

# **Избранные статьи из руководства по программированию и применению контроллеров CANNY®**

**Данный документ автоматически сформирован на  
основании содержания online-энциклопедии CANNY.**

**Актуальная версия документации доступна на сайте**

**<http://wiki.canny.ru/>**

<b>1 CANNY Lab.....</b>	<b>1</b>
1.1 Назначение интегрированной среды разработки CannyLab.....	1
1.2 Минимальные системные требования для работы CannyLab.....	1
1.3 Установка и запуск программы.....	1
1.4 Интерфейс пользователя.....	1
1.5 Работа с файлами.....	3
1.6 Редактор функциональных диаграмм.....	4
1.7 Симулятор CANNY Lab.....	4
1.8 Работа с контроллером.....	5
1.9 Исполняемые файлы автономной загрузки ПО в контроллер.....	5
<b>2 Интерфейс пользователя CANNY Lab.....</b>	<b>6</b>
2.1 Главное окно.....	6
2.2 Главное меню.....	7
2.3 Панели инструментов.....	9
2.4 Рабочая область.....	10
2.5 Строка состояния.....	10
2.6 Панель функциональных блоков.....	11
2.7 Диалог поиска функционального блока по его порядковому номеру.....	11
2.8 Диалог настройки параметров.....	11
2.9 Список горячих клавиш.....	12
<b>3 Редактор функциональных диаграмм.....</b>	<b>14</b>
3.1 Общее описание.....	14
3.2 Добавление функциональных блоков на диаграмму.....	15
3.3 Перемещение функциональных блоков.....	15
3.4 Выделение элементов функциональной диаграммы.....	16
3.5 Удаление элементов функциональной диаграммы.....	16
3.6 Создание соединительной линии.....	17
3.7 Изменение соединительной линии.....	18
3.8 Сегменты и сети соединительных линий.....	19
3.9 Именованные сети.....	20
3.10 Идентификационные номера функциональных блоков.....	21
3.11 Константы.....	22
3.12 Регистры.....	24
<b>4 Симулятор CANNY Lab.....</b>	<b>27</b>
4.1 Включение режима симуляции.....	27
4.2 Непрерывное исполнение диаграммы.....	28

<b>4</b>	<b>Симулятор CANNY Lab</b>	
4.3	Приостановка и сброс	29
4.4	Приостановка по условию	30
4.5	Принудительное изменение значений	31
4.6	Изменение системы счисления отображаемых значений	33
<b>5</b>	<b>Работа с контроллером</b>	<b>34</b>
5.1	Подключение контроллера к персональному компьютеру	34
5.2	Запись системного программного обеспечения в контроллер	34
5.3	Запись в контроллер функциональной диаграммы	35
5.4	Завершение сеанса работы с контроллером	35
5.5	Получение технической информации о подключенном контроллере	35
5.6	Удаление функциональной диаграммы из памяти контроллера	36
5.7	Удаление системного ПО из памяти контроллера	36
<b>6</b>	<b>Исполняемые файлы автономной загрузки ПО в контроллер</b>	<b>37</b>
6.1	Создание SFX-файла	37
6.2	Использование утилиты автоматической загрузки ПО в контроллер	40
<b>7</b>	<b>Язык CFD</b>	<b>42</b>
7.1	Функциональная диаграмма	42
7.2	Функциональный блок	43
7.3	Сеть	44
7.4	Данные	45
7.5	Управление ресурсами контроллера	45
<b>8</b>	<b>Справочник по встроенным функциональным блокам</b>	<b>47</b>
8.1	Константы и регистры	47
<b>9</b>	<b>Установка значения</b>	<b>50</b>
9.1	Особенности	50
9.2	Примеры	50
<b>10</b>	<b>Константа</b>	<b>52</b>
10.1	Особенности	52
10.2	Примеры	52

<b>11 Копирование</b> .....	<b>54</b>
11.1 Особенности.....	54
11.2 Примеры.....	54
<b>12 Регистр чтения</b> .....	<b>55</b>
12.1 Особенности.....	55
12.2 Примеры.....	56
<b>13 Регистр записи</b> .....	<b>57</b>
13.1 Особенности.....	57
13.2 Примеры.....	58
<b>14 Логическое НЕ</b> .....	<b>59</b>
14.1 Особенности.....	59
14.2 Примеры.....	60
<b>15 Логическое И</b> .....	<b>61</b>
15.1 Особенности.....	62
15.2 Примеры.....	62
<b>16 Логическое ИЛИ</b> .....	<b>63</b>
16.1 Особенности.....	64
16.2 Примеры.....	64
<b>17 Логическое исключаящее ИЛИ</b> .....	<b>65</b>
17.1 Примеры.....	66
<b>18 Побитовое И</b> .....	<b>67</b>
18.1 Особенности.....	67
18.2 Примеры.....	68
<b>19 Побитовое ИЛИ</b> .....	<b>69</b>
19.1 Особенности.....	70
19.2 Примеры.....	70
<b>20 Побитовое исключаящее ИЛИ</b> .....	<b>71</b>
20.1 Особенности.....	72
20.2 Примеры.....	72

<b>21 Побитовое НЕ.....</b>	<b>73</b>
21.1 Особенности.....	73
21.2 Примеры.....	74
<b>22 Сдвиг влево.....</b>	<b>75</b>
22.1 Особенности.....	76
22.2 Примеры.....	76
<b>23 Сдвиг вправо.....</b>	<b>77</b>
23.1 Особенности.....	78
23.2 Примеры.....	78
<b>24 Циклический сдвиг влево.....</b>	<b>79</b>
24.1 Особенности.....	80
24.2 Примеры.....	80
<b>25 Циклический сдвиг вправо.....</b>	<b>81</b>
25.1 Особенности.....	82
25.2 Примеры.....	82
<b>26 Равенство.....</b>	<b>83</b>
26.1 Примеры.....	83
<b>27 Больше.....</b>	<b>85</b>
27.1 Примеры.....	86
<b>28 Не больше.....</b>	<b>87</b>
28.1 Примеры.....	88
<b>29 Меньше.....</b>	<b>89</b>
29.1 Примеры.....	90
<b>30 Не меньше.....</b>	<b>91</b>
30.1 Примеры.....	92
<b>31 Сложение.....</b>	<b>93</b>
31.1 Особенности.....	93
31.2 Примеры.....	93

<b>32 Вычитание</b> .....	<b>95</b>
32.1 Особенности.....	95
32.2 Примеры.....	95
<b>33 Умножение</b> .....	<b>97</b>
33.1 Особенности.....	97
33.2 Примеры.....	97
<b>34 Деление</b> .....	<b>99</b>
34.1 Особенности.....	99
34.2 Примеры.....	99
<b>35 Триггер SR</b> .....	<b>101</b>
35.1 Особенности.....	101
35.2 Примеры.....	101
<b>36 Триггер RS</b> .....	<b>103</b>
36.1 Особенности.....	103
36.2 Примеры.....	103
<b>37 Триггер D</b> .....	<b>105</b>
37.1 Особенности.....	105
37.2 Примеры.....	105
<b>38 Триггер T</b> .....	<b>107</b>
38.1 Примеры.....	107
<b>39 Триггер JK</b> .....	<b>108</b>
39.1 Особенности.....	109
39.2 Примеры.....	109
<b>40 Буфер</b> .....	<b>110</b>
40.1 Особенности.....	110
40.2 Примеры.....	110
<b>41 Задержка включения</b> .....	<b>112</b>
41.1 Особенности.....	112
41.2 Примеры.....	113

<b>42</b>	<b>Задержка выключения.....</b>	<b>114</b>
42.1	Особенности.....	114
42.2	Примеры.....	115
<b>43</b>	<b>Генератор импульсов.....</b>	<b>116</b>
43.1	Особенности.....	117
43.2	Примеры.....	117
<b>44</b>	<b>Генератор псевдослучайных чисел.....</b>	<b>118</b>
44.1	Особенности.....	118
44.2	Примеры.....	118
<b>45</b>	<b>Счетчик.....</b>	<b>119</b>
45.1	Особенности.....	119
45.2	Примеры.....	120
<b>46</b>	<b>Детектор переднего фронта.....</b>	<b>121</b>
46.1	Особенности.....	121
46.2	Примеры.....	121
<b>47</b>	<b>Детектор заднего фронта.....</b>	<b>122</b>
47.1	Особенности.....	122
47.2	Примеры.....	122
<b>48</b>	<b>Конвертер 8-в-1.....</b>	<b>123</b>
48.1	Особенности.....	123
48.2	Примеры.....	123
<b>49</b>	<b>Конвертер 16-в-1.....</b>	<b>125</b>
49.1	Особенности.....	125
49.2	Примеры.....	126
<b>50</b>	<b>Конвертер 1-в-8.....</b>	<b>127</b>
50.1	Примеры.....	127
<b>51</b>	<b>Конвертер 1-в-16.....</b>	<b>128</b>
51.1	Примеры.....	128

<b>52 Конвертер строки символов в число</b> .....	<b>130</b>
52.1 Особенности.....	131
52.2 Примеры.....	131
52.3 Примечания.....	131
<b>53 Конвертер числа в строку символов</b> .....	<b>133</b>
53.1 Особенности.....	133
53.2 Примеры.....	134
53.3 Примечания.....	134
<b>54 Отображение</b> .....	<b>135</b>
54.1 Особенности.....	136
54.2 Примеры.....	136
<b>55 Коммутатор 2-в-1</b> .....	<b>137</b>
55.1 Особенности.....	137
55.2 Примеры.....	137
<b>56 Коммутатор 8-в-1</b> .....	<b>138</b>
56.1 Особенности.....	138
56.2 Примеры.....	138
<b>57 Коммутатор 16-в-1</b> .....	<b>140</b>
57.1 Особенности.....	141
57.2 Примеры.....	141
<b>58 Коммутатор 1-в-2</b> .....	<b>142</b>
58.1 Особенности.....	142
58.2 Примеры.....	142
<b>59 Коммутатор 1-в-8</b> .....	<b>143</b>
59.1 Особенности.....	143
59.2 Примеры.....	143
<b>60 Коммутатор 1-в-16</b> .....	<b>145</b>
60.1 Особенности.....	146
60.2 Примеры.....	146



<b>61 Надпись</b> .....	<b>147</b>
61.1 Особенности.....	147
61.2 Примеры.....	147

# 1 CANNY Lab

**CANNY Lab** — интегрированная среда разработки программного обеспечения для программируемых логических контроллеров CANNY 7.

## 1.1 Назначение интегрированной среды разработки CannyLab

Интегрированная среда разработки CannyLab является инструментом разработки прикладных программ с использованием языка функциональных блочных диаграмм CFD и предназначена для написания, отладки и записи программ во внутреннюю память программируемых логических контроллеров CANNY 7, CANNY 5, CANNY 5.2, CANNY 5 nano, CANNY 5 duo.

## 1.2 Минимальные системные требования для работы CannyLab

Тип центрального процессора, тактовая частота процессора:	Intel ® Pentium ® 233MHz или выше.
Операционная система:	Windows ® 2000, Windows XP ™, или выше.
Объем оперативной памяти:	128 Мбайт (256 МБ рекомендуется).
Видеоадаптер и монитор:	SVGA или монитор с более высоким разрешением (1024x768 рекомендуется).
Свободное место на жестком диске:	110 Мбайт (150 МБ рекомендуется).
Устройства ввода:	клавиатура, мышь.
Интерфейс USB:	версии 1.1 или выше.

## 1.3 Установка и запуск программы

Интегрированная среда разработки CannyLab распространяется в варианте "portable", не требующем установки и может запускаться со съемных носителей. Загрузите архив программы, расположенный с интернет-сайта <http://www.canny.ru/c7> и разархивируйте его в папку на своем ПК или на внешнем носителе. Запустите файл **cannylab.exe** из папки программы. Запуск нескольких экземпляров программы не рекомендуется.

## 1.4 Интерфейс пользователя

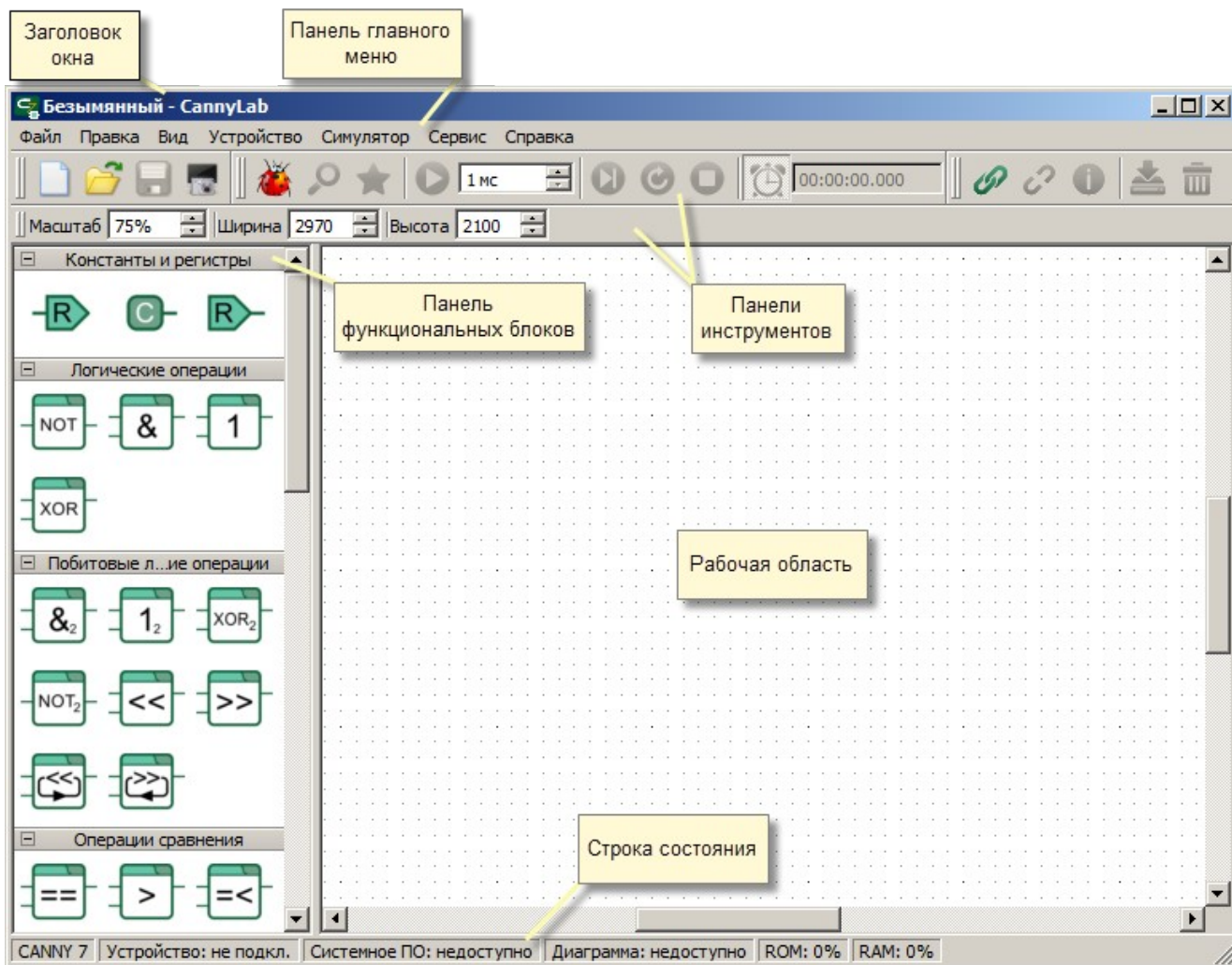
*Основная статья:* **Интерфейс пользователя CANNY Lab**

Интерфейс пользователя CannyLab состоит из единственного окна - главного окна программы.

Главное окно программы делится на несколько областей:

- заголовок окна;
- панель главного меню;
- панели инструментов редактора;
- панель функциональных блоков;
- рабочая область;
- строка состояния.

В зависимости от текущего режима работы редактора те или иные элементы меню и панелей CannyLab могут быть недоступны или скрыты.




## 1.5 Работа с файлами

Интегрированная среда разработки SannyLab поддерживает работу с тремя типами файлов:

- «cfd» файлы функциональных диаграмм;
- «ссх» файлы системного программного обеспечения, с расширением «ссх»;
- «exe» исполняемые файлы автономной загрузки программного обеспечения в контроллер без использования SannyLab.

Файлы функциональных диаграмм и исполняемые файлы для автономной загрузки программного обеспечения в контроллер, могут быть созданы в среде SannyLab. Файлы системного программного обеспечения контроллера поставляются производителем контроллера и могут быть загружены в контроллер из среды SannyLab или использованы при создании исполняемого файла автономной загрузки программного обеспечения в контроллер.


### 1.5.1 Создание нового файла функциональной диаграммы

Создание нового файла функциональной диаграммы производится выбором пункта «Файл» => «Создать» главного меню программы, с помощью комбинации клавиш «Ctrl+N», или при помощи кнопки  панели инструментов.

При внесении в открытый файл изменений, в конце имени файла в заголовке главного окна программы отображается символ «\*». При наличии изменений в открытом файле, во время выхода из редактора или создания нового файла функциональной диаграммы пользователю будет предложено сохранить текущий файл.


### 1.5.2 Сохранение файла функциональной диаграммы

Сохранение файла функциональной диаграммы с новым именем возможно при выборе пункта «Файл» => «Сохранить как...» главного меню программы, при этом имя сохраняемого файла и путь к нему задается во всплывающем диалоговом окне.

Сохранение файла под прежним именем происходит при выборе пункта «Файл» => «Сохранить» главного меню программы, либо при нажатии комбинации клавиш «Ctrl+S» или кнопки  панели инструментов.

Если сохраняемая диаграмма перезапишет файл, созданный в более ранней версии SannyLab, последует предупреждение о том, что этот файл будет невозможно открыть в SannyLab предыдущих версий.

### 1.5.3 Открытие файла функциональной диаграммы

Открытие сохраненного ранее файла диаграммы, возможно при выборе пункта «Файл» => «Открыть...» главного меню программы, нажатием комбинации клавиш «Ctrl+O» или кнопки  панели инструментов, при этом выбор файла для открытия осуществляется пользователем в диалоговом окне.

Кроме того, открыть файл диаграммы можно просто перетащив его в открытое окно CannyLab.

При открытии файла, созданного в более ранней версии CannyLab, будет выведено соответствующее предупреждение. Если при этом текущая версия CannyLab не сможет корректно отобразить всю информацию содержащуюся в файле устаревшего формата, последует предупреждение и об этом.

## 1.6 Редактор функциональных диаграмм

*Основная статья:* **Редактор функциональных диаграмм**

Основным режимом работы CannyLab является режим редактора функциональных диаграмм. В этом режиме, пользователь имеет возможность:

- добавлять на диаграмму функциональные блоки;
- перемещать функциональные блоки в пределах диаграммы;
- выделять и удалять элементы диаграммы;
- добавлять на диаграмму связи между функциональными блоками — сети;
- добавлять на диаграмму и изменять имена сетей;
- объединять, разделять и изменять конфигурацию сетей;
- изменять нумерацию функциональных блоков;
- назначать источники данных входов и получателей данных выходов функциональных блоков, используя константы и регистры ресурсов контроллера.

## 1.7 Симулятор CANNY Lab

*Основная статья:* **Симулятор CANNY Lab**

Работая в режиме симулятора, CannyLab выполняет с данными диаграммы действия, эквивалентные тем, которые будут выполнены с ними вычислительным ядром контроллера при запуске диаграммы на нем, что позволяет моделировать работу контроллера на ПК.

Пользователь имеет возможность запустить симуляцию выполнения функциональной диаграммы для отладки и тестирования на любом этапе ее создания. В режиме симулятора возможно циклическое и пошаговое выполнение диаграммы, задание условий приостановки выполнения, принудительная установка значений входов и выходов функциональных блоков, контроль промежуточных значений.

*Примечание: В текущей версии симулятора SannyLab отсутствует поддержка аппаратных ресурсов контроллера, поэтому результаты чтения из регистра и записи в регистр в симуляторе могут отличаться от результатов полученных при выполнении диаграммы контроллером.*

## 1.8 Работа с контроллером

*Основная статья: **Работа с контроллером***

Интегрированная среда разработки SannyLab предоставляет пользователям возможность загрузки в контроллер системного программного обеспечения, необходимого для выполнения функциональной диаграммы, а так же самой функциональной диаграммы.

## 1.9 Исполняемые файлы автономной загрузки ПО в контроллер

*Основная статья: **Исполняемые файлы автономной загрузки ПО в контроллер***

Интегрированная среда разработки SannyLab предоставляет пользователям возможность создания настраиваемого автономного исполняемого EXE-файла содержащего системное программное обеспечение контроллера и функциональную диаграмму.

Назначением такого файла является быстрое задание параметров работы контроллера в понятном конечному потребителю виде и простая загрузка программного обеспечения в контроллер используя компьютер, на котором не установлена среда SannyLab. В дальнейшем, такой файл будем называть: утилита автоматической загрузки ПО в контроллер или SFX-файл.

## 2 Интерфейс пользователя CANNY Lab

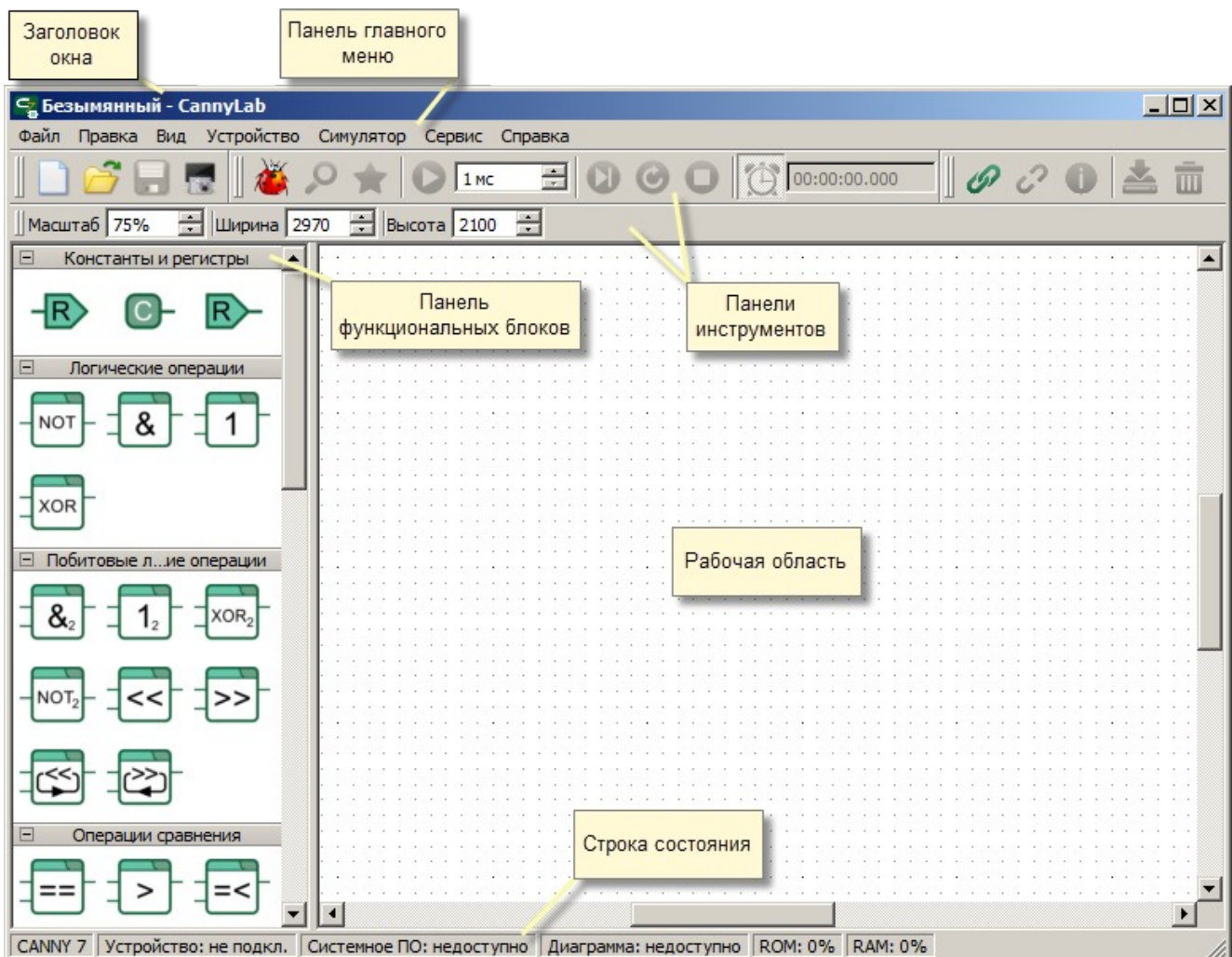
Интерфейс пользователя CannyLab состоит из единственного окна - главного окна программы.

### 2.1 Главное окно

Главное окно программы делится на несколько областей:

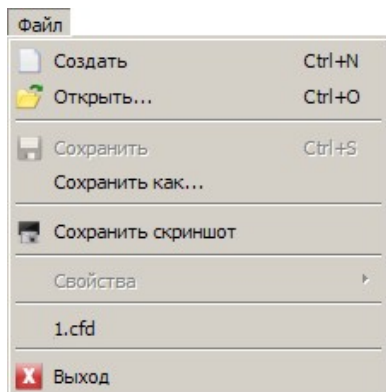
- заголовок окна;
- панель главного меню;
- панели инструментов редактора;
- панель функциональных блоков;
- рабочая область;
- строка состояния.

В зависимости от текущего режима работы редактора те или иные элементы меню и панелей CannyLab могут быть недоступны или скрыты.



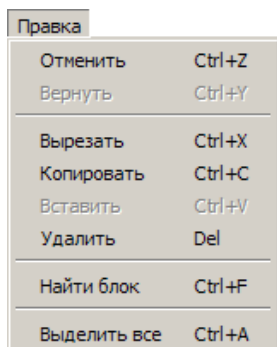
## 2.2 Главное меню

Меню «Файл» содержит команды для работы с файлами CannyLab.



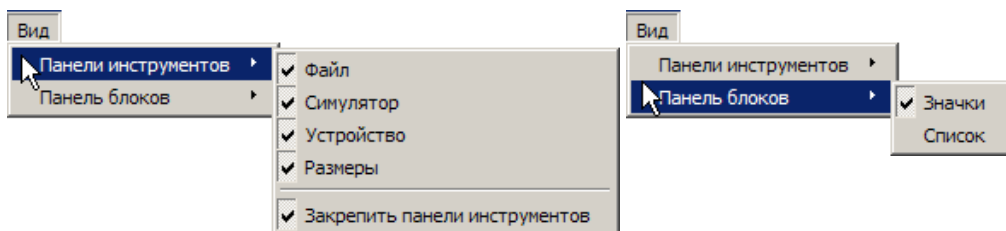
- создание нового файла функциональной диаграммы;
- открытие существующего файла функциональной диаграммы;
- сохранение текущего файла диаграммы;
- сохранение текущего файла диаграммы с новым именем;
- сохранение «снимка» диаграммы в графический файл формата png;
- просмотр свойств текущего файла;
- история недавно открытых файлов (до 5 записей);
- выход из программы.

Меню «Правка» содержит команды управления внешним видом редактора.



- отмена последнего действия, при создании/редактировании схемы;
- возврат последнего действия, при создании/редактировании схемы;
- вырезание выделенной части диаграммы в буфер;
- копирование в буфер выделенной части диаграммы;
- вставка содержащейся в буфере части диаграммы;
- удаление выделенной части диаграммы;
- поиск функционального блока по его порядковому номеру;
- выделение всей диаграммы.

Меню «Вид» содержит настройки отображения панели инструментов и представления панели функциональных блоков.



Меню «Устройство» содержит команды для работы контроллером.



**Устройство**

- Подключить - поиск и установление связи с контроллером подключенным через USB;
- Системное ПО ▶ - действия с системным программным обеспечением контроллера;
- Диаграмма ▶ - действия с пользовательской функциональной диаграммой контроллера;
- Информация - информация о подключенном контроллере;
- Запустить - пункт зарезервирован для реализации в следующих версиях;
- Отключить - завершить сеанс связи с контроллером;

**Системное ПО**

- Записать - загрузить системное программное обеспечение в контроллер;
- Стереть - стереть системное программное обеспечение контроллера;

**Диаграмма**

- Записать - загрузить пользовательскую функциональную диаграмму в контроллер;
- Стереть - стереть пользовательскую функциональную диаграмму из контроллера.

Меню «Симулятор» содержит команды управления режимом программной симуляции работы контроллера в среде CannyLab.

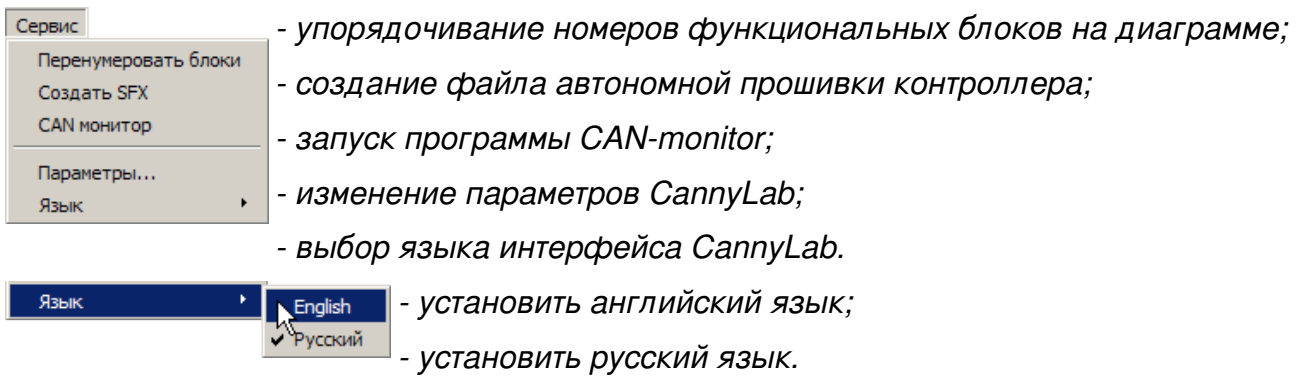
**Симулятор**

- Включить - включение/выключение режима симуляции;
- Показывать блок - включение/выключение удержания фокуса на активном функциональном блоке;
- До изменения - включение/выключение режима приостановки при любом изменении в данных;
- Запуск - запуск/приостановка/продолжение выполнения диаграммы;
- Шаг - выполнение активного функционального блока;
- Цикл - выполнение диаграммы до активного блока;
- Сброс - остановка и сброс симулятора в начальное состояние;
- В реальном времени - привязка масштаба времени симулятора к системному времени windows;
- Формат значений ▶ - установка формата для отображения значений в режиме симуляции;

**Формат значений**

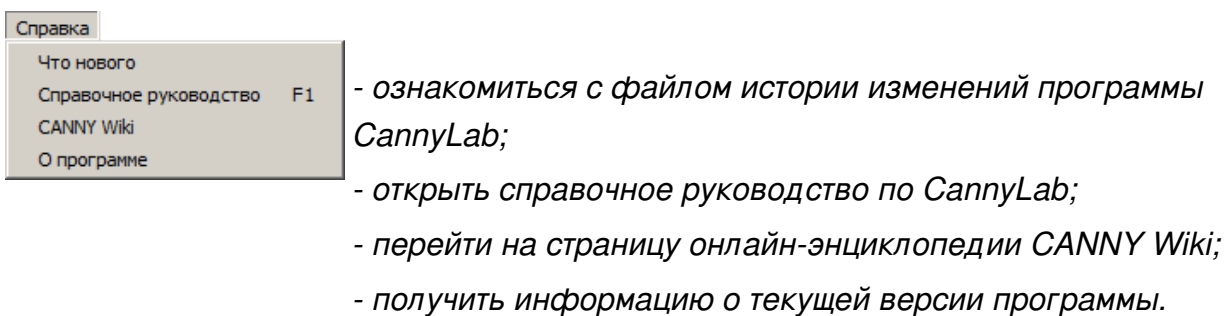
- Dec - установка десятичной системы счисления;
- Hex - установка шестнадцатеричной системы счисления;
- Bin - установка двоичной системы счисления;
- Chr - установка символьного отображения данных.

Меню «Сервис» содержит сервисные команды и настройки CannyLab.



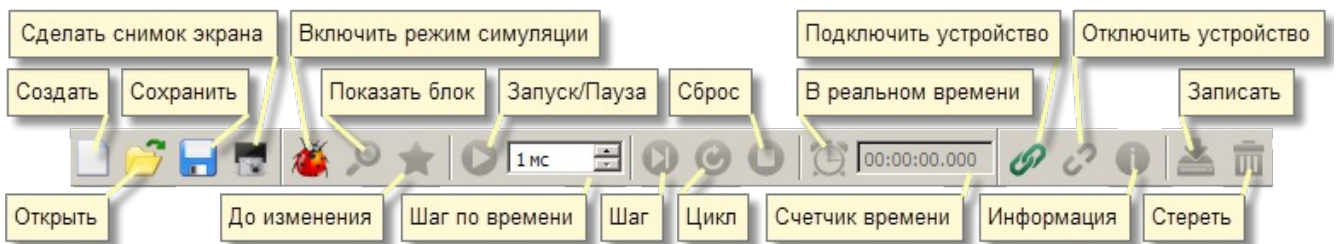
Примечание: Галочкой обозначен текущий установленный язык интерфейса CannyLab.

Меню «Справка» содержит ссылки на данное руководство и информацию о программе.

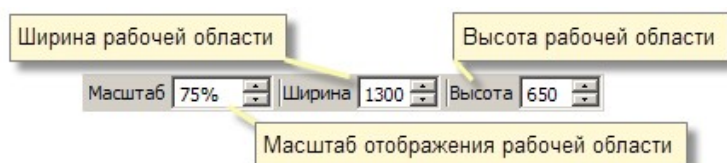


## 2.3 Панели инструментов

Панели инструментов позволяют получить доступ к наиболее востребованным командам при работе с CannyLab.

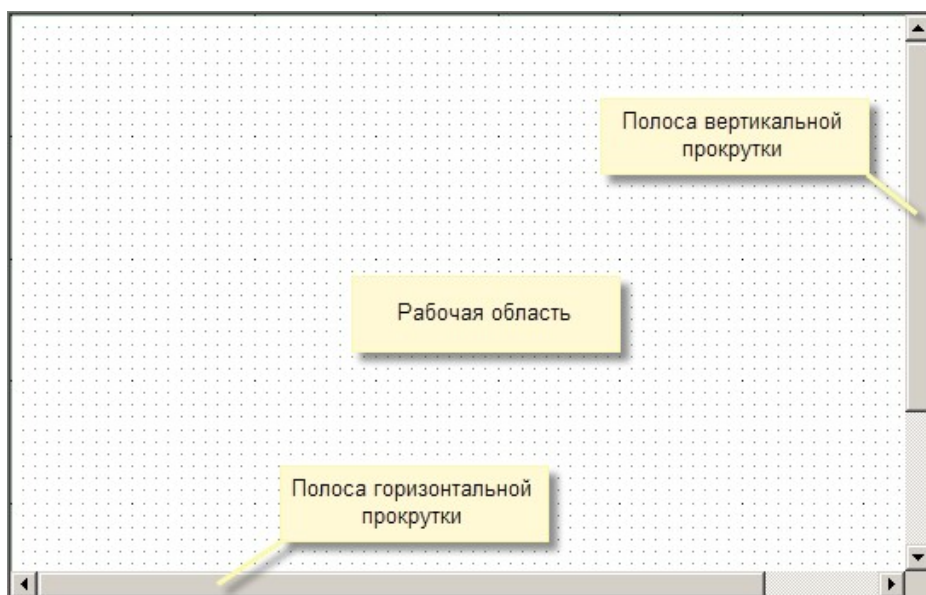


Панель управления рабочим столом позволяет масштабировать изображение рабочей области и изменять его размеры в пикселях. Для быстрого изменения масштаба изображения вращайте колесо мыши удерживая в нажатом состоянии клавишу «Ctrl» клавиатуры.



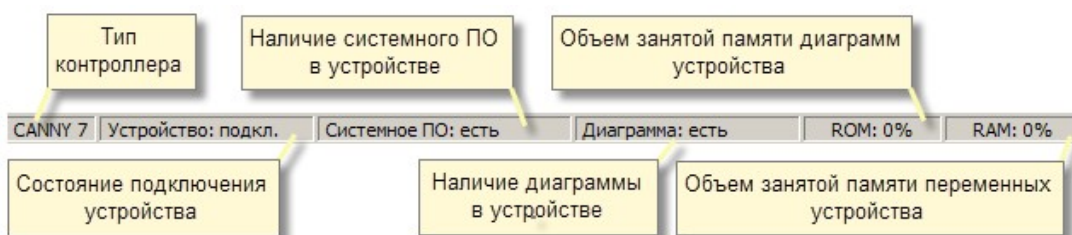
## 2.4 Рабочая область

Рабочая область программы предназначена для отображения, редактирования и отладки функциональных диаграмм. Фоном рабочей области является координатная сетка, облегчающая позиционирование на ней элементов функциональной диаграммы. Для перемещения по диаграмме отображаемой в рабочей области используются вертикальная и горизонтальная полосы прокрутки, колесо мыши или клавиши управления курсором.



## 2.5 Строка состояния

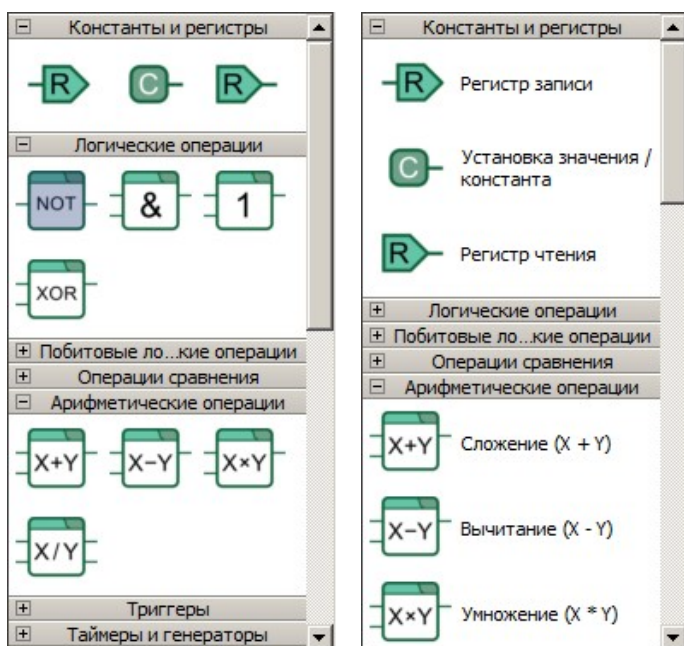
В строке состояния отображаются: для какого типа контроллера предназначена текущая функциональная диаграмма, наличие связи с контроллером, наличие в подключенном контроллере системного программного обеспечения и пользовательской диаграммы. Кроме того, в режиме симуляции в строке состояния отображаются сведения об использованных пользовательской диаграммой ресурсах контроллера.



## 2.6 Панель функциональных блоков

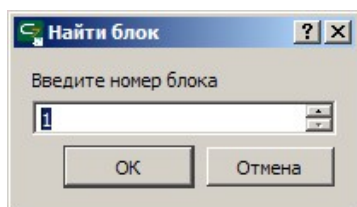
Панель функциональных блоков содержит все доступные для использования при построении функциональных диаграмм типы блоков. Функциональные блоки в данной панели объединены в группы по назначению и, в зависимости от настроек отображения, могут быть представлены в виде значков или списка. Прокрутку панели функциональных блоков можно выполнить при помощи вертикальной полосы прокрутки или колеса мыши.

Два варианта отображения панели функциональных блоков:



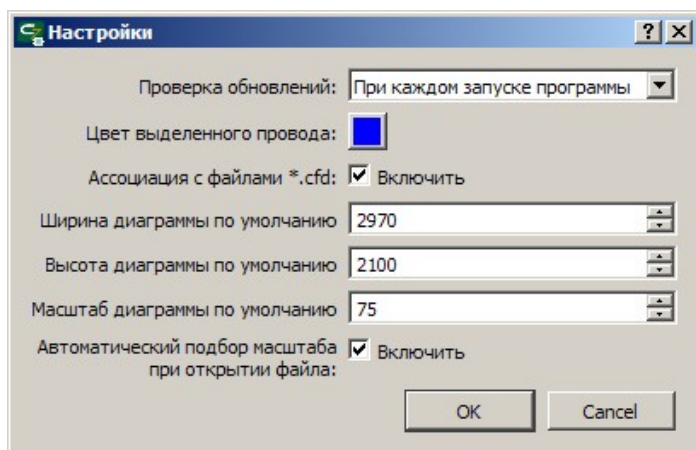
## 2.7 Диалог поиска функционального блока по его порядковому номеру

Доступ к диалогу поиска функционального блока по его порядковому номеру SannyLab можно получить, выбрав пункт «Правка» => «Найти блок» главного меню программы.



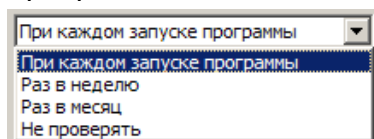
## 2.8 Диалог настройки параметров

Доступ к диалогу настройки параметров SannyLab можно получить, выбрав пункт «Сервис» => «Параметры» главного меню программы.



Проверка обновлений -

Выбор периодичности проверки наличия новых версий программы



Цвет выделенного провода -

Выбор цвета отображения выделенной соединительной линии

Ассоциация с файлами \*.cfd -

Связывает тип файлов функциональных диаграмм с текущим экземпляром CannyLab

Ширина диаграммы по умолчанию -

Ширина области диаграммы при создании нового файла

Высота диаграммы по умолчанию -

Высота области диаграммы при создании нового файла

Масштаб диаграммы по умолчанию -

Масштаб отображения области диаграммы при создании нового файла

Автоматический подбор масштаба при открытии файла -

Способ отображения диаграммы при открытии файла (при включенном подборе масштаба диаграмма, при открытии, будет, по возможности, полностью вписана в экран — отображена в соответствующем масштабе)

## 2.9 Список горячих клавиш

F1	Открыть справочное руководство;
Ctrl+N	Создать новый файл диаграммы;
Ctrl+O	Открыть файл диаграммы;
Ctrl+S	Сохранить файл диаграммы;
Ctrl+A	Выделить все;
Ctrl+Z	Отмена последнего действия при создании и редактировании диаграммы;
Ctrl+Y	Возврат последнего отмененного действия при создании и редактировании диаграммы;

Del	Удаление выделенных элементов диаграммы;
Esc	Выход из режима редактирования сети;
Left, Up, Right, Down	Сдвиг экрана отображения рабочей области влево, вверх, вправо, вниз;
PageUp, PageDown	Сдвиг экрана отображения рабочей области на страницу вверх, вниз;
Ctrl + колесо мыши	Быстрое изменение масштаба отображения рабочей области;
Alt + колесо мыши	Горизонтальная прокрутка рабочей области редактора;
Shift + колесо мыши	Ускоренная вертикальная прокрутка рабочей области редактора;
Нажатое колесо + движение мыши	Перемещение поля диаграммы внутри рабочей области («перетаскивание»).

# 3 Редактор функциональных диаграмм

Основным режимом работы CannyLab является режим редактора функциональных диаграмм.

## 3.1 Общее описание

В режиме редактора функциональных диаграмм, пользователь имеет возможность:

- добавлять на диаграмму функциональные блоки;
- перемещать функциональные блоки в пределах диаграммы;
- выделять и удалять элементы диаграммы;
- добавлять на диаграмму связи между функциональными блоками — сети;
- добавлять на диаграмму и изменять имена сетей;
- объединять, разделять и изменять конфигурацию сетей;
- изменять нумерацию функциональных блоков;
- назначать источники данных входов и получателей данных выходов функциональных блоков, используя константы и регистры ресурсов контроллера.

В редакторе, предусмотрена команда пошаговой отмены последнего выполненного пользователем действия - *UnDo*, которая выполняется при выборе пункта «Правка» => «Отменить» главного меню программы или нажатием комбинации клавиш «Ctrl+Z», а также команда пошагового возврата ранее отмененных действий - *ReDo*, которая выполняется при выборе пункта «Правка» => «Вернуть» главного меню программы или нажатием комбинации клавиш «Ctrl+Y».

Для упрощения разработки пользовательских диаграмм в редакторе предусмотрена возможность использования буфера обмена для вырезания, копирования и вставки фрагментов диаграммы как внутри одного, так и между различными файлами. Доступ к командам «Вырезать», «Копировать» и «Вставить» осуществляется через меню «Правка» или, как и в других приложениях, с помощью привычных сочетаний клавиш «Ctrl+X», «Ctrl+C» и «Ctrl+V» соответственно.

Вертикальная прокрутка рабочей области осуществляется при помощи колеса мыши, клавишами управления курсором *Up* и *Down*, клавишами *PageUp*, *PageDown* или вертикальной полосы прокрутки, горизонтальная — клавишами управления курсором *Left* и *Right*, вращением колеса мыши при нажатой клавише «Alt» или при помощи горизонтальной полосы прокрутки.

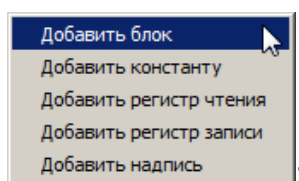
Для быстрого изменения масштаба изображения вращайте колесо мыши удерживая клавишу «Ctrl».

Для быстрого перемещения по полю диаграммы пользователь также может применить метод «*Drag-and-Drop*» (перетаскивание), который позволяет при нажатии и удержании средней кнопки мыши (колесика) передвигать рабочую область редактора с помощью произвольных перемещений мыши, при этом указатель мыши меняет изображение со стрелочки на сжатую в ладонь.

## 3.2 Добавление функциональных блоков на диаграмму

Добавление нового функционального блока на диаграмму, фактически создает новый экземпляр функционального блока выбранного вами типа. Добавление можно осуществить двумя способами:

- выделить блок кликом левой клавиши мыши в панели функциональных блоков, затем кликнуть правой кнопкой мыши в рабочей области диаграммы и выбрать в контекстном меню пункт



- нажать и, удерживая нажатой левую кнопку мыши на блоке в панели функциональных блоков, «перетащить» его в рабочую область редактора диаграмм и отпустить левую кнопку мыши. На время перетаскивания к курсору будет добавлено изображения значка добавляемого блока.

Например:



- вид курсора мыши, при добавлении блока "Логическое И":

## 3.3 Перемещение функциональных блоков

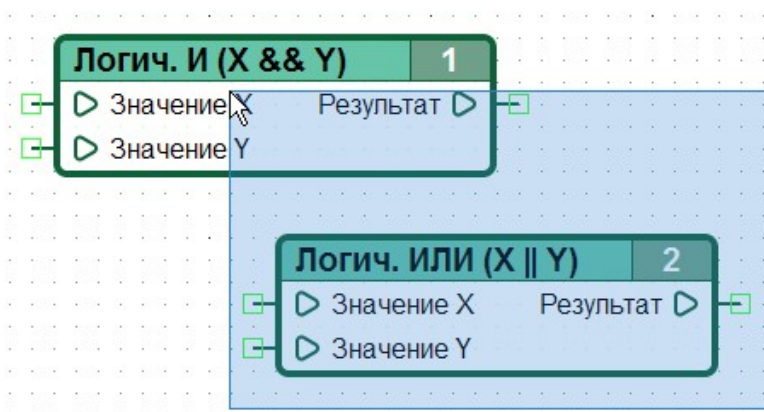
Перемещение функциональных блоков размещенных в рабочей области редактора диаграмм осуществляется «перетаскиванием»: выделив блок размещенный в рабочей области диаграммы кликом левой клавиши мыши (при выделении фоновый цвет блока станет желтым), и удерживая кнопку мыши нажатой, перемещайте указатель мыши. Для окончания перемещения блока отпустите левую кнопку мыши. Таким же образом можно осуществлять одновременное перемещение нескольких элементов диаграммы, при их



групповом выделении.

### 3.4 Выделение элементов функциональной диаграммы

Выделить несколько элементов на диаграмме, возможно «рамкой» – нажав левую кнопку мыши на области диаграммы свободной от элементов и последующим перетаскиванием указателя мыши не отпуская левой кнопки, пока в выделенную область не будут захвачены все подлежащие выделению элементы диаграммы. Даже частичное попадание элемента диаграммы в поле «рамки» приведет к его выделению.



Выбор нескольких объектов можно также выполнить, используя метод выделения указателем мыши с одновременным удержанием в нажатом состоянии клавиши «*Ctrl*». Повторное нажатие левой кнопки мыши на выделенном элементе при нажатой клавише «*Ctrl*» приведет к его исключению из группы выделенных.

Сброс выделения блоков выполняется нажатием левой кнопки мыши, на незанятом элементами диаграммы участке на рабочей области.

### 3.5 Удаление элементов функциональной диаграммы

Удаление функциональных блоков или сетей диаграммы выполняется выделением удаляемого объекта или группы объектов и нажатием на клавиатуре клавиши «*Delete*», либо выбором пункта «*Удалить*» контекстного меню, вызов которого осуществляется нажатием правой кнопки мыши на выделенном элементе диаграммы.

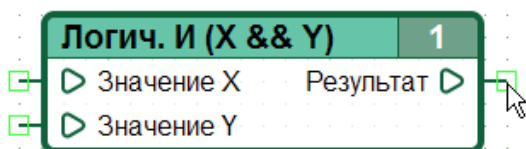
## 3.6 Создание соединительной линии

Построить новую соединительную линию на функциональной диаграмме, возможно:

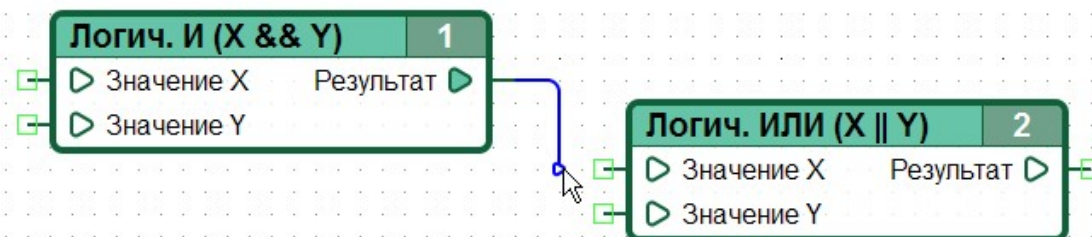
- от выхода одного, к входу того-же или другого функционального блока;
- от входа одного, к выходу того-же или другого функционального блока;
- от входа или выхода функционального блока к уже существующей линии;
- от выхода константы к входу или выходу функционального блока, либо к уже существующей линии.

Кроме того, начав построение линии, можно закончить его в любой свободной от элементов точке диаграммы постановкой конечного узла, образовав таким образом «неподключенную» линию.

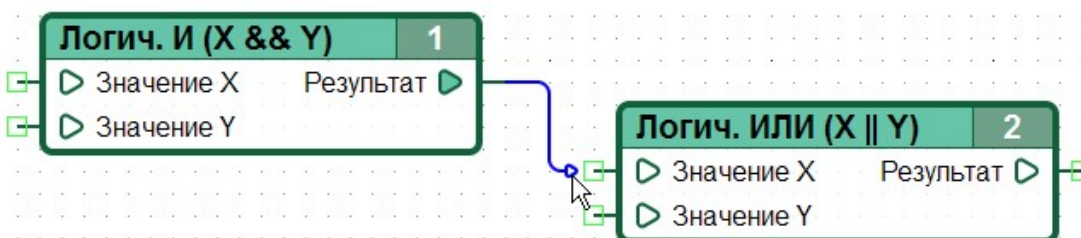
Для начала построения соединительной линии необходимо подвести указатель мыши к точке предполагаемого начала линии, нажать и отпустить левую кнопку мыши.

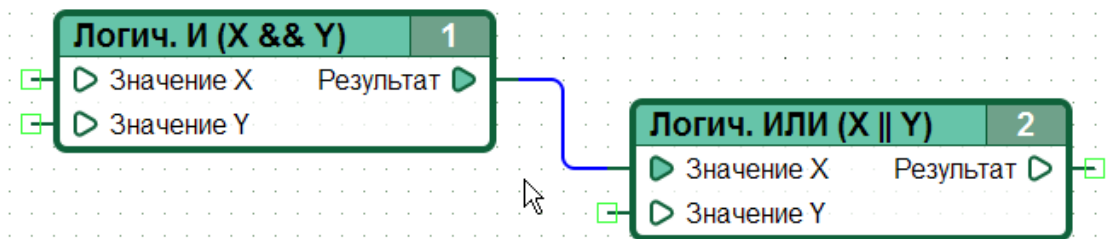


После начала перемещения указателя мыши по рабочей области редактора за ним начнет рисоваться соединительная линия, автоматически добавляя поворот линии, в зависимости от текущего положения указателя мыши.



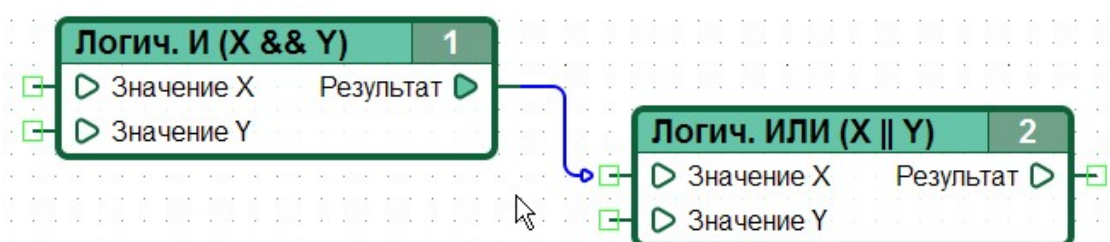
Добавление узла (точки излома) линии или завершения линии в точке ее конечного подключения производится кликом левой кнопки мыши.





Оборвать линию в произвольной точке диаграммы можно двойным кликом левой кнопки мыши в этой точке, отменить построение линии — нажатием кнопки *Esc*.

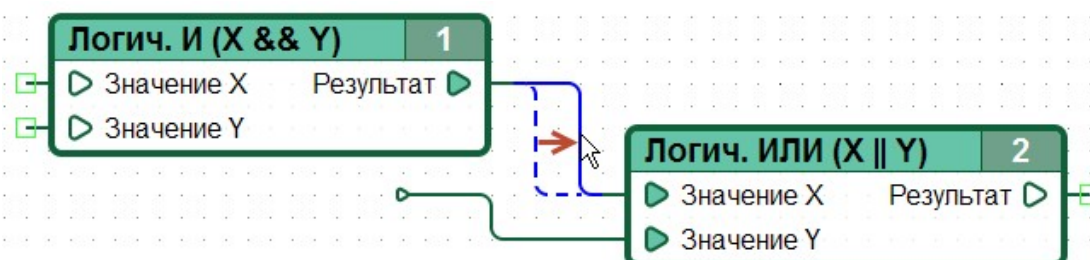
Незавершенная линия оканчивается "интеллектуальной" стрелкой, которая указывает в ту или иную сторону в зависимости от положения линии на плоскости и того, является ли линия источником или приемником данных (также относится к именованным сетям).



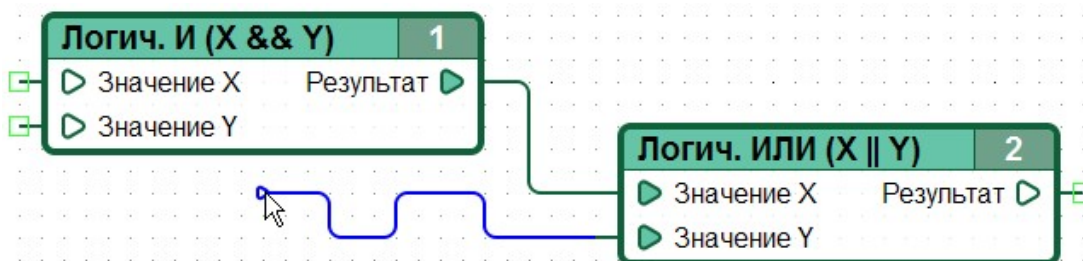
### 3.7 Изменение соединительной линии

Изменение положения и формы соединительной линии производится методом ее «перетаскивания» за прямые отрезки.

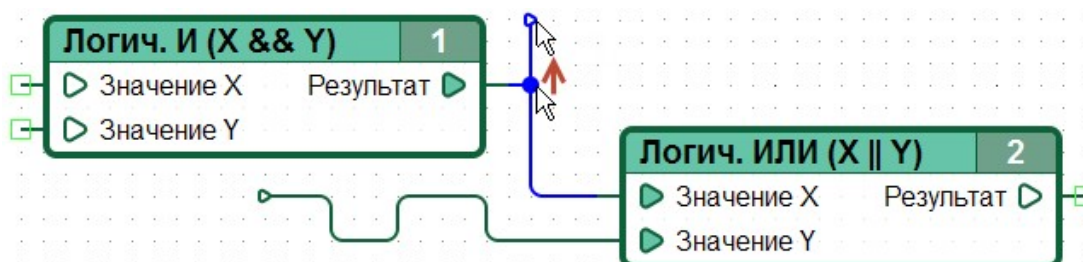
Для изменения соединительной линии необходимо навести указатель мыши на участок требующий изменения, нажать левую кнопку мыши и удерживая ее в нажатом положении, переместить указатель в новое положение, за тем отпустить левую кнопку мыши.



Неподключенные линии можно дорабатывать – продолжать их построение от конечного узла.



Для разветвления линии необходимо кликнуть на нужный узел и продолжить построение линии от данной точки.



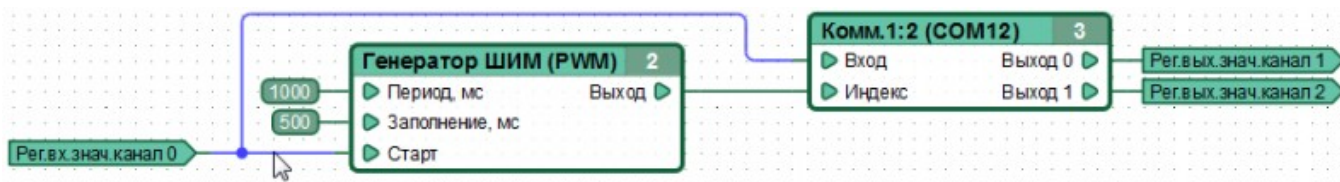
### 3.8 Сегменты и сети соединительных линий

При реализации алгоритма в виде функциональной диаграммы, обычной практикой является связывание одного выхода функционального блока с несколькими входами функциональных блоков, включая свои собственные входы. Таким образом, на диаграмме образуются разветвленные соединительные линии. Соединительную линию, имеющую на функциональной диаграмме ноль или более разветвлений, будем в дальнейшем называть сегментом сети. На рисунке ниже представлено два сегмента.

*Примечание: В соответствии с соглашением принятым в языке CFD, любой выход функционального блока может быть соединен единой сетью с несколькими входами других блоков или своими собственными, но никакой вход функционального блока не может быть соединен с несколькими выходами. Несмотря на то, что редактор диаграмм позволяет производить такие соединения, при попытке сохранения, записи в контроллер или запуске в симуляторе такой диаграммы, CannyLab выведет соответствующее сообщение об ошибке.*



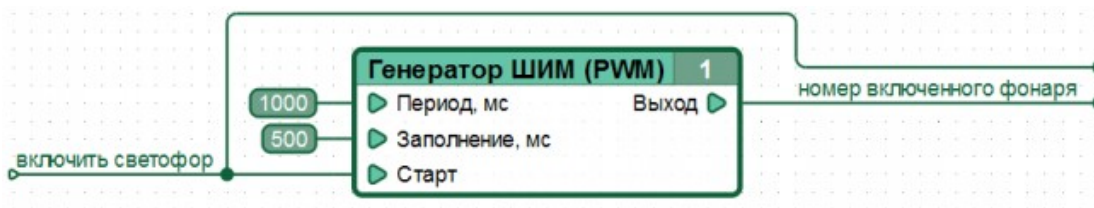
При нажатии левой кнопкой мыши на соединительную линию, принадлежащую какому-либо сегменту, происходит выделение всего сегмента.



Удаление сегмента происходит аналогично удалению любого другого элемента функциональной диаграммы.

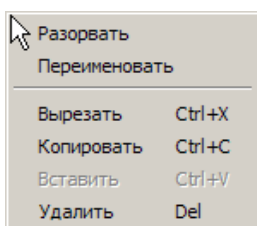
### 3.9 Именованные сети

Для удобства работы с большими функциональными диаграммами, в редакторе предусмотрено создание именованных сетей: пользователь, по своему усмотрению, может назначить тому или иному сегменту собственное имя. Визуально не связанные сегменты имеющие одинаковые имена будут логически объединены, как если бы они были связаны визуально и представляли собой единый сегмент. Такое объединение будем называть сетью. Сеть соединительных линий может состоять из одного или нескольких сегментов.



Использование именованных сетей позволяет сократить общую протяженность и число пересечений соединительных линий на диаграмме, визуально обособить группы блоков реализующих одну функцию, чем повысить читаемость диаграммы.

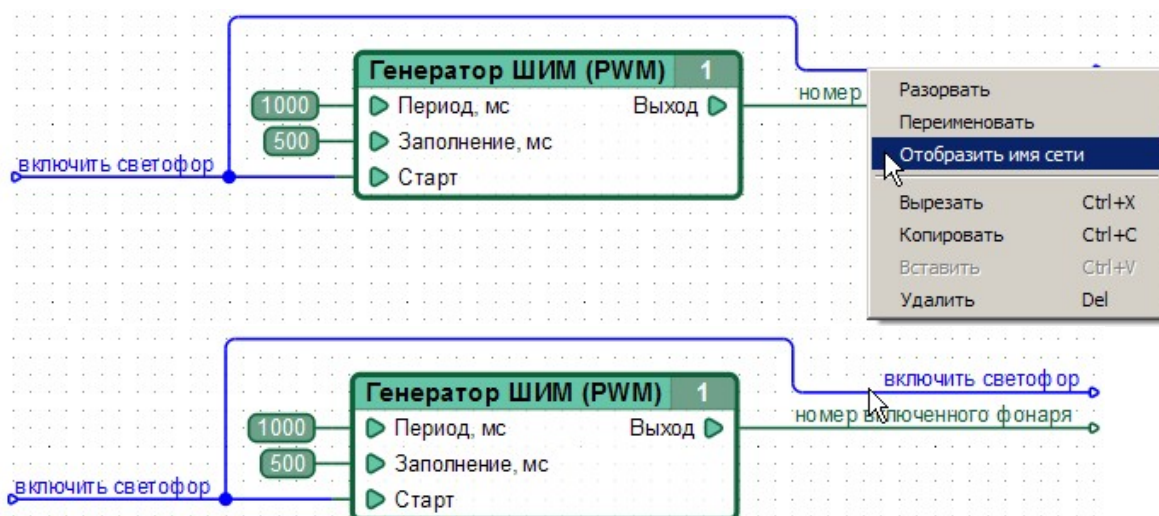
Присвоение сети собственного имени возможно осуществить путем двойного нажатия на левую кнопку мыши, при установке указателя на именуемую сеть, либо выбором пункта «Переименовать» контекстного меню.



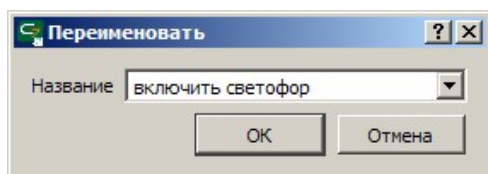
- разорвать сегмент, удалив отрезок между ближайшими узлами;
- задать новое имя данной сети на диаграмме;
- вырезать выделенный элемент или сегменты;
- копировать выделенный сегмент или сегменты;
- вставить скопированный или вырезанный сегмент или сегменты;
- удалить выделенный сегмент или сегменты.

После задания имени сети, в контекстное меню связанное с ней добавляется пункт «Отобразить имя сети», при выборе которого, на диаграмму в текущей точке добавляется

еще одна надпись с именем сети. Если имя сети используется в нескольких местах на диаграмме, то в контекстном меню дополнительно отображаются пункты «Переименовать всю сеть» и «Выделить всю сеть».



Задание имени сети выполняется с помощью диалогового окна, в текстовом поле которого можно ввести новое имя, либо выбрать из заданных ранее, текущего сегмента сети. Переименование всей сети выполняется аналогично.



### 3.10 Идентификационные номера функциональных блоков

Функциональные блоки имеют свои уникальные идентификационные номера, которые присваиваются блоку при его добавлении на диаграмму. Новому блоку назначается первый свободный номер, в диапазоне чисел от единицы до максимального номера блока присутствующего на диаграмме, плюс единица.

Идентификационный номер блока сохраняется в файл диаграммы, и остается неизменным при последующем обращении к данному файлу.

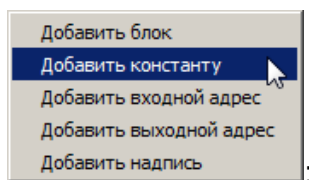
Для упорядочения номеров блоков в любой момент можно воспользоваться пунктом «Сервис» => «Перенумеровать блоки» главного меню программы, при выборе которого, блоки будут перенумерованы в соответствии с очередностью их выполнения, при исполнении контроллером диаграммы в текущем ее виде.

## 3.11 Константы


В качестве входных значений функциональных блоков могут быть использованы числовые константы.

Добавление новой константы на диаграмму можно осуществить следующими способами:

- кликнуть правой кнопкой мыши в рабочей области диаграммы и выбрать в контекстном меню пункт

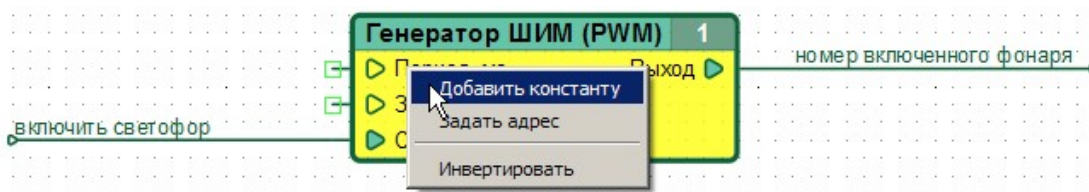


- нажать и, удерживая нажатой левую кнопку мыши на элементе "Константа" в панели функциональных блоков, «перетащить» его в рабочую область редактора диаграмм и отпустить левую кнопку мыши. На время перетаскивания к курсору

будет добавлено изображение соответствующего значка: 

Кроме того, константа может быть добавлена непосредственно к **свободному** входу любого функционального блока.

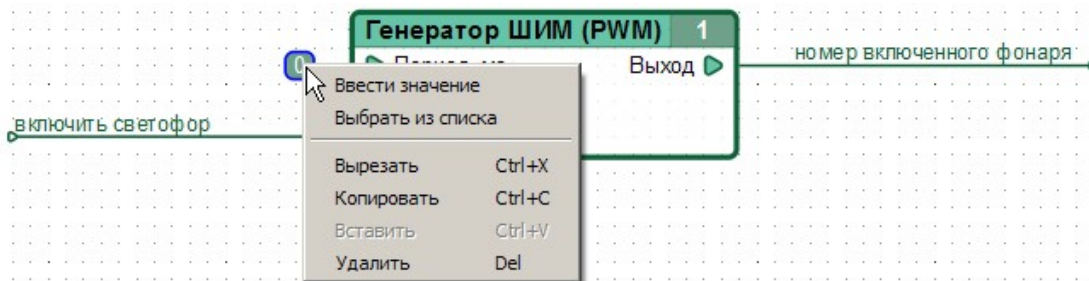
Для задания константы, сначала измените источник данных входа функционального блока, выбрав пункт «Константа» в контекстном меню, вызов которого осуществляется при наведении указателя на вывод блока и нажатии правой кнопки мыши.



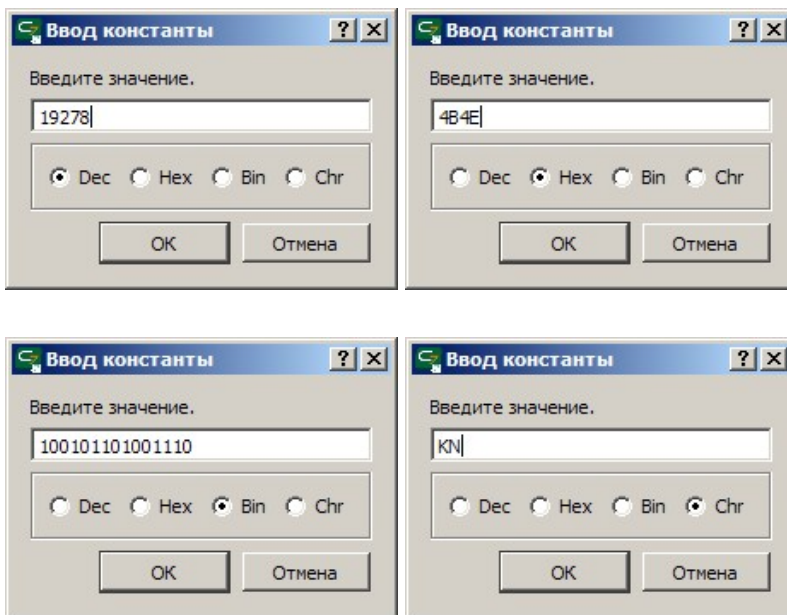
После чего, внешний вид входа блока изменится, отобразив связанную с ним константу. По умолчанию, значение только что добавленной константы равно нулю.



Изменить значение константы, можно воспользовавшись контекстным меню, доступным по нажатию на константе правой кнопки мыши.



При выборе пункта «Ввести значение» пользователь может задать константе числовое значение с помощью диалогового окна. Ввод значения может осуществляться в десятичном, шестнадцатеричном, двоичном формате или в символьном виде, при этом отображение введенного значения на диаграмме будет выполнено в выбранном формате.



Десятичное значение может быть задано 1..5-значным числом в интервале от 0 до 65535, шестнадцатеричное — 1..4-значным числом в интервале от 0 до FFFF (допускается вводить символы как в верхнем так и в нижнем регистре), двоичное — 1..16-значным числом в интервале от 0 до 1111111111111111, а символьное — символами основной таблицы ASCII.

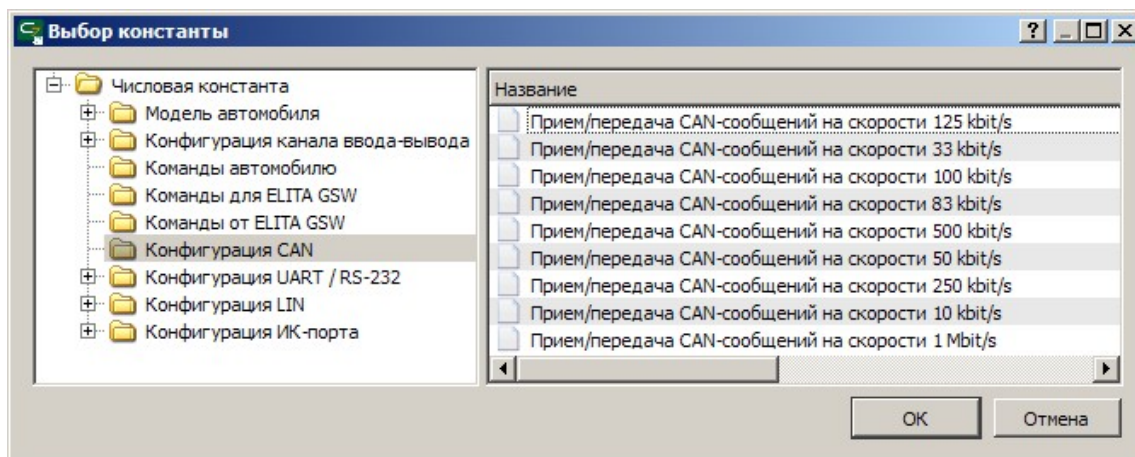
*Примечание: При вводе символьных значений допускается использование специальных символов в следующем формате: «\n» - код символа ASCII 0x0A (перевод строки), «\r» - код символа ASCII 0x0D (возврат каретки), «\t» - код символа ASCII 0x09 (табуляция), «\0» - код символа ASCII 0x00 (ноль, пустое значение), «\xYY» любой символ, где YY – его ASCII код в шестнадцатеричном формате.*

При изменении пользователем формата отображения константы, ее текущее значение будет автоматически преобразовано в выбранную систему исчисления.



Для конфигурации и работы с ресурсами контроллера необходимо использовать специальные именованные константы, доступ к их списку производится выбором соответствующего пункта контекстного меню константы. Выбор именованной константы осуществляется в диалоговом окне.

Именованные константы представлены в виде дерева и сгруппированы по назначению.



При выборе именованной константы, на диаграмме вместо значения будет отображено ее сокращенное наименование.

Стандартный отрицательный вход

Быстрый доступ к списку именованных констант можно получить, используя двойное нажатие левой кнопки мыши на поле константы, при удержании в нажатом состоянии клавиши «Ctrl».

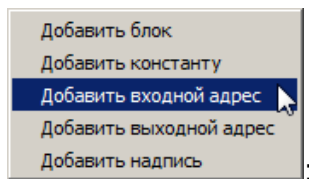
*Примечание: Несмотря на то, что каждая из именованных констант имеет вполне определенное числовое значение, использование именованных констант кроме как для конфигурации и использования связанных с ними ресурсов контроллера не желательно, так как это затруднит чтение и понимание диаграммы.*

## 3.12 Регистры

Доступ к аппаратным ресурсам контроллера из функциональной диаграммы производится путем обращения к определенным регистрам (адресам) в памяти контроллера, отображающим состояние его ресурсов и принимающим данные для управления этими ресурсами. Регистры разделены на предназначенные только для чтения или только для записи.

Добавление нового регистра чтения или записи на диаграмму можно осуществить следующими способами:

- кликнуть правой кнопкой мыши в рабочей области диаграммы и выбрать в контекстном меню пункт

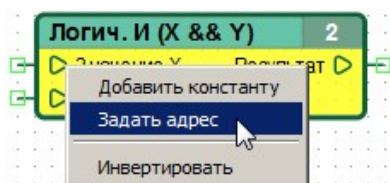


- нажать и, удерживая нажатой левую кнопку мыши на элементе "Регистр чтения" или "Регистр записи" в панели функциональных блоков, «перетащить» его в рабочую область редактора диаграмм и отпустить левую кнопку мыши. На время перетаскивания к курсору будет добавлено изображение значка добавляемого элемента.



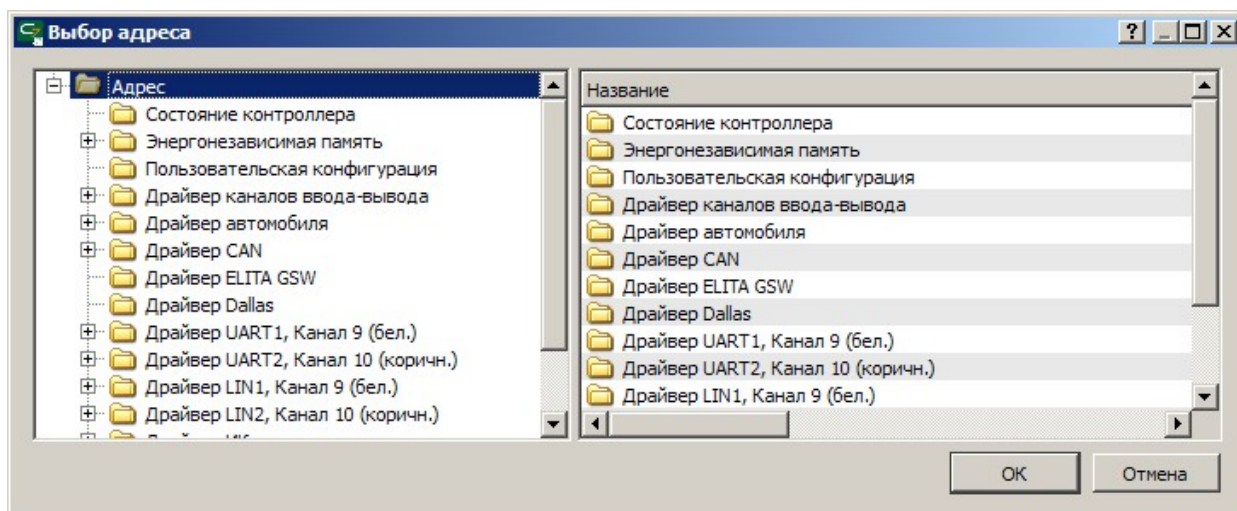
Например, добавление регистра чтения:

Кроме того, регистр можно добавить непосредственно к **свободному** входу или выходу любого функционального блока, кликнув на нем правой кнопкой мыши и выбрав нужный пункт контекстного меню.

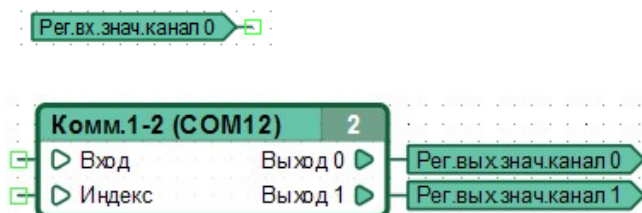


После выбора регистра в качестве источника или приемника данных, открывается диалоговое окно, в котором список доступных для выбора регистров сгруппирован по функциональному назначению.

В отличие от константы, регистр нельзя ввести вручную, допускается только выбор из списка.



При выборе регистра, на диаграмме будет отображено его сокращенное наименование.



*Примечание: В соответствии с соглашением принятым в языке CFD, использовать один и то-же регистр записи в нескольких местах диаграммы не допускается. Несмотря на то, что редактор диаграмм позволяет дублирование регистров записи на диаграмме, при попытке сохранения, загрузки в контроллер или запуске в симуляторе такой диаграммы, CannyLab выведет соответствующее сообщение об ошибке.*

По способу доступа, регистры делятся на регистры чтения и регистры записи. Регистр чтения может являться источником данных для других элементов диаграммы, а регистр записи – приемником данных.


# 4 Симулятор CANNY Lab

Работая в режиме симулятора, CannyLab выполняет с данными диаграммы действия, эквивалентные тем, которые будут выполнены с ними вычислительным ядром контроллера при запуске диаграммы на нем, что позволяет моделировать работу контроллера на ПК.

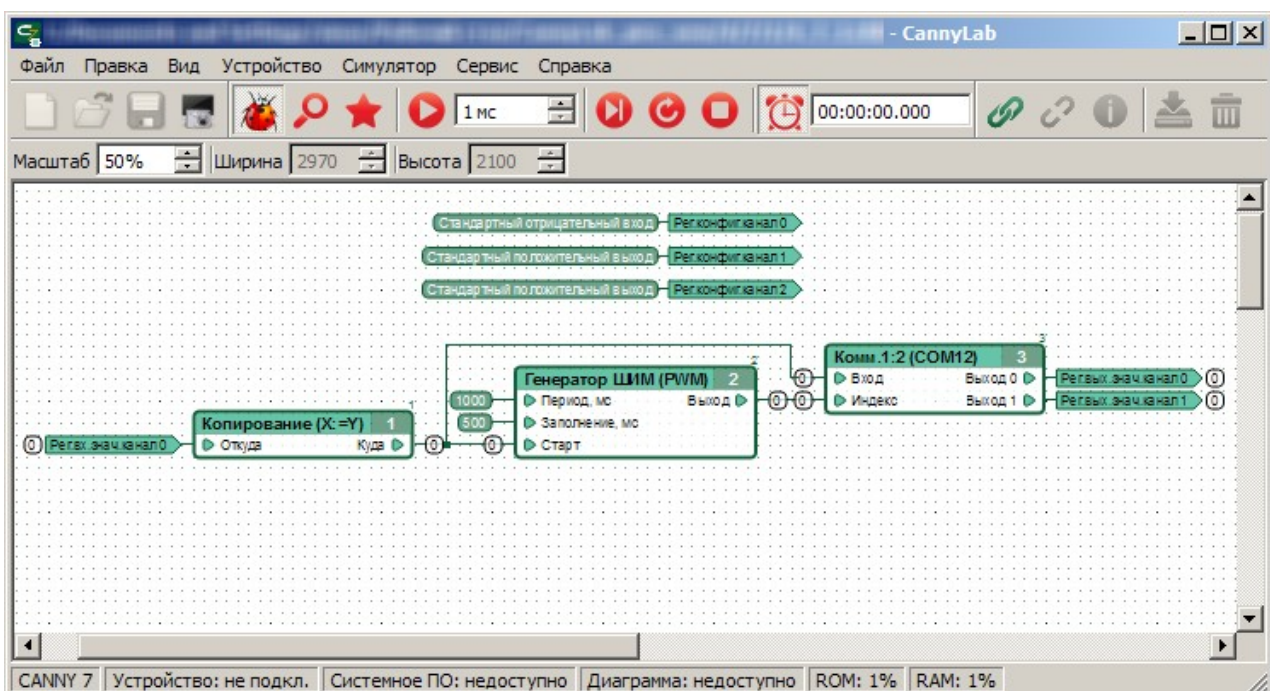
Пользователь имеет возможность запустить симуляцию выполнения функциональной диаграммы для отладки и тестирования на любом этапе ее создания. В режиме симулятора возможно циклическое и пошаговое выполнение диаграммы, задание условий приостановки выполнения, принудительная установка значений входов и выходов функциональных блоков, контроль промежуточных значений.

*Примечание: В текущей версии симулятора CannyLab отсутствует поддержка аппаратных ресурсов контроллера, поэтому результаты чтения из регистра и записи в регистр в симуляторе могут отличаться от результатов полученных при выполнении диаграммы контроллером.*

## 4.1 Включение режима симуляции

Включение и выключение режима симуляции производится выбором соответствующего пункта в разделе «Симуляция» главного меню программы, или нажатием кнопки  панели инструментов.

В режиме симуляции внешний вид главного окна программы и диаграммы изменяется — убираются недоступные в этом режиме элементы диалога, на диаграмму добавляется отладочная информация.







При включении режима симуляции, будет выполнена проверка диаграммы на непротиворечивость и подсчет оперативной памяти (RAM) и памяти программ (ROM) которая потребуется контроллеру для ее запуска. В случае обнаружения проблем пользователь будет о них предупрежден.


Также, при включении режима симуляции будет автоматически определен и отображен порядок исполнения блоков диаграммы. Все промежуточные значения входов и выходов функциональных блоков, внутренняя память блоков и значения всех сетей будут сброшены в 0, указатель следующего блока в очереди выполнения будет установлен на первый блок диаграммы, установлена привязка часов симулятора к реальному времени.


В режиме симуляции, пользователь может свободно перемещаться по диаграмме и изменять масштаб ее отображения, однако редактирование элементов диаграммы в этом режиме работы CannyLab запрещено.

## 4.2 Непрерывное исполнение диаграммы

Запуск непрерывного исполнения диаграммы симулятором производится выбором пункта «Симулятор» => «Запуск» главного меню программы или нажатием кнопки  панели инструментов, которая при этом принимает следующий вид .

*Примечание: Пункты «Запуск» и «Пауза» главного меню «Симулятор», а также кнопки  и  панели инструментов совмещены и активируются в зависимости от текущего состояния симулятора.*

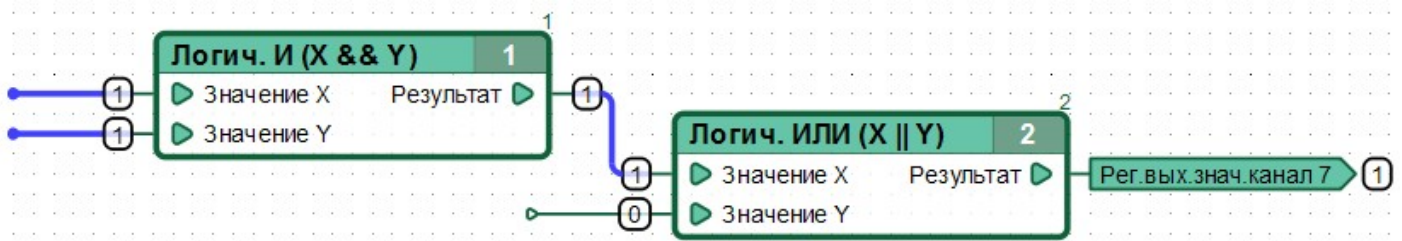
При привязке часов симулятора к реальному времени , которая по умолчанию включена в режиме симуляции, часы симулятора `00:00:00.013` начинают отсчет времени циклического выполнения диаграммы со скоростью системных часов ПК. Если эту привязку отключить, то внутренние часы симулятора начнут идти с максимальной возможной скоростью, определяемой производительностью ПК.

Масштаб времени диаграммы  задает приращение, которое будет добавляться к показаниям часов симулятора после каждого прохода диаграммы. Данный параметр можно рассматривать как время, затрачиваемое контроллером на однократное полное исполнение пользовательской диаграммы.



*Примечание: Для отладки больших пользовательских диаграмм, реализующих алгоритмы критичные к времени исполнения, для ПЛК CANNY 7 масштаб времени рекомендуется устанавливать из расчета 1мс на каждые 30 функциональных блоков.*

При изменении значений входов и выходов функциональных блоков в процессе исполнения диаграммы их новые значения будут отображены на диаграмме непосредственно рядом с ними. Для наглядности, сети связанные с выходами блоков



имеющими значение отличное от нуля, будут выделены цветом.

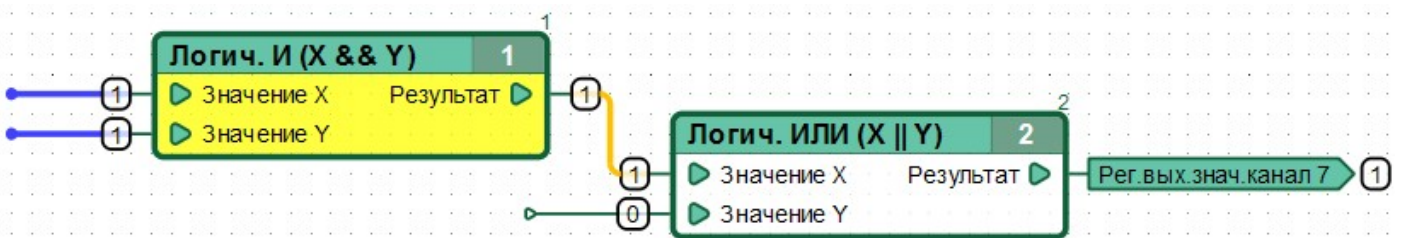



## 4.3 Приостановка и сброс


Исполнение диаграммы симулятором можно приостановить, выбрав пункт «Симулятор» => «Пауза» главного меню программы или нажатием кнопки  панели инструментов, которая при этом принимает следующий вид .


Функциональный блок, на котором исполнение диаграммы было приостановлено, будет выделен желтым цветом. Сети, с изменившимися на последнем шаге значениями так же будут выделены, часы симулятора будут приостановлены.

*Примечание: Пункты «Запуск» и «Пауза» главного меню «Симулятор», а также кнопки  и  панели инструментов совмещены и активируются в зависимости от текущего состояния симулятора.*



При включении опции автоматического слежения за текущим блоком диаграммы, выбором пункта «Симулятор» => «Показывать блок» главного меню программы или нажатием кнопки  панели инструментов, отображение диаграммы в рабочей области будет смещено таким образом, чтобы в рабочую область попал функциональный блок, на котором выполнение было приостановлено.


Продолжение непрерывного исполнения диаграммы с момента, на котором оно было прервано, производится выбором пункта «Симулятор» => «Запуск» главного меню программы или нажатием кнопки .


Полная остановка процесса исполнения диаграммы и сброс симулятора в начальное состояние производится выбором пункта «Симулятор» => «Сброс» главного меню программы или нажатием кнопки .


## 4.4 Приостановка по условию

В симуляторе предусмотрена возможность автоматической приостановки исполнения диаграммы по следующим условиям:

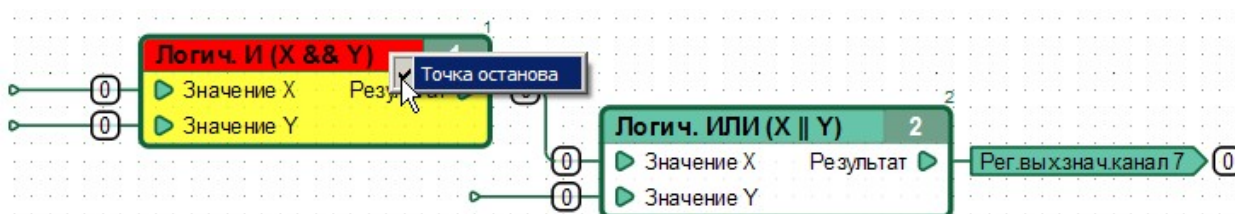
- при изменении любого значения на диаграмме;
- при изменении значения конкретной сети на диаграмме;
- после исполнения очередного функционального блока диаграммы (пошаговый режим);
- перед исполнением конкретного функционального блока на диаграмме;
- по завершению полного цикла диаграммы;

Включение и выключение опции автоматической приостановки исполнения при любом изменении данных производится выбором пункта «Симулятор» => «До изменения» главного меню программы или нажатием кнопки  панели инструментов.

Исполнение очередного функционального блока с последующей приостановкой исполнения диаграммы производится выбором пункта «Симулятор» => «Шаг» главного меню программы или нажатием кнопки  панели инструментов.

Выполнение одного полного цикла диаграммы с приостановкой исполнения на текущем функциональном блоке производится выбором пункта «Симулятор» => «Цикл» главного меню программы или нажатием кнопки  панели инструментов.

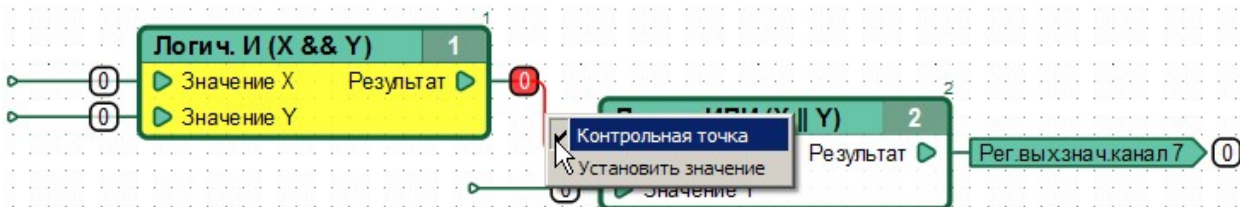
Установка и снятие точки останова на конкретном функциональном блоке диаграммы производится выбором соответствующего пункта контекстного меню, доступного по нажатию правой кнопки мыши на блоке. Фон заголовка функционального блока объявленного точкой останова, в режиме симулятора будет выделен красным цветом.



При наличии на диаграмме функциональных блоков объявленных точками останова, исполнение диаграммы будет приостановлено перед каждым исполнением каждого такого блока.

Установка и снятие точки останова на конкретной сети диаграммы производится выбором соответствующего пункта контекстного меню, доступного по нажатию правой кнопки мыши на любом сегменте сети. Цвет всех сегментов сети объявленной точкой

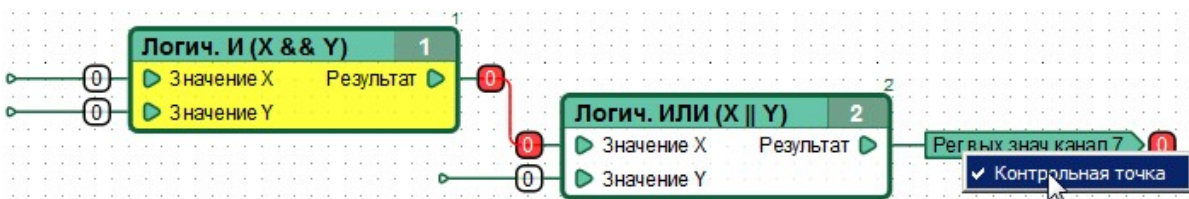
останова, в режиме симулятора будет изменен на красный.



При наличии на диаграмме сетей объявленных точками останова, исполнение диаграммы будет приостановлено при каждом изменении значения каждой такой сети.

*Примечание: Установить или снять точку останова на сети или функциональном блоке можно также при помощи двойного нажатия левой клавиши мыши на этих элементах диаграммы.*

Точка останова может быть также установлена на регистр записи. В этом случае приостановка исполнения диаграммы произойдет при изменении значения данного регистра.



Снятие всех ранее установленных точек останова исполнения диаграммы может быть осуществлено путем нажатия правой кнопки мыши на свободной от каких-либо элементов диаграммы месте рабочей области симулятора и выбора соответствующего пункта контекстного меню.



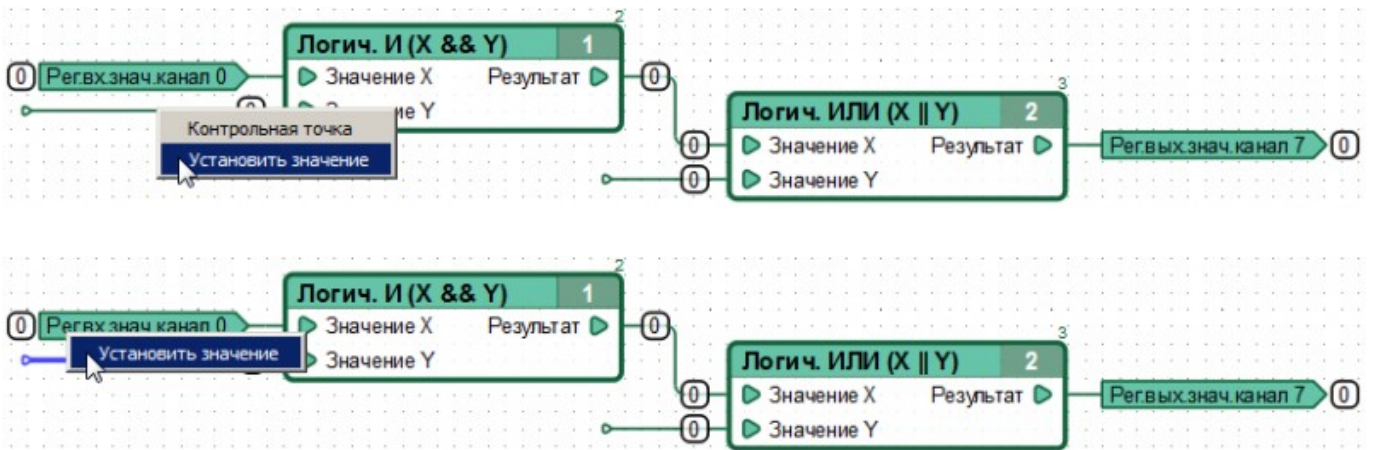
*Примечание: При выходе из симулятора контрольные точки, активные в данной сессии симуляции, сохраняются и будут доступны при следующем запуске симулятора.*

## 4.5 Принудительное изменение значений

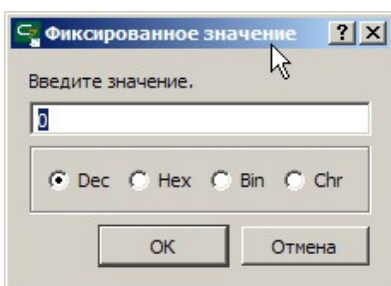
Для имитации внешних воздействий на диаграмму во время симуляции, в симуляторе предусмотрена возможность ручного принудительного изменения текущего значения сетей.

Принудительное изменение значения сети производится выбором пункта контекстного меню «Установить значение» доступного по нажатию правой кнопки мыши на любом сегменте сети или регистре чтения.

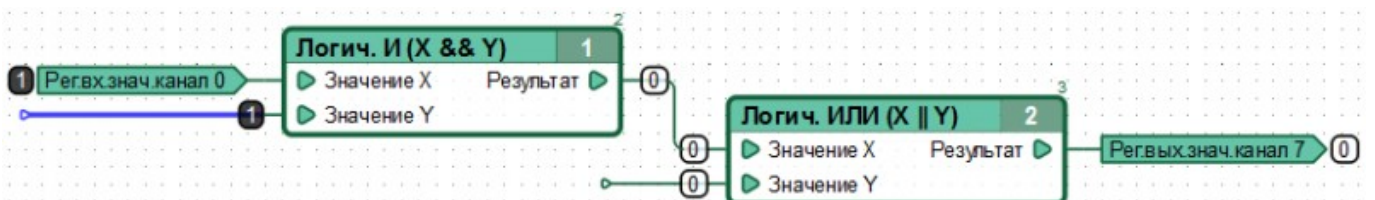




В открывшемся после выбора этого пункта диалоговом окне вводится устанавливаемое принудительно значение.



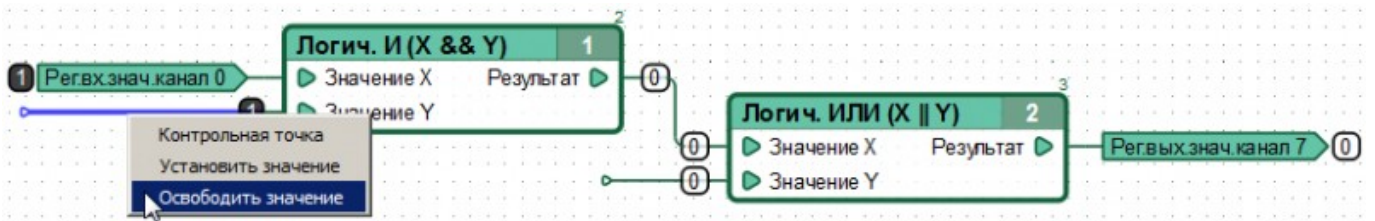
Значение, принудительно установленное для сети, отображается в виде надписи на черном фоне.



Установленное принудительно значение сети будет использоваться функциональными блоками, соединенными с данной сетью вместо её реального значения до тех пор, пока не будет освобождено вручную, либо пока не будет осуществлен выход их режима симуляции.

*Примечание: При выходе из симулятора значения, установленные принудительно в данной сессии симуляции, не сохраняются.*

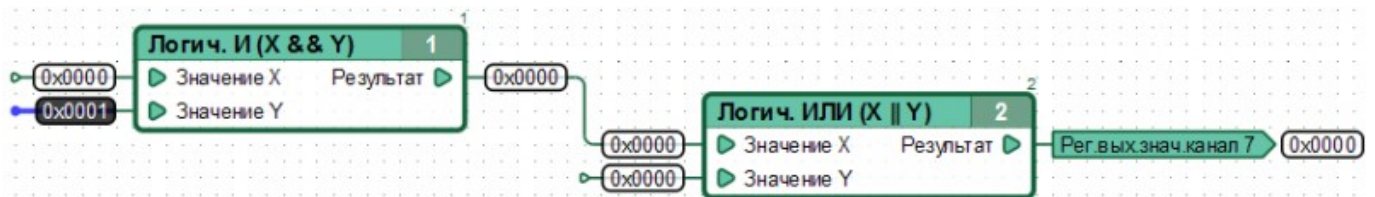
Возврат сети к получению значения определяемому работой диаграммы производится выбором пункта контекстного меню «Освободить значение» доступного по нажатию правой кнопки мыши на любом сегменте сети с принудительно установленным значением.



Быстро принудительно установить значение какой-либо сети равное единице, или быстро освободить значение сети можно также при помощи двойного нажатия левой клавиши мыши на данной сети, удерживая нажатой клавишу «Ctrl».

## 4.6 Изменение системы счисления отображаемых значений

По умолчанию, все значения отображаются в симуляторе в десятичном виде. Однако, систему счисления отображаемых значений можно в любой момент изменить, например на шестнадцатеричную, выбрав соответствующий пункт подменю «Симулятор» «Формат значений» главного меню программы.




# 5 Работа с контроллером

Интегрированная среда разработки SannyLab предоставляет пользователям возможность загрузки в контроллер системного программного обеспечения, необходимого для выполнения функциональной диаграммы, а так же самой функциональной диаграммы.

## 5.1 Подключение контроллера к персональному компьютеру

Отключите контроллер от внешнего питания и соедините его с ПК посредством стандартного кабеля *miniUSB*, зеленый светодиод контроллера при этом включится. При первом подключении контроллера, Windows сообщит об обнаружении нового стандартного устройства и автоматически установит драйвер для него.

Установление связи SannyLab с подключенным к ПК контроллером производится выбором пункта «Устройство» => «Подключить» главного меню программы или нажатием кнопки  панели инструментов.

Если связь с контроллером успешно установлена, зеленый светодиод контроллера начнет прерывисто мерцать, в строке статуса SannyLab отобразится информация о подключении и наличии или отсутствии в подключенном контроллере программного обеспечения, станут доступны пункты меню и кнопки панели инструментов, относящиеся к работе с контроллером.

## 5.2 Запись системного программного обеспечения в контроллер

Запись в подключенный контроллер системного программного обеспечения осуществляется посредством меню «Устройство» => «Системное ПО» => «Записать», при этом выбор файла системного программного обеспечения контроллера осуществляется пользователем во всплывающем диалоговом окне.


Запись системного ПО ПЛК CANNY 7 занимает около минуты. Информация о ходе процесса записи будет отображаться в диалоговом окне. На время записи, частота мерцания зеленого светодиода контроллера увеличится.

*Примечание: В случае наличия в контроллере ранее записанного в него системного ПО и функциональной диаграммы, перед записью нового системного ПО в контроллер, после запроса подтверждения у пользователя они будут*

стерты.

## 5.3 Запись в контроллер функциональной диаграммы

Запись функциональной диаграммы возможна только в контроллер уже содержащий системное ПО. Запись в контроллер текущей открытой в SannyLab функциональной диаграммы осуществляется посредством меню «Устройство» => «Диаграмма» =>


«Записать» или при нажатии кнопки  панели инструментов.

Информация о ходе процесса записи будет отображаться в диалоговом окне. На время записи, частота мерцания зеленого светодиода контроллера увеличится.

*Примечание: В случае наличия в контроллере ранее записанной диаграммы, после запроса подтверждения у пользователя, она будет стерта. Ранее записанное в контроллер системное ПО останется неизменным. Запись диаграммы в контроллер, обычно происходит быстрее, чем запись системного ПО.*


## 5.4 Завершение сеанса работы с контроллером

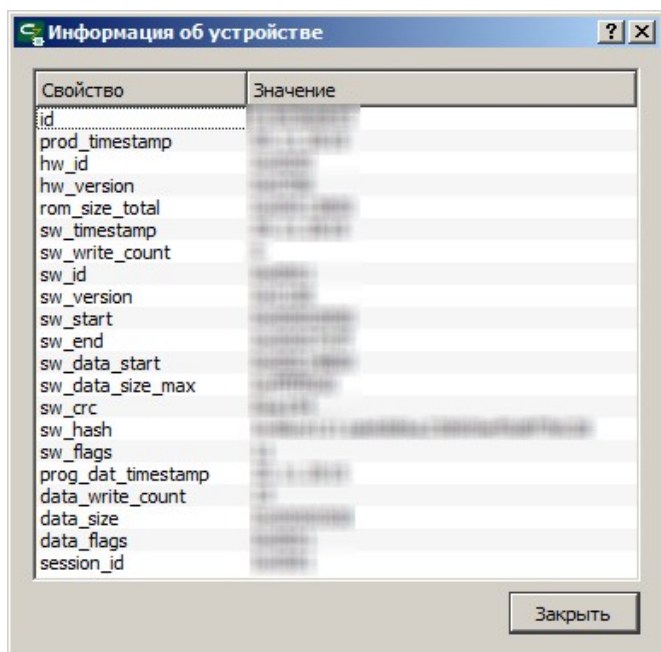
После завершения записи ПО, для безопасного завершения работы с контроллером, достаточно отключить от него шнур USB. Однако, для уверенности в том, что процесс записи контроллера был завершен, рекомендуется перед физическим отключением контроллера, выбрать пункт «Устройство» => «Отключить» главного меню программы,

или нажать кнопку  панели инструментов, при этом зеленый светодиод контроллера перестанет мерцать.


## 5.5 Получение технической информации о подключенном контроллере

Для получения специальной технической информации о подключенном контроллере необходимо выбрать пункт «Устройство» => «Информация» главного меню программы,

или нажать кнопку  панели инструментов. Информация об устройстве будет отображена в специальном окне.



## 5.6 Удаление функциональной диаграммы из памяти контроллера

Для удаления функциональной диаграммы из памяти контроллера необходимо выбрать пункт «Устройство» => «Диаграмма» => «Стереть» главного меню программы, или нажать кнопку  панели инструментов. После запроса подтверждения у пользователя, диаграмма будет стерта из памяти контроллера.

## 5.7 Удаление системного ПО из памяти контроллера

Для удаления системного ПО из памяти контроллера необходимо выбрать пункт «Устройство» => «Системное ПО» => «Стереть» главного меню программы. После запроса подтверждения у пользователя, память контроллера будет очищена.

*Примечание: При удалении системного ПО из памяти контроллера, пользовательская диаграмма так же будет стерта.*

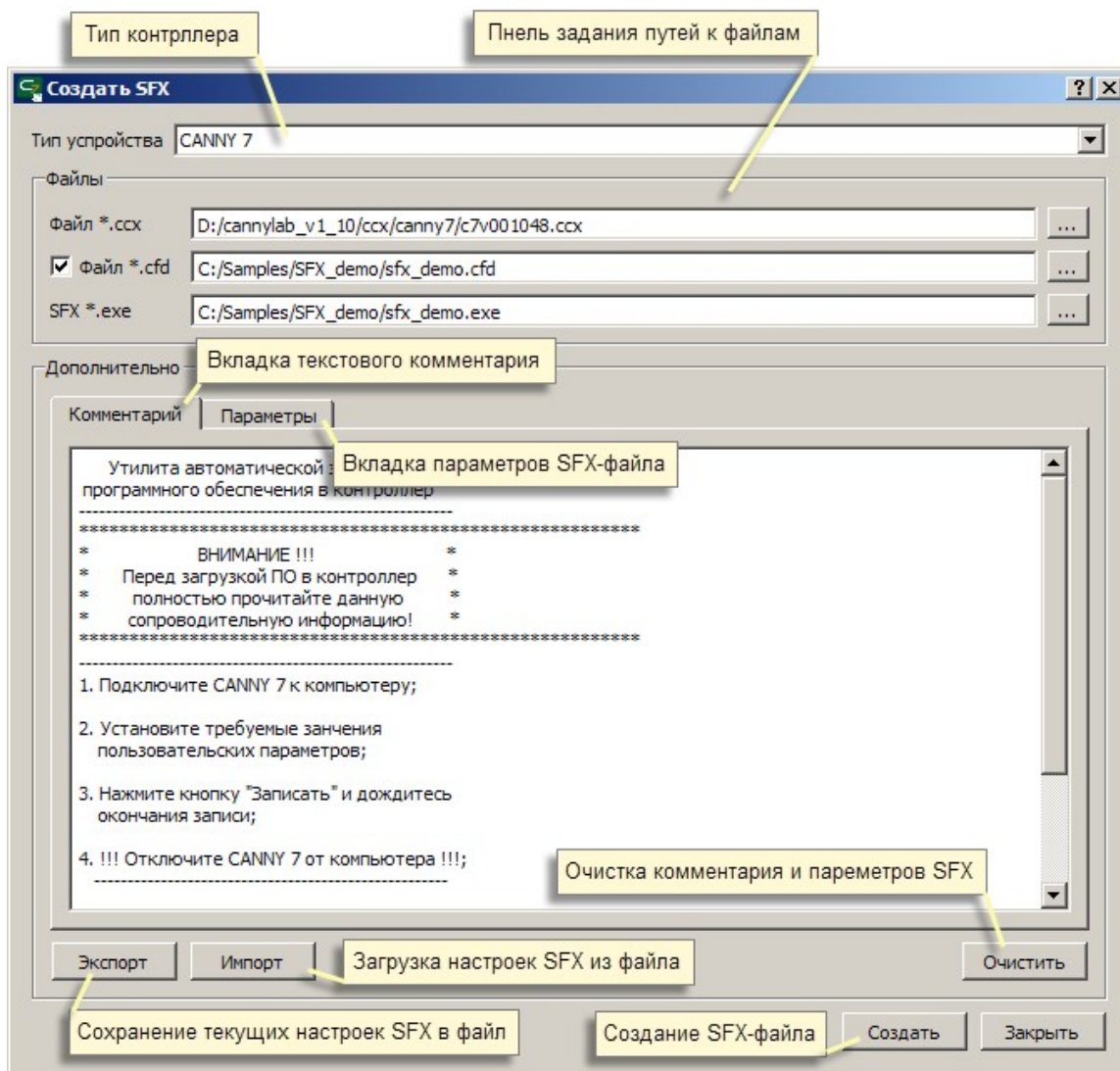
# **6 Исполняемые файлы автономной загрузки ПО в контроллер**

Интегрированная среда разработки CannyLab предоставляет пользователям возможность создания настраиваемого автономного исполняемого EXE-файла содержащего системное программное обеспечение контроллера и функциональную диаграмму.

Назначением такого файла является быстрое задание параметров работы контроллера в понятном конечному потребителю виде и простая загрузка программного обеспечения в контроллер используя компьютер, на котором не установлена среда CannyLab. В дальнейшем, такой файл будем называть: утилита автоматической загрузки ПО в контроллер или SFX-файл.

## **6.1 Создание SFX-файла**

Вызов диалога создания SFX-файла производится выбором пункта «Сервис» => «Создать SFX» главного меню программы

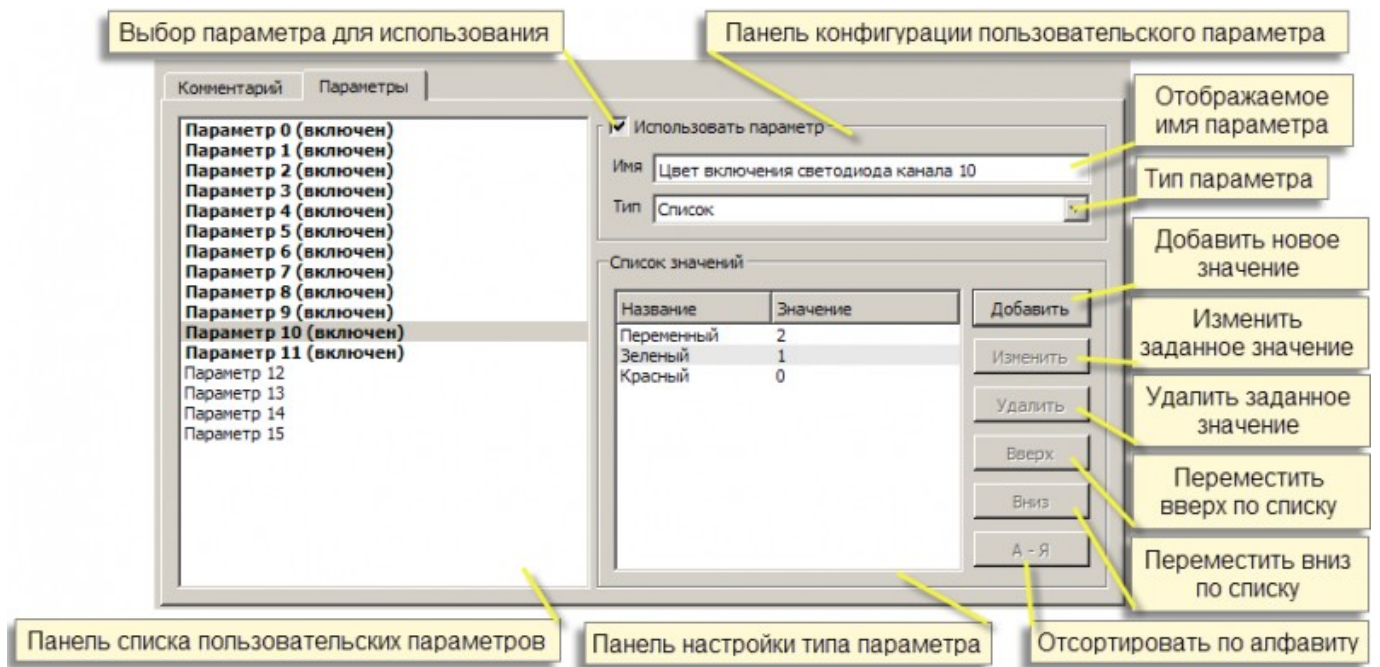


Прежде всего, в диалоговом окне необходимо указать тип контроллера для которого создается SFX-файл. В выпадающем списке нужно выбрать один из предлагаемых типов контроллера: CANNY 7, CANNY 5, CANNY 5 Nano.

В диалоговом окне необходимо указать имена и полные пути к файлам: исходного системного ПО «.ccx», исходного файла функциональной диаграммы «.cfd» и результирующего SFX «.exe» файла.

Текстовое поле вкладки «Комментарий» предназначено для ввода и редактирования текстового сообщения, которое увидит пользователь при запуске полученного SFX-файла. Это могут быть сведения о включенных в него файлах, их версиях и именах, краткие инструкции по пользованию утилитой автоматической загрузки ПО, информация о назначении пользовательских параметров, при их использовании, краткое описание функций, реализуемых программным обеспечением.

Вкладка «Параметры» предназначена задания имен и типов параметров, значения которых будет предложено выбрать пользователю при запуске утилиты автоматической загрузки ПО.



Максимальное число параметров, доступных для выбора пользователем — 16. При создании SFX-файла, каждому параметру задается отображаемое имя и назначается тип. Параметр может быть одного из двух типов — «Число» или «Список».

При запуске SFX-файла, задание параметра типа «Число» будет осуществляться пользователем путем ввода с клавиатуры числового значения. Задание параметра типа «Список» - выбором по названию соответствующего элемента из выпадающего списка.

Созданный таким образом файл, может быть передан на носителе, отправлен по электронной почте или опубликован в интернет и использован конечным потребителем для загрузки и задания параметров работы контроллера без использования какого-либо дополнительного ПО.

**ВНИМАНИЕ! Несмотря на свою абсолютную безвредность, создаваемые CannyLab SFX-файлы имеют необычную для Windows структуру, чем вызывают подозрение некоторого антивирусного ПО. Антивирусное ПО может выдать предупреждение или вовсе удалить файл при его создании или попытке обращения к нему. Если вы столкнулись с подобной ситуацией, то исключите SFX-файл из операций сканирования вашего антивируса.**

По окончании настройки параметров SFX-файла и нажатию кнопки «Создать», производится запись на диск результирующего исполняемого файла.

*Примечание: Файлы системного программного обеспечения (.csx) и файлы диаграмм (.cfd), предназначенные для определенного типа контроллера, не совместимы с другими типами контроллеров и не могут использоваться с ними. При попытке использования некорректных исходных данных,*

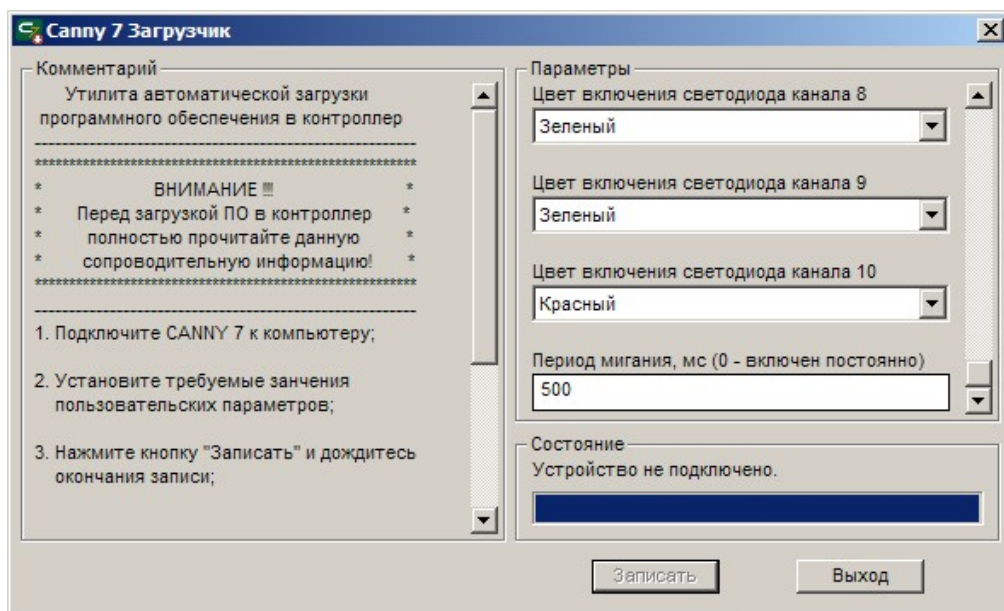


*SFX-файл не будет создан, при этом пользователю будет выдано предупреждение об ошибке.*

Процедура экспорта производит сохранение содержимого комментария и описание параметров SFX в текстовый файл, с возможностью последующего импорта при создании SFX-файлов в дальнейшем, для исключения необходимости повторного ввода данных.

## 6.2 Использование утилиты автоматической загрузки ПО в контроллер

При запуске созданного исполняемого файла утилиты автоматической загрузки ПО в контроллер отображается диалоговое окно, в левой части которого располагается область комментария, а справа — панель задания значений параметров и панель состояния.



В области комментария отражена информация, указанная в соответствующем поле при создании SFX-файла.

Область параметров заполнена элементами, позволяющими задать значения способом, предусмотренным при создании SFX-файла — выбрав значения из списка или введя их с клавиатуры. Если при создании SFX-файла описание параметров не было задано, то данная область будет пуста.

Выбранные пользователем значения параметров, после записи контроллера, будут доступны функциональной диаграмме в регистрах «Регистр параметра пользовательской конфигурации №XX».

*Примечание:*

*При большом объеме комментария и большом числе используемых параметров, они могут не полностью помещаться в соответствующих панелях диалогового окна. Для доступа ко всему содержимому панелей, предусмотрены полосы вертикальной прокрутки в их правых частях.*

Для загрузки выбранных значений параметров и ПО содержащегося в SFX-файле в контроллер, отключите контроллер от внешнего питания и соедините его с ПК посредством стандартного кабеля miniUSB, зеленый светодиод контроллера при этом включится. При первом подключении контроллера, Windows сообщит об обнаружении нового стандартного устройства и автоматически установит драйвер для него.

До момента установления связи с контроллером, в панели состояния будет отображена надпись «Устройство не подключено», кнопка «Запись» будет недоступна. После успешного установления связи, зеленый светодиод контроллера начнет прерывисто мерцать, в панели состояния отобразится надпись «Устройство готово к записи» и кнопка «Запись» станет активна.

*Примечание: Во избежание сбоев в работе утилиты, не запускайте несколько SFX-файлов и не подключайте к ПК более одного контроллера одновременно.*

Для начала загрузки ПО в контроллер нажмите кнопку «Записать». В строке состояния будет отображаться ход процесса записи.

В случае успешного окончания записи ПО в контроллер в панели состояния появится сообщение «Запись закончена».

Если процесс записи ПО в контроллер был прерван по команде пользователя или во время записи произошла ошибка, в панели состояния появится сообщение «Возникла ошибка записи». В данном случае, процесс записи должен быть повторен.

# 7 Язык CFD

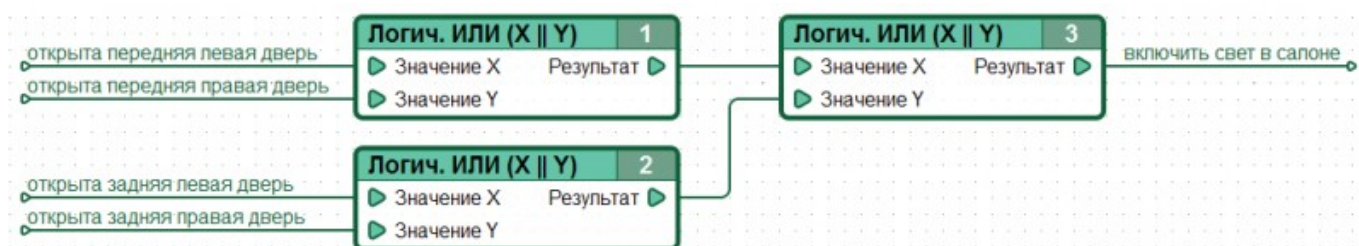
В интегрированной среде разработки CannyLab, для программирования ПЛК CANNY7 используется язык программирования Canny Function Diagram (CFD). При его разработке, была поставлена цель создать инструмент, максимально отвечающий возможностям контроллера и в то же время достаточно близкий к отраслевому стандарту. Язык диаграмм функциональных блоков Function Block Diagram (FBD), взятый за основу при разработке CFD, является, пожалуй, одним из самых сбалансированных среди пяти языков программирования, одобренных международной электротехнической комиссией для создания пользовательских программ ПЛК в рамках стандарта МЭК 61131-3.

Являясь графическим языком программирования, CFD отличается высокой наглядностью, сопоставимой с наглядностью принципиальных схем, но находится на более высоком уровне абстракции, позволяющем скрыть несущественные детали реализации.

## 7.1 Функциональная диаграмма

Программа на языке CFD, представляющая собой чертеж (схему), на котором изображены функциональные блоки, связывающие их друг с другом соединительные линии (сети) и вспомогательные элементы, называется функциональной диаграммой.

Процесс выполнения программы на языке CFD можно описать как последовательное, обычно слева на право, вычисление каждого функционального блока диаграммы, с передачей результата к следующему связанному с ним функциональному блоку. Такую организацию вычислений будем называть потоком выполнения.

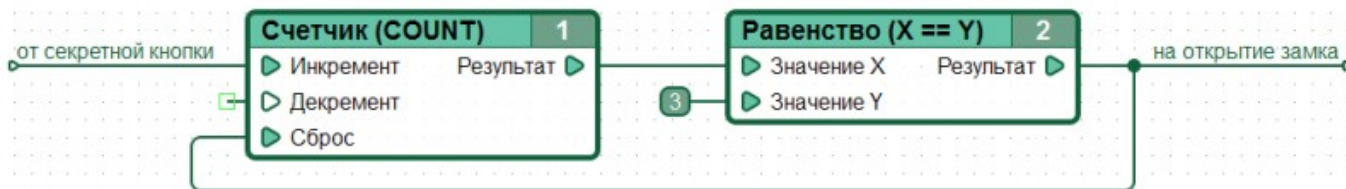


Представление программы подобным образом, позволяет легко проследить процесс ее выполнения, просто рассматривая изображение, переходя взглядом от блока к блоку по соединительным линиям.

В языке CFD, взаимное расположение блоков на ациклической диаграмме не влияет на порядок их выполнения. Порядок выполнения функциональных блоков такой диаграммы вычисляется автоматически по следующему правилу: каждый функциональный блок

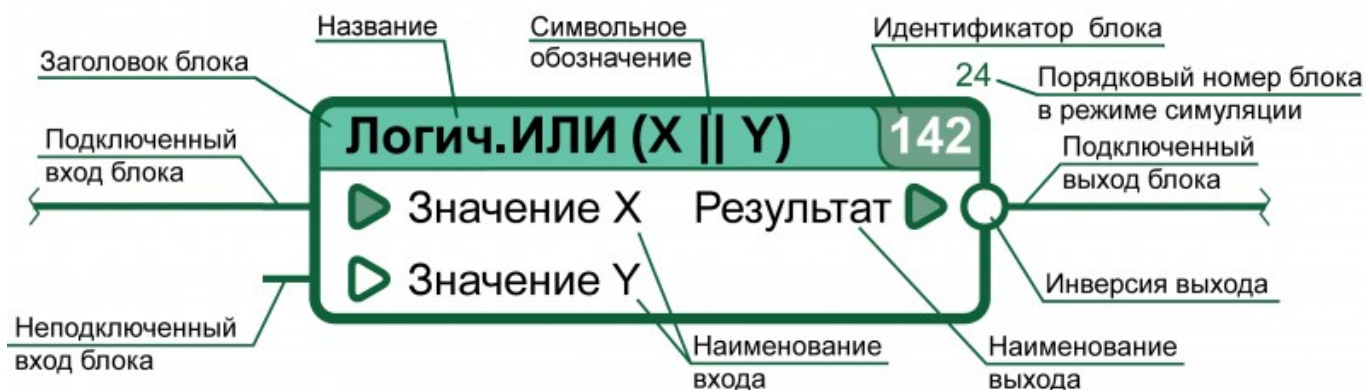
выполняется тогда, когда вычислены все его входы.

При определении порядка выполнения циклической, то есть имеющей обратные связи диаграммы, функциональные блоки участвующие в цикле обратной связи выполняются в таком порядке, чтобы последним исполнился функциональный блок находящийся правее всех остальных участвующих в цикле блоков.



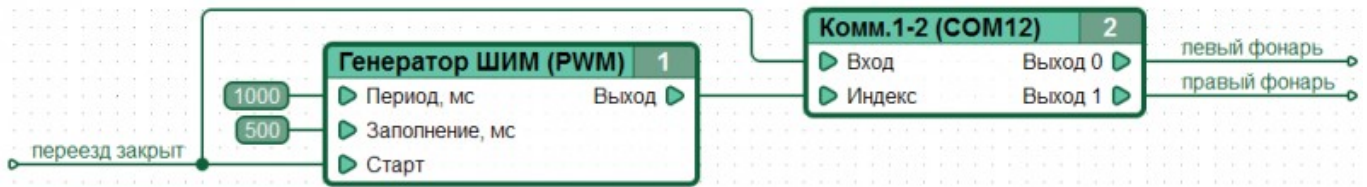
## 7.2 Функциональный блок

Функциональный блок является графическим элементом функциональной диаграммы, обозначающим функцию, алгоритм или математическое выражение, производящее вычисление своих выходных данных в зависимости от полученных извне входных данных. На изображении функционального блока, его входы располагают слева, а выходы справа.



Стандартная библиотека функциональных блоков языка CFD включает в себя как простейшие блоки, выполняющие элементарные логические преобразования над своими входными данными — элементы И, ИЛИ, НЕ, так и более сложные функции: триггеры, генераторы, коммутаторы.

Функциональный блок, в зависимости от его типа, может иметь несколько входов и несколько выходов, использовать для вычислений свою собственную память, а так же обращаться к ресурсам операционной системы контроллера, например к таймерам и использовать полученные данные в расчетах своих выходных значений.



В среде SannyLab, источниками входных данных для функционального блока могут быть функциональные блоки, с чьими выходами он связан соединительными линиями (сетями), введенные пользователем или выбранные из справочника числовые константы либо данные регистров служебной памяти контроллера.

Любой вход или выход функционального блока может быть инвертирован. В случае инверсии входа, в качестве входных данных в функциональный блок будет передано значение являющееся результатом операции «Логическое НЕ» над реальными входными данными. В случае инверсии выхода, перед передачей результата вычисления функционального блока «наружу», к нему будет применена операция «Логическое НЕ».

## 7.3 Сеть

Линия, соединяющая на функциональной диаграмме источник данных с одним или несколькими получателями данных, называется сетью. В языке CFD, источником данных обычно выступает выход какого-либо функционального блока, а получателем вход другого функционального блока. В языке CFD любая сеть может иметь только один источник данных.

В терминах языков программирования, сеть является *переменной*, местом хранения данных. Функциональный блок по окончании вычислений, записывает в сеть своё выходное значение, а выполняющиеся следом за ним функциональные блоки, к чьим входам данная сеть подключена, считывают из этой сети хранящее в ней значение и используют его в своих вычислениях в качестве входных значений. Наряду с использованием анонимных сетей, в SannyLab, сетям можно назначать имена, которые будут отображаться на диаграмме. Логически единую сеть, для которой назначено имя, можно для удобства компоновки, разделять на несколько визуально не связанных между собой сегментов. Например, два сегмента «Давление в шинах, кПа» на приведенном ниже рисунке представляют одну и ту же сеть, и имеют одинаковое значение в каждый момент времени, хотя визуально не связаны.



## 7.4 Данные

В языке CFD, все использующиеся данные являются 16-битными целыми неотрицательными числами в диапазоне от 0 до 65535. Результатом арифметических операций будет числовое значение из указанного диапазона, а результаты логических операций представляются значениями «1» (истина) и «0» (ложь).

При выполнении функциональной диаграммы, там где это необходимо, преобразование данных от логического типа к числовым данным и обратно выполняется автоматически по следующим правилам:

- При преобразовании числа к логическому значению, ноль преобразуется в значение «ложь», любое число отличное от нуля – в значение «истина».
- В арифметических операциях логические значения «ложь» и «истина» представляются как числа 0 и 1 соответственно.
- Инверсия на входе или выходе любого блока, автоматически преобразует его данные к логическому типу.



Результаты арифметических операций берутся по модулю 65535. Это означает, что если результат операции превысил 65535, то из него автоматически вычитается 65536. А если результат оказался меньше нуля, то к нему прибавляется 65535 и еще единица.

Например:  $65535 + 1 = 0$        $1 - 2 = 65535$   
 $65535 + 2 = 1$        $1 - 3 = 65534$   
 $65500 + 122 = 86$      $117 - 259 = 65294$

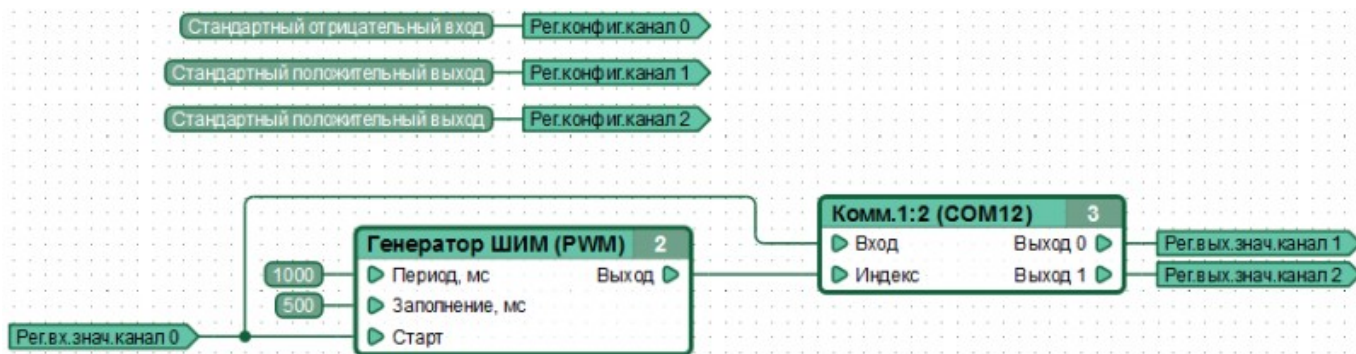
## 7.5 Управление ресурсами контроллера

Для того, чтобы разработанная пользователем функциональная диаграмма могла использовать данные физических входов контроллера и изменять состояние его физических выходов, получать и передавать данные по интерфейсам передачи данных, или хотя бы просто изменять состояние встроенного светодиода контроллера, диаграмму и ресурсы контроллера необходимо связать между собой.

В контроллере CANNY7 доступ к аппаратным ресурсам из пользовательского приложения, реализован в форме чтения и записи данных по определенным адресам (регистрам) отображающим состояние этих ресурсов.

Ресурсы контроллера требуют определенного порядка доступа. Например, в самом простом случае, чтобы включить встроенный светодиод контроллера, достаточно записать в определенный регистр значение «1», а чтобы выключить «0». Но уже для того, чтобы использовать универсальный внешний канал контроллера для управления реле, его необходимо предварительно его инициализировать, записав значение требуемого режима его работы в регистр его конфигурации. Лишь после этого появляется возможность, устанавливая «1», или «0» в регистр выходного значения канала, управлять электрическим потенциалом данного канала контроллера.

Так будет выглядеть приведенная ранее диаграмма управления светофором железнодорожного переезда, после привязки к ресурсам контроллера. Такая диаграмма уже полностью пригодна для записи в ПЛК и управления реальным оборудованием.



Список регистров всех доступных пользователю ресурсов контроллера находится в справочнике регистров CannyLab, который доступен пользователю через контекстное меню регистра или при назначении входов и выходов функциональных блоков через пункт «Задать регистр».

Конфигурация определенных ресурсов контроллера требует использования специальных констант, все они поименованы и собраны в списке констант CannyLab, который доступен пользователю при назначении входов типа «Константа» функциональных блоков.

Подробное описание ресурсов CANNY 7 и порядка доступа к ним, рассматривается в справочном руководстве по контроллеру.

# 8 Справочник по встроенным функциональным блокам

## 8.1 Константы и регистры

- Установка значения
- Константа
- Копирование
- Регистр чтения
- Регистр записи

### 8.1.1 Логические операции

- Логическое НЕ
- Логическое И
- Логическое ИЛИ
- Логическое исключающее ИЛИ

### 8.1.2 Побитовые логические операции

- Побитовое И
- Побитовое ИЛИ
- Побитовое исключающее ИЛИ
- Побитовое НЕ
- Сдвиг влево
- Сдвиг вправо
- Циклический сдвиг влево
- Циклический сдвиг вправо

### 8.1.3 Операции сравнения

- Равенство
- Больше
- Не больше
- Меньше
- Не меньше



## **8.1.4 Арифметические операции**

- Сложение
- Вычитание
- Умножение
- Деление

## **8.1.5 Триггеры**

- Триггер SR
- Триггер RS
- Триггер D
- Триггер T
- Триггер JK
- Буфер

## **8.1.6 Таймеры и генераторы**

- Задержка включения
- Задержка выключения
- Генератор импульсов
- Генератор псевдослучайных чисел

## **8.1.7 Счетчики и детекторы**

- Счетчик
- Детектор переднего фронта
- Детектор заднего фронта

## **8.1.8 Конвертеры**

- Конвертер 8-в-1
- Конвертер 16-в-1
- Конвертер 1-в-8
- Конвертер 1-в-16
- Конвертер строки символов в число
- Конвертер числа в строку символов
- Отображение

## **8.1.9 Коммутаторы**

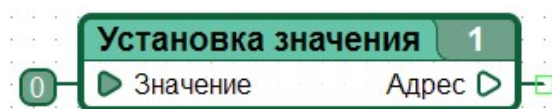
- Коммутатор 2-в-1
- Коммутатор 8-в-1
- Коммутатор 16-в-1
- Коммутатор 1-в-2
- Коммутатор 1-в-8
- Коммутатор 1-в-16

## **8.1.10 Элементы оформления**

- Надпись

# 9 Установка значения

## Внешний вид



## Символьное обозначение

:=

## Группа

Константы и регистры

## Входы

Значение: числовой

## Выходы

Адрес: числовой

## Версия CANNY Lab

с 0.6b до 0.16b



Устанавливает на выходе значение, полученное на входе. Чаще всего, блоки данного типа используются для однократной конфигурации аппаратных ресурсов контроллера или для централизованного введения в функциональную диаграмму параметров-констант.

## 9.1 Особенности

*«Установка значения» – специальный функциональный блок, имеющий два существенных отличия от других функциональных блоков: входом данного функционального блока может быть только константа и поток выполнения заходит в блок такого типа после каждого запуска контроллера лишь единожды и гарантировано раньше выполнения блока любого другого типа.*

С выходом CANNY Lab версии 1.0 необходимость в использовании данного типа блоков исчезла и блок был упразднен.

## 9.2 Примеры

Пример 1



# 10 Константа

## Внешний вид



## Символьное обозначение

:=

## Группа

Константы и регистры

## Версия CANNY Lab

с 1.0



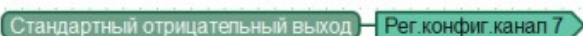
Специальный элемент диаграммы, позволяет задать исходные и неизменные в течении всего времени работы диаграммы значения сетей или регистров. Чаще всего, данный элемент используется для конфигурации аппаратных ресурсов контроллера или введения в функциональную диаграмму параметров. При редактировании диаграммы, для использования в арифметических и логических операциях константы могут быть заданы вручную, в различных системах счисления, при помощи диалога установки значения константы либо, при использовании констант для установки значений в некоторых регистрах контроллера, путем выбора из справочника именованных констант.

## 10.1 Особенности

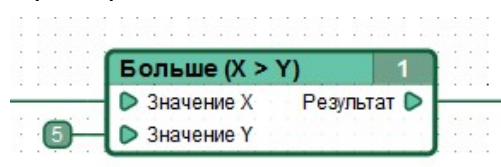
До выхода CANNY Lab версии 1.0, передать константу в регистр контроллера можно было лишь с использованием функционального блока "Установка значения", фигурирующего на многих ранних демонстрационных видео CANNY Lab. С выходом CANNY Lab версии 1.0 необходимость в использовании данного типа блоков исчезла и блок был упразднен.

## 10.2 Примеры

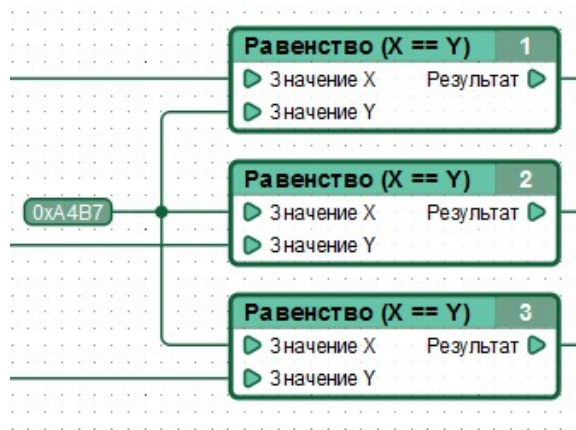
Пример 1



Пример 2

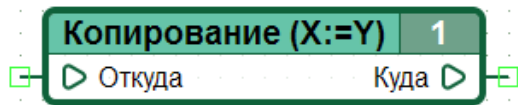


### Пример 3



# 11 Копирование

## Внешний вид



## Символьное обозначение

$X := Y$

## Группа

Константы и регистры

## Входы

Откуда: числовой

## Выходы

Куда: числовой

## Версия CANNY Lab

с 0.6b до 1.15



Копирует значение входа на выход.

## 11.1 Особенности

До выхода CANNY Lab версии 1.16 блок "Копирование", как правило, использовался для записи значения из одного источника в несколько регистров и фигурирует на многих ранних демонстрационных видео CANNY Lab. С выходом CANNY Lab версии 1.16 необходимость в использовании данного типа блоков исчезла и блок был упразднен.

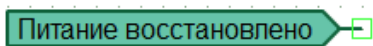
## 11.2 Примеры

Пример 1



# 12 Регистр чтения

## Внешний вид



## Символьное обозначение

R-

## Группа

Константы и регистры

## Выходы

-- числовой

## Версия CANNY Lab

с 1.16



Элемент функциональной диаграммы, позволяющий получить информацию о состоянии контроллера.

Все доступные пользователю из функциональной диаграммы ресурсы контроллера: системные регистры, подсистема ввода-вывода и дополнительные драйверы включенные в состав системного ПО, отображаются на защищенное адресное пространство внутренней памяти контроллера.

Посредством элемента «Регистр чтения» пользователь имеет возможность извлечь и использовать данные о состоянии контроллера, и информацию полученную контроллером из внешнего мира. Например информацию об электрическом потенциале на каком-либо контакте контроллера, или содержимое пакета данных принятого контроллером по CAN.

Подробнее см. документацию к конкретной модели контроллера.

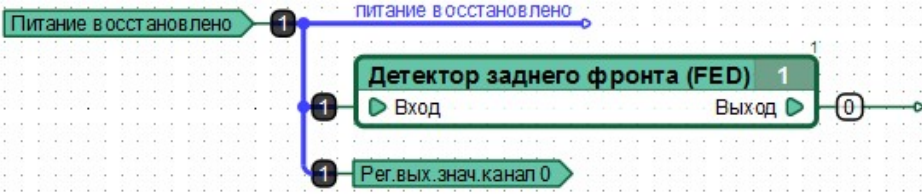
## 12.1 Особенности

Элемент «Регистр чтения» может быть соединен с сетью, непосредственно со входом функционального блока или регистром записи тем самым задавая в них значение, синхронно с изменением значения в соответствующем регистре контроллера.



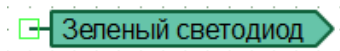
# 12.2 Примеры

Пример использования



# 13 Регистр записи

## Внешний вид



## Символьное обозначение

-R

## Группа

Константы и регистры

## Входы

-- числовой

## Версия CANNY Lab

с 1.16



Элемент функциональной диаграммы, позволяющий изменить внутреннее состояние контроллера или передать информацию во внешний мир.

Посредством элемента «Регистр записи», осуществляется управление ресурсами контроллера из функциональной диаграммы, с возможностью воздействия на объекты внешнего мира. Например, переключить внешнее реле, изменив электрический потенциал на одном из контактов контроллера, к которому подключена его обмотка; включить контрольный светодиод; задать режим работы CAN; отправить пакет данных.

Порядок использования большинства ресурсов контроллера включает в себя задание пользователем необходимых параметров их работы, например полярности выходных каналов, полярности и чувствительности входных каналов, скорости обмена данными по CAN и т. д.

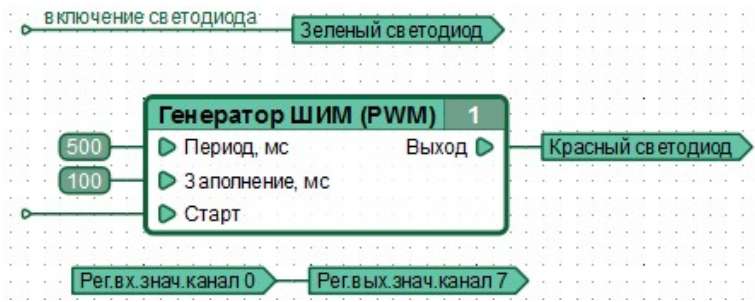
Подробнее см. документацию к конкретной модели контроллера

## 13.1 Особенности

К элементу «Регистр записи» может быть присоединена сеть, непосредственно выход функционального блока или регистр чтения тем самым задавая в соответствующем регистре контроллера значение, синхронно с изменением значения на соответствующем участке диаграммы.

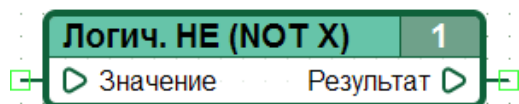
## 13.2 Примеры

Пример использования



# 14 Логическое НЕ

## Внешний вид



## Символьное обозначение

NOT X

## Группа

Логические операции

## Входы

Значение: логический

## Выходы

Результат: логический

## Версия CANNY Lab

с 0.6b



Логическое отрицание. Устанавливает на выходе значение, логически обратное входному.

Иными словами, на выходе будет:

- «1» тогда и только тогда, когда на входе «0»,
- «0» тогда и только тогда, когда на входе «1».

Таким образом, таблица истинности операции имеет следующий вид:

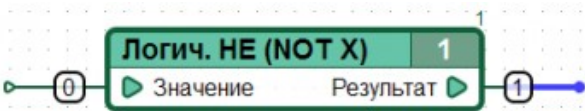
Значение	Результат
0	1
1	0

## 14.1 Особенности

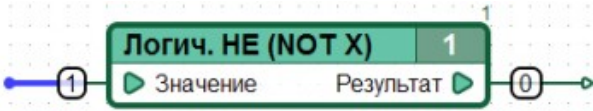
Использование блока 'Логическое НЕ' эквивалентно применению логической инверсии к значению входа или выхода любого функционального блока.

# 14.2 Примеры

Пример 1

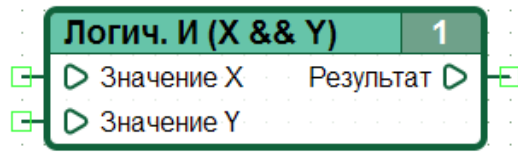


Пример 2



# 15 Логическое И

## Внешний вид



## Символьное обозначение

X & Y

## Группа

Логические операции

## Входы

Значение X: логический

Значение Y: логический

## Выходы

Результат: логический

## Версия CANNY Lab

с 0.6b



Логическое умножение (конъюнкция). Устанавливает на выходе результат логического умножения значений входов.

Иными словами, на выходе будет:

- «1» тогда и только тогда, когда на всех входах «1»,
- «0» тогда и только тогда, когда хотя бы на одном входе «0».

Таким образом, таблица истинности операции имеет следующий вид:

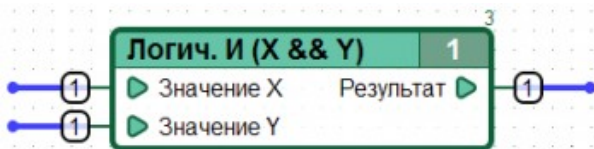
Значение X	Значение Y	Результат
0	0	0
0	1	0
1	0	0
1	1	1

## 15.1 Особенности

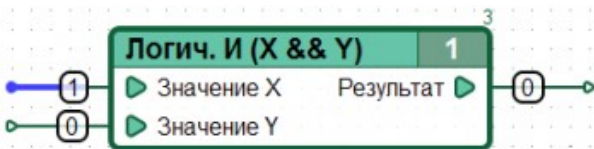
Используя блоки Конвертер 8-в-1 и Конвертер 16-в-1 можно построить схему 'Логического И' для нескольких значений из меньшего количества функциональных блоков.

## 15.2 Примеры

Пример 1

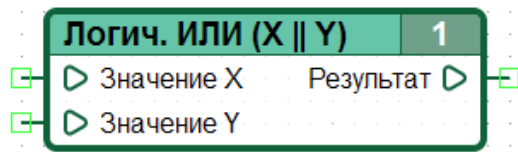


Пример 2



# 16 Логическое ИЛИ

## Внешний вид



## Символьное обозначение

$X \parallel Y$

## Группа

Логические операции

## Входы

Значение X: логический

Значение Y: логический

## Выходы

Результат: логический

## Версия CANNY Lab

с 0.6b



Логическое сложение (дизъюнкция). Устанавливает на выходе результат логического сложения значений входов.

Иными словами, на выходе будет:

- «1» тогда и только тогда, когда хотя бы на одном входе «1»,
- «0» тогда и только тогда, когда на всех входах «0».

Таким образом, таблица истинности операции имеет следующий вид:

Значение X	Значение Y	Результат
0	0	0
0	1	1
1	0	1
1	1	1

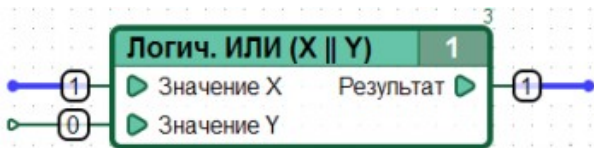


## 16.1 Особенности

Используя блоки Конвертер 8-в-1 и Конвертер 16-в-1 можно построить схему 'Логического ИЛИ' для нескольких значений из меньшего количества функциональных блоков.

## 16.2 Примеры

Пример 1

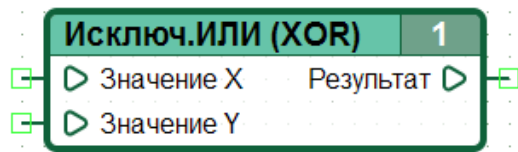


Пример 2



# 17 Логическое исключающее ИЛИ

## Внешний вид



## Символьное обозначение

XOR

## Группа

Логические операции

## Входы

Значение X: логический

Значение Y: логический

## Выходы

Результат: логический

## Версия CANNY Lab

с 0.6b



Сложение по модулю 2. Устанавливает на выходе результат операции XOR над значениями входов.

Иными словами, на выходе будет:

- «1» тогда и только тогда, когда лишь на одном из входов «1»,
- «0» тогда и только тогда, когда значения входов эквивалентны друг другу.

Таким образом, таблица истинности операции имеет следующий вид:

Значение X	Значение Y	Результат
0	0	0
0	1	1
1	0	1
1	1	0

# 17.1 Примеры

Пример 1

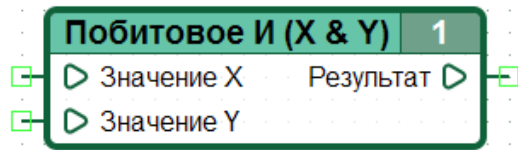


Пример 2



# 18 Побитовое И

## Внешний вид



## Символьное обозначение

X & Y

## Группа

Побитовые логические операции

## Входы

Значение X:            числовой

Значение Y:            числовой

## Выходы

Результат:            числовой

## Версия CANNY Lab

с 0.6b



Побитовое логическое умножение (конъюнкция). Устанавливает на выходе результат логического умножения, примененного к каждой паре битов, которые стоят на одинаковых позициях в двоичных представлениях значений входов.

Иными словами, результирующий двоичный разряд будет равен:

- «1», если оба соответствующих бита значений входов равны «1»;
- «0», если хотя бы один бит из пары равен «0».

Например:

Значение X = 27 = 0x001B = 0b0000000000011011

Значение Y = 58 = 0x003A = 0b0000000000111010

Результат = 26 = 0x001A = 0b000000000011010

## 18.1 Особенности

Операция 'Побитовое И' используется для маскирования — выбора отдельных битов или полей из нескольких битов из двоичного числа.

## 18.2 Примеры

Десятичное представление

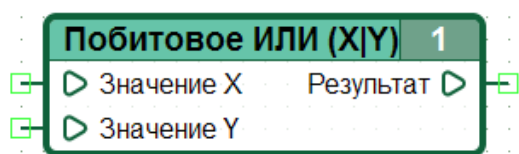


Двоичное представление



# 19 Побитовое ИЛИ

## Внешний вид



## Символьное обозначение

$X | Y$

## Группа

Побитовые логические операции

## Входы

Значение X: числовой

Значение Y: числовой

## Выходы

Результат: числовой

## Версия CANNY Lab

с 0.6b



Побитовое логическое сложение (дизъюнкция). Устанавливает на выходе результат логического сложения, примененного к каждой паре битов, которые стоят на одинаковых позициях в двоичных представлениях значений входов.

Иными словами, результирующий двоичный разряд будет равен:

- «1», если хотя бы один бит из пары соответствующих бит значений входов равен «1»;
- «0», если оба бита из пары равны «0».

Например:

Значение X = 27 = 0x001B = 0b0000000000011011

Значение Y = 58 = 0x003A = 0b0000000000111010

Результат = 59 = 0x003B = 0b0000000000111011

## 19.1 Особенности

Операция 'Побитовое ИЛИ' используется для установки определенного бита или полей из нескольких битов в двоичном числе.

## 19.2 Примеры

Десятичное представление

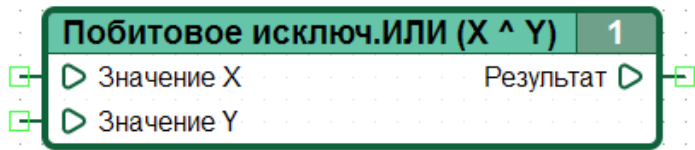


Двоичное представление



# 20 Побитовое исключающее ИЛИ

## Внешний вид



## Символьное обозначение

$$X \oplus Y$$

## Группа

Побитовые логические операции

## Входы

Значение X: числовой

Значение Y: числовой

## Выходы

Результат: числовой

## Версия CANNY Lab

с 0.6b



Побитовое сложение по модулю 2. Устанавливает на выходе результат операции XOR, примененной к каждой паре битов, которые стоят на одинаковых позициях в двоичных представлениях значений входов.

Иными словами, результирующий двоичный разряд будет равен:

- «1», если только один бит из пары соответствующих бит значений входов равен «1»;
- «0», если оба бита из пары равны «0», или оба бита из пары равны «1».

Например:

Значение X = 27 = 0x001B = 0b0000000000011011

Значение Y = 58 = 0x003A = 0b0000000000111010

Результат = 33 = 0x0021 = 0b0000000000100001



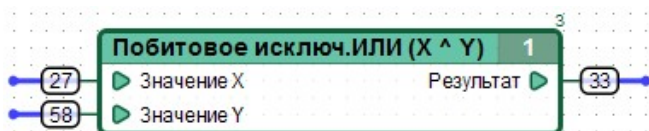
## 20.1 Особенности

Результат 'Побитового исключающего ИЛИ' числа самого с собой всегда равен нулю.

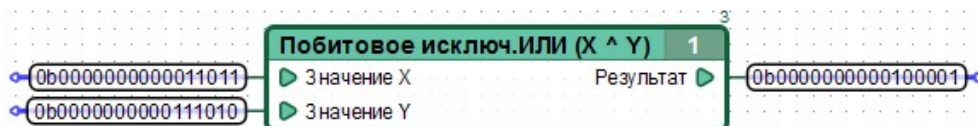
Результат выполнения операции 'Побитовое исключающее ИЛИ' любого X с одним и тем же значением Y дважды, всегда равен исходному значению X.

## 20.2 Примеры

Десятичное представление

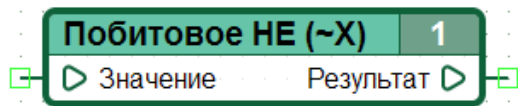


Двоичное представление



# 21 Побитовое НЕ

## Внешний вид



## Символьное обозначение

$\sim X$

## Группа

Побитовые логические операции

## Входы

Значение                      числовой

## Выходы

Результат:                      числовой

## Версия CANNY Lab

с 0.6b



Побитовая инверсия. Устанавливает на выходе результат инверсии, замены на противоположное значение, примененной к каждому биту в двоичном представлении значения входа.

Иными словами, результирующий двоичный разряд будет равен:

- «1», если бит значения входа равен «0»;
- «0», если бит значения входа равен «1».

Например:

Значение = 27        = 0x001B = 0b0000000000011011

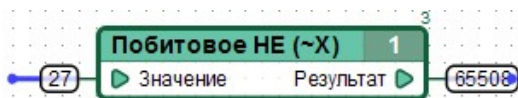
Результат = 65508 = 0xFFE4 = 0b1111111111100100

## 21.1 Особенности

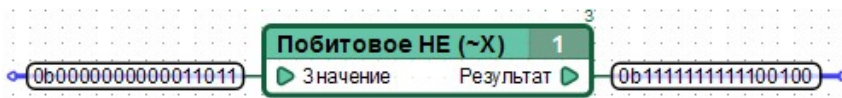
Результат выполнения операции 'Побитовое НЕ' над любым значением дважды, всегда равен исходному значению.

## 21.2 Примеры

Десятичное представление

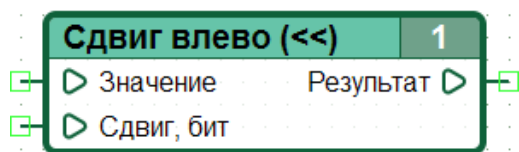


Двоичное представление



## 22 Сдвиг влево

### Внешний вид



### Символьное обозначение

<<

### Группа

Побитовые логические операции

### Входы

Значение: числовой

Сдвиг, бит: числовой

### Выходы

Результат: числовой

### Версия CANNY Lab

с 0.6b



Логический сдвиг влево. Устанавливает на выходе результат операции логического сдвига влево двоичного представления значения входа «Значение», на число бит, заданное значением входа «Сдвиг, бит».

Иными словами, результат эквивалентен результату операции копирования каждого бита в двоичном представлении входа «Значение» в позицию слева от него, произведенной число раз, заданное значением входа «Сдвиг, бит». При этом младший (самый правый) бит в двоичном представлении результата каждый раз получает значение «0».

Например:

Значение = 22 = 0x0016 = 0b0000000000010110

Сдвиг, бит = 2

Результат = 88 = 0x0058 = 0b0000000001011000

## 22.1 Особенности

Результат логического сдвига значения влево:

на один бит — эквивалентен результату его арифметического умножения на 2

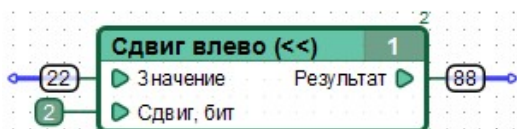
на два бита — эквивалентен результату его арифметического умножения на 4

на три бита — эквивалентен результату его арифметического умножения на 8

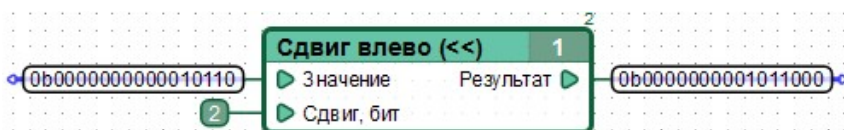
и т.д.

## 22.2 Примеры

Десятичное представление



Двоичное представление



## 23 Сдвиг вправо

### Внешний вид



### Символьное обозначение

>>

### Группа

Побитовые логические операции

### Входы

Значение: числовой

Сдвиг, бит: числовой

### Выходы

Результат: числовой

### Версия CANNY Lab

с 0.6b



Логический сдвиг вправо. Устанавливает на выходе результат операции логического сдвига вправо двоичного представления значения входа «Значение», на число бит, заданное значением входа «Сдвиг, бит».

Иными словами, результат эквивалентен результату операции копирования каждого бита в двоичном представлении входа «Значение» в позицию справа от него, произведенной число раз, заданное значением входа «Сдвиг, бит». При этом старший (самый левый) бит в двоичном представлении результата каждый раз получает значение «0».

Например:

Значение = 22 = 0x0016 = 0b0000000000010110

Сдвиг, бит = 2

Результат = 5 = 0x0005 = 0b0000000000000101

## 23.1 Особенности

Результат логического сдвига значения вправо:

на один бит — эквивалентен результату его целочисленного деления на 2

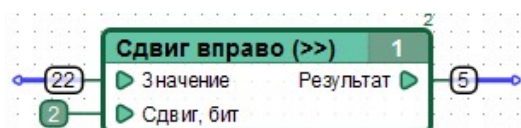
на два бита — эквивалентен результату его целочисленного деления на 4

на три бита — эквивалентен результату его целочисленного деления на 8

и т.д.

## 23.2 Примеры

Десятичное представление

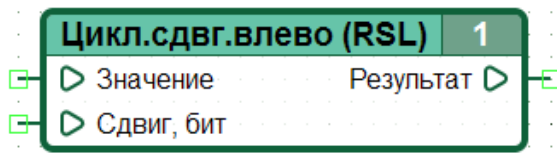


Двоичное представление



## 24 Циклический сдвиг влево

### Внешний вид



### Символьное обозначение

RSL

### Группа

Побитовые логические операции

### Входы

Значение: числовой

Сдвиг, бит: числовой

### Выходы

Результат: числовой

### Версия CANNY Lab

с 1.8



Циклический сдвиг влево. Устанавливает на выходе результат операции логического сдвига влево двоичного представления значения входа «Значение», на число бит, заданное значением входа «Сдвиг, бит», при этом уходящий бит появляется на месте появившегося свободного на другом конце числа.

Иными словами, результат эквивалентен результату операции копирования каждого бита в двоичном представлении входа «Значение» в позицию слева от него, произведенной число раз, заданное значением входа «Сдвиг, бит». При этом младший (самый правый) бит в двоичном представлении результата каждый раз получает значение, равное уходящему (самому старшему, выдвигаемому влево) биту входного значения.

Например:

Значение = 51000 = 0xC738 = 0b1100011100111000

Сдвиг, бит = 2

Результат = 7395 = 0x1CE3 = 0b0001110011100011



## 24.1 Особенности

Операция циклического сдвига 16-и битного значения в любую сторону на 8 бит меняет местами его старший и младший байты.

## 24.2 Примеры

Десятичное представление

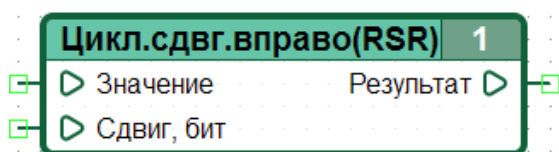


Десятичное представление



## 25 Циклический сдвиг вправо

### Внешний вид



### Символьное обозначение

RSR

### Группа

Побитовые логические операции

### Входы

Значение: числовой

Сдвиг, бит: числовой

### Выходы

Результат: числовой

### Версия CANNY Lab

с 1.8



Циклический сдвиг вправо. Устанавливает на выходе результат операции логического сдвига вправо двоичного представления значения входа «Значение», на число бит, заданное значением входа «Сдвиг, бит», при этом сдвиге уходящий бит появляется на месте появившегося свободного на другом конце числа.

Иными словами, результат эквивалентен результату операции копирования каждого бита в двоичном представлении входа «Значение» в позицию справа от него, произведенной число раз, заданное значением входа «Сдвиг, бит». При этом старший (самый левый) бит в двоичном представлении результата каждый раз получает значение, равное уходящему (самому младшему, выдвигаемому вправо) биту входного значения.

Например:

Значение = 51001 = 0xC739 = 0b1100011100111001

Сдвиг, бит = 2

Результат = 29134 = 0x71CE = 0b0111000111001110

## 25.1 Особенности

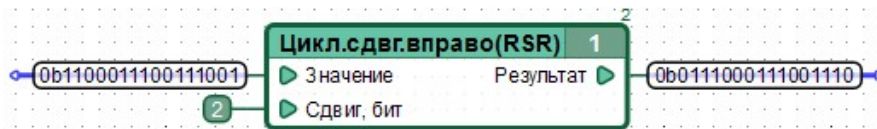
Операция циклического сдвига 16-и битного значения в любую сторону на 8 бит меняет местами его старший и младший байты.

## 25.2 Примеры

Десятичное представление

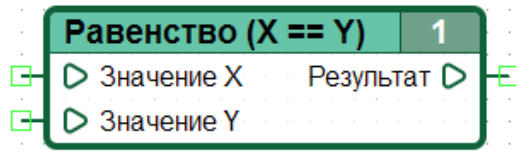


Двоичное представление



# 26 Равенство

## Внешний вид



## Символьное обозначение

$$X == Y$$

## Группа

Операции сравнения

## Входы

Значение X:            числовой  
Значение Y:            числовой

## Выходы

Результат:            логический

## Версия CANNY Lab

с 0.6b



Устанавливает на выходе результат проверки равенства значений входов.

Иными словами, на выходе будет:

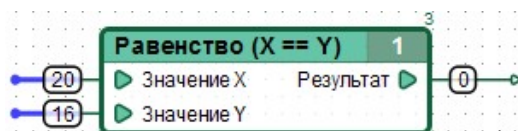
- «1» тогда и только тогда, когда значения входов численно равны друг другу,
- «0» тогда и только тогда, когда значения входов численно не равны друг другу.

Таким образом, данная операция может быть задана таблицей вида:

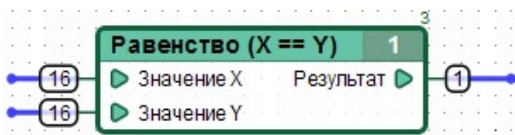
Отношение значений X и Y	Результат операции
$X = Y$	1
$X \neq Y$	0

## 26.1 Примеры

### Пример 1

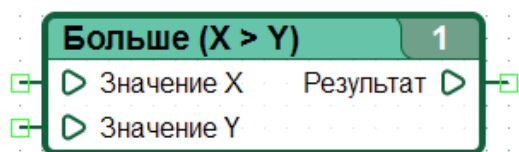


### Пример 2



# 27 Больше

## Внешний вид



## Символьное обозначение

$X > Y$

## Группа

Операции сравнения

## Входы

Значение X: числовой

Значение Y: числовой

## Выходы

Результат: логический

## Версия CANNY Lab

с 0.6b



Устанавливает на выходе результат проверки превышения входным значением X входного значения Y.

Иными словами, на выходе будет:

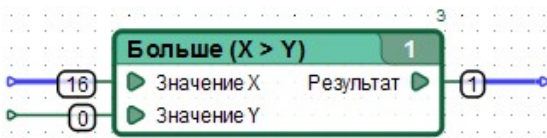
- «1» тогда и только тогда, когда «Значение X» численно больше «Значения Y»,
- «0» тогда и только тогда, когда «Значение X» численно меньше или равно «Значению Y».

Таким образом, данная операция может быть задана таблицей вида:

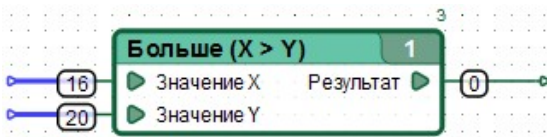
Отношение значений X и Y	Результат операции
$X > Y$	1
$X = Y$	0
$X < Y$	0

## 27.1 Примеры

Пример 1

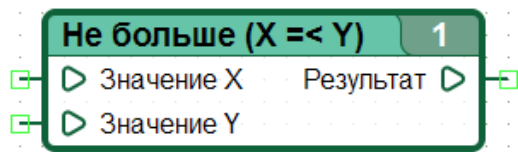


Пример 2



# 28 Не больше

## Внешний вид



## Символьное обозначение

$$X \leq Y$$

## Группа

Операции сравнения

## Входы

Значение X: числовой

Значение Y: числовой

## Выходы

Результат: логический

## Версия CANNY Lab

с 0.6b



Устанавливает на выходе результат проверки не превышения входным значением X входного значения Y.

Иными словами, на выходе будет:

- «1» тогда и только тогда, когда «Значение X» численно меньше или равно «Значению Y»,
- «0» тогда и только тогда, когда «Значение X» численно больше «Значения Y».

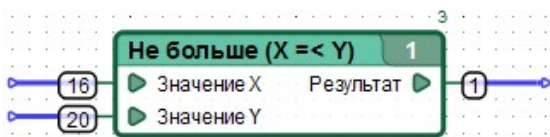
Таким образом, данная операция может быть задана таблицей вида:

Отношение значений X и Y	Результат операции
$X > Y$	0
$X = Y$	1
$X < Y$	1

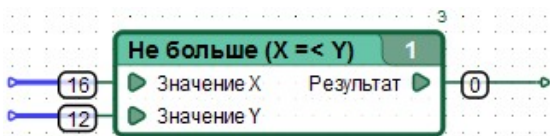


## 28.1 Примеры

Пример 1

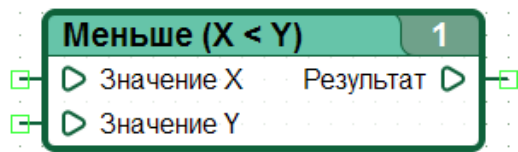


Пример 2



# 29 Меньше

## Внешний вид



## Символьное обозначение

$X < Y$

## Группа

Операции сравнения

## Входы

Значение X: числовой

Значение Y: числовой

## Выходы

Результат: логический

## Версия CANNY Lab

с 0.6b



Устанавливает на выходе результат проверки превышения входным значением Y входного значения X.

Иными словами, на выходе будет:

- «1» тогда и только тогда, когда «Значение X» численно меньше «Значения Y»,
- «0» тогда и только тогда, когда «Значение X» численно больше или равно «Значению Y».

Таким образом, данная операция может быть задана таблицей вида:

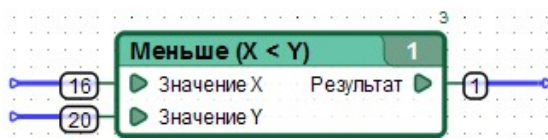
Отношение значений X и Y	Результат операции
$X > Y$	0
$X = Y$	0
$X < Y$	1

## 29.1 Примеры

Пример 1

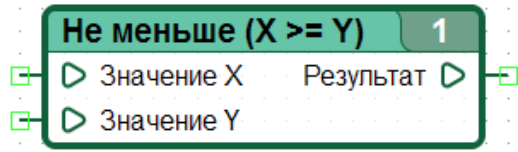


Пример 2



# 30 Не меньше

## Внешний вид



## Символьное обозначение

$X \geq Y$

## Группа

Операции сравнения

## Входы

Значение X: числовой

Значение Y: числовой

## Выходы

Результат: логический

## Версия CANNY Lab

с 0.6b



Устанавливает на выходе результат проверки не превышения входным значением Y входного значения X.

Иными словами, на выходе будет:

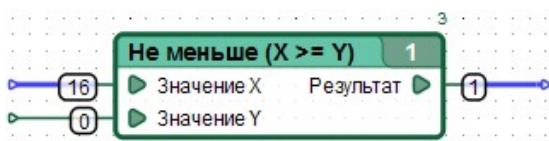
- «1» тогда и только тогда, когда «Значение X» численно больше или равно «Значению Y»,
- «0» тогда и только тогда, когда «Значение X» численно меньше «Значения Y».

Таким образом, данная операция может быть задана таблицей вида:

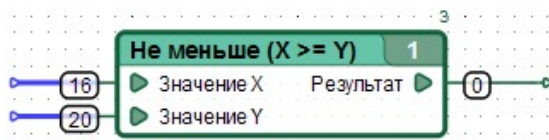
Отношение значений X и Y	Результат операции
$X > Y$	1
$X = Y$	1
$X < Y$	0

## 30.1 Примеры

Пример 1

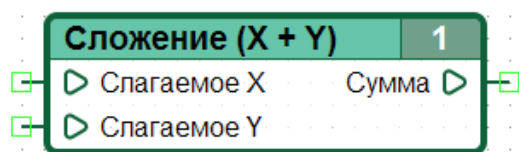


Пример 2



# 31 Сложение

## Внешний вид



## Символьное обозначение

$$X + Y$$

## Группа

Арифметические операции

## Входы

Значение X: числовой

Значение Y: числовой

## Выходы

Сумма: числовой

## Версия CANNY Lab

с 0.6b



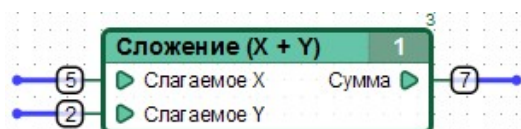
Устанавливает на выходе результат арифметического сложения численных значений входов.

## 31.1 Особенности

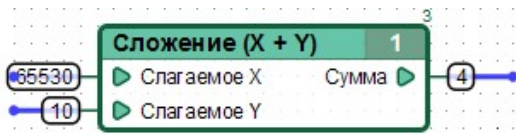
Сложение 16-и битное, с переполнением (сложение по модулю 65535). Иными словами: результат сложения не может превысить 65535, в случае, если значения входов в сумме превышают это значение, результатом будет значение выражения:  $X + Y - 65536$ .

## 31.2 Примеры

### Пример 1

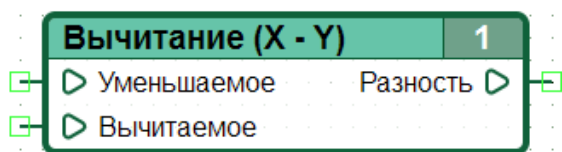


### Пример 2



# 32 Вычитание

## Внешний вид



## Символьное обозначение

$X - Y$

## Группа

Арифметические операции

## Входы

Уменьшаемое: числовой

Вычитаемое: числовой

## Выходы

Разность: числовой

## Версия CANNY Lab

с 0.6b



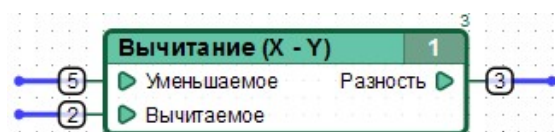
Устанавливает на выходе результат арифметического вычитания численного значения второго входа из первого.

## 32.1 Особенности

Вычитание 16-и битное, с переполнением (разность по модулю 65535). Иными словами: результат вычитания не может быть отрицательным числом, в случае, если значение второго входа превышает значение первого, будет значение выражения:  $X - Y + 65536$ .

## 32.2 Примеры

### Пример 1



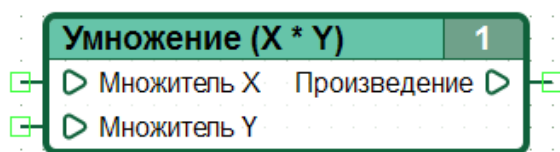
### Пример 2





# 33 Умножение

## Внешний вид



## Символьное обозначение

$X * Y$

## Группа

Арифметические операции

## Входы

Множитель X: числовой

Множитель Y: числовой

## Выходы

Произведение: числовой

## Версия CANNY Lab

с 0.6b



Устанавливает на выходе результат арифметического умножения численных значений входов.

## 33.1 Особенности

Умножение 16-и битное, с переполнением (умножение по модулю 65535). Иными словами: результат умножения не может превысить 65535, в случае, если произведение значений входов превышает это значение, результатом будет остаток от деления этого произведения на 65536.

## 33.2 Примеры

### Пример 1

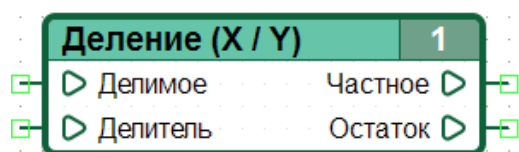


### Пример 2



# 34 Деление

## Внешний вид



## Символьное обозначение

$X / Y$

## Группа

Арифметические операции

## Входы

Делимое: числовой

Делитель: числовой

## Выходы

Частное: числовой

Остаток: числовой

## Версия CANNY Lab

с 1.8



Устанавливает на выходе «Частное» результат целочисленного деления значения на входе «Делимое» на значение на входе «Делитель», остаток от деления устанавливается на выходе «Остаток».

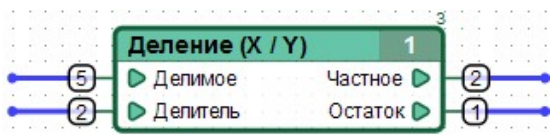
## 34.1 Особенности

Исключение: частным от деления любого числа на ноль будет число 65535, при остатке равном также 65535.

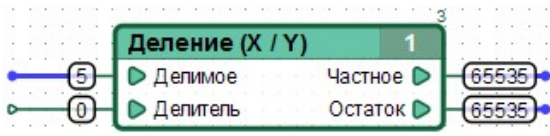
До выхода CANNY Lab версии 1.8, данный функциональный блок был разделен на два блока: 'Деление' и 'Остаток от деления'.

## 34.2 Примеры

Пример 1

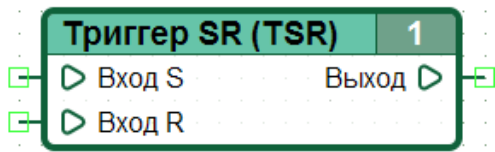


Пример 2



# 35 Триггер SR

## Внешний вид



## Символьное обозначение

TSR

Группа

Триггеры

Входы

Вход S:            числовой

Вход R:            логический

Выходы

Выход:            числовой

Версия CANNY Lab

с 0.6b



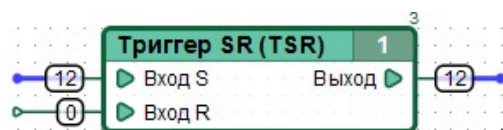
Триггер, который не изменяет значение на выходе при нулевых входах. При наличии ненулевого значения на входе «S» (от англ. Set — установить), это значение копируется на выход. А при наличии ненулевого значения на входе «R» (от англ. Reset — сбросить) выходное состояние становится равным нулю.

## 35.1 Особенности

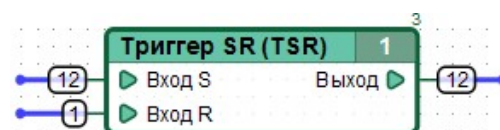
В ситуации, когда на обоих входах установлены ненулевые значения, приоритет у SR-триггера получает вход «S» (значение входа «S» копируется на выход).

## 35.2 Примеры

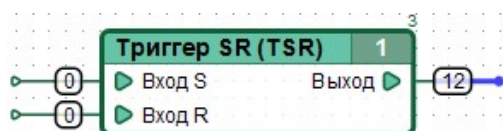
Шаг 1



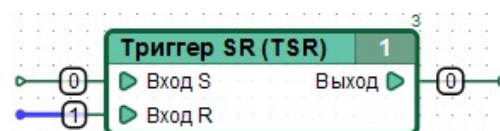
Шаг 2



Шаг 3



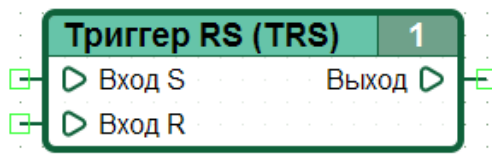
Шаг 4





# 36 Триггер RS

## Внешний вид



## Символьное обозначение

TRS

## Группа

Триггеры

## Входы

Вход S:            числовой  
Вход R:            логический

## Выходы

Выход:            числовой

## Версия CANNY Lab

с 0.6b



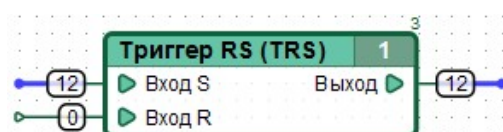
Триггер, который не изменяет значение на выходе при нулевых значениях на обоих входах и меняет своё выходное состояние при наличии на одном из входов ненулевого значения. При наличии ненулевого значения на входе «S» (от англ. Set — установить), это значение копируется на выход. А при наличии ненулевого значения на входе «R» (от англ. Reset — сбросить) на выходе устанавливается значение «0».

## 36.1 Особенности

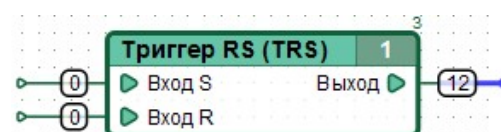
В ситуации, когда на обоих входах установлены ненулевые значения, приоритет у RS-триггера получает вход «R» (на выходе устанавливается значение «0»).

## 36.2 Примеры

Шаг 1



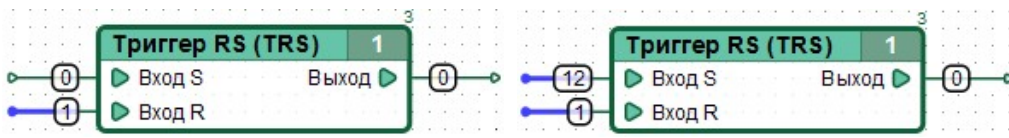
Шаг 2



Шаг 3

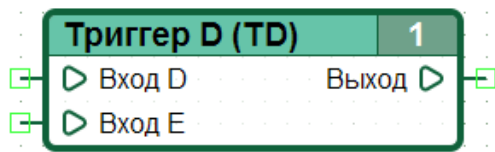
Шаг 4





# 37 Триггер D

## Внешний вид



## Символьное обозначение

TD

## Группа

Триггеры

## Входы

Вход D: числовой

Вход E: логический

## Выходы

Выход: числовой

## Версия CANNY Lab

с 0.6b



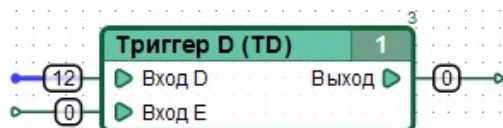
Триггер, который не изменяет (сохраняет) последнее установленное значение на выходе, при нулевом значении на входе «Е». При наличии ненулевого значения на входе «Е», значение входа «D» копируется на выход.

## 37.1 Особенности

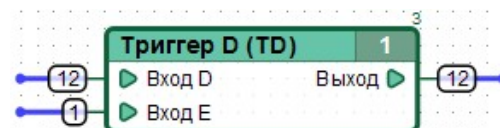
D-триггеры обычно используются в качестве ячеек оперативной памяти.

## 37.2 Примеры

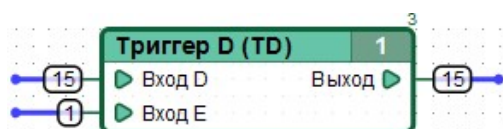
Шаг 1



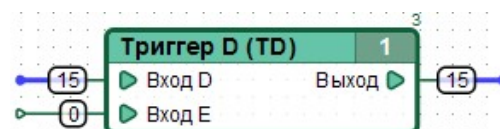
Шаг 2



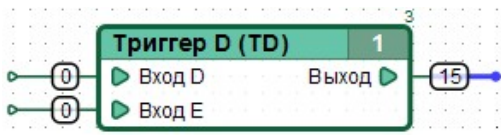
Шаг 3



Шаг 4

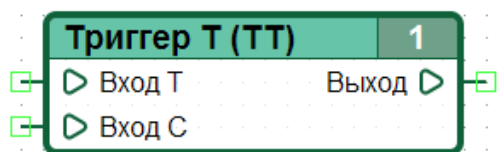


Шаг 5



# 38 Триггер Т

## Внешний вид



## Символьное обозначение

ТТ

## Группа

Триггеры

## Входы

Вход Т: числовой

Вход С: логический

## Выходы

Выход: числовой

## Версия CANNY Lab

с 0.6b



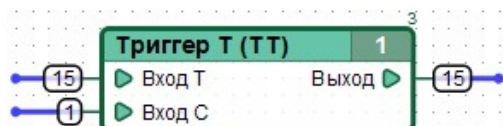
Синхронный триггер, который, при наличии ненулевого значения на входе «Т», изменяет значение на выходе по переднему фронту на входе «С».

Изменения выполняются по каждому переднему фронту (изменению значения с нулевого на ненулевое) на входе «С»:

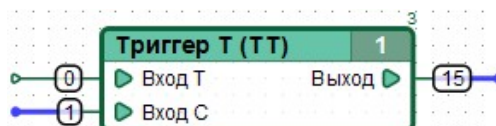
- если значение выхода равно «0», на выход копируется значение входа «Т».
- если значение выхода не равно «0», на выход копируется значение «0».

## 38.1 Примеры

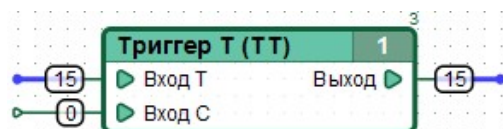
Шаг 1



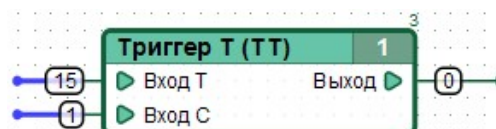
Шаг 2



Шаг 3

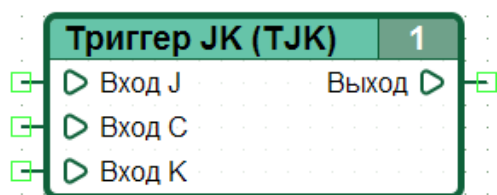


Шаг 4



# 39 Триггер JK

## Внешний вид



## Символьное обозначение

ТJK

### Группа

Триггеры

### Входы

Вход J:            числовой  
Вход C:            логический  
Вход K:            логический

### Выходы

Выход:            числовой

## Версия CANNY Lab

с 0.11b



Синхронный двухступенчатый JK триггер изменяет свое внутреннее состояние по переднему фронту на входе «С», в зависимости от состояния входов «J» и «K», и копирует свое внутреннее состояние на выход по заднему фронту на входе «С».

Варианты работы триггера в зависимости от состояния его входов:

- Если вход «J» = 0 и «K» = 0 — внутреннее состояние не изменится
- Если вход «J» = 0, а «K» > 0 — внутреннее состояние станет равным нулю
- Если вход «J» > 0, а «K» = 0 — внутреннее состояние станет равным значению «J»
- Если вход «J» > 0 и «K» > 0, то:
  - ◆ если текущее внутреннее состояние равно нулю, то оно станет равным значению «J»
  - ◆ если текущее значение выхода не равно нулю, то оно станет равно нулю

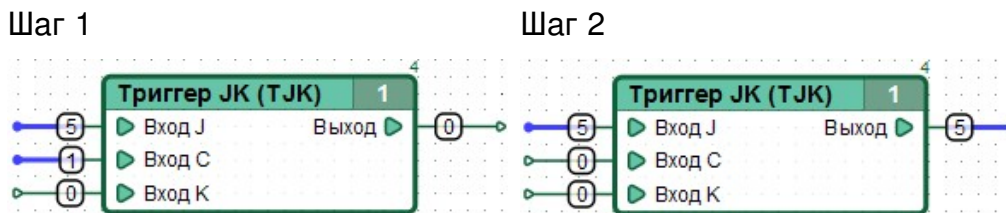
В промежутке времени между фронтами на входе «С» внутреннее состояние и состояние выхода триггера остаются неизменным вне зависимости от значений на входах «J» или «K».

## 39.1 Особенности

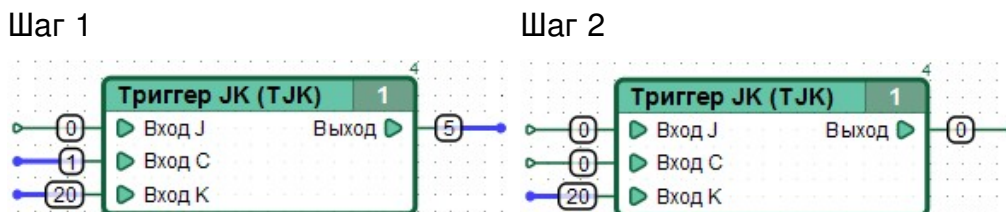
Цепочка из JK-триггеров может использоваться для эффективной организации очередей (FIFO буферов).

## 39.2 Примеры

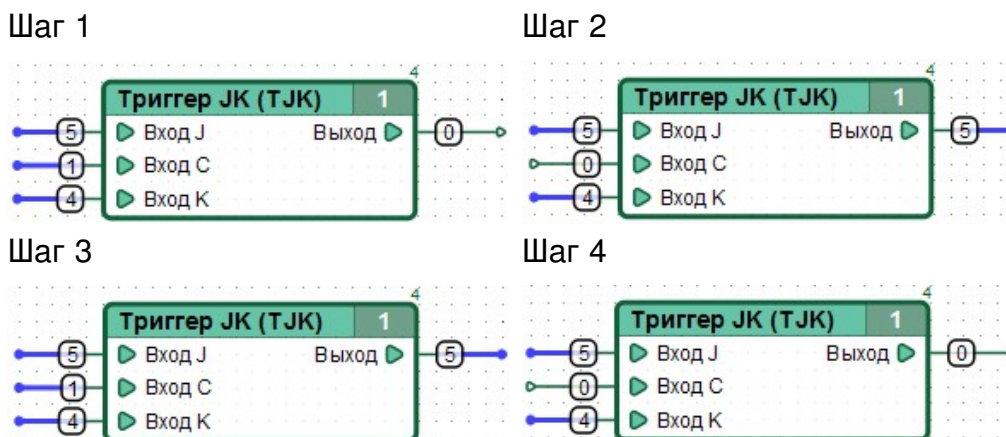
Пример 1:



Пример 2:

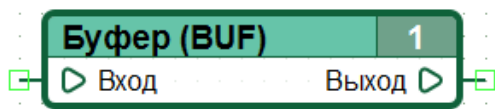


Пример 3:



# 40 Буфер

## Внешний вид



## Символьное обозначение

BUF

## Группа

Триггеры

## Входы

Вход: числовой

## Выходы

Выход: числовой

## Версия CANNY Lab

с 1.18



Отложенное копирование значения.

На каждом цикле выполнения диаграммы, функциональный блок запоминает значение установленное на его входе, чтобы на следующем цикле выполнения диаграммы передать его свой выход. Начальное значение на выходе функционального блока равно нулю.

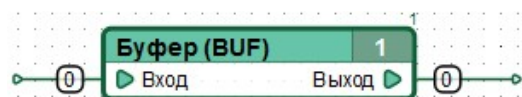
## 40.1 Особенности

Пропуская значение через блок "Буфер" можно определить момент изменения этого значения, сравнивая значение входа и выхода блока.

Цепочка из блоков "Буфер" может использоваться для организации очередей (FIFO буферов).

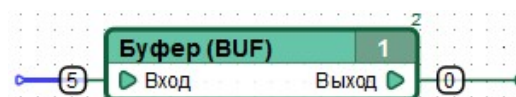
## 40.2 Примеры

Шаг 1

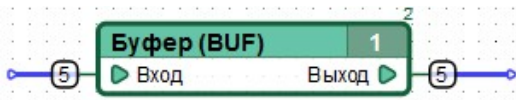


Шаг 3

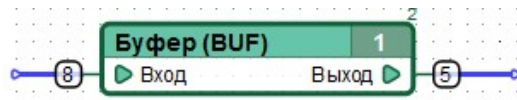
Шаг 2



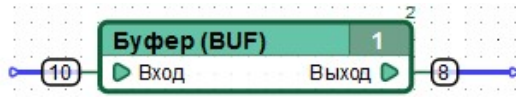
Шаг 4



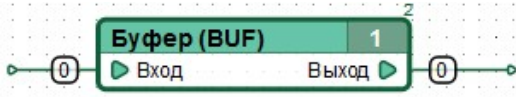
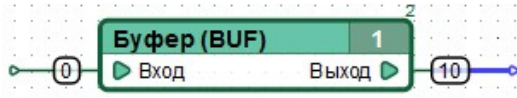
Шаг 5



Шаг 6



Шаг 7





# 41 Задержка включения

## Внешний вид



## Символьное обозначение

DLN

## Группа

Таймеры и генераторы

## Входы

Вход: числовой

Задержка: числовой

## Выходы

Выход: числовой

## Версия CANNY Lab

с 0.6b



При удержании входного значения не равным нулю в течение заданного времени устанавливает выходное значение равным входному.

При изменении значения на входе «Вход» на ненулевое, начинает отсчет времени, при удержании входного значения отличным от нуля в течение времени не меньшем чем заданное входом «Задержка, мс», выполняет копирование входного значения на выход. При изменении входного значения на нулевое, сбрасывает отсчет времени и устанавливает выходное значение равным нулю.

## 41.1 Особенности

При задании временных значений на диаграмме, необходимо учитывать длительность цикла выполнения диаграммы, которая зависит от модели контроллера, размера диаграммы, режима работы контроллера, загруженности периферии контроллера и может изменяться от цикла к циклу в определенных пределах в процессе работы. Точное время исполнения каждого цикла диаграммы во время работы контроллера доступно в регистре "Длительность программного цикла".

Временные интервалы в диаграмме не должны задаваться значениями меньшими чем длительность одного цикла и будут измеряться с погрешностью порядка длительности одного цикла пользовательской диаграммы.

Хорошей практикой при работе с контроллерами CANNY 7 является использование временных интервалов больших чем 5-10 мс.

## 41.2 Примеры

Пример 1

$T=0\text{мс}$



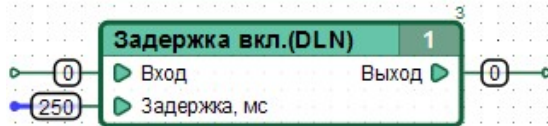
$T=100\text{мс}$



$T>250\text{мс}$



$T>250\text{мс}$



Пример 2

$T=0\text{мс}$



$T=100\text{мс}$



$T=200\text{мс}$



$T>250\text{мс}$



# 42 Задержка выключения

## Внешний вид



## Символьное обозначение

DLF

## Группа

Таймеры и генераторы

## Входы

Вход: числовой

Задержка: числовой

## Выходы

Выход: числовой

## Версия CANNY Lab

с 0.6b



После изменения входного значения на ноль, в течение заданного времени поддерживает на выходе значение, равное последнему ненулевому входному значению.

При удержании на входе «Вход» ненулевого значения, копирует его на выход. При изменении значения на входе на нулевое, не изменяет выходное значение и начинает отсчет времени. При удержании входного значения равным нулю в течение времени не меньшем чем заданное входом «Задержка, мс», устанавливает выходное значение равным нулю. При изменении входного значения на ненулевое, сбрасывает отсчет времени.

## 42.1 Особенности

При задании временных значений на диаграмме, необходимо учитывать длительность цикла выполнения диаграммы, которая зависит от модели контроллера, размера диаграммы, режима работы контроллера, загруженности периферии контроллера и может изменяться от цикла к циклу в определенных пределах в процессе работы. Точное время исполнения каждого цикла диаграммы во время работы контроллера доступно в регистре "Длительность программного цикла".

Временные интервалы в диаграмме не должны задаваться значениями меньшими чем длительность одного цикла и будут измеряться с погрешностью порядка длительности одного цикла пользовательской диаграммы.

Хорошей практикой при работе с контроллерами CANNY 7 является использование временных интервалов больших чем 5-10 мс.

## 42.2 Примеры

$T=0\text{мс}$



$T=1\text{мс}$



$T=100\text{мс}$



$T>250\text{мс}$



# 43 Генератор импульсов

## Внешний вид



## Символьное обозначение

PWM

## Группа

Таймеры и генераторы

## Входы

Период, мс:                    числовой  
Заполнение, мс:                числовой  
Старт:                            логический

## Выходы

Выход:                            логический

## Версия CANNY Lab

с 0.6b



При ненулевом значении на входе «Старт», генерирует на выходе блока единицу с заданной длительностью и периодичностью.

Параметры импульсного сигнала



**Для асинхронной генерации высокоточного ШИМ-сигнала физического уровня на выходе канала контроллера используйте соответствующие ресурсы контроллера, например для контроллеров CANNY 7 - Драйвер каналов ввода-вывода или Драйвер ВЧ ШИМ**

## 43.1 Особенности

При задании временных значений на диаграмме, необходимо учитывать длительность цикла выполнения диаграммы, которая зависит от модели контроллера, размера диаграммы, режима работы контроллера, загруженности периферии контроллера и может изменяться от цикла к циклу в определенных пределах в процессе работы. Точное время исполнения каждого цикла диаграммы во время работы контроллера доступно в регистре "Длительность программного цикла".

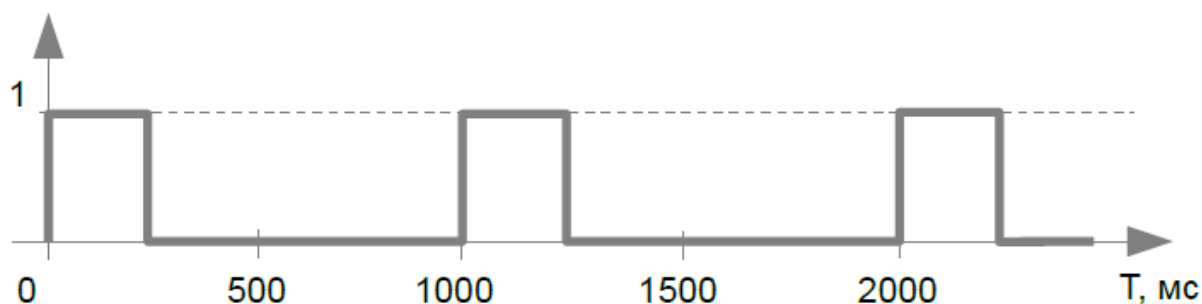
Временные интервалы в диаграмме не должны задаваться значениями меньшими чем длительность одного цикла и будут измеряться с погрешностью порядка длительности одного цикла пользовательской диаграммы.

Хорошей практикой при работе с контроллерами CANNY 7 является использование временных интервалов больших чем 5-10 мс.

## 43.2 Примеры

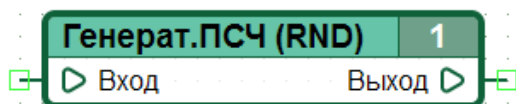


Временная диаграмма



# 44 Генератор псевдослучайных чисел

## Внешний вид



## Символьное обозначение

RND

## Группа

Таймеры и генераторы

## Входы

Вход: числовой

## Выходы

Выход: числовой

## Версия CANNY Lab

с 0.6b



Триггер, который устанавливает на своем выходе значение функции-генератора псевдослучайных чисел в диапазоне 0...65535 от значения Входа.

## 44.1 Особенности

Выходное значение, вычисленное для одного и того же значения Входа будет одним и тем же.

Генератор выполнен по простейшей схеме и не является криптографически стойким.

## 44.2 Примеры

Пример 1

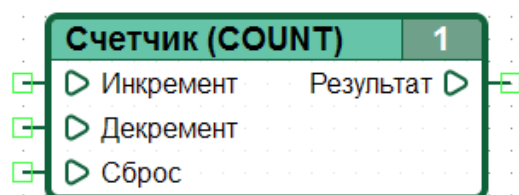


Пример 2



# 45 Счетчик

## Внешний вид



## Символьное обозначение

COUNT

## Группа

Счетчики и детекторы

## Входы

Инкремент: логический

Декремент: логический

Сброс: логический

## Выходы

Результат: числовой

## Версия CANNY Lab

с 0.6b



При нулевом значении на входе «Сброс»: по каждому переднему фронту (изменению значения с нулевого на ненулевое) на входе «Инкремент», прибавляет к выходному значению единицу, а по каждому переднему фронту (изменению значения с нулевого на ненулевое) на входе «Декремент», вычитает из выходного значения единицу.

При ненулевом значении на входе «Сброс», устанавливает выходное значение равным нулю.

## 45.1 Особенности

Выходное значение 16-и битное, с переполнением. Иными словами: выходное значение не может превысить 65535 или стать отрицательным числом. В случае увеличения выходного значения равного 65535 на единицу, результатом будет ноль, а в случае уменьшения выходного значения равного нулю на единицу, результатом будет 65535.



## 45.2 Примеры

Шаг 1



Шаг 2



Шаг 3



Шаг 4

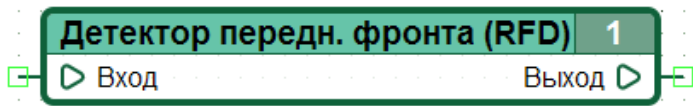


Шаг 5



# 46 Детектор переднего фронта

## Внешний вид



## Символьное обозначение

RFD

## Группа

Счетчики и детекторы

## Входы

Вход: логический

## Выходы

Выход: логический

## Версия CANNY Lab

с 0.6b



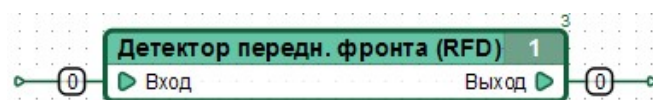
Источник сигнала минимальной длительности (один проход диаграммы), синхронизируемый передним фронтом.

Устанавливает на выходе единицу, при каждом изменении значения на входе с нулевого на ненулевое, ровно на один цикл выполнения диаграммы: при повторном входе потока выполнения в блок, выходное значение устанавливается равным нулю.

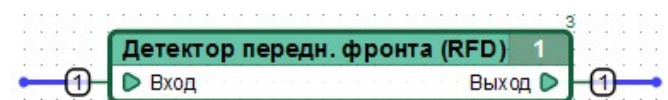
## 46.1 Особенности

## 46.2 Примеры

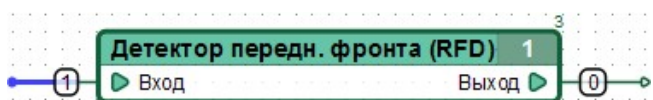
Шаг 1



Шаг 2

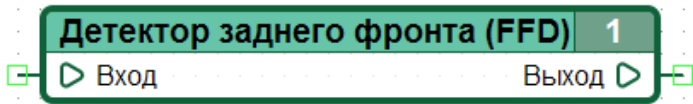


Шаг 3



# 47 Детектор заднего фронта

## Внешний вид



## Символьное обозначение

FFD

## Группа

Счетчики и детекторы

## Входы

Вход: логический

## Выходы

Выход: логический

## Версия CANNY Lab

с 0.6b



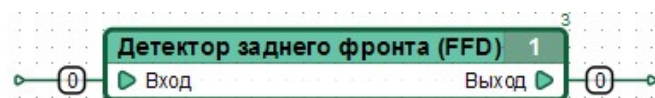
Источник сигнала минимальной длительности, синхронизируемый задним фронтом.

Устанавливает на выходе единицу, при каждом изменении значения на входе с ненулевого на нулевое, ровно на один цикл выполнения диаграммы: при повторном входе потока выполнения в блок, выходное значение устанавливается равным нулю.

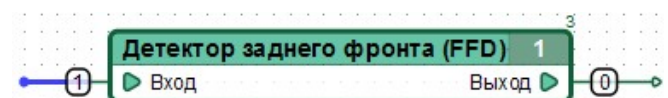
## 47.1 Особенности

## 47.2 Примеры

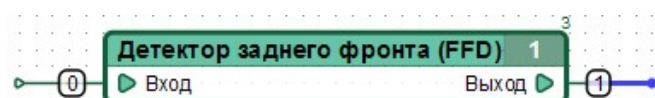
Шаг 1



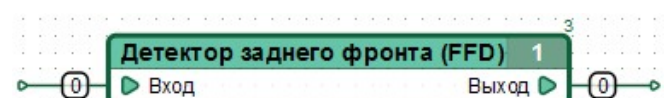
Шаг 2



Шаг 3



Шаг 4



# 48 Конвертер 8-в-1

## Внешний вид



## Символьное обозначение

b2B

## Группа

Конвертеры

## Входы

Бит 0...Бит 7: логический

## Выходы

Выход: числовой

## Версия CANNY Lab

с 0.6b



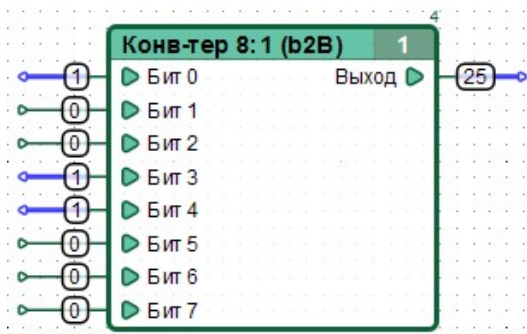
Устанавливает биты в двоичном представлении выходного значения в соответствии со значениями соответствующих входов.

## 48.1 Особенности

Блок может быть использован в качестве многоканального Логического ИЛИ.

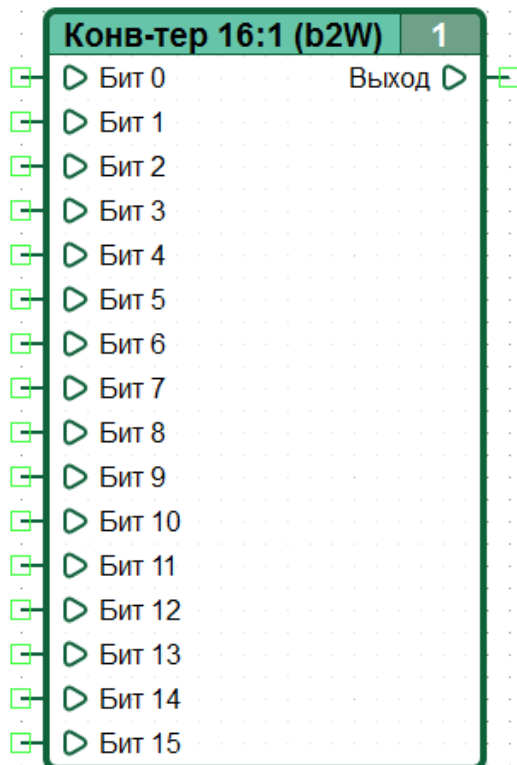
## 48.2 Примеры

Десятичное 25 = 0b00011001



# 49 Конвертер 16-в-1

## Внешний вид



## Символьное обозначение

b2W

## Группа

Конвертеры

## Входы

Бит 0...Бит 15: логический

## Выходы

Выход: числовой

## Версия CANNY Lab

с 0.6b



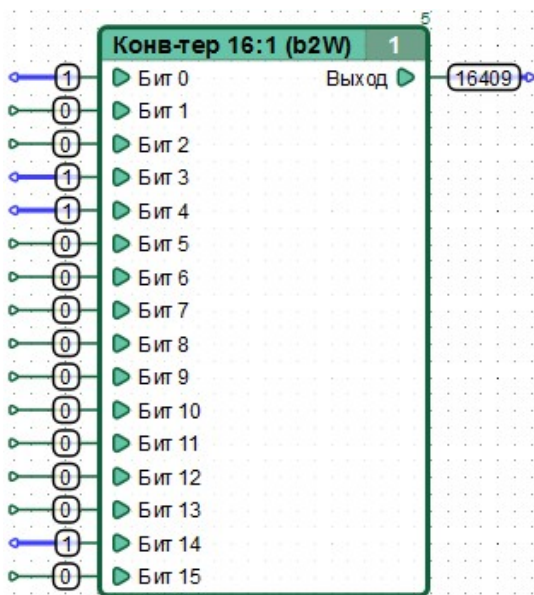
Устанавливает биты в двоичном представлении выходного значения в соответствие со значениями соответствующих входов.

## 49.1 Особенности

Блок может быть использован в качестве многоканального Логического ИЛИ.

## 49.2 Примеры

Десятичное 16409 = 0b0100000000011001



# 50 Конвертер 1-в-8

## Внешний вид



## Символьное обозначение

B2b

## Группа

Конвертеры

## Входы

Вход: числовой

## Выходы

Бит 0...Бит 7: логический

## Версия CANNY Lab

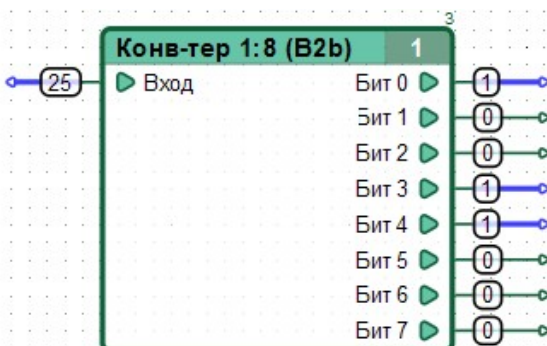
с 0.6b



Устанавливает биты в двоичном представлении выходного значения в соответствие со значениями соответствующих входов.

## 50.1 Примеры

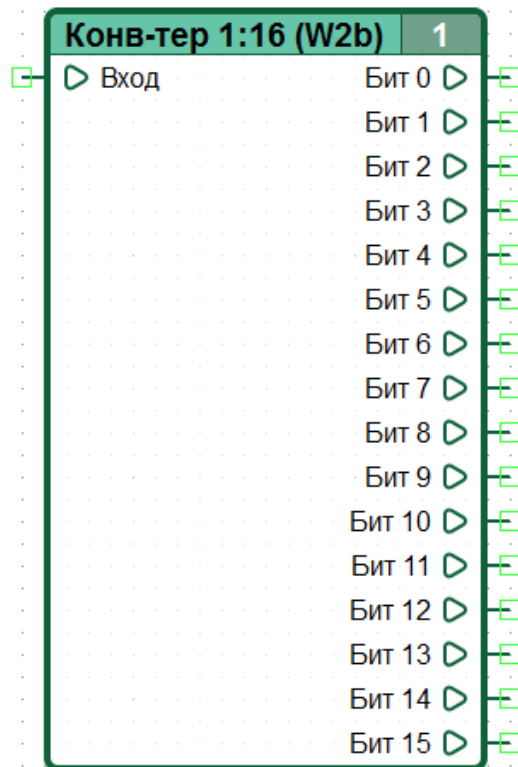
Десятичное 25 = 0b00011001





# 51 Конвертер 1-в-16

## Внешний вид



### Сокращенное название

W2b

### Группа

Конвертеры

### Входы

Вход: числовой

### Выходы

Бит 0...Бит 15: логический

### Версия CANNY Lab

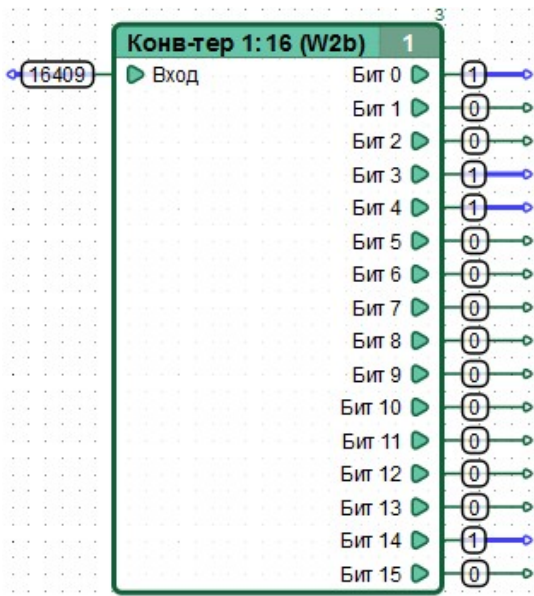
с 0.6b



Устанавливает биты в двоичном представлении выходного значения в соответствие со значениями соответствующих входов.

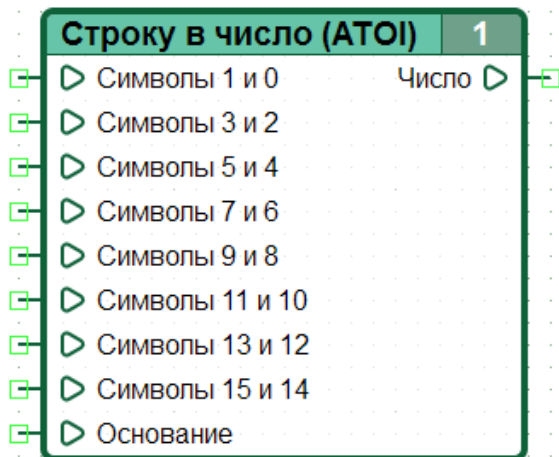
## 51.1 Примеры

Десятичное 16409 = 0b0100000000011001



# 52 Конвертер строки символов в число

## Внешний вид



## Символьное обозначение

atoi

## Группа

Конвертеры

## Входы

Символы 1 и 0: пара ASCII символов

...

Символы 15 и 14: пара ASCII символов

Основание: числовой

## Выходы

Число: числовой

## Версия CANNY Lab

с 1.8



Преобразует строковое представление числа по заданному основанию 2, 10 или 16 в числовое значение.

Исходная строка символов должна содержать допустимые в выбранной системе счисления символы на позициях: 0..15, 11..15 или 12..15 для оснований 2, 10 и 16 соответственно.

## 52.1 Особенности

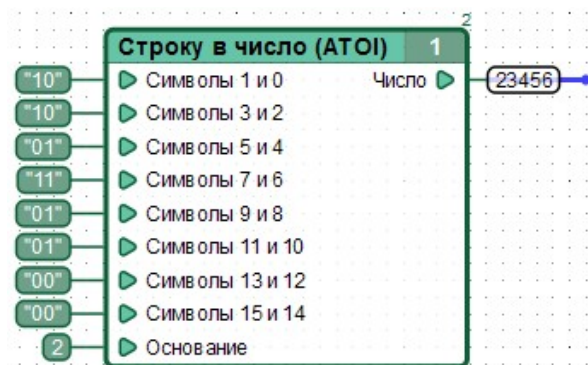
Символы установленные на не входящих в данные диапазоны позициях отбрасываются.

В случае наличия на входах блока недопустимых символов, значение на выходе будет установлено равным 0.

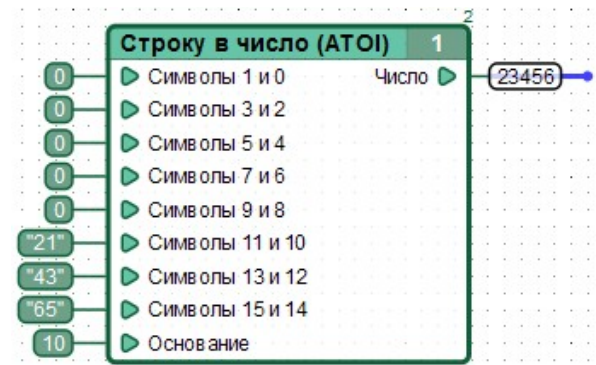
В случае использования недопустимого значения основания, значение на выходе будет установлено равным 0.

## 52.2 Примеры

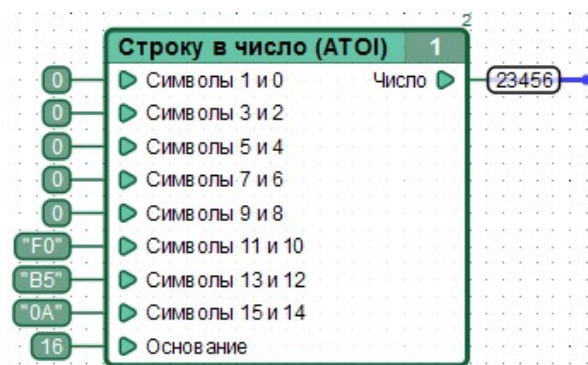
«0101101110100000» по основанию 2 = 23456



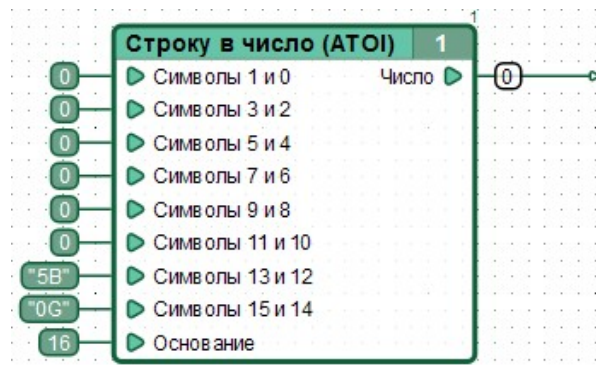
«0000000000123456» по основанию 10 = 23456 (символ "1" отброшен)



«00000000000F5Ba0» по основанию 16 = 23456 (символ "F" отброшен)



«000000000005BG0» по основанию 16 = 0 (недопустимый символ "G")



## 52.3 Примечания

С выходом CANNY Lab версии 1.8 в добавлено новое представление данных - «Пара символов» в котором значения представлены двумя символами ASCII с кодами заданными старшим и младшим байтом отображаемого 16-битного значения.

Так, десятичное число 19778 соответствующее шестнадцатеричному 0x4B4E, отображается парой символов «KN», имеющими ASCII-коды 0x4B и 0x4E соответственно.



# 53 Конвертер числа в строку символов

## Внешний вид



## Символьное обозначение

ИТОА

Группа

Конвертеры

Входы

Число: числовой

Основание: числовой

Выходы

Символы 1 и 0: пара ASCII символов

...

Символы 15 и 14: пара ASCII символов

Версия CANNY Lab

с 1.8



Преобразует числовое значение в шестнадцатизначное строковое представление по заданному основанию 2, 10 или 16.

## 53.1 Особенности

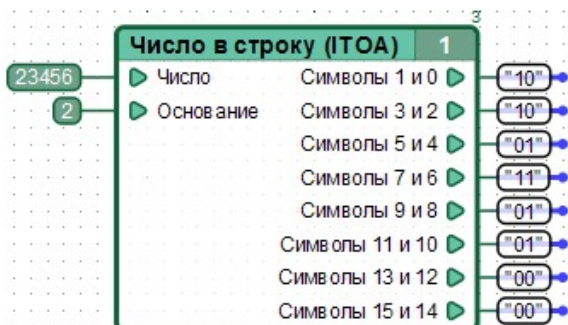
Символы результирующего, выравненного по правому краю строкового представления заданного на входе числа, устанавливаются на соответствующих выходах в верхнем регистре, незадействованные старшие разряды заполняются символами «0».

В случае использования недопустимого значения основания, значение на выходе будет

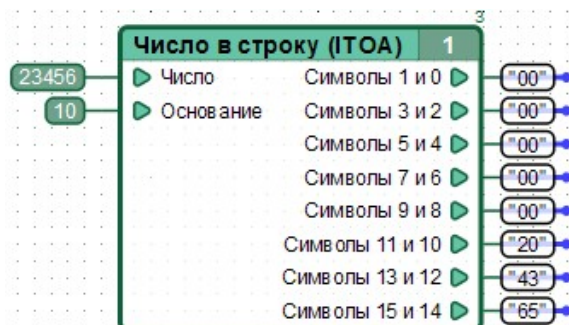
установлено равным «0000000000000000».

## 53.2 Примеры

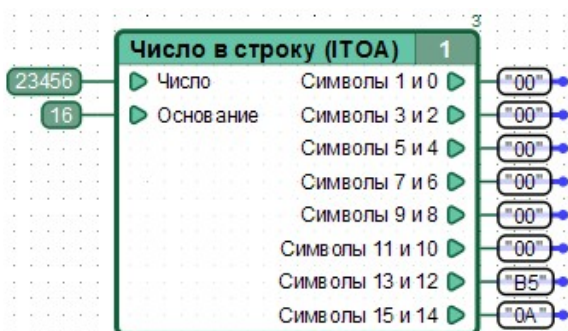
23456 по основанию 2 =  
«010110111010000»



23456 по основанию 10 =  
«0000000000023456»



23456 по основанию 16 =  
«0000000000005BA0»



## 53.3 Примечания

С выходом CANNY Lab версии 1.8 в добавлено новое представление данных - «Пара символов» в котором значения представлены двумя символами ASCII с кодами заданными старшим и младшим байтом отображаемого 16-битного значения.

Так, десятичное число 19778 соответствующее шестнадцатеричному 0x4B4E, отображается парой символов «KN», имеющими ASCII-коды 0x4B и 0x4E соответственно.

# 54 Отображение

## Внешний вид



## Символьное обозначение

MAP

## Группа

Конвертеры

## Входы

Значение:	числовой
Значение от:	числовой
Значение до:	числовой
Результат от:	числовой
Результат до:	числовой

## Выходы

Результат:	числовой
------------	----------

## Версия CANNY Lab

с 1.18



Отображает (масштабирует) значение из диапазона значений в результат из диапазона результатов методом **линейной интерполяции**.

Расчет результата выполняется по формуле:

$$\text{Результат} = \frac{(\text{Результат до} - \text{Результат от}) \cdot (\text{Значение} - \text{Значение от})}{(\text{Значение до} - \text{Значение от})} + \text{Результат от}$$

Дробная часть результата отбрасывается.

При знаменателе равном нулю, значение на выходе блока будет равно 65535

Выход входного значение за пределы указанного диапазона не проверяется, значение результат при этом рассчитывается по той же, указанной выше, формуле.



## 54.1 Особенности

Блок "Отображение" удобно применять для перевода значений из одной размерности в другую, избегая громоздких алгоритмов обработки значений большой разрядности.

Границы диапазонов значений и результата могут задавать как возрастающие, так и убывающие последовательности в различных комбинациях. Таким образом можно, например, при возрастании входного значения получить убывающий результат.

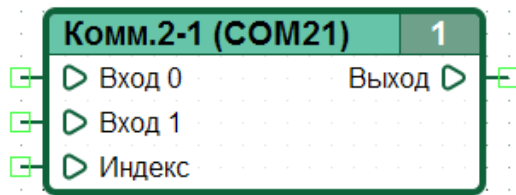
## 54.2 Примеры

Отображение значения 12-битного 5В АЦП в соответствующую ему величину напряжения в милливольтях



# 55 Коммутатор 2-в-1

## Внешний вид



## Символьное обозначение

COM21

## Группа

Коммутаторы

## Входы

Вход 0: числовой

Вход 1: числовой

Индекс: числовой

## Выходы

Выход: числовой

## Версия CANNY Lab

с 0.6b



Устанавливает выходное значение равным значению на входе, определяемом значением входа «Индекс».

## 55.1 Особенности

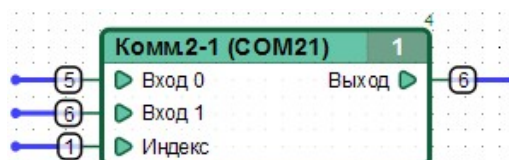
Если значение входа «Индекс» превышает число доступных входов блока, на выходе устанавливается нулевое значение.

## 55.2 Примеры

Пример 1

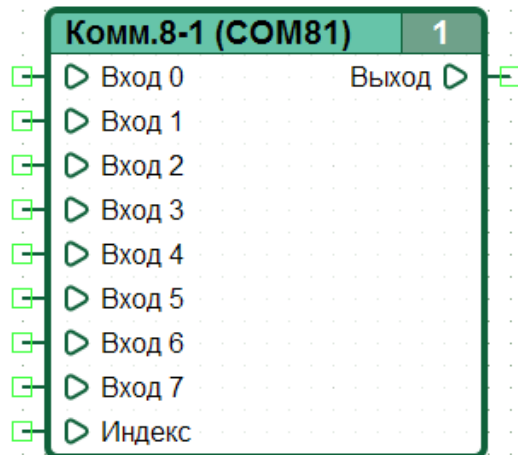


Пример 2



# 56 Коммутатор 8-в-1

## Внешний вид



## Символьное обозначение

COM81

### Группа

Коммутаторы

### Входы

Вход 0...7:            числовой

Индекс:                числовой

### Выходы

Выход:                 числовой

### Версия CANNY Lab

с 0.6b



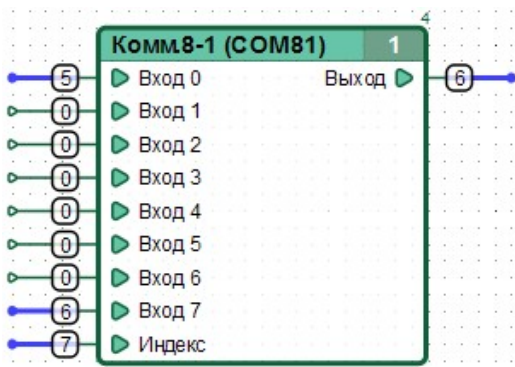
Устанавливает выходное значение равным значению на входе, определяемом значением входа «Индекс».

## 56.1 Особенности

Если значение входа «Индекс» превышает число доступных входов блока, на выходе устанавливается нулевое значение.

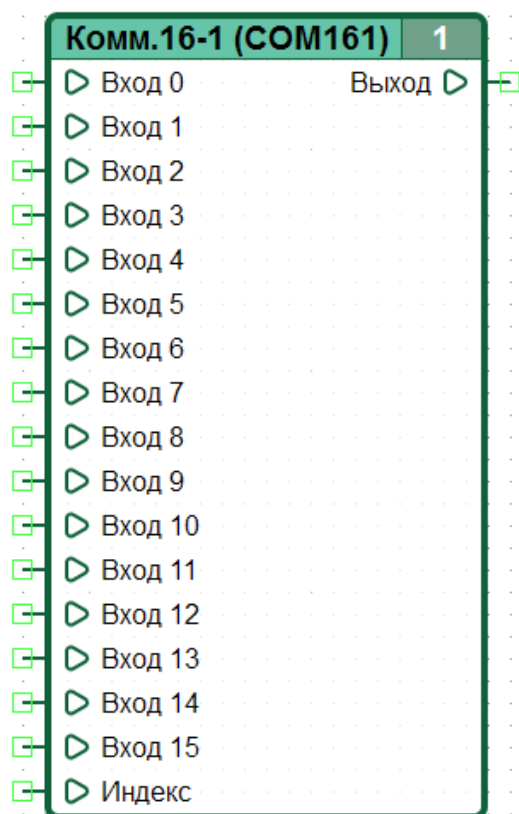
## 56.2 Примеры

Пример 1



# 57 Коммутатор 16-в-1

## Внешний вид



## Символьное обозначение

COM161

### Группа

Коммутаторы

### Входы

Вход 0...15: числовой

Индекс: числовой

### Выходы

Выход: числовой

## Версия CANNY Lab

с 0.6b

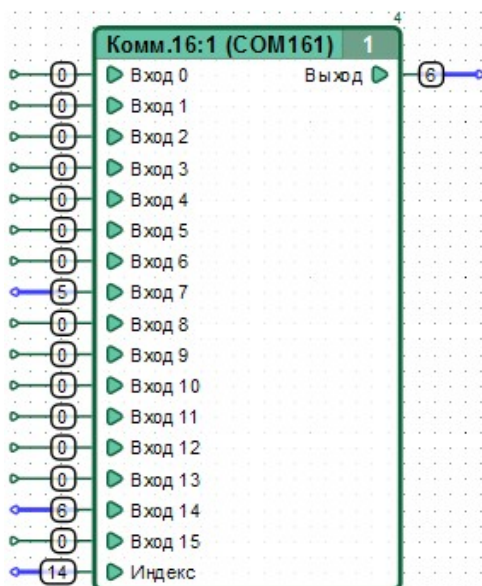


Устанавливает выходное значение равным значению на входе, определяемом значением входа «Индекс».

## 57.1 Особенности

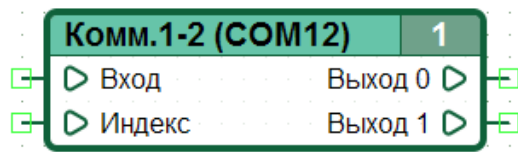
Если значение входа «Индекс» превышает число доступных входов блока, на выходе устанавливается нулевое значение.

## 57.2 Примеры



# 58 Коммутатор 1-в-2

## Внешний вид



## Символьное обозначение

COM12

## Группа

Коммутаторы

## Входы

Вход: числовой

Индекс: числовой

## Выходы

Выход 0: числовой

Выход 1: числовой

## Версия CANNY Lab

с 0.6b



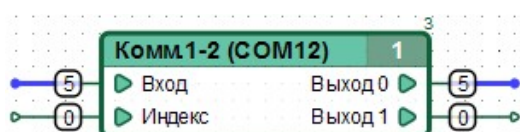
Устанавливает на выходе, определяемом значением входа «Индекс» значение равное значению входа «Вход», на остальных выходах устанавливает нулевые значения.

## 58.1 Особенности

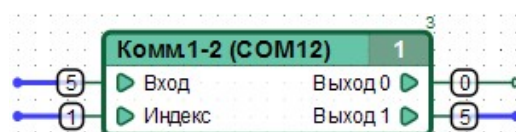
Если значение входа «Индекс» превышает число доступных выходов блока, на всех выходах устанавливаются нулевые значения.

## 58.2 Примеры

Пример 1

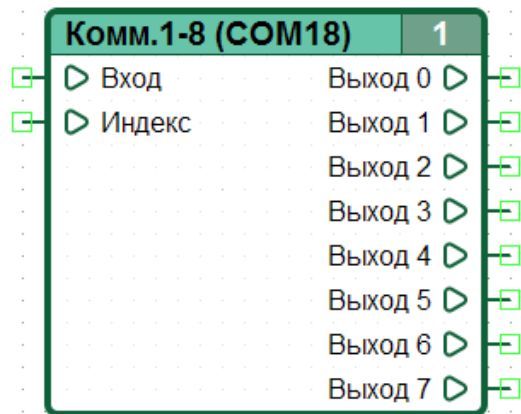


Пример 2



# 59 Коммутатор 1-в-8

## Внешний вид



## Символьное обозначение

COM18

### Группа

Коммутаторы

### Входы

Вход: числовой

Индекс: числовой

### Выходы

Выход 0...7: числовой

### Версия CANNY Lab

с 0.6b



Устанавливает на выходе, определяемом значением входа «Индекс» значение равное значению входа «Вход», на остальных выходах устанавливает нулевые значения.

## 59.1 Особенности

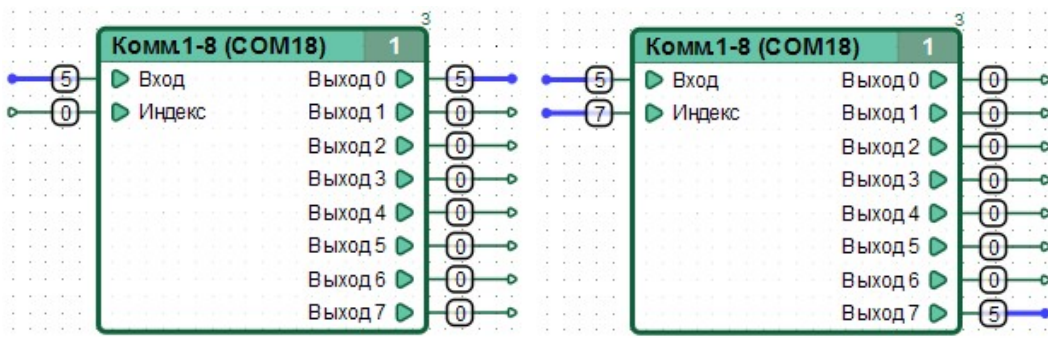
Если значение входа «Индекс» превышает число доступных выходов блока, на всех выходах устанавливаются нулевые значения.

## 59.2 Примеры

Пример 1

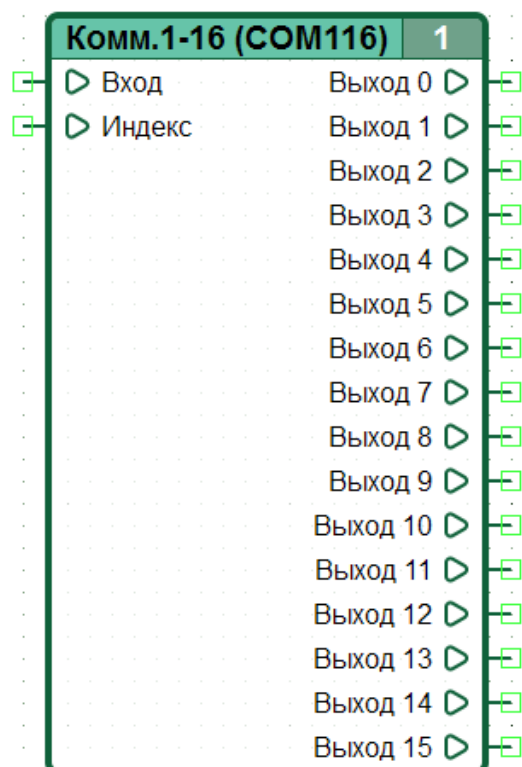
Пример 2





# 60 Коммутатор 1-в-16

## Внешний вид



## Символьное обозначение

COM116

### Группа

Коммутаторы

### Входы

Вход: числовой

Индекс: числовой

### Выходы

Выход 0...15: числовой

### Версия CANNY Lab

с 0.6b

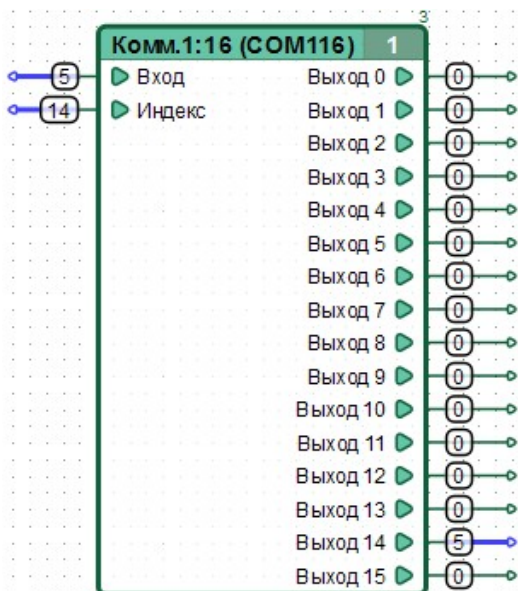


Устанавливает на выходе, определяемом значением входа «Индекс» значение равное значению входа «Вход», на остальных выходах устанавливает нулевые значения.

## 60.1 Особенности

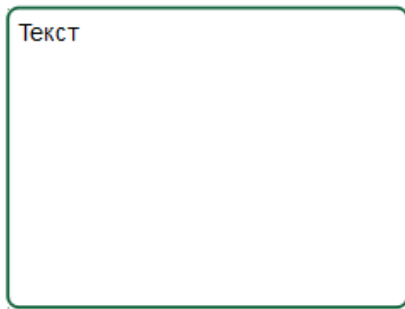
Если значение входа «Индекс» превышает число доступных выходов блока, на всех выходах устанавливаются нулевые значения.

## 60.2 Примеры



# 61 Надпись

## Внешний вид



## Символьное обозначение

A

## Группа

Элементы оформления

**Версия CANNY Lab**

с 1.15



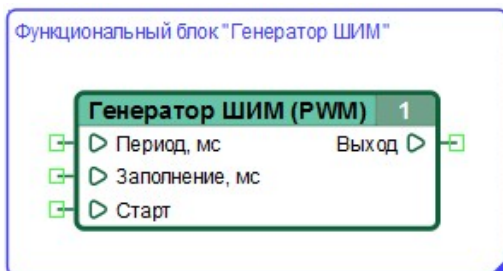
Специальный элемент функциональной диаграммы, позволяющий добавлять комментарии к тем или иным участкам диаграмм, а также сохранять любую текстовую информацию непосредственно в файле диаграммы.

## 61.1 Особенности

Элементы «Надпись» имеют индивидуальные настройки стиля отображения.

## 61.2 Примеры

Пример использования элемента "Надпись"



Окно настроек элемента "Надпись"

