

Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Факультет компьютерного проектирования

Кафедра проектирования информационно-компьютерных систем

Г. А. Пискун

**ПРОГРАММНО-ТЕХНИЧЕСКИЕ СРЕДСТВА
МНОГОПРОФИЛЬНЫХ СИСТЕМ.
ЛАБОРАТОРНЫЙ ПРАКТИКУМ**

*Рекомендовано УМО по образованию в области информатики
и радиоэлектроники в качестве учебно-методического пособия
для специальности 6-05-0713-02 «Электронные системы и технологии»*

Минск БГУИР 2024

УДК 004.42:004.946(076.5)
ББК 32.973я73
ПЗ4

Рецензенты:

кафедра программного обеспечения информационных систем и технологий
Белорусского национального технического университета
(протокол № 7 от 16.02.2024);

генеральный директор ОАО «Планар» доктор технических наук,
доцент С. М. Аваков

Пискун, Г. А.

ПЗ4 Программно-технические средства многопрофильных систем. Лабораторный практикум : учеб.-метод. пособие / Г. А. Пискун. – Минск : БГУИР, 2024. – 199 с. : ил.
ISBN 978-985-543-762-9.

Содержит восемь лабораторных работ, в которых рассмотрены основные возможности среды разработки виртуальных приборов *LabVIEW*, а также приведены подходы к проектированию систем диспетчерского управления и сбора данных в программном пакете *MasterSCADA*.

Предназначено для студентов специальности 6-05-0713-02 «Электронные системы и технологии» и специалистов, занимающихся проектированием многопрофильных систем.

**УДК 004.42:004.946(076.5)
ББК 32.973я73**

ISBN 978-985-543-762-9

© Пискун Г. А., 2024
© УО «Белорусский государственный университет информатики и радиоэлектроники», 2024

СОДЕРЖАНИЕ

Введение	4
Лабораторная работа № 1. Основы работы в среде <i>LabVIEW</i> : разработка виртуальных приборов, исследование функций и построение сложных кривых.....	6
Лабораторная работа № 2. Моделирование политропного процесса сжатия воздуха в изделии цилиндрической формы в инженерной среде <i>LabVIEW</i>	39
Лабораторная работа № 3. Автоматизация экспериментальных исследований по определению электрической емкости конденсатора в программной среде <i>LabVIEW</i>	56
Лабораторная работа № 4. Автоматизация учета времени включения и выключения приборов в среде <i>LabVIEW</i>	70
Лабораторная работа № 5. Функции генерации, ввода и обработки данных в <i>LabVIEW</i> . Интеграция <i>LabVIEW</i> и <i>OPC</i> сервера	92
Лабораторная работа № 6. Создание проекта в <i>MasterSCADA</i> с использованием <i>OPC</i> -сервера и типизации	114
Лабораторная работа № 7. Создание проекта в <i>MasterSCADA</i> по отслеживанию уровня жидкости в баке и контролю его наполнения	152
Лабораторная работа № 8. Создание проекта по контролю смешивания жидкостей разной температуры в <i>MasterSCADA</i>	174
Приложение А Загрузка и установка <i>LabVIEW</i> на <i>Windows</i>	190
Приложение Б Загрузка и установка <i>MasterSCADA</i> на <i>Windows</i>	192
Приложение В Загрузка и установка <i>OPC</i> сервера на <i>Windows</i>	195

ВВЕДЕНИЕ

В условиях конкуренции на рынке производства программно-технических средств многопрофильных систем (далее – ПТСМС) одной из важнейших задач для организаций, занятых в высокотехнологичных сферах экономики, является создание в сжатые сроки высоконадежных программных и технических продуктов с высоким уровнем характеристик и применением передовых технологий.

Для разработки современных ПТСМС используются и постоянно совершенствуются сложные математические модели, специализированные и универсальные программные алгоритмы; внедряются автоматизированные системы проектирования и программные среды для разработки виртуальных приборов; применяются системы сбора данных и оперативного диспетчерского управления технологическими процессами.

При разработке ПТСМС, представляющих собой взаимосвязанную совокупность программных, информационных и технических средств, инженерам необходимо:

- знать принципы автоматизации процессов на производстве, требующих использования программных продуктов, предназначенных для разработки или обеспечения работы в реальном времени систем сбора, обработки, отображения и архивирования информации об объекте мониторинга или управления;
- понимать особенности разработки аппаратно-программных интерфейсов многофункциональных модулей обработки данных с внешним оборудованием в различных программных средах;
- принимать решения по построению систем автоматического управления, основываясь на анализе и выборе наиболее эффективных программных и технических средств обмена, ввода, обработки и вывода информации;
- уметь определять возможность и целесообразность применения новых информационных технологий в своей профессиональной деятельности;
- владеть навыками проектирования распределенных компьютерных систем управления и контроля средствами в различных SCADA-системах.

Разработанный лабораторный практикум нацелен на формирование у обучающихся теоретических знаний и практических навыков в построении ПТСМС с учетом подходов системной инженерии и включает в себя восемь работ, состоящих из цели работы, теоретических сведений, порядка выполнения лабораторной работы, плана содержания отчета, контрольных вопросов и списка вспомогательной литературы.

Выполнение лабораторных работ позволит изучить:

- процессы взаимодействия между элементами виртуальных приборов посредством базовых функций *LabVIEW*, построения графиков сложных кривых и особенности работы с тригонометрическими, квадратичными, кубическими и экспоненциальными функциями (лабораторная работа № 1);
- разработку виртуальных приборов для моделирования политропного сжатия газа с начальными условиями (P_0 , V_0 и T_0), а также способы отображения

результатов на лицевой панели в виде различных индикаторов и графиков (лабораторная работа № 2);

– методы автоматизации обработки данных моделируемого в программной среде *LabVIEW* процесса разрядки конденсатора с учетом задаваемых исходных данных (ЭДС, сопротивление, остаточное напряжение) с возможностью вывода значения величины заряда, емкости и времени его разрядки на лицевую панель (лабораторная работа № 3);

– принципы автоматизации учета времени включения и выключения ламп с возможностью фиксации и записи времени включения и выключения в таблицу на лицевой панели, а также отслеживанием необходимого времени работы лампы (лабораторная работа № 4);

– разработку программы для генерации сигнала и наложения на него шума с возможностью отображения на лицевой панели виртуального прибора сигнала без шума, самого шума, сигнала с шумом и спектром сигнала (лабораторная работа № 5);

– способы расчета активной, реактивной и полной мощностей в программной среде *MasterSCADA* с использованием исходных данных, поступающих с *OPC*-сервера, а также работу функции типизации на основе создания проекта, состоящего из нескольких устройств, одно из которых интегрировано из *LabVIEW* (лабораторная работа № 6);

– моделирование процесса наполнения и опустошения бака с возможностью отслеживания входящего и выходящего потоков, позволяющего оценить уровень воды в баке, а также способы моделирования процесса наполнения бака с организацией перемешивания потоков из трех различных баков с возможностью управления потоками и контролем уровней воды в них (лабораторная работа № 7);

– моделирование процесса смешивания жидкостей различной температуры с возможностью управления потоками воды для достижения необходимой температуры, а также отслеживание показателей температуры и уровней воды (лабораторная работа № 8).

Учебно-методическое пособие для выполнения лабораторных работ по дисциплине «Программно-технические средства многопрофильных систем» для студентов специальности 6-05-0713-02 «Электронные системы и технологии» было разработано в рамках гранта Президента Республики Беларусь на 2024 год в сфере образования (распоряжение Президента Республики Беларусь № 4рп от 3 января 2024 г. «О предоставлении грантов Президента Республики Беларусь на 2024 год»).

Лабораторная работа № 1
**Основы работы в среде *LabVIEW*: разработка виртуальных приборов,
исследование функций и построение сложных кривых**

Цель работы: ознакомиться с организацией кроссплатформенной графической среды разработки приложений *LabVIEW*, изучить правила и инструменты для создания собственных программ в ней, а также освоить основные приемы программирования и отладки программ.

Основные задачи:

- 1 Получить навыки работы с тригонометрическими, квадратичными, кубическими и экспоненциальными функциями.
- 2 Приобрести практические навыки создания, редактирования и отладки компьютерных приборов.
- 3 Закрепить полученные знания путем выполнения индивидуального задания.

1.1 Теоретические сведения

1.1.1 Общие сведения. Функциональные возможности программной среды *LabVIEW*

Инженерная среда программирования *LabVIEW*, или *Laboratory Virtual Instrumentation Engineering Workbench* (рус. «среда разработки лабораторных виртуальных приборов»), создана для разработки компьютерных систем измерения и управления экспериментальными средствами и технологическими процессами. Ее главной особенностью является графический код, который позволяет быстро проектировать и отлаживать сложные программы.

Кроме того, *LabVIEW* может использоваться для проведения различных расчетов, вычисления аналитических функций и построения их графиков; анализа и архивирования данных и генерации отчетов. Среда обладает уникальными возможностями быстрой отладки программ и наблюдения за потоком данных в анимационном режиме. С помощью *LabVIEW* можно моделировать, например, процесс амплитудной модуляции, где в качестве несущего сигнала выступает синусоида, что облегчит вычисления, а также может визуализировать результаты вычислений.

В *LabVIEW* разрабатываемые программные модули называются «*Virtual Instruments*» (рус. «виртуальные приборы») или просто «*VI*». Они сохраняются в файлах с расширением **.vi* и являются своеобразным основанием, на котором строится любая программа в *LabVIEW*.

Стоит отметить, что разработчиком может быть создана своя библиотека виртуальных приборов, которая характерна для особенностей процессов в организации, а также любой *VI* может быть вызван из другого *VI*.

При создании собственных программ в среде *LabVIEW* могут использоваться следующие инструменты: «*Front Panel*», «*Block Diagram*», «*Controls*» и «*Functions*».

ВАЖНО ЗНАТЬ!

«*Front Panel*» (рус. «лицевая панель») – графический интерфейс пользователя (*GUI*) программного средства *LabVIEW*, который позволяет визуально представить элементы управления и отображения данных, созданные пользователем.

«*Block Diagram*» (рус. «блок-диаграмма») – часть графического интерфейса пользователя (*GUI*) среды разработки *LabVIEW*; место, где разрабатывается логика программы с помощью графического языка программирования, известного как *G* (*Graphical*).

«*Controls*» (рус. «элементы управления») – набор графических элементов, которые используются на «*Front Panel*» программы для создания интерфейса пользователя. Эти элементы управления включают в себя различные кнопки, переключатели, ползунки, индикаторы, текстовые поля и другие графические объекты, которые позволяют пользователю взаимодействовать с программой.

«*Functions*» (рус. «функции») – набор графических функциональных блоков, которые используются на «*Block Diagram*» виртуального инструмента (*VI*) для создания логики программы. Эти функциональные блоки представляют собой различные операции и алгоритмы, которые могут быть применены для обработки данных, управления потоком выполнения или выполнения других задач в приложении.

В основе *LabVIEW* лежит парадигма потоков данных, в которой константа и терминал индикатора соединяются между собой линией. Эта линия называется «*Wire*» и представляет собой своеобразный *провод*. По проводам передаются данные от одних элементов другим, при этом сформированные узлы начнут работать только тогда, когда придут все необходимые для этого данные. Вся эта концепция называется «*Data Flow*».

Концепция создания проекта в *LabVIEW* строится на последовательном выполнении следующих шагов:

- создать новый проект;
- добавить в проект новую виртуальную панель;
- на виртуальной панели разместить необходимые элементы управления и отображения данных;
- связать элементы управления и отображения данных с функциональными блоками, содержащими алгоритмы обработки данных и управления программой;
- настроить параметры функциональных блоков и элементов управления;
- скомпилировать программу в исполняемый файл;
- запустить программу и проверить ее работу.

Ниже рассмотрим основные особенности графического интерфейса программной среды, ее основные функции и инструменты, необходимые для разработки виртуальных приборов.

1.1.2 Графический интерфейс, функции и инструменты среды *LabVIEW*

При каждом запуске *LabVIEW* из меню стартового диалогового окна командами «New» → «Blank VI» открываются два окна – «Front Panel» и «Block Diagram» (рисунки 1.1 и 1.2).



Рисунок 1.1 – Фрагмент диалогового окна «Front Panel»



Рисунок 1.2 – Фрагмент диалогового окна «Block Diagram»

В правом верхнем углу каждого окна находится пиктограмма для архивирования созданной программы в качестве нового виртуального прибора. Здесь же размещена традиционная для приложений *Windows* полоса главного меню с одинаковыми для обоих окон пунктами: «File», «Edit», «Operate», «Tools», «Browse», «Windows» и «Help». Краткое описание их функций приведено в таблице 1.1.

Таблица 1.1 – Краткое описание функций главного меню

Пункты главного меню	Описание функции
«File»	Открытие, закрытие, сохранение и печать программ
«Edit»	Редактирование панелей, поиск объектов
«Operate»	Запуск и прерывание выполнения программ
«Tools»	Управление библиотеками программ
«Browse»	Просмотр иерархий программ
«Windows»	Отображение окон и палитр
«Help»	Дополнительная информация об элементах и функциях среды разработки лабораторных виртуальных приборов <i>LabVIEW</i>

Ниже полос главного меню расположены линейки инструментов, которые различны для «*Front Panel*» и «*Block Diagram*» за счет дополнительных кнопок для отладки программ.

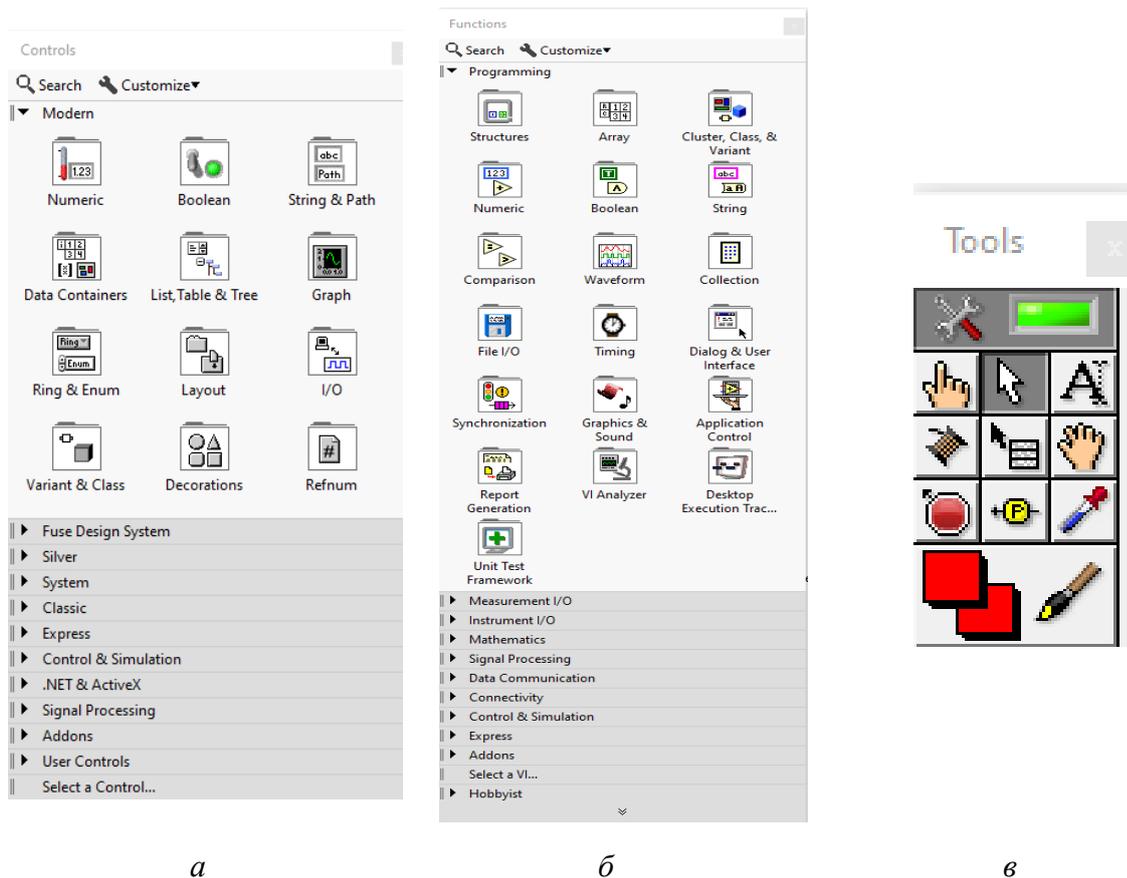
Свободное пространство каждой панели образует рабочую область, снабженную горизонтальной и вертикальной полосами прокрутки. При разработке программ в рабочей области «*Front Panel*» размещаются визуальные элементы управления и индикации, формирующие интерфейс пользователя, а на панели «*Block Diagram*» составляется графический код создаваемого приложения.

Внешний вид пиктограммы и назначение кнопок инструментальных панелей представлены в таблице 1.2.

Таблица 1.2 – Назначение кнопок инструментальных панелей

Пиктограмма	Назначение кнопок инструментальных панелей
	Кнопка « <i>Запуск</i> » (« <i>Run</i> ») при правильно составленной программе
	Вид кнопки « <i>Запуск</i> » (« <i>Run</i> ») при наличии ошибок в программе
	Вид кнопки « <i>Запуск</i> » (« <i>Run</i> ») в процессе выполнения программы
	Вид кнопки « <i>Запуск</i> » (« <i>Run</i> ») в процессе выполнения подпрограммы
	Кнопка « <i>Непрерывный</i> » (повторяющийся) запуск (« <i>Run Continuosly</i> »)
	Кнопка « <i>Остановка выполнения программы</i> » (« <i>Abort Execution</i> »)
	Кнопка временной паузы выполнения программы (« <i>Pause</i> »)
	Анимация потоков данных при отладке программ
	Начало пошагового выполнения отладки программ
	Пошаговое выполнение программ
	Выход из пошагового выполнения программ
	Редактирование текста (шрифт, размер, стиль текста)

Разработка программ осуществляется с помощью трех вспомогательных палитр, в которых находятся все доступные блоки кода и функции, которые используются в процессе разработки проектов: «*Controls*», «*Functions*» и «*Tools*» (рисунок 1.3).



а – палитра «Controls»; *б* – палитра «Functions»; *в* – палитра «Tools»
 Рисунок 1.3 – Вспомогательные палитры

ВАЖНО ЗНАТЬ!

«*Controls*» *Palette* (рус. «палитра элементов управления») – представляет собой иерархический список элементов управления для программирования на *Front Panel*. Элементы управления распределены по категориям, чтобы упростить навигацию и помочь сосредоточиться на конкретных типах данных или стилях пользовательского интерфейса.

«*Functions*» *Palette* (рус. «палитра функций») – панель, содержащая виртуальные приборы, функции и константы, используемые для создания блок-диаграмм.

«*Tools*» *Palette* (рус. «палитра инструментов») – панель, содержащая различные инструменты и элементы, которые используются для создания, редактирования и управления виртуальными инструментами.

LabVIEW имеет интуитивно понятный и простой интерфейс, который позволяет создавать программы путем перетаскивания и соединения блоков кода, а палитры имеют иерархическую структуру, что делает поиск нужных инструментов быстрым и удобным.

Палитра «*Controls*» (рисунок 1.3, *а*) доступна на «*Front Panel*», палитра «*Functions*» (рисунок 1.3, *б*) доступна на «*Block Diagram*», а «*Tools*» (рисунок 1.3, *в*) доступна на обеих панелях.

Инструменты, представленные в «Tools», являются специальным рабочим режимом курсора мыши, который можно выбрать вручную или автоматически. Их обозначение и описание представлены в таблице 1.3.

Перейти в «Tools» также можно с помощью щелчка правой кнопки мыши с удерживанием клавиши «Shift» в Windows или «+» в MacOS.

Таблица 1.3 – Назначения инструментов в LabVIEW

Вид инструмента	Описание
	<i>Управление («Operate value»)</i> – применяется для изменения значения элементов управления или ввода текста, а также для оперирования кнопками, переключателями и другими элементами
	<i>Перемещение («Position/Size/Select»)</i> – применяется для активизации, перемещения или изменения размеров объектов
	<i>Ввод текста («Edit text»)</i> – применяется для создания и редактирования текстовых ярлыков, а также для начала ввода с клавиатуры в текстовые и цифровые поля
	<i>Соединение («Connect Wire»)</i> – применяется для соединения объектов на «Block Diagram», а также используется для подключения элементов управления и индикаторов лицевой панели к терминалам соединительной панели виртуального прибора
	<i>Вызов контекстного меню («Object Shortcut Menu»)</i> – вызывает контекстное меню соответствующего объекта по щелчку левой кнопкой мыши
	<i>Быстрая прокрутка экрана («Scroll Window»)</i> – применяется для просмотра видимой области окна без использования полосы прокрутки
	<i>Ввод контрольной точки («Set/Clear Breakpoint»)</i> – позволяет расставлять контрольные точки в функциях, узлах, проводниках данных, структурах и временно приостанавливать выполнение программы в них
	<i>Установка отладочных индикаторов («Probe data»)</i> – используется для создания зондов на проводниках, соединяющих элементы «Block Diagram», и показывает текущее значение переменных, необходимое при отладке программ и просмотра промежуточных значений
	<i>Копирование цвета («Get color»)</i> – предназначен для копирования цвета из существующего объекта и последующего переноса этого цвета на другие объекты
	<i>Цвет («Set color»)</i> – служит для раскрашивания объектов панелей, переднего и заднего планов. Можно установить цвета переднего и заднего планов, щелкнув мышью по соответствующей цветовой области в палитре

1.1.3 Типы и проводники данных в LabVIEW

При работе в программной среде LabVIEW используются такие типы данных, как «Numeric» (численный тип), который подразделяется на «Floating point» (число с плавающей запятой) и «Integer» (целочисленный тип); «Boolean» (логический тип), который может принимать только два значения – «0» («FALSE») или «1» («TRUE»); «String» (строковый тип); «Path» (путь к файлу); «Array» (массивы); «Cluster» (кластеры), которые включают в себя различные типы данных;

«Waveform» (сигнальный тип), который является кластером элементов, содержащим данные, начальное значение времени и интервал времени между измерениями, а также «Dynamic» (динамический тип).

Обозначения элементов управления и индикаторов, а также их цветовое изображение и описание значения типов данных представлены в таблице 1.4.

Таблица 1.4 – Типы данных в *LabVIEW*

Элемент управления	Индикатор	Тип данных	Цвет	Значение по умолчанию
1	2	3	4	5
		Числовой с плавающей запятой расширенной точности	Оранжевый	0
		Числовой с плавающей запятой двойной точности	Оранжевый	0
		Числовой с плавающей запятой одинарной точности	Оранжевый	0
		Число с фиксированной точкой	Фиолетовый	0
		64-разрядное целое число со знаком	Синий	0
		32-разрядное целое число со знаком	Синий	0
		16-разрядное целое число со знаком	Синий	0
		8-разрядное целое число со знаком	Синий	0
		Беззнаковое 64-битное целое число	Синий	0
		Беззнаковое 32-битное целое число	Синий	0
		Беззнаковое 16-битное целое число	Синий	0
		Беззнаковое 8-битное целое число	Синий	0
		Комплексный с плавающей запятой одинарной точности	Оранжевый	0+i0
		Комплексный с плавающей запятой двойной точности	Оранжевый	0+i0
		Комплексный с плавающей запятой расширенной точности	Оранжевый	0+i0
		Массив – включает тип данных в квадратные скобки и принимает цвет этого типа данных	Серый	–
		Перечисляемый тип	Синий	–
		Логический	Зеленый	FALSE

Продолжение таблицы 1.4

1	2	3	4	5
		Строковый	Розовый	Пустая строка
		Путь	Морской волны (бирюзовый)	Не путь (<i>not a path</i>)
		Кластер – включает разные типы данных	Коричневый или розовый	–

Виртуальный прибор *LabVIEW* представляет собой единое целое за счет проводников данных («*Wires*»), соединяющих элементы. Проводники являются каналами прохождения данных от терминала-источника к одному или нескольким терминалам-приемникам.

Если при разработке программы попытаться присоединить к проводнику более чем один источник или вообще ни одного источника, то *LabVIEW* выдаст сообщение, что проводник «поврежден» («*Broken*»). Может быть только один источник данных, но один или несколько приемников.

Принцип соединения источников и приемников проводниками объясняет, почему управляющие элементы и индикаторы не могут заменять друг друга. Управляющие элементы – это источники, а индикаторы – приемники данных.

Каждый проводник имеет свой стиль и цвет в зависимости от типа данных, проходящих по нему.

Провод, передающий одномерный массив, состоит из одной линии. Линия может иметь форму извилистой линии или пересечения извилистых линий, если передаются дискретные данные. К ним, например, относятся булевские и строковые данные.

Провод, передающий одномерный массив, изображается одной линией, двумерный – состоит из двух линий, трехмерный – тоже из двух линий, но находящихся на большем расстоянии друг от друга, и т. д. Цвет провода соответствует типу элементов массива.

Провода для передачи целочисленных данных имеют светло-синий цвет, для передачи булевских данных – зеленый цвет, а для передачи строковых данных – малиновый цвет.

На провода для передачи массивов очень похожи провода, через которые передаются динамические данные или кластеры данных.

1.2 Задания для выполнения лабораторной работы

Задание 1. Составить простую программу взаимосвязи элементов управления и индикаторов, а также проанализировать полученные данные.

Порядок выполнения задания 1

Существите запуск среды *LabVIEW*.

В появившемся главном окне программы выберите команды: «File» → «New» для создания нового файла. Затем в появившемся окне «New» выберите шаблон «Blank VI» (рисунки 1.4 и 1.5).

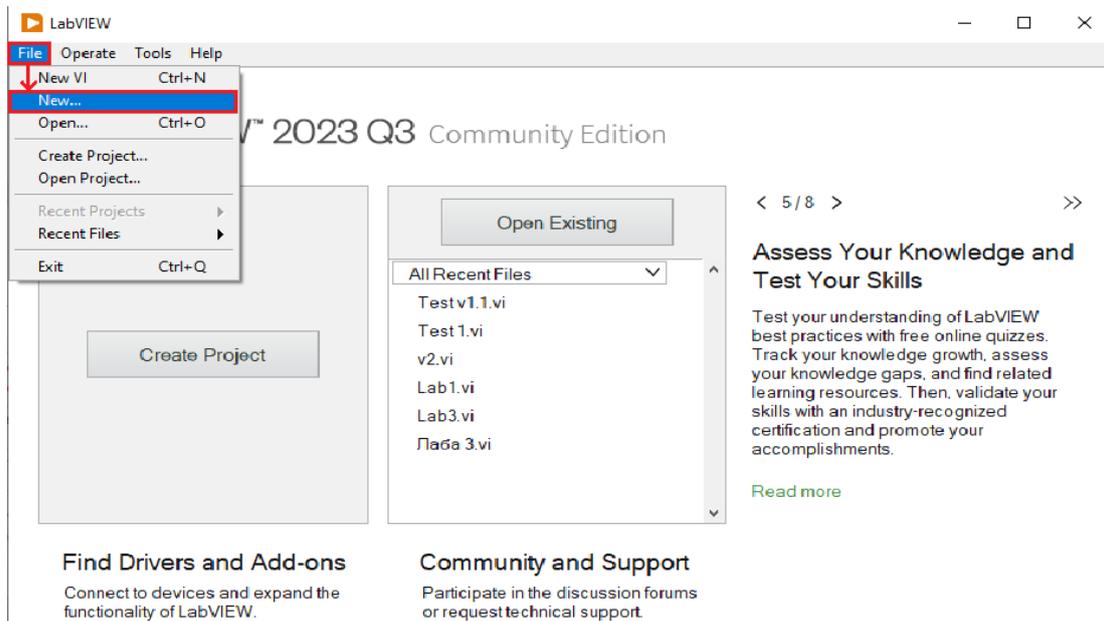


Рисунок 1.4 – Диалоговое окно и контекстное меню для запуска программной среды *LabVIEW*

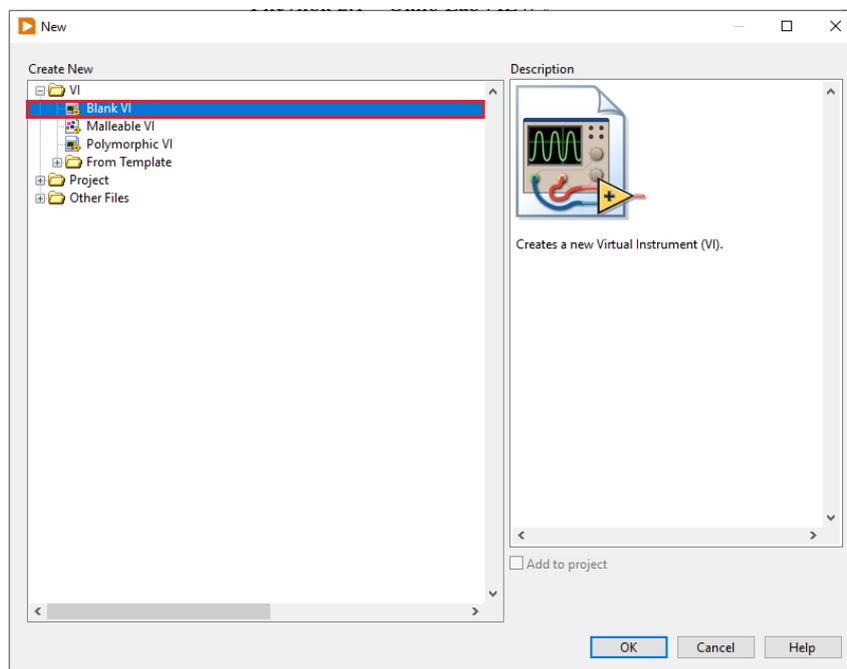


Рисунок 1.5 – Диалоговое окно для выбора нового шаблона

Далее в меню «Window», как показано на рисунке 1.6, выберите «Tile Left and Right» для одновременного отображения на экране двух окон программы: серой и белой панелей.

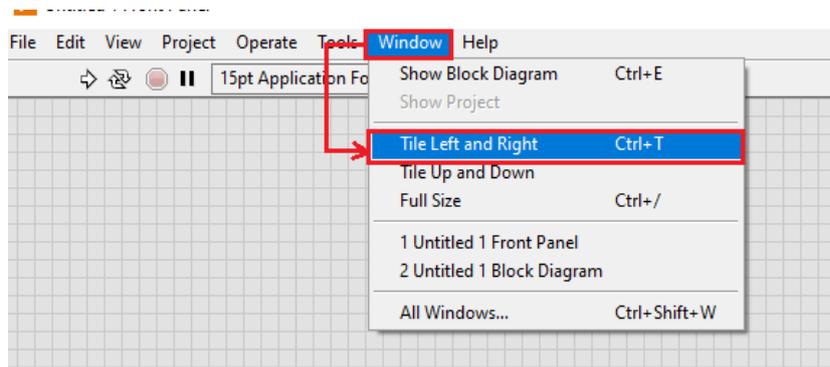


Рисунок 1.6 – Выбор функции одновременного отображения двух окон

Серая «*Front Panel*» (обычно располагается слева) – инструмент пользователя, который предназначен для размещения элементов ввода и вывода данных в виде привычных технических устройств, таких как цифровые указатели, ползунковые реостаты, регуляторы громкости, осциллографы, самописцы, графопостроители и т. д.

Белая (обычно располагается справа) – «*Block Diagram*», на которой вызываются пиктограммы различных функций и структур, а также составляется графический код программы.

Для совершения различных операций с помощью курсора мыши необходимо вызвать палитру инструментов из меню «*View*» → «*Tools Palette*» на «*Front Panel*» или на «*Block Diagram*» (рисунок 1.7).

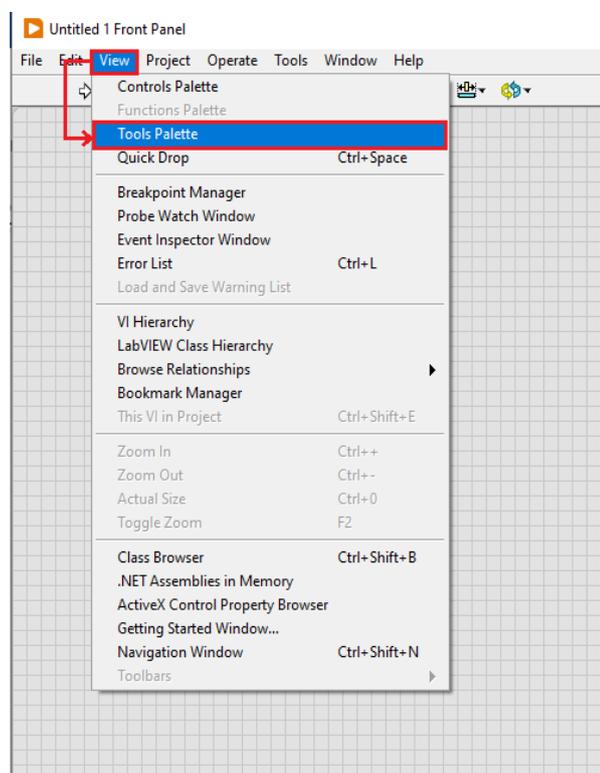


Рисунок 1.7 – Диалоговое окно для вызова «*Tools Palette*»

Щелчком правой кнопки мыши по сетке на «*Front Panel*» вызываем палитру элементов контроля и управления и закрепляем ее в стационарном положении с помощью инструмента  в левом верхнем углу палитры.

В ней активизируем элементы контроля «*Numeric*» для задания исходных параметров. Выделяем курсором и поочередно переносим на верхнюю часть «*Front Panel*» цифровой регулятор – «*Numeric Control*», реостат – «*Horizontal Pointer Slide*», ручку регулятора громкости – «*Knob*» (рисунок 1.8).

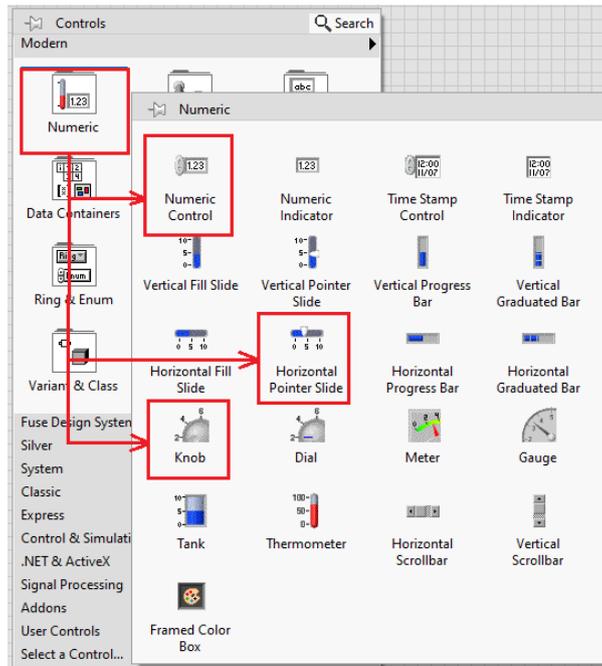


Рисунок 1.8 – Панель «*Numeric*»

Воспользовавшись инструментом «*Edit text*» () , расположенным на панели «*Tools*», переименуем элементы на «*Numerical controller*», «*Rheostat*» и «*Volume control*» (рисунок 1.9).

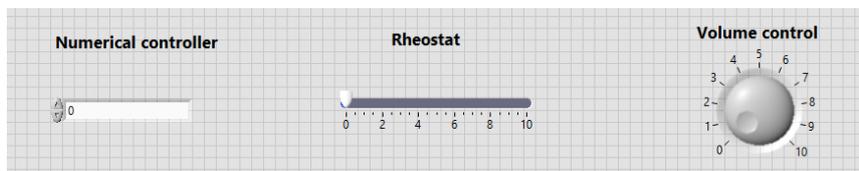


Рисунок 1.9 – Расположение элементов «*Numerical controller*», «*Rheostat*» и «*Volume control*» на «*Front Panel*»

Создадим пять элементов индикации работы этих приборов:

- стрелочный амперметр – «*Meter*»,
- манометр – «*Gauge*»,
- термометр – «*Thermometer*»,
- 2 линейных индикатора – «*Numeric Indicator*» и «*Numeric Indicator 2*»;
- осциллограф – «*Waveform Chart*».

Для этого в панели «Controls» выбираем «Numeric», находим элементы «Meter», «Gauge», «Thermometer», два «Numeric Indicator» и перетаскиваем их в рабочую область «Front Panel» (рисунок 1.10).

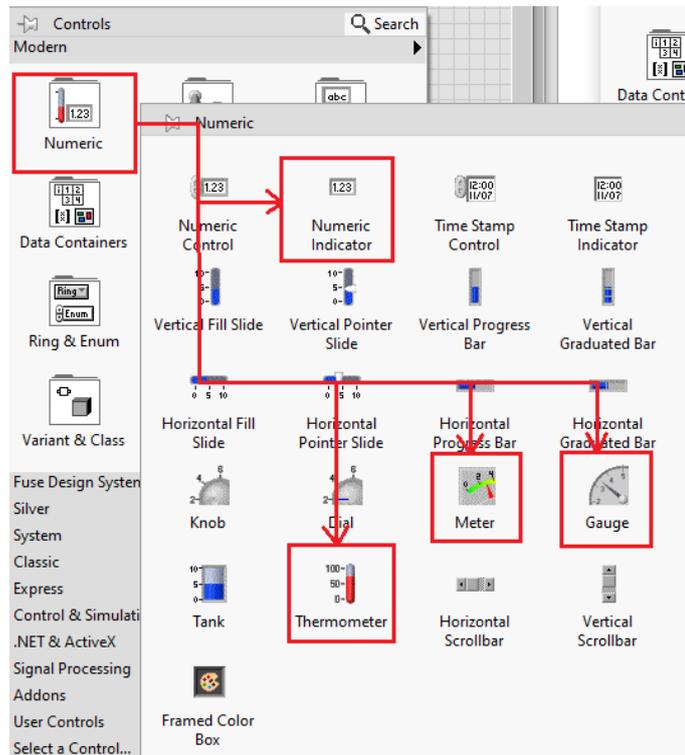


Рисунок 1.10 – Добавление элементов индикации на «Front Panel»

Для добавления осциллографа на «Front Panel» необходимо в «Controls» перейти в «Graph» и выбрать «Waveform Chart» (рисунок 1.11).

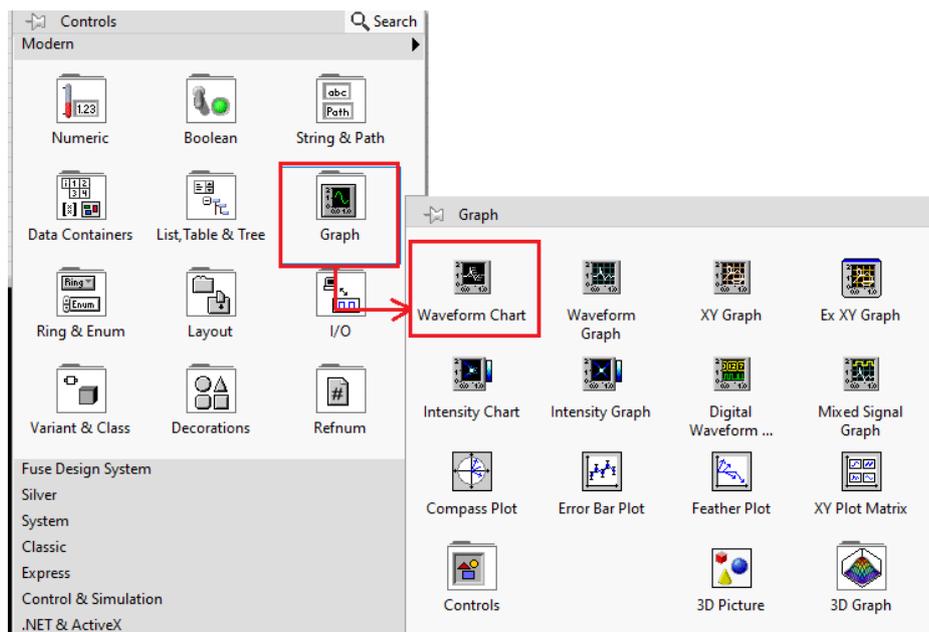


Рисунок 1.11 – Панель «Graph»

На данном этапе на «*Front Panel*» будут отображены добавленные элементы, и она примет следующий вид (рисунок 1.12).

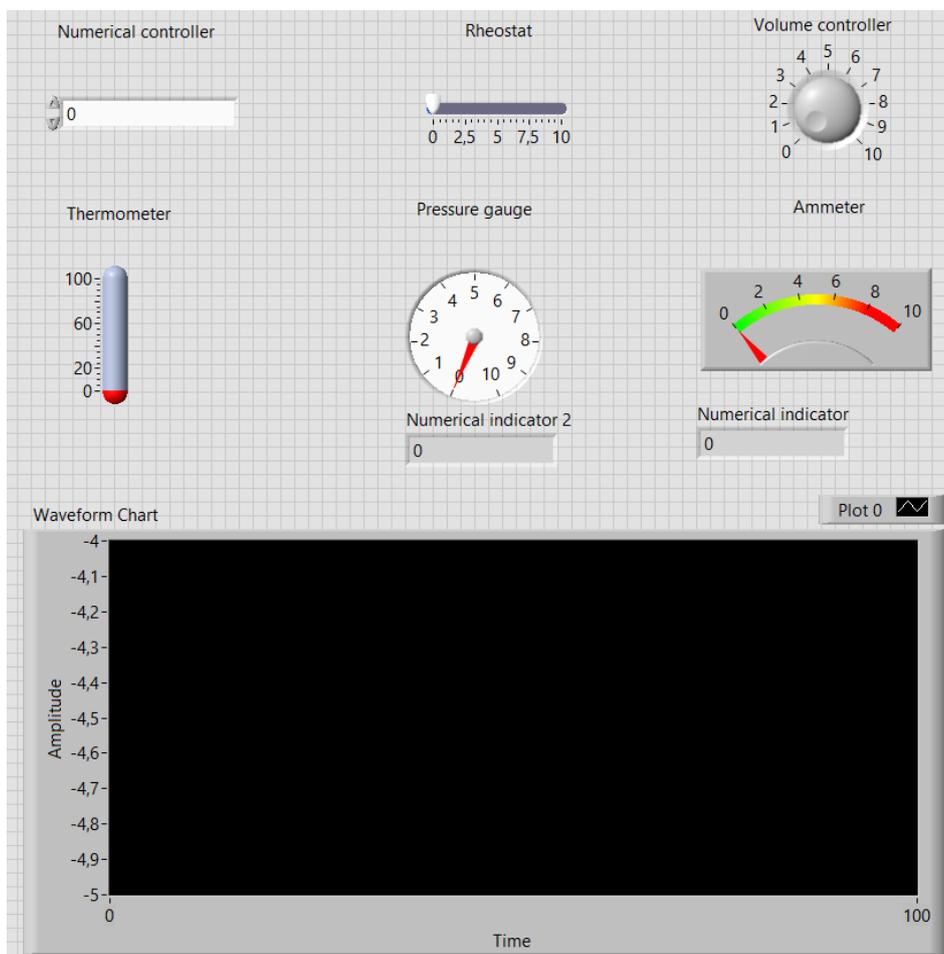


Рисунок 1.12 – Рабочая зона «*Front Panel*» с размещенными элементами

Для работы с «*Block Diagram*» можно вызвать дополнительные инструменты, которые вызываются из главного меню «*Tools Palette*» при помощи последовательного выбора «*View*» → «*Tools Palette*».

Подадим выходные сигналы управляющих элементов на входы индикаторов в соответствии с вариантом, выданным преподавателем, соединяя их проводниками данных с помощью мыши или элемента «*Connect Wire*» на «*Block Diagram*». Поскольку управляющих элементов меньше, чем индикаторов, разделим выходы управляющих элементов «*Volume controller*» и «*Rheostat*» на два за счет присоединения дополнительного проводника к линии передачи данных, после чего присоединим к входу индикатора «*Numerical indicator*» и «*Numerical indicator 2*» соответственно. Затем выход управляющего элемента, указанного в вашем варианте, соедините с входом осциллографа («*Waveform Chart*»), дополнительно разделив выход управляющего элемента.

При белой стрелке «*Run*» (↵) включаем периодический запуск (⏮) работы составленных программ. Изменяя на лицевой панели значения исходных вели-

чин в соответствии с вариантом, проследим отображение этих изменений на показывающих приборах. Обратим внимание на соответствие шкал управляющих элементов и показывающих приборов. При необходимости скорректируйте их с помощью инструмента «редактирование текста».

Добавим виртуальный генератор случайных чисел. Щелчком правой кнопки мыши на «Block Diagram» вызовем панель «Functions» и закрепим ее. В ней находим палитру «Numeric», открываем и переносим на «Block Diagram» два элемента: суммирование –  («Add») и генератор случайных чисел –  («Random Number (0–1)»), который генерирует случайные числа в пределах от 0 до 1 (рисунок 1.13). Генератор случайных чисел будем использовать для создания симуляции осциллограммы.

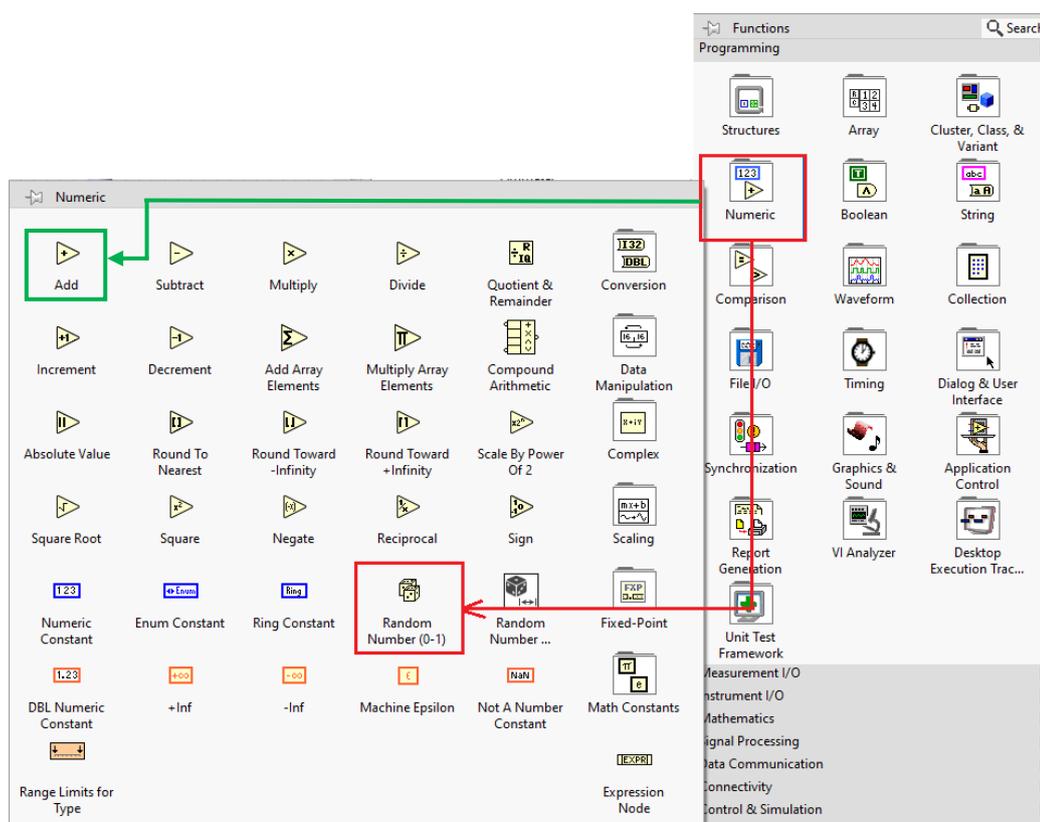
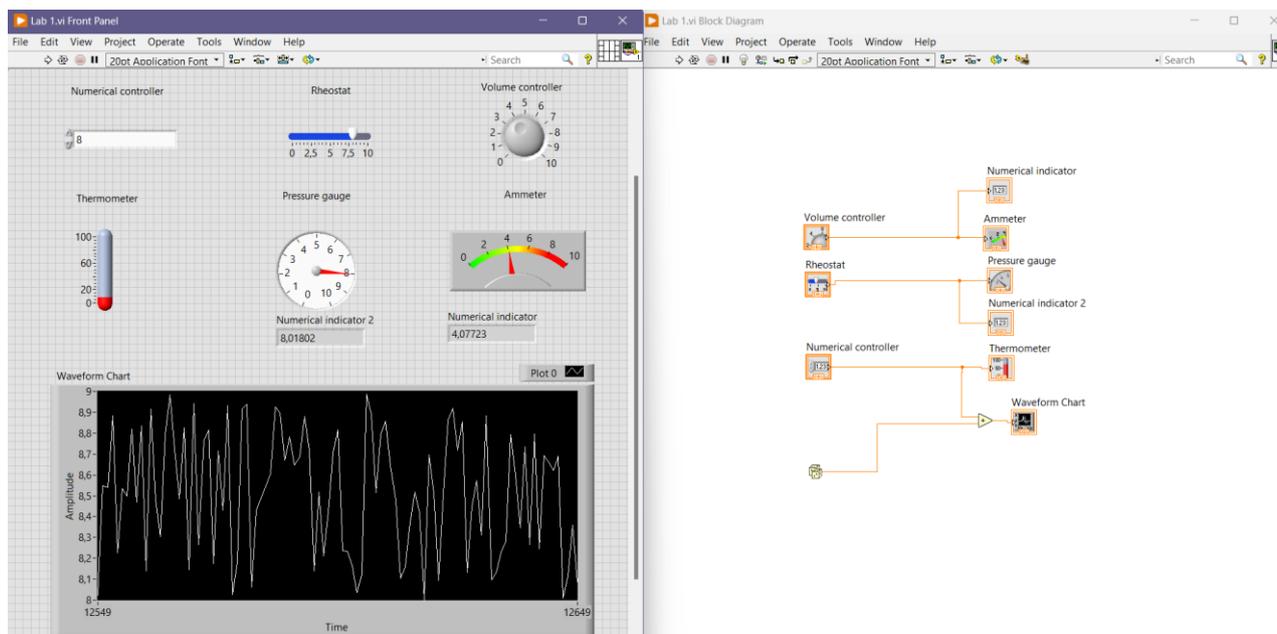


Рисунок 1.13 – Палитра арифметических действий

Удаляем проводник между регулятором и осциллографом.

К нижнему входу элемента «Add» подсоединяем выход элемента «Random Number (0–1)», а к верхнему входу суммирования подсоединяем выход управляющего элемента, указанного в вашем варианте. Запускаем программу нажатием кнопки «Run Continuously» () и, изменяя значения на регуляторах (допускается погрешность $\pm 0,5$) в соответствии с вариантом, выданным преподавателем, наблюдаем изменения на индикаторах (рисунок 1.14).



a

б

Рисунок 1.14 – Интерфейс пользователя (*a*) и графический код (*б*)

Остановите программу кнопкой «*Abort execution*» (🛑). С помощью инструмента «*Highlight Execution*» (💡💡) и кнопки «*Run Continuously*» включите режим анимации потоков данных, используемый при отладке программ. Проследите движение данных по проводникам и их преобразование на элементах «*Block Diagram*».

Задание 2. В программной среде *LabVIEW* разработать программу вычисления координат и построить график окружности, заданной параметрическим способом, по формулам

$$X = X_0 + R \cdot \cos(A), \quad (1.1)$$

$$Y = Y_0 + R \cdot \sin(A), \quad (1.2)$$

где X_0 – координата центра окружности по оси абсцисс;
 R – радиус окружности;
 A – параметр, $A \in [0; 360^\circ]$;
 Y_0 – координата центра окружности по оси ординат.

Порядок выполнения задания 2

Осуществите запуск среды *LabVIEW*. В появившемся главном окне программы выберите команды «*File*» → «*New*» для создания нового файла. Далее выберите меню «*Window*» → «*Tile Left and Right*» для одновременного отображения на экране двух окон программы.

Для совершения различных операций с помощью курсора мыши можно вызвать палитру инструментов («*Tools Palette*») последовательно выбирая в меню «*View*» → «*Tools Palette*» на «*Front Panel*» или на «*Block Diagram*».

Создайте на «*Front Panel*» четыре цифровых элемента управления для исходных данных задачи: X_0 , Y_0 , R , A . Для этого щелчком правой кнопки мыши по «*Front Panel*» вызовите палитру «*Controls*» и закрепите ее, активизировав кнопку в левом верхнем углу палитры.

Откройте пункт меню «*Numeric*», выберите в нем первый элемент в верхнем ряду. На открывшейся палитре «*Controls*» выделить элемент «*Numeric Control*» со спаренной кнопкой изменения значения параметров (рисунок 1.15). Переместите четыре элемента поочередным перетаскиванием на «*Front Panel*» и расположите их горизонтально в одну строку, как показано на рисунке 1.16.

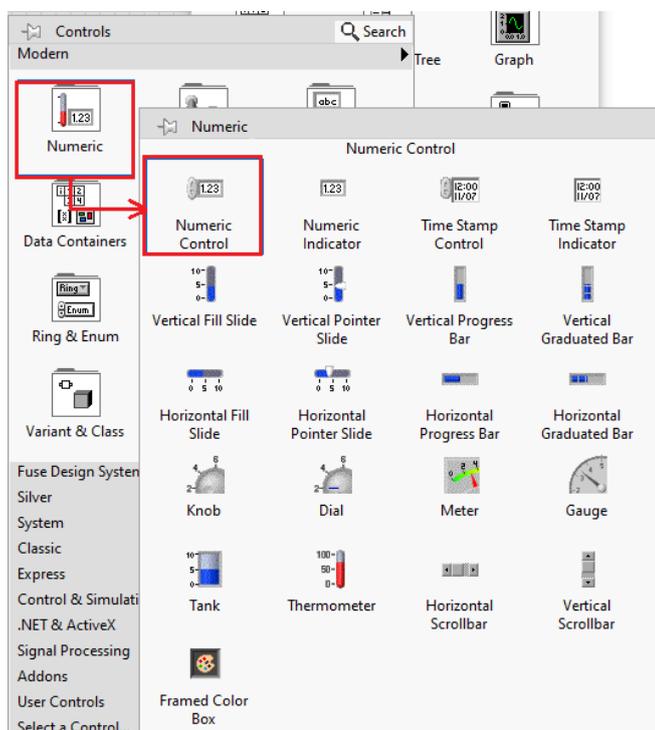


Рисунок 1.15 – Палитра «*Controls*»

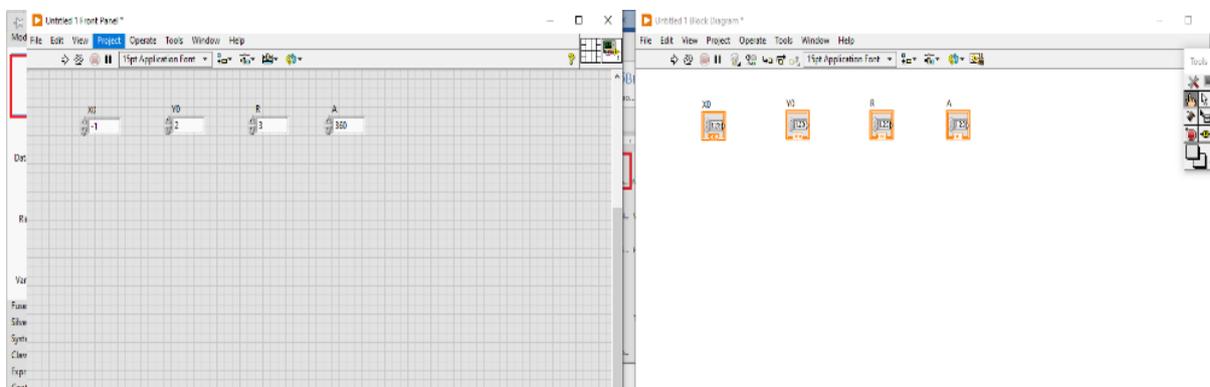


Рисунок 1.16 – Расположение цифровых элементов

Измените собственные метки вызванных регуляторов, подписав вместо «*Numeric*» новые обозначения: X_0 , Y_0 , R , A . Установите в окошках регуляторов соответствующие значения исходных данных. Значения X_0 , Y_0 , R , A набираются либо с помощью клавиш указателей, либо с помощью ввода с клавиатуры.

Создайте на «*Front Panel*» два прибора для отображения полученных данных – «*Waveform Chart*» – двухлучевой запоминающий осциллограф, работающий в режиме реального времени, и «*Ex XY Graph*» – двухкоординатный самописец. Для этого вернитесь на главную панель «*Controls*», откройте «*Graph*» (графические индикаторы), из которых на «*Front Panel*» выносятся «*Waveform Chart*» и «*Ex XY Graph*» (рисунок 1.17).

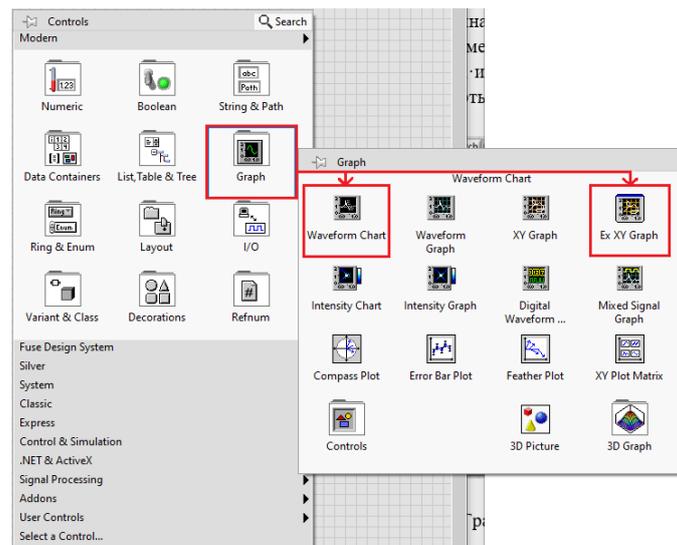


Рисунок 1.17 – Графические индикаторы

После проделанных шагов рабочая зона на «*Front Panel*» и «*Block Diagram*» будет иметь вид, представленный на рисунке 1.18.

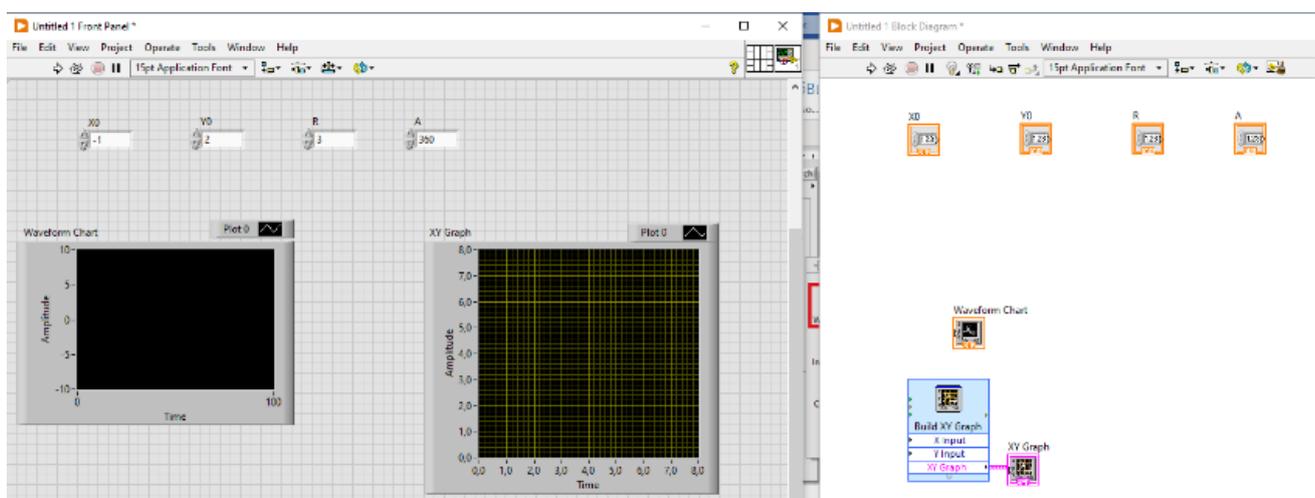


Рисунок 1.18 – Вид рабочей зоны

При появлении элементов на «*Front Panel*» их пиктограммы сразу же появляются на «*Block Diagram*». Освободите среднюю часть «*Block Diagram*» для построения графического кода программы путем перетаскивания имеющихся элементов в верхнюю и нижнюю части экрана. Далее щелчком правой кнопки мыши на «*Block Diagram*» вызовите «*Functions*». Используя кнопку в верхнем левом углу палитры, нужно зафиксировать ее на экране. В «*Functions*» вызываем элемент «*Structures*», далее в нем выбираем «*For loop*» – цикл, повторяющий вычисления с заданным числом итераций (рисунок 1.19). Перетаскиваем его на «*Block Diagram*» и растягиваем на большую часть экрана.

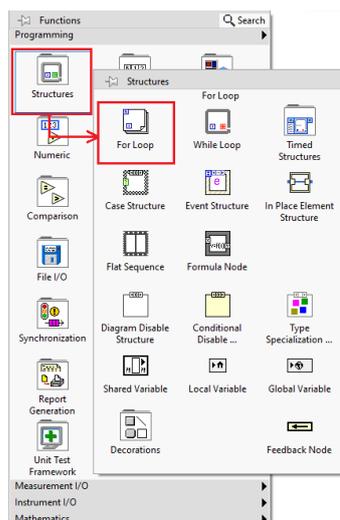


Рисунок 1.19 – Палитра функций для выбора циклических структур

Вернитесь к «*Functions*», откройте «*Programming*» → «*Numeric*» (панель для операций с числами) и зафиксируйте ее на экране с помощью значка . Далее реализуйте формулы, заданные в условии задачи, с помощью узлов подпалитры «*Numeric*» (рисунок 1.20).

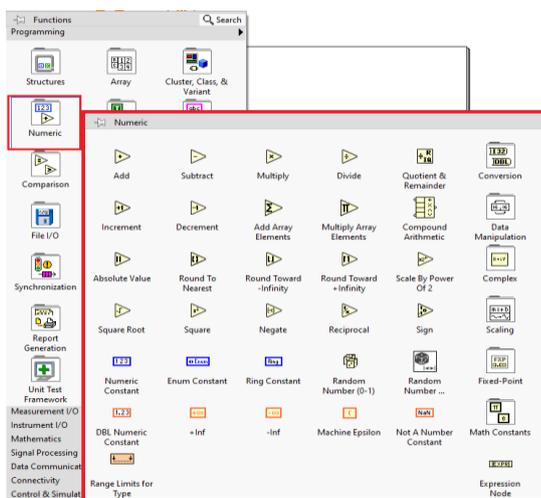


Рисунок 1.20 – Палитра операций с числами

Так как в *LabVIEW*, как и во многих других средах тригонометрические функции работают с радианной мерой угла, переведите градусы в радианы. Для этого выберите число 2π в «*Numeric*» → «*Math Constants*» и перетащите его внутрь цикла на «*Block Diagram*» (рисунок 1.21).

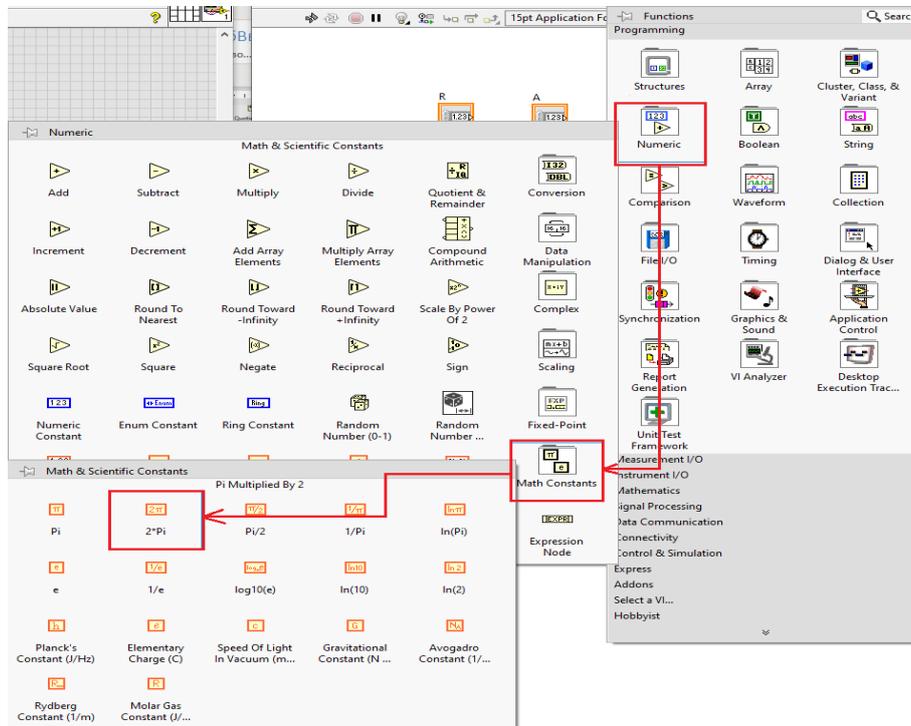


Рисунок 1.21 – Панель математических констант

Вернувшись в «*Numeric*», перенесите внутрь цикла на «*Block Diagram*» два арифметических узла: «*Divide*» – деление и «*Multiply*» – умножение. Щелкнув по выходу цифрового элемента «*A*», выведите его значение на границу цикла. Соедините выход узла 2π с верхним терминалом узла деления, а к нижнему с помощью проводника подведите значение «*A*». Входы каждой пиктограммы располагаются на ее левой стороне, а выходы – на правой.

Выход узла деления соединяем с одним из входов узла умножения, а на другой подаем значение текущей итерации от узла «*i*» внутри цикла. Для этого необходимо соединить находящийся в нижнем левом углу поля цикла синий значок «*i*» с соответствующим входом узла умножения. Полученное на его выходе значение равно текущему значению «*A*» в радианной мере. При выполнении цикла «*i*» последовательно принимает значения от 0 до 359, а величина «*A*» меняется от 0 до 2π .

Далее необходимо добавить тригонометрические функции и пиктограммы функций \sin и \cos (рисунок 1.22). Перенесите их внутрь цикла и соедините входы тригонометрических функций и значение «*A*» в радиальной мере (выход операции умножения).

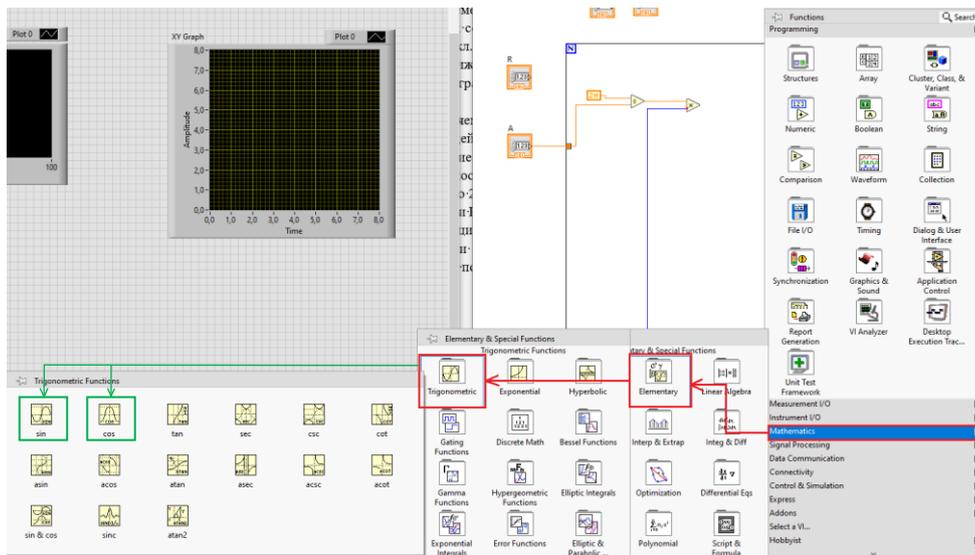


Рисунок 1.22 – Панель тригонометрических функций

Вернитесь на панель «*Numeric*» и добавьте около функций \sin и \cos математические операторы «*Multiply*» (умножение) и «*Add*» (сложение).

Умножьте R на значение $\sin(A)$ и $\cos(A)$. Выходы узлов умножения просуммируйте с Y_0 и X_0 соответственно. Выходы сумматоров выведите на правую границу цикла.

Рабочая область окна «*Block Diagram*» после проделанных операций будет иметь следующий вид (рисунок 1.23).

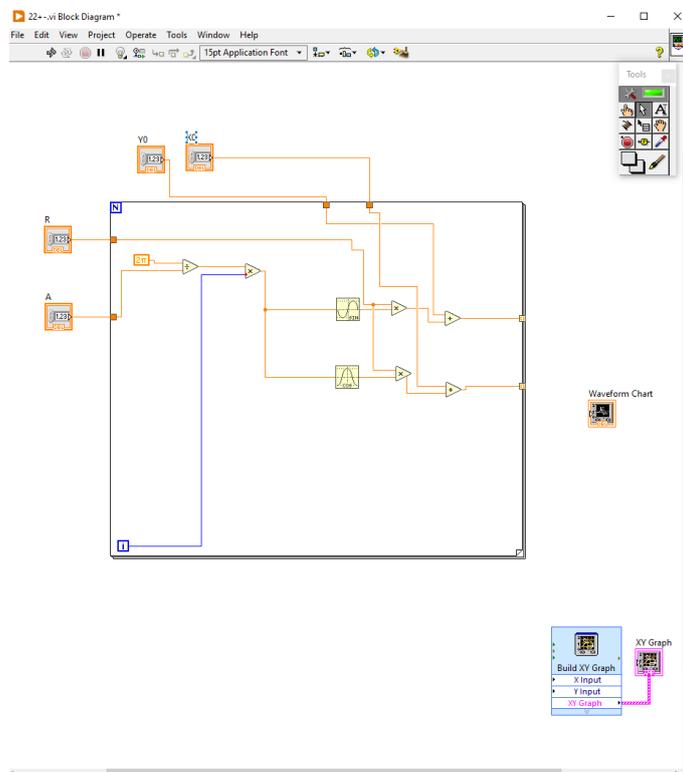


Рисунок 1.23 – Вид окна «*Block Diagram*»

Соедините образовавшиеся массивы данных X и Y с соответствующими входами XU -самописца для получения графика окружности. Обратите внимание на «сломанную стрелку» запуска программы. Активизируйте ее левой кнопкой мыши, в открывшемся окне появится перечень ошибок, допущенных при составлении программы. Одна из них: «*незаполненный терминал счетчика итераций*», который подает команду на завершение циклических вычислений. Исправить ошибку можно продублировав соединение элемента управления « A » с внешним входом счетчика итераций « N ».

После исправления ошибок запустите программу и убедитесь, что построенная кривая является окружностью радиусом R и центром окружности, расположенным в точке с координатами X_0 и Y_0 .

Запустите программу в режиме анимации потоков данных. Для этого активизируйте в строке команд на «*Block Diagram*» элемент в виде «лампочки», а затем стрелку запуска программы. Изучите движение данных и их трансформацию в различных узлах блок-диаграммы и сделайте выводы о возможности использования этого режима при отладке сложных программ.

Для отображения изменения координат окружности в зависимости от параметра внесите внутрь цикла «*Waveform Chart*». Для преобразования его в многолучевой прибор на «*Functions*» → «*Programming*») откройте «*Cluster, Class & Variant*», а в ней пиктограмму – «*Bundle*» (объединение), захватите ее и перенесите внутрь цикла к входу осциллографа (рисунок 1.24).

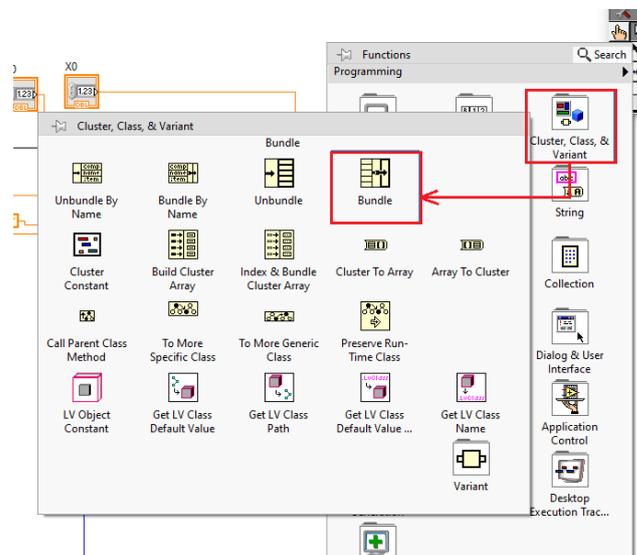


Рисунок 1.24 – Элемент «*Bundle*»

К двум ее входам подключите X и Y , взятые от проводников соответствующих данных. Выход объединительной панели соедините с осциллографом.

Для того чтобы замедлить вычисления и наблюдать построение кривых координат окружностей во времени, с палитры «*Functions*» вызывается палитра «*Timing*», а затем выбираем элемент «*Wait Until Next ms Multiple*» в виде метронома, который переносится внутрь цикла (рисунок 1.25).

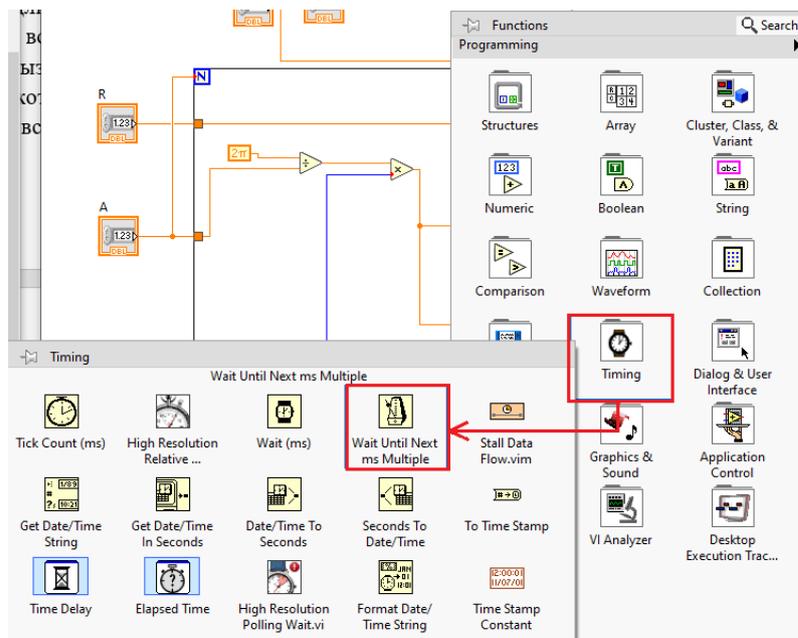
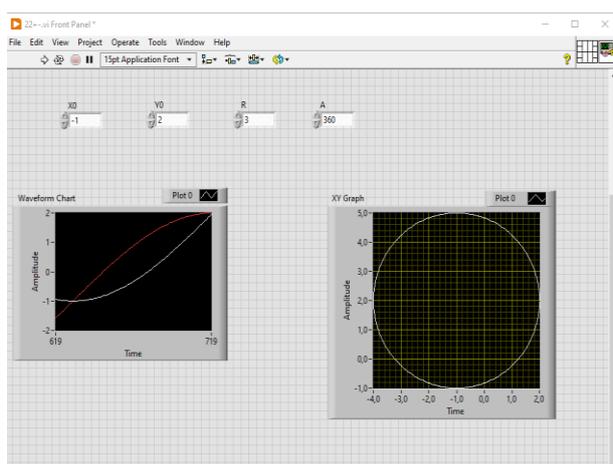


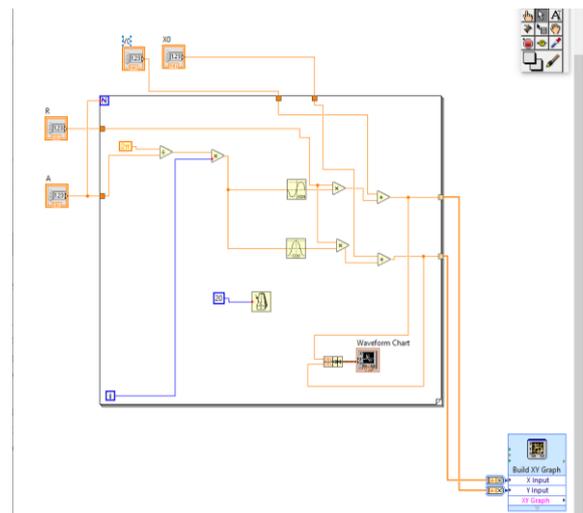
Рисунок 1.25 – Элемент «*Wait Until Next ms Multiple*»

По умолчанию цикл рассчитывает всю программу со скоростью 1000 раз в секунду. Чтобы замедлить этот процесс, подведем к метроному константу «*Numeric Constant*» из палитры «*Numeric*», введем скорость задержки 20 мс и соединим его с метрономом. При этом процесс расчета и отображения координат X и Y замедлится в 20 раз.

Запустите программу и просмотрите процесс построения графиков. Обратите внимание, что первые графики строятся в реальном времени, а сама кривая в координатах X - Y на самописце только после окончания всего цикла вычислений. Окончательный вид представлен на рисунке 1.26.



а



б

а – размещение элементов проекта в окне «*Front Panel*»; б – размещение элементов проекта в окне «*Block Diagram*»

Рисунок 1.26 – Окончательный вид программы для вычисления координат и построения графика окружности

Задание 3. Разработать программу, в которой независимая переменная X в заданных формулах имитируется генератором синусоиды в пределах изменения его амплитуды. Значение амплитуды задается оператором через ползунок.

Порядок выполнения задания 3

Создайте новый файл, как показано в предыдущем задании.

На «*Front Panel*» щелчком правой кнопки мыши вызовите окно «*Controls*». Выберите «*Graph*» и перетащите на «*Front Panel*» элемент «*XY Graph*» (двухкоординатный графический индикатор). Измените размер элемента по желанию. Двойным щелчком левой кнопки мыши по названию можно изменить название элемента.

В палитре «*Controls*» выберите «*Numeric*» и перетащите на «*Front Panel*» элемент «*Horizontal Pointer Slide*», как показано на рисунке 1.27. Измените название регулятора на «*Max X*». Щелчком правой кнопки мыши по регулятору вызовите контекстное меню. В появившемся окне выберите «*Visible Items*» → «*Digital Display*». Это необходимо для точного задания значения X .

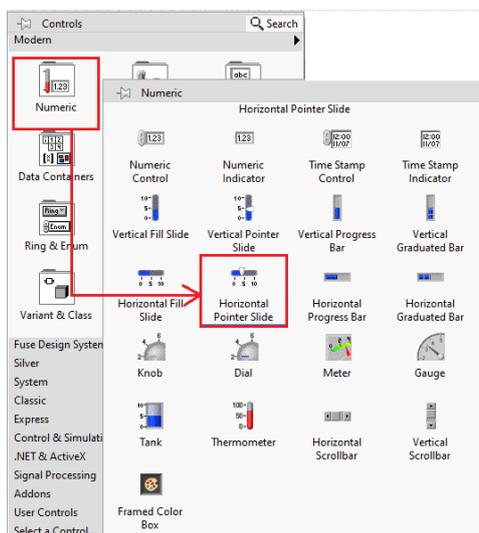


Рисунок 1.27 – Палитра «*Numeric*»

Теперь организуем текстовую индикацию формул и управление выбором нужной формулы с лицевой панели. Для этого в панели «*Controls*» выбираем «*Ring & Enum*» и перетаскиваем оттуда на «*Front Panel*» элемент «*Text Ring*» («*текстовый переключатель*») (рисунок 1.28). Переименуйте данный элемент на «*Function type*».

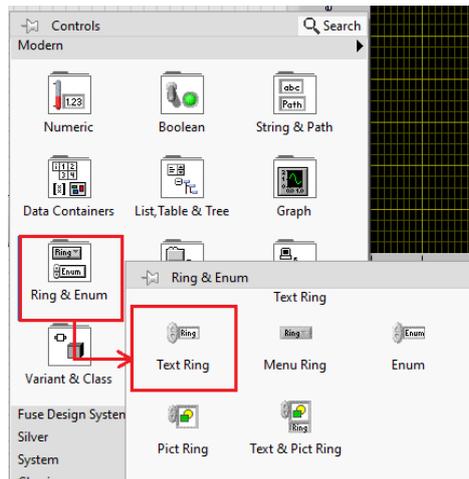


Рисунок 1.28 – Палитра «*Ring & Enum*»

Вызовите контекстное меню «*Text Ring*», щелкнув правой кнопкой мыши по элементу и выбрав «*Properties*». В появившемся окне выберите «*Edit Items*» (рисунок 1.29). В первую строку запишите формулу $Y = X^2$. Далее, выбирая новую строку, запишите последовательно еще три формулы: $Y = X^3$, $Y = \exp(X)$ и $Y = \text{sinc}(X)$.

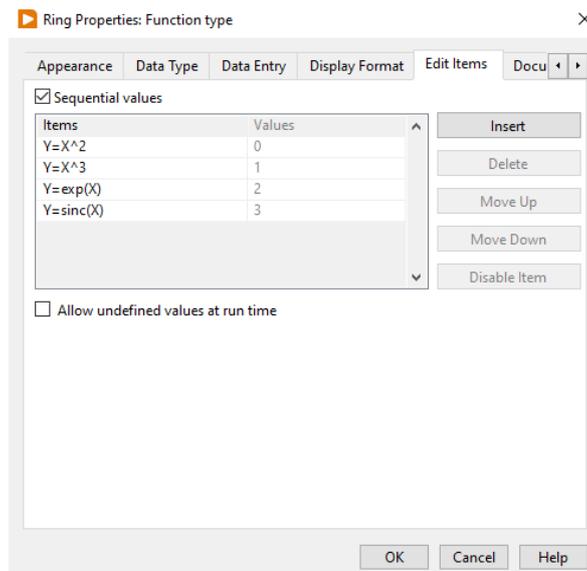


Рисунок 1.29 – Окно «*Edit Items*» с введенными формулами

В палитре «*Controls*» выберем «*Boolean*» и перетащим на «*Front Panel*» элемент «*Stop Button*» (рисунок 1.30).

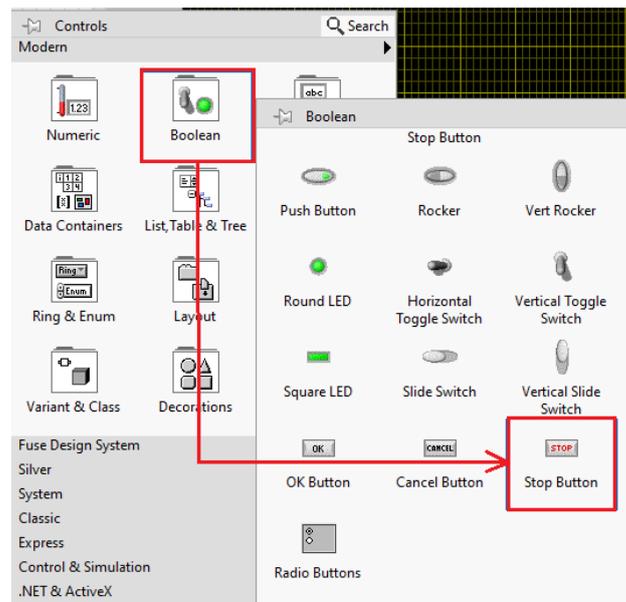


Рисунок 1.30 – Палитра «Boolean»

На данном этапе «Front Panel» будет иметь вид, представленный на рисунке 1.31.

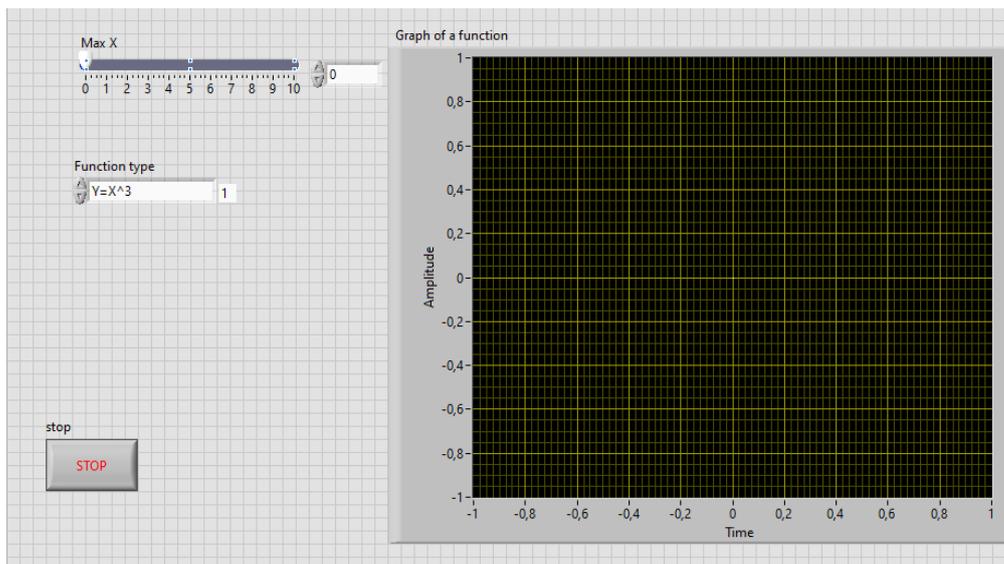


Рисунок 1.31 – Вид «Front Panel»

Перейдем в окно «Block Diagram». Разместите элементы «Max X», «Function type» и «Stop» справа, а «XY Graph» слева.

Щелчком правой кнопки мыши по окну «Block Diagram» вызовите палитру «Functions» и закрепите ее. В «Functions» выберите «Signal Processing», далее выберите «Sig Generation» и оттуда перетащите на «Block Diagram» элемент «Sine Pattern» – генератор синусоиды (рисунок 1.32).

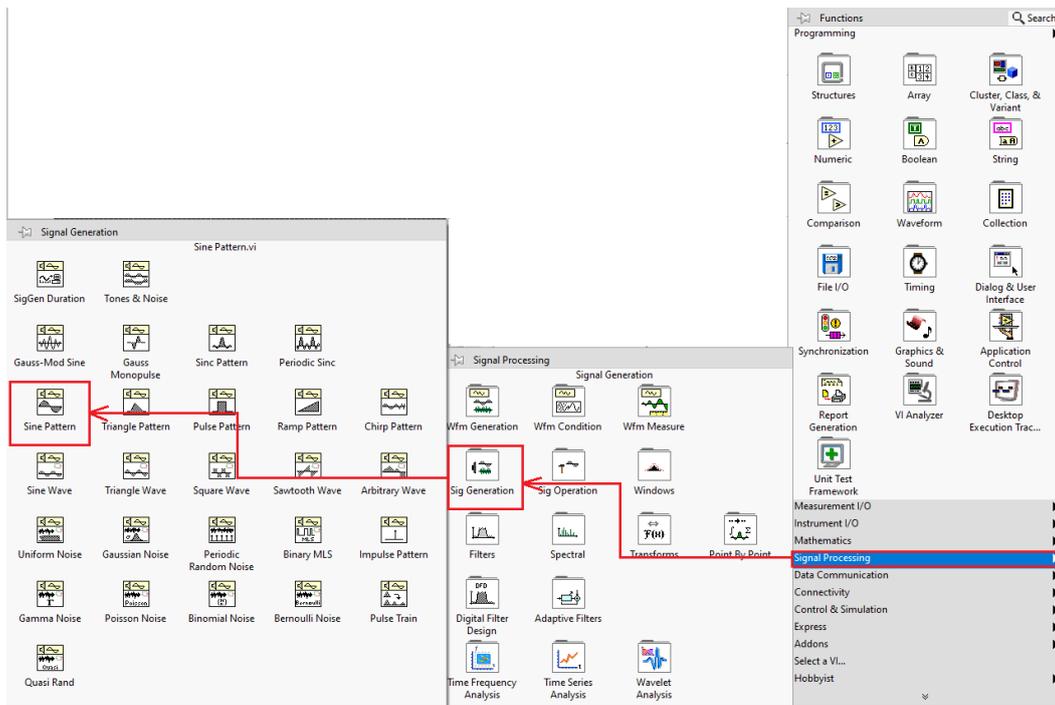


Рисунок 1.32 – Палитра «*Signal Processing*»

Подключите выход элемента «*Max X*» ко входу «*amplitude*» элемента «*Sine Pattern*». Наведите курсор мыши на вход «*samples*» элемента «*Sine Pattern*» и нажмите правую кнопку мыши. В появившемся окне выберите «*Create*» → «*Constant*» и укажите значение константы – 1000. Аналогичную операцию проведите со входами «*phase*» («*degrees*») и «*cycles*» указав значение констант «0» и «1» соответственно. Таким образом задается нулевая начальная фаза и один период генерируемой синусоиды.

В палитре «*Functions*» выберите «*Structures*» и разместите на «*Block Diagram*» элемент «*For Loop*» (рисунок 1.33).

В этой же палитре «*Structures*» выберите элемент «*Case Structure*» и разместите внутри элемента «*For Loop*». Переместите элемент «*Function type*» внутрь элемента «*For Loop*». Соедините выход текстового переключателя с входом «*Case Selector*» элемента «*Case Structure*» (прямоугольник со знаком «?»).

В палитре «*Structures*» выберите элемент «*Formula Node*» и разместите его внутри варианта № 0 элемента «*Case Structure*» (рисунок 1.34). Введите в окно «*Formula Node*» формулу $Y = X ** 2$, обязательно поставьте «;» в конце формулы.

Щелчком правой кнопки мыши по левой стороне «*Formula Node*» выберите «*Add Input*» и подпишите его «*X*». Аналогично, щелкнув справа, выберите «*Add Output*» и подпишите его «*Y*».

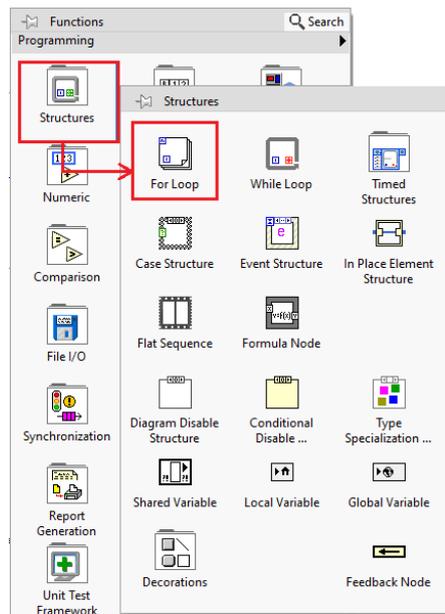


Рисунок 1.33 – Палитра «Structures»

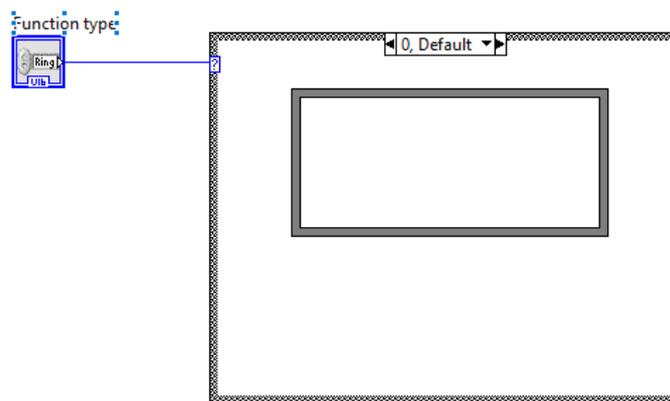


Рисунок 1.34 – Элемент «Formula Node» внутри элемента «Case Structure»

Щелчком правой кнопки мыши по элементу «Case Structure» выберите «Add Case After» и добавьте необходимое количество вариантов, чтобы их было четыре. В каждом из вариантов сделайте те же шаги, что и для нулевого варианта. В окне «Formula Node» введите формулы для всех вариантов соответственно: $Y = X ** 3$, $Y = \exp(X)$, $Y = \text{sinc}(X)$.

В палитре «Functions» выберите «Cluster, Class & Variant» и перетащите на «Block Diagram» элемент «Bundle» (рисунок 1.35). Функция «Bundle» служит для объединения двух потоков данных, соответствующих переменным X и Y , до их подачи на двухкоординатный графический индикатор.

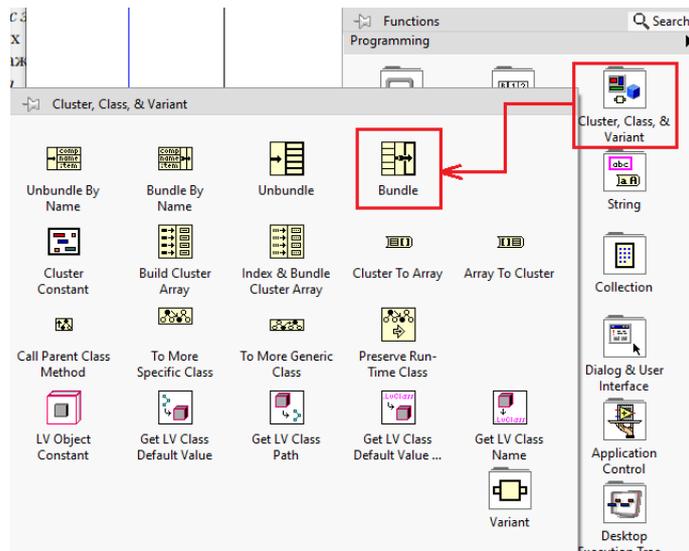


Рисунок 1.35 – Палитра «Cluster, Class & Variant»

В палитре «Function» выберите «Application Control», перетащите элемент «Stop» и разместите его справа от кнопки «stop» (рисунок 1.36).

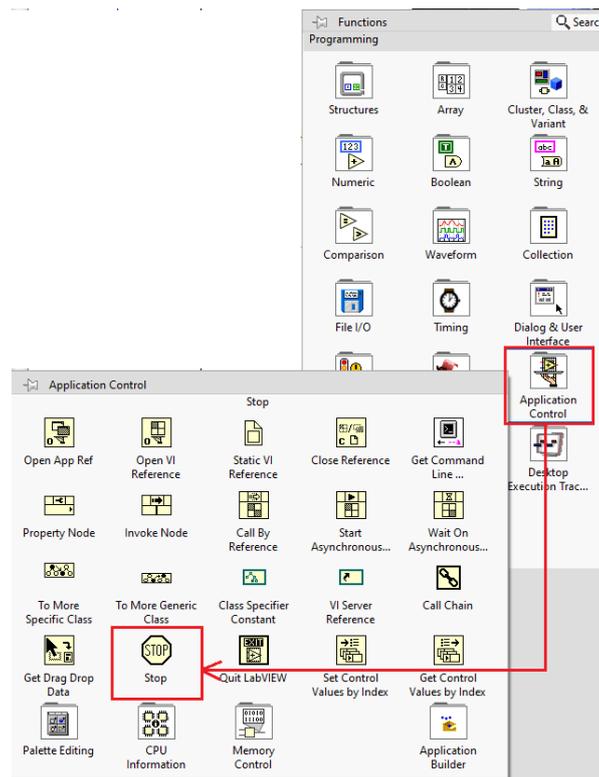


Рисунок 1.36 – Палитра «Application Control»

Подключите к терминалу счета цикла «For Loop» константу 1000, идущую к входу «samples» (рисунок 1.37).

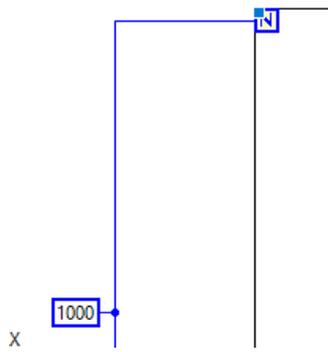


Рисунок 1.37 – Подключение к терминалу счета цикла

Подключите выход элемента «*Sine Pattern*» к входу «*X*» элемента «*Formula Node*» для каждого из вариантов. Также выход элемента «*Sine Pattern*» подключите к элементу «*Bundle*».

Выход «*Y*» элемента «*Formula Node*» подключите к элементу «*Bundle*». Соедините «*Bundle*» и «*XY Graph*».

Соедините выход кнопки «*stop*» с входом элемента «*Stop*».

Окно «*Block Diagram*» на данном этапе будет иметь вид, представленный на рисунке 1.38.

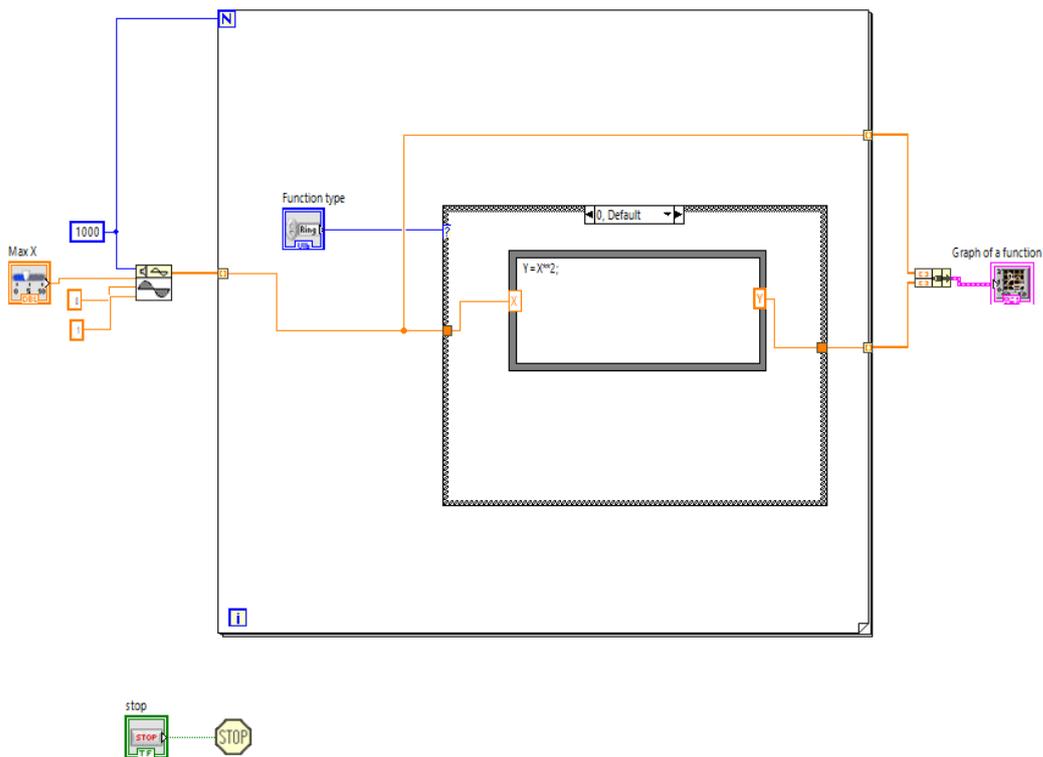


Рисунок 1.38 – Вид «*Block Diagram*»

Запустите программу, нажав кнопку «*Run Continuously*» . Изменяя значения *X* и вариант формулы, можно наблюдать различные графики функций.

1.3 Индивидуальные задания

В рамках лабораторной работы предусмотрено выполнение заданий в соответствии с индивидуальным номером зачетной книжки каждого студента. При этом важно учесть, что повторение заданий не допускается, обеспечивая таким образом самостоятельное выполнение своей работы. В случае совпадения номера зачетной книжки с уже занятым заданием студенту предоставляется возможность выбрать свободный номер из предложенных вариантов.

Варианты для задания 1, задания 2 и задания 3 представлены в таблицах 1.5, 1.6 и 1.7 соответственно.

Таблица 1.5 – Варианты индивидуального задания 1

Вариант	Основной управляющий элемент	Соответствующий управляющему элементу индикатор	Значение, установленное на основном регуляторе
1	<i>Rheostat</i>	<i>Pressure gauge</i>	2
2	<i>Numerical controller</i>	<i>Thermometer</i>	15
3	<i>Numerical controller</i>	<i>Thermometer</i>	75
4	<i>Numerical controller</i>	<i>Thermometer</i>	13
5	<i>Volume controller</i>	<i>Ammeter</i>	1
6	<i>Volume controller</i>	<i>Ammeter</i>	3
7	<i>Rheostat</i>	<i>Pressure gauge</i>	2
8	<i>Numerical controller</i>	<i>Thermometer</i>	22
9	<i>Rheostat</i>	<i>Ammeter</i>	4
10	<i>Numerical controller</i>	<i>Thermometer</i>	5
11	<i>Volume controller</i>	<i>Pressure gauge</i>	4
12	<i>Numerical controller</i>	<i>Thermometer</i>	42
13	<i>Rheostat</i>	<i>Ammeter</i>	9
14	<i>Numerical controller</i>	<i>Thermometer</i>	17
15	<i>Volume controller</i>	<i>Pressure gauge</i>	4
16	<i>Numerical controller</i>	<i>Thermometer</i>	76
17	<i>Numerical controller</i>	<i>Thermometer</i>	8
18	<i>Volume controller</i>	<i>Pressure gauge</i>	10
19	<i>Numerical controller</i>	<i>Thermometer</i>	53
20	<i>Rheostat</i>	<i>Ammeter</i>	7
21	<i>Volume controller</i>	<i>Ammeter</i>	2
22	<i>Numerical controller</i>	<i>Thermometer</i>	23
23	<i>Rheostat</i>	<i>Ammeter</i>	7
24	<i>Volume controller</i>	<i>Ammeter</i>	5
25	<i>Numerical controller</i>	<i>Thermometer</i>	45
26	<i>Rheostat</i>	<i>Ammeter</i>	9
27	<i>Rheostat</i>	<i>Ammeter</i>	0
28	<i>Rheostat</i>	<i>Pressure gauge</i>	6
29	<i>Rheostat</i>	<i>Ammeter</i>	3
30	<i>Numerical controller</i>	<i>Thermometer</i>	29

Таблица 1.6 – Варианты индивидуального задания 2

Вариант	X_0	Y_0	R
1	-7	-7	5
2	10	3	5
3	9	-9	8
4	4	9	3
5	-1	-1	8
6	8	-3	8
7	-4	4	10
8	10	7	2
9	5	9	5
10	2	0	10
11	9	4	4
12	-6	7	4
13	-6	4	4
14	10	8	7
15	9	2	9
16	-4	-3	2
17	-9	1	9
18	-4	10	5
19	-6	-5	1
20	-5	5	6
21	-3	-1	3
22	-2	-8	1
23	10	-1	4
24	0	-8	9
25	-10	4	2
26	-10	3	4
27	-10	10	8
28	9	9	5
29	7	-6	7
30	-3	9	2

Таблица 1.7 – Варианты индивидуального задания 3

Вариант	Функция № 1	Функция № 2	Функция № 3
1	2	3	4
1	X^2	X^3	$\text{sinc}(X)$
2	$\cos(X)$	$\cot(X - 2) + 3$	$\sinh(X - 2) + 1$
3	$\sin(X)$	$\csc(2 * X + 1)$	$\sec(X + 1)$
4	$\cot(X)$	$\cos(3 * X) + 1$	$\text{sqrt}(X) * 2$
5	$\tan(X)$	X^4	$\csc(X)$
6	$\cot(X - 2) + 3$	$\sin(X) + 1$	$\tan(X) - 4$
7	$\csc(X + 1) + 4$	$\sinh(X) + 3$	$3 + X^3$
8	$\sec(X + 1)$	$\cot(X - 3) + 3$	$\sin(X) + 2$
9	$\sinh(X + 1) - 2$	$\csc(X + 2) - 3$	$\text{sinc}(X) - 3$

Продолжение таблицы 1.7

1	2	3	4
10	$\tan(X) + 3$	$\sin(X) + 4$	$\cot(X + 2) + 4$
11	$\sqrt{X - 3} + 1$	$\sec(X+1)$	$\csc(X - 1) + 3$
12	$\sin(X - 4)$	$\cot(X + 3) - 1$	$\sinh(X + 2) - 2$
13	$\text{sinc}(X + 2)$	$\csc(X + 2) + 2$	$\sec(X + 1)$
14	$\cot(X + 1) + 1$	$\sin(X - 2)$	$(X^2) - 3$
15	$\csc(X - 2) + 3$	$\sec(X + 1)$	$\tan(X - 2) + 3$
16	$(X^3) - 5$	$\cot(X + 4) + 5$	$\sin(X + 3)$
17	$\sec(X + 1)$	$\sinh(X) + 1$	$17 - X^3$
18	$(X^2) + 5$	$\sin(X - 2) + 3$	$\cot(X + 5) + 3$
19	$\sinh(X - 1) + 4$	$\sec(X + 1)$	$\csc(X - 1) + 2$
20	$\sin(X - 3) + 2$	$\cot(X + 2) + 4$	$\sqrt{X + 2} + 2$
21	$\tan(X + 1) - 3$	$\csc(X + 4) - 2$	$\sec(X + 1)$
22	$\cot(X + 3) - 2$	$\sin(X + 2) - 1$	$\sinh(5 - X) + 2$
23	$\csc(X - 3) + 2$	$\sec(X + 1)$	$\text{sinc}(X - 3) + 2$
24	$\sec(X + 1)$	$\cot(X + 2) - 3$	$\sin(X + 4) - 2$
25	$\sinh(X - 4) + 2$	$\csc(X - 2) + 5$	$\tan(X - 2) - 4$
26	$20 + X^2$	$\sec(X + 1)$	$\cot(X - 3) + 3$
27	$\sin(X - 2) + 4$	$\sinh(X - 3) + 2$	$\csc(X + 2) - 5$
28	$\tan(X + 1) + 2$	$\cot(X - 1) + 2$	$\sec(X + 1)$
29	$\sqrt{X} - 3$	$\csc(X + 3) - 5$	$\sin(X - 1) + 1$
30	$\cot(X - 1) + 2$	$\sec(X + 1)$	$\sinh(X - 3) + 5$

1.4 Содержание отчета

В рамках лабораторной работы № 1 необходимо выполнить три индивидуальных задания. В связи с этим отчет по лабораторной работе должен содержать:

- 1) титульный лист;
- 2) введение:
 - 2.1) цель выполнения задания;
 - 2.2) краткое описание задания;
- 3) основная часть:
 - 3.1) ход выполнения задания;
 - 3.2) обработка и анализ результатов выполнения задания;
- 4) выводы по результатам выполнения заданий.

1.5 Контрольные вопросы

- 1 Какие команды используются в среде *LabVIEW* для отладки программ?
- 2 Имеет ли какое-либо значение порядок подключения проводников к элементам суммирования и вычитания, умножения и деления?
- 3 Какая форма графического представления результатов работы программы в наибольшей мере отражает дискретный принцип работы персонального компьютера?

- 4 В каких случаях целесообразнее использовать графики?
- 5 Как можно отобразить «*Tools Palette*» для работы с «*Block Diagram*»?
- 6 Какие существуют типы данных для комплексных чисел в *LabVIEW*?
- 7 С помощью какого инструмента создаются проводники данных на блок-диаграмме?
- 8 Какие виды запуска программы вы знаете?
- 9 Для чего предназначены окна «*Front Panel*» и «*Block Diagram*»?
- 10 Для чего предназначены функции «*Edit*» и «*Tools*» на панели главного меню?
- 11 Откройте на «*Block Diagram*» палитру «*Functions*» и запишите в отчет ее основные элементы.
- 12 Каковы отличительные черты интерфейса среды?
- 13 В чем заключаются принципиальные отличия разработки программ в графической и текстовой средах программирования?
- 14 Какими способами можно увеличить число каналов компьютерного осциллографа в *LabVIEW*?
- 15 Какие инструменты служат для отладки программ и обнаружения ошибок?
- 16 Для каких целей используется режим анимации потоков данных?
- 17 Чем отличается работа узлов и индикаторов, расположенных внутри и снаружи цикла?
- 18 Чем отличаются циклы *For* и *While*?
- 19 В каких случаях применяется функция задержки времени?
- 20 Объясните, почему *LabVIEW* называют графической средой программирования.
- 21 Перечислите структуры данных, которые можно использовать в программе *LabVIEW*.

Вспомогательная литература

- 1 Трэвис, Д. *LabVIEW для всех* / Д. Трэвис, Д. Кринг. – 4-е изд., перераб. и доп. – М. : ДМК Пресс, 2015. – 904 с.
- 2 Пилипенко, О. В. Основы программирования, математического моделирования и обработки данных в среде *LabVIEW*: практикум / О. В. Пилипенко, Н. Б. Горбачёв, М. А. Музалевская. – Орел : ОрелГТУ, 2008. – 70 с.
- 3 Автоматизация физических исследований и эксперимента: компьютерные измерения и виртуальные приборы на основе *LabVIEW 7* (30 лекций) / П. А. Бутыркин [и др.]. – М. : ДМК Пресс, 2005. – 264 с.
- 4 Васильев, А. С. Основы программирования в среде *LabVIEW* / А. С. Васильев, О. Ю. Лашманов. – СПб. : Университет ИТМО, 2015. – 82 с.
- 5 Евдокимов, Ю. К. *LabVIEW для радиоинженера: от виртуальной модели до реального прибора* / Ю. К. Евдокимов, В. Р. Линдваль, Г. И. Щербаков. – М. : ДМК Пресс, 2007. – 400 с.

Лабораторная работа № 2
**Моделирование политропного процесса сжатия воздуха
в изделии цилиндрической формы
в инженерной среде *LabVIEW***

Цель работы: осуществить в инженерной среде *LabVIEW* моделирование термодинамического процесса, во время которого теплоемкость газа остается неизменной, с учетом сжатия воздуха в изделии цилиндрической формы.

Основные задачи:

- 1 Изучить возможности среды для математического моделирования физических явлений и процессов, создания подпрограмм обработки данных.
- 2 Научиться моделировать политропный процесс сжатия воздуха в цилиндре путем выполнения индивидуального задания.

2.1 Теоретические сведения

В настоящее время решение большинства научно-технических задач базируется на использовании компьютерного моделирования, когда этапам проведения физических экспериментов, разработки новых изделий и технологий предшествуют сложные математические расчеты, создание имитационных моделей и т. п.

В качестве примера в настоящем учебно-методическом пособии рассматриваются возможности инженерной среды графического программирования для моделирования термодинамических процессов идеального газа. Процессом называется любое изменение параметров его состояния. Обычно изменяются все три параметра, связанные между собой уравнением $PV = RT$. Существует ряд процессов, в течение которых сохраняется постоянное отношение выполненной работы и количества тепла, участвующего в теплообмене с внешней средой. Такие процессы называются политропными. Для них выполняется дополнительное соотношение

$$PV^n = \text{const}, \quad (2.1)$$

где P – давление, Па;
 V – объем газа, л;
 n – показатель политропы.

Если в политропном процессе воздух, являющийся идеальным газом, сжимается очень быстро, то при уменьшении объема в 15 раз температура его повышается до 650 °С. В него можно впрыснуть дизельное топливо, и оно самовоспламеняется. Таким способом может быть реализован один из процессов, с помощью которого приводится в действие дизельный двигатель.

При той же степени сжатия, осуществляемого очень медленно, температура остается без изменений. Это связано с тем, что в медленном процессе тепловая энергия, которая образуется при сжатии газа (воздуха), успевает рассеяться в окружающей среде. Таким образом, характер изменения параметров состояния фактически зависит от скорости процесса. В первом случае показатель политропы n равен коэффициенту адиабаты – $n = 1,4$ (адиабатический процесс), во втором – $n = 1$ (изотермический процесс), в третьем – $n = 0$ (изобарный процесс) и в четвертом – $n = \infty$ (изохорный процесс)

Моделирование такого рода ситуаций необходимо на производствах, где используются компрессоры, так как изменения давления и объема газа в них может быть описано уравнением политропы. Использование политропы помогает инженерам и проектировщикам более точно моделировать и управлять процессами сжатия газа в различных технических системах.

2.2 Задание для выполнения лабораторной работы

Задание. Разработать программу моделирования политропного процесса сжатия воздуха в цилиндре объемом V_0 с начальным давлением P_0 и температурой T_0 при степени сжатия $\lambda = \frac{V_0}{V_k} = 5$. Результаты вычислений отобразить в виде индикаторов традиционных приборов, служащих для измерения параметров состояния V , P , T , графиков их изменения по времени и P - V диаграммы исследованного процесса.

Порядок выполнения задания.

Осуществите запуск среды *LabVIEW*. В появившемся главном окне программы выберите команды: «*File*» → «*New*» для создания нового файла. Затем в появившемся окне «*New*» выберите шаблон «*Blank VI*».

Создайте на «*Front Panel*» четыре цифровых элемента управления для исходных данных задачи: V_0 , P_0 , T_0 и n (рисунок 2.1). Для этого щелчком правой кнопки мыши по серой панели вызовите «*Controls*» и закрепите ее, активизировав кнопку в левом верхнем углу палитры.

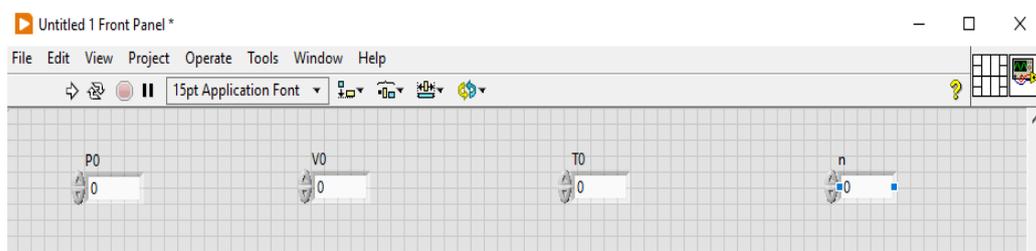


Рисунок 2.1 – Расположение цифровых элементов на «*Front Panel*»

Откройте меню «*Numeric*», выберите в нем первый элемент в верхнем ряду (рисунок 2.2). На открывшейся палитре «*Controls*» выделить элемент «*Numeric*

Control». Переместите четыре элемента поочередным перетаскиванием на «*Front Panel*» и расположите их горизонтально в одну строку.

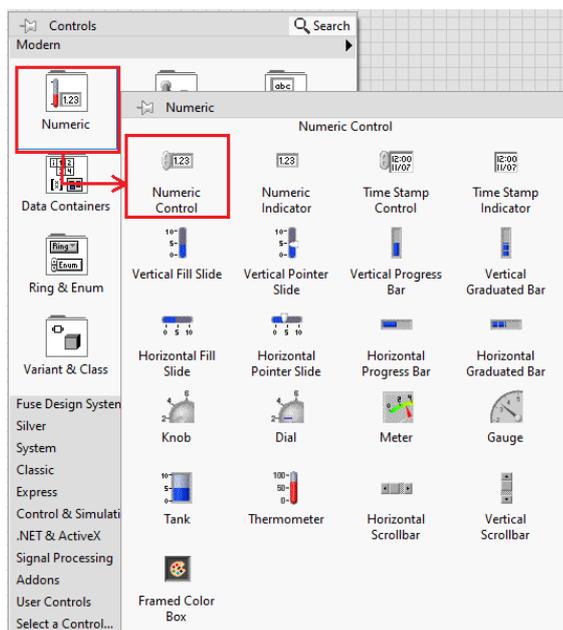


Рисунок 2.2 – Палитра «*Controls*»

Измените собственные метки управляющих элементов, подписав вместо «*Numeric*» новые обозначения: V_0 , P_0 , T_0 и n . Установите в окнах регуляторов соответствующие значения исходных данных ($n = 1$).

Для отображения полученных данных V , P , T создайте на «*Front Panel*» три прибора: мерную емкость – «*Tank*», манометр – «*Gauge*» и индикатор температуры – «*Thermometer*» (рисунок 2.3).

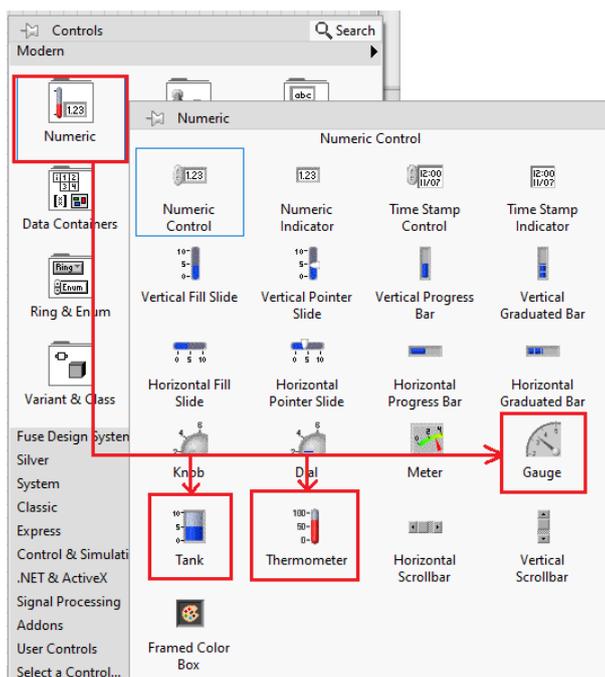


Рисунок 2.3 – Элементы «*Tank*», «*Gauge*» и «*Thermometer*»

Подпишите названия этих приборов и измените верхние пределы их шкал с помощью двойного щелчка левой кнопки мыши по наибольшему значению: для объема – 1000 мл, давления – 2000 кПа, температуры – 1000 К, как показано на рисунках 2.4 и 2.5.

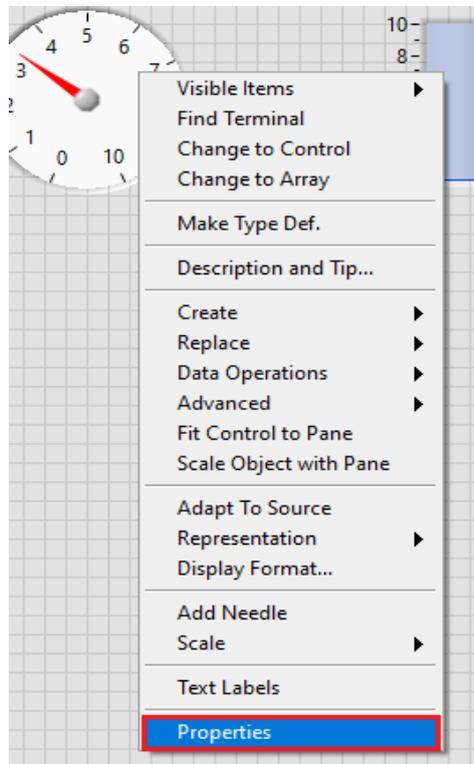


Рисунок 2.4 – Вызов контекстного меню элемента

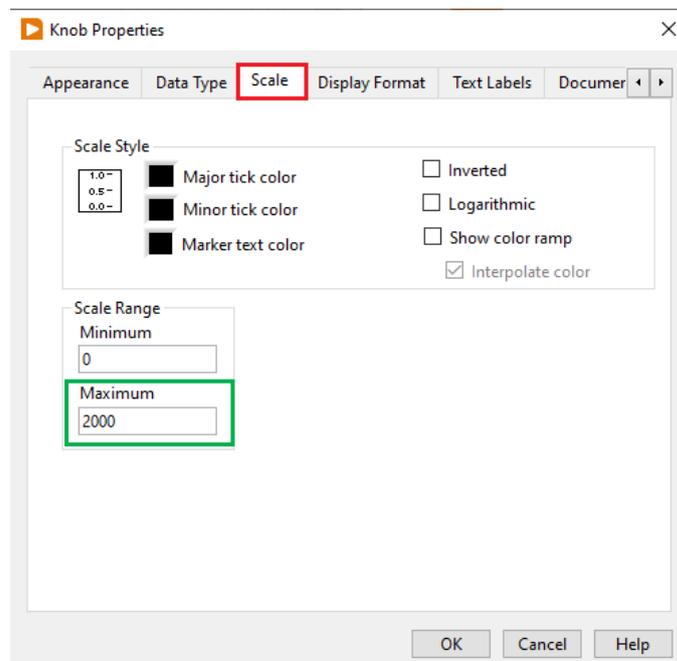


Рисунок 2.5 – Окно настроек элемента

Промежуточный вид лицевой панели имеет вид, представленный на рисунке 2.6.

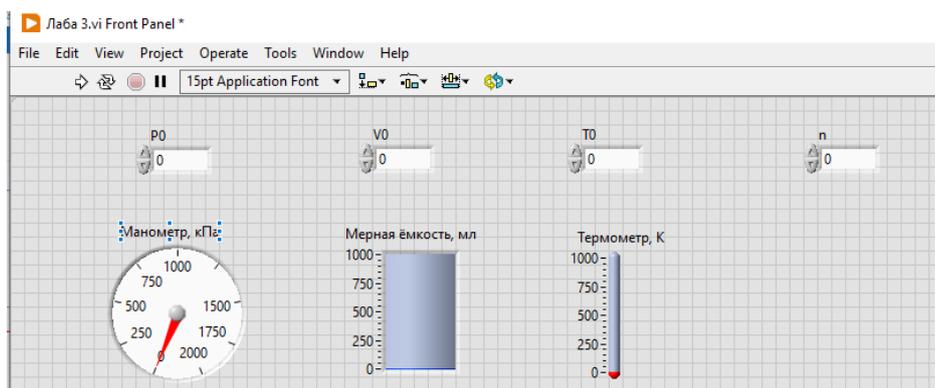


Рисунок 2.6 – Промежуточный вид «*Front Panel*»

Для наблюдения за ходом процесса создайте на «*Front Panel*» трехлучевой запоминающий осциллограф и XY-самописец для построения *P-V* диаграммы процесса – панель «*Controls*», откройте «*Graph*», из которых на «*Front Panel*» выносится первый («*Waveform Chart*») и четвертый элемент («*Ex XY Graph*») (рисунок 2.7).

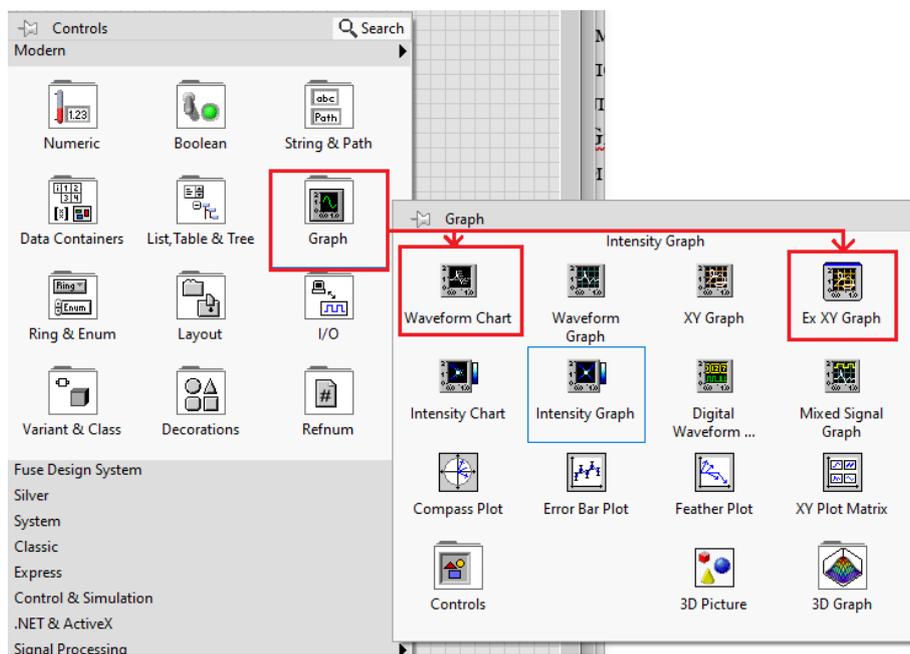


Рисунок 2.7 – Графические индикаторы

Нажмите правой кнопкой мыши на элемент «*Waveform Chart*» и выберите «*Properties*», после чего измените легенды шкал, как можно увидеть на рисунках 2.8 и 2.9.

То же самое сделайте для «*Ex XY Graph*», пользуясь рисунками 2.10, 2.11.

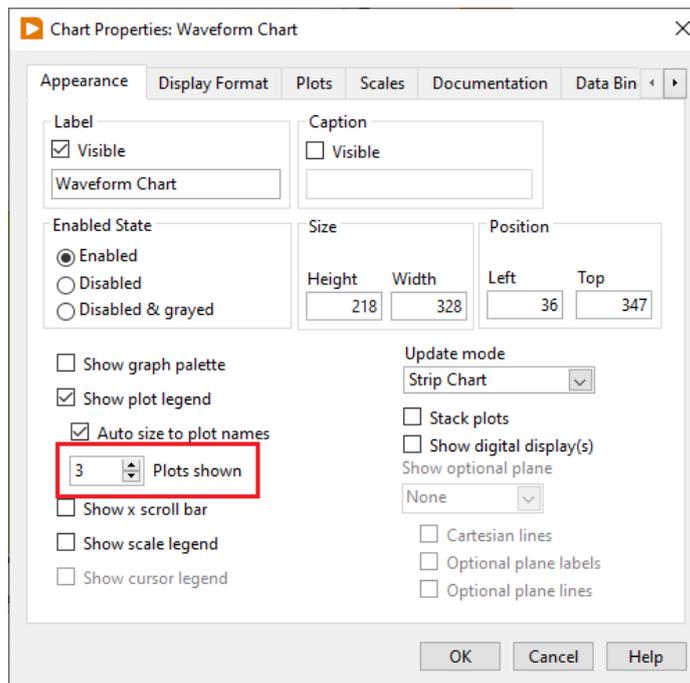


Рисунок 2.8 – Изменение количества отображаемых графиков в окне «*Chart Properties: Waveform Chart*»

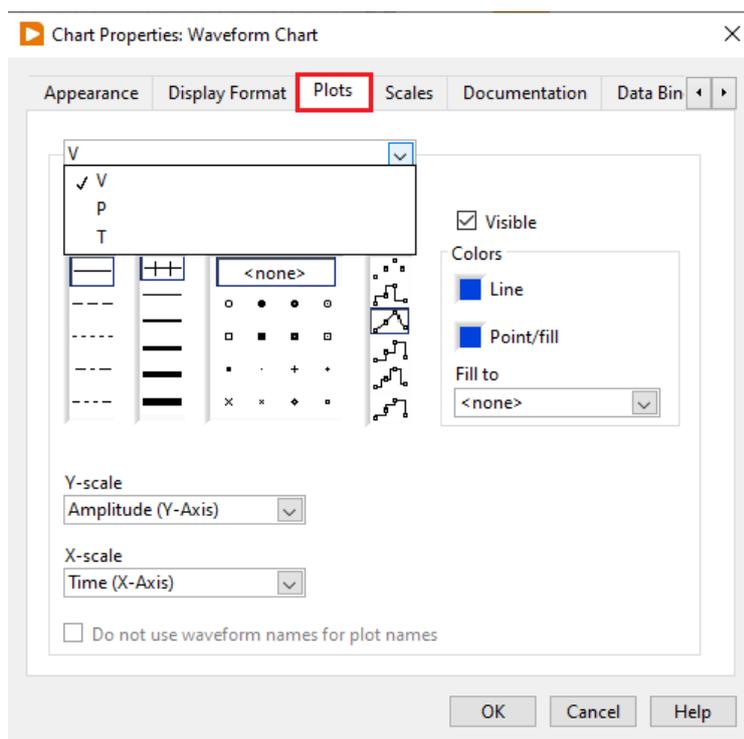


Рисунок 2.9 – Изменение названий графиков в окне «*Chart Properties: Waveform Chart*»

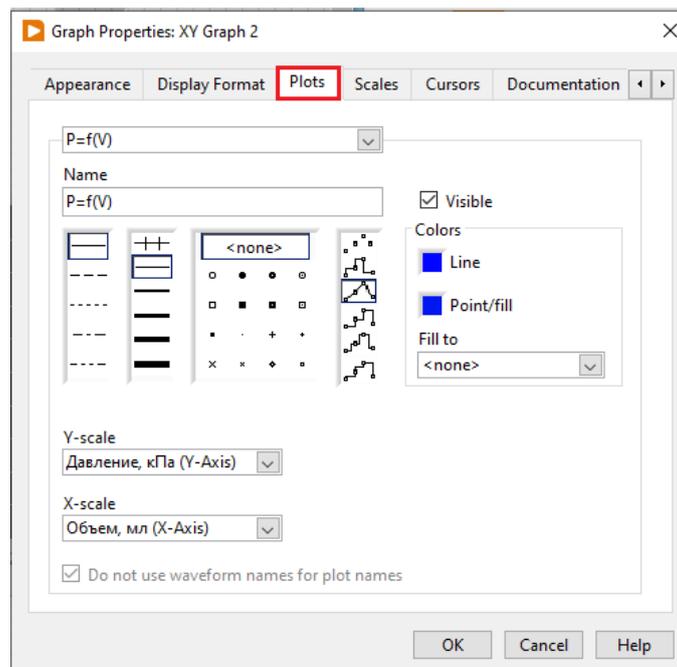


Рисунок 2.10 – Изменение названия графика в окне «*Chart Properties: Ex XY Graph 2*»

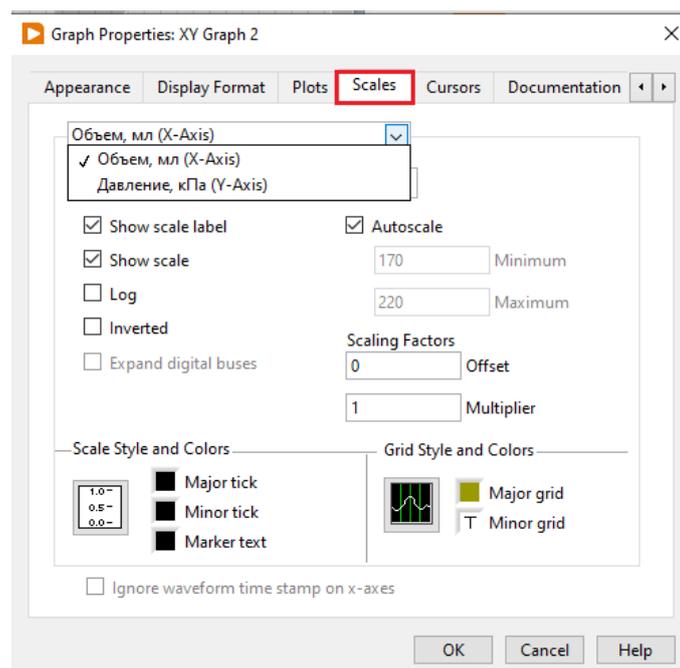


Рисунок 2.11 – Изменение названий переменных в окне «*Chart Properties: Ex XY Graph 2*»

Освободите среднюю часть «*Block Diagram*» для построения графического кода программы. Щелчком правой кнопки мыши на «*Block Diagram*» вызовите «*Functions*». Используя кнопку в верхнем левом углу палитры, зафиксируйте ее на экране.

В палитре всех функций вызовите элемент «*Structures*», далее в нем выбираем цикл по условию «*While Loop*», перетаскиваем его на «*Block Diagram*» и растягиваем на большую часть экрана. Вернитесь к «*Structures*», выберите формульный узел «*Formula Node*» и перенесите его внутрь цикла (рисунок 2.12).

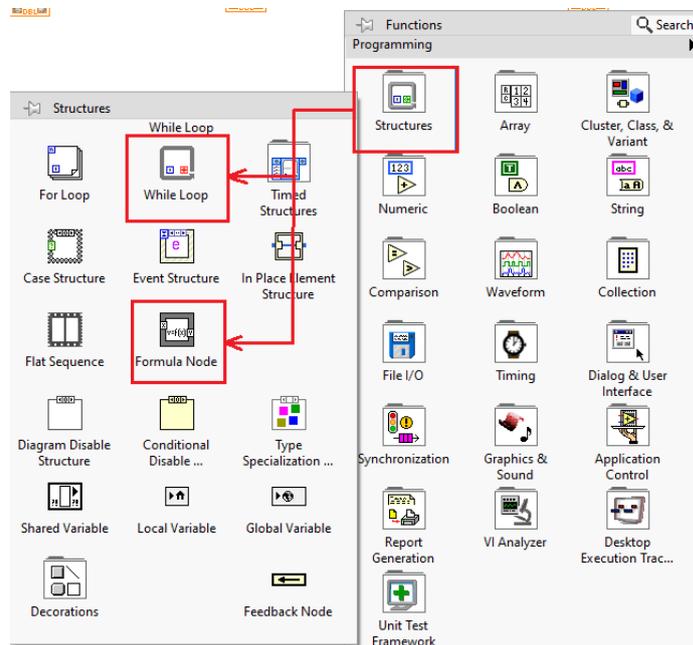


Рисунок 2.12 – Элементы «*While Loop*» и «*Formula Node*»

В нашем случае определяющим процессом в моделировании является движение поршня и соответствующее изменение объема сжатого воздуха. Свяжем скорость этого процесса с показателем политропы по формуле

$$V = V_0 - 0,002 \cdot k \cdot n^5, \quad (2.2)$$

где V_0 – начальный объем, л;
 $k = i$.

Таким образом, при моделировании время процесса принимается равным значению i – номеру текущей итерации цикла. Это означает, что за каждый цикл моделирования процесса объем сжимаемого газа линейно уменьшается на $(2 \times n^5)$ мл. Введем эту формулу в подготовленный узел и создадим на его границе три терминала ввода, необходимых для расчета V данных – V_0 , k и n , а также один терминал для вывода V .

Для этого щелчком правой кнопки мыши на границе узла вызываем всплывающее меню, в нем – три раза «*Add Input*» и один раз «*Add Output*», как показано на рисунке 2.13.

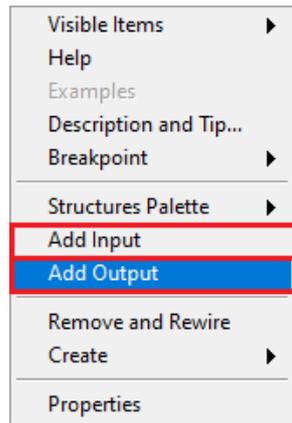


Рисунок 2.13 – Всплывающее меню элемента «*Formula Node*»

Впишем в появившиеся терминалы буквенные обозначения переменных и соединим их соответственно с пиктограммами V_0 , n и узлом счетчика итераций, считая, что $k = i$ (рисунок 2.14).

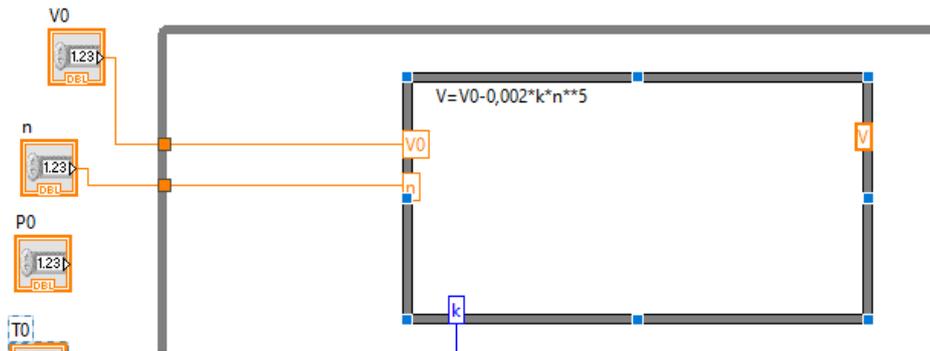


Рисунок 2.14 – Фрагмент диалогового окна «*Formula Node*»

Выведем полученное значение V на границу цикла, внесем в цикл пиктограмму мерной емкости, осциллограф и подадим значение V на их входы. Так как условие $\lambda = \frac{V_0}{V_k} = 5$ (V_k меньше или равно 0,2 л) используется для завершения цикла, находим в «*Functions*» подпалитру «*Comparison*», в ней узел «*Less Or Equal?*» (рисунок 2.15).

Подводим текущее значение V к верхнему терминалу логического узла, а к нижнему подводим константу 0,2. Логический результат (*False* или *True*) присоединяем к терминалу завершения цикла. Обратим внимание на цвет проводника, соответствующий логическому типу данных. Так как по умолчанию каждый цикл рассчитывается всего за 1 мс, то для отслеживания динамики процесса установим задержку цикла, равную 20 мс. Для этого на «*Functions*» выбираем пиктограмму прибора времени «*Timing*», а в ней выберем «метроном» («*Wait Until Next ms Multiply*») (рисунок 2.16).

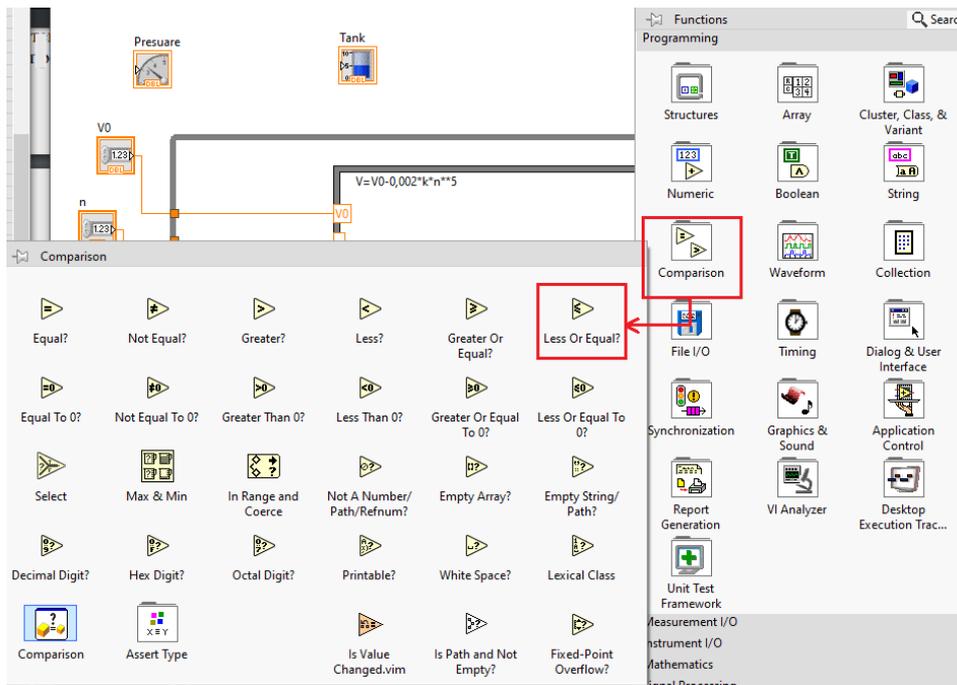


Рисунок 2.15 – Узел «Less Or Equal?» в палитре «Functions»

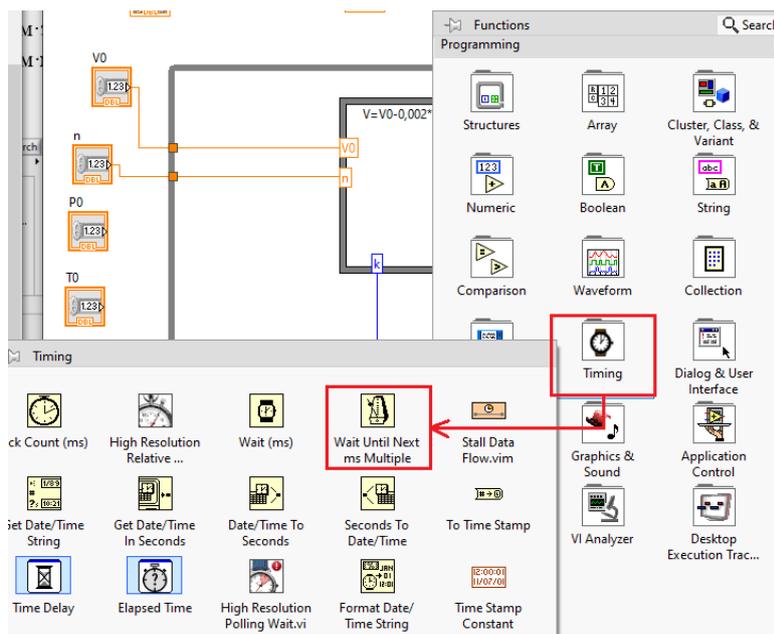


Рисунок 2.16 – Элемент «Wait Until Next ms Multiple» в палитре «Functions»

Помещаем его внутри цикла, находим входной терминал и щелчком правой кнопки мыши вызываем всплывающее меню, в нем «Functions» → «Programming» → «Numeric» → «Numeric Constant» (рисунок 2.17).

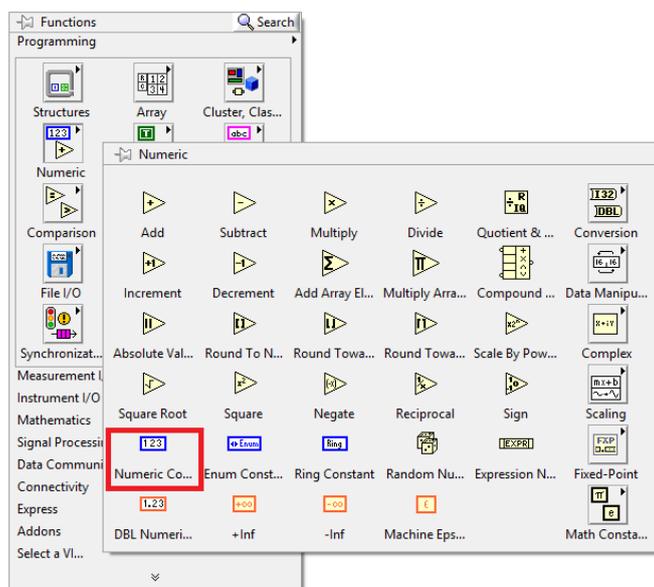


Рисунок 2.17 – Элемент «*Numeric Constant*»

В появившемся прямоугольнике с клавиатуры набираем число 20.

Убедитесь, что стрелка запуска цикла имеет правильную, не изломанную форму. Это означает, что программа составлена правильно и готова к запуску. В противном случае щелчком правой кнопки мыши по стрелке вызываем контекстное меню с распечаткой допущенных ошибок. Устраняем их и запускаем программу. При этом объем сжатого воздуха в мерной емкости за 10 – 15 с уменьшается от 1 до 0,2 л, а на осциллографе появляется график уменьшения объема в виде прямой наклонной линии. На этом создание и отладка программного управления изменением объема сжимаемого газа завершается.

Соответствующее рассмотренному процессу изменение давления в цилиндре описывается формулой

$$P = P_0 \cdot \left(\frac{V_0}{V}\right)^n, \quad (2.3)$$

где P_0 – начальное давление, Па.

Вносим это соотношение в формульный узел. Добавляем дополнительный вход для ввода P_0 и с помощью проводника связываем его с соответствующим элементом управления. На правой границе формульного узла создаем выход P и соединяем его с пиктограммой манометра, внесенной внутрь цикла.

Для одновременного отображения графиков изменения давления и объема воздуха на одном и том же приборе преобразуем однолучевой осциллограф в двухлучевой. Для этого выбираем «*Functions*» → «*Programming*» и открываем «*Cluster, Class & Variant*», а в ней пиктограмму «*Bundle*» (рисунок 2.18).

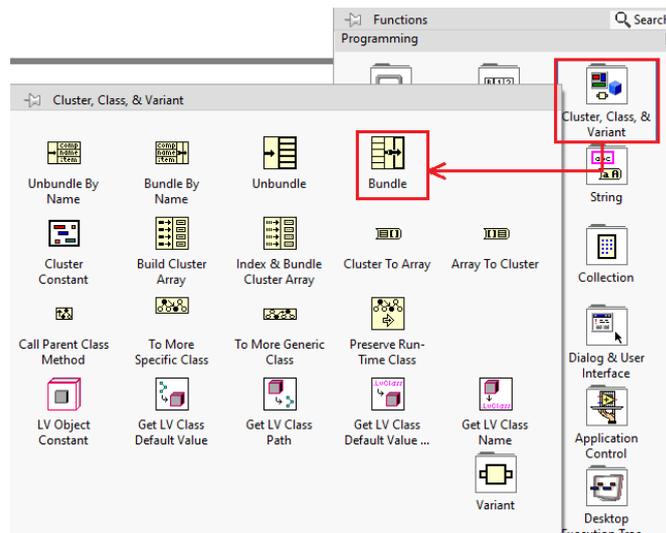


Рисунок 2.18 – Элемент «*Bundle*»

Активизируем проводник, соединяющий осциллограф с выходом V и удаляем его. К нижнему входу элемента «*Bundle*» подводим значение P , а у верхнего восстанавливаем соединение с V . Выход элемента «*Bundle*» соединяем с входом компьютерного осциллографа, который с этого момента становится двухлучевым. Элемент «*Ex XY Graph*» размещаем вне цикла *While*. Также подводим значения V и P к входам «*X Input*» и «*Y Input*» элемента «*Ex XY Graph*» соответственно. На границе цикла *While* появятся оранжевые квадраты. Кликаем по каждому из них правой кнопкой мыши, затем выбираем «*Tunnel Mode*» → «*Indexing*».

Работа по созданию подпрограммы моделирования изменения давления при сжатии может считаться завершённой, если стрелка запуска имеет правильный вид. Для проверки правильности работы этой программы необходимо очистить предыдущий график щелчком правой кнопки мыши по экрану осциллографа набором команд во всплывающем меню «*Data Operations*» → «*Clear Chart*» (рисунок 2.19).

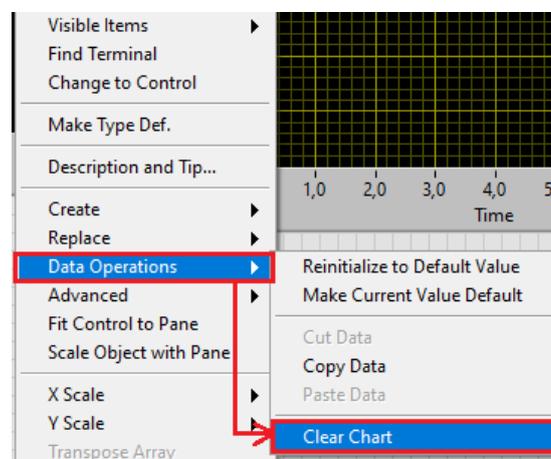


Рисунок 2.19 – Всплывающее меню «*Data Operations*»

Запустить программу и убедиться в том, что за время процесса давление в цилиндре изменяется от 100 до 500 кПа, а график его изменения по времени представляет собой возрастающую экспоненциальную функцию.

Изменение температуры в исследуемом процессе определяется соотношением $T = T_0 \cdot \left(\frac{P \cdot V}{P_0 \cdot V_0}\right)$. Введем его в формульный узел, добавим дополнительный вход « T_0 » и выход для полученного значения « T ». Соединим выход « T » и пиктограмму термометра, помещенную внутри цикла. Добавим еще один канал соединения с осциллографом. Для этого с помощью курсора в виде стрелки активизируем элемент «объединение» и растянем его вниз на одну новую позицию. Подведем к образовавшемуся новому входу сигнал « T » и соединим общий выход с осциллографом.

Далее необходимо убедиться в правильности составленной подпрограммы, для этого необходимо очистить прежний график и запустить программу целиком. При $n = 1$ показания термометра остаются на том же уровне, так как этот изотермический процесс характеризуется как раз постоянным значением температуры. Давление и объем изменяются, как и в предыдущем случае.

Окончательный вид программы для моделирования политропного процесса сжатия воздуха в изделии цилиндрической формы представлен на рисунках 2.20 и 2.21.

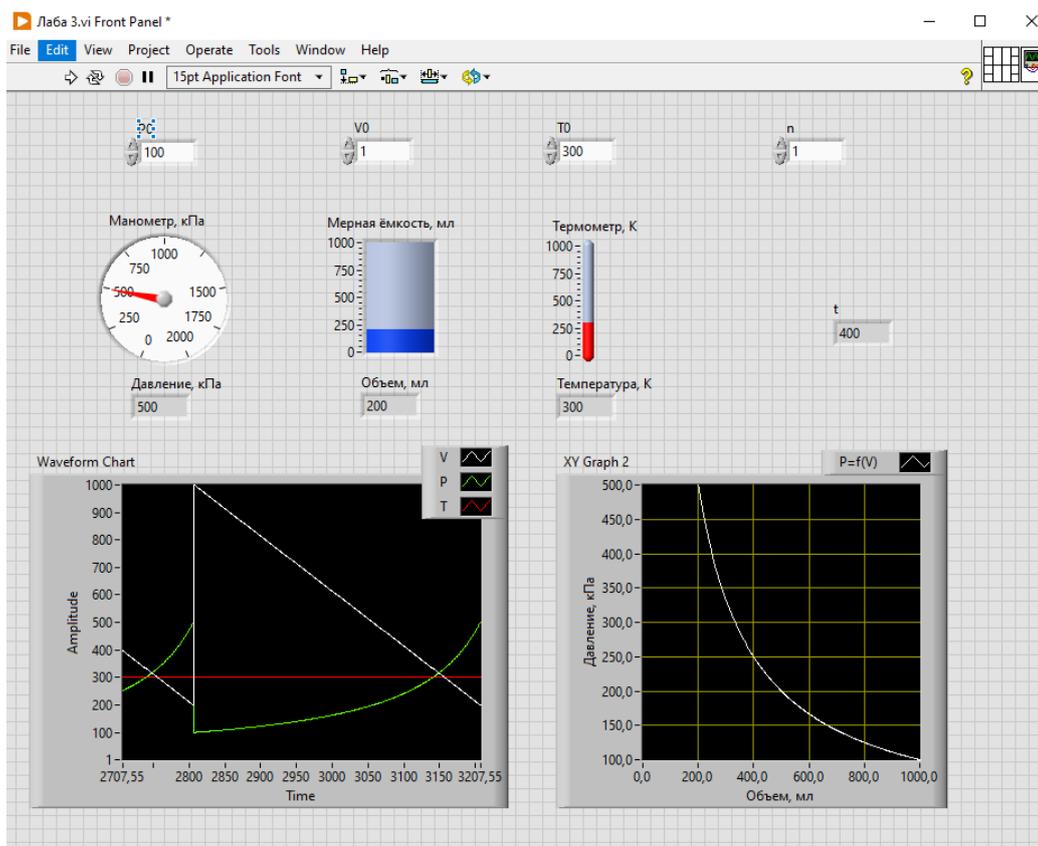


Рисунок 2.20 – Окончательный вид программы, окно «*Front Panel*»

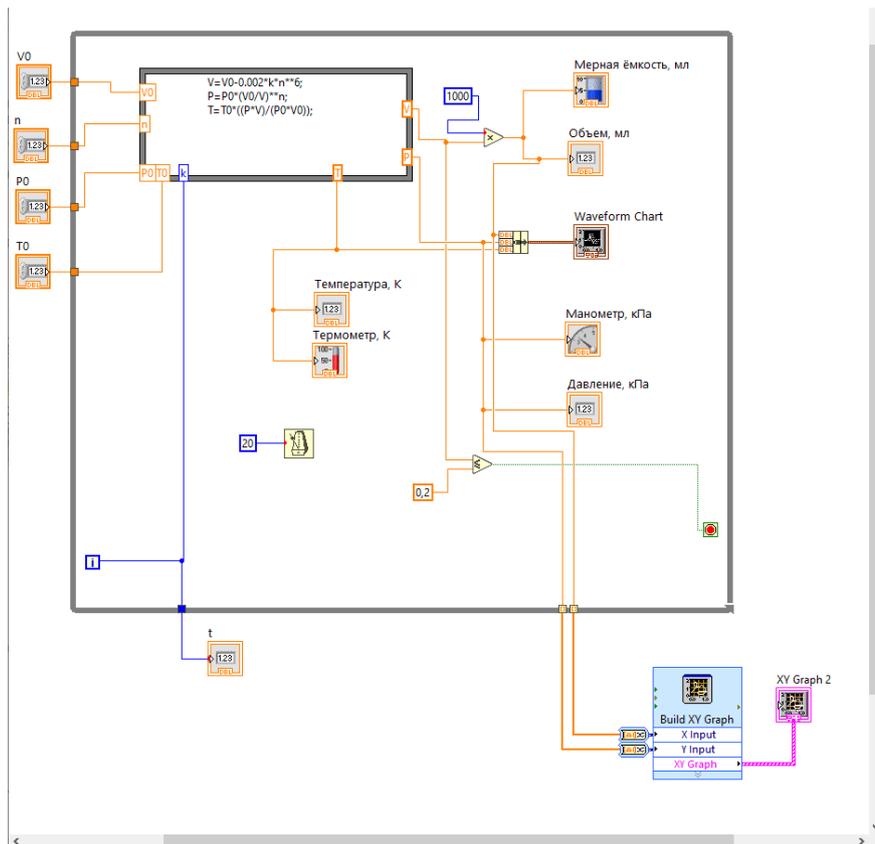


Рисунок 2.21 – Окончательный вид программы, окно «*Block Diagram*»

Предусмотрим вывод конечных значений параметров моделируемого процесса на лицевую панель работы. Для этого щелчком правой кнопки мыши по окну «*Front Panel*» необходимо добавить пять цифровых индикаторов («*Numeric Indicator*») и расположить их в следующей последовательности в соответствии с таблицей 2.1.

Таблица 2.1 – Цифровые индикаторы

n	t , мс	V_k , л	P_k , кПа	T_k , К
-----	----------	-----------	-------------	-----------

Расположите индикаторы на «*Block Diagram*» вне цикла, затем выведите значения соответствующих параметров на правую границу цикла и соедините их с соответствующими индикаторами, как показано на рисунке 2.22.

Повторите запуски программы при $n = 1,2$ и $n = 1,4$. Перенести измеренные значения на основе вашего варианта в таблицу 2.2.

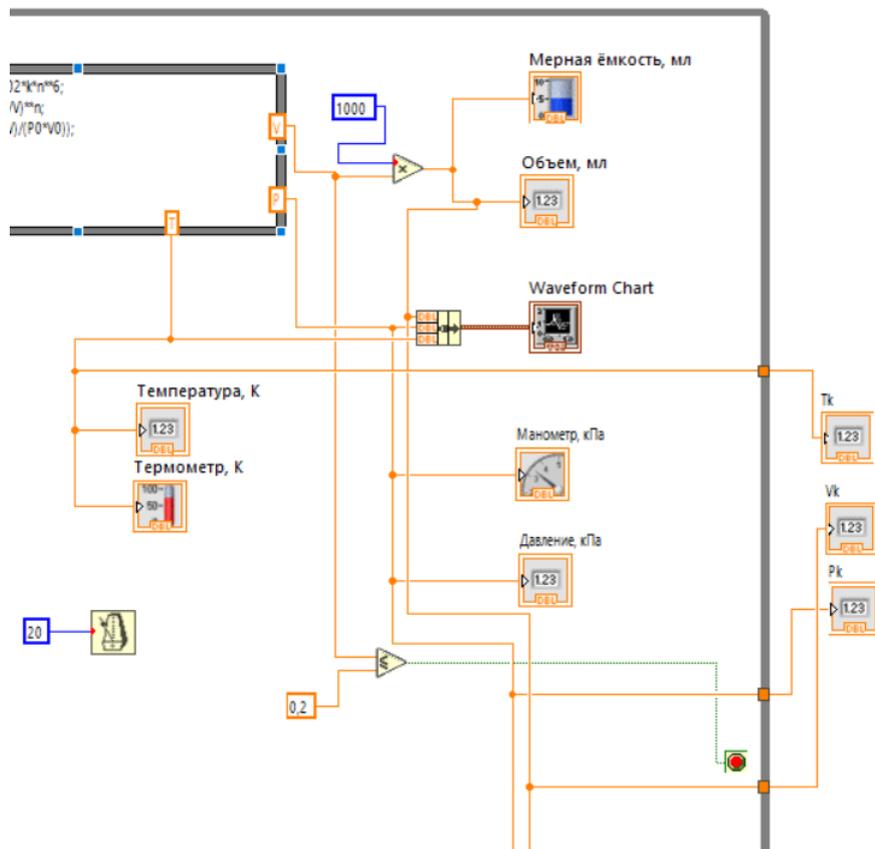


Рисунок 2.22 – Добавление дополнительных цифровых индикаторов на «Block Diagram»

Таблица 2.2 – Результаты моделирования

Параметры				
n	t , мс	V_k , л	P_k , кПа	T_k , К
1	400	0,2	500	300
1,2	134	0,199	690,9	414
1,4	54	0,186	1047	586,9

Полученные значения параметров состояния могут быть использованы для автоматического построения P - V диаграммы исследуемого процесса. Для этого необходимо вывести на границу цикла текущее значение P и V . По умолчанию выходные терминалы в цикле «*While Loop*» сохраняют только последние значения цикла. Для того чтобы при моделировании были бы сохранены все значения, необходимо для P и V создать еще по одному параллельному выходу и изменить их вид. Для этого щелчком правой кнопки мыши по терминалу вызвать контекстное меню и поменять выходы. Далее соединить выходные терминалы V и P соответственно с X - Y входами двухкоординатного самописца. Обратите внимание, что толщина проводников для массивов чисел, передаваемых из выходных терминалов цикла, больше, чем у проводников одиночных скалярных величин.

2.3 Индивидуальные задания

В рамках лабораторной работы предусмотрено выполнение заданий в соответствии с индивидуальным номером зачетной книжки каждого студента. При этом важно учесть, что повторение заданий не допускается, обеспечивая таким образом самостоятельное выполнение своей работы. В случае совпадения номера зачетной книжки с уже занятым заданием студенту предоставляется возможность выбрать свободный номер из предложенных вариантов.

Варианты заданий представлены в таблице 2.3.

Таблица 2.3 – Варианты индивидуальных заданий

Вариант	P_0 , кПа	V_0 , л	T_0 , К	Вариант	P_0 , кПа	V_0 , л	T_0 , К
1	100	1	300	16	169	2	381
2	120	2	334	17	95	3	340
3	142	3	347	18	112	2	346
4	178	1	381	19	151	3	340
5	158	1	341	20	170	1	355
6	145	2	386	21	172	1	326
7	161	1	314	22	122	2	360
8	147	3	299	23	115	1	344
9	193	1	333	24	186	1	348
10	115	3	391	25	138	2	310
11	171	3	397	26	160	3	391
12	164	2	366	27	131	3	332
13	127	2	341	28	133	1	400
14	165	1	347	29	183	2	366
15	149	1	321	30	96	3	366

2.4 Содержание отчета

В рамках лабораторной работы № 1 необходимо выполнить три индивидуальных задания. В связи с этим отчет по лабораторной работе должен содержать:

- 1) титульный лист;
- 2) введение:
 - 2.1) цель выполнения задания;
 - 2.2) краткое описание задания;
- 3) основная часть:
 - 3.1) ход выполнения задания;
 - 3.2) обработка и анализ результатов выполнения задания;
- 4) выводы по результатам выполнения заданий.

2.5 Контрольные вопросы

- 1 Приведите примеры других задач, которые могут решаться с помощью компьютерного моделирования.
- 2 Объясните, как работает цикл *While*.
- 3 Для чего на границах цикла формируются массивы данных *P* и *V*?
- 4 Что дает представление полученных зависимостей в виде площади под кривой процесса?
- 5 Как можно изменить количество точек, отображаемых на графике?
- 6 Назовите основные типы данных, использованных при выполнении задания.
- 7 Каким образом можно изменить скорость сжатия воздуха?
- 8 Почему для моделирования политропного процесса сжатия воздуха необходимо использовать показатель политропы?
- 9 Какие преимущества дает использование компьютерного моделирования по сравнению с проведением реальных экспериментов?
- 10 Каким образом можно изменить вид *P-V* диаграммы?

Вспомогательная литература

- 1 Трэвис, Д. *LabVIEW* для всех / Д. Трэвис, Д. Кринг. – 4-е изд., перераб. и доп. – М. : ДМК Пресс, 2015. – 904 с.
- 2 Пилипенко, О. В. Основы программирования, математического моделирования и обработки данных в среде *LabVIEW* : практикум / О. В. Пилипенко, Н. Б. Горбачёв, М. А. Музалевская. – Орел : ОрелГТУ, 2008. – 70 с.
- 3 Корниенко, В. Т. Обеспечение безопасности передачи информации в радиотехнических системах с примерами в проектах *LabVIEW* : учеб. пособие / В. Т. Корниенко. – Таганрог : ЮФУ, 2016. – 80 с.
- 4 Федосов, В. П. Цифровая обработка сигналов в *LabVIEW* / В. П. Федосов, А. К. Нестеренко. – М. : ДМК Пресс, 2015. – 456 с.
- 5 Евдокимов, Ю. К. *LabVIEW* в научных исследованиях / Ю. К. Евдокимов, В. Р. Линдваль, Г. И. Щербаков. – М. : ДМК Пресс, 2018. – 400 с.
- 6 *LabVIEW*: Практикум по основам измерительных технологий / В. К. Батоврин [и др.]. – М. : ДМК Пресс, 2005. – 208 с.

Лабораторная работа № 3
**Автоматизация экспериментальных исследований
по определению электрической емкости конденсатора
в программной среде *LabVIEW***

Цель работы: методом суммирования количества зарядов, стекающих с его обкладок, определить емкость конденсатора при помощи разработанной подпрограммы автоматизации обработки данных физического эксперимента.

Основные задачи:

- 1 Приобрести навыки работы с автоматизированными системами сбора экспериментальных данных.
- 2 Научиться использовать формульные узлы и сдвиговые регистры для обработки данных в режиме реального времени.
- 3 Ознакомиться с методикой моделирования процесса разрядки конденсаторов.

3.1 Теоретические сведения

Электрическая емкость уединенного проводника – физическая скалярная величина, количественно характеризующая способность проводника накапливать электрический заряд и равная отношению заряда проводника к его потенциалу. Единица электрической емкости является фарад (Ф).

Увеличить величину накапливаемого электрического заряда, а следовательно, и энергию электростатического поля можно использованием системы заряженных проводников, представляющих собой плоские пластины, концентрические цилиндрические или сферические поверхности.

Электрическая емкость конденсатора – это физическая величина, равная отношению заряда на одной из его поверхностей к разности потенциалов между соседними поверхностями, определяющаяся по формуле

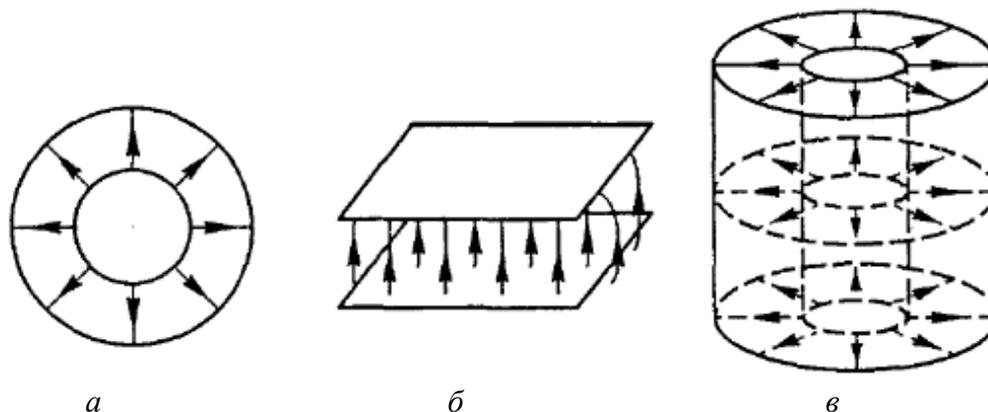
$$C = \frac{Q}{U}, \quad (3.1)$$

где Q – величина заряда, Кл;
 U – разность потенциалов, В.

Две проводящие поверхности, образующие конденсатор, называются его *обкладками*. При равных по модулю и противоположных по знаку зарядов этих поверхностей электрическое поле конденсатора практически целиком заключено в пространстве между его обкладками. Чтобы на емкость конденсатора не оказывали влияние окружающие тела, его обкладкам придают такую форму, при которой поле, создаваемое разноименными зарядами обкладок, оказывается сосредоточенным между ними. Этому условию идеально удовлетворяют две

концентрические сферы (сферический конденсатор), почти идеально – два коаксиальных цилиндра, ширина зазора между которыми существенно меньше их длины (цилиндрический конденсатор), и две близко расположенные пластины, размеры которых существенно превышают ширину зазора между ними (плоский конденсатор).

Характер силовых линий в конденсаторах различного типа показан на рисунке 3.1.



a – сферический; *б* – плоский; *в* – цилиндрический

Рисунок 3.1 – Электрическое поле в различных видах конденсаторов

Величину заряда, находящегося на обкладках конденсатора, можно вычислить, измеряя ток при разрядке конденсатора, формула (3.2). Действительно, сила тока характеризуется количеством элементарных зарядов, протекающих через проводник в единицу времени.

$$Q = n \cdot e, \quad (3.2)$$

где n – количество элементарных зарядов;

e – модуль элементарного электрического заряда, равный $1,6 \times 10^{-19}$ Кл.

Суммируя произведения силы тока за малые интервалы времени за все время разрядки, можно получить величину заряда, находившуюся на обкладках конденсатора, по формуле

$$Q = I_1 \cdot \Delta t_1 + I_2 \cdot \Delta t_2 + \dots + I_n \cdot \Delta t_n, \quad (3.3)$$

где Q – величина заряда, Кл;

I – сила тока, А;

Δt_n – интервал времени, с.

Разделив полученную величину на начальное напряжение на конденсаторе, получим значение электроемкости конденсатора.

3.2 Задание для выполнения лабораторной работы

Задание. Разработать подпрограмму автоматизации обработки данных физического эксперимента по определению электрической емкости конденсатора методом суммирования количества зарядов, стекающих с его обкладок. Запустить программу и исследовать зависимость времени разрядки конденсатора при сопротивлении нагрузки 100, 500 и 1000 кОм.

Порядок выполнения задания

Осуществите запуск среды *LabVIEW*. В появившемся главном окне программы выберите команды «*File*» → «*New*» для создания нового файла. Затем в появившемся окне «*New*» выберите шаблон «*Blank VI*».

Создадим на «*Front Panel*» стрелочный прибор для контроля падения напряжения на конденсаторе; три цифровых элемента управления для ввода данных (ЭДС, сопротивление R и остаточное напряжение U_k) и три цифровых индикатора для отображения величины заряда Q , емкости конденсатора C и времени разрядки t (рисунок 3.2).

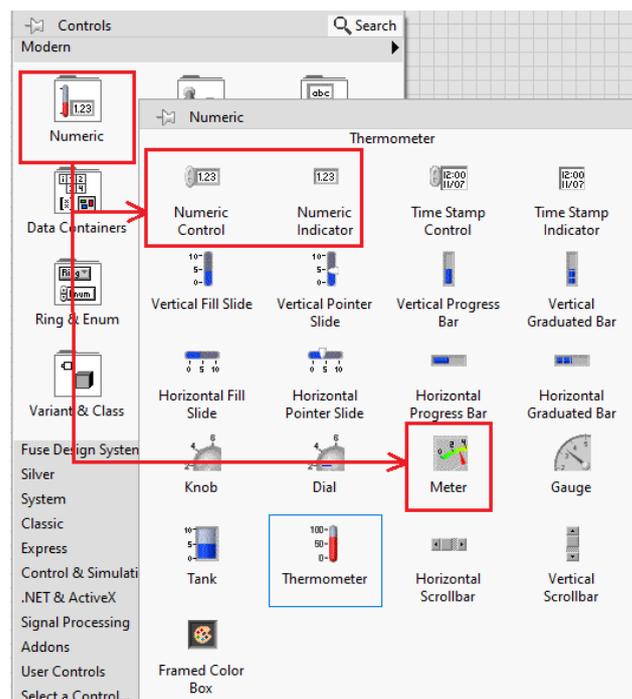


Рисунок 3.2 – Панели «*Controls*» и «*Numeric*»

Для наблюдения за процессом разрядки конденсатора поместим на «*Front Panel*» три осциллографа («*Waveform Chart*») для регистрации следующих текущих значений:

- напряжение на обкладках конденсатора;
- сила тока;
- количество зарядов, стекающих с конденсатора (рисунок 3.3).

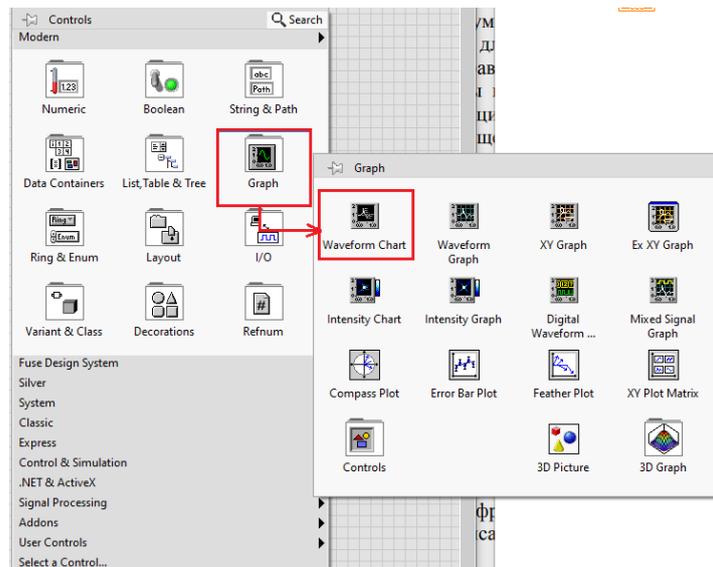


Рисунок 3.3 – Панели «*Controls*» и «*Graph*»

Изменяем собственные метки индикаторов и элементов контроля и вводим значения ЭДС $U = 3 \text{ В}$, $R = 1000 \text{ Ом}$, $U_k = 0,1 \text{ В}$.

Вызываем на «*Block Diagram*» элемент «*Functions*», в них «*Structures*», далее цикл «*While Loop*», переносим его пиктограмму на «*Block Diagram*» и растягиваем на большую часть экрана. Вновь возвращаемся в «*Structures*», активизируем узел «*Formula Node*» и переносим его внутрь цикла (рисунок 3.4).

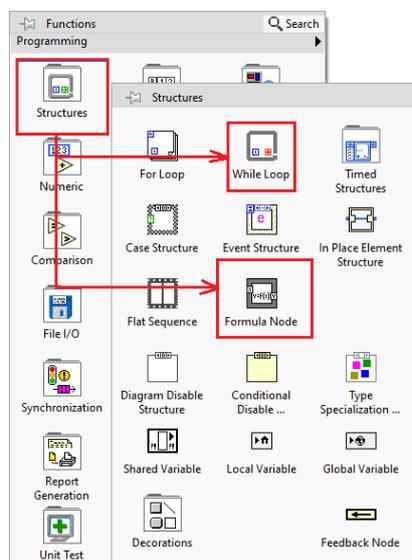


Рисунок 3.4 – Элементы «*While Loop*» и «*Formula Node*»

Вводим в узел формулы определения силы тока разрядки $I = \frac{U}{R}$ и суммирование величины зарядов, стекающих с обкладок, $Q = Q_0 + I \cdot dt$ (рисунок 3.5).

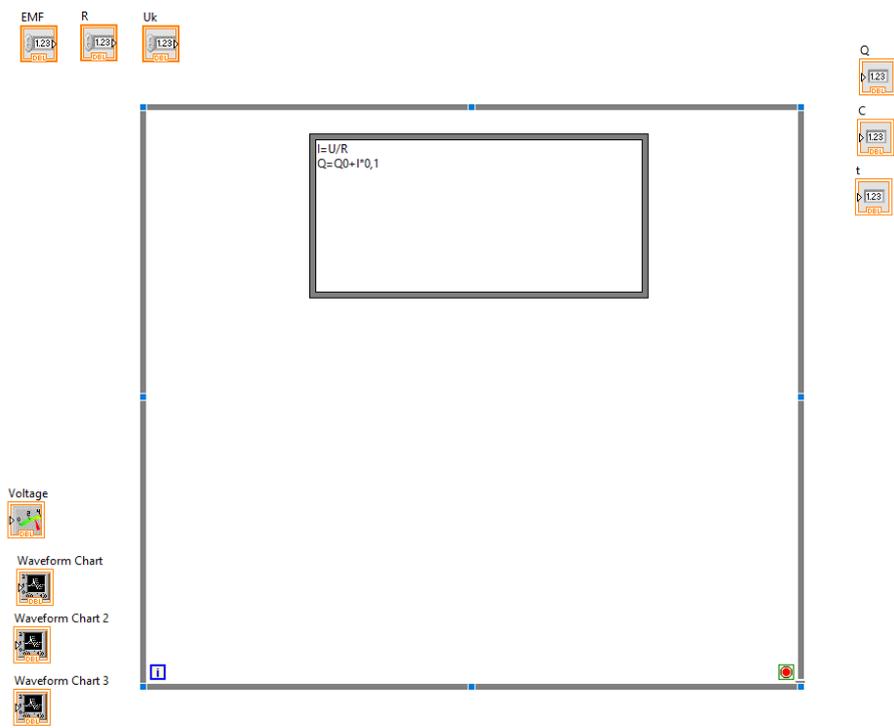


Рисунок 3.5 – Окно «Block Diagram» с заполненным формульным узлом

Вторая формула предусматривает организацию процесса численного интегрирования, в котором на каждой новой итерации используется предыдущее значение Q , которое в формуле всякий раз учитывается как новое значение Q_0 . Для запоминания вычисленного значения Q и возвращения его в цикл используется «сдвиговый регистр», который устанавливается щелчком правой кнопки мыши на правой границе цикла по команде «Add Shift Register» (рисунок 3.6).

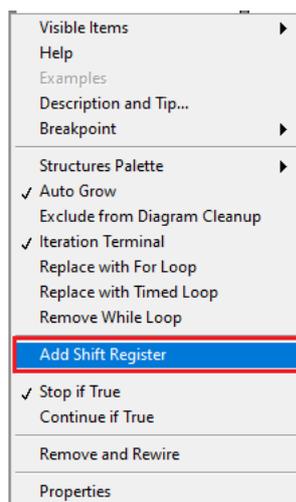


Рисунок 3.6 – Добавление сдвигового регистра

При этом на левой и правой границе появляются два терминала – регистры с разнонаправленными стрелками. Правый регистр указывает, что поступившие

значения выводятся за границы цикла, а левый – что выведенное значение вновь возвращается в цикл. Для исключения неопределенности при выполнении первой итерации необходимо задать начальное значение входного регистра равным нулю. Для этого щелчком правой кнопки мыши по нему вызвать всплывающее меню, в нем «Create» → «Constant», в появившемся прямоугольнике с клавиатуры наберите «0» (рисунок 3.7). Если не получается этим способом, то добавьте «Numeric Constant» из палитры «Functions», введите в ней «0» и соедините ее с входным регистром.

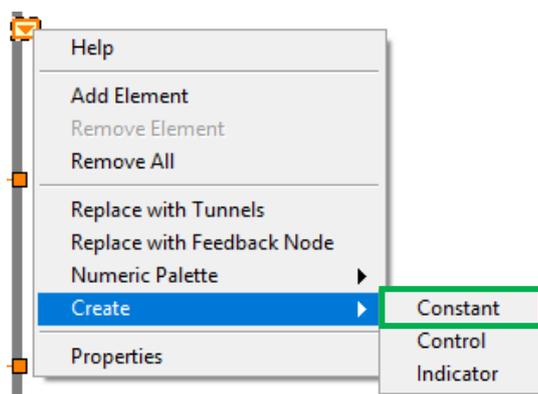


Рисунок 3.7 – Задание начального значения входного регистра

Для работы формульного узла необходимо ввести в него данные Q_0 , R и U . Щелчком правой кнопки мыши на границах узла создаем три входа («Add Input»): в « Q_0 » подводим значение с левого узла сдвигового регистра, а в « R » – значение сопротивления (рисунок 3.8).

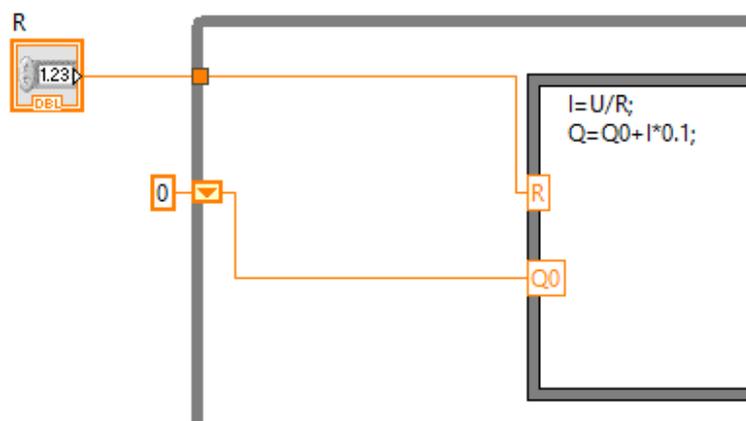


Рисунок 3.8 – Ввод данных в формульный узел в «Block Diagram»

В проводимом эксперименте сигнал, соответствующий напряжению на обкладках конденсатора, поступает с платы автоматизированного сбора данных, подключенной к лабораторной установке.

Для моделирования экспериментального сигнала из палитры арифметических функций выбираем пиктограмму экспоненциальной функции e^x и переносим ее внутрь цикла (рисунок 3.9).

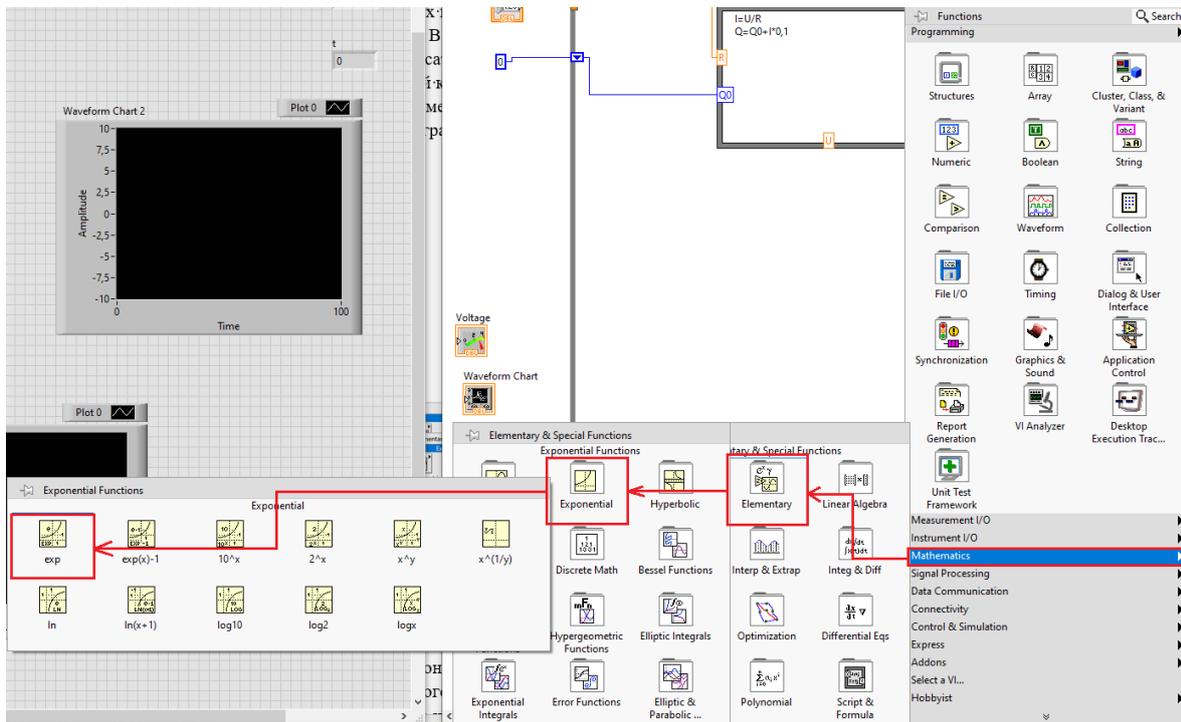


Рисунок 3.9 – Задание экспоненциальной функции

В качестве показателя экспоненциальной функции принимаем число «i», инвертированное до отрицательного числа и деленное на 10. Для этого воспользуемся инструментами «Negate», «Divide» и «Numeric Constant» из подпалитры «Numeric» (рисунок 3.10).

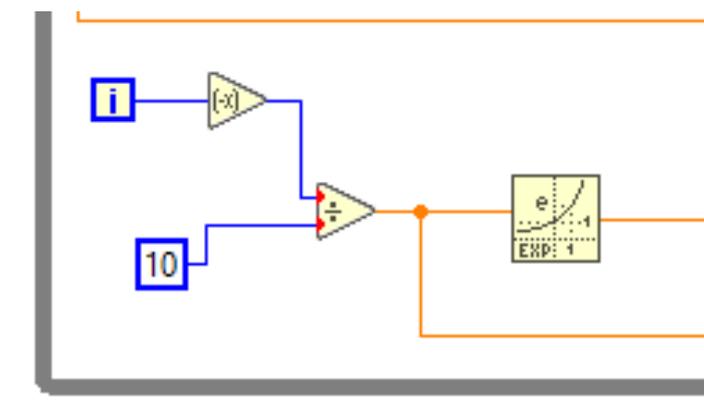


Рисунок 3.10 – Подключение показателя exp

Выход экспоненты умножаем (инструмент «*Multiply*» из подпалитры «*Numeric*») на величину ЭДС и вводим формульный узел в качестве текущего значения U , а также подключаем к первому осциллографу и стрелочному прибору.

Создаем на правой границе формульного узла два выхода: I и Q (рисунок 3.11), сигналы с которых вывести на входы второго и третьего осциллографов. Для обеспечения операции суммирования заряда продублировать соединение Q с правым регистром. То же выполнить для напряжения, подведя соответствующий сигнал к пиктограмме вольтметра и первому осциллографу.

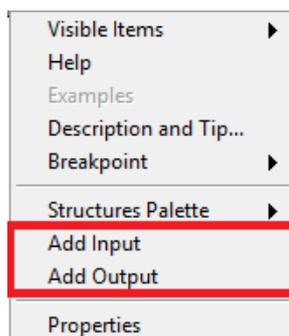


Рисунок 3.11 – Всплывающее меню элемента «*Formula Node*»

Окно «*Block Diagram*» на данном этапе будет иметь вид, представленный на рисунке 3.12.

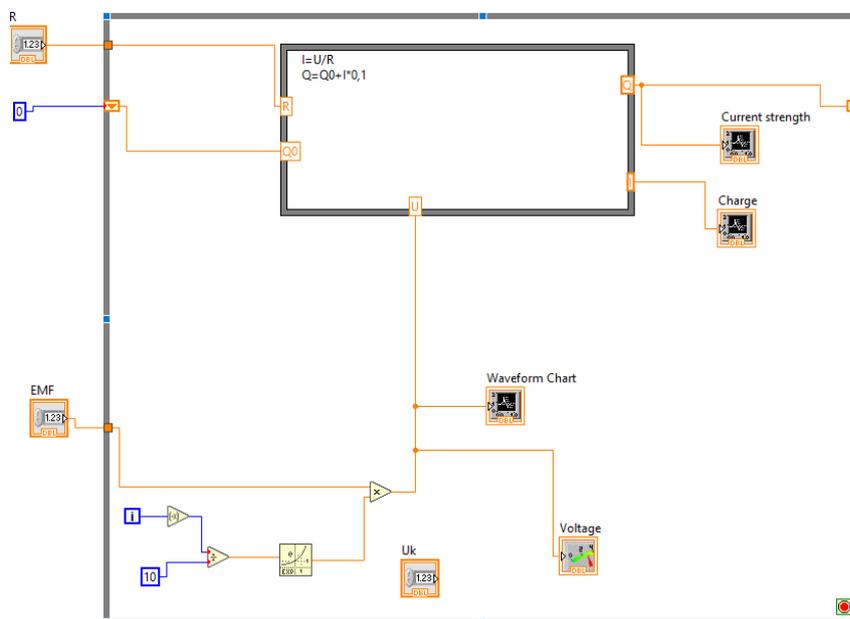


Рисунок 3.12 – Окно «*Block Diagram*»

Текущее значение напряжения U используется в качестве входного параметра для системы автоматического отключения работы цикла. Условие выключения – $U \leq U_k$. Для его реализации необходимо вызывать с палитры «*Functions*»

подпалитру «*Comparison*», в ней – узел «*Less Or Equal?*» (рисунок 3.13), и ввести на верхний терминал узла текущее значение U , а на нижнее – заданное значение остаточного напряжения U_k от элемента управления, созданного ранее на «*Front Panel*».

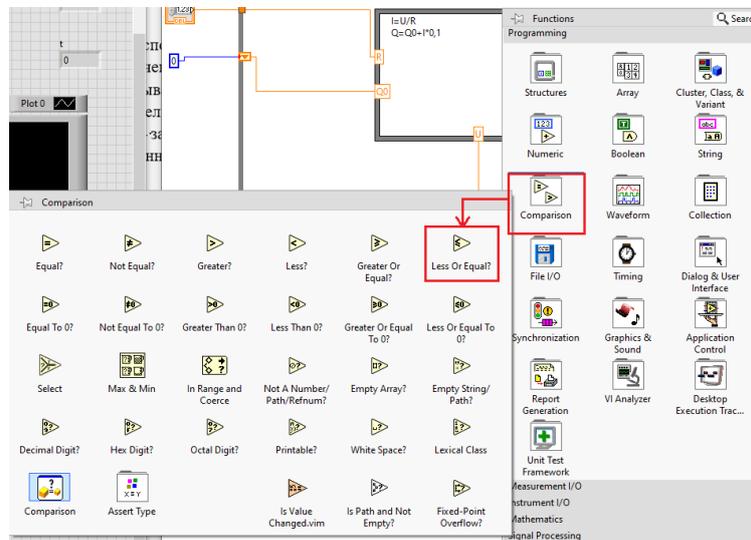


Рисунок 3.13 – Узел «*Less Or Equal?*»

Результат «*False*» или «*True*» присоединить к терминалу завершения цикла проводником зеленого цвета, который окрашивается автоматически при появлении логического сигнала.

По умолчанию каждый цикл рассчитывается за одну миллисекунду, поэтому для отслеживания динамики процесса установите задержку цикла, равную 100 мс. Для этого на панели «*Functions*» выберите приборы времени, в ней пиктограмму метронома («*Wait Until Next ms Multiply*») (рисунок 3.14).

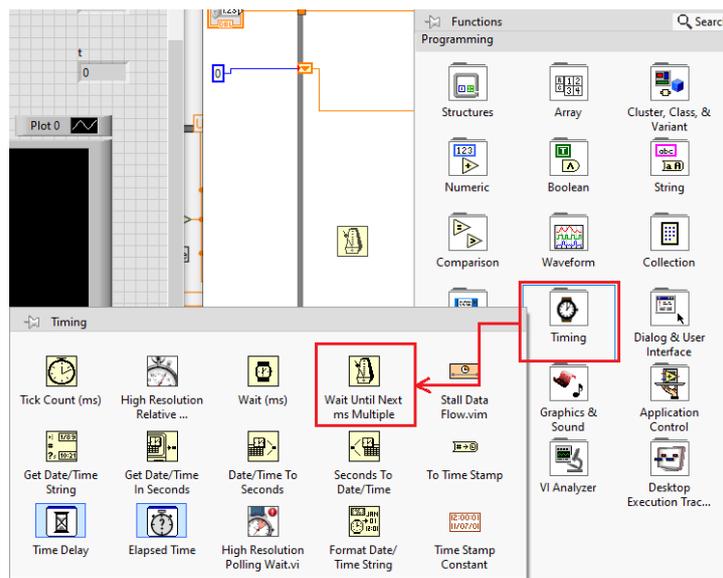


Рисунок 3.14 – Элемент метроном

Поместите данный элемент внутри цикла, найдите входной терминал и щелчком правой кнопки мыши вызовите всплывающее меню. В нем необходимо выбрать команды «Create» → «Constant», как показано на рисунке 3.15.

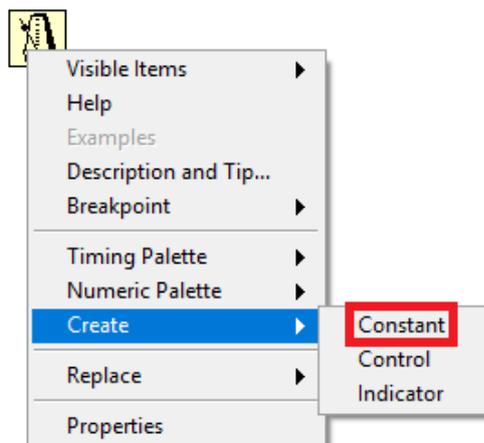


Рисунок 3.15 – Создание задержки цикла

В появившемся прямоугольнике введите число 100.

Значения Q и t , сохраненные в цикле после выполнения последней итерации, выводятся на цифровые индикаторы величины исходного заряда и времени разрядки. Кроме того, вне цикла, после его выполнения, рассчитывается емкость конденсатора. Для удобства отображения значений значение Q дополнительно умножим на 1000, а значение C – на 10^6 .

Убедитесь, что стрелка запуска цикла имеет правильную, не изломанную форму. Это означает, что программа составлена правильно и готова к запуску. В противном случае щелчком правой кнопки мыши по стрелке вызываем контекстное меню с распечаткой допущенных ошибок. Устраняем их и запускаем программу.

При этом на первых двух осциллографах строятся кривые падения напряжения на обкладках конденсатора и соответствующего уменьшения тока через сопротивление, как показано на рисунке 3.16.

На третьем осциллографе демонстрируется подсчет по времени величины суммарного заряда, сошедшего с обкладок конденсатора. В конце цикла на цифровых индикаторах появляются значения заряда – Q , емкости конденсатора – C и времени разрядки – t .

Окончательный вид подпрограммы моделирования разрядки конденсатора представлен на рисунках 3.16 и 3.17.

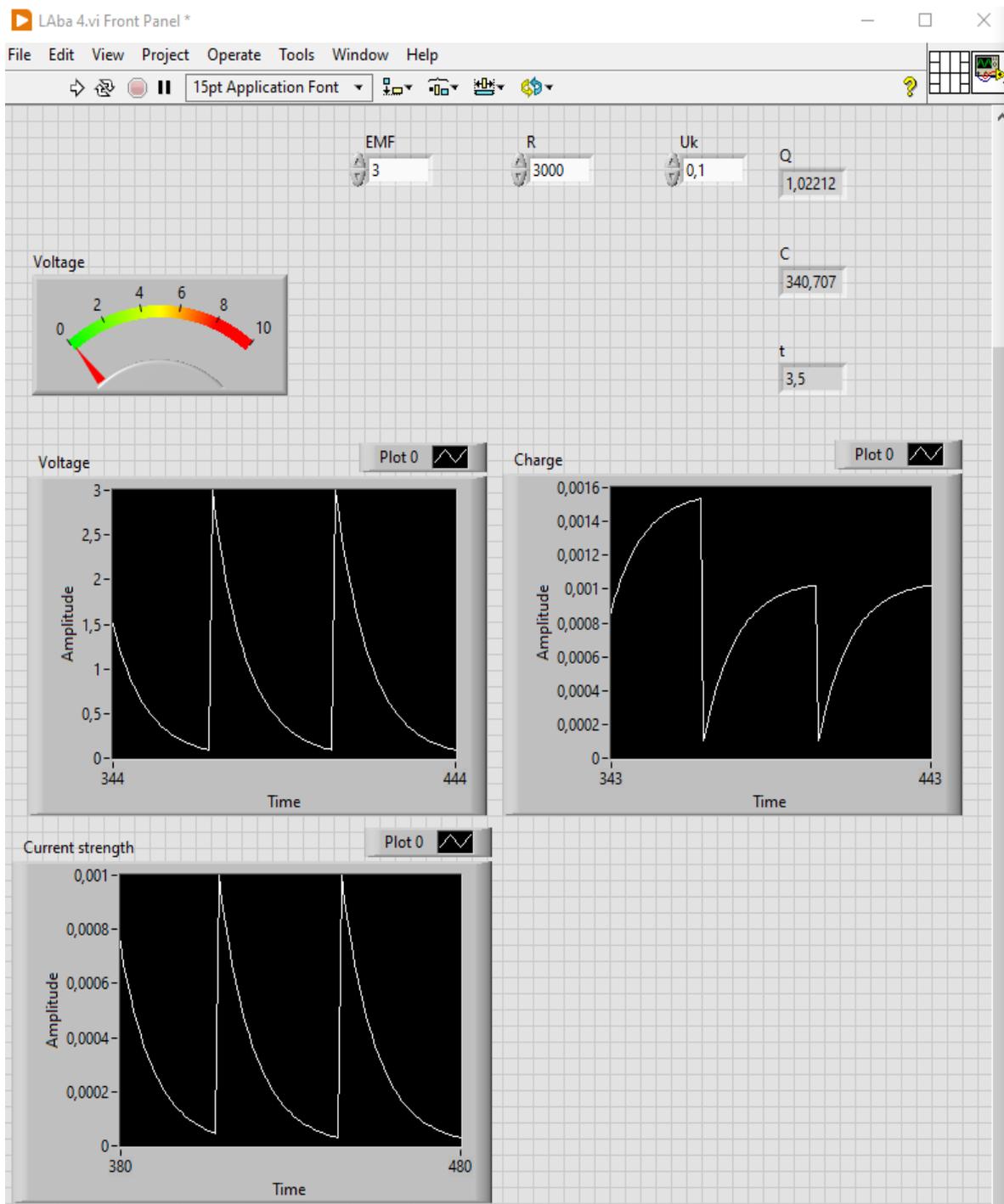


Рисунок 3.16 – Подпрограмма моделирования разрядки конденсатора с отображением результатов в окне «*Front Panel*»

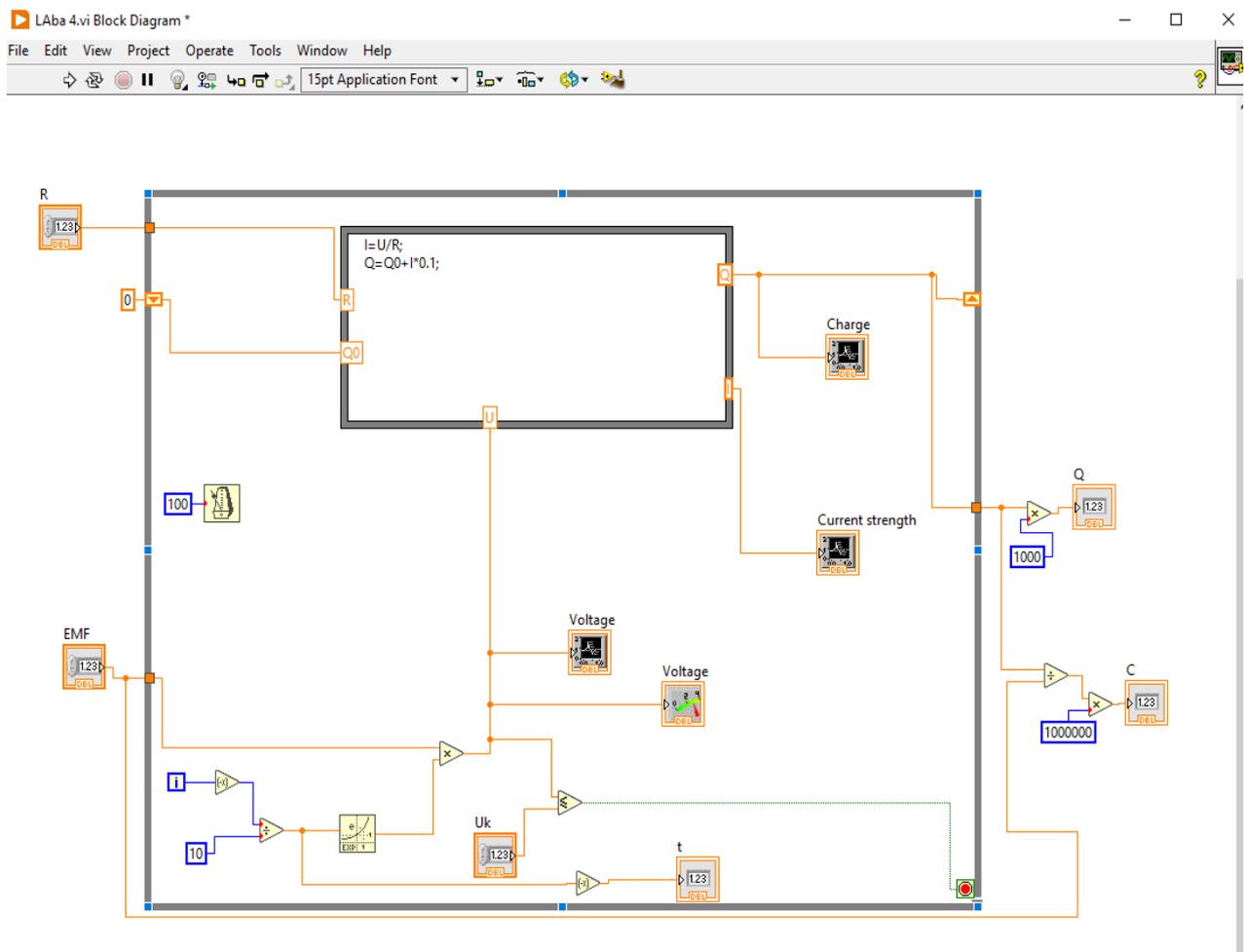


Рисунок 3.17 – Подпрограмма моделирования разрядки конденсатора, окно «*Block Diagram*»

Провести моделирование процесса при различных значениях сопротивления резистора в цепи разрядки, занести эти значения в таблицу 3.1 и построить график зависимости $C = f(R)$.

Таблица 3.1 – Результаты измерений

R , Ом	t , с	Q , Кл	C , мкФ
R_1	3,5	3,75	1249,5
R_2	3,5	1,68	561,6
R_3	3,5	1,14	379,4
R_4	3,5	0,98	327,1

3.3 Индивидуальные задания

В рамках лабораторной работы предусмотрено выполнение заданий в соответствии с индивидуальным номером зачетной книжки каждого студента. При этом важно учесть, что повторение заданий не допускается, обеспечивая таким

образом самостоятельное выполнение своей работы. В случае совпадения номера зачетной книжки с уже занятым заданием студенту предоставляется возможность выбрать свободный номер из предложенных вариантов.

Варианты заданий представлены в таблице 3.2.

Таблица 3.2 – Варианты индивидуальных заданий

Вариант	<i>R1</i>	<i>R2</i>	<i>R3</i>	<i>R4</i>
1	500	1000	2000	3000
2	558	1721	2071	3880
3	202	1331	2073	3028
4	601	1184	2446	3092
5	818	1820	2694	3125
6	893	1010	2109	3305
7	857	1523	2678	3098
8	343	1016	2846	3649
9	900	1510	2608	3596
10	419	1472	2436	3379
11	201	1821	2335	3607
12	589	1430	2381	3858
13	530	1078	2674	3348
14	667	1400	2270	3049
15	806	1539	2400	3684
16	754	1880	2669	3342
17	836	1073	2165	3334
18	790	1758	2105	3871
19	373	1477	2243	3369
20	504	1124	2866	3790
21	581	1681	2289	3051
22	241	1626	2688	3238
23	463	1454	2530	3380
24	212	1846	2415	3601
25	568	1132	2093	3467
26	656	1790	2237	3473
27	692	1597	2696	3543
28	766	1302	2262	3773
29	290	1598	2296	3326
30	396	1673	2170	3140

3.4 Содержание отчета

В рамках выполнения лабораторной работы № 3 необходимо выполнить одно индивидуальное задание. В связи с этим отчет по лабораторной работе должен содержать:

- 1) титульный лист;
- 2) введение;

- 2.1) цель выполнения задания;
- 2.2) краткое описание задания;
- 3) основная часть:
 - 3.1) ход выполнения задания;
 - 3.2) обработка и анализ результатов выполнения задания;
- 4) выводы по результатам выполнения задания.

3.5 Контрольные вопросы

1 Какие новые элементы и структуры среды *LabVIEW* использованы в настоящей работе?

2 Для чего выполнение программы измерений и интегрирования зарядов прекращается при остаточном напряжении на обкладках конденсатора 0,1 В?

3 Каким будет результат подсчета суммарного заряда при уменьшении порога отключения до 0,01 В?

4 В чем состоят основные отличия программ моделирования и автоматизации реального эксперимента?

5 Назовите величину элементарного заряда.

6 В каком случае два проводящих тела образуют конденсатор?

7 Что называется зарядом конденсатора?

8 От каких величин зависит время разрядки конденсатора?

9 Определите энергию электростатического поля испытанного конденсатора, заряженного 3,5 В.

10 Какие выводы можно сделать о зависимости времени разрядки конденсатора от сопротивления нагрузки?

11 Какие преимущества и недостатки имеет использование экспоненциальной функции для моделирования падения напряжения на обкладках конденсатора?

Вспомогательная литература

1 Трэвис, Д. *LabVIEW для всех* / Д. Трэвис, Д. Кринг. – 4-е изд., перераб. и доп. – М. : ДМК Пресс, 2015. – 904 с.

2 Пилипенко, О. В. Основы программирования, математического моделирования и обработки данных в среде *LABVIEW* : практикум / О. В. Пилипенко, Н. Б. Горбачёв, М. А. Музалевская. – Орел : ОрелГТУ, 2008. – 70 с.

3 Васильев, А. С. Основы программирования в среде *LabVIEW* / А. С. Васильев, О. Ю. Лашманов. – СПб. : Университет ИТМО, 2015. – 82 с.

4 Евдокимов, Ю. К. *LabVIEW в научных исследованиях* / Ю. К. Евдокимов, В. Р. Линдваль, Г. И. Щербаков. – М. : ДМК Пресс, 2018. – 400 с.

5 *LabVIEW: Практикум по основам измерительных технологий* / В. К. Батоврин [и др.]. – М. : ДМК Пресс, 2005. – 208 с.

Лабораторная работа № 4
**Автоматизация учета времени включения и выключения приборов
в среде *LabVIEW***

Цель работы: используя возможности среды *LabVIEW*, необходимо разработать программу для автоматического учета времени включения и выключения приборов.

Основные задачи:

- 1 Изучить функции *LabVIEW* для работы с форматом времени.
- 2 Ознакомиться с возможностями среды виртуальных приборов *LabVIEW* по учету данных.

4.1 Теоретические сведения

Автоматический учет данных позволяет сократить большую часть производственного процесса. Запись времени включения и выключения приборов, времени прибытия и отбытия сотрудников, температуры воздуха в помещении, расхода электричества и воды и т. д. позволяет снизить расходы и оптимизировать производственные процессы.

Функционал *LabVIEW* позволяет автоматизировать учет большинства возможных производственных процессов. Для обеспечения учета времени используется широкий выбор функции для работы с данными, например «*Timestamp*», которая преобразует число во временную метку, показывающую, когда произошло определенное событие.

Метка времени *LabVIEW* – это 128-битный тип данных, представляющий абсолютное время. Этот тип данных можно интерпретировать как знаковое 128-битное число с фиксированной запятой и 64-битным основанием.

Эту информацию важно знать при передаче метки времени узлу библиотеки вызовов или пользовательской библиотеке. Наиболее значимые 64 бита следует интерпретировать как 64-битное целое число со знаком, дополненное до двух, а наименее значимые 64 бита – как 64-битное целое число без знака.

Обычно при записи данных в файл временные метки преобразуются в строки и существует ряд встроенных функций *LabVIEW*, которые позволяют это сделать (*VI Get Date/Time String*, *VI Format Date/Time String*). Однако если нужно преобразовать строку даты и времени обратно в метку времени, то встроенной функции в *LabVIEW* для этого нет.

Рассмотрим решение, которое позволяет автоматизировать учет времени включения и выключения приборов при помощи взятия строки даты и строки времени (отформатированную в соответствии с выходными данными виртуального прибора *Get Date/Time String*) и создания временной метки при помощи встроенной функции «*Timestamp*».

4.2 Задание для выполнения лабораторной работы

Задание. Разработать программу по автоматическому учету времени включения и выключения ламп.

Порядок выполнения задания

Откроем «*Block Diagram*» и создадим структуру «*While Loop*», а внутри нее добавим структуру «*Event Structure*» (рисунок 4.1).

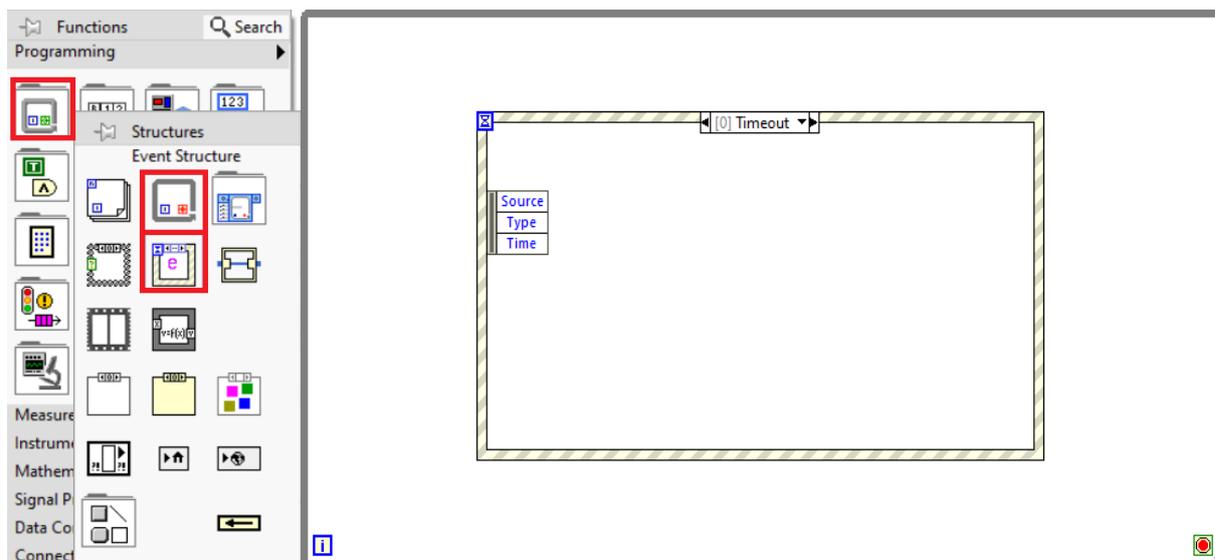


Рисунок 4.1 – Выбор структур и нанесение на «*Block Diagram*»

Добавим кнопку завершения цикла. Для этого наведем курсор на «*Loop Condition*» (находится в правом нижнем углу *While*-цикла), а затем нажмем правой кнопкой мыши и выберем «*Create Control*».

Добавим две кнопки, для этого зайдём в «*Front Panel*», выберем «*Controls*» → «*Boolean*» → «*Round LED*». В «*Block Diagram*» кликнем по пиктограмме правой кнопкой мыши и выберем «*Change to Control*» (рисунок 4.2).

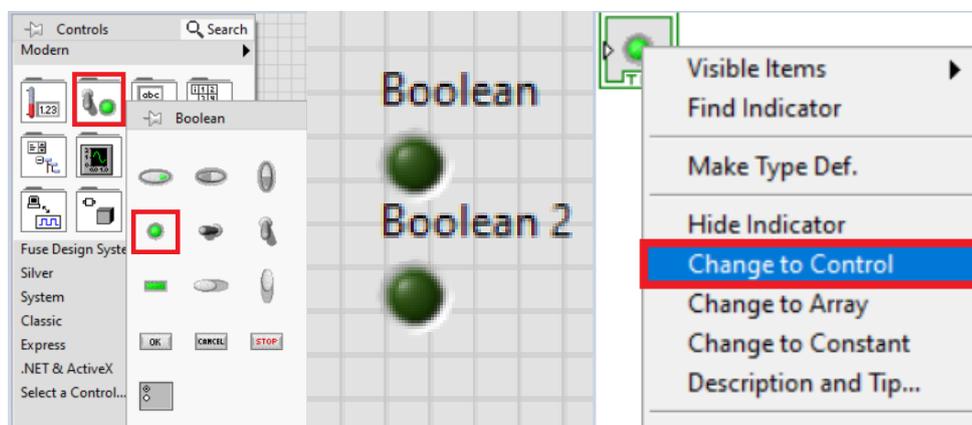


Рисунок 4.2 – Установка кнопки

Вторую кнопку можно продублировать уже в «*Block Diagram*».

Создадим событие в структуре «*Event Structure*». Событие («*Event*») применяется для ситуаций, когда необходимо совершить какое-либо действие при активации объекта. Кликаем правой кнопкой мыши по рамке структуры и выбираем «*Edit Events Handled by this Case*», далее выбираем в поле «*Event Sources*» → «*Boolean*» и нажимаем «ОК». Так добавили событие для первой кнопки.

Для того чтобы добавить событие для второй кнопки, кликнем правой кнопкой мыши по рамке структуры и выберем «*Add Event Case*», далее «*Event Sources*» → «*Boolean 2*» и нажимаем «ОК». И также нужно добавить событие по нажатию кнопки «*Stop*», которую создали ранее (рисунки 4.3 – 4.6).

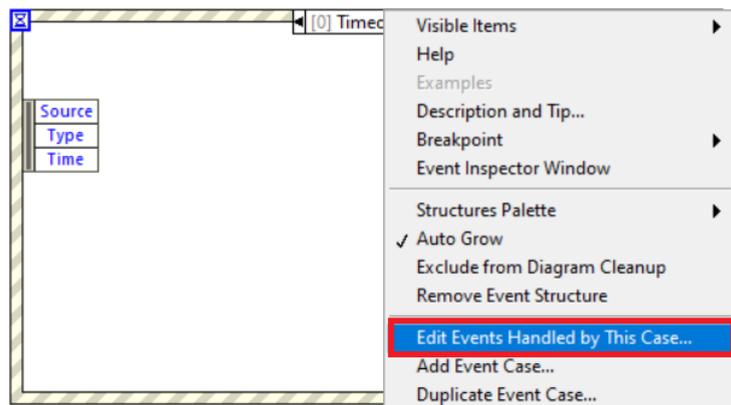


Рисунок 4.3 – Создание события для первой кнопки

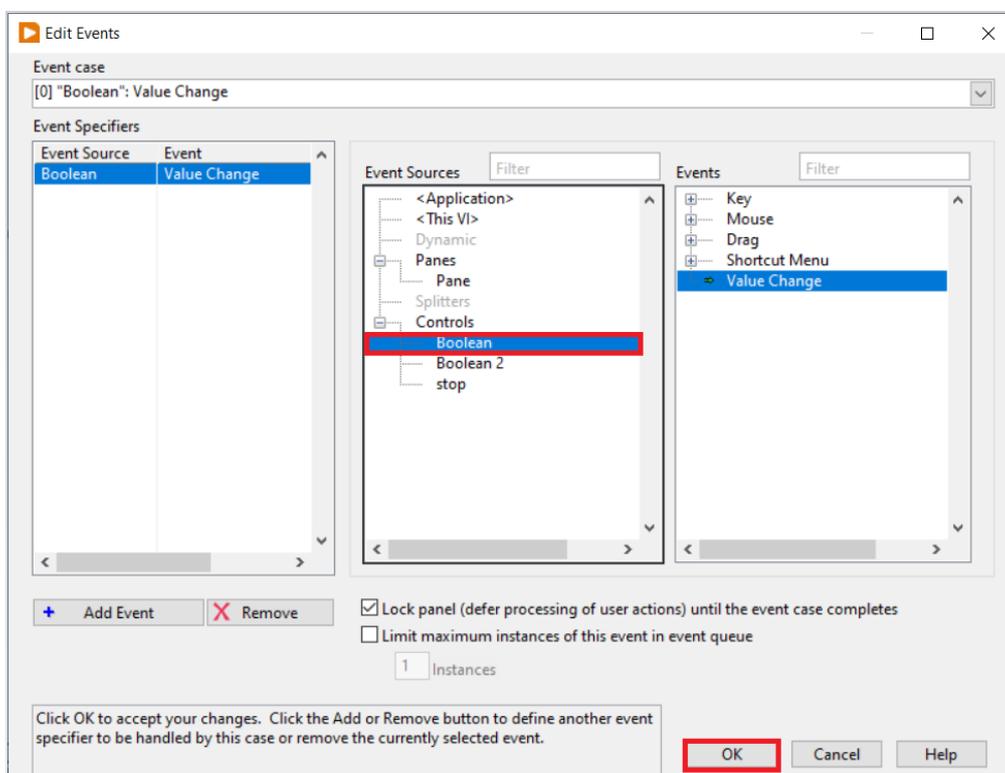


Рисунок 4.4 – Выбор кнопки *Boolean* для события

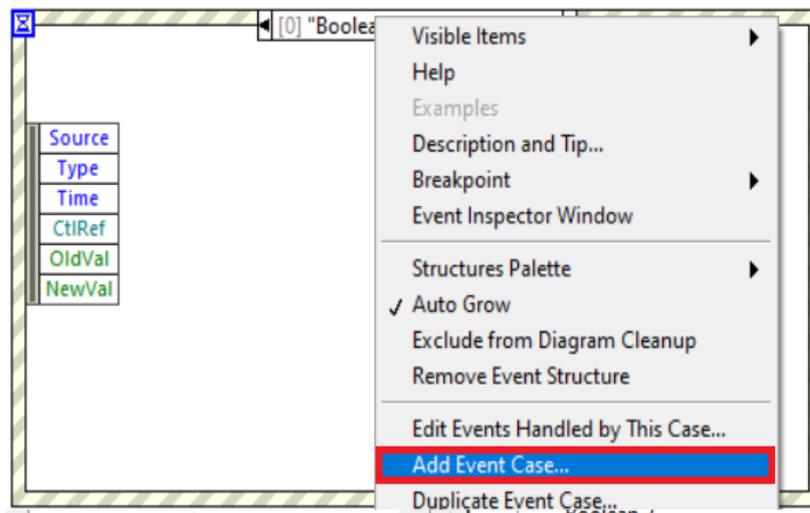


Рисунок 4.5 – Создание события для второй кнопки

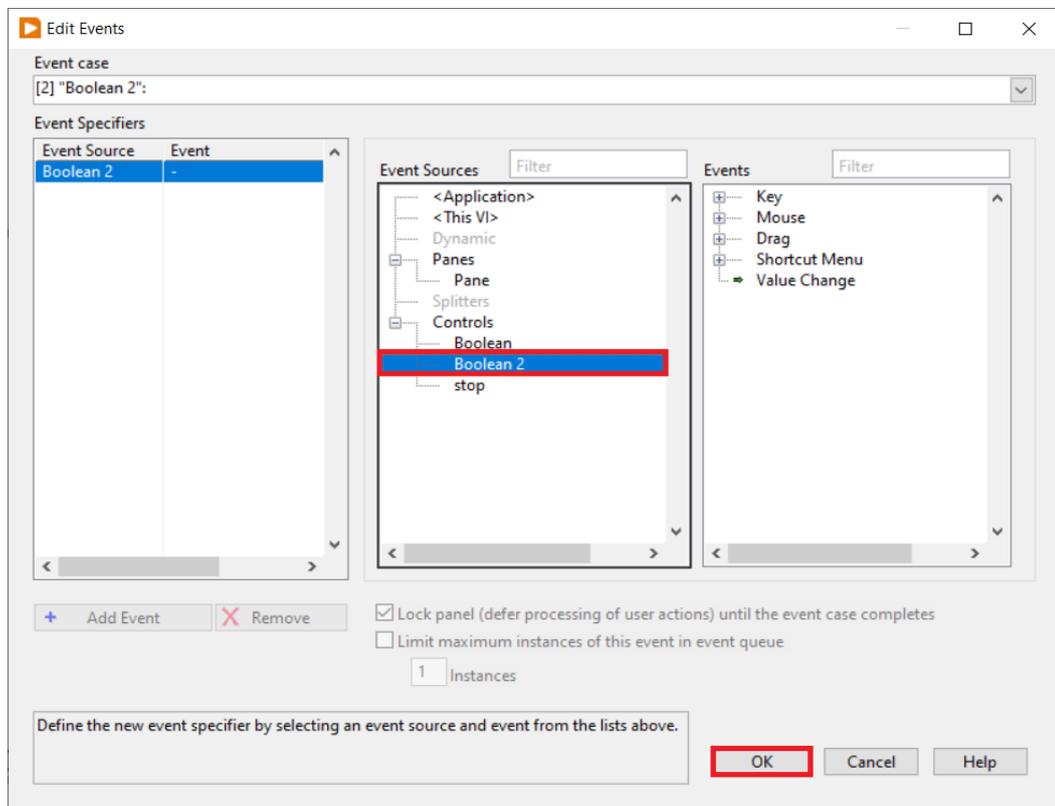


Рисунок 4.6 – Выбор кнопки *Boolean 2* для события

Перенесем кнопки внутрь «*Event Structure*», предварительно выбрав для них события, которые создали, как показано на рисунке 4.7.

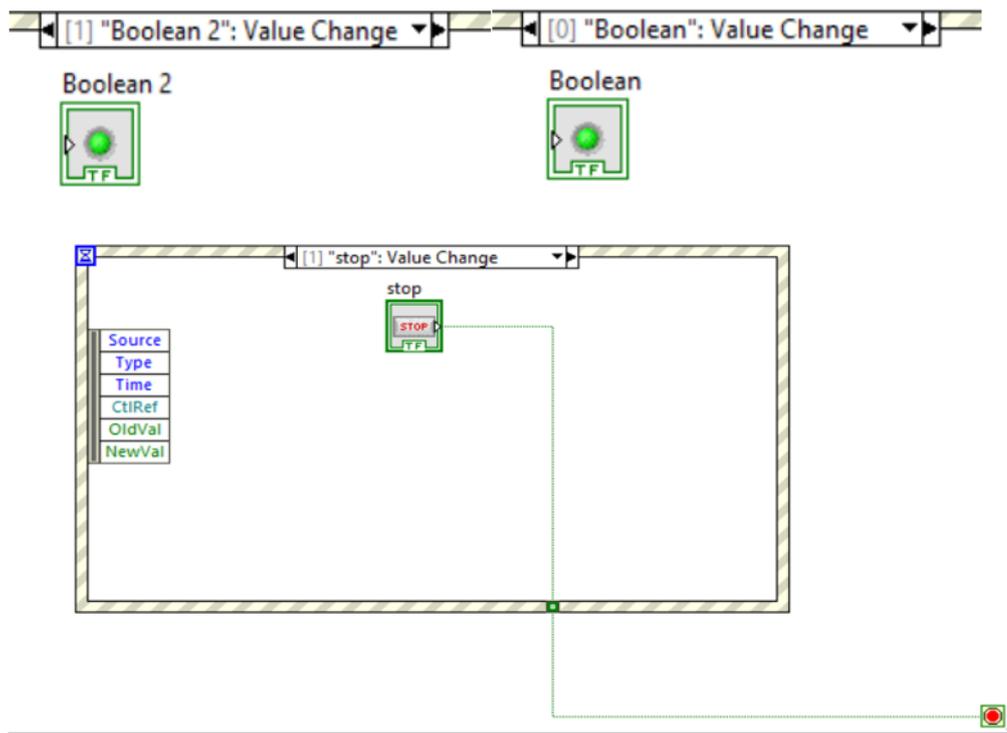
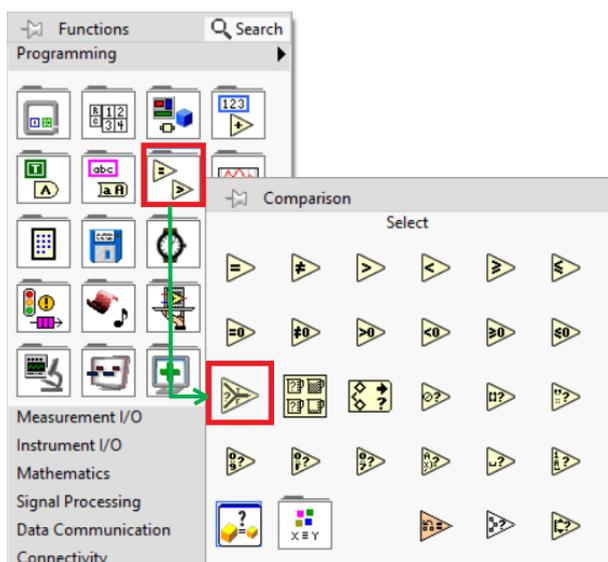


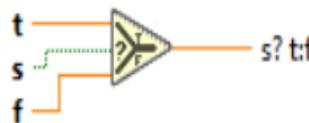
Рисунок 4.7 – Перенос кнопок в структуру

Добавим в событие «*Boolean*» элемент «*Select*»: «*Functions*» → «*Comparison*» → «*Select*».

Элемент «*Select*» выбирает значение «*t*» или «*f*» и подает его на выход в зависимости от значения «*s*», если «*s*» = *TRUE*, то на выход идет значение «*t*», если «*s*» = *FALSE*, то на выход идет значение «*f*» (рисунок 4.8).



а



б

а – выбор элемента «*Select*»; б – принцип выбора значения

Рисунок 4.8 – Элемент «*Select*»

В окне «*Front Panel*» кликнете правой кнопкой мыши по каждой из кнопок и выберем «*Change to Control*». Соединим выход кнопки «*Boolean*» и вход «*s*» элемента «*Select*». Для входов «*t*» и «*f*» элемента «*Select*» создадим строки «*On*» и «*Off*» соответственно. Строка находится по пути «*Functions*» → «*String*» → «*String Constant*» (рисунок 4.9).

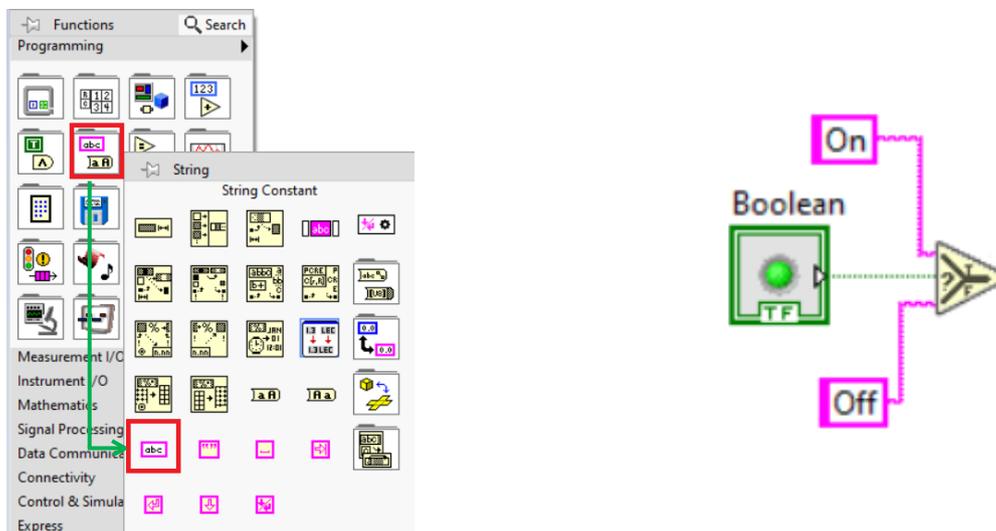


Рисунок 4.9 – Инициализация входов элемента «*Select*»

Добавим в событие два новых элемента «*Get Date/Time In Seconds*» и «*Get Date/Time String*». Оба эти элемента располагаются в библиотеке «*Functions*» → «*Timing*». «*Get Date/Time In Seconds*» в данной лабораторной работе необходим, чтобы получить текущее время, но получаем время в формате «*TimeStamp*». С помощью элемента «*Get Date/Time String*» можем конвертировать «*TimeStamp*» в «*String*» для формирования отчета о включении/выключении кнопки (рисунок 4.10).



Рисунок 4.10 – Запись текущего времени

Создадим строку для формирования отчета. Для этого воспользуемся элементом «Concatenate Strings»: «Functions» → «String» → «Concatenate Strings». «Concatenate Strings» соединяет входящие строки в одну. Так как они не разделяются пробелом, добавим пустую строку, чтобы разделять строки. Для этого просто вставляем «String Constant» и оставляем ее пустой: «Functions» → «String» → «String Constant». Чтобы увеличивать количество строк, входящих в «Concatenate Strings», необходимо вытянуть элемент вниз (рисунок 4.11).

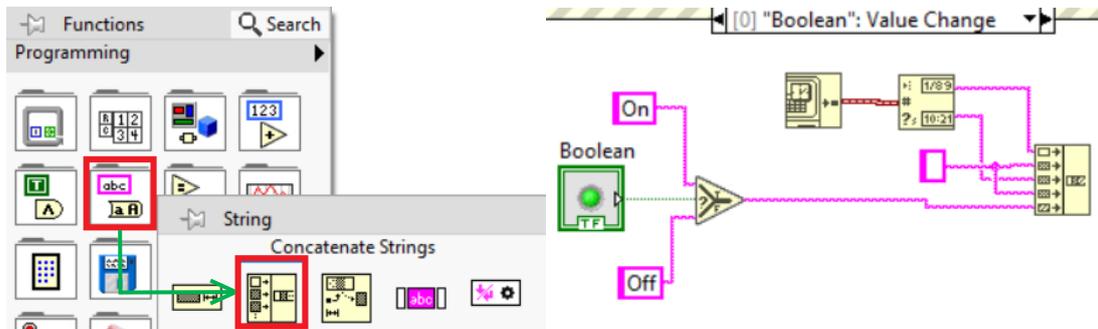


Рисунок 4.11 – Формирование строки для отчета

Конвертируем нашу строку в массив, для этого используем элемент «Insert Into Array», который находится в ветке «Functions» → «Array» (рисунок 4.12).

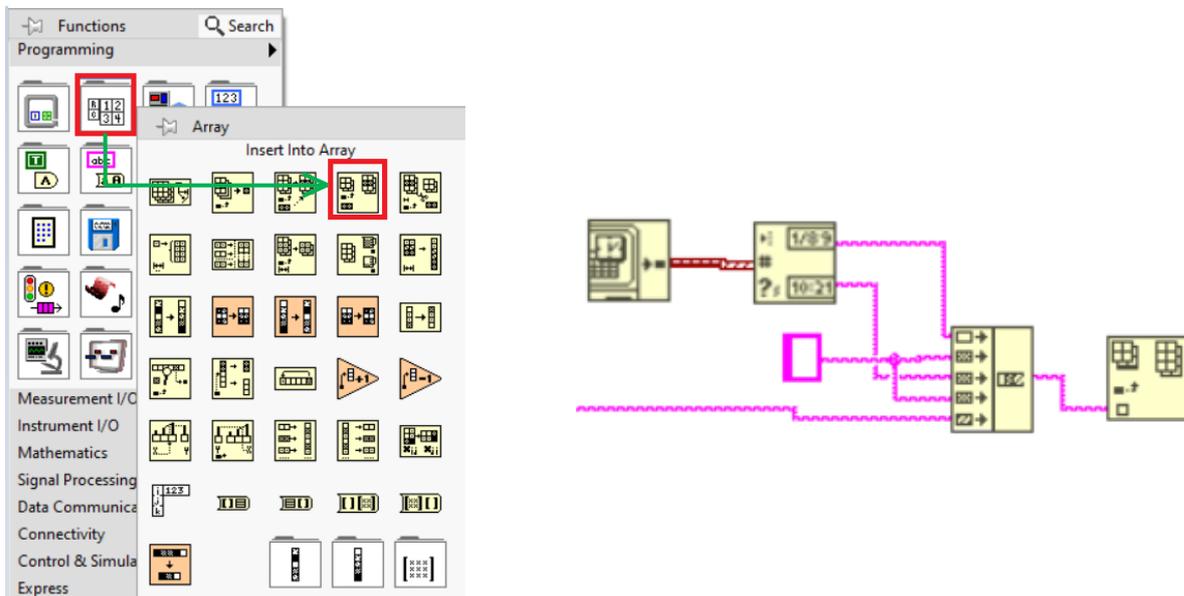


Рисунок 4.12 – Конвертирование строки в массив

Соединим выход «Concatenate Strings» с входом «new element/subarray» элемента «Insert Into Array». Для того чтобы наш массив сохранялся внутри цикла, нужно сдвинуть регистр, проведем выход «Insert Into Array» за структуру «Event Structure» к границе «While Loop», кликнем на появившуюся ручку правой кнопкой мыши и выберем «Replace with Shift Register» (рисунок 4.13).

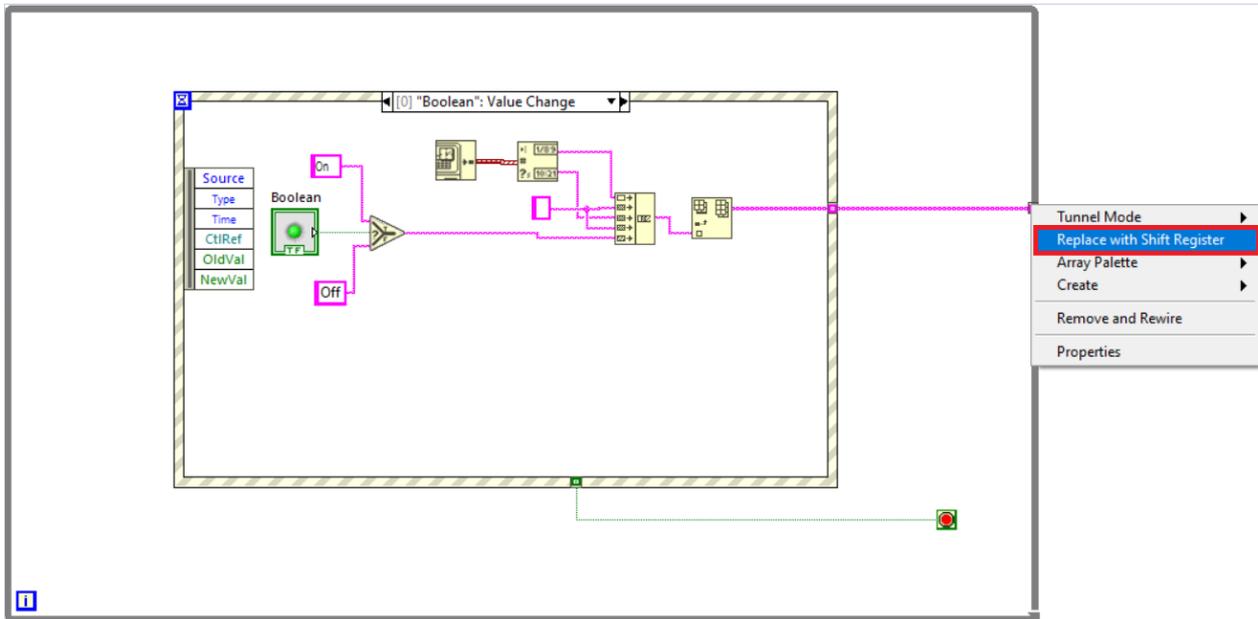


Рисунок 4.13 – Сдвиг регистра

Добавим заголовок для строки формирования отчета, так как он будет одинаковый для всех строк в данном событии («*Boolean*») вынесем его за структуру «*While Loop*», для этого кликнем правой кнопкой мыши на ручку слева, которая появилась в результате формирования сдвига регистра, выберем «*Create*» → «*Constant*» и внутри появившейся строки кликаем правой кнопкой мыши и выбираем «*Change to Array*», после чего пишем название заголовка «*Boolean*» в правой колонке. Соединим выход ручки с входом «*Array*» элемента «*Insert Into Array*», как показано на рисунке 4.14.

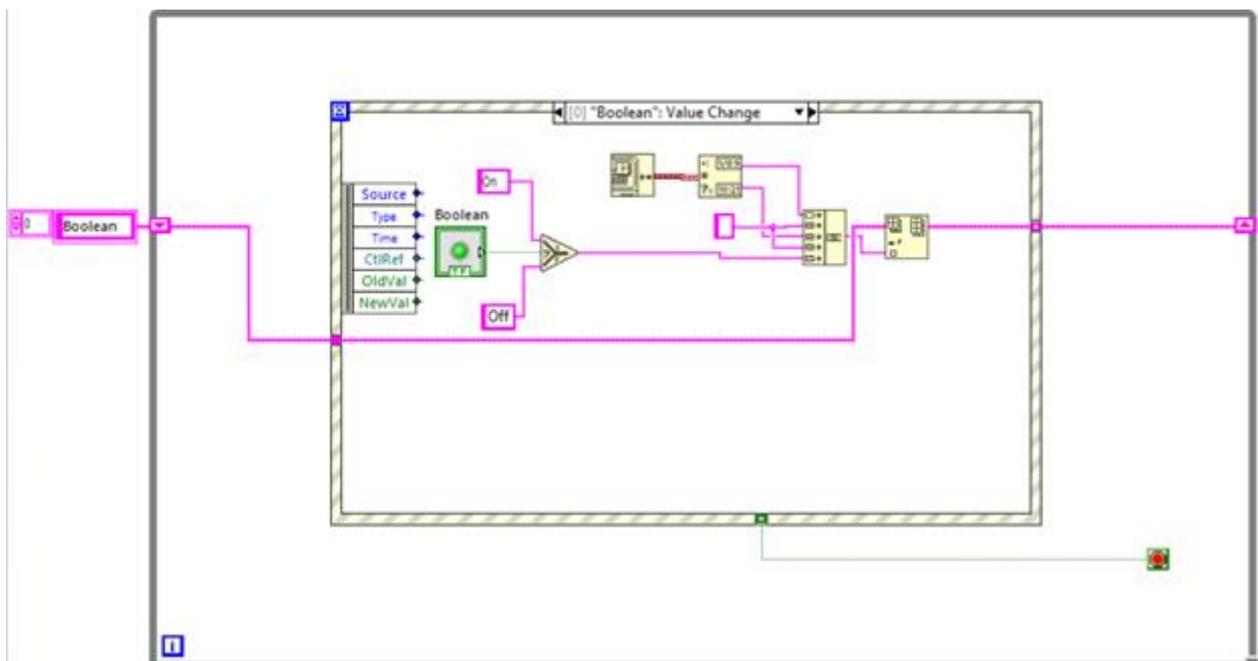


Рисунок 4.14 – Добавление заголовка к массиву

Для работы элемента «*Insert Into Array*» нужно проиндексировать массив, для этого воспользуемся элементом «*Increment*», находящимся по пути «*Functions*» → «*Numeric*» (рисунок 4.15). Добавим его внутрь события «*Boolean*», проведем выход «*Increment*» к границе «*While Loop*» и сделаем сдвиг регистра, после от появившейся слева ручки проведем к входу «*Increment*», после выход «*Increment*» также подведем к входу «*Index*» элемента «*Insert Into Array*», как показано на рисунке 4.16.

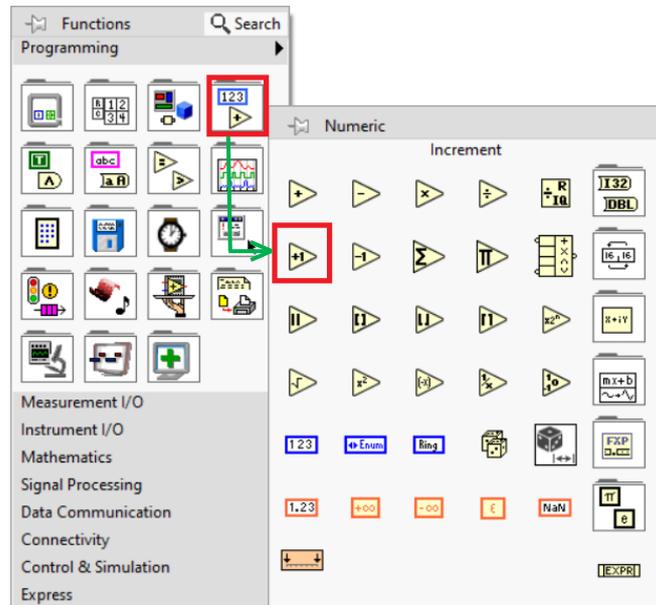


Рисунок 4.15 – Добавление элемента «*Increment*»

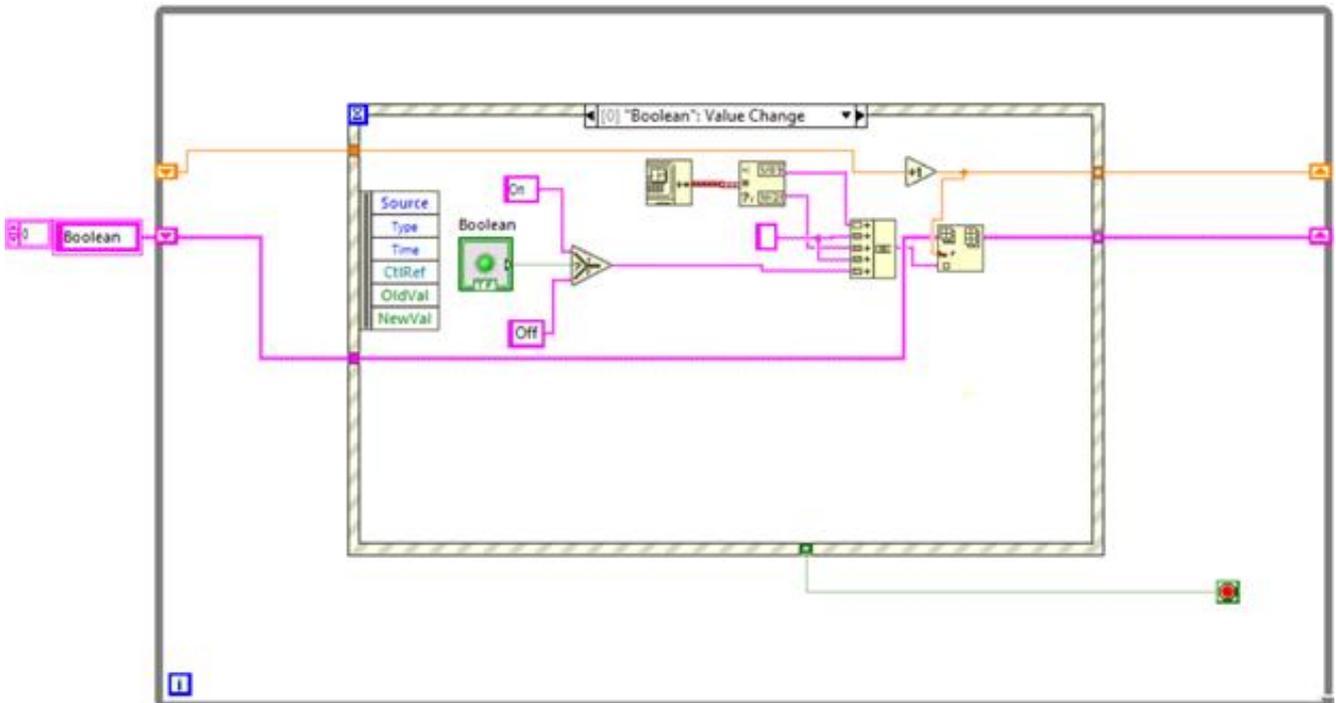


Рисунок 4.16 – Формирование индексации для массива

Необходимо начальное значение для элемента «*Increment*». Для этого в палитре функций найдем «*Numeric Constant*», которое расположено в ветке «*Functions*» → «*Numeric*» (рисунок 4.17), и разместим за структурой «*While Loop*» возле левой ручки, которая ведет к входу элемента «*Increment*» (рисунок 4.18). Внутри «*Numeric Constant*» зададим значение, равное «0», и соединим с ручкой (рисунок 4.19).

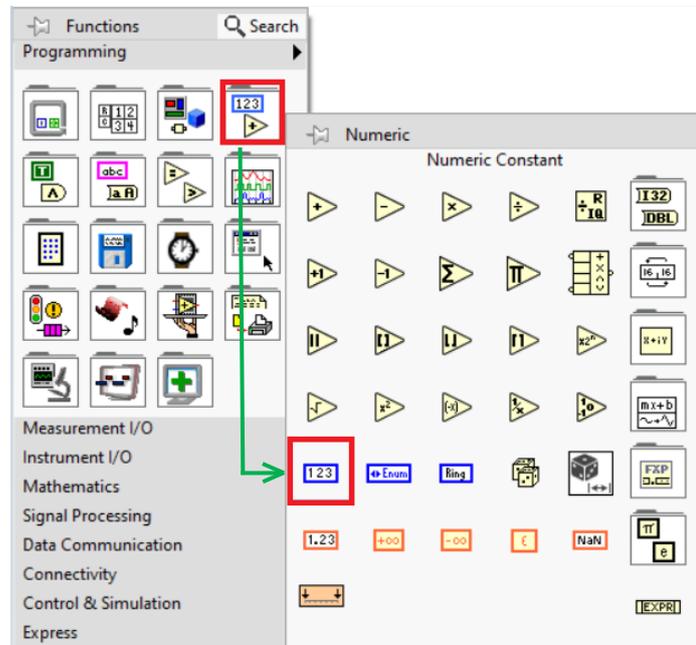


Рисунок 4.17 – Добавление элемента «*Increment*»

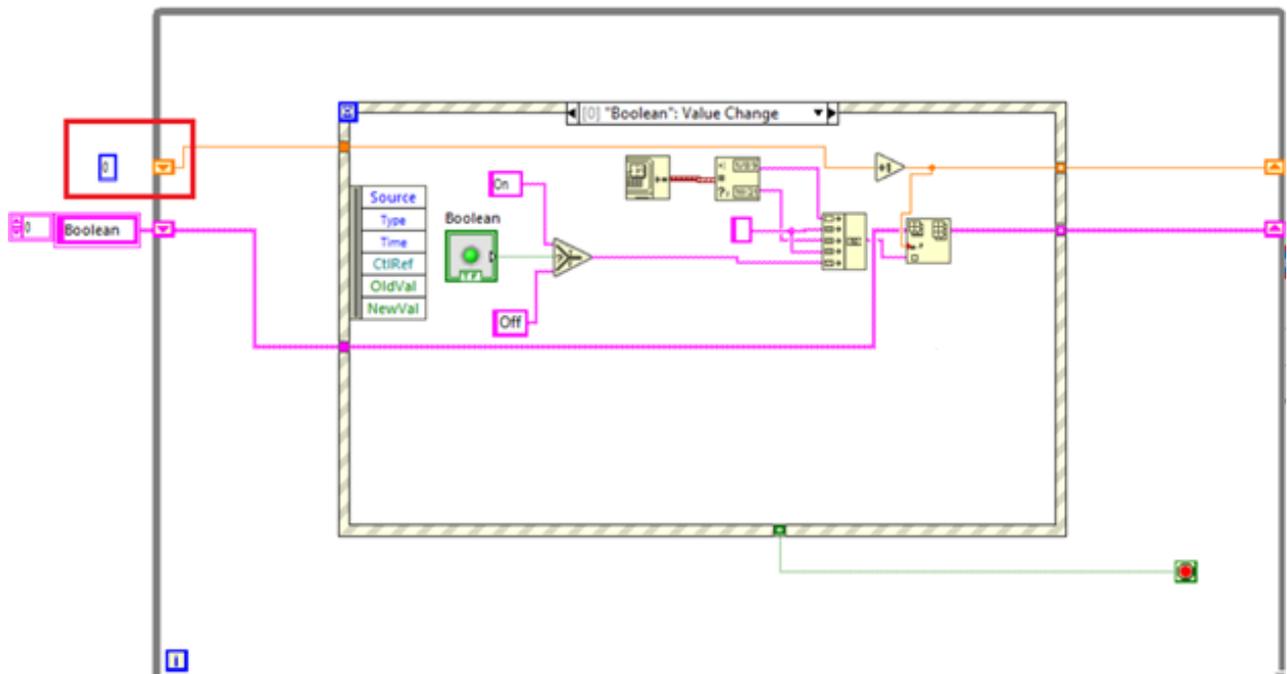


Рисунок 4.18 – Размещение элемента «*Increment*»

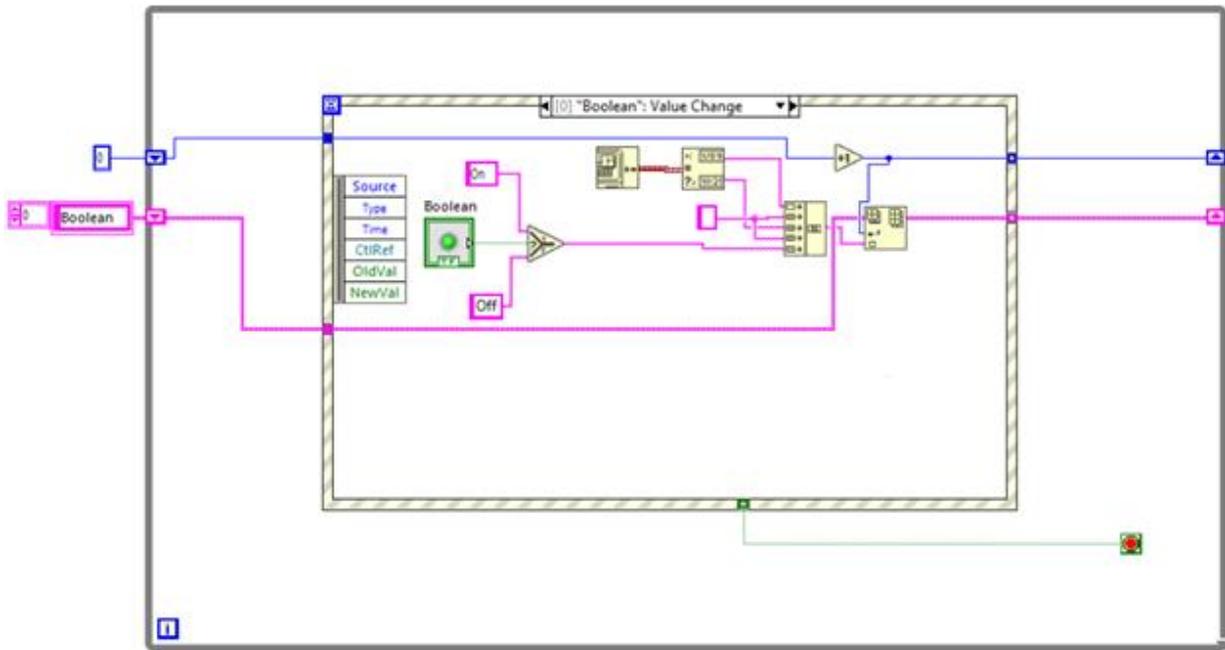


Рисунок 4.19 – Формирования начального значения для «Increment»

Создаем «*Numeric Constant*», а не константу через клик правой кнопкой мыши на ручку, так как изначально у «*Increment*» тип данных «*Double*», а для входа «*Array*» элемента «*Insert Into Array*» нужен тип данных «*Long*», который имеет «*Numeric Constant*».

Добавим элемент «*Elapsed Time*», который будет засекаеть время, в течение которого кнопка была включена, находящийся в ветке «*Functions*» → «*Timing*». Разместим «*Elapsed Time*» внутри события «*Boolean*» (рисунок 4.20).

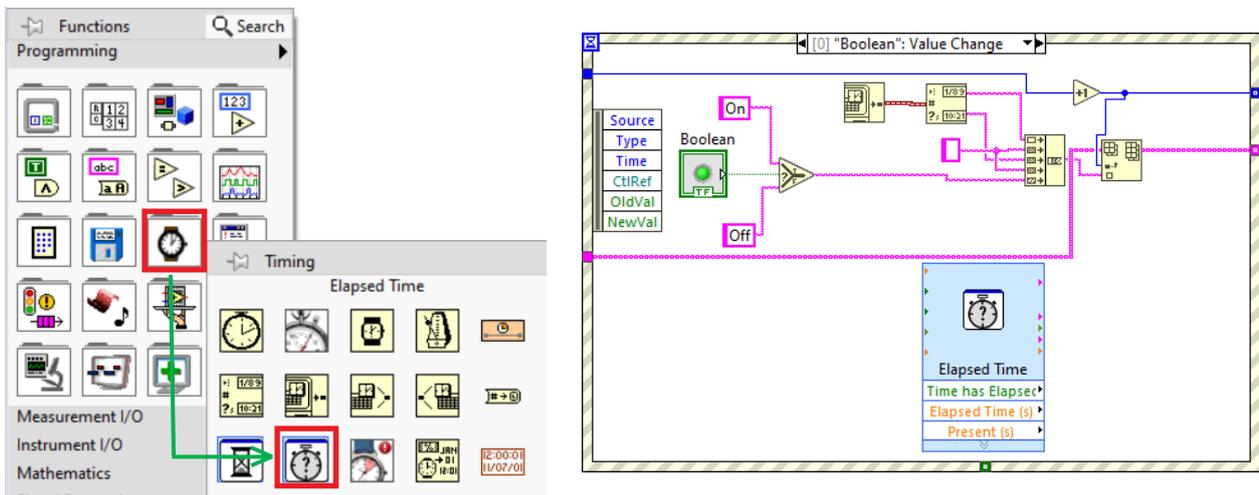


Рисунок 4.20 – Добавление «Elapsed Time»

При добавлении элемента «*Elapsed Time*» появится окно (рисунок 4.21), в котором нужно будет указать время, которое будем засекаеть. Также можно указывать время с помощью индикатора, подключив к входу «*Time Target*» выход индикатора.

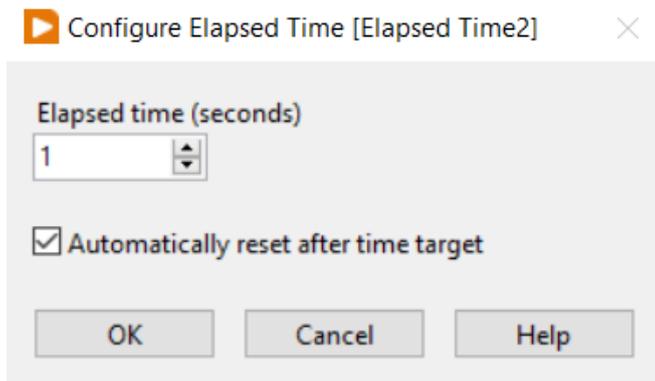


Рисунок 4.21 – Установка времени засечки

Проведем соединение от «*Boolean*» к входу «*Reset*» элемента «*Elapsed Time*», так при включении кнопки будет подаваться значение «*True*» на вход и начнется отсчет времени, а при выключении кнопки будет подаваться значение «*False*» и отсчет прекратится (рисунок 4.22).

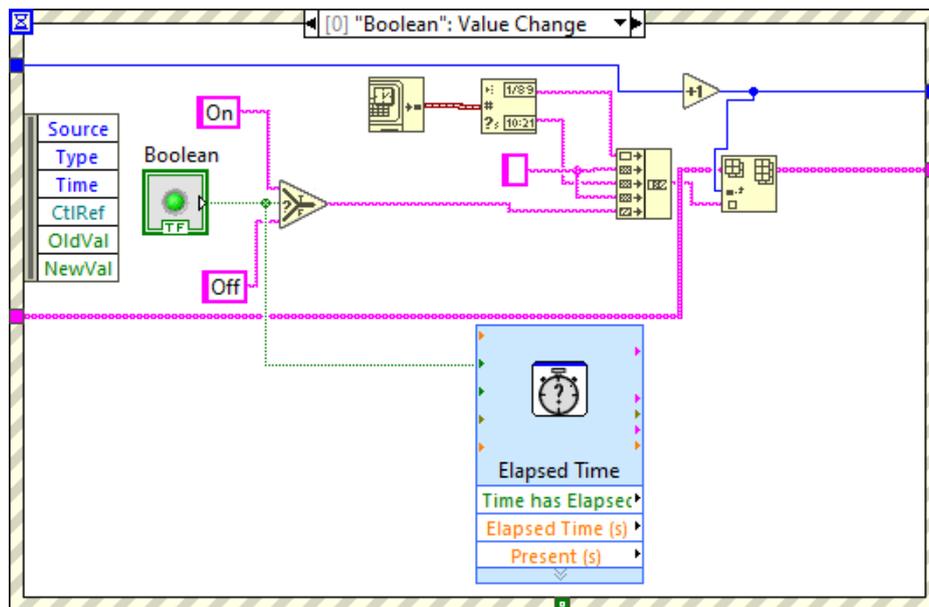


Рисунок 4.22 – Соединения «*Boolean*» и входа «*Reset*» элемента «*Elapsed Time*»

Засеченное время можно получить с входа «*Elapsed Time*», но его нужно перевести в массив, для этого опять вставляем элемент «*Insert Into Array*» (именно вставляем новый, а не копируем существующий). Выход «*Elapsed Time*» элемента «*Elapsed Time*» соединим с входом «*new element/subarray*» элемента «*Insert Into Array*». Вход «*index*» соединим с выходом уже созданного элемента «*Increment*». Выход «*Insert Into Array*» соединим с границей структуры «*While Loop*», добавим сдвиг регистра и появившуюся слева ручку соединим с входом

«array» элемента «Insert Into Array». Добавим константу к левой ручке, так же как делали это ранее (рисунок 4.23), в появившемся поле поставим «0».

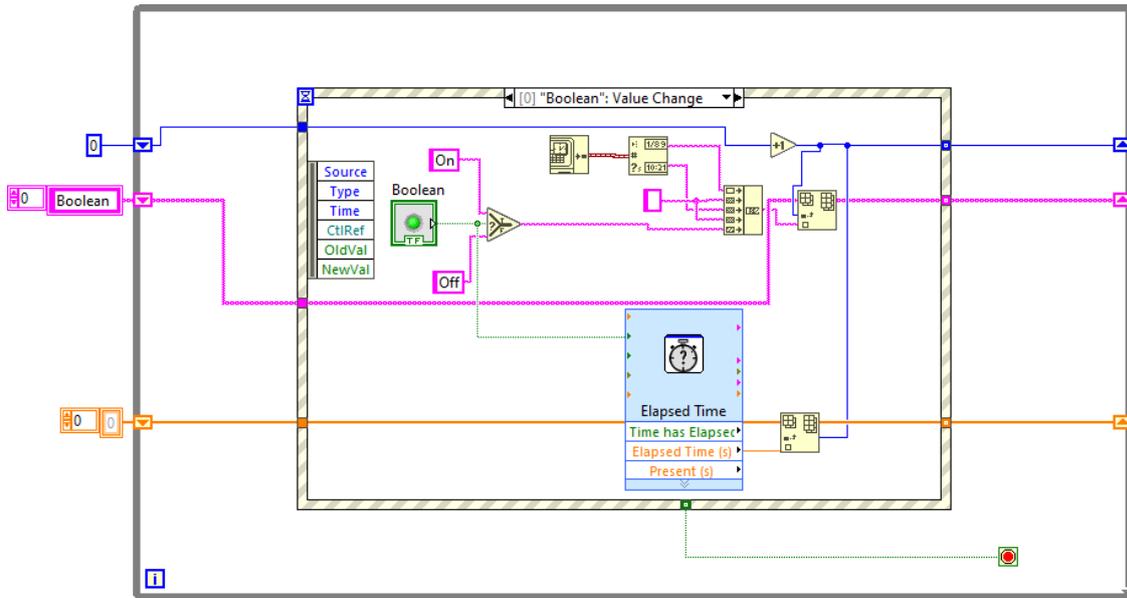


Рисунок 4.23 – Вставка данных в массив

Откроем событие «Boolean 2» и соединим ручки на границах структуры «Event Structure», как показано на рисунке 4.24.

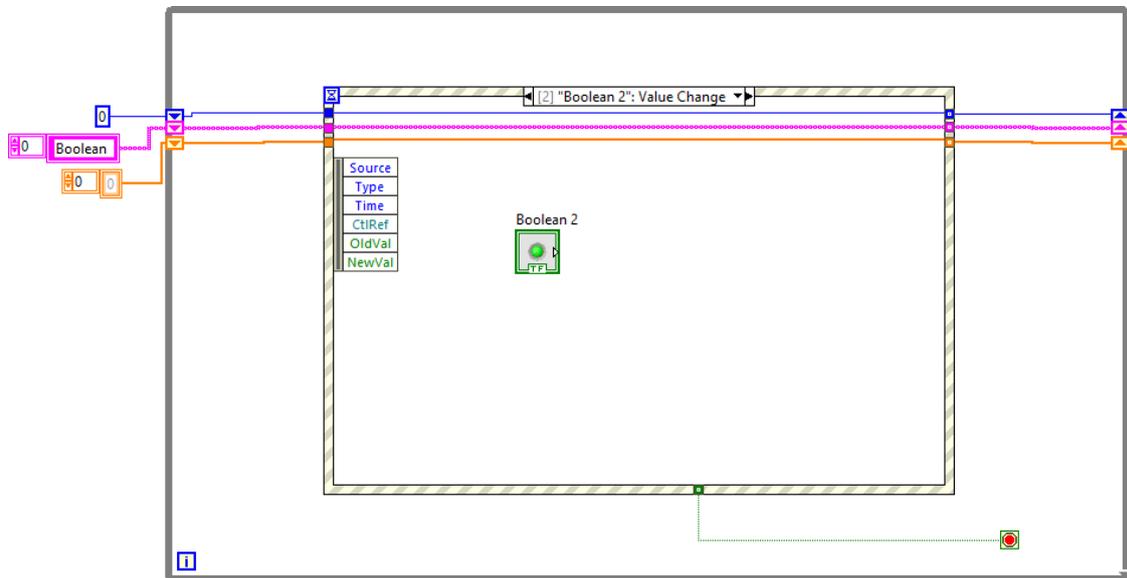


Рисунок 4.24 – Соединение ручек в событии «Boolean 2»

Скопируем все элементы события «Boolean» и вставим в событие «Boolean 2», кроме кнопки, она уже была туда добавлена ранее (рисунок 4.25).

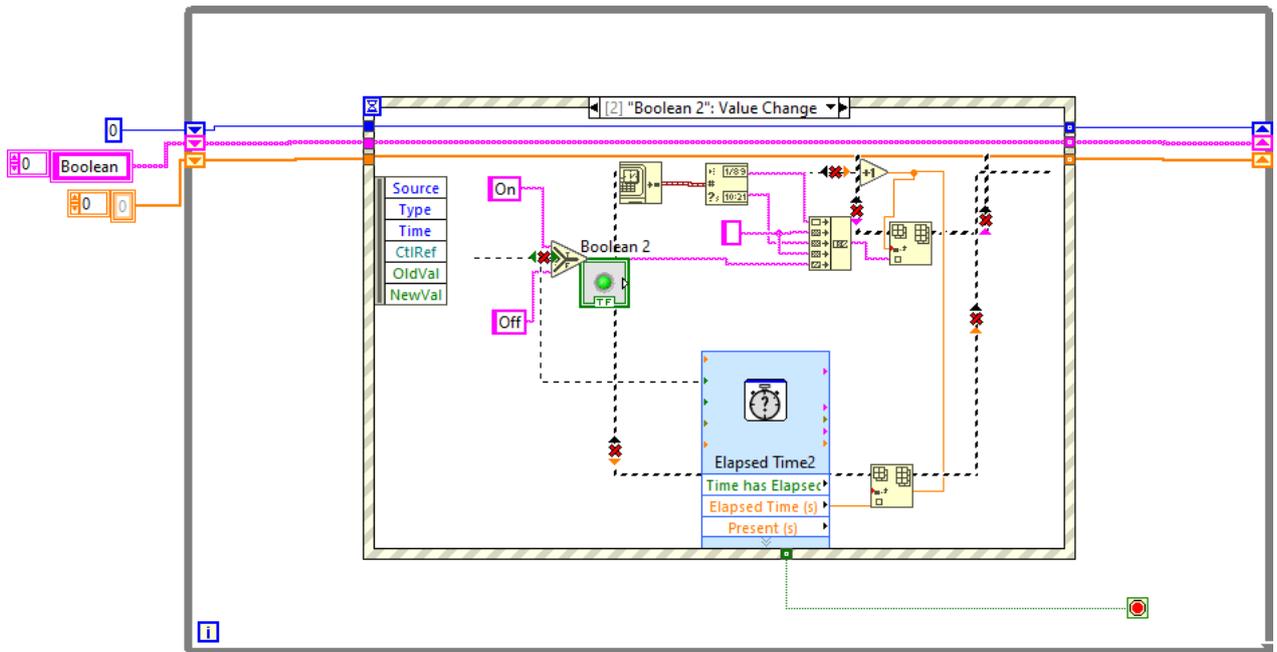


Рисунок 4.25 – Перенос данных в событие «*Boolean 2*»

Соединим выход «*Boolean 2*» с входом «*s*» элемента «*Select*». Скопируем все константы за пределами структуры «*While Loop*» и вставим их ниже, в строковой константе *с* аналогично с событием «*Boolean*» (рисунок 4.26).

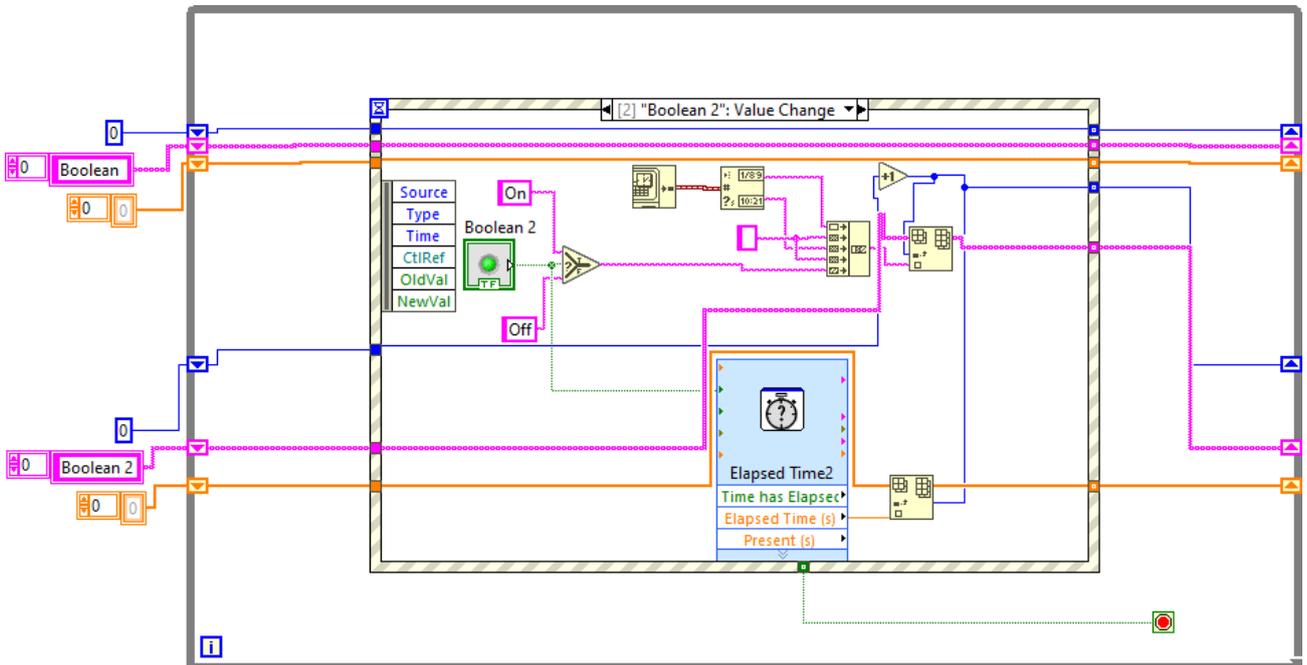


Рисунок 4.26 – Соединение нерабочих связей

Двойным кликом левой кнопкой мыши по пиктограмме таймера элемента «*Elapsed Time*», выберем время, которое будем засекаать согласно варианту индивидуального задания.

Откроем событие «*Stop*» и соединим там все ручки (рисунок 4.27).

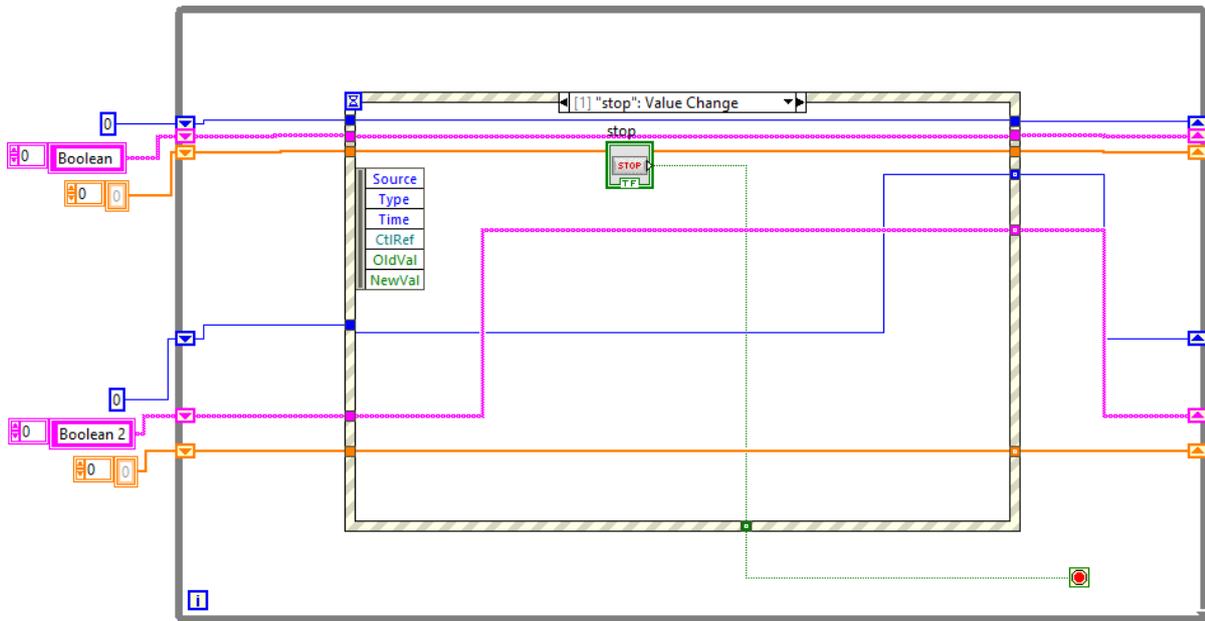


Рисунок 4.27 – Соединение ручек в событии «Stop»

Добавим индикатор к выходу «*Time Has Elapsed*» элемента «*Elapsed Time*» в событиях «*Boolean*» и «*Boolean 2*». Для этого кликаем правой кнопкой мыши по выходу «*Time Has Elapsed*», далее «*Create*» и потом «*Indicator*», как показано на рисунке 4.28. При отключении кнопки индикатор будет загораться, если указанное время прошло.

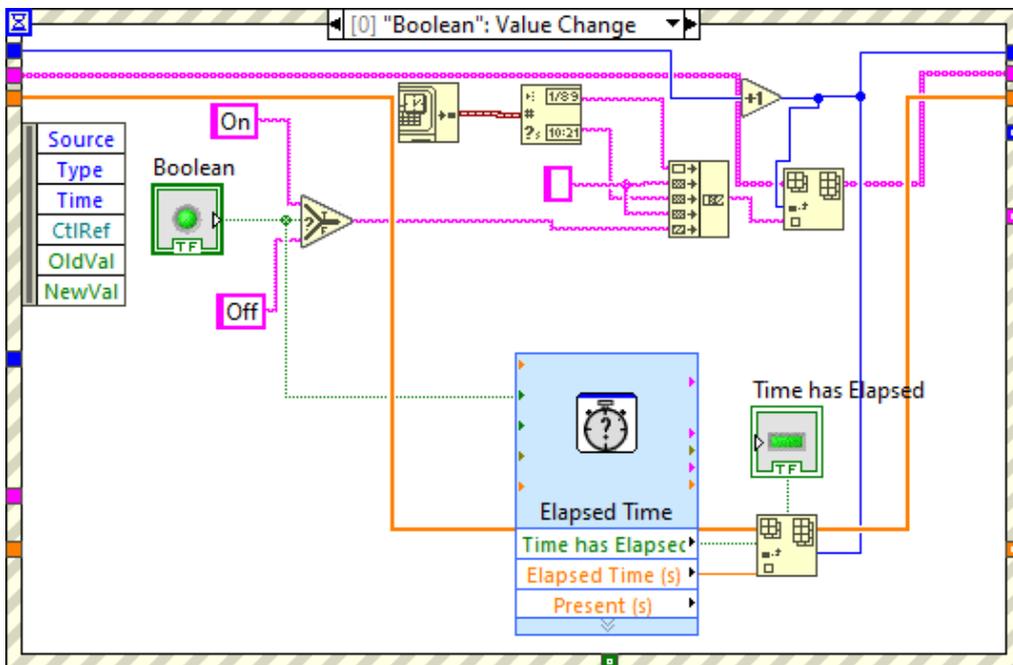


Рисунок 4.28 – Добавление индикатора к выходу «*Time Has Elapsed*» элемента «*Elapsed Time*»

Чтобы выводить строку отчетности, воспользуемся элементом «*Table Control*». Для этого на «*Front Panel*» выберем «*Controls*» → «*List, Table & Tree*» → «*Table*». Разместим его на «*Block Diagram*» внутри структуры «*While Loop*», кликнем по нему правой кнопкой мыши и выберем «*Change to Indicator*» (рисунок 4.29).

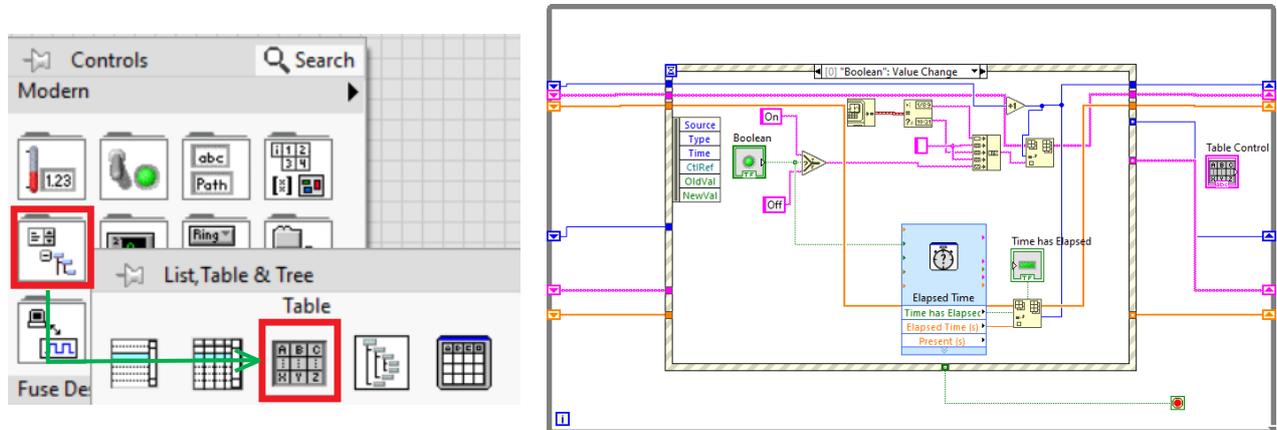


Рисунок 4.29 – Выбор элемента «*Table Control*» и добавление его на «*Block Diagram*»

Создадим массив, который будем подавать в «*Table Control*». Для этого добавим элемент «*Build Array*», расположенный в ветке «*Functions*» → «*Array*». Разместим его внутри «*While Loop*» и растянем его до четырех входов, как показано на рисунке 4.30.

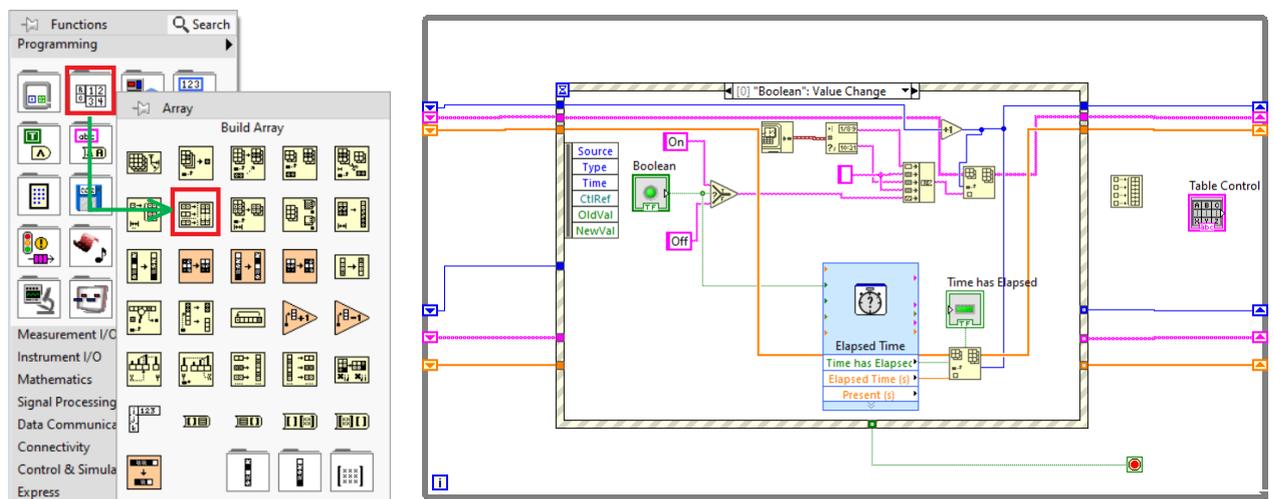


Рисунок 4.30 – Выбор элемента «*Build to Array*» и добавление его на «*Block Diagram*»

В первый вход протянем выход элемента «*Insert Into Array*», в котором формировалась строка, из события «*Boolean*». Во второй вход нужно протянуть выход «*Insert Into Array*», в котором хранится время, но так как оно относится к

типу данных «double», то нужно перевести его в тип данных «string». Для этого воспользуемся элементом «Number to Fractional String», который находится в ветке «Functions» → «String» → «Number/String Conversion» (рисунок 4.31).

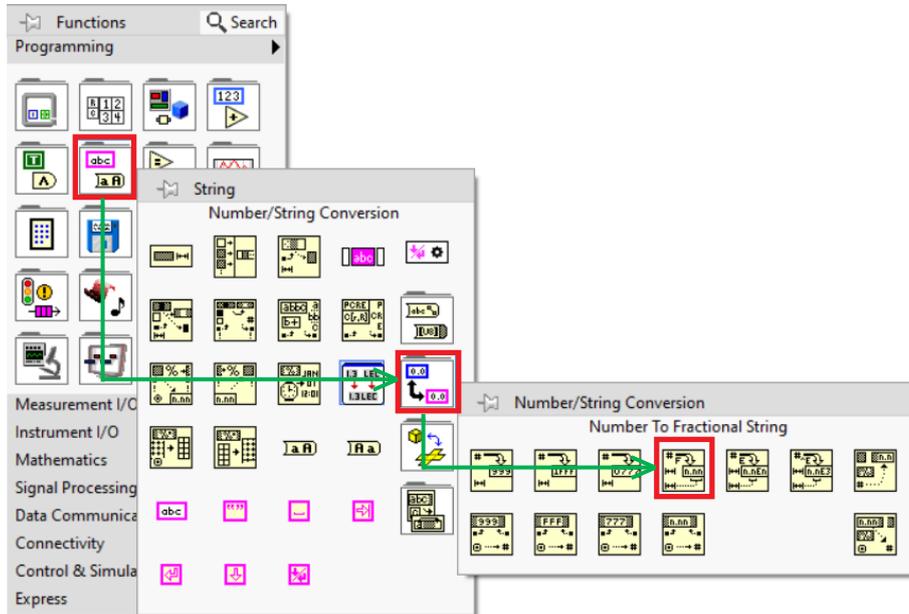


Рисунок 4.31 – Элемент «Number to Fractional String»

Соединим выход «Insert Into Array» с входом «number» элемента «Number to Fractional String», а выход «Number to Fractional String» – со вторым входом элемента «Build Array». Для третьего и четвертого входов «Build Array» проделаем аналогичные операции, но с событием «Boolean 2» (рисунок 4.32).

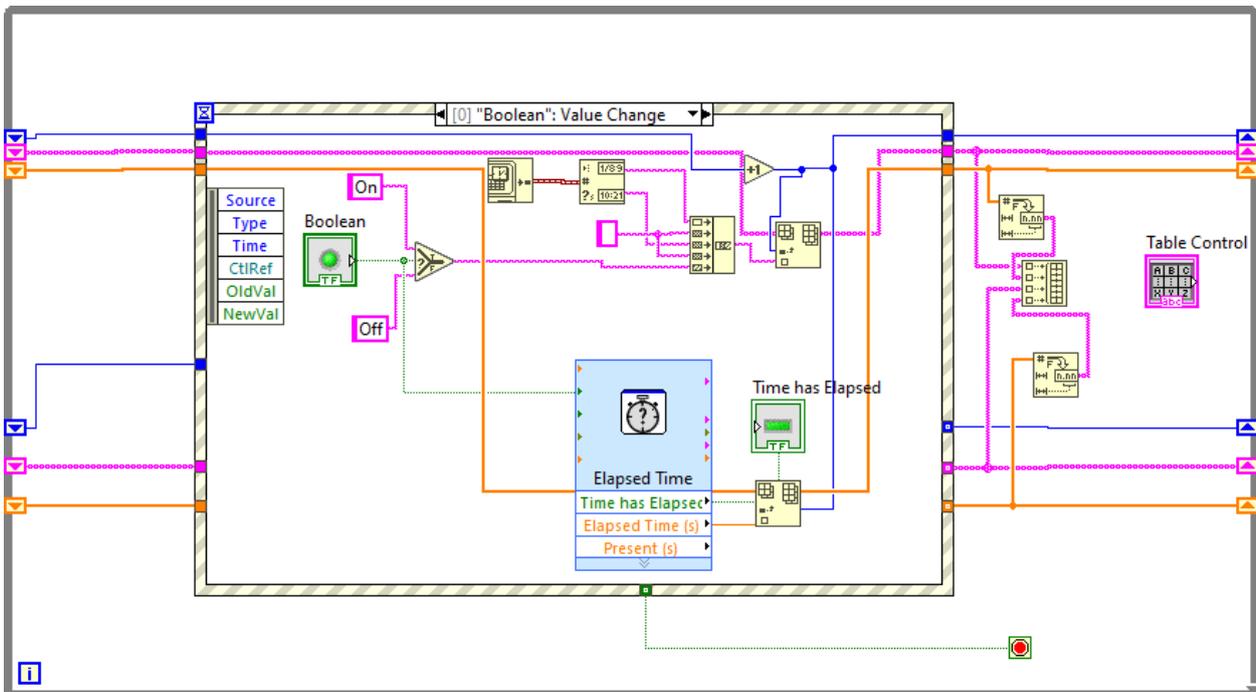


Рисунок 4.32 – Формирование массива

Для того чтобы данные в таблице отображались не в строку, необходимо транспонировать массив, добавив элемент «*Transpose 2D Array*», который находится в ветке «*Functions*» → «*Array*» (рисунок 4.33).

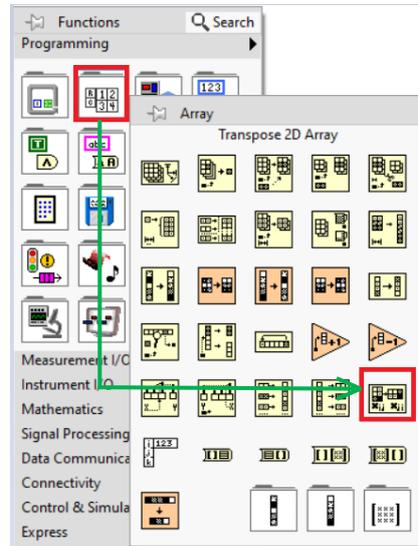


Рисунок 4.33 – Элемент «*Transpose 2D Array*»

Соединим выход «*Build Array*» с входом «*Transpose 2D Array*», а выход «*Transpose 2D Array*» – с входом «*Table Control*» (рисунок 4.34).

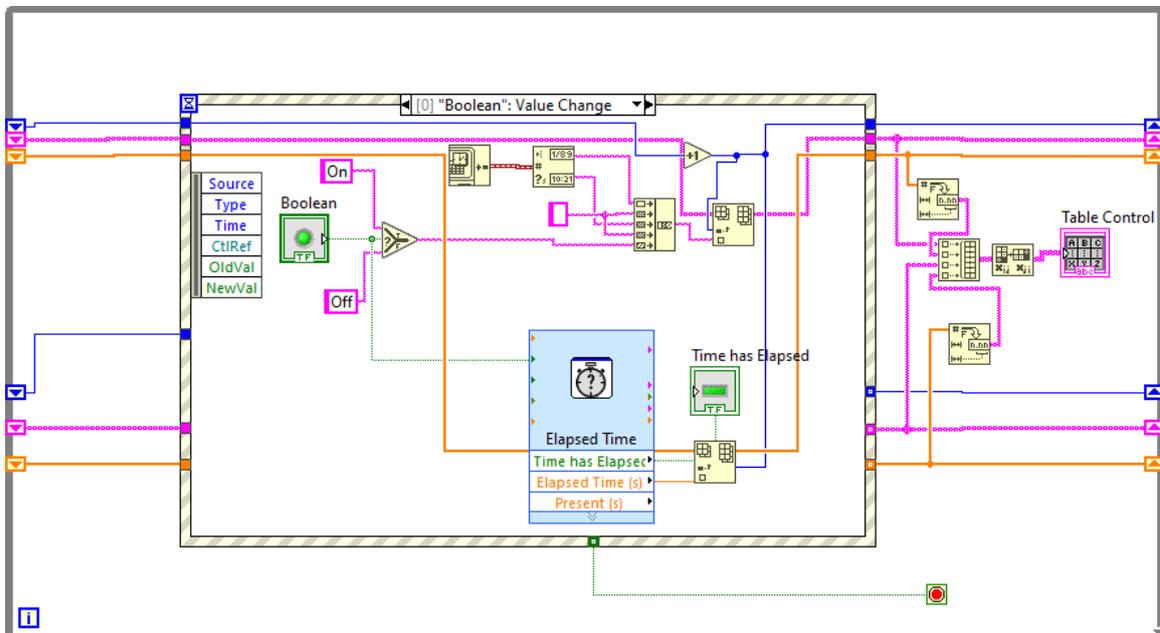


Рисунок 4.34 – Транспонирование массива

В событиях «*Boolean*» и «*Boolean 2*» кликнем правой кнопкой мыши по ручке кнопки «*stop*» и выберем «*Create*» → «*Constant*», установим значение константы *false* (*F*) (рисунок 4.35).

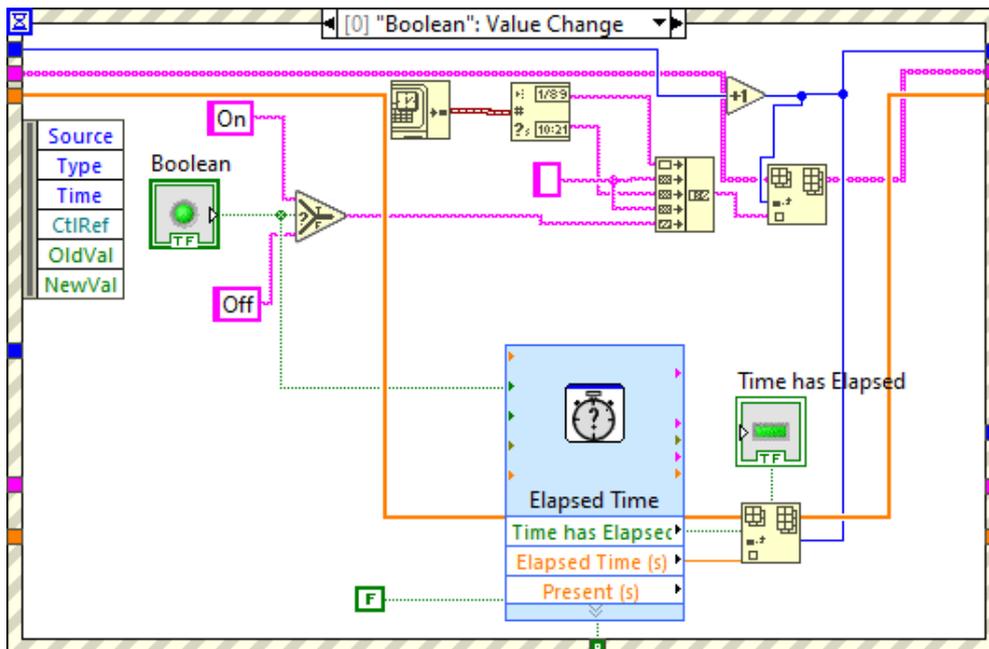


Рисунок 4.35 – Создание константы

Для того чтобы при новом запуске программы значения в таблице стирались, добавим локальные значения для кнопок. Кликаем правой кнопкой мыши по кнопке «Boolean» и выбираем «Create» → «Local Variable», выносим за пределы структуры «While Loop», создаем для него константу «false». Для этого нажимаем правой кнопкой мыши по «Local Variable», далее «Create» → «Constant», создаем «Local Variable» для второй кнопки и также соединяем с константой «false» (рисунки 4.36 – 4.38).

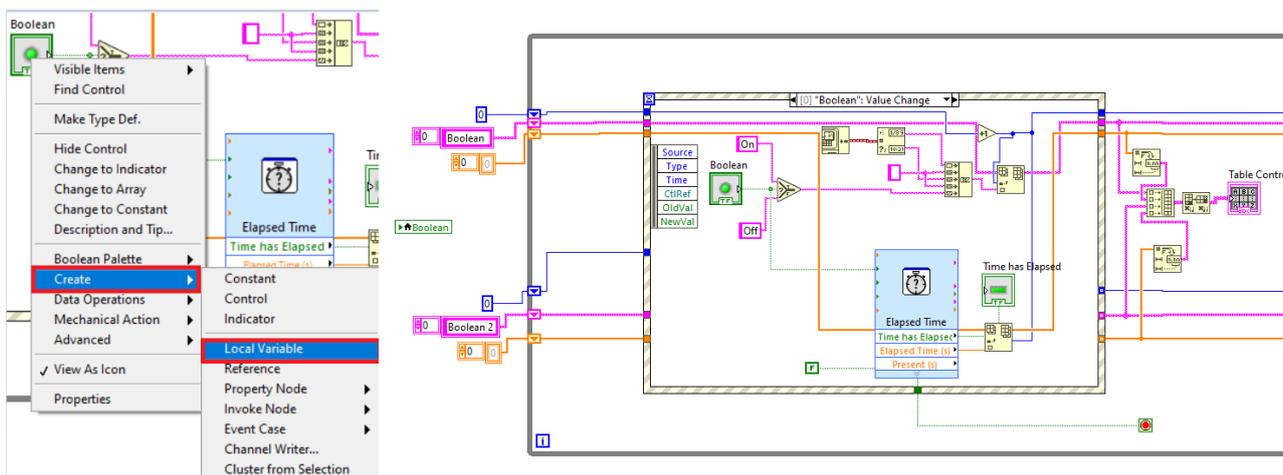


Рисунок 4.36 – Выбор элемента «Local Variable» и добавление его на «Block Diagram»

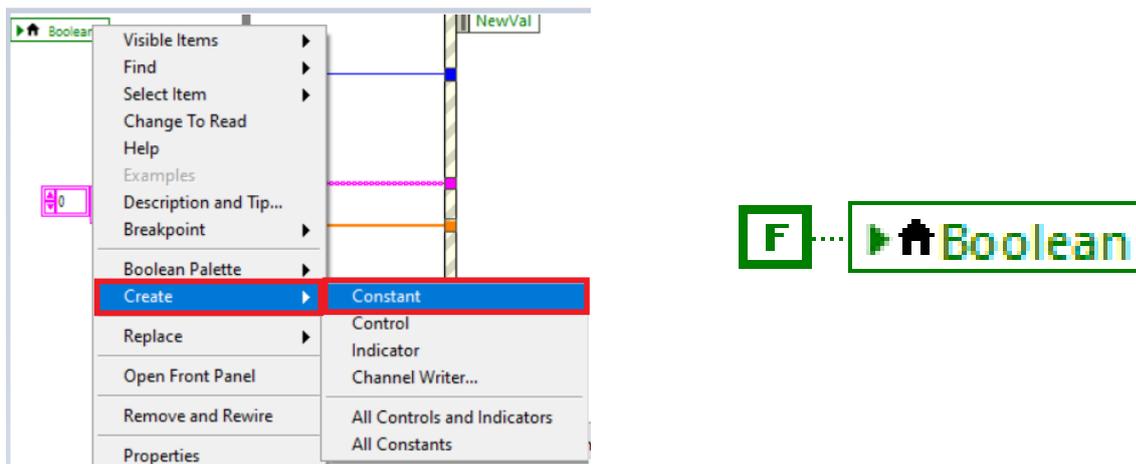


Рисунок 4.37 – Создание константы для локального значения



Рисунок 4.38 – Соединение константы с локальным значением второй кнопки

Разместим элементы на «*Front Panel*» наглядно и запустим программу, как показано на рисунке 4.39.

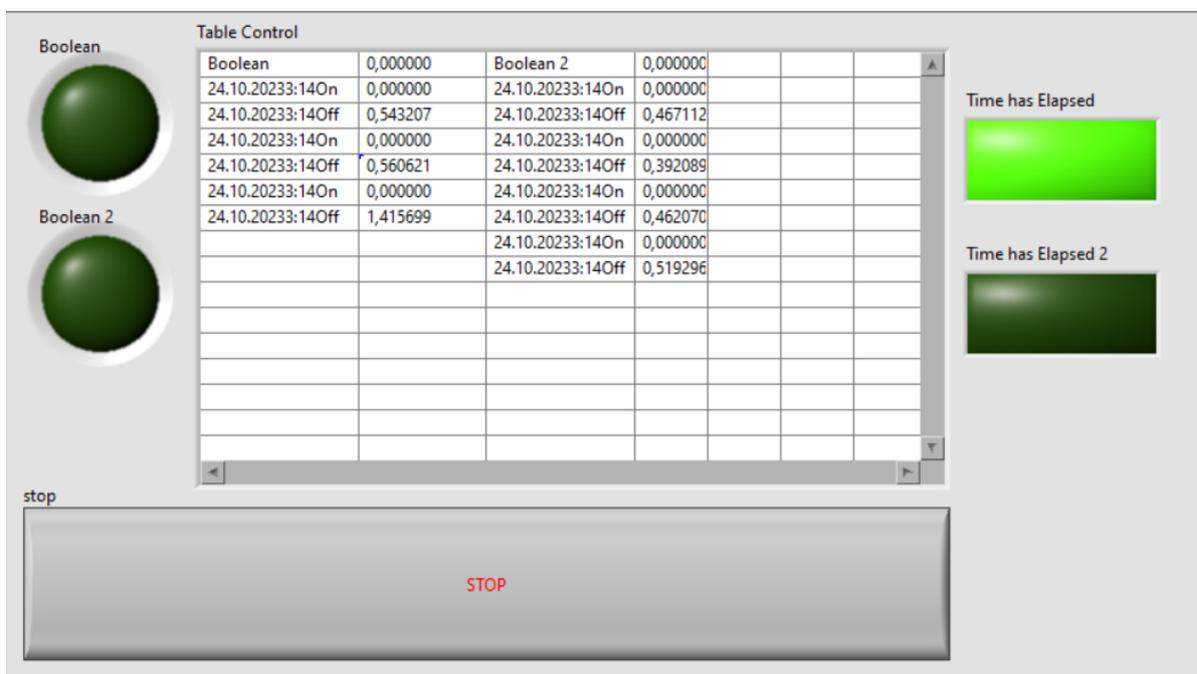


Рисунок 4.39 – Запуск программы

Первый индикатор горит, так как на «*Elapsed Time*» была установлена 1 с.

4.3 Индивидуальные задания

В рамках лабораторной работы предусмотрено выполнение заданий в соответствии с индивидуальным номером зачетной книжки каждого студента. При этом важно учесть, что повторение заданий не допускается, обеспечивая таким образом самостоятельное выполнение своей работы. В случае совпадения номера зачетной книжки с уже занятым заданием студенту предоставляется возможность выбрать свободный номер из предложенных вариантов.

Варианты заданий представлены в таблице 4.1.

Таблица 4.1 – Варианты индивидуальных заданий

Вариант	Время для лампы № 1, с	Время для лампы № 2, с
1	8	11
2	11	12
3	3	8
4	6	7
5	6	7
6	7	14
7	12	12
8	12	3
9	8	15
10	8	12
11	13	8
12	6	7
13	6	15
14	9	8
15	15	11
16	13	10
17	4	13
18	12	13
19	15	15
20	3	9
21	6	12
22	7	8
23	3	13
24	11	5
25	13	11
26	14	3
27	6	13
28	14	9
29	5	4
30	11	3

4.4 Содержание отчета

В рамках лабораторной работы № 4 необходимо выполнить одно индивидуальное задание. В связи с этим отчет по лабораторной работе должен содержать:

- 1) титульный лист;
- 2) введение:
 - 2.1) цель выполнения задания;
 - 2.2) краткое описание задания;
- 3) основная часть:
 - 3.1) ход выполнения задания;
 - 3.2) обработка и анализ результатов выполнения задания;
- 4) выводы по результатам выполнения задания.

4.5 Контрольные вопросы

- 1 Какие команды используются в среде *LabVIEW* для отладки программ?
- 2 Имеет ли какое-либо значение порядок подключения проводников к элементам суммирования и вычитания, умножения и деления?
- 3 Каким образом в программе реализована возможность учета времени включения и выключения ламп?
- 4 Какая форма графического представления результатов работы программы в наибольшей мере отражает дискретный принцип работы персонального компьютера?
- 5 В каких случаях целесообразнее использовать графики?
- 6 С помощью чего в программе реализована возможность вывода строк отчетности в таблицу?
- 7 Как можно изменить программу таким образом, чтобы она позволяла пользователю изменять время, которое учитывается в программе?
- 8 Каким образом в программе реализована возможность стирания значений в таблице при новом запуске?
- 9 Как можно изменить программу, чтобы она учитывала не только две лампы, но и любое количество ламп?

Вспомогательная литература

- 1 Трэвис, Д. *LabVIEW для всех* / Д. Тревис : пер. с англ. Н. А. Клушина – М. : ДМК Пресс ; ПриборКомплект, 2005. – 544 с.
- 2 Пилипенко, О. В. Основы программирования, математического моделирования и обработки данных в среде *LabVIEW* : практикум / О. В. Пилипенко, Н. Б. Горбачёв, М. А. Музалевская. – Орел : ОрелГТУ, 2008. – 70 с.
- 3 Евдокимов, Ю. К. *LabVIEW в научных исследованиях* / Ю. К. Евдокимов, В. Р. Линдваль, Г. И. Щербаков. – М. : ДМК Пресс, 2018. – 400 с.

Лабораторная работа № 5
Функции генерации, ввода и обработки данных в *LabVIEW*.
Интеграция *LabVIEW* и *OPC* сервера

Цель работы: разработать программу для генерации различных сигналов, добавить к полученному сигналу шум и проанализировать полученный спектр шума и сигнала, а также загрузить полученные результаты в *OPC* сервер.

Основные задачи:

- 1 Изучить функции *LabVIEW* для ввода и обработки данных во временной и частотной области.
- 2 Ознакомиться с методикой создания генератора сигналов и шумов в среде *LabVIEW*.
- 3 Научиться экспортировать полученные данные в *MS Excel*.
- 4 Получить практические навыки по интеграции *OPC* сервера со средой *LabVIEW*.
- 5 Закрепить полученные навыки и знания путем выполнения индивидуального задания.

5.1 Теоретические сведения

LabVIEW предоставляет широкий набор функциональных возможностей для отладки сложных программ, тестирования реальных систем измерения и регулирования и развернутого анализа получаемых данных. Так функции генерации сигналов и шумов используются для формирования детерминированных и случайных сигналов с заданным набором параметров.

Генерация сигналов, процесс создания электрических или электромагнитных волн с определенными характеристиками, широко применяется в различных областях. В физике генерация сигналов используется для создания стандартизированных волн, применяемых в экспериментах для проверки теоретических предсказаний и изучения физических законов. В электронике генерация сигналов необходима для тестирования и отладки устройств, таких как усилители, фильтры и аналого-цифровые преобразователи. В обработке сигналов генерация сигналов с различными характеристиками, включая шум, используется для обучения и тестирования алгоритмов обработки, таких как фильтрация, декодирование и распознавание образов. Аудиоиндустрия использует генерацию сигналов для создания звуков и музыкальных эффектов, где синтезаторы и цифровые аудиопроцессоры создают уникальные звуковые образы.

В общем генерация сигналов является важным инструментом в научных и технических приложениях, обеспечивая контролируемую среду для экспериментов, тестирования и исследований в различных областях.

Имеющиеся в *LabVIEW* генераторы сигналов можно подразделить на многофункциональные программно-регулируемые генераторы сигналов с широким

набором контролируемых параметров; приборы, предназначенные для генерации наиболее широко применяемых детерминированных периодических сигналов, а также для генерации шумов с различными законами амплитудного и спектрального распределения.

Среди них генератор с заданной длительностью сигналов; гармонические колебания и шум; отрезки синусоидального, импульсного, пилообразного, $\sin(x)/y$, прямоугольного и частотно-модулированного сигналов; синусоидальные, треугольные, прямоугольные, пилообразные и произвольные колебания любой длительности; равномерный, гауссовский, периодический случайный шум и двоичная последовательность максимальной длины; гамма-шум, пуассоновский, биномиальный шум, шум Бернулли.

5.2 Задания для выполнения лабораторной работы

Задание 1. Создать генератор сигналов с заданными в варианте типом сигнала, длительностью, частотой, амплитудой, начальным значением фазы. Подать выходной сигнал на однолучевой осциллограф. Добавить к полученному сигналу шум, просмотреть спектр шума и сигнала, определить смещение. Сохранить полученные данные в *MS Excel*, построить график и описать свойства сигнала.

Порядок выполнения задания 1

В «*Functions palette*» на «*Block Diagram*» необходимо выбрать функцию «*SigGen Duration*», которая находится в ветке «*Function*» → «*Signal Processing*» → «*Sig Generation*». Для получения информации по какому-либо виртуальному прибору требуется активизировать его изображение при помощи левой кнопки мыши и выбрать пункт меню «*Help*» (рисунок 5.1).

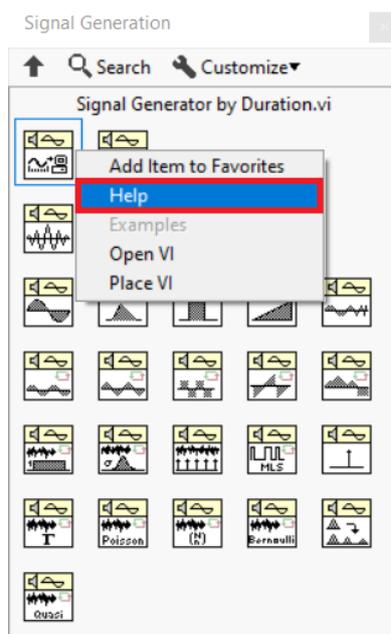


Рисунок 5.1 – Выбор генератора сигналов с заданной длительностью

Рассматриваемый виртуальный прибор имеет девять входов и три выхода. Входы: сбросить фазу, длительность, тип сигнала, число выборок, частота, амплитуда, постоянная составляющая, вход фазы, заполнение цикла прямоугольного колебания (%). Выходы: сигнал, частота выборок, выход фазы.

Он генерирует «*signal*», имеющий форму, задаваемую на входе «*signal type*». Вход «*reset phase*» определяет начальную фазу выходного сигнала. По умолчанию на входе установлено состояние «ИСТИНА». При этом начальная фаза сигнала устанавливается в соответствии со значением на входе «*phase in*». Если на входе «*reset phase*» установлено состояние «ЛОЖЬ», то начальная фаза устанавливается равной значению фазы на «*phase out*» при последнем выполнении этого ВП (рисунок 5.2).

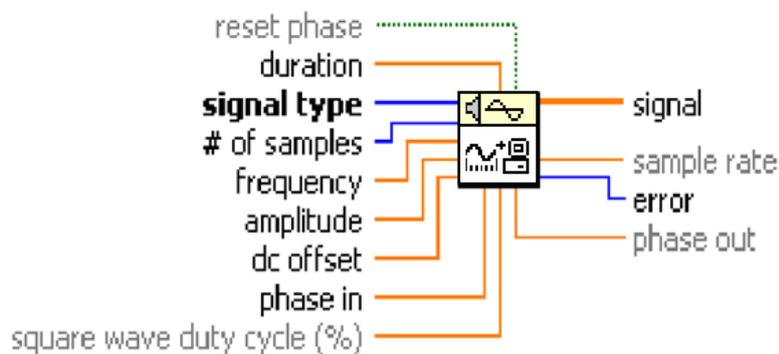


Рисунок 5.2 – Генератор сигналов с заданной длительностью

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: 0 – синусоидальный, 1 – косинусоидальный, 2 – треугольный, 3 – прямоугольный, 4 – пилообразный, 5 – линейно нарастающий, 6 – линейно спадающий.

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 < число выборок / (2*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 «фаза» («*phase in*») определяет начальную фазу (в градусах) выходного сигнала. По умолчанию значение на входе фазы равно «0».

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

Выход представляет сгенерированный массив выборок сигнала.

Добавим структуру «*While Loop*»: «*Functions*» → «*Structures*» → «*While Loop*» (рисунок 5.3). Перенесем туда «*Signal Generation by Duration*».

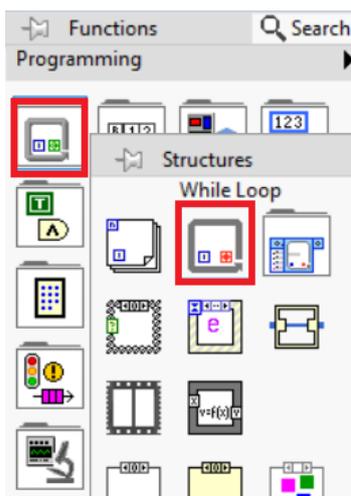


Рисунок 5.3 – Выбор структуры «*While Loop*»

Щелкнем правой кнопкой мыши и добавим все индикаторы и управляющие элементы для «*Signal Generation by Duration*» (рисунок 5.4).

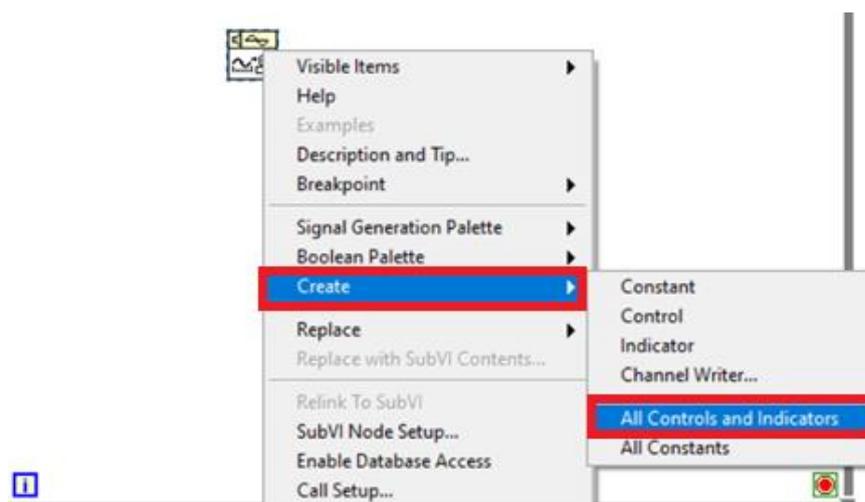


Рисунок 5.4 – Создание управляющих элементов и индикаторов для «*Signal Generation by Duration*»

Добавим «*Waveform Graph*». На «*Front Panel*» открываем «*Controls*» → «*Graph*» → «*Waveform Graph*» (рисунок 5.5).

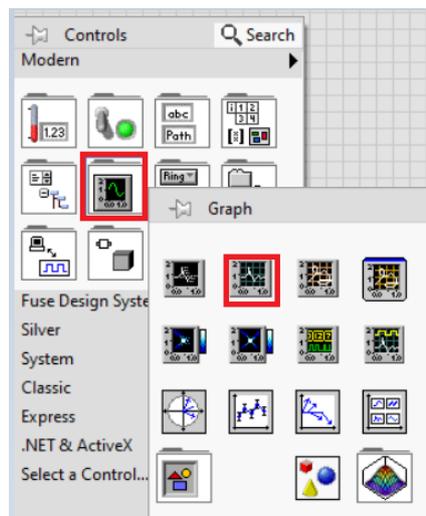


Рисунок 5.5 – Выбор «*Waveform Graph*»

На «*Block Diagram*» соединим выход «*signal*» элемента «*Generation by Duration*» и вход «*Waveform Graph*». Для этого предварительно удаляем индикатор «*signal*» с помощью выделения и последующего нажатия кнопки «*Delete*», после чего связываем выход «*signal*» с входом «*Waveform Graph*» напрямую, как показано на рисунке 5.6.

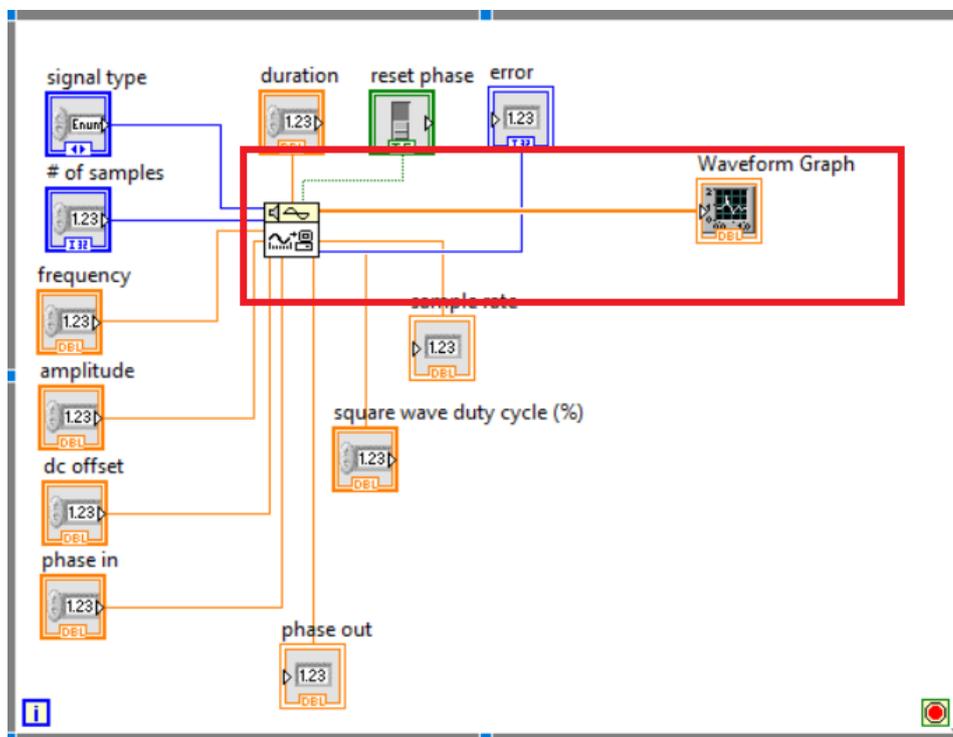


Рисунок 5.6 – Соединение выхода «*signal*» элемента «*Generation by Duration*» и входа «*Waveform Graph*»

Добавим кнопку для завершения цикла «*Loop Condition*» → «*Create Control*» (рисунок 5.7).

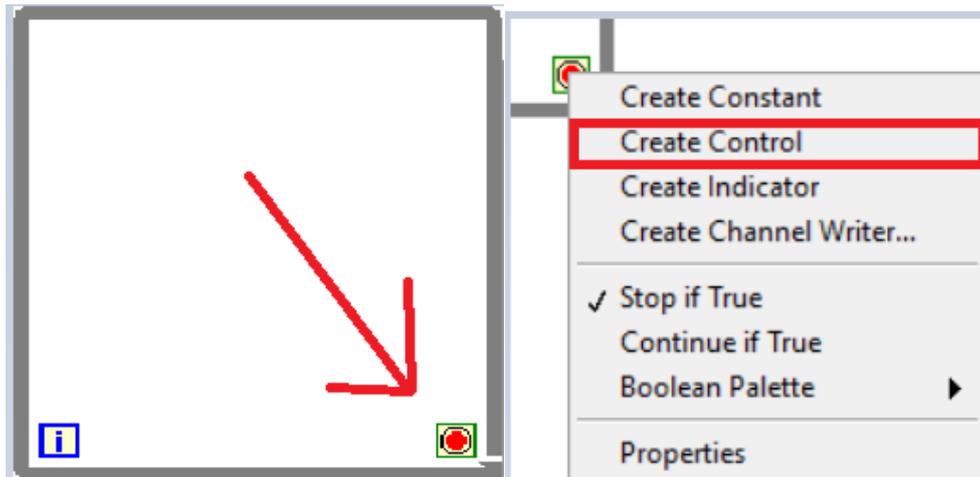


Рисунок 5.7 – Создание кнопки завершения цикла

Размещаем на «*Front Panel*» управляющие элементы и индикаторы как удобно, заполняем поля: тип сигнала («*signal type*»), длительность («*duration*»), число выборок («*# of samples*»), частотой («*frequency*»), амплитудой («*amplitude*»), постоянным смещением («*dc offset*»), начальным значением фазы («*phase in*») в соответствии с вариантом и запускаем программу (рисунок 5.8).

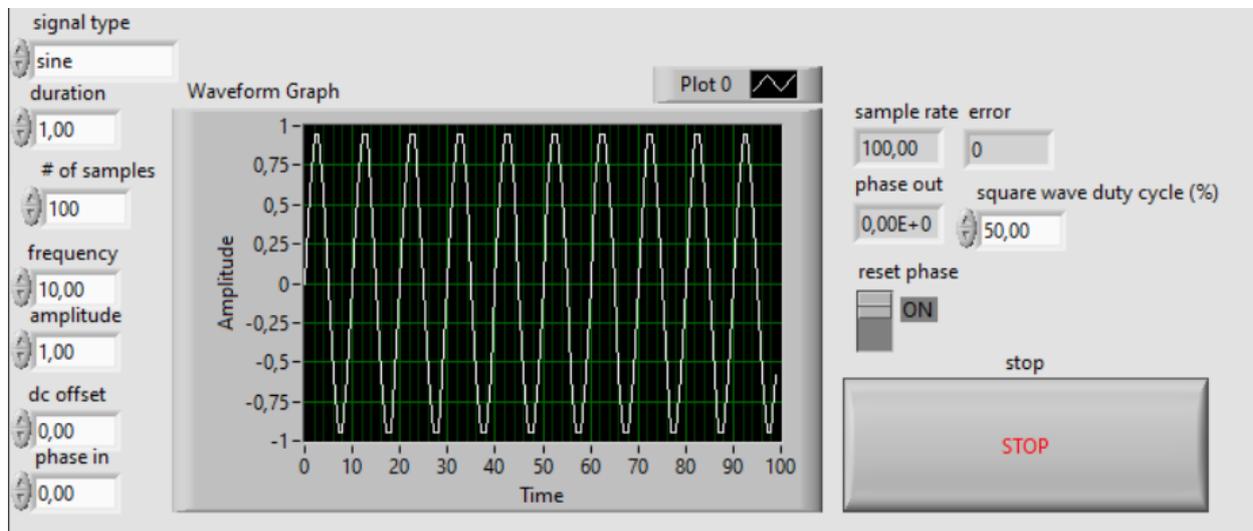


Рисунок 5.8 – Размещенные на «*Front Panel*» элементы для вывода сигнала на осциллограф

Добавим к сигналу шум. Для этого возвращаемся на «*Block Diagram*», из панели «*Sig Generation*» выбираем генератор шума согласно варианту и вносим его на «*Block Diagram*», после добавим к нашему сигналу шум с помощью функции «*Functions*» → «*Numeric*» → «*Add*» (рисунки 5.9 – 5.11).

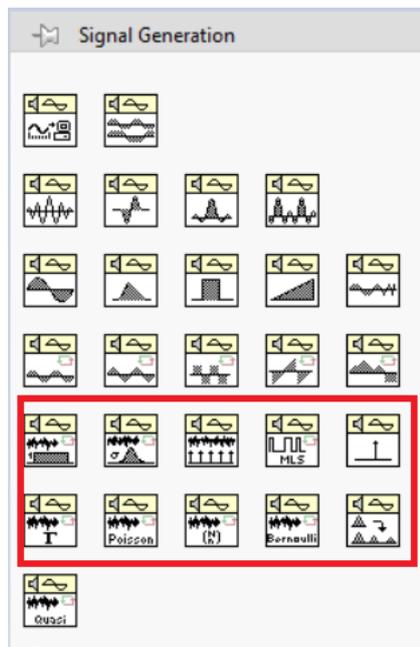


Рисунок 5.9 – Выбор генератора шума

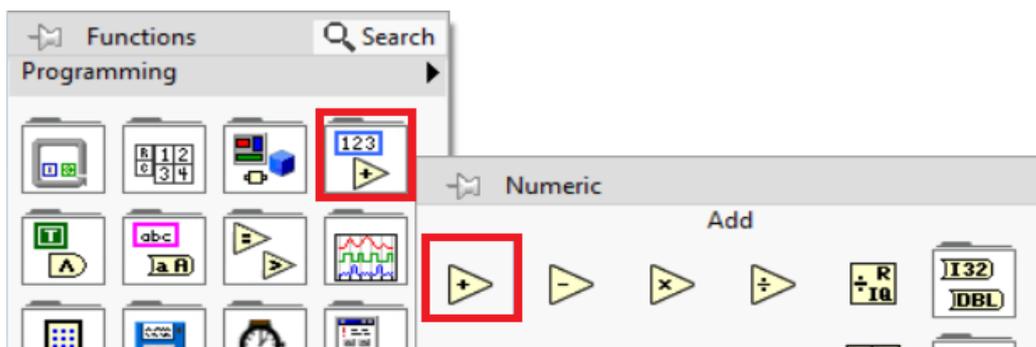


Рисунок 5.10 – Выбор функции «Add»

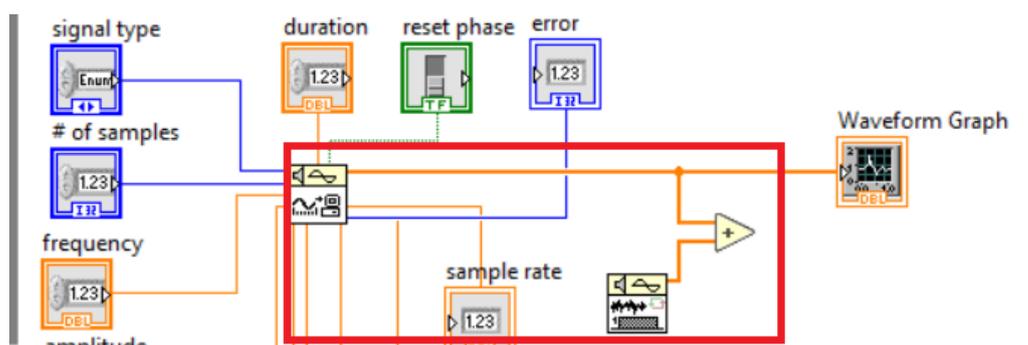


Рисунок 5.11 – Добавление шума к сигналу на «Block Diagram»

Добавим управляющий элемент для амплитуды шума. Для этого наведем курсор мыши на вход «*amplitude*» и нажатием правой кнопки мыши вызовем контекстное меню. В меню выбираем «*Create*» → «*Control*», как показано на рисунке 5.12.

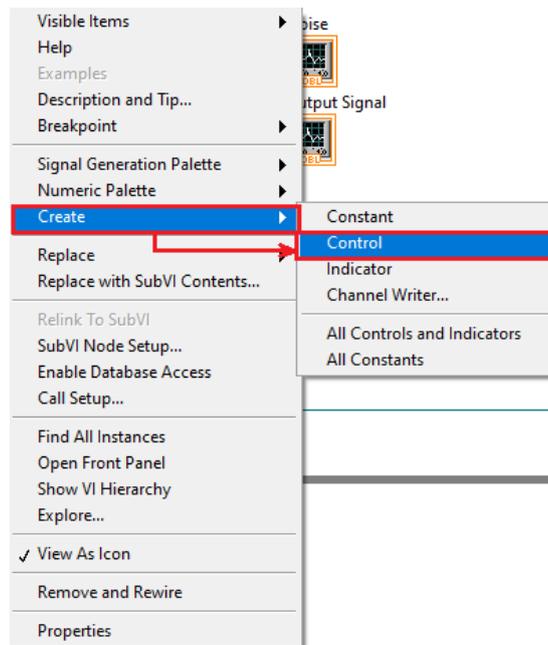


Рисунок 5.12 – Добавление управляющего элемента

Добавим на «*Block Diagram*» функцию для отображения спектра измененного сигнала. Для этого подходит функция «*Auto Pwr Spectr*», которая находится в ветке «*Functions*» → «*Signal Processing*» → «*Spectral*» (рисунок 5.13).

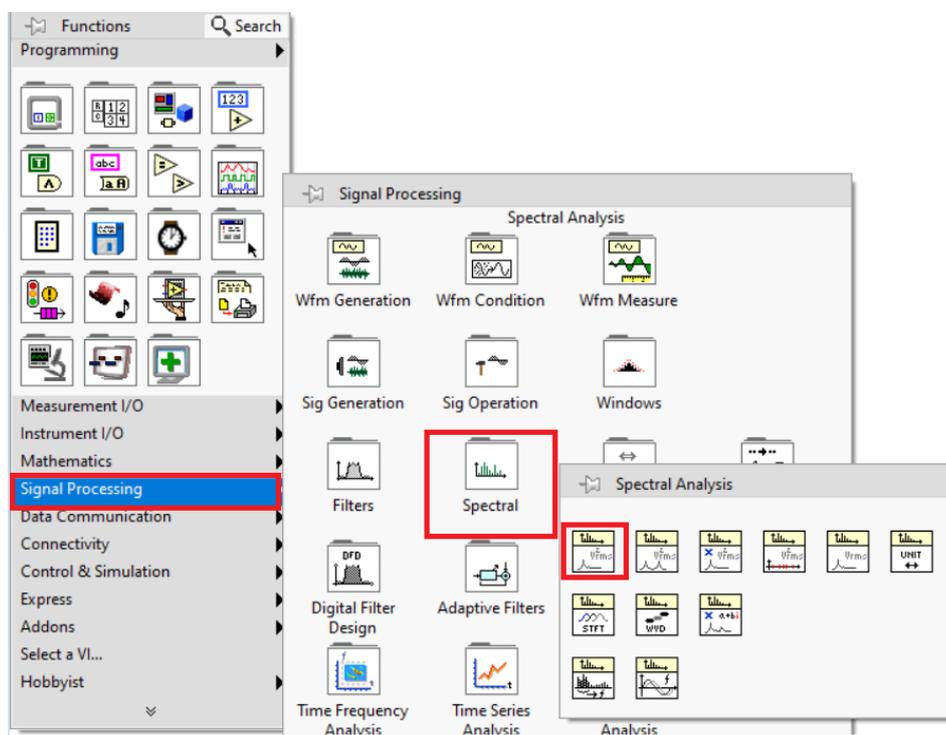


Рисунок 5.13 – Выбор функции «*Auto Pwr Spectr*»

Применим данную функцию к выходному сигналу (рисунок 5.14).

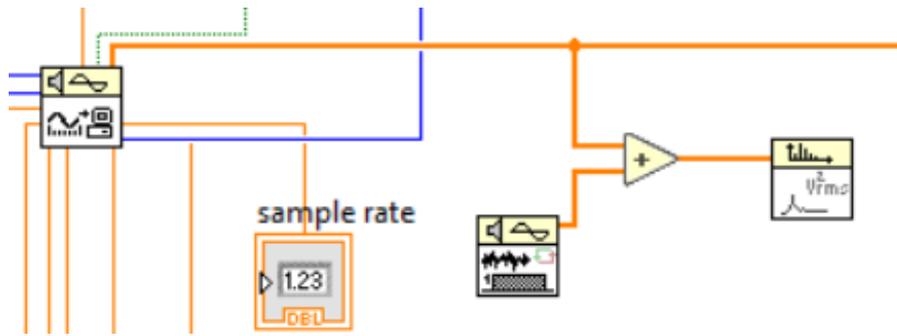


Рисунок 5.14 – Применение функции «*Auto Pwr Spectr*» к выходному сигналу

Добавим еще три осциллографа («*Waveform Graph*») для отображения шума, выходного сигнала и спектра выходного сигнала, как показано на рисунках 5.15 и 5.16.

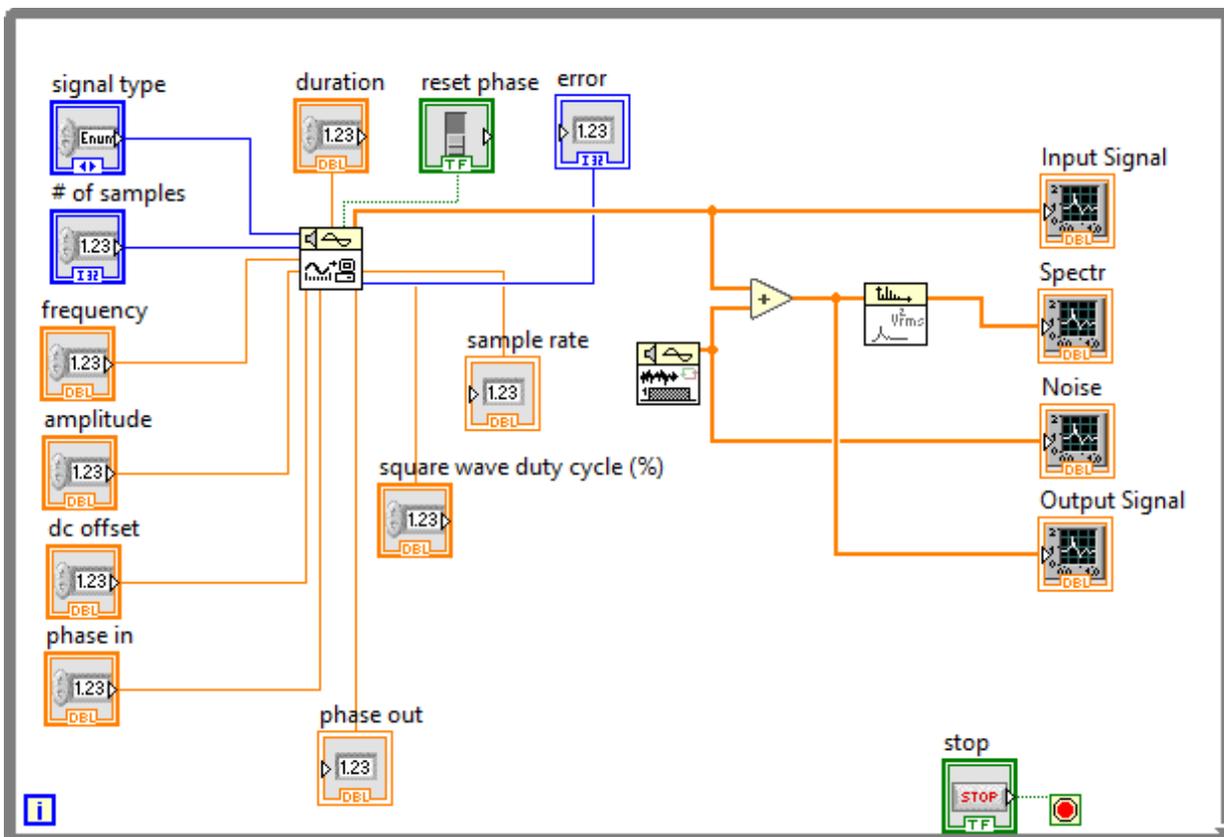


Рисунок 5.15 – Окно «*Block Diagram*» разработанного генератора сигналов

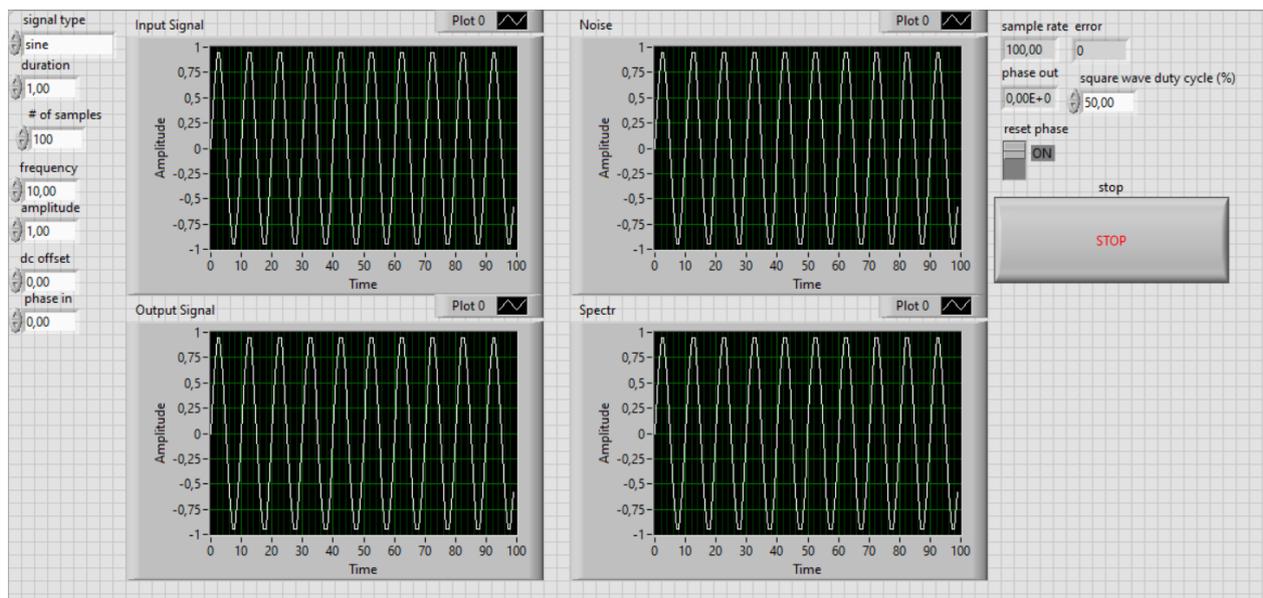


Рисунок 5.16 – Окно «*Front Panel*» разработанного генератора сигналов

Запустите программу и можно будет увидеть, как отображаются данные в «*Waveform Graph*» (рисунок 5.17).

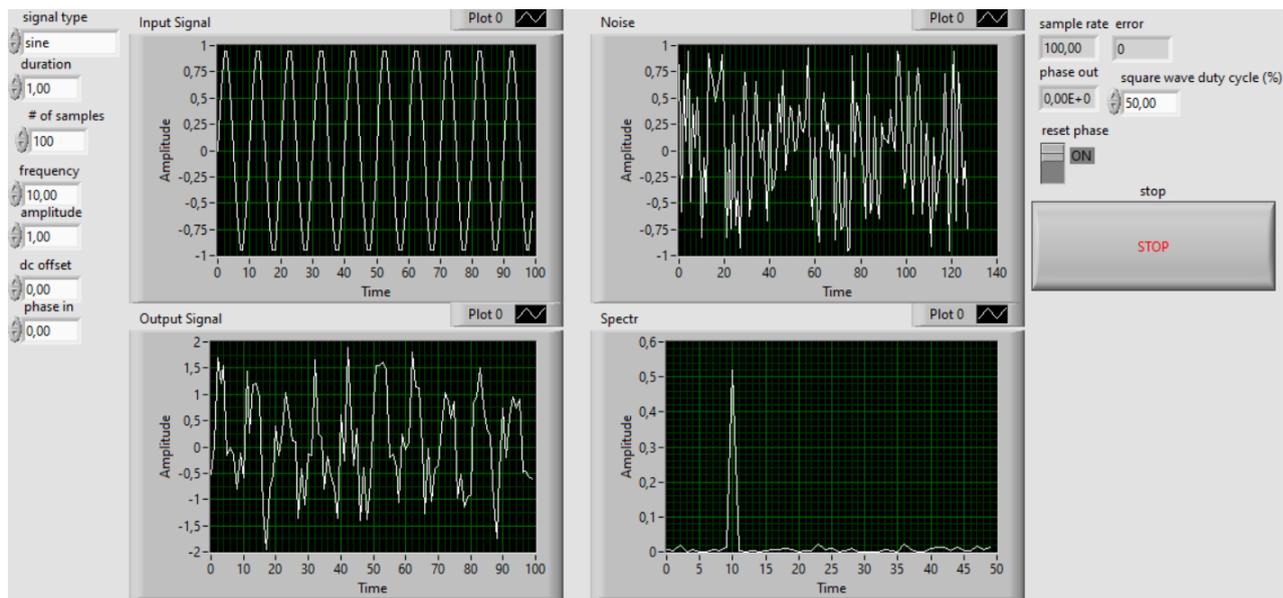


Рисунок 5.17 – Запуск генератора сигналов с отображением исходного сигнала, шума, выходного сигнала и спектра выходного сигнала

Сохраните результаты выходного сигнала в *MS Excel*. Для этого щелкните правой кнопкой мыши по осциллографу, на котором отображается выходной сигнал, выберите «*Export*» → «*Export Data To Excel*» (рисунок 5.18).

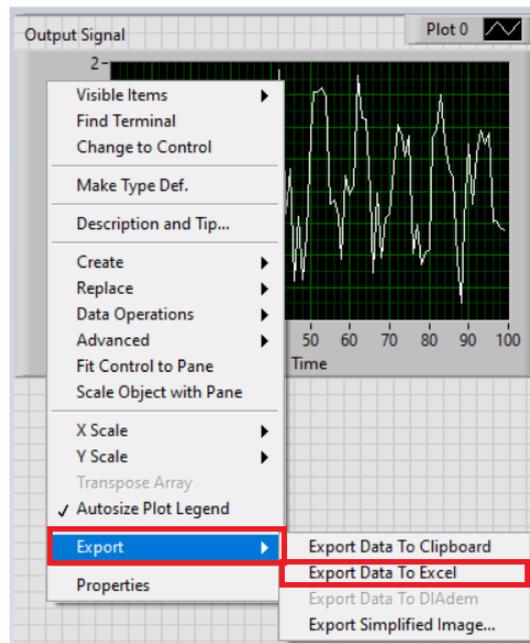


Рисунок 5.18 – Сохранение данных в MS Excel

С помощью экспортированных данных построим график и опишем свойства сигнала. Для этого выделим все экспортированные данные, далее над панелью инструментов выберите «Вставка» → «Вставить точечную (X, Y) или пузырьковую диаграмму» → «Точечная с гладкими кривыми и маркерами», как показано на рисунке 5.19.

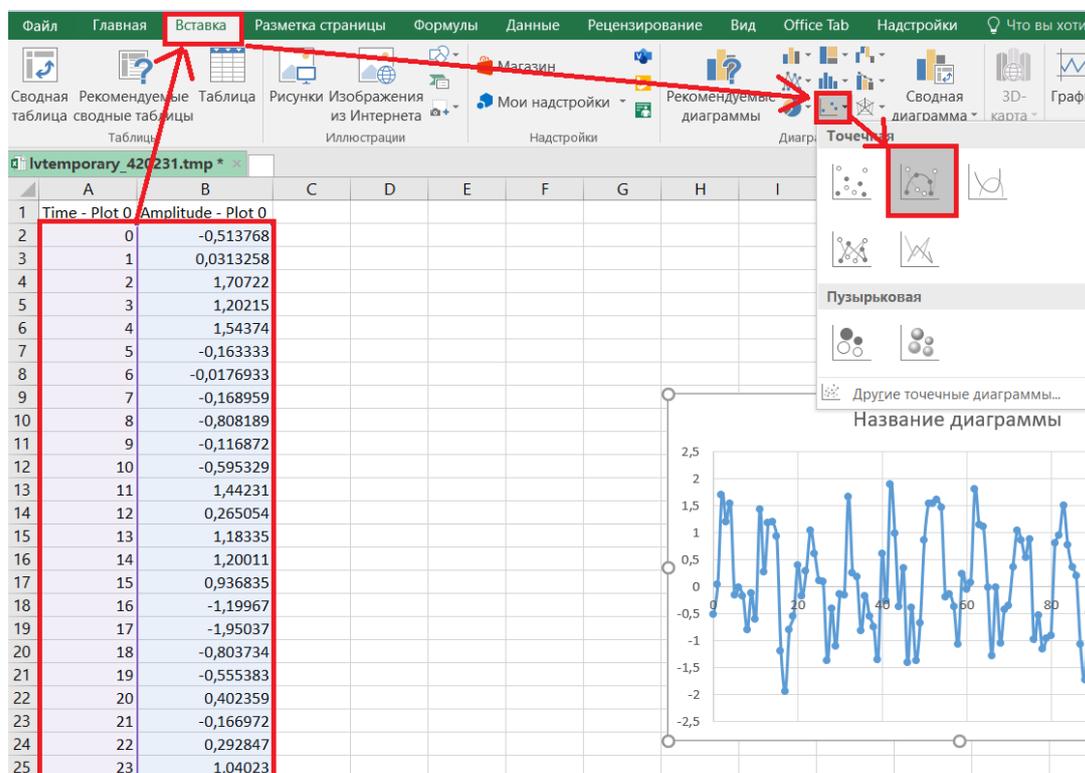


Рисунок 5.19 – Построение графика по экспортированным данным

Задание 2. Использовать генератор, созданный по заданию 1 для интеграции с *OPC* сервером. Используя набор функций, который предлагает *LabVIEW*, загрузить данные генератора сигнала из задания 1 в *OPC* сервер.

Порядок выполнения задания 2

Открываем *OPC* сервер в «*MasterOPC Universal Modbus Server*» (приложение В). Создадим сервер, для этого кликаем по полю «Создать» (рисунок 5.20).

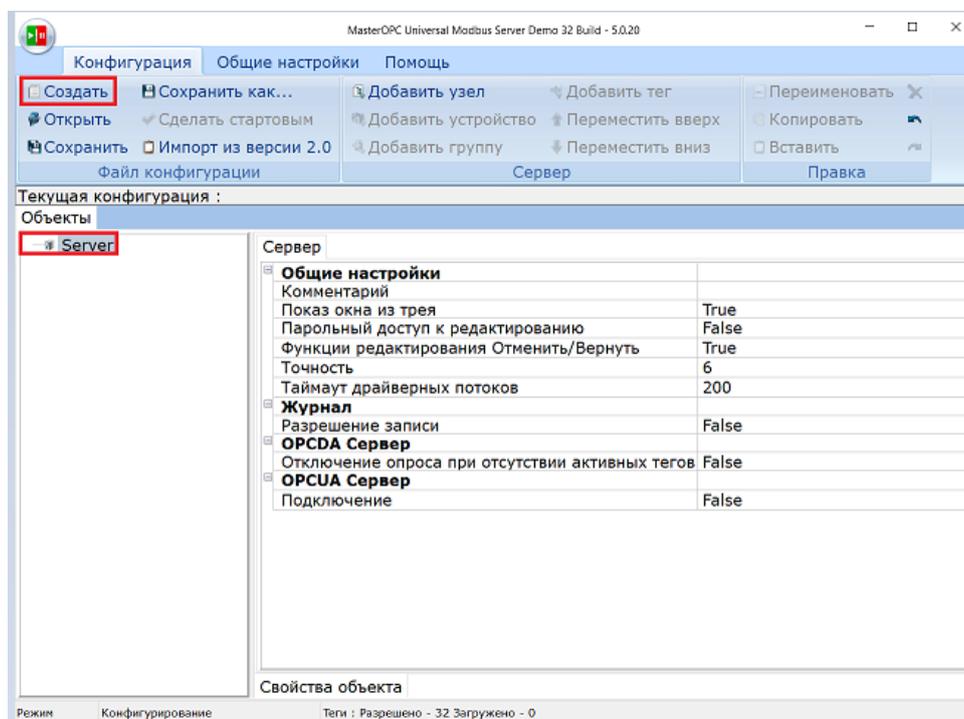


Рисунок 5.20 – Создание *OPC* сервера

ВАЖНО ЗНАТЬ!

Сервер OPC предоставляет единый согласованный интерфейс для связи с несколькими устройствами через стандарт *OPC*.

OPC (Object Linking and Embedding for Process Control) – это набор повсеместно принятых спецификаций, предоставляющих универсальный механизм обмена данными в системах контроля и управления.

Стандарт *OPC* может быть использован не только для взаимодействия *SCADA* с «железом», но и для обмена данными с любым источником данных, например, с различными базами данных.

Добавим в сервер коммуникационный узел, кликаем по нему правой кнопкой мыши, нажимаем «Добавить» и далее «Коммуникационный узел», появится окно с настройками, оставляем все настройки и меняем название на «Коммуникационный узел» и нажимаем кнопку «Да» (рисунок 5.21).

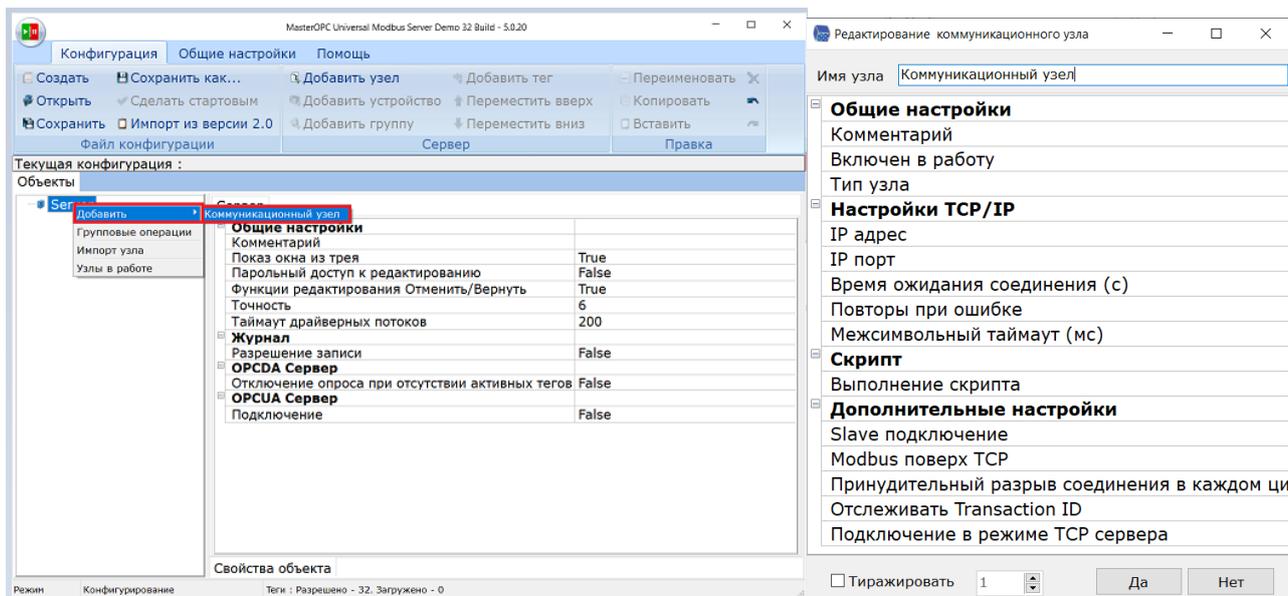


Рисунок 5.21 – Создание коммуникационного узла

Добавим устройство в коммуникационный узел, для этого кликаем правой кнопкой мыши по узлу, выбираем «Добавить» и далее «Устройство», появится окно настроек, все оставляем, меняем название на «Устройство» и нажимаем кнопку «Да» (рисунок 5.22).

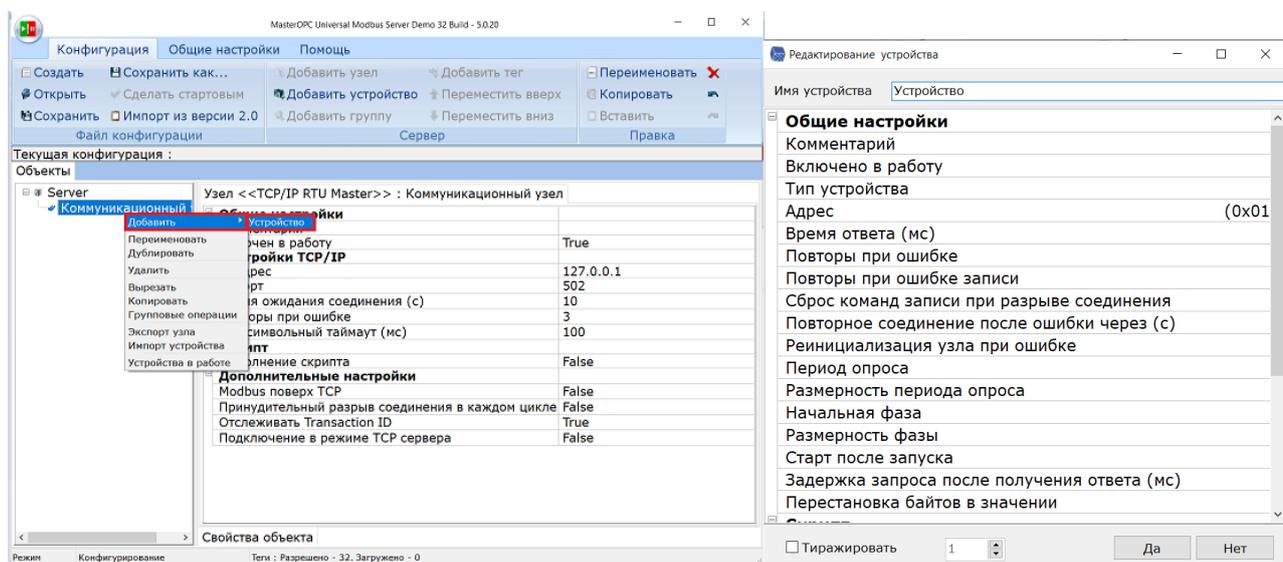


Рисунок 5.22 – Создание устройства в коммуникационном узле

Добавим в устройство тег, для этого кликаем правой кнопкой мыши по устройству и выбираем «Добавить» и далее «Тег», появится окно настроек, проверяем, чтобы тип доступа был «ReadWrite» (растяните окно, чтобы увидеть параметры), все оставляем, меняем название на «Сигнал», как показано на рисунке 5.23.

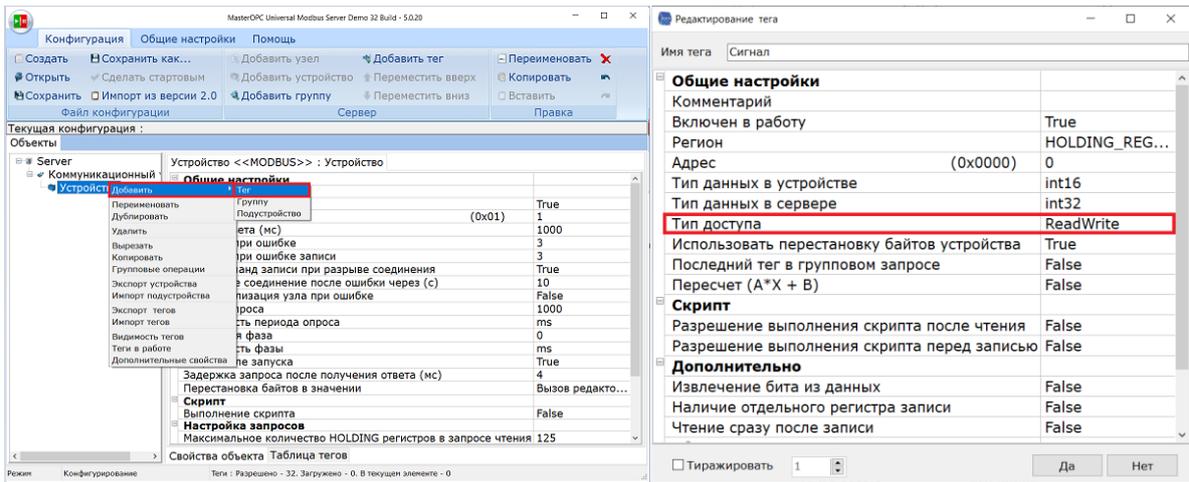


Рисунок 5.23 – Создание тега и проверка типа доступа

Открываем проект, созданный в задании 1.

Для выполнения задания необходимо настроить подключение к OPC серверу. Для этого надо выбрать «Functions», далее выбрать «Data Communication», нажать на «DataSocket» и выбрать три элемента: «DataSocket Open», «DataSocket Write», «DataSocket Close» (рисунок 5.24).

Элемент «DataSocket Open» расположим слева от цикла «While Loop», элемент «DataSocket Write» – внутри цикла, а элемент «DataSocket Close» – справа от цикла.

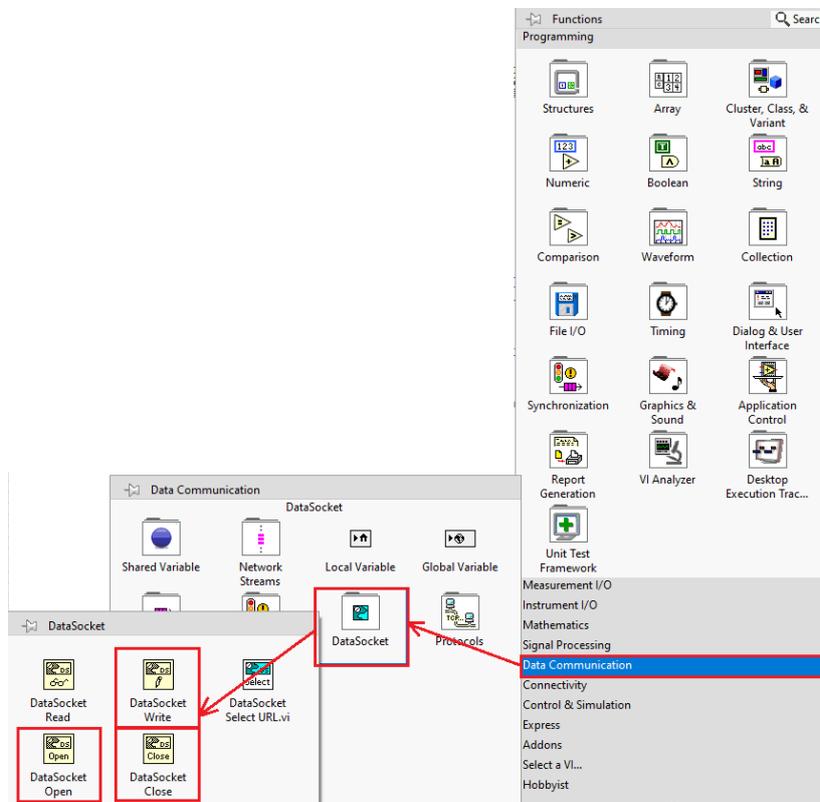


Рисунок 5.24 – Палитра «DataSocket»

Далее нажатием правой кнопкой мыши по контуру цикла «*While Loop*» добавим сдвиговой регистр («*Add Shift Register*»). Соединим выход «*connection id*» элемента «*DataSocket Open*» со сдвиговым регистром, далее соединим сдвиговой регистр с входом «*connection in*» элемента «*DataSocket Write*», а выход этого элемента «*connection out*» соединим со сдвиговым регистром справа. Правый сдвиговой регистр соединяем с входом «*connection id*» элемента «*DataSocket Close*» (рисунок 5.25).



Рисунок 5.25 – Вид соединения элементов

Следующим шагом будет подключение элемента «*DataSocket Select URL.vi*» для выбора *OPC* сервера. Для этого откроем окно «*Functions*», выберем «*Data Communications*» → «*DataSocket*» → «*DataSocket Select URL.vi*» (рисунок 5.26). Соединим выход «*URL*» этого элемента с входом «*URL*» элемента «*DataSocket Open*».

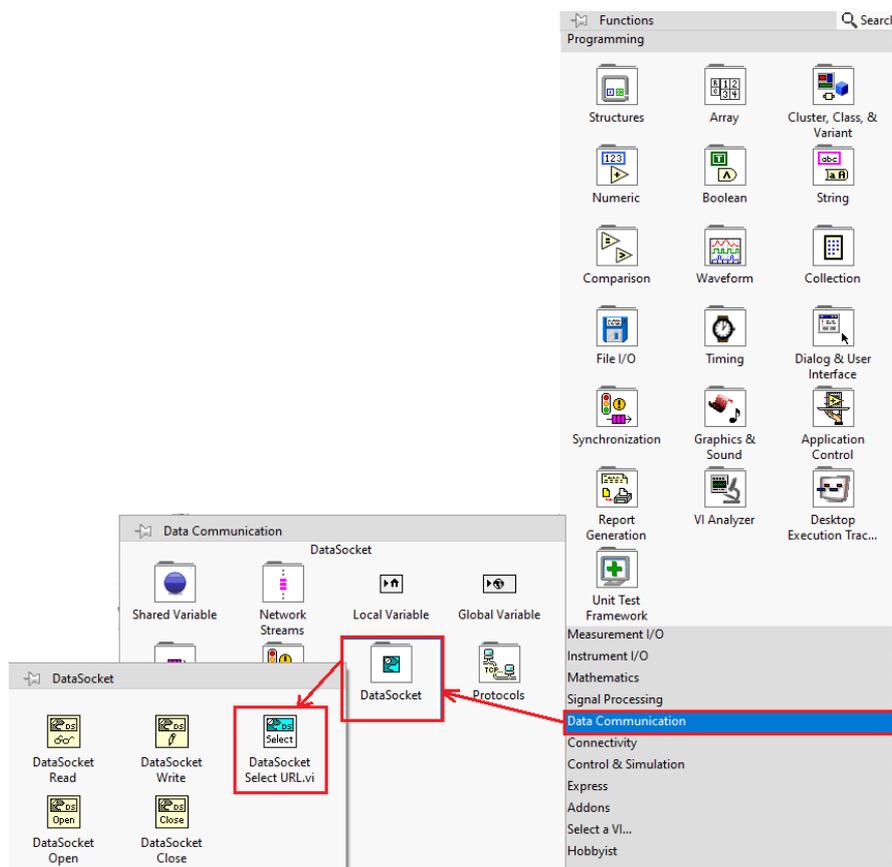


Рисунок 5.26 – Выбор элемента «*DataSocket Select URL.vi*»

Теперь для выбора режима работы с данными создадим константу перечисления. Откроем окно «*Functions*», выберем «*Numeric*» и перенесем на «*Block Diagram*» элемент «*Enum Constant*» (рисунок 5.27).

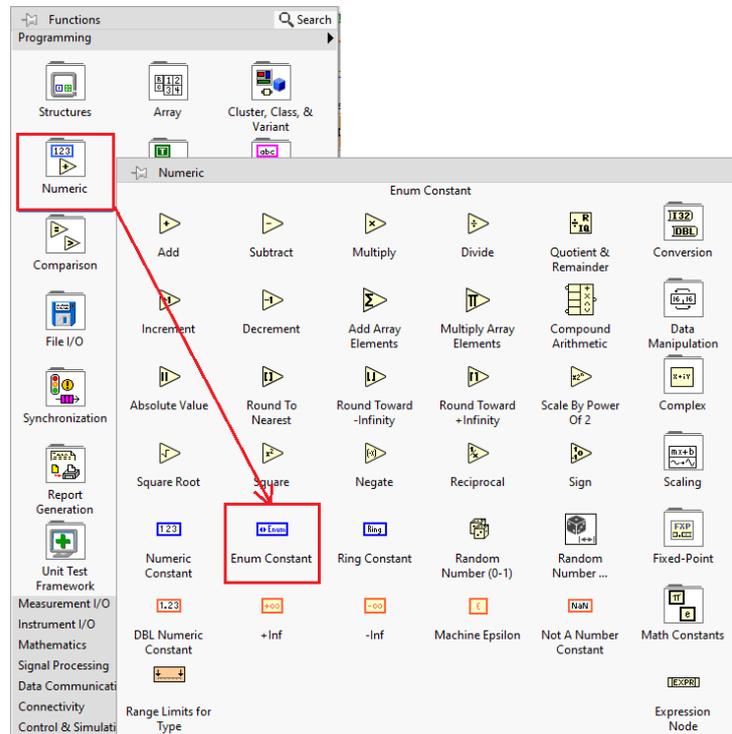


Рисунок 5.27 – Выбор элемента «*Enum Constant*»

Теперь необходимо настроить элемент «*Enum Constant*». Правой кнопкой мыши вызовем контекстное меню и выберем «*Properties*». Далее переходим на вкладку «*Edit Items*» и добавляем константы, как показано на рисунке 5.28. После чего соединяем «*Enum Constant*» с входом «*mode*» элемента «*DataSocket Open*».

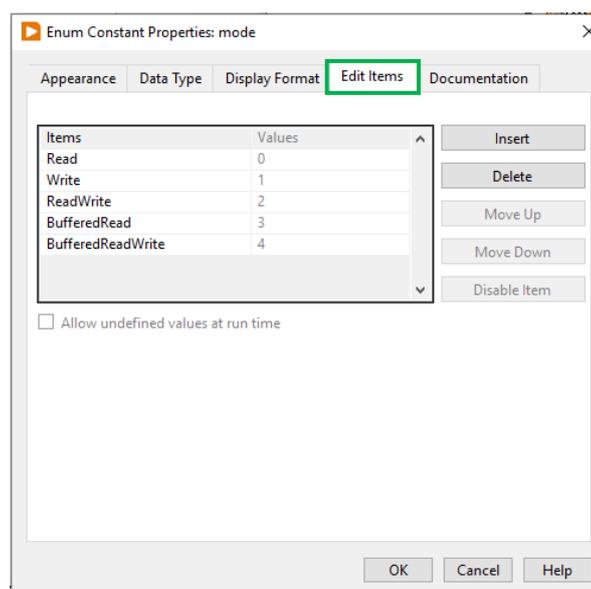


Рисунок 5.28 – Параметры элемента «*Enum Constant*»

Далее необходимо подать значение, которое следует записать в *OPC* сервер. Для начала добавим элемент для перевода данных в 32-битное целое число. Откроем окно «*Functions*», выберем «*Numeric*», затем выбираем «*Conversion*» и перетаскиваем внутрь цикла «*While Loop*» элемент «*To Long Integer*», как показано на рисунке 5.29.

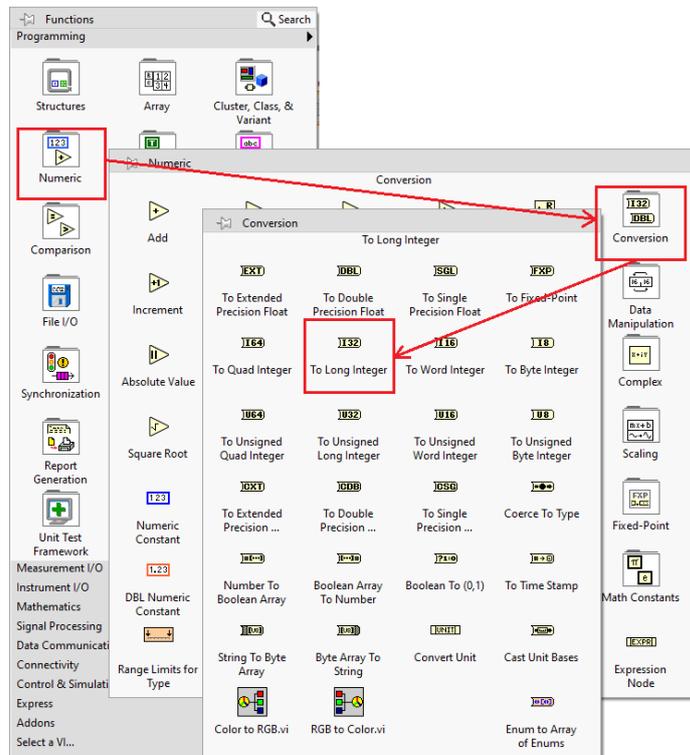


Рисунок 5.29 – Выбор элемента «*To Long Integer*»

Теперь подадим на вход этого элемента результат суммы первоначального сигнала и шума.

Для упрощения будем подавать на *OPC* сервер по одному значению, а не целый массив данных. Откроем окно «*Functions*», выберем «*Array*» и перенесем внутрь цикла элемент «*Index Array*» (рисунок 5.30). Соединим выход элемента «*To Long Integer*» с входом «*array*» элемента «*Index Array*». Выход «*element*» соединим с входом «*data*» элемента «*DataSocket Write*».

Для выбора номера элемента массива добавим еще один сдвиговый регистр нажатием правой кнопкой мыши по контуру цикла «*While Loop*» и последующим выбором «*Add Shift Register*», на вход которого подадим константу равную «0». Соединим сдвиговый регистр с входом «*index*» элемента «*Index Array*». Добавим операцию прибавления 1 для увеличения каждого следующего номера. Для этого откроем окно «*Functions*» и в «*Numeric*» выберем элемент «*Increment*». Расположим этот элемент на пути соединения сдвиговых регистров.

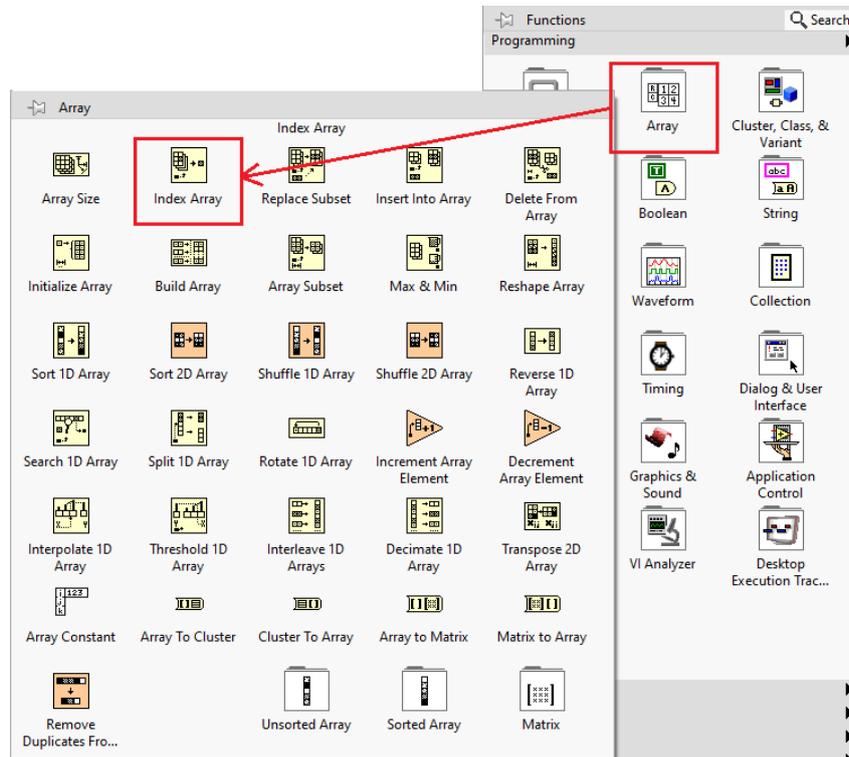


Рисунок 5.30 – Выбор элемента «*Index Array*»

Для удобства считывания данных замедлим скорость вычислений с помощью метронома. Для этого откроем окно «*Functions*», выберем «*Timing*» и перенесем внутрь цикла элемент «*Wait Until Next ms Multiple*» (рисунок 5.31). Наведем курсор мыши на левый вход элемента, щелчком правой кнопкой мыши вызовем контекстное меню, выберем «*Create*» → «*Constant*» и укажем значение 1000.

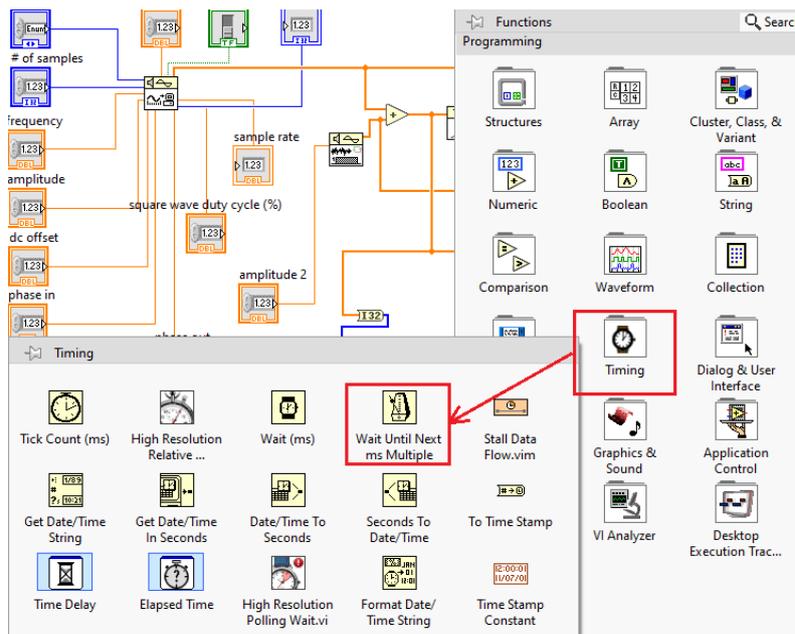


Рисунок 5.31 – Выбор элемента «метроном»

Конечный вид «Block Diagram» представлен на рисунке 5.32.

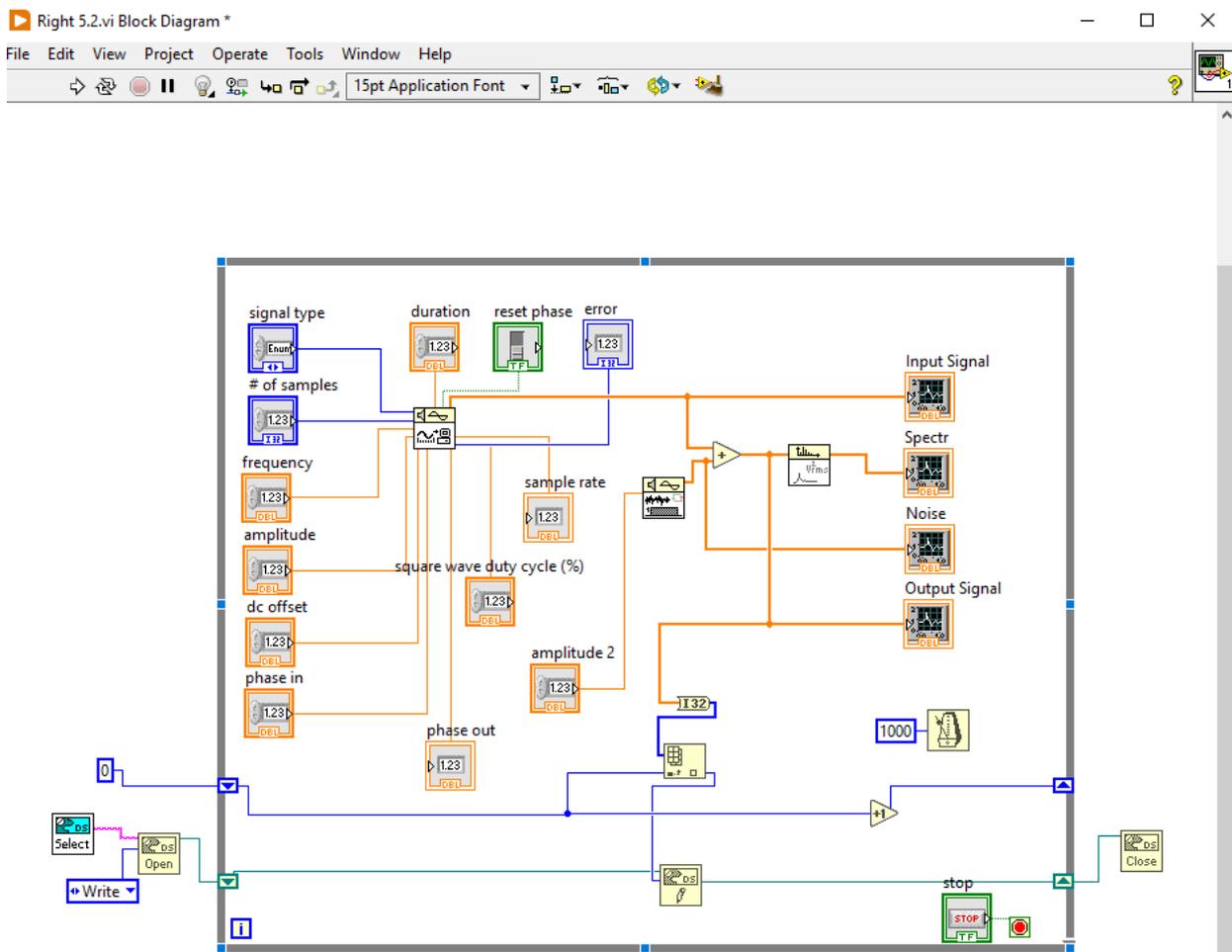


Рисунок 5.32 – Конечный вид окна «Block Diagram»

После запуска проекта программа попросит указать путь к *OPC* серверу (рисунок 5.33). Указываем необходимые сервер и тег, куда будут поступать значения. Значения на *OPC* сервер могут поступать с различной задержкой, зависящей от различных конфигураций.

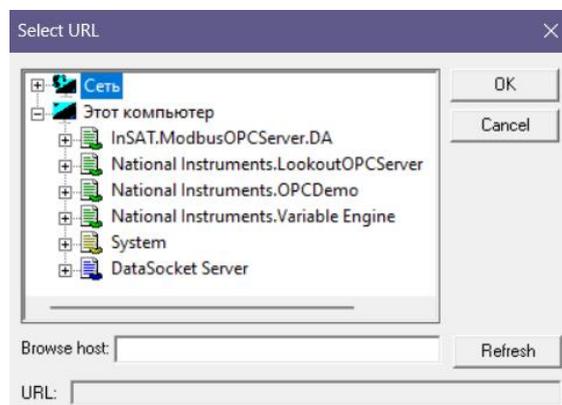


Рисунок 5.33 – Окно выбора пути к *OPC* серверу

5.3 Индивидуальные задания

В рамках лабораторной работы предусмотрено выполнение заданий в соответствии с индивидуальным номером зачетной книжки каждого студента. При этом важно учесть, что повторение заданий не допускается, обеспечивая таким образом самостоятельное выполнение своей работы. В случае совпадения номера зачетной книжки с уже занятым заданием студенту предоставляется возможность выбрать свободный номер из предложенных вариантов.

Варианты заданий представлены в таблице 5.1.

Таблица 5.1 – Варианты индивидуальных заданий

Вариант	Вид сигнала	Частота	Амплитуда	Длительность	Амплитуда шума
1	Косинус	7	8	2	1
2	Треугольник	11	2	1	3
3	Косинус	12	2	1	3
4	Квадрат	5	13	2	2
5	Треугольник	5	11	1	3
6	Квадрат	9	5	1	2
7	Треугольник	10	4	2	3
8	Синус	8	4	2	2
9	Треугольник	6	12	2	1
10	Пила	10	2	1	3
11	Косинус	9	3	1	2
12	Квадрат	10	8	1	3
13	Пила	7	7	1	2
14	Косинус	5	2	2	2
15	Синус	8	6	2	2
16	Синус	6	9	1	1
17	Пила	5	6	1	1
18	Треугольник	11	7	1	2
19	Квадрат	10	4	2	3
20	Пила	13	7	1	1
21	Треугольник	10	2	1	3
22	Треугольник	11	4	2	1
23	Треугольник	7	9	2	3
24	Квадрат	7	2	1	2
25	Треугольник	10	2	1	2
26	Пила	5	9	2	2
27	Синус	8	7	1	2
28	Косинус	12	3	2	3
29	Синус	12	13	1	3
30	Квадрат	6	3	2	2

5.4 Содержание отчета

В рамках лабораторной работы № 5 необходимо выполнить два индивидуальных задания. В связи с этим отчет по лабораторной работе должен содержать:

- 1) титульный лист;
- 2) введение:
 - 2.1) цель выполнения задания;
 - 2.2) краткое описание задания;
- 3) основная часть:
 - 3.1) ход выполнения задания;
 - 3.2) обработка и анализ результатов выполнения задания;
- 4) выводы по результатам выполнения задания.

5.5 Контрольные вопросы

- 1 Для каких целей в *LabVIEW* используются функции генерации шумов и сигналов?
- 2 Как изменится спектр сигнала, если к нему добавить шум?
- 3 Какие свойства тестируемого объекта характеризует передаточная функция?
- 4 С помощью контекстной справки определите обязательные, рекомендуемые и опциональные входы и выходы генератора сигнала с заданной длительностью.
- 5 Опишите спектр выходного сигнала.
- 6 Какой из инструментов можно использовать для интеграции генератора сигналов с *OPC* сервером?
- 7 Соотнесите виды сигнала с его типами, представленными в таблице 5.2.

Таблица 5.2 – Виды сигнала и его типы

Вид сигнала	Тип сигнала
Синусоидальный	<i>Triangle</i>
Прямоугольный	<i>Sawtooth</i>
Косинусоидальный	<i>Sine</i>
Треугольный	<i>Cosine</i>
Пилообразный	<i>Pulse</i>

- 8 С помощью какой функции можно экспортировать полученные данные в *Excel*?
- 9 Как изменится график сигнала, если к нему добавить шум?

Вспомогательная литература

- 1 Тревис, Дж. *LabVIEW для всех* / Дж. Тревис : пер. с англ. Н. А. Клушин – М. : ДМК Пресс ; ПриборКомплект, 2005. – 544 с.
- 2 Пилипенко, О. В. Основы программирования, математического моделирования и обработки данных в среде LABVIEW : практикум / О. В. Пилипенко, Н. Б. Горбачёв, М. А. Музалевская. – Орел : ОрелГТУ, 2008. – 70 с.
- 3 *LabVIEW: Практикум по основам измерительных технологий* / В. К. Батоврин [и др.]. – М. : ДМК Пресс, 2005. – 208 с.
- 4 Магда, Ю. С. *LabVIEW. Практический курс для инженеров и разработчиков* / Ю. С. Магда. – М. : ДМК Пресс, 2014. – 207 с.
- 5 Загидуллин Р. Ш. *LabVIEW в исследованиях и разработках* / Р. Ш. Загидуллин. – М. : Горячая линия – Телеком, 2005. – 352 с.

Лабораторная работа № 6
Создание проекта в *MasterSCADA* с использованием
OPC сервера и типизации

Цель работы: используя *OPC* сервер, создать в программной среде *MasterSCADA* проект для расчета активной, реактивной и полной мощностей оборудования систем электроснабжения.

Основные задачи:

1 Ознакомиться с графическим интерфейсом программного комплекса *MasterSCADA*.

2 Изучить правила и инструменты создания проектов моделирования в *MasterSCADA*.

3 Ознакомиться с возможностью интегрирования проектов *LabVIEW* в *MasterSCADA*.

4 Получить практические навыки работы с инструментами *MasterSCADA*.

5 Закрепить полученные навыки и умения путем выполнения задания.

6.1 Теоретические сведения

6.1.1 Общая информация

В связи с тенденцией к интеграции систем управления технологическими процессами и систем управления предприятием все чаще возникает необходимость использования *SCADA* в качестве источника данных для вышестоящих систем. Некоторые *SCADA* могут выступать и как сервер консолидации всех технологических данных, и как сервер генерации отчетов на базе этих данных.

Некоторые независимые от производителей оборудования *SCADA*-системы предлагают набор драйверов к большинству имеющихся на рынке контроллерам и интеллектуальными приборам. Наиболее универсальный способ взаимодействия – это использование драйверов, разработанных в соответствии со стандартом *OPC*. Такие *OPC* серверы могут быть разработаны производителями контроллеров или независимыми разработчиками, а использоваться вместе с любой *SCADA*-системой.

ВАЖНО ЗНАТЬ!

SCADA-система (англ. «*Supervisory Control And Data Acquisition*») – это инструментальная программа, обеспечивающая создание программного обеспечения для автоматизации контроля и управления технологическим процессом в режиме реального времени.

Мнемосхема – графический интерфейс, который обычно используется в *SCADA*-системах для визуализации процессов, устройств, систем и данных, собранных с различных 7774, в реальном времени.

Основная цель создаваемой с помощью *SCADA* программы – предоставить оператору, управляющему технологическим процессом, полную информацию об этом процессе и необходимые средства для воздействия на него.

К основным задачам данных систем относятся:

- сбор данных от датчиков и представление их оператору в удобном для него виде, включая графики изменения параметров во времени;
- дистанционное управление исполнительными механизмами;
- ввод заданий алгоритмам автоматического управления;
- реализация алгоритмов автоматического контроля и управления (чаще эти задачи возлагаются на контроллеры, но *SCADA*-системы тоже способны их решать);
- распознавание аварийных ситуаций и информирование оператора о состоянии процесса;
- формирование отчетности о ходе процесса и выработке продукции.

От надежности, быстродействия и эргономичности *SCADA*-системы зависит не только эффективность управления технологическим процессом, но и его безопасность.

6.1.2 Расчет мощностей в электроэнергетике: основные понятия и формулы

В электроэнергетике расчет мощностей является важной задачей при оценке эффективности работы систем электроснабжения. Мощность определяется как количество энергии, передаваемой или потребляемой в единицу времени. Важными характеристиками являются полная, активная и реактивная мощности.

Активная мощность (P) представляет собой фактическую потребляемую мощность, измеряемую в ваттах (Вт). Это «полезная» мощность, которая преобразуется в работу. В цепи постоянного тока мощность, питающая нагрузку постоянного тока, определяется как произведение напряжения (U) на нагрузке и протекающего тока (I), то есть

$$P = U \times I, \quad (6.1)$$

потому что в цепи постоянного тока нет понятия фазового угла между током и напряжением. Другими словами, в цепи постоянного тока нет никакого коэффициента мощности. Но при синусоидальных сигналах, то есть в цепях переменного тока, ситуация сложнее из-за наличия разности фаз между током и напряжением. Поэтому среднее значение мощности (активная мощность), которая в действительности питает нагрузку, определяется как

$$P = U \times I \times \cos \theta. \quad (6.2)$$

Реактивная мощность (Q) представляет собой мощность, которая не выполняет фактической работы, но требуется для обеспечения нормального функционирования индуктивных устройств. Единицей измерения является вольт-ампер реактивный (ВАр). Реактивная мощность определяется как

$$Q = U \cdot I \cdot \sin \theta, \quad (6.3)$$

и может быть положительной ($+Ue$) для индуктивной нагрузки и отрицательной ($-Ue$) для емкостной нагрузки.

Полная мощность (S) обозначает общую мощность, переносимую электрическим устройством. Ее измеряют в вольт-амперах (ВА). Рассчитывается как

$$S = \sqrt{P^2 + Q^2}, \quad (6.4)$$

где P – активная мощность (Вт);

Q – реактивная мощность (ВАр).

Практическое применение данного проекта включает в себя:

- мониторинг и оптимизация энергопотребления. Позволяет отслеживать мощности в реальном времени, выявлять периоды пикового потребления и оптимизировать энергопотребление для снижения эксплуатационных затрат;
- диагностика и предупреждение отказов. Анализ мощностей позволяет выявлять аномалии, предупреждая о возможных отказах оборудования и предотвращая от потенциальных аварий;
- управление качеством энергии. Позволяет контролировать факторы, влияющие на качество электроэнергии, такие как мощность, напряжение и ток, для обеспечения стабильной работы оборудования.

6.1.2 Типизация в *MasterSCADA*: как структурировать проекты для автоматизации промышленных процессов

В рамках автоматизации промышленных процессов *MasterSCADA* предоставляет возможности по структурированию проектов с использованием типизации. Типизация в *MasterSCADA* включает в себя определение типов данных для переменных и объектов, что способствует более организованному и читаемому коду.

Основные принципы типизации в *MasterSCADA* включают в себя использование различных типов данных, создание объектов и их экземпляров, а также работу со структурами и массивами для организации данных.

Преимущества использования типизации в *MasterSCADA* заключаются в возможности более четкого структурирования проекта, повторном использовании кода, что экономит время и снижает риск ошибок, а также обеспечивает безопасность работы системы.

При создании проекта в *MasterSCADA* с использованием типизации рекомендуется определить структуру проекта, создать типы объектов, применить соответствующие типы данных к переменным, структурировать данные с использованием структур и массивов, а также использовать типы данных для создания единообразных шаблонов устройств.

Такой подход улучшает сопровождаемость проекта, делает его более понятным и масштабируемым. Он также обеспечивает быстрое масштабирование проекта при добавлении новых компонентов, что делает *MasterSCADA* более эффективным инструментом для создания систем мониторинга и управления в промышленных процессах.

В контексте данного проекта используются несколько устройств, которые генерируют различные сигналы. Среди этих устройств симулируем неисправность в одном из них при помощи интеграции проекта из *LabVIEW*.

6.2 Задания для выполнения лабораторной работы

Задание 1. Создать проект с помощью *OPC* сервера в *MasterSCADA*, в котором будет произведен расчет мощностей.

Порядок выполнения задания 1

Настройка сервера в *MasterSCADA* состоит из следующих пунктов.

- 1 Запуск сервера.
- 2 Создание конфигурации.
- 3 Настройка тегов.

На рисунке 6.1 показана рабочая область загруженного сервера *MasterOPC Universal Modbus Server*.

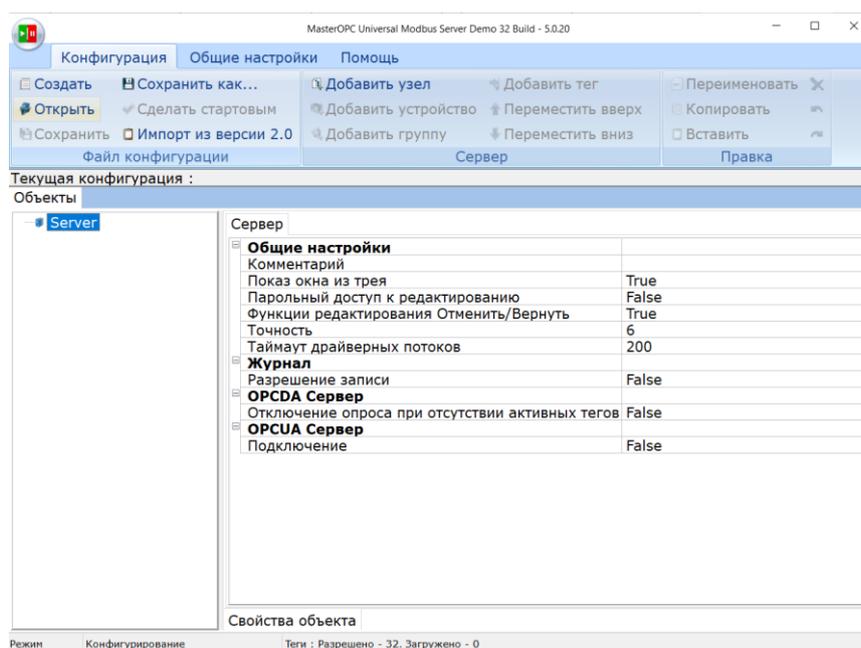


Рисунок 6.1 – Рабочая область сервера

Для создания конфигурации необходимо нажать кнопку «Создать», как показано на рисунке 6.2.

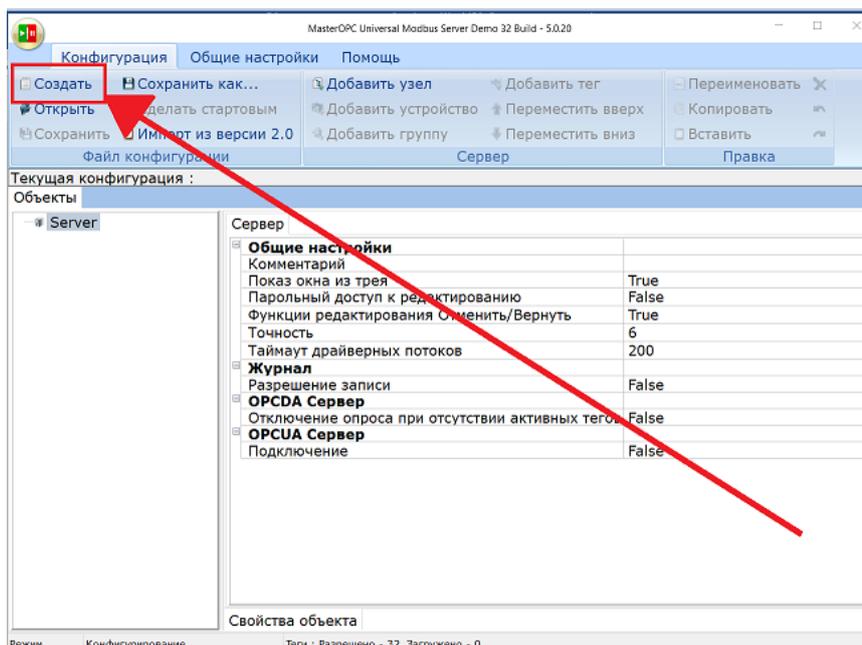


Рисунок 6.2 – Создание конфигурации сервера

Далее нужно добавить коммуникационный узел. Сделать это можно двумя способами: либо кликнуть правой кнопкой мыши по объекту «Server», либо кликнуть «Добавить узел» (рисунки 6.3 и 6.4), после в редактировании коммуникационного узла нужно изменить имя узла и поставить в типе узла значение *TCP/IP*.

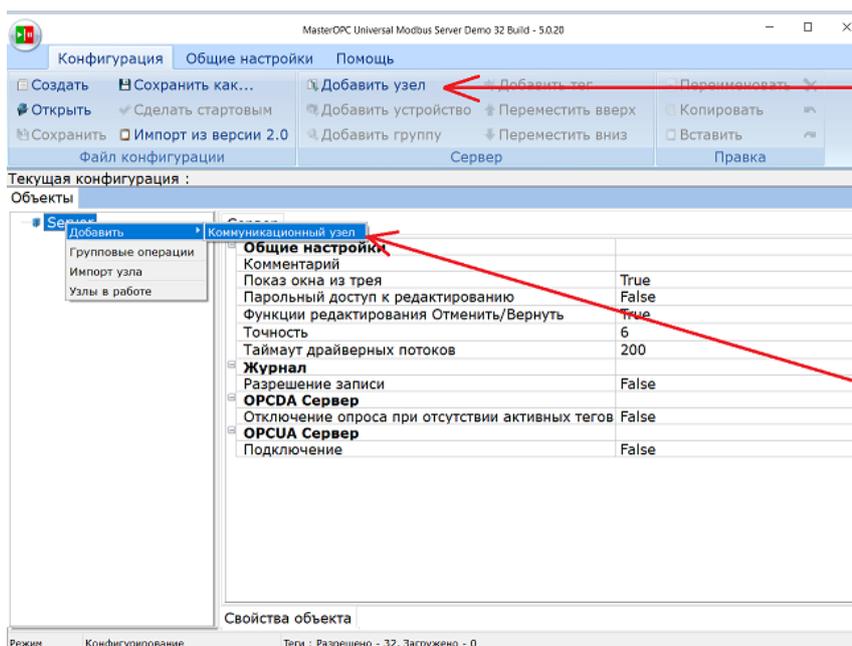


Рисунок 6.3 – Добавление коммуникационного узла

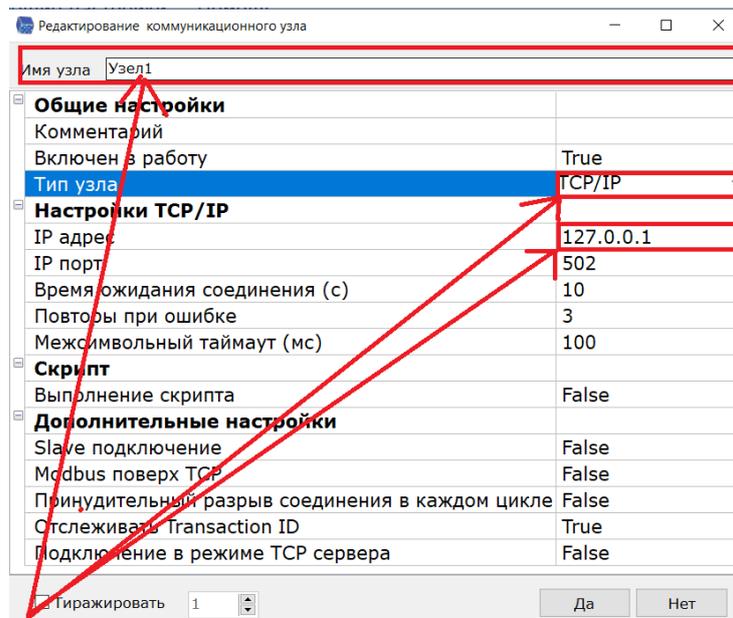


Рисунок 6.4 – Редактирование коммуникационного узла

Далее добавляем и меняем имя устройства (рисунки 6.5 и 6.6).

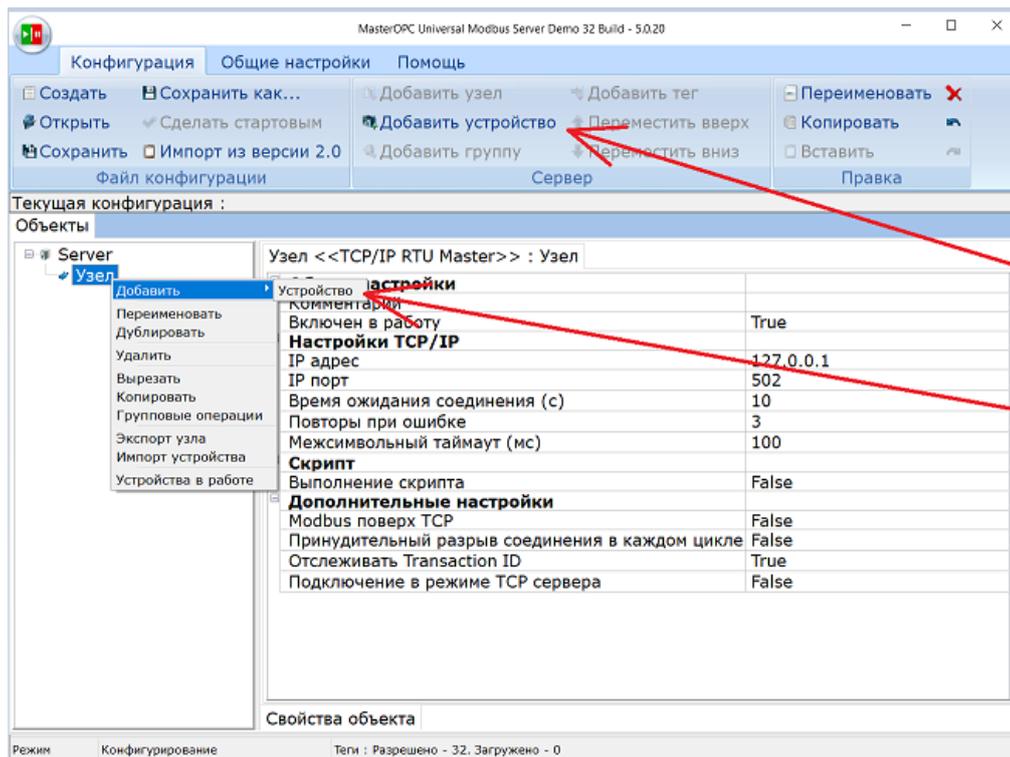


Рисунок 6.5 – Добавление устройства на сервере *MasterOPC Universal Modbus Server*

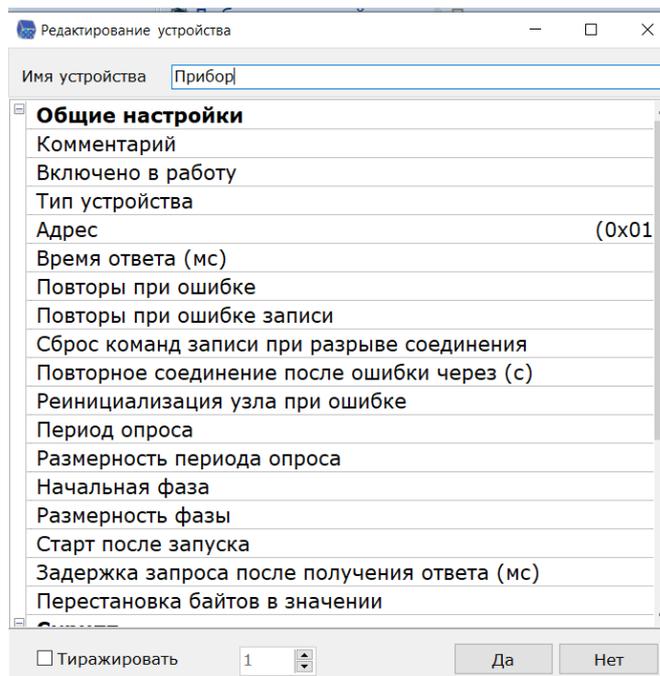


Рисунок 6.6 – Редактирование имени устройства

Также нужно добавить три тега, в которых потом будем изменять параметры: «Ток», «Напряжение» и «Угол», как показано на рисунках 6.7 и 6.8.

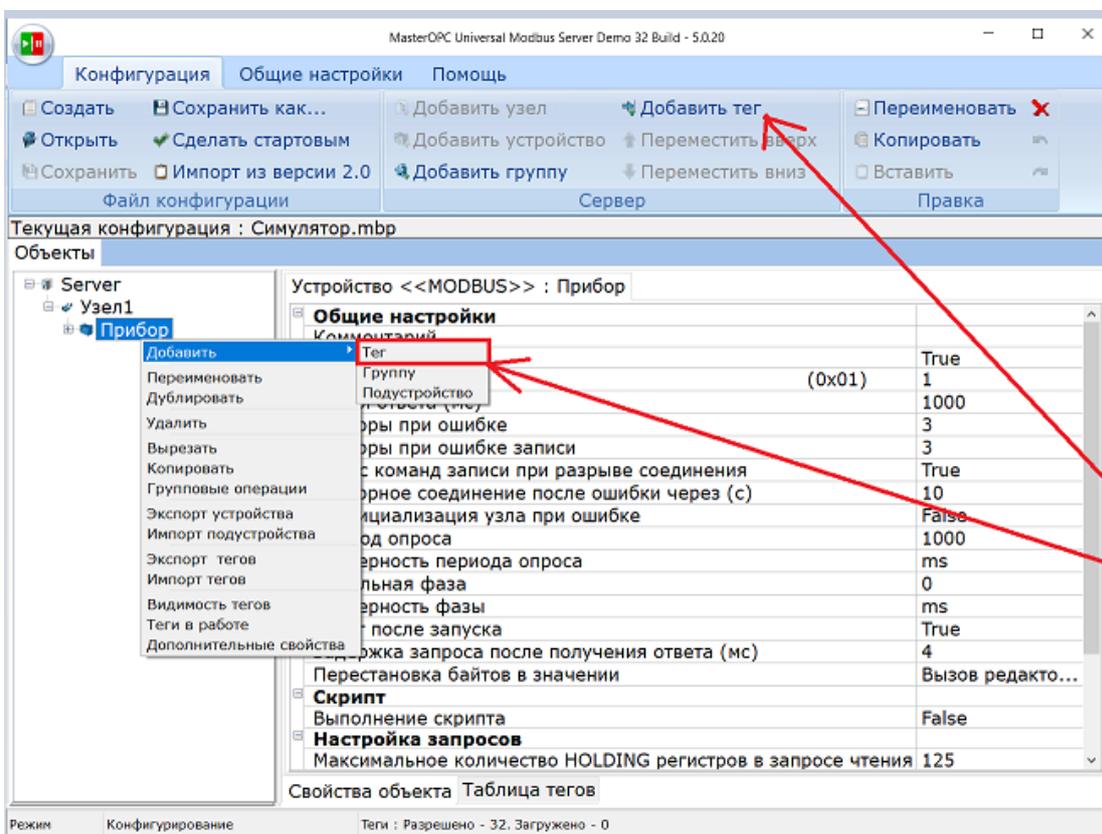


Рисунок 6.7 – Добавление тега

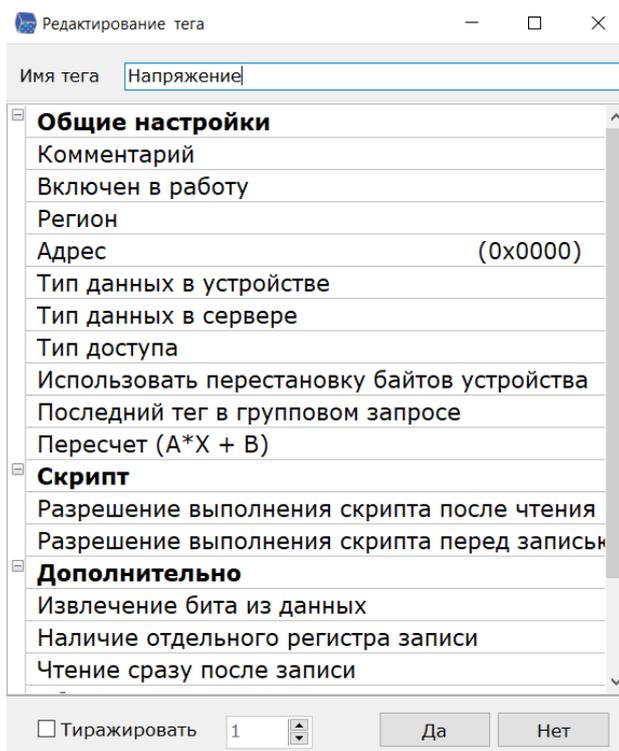


Рисунок 6.8 – Редактирование имени тега

После добавления тегов нужно изменить несколько полей. В тегах «Напряжение» и «Ток» нужно установить в поле «Пересчет ($A \times X + B$)» значение «True». Должны были появиться поля «Коэффициент A » и «Коэффициент B », нужно установить в них значения 0,5 и 200 соответственно для напряжения и 0,1 и 10 соответственно для тока (рисунок 6.9).

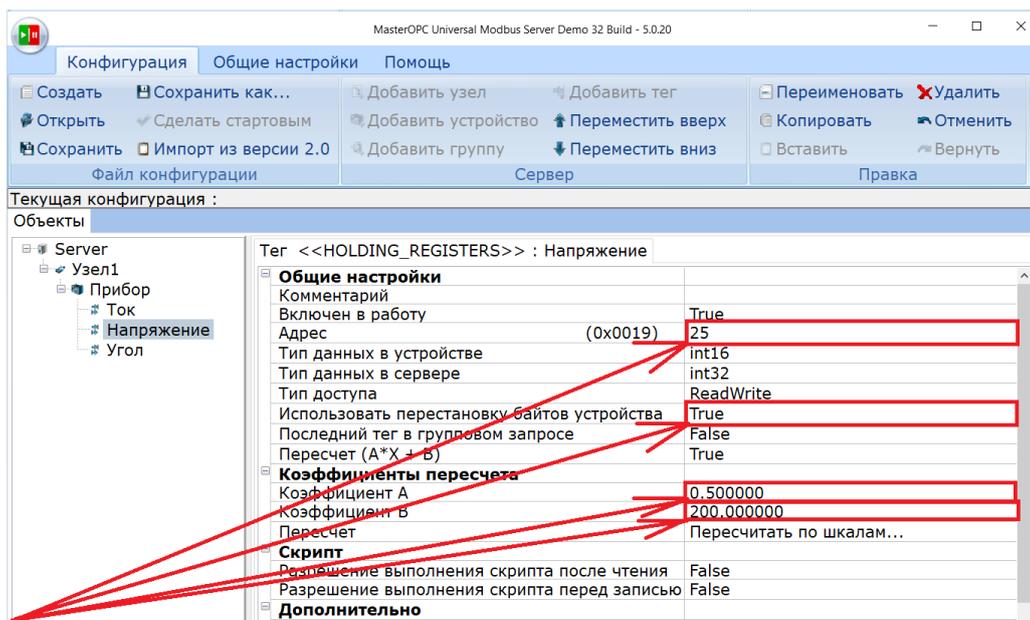


Рисунок 6.9 – Изменение полей в тегах

В конце нужно сохранить нашу конфигурацию под именем «Симулятор» и нажать «Сделать стартовым» (рисунок 6.10).

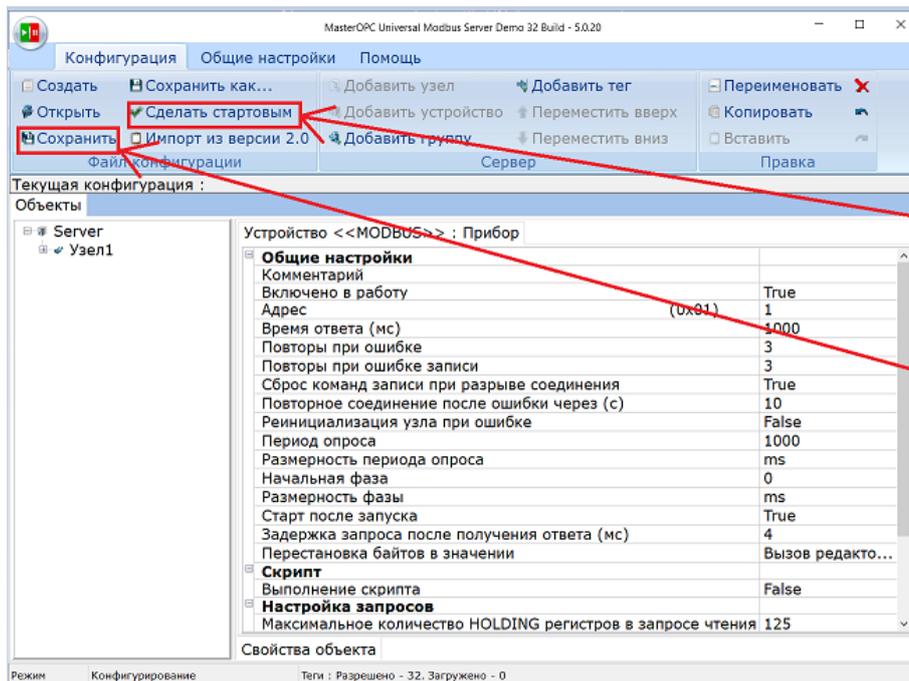


Рисунок 6.10 – Сохранение конфигурации

Настройка сервера для данной лабораторной работы завершена.

Далее запускаем *MasterSCADA*, в открывшемся диалоговом окне, как показано на рисунке 6.11, необходимо указать, с каким проектом вы будете работать: новым, предыдущим, существующим или учебным.

Создаем новый проект.

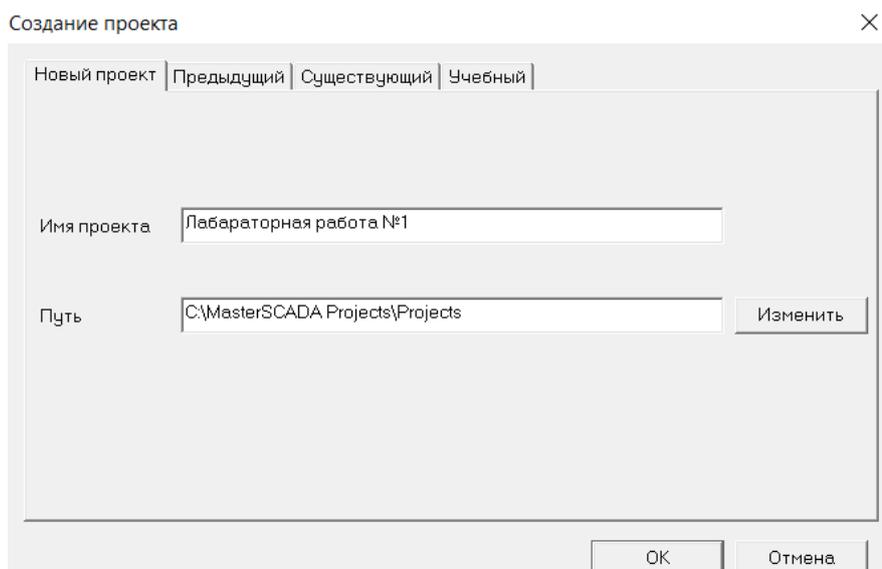


Рисунок 6.11 – Стартовое диалоговое окно

После создания нового проекта появляется диалоговое окно для добавления пароля к проекту. Создаем проект без пароля.

Созданный проект будет отображать на мнемосхеме действующее напряжение, действующий ток, активную, реактивную и полную мощность. Данные о напряжении, тока и угла φ поступают от прибора через *OPC* сервер. Мощность необходимо рассчитать в проекте.

На рисунке 6.12 показана рабочая область *MasterSCADA*.

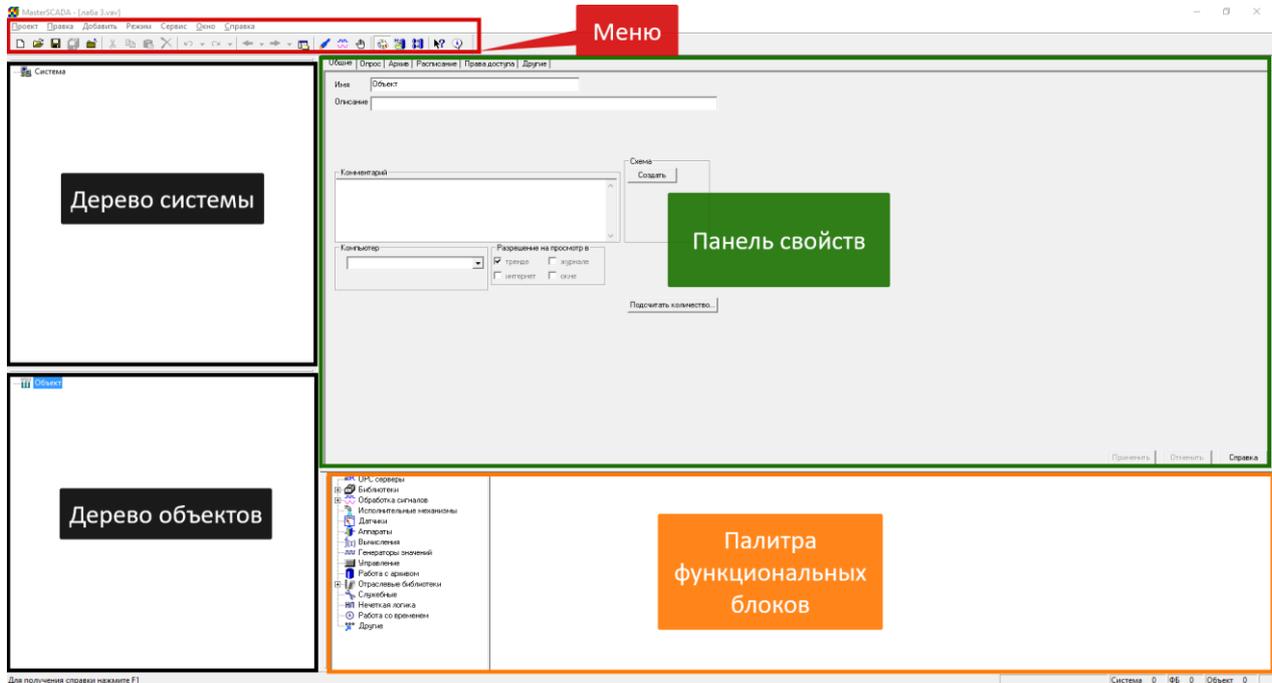


Рисунок 6.12 – Рабочая область в *MasterSCADA*

Добавим в дерево системы компьютер (рисунок 6.13). Затем необходимо провести поиск доступных *OPC* серверов на компьютере (рисунок 6.14).

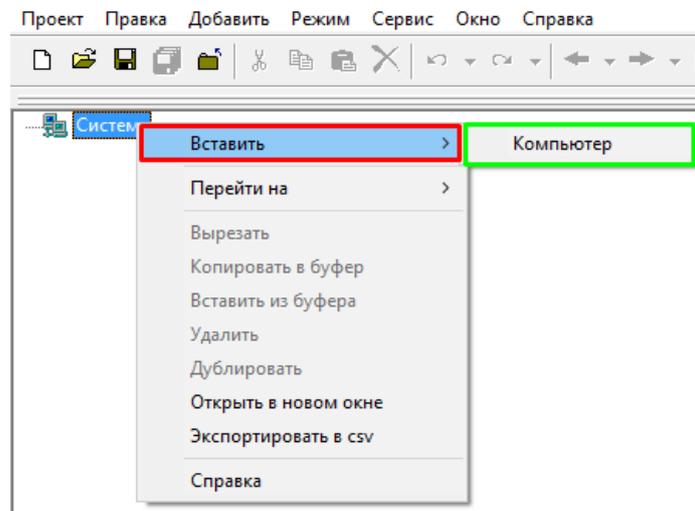


Рисунок 6.13 – Добавление в дерево системы компьютера

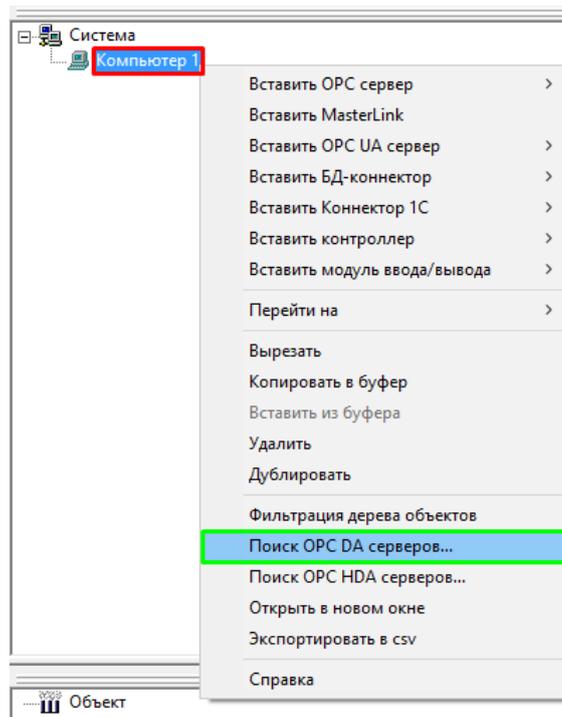


Рисунок 6.14 – Поиск *OPC* серверов на компьютере

В открывшемся окне выбираем *InSAT Modbus OPC Server DA*, как показано на рисунке 6.15. Затем добавляем его через контекстное меню на «Компьютер 1» (рисунок 6.16).

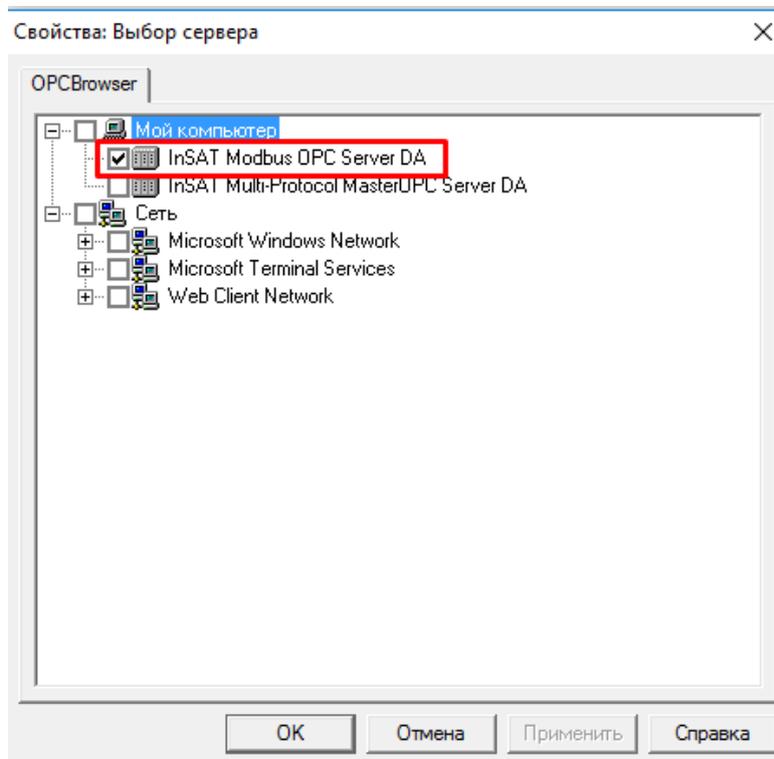


Рисунок 6.15 – Добавление *OPC* сервера

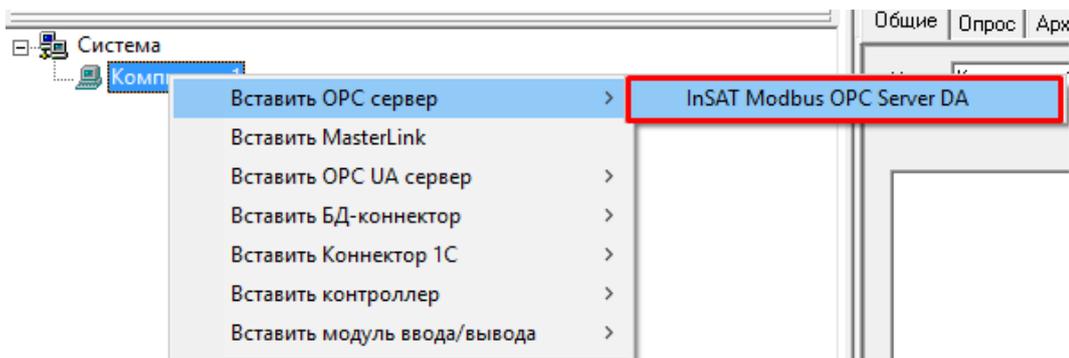


Рисунок 6.16 – Вставка *OPC* сервера

Далее добавляем в дерево системы переменные через контекстное меню на *OPC* сервере (рисунок 6.17). Когда все переменные будут добавлены, дерево системы можно считать настроенным.

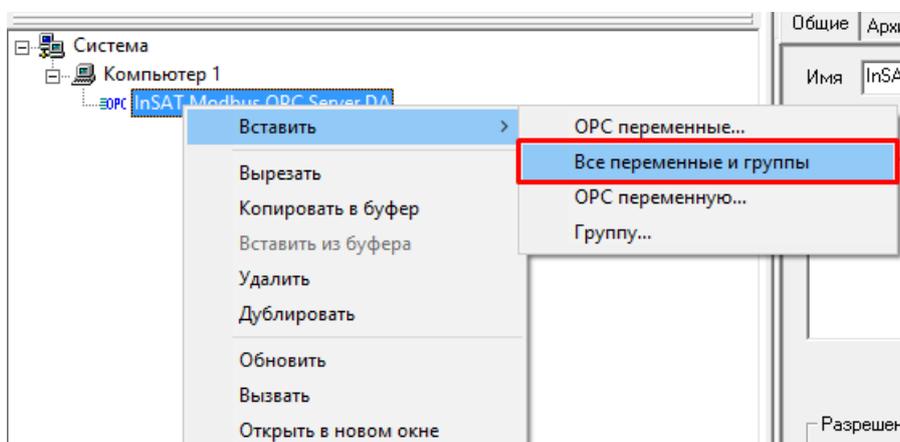


Рисунок 6.17 – Добавление переменных

Далее идет настройка дерева объекта. Объект – это элемент программы, предназначенный для размещения в нем других элементов, переменных, функциональных окон, а также других объектов. Объекты добавляются через контекстное меню «Объект» (рисунок 6.18).

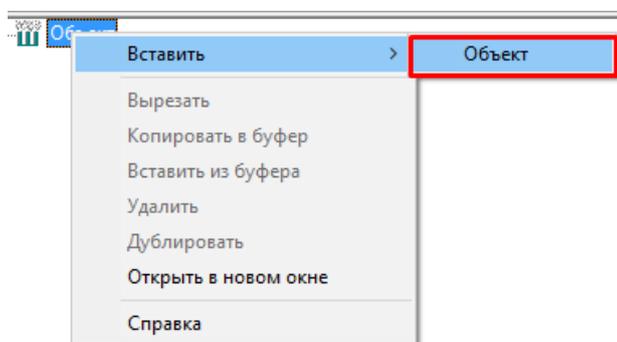


Рисунок 6.18 – Добавление объекта в дерево объекта

Меняем имя «Объект 1» на имя «Установка» во вкладке «Общие» свойств «Объекта» (рисунок 6.19).

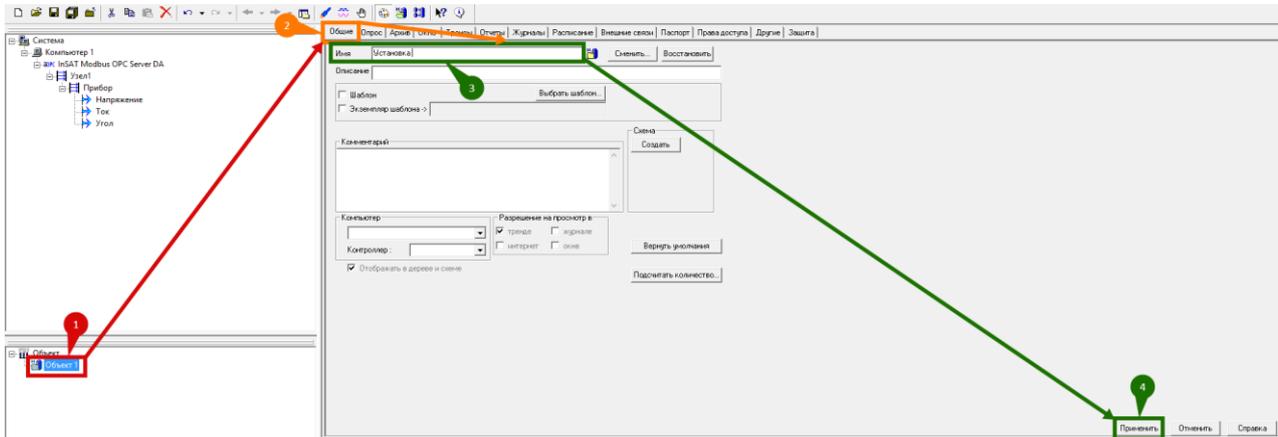


Рисунок 6.19 – Настройка имени «Объект 1»

Добавим в «Установку» объект из контекстного меню и переименуем в «Реактивная». В этом объекте будет высчитываться реактивная мощность по формуле (6.3)

Добавим в объект «Реактивная» синус угла, используя палитру функционального блока (рисунок 6.20), перетаскиваем \sin к объекту «Реактивная». В случае, если отсутствует этот пункт, необходимо выбрать в меню «Сервис» → «Организатор» → «ФБ...» и в открывшемся окне найти «Синус» в списке слева и добавить его в раздел «Вычисления» в списке справа с помощью кнопки «> > >», после чего нажимаем «Принять настройки и обновить палитру».

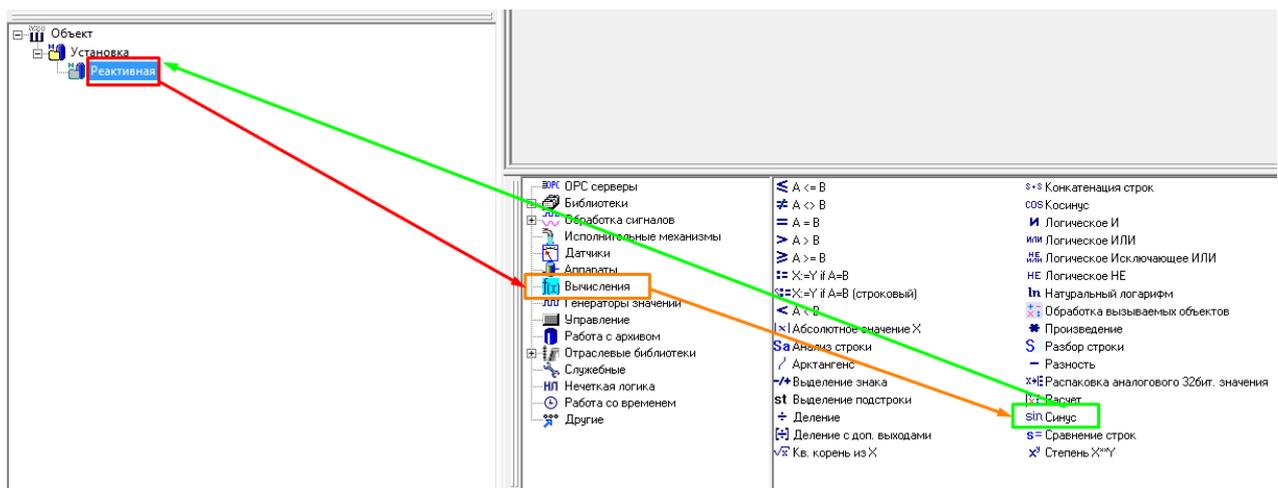


Рисунок 6.20 – Добавление синуса угла

Зададим значение «Период» и «Диапазон» во вкладке «Опрос» (рисунок 6.21) 360 и 2 соответственно.

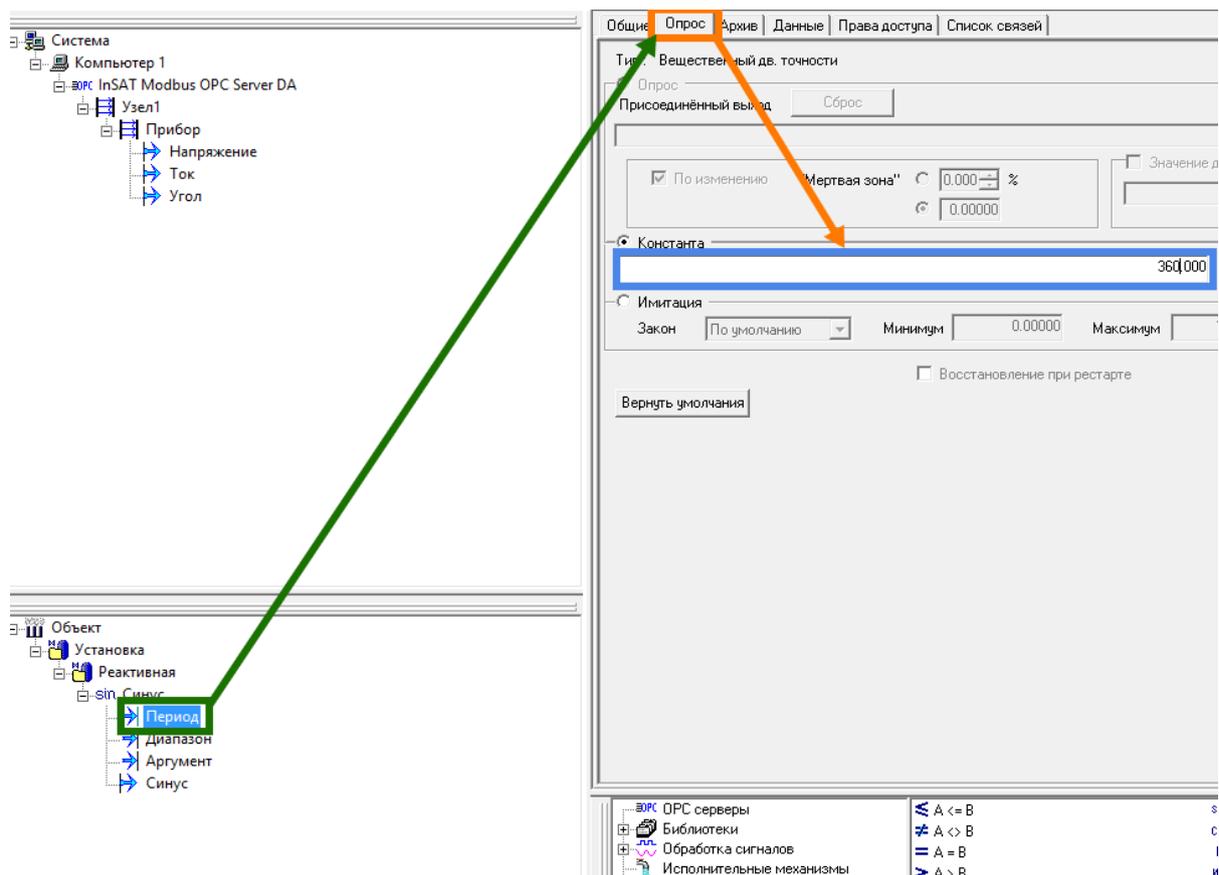


Рисунок 6.21 – Редактирование значения синуса угла

Значение «Аргумент» получаем из *OPC* сервера. Поэтому необходимо установить связь между входом «Аргумента» и *OPC* переменной из дерева системы. Для этого перетащим значение «Угол» из дерева системы в значение «Аргумент». У *OPC* переменной и входа появилась красная метка, сигнализирующая, что связь установлена (рисунок 6.22).

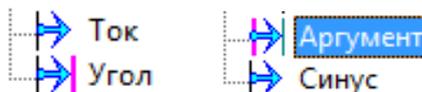


Рисунок 6.22 – Установление связи между значениями «Угол» и «Аргумент»

Для того чтобы посчитать произведение трех компонентов, для этого добавим из функционального блока «Произведение» к «Синусу» в дереве объекта. Если данный пункт отсутствует, то повторите действия, описанные выше, с синусом. В появившейся переменной имеется значение произведение и группа множителей. Добавим третий множитель. Для этого во вкладке настройки изменим число множителей на «3» (рисунок 6.23).

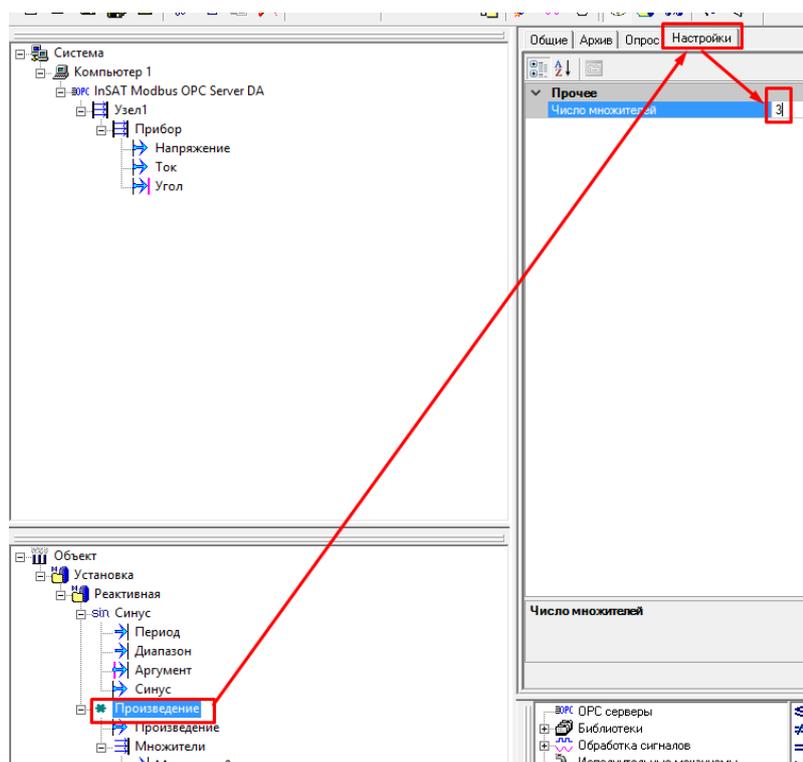


Рисунок 6.23 – Настройка множителей

Установим связь между «Множитель 0» и «Напряжение», «Множитель 1» и «Ток», «Множитель 2» и «Синус» с помощью перетаскивания.

Создадим объект в «Установка» и переименуем его в «Активная». Активная мощность вычисляться по формуле (6.2).

Добавим в объект «Активная» косинус угла φ . Аналогично синусу угла объекта «Реактивная» настроим косинус угла φ (360 – «Период», 2 – «Диапазон»). Далее скопируем реактивное произведение в буфер и вставим в объект «Активная» (рисунок 6.24). В окне «Восстановление внешних связей» восстановим связи произведения при помощи нажатия кнопки «Восстановить».

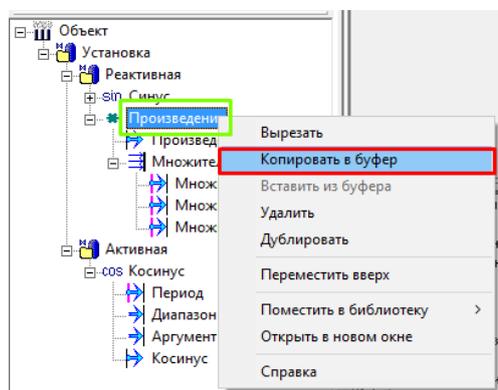


Рисунок 6.24 – Копирование произведения в буфер

Установим связь между «Множитель 2» и «Косинус» (рисунок 6.25).

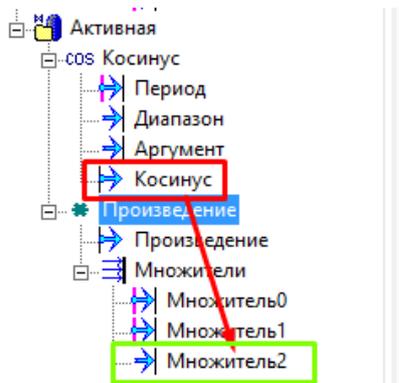


Рисунок 6.25 – Установка связей между «Множитель 2» и «Косинус»

Добавим в объект «Установка» дополнительный объект «Полная». Полная мощность вычисляется по формуле (6.4).

Вспользуемся функцией модуля «Расчет» (рисунок 6.26). Переименуем расчет в «Полная». Перейдем во вкладку «Формула». В нее перетащим из дерева системы «Ток» и «Напряжение» (рисунок 6.27).

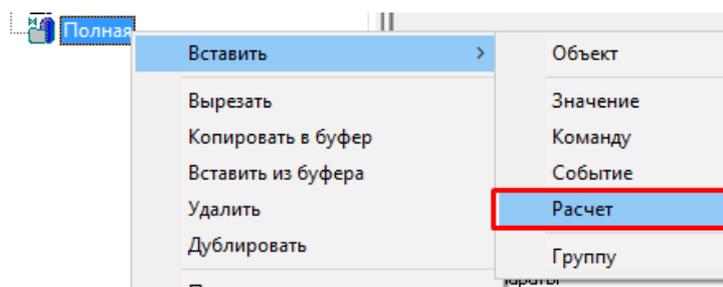


Рисунок 6.26 – Добавление модуля расчета

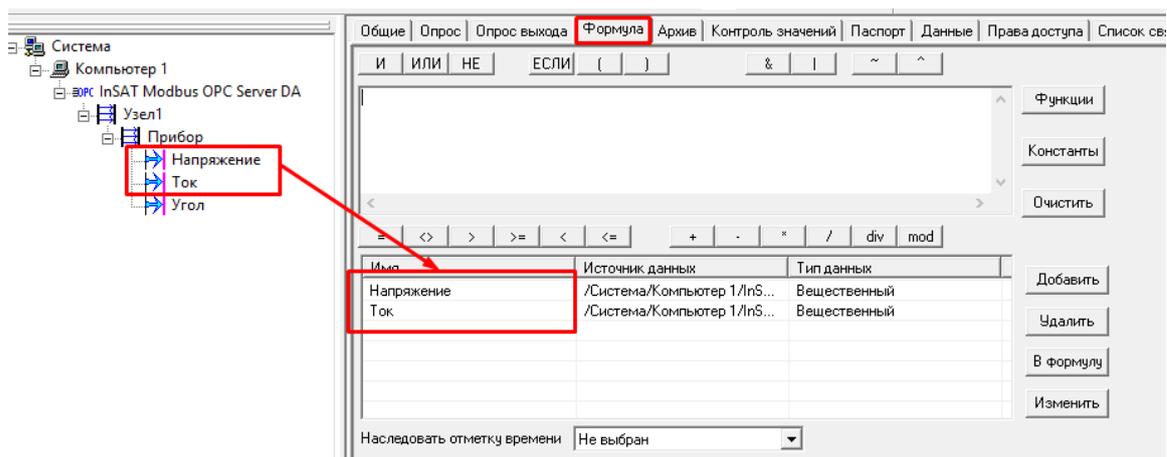


Рисунок 6.27 – Добавление переменных

Двойным щелчком левой кнопкой мыши добавим «Напряжение», знак умножения и «Ток» (рисунок 6.28), нажмем кнопку «Применить».

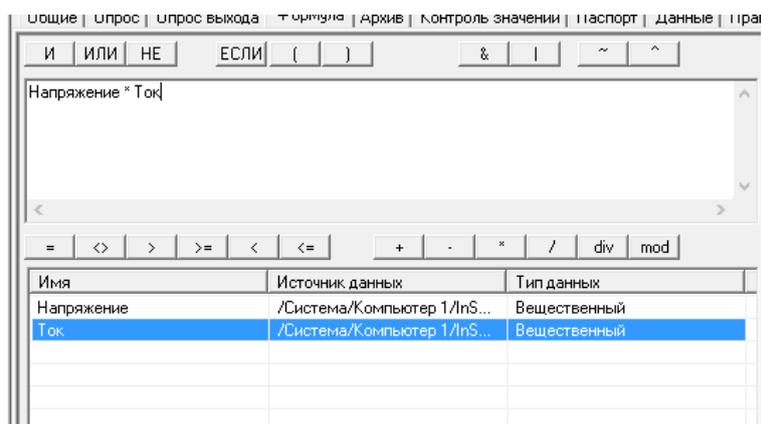


Рисунок 6.28 – Настройка модуля «Расчет»

Добавим в дерево объектов параметры сети (объект «Параметры»). Перетащим группу прибор из OPC сервера в данный объект. Теперь переменные добавлены в дерево объектов. Дерево объекта настроено.

Создадим мнемосхему для отображения графиков. Мнемосхема создается у объектов на закладке «Окна» (рисунок 6.29).

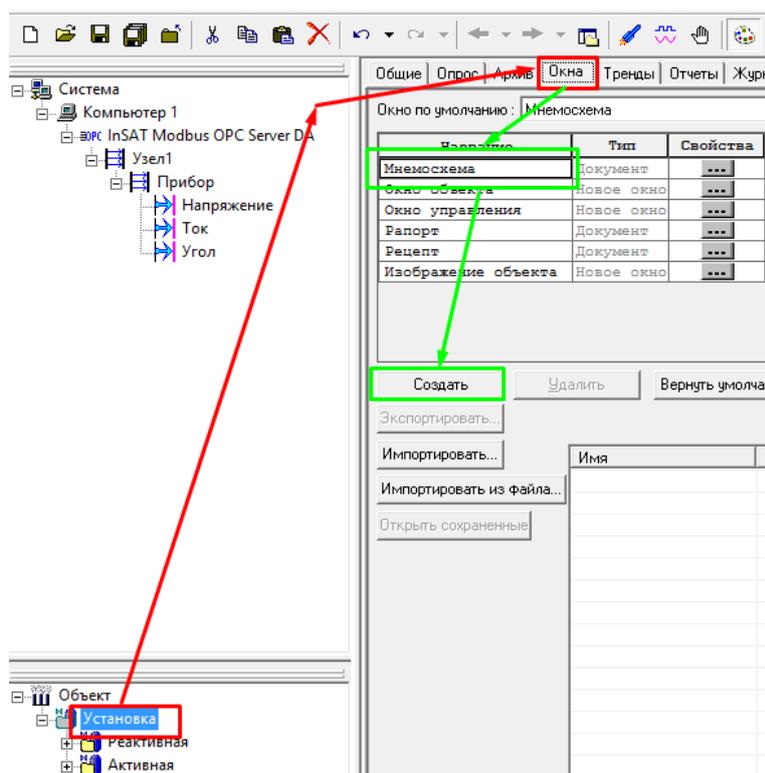


Рисунок 6.29 – Создание мнемосхемы

Перетащим из дерева объектов правой кнопкой мыши значения переменных (объект «Параметры»). В появившемся окне выбираем «Стрелочный прибор». Чтобы не подгонять каждый элемент прибора, можно скопировать элемент и установить связь с другим значением переменной. Для этого перетащим нужное значение в окно «Свойства элемента» (рисунок 6.30).

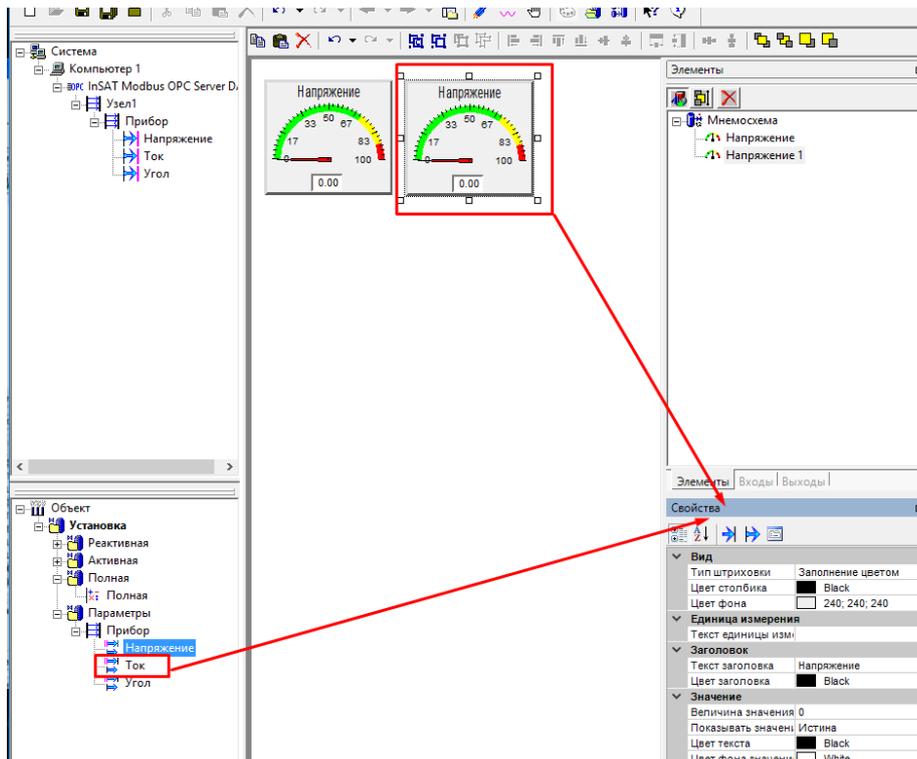


Рисунок 6.30 – Добавление связи в мнемосхеме

Создадим следующие «стрелочные приборы» (рисунок 6.31):

- напряжение;
- ток;
- реактивная;
- активная;
- полная.

Для этого перетащим правой кнопкой мыши переменную «Произведение» объектов «Активная» и «Реактивная» и переименуем их соответственно. Для объекта «Полная» перетащим правой кнопкой мыши в модуль «Расчет» (см. рисунок 6.28).

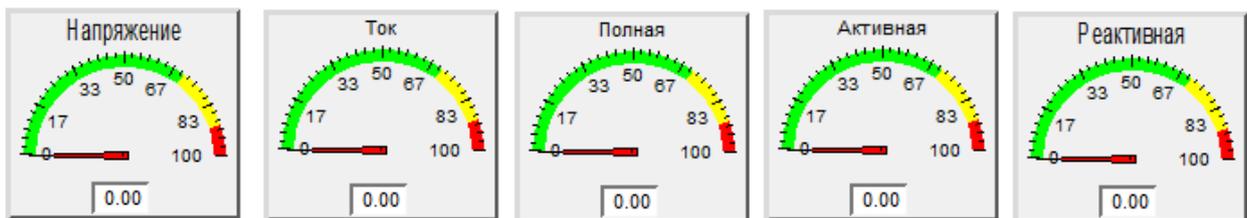


Рисунок 6.31 – Добавление значений на мнемосхему

Все приборы имеют значение от 0 до 100, что не соответствует нашим значениям, кроме того, отсутствуют единицы значения. Для этого выйдем из мнемосхемы (рисунок 6.32) и создадим шкалу на вкладке «Шкала» в свойствах «Системы» (рисунок 6.33).

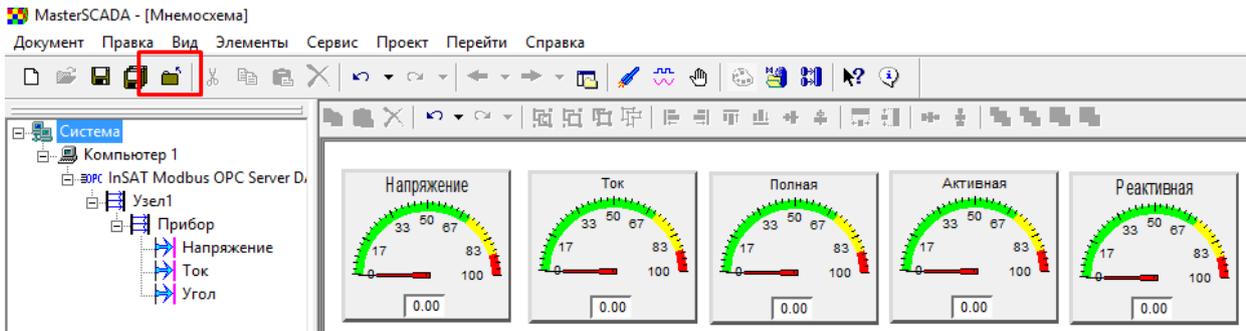


Рисунок 6.32 – Выход из мнемосхемы

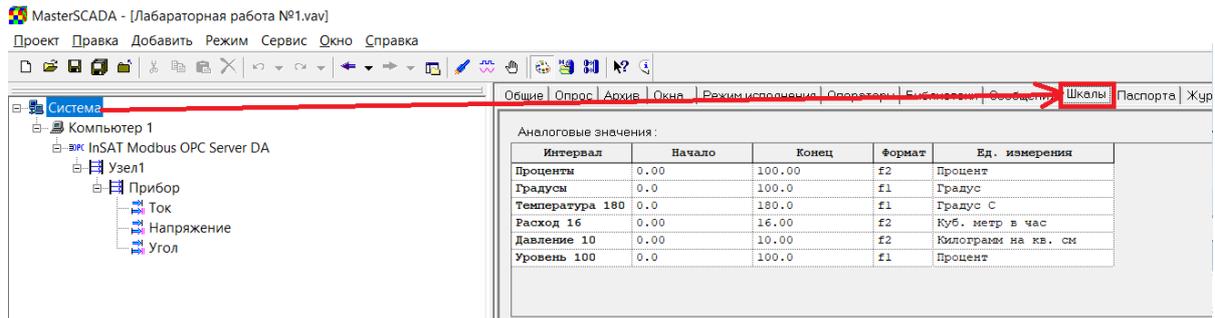


Рисунок 6.33 – Создание шкалы в разделе «Шкалы»

Создадим шкалу для напряжения (рисунок 6.34). Аварийные и предупредительные нижние границы уберем, так как отсутствие напряжения не является аварией. Аналогично создадим шкалу для тока (рисунок 6.35).

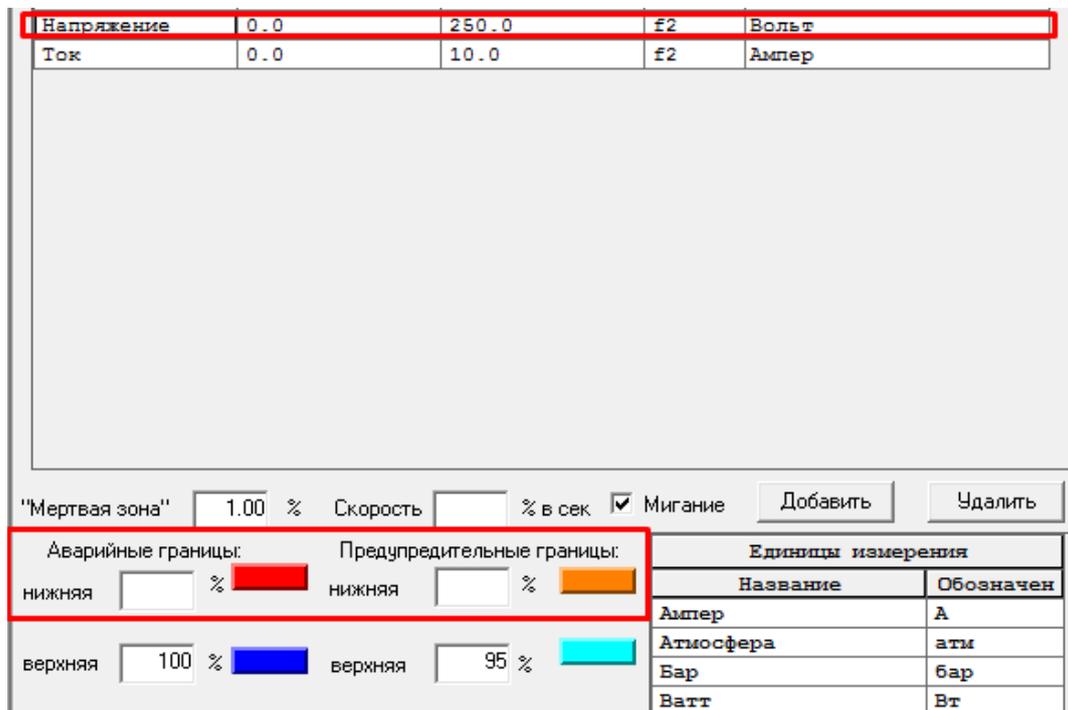


Рисунок 6.34 – Шкала для напряжения

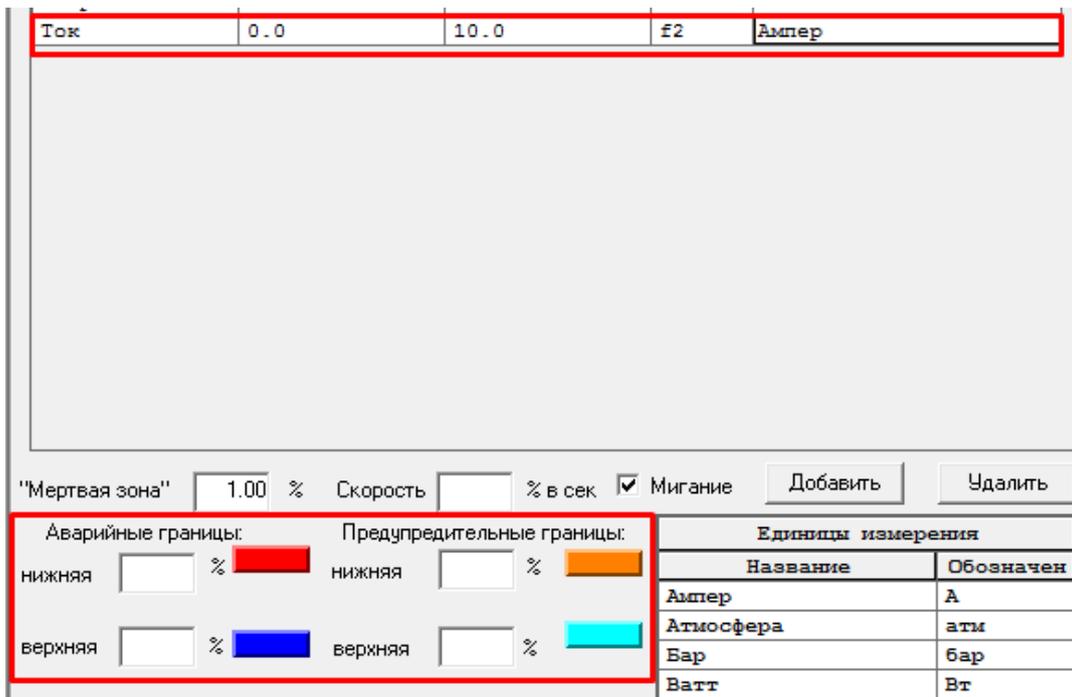


Рисунок 6.35 – Шкала для тока

Создадим шкалу для реактивной мощности. Верхнее значение зададим равным 2500. Так как реактивная мощность измеряется в варах, а данная единица отсутствует, поэтому необходимо создать ее, как показано на рисунке 6.36. Для этого нажимаем на «Единицы измерения», чтобы таблица загорелась красным, а затем – «Добавить».

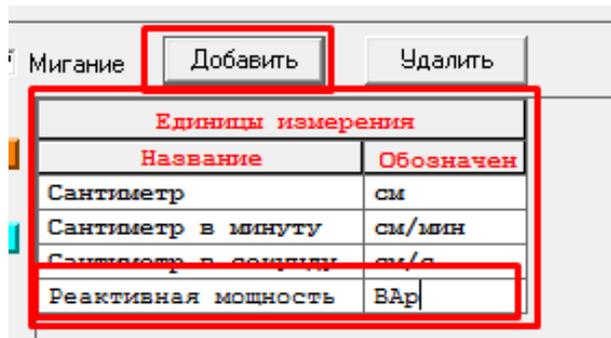


Рисунок 6.36 – Создание переменных

Аналогично создадим единицу измерения для «Полной мощности» и нажимаем кнопку «Применить».

На рисунке 6.37 показаны все созданные для проекта шкалы с обозначенными единицами измерения.

Напряжение	0.00	250.00	f2	Вольт
Ток	0.00	10.00	f2	Ампер
Реактивная	0.00	2500.00	f2	Реактивная мощность
Активная	0.0	2500.0	f2	Ватт
Полная	0.0	2500.0	f2	Полная мощность

"Мертвая зона" % Скорость % в сек Мигание

Аварийные границы: Предупредительные границы:

нижняя % нижняя %

верхняя % верхняя %

Единицы измерения	
Название	Обозначен
Сантиметр в минуту	см/мин
Сантиметр в секунду	см/с
Реактивная мощность	ВАр
Полная мощность	ВА

Рисунок 6.37 – Шкалы для проекта

Далее необходимо добавить для каждого параметра свою шкалу. Это делается на закладке «Общие» в параметре «Диапазон изменения» (рисунок 6.38). Для каждой созданной шкалы применим подходящий для нее параметр.

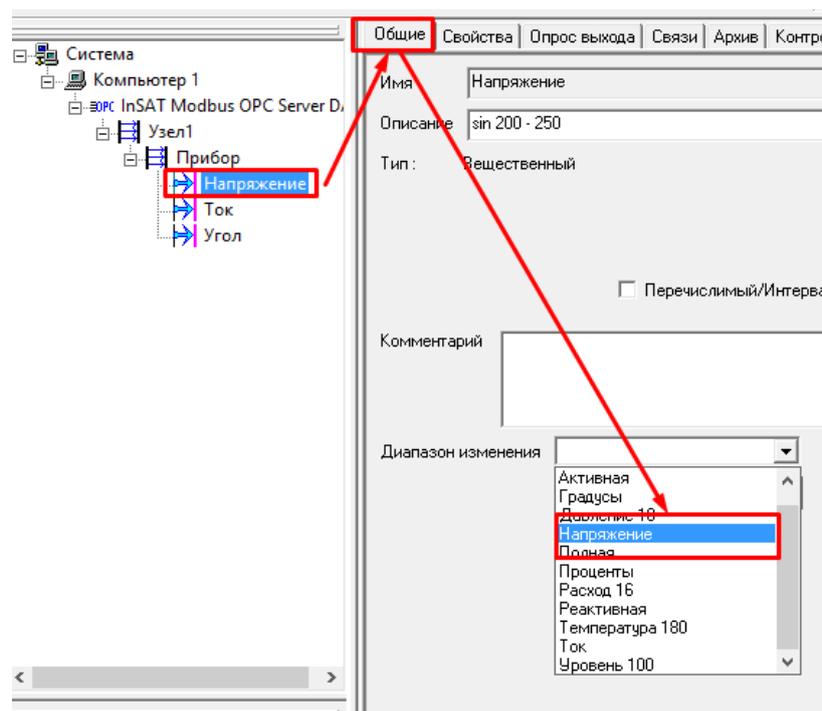


Рисунок 6.38 – Добавление диапазона изменения

Откроем мнемосхему и заметим, что индикаторы поменяли свою шкалу на заданную ранее.

Добавим из палитры мнемосхемы «Тренд» (рисунок 6.39) и подкорректируем размер тренда. Для этого нажимаем левой кнопкой мыши на «Тренд» и на мнемосхеме нарисуем левой кнопкой мыши область тренда.

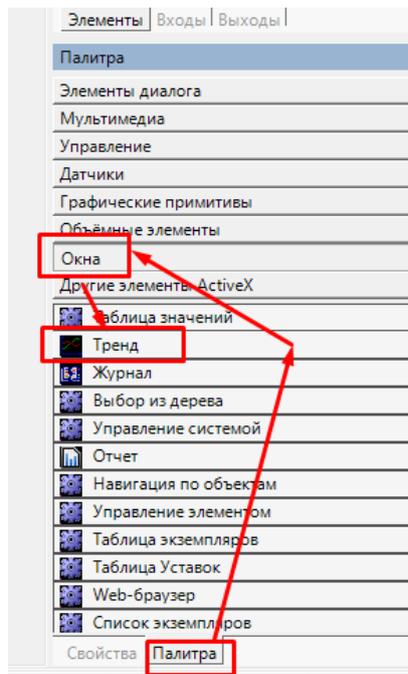


Рисунок 6.39 – Добавление тренда

Добавим переменные в тренд перетаскиванием значений (рисунок 6.40).

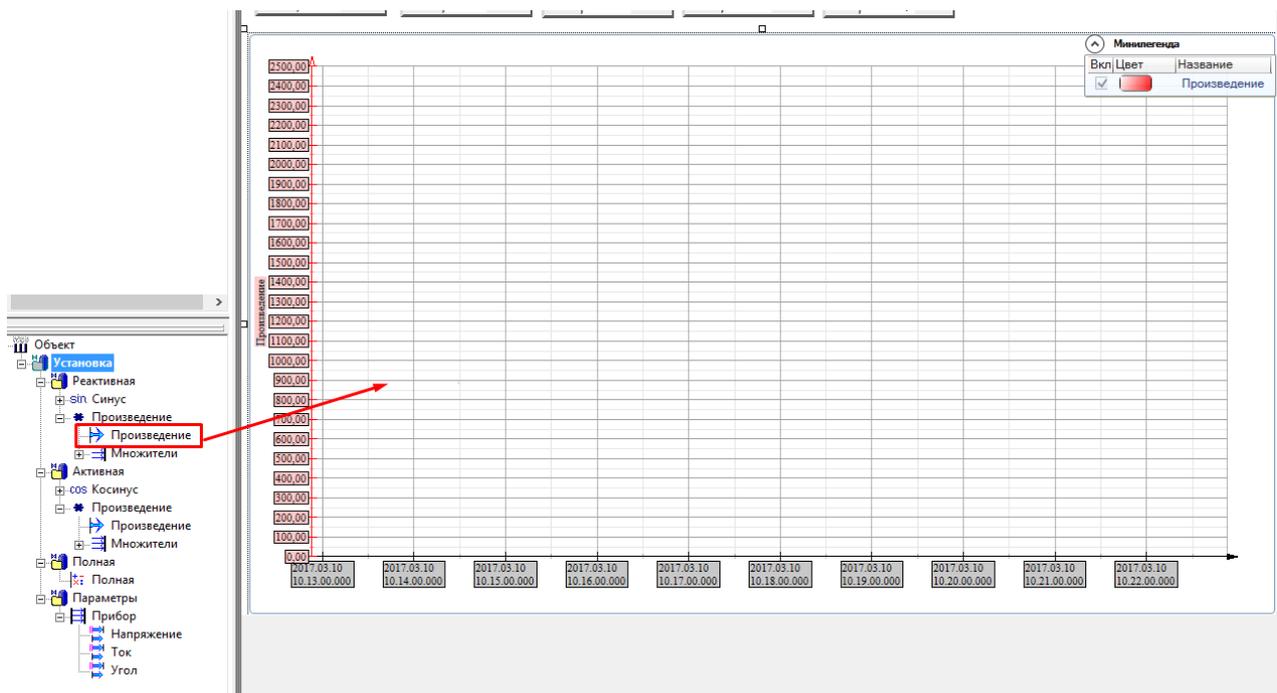


Рисунок 6.40 – Добавление переменных в тренд

Подкорректируем имена трендов, для этого на тренде откроем настройки и выберем значение «Базовый» и переименуем легенду, как показано на рисунках 6.41 и 6.42.

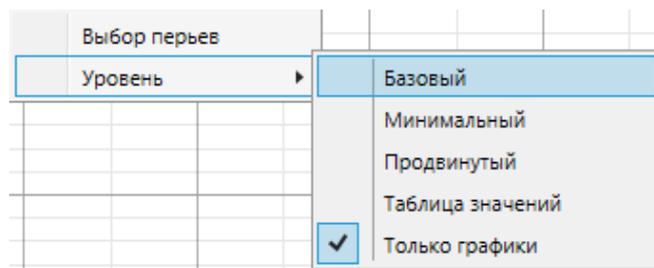


Рисунок 6.41 – Редактирование легенды

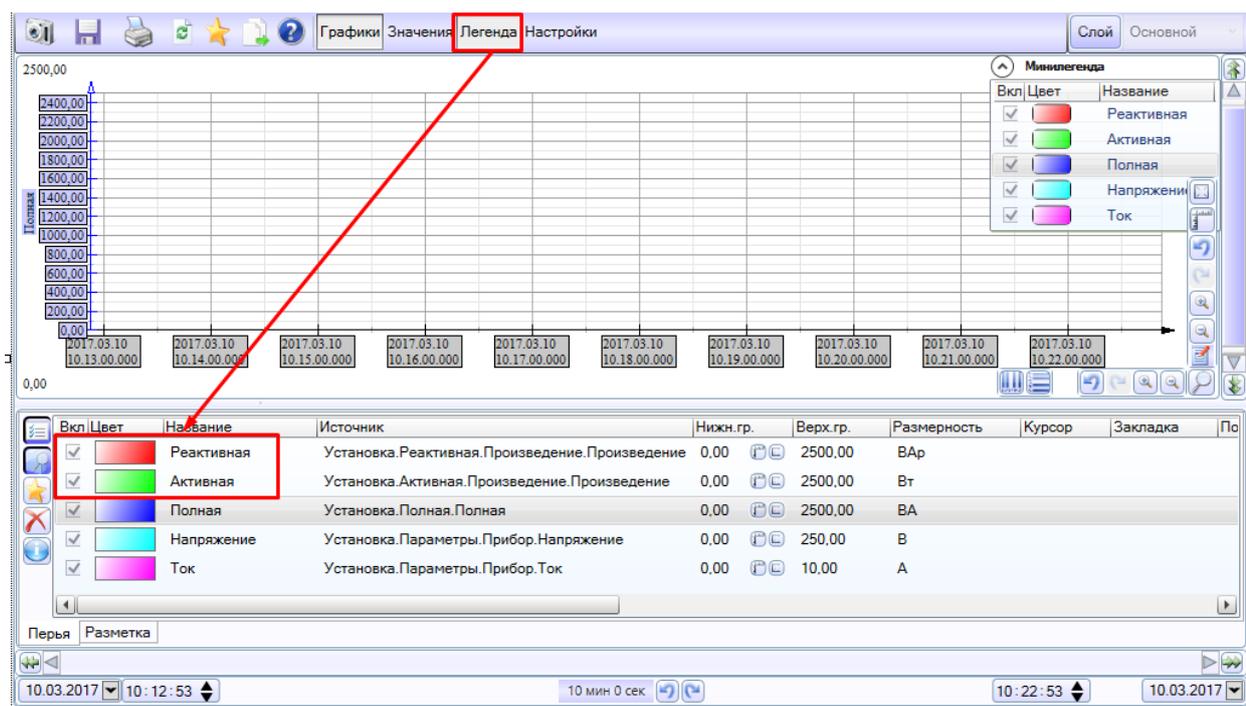


Рисунок 6.42 – Переименование легенды

Как видно, каждому параметру теперь соответствуют следующие индивидуальные значения: «цвет», «название», «источник», «нижние границы», «верхние границы», а также «размерность».

После того как закончите редактирование, необходимо убрать легенду и сохранить проект. Выйдем из мнемосхемы.

Далее для объекта во вкладке «Общие» добавим компьютер (рисунок 6.43) и нажмем «Применить».

Также сделаем так, чтобы при запуске проекта появлялась мнемосхема. Для этого в объекте «Установка» во вкладке «Окна» установить галочку «Стартовая мнемосхема компьютера» (рисунок 6.44).

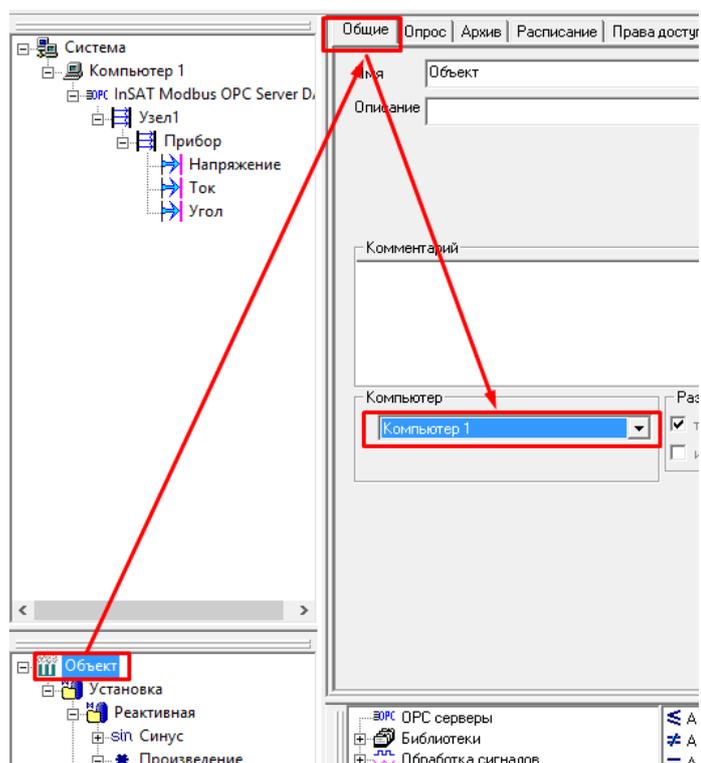


Рисунок 6.43 – Добавление компьютера

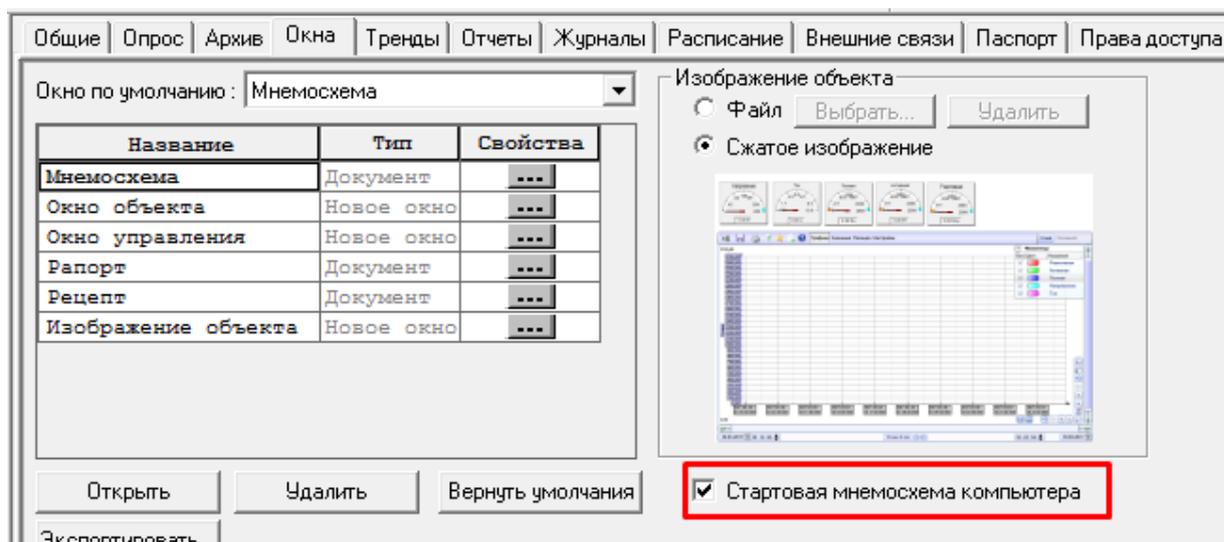


Рисунок 6.44 – Установка стартовой мнемосхемы

Создание проекта завершено. Запустим проект, нажав иконку «Пуск» (рисунок 6.45).

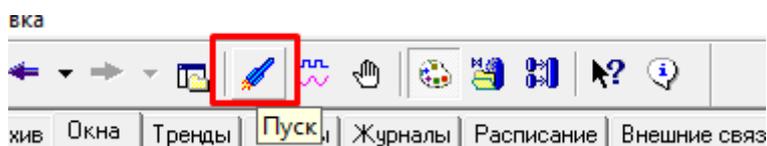


Рисунок 6.45 – Запуск проекта

В появившемся окне нажмем кнопку «Ок». После загрузки появится окно сообщения. В нем отображается различная информация:

- информация о ходе работы оператора системы;
- контроль нарушения границ;
- и другие сообщения.

Параметры задаются через сервер. Далее нужно запустить сервер (рисунок 6.46).

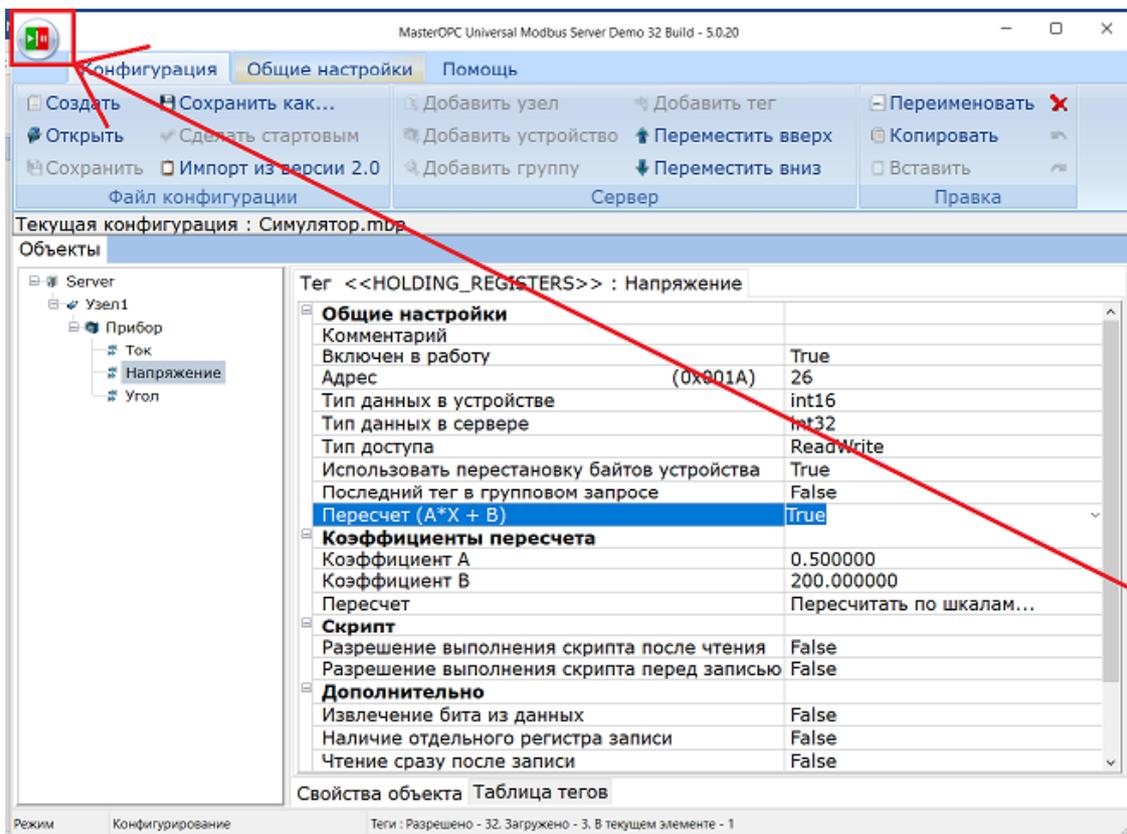


Рисунок 6.46 – Задание параметров на сервере *MasterOPC Universal Modbus Server* и его запуск

После запуска *MasterOPC Universal Modbus Server* необходимо указать значение для переменных в соответствии с вариантом. Для этого двойным щелчком левой кнопкой мыши по полю «EMPTY» колонки «Значение» указываем значение по варианту для каждой переменной и нажимаем «Да». Начнется построение графиков, а на приборах начнет отображаться значение параметров. Если какой-либо из параметров попадет в аварийную или предупредительную зону, то появится сообщение с информацией об этом.

Если информация еще актуальна, то текст сообщения выделен жирным, как показано на рисунке 6.47. На этом создание проекта завершено.

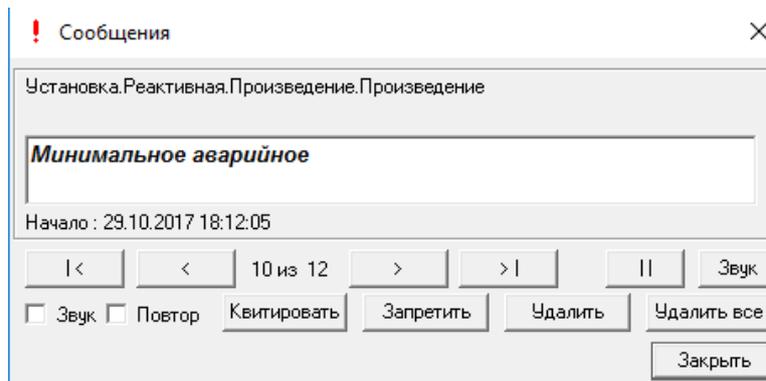


Рисунок 6.47 – Текстовое сообщение

На этом создание проекта с помощью *OPC* сервера в *MasterSCADA*, в котором будет произведен расчет мощностей, завершено.

Задание 2. Создать проект с использованием *OPC* сервера и типизации. Проанализировать полученные данные.

Порядок выполнения задания 2

Создадим конфигурацию в *MasterOPC Universal Modbus Server*. Для этого нажмем «Создать», откроется окно настроек сервера (рисунок 6.48).

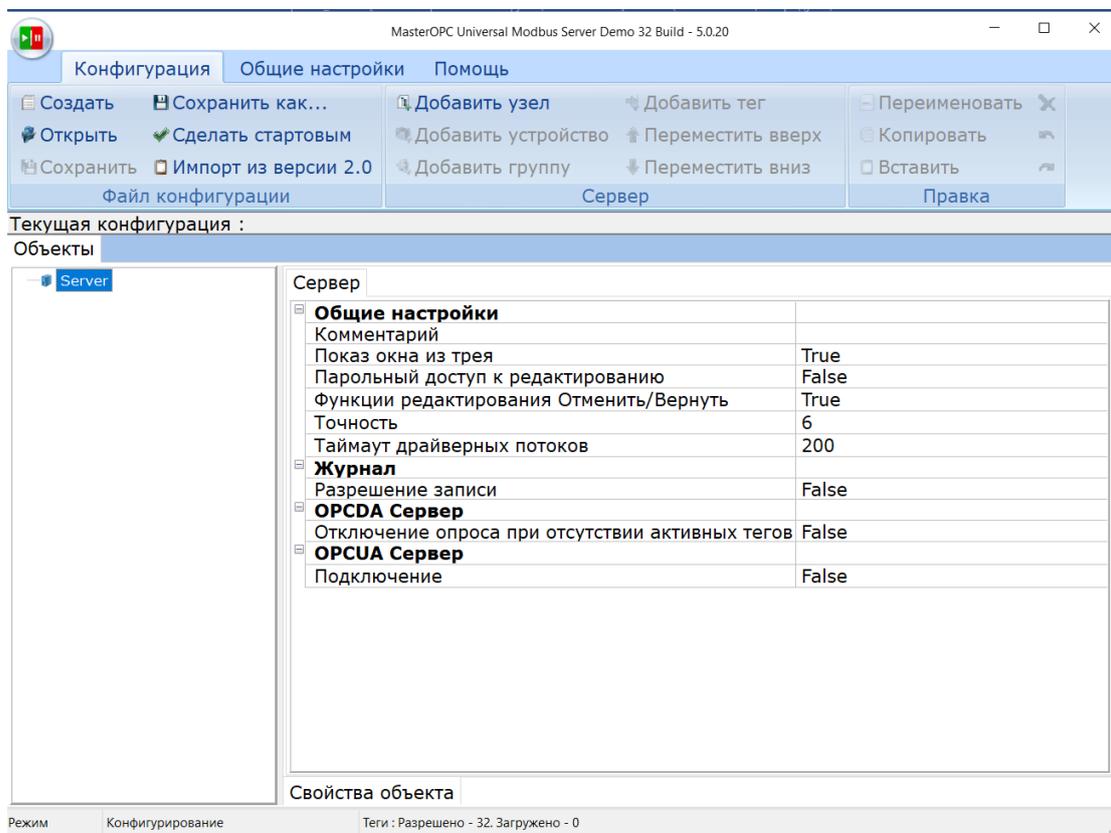


Рисунок 6.48 – Настройки сервера

Далее кликаем правой кнопкой мыши по серверу и нажимаем «Добавить коммуникационный узел», переименовываем его в «Узел». После кликаем правой кнопкой мыши по узлу и нажимаем «Добавить устройство», называем «Устройство 1». Добавляем теги в «Устройство». Нужно добавить три тега с именами «Пила», «Синус» и «Случайный». При создании каждого тега устанавливаем в поле «Тип доступа» значение «ReadWrite». Продублируйте устройства так, чтобы их стало 10.

Итак, есть конфигурация с десятью устройствами, каждое из которых содержит по три тега – «Синус», «Пила» и «Случайный» (рисунок 6.49).

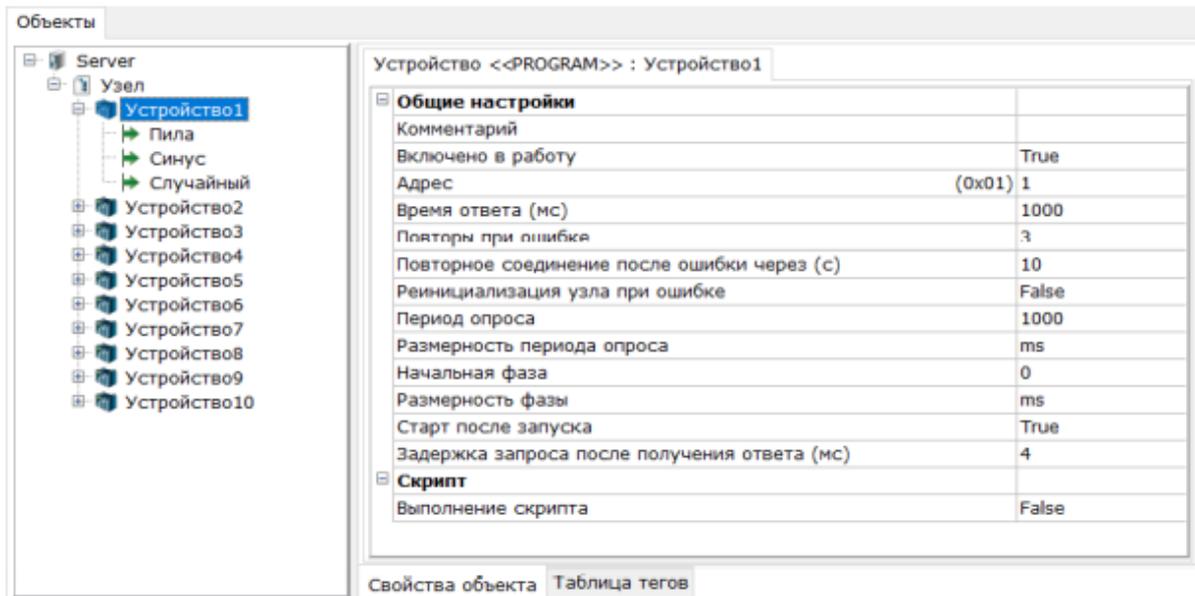


Рисунок 6.49 – Конфигурация устройств

Построим проект с использованием типизации. Сначала добавим в дерево системы компьютер, OPC сервер и переменные (рисунок 6.50).

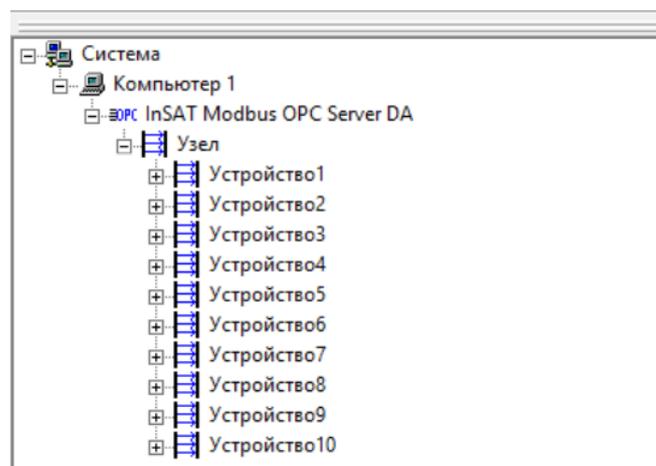


Рисунок 6.50 – Созданное дерево системы

Зайдем в «Устройство 1» и в каждой переменной зададим закон, по которой она будет изменяться (рисунок 6.51).

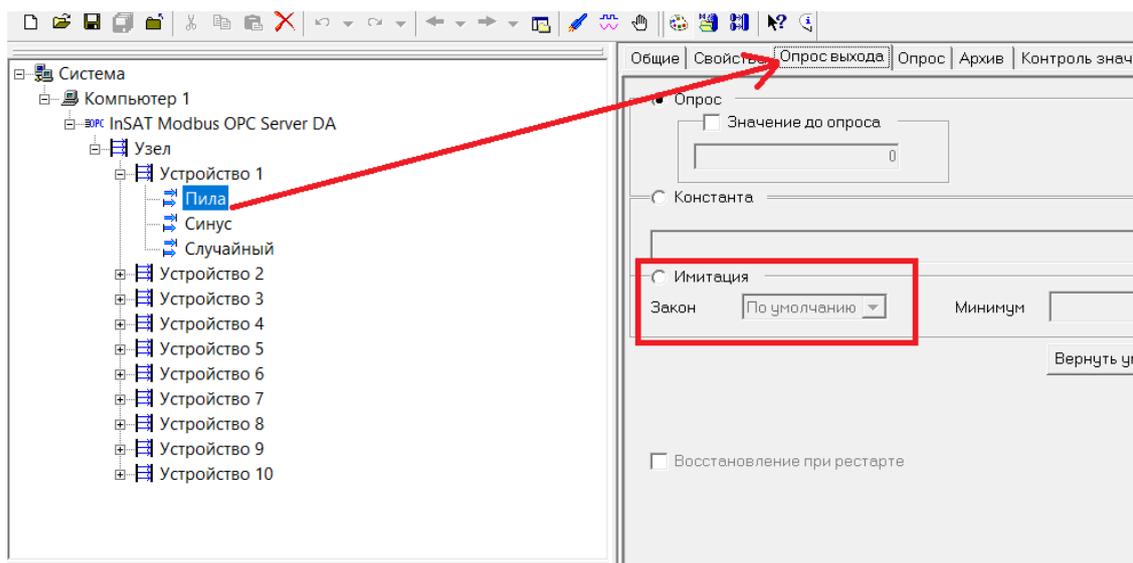


Рисунок 6.51 – Выбор закона

Для переменной «Пила» зададим закон «Пила», для переменной «Синус» зададим закон «Синус», для переменной «Случайный» зададим закон «Шум». Также зададим минимум и максимум по варианту. Прделаем то же самое для остальных устройств в дереве системы, кроме одного на выбор (для остальных устройств «Пила» и «Синус» лучше указывать другие границы того же диапазона). Выбранное устройство будет получать значения через подключенный к *OPC*-серверу проект, выполненный в *LabVIEW* (см. «Лабораторная работа № 5»). Подключаем его к необходимой переменной и не задаем закон изменения, оставляя «Опрос».

ВАЖНО!

Проект из *LabVIEW* необходимо интегрировать непосредственно перед запуском проекта в *MasterSCADA*, предварительно запустив *OPC* сервер. Это обусловлено тем, что проект в *LabVIEW* выполнен с конечным количеством итераций.

Создадим объект в дереве объектов и назовем его «Общий объект», в нем создадим еще один объект и назовем «Устройство 1». Добавим в этот объект три переменных-значения – «Пила», «Синус» и «Случайный». Эти переменные будут индцировать значения из *OPC* сервера. А также сделаем еще одну переменную-значение «Название», которая будет содержать название устройства (рисунок 6.52).

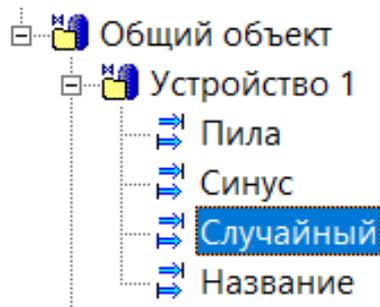


Рисунок 6.52 – Добавленные переменные

Сделаем изображение объекта и несложную мнемосхему.

В мнемосхеме объекта «Устройство 1» добавим тренд и перенесем в него переменные «Пила», «Синус», «Случайный», как показано на рисунке 6.53.

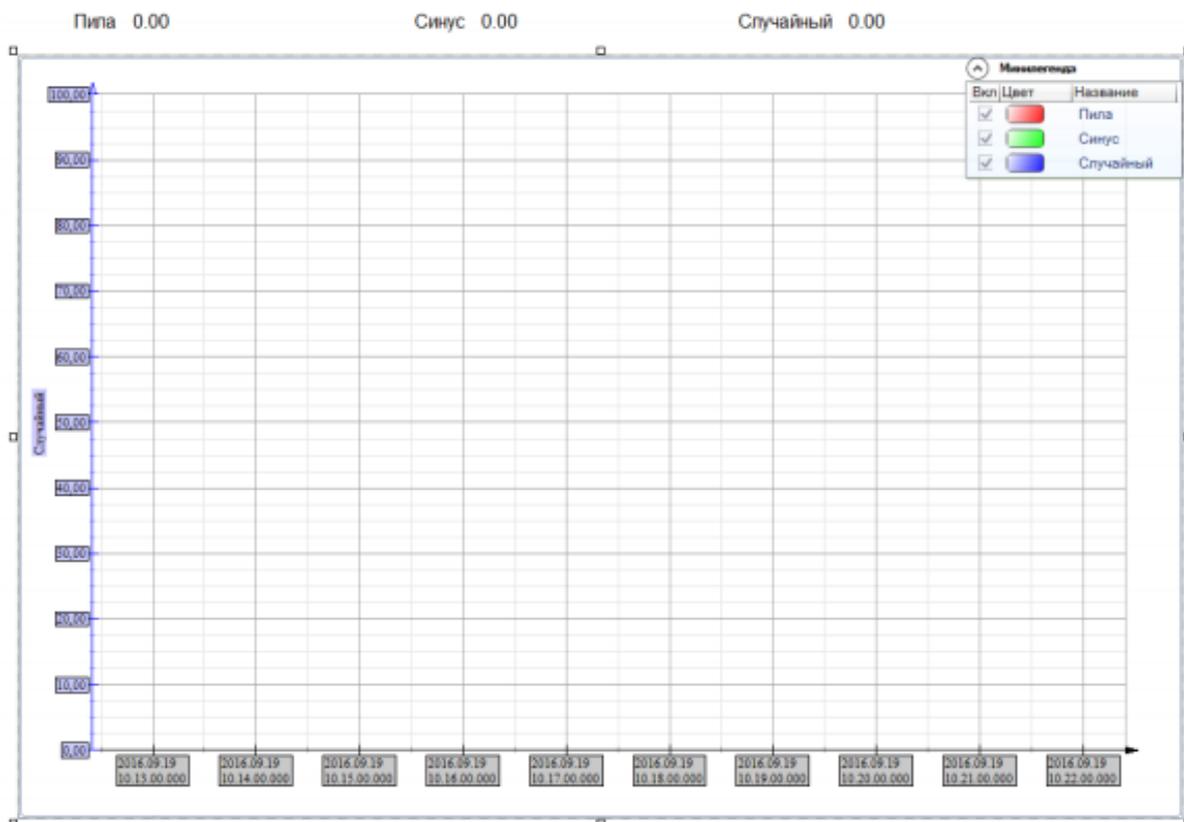


Рисунок 6.53 – Мнемосхема объекта «Устройство 1»

Для создания изображения объекта «Устройство 1» откроем вкладку «Окна», в ней выберем «Изображение объекта» в поле «Окно по умолчанию», выберем «Изображение объекта» в выпадающем списке, затем выделим «Изображение объекта» в таблице и нажмем «Создать». Перенесем переменные на изображение объекта так, чтобы получилось, как на рисунке 6.54.

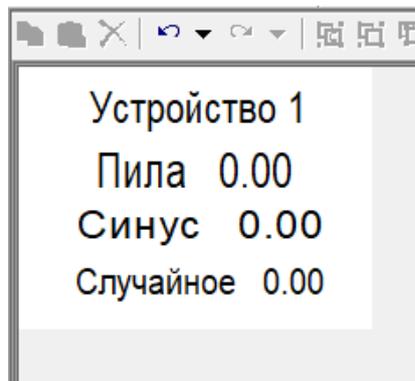


Рисунок 6.54 – Изображение объекта «Устройство 1»

Пока что делаем проект, как и при «классическом» построении, разве что внешние связи не расставляли (внутренние связи в типизированном объекте могут быть). Теперь начнем использовать типизацию.

Первое, что нужно сделать, – это создать список экземпляров. Это делается на вкладке «Система» → «Типизация». Создадим список с такими же именами, как в *OPC* сервере: от «Устройство 1» до «Устройство 10» (рисунок 6.55).

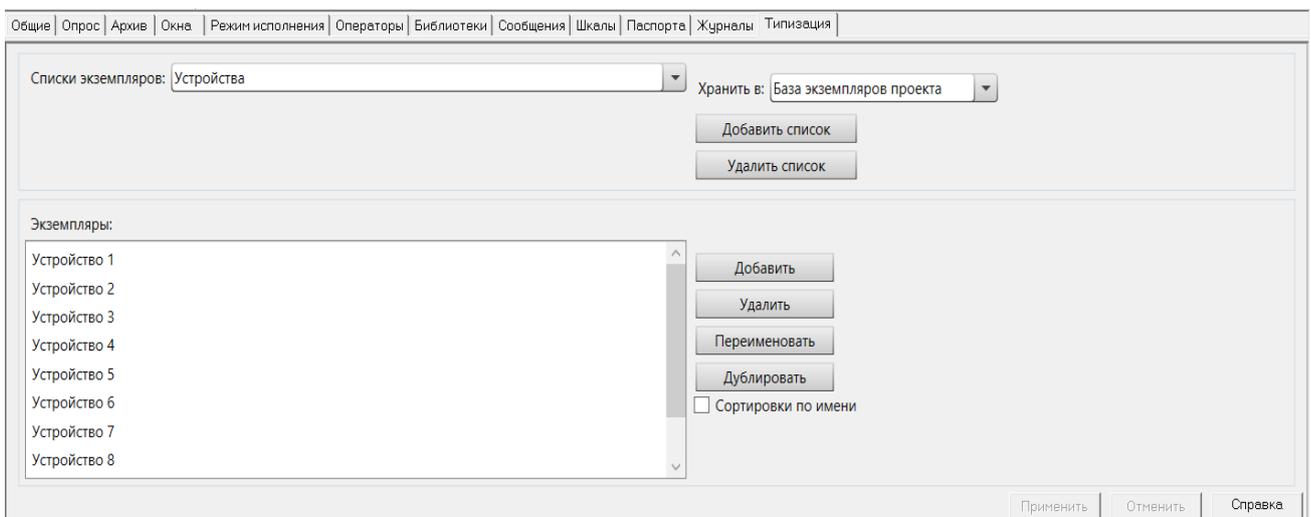


Рисунок 6.55 – Вкладка «Типизация» с созданным списком

Список экземпляров создан.

Следующим шагом необходимо назначить этот список экземпляров нужному объекту, т. е. объекту «Устройство 1» (рисунок 6.56).

Теперь еще один обязательный шаг – у объекта нужно включить собственный цикл на вкладке «Опрос», как показано на рисунке 6.57. Если этого не сделать, то при запуске программа выдаст сообщение об ошибке.

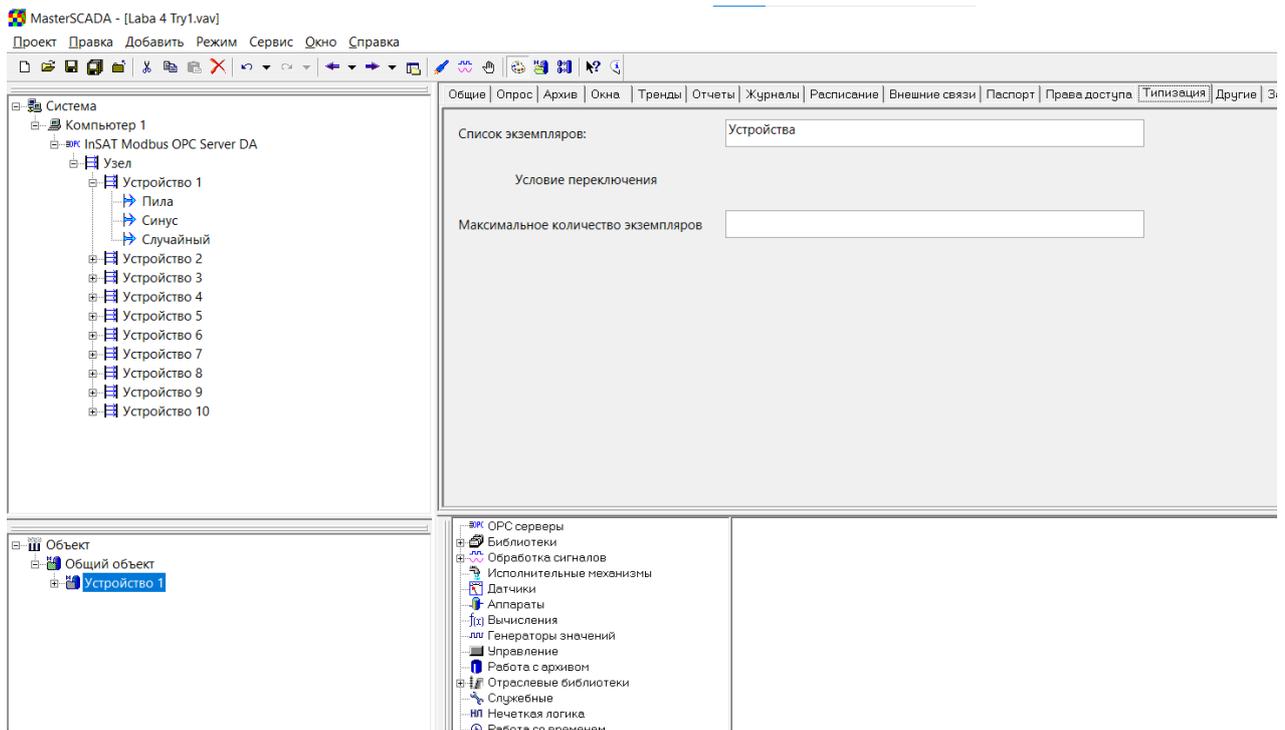


Рисунок 6.56 – Вкладка «Типизация» объекта «Устройство 1»

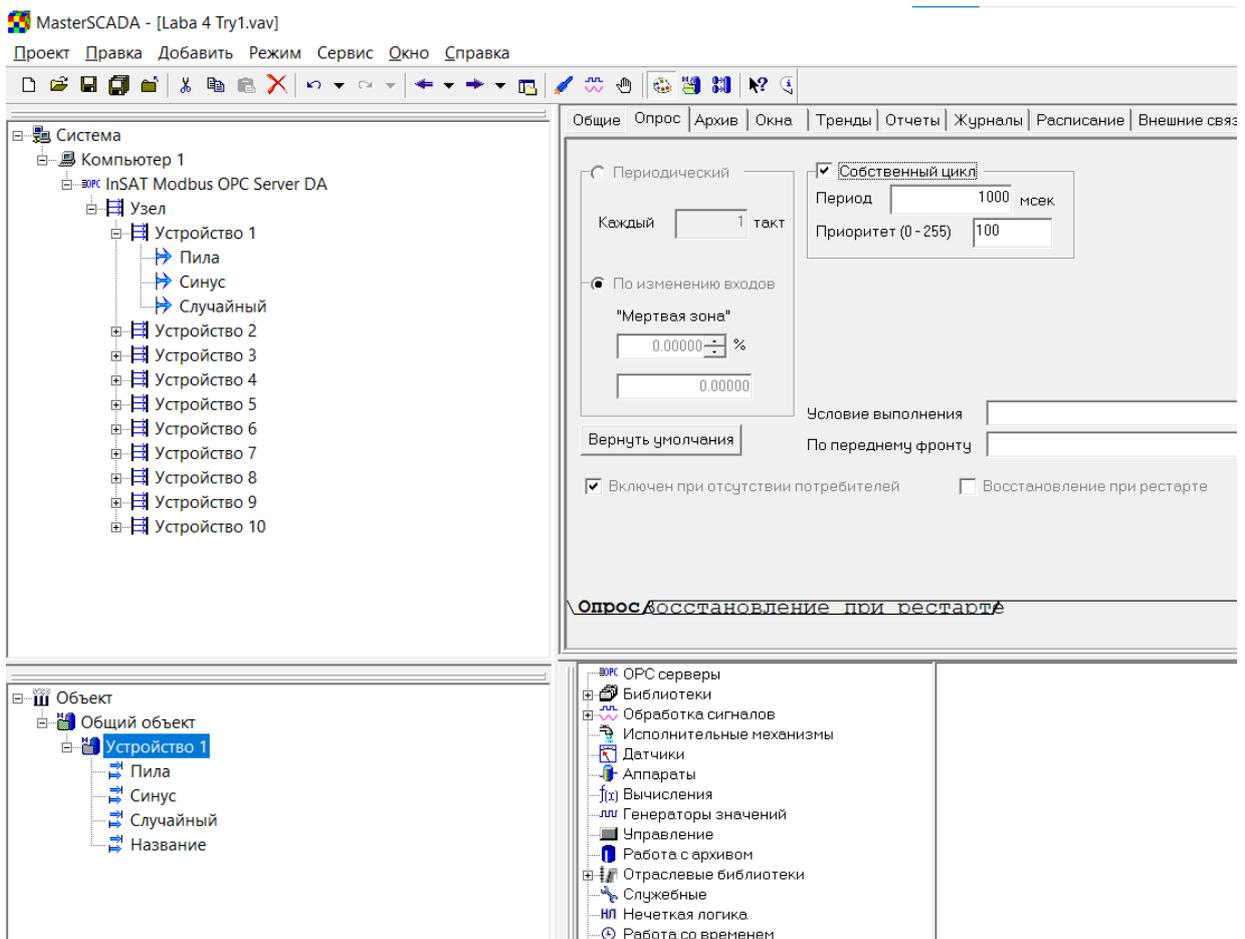


Рисунок 6.57 – Вкладка «Опрос»

Теперь нужно расставить связи. Когда назначили объекту типизацию, у его переменных также появляется вкладка «Типизация», именно на ней и происходит расстановка связей (рисунок 6.58).

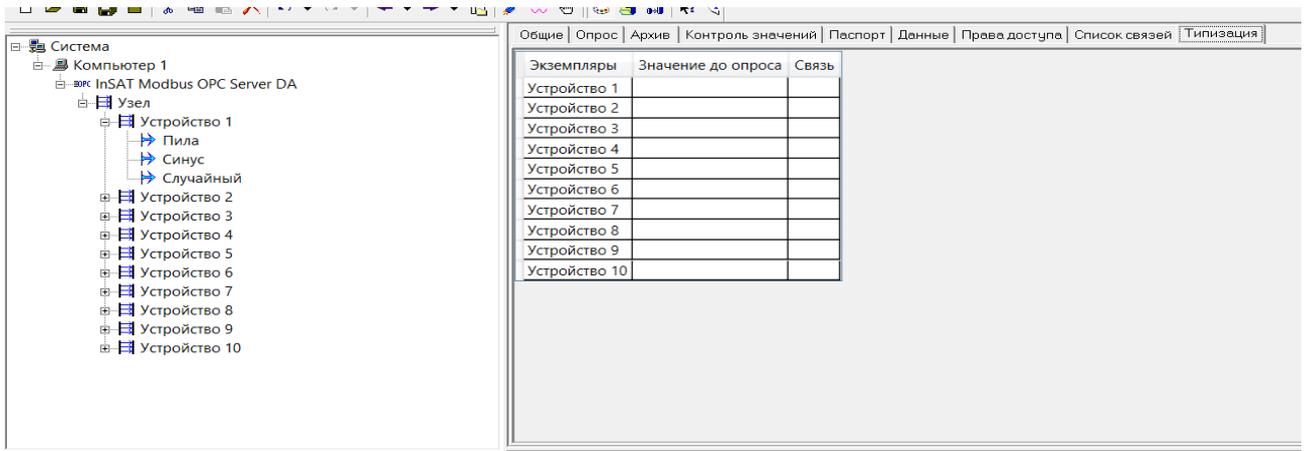


Рисунок 6.58 – Вкладка «Типизация» переменной

Нужно перетащить *OPC* переменную (или любую внешнюю переменную) в нужное поле «Связь». Перетащим *OPC* переменную «Пила» группы «Устройство 1» в поле связи переменной «Пила», созданного объекта «Устройство 1», как показано на рисунке 6.59.

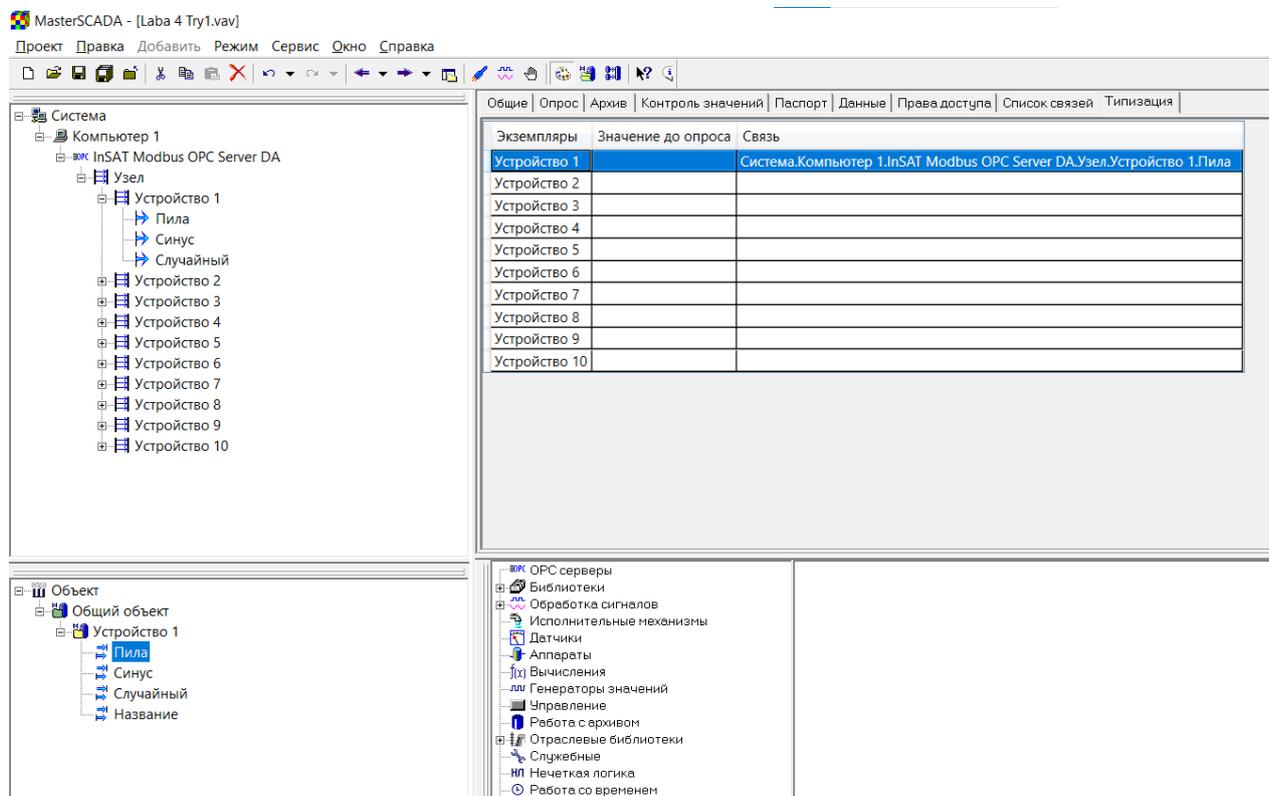


Рисунок 6.59 – Созданная связь переменной «Пила»

По такому же принципу формируем все остальные связи. Для ускорения процесса формирования связей можно копировать текст из полей связи и вставлять в следующее поле, меняя цифру у устройства (рисунки 6.60 – 6.62).

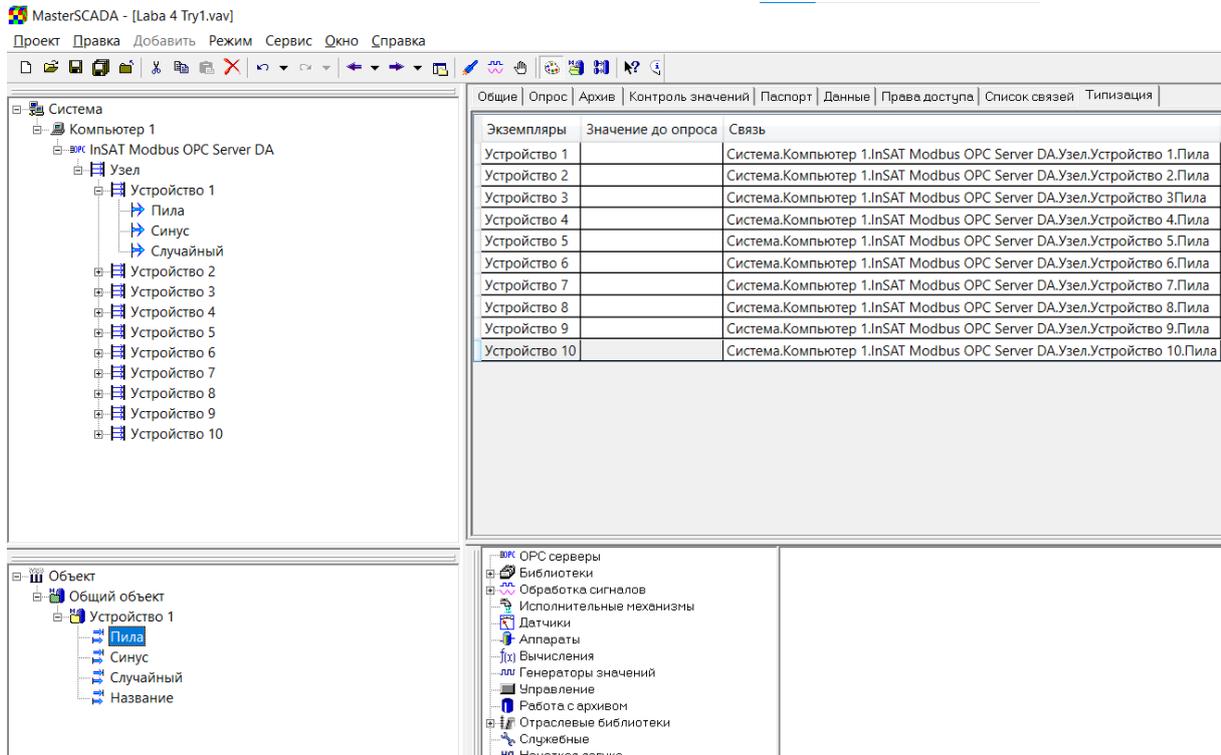


Рисунок 6.60 – Связи переменной «Пила»

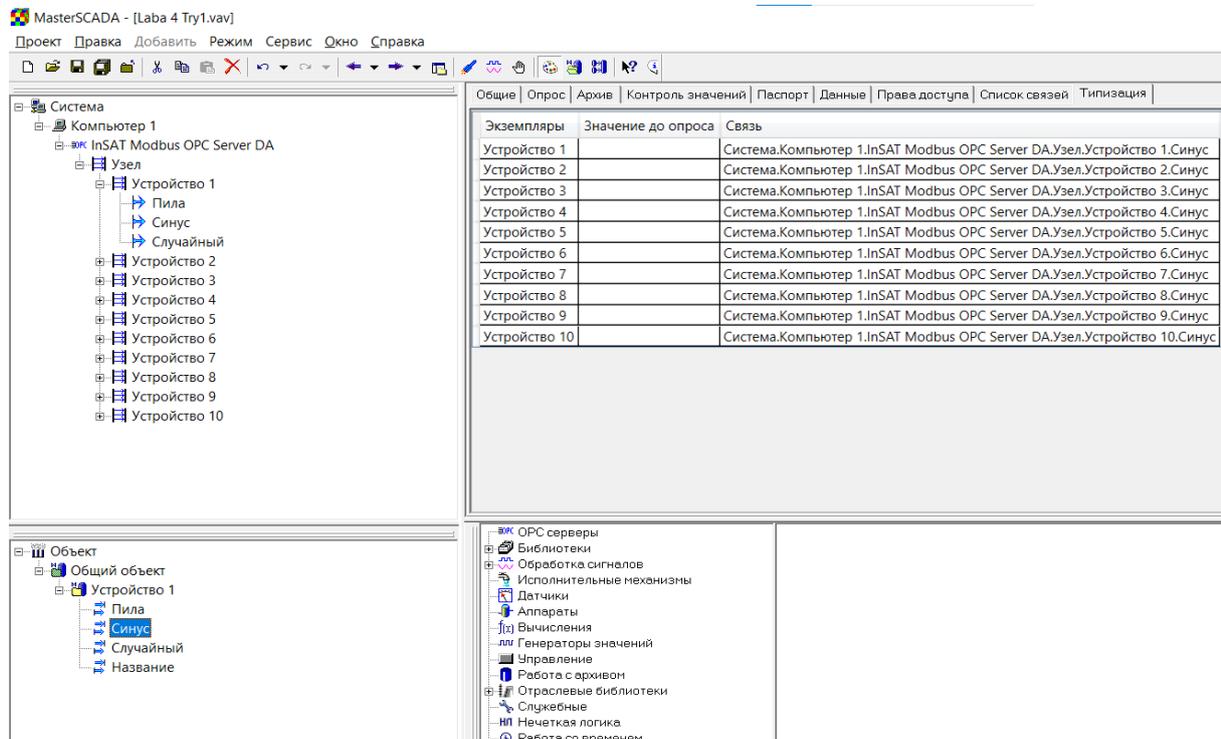


Рисунок 6.61 – Связи переменной «Синус»

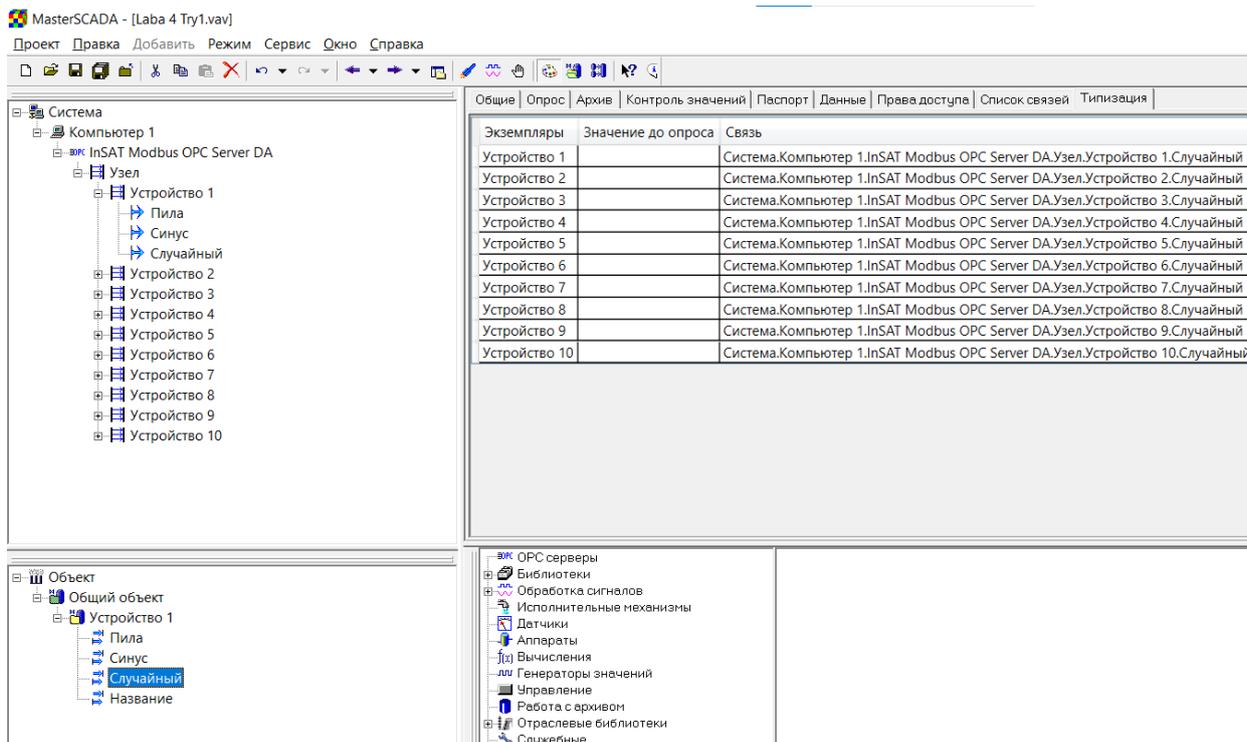


Рисунок 6.62 – Связи переменной «Случайный»

Для переменной «Название» в поле «Значение до опроса» напишем название каждого устройства (рисунок 6.63).

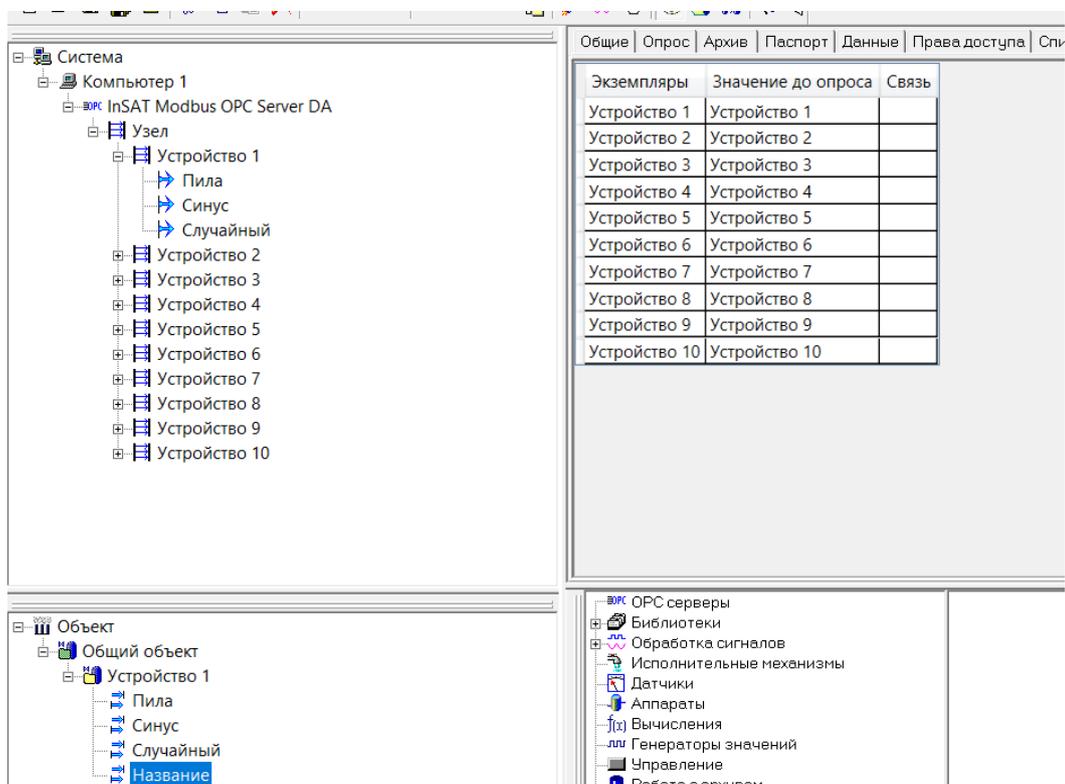


Рисунок 6.63 – Поле «Значение до опроса» переменной «Название»

Для того чтобы использовать типизированные объекты, в дерево их больше добавлять не нужно. Теперь их нужно использовать на мнемосхемах, добавлять изображения объектов, кнопки вызова и т. д. Разместим таким образом изображение объекта «Устройство 1» на мнемосхеме объекта «Общий объект» (рисунок 6.64).

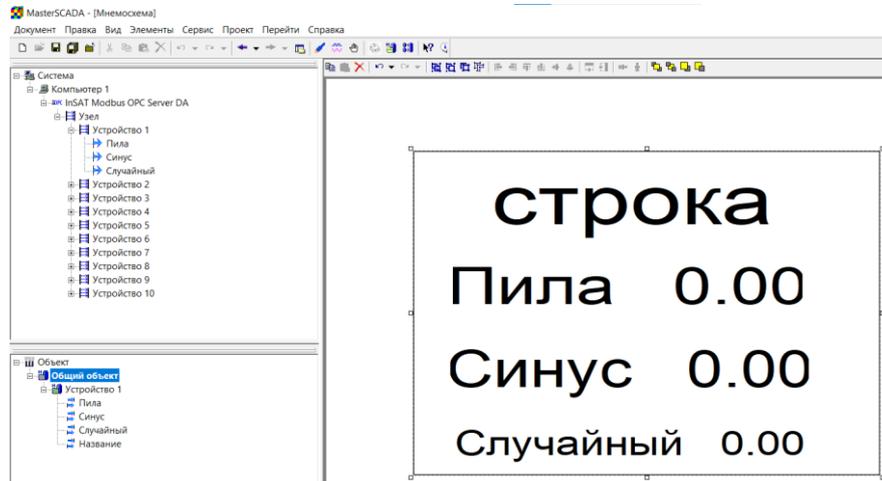


Рисунок 6.64 – Мнемосхема объекта «Общий объект»

Теперь можно запустить проект в режим исполнения, при этом не забыв указать в «Общем объекте» компьютер. Кликая на изображение объекта, можно выбрать требуемое устройство (рисунок 6.65).

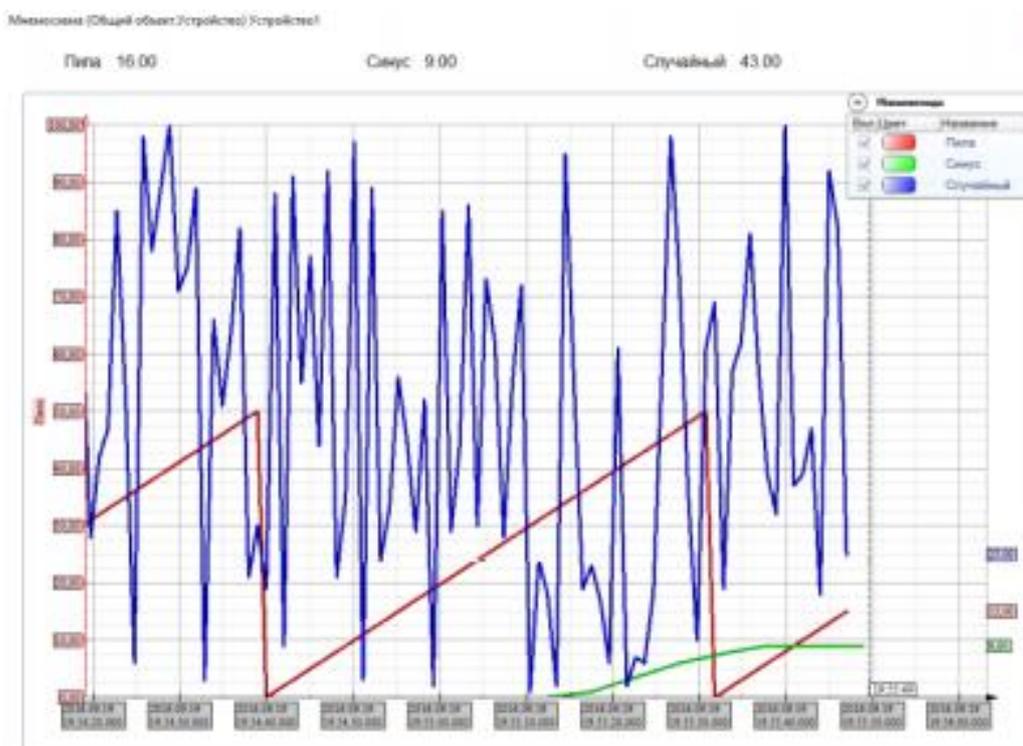


Рисунок 6.65 – Мнемосхема случайного объекта

6.3 Индивидуальные задания

В рамках лабораторной работы предусмотрено выполнение заданий в соответствии с индивидуальным номером зачетной книжки каждого студента. При этом важно учесть, что повторение заданий не допускается, обеспечивая таким образом самостоятельное выполнение своей работы. В случае совпадения номера зачетной книжки с уже занятым заданием студенту предоставляется возможность выбрать свободный номер из предложенных вариантов.

Варианты для задания 1 и задания 2 представлены в таблицах 6.1 и 6.2 соответственно.

Таблица 6.1 – Варианты индивидуального задания 1

Вариант	Значение для переменной «Ток»	Значение для переменной «Напряжение»	Значение для переменной «Угол»
1	9	217	336
2	3	190	152
3	1	230	220
4	4	105	272
5	6	140	91
6	8	154	290
7	10	99	175
8	3	23	156
9	2	231	256
10	7	157	186
11	8	86	252
12	6	10	155
13	2	93	89
14	3	12	324
15	9	163	280
16	9	131	56
17	2	228	58
18	7	142	5
19	4	57	339
20	3	217	129
21	4	165	131
22	9	232	31
23	2	173	87
24	7	184	332
25	6	148	148
26	2	212	169
27	6	17	111
28	5	133	49
29	8	134	55
30	4	5	298

Таблица 6.2 – Варианты индивидуального задания 2

Вариант	Диапазон «Синус»	Диапазон «Пила»	Диапазон «Случайный»
1	0–50	0–20	0–100
2	10–45	5–30	10–70
3	8–30	5–31	0–97
4	12–60	4–27	16–74
5	10–55	12–29	9–100
6	0–30	15–32	12–119
7	5–44	4–25	16–89
8	15–56	0–38	18–76
9	19–63	2–21	25–70
10	14–76	6–42	21–117
11	11–41	7–27	3–110
12	5–75	10–41	8–106
13	0–68	4–33	24–73
14	7–46	4–19	21–119
15	20–77	15–26	20–99
16	9–43	5–28	5–80
17	4–77	11–31	14–88
18	13–54	14–39	6–93
19	6–41	9–19	10–89
20	14–52	6–41	19–97
21	6–59	13–45	18–90
22	11–50	8–38	23–120
23	12–47	11–29	16–77
24	14–73	1–25	14–115
25	8–49	3–44	1–73
26	2–43	8–39	0–90
27	13–76	6–44	5–80
28	5–70	8–25	3–83
29	1–47	0–35	1–79
30	13–42	11–37	22–85

6.4 Содержание отчета

В рамках лабораторной работы № 6 необходимо выполнить два индивидуальных задания. В связи с этим отчет по лабораторной работе должен содержать:

- 1) титульный лист;
- 2) введение:
 - 2.1) цель выполнения задания;
 - 2.2) краткое описание задания;
- 3) основная часть:
 - 3.1) ход выполнения задания;
 - 3.2) обработка и анализ результатов выполнения задания;
- 4) выводы по результатам выполнения задания.

6.6 Контрольные вопросы

- 1 Что такое мнемосхема?
- 2 Напишите формулы расчета полной, реактивной и активной мощности.
- 3 Что такое SCADA-системы?
- 4 Какое основное назначение у OPC серверов?
- 5 Какие области применения у OPC серверов?
- 6 В каких случаях для управления используют аппаратные мощности MasterSCADA?
- 7 В каких случаях для управления используют данные, полученные от OPC сервера?
- 8 Как типизация позволяет автоматизировать процесс расстановки связей между объектами?
- 9 Типизация позволяет удобно использовать динамические объекты. Приведите пример использования типизации для динамических объектов.
- 10 Какие закономерности можно выявить в поведении переменных «Пила», «Синус», «Случайный»?

Вспомогательная литература

- 1 Пьявченко, Т. А. Проектирование АСУТП в SCADA-системе / Т. А. Пьявченко, Н. А. Куцевич, О. В. Синенко. – Таганрог : ТРТУ, 2007. – 78 с.
- 2 Андреев, Е. Б. SCADA-системы: взгляд изнутри / Е. Б. Андреев, Н. А. Куцевич, О. В. Синенко. – М. : ООО «РТ Софт», 2004. – 197 с.
- 3 Барашко, О. Г. Автоматика, автоматизация и автоматизированные системы управления / О. Г. Барашко, Н. А. Куцевич, О. В. Синенко. – Минск : БГТУ, 2011. – 327 с.
- 4 Кангин, В. В. Разработка SCADA-систем: учеб. пособие / В. В. Кангин, М. В. Кангин, Д. Н. Ямолдинов. – Москва, Вологда : Инфра-Инженерия, 2019. – 565 с.
- 5 Интегрированные системы проектирования и управления: SCADA-системы : учеб. пособие / И. А. Елизаров [и др.]. – Тамбов : Изд-во ФГБОУ ВПО «ТГТУ», 2015. – 160 с
- 6 ИнСАТ «Master SCADA. Руководство пользователя. Версия 3.X.». – Москва, 2017. – 574 с.
- 7 ИнСАТ «Master SCADA. Основы проектирования: методическое пособие». – Москва, 2016. – 277 с.
- 8 Master SCADA: учеб. пособие для студентов специальности 5311000 – Автоматизация и управление технологических процессов (в водном хозяйстве) / Р. Т. Газиева [и др.]. – Ташкент, 2020. – 105 с.

Лабораторная работа № 7
Создание проекта в *MasterSCADA* по отслеживанию
уровня жидкости в баке и контролю его наполнения

Цель работы: в среде *MasterSCADA* разработать программу по отслеживанию уровня жидкости в баке, а также с функцией контроля степени наполнения.

Основные задачи:

- 1 Разработать математическую модель процесса наполнения бака жидкостью.
- 2 Спроектировать систему автоматического управления подачей жидкости в бак.
- 3 Получить практические навыки работы с инструментами *MasterSCADA*.
- 4 Закрепить полученные навыки и умения путем выполнения задания.

7.1 Теоретические сведения

7.1.1 Умный бак: как автоматизация помогает экономить воду

Рассмотрим сценарий водоснабжения, при котором из бака с водой происходит ее бесперебойная подача. Если уровень в баке снижается ниже заданного порога, система автоматически открывает клапан для подачи дополнительной воды. Когда уровень достигает предельной отметки, клапан закрывается, предотвращая перелив и оптимизируя расход воды.

Необходимо создать проект, в котором будет осуществляться поддержание постоянного уровня жидкости в баке (рисунок 7.1). На мнемосхеме необходимо отобразить бак с жидкостью и параметры: «входной поток», «выходной поток» и «установка».

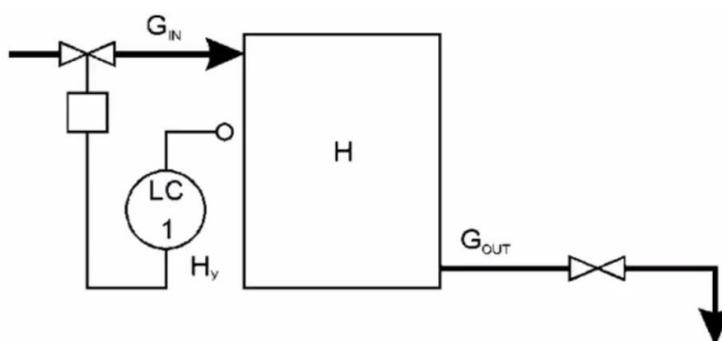


Рисунок 7.1 – Вид модели проекта

Пользователь может изменять вышеприведенные параметры, вводя их с клавиатуры, при этом текущий уровень жидкости в баке H должен высчитываться по формуле

$$H = H + (G_{in} - G_{out}), \quad (7.1)$$

где G_{in} – входной поток, л;
 G_{out} – выходной поток, л.

Уровень должен выводиться на мнемосхему в виде графика и числового значения.

7.1.2 Автоматизация процесса смешивания жидкостей

Объект управления состоит из трех емкостей с жидкостью, аппарата для смешивания, а также вспомогательного оборудования.

Аппарат для смешивания представляет собой емкость с мешалкой, в которую через клапаны заливается часть жидкости из баков в заданной пропорции. После заполнения емкости субстанцией запускается ее перемешивание.

От SCADA-системы требуется отображать состояние параметров:

- уровень в баках;
- уровень в емкости;
- количество жидкости, прошедшей через каждый из клапанов.

Параметры, задаваемые оператором:

- количество каждой жидкости, необходимое для смешивания;
- количество жидкости, выкачиваемое из бака, за единицу времени.

Баки с жидкостью в данном проекте следует понимать, как условное обозначение, так как это могут быть любые резервуары с каким-либо наполнением. По линиям передачи содержимое данных резервуаров поступает в пункт, где происходит смешивание. Такое обозначение выбрано для удобства визуализации происходящих процессов и их непосредственного контроля.

7.2 Задания для выполнения лабораторной работы

Задание 1. Создать проект среде *MasterSCADA* по отслеживанию уровня жидкости в баке.

Порядок выполнения задания 1

Создадим шкалы для удобства отображения всех необходимых параметров. Для этого в дереве системы выбираем «Система» и на панели свойств выбираем вкладку «Шкалы» (рисунок 7.2).

Далее создаем две шкалы: «Уровень» и «Поток». Уровень жидкости будет измеряться в сантиметрах и изменяться от 0 до 100. Чтобы создать новую шкалу, нужно кликнуть по любой ячейке в таблице аналоговых значений, а затем нажать на кнопку «Добавить» (рисунок 7.3).

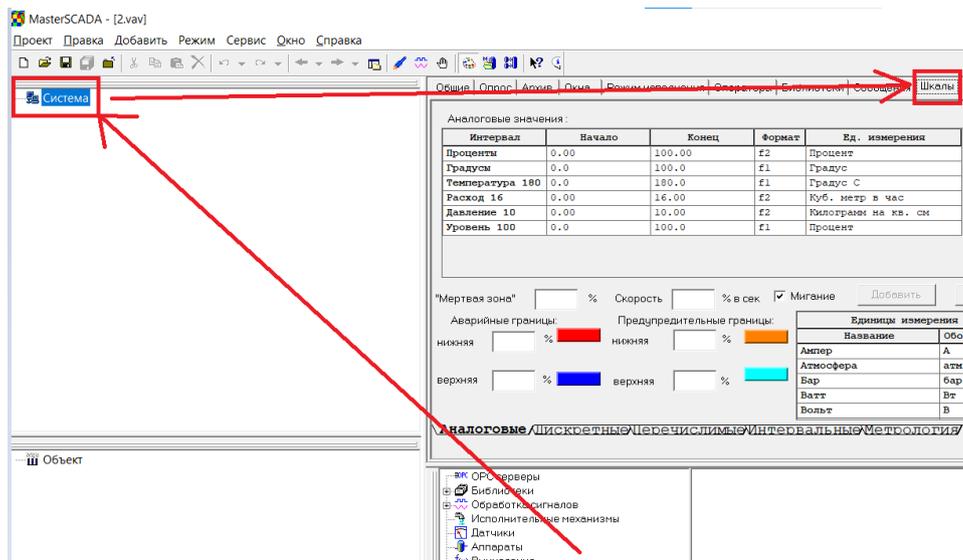


Рисунок 7.2 – Задание шкалы

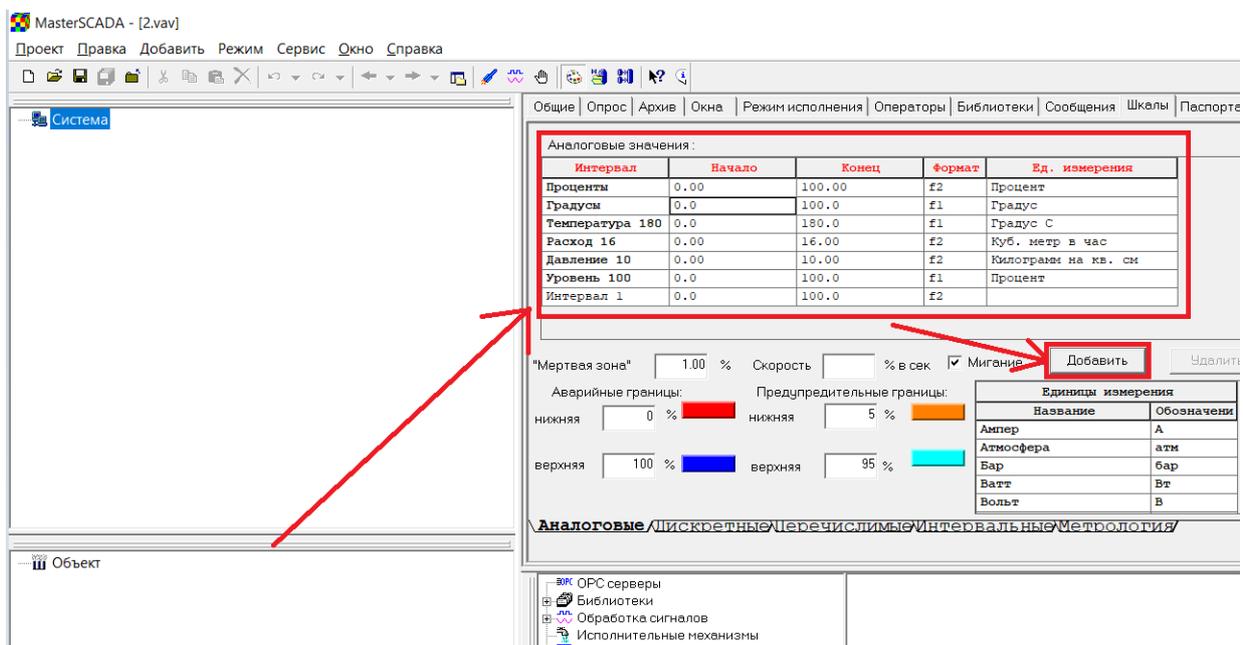


Рисунок 7.3 – Добавление шкалы

Для редактирования значений нужно нажать на ячейку, которую хотим отредактировать, и ввести нужное значение.

Добавляем новую шкалу «Уровень 100 см» с форматом вывода плавающей точки «f1». Добавим шкалу «Поток 10 л/мин», измеряемую в л/мин в диапазоне от 0 до 10. Так как единицы измерения литр в минуту нет в библиотеке, создадим новую единицу измерения.

Для того чтобы добавить новую единицу измерения, нужно по аналогии с добавлением новой шкалы кликнуть на любую ячейку в таблице «Единицы измерения» и нажать кнопку «Добавить» (рисунок 7.4).

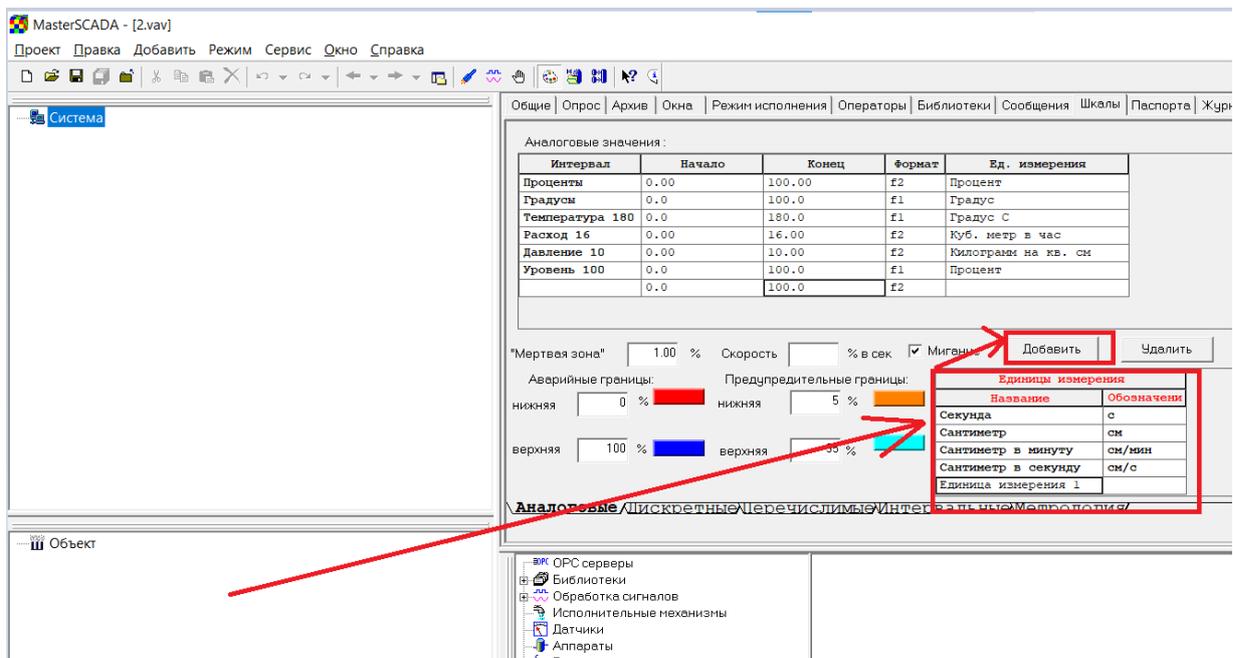


Рисунок 7.4 – Создание новой единицы измерения

Создали две шкалы и применили изменения. Добавляем новый компьютер в систему. OPC сервер добавлять не будем, так как достаточно только встроенных объектов. Добавляем новый объект в дереве объектов и переименовываем в «Аппарат». Создадим команду с именем «Входной поток» с диапазоном измерения «Поток 10 л/мин». Переходим на вкладку «Опрос выхода» и задаем начальный параметр, равный 6, в графе «Значение до опроса». Создаем еще одну команду «Выходной поток» с тем же диапазоном и начальным параметром, равным 5 л/мин. Добавляем команду «Установка» со шкалой «Уровень 100 см» и начальным 20 см. Уровень является расчетным параметром и только выводится, поэтому создаем в объекте «Аппарат» элемент «Значение» с именем «Уровень» и шкалой «Уровень 100 см». Необходимые переменные для работы созданы. Для реализации математической модели создаем элемент «Расчет» в объекте «Аппарат» с именем «Расчет уровня» и шкалой измерения шкалой «Уровень 100 см». На вкладке «Формула» в нижнюю вкладку необходимо перетащить параметры «Уровень», «Входной поток» и «Выходной поток» (рисунок 7.5).

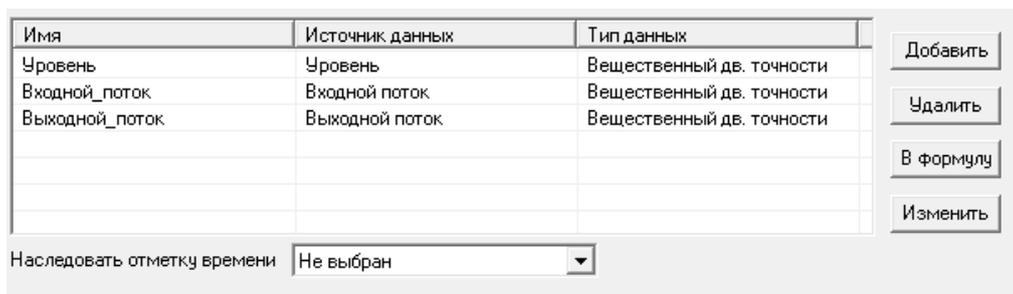


Рисунок 7.5 – Добавление переменных

Из этих трех переменных создаем математическую модель. Двойным нажатием добавляем переменные в формулу (рисунок 7.6).

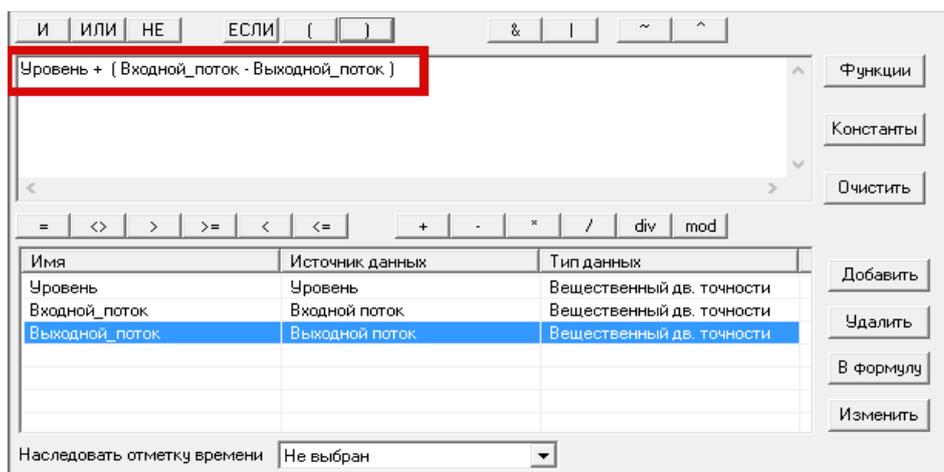


Рисунок 7.6 – Создание математической модели

Далее в «Расчет уровня» заходим во вкладку «Опрос выхода», устанавливаем галочку в поле «Значение до опроса» (рисунок 7.7).

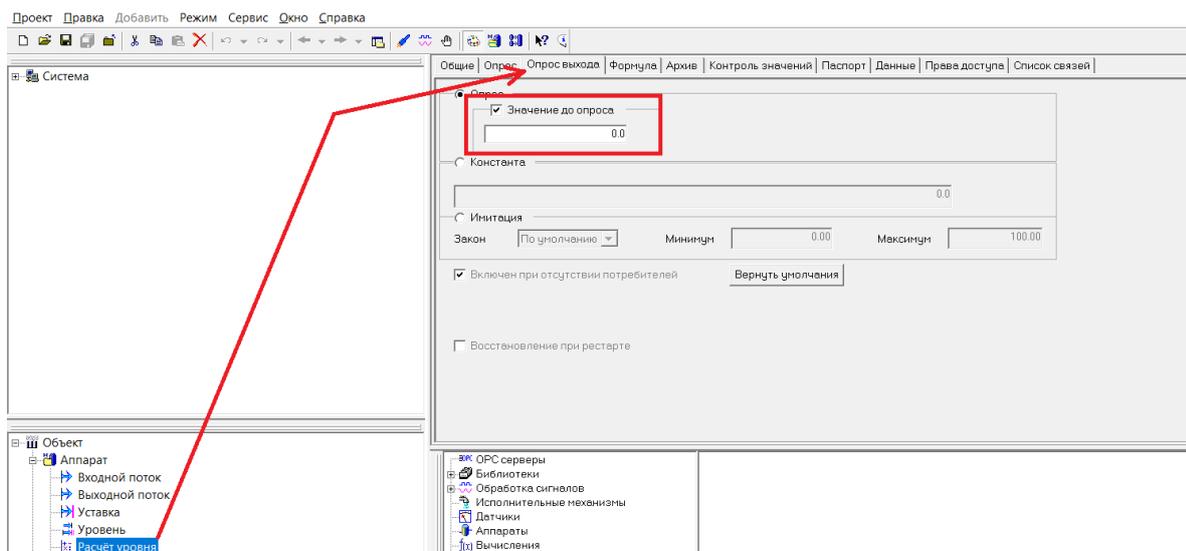


Рисунок 7.7 – Установка значения до опроса

Создаем связь между «Расчет уровня» и значением «Уровень». Добавляем элемент «Емкость» из библиотеки «Аппараты». Создаем связь между значением «Уровень» и входным параметром «Уровень» объекта «Емкость». Выделяем объект «Емкость» и на вкладке «Вид» выбираем тип отображения «Бочка».

Создаем мнемосхему объекта «Аппарат». Перетаскиваем объект «Емкость». Добавляем трубопроводы в палитре мнемосхемы из вкладки «Объемные элементы» и параметры модели (рисунок 7.8). Двойной щелчок левой кнопкой мыши при выбранном трубопроводе создает искривление трубы в направлении курсора мыши.

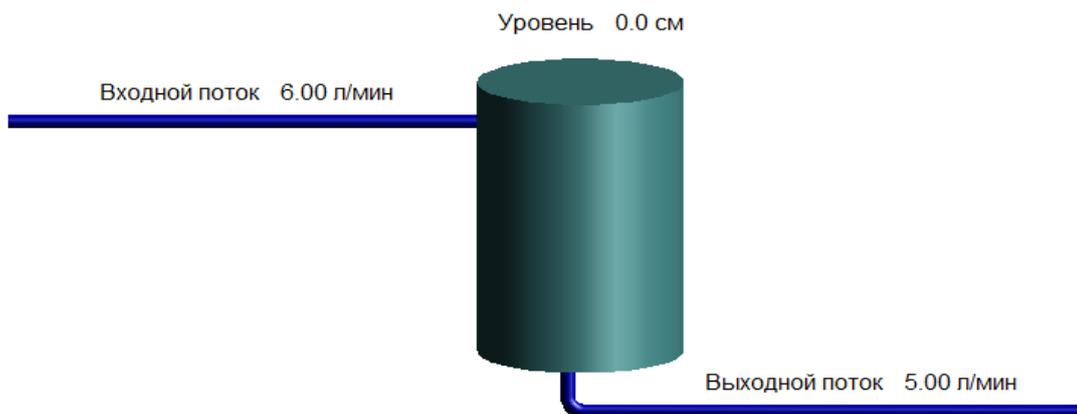


Рисунок 7.8 – Создание мнемосхемы

Перетаскиваем элемент «Установка» правой кнопкой мыши и выбираем «Насос». Далее на палитре «Исполнительные механизмы» выбираем элемент «Турбина» и располагаем его на мнемосхеме на входе трубопровода. Далее заходим в параметры «Свойства» и изменяем изображение элемента (рисунок 7.9).

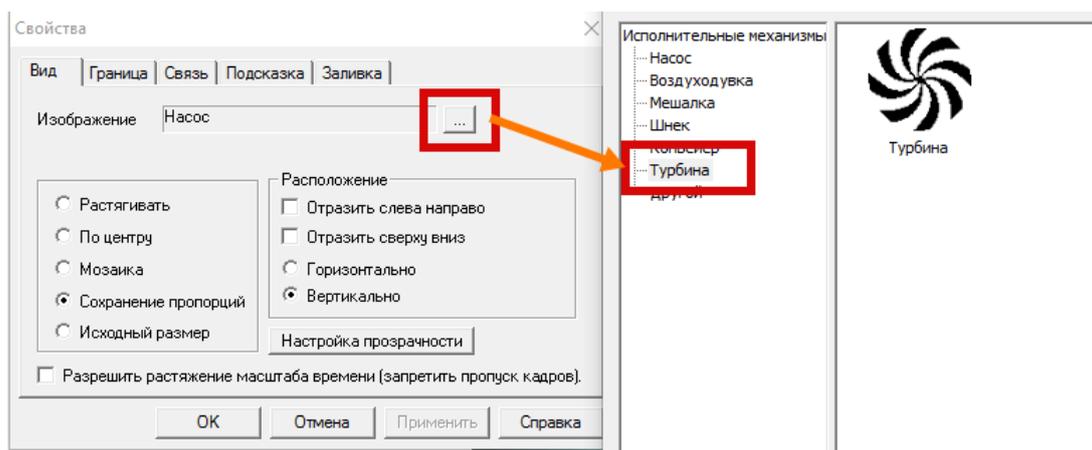


Рисунок 7.9 – Изменение отображения элемента

Сохраняем и закрываем мнемосхему. Выделяем в дереве объекта «Объект» и добавляем в выпадающем списке «Компьютер 1». Выделяем «Аппарат», переходим на вкладку «Окна», выделяем «Мнемосхема» и ставим галочку «Стартовая мнемосхема компьютера». Запускаем проект и проверяем работоспособность. Закрываем мнемосхему.

Добавляем «Регулятор» из библиотеки «Управление» в объект «Аппарат». Во вкладке «Настройки» устанавливаем «Каскад» для значения «Режим при старте». Из всех параметров «Регулятора» в проекте необходимы будут три: «Параметр», «Задание» и «Управление».

Параметр – это величина, по которой ведется управление уровнем. Создаем связь между значением «Уровень» и входным элементом «Параметр». Создаем связь между командой «Установка» и входным элементом «Задание». В

элементе «Настройки» объекта «Регулятор» выделим выходной параметр «Т интегр», перейдем во вкладку «Опрос» и установим константу, равную 10. Аналогичные действия произведем с параметром «Т диффер» с константой 1,25.

Добавляем в объект «Аппарат» еще один расчет с именем «Расчет входного потока». Добавляем шкалу «Поток 10 л/мин». Во вкладке «Формула» добавляем переменные «Входной поток» и «Управление». Создаем формулу, как показано на рисунке 7.10.

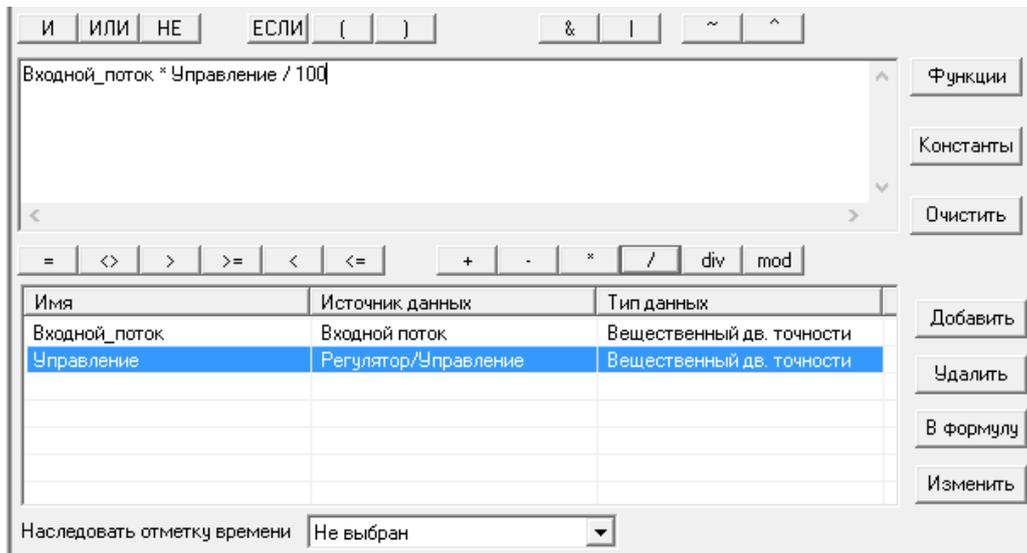


Рисунок 7.10 – Формула для расчета входного потока

Внесем коррективы в «Расчет уровня». Удалим параметр «Входной поток» и добавим «Расчет входного потока». Так как уровень жидкости не может быть отрицательным, зададим параметр расчета функцией «ЕСЛИ» (рисунок 7.11).

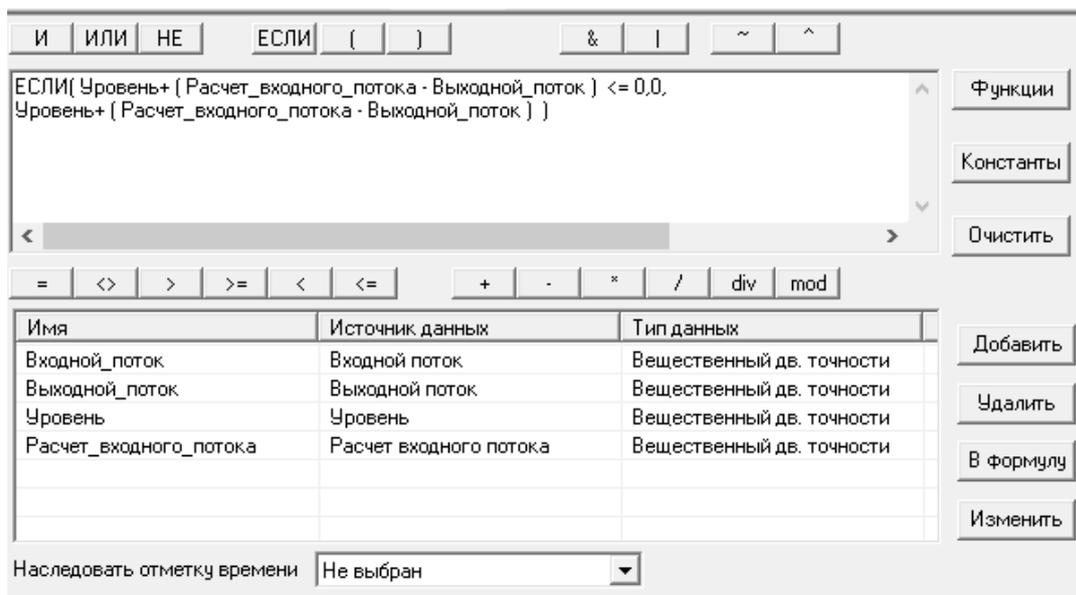


Рисунок 7.11 – Новая формула для расчета уровня

Переходим во вкладку редактирования мнемосхемы объекта «Аппарат». Перетаскиваем «Регулятор» и «Расчет входного потока» на мнемосхему, как показано на рисунке 7.12.

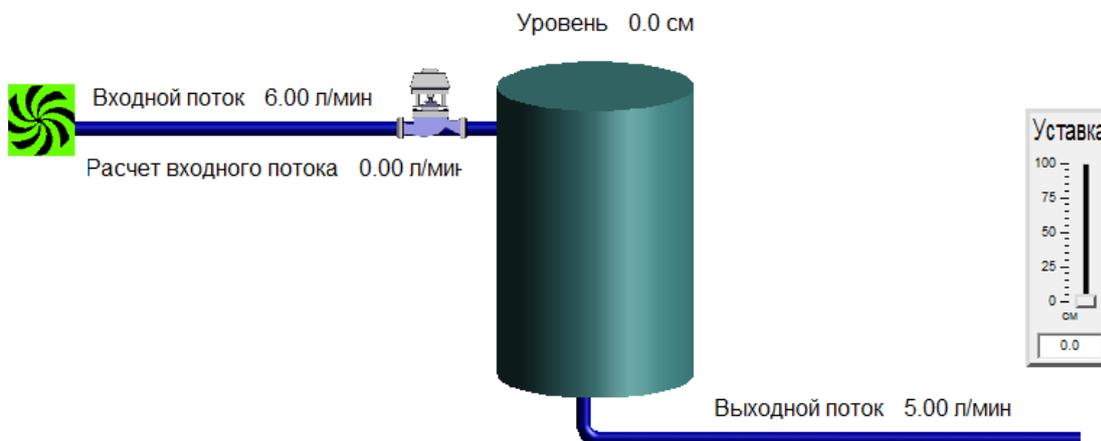


Рисунок 7.12 – Редактирование мнемосхемы

Размещаем на мнемосхему тренд из вкладки «Окна» на палитре мнемосхемы. Связываем значение «Уровень» с трендом. Сохраняем и запускаем проект. Изменяя параметр установки, можно видеть, как увеличивается или уменьшается уровень, как изменяется входной поток для поддержания выбранной установки. Выходим из режима исполнения.

Если при задании начальных параметров «Выходной поток» будет больше, чем «Входной поток», «Уровень» все равно не опустится ниже 0, его нижняя граница будет рассчитана по формуле

$$\text{Уровень} = \text{Установка} - (\text{Выходной поток} - \text{Входной поток}).$$

Для привлечения внимания при аварии используют сигнальные лампы (аварийные лампы). Создадим объект с именем «Аварийная лампа». Добавим «Индикатор состояния» из библиотеки «Датчики». Перейдем на вкладку «Цвета» и зададим на «Вход 0» цвет зеленый, на «Вход 1» – красный. Так как входы 0 и 1 имеют логический тип данных, необходимо добавить элементы сравнения. Для этого из библиотеки «Вычисления» добавляем « $A > B$ » для «Индикатора состояния». В качестве входного «Значения A » возьмем значение «Уровень», связывая их перетаскиванием. В качестве «Значения B » возьмем константу. Перейдем во вкладку «Опрос» и введем значение, равное 70. Если «Значение A » будет больше 70, то должен загореться красный индикатор. Для этого выходной параметр «Результат» перетаскиваем на вход «Вход 1» элемента «Индикатор состояния». Добавим « $A < B$ » из библиотеки «Вычисления» для элемента «Индикатор состояния». В качестве входного «Значения A » возьмем значение «Уровень», связывая их перетаскиванием. В качестве «Значения B » возьмем константу, перейдем во вкладку «Опрос» и введем значение, равное 70. Если «Значение A » будет меньше

70, то должен загореться зеленый индикатор. Для этого выходной параметр «Результат» перетаскиваем на вход «Вход 0» элемента «Индикатор состояния». Открываем мнемосхему объекта «Аппарат» и добавляем «Индикатор состояния». Запускаем режим исполнения. Создание проекта завершено.

Задание 2. Создать проект по контролю наполнения бака жидкостью.

Порядок выполнения задания 2

Для начала создадим шкалы, необходимые для более наглядного отображения параметров. Выделим узел «Система» в дереве системы и перейдем на вкладку «Шкалы» и нажмем «Добавить» (рисунок 7.13). Назовем шкалу «Уровень 5000». Эта шкала предназначена для отображения уровня жидкости в баках. Начало зададим 0, конец – 5000. Столбец «Формат» показывает формат отображения значений с плавающей точкой. Цифра, стоящая после символа «f», показывает количество знаков после запятой. Зададим формат «f0», единицы измерения «Литр». Удалим предупредительные границы.

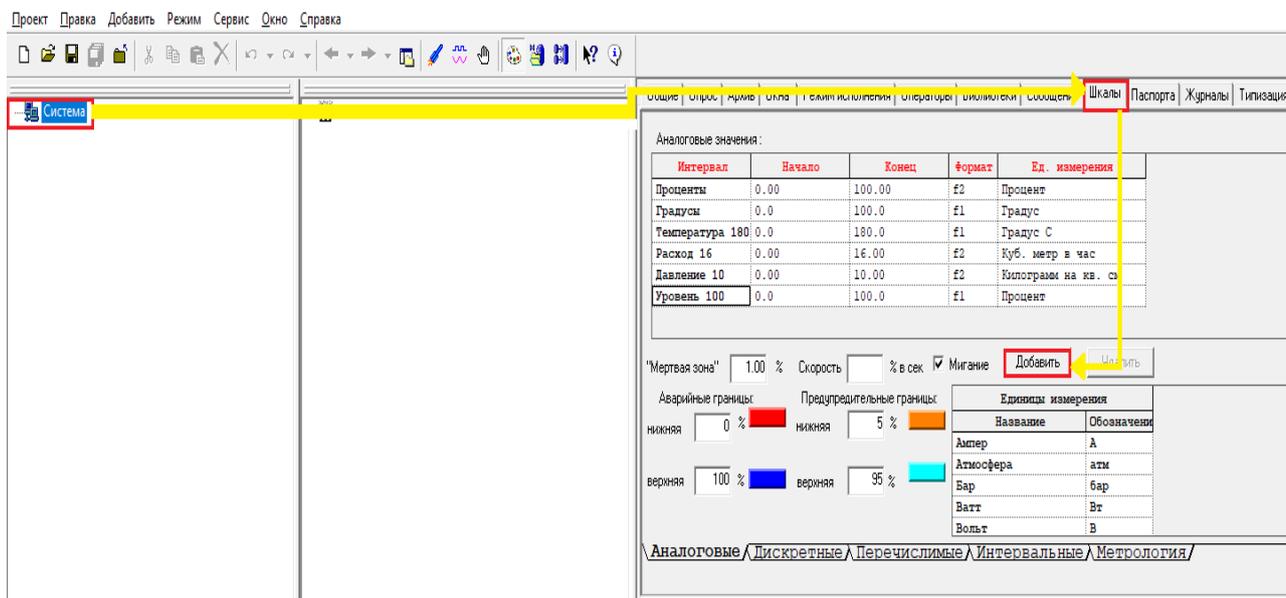


Рисунок 7.13 – Добавление шкалы

Следующая шкала будет предназначена для отображения уровня в емкости для перемешивания. Создавать ее нужно на основании вашего варианта. Допустим, в варианте указан уровень 100, тогда назовем ее «Уровень 1000». Начало зададим 0, конец – 1000. Формат и единицы измерения такие же.

Создадим еще одну шкалу. Назовем ее «Поток». Она будет служить для отображения количества выкачиваемой жидкости. Диапазон шкалы от 0 до 10. Формат «f1». Добавим новую единицу измерения, так как имеющиеся не подходят для выполнения задания. Для этого необходимо щелкнуть в любое место в поле «Единицы измерения» и нажать «Добавить». Назовем единицу измерения «Литр в минуту», обозначение л/мин. Выберем ее для нашей шкалы.

Перейдем к созданию дерева объектов. Щелчком правой кнопки мыши по узлу «Система» вызовем контекстное меню, выберем «Вставить» → «Компьютер». Далее в другом столбце аналогичным образом добавим «Объект», назовем его «Участок» и назначим исполнение в добавленном ранее компьютере (рисунок 7.14).

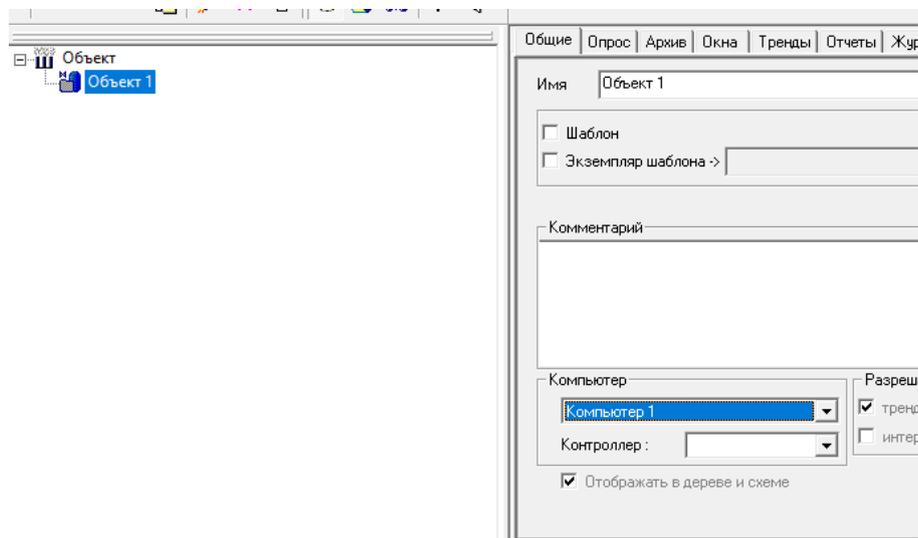


Рисунок 7.14 – Назначение исполняющего устройства

Через контекстное меню добавим в «Участок» новый объект и назовем его «Бак красный». Далее из палитры  добавим аппарат «Емкость», как показано на рисунке 7.15.

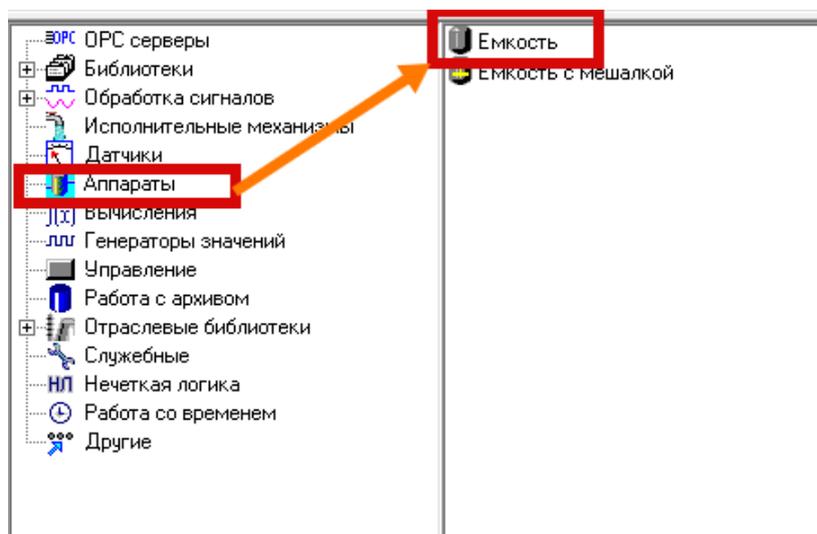


Рисунок 7.15 – Добавление аппарата «Емкость»

Добавленный модуль представляет собой визуально функциональный блок, который может отображаться на мнемосхеме. Выделим блок «Емкость» и перейдем на вкладку «Вид». Выберем тип отображения «Цистерна».

Добавим через контекстное меню в объект «Бак красный» значение и назовем его «Уровень красный». Аналогичным образом добавляем две команды «Поток красной» и «Нужно красной», а также три расчета – «Расчет уровня», «Кол-во красной» и «Управление потоком». Теперь перейдем в команду «Поток красной», откроем вкладку «Опрос выхода» и установим галочку напротив «Значение до опроса» и нажимаем «Принять». Аналогичную операцию проведем с командой «Нужно красной».

Перейдем к значению «Уровень красный». На вкладке «Общее» выберем диапазон измерения «Уровень 5000». Таким же образом у расчета «Расчет уровня», у команды «Поток красной» выберем «Поток», у команды «Нужно красной» и расчета «Кол-во красной» – «Уровень 1000».

Далее нажмем на расчет «Расчет уровня», перейдем на вкладку «Формула» и удерживая левую кнопку мыши перенесем «Уровень красный» и «Управление потоком» в область значений (рисунок 7.16).

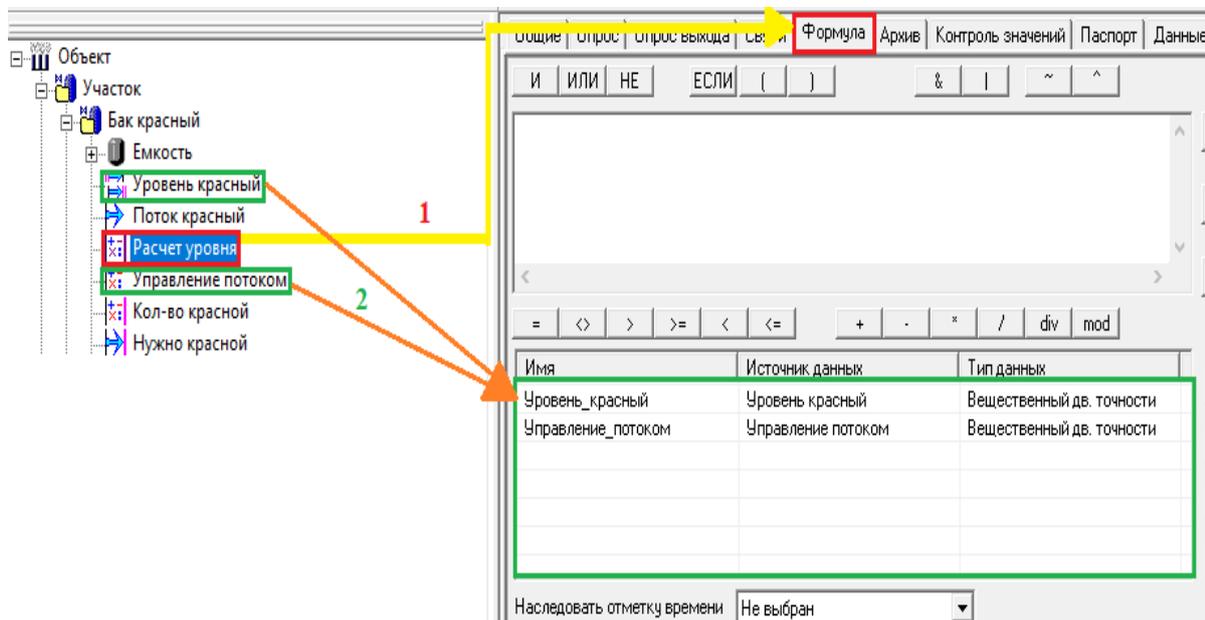


Рисунок 7.16 – Добавление значений в формулу

Двойным щелчком правой кнопкой мыши добавляем значение в рабочую область и составляем формулу «Уровень красный» – «Управление потоком».

Создадим связь между «Расчет уровня» и «Уровень красный». Удерживая «Расчет уровня», переносим его на «Уровень красный». Аналогичным образом создадим связь между входным параметром аппарата «Емкость» и «Уровень красный» – перетаскиваем «Уровень красный» на «Уровень». У значения «Уровень красный» на вкладке «Опрос» установим галочку напротив «Значение до опроса» и укажем значение 5000. Перейдем к расчету «Кол-во красной» и составим формулу «Кол-во красной» + «Управление потоком». На вкладке «Опрос выхода» укажем «значение до опроса», равное 0. Далее переходим к расчету «Управление потоком» и составляем формулу, как показано на рисунке 7.17.

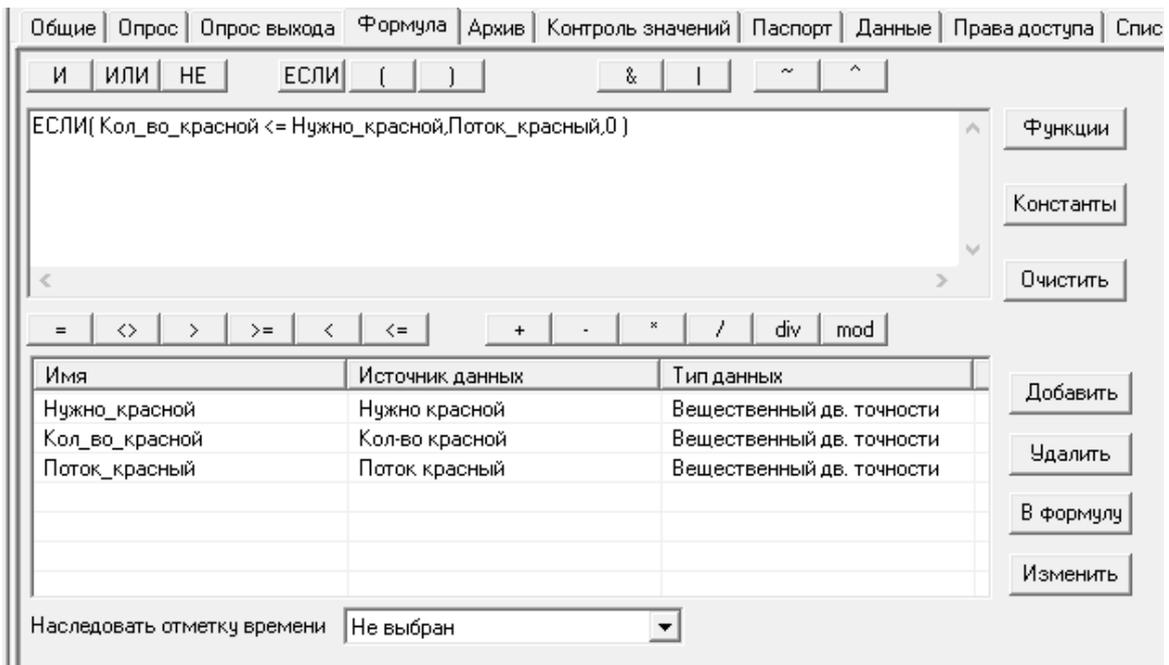


Рисунок 7.17 – Управление потоком

Эта формула позволяет автоматически ограничить поток жидкости при достижении необходимого количества.

Следующим шагом добавим насос и элементы для управления им. Переходи на палитру и выбираем «Исполнительные механизмы» → «Насос мнемонический» и добавляем в объект «Бак красный» (рисунок 7.18).

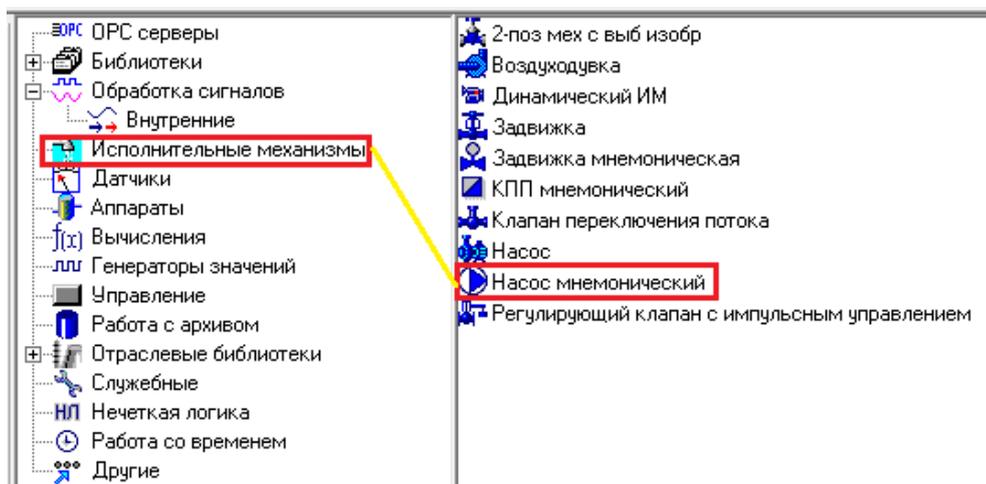


Рисунок 7.18 – Добавление насоса

Также из палитры добавим элементы вычислений « $A > B$ » и « $A = B$ », как показано на рисунке 7.19.

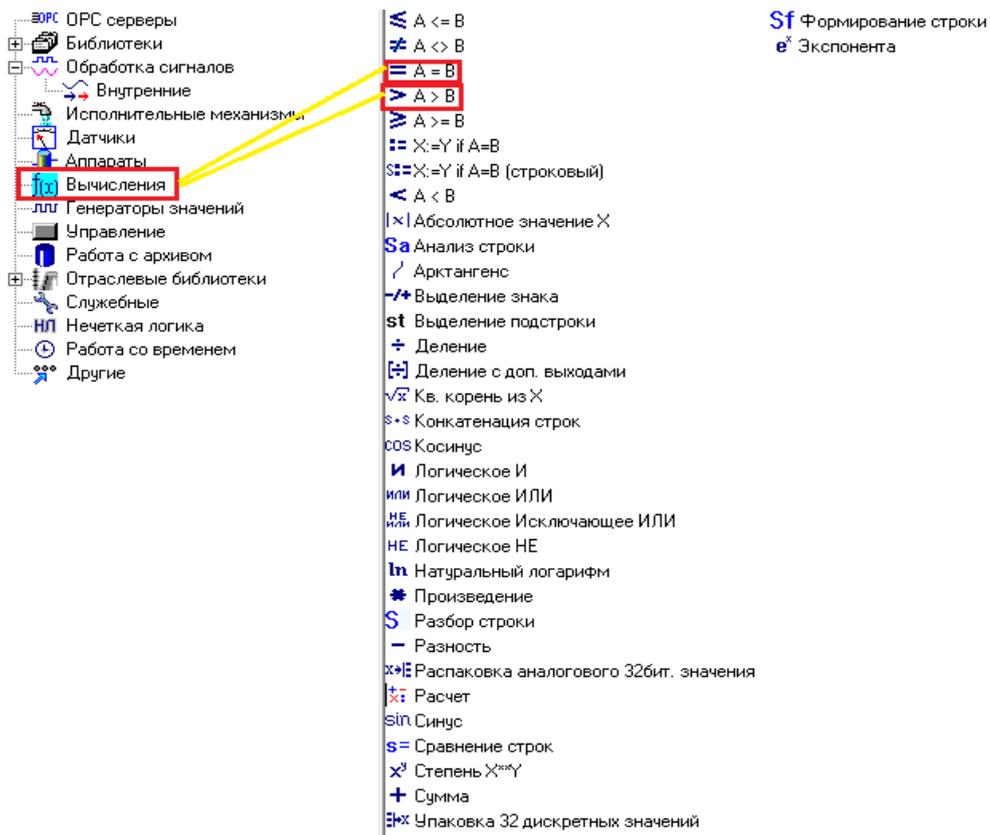


Рисунок 7.19 – Добавление вычислений

Элемент « $A > B$ » назовем «Открыт», а « $A = B$ » – «Закрыт». Перейдем к «Открыт». Перетащим «Нужно красной» во вход «Значение A », «Кол-во красной» во вход «Значение B », а «Результат» перетащим во вход насоса «Включен» (рисунок 7.20).

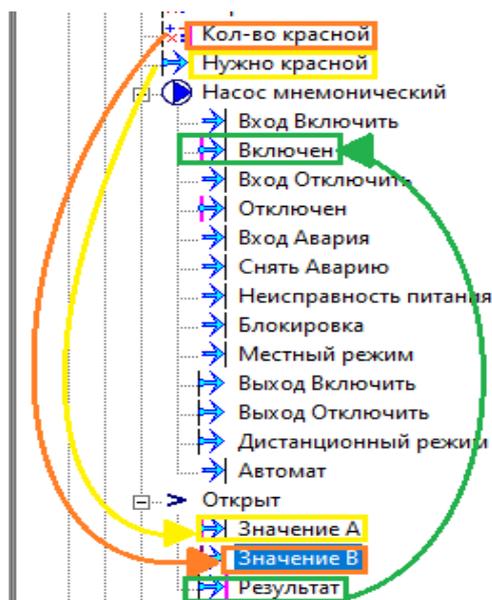


Рисунок 7.20 – Составление связей

Аналогично составляем связи с элементом «Закрыт», однако выход «Результат» связываем с входом насоса «Отключен».

Так как необходимо иметь три одинаковых бака, дублируем уже имеющийся два раза. Для этого щелчком правой кнопкой мыши по «Бак красный» вызываем контекстное меню и нажимаем «Дублировать» и выбираем количество копий – две. Появившиеся баки переименовываем в «Бак желтый» и «Бак зеленый». Также необходимо переименовать внутренние элементы, чтобы их было удобно различать. После всех манипуляций должно получиться дерево объектов, представленное на рисунке 7.21.

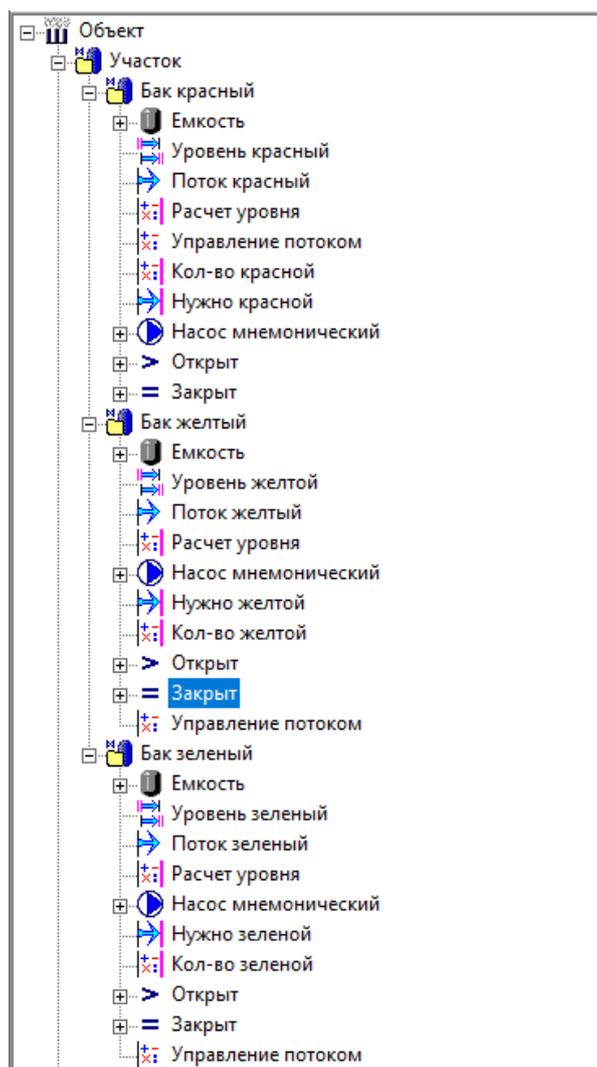


Рисунок 7.21 – Дерево объектов

Теперь необходимо создать аппарат для смешивания и клапаны. Для начала создадим новый объект в объекте «Участок» и назовем его «Оборудование». Далее переходим на палитру, выбираем «Аппараты» → «Емкость с мешалкой» и добавляем в «Оборудование». Также из вкладки палитры «Исполнительные механизмы» добавим элемент «Задвижка мнемоническая», назовем ее «Кла-

пан 1» и дублируем два раза. Еще раз переходим в палитру, находим там «Управление» и добавляем оттуда элемент «Кнопка» в объект «Оборудование» для управления мешалкой. Через контекстное меню объекта «Оборудование» добавим значение и расчет и назовем их «Уровень мешалки» и «Расчет уровня мешалки» соответственно. Основные элементы «Оборудование» добавлены, перейдем к их настройке и составлению связей.

Переходим к «Расчет уровня мешалки» и на вкладке «Общее» выбираем диапазон измерений «Уровень 1000». Далее переходим на вкладку «Формула», перетаскиваем туда «Уровень мешалки» и «Управление потоком» из каждого бака и составляем формулу (рисунок 7.22).

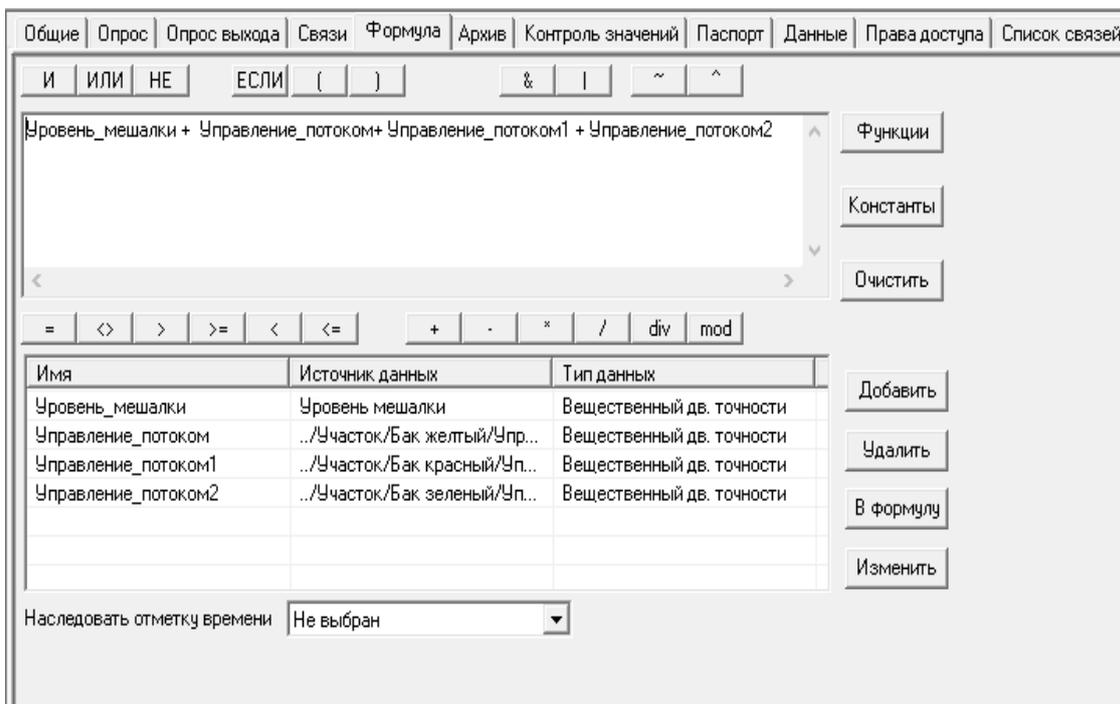


Рисунок 7.22 – Формула расчета уровня мешалки

Далее переносим «Расчет уровня мешалки» в значение «Уровень мешалки», тем самым связывая их. В свою очередь, «Уровень мешалки» переносим на входной параметр «Уровень» объекта «Емкость с мешалкой».

Теперь перейдем во вкладку «Опрос» элемента «Уровень мешалки» и установим галочку возле показателя «Значение до опроса» и укажем его равным «0».

Для упрощения работы клапанов свяжем их такими же параметрами, что и насосы. Для этого выходы «Результат» элементов «Открыт» и «Закрыт» объекта «Бак красный» связываем с входами «Открыт» и «Закрыт» элемента «Клапан 1» соответственно. Повторяем операцию с «Бак желтый» и «Клапан 2», а также «Бак зеленый» и «Клапан 3».

Для управления мешалкой свяжем выход «Состояние» элемента «Кнопка» с входом «Мешалка» элемента «Емкость с мешалкой», как показано на рисунке 7.23.

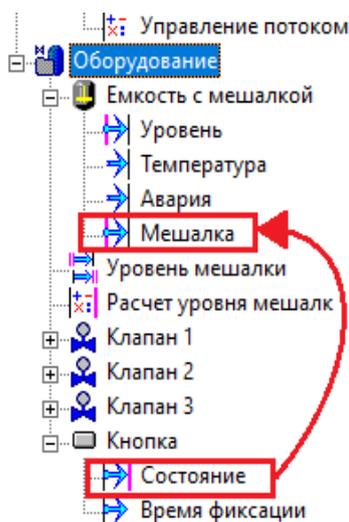


Рисунок 7.23 – Создание связи для управления мешалкой

Теперь перейдем к созданию мнемосхемы (рисунок 7.24).

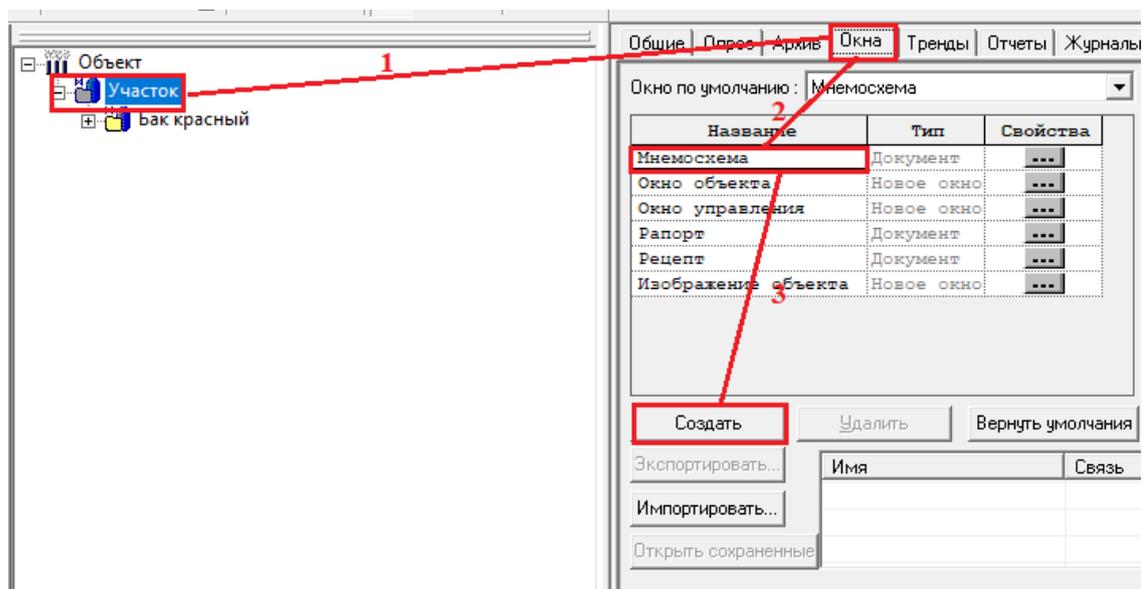


Рисунок 7.24 – Создание мнемосхемы

Перетащим элементы «Емкость» из объектов «Бак красный», «Бак желтый» и «Бак зеленый» на мнемосхему (рисунок 7.25).

Аналогичным образом перетаскиваем насосы, клапаны и емкость с мешалкой на мнемосхему. Выделяем насосы и клапаны и в панели свойств, в правом нижнем углу, выбираем направление «Вертикально» (рисунок 7.26).

Далее добавим элементы для задания параметров. Для этого, удерживая правую кнопку мыши, по очереди перетаскиваем «Поток красный», «Поток желтый», «Поток зеленый», «Нужно красной», «Нужно желтой», «Нужно зеленой» и в появившемся контекстном меню выбираем «Слайдер». Располагаем их в удобное место.

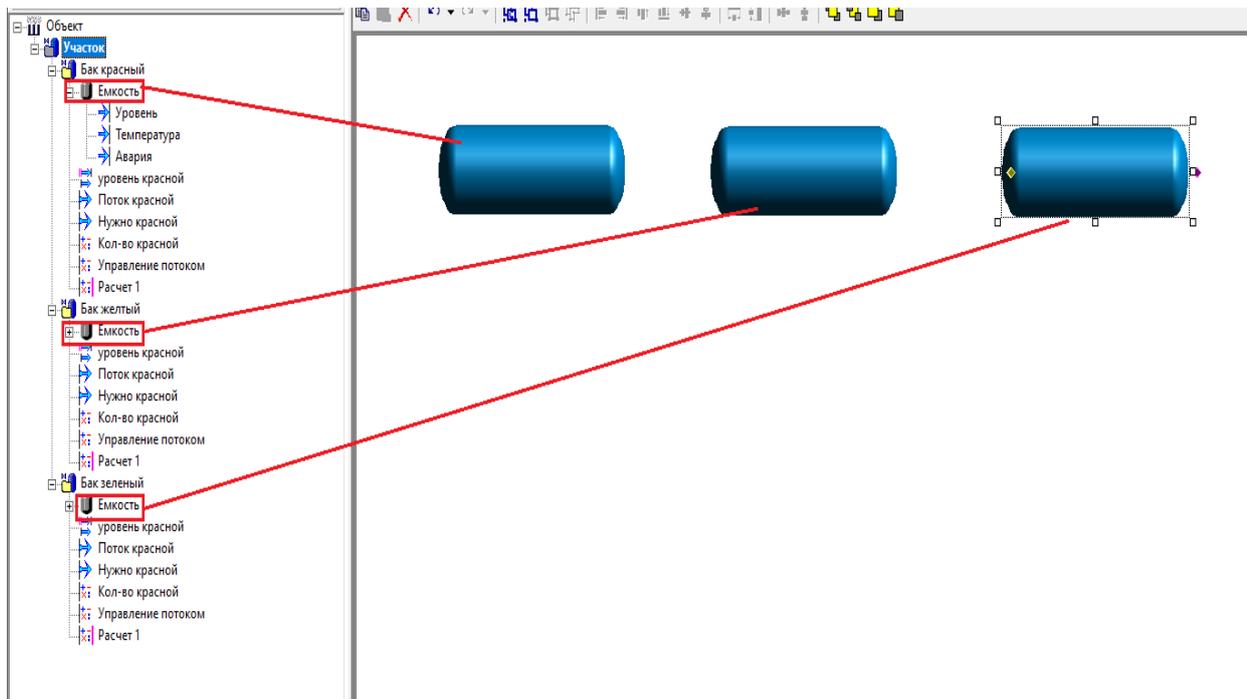


Рисунок 7.25 – Емкости на мнемосхеме

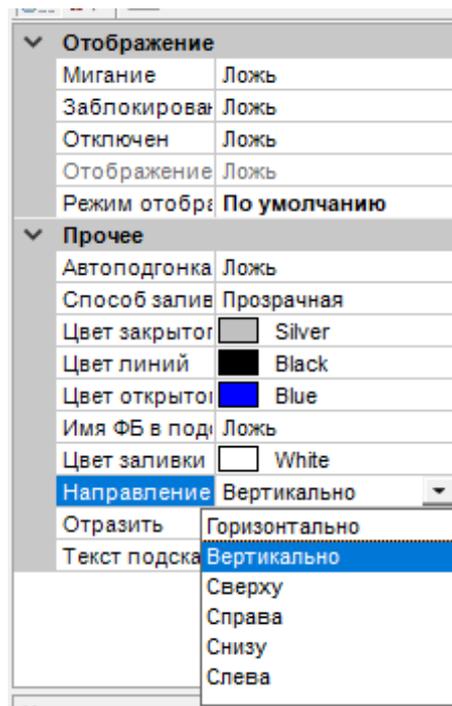


Рисунок 7.26 – Выбор направления

Также необходимо добавить элементы контроля параметров уровней жидкостей. Для этого, удерживая левую клавишу мыши, перетаскиваем значения «Уровень красный», «Уровень желтый», «Уровень зеленый» на мнемосхему и располагаем над соответствующими баками. Удерживая правую клавишу мыши, перенесем на мнемосхему «Уровень мешалки», выберем «Индикатор», вызовем

контекстное меню, нажатием правой клавиши по индикатору перейдем в «Свойства» → «Вид» и укажем заголовок «Уровень».

Добавим элементы контроля количества прошедшей через клапаны жидкости. Для этого перенесем расчеты «Кол-во красной», «Кол-во желтой» и «Кол-во зеленой» на мнемосхему. Далее перейдем в «Свойства». На вкладке «Вид» выберем тип штриховки/отображения «Заполнение цветом». На вкладке «Граница» укажем толщину 1 и вид «Плоская». На вкладке «Цвет» выберем цвет фона белый. На вкладке «Шрифт» укажем размер 10. Расположим элементы рядом с клапанами.

Элемент «Кнопка» перетащим аналогичным образом на мнемосхему и расположим в удобном месте.

Для наглядности работы проведем трубопровод от баков до аппарата смешивания. Для этого в правом нижнем углу выберем «Палитра» → «Объемные элементы» → «Трубопровод» (рисунок 7.27).

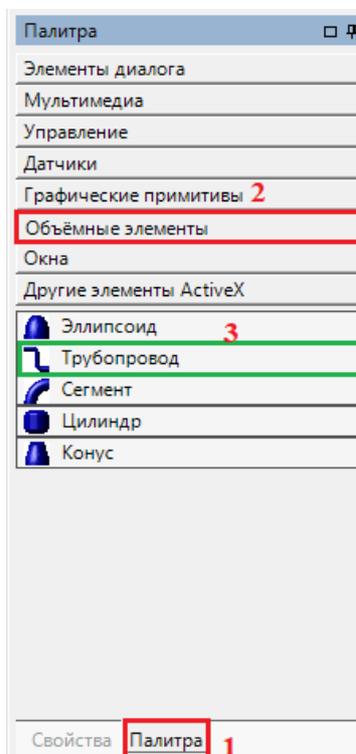


Рисунок 7.27 – Добавление трубопровода

Добавляем трубопровод на мнемосхему и проводим от баков до аппарата смешивания. Двойной щелчок левой кнопкой мыши при выбранном трубопроводе создает искривление трубы в направлении курсора мыши.

Полученная мнемосхема может иметь вид, который представлен на рисунке 7.28.

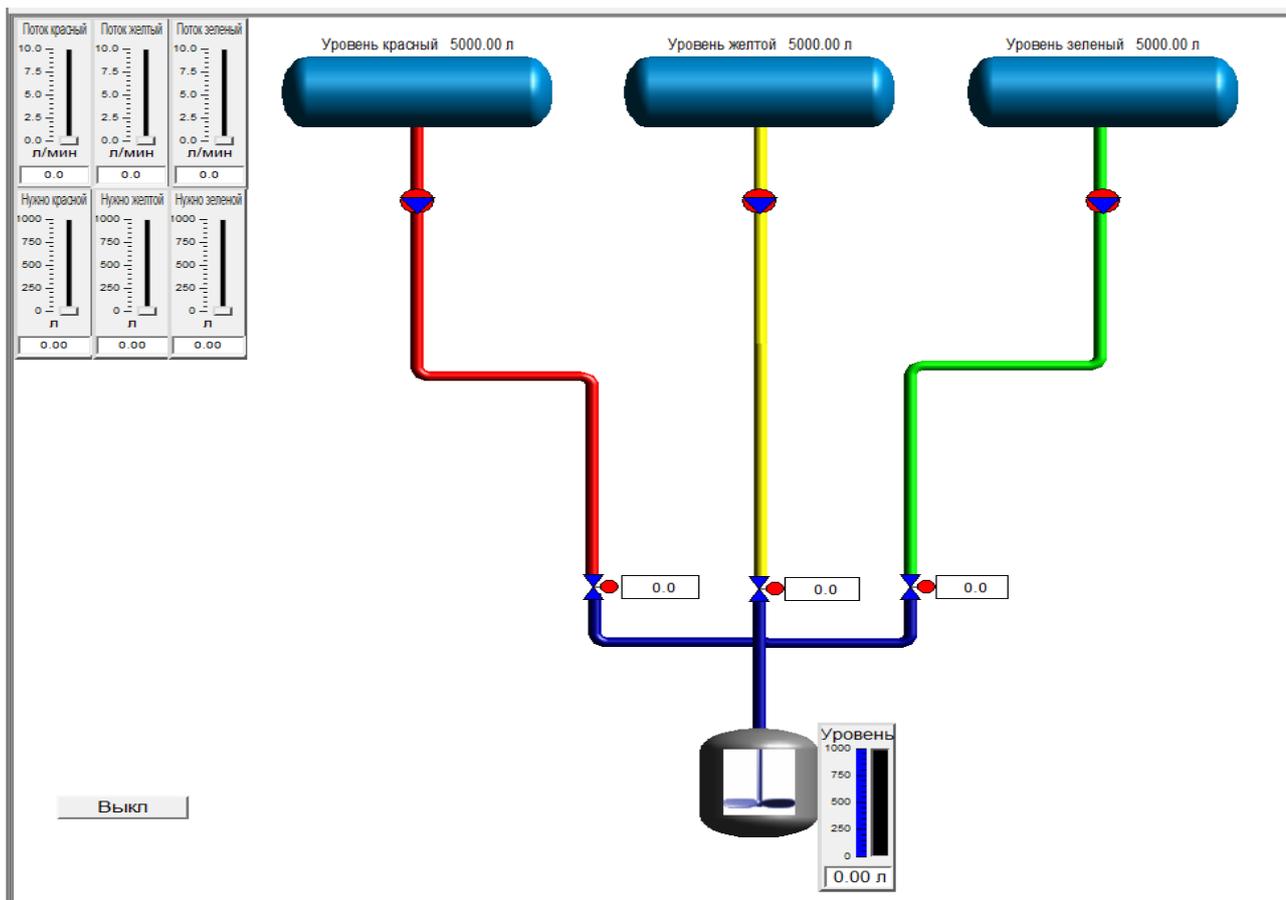


Рисунок 7.28 – Готовая мнемосхема для проекта по контролю наполнения бака жидкостью

Нажатием на «Показать проект»  закроем окно мнемосхемы и вернемся в рабочую область. Чтобы при запуске проекта сразу отрывалась мнемосхема, необходимо перейти в объект «Участок» и на вкладке «Окна» поставить галочку напротив «Сделать мнемосхему стартовой».

7.3 Индивидуальные задания

В рамках лабораторной работы предусмотрено выполнение заданий в соответствии с индивидуальным номером зачетной книжки каждого студента. При этом важно учесть, что повторение заданий не допускается, обеспечивая таким образом самостоятельное выполнение своей работы. В случае совпадения номера зачетной книжки с уже занятым заданием студенту предоставляется возможность выбрать свободный номер из предложенных вариантов.

Варианты для задания 1 представлены в таблице 7.1, для задания 2 – в таблице 7.2.

Таблица 7.1 – Варианты индивидуального задания 1

Вариант	Входной поток	Выходной поток
1	8	6
2	8	2
3	11	6
4	9	5
5	11	1
6	8	2
7	9	2
8	6	2
9	6	5
10	10	4
11	8	6
12	9	5
13	10	4
14	6	2
15	10	5
16	8	3
17	7	2
18	8	3
19	10	1
20	7	6
21	11	5
22	10	1
23	7	3
24	7	4
25	11	4
26	6	5
27	9	6
28	10	6
29	6	4
30	11	1

Таблица 7.2 – Варианты индивидуального задания 2

Вариант	Количество красной	Количество желтой	Количество зеленой	Уровень мешалки
1	2	3	4	5
1	330	330	340	1000
2	270	500	230	1000
3	410	260	330	1000
4	390	150	460	1000
5	220	140	340	700
6	380	210	110	700
7	180	340	180	700
8	250	190	260	700
9	500	500	500	1500

Продолжение таблицы 7.2

1	2	3	4	5
10	400	800	300	1500
11	620	290	590	1500
12	350	540	610	1500
13	900	540	860	2300
14	670	930	700	2300
15	500	400	1400	2300
16	720	880	700	2300
17	950	430	1320	2700
18	720	890	1090	2700
19	1000	320	1380	2700
20	640	1300	760	2700
21	1000	1000	1000	3000
22	1700	340	960	3000
23	1050	880	1070	3000
24	440	560	2000	3000
25	570	340	790	1700
26	280	760	660	1700
27	380	380	940	1700
28	600	800	300	1700
29	300	400	200	900
30	300	300	250	900

7.4 Содержание отчета

В рамках лабораторной работы № 7 необходимо выполнить два индивидуальных задания. В связи с этим отчет по лабораторной работе должен содержать:

- 1) титульный лист;
- 2) введение:
 - 2.1) цель выполнения задания;
 - 2.2) краткое описание задания;
- 3) основная часть:
 - 3.1) ход выполнения задания;
 - 3.2) обработка и анализ результатов выполнения задания;
- 4) выводы по результатам выполнения задания.

7.5 Контрольные вопросы

- 1 Каким образом контролируется уровень жидкости в баках?
- 2 Приведите формулы расчета потока.
- 3 Где применяются SCADA-системы?
- 4 Какое основное предназначение элемента «Емкость»?
- 5 Какое функциональное назначение элемента «Насос мнемонический»?

6 Какую роль играет *MasterSCADA* в автоматизированных системах управления?

7 Возможно ли реальное применение данного проекта? Где?

8 Каким образом контролируется уровень жидкости в баках?

9 Как осуществляется смешивание жидкостей?

Вспомогательная литература

1 Пьявченко, Т. А. Проектирование АСУТП в SCADA-системе / Т. А. Пьявченко, Н. А. Куцевич, О. В. Синенко. – Таганрог : ТРТУ, 2007. – 78 с.

2 Барашко, О. Г. Автоматика, автоматизация и автоматизированные системы управления / О. Г. Барашко, Н. А. Куцевич, О. В. Синенко. – Минск : БГТУ, 2011. – 327 с.

3 Медведев, А. Е. Автоматизация производственных процессов / А. Е. Медведев, А. В. Чупин, О. В. Синенко. – Кемерово : КГТУ, 2009. – 345 с.

4 Скороспешкин, В. Н. Автоматизированные информационно-управляющие системы / В. Н. Скороспешкин, А. В. Чупин, О. В. Синенко. – Томск : ТПУ, 2009. – 57 с.

5 ИнСАТ «Master SCADA. Руководство пользователя. Версия 3.X.». – Москва, 2017. – 574 с.

6 ИнСАТ «Master SCADA. Основы проектирования: Метод. пособие». – Москва, 2016. – 277 с.

7 Master SCADA: учеб. пособие для студентов специальности 5311000 – Автоматизация и управление технологических процессов (в водном хозяйстве) / Р. Т. Газиева [и др.]. – Ташкент, 2020. – 105 с.

Лабораторная работа № 8
**Создание проекта по контролю смешивания жидкостей
разной температуры в *MasterSCADA***

Цель работы: создать проект по смешиванию жидкостей различной температуры, изначально находящихся в двух баках, получившуюся смесь необходимо откачать насосом из емкости.

Основные задачи:

- 1 Разработать систему мониторинга и диагностики оборудования, которая будет отслеживать состояние оборудования и своевременно выявлять неисправности.
- 2 Разработать систему мониторинга и контроля качества смешивания жидкости в емкости.
- 3 Получить практические навыки работы с инструментами *MasterSCADA*.
- 4 Закрепить полученные навыки и умения путем выполнения задания.

8.1 Теоретические сведения

В сфере автоматизации и мониторинга промышленных процессов *MasterSCADA* предоставляет широкие возможности. Создание проекта для смешивания жидкостей разной температуры в *MasterSCADA* позволяет эффективно управлять этим процессом, обеспечивая точность и безопасность.

В основе такого проекта лежат следующие компоненты:

- 1 Датчики температуры, которые размещаются в различных баках для измерения температуры каждой жидкости.
- 2 Клапаны и насосы, которые применяются для управления потоком жидкостей и обеспечения их смешивания.
- 3 Программирование логики, которое учитывает текущие температуры жидкостей и заданные параметры смешивания.

Термодинамические аспекты играют ключевую роль:

- 1 Смешивание жидкостей с различной температурой связано с передачей тепла, и уравнение теплового баланса имеет важное значение.
- 2 Система должна регулировать температуру смеси, учитывая изменения входных параметров.

Программирование в *MasterSCADA* включает в себя:

- 1 Интеграцию датчиков температуры и управляющих устройств в *MasterSCADA*.
- 2 Разработку логики управления, учитывающую температурные данные и параметры смешивания.
- 3 Создание пользовательского интерфейса для мониторинга и управления процессом смешивания.

Важными аспектами такого проекта является разработка системы безопасности, предотвращающей аварийные ситуации.

Создание проекта для смешивания жидкостей разной температуры в *MasterSCADA* требует комплексного подхода, включая термодинамические расчеты, логику управления и обеспечение безопасности. Это обеспечивает эффективное управление процессом и высокую точность в промышленных условиях.

Проект состоит из двух емкостей с жидкостью различной температуры, аппарата для смешивания, а также вспомогательного оборудования.

Аппарат для смешивания представляет собой емкость с мешалкой, в которую через клапаны заливается часть жидкости из баков в заданной пропорции. После заполнения емкости субстанцией запускается перемешивание. Завершив смешивание, смесь необходимо откачать насосом из емкости с мешалкой.

От *SCADA*-системы требуется отображать состояние параметров:

- уровень в баках;
- уровень в емкости;
- температуру смеси;
- количество жидкости, прошедшей через каждый из клапанов.

Параметры, задаваемые оператором:

- количество каждой жидкости, необходимое для смешивания;
- количество жидкости, выкачиваемое из бака, за единицу времени.

8.2 Задание для выполнения лабораторной работы

Задание. Создайте проект по смешиванию жидкостей различной температуры, изначально находящихся в двух баках. После смешивания смесь необходимо откачать насосом из емкости. Учтите отображение состояния уровня в баках, в емкости и температуру смеси.

Порядок выполнения задания

Для начала создадим шкалы, необходимые для более наглядного отображения параметров. Выделим узел «Система» в дереве системы и перейдем на вкладку «Шкалы» и нажмем «Добавить», как показано на рисунке 8.1. Назовем шкалу «Уровень 2000». Эта шкала предназначена для отображения уровня жидкости в баках. Начало зададим 0, конец – 2000. Столбец «Формат» показывает формат отображения значений с плавающей точкой. Цифра, стоящая после символа «f», показывает количество знаков после запятой. Зададим формат «f0», единицы измерения «Литр». Удалим предупредительные границы.

Следующую шкалу назовем «Уровень 1000». Она будет предназначена для отображения уровня в емкости для перемешивания. Начало зададим 0, конец – 1000. Формат и единицы измерения такие же.

Создадим еще одну шкалу. Назовем ее «Поток». Она будет служить для отображения количества выкачиваемой жидкости. Диапазон шкалы 0...10. Формат «f1». Добавим новую единицу измерения, так как имеющиеся не подходят. Для этого необходимо щелкнуть в любое место в поле «Единицы измерения» и

нажать «Добавить». Назовем единицу измерения «Литр в минуту», обозначение л/мин. Выберем ее для нашей шкалы.

Также необходимо создать шкалу для температур. Назовем ее «Температура», диапазон шкалы от 0 до 100. Формат «f0». Единицу измерения выбираем «°C».

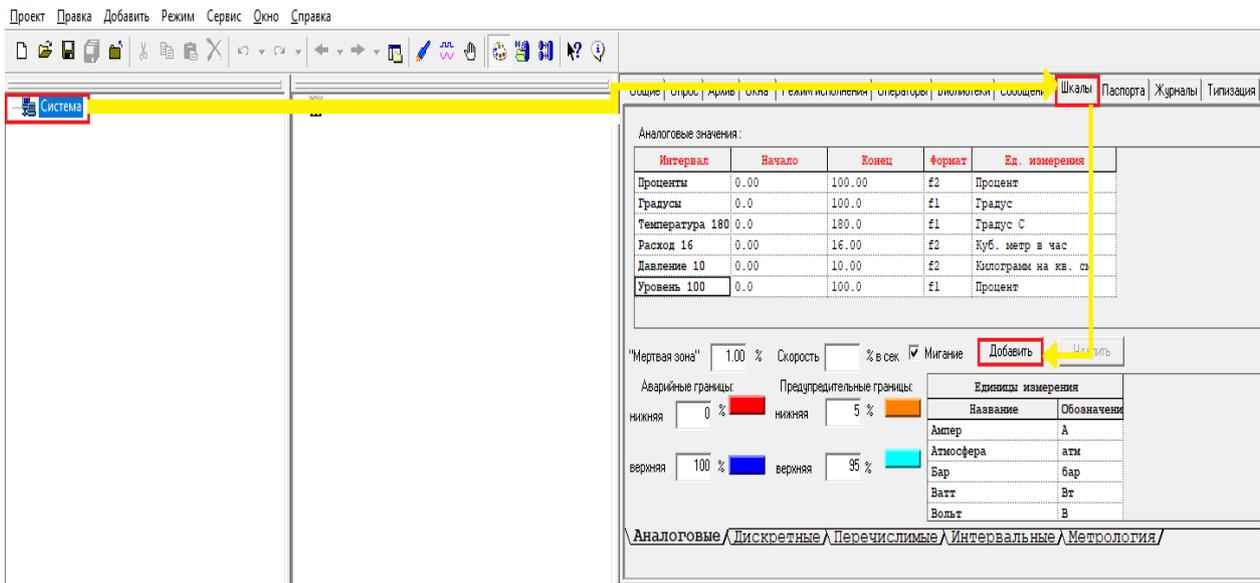


Рисунок 8.1 – Добавление шкалы

Перейдем к созданию дерева объектов. Щелчком правой кнопки мыши по узлу «Система» вызовем контекстное меню, выберем «Вставить» → «Компьютер». Далее в другом столбце аналогичным образом добавим «Объект», назовем его «Участок» и назначим исполнение в добавленном ранее компьютере, как показано на рисунке 8.2.

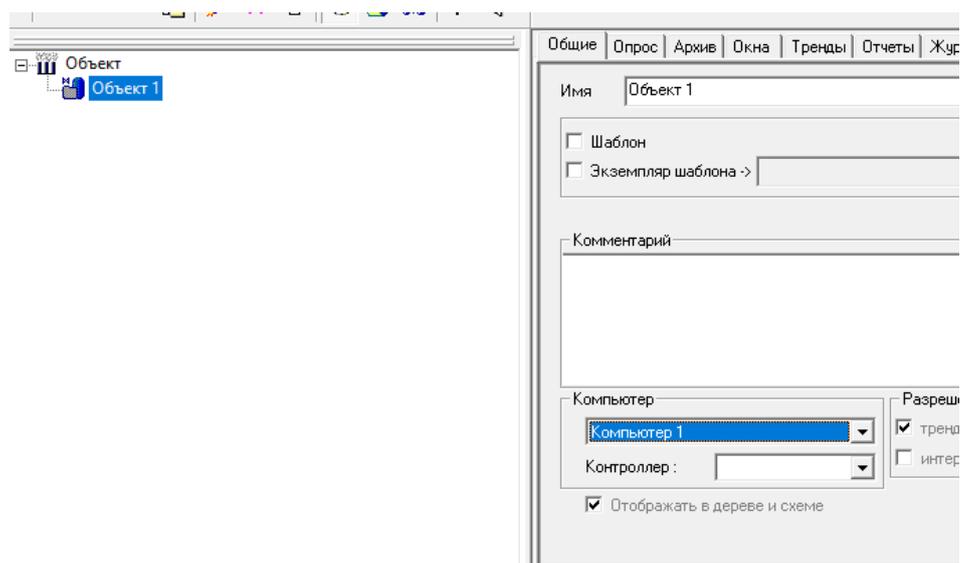


Рисунок 8.2 – Назначение исполняющего устройства

Через контекстное меню добавим в «Участок» новый объект и назовем его «Бак холодный». Далее из палитры  добавим аппарат «Емкость», как показано на рисунке 8.3.

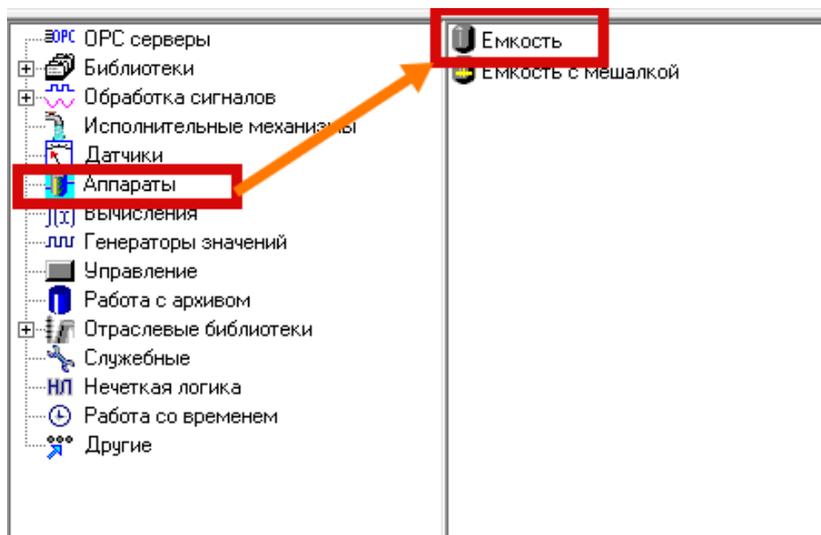


Рисунок 8.3 – Добавление аппарата «Емкость»

Добавленный модуль представляет собой визуально функциональный блок, который может отображаться на мнемосхеме. Выделим блок «Емкость» и перейдем на вкладку «Вид». Выберем тип отображения «Цистерна».

Добавим через контекстное меню в объект «Бак холодный» значение и назовем его «Уровень холодной» (рисунок 8.4).

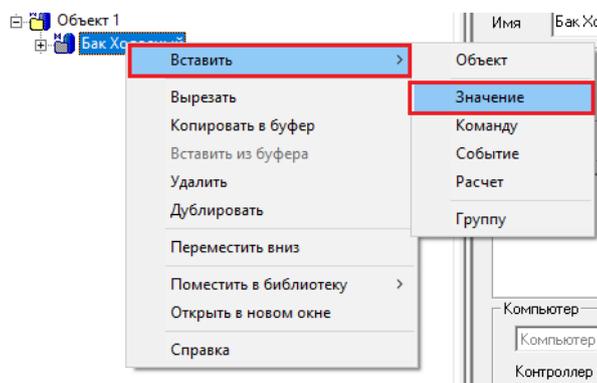


Рисунок 8.4 – Добавление значения в объект

Аналогичным образом добавляем три команды «Поток холодной», «Нужно холодной» и «Температура холодной», а также три расчета – «Расчет уровня», «Кол-во холодной» и «Управление потоком». Перейдем к команде «Поток холодной», откроем вкладку «Опрос выхода» и установим галочку напротив «Значение до опроса» и нажимаем «Принять». Аналогичную операцию проведем с командой «Нужно холодной».

Перейдем к значению «Уровень холодной». На вкладке «Общее» выберем диапазон измерения «Уровень 2000». Таким же образом у расчета «Расчет уровня», у команды «Поток холодной» выберем «Поток», у команды «Нужно холодной» и расчета «Кол-во холодной» – «Уровень 1000». Для команды «Температура холодной» выберем диапазон измерений «Температура», затем перейдем на вкладку «Опрос выхода», установим галочку возле «Значение до опроса» и впишем в поле «1».

Далее нажмем на расчет «Расчет уровня», перейдем на вкладку «Формула» и, удерживая левую кнопку мыши, перенесем «Уровень холодной» и «Управление потоком» в область значений (рисунок 8.5).

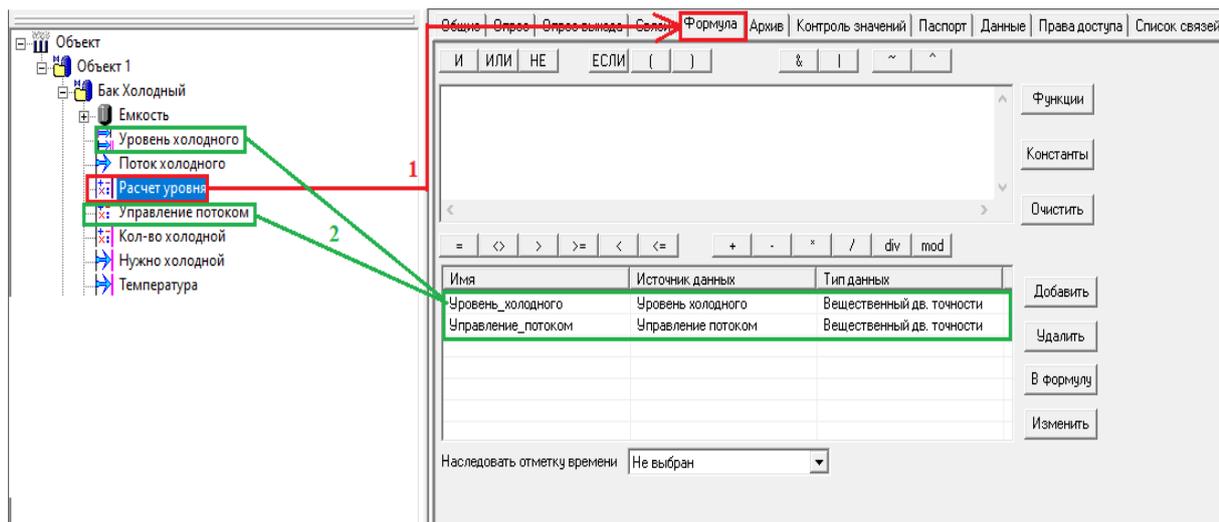


Рисунок 8.5 – Добавление значений в формулу

Двойным щелчком добавляем значение в рабочую область и составляем формулу «Уровень холодной» – «Управление потоком».

Создадим связь между «Расчет уровня» и «Уровень холодной». Удерживая «Расчет уровня», переносим его на «Уровень холодной». Аналогичным образом создадим связь между входным параметром аппарата «Емкость» и «Уровень холодной» – перетаскиваем «Уровень холодной» на «Уровень». У значения «Уровень холодной» на вкладке «Опрос» установим галочку напротив «Значение до опроса» и укажем значение 2000. Перейдем к расчету «Кол-во холодной» и составим формулу «Кол-во холодной» + «Управление потоком». На вкладке «Опрос выхода» укажем «Значение до опроса», равное 0. Далее переходим к расчету «Управление потоком» и составляем формулу, как показано на рисунке 8.6.

Эта формула позволяет автоматически ограничить поток жидкости при достижении необходимого количества.

Следующим шагом добавим насос и элементы для управления им. Переходим на палитру и выбираем «Исполнительные механизмы» → «Насос мнемонический» и добавляем в объект «Бак холодный» (рисунок 8.7).

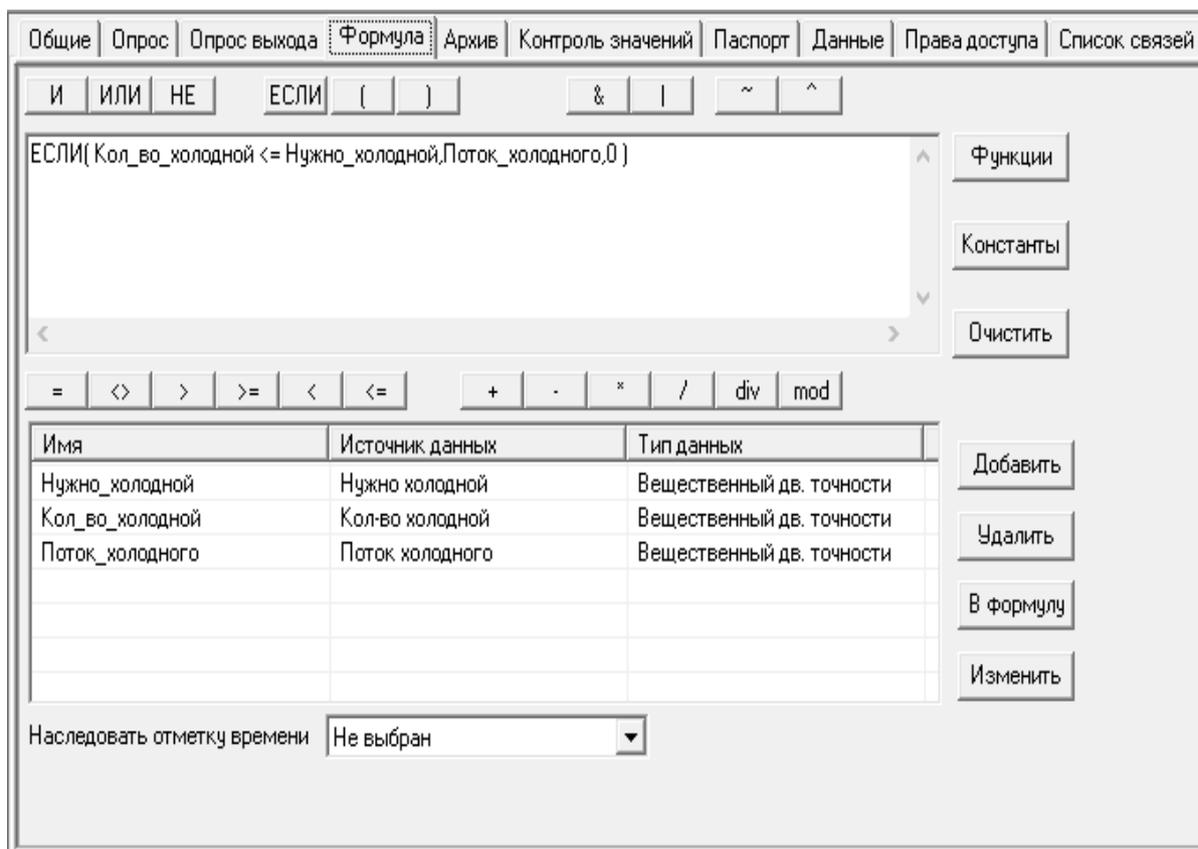


Рисунок 8.6 – Управление потоком

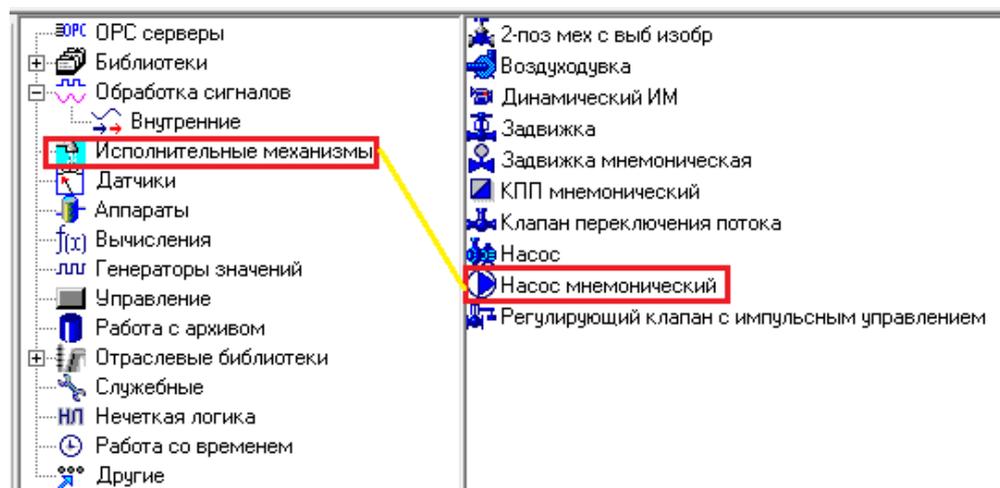


Рисунок 8.7 – Добавление насоса

Также из палитры добавим элемент вычислений « $A > B$ » и « $A = B$ », как показано на рисунке 8.8.

Элемент « $A > B$ » назовем «Открыт», а « $A = B$ » – «Закрыт». Перейдем к «Открыт» и перетащим «Нужно холодной» во вход «Значение A », «Кол-во холодной» во вход «Значение B », а «Результат» перетащим во вход насоса «Включен» (рисунок 8.9).



Рисунок 8.8 – Добавление вычислений

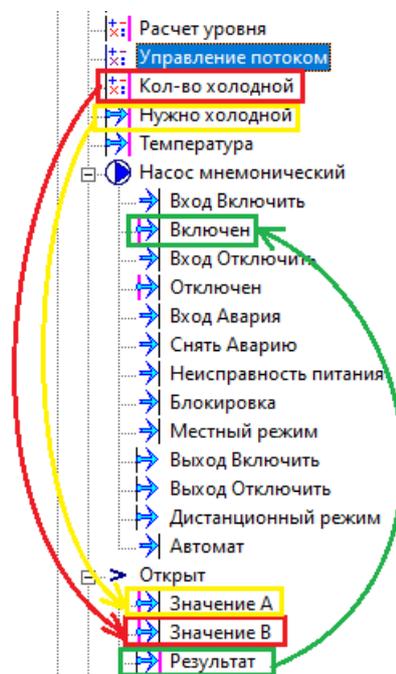


Рисунок 8.9 – Составление связей

Аналогично составляем связи с элементом «Закрыт», однако выход «Результат» связываем с входом насоса «Отключен».

Также необходимо связать команду «Температура холодной» с одноименным параметром объекта «Емкость». Удерживая левую кнопку мыши, перетаскиваем «Температура холодной» на параметр «Температура» объекта «Емкость», тем самым создавая связь между ними.

Так как необходимо иметь два одинаковых бака, дублируем уже имеющийся один раз. Для этого щелчком правой кнопки мыши по «Бак холодный» вызываем контекстное меню и нажимаем «Дублировать» и выбираем количество копий. Появившийся бак переименовываем в «Бак горячий». Также необходимо переименовать внутренние элементы, чтобы их было удобно различать. Затем необходимо перейти во вкладку «Опрос до выхода» элемента «Температура горячей» и ввести в «Значение до опроса» величину «100». После всех манипуляций дерево объектов будет иметь вид, приведенный на рисунке 8.10.

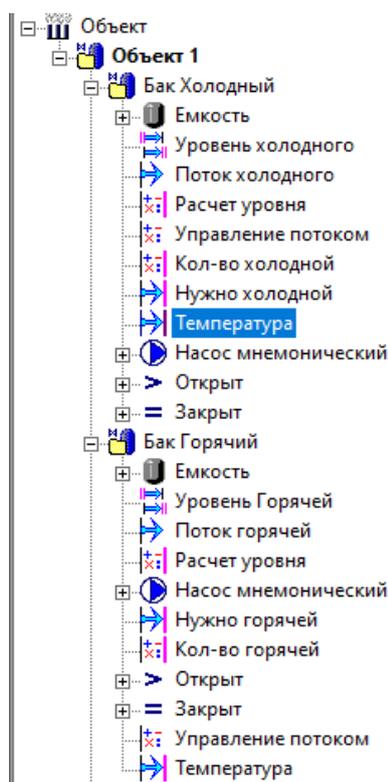


Рисунок 8.10 – Дерево объектов

Теперь необходимо создать аппарат для смешивания, клапаны и насос для откачки. Для начала создадим новый объект в объекте «Участок» и назовем его «Оборудование». Далее переходим на палитру, выбираем «Аппараты» → «Емкость с мешалкой» и добавляем в «Оборудование». Также из вкладки палитры «Исполнительные механизмы» добавим элемент «Задвижка мнемоническая», назовем ее «Клапан 1» и дублируем один раз. Также из «Исполнительных механизмов» добавляем элемент «Насос мнемонический». Еще раз переходим в палитру, находим там «Управление» и добавляем оттуда элемент «Кнопка» в объект «Оборудование» для управления мешалкой. Через контекстное меню объекта

«Оборудование» добавим значение «Уровень мешалки», два расчета «Расчет уровня мешалки» и «Температура смеси» и команду «Выход». Основные элементы «Оборудования» добавлены, перейдем к их настройке и составлению связей.

Переходим к «Расчет уровня мешалки» и на вкладке «Общее» выбираем диапазон измерений «Уровень 1000». Далее переходим на вкладку «Формула», перетаскиваем туда «Уровень мешалки», «Выход» и «Управление потоком» из каждого бака и составляем формулу (рисунок 8.11).

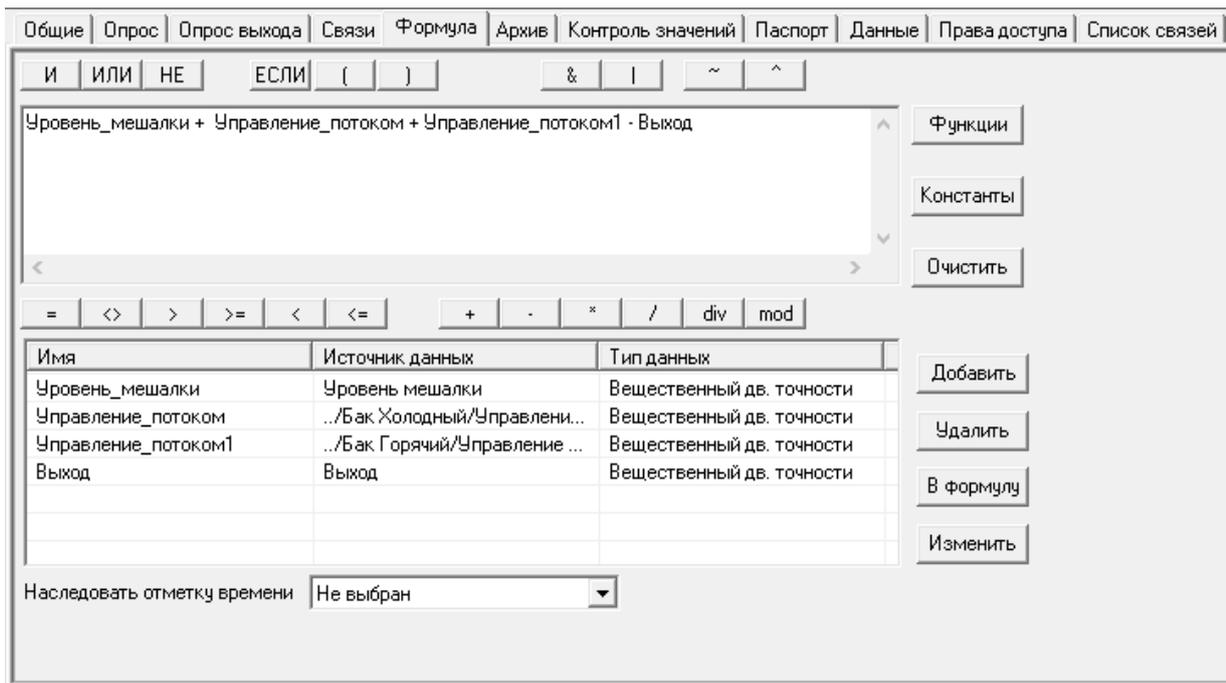


Рисунок 8.11 – Формула расчета уровня мешалки

Далее переносим «Расчет уровня мешалки» в значение «Уровень мешалки», тем самым связывая их. В свою очередь, «Уровень мешалки» переносим на входной параметр «Уровень» объекта «Емкость с мешалкой». Теперь перейдем во вкладку «Опрос» элемента «Уровень мешалки» и установим галочку возле показателя «Значение до опроса» и укажем его равным 0.

Для упрощения работы клапанов свяжем их такими же параметрами, что и насосы. Для этого выходы «Результат» элементов «Открыт» и «Закрыт» объекта «Бак холодный» связываем с входами «Открыт» и «Закрыт» элемента «Клапан 1» соответственно. Повторяем операцию для «Бак горячий» и «Клапан 2».

Для управления мешалкой свяжем выход «Состояние» элемента «Кнопка» с входом «Мешалка», как показано на рисунке 8.12.

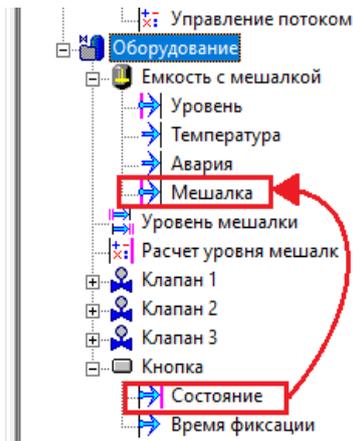


Рисунок 8.12 – Создание связи для управления мешалкой

Далее необходимо настроить отображение температуры смеси. Для этого воспользуемся формулой определения температуры смеси:

$$T_c = \frac{(V_1 \times T_1 + V_2 \times T_2)}{V_1 + V_2}, \quad (8.1)$$

где V_1 и V_2 – это количество холодной и горячей жидкости соответственно, л;
 T_1 и T_2 – их температуры соответственно, °С.

Поэтому нажимаем на «Температура смеси», переходим в «Формула» и записываем следующую формулу (рисунок 8.13).

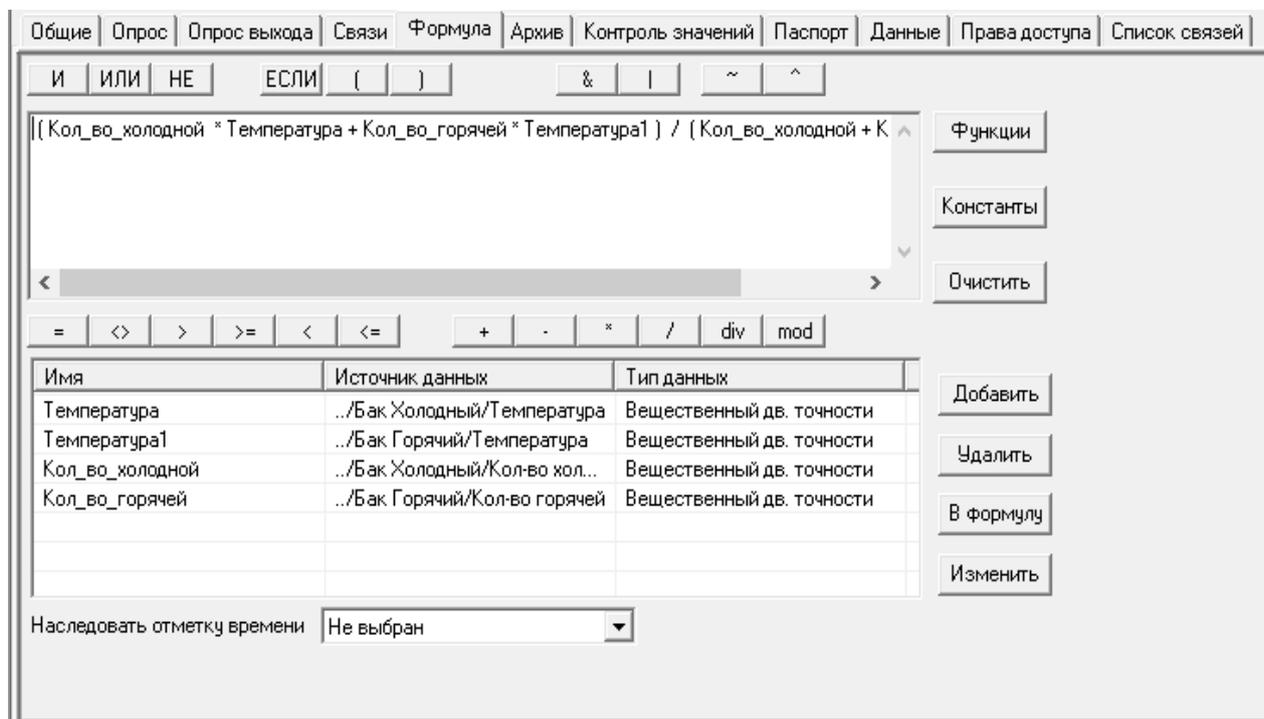


Рисунок 8.13 – Формула определения температуры смеси

Также необходимо настроить включение насоса при откачке жидкости. Для этого добавим из палитры «Вычисления» элементы « $A > B$ » и « $A = B$ ». Переименуем их как «Открыт» и «Закрыт» соответственно. Свяжем элемент «Выход» с параметрами «Значение A » элементов «Открыт» и «Закрыт». В свою очередь в «Значение B » зададим значение константой «0» (рисунок 8.14).

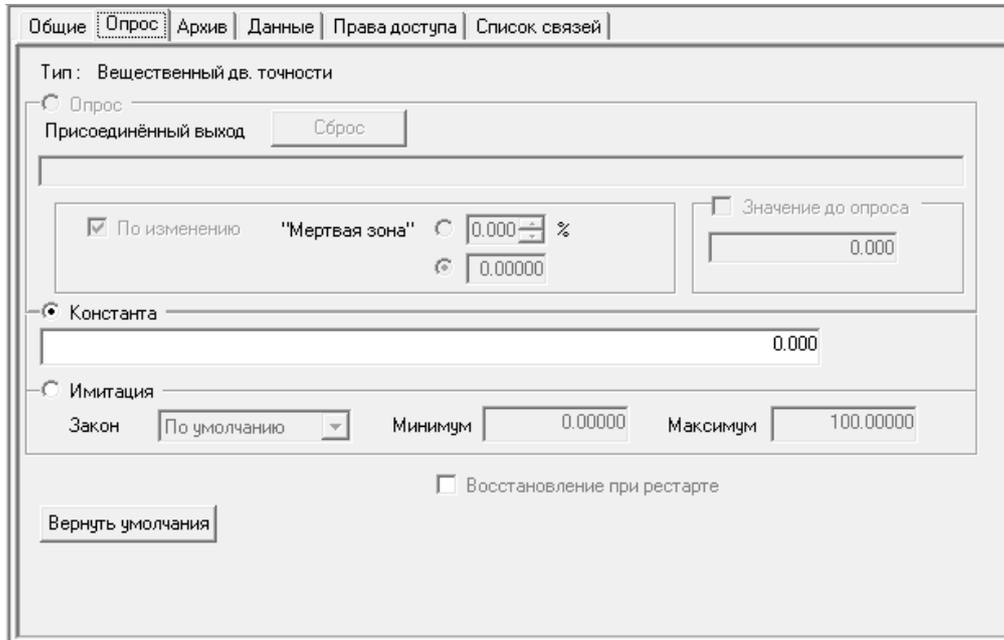


Рисунок 8.14 – Задание значения

Теперь перейдем к созданию мнемосхемы (рисунок 8.15).

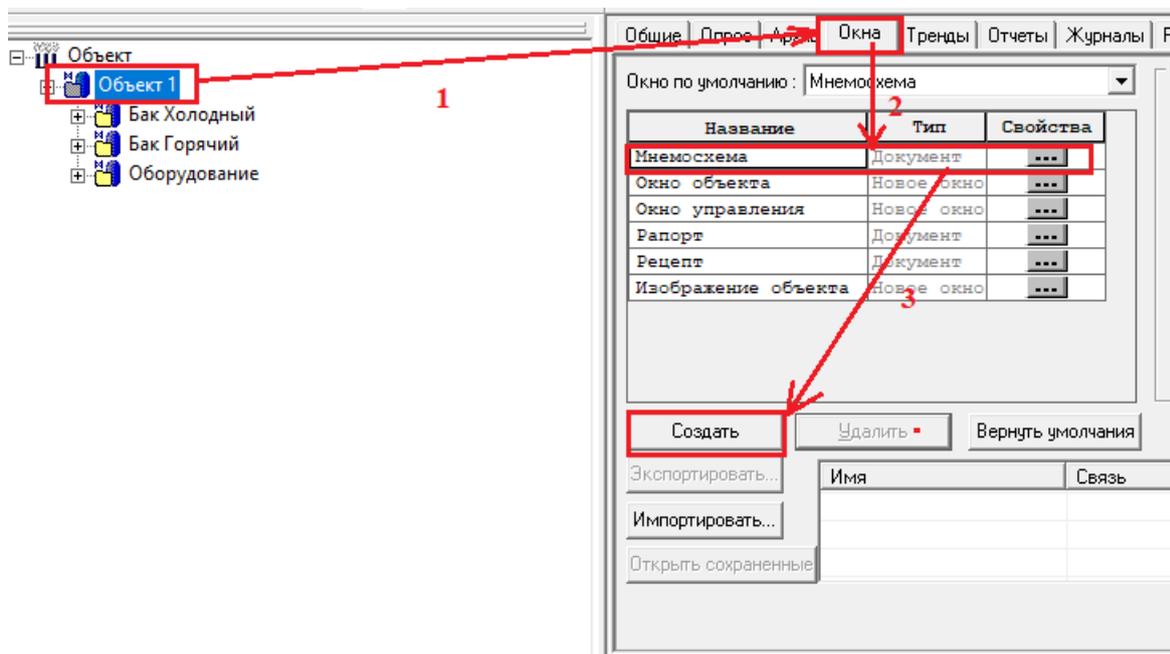


Рисунок 8.15 – Создание мнемосхемы

Далее необходимо, удерживая левую кнопку мыши, перетащить элементы «Емкость» из объектов «Бак холодный» и «Бак горячий» (рисунок 8.16).

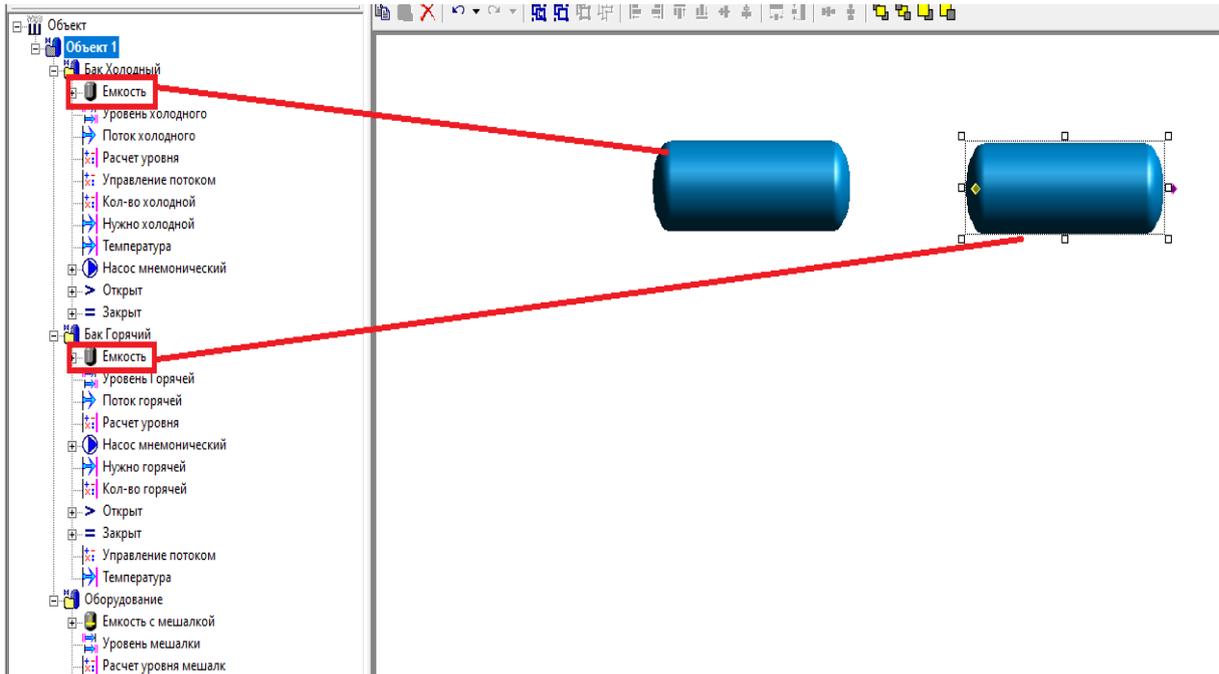


Рисунок 8.16 – Добавление элементов на мнемосхему

Аналогичным образом перетаскиваем насосы, клапаны и емкость с мешалкой на мнемосхему. Выделяем насосы и клапаны и в панели свойств, в правом нижнем углу, выбираем направление «Вертикально» (рисунок 8.17).

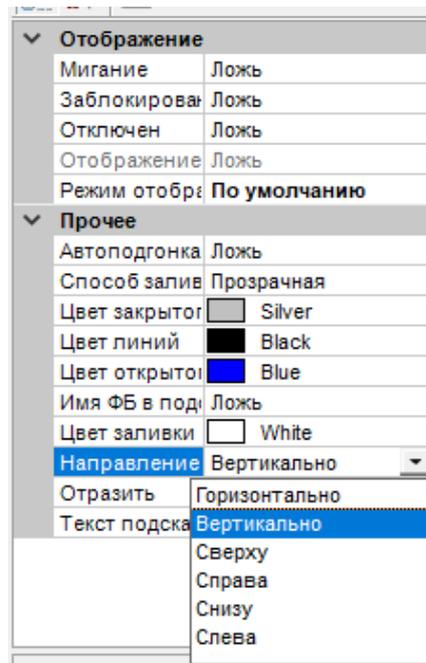


Рисунок 8.17 – Выбор направления

Далее добавим элементы для задания параметров. Для этого, удерживая правую кнопку мыши, по очереди перетаскиваем «Поток холодной», «Поток горячей», «Нужно холодной», «Нужно горячей», «Выход» и в появившемся контекстном меню выбираем «Слайдер». Располагаем их в удобное место.

Также необходимо добавить элементы контроля параметров уровней жидкостей. Для этого, удерживая левую клавишу мыши, перетаскиваем значения «Уровень холодной» и «Уровень горячей» на мнемосхему и располагаем над соответствующими баками. Удерживая правую клавишу мыши, перенесем на мнемосхему «Уровень мешалки», выберем «Индикатор», вызовем контекстное меню нажатием правой кнопкой мыши по индикатору, перейдем в «Свойства» → «Вид» и укажем заголовок «Уровень». Также необходимо добавить индикатор температуры смеси. Для этого, удерживая правую кнопку мыши, перетаскиваем расчет «Температура смеси» на мнемосхему и выбираем «Индикатор».

Добавим элементы контроля количества прошедшей через клапаны жидкости. Для этого перенесем расчеты «Кол-во холодной» и «Кол-во горячей». Далее перейдем в «Свойства». На вкладке «Вид» выберем тип отображения «Заполнение цветом». На вкладке «Граница» укажем толщину 1 и вид «Плоская». На вкладке «Цвет» выберем цвет фона белый. На вкладке «Шрифт» укажем размер равный 10. Расположим элементы рядом с клапанами.

Элемент «Кнопка» перетащим аналогичным образом на мнемосхему и расположим в удобном месте.

Для наглядности работы проведем трубопровод от баков до аппарата смешивания. В правом нижнем углу необходимо выбрать «Палитра» → «Объемные элементы» → «Трубопровод» (рисунок 8.18).

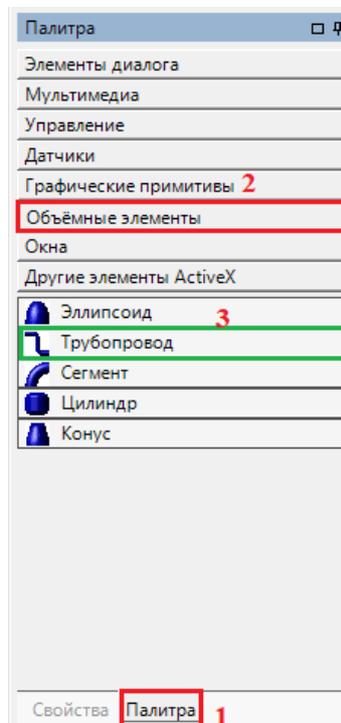


Рисунок 8.18 – Добавление трубопровода

Добавляем трубопровод на мнемосхему и проводим от баков до аппарата смешивания. Двойной щелчок мыши при выбранном трубопроводе создает искривление трубы в направлении курсора мыши.

Конечная мнемосхема может иметь вид, показанный на рисунке 8.19.

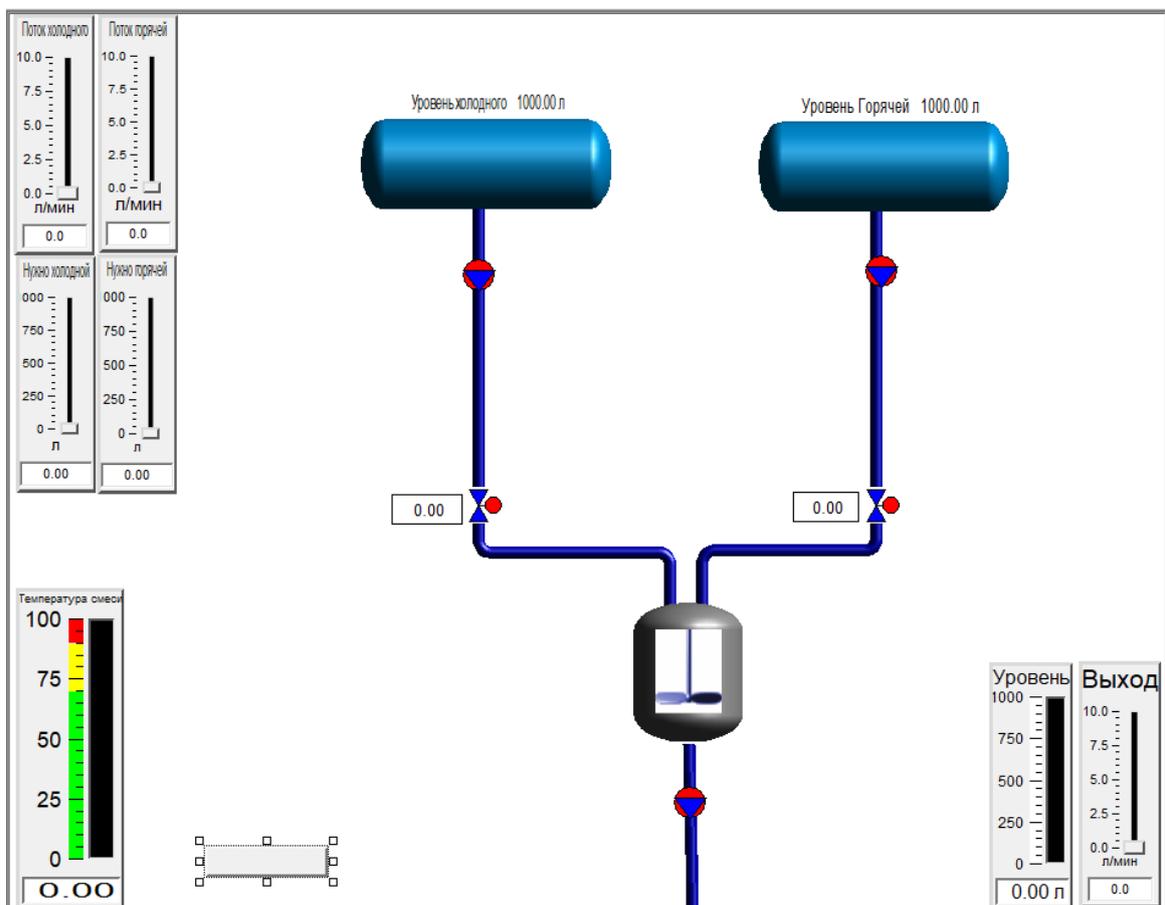


Рисунок 8.19 – Конечная мнемосхема

Нажатием на «Показать проект»  закроем окно мнемосхемы и вернемся в рабочую область. Чтобы при запуске проекта сразу отрывалась мнемосхема, необходимо перейти в объект «Участок» и на вкладке окна поставить галочку напротив «Сделать мнемосхему стартовой».

8.3 Индивидуальные задания

В рамках лабораторной работы предусмотрено выполнение заданий в соответствии с индивидуальным номером зачетной книжки каждого студента. При этом важно учесть, что повторение заданий не допускается, обеспечивая таким образом самостоятельное выполнение своей работы. В случае совпадения номера зачетной книжки с уже занятым заданием студенту предоставляется возможность выбрать свободный номер из предложенных вариантов.

Варианты заданий представлены в таблице 8.1.

Таблица 8.1 – Варианты индивидуальных заданий

Вариант	Нужно холодной	Нужно горячей
1	15	340
2	823	44
3	517	483
4	563	420
5	162	804
6	730	91
7	196	765
8	797	126
9	939	61
10	356	559
11	271	310
12	440	357
13	388	475
14	303	642
15	761	77
16	287	698
17	690	204
18	597	105
19	616	73
20	862	113
21	787	202
22	25	351
23	733	256
24	250	594
25	654	302
26	212	686
27	232	750
28	931	69
29	632	303
30	852	122

8.4 Содержание отчета

В рамках лабораторной работы № 8 необходимо выполнить одно индивидуальное задание. В связи с этим отчет по лабораторной работе должен содержать:

- 1) титульный лист;
- 2) введение:
 - 2.1) цель выполнения задания;
 - 2.2) краткое описание задания;
- 3) основная часть:
 - 3.1) ход выполнения задания;
 - 3.2) обработка и анализ результатов выполнения задания;
- 4) выводы по результатам выполнения задания.

8.5 Контрольные вопросы

- 1 Какое основное предназначение элемента «Задвижка мнемоническая»?
- 2 Какие компоненты необходимы для создания проекта для смешивания жидкостей разной температуры в *MasterSCADA*?
- 3 Какие термодинамические аспекты играют ключевую роль в таком проекте?
- 4 Как можно улучшить эффективность процесса смешивания жидкостей?
- 5 Как определить температуры смеси?
- 6 Как можно обеспечить, чтобы температура смеси не превышала допустимого значения?
- 7 Возможно ли реальное применение данного проекта? Где?

Вспомогательная литература

- 1 Пьявченко, Т. А. Проектирование АСУТП в SCADA-системе / Т. А. Пьявченко, Н. А. Куцевич, О. В. Синенко. – Таганрог : ТРТУ, 2007. – 78 с.
- 2 Скороспешкин, В. Н. Автоматизированные информационно-управляющие системы / В. Н. Скороспешкин, А. В. Чупин, О. В. Синенко. – Томск : ТПУ, 2009. – 57 с.
- 3 Денисенко В. В. Компьютерное управление технологическим процессом, экспериментом, оборудованием / В. В. Денисенко. – М. : Горячая линия – Телеком, 2009. – 608 с.
- 4 ИнСАТ «Master SCADA. Руководство пользователя. Версия 3.X.». – Москва, 2017. – 574 с.
- 5 ИнСАТ «Master SCADA. Основы проектирования. Методическое пособие». – Москва, 2016. – 277 с.
- 6 Master SCADA: учеб. пособие для студентов специальности 5311000 – Автоматизация и управление технологических процессов (в водном хозяйстве) / Р. Т. Газиева [и др.]. – Ташкент, 2020. – 105 с.

ПРИЛОЖЕНИЕ А

(обязательное)

Загрузка и установка *LabVIEW* на *Windows*

Для загрузки программы *LabVIEW* необходимо перейти на сайт организации *National Instruments* по адресу <https://www.ni.com/en.html> (рисунок А.1).

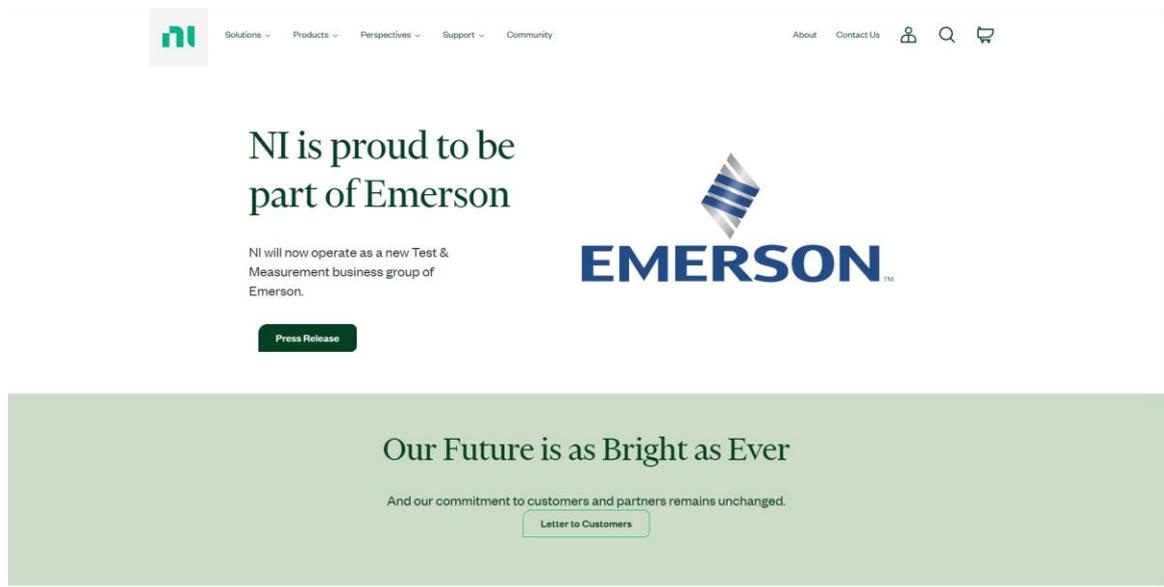


Рисунок А.1 – Главная страница сайта *National Instruments*

Далее необходимо нажать на «*Products*» и в разделе «*Software*» выбрать *LabVIEW* (рисунок А.2).

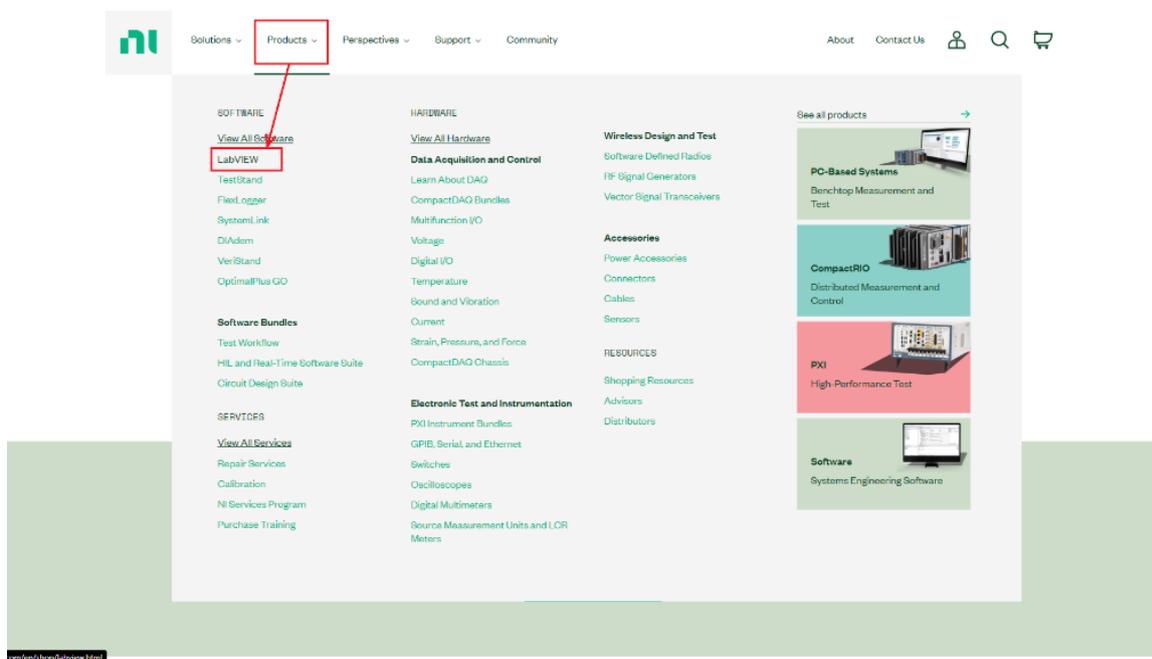


Рисунок А.2 – Вкладка «*Products*»

На странице *LabVIEW*, немного пролистав ниже, необходимо нажать кнопку «*Download LabVIEW*» (рисунок А.3).

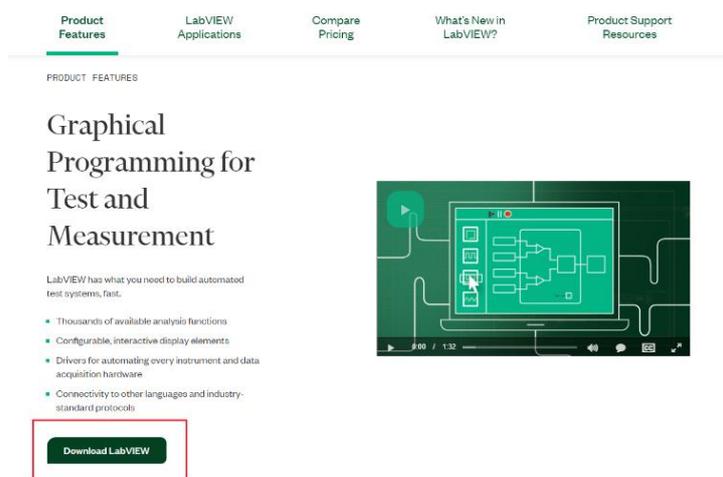


Рисунок А.3 – Страница *LabVIEW*

Вы перейдете на страницу выбора версии *LabVIEW* для скачивания. Компания *National Instrument* предлагает бесплатные *Community* версии, начиная с 2023 года. Поэтому выберите необходимый год и установите маркер рядом с *Community* (рисунок А.4). Обратите внимание на то, чтобы в поле «*Application Bitness*» было значение «*32-bit*», так как именно эта версия необходима для успешного выполнения всех лабораторных работ.

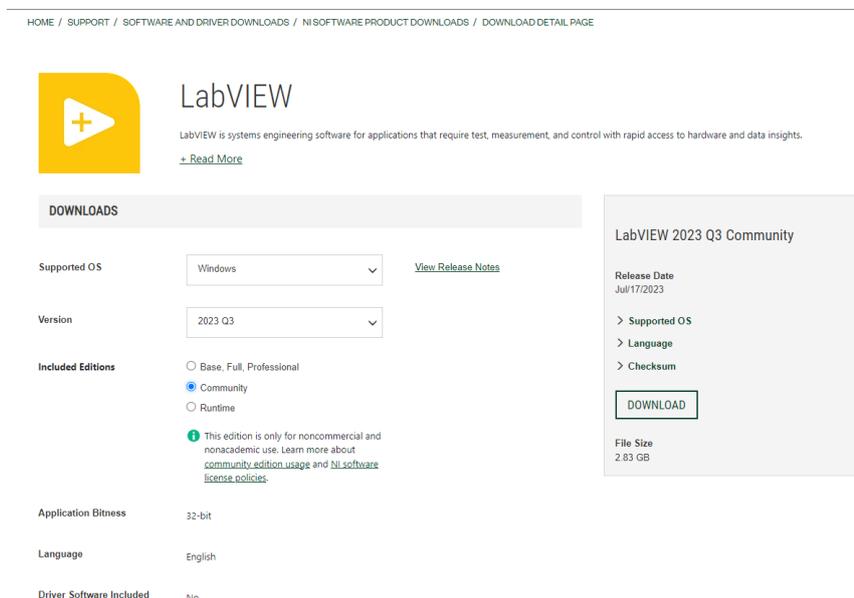


Рисунок А.4 – Страница скачивания *LabVIEW*

После нажимаем «*Download*» и ждем окончания загрузки.

По окончании загрузки открываем скачанный *iso*-файл и запускаем установщик  *Install.exe*.

ПРИЛОЖЕНИЕ Б

(обязательное)

Загрузка и установка *MasterSCADA* на *Windows*

Для начала необходимо запустить файл . В появившемся окне нажимаем «Далее» (рисунок Б.1).

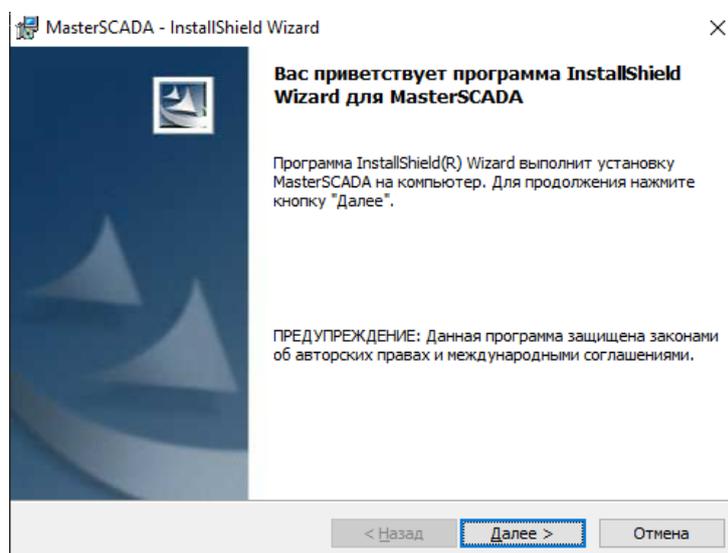


Рисунок Б.1 – Окно установки *MasterSCADA*

После следует окно лицензионного соглашения. Ознакомившись с ним, принимаем условия лицензионного соглашения и нажимаем «Далее», как показано на рисунке Б.2.

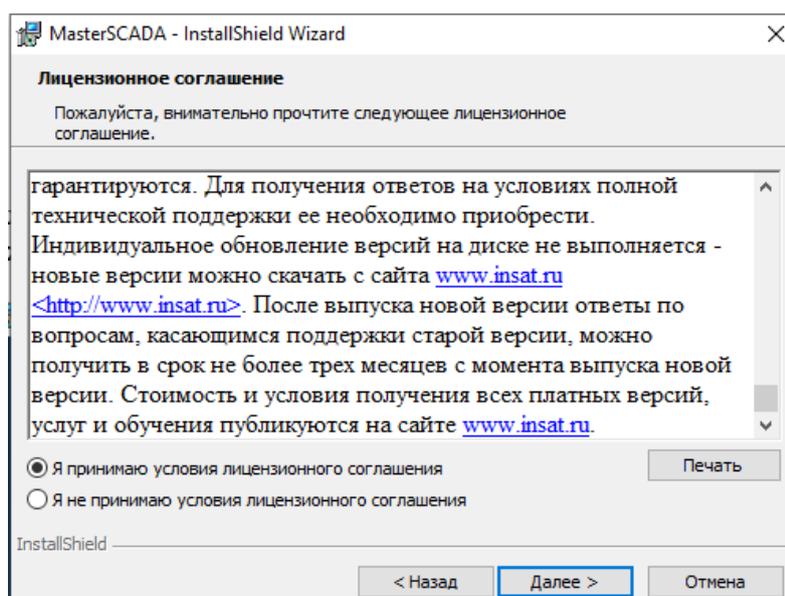


Рисунок Б.2 – Окно принятия лицензионного соглашения

Укажем путь для хранения проектов и нажимаем «Далее» (рисунок Б.3).

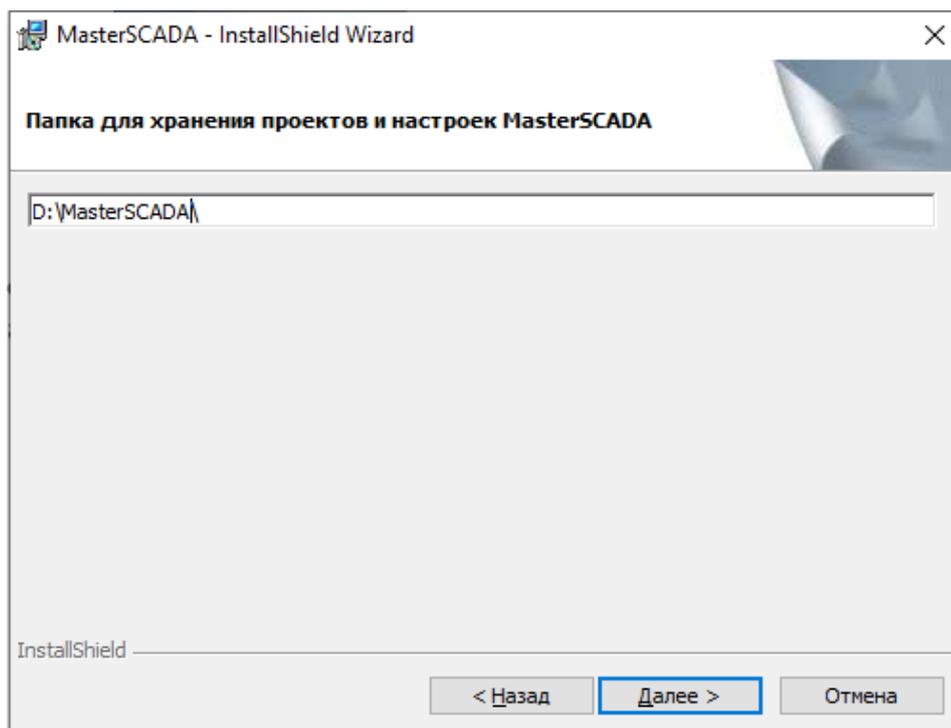


Рисунок Б.3 – Окно указания места хранения проектов

В следующем окне выбираем путь установки программного обеспечения и нажимаем «Далее» (рисунок Б.4).

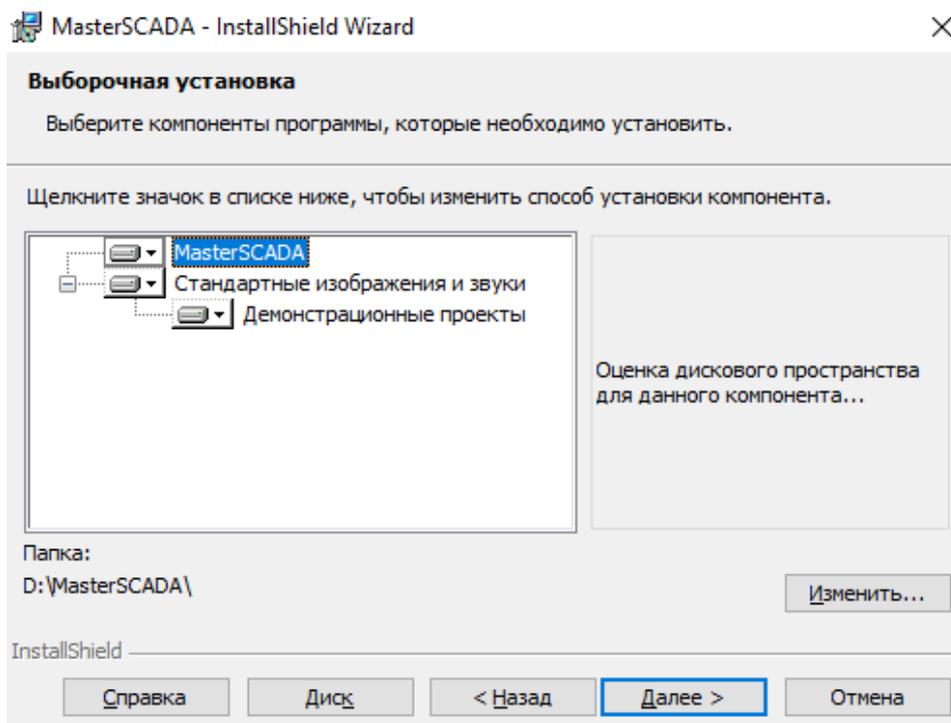


Рисунок Б.4 – Окно указания пути установки

Нажимаем «Установить» и дожидаемся конца установки (рисунок Б.5).

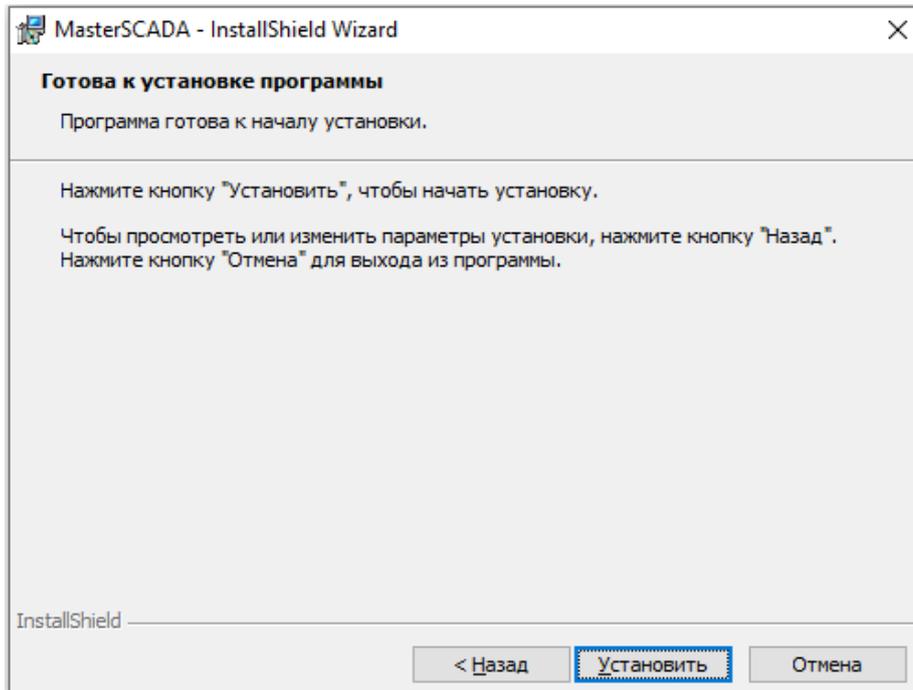


Рисунок Б.5 – Окно подтверждения установки

Далее нажимаем «Готово» для завершения установки (рисунок Б.6).

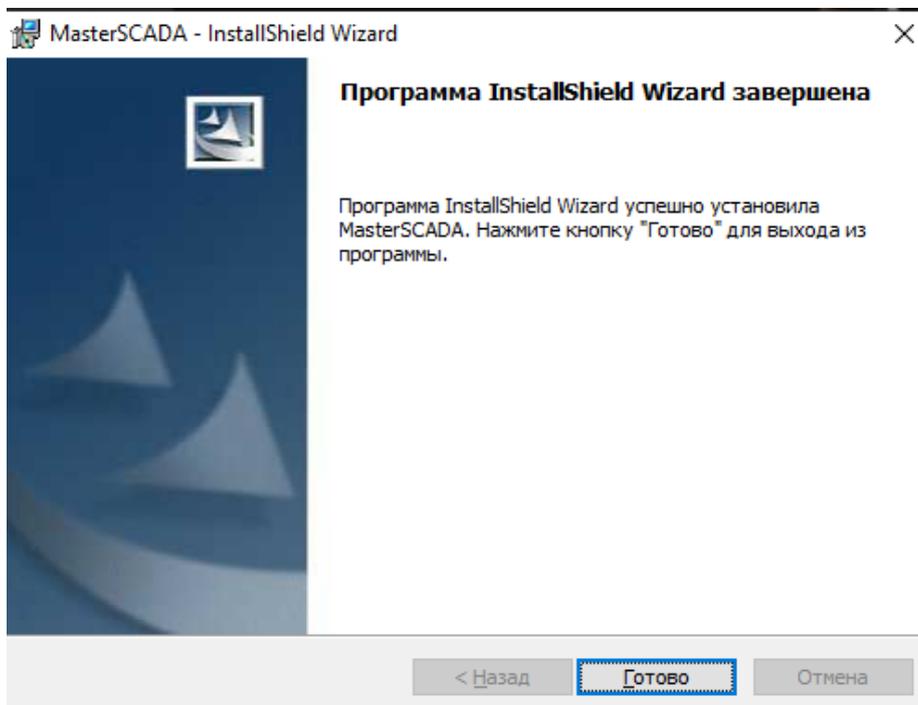


Рисунок Б.6 – Окно завершения установки

ПРИЛОЖЕНИЕ В

(обязательное)

Загрузка и установка *OPC* сервера на *Windows*

Для начала необходимо перейти по ссылке <https://masteropc.ru/download> в любом браузере и установить *Modbus Universal MasterOPC 32* (рисунок В.1).



Демонстрационные версии *Modbus Universal MasterOPC*



Рисунок В.1 – Скачивание установочного файла *Modbus Universal MasterOPC 32*

После этого запустите установочный файл, находящийся в *ZIP*-архиве. В появившемся окне нажимаем «Далее» (рисунок В.2).



Рисунок В.2 – Окно начала установки

После следует окно, где необходимо указать путь установки сервера. Выбрав место установки, нажимаем «Далее» (рисунок В.3).

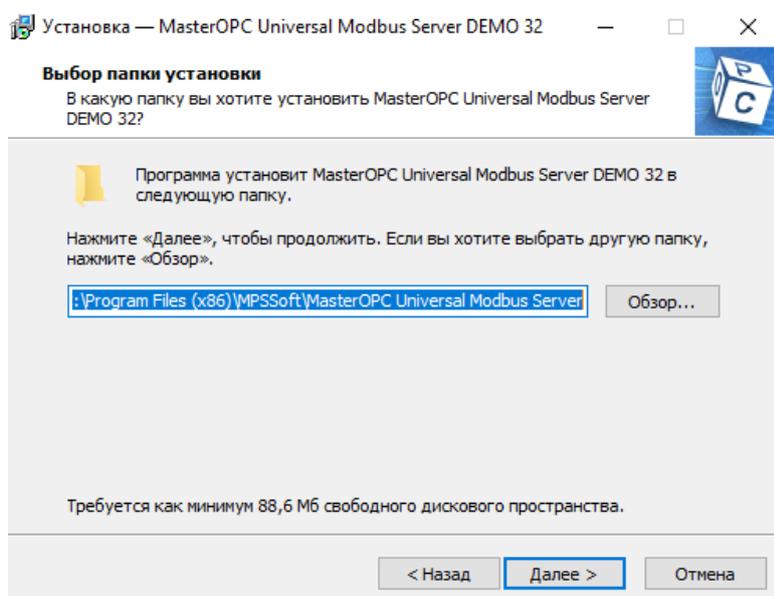


Рисунок В.3 – Окно указания пути установки

В следующем окне предлагают выбрать папку в меню «Пуск». Выбрав папку или отказавшись от создания папки в меню «Пуск», нажимаем «Далее» (рисунок В.4).

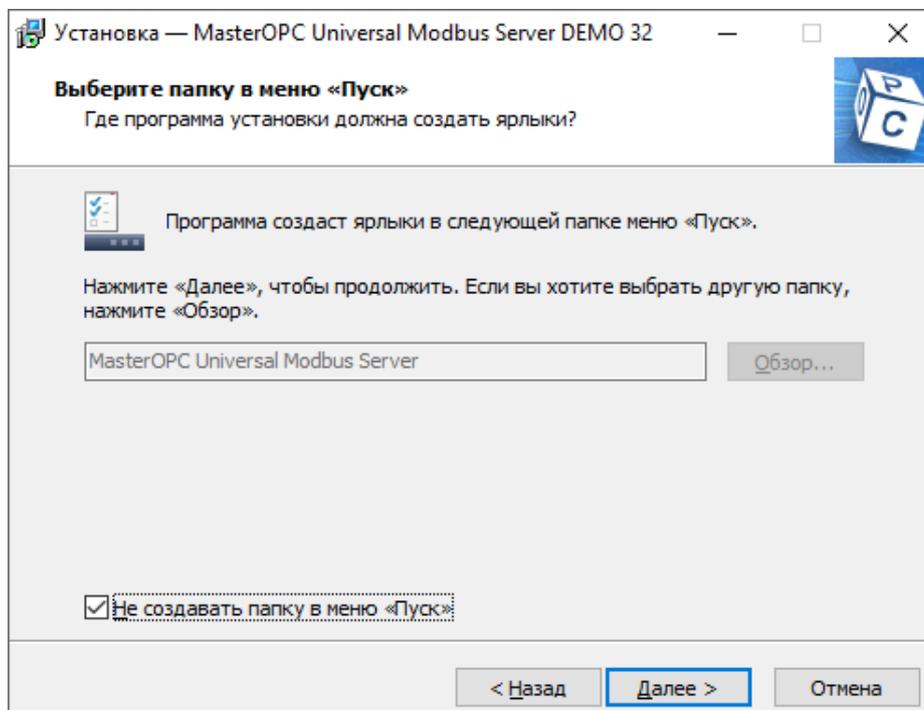


Рисунок В.4 – Окно создания папки в меню «Пуск»

Далее следует окно создания ярлыка на рабочем столе. Убираем или оставляем галочку и нажимаем «Далее» (рисунок В.5).

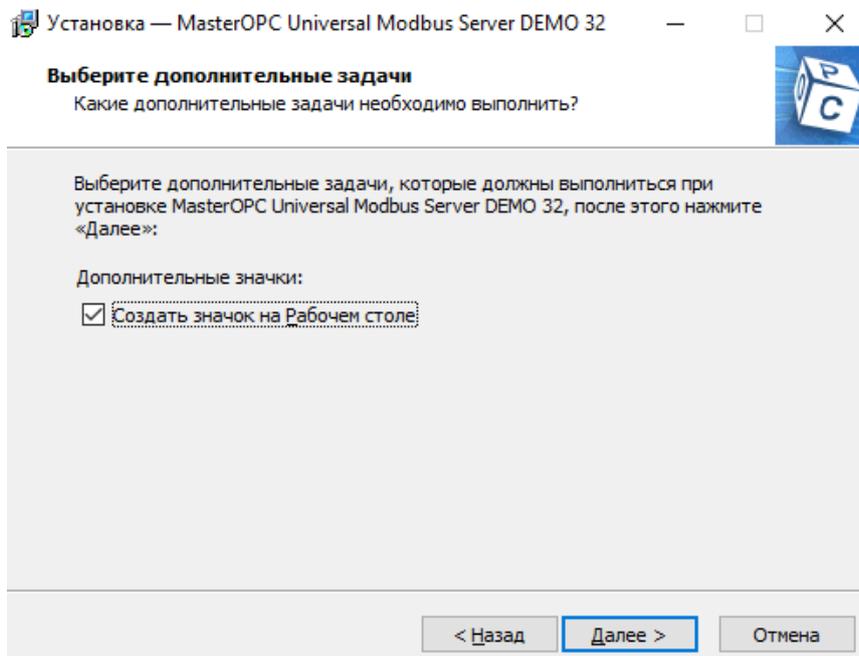


Рисунок В.5 – Окно создания ярлыка на рабочем столе

Нажимаем «Установить» и ждем конца установки (рисунок В.6).

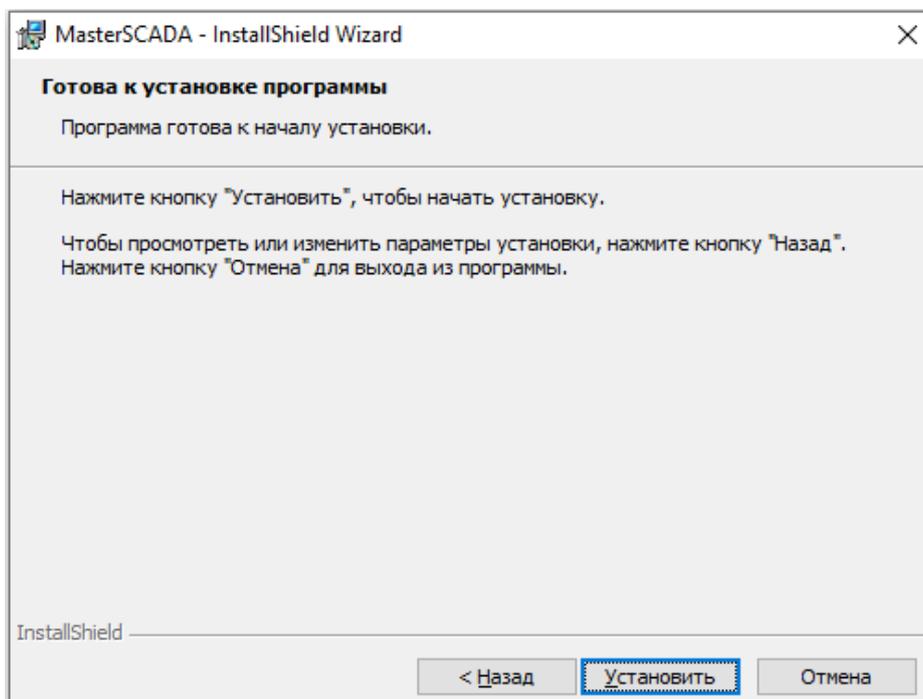


Рисунок В.6 – Окно подтверждения установки

Далее нажимаем «Завершить» для завершения установки (рисунок В.7).

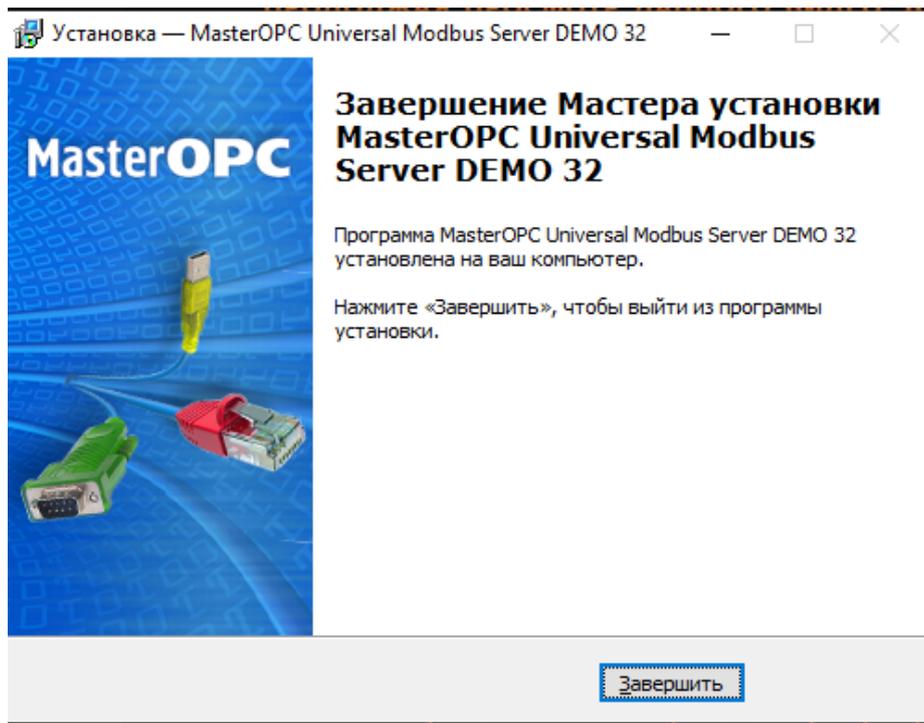


Рисунок В.7 – Окно завершения установки

Учебное издание

Пискун Геннадий Адамович

**ПРОГРАММНО-ТЕХНИЧЕСКИЕ СРЕДСТВА
МНОГОПРОФИЛЬНЫХ СИСТЕМ.
ЛАБОРАТОРНЫЙ ПРАКТИКУМ**

УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ

Корректор *Е. Н. Батурчик*

Компьютерная правка, оригинал-макет *В. М. Задоля*

Подписано в печать 31.05.2024. Формат 60×84 1/16. Бумага офсетная. Гарнитура «Таймс».
Отпечатано на ризографе. Усл. печ. л. 11,74. Уч.-изд. л. 12,7. Тираж 30 экз. Заказ 37.

Издатель и полиграфическое исполнение: учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники».
Свидетельство о государственной регистрации издателя, изготовителя,
распространителя печатных изданий №1/238 от 24.03.2014,
№2/113 от 07.04.2014, №3/615 от 07.04.2014.
Ул. П. Бровки, 6, 220013, г. Минск