**Загрузить *LabView 6.1* можно по ссылке:** [**https://yadi.sk/d/JRhS6ITDqHZqC**](https://yadi.sk/d/JRhS6ITDqHZqC)

**Загрузить *LabView 2016* можно по ссылке:**

[**https://drive.google.com/open?id=18EHK-h-3kCJtAuDELiijaligTm5r8ohQ**](https://drive.google.com/open?id=18EHK-h-3kCJtAuDELiijaligTm5r8ohQ)

**Задание 6.** В пакете прикладных программ *National Instruments* *LabView* разработать виртуальный прибор «Простой Калькулятор», позволяющий оператору по итогам работы этого виртуального прибора наблюдать результаты выполнения следующих арифметических операций:

* сложения двух элементов,
* вычитания двух элементов,
* умножения двух элементов,
* деления двух элементов.

В рамках разработки графического пользовательского интерфейса предусмотреть:

* две ячейки (контроллера) для ввода числовой информации (по количеству элементов, над которыми выполняются арифметические операции);
* четыре ячейки (индикатора) для вывода числовой информации (по количеству используемых арифметических операций);
* лампочку (логический индикатор) для вывода логической информации с целью реагирования на ошибочный ввод исходных данных – ноль в знаменателе.

Все интерфейсные элементы должны содержать значащие имена, отражающие назначение этих элементов, например, ячейки для ввода информации могут быть названы:

* «Операнд 1», «Операнд 2»;
* «Слагаемое 1 / Уменьшаемое / Множитель 1 / Делимое»,

«Слагаемое 2 / Вычитаемое / Множитель 2 / Делитель».

Необходимо продумать индивидуальный авторский дизайн графического пользовательского интерфейса «Простого Калькулятора». Двух одинаковых дизайнерских решений для виртуального прибора «Простой Калькулятор» в одной учебной студенческой группе быть не должно.

Изменения, вносимые в исходные данные, должны сказываться на отображаемых результатах в реальном времени.

При подготовке блок-диаграммы графического кода программы минимизировать количество пересечений линий связи, а также попытаться создать максимально компактную и читаемую схему.

**Выше сформулированы требования для выполнения общей части работы.**

По итогам выполнения работы сдаются строго четыре файла:

- отчёт, выполненный в текстовом редакторе *Microsoft Office Word* (*\*.doc* или *\*.docx*);

- файл виртуального прибора *National Instruments LabView* (*\*.vi*) по общей части работы;

- файл виртуального прибора *National Instruments LabView* (*\*.vi*) по индивидуальной части работы, связанной с закреплением навыков использования переменных (контроллеров и индикаторов);

- файл виртуального прибора *National Instruments LabView* (*\*.vi*) по индивидуальной части работы, связанной с закреплением навыков использования констант и некоторых стандартных функций.

Отправленные поодиночке файлы проверке не подлежат. При отсутствии одного из упомянутых файлов зачёт по заданию не выставляется.

**Требования к именам файлов:**

**Общий вид формата имени файла:** «*Дата. Задание. Фамилия.mcdx*»

**Формат записи даты:** «*ГГГГММДД*», где *ГГГГ* – четыре цифры текущего года, *ММ* – две цифры текущего месяца, *ДД* – две цифры текущего дня.

**Формат записи задания:** «Задание *NNk*», где *NN* – две цифры номера задания, *k* – обозначение «о», если файл содержит общую часть; обозначение «и», если файл содержит индивидуальную часть; обозначение «ои», если файл содержит как общую, так и индивидуальную части.

**Если устранить замечания по работе удаётся в тот же день:** после фамилии ставится пробел и в круглых скобках записывается номер попытки исправления.

**Примеры правильных имён файлов, которые сдаются на проверку впервые:**

«*20181128. Задание 06ои. Иванов.docx*»

«*20181128. Задание 06ои. Иванов.vi*»

**Примеры правильных имён файлов, которые сдаются на проверку повторно в тот же день:**

«*20181128. Задание 06ои. Иванов (1).docx*»

«*20181128. Задание 06ои. Иванов (1).vi*»

**Внимание!** Не забудьте выполнить автоматическую нумерацию страниц в отчёте.

Отчёт по выполненной работе должен содержать:

0. Титульный лист.

1. Формулировку цели работы.

2. Описание задачи согласно выданному варианту.

2.1. Общая часть.

2.2. Индивидуальная часть.

3. Составление блок-схемы алгоритма программы.

3.1. Общая часть.

3.2. Индивидуальная часть.

4. Подбор и расчёт тестовых примеров.

4.1. Общая часть.

4.2. Индивидуальная часть.

5. Листинг кода составленного программного обеспечения (блок-диаграммы *LabView*).

5.1. Общая часть.

5.2. Индивидуальная часть.

6. Графический пользовательский интерфейс программного обеспечения (передняя панель виртуального прибора *LabView*) и его описание.

6.1. Общая часть.

6.2. Индивидуальная часть.

7. Расчёт тестовых примеров с использованием составленного программного обеспечения.

7.1. Общая часть.

7.2. Индивидуальная часть.

8. Формулировку вывода о проделанной работе (обезличено – исключить из вывода местоимения, такие как «я», «мы» и другие).

Рекомендации к отчёту, доказывающие самостоятельность выполнения работы и упрощающие процедуру проверки отчёта преподавателем:

1. Выполнение дополнительных скриншотов для случаев, когда текстовое описание проделанных действий становится громоздким или трудным к восприятию.

2. Нумерация рисунков (если есть) с подписями, содержащими названия рисунков, например, «Рисунок 1 – Пользовательский интерфейс *Microsoft Office Excel*».

----------------------------------------------------------------------------------------------------

**Ознакомление с графическим пользовательским интерфейсом пакета прикладных программ *National Instruments LabView 6.1*.**

После запуска исполняемого файла пакета прикладных программ *National Instruments LabView 6.1* открывается (если настройками не предусмотрено другого) диалоговое окно, в котором можно (Рисунок 1):

- приступить к созданию нового виртуального прибора;

- продолжить разработку ранее созданного виртуального прибора;

- приступить к изучению материалов и примеров, позволяющих быстро начать работу с пакетом прикладных программ *National Instruments LabView 6.1*.

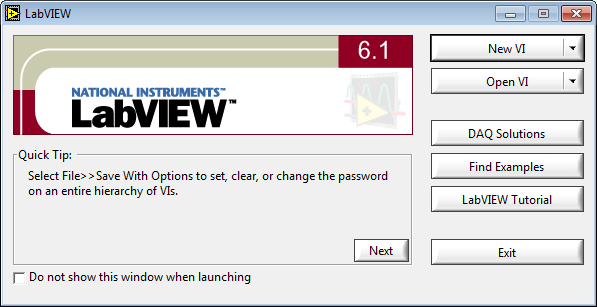


Рисунок 1 – Начальное диалоговое окно *National Instruments LabView 6.1*

Для начала работ по созданию нового виртуального прибора необходимо нажать на кнопку «*New VI*». Сокращение *VI* в данном случае является обозначением самого виртуального прибора (*Virtual Instrument* в англоязычной литературе, Виртуальный Прибор – в русскоязычной литературе). Выбор данного режима автоматически генерирует две рабочих области, каждая из которых имеет собственное устойчивое наименование. Как правило, сверху расположена передняя панель (*Front Panel*) – графический пользовательский интерфейс виртуального прибора, под ней – блок-диаграмма кода (*Block Diagram*). Область передней панели нового виртуального прибора показана на Рисунке 2.

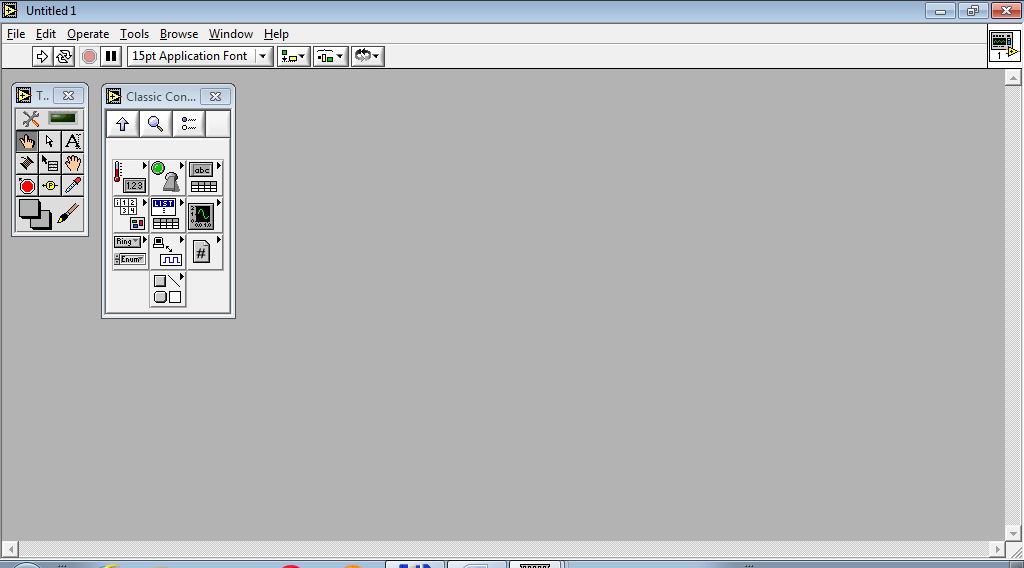


Рисунок 2 – Область передней панели нового виртуального прибора *National Instruments LabView 6.1*

Для перехода от передней панели к блок-диаграмме опытные пользователи применяют сочетание клавиш «*Ctrl*» + «*E*», менее опытные – используют переход через меню «*Window* > *Show Diagram*», как это показано на Рисунке 3.

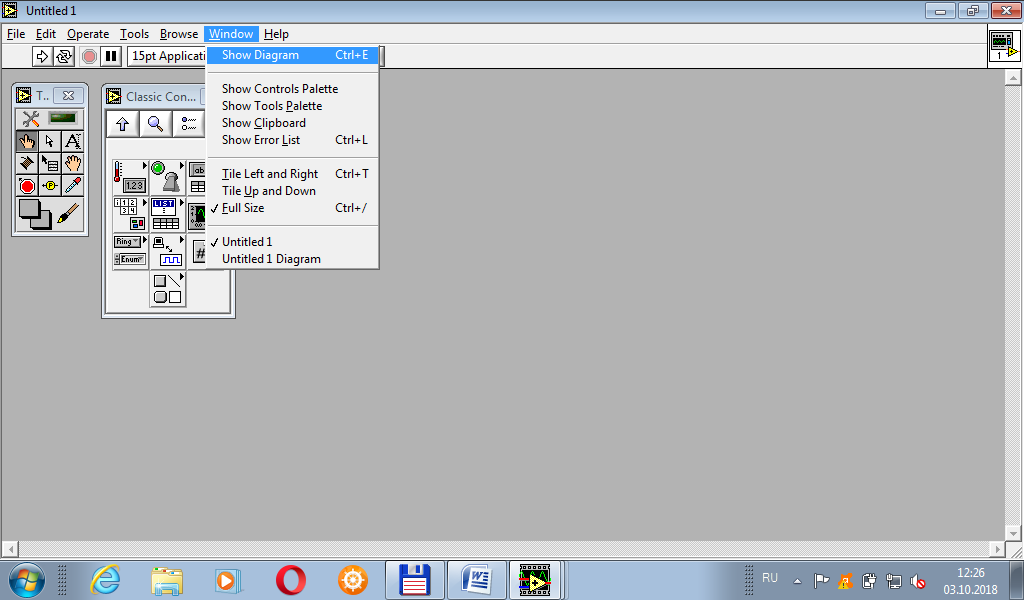


Рисунок 3 – Переход к блок-диаграмме через пункты меню

На Рисунке 4 показана область блок-диаграммы нового виртуального прибора.

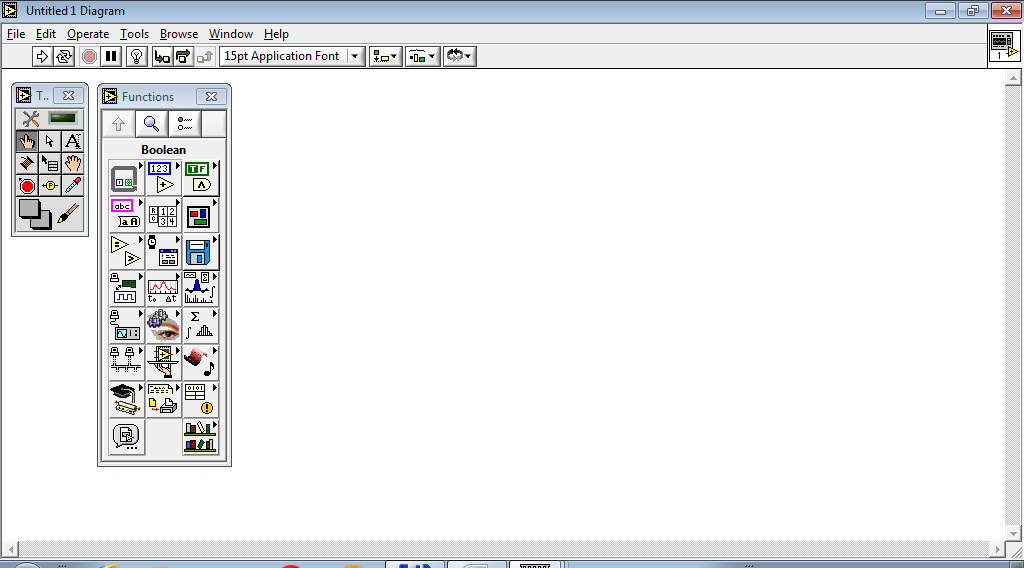


Рисунок 4 – Область блок-диаграммы нового виртуального прибора *National Instruments LabView 6.1*

**Настройка шрифтов, адекватно отображающих русскоязычный текст (кириллицу)**

На сегодняшний день известна одна из проблем *National Instruments LabView 6.1*, связанная с некорректным отображением кириллических шрифтов при настройках, заданных по умолчанию. Данная проблема имеет решение.

На панели инструментов, расположенной строкой ниже по отношению к главному меню, необходимо нажать на самую большую кнопку (она ответственна за выбор и настройку параметров шрифта). Переход к основным настройкам производится путём нажатия на самую первую строку выпадающего списка – это вызов диалогового окна шрифтов (*Font Dialog…*). Расположение показано на Рисунке 5.

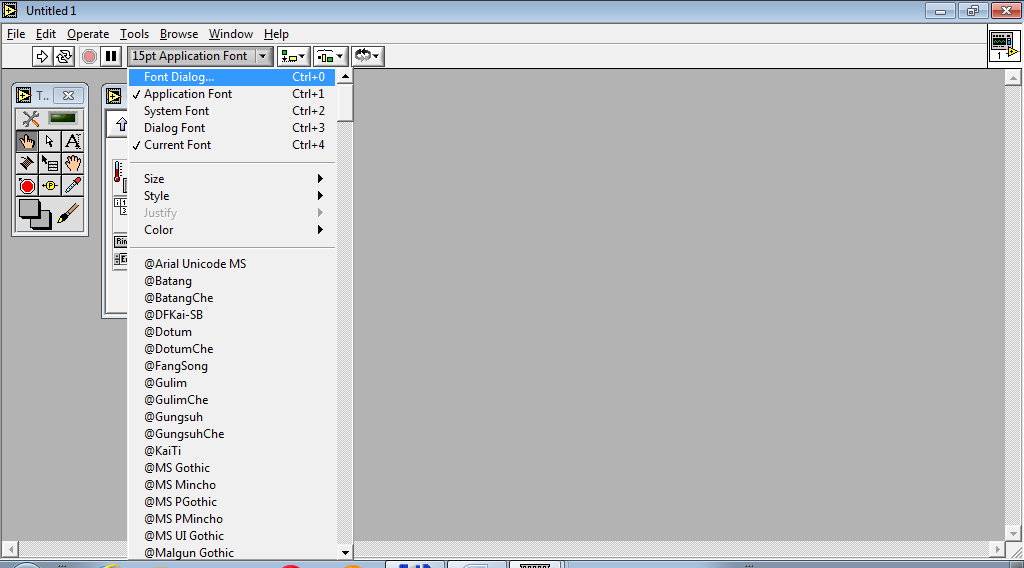


Рисунок 5 – Переход к диалоговому окну шрифтов

В раскрывшемся диалоговом окне необходимо проследить за тем, чтобы обе галочки «Умолчания для панели» (*Panel Default*) и «Умолчания для диаграммы» (*Diagram Default*) были выставлены. После этого в комбинированном списке шрифтов (*Font*) выбирается любой кириллический шрифт – для него характерен постфикс «*Cyr*». Все остальные умолчания подбираются на усмотрение автора виртуального прибора, либо оставляются без изменений (Рисунок 6).

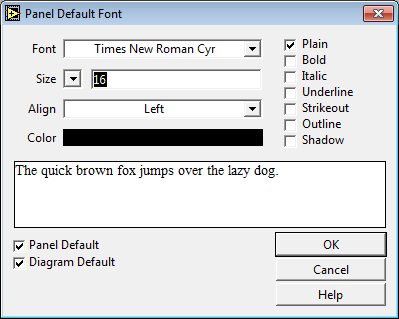


Рисунок 6 – Рекомендации по настройке шрифта, заданного в *National Instruments LabView* по умолчанию

При выставленных флагах «*Panel Default*» и «*Diagram Default*» указанные настройки шрифта будут установлены по умолчанию для разрабатываемого виртуального прибора.

**Подготовка среды *National Instruments LabView* к нормальной работе**

Стоит отметить, что при всём многообразии функций и возможностей пакета прикладных программ *National Instruments LabView* в первой работе (Задание 9) рассматривается и закрепляется лишь небольшой их фрагмент.

Сперва проверим готовность среды к нормальной работе. Допустим, при первом запуске *National Instruments LabView* все панели инструментов закрыты (Рисунок 7).

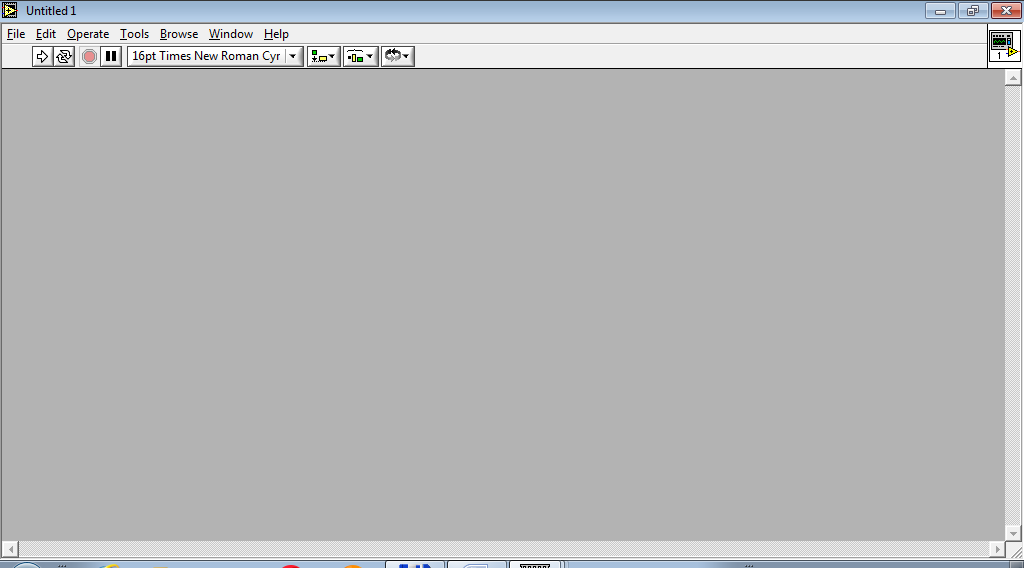


Рисунок 7 – Состояние *National Instruments LabView 6.1*, при котором основные панели инструментов скрыты

Для отображения в рабочей области основных панелей инструментов, достаточно помнить места расположения пунктов меню, связанных с этими программными компонентами. Они находятся в том же пункте меню «*Window*», через который осуществляется переход между областями передней панели и блок-диаграмы. Сначала отобразим панель элементов управления, которые позволяют обеспечить взаимодействие конечного пользователя виртуального прибора с его внутренними механизмами. Для этого проходим по меню путь «*Window > Show Controls Palette*» (Рисунок 8).

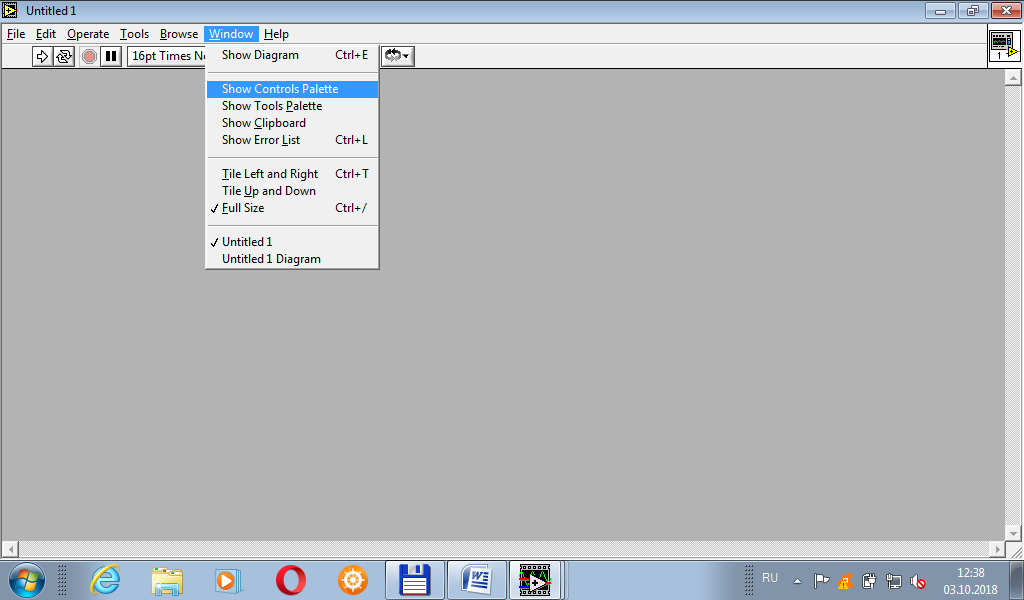


Рисунок 8 – Переход по меню к отображению панели элементов управления

В результате выбора указанного пункта меню над передней панелью отобразится структура, представленная на Рисунке 9.

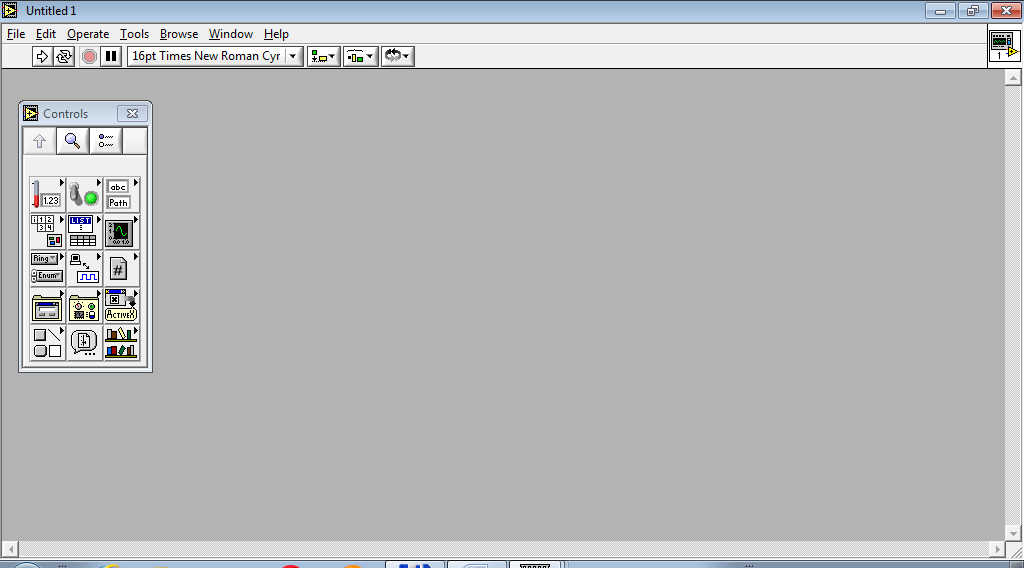


Рисунок 9 – Отображение панели элементов управления

Далее отобразим панель основных инструментов, предоставляющих возможность манипулировать элементами интерфейса и кода. Для этого проходим по меню путь «*Window > Show Tools Palette*» (Рисунок 10).

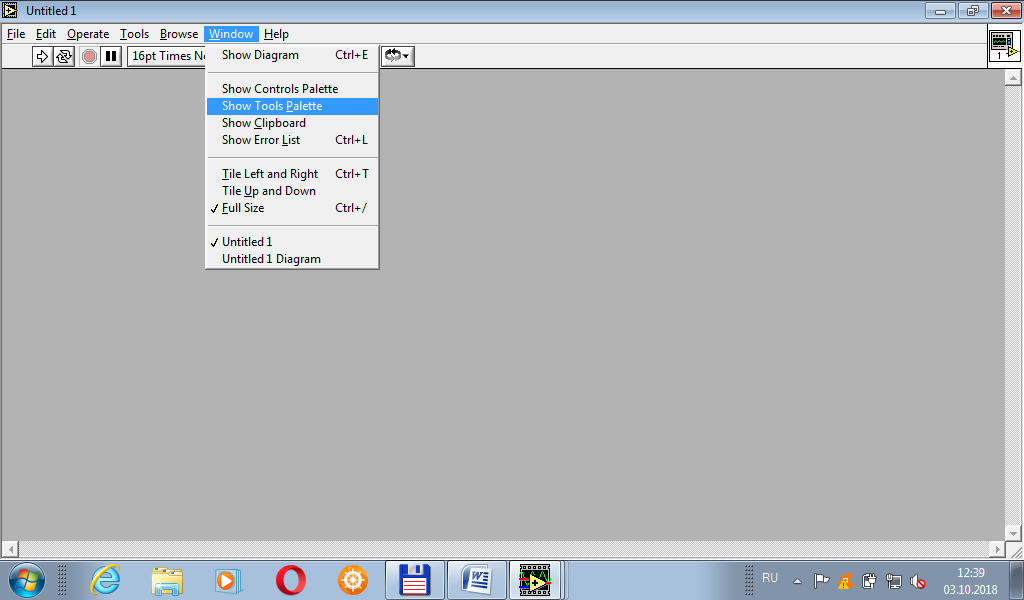


Рисунок 10 – Переход по меню к отображению панели основных инструментов

В результате выбора указанного пункта меню над передней панелью отобразится структура, представленная на Рисунке 11.

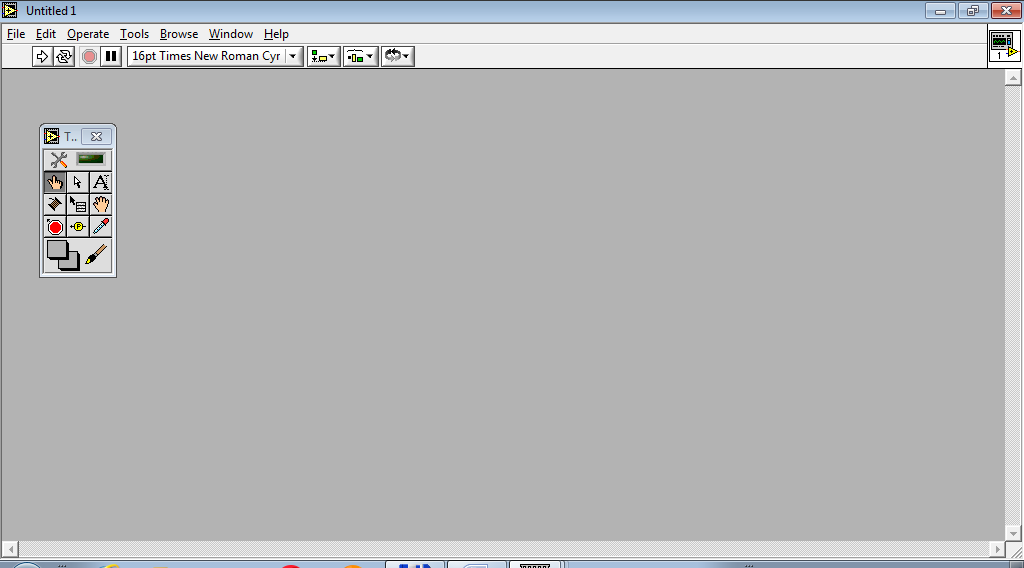


Рисунок 11 – Отображение панели основных инструментов

Панель основных инструментов является общей как для передней панели, так и для блок-диаграммы. К настоящему моменту без рассмотрения осталась лишь одна панель из числа часто используемых – панель функций среды. Она доступна для вызова только в режиме составления блок-диаграммы и вызывается через меню по следующему пути «*Window > Show Functions Palette*» (Рисунок 12).

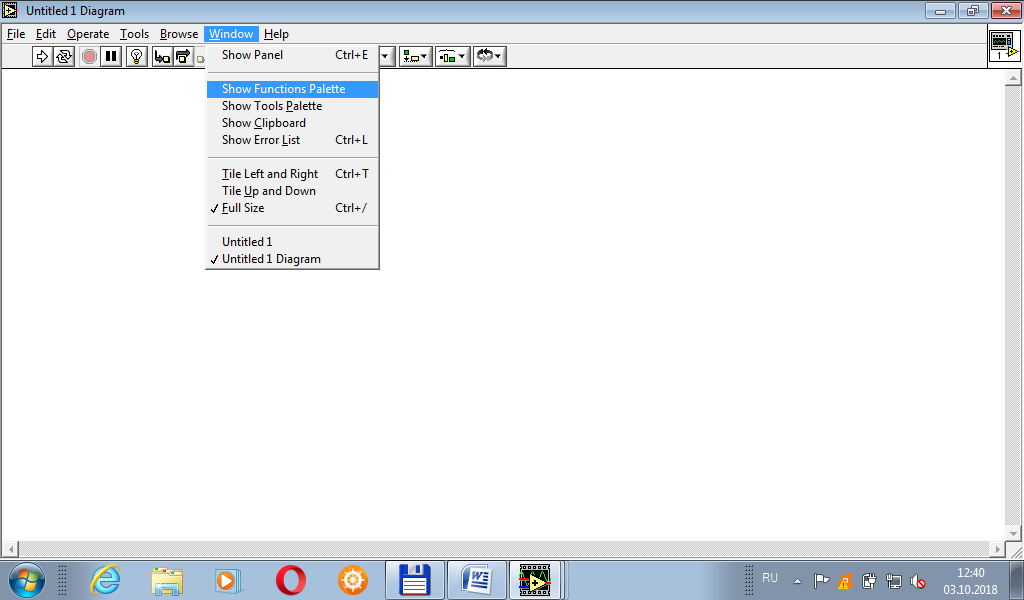


Рисунок 12 – Переход по меню к отображению панели функций среды

В результате выбора указанного пункта меню над блок-диаграммой отобразится структура, представленная на Рисунке 13.

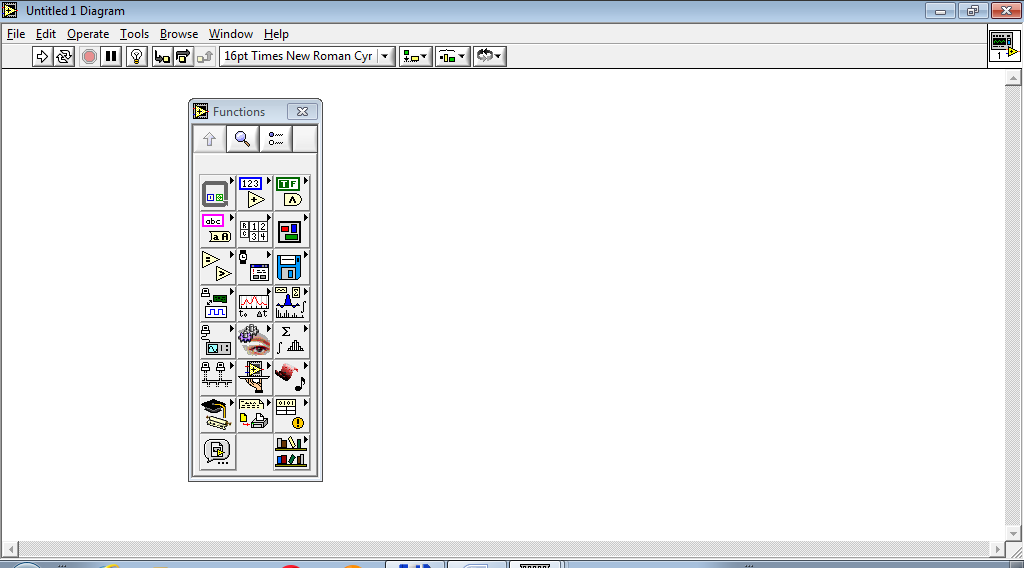


Рисунок 13 – Отображение панели функций среды

После вызова всех упомянутых панелей считается, что пакет прикладных программ *National Instruments LabView* готов к нормальному взаимодействую по части разработки нового виртуального прибора.

**Организация однозначной связи между численными элементами управления**

К настоящему моменту организован доступ ко всем необходимым компонентам среды, позволяющий создать простой виртуальный прибор. Нет ничего проще, чем задать однозначное численное соответствие между выходом и входом. Решим эту задачу.

Перейдём для начата в раздел численных элементов управления (Рисунок 14).

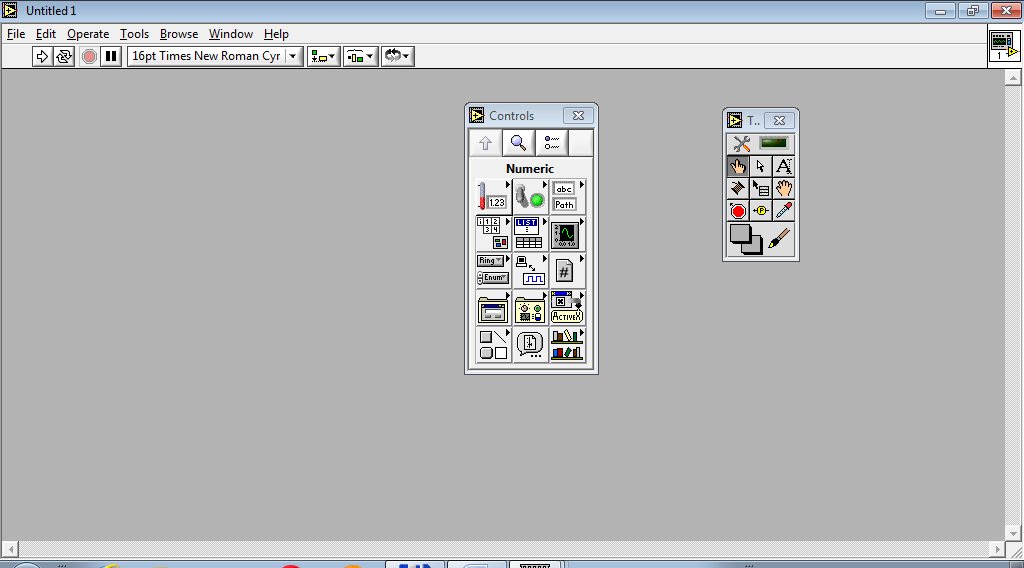


Рисунок 14 – Подсветка в панели элементов управления раздела численных элементов (*Numeric*)

Там выберем числовой контроллер (Рисунок 15), необходимый для ввода каких-либо значений в программу, написанную на графическом языке программирования.

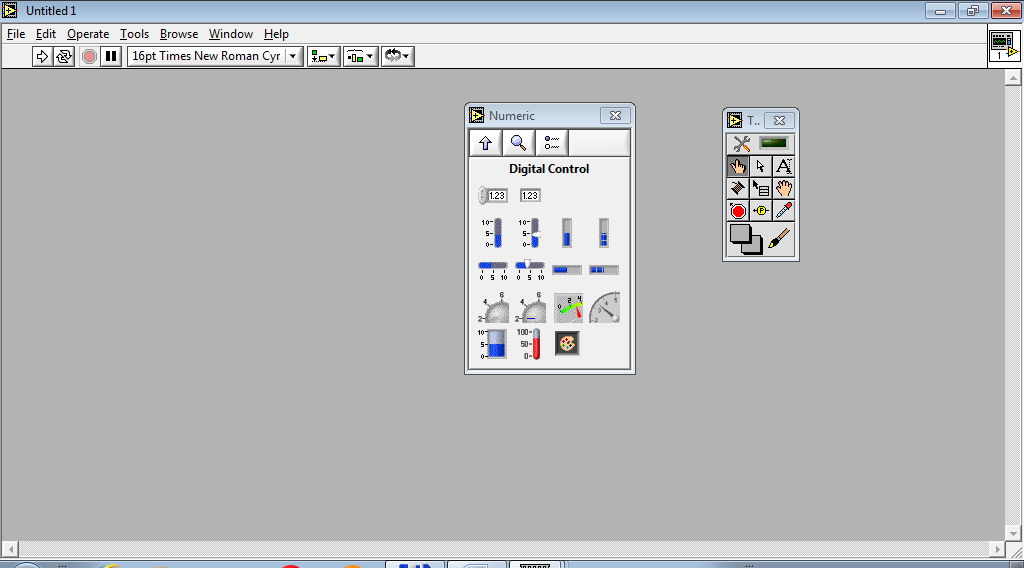


Рисунок 15 – Содержимое раздела численных элементов (*Numeric*) с подсветкой контроллера (*Digital Control*)

И разместим выбранный контроллер на передней панели виртуального прибора (Рисунок 16). Аналогичным способом на передней панели может быть размещено сколько угодно таких же численных контроллеров. Одновременно с размещением на панели образа элемента, на блок-диаграмме размечается его представление в виде графической переменной (это будет показано позже).

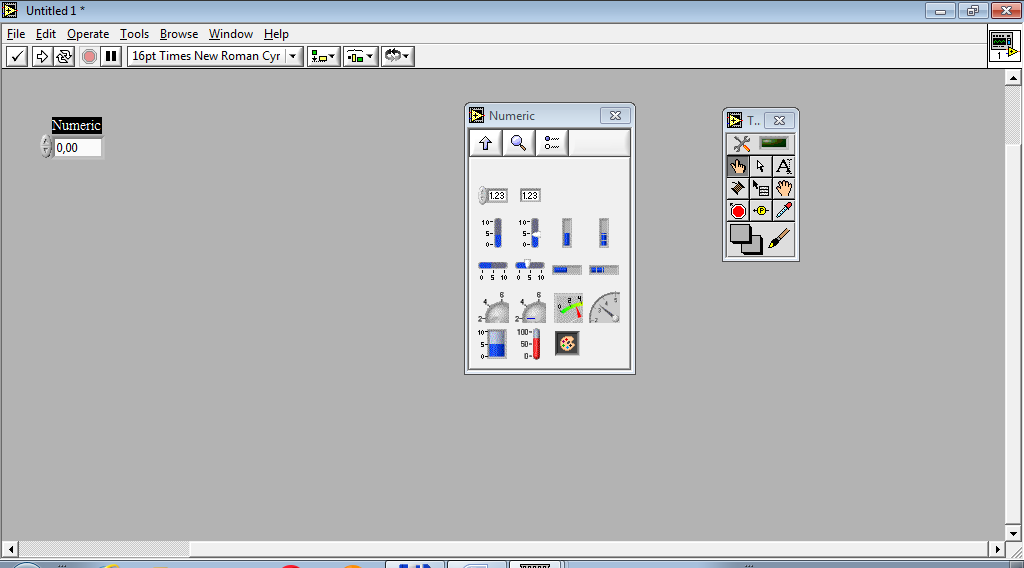


Рисунок 16 – Размещение контроллера (*Digital Control*) на передней панели

На панели раздел численных элементов нами не закрывался, потому выберем в нём теперь соседний элемент – численных индикатор (Рисунок 17), необходимый для вывода результирующей или промежуточной расчётной информации.

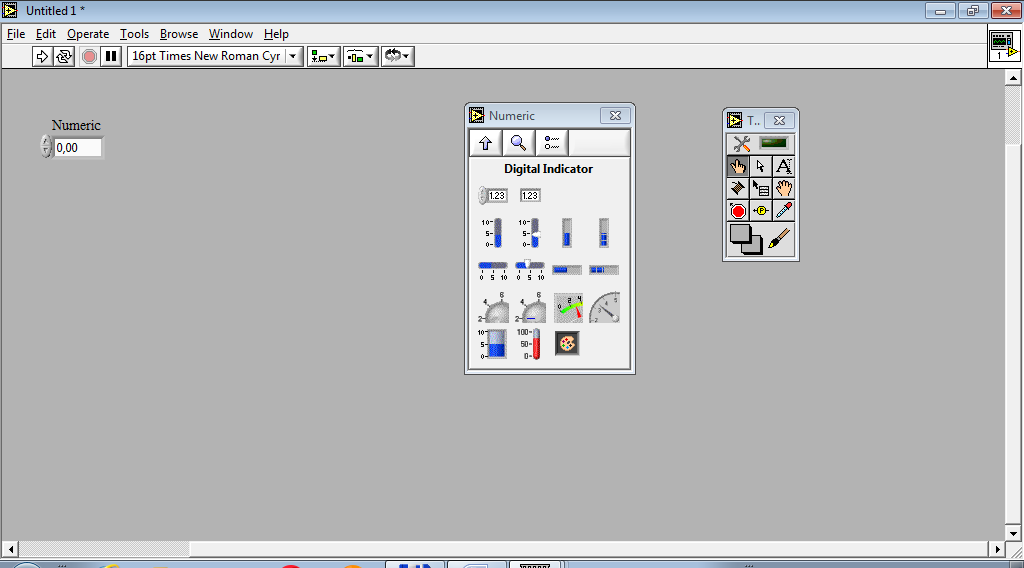


Рисунок 17 – Содержимое раздела численных элементов (*Numeric*) с подсветкой индикатора (*Digital Indicator*)

Разместим выбранный индикатор на передней панели рядом с ранее размещённым контроллером (Рисунок 18).

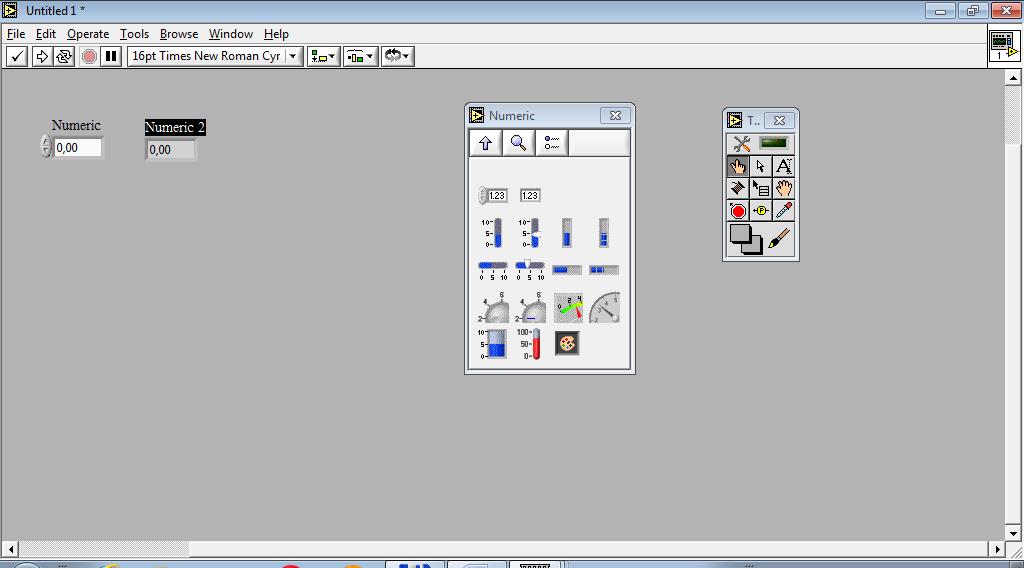


Рисунок 18 – Размещение индикатора (*Digital Indicator*) на передней панели

После щелчка в свободной области передней панели снимаем выделение с последнего размещённого на графическом пользовательском интерфейсе компонента и в этот момент считаем заготовку/болванку для виртуального прибора выполненной. Согласно одному из правил вежливости программиста, переменным необходимо давать значащие имена, которые бы позволили лицу, намеренному или поставленному продолжить написание ранее составленного кода, быстрее в этом коде разобраться. В *LabView*, что приятно, имена переменных однозначно связаны с именами элементов на передней панели, потому проявляя заботу о конечном пользователе одновременно получается заботиться и о программисте. Для выдачи значащих имён интерфейсным элементам управления используется инструмент редактирования текста (Рисунок 19).

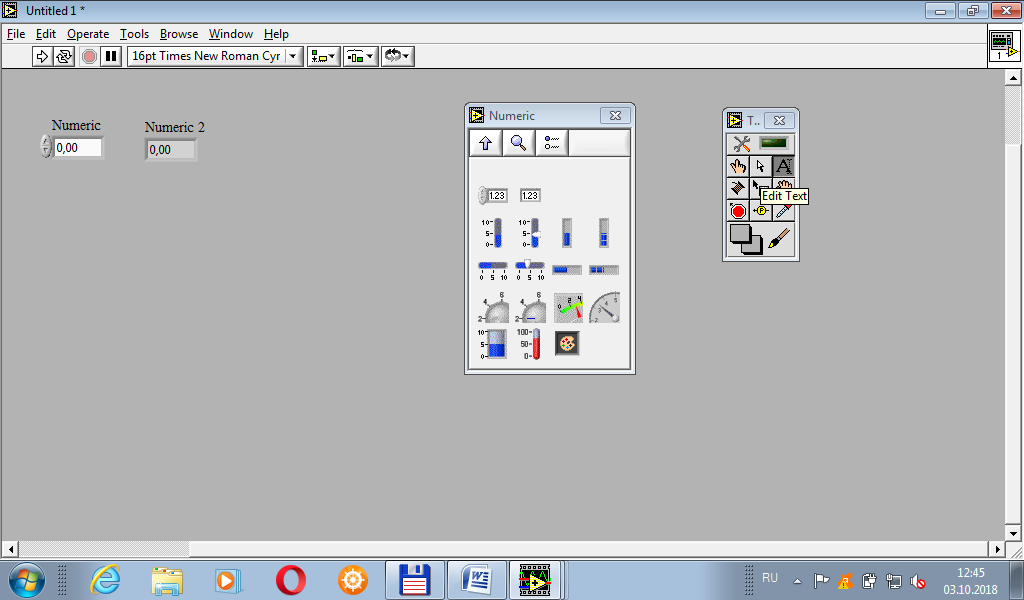


Рисунок 19 – Инструмент для редактирования текста

Программный продукт, не являющийся дружественным для пользователя, является непригодным для использования. Итого, руководствуясь вышесказанным, назовём контроллер входом, а индикатор – выходом (можно назвать синонимично вводом и выводом, соответственно). Результат показан на Рисунке 20.

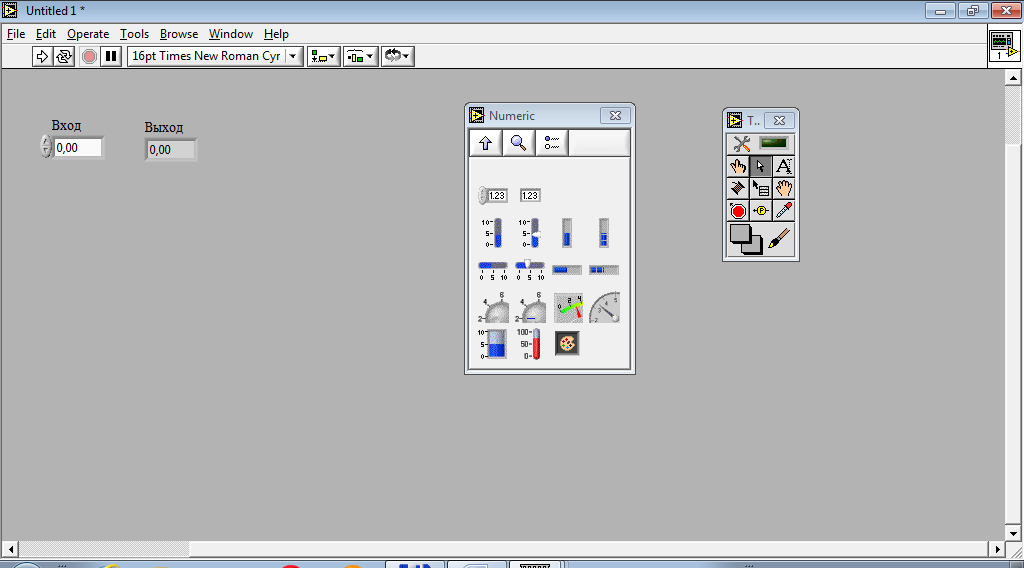


Рисунок 20 – Присвоение значащих имён переменным, связанным с контроллером и индикатором

Если с эстетической точки зрения начинает казаться, что элементы расположены криво, можно произвести их выравнивание относительно друг друга, предварительно выделив их и выбрав одну из операций выравнивания, как это показано на Рисунке 21.

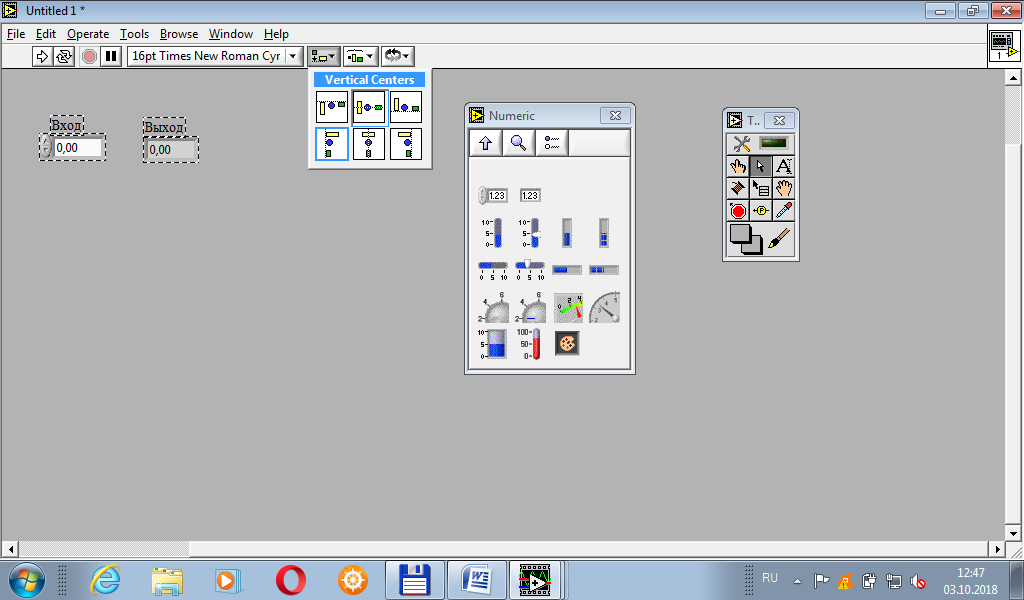


Рисунок 21 – Выравнивание контроллера и индикатора относительно друг друга по общей для них центральной и горизонтальной оси

После наведения порядка на передней панели переходим к кодированию, на блок-диаграмму (Рисунок 22).

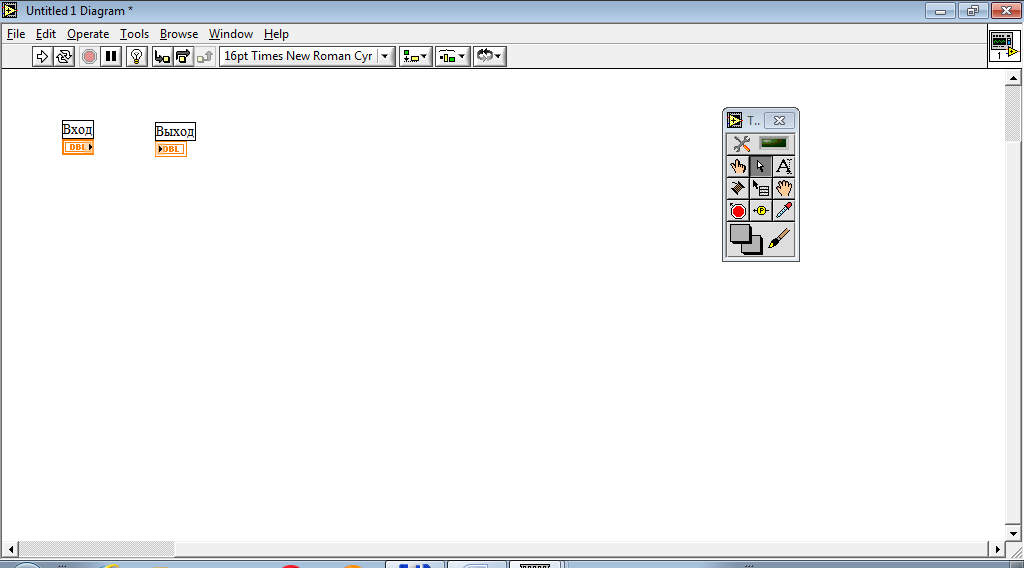


Рисунок 22 – Отображение тех же контроллера и индикатора на блок-диаграмме

Хорошо видно, что графические переменные однозначно соответствуют графическим элементам, размещённым ранее на передней панели. Решим поставленную простую задачу – соединим переменные между собой. Для чего выберем соединительный провод (Рисунок 23) и последовательно нажмём левой кнопкой мыши на мигающей правой границе входа и затем на мигающей левой границе выхода.

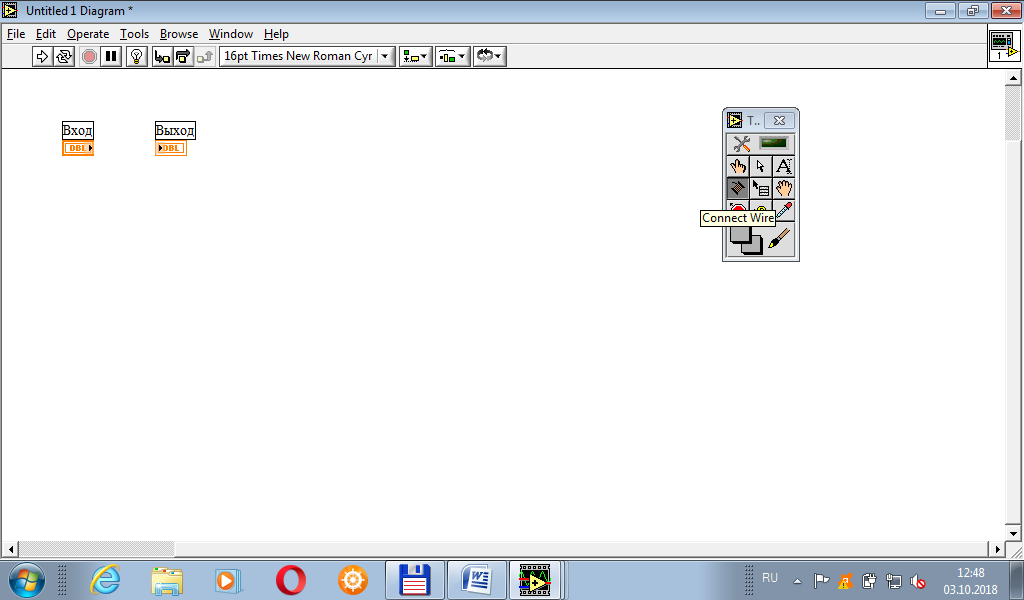


Рисунок 23 – Выбор инструмента соединительный провод (*Connect Wire*)

Результат соединения можно наблюдать на Рисунке 24.

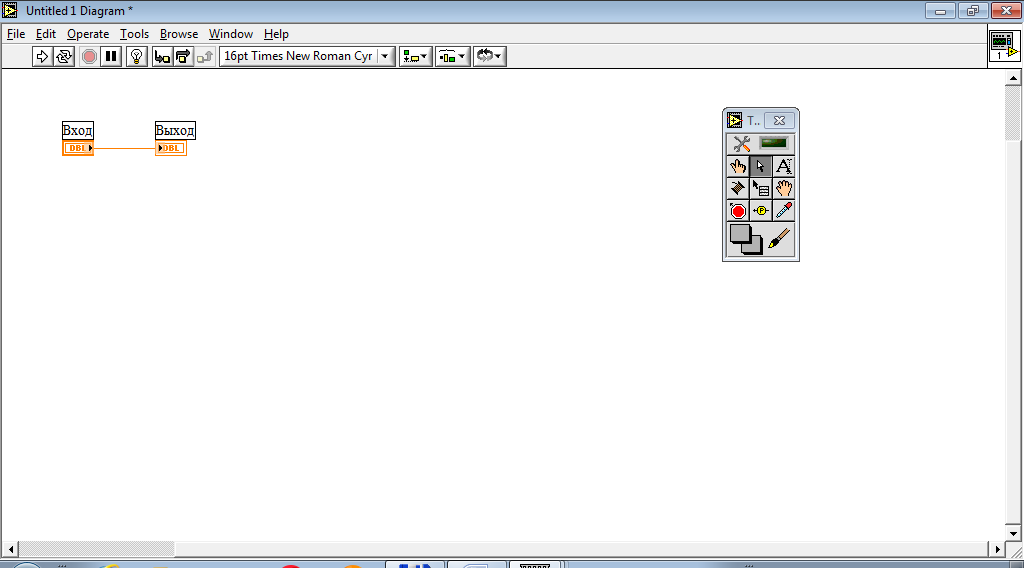


Рисунок 24 – Результат соединения контроллера и индикатора

Для обеспечения возможности изменения значения в контроллере, сменим соединительный провод на инструмент изменения значений, выглядящий как указательный палец правой руки (Рисунок 25).

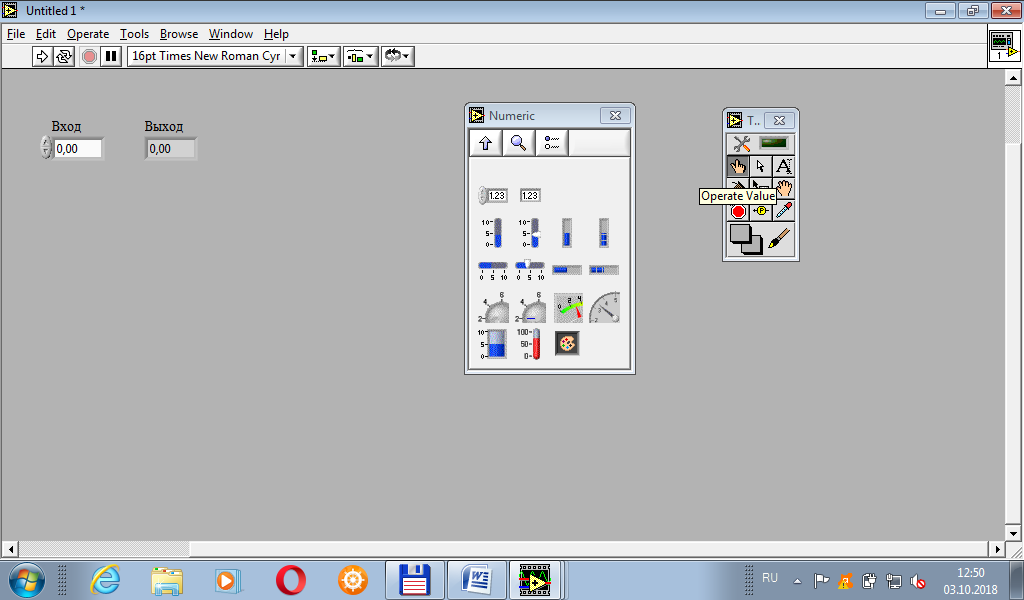


Рисунок 25 – Выбор инструмента изменения значений элементов (*Operate Value*)

С использованием данного инструмента увеличим значение на входе до «5,00», как показано на Рисунке 26 и запустим программу в режиме однократного исполнения.

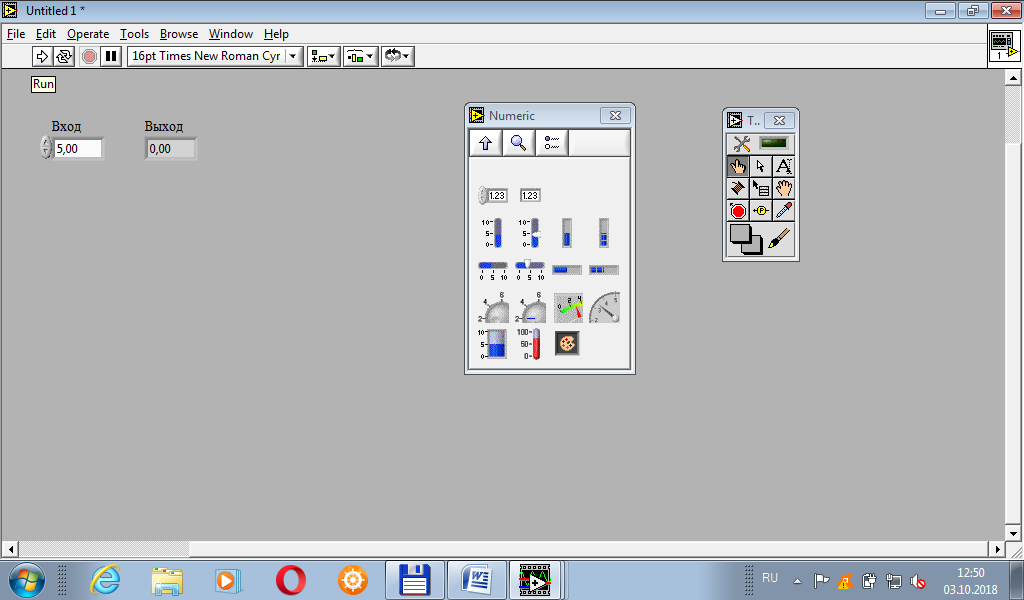


Рисунок 26 – Изменение значения, хранимого в контроллере и переход к разовому запуску программы на исполнения

Хорошо видно (Рисуное 27), что результатом работы программы становится настройка однозначного соответствия выхода со входом.

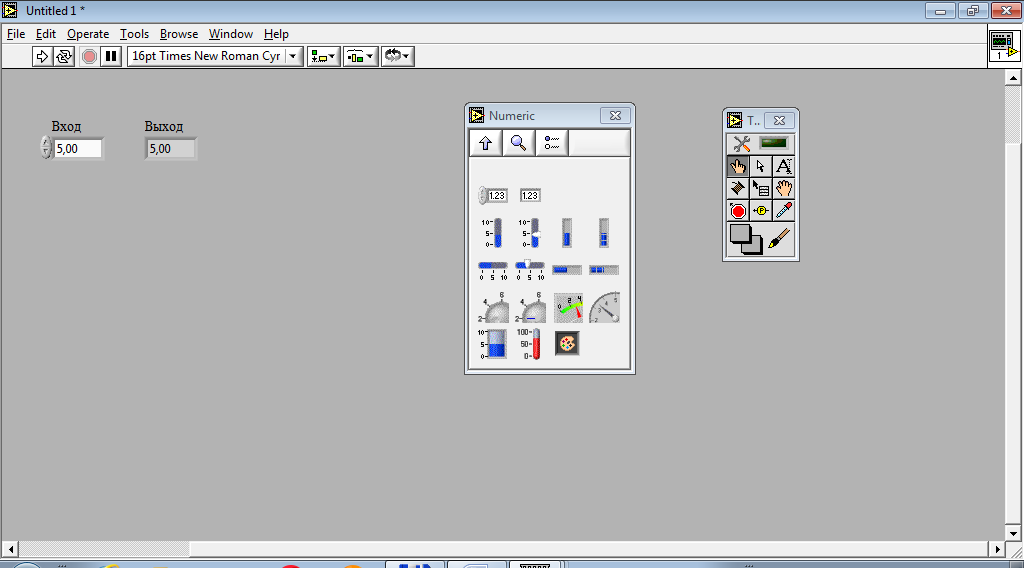


Рисунок 27 – Результат разового запуска программы на исполнение

Попробуйте проделать самостоятельно серию экспериментов, изменяя значения на входе и запуская программу в режиме однократного исполнения, а также запустив программу в режиме циклического исполнения и меняя значения на входе.

**Организация связи между численным и логическим элементами**

Далее рассмотрено решение несколько иной задачи, в которой проверяется логика наличия соответствия между входной информацией и некоторой наперёд заданной константой (пусть, например, её значение равно «5»). Соответствие, как известно, может быть, а также его может и не быть. Так наш ответ на вход должен получиться логическим. Это означает, что мы имеем возможность модифицировать ранее созданную программу виртуального прибора, исключив из неё численный индикатор и заменив его логическим индикатором. Все логические элементы содержатся в разделе логических или булевских (Рисунок 28).

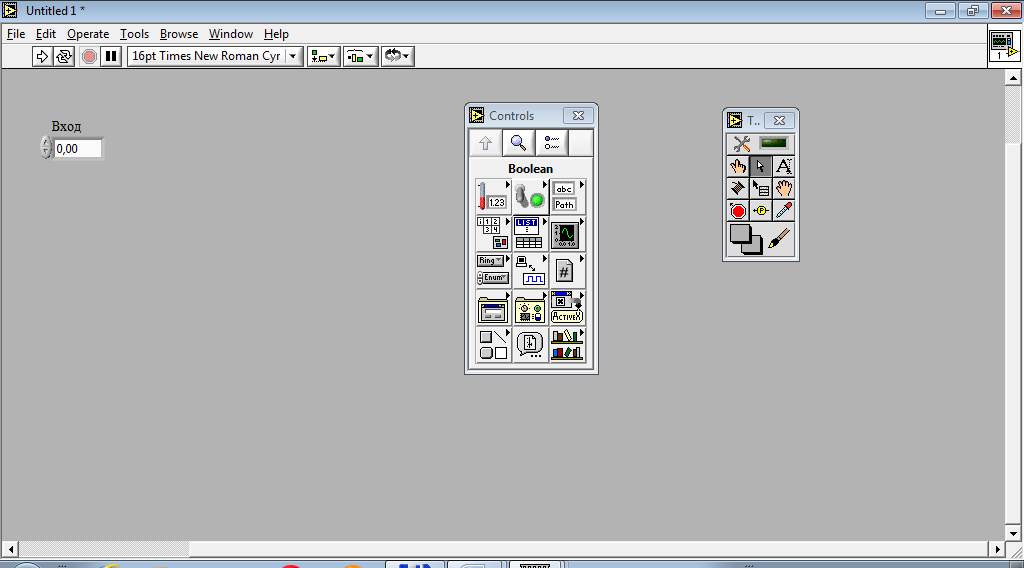


Рисунок 28 – Подсветка в панели элементов управления раздела логических элементов (*Boolean*)

В перечне логических элементов существуют два наиболее примечательных индикатора: квадратный (это только название и на деле – элемент прямоугольной формы) светодиод и круглый светодиод. По смысловой нагрузке эти элементы равноценны, потому размещаются на графическом пользовательском интерфейсе по усмотрению автора виртуального прибора (Рисунок 29).

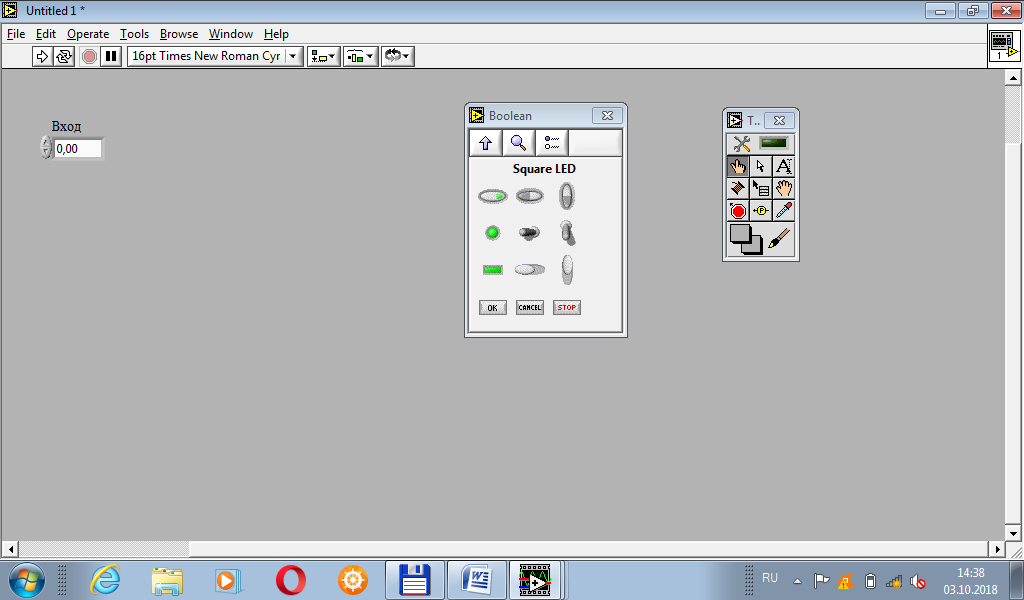


Рисунок 29 – Содержимое раздела логических элементов (*Boolean*) с подсветкой логического индикатора типа «прямоугольный светодиод» (*Square LED*)

Как правило, круглые элементы менее строги к вниманию оператора, прямоугольные элементы – более требовательны к вниманию (правило из области психологии человека). В то время как Информатика – строгая практическая дисциплина, требующая не менее строгой отчётности, применим к графическому пользовательскому интерфейсу логический индикатор прямоугольной формы (Рисунок 30).

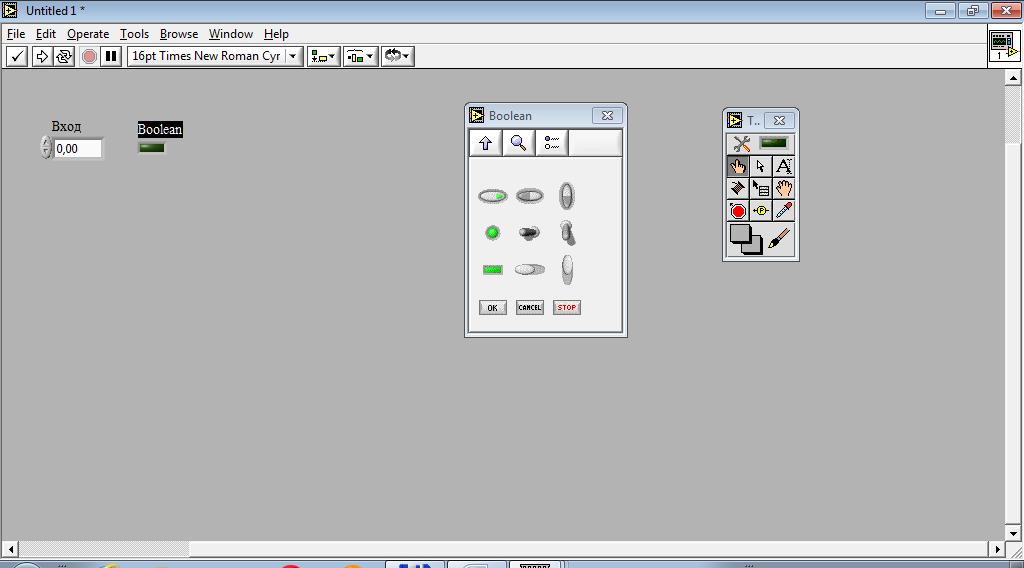


Рисунок 30 – Размещение прямоугольного светодиода на передней панели

Как и для случая с численным индикатором не стоит забывать о существовании правил вежливости программиста, а также правил обеспечения дружественности графического пользовательского интерфейса, потому введём значащее имя для нового логического индикатора (Рисунок 31).

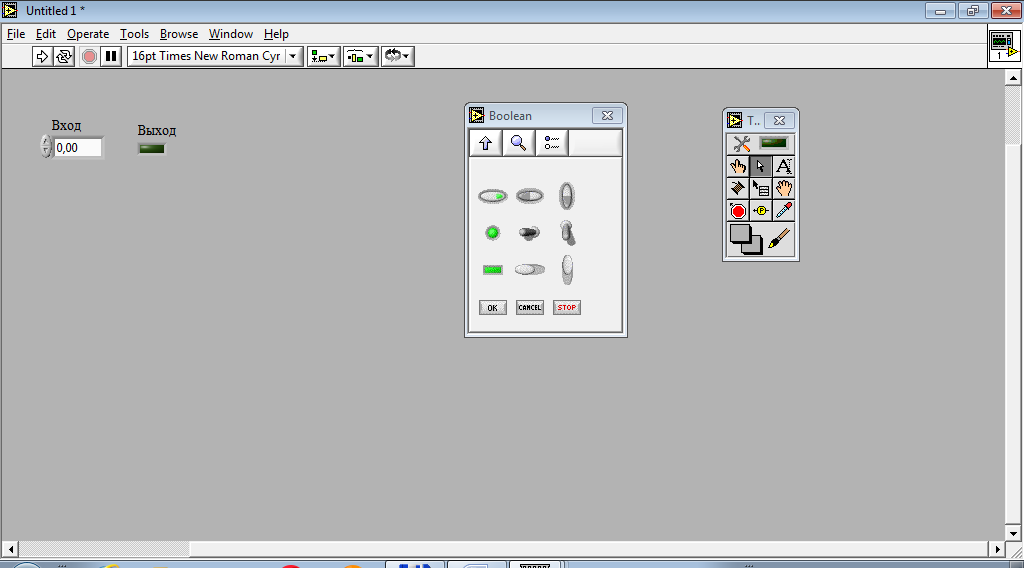


Рисунок 31 – Формирование значащего имени для переменной, связанной с логическим индикатором

Далее потребуется ещё раз обратиться к разделу численных элементов, но уже в рамках блок-диаграммы создаваемого виртуального прибора (Рисунок 32).

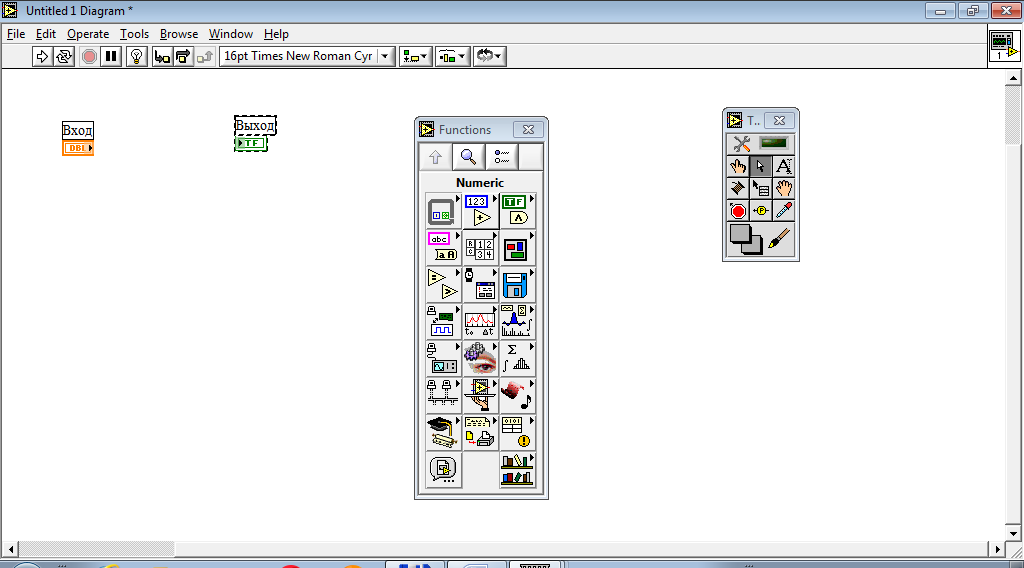


Рисунок 32 – Подсветка в панели функций раздела численных элементов (*Numeric*)

Из раздела понадобится численная константа, расположенная в левом нижнем углу окна с перечнем доступных для использования элементов данного раздела (Рисунок 33).

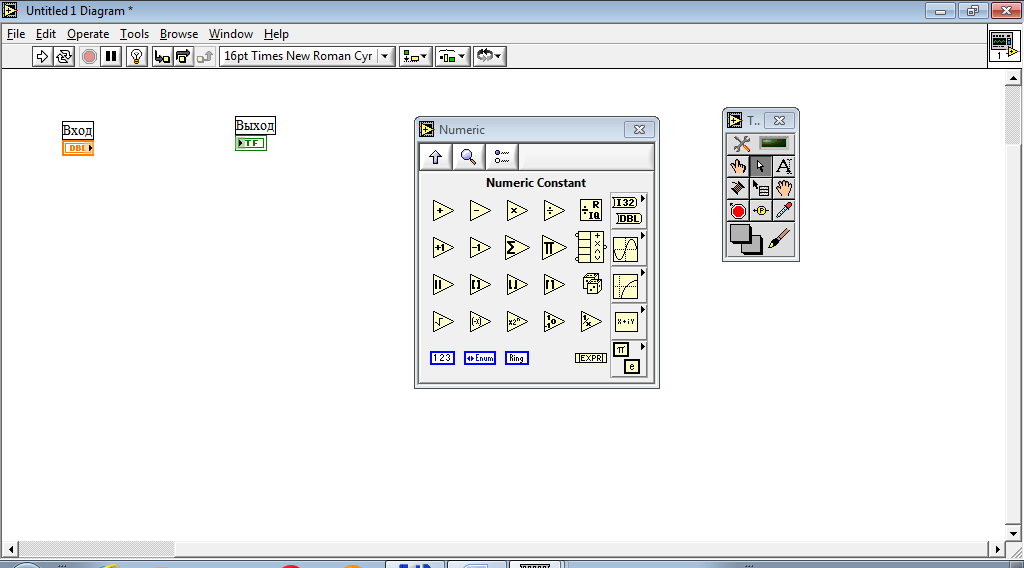


Рисунок 33 – Содержимое раздела численных функций (*Numeric*) с подсветкой численной константы (*Numeric Constant*)

Напомним, что любая константа характеризуется типом и значением, причём значение константы определяет её тип. Таким образом, пока в рамке указано целое число – это будет целочисленная константа (Рисунок 34), как только там появится значение с плавающей запятой – это будет вещественная константа. Причём, существует возможность задать константу «*NaN*» (*Not a Number* – не числовое значение), позволяющую в отдельных случаях отлавливать ошибки.

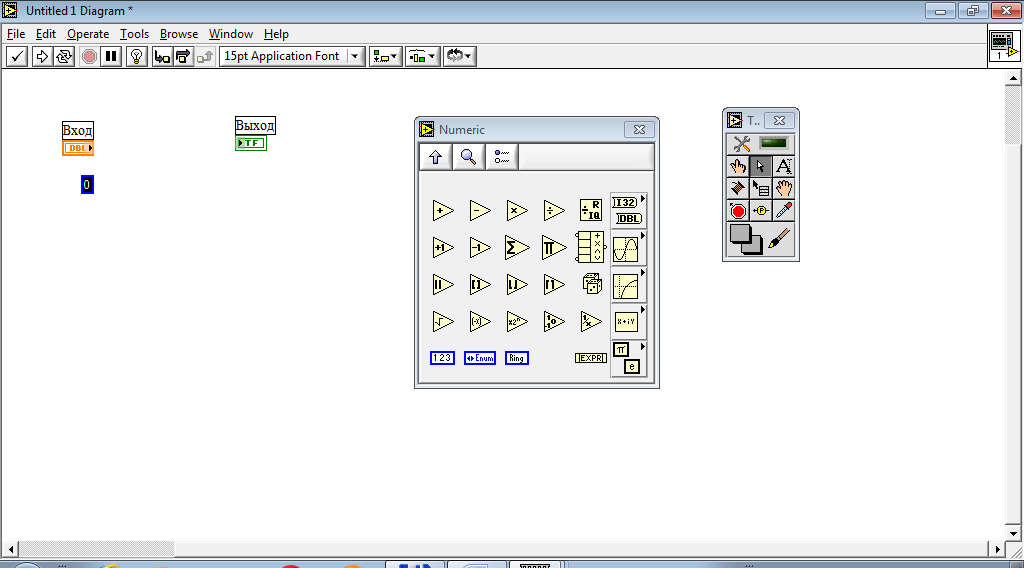


Рисунок 34 – Размещение целочисленной константы на блок-диаграмме

Не будем выходить за область целых чисел, потому в качестве константы в данном примере выставим значение равное «5» (Рисунок 35).

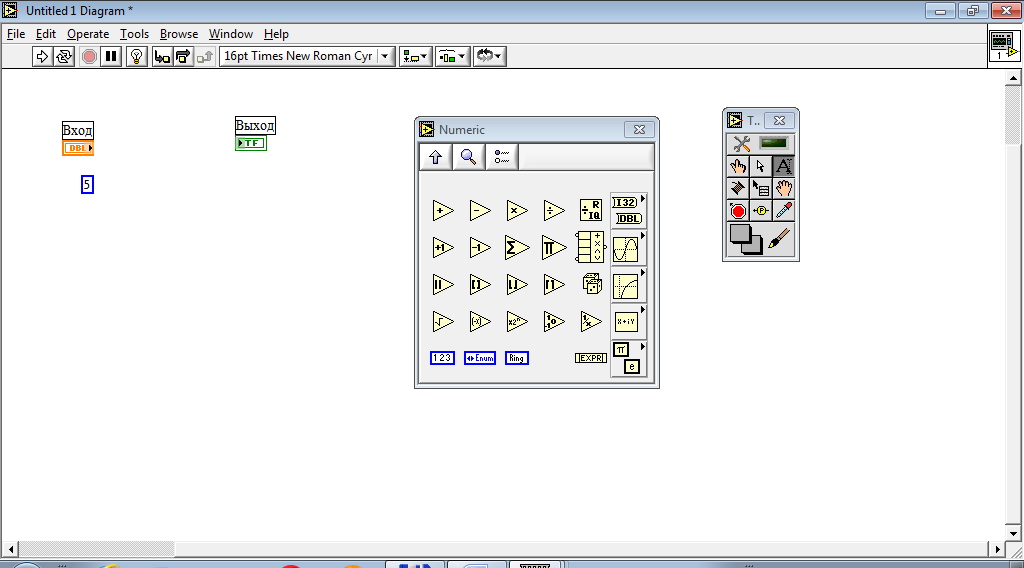


Рисунок 35 – Изменение значения целочисленной константы с «0» на «5»

После этого следует перейти к сравнению значения переменной, задаваемому численным контроллером, с константой, для чего обратиться в панели функций области блок-диаграммы к разделу элементов сравнения (Рисунок 36).

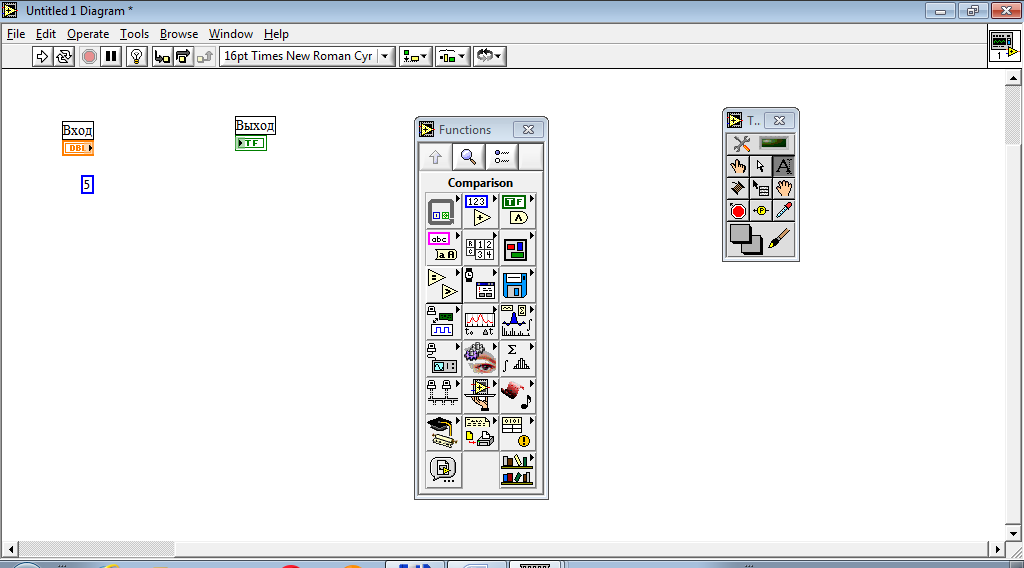


Рисунок 36 – Подсветка в панели функций раздела элементов сравнения (*Comparison*)

Для всех случаев, где сравнение требуется со значениями, отличными от нуля требуются на входе два операнда, для случаев сравнения с нулём – один операнд. Далее рассматривается простейший случай сравнения на точное соответствие элементов – их эквивалентность (Рисунок 37). Соответствующий элемент располагается в левом верхнем углу окна, содержащего элементы, доступные для размещения на блок-диаграмме из этого раздела.

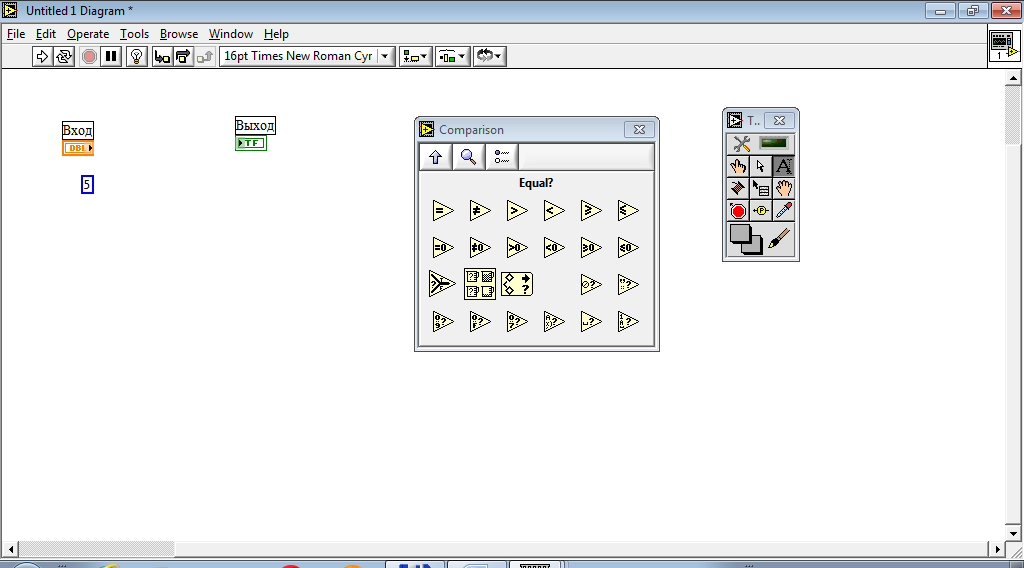


Рисунок 37 – Содержимое раздела функций сравнения (*Comparison*) с подсветкой оператора однозначного соответствия / проверки на эквивалентность (*Equal?*)

Итого размещаем элемент на блок-диаграмме и видим, что на ней теперь всего достаточное, остаётся лишь настроить связи между элементами (Рисунок 38).

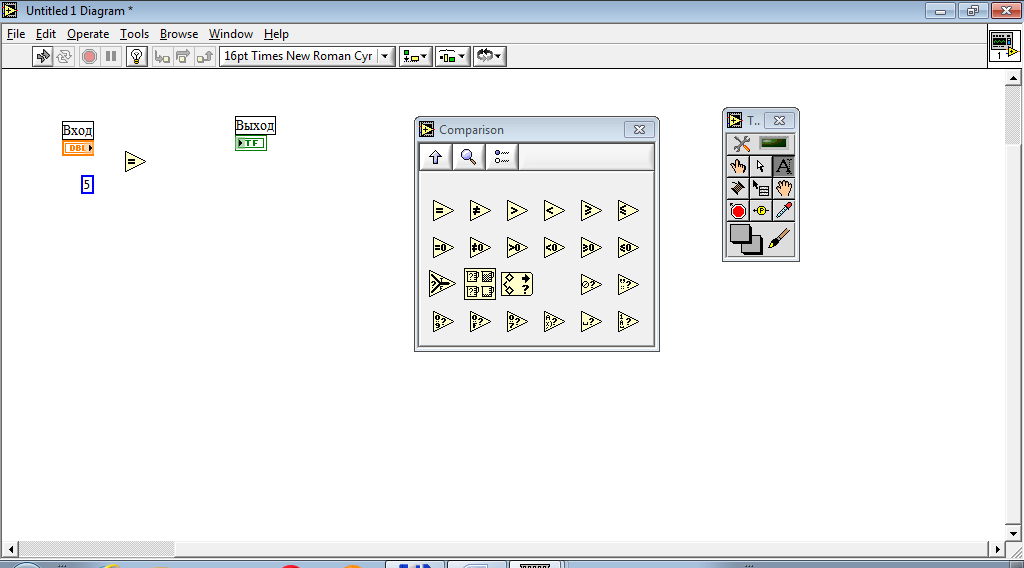


Рисунок 38 – Размещение оператора однозначного соответствия на блок-диаграмме

Не стоит забывать о правилах вежливости программиста: написанный код должен обладать читаемостью. Для графического кода в этом смысле существует куда больше дополнений, ограничивающих понятие читаемости. Так, например, необходимо стремиться к минимизации изломов линий связи – они отвлекают внимание на себя, а также стремиться к минимизации занимаемого графическим кодом пространства. Последнее пришло, скорее, из электроники, где при проектировании печатных плат стараются минимизировать количество использованного проводника, поскольку каждый сантиметр потраченного медного провода повышает стоимость разработки. В работе, естественно, создаётся только модель, но о минимизации затрат на токопроводящий материал нужно привыкать думать и помнить уже на данном этапе (Рисунок 39). Если выработается привычка минимизации занимаемого пространства элементами схемы в рамках Информатики – это значимый шаг для решения практических инженерных задач в будущем.

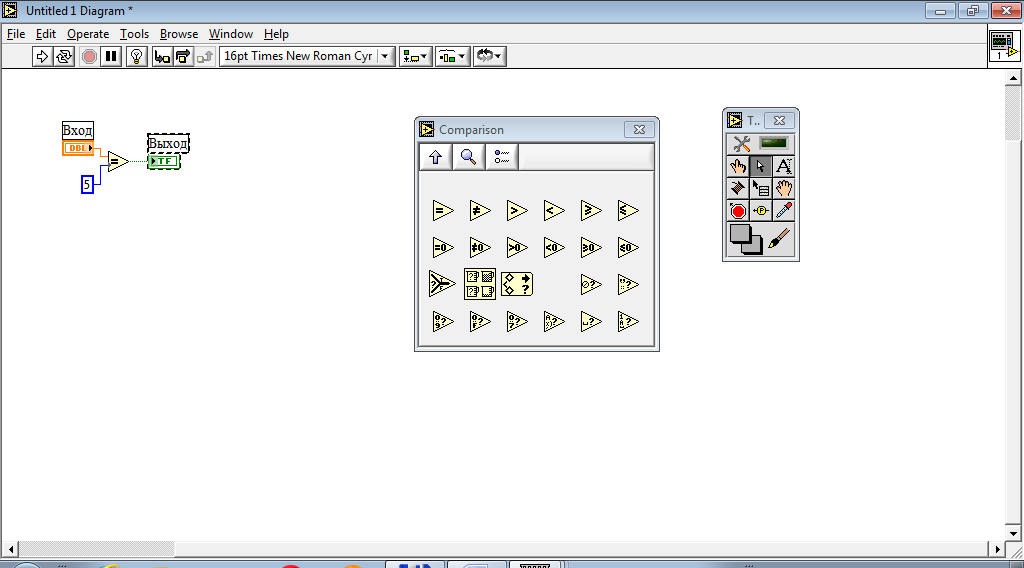


Рисунок 39 – Соединение размещённых элементов между собой

Протестируем созданный, очень простой виртуальный прибор. В алгоритме имеет место разветвляющийся вычислительный процесс, в котором существуют лишь два исхода, потому для доказательства работоспособности нового программного обеспечения необходимо подобрать и отработать два тестовых примера. Подадим для начала на вход численного контроллера виртуального прибора значение равное «2». В этих условиях логическое выражение отвечает ложным результатом и лампочка не загорается (Рисунок 40).

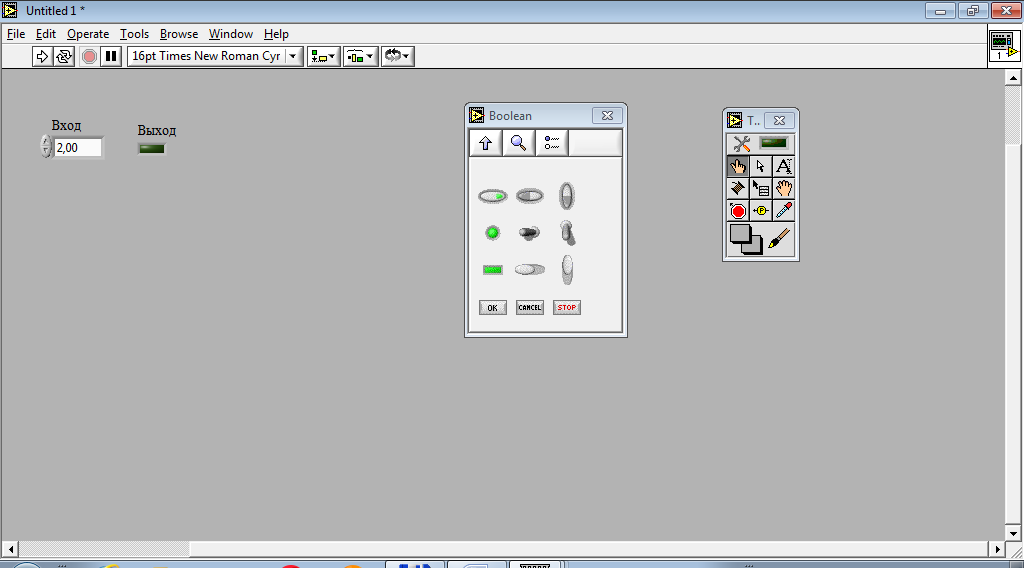


Рисунок 40 – Отклик лампы на значение «2,00», поданное на вход

Затем подадим на вход численного контроллера виртуального прибора значение, равное «5». Здесь логическое выражение отвечает истинным результатом – лампочка загорается.

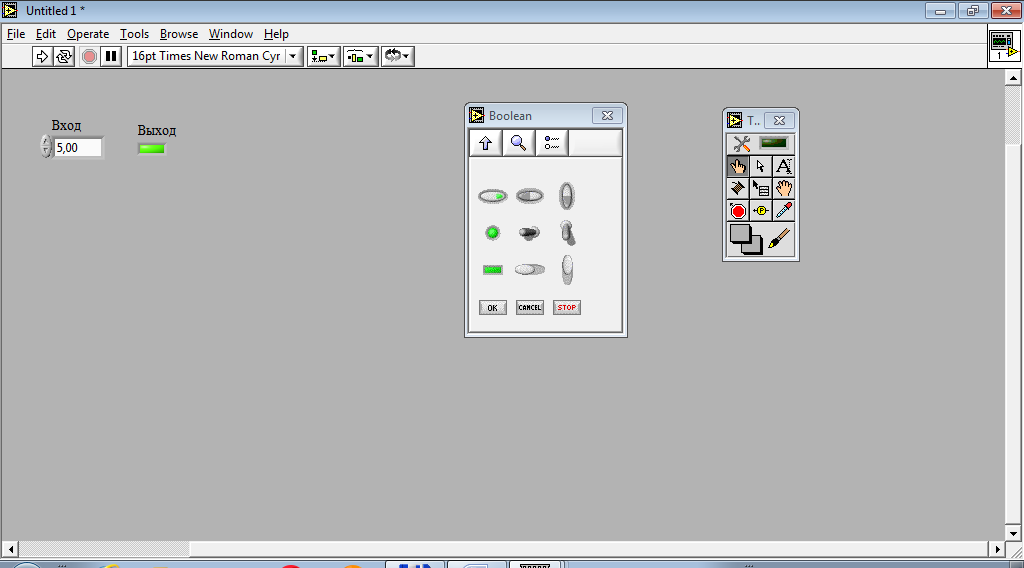


Рисунок 41 – Отклик лампы на значение «5,00», поданное на вход

Рассмотренных выше примеров достаточно для предоставления обучающимся базовых умений и навыков для решения поставленной задачи.

----------------------------------------------------------------------------------------------------

**Цель работы (одна из возможных формулировок)**: закрепление навыков работы в пакете прикладных программ *National Instruments* *LabView*, развитие навыков алгоритмизации и графического программирования.

**Индивидуальная часть работы (переменные):**

В дополнительном файле *\*.vi* создать развитие структуры «Простого Калькулятора» посредством добавления в неё элементов ввода и вывода информации, а также индикации (в случае необходимости), достаточных для демонстрации результата расчёта выражения, заданного по варианту.

Без острой на то необходимости графический код не усложнять! Например, операция возведения в степень не должна вызывать потребности у автора виртуального прибора в использовании специальной функции возведения числа в некоторую степень *N*.

|  |  |  |
| --- | --- | --- |
| **Вар.** | **Выражение** | **Логический индикатор** |
| **1** | (a + b + c + d) \* (– a – b – c – d) | Не требуется |
| **2** | (a + b + c)2 | Не требуется |
| **3** | (a – b + c – d) \* (– a + b – c – d) | Не требуется |
| **4** | (b / a) – (d / c) | Требуется |
| **5** | (a + (b – d) \* c)2 | Не требуется |
| **6** | (d / (b + a)) – (c / (d + a)) | Требуется |
| **7** | 3 \* a + b2 – c + 5 \* d3 | Не требуется |
| **8** | –((b + c) / (a + d)) | Требуется |
| **9** | (a + (b \* c) / d) / (a / (b \* d) – c) | Требуется |
| **10** | ((b + 2) \* (a + 4) / (d + 6) \* (c + 8)) | Требуется |
| **11** | –((b + c) \* (a + d) / c2) | Требуется |
| **12** | 10 \* a + 15 \* b + 20 \* c + 25 \* d | Не требуется |
| **13** | 7 \* (a + b + c + d) \* (– a – 7 \* b – c – d) | Не требуется |
| **14** | (a + b + c + d) / (– a – b – c – d) | Требуется |
| **15** | (a + b + c + d)3 | Не требуется |
| **16** | (a – b + c – d) / (– a + b – c – d) | Требуется |
| **17** | (a / b) + (c / d) | Требуется |
| **18** | (a \* (b – d) + c)2 | Не требуется |
| **19** | (a / (c + a)) + (b / (d \* a)) | Требуется |
| **20** | ((b / a) / (d / c)) – ((a / b) / (c / d)) | Требуется |
| **21** | 5 \* a + b2 + c – 3 \* d3 | Не требуется |
| **22** | –((a + b) / (d – c)) | Требуется |
| **23** | (a – (b \* c) / d) \* (a / (b \* d) + c) | Требуется |
| **24** | 10 \* d – 5 \* c + 20 \* a – 15 \* b | Не требуется |
| **25** | –((b – c) \* (a + d) / (c + 4)) | Требуется |
| **26** | 3 \* (a + c + d) \* (–7 \* b – d) | Не требуется |
| **27** | (a + b + c + d)4 | Не требуется |
| **28** | (a \* (c + a)) / (b \* (d – a)) | Требуется |
| **29** | (a / b)2 + (c / d)2 | Требуется |
| **30** | (a2 + b2 + c2) | Не требуется |

**Индивидуальная часть работы (константы):**

В ещё одном дополнительном файле *\*.vi* создать независимый от двух предыдущих виртуальных приборов иной виртуальный прибор, который смоделировал бы правильный ответ на сложное выражение, заданное по варианту.

Графический пользовательский интерфейс должен содержать в своём составе в удобной для чтения форме исходное сложное выражение, заданное по варианту, а также индикатор, выдающий правильный ответ на упомянутое сложное выражение.

