Министерство транспорта Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение высшего образования

«Российский университет транспорта (МИИТ)» (РУТ МИИТ)

Институт транспортной техники и систем управления

Кафедра «Управление и защиты информации»

Отчёт

по практике

«Учебная практика»

(вид практики)

Ознакомительная практика

(наименование практики)

«Проектирование и создание виртуальных приборов в *National Instruments LabView*»

Выполнил: ст. гр. ТКИ-241

Пылаева С.А.

Вариант №12

Проверил: доц. Сафронов А.И.

Москва – 2025 г.

Оглавление

[1. Задание 1 2](#_Toc203923892)

[2. Задание 2 6](#_Toc203923893)

[3. Задание 3 17](#_Toc203923894)

[4. Задание 4 27](#_Toc203923895)

# Задание 1

**Цель работы**

Освоение навыков разработки пользовательских процедур. Закрепление раздела «Ряды» из курса высшей математики. Закрепление раздела «Функции и процедуры» из курса алгоритмизации и технологий программирования.

**Формулировка задачи**

Разработать пользовательскую подпрограмму (виртуальный прибор), реализующую вычисление с заданной точностью указанных по варианту функций. В случае образования функции факториала при разложении указанной функции в ряд использовать подпрограмму, составленную в общей части данного задания. По итогам создания и отладки пользовательской подпрограммы создать виртуальный прибор, блок-диаграмма которого содержала бы не более чем:

- элементы ввода данных,

- элементы вывода данных,

- пользовательскую подпрограмму.

Натуральный логарифм числа (*x* + 2), вычисленный с учётом *N* членов ряда (циклическая структура *For*).

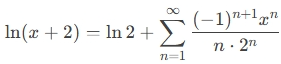


Рисунок 1 – Разложение в ряд Маклорена,

**Блок-схемы алгоритма программы**

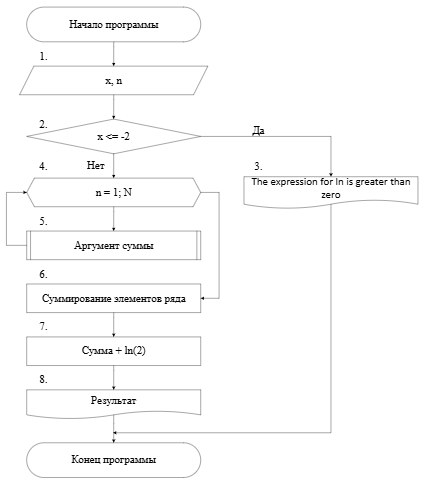


Рисунок 2 – Блок-схема алгоритма разложения в ряд

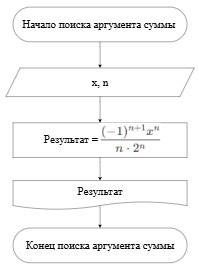


Рисунок 3 – Блок-схема подпрограммы, рассчитывающей аргумент под знаком суммы

**Блок-диаграммы *LabView***

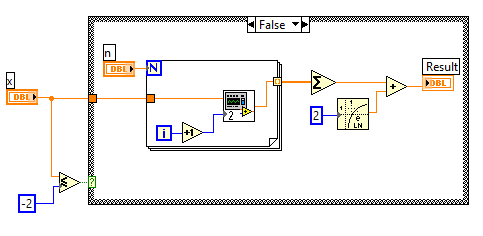


Рисунок 4 – Блок-диаграмма программы, при условии, что

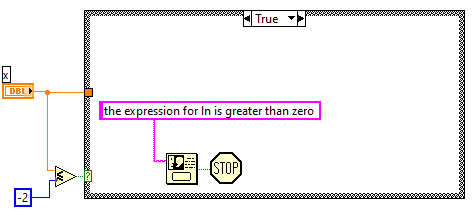


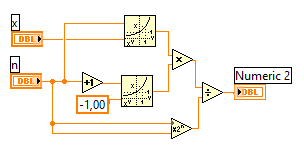
Рисунок 5 – Блок-диаграмма программы, при условии, что 

Рисунок 6 – Блок-диаграмма подпрограммы, рассчитывающей аргумент под знаком суммы

**Пользовательские интерфейсы *LabView***

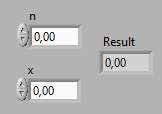


Рисунок 7 – Пользовательский интерфейс программы разложения в ряд

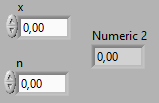


Рисунок 8 – Пользовательский интерфейс подпрограммы, рассчитывающей аргумент под знаком суммы

**Расчёт тестовых примеров**

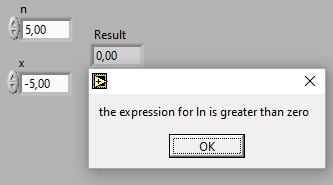


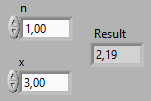
Рисунок 9 – Результат, при недопустимых введенных данных

Рисунок 10 – Результат расчётов при 1 члене ряда

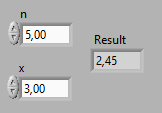


Рисунок 11 – Результат расчётов при 5 членах ряда

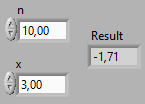


Рисунок 12 – Результат расчётов при 10 членах ряда

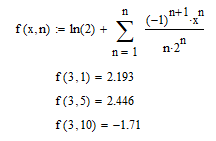


Рисунок 13 – Проверка расчётов в *Mathcad*

Результаты расчётов из *LabView* и *Mathcad* совпадают.

**Вывод**

В ходе проделанной работы были рассчитаны, с учётом *N* членов ряда (циклическая структура *For*), значения разложения в ряд Маклорена с помощью виртуального прибора в *LabView*. Также была использована подпрограмма для расчёта значения аргумента под знаком суммы.

# Задание 2

**Цель работы**

Закрепление навыков использования графической структуры цикла *For* (или структуры цикла *While* в зависимости от реализации), приобретение навыков конвертации и проецирования значений периодических и не периодических функций на окружность (преобразования в полярную систему координат).

**Формулировка задачи**

В пакете прикладных программ *National Instruments LabView* создать виртуальный прибор «Полярные координаты», в рамках которого выполнить построение графика заданной по варианту функции в декартовой системе координат.

Виртуальный прибор должен предусматривать возможность ввода пользователем диапазона построения графика функциональной зависимости в формате:

* начало диапазона,
* конец диапазона,
* шаг дискретизации.

Исключить ситуацию ошибочного запуска виртуального прибора с нулевым значением шага построения. Любым известным способом.

Учащимся с чётными номерами вариантов следует предусмотреть на графическом пользовательском интерфейсе только одну область для построения графика «*XY Graph*», куда в зависимости от указанного диапазона построения графика функции в декартовой системе координат центрировано выводить изображение графика функции в полярной системе координат, а также изображение графика функции в Декартовой системе координат.

**Блок-схемы алгоритма программы**

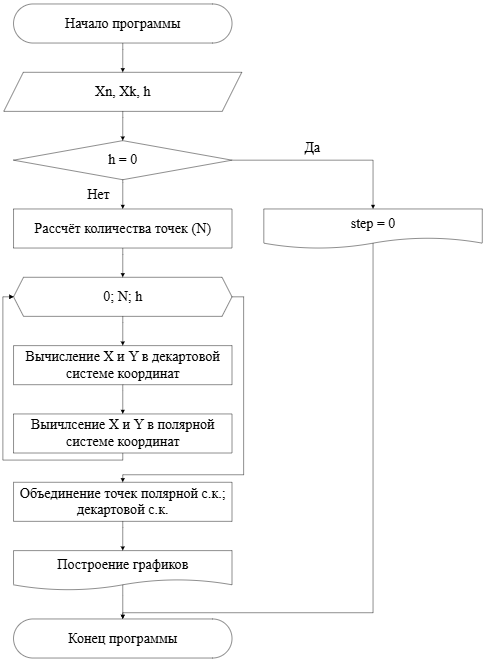
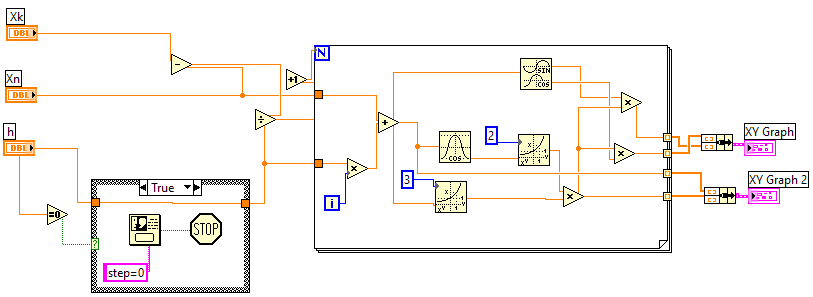
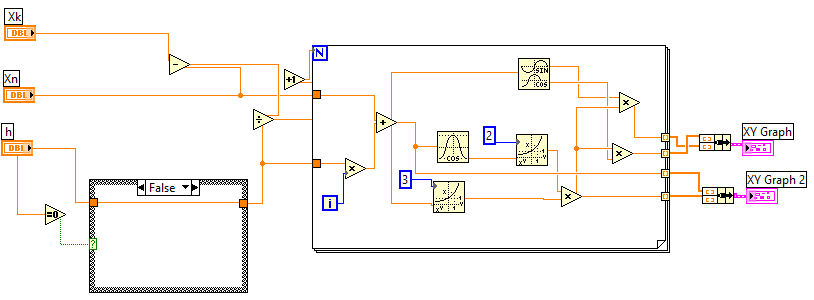
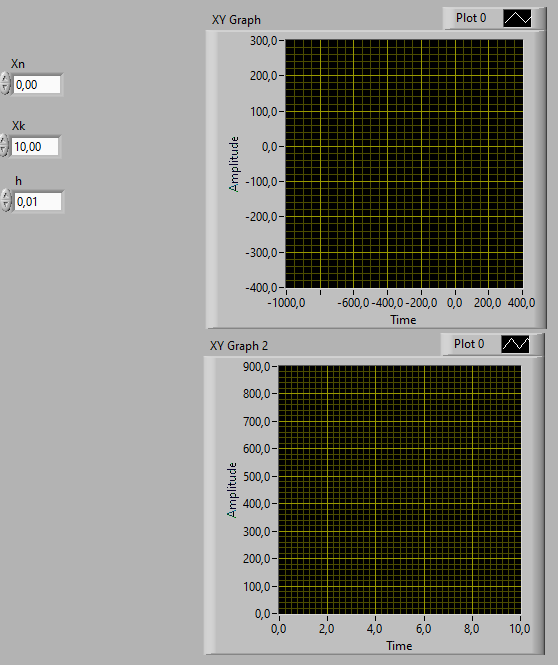


Рисунок 14 – Блок-схема алгоритма программы

**Блок-диаграммы *LabView***

Рисунок 15 – Блок-диаграмма программы при нулевом шагеРисунок 16 – Блок-диаграмма программы при ненулевом шаге

**Пользовательские интерфейсы *LabView***

Рисунок 17 – Пользовательский интерфейс программы

**Расчёт тестовых примеров**

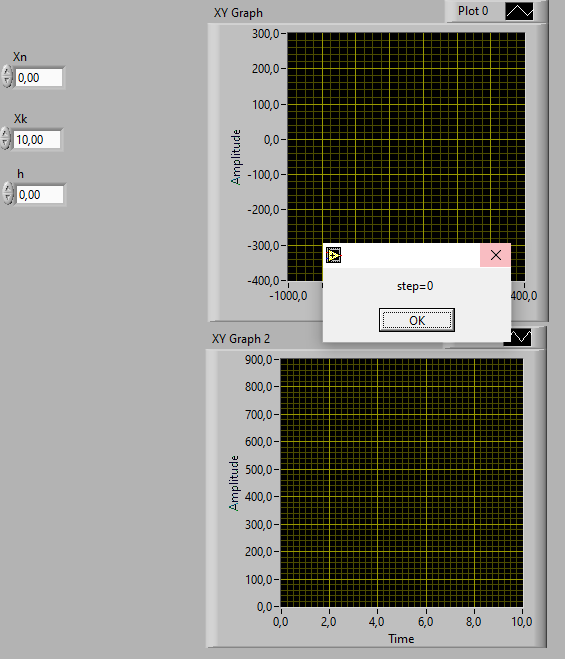
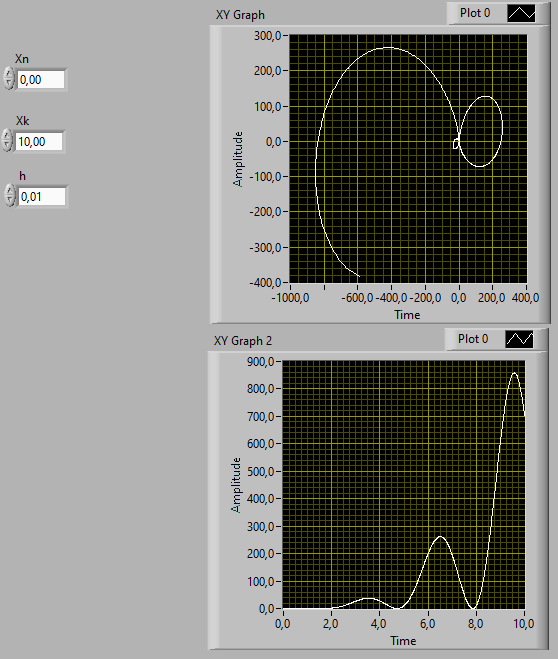


Рисунок 18 – Вывод программы при нулевом шаге

Рисунок 19 – Вывод программы с шагом 0,01, интервал [0; 10]

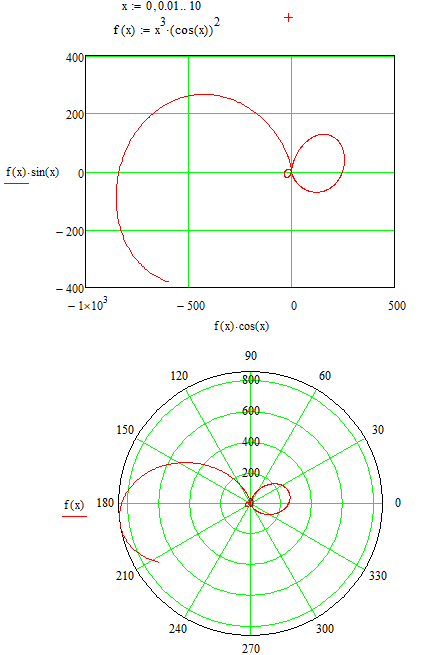


Рисунок 20 – График в полярной системе координат с шагом 0,01,

интервал [0; 10] *Mathcad*

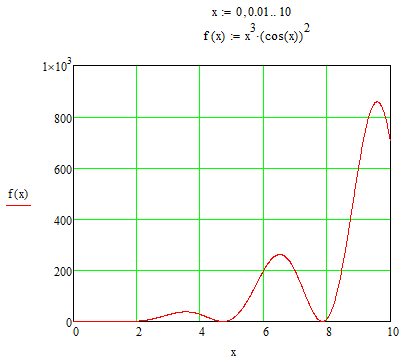


Рисунок 21 – График в декартовой системе координат с шагом 0,01,

интервал [0; 10] *Mathcad*

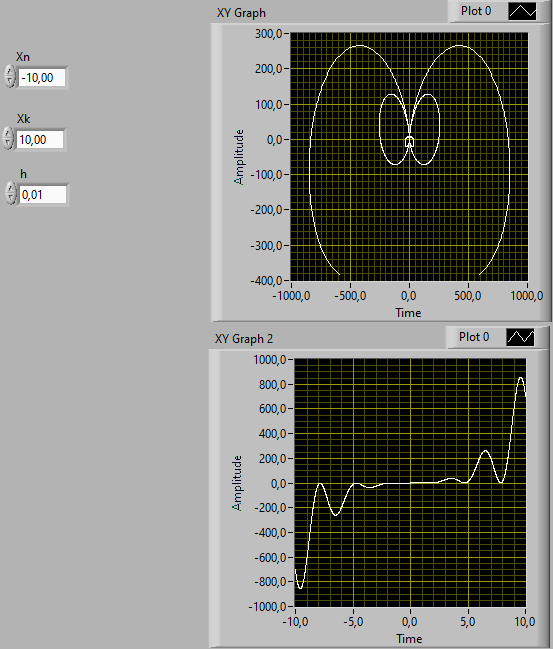


Рисунок 22 – Вывод программы с шагом 0,01, интервал [-10; 10]

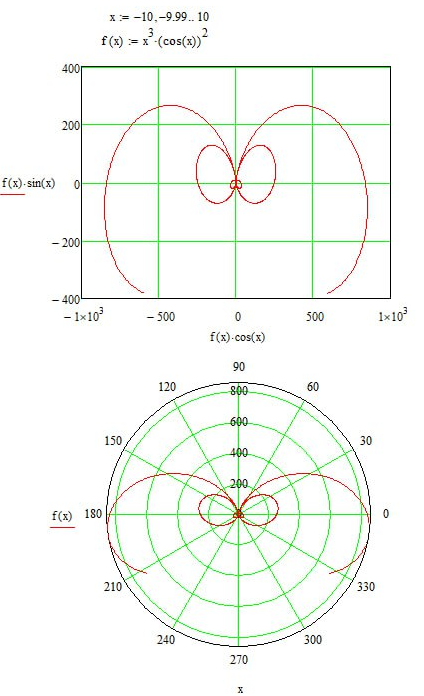


Рисунок 23 – График в полярной системе координат с шагом 0,01,

интервал [-10; 10] *Mathcad*

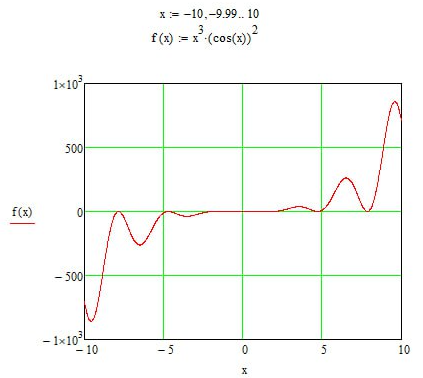


Рисунок 24 – График в декартовой системе координат с шагом 0,01,

интервал [-10; 10] *Mathcad*

**Вывод**

В ходе проделанной работы был создан виртуальный прибор в *LabView*, рассчитывающий значения *X* и *Y* при заданном диапазоне и шаге построения, для создания графиков в декартовой и полярной системах координат.

# Задание 3

**Цель работы**

Освоение навыков работы с текстовыми файлами как протоколами передачи информации между средами программирования и пакетами прикладных программ; приобретение знаний, касающихся настройки диалогов и событий; знакомство с механизмом обработки событий в среде с графическим языком программирования.

**Формулировка задачи**

В пакете прикладных программ *National Instruments LabView* создать виртуальные приборы для загрузки данных из файла и сохранения данных в файл.

Для чётных вариантов необходимо реализовать два виртуальных прибора: один для загрузки, другой – для сохранения.

Использовать следующие схемы загрузки/сохранения:

а) Расчёт значений функции выполняется в *Microsoft Office Excel* и записывается в файл *fromExcel.txt* (в текстовый файл копируются два столбца: в одном содержатся значения абсцисс, в другом – значения ординат). Между столбцами вручную должны быть удалены элементы табуляции, разделяющие значения, и на их место должны быть вставлены разделители в соответствии с таблицей вариантов. В пакете *National Instruments LabView* выполняется построчное считывание данных из файла fromExcel.txt и конвертация их для построения графика функциональной зависимости.

б) Расчёт значений функции выполняется в пакете *National Instruments* *LabView* и записывается в файл *fromLabView.txt.* В *Microsoft Office Excel* из файла *fromLabView.txt* копируются рассчитанные значения, и далее по ним проводится построение графика функциональной зависимости.

Для вариантов, дающих в остатке двойку при делении на 5, – точность до трёх знаков после запятой.

Разделитель: **~**

**Блок-схемы алгоритма программы**



Рисунок 25 – Блок-схема алгоритма программы, считывающей данные из файла

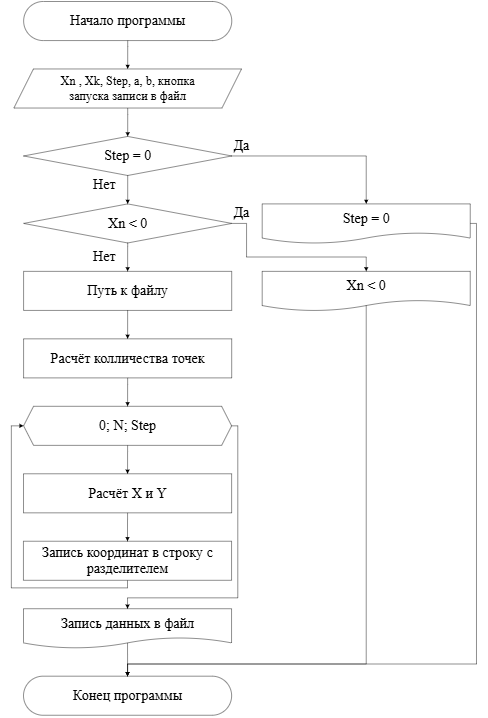


Рисунок 26 – Блок-схема алгоритма программы, записывающей данные в файл

**Блок-диаграммы *LabView***

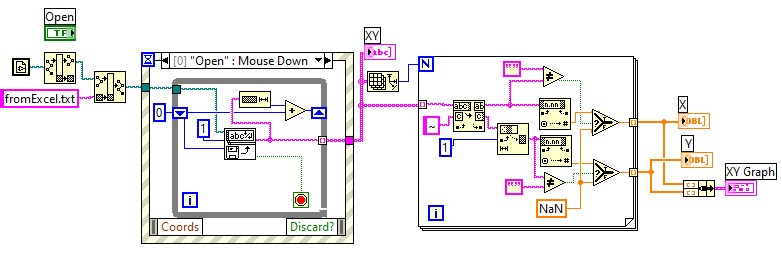


Рисунок 27 – Блок-диаграмма программы, считывающей данные из файла

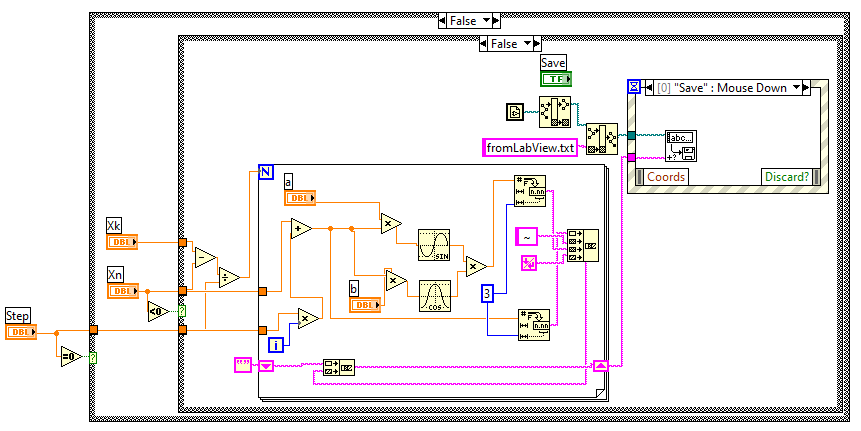
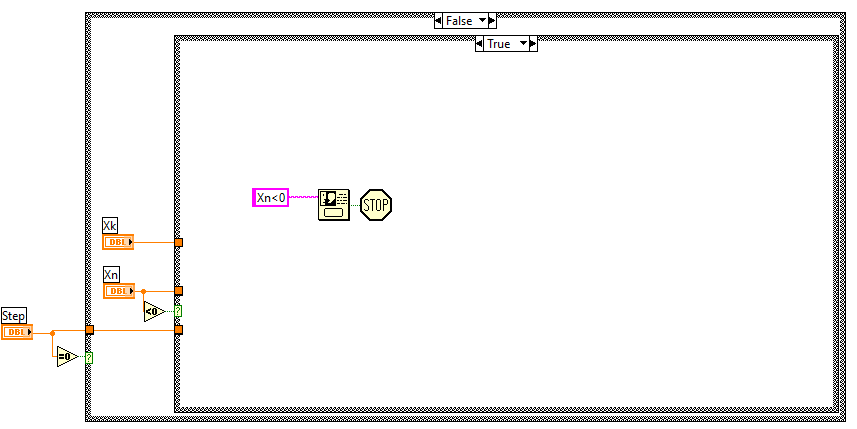
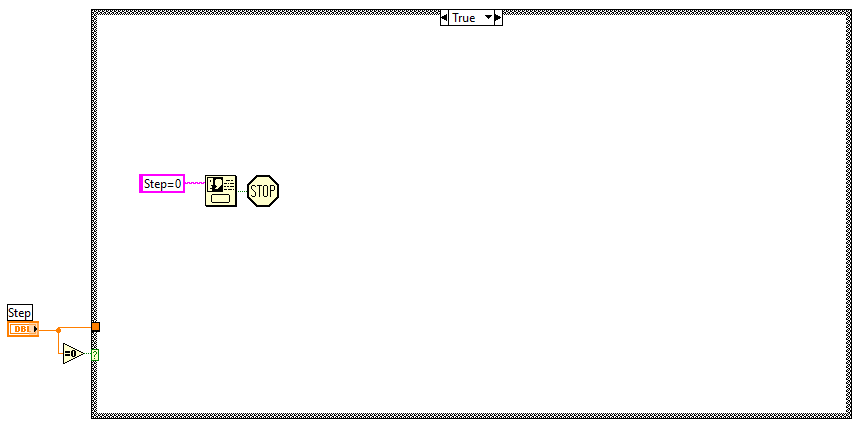


Рисунок 28 – Блок-диаграмма программы, записывающей данные в файл

Рисунок 29 – Блок-диаграмма программы, записывающей данные в файл, если Xn < 0

Рисунок 30 – Блок-диаграмма программы, записывающей данные в файл, если Step = 0

**Пользовательские интерфейсы *LabView***

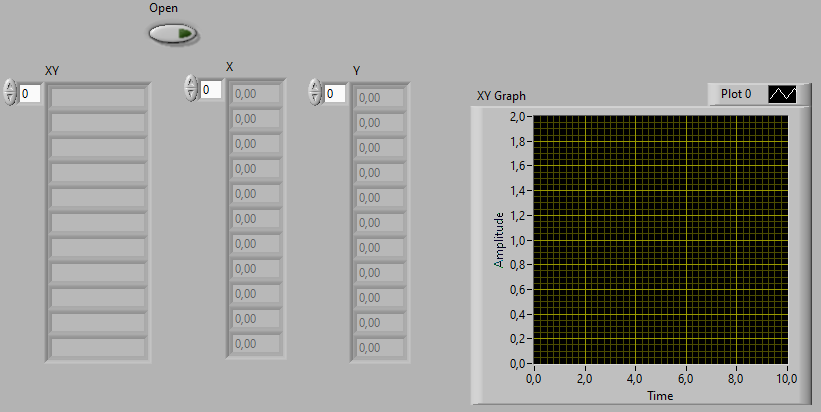


Рисунок 31 – Пользовательский интерфейс программы, считывающей данные из файла

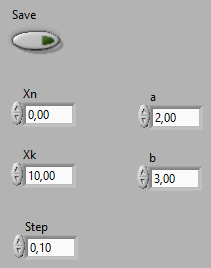


Рисунок 32 – Пользовательский интерфейс программы, записывающей данные в файл

**Расчёт тестовых примеров**

Для пункта а) создаётся файл *Excel*, в который записываются координаты 101 точки: в столбце B значения *X*, в столбце C рассчитанные по формуле значения *Y*.

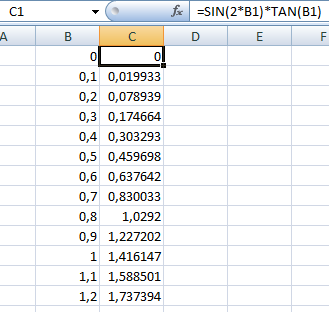


Рисунок 33 – Фрагмент записанных в *Excel* данных для пункта а)

Затем, эти данные записываются в текстовый файл. Разделитель столбцов заменяется вручную на тот, который указан в варианте.

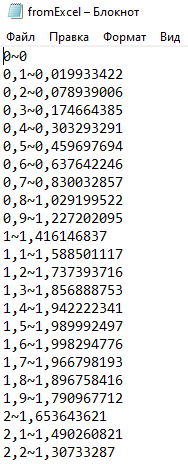


Рисунок 34 – Фрагмент текстового файла для пункта а)

В *LabView* считываются данные из текстового файла, и по ним строится график.

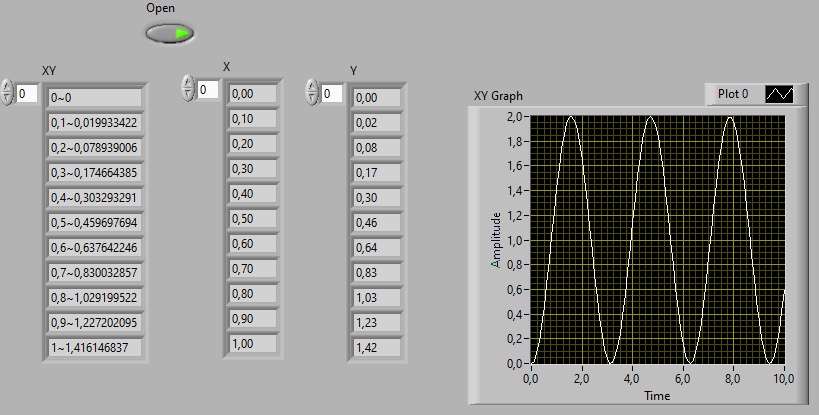


Рисунок 35 – Результат выполнения программы для пункта а)

Для пункта б) сначала создаётся файл *LabView*, в него вводятся диапазон и шаг построения графика, а также остальные параметры, необходимые для вычислений.

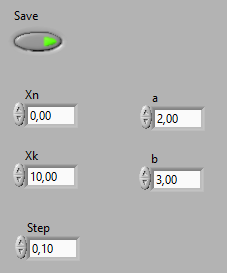


Рисунок 36 – Пользовательский интерфейс программы пункта б) с введёнными параметрами

Программа производит вычисления значений *X* и *Y*, записывая результат в текстовый файл.

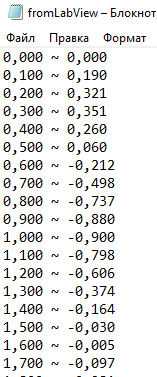


Рисунок 37 – Фрагмент текстового файла для пункта б)

Затем, значения из текстового файла копируются в *Excel*, и по ним строится график.

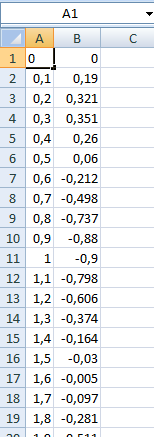


Рисунок 38 – Фрагмент записанных в *Excel* данных для пункта б)

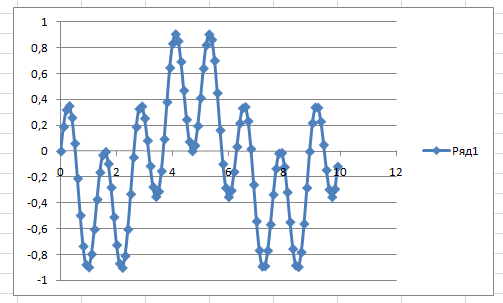


Рисунок 39 – График, построенный в *Excel* для пункта б)

**Вывод**

В ходе проделанной работы были созданы две программы: одна для загрузки данных из файла, другая для сохранения данных в файл. Как результат, в *LabView* и *Excel* построены графики функциональной зависимости.

# Задание 4

**Цель работы**

Освоение навыков работы с локальными и глобальными переменными, а также структурой последовательного вычислительного процесса внутри системы, адаптированной под потоковые вычисления.

**Формулировка задачи**

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.

Смоделировать работу пешеходного светофора с числовой индикацией (показывать количество секунд, оставшихся до переключения).

**Блок-схемы алгоритма программы**

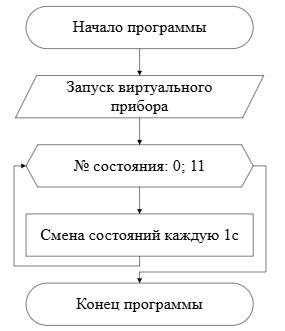


Рисунок 40 – Блок-схема алгоритма программы

**Блок-диаграммы LabView**

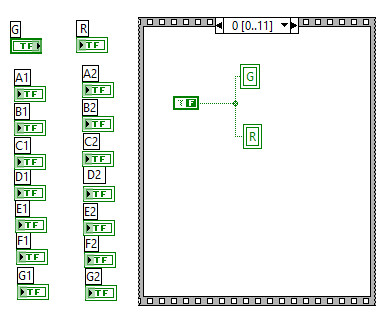


Рисунок 41 – Блок-диаграмма состояния 0

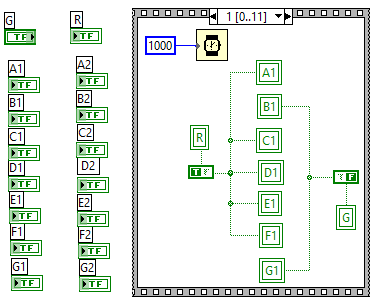


Рисунок 42 – Блок-диаграмма состояния 1

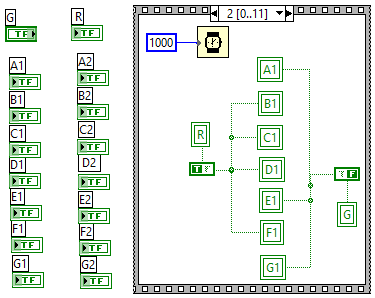


Рисунок 43 – Блок-диаграмма состояния 2

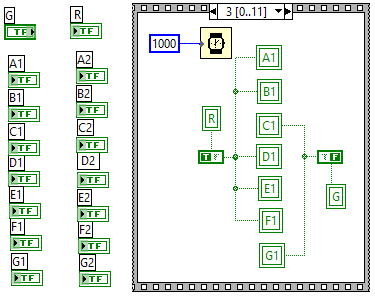


Рисунок 44 – Блок-диаграмма состояния 3

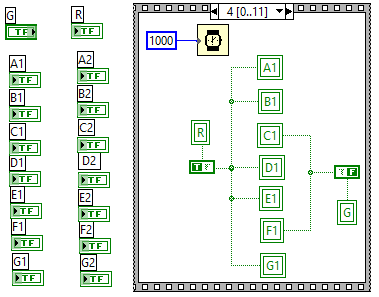


Рисунок 45 – Блок-диаграмма состояния 4

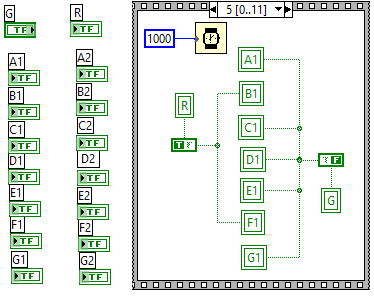


Рисунок 46 – Блок-диаграмма состояния 5

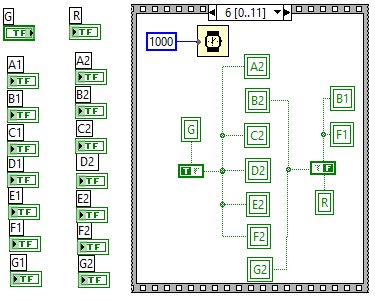


Рисунок 47 – Блок-диаграмма состояния 6

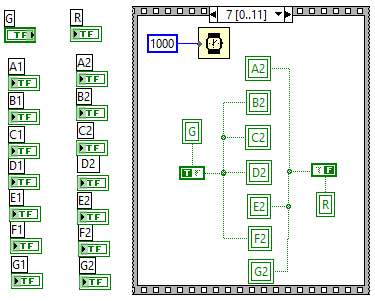


Рисунок 48 – Блок-диаграмма состояния 7

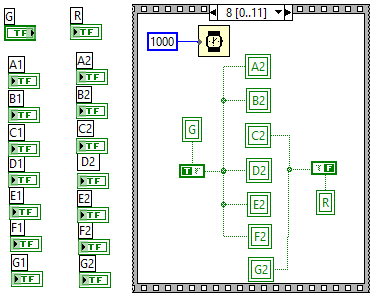


Рисунок 49 – Блок-диаграмма состояния 8

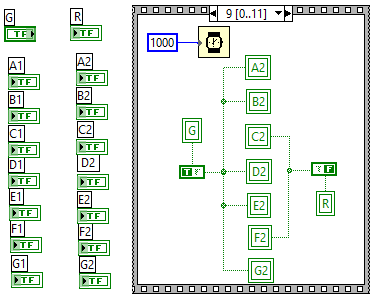


Рисунок 50 – Блок-диаграмма состояния 9

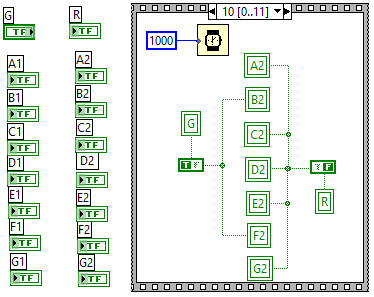


Рисунок 51 – Блок-диаграмма состояния 10

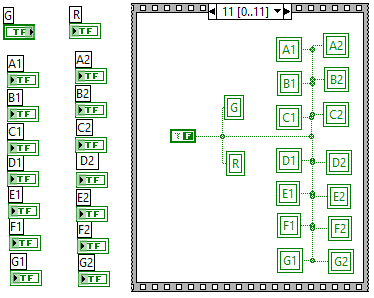


Рисунок 52 – Блок-диаграмма состояния 11

**Пользовательские интерфейсы LabView**

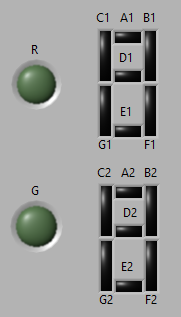


Рисунок 53 – Пользовательский интерфейс

**Расчёт тестовых примеров**

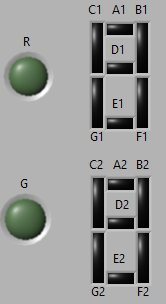


Рисунок 54 – Светофор до включения (состояние 0)

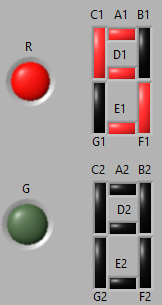


Рисунок 55 – Красный сигнал светофора на 5 секунде (состояние 1)

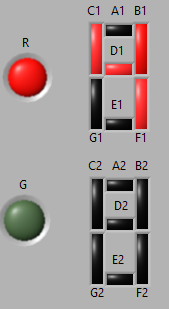


Рисунок 56 – Красный сигнал светофора на 4 секунде (состояние 2)

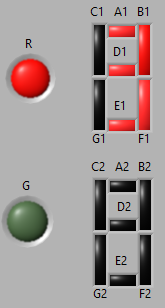


Рисунок 57 – Красный сигнал светофора на 3 секунде (состояние 3)

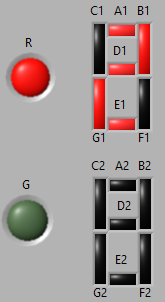


Рисунок 58 – Красный сигнал светофора на 2 секунде (состояние 4)

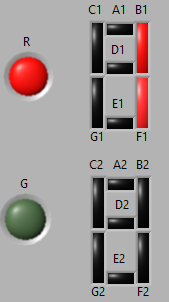


Рисунок 59 – Красный сигнал светофора на 1 секунде (состояние 5)

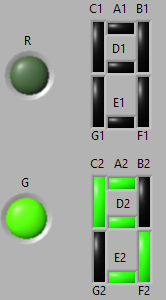


Рисунок 60– Зелёный сигнал светофора на 5 секунде (состояние 6)

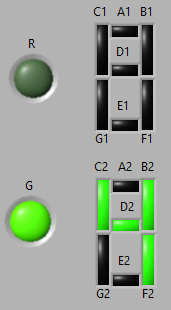


Рисунок 61 – Зелёный сигнал светофора на 4 секунде (состояние 7)

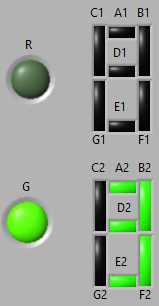


Рисунок 62 – Зелёный сигнал светофора на 3 секунде (состояние 8)

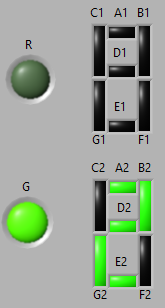


Рисунок 63 – Зелёный сигнал светофора на 2 секунде (состояние 9)

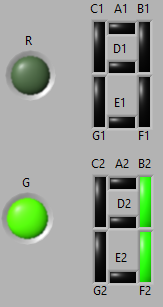


Рисунок 64 – Зелёный сигнал светофора на 1 секунде (состояние 10)

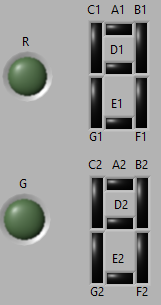


Рисунок 65 – Светофор после выключения (состояние 11)

**Вывод**

В ходе проделанной работы был создан виртуальный прибор в LabView, модулирующий работу пешеходного светофора с числовой индикацией. Освоены навыки работы с локальными и глобальными переменными, а также структурой последовательного вычислительного процесса внутри системы, адаптированной под потоковые вычисления.