**Задание 03.** 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

Для нечётных вариантов необходимо реализовать один виртуальный прибор, сочетающий в себе как загрузку из файла, так и сохранение в файл.

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, делящихся на 5 без остатка, реализовать точность вычислений до одного знака после запятой.

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

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

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

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

**Цель работы (одна из возможных формулировок)**: 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.

**Настройка диалога с проверкой на нулевой шаг при построении графика функциональной зависимости:**

Пример графического пользовательского интерфейса виртуального прибора с ожиданием возникновения ошибки при нулевом введённом шаге:

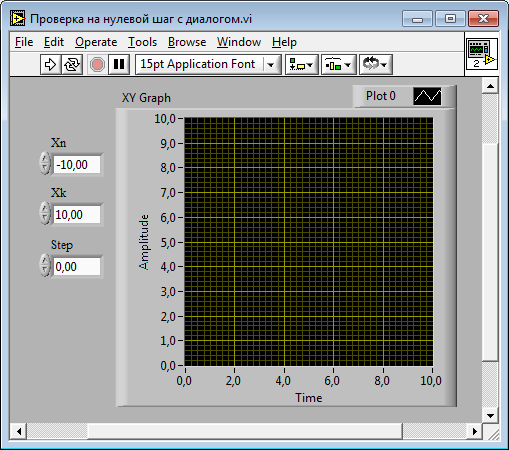


Рисунок 1 – Стандартный графический пользовательский интерфейс для построения графиков функциональных зависимостей

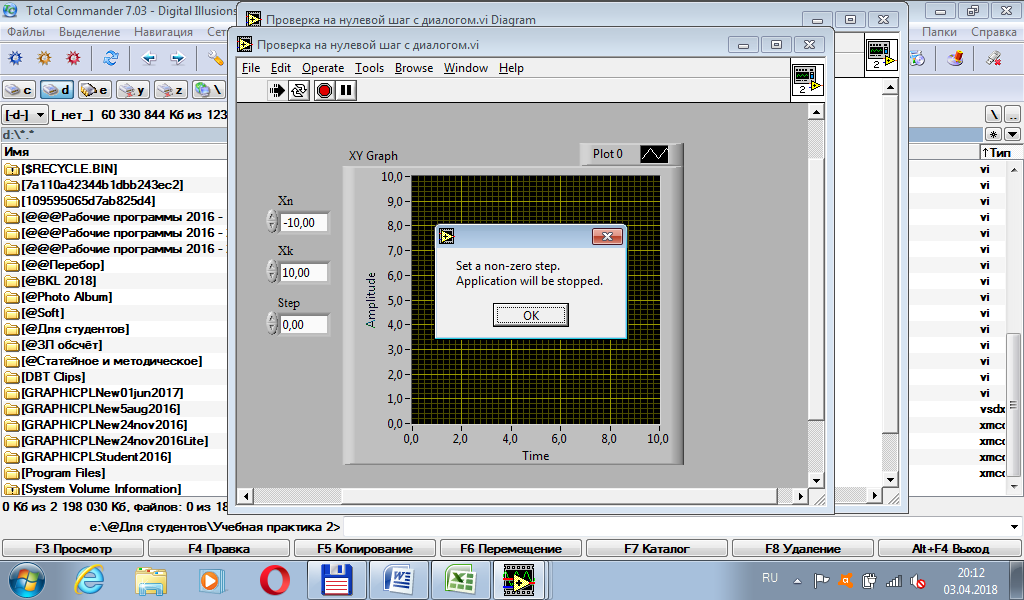


Рисунок 2 – Побуждающее сообщение об ошибке при нулевом шаге

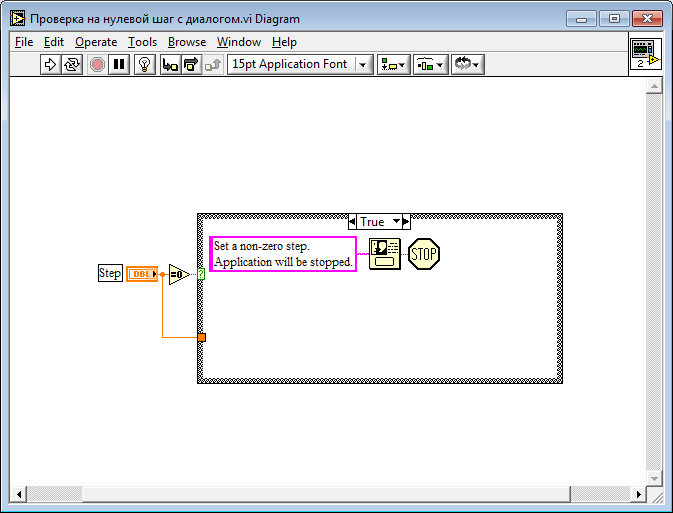


Рисунок 3 – Фрагмент блок-диаграммы с проверкой шага на нулевое значение. В случае нулевого шага – выдача побуждающего сообщения и прерывание работы программы

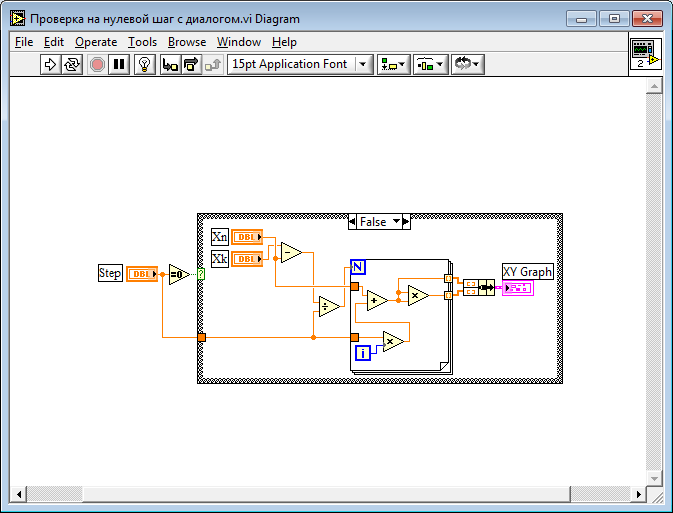


Рисунок 4 – Фрагмент блок-диаграммы для случая, когда задан не нулевой шаг

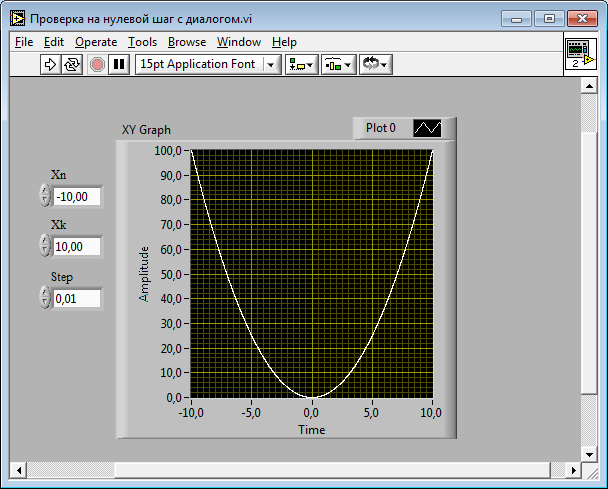


Рисунок 5 – При не нулевом шаге выполняется построение графика функциональной зависимости

**Памятка программисту:**

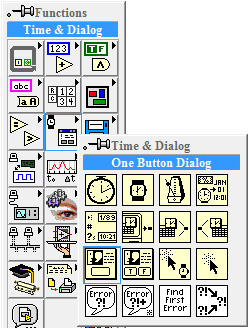


Рисунок 6 – Расположение элемента «Диалог с одной единственной кнопкой» (*ОК*)

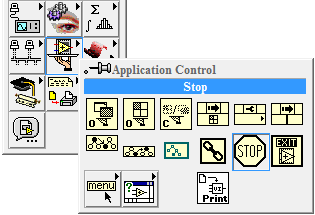


Рисунок 7 – Расположение элемента «Прерывание работы программы» (*STOP*)

**Формирование пути к файлу:**

Файлы, относящиеся к проекту, рекомендуется хранить в непосредственной близости с этим проектом. Таким образом, отталкиваться желательно от пути запуска на исполнение приложения – этот путь гарантированно существует. Для понимания обучающимися идеи формирования путей к файлам в среде *LabView* разработан учебный пример.

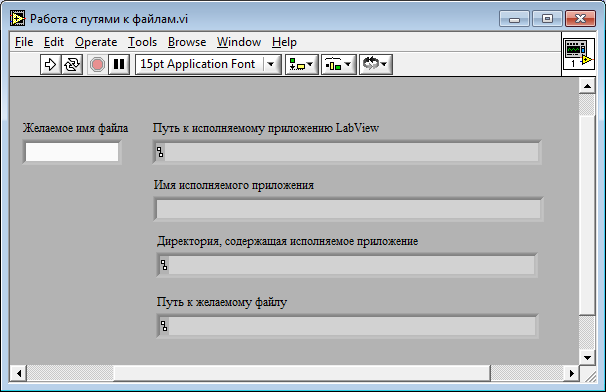
****

Рисунок 8 – Графический пользовательский интерфейс виртуального прибора, обучающего формированию пути к файлу

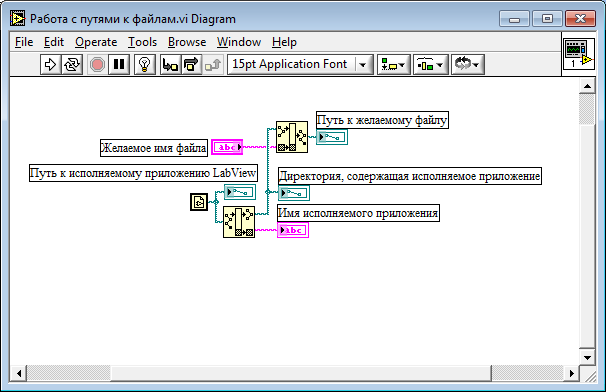
****

Рисунок 9 – Блок-диаграмма виртуального прибора, обучающего формированию пути к файлу

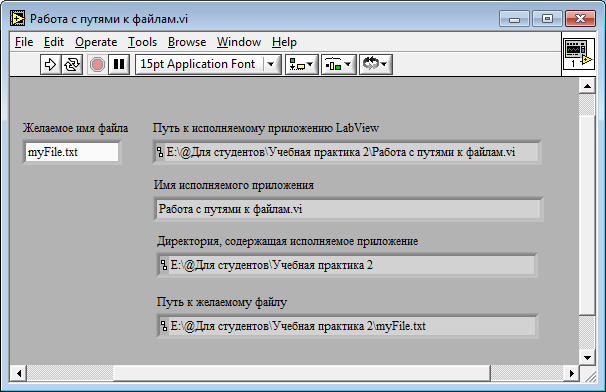
****

Рисунок 10 – Результат запуска виртуального прибора, обучающего формированию пути к файлу «*myFile.txt*»

**Памятка программисту:**

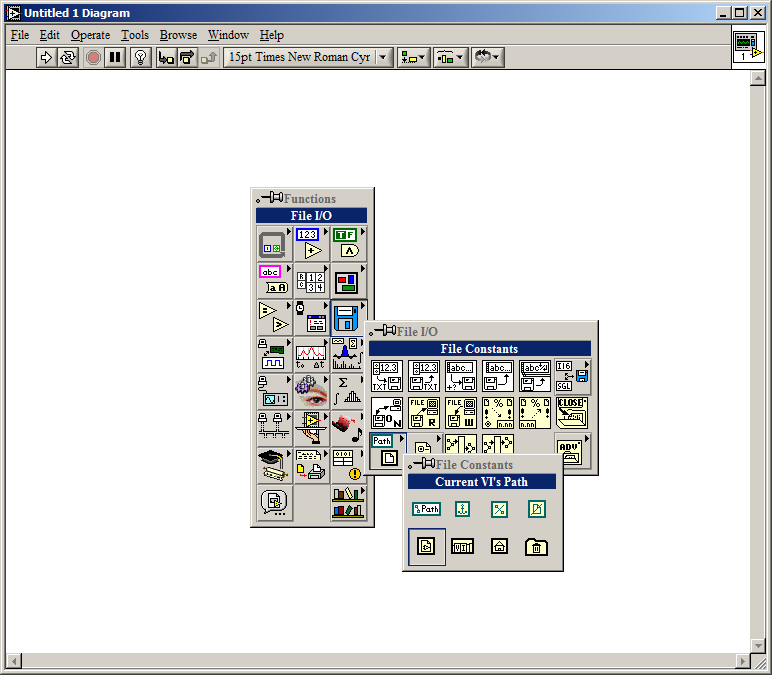


Рисунок 11 – Размещение элемента «Текущий путь к виртуальному прибору»

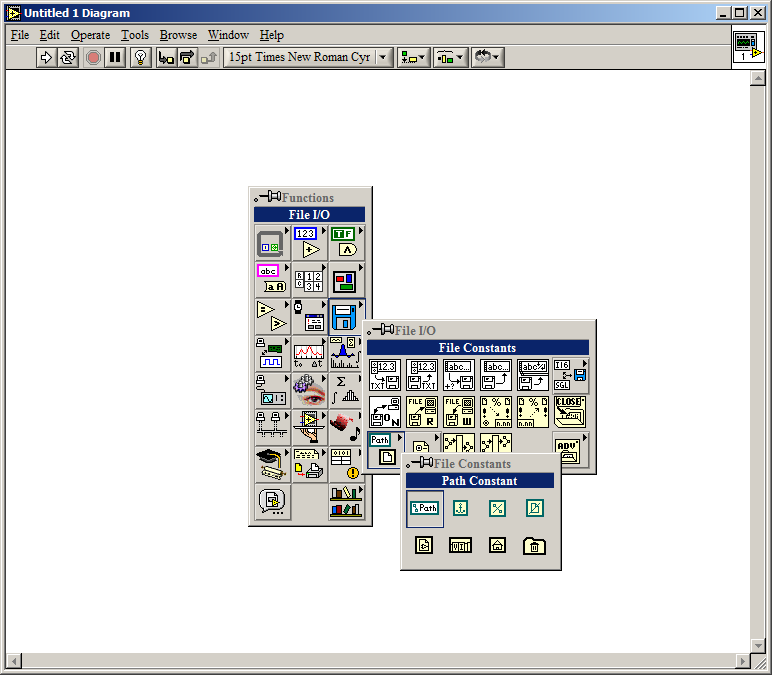


Рисунок 12 – Размещение элемента «Путевая константа»

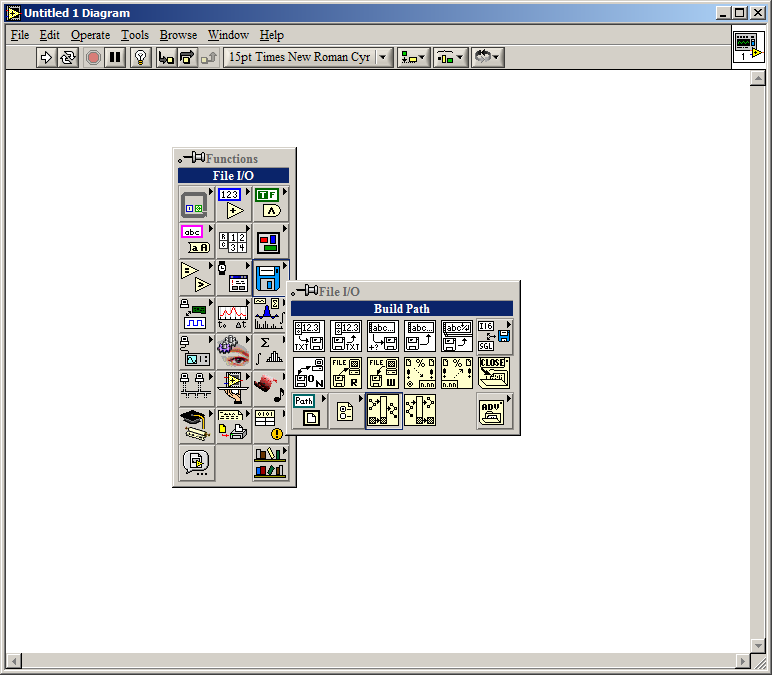


Рисунок 13 – Размещение элемента «Построение пути»

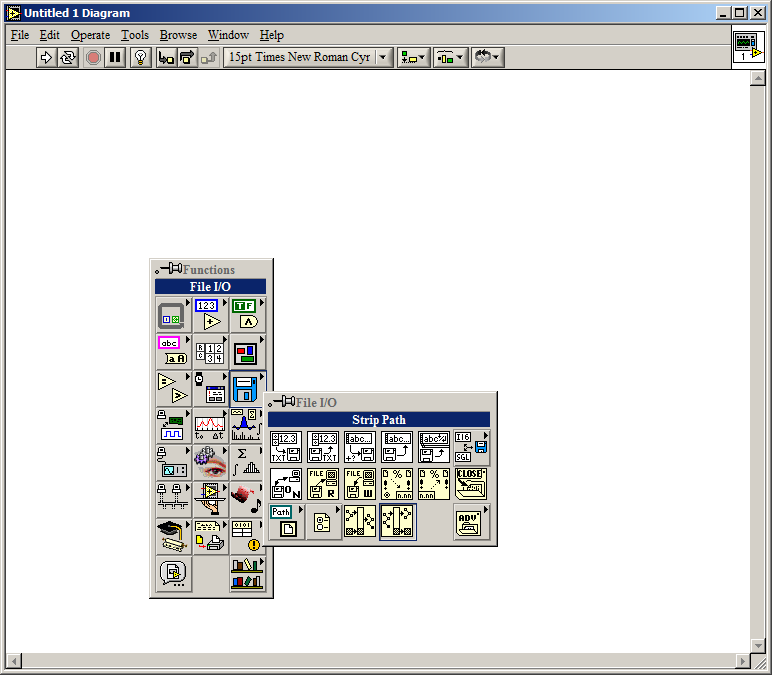


Рисунок 14 – Размещение элемента «Разбор пути»

**Реализация виртуального прибора, записывающего данные в файл:**

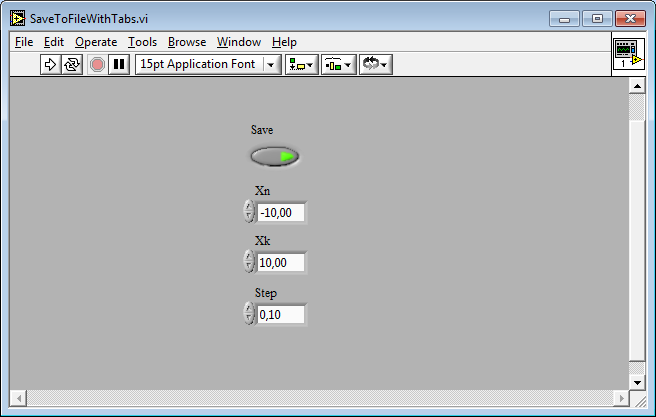
****

Рисунок 15 – Графический пользовательский интерфейс виртуального прибора, реализующего запись в файл

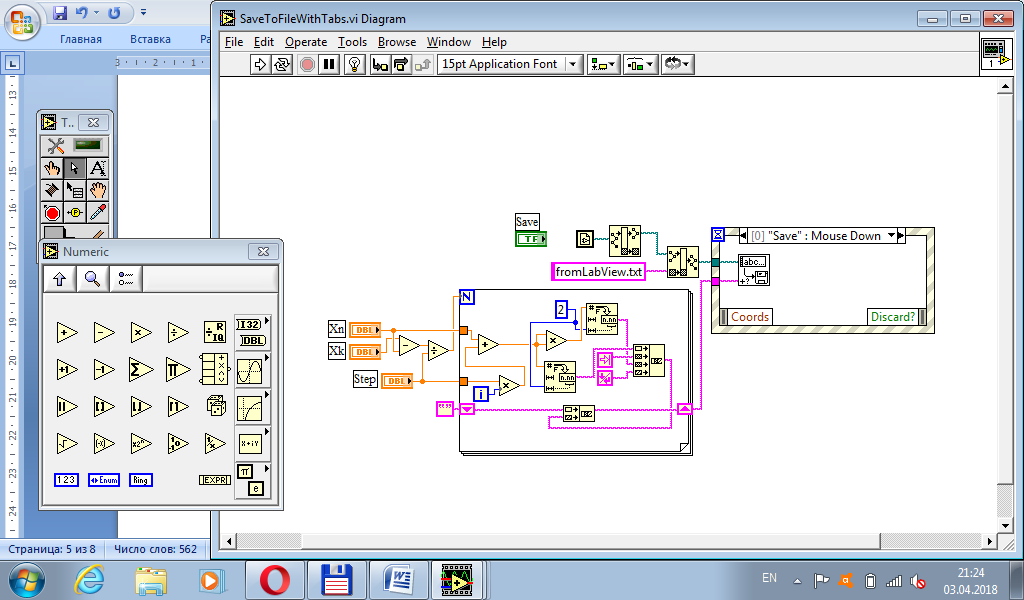
****

Рисунок 16 – Блок-диаграмма виртуального прибора, реализующего запись в текстовый файл. Кнопка записи не связана ни с чем и принципиально не должна быть связанной

**Памятка программисту:**

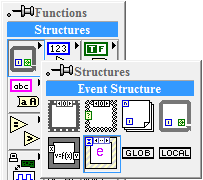


Рисунок 17 – Размещение структуры «Событие» в меню

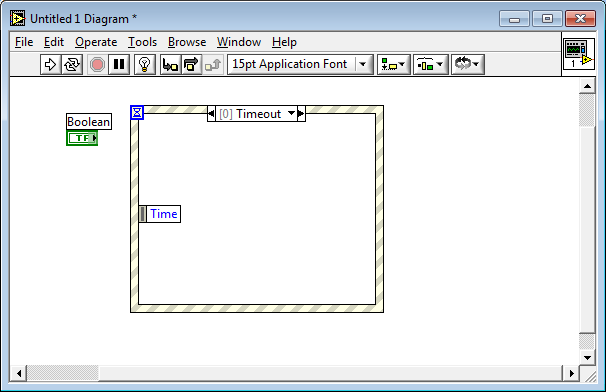
****

Рисунок 18 – Позиционирование структуры «Событие» на блок-диаграмме виртуального прибора

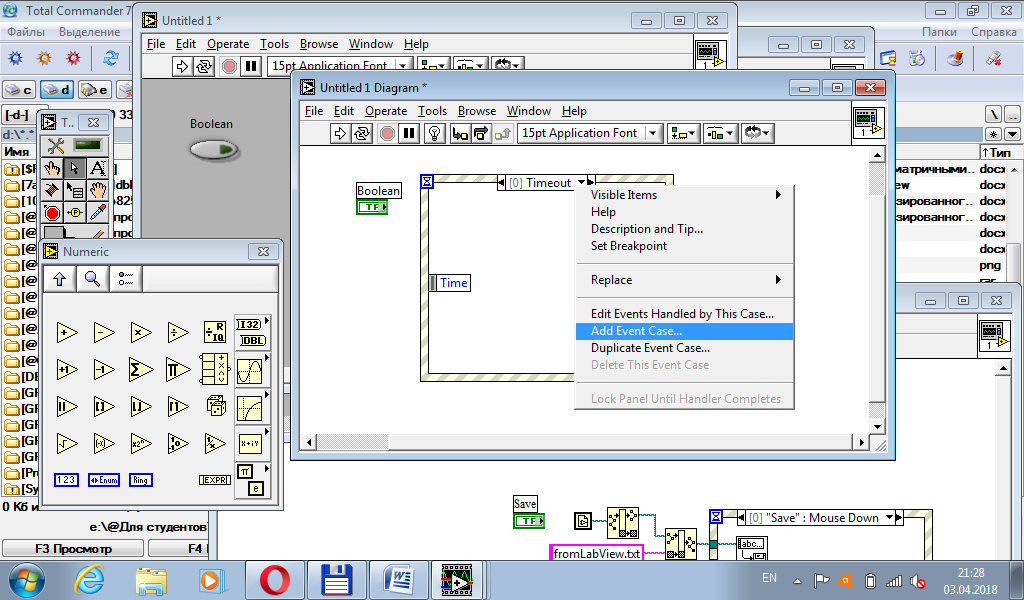
****

Рисунок 19 – Добавление нового события в структуру. Элемент, с которым должно быть связано событие, уже должен находиться на графическом пользовательском интерфейсе

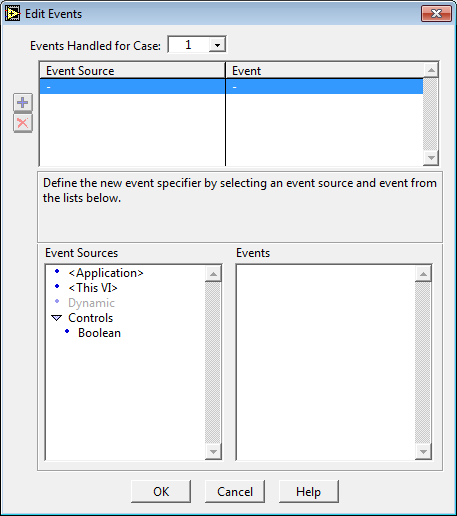
****

Рисунок 20 – Меню настройки нового события. Слева снизу указывается перечень возможных источников событий

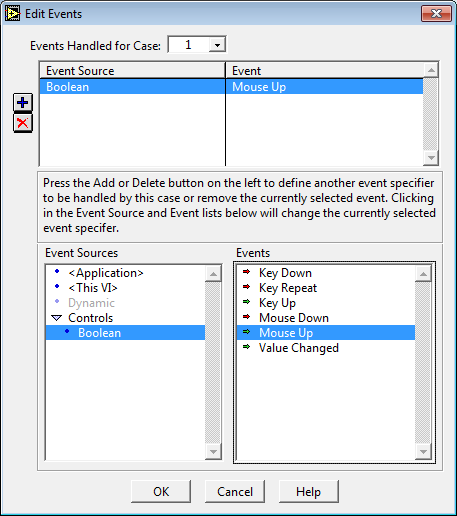
****

Рисунок 21 – Меню настройки события для компонента *Boolean*. Справа снизу указывается вид события, которое необходимо отслеживать (отпускание кнопки мыши после нажатия)

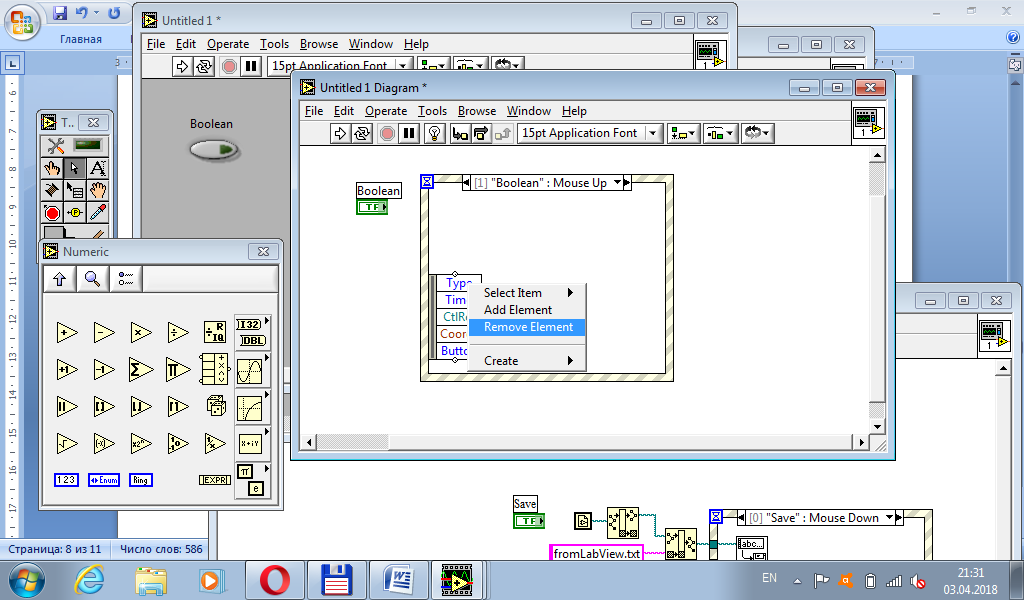
****

Рисунок 22 – Удаление избыточных параметров во вновь созданном событии

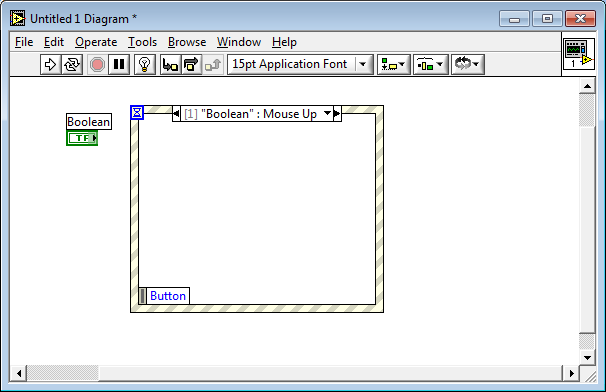
****

Рисунок 23 – Конечный вид события, отслеживающего отпускание кнопки мыши после нажатия на кнопку *Boolean*

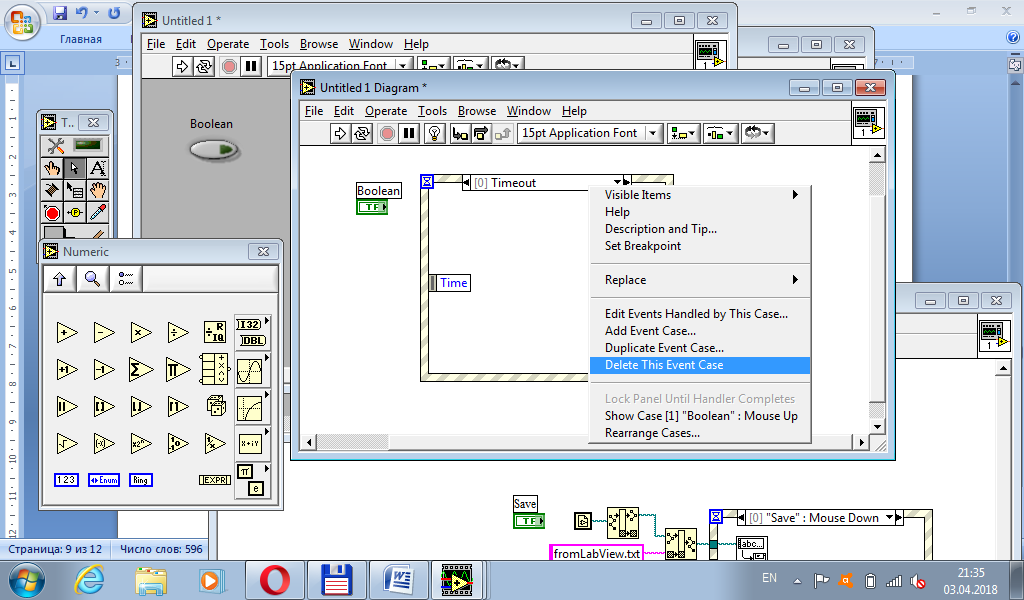
****

Рисунок 24 – Возврат к исходной вкладке, создаваемой по умолчанию в структуре события, для её удаления

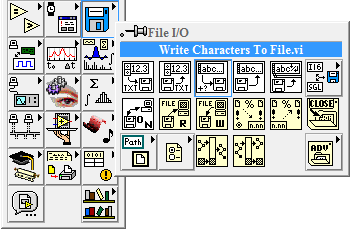


Рисунок 25 – Размещение в меню элемента «Запись символов в файл»

**Реализация виртуального прибора, считывающего данные из файла:**

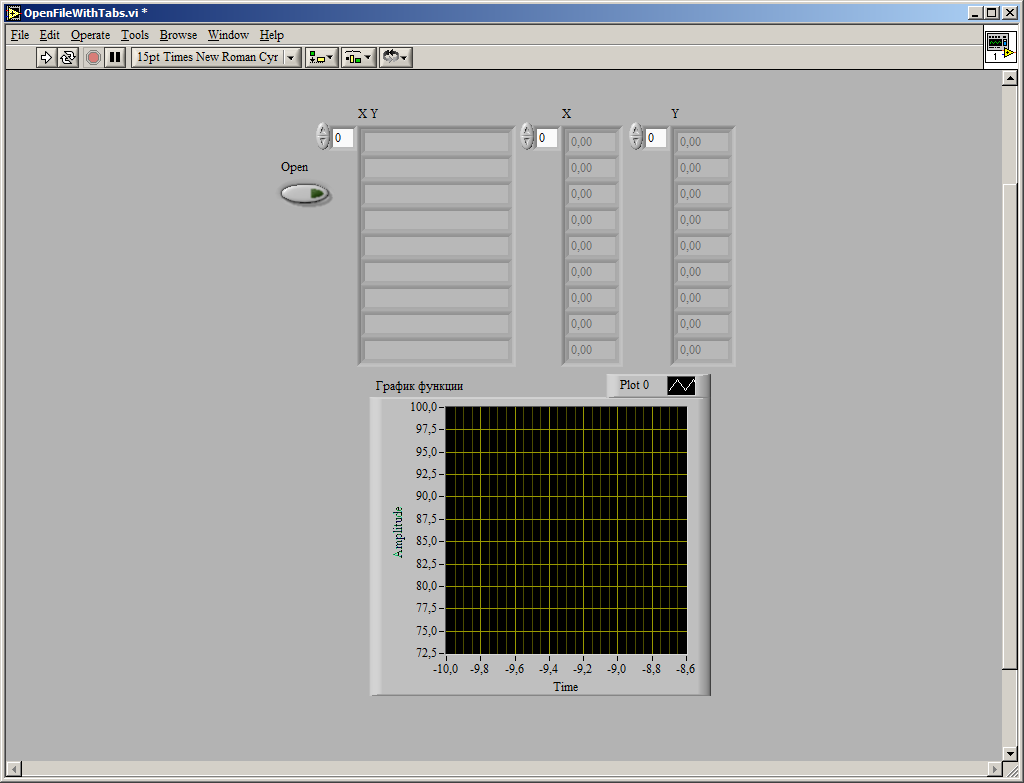


Рисунок 26 – Графический пользовательский интерфейс виртуального прибора, реализующего чтение из файла

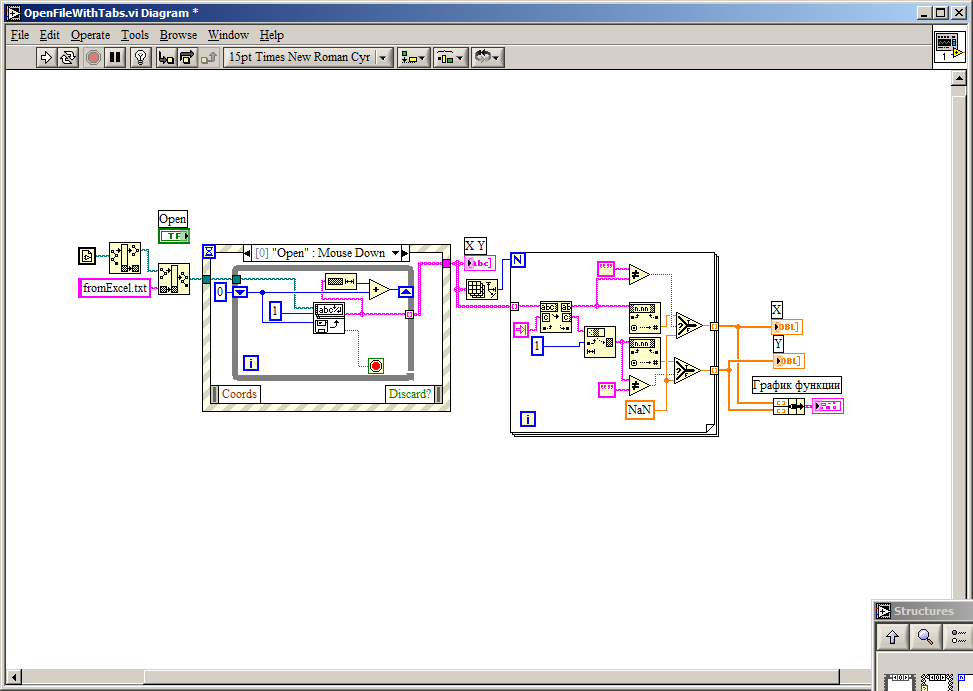


Рисунок 27 – Блок-диаграмма виртуального прибора, реализующего запись в файл

**Варианты функций:**

|  |
| --- |
| 1. **а)** sin(2·x)·ctg(x); **б)** sin(2·x)·cos(4·x); |
| 2. **а)** cos(x)·x2; **б)** log2(x); |
| 3. **а)** 1/sin(x); **б)** x2+2·x+1; |
| 4. **а)** tg(2·x)·cos(2·x); **б)** x2·sin(2·x); |
| 5. **а)** x·sin(2·x)+2·x·cos(x); **б)** sin(x) ·x; |
| 6. **а)** x2/sin(x)+cos(x)/x; **б)** log2(x)·cos(x); |
| 7. **а)** 1/sin(x)+1/cos(x); **б)** x2·sin(x); |
| 8. **а)** ctg(3·x)·x; **б)** tg(2·x)4; |
| 9. **а)** sinh(2·x)+ex·sin(x); **б)** x·lg(x); |
| 10. **а)** x2·cos(x)+ x·sin(x)+x; **б)** lg(x); |
| 11. **а)** sin(a·x)·cos(b·x); **б)** sin(2·x)·tg(x); |
| 12. **а)** sin(2·x)·tg(x); **б)** sin(a·x)·cos(b·x); |
| 13. **а)** lg(x); **б)** x2·cos(x)+ x·sin(x)+x; |
| 14. **а)** x·lg(x); **б)** sinh(2·x)+ex·sin(x); |
| 15. **а)** tg(2·x)4; **б)** ctg(3·x)·x; |
| 16. **а)** x2·sin(x); **б)** 1/sin(x)+1/cos(x); |
| 17. **а)** log2(x)·cos(x); **б)** x2/sin(x)+cos(x)/x; |
| 18. **а)** sin(x) ·x; **б)** x·sin(2·x)+2·x·cos(x); |
| 19. **а)** x2·sin(2·x); **б)** tg(2·x)·cos(2·x); |
| 20. **а)** x2+2·x+1; **б)** 1/sin(x); |
| 21. **а)** log2(x); **б)** cos(x)·x2; |
| 22. **а)** sin(2·x)·cos(4·x); **б)** sin(2·x)· ctg(x); |
| 23. **а)** ctg(5·x); **б)** tg(x)3; |
| 24. **а)** x4·sin(x); **б)** tg(x)·sin(3·x); |
| 25. **а)** x3·cos(x)+x+1; **б)** x/(lg(x) + 1); |
| 26. **а)** sin(x2)· ctg(1/x); **б)** sin(1/(2·x))·cos(1/(3·x)); |
| 27. **а)** sin(1/(a·x))·cos(b·x); **б)** sin(x)·tg(1/x); |
| 28. **а)** sinh(x/e)+ex+1 **б)** 1/(x·lg(x)); |
| 29. **а)** tg(2·x)·cos(2·x)·lg(2·x); **б)** x·sin(x3); |
| 30. **а)** x4+2·x2+x+1; **б)** x/(1+1/x); |

Варианты используемых разделителей при чтении из файла:

|  |  |  |
| --- | --- | --- |
| 1. **;** | 11. **/** | 21. **+** |
| 2. **@** | 12. **~** | 22. **=** |
| 3. **%** | 13. **?** | 23. **№** |
| 4. **&** | 14. **<** | 24. **[** |
| 5. \* | 15. **>** | 25. **]** |
| 6. **!** | 16. **{** | 26. \_ |
| 7. **#** | 17. **}** | 27. **Space** |
| 8. **^** | 18. **(** | 28. **//** |
| 9. **|** | 19. **)** | 29. ++ |
| 10. **\** | 20. **-** | 30. **##** |