**Задание 04.** В пакете прикладных программ *National Instruments LabView* создать виртуальный прибор, моделирующий поведение типовых сигнализаторов и индикаторов (заданных по вариантам).

На блок-диаграмме все функциональные, то есть значащие пересечения линий связи (узлы) обозначить точками. Для этого в меню перейти к настройкам «*Tools > Options…*», далее в выпадающем списке перейти к настройкам блок-диаграммы (*Block Diagram*) и выставить галочку напротив пункта «*Show dots at wire junctions*».

По итогам выполнения работы сдаются следующие файлы:

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

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

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:**

«*20190412. Задание 04ои. Иванов.docx*»

«*20190412. Задание 04ои. Иванов.vi*»

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

«*20190412. Задание 04ои. Иванов (1).docx*»

«*20190412. Задание 04ои. 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.

3. Составление блок-схемы алгоритма программы.

4. Подбор и расчёт тестовых примеров.

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

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

7. Расчёт тестовых примеров с использованием составленного программного обеспечения.

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*».

**Цель работы (одна из возможных формулировок)**: освоение навыков работы с локальными и глобальными переменными, а также структурой последовательного вычислительного процесса внутри системы, адаптированной под потоковые вычисления.

**Пример реализации последовательности:**

В данном разделе рассматривается работа с последовательностями (структура «Последовательность» / «*Sequence*») в пакете прикладных программ *National Instruments LabView*. Прежде, чем приступить к работе с последовательностями необходимо понять и освоить механизм работы с локальными переменными, поскольку именно через локальные переменные в *National Instruments LabView* возможно реализовать изменение состояний одних и тех же интерфейстных элементов.

Локальные переменные, связанные с интерфейсными элементами, вообще говоря, являются их образами («*Views*»). Формально у интерфейсных элементов управления может быть только одна линия связи, поступаяющая либо на вход, либо на выход. И для случаев подачи различных сигналов в различные моменты времени необходимо дублирование элементов связи с интерфейсными элементами управления на блок-диаграмме. Однако, при попытке продублировать интерфейсные элементы управления на блок-диаграмме и на передней панели возникают их дубликаты. Это совершенно не тот эффект, которого хотелось бы добиться в данной ситуации. Таким образом, дальнейшая работа ведётся через локальные переменные. Их можно насоздавать столько, сколько необходимо для нормальной работы виртуального прибора и это не вызовет появления дубликатов на передней панели.

Итак, для создания локальной переменной, связанной с интерфейсным элементом управления, необходимо в контекстном меню настраиваимого под смену состояний компонента перейти в раздел «*Create >*» («Создать >»), в котором выбрать пункт «*Local Variable*» («Локальная переменная»). Соответсвующий переход по меню представлен на Рисунке 1.

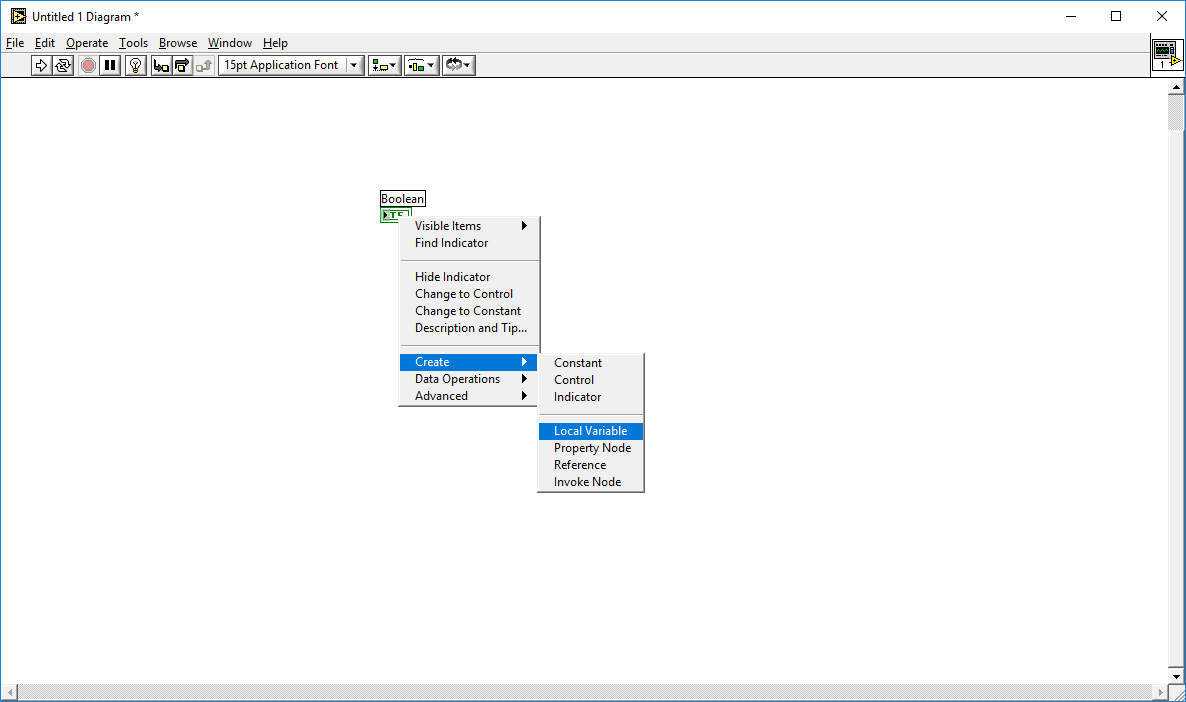


Рисунок 1 – Расположение пункта контекстного меню, ответственного за создание локальной переменной рассматриваемого интерфейсного элемента управления

По итогам выбора соответствующего пункта меню на блок-диаграмме появится прямоугольный блок с двойным обрамлением наименования интерфейсного элемента управления в цвете, соответствующем типу данных интерфейсного элемента управления (Рисунок 2).

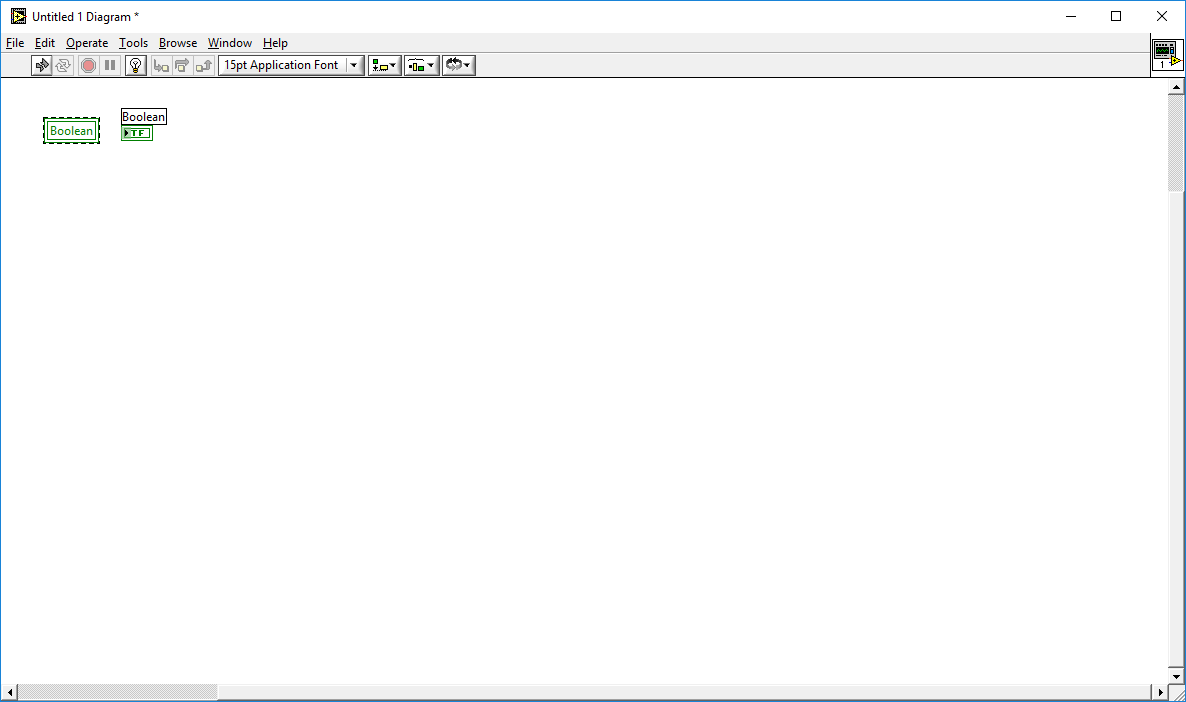


Рисунок 2 – Результат создания локальной переменной рассматриваемого интерфейсного элемента управления

Вместе с тем вычисления в последовательностях связаны с задержками по времени, потому необходимо знать о существовании соответствующих фуекций и возможностях работы с таковыми. На Рисунке 3 показано место расположения функционального блока задержки («*Wait*») вычислительного процесса на указанное количество миллисекунд, переданных на вход в виде константы или в виде значения переменной.

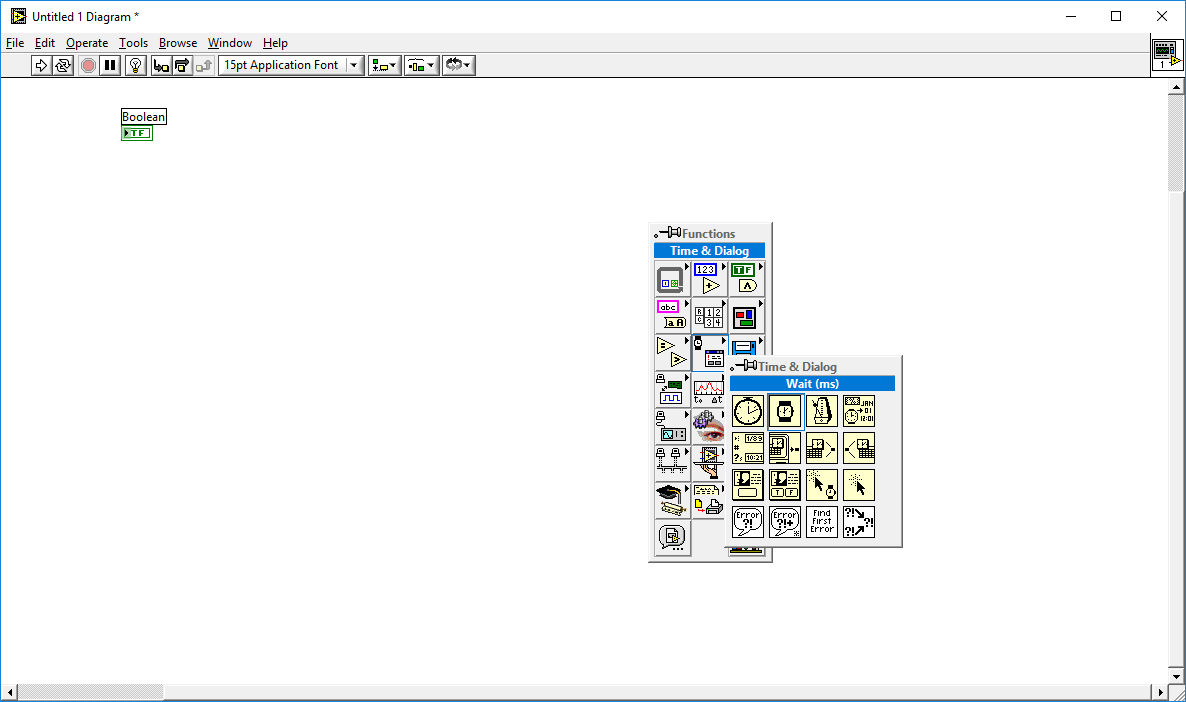


Рисунок 3 – Место расположения элемента для организации задержки вычислительного процесса на величину, заданную в миллисекундах

После рассмотрения указанных компонентов можно перейти к самим структурам последовательностей. Исходя из названия последовательности содержатся в разделе структур (Рисунок 4). Размещается такая структура на блок-диаграмме по аналогии с ранее изученными структурами, такими как: *Case*, *For*, *While*, *Event* и *Formula Node*.

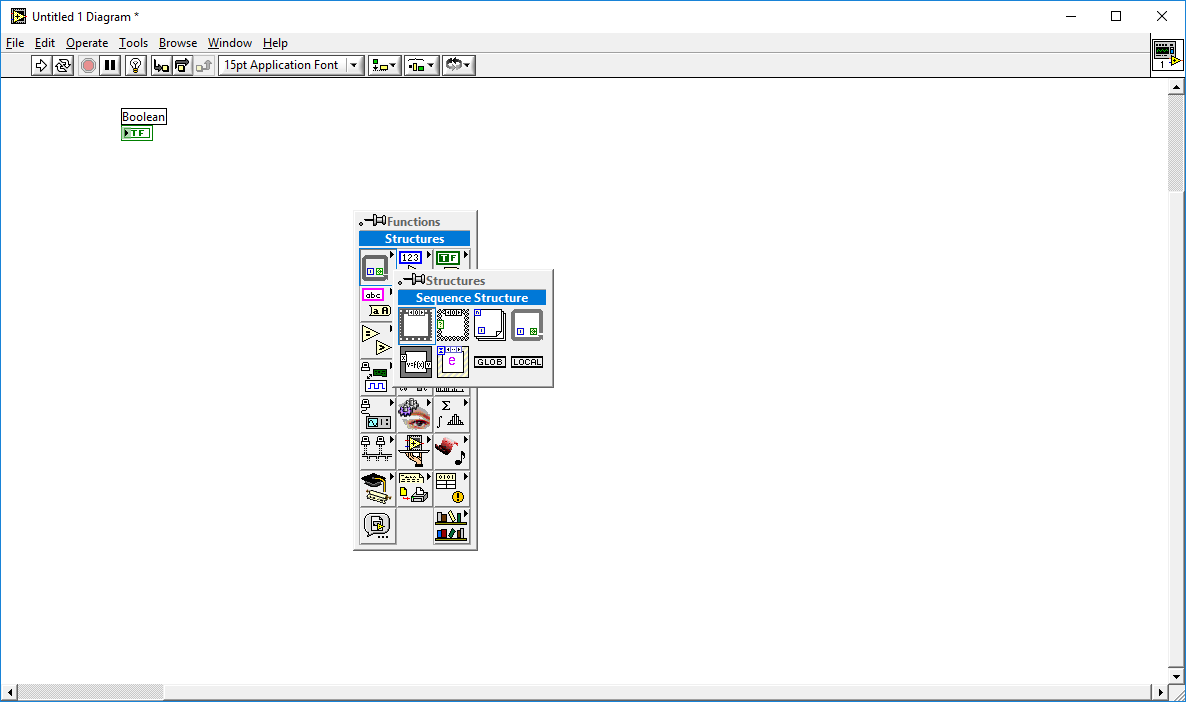


Рисунок 4 – Место расположения структуры «Последовательность» (*Sequence*)

Иллюстрация последовательности, как и прежде в *National Instruments LabView*, весьма показательна. Вычислительный процесс будет протекать подобно кадрам киноплёнки, один за другим. Разве что иллюстрироваться кадры будут один под другим, а не один вслед за другим, но это не имеет для решения инженерных задач какого-то принципиального значения. Через контекстное меню структуры последовательности можно выбрать добавление последующего кадра («*Add Frame After*»), добавление предыдущего кадра («*Add Frame Before*»), дублирование кадра («*Duplicate Frame*»).

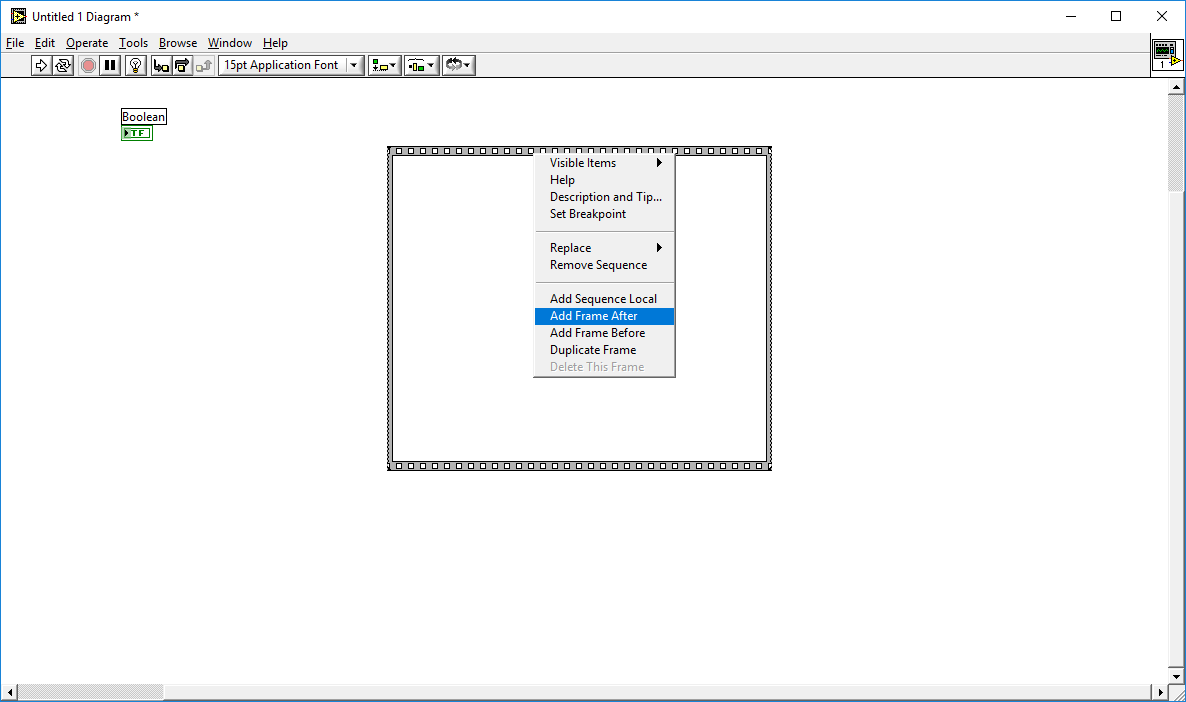


Рисунок 5 – Место расположения в контекстном меню пункта, ответственного для создания ещё одного кадра последовательности, расположенного после рассматриваемого

Далее на Рисунках 6-14 рассмотрено решение задачи «Автомобильный светофор». На Рисунках 6-9 представлены состояния, которые последовательно сменяются в автомобильном светофоре с момента запуска виртуального прибора на исполнение. На Рисунках 10-14 представлены блок-диаграммы с последовательно отображённым кадрами и их содержимым. В начале работы программы все сигналы выключены (Рисунок 6).

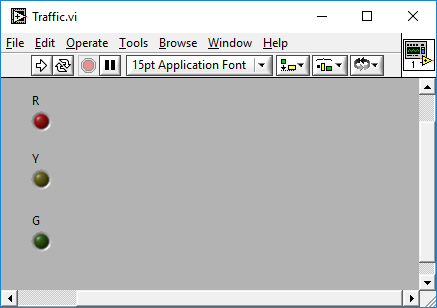


Рисунок 6 – Исходное, нерабочее состояние виртуального прибора «Автомобильный светофор»

Через некоторое время после запуска виртуального прибора на исполнение загорается красный сигнал светофора (Рисунок 7).

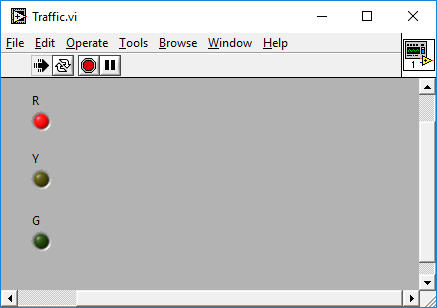


Рисунок 7 – Первое из рабочих состояний виртуального прибора «Автомобильный светофор»

Спустя ещё время загорается жёлтый сигнал светофора (Рисунок 8).

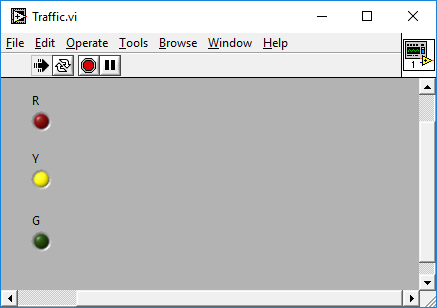


Рисунок 8 – Второе из рабочих состояний виртуального прибора «Автомобильный светофор»

Далее загорается зелёный сигнал светофора (Рисунок 9).

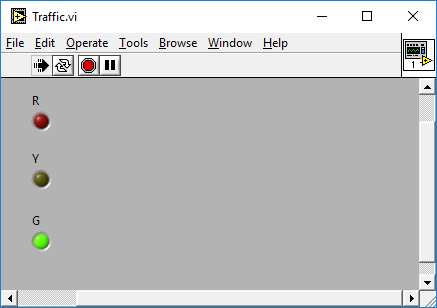


Рисунок 9 – Третье из рабочих состояний виртуального прибора «Автомобильный светофор»

Четвёртое состояние на передней панели не отображено, поскольку оно полностью дублирует самое первое, нерабочее состояние. Тем не менее его нужно программировать отдельным звеном цепочки вычислений в последовательности.

Итого, далее рассмотрим, как именно программируется представленный вычислительный процесс. На Рисунке 10 представлена первая (или нулевая) цепочка последовательности. Она необходима для выдерживания некоторого времени с момента запуска. Начало переходного процесса – это переход от выключенного состояния ко включённому на протяжении некоторого времени (или же сброс в исходное состояние для случаев, когда цепочка выполнения событий была прервана на одной из промежуточных стадий выполнения).

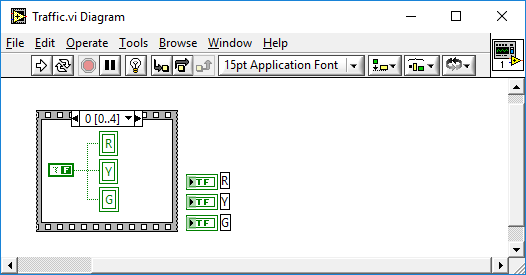


Рисунок 10 – Нулевое состояние виртуального прибора – сброс к исходному. Все сигналы светофора выключены

На Рисунке 11 показано как зажигается красный сигнал светофора, который горит на протяжении 1000 миллисекунд. Реально здесь может выполняться только установка красного сигнала светофора. Локальные переменные для жёлтого и зелёного сигналов оставлены с целью обеспечения единообразия схемы на блок-диаграмме. Это эстетический момент, который с точки зрения программирования избыточен.

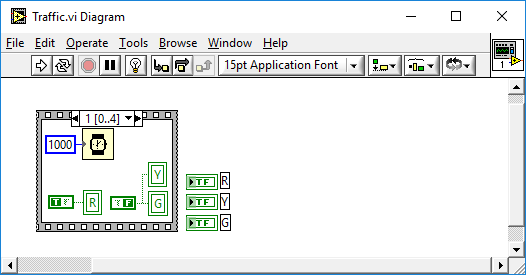


Рисунок 11 – Первое состояние виртуального прибора: горит только красный сигнал светофора

На Рисунке 12 показано как зажигается жёлтый сигнал светофора, который горит на протяжении 1000 миллисекунд. Реально здесь может выполняться только установка жёлтого и сброс красного сигналов светофора. Локальная переменная для зелёного сигнала оставлена с целью обеспечения единообразия схемы. С точки зрения программирования представленное наполнение звена последовательности обладает избыточностью.

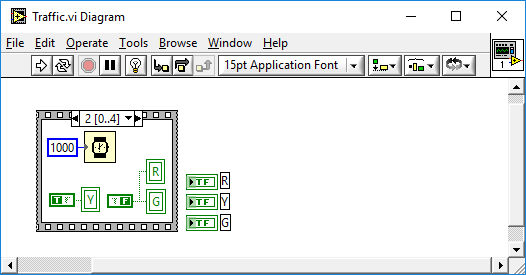


Рисунок 12 – Второе состояние виртуального прибора: горит только жёлтый сигнал светофора

На Рисунке 13 показано как зажигается зелёный сигнал светофора, который горит на протяжении 1000 миллисекунд. Реально здесь может выполняться только установка зелёного сигнала и сброс жёлтого сигналов светофора.

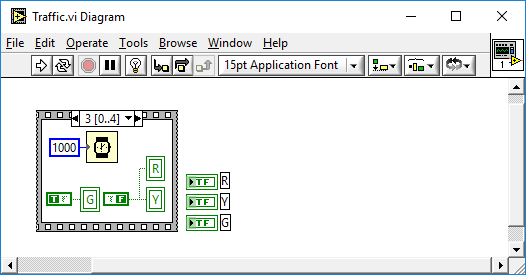


Рисунок 13 – Третье состояние виртуального прибора: горит только зелёный сигнал светофора

На Рисунке 14 показано как гасятся сигналы светофора. Реально здесь может выполняться только сброс зелёного сигнала светофора.

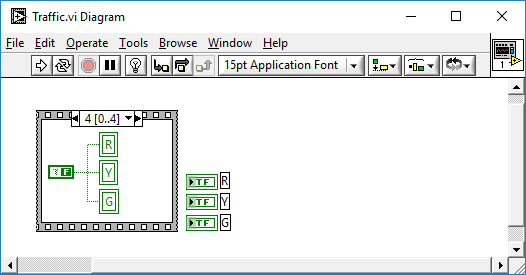


Рисунок 14 – Четвёртое состояние виртуального прибора: все сигналы светофора выключены

Для большего осмысления решённой задачи «Автомобильный светофор» с точки зрения программирования и алгоритмизации рекомендуется зациклить рассмотренную последовательность.

**Варианты сигнализаторов и индикаторов:**

Таблица 1 – Типовые сигнализаторы и индикаторы

|  |
| --- |
| 1. Смоделировать графический индикатор загрузки файла указанного объёма при варьируемой скорости соединения без числовой индикации прогресса. |
| 2. Смоделировать работу термостата на электрическом нагревателе (радиаторе) без числовой индикации. |
| 3. Смоделировать обратный отсчёт на семисегментном индикаторе с девяти до нуля. |
| 4. Смоделировать работу автомобильного светофора с числовой индикацией (обратный отсчёт до смены сигнала). |
| 5. Смоделировать прямой счёт на семисегментном индикаторе от нуля до девяти. |
| 6. На светодиодной матрице размерностью [5x10] выполнить прямой счёт от нуля до девяти. |
| 7. Смоделировать индикатор загрузки файла указанного объёма при варьируемой скорости соединения с числовой индикацией прогресса (показывать, сколько байтов загружено). |
| 8. Смоделировать работу пешеходного светофора без числовой индикации. |
| 9. Смоделировать работу термостата на электрическом нагревателе с числовой индикацией (показывать количество градусов по Цельсию с точностью до двух знаков). |
| 10. Смоделировать работу трамвайного светофора без числовой индикации. |
| 11. Смоделировать обратный отсчёт на двух семисегментных индикаторах от шестнадцати до нуля. |
| 12. Смоделировать работу пешеходного светофора с числовой индикацией (показывать количество секунд, оставшихся до переключения). |
| 13. Смоделировать работу сливного бачка унитаза (нажатие на кнопку инициирует сброс воды, по окончании сброса начинается наполнение до указанного уровня). |
| 14. Смоделировать прямой счёт на двух семисегментных индикаторах от нуля до шестнадцати. |
| 15. Смоделировать работу пешеходного светофора с индикацией обратного отсчёта на семисегментных индикаторах. |
| 16. На светодиодной матрице размерностью [5x10] выполнить обратный отсчёт от девяти до нуля. |
| 17. Смоделировать индикатор загрузки приложения известного объёма при варьируемой скорости соединения. |
| 18. Смоделировать работу трамвайного светофора с числовой индикацией (обратный отсчёт до смены состояний). |
| 19. Выполнить горизонтальное послойное заполнение светодиодной матрицы [5x10] с сокращающийся длительностью задержки на каждом слое. |
| 20. Выполнить вертикальное послойное заполнение светодиодной матрицы [5x10] с увеличивающейся длительностью задержки на каждом слое. |