диалогового окна при его открытии из MAX. С помощью набора кнопок в правой части окна производится выбор класса измерения или генерации сигнала. Набор основных классов задач включает Аналоговый ввод (Analog Input), Аналоговый вывод (Analog Output), Подсчет на входе (Counter Input), Подсчет на выходе (Counter Output) и Цифровой ввод/вывод (Digital I/O). Более подробная иерархия видов задач измерения и генерации сигналов приведена в таблице.

Таблица 9.1 – Виды измерений и соответствующие им классы

Класс измерения	Вид измерения
1	2
Аналоговый ввод (Analog Input)	Напряжение (Voltage) Температура (Temperature) • терморезистор с подачей тока терморезистор с подачей тока терморезистор с подачей тока терморезистор с подачей тока терморезистор с подачей тока (Iex Termistor) • терморезистор с подачей напряжения (Vex Termistor) • термометр сопротивления (RTD) • термопара (Thermocouple) Деформация (Strain) Ток (Current) Сопротивление (Resistance) Частота (Frequency) Положение (Position)

Продолжение таблицы 9.1

1	2
	• датчик перемещения на основе линейного дифференциального трансформатора (LVDT) • датчик углового перемещения на основе вращающегося датчик углового перемещения • дифференциального трансформатора (RVDT) Ускорение (Acceleration) Специальное напряжение с возбуждением (Custom Voltage with Excitation) Звуковое давление (Sound Pressure
Аналоговый вывод (Analog Output)	Напряжение (Voltage) Ток (Current)
Подсчет на входе (Counter Input)	Подсчет перепадов (Edge Count) Частота (Frequency) Период (Period) Ширина импульса (Pulse Width) Полупериод (Semi Period) Интервал между двумя перепадами (Two Edge Separation)
Подсчет на выходе (Counter Output)	Вывод импульса (Pulse Output)
Цифровой ввод/вывод (Digital I/O)	Ввод линии (Line Input) Ввод порта (Port Input) Вывод линии (Line Output) Вывод порта (Port Output)
Датчики с электронными таблицами (TEDS)	Те же виды измерений, что и в п. Аналоговый ввод

После выбора задачи производится выбор физического канала (каналов) из списка установленных.

После выбора физического канала и ввода имени задачи в следующем диалоговом окне производится настройка параметров физического канала. Настройка в зависимости от вида измерения включает установку параметров в следующих разделах окна:

1) раздел Установки (Settings). На рисунке 9.3 показан вид данного раздела для задачи измерения температуры с помощью термометра сопротивления (Resistance temperature detector – RTD);

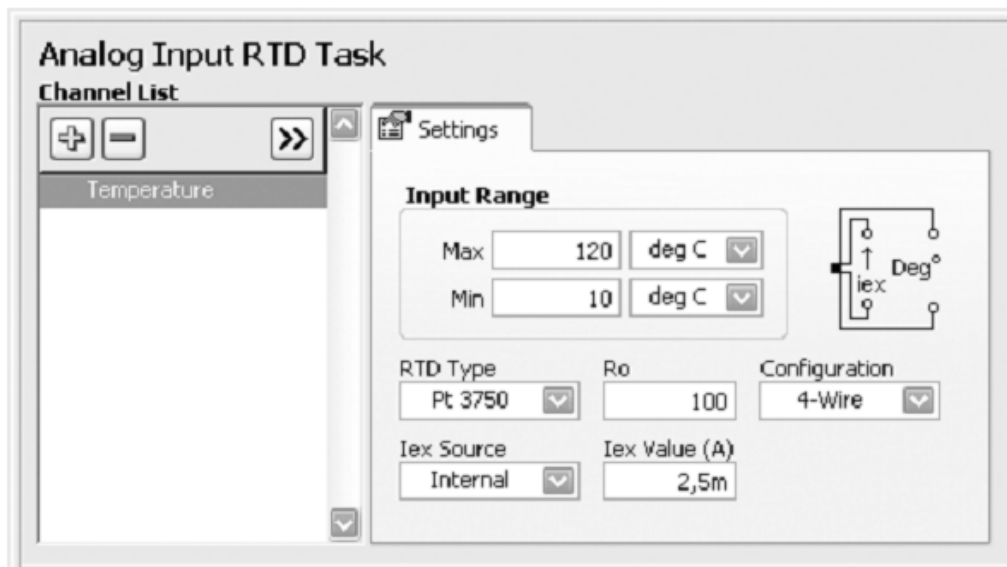


Рисунок 9.3 – Вид раздела Установки диалогового окна настройки параметров физического канала.

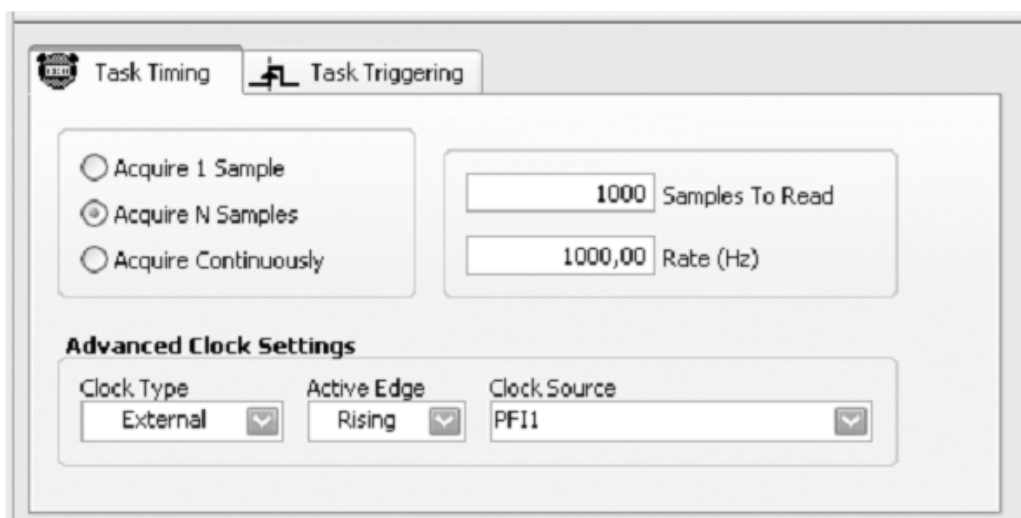


Рисунок 9.4 – Вид раздела Синхронизация задачи.

2) раздел Синхронизация задачи (Task Timing). На рисунке 9.4 показан вид раздела с выбранными опциями Получение N выборок (Acquire N

Samples), Внешний (External) в окне Тип тактирования (Clock Type), Нарастающий (Rising) в окне Активный фронт (Active Edge) и PFI1 в окне Источник тактирования (Clock Source);

3) раздел Запуск задачи (Task Triggering). Раздел содержит две группы параметров: Начало (Start) и Реперная точка (Reference). Параметры группы Начало определяют сигнал, запускающий сбор данных, а параметры группы Реперная точка – сигнал, устанавливающий реперную точку в наборе входных значений. На рисунке 9.5 показан возможный вид раздела с параметрами групп Начало и Реперная точка. В состав первой группы входят параметры Цифровой фронт (Digital Edge) в окне Тип запуска (Trigger Type), Нарастающий (Rising) в окне Фронт (Edge) и Источник (Source). Набор параметров второй группы включает Аналоговый фронт в окне Тип запуска, Выборки перед запуском (Pretrigger Samples), Спадающий (Falling) в окне Наклон (Slope) и Уровень (Level).

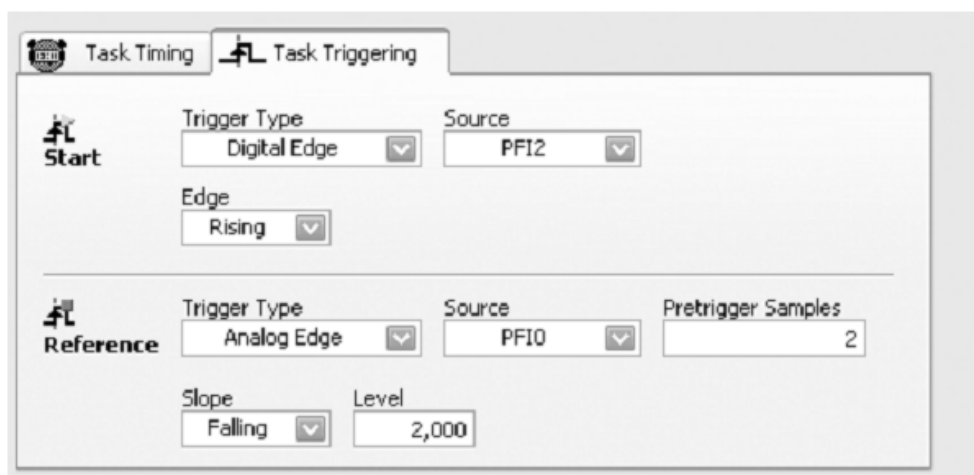


Рисунок 9.5 – Вид раздела Запуск задачи

Ввод и вывод цифровых сигналов.

Измерение и генерация цифровых сигналов используется в большом количестве приложений, включая мониторинг систем безопасности. В основном, генерация и измерение цифровых сигналов применяется в лабораторных исследованиях, тестировании продуктов и промышленных процессах мониторинга и контроля.

Цифровой ввод-вывод представляет собой чтение или запись значения в цифровую линию или во весь цифровой порт, состоящий из совокупности линий.

Вы можете использовать цифровые линии DAQ-устройства для сбора цифровых значений. Этот сбор данных основывается на программном задании временного такта выполнения операций. На некоторых устройствах можно настраивать цифровые линии независимо для генерации или сбора данных. Каждая цифровая линия является отдельным каналом в LabVIEW.

Вы можете использовать цифровые порты DAQ-устройства для сбора данных совокупности цифровых линий. Этот сбор данных основывается на программном задании временного такта выполнения операций. Вы можете настраивать цифровые порты независимо для генерации или сбора данных. Каждый цифровой порт является отдельным каналом в LabVIEW.

Технические характеристики USB-6008/6009

УСД NI USB-6008/6009 присоединяется к компьютеру посредством интерфейса full-speed USB и содержит восемь каналов ввода аналоговых сигналов (AI), два канала генерации аналоговых сигналов (AO), 12 каналов цифрового ввода/вывода (DIO) и 32-разрядный счетчик.

Таблица 9.1 – Различие характеристик USB-6008 и USB-6009

Характеристика	USB-6008	USB-6009
Разрешение при аналоговом вводе	12 бит (дифференциальное подключение) 11 бит (подключение с общим проводом)	14 бит (дифференциальное подключение) 13 бит (подключение с общим проводом)
Максимальная частота дискретизации, один канал*	10 кГц	48 кГц
Максимальная частота дискретизации, несколько каналов (Суммарная)*	10 кГц	42 кГц
Конфигурация цифрового ввода/вывода	Открытый коллектор	Открытый коллектор или активное управление
* Может зависеть от конфигурации системы.		

Аппаратная часть

На следующей блок-схеме приведены ключевые компоненты плат USB-6008/6009.

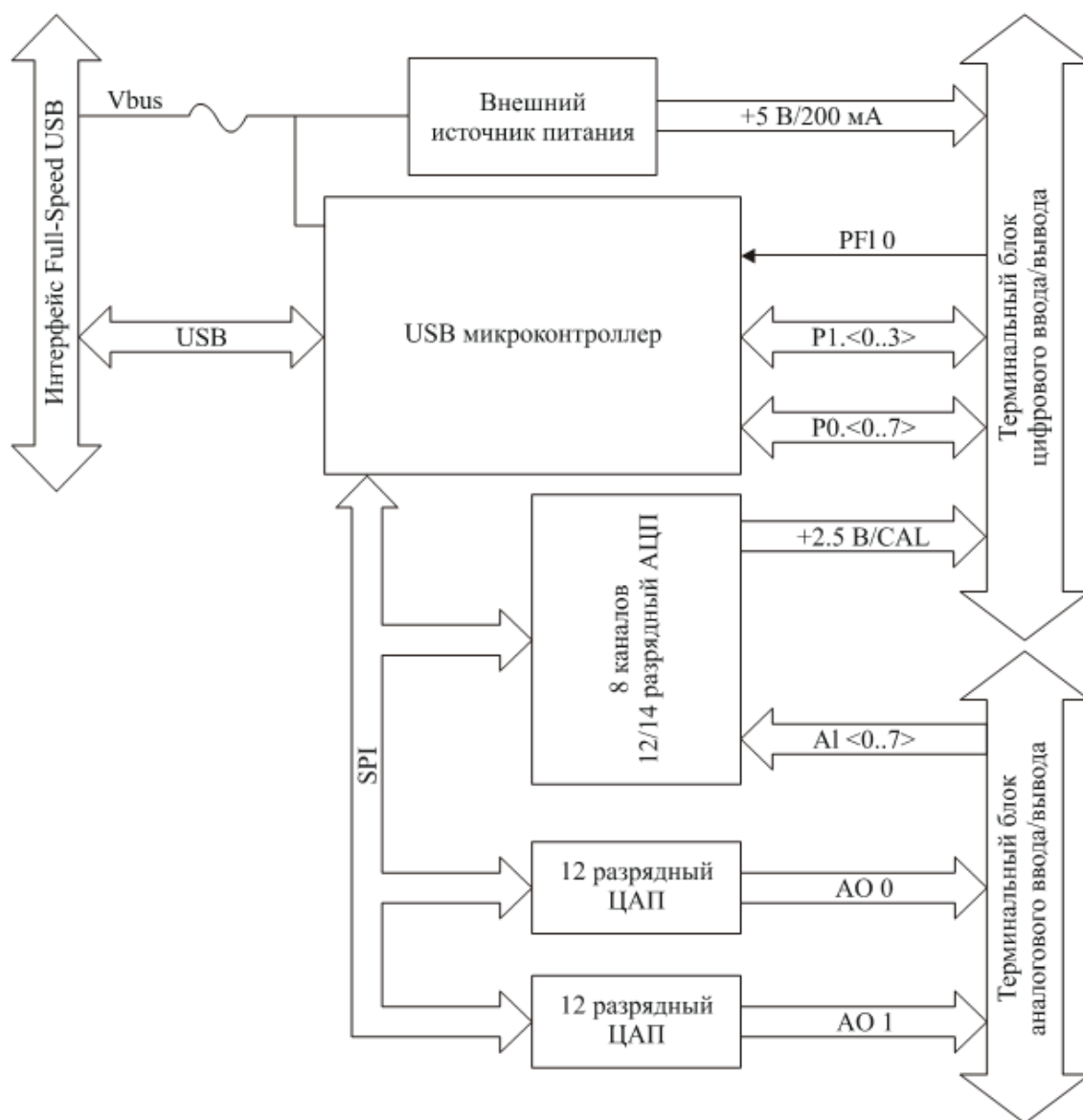


Рисунок 9.6 – Блок-схема устройства

Монтаж устройства

Выполните монтаж устройства следующим образом:

1. Установите блоки клемм с винтовым креплением в соответствующие разъемы. Примечание. Платы сбора данных USB-6008/6009 поставляются вместе с наклейками, облегчающими идентификацию подключаемых сигналов.

2. В инструкции по эксплуатации посмотрите правильную ориентацию наклеек и прикрепите их к соответствующим блокам клемм. До тех пор пока наклейки не прикреплены, терминальные блоки можно установить в любой из разъемов.

Соединительная панель

Комплект поставки USB-6008/6009 включает по одному съемному терминальному блоку для цифровых и аналоговых сигналов. Эти блоки предназначены для винтового крепления проводов с диаметром от 16 до 28 в единицах американского стандарта AWG.

В таблицах 9.1 и 9.2 приведены обозначения терминалов ввода/вывода аналоговых и цифровых

Таблица 9.1 – Обозначения терминалов ввода/вывода аналоговых сигналов

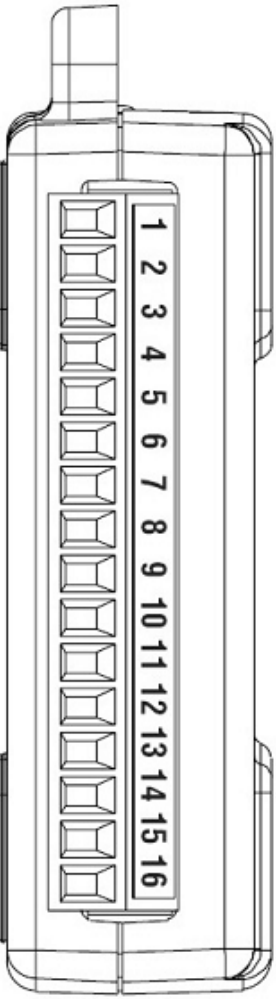
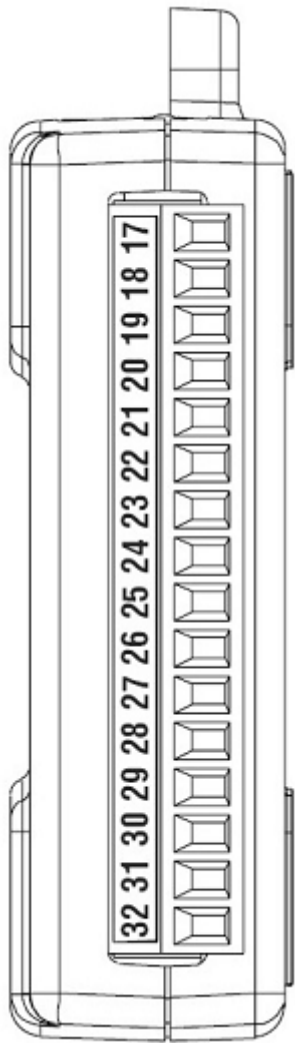
Модуль	Терминал	Схема подключения с общим проводом	Дифференциальная схема подключения
	1	GND	GND
	2	AI 0	AI 0+
	3	AI 4	AI 0-
	4	GND	GND
	5	AI 1	AI 1+
	6	AI 5	AI 1-
	7	GND	GND
	8	AI 2	AI 2+
	9	AI 6	AI 2-
	10	GND	GND
	11	AI 3	AI 3+
	12	AI 7	AI 3-
	13	GND	GND
	14	AO 0	AO 0
	15	AO 1	AO 1
	16	GND	GND

Таблица 9.2 – Обозначения терминалов ввода/вывода цифровых сигналов

Модуль	Терминал	Сигнал
	17	P0.0
	18	P0.1
	19	P0.2
	20	P0.3
	21	P0.4
	22	P0.5
	23	P0.6
	24	P0.7
	25	P1.0
	26	P1.1
	27	P1.2
	28	P1.3
	29	PFI 0
	30	+2.5 V
	31	+5 V
	32	GND

Описание сигналов.

В таблице 5 описаны сигналы, которые можно подключать к терминалам ввода/вывода.

Таблица 9.2 – Описание сигналов

Наименование сигнала	Источник опорного напряжения	Направление передачи	Описание
GND	—	—	Ground – Точка подключения опорного напряжения при

			аналоговом вводе в схеме с общим проводом, точка вытекания тока смещения при дифференциальном вводе сигнала, при генерации аналоговых сигналов, при цифровом вводе/выводе, для обеспечения питания постоянным напряжением +5 В, для обеспечения постоянным опорным напряжением +2.5 В
AI <0..7>	Различный	Ввод	Каналы ввода аналоговых сигналов 0 – 7. Служат для измерений напряжения в схеме с общим проводом; каждая линия предназначена для ввода одного сигнала. В дифференциальной схеме AI 0 и AI 4 являются положительной и отрицательной линиями канала 0. Следующие пары линий также образуют дифференциальные входные каналы: <AI 1, AI 5>, <AI 2, AI 6> и <AI3, AI7>.
AO 0	GND	Вывод	Канал вывода аналогового сигнала – Генерация напряжения в нулевом канале
AO 1	GND	Вывод	Канал вывода аналогового сигнала – Генерация напряжения в первом канале
P1.<0..3> P0.<0..7>	GND	Двунаправленные	Ввод/вывод цифровых сигналов – Каждый из каналов можно индивидуально настраивать на ввод или

			вывод.
+2.5 В	GND	Вывод	Источник опорного напряжения.
+5 В	GND	Вывод	Источник постоянного напряжения + 5 В и максимальным током 200 мА.
PFI 0	GND	Ввод	В зависимости от настройки, ввод цифрового триггера или счетчика.

Задание

Необходимо создать виртуальный прибор управляющий конфигурированием, запуском и отображением генерируемых и вычисляемых величин.

Для корректной работы виртуального прибора необходимо соединить логические выводы мультиметра NI USB-6008 по схеме указанной на рисунке 9.7, эмитирующей цифровой канал.

Прибор должен измерять аналоговый сигнал напряжения при помощи эмулируемого устройства (виртуальный сигнал от созданного в предыдущей лабораторной эмулированного NI USB-6008), преобразовывать его в цифровой формат и организовывать его передачу по цифровому каналу. Цифровой канал представлен передающим портом P0 (каналы P0.0-P0.3) устройства USB-6008 и принимающим портом P1 (каналы P1.0-P1.3).

В лабораторной работе используется принцип ступенчатого регулятора напряжения. Измерив отклонение напряжение от номинального значения выбирается ступень регулятора, обеспечивающая уровень напряжения близкий к номинальному в пределах допустимого значения отклонения уровня напряжения.

Необходимо выбрать из таблицы 9.3 согласно варианту значение номинального напряжения и, задав его значение на фронтальной панели, вычислить значение шага регулятора при условии, что напряжение после

регулятора не будет отклоняться от номинального значения больше чем на 5%.

Таблица 9.3 – Варианты значений номинального напряжения.

В	0	1	2	3	4
В	5	6	7	8	9
Uном, кВ	6	10	15	20	35

Примечание: В – вторая цифра номера варианта (№ вариант – последние две цифры номера зачетной книжки)

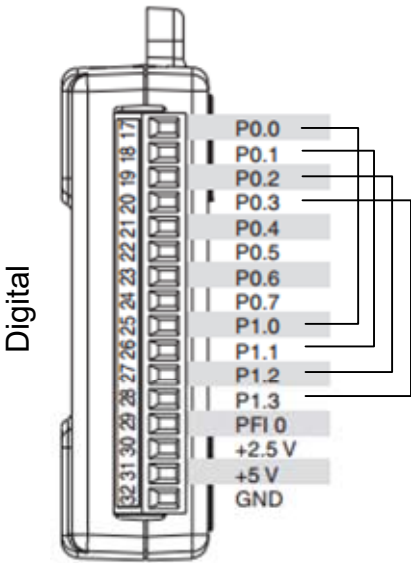


Рисунок 9.7 – Схема соединения цифровых каналов NI USB-6008

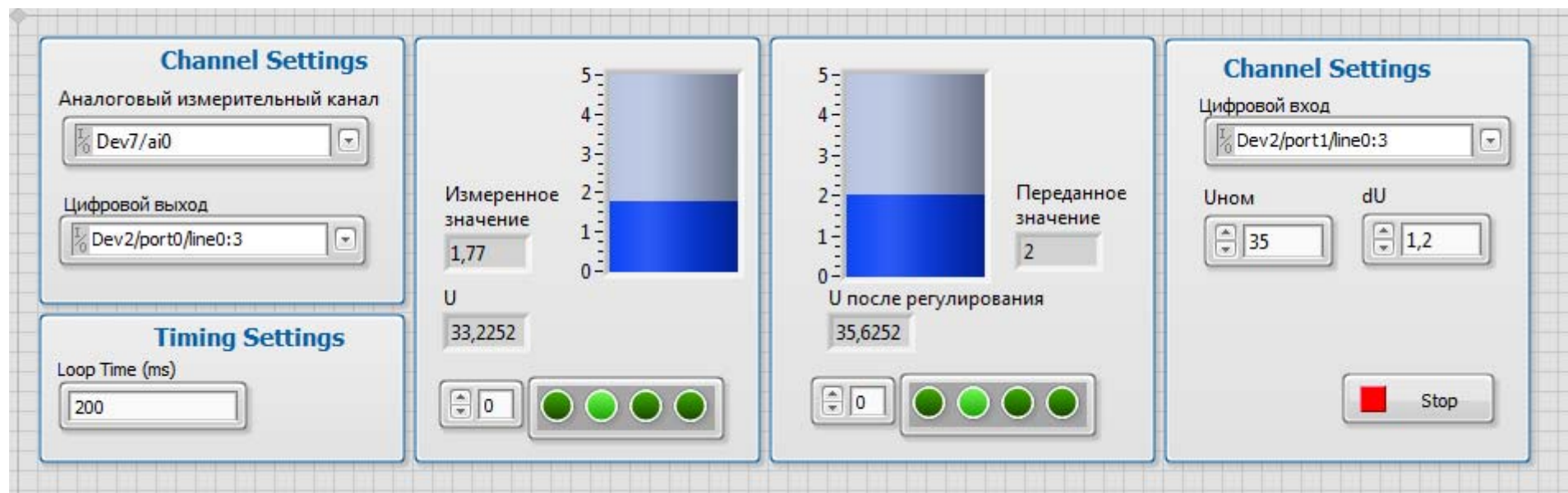


Рисунок 9.8 – Фронтальная панель виртуального прибора ввода/вывода аналоговых и цифровых сигналов

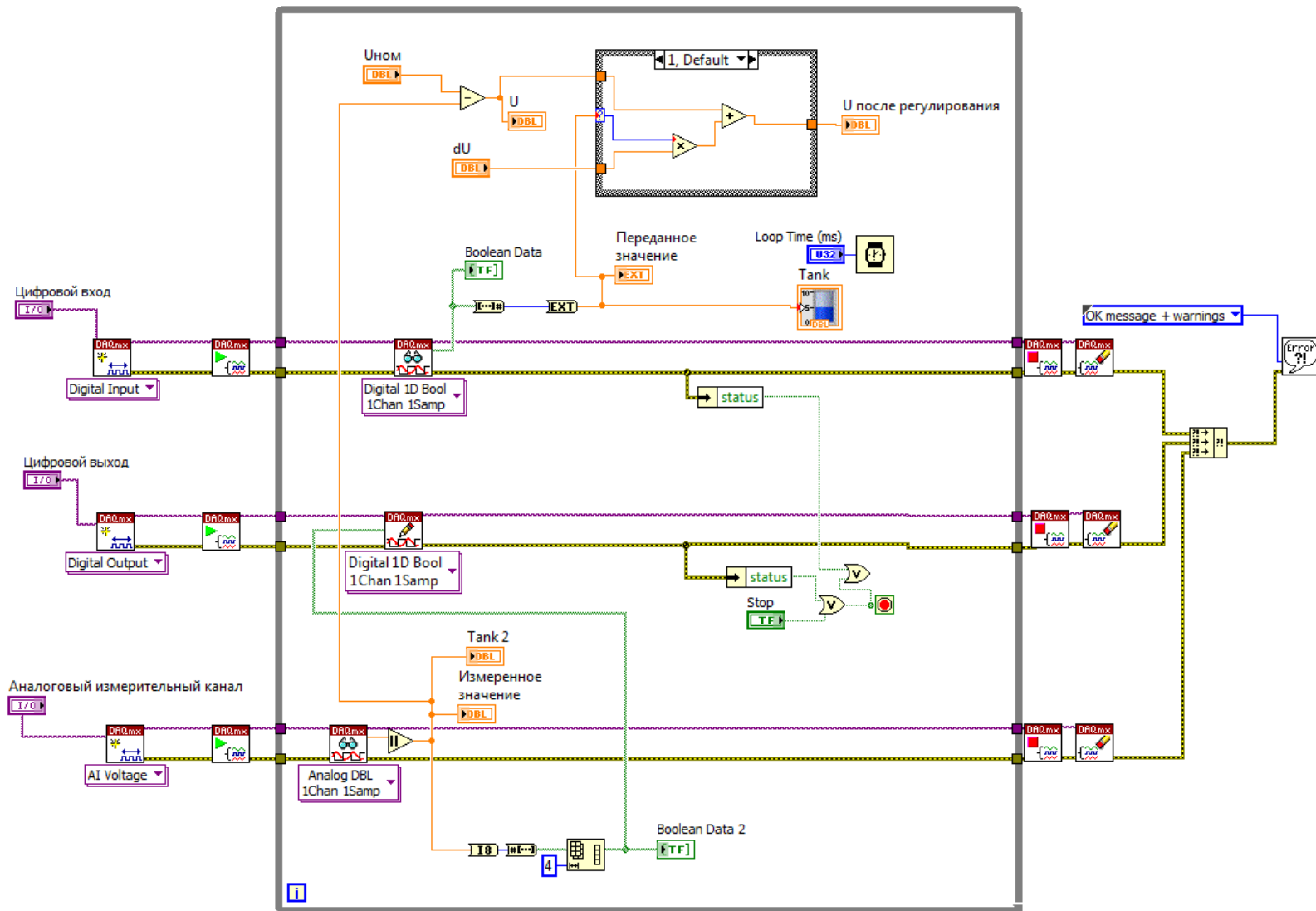


Рисунок 9.9 – Блок-диаграмма виртуального прибора ввода/вывода аналоговых и цифровых сигналов

Результат выполнения работы.

В результате выполнения работы необходимо создать виртуальный прибор по выше предлагаемому заданию и написать отчет о лабораторной работе.

Отчет о лабораторной работе должен содержать:

1. Титульный лист;
2. Задание с указанием варианта;
3. Принт-скрины фронтальной панели и блок диаграммы виртуального прибора;
4. Значение номинального напряжения по заданию и вычисленное значение ступени регулятора напряжения.

Контрольные вопросы

1. Что такое цифровой сигнал?
2. Что такое цифровая линия?
3. Какие параметры задаются при настройке цифрового канала?
4. Варианты настройки аналогового измерительного канала?
5. Объясните обозначение названий цифровых каналов P0.0 от P1.0?
6. Какие типы каналов ввода и вывода имеет NI USB 6008?
7. Как зависит потеря точности данных от характеристик цифровой линии?
8. Какую функцию выполняет элемент Loop Time?
9. Поясните поведение программы в результате возникновения ошибки оборудования?
10. Конвертеры каких типов данных применяются в работе?

Литература: 2, 3.

Литература

1. Суранов, Александр Яковлевич. LabVIEW 7: справочник по функциям. – М.: ДМК Пресс, 2005. – 510 с.
2. Тревис, Джеффри. LabVIEW для всех: пер. с англ. – М.: ПриборКомплект: ДМК Пресс, 2005. – 537 с.
3. Евдокимов, Юрий Кириллович. Labview для радиоинженера: от виртуальной модели до реального прибора: учебное пособие для вузов. – М.: ДМК Пресс, 2007. – 400 с
4. Учебный курс LabVIEW ОСНОВЫ I. Авторский коллектив: Михеев П.К., Крылова С.И., Лукьянченко В.А., Урюпина Д.С. Учебно-технический центр «Систем автоматизации научных исследований» Международный учебно-научный лазерный центр, Московский государственный университет им, Март 2007. – 365 с.
5. <http://e-lib.kemtipp.ru/uploads/25/eteo156.pdf> - Учебный курс Labview.

**Инструкция
по технике безопасности
при выполнении работ в лабораториях кафедры «Автоматизированных
электроэнергетических систем и электроснабжения»**

Перед началом лабораторной работы

1. Изучить инструкцию по охране труда при работе студентов в электротехнических лабораториях.
2. Пройти инструктаж на рабочем месте. Ознакомиться с безопасными методами выполнения лабораторной работы.
3. Изучить теоретические положения требований методики проведения лабораторной работы.
4. Получить допуск у преподавателя на выполнение лабораторной работы.
5. Ознакомиться с приборами тушения пожара и правилами обращения с огнетушителями ОУ-03, ОП-05, ОВЭ-6(з)-АВЕ-01.
6. Внимательно осмотреть лабораторный стенд, наличие и соответствие предохранителей, приборов и крепление деталей и узлов.

Требования безопасности во время работы

1. Лабораторная работа должна выполняться не менее чем двумя студентами.
2. Перед началом сборки цепи следует убедиться в том, что стенд обесточен.
3. Сборку схемы осуществлять исправными соединительными проводами, используя при этом приведенные в лабораторной работе принципиальные схемы экспериментов.
4. Собранная цепь проверяется преподавателем и может включаться

только по его разрешению. О включении напряжения предупреждают всех членов бригады, выполняющих работу.

5. При работе с регулируемыми источниками питания не следует превышать допустимый уровень напряжения.

6. Изменения схемы производят только при выключенном напряжении на стенде, а вновь собранная схема перед подачей на нее напряжения проверяется преподавателем.

7. По окончании испытания или при перерыве в работе схему отключают от напряжения питания. Разборку схему осуществляют по разрешению преподавателя. Запрещается оставлять включенную схему без наблюдения.

8. Запрещается включение лабораторного стенда в сеть в присутствии легковоспламеняющихся жидкостей или в атмосфере, содержащей горючие газы.

9. При возникновении неисправностей, а также в случае появления запаха, свидетельствующего о возгорании электрических проводов или компонентов, следует незамедлительно прекратить работу с лабораторным стендом, выключив его из сети, известить об этом преподавателя и обратиться к обслуживающему персоналу.

10. По окончании экспериментов установить тумблеры в положение «отключено», а рукоятки регулируемых источников в нулевое положение. Разобрать схему и сдать соединительные провода. Разборку схемы осуществляют по разрешению преподавателя.

Требования безопасности в аварийных ситуациях

1. В ситуации, угрожающей аварией (скачке тока в цепи; появлении дыма или запаха горящей изоляции или др.) обучающийся должен:

1.1. Немедленно прекратить выполнение работы.

1.2. Обесточить прибор, установку (лабораторный стенд), выключив тумблер СЕТЬ.

2. В нестандартной ситуации - выключить автоматический выключатель (или рубильник) на распределительном щите лаборатории.

Требования безопасности по окончании выполнения лабораторной работы

1. Записать результаты измерений в отчет по лабораторной работе.
2. Получить разрешение на выключение лабораторного стенда.
3. Привести в порядок рабочее место и лабораторный стенд.
4. Убрать электроизмерительные приборы в отведенное место.
5. Сдать лабораторный стенд лаборанту.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Методические указания для самостоятельной работы
студента по дисциплине
«Моделирование в среде LabView»
направление подготовки 13.03.02. -
"Электроэнергетика и электротехника"
Направленность(профиль) - Электроснабжение

Ставрополь 2026

Оглавление

Введение	3
1. Основные мотивы самостоятельной работы студентов	4
2. Способы самостоятельной работы при чтении учебной и научной литературы	5
3. Формы ведения записей.....	15
4. Как слушать и конспектировать лекции	24
5. Использование компьютера в процессе самостоятельной работы студентов.	28
6. Подготовка к экзаменам и зачетам.....	36
Рекомендуемая литература.....	39

Введение

Основная задача высшего образования заключается в формировании творческой личности специалиста, способного к саморазвитию, самообразованию, инновационной деятельности. Решение этой задачи вряд ли возможно только путем передачи знаний в готовом виде от преподавателя к студенту. Для решения этой задачи в учебные планы всех специальностей включена самостоятельная работа, составляющая 50% учебного времени студента.

В связи с этим, студенту из пассивного потребителя знаний необходимо превратиться в активного их творца, умеющего сформулировать проблему, проанализировать пути ее решения, найти оптимальный результат и доказать его правильность. Происходящая в настоящее время реформа высшего образования связана по своей сути с переходом парадигмы обучения к парадигме образования. В этом плане следует признать, что самостоятельная работа студентов является не просто важной формой образовательного процесса, а должна стать его основой. Это предполагает ориентацию на активные методы овладения знаниями, развитие творческих способностей студентов, переход от поточного к индивидуализированному обучению с учетом потребностей и возможностей личности.

В широком смысле под самостоятельной работой студентов следует понимать совокупность всей самостоятельной деятельности студентов как в учебной аудитории, так и вне ее, в контакте с преподавателем и в его отсутствие.

Самостоятельная работа реализуется:

1. Непосредственно в процессе аудиторных занятий – на лекциях, практических и семинарских занятиях, при выполнении лабораторных работ.
2. В контакте с преподавателем вне рамок расписания – на консультациях по учебным вопросам, в ходе творческих

контактов, при ликвидации задолженностей, при выполнении студентом учебных и творческих задач.

Границы между этими видами работ достаточно размыты, а сами виды самостоятельной работы пересекаются.

Следует с первых дней обучения в вузе приучать себя к целесообразному распределению занятий по месяцам и неделям семестра. При этом очень важна равномерная работа как основное условие успешных занятий в вузе.

Высшей ступенью обучения является приобретение навыков самостоятельной творческой работы, а творческому применению знаний нужно учиться параллельно с их приобретением.

1. Основные мотивы самостоятельной работы студентов

Активная самостоятельная работа студентов возможна только при наличии серьезной и устойчивой мотивации. Самый сильный мотивирующий фактор – подготовка к дальнейшей эффективной профессиональной деятельности. Среди внутренних факторов, способствующих активизации самостоятельной работы выделяют следующие:

1. *Полезность* выполняемой работ заключается в том, что результаты самостоятельной работы могут быть использованы на семинарских и практических занятиях, лабораторном практикуме, при подготовке публикации. Другим вариантом использования фактора полезности является активное применение результатов работы в профессиональной подготовке. Так, например, если студент получил задание на дипломную работу на одном из младших курсов, он может выполнять самостоятельные задания по ряду дисциплин гуманитарного и социально-экономического, естественно-научного и общепрофессионального циклов дисциплин, которые затем войдут в его выпускную квалификационную работу.

2. *Творческая деятельность.* Это может быть участие в научно-исследовательской, опытно-конструкторской или методической работе, проводимой на той или иной кафедре.
3. *Участие* в олимпиадах по учебным дисциплинам, конкурсам научно-исследовательских или прикладных работ и т.д.
4. *Участие* в научно – практических конференциях.
5. *Подготовка публикаций* для сборников тезисов и докладов научно-практических конференций, журналов, учебных пособий и т.д.
6. *Участие в грантовых конкурсах.*

2. Способы самостоятельной работы при чтении учебной и научной литературы

При остром недостатке времени у студентов встает вопрос об оптимизации обучения, то есть такой организации учебного процесса, которая обеспечила бы условия для продуктивного самообучения и самовоспитания. Важнейшую роль здесь играет овладение способами самостоятельной работы. Речь идет о том, что чтобы прежде всего научиться рациональному использованию времени при работе с книгой.

В этом особенно нуждаются первокурсники, которые еще недостаточно владеют навыками умственной деятельности, обеспечивающей успешное обучение.

Начинать самостоятельные занятия следует с первых же дней учебы в вузе. Первые дни семестра важны, чтобы включиться в работу, установить определенный равномерный ритм на весь семестр. Чтобы выполнить весь объем самостоятельной работы, необходимо заниматься самостоятельно по 4-5 часов ежедневно, кроме выходных дней.

Под ритмом работы понимают ежедневные занятия в одни и те же часы, при чередовании их с перерывами для отдыха. Вначале для организации ритмичной работы требуется сознательное напряжение воли, затем

принуждение снимается, возникает привычка и работа становится потребностью.

Ритмичная работа позволяет студенту заниматься много, не уставая, не снижая производительности и не перегружая себя. Для этого необходимо:

- Сменять один вид работы другим, что позволяет сохранять высокую работоспособность, поскольку при однообразных видах занятий человек утомляется больше, чем при работе разного характера.
- Заниматься несколькими предметами в один и тот же день не всегда целесообразно, поскольку при каждом переходе нужно вновь концентрировать внимание и затрачивать время.
- Умение сосредотачиваться – необходимое условие для умственного труда, иначе работа оказывается малопродуктивной и даже бесполезной.
- Начинать занятия немедленно, как только сел за стол. Следует начинать с уверенностью, что вскоре придет сосредоточенное состояние, но, если внимание наступает не сразу или нарушается на время, нужно выяснить и устранить причины этого.
- Нужно научиться не прерывать внимания, пока читаемое не получит логического завершения, пока не будет пройден какой-либо этап. Нередко внимание отвлекается посторонними мыслями, которые во время занятий следует решительно отгонять. Перерыв в занятиях следует приурочить к концу изучения параграфа, раздела или главы книги, так как в этом случае не будет потери времени при возобновлении работы. Умение сосредоточиться, углубиться в работу приобретается в результате практики, создающей определенные навыки.
- Повысить производительность умственного труда может порядок на рабочем месте и обстановка, благоприятствующая работе.

Большая часть самостоятельной работы студента состоит в **изучении литературы**. Одна из задач студента – научиться самостоятельно работать с книгой, а это требует определенных затрат энергии и времени. Поэтому надо научиться делать эту работу рационально, то есть необходимо учиться читать.

Как работать с учебной и научной книгой

Методы эффективной работы с книгой в целях развития интеллекта можно условно разделить на две группы:

1. Правильная организация процесса чтения
2. Повышение скорости чтения и восприятия.

В комплексе оба метода могут в 2-3 раза сократить время прочтения различных материалов.

При чтении текста мозг формирует «свою трактовку содержания» прочитанного. Происходит перекодирование сообщения на языке собственных мыслей читателя. Мозг выделяет «ядерное», сущностное значение из текста. Эффективность такой перекодировки зависит от осмысления и внимательности чтения.

Как показали эксперименты, знание и умелое применение некоторых упражнений дают возможность извлекать «ядерное» значение в тексте быстро и надежно. Эти упражнения основаны на использовании дифференциального алгоритма чтения. Центральное место в этом алгоритме занимает «блок доминанта». Это слово в переводе с латинского языка означает «господствующий, основной, главный». Что же такое доминанта применительно к тексту?

Доминанта – главная смысловая часть текста. Она выражается своими словами, на языке собственных мыслей, является результатом переработки текста, его осмысления в соответствии с индивидуальными особенностями читателя, выявления основного замысла автора.

Дифференциальный алгоритм чтения в соответствии с блоками позволяет реализовать логико-семантический анализ текста: вначале выделить ключевые слова, затем построить смысловые ряды и, наконец, выделив цепь знаний, сформулировать доминанту. Именно так и только так (по О.А. Андрееву) можно увидеть главное, действительно, проникнуть в суть вещей, явлений, излагаемых автором.

Возможны три основных способа чтения.

- Первый способ – артикуляция или проговаривание вслух (или почти вслух) того, что читаешь. Скорость такого чтения невелика.
- Второй способ – чтение про себя, при котором речевой процесс проявлен в форме внутренней речи, то есть без открытой артикуляции. Текст, при этом усваивается более эффективно. Способ в принципе допускает быстрое чтение.
- Третий, наиболее совершенный способ чтения – тоже молча, но в условиях максимального сжатия внутренней речи, при котором она проявляется в виде коротких залпов ключевых слов и смысловых рядов, адекватно отражающих смысл текста.

Итак, артикуляция замедляет процесс чтения и от нее необходимо избавиться. Однако не приведет ли сокращение артикуляции при повышении скорости чтения к снижению качества восприятия и осмысления полученной информации?

Как показали исследования психологов, иногда при чтении слова могут быть заменены наглядными представлениями, пространственными схемами. Целые группы слов – одним словом.

Быстро читающие люди обладают способностью, не проговаривая читаемый текст, сразу улавливать и фиксировать замысел автора, а затем усваивать его на уровне внутренней речи. В этом случае, несмотря на высокую скорость чтения, происходит глубокое понимание и усвоение прочитанного, так как основная идея понятна с самого начала. Задачу научиться такому чтению можно решать в два этапа. Первый предполагает

сокращение артикуляции, если она ярко выражена, второй – овладение приемами чтения, при которых текст воспринимается крупными информационными блоками.

Как известно, людей по способу восприятия и переработки информации делят на три типа: зрительный, слуховой и кинестетический. Люди зрительного типа при чтении используют код наглядных образов, тогда как люди слухового типа применяют менее производительный код речедвижений. Наблюдения за людьми, читающими быстро, показывают, что они, как правило, относятся к зрительному типу. Вот пример, как описывает О.Бальзак процесс быстрого чтения: «Впитывание мысли в процессе чтения достигло у него способности феноменальной. Взгляд его охватывал семь – восемь строчек сразу, и разум постигал смысл со скоростью, соответствующей скорости глаз. Часто одно-единственное слово позволяло ему усвоить смысл целой фразы».

Направленным обучением можно практически любого здорового человека научить в процессе чтения использовать код наглядных зрительных образов при соответствующем сокращении артикуляции.

С опорой на работу Л.Г. Одинцова «Как научиться хорошо учиться» (М., 1996) приводим следующие рекомендации по работе с книгой.

- В тексте всегда есть элементы, нахождение и использование которых позволяет извлечь требуемую информацию наиболее быстро. Например, при чтении учебника в первую очередь отыскивается наиболее важная информация данной главы, параграфа, а она часто следует после слов: в итоге, в результате, выводы и т.д.
- Попробуйте в процессе чтения мысленно заглянуть вперед, представить себе, о чем будет идти речь, к какому выводу придет автор, как далее будет строиться изложение и т.д., например, если описывается одна сторона явления, то, очевидно, далее будет

описана и другая и т.д. Это позволяет предварительно подготовиться к последующей информации.

- Хорошим упражнением по развитию навыков «предвидения» является остановка чтения в момент, когда, по вашему мнению, заканчивается какая-то часть текста. Попробуйте предугадать содержание следующей части.
- До начала чтения текста важно собрать о нем как можно больше информации, чтобы точнее представить, что можно получить из данного текста и как лучше работать с ним. Это помогут сделать название, автор, издательство, аннотация, оглавление, предисловие и заключение. Предварительное ознакомление с книгой перед настоящим чтением позволяет сберечь время и труд.

Как правило, предисловие пишется крупным специалистом в данной области, и поэтому излагаемая проблема показывается как бы целиком, в общем плане, без подробностей. А это позволяет лучше сориентироваться, начинать чтение, зная основную цель автора.

- Перед углубленным чтением любого текста (статьи, книги, конспекта, лекции перед экзаменом) сначала бегло просмотрите его целиком. При этом постарайтесь выявить основные стержневые идеи, наиболее крупные части и логику их изложения. Лишь после такого просмотра переходите к более детальному чтению.
- Перед чтением статьи или параграфа учебника попробуйте проделать следующее: прочитайте внимательно первый абзац, потом бегло просмотрите первые или последние фразы следующих абзацев (в них обычно содержится основная информация), обратите внимание на курсивы, разрядки, подзаголовочный текст и, наконец, внимательно прочтите один-два последних абзаца; постарайтесь выявить основное направление текста и его построение.
- Прочитав в тексте интересную идею, полезно остановить свое внимание на ней, прислушаться к тем мыслям, которые она у вас

вызвала, подумать о тех последствиях, которые из нее вытекают, попытаться развивать ее дальше.

- Существенно замедляют чтение регрессии – частые возвратные движения глаз, многократное повторное прочитывание материала. Возвратиться к уже прочитанному, но недостаточно хорошо понятому участку лучше всего, когда прочитан законченный смысловой фрагмент текста и сделана хотя бы попытка его осмысления, а не в процессе чтения предложения.
- Любой текст не однороден по своей информационной насыщенности. В некоторых предложениях, абзацах сконцентрировано очень много информации, например, формулируются основные положения, ведущие идеи и т.д., а другие служат лишь иллюстрацией, фоном. Таким образом, текст имеет «смысловой рельеф». Чем точнее читатель умеет определить степень важности каждого отрезка текста и приспособить к «смысловому барьеру» способ своего чтения (то есть замедлить и углубить в более важных местах и ускорять в менее важных), тем продуктивнее чтение. Постарайтесь гибко варьировать способ работы с текстом в соответствии с его «смысловым барьером».
- Чтобы чтение было эффективным, попробуйте по прочитанному всегда отвечать на 6 вопросов: *«Кто делает? Что делает? Когда? Почему? Где? Как?»*

Освойте технику быстрого чтения по специальной методике, например по книге О.А. Андреева, Л.Н. Хромова. Учитесь быстро читать (М., 1991).

Большое значение при чтении учебной и научной литературы имеет умение запоминать прочитанный материал, а для этого необходимо тренировать память. Существуют приемы, позволяющие тренировать память, которыми необходимо овладеть, что позволит повысить эффективность работы с учебной и научной литературой.

Тренировка памяти. В учебной деятельности важно не только, и не столько быстро читать, но и усваивать материал, сохранять в памяти. Память прекрасно тренируема и управляема. Однако, прежде чем ее развивать, подумайте, какая именно память вам нужна: на идеи, на логику изложения материала, на схемы и формулы. Это разные виды памяти и развивать их надо по-разному.

Наблюдая за собой, выясните, как вам легче запомнить информацию – если вы ее видите, слышите или записываете. В дальнейшем постарайтесь так организовать работу, чтобы максимально использовать ведущий тип своей памяти.

Если у вас хорошая **зрительная память**, то хорошо запоминаются рисунки, расположение информации на странице, цвет и т.д. помогите себе, выделяя цветными карандашами отдельные места конспекта, обводя рамками, делая значки, пометки на полях, представляя зрительно отдельные аспекты текста.

При хорошей **слуховой памяти** лучше запоминается звучащая речь. Используйте эту особенность, выделяя интонацией, тембром голоса отдельные места текста, слушая его в записи на магнитофоне, рассуждая в слух и т.д.

В случае **памяти на движение** помогает повторная сокращенная запись запоминаемого материала, например выводов, основных положений текста, рисование таблиц, графиков, схем, а при выполнении лабораторных работ лучше все потрогать и проделать самому.

Наряду с использованием ведущего типа памяти, специально позаботьтесь и о развитии отстающих, так как при многих видах профессиональной деятельности они также могут потребоваться.

Использование приемов логического, осмысленного запоминания в несколько раз повышает продуктивность деятельности. Например, при запоминании лекции, глав учебников особенно действенным является

основные аспекты содержания, но и запомнить логику – целесообразную связь отдельных частей материала.

Постарайтесь с первого курса развивать память на то, что непосредственно касается вашей будущей профессии. Это и основной круг идей данной отрасли знаний, и методы, и наиболее интересные факты, и фамилии ведущих специалистов области и т.д. при этом лучше не ждать, что запомнится само, а специально стараться запомнить нужное.

Вообще установка на запоминание, особенно длительное, положительно сказывается на прочности и точности сохранения материала в памяти. Прикажите себе запомнить надолго, а не так как нерадивый студент, спешно «набивающий» себе голову информацией непосредственно перед экзаменом с единственной целью – удержать выученное на один – два дня.

Любая информация запоминается лучше, если в ней намечены какие-то спорные моменты – ориентиры. И как по камушкам переходят реку, так и по этим ориентирам потом легче воспроизвести содержание. При запоминании текста выделяйте «смысловые опорные пункты», которые легко запоминаются, но с которыми тесно связаны целые фрагменты материала. Это может быть крылатая фраза, яркая цитата, пример, идея и т.д.

Материал запоминается непроизвольно, то есть легко и без затраты специальных усилий, если он является целью какой-либо поисковой деятельности. Например, если вы задались вопросом и нашли ответ на то, что долго искали, или нашли подтверждение гипотезы, которую вы сами выдвинули, то это запоминается само собой. Отсюда вывод – организуйте свою деятельность так, чтобы предмет запоминался, являлся целью этой деятельности. Например, ищите, выделяйте в тексте наиболее важные его положения – и они запомнятся, делите текст на части, анализируйте связи между ними – и запомнится логика текста.

При повторении курса лекций, запоминая материал по отдельным темам или даже вопросам, не забывайте повторить связь между ними.

Именно тогда в голове укладывается система знаний, которая гораздо эффективнее, чем разрозненные обрывки.

В процессе развития памяти старайтесь не использовать стихийно сложившиеся мнения, механическое зазубривание, а применяйте научно обоснованные методы сознательной и рациональной организации развития памяти и поиск новых приемов.

Предпосылкой хорошей памяти являются осознание человеком своей деятельности и разграничение информации на ту, которая решающим образом помогает скорейшему достижению своих целей, и на менее существенную информацию. Начинайте любое дело с четкой и ясной формулировки его цели; определите, какая информация может оказать решающее воздействие на ее достижение, и сконцентрируйтесь на ней.

Прочному запоминанию способствует многомодельность восприятия, то есть запоминаемый текст читается, проговаривается и прослушивается. Везде, где это возможно, постарайтесь использовать три приема (слух, зрение и чувства) обработки запоминаемой информации сразу несколько органов чувств.

Не очень осмысленную вами информацию, которую, тем не менее, надо запомнить, можно удержать с помощью ассоциативных приемов мнемотехники, суть которых в том, что новое связывается с известным не прямо, а через цепочку дополнительных промежуточных ассоциаций (помните цвета спектра – «Каждый охотник желает знать, где сидит фазан»). Везде, где трудно запомнить прямо, найдите дополнительный связующий мостик. Такими «связующими» мостиками являются буквальные «узелки» на память, завязываемые многими на носовом платке. В течение дня человек неизбежно пользуется носовым платком, а там – узелок, «напоминающий», что нужно не забыть сделать определенное дело.

Память будет работать прекрасно, если наряду с имеющимися приемами вы будете придумывать все новые, адекватные различным видам

информации. Если такая работа привычна для вас, то с каждым годом память будет становиться все более мощной и продуктивной.

3. Формы ведения записей

Самостоятельная работа с книгой может быть успешной, если текст не только прочитан, но и законспектирован. Существует несколько **форм записей**, но любая форма записи не даст нужного результата, если не будет пробуждать мысли того, кто ее ведет, если отсутствует активная работа ума и формирование своих выводов из прочитанного.

Выбор формы записи зависит от индивидуальных особенностей человека, его образованности и опыта. При этом не меньшую роль играет назначение записей, то есть то, какие задачи ставит перед собой человек (для самообразования, для выступления на семинаре, для использования в будущем).

Введение записей мобилизует наряду со зрительной памятью, также и моторную память. Кроме того, у человека, систематически ведущего записи изучаемой литературы, создается свой фонд материалов для быстрого повторения и мобилизации накопленных знаний.

Все записи должны быть убористыми и компактными. Интервалы между строками должны быть достаточными, чтобы вписывать дополнения. Рекомендуется вести записи ручкой, а карандашом или ручкой другого цвета пользоваться для отметок и выделений при последующей работе. Полезно также датировать записи.

Записи могут носить различный характер: план, выписки, тезисы, аннотирование, конспектирование, реферирование.

1. План - наиболее краткая формой записи. Это перечень вопросов, рассматриваемых в книге или статье. План обычно раскрывает структуру произведения, логику автора, способствует лучшей ориентации в содержании.

Так, составленным планом можно воспользоваться, чтобы вспомнить прочитанное или быстро отыскать в книге нужное место. Представление об основных пунктах плана дает оглавление книги, поэтому во многих случаях наименования глав и разделов можно использовать в качестве пунктов. Составление плана приучает логически мыслить, вырабатывать умение сжато и последовательно излагать суть вопроса в письменной и устной форме.

Существует два способа составления плана: работа над ним по ходу чтения и составление плана после ознакомления с произведением. При этом план получается более последовательным и стройным.

Трудность составления плана состоит в том, что надо выяснить для себя, прежде всего, построение изучаемого текста, ход мыслей автора и лишь затем изложить содержание работы кратко и ясно. При всей своей краткости план дает представление о содержании прочитанного. Следует учесть, что форма плана не исключает цитирования отдельных мест и обобщений. Различают простой и развернутый план. В отличие от простого плана развернутый план не только содержит перечисление вопросов, но и раскрывает основные идеи произведения, может включать выдержки из него, схемы, таблицы. Планом, особенно развернутым, необходимо пользоваться при написании выступления или статьи.

В целом развернутый план дает гораздо большее представление о произведении, его основных идеях, задачах, которые в нем решаются. Он может включать положения, замечания, собственные мысли студента.

Важно знать, что составление планов помогает вырабатывать способность к отвлеченному, абстрактному мышлению, но наибольшую пользу составление плана даст подготовленным лицам, которые бывают достаточно лишь взглянуть на перечень основных вопросов, чтобы воспроизвести содержание прочитанного.

2. Тезисы – более сложная и совершенная форма записи, чем составление плана.

Это сжатое изложение основных мыслей прочитанного произведения или подготовляемого вступления. Особенностью тезисов является их утвердительный характер.

В них сосредотачивается самое главное, только выводы и обобщения, в них меньше доказательств, иллюстрации и пояснений. Тезисы не должны повторять дословно текст, но в ряде мест могут быть близки к нему, воспроизводя некоторые характерные выражения автора, важные для понимания хода его мыслей. Составление тезисов помогает глубже понять основные идеи произведения, выделить главное в нем; приучают сжато, точно и четко сформулировать свои мысли, повышает культуру речи и письма. При составлении тезисов учитывают следующее. Прежде всего, если произведение небольшое, необходимо внимательно изучать его в целом, если большое – изучать по главам и разделам. Затем, когда будут ясны основные идеи, кратко и последовательно излагать их в виде пунктов.

Различают простые и сложные, развернутые тезисы. Если записывают только утверждение чего – либо, такой тезис называют простым, а сложным тезисом будет выражение главной мысли, содержащее, кроме утверждения, еще и краткое ее доказательство.

Часто тезисы формулируются самим автором как выводы и обобщения в заключении книги или разделах книги. Нередко тезисы выделяются в тексте другим шрифтом.

Рекомендуется делать тезисные записи своими словами, причем можно записывать один абзац за другим, учитывая смысловую связь между ними. Но в большинстве случаев следует составлять сводный тезис, сложный по форме. При этом объединяется несколько утверждений, тесно связанных между собой.

Тезисы по содержанию очень близки к **конспекту**, но конспект носит более описательный характер, и его положения не столь категоричны, как в тезисах. Кроме того, конспект представляет собой более полную форму записи.

Следует отметить, что различие между формами записей условно, но в любой форме запись – важнейшая часть самостоятельной работы с книгой.

3. Выписки. Это записи текста из книги: теоретических положений, статистических данных, имеющих по мнению читателя важное значение.

Достоинство выписок состоит в точности воспроизведения текста книги, удобстве пользования записями при последующей работе, в накоплении обобщений и фактического материала. Выписки полезны для повторения, освежения в памяти прочитанного, для быстрой мобилизации своих знаний, когда необходимо в короткий срок вспомнить материал. Выписки выделяют из текста самое главное и тем самым помогают глубже понять его. Без них трудно обойтись при подготовке доклада, реферата, выступления. Выписки следует рассматривать как составную часть тезисов и конспектов.

Выписывать текст можно и по ходу чтения и после его завершения. В последнем случае надо замечать места, которые потом будут выписаны. Необходимо каждую выписку снабжать ссылкой на источник с указанием соответствующей страницы. Это нужно, чтобы в последствии можно было быстро найти в книге соответствующее место. Целесообразно выписывать из текста только такие места, в которых содержится самое главное, суть вопроса. Выписки должны быть ориентированы на изучение произведения в целом, а не отдельных мест, поскольку положения, вырванные из общего контакта, понимаются нередко совсем не так, как этого хотел автор. Иначе говоря, отдельно взятые, лишенные пояснений выдержки могут быть не поняты или поняты неправильно.

Выписки бывают дословные (цитаты) и «свободные», когда мысли автора излагаются своими словами. Следует учесть, что большие отрывки, которые трудно цитировать, целесообразнее в краткой форме переложить своими словами, но «яркие» и важные места лучше выписывать дословно. Каждую цитату следует заключать в кавычки. Если ее берут из середины

предложения, то после вводных кавычек ставят три точки. Ставят их и в конце цитаты, если из предложения опущены последние слова.

Следует знать, что какого-либо единого метода выписок, годного для всех случаев, не существует, поскольку у каждого человека свои особенности мышления и восприятия, свой подход к теме. Все это влияет на содержание и характер выписок.

Выписки рекомендуется хранить в картотеке, конвертах или папках, на которых следует обозначить общую тему.

4. Аннотация – еще одна форма записи, являющаяся кратким обобщением содержания книги. Ею удобно пользоваться, если имеется намерение вернуться к изучаемому произведению. Аннотация может быть необходима и для того, чтобы не забыть о нем.

Для составления аннотации надо сначала полностью прочитать и глубоко продумать произведение. При всей своей краткости аннотация может содержать отдельные фрагменты авторского текста, а не только оценку книги или статьи.

5. Резюме очень близко к аннотации. Это запись, являющаяся краткой оценкой прочитанного материала. Различие между ними состоит в том, что аннотация сжато характеризует произведение в целом, а резюме концентрирует внимание на его выводах, главных итогах.

6. Конспект – наиболее совершенная и наиболее сложная форма записи. Слово «конспект» происходит от латинского «conspectus», что означает «обзор, изложение». В правильно составленном конспекте обычно выделено самое основное в изучаемом тексте, сосредоточено внимание на наиболее существенном, в кратких и четких формулировках обобщены важные теоретические положения.

Конспект представляет собой относительно подробное, последовательное изложение содержания прочитанного. На первых порах целесообразно в записях ближе держаться тексту, прибегая зачастую к

прямому цитированию автора. В дальнейшем, по мере выработки навыков конспектирования, записи будут носить более свободный и сжатый характер.

Конспект книги обычно ведется в тетради. В самом начале конспекта указывается фамилия автора, полное название произведения, издательство, год и место издания. При цитировании обязательная ссылка на страницу книги. Если цитата взята из собрания сочинений, то необходимо указать соответствующий том. Следует помнить, что четкая ссылка на источник – непереносимое правило конспектирования. Если конспектируется статья, то указывается, где и когда она была напечатана.

Конспект подразделяется на части в соответствии с заранее продуманным планом. Пункты плана записываются в тексте или на полях конспекта. Писать его рекомендуется четко и разборчиво, так как небрежная запись с течением времени становится малопонятной для ее автора. Существует правило: конспект, составленный для себя, должен быть по возможности написан так, чтобы его легко прочитал, и кто-либо другой.

Формы конспекта могут быть разными и зависят от его целевого назначения (изучение материала в целом или под определенным углом зрения, подготовка к докладу, выступлению на занятии и т.д.), а также от характера произведения (монография, статья, документ и т.п.). Если речь идет просто об изложении содержания работы, текст конспекта может быть сплошным, с выделением особо важных положений подчеркиванием или различными значками.

В случае, когда не ограничиваются переложением содержания, а фиксируют в конспекте и свои собственные суждения по данному вопросу или дополняют конспект соответствующими материалами их других источников, следует отводить место для такого рода записей. Рекомендуется разделить страницы тетради пополам по вертикали и в левой части вести конспект произведения, а в правой свои дополнительные записи, совмещая их по содержанию.

Конспектирование в большей мере, чем другие виды записей, помогает вырабатывать навыки правильного изложения в письменной форме важные теоретических и практических вопросов, умение четко их формулировать и ясно излагать своими словами.

Таким образом, составление конспекта требует вдумчивой работы, затраты времени и труда. Зато во время конспектирования приобретаются знания, создается фонд записей.

Конспект может быть текстуальным или тематическим. В **текстуальном конспекте** сохраняется логика и структура изучаемого произведения, а запись ведется в соответствии с расположением материала в книге. За основу тематического конспекта берется не план произведения, а содержание какой-либо темы или проблемы.

Текстуальный конспект желательно начинать после того, как вся книга прочитана и продумана, но это, к сожалению, не всегда возможно. В первую очередь необходимо составить план произведения письменно или мысленно, поскольку в соответствии с этим планом строится дальнейшая работа. Конспект включает в себя тезисы, которые составляют его основу. Но, в отличие от тезисов, конспект содержит краткую запись не только выводов, но и доказательств, вплоть до фактического материала. Иначе говоря, конспект – это расширенные тезисы, дополненные рассуждениями и доказательствами, мыслями и соображениями составителя записи.

Как правило, конспект включает в себя и выписки, но в него могут войти отдельные места, цитируемые дословно, а также факты, примеры, цифры, таблицы и схемы, взятые из книги. Следует помнить, что работа над конспектом только тогда будет творческой, когда она не ограничена текстом изучаемого произведения. Нужно дополнять конспект данными из другими источниками.

В конспекте необходимо выделять отдельные места текста в зависимости от их значимости. Можно пользоваться различными способами: подчеркиваниями, вопросительными и восклицательными знаками,

репликами, краткими оценками, писать на полях своих конспектов слова: «важно», «очень важно», «верно», «характерно».

В конспект могут помещаться диаграммы, схемы, таблицы, которые придадут ему наглядность.

Составлению **тематического конспекта** предшествует тщательное изучение всей литературы, подобранной для раскрытия данной темы. Бывает, что какая-либо тема рассматривается в нескольких главах или в разных местах книги. А в конспекте весь материал, относящийся к теме, будет сосредоточен в одном месте. В плане конспекта рекомендуется делать пометки, к каким источникам (вплоть до страницы) придется обратиться для раскрытия вопросов. Тематический конспект составляется обычно для того, чтобы глубже изучить определенный вопрос, подготовиться к докладу, лекции или выступлению на семинарском занятии. Такой конспект по содержанию приближается к реферату, докладу по избранной теме, особенно если включает и собственный вклад в изучение проблемы.

Следующим методом самостоятельной работы с книгой является **реферирование** на определенную тему. Слово реферат употребляется в двух различных значениях:

1. Краткое изложение содержания книги, научной работы;
2. Доклад за заданную тему на основе критического образа литературных источников.

7. Реферат – это один из самых сложных видов самостоятельной работы с книгой, а для этого следует овладеть более простыми приемами работы – разработкой плана, составлением тезисов и конспектов. Подготовка реферата и выступление с его изложением углубляет знания, расширяет кругозор, приучает логически, творчески мыслить, развивать культуру речи.

При просмотре литературы намечается ориентировочный план реферата, в который включается обычно 3-4 основных вопроса или раздела. Каждом из разделов формулируются подвопросы, помогающие последовательно раскрыть содержание проблемы.

В процессе изучения материала формулировки подвопросов и разделов обычно уточняются. При реферировании следует делать выписки, записывать мысли, возникающие при чтении; следует также точно записывать и определения тех понятий, которые будут использованы в реферате. Из прочитанной литературы нужно заимствовать не буквальный текст, а важнейшие мысли, идеи, теоретические положения; можно цитировать небольшие отрывки, приводить диаграммы, схемы, чертежи, но главное – высказывать собственные соображения по вопросам реферата. Приведенные выше советы следует рассматривать как примерные, предполагающие и другие подходы, поскольку у каждого человека вырабатываются свои приемы и навыки составления рефератов. Большую помощь в работе над рефератом оказывают предисловия к монографиям и сборникам. В них можно найти сведения о цели издания, а также о существующих пробелах в исследовании.

При разработке плана реферата важно учитывать, чтобы каждый его пункт раскрывал одну из сторон избранной темы, а все пункты в совокупности охватывали тему целиком. Различают несколько композиционных решений реферата: во-первых, хронологическое, когда тема раскрывается в исторической последовательности; во-вторых, описательное, при котором тема расчленяется на составные части, в целом раскрывающие определенное явление; в-третьих, аналитическое, когда тема исследуется в ее причинно-следственных связях и взаимозависимых проблемах. Важно следить за тем, чтобы каждый пункт плана был соотнесен с главной темой и не содержал повторения в других пунктах. Важными разделами реферата является вступление и заключение. Во вступлении надо обосновать актуальность темы, обозначить круг составляющих ее проблем, четко и кратко определить задачу своей работы. В заключении делаются краткие выводы, подводятся итоги. В конце реферата должен быть приложен список литературы.

В отличие от тематического конспекта реферат требует большей творческой активности, самостоятельности в обобщении изученной литературы, умения логически стройно изложить материал, оценить различные точки зрения на исследуемую проблему и высказать о ней собственное мнение. В реферате важно связать теоретические положения с практикой.

Итак, реферат – это самостоятельное произведение автора, которое должно свидетельствовать о знании литературы по данной теме, ее основной проблематике, отражать точку зрения автора реферата на эту проблематику, его умение осмысливать явления жизни на основе теоретических знаний.

При оценке реферата обычно руководствуются следующими критериями:

1. Удалось ли его автору раскрыть сущность данной проблемы;
2. Сумел ли автор показать связь рассматриваемой проблемы с жизнью;
3. Проявил ли автор самостоятельность и творческий подход в изложении реферата;
4. Можно ли считать реферат логически стройным и т.д.

4. Как слушать и конспектировать лекции

Основы знаний закладываются на лекциях, им принадлежит ведущая роль в учебном процессе. На лекциях дается самое важное, основное в изучаемой дисциплине. Основные задачи, стоящие перед лектором: помочь студентам понять основы и усвоить материал на самой лекции, дать указания на то, что требует наибольшего внимания, учить правильному мышлению и создавать ясное представление о методологии изучаемой науки.

Лекции являются эффективным видом занятий для формирования у студентов способности быстро воспринимать новые факты, идеи, обобщать их, а также самостоятельно мыслить.

Лектор излагает теоретический и практический материал, относящийся к основному курсу. Из большого числа монографий, учебников, сборников лектор выбирает самое главное, помогает усвоить логику рассуждений. Интонацией голоса и манерой изложения лектором подчеркивает наиболее существенное, выделяет главное и второстепенное.

Лектор может приводить наблюдения и факты из своего личного опыта, что придает материалу убедительность, повышает интерес к предмету лекции, способствует его усвоению.

Важно помнить, что лекция – это творческий процесс, в котором участвуют одновременно и лектор, и студенты, поэтому она требует атмосферы сотрудничества и уважительного отношения к труду лектора.

Студенту следует научиться понимать и основную идею лекции, а также, следуя за лектором, участвовать в усвоении новых мыслей. Но для этого надо быть подготовленным к восприятию очередной темы. Время, отведенное на лекцию, можно считать использованным полноценно, если студенты понимают роль лектора, задачи лекции, если работают вместе с лектором, а не бездумно ведут конспект.

Подготовленным можно считать такого студента, который, присутствуя на лекции, усвоил ее содержание, а перед лекцией припомнил материал раздела, излагаемого на ней или просмотрел свой конспект, или учебник.

Перед лекцией необходимо прочитывать конспект предыдущей лекции, а после окончания крупного раздела курса рекомендуется проработать его по конспектам и учебникам.

Для наиболее важных дисциплин, вызывающих наибольшие затруднения, рекомендуется перед каждой лекцией просматривать содержание предстоящей лекции по учебнику с тем, чтобы лучше воспринять

материал лекции. В этом случае предмет усваивается настолько, что перед экзаменом остается сделать немного для закрепления знаний.

Важно помнить, что ни одна дисциплина не может быть изучена в необходимом объеме только по конспектам. Для хорошего усвоения курса нужна систематическая работа с учебной и научной литературой, а конспект может лишь облегчить понимание и усвоение материала.

Основная задача при слушании лекции – учиться мыслить, понимать идеи, излагаемые лектором. Большую помощь при этом может оказать конспект. Передача мыслей лектора своими словами помогает сосредоточить внимание, не дает перейти на механическое конспектирование. Механическая запись лекции приносит мало пользы.

Ведение конспекта создает благоприятные условия для запоминания услышанного, т.к. в этом процессе принимают участие слух, зрение и рука. Конспектирование способствует запоминанию только в том случае, если студент понимает излагаемый материал. При механическом ведении конспекта, когда просто записываются слова лектора, присутствие на лекции превращается в бесполезную трату времени.

Некоторые студенты полагают, что при наличии учебных пособий, учебников нет необходимости вести конспект. Такие студенты нередко совершают ошибку, так как не используют конспект как средство, позволяющее активизировать свою работу на лекции или полнее и глубже усвоить ее содержание.

Определенная часть студентов считает, что конспекты лекции могут заменить учебники, поэтому они стремятся к дословной записи лекции и нередко не задумываются над ее содержанием. В результате при разборе учебного материала по механической записи требуется больше труда и времени, чем при понимании и кратком конспектировании лекции.

Конспект ведется в тетради или на отдельных листах. Записи в тетради легче оформить, их удобно брать с собой на лекцию или практические занятия. Рекомендуется в тетради оставлять поля для дополнительных

записей, замечаний и пунктов плана. Но конспектирование в тетради имеет и недостаток: в нем мало места для пополнения новыми материалами, выводами и обобщениями. В этом отношении более удобен конспект на отдельных листах (карточках). Из него нетрудно извлечь отдельную необходимую запись, конспект можно быстро пополнить листами, в которых содержатся новые выводы, обобщения, фактические данные. При подготовке выступлений, докладов легко подобрать листки из различных конспектов и свести их вместе. В результате такой работы конспект может стать тематическим.

Но вести конспект на отдельных листках или карточках более трудоемко, чем в тетради. Карточки легко рассыпать и перепутать, приходится обзаводиться ящичками для хранения карточек, возникает необходимость на каждом листке писать его порядковый номер.

Но затрата труда и времени окупается преимуществами конспектирования на карточках перед конспектом в тетради.

Рекомендуется делать такие карточки, которые помещаются в обычный почтовый конверт. Карточки удобно тасовать, менять при необходимости их последовательность, раскладывать на столе для обзора. Например, учителю истории карточки служили бы долго. Перед уроком можно взять соответствующий конверт и найти в нем материал по узловым вопросам темы, записанный еще в вузе.

При конспектировании допускается сокращение слов, но необходимо соблюдать меру. Каждый студент обычно вырабатывает свои правила сокращения. Но если они не введены в систему, то лучше их не применять, т.к. случайные сокращения ведут к тому, что спустя некоторое время конспект становится непонятным.

Следует знать, что не существует какого-либо единого, годного для всех метода конспектирования. Каждый ведет записи так, как ему представляется наиболее целесообразным и удобным. Собственный метод складывается по мере накопления опыта, но во всех случаях надо стремиться к

тому, чтобы конспективные записи были краткими и наилучшим образом содействовали глубокому усвоению изучаемого материала. Известный отечественный педагог В.А. Сухомлинский, рекомендовал учиться думать над конспектом уже на лекции и работать над записями ежедневно хотя бы в течение 2 часов. Он советовал также делить конспект на две графы: в первой кратко записывать изложенные лекции, а во второй – то, над чем надо подумать; сюда же следовало заносить узловые, главные вопросы, над которыми надо подумать постоянно, связывая с этим повседневное чтение. Он подчеркивал, что узловые вопросы предмета будут программой, на основе которой припоминается весь материал.

5. Использование компьютера в процессе самостоятельной работы студентов.

Наиболее комплексный ряд заданий, выполняемых студентом в процессе учебы в вузе, развивающих самостоятельность – это написание реферативных, курсовых и дипломных работ, выполнение которых требует применения всего спектра знаний, умений и навыков, приобретенных студентом в процессе обучения. Алгоритм, методика и формы выполнения этих работ практически одинаковы, они различаются содержанием и глубиной проработанности материала. И реферат, и курсовая, и дипломная работы должны выполняться в соответствии с действующими требованиями ГОСТов.

На современном этапе никто уже не представляет себе самостоятельную работу без использования международной информационной сети – **Интернет**. Необходимость использования Интернета возникает не только при подготовке к практическим и семинарским занятиям, но, в большей степени, при написании различных исследовательских и творческих работ. Многие современные монографии,

периодические журналы изданы только в электронном виде и с ними можно познакомиться только в Интернете.

Написание работ творческого и исследовательского характера требует знания и умения применять различные компьютерные технологии. Можно предложить следующий алгоритм работы по написанию исследовательских и творческих работ с использованием компьютера.

- Первый этап заключается в наборе материала на компьютере. Для этого необходимо, чтобы на компьютер были установлен текстовый и графический редакторы для набора текста и выполнения различных рисунков, графиков или схем. Если материал неоднородный, т.е. содержит графики, схемы, чертежи, текст, то для этих целей лучше выбрать интегрированный пакет, который позволяет совмещать различного формата файлы (например Word, PageMaker и др.). Цитаты из книг и журналов можно переснимать на сканере – удобно и быстро. Здесь как раз и понадобится база данных, которая значительно упростит работу с выбранной литературой.
- Второй этап – корректировка ошибок, недочетов. Практика показывает, что чтение с листа более привычно и корректировать удобнее файлы, имея распечатанный образец перед собой.
- Третий этап – печать начисто. Откорректированный и исправленный текст необходимо не забыть проверить на орфографию (по возможности и стилистику) перед тем, как распечатать. Чертежи лучше выводить на бумагу на графопостроителе.
- Четвертый этап – рецензия специалистов, работающих в данной области.
- Пятый этап – защита курсовых или дипломных работ на кафедре или в лаборатории. Желательно использовать презентационные компьютерные программы (например, Power point) при ответе – это увеличит наглядность доклада и использовать презентационные средства типа Proxima – проектор, позволяющий выводить на экран

содержимое дисплея. Можно также использовать телевизор вместо монитора при наличии специального блока сопряжения.

Почти на всех этапах студент работает самостоятельно. За время выполнения исследования у него развиваются:

1. Навыки и методы работы с литературой: ее анализ, отбор необходимого материала.
2. Навыки и методы работы с персональным компьютером: профессиональный набор текста, выполнение рисунков и чертежей, схем и др.
3. Исследовательские навыки и др.

Поиск в Интернете.

1. Поиск информации в Интернете лучше всего начинать с работы в **Интернет-каталоге**.

Один из наиболее полных и хорошо систематизированных каталогов в русскоязычном секторе Интернета находится на сайте www.aport.ru. Есть много других Интернет-каталогов: www.yandex.ru, www.list.ru, www.rambler.ru (русскоязычные), www.altavista.com (англоязычный) и др. Выбор каталога зависит от вкусов пользователя, степени проработанности его тематической структуры, скорости доступа к ресурсам каталога и т.д.

2. Чтобы попасть на эту страничку, вам надо вписать URL (адрес) данного сайта в адресную строку вашего Интернет-обозревателя (браузера), которая находится в верхней части окна.

3. Перед вами откроется главная страница поисковой системы, например «Апорт».

4. Находим на этой странице ссылку на подкаталог «Наука и образование» и кликаем на ней мышью. Теперь мы попадаем на следующую

страницу каталога, где пользователю предлагается выбрать интересующую его рубрику.

5. Ищем на этой странице ссылку на рубрики. Кликаем на нее. Загружается следующая страница, на которой будут ссылки на подрубрики. Под списком рубрик появятся ссылки на конкретные Интернет-ресурсы. Вы выбираете интересующий вас ресурс (при этом можно пользоваться краткой аннотацией, рейтингом популярности сайта, информацией о времени его последнего обновления) и кликаете на его ссылке. Откроется новое окно браузера, в которое будет загружен выбранный вами сайт.

Помимо **тематического поиска** в любом Интернет-каталоге есть **контекстный поиск**. Попробуем по Интернет-каталогу найти **ссылки** на газету «География».

1. Набираем в окне браузера адрес любого из русскоязычных каталогов.

2. В появившемся поисковом окне набираем целиком словосочетание, например «газета География», если система может работать со словосочетаниями, или слово «газета», а потом во втором поиске «география», если такой функции в системе не предусмотрено (www.aport.ru). Не забудьте только при втором поиске напротив поискового окна поставить галочку «Искать в найденном».

3. Через несколько секунд на экране вашего компьютера появятся первые десять или двадцать ссылок на Интернет-страницы, где поисковая система нашла указанные вами слова. При необходимости после просмотра первой порции ссылок на Интернет-ресурсы можно перейти к следующим. Ссылки на них вы найдете внизу экрана.

4. Для сохранения интересующих вас Интернет-страниц достаточно кликнуть мышкой на меню «файл» (оно находится в самой верхней части браузера) и выбрать пункт «сохранить как».

5. Появится диалоговое окно, где вам нужно будет указать папку для сохранения данной страницы, вписать имя, под которым она будет сохранена

(по умолчанию страница сохраняется под тем же именем, что она имеет в Интернете).

6. Если все указано правильно, смело нажимайте на «ОК», в противном случае выбирайте «Отмену».

Часто бывает так, что всю страницу сохранять необязательно, так как интерес вызывают лишь отдельные ее элементы. Текстовая часть страницы без графики и средств мультимедиа сохраняется как файл языка HTML. Часто имеет смысл сохранять только текст, так как любые графические объекты занимают много места на дискетах и жестких дисках компьютера.

Если вам необходимо сохранить только графические элементы страницы (рисунки, фотографии и т.д.), достаточно кликнуть на интересующей вас картинке правой клавишей мыши. Появится диалоговое окно, в котором следует выбрать пункт «Сохранить рисунок как».

Следует помнить, что вы не сможете редактировать составленные в формате HTML Интернет-страницы. В случае если вам по каким-то причинам нужно **внести** в них **правку**, дополнить своими материалами, включить в готовый текстовый документ, то нужно:

1. Выделить мышью в окне браузера необходимый фрагмент текста (весь текст выделяется после нажатия на команды меню «правка» и «выделить все»);
2. Копировать его (команда «копировать» на вкладке «Правка» / «edit»);
3. Вставить в нужный текстовый файл в программе Word (команда «Вставить» / «Paste»).

Вы также можете **распечатать** нужные вам страницы:

1. Одновременно нажмите клавиши ctrl и P или выберите команду «Печать» / «Print» на вкладке «Файл».
2. Если вам не нужно распечатывать всю страницу, то вы можете распечатать ее фрагмент. Для этого вы выделяете интересующий

участок страницы мышью, а в диалоговом окне печати, указывая диапазон печати, выберите пункт «Выделенный фрагмент».

3. Если вы хотите вернуться на предыдущую страницу, достаточно кликнуть мышкой кнопку «Назад», которая находится в левом верхнем углу окна вашего браузера. Обратный шаг можно сделать, нажав на стрелку «Вперед».

Для того чтобы в следующий раз точно попасть на нужную вам страницу Интернета, совсем не обязательно переписывать ее адрес, часто громоздкий и сложный. Достаточно всего лишь добавить ссылку на страницу в папке «Избранное» (она расположена вверху экрана, на рабочей панели браузера). Если вы хотите запомнить много страниц и к тому же систематизировать их, то направляйтесь на специальный сайт www.zakladki.ru, где вы сможете сохранить гиперссылку на любую Интернет-страницу. В этом случае вы сможете работать не только со ссылками, подобранными вами, но и другими пользователями (при условии, что доступ к ним не закрыт паролем).

Использование электронной почты

Интернет предоставляет еще одну уникальную возможность – вступить в переписку с другими пользователями глобальной компьютерной сети посредством **электронной почты** (e-mail).

В настоящий момент существует много серверов (такие, как www.mail.ru, www.hotmail.ru и др.), которые дают возможность завести бесплатный электронный почтовый ящик.

Чтобы пользоваться почтовым ящиком вам надо:

1. Зарегистрироваться на этих сайтах.
2. Внимательно изучите все условия пользования своим электронным ящиком. Помните, что многие бесплатные почтовые ящики

прекращают свое существование, если клиент к ним долгое время не обращается.

3. Не забывайте также и то, что сам электронный ящик находится у вас дома лишь виртуально. На самом деле он размещен на сервере обслуживающей вас компании, которая предоставляет вам определенный участок сервера (как правило, около 5 мб).
4. Чтобы не возникало проблем с приемом и пересылкой сообщений, не забывайте регулярно чистить свой электронный почтовый ящик, уничтожая старые сообщения или пересохраняя их на жестком диске вашего компьютера.

Чтобы пользоваться **электронной почтой**, вам необходимо уметь проделывать следующие операции:

1. Вызовите программу по отправке почтовых сообщений (например «Outlook express»).
2. Если вы хотите отправить электронное сообщение, необходимо кликнуть мышкой на кнопке «Создать сообщение» или «Написать письмо».
3. Появится окно вашего письма.
4. В строке «Кому» указывается адрес электронной почты получателя. Адрес обязательно должен включать символ @.
5. Наберите с клавиатуры компьютера электронный адрес получателя вашего сообщения, например geo@1september.ru – адрес электронного почтового ящика газеты «География».
6. В строке «Копия» указываются адреса электронной почты тех адресатов, кому вы хотите послать копию данного сообщения (поле необязательно к заполнению).
7. В строке «Тема» указывается то словосочетание, которое увидит получатель вашего сообщения еще до ознакомления с его содержанием. Часто в этой строке в кратком виде формулируют основную идею послания (например, «Привет!» или «Материал к докладу» и др.).

8. После того как заполнена шапка письма, можете переходить в основное окно, где непосредственно пишете текст вашего сообщения.

По электронной почте вы можете **переслать любой файл с вашего компьютера** любому адресату.

1. При создании письма вам следует кликнуть по виртуальной кнопке «Вложить» в верхней части экрана.
2. Появится диалоговое окно, которое позволит выбрать нужный файл (или файлы) для пересылки. Вы должны знать, что прикреплять к вашему электронному сообщению можно не только текстовые файлы, но и графические, установочные и вообще любые.
3. Когда письмо готово, еще раз проверьте, верно ли указан адрес, после чего нажимаете клавишу «Отправить». Ваше сообщение отправлено.
4. Если вы по ошибке указали несуществующий адрес или электронный ящик вашего адресата по каким-нибудь причинам не работает, то ваше сообщение через несколько минут вернется обратно с уведомлением о существующих проблемах пересылки.

Для прочтения той почты, которая приходит к вам, необходимо:

1. Открыть папку «Входящие» вашей почтовой программы. Кликните мышкой на ссылке «Входящие».
2. Перед вами появится окно с обратными адресами, темами и сроками прихода всех неудаленных писем, которые пришли на ваш электронный адрес.
3. Чтобы прочесть письмо, достаточно кликнуть на его заголовке. Перед вами откроется окно с содержанием данного письма.
4. Вы можете ответить на это письмо, кликнув по ссылке «ответить». В этом случае компьютер автоматически заполнит строку с адресом, вписав туда адрес, с которого вам пришло это письмо. Весь текст исходного письма отобразится в основном окне и станет доступен для редактирования и дополнения.

5. Чтобы сохранить на своем компьютере пришедший к вам вложенный файл (письма с вложенными файлами помечены символом скрепки), надо открыть это письмо. Если вложен графический файл, он будет открыт в основном окне сообщения. В том случае, если прикрепленный файл имеет какой-либо другой формат, то, нажав на скрепку, вы можете ознакомиться с именами и форматом приложенных файлов. Эти файлы можно открыть, кликнув на них мышью, а можно сохранить, не открывая, выбрав пункт «сохранить вложения» из нисходящего списка под изображением скрепки.

При необходимости вы можете **переслать пришедшее вам письмо** другому адресату по вашему выбору, для этого надо кликнуть правой клавишей мыши по ссылке на это сообщение в окне «Входящие» и выбрать из нисходящего списка команду «Переслать».

6. Подготовка к экзаменам и зачетам

Экзаменационная сессия – очень тяжелый период работы для студентов и ответственный труд для преподавателей. Главная задача экзаменов – проверка качества усвоения содержания дисциплины.

На основе такой проверки оценивается учебная работа не только студентов, но и преподавателей: по результатам экзаменов можно судить и о качестве всего учебного процесса. При подготовке к экзамену студенты повторяют материал курсов, которые они слушали и изучали в течение семестра, обобщают полученные знания, выделяют главное в предмете, воспроизводят общую картину для того, чтобы яснее понять связь между отдельными элементами дисциплины.

Экзаменам, как правило, предшествует сдача зачетов. К экзаменам допускаются только те студенты, которые сдали зачеты.

При подготовке к экзаменам основное направление дают программы курса и конспект, которые указывают, что в курсе наиболее важно. Основной материал должен прорабатываться по учебнику, поскольку конспекта недостаточно для изучения дисциплины. Учебник должен быть проработан в течение семестра, а перед экзаменом важно сосредоточить внимание на основных, наиболее сложных разделах. Подготовку по каждому разделу следует заканчивать восстановлением в памяти его краткого содержания в логической последовательности.

До экзамена обычно проводится консультация, но она не может возместить отсутствия систематической работы в течение семестра и помочь за несколько часов освоить материал, требующийся к экзамену. На консультации студент получает лишь ответы на трудные или оставшиеся неясными вопросы. Польза от консультации будет только в том случае, если студент до нее проработает весь материал. Надо учиться задавать вопросы, вырабатывать привычку пользоваться справочниками, энциклопедиями, а не быть на иждивении у преподавателей, который не всегда может тут же, «с ходу» назвать какой-либо факт, имя, событие.

На экзамене нужно показать не только знание предмета, но и умение логически связно построить устный ответ.

Получив билет, надо вдуматься в поставленные вопросы для того, чтобы правильно понять их. Нередко студент отвечает не на тот вопрос, который поставлен, или в простом вопросе ищет скрытого смысла. Не поняв вопроса и не обдумав план ответа, не следует начинать писать. Конспект своего ответа надо рассматривать как план краткого сообщения на данную тему и составлять ответ нужно кратко. При этом необходимо показать умение выражать мысль четко и доходчиво.

Отвечать нужно спокойно, четко, продуманно, без торопливости, придерживаясь записи своего ответа.

На экзаменах студент показывает не только свои знания, но и учится владеть собой. После ответа на билет могут следовать вопросы, которые

имеют целью выяснить понимание других разделов курса, не вошедших в билет. Как правило, на них можно ответить кратко, достаточно показать знание сути вопроса. Часто студенты при ответе на дополнительные вопросы проявляют поспешность: не поняв смысла того, что у них спрашивают, начинают отвечать и нередко говорят не по сути.

Студент должен знать, что на экзамене осуществляется не только контроль и выставляется оценка, но это еще и дополнительная возможность, систематизация знаний. Если говорить о сверхзадаче экзаменатора, то она состоит в уяснении не только и не столько того, что студент выучил, сколько того, чему он научился и что останется у него после экзамена, поскольку этот остаток будет характеризовать образовательный уровень студента.

Следует помнить, что необходимым условием правильного режима работы в период экзаменационной сессии является нормальный сон, поэтому подготовка к экзаменам не должна быть в ущерб сну. Установлено, что сильное эмоциональное напряжение во время экзаменов неблагоприятно отражается на нервной системе и многие студенты из-за волнений не спят ночи перед экзаменами. Обычно в сессию студенту не до болезни, так как весь организм озабочен одним - сдать экзамены. Но это еще не значит, что последствия неправильно организованного труда и чрезмерной занятости не скажутся потом. Поэтому каждый студент помнить о важности рационального распорядка рабочего дня и о своевременности снятия или уменьшения умственного напряжения.

Рекомендуемая литература

1. Борикина Л.В., Виноградова Н.А. Пишем реферат, доклад, выпускную квалификационную работу: Учеб. пособ. М., 2000.
2. Виноградова Н.А. Методические рекомендации по выполнению письменных работ. М., 1999.
3. Гарунов М.Г., Пидкасистый П.И. Самостоятельная работа студентов. М., 1978.
4. Демидова А.К. Пособие по русскому языку: научный стиль. Оформление научной работы. М., 1991.
5. Самостоятельная познавательная деятельность студента: методическое пособие / Сост. Е.В. Гребенникова. М., 2002.
6. Соловьева Н.Н. Основы подготовки к научной деятельности и оформление ее результатов (для студентов и аспирантов). М., 2000.
7. Соловьева Н.Н. Учусь общаться: читаю, слушаю, говорю, пишу: Учеб. пособ. по культуре речи и речевому общению. М., 1996.
8. Чернилевский Д.В. Дидактические технологии в высшей школе. М., 2002.
9. Загашев И.О., Заир — Бек С.И. Критическое мышление: технологии развития. Спб, 2003.- 284 с.