

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Агабекян Раиса Левоновна

Должность: ректор

Дата подписания: 18.06.2022 12:15:08

Уникальный программный ключ:
4237c7ccb9b9e111bba61f4fcd9201d015c4dbaa123ff774747307b9b9fbcb

**Негосударственное аккредитованное некоммерческое
частное образовательное учреждение высшего образования
«Академия маркетинга и социально-информационных технологий –
ИМСИТ»
(г. Краснодар)**

Институт информационных технологий и инноваций

Кафедра математики и вычислительной техники

Рассмотрено и одобрено на
заседании кафедры математики и
вычислительной техники Академии
ИМСИТ, протокол №8 от 27 марта 2021
года, зав. кафедрой МиВТ, доцент Н.С.
Нестерова

УТВЕРЖДАЮ
Проректор по учебной работе,
доцент Н.И. Севрюгина
29 марта 2021 г.

Б2.О.01(У)

**ПРОГРАММА УЧЕБНОЙ ПРАКТИКИ
Эксплуатационная ПРАКТИКА**

для обучающихся направления подготовки
09.03.04 Программная инженерия

Направленность (профиль) образовательной программы
«Разработка информационно-вычислительных систем»
Квалификация (степень выпускника)
Бакалавр

Краснодар
2021

Программа учебной практики для студентов направления 09.03.04 Программная инженерия/ сост. кандидат технических наук, доцент Нестерова Н.С., – Краснодар, ИМСИТ, 2021.

Программа составлена в соответствии с требованиями федерального государственного образовательного стандарта высшего образования (ФГОС ВО) по направлению подготовки 09.03.04 Программная инженерия, утвержденного приказом Министерства образования и науки Российской Федерации от « 19 » 09. 2017 г. № 920

Программа содержит методические указания по учебной: технологической (проектно-технологической) практике для студентов направления 09.03.04 Программная инженерия.

В ней изложены цели и задачи практики, порядок организации и контроля. Дан перечень компетенций, формируемых в процессе прохождения практики. Приведены необходимые учебные материалы с набором индивидуальных практических заданий.

Программа содержит указания по составлению и оформлению отчета, порядок аттестации по итогам практики.

Составитель, канд. техн. наук, доцент

Н С. Нестерова

Программа рассмотрена и рекомендована на заседании кафедры Математики и вычислительной техники от 27.03.2021 г., протокол №8

Зав. кафедрой математики и вычислительной
техники, канд. техн. наук, доцент

Н.С. Нестерова

Программа утверждена на заседании Научно-методического совета Академии ИМСИТ протокол № 7 от 29 марта 2021 г.

Председатель Научно-методического Совета Академии ИМСИТ,
профессор

Н.Н. Павелко

Согласовано:

Проректор по учебной работе,
доцент

Н.И. Севрюгина

Проректор по качеству образования,
доцент

К.В. Писаренко

Рецензенты:

Видовский Л.А., д.т.н., профессор, профессор кафедры информационных систем и программирования КубГТУ

Глебов О.В., директор АО «ЮГ-СИСТЕМА ПЛЮС»

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
1. Цель и задачи практики.....	4
2. Организация, руководство и контроль прохождения практики	6
3. Рабочие места и режим работы студентов	7
4. Компетенции, формируемые при прохождении практик	7
5. Содержание учебной практики.....	8
5.1 Содержание первого этапа пратики	8
5.2 Содержание второго этапа практики	8
6. Общие сведения о пакетах программ.....	9
6.1. Знакомство с MathCAD	9
6.2. Знакомство со средой графического программирования LabVIEW	18
7. Методические указания к практическим работам.....	25
7.1. Практическая работа 1.....	Ошибка! Закладка не определена. 5
7.2. Практическая работа 2.....	Ошибка! Закладка не определена. 1
7.3. Практическая работа 3	35
7.4. Практическая работа 4.....	41
7.5 Практическая работа 5	47
7.6 Практическая работа 6	48
8. Индивидуальные задания.....	Ошибка! Закладка не определена.
8. Фонд оценочных средств	39
8.1 Цель фонда оценочных средств	50
8.2 Задачи фонда оценочных средств	50
8.3 Функции фонда оценочных средств дисциплины	51
8.4 Список контрольных вопросов	51
8.5 Критерии оценки результатов учебной практики	52
9 Использование информационных технологий в период практики	52
10 Содержание отчета по учебной практике	52
11 Организация учебной практики для лиц с ограниченными возможностями здоровья	53
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	55
ПРИЛОЖЕНИЯ.....	56

ВВЕДЕНИЕ

Учебная практика является частью учебного процесса студентов направления подготовки бакалавров 09.03.04 «Программная инженерия» и предназначена для совершенствования профессиональной подготовки студентов.

Указанная практика отражает важную составляющую образовательной стратегии академии (ИМСИТ) – обеспечение неразрывности теоретического обучения и профессиональной практической подготовки будущих бакалавров и предоставляет возможности для формирования и оценивания их общекультурных и профессиональных компетенций.

Программа практики разработана в соответствии с требованиями федерального государственного образовательного стандарта высшего образования (ФГОС ВО) по направлению подготовки 09.03.04 Программная инженерия, утвержденного приказом Министерства образования и науки Российской Федерации от « 19 » 09. 2017 г. № 920

1. Цель и задачи учебной практики

Практика обеспечивает соответствие уровня теоретической подготовки практической направленности в системе обучения и будущей деятельности выпускника.

Цель практики:

- закрепление, расширение, углубление и систематизация знаний, полученных при изучении обязательных дисциплин базовой части учебного плана;
- подготовка к выполнению самостоятельных и курсовых работ в последующих семестрах;
- обеспечение возможности применения студентами теоретических знаний для решения практических задач;
- развитие организаторских способностей студентов;
- формирование общего представления студентов о будущей профессиональной деятельности и развитие интереса к профессии.

Учебная практика базируется на дисциплинах:

- вычислительная математика;
- математика;
- информатика;
- программирование;
- математическая логика и теория алгоритмов;
- дискретная математика;
- теория вероятностей и математическая статистика;
- системное программное обеспечение.

Основные задачи первого этапа учебной практики:

- ознакомление с математическими и статистическими пакетами программ;

- изучение основных приемов работы в пакете MathCAD;
- овладение навыками работы в указанном пакете для решения задач научно-исследовательского и производственного характера.

В результате овладения программой практики студент должен изучить основные принципы программирования в LabVIEW.

ЗНАТЬ:

- назначение и состав передней панели и блок-диаграммы программы LabVIEW;
- элементарные функции в LabVIEW
- справочные ресурсы LabVIEW;

ВЛАДЕТЬ:

- языком графического программирования G;
- методами моделирования в LabVIEW.

2. Организация, руководство и контроль прохождения практики

За время обучения студенты проходят в соответствии с учебным планом учебную и производственную практики (таблица 1).

Таблица 1 – Организация практик

Вид практики	курс	Семестр курса	Продолжительность недель
Учебная практика: эксплуатационная практика	2	2	42
Учебная практика: технологическая (проектно-технологическая) практика	3	2	4
Производственная практика: Эксплуатационная	4	1	2
Производственная практика: Технологическая (проектно-технологическая) практика	4	2	2
Производственная практика: Преддипломная практика	4	2	2

Учебная практика является одним из видов учебной работы, когда студент обязан выполнить практические и индивидуальные задания в компьютерном классе, подготовить и защитить отчет по практике.

Руководство учебной практикой осуществляет заведующий выпускающей кафедрой.

Руководитель учебной практики:

- составляет программу учебной практики;
- разрабатывает темы индивидуальных заданий;
- осуществляет методическое обеспечение практики;
- контролирует выполнение заданий и консультирует студентов.

Студент представляет руководителю практики от академии отчет, который защищает в комиссии, образованной из преподавателей выпускающей кафедры.

Отчет необходимо защитить в течение первой недели после окончания практики.

3. Рабочие места и режим работы студентов

Учебная практика осуществляется:

- в мультимедийных аудиториях академии при проведении лекций, установочных занятий и презентации научных докладов по темам заданий;
- в компьютерных классах при выполнении учебных заданий, в том числе экспериментальных исследований и обработке их результатов;
- в библиотеке академии для подбора материала по темам индивидуальных заданий

Допускается прохождение практики по индивидуальной программе в компьютерных фирмах, НИИ или КБ, использующих передовые информационные технологии

Студент, проходящий практику, обязан:

- знать и строго соблюдать правила внутреннего распорядка, охраны труда, техники безопасности и производственной санитарии;
- выполнять индивидуальные задания;
- вести учет выполненных работ и прослушанных лекций, бесед, экскурсий и т.д.;
- по окончании практики представить руководителю отчет о практике и защитить его в комиссии в установленные сроки.

До начала практики выпускающая кафедра должна организовать общее собрание студентов с изложением основных целей и задач практики и других организационных вопросов;

4. Компетенции, формируемые при прохождении учебной практики (ознакомительная)

В результате прохождения учебной практики студент бакалавриата должен приобрести следующие компетенции

:
.

Общепрофессиональные:

ОПК-1 Способен применять естественнонаучные и общеинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности;

ОПК-2 Способен использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности;

ОПК-3 Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности;

ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов;

ОПК-7 Способен применять в практической деятельности основные концепции, принципы, теории и факты, связанные с информатикой;

ОПК-8. Способен 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.

Профессиональные:

ПК-2 Владение навыками использования различных технологий разработки программного обеспечения

ПК-3 Владение концепциями и атрибутами качества программного обеспечения (надежности, безопасности, удобства использования), в том числе роли людей, процессов, методов, инструментов и технологий обеспечения качества

ПК-5 Владение навыками моделирования, анализа и использования формальных методов конструирования программного обеспечения

Соответствие выделенной частично (или полностью) ОТФ (обобщенной трудовой функции) профессиональной деятельности выпускников, на основе установленных профессиональным стандартом для ОТФ уровня квалификации и требований раздела ФГОС «Требования к образованию и обучению» в наборе профессиональных компетенций по дисциплине:

Наименование стандарта	Индекс ОТФ	Наименование ОТФ	Компетенции дисциплины	Требования к образованию установленные

Профессиональный стандарт «Программист», утвержденный приказом Министерства труда и социальной защиты Российской Федерации от 18 ноября 2013 г. № 679н (зарегистрирован Министерством юстиции Российской Федерации 18 декабря 2013 г., регистрационный № 30635), с изменением, внесенным приказом Министерства труда и социальной защиты Российской Федерации от 12 декабря 2016 г. № 727н (зарегистрирован Министерством юстиции Российской Федерации 13 января 2017 г., регистрационный № 45230)	D	Разработка требований и проектирование программного обеспечения	ПК-2 ПК-3 ПК-5	профстандартом "Высшее образование Повышение квалификации"
--	---	---	----------------------	--

5. Содержание учебной практики

Содержание учебной практики отражено в таблице 3.

Таблица 3 – Содержание учебной практики

№ п/п	Наименование вопросов (работ, заданий) подлежащих изучению в период практики	Количество академ. часов/зач. единиц
	Знакомство со средой графического программирования LabView. Многократные повторения и циклы	60
	Массивы. Принятие решений. Структуры Select и Case	30
	Создание подпрограмм виртуальных приборов. Кластеры	40
	Графическое отображение данных	40
	Выполнение индивидуального задания	46
	ИТОГО	216

6 Среда графического программирования LabVIEW

6.1 Знакомство со средой

LabVIEW или Laboratory Virtual Instrument Engineering Workbench (Среда разработки лабораторных виртуальных приборов) представляет собой мощную и гибкую среду графического программирования, которая широко используется в промышленности, образовании и научно-исследовательских лабораториях в качестве стандартного инструмента для сбора данных и управления приборами. LabVIEW - многоплатформенная среда, которая может быть установлена на компьютерах с операционными системами семейства Windows, MacOS или Linux.

Программы LabVIEW называются виртуальными приборами (ВП, VI – virtual instruments), так как они функционально и внешне подобны реальным приборам.

После запуска LabVIEW появляется стартовое диалоговое окно (рисунок 8.1), для выбора дальнейших действий. Окно разделено на две части с двумя группами вариантов действий пользователя. Левая часть окна позволяет создать новый ВП (Blank VI), пустой проект (Empty project) или создать ВП по шаблону (VI from Template). Можно также открыть недавние проекты и файлы (ссылки под надписью Open).

Правая часть диалогового окна позволяет получить доступ к справочным материалам (Help), материалам на сайте ni.com (Online Support), найти и запустить готовые примеры ВП (Find Example).

При создании пустого ВП (Blank VI) открываются два окна, содержащие лицевую панель (Front Panel) и блок-диаграмму (Block Diagram) (рисунок 6.2) виртуального прибора.

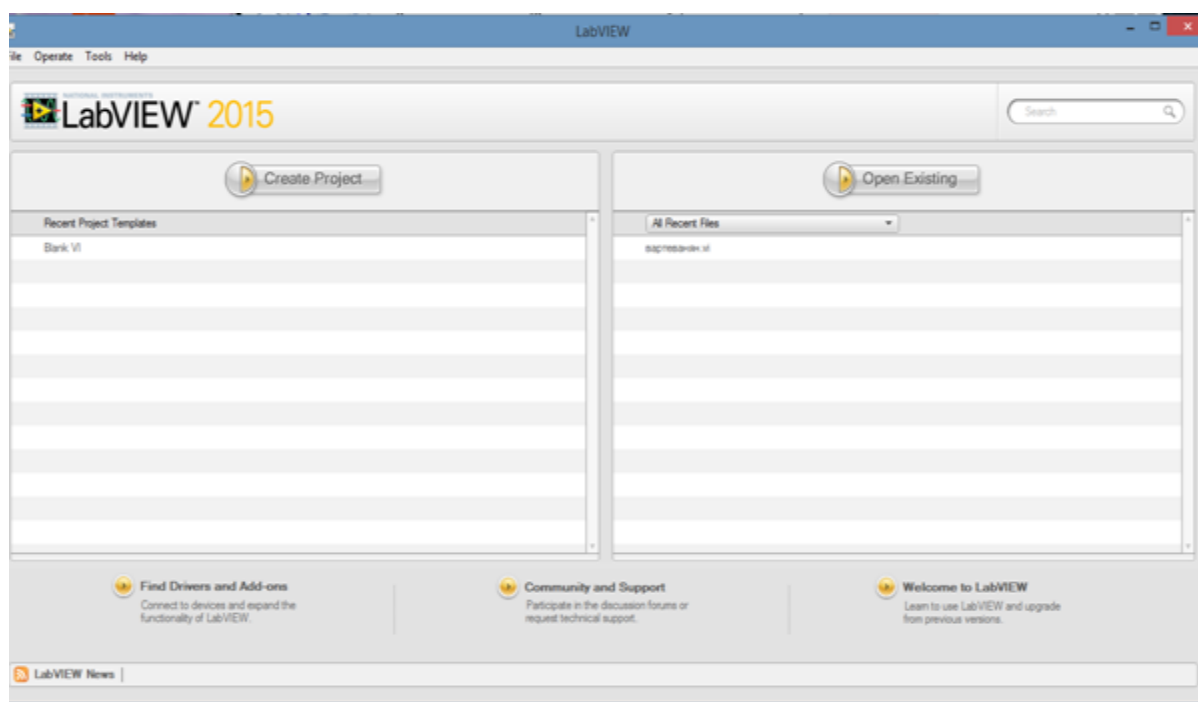


Рисунок 6. 1- Главное окно программы

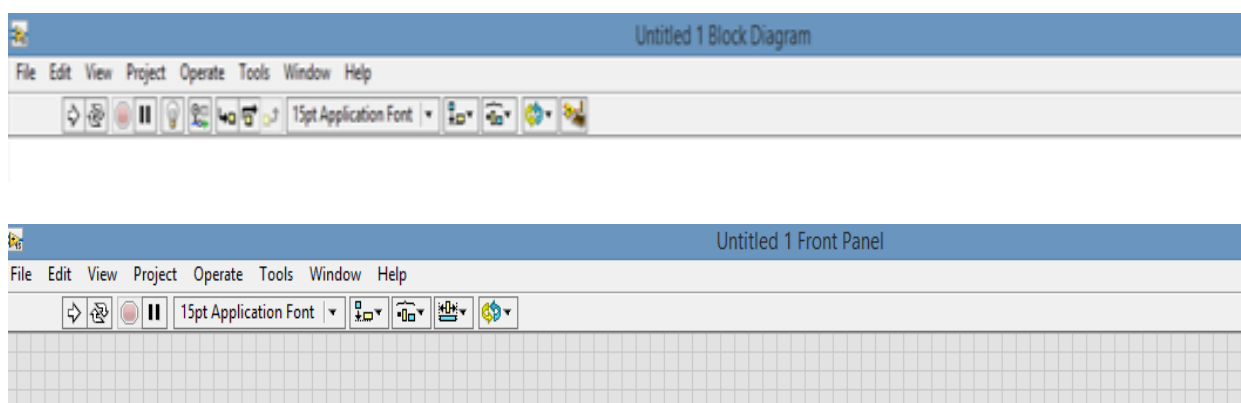


Рисунок 6.2- Лицевая панель и блок-диаграмма ВП

На лицевой панели разрабатывают внешний вид будущего виртуального прибора. На ней создают элементы управления и отображения, которые являются интерактивными средствами ввода и вывода данных этого виртуального прибора.

На блок-диаграмме разрабатывают исходный код будущего виртуального прибора. Исходный код LabVIEW представляет собой блок-диаграмму, где все команды, операторы циклов и сравнения изображают графическими обозначениями. Блок-диаграмма состоит из узлов, терминалов и проводников данных.

В верхней части каждого окна располагается главное меню с пунктами File, Edit, View, Project, Operate, Tools, Windows и Help. Под главным меню расположена инструментальная панель, служащая для запуска и редактирования ВП. Инструментальная панель окна блок-диаграммы отличается дополнительными инструментами для отладки ВП. В правом верхнем углу каждого окна находится иконка, наложенная на соединительную панель ВП. Окна блок-диаграммы и лицевой панели содержат инструментальную панель (рисунок 2).

В таблице 6.1 приведено краткое описание функций пунктов главного меню

Таблица 6.1 – Функции пунктов главного меню

Файл(File)	Используется для открытия новых или существующих ВП и проектов, закрытия, сохранения и вывода на печать ВП и проектов, а также для доступа к свойствам ВП
Правка (Edit).	Применяется для редактирования панелей ВП, поиска объектов и удаления неисправных проводников с блокдиаграммы, создания подприборов и установления значений элементов по умолчанию
Вид (View).	Используется для вывода доступных палитр, списка ошибок, иерархии ВП и взаимосвязей между ВП и подприборами, работы с проводником классов
Проект (Project).	Позволяет работать с проектом: создавать новый, открывать или сохранять существующий, добавлять элементы, получать информацию о файлах или устанавливать свойства проекта

Управление (Operate).	Реализует запуск и прерывание выполнения ВП, соединение с удаленной панелью и отладку приложения или общей библиотеки
Инструменты (Tools).	Используется для запуска программы анализа измерений и автоматизации (MAX) и поиска драйверов приборов, поиска ВП на диске, открытия окна MathScript, управления библиотеками ВП, управления соединением с удаленными ВП и публикацией панелей ВП в Web, конфигурирования ВП и для выполнения ряда прикладных функций
Окно (Window)	Используется для отображения окон LabVIEW и списка открытых ВП и проектов
Справка (Help).	Служит для получения информации об элементах и о функциях LabVIEW

Выпадающее меню **File** – Файл (рисунок 6.3) содержит опции:

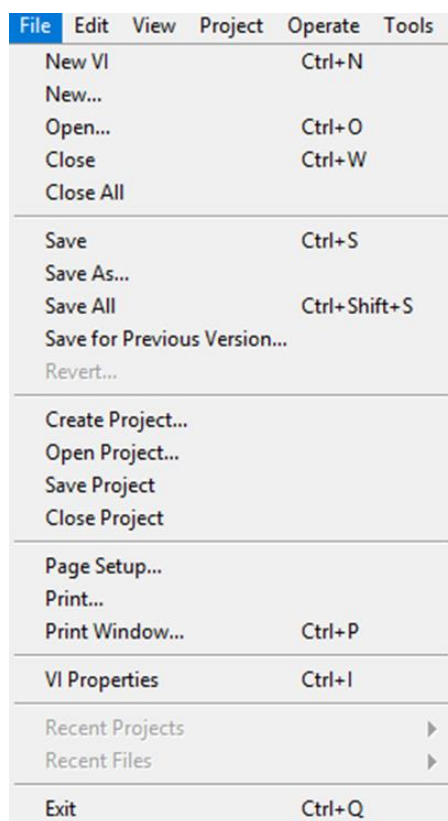


Рисунок 6.3 – Вкладка файл.

New IV - создание нового виртуального инструмента.

New - создание нового компонента: виртуального инструмента, глобальной переменной, элемента управления/индикатора, шаблона ВИ, шаблона глобальной переменной, шаблона элемента управления/ индикатора, полиморфного виртуального инструмента и др.

Open –открытие уже существующего виртуального инструмента.

Close – закрытие активного виртуального инструмента.

Close All – закрытие всех открытых приложений.

Save – сохранение текущего виртуального инструмента на диске.

Save As – сохранение виртуального инструмента с новым именем.

Save All – сохранение всех открытых приложений.

Save for Previous Version – запись виртуального инструмента в формате предыдущих версий LabView.

Revert – возвращение виртуального инструмента к последней сохраненной версии.

Create Project – создать проект.

Open Project – открыть проект.

Save Project – сохранить проект.

Close Project – закрыть проект.

Page Setup – установка опций печати на принтер.

Print – печать виртуального инструмента на принтер с возможностью выбора содержания.

Print Window - печать содержания текущего окна на принтер.

VI Properties - установка свойств виртуального инструмента.

Recent Projects – недавний проект.

Recent Files – недавние файлы.

Exit – ВЫХОД.

Выпадающее меню **Project** – проект (рисунок 6.4) содержит опции:

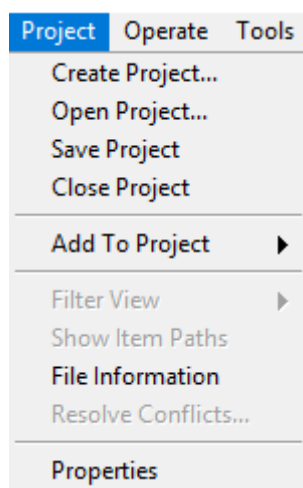


Рисунок 6.4 – Вкладка проект.

Create Project – создать проект.

Open Project – открыть проект.

Save Project – сохранить проект.

Close Project – закрыть проект.

Add To Project - добавить в проект.

Filter View - вид фильтра.

Show Item Paths - показать пути элемента.

File Information – данные о файле.

Resolve Conflicts – Решение конфликтов.

Properties – свойства.

Выпадающее меню **Operate** – действия (рисунок 6.5) содержит опции:

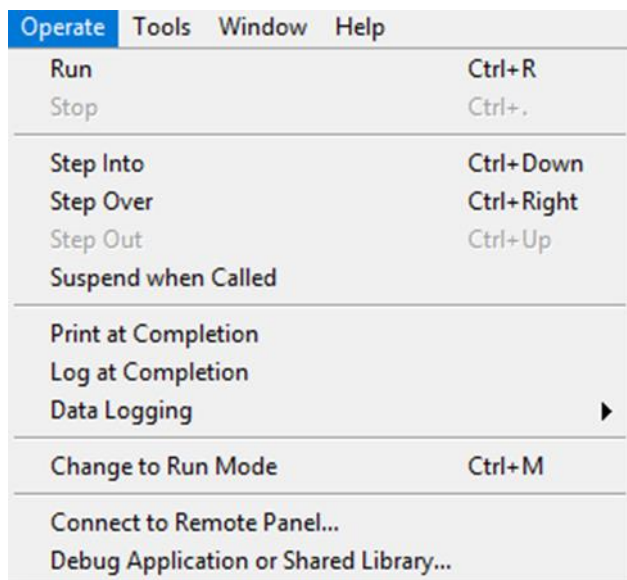


Рисунок 6.5- Вкладка действия.

Run - запуск программы на выполнение.

Stop - прекращение выполнения программы.

Step Into – шаг в.

Step Over – переступить.

Step Out – выйти.

Suspend when Called - пауза во время вызова виртуального инструмента.

Print at Completion - печать интерфейсной панели виртуального инструмента после завершения его работы.

Log at Completion - регистрация данных после завершения работы программы.

Data Logging - настройка регистрации данных.

Change to Run Mode - переключатель между режимами выполнения программы и редактирования.

Connect to Remote Panel – подключение к удаленной панели.

Debug Application or Shared library - отладка приложений или общей библиотеки.

Выпадающее меню **Edit** – редактировать (рисунок 8.6) содержит опции:

.

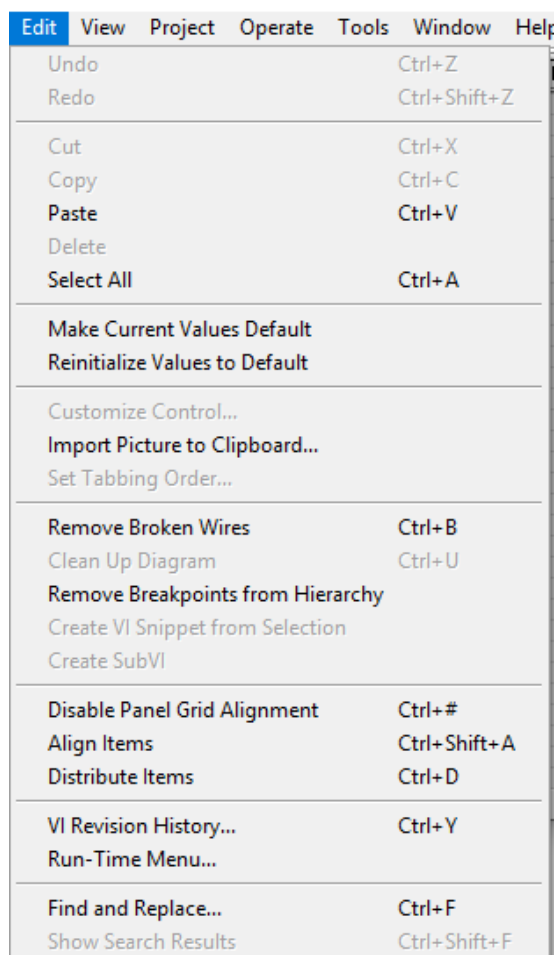


Рисунок 6.6 – Вкладка редактировать.

Undo - отмена действия.

Redo - "возвращение" отмененного действия.

Cut - удаление выделенного объекта и копирования его в буфер обмена.

Copy - копирование выделенного объекта в буфер обмена.

Paste - вставка объекта из буфера обмена.

Delete – удаление объекта.

Select All - выбрать всё.

Make Current Values Default - установление текущих значений элементов управления или индикаторов, как значений по-умолчанию.

Reinitialize Values to Default - установление всех значений элементов управления и индикаторов в значение по умолчанию.

Customize Control - открытие окна редактирования элементов управления и индикации.

Import Picture to Clipboard - импортировать картинку в буфер обмена.

Set Tabbing Order - установление последовательности активизации объектов интерфейсной панели во время выполнения программы при помощи клавиши Tab.

Remove Broken Wires – удаление незадействованных или неправильных связей на функциональной панели.

Clean Up Diagram - очистить диаграмму.

Remove Breakpoints from Hierarchy - удалить контрольные точки из иерархии.

Create VI Snippet from Selection - создать IV фрагмент выбора.

Create SubVi - создание подпрограммы из выбранных объектов на функциональной панели.

Disable Panel Grid Alignment - выравнивание по сетке.

Align Items - выровнять элементы.

Distribute Items – распространять предметы.

VI Revision History - История изменений.

Run-Time Menu - редактирование меню.

Find and Replace - найти и заменить.

Show Search Results - отображение результата поиска.

Вкладка **View** – Посмотреть (рисунок 6.7) содержит опции:

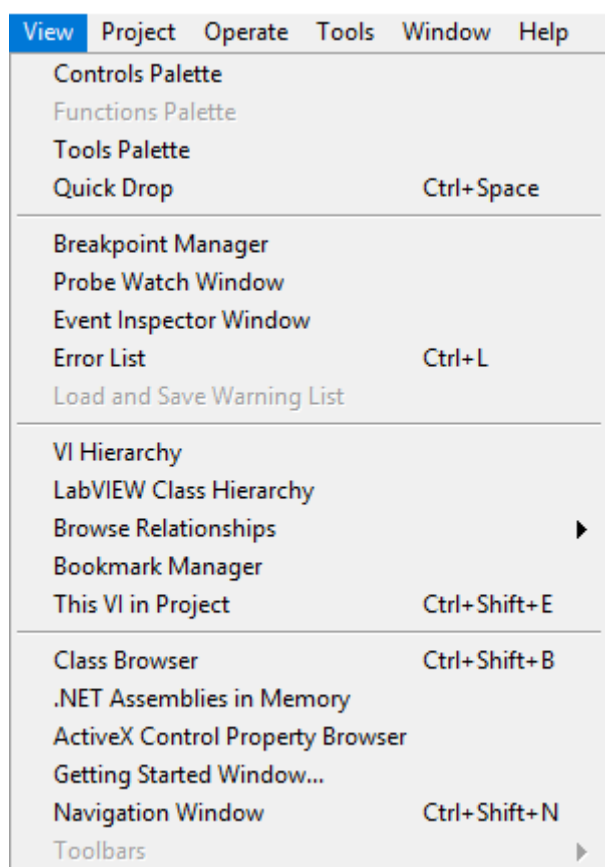


Рисунок 6.7 – Вкладка посмотреть.

Controls Palette – управление палитрой.

Functions Palette – функции палитрой.

Tools Palette – инструменты палитры.

Quick Drop – быстрое снижение.

Breakpoint Manager – менеджер контрольной точки.

Probe Watch Window – окно наблюдения.

Event Inspector Window – окно контролера событий.

Error List – список ошибок.

Load and Save Warning List - загрузить и сохранить список предупреждений.

VI Hierarchy – иерархия.

LabView Class Hierarchy – иерархия классов LabView.

Browse Relationships – просмотр взаимосвязей.

Bookmark manager – менеджер закладок.

This VI in Project – этот виртуальный инструмент в проекте.

Class Browser – класс программы для просмотра.

.NET Assemblies in Memory – сборка памяти.

ActiveX Control Property Browser – свойства программы.

Getting Started Window – окно начала работы.

Navigation Window – окно навигации.

Toolbars – панель инструментов.

Вкладка **Tools** – инструменты (рисунок 6.8) содержит опции:

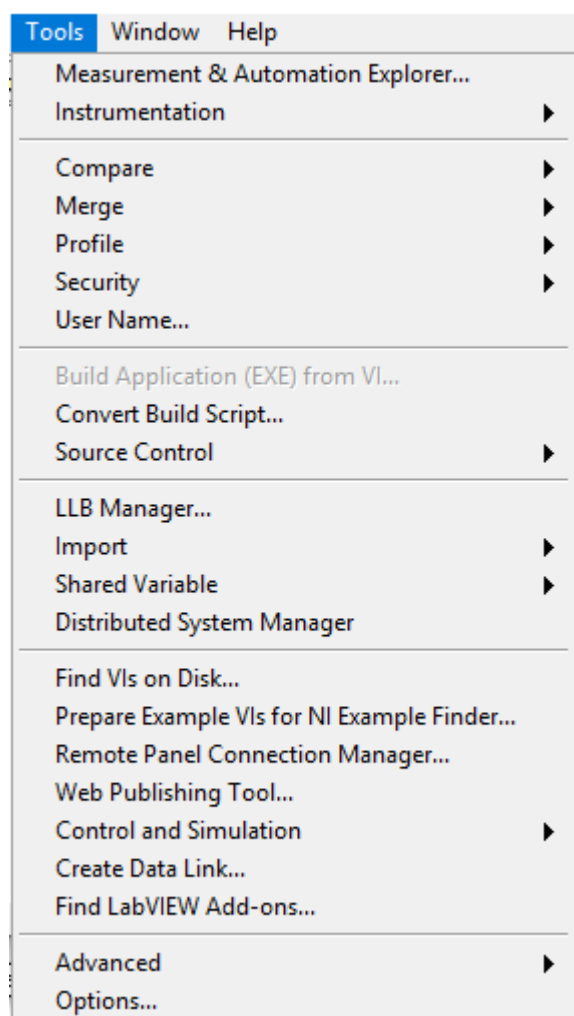


Рисунок 6.8 – Вкладка инструменты.

Measurement & Automation Explorer - запуск программ конфигурирования устройств DAQ, GPIB, IMAQ, IVI, Motion, VISA, VXI.

Instrumentation - возобновление VXI-драйверов и других функций.

Compare - сравнение виртуальных инструментов, их иерархий, сравнение файлов.

Merge – слияние виртуальных инструментов.

Profile – профиль.

Security – безопасность.

User Name - изменение имени пользователя.

Build Application (EXE) from VI - построить приложение из виртуального инструмента.

Convert Build Script - преобразование сценария сборки.

Source Control – управление источником.

LLB Manager – менеджер LLB.

Import – импортировать.

Shared Variable – общие переменные.

Distributed System Manager - менеджер по распределению системы.

Find Vis on Disk – найти виртуальные инструменты на диске.

Prepare Example Vis for NI Example Finder – подготовить пример виртуального инструмента к средству поиска NI в качестве примера.

Remote Panel Connection Manager - удаленный менеджер соединений панели.

Web Publishing Tools - инструменты веб-публикаций.

Control and Simulation - управление и моделирование.

Create Data Link – создать канал передачи данных.

Find LabView Add-ons – найти дополнения LabView.

Advanced - опция, которая позволяет включать в свои проекты компонент Active, осуществлять компиляцию, устанавливать метрики VI; позволяет импортировать и экспортировать текстовые файлы из проекта.

Options - опции виртуального инструмента.

Вкладка **Window** – окно (рисунок 6.9) содержит опции:

Window	Help
Show Front Panel	Ctrl+E
Show Project	
Tile Left and Right	Ctrl+T
Tile Up and Down	
Full Size	Ctrl+ /
1 Untitled 1 Block Diagram on Untitled Project 1/My Computer	
2 Untitled 1 Front Panel on Untitled Project 1/My Computer	
3 Untitled Project 1 * - Project Explorer	
All Windows...	Ctrl+Shift+W

Рисунок 6.9 – Вкладка окно.

Show Front Panel – показать фронтальную панель.

Show Project – показать проект.

Tile Left and Right - расположение основных панелей в левой и в правой сторонах.

Tile Up and Down - Tile Up and Down - расположение основных панелей сверху и снизу.

Full Size - увеличение активного окна до размеров экрана.

1 Untitled 1 Block Diagram on Untitled Project/My Computer и **2 Untitled 1 Front Panel on Untitled Project 1/My Computer** – перемещение окон друга на друга.

3 Untitled Project 1* - Project Explorer – развернуть данное окно.

All Windows – развернуть все окна.

Вкладка **Help** – помощь (рисунок 6.10) содержит опции:

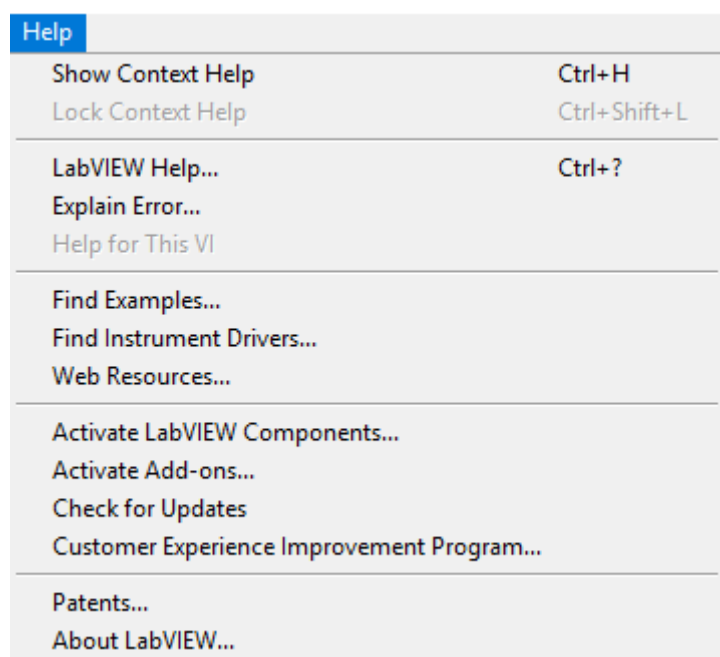


Рисунок 6.10 – Вкладка помощь.

Show Context Help - отображение контекстной подсказки.

Lock Context Help - закрепление контекстной подсказки.

LabVIEW Help – помощь LabView.

Explain Error - объяснение ошибок программы, если они существуют.

Find Examples – найти примеры.

Find Instrumental Drivers – найти инструментальные драйвера.

Web Resources – Веб-ресурсы.

Activate LabView Components – активация компонентов LabView.

Activate Add-ons – Активация надстроек.

Check for Updates – проверить наличие обновлений.

Customer Experience Improvement Program – программа улучшения клиентов.

Patents – патенты.

About LabVIEW - о среде разработки LabVIEW.

В таблице 6. 2 описаны функции кнопок инструментальных панелей, которые служат для отладки и управления ходом исполнения ВП.

Таблица 6.2 – Описание функций кнопок инструментальных панелей

Кнопка	Описание
	Кнопка Запуск (Run) работоспособного ВП
	Вид кнопки Запуск (Run) при наличии ошибок в блок-диаграмме ВП
	Вид кнопки Запуск (Run) ВП в процессе выполнения
	Вид кнопки Запуск (Run) в процессе выполнения подприбора
	Кнопка Непрерывный запуск (Run Continuously) вызывает непрерывный запуск ВП до момента нажатия кнопки Стоп (Stop) или Прервать (Abort)
	Кнопка Прервать выполнение (Abort Execution) вызывает остановку выполняющегося ВП
	Кнопка Пауза (Pause) временно останавливает выполнение ВП
 	Кнопка Подсветка выполнения (Highlight Execution) вызывает режим анимационного показа процесса передачи данных по блок-диаграмме и отображения значений данных на выходе узлов и терминалов
	Кнопка Сохранять (Не сохранять) значения провода (Retain (Do Not Retain) Wire Values) позволяет сохранить последнее значение, переданное по проводу, и просмотреть его после окончания выполнения при установке Пробника данных (Probe Data)
	Кнопка Начало пошагового выполнения (Start Single Stepping)
	Кнопка Выход из пошагового выполнения (Step Out) завершает пошаговое выполнение ВП

Построение ВП осуществляют с помощью трех вспомогательных палитр: палитры **Элементы управления (Controls Palette)**, палитры **Функции (Functions Palette)** и палитры **Инструменты (Tools Palett)**. Все перечисленные палитры можно вывести для постоянного или временного отображения и разместить в любом месте экрана. Вывод для постоянного отображения осуществляют с помощью разделов меню **Вид (View)**. Так, в частности, при активном окне лицевой панели с помощью строки **Палитра элементов управления (Controls Palette)** меню **Вид** на эту панель можно вывести палитру элементов, а при активном окне панели блок диаграммы на нее можно вывести палитру функций, пользуясь строкой **Палитра функций (Functions Palette)** этого же меню.

Для вывода палитры инструментов необходимо использовать строку **Палитра инструментов (Tools Palette)** меню **Вид**. Однако может оказаться, что пользователю более удобен временный вывод первых двух палитр, который реализуют как вызов контекстного меню каждой панели с помощью щелчка на ее рабочем пространстве правой кнопкой мыши (ПКМ). Выбор конкретного объекта из палитры элементов или палитры функций производят перемещая курсор мыши по разделам палитр. Объект выбирают из палитры посредством щелчка левой кнопкой мыши (ЛКМ) и переносят в заданную область соответствующей панели, после чего фиксируют здесь повторным щелчком ЛКМ (технология Перенес и бросил (**Drag and Drop**)). Эту же операцию можно выполнить с помощью щелчка ЛКМ на выбранном объекте, последующим удержанием клавиши во время переноса объекта и отпусканием клавиши в момент его фиксации. Такие объекты палитры функций, как **Структуры (Structures)** или строковые константы, перед фиксацией можно увеличивать до необходимых размеров рисованием модифицированным курсором мыши прямоугольного контура объекта при постоянно нажатой ЛКМ. Временную версию палитры инструментов можно вывести с помощью щелчка ПКМ при нажатой клавише **<Shift>**.

Все операции по созданию, редактированию и отладке ВП выполняются с помощью палитры Инструменты (**ToolsPalette**). При выборе определенного инструмента из палитры инструментов значок курсора мыши приобретает форму этого инструмента. При включенном автоматическом выборе инструмента наведение курсора на объект лицевой панели или блок диаграммы LabVIEW приводит к автоматическому выбору соответствующего инструмента из палитры инструментов. Автоматический выбор инструментов включают нажатием кнопки Автоматический выбор инструмента (**Automatic Tool Selection**) палитры инструментов или нажатием клавиш **<Shift+Tab>**.

Выбор любого другого инструмента приводит к отключению автоматического выбора инструмента. При этом можно циклически менять инструменты с помощью клавиши **<Tab>**. Для переключения между инструментами Перемещение и Соединение на блок диаграмме или между инструментами Перемещение и Управление на лицевой панели достаточно нажать пробел.

Инструмент Управление (**Operate Value, «палец»**) используют для изменения значений элементов управления или ввода текста. При работе со строковыми элементами управления вид инструмента изменяют на следующий:

Инструмент Перемещение (**Position/Size/Select, «стрелка»**) служит для выбора, перемещения или изменения размеров объектов. Для изменения размеров в LabVIEW используют подвижные прямоугольные элементы, появляющиеся в зависимости от допустимого направления изменения в центре сторон или на углах контура объекта при установке инструмента Перемещение внутри этого контура. Инструмент Редактирование текста (**Edit Text, «буква»**) используют для ввода и редактирования текста и создания свободных меток. При создании текстовых элементов вид инструмента изменяют: Инструмент Соединение (**Connect Wire, «катушка»**) применяют для соединения объектов на блок диаграмме. Его также используют для

условного (невидимого) подключения элементов управления и индикаторов лицевой панели к терминалам соединительной панели ВП. Инструмент Контекстное меню объекта (Object Shortcut Menu) вызывает контекстное меню.

Для создания программ в LabVIEW используют специальные всплывающие меню – палитры, которые могут автоматически отображаться при старте или быть скрыты. Для разных объектов внешний вид палитр может отличаться. При работе палитры помещают в любое удобное место окон. Рассмотрим основные палитры подробнее.

Палитра инструментов (Tools) (рисунок 6. 11)



Рисунок 6.11 - Палитра инструментов (Tools)

Элементы палитры инструментов используют для редактирования и управления функциями программы, собственно инструментом является специальный рабочий режим курсора манипулятора «мышка». Перейти в палитру инструментов можно посредством щелчка правой кнопки мыши(с удерживанием клавиши «shift»). Для того, чтобы выбрать нужный инструмент, необходимо щелкнуть мышкой по соответствующей кнопке палитры инструментов, и курсор мыши соответственно изменится. Если нет уверенности в правильности выбора инструмента, следует удерживать курсор на кнопке до появления подсказки. Назначение основных элементов палитры инструментов дано в таблице 3

Таблица 6. 3 - Основные рабочие инструменты LabView

-

Рабочий Инструмент	Оригинальное название	Русское название	Функции инструмента
--------------------	-----------------------	------------------	---------------------



Operating tool

Инструмент действия – “Рука

”

Размещает объекты палитр «**Controls**» - и «**Functions**» на лицевой панели и структурной схеме

	<i>Positioning tool</i>	Позиционный инструмент “Стрелка”	. Размещает объекты, изменяет их размеры и выбирает их.
	<i>Labeling tool</i>	Меточный инструмент	Редактирует тексты меток объектов и создает свободные метки
	<i>Wiring tool</i>	Монтажный инструмент “Катушка”	Подключает объекты друг к другу на структурной схеме.
	<i>Object pop-up menu tool</i>	Инструмент объектного меню	Вызывает объектное меню -
	<i>Scroll tool</i>	Инструмент прокрутки	Прокручивает окно без использования слайдеров
	<i>Breakpoint tool</i>	Инструмент контрольной точки	Устанавливает контрольные точки на функциях - , циклах, структурах
	<i>Probe tool</i>	Инструмент пробы	Создает пробные измерители на проводах.-
	<i>Color Copy tool</i>	Инструмент копии цвета – “Пипетка”	Копирует цвета для вставки с помощью Цветового инструмента -
	<i>Color tool</i>	Цветовой инструмент – “Кисть”	Устанавливает цвета переднего плана и фона -

Палитры элементов управления и функций

Для размещения элементов управления и отображения данных на лицевую панель используют палитру элементов управления и отображения (рисунок 8.12). Палитра элементов управления и отображения доступна только с лицевой панели. Для ее вывода на экран следует щелкнуть правой кнопкой мыши в рабочем пространстве лицевой панели.

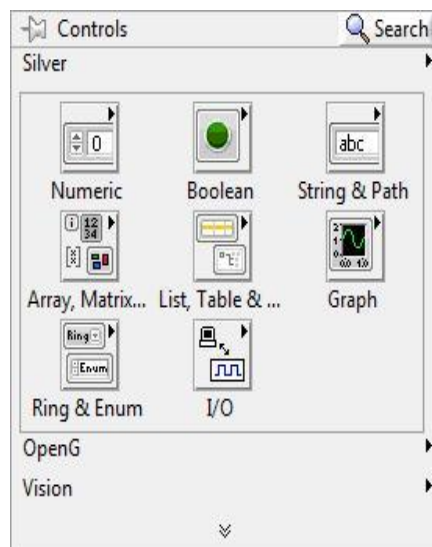


Рисунок 6.11 – Палитра элементов управления и отображения

Здесь:

- *Numeric (числовые значения)*. Состоит из элементов управления и элементов отображения для числовых данных;
- *Boolean (булевы значения)*. Состоит из элементов управления и элементов отображения для булевых величин;
- *String&Path (строковые значения и таблицы)*. Состоит из элементов управления и элементов отображения для ASCII строк и таблиц;
- *List & Table (списки и закольцованные списки)*. Состоит из элементов управления и элементов отображения для меню, выполненных в форме списков и закольцованных списков;
- *Array & Matrix (массивы и матрицы)*. Состоит из элементов управления и элементов отображения для группировки наборов типов данных;
- *Graph (виртуальные осциллографы)*. Состоит из элементов отображения для построения графиков данных в графах или диаграммах в реальном масштабе времени;

•

Все элементы управления и отображения на палитре сгруппированы по разделам. Каждый раздел может делиться, в свою очередь, на подразделы (числовые, строковые индикаторы, кнопки и т.д.).

На блок-диаграмме разрабатывают исходный текст программы, используя палитру функций (Functions palette). Для ее вывода на экран следует щелкнуть правой кнопкой мыши в рабочем пространстве блок-диаграммы. Все элементы на палитре сгруппированы по разделам (рисунок 68.5).

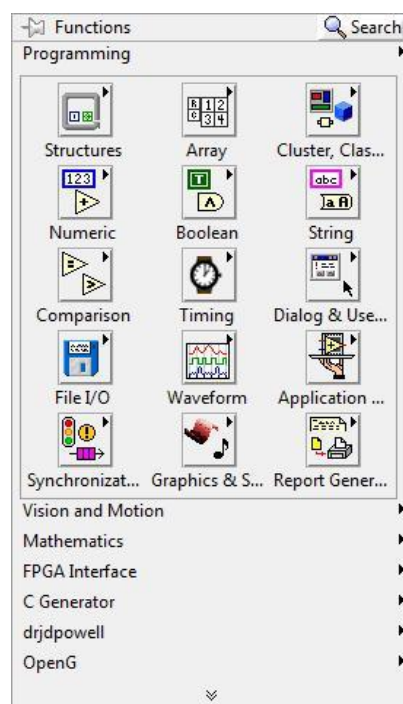


Рисунок 6. 12 – Палитра функций

Здесь

- *Structures (структуры)*. Состоит из управляющих структур программы, таких как циклы For Loop, While Loop и др.;
- *Numeric (числовые функции)*. Состоит из тригонометрических, логарифмических и других функций;
- *Boolean (булевы функции)*. Состоит из логических и булевых функций;
- *String (строковые функции)*. Состоит из функций для работы со строковыми величинами;
 - *Array (массивы)*. Состоит из функций для обработки массивов
- *Cluster (кластеры)*. Состоит из функций для обработки кластеров;
- *Comparison (сравнение)*. Состоит из функций для сравнения переменных;
- *Dialog (время и диалог)*. Состоит из функций для диалоговых окон, синхронизации и обработки ошибок;
- *Timing (время)*
- *File I/O (ввода/вывода файла)*. Состоит из функций для осуществления операций по вводу/выводу файлов;
- *Waveform (форма сигнала)*

Для проектирования ВП в среде LabVIEW необходимо сформировать его лицевую панель и разработать блок-диаграмму.

После помещения элементов управления или отображения данных на лицевую панель они получают свое графическое отображение (в виде терминала данных) на блок-диаграмме. Символы на терминале соответствуют типу данных терминала. Например:

DBL – терминал представляет данные в виде вещественных чисел с двойной точностью,

TF – логический терминал,

I16 – терминал 16-битных целых и др.

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

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

При формировании лицевой панели выбирают и устанавливают на ней элементы управления и индикаторы из палитры элементов данной панели. Аналогично при разработке блок-диаграммы выбирают и устанавливают на ней функциональные элементы и подприборы из палитры функций данной панели.

Установка каждого элемента на лицевой панели сопровождается появлением соответствующего терминала данных (terminal) на панели блок-диаграммы. Терминалы элементов управления представляют собой порты ввода информации в блок-диаграмму, а терминалы индикаторов – порты вывода информации из блок-диаграммы на лицевую панель. Для обработки введенной информации и программного управления параметрами и режимами работы элементов лицевой панели на панели блок-диаграммы размещают необходимые константы, функции (Functions), подприборы (SubVI) и структуры (Structures), также имеющие терминалы для ввода и вывода информации. Все перечисленные элементы представляют узлы (nodes) блок-диаграммы, которые соединяются с терминалами элементов управления и индикации и между собой линиями - проводниками (wires). В такой схеме через узлы в процессе обработки проходят данные, распространяющиеся по проводникам от входных терминалов к выходным. Узлы – это объекты на блок-диаграмме, которые имеют одно или более полей ввода/вывода данных и выполняют алгоритмические операции ВП. Они аналогичны операторам, функциям и подпрограммам текстовых языков программирования.

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**». 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 Front 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**».

Создание, копирование и удаление объектов

В LabVIEW предусмотрена возможность создания элементов управления и отображения данных, констант по щелчку правой кнопкой мыши (ПКМ) на узле или предполагаемом месте их расположения. Для этого в контекстном меню следует выбрать пункт **Create → Constant** для создания констант, отображающихся только на блок-диаграмме; **Control** - для создания элемента управления на лицевой панели и блок-диаграмме; **Indicator** - для создания элемента отображения данных на блок-диаграмме и лицевой панели.

Объекты можно копировать, перемещая выделенный объект и одновременно удерживая клавишу Ctrl. После переноса выбранного объекта на новое место отпускается сначала кнопка мыши, а затем клавиша Ctrl. В этом месте появляется копия объекта, а первоначальный объект остается на старом месте. Можно копировать объекты стандартным способом, выбирая пункты главного меню **Edit → Copy** затем **Edit → Paste**.

Для удаления объекта следует выделить его с помощью инструмента активации объекта (палитра Tools) затем нажать на клавиатуре клавишу **Delete** или выбрать пункты главного меню **Edit → Clear**.

Отмена и восстановление действий

В процессе редактирования виртуального прибора может быть допущена ошибка. Можно отменить и восстановить действия, выбрав **Undo** (Отменить) или **Redo** (Восстановить) в пункте главного меню **Edit** (Редактирование). Установка количества действий, подлежащих отмене или восстановлению, производится в пункте главного меню **Tools** → **Options**. Для этого из выпадающего меню следует выбрать раздел **Block Diagram**. Установка небольшого числа повторений сохраняет ресурсы памяти компьютера.

Идентификация объектов

Для идентификации объектов используются метки. Среда LabVIEW имеет два вида меток - свободные и собственные. Собственные метки принадлежат объекту, описывают только его и двигаются вместе с ним. Собственную метку можно перемещать независимо от объекта, но при перемещении самого объекта его собственная метка перемещается вместе с ним. Свободные метки не принадлежат объектам. Их можно создавать, перемещать, вращать или удалять независимо. Они используются для описания объектов на русском языке, ввода комментариев на лицевой панели и блок-диаграмме.

Для создания свободной метки используется инструмент «Редактирование» на палитре Tools. Выбрав его, необходимо щелкнуть на свободном пространстве одной панели и ввести текст. После ввода текста метки поместить курсор в пространство вне метки или нажать кнопку **Enter** на инструментальной панели.

Выделение и удаление проводников данных

Сегмент проводника данных - это отдельная его горизонтальная или вертикальная часть. Место соединения двух сегментов - излом проводника данных. Точка, в которой встречаются два, три или четыре проводника данных, называется точкой соединения. Проводник данных содержит все сегменты между точками соединения и между терминалами данных, если нет точек соединений. Для выделения сегмента используется инструмент активизации на палитре **Tools**. Один щелчок мыши по выбранному сегменту проводника выделяет этот сегмент, двойной выделяет излом проводника данных, тройной щелчок – все проводники, соединяющиеся с выделенным сегментом. Разорванный проводник данных выглядит как черная штриховая линия с красным крестом посередине. Разрыв образуется при попытке соединения объектов с несовместимыми с ними типами данных. Описание причины разрыва проводника данных появляется в окне всплывающей подсказки после наведения на проводник инструмента «Соединение». Удаление всех разорванных проводников производится через пункт главного меню **Edit** → **Remove Broken Wires**.

Редактирование текста

Выбрав пункт меню **Text Setting** на инструментальной панели, можно изменить шрифт, стиль, размер и провести выравнивание любого текста внутри меток или на дисплеях элементов управления и отображения.

Изменение размеров объектов

Большинство объектов лицевой панели допускают изменение размеров. Чтобы подготовить объект к изменению размера, необходимо навести на него инструмент активизации. При этом по углам объекта появляются маркеры. Следует установить курсор на один из маркеров и, удерживая нажатой левую кнопку мыши, переместить маркер. Размер шрифта при этом не меняется. Промежуточные границы изменяемого размера обозначаются штриховой линией. Когда нужный размер элемента достигнут, кнопку мыши следует отпустить. Удержание клавиши **Shift** во время перемещения маркеров сохраняет пропорции объекта. Можно изменять размеры и объектов блок-диаграммы, таких как структуры и константы.

6.2 Методические указания к практическим работам

Практическая работа 1

Создание простого вычислительного устройства

1. Откройте новый виртуальный прибор *File>>New VI*.
2. На лицевой панели (*Front Panel*) разместите два управляющих элемента для ввода двух чисел *A* и *B*. Из подпалитры «*Numeric Controls*» палитры «*Controls*» выберите цифровой элемент ввода чисел «*Numeric Controls*». Разместите элемент на поле лицевой панели и в появившейся метке введите название переменных *A* или *B*.
3. На лицевой панели создайте четыре цифровых индикатора для вывода результатов вычисления. Из подпалитры «*Numeric Controls*» палитры «*Controls*» выберите цифровой элемент ввода чисел «*Numeric Indicator*». Разместите элемент на поле лицевой панели и в появившейся метке введите название «*A+B*», «*A-B*», «*A*B*», «*A/B*».

Вид лицевой панели дан на рисунке 6.13.

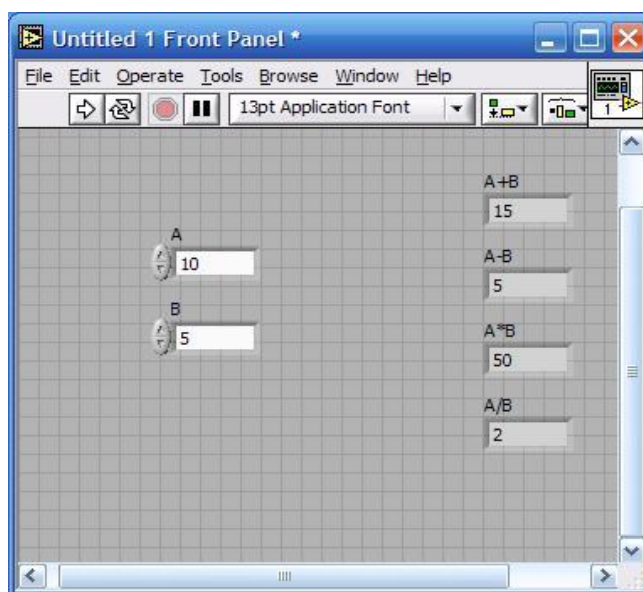


Рисунок 6.13 – Лицевая панель

4. 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

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

6. Выберите функции вычислительного устройства из палитры «*Functions*» >> «*Numeric*» >> «*Express Numeric*»: сложение (*Add*), вычитание (*Subtract*), умножение (*Multiply*) и деление (*Divide*) (рисунок 6.14).

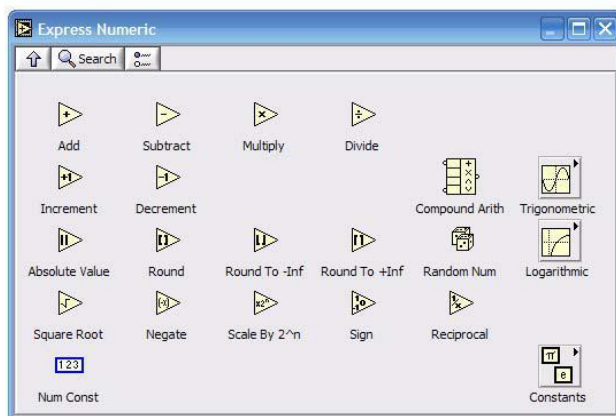


Рисунок 6.14 – Функции вычислительного прибора

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

На этом монтаж структурной схемы закончен. Структурная схема прибора имеет вид, показанный на рисунке 6.15.

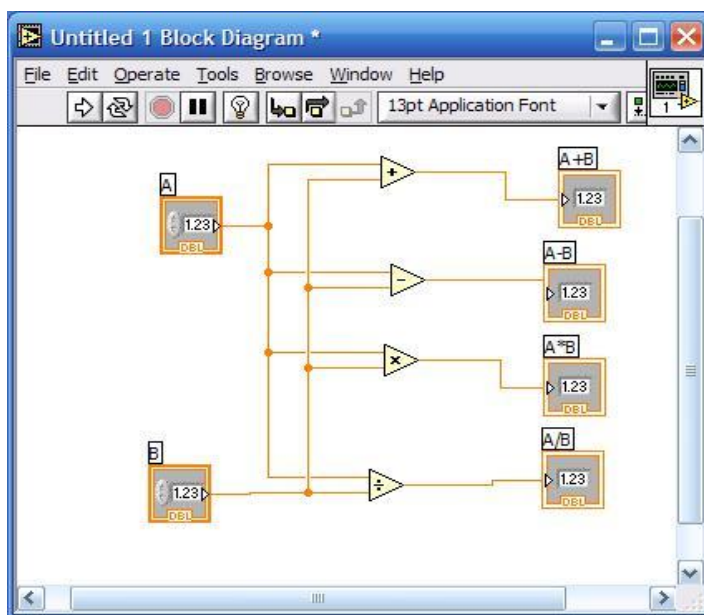


Рисунок 6.15 – Структурная схема прибора

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

9. Напишите отчет по работе.

Содержание отчета

- Титульный лист.
- Цель работы.
- Изображения лицевой панели прибора и структурной схемы.
- Выводы по работе.

Практическая работа 2

Разработка генератора синусоидального напряжения

Требуется разработать генератор синусоидального напряжения, с возможностью:

- регулирования частоты в пределах $0 \div 100$ Гц;
- регулирования выходного напряжения в пределах $0 \div 5$ В;
- измерения амплитудного и действующего значений выходного напряжения.

С этой целью необходимо выполнить следующее:

1. на лицевой панели разместите две ручки управления (Controls>Numeric>Knob) с надписями «Амплитуда» и «Частота»;
2. задайте максимальные значения шкалы амплитуды и частоты «5» и «100» соответственно. Для этого щелкните правой кнопкой мыши по соответствующей регулировочной ,ручке зайдите в контекстное меню Properties>Scale>Maximum и задайте требуемые значения;
3. установите стрелочный измерительный прибор (Controls>Numeric>Meter), выполните надпись «Напряжение», выберите шкалу «5»;
4. установите графический индикатор (Controls>Graph>Waveform Graph).

После всех проделанных операций лицевая панель прибора будет иметь вид, изображенный на рисунке 8.16.

5. на блок-диаграмме разместите:

источник синусоидального напряжения (Functions>Signal Processing>Wfm Generation>Simulate Sig). Задайте амплитуду сигнала Signal Amplitude 5. С помощью мышки растяните нижнюю границу иконки Simulate Sig

- ВП для измерения амплитуды и уровня сигнала (Functions>Signal Processing>Wfm Measure>Amp&Level). В меню ВП отметьте галочкой RMS (Root Mean Square) – действующее значение напряжения.

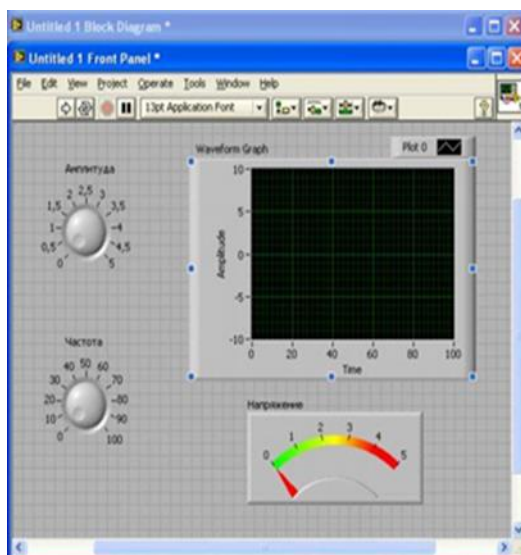


Рисунок 6.16 – Лицевая панель

6. Используя палитру инструментов, выполните необходимые соединения. Панель блок-диаграмм после выполнения всех указанных операций будет иметь вид, изображенный на рисунке 6.17. Активизируйте лицевую панель.

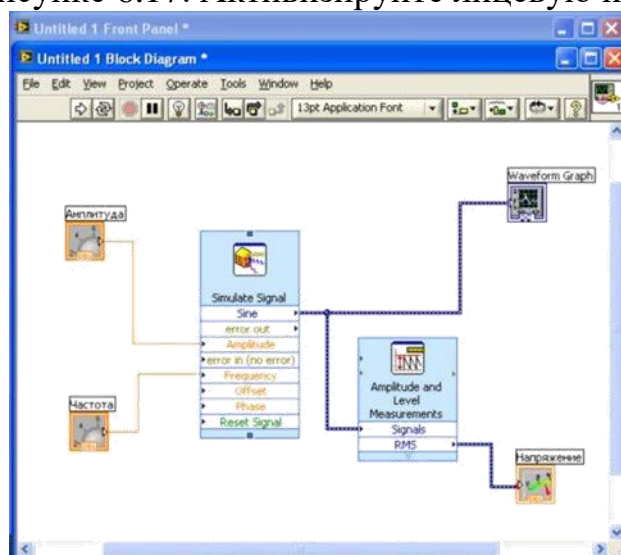


Рисунок 6.17 – Блок схема.

7. В палитре инструментов перейдите в режим Управление и, используя курсор мыши, ручками Амплитуда и Частота установите 2 В и 50 Гц соответственно.

8. Запустите программу, запишите в отчет амплитудное и действующее значения выходного напряжения генератора.

9. Установите амплитуду и частоту 5 В и 100 Гц и вновь запустите программу, запишите результаты измерений.

10. В отчете приведите изображение лицевых панелей и блок-диаграмм разработанных виртуальных приборов и показатели их работы.

Практическая работа 3 Моделирование работы базовых элементов цифровой техники

Цель работы: изучение и моделирование работы простейших базовых логических элементов в среде LabView; ознакомление с типом данных **Boolean**; создание и использование библиотеки подпрограмм.

Общие сведения

Логические элементы – это базовые блоки цифровых логических схем. Они могут открываться или закрываться, позволяя или отказывая пропускать логический сигнал. На основе небольшого количества основных логических элементов (И, ИЛИ, исключающее ИЛИ, НЕ) может быть построено огромное количество логических функций.

Логический элемент И. Базовый логический элемент И состоит из двух входов и выхода. Два входа назовем соответственно *A* и *B*. Выход назовем *Q*). Выход находится в состоянии «включено» только тогда, когда оба входа *A* и *B* находятся в состоянии «включено» (таблица 6.4).

Таблица 6.4 –таблица истинности

<i>A</i>	<i>B</i>	$Q=A \text{ И } B$
0	0	0
0	1	0
1	0	0
1	1	1

В цифровой электронике состояние «включено» обычно представляется в виде 1, а состояние «выключено» в виде 0. Соотношение между входными и выходными сигналами представляют в виде таблицы истинности, в которой сопоставляются все возможные состояния входов и результирующих выходов. Для логического элемента И существуют четыре возможные комбинации входного состояния: $A=0, B=0$; $A=0, B=1$; $A=1, B=0$ и $A=1, B=1$. Эти значения представлены в следующей таблице истинности в левом и среднем столбцах. Выход логического элемента И отображен в правом столбце.

В среде LabView можно определять состояние логического входа переключением логического выключателя, а логический светодиодный индикатор может показывать состояние выхода. Поскольку в среде LabView элемент И является одной из основных встроенных функций, то можно легко создать простейший виртуальный прибор, демонстрирующий работу этого логического элемента, присоединив два выключателя к его входу и светодиодный индикатор к его выходу (рисунок 8.18).

1. Создайте виртуальный прибор, моделирующий работу элемента И. Используйте для передней панели прибора элементы из группы «Boolean», для структурной схемы элемент «AND» из группы «Boolean». Нажимая на две входные кнопки, наблюдайте изменения выходного индикатора. Проверьте таблицу истинности, показанную выше.

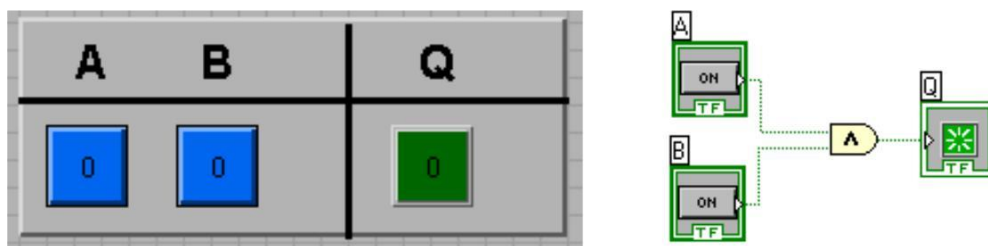


Рисунок 6.18 - Функция LabView «И», присоединенная к входным и выходному терминалам

Аналогичным образом может быть промоделирована работа элементов И, ИЛИ, НЕ, ИСКЛЮЧАЮЩЕЕ ИЛИ и т. д.

В пакете LabView содержатся все основные двухвходовые логические элементы, но можно использовать и больше входов. Из двух двухвходовых элементов можно построить виртуальный прибор, реализующий элемент И с тремя входами (рисунок 6.19)..

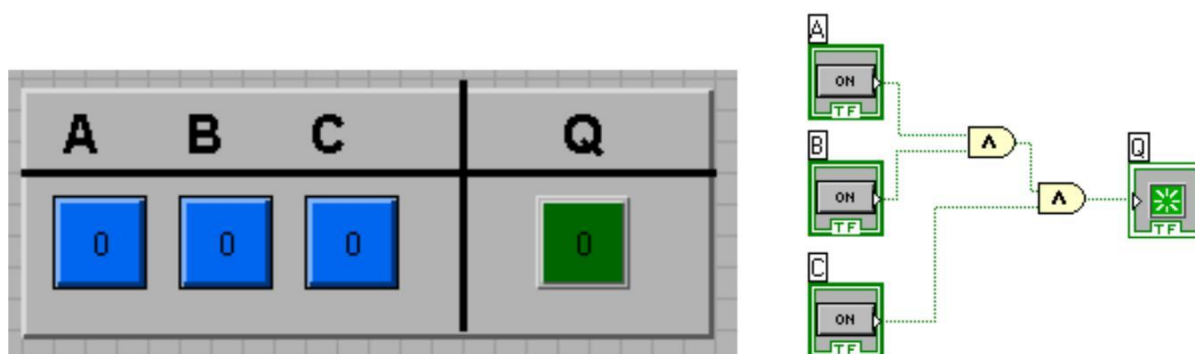


Рисунок 6.19 - Модель логического элемента И с тремя входами

Подпрограммы. Любая программа (VI, виртуальный прибор) может быть использована в блок-схеме другой программы как ее составная часть. Другими словами, она может быть, вложена как подпрограмма, *SubVI*. Эта особенность позволяет основной, главной программе быть модульной, быть легче читаемой и быть проще для понимания

Для вставки ранее разработанного VI (*SubVI*) в программу более высокого уровня необходимо использовать опции «*Select a VI...*» в палитре «*Function*» (рисунок 8.20). В ответ на запрос диалогового окна необходимо выбрать файл подпрограммы и установить ее на диаграмме основной программы.



Рисунок 6.20 - Палитра «Function»

Главная программа может иметь множество вызовов подпрограммы. На диаграмме главной программы подпрограмма появится в виде кубика со значком, заданным по умолчанию. С помощью двойного щелчка можно открыть эту подпрограмму, а при необходимости провести некоторые настройки.

Ранее была создана программа, моделирующая работу логического элемента И. Теперь создадим из этой программы полнофункциональную подпрограмму. Чтобы подпрограмма получала данные из основной программы и передавала их назад в основную необходимо сделать определенные действия.

Сначала следует открыть лицевую панель разработанной подпрограммы, и, разместив курсор в правом верхнем углу, на иконке, вызвать контекстное меню (рисунок 6.21).

Выбрав в нем пункт «Edit Icon», можно отредактировать внешний вид иконки создаваемой подпрограммы (рисунок 6.22).

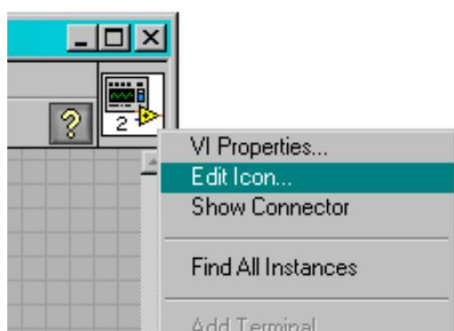


Рисунок 6.21 Выбор режима

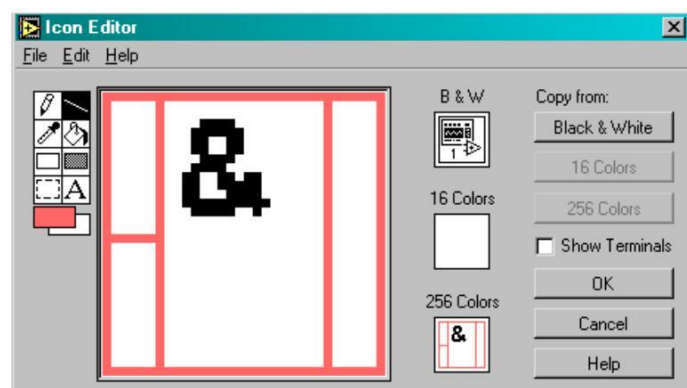


Рисунок 6.22 -. Редактор изображения иконки подпрограммы

Далее следует снова вызвать контекстное меню и выбрать пункт «Show Connector». При этом открывается доступ к параметрам подпрограммы – коннек-

торам или соединителям, элементам взаимодействия подпрограммы с внешним миром. Коннектор передает и принимает данные от вызывающей программы. Сразу после выбора этого пункта вид иконки программы изменится, а курсор приобретет вид соединительной катушки.

LabView позаботится о том, чтобы появились коннекторы – квадратики и прямоугольник – по числу элементов управления и индикации на лицевой панели прибора. Теперь необходимо установить соответствия между коннектором и элементом управления или индикации на лицевой панели. Для этого необходимо указать курсором-катушкой элемент на лицевой панели и щелкнуть по коннектору-квадратику, за которым он будет закреплен. Признаком подсоединения будет изменение цвета этих квадратиков. В данном примере квадратики слева – это логические сигналы *A* и *B*, прямоугольник справа – результат логического умножения *Q*.

Теперь подпрограмма, после ее установки в главную программу, даст возможность видеть входные и выходные переменные

При очень большом количестве подпрограмм их целесообразно объединить в библиотеки, файлы с расширением «*LLB*». Это можно сделать на этапе сохранения программы. В диалоге сохранения необходимо выбрать кнопку «*New VI Library*», указав после нажатия этой кнопки имя новой библиотеки, а далее сохранить текущий файл в созданной библиотеке.

Для обслуживания библиотек может быть использован пункт меню *LabView* «*LabView VI library manager*».

Выполнение работы

Задание 1.

Создайте виртуальный прибор, демонстрирующий работу основных логических элементов, лицевая панель прибора должна быть исполнена так, как показано на рисунке 6.23.

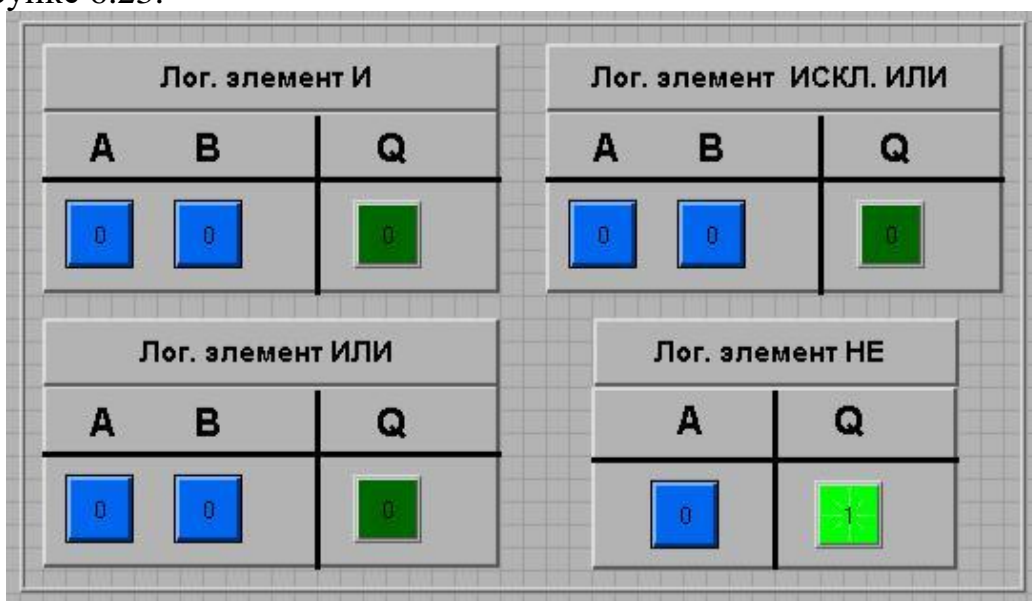


Рисунок 6.23 - Возможный вид лицевой панели разрабатываемого прибора

Для отчета сохраните снимок экрана (*screenshot*) лицевой панели прибора и структурной схемы прибора. Разработанный виртуальный прибор сохраните на своем носителе информации для его демонстрации при защите отчета.

Задание 2.

Оформите созданные виртуальные приборы – И, ИЛИ, НЕ, ИСКЛ. ИЛИ в виде подпрограмм и сохраните их в библиотеке с именем «*lab_2_library.llb*».

Задание 3.

Реализуйте логическую функцию: _____

$$Y = (\overline{A \cdot B}) \cdot (A \cdot B) .$$

При ее реализации использовать только *SubVI*, созданные в ходе выполнения задания 2. Лицевая панель прибора должна содержать два переключателя и один элемент индикации типа «*Boolean*».

Для отчета сохраните снимок экрана (*screenshot*) лицевой панели прибора и структурной схемы прибора.

Содержание отчета

1. Титульный лист.
2. Цель работы.
3. Виртуальный прибор, демонстрирующий работу логических элементов И, ИЛИ, НЕ, ИСКЛ. ИЛИ; лицевая панель данного виртуального прибора и структурная схема.
4. Виртуальный прибор, реализующий логическую функцию задания 3, его лицевая панель и структурная схема.
5. Выводы по работе.

6.3 Контрольные вопросы и задания

1. В данной работе все виртуальные приборы оперируют с типом данных «*Boolean*», что это за тип данных?
2. Зачем нужно создавать подпрограммы (*SubVI*)? Какие преимущества они дают?
3. Какие из базовых логических функций уже реализованы в LabView?
4. Зачем нужны библиотеки подпрограмм? Можно ли без них обойтись?
5. Сколько раз основная программа может иметь в своем теле вызовов подпрограмм?
6. Может ли основная программа при вызове подпрограммы передавать туда какие-либо данные и получать их назад?

7. На структурных схемах приборов в данной лабораторной работе видны соединительные провода зеленого цвета, что он обозначает в LabView?

7. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

7.1 Цель фонда оценочных средств

Основная **цель фонда оценочных средств** практики сделать контроль ее прохождения студентами постоянным и систематическим элементом учебного процесса.

7.2 Задачи фонда оценочных средств

- обеспечить контроль качества прохождения практики;
 - организовать работу в целях обеспечения лично мотивированной ответственности студента за результаты практики;
 - обеспечить контроль знаний студентов, что позволит оценить уровень приобретенных знаний, умений и сформированных компетенций обучающихся и степень их готовности к практической и научной деятельности;
 - сформировать у студентов навыки академической и научно-исследовательской работы, специфические для уровня обучения в магистратуре;
- диагностировать степень готовности студентов к тем видам деятельности, которые предусмотрены в ФГОС ВПО и ООП магистратуры

7.3 Функции фонда оценочных средств дисциплины

- воспитывающая: фонд оценочных средств способствует развитию компетенции обучающихся, приобретению совокупности знаний, умений, свойств и качеств личности, необходимых для выполнения научной деятельности;
- управляющая: фонд оценочных средств обеспечивает контроль степени усвоения знаний и применения их в профессиональной и научно-исследовательской деятельности и, прежде всего, при выполнении магистерской диссертации;
- обучающая: фонд оценочных средств помогает студентам рационально планировать процесс изучения дисциплины, последовательно выполняя про грамм практики;
- стимулирующе-мотивационная: фонд оценочных средств способствует созданию положительного мотивационного фона работы студентов, обеспечивает рост интереса к профессиональной и научной деятельности;
- контролирующая: фонд оценочных средств призван проверить планомерность, систематичность работы студентов, определить качество прохождения практики;
- корректирующая: фонд оценочных средств сигнализирует о необходимости использования мер воздействия на студентов с целью внесения корректив в ход и содержание процесса практики.

7.4 Критерии оценки результатов учебной практики

«**Отлично**» - студент предоставил отчет по практике, полностью отражающий цели и задачи практики, с результатами выполненных работ, формулировкой выводов и рекомендаций. В процессе защиты отчета студент правильно и четко ответил на поставленные комиссией вопросы и решил задачи.

«**Хорошо**» - студент предоставил отчет по практике, полностью отражающий цели и задачи практики, с результатами выполненных работ, формулировкой выводов и рекомендаций. В процессе защиты ответил не на все поставленные вопросы либо отвечал неточно.

«**Удовлетворительно**» - студент предоставил отчет по практике, не полностью отражающий цели и задачи и результаты практики. В процессе защиты затруднялся ответить на большинство вопросов. Имеются замечания по содержанию и оформлению отчета.

8 ИСПОЛЬЗОВАНИЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ В ПЕРИОД ПРАКТИКИ

В период практики рекомендуется использовать пакет LABVIEW.

11. СОДЕРЖАНИЕ ОТЧЕТА ПО УЧЕБНОЙ ПРАКТИКЕ

Отчет по практике должен состоять из следующих разделов и подразделов:

1. Титульный лист (приложение А)
2. Задание на практику (приложение Б)
3. Отчет по работе 1
4. Отчет по работе 2
5. Отчет по работе 3
6. Анализ функциональных возможностей LABVIEW
7. Результаты выполненного индивидуального задания в среде. LABVIEW

Оформление текстового материала должно отвечать стандартным требованиям

9 ОРГАНИЗАЦИЯ УЧЕБНОЙ ПРАКТИКИ ДЛЯ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ

Для студентов из числа лиц с ограниченными возможностями здоровья практика проводится Академией с учетом особенностей их психофизического развития, их индивидуальных возможностей и состояния здоровья .

При проведении практики обеспечивается соблюдение следующих общих требований:

- проведение практики для лиц с ограниченными возможностями здоровья в одной аудитории совместно со студентами, не имеющими ограниченных возможностей здоровья, если это не создает трудностей для них в процессе обучения;

- присутствие в аудитории ассистента, оказывающего обучающимся необходимую техническую помощь с учетом их индивидуальных особенностей (занять рабочее место, передвигаться, прочитать и оформить задание, общаться с преподавателем);
- пользование необходимыми обучающимся техническими средствами при выполнении практических и других работ в соответствии с учебным планом с учетом их индивидуальных особенностей;

В зависимости от индивидуальных особенностей обучающихся с ограниченными возможностями здоровья образовательная среда Академии обеспечивает выполнение следующих требований при организации учебной практики:

а) для слепых:

- задания и иные материалы для аттестации зачитываются ассистентом;
- письменные задания надиктовываются обучающимся ассистенту;

б) для слабовидящих:

- задания и иные учебно-методические материалы оформляются увеличенным шрифтом;
- обеспечивается индивидуальное равномерное освещение не менее 300 люкс;
- при необходимости обучающимся предоставляется увеличивающее устройство, допускается использование увеличивающих устройств, имеющихся у обучающихся;

в) для глухих и слабослышащих, с тяжелыми нарушениями речи:

- обеспечивается наличие звукоусиливающей аппаратуры коллективного пользования, при необходимости обучающимся предоставляется звукоусиливающая аппаратура индивидуального пользования;
- по их желанию аттестационные испытания проводятся в письменной форме;

г) для лиц с нарушениями опорно-двигательного аппарата (тяжелыми нарушениями двигательных функций верхних конечностей или отсутствием верхних конечностей):

- письменные задания надиктовываются ассистенту;
- по их желанию все аттестационные испытания проводятся в устной форме.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

-

1. Лабораторная работа Работа с циклическими структурами в labview [Электронный ресурс] .-Режим доступа:
http://www.meteoinfo12.ru/files/lab_rab_2_labview.pdf
2. Система LabVIEW[Электронный ресурс] .-Режим доступа:
http://www.studfiles.ru/_preview/3564161/
3. NI LabVIEW для образования [Электронный ресурс] .-Режим доступа:
<http://russia.ni.com/labview/applications/academic>
- 4 Лабораторная работа Введение в LABVIEW[Электронный ресурс] .-Режим доступа: <https://mipt.ru/dppe/upload/136/08-arpfqaxchop.pdf>

ПРИЛОЖЕНИЕ А

Образец титульного листа отчета по учебной практике

Негосударственное аккредитованное некоммерческое
частное образовательное учреждение высшего образования
«Академия маркетинга и социально-информационных технологий – ИМСИТ»
(г. Краснодар)
Институт информационных технологий и инноваций
Кафедра математики и вычислительной техники

Отчет по учебной практике: технологической (проектно-технологической)
в Академии маркетинга и информационных технологий (ИМСИТ) г. Краснодар

Направление 09.03.04 «Программная инженерия»

Отчет выполнил
студент 2 курса,
группы 11-ИВТ-01
ф.и.о.

Руководитель практики _____ ф.и.о.
« ____ » _____ 20 ____ г .

Отчет защищен с оценкой зачтено с оценкой _____
« ____ » _____ 2021 г.

Краснодар
2021

ПРИЛОЖЕНИЕ Б

Негосударственное аккредитованное некоммерческое
частное образовательное учреждение высшего образования
«Академия маркетинга и социально-информационных технологий – ИМСИТ»
(г. Краснодар)
Институт информационных технологий и инноваций
Кафедра математики и вычислительной техники

Утверждаю
Заведующий кафедрой

« ____ » _____ 20__ г.

ЗАДАНИЕ

на учебную практику: технологическую (проектно-технологическую)

Студенту _____ курса группы _____ ф.и.о. _____

Основные вопросы, подлежащие разработке:

- Знакомство с пакетом MatCad
- Выполнение практической работы 1
- Выполнение практической работы 2
- Выполнение практической работы 2
- Выполнение практической работы 2
- Выполнение индивидуального задания

Срок представления отчета « ____ » _____ 202__ г.

Дата выдачи задания: « ____ » _____ 202__ г.

Руководитель _____ / _____ /

Задание получил: « » 202 г.

Студент

ПРИЛОЖЕНИЕ В

Образец календарного плана прохождения учебной практики: технологической
(проектно-технологической)

Календарный план прохождения учебной практики

Студентом _____

№ п/п	Наименование работ и индивидуальных заданий	Период выполнения работ и заданий
1	2	3
1		
2		
3		
4		

Студент

(подпись,)

Руководитель практики от академии:

(подпись)

Руководитель практики от организации
(печать)

(подпись,

ПРИЛОЖЕНИЕ Г

Образец дневника практики

ДНЕВНИК ПРОХОЖДЕНИЯ УЧЕБНОЙ ПРАКТИКИ: ТЕХНОЛОГИЧЕСКОЙ (ПРОЕКТНО- ТЕХНОЛОГИЧЕСКОЙ)

(фамилия, имя, отчество)

Обучающегося курса, группы 1

Направление подготовки (специальность), направленность (профиль)
образовательной программы (специализация): 09.03.04 «Программная
инженерия»

Место прохождения практики:

Сроки практики: с _____ 202 г. по _____ 202 г.

Руководитель практики от организации

Дата (период)	Содержание проведенной работы	Результат работы	Оценки, замечания и предложения по работе

Студент _____

« » _____ 202 г.

Руководитель практики от академии_

« » _____ 202 г.

Руководитель практики от организации _____

« » _____ 202 г.

ПРИЛОЖЕНИЕ Д

Образец индивидуального задания на практику

Негосударственное аккредитованное некоммерческое
частное образовательное учреждение высшего образования
«Академия маркетинга и социально-информационных технологий – ИМСИТ»
(г. Краснодар)

Институт информационных технологий и инноваций

Кафедра математики и вычислительной техники

Индивидуальное задание, выполняемое в период проведения учебной практики

Направление подготовки 09.03.04 Программная инженерия
(профиль)

«Разработка информационно-вычислительных систем

Студенту _____

Сроки прохождения практики

с «_____» _____ 202 г. по с «_____» _____ 202 г.

Цель учебной практики (Ознакомительная, в соответствии с основной профессиональной образовательной программой по направлению подготовки 09.03.04 Программная инженерия, направленность (профиль) образовательной программы «Разработка информационно-вычислительных систем»: закрепление, расширение, углубление и систематизация знаний, полученных при изучении обязательных дисциплин обязательной части учебного плана; подготовка к выполнению самостоятельных и курсовых работ в последующих семестрах; обеспечение возможности применения студентами теоретических знаний для решения практических задач; развитие организаторских способностей студентов; формирование общего представления студентов о будущей профессиональной деятельности и развитие интереса к профессии., регламентированы ФГОС ВО по направлению подготовки 09.03.04 Программная инженерия, утвержденного приказом Министерства образования и науки Российской Федерации от « 19 » 09. 2017 г. №920

ОПК-1; ОПК-2; ОПК-3; ОПК-6; ОПК-7; ОПК-8; ПК-5; ПК-2; ПК-3

Перечень вопросов (заданий, поручений) для прохождения учебной практики:

№п/п	Этапы работы (виды деятельности) при прохождении практики	Код формируемых компетенций	Сроки	Отметка руководителя от академии
1	Организация практики, подготовительный этап, включающий инструктаж по технике безопасности	ОПК-1; ОПК-2; ОПК-		

№п/п	Этапы работы (виды деятельности) при прохождении практики	Код формируемых компетенций	Сроки	Отметка руководителя от академии
2	Содержательный этап: - ознакомление с техническими средствами и программным обеспечением - выполнение практических работ (№1-№3) и индивидуального задания в среде LabView	ОПК-1; ОПК-2; ОПК-3; ОПК-6; ОПК-7; ОПК-8 ПК-5		
3	Выполнение индивидуального задания на тему: «Язык программирования Python как средство анализа данных»	ОПК-1; ОПК-2; ОПК-3; ОПК-6; ОПК-7; ОПК-8 ПК-5		
4	Отчетный этап: - оформление отчета и дневника практики	; ПК-2; ПК-3		

Ознакомлен _____
« » _____ 202 г.

Руководитель практики от академии _____
« » _____ 202 г.

Руководитель практики от организации _____
« » _____ 202 г.

ПРИЛОЖЕНИЕ Ж

Образец отзыва руководителя на учебную практику

Негосударственное аккредитованное некоммерческое частное
образовательное учреждение высшего образования
«Академия маркетинга и социально-информационных технологий – ИМСИТ»
(г. Краснодар)

Институт информационных технологий и инноваций

Кафедра математики и вычислительной техники

ОТЗЫВ РУКОВОДИТЕЛЯ НА УЧЕБНУЮ ПРАКТИКУ СТУДЕНТА

**Направление подготовки 09.03.04 Программная инженерия
(профиль)**

«Разработка информационно-вычислительных систем

Студент _____

Наименование предприятия (базы практики):

Сформированность компетенций у выпускника по итогам выполнения заданий на практике

Этапы работы (виды деятельности) при прохождении практики	Код формируемых компетенций	Уровень сформированности компетенций*
<i>Подготовительный этап:</i> - инструктаж по безопасности жизнедеятельности	ОПК-1; ОПК-2; ОПК-	
<i>Содержательный этап:</i> - ознакомление с техническими средствами и программным обеспечением - выполнение практических работ (№1-№3) и индивидуального задания в среде LabView	ОПК-1; ОПК-2; ОПК-3; ОПК-6; ОПК-7; ОПК-8 ПК-5	
<i>Выполнение индивидуального задания:</i> - выполнение индивидуального задания на тему: «Язык программирования Python как средство анализа данных»	ОПК-1; ОПК-2; ОПК-3; ОПК-6; ОПК-7; ОПК-8 ПК-5	
<i>Отчетный этап:</i> - оформление отчета	; ПК-2; ПК-3	

*Отметить «Нулевой», «Низкий», «Средний», «Высокий»

Соответствие отчета по практике требованиям

Наименование требования	Заключение о соответствии требованиям*
1. Качество подобранного материала для проведения исследования	
1.1 Наличие источников информации в соответствии с заданием	
1.2 Наличие актуальных первичных данных, материалов	
2. Качественная оценка проведенного исследования собранных материалов	
2.1 Оценка требований к содержательной части отчета, соответствие заданию	
2.2 Оценка степени самостоятельности проведенного исследования	
2.3 Оценка качества проведенного исследования собранных материалов, данных	
3. Выполнение общих требований к проведению практики	
3.1 Выполнение требований руководителя по своевременному выполнению задания	
3.2 Выполнение требований к оформлению отчета по практике	

*Отметить «соответствует», «соответствует не в полной мере», или «не соответствует».

Достоинства содержательной части отчета по практике:

Ошибки и недостатки содержательной части отчета по практике:

Отчет защищен с оценкой зачтено с оценкой

« » _____ 202 г.

Руководитель практики от академии С.

« » _____ 202 г.

