

2016



СПК

Визуализация

Руководство для начинающих и продвинутых
пользователей

Версия: 1.1
Дата: 12.12.16



Оглавление

1. Цель документа	7
2. Глоссарий	8
3. Добавление визуализации в проект	9
3.1. Добавление экрана визуализации в проект	9
3.2. Свойства экрана визуализации	11
3.3. Добавление диалога в проект	13
4. Менеджер визуализации.....	14
4.1. Настройки Менеджера визуализации.....	14
4.1.1. Вкладка «Установки».....	14
4.1.2. Вкладка «Горячие клавиши по умолчанию».....	16
4.1.3. Вкладка «Визуализации».....	16
4.1.4. Вкладка «Управление пользователями».....	17
4.1.5. Вкладка «Установки списка текстов».....	18
4.2. Типы визуализаций	19
4.2.1. Настройки Target-визуализации.....	20
4.2.2. Настройки Web-визуализации	21
4.2.3. Настройки RemoteTarget-визуализации.....	22
4.2.4. Настройки HMI-визуализации	22
5. Стили и профили визуализации	25
5.1. Стили визуализации	25
5.2. Профили визуализации.....	27
6. Редактор визуализации	29
6.1. Интерфейс редактора визуализации.....	29
6.2. Добавление элемента на экран визуализации. Взаимодействие с элементами	30
6.3. Команды редактора визуализации.....	32
6.4. Вспомогательные редакторы экранов визуализации.....	36
6.4.1. Редактор интерфейсов.....	36
6.4.2. Редактор горячих клавиш	37
6.4.3. Редактор списка визуальных элементов	37
6.5. Включение и отключение сетки	38
7. Вспомогательные компоненты визуализации.....	39
7.1. Пул изображений	39

7.2. Список текстов	41
8. Описание графических примитивов	43
8.1. Структура главы	43
8.2. Базовая группа элементов	44
8.2.1. Прямоугольник, Скругленный прямоугольник, Эллипс. Типичные параметры графических примитивов	44
8.2.2. Линия	50
8.2.3. Полигон, Ломаная, Кривая Безье	51
8.2.4. Секторная диаграмма	53
8.2.5. Изображение	54
8.2.6. Фрейм элемента	56
8.3. Стандартная группа элементов	58
8.3.1. Метка	58
8.3.2. Комбинированное окно (Целочисленное)	59
8.3.3. Комбинированное окно (Массив)	61
8.3.4. Набор вкладок	63
8.3.5. Кнопка	65
8.3.6. Группа	67
8.3.7. Таблица	68
8.3.8. Текстовое поле	71
8.3.9. Полоса прокрутки	72
8.3.10. Бегунок	74
8.3.11. Управление вращением	76
8.3.12. Невидимый вход	77
8.3.13. Радио-кнопка	78
8.3.14. Кнопка-флажок	81
8.3.15. Индикатор выполнения	82
8.4. Элементы управления измерением	83
8.4.1. Отображение линейки	83
8.4.2. Стрелочный индикатор (90°/180°/360°)	87
8.4.3. Потенциометр	92
8.4.4. Гистограмма	93
8.5. Индикаторы/Переключатели/Изображения	97
8.5.1. Переключатель изображения	97
8.5.2. Индикатор	100

8.5.3. Переключатели/Выключатели	102
8.6. Специальные элементы управления	105
8.6.1. Трассировка	105
8.6.2. Тренд	113
8.6.3. Легенда.....	118
8.6.4. Элемент ActiveX	119
8.6.5. WebBrowser.....	120
8.6.6. Символ ожидания Кубик/Цветок	122
8.6.7. Текстовый редактор	123
8.6.8. Путь 3D.....	126
8.7. Элементы управления датой/временем.....	127
8.7.1. Селектор диапазона дат	127
8.7.2. Селектор времени	129
8.8. Менеджер тревог	130
8.8.1. Компонент Конфигуратор тревог.....	130
8.8.2. Таблица тревог.....	138
8.8.3. Баннер тревог	141
9. Отображение значений переменных. Форматы вывода и их спецификаторы	142
9.1. Отображение значений переменных	142
9.2. Спецификаторы формата вывода переменных.....	143
9.3. Управляющие последовательности для STRING переменных	145
9.4. Системное время в визуализации	146
9.5. Пример отображения значения переменной.....	147
10. Привязка действий к элементам визуализации	148
10.1. Вкладка InputConfiguration	148
10.2. Действия для OnMouse<Something>.....	151
10.2.1. Управление пользователями	152
10.2.2. Закрывать диалог	154
10.2.3. Открыть диалог.....	155
10.2.4. Изменить язык	156
10.2.5. Изменить отображаемую визуализацию	157
10.2.6. Выполнить команду	158
10.2.7. Переключить визуализацию в фрейме.....	160
10.2.8. Записать переменную	161
10.2.9. Выполнить ST-код.....	163

10.2.10. Переключить переменную	164
11. Примеры.....	165
11.1. Структура главы. Запуск примеров на виртуальном контроллере	165
11.2. Примеры работы с графическими примитивами	167
11.2.1. Типичные параметры графических примитивов	167
11.2.2. Фрейм	206
11.2.3. Комбинированное окно	215
11.2.4. Кнопка.....	222
11.2.5. Таблица.....	230
11.2.6. Отображение линейки и Потенциометр	244
11.2.7. Гистограмма	250
11.2.8. Индикаторы/Переключатели/Изображения	257
11.2.9. Трассировка	263
11.2.10. Тренд	272
11.2.11. WebBrowser.....	285
11.2.12. Текстовый редактор	292
11.2.13. Таблица тревог.....	315
11.3. Примеры решения типовых задач	334
11.3.1. Переключение экранов в проекте	334
11.3.2. Использование диалогов.....	344
11.3.3. Использование интерфейса фрейма	353
11.3.4. Создание анимации (элемент Изображение)	363
11.3.5. Создание мультязычного проекта	371
11.3.6. Тиражирование элементов (Multiply Visu Elements).....	383
11.3.7. Управление пользователями. Парольный доступ.....	386
11.3.8. Менеджер рецептов	397
11.3.9. Конвертация значений (Unit Conversion).....	421
11.4. Примеры, созданные по запросам пользователям	430
11.4.1. CODESYS HMI	430
11.4.2. Работа с динамическими точками (элемент Ломаная).....	445
11.4.3. Считывание координат курсора.....	452
11.4.4. Многопользовательский доступ. Независимая обработка клиентов визуализации	461
11.4.5. Подсчет количества клиентов web-визуализации.....	466
11.4.6. Обработка нажатий аппаратных кнопок	471
12. Визуализация на СПК: ограничения и рекомендации	474

12.1. Размер экранов визуализации	474
12.2. Отображение кириллического текста в визуализации	474
12.3. Тип ввода по умолчанию	475
12.4. Отсутствие поддержки элементов ActiveX и Путь3D.....	476
12.5. Поддержка трендов	476
12.6. Изображения	476
12.7. Градиенты	476
12.8. Время цикла визуализации	477
12.9. Поддерживаемые шрифты.....	478
13. Список изменений в новых сервис-паках (SP) CODESYS.....	482
13.1. CODESYS 3.5 SP7	482
13.1.1. Внешний вид Панели инструментов.....	482
13.1.2. Библиотека VisuSymbols.....	483
13.1.3. Дополнительные оси для трендов.....	484
13.1.4. Экспорт/импорт групп тревог.....	485
13.1.5. Параметр Text Format	486
13.1.6. Изменения в локализации.....	486
13.2. CODESYS 3.5 SP8	487
13.2.1. Графический примитив Часы.....	487
13.2.2. Толщина линий Трассировок/Трендов.....	487
13.2.3. Таблица тревог.....	488
13.2.4. Диалог ввода NumpadExtended.....	489
13.3. CODESYS 3.5 SP9	490
13.3.1. Графический примитив Календарь.....	490
13.3.2. Графический примитив Установка даты-времени.....	491
13.3.3. Отображение текста перечислений.....	492
13.3.4. Выбор позиции и выделение в диалогах ввода	493
13.3.5. Параметр Use real values.....	494
13.3.6. Локализация названий месяцев и дней.....	495
13.3.7. Текстовое поле.....	496

1. Цель документа

Данный документ представляет собой руководство по созданию экранов визуализации для панельных контроллеров [СПК1хх](#) и [СПК2хх](#) в среде **CODESYS V3.5** с подробным описанием характеристик и настроек всех графических примитивов, а также примерами работы с ними.

Документ может быть интересен как начинающим, так и опытным пользователям; подразумевается, что читатель обладает базовыми навыками работы со средой **CODESYS**.

При написании документа использовалась среда программирования **CODESYS V3.5 SP6 Patch 2**, в более ранних или поздних версиях некоторые компоненты могут отсутствовать или отличаться от приведенных на скриншотах. В [п. 13](#) описаны те изменения (касающиеся визуализации), которые внесли в среду программирования следующие сервис-паки (**SP7** и далее).

Документ рассчитан на использование профессионального режима настроек среды **CODESYS**. Изменить режим можно в меню **Инструменты**, вкладка **Опции**, раздел **Свойства**, кнопка **Заданные наборы свойств** (в версии **3.5 SP7** и выше настройка режима не требуется):

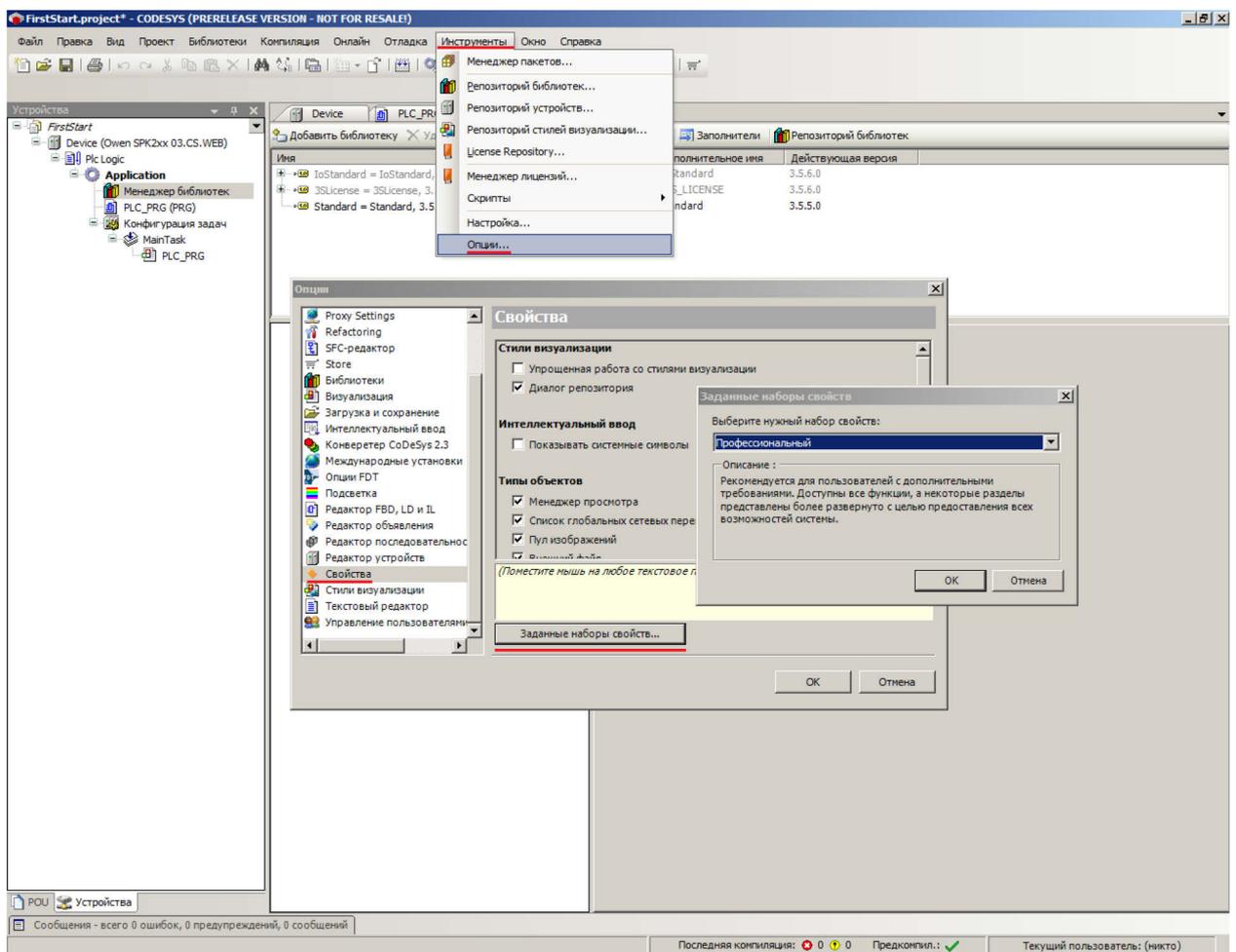


Рис. 1.1. Изменение режима настроек среды **CODESYS**

2. Глоссарий

Целевое устройство визуализации – устройство, на котором запускается и обрабатывается (но не обязательно отображается) визуализация. Обычно под целевым устройством подразумевается контроллер;

Клиент визуализации – устройство, на котором отображается (но не обязательно запускается и обрабатывается) визуализация. Таким устройством, например, может являться сенсорный панельный контроллер, панель оператора, web-браузер или ПК с установленным приложением **CODESYS HMI**;

Визуализация – совокупность экранов визуализации, представляющая собой графический интерфейс проекта **CODESYS**;

Экран визуализации – структурная единица визуализации, представляющая собой обособленную область отображения графических элементов;

Диалог – вспомогательный экран визуализации, открываемый поверх основного. Таким экраном, например, может являться диалог ввода пароля;

Фрейм – вспомогательный экран визуализации, открываемый в плоскости основного экрана визуализации с помощью [графического примитива](#). Переключая фреймы с помощью кнопок (или из кода программы), можно в реальном времени изменять целые фрагменты экрана визуализации;

Менеджер визуализации – компонент **CODESYS**, определяющий общие настройки для всех экранов визуализации в данном проекте;

Редактор визуализации – редактор **CODESYS**, используемый для создания экранов визуализации;

Графический примитив – готовый графический объект с заданным набором параметров;

Элемент – экземпляр графического примитива с конкретными значениями параметров;

ПК – персональный компьютер;

ЛКМ/ПКМ – левая/правая кнопка мыши.

3. Добавление визуализации в проект

3.1. Добавление экрана визуализации в проект

Для добавления в проект экрана визуализации необходимо нажать **ПКМ** на компонент **Application** в **Панели устройств**, раскрыть вкладку **Добавить объект** и выбрать компонент **Визуализация**.

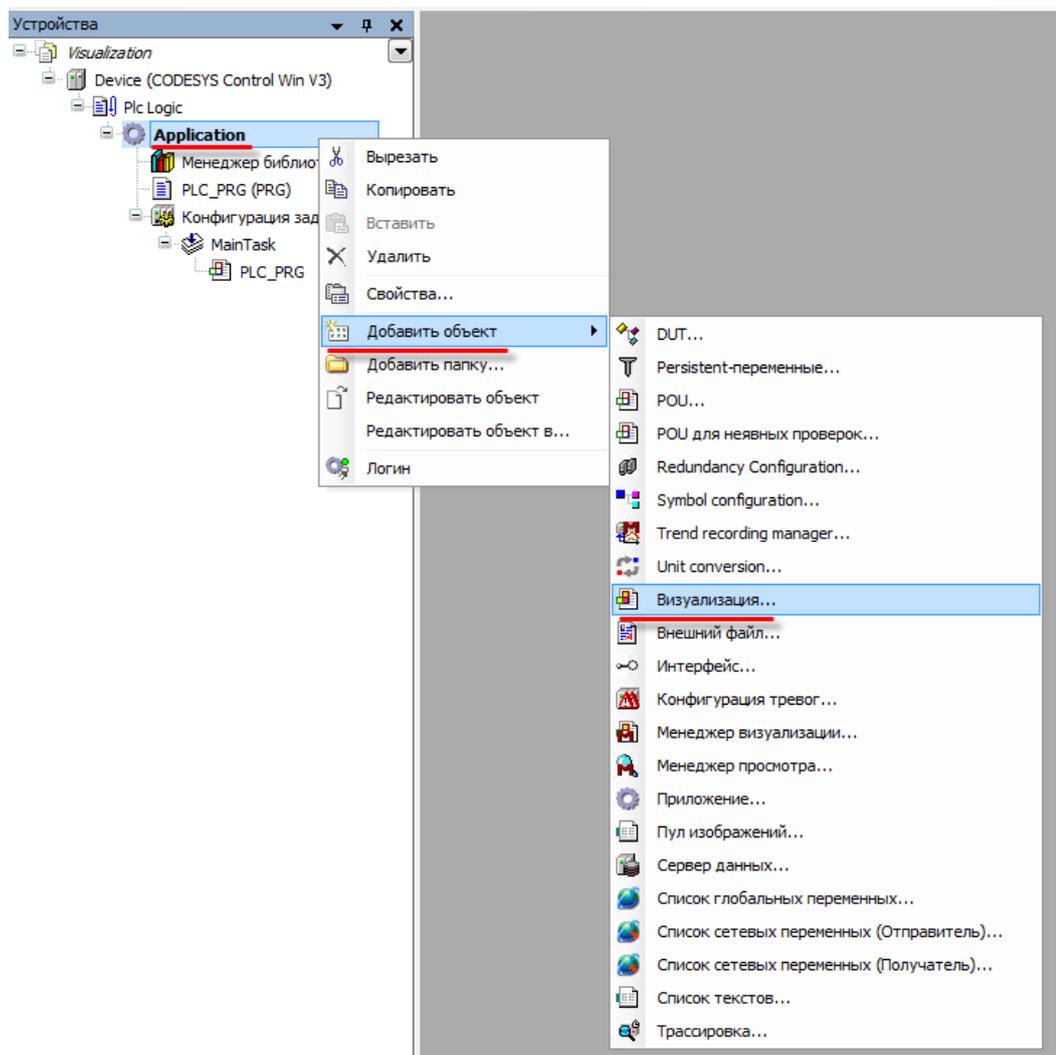


Рис. 3.1. Добавление экрана визуализации в проект

При создании экрана визуализации необходимо указать его имя. Имя экрана визуализации должно отвечать стандартным требованиям к именам в **CODESYS** – т.е. не содержать пробелов, кириллических и системных символов, не начинаться с цифр, не содержать более одного символа подчеркивания подряд:

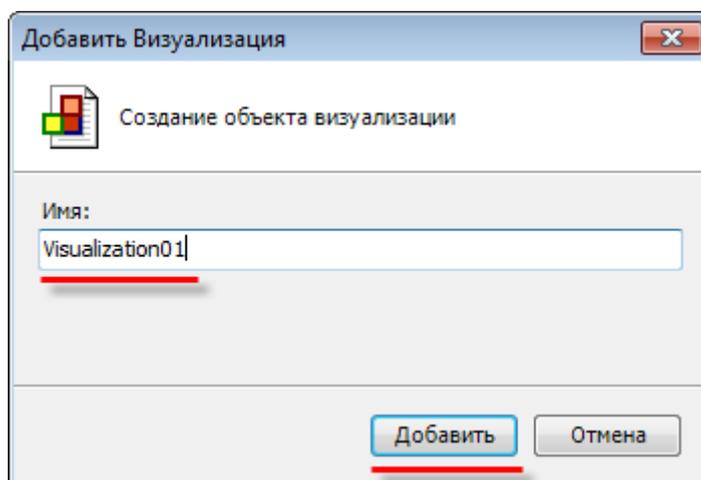


Рис. 3.2. Ввод имени экрана визуализации

Одновременно с созданием в проекте первого экрана визуализации, будет автоматически добавлен компонент [Менеджер визуализации](#), содержащий подкомпоненты [Таргет-визуализация](#) и [WEB-визуализация](#).

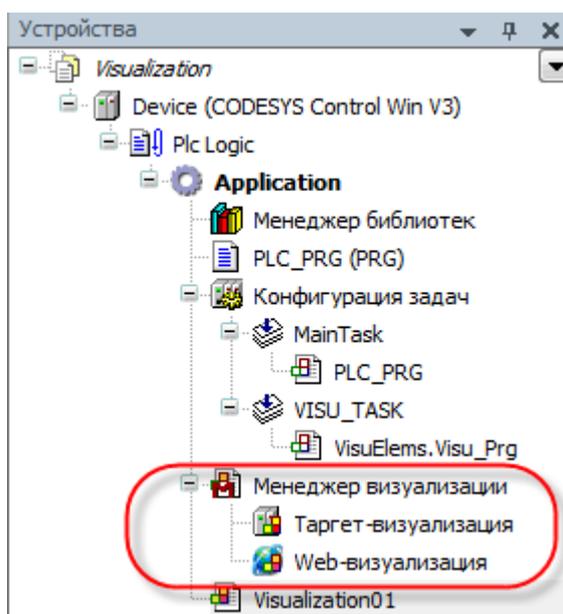


Рис. 3.3. Компонент **Менеджер визуализации** в **Панели устройств** проекта

3.2. Свойства экрана визуализации

Для того чтобы настроить тип и размер экрана визуализации, необходимо нажать **ПКМ** на его название в **Панели устройств** и в появившемся контекстном меню выбрать пункт **Свойства**:

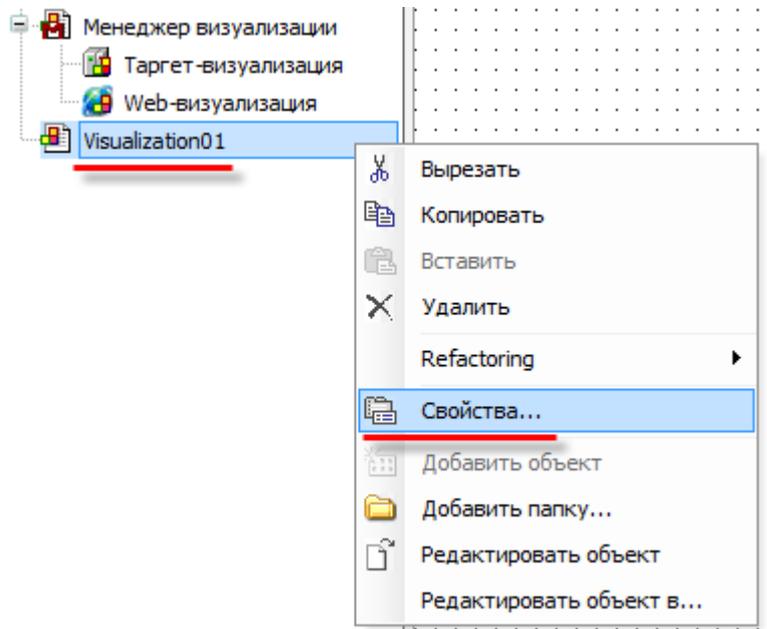


Рис. 3.4. Контекстное меню экрана визуализации

Вкладки **Общее**, **Компиляция** и **Контроль доступа** являются стандартными для всех компонентов; настройки экрана визуализации расположены на вкладке **Визуализация**:

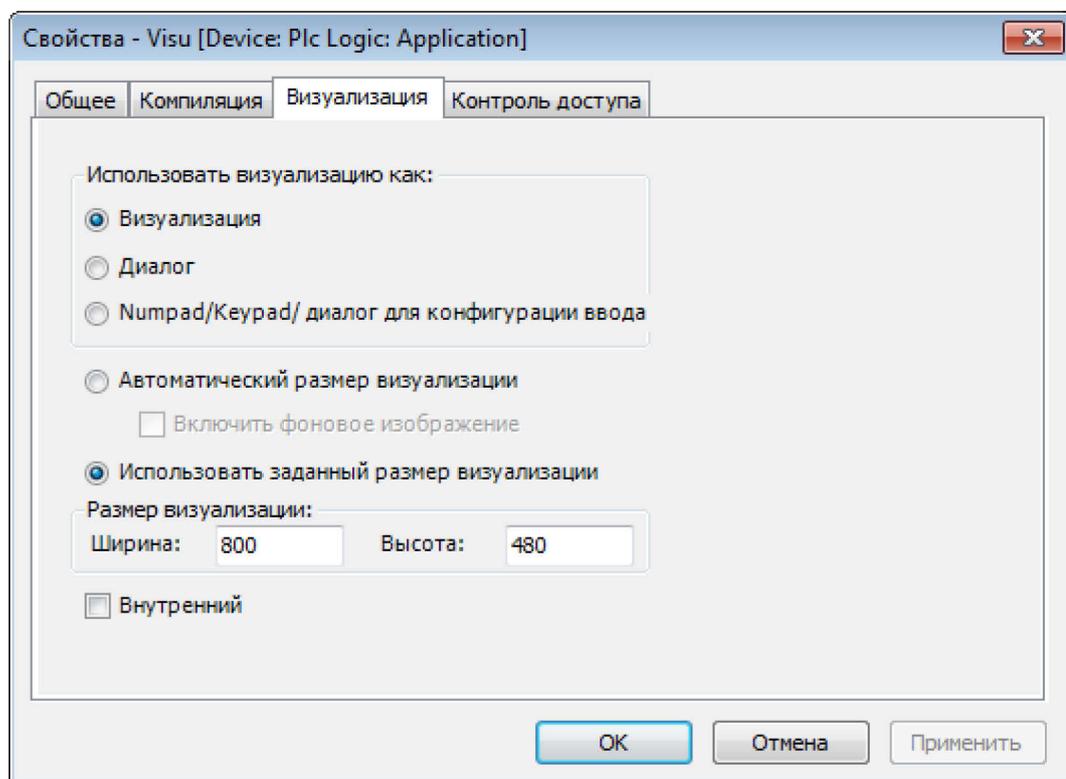


Рис. 3.5. Свойства экрана визуализации

1. Тип экрана визуализации (Использовать визуализацию как):

Визуализация – визуализация будет представлять собой стандартный экран визуализации;

Диалог – визуализация будет представлять собой диалоговое окно, которое может быть открыто с помощью элементов на экране визуализации (см. [п. 3.3](#));

Numpad/Keypad/Диалог для конфигурации ввода – визуализация будет представлять собой диалог для ввода цифровых/текстовых значений. **Обратите внимание**, что программный [интерфейс](#) такого диалога должен быть идентичен интерфейсам стандартных диалогов **Numpad/Keypad** из системной библиотеки **VisuDialogs.library** (расположенной в папке установки CODEDYS по адресу ...\\CODESYS\\Projects\\Visu\\Dialogs).

2. Размер экрана визуализации:

Автоматический размер визуализации – размер визуализации будет зависеть от размера области, занимаемой графическими элементами. Если фоновое изображение выходит за эту область, то оно будет обрезано. Для предотвращения этого необходимо поставить галочку **Включить фоновое изображение**.

Использовать заданный размер визуализации – в этом случае пользователь сам определяет ширину и высоту экрана визуализации (в пикселях). **Обратите внимание**, что в [Менеджере визуализации](#) можно указать, должен ли размер экрана визуализации подгоняться под размер экрана клиента визуализации.

Для контроллеров **СПК107**, **СПК110** и **СПК207** разрешение экрана составляет **800x480**.

Для контроллеров **СПК105** – **480x272**.

3. **Галочка Внутренний** используется для создания внутренних экранов визуализации, используемых в библиотеках. При подключении такой библиотеки к проекту, данный экран визуализации не будет доступен пользователю. Эта возможность применяется для создания вспомогательных графических материалов, поясняющих работу библиотечных функций.

3.3. Добавление диалога в проект

Частным случаем экрана визуализации является **Диалог** – вспомогательный экран визуализации, открываемый поверх основного. В качестве примера можно привести диалог ввода пароля при попытке управления элементом:

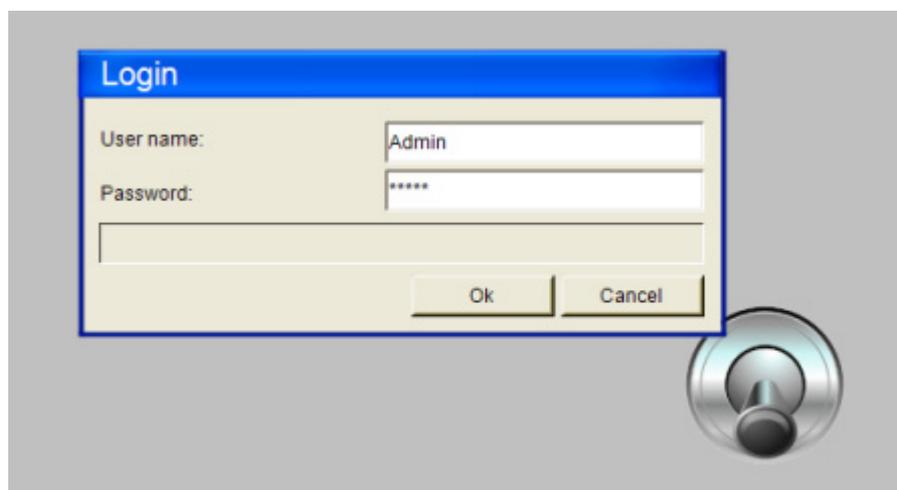


Рис. 3.6. Диалоговое окно ввода пароля

Для использования экрана визуализации в качестве диалога, необходимо в его **Свойствах** в пункте **Использовать визуализацию как** выбрать значение **Диалог** (см. [рис. 3.5](#)).

4. Менеджер визуализации

4.1. Настройки Менеджера визуализации

Компонент **Менеджер визуализации** содержит настройки, применяемые ко всем экранам визуализации в проекте. Он добавляется в проект автоматически при создании первого экрана визуализации.

4.1.1. Вкладка «Установки»

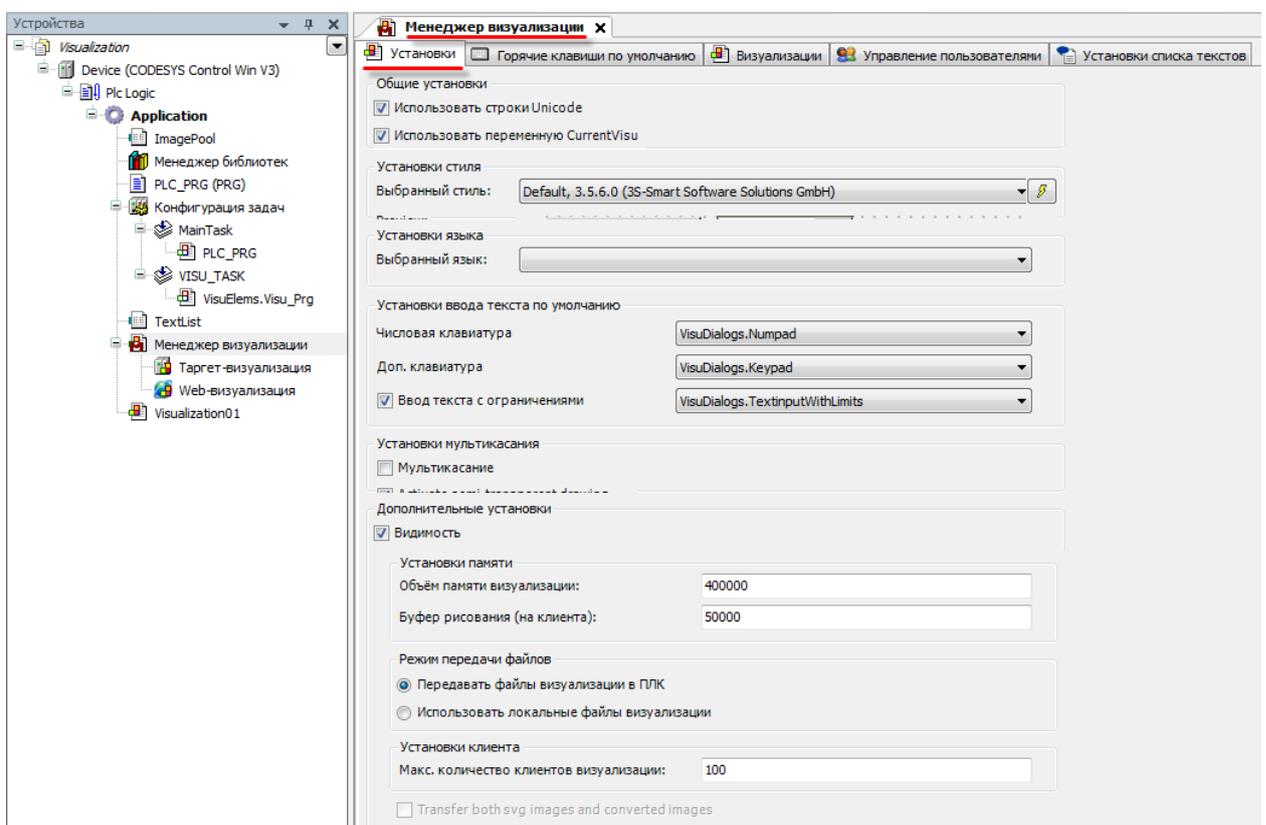


Рис. 4.1. Менеджер визуализации, вкладка **Установки**

Описание настроек приведено в табл. 1.

Табл. 1. Установки Менеджера визуализации

Общие установки	
Использовать строки Unicode	При наличии данной галочки, статический текст, отображаемый в визуализации, преобразуется в формат Юникода – это, например, позволяет отображать статические кириллические и иероглифические надписи.
Использовать переменную CurrentVisu	При наличии галочки, появляется возможность использования системной строковой переменной VisuElems.CurrentVisu , которая определяет, какой из экранов визуализации отображается в данный момент (для любого из клиентов визуализации). Соответственно, записывая в нее названия экранов визуализации, можно осуществлять переключения между ними. Пример использования приведен в п. 11.3.1 .
Установки стиля	
Выбранный стиль	Позволяет указать стиль визуализации (см. п. 5.1).
Установки языка	
Выбранный язык	Позволяет указать язык динамических текстов, отображаемый по умолчанию при запуске проекта (для мультиязычных проектов).
Установки ввода текста по умолчанию	
Числовая клавиатура	Позволяет указать диалог, который будет использоваться при вводе на экране визуализации числовых значений. По умолчанию используется диалог Numpad из библиотеки VisuDialogs.library .
Доп. клавиатура	Позволяет указать диалог, который будет использоваться при вводе на экране визуализации текстов. По умолчанию используется диалог Keypad из библиотеки VisuDialogs.library .
Ввод текста с ограничениями	Позволяет активировать и указать диалог для ввода текста ограниченной длины. По умолчанию используется диалог TextInputWithLimits из библиотеки VisuDialogs.library .
Установки мультикасания	
Мультикасание	Если клиент визуализации поддерживает возможность мультикасания, то для его использования необходимо активировать эту настройку.
Дополнительные установки	
Видимость	При наличии галочки отображаются последующие настройки.
Объем памяти визуализации	Область памяти в байтах, отводимая для элементов визуализации. Значение по умолчанию - 400000 байт.
Буфер рисования	Область памяти в байтах, отводимая для действий рисования. Значение по умолчанию - 50000 байт.
Режим передачи файлов	Опция доступна только для PC-based контроллеров и позволяет использовать локальные файлы визуализации (графические файлы, файлы списков текстов и т.д.). Путь к ним указывается в Установках проекта , вкладка Визуализация .
Макс. количество клиентов	Позволяет указать максимальное количество клиентов визуализации. Значение по умолчанию - 100.

4.1.2. Вкладка «Горячие клавиши по умолчанию»

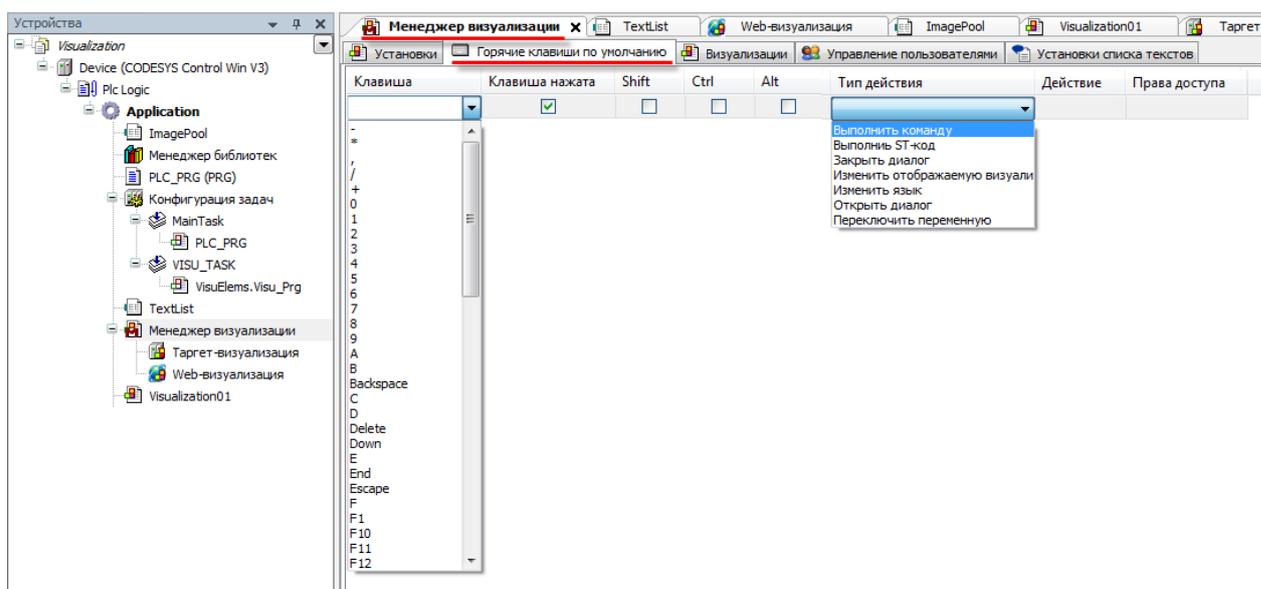


Рис. 4.2. Менеджер визуализации, вкладка **Горячие клавиши по умолчанию**

На этой вкладке можно привязать к нажатию клавиш выполнение действий. Для этого целевое устройство должно иметь соответствующие кнопки. При отсутствии галочки **Клавиша нажата** действие будет выполняться при отпускании зажатой клавиши. Можно создавать комбинации клавиш с использованием **Shift/Ctrl/Alt**. Для клавиш и их комбинаций можно настраивать права доступа (при наличии в проекте [Управления пользователями](#)) – т.е. разрешить или запретить определенным пользователям использование определенных клавиш.

4.1.3. Вкладка «Визуализации»

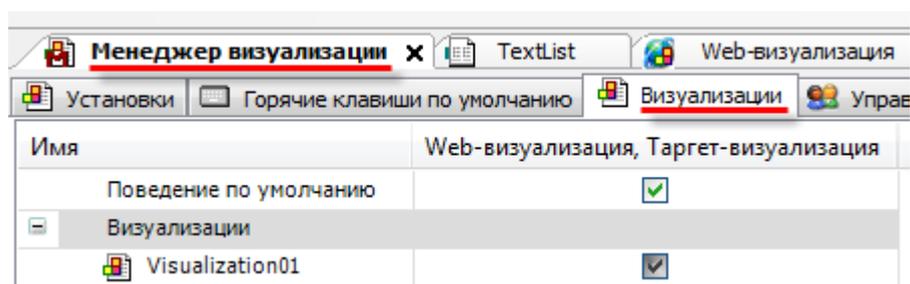


Рис. 4.3. Менеджер визуализации, вкладка **Визуализации**

На этой вкладке содержится список всех экранов визуализации для всех типов визуализаций, используемых в проекте. Галочка характеризует загрузку экрана визуализации на целевое устройство при компиляции проекта.

4.1.4. Вкладка «Управление пользователями»

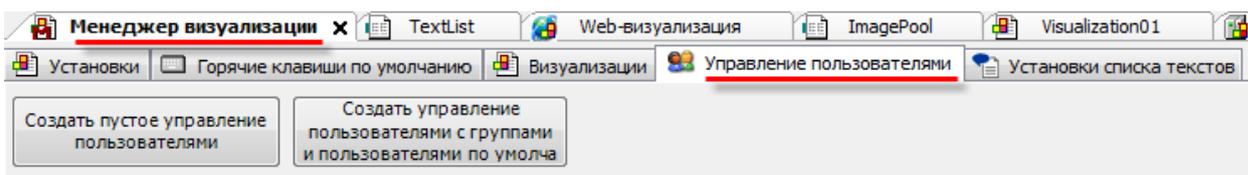


Рис. 4.4. Менеджер визуализации, вкладка **Управление пользователями**

На этой вкладке настраиваются параметры управления пользователями. По умолчанию в проекте управление пользователями отсутствует; его надо либо создать с нуля (левая кнопка), либо воспользоваться готовым шаблоном (правая кнопка).

Шаблон управления пользователям имеет следующий вид:

Имя группы	Автоматический ...	Время выхода	Разрешение изменить польз...	Описание
Admin имеет пользователя 'Admin'	<input checked="" type="checkbox"/>	1 минут	<input checked="" type="checkbox"/>	
Service имеет пользователя 'Service'	<input type="checkbox"/>	1 минут	<input type="checkbox"/>	
Operator имеет пользователя 'Operator'	<input type="checkbox"/>	1 минут	<input type="checkbox"/>	
None	<input type="checkbox"/>	1 минут	<input type="checkbox"/>	

Рис. 4.5. Менеджер визуализации, шаблон **Управления пользователями**

На вкладке **Группы** создаются группы пользователей; для них можно задать автоматический выход (**logout**) по истечению заданного времени бездействия и разрешение на изменение пользовательских данных.

На вкладке **Пользователи** создаются пользователи; для каждого из них задается логин, пароль и группа.

На вкладке **Установки** задаются параметры загрузки информации пользователей при загрузке проекта на целевое устройство (**Всегда/Никогда/Решение при каждой загрузке**), а также имеется возможность настроить переключение на стартовый экран визуализации при выходе пользователя.

Пример создания проекта с **Управлением пользователями** приведен в [п. 11.3.7](#).

4.1.5. Вкладка «Установки списка текстов»

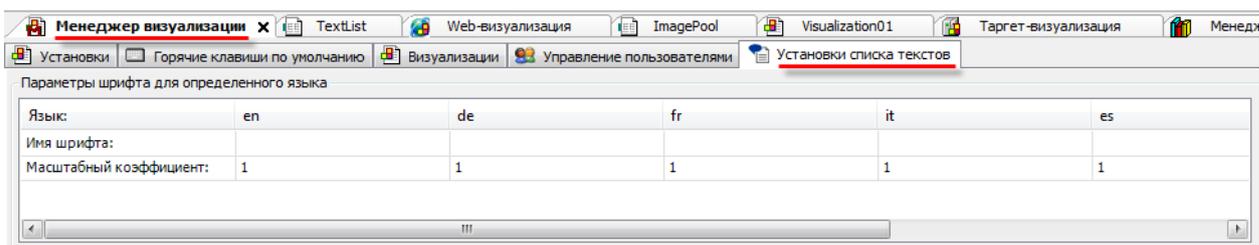


Рис. 4.6. Менеджер визуализации, вкладка **Установки списков текстов**

На этой вкладке можно настроить шрифт и масштабный коэффициент для каждого языка, используемого в визуализации (для мультязычных визуализаций). Пример создания мультязычного проекта приведен в [п. 11.3.5](#).

4.2. Типы визуализаций

Менеджер визуализации определяет типы визуализаций, используемых в проекте. По умолчанию проект содержит два типа – **Target-визуализацию** и **Web-визуализацию**. Пользователь также может добавить **RemoteTarget-визуализацию** и **HMI-визуализацию**:

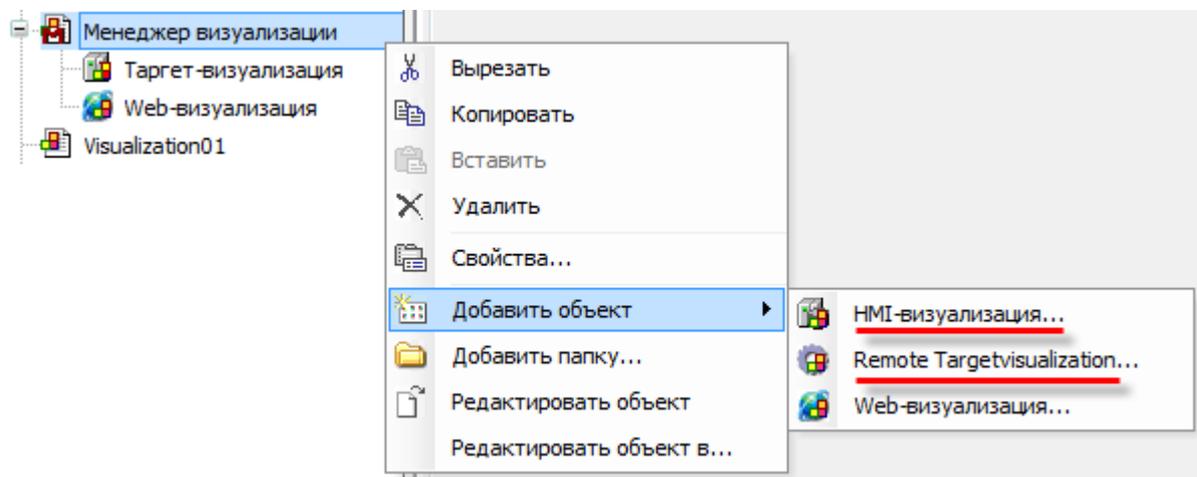


Рис. 4.7. Добавление новых типов визуализации в проект

Target-визуализация – это визуализация, отображаемая на целевом устройстве (например, панельном контроллере). Каждый проект может содержать только одну target-визуализацию. Настройки target-визуализации описаны в [п. 4.2.1](#).

Web-визуализация – это визуализация, транслируемая на заданный IP-адрес. Она может быть открыта на любом устройстве с браузером, который поддерживает **HTML5**. Для запуска web-визуализации, целевое устройство должно содержать web-сервер. Количество web-визуализаций в проекте не ограничено. Настройки web-визуализации описаны в [п. 4.2.2](#).

RemoteTarget-визуализация – это визуализация, отображаемая на дисплее, подключаемом к целевому устройству (например, на панели оператора, подключенной к контроллеру), при этом создание и обработка визуализации происходит на самом целевом устройстве. Оба устройства должны поддерживать работу соответствующего компонента. Каждый проект может содержать только одну RemoteTarget-визуализацию.

HMI-визуализация – это визуализация, транслируемая с целевого устройства на компьютер (APM) с установленным приложением **CODESYS HMI**. На целевом устройстве реализуется получение, обработка и передача данных, а в **CODESYS HMI** – их отображение и управление технологическим процессом. Количество HMI-визуализаций в проекте не ограничено. Настройки HMI-визуализации описаны в [п. 4.2.4](#). Пример создания HMI-визуализации и работы с **CODESYS HMI** приведен в [п. 11.3.10](#).

Также существует **сервисная визуализация**, отображаемая в **Редакторе визуализации** среды программирования **CODESYS** при подключении к целевому устройству с запущенным проектом.

4.2.1. Настройки Target-визуализации

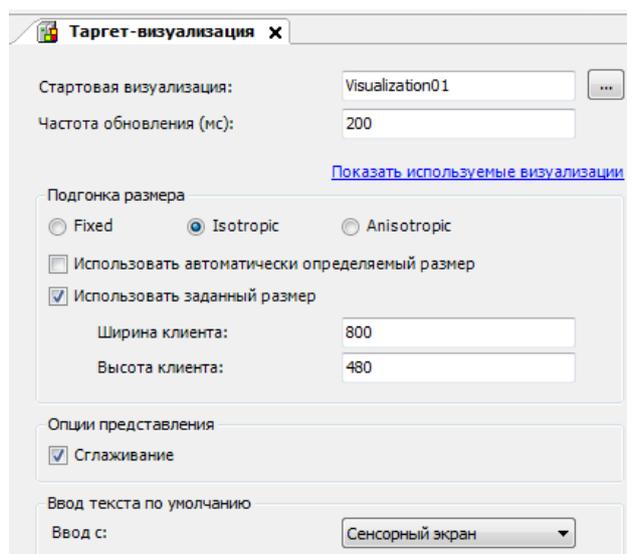


Рис. 4.8. Настройки target-визуализации

Стартовая визуализация – позволяет выбрать экран визуализации, который будет отображен на целевом устройстве после запуска проекта.

Частота обновления – позволяет настроить частоту обновления экранов визуализации целевого устройства. Значение задается в миллисекундах.

Подгонка размера – позволяет выбрать режим масштабирования экранов визуализации на целевом устройстве:

Fixed – размер экрана визуализации на целевом устройстве будет соответствовать оригинальному размеру экрану визуализации (указанному в [Свойствах](#) экрана визуализации);

Isotropic – позволяет задавать размер визуализации в пикселях, при этом в случае несоответствия размеров экрана визуализации и дисплея целевого устройства, экран визуализации будет *масштабироваться с сохранением* соотношения сторон;

Anisotropic – позволяет задавать размер визуализации в пикселях, при этом в случае несоответствия размеров экрана визуализации и дисплея целевого устройства, экран визуализации будет *масштабироваться без сохранения* соотношения сторон.

Сглаживание – позволяет включить сглаживание элементов. Это улучшает их внешний вид, но может привести к падению производительности и нежелательным графическим артефактам (например, у всплывающих подсказок и невидимых элементов).

Ввод текста по умолчанию – позволяет определить основное устройство ввода для целевого устройства – сенсорный экран или клавиатуру.

4.2.2. Настройки Web-визуализации

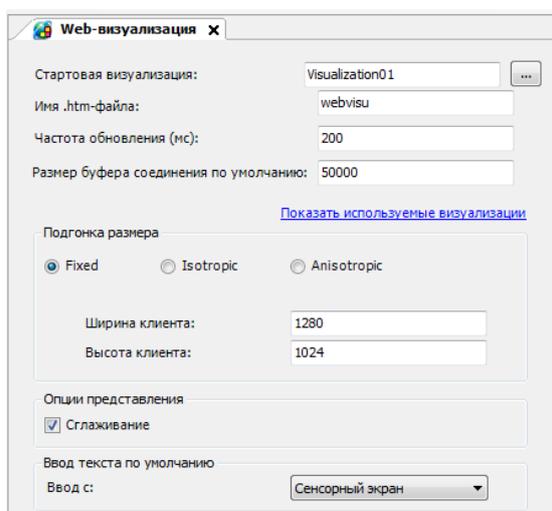


Рис. 4.9. Настройки web-визуализации

Стартовая визуализация – позволяет выбрать экран визуализации, который будет отображен на клиенте web-визуализации при открытии данной веб-страницы.

Имя .htm-файла – название файла, которое указывается в web-адресе визуализации. По умолчанию имя файла – **webvisu**, и адрес web-визуализации соответственно

http://<IP-адрес целевого устройства>:8080/webvisu.htm

Частота обновления – позволить настроить частоту обновления экранов визуализации на клиенте web-визуализации. Значение задается в миллисекундах.

Размер буфера соединения по умолчанию – максимальный размер буфера данных, передаваемых клиенту web-визуализации.

Подгонка размера – позволяет выбрать режим масштабирования экранов визуализации на клиенте web-визуализации:

Fixed – позволяет задавать фиксированный размер визуализации в пикселях;

Isotropic – экран визуализации будет **масштабироваться** до размеров дисплея клиента web-визуализации **с сохранением** соотношения сторон;

Anisotropic – экран визуализации будет **масштабироваться** до размеров дисплея клиента web-визуализации **без сохранения** соотношения сторон.

Сглаживание – позволяет включить сглаживание элементов. Это улучшает их внешний вид, но может привести к падению производительности и нежелательным графическим артефактам (например, у всплывающих подсказок и невидимых элементов).

Ввод текста по умолчанию – позволяет определить основное устройство ввода для клиента web-визуализации – сенсорный экран или клавиатуру.

4.2.3. Настройки RemoteTarget-визуализации

RemoteTarget-визуализация не обладает настройками.

4.2.4. Настройки HMI-визуализации

1. Вкладка HMI-установки

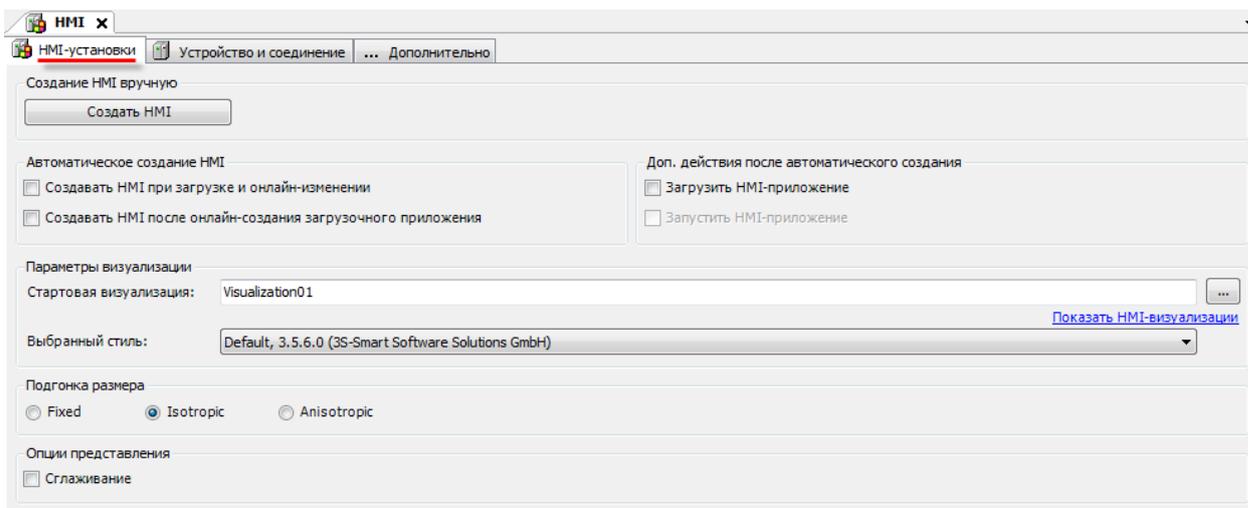


Рис. 4.10. Настройки HMI-визуализации, вкладка HMI-установки

Создать HMI – на базе проекта генерируется идентичный проект для CODESYS HMI.

Опции автоматического создания HMI:

Создавать HMI при загрузке и онлайн-изменении – при каждой загрузке/онлайн-изменении проекта на целевое устройство, HMI-приложение будет автоматически обновляться;

Создавать HMI после онлайн-создания загрузочного приложения – при каждом онлайн-создании загрузочного приложения для целевого устройства, HMI-приложение будет автоматически обновляться;

Загрузить HMI-приложение – после автоматического создания/обновления HMI-приложения, оно будет загружаться на HMI-клиент;

Запустить HMI-приложение – после автоматического создания/обновления HMI-приложения и его загрузки на HMI-клиент, оно будет автоматически запускаться.

Стартовая визуализация – позволяет выбрать экран визуализации, который будет отображен на HMI-клиенте после запуска приложения.

Выбранный стиль – позволяет указать [стиль](#) HMI-визуализации.

Подгонка размера – позволяет выбрать режим масштабирования экранов визуализации на HMI-клиенте:

Fixed – размер экрана визуализации на HMI-клиенте будет соответствовать оригинальному размеру экрану визуализации (указанному в [Свойствах](#) экрана визуализации);

Isotropic – экран визуализации будет *масштабироваться* до размеров дисплея HMI-клиента *с сохранением* соотношения сторон;

Anisotropic – экран визуализации будет *масштабироваться* до размеров дисплея HMI-клиента *без сохранения* соотношения сторон.

Сглаживание – позволяет включить сглаживание элементов. Это улучшает их внешний вид, но может привести к падению производительности

2. Вкладка **Устройство и соединение**

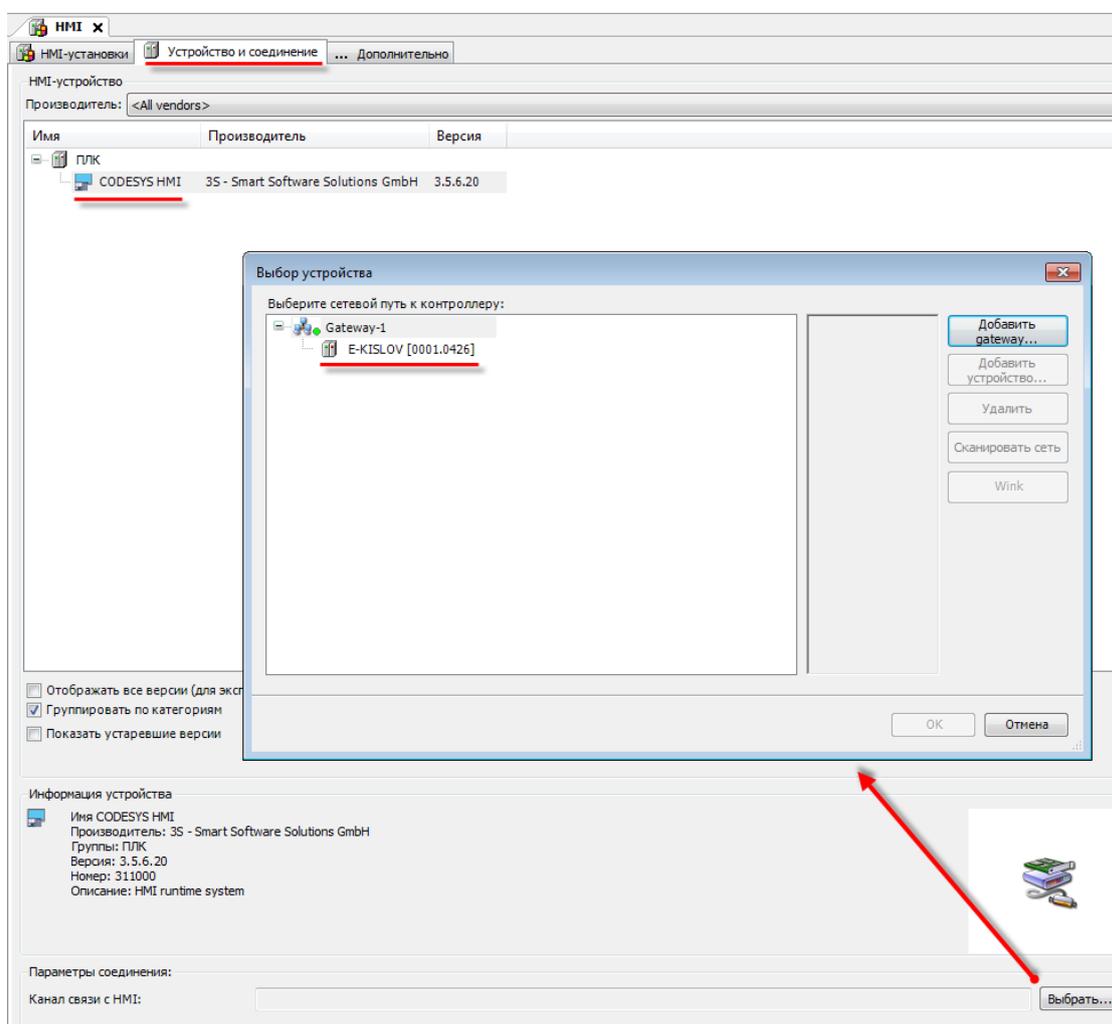


Рис. 4.11. Настройки HMI-визуализации, вкладка **Устройство и соединение**

На этой вкладке нужно выбрать версию приложения **CODESYS HMI**, которая будет использоваться для отображения визуализации, и указать целевое устройство, который будет передавать визуализацию HMI-клиенту.

3. Вкладка **Дополнительно**

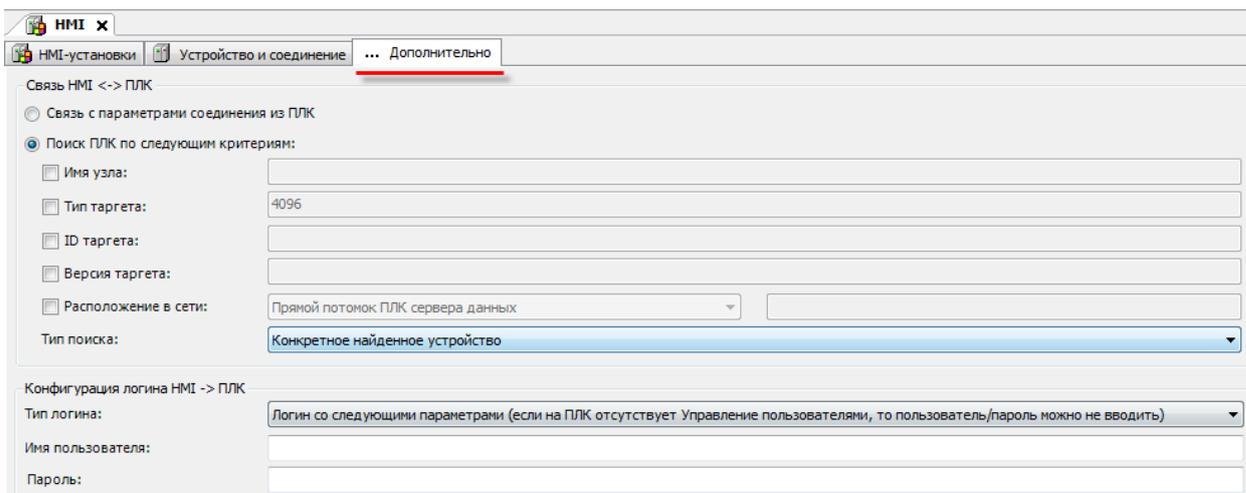


Рис. 4.12. Настройки NMI-визуализации, вкладка **Дополнительно**

На этой вкладке можно настроить параметры поиска целевого устройства, которое будет использоваться для создания NMI-визуализации и ее передачи NMI-клиенту (например, в тех случаях, когда проект создается удаленно). При выборе режима **Связь с параметрами соединения из ПЛК** используется контроллер, выбранный на вкладке **Устройство и соединение**.

Также на данной вкладке можно настроить логин/пароль для доступа к контроллеру (если в проекте контроллера используется [Управление пользователями](#)).

5. Стили и профили визуализации

5.1. Стили визуализации

Стиль визуализации определяет характеристики графических примитивов проекта (в основном, их внешний вид):



Рис. 5.1. Разница между двумя стилями визуализации

Выбрать стиль визуализации можно в установках **Менеджера визуализации**:

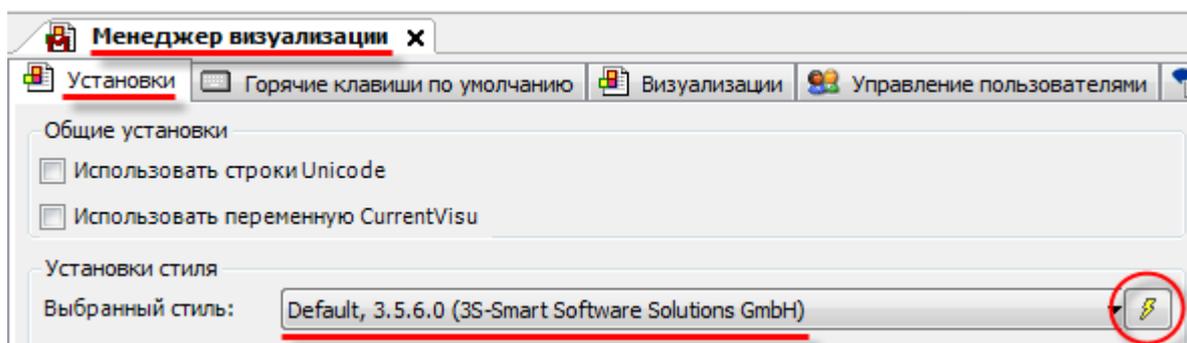


Рис. 5.2. Выбор стиля визуализации

Нажатие на пиктограмму рядом с названием стиля приводит к открытию **Редактора стилей визуализации**:

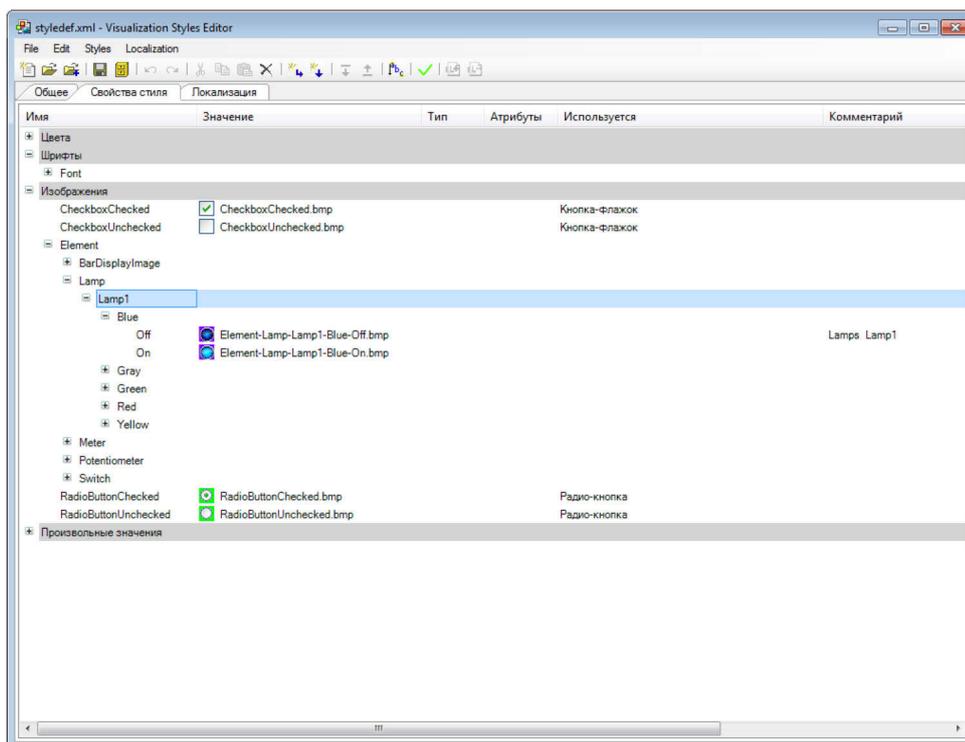


Рис. 5.3. Окно редактора стилей визуализации

Файлы стили визуализации по умолчанию хранятся по адресу

C:\ProgramData\CODESYS\Visualization Styles

в формате **.xml**.

После создания нового стиля визуализации (или редактирования существующего), его необходимо добавить в **Репозиторий стилей визуализации** с помощью нажатия соответствующей иконки:

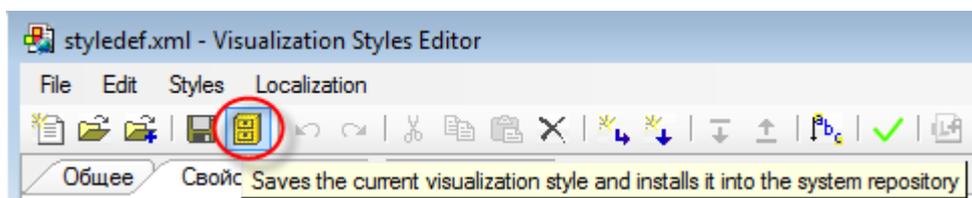


Рис. 5.4. Сохранение стиля визуализации и добавления его в редактор визуализации

Перезапустите **CODESYS**. После этого сохраненный стиль можно будет выбрать в установках **Менеджера визуализации** (см. [п. 4.1](#)).

5.2. Профили визуализации

Профиль визуализации определяет версии библиотек визуализации, используемые в проекте, что, в свою очередь, влияет на набор доступных графических примитивов и количество их настроек.

Содержание профиля можно увидеть в **Репозитории визуальных элементов** (доступен только в том случае, если в **CODESYS** не открыт проект):

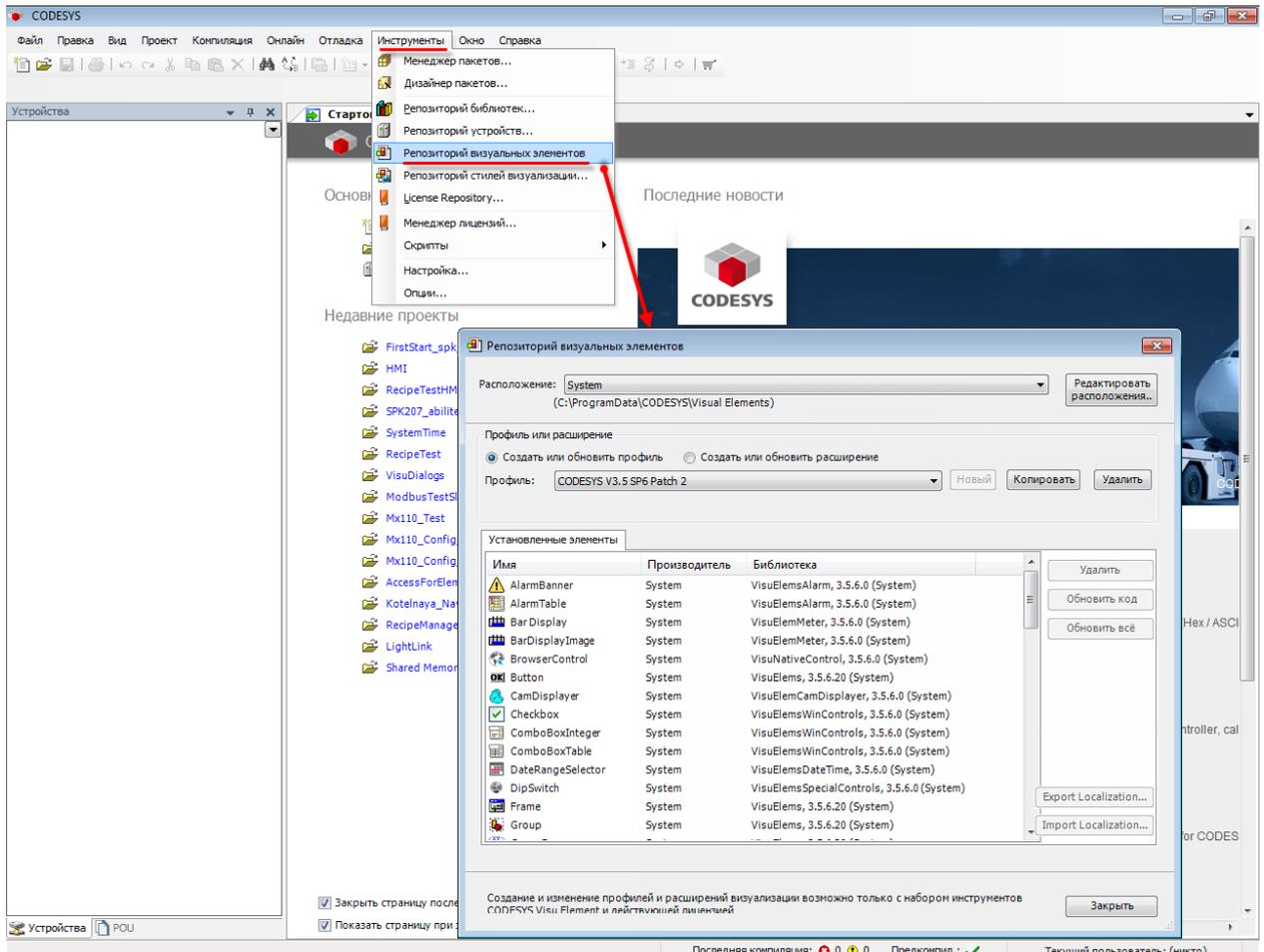


Рис. 5.5. Репозиторий визуальных элементов

Сменить профиль визуализации, используемый в проекте, можно в **Установках проекта**:

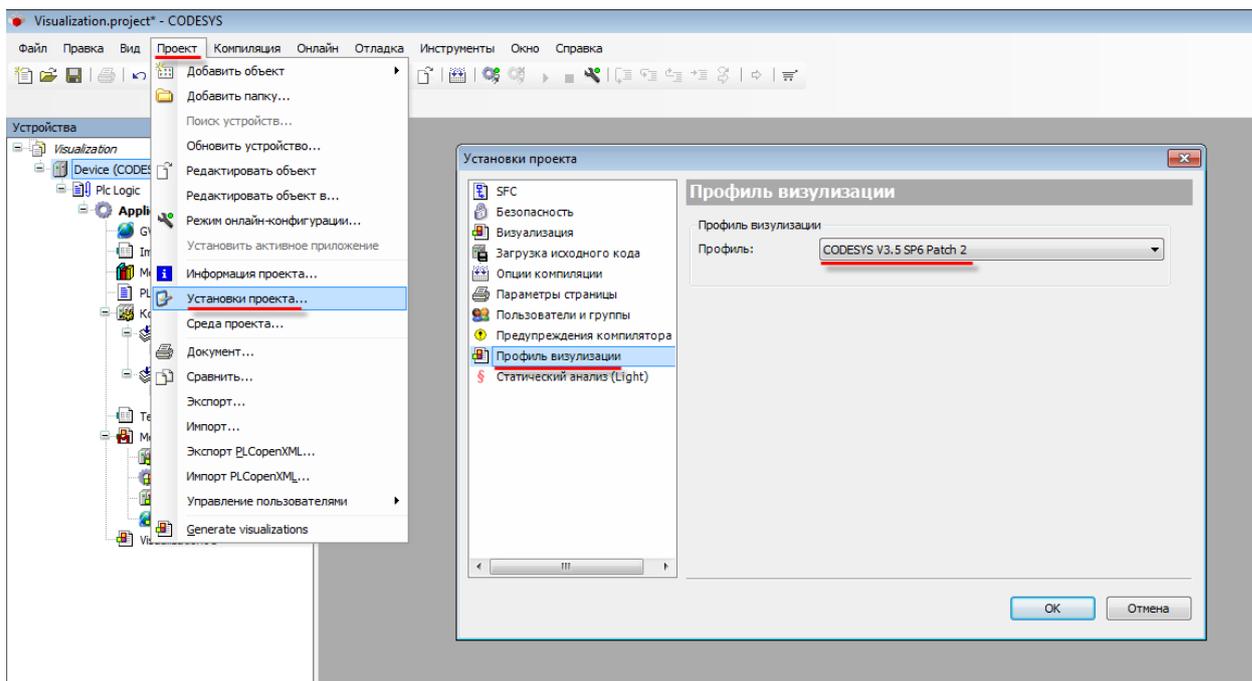


Рис. 5.6. Выбор профиля визуализации, используемого в проекте

6. Редактор визуализации

6.1. Интерфейс редактора визуализации

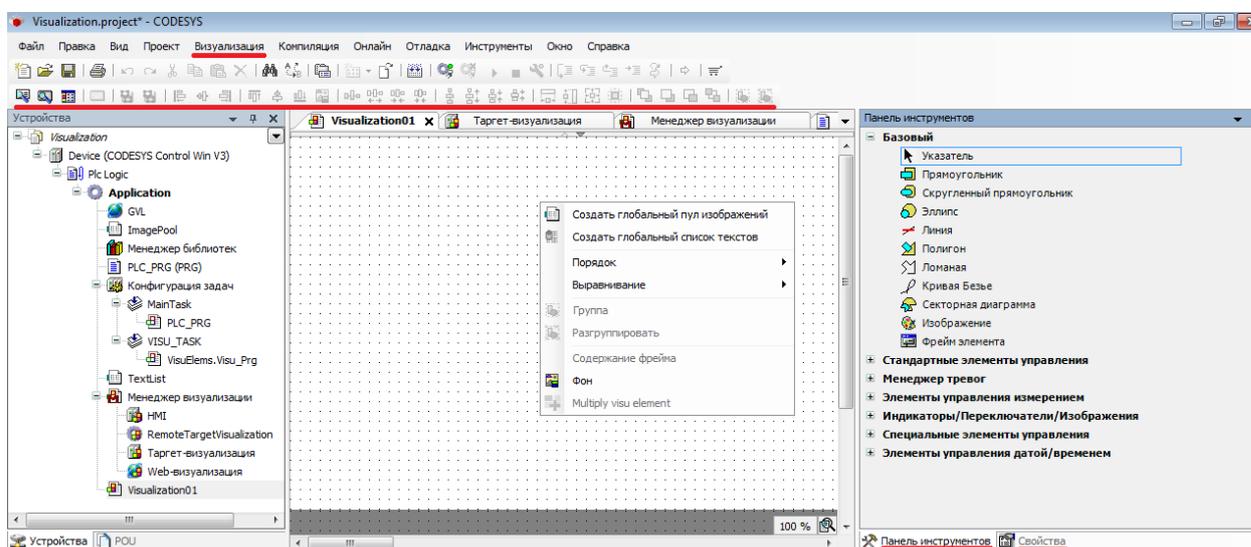


Рис. 6.1. Интерфейс редактора визуализации

Редактор визуализации открывается при выборе любого экрана визуализации проекта. Редактор включает в себя:

1. **Рабочую область**, в которой происходит создание экрана визуализации;
2. **Панель инструментов редактора**, содержащую набор графических примитивов;
3. **Панель свойств**, содержащую параметры графического элемента (открывается на месте **Панели инструментов редактора** при выборе соответствующего элемента);
4. Вспомогательные редакторы (**Редактор интерфейса**, **Редактор горячих клавиш**, **Редактор списка визуальных элементов**).

Описание графических примитивов и их параметров приведены в [п. 8](#).

Описание вспомогательных редакторов приведено в [п. 6.4](#).

Вкладка **Визуализация** в меню **CODESYS** содержит команды, которые продублированы ярлыками, расположенными чуть ниже; также эти команды присутствуют в контекстном меню **Редактора визуализации**, открываемом с помощью нажатия **ПКМ** на любое место рабочей области. Описание команд приведено в [п. 6.3](#).

6.2. Добавление элемента на экран визуализации. Взаимодействие с элементами

Для добавления элемента на экран визуализации, необходимо выбрать его на **Панели инструментов редактора**, после чего разместить на рабочем поле с помощью одиночного нажатия **ЛКМ**.

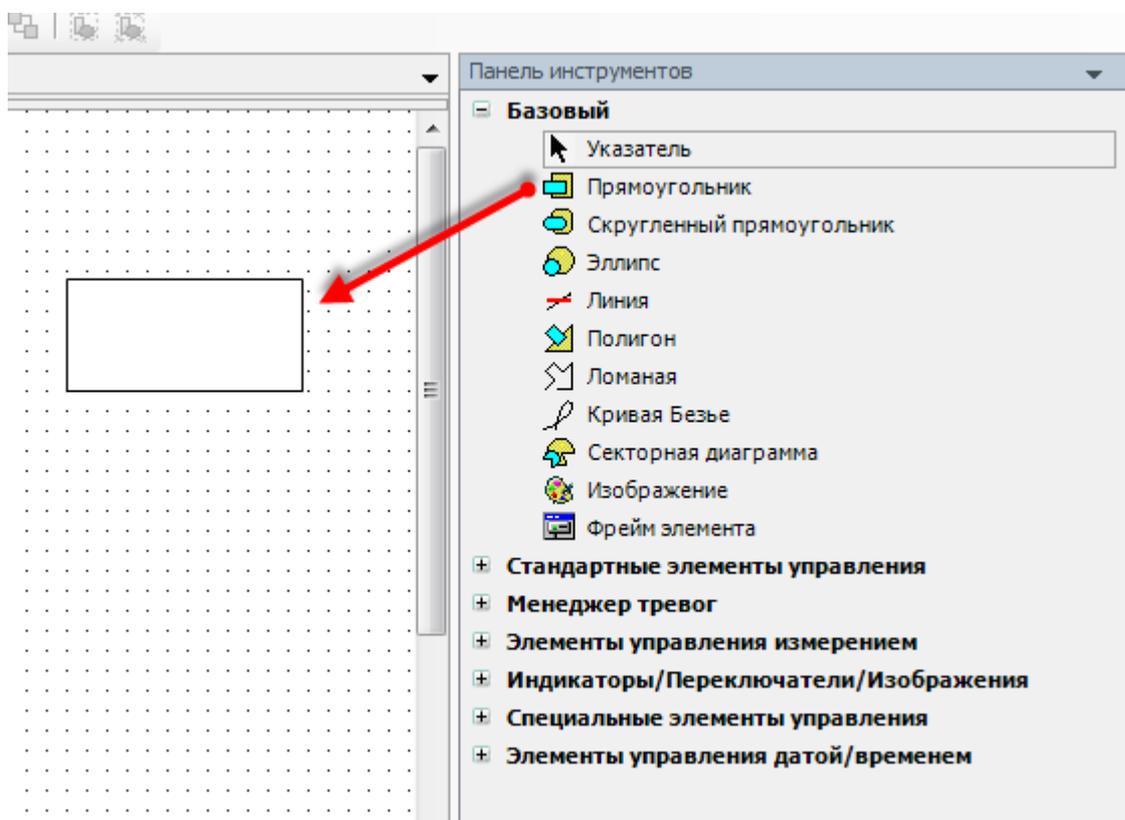


Рис. 6.2. Область рабочего поля с добавленным элементом **Прямоугольник**

Для взаимодействия с элементом, необходимо выделить его одиночным нажатием **ЛКМ** и дождаться, пока курсор примет соответствующую форму:

1. Для **перетаскивания** выделите элемент, наведите на него мышью для появления

курсора перетаскивания () и, зажав **ЛКМ**, перетащите элемент в нужное место экрана;

2. Для **изменения размера** выделите элемент, наведите мышью на одну из его

опорных точек () для появления курсора деформации () и, зажав **ЛКМ**, растяните элемент до нужных размеров;

3. Для **изменения текста элемента** наведите на текстовое поле элемента мышью

для появления курсора ввода (), после чего однократным нажатием **ЛКМ** перейдите к вводу/редактированию текста. Данная операция возможна только для элементов, которые могут отображать текст. Введенный текст попадет в глобальный список текстов (см. рис. 6.5).

4. Для **перемещения центра элемента** выделите элемент, наведите мышью на его

центр () мышью для появления курсора перетаскивания () и, зажав **ЛКМ**, перетащите центр элемента в нужное место экрана. Центр элемента не является центром в геометрическом смысле – т.е. его местоположение не зависит от расположения элемента, но вокруг этого центра, например, может происходить относительное вращение элемента.

Для выделения нескольких элементов нужно либо, зажав **ЛКМ**, обвести их рамкой, после чего отпустить кнопку мыши, либо же зажать клавишу **Shift** и последовательно нажимать на элементы **ЛКМ**. Во втором случае опорные точки элемента, выделенного первым, сохраняют синий цвет:

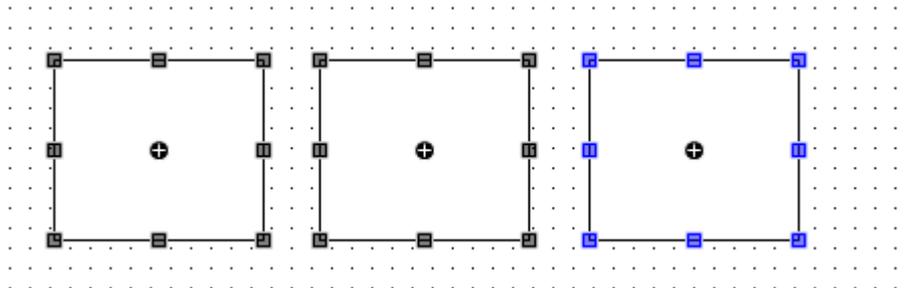


Рис. 6.3. Одновременное выделение нескольких элементов

6.3. Команды редактора визуализации

Вкладка **Визуализация** в строке меню **CODESYS** содержит команды, которые продублированы ярлыками, расположенными чуть ниже; также эти команды присутствуют в контекстном меню **Редактора визуализации**, открываемом с помощью нажатия **ПКМ** на любое место рабочей области.

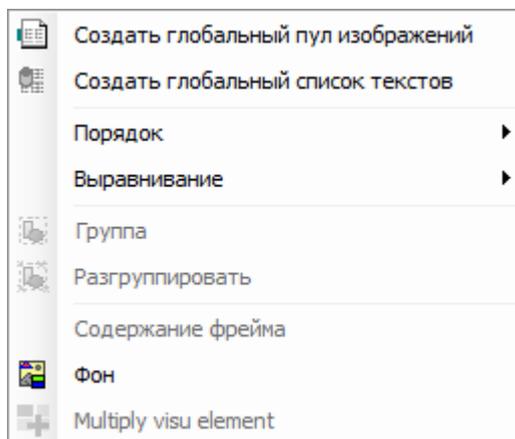


Рис. 6.4. Команды контекстного меню редактора визуализации

1. Команды **Создать глобальный пул изображений** и **Создать глобальный список текстов** создают соответствующие компоненты на **Панели POU**. В отличие от **локальных** (созданных на **Панели устройств** для конкретного приложения), действие этих компонентов распространяется на все приложения проекта.

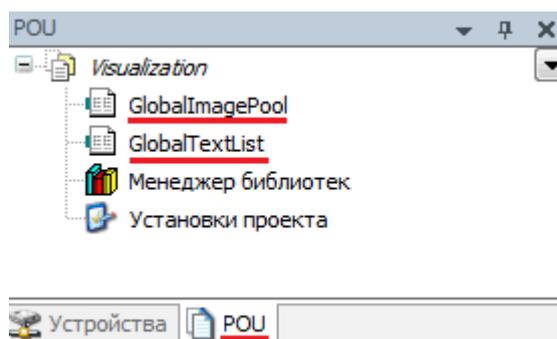


Рис. 6.5. Глобальные пул изображений и список текстов

2. Команды вкладки **Порядок** определяют размещение элементов по слоям относительно друг друга. Элемент, расположенный в верхнем слое, перекрывает элементы, расположенные в слоях ниже:

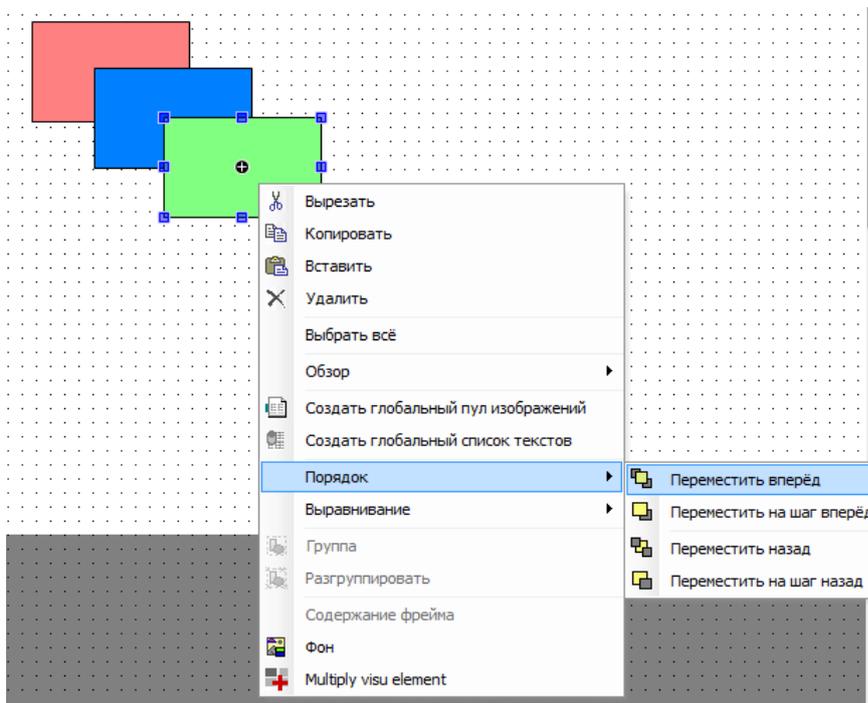


Рис. 6.6. Вкладка настройки порядка отображения элементов

3. Команды вкладки **Выравнивание** (становится доступной только при одновременно выделении нескольких элементов) позволяет выравнивать элементы, а также настраивать их отступы и междустрочные интервалы:

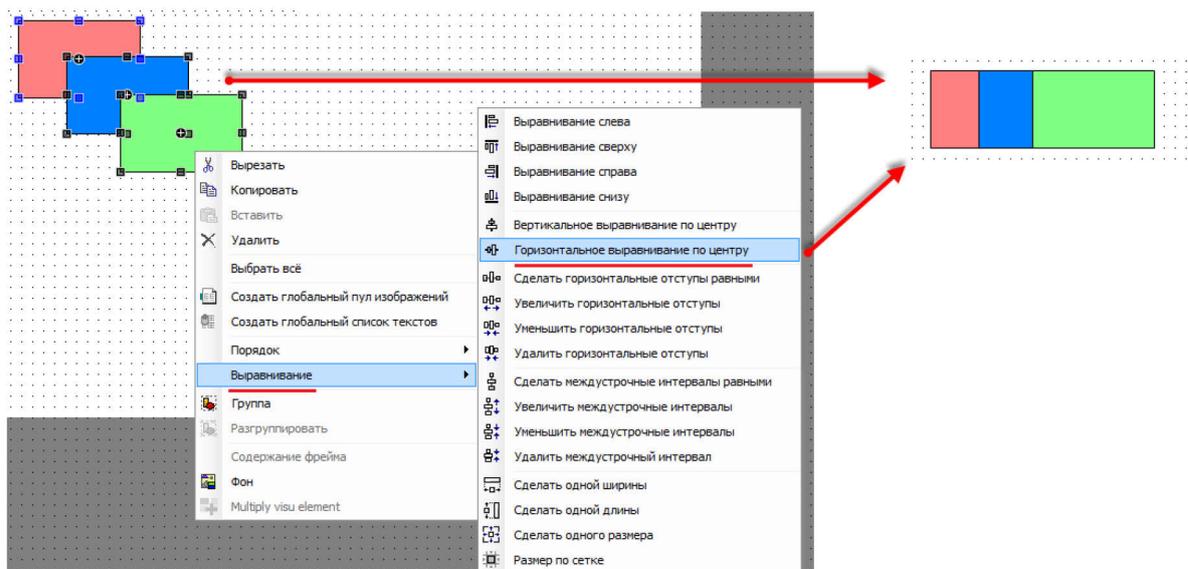


Рис. 6.7. Вкладка выравнивания элементов

4. Команда **Группа** позволяет сгруппировать элементы – т.е. все взаимодействие с ними (выделение, перемещение и т.д.) будет происходить как с единым целым. При этом сгруппированный элемент может получить новые свойства: например, элементу [Индикатор](#) нельзя присвоить [действие](#) (у него нет вкладки [InputConfiguration](#)), но сгруппировав его с любым элементом (например, невидимой точкой), можно получить такую возможность.

Команда **Разгруппировать**, соответственно, разгруппировывает элементы.

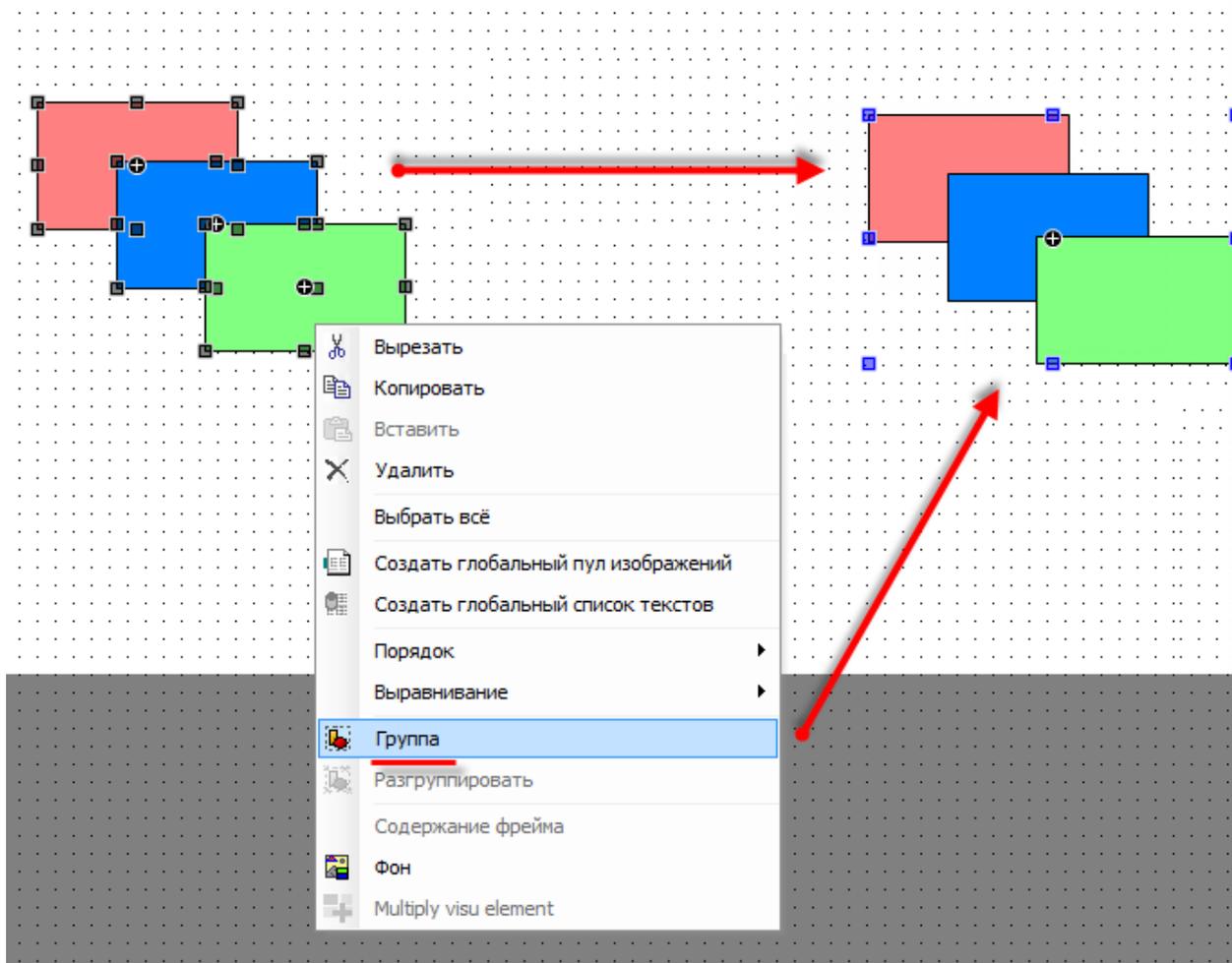


Рис. 6.8. Группировка элементов

5. Команда **Содержимое фрейма** доступна для элементов [Фрейм](#) и [Набор вкладок](#). Она позволяет выбрать экраны визуализации, которые будут отображаться в этих элементах.

6. Команда **Фон** позволяет выбрать цвет фона экрана визуализации или фоновое изображение. Предварительно фоновое изображение должно быть загружено в [Пул изображений](#). В данный момент (**CODESYS 3.5 SP6 Patch 2**) меню выбора фона отображается некорректно в русскоязычной версии **CODESYS** (*исправлено в SP7*).

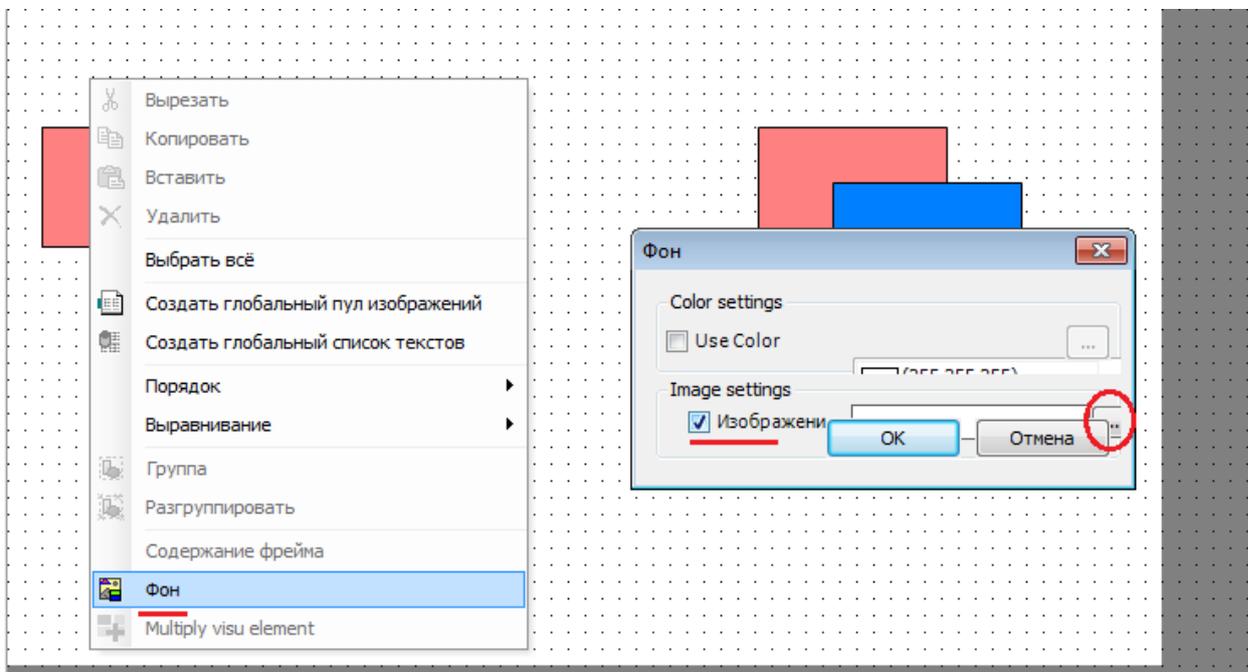


Рис. 6.9. Выбор фоновое изображение экрана визуализации

7. Команда **Multiply Visu Element** позволяет создавать группы однотипных элементов (например, ряды переключателей). Пример ее использования приведен в [п. 11.3.6](#).

6.4. Вспомогательные редакторы экранов визуализации

6.4.1. Редактор интерфейсов

Редактор интерфейсов предназначен для объявления переменных, которые используются на данном экране визуализации. Обычно такие переменные применяются в диалогах и экранах, открываемых во [фреймах](#). Подробную информацию о редакторе интерфейсов можно найти в справке **CODESYS**.

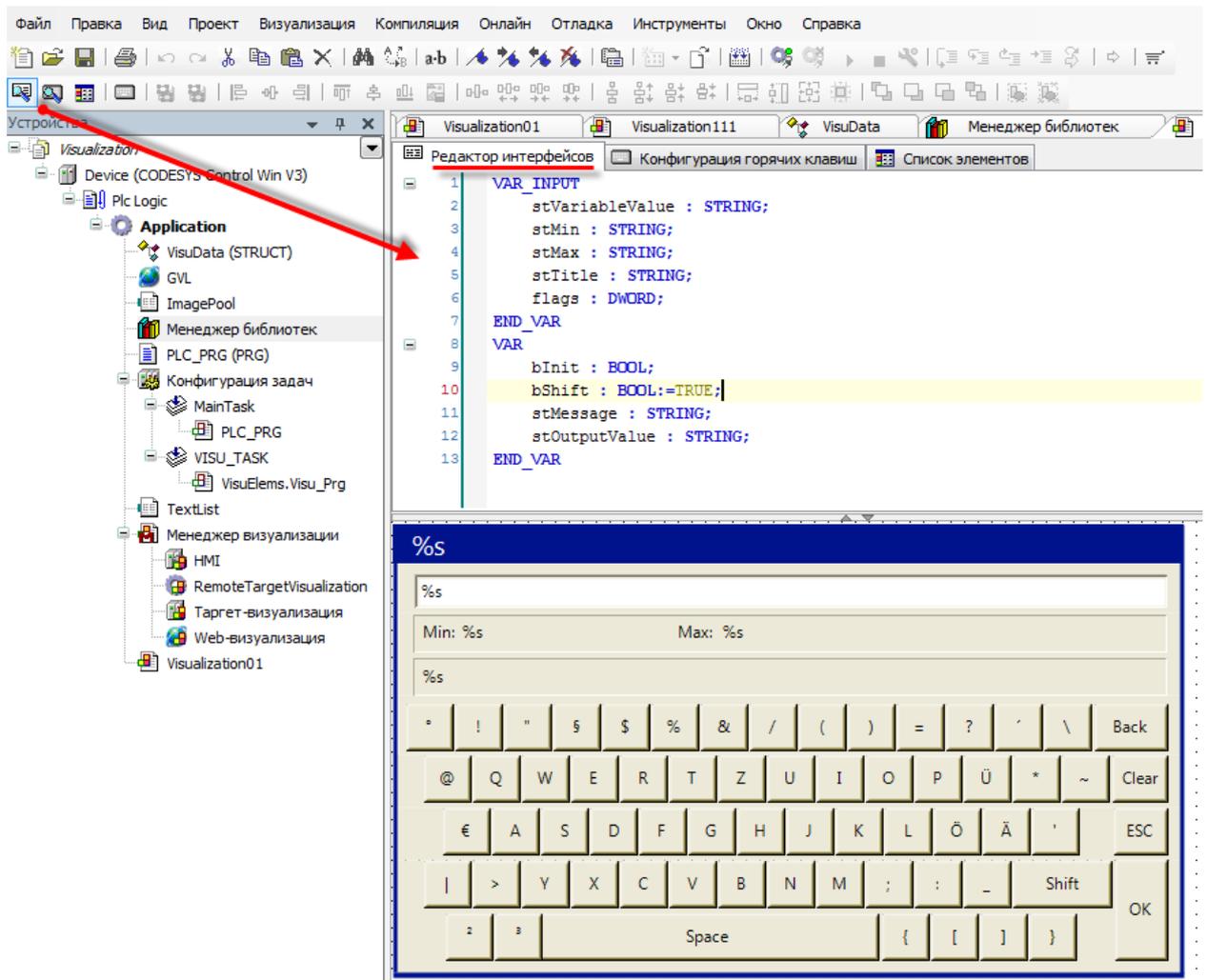


Рис. 6.10. Редактор интерфейсов

6.4.2. Редактор горячих клавиш

В Редакторе горячих клавиш можно привязать к нажатию клавиш выполнение действий. Для этого целевое устройств должно иметь соответствующие кнопки. При отсутствии галочки **Клавиша нажата** действие будет выполняться при отпускании зажатой клавиши. Можно создавать комбинации клавиш с использованием **Shift/Ctrl/Alt**. Для клавиш и их комбинаций можно настраивать права доступа (при наличии в проекте [Управления пользователями](#)) – т.е. разрешить или запретить определенным пользователям использование определенных клавиш.

В отличие от вкладки [Горячие клавиши по умолчанию](#), расположенной в **Менеджере визуализации**, действия клавиш, настроенные для экрана визуализации, будут выполняться только на этом экране.

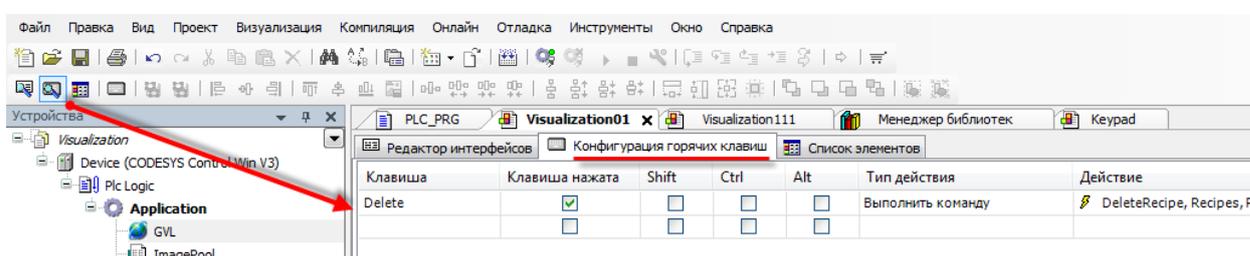


Рис. 6.11. Редактор горячих клавиш экрана визуализации

6.4.3. Редактор списка визуальных элементов

Этот редактор представляет собой список элементов текущего экрана визуализации с возможностью быстрого изменения их размеров и положения на экране.

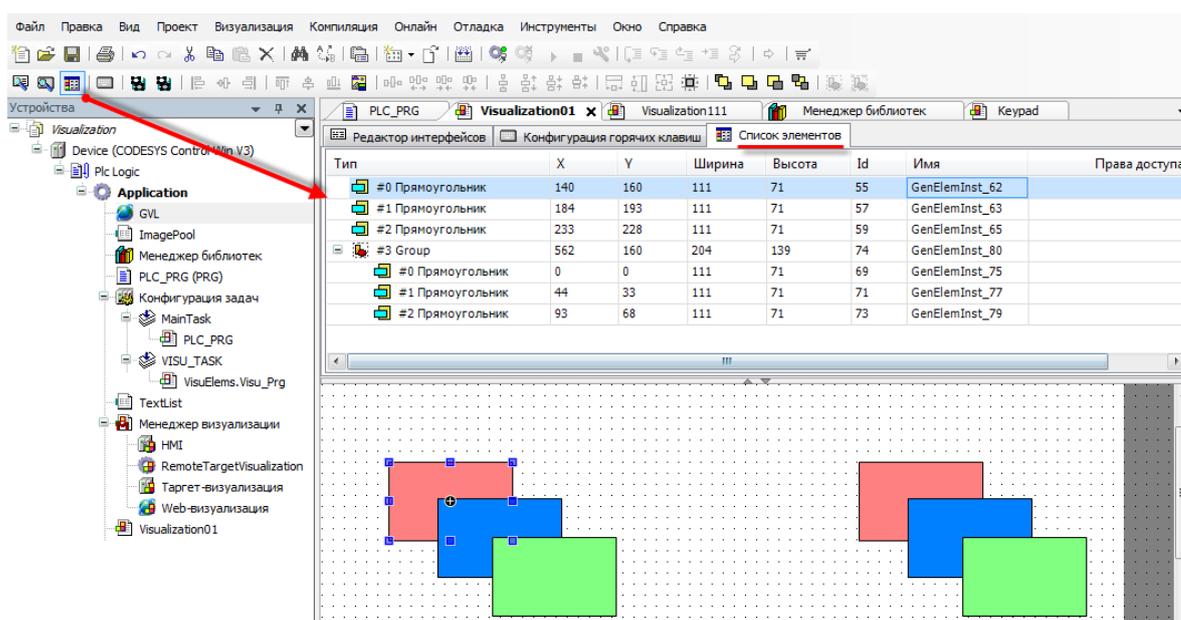


Рис. 6.12. Редактор списка визуальных элементов

6.5. Включение и отключение сетки

По умолчанию в редакторе визуализация включена **сетка**, которая облегчает размещение элементов относительно друг друга. Включить/отключить сетку и настроить ее шаг можно в меню **Инструменты**, вкладка **Опции**, раздел **Визуализация**:

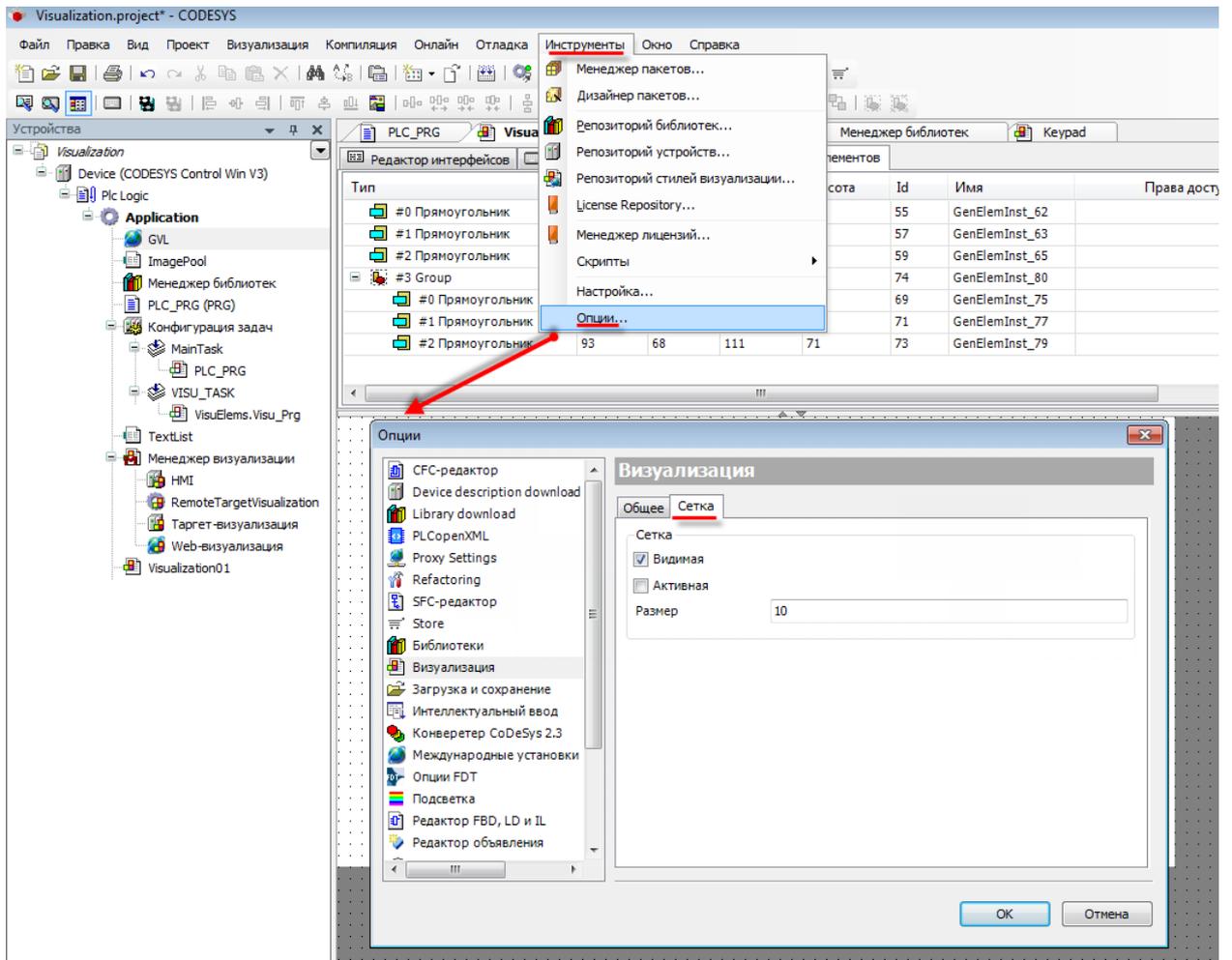


Рис. 6.13. Настройки сетки редактора визуализации

7. Вспомогательные компоненты визуализации

7.1. Пул изображений

Пул изображений представляет собой хранилище графических файлов, используемых в проекте. Поддерживается большинство популярных форматов графических файлов, например, .jpg, .png, .bmp, .svg и т.д.). **Обратите внимание**, что названия файлов не должны содержать кириллические символы, спецсимволы и точки.

Для добавления в проект **Пула изображений** необходимо нажать **ПКМ** на компонент **Application** в **Панели устройств**, раскрыть вкладку **Добавить объект** и выбрать соответствующий компонент:

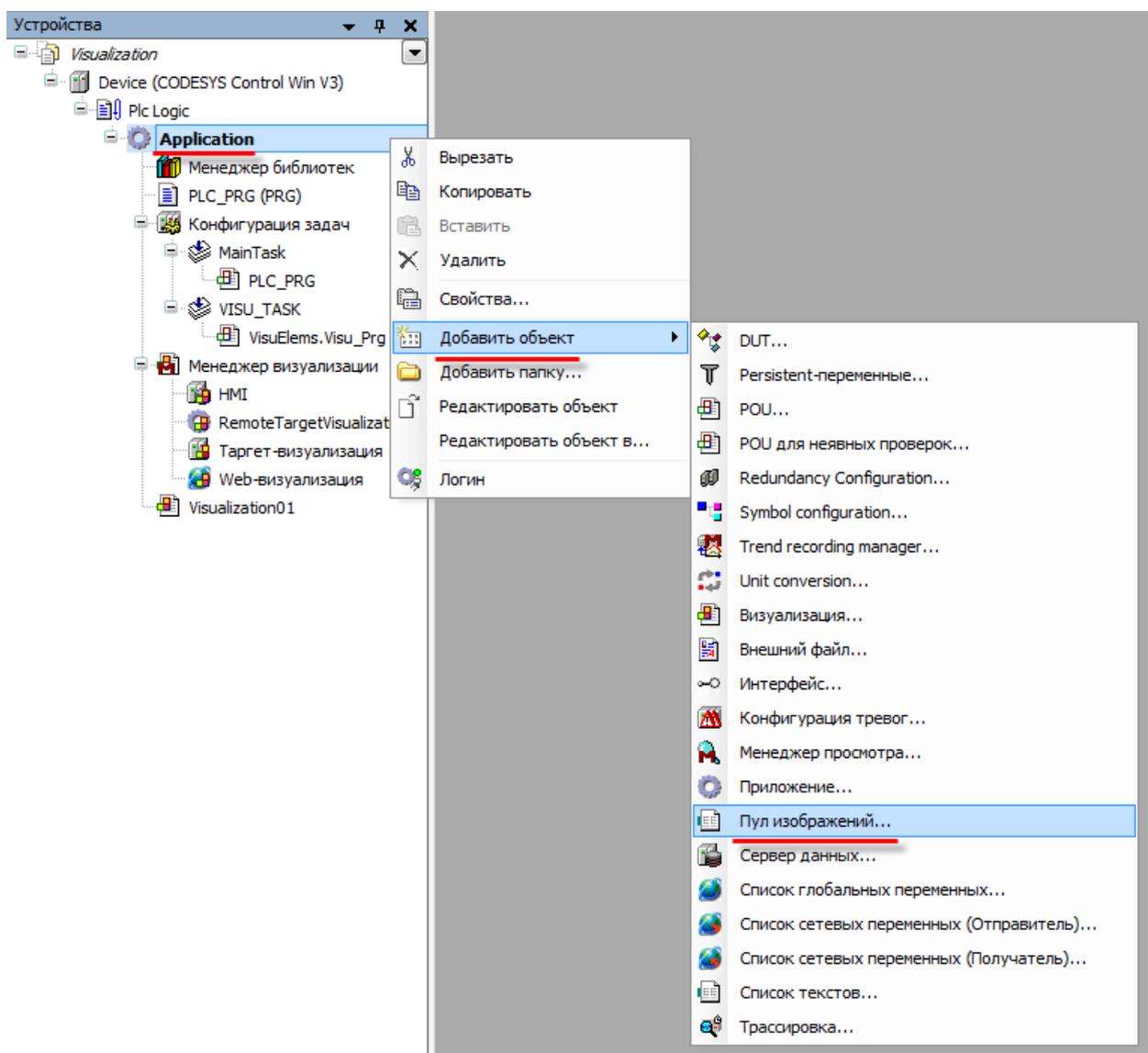


Рис. 7.1. Добавление Пула изображений в проект

Для добавления изображения следует нажать **ЛКМ** на ячейку **Имя файла** и с помощью появившейся кнопки перейти к выбору файла:

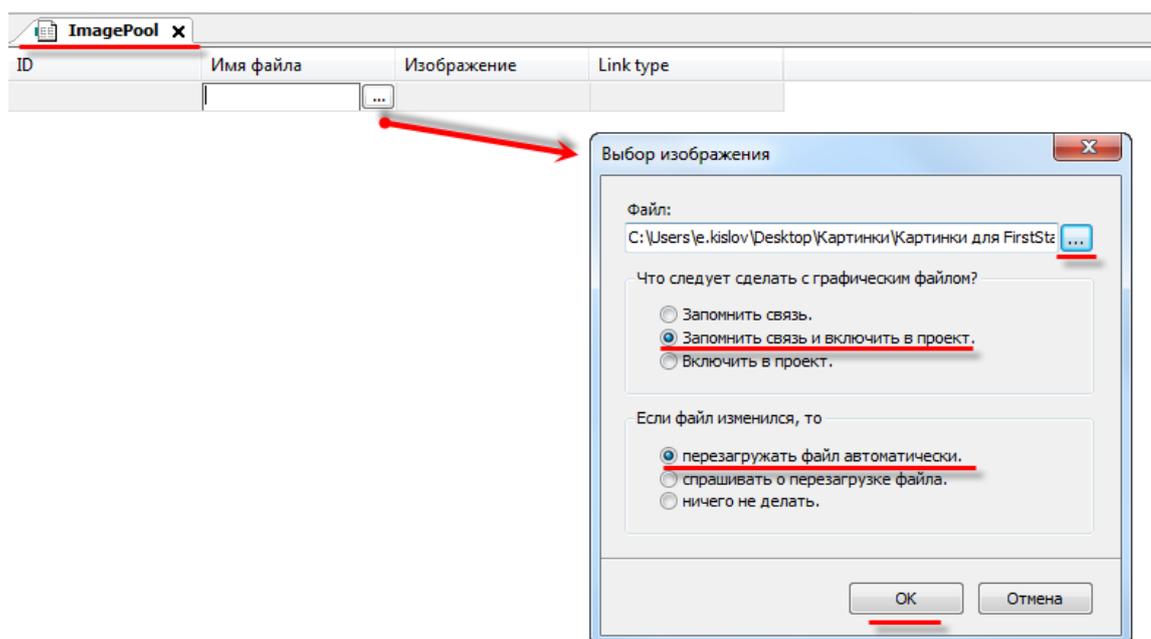


Рис. 7.2. Выбор изображения для загрузки

После указания пути к графическому файлу, рекомендуется в расположенных ниже меню выбрать пункты **Запомнить связь и включить в проект** и **Перезагружать файл автоматически**. Это позволяет не совершать дополнительных операций при изменении изображения – оно будет автоматически меняться в проекте.

После добавления изображения, его пиктограмма отобразится в **Пуле**; также рядом с ним будет указан **идентификатор (ID)** и **тип связи**.

ID	Имя файла	Изображение	Link type
0	luxfon.com_10604.bmp		Embedded and link to file

Рис. 7.3. Пул изображений после добавления файла

Добавленные графические файлы можно использовать для:

1. создания фонового изображения экрана визуализации;
2. создания статических изображений с помощью элемента [Изображение](#);
3. создания динамических (переключаемых по значению логической переменной) изображений с помощью элемента [Переключатель изображения](#);
4. создания пиктограмм для [Менеджера тревог](#).

7.2. Список текстов

Список текстов представляет собой хранилище текстов, используемых в визуализации для реализации *динамических* и *мультязычных* надписей.

Для добавления в проект **Списка текстов** необходимо нажать **ПКМ** на компонент **Application** в **Панели устройств**, раскрыть вкладку **Добавить объект** и выбрать соответствующий компонент:

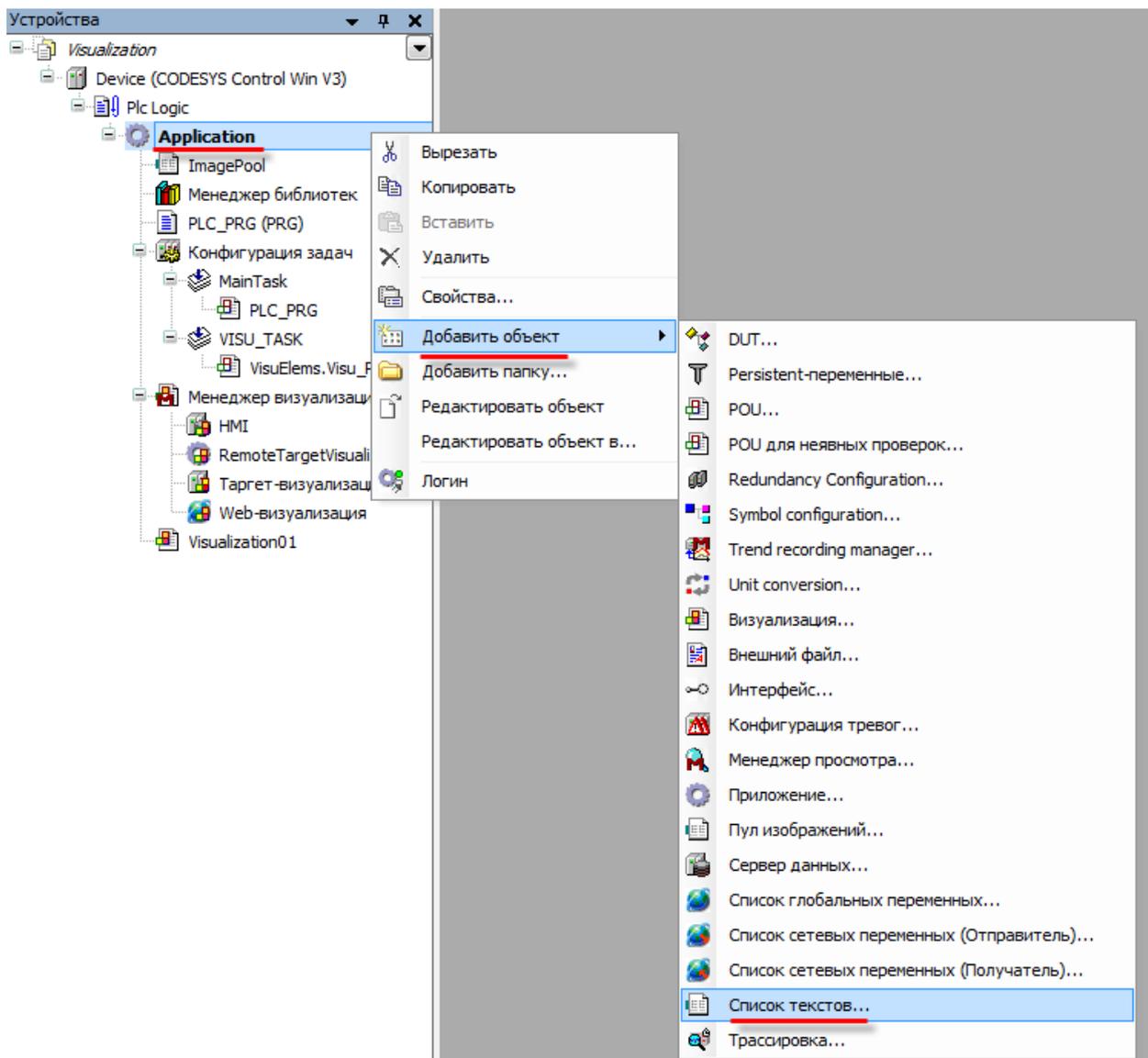


Рис. 7.4. Добавление **Списка текстов** в проект

Контекстное меню, открываемое по нажатию **ПКМ**, позволяет добавить в список текстов новый язык, а также импортировать и экспортировать списки текстов в формате .csv.

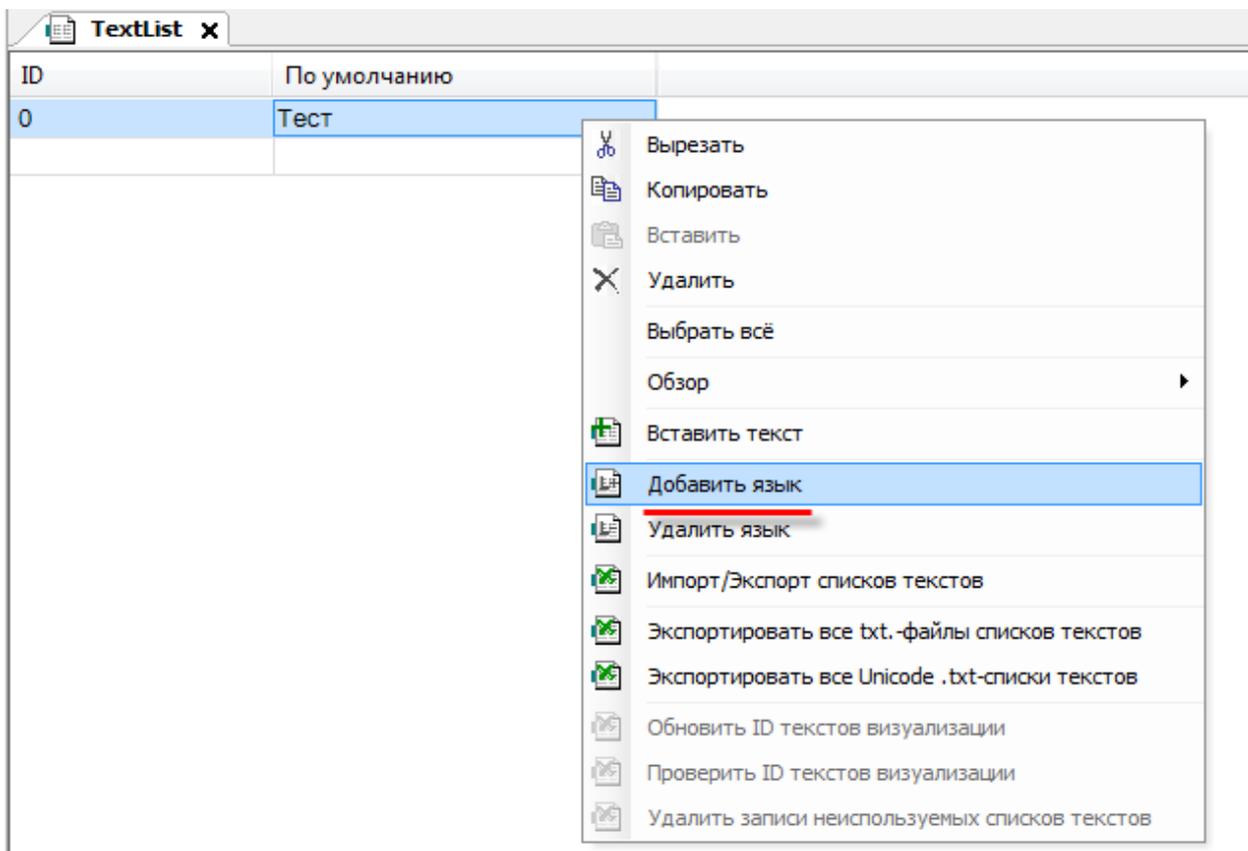


Рис. 7.5. Добавление нового языка в список текстов

The screenshot shows the 'TextList' application window with a table containing three rows of text in three different languages: Russian, English, and Japanese.

ID	По умолчанию	Русский	English	日本語
0		Тест1	Test1	テスト1
1		Тест2	Test2	テスト2
2		Тест3	Test3	テスト3

Рис. 7.6. Мультиязычный список текстов

Пример реализации **мультиязычного проекта** приведен в [п. 11.3.5.](#)

8. Описание графических примитивов

8.1. Структура главы

В данной главе приводится описание параметров всех графических примитивов, присутствующих в CODESYS версии 3.5 SP6 Patch 2. Поскольку значительная часть параметров является общей для большинства элементов, они будут рассмотрены один раз – на примере элементов базовой группы [Прямоугольник](#), [Скругленный прямоугольник](#), [Эллипс](#).

Параметры элементов отображаются на **Панели свойств**, открывающейся при выделении элемента нажатием **ЛКМ**. **Обратите внимание**, что для отображения всех параметров элемента, необходимо поставить галочку **Эксперт**:

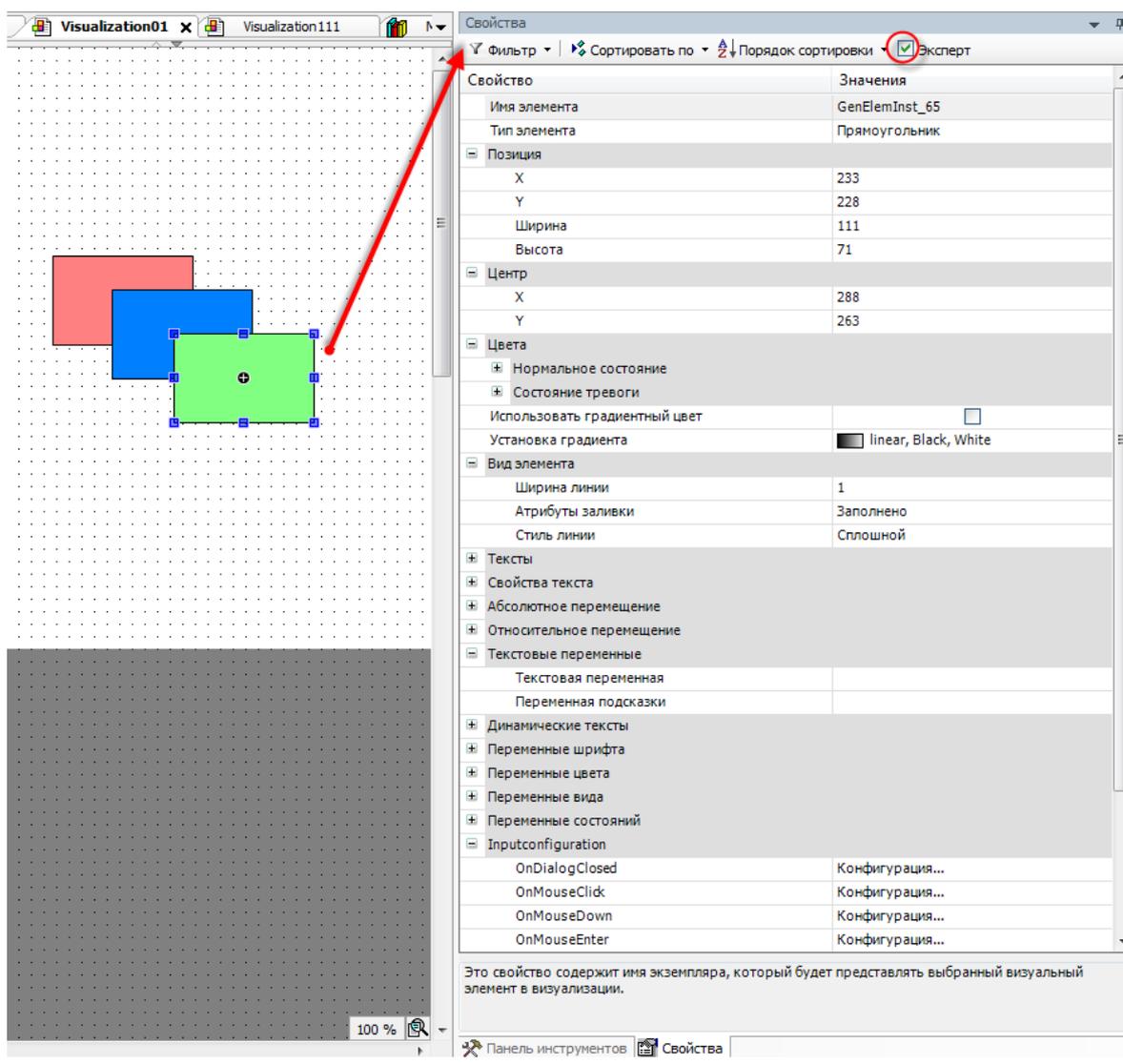


Рис. 8.1. Настройки сетки редактора визуализации

8.2. Базовая группа элементов

8.2.1. Прямоугольник, Скругленный прямоугольник, Эллипс.

Типичные параметры графических примитивов

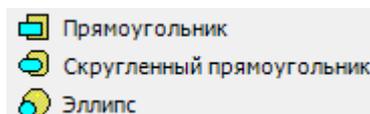


Рис. 8.2. Пиктограммы графических примитивов
Прямоугольник, Скругленный прямоугольник, Эллипс

Графические примитивы **Прямоугольник, Скругленный прямоугольник, Эллипс** представляют собой геометрические фигуры, обладающие идентичными параметрами и отличающиеся только внешним видом:

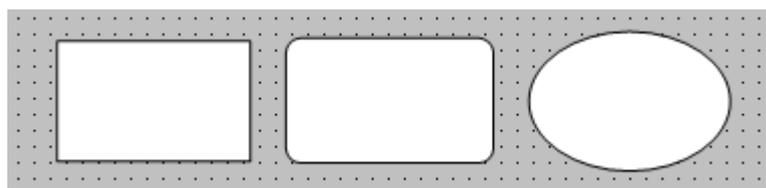


Рис. 8.3. Внешний вид элементов **Прямоугольник, Скругленный прямоугольник, Эллипс**

Элементы могут преобразоваться друг в друга с помощью изменения **типа**:

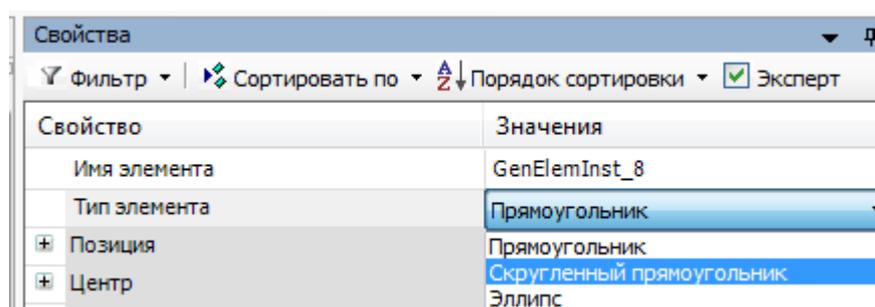


Рис. 8.4. Изменение типа элемента

Описание параметров данных графических примитивов приведено в табл. 2. Поскольку значительная часть этих параметров присутствует и у других графических примитивов, в дальнейших главах будет значительное количество ссылок на эту таблицу.

Пример работы с параметрами данных графических примитивов приведен в [п. 11.2.1.](#)

Табл. 2. Параметры графических примитивов
Прямоугольник, Скругленный прямоугольник, Эллипс

№	Название параметра	Описание параметра
1.	Позиция	
1.1.	X	Координата левой верхней опорной точки элемента по горизонтальной оси в пикселях. Возможные значения: -32768...32767
1.2.	Y	Координата левой верхней опорной точки элемента по вертикальной оси в пикселях. Возможные значения: -32768...32767
1.3.	Ширина	Ширина элемента в пикселях. Возможные значения: -32768...32767
1.4.	Высота	Высота элемента в пикселях. Возможные значения: -32768...32767
2.	Центр	
2.1.	X	Координата опорной точки центра (\oplus) элемента по горизонтальной оси в пикселях. Эта точка не является геометрическим центром элемента и может находиться за его пределами; при абсолютном вращении элемент перемещается относительно этой точки. Возможные значения: -32768...32767
2.2.	Y	Координата опорной точки центра (\oplus) элемента по вертикальной оси в пикселях. Возможные значения: -32768...32767
3.	Цвета	
3.1.	Нормальное состояние (см. пп. 12.1)	
3.1.1.	Цвет фрейма	Цвет контура элемента в нормальном состоянии.
3.1.1.1.	Цвет фрейма/ Прозрачность	Прозрачность контура элемента в нормальном состоянии. Возможные значения: 0...255 0 соответствует полностью прозрачному контуру, 255 – полностью непрозрачному.
3.1.2.	Цвет заливки	Цвет заливки элемента в нормальном состоянии.
3.1.2.1.	Цвет заливки/ Прозрачность	Прозрачность заливки элемента в нормальном состоянии. Возможные значения: 0...255 0 соответствует полностью прозрачной заливке, 255 – полностью непрозрачной.
3.2.	Состояние тревоги (см. пп. 12.1)	
3.2.1.	Цвет фрейма	Цвет контура элемента в состоянии тревоги.
3.2.1.1.	Цвет фрейма/ Прозрачность	Прозрачность контура элемента в состоянии тревоги. Возможные значения: 0...255 . 0 соответствует полностью прозрачному контуру, 255 – полностью непрозрачному.
3.2.2.	Цвет заливки	Цвет заливки элемента в состоянии тревоги.
3.2.2.1.	Цвет заливки/ Прозрачность	Прозрачность заливки в состоянии тревоги. Возможные значения: 0...255 . 0 соответствует полностью прозрачному контуру, 255 – полностью непрозрачному.

3.3.	Использовать градиентный цвет	При наличии галочки в данном параметре, для заливки элемента используется градиентный цвет.
3.4.	Установка градиента	Настройки градиентного цвета.
4.	Вид элемента	
4.1.	Ширина линии	Ширина линии контура элемента в пикселях. Возможные значения: -32768...32767 . Отрицательные значения соответствуют отсутствию контура.
4.2.	Атрибуты заливки	Данный параметр позволяет включать/отключать заливку элемента. Возможные значения: Заполнено/Пустой
4.3.	Стиль линии	Стиль линии контура элемента. Возможные значения: Сплошная  Штриховая  Пунктирная  Штрих-пунктирная линия (одна точка)  Штрих-пунктирная линия (две точки)  Невидимая
5.	Тексты	
5.1.	Текст	Числовая и/или текстовая информация, отображаемая элементом. Можно использовать спецификаторы формата вывода для отображения привязанной к элементу переменной (параметр Текстовая переменная , пп. 9.1). Информацию о форматировании текста см. в п. 9 .
5.2.	Подсказка	Числовая и/или текстовая информация, отображаемая подсказкой элемента. Подсказка всплывает над элементом при наведении на него курсором. Можно использовать спецификаторы формата вывода для отображения привязанной к элементу переменной (параметр Переменная подсказки , пп. 9.2). Информацию о форматировании текста см. в п. 9 .
6.	Свойства текста	
6.1.	Горизонтальное выравнивание	Возможные значения: Слева/По центру/Справа
6.2.	Вертикальное выравнивание	Возможные значения: Слева/По центру/Справа
6.3.	Шрифт	Позволяет выбрать шрифт текста элемента, а также определить его размер и стиль (жирный, курсив и т.д.).
6.4.	Цвет шрифта	Цвет шрифта текста элемента.
6.4.1.	Цвет шрифта/ Прозрачность	Прозрачность текста элемента. Возможные значения: 0...255 0 соответствует полностью прозрачному тексту, 255 – полностью непрозрачному.

7.	Абсолютное перемещение	
7.1.	<i>Перемещение</i>	
7.1.1.	X	Целочисленная переменная, определяющая расстояние в пикселях, на которое переместится элемент по горизонтальной оси относительно своего начального расположения.
7.1.2.	Y	Целочисленная переменная, определяющая расстояние в пикселях, на которое переместится элемент по вертикальной оси относительно своего начального расположения.
7.2.	Вращение	Целочисленная переменная, определяющая вращение в градусах элемента вокруг центральной опорной точки (⊕, пп. 2), при этом элемент остается неподвижным относительно своего геометрического центра.
7.3.	Масштабирование	Целочисленная переменная, определяющая коэффициент линейного масштабирования элемента. Исходному масштабу соответствует значение 1000 . При отрицательном коэффициенте элемент зеркально (по обоим осям) отрисовывается относительно центральной опорной точки (⊕, пп. 2).
7.4.	Внутреннее вращение	Целочисленная переменная, определяющая вращение в градусах элемента вокруг центральной опорной точки (⊕, пп. 2), при этом элемент также вращается вокруг своего геометрического центра (на то же число градусов).
8.	Относительное перемещение	
8.1.	<i>Перемещение сверху налево</i>	
8.1.1.	X	Целочисленная переменная, определяющая перемещение левой стороны элемента вдоль горизонтальной оси в пикселях, при этом правая сторона сохраняет неподвижность.
8.1.2.	Y	Целочисленная переменная, определяющая перемещение верхней стороны элемента вдоль вертикальной оси в пикселях, при этом нижняя сторона сохраняет неподвижность.
8.2.	<i>Перемещение снизу вверх</i>	
8.2.1.	X	Целочисленная переменная, определяющая перемещение правой стороны элемента вдоль горизонтальной оси в пикселях, при этом левая сторона сохраняет неподвижность.
8.2.2.	Y	Целочисленная переменная, определяющая перемещение нижней стороны элемента вдоль вертикальной оси в пикселях, при этом верхняя сторона сохраняет неподвижность.
9.	Текстовые переменные	
9.1.	Текстовая переменная	Переменная (может быть как строковой, так и числовой), значение которой будет отображаться элементом при использовании соответствующего спецификатора (см. параметр Текст , пп. 5.1).
9.2.	Переменная подсказки	Переменная (может быть как строковой, так и числовой), значение которой будет отображаться в подсказке элементов при использовании соответствующего спецификатора (см. параметр Подсказка , пп. 5.2).

10.	Динамические тексты	
10.1.	Список текстов	Список текстов , используемый для динамического текста данного элемента.
10.2.	Индекс текста	Строковая переменная, содержащая ID из указанного списка текстов, используемого для динамического текста данного элемента.
10.3.	Индекс подсказки	Строковая переменная, содержащая ID из указанного списка текстов, используемого для динамического текста подсказки элемента.
11.	Переменные шрифта	
11.1.	Имя шрифта	Строковая переменная, содержащая имя шрифта текста элемента. Обратите внимание , что шрифт должен поддерживаться целевым устройством.
11.2.	Размер	Переменная типа INT , определяющая размер текста элемента в пикселях.
11.3.	Флаги	Переменная типа DWORD , определяющая стиль текста элемента. Возможные значения: 1 – <i>курсив</i> 4 – <u>подчеркнутый</u> 2 – жирный 8 – зачеркнутый Флаги могут складываться, например: 14=8+4+2 – зачеркнутый подчеркнутый жирный текст
11.4.	Набор символов	Переменная типа DWORD , определяющая кодировку текста элемента. В данный момент (CODESYS 3.5 SP6) этот функционал не работает.
11.5.	Цвет	Переменная типа DWORD , определяющая цвет текста по модели RGBA .
11.6.	Flags for text alignment	Целочисленная переменная, определяющая выравнивание текста элемента. Возможные значения: 0 – сверху по левому краю 1 – сверху по центру 2 – сверху по правому краю 4 – по центру по левому краю 5 – по центру 6 – по центру по правому краю 8 – снизу по левому краю 9 – снизу по центру 10 – снизу по правому краю
12.	Переменные цвета	
12.1.	Переключить цвет	Переменная типа BOOL , определяющая состояние цвета элемента. FALSE – нормальное состояние, TRUE – состояние тревоги
12.2.	Нормальное состояние	
12.2.1.	Цвет фрейма	Переменная типа DWORD , определяющая цвет и прозрачность контура элемента в нормальном состоянии по модели RGBA .
12.2.2.	Цвет заливки	Переменная типа DWORD , определяющая цвет и прозрачность заливки элемента в нормальном состоянии по модели RGBA .
12.3.	Состояние тревоги	

12.3.1.	Цвет фрейма	Переменная типа DWORD , определяющая цвет и прозрачность контура элемента в состоянии тревоги по модели RGBA .
12.3.2.	Цвет заливки	Переменная типа DWORD , определяющая цвет и прозрачность заливки элемента в состоянии тревоги по модели RGBA .
13.	Переменные вида	
13.1.	Ширина линии	Целочисленная переменная, определяющая толщину линии контура элемента в пикселях.
13.2.	Атрибуты заливки	Переменная типа DWORD , определяющая активность заливки элемента. Возможные значения: 0 – заливка включена >0 – заливка отключена (элемент прозрачен)
13.3.	Стиль линии	Переменная типа DWORD , определяющая стиль линии контура элемента. Возможные значения: 0 – сплошная , 1 – штриховая, 2 – пунктирная 3 – штрих-пунктирная (одна точка) 4 – штрих-пунктирная (две точки) 8 – невидимая
14.	Переменные состояний	
14.1.	Невидимый	Переменная типа BOOL , определяющая видимость элемента. Невидимый элемент становится неактивным. Возможные значения: FALSE – видимый TRUE – невидимый
14.2.	Отключение ввода	Переменная типа BOOL , определяющая реакцию элемента на нажатие. Возможные значения: FALSE – есть реакция TRUE – нет реакции. Такой элемент отображается в визуализации менее насыщенным цветом
15.	InputConfiguration	
	Данная вкладка содержит настройки действий, выполняемых при нажатии элемент. Ее подробное описание приведено в п. 10 .	
16.	Права доступа	
	Данная вкладка доступна только при наличии в проекте Управления пользователями и позволяет задать для каждой группы пользователей права доступа к данному элементу.	
	Возможные значения: Действующий – элемент является видимым для данной группы пользователей и реагирует на нажатие; Only visible – элемент является видимым для данной группы пользователей, но не реагирует на нажатие; Невидимый – элемент является невидимым для данной группы пользователей.	

8.2.2. Линия

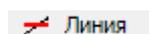


Рис. 8.5. Пиктограмма графического примитива **Линия**

Графический примитив **Линия** представляет собой отрезок.

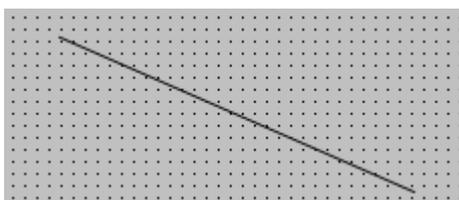


Рис. 8.6. Внешний вид элемента **Линия**

Описание типичных параметров графического примитива приведено в [табл.2](#).

Уникальными параметрами графического примитива являются только координаты точек начала и конца отрезка в пикселях:

Свойства	
Свойство	Значения
Имя элемента	GenElemInst_104
Тип элемента	Линия
Позиция	
Точки	
[0]	
X	344
Y	821
[1]	
X	638
Y	950

Рис. 8.7. Уникальные параметры графического примитива **Линия**

8.2.3. Полигон, Ломаная, Кривая Безье

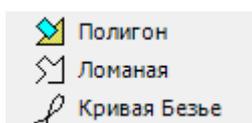


Рис. 8.8. Пиктограммы графических примитивов **Полигон, Ломаная, Кривая Безье**

Графические примитивы **Полигон, Ломаная, Кривая Безье** представляют собой соответствующие геометрические фигуры. Элементы можно конвертировать друг в друга, меняя **тип элемента**:

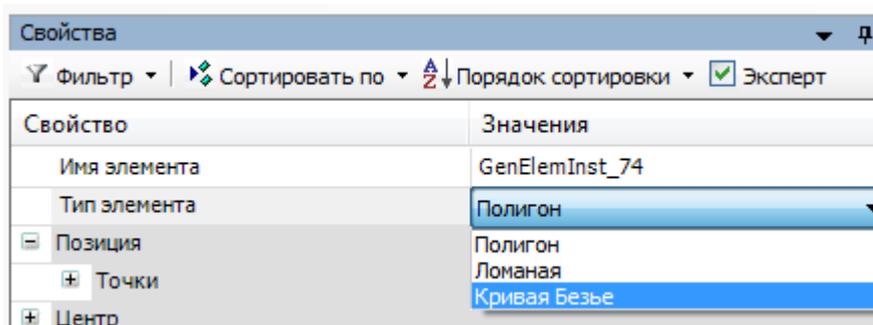


Рис. 8.9. Изменение типа элемента

После выделения графического примитива на **Панели инструментов редактора**, каждое нажатие **ЛКМ** на рабочем поле создает точку и соединяет ее с предыдущей (в случае полигона – с предыдущей и первой). Для окончания добавления точек необходимо нажать **ПКМ** на любой области рабочего поля. После окончания создания фигуры можно добавить точку, нажав на одну из существующих точек **ЛКМ** при зажатой клавише **Ctrl** (новая точка создается поверх существующей). Удалить точку можно нажатием на нее **ЛКМ** при зажатых клавишах **Ctrl** и **Shift**.

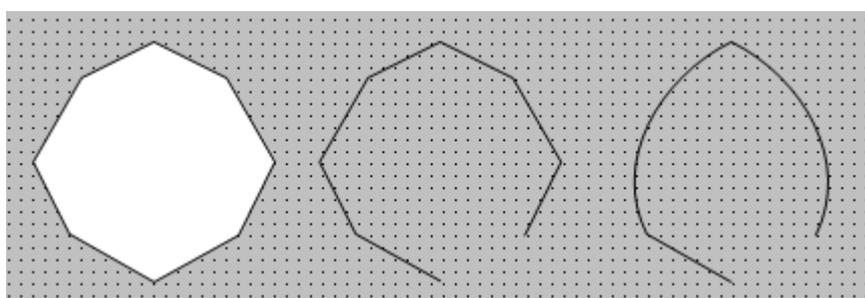


Рис. 8.10. Внешний вид графических примитивов **Полигон, Ломаная, Кривая Безье**

Описание типичных параметров графического примитива приведено в [табл. 2](#).

Уникальными параметрами графического примитива являются параметры вкладки
Динамические точки:

Динамические точки	
Массив точек	
Число точек	

Массив точек – это адрес массива типа **VisuStructPoint**, содержащий текущие координаты точек элемента.

Число точек – целочисленная переменная, содержащая число точек элемента.

Изменение этих параметров в процессе работы программы позволяет менять внешний вид элемента. Пример их использования приведен в [п. 11.4.2](#).

8.2.4. Секторная диаграмма

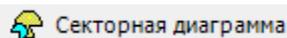


Рис. 8.11. Пиктограмма графического примитива **Секторная диаграмма**

Графический примитив **Секторная диаграмма** представляет собой окружность с вырезанным сектором.

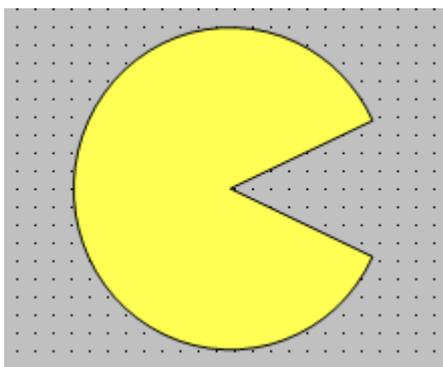
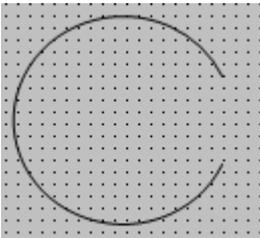


Рис. 8.12. Внешний вид графического примитива **Секторная диаграмма**

Описание типичных параметров графического примитива приведено в [табл. 2](#). Уникальные параметры примитива описаны в табл. 3:

Табл. 3. Уникальные параметры графического примитива **Секторная диаграмма**

№	Название параметра	Описание параметра
1.	Начало	Угол в градусах между центром элемента и начальной точкой сектора.
2.	Конец	Угол в градусах между центром элемента и конечной точкой сектора.
3.	Переменная для начала	Целочисленная переменная, определяющая значение угла в градусах между центром элемента и начальной точкой сектора.
4.	Переменная для конца	Целочисленная переменная, определяющая значение угла в градусах между центром элемента и конечной точкой сектора.
5.	Показывать только линию круга	При наличии галочки секторная диаграмма отображается как неполная окружность без заливки: 

8.2.5. Изображение

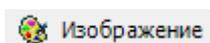


Рис. 8.13. Пиктограмма графического примитива **Изображение**

Графический примитив **Изображение** используется для отображения графического файла. Поддерживается большинство популярных графических форматов, таких как .jpg, .png, .bmp, .svg и т.д. Можно выбрать либо одно из системных изображений (используемых для отображения графических примитивов), либо пользовательское изображение, добавленное в [Пул изображений](#).



Рис. 8.14. Внешний вид элемента **Изображение**

Описание типичных параметров графического примитива приведено в [табл. 2](#).

Уникальные параметры примитива описаны в табл. 4:

Табл. 4. Уникальные параметры графического примитива **Изображение**

№	Название параметра	Описание параметра
1.	Показать фрейм	При наличии галочки вокруг изображения отображается контур.
2.	Кадрирование	При наличии галочки и типа шкалы Фиксированный (см. пп. 5) в элементе отображается только соответствующая его размеру часть изображения.
3.	Прозрачный	При наличии галочки, цвет, указанный в параметре Прозрачный цвет , не отображается элементом. Это можно использовать в тех случаях, когда исходное изображение имеет непрозрачный фон, отличный от фона экрана визуализации.
4.	Прозрачный цвет	Указанный здесь цвет при наличии галочки Прозрачный не будет отображаться.
5.	Тип шкалы	<p>Тип масштабирования изображения при изменении размеров элемента. Возможные значения:</p> <p>Изотропия – при изменении размеров элемента, изображение масштабируется с сохранением соотношения сторон;</p> <p>Анизотропный – при изменении размеров элемента, изображение масштабируется без сохранения соотношения сторон;</p> <p>Фиксированный – при изменении размеров элемента, изображение сохраняет свой размер. При наличии галочки в параметре Кадрирование в элементе отображается только соответствующая размеру элемента часть изображения.</p>
6.	Переменная ID изображения/ ID изображения	<p>Строковая переменная, определяющая имя пула изображений и ID файла из него.</p> <p>Пример: MyImagePool.MyID</p> <p>При обращении к пулу изображений, добавленному в проект первым, имя пула можно опускать.</p> <p>Меняя значение этой переменной в цикле программы, можно реализовать анимацию.</p>

8.2.6. Фрейм элемента

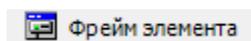


Рис. 8.15. Пиктограмма графического примитива **Фрейм элемента**

Графический примитив **Фрейм элемента** используется для открытия одного экрана визуализации в плоскости другого. Фрейм может содержать несколько экранов визуализации с возможностью переключения между ними.

Обратите внимание, что если у экрана, открываемого во фрейме, объявлены переменные [интерфейса](#), то в конфигурации фрейма к ним можно привязать переменные программы. Иными словами, с помощью фрейма можно открывать разные экземпляры одного экрана визуализации, которые будут отличаться только привязанными переменными. Пример использования интерфейса фрейма приведен в [п. 11.3.3](#).



Рис. 8.16. Внешний вид элемента **Фрейм**

Описание типичных параметров графического примитива приведено в [табл. 2](#).

Уникальные параметры примитива описаны в табл. 5.

Пример работы с элементом приведен в [п. 11.2.2](#).

Табл. 5. Уникальные параметры графического примитива **Фрейм**

№	Название параметра	Описание параметра
1.	Кадрирование	При наличии галочки и типа шкалы Фиксированный (пп. 3) во фрейме отображается только соответствующая его размеру часть экрана визуализации.
2.	Показать фрейм	При наличии галочки вокруг фрейма отображается контур.
3.	Тип шкалы	<p>Тип масштабирования экрана визуализации в фрейме при изменении его (фрейма) размера. Возможные значения:</p> <p>Изотропия – при изменении размера фрейма, открытый в нем экран визуализации масштабируется с сохранением соотношения сторон;</p> <p>Анизотропный – при изменении размера фрейма, открытый в нем экран визуализации масштабируется без сохранения соотношения сторон;</p> <p>Фиксировано – при изменении размера фрейма, открытый в нем экран визуализации сохраняет свой размер. При наличии галочки в параметре Кадрирование во фрейме отображается только соответствующую размеру фрейма часть экрана визуализации;</p> <p>Фиксирована и прокручивается – открытый в фрейме экран визуализации отображается без масштабирования. Если размер экрана превышает размер фрейма, то перемещение по экрану осуществляется с помощью полос прокрутки. При выборе данного типа масштабирования параметр Кадрирование не отображается.</p>
4.	Параметры полосы прокрутки	<p>(данный параметр доступен только при выборе в пп. 3 типа шкалы Фиксирована и прокручивается)</p> <p>Целочисленные переменные, определяющие текущие позиции горизонтальной/вертикальной полосы прокрутки. Рекомендуется использовать различные переменные для разных клиентов визуализации для реализации возможности независимой прокрутки содержимого фрейма каждым из клиентов.</p>
5.	Отключить нанесение фона	Для оптимизации производительности, статические элементы фрейма по умолчанию отображаются как фоновое изображение. Это может привести к неожиданным графическим эффектам; при наличии галочки у данного параметра, оптимизация отключается.
6.	Ссылки	Данный параметр определяет экраны визуализации, которые могут отображаться в данном фрейме, а также позволяет привязать к переменным интерфейса открываемого экрана переменные программы (см. п. 11.3.3).
7.	Переключить переменную фрейма	<p>Целочисленная переменная, содержащая индекс отображаемого во фрейме в данный момент экрана визуализации; меняя значение переменной, можно переключать экраны фрейма.</p> <p>Индекс первого экрана визуализации фрейма – 0, второго – 1 и т.д. Порядок экранов определяется в настройках п. 6.</p>

8.3. Стандартная группа элементов

8.3.1. Метка

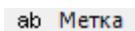


Рис. 8.17. Пиктограмма графического примитива **Метка**

Графический примитив **Метка** используется для отображения статического текста. Переход на следующую строку во время набора текста осуществляется комбинацией клавиш **Ctrl+Enter**.

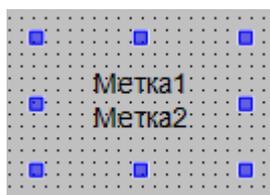


Рис. 8.18. Внешний вид элемента **Метка**

Описание типичных параметров графического примитива приведено [в табл. 2](#).

8.3.2. Комбинированное окно (Целочисленное)

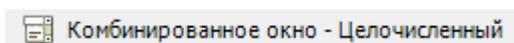


Рис. 8.19. Пиктограмма графического примитива **Комбинированного окна (Целочисленное)**

Графический примитив **Комбинированное окно (Целочисленное)** используется для реализации выбора из выпадающего списка. Выпадающий список может содержать изображения, добавленные [Пул изображений](#) и тексты, добавленные в [Список текстов](#).

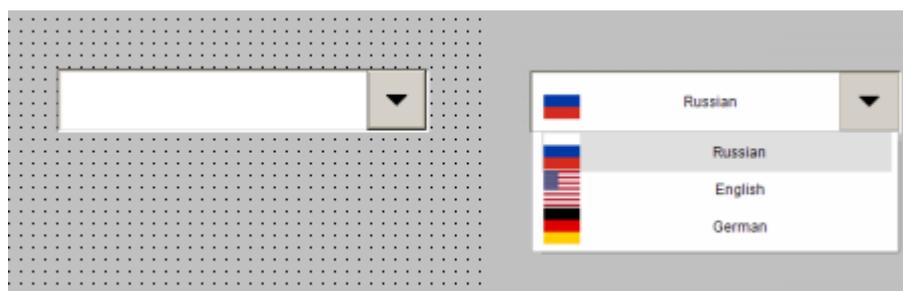


Рис. 8.20. Внешний вид элемента **Комбинированного окна (Целочисленное)**

Описание типичных параметров графического примитива приведено в [табл. 2](#).

Уникальные параметры примитива описаны в табл. 6.

Пример работы с элементом приведен в [п. 11.2.3](#).

Табл. 6. Уникальные параметры графического примитива
Комбинированное окно (Целочисленное)

№	Название параметра	Описание параметра
1.	Переменная	Целочисленная переменная, содержащая индекс выбранного элемента списка.
2.	Список текстов	Список текстов , содержимое которого отображается в списке элемента; ID из списка текстов должны соответствовать возможным значениям переменной из пп. 1.
3.	Пул изображений	Пул изображений , содержимое которого отображается в списке элемента; ID из пула изображений должны соответствовать возможным значениям переменной из пп. 1.
4.	Максимальное значение	Максимальное количество элементов списка.
5.	Параметры списка	
	При значениях From Style будут использованы настройки выбранного стиля визуализации.	
5.1.	Высота строки	Высота строки списка в пикселях.
5.2.	Число видимых строк	Число видимых строк списка. Если значение меньше, чем установленное в пп. 4, то отображается полоса прокрутки.
5.3.	Высота изображения	Высота изображений в списке в пикселях.
5.4.	Ширина изображения	Ширина изображений в списке в пикселях.
5.5.	Сдвиг изображения	Сдвиг изображений относительно левого края списка в пикселях.
5.6.	Размер полосы прокрутки	Ширина полосы прокрутки в пикселях.
6.	Поддиапазон	
6.1.	Использовать поддиапазон	При наличии галочки, для переменной из пп. 1 определяется диапазон возможных значений.
6.2.	Начальный индекс	Начальный индекс диапазона переменной из пп. 1.
6.3.	Конечный индекс	Конечный индекс переменной диапазона из пп. 1.
6.4.	Отбрасывать недостающие тексты	При наличии галочки, пустые строки в списке не отображаются.
7.	Свойства текста	
7.1.	Использование	Позволяет использовать для текста элементов списка либо настройки стиля визуализации , либо индивидуальные установки (см. пп. 7.2).
7.2.	Индивидуальные свойства текста	(данные параметры доступны только при выборе значения Индивидуальные установки в пп. 7.1). Свойства текста элементов списка. Описание параметров приведено в табл. 2 , пп. 6.

8.3.3. Комбинированное окно (Массив)

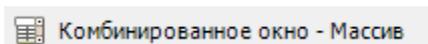


Рис. 8.21. Пиктограмма графического примитива **Комбинированного окна (Массив)**

Графический примитив **Комбинированное окно (Массив)** используется для реализации выбора строки из выпадающей таблицы. Таблица представляет собой одно- или двухмерный массив типа **ARRAY**, формируемый пользователем.

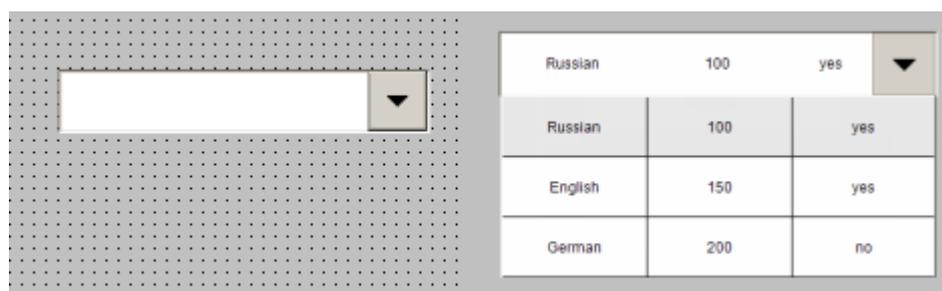


Рис. 8.22. Внешний вид элемента **Комбинированного окна (Массив)**

Описание типичных параметров графического примитива приведено в [табл. 2](#).

Уникальные параметры примитива описаны в табл. 7.

Пример работы с элементом приведен в [п. 11.2.3](#).

Табл. 7. Уникальные параметры графического примитива **Комбинированное окно (Массив)**

№	Название параметра	Описание параметра
1.	Переменная	Целочисленная переменная, содержащая индекс выбранного элемента списка.
2.	Массив данных	Массив данных, отображаемых элементом. Обычно задается в виде двумерного массива ARRAY [0..x, 0..y] , где x – число столбцов, y – число строк.
3.	Столбцы	
3.1.	<i>Столбец</i>	
3.1.	Число столбцов определяется массивом, выбранным в пп. 2.	
3.1.1.	[№]	При наличии галочки, данный массив отображается в таблице элемента.
3.1.2.	Ширина	Ширина данного столбца в пикселях.
3.1.3.	Столбец изображения	При наличии галочки, данный столбец используется для отображения изображений. Значения столбца должны содержать ID (и только ID) изображений из пула изображений . Если в проекте несколько пулов, то элемент будет связан с пулом, добавленным в проект первым.
3.1.4.	Конфигурация изображения	Позволяет выбрать способ отображения изображений – по центру ячейки или растягивание по размеру ячейки (заливка ячейки), а также отображение одного из цветов изображения как прозрачного.
3.1.5.	Выравнивание текста в столбце	Выравнивание текста в данном столбце. Возможные значения: Лево/Право/По центру
3.2.	Максимальный индекс массива	Целочисленная переменная, определяющая число строк, отображаемых в таблице элемента (индекс последней строки из массива, указанного в пп. 2).
3.3.	Высота строки	Высота строки таблицы в пикселях.
3.4.	Число видимых строк	Число видимых строк таблицы элемента. Если значение меньше, чем установленное в пп. 2 и пп. 3.2, то отображается полоса прокрутки.
3.5.	Размер полосы прокрутки	Ширина полосы прокрутки в пикселях.
4.	Свойства текста	
4.1.	Использование	Позволяет использовать для текста элементов списка либо настройки стиля визуализации , либо индивидуальные установки (см. пп. 4.2).
4.2.	Индивидуальные свойства текста	(данные параметры доступны только при выборе значения Индивидуальные установки в пп. 4.1). Свойства текста элементов списка. Описание параметров приведено в табл. 2 , пп. 6.

8.3.4. Набор вкладок

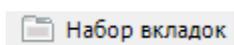


Рис. 8.23. Пиктограмма графического примитива **Набор вкладок**

Графический примитив **Набор вкладок** представляет собой частный случай графического примитива [Фрейм](#) с настроенным интерфейсом переключения экранов визуализаций:

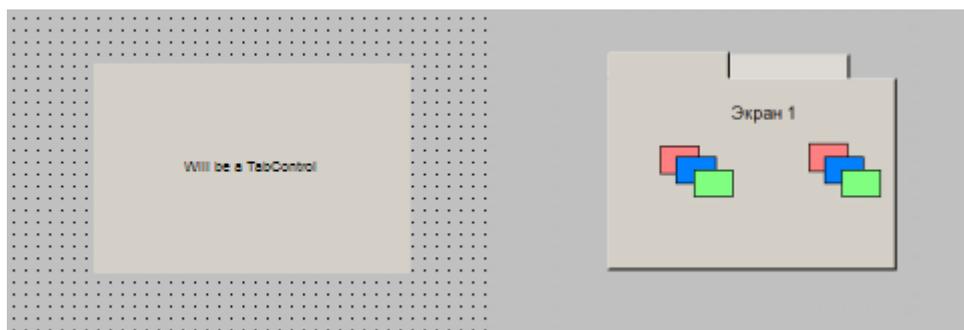


Рис. 8.24. Внешний вид элемента **Набор вкладок**

Описание типичных параметров графического примитива приведено в [табл. 2](#).

Уникальные параметры примитива описаны в табл. 8:

Табл. 8. Уникальные параметры графического примитива **Набор вкладок**

№	Название параметра	Описание параметра
1.	Ширина вкладки	Ширина строки кнопок переключения экранов элемента в пикселях.
2.	Тип шкалы	<p>Тип масштабирования экрана визуализации при изменении размера его фрейма. Возможные значения:</p> <p>Изотропия – при изменении размера фрейма, открытый в нем экран визуализации масштабируется с сохранением соотношения сторон;</p> <p>Анизотропный – при изменении размера фрейма, открытый в нем экран визуализации масштабируется без сохранения соотношения сторон;</p> <p>Фиксировано – при изменении размера фрейма, открытый в нем экран визуализации сохраняет свой размер. При наличии галочки в параметре Кадрирование во фрейме отображается только соответствующую размеру фрейма часть экрана визуализации;</p> <p>Фиксирована и прокручивается – открытый во фрейме экран визуализации отображается без масштабирования. Если размер экрана превышает размер фрейма, то перемещение по экрану осуществляется с помощью полос прокрутки. При выборе данного типа масштабирования параметр Кадрирование не отображается.</p>
3.	Отключить нанесение фона	Для оптимизации производительности, статические элементы фрейма по умолчанию отображаются как фоновое изображение. Это может привести к неожиданным графическим эффектам; при наличии галочки у данного параметра, оптимизация отключается.
4.	Ссылки	Данный параметр определяет экраны визуализации, которые могут отображаться в данном фрейме.
5.	Переключить переменную фрейма	<p>Целочисленная переменная, содержащая индекс отображаемого во фрейме в данный момент экрана визуализации; меняя значение переменной, можно переключать экраны во фрейме.</p> <p>Индекс первого экрана визуализации фрейма – 0, второго – 1 и т.д. Порядок экранов определяется в настройках пп. 4.</p>

8.3.5. Кнопка



Рис. 8.25. Пиктограмма графического примитива **Кнопка**

Графический примитив **Кнопка** в основном используется для переключения значений логических (**BOOL**) переменных.

Переход на следующую строку во время набора текста осуществляется комбинацией клавиш **Ctrl+Enter**.

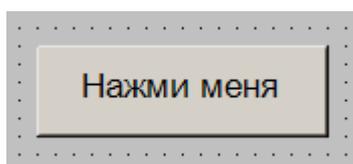


Рис. 8.26. Внешний вид элемента **Кнопка**

Описание типичных параметров графического примитива приведено в [табл. 2](#).

Уникальные параметры примитива описаны в табл. 9.

Пример работы с элементом приведен в [п. 11.2.4](#).

Табл. 9. Уникальные параметры графического примитива **Кнопка**

№	Название параметра	Описание параметра
1.	Высота кнопки	Размер зоны тени/блика краев кнопки в пикселях.
2.	Информация изображения	
2.1.	Статический ID	Изображение, которое будет отображаться данным элементом.
2.2.	Тип шкалы	<p>Тип масштабирования изображения кнопки при изменении ее размера. Возможные значения:</p> <p>Изотропия – при изменении размера кнопки, ее изображение масштабируется с сохранением соотношения сторон;</p> <p>Анизотропный – при изменении размера кнопки, ее изображение масштабируется без сохранения соотношения сторон;</p> <p>Фиксировано – при изменении размера кнопки, его изображение сохраняет свой размер.</p>
2.3.	Прозрачный	При наличии галочки, цвет, указанный в параметре Прозрачный цвет не отображается элементом. Это можно использовать в тех случаях, когда исходное изображение имеет непрозрачный фон, отличный от фона экрана визуализации.
2.4.	Прозрачный цвет	Указанный здесь цвет при наличии галочки Прозрачный не будет отображаться.
2.5.	Горизонтальное выравнивание	Выравнивание изображения кнопки по горизонтальной оси. Возможные значения: Лево/По центру/Право
2.6.	Вертикальное выравнивание	Выравнивание изображения кнопки по вертикальной оси. Возможные значения: Верх/По центру/Низ
3.	Переменная состояния кнопки	
3.1.	Двоичная переменная	<p>Переменная типа BOOL, определяющая текущее состояние кнопки. Возможные значения:</p> <p>TRUE – кнопка нажата FALSE – кнопка не нажата</p>
4.	Переменная ID изображения	
4.1.	ID изображения	Строковая переменная, содержащая ID из пула изображений для текущего изображения кнопки (MyImagePool.MyImageID). Меняя значение этой переменной, можно реализовать изменение изображения кнопки при смене ее состояния.

8.3.6. Группа

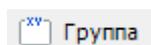


Рис. 8.27. Пиктограмма графического примитива **Группа**

Графический примитив **Группа** представляет собой рамку с заголовком. Все элементы, помещенные внутрь этой рамки, воспринимаются редактором визуализации как сгруппированные (т.е. перемещаются при перемещении рамки), но при этом у пользователя остается возможность настраивать параметры каждого из них.

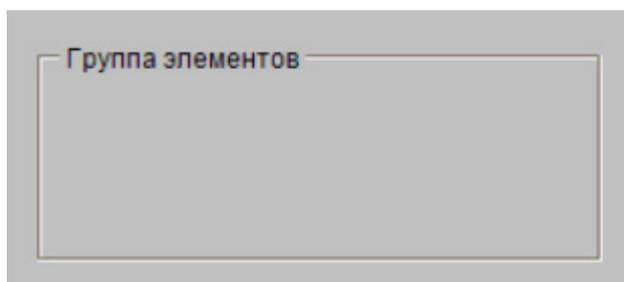


Рис. 8.28. Внешний вид элемента **Группа**

Описание типичных параметров графического примитива приведено в [табл. 2](#).

8.3.7. Таблица



Рис. 8.29. Пиктограмма графического примитива **Таблица**

Графический примитив **Таблица** используется для отображения и ввода данных одно- и двухмерных массивов, структур, переменных функциональных блоков.

Нажатие на любую из ячеек приводит к ее выделению (см. пп. 4.2. в табл. 10). Для снятия выделения достаточно нажать на заголовок столбца или номер строки.

В процессе разработки экрана визуализации, столбец таблицы можно продублировать, нажав на его заголовок **ЛКМ** при зажатой клавише **Ctrl**, или удалить, нажав на его заголовок при зажатых клавишах **Ctrl** и **Shift**.



	Время	Значение	Приоритет
1	15:36:40	11.1	1
2	15:36:50	11.2	1
3	15:37:00	44.4	4
4	15:37:10	11.3	1

Рис. 8.30. Внешний вид элемента **Таблица**

Описание типичных параметров графического примитива приведено в [табл. 2](#).

Уникальные параметры примитива описаны в табл. 10.

Пример работы с элементом приведен в [п. 11.2.5](#).

Табл. 10. Уникальные параметры графического примитива **Таблица**

№	Название параметра	Описание параметра
1.	Массив данных	Массив данных, отображаемых элементом. Может представлять собой переменную типа ARRAY (одно- или двухмерный массив), структуру, функциональный блок и т.д.
2.	Максимальный индекс массива	Целочисленная переменная, определяющая число строк, отображаемых в таблице элемента (индекс последней строки из массива, указанного в п. 1).
3.	Столбцы	
3.1.	Столбец	
	Число столбцов определяется массивом, выбранным в п. 1.	
3.1.1.	[№]	При наличии галочки, данный столбец отображается элементом.
3.1.2.	Ширина	Ширина данного столбца в пикселях.
3.1.3.	Столбец изображения	При наличии галочки, данный столбец используется для отображения изображений. Значения столбца должны содержать ID (и только ID) изображений из пула изображений . Если в проекте несколько пулов, то элемент будет связан с пулом, добавленным в проект первым.
3.1.4.	Конфигурация изображения	Позволяет выбрать способ отображения изображений – по центру ячейки или растягивание по размеру ячейки (заливка ячейки), а также отображение одного из цветов изображения как прозрачного.
3.1.5.	Выравнивание текста заголовка	Выравнивание текста в данном столбце. Возможные значения: Лево/Право/По центру
3.1.6.	Использовать шаблон	При наличии галочки, ячейки данного столбца становятся элементами типа Прямоугольник , и появляется возможность использовать для них настройки соответствующего графического примитива. Стоит отметить, что при этом становится доступна вкладка InputConfiguration , с помощью которой можно реализовать возможность ввода данных в таблицу.
3.1.7.	Выравнивание текста заголовка из шаблона	При наличии галочки, для заголовка используется выравнивание из шаблона (см. пп. 3.1.6).
3.2.	Показать заголовок строки	При наличии галочки, в таблице отображается строка заголовков.
3.3.	Показать заголовок столбца	При наличии галочки, в таблице отображается столбец с номерами строк.
3.4.	Высота строки	Высота строки элемента в пикселях.
3.5.	Ширина строки заголовка	Ширина столбца номеров строк в пикселях.

3.6.	Размер полосы прокрутки	Ширина полосы прокрутки в пикселях.
4.	Выбор	
4.1.	Цвет выбора	Цвет заливки ячейки при ее выделении нажатием курсора.
4.2.	Тип выбора	Тип выделения ячеек таблицы. Возможные значения: Нет выделения/Выбор ячеек/Выбор строк/Выбор столбцов/Выбор строк и столбцов
4.3.	Фрейм вокруг выбранных ячеек	При наличии галочки, вокруг выделенных элементов таблицы отображается контур.
4.4.	Переменная для выбранного столбца	Переменная типа INT , в которую записывается номер выделенного столбца.
4.5.	Переменная для выбранной строки	Переменная типа INT , в которую записывается номер выделенной строки.
4.6.	Переменная для выбора действительного столбца	Переменная типа BOOL , принимающая значение TRUE при выделении любого столбца таблицы (в т.ч. при выделении любой ячейки).
4.7.	Переменная для выбора действительной строки	Переменная типа BOOL , принимающая значение TRUE при выделении любой строки таблицы (в т.ч. при выделении любой ячейки).

8.3.8. Текстовое поле

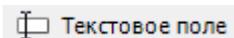


Рис. 8.31. Пиктограмма графического примитива **Текстовое поле**

Графический примитив **Текстовое поле** используется для отображения информации. В отличие от графического примитива [Метка](#), данный примитив способен отображать переменные, а также имеет фон.

Переход на следующую строку во время набора текста осуществляется комбинацией клавиш **Ctrl+Enter**.



Рис. 8.32. Внешний вид элемента **Текстовое поле**

Описание типичных параметров графического примитива приведено в [табл. 2](#).

Уникальные параметры примитива описаны в табл. 11.

Табл. 11. Уникальные параметры графического примитива **Текстовое поле**

№	Название параметра	Описание параметра
1.	Тип тени	Тип тени элемента. Возможные значения: Интенсивный – в тени находятся левая и верхняя сторона элемента Нет тени – тень отсутствует Увеличено – в тени находятся правая и нижняя сторона элемента Из стиля – тип тени определяется текущим стилем визуализации

8.3.9. Полоса прокрутки

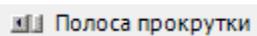


Рис. 8.33. Пиктограмма графического примитива **Полоса прокрутки**

Графический примитив **Полоса прокрутки** используется для управления значением переменной. Для этого могут использоваться кнопки «**Больше**»/ «**Меньше**», перетаскивание ползунка по элементу при зажатой **ЛКМ** или перемещение ползунка по нажатию **ЛКМ** на любую точку полосы прокрутки.



Рис. 8.34. Внешний вид элемента **Полоса прокрутки**

Описание типичных параметров графического примитива приведено в [табл. 2](#).

Уникальные параметры примитива описаны в табл. 12.

Табл. 12. Уникальные параметры графического примитива **Полоса прокрутки**

№	Название параметра	Описание параметра
1.	Значение	Целочисленная переменная, чье значение управляется (изменяется) элементом.
2.	Минимальное значение	Целочисленная переменная, определяющая минимальное значение переменной, управляемой элементом.
3.	Максимальное значение	Целочисленная переменная, определяющая максимальное значение переменной, управляемой элементом.
4.	Размер страницы	Числовая переменная, определяющая шаг изменения переменной, управляемой элементом (при нажатии на полосу прокрутки, не влияет на кнопки « Больше »/« Меньше »).
5.	Прокрутка выполнена	При наличии галочки, параметр Размер страницы игнорируется. При нажатии на любую точку полосы прокрутки, переменная элемента принимает соответствующее значение.
6.	Линейка	
6.1.	Ориентация	Ориентация элемента. Однозначно определяется соотношением высоты и ширины элемента: ширина > высота – горизонтальная ориентация высота > ширина – вертикальная ориентация
6.2.	Направление движения	Положение точки отсчета элемента. Зависит от ориентации. Возможные значения при горизонтальной ориентации: Слева направо/Справа налево Возможные значения при вертикальной ориентации: Сверху вниз/Снизу вверх

8.3.10. Бегунок

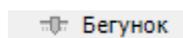


Рис. 8.35. Пиктограмма графического примитива **Бегунок**

Графический примитив **Бегунок** используется для управления значением переменной.



Рис. 8.36. Внешний вид элемента **Бегунок**

Описание типичных параметров графического примитива приведено в [табл. 2](#).

Уникальные параметры примитива описаны в табл. 13.

Табл. 13. Уникальные параметры графического примитива **Бегунок**

№	Название параметра	Описание параметра
1.	Переменная	Числовая переменная, чье значение управляется (изменяется) элементом.
2.	Размер страницы	Числовая переменная, определяющая шаг изменения переменной, управляемой элементом (при нажатии на элемент).
3.	Прокрутка выполнена	При наличии галочки, параметр Размер страницы игнорируется. При нажатии на любую точку бегунка, переменная элемента принимает соответствующее значение.
4.	Шкала	
4.1.	Показать шкалу	При наличии галочки, рядом с элементом отображается шкала с делениями.
4.2.	Начало шкалы	Точка начала шкалы (минимальное отображаемое значение переменной элемента).
4.3.	Конец шкалы	Точка конца шкалы (максимальное отображаемое значение переменной элемента).
4.4.	Основная шкала	Цена деления основной шкалы.
4.5.	Подшкала	Цена деления дополнительной шкалы.
4.6.	Формат шкалы	Спецификатор формата вывода делений шкалы.
5.	Линейка	
5.1.	Тип диаграммы	Место отображения шкалы элемента. Шкала отображается только при наличии галочки в пп. 4.1. Зависит от ориентации элемента: Возможные значения при горизонтальной ориентации: Слева/Справа/Слева и справа Возможные значения при вертикальной ориентации: Сверху/Снизу/Сверху и снизу
5.2.	Ориентация	Ориентация элемента. Однозначно определяется соотношением высоты и ширины элемента: ширина > высота – горизонтальная ориентация высота > ширина – вертикальная ориентация
5.3.	Направление движения	Положение точки отсчета элемента. Зависит от ориентации. Возможные значения при горизонтальной ориентации: Слева направо/Справа налево Возможные значения при вертикальной ориентации: Сверху вниз/Снизу вверх

8.3.11. Управление вращением

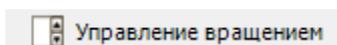


Рис. 8.37. Пиктограмма графического примитива **Управление вращением**

Графический примитив **Управления вращением** используется отображения значения переменной и его изменения кнопками «Больше»/ «Меньше».

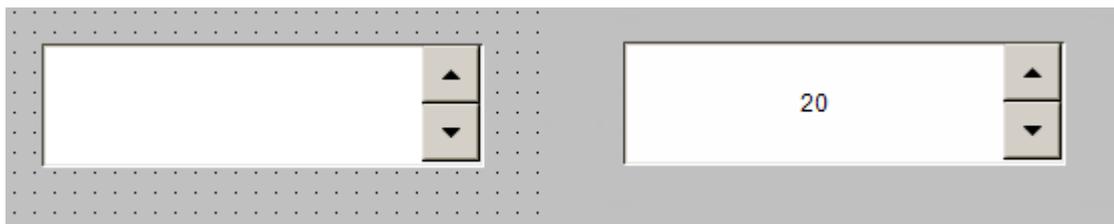


Рис. 8.38. Внешний вид элемента Управление вращением

Описание типичных параметров графического примитива приведено в [табл. 2](#).

Уникальные параметры примитива описаны в табл. 14.

Табл. 14. Уникальные параметры графического примитива **Управление вращением**

№	Название параметра	Описание параметра
1.	Переменная	Числовая переменная, чье значение отображается и управляется (изменяется) элементом.
2.	Числовой формат	Спецификатор формата вывода переменной, управляемой элементом.
3.	Интервал	Числовая переменная, определяющая шаг изменения переменной, управляемой элементом.
4.	Value Range	
4.1.	Минимальное значение	Переменная, определяющая нижнюю границу значений переменной, управляемой элементом.
4.2.	Максимальное значение	Переменная, определяющая верхнюю границу значений переменной, управляемой элементом.

8.3.12. Невидимый вход

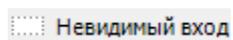


Рис. 8.39. Пиктограмма графического примитива **Невидимый вход**

Графический примитив **Невидимый вход** представляет собой невидимый элемент, способный реагировать на нажатие.

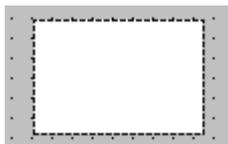


Рис. 8.40. Внешний вид элемента **Невидимый вход**

Описание типичных параметров графического примитива приведено в [табл. 2](#).

8.3.13. Радио-кнопка

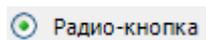


Рис. 8.41. Пиктограмма графического примитива **Радио-кнопка**

Графический примитив **Радио-кнопка** представляет собой набор кнопок, выбор одной из которых определяет значение привязанной к элементу переменной. В каждый момент времени может быть выбрана только одна кнопка.

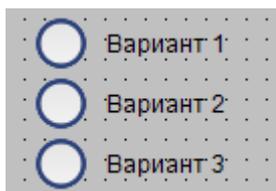


Рис. 8.42. Внешний вид элемента **Радио-кнопка**

Описание типичных параметров графического примитива приведено в [табл. 2](#).

Уникальные параметры примитива описаны в табл. 15.

Табл. 15. Уникальные параметры графического примитива **Радио-кнопка**

№	Название параметра	Описание параметра
1.	Переменная	Целочисленная переменная, значение которой зависит от выбранной в данный момент кнопки элемента.
2.	Число столбцов	Число столбцов кнопок элемента.
3.	Порядок радио-кнопки	<p>Выбору каждой из кнопок соответствует значение целочисленной переменной. Значение "0" всегда соответствует выбору верхней левой кнопки. Остальные значения определяются данным параметром:</p> <p>Сверху вниз: кнопкам присваиваются значения сверху вниз (т.е. "1" – вторая кнопка первого столбца, "n" – последняя кнопка первого столбца, "n+1" – первая кнопка второго столбца и т.д.)</p>  <p>Слева направо: кнопкам присваиваются значения слева направо (т.е. "1" – вторая кнопка второго столбца, "n" – первая кнопка последнего столбца, "n+1" – вторая кнопка первого столбца и т.д.)</p> 
4.	Размер фрейма	Расстояние от края элемента до первого столбца кнопок в пикселях. Изменение значения параметра не смещает кнопку, а увеличивает/уменьшает ее размер.
5.	Высота строки	Высота одной строки кнопок в пикселях.
6.	Свойства текста	
6.1.	Использование	Позволяет использовать для текста элементов списка либо настройки стиля визуализации , либо индивидуальные установки (см. пп. 7.2).
6.2.	Индивидуальные свойства текста	(данные параметры доступны только при выборе значения Индивидуальные установки в пп. 7.1). Свойства текста элементов списка. Описание параметров приведено в табл. 2 , пп. 6.

	Параметры радио-кнопки/Радио-кнопка	
7.	В этой вкладке определяется число кнопок элемента. Для создания новой кнопки используется кнопка Создать новый , для удаления существующей – Удалить . Каждая из кнопок обладает следующими настройками:	
7.1.	Текст	Текст кнопки.
7.2.	Подсказка	Текст подсказки кнопки.
7.3.	Отступы строки в пикселях	Расстояние между строками кнопок в пикселях..

8.3.14. Кнопка-флажок

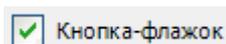


Рис. 8.43. Пиктограмма графического примитива **Кнопка-флажок**

Графический примитив **Кнопка-флажок** используется для переключения значения логической (типа **BOOL**) переменной.



Рис. 8.44. Внешний вид элемента **Кнопка-флажок**

Описание типичных параметров графического примитива приведено в [табл. 2](#).

Уникальные параметры примитива описаны в табл. 16.

Табл. 16. Уникальные параметры графического примитива **Кнопка-флажок**

№	Название параметра	Описание параметра
1.	Переменная	Логическая переменная, чье значение отображается и переключается кнопкой. При значении FALSE – флажок отсутствует, при значении TRUE – присутствует.
2.	Размер фрейма	Расстояние от края элемента до кнопки-флажка в пикселях. Изменение значения параметра не смещает кнопку, а увеличивает/уменьшает ее размер.

8.3.15. Индикатор выполнения

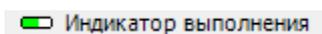


Рис. 8.45. Пиктограмма графического примитива **Индикатор выполнения**

Графический примитив **Индикатор выполнения** используется для графического отображения значения привязанной переменной. Вертикальная ориентация для элемента не предусмотрена.



Рис. 8.46. Внешний вид элемента **Индикатор выполнения**

Описание типичных параметров графического примитива приведено в [табл. 2](#).

Уникальные параметры примитива описаны в табл. 16.

Табл. 16. Уникальные параметры графического примитива **Индикатор выполнения**

№	Название параметра	Описание параметра
1.	Переменная	Числовая переменная, чье значение отображается элементом.
2.	Минимальное значение	Нижняя граница значений, отображаемых элементом.
3.	Максимальное значение	Верхняя граница значений, отображаемых элементом.
4.	Стиль	Стиль заполнения элемента. Возможные значения: Блоки/Линейка

8.4. Элементы управления измерением

8.4.1. Отображение линейки

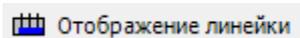


Рис. 8.47. Пиктограмма графического примитива **Отображение линейки**

Графический примитив **Отображение линейки** используется для визуализации численной переменной с помощью полосы заполнения совместно со шкалой значений.

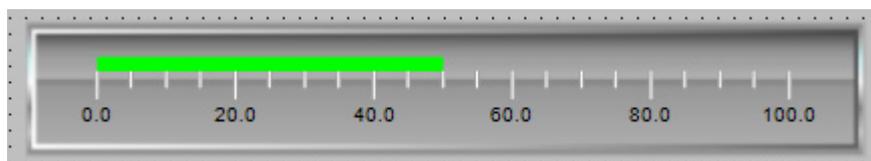


Рис. 8.48. Внешний вид элемента **Отображение линейки**

Описание типичных параметров графического примитива приведено в [табл. 2](#).

Уникальные параметры примитива описаны в табл. 17.

Пример работы с элементом приведен в [п. 11.2.6](#).

Табл. 17. Уникальные параметры графического примитива **Отображение линейки**

№	Название параметра	Описание параметра
1.	Значение	Числовая переменная, значение которой отображается элементом.
2.	Фон	
2.1.	Цвет изображения	Цвет фона элемента. Возможные значения: Yellow/Red/Green/Blue/Gray
2.2.	<i>Собственное изображение</i>	
2.2.1.	Изображение	Графический файл из пула изображений , используемый в качестве фонового изображения элемента.
2.2.2.	Цвет прозрачности	Указанный здесь цвет будет прозрачным при отображении фонового изображения элемента. Это можно использовать в тех случаях, когда исходное изображение имеет непрозрачный фон.
2.3.	Оптимизация нанесения	При наличии галочки, фоновое изображение не перерисовывается при изменении динамических частей элемента.
3.	Линейка	
3.1.	Тип диаграммы	Взаимное расположение полосы заполнения (линейки) элемента и шкалы. Возможные значения: Шкала возле линейки/Шкала в линейке/Линейка в шкале/Нет шкалы
3.2.	Ориентация	Ориентация элемента. Однозначно определяется соотношением высоты и ширины элемента: ширина > высота – горизонтальная ориентация высота > ширина – вертикальная ориентация
3.3.	Направление движения	Положение начала шкалы элемент (см. пп. 4.1). Зависит от ориентации элемента. Возможные значения при горизонтальной ориентации: Слева направо/Справа налево Возможные значения при вертикальной ориентации: Сверху вниз/Снизу вверх
4.	Шкала	
4.1.	Начало шкалы	Точка начала шкалы (минимальное отображаемое значение переменной элемента).
4.2.	Конец шкалы	Точка конца шкалы (максимальное отображаемое значение переменной элемента).

4.3.	Основная шкала	Цена деления основной шкалы.
4.4.	Подшкала	Цена деления дополнительной шкалы.
4.5.	Ширина линии шкалы	Ширина делений шкалы в пикселях.
4.6.	Цвет шкалы	Цвет делений шкалы элемента.
4.7.	Шкала в 3D	При наличии галочки, шкала отображается с тенью.
4.8.	Фрейм элемента	При наличии галочки, вокруг элемента отображается контур.
5.	Метка	
5.1.	Единица	Текст, отображаемый элементом. Обычно используется для указания единицы измерения.
5.2.	Шрифт	Шрифт текста элемента и цены делений шкалы.
5.3.	Формат шкалы	Спецификатор формата вывода переменной, отображаемой элементом.
5.4.	Максимальная ширина текста меток	Максимальная ширина цены делений в пикселях. Указанное здесь значение используется только при горизонтальной ориентации элемента.
5.5.	Высота текста меток	Максимальная высота цены делений в пикселях. Указанное здесь значение используется только при вертикальной ориентации элемента.
5.6.	Цвет шрифта	Цвет текста элемента и цены делений шкалы.
6.	Расположение	
6.1.	Горизонтальный сдвиг	Расстояние между шкалой и левой стороной элемента в пикселях. Значение 0 соответствует расстоянию по умолчанию.
6.2.	Вертикальный сдвиг	Расстояние между шкалой и верхней стороной элемента в пикселях. Значение 0 соответствует расстоянию по умолчанию.
6.3.	Horizontal Scaling	Коэффициент масштабирования элемента по горизонтали. Положительным значениям соответствует сжатие, отрицательным – растягивание.
6.4.	Vertical Scaling	Коэффициент масштабирования элемента по вертикали. Положительным значениям соответствует сжатие, отрицательным – растягивание.
7.	Цвета	
7.1.	Цвет графика	Цвет полосы заполнения элемента.
7.2.	Фон линейки	При наличии галочки, вокруг полосы заполнения отображается фоновый прямоугольник.
7.3.	Цвет фрейма	Цвет контура элемента. Используется при наличии галочки в пп. 4.8. Обратите внимание , что данный цвет относится к контуру всего элемента, а не к контуру полосы заполнения (см. пп. 7.2).
7.4.	Переключить весь цвет	При наличии галочки, цвет всей полосы заполнения определяется цветом цветовой области, в которой находится текущее значение переменной элемента (см. пп. 7.7).
7.5.	Использовать градиентный цвет	При наличии галочки, цвет полосы заполнения отображается с градиентом.
7.6.	Маркеры цветовых диапазонов	Тип отображения границ цветовых областей (см. пп. 7.7). Возможные значения: Нет маркеров/Маркер вперед/Маркер назад

7.7.	Цветовые области	
	Цветовые области представляю собой зоны полосы заполнения элемента, отличающиеся выбранным цветом. Цветовые области отображаются по мере достижения переменной элемента соответствующих значений (при этом при достижении следующей области, предыдущая продолжает отображаться). Обратите внимание на настройку Переключить весь цвет (см. пп. 7.4).	
7.7.1	Начало области	Точка начала цветовой области на полосе заполнения.
7.7.2.	Конец области	Точка конца цветовой области на полосе заполнения.
7.7.3.	Цвет	Цвет области.

8.4.2. Стрелочный индикатор (90°/180°/360°)

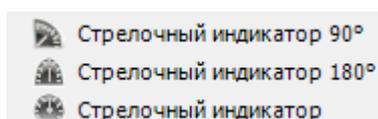


Рис. 8.49. Пиктограммы графического примитива **Стрелочный индикатор**

Графический примитив **Стрелочный индикатор** используется для отображения значения переменной.

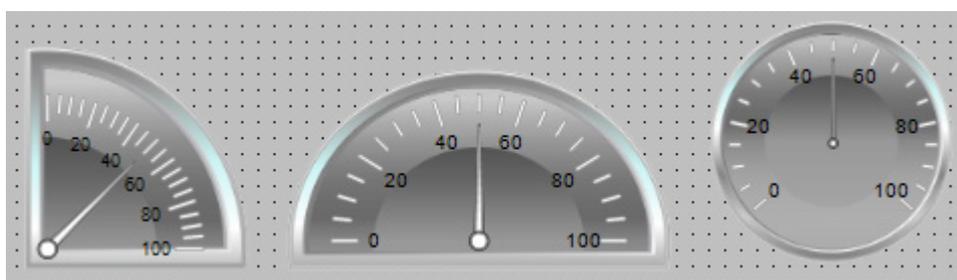
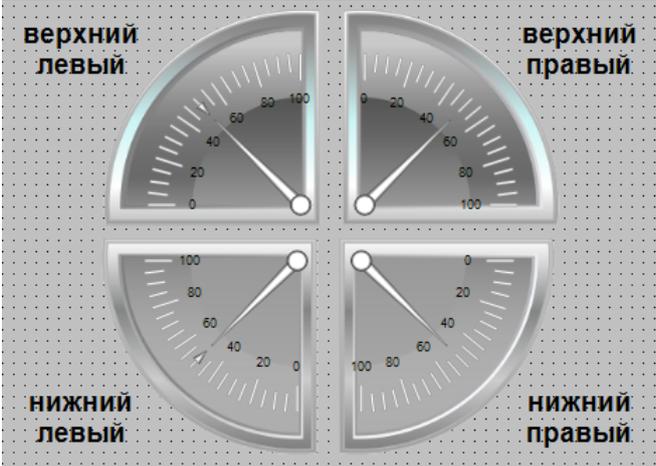


Рис. 8.50. Внешний вид элемента **Стрелочный индикатор**

Описание типичных параметров графического примитива приведено в [табл. 2](#).

Уникальные параметры примитива описаны в табл. 18.

Табл. 18. Уникальные параметры графического примитива **Стрелочный индикатор**

№	Название параметра	Описание параметра
1.	Значение	Числовая переменная, значение которой отображается элементом.
2.	Фон	
2.1.	Цвет изображения	Цвет фона элемента. Возможные значения: Yellow/Red/Green/Blue/Gray
2.2.	Собственное изображение	
2.2.1.	Изображение	Графический файл из пула изображений, используемый в качестве фонового изображения элемента.
2.2.2.	Цвет прозрачности	Указанный здесь цвет будет прозрачным при отображении фонового изображения элемента. Это можно использовать в тех случаях, когда исходное изображение имеет непрозрачный фон.
2.3.	Оптимизация нанесения	При наличии галочки, фоновое изображение не перерисовывается при изменении динамических частей элемента.
3.	Стрелка	
3.1.	Тип стрелки	Тип стрелки элемента. Возможные значения: Нормальная стрелка/Тонкая стрелка/Широкая стрелка/Тонкая игла/Тонкая 3D стрелка/Тонкая 3D игла
3.2.	Цвет	Цвет стрелки элемента.
3.3. (90°/ 180°)	Угловой диапазон	<p>Данный параметр доступен только для графических примитивов Стрелочный индикатор 90° и Стрелочный индикатор 180° и определяет их внешний вид.</p> <p>Возможные значения для графического примитива Стрелочный индикатор 90°:</p> 

		<p>Возможные значения для графического примитива Стрелочный индикатор 180°:</p> 
3.4. (360°)	Начало стрелки	<p>Данный параметр доступен только для графического примитива Стрелочный индикатор 360°.</p> <p>Угол между положительным направлением горизонтальной оси и началом шкалы элемента в градусах. Положительное значение соответствует движению против часовой стрелки, отрицательное – по часовой.</p>
3.5. (360°)	Конец стрелки	<p>Данный параметр доступен только для графического примитива Стрелочный индикатор 360°.</p> <p>Угол между положительным направлением горизонтальной оси и концом шкалы элемента в градусах. Положительное значение соответствует движению против часовой стрелки, отрицательное – по часовой.</p>
3.6.	Дополнительная стрелка	<p>При наличии галочки, конец стрелки дополнительно выделяется на шкале.</p>

4.	Шкала	
4.1.	Позиция подшкалы	Положение дополнительной шкалы элемента. Возможные значения: Снаружи/Внутри
4.2.	Тип шкалы	Тип делений на шкале элемента. Возможные значения: Линии/Точки/Квадраты
4.3.	Начало шкалы	Точка начала шкалы (минимальное отображаемое значение переменной элемента).
4.4.	Конец шкалы	Точка конца шкалы (максимальное отображаемое значение переменной элемента).
4.5.	Основная шкала	Цена деления основной шкалы.
4.6.	Подшкала	Цена деления дополнительной шкалы.
4.7.	Ширина линии шкалы	Ширина делений шкалы в пикселях.
4.8.	Цвет шкалы	Цвет делений шкалы элемента.
4.9.	Шкала в 3D	При наличии галочки, шкала отображается с тенью.
4.10.	Показать шкалу	При отсутствии галочки, шкала элемента не отображается.
4.11.	Фрейм внутри	При наличии галочки, нижняя сторона шкалы элемента выделяется контуром.
4.12.	Фрейм снаружи	При наличии галочки, верхняя сторона шкалы элемента выделяется контуром.
5.	Метка	
5.1.	Метка	Тип расположения шкалы элемента. Возможны значения: Внутри – шкала располагается ближе к краю элемента Снаружи – шкала располагается ближе к центру элемента
5.2.	Единица	Текст, отображаемый элементом. Обычно используется для указания единицы измерения.
5.3.	Шрифт	Шрифт текста элемента и цены делений шкалы.
5.4.	Формат шкалы	Спецификатор формата вывода переменной, отображаемой элементом.
5.5.	Макс. ширина текста меток	Максимальная ширина цены делений в пикселях.
5.6.	Высота текста меток	Максимальная высота цены делений в пикселях.
5.7.	Цвет шрифта	Цвет текста элемента и цены делений шкалы.
6.	Расположение	
	При выборе Значений стиля по умолчанию будут использоваться настройки выбранного стиля визуализации.	
6.1.	Движение иглы	Длина стрелки элемента в пикселях.
6.2.	Сдвиг метки	Расстояние по вертикали между началом стрелки элемента и значениями цены делений шкалы в пикселях.
6.3.	Сдвиг единицы	Расстояние по вертикали между началом стрелки элемента и единицей измерения (см. пп. 5.2) в пикселях.
6.4. (360°)	Сдвиг начала	Данный параметр доступен только для графического примитива Стрелочный индикатор 360° . Расстояние между шкалой и фоновым изображением.

6.5.	Перемещение шкалы	<p>Данный параметр доступен только при выборе фонового изображения (см. пп. 2.2.1).</p> <p>Расстояние между шкалой и центром элемента в пикселях.</p>
6.6.	Длина шкалы	<p>Данный параметр доступен только при выборе фонового изображения (см. пп. 2.2.1).</p> <p>Длина (в проекции на горизонтальную ось) шкалы элемента в пикселях.</p>
7.	Цвета	
7.1.	Долговременные цветовые области	При наличии галочки, цветовые области (см. пп. 7.2) отображаются независимо от значения переменной элемента.
7.2.	Цветовые области	
	<p>Цветовые области представляю собой зоны шкалы элемента, отличающиеся выбранным цветом. Цветовая область отображаются только в то время, когда переменная находится в интервале ее значений. Если необходимо отображать цветовые области постоянно, то следует воспользоваться параметром Долгосрочные цветовые области (см. пп. 7.1).</p>	
7.2.1	Начало области	Точка начала цветовой области на шкале элемента.
7.2.2.	Конец области	Точка конца цветовой области на шкале элемента.
7.2.3.	Цвет	Цвет области.

8.4.3. Потенциометр

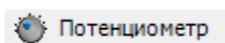


Рис. 8.51. Пиктограмма графического примитива **Потенциометр**

Графический примитив **Потенциометр** в целом аналогичен примитиву [Стрелочный индикатор 360°](#), но помимо отображения значения переменной, позволяет управлять им.

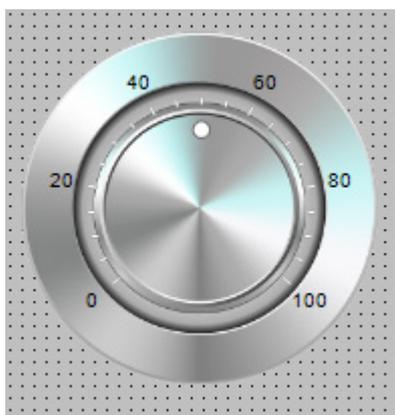


Рис. 8.52. Внешний вид элемента **Потенциометр**

Описание типичных параметров графического примитива приведено в табл. 2.

Уникальные параметры примитива описаны в табл. 18 (поскольку они совпадают с уникальными настройками примитива Стрелочный индикатор). Отличаются только два параметра:

1. **Переменная** – выбор переменной, которая будет отображаться и управляться элементом;
2. **Стрелка - Тип стрелки** – в отличие от **Стрелочного индикатора**, возможные значения этого параметра в данном случае: **Круг/Маленькая стрелка**.

Пример работы с элементом приведен в [п. 11.2.6](#).

8.4.4. Гистограмма

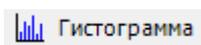


Рис. 8.53. Пиктограмма графического примитива **Гистограмма**

Графический примитив **Гистограмма** используется для отображения значений одномерных массивов числовых переменных.

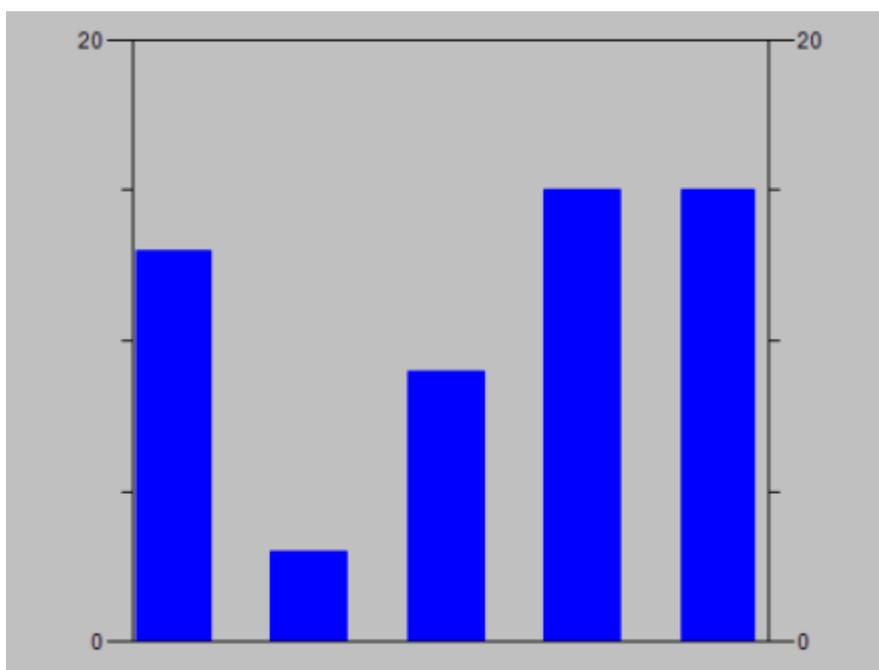


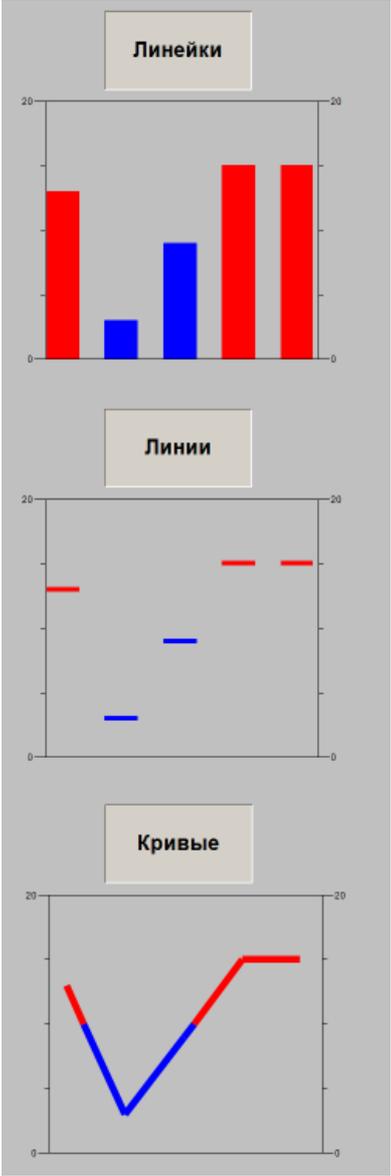
Рис. 8.54. Внешний вид элемента **Гистограмма**

Описание типичных параметров графического примитива приведено в табл. 2.

Уникальные параметры примитива описаны в табл. 19.

Пример работы с элементом приведен в [п. 11.2.7.](#)

Табл. 19. Уникальные параметры графического примитива **Гистограмма**

№	Название параметра	Описание параметра
1.	Массив данных	Одномерный массив числовых переменных, отображаемый элементом.
2.	Поддиапазон массива	
2.1.	Использовать поддиапазон	При наличии галочки, отображается только часть массива, выбранного в пп. 1.
2.2.	Начальный индекс	Первый индекс отображаемого поддиапазона массива.
2.3.	Конечный индекс	Последний индекс отображаемого поддиапазона массива.
3.	Тип отображения	<p>Тип отображения значений массива. Возможные значения: Линейки/Линии/Кривые</p> 

4.	Ширина линии	Данный параметр доступен только для Типа отображения (пп. 3) Кривые и определяет ширину линий в пикселях.
5.	Показывать горизонтальные линии	При наличии галочки, на элементе отображаются горизонтальные линии сетки (по делениями основной шкалы, см. пп. 7.3).
6.	Относительная ширина линейки	Определяет ширину столбцов/линий гистограммы. Может принимать значения от 1 до 100. Значению 100 соответствует отсутствие зазоров между столбцами/линиями. Данный параметр не применяется при Типе отображения (см.пп.3) Кривые .
7.	Шкала	
7.1.	Начало шкалы	Точка начала шкалы (минимальное отображаемое значение переменной элемента).
7.2.	Конец шкалы	Точка конца шкалы (максимальное отображаемое значение переменной элемента).
7.3.	Основная шкала	Цена деления основной шкалы.
7.4.	Подшкала	Цена деления дополнительной шкалы.
7.5.	Цвет шкалы	Точка начала шкалы (минимальное отображаемое значение переменной элемента).
7.6.	Base line	Нулевая линия гистограммы. По умолчанию равно 0 (т.е. является нижней стороной гистограммы). При задании здесь какого-либо числа, на гистограмме отображается разность между значениями массива (см. пп. 1) и значением base line. Если получившее число положительно, оно отображается в верхней полуплоскости гистограммы, если отрицательно – то в нижней.
8.	Метка	
8.1.	Единица	Текст, отображаемый элементом. Обычно используется для указания названия гистограммы и единиц измерения.
8.2.	Шрифт	Шрифт текста элемента.
8.3.	Формат шкалы	Спецификатор формата вывода переменной, отображаемой элементом.
8.4.	Максимальная ширина текста меток	Максимальная ширина текста элемента в пикселях.
8.5.	Высота текста меток	Максимальная высота текста элемента в пикселях.
8.6.	Цвет шрифта	Цвет текста элемента.
9.	Цвета	
9.1.	Цвет графика	Цвет столбцов/линий гистограммы.
9.2.	Цвет тревоги	
9.2.1.	Значение тревоги	Значение, которое определяет смену цвета столбцов/линий гистограммы в случае выполнения условия тревоги (см. пп. 9.2.2).

9.2.2.	Условие тревоги	Условие, при выполнении которого происходит смена цвета столбцов/линий гистограммы на цвет тревоги. Возможные значения: Больше – значение элемента массива превышает значение тревоги (см. пп. 9.2.1) Меньше – значение элемента массива не достигает значения тревоги (см. пп. 9.2.1)
9.2.3.	Цвет тревоги	Цвет, в который окрашиваются столбцы/линии гистограммы при выполнении условия тревоги (см. пп. 9.2.2).
9.3.	Использовать цветовые области	При наличии галочки, на гистограмме отображаются цветовые области (см. пп. 9.4).
9.4.	Цветовые области	
9.4.	Цветовые области представляю собой зоны гистограммы , отличающиеся выбранным цветом. Цветовые области отображаются независимо от значений переменных массива гистограммы.	
9.4.1.	Начало области	Точка начала цветовой области по вертикальной оси гистограммы.
9.4.2.	Конец области	Точка конца цветовой области по вертикальной оси гистограммы.
9.4.3.	Цвет	Цвет области.

8.5. Индикаторы/Переключатели/Изображения

8.5.1. Переключатель изображения

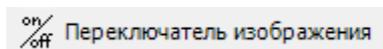


Рис. 8.55. Пиктограмма графического примитива **Переключатель изображения**

Графический примитив **Переключатель изображения** используется для отображения двух графических файлов, соответствующих состоянию **TRUE/FALSE** переменной типа **BOOL**, привязанной к элементу.

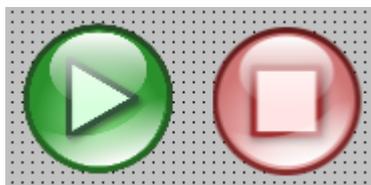


Рис. 8.56. Внешний вид элемента **Переключатель изображения** (состояния **TRUE/FALSE**).

Описание типичных параметров графического примитива приведено в [табл. 2](#).

Уникальные параметры примитива описаны в табл. 20.

Пример работы с элементом приведен в [п. 11.2.8](#).

Табл. 20. Уникальные параметры графического примитива **Переключатель изображения**

№	Название параметра	Описание параметра
1.	Переменная	Переменная типа BOOL , которая определяет текущее изображение, отображаемое элементом.
2.	Параметры изображения	
2.1.	Изображение вкл.	Графический файл из пула изображений , отображаемый элементом в случае, если его переменная принимает значение TRUE .
2.2.	Изображение выкл.	Графический файл из пула изображений , отображаемый элементом в случае, если его переменная принимает значение FALSE .
2.3.	Прозрачный	При наличии галочки, цвет, указанный в параметре Прозрачный цвет не отображается элементом. Это можно использовать в тех случаях, когда исходное изображение имеет непрозрачный фон, отличный от фона экрана визуализации.
2.4.	Прозрачный цвет	Указанный здесь цвет при наличии галочки Прозрачный не будет отображаться.
2.5.	Image Pushed	Данный параметр доступен только при выборе в Поведении элемента (см. пп. 3) значения Переключатель изображения . Здесь можно указать графический файл из пула изображений , который будет отображаться элементом при нажатии на него до момента отпускания. Может использоваться при необходимости отображать промежуточное состояние элемента.
2.5.	Изотропный тип	Тип масштабирования изображения при изменении размеров элемента. Возможные значения: Изотропия – при изменении размеров элемента, изображение масштабируется с сохранением соотношения сторон; Анизотропный – при изменении размеров элемента, изображение масштабируется без сохранения соотношения сторон; Фиксированный – при изменении размеров элемента, изображение сохраняет свой размер.
2.6.	Горизонтальное выравнивание	Выравнивание изображения относительно элемента по горизонтали. Возможные значения: Лев о/ По центру / Прав о
2.7.	Вертикальное выравнивание	Выравнивание изображения относительно элемента по вертикали. Возможные значения: Верх / По центру / Низ

3.	Поведение элемента	<p>Определяет действие, необходимое для изменения значения переменной и изображения элемента:</p> <p>Клавиша изображения – зажатие курсора при наведении на элемент. После того, как курсор будет отпущен, значение переменной и изображение сменяются на исходные;</p> <p>Переключатель изображения – однократное нажатие курсора при наведении на элемент.</p>
4.	Переключить на FALSE	<p>Данный параметр доступен только при выборе в Поведении элемента значения Клавиша изображения.</p> <p>По умолчанию (при отсутствии галочки) на время зажатия курсора при наведении его на элемент, переменная, привязанная к элементу, меняет свое значение с FALSE на TRUE. Если переменная уже имеет значение TRUE (например, это ее начальное значение), то оно будет переключено на FALSE.</p> <p>При наличии галочки в данном параметре, переменная будет менять значение с TRUE на FALSE. Если переменная уже имеет значение TRUE (например, это ее начальное значение), то оно будет переключено на FALSE.</p>

8.5.2. Индикатор

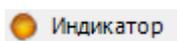


Рис. 8.57. Пиктограмма графического примитива **Индикатор**

Графический примитив **Индикатор** используется для отображения состояния переменной типа **BOOL**.



Рис. 8.58. Внешний вид элемента **Индикатор** (состояния **TRUE/FALSE**).

Описание типичных параметров графического примитива приведено в [табл. 2](#).

Уникальные параметры примитива описаны в табл. 21.

Пример работы с элементом приведен в [п. 11.2.8](#).

Табл. 21. Уникальные параметры графического примитива **Индикатор**

№	Название параметра	Описание параметра
1.	Переменная	Переменная типа BOOL , значение которой отображается элементом. Значению TRUE соответствует горящий индикатор, значение FALSE – потухший.
2.	Параметры изображения	
2.1.	Прозрачный	При наличии галочки, цвет, указанный в параметре Прозрачный цвет не отображается элементом. Это можно использовать в тех случаях, когда исходное изображение имеет непрозрачный фон, отличный от фона экрана визуализации.
2.2.	Прозрачный цвет	Указанный здесь цвет при наличии галочки Прозрачный не будет отображаться.
2.3.	Изотропный тип	<p>Тип масштабирования изображения при изменении размеров элемента. Возможные значения:</p> <p>Изотропия – при изменении размеров элемента, изображение масштабируется с сохранением соотношения сторон;</p> <p>Анизотропный – при изменении размеров элемента, изображение масштабируется без сохранения соотношения сторон;</p> <p>Фиксированный – при изменении размеров элемента, изображение сохраняет свой размер.</p>
2.4.	Горизонтальное выравнивание	Выравнивание изображения относительно элемента по горизонтали. Возможные значения: Лево/По центру/Право
2.5.	Вертикальное выравнивание	Выравнивание изображения относительно элемента по вертикали. Возможные значения: Верх/По центру/Низ
3.	Фон	
3.1.	Изображение	Цвет индикатора. Возможные значения: Yellow/Red/Green/Blue/Gray

8.5.3. Переключатели/Выключатели

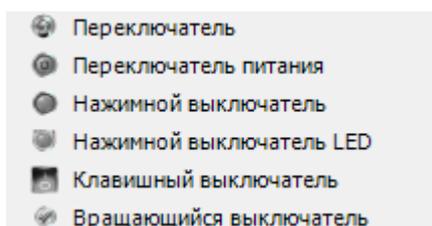


Рис. 8.59. Пиктограммы графических примитивов **Переключатели/Выключатели**

Семейство графических примитивов **Переключатели/Выключатели** используется для управления состояниями переменных типа **BOOL**.

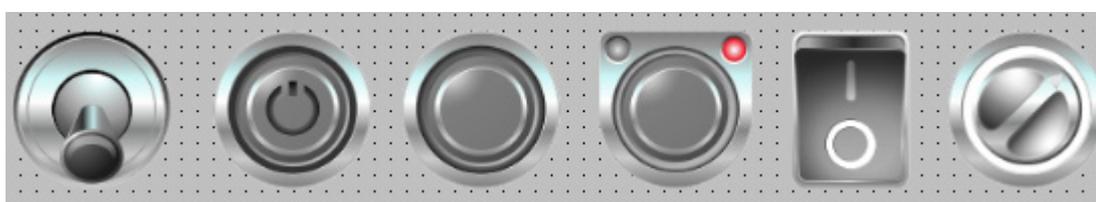


Рис. 8.60. Внешний вид элементов **Переключатели/Выключатели**

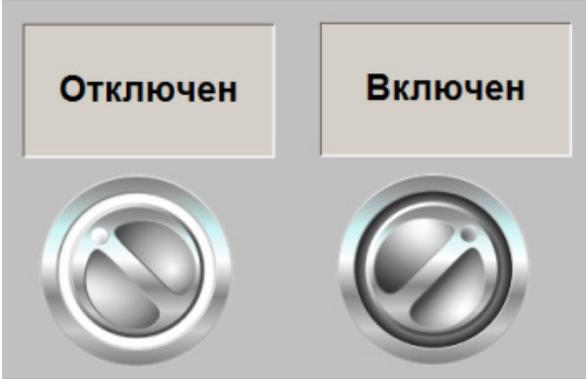
Описание типичных параметров графического примитива приведено в [табл. 2](#).

Уникальные параметры примитива описаны в табл. 22.

Пример работы с элементом приведен в [п. 11.2.8](#).

Табл. 22. Уникальные параметры графических примитивов **Переключатели/Выключатели**

№	Название параметра	Описание параметра
1.	Переменная	Переменная типа BOOL , которая управляется элементом.
2.	Параметры изображения	
2.1.	Прозрачный	При наличии галочки, цвет, указанный в параметре Прозрачный цвет не отображается элементом. Это можно использовать в тех случаях, когда исходное изображение имеет непрозрачный фон, отличный от фона экрана визуализации.
2.2.	Прозрачный цвет	Указанный здесь цвет при наличии галочки Прозрачный не будет отображаться.
2.3.	Изотропный тип	Тип масштабирования изображения при изменении размеров элемента. Возможные значения: Изотропия – при изменении размеров элемента, изображение масштабируется с сохранением соотношения сторон; Анизотропный – при изменении размеров элемента, изображение масштабируется без сохранения соотношения сторон; Фиксированный – при изменении размеров элемента, изображение сохраняет свой размер.
2.4.	Горизонтальное выравнивание	Выравнивание изображения относительно элемента по горизонтали. Возможные значения: Лев о/ По центру / Прав о
2.5.	Вертикальное выравнивание	Выравнивание изображения относительно элемента по вертикали. Возможные значения: Верх / По центру / Низ
3.	Поведение элемента	Определяет действие, необходимое для изменения значения переменной и изображения элемента: Клавиша изображения – зажатие курсора при наведении на элемент. После того, как курсор будет отпущен, значение переменной и изображение сменятся на исходные; Переключатель изображения – однократное нажатие курсора при наведении на элемент.
4.	Переключить на FALSE	Данный параметр доступен только при выборе в Поведении элемента значения Клавиша изображения . По умолчанию на время зажатия курсора при наведении его на элемент, переменная, привязанная к элементу, меняет свое значение с FALSE на TRUE . Если переменная

		<p>уже имеет значение TRUE (например, это ее начальное значение), то оно будет переключено на FALSE.</p> <p>При наличии галочки в данном параметре, переменная будет менять значение с TRUE на FALSE. Если переменная уже имеет значение TRUE (например, это ее начальное значение), то оно будет переключено на FALSE.</p>
5.	Ориентация	<p>Данный параметр доступен только для примитива Вращающийся выключатель.</p> <p>Определяет внешний вид элемента. Возможные значения: Наверху/С боку</p> 
6.	Изменение цвета	<p>Данный параметр доступен только для примитива Вращающийся выключатель.</p> <p>При наличии галочки, в состоянии включен (TRUE) элемент меняет свой цвет.</p> 
7.	Фон	
7.1.	Изображение	<p>Цвет элемента. Возможные значения: Yellow/Red/Green/Blue/Gray</p>

8.6. Специальные элементы управления

8.6.1. Трассировка

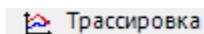


Рис. 8.61. Пиктограмма графического примитива **Трассировка**

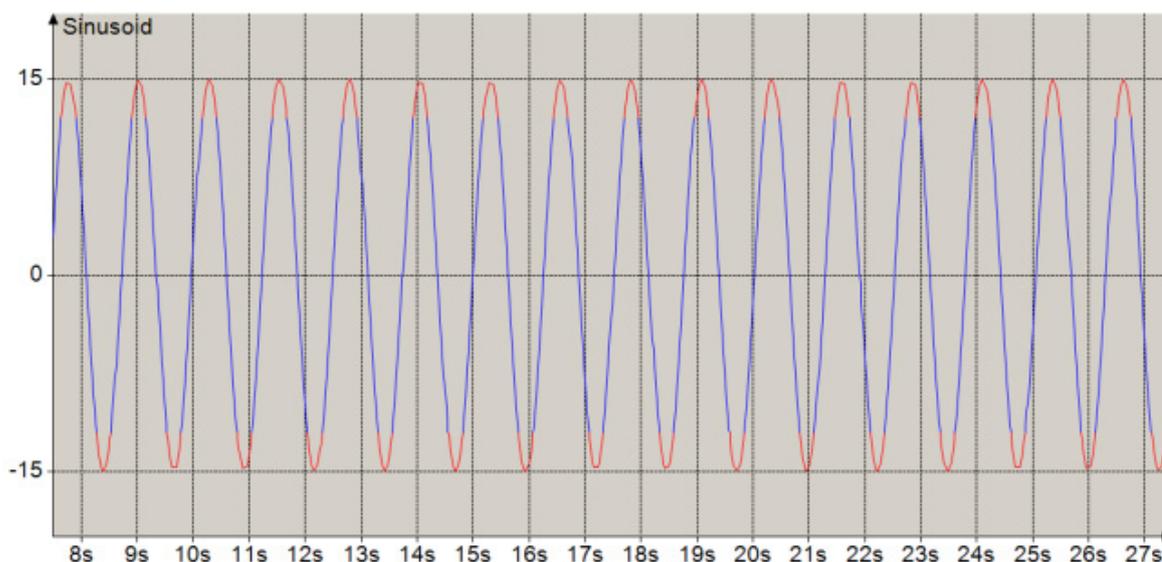


Рис. 8.62. Внешний вид элемента **Трассировка**

Графический примитив **Трассировка** используется для отображения графика в реальном времени. Буфер трассировки не должен превышать 2000 точек – т.е. при записи значений переменной с частотой 10 мс, возможно отобразить на графике значения за последние 19 секунд.

При добавлении графического элемента **Трассировка** на экран визуализации, автоматически создается компонент **Трассировка** с названием

<Имя экрана визуализации>_Tgrace<Номер трассировки>

Этот компонент доступен только при выделении графического элемента и содержит настройки его записи. Он также может быть добавлен в проект независимо от графического элемента (**Application – Добавить объект – Трассировка**) – и в этом случае будет отображаться на **Панели устройств**. Это может использоваться в тех случаях, когда трассировка применяется только в отладочных целях и нет необходимости отображать ее в визуализации. Настройки такой трассировки могут быть импортированы в соответствующий графический элемент.

Обратите внимание, что примитив **Трассировка** не рассчитан на хранение истории значений переменных – для решения этой задачи рекомендуется использовать графический примитив [Тренд](#).

Описание типичных параметров графического примитива приведено в [табл. 2](#).

Уникальные параметры примитива описаны в табл. 23.

Настройки компонента Трассировка приведены в табл. 24.

Пример работы с элементом приведен в [п. 11.2.9](#).

Табл. 23. Уникальные параметры графического примитива **Трассировка**

№	Название параметра	Описание параметра
1.	Показать курсор	При наличии галочки, после остановки трассировки по триггеру отображается курсор (метка со значением переменной).
2.	Заменить существующую трассировку	При наличии галочки, в случае наличия на целевом устройстве файла трассировки с названием, идентичным названию данной трассировки, при загрузке текущего проекта он будет перезаписан.
3.	Управляющие переменные	
	Этот пункт содержит переменные, которые используются для управления трассировкой. Обратите внимание , что их можно создать автоматически, выбрав из контекстного меню трассировки (открывается по нажатию ПКМ на элемент) пункт Вставить элементы управления трассировкой . Это приведет к автоматическому объявлению переменных в интерфейсе экрана визуализации, привязки их к данному элементу и добавлению на экран визуализации кнопок управления этими переменными.	
3.1.	Сброс триггера	Переменная типа BOOL . Когда она принимает значение TRUE , триггер трассировки (см. табл. 24, пп. 1) сбрасывается.
3.2.	Запустить трассировку	Переменная типа BOOL . Когда она принимает значение TRUE , остановленная трассировка запускается. Параметры 3.2-3.3 являются частным случаем триггера (см. табл. 24, пп. 1).
3.3.	Остановить трассировку	Переменная типа BOOL . Когда она принимает значение TRUE , трассировка останавливается.
3.4.	Сохранить трассировку в файл	
3.4.1.	Сохранить трассировку	Переменная типа BOOL . Когда она принимает значение TRUE , конфигурация трассировки сохраняется в файл.
3.4.2.	Имя файла	Переменная типа STRING , определяющая имя файла, в который сохраняется конфигурация трассировки.
3.5.	Загрузить трассировку из файла	
3.5.1.	Загрузить трассировку	Переменная типа BOOL . Когда она принимает значение TRUE , конфигурация трассировки загружается из файла.
3.5.2.	Имя файла	Переменная типа STRING , определяющая имя файла, из которого загружается конфигурация трассировки.

Для открытия конфигурации, необходимо либо выбрать пункт **Трассировка** в свойствах графического элемента, либо использовать команду **Конфигурация трассировки** в контекстном меню элемента (открывается по нажатию **ПКМ**).

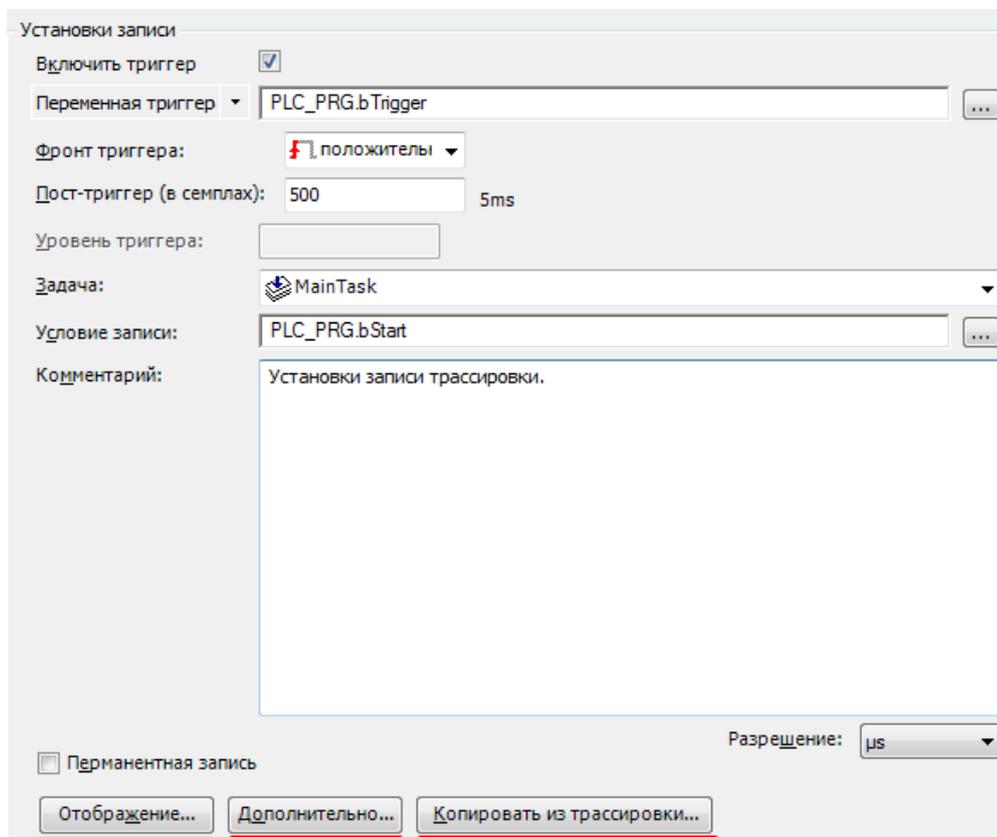


Рис. 8.63. Конфигурация трассировки - **Установки записи**

Табл. 24. Конфигурация элемента **Трассировка**

№	Название параметра	Описание параметра
1.	Установки записи	
1.1.	Включить триггер	При наличии галочки, становятся активными настройки триггера – переменной, которая позволяет запускать/останавливать трассировку.
1.2.	Переменная триггера	Переменная типа BOOL или целочисленная переменная, которая будет являться триггером трассировки.
1.3.	Фронт триггера	Условие остановки трассировки. При его выполнении, трассировка будет остановлена. Если условие перестанет выполняться, трассировка будет запущена вновь. Возможные значения: Положительный – трассировка останавливается, когда триггер типа BOOL принимает значение TRUE или целочисленный триггер в процессе увеличения достигает значения, указанного в пп. 1.5.

		<p>Отрицательный – трассировка останавливается, когда триггер типа BOOL принимает значение FALSE или целочисленный триггер в процессе уменьшения достигает значения, указанного в пп. 1.5.</p> <p>Оба – трассировка останавливается, когда триггер типа BOOL меняет свое значение или целочисленный триггер в процессе увеличения/уменьшения достигает значения, указанного в пп. 1.5.</p>
1.4.	Пост-триггер	Время после срабатывания триггера (в секундах, пересчитанное из кол-ва семплов), через которое трассировка останавливается.
1.5.	Уровень триггера	<p>Данный параметр доступен только в случае указания в пп. 1.2. целочисленной переменной.</p> <p>Значение, при достижении которого срабатывает целочисленный триггер (см. пп. 1.3).</p>
1.6.	Задача	Задача, цикл которой определяет частоту записи значений переменных трассировки.
1.7.	Условие записи	Переменная типа BOOL . Когда она принимает значение TRUE , трассировка начинает записываться (в отличие от переменных запуска/остановки и триггера, эта переменная не влияет на визуальное отображение трассировки)
1.8.	Комментарий	Комментарий к данной трассировке.
1.9.	Перманентная запись	При наличии галочки, конфигурация и буфер трассировки сохраняются в энергонезависимой памяти целевого устройства.
1.10.	Разрешение	<p>Минимальная единица отображения времени на трассировке.</p> <p>Возможные значения: миллисекунды/микросекунды</p> <p><i>Настоятельно рекомендуется</i> использовать микросекунды.</p>

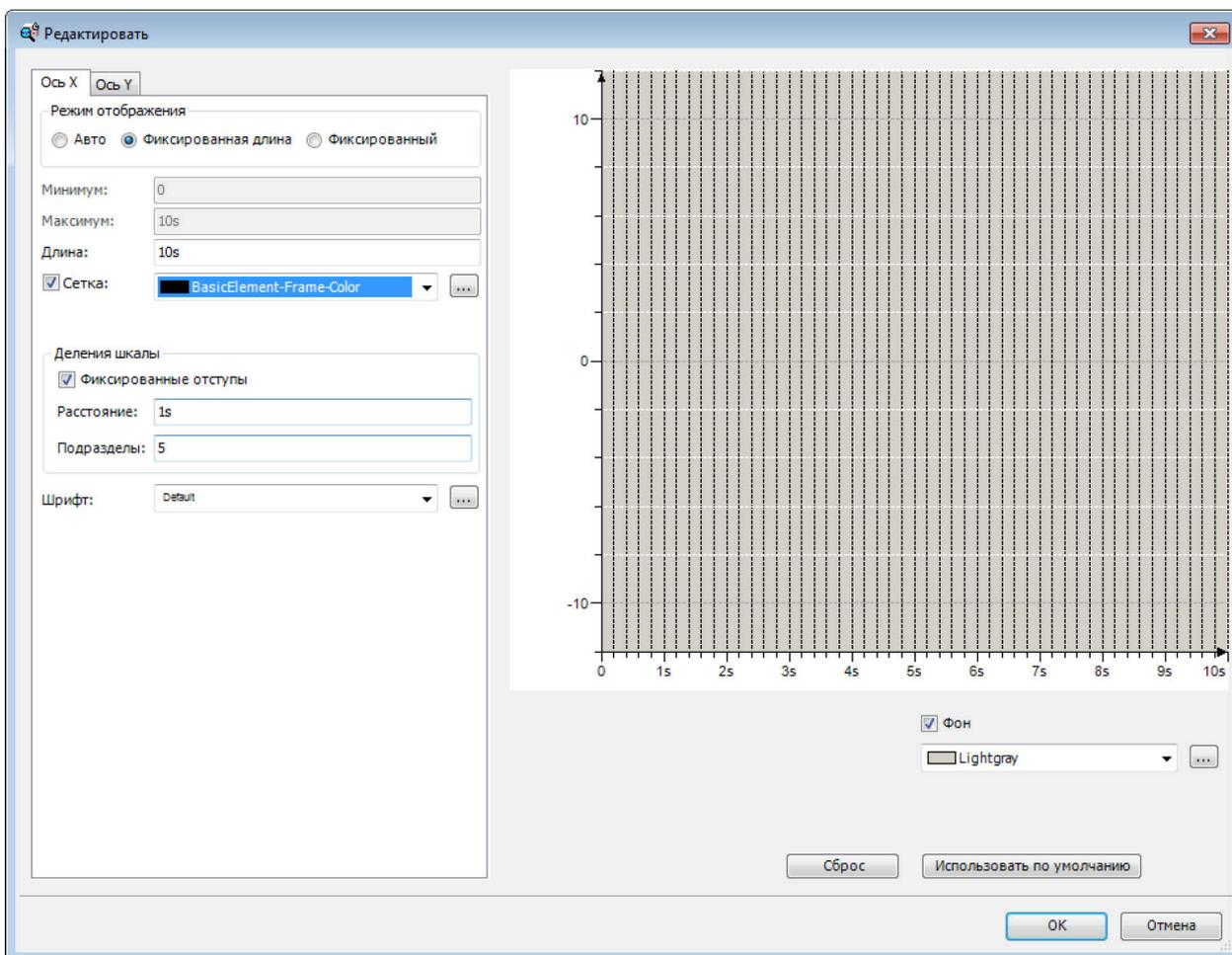


Рис. 8.64. Конфигурация трассировки - **Отображение**

№	Название параметра	Описание параметра
Отображение		
2.		Данное меню имеет две вкладки (Ось X и Ось Y), которые содержат идентичные параметры (пп. 2.1 – 2.10).
2.1.	Режим отображения	Размер шкалы трассировки. Возможные значения: Авто – размер определяется автоматически в зависимости от отображаемых значений Фиксированная длина (<u>только для оси X</u>) – позволяет указать длину шкалы трассировки по оси X. Обратите внимание на пп. 1.10. Фиксированный – позволяет указать начало и конец шкалы трассировки.
2.2.	Минимум	Данный параметр доступен только при выборе режима отображения Фиксированный . Минимум шкалы трассировки.
2.3.	Максимум	Данный параметр доступен только при выборе режима отображения Фиксированный . Максимум шкалы трассировки.

2.4.	Длина	Данный параметр доступен только для оси X при выборе режима отображения Фиксированная длина . Длина шкалы трассировки. Обратите внимание на пп. 1.10.
2.5.	Сетка	При наличии галочки, на трассировке отображается сетка.
2.6.	Описание	Данный параметр доступен только для оси Y . Название трассировки.
2.7.	Фиксированные отступы	При наличии галочки, появляется возможность настроить деления шкалы трассировки.
2.8.	Расстояние	Данный параметр доступен только при наличии галочки в пп. 2.7. Цена делений основной шкалы (в ед. времени).
2.9.	Подразделы	Количество делений подшкалы (в шт.).
2.10.	Шрифт	Шрифт единиц измерения шкалы. В данный момент (CODESYS 3.5 SP6) изменение данного параметра приводит к критической ошибке.
2.11.	Фон	При наличии галочки, появляется возможность выбрать цвет фона трассировки.
2.12.	Цвет фона	Данный параметр доступен только при наличии галочки в пп. 2.11. Цвет фона трассировки.
2.13.	Сброс	Сброс настроек отображения трассировки к значениям по умолчанию.
2.14.	Использовать по умолчанию	При нажатии, текущие настройки отображения трассировки становятся настройками по умолчанию, и нажатие кнопки Сброс присваивает их любой трассировке проекта.

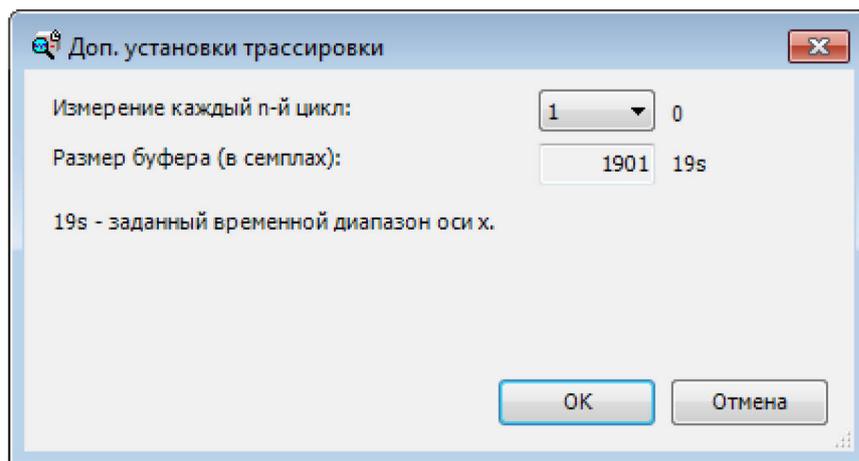


Рис. 8.65. Настройки трассировки - **Дополнительно**

№	Название параметра	Описание параметра
3.	Измерение каждый n-й цикл	Частота отображения и записи значений переменных, выраженная в циклах задачи (см. пп. 1.6). Возможные значения: 1/2/3/.../20/100/200/500/1000 Не рекомендуется использовать значения 100 и выше.
4.	Размер буфера	Текущий размер буфера трассировки. Это значение формируется автоматически и не должно превышать 2000 . Его можно изменять, варьируя время цикла задачи (пп. 1.6), шкалу трассировки (пп. 2.1 – 2.4) и частоту записи переменных трассировки (пп. 3).

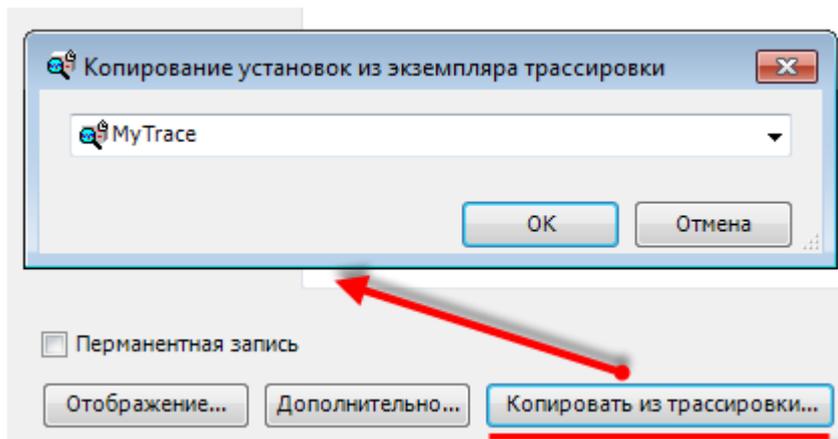


Рис. 8.66. Конфигурация трассировки – **Копировать из трассировки**

С помощью кнопки **Копировать из трассировки** можно перенести настройки компонента **Трассировка** в элемент **Трассировка**.

Для добавления переменных, значения которых будут отображаться и записываться трассировкой, необходимо нажать кнопку **Add variable** и выбрать необходимую переменную:

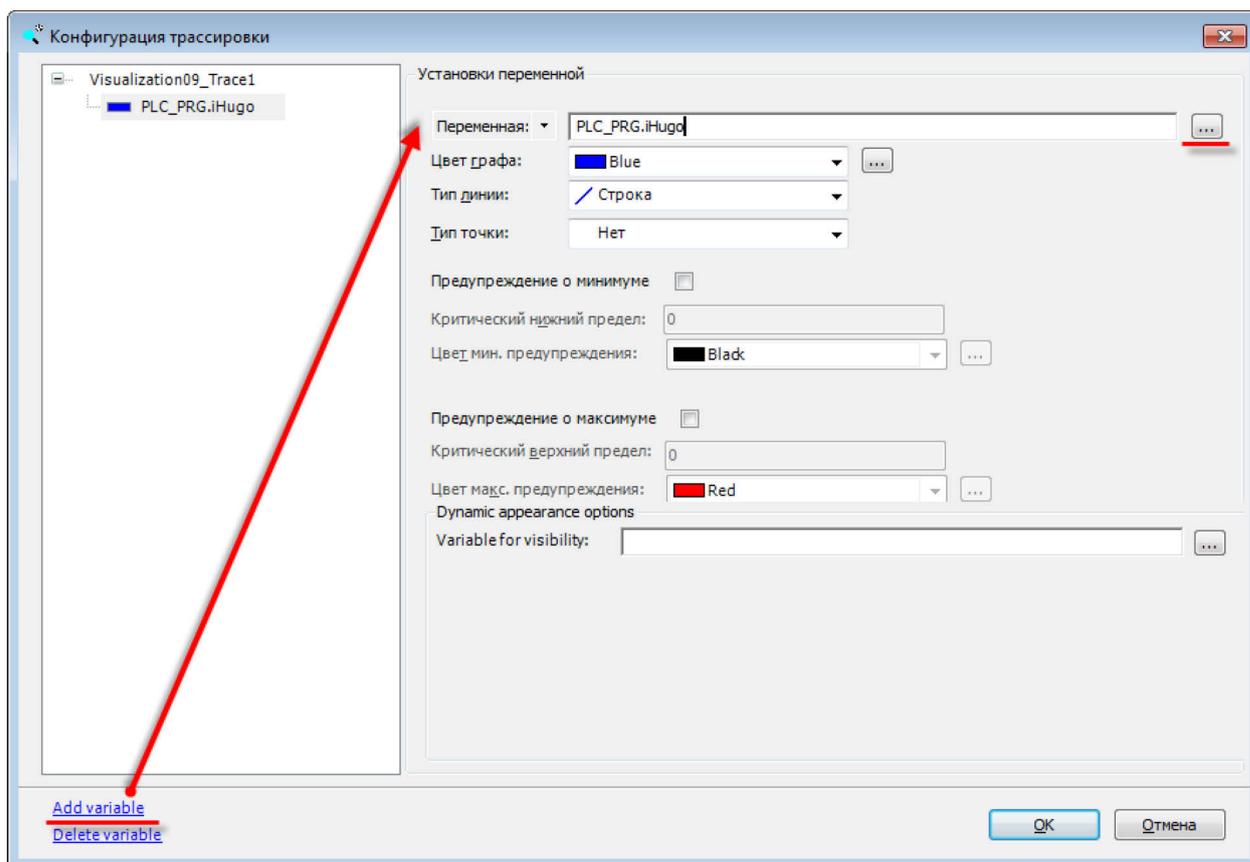


Рис. 8.67. Конфигурация трассировки - **Установки переменной**

№	Название параметра	Описание параметра
5.	Установки переменной	
5.1.	Переменная	Числовая переменная, отображаемая трассировкой.
5.2.	Цвет графа	Цвет графика данной переменной.
5.3.	Тип линии	Тип соединения точек на графике. Возможные значения: Строка – точки соединяются линиями Шаг – точки соединяются прямоугольными импульсами Нет – точки не соединяются
5.4.	Тип точки	Тип точек на графике. Возможные значения: Точка/Крест/Нет
5.5.	Предупреждение о минимуме	При наличии галочки, появляется возможность окрашивать часть графика, выходящую за критический нижний предел (пп. 5.6), в заданный цвет (пп. 5.7).
5.6.	Критический нижний предел	Данный параметр доступен только при наличии галочки в пп. 5.5. Если значение переменной меньше, чем указанное здесь, то цвет графика ниже этого значения окрашивается в заданный цвет.
5.7.	Цвет мин. предупреждения	Данный параметр доступен только при наличии галочки в пп. 5.5. Цвет, в который окрашивается часть графика, выходящая за критический нижний предел.
5.8.	Предупреждение о максимуме	При наличии галочки, появляется возможность окрашивать часть графика, выходящую за критический верхний предел (пп. 5.9), в заданный цвет (пп. 5.10).
5.9.	Критический верхний предел	Данный параметр доступен только при наличии галочки в пп. 5.8. Если значение переменной больше, чем указанное здесь, то цвет графика выше этого значения окрашивается в заданный цвет.
5.10.	Цвет макс. предупреждения	Данный параметр доступен только при наличии галочки в пп. 5.8. Цвет, в который окрашивается часть графика, выходящая за критический верхний предел.
5.11.	Variable for visibility	Переменная типа BOOL . Когда она принимает значение FALSE , данная переменная перестает отображаться на трассировке (но продолжает записываться).

8.6.2. Тренд

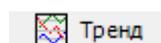


Рис. 8.68. Пиктограмма графического примитива **Тренд**

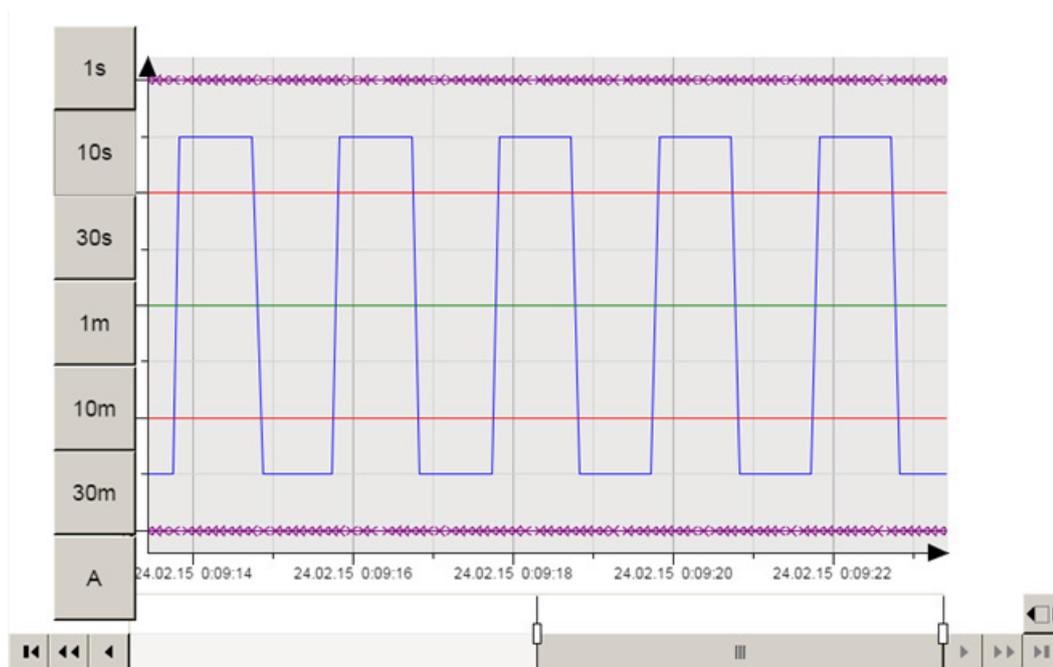


Рис. 8.69. Внешний вид элемента **Тренд** с вспомогательными элементами [Селектор диапазона дат](#) и [Селектор времени](#)

Графический примитив **Тренд** используется для отображения графика с возможностью просмотра истории.

При добавлении графического элемента **Тренд** на экран визуализации, на **Панели устройств** автоматически создается компонент **TrendRecordingManager** с подкомпонентом, имя которого выглядит как

<Имя экрана визуализации>_Trend<Номер тренда>

и задача **TrendRecordingTask**. Компонент **TrendRecordingManager** также может быть добавлен в проект независимо от графического элемента (**Application – Добавить объект – Trend Recording Manager**). Это может использоваться в тех случаях, когда тренд применяется только для сохранения значений переменных в памяти целевого устройства без необходимости их визуализации. Архивы тренда сохраняются в памяти целевого устройства в формате [.sqlite](#). Файлы данного формата можно открыть с помощью соответствующего ПО ([SqliteBrowser](#), [SqliteStudio](#) и др.).

Примитив **Тренд** в целом аналогичен примитиву [Трассировка](#) и отличается только дополнительными программными (**Trend Recording Manager**) и графическими ([Селектор диапазона дат](#), [Селектор времени](#), [Легенда](#)) компонентами. **Обратите внимание**, что без [Селектора диапазона дат](#) на тренде не будут отображаться значения переменных.

Описание типичных параметров графического примитива приведено в [табл. 2](#).

Уникальные параметры примитива описаны в табл. 25.

Настройки компонента **Trend Recording Manager** приведены в табл. 26.

Пример работы с элементом приведен в [п. 11.2.10](#).

Табл. 25. Уникальные параметры графического примитива **Тренд**

№	Название параметра	Описание параметра
1.	Запись тренда	Вызов меню, в котором можно выбрать компонент Trend Recording Manager , который будет использоваться для сохранения значений переменных тренда.
2.	Внешний вид диаграммы	Настройки отображения тренда. Аналогичны настройкам трассировки, см. табл. 24 , пп. 2.
3.	Показать курсор	При наличии галочки, при просмотре истории тренда отображается курсор (метка со значением).
4.	Показать подсказку	Данный параметр доступен только при наличии галочки в пп. 2. При наличии галочки, курсор тренда отображает информация о значениях переменных.
5.	Показать фрейм	При наличии галочки, вокруг элемента отображается контур.
6.	Обозначения шкалы	
6.1.	Временные отметки	Тип временных отметок на тренде. Возможные значения: Абсолютные временные отметки – на оси X тренда отображается текущее системное время Относительные временные отметки – на оси X тренда отображается время, начиная с 0 часов 0 минут 0 секунд 1 января 1970 года (точка отсчета для Unix-времени).
6.2.	Обозначения в две строки	При наличии галочки, временная отметка отображается в две строки: на первой – дата, на второй – время.
6.3.	Опустить незначимую информацию	При наличии галочки, крайняя левая временная отметка тренда всегда отображается в формате День-Месяц-Год Часы-Минуты-Секунды , а все остальные – в формате Минуты-Секунды . Пп. 5.4. в данном случае неактивен.
6.4.	Интернационализация	
6.4.1.	Дата	Данный параметр доступен только при отсутствии галочки в пп. 5.4. Формат вывода даты (см. п. 9.3).
6.4.2.	Время	Данный параметр доступен только при отсутствии галочки в пп. 5.4. Формат вывода времени (см. п. 9.3).

Прикрепленные экземпляры элементов управления		
7.		В этом пункте указываются названия вспомогательных элементов, привязываемых к данному тренду. Можно сформировать их автоматически из контекстного меню (открывается по нажатию ПКМ на тренд) с помощью команды Вставить элементы для управления трендом .
7.1.	Селектор диапазона дат	Селектор диапазона дат данного тренда. Используется для просмотра истории тренда. Без этого элемента на тренде не будет отображаться изменение значений переменных. Настройки элемента описаны в п. 8.7.1 .
7.2.	Селектор времени	Селектор времени данного тренда. Используется для масштабирования графиков тренда. Настройки элемента описаны в п. 8.7.2 .
7.3.	Легенда	Легенда тренда. Используется для описания соответствия переменных и цветов/типов графиков, которыми отображаются их значения. Также отображает значения переменных в точке курсора. Настройки элемента описаны в п. 8.6.3 .

Для открытия конфигурации, необходимо либо открыть соответствующий подкомпонент компонента **Trend Recording Manager** на **Панели устройств**, либо использовать команду **Конфигурация трассировки** в контекстном меню элемента (открывается по нажатию ПКМ).

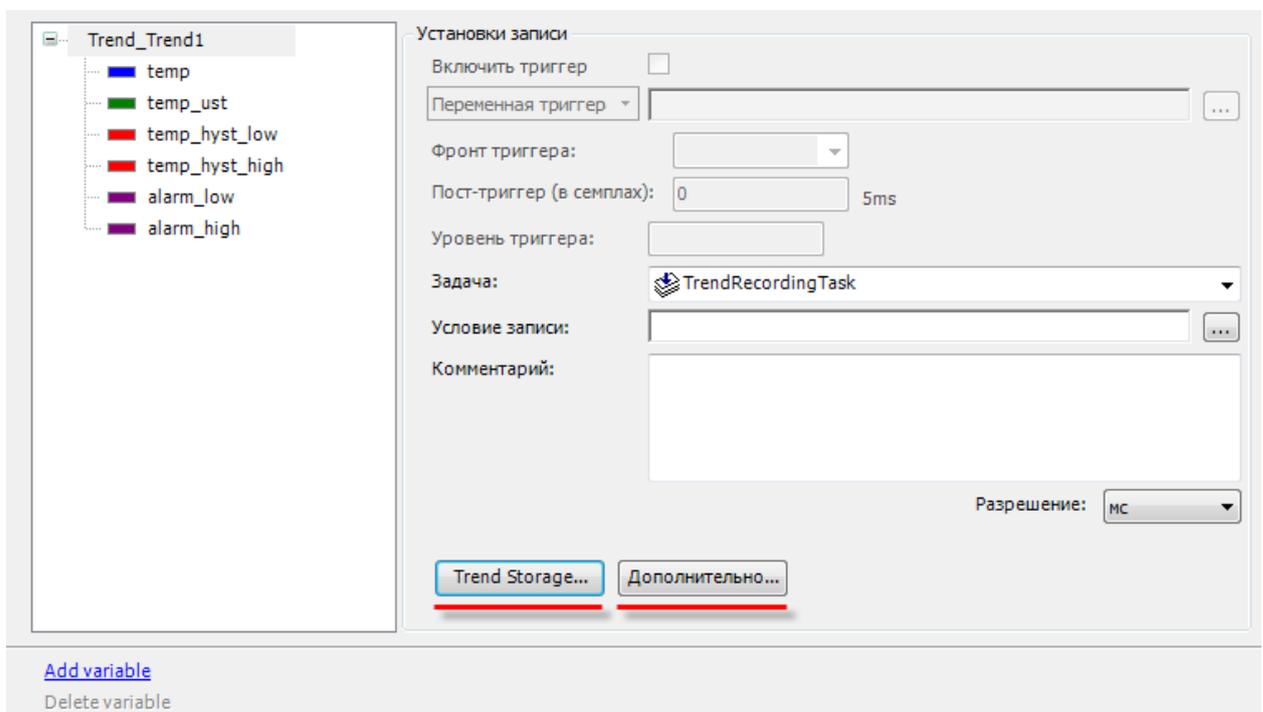


Рис. 8.70. Конфигурация тренда

Установки записи и установки переменных в конфигурации тренда идентичны соответствующим установкам трассировки, поэтому см. их описание в [табл. 24](#), пп.1 и пп. 5.

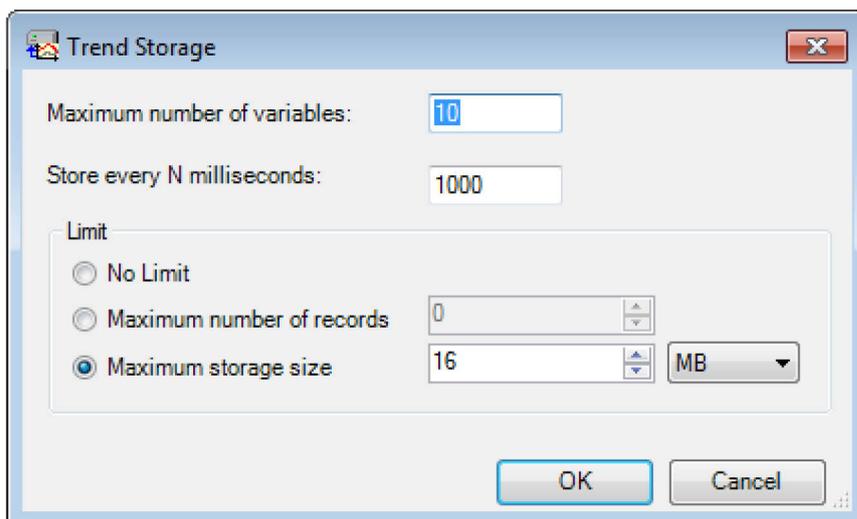


Рис. 8.71. Конфигурация тренда – Trend Storage

Табл. 26. Конфигурация элемента Тренд

№	Название параметра	Описание параметра
1.	Trend Storage	
1.1.	Maximum Number Of Variables	Максимальное число переменных, отображаемых и архивируемых трендом. По умолчанию оно равно 10, максимально возможное значение – 199. Но для корректной работы проекта крайне рекомендуется не увеличивать значение этого параметра.
1.2.	Store every N milliseconds	Частота отображения и записи переменных тренда. По умолчанию она равна 1000 мс. Для корректной работы проекта крайне рекомендуется не уменьшать значение этого параметра.
1.3.	Limit	Тип ограничения на размер файла тренда. Возможные значения: No Limit – ограничение отсутствует (строго говоря, в данном случае ограничением является объем памяти целевого устройства); Maximum number of records – ограничение на количество записей в файл; Maximum storage size – ограничение на размер файла.

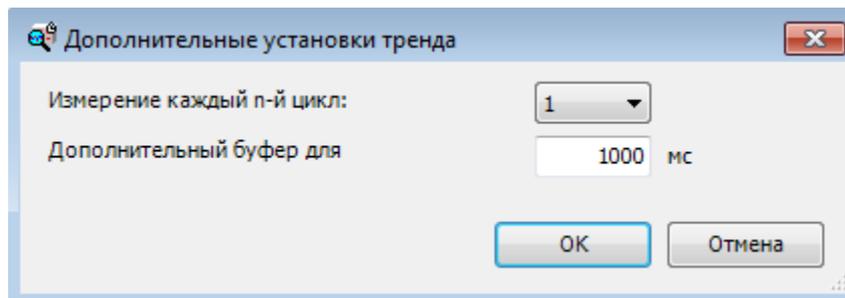


Рис. 8.72. Конфигурация тренда – **Дополнительно**

№	Название параметра	Описание параметра
2.	Дополнительно	
2.1.	Измерение каждый n-й цикл	Частота отображения и записи значений переменных, выраженная в циклах задачи тренда. Возможные значения: 1/2/3/.../20/100/200/500/1000
2.2.	Дополнительный буфер для	Буфер для хранения значений, не записанных трендом из-за непредвиденных задержек. Используется для предотвращения потери данных.

8.6.3. Легенда

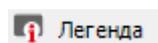


Рис. 8.73. Пиктограмма графического примитива **Легенда**

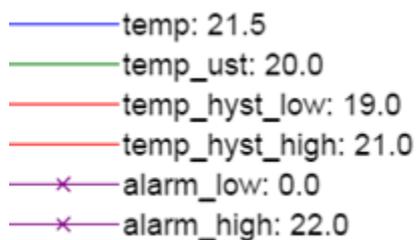


Рис. 8.74 Внешний вид элемента **Легенда**

Графический примитив **Легенда** используется совместно с элементом [Тренд](#) для описания соответствия переменных и цветов/типов графиков, которыми отображаются их значения. Также отображает значения переменных в точке курсора.

Легенда должна быть привязана к одному из трендов (см. табл. 25. пп. 7.3).

Описание типичных параметров графического примитива приведено в [табл. 2](#).

Уникальные параметры примитива описаны в табл. 27.

Табл. 27. Уникальные параметры графического примитива Легенда

№	Название параметра	Описание параметра
1.	Ориентация	Ориентация элемента. Однозначно определяется соотношением высоты и ширины элемента: ширина > высота – горизонтальная ориентация высота > ширина – вертикальная ориентация
2.	Показать фрейм	При наличии галочки, у элемента отображается контур и фон.
3.	Размещение	
3.1.	Макс. число строк	Число строк элемента.
3.2.	Макс. число столбцов	Число столбцов элемента.

8.6.4. Элемент ActiveX

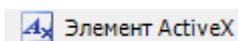


Рис. 8.75. Пиктограмма графического примитива **ActiveX**

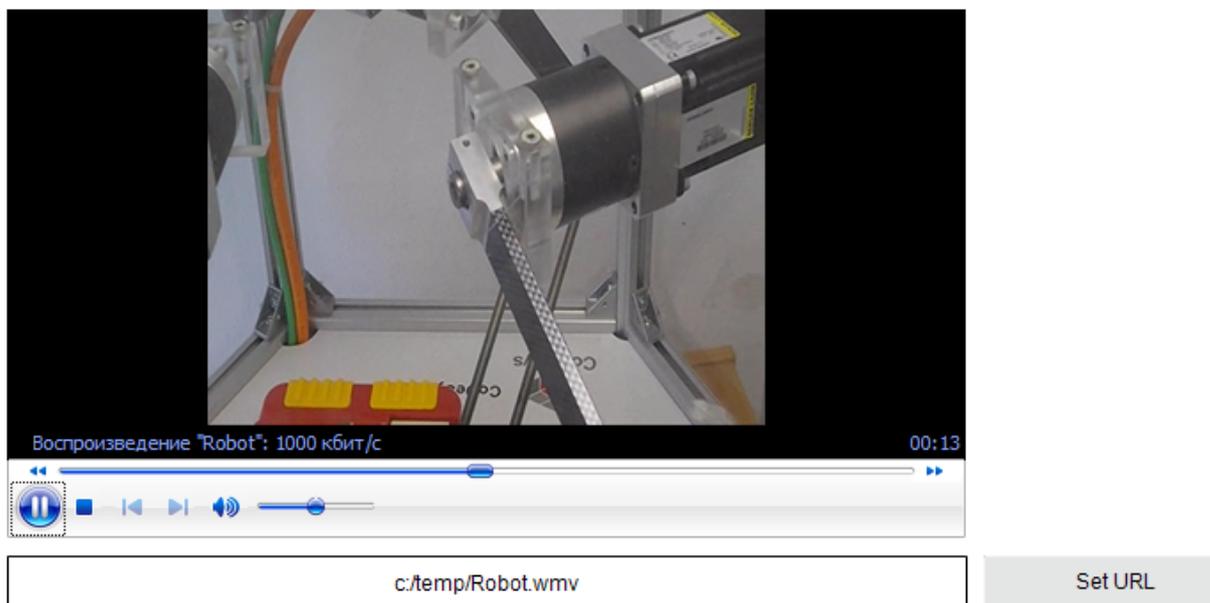


Рис. 8.76. Пример использования графического примитива **ActiveX** – проигрывание видеофайлов с помощью **Windows Media Player** на виртуальном контроллере **CODESYS Control Win V3**

Графический примитив **ActiveX** используется для интеграции приложений с поддержкой технологии **ActiveX** в проект **CODESYS**. *Обратите внимание*, что использование примитива возможно только на целевых устройствах с ОС **Windows**.

Настройка элемента заключается в выборе методов приложения и используемых ими переменных.

Поскольку этот функционал в данный момент не поддерживается устройствами компании **ОВЕН**, а также является достаточно сложным в использовании, то его описание не приводится. При необходимости можно воспользоваться справкой **CODESYS** и примером использования элемента от компании [3S](#) (разработчик **CODESYS**): [ActiveXExample.package](#)

Пример устанавливается с помощью **Менеджера пакетов** (вкладка **Инструменты**).

8.6.5. WebBrowser

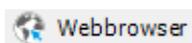


Рис. 8.77. Пиктограмма графического примитива **WebBrowser**

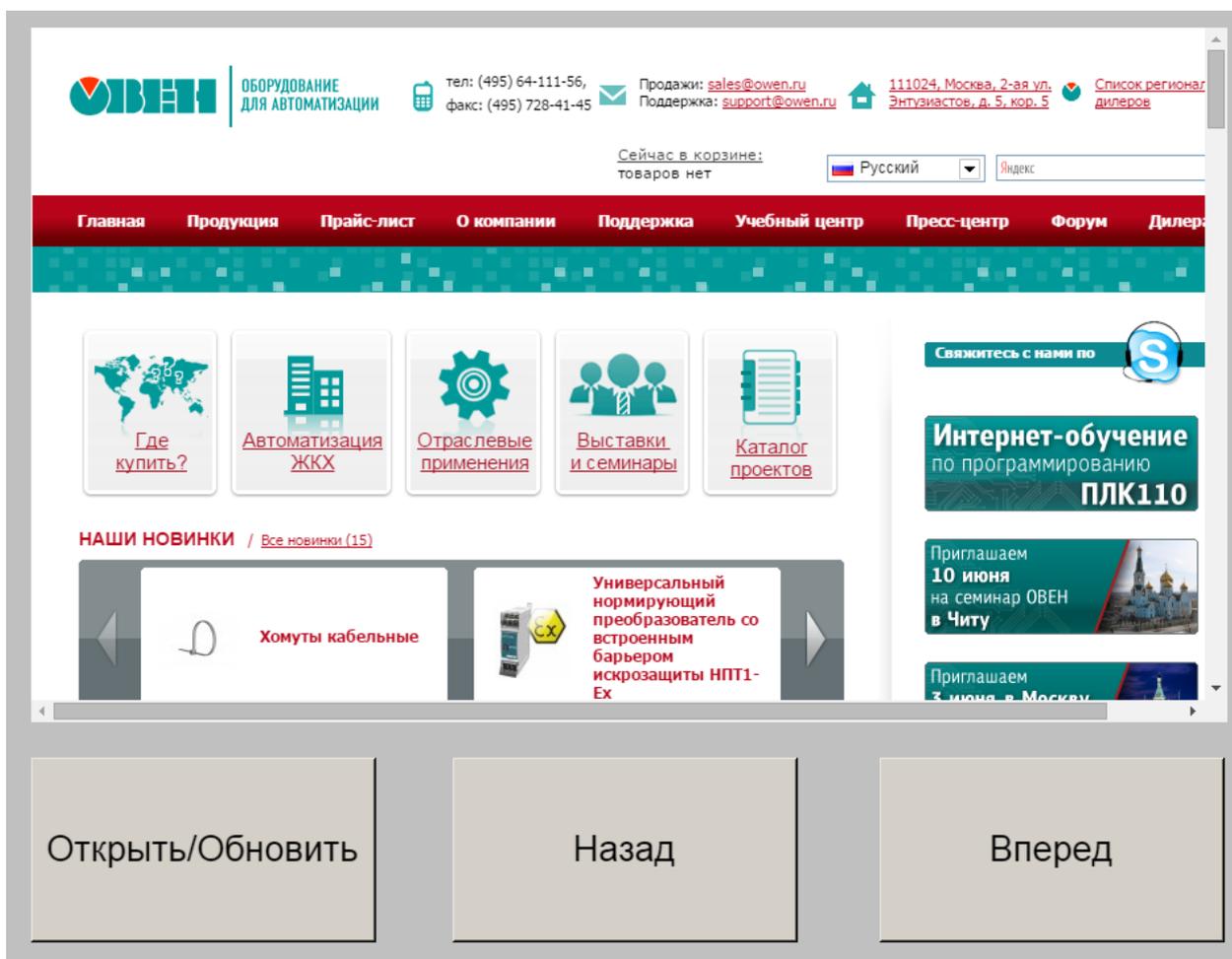


Рис. 8.78. Внешний вид элемента **WebBrowser**

Графический примитив **WebBrowser** используется для отображения веб-страниц. **Обратите внимание**, что использование примитива возможно только на целевых устройствах с поддержкой **Java** и **HTML5**. При этом невозможно гарантировать открытие в элементе любой веб-страницы.

Описание типичных параметров графического примитива приведено в [табл. 2](#).

Уникальные параметры примитива описаны в табл. 28.

Пример работы с элементов приведен в [п. 11.2.11](#).

Табл. 28. Уникальные параметры графического примитива **WebBrowser**

№	Название параметра	Описание параметра
1.	Control variables	
1.1.	URL	Переменная типа STRING , содержащая адрес веб-страницы, отображаемой элементом (в виде http://owen.ru).
1.2.	Show	Переменная типа BOOL , используемая для открытия веб-страниц в элементе. Если она принимает значение TRUE , то веб-страница, указанная в пп. 1, открывается элементом. Обратите внимание , что значение FALSE не приводит к прекращению отображения страницы.
1.3.	Back	Переменная типа BOOL , используемая для возвращения к просматриваемой ранее странице (по значению TRUE).
1.4.	Forward	Переменная типа BOOL , используемая для возвращения к странице, с которой был сделан переход по кнопке Back (по значению TRUE).

8.6.6. Символ ожидания Кубик/Цветок

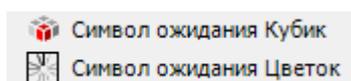


Рис. 8.79. Пиктограмма графического примитива **Символ ожидания**



Рис. 8.80. Внешний вид элемента **Символ ожидания**

Графический примитив **Символ ожидания** используется для отображения соответствующих анимированных элементов.

Описание типичных параметров графического примитива приведено в [табл. 2](#).

8.6.7. Текстовый редактор

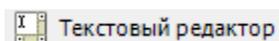


Рис. 8.81. Пиктограмма графического примитива **Текстовый редактор**

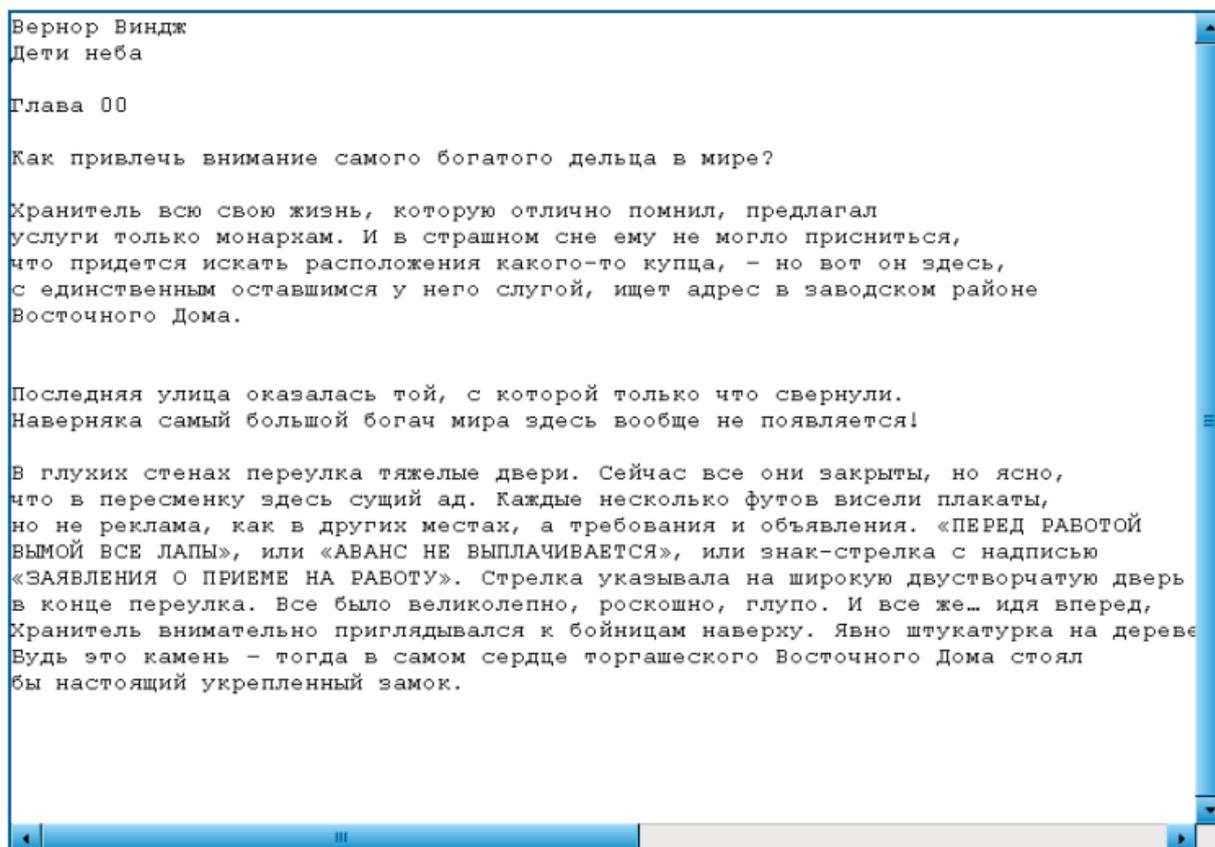


Рис. 8.82. Внешний вид элемента **Текстовый редактор**

Графический примитив **Текстовый редактор** используется для просмотра и изменения текстовых файлов (формата **.txt**).

Описание типичных параметров графического примитива приведено в [табл. 2](#).

Уникальные параметры примитива описаны в табл. 29.

Пример работы с элементом приведен в [п. 11.2.12](#).

Табл. 29. Уникальные параметры графического примитива **Текстовый редактор**

№	Название параметра	Описание параметра
1.	Управляющие переменные	
1.1.	Файл	
1.1.1.	Имя файла	Переменная типа STRING , содержащая путь к файлу, над которым производятся операции, описанные ниже.
1.1.2.	Открыть	Переменная типа BOOL , управляющая открытием файла (по значению TRUE).
1.1.3.	Закрыть	Переменная типа BOOL , управляющая закрытием файла (по значению TRUE).
1.1.4.	Сохранить	Переменная типа BOOL , управляющая сохранением файла (по значению TRUE).
1.1.5.	Новый	Переменная типа BOOL , управляющая созданием нового файла с именем, указанным в пп. 1.1.1 (по значению TRUE).
1.2.	Редактировать	
1.2.1.	Искать	Переменная типа STRING , содержащая текстовый фрагмент, поиск которого будет производиться в открытом файле.
1.2.2.	Найти	Переменная типа BOOL , управляющая началом поиска в файле (по значению TRUE).
1.2.3.	Найти далее	Переменная типа BOOL , управляющая шагами поиска в файле (по значению TRUE). Если переменная имеет значение TRUE , то каждый раз, когда переменная Найти принимает значение TRUE , осуществляется перемещение курсора к следующему искомому фрагменту в файле.
1.3.	Позиция каретки	
1.3.1.	Линия	Целочисленная переменная, содержащая номер строки, на которой находится курсор (нумерация ведется с 0).
1.3.2.	Столбец	Целочисленная переменная, содержащая номер столбца, на котором находится курсор (нумерация ведется с 0).
1.3.3.	Позиция	Целочисленная переменная, содержащая номер символа в файле, за которым находится курсор (нумерация ведется с 0).
1.3.4.	Задать триггер	Переменная типа BOOL , управляющая перемещением курсора. Если эта переменная принимает значение TRUE , то курсор перемещается в положение, определяемое значениями переменных п. 1.3.1 – 1.3.2.
1.4.	Выбор	
1.4.1.	Стартовая позиция	Целочисленная переменная, содержащая номер первого символа из выделенного фрагмента (нумерация ведется с 0).
1.4.2.	Конечная позиция	Целочисленная переменная, содержащая номер последнего символа из выделенного фрагмента (нумерация ведется с 0).
1.4.3.	Стартовый номер строки	Целочисленная переменная, содержащая номер первой строки выделенного фрагмента (нумерация ведется с 0).
1.4.4.	Начальный индекс столбца	Целочисленная переменная, содержащая номер первого столбца выделенного фрагмента (нумерация ведется с 0).

1.4.5.	Конечный индекс строки	Целочисленная переменная, содержащая номер последней строки выделенного фрагмента (нумерация ведется с 0).
1.4.6.	Конечный индекс столбца	Целочисленная переменная, содержащая номер последнего столбца выделенного фрагмента (нумерация ведется с 0).
1.4.7.	Выбираемая линия	Целочисленная переменная, содержащая номер выделенной строки (нумерация ведется с 0). Используется совместно с пп. 1.4.8.
1.4.8.	Выбор триггера	Переменная типа BOOL , управляющая выделением строк. Если это переменная принимает значение TRUE , то выделяется строка, указанная в пп. 1.4.7.
1.5.	Обработка ошибок	
1.5.1.	Переменная для кода ошибки	Целочисленная переменная, содержащая код последней ошибки, возникшей при использовании элемента. Коды ошибок приведены в списке GVL_ErrorCodes библиотеки VisuElemTextEditor .
1.6.	Переменная для содержимого изменилась	Переменная типа BOOL , которая принимает значение TRUE при изменении содержимого файла через элемент.
1.7.	Переменная для режима доступа	Переменная типа BOOL , определяющая тип доступа к файлу. FALSE – чтение/запись TRUE – только чтение
2.	Макс. длина линии	Максимальная длина строки элемента.
3.	Режим редактора	Режим открытия файлов. Возможные значения: Только чтение/Чтение-запись См. также пп. 1.7.
4.	Новые файлы	
4.1.	Кодировка	Кодировка созданного файла (см. пп. 1.1.5). Возможные значения: ASCII/ Unicode (Little endian)/Unicode (Big endian)
4.2.	Новая последовательность символов строки	Символ перехода на новую строку в созданном файле (см. пп. 1.1.5). Возможные значения: CR/LF – обычно используется в ОС Windows LF – обычно используется в ОС Unix

8.6.8. Путь 3D

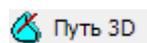


Рис. 8.83. Пиктограмма графического примитива **Путь 3D**

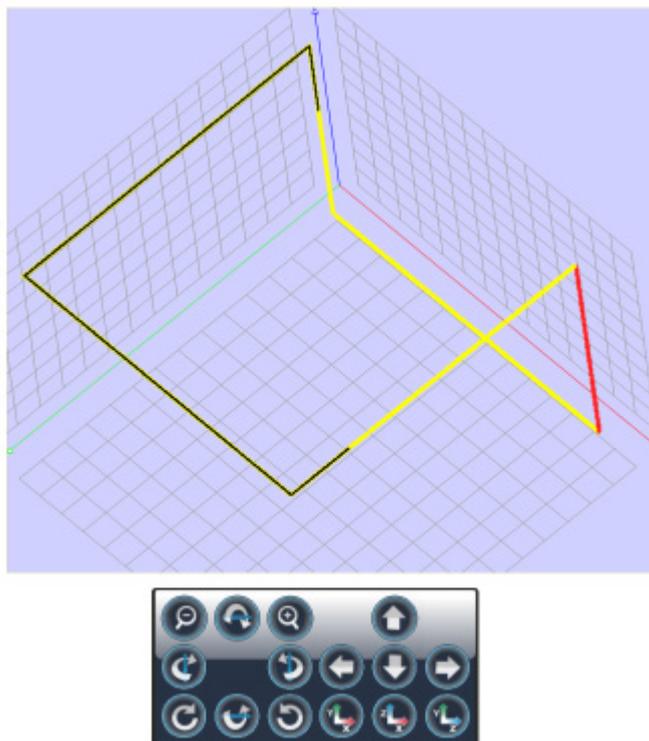


Рис. 8.84. Внешний вид элемента **Путь 3D**

Графический примитив **Путь 3D** используется для прорисовки в визуализации двух- и трехмерных фигур. Основное применение – отрисовка движений станков ЧПУ и роботов.

Поскольку этот функционал в данный момент не поддерживается устройствами компании **ОВЕН**, а также является достаточно сложным в использовании, то его описание не приводится. При необходимости можно воспользоваться справкой **CODESYS** и примером использования элемента от компании [3S](#) (разработчик **CODESYS**): [3DPathGenerator.package](#)

Пример устанавливается с помощью **Менеджера пакетов** (вкладка **Инструменты**).

8.7. Элементы управления датой/временем

8.7.1. Селектор диапазона дат

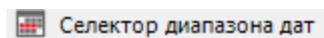


Рис. 8.85. Пиктограмма графического примитива **Селектор диапазона дат**

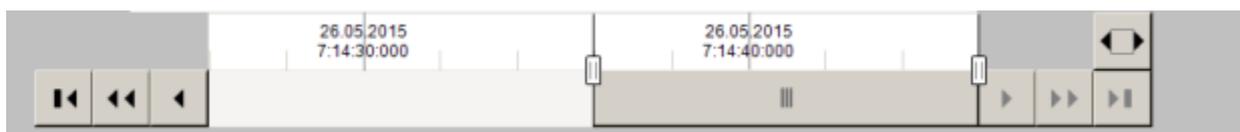


Рис. 8.86. Внешний вид элемента **Селектор диапазона дат**

Графический примитив **Селектор диапазона дат** используется для просмотра истории элемента [Тренд](#). Селектор должен быть привязан к одному из трендов (см. табл. 25, пп. 7.3).

Описание типичных параметров графического примитива приведено в [табл. 2](#).

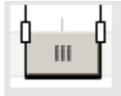
Уникальные параметры примитива описаны в табл. 30.

Функции кнопок элемента описаны в табл. 31.

Табл. 30. Уникальные параметры графического примитива **Селектор диапазона дат**

№	Название параметра	Описание параметра
1.	Прикрепленный экземпляр элемента	Элемент Тренд , к которому прикреплен данный селектор. См. также табл. 25, пп. 7.3.
2.	Обозначения шкалы	
2.1.	Обозначения в две строки	При наличии галочки, временная отметка отображается в две строки: на первой – дата, на второй – время.
2.2.	Опустить незначимую информацию	При наличии галочки, крайняя левая временная отметка тренда всегда отображается в формате День-Месяц-Год Часы-Минуты-Секунды , а все остальные – в формате Минуты-Секунды . Пп. 2.3 в данном случае неактивен.
2.3.	Интернационализация	
2.3.1.	Дата	Данный параметр доступен только при отсутствии галочки в пп. 2.2. Определяет формат вывода даты (см. п. 9.4).
2.3.2.	Время	Данный параметр доступен только при отсутствии галочки в пп. 2.2. Определяет формат вывода времени (см. п. 9.4).
3.	Дополнительные кнопки	
3.1.	Переход к максимальной временной отметке	При наличии галочки, у элемента присутствует кнопка Переход к максимальной временной отметке (см. табл. 31, пп. 5).
3.2.	Переход к минимальной временной отметке	При наличии галочки, у элемента присутствует кнопка Переход к минимальной временной отметке (см. табл. 31, пп. 4).
3.3.	Уменьшить масштаб	При наличии галочки, у элемента присутствует кнопка Уменьшить масштаб (см. табл. 31, пп. 6).

Табл. 31. Функции кнопок элемента **Селектор диапазона дат**

№	Внешний вид кнопки	Функция кнопки
1.		Ползунок селектора. Перемещая его, пользователь изменяет просматриваемую область истории. Ширину ползунка можно менять – это влияет на масштаб тренда и шаг ползунка
2.		Смещение просматриваемой области истории на одно значение назад/вперед.
3.		Смещение просматриваемой области истории на одну временную отметку (определяется размером ползунка) назад/вперед.
4.		Переход к первому значению историю.
5.		Переход к последнему значению истории. Нажатие на эту кнопку переключает тренд из режима просмотра истории в режим обновления в реальном времени.
6.		Переключение тренда в режим минимального масштаба.

8.7.2. Селектор времени



Рис. 8.87. Пиктограмма графического примитива **Селектор времени**



Рис. 8.88. Внешний вид элемента **Селектор времени**

Графический примитив **Селектор времени** используется для масштабирования истории элемента **Тренд** и определяет временную область, отображаемую элементом. Селектор должен быть привязан к одному из трендов (см. табл. 25, пп. 7.3).

Описание типичных параметров графического примитива приведено в [табл. 2](#).

Уникальные параметры примитива описаны в табл. 32.

Табл. 32. Уникальные параметры графического примитива **Селектор времени**

№	Название параметра	Описание параметра
1.	Прикрепленный экземпляр элемента	Элемент Тренд , к которому прикреплен данный селектор. См. также табл. 25, пп. 7.3.
2.	Тексты	
2.1.	Текст	Текст элемента.
3.	Времена	
	В данной вкладке создаются кнопки управления масштабом тренда. Каждой кнопке соответствует временной диапазон, который отображается на тренде при ее нажатии.	
3.1.	Выбор «Все»	При наличии галочки, у элемента присутствует кнопка « All », временная область которой соответствует всей истории тренда.
4.	Управляющие переменные	
4.1.	Время	Переменная типа LTIME , принимающая значение отображаемой на тренде временной области. В случае выбора всей истории (см. пп. 3.1), в переменную не записывается значение, при этом переменная, привязанная к параметру Все выбранные (пп. 4.2) принимает значение TRUE .
4.2.	Все выбранные	Переменная типа BOOL , принимающая значение TRUE при выборе в качестве временной области всей истории тренда (см. пп. 3.1).

8.8. Менеджер тревог

8.8.1. Компонент Конфигуратор тревог

Работа графических примитивов **Таблица тревог** и **Баннер тревог**, входящих в состав группы **Менеджер тревог** возможна только при наличии в проекте компонента **Конфигуратор тревог**.

Конфигуратор тревог используется для создания списков аварийных сигналов, информация об изменении состояния которых будет сохраняться в памяти целевого устройства.

Обратите внимание, что в текущей версии **CODESYS** (3.5 SP6) имеются проблемы с добавлением **Конфигуратора тревог** при использовании **русскоязычного интерфейса** среды программирования. В этом случае рекомендуется переключить язык интерфейса на английский (**Инструменты – Опции – Международные установки**; потребуется перезапуск **CODESYS**), добавить компонент **AlarmConfiguration**, после чего переключить язык интерфейса обратно на русский (*исправлено в SP7*).

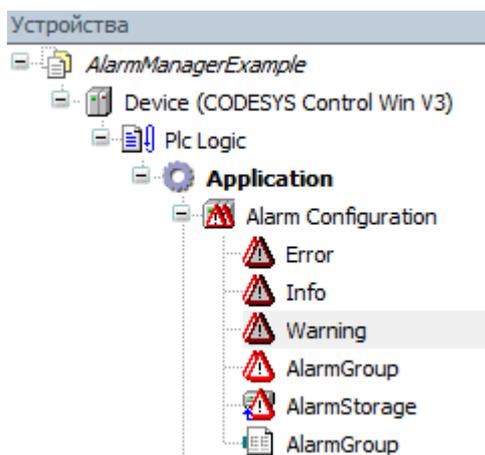


Рис. 8.89. Компонент **Конфигуратор тревог** и его дочерние компоненты

Компонент включает в себя четыре дочерних компонента:

1. **Классы тревог** предназначены для разделения тревог на отдельные типы. Иными словами, они определяют набор базовых параметров (таких, как приоритет, способ подтверждения и т.д.), который может соответствовать как одной, так и нескольким тревогам. Пользователь может создавать собственные классы тревог, помимо имеющихся по умолчанию;
2. **Группы тревог** представляют собой наборы тревог, которые могут относиться как к одному, так и к разным классам. На уровне группы настраиваются индивидуальные параметры тревоги – условия появления и пропадания, текст сообщения и т.д.;

3. **Списки текстов** содержат тексты сообщений тревог и их идентификаторы. Они идентичны компоненту [Список текстов](#);
4. **Хранилище тревог** обеспечивает архивацию тревог в память целевого устройства (в формате **.sqlite**). В отличие от остальных подкомпонентов, в проекте может существовать только одно хранилище тревог.

По умолчанию, добавленный в проект конфигуратор содержит **три класса тревог** (Error, Info, Warning) и хранилище **AlarmStorage**.

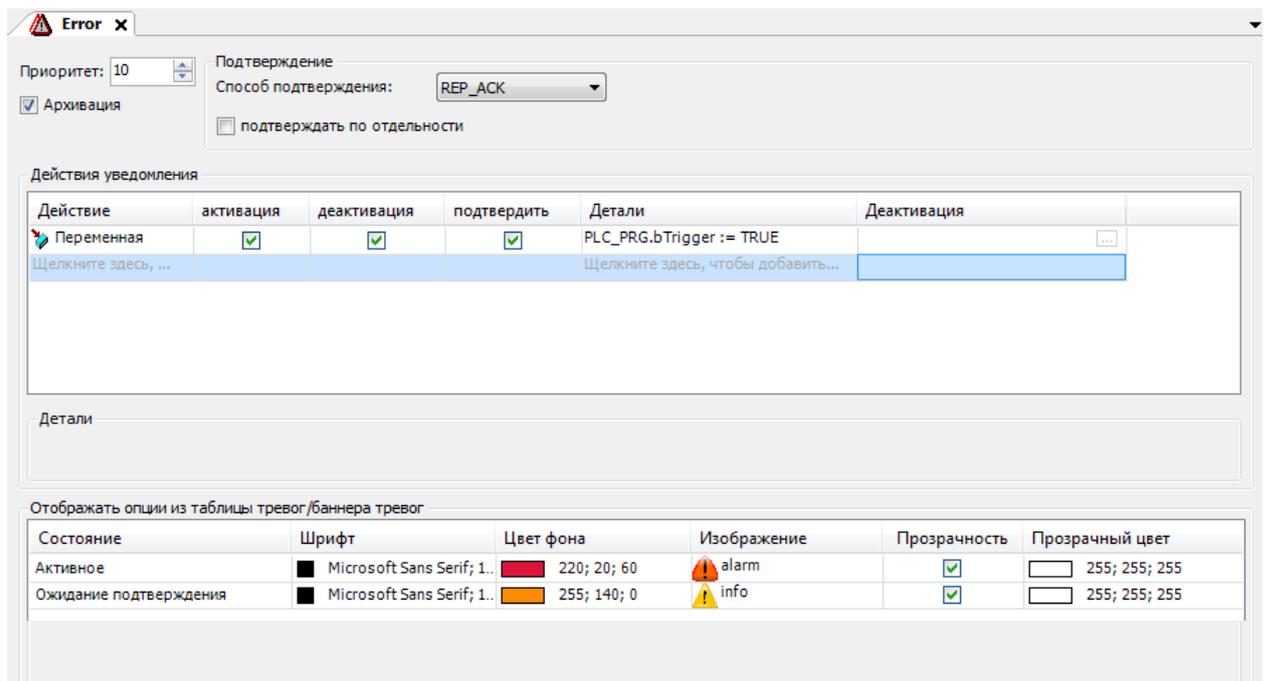


Рис. 8.90. Настройки класса тревог **Error**

В зависимости от **способа подтверждения**, тревога может находиться в следующих состояниях:

1. **Нормальное** – условие возникновения тревоги не выполняется;
2. **Активна** – условие возникновения тревоги выполнилось и продолжает выполняться в данный момент;
3. **Ожидание подтверждения** – условие выполнения тревоги перестало выполняться, но тревога не подтверждена пользователем;
4. **Активна, подтверждена** – условие возникновения тревоги выполняется в данный момент, при этом тревога сквитирована пользователем.

Существуют следующие варианты смены состояний:

1. **Активация** – условия возникновения тревоги начало выполняться;
2. **Деактивация** – условия возникновения тревоги перестало выполняться (*обратите внимание*, что это не обязательно соответствует исчезновению сигнала тревоги);
3. **Подтверждение** – пользователь сквитировал тревогу.

Табл. 33. Настройки классов тревог

№	Название параметра	Описание параметра
1.	Приоритет	Приоритет тревог, привязанных к данному классу. Высший приоритет – 0 , низший – 255 .
2.	Архивация	При наличии галочки, история тревог данного класса будет сохраняться в файл. См. также табл. 34, пп. 2.
3.	Подтверждение	
3.1.	Способ подтверждения	<p>Условие возвращения тревоги данного класса в нормальное состояние. Возможные значения:</p> <p>ACK – квитирование тревоги;</p>  <p>REP – деактивация тревоги;</p>  <p>ACK_REP – подтверждение деактивированной или деактивация сквитированной тревоги;</p>  <p>REP_ACK - подтверждение деактивированной тревоги;</p>

		 <p>ACK_REP_ACK - подтверждение деактивированной тревоги, причем перед деактивацией тревога опционально могла быть сквитирована.</p> 
3.2.	Подтверждать по отдельности	При наличии галочки, тревоги данного класса требуется подтверждать по отдельности (при отсутствии – можно все сразу).
4.	Действие уведомления	
4.1.	Действие	<p>Действие, выполняемое при активации/деактивации/подтверждения (см. пп. 4.2 – 4.4) тревог данного класса. Возможные значения:</p> <p>Переменная – изменение значения переменной</p> <p>Выполнение – запуск файла</p> <p>Вызов – вызов функционального блока. Блок должен поддерживать интерфейс IAIAlarmNotifiable библиотеки AlarmManager.library.</p>
4.2.	Активация	При наличии галочки, действие выполняется при активации тревог данного класса.
4.3.	Деактивация	При наличии галочки, действие выполняется при деактивации тревог данного класса.
4.4.	Подтвердить	При наличии галочки, действие выполняется в случае подтверждения тревог данного класса.
4.5.	Детали	Детали действия (выбор переменной/файла/функционального блока).
4.6.	Деактивация	Переменная типа BOOL , которая отключает выполнение действий данного класса тревог (по значению TRUE).
5.	Отображать опции из таблицы тревог/ баннера тревог	
	Обратите внимание , что настройки данной вкладки могут работать некорректно на виртуальном контроллере при использовании версии CODESYS выше 3.5 SP6 и всех его патчей.	
5.1.	Состояние	Состояние тревоги (набор состояний определяется способом подтверждения, см. пп. 3.1), для которого применяются описанные ниже настройки.
5.2.	Шрифт	Шрифт текста тревог данного класса тревог в данном состоянии.
5.3.	Цвет фона	Цвет фона строк тревог данного класса в данном состоянии.
5.4.	Изображение	Пиктограмма тревог данного класса в данном состоянии.
5.5.	Прозрачность	При наличии галочки, цвет, указанный в параметре Прозрачный цвет , не отображается изображением (пп. 5.4). Это можно использовать в тех случаях, когда пиктограмма тревоги имеет непрозрачный фон.
5.6.	Прозрачный цвет	Указанный здесь цвет при наличии галочки Прозрачный не будет отображаться.

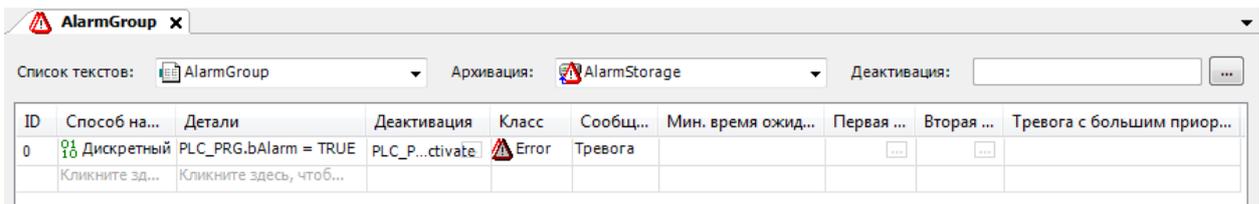


Рис. 8.91. Настройки группы тревог AlarmGroup

Табл. 34. Настройки групп тревог

№	Название параметра	Описание параметра
1.	Список текстов	Список текстов, связанный с данной группой тревог.
2.	Архивация	Хранилище тревог, связанное с данной группой тревог. Без выбора хранилища архивация тревог производиться не будет.
3.	Деактивация	Переменная типа BOOL , которая определяет возможность срабатывания тревог данной группы: TRUE – тревоги не срабатывают FALSE – тревоги срабатывают
4.	Настройки тревог	
4.1.	ID	Идентификатор, который автоматически присваивается тревоге при ее создании. Этот ID соответствует номеру, используемому в списке текстов тревог. Его можно изменить, однако он должен оставаться уникальным в группе тревог. При изменении ID в таблице, он автоматически обновляется в списке текстов, и наоборот.
4.2.	Способ наблюдения	Способ наблюдения определяет условие появления данной тревоги. Описание доступных способов приведено в табл. 35.
4.3.	Детали	Поле конфигурации тревоги.
4.4.	Деактивация	Переменная типа BOOL , отключающая срабатывание данной тревоги (по значению TRUE).
4.5.	Класс	Класс данной тревоги
4.6.	Сообщение	Текст сообщения, которое будет отображаться в Таблице тревог/Баннере тревог при активации тревоги. Этот текст автоматически добавляется в Список текстов тревог , заданный для группы. Помимо фиксированного текста, можно использовать заполнители. Список заполнителей приведен в табл. 36.
4.7.	Мин. время ожидания	Время задержки появления тревоги при выполнении условия активации.
4.8.	Первая триггерная переменная	Переменные, которые используются в случае необходимости записи доп. информации при активации тревоги (например, ее ID). Это реализуется при помощи библиотеки AlarmManager.library . Триггерная переменная не должна быть массивом или строкой.
4.9.	Вторая триггерная переменная	

4.10.	Тревога с большим приоритетом	<p>Здесь можно указать существующую тревогу с более высоким приоритетом по отношению к создаваемой. В этом случае при одновременном наступлении двух тревог будет автоматически подтверждаться текущая. Это позволяет выполнять двухэтапную обработку тревоги, поскольку тревога с меньшим приоритетом перекрывается тревогой с большим приоритетом.</p> <p>Пример для системы контроля температуры: вы задаете тревогу с меньшим приоритетом (например, 10) для предупреждения о температуре, превышающей 30 °С. Предварительно вы уже задали другую тревогу с более высоким приоритетом (например, 1), которая срабатывает при достижении температуры 50 °С (критическое состояние). Эту критическую тревогу можно ввести здесь, в конфигурации предупреждающей тревоги как «Тревогу с большим приоритетом». При этом существующая предупреждающая тревога будет автоматически подтверждаться при срабатывании критической тревоги.</p>
-------	-------------------------------	--

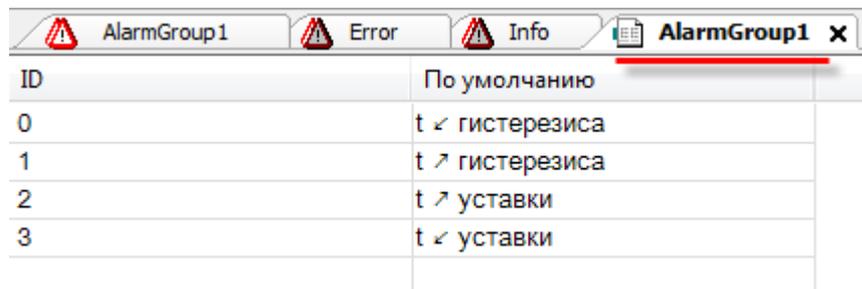
Табл. 35. Способы наблюдения тревог (см. табл. 34. пп. 4.2)

Способ наблюдения тревоги	Возможные настройки
Дискретный	Выражение: В левой части введите наблюдаемое выражение, в правой части - выражение, с которым вы хотите его сравнивать (предел); посередине выберите нужный оператор сравнения (= или <>).
Верхний предел	Выражение: то же, что и для типа «Дискретный» (см. выше), но с операторами сравнения > или >= и возможным параметром Отклонение в %¹ .
Нижний предел	Выражение: то же, что и для типа «Дискретный» (см. выше), но с операторами сравнения < или <= и возможным параметром Отклонение в %¹ .
Внутренний диапазон	Выражение: введите наблюдаемое выражение. Область: тревожная ситуация возникнет, как только наблюдаемое выражение примет значение из заданного диапазона. Слева введите выражение, определяющее нижнюю границу области, справа - верхнюю границу. Наблюдаемое выражение будет отображено в поле посередине. Выберите нужные операторы сравнения, а так же задайте при желании Отклонение в %¹ .
Внешний диапазон	Выражение: введите наблюдаемое выражение. Область: тревожная ситуация возникнет, как только наблюдаемое выражение примет значение вне заданного диапазона. Слева введите выражение, определяющее нижнюю границу области, справа - верхнюю границу. Наблюдаемое выражение будет отображено в поле посередине. Выберите нужные операторы сравнения, а также задайте при желании Отклонение в %¹ .
Изменение	Выражение: введите наблюдаемое выражение. Тревожная ситуация возникнет, как только его значение изменится.
Событие	В этом случае состояние тревоги переключается через приложение с помощью функций библиотеки AlarmManager.library .
<p>¹ Отклонение в %: если задан этот параметр, то тревожная ситуация будет иметь место до тех пор, пока не будет достигнуто определенное отклонение от указанного предельного значения. Размер отклонения задается в процентах (%) от предельного значения.</p> <p>Пример: Верхний предел: «i_temp >= 30», Отклонение: «10%». Как только значение переменной i_temp достигнет или превысит 30, возникнет тревожная ситуация. Пока значение не упадет до 27, сообщение о тревоге будет сохраняться.</p>	

Табл. 36. Заполнители сообщений тревог (см. табл. 34. пп. 4.6)

Заместитель	Значение
DATE	Дата перехода тревоги в текущее состояние.
TIME	Время последнего изменения состояния тревоги.
EXPRESSION	Выражение (задается в настройках тревоги), переключившее тревогу из одного состояния в другое.
PRIORITY	Приоритет тревоги (см. табл. 33, пп. 1).
TRIGGERVALUE ¹	Значение, вызвавшее тревогу.
ALARMID	ID тревоги (табл. 34, пп. 4.1).
CLASS	Имя класса тревог (табл. 34, пп. 4.5).
CURRENTVALUE ¹	Текущее значение наблюдаемой переменной.
LATCH1, LATCH2 ¹	Значение первой и второй триггерной переменной (см. табл. 34, пп. 4.8-4.9).
ALARM	TRUE , если состояние тревоги «активное», FALSE в любом другом случае.
STATE	Состояние тревоги: 0 - «отсутствие тревоги», 1 - «активное», 2 - «ожидание подтверждения», 3 - «активное, подтверждено»
¹ Для TRIGGERVALUE, CURRENTVALUE, LATCH1 и LATCH2 можно также использовать форматирование, например: <CURRENTVALUE %2.2f>	

Список текстов тревог представляет собой набор текстовых сообщений, используемых для отображения в **Таблице тревог**.



ID	По умолчанию
0	t < гистерезиса
1	t > гистерезиса
2	t > уставки
3	t < уставки

Рис. 8.92. Внешний вид **Списка текстов тревог**

Компонент **Хранилище тревог** содержит настройки хранения файла тревог.

Файл, в который записывается информация о тревогах, хранится во **внутренней** памяти целевого устройства. Пользователь не может изменять его имя, поскольку оно автоматически формируется из имени приложения следующим образом: **<имя приложения>.<имя хранилища тревог>.sqlite**. Использование файла записи определяется для классов (галочка **Архивация**) и групп тревог (указание **Хранилища**).

Хранилище тревог имеет следующие настройки:

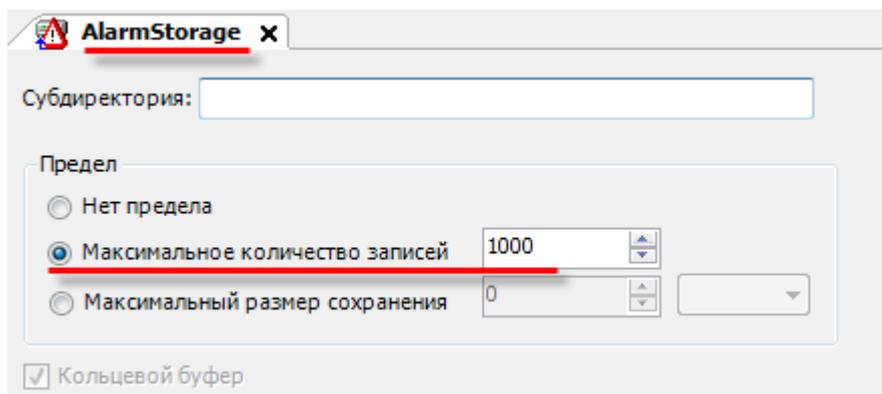


Рис. 8.93. Внешний вид **Хранилища тревог**

1. **Субдиректория** – это папка целевого устройства, расположенная по адресу, где будет сохраняться файл история тревог;
2. **Предел** позволяет ограничить количество записей в базе данных – либо по максимальному количеству, либо по размеру файла записи. При превышении ограничения старые записи начнут удаляться (**кольцевой буфер**).

При выполнении **заводского сброса (Онлайн – Сброс заводской)** файл истории тревог удаляется.

8.8.2. Таблица тревог

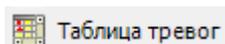


Рис. 8.94. Пиктограмма графического примитива Таблица тревог

Временная отметка	Сообщение	Изображение	Приоритет
24.02.2015 09:55:51	t / гистерезиса		100
24.02.2015 09:55:50	t / гистерезиса		100
24.02.2015 09:55:49	t / гистерезиса		100
24.02.2015 09:55:48	t / гистерезиса		100
24.02.2015 09:55:47	t / гистерезиса		100
24.02.2015 09:55:46	t / гистерезиса		100
24.02.2015 09:55:43	t / гистерезиса		100
24.02.2015 09:48:02	t / уставки		10
24.02.2015 09:47:46	t / гистерезиса		100
24.02.2015 09:47:39	t / гистерезиса		100
24.02.2015 09:47:39	t / гистерезиса		100
24.02.2015 09:47:38	t / гистерезиса		100
24.02.2015 09:47:37	t / гистерезиса		100
24.02.2015 09:47:36	t / гистерезиса		100
24.02.2015 09:47:35	t / гистерезиса		100

Рис. 8.95. Внешний вид элемента **Таблица тревог** (режим просмотра истории)

Графический примитив **Таблица тревог** используется для отображения состояний аварийных сигналов (тревог) с возможностью просмотра истории их срабатываний. **Обратите внимание**, что в обычном режиме в таблице отображаются только активные тревоги. У пользователя имеется возможность переключать таблицу в режим просмотра истории, чтобы увидеть прошедшие тревоги.

В процессе разработки экрана визуализации, столбец таблицы можно продублировать, нажав на его заголовок **ЛКМ** при зажатой клавише **Ctrl**, или удалить, нажав на его заголовок при зажатых клавишах **Ctrl** и **Shift**.

Данный графический примитив используется совместно с компонентом **Конфигуратор тревог**. Описание компонента приведено в [п. 8.8.1](#).

Описание типичных параметров графического примитива приведено в [табл. 2](#).

Уникальные параметры примитива описаны в табл. 37.

Пример работы с элементом приведен в [п. 11.2.13](#).

Табл. 37. Уникальные параметры графического примитива **Таблица тревог**

№	Название параметра	Описание параметра
1.	Конфигурация тревоги	
1.1.	Группы тревог	Группы тревог (см. табл. 33), тревоги которых будут отображаться элементом.
1.2.	Приоритет из	Минимальный приоритет тревог, отображаемых элементом.
1.3.	Приоритет для	Максимальный приоритет тревог, отображаемых элементом.
1.4.	Классы тревог	Классы тревог (см. табл. 34), тревоги которых будут отображаться данным элементом.
2.	Общая конфигурация элемента	
2.1.	Показать заголовок строки	При наличии галочки, у элемента присутствует столбец с номерами строк.
2.2.	Показать заголовок столбца	При наличии галочки, у элемента присутствует строка с заголовками столбцов.
2.3.	Высота строки	Высота строки элемента в пикселях.
2.4.	Ширина строки заголовка	Ширина столбца номеров строк элемента в пикселях.
2.5.	Размер полосы прокрутки	Ширина полосы прокрутки в пикселях.
2.6.	Автоматический разрыв строки в сообщениях	При наличии галочки, если сообщение тревоги не помещается в одну строку ячейки, то переносится на следующую.
3.	Столбцы	
	Данная вкладка позволяет создавать новые/удалять существующие столбцы элемента и настраивать их параметры.	
3.1.	Заголовок столбца	Заголовок столбца.
3.2.	Использовать выравнивание текста в заголовке	При наличии галочки, заголовок столбца выравнивается в соответствии со значением, указанным в пп. 3.5.
3.3.	Ширина	Ширина столбца в пикселях.
3.4.	Тип данных	<p>Тип данных, отображаемых в столбце. Возможные значения:</p> <p>Изображение – пиктограмма тревоги (см. табл. 33, пп. 5.4);</p> <p>Временная отметка – дата и время последнего изменения состояния тревоги;</p> <p>Временная отметка неактивна – дата и время последней активации тревоги;</p> <p>Временная отметка активна – дата и время последней деактивации тревоги;</p>

		<p>Подтверждение временной отметки – дата и время последнего подтверждения тревоги;</p> <p>Значение – значение переменной тревоги (см. табл. 34 пп. 4.3);</p> <p>Сообщение – сообщение тревоги (см. табл. 34 пп. 4.6);</p> <p>Приоритет – приоритет тревоги (см. табл. 33 пп. 1);</p> <p>Класс – класс тревоги (см. табл. 34 пп. 4.5);</p> <p>Состояние – состояние тревоги (см. рис. 8.90).</p>
3.5.	Выравнивание текста	<p>Тип выравнивания текста в столбце. Применяется в случае наличия галочки в пп. 3.2. Возможные значения:</p> <p>Лево/Право/По центру</p>
4.	Выбор	
4.1.	Цвет выбора	Цвет выделения ячеек элемента.
4.2.	Фрейм вокруг выбранных ячеек	При наличии галочки, вокруг выделенных ячеек элемента отрисовывается контур.
4.3.	Переменная для выбранной группы тревог	Переменная типа STRING , содержащая имя группы тревоги выделенной ячейки.
4.4.	Переменная для выбранного ID тревог	Переменная типа STRING , содержащая имя списка текстов и ID выделенной тревоги.
5.	Управляющие переменные	
	<p>Данная вкладка содержит переменные типа BOOL, с помощью изменения значений которых можно управлять таблицей тревог. Эти переменные можно сформировать автоматически, выбрав в контекстом меню элемента (открывается по нажатию ПКМ) команду Вставить элементы для подтверждения тревог. Вместе с переменными также будут созданы кнопки для управления ими.</p>	
5.1.	Подтвердить выбранное	Переменная типа BOOL , используемая для подтверждения выделенной тревоги (по значению TRUE).
5.2.	Подтвердить все видимое	Переменная типа BOOL , используемая для подтверждения всех тревог, отображаемых элементом (по значению TRUE).
5.3.	История	<p>Переменная типа BOOL, определяющая режим работы элемента:</p> <p>FALSE – режим текущих значений (отображаются только активные тревоги)</p> <p>TRUE – режим истории (отображаются все когда-либо произошедшие тревоги)</p>
5.4.	Фиксировать позицию прокрутки	Переменная типа BOOL , используемая для отмены автоматической прокрутки таблицы к первой строке в режиме истории при появлении новой тревоги (по значению TRUE).

8.8.3. Баннер тревог

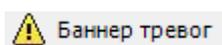


Рис. 8.96. Пиктограмма графического примитива **Баннер тревог**



Рис. 8.97. Внешний вид элемента **Баннер тревог**

Графический примитив **Баннер тревог** представляет собой частный случай [Таблицы тревог](#), используемый для отображения только одной тревоги (последней активной или наиболее важной).

Параметры графического примитива совершенно идентичны параметрам [Таблицы тревог](#), за исключением **Критерия фильтра** – данный параметр определяет, какая тревога будет отображаться элементом. Возможные значения: **Наиболее важный/Новейший**.

9. Отображение значений переменных. Форматы вывода и их спецификаторы

9.1. Отображение значений переменных

Все [Базовые](#) и часть [Стандартных](#) графических примитивов способны отображать значения привязанных к ним переменных в цифровом или текстовом виде. Свойства таких примитивов содержат вкладки **Текстовые переменные** и **Текст**:

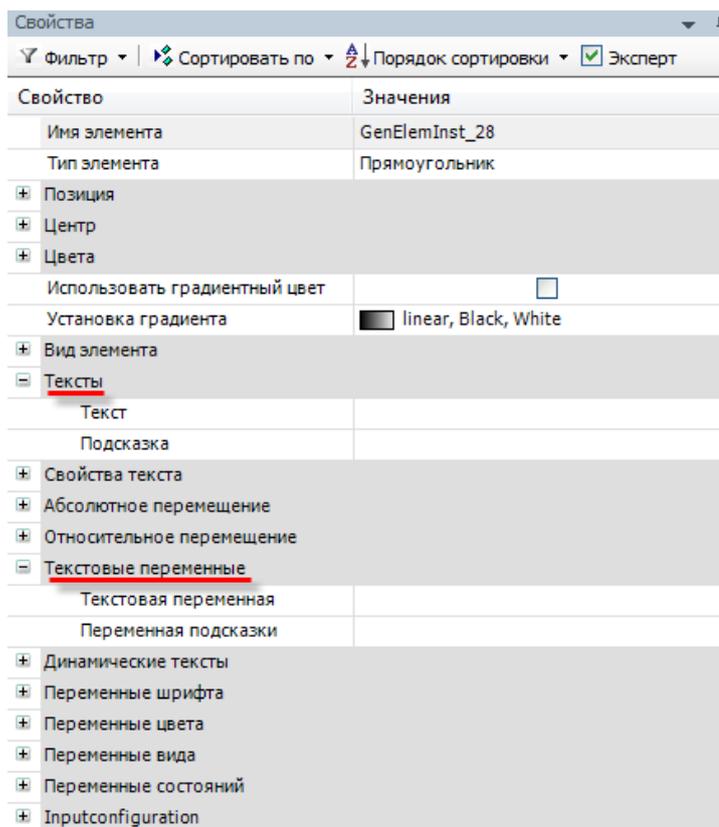


Рис. 9.1. Свойства элемента, вкладки **Текстовые переменные** и **Текст**

Каждая из этих вкладок содержит два параметра:

1. **Текстовая переменная/Текст** – переменная (любого стандартного типа данных), значение которой будет отображаться элементом/[спецификатор формата вывода](#) этой переменной;
2. **Переменная подсказки/Подсказка** – переменная (любого стандартного типа данных), значение которой будет отображаться элементом только при наведении на него курсора/[спецификатор формата вывода](#) этой переменной.

Каждый элемент может отображать только одно значение только одной текстовой переменной и одно значение переменной подсказки (например, невозможно на одной строке текста элемента отображать значение переменной в представлении с плавающей точкой, а на другой – в экспоненциальной форме).

9.2. Спецификаторы формата вывода переменных

Синтаксис заполнения параметров **Текст/Подсказка** следующий (в фигурных скобках указаны опциональные заполнители):

`{<Текст>} %[.<точность>]<спецификатор формата вывода переменной> {<Текст>}`

где

<Текст> – вспомогательный текст (например, название и размерность переменной);

% – спецсимвол, после которого указывается тип отображаемой переменной (перед типом могут быть указаны настройки формата вывода). Если необходимо отображать символ процента в качестве текста, его следует записать как %%;

<точность> – количество выводимых символов после запятой (для целочисленных переменных), количество выводимых символов (для строковых переменных);

<спецификатор формата вывода переменной> – символ формата вывода отображаемой переменной. Список символов приведен в табл. 38:

Табл. 38. Спецификаторы типов переменных

Символ	Тип <u>отображаемых</u> данных	Пример использования		
		Фактическое значение	Форматирование	Отображаемое значение
%d	десятичное число	10	%d	10
%b	двоичное число	10	%b	00000000 00001010
%o	восьмеричное число без знака (без ведущего нуля)	10	%o	12
%x	шестнадцатеричное число без знака (без ведущего нуля)	10	%x	a
%u	десятичное число без знака	-10	%u	10
%c	символ таблицы ASCII (переменная типа byte)	33	%c	!
%s	строка	abcdef	%s %.3s	abcdef abc
%f	действительное число с плавающей точкой	10.1111	%f %2.2f	10.111100 10.11
%e	действительное число с плавающей точкой (переменная типа REAL)	7389056099	%.3e	7.389e+009
%g	действительное число с плавающей точкой (переменная типа REAL)	99990000 100000000 7389056099	%g %g %.3g	99990000.00000 1.000000e+008 7.389e+009
%t[формат времени] см. п. 9.4.	время/системное время	12:48:52	%t[HH:mm:ss]	12:48:52
%%	символ процента	-	%%	%

9.3. Управляющие последовательности для STRING переменных

Список доступных управляющих последовательностей для переменных типа **STRING**, которые используются при работе с текстом (переход на новую строку, возврат каретки и т.д.), приведен в табл. 39:

Табл. 39. Управляющие последовательности для **STRING** переменных

Символ	Результат использования/Отображаемое значение
\$\$	\$ (символ доллара)
\$'	' (апостроф)
\$L	перевод строки
\$N	новая строка
\$R	возврат каретки
\$P	новая страница
\$T	табуляция
\$xx, где xx - число	символ таблицы ASCII (xx – код символа в HEX)

9.4. Системное время в визуализации

При использовании спецификатора **%t[формат времени]** можно отображать значения переменных типа **TIME, DATE_AND_TIME (DT)** и т.д. **Обратите внимание**, что использование спецификатора **%t** без форматирования бесполезно.

Если необходимо отображать в элементе **системное время** целевого устройства визуализации, то параметр **Текстовая переменная** следует оставить **пустым**.

Список заполнителей формата времени приведен в табл. 40:

Табл. 40. Заполнители формата времени

Заполнитель	Отображаемое значение	Пример отображения
ddd	Сокращенное название дня недели	Fri (пятница)
dddd	Полное имя дня недели	Monday (понедельник)
dddddd	День недели в виде числа	0 (воскресение), 1 (понедельник)
MMM	Сокращенное название месяца	Feb (февраль)
MMMM	Полное название месяца	February (февраль)
d	День в виде числа (1 – 31)	8
dd	День с ведущим нулем (01 – 31)	08
M	Месяц в виде числа (1 – 12)	8
MM	Месяц с ведущим нулем (01 – 12)	08
jjj	День в году с ведущим нулем (001-366)	253
y	Год века (0 – 99)	8
yy	Год века с ведущим нулем (00 – 99)	08
yyyy	Год	2008
HH	Час в 24-часовом формате (01– 24)	08
hh	Час в 12-часовом формате (01 – 12)	08 (и для 8-00, и для 20-00)
m	Минуты (0 – 59)	8
mm	Минуты с ведущим нулем (00 – 59)	08
s	Секунды (0 – 59)	8
ss	Секунды с ведущим нулем (00 – 59)	08
ms	Миллисекунды (0 – 999)	888
t	Идентификатор для 12-часового формата: A (часы <12) и P (часы >12)	A (8 часов)
tt	Идентификатор для 12-часового формата: AM (часы <12) и PM (часы >12)	PM (15 часов)

Если помимо времени необходимо отображать сопроводительный текст, то следует заключать этот текст в одиночные кавычки (например, **%t[dd 'days' hh 'hours']**).

9.5. Пример отображения значения переменной

Объявим в **Глобальном списке переменных** переменную **iVar** типа **REAL** с начальным значением **11.2233**:

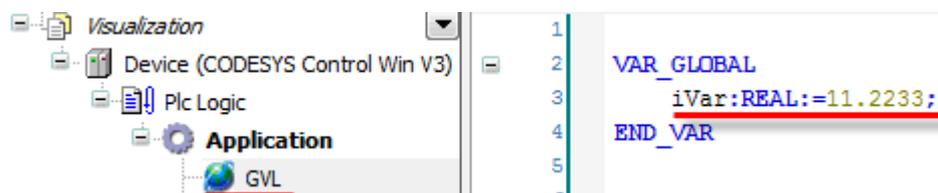


Рис. 9.2. Объявление переменной в **Глобальном списке переменных**

Привяжем эту переменную к параметрам **Текстовая переменная** и **Переменная подсказки** графического элемента типа **Прямоугольник**. Параметру **Текст** зададим форматирование **%.3f** (действительно число с тремя знаками после запятой), а параметру **Подсказка** - **%d** (десятичное число без плавающей точки):

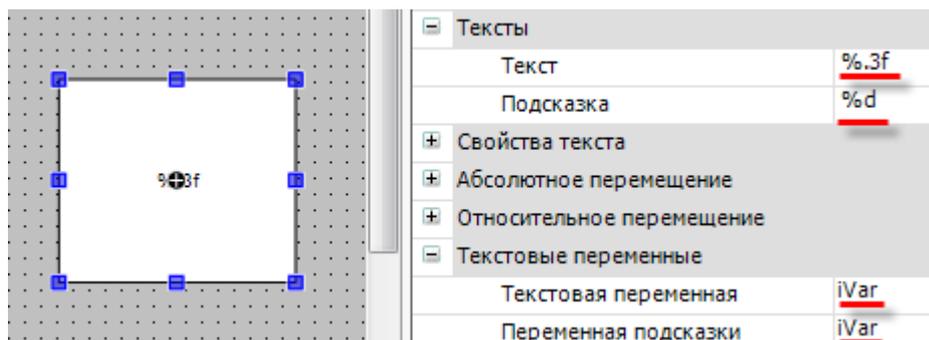


Рис. 9.3. Настройка параметров элемента

В результате, во время работы программы элемент будет отображать следующие значения:



Рис. 9.4. Отображение значения переменной в визуализации

10. Привязка действий к элементам визуализации

10.1. Вкладка InputConfiguration

Некоторые элементы визуализации по умолчанию могут реагировать на нажатие: например, [Переключатель](#) меняет состояние привязанной к нему логической переменной. К другим элементам можно привязать различные действия (см. п. 10.2) с помощью настройки вкладки **InputConfiguration**. Эта вкладка присутствует у всех [Базовых](#) графических примитивов, а так же [Стандартных](#) графических примитивов [Кнопка](#), [Текстовое поле](#), [Невидимый вход](#). **Обратите внимание**, что эта вкладка также присутствует у любого [сгруппированного](#) элемента.

Inputconfiguration	
OnDialogClosed	Конфигурация...
OnMouseClicked	Конфигурация...
OnMouseDown	Конфигурация...
OnMouseEnter	Конфигурация...
OnMouseLeave	Конфигурация...
OnMouseMove	Конфигурация...
OnMouseUp	Конфигурация...
Нажать	
Переменная	
Переключить на FALSE	<input type="checkbox"/>
Переключить на вход, ес..	<input type="checkbox"/>
Переключить	
Переменная	
Переключить вверх, есл...	<input type="checkbox"/>
Горячая клавиша	
Клавиша	

Рис. 10.1. Параметры вкладки **InputConfiguration**

Описание вкладки **InputConfiguration** приведено в табл. 41:

Табл. 41. Параметры вкладки **InputConfiguration**

№	Название параметра	Описание параметра
1.	OnDialogClosed	Действие, привязанное к элементу, выполняется после закрытия диалога, открытого или в результате нажатия на элемент.
2.	OnMouseClicked	Действие, привязанное к элементу, выполняется после нажатия и последующего отпущения курсора в пределах элемента.
3.	OnMouseDown	Действие, привязанное к элементу, выполняется после нажатия курсора на элемент.
4.	OnMouseEnter	Действие, привязанное к элементу, выполняется после наведения курсора на элемент (изначально курсор должен находиться за пределами элемента).
5.	OnMouseLeave	Действие, привязанное к элементу, выполняется при наведении и последующего перемещения курсора за границы элемента.
6.	OnMouseMove	Действие, привязанное к элементу, выполняется при движении курсора внутри элемента.
7.	OnMouseUp	Действие, привязанное к элементу, выполняется после нажатия и последующего отпущения курсора, причем, в отличие от пп. 2, курсор может быть отпущен за пределами элемента.
8.	Нажать	
8.1.	Переменная	Переменная типа BOOL , которая переключает свое значение на TRUE на время нажатия курсора на элемент. После отпущения курсора переменной возвращается значение FALSE .
8.2.	Переключить на FALSE	По умолчанию на время зажатия курсора при наведении его на элемент, переменная, указанная в п. 8.1, меняет свое значение с FALSE на TRUE . Если переменная уже имеет значение TRUE (например, это ее начальное значение), то оно будет переключено на FALSE . При наличии галочки в данном параметре, переменная будет менять значение с TRUE на FALSE . Если переменная уже имеет значение TRUE (например, это ее начальное значение), то оно будет переключено на FALSE .
8.3.	Переключить на вход, если нажата	При наличии галочки, переменная, указанная в п. 8.1, будет менять свое состояние, если нажатый на элементе курсор без отпущения будет перемещаться за границы элемента и обратно. Иными словами: предположим, начальное значение переменной – FALSE . При зажатии курсора на элементе, переменная получит значение TRUE . При перемещении зажатого курсора за границы элемента, переменная получит значение FALSE . При возвращении зажатого курсора в пределы элемента, переменная получит значение TRUE . После отпущения курсора (неважно, в пределах элемента это произойдет, или нет), переменной будет присвоено начальное значение – т.е. FALSE .

9.	Переключить	
9.1.	Переменная	Переменная типа BOOL , которая будет переключать свое значение при нажатии и последующем отпуске курсора на элементе (т.е. клике).
9.2.	Переменная вверх, если нажата	При наличии галочки, переменная, указанная в п. 9.1, будет переключаться, даже если курсор отпущен за пределами элемента.
10.	Горячая клавиша	
	Позволяет привязать клавишу клиента визуализации к элементу. Иными словами, нажатие данной клавиши будет эквивалентно нажатию курсора на элемент. Данная вкладка представляет собой часть настроек Редактора горячих клавиш .	
10.1.	Клавиша	Клавиша клиента визуализации.
10.2.	События	Событие, вызываемое нажатием на клавишу. Возможные значения: Нет – событие отсутствует; Mouse Down – нажатие курсора на элемент (см. п. 3); Mouse Up – отпущение зажатого курсора с элемента (см. п. 7); Mouse Down/Up – нажатие и последующее отпущение курсора с элемента, «клик» (см. п. 2).
10.3.	Shift	При наличии галочки, клавиша должна нажиматься с зажатой клавишей Shift .
10.4.	Control	При наличии галочки, клавиша должна нажиматься с зажатой клавишей Control .
10.5.	Alt	При наличии галочки, клавиша должна нажиматься с зажатой клавишей Alt .

10.2. Действия для OnMouse<Something>

При нажатии в **Свойствах** элемента на один из основных параметров вкладки **InputConfiguration** (см. пп. 1-7 в табл. 41), открывается окно конфигурации элемента, в котором содержится список действий, которые будут выполняться при нажатии (или других условиях) на элемент. Для этого необходимо присвоить действие элемента с помощью нажатия кнопки «>» и настроить его параметры:

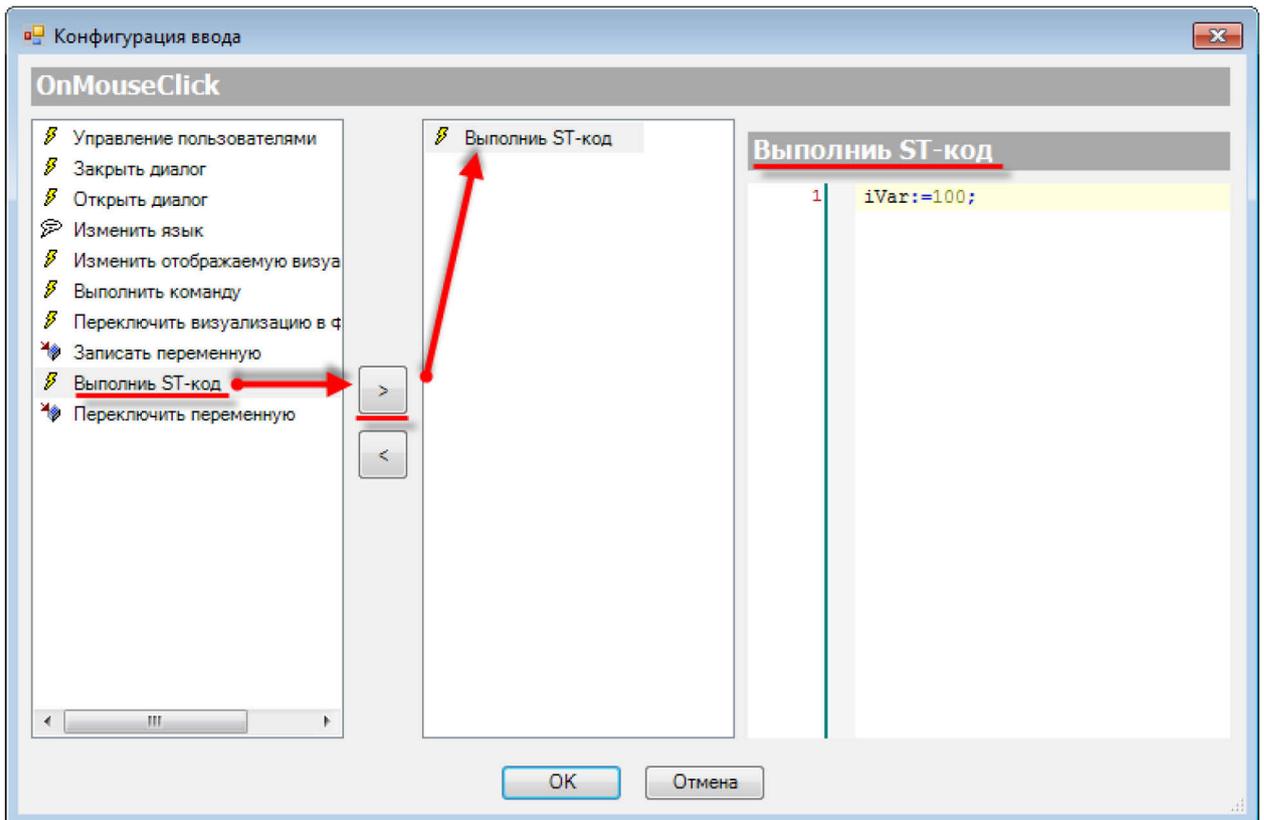


Рис. 10.2. Настройка действий элемента

Описание доступных действий приводится ниже.

10.2.1. Управление пользователями

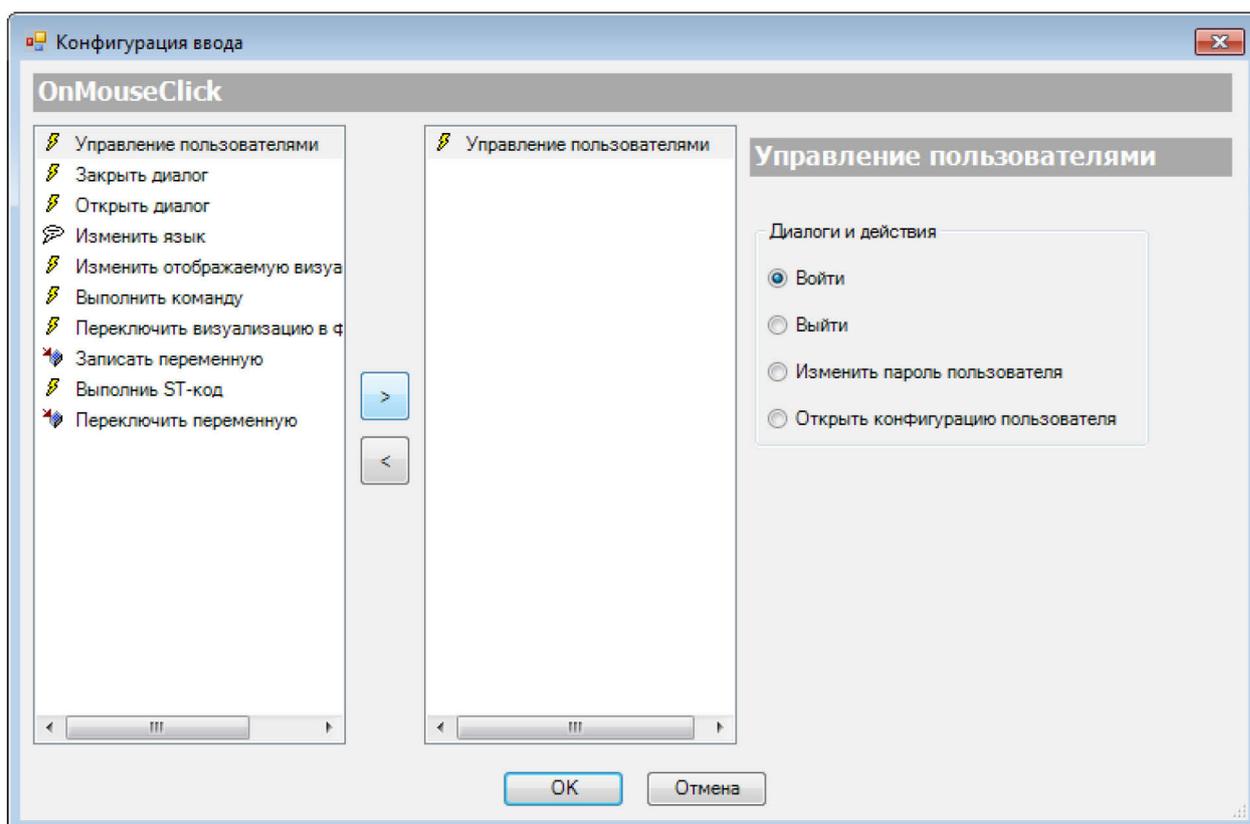


Рис. 10.3. Параметры действия **Управление пользователями**

Данная вкладка доступна только при наличии в проекте [Управления пользователями](#).

Пример создания многопользовательского проекта приведен в [п. 11.3.7](#).

Действие **Управления пользователями** используется для привязки к элементу одного из диалогов библиотеки **VisuUserManagement**:

1. **Войти** – диалог аутентификации пользователя (VUM_Login);

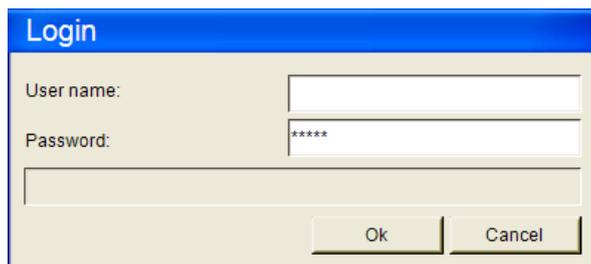


Рис. 10.4. Диалог **Войти**

2. **Выйти** – операция выхода текущего пользователя из системы;

3. **Изменить пароль пользователя** – диалог изменения пароля текущего пользователя (VUM_ChangePassword);

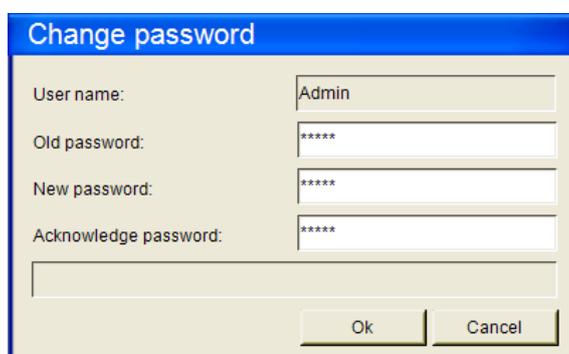


Рис. 10.5. Диалог **Выйти**

4. **Открыть конфигурацию пользователя** – диалог изменения настроек **Управления пользователями (VUM_UserManagement)**;

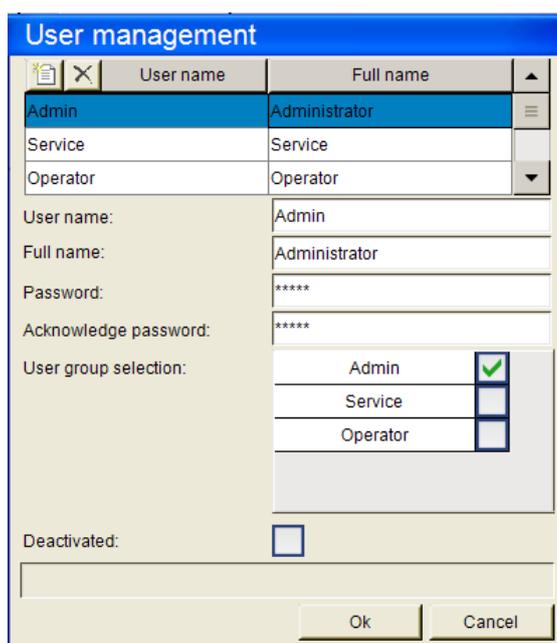


Рис. 10.6. Диалог **Изменение конфигурации пользователей**

10.2.2. Закреть диалог

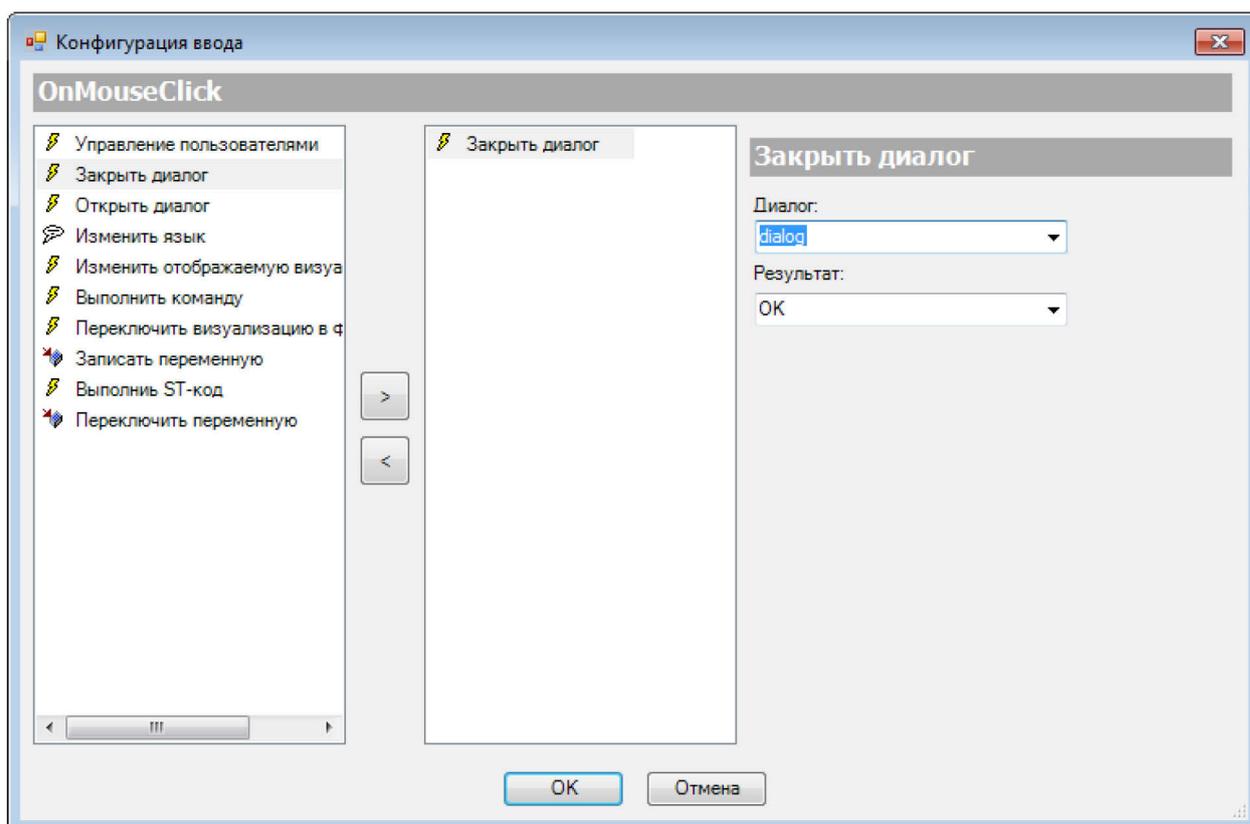


Рис. 10.7. Параметры действия **Закреть диалог**

Действие **Закреть диалог** приводит к закрытию выбранного диалогового окна. В параметрах действия можно выбрать результат закрытия диалога, в случае которого значения переменных [интерфейса](#) диалога будут присвоены переменным программы (см. п. 10.2.3).

Пример работы с диалогами приведен в [п. 11.3.2](#).

10.2.3. Открыть диалог

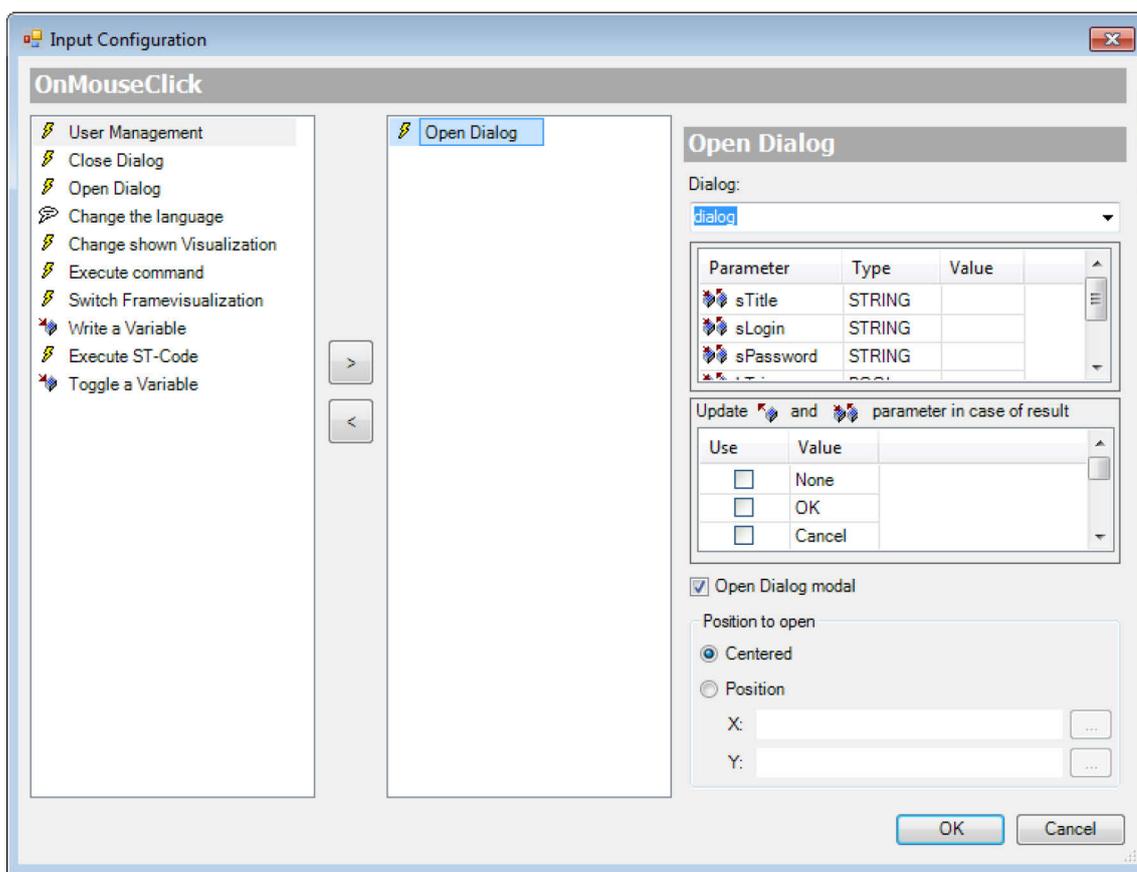


Рис. 10.8. Параметры действия **Открыть диалог**

Действие **Открыть диалог** приводит к открытию выбранного диалогового окна. Параметры действия:

1. Переменные [интерфейса](#) диалога с возможностью привязки к ним переменных программы (**Value**);
2. Результаты закрытия диалога, в случае которых происходит присвоение значений переменных диалога привязанным переменным программы (см. п. 10.2.2);
3. Галочка модальности диалога (Open Dialog Modal) – при ее наличии после открытия диалога основной экран визуализации блокируется;
4. Координаты открытия диалога – по умолчанию диалог открывается по центру экрана визуализации, но можно указать переменные типа **INT** для координат по горизонтальной и вертикальной оси соответственно.

Обратите внимание, что в данный момент (**CODESYS 3.5 SP6**) окно настройки данного действия отображается некорректно в *русскоязычной* версии **CODESYS** (*исправлено в SP7*).

Пример работы с диалогами приведен в [п. 11.3.2](#).

10.2.4. Изменить язык

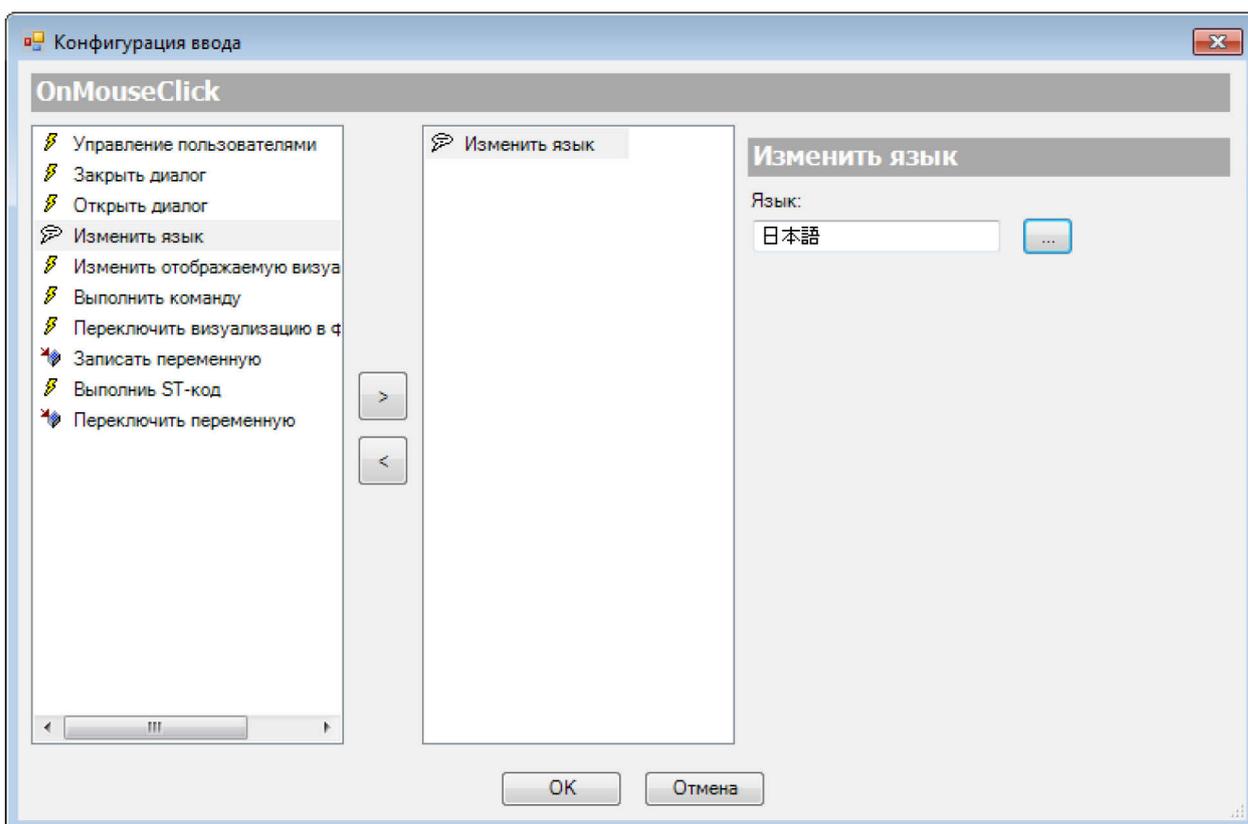


Рис. 10.9. Параметры действия **Изменить язык**

Действие **Изменить язык** приводит к смене текущего языка визуализации. Предварительно язык должен быть добавлен в [Список текстов](#).

Пример создания мультязычного проекта приведен в [п. 11.3.5](#).

10.2.5. Изменить отображаемую визуализацию

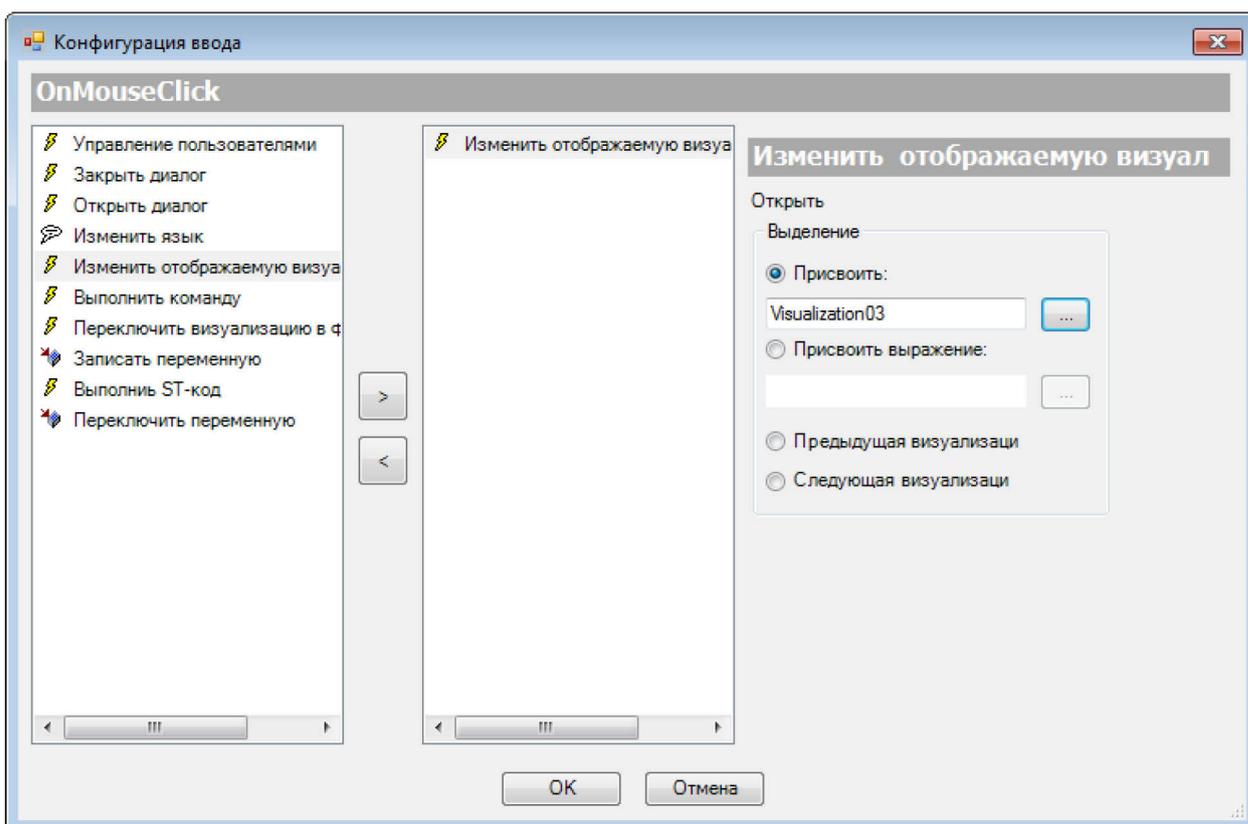


Рис. 10.10. Параметры действия **Изменить отображаемую визуализацию**

Действие **Изменить отображаемую визуализацию** приводит к смене текущего экрана визуализации. Возможные значения:

1. **Присвоить** – непосредственное указание экрана визуализации, на который будет осуществлен переход;
2. **Присвоить выражение** – переменная типа **STRING**, содержащая имя экрана визуализации, на который будет осуществлен переход;
3. **Предыдущая визуализация** – переход будет осуществлен на экран визуализации, который отображался перед переходом на текущий. Может использоваться, например, в случае вспомогательного экрана визуализации, переход на который может осуществляться с множества других экранов;
4. **Следующая визуализация** – переход будет осуществлен на экран, с которого последний раз был осуществлен переход с помощью данного действия при использовании параметра **Предыдущая визуализация**.

Пример реализации переключения экранов в проекте приведен в [п. 11.3.1](#).

10.2.6. Выполнить команду

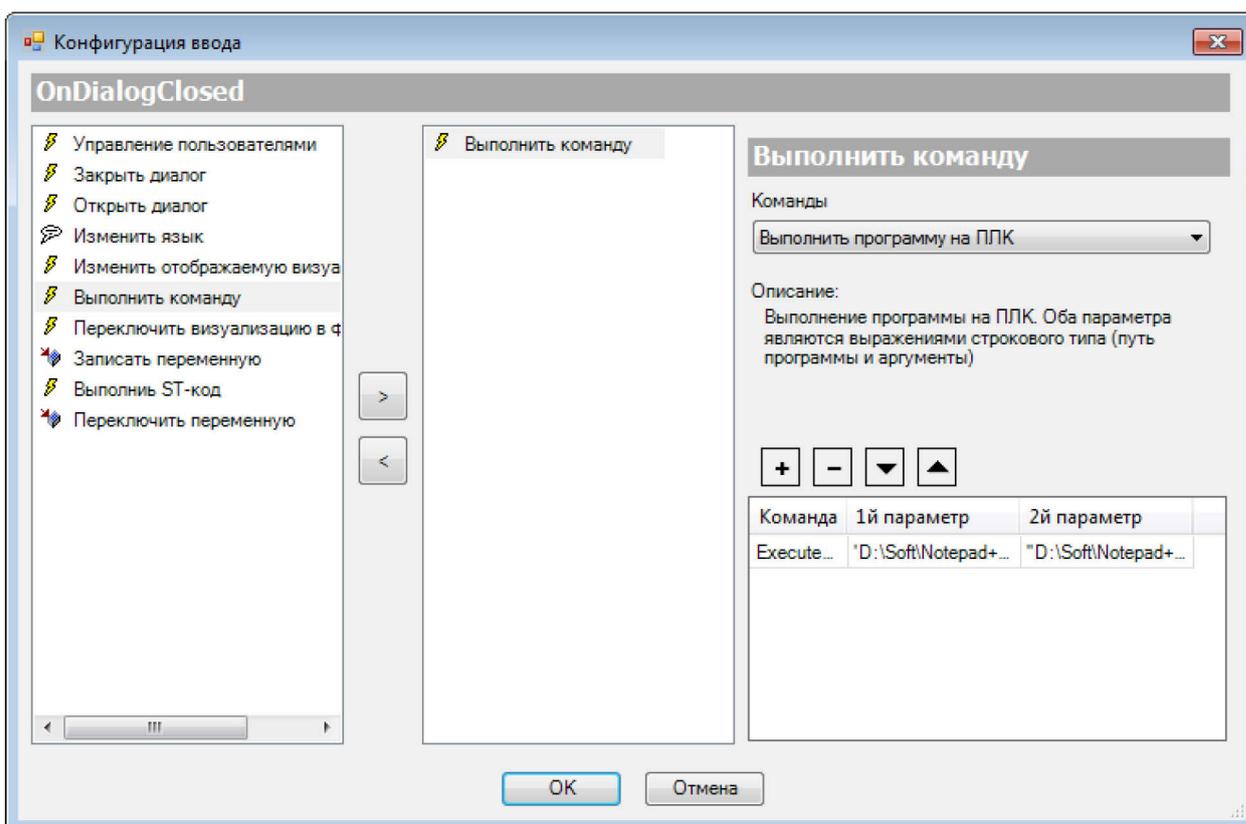


Рис. 10.11. Параметры действия **Выполнить команду**

Действие **Выполнить команду** приводит к выполнению команды (или группы команд) на целевом устройстве. Добавление команды в список выполняемых осуществляется с помощью кнопки «+», удаление – «-», изменение порядка выполнения команд – «**вверх**»/ «**вниз**». Каждая команда (за исключением **Print**) имеет два параметра для настройки, которые представляют собой значения типа **STRING** (можно также использовать конкретные значения, заключенные в кавычки).

Описание доступных команд приведено в табл. 42:

Табл. 42. Список команд действия **Выполнить команду**

Команда	Описание команды	1-й параметр	2-й параметр
Выполнить программу на ПЛК/клиенте (ExecutePlcProgram/ExecuteClientProgram)	Запуск приложения/открытие файла с помощью приложения	Путь к приложению	Путь к файлу
Пример	<p>Открытие текстового файла readme.txt с помощью приложения Notepad++.</p> <p>1-й параметр: 'D:\Soft\Notepad++\notepad++.exe' 2-й параметр: 'D:\Soft\Notepad++\readme.txt'</p> <p>Работа данного функционала гарантируется только на ОС Windows с использованием CODESYS HMI.</p>		
Печать (Print)	Печать содержимого экрана визуализации	отсутствует	отсутствует
Комментарий	Работа данного функционала гарантируется только на ОС Windows с использованием CODESYS HMI.		
Navigate to URL	Переход по ссылке в веб-клиенте визуализации.	Адрес веб-страницы	<p>При отсутствии значения, веб-страница открывается в новом окне.</p> <p>При значении 'replace', веб-страница открывается вместо веб-визуализации.</p>
Пример	<p>Открытие в новом окне страницы www.owen.ru</p> <p>1-й параметр: 'http://owen.ru' 2-й параметр: ''</p> <p>Данный функционал работает только на веб-клиентах визуализации.</p>		
Команды работы с Менеджером рецептов (см. п. 11.3.8)			
Прочитать список (ReadRecipe)	Запись значений переменных программы в список рецепта.	Имя группы рецептов.	Имя рецепта.
Записать список (WriteRecipe)	Запись значений рецепта в переменные программы.		
Загрузить список из файла (LoadRecipeFrom)	Загрузка значений рецепта из файла.		
Создать список (CreateRecipe)	Создание нового рецепта.		
Удалить список (DeleteRecipe)	Удаление существующего рецепта.		
Сохранить список в файл (SaveRecipeAs)	Сохранение значений рецепта в файл.		

10.2.7. Переключить визуализацию в фрейме

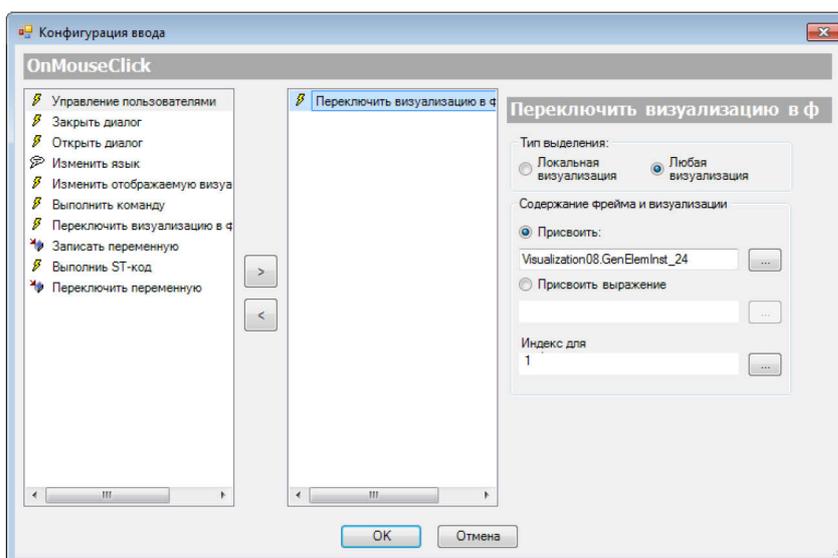


Рис. 10.12. Параметры действия **Переключить визуализацию во фрейме**

Действие **Переключить визуализацию во фрейме** приводит к смене экрана визуализации в элементе типа [Фрейм](#).

Тип выделения характеризует расположение фрейма в проекте – **Локальная визуализация** позволяет переключать только визуализации фреймов, расположенных на данном экране визуализации, в то время как **Любая визуализация** позволяет делать это с любым фреймом проекта.

При использовании **Любой визуализации** пользователь может либо выбрать переключаемый фрейм с помощью вкладки **Присвоить**, либо с помощью вкладки **Присвоить выражение** указать переменную типа **STRING**, содержащую имя экрана визуализации и фрейма (например, **Visu1.GemElementInst_01**).

В пункте **Индекс для** указывается переменная типа **INT**, содержащая номер экрана визуализации, на который будет переключен фрейм. Список экранов можно посмотреть в конфигурации фрейма (вкладка **Ссылки в Свойствах** элемента), при этом первому из них соответствует индекс **0**, второму – **1**, и т.д.

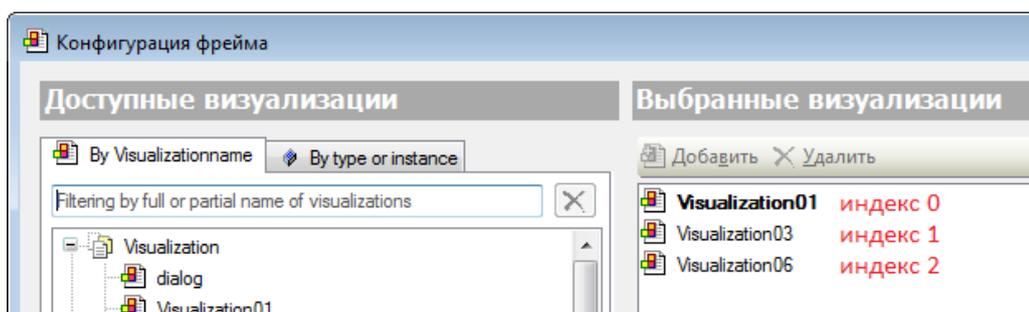


Рис. 10.13. Конфигурация фрейма

Примеры реализации переключения экранов во фрейме приведены в [п. 11.2.2](#) и [п. 11.3.3](#).

10.2.8. Записать переменную

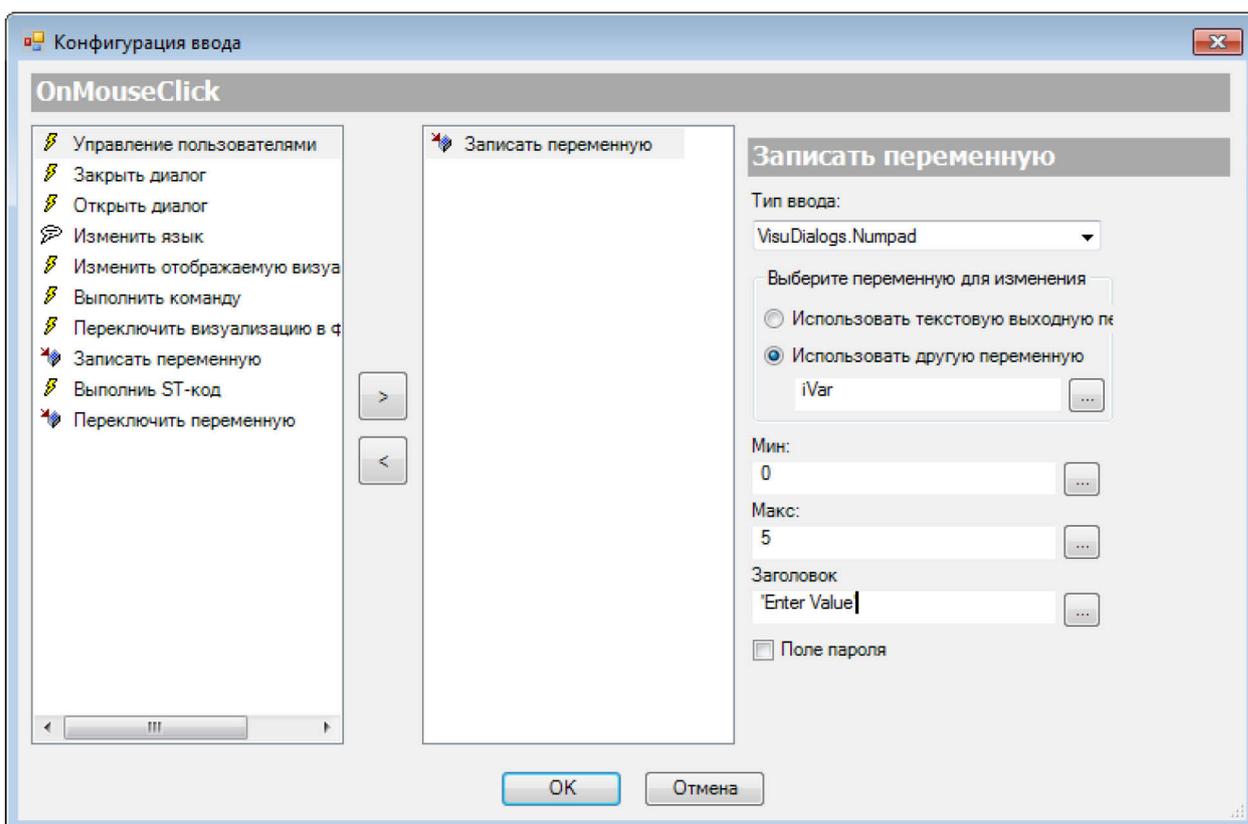


Рис. 10.14. Параметры действия **Записать переменную**

Действие **Записать переменную** используется для изменения значений строковой или числовой переменной.

Тип ввода определяет способ ввода значений:

По умолчанию – используются настройки из [Менеджера визуализации](#);

Ввод текста – используются аппаратные клавиши (например, компьютерная клавиатура);

Ввод текста с ограничениями – используются аппаратные клавиши, при этом отображаются ограничения на величину переменной (см. ниже);

VisuDialogs.Keypad – используется экранная клавиатура (диалог **Keypad** из библиотеки **VisuDialogs**);

VisuDialogs.Numpad – используется цифровая экранная клавиатура (диалог **Numpad** из библиотеки **VisuDialogs**);

В качестве изменяемой переменной можно выбрать либо текстовую переменную, привязанную к элементу (**Использовать текстовую выходную переменную**), либо указать необходимую переменную вручную (**Использовать другую переменную**).

Параметры **Мин** и **Макс** определяют, соответственно, нижний и верхний предел задаваемых значений переменной (для числовых переменных) или минимально/максимально возможное количество символов (для строковых переменных). К параметрам можно привязать как переменные, так и просто указать значения (как на рис. 10.14).

В параметре **Заголовок** можно указать переменную типа **STRING** (или выражение; в этом случае, оно должно быть заключено в кавычки), которое будет выводиться в заголовке диалогового окна (при использовании соответствующего типа ввода).

При наличии галочки в параметре **Поле пароля**, вводимое значение будет отображаться звездочками (*****).

10.2.9. Выполнить ST-код

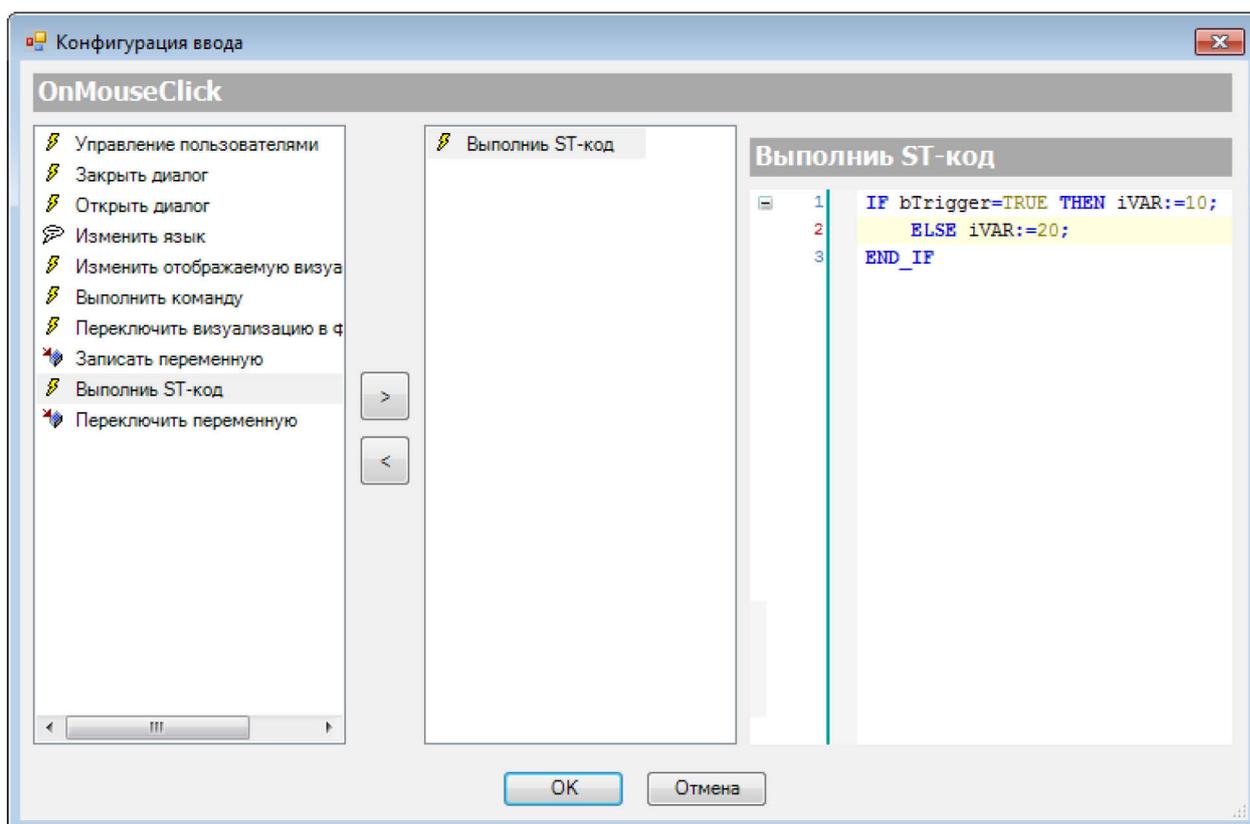


Рис. 10.15. Параметры действия **Выполнить ST-код**

Действие **Выполнить ST-код** используется для выполнения программы на языке **ST**.

10.2.10. Переключить переменную

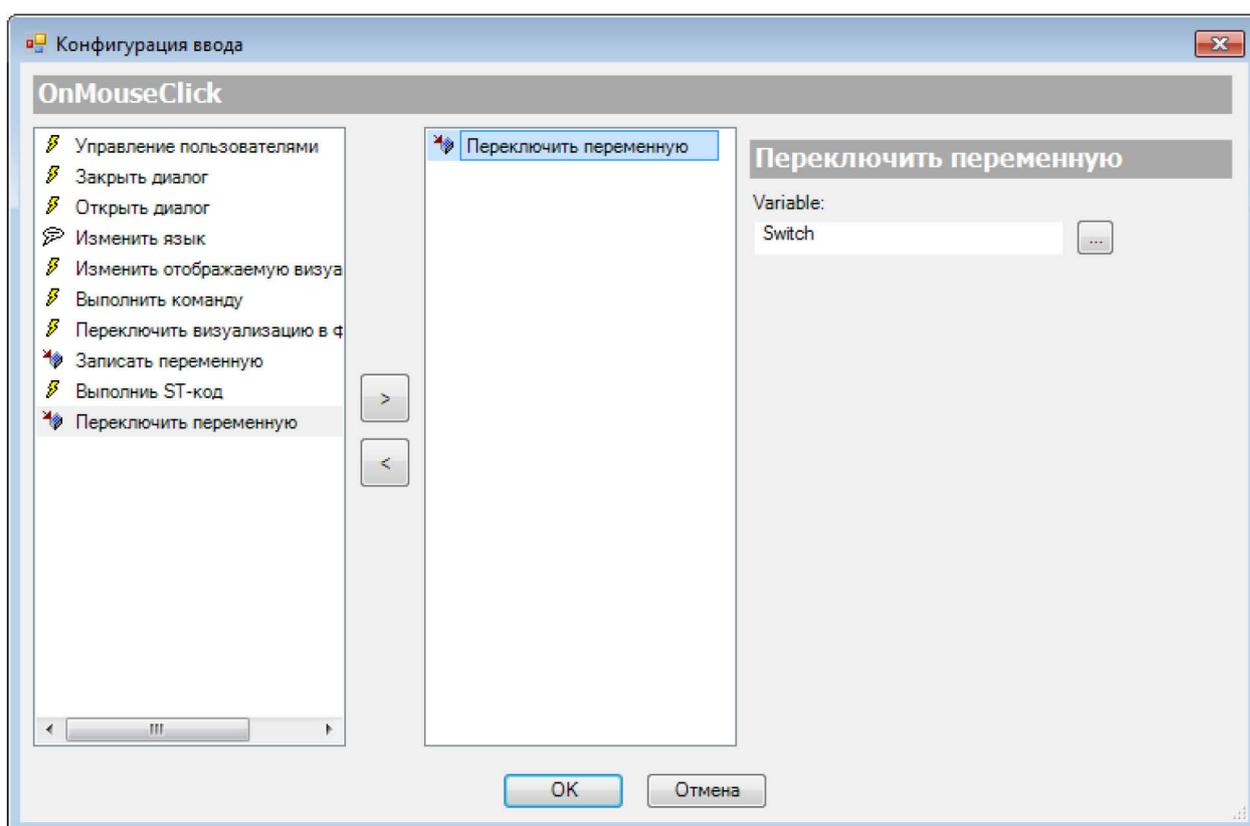


Рис. 10.16. Параметры действия **Переключить переменную**

Действие **Переключить переменную** используется для переключения значений переменной типа **BOOL**.

11. Примеры

11.1. Структура главы. Запуск примеров на виртуальном контроллере

В данной главе приведены примеры разработки экранов визуализации в **CODESYS**. Примеры делятся на две группы:

1. Примеры использования графических примитивов. Поскольку число примитивов достаточно велико, а их настройки зачастую практически идентичны, то будут рассмотрены только наиболее часто используемые (такие, как [Кнопка](#)) и сложные (такие, как [Таблица тревог](#)) примитивы.
2. Примеры использования в визуализации других компонентов ([Управление пользователями](#), [Менеджер рецептов](#) и т.д.) и решения часто возникающих задач ([переключение экранов визуализации](#), [вызов диалогов](#) и т.д.).

Каждый пример содержит инструкцию по разработке проекта и ссылку на архив готового проекта. Поскольку показать работу примеров на всех модификациях и прошивках контроллеров **СПК** не представляется возможным, примеры разрабатывались с прицелом на запуск на виртуальном контроллере **CODESYS Control Win V3**, который входит в состав среды **CODESYS** и представляет собой программную эмуляцию реального контроллера, запускаемую на ПК с ОС семейства Windows.

При необходимости, большинство примеров можно запустить на **СПК**, выбрав соответствующий target-файл. Исключение составляют только примеры, затрагивающие обширные предметные области (например, [Типичные параметры графических примитивов](#), [Текстовый редактор](#), [Таблица тревог](#), [Менеджер рецептов](#)) – размер их экранов визуализации превышает **800x480**, в связи с чем они будут некорректно отображаться на дисплее **СПК**.

После открытия проекта примера в **CODESYS**, необходимо запустить виртуальный контроллер с помощью иконки на панели задач Windows:

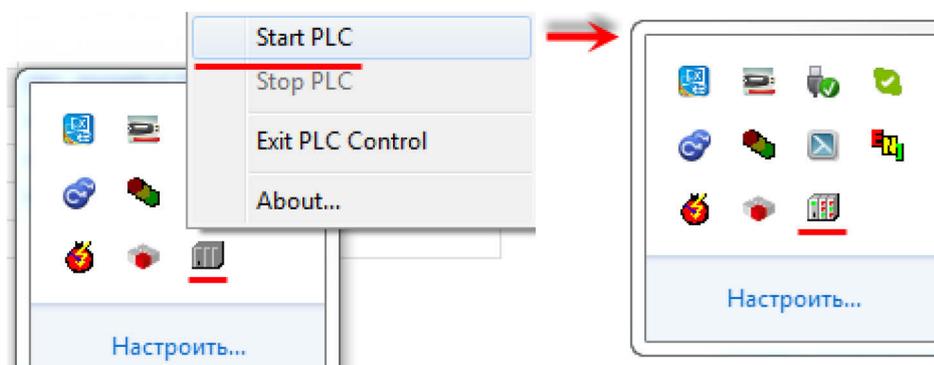


Рис. 11.1. Запуск виртуального контроллера

В установках соединения **CODESYS** произвести сканирование сети и выбрать устройство, чье имя совпадает с именем ПК:

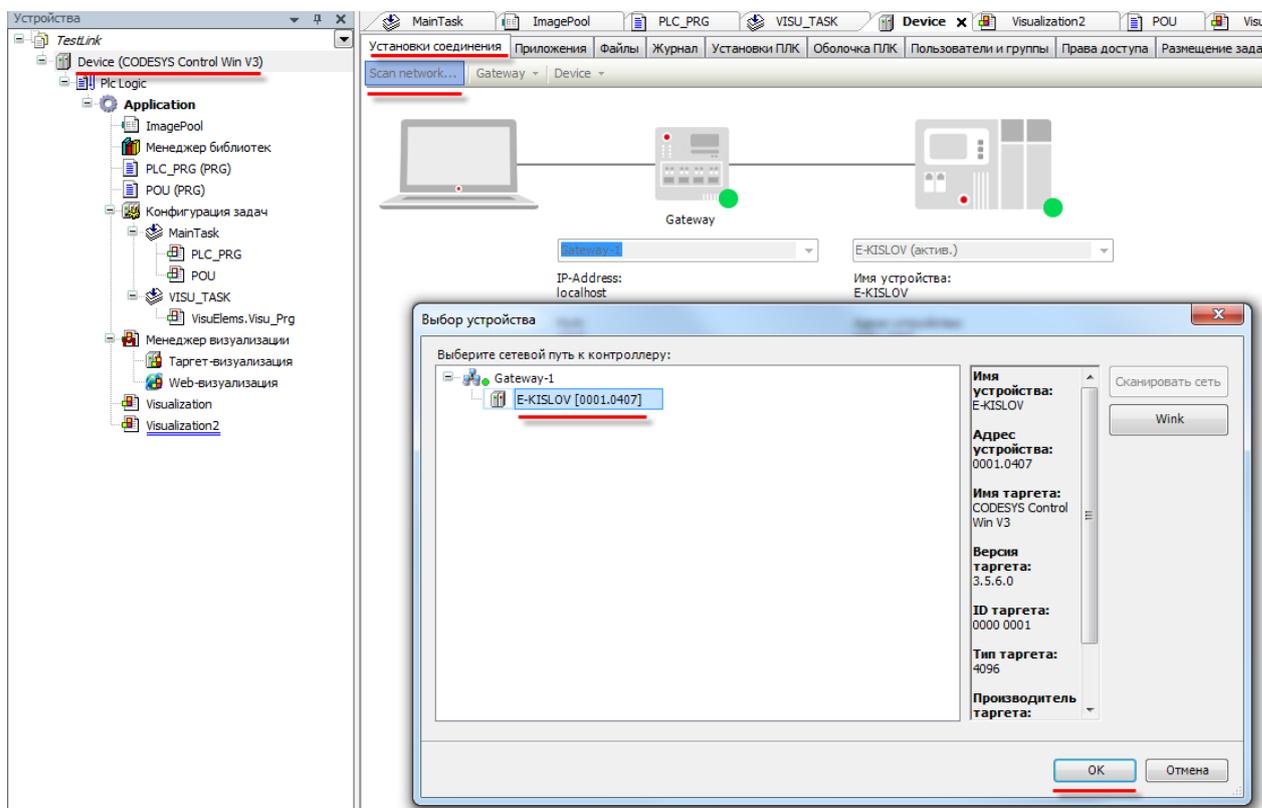


Рис. 11.2. Окно сканирования сети. Подключение к виртуальному контроллеру

Обратите внимание, что перед загрузкой проекта в виртуальный контроллер рекомендуется выполнить команды вкладки **Компиляция** – **Очистить все** и **Перекомпиляция**.

Необходимо отметить, что вместе с **CODESYS** распространяется версия виртуального контроллера с **ограничением времени** непрерывной работы (около получаса). После этого контроллер можно перезапустить. При необходимости лицензию на виртуальный контроллер можно приобрести у компании [3S](#) (разработчик **CODESYS**).

11.2. Примеры работы с графическими примитивами

11.2.1. Типичные параметры графических примитивов

Данный пример посвящен работе с типичными параметрами графических примитивов.

В качестве рассматриваемого примитива выбран [Прямоугольник](#) – он обладает наибольшим числом параметров, к которым можно привязать переменные (около трех десятков). Помимо этого, в примере использованы примитивы [Метка](#), [Группа](#), [Текстовое поле](#), [Полоса прокрутки](#), [Бегунок](#), [Управление вращением](#), [Радио-кнопка](#), [Переключатель изображения](#), [Индикатор](#) и [Клавишный выключатель](#).

Суть примера заключается в привязке ко всем доступным для этого переменным [Прямоугольника](#) переменных программы и настройка элементов управления этими переменными.

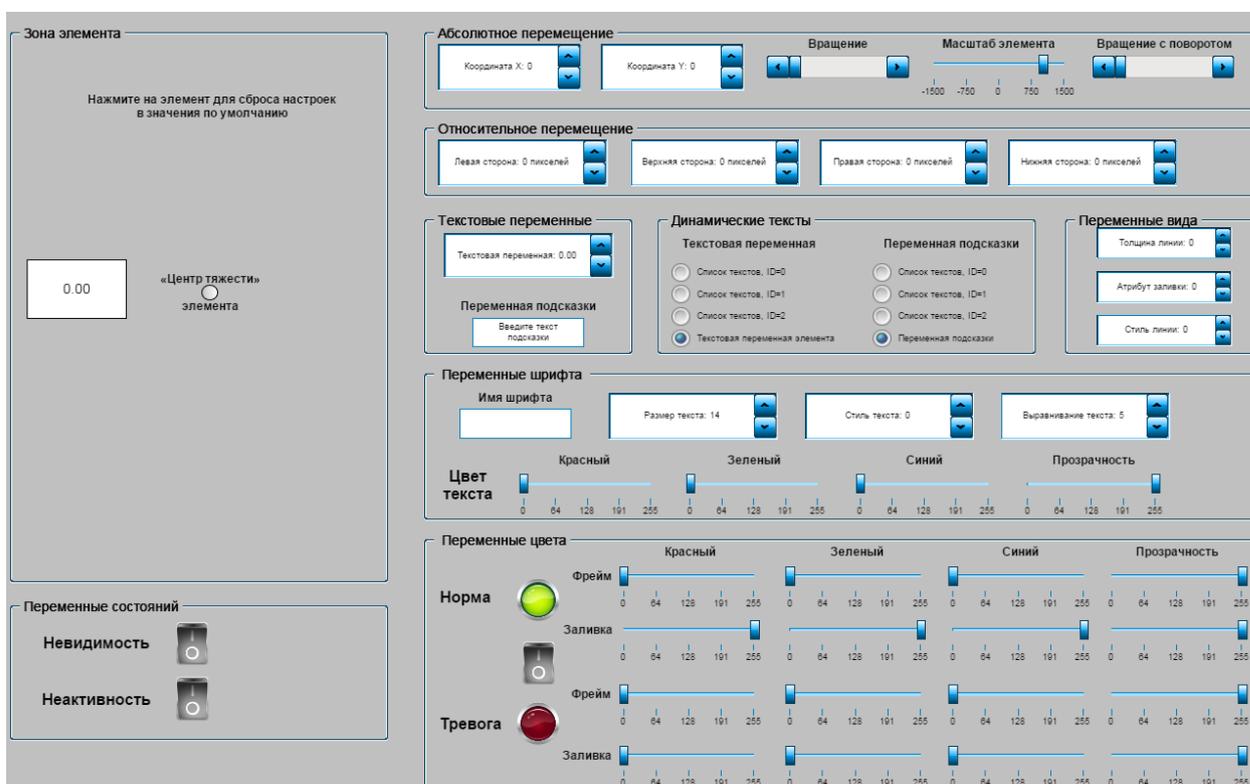


Рис. 11.2.1.1. Внешний вид примера **Типичные свойства графических примитивов**

Пример создан в среде **CODESYS 3.5 SP6 Patch 2** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с target-файлом **3.5.6.0**.

Пример доступен для скачивания: [Example_Rectangle.projectarchive](#)

1. Создадим новый *стандартный* проект CODESYS с названием Example_Rectangle и настройками по умолчанию: target – CODESYS Control Win V3, язык PLC_PRG – ST.

2. В программе PLC_PRG объявим следующие переменные:

```

1  PROGRAM PLC_PRG
2  VAR
3      // Абсолютное перемещение                // Название параметра в CODESYS
4      iCoord_X:INT;                            // X
5      iCoord_Y:INT;                            // Y
6      iRotation:INT;                          // Вращение
7      iScale:INT:=1000;                        // Масштабирование
8      iSelfRotation:INT;                      // Внутреннее вращение
9
10     // Относительное перемещение             // Название параметров в CODESYS некорректны
11     iLeftSideMovement:INT;                   // Перемещение сверху налево: X
12     iUpSideMovement:INT;                    // Перемещение сверху налево: Y
13     iRightSideMovement:INT;                 // Перемещение снизу вверх: X
14     iDownSideMovement:INT;                  // Перемещение снизу вверх: Y
15
16     // Текстовые переменные
17     rElementVariable:REAL;                  // Текстовая переменная
18     wsTooltip:WSTRING:="Введите текст$Nподсказки"; // Переменная подсказки
19
20     // Динамические тексты
21     sTextIndex:STRING;                      // Индекс текста
22     sTooltipIndex:STRING;                   // Индекс подсказки
23     iTTextIndex:INT:=3;                     // дополнительные переменные...
24     iTooltipIndex:INT:=3;                   // ...для радио-кнопок
25
26     // Переменные шрифта
27     sFontName:STRING;                       // Имя шрифта
28     iTTextSize:INT:=14;                     // Размер
29     dwTextFlag:DWORD;                       // флаги
30     // dwTextCharset:DWORD;                  // Набор символов. Не работает в 3.5 SP6
31     dwTextColor:DWORD;                      // Цвет
32     iTTextAligment:INT:=5;                  // Flags for text alignment
33
34     // Переменные цвета
35     bSwitchStateColor:BOOL;                 // Переключить цвет
36     dwNormalState_FrameColor:DWORD;         // Нормальное состояние: Цвет фрейма
37     dwNormalState_FillColor:DWORD;         // Нормальное состояние: Цвет заливки
38     dwAlarmState_FrameColor:DWORD;         // Состояние тревоги: Цвет фрейма
39     dwAlarmlState_FillColor:DWORD;         // Состояние тревоги: Цвет заливки
40
41     // Переменные вида
42     iLineThickness:INT;                     // Ширина линии
43     dwFillEnable:DWORD;                     // Атрибуты заливки
44     dwLineStyle:DWORD;                      // Стиль линии

```

Рис. 11.2.1.2. Объявление переменных программы PLC_PRG (начало)

```

51 // Дополнительные переменные для изменения цвета текста
52 // фрейма и заливки элемента (по модели RGBA)
53
54 byTextColorAlpha:BYTE:=255;
55 byTextColorRed:BYTE;
56 byTextColorGreen:BYTE;
57 byTextColorBlue:BYTE;
58
59 byNormalState_FrameColorAlpha:BYTE:=255;
60 byNormalState_FrameColorRed:BYTE:=0;
61 byNormalState_FrameColorGreen:BYTE:=0;
62 byNormalState_FrameColorBlue:BYTE:=0;
63
64 byNormalState_FillColorAlpha:BYTE:=255;
65 byNormalState_FillColorRed:BYTE:=255;
66 byNormalState_FillColorGreen:BYTE:=255;
67 byNormalState_FillColorBlue:BYTE:=255;
68
69 byAlarmState_FrameColorAlpha:BYTE:=255;
70 byAlarmState_FrameColorRed:BYTE;
71 byAlarmState_FrameColorGreen:BYTE;
72 byAlarmState_FrameColorBlue:BYTE;
73
74 byAlarmState_FillColorAlpha:BYTE:=255;
75 byAlarmState_FillColorRed:BYTE;
76 byAlarmState_FillColorGreen:BYTE;
77 byAlarmState_FillColorBlue:BYTE;

```

Рис. 11.2.1.3. Объявление переменных программы **PLC_PRG** (продолжение)

3. У прямоугольника есть параметры, определяющие цвет его текста, контура и заливки. К этим параметрам привязываются переменные типа **DWORD**. Переменная определяет цвет по модели **RGBA**. Но управлять цветом с помощью только одной переменной крайне неудобно; поэтому предусмотрим возможность изменения каждого из оттенков (как в большинстве графических редакторов) – красного, синего, зеленого + прозрачности (альфа-канала). Для каждой цветовой **DWORD** переменной, которые будут привязаны к прямоугольнику, нам потребуются четыре переменные типа **BYTE**, которые будут привязаны к управляющим элементам визуализации (см. рис. 11.2.1.3).

Чтобы собирать цвет из оттенков, создадим функциональный блок **RGBA_TO_DWORD** на языке **ST**:

```

1 FUNCTION_BLOCK RGBA_TO_DWORD
2 VAR_INPUT
3   byAlpha:BYTE;
4   byRed:BYTE;
5   byGreen:BYTE;
6   byBlue:BYTE;
7 END_VAR
8 VAR_OUTPUT
9   dwColor:DWORD;
10 END_VAR
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Рис. 11.2.1.4. Создание функционального блока сборки цвета

4. Объявим в **PLC_PRG** экземпляры созданного нами функционального блока:

```
77     TextColor:RGBA_TO_DWORD;  
78     NormalState_FrameColor:RGBA_TO_DWORD;  
79     NormalState_FillColor:RGBA_TO_DWORD;  
80     AlarmState_FrameColor:RGBA_TO_DWORD;  
81     AlarmState_FillColor:RGBA_TO_DWORD;  
82  
83     END_VAR  
84  
85
```

Рис. 11.2.1.5. Объявление переменных программы **PLC_PRG** (окончание). Объявление экземпляров функционального блока **RGBA_TO_DWORD**

5. Код программы **PLC_PRG** будет выглядеть следующим образом:

```
1  
2     // для управления динамическими текстами  
3     sTextIndex:=INT_TO_STRING(iTextIndex);  
4     sTooltipIndex:=INT_TO_STRING(iTooltipIndex);  
5  
6     // для управления цветами  
7  
8     TextColor // собираем цвет текста  
9     (  
10    byAlpha:=byTextColorAlpha,  
11    byRed:=byTextColorRed,  
12    byGreen:=byTextColorGreen,  
13    byBlue:=byTextColorBlue,  
14    dwColor=>dwTextColor  
15    );  
16  
17    NormalState_FrameColor // собираем цвет фрейма в нормальном состоянии  
18    (  
19    byAlpha:=byNormalState_FrameColorAlpha,  
20    byRed:=byNormalState_FrameColorRed,  
21    byGreen:=byNormalState_FrameColorGreen,  
22    byBlue:=byNormalState_FrameColorBlue,  
23    dwColor=>dwNormalState_FrameColor  
24    );  
25
```

Рис. 11.2.1.6. Код программы **PLC_PRG** (начало)

```

26
27 NormalState_FillColor // собираем цвет заливки в нормальном состоянии
28 (
29   byAlpha:=byNormalState_FillColorAlpha,
30   byRed:=byNormalState_FillColorRed,
31   byGreen:=byNormalState_FillColorGreen,
32   byBlue:=byNormalState_FillColorBlue,
33   dwColor=>dwNormalState_FillColor
34 );
35
36 AlarmState_FrameColor // собираем цвет фрейма в состоянии тревоги
37 (
38   byAlpha:=byAlarmState_FrameColorAlpha,
39   byRed:=byAlarmState_FrameColorRed,
40   byGreen:=byAlarmState_FrameColorGreen,
41   byBlue:=byAlarmState_FrameColorBlue,
42   dwColor=>dwAlarmState_FrameColor
43 );
44
45 AlarmState_FillColor // собираем цвет заливки в состоянии тревоги
46 (
47   byAlpha:=byAlarmState_FillColorAlpha,
48   byRed:=byAlarmState_FillColorRed,
49   byGreen:=byAlarmState_FillColorGreen,
50   byBlue:=byAlarmState_FillColorBlue,
51   dwColor=>dwAlarmState_FillColor
52 );

```

Рис. 11.2.1.7. Код программы **PLC_PRG** (окончание)

Программа представляет собой вызовы «цветовых» функциональных блоков и преобразование **STRING_TO_INT** для переменных динамических текстов. Подробнее это преобразование будет рассмотрено в пп. 15.

6. Программа **PLC_PRG** по умолчанию привязана к задаче **MainTask**. Настроим частоту вызова задачи (**t#10ms**):

Конфигурация

Приоритет (0..15): 0 1

Тип: Циклическое

Интервал (напр., t#200ms): t#10ms

Сторожевой таймер

Включить

Время (напр., t#200ms):

Восприимчивость: 1

Добавить вызов
 Удалить вызов
 Изменить вызов
 Переместить вверх
 Переместить вниз
 Открыть POU

POU	Комментарий
PLC_PRG	

Рис. 11.2.1.8. Настройки частоты вызова задачи **MainTask**

7. [Добавим в проект экран визуализации](#) с названием **Visualization**. В его [свойствах](#) выберем размер **1760x1080**. Подразумевается, что проект будет запускаться на виртуальном контроллере с отображением в веб-браузере.

Создание экрана визуализации приведет к автоматическому добавлению в проект [Менеджера визуализации](#). Зададим ему следующие настройки:

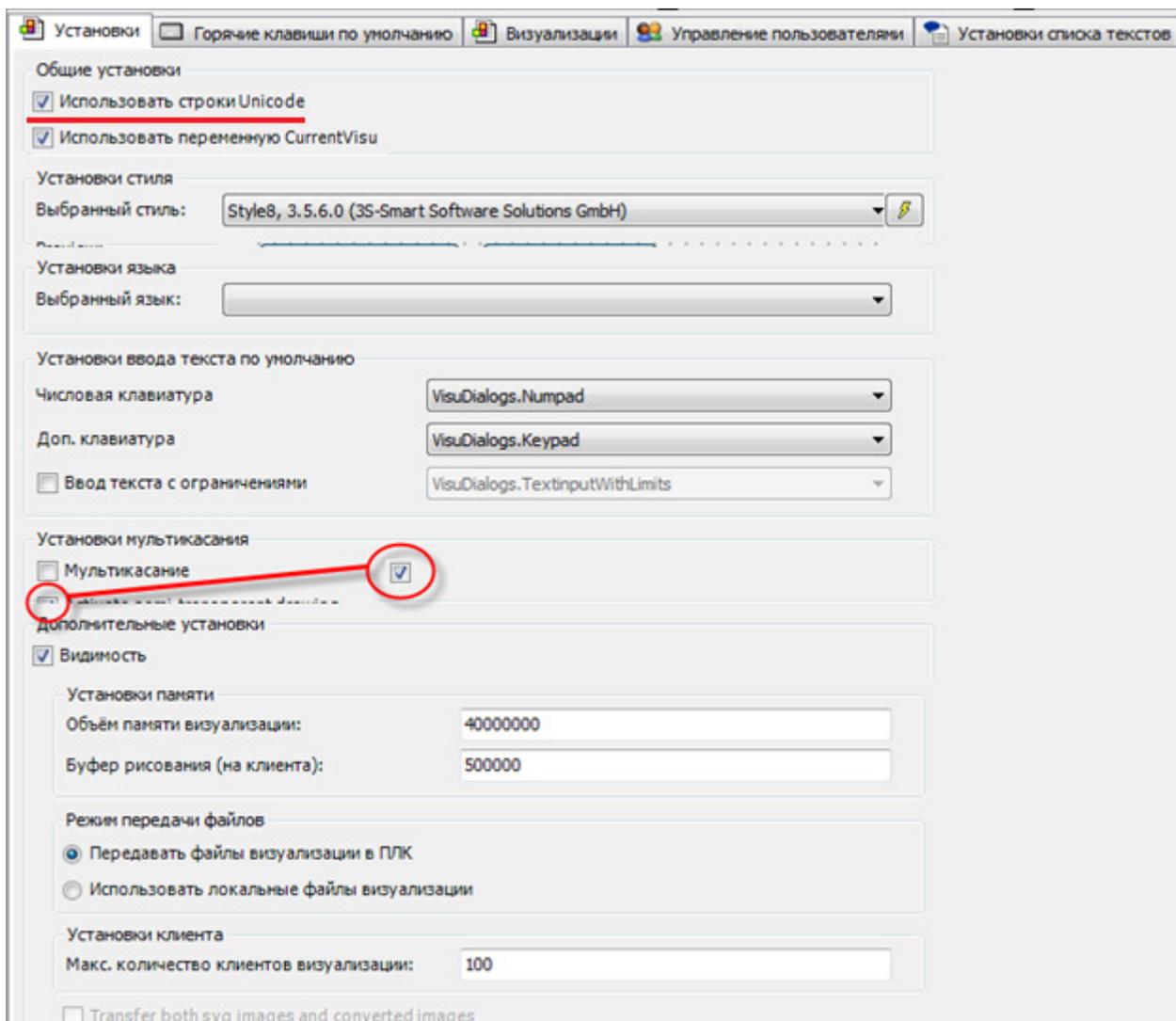
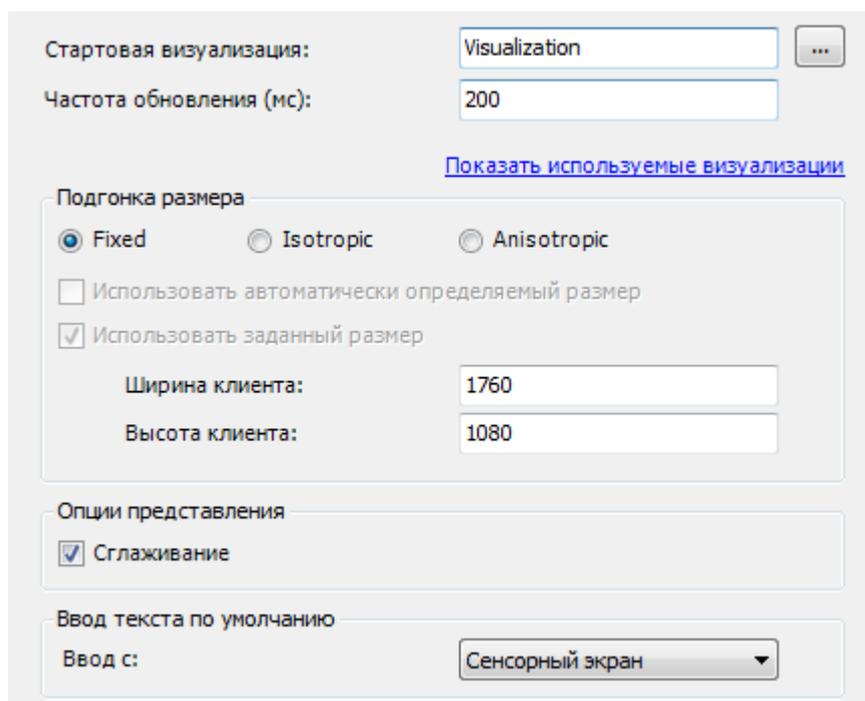


Рис. 11.2.1.9. Настройки Менеджера визуализации

Настройки **target**- и **web**-визуализации будут следующими:



Start visualization: Visualization

Refresh rate (ms): 200

[Show used visualizations](#)

Size adjustment

Fixed Isotropic Anisotropic

Use automatically determined size

Use specified size

Client width: 1760

Client height: 1080

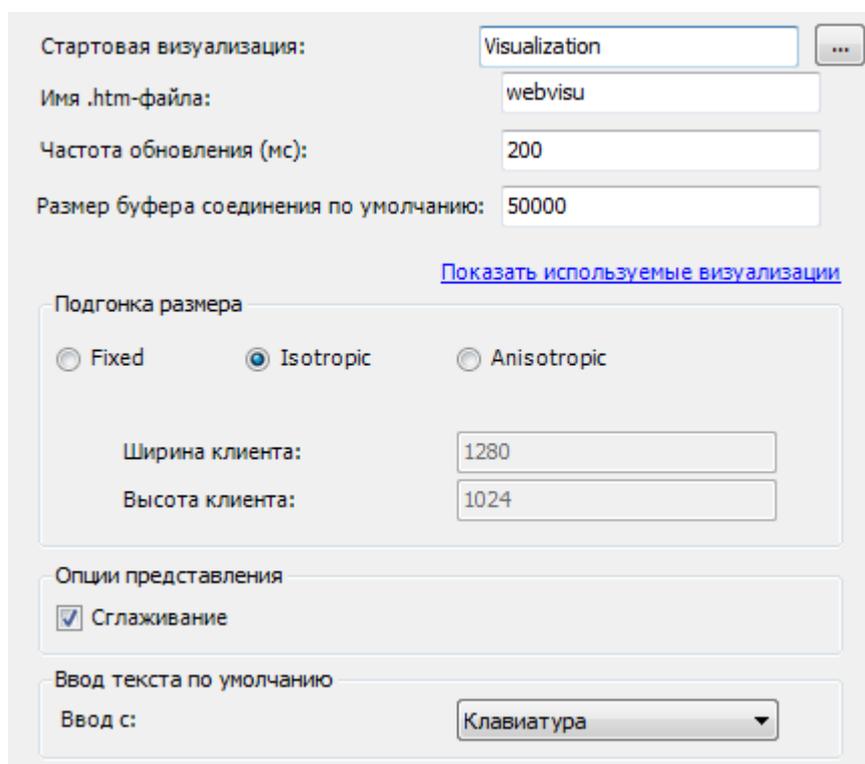
Display options

Smoothing

Text input by default

Input device: Сенсорный экран

Рис. 11.2.1.10. Настройки **target**-визуализации



Start visualization: Visualization

File name: webvisu

Refresh rate (ms): 200

Default connection buffer size: 50000

[Show used visualizations](#)

Size adjustment

Fixed Isotropic Anisotropic

Client width: 1280

Client height: 1024

Display options

Smoothing

Text input by default

Input device: Клавиатура

Рис. 11.2.1.11. Настройки **web**-визуализации

8. Добавим в проект [Список текстов](#) с названием **TextList**. Этот компонент потребуется для управления динамическими текстами прямоугольника:

ID	По умолчанию
0	Текст 1
1	Текст 2
2	Текст 3

Рис. 11.2.1.12. Содержимое списка текстов **TextList**

9. Менеджер библиотек нашего проекта должен выглядеть следующим образом:

Имя	Дополнительное имя	Действующая версия
Standard = Standard, 3.5.5.0 (System)	Standard	3.5.5.0
System_VisuElemMeter = VisuElemMeter, 3.5.6.0 (System)	VisuElemMeter	3.5.6.0
System_VisuElemsSpecialControls = VisuElemsSpecialControls, 3.5.6.0 (System)	VisuElemsSpecialControls	3.5.6.0
System_VisuElemsWinControls = VisuElemsWinControls, 3.5.6.0 (System)	VisuElemsWinControls	3.5.6.0
System_VisuElemTextEditor = VisuElemTextEditor, 3.5.6.0 (System)	VisuElemTextEditor	3.5.6.0
System_VisuElemTrace = VisuElemTrace, 3.5.6.0 (System)	VisuElemTrace	3.5.6.0
System_VisuNativeControl = VisuNativeControl, 3.5.6.0 (System)	VisuNativeControl	3.5.6.0
System_VisuElemsAlarm = VisuElemsAlarm, 3.5.6.0 (System)	VisuElemsAlarm	3.5.6.0
System_VisuElemCamDisplayer = VisuElemCamDisplayer, 3.5.6.0 (System)	VisuElemCamDisplayer	3.5.6.0
System_VisuElem3DPath = VisuElem3DPath, 3.5.6.0 (System)	VisuElem3DPath	3.5.6.0
System_VisuElemsDateTime = VisuElemsDateTime, 3.5.6.0 (System)	VisuElemsDateTime	3.5.6.0
system_visuinputs = visuinputs, 3.5.6.0 (system)	visuinputs	3.5.6.0
VisuDialogs = VisuDialogs, 3.5.6.0 (System)	VisuDialogs	3.5.6.0
Breakpoint.Logging = Breakpoint Logging Functions, 3.5.5.0 (3S - Smart Software Solutions GmbH)	BPLog	3.5.5.0
VisuElems, 3.5.6.20 (System)	VisuElems	3.5.6.20
IoStandard = IoStandard, 3.5.6.0 (System)	IoStandard	3.5.6.0
3SLicense = 3SLicense, 3.5.6.0 (3S - Smart Software Solutions GmbH)	_3S_LICENSE	3.5.6.0

Рис. 11.2.1.13. Список библиотек, используемых в проекте

10. Приступим к наполнению экрана **Visualization**. Внешний вид готового экрана был показан на рис. 11.2.1.1. Ниже будут описаны настройки содержащихся на нем элементов.

11. Добавим на экран элемент **Прямоугольник** и привяжем к нему переменные, объявленные в программе **PLC_PRG** (не отображенные параметры находятся в значениях по умолчанию):

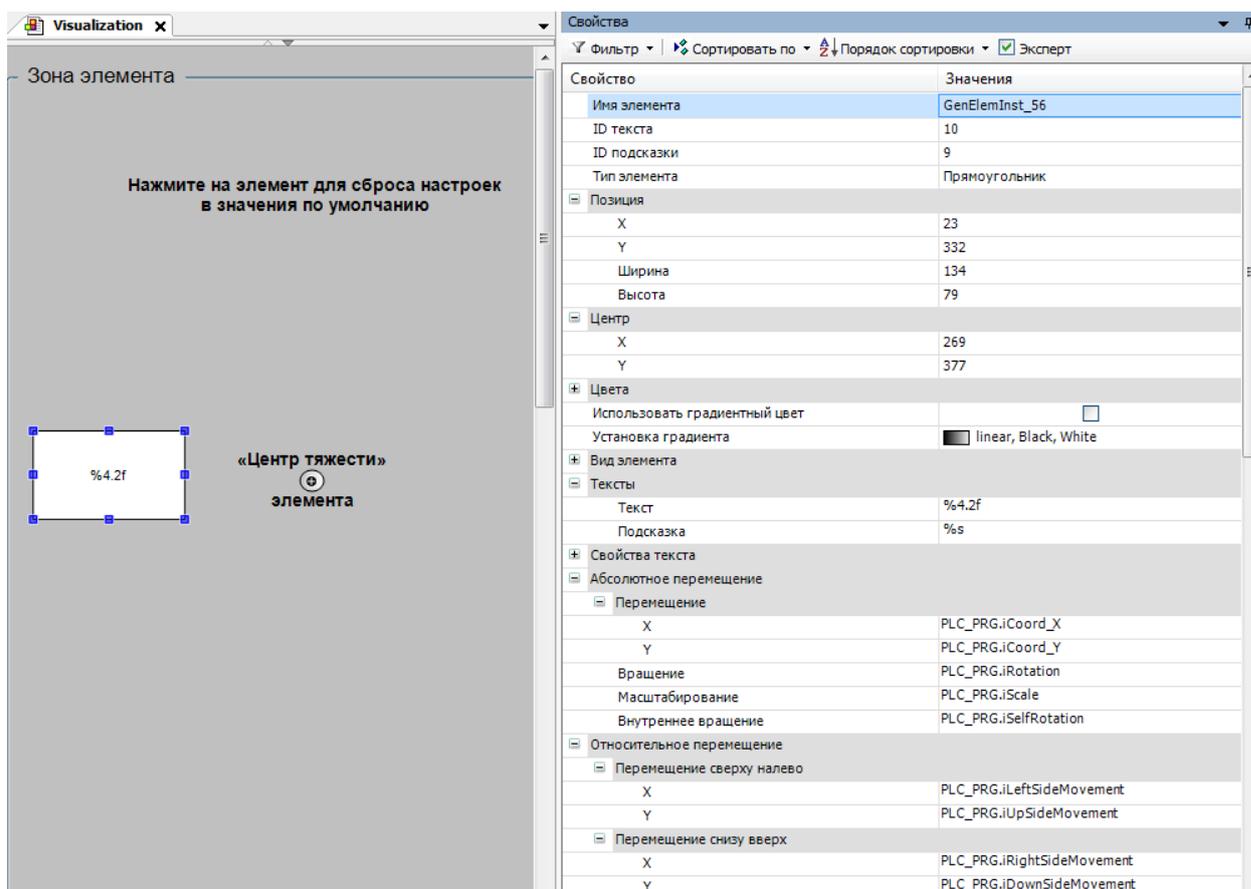


Рис. 11.2.1.14. Настройки элемента **Прямоугольник** (начало)

Свойства	
Фильтр	Сортировать по
	Порядок сортировки
	Эксперт
Свойство	Значения
Относительное пере...	
Текстовые переменные	
Текстовая перемен...	PLC_PRG.rElementVariable
Переменная подс...	PLC_PRG.wsTooltip
Динамические тексты	
Список текстов	'TextList'
Индекс текста	PLC_PRG.sTextIndex
Индекс подсказки	PLC_PRG.sTooltipIndex
Переменные шрифта	
Имя шрифта	PLC_PRG.sFontName
Размер	PLC_PRG.iTextSize
Флаги	PLC_PRG.dwTextFlag
Набор символов	
Цвет	PLC_PRG.dwTextColor
Flags for text alig...	PLC_PRG.iTextAligment
Переменные цвета	
Переключить цвет	PLC_PRG.bSwitchStateColor
Нормальное сост...	
Цвет фрейма	PLC_PRG.dwNormalState_FrameColor
Цвет заливки	PLC_PRG.dwNormalState_FillColor
Состояние тревоги	
Цвет фрейма	PLC_PRG.dwAlarmState_FrameColor
Цвет заливки	PLC_PRG.dwAlarmlState_FillColor
Переменные вида	
Ширина линии	PLC_PRG.iLineThickness
Атрибуты заливки	PLC_PRG.dwFillEnable
Стиль линии	PLC_PRG.dwLineStyle
Переменные состояний	
Невидимый	PLC_PRG.bInvisible
Отключение ввода	PLC_PRG.bNoTouchReaction
Inputconfiguration	
OnDialogClosed	Конфигурация...
OnMouseClicked	Конфигурация...
Выполни ST...	⚡ PLC_PRG.iCoord_X:=0;

Рис. 11.2.1.15. Настройки элемента **Прямоугольник** (окончание)

На центр элемента наложим элемент **Эллипс** (чтобы можно было видеть, вокруг чего вращается элемент). Во вкладке **Абсолютное перемещение/Перемещение** привяжем те же переменные, что и у прямоугольника (чтобы при перемещении элемента, центр его тяжести также смещался).

Абсолютное перемещение	
Перемещение	
X	PLC_PRG.iCoord_X
Y	PLC_PRG.iCoord_Y

Текстовые надписи создадим с помощью элементов [Метка](#). Границы области элемента (Зона элемента) представляет собой элемент [Группа](#).

Во вкладке [InputConfiguration](#) к параметру **OnClick** привяжем действие [Выполнить ST-код](#). ST-код содержит начальные значения переменных элемента. По нажатию на элемент, его параметры будут возвращены к этим значениям.

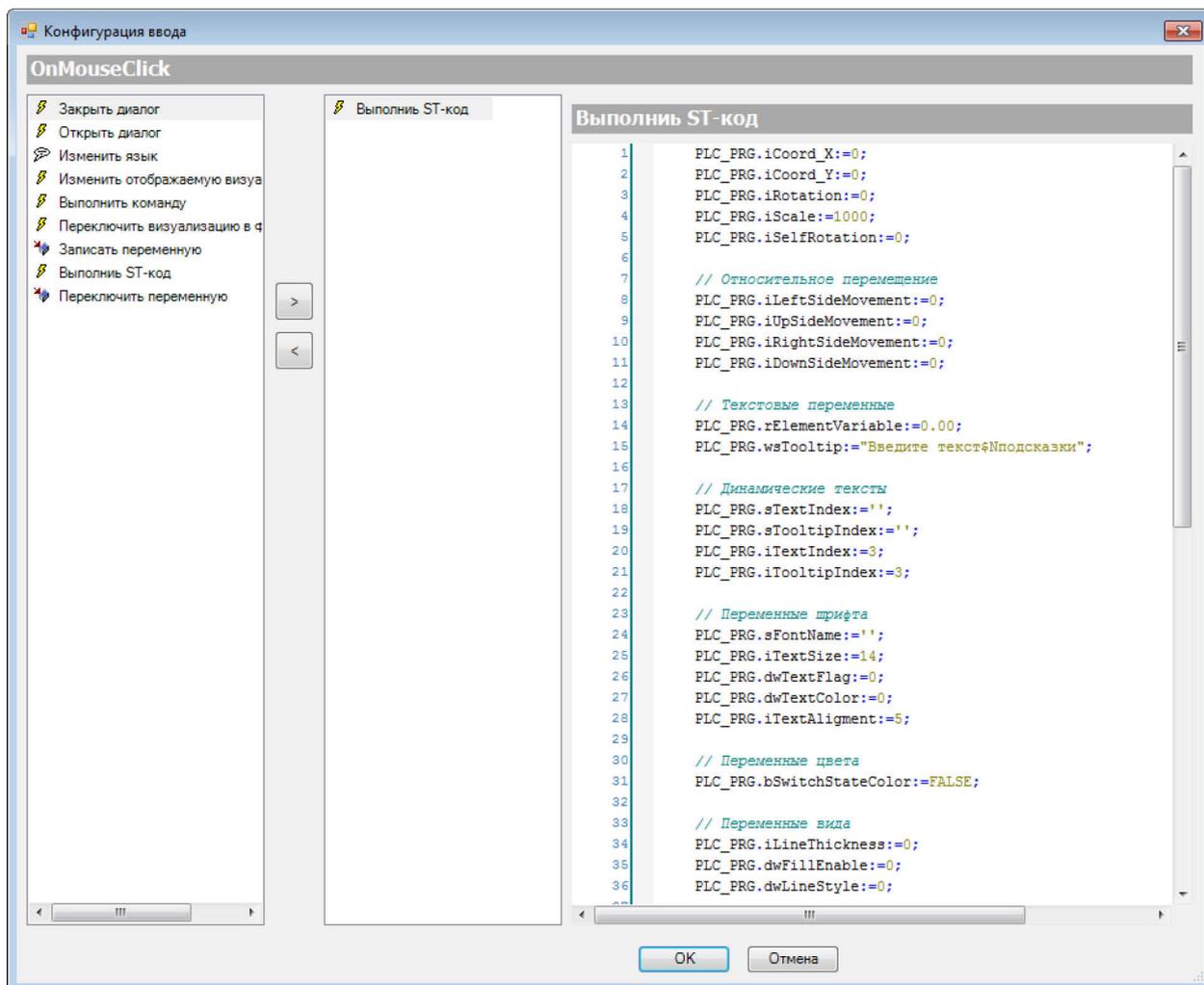


Рис. 11.2.1.16. Настройки параметра **OnClick** вкладки **InputConfiguration** элемента **Прямоугольник** (начало)

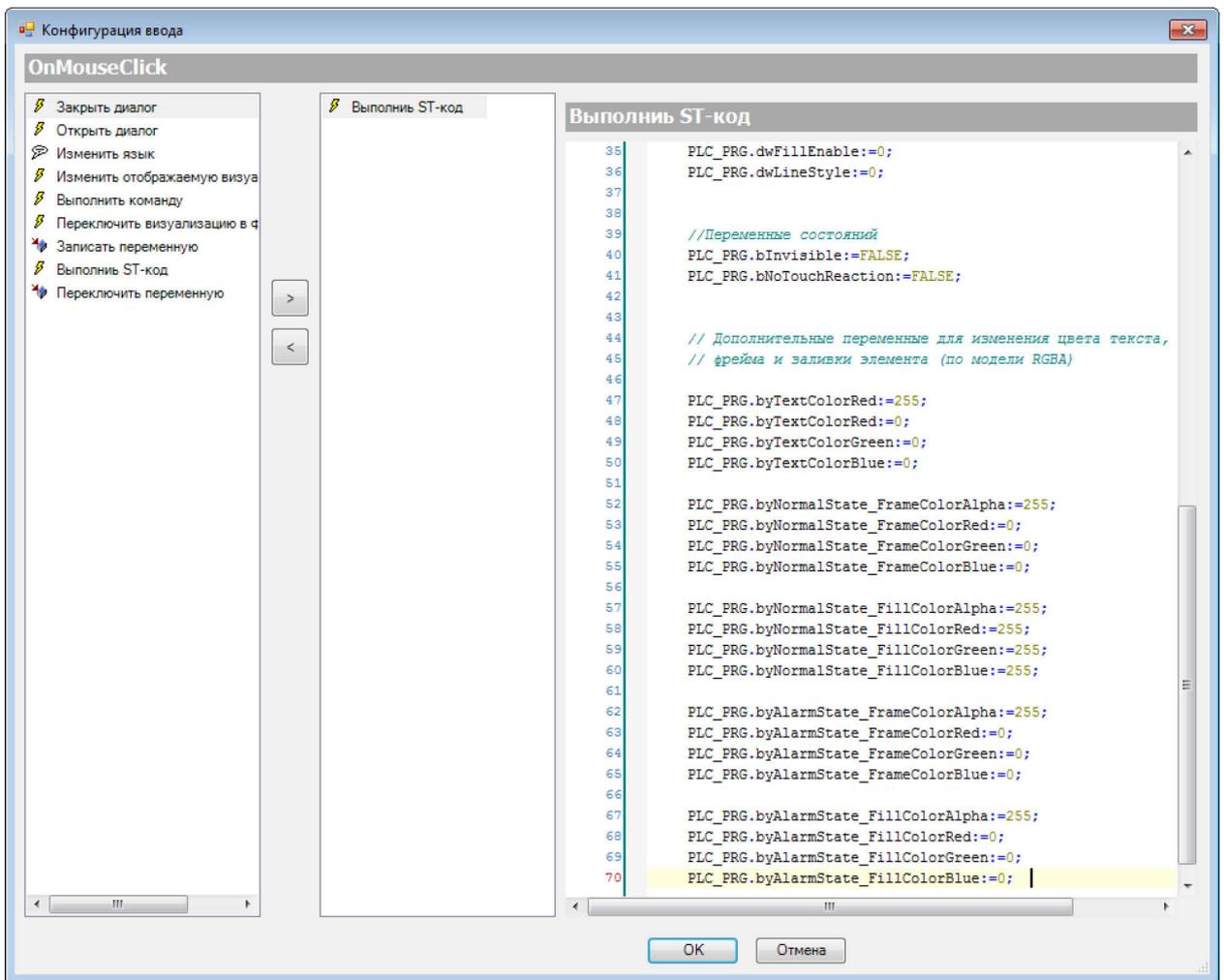


Рис. 11.2.1.17. Настройки параметра **OnMouseClicked** вкладки **InputConfiguration** элемента **Прямоугольник** (окончание)

12. Добавим на экран панель управления параметрами вкладки **Абсолютное перемещение**. Эти параметры будут использоваться для перемещения и вращения элемента, а также изменения его масштаба.

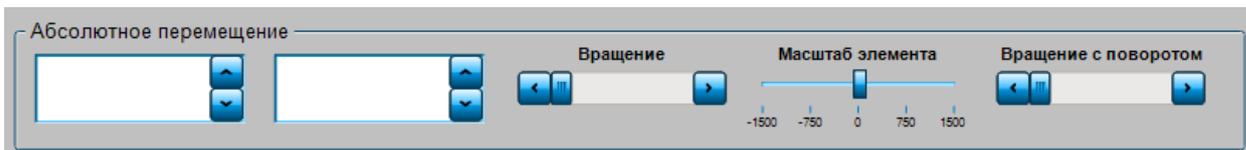


Рис. 11.2.1.18. Панель управления параметрами вкладки **Абсолютное перемещение**

Панель представляет собой элемент [Группа](#), в которой расположена два элемента [Управление вращением](#), два элемента [Полоса прокрутки](#) и [Бегунок](#). Три текстовых надписи сделаны с помощью элемента [Метка](#). Параметры элементов приведены ниже (не отображенные параметры находятся в значениях по умолчанию).

Свойства	
Свойство	Значения
Имя элемента	GenElemInst_66
Тип элемента	Управление вращением
Позиция	
X	20
Y	40
Ширина	190
Высота	60
Переменная	PLC_PRG.iCoord_X
Числовой формат	Координата X: %d
Интервал	1

Рис. 11.2.1.19. Настройки элемента **Управление вращением 1**

Свойства	
Свойство	Значения
Имя элемента	GenElemInst_157
Тип элемента	Управление вращением
Позиция	
X	240
Y	40
Ширина	190
Высота	60
Переменная	PLC_PRG.iCoord_Y
Числовой формат	Координата Y: %d
Интервал	1

Рис. 11.2.1.20. Настройки элемента **Управление вращением 2**

Свойства	
Фильтр Сортировать по Порядок сортировки Эксперт	
Свойство	Значения
Имя элемента	GenElemInst_70
Тип элемента	Полоса прокрутки
ID подсказки	60
Значение	PLC_PRG.iRotation
Минимальное значение	0
Максимальное значение	360
Размер страницы	10
Прокрутка выполнена	<input type="checkbox"/>
[-] Позиция	
X	463
Y	55
Ширина	190
Высота	29
[+] Линейка	
[+] Цвета	
[-] Тексты	
Текст	
Подсказка	Вращение вокруг центральной опорной точки э...
[+] Свойства текста	
[-] Текстовые переменные	
Текстовая переменная	
Переменная подсказки	PLC_PRG.iRotation
[+] Динамические тексты	
[+] Переменные шрифта	
[+] Переменные цвета	
[+] Переменные состояний	

Рис. 11.2.1.21. Настройки элемента **Полоса прокрутки 1**

Свойства	
Свойство	Значения
Имя элемента	GenElemInst_169
Тип элемента	Бегунок
[-] Позиция	
X	674
Y	44
Ширина	205
Высота	50
Переменная	PLC_PRG.iScale
Размер страницы	
Прокрутка выполнена	<input checked="" type="checkbox"/>
[-] Шкала	
Показать шкалу	<input checked="" type="checkbox"/>
Начало шкалы	-1500
Конец шкалы	1500
Основная шкала	750
Подшкала	0
Формат шкалы (с-синт...	%d
[+] Линейка	
[+] Переменные состояний	

Рис. 11.2.1.22. Настройки элемента **Бегунок**

Свойства	
Свойство	Значения
Имя элемента	GenElemInst_74
Тип элемента	Полоса прокрутки
ID подсказки	62
Значение	PLC_PRG.iSelfRotation
Минимальное значение	0
Максимальное значение	360
Размер страницы	10
Прокрутка выполнена	<input type="checkbox"/>
[-] Позиция	
X	1468
Y	65
Ширина	190
Высота	29
[-] Линейка	
Ориентация	Горизонтально
Направление движения	Слева направо
[+] Цвета	
[-] Тексты	
Текст	
Подсказка	Вращение вокруг центральной опорной точки элемента с...
[+] Свойства текста	
[-] Текстовые переменные	
Текстовая переменная	PLC_PRG.iSelfRotation
Переменная подсказки	
[+] Динамические тексты	
[+] Переменные шрифта	
[+] Переменные цвета	
[+] Переменные состояний	

Вращение вокруг центральной опорной точки элемента с одновременным вращением вокруг своей оси на %d°

Рис. 11.2.1.23. Настройки элемента **Полоса прокрутки 2**

13. Добавим на экран панель управления параметрами вкладки **Относительное перемещение**. Эти параметры будут использоваться для изменения размеров элемента.



Рис. 11.2.1.24. Панель управления параметрами вкладки **Относительное перемещение**

Панель представляет собой элемент [Группа](#), в которой расположены четыре элемента [Управление вращением](#). Параметры элементов приведены ниже (не отображенные параметры находятся в значениях по умолчанию).

Свойства	
Фильтр Сортировать по Порядок сортировки Эксперт	
Свойство	Значения
Имя элемента	GenElemInst_84
Тип элемента	Управление вращением
Позиция	
X	20
Y	40
Ширина	225
Высота	60
Переменная	PLC_PRG.iLeftSideMovement
Числовой формат	Левая сторона: %d пикселей
Интервал	1

Рис. 11.2.1.25. Настройки элемента **Управление вращением 1**

Свойства	
Фильтр Сортировать по Порядок сортировки Эксперт	
Свойство	Значения
Имя элемента	GenElemInst_85
Тип элемента	Управление вращением
Позиция	
X	280
Y	40
Ширина	225
Высота	60
Переменная	PLC_PRG.iUpSideMovement
Числовой формат	Верхняя сторона: %d пикселей
Интервал	1

Рис. 11.2.1.26. Настройки элемента **Управление вращением 2**

Свойства	
Фильтр Сортировать по Порядок сортировки Эксперт	
Свойство	Значения
Имя элемента	GenElemInst_102
Тип элемента	Управление вращением
Позиция	
X	535
Y	40
Ширина	225
Высота	60
Переменная	PLC_PRG.iRightSideMovement
Числовой формат	Правая сторона: %d пикселей
Интервал	1

Рис. 11.2.1.27. Настройки элемента **Управление вращением 3**

Свойства	
Фильтр Сортировать по Порядок сортировки Эксперт	
Свойство	Значения
Имя элемента	GenElemInst_316
Тип элемента	Управление вращением
Позиция	
X	790
Y	40
Ширина	225
Высота	60
Переменная	PLC_PRG.iDownSideMovement
Числовой формат	Нижняя сторона: %d пикселей
Интервал	1

Рис. 11.2.1.28. Настройки элемента **Управление вращением 4**

14. Добавим на экран панель управления параметрами вкладки **Текстовые переменные**. Эти параметры будут отображаться элементом (**Текстовая переменная** - число с плавающей точкой, **Переменная подсказки** – строковая переменная).

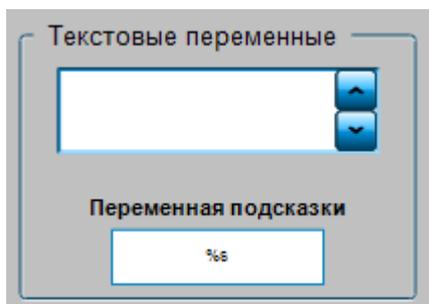


Рис. 11.2.1.29. Панель управления параметрами вкладки **Текстовые переменные**

Панель представляет собой элемент [Группа](#), в которой расположены элемент [Управление вращением](#) и элемент [Текстовое поле](#). Пояснительный текст (Переменная подсказки) создан с помощью элементов [Метка](#). Параметры элементов приведены ниже (не отображенные параметры находятся в значениях по умолчанию).

Свойства	
Свойство	Значения
Имя элемента	GenElemInst_190
Тип элемента	Управление вращением
Позиция	
X	28
Y	42
Ширина	225
Высота	58
Переменная	PLC_PRG.rElementVariable
Числовой формат	Текстовая переменная: %4.2f
Интервал	0.1

Рис. 11.2.1.30. Настройки элемента **Управление вращением**

Свойства	
Фильтр	Сортировать по
Порядок сортировки	Эксперт
Свойство	Значения
Имя элемента	GenElemInst_109
Тип элемента	Текстовое поле
ID текста	9
Позиция	
X	65
Y	156
Ширина	150
Высота	39
Цвета	
Вид элемента	
Тип тени	Из стиля
Тексты	
Текст	%s
Подсказка	
Свойства текста	
Текстовые переменные	
Текстовая переменная	PLC_PRG.wsTooltip
Переменная подсказки	
Динамические тексты	
Переменные шрифта	
Переменные цвета	
Переменные состояний	
Inputconfiguration	
OnDialogClosed	Конфигурация...
OnMouseClicked	Конфигурация...
Записать переменную	Variable : , InputType : Edit, Min : , Max :

Рис. 11.2.1.31. Настройки элемента **Текстовое поле**

У элемента **Текстовое поле** во вкладке [InputConfiguration](#) к параметру **OnMouseClicked** привяжем действие [Записать переменную](#).

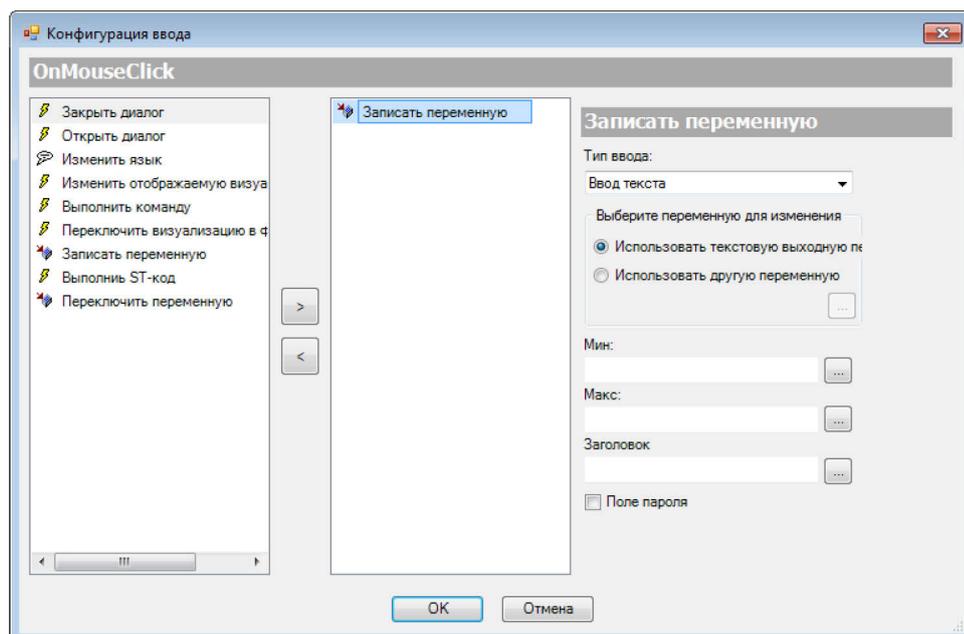


Рис. 11.2.1.32. Настройки действия элемента **Текстовое поле**

15. Добавим на экран панель управления параметрами вкладки **Динамические тексты**. Эти параметры позволяют выбирать тексты из [Списка текстов](#), которые будут отображаться элементом и его подсказкой (вместо переменных из вкладки **Текстовые переменные**).

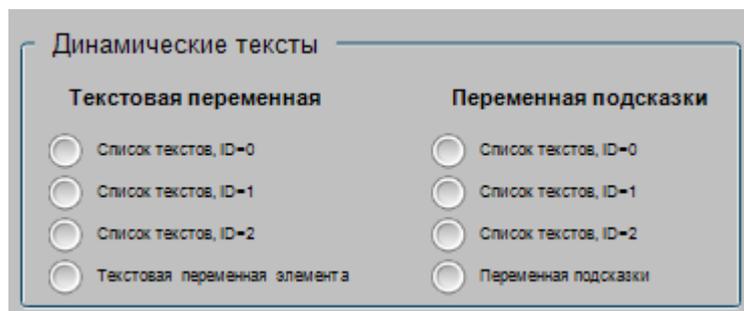


Рис. 11.2.1.33. Панель управления параметрами вкладки **Динамические тексты**

Панель представляет собой элемент [Группа](#), в которой расположены два элемента [Радио-кнопка](#). Две текстовые надписи созданы с помощью элементов [Метка](#). Параметры элементов приведены ниже (не отображенные параметры находятся в значениях по умолчанию).

К элементам привязаны переменные типа **INT**, в которые записывается **ID** из списка текстов, преобразованные из типа **STRING**. Преобразование происходит в двух первых строчках программы **PLC_PRG** (см. рис. 11.2.1.6).

Это возможно, потому что наши **ID** представляют собой числа (см. рис. 11.2.1.12).

Свойство	Значения
Имя элемента	GenElemInst_123
Тип элемента	Радио-кнопка
Позиция	
X	17
Y	80
Ширина	234
Высота	140
Переменная	PLC_PRG.iTextIndex
Число столбцов	1
Порядок радио-кнопки	Слева направо
Размер фрейма	From style
Высота строки	From style
Свойства текста	
Переменные состояний	
Параметры радио-кнопки	
Радио-кнопка	Создать новый
Области	
[0]	Удалить
Текст	Список текстов, ID=0
Подсказка	
Отступы строки в пик...	0
[1]	Удалить
Текст	Список текстов, ID=1
Подсказка	
Отступы строки в пик...	0
[2]	Удалить
Текст	Список текстов, ID=2
Подсказка	
Отступы строки в пик...	0
[3]	Удалить
Текст	Текстовая переменная элемента
Подсказка	
Отступы строки в пик...	0

Рис. 11.2.1.34. Настройки элемента **Радио-кнопка 1**

Свойства	
Фильтр Сортировать по Порядок сортировки Эксперт	
Свойство	Значения
Имя элемента	GenElemInst_129
Тип элемента	Радио-кнопка
Позиция	
X	289
Y	80
Ширина	200
Высота	140
Переменная	PLC_PRG.iTooltipIndex
Число столбцов	1
Порядок радио-кнопки	Слева направо
Размер фрейма	From style
Высота строки	From style
Свойства текста	
Переменные состояний	
Параметры радио-кнопки	
Радио-кнопка	Создать новый
Области	
[0]	Удалить
Текст	Список текстов, ID=0
Подсказка	
Отступы строки в пик...	0
[1]	Удалить
Текст	Список текстов, ID=1
Подсказка	
Отступы строки в пик...	0
[2]	Удалить
Текст	Список текстов, ID=2
Подсказка	
Отступы строки в пик...	0
[3]	Удалить
Текст	Переменная подсказки
Подсказка	
Отступы строки в пик...	0

Рис. 11.2.1.35. Настройки элемента **Радио-кнопка 2**

16. Добавим на экран панель управления параметрами вкладки **Переменные вида**. Эти параметры позволяют настраивать толщину контура элемента, выбирать режим его заливки (0 - включена/ >0 - отключена) и стиль.

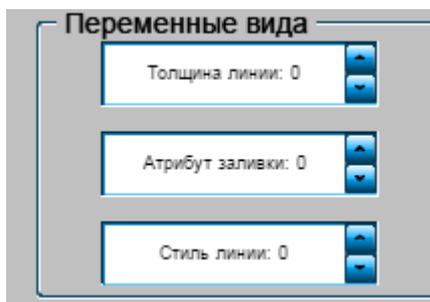


Рис. 11.2.1.36. Панель управления параметрами вкладки **Переменные вида**

Панель представляет собой элемент [Группа](#), в которой расположены три элемента [Управление вращением](#). Параметры элементов приведены ниже (не отображенные параметры находятся в значениях по умолчанию).

Свойства	
Свойство	Значения
Имя элемента	GenElemInst_100
Тип элемента	Управление вращением
[-] Позиция	
Х	20
У	43
Ширина	225
Высота	60
Переменная	PLC_PRG.iLineThickness
Числовой формат	Толщина линии: %d
Интервал	1
[-] Value range	
Минимальное значение	0
Максимальное значение	

Рис. 11.2.1.37. Настройки элемента **Управление вращением 1**

Свойства	
Фильтр Сортировать по Порядок сортировки Эксперт	
Свойство	Значения
Имя элемента	GenElemInst_104
Тип элемента	Управление вращением
Позиция	
X	20
Y	123
Ширина	225
Высота	60
Переменная	PLC_PRG.dwFillEnable
Числовой формат	Атрибут заливки: %d
Интервал	1
Value range	
Минимальное значение	0
Максимальное значение	1

Рис. 11.2.1.38. Настройки элемента **Управление вращением 2**

Свойства	
Фильтр Сортировать по Порядок сортировки Эксперт	
Свойство	Значения
Имя элемента	GenElemInst_359
Тип элемента	Управление вращением
Позиция	
X	43
Y	156
Ширина	180
Высота	40
Переменная	PLC_PRG.dwLineStyle
Числовой формат	Стиль линии: %d
Интервал	1
Value range	
Минимальное значение	0
Максимальное значение	8
Свойства текста	
Переменные цвета	
Переменные состояний	
Inputconfiguration	

Рис. 11.2.1.39. Настройки элемента **Управление вращением 3**

17. Добавим на экран панель управления параметрами вкладки **Переменные шрифта**. Эти параметры позволяют настраивать шрифт, размер стиль и цвет текста элемента. Как упоминалось ранее (п. 3), цвет будет собираться из значений задаваемых пользователем оттенков. Параметр Набор символов не работает в текущей (3.5 SP6) версии **CODESYS**, и поэтому в данном примере не рассматривается.

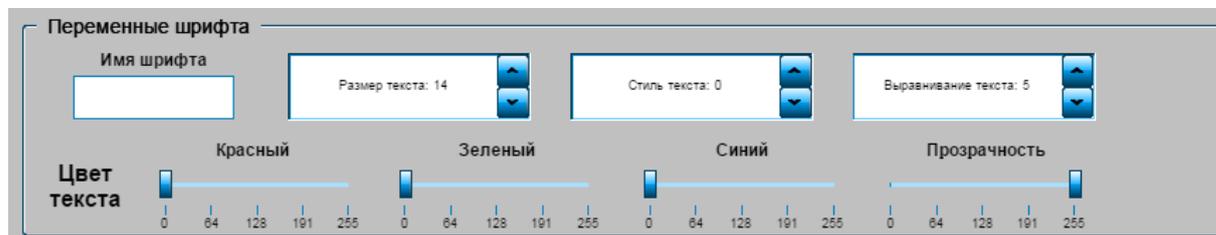


Рис. 11.2.1.40. Панель управления параметрами вкладки **Переменные шрифта**

Панель представляет собой элемент [Группа](#), в которой расположен элемент [Текстовое поле](#), три элемента [Управление вращением](#) и три элемента [Бегунок](#). Пять текстовых надписей сделаны с помощью элементов [Метка](#). Параметры элементов приведены ниже (не отображенные параметры находятся в значениях по умолчанию).

Свойство	Значения
Имя элемента	GenElemInst_153
Тип элемента	Текстовое поле
ID текста	9
Позиция	
X	48
Y	70
Ширина	150
Высота	40
Цвета	
Вид элемента	
Тип тени	Из стиля
Тексты	
Текст	%s
Подсказка	
Свойства текста	
Текстовые переменные	
Текстовая переменная	PLC_PRG.sFontName
Переменная подсказки	
Динамические тексты	
Переменные шрифта	
Переменные цвета	
Переменные состояний	
Inputconfiguration	
OnDialogClosed	Конфигурация...
OnMouseClicked	Конфигурация...
Записать переменную	Variable : , InputType : ..

Рис. 11.2.1.41. Настройки элемента **Текстовое поле**

У элемента [Текстовое поле](#) во вкладке [InputConfiguration](#) к параметру **OnMouseClicked** привяжем действие [Записать переменную](#).

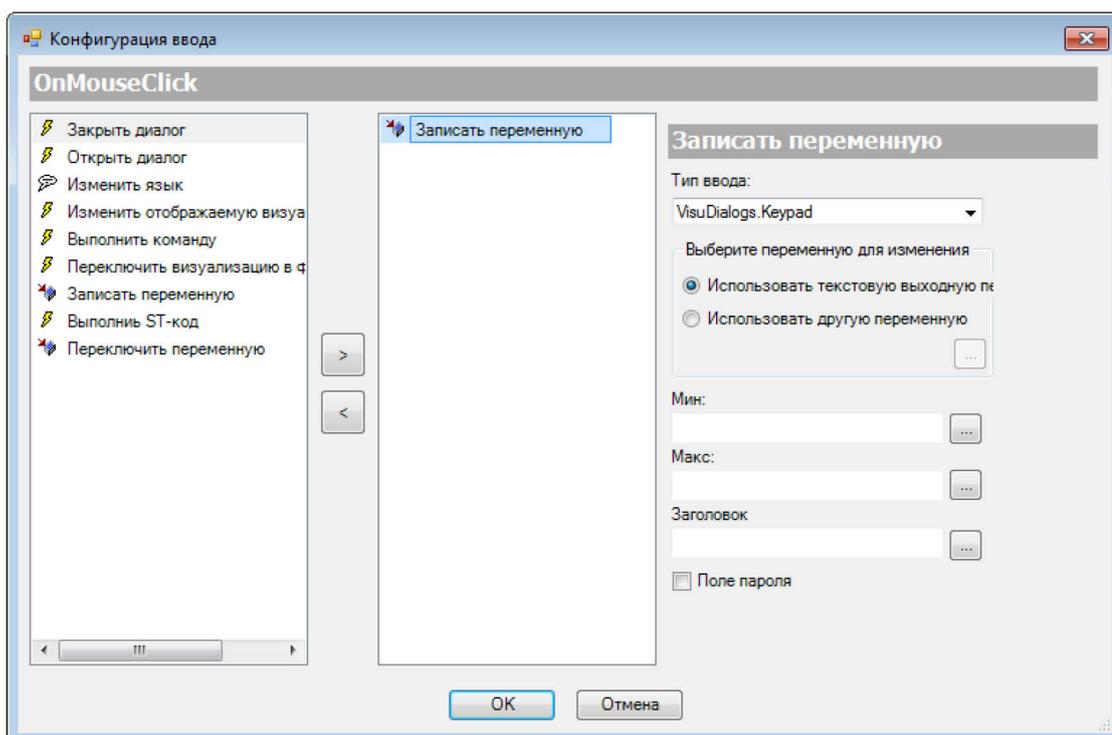


Рис. 11.2.1.42. Настройки действия элемента **Текстовое поле**

Свойства	
Свойство	Значения
Имя элемента	GenElemInst_202
Тип элемента	Управление вращением
Позиция	
X	250
Y	50
Ширина	225
Высота	60
Переменная	PLC_PRG.iTextSize
Числовой формат	Размер текста: %d
Интервал	1

Рис. 11.2.1.43. Настройки элемента **Управление вращением 1**

Свойства	
<input type="checkbox"/> Фильтр <input checked="" type="checkbox"/> Сортировать по <input type="checkbox"/> Порядок сортировки	
<input checked="" type="checkbox"/> Эксперт	
Свойство	Значения
Имя элемента	GenElemInst_200
Тип элемента	Управление вращением
<input checked="" type="checkbox"/> Позиция	
X	515
Y	50
Ширина	225
Высота	60
Переменная	PLC_PRG.dwTextFlag
Числовой формат	Стиль текста: %d
Интервал	1
<input checked="" type="checkbox"/> Value range	
Минимальное значение	0
Максимальное значение	15

Рис. 11.2.1.44. Настройки элемента **Управление вращением 2**

Свойства	
<input type="checkbox"/> Фильтр <input checked="" type="checkbox"/> Сортировать по <input type="checkbox"/> Порядок сортировки	
<input checked="" type="checkbox"/> Эксперт	
Свойство	Значения
Имя элемента	GenElemInst_318
Тип элемента	Управление вращением
<input checked="" type="checkbox"/> Позиция	
X	780
Y	50
Ширина	225
Высота	60
Переменная	PLC_PRG.iTextAligment
Числовой формат	Выравнивание текста: %d
Интервал	1
<input checked="" type="checkbox"/> Value range	
Минимальное значение	0
Максимальное значение	10

Рис. 11.2.1.45. Настройки элемента **Управление вращением 3**

Свойства	
̳ Фильтр ̳ Сортировать по ̳ Порядок сортировки	
<input checked="" type="checkbox"/> Эксперт	
Свойство	Значения
Имя элемента	GenElemInst_173
Тип элемента	Бегунок
[-] Позиция	
X	120
Y	150
Ширина	200
Высота	50
Переменная	PLC_PRG.byTextColorRed
Размер страницы	
Прокрутка выполнена	<input checked="" type="checkbox"/>
[-] Шкала	
Показать шкалу	<input checked="" type="checkbox"/>
Начало шкалы	0
Конец шкалы	255
Основная шкала	63
Подшкала	0
Формат шкалы (с-синтаксис)	%d

Рис. 11.2.1.46. Настройки элемента **Бегунок 1**

Свойства	
̳ Фильтр ̳ Сортировать по ̳ Порядок сортировки	
<input checked="" type="checkbox"/> Эксперт	
Свойство	Значения
Имя элемента	GenElemInst_175
Тип элемента	Бегунок
[-] Позиция	
X	345
Y	150
Ширина	200
Высота	50
Переменная	PLC_PRG.byTextColorGreen
Размер страницы	
Прокрутка выполнена	<input checked="" type="checkbox"/>
[-] Шкала	
Показать шкалу	<input checked="" type="checkbox"/>
Начало шкалы	0
Конец шкалы	255
Основная шкала	63
Подшкала	0
Формат шкалы (с-синтаксис)	%d

Рис. 11.2.1.47. Настройки элемента **Бегунок 2**

Свойства	
Фильтр Сортировать по Порядок сортировки	
<input checked="" type="checkbox"/> Эксперт	
Свойство	Значения
Имя элемента	GenElemInst_196
Тип элемента	Бегунок
<input checked="" type="checkbox"/> Позиция	
X	575
Y	150
Ширина	200
Высота	50
Переменная	PLC_PRG.byTextColorBlue
Размер страницы	
Прокрутка выполнена	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> Шкала	
Показать шкалу	<input checked="" type="checkbox"/>
Начало шкалы	0
Конец шкалы	255
Основная шкала	63
Подшкала	0
Формат шкалы (с-синтаксис)	%d

Рис. 11.2.1.48. Настройки элемента **Бегунок 3**

18. Добавим на экран панель управления параметрами вкладки **Переменные цвета**. Эти параметры позволяют настраивать цвета контура и заливки элемента в нормальном и аварийном состоянии. Как упоминалось ранее (пп. 3), цвет будет собираться из значений задаваемых пользователем оттенков.



Рис. 11.2.1.49. Панель управления параметрами вкладки **Переменные цвета**

Панель представляет собой элемент [Группа](#), в которой расположен элемент [Клавишный выключатель](#), элемент [Переключатель изображения](#) (зеленая лампа), элемент [Индикатор](#) (красная лампа) и 16 элементов [Бегунок](#). Десять текстовых надписей сделаны с помощью элементов [Метка](#). Параметры элементов приведены ниже (не отображенные параметры находятся в значениях по умолчанию).

Свойство	Значения
Имя элемента	GenElemInst_329
Тип элемента	Клавишный выключатель
ID подсказки	70
Позиция	
X	132
Y	162
Ширина	45
Высота	60
Переменная	PLC_PRG.bSwitchStateColor
Параметры изображения	
Поведение элемента	Переключатель изображения
Тексты	
Подсказка	Состояние элемента (Норма/Тревога)
Переменные состояний	
Невидимый	
Отключение ввода	
Фон	
Изображение	Gray

Рис. 11.2.1.50. Настройки элемента **Клавишный выключатель**

Элемент [Клавишный переключатель](#) будет определять состояние цвета элемента (**TRUE** – тревога, **FALSE** – норма). Чтобы визуализировать это состояние, необходимо два индикатора – зеленый (**FALSE**, норма) и красный (**TRUE**, тревога). Но элемент [Индикатор](#) в состоянии **FALSE** всегда отображается «потухшим». Поэтому вместо индикатора, для состояния «Норма» воспользуемся элементом [Переключатель изображения](#), в котором пропишем ссылки на картинки индикатора из [пула изображений](#) стиля визуализации:

Свойства	
Фильтр	Сортировать по
Порядок сортировки	Эксперт
Свойство	Значения
Имя элемента	GenElemInst_347
Тип элемента	Переключатель изображения
Позиция	
X	125
Y	75
Ширина	70
Высота	60
Переменная	PLC_PRG.bSwitchStateColor
Параметры изображения	
Изображение вкл.	__VisualizationStyle.Element-Lamp-Lamp1-Green-Off
Изображение выкл.	__VisualizationStyle.Element-Lamp-Lamp1-Green-On
Image pushed	
Прозрачный	<input type="checkbox"/>
Прозрачный цвет	Black
Изотропный тип	Изотропия
Горизонтальное выра...	Лево
Вертикальное вырав...	Верх
Поведение элемента	Переключатель изображения
Тексты	
Переменные состояний	

Рис. 11.2.1.51. Настройки элемента **Переключатель изображения** («Норма»)

Индикатор «Тревога» будет представлять собой стандартный элемент [Индикатор](#):

Свойства	
Фильтр	Сортировать по
Порядок сортировки	Эксперт
Свойство	Значения
Имя элемента	GenElemInst_294
Тип элемента	Индикатор
Позиция	
X	125
Y	244
Ширина	70
Высота	60
Переменная	PLC_PRG.bSwitchStateColor
Параметры изображения	
Тексты	
Переменные состояний	
Фон	
Изображение	Red

Рис. 11.2.1.52. Настройки элемента **Индикатор** («Тревога»)

Элементы [Бегунок](#) отличаются только привязываемыми к ним переменными (см. табл. 43):

Свойства	
Фильтр	Сортировать по
Порядок сортировки	Эксперт
Свойство	Значения
Имя элемента	GenElemInst_246
Тип элемента	Бегунок
Позиция	
X	820
Y	750
Ширина	205
Высота	50
Переменная	PLC_PRG.byNormalState_FrameColorRed
Размер страницы	
Прокрутка выполнена	<input checked="" type="checkbox"/>
Шкала	
Показать шкалу	<input checked="" type="checkbox"/>
Начало шкалы	0
Конец шкалы	255
Основная шкала	63
Подшкала	0
Формат шкалы (с-син...	%d
Линейка	
Переменные состояний	

Рис. 11.2.1.53. Настройки элемента **Бегунок**

Табл. 43. Переменные элементов **Бегунок** панели управление **Переменные цвета**

Цвет	Фрейм	Заливка
Состояние Норма		
Красный	PLC_PRG.byNormalState_FrameColorRed	PLC_PRG.byNormalState_FillColorRed
Зеленый	PLC_PRG.byNormalState_FrameColorGreen	PLC_PRG.byNormalState_FillColorGreen
Синий	PLC_PRG.byNormalState_FrameColorBlue	PLC_PRG.byNormalState_FillColorBlue
Прозрачность	PLC_PRG.byNormalState_FrameColorAlpha	PLC_PRG.byNormalState_FillColorAlpha
Состояние Тревога		
Красный	PLC_PRG.byAlarmState_FrameColorRed	PLC_PRG.byAlarmState_FillColorRed
Зеленый	PLC_PRG.byAlarmState_FrameColorGreen	PLC_PRG.byAlarmState_FillColorGreen
Синий	PLC_PRG.byAlarmState_FrameColorBlue	PLC_PRG.byAlarmState_FillColorBlue
Прозрачность	PLC_PRG.byAlarmState_FrameColorAlpha	PLC_PRG.byAlarmState_FillColorAlpha

19. Добавим на экран панель управления параметрами вкладки **Переменные состояний**. Эти параметры позволяют управлять видимостью и активностью (реакцией на касание) элемента.

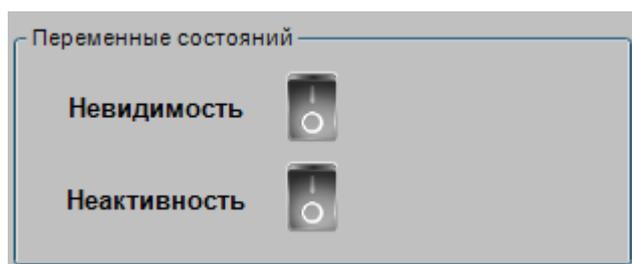


Рис. 11.2.1.54. Панель управления параметрами вкладки **Переменные состояний**

Панель представляет собой элемент [Группа](#), в которой расположены два элемента [Клавишный выключатель](#). Две текстовых надписи сделаны с помощью элементов [Метка](#). Параметры элементов приведены ниже (не отображенные параметры находятся в значениях по умолчанию).

Свойство	Значения
Имя элемента	GenElemInst_331
Тип элемента	Клавишный выключатель
Позиция	
X	225
Y	45
Ширина	45
Высота	60
Переменная	PLC_PRG.bInvisible
Параметры изображения	
Поведение элемента	Переключатель изображения
Тексты	
Переменные состояний	
Фон	
Изображение	Gray

Рис. 11.2.1.55. Настройки элемента **Клавишный выключатель 1**

Свойства	
Фильтр Сортировать по Порядок сортировки Эксперт	
Свойство	Значения
Имя элемента	GenElemInst_215
Тип элемента	Клавишный выключатель
ID подсказки	71
Позиция	
X	225
Y	120
Ширина	45
Высота	60
Переменная	PLC_PRG.bNoTouchReaction
Параметры изображения	
Поведение элемента	Переключатель изображения
Тексты	
Подсказка	Наличие у элемента реакции на нажатие
Переменные состояний	
Фон	
Изображение	Gray

Рис. 11.2.1.56. Клавишный выключатель 2

20. [Запустим проект на виртуальном контроллере](#). Откроем в веб-браузере страницу <http://localhost:8080/webvisu.htm>, чтобы увидеть web-визуализацию проекта. **Обратите внимание**, что браузер должен поддерживать **HTML5**. Проверим функционал проекта.

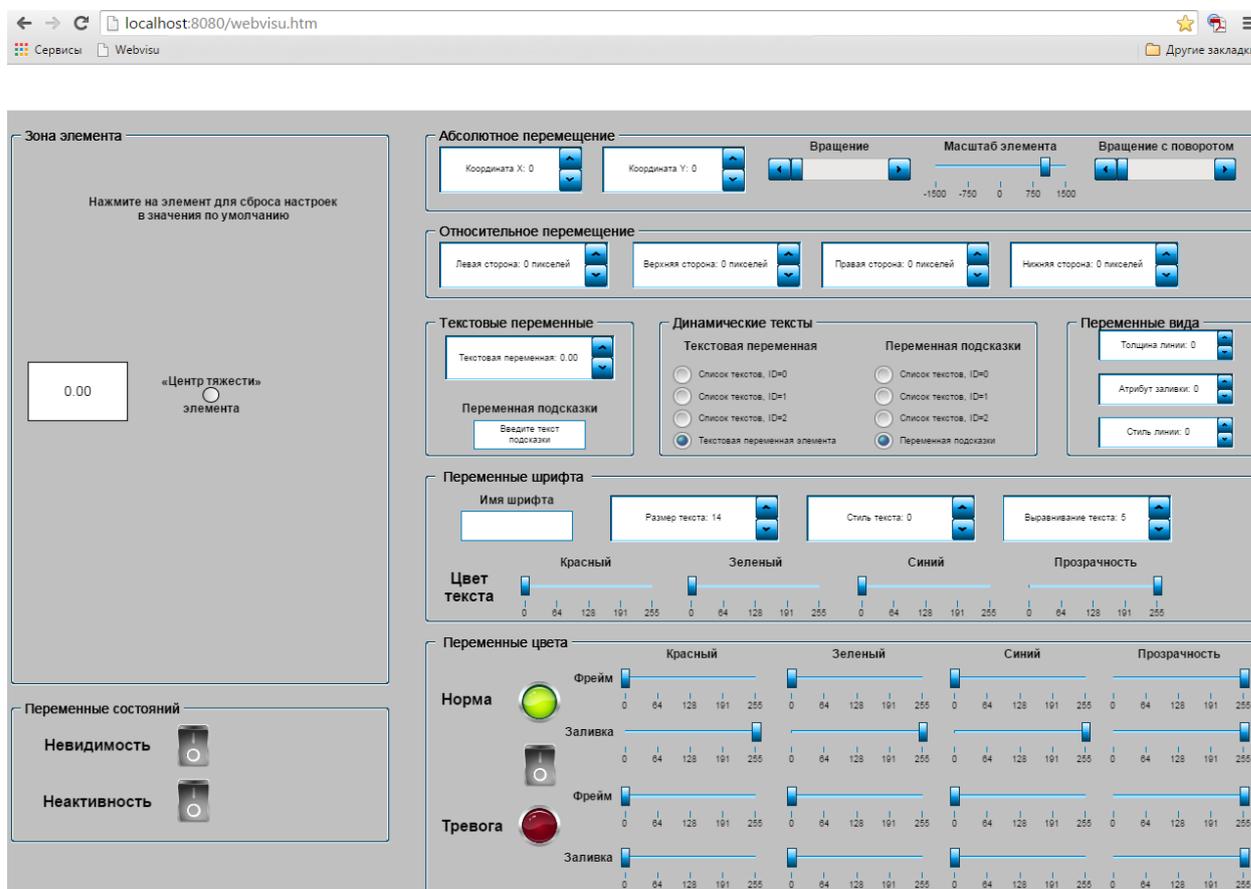


Рис. 11.2.1.57. Web-визуализация проекта в браузере **Chrome**

21. Проверим функционал вкладки **Абсолютное перемещение**:

- изменение координат X и Y приводит к соответствующему перемещению элемента и его центра тяжести;
- вращение элемента приводит к его вращению вокруг центра тяжести;
- изменение масштаба приводит к изменению размеров элемента; отрицательный масштаб соответствует зеркальному (по обеим осям) отображению элемента относительно центра тяжести;
- вращение с поворотом приводит к вращению элемента вокруг центра тяжести, при этом одновременно элемент поворачивается вокруг своего геометрического центра.

22. Проверим функционал вкладки Относительное перемещение:

- изменение значений приводит к соответствующему изменению размеров элемента.

23. Проверим функционал вкладки Текстовые переменные:

- изменение текстовой переменной приводит к изменению значения, отображаемого элементом;
- после ввода подсказки она будет отображена элементом.

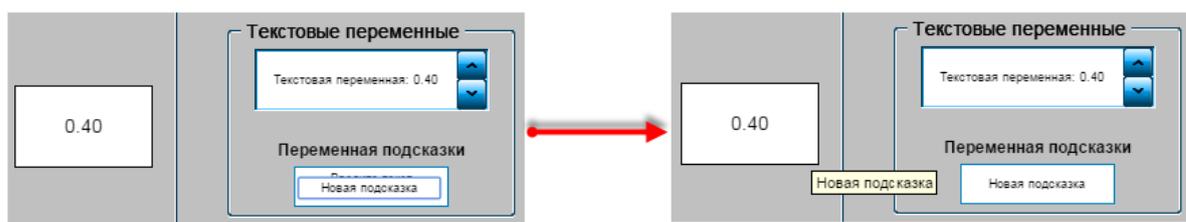


Рис. 11.2.1.58. Ввод и отображение подсказки элемента

24. Проверим функционал вкладки Динамические тексты:

- с помощью радиокнопок можно менять отображаемое значение и подсказку элемента на тексты из [Списка текстов](#).

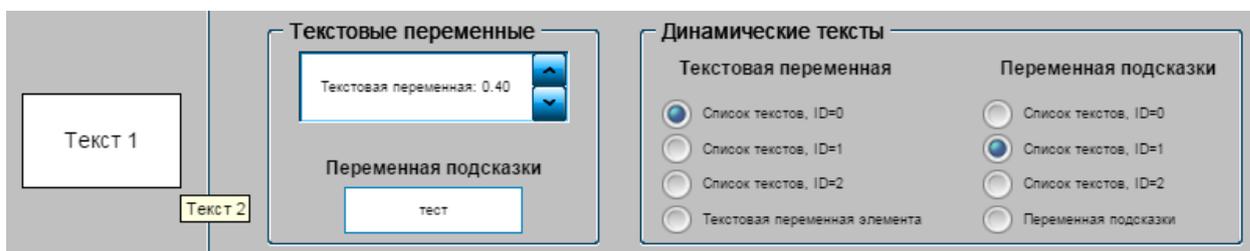


Рис. 11.2.1.59. Использование динамических текстов

25. Проверим функционал вкладки **Переменные вида:**

- изменение толщины линии приводит к изменению толщины контура элемента;
- изменение атрибута заливки с 0 на 1 отключает заливку элемента (делает его прозрачным);
- изменение стиля линии контура элементов web-визуализации работает только частично (значение 5 делает контур прозрачным):



Рис. 11.2.1.60. Стиль линий в web-визуализации



Рис. 11.2.1.61. Стиль линий в визуализации редактора **CODESYS**

26. Проверим функционал вкладки **Переменные шрифта**:

- меняя имя шрифта, можно выбирать шрифт текста элемента (шрифт подсказки это не затрагивает). **Обратите внимание**, что шрифт должен поддерживаться целевым устройством:

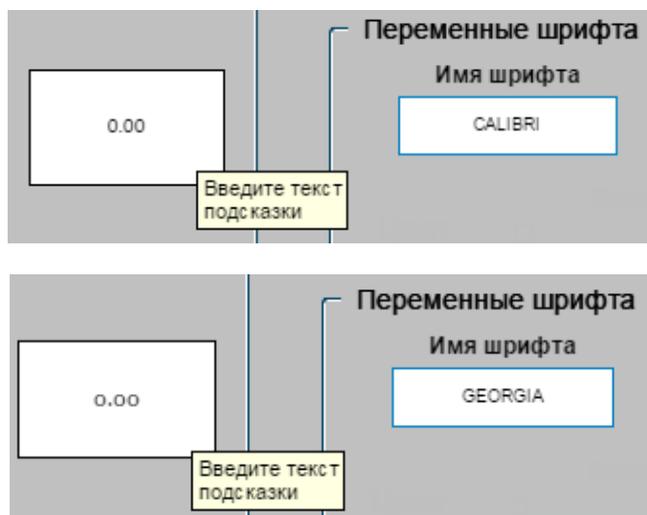


Рис. 11.2.1.62. Изменение шрифта текста элемента

- изменение размера текста работает корректно;
- изменение стиля текста в web-визуализации работает только частично – зачеркнутый и подчеркнутый стиль не поддерживаются:

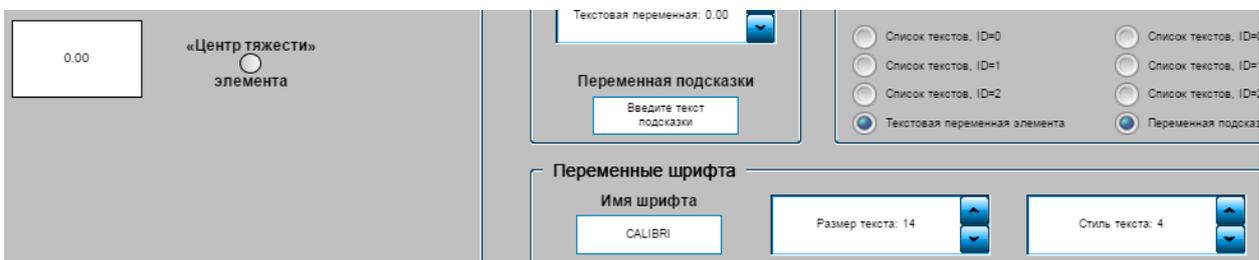


Рис. 11.2.1.63. Стиль текста в web-визуализации

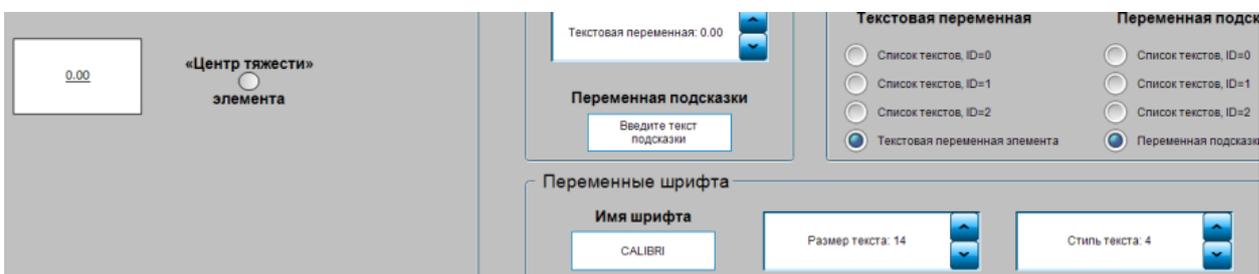


Рис. 11.2.1.64. Стиль текста в визуализации редактора **CODESYS**

- изменение выравнивания текста работает корректно;
- изменение цвета текста работает корректно.

27. Проверим функционал вкладки **Переменные цвета:**

- пользователь может задавать цвет контура и заливки элемента в обоих состояниях (норма/тревога) и переключать эти состояния.

28. Проверим функционал вкладки **Переменные состояний:**

- пользователь может управлять видимостью элемента;

- пользователь может управлять активностью элемента. Если элемент неактивен, то нажатие на прямоугольник не приведет к сбросу его настроек в значения по умолчанию.

11.2.2. Фрейм

Данный пример посвящен работе с графическим примитивом [Фрейм](#), который позволяет открывать в плоскости одних экранов визуализации другие.

Работа с [интерфейсом](#) фрейма рассмотрена в [п. 11.3.3](#).

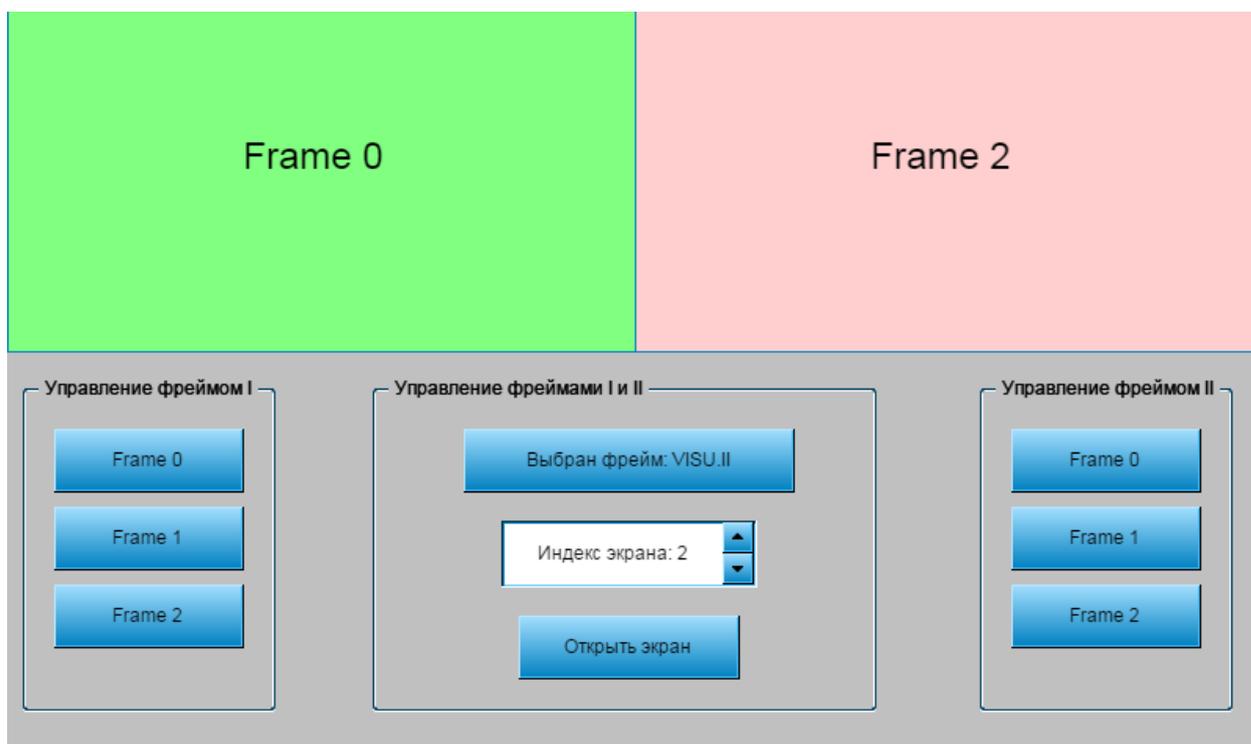


Рис. 11.2.2.1. Внешний вид примера **Фрейм**

Пример создан в среде **CODESYS 3.5 SP6 Patch 2** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с target-файлом **3.5.6.0**.

Пример доступен для скачивания: [Example_Frame.projectarchive](#)

1. Создадим новый *стандартный* проект **CODESYS** с названием **Example_Frame** и настройками по умолчанию: target – **CODESYS Control Win V3**, язык PLC_PRG – **ST**.

2. В программе **PLC_PRG** объявим следующие переменные:

```
1 PROGRAM PLC_PRG
2 VAR
3     iFrameIndex:INT; // индекс экрана, открытого во фрейме
4     sCurrentFrameName:STRING:='Visu.I'; // имя фрейма, в котором открывается экран
5 END VAR
```

Рис. 11.2.2.2. Объявление переменных программы **PLC_PRG**

3. Добавим в проект экран визуализации с названием **Frame_0**. В его свойствах выберем размер **400x220**. Этот экран будет одним из экранов, отображаемых элементом Фрейм. Добавим на него метку с названием экрана и выберем фон (с помощью вкладки Фон контекстного меню, открываемого по нажатию **ПКМ** на рабочую область).

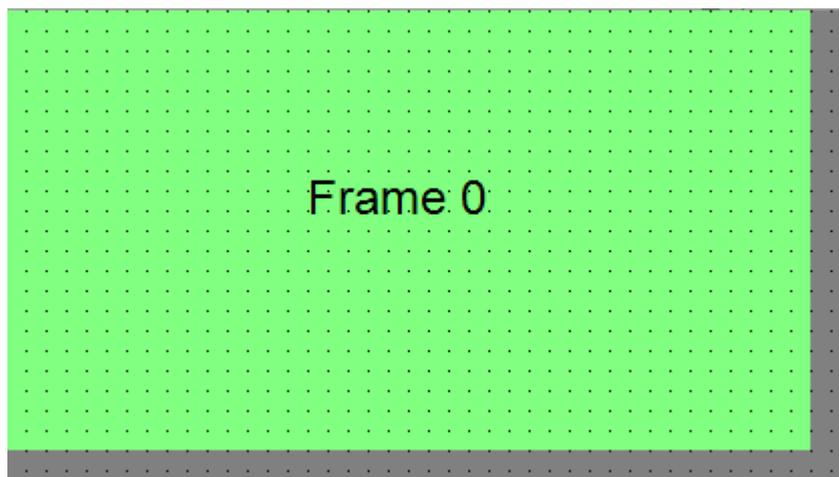


Рис. 11.2.2.3. Экран визуализации **Frame_0**

4. Добавим в проект экран визуализации с названием **Frame_1** и **Frame_2**. От экрана **Frame_0** они будут отличаться только текстом метки и фоном (**Frame_1** – синий, **Frame_2** – красный).

5. [Добавим в проект экран визуализации](#) с названием **Visu**. В его [свойствах](#) выберем размер **800x480**. Этот экран будет основным экраном проекта.

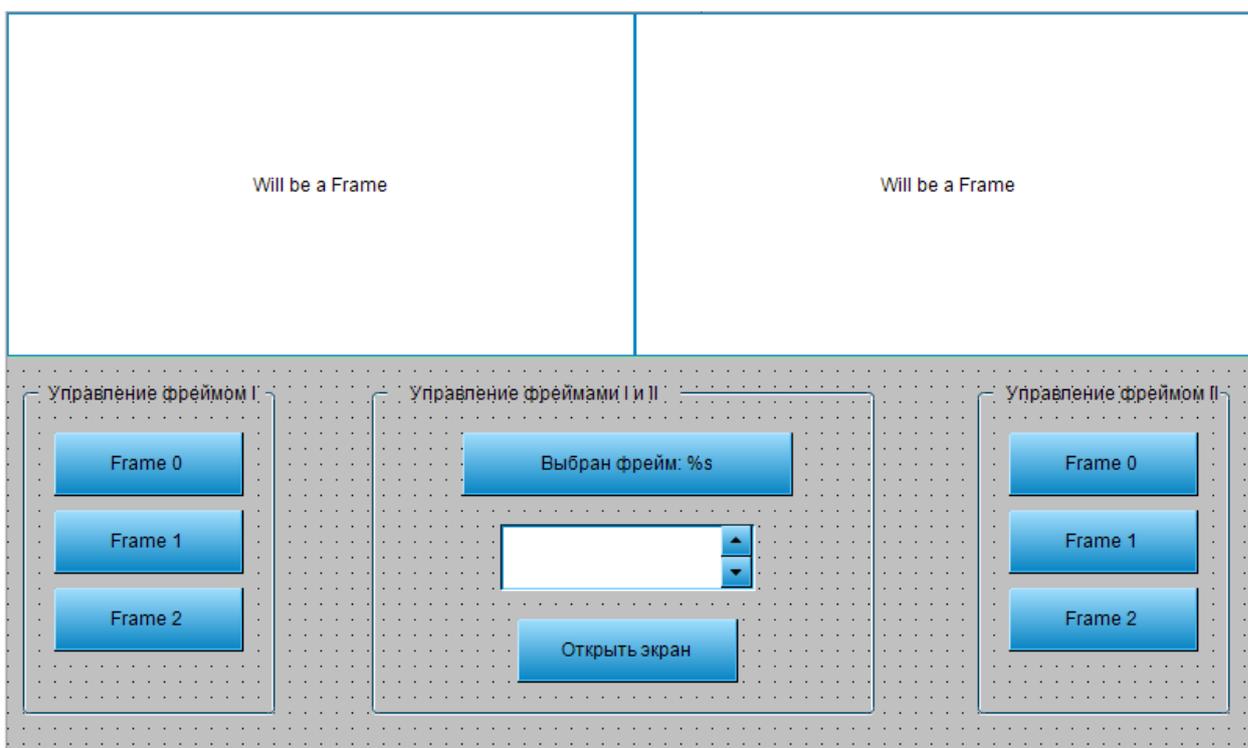


Рис. 11.2.2.4. Экран визуализации **Visu** (фреймы без конфигурации)

Экран содержит следующие элементы:

- два элемента [Фрейм](#) с пустой конфигурацией;
- вкладку [Управление фреймом I](#) (элемент [Группа](#)) с тремя элементами [Кнопка](#);
- вкладку [Управление фреймом II](#) (элемент [Группа](#)) с тремя элементами [Кнопка](#);
- вкладку [Управление фреймами I и II](#) (элемент [Группа](#)) с двумя элементами [Кнопка](#) и одним элементом [Управление вращением](#).

Настройки элементов приведены ниже.

6. Настроим первый элемент **Фрейм**. При добавлении его на экран визуализации, появляется меню с выбором экранов, которые будут отображаться данным фреймом. Чтобы опять открыть это меню, в параметрах фрейма нажмите кнопку конфигурация:

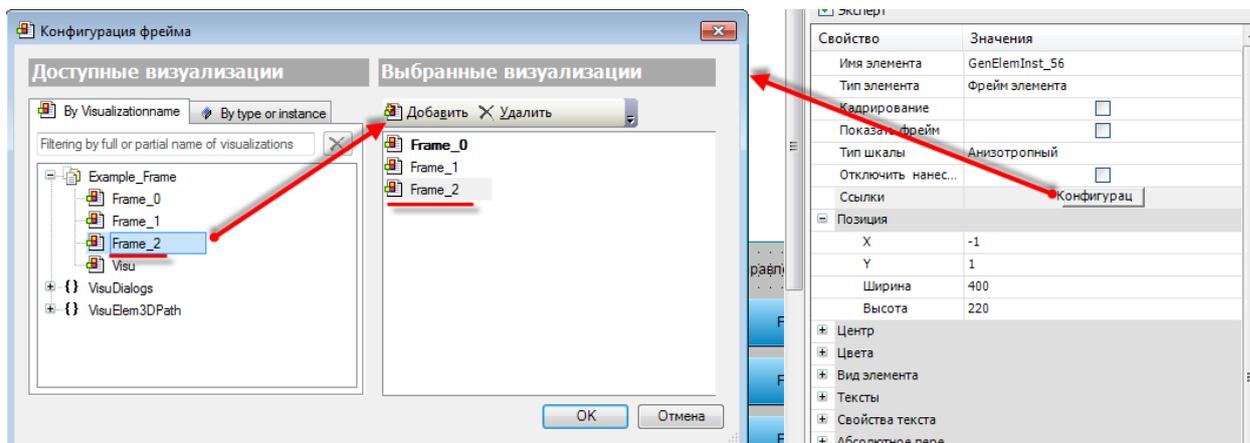


Рис. 11.2.2.5. Конфигурация фрейма

Привяжем к фрейму экраны **Frame_0**, **Frame_1**, **Frame_2**. Экран **Frame_0** в конфигурации выделен жирным шрифтом, т.к. был добавлен первым и стал для фрейма экраном по умолчанию.

Настроим фрейм следующим образом:

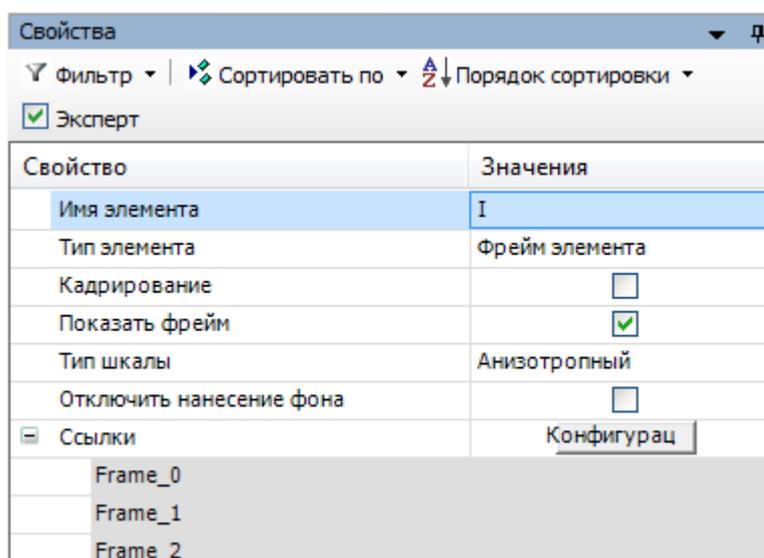


Рис. 11.2.2.6. Параметры элемента **Фрейм**

Конфигурация и настройки второго элемента фрейм будут отличаться только именем элемента – задайте ему имя **II**.

После настройки фреймов, экран **Visu** будет выглядеть следующим образом:

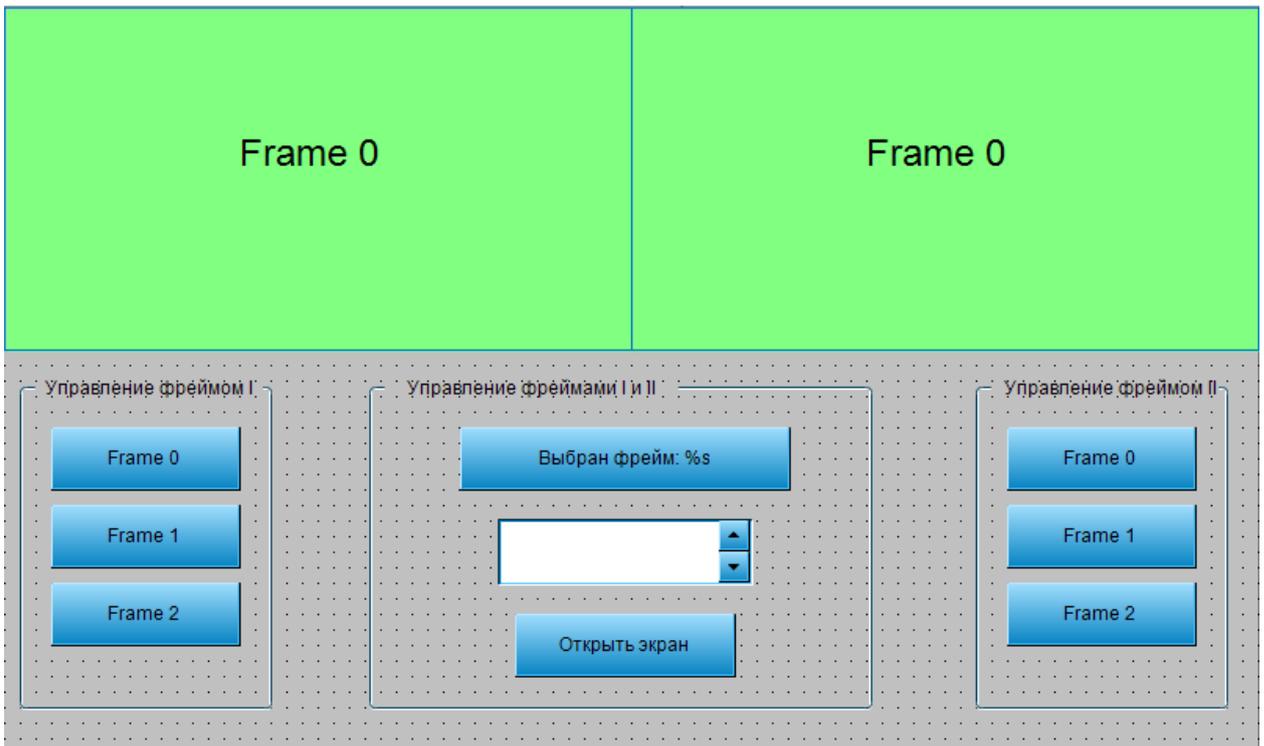


Рис. 11.2.2.7. Экран визуализации Visu (фреймы сконфигурированы)

7. Настроим элемент [Кнопка](#) с названием **Frame 0** панели **Управление фреймом I**, которая будет использоваться для переключения экранов в элементе **Фрейм I** (см. пп. 6).

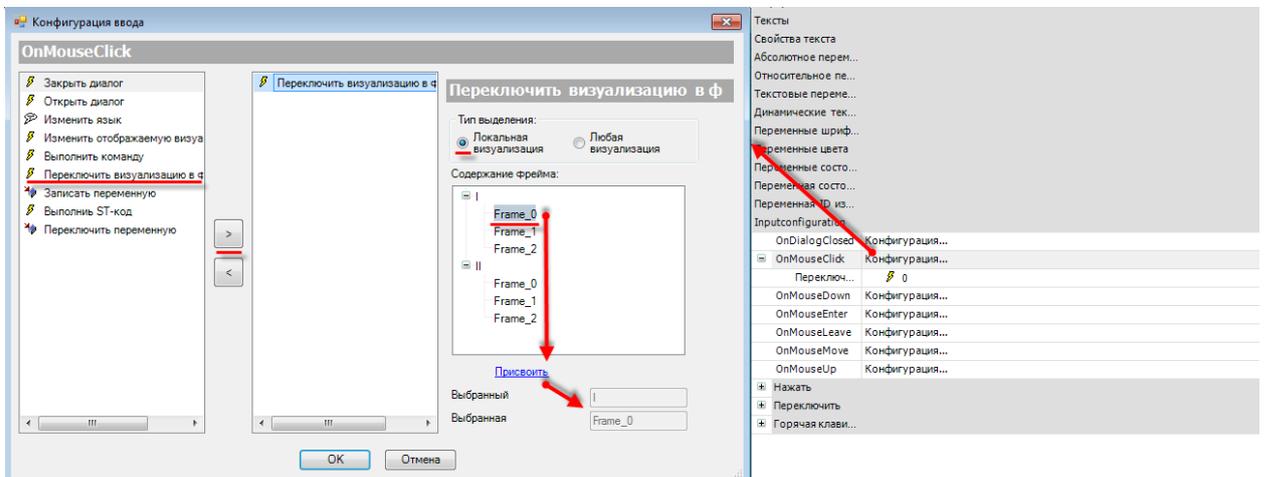


Рис. 11.2.2.8. Настройки действия кнопки Frame_0

Во вкладке [InputConfiguration](#) к параметру **OnMouseClicked** привяжем действие [Переключить визуализацию во фрейме](#). Выберем тип выделения **Локальная визуализация** и присвоим ей экран **Frame_0** элемента **Фрейм I**. Таким образом, по нажатию на кнопку, в элементе **Фрейм I** будет открываться экран **Frame_0**.

8. По аналогии с пп. 7 настроим все кнопки панелей **Управление фреймом I** и **Управление фреймом II**. Очевидно, они будут отличаться только привязанными экранами (см. рис. 11.2.2.8).

9. Настроим панель **Управление фреймом I и II**. С помощью этой панели можно будет переключать экраны в любом из фреймов, предварительно введя его название.

Настроим кнопку **Выбран фрейм**:

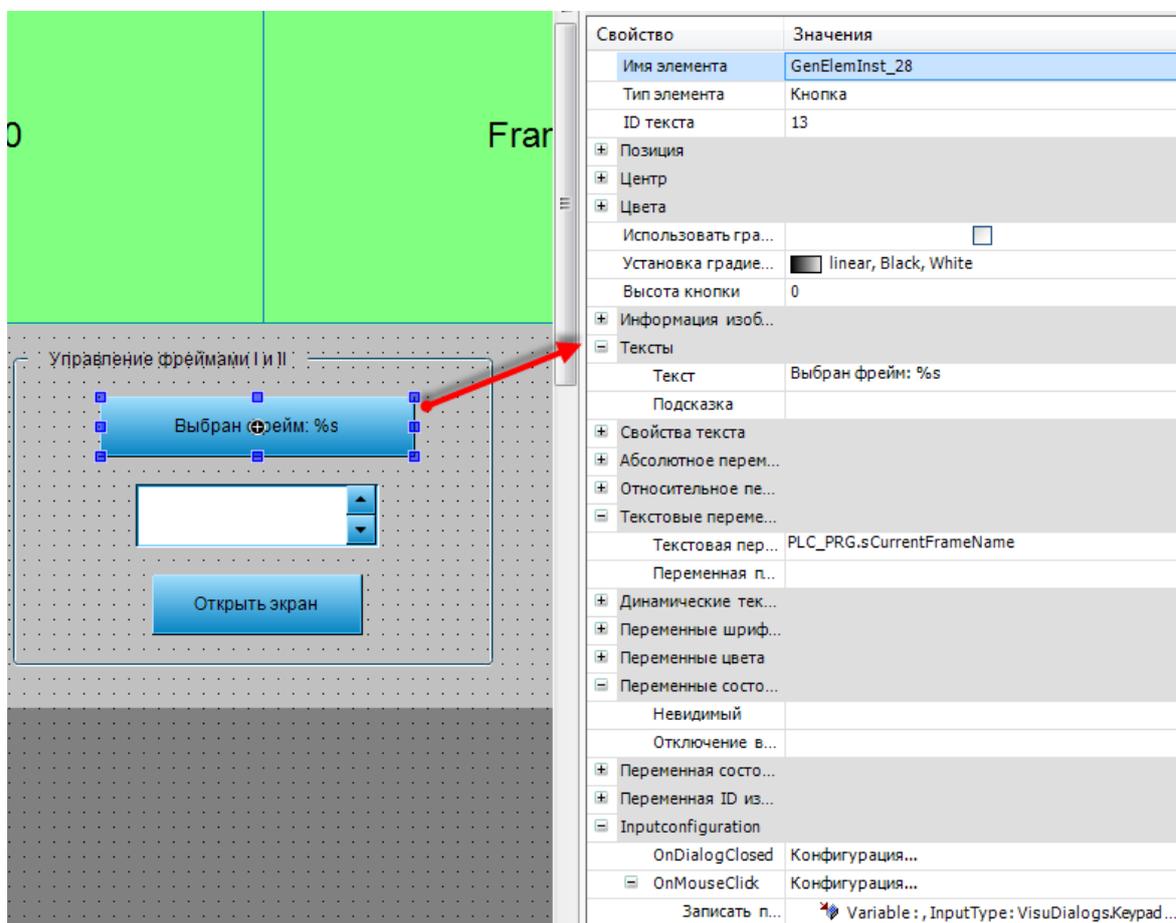


Рис. 11.2.2.9. Настройки кнопки **Выбран фрейм**

Во вкладке [InputConfiguration](#) к параметру **OnMouseClicked** привяжем действие [Записать переменную](#):

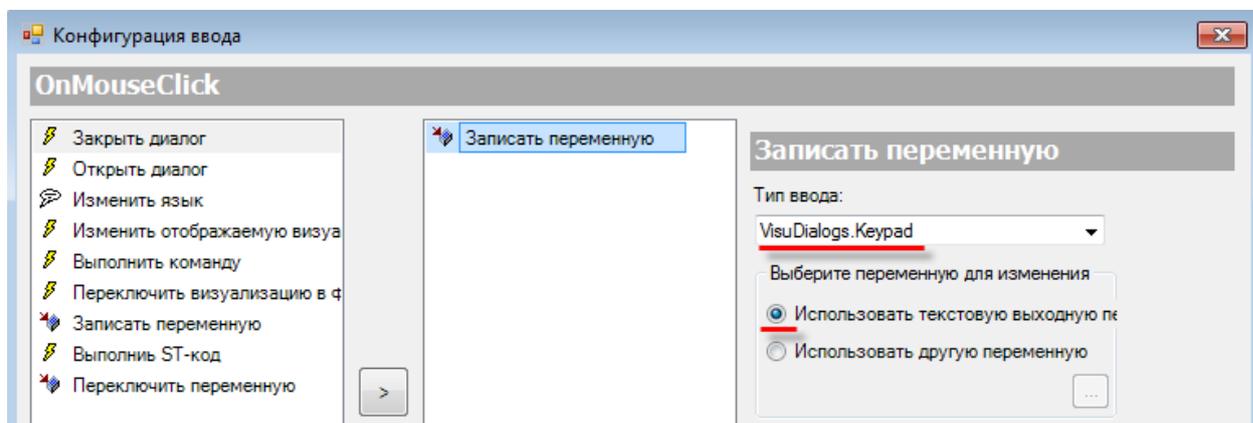


Рис. 11.2.2.10. Настройки действия кнопки **Выбран фрейм**

Настроим элемент [Управление вращением](#), который будет использоваться для выбора индекса экрана, открываемого в фрейме:

Свойство	Значения
Имя элемента	GenElemInst_32
Тип элемента	Управление вращением
Позиция	
Переменная	PLC_PRG.iFrameIndex
Числовой формат	Индекс экрана: %d
Интервал	1
Value range	
Минимальное значение	0
Максимальное значение	2

Рис. 11.2.2.11. Настройки элемента **Управление вращением**

Настроим кнопку **Открыть экран**. Во вкладке [InputConfiguration](#) к параметру **OnClick** привяжем действие [Переключить визуализацию во фрейме](#). Выберем тип выделения **Любая визуализация** и присвоим ей переменные, объявленные в программе **PLC_PRG** (п. 2).

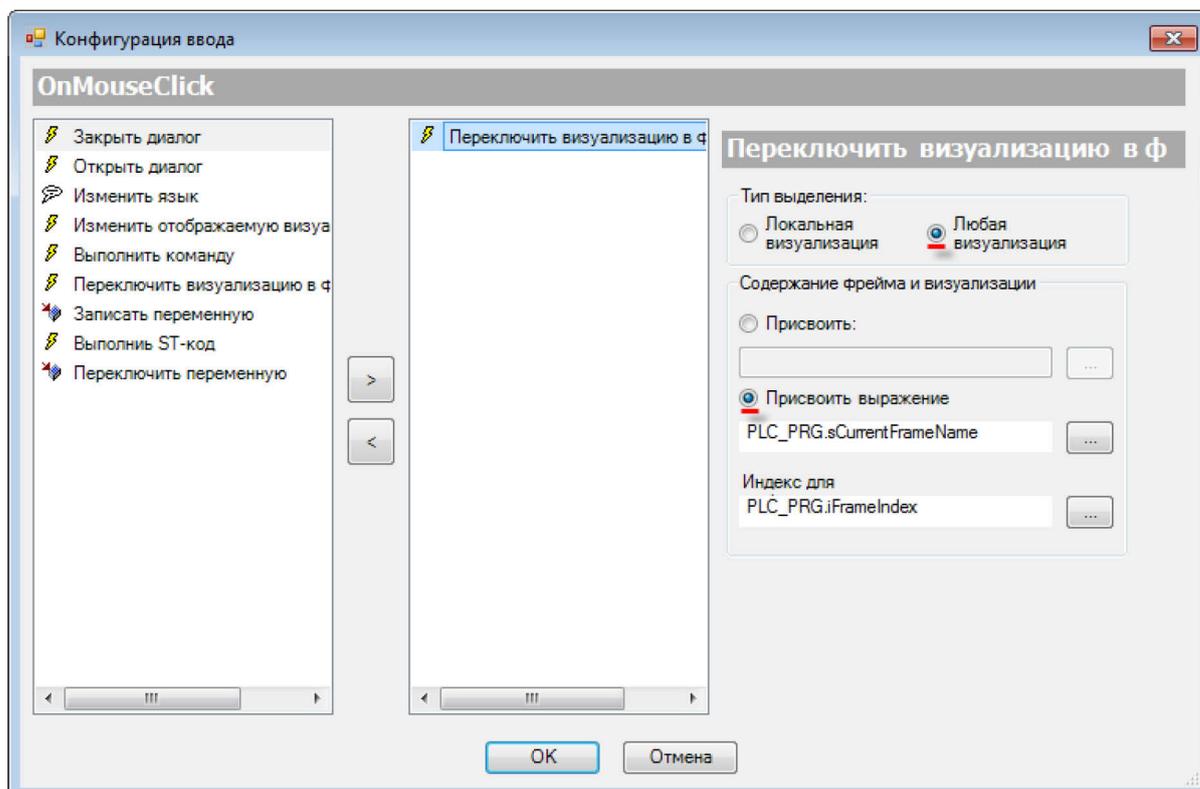
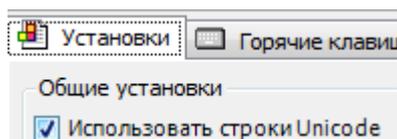


Рис. 11.2.2.12. Настройки действия кнопки **Открыть экран**

10. Настроим компонент [Менеджер визуализации](#). В его установках поставим галочку **Использовать строки Unicode**:



Настройки **target**- и **web**-визуализации будут выглядеть следующим образом:

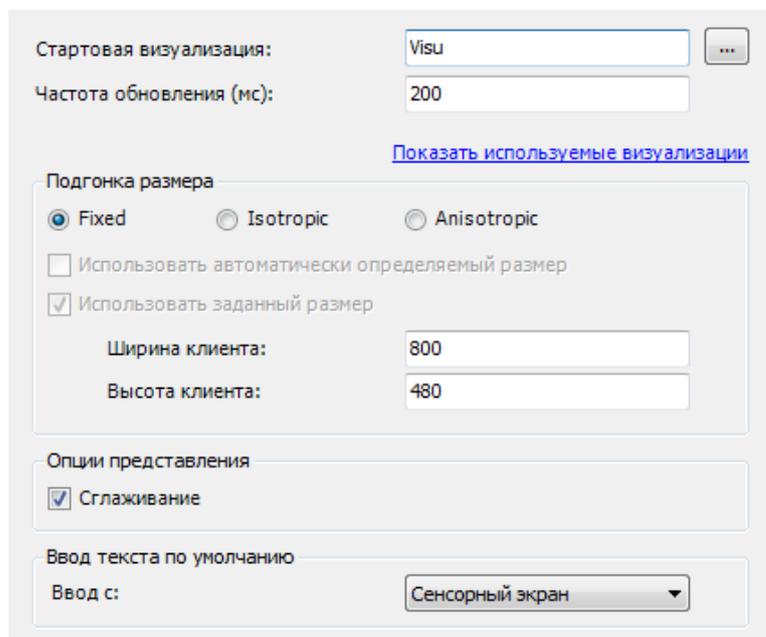


Рис. 11.2.2.13. Настройки **target**-визуализации

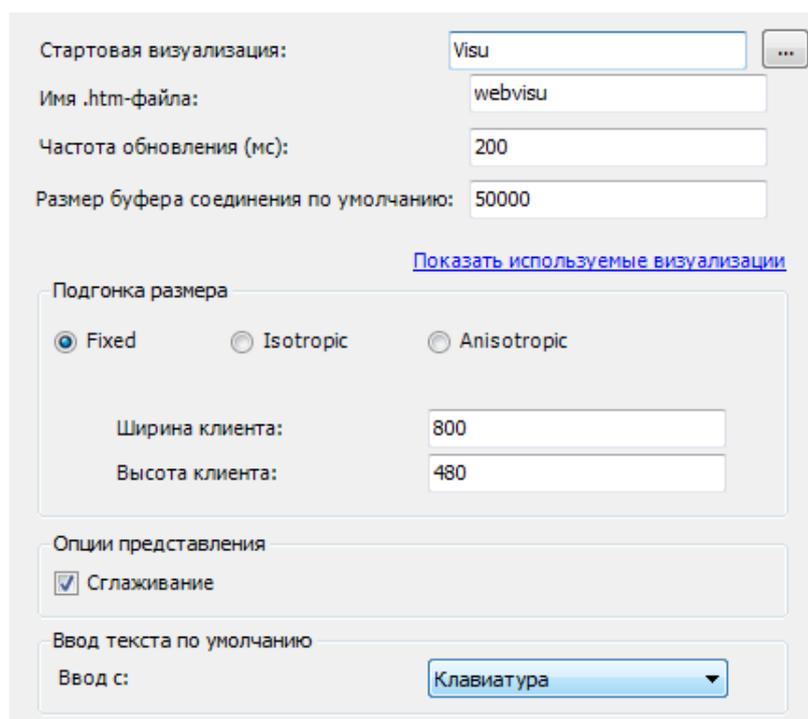


Рис. 11.2.2.14. Настройки **web**-визуализации

11. [Запустим проект на виртуальном контроллере](#). Откроем в веб-браузере страницу <http://localhost:8080/webvisu.htm>, чтобы увидеть web-визуализацию проекта. **Обратите внимание**, что браузер должен поддерживать **HTML5**. Проверим функционал проекта.

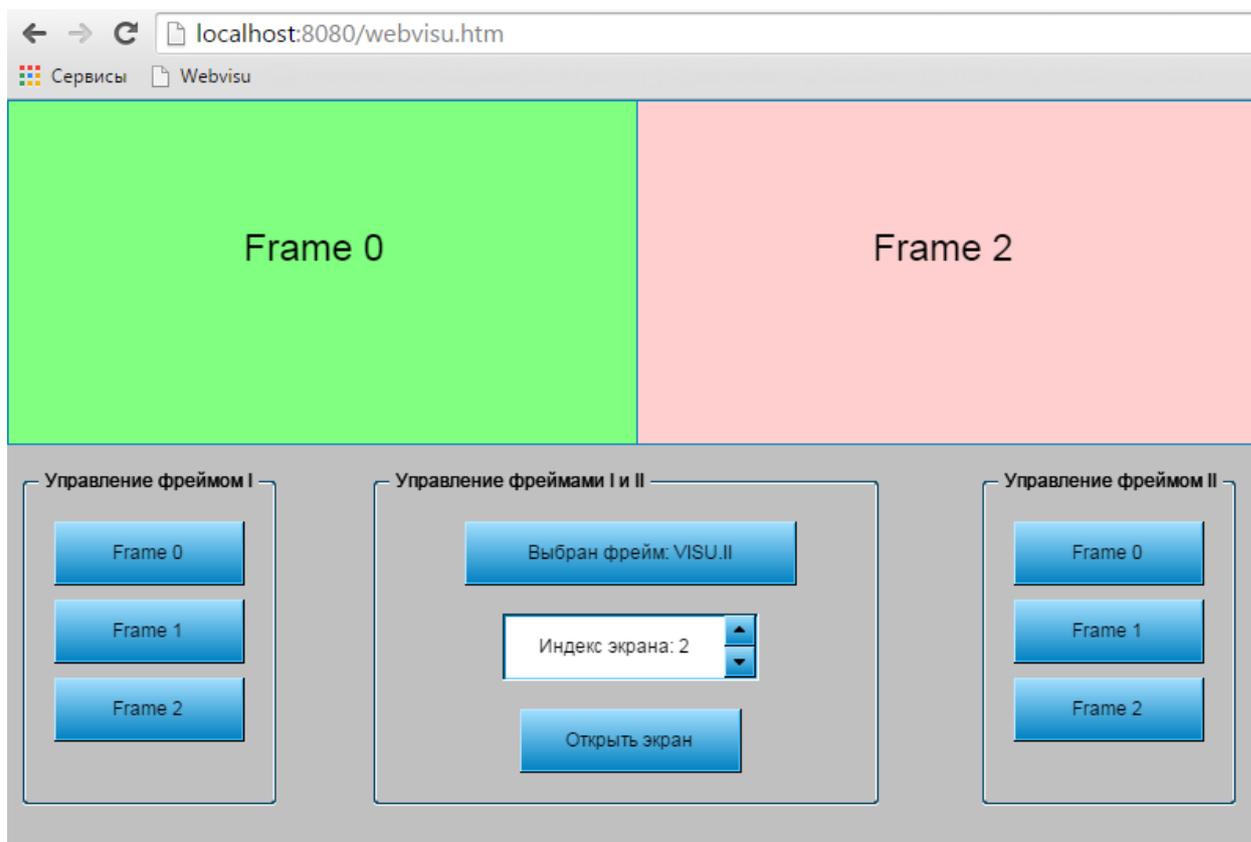


Рис. 11.2.2.15. Web-визуализация проекта в браузере **Chrome**

Нажатие на кнопки **Frame_0**, **Frame_1**, **Frame_2** панели **Управление фреймом I** открывает соответствующий экран в данном фрейме.

Нажатие на кнопки **Frame_0**, **Frame_1**, **Frame_2** панели **Управление фреймом II** открывает соответствующий экран в данном фрейме.

Панель **Управление фреймами I и II** работает следующим образом:

1. нажмите на кнопку **Выбран элемент** и введите путь к элементу в виде **Имя экрана визуализации.Имя элемента** (соответственно, в нашем случае это **Visu.I** для первого фрейма и **Visu.II** для второго);
2. Выберите индекс фрейма;
3. Нажмите на кнопку **Открыть этот фрейм**, чтобы открыть экран с данным индексом в выбранном фрейме.

11.2.3. Комбинированное окно

Данный пример посвящен работе с графическими примитивами [Комбинированное окно – Целочисленное](#) и [Комбинированное окно – Массив](#). Оба этих элемента предназначены для реализации выпадающих списков.

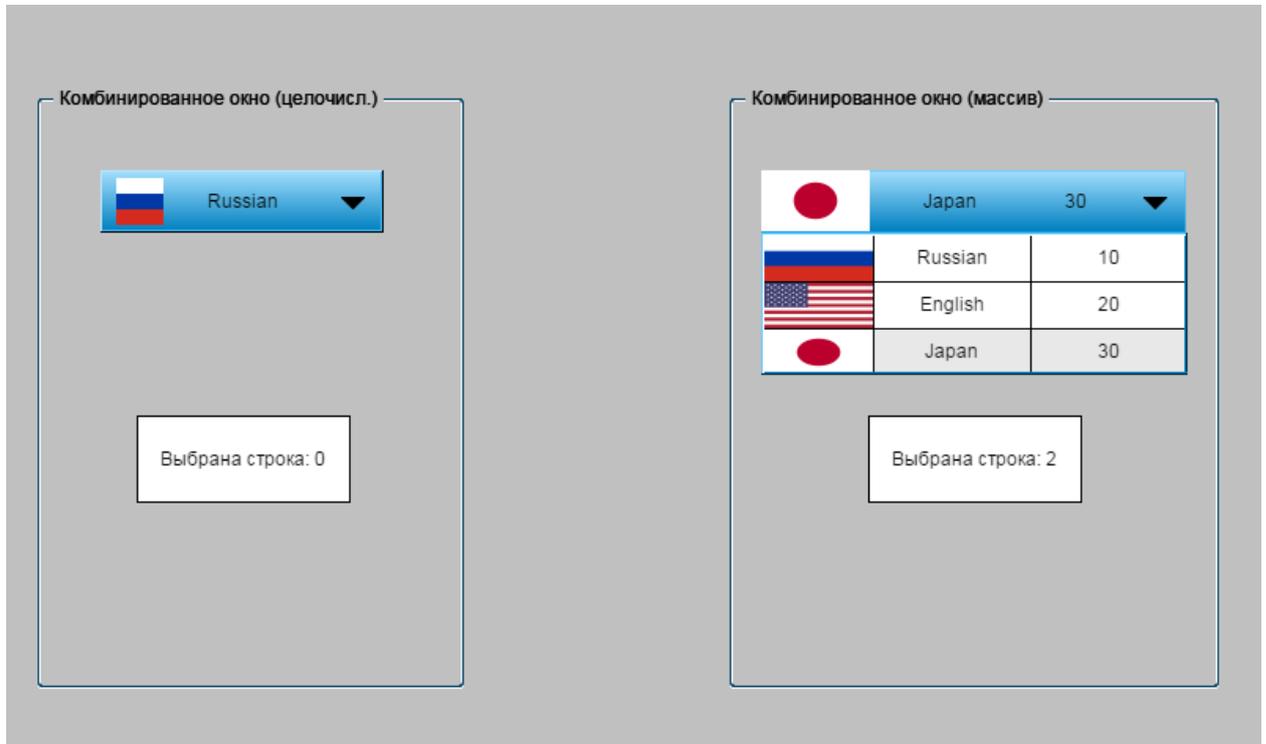


Рис. 11.2.3.1. Внешний вид примера **Комбинированное окно**

Пример создан в среде **CODESYS 3.5 SP6 Patch 2** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с target-файлом **3.5.6.0**.

Пример доступен для скачивания: Example_Combobox.projectarchive

1. Создадим новый **стандартный** проект **CODESYS** с названием **Example_Combobox** и настройками по умолчанию: target – **CODESYS Control Win V3**, язык PLC_PRG – **ST**.

2. В программе **PLC_PRG** объявим следующие переменные:

```

1 PROGRAM PLC_PRG
2 VAR
3     iVar1:INT; // переменная, содержащая номер строки, выбранной элементом Комбинированное окно - Целочисленное
4     iVar2:INT; // переменная, содержащая номер строки, выбранной элементом Комбинированное окно - Массив
5
6     arrTable : ARRAY [0..2, 0..2] OF STRING := ['0','1','2', // Массив для элемента...
7                                                'Russian','English','Japan', // ...Комбинированное окно - Массив
8                                                '10','20','30'];
9 END VAR

```

Рис. 11.2.3.2. Объявление переменных программы **PLC_PRG**

3. Добавим в проект [Пул изображений](#) с названием **ImagePool**. Загрузим в него три картинки с ID 0, 1 и 2:

ID	Имя файла	Изображение	Link type
0	Flag_of_Russia.svg		Embedded and link to file
1	Flag_of_the_United_States.svg		Embedded and link to file
2	Flag_of_Japan.svg		Embedded and link to file

Рис. 11.2.3.3. Содержимое пула изображений **ImagePool**

4. Добавим в проект [СПИСОК ТЕКСТОВ](#) с названием **TextList**:

ID	По умолчанию
0	Текст 1
1	Текст 2
2	Текст 3

Рис. 11.2.3.4. Содержимое списка текстов **TextList**

5. Добавим в проект экран визуализации с названием **Visualization**. В его свойствах выберем размер **800x480**. Добавим на экран два элемента **Группа**, элемент **Комбинированное окно – Целочисленное**, элемент **Комбинированное окно – Массив** и два элемента **Прямоугольник**.

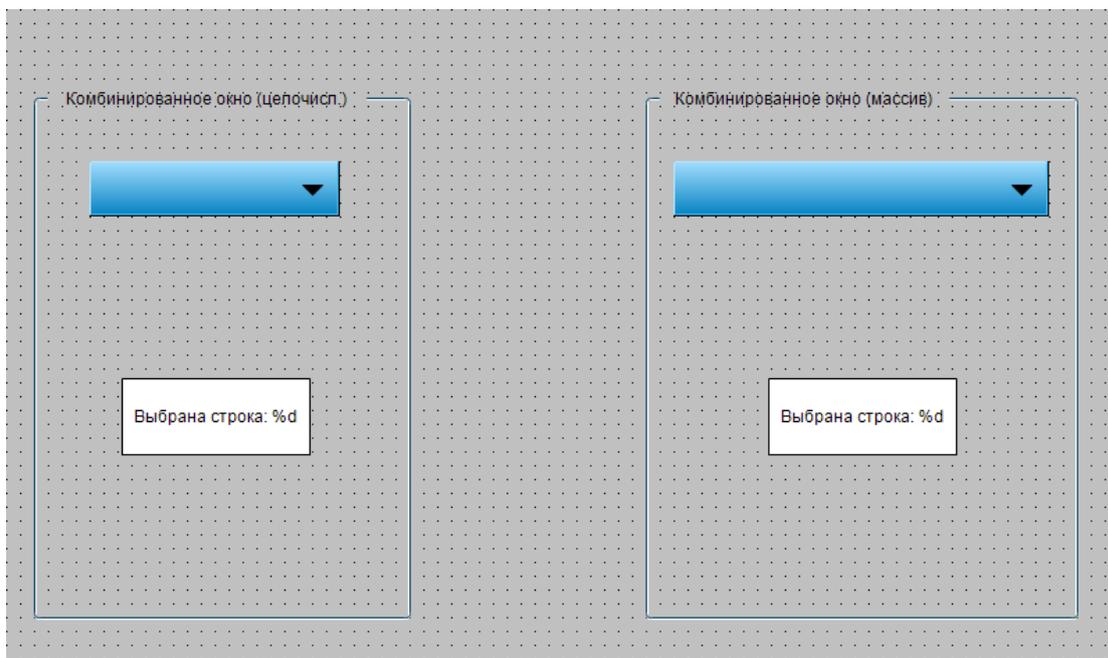


Рис. 11.2.3.5. Содержимое экрана **Visualization**

6. Настроим элемент **Комбинированное окно – Целочисленное**:

Свойства	
Фильтр	Сортировать по
Порядок сортировки	Эксперт
Свойство	Значения
Имя элемента	GenElemInst_8
Тип элемента	Комбинированное окно - Целочисленный
Позиция	
X	40
Y	60
Ширина	180
Высота	40
Переменная	PLC_PRG.iVar1
Список текстов	'TextList'
Пул изображений	'ImagePool'
Максимальное значение	2
Параметры списка	
Тексты	
Поддиапазон	
Использовать поддиапазон	<input checked="" type="checkbox"/>
Начальный индекс	0
Конечный индекс	2
Отбрасывать недостающ...	<input checked="" type="checkbox"/>
Свойства текста	
Переменные состояний	

Рис. 11.2.3.6. Параметры элемента **Комбинированное окно - Целочисленное**

7. Настроим элемент [Комбинированное окно – Массив](#):

Свойства	
Фильтр <input type="checkbox"/> Сортировать по <input type="checkbox"/> Порядок сортировки <input checked="" type="checkbox"/> Эксперт	
Свойство	Значения
Имя элемента	GenElemInst_14
Тип элемента	Комбинированное окно - Массив
Позиция	
Переменная	PLC_PRG.iVar2
Массив данных	PLC_PRG.arrTable
Столбцы	
Столбец	
[0]	<input checked="" type="checkbox"/>
Ширина	70
Столбец изображ...	<input checked="" type="checkbox"/>
Конфигурация изо...	
Выравнивание те...	По центру
[1]	<input checked="" type="checkbox"/>
Ширина	100
Столбец изображ...	<input type="checkbox"/>
Конфигурация изо...	
Выравнивание те...	По центру
[2]	<input checked="" type="checkbox"/>
Ширина	100
Столбец изображ...	<input type="checkbox"/>
Конфигурация изо...	
Выравнивание те...	По центру
Макс. индекс массива	2
Высота строки	30
Число видимых строк	3
Размер полосы прокрутки	20

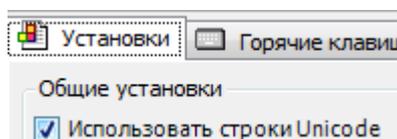
Рис. 11.2.3.7. Параметры элемента **Комбинированное окно - Массив**

8. Настроим элементы [Прямоугольник](#). Они будут отличаться только привязываемыми переменными – **iVar1** для первого и **iVar2** для второго соответственно.

Свойство	Значения
Имя элемента	GenElemInst_10
ID текста	7
Тип элемента	Прямоугольник
<input checked="" type="checkbox"/> Позиция	
<input checked="" type="checkbox"/> Центр	
<input checked="" type="checkbox"/> Цвета	
Использовать градиентный ц...	<input type="checkbox"/>
Установка градиента	linear, Black, White
<input checked="" type="checkbox"/> Вид элемента	
<input checked="" type="checkbox"/> Тексты	
Текст	Выбрана строка: %d
Подсказка	
<input checked="" type="checkbox"/> Свойства текста	
<input checked="" type="checkbox"/> Абсолютное перемещение	
<input checked="" type="checkbox"/> Относительное перемещение	
<input checked="" type="checkbox"/> Текстовые переменные	
Текстовая переменная	PLC_PRG.iVar1
Переменная подсказки	

Рис. 11.2.3.8. Параметры элемента **Прямоугольник 1**

9. Настроим [Менеджер визуализации](#). В его установках поставим галочку **Использовать строки Unicode**:



Настройки **target-** и **web-**визуализации будут выглядеть следующим образом:

Стартовая визуализация: Visu

Частота обновления (мс): 200

[Показать используемые визуализации](#)

Подгонка размера

Fixed
 Isotropic
 Anisotropic

Использовать автоматически определяемый размер
 Использовать заданный размер

Ширина клиента: 800
Высота клиента: 480

Опции представления

Сглаживание

Ввод текста по умолчанию

Ввод с: Сенсорный экран

Рис. 11.2.3.9. Настройки **target-**визуализации

Стартовая визуализация: Visu

Имя .htm-файла: webvisu

Частота обновления (мс): 200

Размер буфера соединения по умолчанию: 50000

[Показать используемые визуализации](#)

Подгонка размера

Fixed Isotropic Anisotropic

Ширина клиента: 800

Высота клиента: 480

Опции представления

Сглаживание

Ввод текста по умолчанию

Ввод с: Клавиатура

Рис. 11.2.3.10. Настройки **web**-визуализации

10. [Запустим проект на виртуальном контроллере](#). Откроем в веб-браузере страницу <http://localhost:8080/webvisu.htm>, чтобы увидеть web-визуализацию проекта. **Обратите внимание**, что браузер должен поддерживать **HTML5**. Проверим функционал проекта.

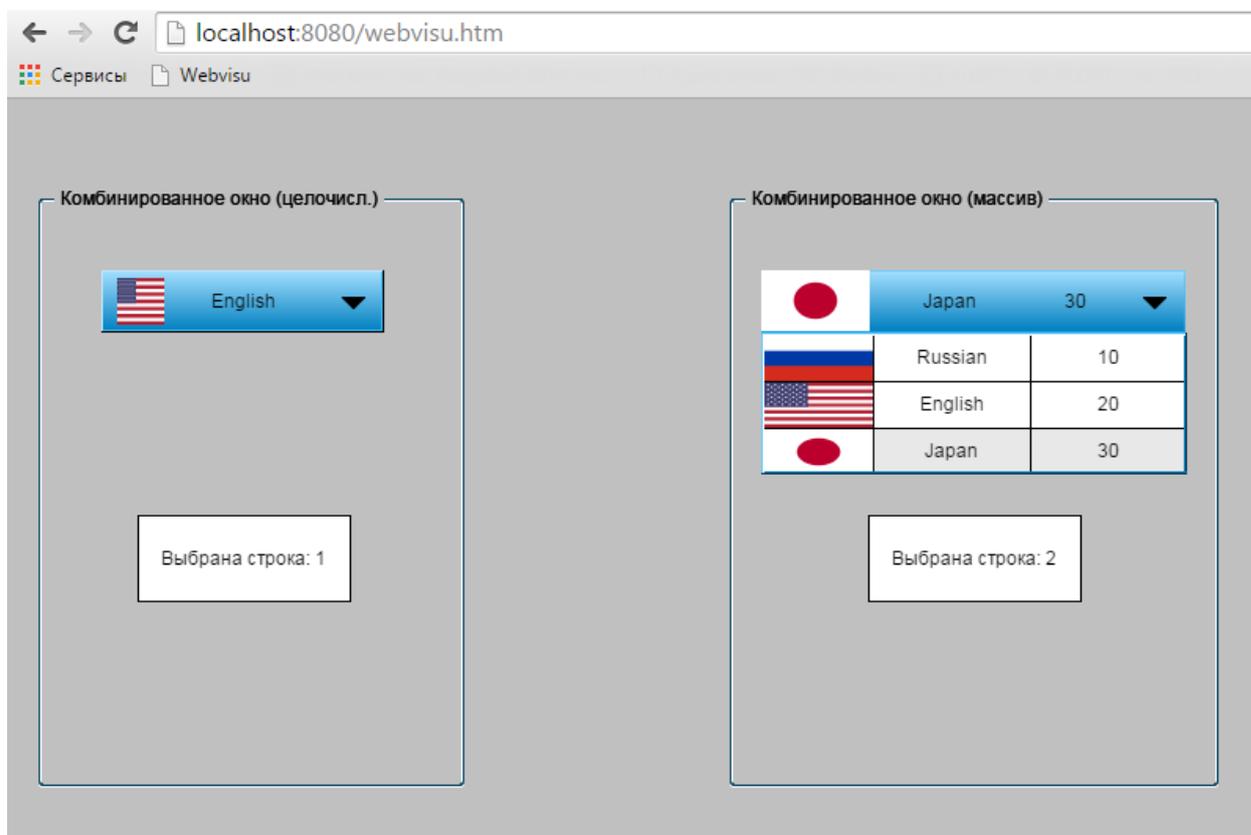


Рис. 11.2.3.11. Web-визуализация проекта в браузере **Chrome**

После нажатия на элемент **Комбинированное окно – Целочисленное** раскрывается список, сформированный из содержимого [Пула изображений](#) и [Списка текстов](#). Выбор одной из строк приводит к записи в переменную **iVar1** номера выбранной строки.

После нажатия на элемент **Комбинированное окно – Массив** раскрывается список, сформированный из содержимого переменной, привязанной к элементу. Выбор одной из строк приводит к записи в переменную **iVar2** номера выбранной строки.

11.2.4. Кнопка

Данный пример посвящен работе с графическим примитивом [Кнопка](#). Примитив имеет две основные функции:

1. переключение значений переменных типа **BOOL**;
2. выполнение действий, привязанных во вкладке [InputConfiguration](#).

В рамках этого примера будет рассмотрена только первая функция кнопки.

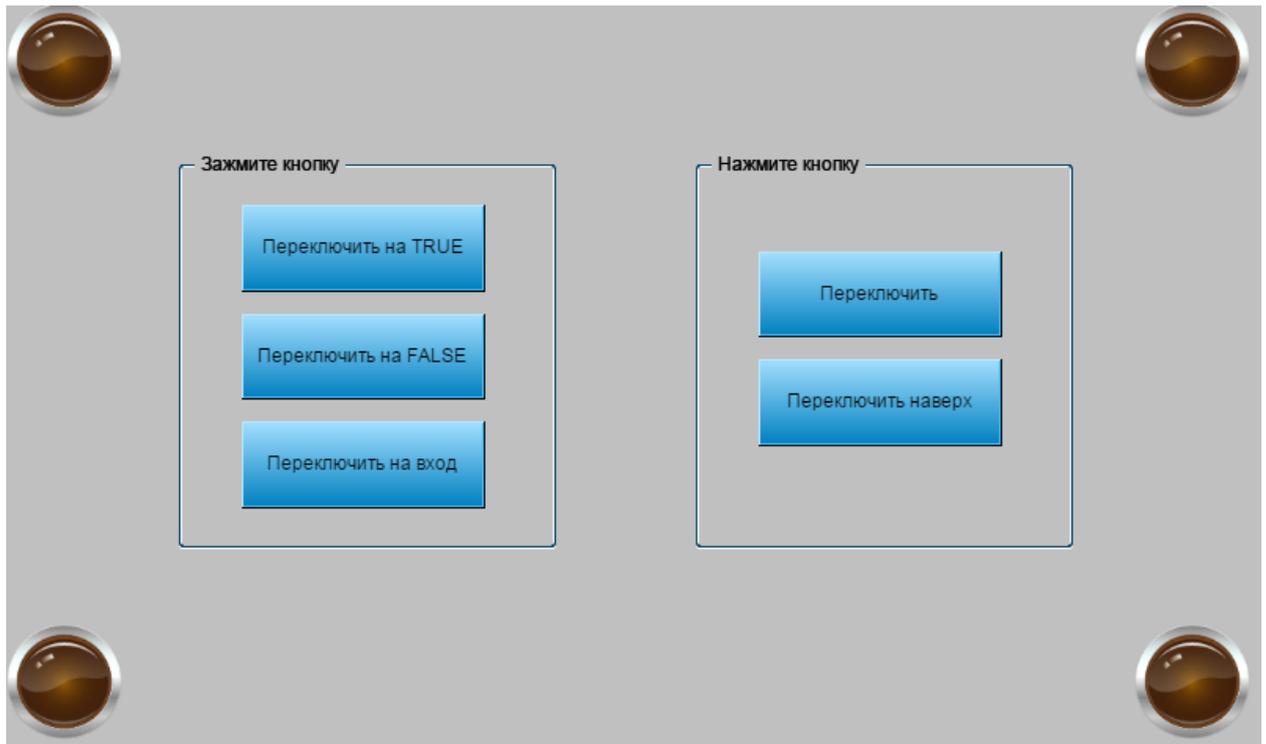


Рис. 11.2.4.1. Внешний вид примера **Кнопка**

Пример создан в среде **CODESYS 3.5 SP6 Patch 2** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с target-файлом **3.5.6.0**.

Пример доступен для скачивания: [Example Button.projectarchive](#)

1. Создадим новый *стандартный* проект **CODESYS** с названием **Example_Button** и настройками по умолчанию: target – **CODESYS Control Win V3**, язык PLC_PRG – **ST**.

2. В программе **PLC_PRG** объявим переменную **bVar** типа **BOOL**:

```
1 | PROGRAM PLC_PRG
2 | VAR
3 |     bVar:BOOL;
4 | END VAR
```

Рис. 11.2.4.2. Объявление переменных программы **PLC_PRG**

3. Добавим в проект экран визуализации **Visualization**. В его свойствах выберем размер **800x480**. Экран будет содержать четыре элемента Индикатор, два элемента Группа и пять элементов Кнопка.

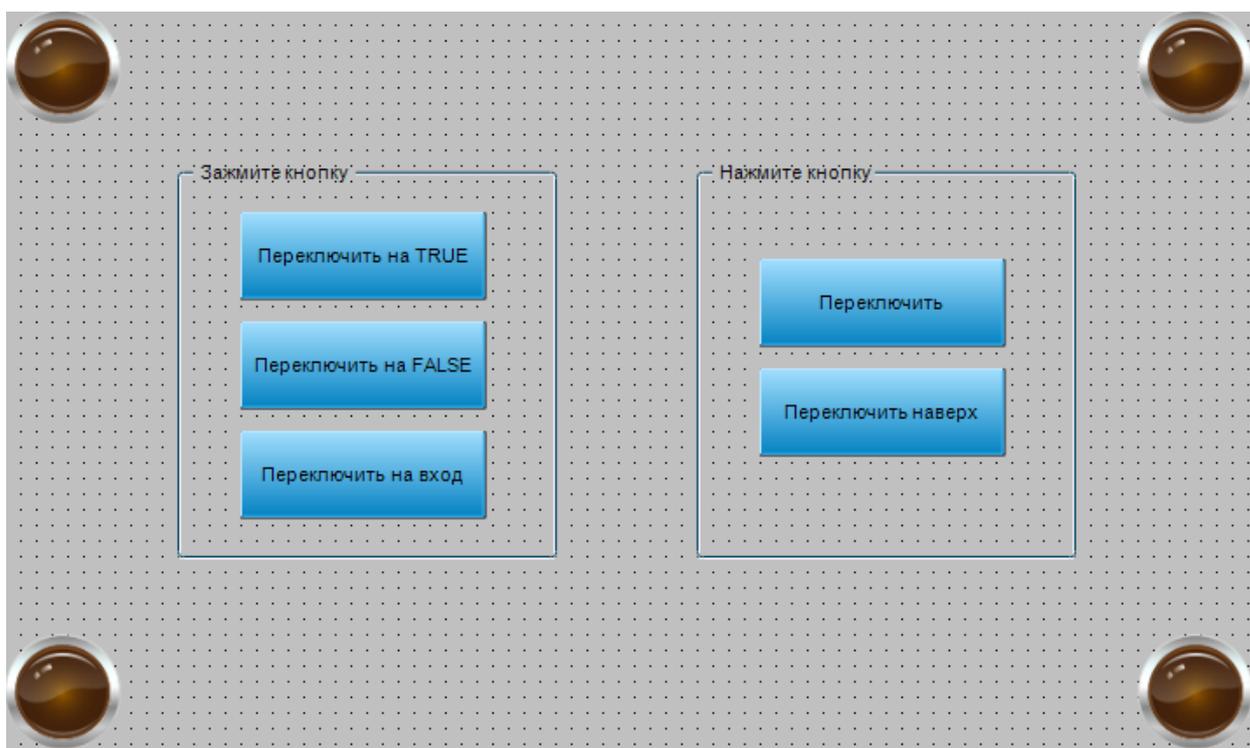


Рис. 11.2.4.3. Содержимое экрана **Visualization**

4. К каждому из четырех [индикаторов](#) привяжем переменную **bVar**:

Свойства	
Свойство	Значения
Имя элемента	GenElemInst_28
Тип элемента	Индикатор
[-] Позиция	
X	0
Y	0
Ширина	75
Высота	75
Переменная	PLC_PRG.bVar
[+] Параметры изоб...	
[+] Тексты	
[+] Переменные сос...	
[-] Фон	
Изображение	Yellow

Рис. 11.2.4.4. Настройки элемента **Индикатор**

5. Настроим [кнопку](#) **Переключить на TRUE**:

[-] Inputconfiguration	
OnDialogClosed	Конфигурация...
OnMouseClicked	Конфигурация...
OnMouseDown	Конфигурация...
OnMouseEnter	Конфигурация...
OnMouseLeave	Конфигурация...
OnMouseMove	Конфигурация...
OnMouseUp	Конфигурация...
[-] Нажать	
Переменная	PLC_PRG.bVar
Переключить на FALSE	<input type="checkbox"/>
Переключить на вхо...	<input type="checkbox"/>
[-] Переключить	
Переменная	
Переключить наверх..	<input type="checkbox"/>
[+] Горячая клавиша	

Рис. 11.2.4.5. Настройки кнопки **Переключить на TRUE**

6. Настроим [кнопку](#) **Переключить на FALSE**:

Inputconfiguration		
OnDialogClosed	Конфигурация...	
OnMouseClicked	Конфигурация...	
OnMouseDown	Конфигурация...	
OnMouseEnter	Конфигурация...	
OnMouseLeave	Конфигурация...	
OnMouseMove	Конфигурация...	
OnMouseUp	Конфигурация...	
Нажать		
Переменная	PLC_PRG.bVar	
Переключить на FALSE		<input checked="" type="checkbox"/>
Переключить на вход, если...		<input type="checkbox"/>
Переключить		
Переменная		
Переключить наверх, если н...		<input type="checkbox"/>
Горячая клавиша		

Рис. 11.2.4.6. Настройки кнопки **Переключить на FALSE**

7. Настроим [кнопку](#) **Переключить на вход**:

Inputconfiguration		
OnDialogClosed	Конфигурация...	
OnMouseClicked	Конфигурация...	
OnMouseDown	Конфигурация...	
OnMouseEnter	Конфигурация...	
OnMouseLeave	Конфигурация...	
OnMouseMove	Конфигурация...	
OnMouseUp	Конфигурация...	
Нажать		
Переменная	PLC_PRG.bVar	
Переключить на FALSE		<input type="checkbox"/>
Переключить на вход, если нажата		<input checked="" type="checkbox"/>

Рис. 11.2.4.7. Настройки кнопки **Переключить на вход**

8. Настроим [кнопку Переключить](#):

[-] Переменная состояния кнопки		
Двоичная переменная	PLC_PRG.bVar	
[+] Переменная ID изображения		
[-] Inputconfiguration		
OnDialogClosed	Конфигурация...	
OnMouseClicked	Конфигурация...	
OnMouseDown	Конфигурация...	
OnMouseEnter	Конфигурация...	
OnMouseLeave	Конфигурация...	
OnMouseMove	Конфигурация...	
OnMouseUp	Конфигурация...	
[-] Нажать		
Переменная		
Переключить на FALSE		<input type="checkbox"/>
Переключить на вход, если нажата		<input type="checkbox"/>
[-] Переключить		
Переменная	PLC_PRG.bVar	
Переключить наверх, если нажата		<input type="checkbox"/>
[-] Горячая клавиша		
Клавиша	Space	
События	Mouse down/up	
Shift		<input type="checkbox"/>
Control		<input type="checkbox"/>
Alt		<input type="checkbox"/>

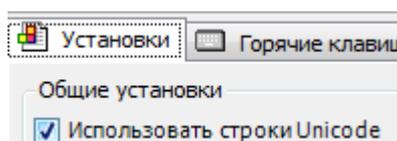
Рис. 11.2.4.8. Настройки кнопки **Переключить**

9. Настроим [кнопку Переключить наверх](#):

[-] Inputconfiguration		
OnDialogClosed	Конфигурация...	
OnMouseClicked	Конфигурация...	
OnMouseDown	Конфигурация...	
OnMouseEnter	Конфигурация...	
OnMouseLeave	Конфигурация...	
OnMouseMove	Конфигурация...	
OnMouseUp	Конфигурация...	
[-] Нажать		
Переменная		
Переключить на FALSE		<input type="checkbox"/>
Переключить на вход, если нажата		<input type="checkbox"/>
[-] Переключить		
Переменная	PLC_PRG.bVar	
Переключить наверх, если нажата		<input checked="" type="checkbox"/>

Рис. 11.2.4.9. Настройки кнопки **Переключить наверх**

10. Настроим компонент [Менеджер визуализации](#). В его установках поставим галочку **Использовать строки Unicode**:



Настройки **target**- и **web**-визуализации будут выглядеть следующим образом:

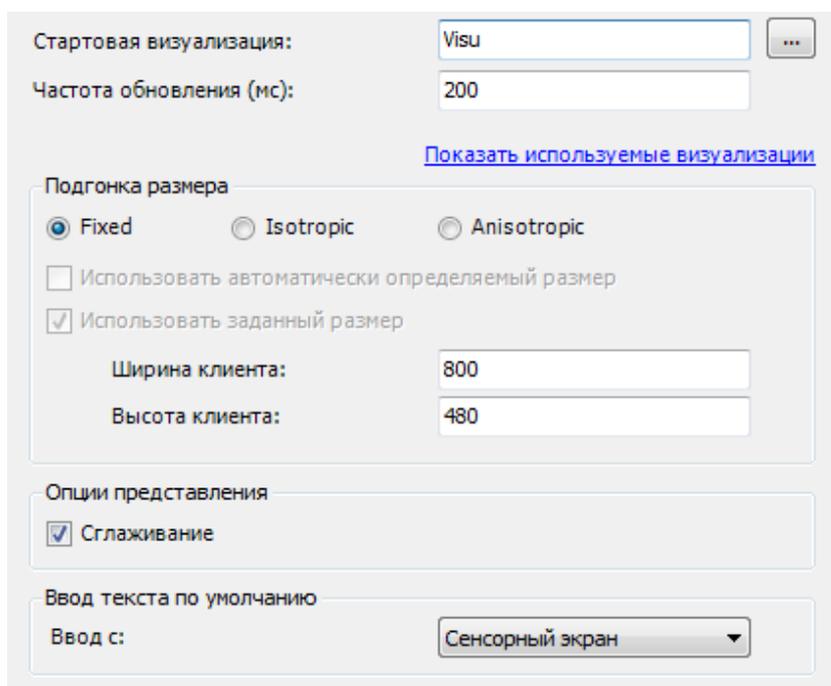


Рис. 11.2.4.10. Настройки **target**-визуализации

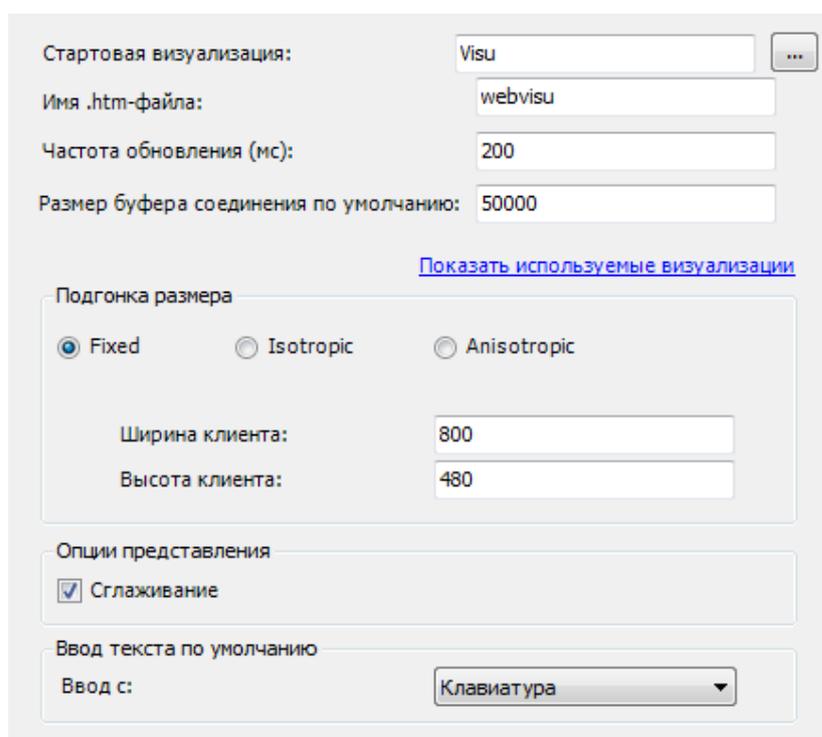


Рис. 11.2.4.11. Настройки **web**-визуализации

11. [Запустим проект на виртуальном контроллере](#). Откроем в веб-браузере страницу <http://localhost:8080/webvisu.htm>, чтобы увидеть web-визуализацию проекта. **Обратите внимание**, что браузер должен поддерживать **HTML5**. Проверим функционал проекта (пункты выполняются последовательно).



Рис. 11.2.4.12. Web-визуализация проекта в браузере **Chrome**

1. При зажатии кнопки **Переключить на TRUE**, переменная **bVar** переключается в **TRUE** – лампы загораются. При отпуске кнопки – переменная возвращается в значение **FALSE**, лампы потухают.
2. После первого нажатия на кнопку **Переключить на FALSE**, переменная переключится в состояние **TRUE** (этот эффект связан с тем, что мы пытаемся переключить в **FALSE** переменную, которая и так находится в этом состоянии). Последующее зажатие кнопки приводит к переключению переменной в **FALSE**, отпущение – к возвращению ее в **TRUE**.
3. После первого нажатия на кнопку **Переключить на вход**, переменная переключится в состояние **FALSE** (этот эффект связан с тем, что мы пытаемся переключить в **TRUE** переменную, которая и так находится в этом состоянии). Последующее зажатие кнопки приводит к переключению переменной в **TRUE**. Не отпуская кнопку, перетащите курсор за ее пределы – переменная переключится в **FALSE**. Верните курсор в пределы элемента – переменная переключится в **TRUE**. Отпустите кнопку – переменная переключится в **FALSE**.

4. После нажатия на кнопку **Переключить**, переменная изменит свое состояние на противоположное (с **FALSE** на **TRUE** или с **TRUE** на **FALSE**). При этом в состоянии **TRUE** кнопка будет отображаться «зажатой». В web-визуализации вместо нажатия на саму кнопку, можно нажимать на кнопку **Пробел** на клавиатуре.

5. Нажмите на кнопку **Переключить вверх** и, не отпуская ее, переместите курсор за пределы элемента, после чего отпустите кнопку. Переменная изменит свое состояние на противоположное (с **FALSE** на **TRUE** или с **TRUE** на **FALSE**).

11.2.5. Таблица

Данный пример посвящен работе с графическим примитивом [Таблица](#). В рамках примера рассмотрена запись в таблицу структуры данных по команде пользователя.

	Время	Значение
0	07:57:45 27.06.2016	16.40003
1	07:57:48 27.06.2016	28.20007
2	07:57:49 27.06.2016	36.40002
3	07:57:52 27.06.2016	46.69986
4	07:57:53 27.06.2016	55.89972
5	07:57:56 27.06.2016	71.39949
6	07:58:02 27.06.2016	97.89908
7	07:58:08 27.06.2016	131.1988
8		
9		
10		

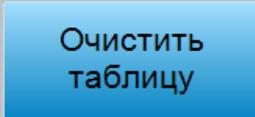
 Запись в таблицу	Выделен столбец №1		Выделен один из столбцов	 Очистить таблицу
	Выделена строка №4		Выделена одна из строк	

Рис. 11.2.5.1. Внешний вид примера **Таблица**

Пример создан в среде **CODESYS 3.5 SP6 Patch 2** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с target-файлом **3.5.6.0**.

Пример доступен для скачивания: [Example_Table.projectarchive](#)

1. Создадим новый **стандартный** проект **CODESYS** с названием **Example_Table** и настройками по умолчанию: target – **CODESYS Control Win V3**, язык PLC_PRG – **ST**.

2. Добавим в проект компонент **DUT** типа **Структура** с названием **TableStruct**:

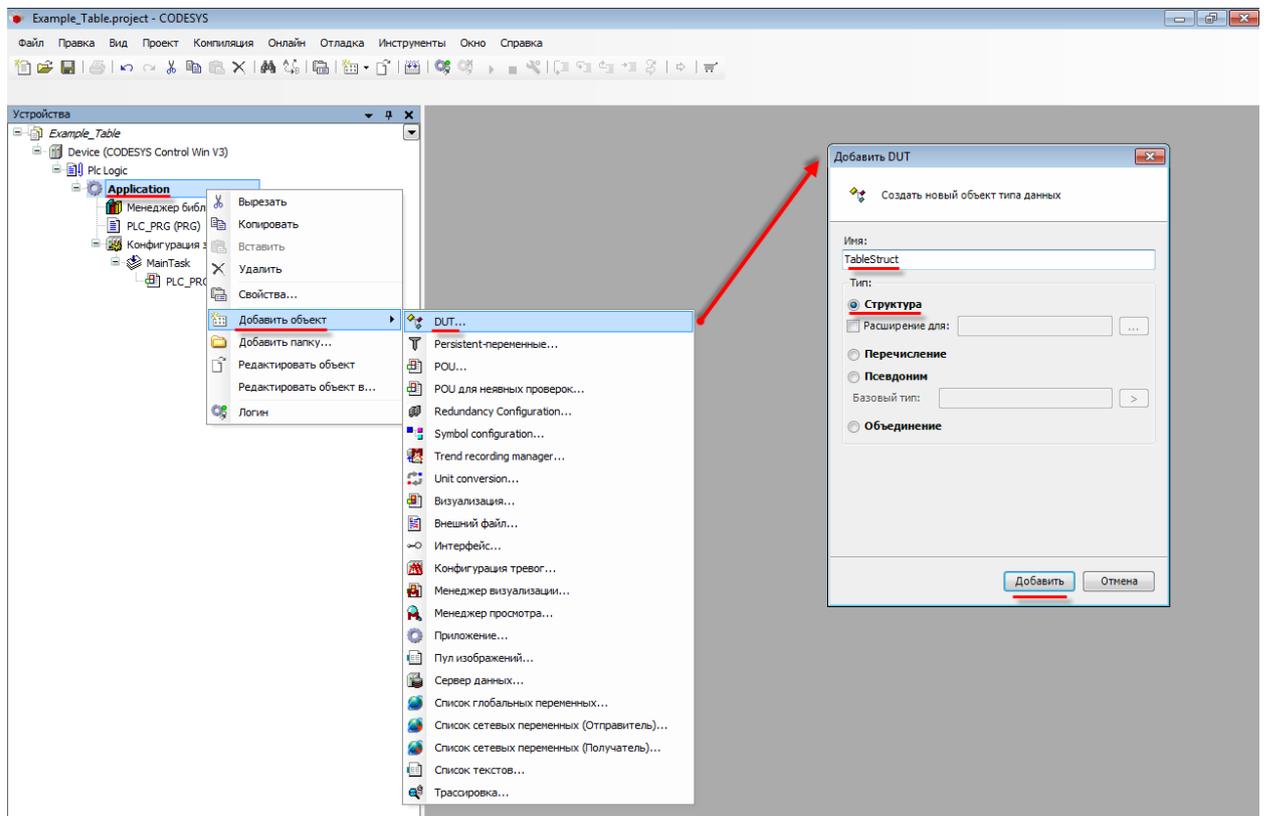


Рис. 11.2.5.2. Добавление в проект структуры **TableStruct**

3. Структура представляет собой пользовательский тип данных, который содержит набор переменных разных типов. Наша структура будет содержать две переменные типа **STRING**:

```
1 TYPE TableStruct :           // структура, отображаемая в таблице
2 STRUCT
3     sDateAndTime:STRING; // столбец метки времени
4     sValue:STRING;         // столбец значений
5 END_STRUCT
6 END_TYPE
7
```

Рис. 11.2.5.3. Объявление переменных структуры **TableStruct**

Почему мы используем тип **STRING**? Это связано с наличием у переменных начальных значений (по умолчанию – «нулевых»). При использовании типов **DT** и **REAL**, в начальный момент времени наша таблица имела бы приблизительно такой вид:

dt#1970-01-01-00:00:00	0.0

Чтобы избежать этого, мы используем переменные типа **STRING**, начальные значения которых (") в визуализации представляются пустой строкой.

4. В Менеджере библиотек добавим библиотеку **Time and Date** версии 3.5.1.0 (она понадобится нам, чтобы работать с системным временем контроллера):

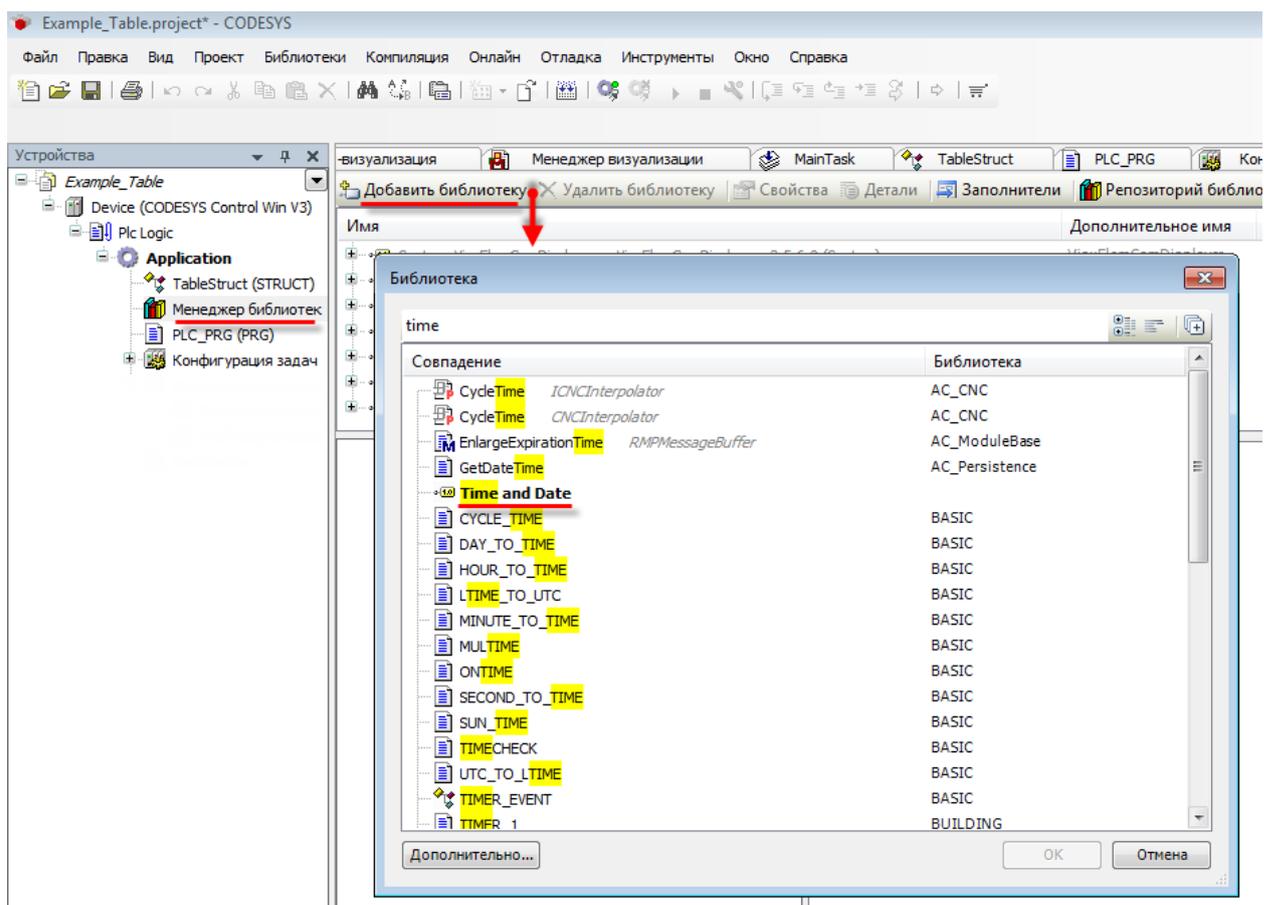


Рис. 11.2.5.4. Добавление в проект библиотеки **Time and Date**

5. В программе **PLC_PRG** объявим следующие переменные:

```

1 | PROGRAM PLC_PRG
2 | VAR
3 |     arrTable: ARRAY [0..10] OF TableStruct; // массив для структуры
4 |
5 |     MyGetDateAndTime: DTU.GetDateAndTime; // экземпляр функционального блока чтения системного времени
6 |     dtCurrentDateAndTime:DT; // системное время в формате DT
7 |
8 |     bTrigger:BOOL; // триггер записи
9 |
10 |     rValue:REAL; // переменная, значение которой отображается в таблице
11 |     iIndex:INT; // номер строки таблицы, в которую осуществляется запись
12 |
13 |     iCounter:INT; // счетчик (для очистки таблицы по нажатию на кнопку)
14 |
15 |     iColumnSelection:INT; // номер выделенного столбца
16 |     iLineSelection:INT; // номер выделенной строки
17 |     bColumnSelection:BOOL; // флаг выделения любого из столбцов таблицы
18 |     bLineSelection:BOOL; // флаг выделения любой из строк таблицы
19 | END_VAR

```

Рис. 11.2.5.5. Объявление переменных программы **PLC_PRG**

6. Код программы **PLC_PRG** будет выглядеть следующим образом:

```
1   rValue:=rValue+0.1; // эмуляция изменения значений переменной
2
3   MyGetDateAndTime (xExecute:=NOT(MyGetDateAndTime.xDone)); // вызов ФБ считывания системного времени
4
5   IF MyGetDateAndTime.xDone=TRUE THEN // если ФБ успешно завершил работу...
6       dtCurrentDateAndTime:=MyGetDateAndTime.dtDateAndTime; // ...то считыванием системное время в переменную
7   END_IF
8
9
10  IF bTrigger=TRUE THEN // запись строки в таблицу по триггеру
11
12      arrTable[iIndex].sDateAndTime:=TIME_TO_TABLE(dtCurrentDateAndTime); // запись текущего времени, преобразованного...
13      // ...с помощью функции TIME_TO_TABLE
14      arrTable[iIndex].sValue:=REAL_TO_STRING(rValue); // запись текущего значения переменной
15      iIndex:=iIndex+1; // переход на следующую строку таблицы
16      bTrigger:=FALSE; // сброс триггера
17  END_IF
18
19  IF iIndex>10 THEN iIndex:=0; // после заполнения таблицы - переход на первую строку
20  END_IF // (запись в режиме циклического буфера)
```

Рис. 11.2.5.6. Код программы **PLC_PRG**

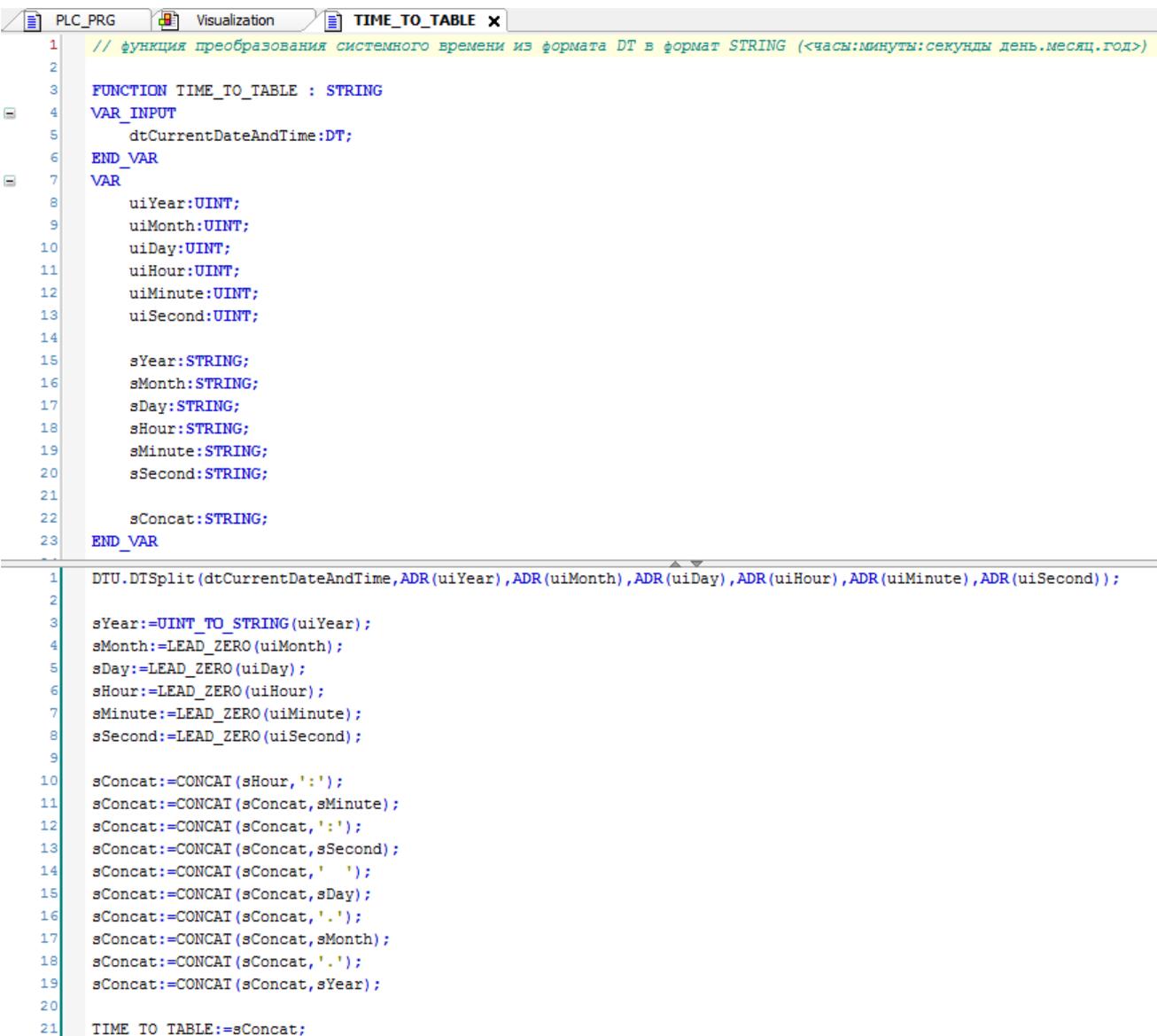
Работа с функциональным блоком **GetDateAndTime** (как и всей библиотекой **Time And Date**) описана в документе **СПК. Системное время**.

Если переменная **bTrigger** имеет значение **TRUE** (т.е. если пользователь нажал на кнопку записи), то в строку массива **arrTable**, которая представляет собой структуру **TableStruct**, записывается текущее значение системного времени, преобразованное с помощью функции **TIME_TO_TABLE**, и переменной **rValue**. После этого осуществляется увеличение индекса записываемой строки на 1 и сброс триггера записи.

Если индекс записываемой строки превышает 10, то осуществляется переход на нулевую строку (чтобы избежать переполнения массива).

Функция **TIME_TO_TABLE** используется для преобразования времени из формата **dt#2016-06-20-10:00:00** в формат **10:00:00 20.06.2016**.

Ее код выглядит следующим образом:



```
1 // функция преобразования системного времени из формата DT в формат STRING (<часы:минуты:секунды день.месяц.год>)
2
3 FUNCTION TIME_TO_TABLE : STRING
4 VAR_INPUT
5     dtCurrentDateAndTime:DT;
6 END_VAR
7 VAR
8     uiYear:UINT;
9     uiMonth:UINT;
10    uiDay:UINT;
11    uiHour:UINT;
12    uiMinute:UINT;
13    uiSecond:UINT;
14
15    sYear:STRING;
16    sMonth:STRING;
17    sDay:STRING;
18    sHour:STRING;
19    sMinute:STRING;
20    sSecond:STRING;
21
22    sConcat:STRING;
23 END_VAR
24
25 DTU.DTSplit(dtCurrentDateAndTime,ADR(uiYear),ADR(uiMonth),ADR(uiDay),ADR(uiHour),ADR(uiMinute),ADR(uiSecond));
26
27 sYear:=UINT_TO_STRING(uiYear);
28 sMonth:=LEAD_ZERO(uiMonth);
29 sDay:=LEAD_ZERO(uiDay);
30 sHour:=LEAD_ZERO(uiHour);
31 sMinute:=LEAD_ZERO(uiMinute);
32 sSecond:=LEAD_ZERO(uiSecond);
33
34 sConcat:=CONCAT(sHour,':');
35 sConcat:=CONCAT(sConcat,sMinute);
36 sConcat:=CONCAT(sConcat,':');
37 sConcat:=CONCAT(sConcat,sSecond);
38 sConcat:=CONCAT(sConcat,' ');
39 sConcat:=CONCAT(sConcat,sDay);
40 sConcat:=CONCAT(sConcat, '.');
41 sConcat:=CONCAT(sConcat,sMonth);
42 sConcat:=CONCAT(sConcat, '.');
43 sConcat:=CONCAT(sConcat,sYear);
44
45 TIME_TO_TABLE:=sConcat;
```

Рис. 11.2.5.7. Код функции **TIME_TO_TABLE**

Функция **LEAD_ZERO** используется для добавления ведущих нулей к разрядам времени.

```

PLC_PRG Visualization TIME_TO_TABLE LEAD_ZERO x
1 // функция добавления ведущего нуля к разрядам времени
2
3 FUNCTION LEAD_ZERO : STRING
4 VAR_INPUT
5     uiInput:UINT;
6 END_VAR
7 VAR
8 END_VAR
9
10 IF uiInput>9 THEN
11     LEAD_ZERO:=UINT_TO_STRING(uiInput);
12 ELSE LEAD_ZERO:=CONCAT('0',UINT_TO_STRING(uiInput));
13 END_IF

```

Рис. 11.2.5.8. Код функции **LEAD_ZERO**

7. Добавим в проект экран визуализации **Visualization**. В его свойствах выберем размер **800x480**. Экран будет содержать элемент Таблица, элемент Нажимной выключатель, два элемента Управление вращением, два элемента Индикатор и элемент Кнопка. Три текстовые надписи сделаны при помощи элементов Метка.

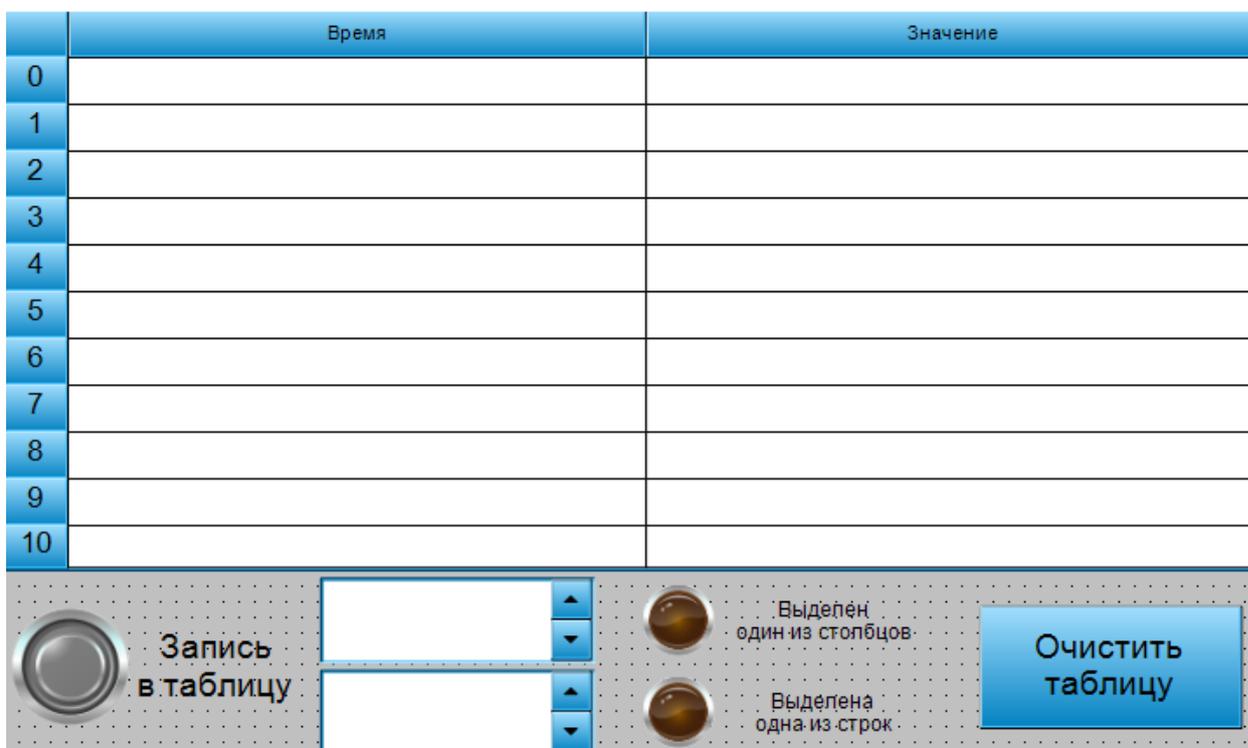


Рис. 11.2.5.9. Содержимое экрана **Visualization**

Настройки элементов описаны ниже.

8. Настроим элемент Таблица:

Свойство	Значения
Имя элемента	GenElemInst_24
Тип элемента	Таблица
Массив данных	PLC_PRG.arrTable
Макс. индекс массива	10
[-] Столбцы	
[-] Столбец	
[-] [0]	<input checked="" type="checkbox"/>
Заголовок столбца	Время
Ширина	368
Столбец изображения	<input type="checkbox"/>
+ Конфигурация изобра...	
Выравнивание текста...	По центру
Использовать шаблон	<input type="checkbox"/>
Выравнивание текста...	<input type="checkbox"/>
[-] [1]	<input checked="" type="checkbox"/>
Заголовок столбца	Значение
Ширина	389
Столбец изображения	<input type="checkbox"/>
+ Конфигурация изобра...	
Выравнивание текста...	По центру
Использовать шаблон	<input type="checkbox"/>
Выравнивание текста...	<input type="checkbox"/>
Показать заголовок строки	<input checked="" type="checkbox"/>
Показать заголовок столбца	<input checked="" type="checkbox"/>
Высота строки	30
Ширина строки заголовка	40
Размер полосы прокрутки	20
[-] Позиция	
X	0
Y	0
Ширина	800
Высота	360
[-] Свойства текста	
Горизонтальное выравнивание	По центру
Вертикальное выравнивание	По центру
Шрифт	Arial; 14
+ Цвет шрифта	 Font-Default-Color
+ Динамические тексты	
+ Переменные шрифта	
+ Переменные состояний	
[-] Выбор	
Цвет выбора	 Element-Alarm-Frame-Color
Тип выбора	Выбор ячеек
Фрейм вокруг выбранных яче...	<input checked="" type="checkbox"/>
Переменная для выбранного...	PLC_PRG.iColumnSelection
Переменная для выбранной с...	PLC_PRG.iLineSelection
Переменная для выбора дейс...	PLC_PRG.bColumnSelection
Переменная для выбора дейс...	PLC_PRG.bLineSelection

Рис. 11.2.5.10. Параметры элемента **Таблица**

9. К элементу [Нажимной выключатель](#) привяжем переменную **bTrigger**:

Свойство	Значения
Имя элемента	GenElemInst_26
Тип элемента	Нажимной выключатель
[-] Позиция	
X	5
Y	384
Ширина	89
Высота	77
Переменная	PLC_PRG.bTrigger
[+] Параметры изображения	
Поведение элемента	Переключатель изображения
[+] Тексты	
[+] Переменные состояний	
[-] Фон	
Изображение	Gray

Рис. 11.2.5.11. Параметры элемента **Нажимной выключатель**

10. Настроим элементы [Управление вращением](#):

Свойства	
̷ Фильтр ̣ Сортировать по ̡ Порядок сортировки ̣ Эксперт	
Свойство	Значения
Имя элемента	GenElemInst_40
Тип элемента	Управление вращением
[-] Позиция	
X	203
Y	366
Ширина	172
Высота	52
Переменная	PLC_PRG.iColumnSelection
Числовой формат	Выделен столбец №%d
Интервал	1
[-] Value range	
Минимальное значение	0
Максимальное значение	2

Рис. 11.2.5.12. Параметры элемента **Управление вращением 1**

Свойства	
Фильтр Сортировать по Порядок сортировки Эксперт	
Свойство	Значения
Имя элемента	GenElemInst_38
Тип элемента	Управление вращением
Позиция	
X	203
Y	425
Ширина	172
Высота	52
Переменная	PLC_PRG.iLineSelection
Числовой формат	Выделена строка №%d
Интервал	1
Value range	
Минимальное значение	0
Максимальное значение	10

Рис. 11.2.5.13. Параметры элемента **Управление вращением 2**

11. Настроим элементы [Индикатор](#):

Свойства	
Фильтр Сортировать по Порядок сортировки Эксперт	
Свойство	Значения
Имя элемента	GenElemInst_48
Тип элемента	Индикатор
Позиция	
X	407
Y	369
Ширина	47
Высота	47
Переменная	PLC_PRG.bColumnSelection
Параметры изображения	
Тексты	
Переменные состояний	
Фон	
Изображение	Yellow

Рис. 11.2.5.14. Параметры элемента **Индикатор (Выделен один из столбцов)**

Свойство	Значения
Имя элемента	GenElemInst_44
Тип элемента	Индикатор
Позиция	
X	407
Y	428
Ширина	47
Высота	47
Переменная	PLC_PRG.bLineSelection
Параметры изображения	
Тексты	
Переменные состояний	
Фон	
Изображение	Yellow

Рис. 11.2.5.15. Параметры элемента **Индикатор** (Выделена одна из строк)

12. К элементу [Кнопка](#) (**Очистить таблицу**) во вкладке [InputConfiguration](#) к параметру **OnClick** привяжем действие [Выполнить ST-код](#):

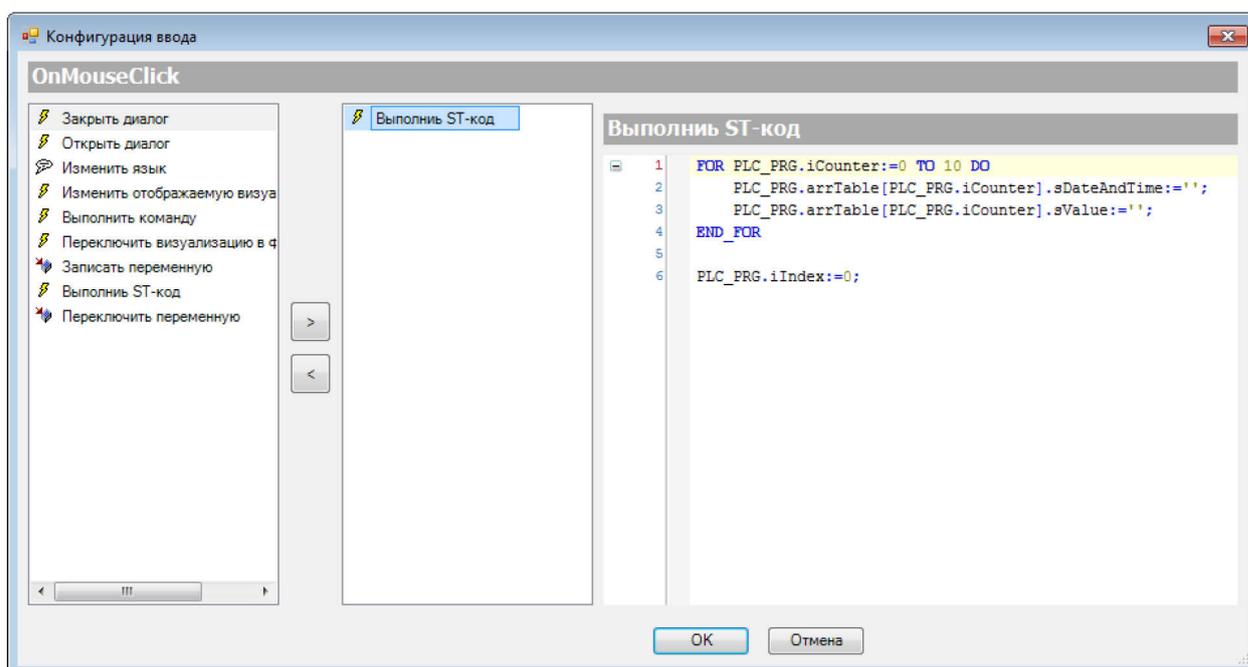
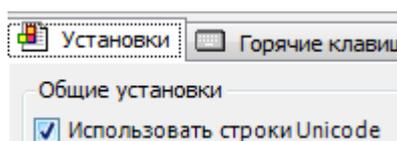


Рис. 11.2.5.16. ST-код элемента **Кнопка** (**Очистить таблицу**)

13. Настроим [Менеджер визуализации](#). В его установках поставим галочку **Использовать строки Unicode**:



Настройки **target**- и **web**-визуализации будут выглядеть следующим образом:

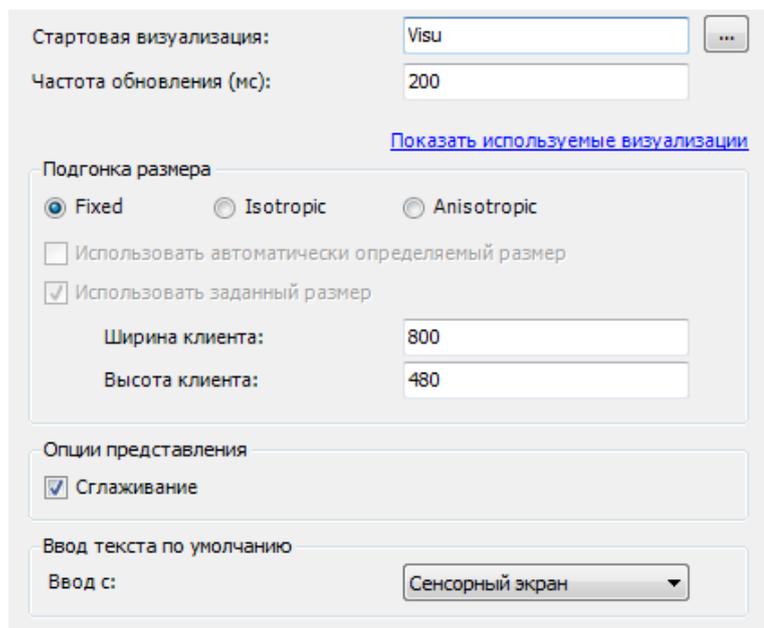


Рис. 11.2.5.17. Настройки **target**-визуализации

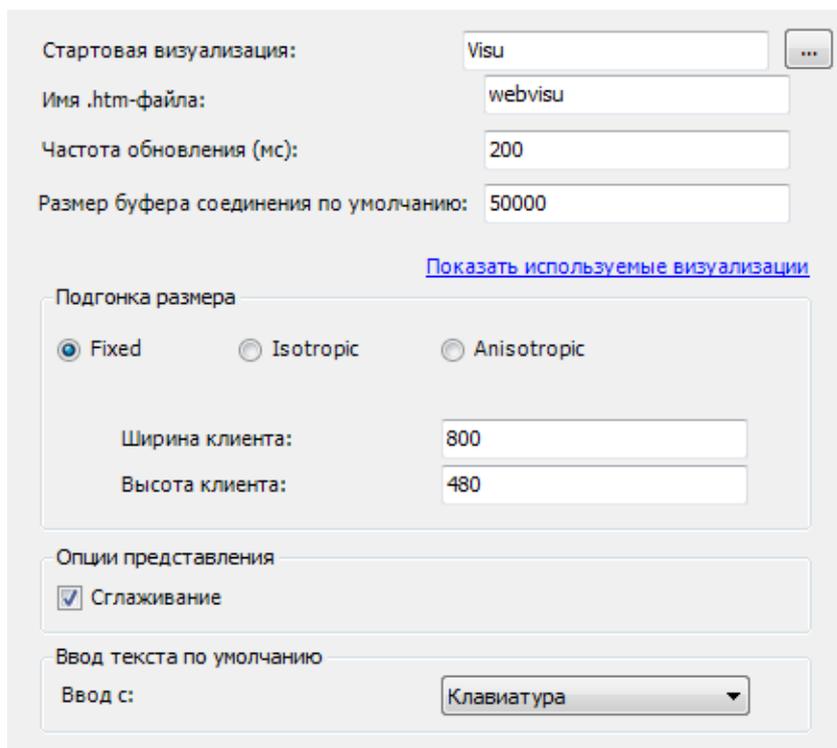


Рис. 11.2.5.18. Настройки **web**-визуализации

14. [Запустим проект на виртуальном контроллере](#). Откроем в веб-браузере страницу <http://localhost:8080/webvisu.htm>, чтобы увидеть web-визуализацию проекта. **Обратите внимание**, что браузер должен поддерживать **HTML5**. Проверим функционал проекта.

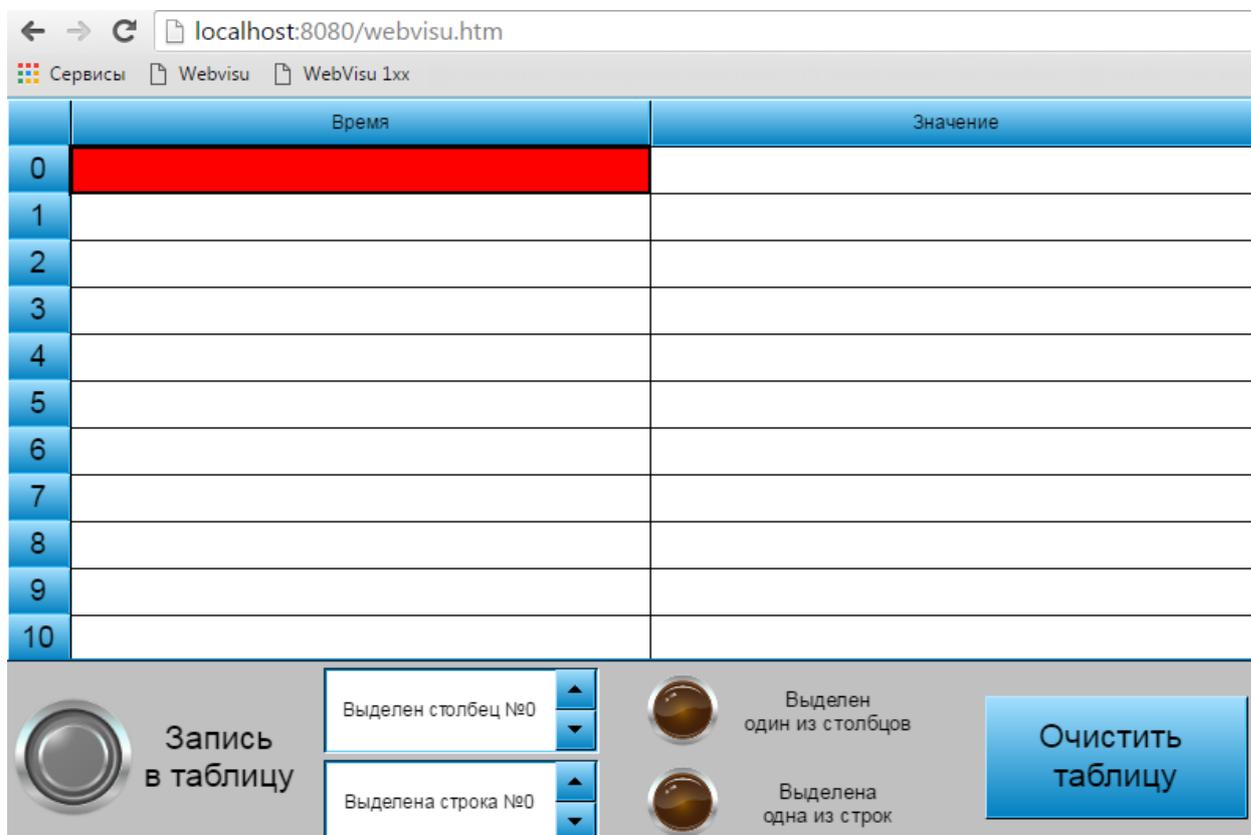


Рис. 11.2.5.19. Web-визуализация проекта в браузере **Chrome**

По умолчанию выделена нулевая строка нулевого столбца.

По нажатию на выключатель в строку таблицы записывается текущее время и значение переменной:

	Время	Значение
0	08:01:44 27.06.2016	196.0028
1	08:01:48 27.06.2016	215.804
2	08:01:49 27.06.2016	223.5045
3	08:01:58 27.06.2016	267.8071
4	08:02:01 27.06.2016	281.808

Рис. 11.2.5.20. Отображение структуры в таблице

Нажмите на десятую ячейку нулевого столбца таблицы. В элементах **Управление вращением** отобразятся номера строки и столбца ячейки, а **Индикаторы** выделения загорятся:

9	08:03:38 27.06.2016	771.7586
10	08:03:41 27.06.2016	786.2551

Запись в таблицу

Выделен столбец №0

Выделена строка №10

Выделен один из столбцов

Выделена одна из строк

Очистить таблицу

Рис. 11.2.5.21. Информация о выделенной ячейке таблицы

Нажмите на верхнюю левую клетку таблицу (пустую), чтобы сбросить выделение.

Нажмите на кнопку **Очистить таблицу**, чтобы стереть отображаемые в ней значения.

11.2.6. Отображение линейки и Потенциометр

Данный пример посвящен работе с графическими примитивами [Отображение линейки](#) (используется для отображения значения привязанной численной переменной) и [Потенциометр](#) (используется для изменения значения привязанной численной переменной).

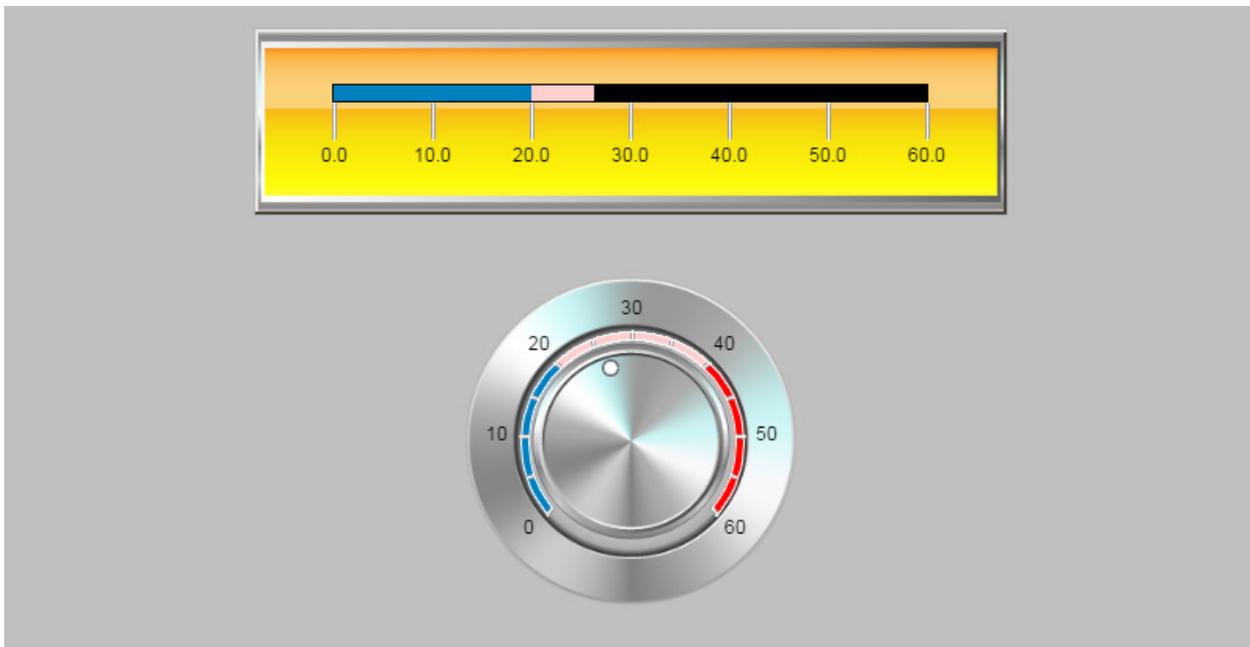


Рис. 11.2.6.1. Внешний вид примера **Отображение линейки** и **Потенциометр**

Пример создан в среде **CODESYS 3.5 SP6 Patch 2** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с target-файлом **3.5.6.0**.

Пример доступен для скачивания: [Example_BarDisplay.projectarchive](#)

1. Создадим новый *стандартный* проект CODESYS с названием **Example_BarDisplay** и настройками по умолчанию: target – **CODESYS Control Win V3**, язык PLC_PRG – **ST**.

2. В программе **PLC_PRG** объявим переменную **rValue** типа **REAL**:

```
1 | PROGRAM PLC_PRG
2 | VAR
3 |     rValue:REAL;
4 | END_VAR
```

Рис. 11.2.6.2. Объявление переменных программы **PLC_PRG**

3. Добавим в проект экран визуализации Visualization. В его свойствах выберем размер **800x480**. Экран будет содержать элемент Отображение линейки и элемент Потенциометр.

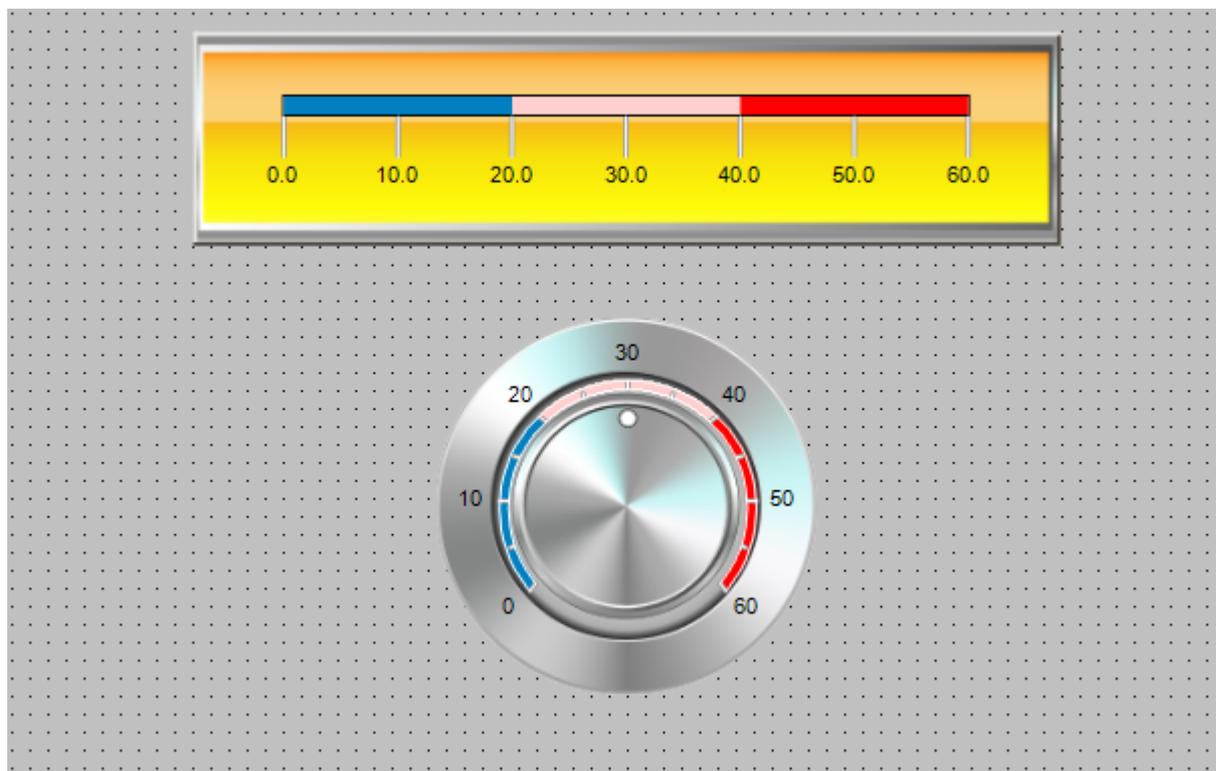


Рис. 11.2.6.3. Содержание экрана **Visualization**

Настройки элементов приведены ниже.

4. Настроим элемент Отображение линейки:

Свойства	
Фильтр Сортировать по Порядок сортировки Эксперт	
Свойство	Значения
Имя элемента	GenElemInst_58
Тип элемента	Отображение линейки
Значение	PLC_PRG.rValue
Позиция	
X	160
Y	20
Ширина	480
Высота	120
Фон	
Цвет изображения	Gray
Собственное изображение	
Изображение	__VisualizationStyle.Element-BarDisplayImage-Yellow-Horizontal
Цвет прозрачности	0; 0; 0
Оптимизация нанесения	<input checked="" type="checkbox"/>
Линейка	
Тип диаграммы	Шкала возле линейки
Ориентация	Горизонтально
Направление движения	Слева направо
Шкала	
Начало шкалы	0
Конец шкалы	60
Основная шкала	10
Подшкала	0
Ширина линии шкалы	1
Цвет шкалы	<input type="text" value="White"/>
Шкала в 3D	<input checked="" type="checkbox"/>
Фрейм элемента	<input checked="" type="checkbox"/>
Метка	
Единица	
Шрифт	Font-Standard
Формат шкалы (с-синтаксис)	%.1f
Макс. ширина текста меток	31
Высота текста меток	15
Цвет шрифта	Font-Default-Color
Расположение	
Горизонтальный сдвиг	0
Вертикальный сдвиг	0
Horizontal scaling	5
Vertical Scaling	0
Цвета	
Цвет графика	<input type="text" value="White"/>
Фон линейки	<input checked="" type="checkbox"/>
Цвет фрейма	<input type="text" value="212; 208; 200"/>
Переключить весь цвет	<input type="checkbox"/>
Использовать градиентный цвет для лин...	<input type="checkbox"/>
Маркеры цветовых диапазонов	Нет маркеров
Цветовые области	<input type="button" value="Создать новый"/>
Области	
[0]	<input type="button" value="Удалить"/>
Начало области	0
Конец области	20
Цвет	Element-Control-Color
[1]	<input type="button" value="Удалить"/>
Начало области	20
Конец области	40
Цвет	Element-Alarm-Fill-Color
[2]	<input type="button" value="Удалить"/>
Начало области	40
Конец области	60
Цвет	Element-Alarm-Frame-Color

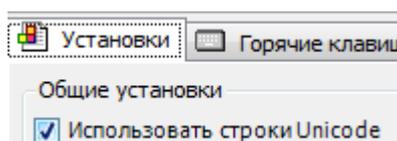
Рис. 11.2.6.4. Параметры элемента **Отображение линейки**

5. Настроим элемент [Потенциометр](#):

Свойства	
Фильтр Сортировать по Порядок сортировки Эксперт	
Свойство	Значения
Имя элемента	GenElemInst_70
Тип элемента	Потенциометр
Переменная	PLC_PRG.rValue
[-] Позиция	
X	295
Y	180
Ширина	210
Высота	210
[-] Фон	
Цвет изображения	Gray
[-] Собственное изображение	
Изображение	<default>
Цвет прозрачности	0; 0; 0
Оптимизация нанесения	<input checked="" type="checkbox"/>
[-] Стрелка	
Тип стрелки	Круг
Цвет	255; 255; 255
Начало стрелки	220
Конец стрелки	-40
[-] Шкала	
Позиция подшкалы	Внутри
Тип шкалы	Линии
Начало шкалы	0
Конец шкалы	60
Основная шкала	10
Подшкала	5
Ширина линии шкалы	1
Цвет шкалы	Element-Potentiometer-ScaleColor
Шкала в 3D	<input checked="" type="checkbox"/>
Показать шкалу	<input checked="" type="checkbox"/>
Фрейм внутри	<input checked="" type="checkbox"/>
Фрейм снаружи	<input checked="" type="checkbox"/>
[-] Метка	
Метка	Снаружи
Единица	
Шрифт	Font-Standard
Формат шкалы (с-синтаксис)	%d
Макс. ширина текста меток	28
Высота текста меток	15
Цвет шрифта	Element-Potentiometer-LabelFontColor
[-] Расположение	
[-] Цвета	
[-] Цветовые области	
Долговременные цветовые обл...	<input checked="" type="checkbox"/>
[-] Цветовые области	
[-] Области	
[-] [0]	
Начало области	0
Конец области	20
Цвет	Element-Control-Color
[-] [1]	
Начало области	20
Конец области	40
Цвет	BasicElement-Alarm-Fill-Color
[-] [2]	
Начало области	40
Конец области	60
Цвет	Element-Alarm-Frame-Color

Рис. 11.2.6.5. Параметры элемента **Потенциометр**

6. Настроим компонент [Менеджер визуализации](#). В его установках поставим галочку **Использовать строки Unicode**:



Настройки **target**- и **web**-визуализации будут выглядеть следующим образом:

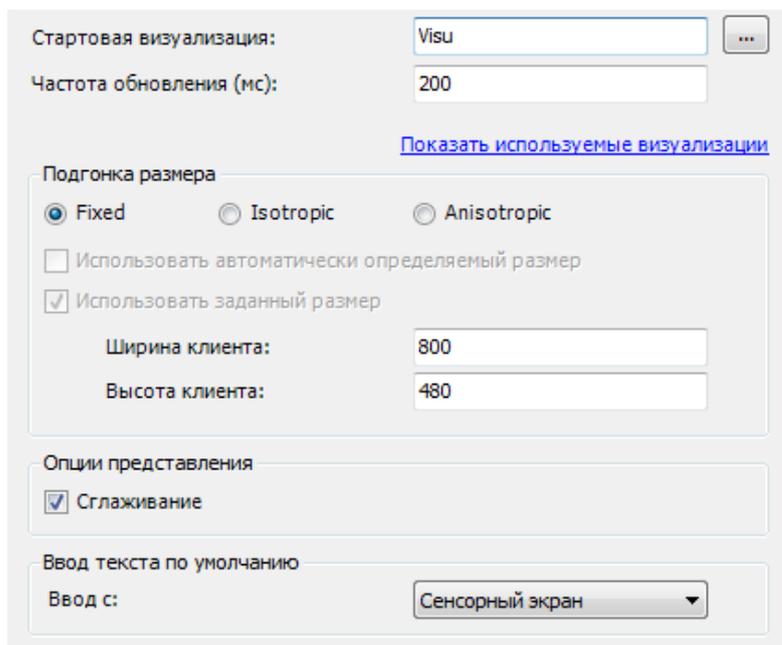


Рис. 11.2.6.6. Настройки **target**-визуализации

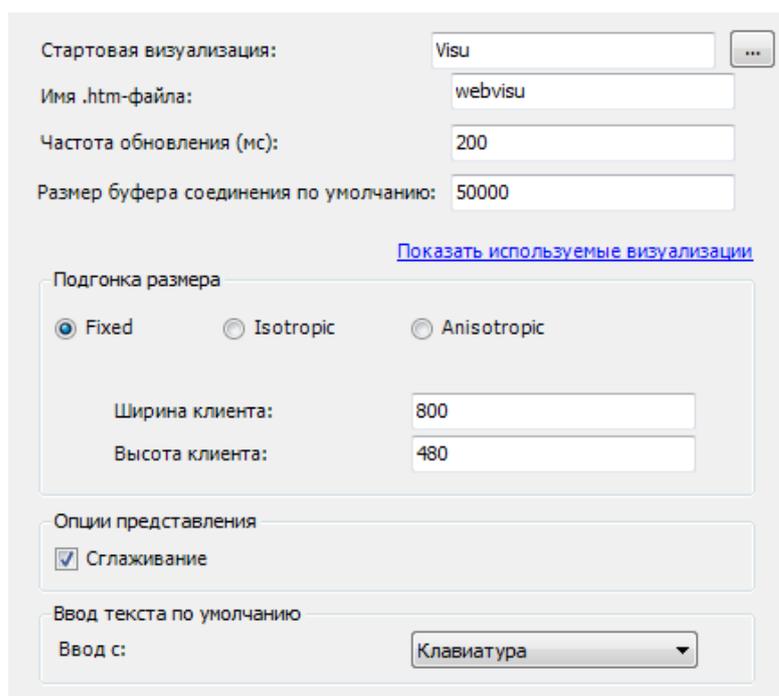


Рис. 11.2.6.7. Настройки **web**-визуализации

7. [Запустим проект на виртуальном контроллере](#). Откроем в веб-браузере страницу <http://localhost:8080/webvisu.htm>, чтобы увидеть web-визуализацию проекта. **Обратите внимание**, что браузер должен поддерживать **HTML5**. Проверим функционал проекта.

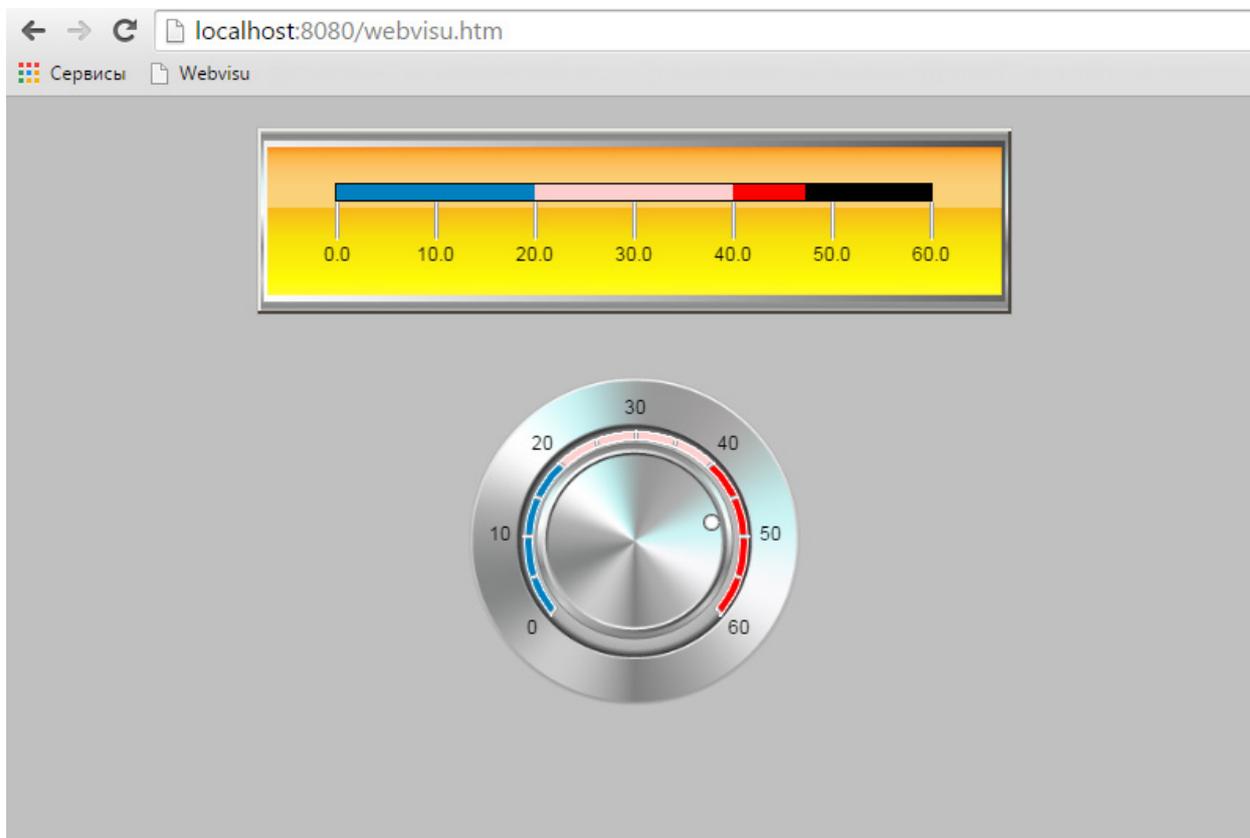


Рис. 11.2.6.8. Web-визуализация проекта в браузере **Chrome**

Меняя значение переменной с помощью элемента **Потенциометр** (доступно как нажатие на шкалу элемента, так и перетаскивание стрелки), можно наблюдать соответствующие изменения в элементе **Отображение линейки**.

11.2.7. Гистограмма

Данный пример посвящен работе с графическим примитивом [Гистограмма](#), используемым для отображения в графическом виде одномерных массивов значений.

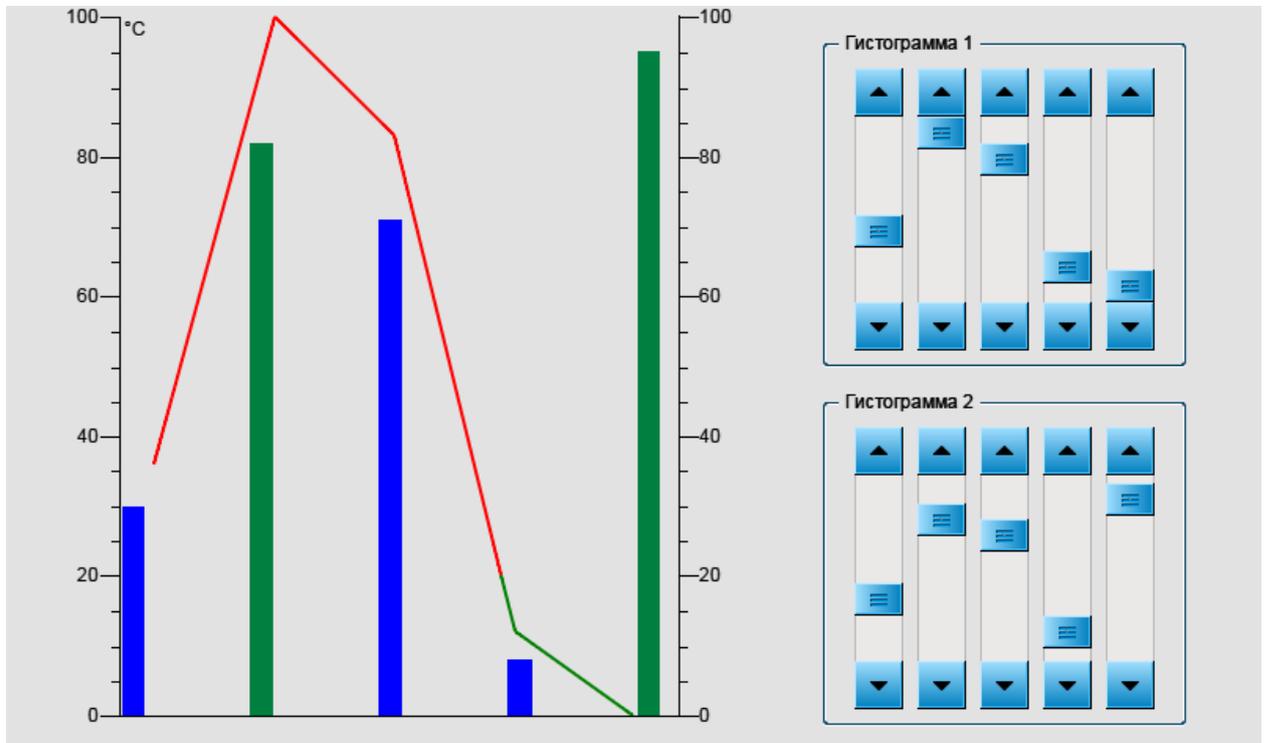


Рис. 11.2.7.1. Внешний вид примера **Гистограмма**

Пример создан в среде **CODESYS 3.5 SP6 Patch 2** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с target-файлом **3.5.6.0**.

Пример доступен для скачивания: [Example_Histogram.projectarchive](#)

1. Создадим новый **стандартный** проект **CODESYS** с названием **Example_Histogram** и настройками по умолчанию: target – **CODESYS Control Win V3**, язык PLC_PRG – **ST**.

2. В программе **PLC_PRG** объявим переменную **rValue** типа **REAL**:

```
1 PROGRAM PLC_PRG
2 VAR
3   arrHist1:ARRAY [0..4] OF INT;
4   arrHist2:ARRAY [0..4] OF INT;
5 END_VAR
```

Рис. 11.2.7.2. Объявление переменных программы **PLC_PRG**

3. **Добавим в проект** экран визуализации **Visualization**. В его **свойствах** выберем размер **800x480**. Экран будет содержать два элемента **Гистограмма** (наложенных друг на друга), два элемента **Группа** и десять элементов **Полоса прокрутки**.

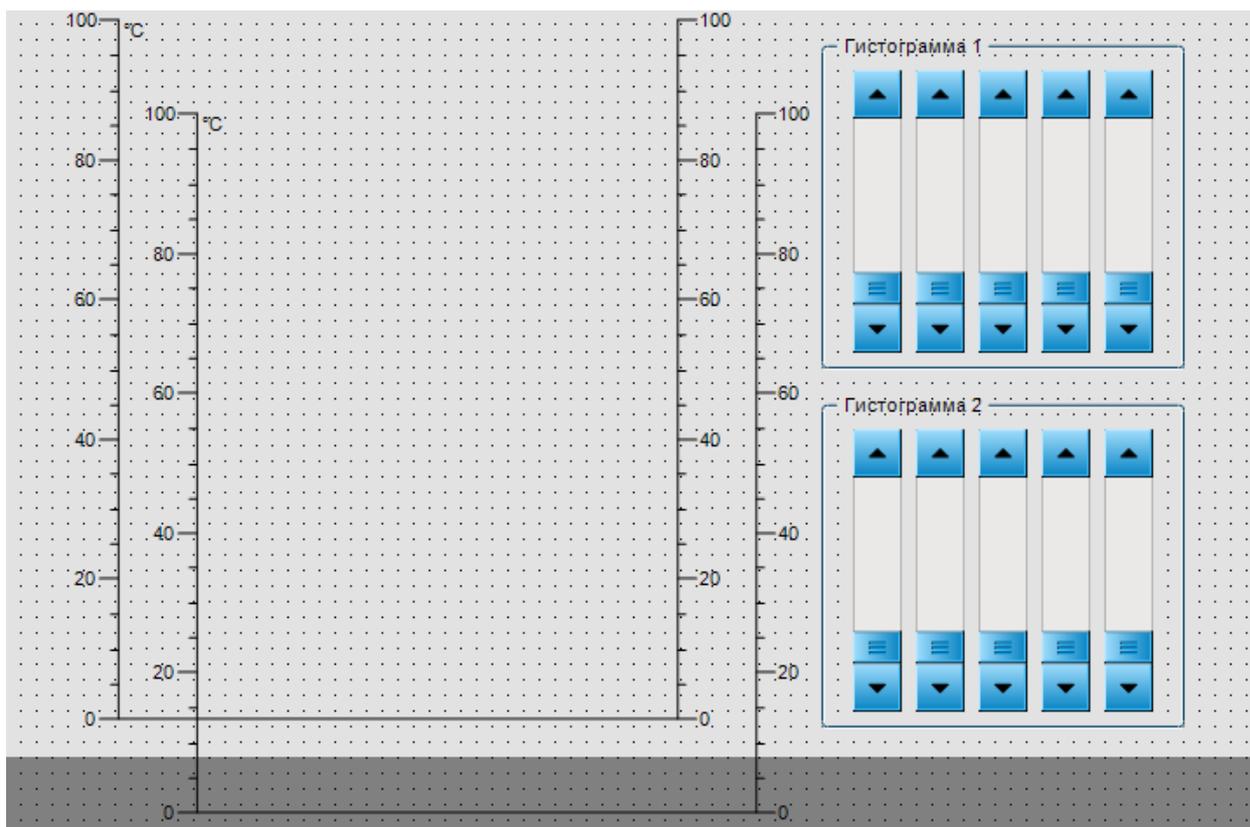


Рис. 11.2.7.3. Содержание экрана **Visualization** (гистограммы еще не наложены друг на друга)

Настройки элементов приведены ниже.

4. Настроим элементы [Гистограмма 1](#) и Гистограмма 2:

Свойства	
У Фильтр ▾ ↕ Сортировать по ▾ A Z ↓ Порядок сортировки ▾	
<input checked="" type="checkbox"/> Эксперт	
Свойство	Значения
Тип элемента	Гистограмма
Массив данных	PLC_PRG.arrHist1
⊕ Поддиапазон массива	
Тип отображения	Кривая
Ширина линии	3
Показывать горизонтальные линии	<input type="checkbox"/>
Относительная ширина линейки	60
⊖ Позиция	
X	10
Y	0
Ширина	480
Высота	470
⊖ Шкала	
Начало шкалы	0
Конец шкалы	100
Основная шкала	20
Подшкала	5
Цвет шкалы	█ Black
Base line	0
⊖ Метка	
Единица	°C
Шрифт	Font-Standard
Формат шкалы (с-синтаксис)	%d
Макс. ширина текста меток	50
Высота текста меток	15
Цвет шрифта	█ Font-Default-Color
⊖ Цвета	
Цвет графика	█ 255; 0; 0
⊖ Цвет тревоги	
Значение тревоги	20
Условие тревоги	Меньше
Цвет тревоги	█ 0; 128; 0
Использовать цветовые облас...	<input checked="" type="checkbox"/>
⊕ Цветовые области	█ Создать новый

Рис. 11.2.7.4. Параметры элемента **Гистограмма 1**

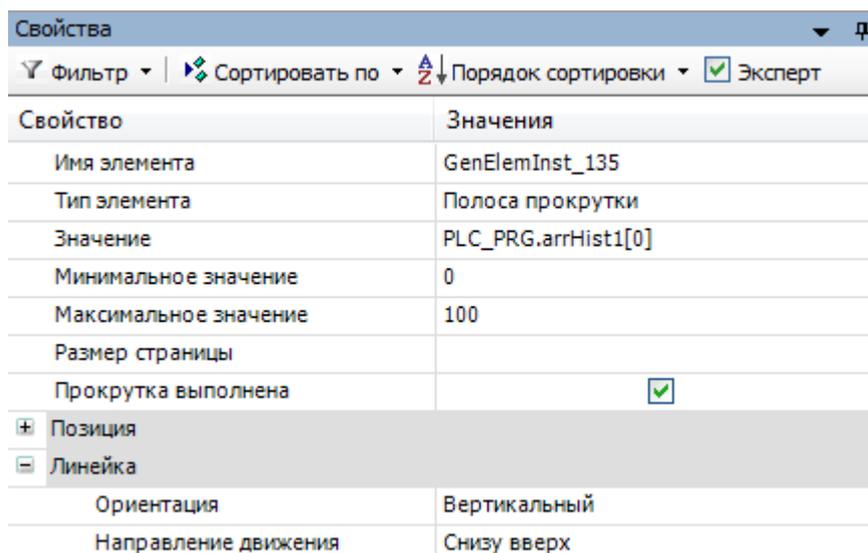
Вторая гистограмма отличается от первой только привязываемой переменной, типом отображения и условием тревоги:

Свойства	
̄ Фильтр ̄ Сортировать по ̄ Порядок сортировки	
<input checked="" type="checkbox"/> Эксперт	
Свойство	Значения
Имя элемента	GenElemInst_137
Тип элемента	Гистограмма
Массив данных	PLC_PRG.arrHist2
+ Поддиапазон массива	
Тип отображения	Линейки
Показывать горизонтальные линии	<input type="checkbox"/>
Относительная ширина линейки	20
- Позиция	
X	60
Y	60
Ширина	480
Высота	470
- Шкала	
Начало шкалы	0
Конец шкалы	100
Основная шкала	20
Подшкала	5
Цвет шкалы	█ Black
Base line	0
- Метка	
Единица	°C
Шрифт	Font-Standard
Формат шкалы (с-синтаксис)	%d
Макс. ширина текста меток	50
Высота текста меток	15
Цвет шрифта	█ Font-Default-Color
- Цвета	
Цвет графика	█ 0; 0; 255
- Цвет тревоги	
Значение тревоги	80
Условие тревоги	Больше
Цвет тревоги	█ 0; 128; 64
Использовать цветовые облас...	<input checked="" type="checkbox"/>
+ Цветовые области	Создать новый

Рис. 11.2.7.5. Параметры элемента **Гистограмма 2**

После окончания настройки элементов, наложим их друг на друга.

4. Настроим элементы [Полоса прокрутки](#):



Свойство	Значения
Имя элемента	GenElemInst_135
Тип элемента	Полоса прокрутки
Значение	PLC_PRG.arrHist1[0]
Минимальное значение	0
Максимальное значение	100
Размер страницы	
Прокрутка выполнена	<input checked="" type="checkbox"/>
Позиция	
Линейка	
Ориентация	Вертикальный
Направление движения	Снизу вверх

Рис. 11.2.7.6. Параметры элемента **Полоса прокрутки 1** (группа **Гистограмма 1**)

Остальные элементы будут отличаться только привязываемыми переменными.

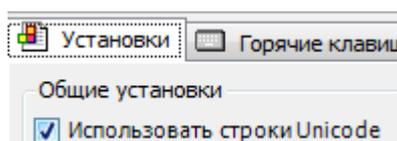
Для группы **Гистограмма 1** это, соответственно, будут:

```
PLC_PRG.arrHist1[0]  
PLC_PRG.arrHist1[1]  
PLC_PRG.arrHist1[2]  
PLC_PRG.arrHist1[3]  
PLC_PRG.arrHist1[4]
```

а для группы **Гистограмма 2**:

```
PLC_PRG.arrHist2[0]  
PLC_PRG.arrHist2[1]  
PLC_PRG.arrHist2[2]  
PLC_PRG.arrHist2[3]  
PLC_PRG.arrHist2[4]
```

5. Настроим компонент [Менеджер визуализации](#). В его установках поставим галочку **Использовать строки Unicode**:



Настройки **target**- и **web**-визуализации будут выглядеть следующим образом:

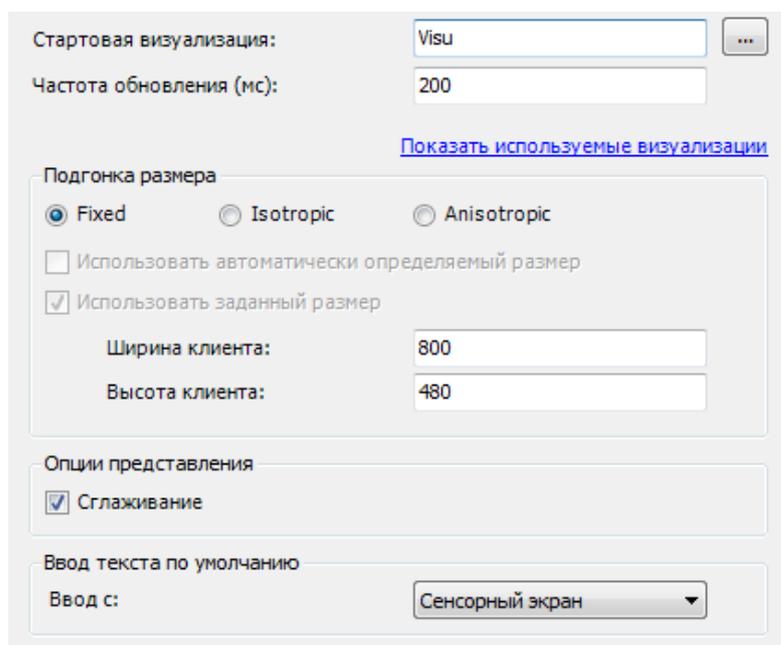


Рис. 11.2.7.7. Настройки **target**-визуализации

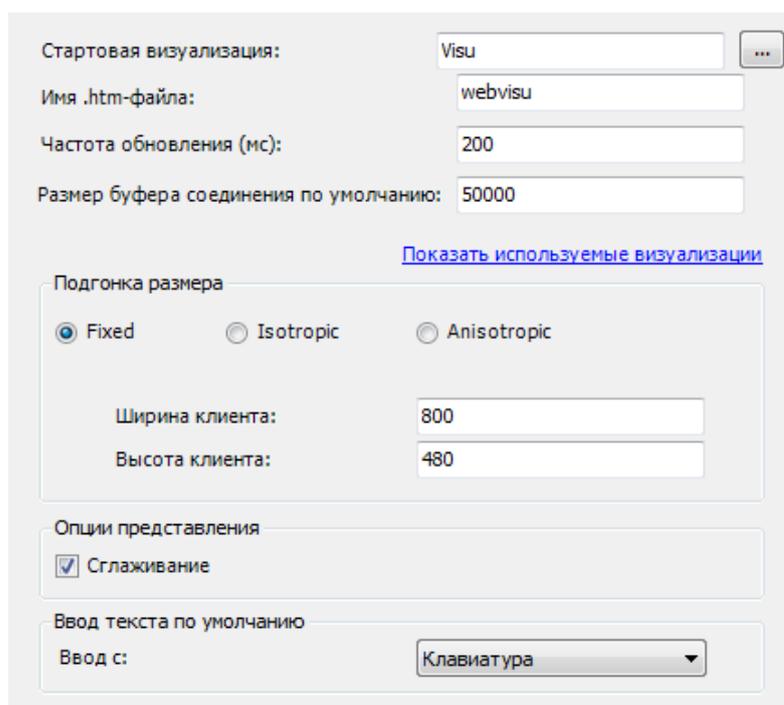


Рис. 11.2.7.8. Настройки **web**-визуализации

8. [Запустим проект на виртуальном контроллере](#). Откроем в веб-браузере страницу <http://localhost:8080/webvisu.htm>, чтобы увидеть web-визуализацию проекта. **Обратите внимание**, что браузер должен поддерживать **HTML5**. Проверим функционал проекта.

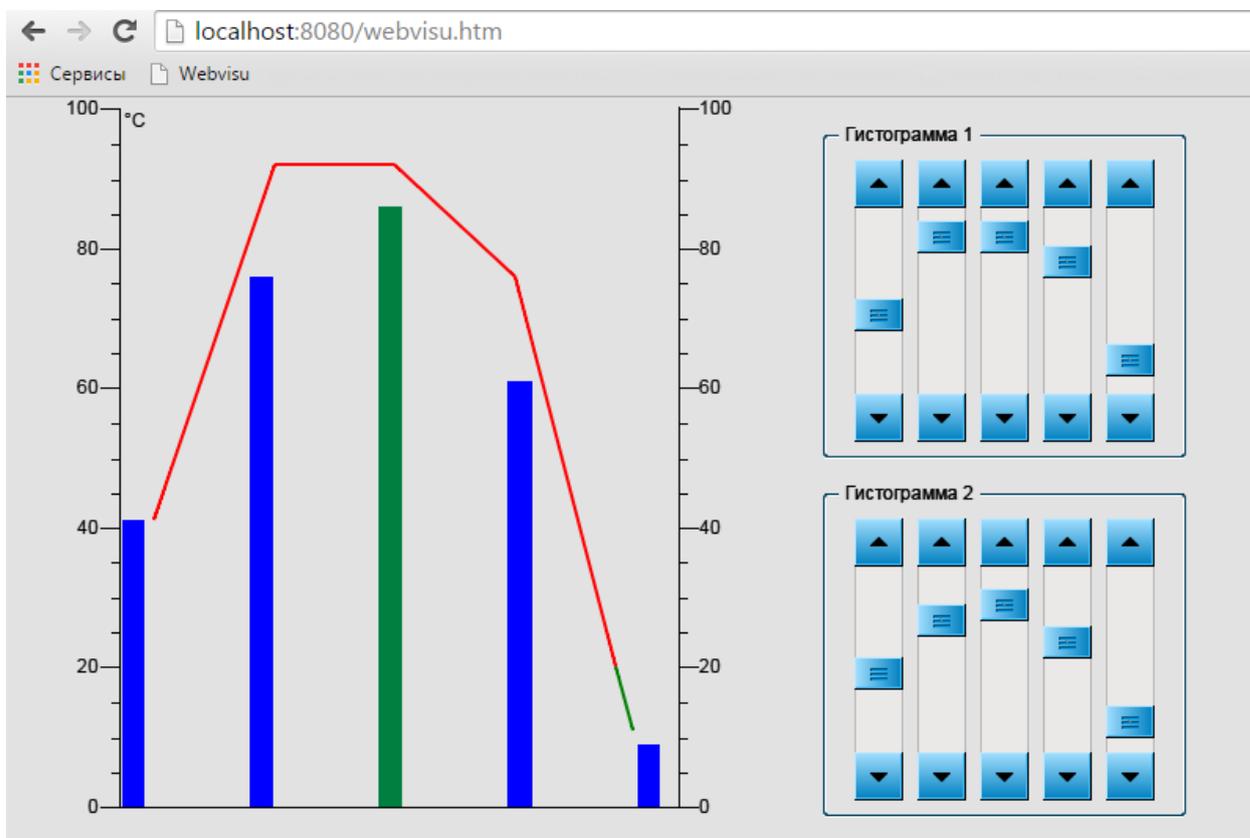


Рис. 11.2.7.9. Web-визуализация проекта в браузере Chrome

Меняя значение элементов массива с помощью элементов **Полоса прокрутки**, можно наблюдать соответствующие изменения в элементе **Гистограмма**.

Для **Гистограммы 1** значения, превышающие 80, являются тревожными – в этом случае столбец окрашивается в зеленый цвет.

Для **Гистограммы 2** тревожными являются значения менее 20-ти – в этом случае линия графика окрашивается в зеленый.

11.2.8. Индикаторы/Переключатели/Изображения

Данный пример посвящен работе с элементами [Индикатор](#), [Переключатель](#) и [Переключатель изображения](#), предназначенными для отображения и переключения переменных типа **BOOL**.



Рис. 11.2.8.1. Внешний вид примера **Индикаторы/Переключатели/Изображения**

Пример создан в среде **CODESYS 3.5 SP6 Patch 2** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с target-файлом **3.5.6.0**.

Пример доступен для скачивания: [Example Lamps.projectarchive](#)

1. Создадим новый *стандартный* проект **CODESYS** с названием **Example_Lamps** и настройками по умолчанию: target – **CODESYS Control Win V3**, язык PLC_PRG – **ST**.

2. В программе **PLC_PRG** объявим следующие переменные:

```
1 PROGRAM PLC_PRG
2 VAR
3     bVar1:BOOL;
4     bVar2:BOOL;
5     bVar3:BOOL;
6     bVar4:BOOL;
7     bVar5:BOOL;
8     bVar6:BOOL;
9 END_VAR
```

Рис. 11.2.8.2. Объявление переменных программы **PLC_PRG**

3. Добавим в проект экран визуализации **Visualization**. В его свойствах выберем размер **800x480**. Экран будет содержать пять элементов Индикатор, один элемент Переключатель изображения и шесть элементов Переключатель/Выключатель (разных типов).

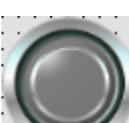


Рис. 11.2.8.3. Содержание экрана **Visualization**

Настройки элементов приведены ниже.

4. Настроим элементы [Индикаторы](#) и [Переключатели](#) согласно табл. 44:

Табл. 44. Параметры элементов **Индикаторы** и **Переключатели/Выключатели**

№ пп.	Тип элемента	Внешний вид элемента	Переменная	Фон
1.	Индикатор		PLC_PRG.bVar1	Yellow
2.	Индикатор		PLC_PRG.bVar2	Red
3.	Индикатор		PLC_PRG.bVar3	Green
4.	Индикатор		PLC_PRG.bVar4	Blue
5.	Индикатор		PLC_PRG.bVar5	Gray
6.	Переключатель		PLC_PRG.bVar1	Yellow
7.	Переключатель питания		PLC_PRG.bVar2	Red
8.	Нажимной выключатель		PLC_PRG.bVar3	Green

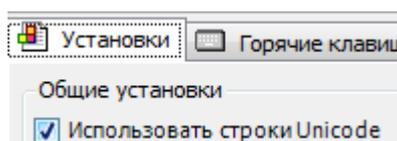
9.	Нажимной выключатель LED		PLC_PRG.bVar4	Blue
10.	Клавишный выключатель		PLC_PRG.bVar5	Gray
11.	Вращающийся выключатель		PLC_PRG.bVar6	Gray

5. Настроим элемент [Переключатель изображения](#):

Свойства	
̣ Фильтр ̣ Сортировать по ̣ Порядок сортировки ̣ Эксперт	
Свойство	Значения
Имя элемента	GenElemInst_215
Тип элемента	Переключатель изображения
[-] Позиция	
X	650
Y	90
Ширина	85
Высота	85
Переменная	PLC_PRG.bVar6
[-] Параметры изображения	
Изображение вкл.	__VisualizationStyle.Element-Lamp-Lamp1-Green-On
Изображение выкл.	__VisualizationStyle.Element-Lamp-Lamp1-Red-On
Image pushed	__VisualizationStyle.Element-Lamp-Lamp1-Blue-On
Прозрачный	<input type="checkbox"/>
Прозрачный цвет	 Black
Изотропный тип	Изотропия
Горизонтальное выравнивание	Лево
Вертикальное выравнивание	Верх
Поведение элемента	Переключатель изображения
[-] Тексты	
Подсказка	
[-] Переменные состояний	
Невидимый	
Отключение ввода	

Рис. 11.2.8.4. Параметры элемента **Переключатель изображения**

6. Настроим компонент [Менеджер визуализации](#). В его установках поставим галочку **Использовать строки Unicode**:



Настройки **target**- и **web**-визуализации будут выглядеть следующим образом:

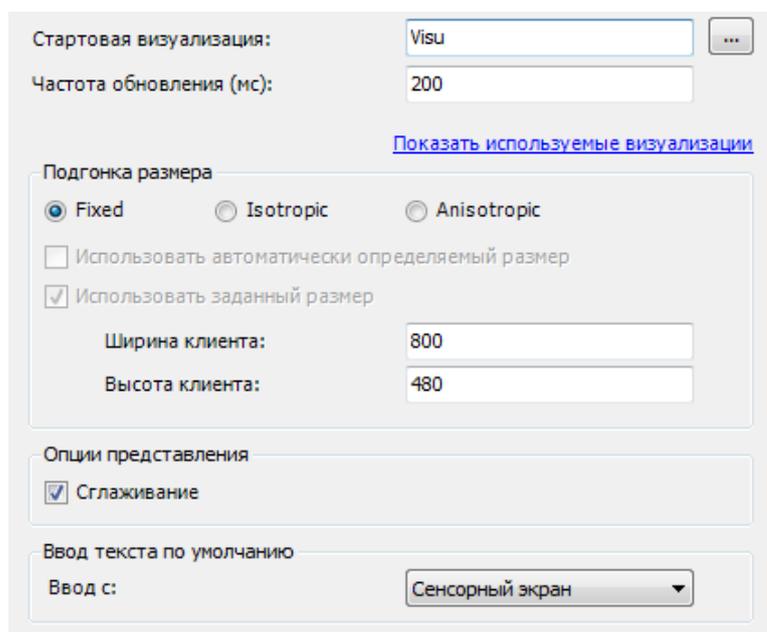


Рис. 11.2.8.5. Настройки **target**-визуализации

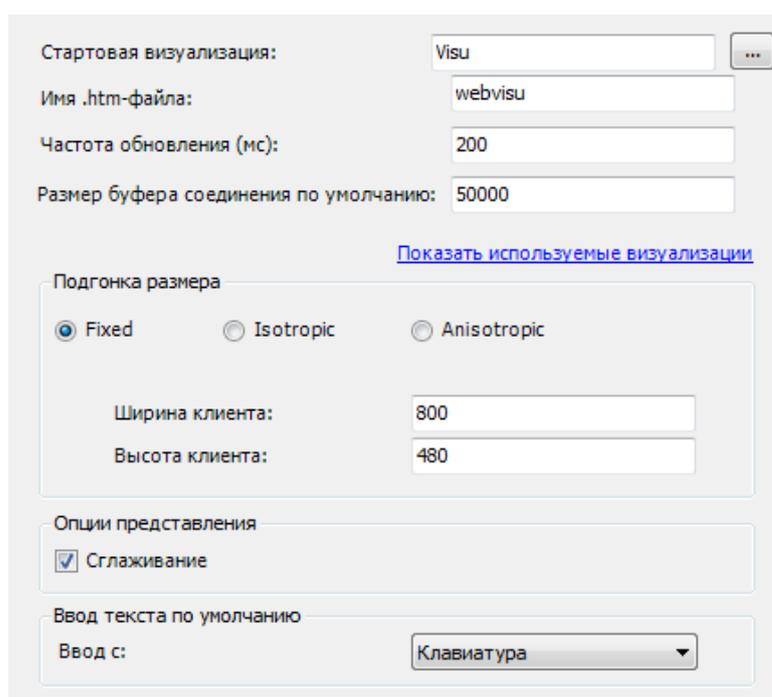


Рис. 11.2.8.6. Настройки **web**-визуализации

7. [Запустим проект на виртуальном контроллере](#). Откроем в веб-браузере страницу <http://localhost:8080/webvisu.htm>, чтобы увидеть web-визуализацию проекта. **Обратите внимание**, что браузер должен поддерживать **HTML5**. Проверим функционал проекта.



Рис. 11.2.8.7. Web-визуализация проекта в браузере **Chrome**

Нажатие на переключатели будет включать/отключать соответствующие лампы.

Лампа, сделанная на основе элемента **Переключатель изображения**, сама доступна для нажатия. При этом на время зажатия она принимает третье («промежуточное») изображение.

11.2.9. Трассировка

Данный пример посвящен работе с элементом [Трассировка](#), который используется для построения графика значения переменной в реальном времени.

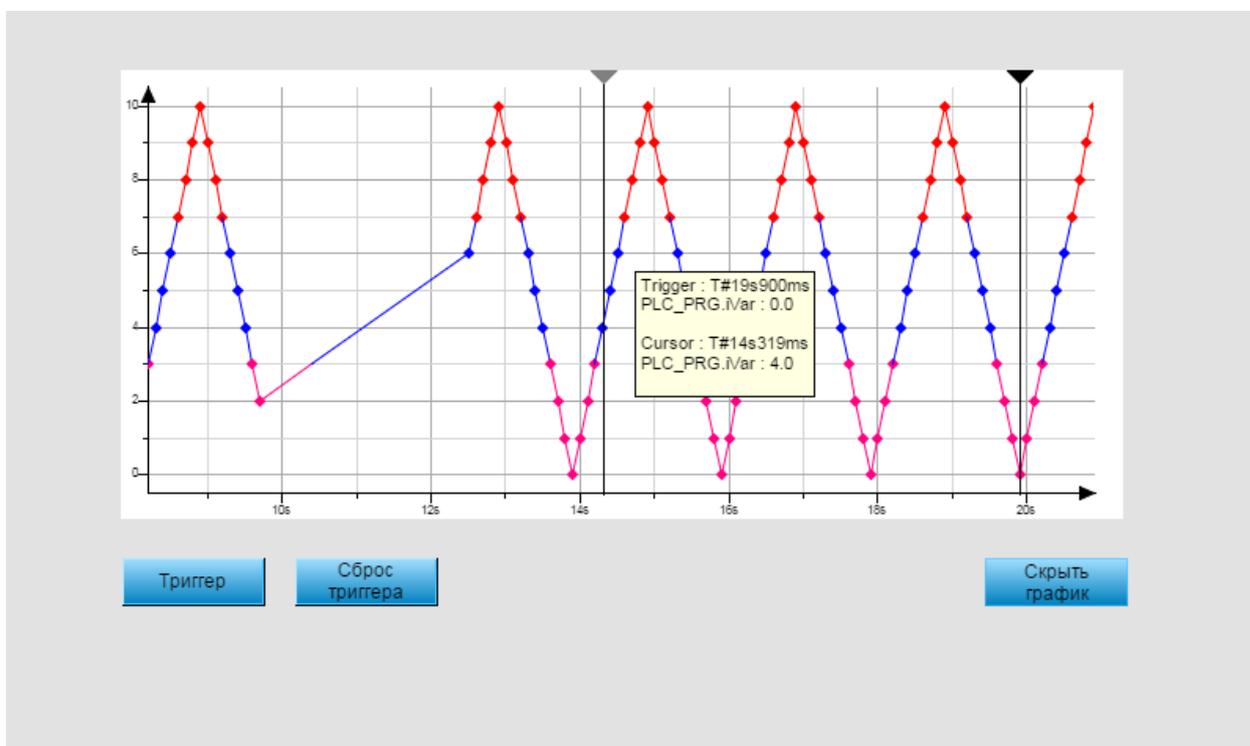


Рис. 11.2.9.1. Внешний вид примера Трассировка

Пример создан в среде **CODESYS 3.5 SP6 Patch 2** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с target-файлом **3.5.6.0**.

Пример доступен для скачивания: [Example Trace.projectarchive](#)

1. Создадим новый *стандартный* проект **CODESYS** с названием **Example_Trace** и настройками по умолчанию: target – **CODESYS Control Win V3**, язык PLC_PRG – **ST**.

2. В программе **PLC_PRG** объявим следующие переменные:

```
1 PROGRAM PLC_PRG
2 VAR
3     bVar:BOOL;           // переменная для изменения значений переменной iVar
4     iVar:INT;           // переменная, отображаемая трассировкой
5
6     bTrigger:BOOL;      // триггер остановки трассировки
7     bResetTrigger:BOOL; // переменная сброса триггера
8     bVisible:BOOL:=TRUE; // переменная видимости графика
9 END_VAR
```

Рис. 11.2.9.2. Объявление переменных программы **PLC_PRG**

3. Код программы **PLC_PRG** будет выглядеть следующим образом:

```
1 // эмуляция изменения значений
2
3 IF bVar=FALSE THEN iVar:=iVar+1;
4     ELSE iVar:=iVar-1;
5 END_IF
6
7 IF          iVar=0 THEN bVar:=FALSE;
8     ELSIF  iVar=10 THEN bVar:=TRUE;
9 END_IF
10
```

Рис. 11.2.9.3. Код программы **PLC_PRG**

4. [Добавим в проект](#) экран визуализации **Visualization**. В его [свойствах](#) выберем размер **800x480**. Экран будет содержать элемент [Трассировка](#) и три элемента [Кнопка](#).

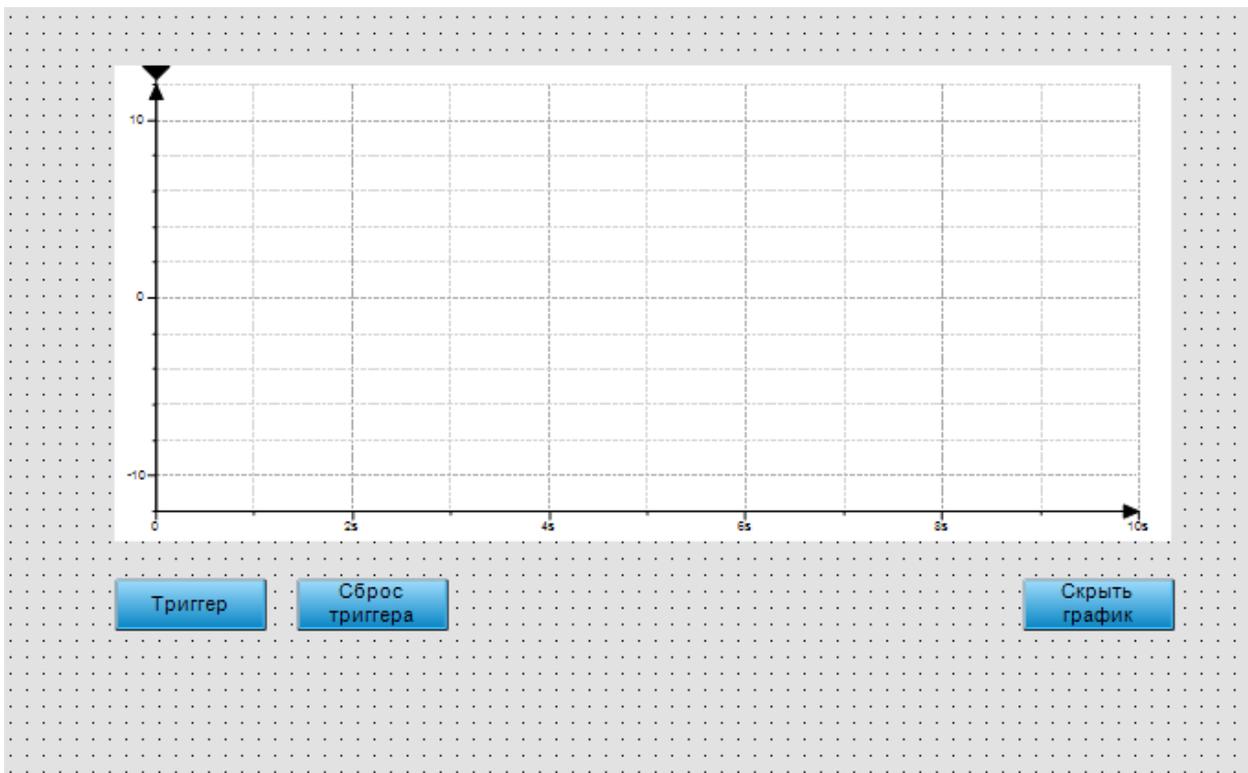


Рис. 11.2.9.4. Содержания экрана **Visualization**

Настройки элементов приведены ниже.

5. Настроим элемент [Трассировка](#):

Свойства	
Свойство	Значения
Имя элемента	GenElemInst_241
Приложение	Device.Application
Тип элемента	Трассировка
Трассировка	Visualization_Trace1
Позиция	
X	73
Y	40
Ширина	639
Высота	289
Показать курсор	<input checked="" type="checkbox"/>
Заменить существующую т...	<input checked="" type="checkbox"/>
Управляющие переменные	
Сброс триггера	PLC_PRG.bResetTrigger
Запустить трассировку	
Остановить трассировку	
Сохранить трассировку...	
Загрузить трассировку...	

Рис. 11.2.9.5. Параметры элемента **Трассировка**

Двойным нажатием на значение параметра **Трассировка (Visualization_Trace1)** откроем меню конфигурации трассировки. Настроим конфигурацию следующим образом:

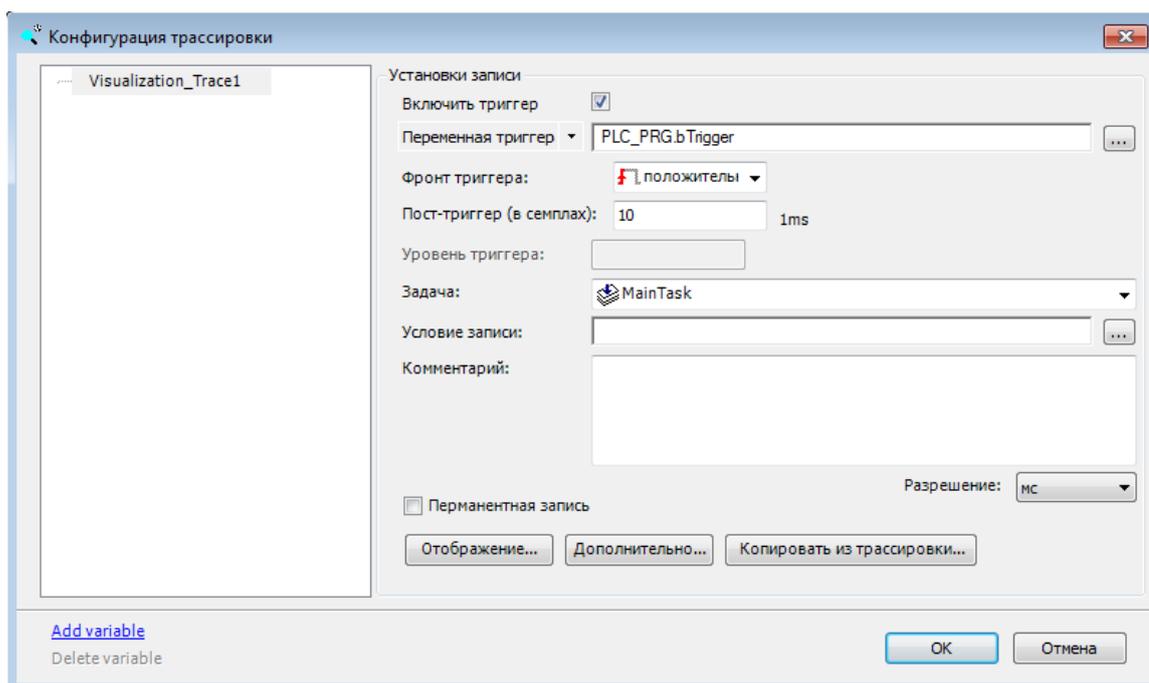


Рис. 11.2.9.6. Конфигурация трассировки

Нажмем на кнопку **Add Variable**, чтобы привязать к элементу переменную, которая будет отображаться на графике:

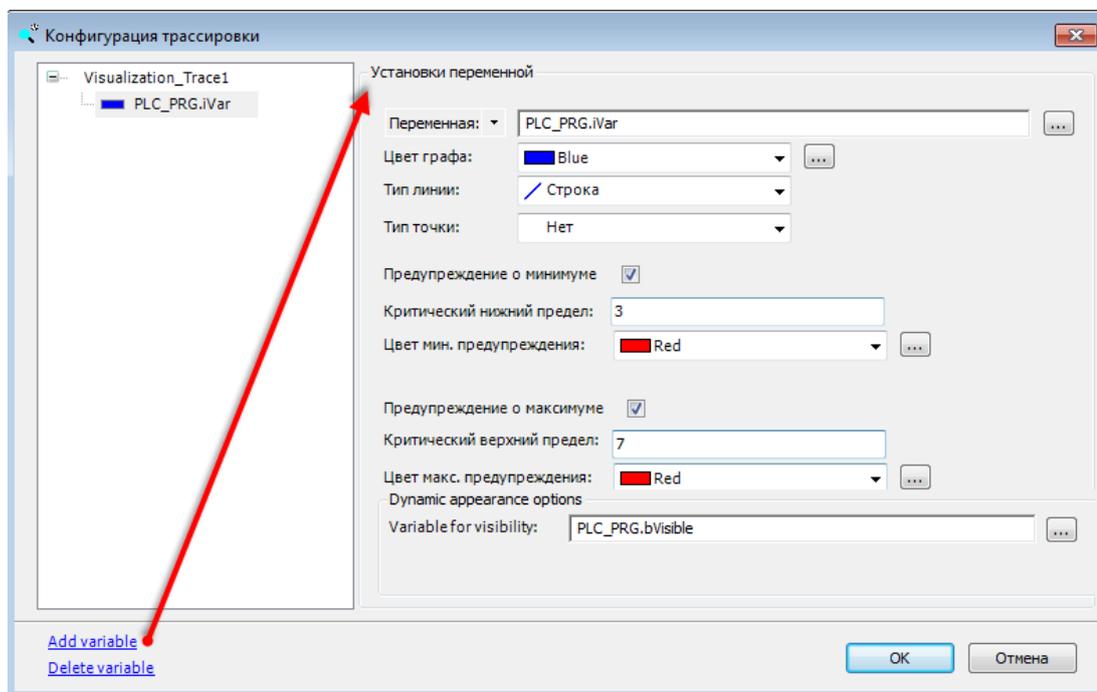


Рис. 11.2.9.7. Привязка переменной к трассировке

6. К кнопке **Триггер** во вкладке [InputConfiguration](#) к параметру **OnClick** привяжем действие [Выполнить ST-код](#):

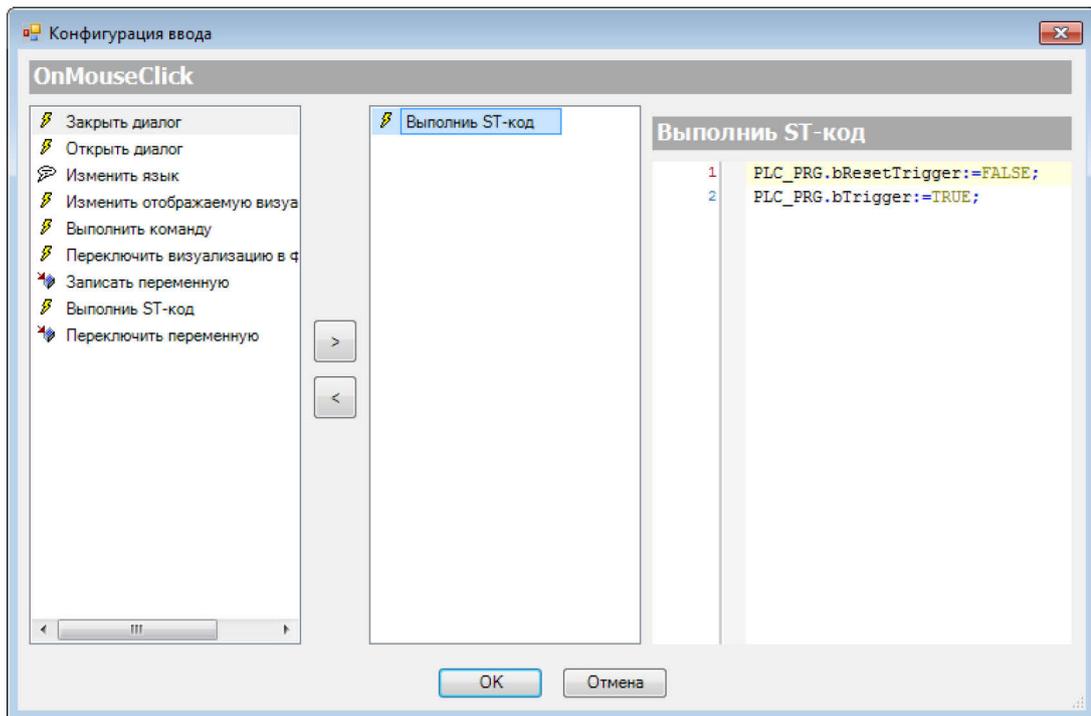


Рис. 11.2.9.8. ST-код кнопки **Триггер**

7. К кнопке **Сброс триггера** во вкладке [InputConfiguration](#) к параметру **OnClick** привяжем действие [Выполнить ST-код](#):

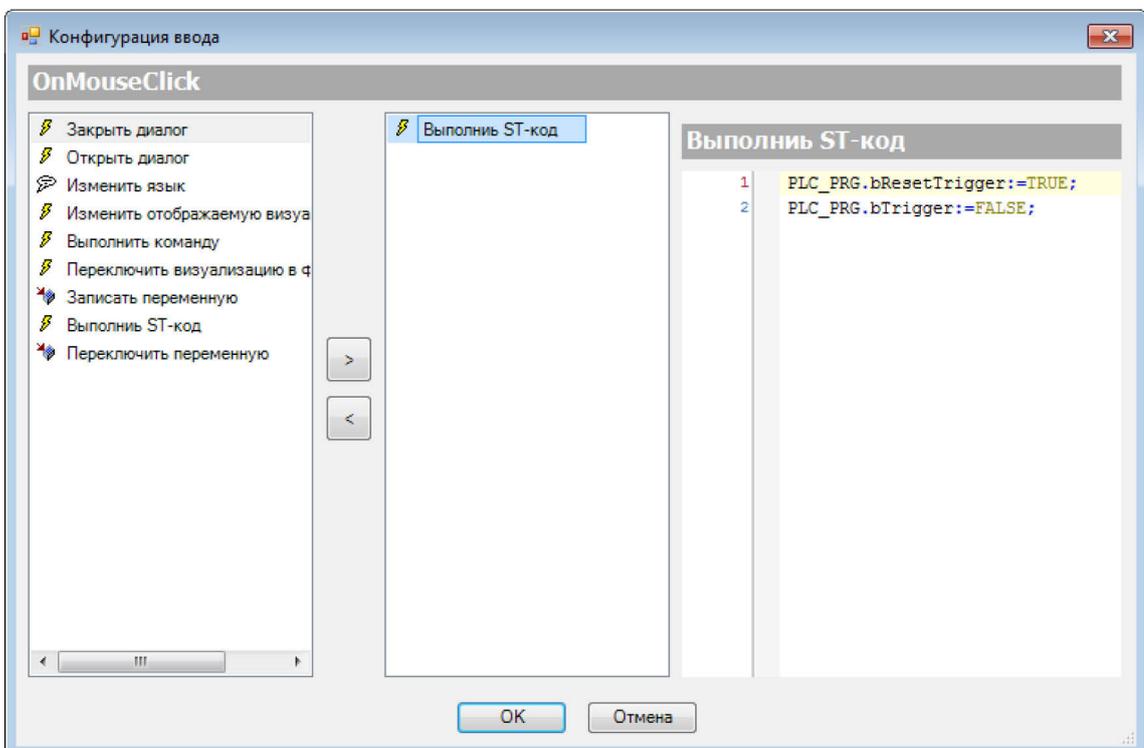


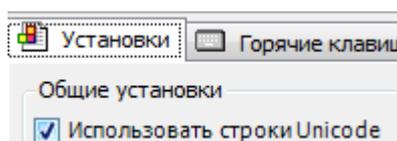
Рис. 11.2.9.9. ST-код кнопки **Сброс триггера**

8. К кнопке **Скрыть график** привяжем переменную **bVisible**:

[-] Inputconfiguration	
OnDialogClosed	Конфигурация...
OnMouseClicked	Конфигурация...
OnMouseDown	Конфигурация...
OnMouseEnter	Конфигурация...
OnMouseLeave	Конфигурация...
OnMouseMove	Конфигурация...
OnMouseUp	Конфигурация...
[+] Нажать	
[-] Переключить	
Переменная	PLC_PRG.bVisible
Переключить наверх...	<input type="checkbox"/>
[+] Горячая клавиша	

Рис. 11.2.9.10. Параметры кнопки **Скрыть график**

9. Настроим компонент [Менеджер визуализации](#). В его установках поставим галочку **Использовать строки Unicode**:



Настройки **target**- и **web**-визуализации будут выглядеть следующим образом:

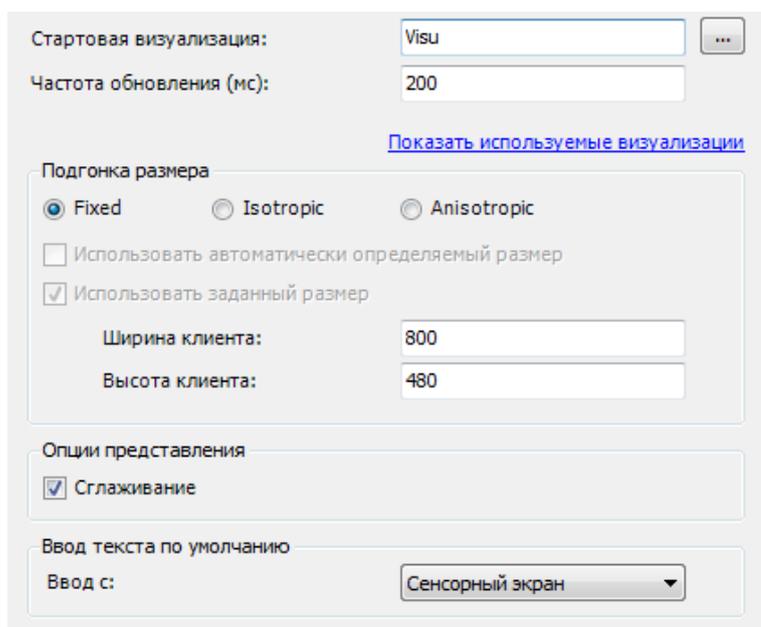


Рис. 11.2.9.11. Настройки **target**-визуализации

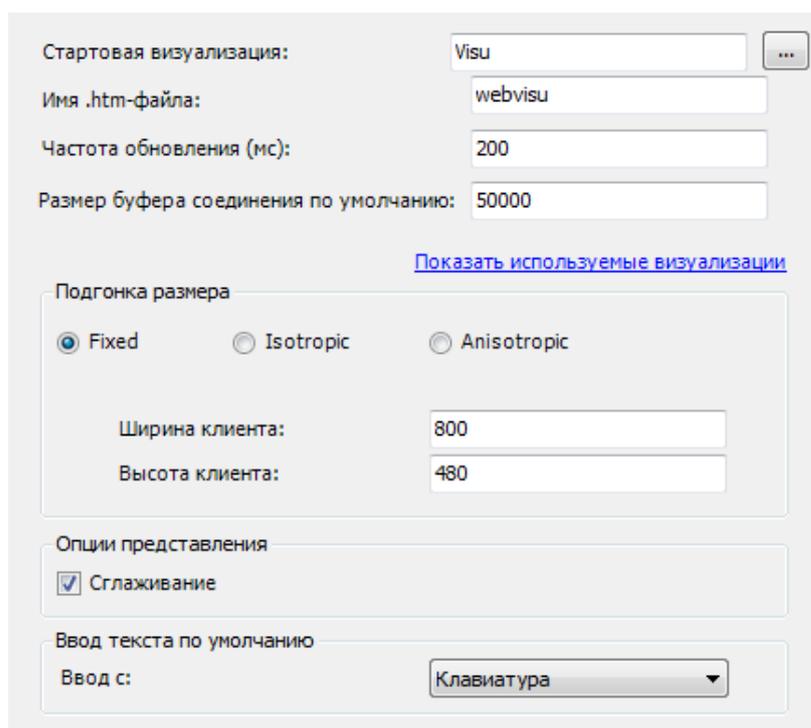


Рис. 11.2.9.12. Настройки **web**-визуализации

10. [Запустим проект на виртуальном контроллере](#). Откроем в веб-браузере страницу <http://localhost:8080/webvisu.htm>, чтобы увидеть web-визуализацию проекта. **Обратите внимание**, что браузер должен поддерживать **HTML5**. Проверим функционал проекта.

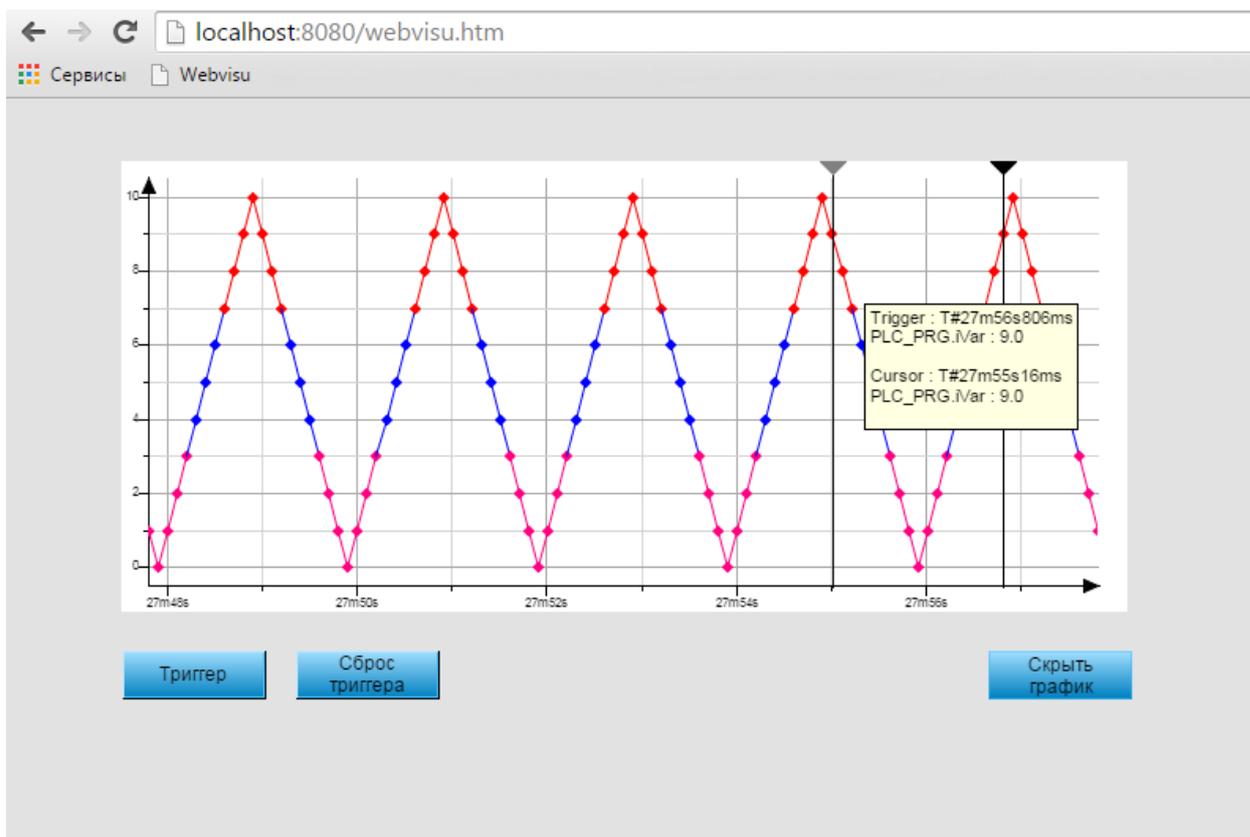


Рис. 11.2.9.13. Web-визуализация проекта в браузере **Chrome**

Нажатие на кнопку **Триггер** останавливает график. При этом при наведении на график отображается курсор (метка со значением времени и переменной).

Нажатие на кнопку **Сброс триггера** запускает график.

Нажатие на кнопку **Скрыть график** делает график невидимым. Повторное нажатие возвращает видимость графику.

11.2.10. Тренд

Данный пример посвящен работе с графическим примитивом [Тренд](#), который используется для построения графика значения переменной с возможностью просмотра истории.

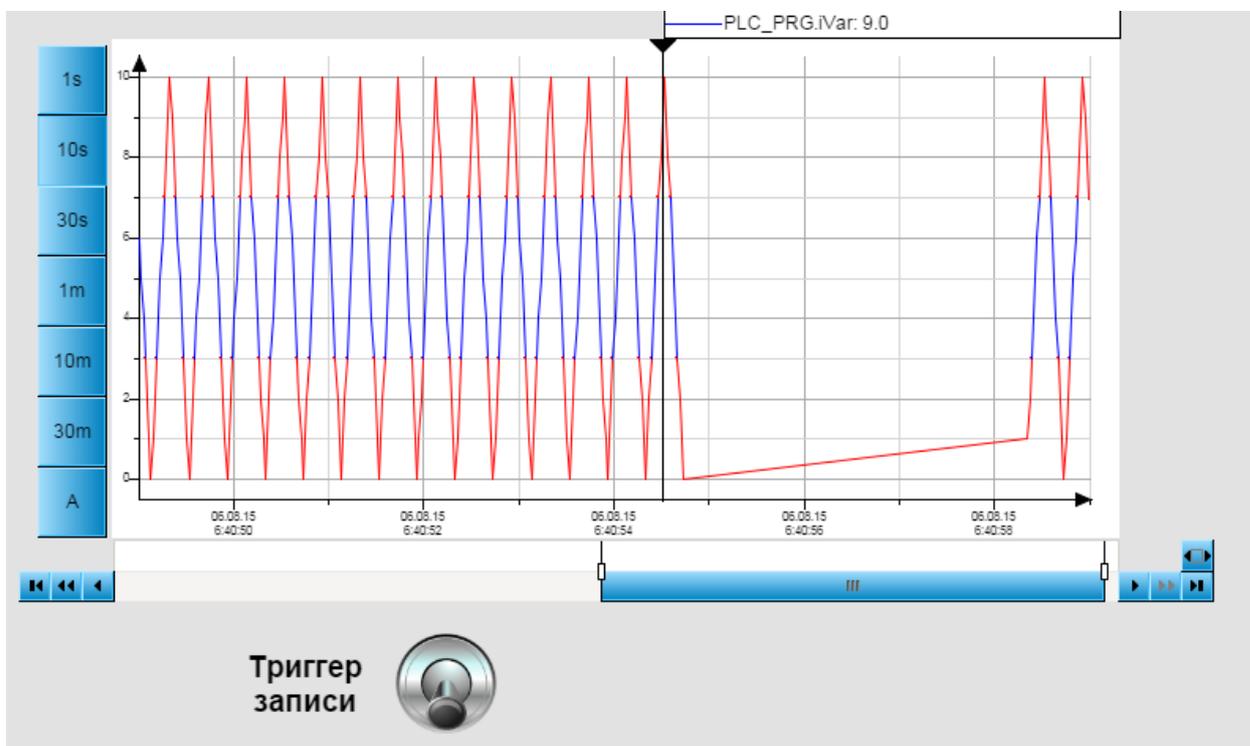


Рис. 11.2.10.1. Внешний вид примера **Тренд**

Пример создан в среде **CODESYS 3.5 SP6 Patch 2** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с target-файлом **3.5.6.0**.

Пример доступен для скачивания: [Example_Trend.projectarchive](#)

1. Создадим новый *стандартный* проект **CODESYS** с названием **Example_Trend** и настройками по умолчанию: target – **CODESYS Control Win V3**, язык PLC_PRG – **ST**.

2. В программе **PLC_PRG** объявим следующие переменные:

```
1 PROGRAM PLC_PRG
2 VAR
3     bVar:BOOL;           // переменная для изменения значений переменной iVar
4     iVar:INT;           // переменная, отображаемая трендом
5
6     bTrigger:BOOL:=TRUE; // триггер запуска/остановки записи
7 END VAR
```

Рис. 11.2.10.2. Объявление переменных программы **PLC_PRG**

3. Код программы **PLC_PRG** будет выглядеть следующим образом:

```
1 // эмуляция изменения значений
2
3 IF bVar=FALSE THEN iVar:=iVar+1;
4     ELSE iVar:=iVar-1;
5 END_IF
6
7 IF     iVar=0 THEN bVar:=FALSE;
8     ELSIF iVar=10 THEN bVar:=TRUE;
9 END_IF
10
```

Рис. 11.2.10.3. Код программы **PLC_PRG**

4. [Добавим в проект](#) экран визуализации **Visualization**. В его [свойствах](#) выберем размер **800x480**. Добавим на экран элемент [Тренд](#). При этом автоматически откроется окно конфигурации тренда. Настроим ее следующим образом:

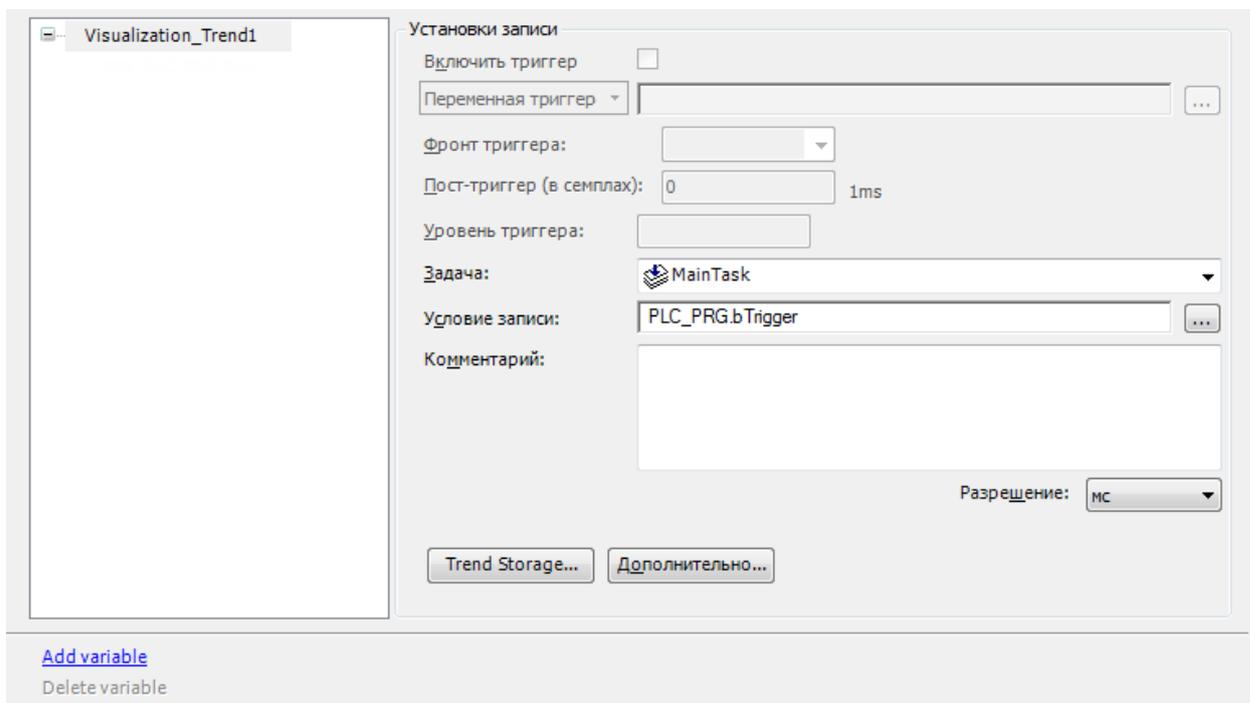


Рис. 11.2.10.4. Конфигурация тренда

Нажмем на кнопку **Add Variable**, чтобы привязать к элементу переменную, которая будет отображаться на графике:

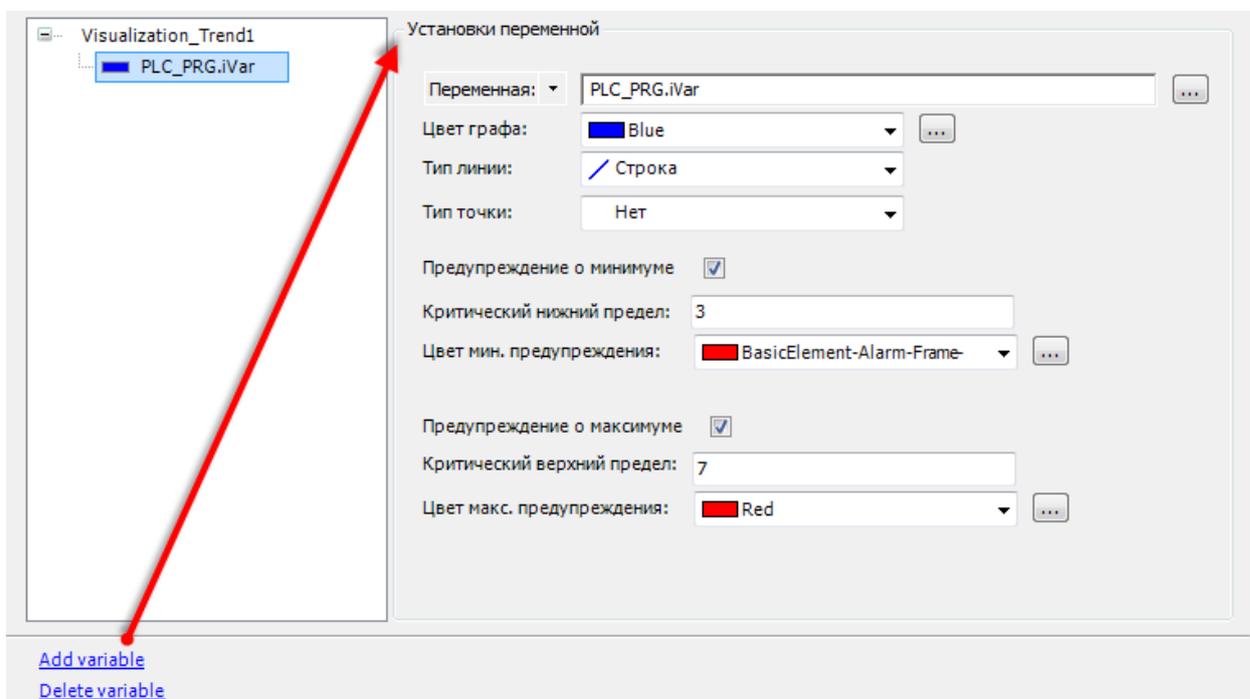


Рис. 11.2.10.5. Привязка переменной к тренду

Растяните тренд с помощью перетаскивания его опорных точек:

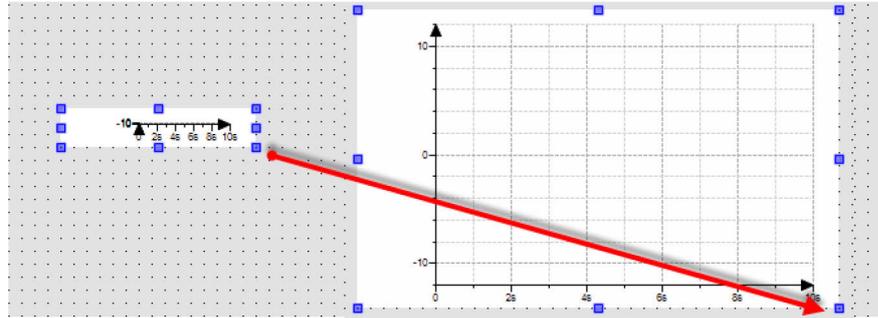


Рис. 11.2.10.6. Изменение размера тренда

Нажмите **ПКМ** на элемент и выберите команду **Вставить элементы управления трендом**:

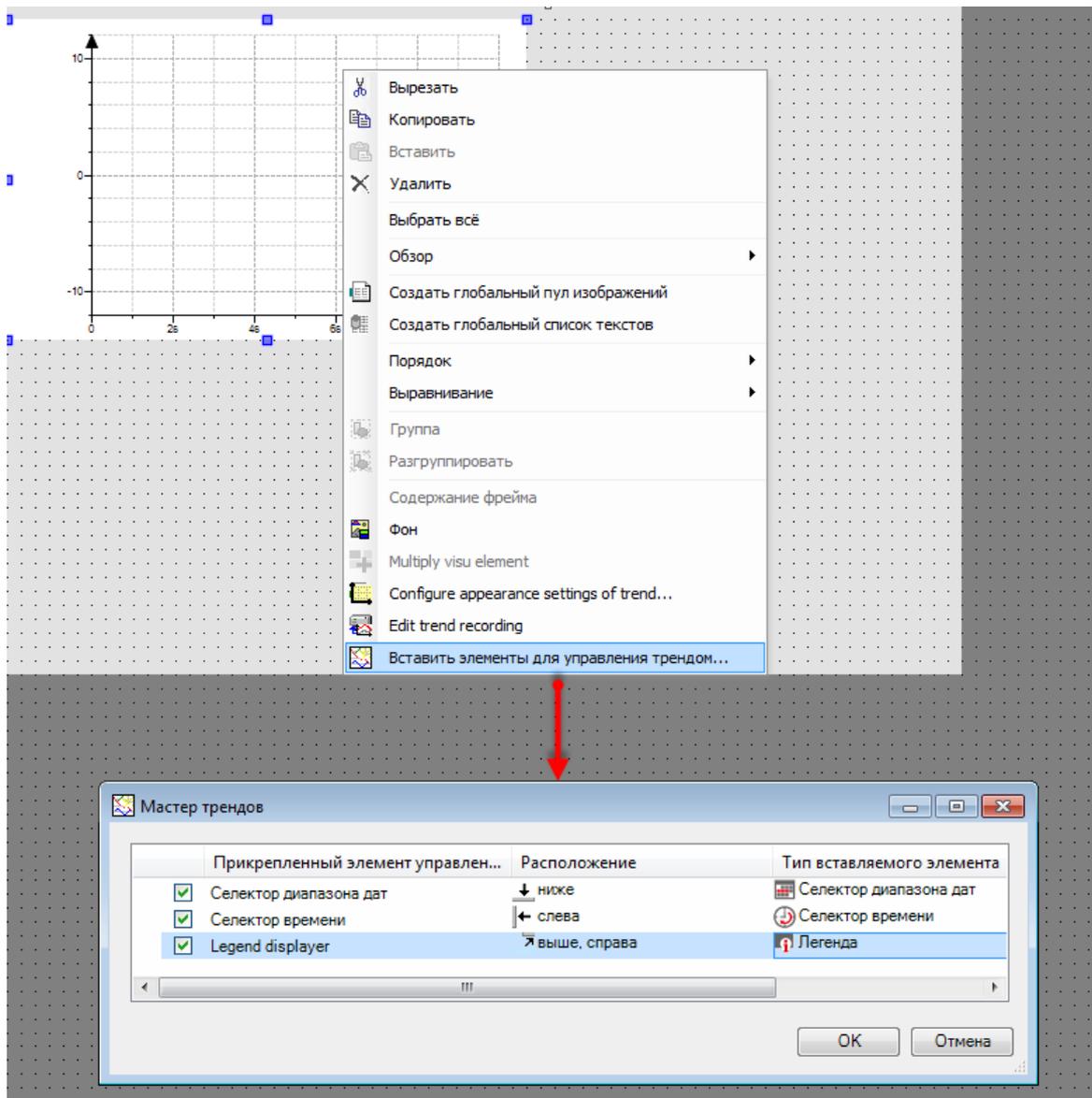


Рис. 11.2.10.7. Добавление элементов управления трендом

Это приведет к автоматическому добавлению и привязке к тренду элементов [Селектор диапазона дат](#), [Селектор времени](#) и [Легенда](#). Настройки элементов приведены ниже.

Свойства	
Ү Фильтр Сортировать по Порядок сортировки	
<input checked="" type="checkbox"/> Эксперт	
Свойство	Значения
Имя элемента	GenElemInst_241
Приложение	Device.Application
Тип элемента	Тренд
Запись тренда	Visualization_Trend1
Внешний вид диагра...	Click here to edit...
<input type="checkbox"/> Позиция	
X	70
Y	20
Ширина	640
Высота	320
Показать курсор	<input checked="" type="checkbox"/>
Показать подсказку	<input checked="" type="checkbox"/>
Показать фрейм	<input type="checkbox"/>
<input type="checkbox"/> обозначения шкалы	
Временные отмет...	Абсолютные временные отметки
Обозначения в д...	<input checked="" type="checkbox"/>
Опустить незнач...	<input type="checkbox"/>
<input type="checkbox"/> Интернационализ...	
Дата	dd.MM.yy
Время	H:mm:ss
<input type="checkbox"/> Прикрепленные экзе...	
Селектор диапазо...	Trend1DateRangeSelector
Селектор времени	Trend1TimeSelector
Легенда	Trend1Legend

Рис. 11.2.10.8. Параметры элемента **Тренд**

Свойства	
̣ Фильтр ̣ Сортировать по ̣ Порядок сортировки ̣ Эксперт	
Свойство	Значения
Имя элемента	Trend1DateRangeSelector
Тип элемента	Селектор диапазона дат
[-] Позиция	
X	10
Y	340
Ширина	760
Высота	40
Прикрепленный экземпляр элемента	GenElemInst_241
[-] обозначения шкалы	
Обозначения в две строки	<input checked="" type="checkbox"/>
Опустить незначимую информаци...	<input type="checkbox"/>
[-] Интернационализация (Format stri...	
Дата	
Время	
[-] Свойства текста	
Горизонтальное выравнивание	По центру
Вертикальное выравнивание	По центру
Шрифт	Arial; 9
[-] Цвет шрифта	
Transparency	255
[-] Дополнительные кнопки	
Переход к максимальной временн...	<input checked="" type="checkbox"/>
Переход к минимальной временн...	<input checked="" type="checkbox"/>
Уменьшить масштаб	<input checked="" type="checkbox"/>

Рис. 11.2.10.9. Параметры элемента **Селектор диапазона дат**

Свойства	
̄ Фильтр ̄ Сортировать по ̄ Порядок сортировки ̄ Эксперт	
Свойство	Значения
Имя элемента	Trend1TimeSelector
Тип элемента	Селектор времени
[-] Позиция	
X	20
Y	10
Ширина	50
Высота	340
Ориентация	Вертикальный
Экземпляр прикрепленного элемента	GenElemInst_241
[-] Тексты	
Текст	
[-] Свойства текста	
Шрифт	Font-Standard
[-] Цвет шрифта	Element-Button-FontColor
Transparency	255
[-] Времена	
Выбор "Все"	<input checked="" type="checkbox"/>
[-] Времена	Создать новый
[-] Времена	
[-] [0]	Удалить
Время	1s
[-] [1]	Удалить
Время	10s
[-] [2]	Удалить
Время	30s
[-] [3]	Удалить
Время	1m
[-] [4]	Удалить
Время	10m
[-] [5]	Удалить
Время	30m
[-] Управляющие переменные	
Время	
Все выбранные	

Рис. 11.2.10.10. Параметры элемента **Селектор времени**

Свойства	
Фильтр	Сортировать по
Эксперт	Порядок сортировки
Свойство	Значения
Имя элемента	Trend1Legend
Тип элемента	Легенда
Позиция	
X	420
Y	0
Ширина	290
Высота	20
Ориентация	Горизонтально
Показать фрейм	<input checked="" type="checkbox"/>
Размещение	
Макс. число строк	1
Макс. число столбцов	2
Свойства текста	
Шрифт	Font-Standard
Цвет шрифта	Font-Default-Color
Transparency	255

Рис. 11.2.10.11. Параметры элемента **Легенда**

Добавьте на экран элемент [Переключатель](#) и привяжите к нему переменную **bTrigger**.
Сделайте поясняющую надпись (**Триггер записи**) с помощью элемента [Метка](#).

Свойства	
Фильтр	Сортировать по
Эксперт	Порядок сортировки
Свойство	Значения
Имя элемента	GenElemInst_247
Тип элемента	Переключатель
Позиция	
X	250
Y	400
Ширина	70
Высота	65
Переменная	PLC_PRG.bTrigger
Параметры изображения	
Поведение элемента	Переключатель изображения
Тексты	
Переменные состояний	
Фон	
Изображение	Gray

Рис. 11.2.10.12. Параметры элемента **Переключатель**

5. **Обратите внимание**, что после добавления на экран элемента [Тренд](#), в проект автоматически был добавлен компонент **Trend Recording Manager** с подкомпонентом **Visualization_Trend1**, в котором хранится конфигурация тренда, а также системная задача **TrendRecordingTask**:

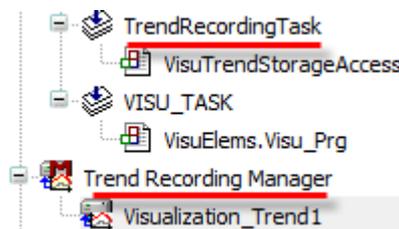
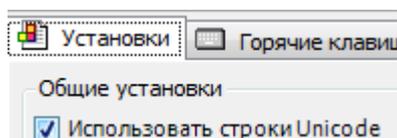


Рис. 11.2.10.13. Автоматически созданные компоненты тренда

6. Настроим компонент [Менеджер визуализации](#). В его установках поставим галочку **Использовать строки Unicode**:



Поскольку **Тренд** является ресурсоемким элементом, поставьте галочку **Дополнительные установки** и увеличьте размер буфера визуализации:

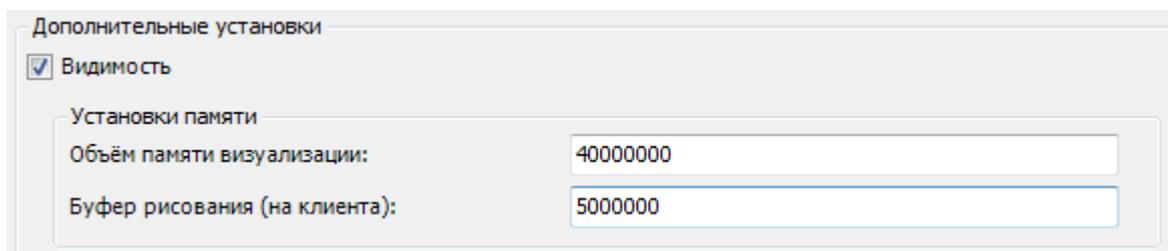
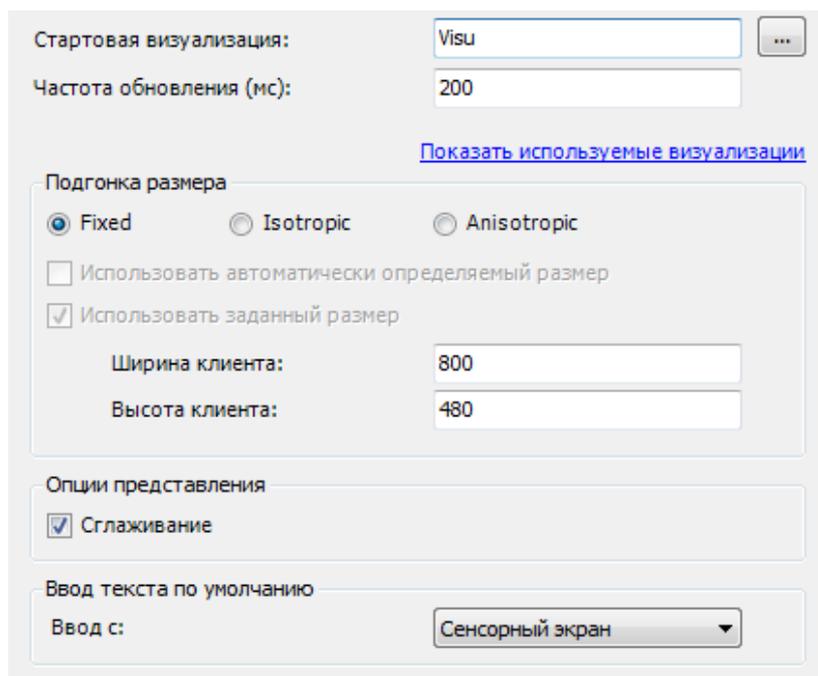


Рис. 11.2.10.14. Дополнительные установки **Менеджера визуализации**

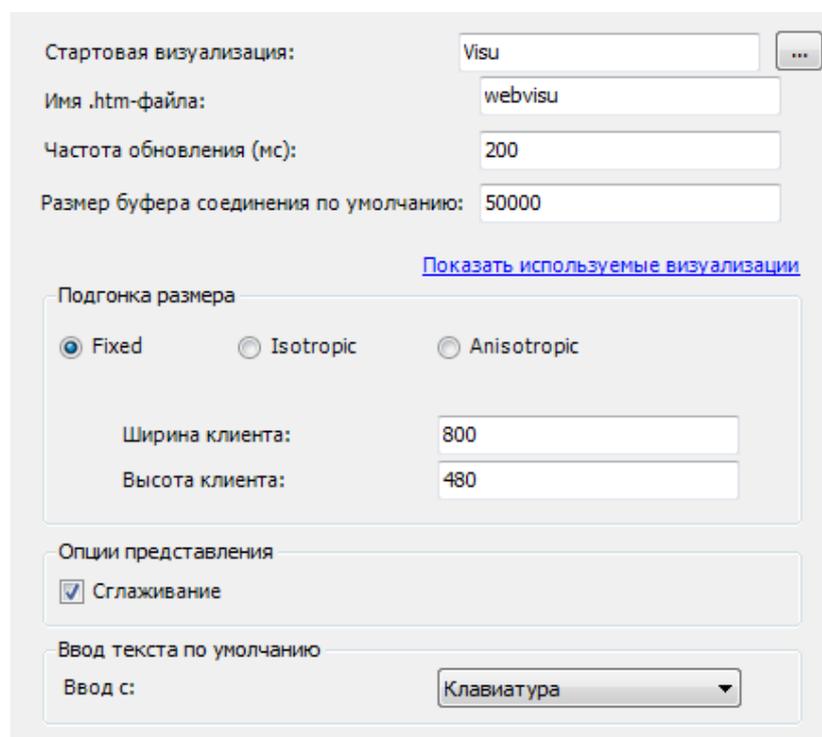
Настройки **target**- и **web**-визуализации будут выглядеть следующим образом:



Скриншот панели настроек для target-визуализации. Включены следующие параметры:

- Стартовая визуализация: Visu
- Частота обновления (мс): 200
- Подгонка размера: Fixed (выбрано), Isotropic, Anisotropic
- Использовать автоматически определяемый размер:
- Использовать заданный размер:
- Ширина клиента: 800
- Высота клиента: 480
- Опции представления: Сглаживание (выбрано)
- Ввод текста по умолчанию: Сенсорный экран

Рис. 11.2.10.15. Настройки **target**-визуализации



Скриншот панели настроек для web-визуализации. Включены следующие параметры:

- Стартовая визуализация: Visu
- Имя .htm-файла: webvisu
- Частота обновления (мс): 200
- Размер буфера соединения по умолчанию: 50000
- Подгонка размера: Fixed (выбрано), Isotropic, Anisotropic
- Ширина клиента: 800
- Высота клиента: 480
- Опции представления: Сглаживание (выбрано)
- Ввод текста по умолчанию: Клавиатура

Рис. 11.2.10.16. Настройки **web**-визуализации

7. [Запустим проект на виртуальном контроллере](#). Откроем в веб-браузере страницу <http://localhost:8080/webvisu.htm>, чтобы увидеть web-визуализацию проекта. **Обратите внимание**, что браузер должен поддерживать **HTML5**. Проверим функционал проекта.

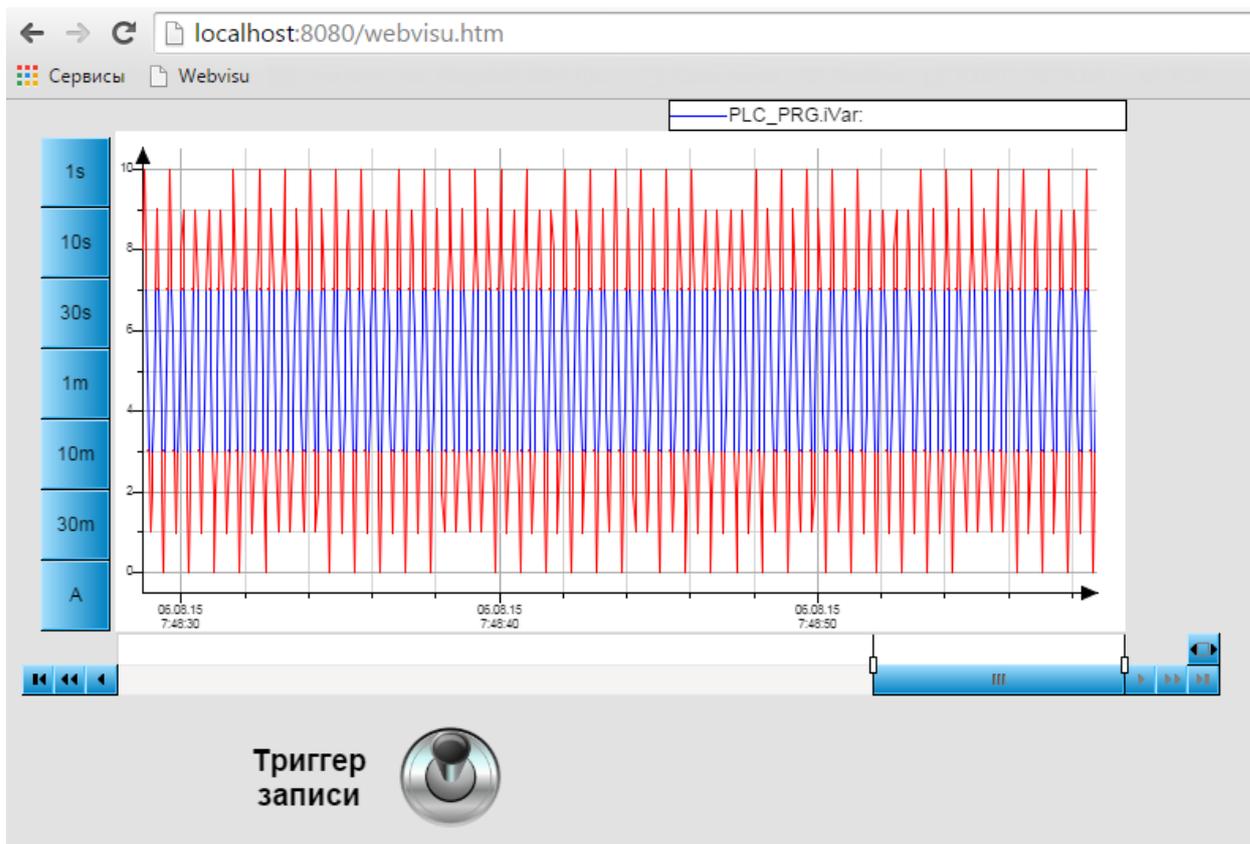


Рис. 11.2.10.17. Web-визуализация проекта в браузере **Chrome**

Нажатие на переключатель **Триггер записи** останавливает запись тренда. Повторное нажатие запускает запись заново, причем последняя точка предыдущей записи будет соединена с первой точкой новой:

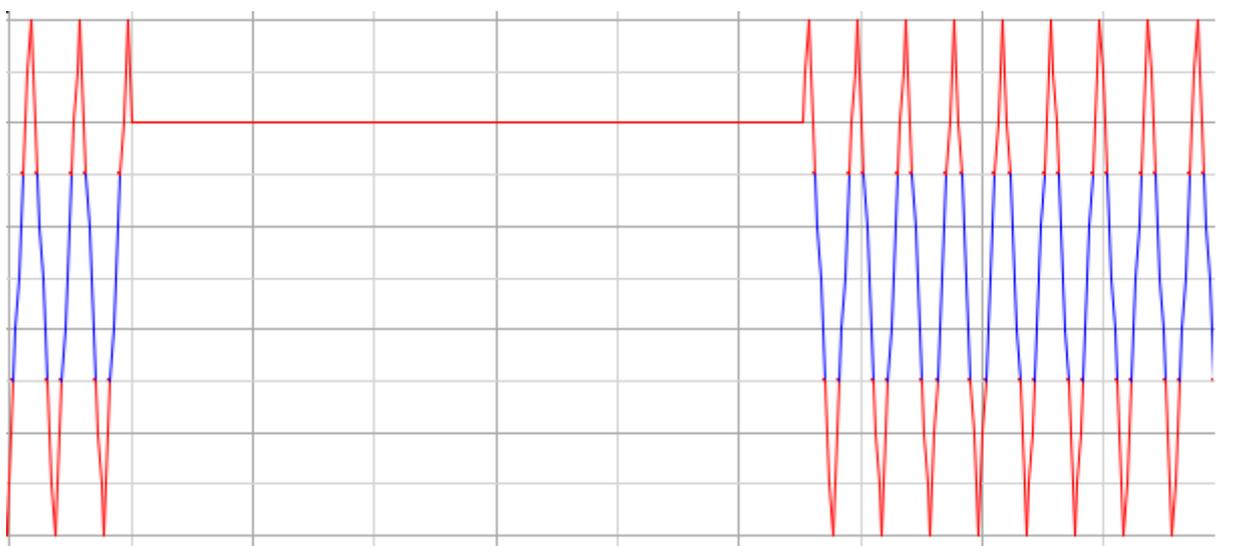


Рис. 11.2.10.18. Внешний вид тренда после включения и отключения триггера записи

С помощью кнопок **Селектора времени** можно менять масштаб тренда по оси времени:

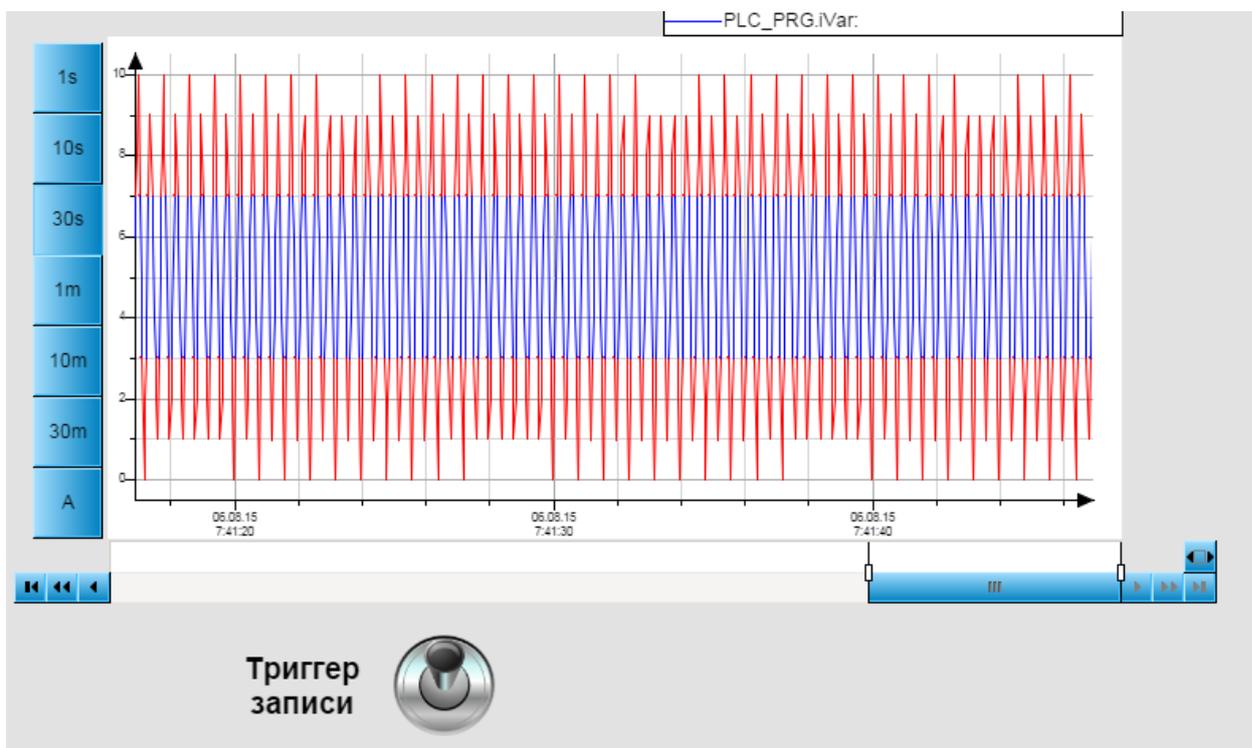


Рис. 11.2.10.19. Внешний вид тренда (масштаб времени = 30 с)

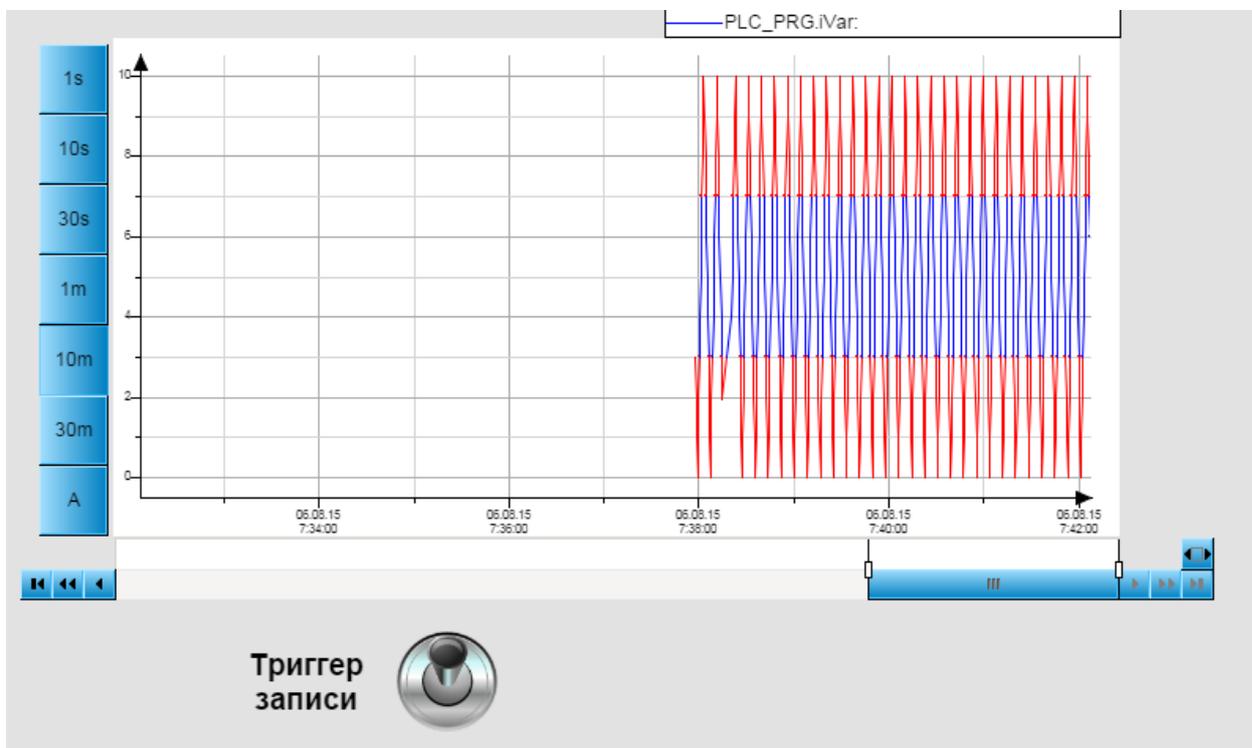


Рис. 11.2.10.20. Внешний вид тренда (масштаб времени = 10 мин.)

С помощью кнопок **Селектора диапазона дат** можно просматривать историю тренда. Функции кнопок описаны в табл. 31. В режиме просмотра истории при наведении курсора на график, текущее значение переменной отображается легендой, а также в метке курсора:

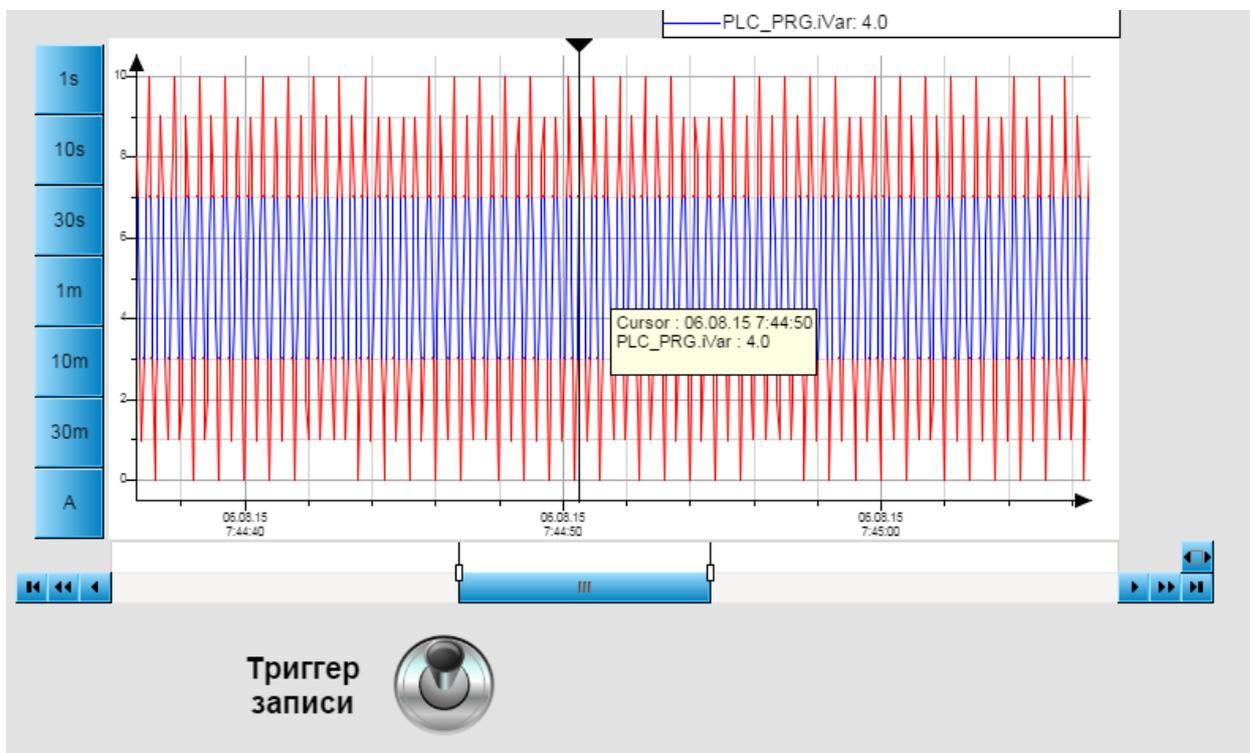


Рис. 11.2.10.21. Внешний вид тренда в режиме просмотра истории

Для возвращения тренда в режим обновления в реальном времени нажмите кнопку .

11.2.11. WebBrowser

Данный пример посвящен работе с графическим примитивом [WebBrowser](#), используемом для открытия в плоскости экрана визуализации веб-страниц. **Обратите внимание**, что использование примитива возможно только на целевых устройствах с поддержкой **Java** и **HTML5**. При этом невозможно гарантировать открытие в элементе любой веб-страницы.

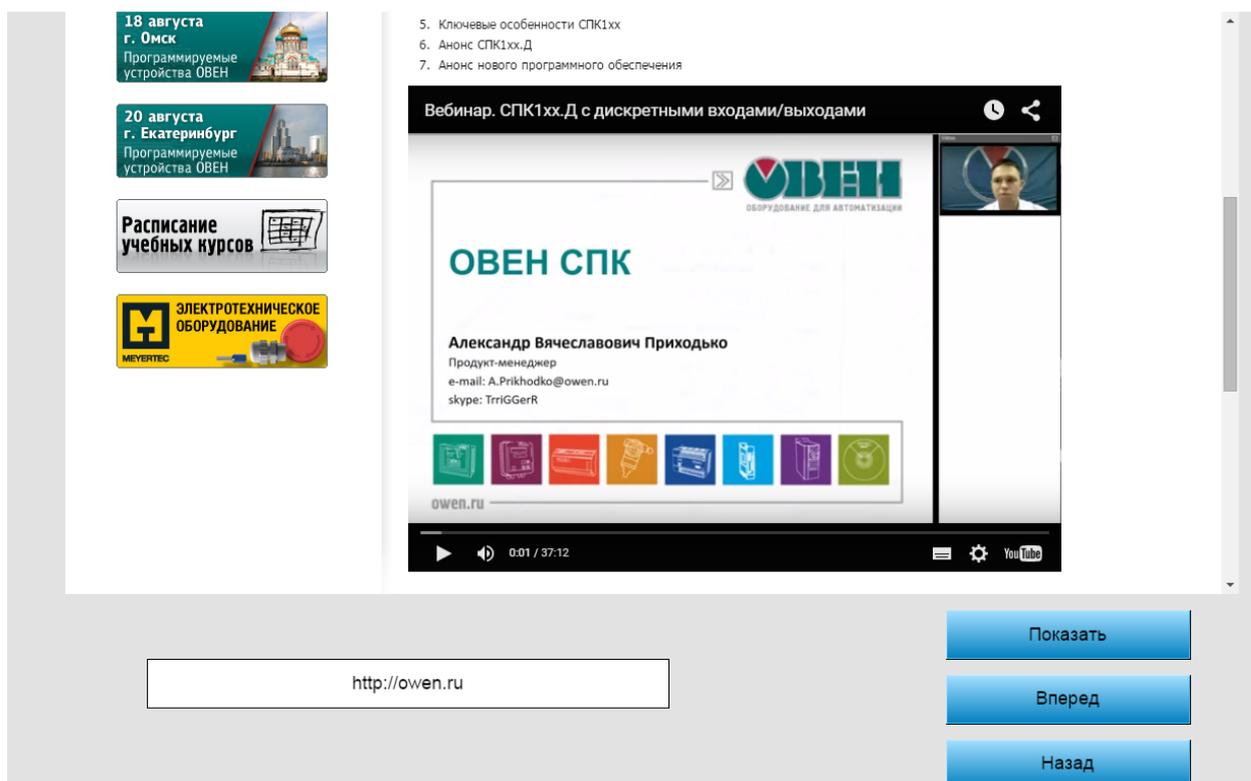


Рис. 11.2.11.1. Внешний вид примера **WebBrowser**

Пример создан в среде **CODESYS 3.5 SP6 Patch 2** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с target-файлом **3.5.6.0**.

Пример доступен для скачивания: [Example_WebBrowser.projectarchive](#)

1. Создадим новый *стандартный* проект **CODESYS** с названием **Example_WebBrowser** и настройками по умолчанию: target – **CODESYS Control Win V3**, язык PLC_PRG – **ST**.

2. В программе **PLC_PRG** объявим следующие переменные:

```
1 PROGRAM PLC_PRG
2 VAR
3     sURL:STRING:='Enter your URL (http://owen.ru)';
4     bShow:BOOL;
5     bBack:BOOL;
6     bForward:BOOL;
7 END_VAR
```

Рис. 11.2.11.2. Объявление переменных программы **PLC_PRG**

3. Добавим в проект экран визуализации **Visualization**. В его свойствах выберем размер **800x480**. Экран будет содержать элемент WebBrowser, элемент Текстовое поле и три элемента Кнопка.

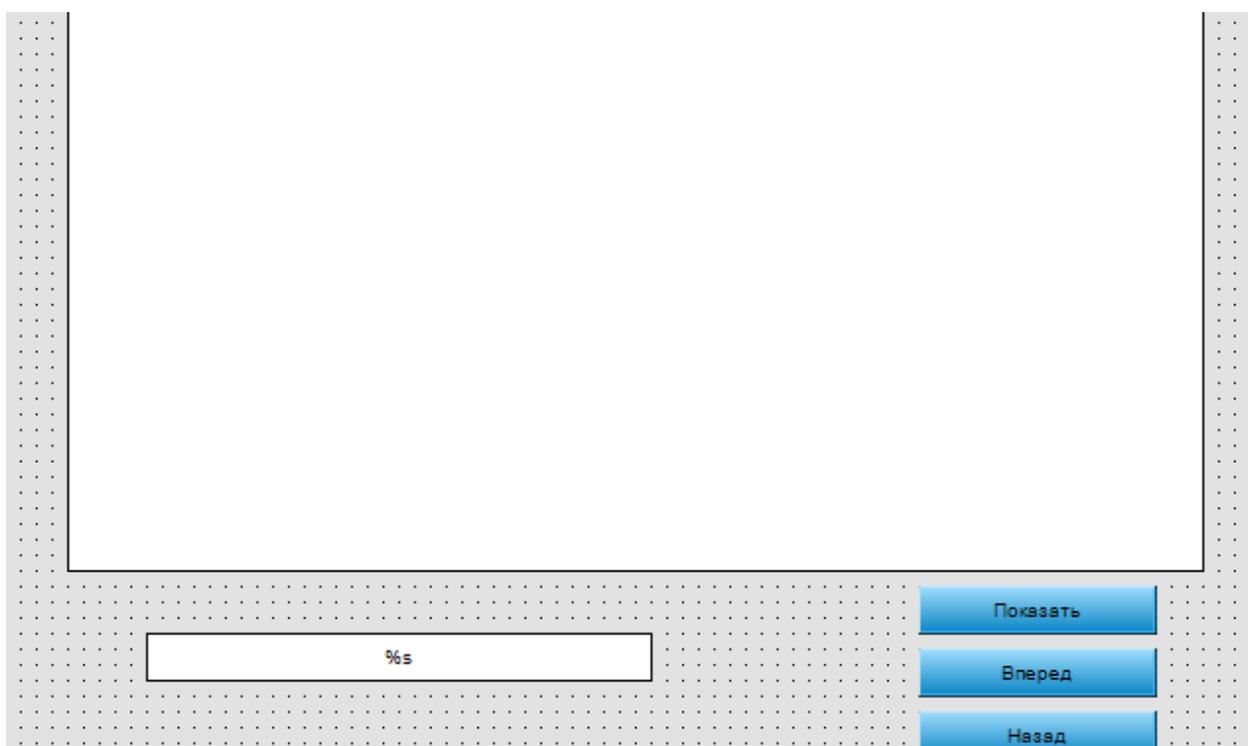


Рис. 11.2.11.3. Содержание экрана **Visualization**

Настройки элементов приведены ниже.

4. Настроим элемент [WebBrowser](#):

Свойства	
Свойство	Значения
Имя элемента	GenElemInst_241
Тип элемента	
[-] Позиция	
X	40
Y	0
Ширина	720
Высота	360
[-] Абсолютное перемещение	
[-] Перемещение	
X	
Y	
[-] Переменные состояний	
Невидимый	
[-] Control variables	
URL	PLC_PRG.sURL
Show	PLC_PRG.bShow
Back	PLC_PRG.bBack
Forward	PLC_PRG.bForward

Рис. 11.2.11.4. Параметры элемента **WebBrowser**

5. Настроим элемент [Текстовое поле](#):

Свойство	Значения
Имя элемента	GenElemInst_243
ID текста	0
Тип элемента	Прямоугольник
Позиция	
X	90
Y	400
Ширина	320
Высота	30
Центр	
Цвета	
Использовать градиентный цв...	<input type="checkbox"/>
Установка градиента	linear, Black, White
Вид элемента	
Тексты	
Текст	%s
Подсказка	
Свойства текста	
Абсолютное перемещение	
Относительное перемещение	
Текстовые переменные	
Текстовая переменная	PLC_PRG.sURL
Переменная подсказки	
Динамические тексты	
Переменные шрифта	
Переменные цвета	
Переменные вида	
Переменные состояний	
Inputconfiguration	
OnDialogClosed	Конфигурация...
OnMouseClicked	Конфигурация...
Записать переменную	Variable ; , InputType : Edit, Min : , Max : , Dialo...

Рис. 11.2.11.5. Параметры элемента **Текстовое поле**

Во вкладке [InputConfiguration](#) к параметру **OnMouseClicked** привяжем действие [Записать переменную](#):

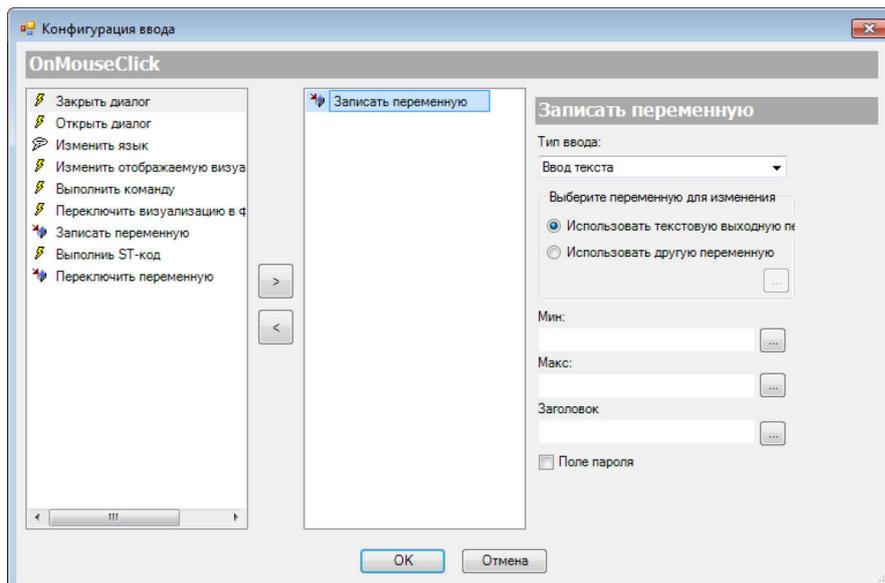


Рис. 11.2.11.6. Настройки действия элемента **Текстовое поле**

6. К кнопке **Показать** привяжем переменную **bShow**:

Inputconfiguration	
OnDialogClosed	Конфигурация...
OnMouseClicked	Конфигурация...
OnMouseDown	Конфигурация...
OnMouseEnter	Конфигурация...
OnMouseLeave	Конфигурация...
OnMouseMove	Конфигурация...
OnMouseUp	Конфигурация...
Нажать	
Переменная	PLC_PRG.bShow
Переключить на FALSE	<input type="checkbox"/>
Переключить на вход,...	<input type="checkbox"/>
+ Переключить	
+ Горячая клавиша	

Рис. 11.2.11.7. Параметры кнопки **Показать**

К кнопке **Вперед** привяжем переменную **bForward**:

Inputconfiguration	
OnDialogClosed	Конфигурация...
OnMouseClicked	Конфигурация...
OnMouseDown	Конфигурация...
OnMouseEnter	Конфигурация...
OnMouseLeave	Конфигурация...
OnMouseMove	Конфигурация...
OnMouseUp	Конфигурация...
Нажать	
Переменная	PLC_PRG.bForward
Переключить на FALSE	<input type="checkbox"/>
Переключить на вход,...	<input type="checkbox"/>
+ Переключить	
+ Горячая клавиша	

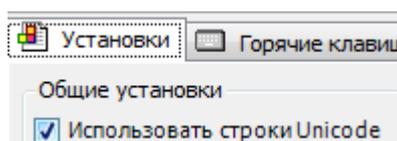
Рис. 11.2.11.8. Параметры кнопки **Вперед**

К кнопке **Назад** привяжем переменную **bBack**:

Inputconfiguration	
OnDialogClosed	Конфигурация...
OnMouseClicked	Конфигурация...
OnMouseDown	Конфигурация...
OnMouseEnter	Конфигурация...
OnMouseLeave	Конфигурация...
OnMouseMove	Конфигурация...
OnMouseUp	Конфигурация...
Нажать	
Переменная	PLC_PRG.bBack
Переключить на FALSE	<input type="checkbox"/>
Переключить на вход,...	<input type="checkbox"/>
+ Переключить	
+ Горячая клавиша	

Рис. 11.2.11.9. Параметры кнопки **Назад**

7. Настроим компонент [Менеджер визуализации](#). В его установках поставим галочку **Использовать строки Unicode**:



Настройки **target**- и **web**-визуализации будут выглядеть следующим образом:

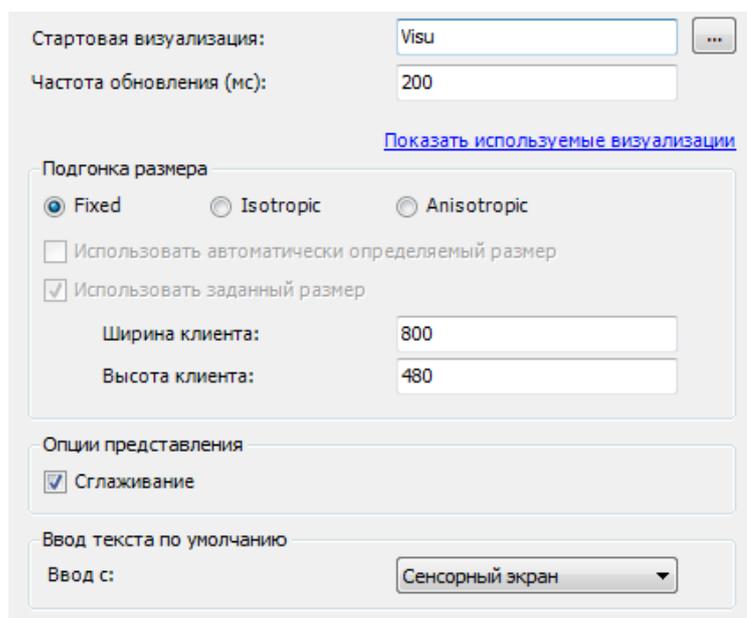


Рис. 11.2.11.10. Настройки **target**-визуализации

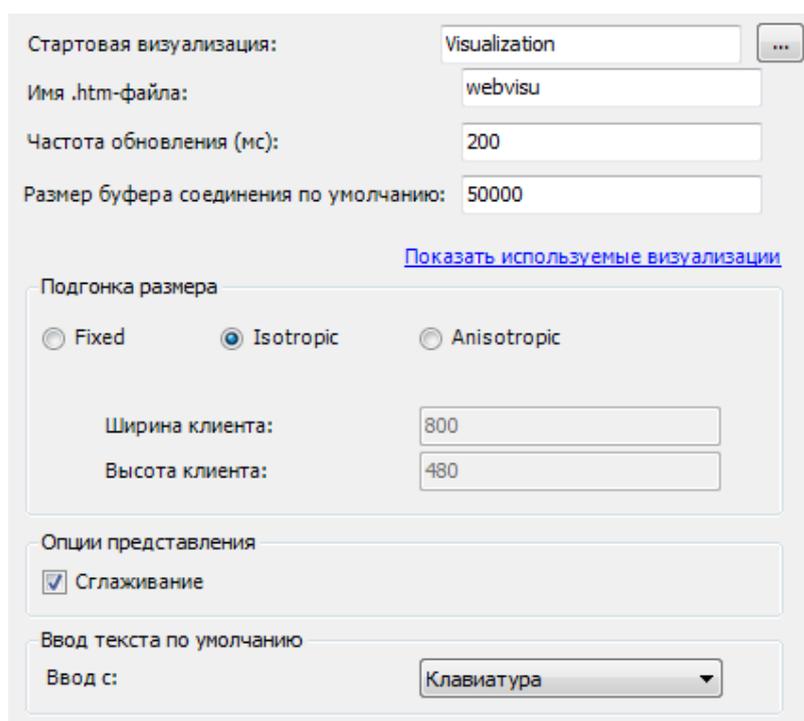


Рис. 11.2.11.11. Настройки **web**-визуализации

8. [Запустим проект на виртуальном контроллере](#). Откроем в веб-браузере страницу <http://localhost:8080/webvisu.htm>, чтобы увидеть web-визуализацию проекта. **Обратите внимание**, что браузер должен поддерживать **HTML5**. Проверим функционал проекта.



Рис. 11.2.11.12. Web-визуализация проекта в браузере **Chrome**

Введите в текстовом поле адрес веб-страницы (в формате <http://owen.ru>) и нажмите кнопку **Показать**. Это приведет к открытию соответствующей веб-страницы в элементе.

Откройте в элементе другую страницу (с помощью ввода ее адреса). Нажмите кнопку **Назад**, чтобы вернуться на предыдущую страницу. Нажмите кнопку **Вперед**, чтобы перейти обратно.

11.2.12. Текстовый редактор

Данный пример посвящен работе с графическим примитивом [Текстовый редактор](#), используемым для просмотра и редактирования текстовых файлов (в формате `.txt`).

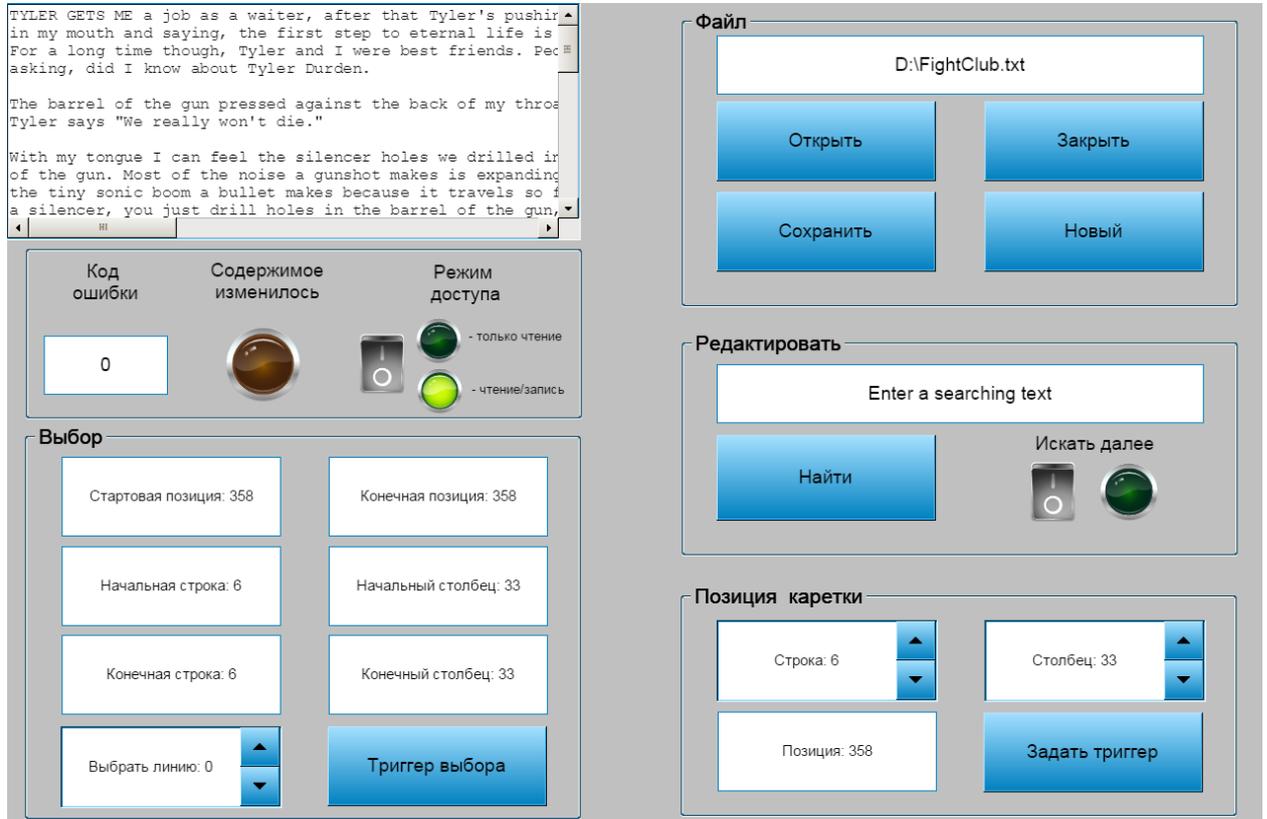


Рис. 11.2.12.1. Внешний вид примера **Текстовый редактор**

Пример создан в среде **CODESYS 3.5 SP6 Patch 2** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с target-файлом **3.5.6.0**.

Пример доступен для скачивания: [Example_TextEditor.projectarchive](#)

1. Создадим новый *стандартный* проект **CODESYS** с названием **Example_TextEditor** и настройками по умолчанию: target – **CODESYS Control Win V3**, язык PLC_PRG – **ST**.

2. В программе **PLC_PRG** объявим следующие переменные:

```
1  PROGRAM PLC_PRG
2  VAR
3      // файл
4      sFileName:STRING:='Choose a path to file (D:\1.txt)'; // Имя файла
5      bOpen:BOOL; // Открыть
6      bClose:BOOL; // Закрыть
7      bSave:BOOL; // Сохранить
8      bNew:BOOL; // Новый
9
10     // Редактировать
11     sSearch:STRING:='Enter a searching text'; // Искать
12     bFind:BOOL; // Найти
13     bFindNext:BOOL; // Найти далее
14
15     // Позиция каретки
16     iLine:INT; // Линия
17     iColumn:INT; // Столбец
18     iPosition:INT; // Позиция
19     bTriggerSet:BOOL; // Задать триггер
20
21     // Выбор
22     iStartPosition:INT; // Стартовая позиция
23     iEndPosition:INT; // Конечная позиция
24     iStartLineNumber:INT; // Начальный номер строки
25     iStartColumnIndex:INT; // Начальный индекс столбца
26     iEndLineNumber:INT; // Конечный номер строки
27     iEndColumnIndex:INT; // Конечный индекс столбца
28     iLineToSelect:INT; // Выбираемая линия
29     bTriggerSelect:BOOL; // Выбор триггера
30
31     // Обработка кода ошибок
32     iErrorCode:INT; // Переменная для кода ошибки
33
34     //
35     bContentChanged:BOOL; // Переменная для содержимого изменилась
36     bAccessMode:BOOL; // Переменная для режима доступа
37     bAccessModeRev:BOOL; // Инвертированное значение режима доступа к файлу (для лампы)
38 END_VAR
```

Рис. 11.2.12.2. Объявление переменных программы **PLC_PRG**

3. Код программы будет выглядеть следующим образом:

```
1  bAccessModeRev:=NOT bAccessMode; // инвертированное значение режима доступа к файлу (для лампы)
```

Рис. 11.2.12.3. Код программы **PLC_PRG**

4. Добавим в проект экран визуализации **Visualization**. В его свойствах выберем размер **1280x860**. Экран будет содержать элемент Текстовый редактор, три элемента Управление вращением, семь элементов Кнопка, семь элементов Текстовое поле, четыре элемента Индикатор, два элемента Клавишный выключатель и пять элементов Группа. Шесть пояснительных текстовых надписей сделаны с помощью элементов Метка.

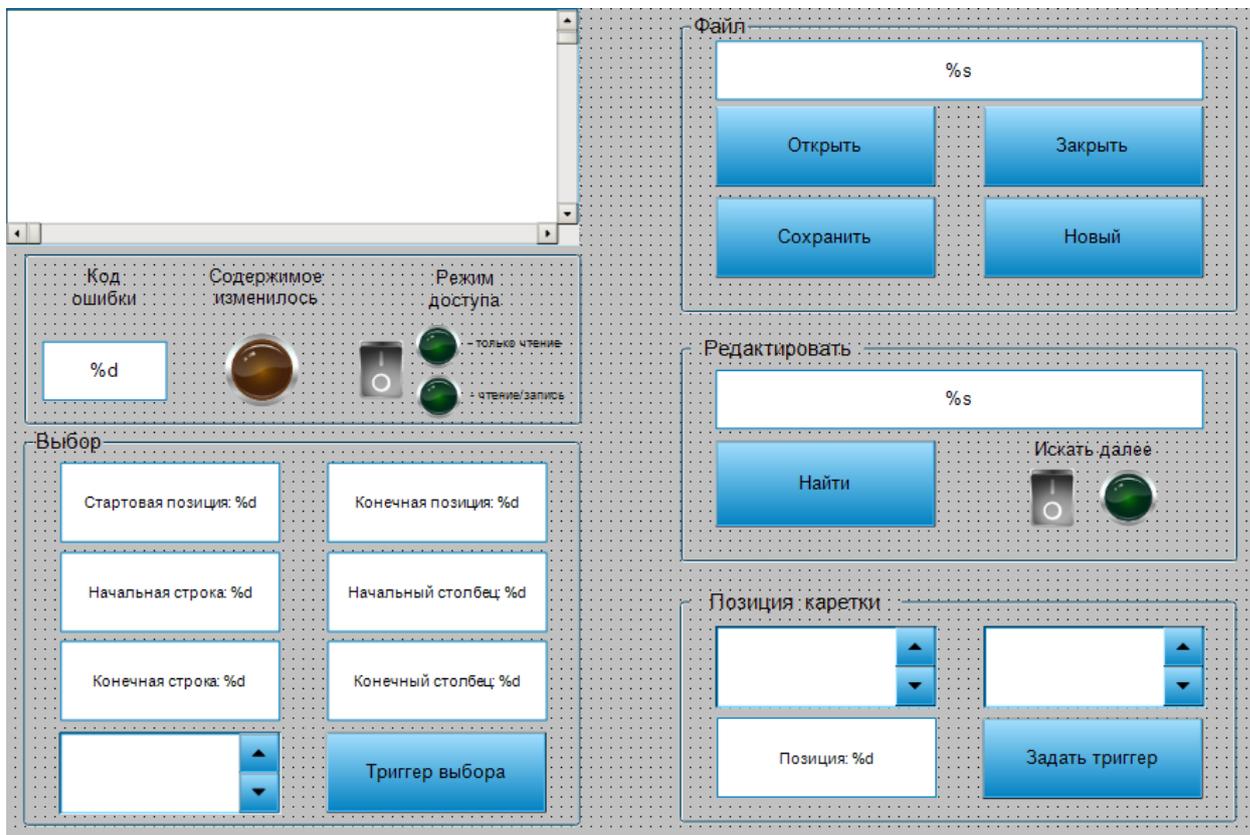


Рис. 11.2.12.4. Содержание экрана **Visualization**

Настройки элементов приведены ниже.

5. Настроим элемент Текстовый редактор:

Свойства	
Фильтр Сортировать по Порядок сортировки Эксперт	
Свойство	Значения
Имя элемента	GenElemInst_54
Тип элемента	Текстовый редактор
Позиция	
X	0
Y	0
Ширина	578
Высота	240
Шрифт	
Имя шрифта	Courier New
Размер	12
Управляющие переменные	
Файл	
Имя файла	PLC_PRG.sFileName
Открыть	PLC_PRG.bOpen
Закрыть	PLC_PRG.bClose
Сохранить	PLC_PRG.bSave
Новый	PLC_PRG.bNew
Редактировать	
Искать	PLC_PRG.sSearch
Найти	PLC_PRG.bFind
Найти далее	PLC_PRG.bFindNext
Позиция каретки	
Линия	PLC_PRG.iLine
Столбец	PLC_PRG.iColumn
Позиция	PLC_PRG.iPosition
Задать триггер	PLC_PRG.bTriggerSet
Выбор	
Стартовая позиция	PLC_PRG.iStartPosition
Конечная позиция	PLC_PRG.iEndPosition
Стартовый номер строки	PLC_PRG.iStartLineNumber
Начальный индекс столбца	PLC_PRG.iStartColumnIndex
Конечный индекс строки	PLC_PRG.iEndLineNumber
Конечный индекс столбца	PLC_PRG.iEndColumnIndex
Выбираемая линия	PLC_PRG.iLineToSelect
Выбор триггера	PLC_PRG.bTriggerSelect
Обработка ошибок	
Переменная для кода ошибки	PLC_PRG.iErrorCode
Переменная для содержимого изменилась	PLC_PRG.bContentChanged
Переменная для режима доступа	PLC_PRG.bAccessMode
Макс. длина линии	200
Режим редактора	Чтение/запись
Новые файлы	
Кодировка	ASCII
Новая последовательность символов строки	CR/LF

Рис. 11.2.12.5. Параметры элемента **Текстовый редактор**

6. Добавим на экран панель управления параметрами вкладки **Файл**. Эти параметры позволяют открывать, закрыть, сохранять и создавать текстовые файлы.

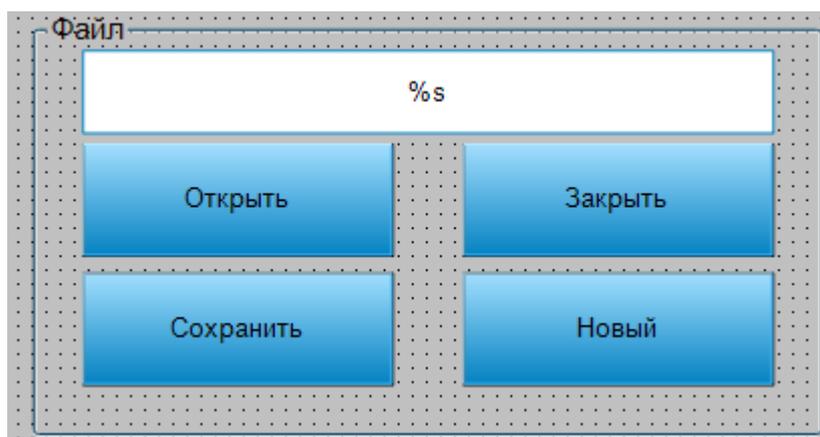


Рис. 11.2.12.6. Панель управления параметрами вкладки **Файл**

Панель представляет собой элемент [Группа](#), в которой расположен элемент [Текстовое поле](#) и четыре элемента [Кнопка](#). Параметры элементов приведены ниже (не отображенные параметры находятся в значениях по умолчанию).

Свойства	
̣ Фильтр ̣ Сортировать по ̣ Порядок сортировки ̣ Эксперт	
Свойство	Значения
Имя элемента	GenElemInst_66
Тип элемента	Текстовое поле
ID текста	0
⊕ Позиция	
⊕ Цвета	
⊕ Вид элемента	
Тип тени	Из стиля
⊖ Тексты	
Текст	%s
Подсказка	
⊕ Свойства текста	
⊖ Текстовые переменные	
Текстовая переменная	PLC_PRG.sFileName
Переменная подсказки	
⊕ Динамические тексты	
⊕ Переменные шрифта	
⊕ Переменные цвета	
⊕ Переменные состояний	
⊖ Inputconfiguration	
OnDialogClosed	Конфигурация...
⊖ OnMouseClicked	Конфигурация...
Записать переменную	Variable : , InputType : Edit, Min : , Max : ,
OnMouseDown	Конфигурация...

Рис. 11.2.12.7. Параметры элемента **Текстовое поле**

Во вкладке [InputConfiguration](#) к параметру **OnClick** привяжем действие [Записать переменную](#):

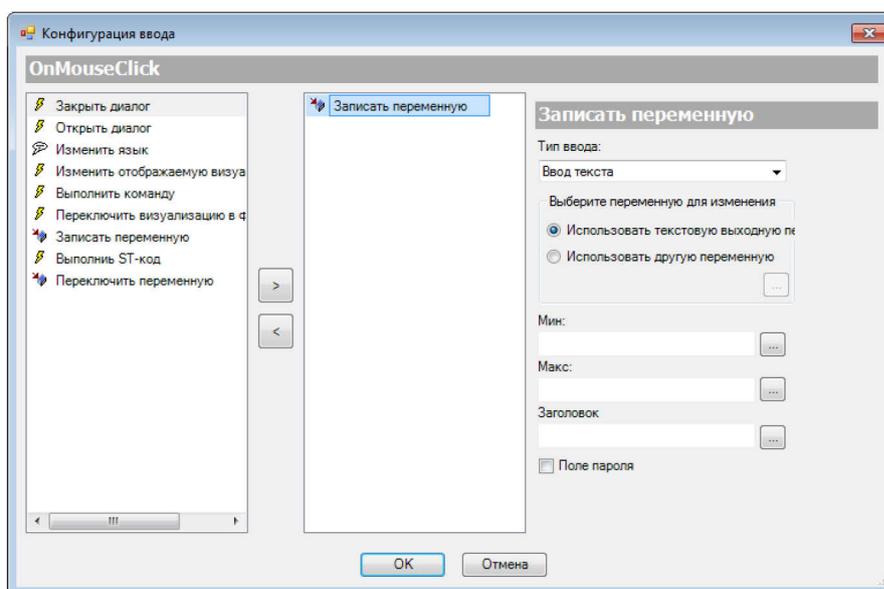


Рис. 11.2.12.8. Настройки действия элемента **Текстовое поле**

К кнопкам привяжем соответствующие переменные:

Inputconfiguration	
OnDialogClosed	Конфигурация...
OnClick	Конфигурация...
OnMouseDown	Конфигурация...
OnMouseEnter	Конфигурация...
OnMouseLeave	Конфигурация...
OnMouseMove	Конфигурация...
OnMouseUp	Конфигурация...
+ Нажать	
- Переключить	
Переменная	PLC_PRG.bOpen
Переключить вверх, есл...	<input type="checkbox"/>
+ Горячая клавиша	

Рис. 11.2.12.9. Параметры кнопки **Открыть**

Inputconfiguration	
OnDialogClosed	Конфигурация...
OnMouseClicked	Конфигурация...
OnMouseDown	Конфигурация...
OnMouseEnter	Конфигурация...
OnMouseLeave	Конфигурация...
OnMouseMove	Конфигурация...
OnMouseUp	Конфигурация...
+ Нажать	
- Переключить	
Переменная	PLC_PRG.bClose
Переключить вверх, есл...	<input type="checkbox"/>
+ Горячая клавиша	

Рис. 11.2.12.10. Параметры кнопки **Закреть**

Inputconfiguration	
OnDialogClosed	Конфигурация...
OnMouseClicked	Конфигурация...
OnMouseDown	Конфигурация...
OnMouseEnter	Конфигурация...
OnMouseLeave	Конфигурация...
OnMouseMove	Конфигурация...
OnMouseUp	Конфигурация...
+ Нажать	
- Переключить	
Переменная	PLC_PRG.bSave
Переключить вверх, есл...	<input type="checkbox"/>
+ Горячая клавиша	

Рис. 11.2.12.11. Параметры кнопки **Сохранить**

Inputconfiguration	
OnDialogClosed	Конфигурация...
OnMouseClicked	Конфигурация...
OnMouseDown	Конфигурация...
OnMouseEnter	Конфигурация...
OnMouseLeave	Конфигурация...
OnMouseMove	Конфигурация...
OnMouseUp	Конфигурация...
+ Нажать	
- Переключить	
Переменная	PLC_PRG.bNew
Переключить вверх, есл...	<input type="checkbox"/>
+ Горячая клавиша	

Рис. 11.2.12.12. Параметры кнопки **Новый**

7. Добавим на экран панель управления параметрами вкладки **Редактировать**. Эти параметры позволяют осуществлять поиск по текстовому файлу.

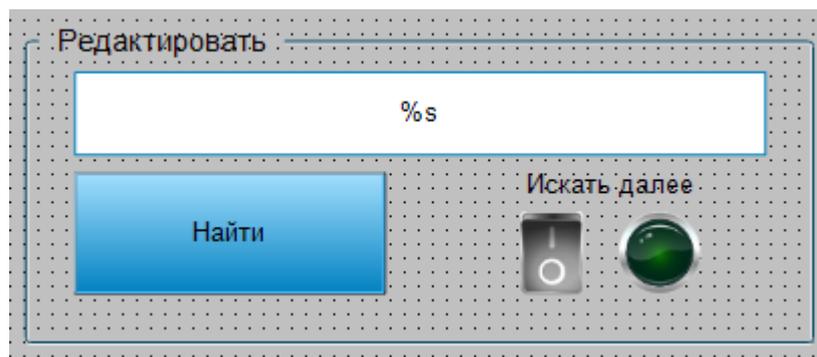


Рис. 11.2.12.13. Панель управления параметрами вкладки **Файл**

Панель представляет собой элемент [Группа](#), в которой расположен элемент [Текстовое поле](#), элемент [Кнопка](#), элемент [Клавишный выключатель](#) и элемент [Индикатор](#). Параметры элементов приведены ниже (не отображенные параметры находятся в значениях по умолчанию).

Свойства	
Свойство	Значения
Имя элемента	GenElemInst_82
Тип элемента	Текстовое поле
ID текста	0
Позиция	
Цвета	
Вид элемента	
Тип тени	Из стиля
Тексты	
Текст	%s
Подсказка	
Свойства текста	
Текстовые переменные	
Текстовая переменная	PLC_PRG.sSearch
Переменная подсказки	
Динамические тексты	
Переменные шрифта	
Переменные цвета	
Переменные состояний	
Inputconfiguration	
OnDialogClosed	Конфигурация...
OnMouseClicked	Конфигурация...
Записать переменную	Variable : , InputType : Edit, Min : , Max :
OnMouseDown	Конфигурация...

Рис. 11.2.12.14. Параметры элемента **Текстовое поле**

Во вкладке [InputConfiguration](#) к параметру **OnClick** привяжем действие [Записать переменную](#):

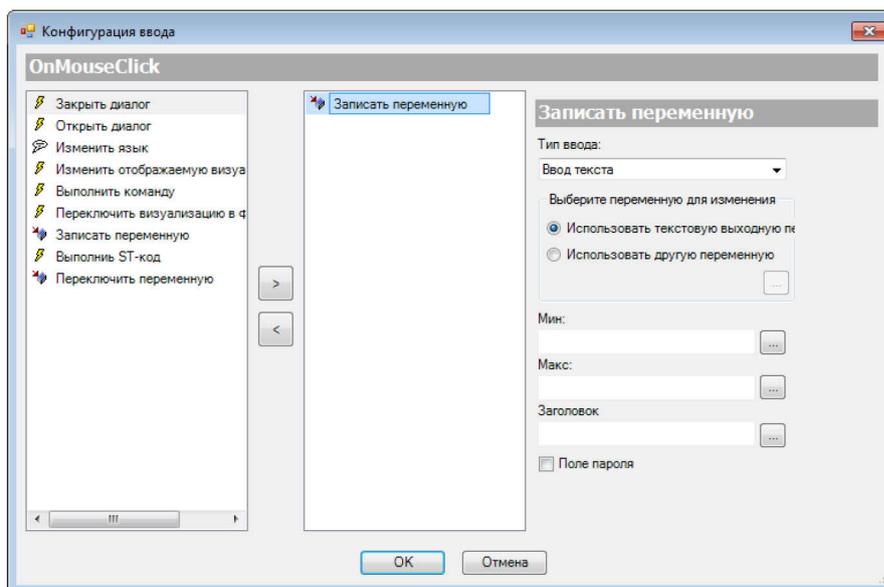


Рис. 11.2.12.15. Настройки действия элемента **Текстовое поле**

К кнопке **Найти** привяжем переменную **bFind**:

Inputconfiguration	
OnDialogClosed	Конфигурация...
OnMouseClicked	Конфигурация...
OnMouseDown	Конфигурация...
OnMouseEnter	Конфигурация...
OnMouseLeave	Конфигурация...
OnMouseMove	Конфигурация...
OnMouseUp	Конфигурация...
Нажать	
Переменная	PLC_PRG.bFind
Переключить на FA...	<input type="checkbox"/>
Переключить на вхо...	<input type="checkbox"/>
Переключить	
Переменная	
Переключить на вер...	<input type="checkbox"/>
Горячая клавиша	

Рис. 11.2.12.16. Параметры кнопки **Найти**

К [клавишному выключателю](#) и [индикатору \(Искать далее\)](#) привяжем переменную **bFindNext**:

Позиция	
X	351
Y	139
Ширина	52
Высота	62
Переменная	PLC_PRG.bFindNext

Рис. 11.2.12.17. Параметры элемента **Клавишный выключатель**
(к индикатору привязана та же переменная)

8. Добавим на экран панель управления параметрами вкладки **Позиция каретки**. Эти параметры управляют перемещением курсора по текстовому файлу.

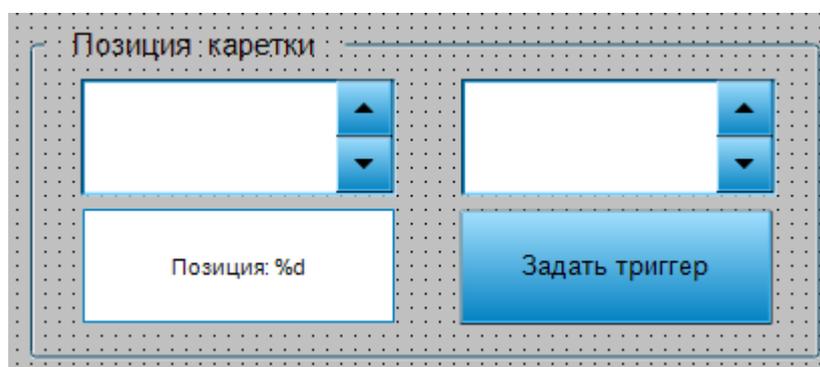


Рис. 11.2.12.18. Панель управления параметрами вкладки **Позиция каретки**

Панель представляет собой элемент [Группа](#), в которой расположены два элемента [Управление вращением](#), элемент [Текстовое поле](#) и элемент [Кнопка](#). Параметры элементов приведены ниже (не отображенные параметры находятся в значениях по умолчанию).

Свойства	
Фильтр	Сортировать по
Порядок сортировки	Эксперт
Свойство	Значения
Имя элемента	GenElemInst_99
Тип элемента	Управление вращением
Позиция	
X	37
Y	44
Ширина	220
Высота	80
Переменная	PLC_PRG.iLine
Числовой формат	Строка: %d
Интервал	1
Value range	
Свойства текста	
Переменные цвета	
Переменные состояний	
Inputconfiguration	

Рис. 11.2.12.19. Параметры элемента **Управление вращением 1**

Свойства	
Фильтр	Сортировать по
Порядок сортировки	Эксперт
Свойство	Значения
Имя элемента	GenElemInst_101
Тип элемента	Управление вращением
Позиция	
X	307
Y	44
Ширина	220
Высота	80
Переменная	PLC_PRG.iColumn
Числовой формат	Столбец: %d
Интервал	1
Value range	
Свойства текста	
Переменные цвета	
Переменные состояний	
Inputconfiguration	

Рис. 11.2.12.20. Параметры элемента **Управление вращением 2**

Свойства	
Фильтр	Сортировать по
<input checked="" type="checkbox"/> Эксперт	Порядок сортировки
Свойство	Значения
Имя элемента	GenElemInst_153
Тип элемента	Текстовое поле
ID текста	47
Позиция	
X	37
Y	136
Ширина	220
Высота	80
Цвета	
Вид элемента	
Тип тени	Из стиля
Тексты	
Текст	Позиция: %d
Подсказка	
Свойства текста	
Текстовые переменные	
Текстовая переменная	PLC_PRG.iPosition
Переменная подсказки	
Динамические тексты	
Переменные шрифта	
Переменные цвета	
Переменные состояний	
Невидимый	
Отключение ввода	
Inputconfiguration	

Рис. 11.2.12.21. Параметры элемента **Текстовое поле**

К кнопке **Задать триггер** привяжем переменную **bTriggerSet**:

Inputconfiguration	
OnDialogClosed	Конфигурация...
OnMouseClicked	Конфигурация...
OnMouseDown	Конфигурация...
OnMouseEnter	Конфигурация...
OnMouseLeave	Конфигурация...
OnMouseMove	Конфигурация...
OnMouseUp	Конфигурация...
Нажать	
Переключить	
Переменная	PLC_PRG.bTriggerSet
Переключить наверх...	<input type="checkbox"/>
Горячая клавиша	

Рис. 11.2.12.22. Параметры элемента **Кнопка (Задать триггер)**

9. Добавим на экран панель управления параметрами вкладки **Выбор**. Эти параметры характеризуют выделенный фрагмент в текстовом файле.

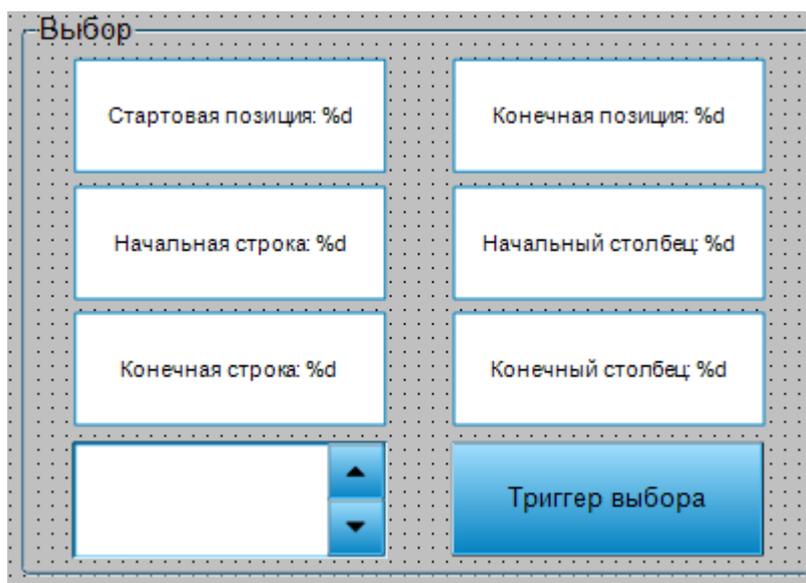


Рис. 11.2.12.23. Панель управления параметрами вкладки **Выбор**

Панель представляет собой элемент [Группа](#), в которой расположены 6 элементов [Текстовое поле](#), элемент [Управление вращением](#) и элемент [Кнопка](#). Параметры элементов приведены ниже (не отображенные параметры находятся в значениях по умолчанию).

Свойство	Значения
Имя элемента	GenElemInst_155
Тип элемента	Текстовое поле
ID текста	48
Позиция	
X	37
Y	39
Ширина	220
Высота	80
Цвета	
Вид элемента	
Тип тени	Из стиля
Тексты	
Текст	Стартовая позиция: %d
Подсказка	
Свойства текста	
Текстовые переменные	
Текстовая переменная	PLC_PRG.iStartPosition
Переменная подсказки	
Динамические тексты	
Переменные шрифта	
Переменные цвета	
Переменные состояний	
Inputconfiguration	

Рис. 11.2.12.24. Параметры элемента **Текстовое поле 1**

Свойства	
Фильтр	Сортировать по
<input checked="" type="checkbox"/> Эксперт	Порядок сортировки
Свойство	Значения
Имя элемента	GenElemInst_157
Тип элемента	Текстовое поле
ID текста	49
Позиция	
X	306
Y	39
Ширина	220
Высота	80
Цвета	
Вид элемента	
Тип тени	Из стиля
Тексты	
Текст	Конечная позиция: %d
Подсказка	
Свойства текста	
Текстовые переменные	
Текстовая переменная	PLC_PRG.iEndPosition
Переменная подсказки	
Динамические тексты	
Переменные шрифта	
Переменные цвета	
Переменные состояний	
Inputconfiguration	

Рис. 11.2.12.25. Параметры элемента **Текстовое поле 2**

Свойства	
Фильтр	Сортировать по
<input checked="" type="checkbox"/> Эксперт	Порядок сортировки
Свойство	Значения
Имя элемента	GenElemInst_159
Тип элемента	Текстовое поле
ID текста	50
Позиция	
X	37
Y	130
Ширина	220
Высота	80
Цвета	
Вид элемента	
Тип тени	Из стиля
Тексты	
Текст	Начальная строка: %d
Подсказка	
Свойства текста	
Текстовые переменные	
Текстовая переменная	PLC_PRG.iStartLineNumber
Переменная подсказки	
Динамические тексты	
Переменные шрифта	
Переменные цвета	
Переменные состояний	
Inputconfiguration	

Рис. 11.2.12.26. Параметры элемента **Текстовое поле 3**

Свойства	
Свойство	Значения
Имя элемента	GenElemInst_161
Тип элемента	Текстовое поле
ID текста	51
Позиция	
X	306
Y	130
Ширина	220
Высота	80
Цвета	
Вид элемента	
Тип тени	Из стиля
Тексты	
Текст	Начальный столбец: %d
Подсказка	
Свойства текста	
Текстовые переменные	
Текстовая переменная	PLC_PRG.iStartColumnIndex
Переменная подсказки	
Динамические тексты	
Переменные шрифта	
Переменные цвета	
Переменные состояний	
Inputconfiguration	

Рис. 11.2.12.27. Параметры элемента **Текстовое поле 4**

Свойства	
Свойство	Значения
Имя элемента	GenElemInst_163
Тип элемента	Текстовое поле
ID текста	52
Позиция	
X	37
Y	220
Ширина	220
Высота	80
Цвета	
Вид элемента	
Тип тени	Из стиля
Тексты	
Текст	Конечная строка: %d
Подсказка	
Свойства текста	
Текстовые переменные	
Текстовая переменная	PLC_PRG.iEndLineNumber
Переменная подсказки	
Динамические тексты	
Переменные шрифта	
Переменные цвета	
Переменные состояний	
Inputconfiguration	

Рис. 11.2.12.28. Параметры элемента **Текстовое поле 5**

Свойства	
<input type="checkbox"/> Фильтр <input type="checkbox"/> Сортировать по <input type="checkbox"/> Порядок сортировки <input checked="" type="checkbox"/> Эксперт	
Свойство	Значения
Имя элемента	GenElemInst_165
Тип элемента	Текстовое поле
ID текста	53
<input type="checkbox"/> Позиция	
X	306
Y	220
Ширина	220
Высота	80
<input type="checkbox"/> Цвета	
<input type="checkbox"/> Вид элемента	
Тип тени	Из стиля
<input type="checkbox"/> Тексты	
Текст	Конечный столбец: %d
Подсказка	
<input type="checkbox"/> Свойства текста	
<input type="checkbox"/> Текстовые переменные	
Текстовая переменная	PLC_PRG.iEndColumnIndex
Переменная подсказки	
<input type="checkbox"/> Динамические тексты	
<input type="checkbox"/> Переменные шрифта	
<input type="checkbox"/> Переменные цвета	
<input type="checkbox"/> Переменные состояний	
<input type="checkbox"/> Inputconfiguration	

Рис. 11.2.12.29. Параметры элемента **Текстовое поле 6**

Свойства	
<input type="checkbox"/> Фильтр <input type="checkbox"/> Сортировать по <input type="checkbox"/> Порядок сортировки <input checked="" type="checkbox"/> Эксперт	
Свойство	Значения
Имя элемента	GenElemInst_103
Тип элемента	Управление вращением
<input type="checkbox"/> Позиция	
X	37
Y	314
Ширина	220
Высота	80
Переменная	PLC_PRG.iLineToSelect
Числовой формат	Выбрать линию: %d
Интервал	1
<input type="checkbox"/> Value range	
<input type="checkbox"/> Свойства текста	
<input type="checkbox"/> Переменные цвета	
<input type="checkbox"/> Переменные состояний	
<input type="checkbox"/> Inputconfiguration	

Рис. 11.2.12.30. Параметры элемента **Управление вращением**

К кнопке **Триггер выбора** привяжем переменную **bTriggerSelect**:

[-] Inputconfiguration		
OnDialogClosed	Конфигурация...	
OnMouseClicked	Конфигурация...	
OnMouseDown	Конфигурация...	
OnMouseEnter	Конфигурация...	
OnMouseLeave	Конфигурация...	
OnMouseMove	Конфигурация...	
OnMouseUp	Конфигурация...	
[-] Нажать		
Переменная		
Переключить на FA...		<input type="checkbox"/>
Переключить на вхо...		<input type="checkbox"/>
[-] Переключить		
Переменная	PLC_PRG.bTriggerSelect	
Переключить на вер...		<input type="checkbox"/>
[+] Горячая клавиша		

Рис. 11.2.12.31. Параметры элемента **Кнопка (Триггер выбора)**

10. Добавим на экран панель управления остальными параметрами элемента.

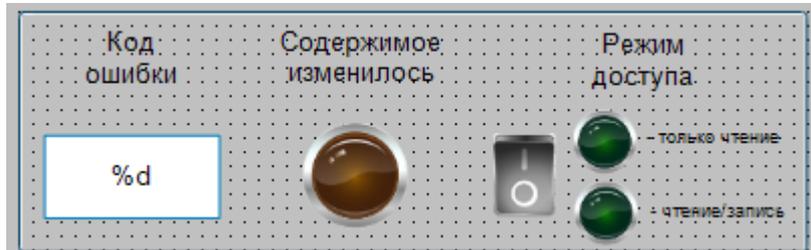


Рис. 11.2.12.32. Панель управления остальными параметрами элемента

Панель представляет собой элемент [Группа](#), в которой расположен элемент [Текстовое поле](#), три элемента [Индикатор](#) и элемент [Клавишный переключатель](#). Пять поясняющих надписей сделаны при помощи элемента [Метка](#). Параметры элементов приведены ниже (не отображенные параметры находятся в значениях по умолчанию).

Свойства	
Фильтр	Сортировать по
Порядок сортировки	Эксперт
Свойство	Значения
Имя элемента	GenElemInst_119
Тип элемента	Текстовое поле
ID текста	1
Позиция	
X	18
Y	106
Ширина	124
Высота	59
Цвета	
Вид элемента	
Тип тени	Из стиля
Тексты	
Текст	%d
Подсказка	
Свойства текста	
Текстовые переменные	
Текстовая переменная	PLC_PRG.iErrorCode
Переменная подсказки	

Рис. 11.2.12.33. Параметры элемента **Текстовое поле**

Свойства	
Фильтр	Сортировать по
Порядок сортировки	Эксперт
Свойство	Значения
Имя элемента	GenElemInst_133
Тип элемента	Индикатор
Позиция	
X	201
Y	97
Ширина	85
Высота	77
Переменная	PLC_PRG.bContentChanged
Параметры изображения	
Прозрачный	<input type="checkbox"/>
Прозрачный цвет	Black
Изотропный тип	Изотропия
Горизонтальное выравн...	Лево
Вертикальное выравнив...	Верх
Тексты	
Переменные состояний	
Фон	
Изображение	Yellow

Рис. 11.2.12.34. Параметры индикатора **Содержимое изменилось**

Свойства	
Ү Фильтр Сортировать по Z↓ Порядок сортировки Эксперт	
Свойство	Значения
Имя элемента	GenElemInst_137
Тип элемента	Клавишный выключатель
[-] Позиция	
X	336
Y	104
Ширина	52
Высота	62
Переменная	PLC_PRG.bAccessMode
[-] Параметры изображения	
Прозрачный	<input type="checkbox"/>
Прозрачный цвет	 Black
Изотропный тип	Изотропия
Горизонтальное выравн...	Лево
Вертикальное выравнив...	Верх
Поведение элемента	Переключатель изображения
[-] Тексты	
Подсказка	
[-] Переменные состояний	
Невидимый	
Отключение ввода	
[-] Фон	
Изображение	Gray

Рис. 11.2.12.35. Параметры элемента **Клавишный выключатель**

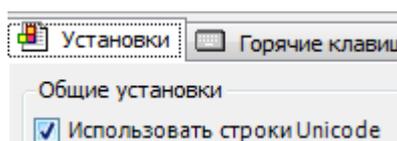
Свойства	
̣ Фильтр ̣ Сортировать по ̣ Порядок сортировки ̣ Эксперт	
Свойство	Значения
Имя элемента	GenElemInst_135
Тип элемента	Индикатор
[-] Позиция	
X	393
Y	88
Ширина	50
Высота	47
Переменная	PLC_PRG.bAccessMode
[-] Параметры изображения	
Прозрачный	<input type="checkbox"/>
Прозрачный цвет	Black
Изотропный тип	Изотропия
Горизонтальное выравн...	Лево
Вертикальное выравнив...	Верх
[+] Тексты	
[+] Переменные состояний	
[-] Фон	
Изображение	Green

Рис. 11.2.12.36. Параметры индикатора **Только чтение**

Свойства	
̣ Фильтр ̣ Сортировать по ̣ Порядок сортировки ̣ Эксперт	
Свойство	Значения
Имя элемента	GenElemInst_131
Тип элемента	Индикатор
[-] Позиция	
X	394
Y	139
Ширина	50
Высота	47
Переменная	PLC_PRG.bAccessModeRev
[-] Параметры изображения	
Прозрачный	<input type="checkbox"/>
Прозрачный цвет	Black
Изотропный тип	Изотропия
Горизонтальное выравн...	Лево
Вертикальное выравнив...	Верх
[+] Тексты	
[+] Переменные состояний	
[-] Фон	
Изображение	Green

Рис. 11.2.12.37. Параметры индикатора **Чтение/запись**

11. Настроим [Менеджер визуализации](#). В его установках поставим галочку **Использовать строки Unicode**:



Настройки **target**- и **web**-визуализации будут выглядеть следующим образом:

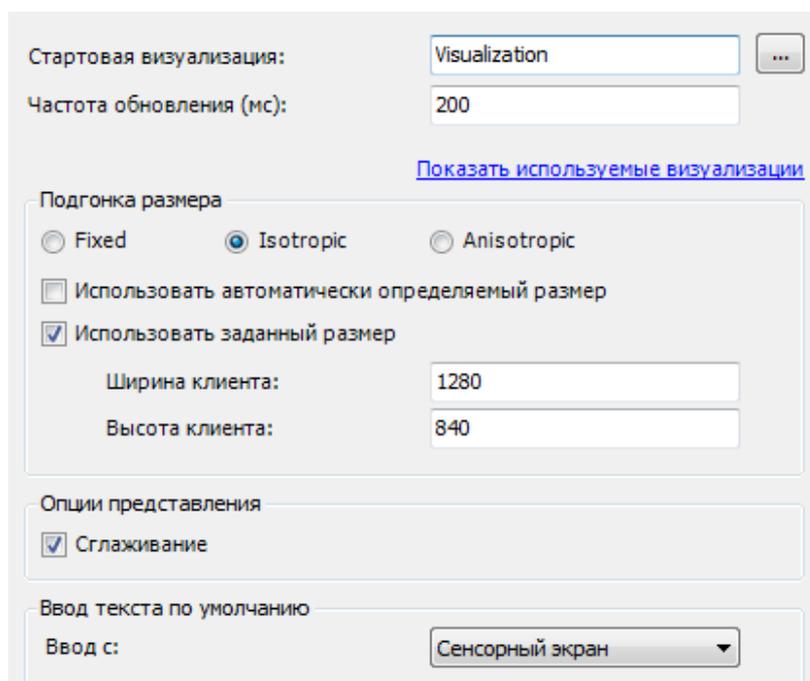


Рис. 11.2.12.38. Настройки **target**-визуализации

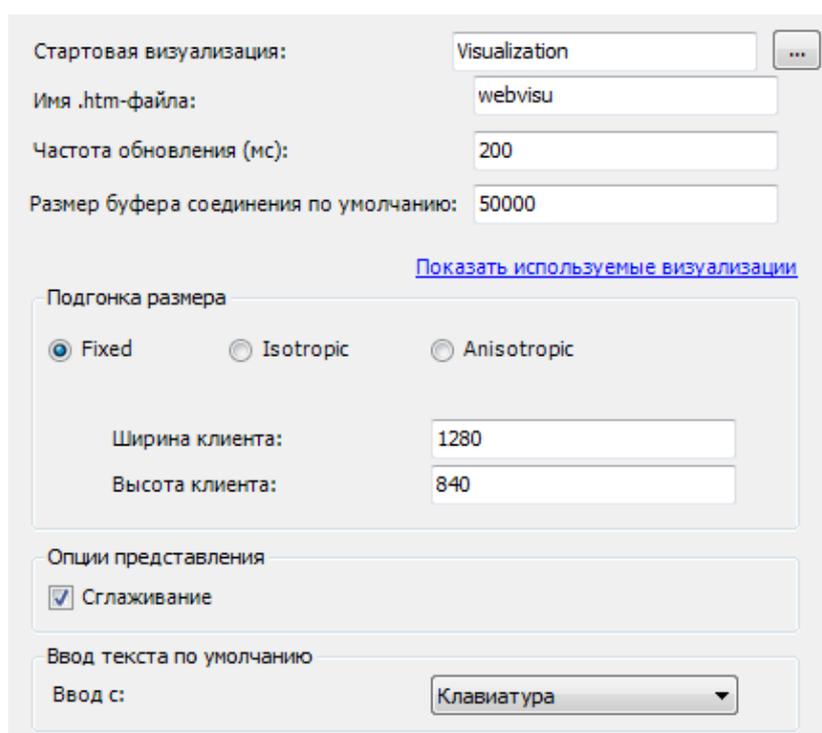


Рис. 11.2.12.39. Настройки **web**-визуализации

12. [Запустим проект на виртуальном контроллере](http://localhost:8080/webvisu.htm). Откроем в веб-браузере страницу <http://localhost:8080/webvisu.htm>, чтобы увидеть web-визуализацию проекта. **Обратите внимание**, что браузер должен поддерживать **HTML5**. Проверим функционал проекта.

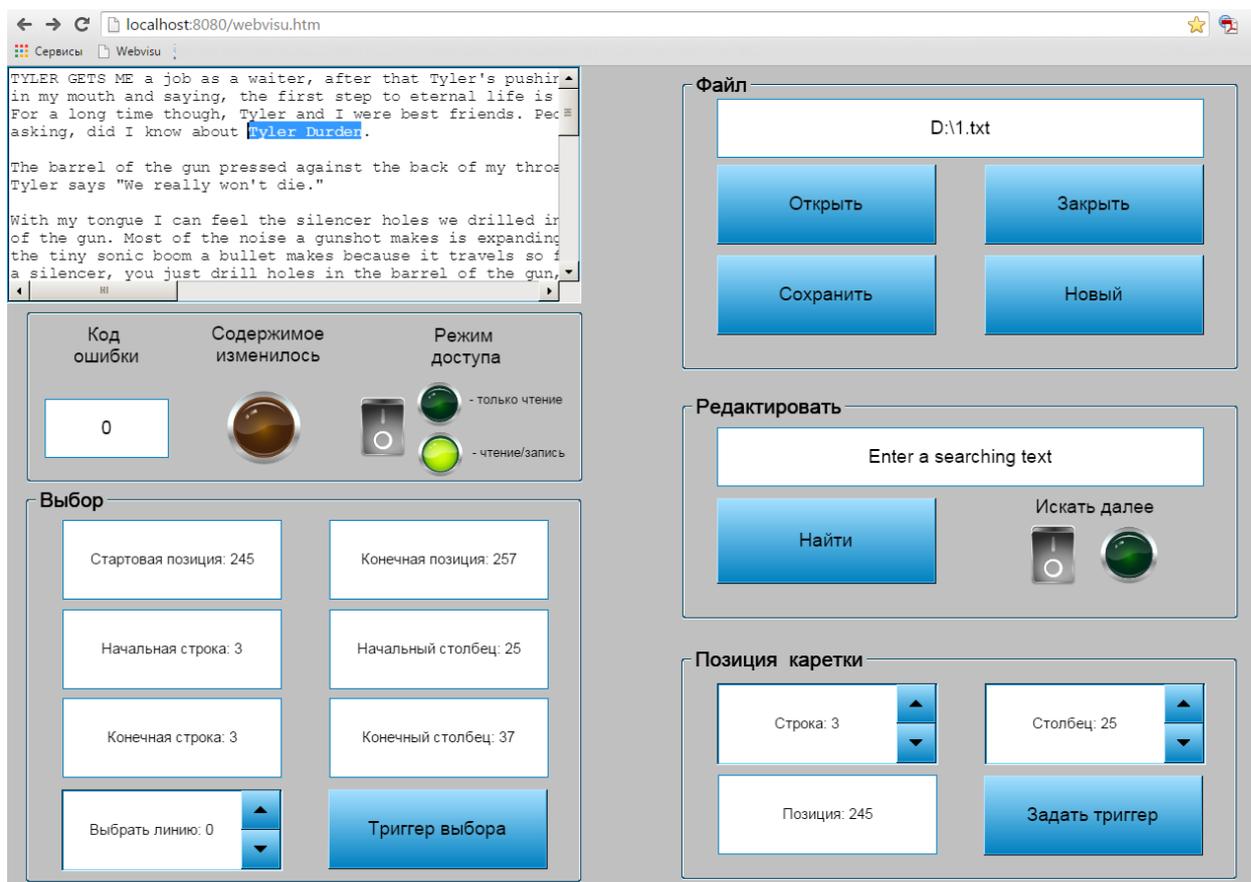


Рис. 11.2.12.40. Web-визуализация проекта в браузере **Chrome**

На панели **Файл** укажите путь к файлу. Путь не должен содержать кириллических символов. Если файл содержит текст на русском языке, то должен быть сохранен в кодировке **Юникод**. **Обратите внимание** на ограничение длины строк в файле (табл. 29, пп. 2).

Нажатие на кнопку **Открыть** откроет выбранный файл в текстовом редакторе. Нажатие на кнопку **Закреть** – закроет. Нажатие на кнопку **Сохранить** – сохранит изменения в данном файле. Нажатие на кнопку **Новый** приведет к созданию нового пустого текстового файла.

Во вкладке **Редактировать** введите текстовый фрагмент, поиск которого необходимо произвести в файле. Поиск происходит после нажатия на кнопку **Найти**. В режиме **Искать далее** каждое нажатие на кнопку **Найти** осуществляет переход к следующему найденному фрагменту.

Во вкладке **Позиция каретки** отображается информация о текущем положении курсора. После ввода новых значений параметров **Строка** и **Столбец**, можно переместить курсор нажатием кнопки **Задать триггер**. Параметр **Позиция** является неизменяемым.

Во вкладке **Выбор** отображается информация о текущем выделенном текстовом фрагменте. Параметры являются неизменяемыми. Выберите номер линии и нажмите кнопку **Триггер выбора**, чтобы выделить одну из строк файла.

В поле **Код ошибки** отображается код последней ошибки, возникшей при работе с файлом. Например, при попытке открыть несуществующий файл, в поле отобразится код 111. Коды ошибок приведены в списке **GVL_ErrorCodes** библиотеки **VisuElemTextEditor**.

По умолчанию редактор находится в режиме **чтение/запись**, что позволяет редактировать содержимое файла. Удаление символа, расположенного за курсором, осуществляется с помощью нажатия клавиши **Del**. Переход на следующую строку осуществляется нажатием клавиши **Enter**. После внесения изменений в файл загорится индикатор **Содержимое изменилось**. Он потухнет после сохранения файла. С помощью переключателя можно перевести редактор в режим **Только чтение**, в котором пользователь не может редактировать файлы.

11.2.13. Таблица тревог

Данный пример посвящен работе с графическим примитивом [Таблица тревог](#), используемым для отображения состояний аварийных сигналов (тревог) с возможностью просмотра истории их срабатываний.

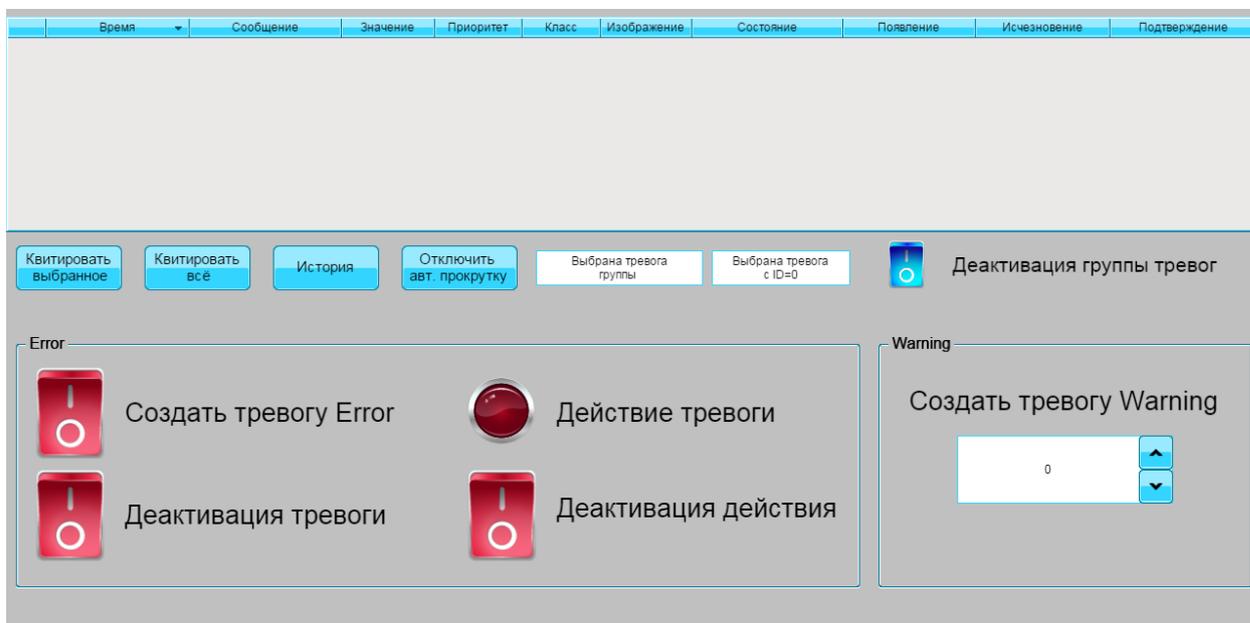


Рис. 11.2.13.1. Внешний вид примера **Таблица тревог**

Пример создан в среде **CODESYS 3.5 SP5 Patch5** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с **target-файлом 3.5.5.0**.

Пример доступен для скачивания: Example_AlarmTable.projectarchive

1. Создадим новый **стандартный** проект **CODESYS** с названием **Example_AlarmTable** и настройками по умолчанию: target – **CODESYS Control Win V3**, язык PLC_PRG – **ST**.

2. В программе **PLC_PRG** объявим следующие переменные:

```
1 PROGRAM PLC_PRG
2 VAR
3     bAlarm:BOOL;           // переменная тревоги типа BOOL (класс Error)
4     bAlarmDeactivate:BOOL; // деактивации тревоги
5
6     bAlarmLamp:BOOL;       // переменная действия тревоги
7     bAlarmLampOff:BOOL;    // деактивация действия тревоги
8
9     bAlarmGroupDeactivate:BOOL; // деактивация группы тревог
10
11     iAlarm:INT;           // переменная тревоги типа INT (класс Warning)
12
13     sSelectedAlarmGroupName:STRING; // имя группы, тревога которой выбрана в таблице в данный момент
14     uiSelectedAlarmID:UINT; // ID тревоги, выбранной в таблице в данный момент
15 END_VAR
```

Рис. 11.2.13.2. Объявление переменных программы **PLC_PRG**

3. Добавим в проект компонент **Конфигуратор тревог (Alarm Configuration)**.

Предварительно потребуется переключить язык интерфейса на **английский**, поскольку в русскоязычной версии добавление компонента происходит **некорректно** (в версии 3.5 SP6):

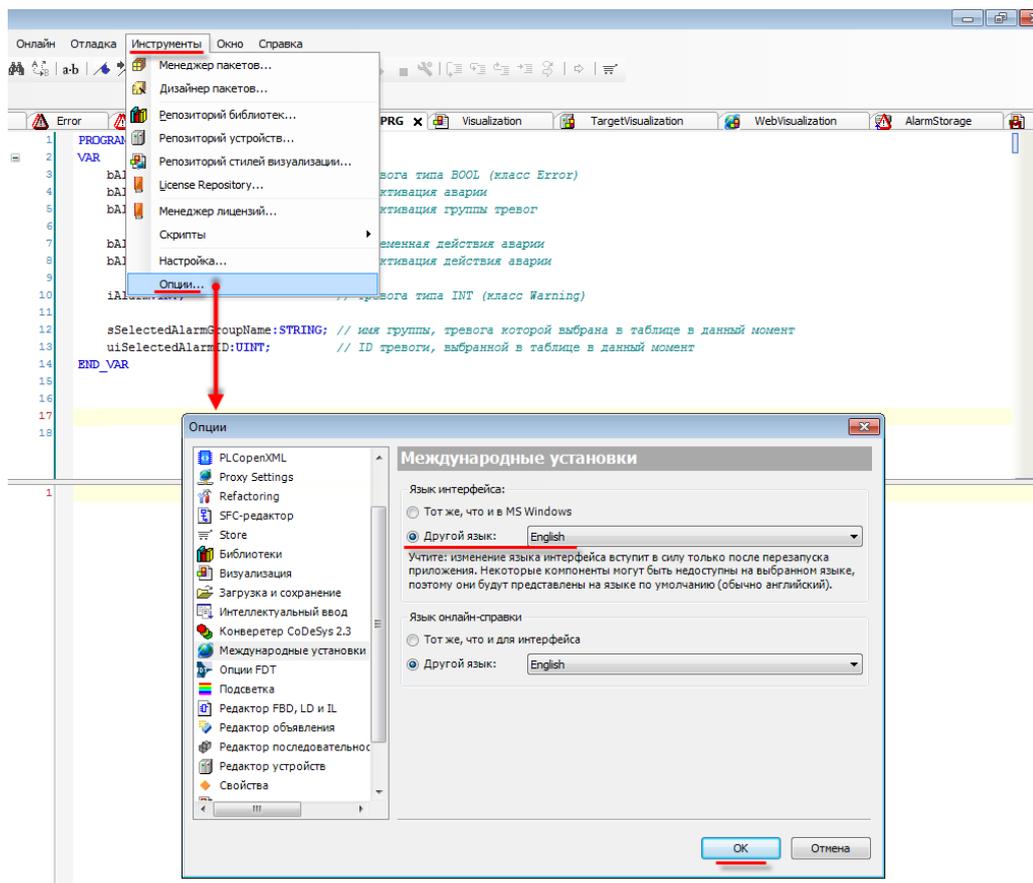


Рис. 11.2.13.3. Переключение языка интерфейса **CODESYS**

После этого потребуется перезапустить **CODESYS**.

4. Добавим в проект компонент [Alarm Configuration](#):

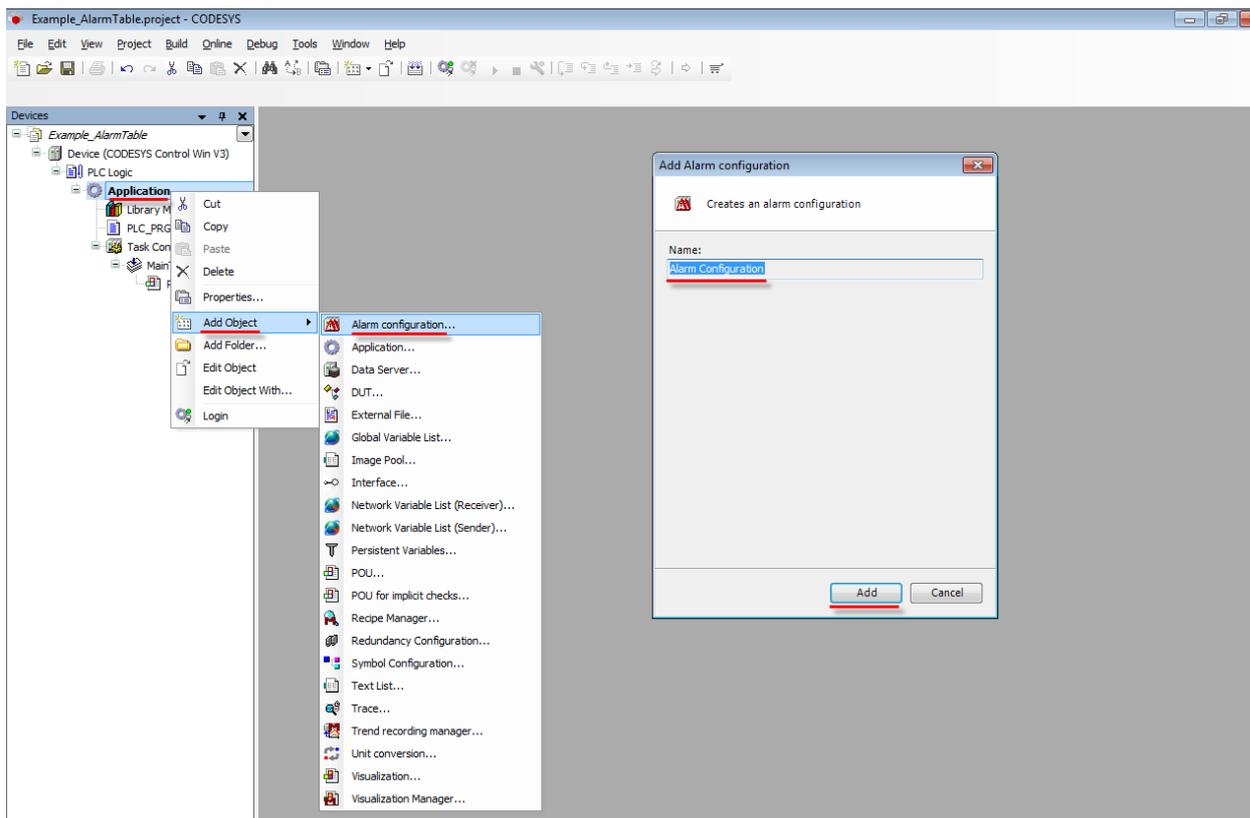


Рис. 11.2.13.4. Добавление компонента Alarm Configuration

Панель устройств после добавления компонента будет выглядеть следующим образом:

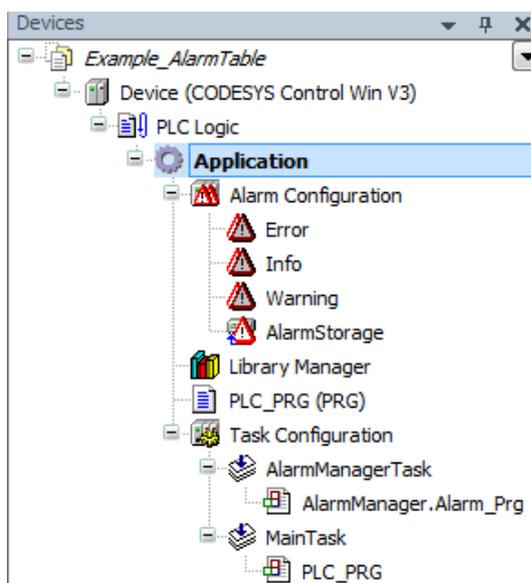


Рис. 11.2.13.5. Панель устройств после добавление компонента **Alarm Configuration**

Теперь можно переключить язык интерфейса **CODESYS** обратно на русский.

5. Создадим в [Alarm Configuration](#) группу тревог с названием **Alarm Group**. Автоматически будет создан [СПИСОК ТЕКСТОВ](#) с тем же названием.

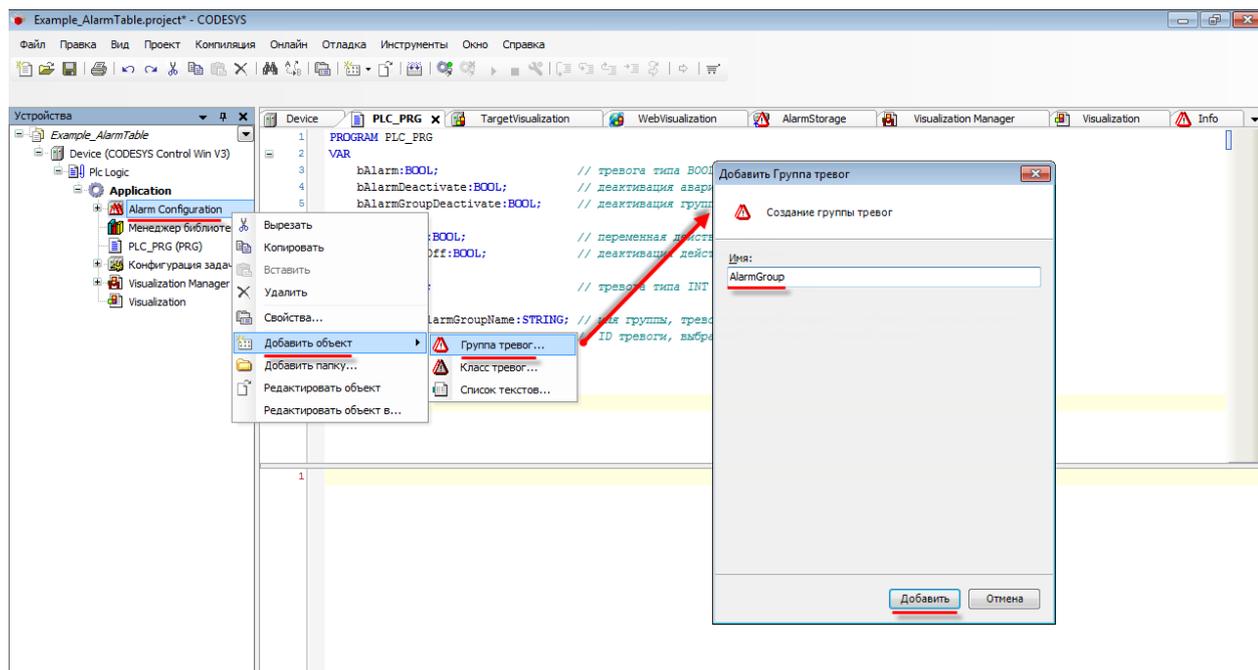


Рис. 11.2.13.6. Создание группы тревог **AlarmGroup**

Удалим класс тревог **Info**, поскольку он не будет использоваться в нашем проекте:

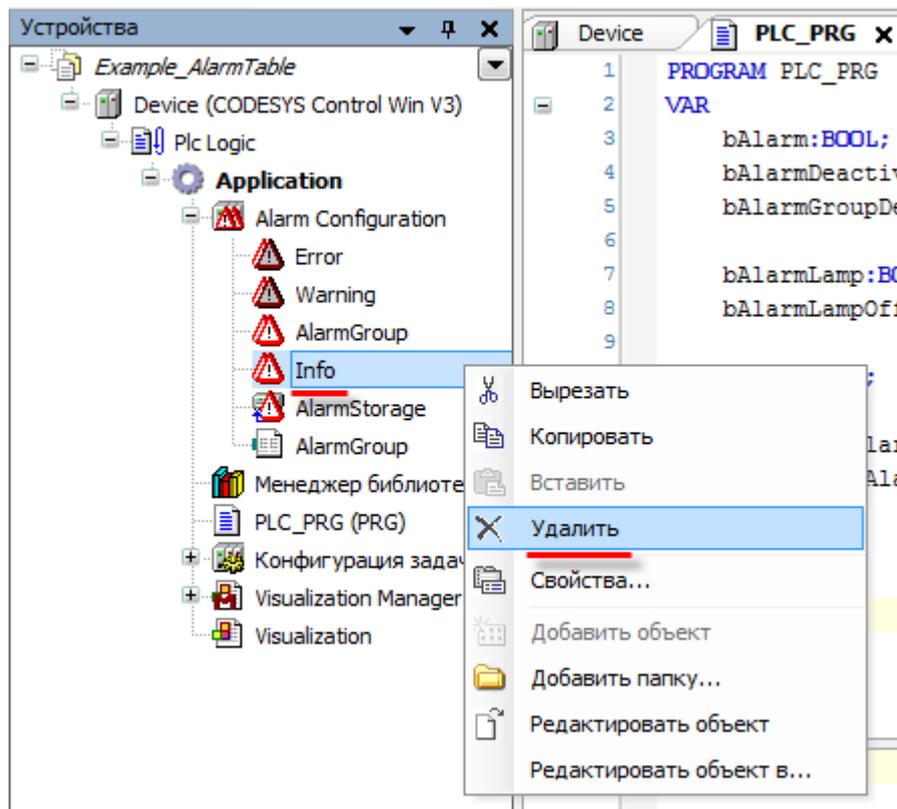


Рис. 11.2.13.7. Удаление класса тревог **Info**

6. Настроим класс тревог **Error**:

Приоритет: 10

Подтверждение
Способ подтверждения: REP_ACK

Архивация

подтверждать по отдельности

Действия уведомления

Действие	активация	деактивация	подтвердить	Детали	Деактивация
Переменная	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	PLC_PRG.bAlarmLamp := TRUE	PLC_PRG.bAlarmLampOff
Щелкните здесь ...				Щелкните здесь, чтобы добавить...	

Детали

Отображать опции из таблицы тревог/баннера тревог

Состояние	Шрифт	Цвет фона	Изображение	Прозрачность	Прозрачный цвет
Активное		220; 20; 60	alarm	<input type="checkbox"/>	
Ожидание подтверждения		255; 140; 0		<input type="checkbox"/>	

Рис. 11.2.13.8. Настройки класса тревог **Error**

Для добавления изображения введите любое название, нажмите **Enter** и укажите путь к графическому файлу.

7. Настроим класс тревог **Warning**:

Приоритет: 30

Подтверждение
Способ подтверждения: REP

Архивация

подтверждать по отдельности

Действия уведомления

Действие	активация	деактивация	Детали	Деактивация
Щелкните здесь, ...			Щелкните здесь, чтобы добавить...	

Детали

Отображать опции из таблицы тревог/баннера тревог

Состояние	Шрифт	Цвет фона	Изображение	Прозрачность	Прозрачный цвет
Активное		240; 230; 140	warning	<input type="checkbox"/>	

Рис. 11.2.13.9. Настройки класса тревог **Warning**

8. Настроим группу тревог Alarm Group:

The screenshot displays the configuration interface for an Alarm Group. At the top, there are tabs for 'AlarmGroup', 'PLC_PRG', and 'Visualization'. Below the tabs, there are dropdown menus for 'Список текстов: AlarmGroup', 'Архивация: AlarmStorage', and 'Деактивация: PLC_PRG.bAlarmGroupDeactivate'. The main part of the interface is a table with the following columns: ID, Способ наблюдения, Детали, Деактивация, Класс, Сообщение, and Мин. время ожидания.

ID	Способ наблюдения	Детали	Деактивация	Класс	Сообщение	Мин. время ожидания
0	Дискретный	PLC_PRG.bAlarm = TRUE	PLC_PRG.bAlarmDeactivate	Error	Авария	
1	Верхний предел	PLC_PRG.iAlarm > 10		Warning	Тревога	

Below the table, there is a configuration panel for the selected alarm 'Верхний предел'. It contains the following fields:

- Выражены: PLC_PRG.iAlarm
- Отклонение в: 20
- Value field: 10

Рис. 11.2.13.10. Настройки группы тревог AlarmGroup

Настройка деталей каждой тревоги происходит на панели, расположенной под компонентом.

9. [Добавим в проект](#) экран визуализации **Visualization**. В его [свойствах](#) выберем размер **1280x640**. Экран будет содержать элемент [Таблица тревог](#), четыре элемента [Кнопка](#) (созданы автоматически в контекстном меню таблицы), два элемента [Текстовое поле](#), два элемента [Группа](#), четыре элемента [Клавишный выключатель](#), элемент [Переключатель изображения](#) и элемент [Управление вращением](#). Шесть пояснительных текстовых надписей сделаны с помощью элементов [Метка](#).

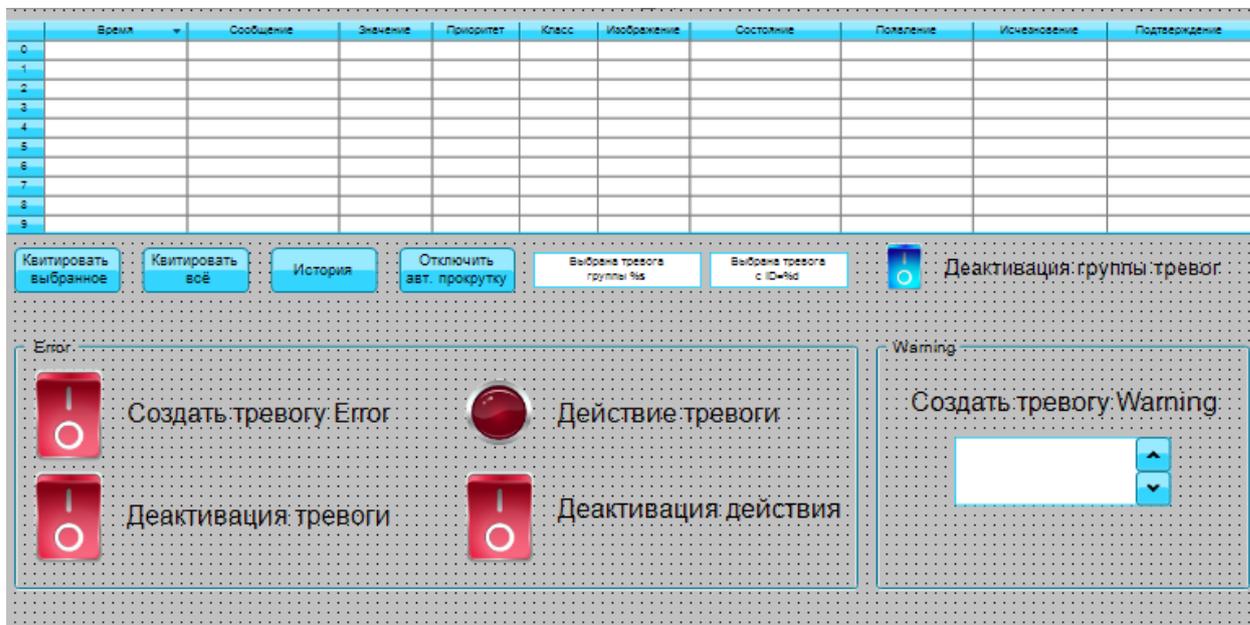


Рис. 11.2.13.11. Содержание экрана **Visualization**

Настройки элементов приведены ниже.

10. Настроим элемент [Таблица тревог](#):

Свойства	
Фильтр	Сортировать по
Порядок сортировки	Эксперт
Свойство	Значения
Имя элемента	GenElemInst_10
Тип элемента	Таблица тревог
Приложение	Device.Application
Конфигурация тревоги	
Группы тревог	⚠ Все
Приоритет из	0
Приоритет для	255
Классы тревог	⚠ Все
Общая конфигурация таблицы	
Показать заголовок строки	<input checked="" type="checkbox"/>
Показать заголовок столбца	<input checked="" type="checkbox"/>
Высота строки	20
Ширина строки заголовка	40
Размер полосы прокрутки	20
Автоматический разрыв строки в сообщениях тревог	<input checked="" type="checkbox"/>

Рис. 11.2.13.12. Настройки **Таблицы тревог** (начало)

Столбцы таблицы создаются с помощью кнопки **Создать новый** во вкладке **Столбцы**:

Свойство	Значения
Столбцы	Создать новый
Столбец	
[0]	Удалить
Заголовок столбца	Время
Использовать выравнивание текста в заголовке	<input type="checkbox"/>
Ширина	146
Тип данных	Временная отметка
Выравнивание текста	По центру
[1]	Удалить
Заголовок столбца	Сообщение
Использовать выравнивание текста в заголовке	<input type="checkbox"/>
Ширина	155
Тип данных	Сообщение
Выравнивание текста	Лево
[2]	Удалить
Заголовок столбца	Значение
Использовать выравнивание текста в заголовке	<input type="checkbox"/>
Ширина	95
Тип данных	Значение
Выравнивание текста	По центру
[3]	Удалить
Заголовок столбца	Приоритет
Использовать выравнивание текста в заголовке	<input type="checkbox"/>
Ширина	89
Тип данных	Приоритет
Выравнивание текста	По центру
[4]	Удалить
Заголовок столбца	Класс
Использовать выравнивание текста в заголовке	<input type="checkbox"/>
Ширина	80
Тип данных	Класс
Выравнивание текста	По центру
[5]	Удалить
Заголовок столбца	Изображение
Использовать выравнивание текста в заголовке	<input type="checkbox"/>
Ширина	94
Тип данных	Изображение
Выравнивание текста	По центру
[6]	Удалить
Заголовок столбца	Состояние
Использовать выравнивание текста в заголовке	<input type="checkbox"/>
Ширина	154
Тип данных	Состояние
Выравнивание текста	По центру
[7]	Удалить
Заголовок столбца	Появление
Использовать выравнивание текста в заголовке	<input type="checkbox"/>
Ширина	135
Тип данных	Временная отметка активна
Выравнивание текста	По центру
[8]	Удалить
Заголовок столбца	Исчезновение
Использовать выравнивание текста в заголовке	<input type="checkbox"/>
Ширина	137
Тип данных	Временная отметка неактивна
Выравнивание текста	По центру
[9]	Удалить
Заголовок столбца	Подтверждение
Использовать выравнивание текста в заголовке	<input type="checkbox"/>
Ширина	150
Тип данных	Подтверждение временной отме...
Выравнивание текста	По центру

Рис. 11.2.13.13. Настройки **Таблицы тревог** (продолжение)

Позиция	
X	0
Y	10
Ширина	1280
Высота	220
Свойства текста	
Шрифт	Font-Standard
Цвет шрифта	Font-Default-Color
Выбор	
Цвет выбора	157; 169; 225
Фрейм вокруг выбранных ячеек	<input checked="" type="checkbox"/>
Переменная для выбранной группы тревог	PLC_PRG.sSelectedAlarmGroupNa...
Переменная для выбранного ID тревог	PLC_PRG.uiSelectedAlarmID
Управляющие переменные	
Подтвердить выбранное	bAckSel
Подтвердить все видимое	bAckVisible
История	bHistory
Фиксировать позицию прокрутки	bFreezeScrlPos

Рис. 11.2.13.14. Настройки **Таблицы тревог** (окончание)

Управляющие переменные таблицы и их кнопки можно создать автоматически; для этого необходимо нажать на таблицу ПКМ и в контекстном меню выбрать пункт **Вставить элементы для подтверждения тревог**:

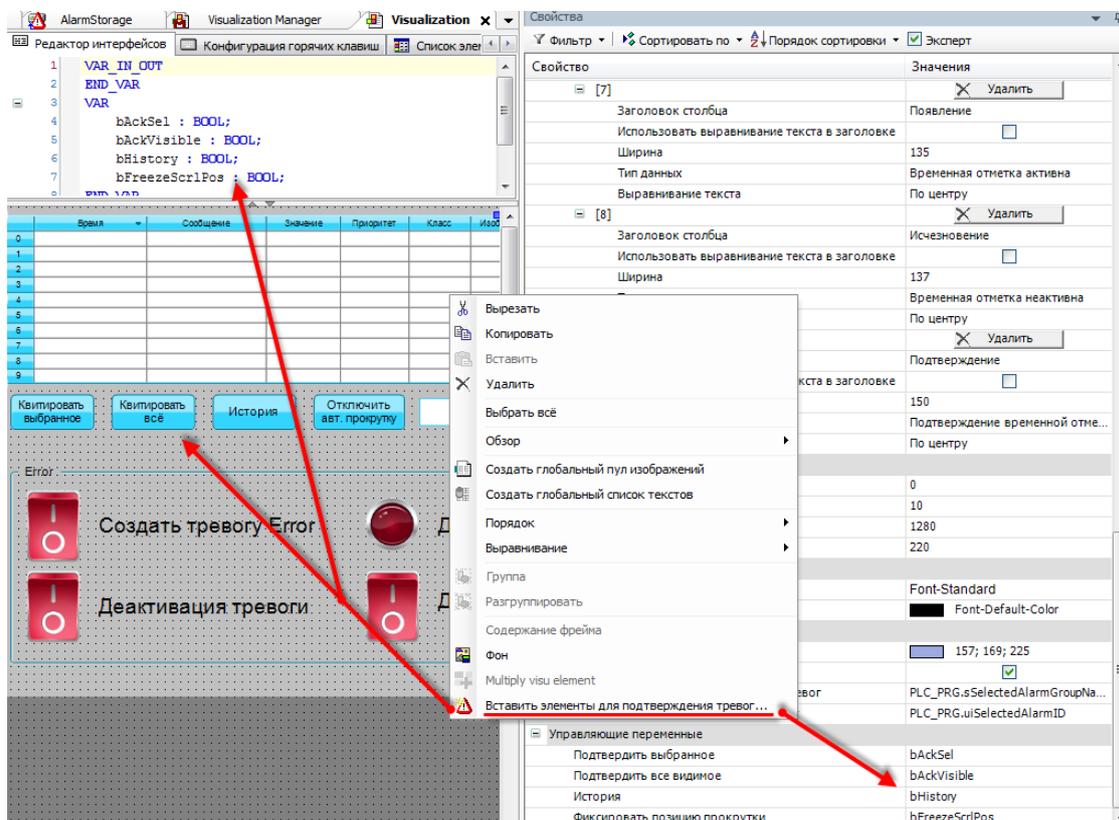


Рис. 11.2.13.15. Создание кнопок подтверждения тревог

11. У кнопок **История** и **Отключить авт.прокрутку** привяжем переменные состояния (они будут соответствовать переключаемым переменным), чтобы видеть, когда активны соответствующие режимы:

[-] Переменная состояния кнопки	
Двоичная переменная	bHistory
[+] Переменная ID изображения	
[-] Inputconfiguration	
OnDialogClosed	Конфигурация...
OnMouseClicked	Конфигурация...
OnMouseDown	Конфигурация...
OnMouseEnter	Конфигурация...
OnMouseLeave	Конфигурация...
OnMouseMove	Конфигурация...
OnMouseUp	Конфигурация...
[-] Нажать	
Переменная	
Переключить на FALSE	<input type="checkbox"/>
Переключить на вход, есл...	<input type="checkbox"/>
[-] Переключить	
Переменная	bHistory
Переключить наверх, есл...	<input type="checkbox"/>
[+] Горячая клавиша	

Рис. 11.2.13.16. Настройки кнопки **История**

Для кнопки **Отключить авт. прокрутку** двоичной переменной, соответственно, будет **bFreezeScrlPos**.

12. Настроим элементы [Текстовое поле](#), используемые для вывода информации о выделенной строке таблицы:



Рис. 11.245. Элементы **Текстовое поле 1** и **2**

Свойства	
Свойство	Значения
Имя элемента	GenElemInst_55
Тип элемента	Текстовое поле
ID текста	33
Позиция	
X	540
Y	250
Ширина	170
Высота	35
Цвета	
Вид элемента	
Тип тени	Из стиля
Тексты	
Текст	Выбрана тревога группы %s
Подсказка	
Свойства текста	
Текстовые переменные	
Текстовая переменная	PLC_PRG.sSelectedAlarmGroupName
Переменная подсказки	

Рис. 11.2.13.17. Настройки элемента **Текстовое поле 1**

Свойства	
Свойство	Значения
Имя элемента	GenElemInst_57
Тип элемента	Текстовое поле
ID текста	34
Позиция	
X	720
Y	250
Ширина	140
Высота	35
Цвета	
Вид элемента	
Тип тени	Из стиля
Тексты	
Текст	Выбрана тревога с ID=%d
Подсказка	
Свойства текста	
Текстовые переменные	
Текстовая переменная	PLC_PRG.uiSelectedAlarmID
Переменная подсказки	

Рис. 11.2.13.18. Настройки элемента **Текстовое поле 2**

13. К [клавишному переключателю Деактивация таблицы](#) привяжем переменную `bAlarmGroupDeactivate`:

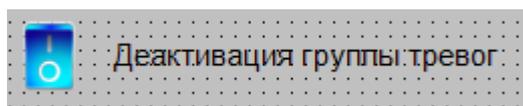


Рис. 11.2.13.19. Переключатель деактивации группы тревог

Свойства	
Ƴ Фильтр Сортировать по Порядок сортировки Эксперт	
Свойство	Значения
Имя элемента	GenElemInst_22
Тип элемента	Клавишный выключатель
[-] Позиция	
X	900
Y	240
Ширина	50
Высота	50
Переменная	PLC_PRG.bAlarmGroupDeactivate
[+] Параметры изображения	
Поведение элемента	Переключатель изображения
[+] Тексты	
[+] Переменные состояний	
[-] Фон	
Изображение	Blue

Рис. 11.2.13.20. Настройки элемента **Клавишный переключатель**

14. Настроим элементы группы **Error**, используемые для управления тревогой одноименного класса:

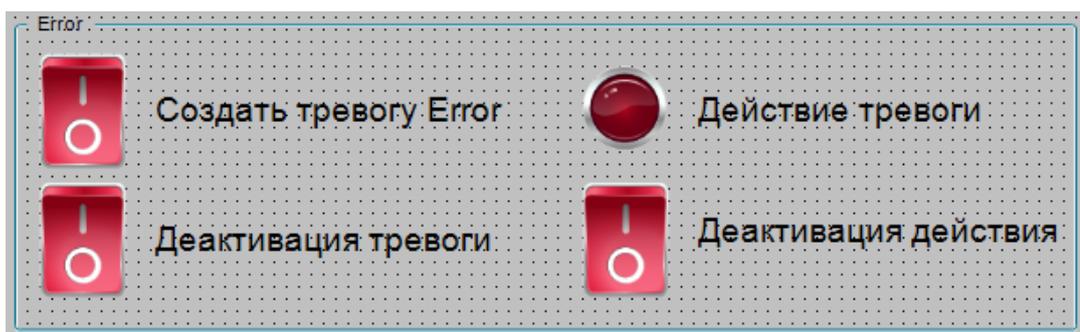


Рис. 11.2.13.21. Элементы группы **Error**

Свойства	
Фильтр	Сортировать по
Порядок сортировки	Эксперт
Свойство	Значения
Имя элемента	GenElemInst_8
Тип элемента	Клавишный выключатель
ID подсказки	30
Позиция	
X	30
Y	370
Ширина	90
Высота	95
Переменная	PLC_PRG.bAlarm
Параметры из...	
Поведение эл...	Переключатель изображения
Тексты	
Переменные...	
Фон	
Изображе...	Red

Рис. 11.2.13.22. Настройки элемента **Клавишный выключатель 1**

Свойства	
Фильтр	Сортировать по
Порядок сортировки	Эксперт
Свойство	Значения
Имя элемента	GenElemInst_51
Тип элемента	Переключатель изображения
ID подсказки	31
Позиция	
X	460
Y	50
Ширина	70
Высота	70
Переменная	PLC_PRG.bAlarmLamp
Параметры изображения	
Изображение вкл.	__VisualizationStyle.Element-Lamp-Lamp1-Red-On
Изображение выкл.	__VisualizationStyle.Element-Lamp-Lamp1-Red-Off
Image pushed	
Прозрачный	<input type="checkbox"/>
Прозрачный цвет	Black
Изотропный тип	Изотропия
Горизонтально вы...	Лево
Вертикально вы...	Верх
Поведение элемента	Переключатель изображения

Рис. 11.2.13.23. Настройки элемента **Переключатель изображения**

Свойства	
̸ Фильтр ̢ Сортировать по ̢ Порядок сортировки ̣ Эксперт	
Свойство	Значения
Имя элемента	GenElemInst_18
Тип элемента	Клавишный выключатель
[-] Позиция	
X	30
Y	475
Ширина	75
Высота	95
Переменная	PLC_PRG.bAlarmDeactivate
[+] Параметры изображения	
Поведение элемента	Переключатель изображения
[+] Тексты	
[+] Переменные состояний	
[-] Фон	
Изображение	Red

Рис. 11.2.13.24. Настройки элемента **Клавишный выключатель 2**

Свойства	
̸ Фильтр ̢ Сортировать по ̢ Порядок сортировки ̣ Эксперт	
Свойство	Значения
Имя элемента	GenElemInst_27
Тип элемента	Клавишный выключатель
[-] Позиция	
X	470
Y	475
Ширина	75
Высота	95
Переменная	PLC_PRG.bAlarmLampOff
[+] Параметры изображения	
Поведение элемента	Переключатель изображения
[+] Тексты	
[+] Переменные состояний	
[-] Фон	
Изображение	Red

Рис. 11.2.13.25. Настройки элемента **Клавишный выключатель 3**

15. Настроим элементы группы **Warning**, используемые для управления тревогой одноименного класса:

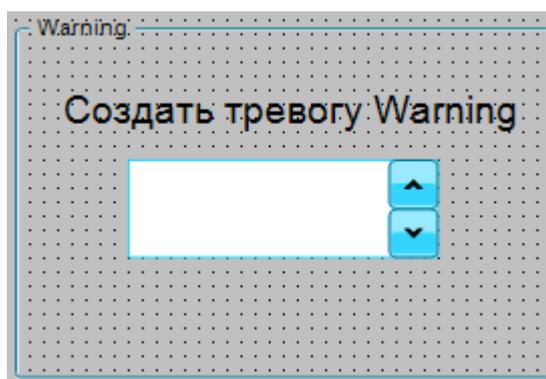
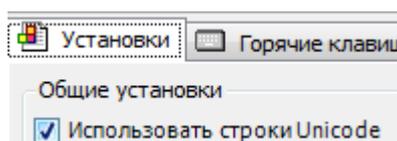


Рис. 11.2.13.26. Элементы группы **Warning**

Свойства	
🔍 Фильтр 🔗 Сортировать по 📏 Порядок сортировки <input checked="" type="checkbox"/> Эксперт	
Свойство	Значения
Имя элемента	GenElemInst_47
Тип элемента	Управление вращением
<input checked="" type="checkbox"/> Позиция	
X	970
Y	440
Ширина	220
Высота	70
Переменная	PLC_PRG.iAlarm
Числовой формат	%d
Интервал	1

Рис. 11.2.13.27. Настройки элемента **Управление вращением**

16. Настроим [Менеджер визуализации](#). В его установках поставим галочку **Использовать строки Unicode**:



Настройки **target**- и **web**-визуализации будут выглядеть следующим образом:

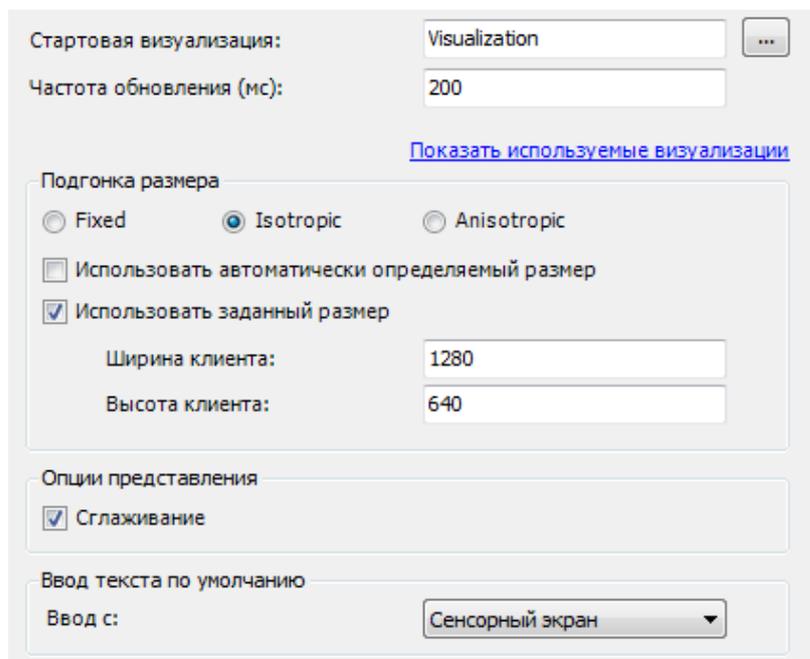


Рис. 11.2.13.28. Настройки **target**-визуализации

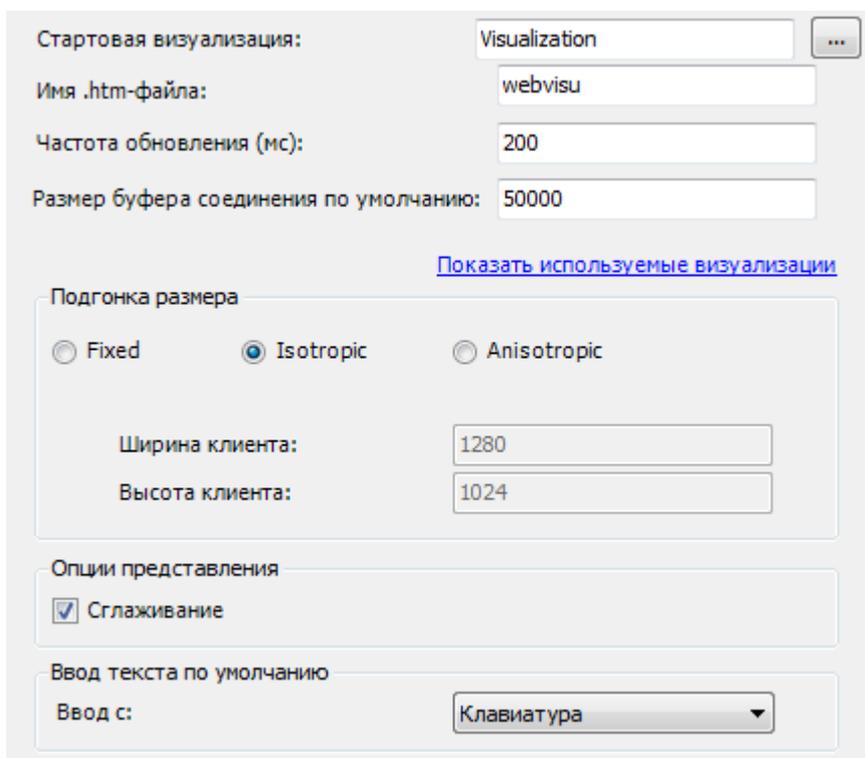


Рис. 11.2.13.29. Настройки **web**-визуализации

17. [Запустим проект на виртуальном контроллере](#). Откроем в веб-браузере страницу <http://localhost:8080/webvisu.htm>, чтобы увидеть web-визуализацию проекта. **Обратите внимание**, что браузер должен поддерживать **HTML5**. Проверим функционал проекта.

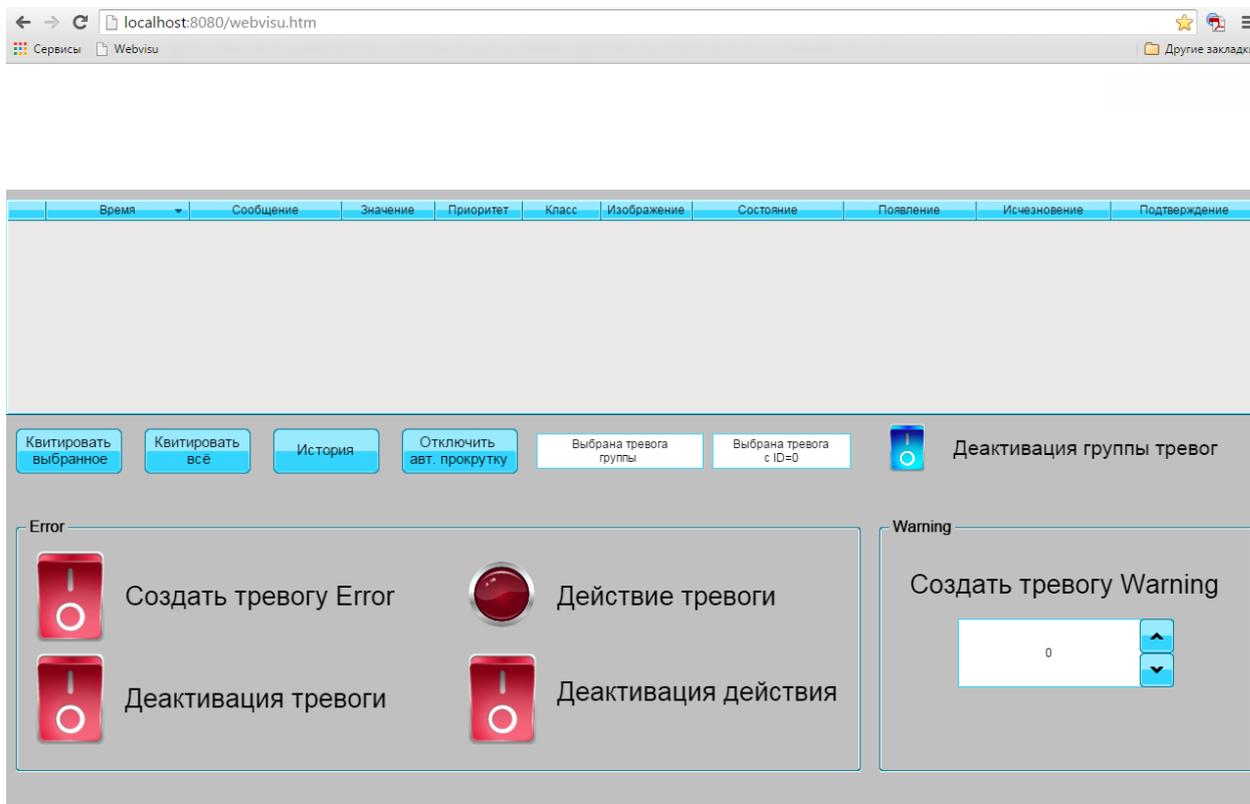


Рис. 11.2.13.30. Web-визуализация проекта в браузере **Chrome**

Нажмите переключатель **Создать тревогу Error**. Элемент **Действие тревоги** загорится красным, а в таблице отобразится соответствующая строка:

	Время	Сообщение	Значение	Приоритет	Класс	Изображение	Состояние	Появление	Исчезновение	Подтверждение
0	18.08.2015 09:52:53	Авария	TRUE	10	Error		Active	18.08.2015 09:52:53		

Рис. 11.2.13.31. Отображение в таблице активной тревоги класса **Error**

Нажмите на элемент **Действие тревоги**, чтобы отключить его. Повторно нажмите переключатель **Создать тревогу Error**, чтобы убрать условие появления тревоги. Строка таблицы изменится следующим образом:

	Время	Сообщение	Значение	Приоритет	Класс	Изображение	Состояние	Появление	Исчезновение	Подтверждение
0	18.08.2015 09:56:00	Авария	FALSE	10	Error		WaitingForConfirmation	18.08.2015 09:52:53	18.08.2015 09:56:00	

Рис. 11.2.13.32. Отображение в таблице тревоги, ожидающей подтверждения

Нажмите на строку таблицы, чтобы выделить ее. В элементах **Текстовое поле** отобразится информация о выделенной тревоге:

	Время	Сообщение	Значение	Приоритет	Класс	Изображение	Состояние	Появление	Исчезновение	Подтверждение
0	18.08.2015 09:56:00	Авария	FALSE	10	Error		WaitingForConfirmation	18.08.2015 09:52:53	18.08.2015 09:56:00	

Квитировать выбранное
Квитировать все
История
Отключить авт. прокрутку
Выбрана тревога группы AlarmGroup
Выбрана тревога с ID=0
Деактивация группы тревог

Рис. 11.2.13.33. Отображение в таблице выделенной тревоги

Нажмите на кнопку **Квитировать выбранное**, чтобы подтвердить тревогу. Это приведет к ее исчезновению из таблицы.

В элемент **Создать тревогу Warning** введите значение, превышающее 10 (например, 11). В таблице отобразится соответствующая строка:

	Время	Сообщение	Значение	Приоритет	Класс	Изображение	Состояние	Появление	Исчезновение	Подтверждение
0	18.08.2015 10:00:12	Тревога	11	30	Warning	!	Active	18.08.2015 10:00:12		

Рис. 11.2.13.34. Отображение в таблице активной тревоги класса Warning

Введите значение 9. Тревога не пропадет, хотя значение переменной теперь не превышает 10. Это связано с тем, для тревоги был задан гистерезис в 20% от значения срабатывания (см. рис. 11.2.13.10). Соответственно, чтобы отключить тревогу, значение переменной должно уменьшиться как минимум до 80% от значения срабатывания. Введите 8. Тревога исчезнет из таблицы.

Нажмите кнопку **История**, чтобы просмотреть историю тревог:

	Время	Сообщение	Значение	Приоритет	Класс	Изображение	Состояние	Появление	Исчезновение	Подтверждение
0	18.08.2015 10:10:05	Тревога	11	30	Warning		Normal	18.08.2015 10:10:04		
1	18.08.2015 10:09:55	Авария	TRUE	10	Error		Normal	18.08.2015 10:09:49	18.08.2015 10:09:52	18.08.2015 10:09:55

Квитировать выбранное
Квитировать все
История
Отключить авт. прокрутку
Выбрана тревога группы AlarmGroup
Выбрана тревога с ID=0
Деактивация группы тревог

Рис. 11.2.13.35. Таблица тревог в режиме просмотра истории

В столбце **Время** отображается время последнего изменения состояния тревоги. Для тревоги класса **Warning** столбец **Исчезновение** остается пустым, а время исчезновения отображается в столбце **Время** – это связано с тем, что для нее выбран способ подтверждения **REP** (см. рис. 11.2.13.9), который подразумевает только одно состояние тревоги (активна).

Переключатели деактивации работают следующие образом:

1. при деактивации группы тревог, сообщения данной группы тревог не отображаются в таблице и ее истории;
2. при деактивации тревоги, сообщения данной тревоги не отображаются в таблице и ее истории;
3. при деактивации действия, действия данной тревоги не выполняются при ее активации.

11.3. Примеры решения типовых задач

11.3.1. Переключение экранов в проекте

Данный пример посвящен реализации переключения экранов визуализации в пользовательском проекте. Рассмотрено два способа переключения:

1. переключение с помощью ручного ввода (действие элемента [Кнопка](#));
2. переключение из кода программы с использованием системной переменной **CurrentVisu**.

Обратите внимание, что при использовании в проекте системной переменной **CurrentVisu**, переключение экранов будет затрагивать **всех клиентов визуализации**. Соответственно, при необходимости многопользовательского доступа, использование этой переменной крайне нежелательно.

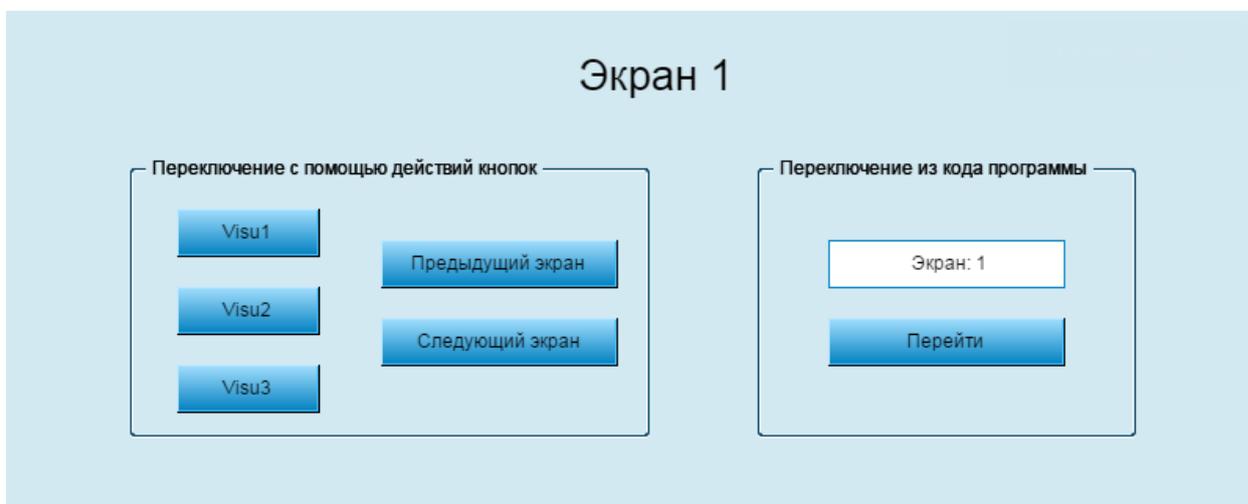


Рис. 11.3.1.1. Внешний вид примера **Переключение экранов**

Пример создан в среде **CODESYS 3.5 SP6 Patch 2** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с target-файлом **3.5.6.0**.

Пример доступен для скачивания: Example_VisuSwitch.projectarchive

1. Создадим новый *стандартный* проект **CODESYS** с названием **Example_VisuSwitch** и настройками по умолчанию: target – **CODESYS Control Win V3**, язык PLC_PRG – **ST**.

2. В программе **PLC_PRG** объявим переменную **iVisuIndex** типа **INT**:

```
1 PROGRAM PLC_PRG
2 VAR
3     iVisuIndex:INT; // индекс экрана визуализации, на который осуществляется переход
4 END VAR
```

Рис. 11.3.1.2. Объявление переменных программы **PLC_PRG**

3. Код программы **PLC_PRG** будет выглядеть следующим образом:

```
1 // в зависимости от числа, введенного пользователем...
2 // ...происходит переход на соответствующий экран визуализации
3
4 CASE iVisuIndex OF
5     1: VisuElems.CURRENTVISU:='Visu1';
6     2: VisuElems.CURRENTVISU:='Visu2';
7     3: VisuElems.CURRENTVISU:='Visu3';
8 END_CASE
9
10 // сброс индекса экрана после перехода для возможности...
11 // ...переключения экранов действиями кнопок
12
13 iVisuIndex:=0;
```

Рис. 11.3.1.3 Код программы **PLC_PRG**

Обратите внимание, что использование системной переменной **VisuElems.CurrentVisu** возможно только при наличии соответствующей галочки в установках [Менеджера визуализации](#). Напомним, что при наличии в проекте этой галочке, переключение экранов визуализации затрагивает всех клиентов, что делает нежелательным использование этой переменной в проектах с возможностью многопользовательского доступа.

4. [Добавим в проект](#) три экрана визуализации с названиями **Visu1**, **Visu2** и **Visu3**. В их [свойствах](#) выберем размер **800x480**. На каждом из экранов добавим элемент [Метка](#) с названием экрана (**Экран 1**, **Экран 2**, **Экран 3**). Выберем для каждого из экранов свой цвет [фона](#).

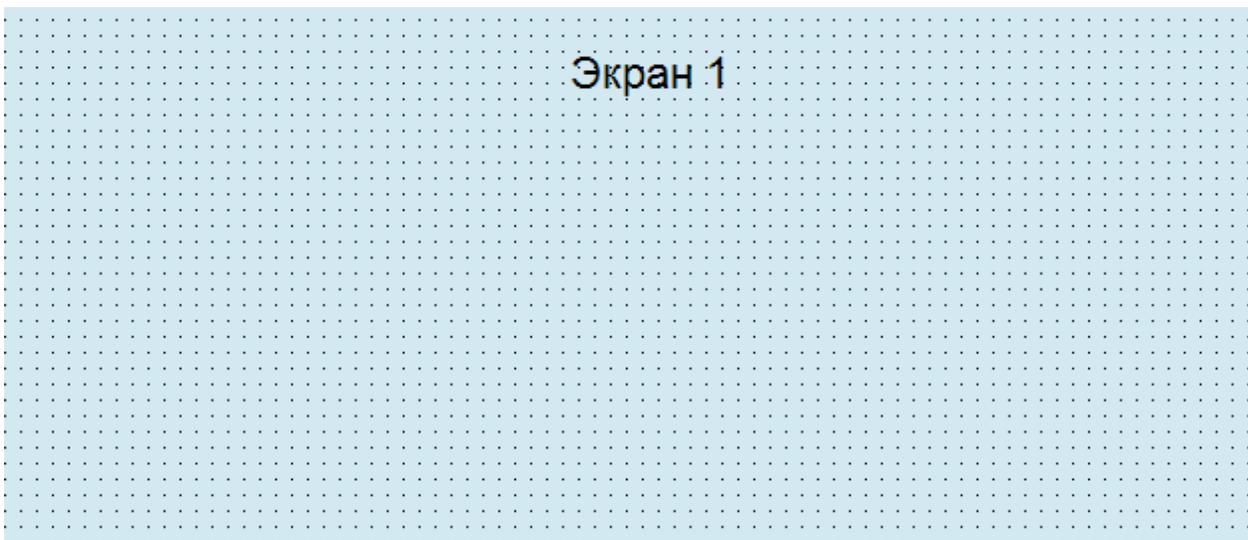


Рис. 11.3.1.4. Экран визуализации **Visu1**

Настройка экранов описана ниже.

5. В [интерфейсе](#) каждого из экранов х объявите локальную переменную **iSelect** типа **INT**, начальное значение которой будет совпадать с номером экрана:

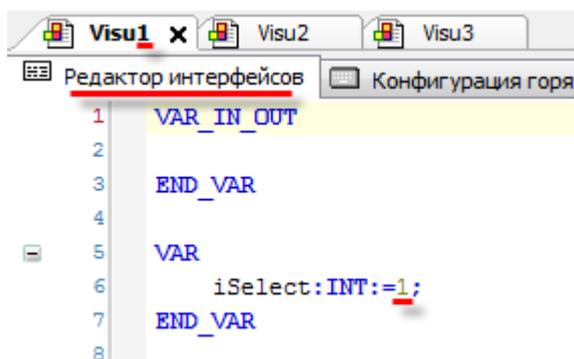


Рис. 11.3.1.5. Интерфейс экрана визуализации **Visu1**

Соответственно, для экрана **Visu2** переменная **iSelect** будет иметь начальное значение **2**, а для **Visu3** – **3**.

6. Экран **Visu1** будет содержать два элемента [Группа](#), шесть элементов [Кнопка](#) и элемент [Текстовое поле](#):

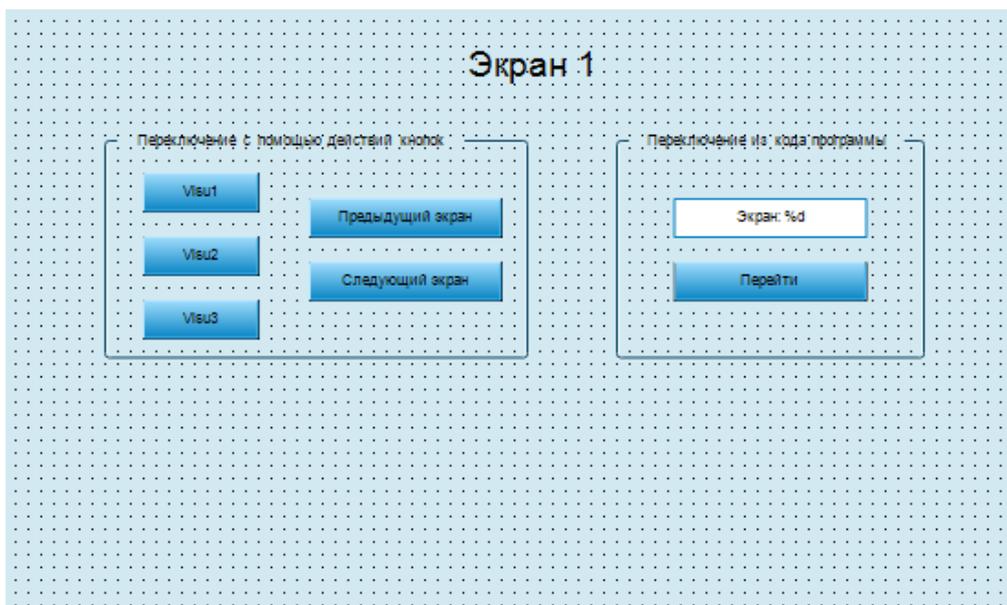


Рис. 11.3.1.6. Содержимое экрана визуализации **Visu1**

Настроим кнопки **Visu1**, **Visu2**, **Visu3**. В свойствах элементов во вкладке [InputConfiguration](#) выберем параметр **OnClick** и привяжем к нему действие [Изменить отображаемую визуализацию](#), которому присвоим название соответствующего экрана:

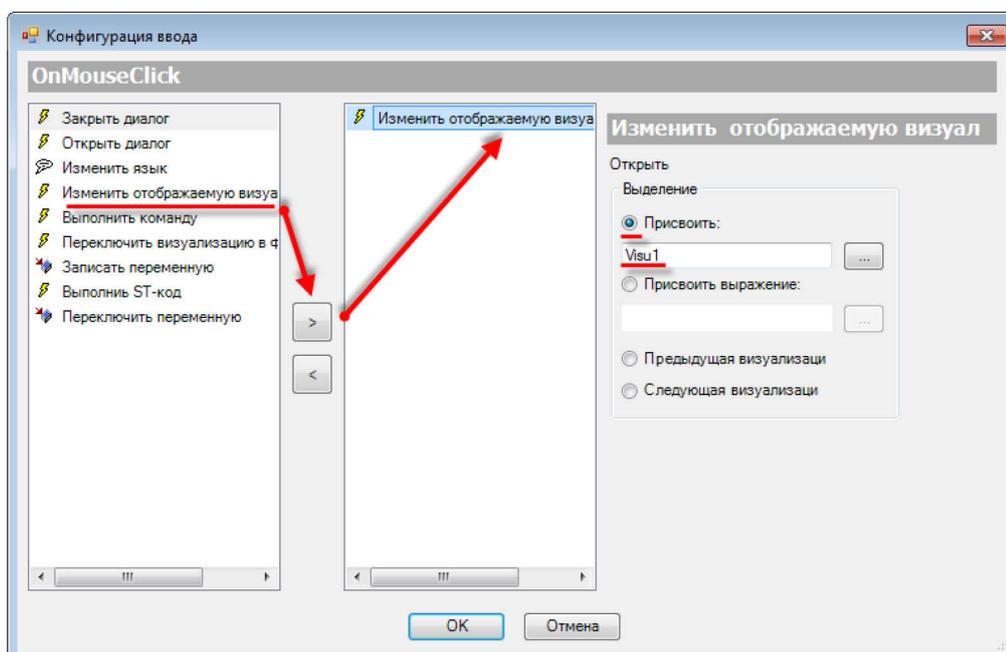


Рис. 11.3.1.7. Настройка действия кнопки **Visu1**

Настройки кнопок **Visu2** и **Visu3** аналогичны.

Действия кнопок **Предыдущая визуализация** и **Следующая визуализация**, соответственно, будут выглядеть следующим образом:

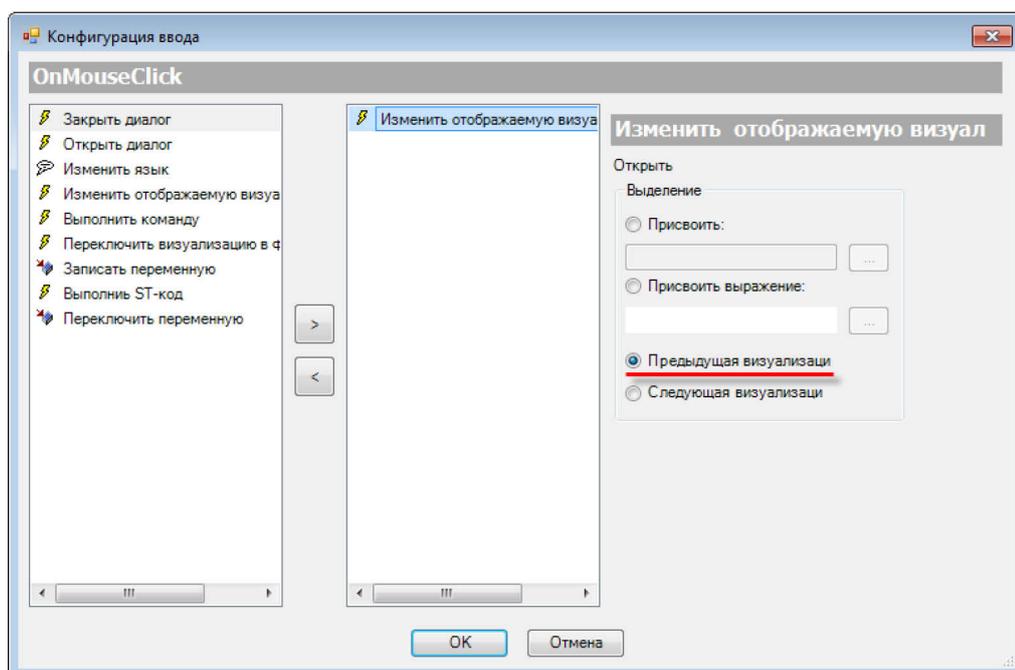


Рис. 11.3.1.8. Настройки действия кнопки **Предыдущая визуализация**

Для кнопки **Следующая визуализация** выберите пункт **Следующая визуализация**.

Настройки элемента [Текстовое поле](#) будут выглядеть следующим образом:

Свойства	
Фильтр Сортировать по Порядок сортировки	
<input checked="" type="checkbox"/> Эксперт	
Свойство	Значения
Имя элемента	GenElemInst_273
Тип элемента	Текстовое поле
ID текста	38
+ Позиция	
+ Цвета	
+ Вид элемента	
Тип тени	Из стиля
- Тексты	
Текст	Экран: %d
Подсказка	
+ Свойства текста	
- Текстовые переменные	
Текстовая перемен...	iSelect
Переменная подска...	
+ Динамические тексты	
+ Переменные шрифта	
+ Переменные цвета	
+ Переменные состояний	
- Inputconfiguration	
OnDialogClosed	Конфигурация...
- OnMouseClicked	
Записать перем...	Variable : , InputType: VisuDialogs.Key..
OnMouseDown	Конфигурация...
OnMouseEnter	Конфигурация...
OnMouseLeave	Конфигурация...
OnMouseMove	Конфигурация...
OnMouseUp	Конфигурация...

Рис. 11.3.1.9. Параметры элемента **Текстовое поле**

В свойствах [Текстового поля](#) во вкладке [InputConfiguration](#) к параметру **OnClick** привяжите действие [Записать переменную](#) со следующими настройками:

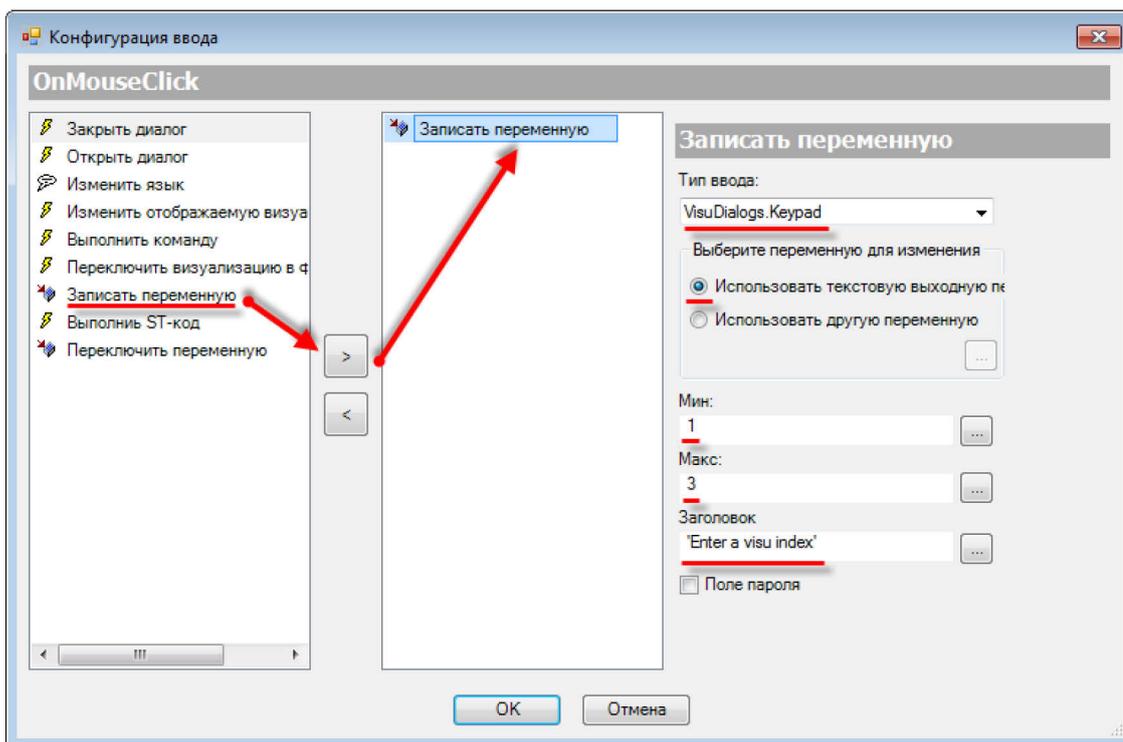


Рис. 11.3.1.10. Настройки действия элемента **Текстовое поле**

В свойствах кнопки **Применить** во вкладке [InputConfiguration](#) к параметру **OnClick** привяжите действие [Выполнить ST-код](#) со следующими настройками:

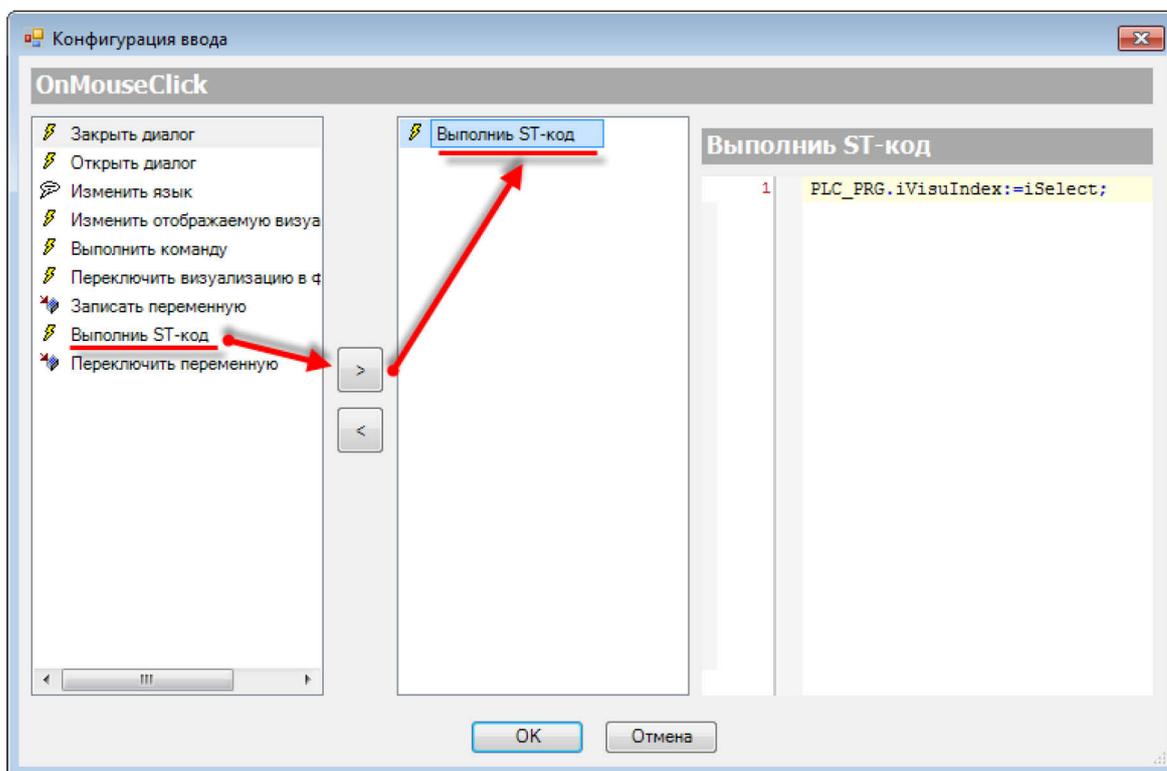


Рис. 11.3.1.11. Настройки действия кнопки **Применить**

7. Скопируем содержимое экрана **Visu1** на экраны **Visu2** и **Visu3**:

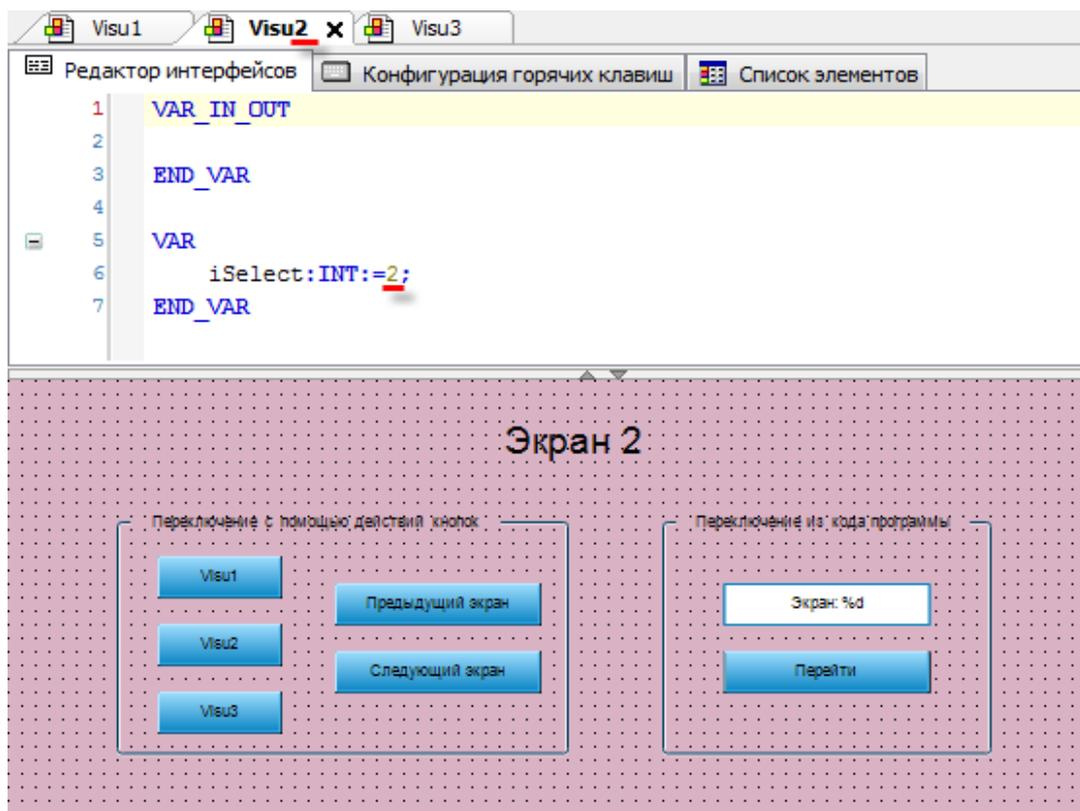


Рис. 11.3.1.12. Содержимое экрана визуализации **Visu2**

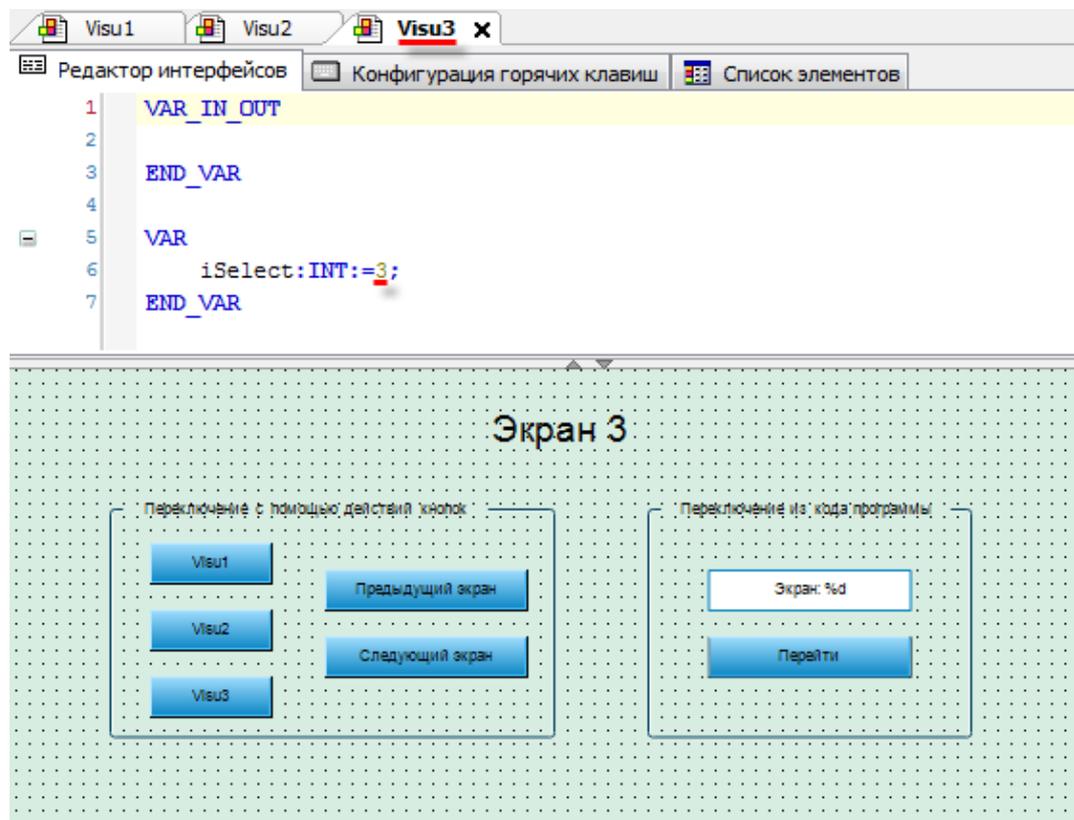
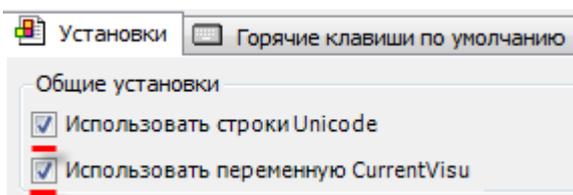


Рис. 11.3.1.13. Содержимое экрана визуализации **Visu3**

8. Настроим [Менеджер визуализации](#). В его установках поставим галочки **Использовать строки Unicode** и **Использовать переменную CurrentVisu**:



Настройки **target**- и **web**-визуализации будут выглядеть следующим образом:

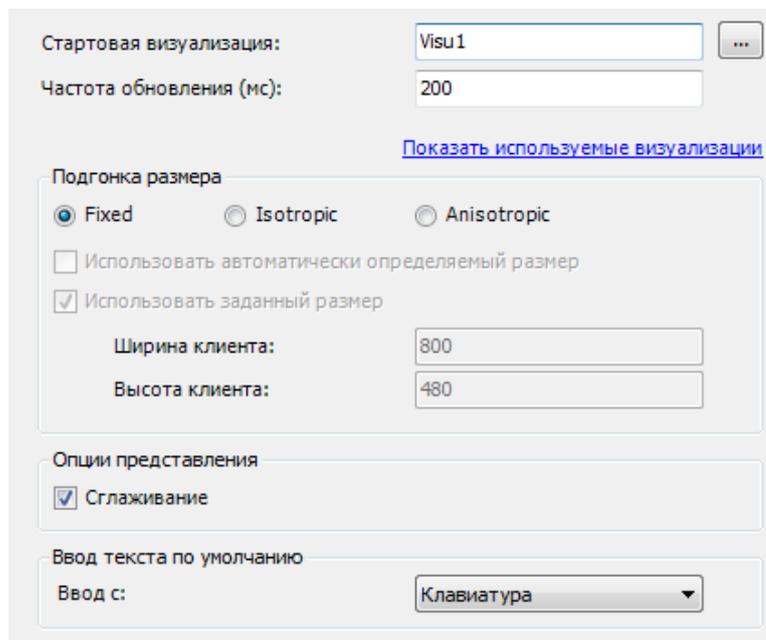


Рис. 11.3.1.14. Настройки **target**-визуализации

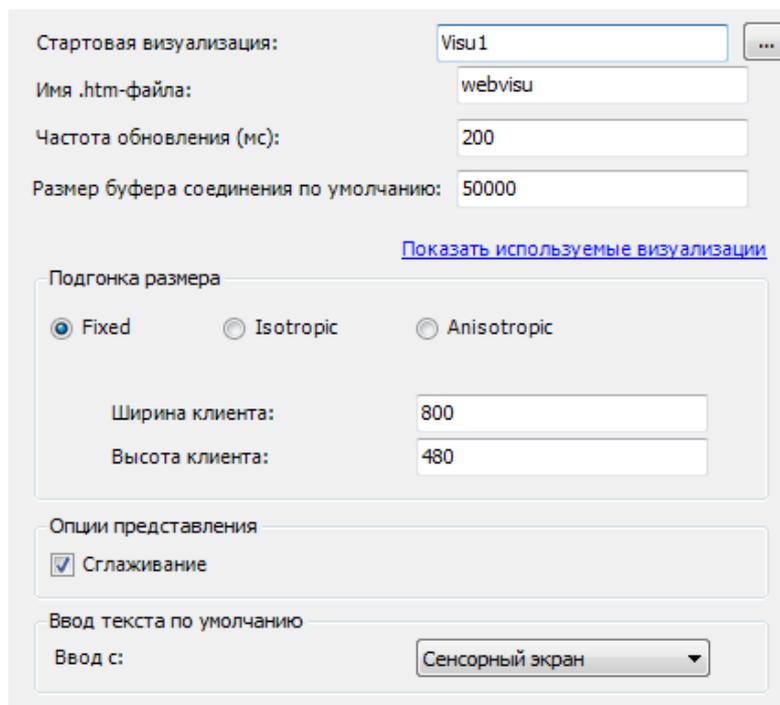


Рис. 11.3.1.15. Настройки **web**-визуализации

9. [Запустим проект на виртуальном контроллере](#). Откроем в веб-браузере страницу <http://localhost:8080/webvisu.htm>, чтобы увидеть web-визуализацию проекта. **Обратите внимание**, что браузер должен поддерживать **HTML5**. Проверим функционал проекта.

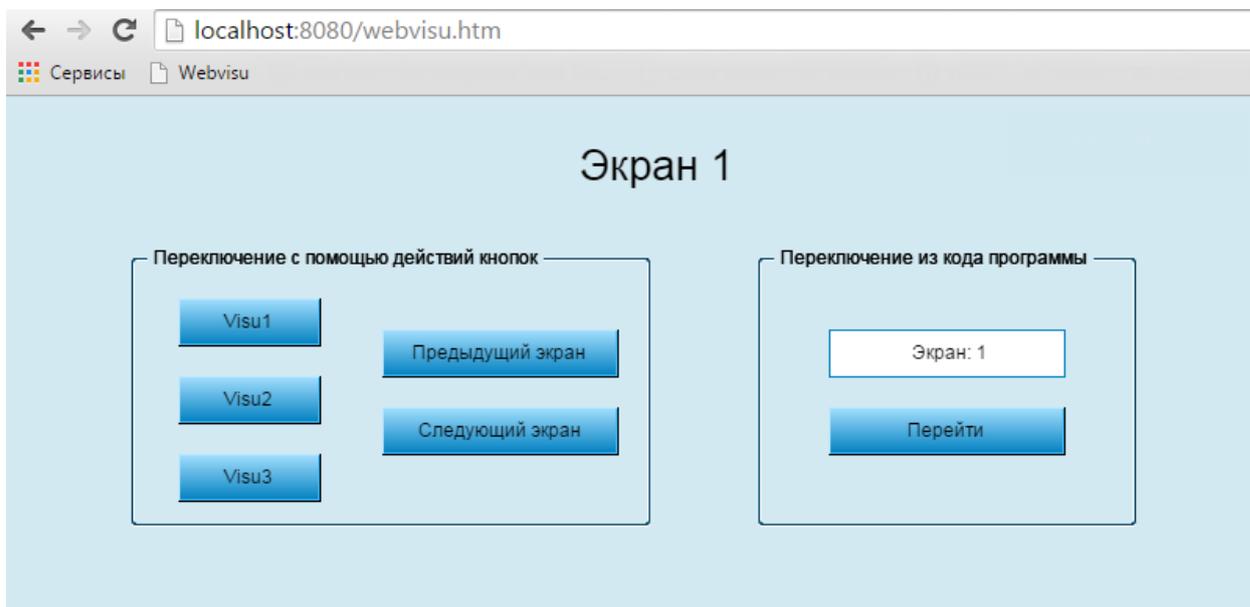


Рис. 11.3.1.16 Web-визуализация проекта в браузере **Chrome**

При запуске проекта пользователь находится на экране 1.

Во вкладке **Переключение** с помощью действий кнопок нажмите кнопку **Visu2**, чтобы перейти на экран 2. Нажмите кнопку **Visu3**, чтобы перейти на экран 3. Нажмите на кнопку **Предыдущий экран** – это приведет к переходу на экран 2. Нажмите на кнопку **Следующий экран** – это приведет к возвращению на экран 3.

Во вкладке **Переключение из кода программы** нажмите на текстовое поле **Экран** и введите 2. Для переключения раскладки экранной клавиатуры воспользуйтесь клавишей **Shift**.



Рис. 11.3.1.17. Переключение раскладки экранной клавиатуры

Нажмите кнопку **Перейти**, чтобы перейти на экран с выбранным номером. **Обратите внимание**, что переключение экранов в этом случае происходит в коде программы с помощью системной переменной **CurrentVisu**.

11.3.2. Использование диалогов

Данный пример посвящен использованию [диалоговых окон](#) в пользовательском проекте.

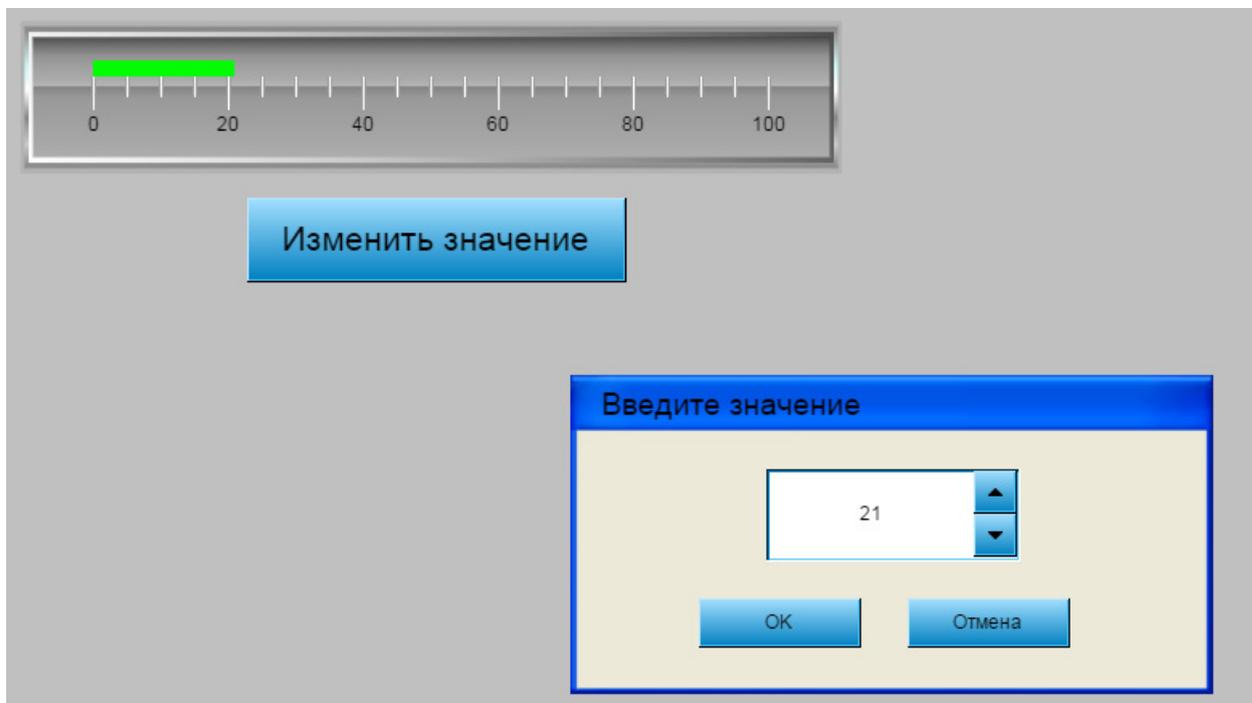


Рис. 11.3.2.1. Внешний вид примера **Использование диалогов**

Пример создан в среде **CODESYS 3.5 SP6 Patch 2** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с target-файлом **3.5.6.0**.

Пример доступен для скачивания: Example_VisuDialogs.projectarchive

1. Создадим новый **стандартный** проект **CODESYS** с названием **Example_VisuDialogs** и настройками по умолчанию: target – **CODESYS Control Win V3**, язык PLC_PRG – **ST**.

2. В программе **PLC_PRG** объявим переменную **iVisuVar** типа **INT**:

```
1 PROGRAM PLC_PRG
2 VAR
3   iVisuVar:INT; // переменная экрана | визуализации, значение которой изменяется в окне диалога
4 END_VAR
5
6
```

Рис. 11.3.2.2. Объявление переменных программы **PLC_PRG**

3. Добавим в проект экран визуализации **ChangeValue**, который будет использоваться в качестве диалога. В его свойствах выберем тип **Диалог** и размер **400x200**:

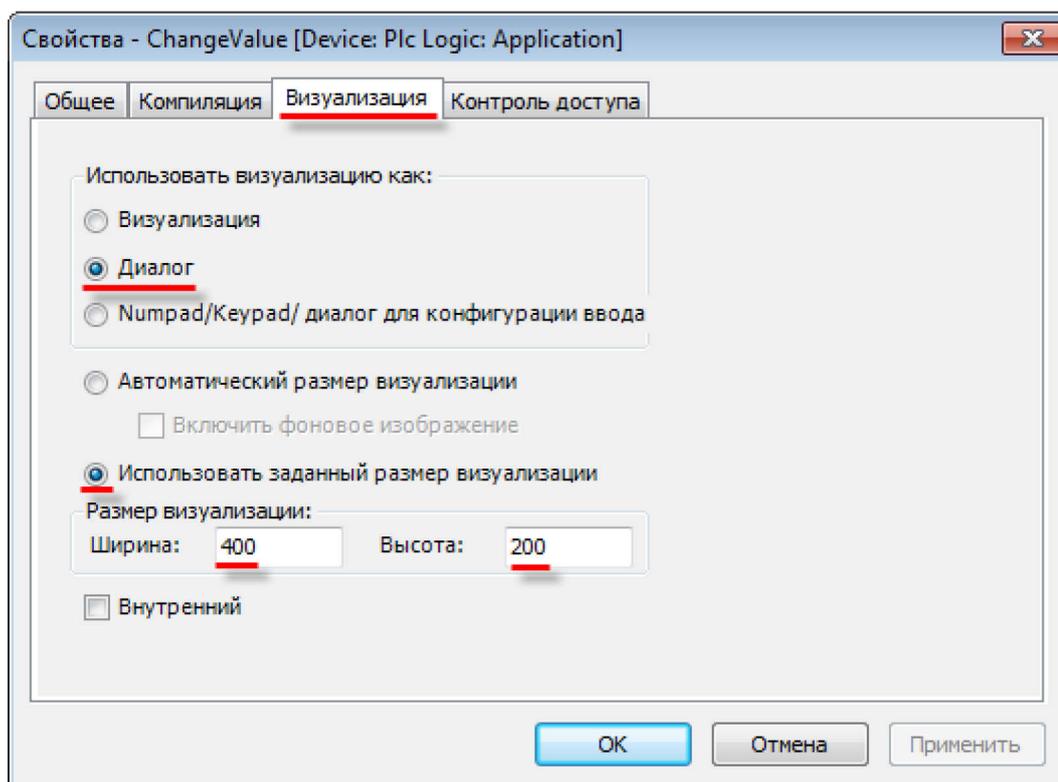


Рис. 11.3.2.3. Свойства экрана **ChangeValue**

4. Диалоговое окно **ChangeValue** содержит элемент [Изображение](#), элемент [Метка](#), элемент [Управление вращением](#) и два элемента [Кнопка](#), а также выходную переменную [интерфейса](#) `iDialogVar` типа `INT`:

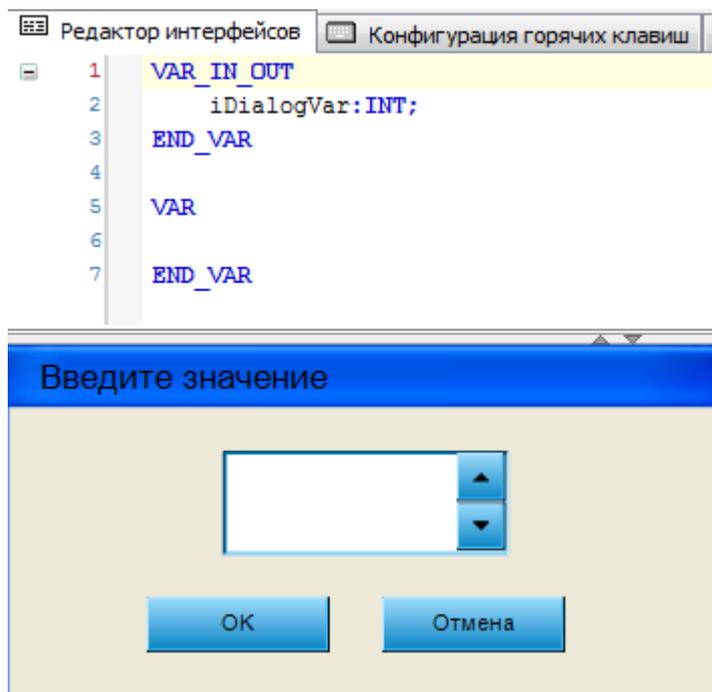


Рис. 11.3.2.4. Интерфейс и содержимое экрана **ChangeValue**

Настройки элементов приведены ниже.

Свойства	
Свойство	Значения
Имя элемента	GenElemInst_30
Тип элемента	Изображение
Статический ID	VisuDialogs.ImagePoolDialogs.Login
Показать фрейм	<input type="checkbox"/>
Кадрирование	<input type="checkbox"/>
Прозрачный	<input type="checkbox"/>
Прозрачный цвет	 Black
Тип шкалы	Анизотропный

Рис. 11.3.2.5. Параметры элемента **Изображение**

Элемент [Изображение](#) расположен в нижнем слое (см. [п. 6.2](#), пп. 2) по отношению к остальным элементам, чтобы не перекрывать их.

Свойства	
Ү Фильтр Сортировать по Порядок сортировки	
<input checked="" type="checkbox"/> Эксперт	
Свойство	Значения
Имя элемента	GenElemInst_24
Тип элемента	Управление вращением
<input checked="" type="checkbox"/> Позция	
Переменная	iDialogVar
Числовой формат	
Интервал	1

Рис. 11.3.2.6. Параметры элемента **Управление вращением**

В свойствах кнопки **OK** во вкладке [InputConfiguration](#) к параметру **OnClick** привяжем действие [Закреть диалог](#):

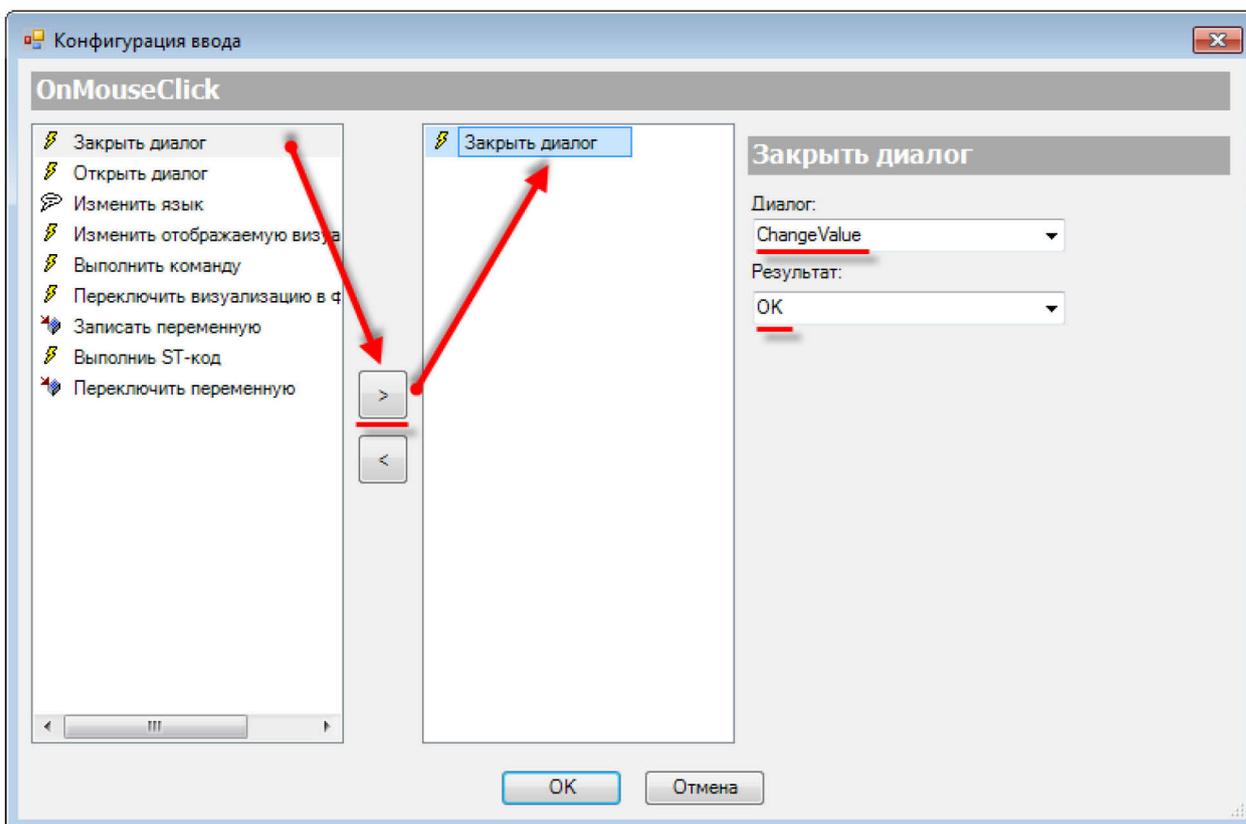


Рис. 11.3.2.7. Настройки действия кнопки **OK**

Настройки действия кнопки **Отмена** будут отличаться только результатом (результат: **Отмена**).

4. [Добавим в проект](#) экран визуализации **Visualization**. В его [свойствах](#) выберем размер **800x480**. Этот экран будет являться основным экраном проекта, и именно на нем будет располагаться кнопка вызова диалога.

Экран содержит элемент [Отображение линейки](#) и элемент [Кнопка](#):

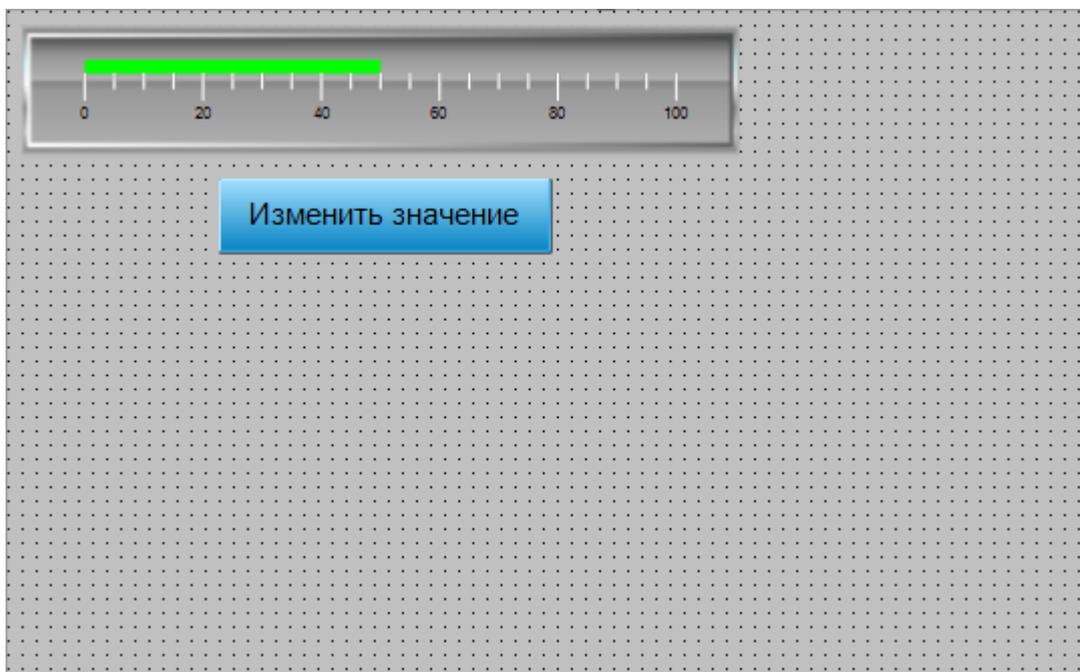


Рис. 11.3.2.8. Содержимое экрана **Visualization**

Свойства	
<input type="checkbox"/> Фильтр <input type="checkbox"/> Сортировать по <input type="checkbox"/> Порядок сортировки	
<input checked="" type="checkbox"/> Эксперт	
Свойство	Значения
Имя элемента	GenElemInst_293
Тип элемента	Отображение линейки
Значение	PLC_PRG.iVisuVar
<input type="checkbox"/> Позиция	
<input type="checkbox"/> Фон	
<input type="checkbox"/> Линейка	
<input type="checkbox"/> Шкала	
<input type="checkbox"/> Метка	
Единица	
Шрифт	Font-Standard
Формат шкалы (с-си...	%d
Макс. ширина текст...	35
Высота текста меток	15
Цвет шрифта	 Font-Default-Color

Рис. 11.3.2.9. Параметры элемента **Отображение линейки**

Нам осталось настроить действие кнопки **Изменить значение**. В данный момент (3.5 SP6), настройка открытия диалогов работает некорректно в русскоязычной версии среды **CODESYS**, поэтому переключим среду программирования на англоязычную версию:

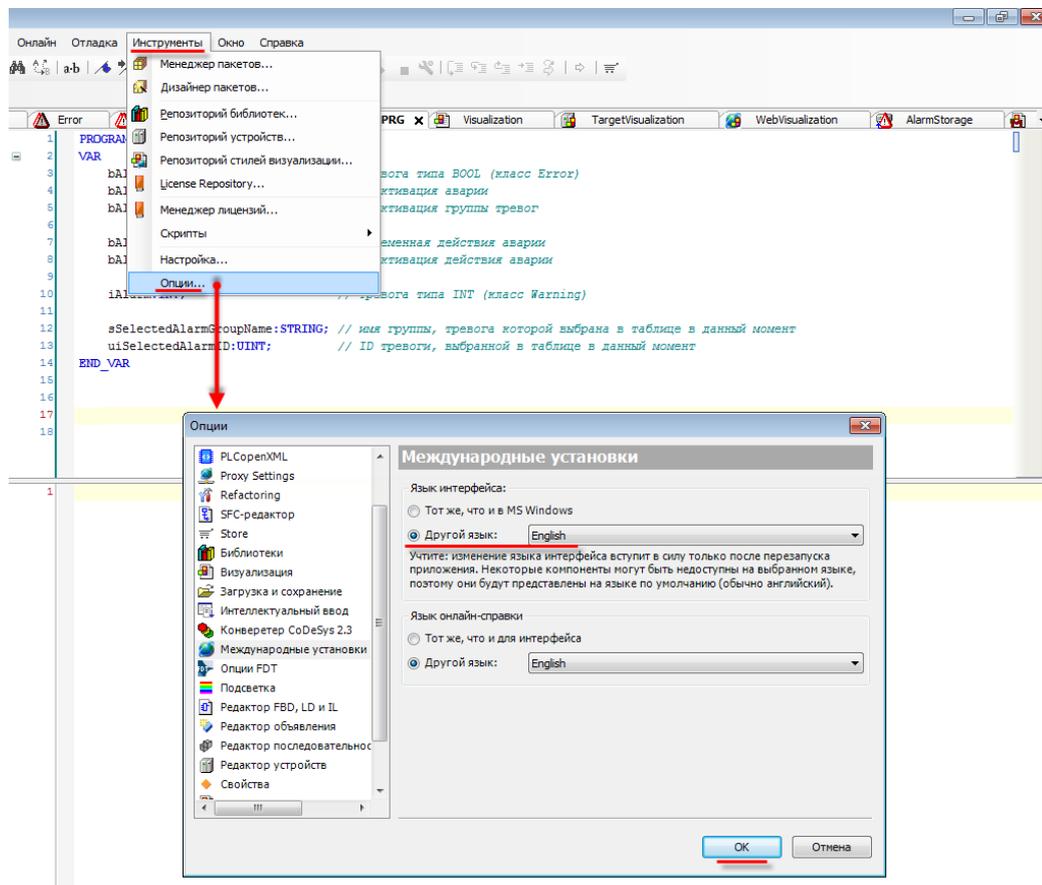


Рис. 11.3.2.10. Переключение языка интерфейса **CODESYS**

После этого потребуется перезапустить **CODESYS**.

В свойствах кнопки **Изменить значение** во вкладке [InputConfiguration](#) к параметру **OnClick** привяжем действие [Открыть диалог](#):

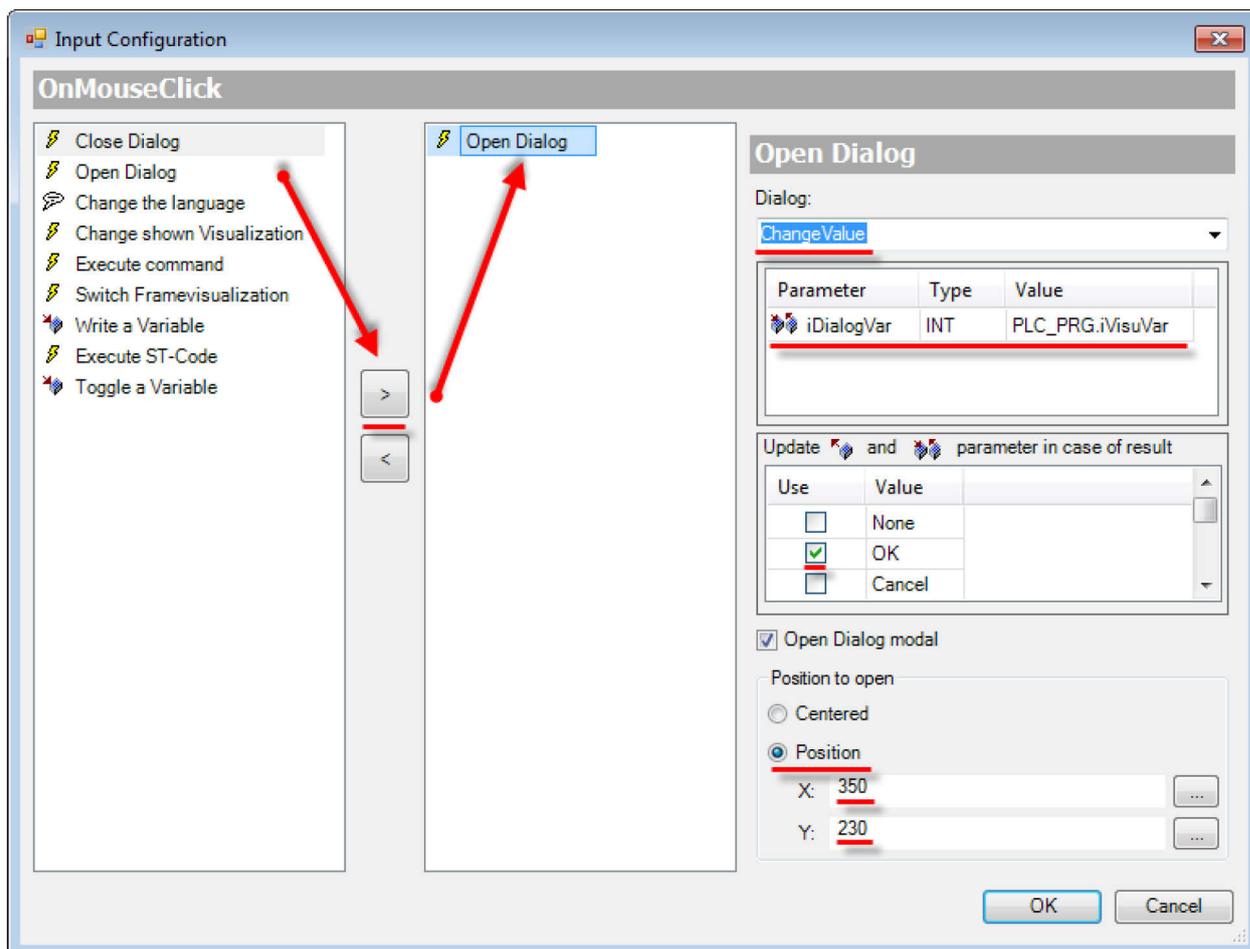
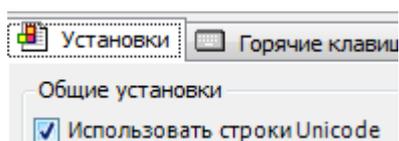


Рис. 11.3.2.11. Настройки действия кнопки **Изменить значение**

После такой настройки, в результате закрытия диалога **ChangeValue** с результатом **OK** (т.е. после нажатия на кнопку **OK**), значение локальной переменной диалога **iDialogVar** будет присвоено переменной программы **PLC_PRG** (**PLC_PRG.iVisuVar**) и отобразится элементом **Отображение линейки**.

5. Настроим [Менеджер визуализации](#). В его установках поставим галочку **Использовать строки Unicode**:



Настройки **target**- и **web**-визуализации будут выглядеть следующим образом:

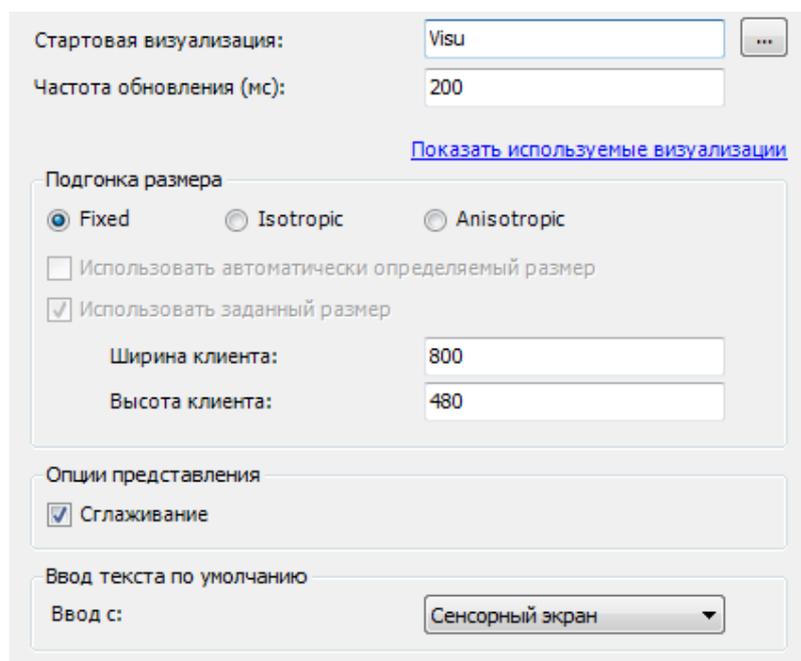


Рис. 11.3.2.12. Настройки **target**-визуализации

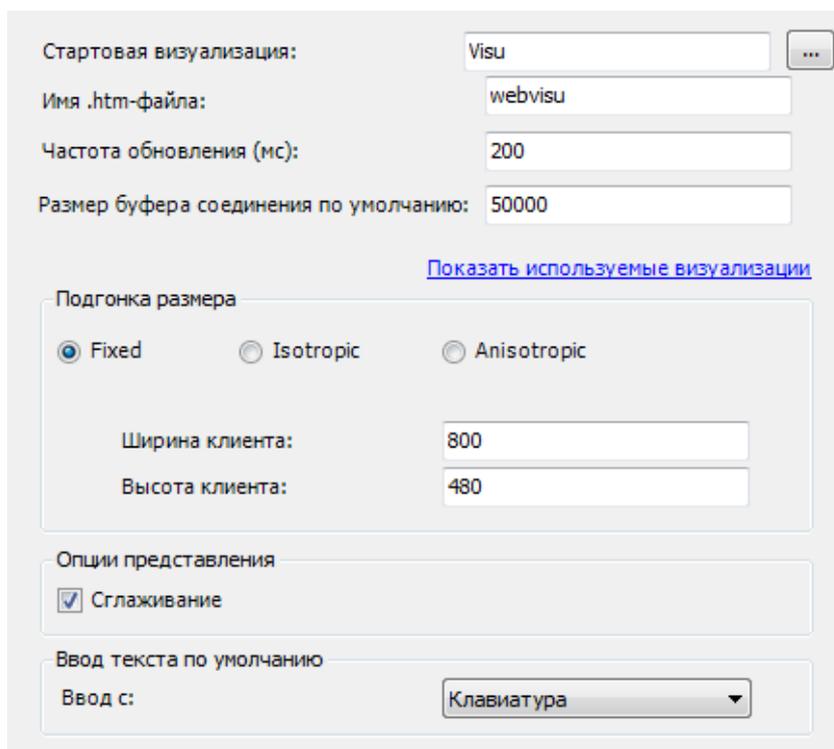


Рис. 11.3.2.13. Настройки **web**-визуализации

6. [Запустим проект на виртуальном контроллере](#). Откроем в веб-браузере страницу <http://localhost:8080/webvisu.htm>, чтобы увидеть web-визуализацию проекта. **Обратите внимание**, что браузер должен поддерживать **HTML5**. Проверим функционал проекта.

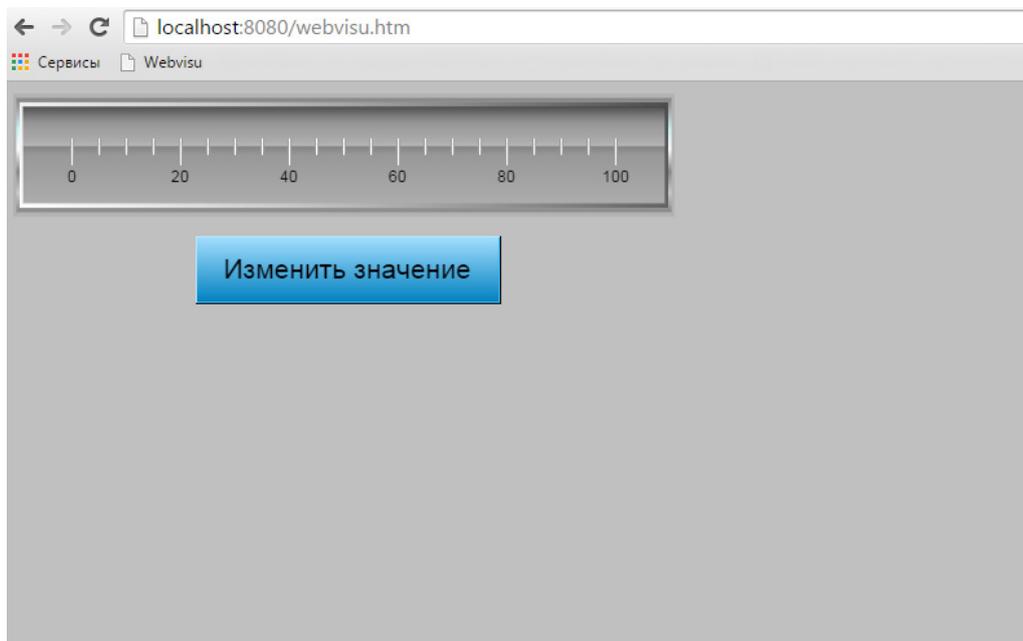


Рис. 11.3.2.14 Web-визуализация проекта в браузере **Chrome**

Нажмите кнопку изменить значение для открытия диалога **ChangeValue**:

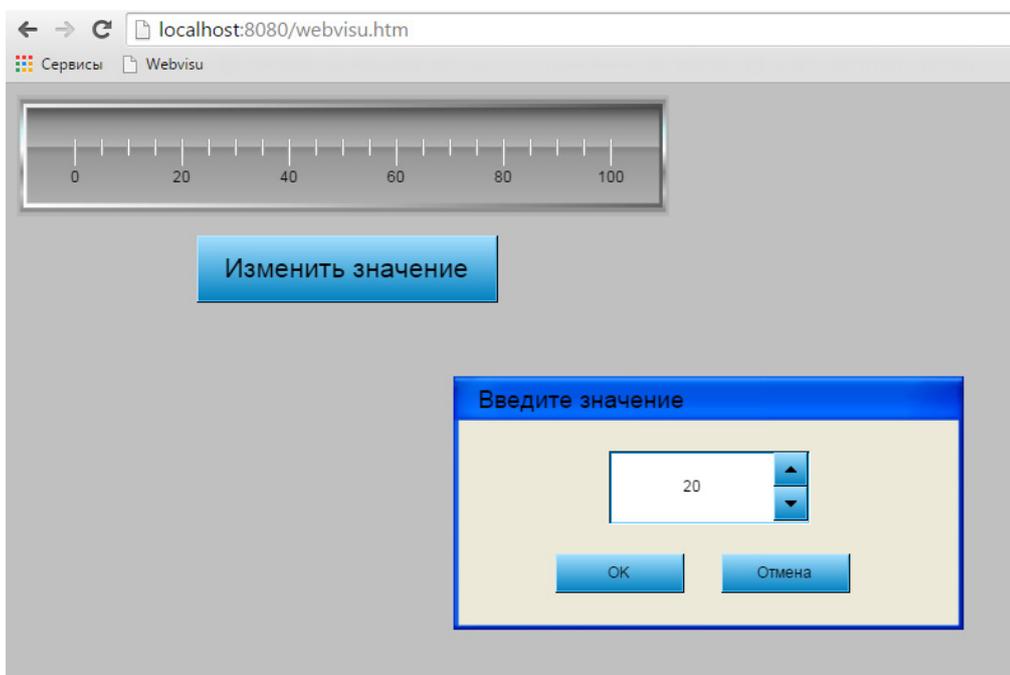


Рис. 11.3.2.15. Изменение значения с помощью диалога

Введите число и нажмите **ОК**. Введение значение отобразится элементом **Отображение линейки**. Нажатие на кнопку **Отмена** закроет диалог без записи значения в программу.

11.3.3. Использование интерфейса фрейма

Данный пример посвящен работе с [интерфейсом](#) элемента [Фрейм](#).

Довольно часто в системах автоматизации присутствуют группы типовых устройств, для которых необходимо осуществлять мониторинг и управление. Размещать их на одном экране или создавать индивидуальный экран для каждой группы устройств иногда является нецелесообразным; в этом случае можно воспользоваться [интерфейсом](#) элемента **Фрейм**.

Использование интерфейса элемента **Фрейм** позволяет открывать внутри фрейма один и тот же экран визуализации с разными привязками переменных. Иными словами, можно сделать следующее:

1. создать экран визуализации шаблона устройства;
2. привязать к нему входные переменные интерфейса;
3. в конфигурации **Фрейма** добавить несколько экземпляров экрана шаблона устройства;
4. в параметрах **Фрейма** к каждому из экземпляров привязать переменные соответствующего устройства.

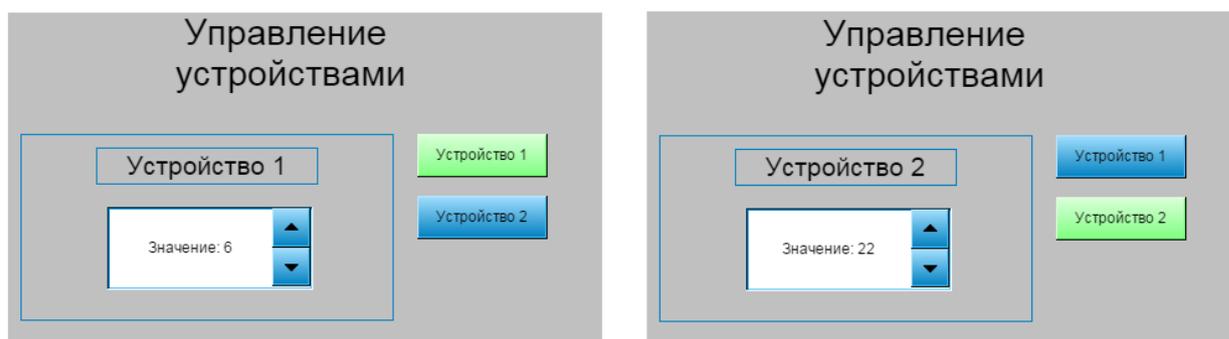


Рис. 11.3.3.1. Внешний вид примера **Интерфейс фрейма**.

На обоих рисунках в фрейме открыт один и тот же экран визуализации

Пример создан в среде **CODESYS 3.5 SP6 Patch 2** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с target-файлом **3.5.6.0**.

Пример доступен для скачивания: [Example_FrameInterface.projectarchive](#)

1. Создадим новый **стандартный** проект **CODESYS** с названием **Example_FrameInterface** и настройками по умолчанию: target – **CODESYS Control Win V3**, язык PLC_PRG – **ST**.

2. В программе **PLC_PRG** объявим следующие переменные:

```
1 PROGRAM PLC_PRG
2 VAR
3     // Переменные устройства 1
4     bButtonDevice1:BOOL:=TRUE; // переменная для изменения цвета кнопки при выборе устройства
5     wsNameDevice1:WSTRING:="Устройство 1"; // название устройства
6     iValueDevice1:INT; // параметр устройства
7
8     // Переменные устройства 2
9     bButtonDevice2:BOOL;
10    wsNameDevice2:WSTRING:="Устройство 2";
11    iValueDevice2:INT;
12 END_VAR
```

Рис. 11.3.3.2. Объявление переменных программы **PLC_PRG**

3. Добавим в проект экран визуализации **Frame**. В его свойствах выберем размер **300x150**. Экран будет содержать элемент Текстовое поле, элемент Управление вращением и две входных переменных интерфейса:

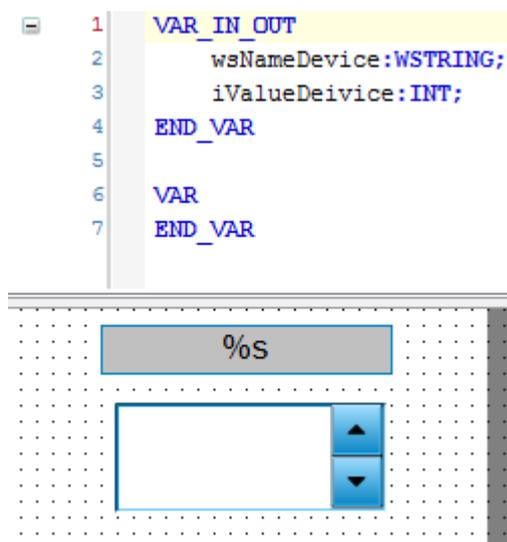


Рис. 11.3.3.3. Интерфейс и содержимое экрана **Frame**

Свойства	
<input type="text" value="Фильтр"/> <input type="button" value="Сортировать по"/> <input type="button" value="Порядок сортировки"/> <input checked="" type="checkbox"/> Эксперт	
Свойство	Значения
Имя элемента	GenElemInst_2
Тип элемента	Текстовое поле
ID текста	0
Позиция	
X	60
Y	10
Ширина	180
Высота	30
Цвета	
Нормальное состояние	
Цвет фрейма	 Element-Frame-Color
Цвет заливки	 192; 192; 192
Состояние тревоги	
Вид элемента	
Ширина линии	1
Атрибуты заливки	Заполнено
Стиль линии	Сплошной
Тип тени	Из стиля
Тексты	
Текст	%s
Подсказка	
Свойства текста	
Текстовые переменные	
Текстовая переменная	wsNameDevice
Переменная подсказки	

Рис. 11.3.3.4. Параметры элемента **Текстовое поле**

Свойства	
<input type="text" value="Фильтр"/> <input type="button" value="Сортировать по"/> <input type="button" value="Порядок сортировки"/> <input checked="" type="checkbox"/> Эксперт	
Свойство	Значения
Имя элемента	GenElemInst_4
Тип элемента	Управление вращением
Позиция	
X	70
Y	60
Ширина	165
Высота	65
Переменная	iValueDeivice
Числовой формат	Значение: %d
Интервал	1

Рис. 11.3.3.5. Параметры элемента **Управление вращением**

5. [Добавим в проект](#) экран визуализации **Visualization**. В его [свойствах](#) выберем размер **480x272**. Экран будет содержать элемент [Фрейм](#) и два элемента [Кнопка](#). Название экрана создано с помощью элемента [Метка](#).

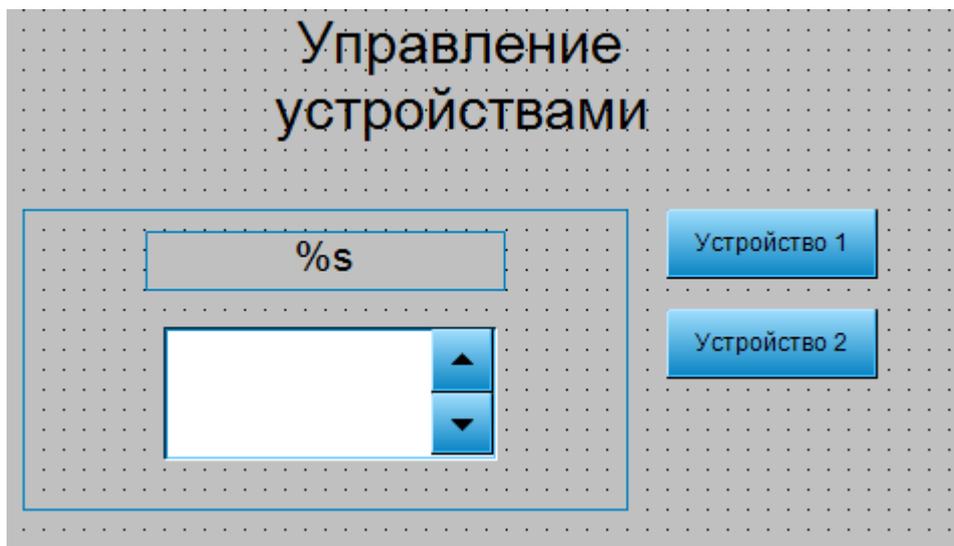


Рис. 11.3.3.6. Содержимое экрана **Visualization**

Настройки элементов приведены ниже.

Привяжем к элементу **Фрейм** два экземпляра экрана **Frame**, и к интерфейсу каждого из них привяжем соответствующие переменные программы (вкладка **Ссылки**):

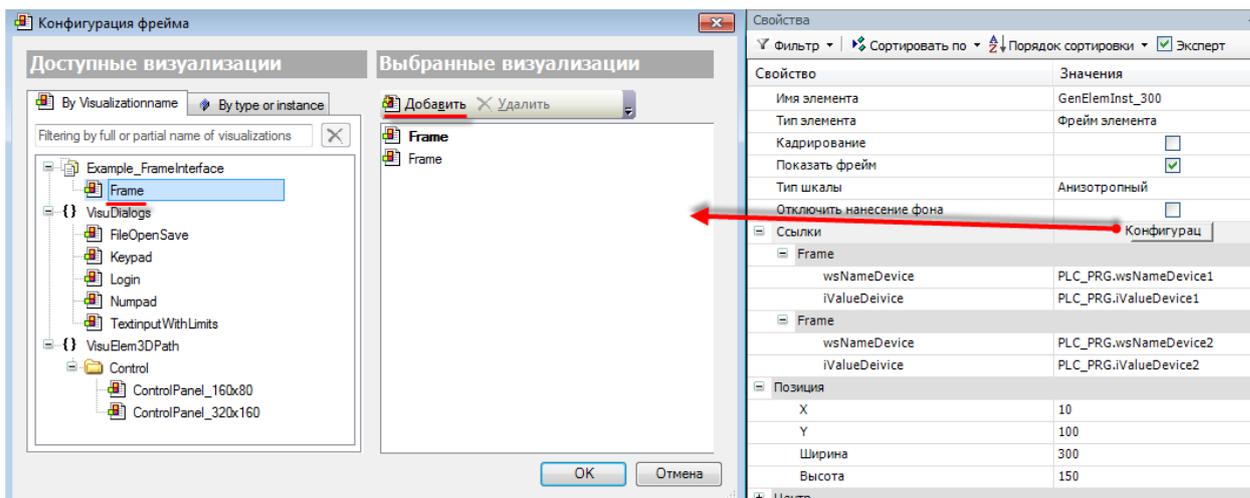


Рис. 11.3.3.7. Параметры и конфигурация элемента **Фрейм**

Настроим кнопку **Устройство 1**.

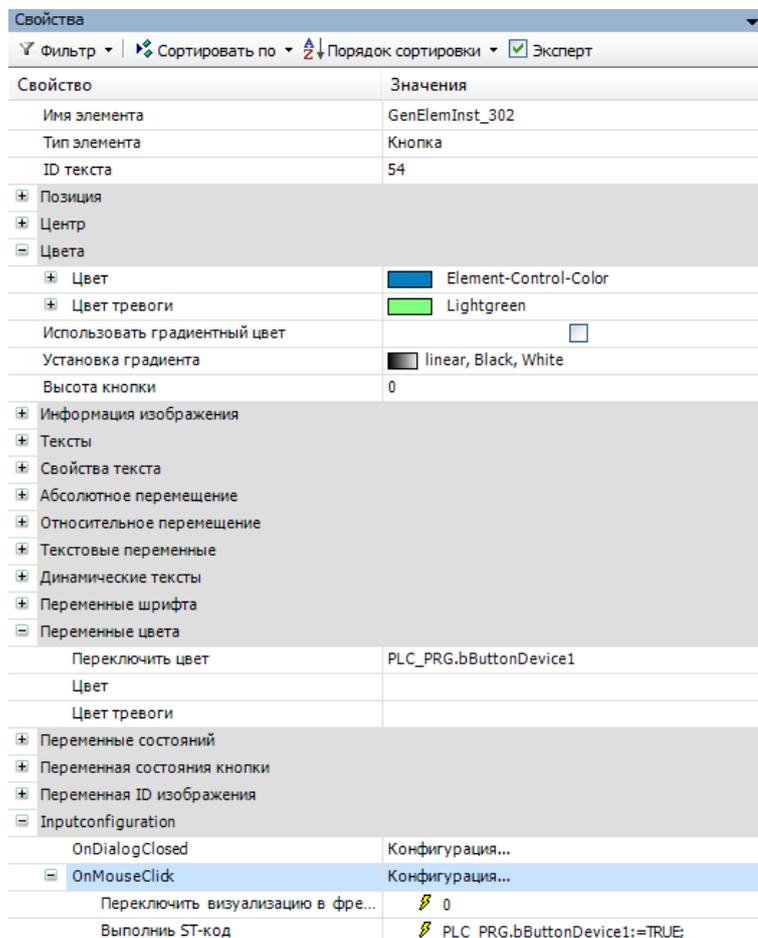


Рис. 11.3.3.8. Параметры кнопки **Устройство 1**

На вкладке [InputConfiguration](#) к параметру **OnMouseClicked** привяжем действия [Переключить визуализацию во фрейме](#) и [Выполнить ST-код](#) (для изменения цвета кнопки при выборе экрана устройства):

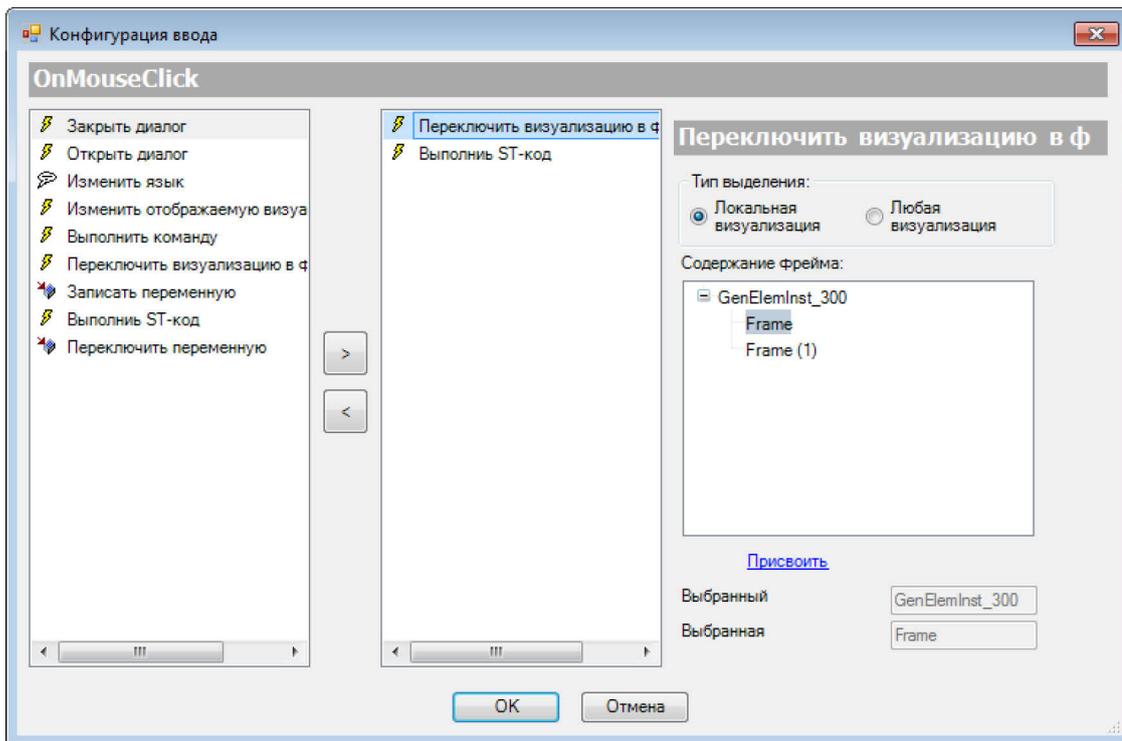


Рис. 11.3.3.9. Настройки действия кнопки **Устройство 1** (Переключить визуализацию в фрейме)

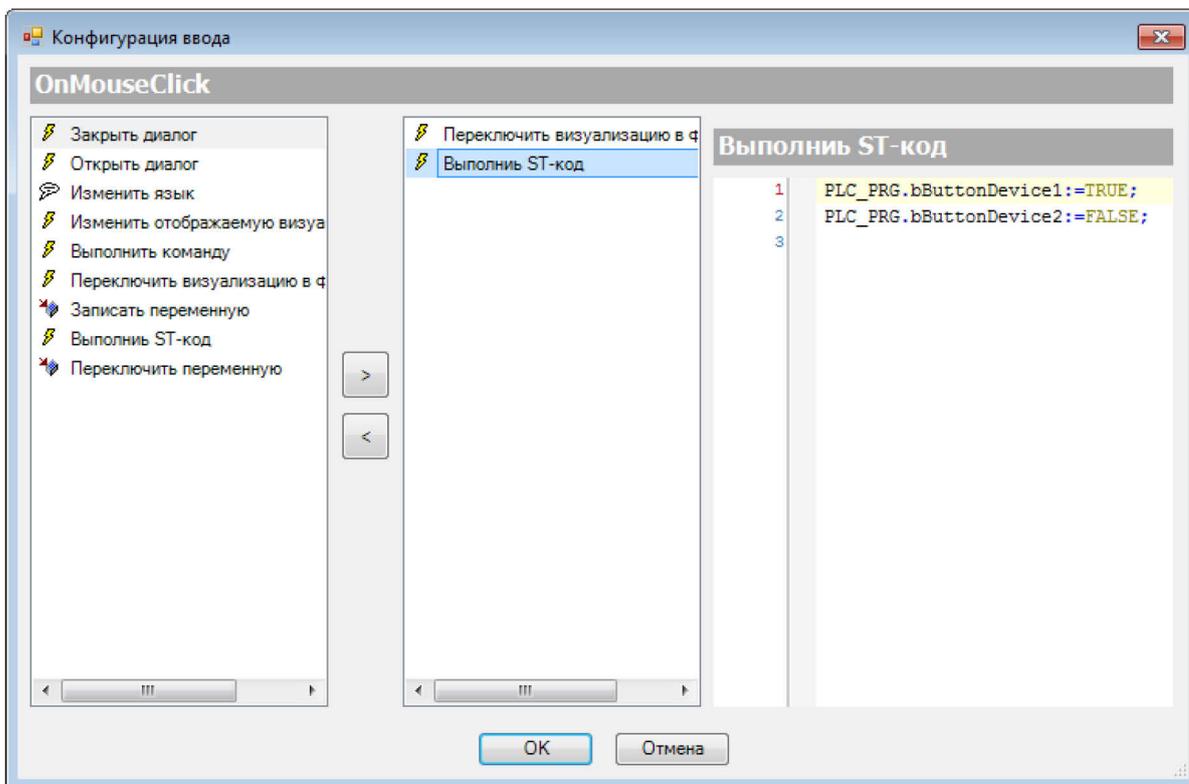


Рис. 11.3.3.10. Настройки действия кнопки **Устройство 1** (Выполнить ST-код)

Настроим кнопку **Устройство 2**.

Свойства	
Ү Фильтр Сортировать по Порядок сортировки Эксперт	
Свойство	Значения
Имя элемента	GenElemInst_304
Тип элемента	Кнопка
ID текста	55
⊕ Позция	
⊕ Центр	
⊖ Цвета	
⊕ Цвет	 Element-Control-Color
⊕ Цвет тревоги	 Lightgreen
Использовать градиентный цвет	<input type="checkbox"/>
Установка градиента	 linear, Black, White
Высота кнопки	0
⊕ Информация изображения	
⊕ Тексты	
⊕ Свойства текста	
⊕ Абсолютное перемещение	
⊕ Относительное перемещение	
⊕ Текстовые переменные	
⊕ Динамические тексты	
⊕ Переменные шрифта	
⊖ Переменные цвета	
Переключить цвет	PLC_PRG.bButtonDevice2
Цвет	
Цвет тревоги	
⊕ Переменные состояний	
⊕ Переменная состояния кнопки	
⊕ Переменная ID изображения	
⊖ Inputconfiguration	
OnDialogClosed	Конфигурация...
⊖ OnMouseClicked	Конфигурация...
Переключить визуализацию в фре...	 1
Выполни ST-код	 PLC_PRG.bButtonDevice1:=FALSE;

Рис. 11.3.3.11. Параметры элемента **Устройство 2**

На вкладке [InputConfiguration](#) к параметру **OnClick** привяжем действия [Переключить визуализацию во фрейме](#) и [Выполнить ST-код](#) (для изменения цвета кнопки при выборе экрана устройства):

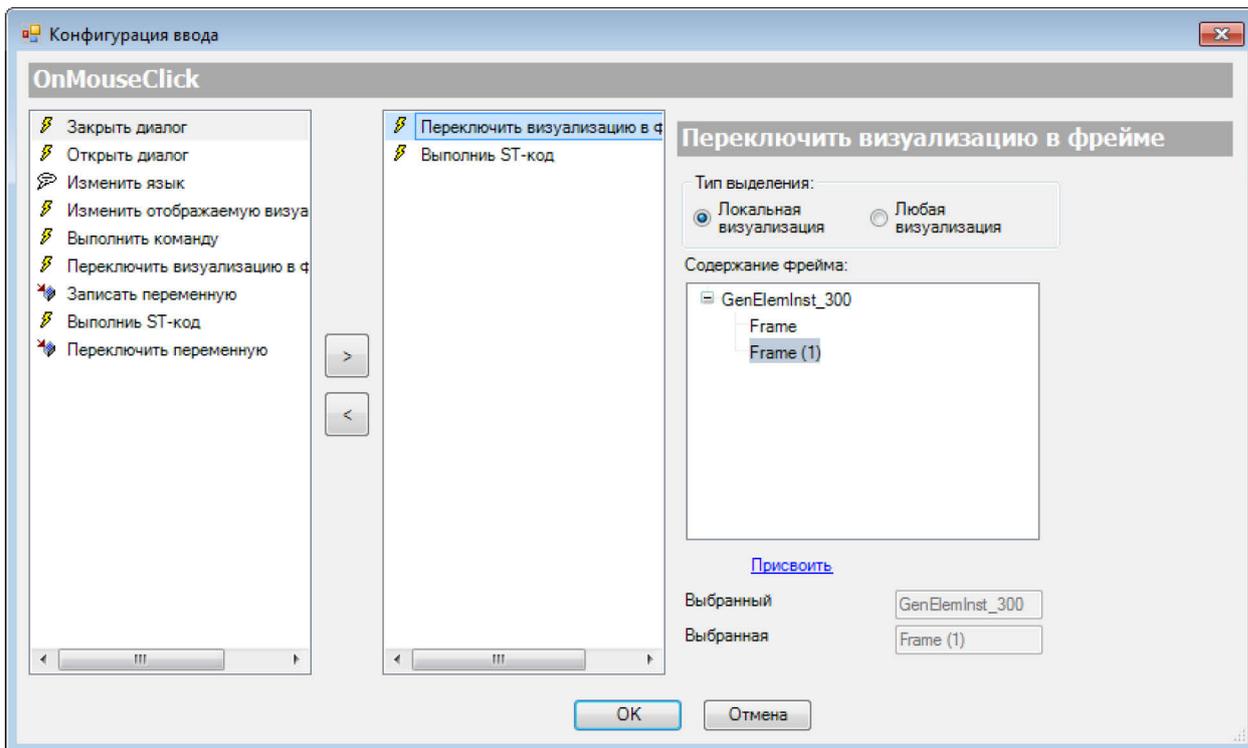


Рис. 11.3.3.12. Настройки действия кнопки **Устройство 2** (Переключить визуализацию в фрейме)

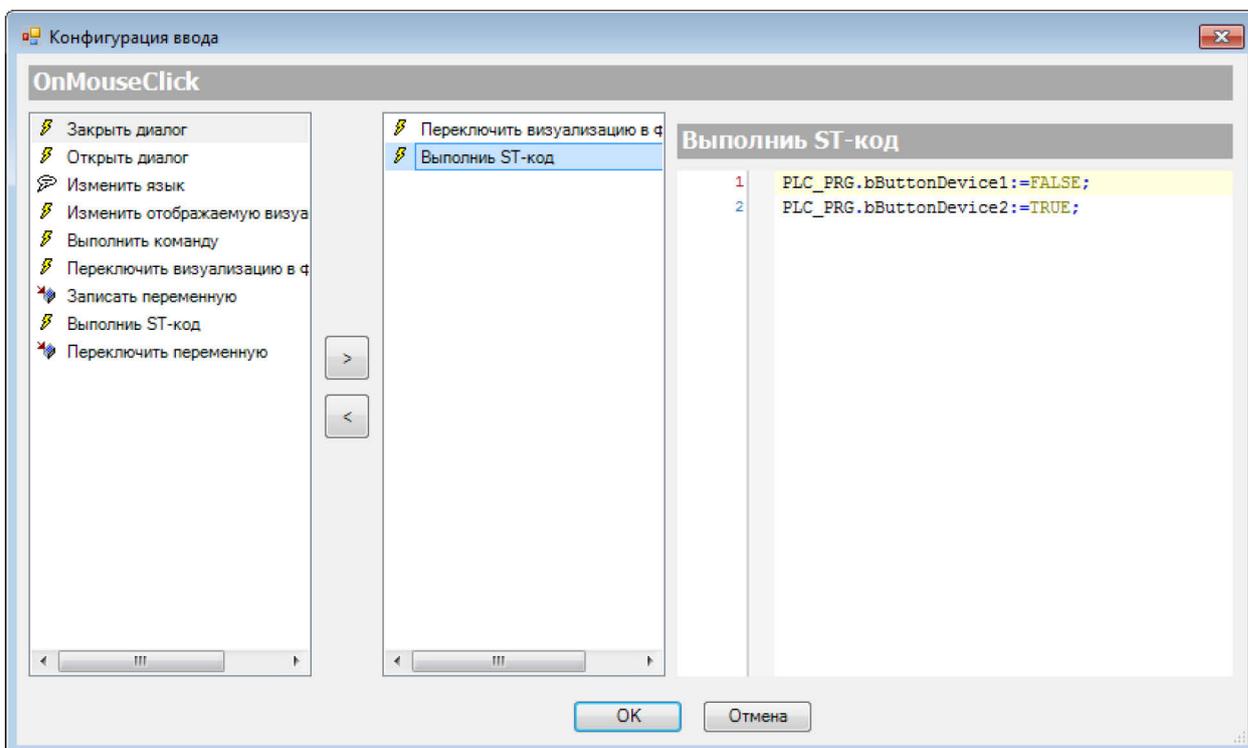
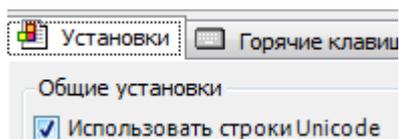


Рис. 11.3.3.13. Настройки действия кнопки **Устройство 2** (Выполнить ST-код)

6. Настроим [Менеджер визуализации](#). В его установках поставим галочку **Использовать строки Unicode**:



Настройки **target**- и **web**-визуализации будут выглядеть следующим образом:

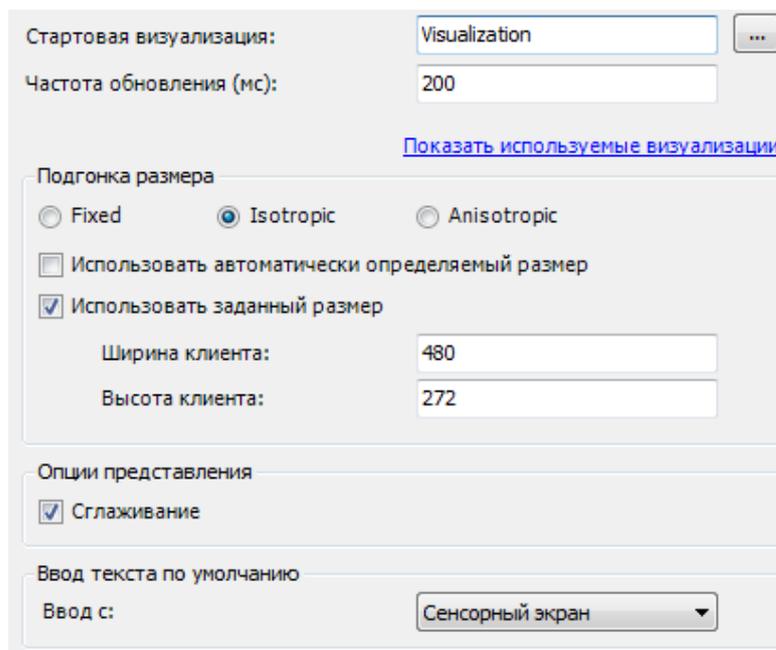


Рис. 11.3.3.14. Настройки **target**-визуализации

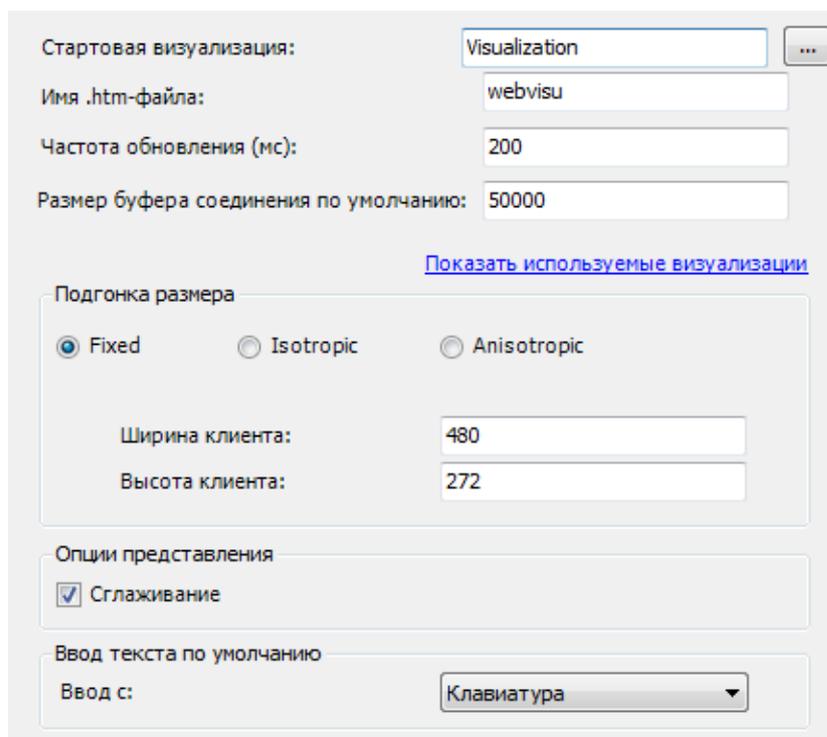


Рис. 11.3.3.15. Настройки **web**-визуализации

7. [Запустим проект на виртуальном контроллере](#). Откроем в веб-браузере страницу <http://localhost:8080/webvisu.htm>, чтобы увидеть web-визуализацию проекта. **Обратите внимание**, что браузер должен поддерживать **HTML5**. Проверим функционал проекта.

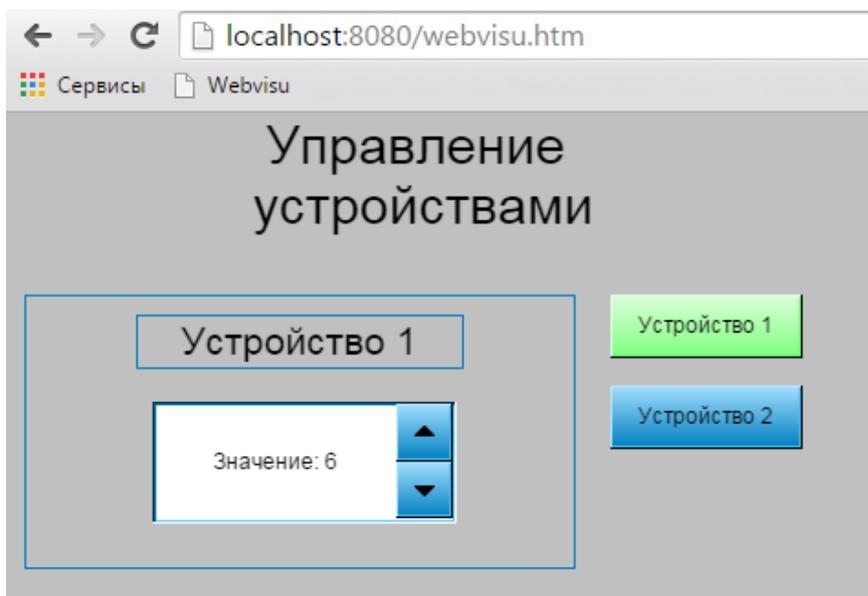


Рис. 11.3.3.16. Web-визуализация проекта в браузере **Chrome**

По умолчанию во фрейме отображается экран устройства **1**. Введите значение (например, 6). Нажмите кнопку **Устройство 2**, чтобы открыть во фрейме другой экземпляр экрана визуализации:

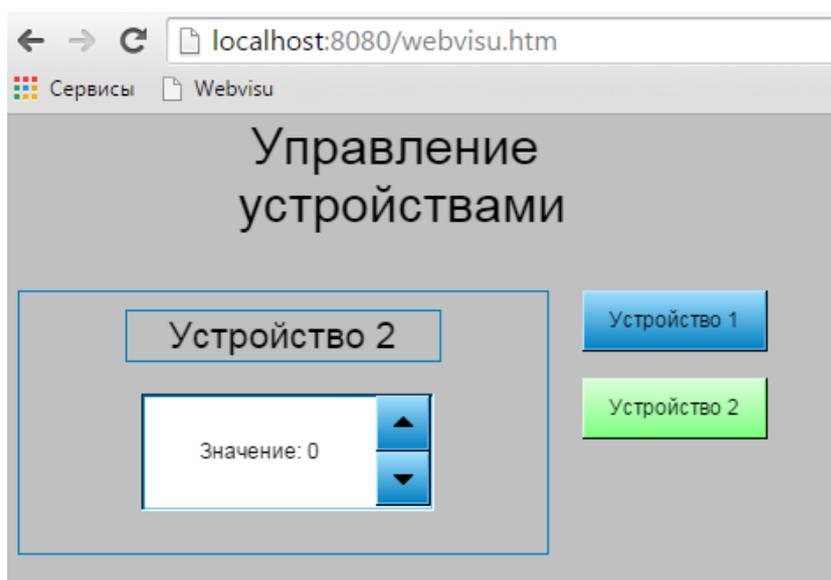


Рис. 11.3.3.17. Переключение экземпляра экрана во фрейме

Опять переключитесь на устройство **1**, чтобы увидеть, что ранее введенное вами значение сохранилось.

11.3.4. Создание анимации (элемент Изображение)

Данный пример посвящен созданию анимации с помощью элемента [Изображение](#).



Рис. 11.3.4.1. Внешний вид примера **Анимация** (кадры 1,2,3 и 4)

Пример создан в среде **CODESYS 3.5 SP6 Patch 2** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с target-файлом **3.5.6.0**.

Пример доступен для скачивания: [Example_Animation.projectarchive](#)

1. Создадим новый **стандартный** проект **CODESYS** с названием **Example_Animation** и настройками по умолчанию: target – **CODESYS Control Win V3**, язык PLC_PRG – **ST**.

2. Добавим в проект компонент **Пул изображений** с названием по умолчанию **ImagePool**:

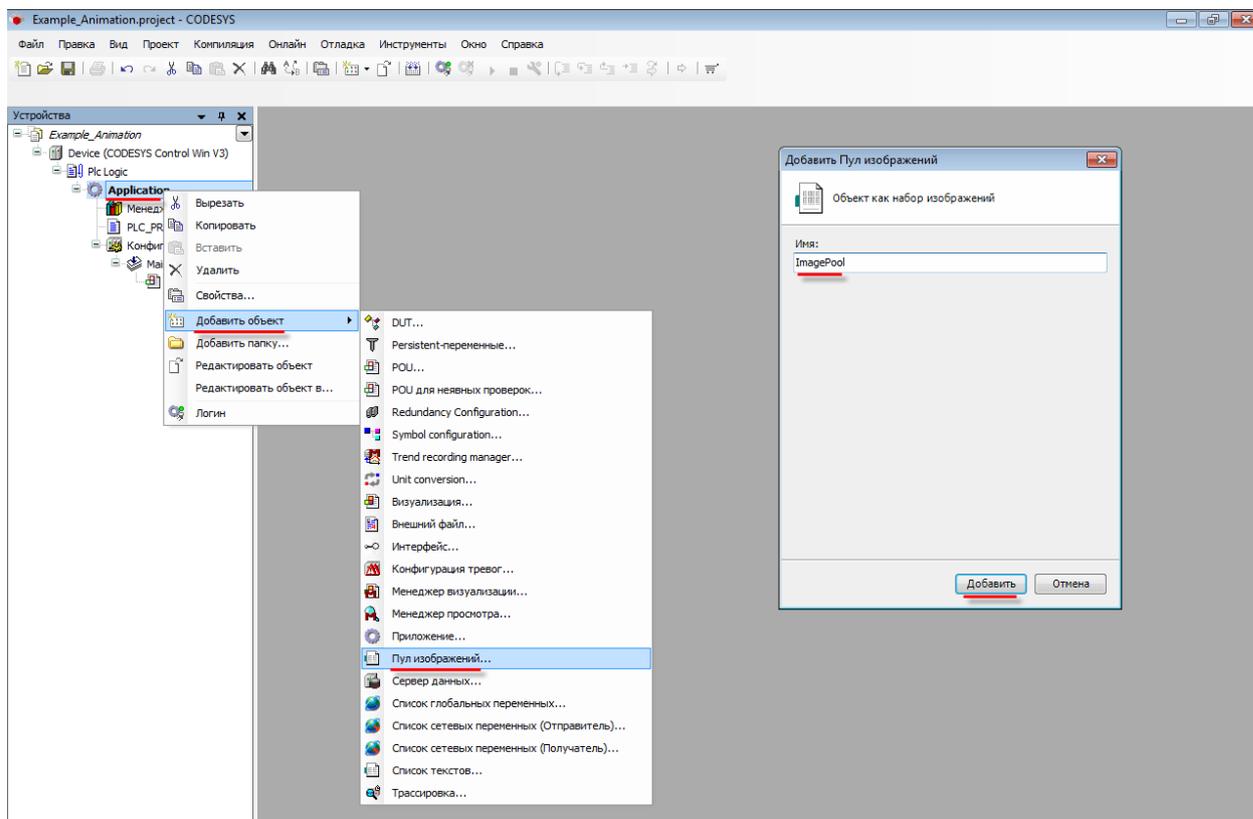


Рис. 11.3.4.2. Добавление в проект пула изображений

3. Добавим в пул изображений графические файлы, которые будут использоваться в качестве кадров анимации:

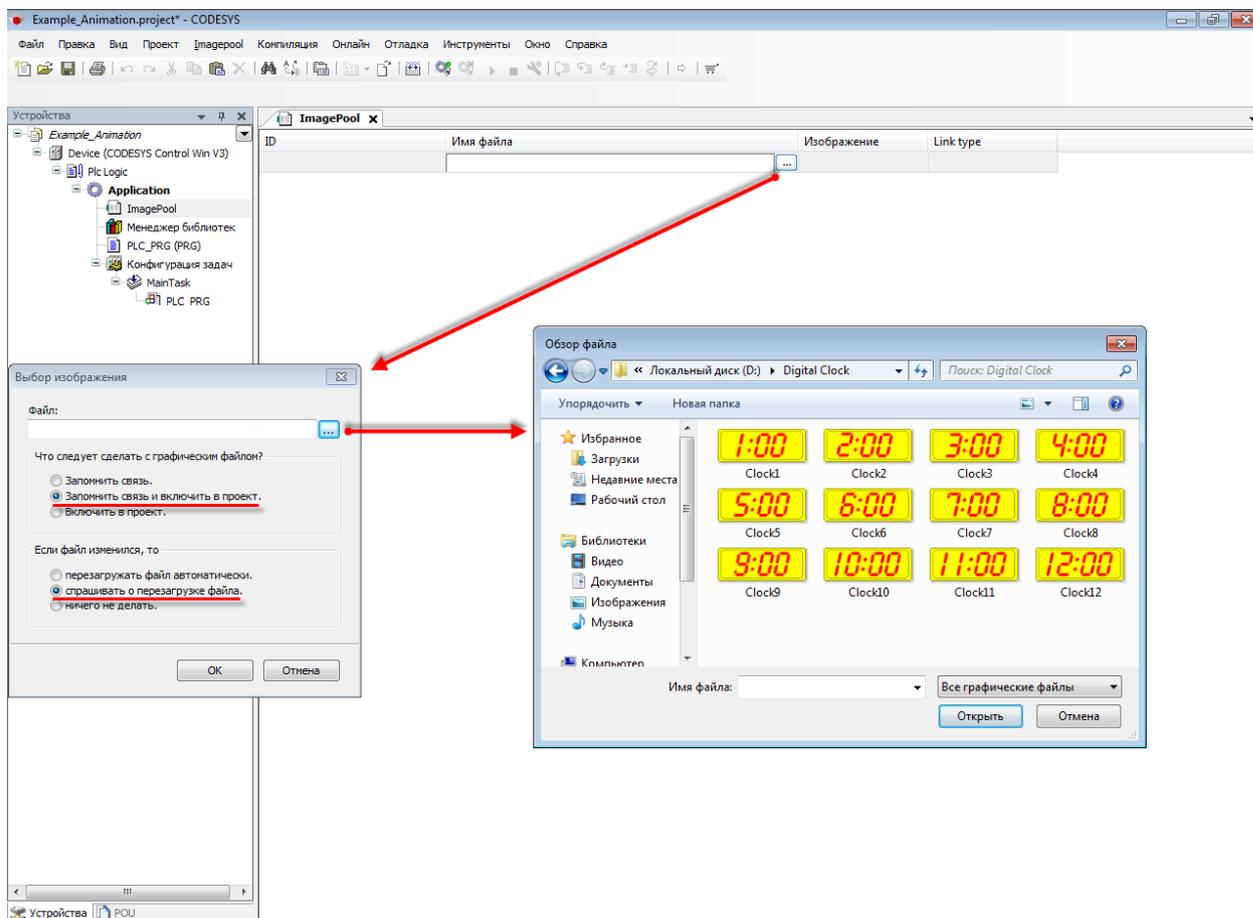


Рис. 11.3.4.3. Добавление графических файл в пул изображений

ID	Имя файла	Изображение	Link type
Clock1	Clock1.png	1:00	Embedded and link to file
Clock2	Clock2.png	2:00	Embedded and link to file
Clock3	Clock3.png	3:00	Embedded and link to file
Clock4	Clock4.png	4:00	Embedded and link to file
Clock5	Clock5.png	5:00	Embedded and link to file
Clock6	Clock6.png	6:00	Embedded and link to file
Clock7	Clock7.png	7:00	Embedded and link to file
Clock8	Clock8.png	8:00	Embedded and link to file
Clock9	Clock9.png	9:00	Embedded and link to file
Clock10	Clock10.png	10:00	Embedded and link to file
Clock11	Clock11.png	11:00	Embedded and link to file
Clock12	Clock12.png	12:00	Embedded and link to file

Рис. 11.3.4.4. Внешний вид пула изображений после добавления графических файлов

4. В программе **PLC_PRG** объявим следующие переменные:

```
1 PROGRAM PLC_PRG
2 VAR
3     sImageID:STRING; // переменная, которая содержит ID пула изображений для текущего кадра
4     iImageIndex:INT:=1; // номер текущего кадра
5     bStart:BOOL; // переменная запуска/остановки анимации
6     bReset:BOOL; // переменная сброса анимации на первый кадр
7 END VAR
```

Рис. 11.3.4.5. Объявление переменных программы **PLC_PRG**

5. Код программы **PLC_PRG** будет выглядеть следующим образом:

```
1 // реализация циклической прокрутки индексов кадров анимации с учетом возможности сброса
2
3 IF bReset=TRUE THEN
4     iImageIndex:=1;
5     bReset:=FALSE;
6     ELSIF bStart=TRUE AND bReset=FALSE AND iImageIndex<>12 THEN
7         iImageIndex:=iImageIndex+1;
8     ELSIF iImageIndex=12 THEN
9         iImageIndex:=1;
10 END_IF
11
12 // склеивание фиксированной части ID изображения с индексом текущего кадра...
13 // ...в переменную типа STRING, которая привязывается к элементу Изображения
14
15 sImageID:=CONCAT('clock', INT_TO_STRING(iImageIndex));
16
```

Рис. 11.3.4.6. Код программы **PLC_PRG**

6. Добавим в проект экран визуализации **Visualization**. В его свойствах выберем размер **480x272**. Экран будет содержать элемент Изображение и два элемента Клавишный переключатель. Названия переключателей сделаны с помощью элементов Метка.

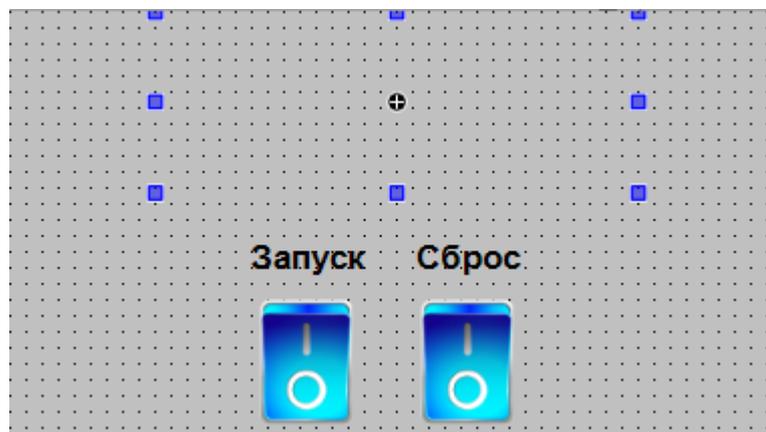


Рис. 11.3.4.7. Содержимое экрана **Visualization**

Настроим элемент [Изображение](#). К параметру **ID изображения** привяжем переменную **sImageID**:

Свойство	Значения
Имя элемента	GenElemInst_300
Тип элемента	Изображение
Статический ID	
Показать фрейм	<input type="checkbox"/>
Кадрирование	<input type="checkbox"/>
Прозрачный	<input checked="" type="checkbox"/>
Прозрачный цвет	<input type="text" value="White"/>
Тип шкалы	Анизотропный
Позиция	
X	90
Y	0
Ширина	300
Высота	113
Центр	
Цвета	
Вид элемента	
Тексты	
Текст	
Подсказка	
Свойства текста	
Переменная ID изображения	
ID изображения	PLC_PRG.sImageID

Рис. 11.3.4.8. Параметры элемента **Изображение**

К [клавишному переключателю](#) **Запуск** привяжем переменную **bStart**, а к переключателю **Сброс** – **bReset**:

Свойство	Значения
Имя элемента	GenElemInst_304
Тип элемента	Клавишный выключатель
Позиция	
X	155
Y	180
Ширина	60
Высота	90
Переменная	PLC_PRG.bStart
Параметры изображения	
Прозрачный	<input type="checkbox"/>
Прозрачный цвет	<input type="text" value="Black"/>
Изотропный тип	Изотропия
Горизонтальное выравнив...	Лево
Вертикальное выравнив...	Верх
Поведение элемента	Переключатель изображения
Тексты	
Переменные состояний	
Фон	
Изображение	Blue

Рис. 11.3.4.9. Параметры переключателя **Старт**

Свойство	Значения
Имя элемента	GenElemInst_306
Тип элемента	Клавишный выключатель
Позиция	
X	255
Y	180
Ширина	60
Высота	90
Переменная	PLC_PRG.bReset
Параметры изображения	
Прозрачный	<input type="checkbox"/>
Прозрачный цвет	Black
Изотропный тип	Изотропия
Горизонтальное выравн...	Лево
Вертикальное выравнив...	Верх
Поведение элемента	Переключатель изображения
Тексты	
Переменные состояний	
Фон	
Изображение	Blue

Рис. 11.3.4.10. Параметры переключателя **Сброс**

7. Теперь настроим частоту смены кадров анимации. Для этого откроем компонент **Конфигурация тревог** и настроим время цикла задачи **MainTask**, к которой привязана программа **PLC_PRG**:

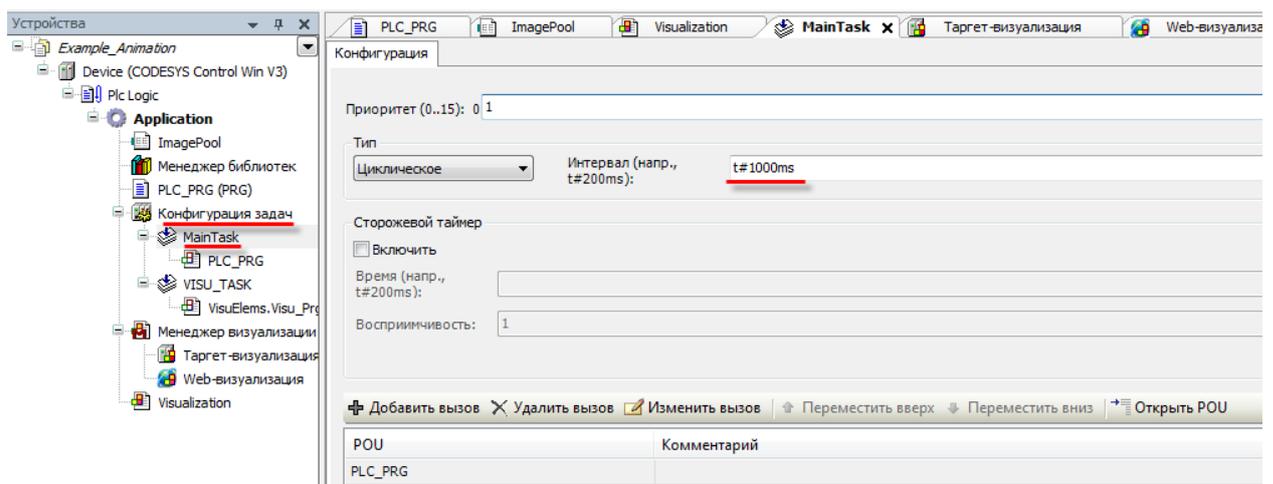
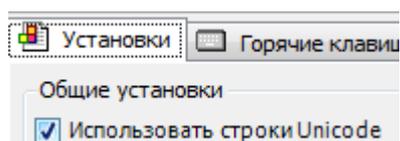


Рис. 11.3.4.11. Настройка цикла задачи **MainTask** (к которой привязана программа **PLC_PRG**)

8. Настроим [Менеджер визуализации](#). В его установках поставим галочку **Использовать строки Unicode**:



Настройки **target**- и **web**-визуализации будут выглядеть следующим образом:

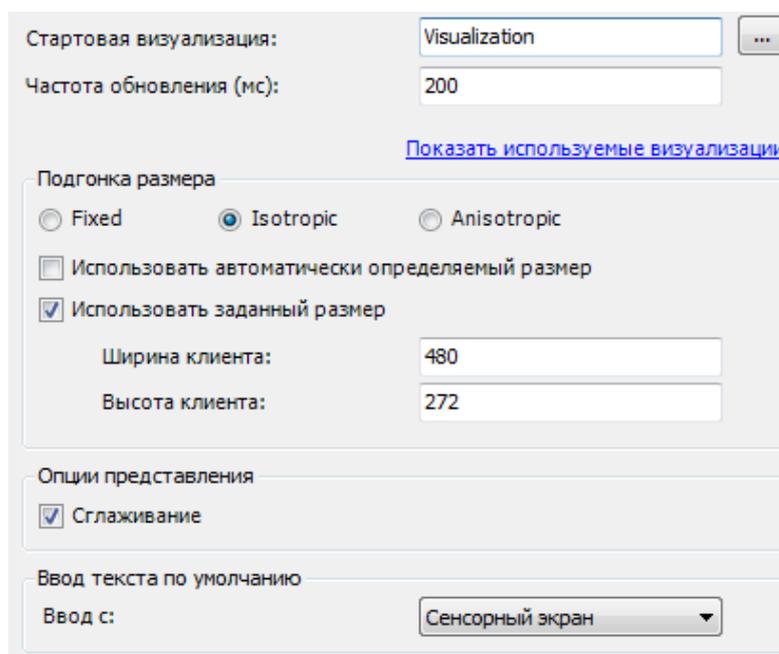


Рис. 11.3.4.12. Настройки **target**-визуализации

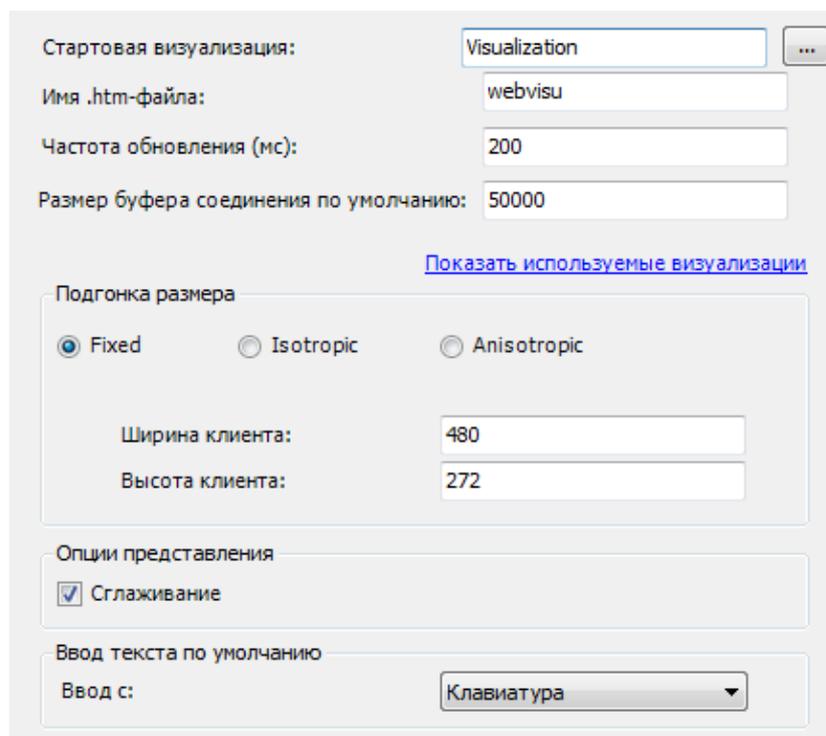


Рис. 11.3.4.13. Настройки **web**-визуализации

9. [Запустим проект на виртуальном контроллере](#). Откроем в веб-браузере страницу <http://localhost:8080/webvisu.htm>, чтобы увидеть web-визуализацию проекта. **Обратите внимание**, что браузер должен поддерживать **HTML5**. Проверим функционал проекта.

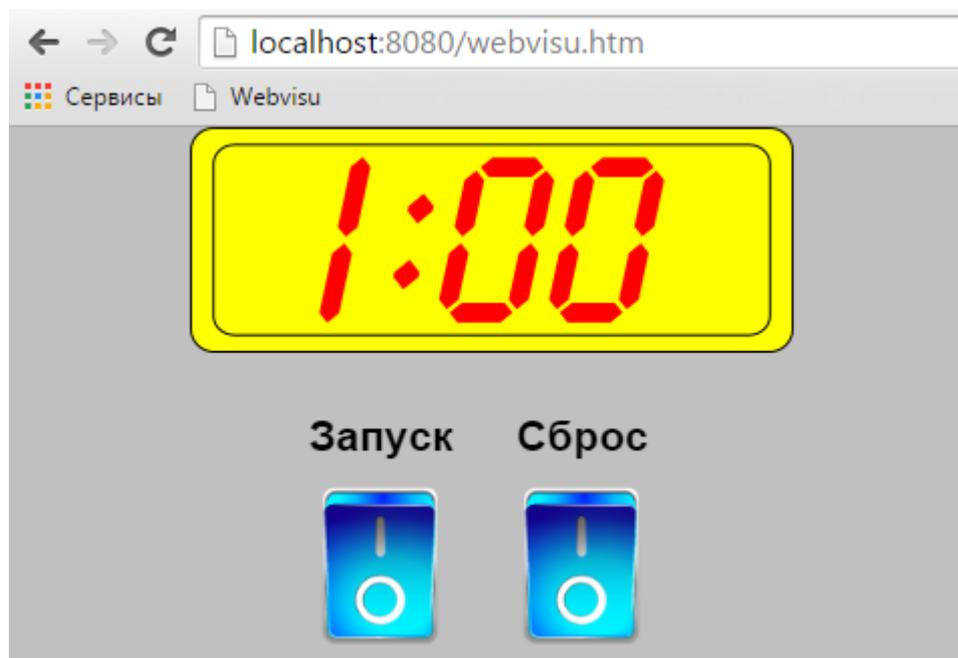


Рис. 11.3.4.14 Web-визуализация проекта в браузере **Chrome**

Нажмите переключатель **Запуск**, чтобы запустить анимацию. Повторное нажатие переключателя остановит ее.

Нажатие на переключатель **Сброс** приведет к сбросу анимации на первый кадр.

11.3.5. Создание мультиязычного проекта

Данный пример посвящен созданию *мультиязычного* проекта – т.е. проекта, язык текстов которого переключается по команде оператора или в процессе функционирования программы целевого устройства.

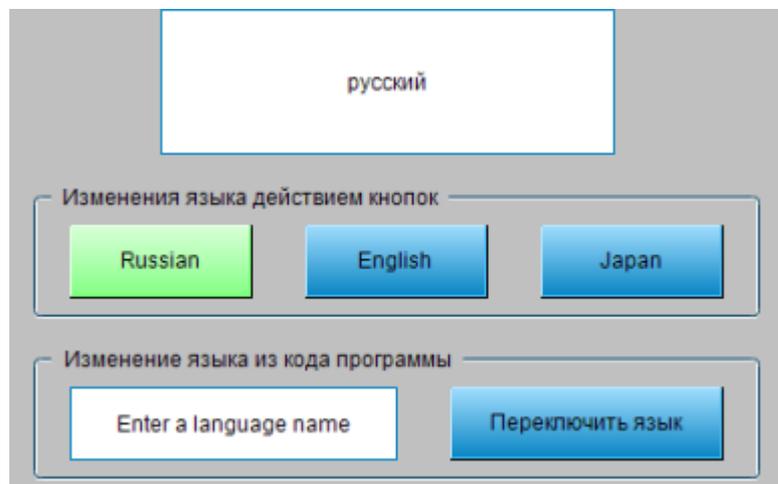


Рис. 11.3.5.1. Внешний вид примера **Мультиязычность**

Пример создан в среде **CODESYS 3.5 SP6 Patch 2** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с target-файлом **3.5.6.0**.

Пример доступен для скачивания: [Example_Multilanguage_projectarchive](#)

1. Создадим новый **стандартный** проект **CODESYS** с названием **Example_Multilanguage** и настройками по умолчанию: target – **CODESYS Control Win V3**, язык PLC_PRG – **ST**.

2. Добавим в проект компонент **Список текстов** с названием по умолчанию **TextList**:

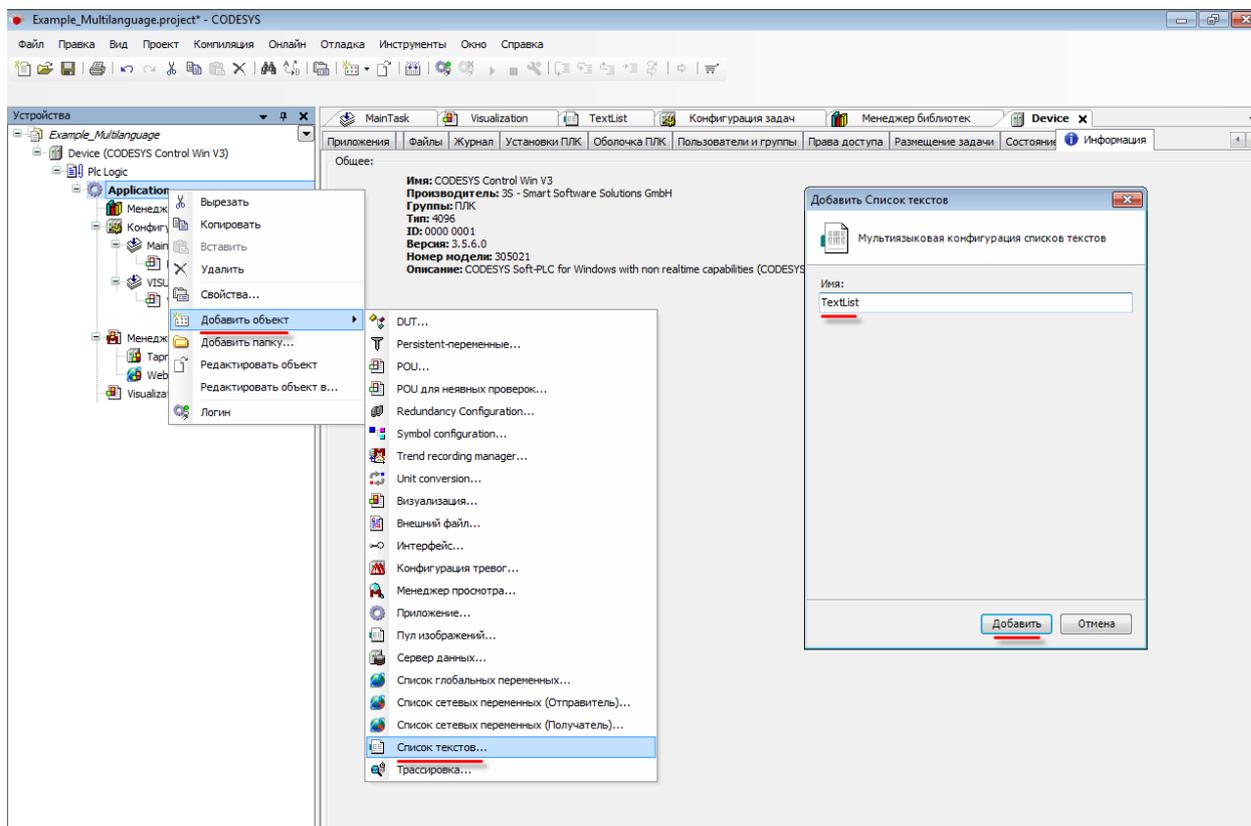


Рис. 11.3.5.2. Добавление в проект списка текстов

3. Нажмем на любую область списка текстов **ПКМ** и выберем команду **Добавить язык**:

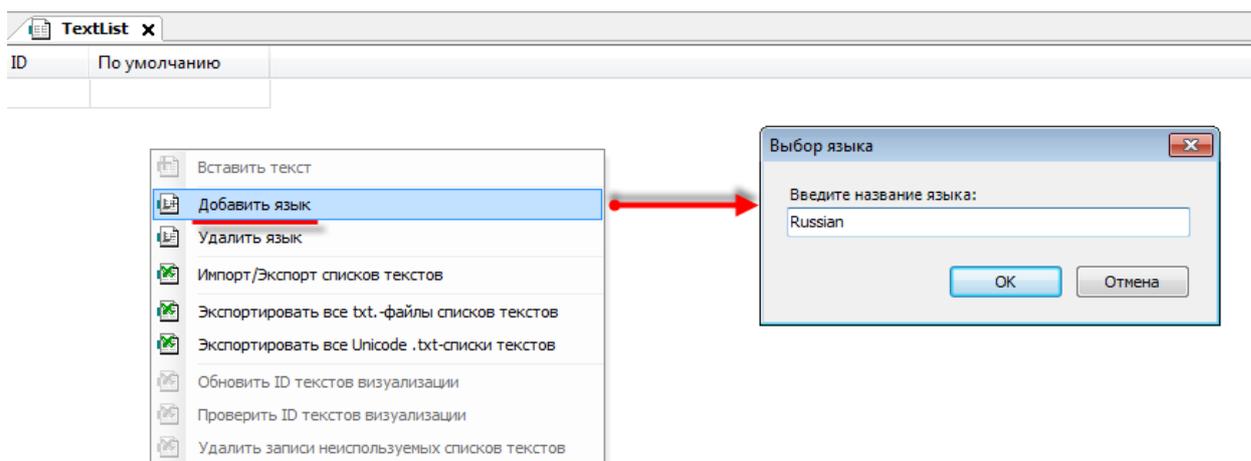


Рис. 11.3.5.3. Добавление языка в список текстов

Добавьте в список три языка с названиями **Russian**, **English** и **Japan**.

4. Добавим в список текст со следующими настройками:

ID	По умолчанию	Russian	English	Japan
0		русский	english	日本語

Рис. 11.3.5.4. Содержимое списка текстов

5. В программе **PLC_PRG** объявим следующие переменные:

```
1 PROGRAM PLC_PRG
2 VAR
3     bRussian:BOOL:=TRUE;           // переменная цвета кнопки Russian
4     bEnglish:BOOL;                 // переменная цвета кнопки English
5     bJapan:BOOL;                   // переменная цвета кнопки Japan
6
7     sSetLanguageName:STRING:='Enter a language name'; // переменная, в которую записывается...
8                                     // ...имя языка, выбранного пользователем
9     bSetLanguage:BOOL;             // переменная выбора языка
10 END VAR
```

Рис. 11.3.5.5. Объявление переменных программы **PLC_PRG**

6. Код программы **PLC_PRG** будет выглядеть следующим образом:

```
1 // переключение языка проект на язык, выбранный пользователем
2
3 IF bSetLanguage=TRUE THEN
4     VisuElems.CURRENTLANGUAGE:=sSetLanguageName;
5 END_IF
6
7 // окраска кнопки выбранного языка
8
9 IF VisuElems.CURRENTLANGUAGE='russian' OR VisuElems.CURRENTLANGUAGE='RUSSIAN' THEN
10     bRussian:=TRUE;
11     bEnglish:=FALSE;
12     bJapan:=FALSE;
13 ELSIF VisuElems.CURRENTLANGUAGE='english' OR VisuElems.CURRENTLANGUAGE='ENGLISH' THEN
14     bEnglish:=TRUE;
15     bRussian:=FALSE;
16     bJapan:=FALSE;
17 ELSIF VisuElems.CURRENTLANGUAGE='japan' OR VisuElems.CURRENTLANGUAGE='JAPAN' THEN
18     bJapan:=TRUE;
19     bRussian:=FALSE;
20     bEnglish:=FALSE;
21 END_IF
```

Рис. 11.3.5.6. Код программы **PLC_PRG**

7. [Добавим в проект](#) экран визуализации **Visualization**. В его [свойствах](#) выберем размер **480x272**. Экран будет содержать два элемента [Текстовое поле](#), четыре элемента [Кнопка](#) и два элемента [Группа](#).

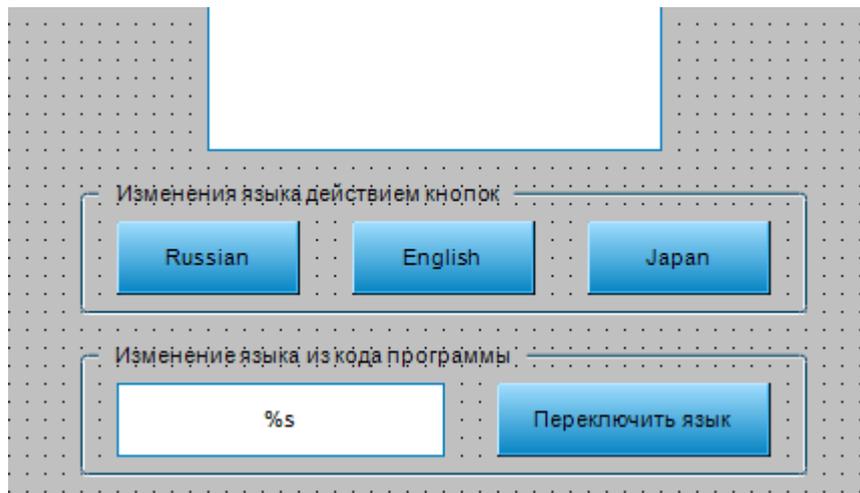


Рис. 11.3.5.7. Содержимое экрана **Visualization**

Настроим текстовое поле (верхнее), в котором будет отображаться мультязычный текст:

Свойства	
Фильтр Сортировать по Порядок сортировки Эксперт	
Свойство	Значения
Имя элемента	GenElemInst_316
Тип элемента	Текстовое поле
Позиция	
X	110
Y	0
Ширина	250
Высота	80
Цвета	
Вид элемента	
Тип тени	Из стиля
Тексты	
Свойства текста	
Текстовые переменные	
Динамические тексты	
Список текстов	'TextList'
Индекс текста	0
Индекс подсказки	

Рис. 11.3.5.8. Параметры элемента **Текстовое поле**

Настроим кнопки группы **Изменение языка действием кнопок**.

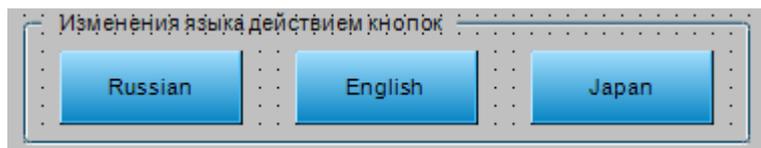


Рис. 11.3.5.9. Элементы группы Изменение языка действием кнопок

Во вкладке **Переменные состояния** к параметру **Переключить цвет** привяжем переменные **bRussian**, **bEnglish**, **bJapan** соответственно:

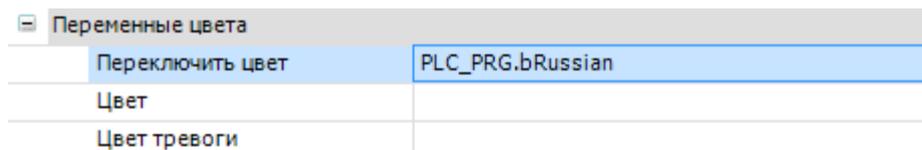


Рис. 11.3.5.10. Параметры кнопки **Russian**

Во вкладке [InputConfiguration](#) к параметру **OnMouseClicked** привяжем действия [Изменить язык](#) и [Выполнить ST-код](#):

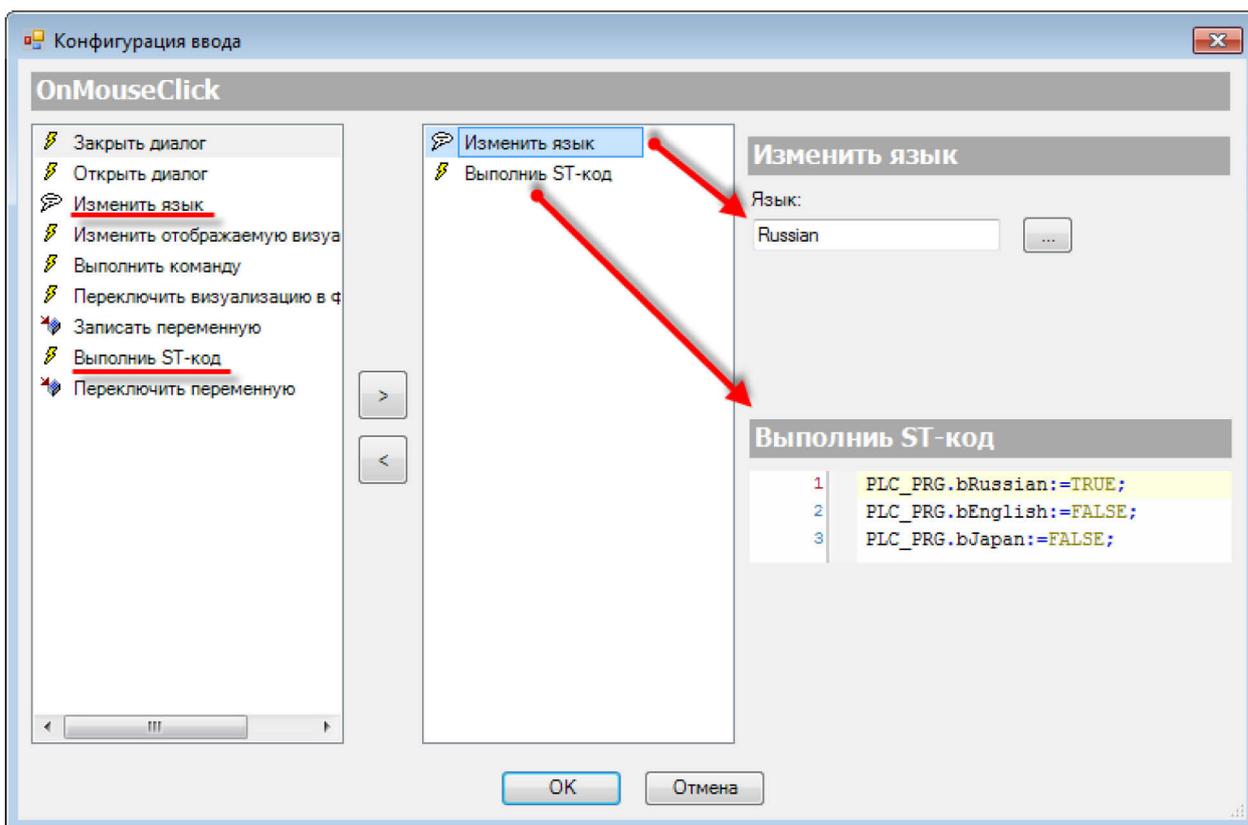


Рис. 11.3.5.11. Настройки действия кнопки **Russian**

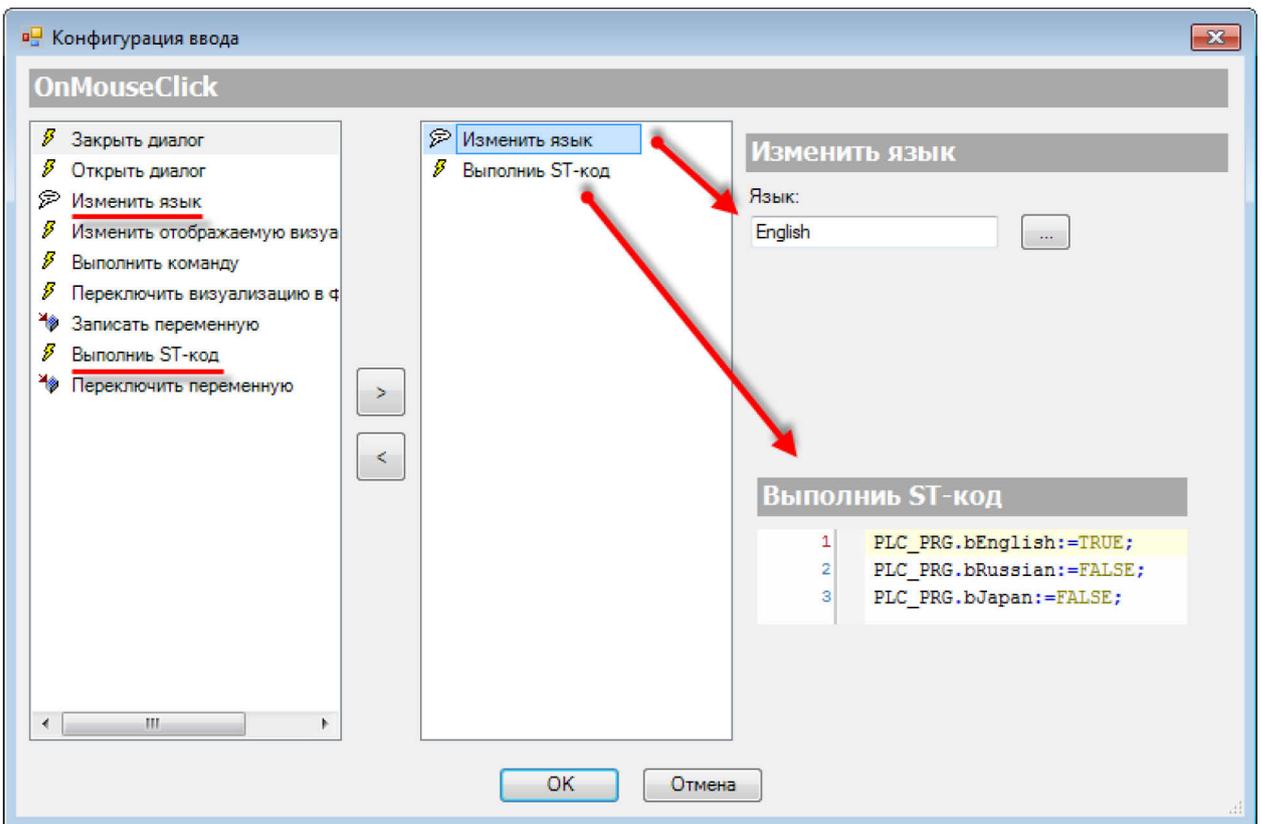


Рис. 11.3.5.12. Настройки действия кнопки **English**

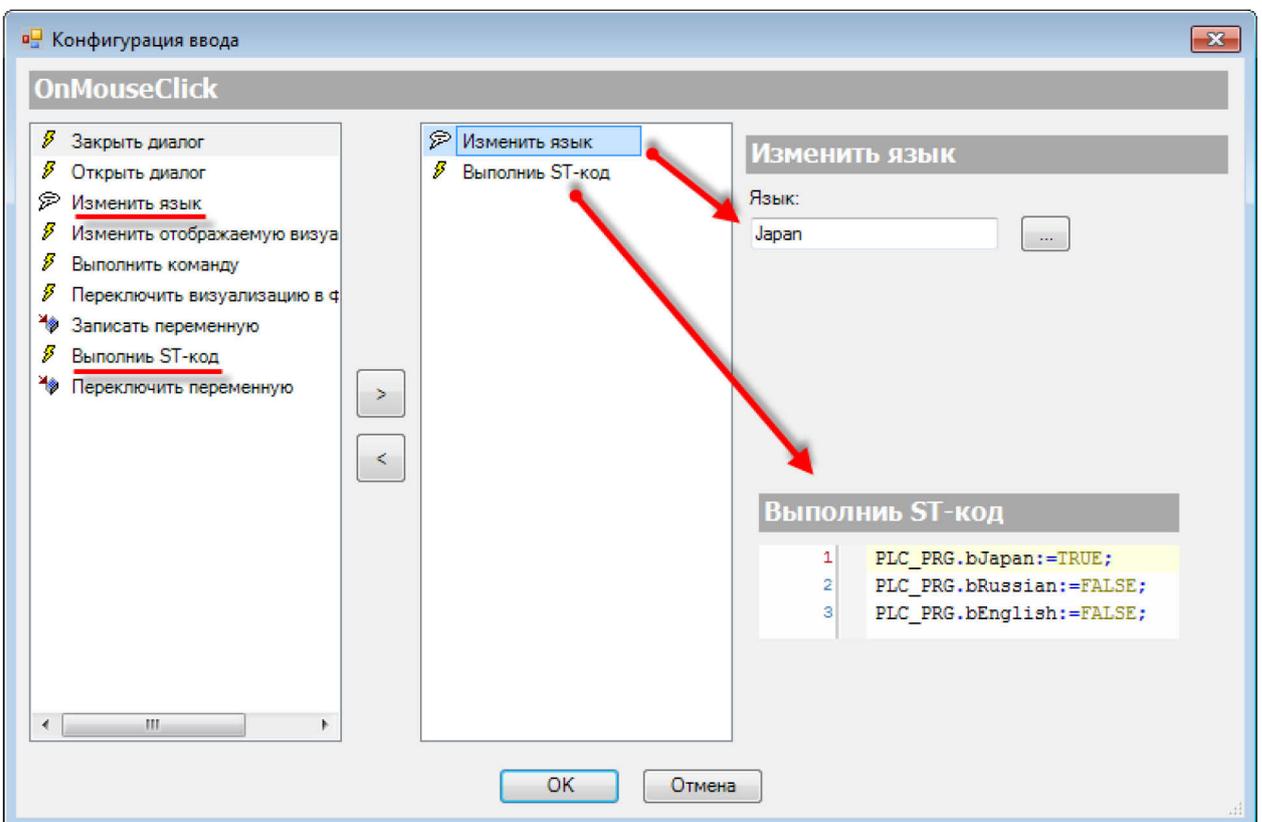


Рис. 11.3.5.13. Настройки действия кнопки **Japan**

Настроим элементы группы **Изменение языка из кода программы**.

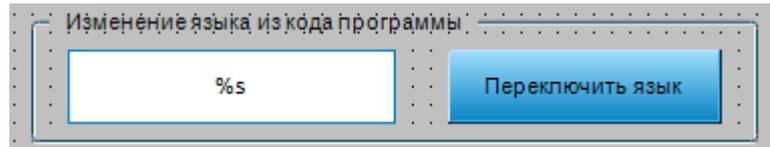


Рис. 11.3.5.14. Элементы группы **Изменение языка из кода программы**

К полю ввода языка привяжем переменную **sSetLanguageName**:

Свойства	
Свойство	Значения
Имя элемента	GenElemInst_324
Тип элемента	Текстовое поле
ID текста	6
ID подсказки	11
Позиция	
X	60
Y	210
Ширина	180
Высота	40
Цвета	
Вид элемента	
Тип тени	Из стиля
Тексты	
Текст	%s
Подсказка	Введите RUSSIAN, ENGLISH или JAPAN
Свойства текста	
Горизонтальное выравнива...	По центру
Вертикальное выравнивание	По центру
Шрифт	Font-Standard
Цвет шрифта	Font-Default-Color
Transparency	255
Текстовые переменные	
Текстовая переменная	PLC_PRG.sSetLanguageName
Переменная подсказки	
Динамические тексты	
Переменные шрифта	
Переменные цвета	
Переменные состояний	
Inputconfiguration	
OnDialogClosed	Конфигурация...
OnMouseClicked	Конфигурация...
Записать переменную	Variable : , InputType: VisuDialogs.Keypa

Рис. 11.3.5.15. Параметры элемента **Текстовое поле**

У элемента [Текстовое поле](#) во вкладке [InputConfiguration](#) к параметру **OnMouseClicked** привяжем действие [Записать переменную](#):

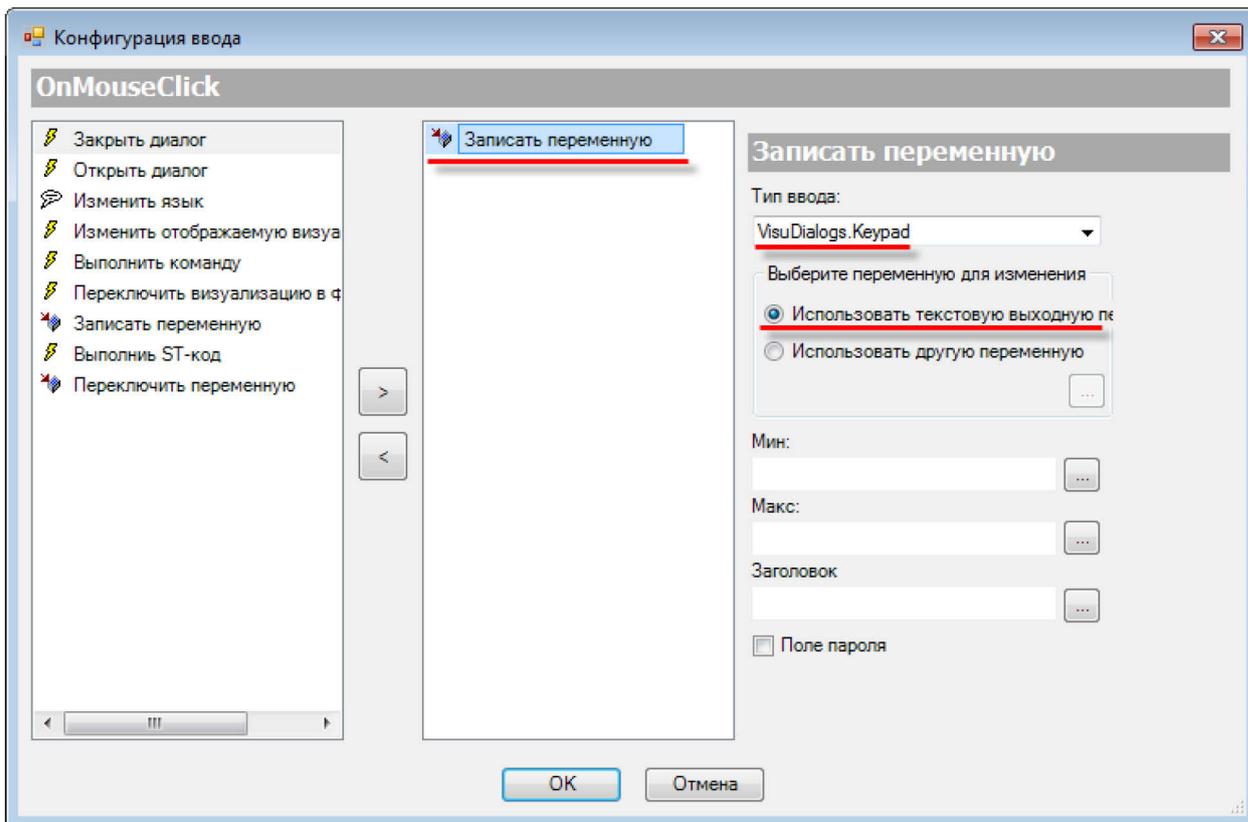


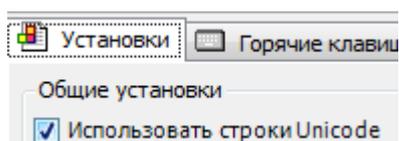
Рис. 11.3.5.16. Настройки действия элемента **Текстовое поле**

К кнопке **Переключить язык** привяжем переменную **bSetLanguage**:

Свойства	
Фильтр Сортировать по Порядок сортировки Эксперт	
Свойство	Значения
Имя элемента	GenElemInst_326
Тип элемента	Кнопка
ID текста	7
+ Позиция	
+ Центр	
+ Цвета	
Использовать градиентный цвет	<input type="checkbox"/>
Установка градиента	linear, Black, White
Высота кнопки	0
+ Информация изображения	
- Тексты	
Текст	Переключить язык
Подсказка	
+ Свойства текста	
+ Абсолютное перемещение	
+ Относительное перемещение	
+ Текстовые переменные	
+ Динамические тексты	
+ Переменные шрифта	
+ Переменные цвета	
- Переменные состояний	
Невидимый	
Отключение ввода	
+ Переменная состояния кнопки	
+ Переменная ID изображения	
- Inputconfiguration	
OnDialogClosed	Конфигурация...
OnMouseClicked	Конфигурация...
OnMouseDown	Конфигурация...
OnMouseEnter	Конфигурация...
OnMouseLeave	Конфигурация...
OnMouseMove	Конфигурация...
OnMouseUp	Конфигурация...
- Нажать	
Переменная	PLC_PRG.bSetLanguage
Переключить на FALSE	<input type="checkbox"/>

Рис. 11.3.5.17. Параметры кнопки **Переключить язык**

8. Настроим [Менеджер визуализации](#). В его установках поставим галочку **Использовать строки Unicode**:



Настройки **target**- и **web**-визуализации будут выглядеть следующим образом:

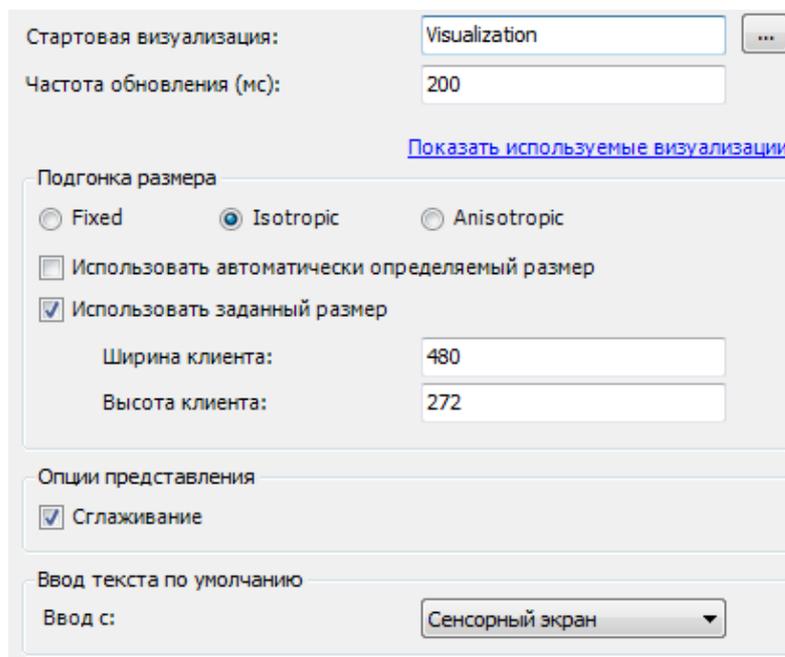


Рис. 11.3.5.18. Настройки **target**-визуализации

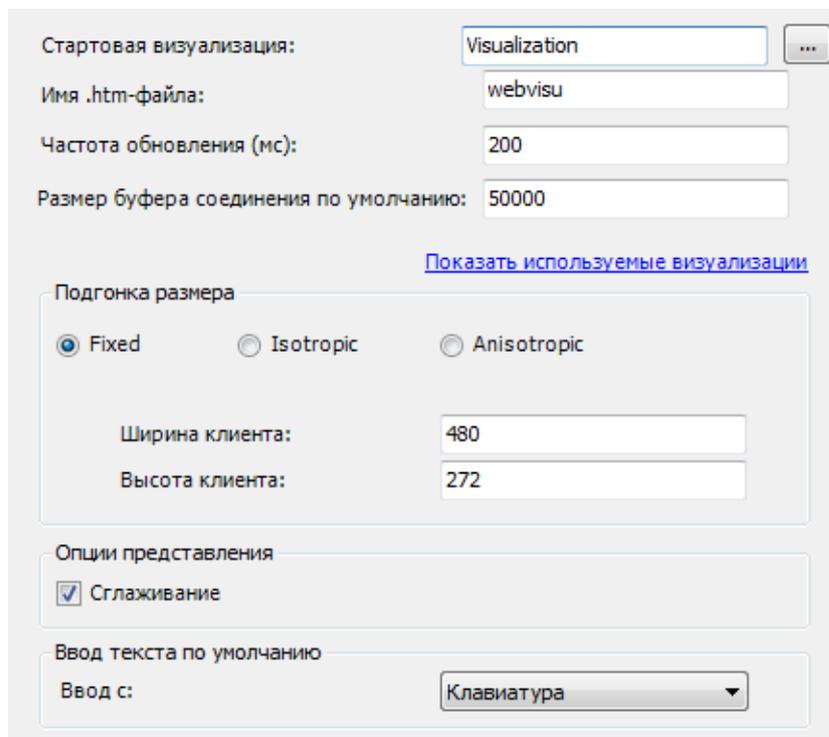


Рис. 11.3.5.19. Настройки **web**-визуализации

9. [Запустим проект на виртуальном контроллере](#). Откроем в веб-браузере страницу <http://localhost:8080/webvisu.htm>, чтобы увидеть web-визуализацию проекта. **Обратите внимание**, что браузер должен поддерживать **HTML5**. Проверим функционал проекта.

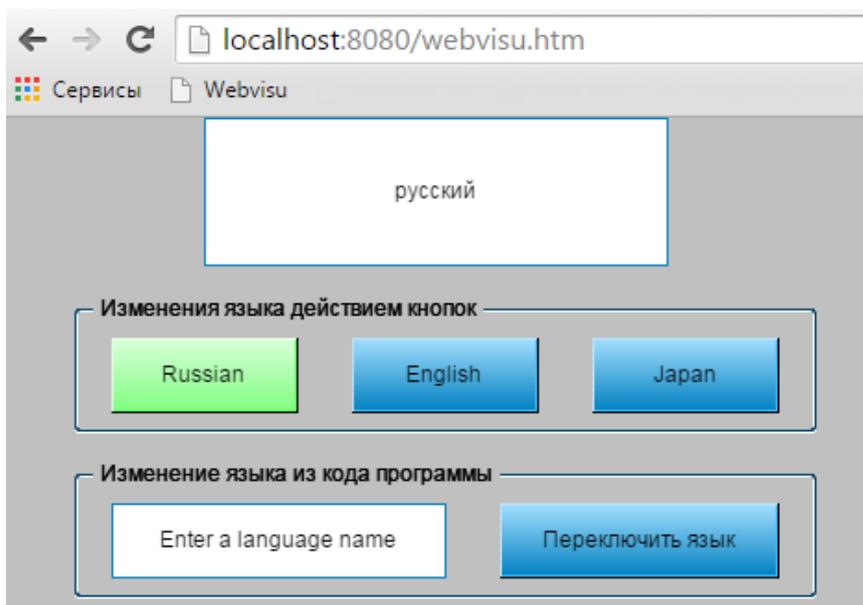


Рис. 11.3.5.20. Web-визуализация проекта в браузере **Chrome**

По умолчанию в проекте выбран русский язык. Нажмите кнопку **English**, чтобы переключить язык текста на английский. Нажмите кнопку **Japan**, чтобы переключить язык текста на японский.

Нажмите на поле ввода имени язык и введите **ENGLISH**. Нажмите кнопку **Переключить язык**.

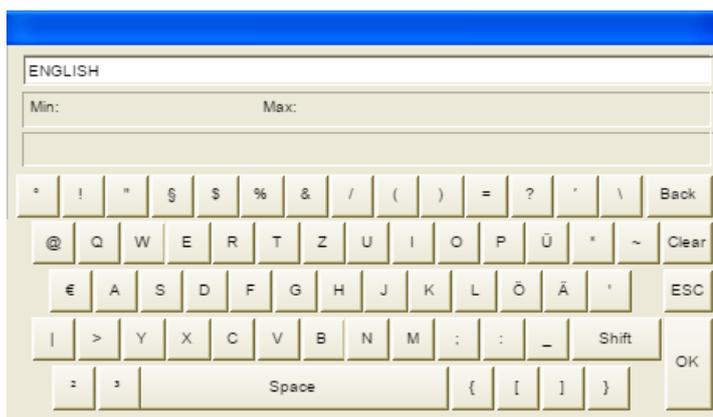


Рис. 11.3.5.21. Ввод имени языка с помощью экранной клавиатуры

Это приведет к переключению языка текста на английский. **Обратите внимание**, что это реализовано из кода программы с помощью системной переменной **VisuElems.CurrentLanguage**. Таким образом пользователь может переключать язык проекта в зависимости от значений переменных, сохранять язык после перезагрузки (чтение названия языка из энергонезависимой **RETAIN** переменной) и т.д.

Обратите внимание, что создаваемый в визуализации текст автоматически попадает в список текстов **GlobalTextList**, расположенный на панели устройств во вкладке **POU**:

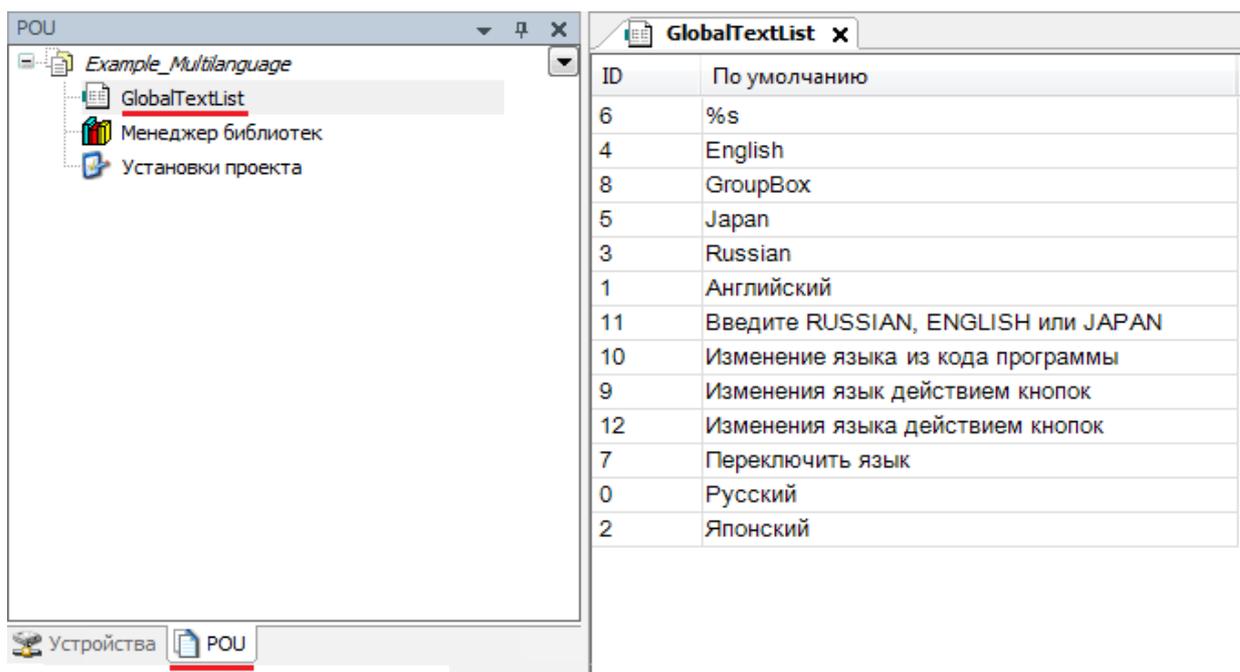


Рис. 11.3.5.22. Глобальный список текстов проекта

11.3.6. Тиражирование элементов (Multiply Visu Elements)

Данный пример посвящен тиражированию элементов.

При разработке экранов визуализации довольно часто существует необходимость создавать группы однотипных элементов (ламп, переключателей, полей ввода-вывода и т.д.), настройки которых отличаются только привязанными переменными. В этом случае, вместо создания элементов вручную, можно воспользоваться командой тиражирования элементов (**Multiply Visu Elements**). Для этого переменные элементов, создаваемых тиражированием, предварительно должны быть собраны в массив.

Тиражировать можно следующие элементы:

1. все элементы группы [Базовые](#);
2. все элементы группы [Индикаторы/Переключатели/Изображения](#);
3. элементы [Кнопка](#), [Текстовое поле](#), [Кнопка-флажок](#), [Управление вращением](#).

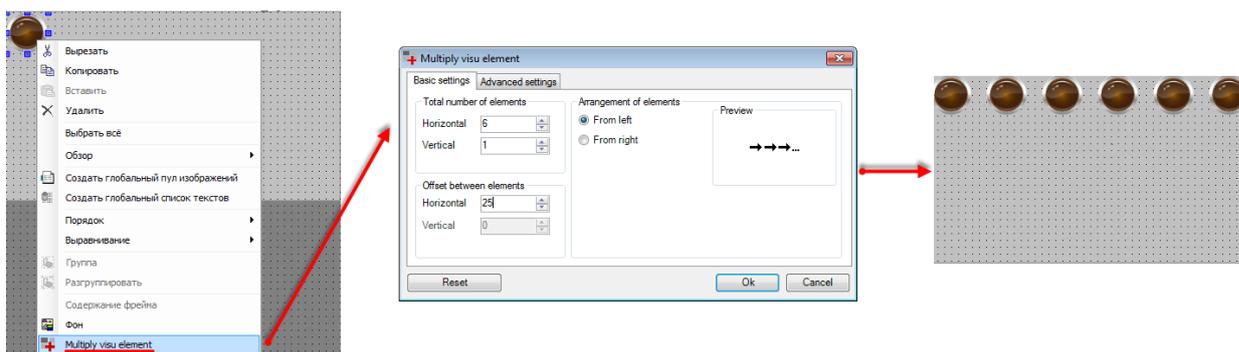


Рис. 11.3.6.1. Использование тиражирования элементов в проекте

1. Создадим новый **стандартный** проект **CODESYS** с названием **Example_MultiplyVisuElements** и настройками по умолчанию: target – **CODESYS Control Win V3**, язык **PLC_PRG – ST**.

2. В программе **PLC_PRG** объявим массив переменных типа **BOOL**:

```
1 | PROGRAM PLC_PRG
2 | VAR
3 |     arrLamps:ARRAY [1..6] OF BOOL; // массив переменных тиражируемых элементов
4 | END_VAR
```

Рис. 11.3.6.2. Объявление переменных программы **PLC_PRG**

3. Создадим в проекте экран визуализации **Visualization**. Добавим на экран элемент [Индикатор](#) и привяжем к нему переменную-заполнитель **arrLamps[\$FIRSTDIM\$]**:

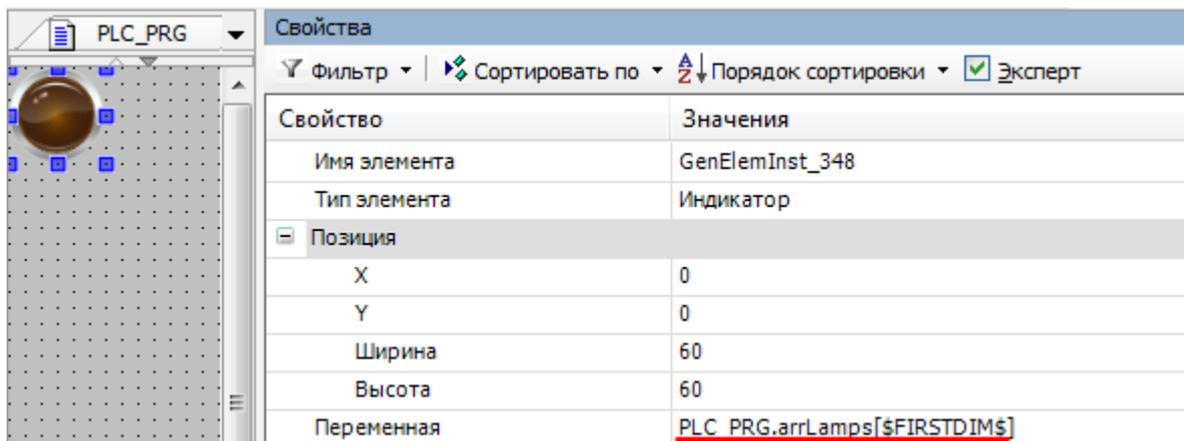


Рис. 11.3.6.3. Параметры элемента **Индикатор**

Теперь нажмем на элемент **ПКМ** и в контекстном меню выберем команду **Multiply Visu Elements**:

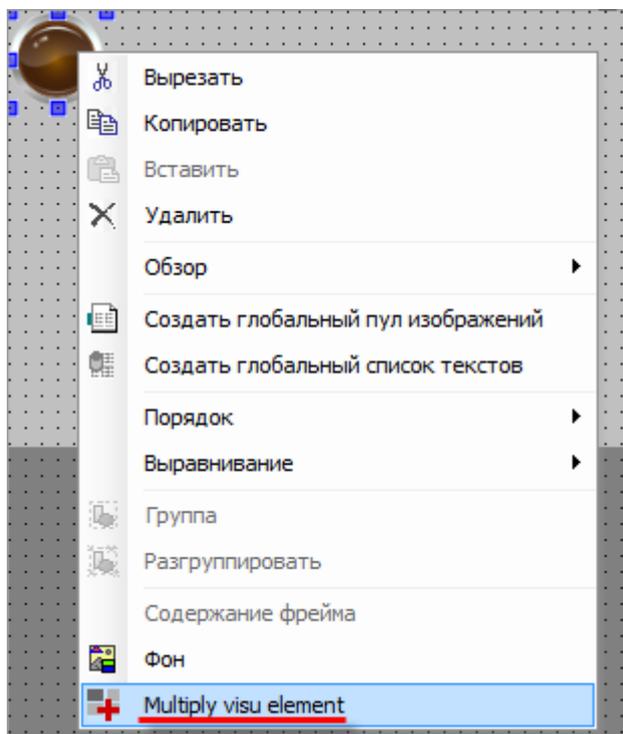


Рис. 11.3.6.4. Команда тиражирования элементов

В открывшемся окне конфигурации тиражирования зададим следующие настройки:

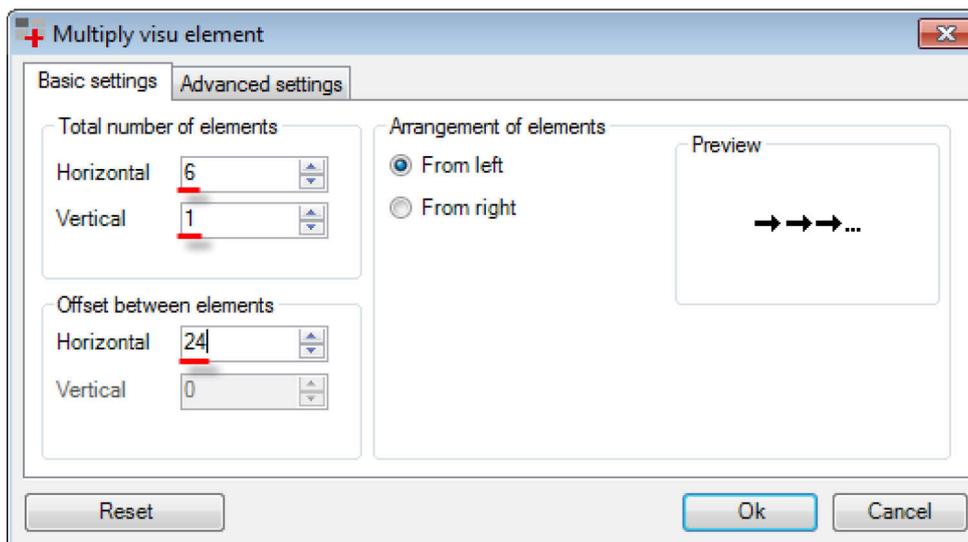


Рис. 11.3.6.5. Окно конфигурации тиражирования

В результате будут созданы шесть индикаторов, расположенные в один ряд, с зазором в 24 пикселя между соседними элементами:

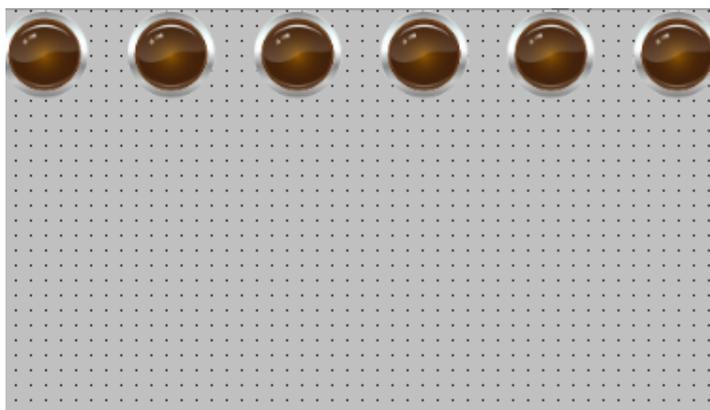


Рис. 11.3.5.6. Результат тиражирования элемент Индикатор

К первому индикатору теперь привязана переменная **arrLamps[1]**, ко второму – **arrLamps[2]** и т.д. При необходимости, номер первого элемента массива и шаг между ними может быть задан на вкладке **Advanced Settings** окна тиражирования элементов.

11.3.7. Управление пользователями. Парольный доступ

Данный пример посвящен использованию настроек вкладки [Управление пользователями](#), которая входит в состав [Менеджера визуализации](#).

Довольно часто в системах автоматизации возникает задача реализации многопользовательского доступа. Например, инженеру-наладчику нужно иметь доступ к экранам визуализации настроек системы, в то время как оператор такой возможности должен быть лишен.

Перед началом работы пользователь должен авторизоваться, введя свой логин и пароль. Обычно это происходит на стартовом экране проекта.

Данные пользователя (например, его логин или группа) могут обрабатываться в программе с целью контроля доступа к экранам (см. [п. 11.3.1](#)), изменения языка (см. [п. 11.3.5](#)) и т.д.

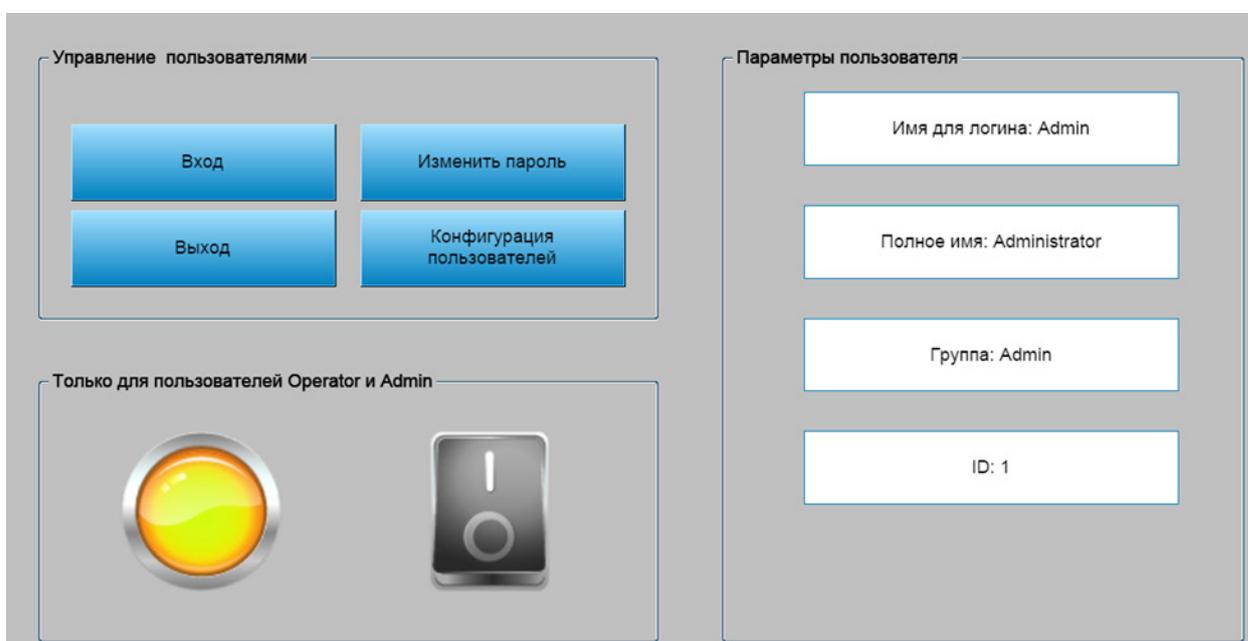


Рис. 11.3.7.1. Внешний вид примера **Управление пользователями**

Пример создан в среде **CODESYS 3.5 SP6 Patch 2** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с target-файлом **3.5.6.0**.

Пример доступен для скачивания: [Example_UserManagment.projectarchive](#)

1. Создадим новый **стандартный** проект **CODESYS** с названием **Example_UserManagement** и настройками по умолчанию: target – **CODESYS Control Win V3**, язык **PLC_PRG** – **ST**.

2. В программе **PLC_PRG** объявим переменную **bLamp** типа **BOOL**:

```
1 | PROGRAM PLC_PRG
2 | VAR
3 |     bLamp:BOOL;           // переменная индикатора
4 | END_VAR
```

Рис. 11.3.7.2. Объявление переменных программы **PLC_PRG**

3. Добавим в проект экран визуализации **Visualization**. В его свойствах выберем размер **800x480**.

4. В Менеджере визуализации откроем вкладку Управление пользователями и нажмем кнопку **Создать управление пользователями с группами и пользователями по умолчанию**:

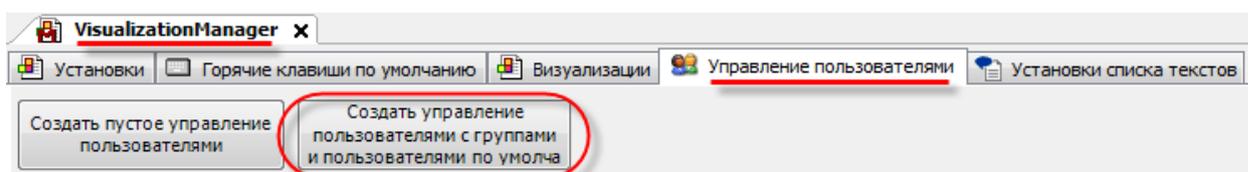


Рис. 11.3.7.3. Создание управления пользователями

Это приведет к автоматическому созданию четырех групп доступа (**Admin**, **Service**, **Operator**, **None**), первые три из которых содержат по одному пользователю – с именами **Admin**, **Service**, **Operator** соответственно. Группа **None** используется для идентификации пользователей до момента авторизации в системе.

Настроим группы пользователей:

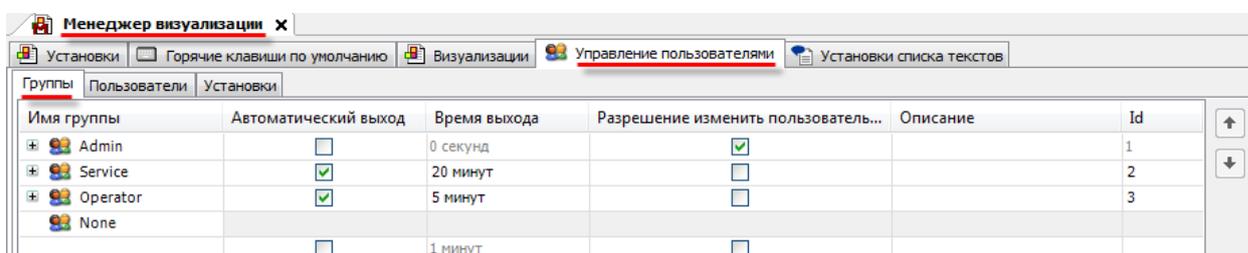


Рис. 11.3.7.4. Настройки групп пользователей

Для групп **Service** и **Operator** установим галочку **Автоматический выход** и укажем время бездействия, через которое будет произведен принудительный выход пользователя из системы. Для группы **Admin** поставим галочку **Разрешение изменить пользовательские данные** – это позволит членам этой группы открыть соответствующее диалоговое окно в процессе работы проекта.

Настроим пользователей:

Имя для логина	Полное имя	Пароль	Группа пользоват...	Отключить	Описание
Admin	Administrator	*****	Admin	<input type="checkbox"/>	
Service	Service	*****	Service	<input type="checkbox"/>	
Operator	Operator	*****	Operator	<input type="checkbox"/>	

Рис. 11.3.7.5. Настройки пользователей

Зададим пользователю **Admin** пароль **1**, пользователю **Service** – пароль **2**, пользователю **Operator** – пароль **3**. **Обратите внимание**, что даже после изменения числа символов, пароль в столбце будет отображаться в виде пяти «звездочек» (*****).

На вкладке **Установки** присутствует настройка **Поведение при выходе**, которая позволяет при выходе пользователя из системы переключать визуализацию на стартовый экран. В данном примере эта возможность **использована не будет** (поскольку проект состоит из одного экрана визуализации).

5. Экран **Visualization** будет содержать четыре элемента [Кнопка](#), пять элементов [Текстовое поле](#), элемент [Индикатор](#), элемент [Клавишный переключатель](#) и три элемента [Группа](#):

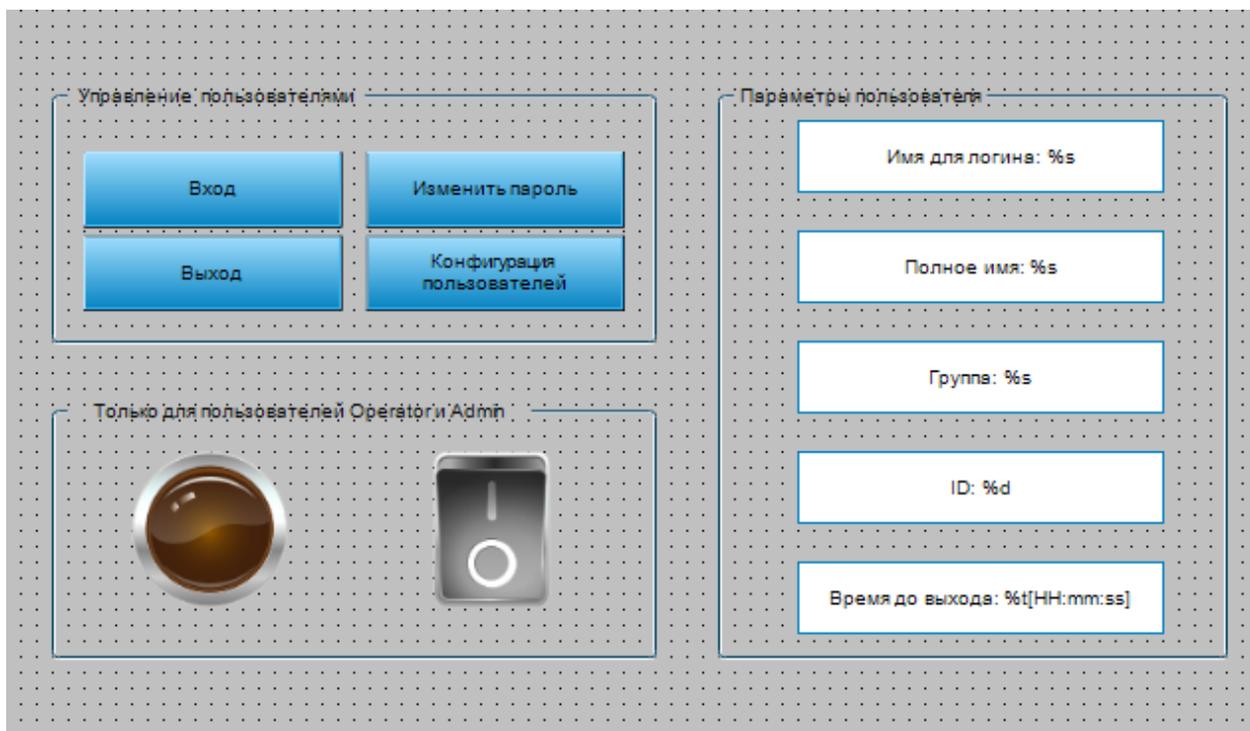


Рис. 11.3.7.6. Содержание экрана **Visualization**

Настроим кнопки вкладки **Управление пользователями**.

У кнопки **Вход** во вкладке [InputConfiguration](#) к параметру **OnMouseClicked** привяжем действие [Управление пользователями](#) с диалогом **Войти**. *Обратите внимание*, что это действие становится доступным только после настройки [одноименной вкладки](#) в Менеджере визуализации.

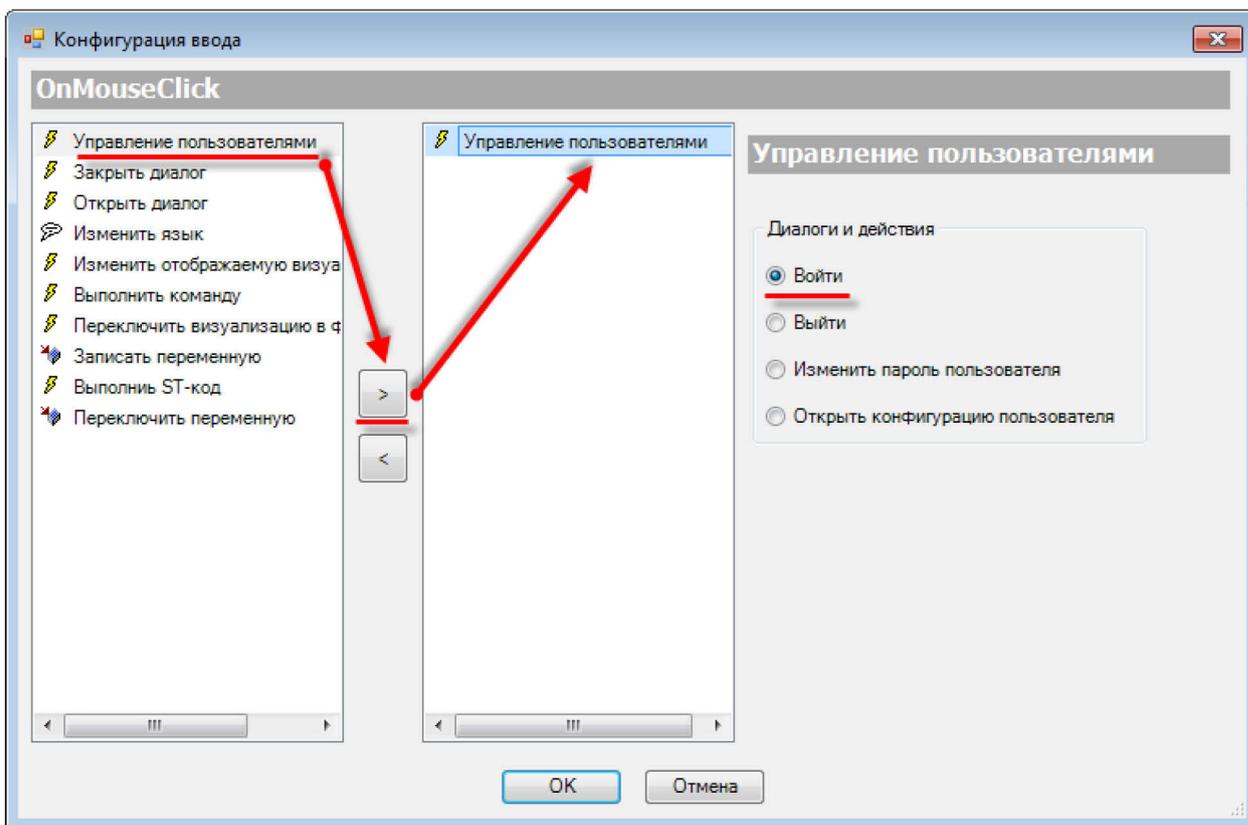


Рис. 11.3.7.7. Настройки действия **Управление пользователями**

К кнопке **Выход** привяжем действие **Управление пользователями** с диалогом **Выйти**.

К кнопке **Изменить пароль** привяжем действие **Управление пользователями** с диалогом **Изменить пароль пользователя**.

К кнопке **Конфигурация пользователей** привяжем действие **Управление пользователями** с диалогом **Открыть конфигурацию пользователей**.

Настроим элементы вкладки **Только для пользователей Operator и Admin**.

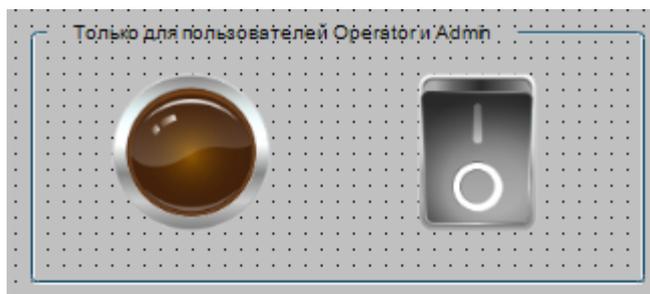


Рис. 11.3.7.8. Элементы вкладки **Только для пользователей Operator и Admin**

Вкладка будет содержать элемент [Индикатор](#) и [Клавишный переключатель](#), причем оба элемента будут видны только пользователям групп **Operator** и **Admin**, а **Клавишный переключатель** будет реагировать на нажатие только пользователей группы **Admin**.

Для этого в настройках элемента **Индикатор** дважды нажмем **ЛКМ** на параметр **Права доступа**. **Обратите внимание**, что этот параметр становится доступным только после настройки вкладки [Управление пользователями](#) в **Менеджере визуализации**.

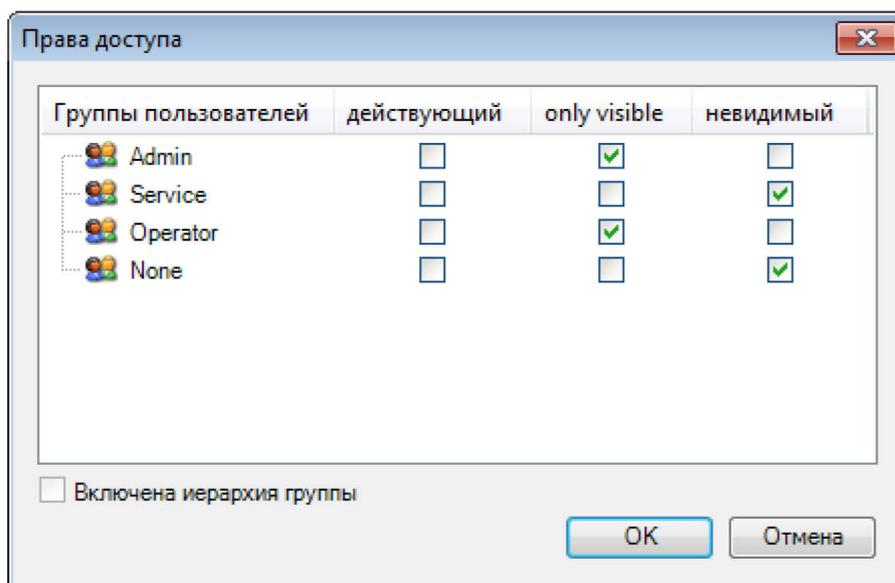


Рис. 11.3.7.9. Настройка прав доступа для элемента **Индикатор**

Для пользователей **None** и **Service** поставим галочку **Невидимый**, для пользователей **Operator** и **Admin** – галочку **Only Visible**.

Настройки прав доступа для элемента [Клавишный переключатель](#), соответственно, будут выглядеть следующим образом:

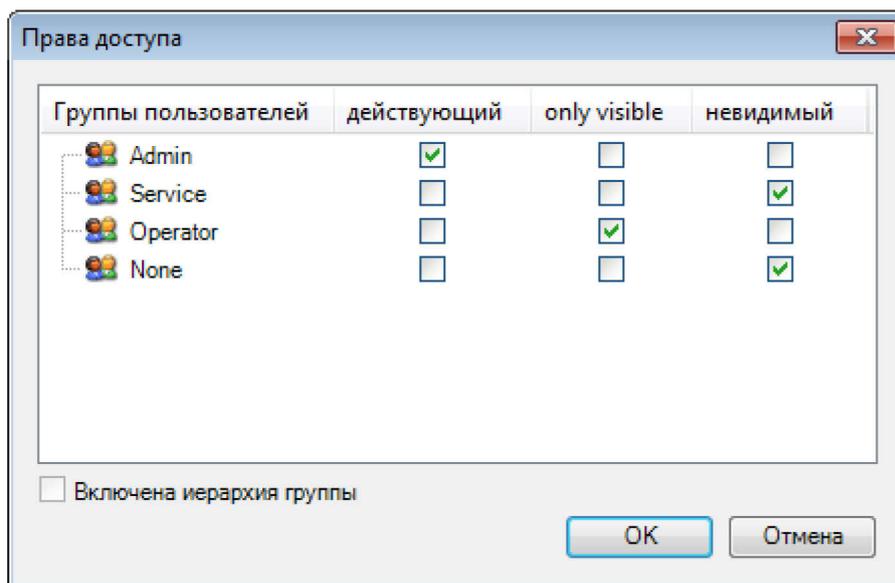


Рис. 11.3.7.10. Настройка прав доступа для элемента **Клавишный переключатель**

Настроим элементы вкладки **Параметры пользователя**. Вкладка содержит шесть элементов [Текстовое поле](#), которые будут использоваться для отображения информации о текущем пользователе с помощью системных переменных.

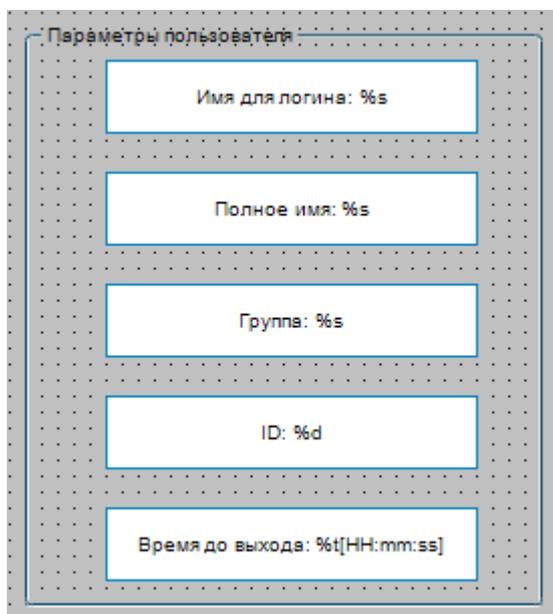


Рис. 11.3.7.11. Элементы вкладки **Параметры пользователя**

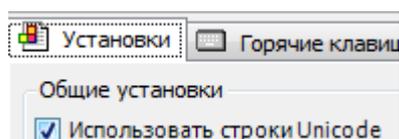
Настройки вкладки приведены в табл. 45.

Табл. 45. Переменные элементов вкладки **Параметры пользователя**

№ пп.	Текст	Текстовая переменная	Тип переменной
1	Имя для логина: %s	VisuElems.CurrentUserName	WSTRING
2	Полное имя: %s	VisuElems.CurrentFullName	WSTRING
3	Группа: %s	VisuElems.CurrentUserGroupName	WSTRING
4	ID: %d	VisuElems.CurrentUserGroupId	DWORD
5	Время до выхода: %t[HH:mm:ss]	VisuElems.CurrentRemainingAutoLogoutTime	TIME
№ пп.	Текст	Переменная невидимости	Тип переменной
6	Время до выхода: %t[HH:mm:ss]	NOT(VisuElems.CurrentUseAutoLogoutTime)	BOOL

Эти переменные могут использоваться в коде пользовательской программы (например, для переключения экрана визуализации после входа в систему определенного пользователя).

6. Настроим [Менеджер визуализации](#). В его установках поставим галочку **Использовать строки Unicode**:



Настройки **target**- и **web**-визуализации будут выглядеть следующим образом:

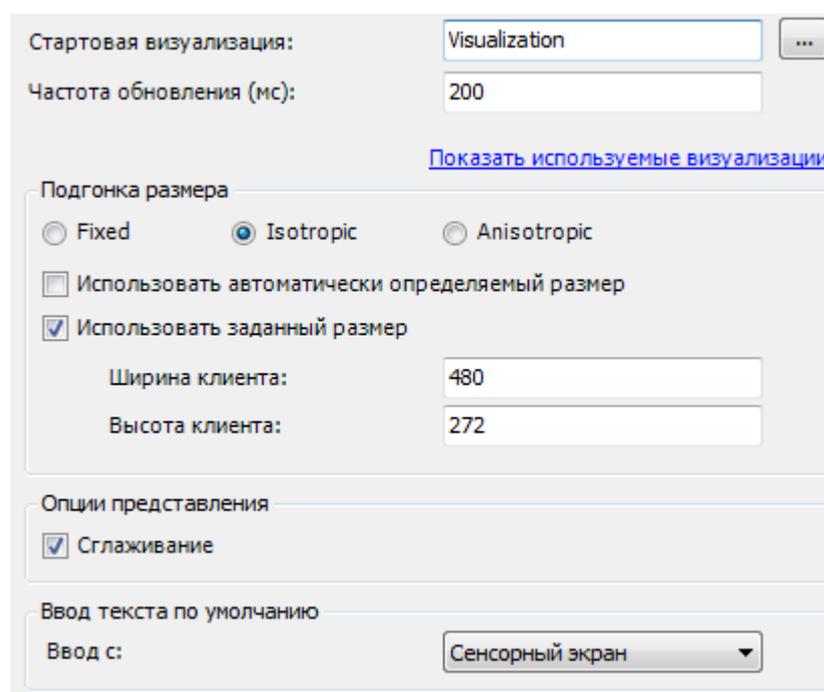


Рис. 11.3.7.12. Настройки **target**-визуализации

Стартовая визуализация: Visualization

Имя .htm-файла: webvisu

Частота обновления (мс): 200

Размер буфера соединения по умолчанию: 50000

[Показать используемые визуализации](#)

Подгонка размера

Fixed Isotropic Anisotropic

Ширина клиента: 480

Высота клиента: 272

Опции представления

Сглаживание

Ввод текста по умолчанию

Ввод с: Клавиатура

Рис. 11.3.7.13. Настройки **web**-визуализации

7. [Запустим проект на виртуальном контроллере](#). Откроем в веб-браузере страницу <http://localhost:8080/webvisu.htm>, чтобы увидеть web-визуализацию проекта. **Обратите внимание**, что браузер должен поддерживать **HTML5**. Проверим функционал проекта.

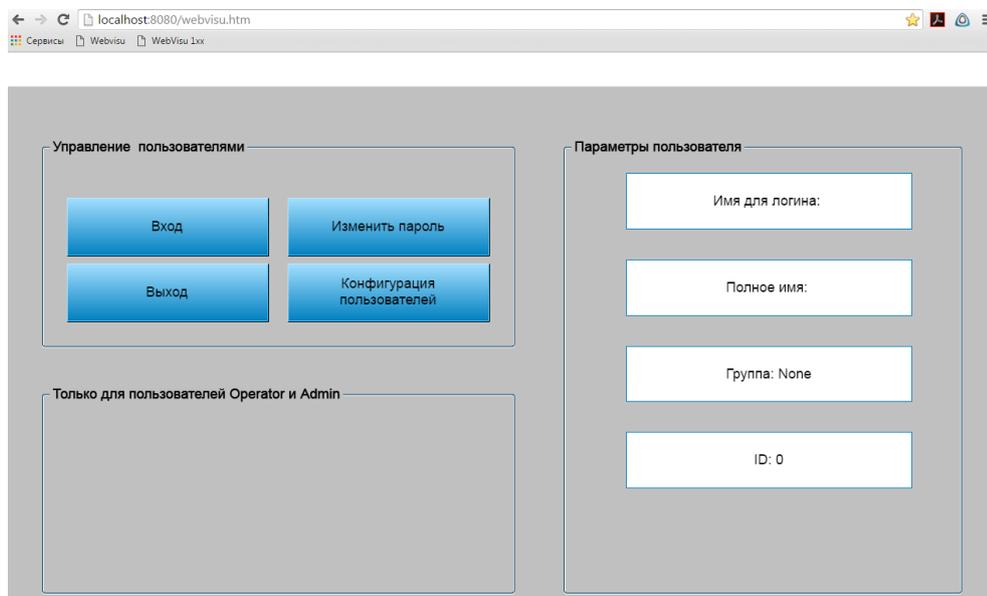


Рис. 11.3.7.14. Web-визуализация проекта в браузере **Chrome**

После загрузки проекта, в системе еще нет авторизованного пользователя – это интерпретируется как наличие безымянного пользователя группы **None**. Поле **Время до выхода** отсутствует (благодаря привязке к параметру **Переменная невидимости инвертированного** значения системной переменной **VisuElems.CurrentUseAutoLogoutTime** . Вкладка **Только для пользователей Operator и Admin** в данный момент является пустой.

Нажмите кнопку **Вход** и введите логин **Operator** (регистр имеет значение), пароль – **3**.

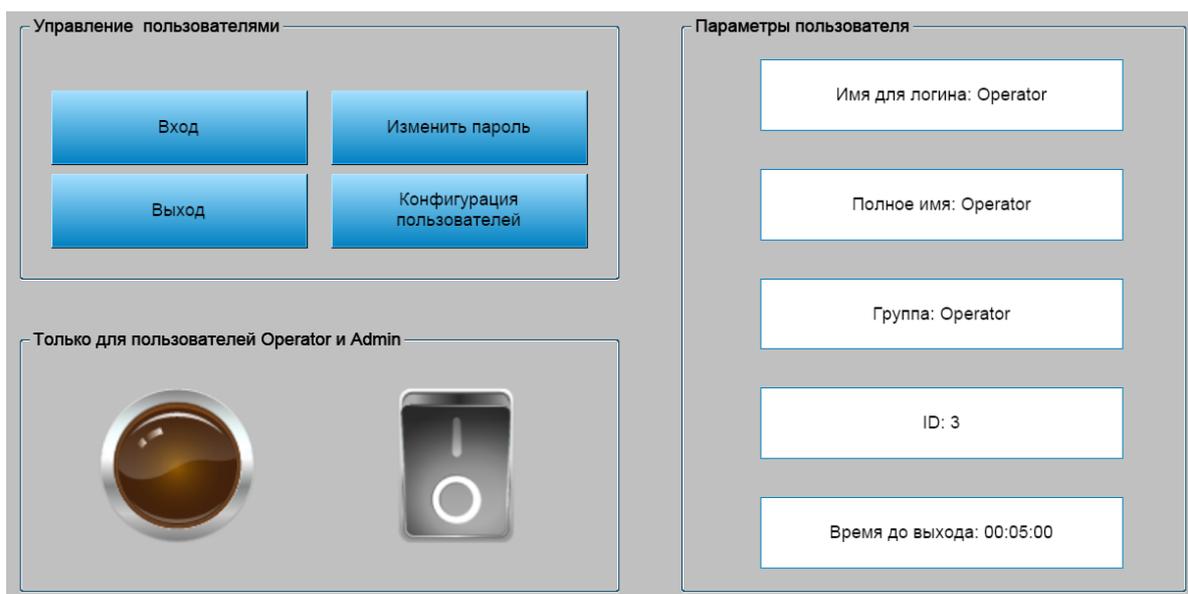


Рис. 11.3.7.15. Авторизация пользователя **Operator**

Во вкладке **Только для пользователей Operator и Admin** отобразятся индикатор и переключатель, а во вкладке **Параметры пользователя** – информация о пользователе **Operator**. Спустя пять минут бездействия произойдет принудительный выход пользователя из системы.

Обратите внимание, что переключатель и кнопка **Конфигурация пользователей** являются неактивными для пользователя **Operator**.

Нажмите кнопку **Изменить пароль**:

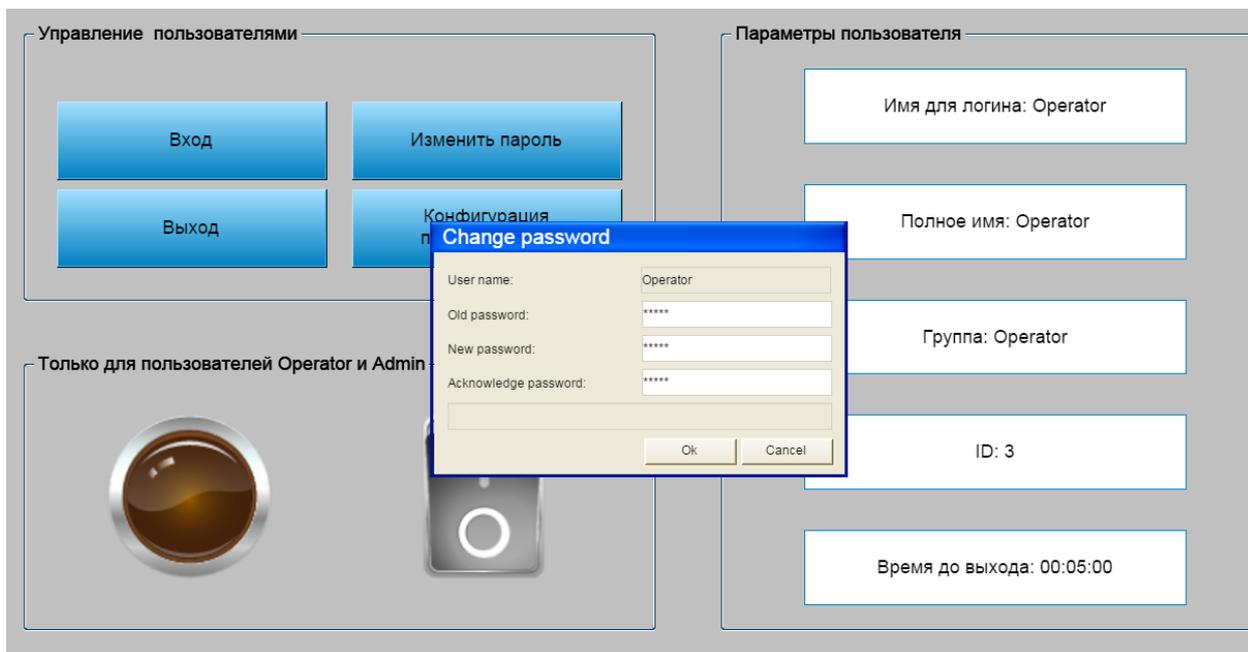


Рис. 11.3.7.16. Диалоговое окно изменения пароля

Для изменения пароля, необходимо один раз ввести старый пароль (**Old password**) и два раза – новый (**New password, Acknowledge password**).

Нажмите кнопку **Вход** и введите логин **Admin** (регистр имеет значение), пароль – **1**.

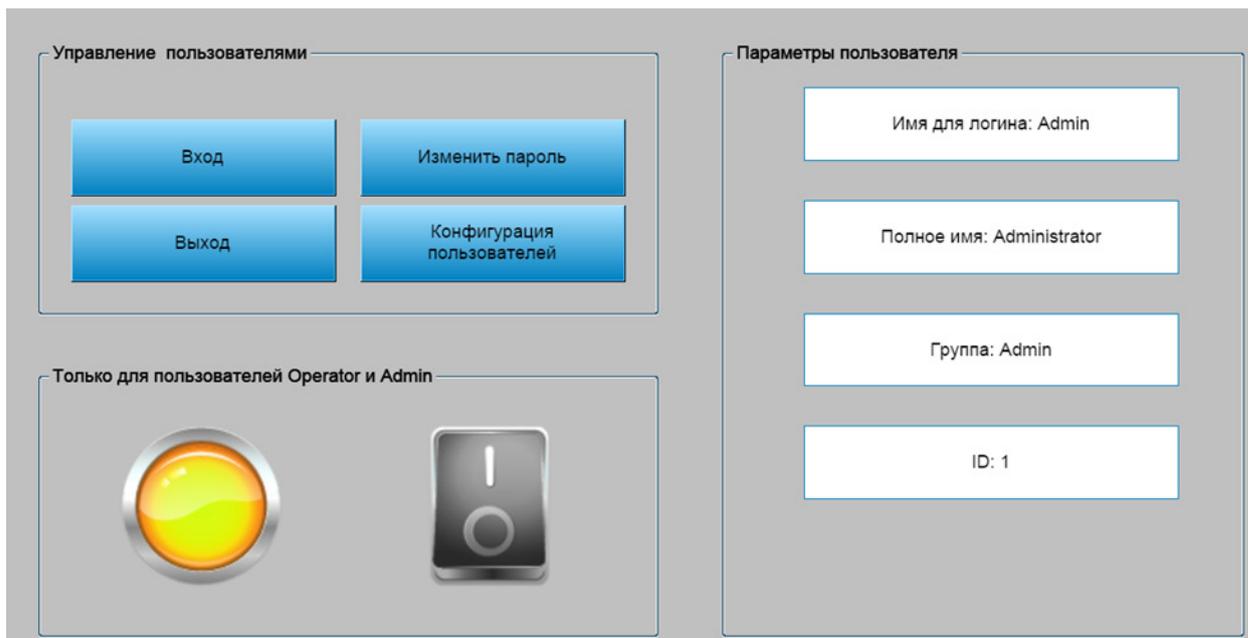


Рис. 11.3.7.17. Авторизация пользователя **Admin**

Для пользователя **Admin** доступно управление индикатором с помощью нажатия переключателя. Поле **Время до выхода** показывает некорректное значение, поскольку для данной группы пользователей отключен автоматический выход из системы.

Нажмите кнопку **Конфигурация пользователей**:

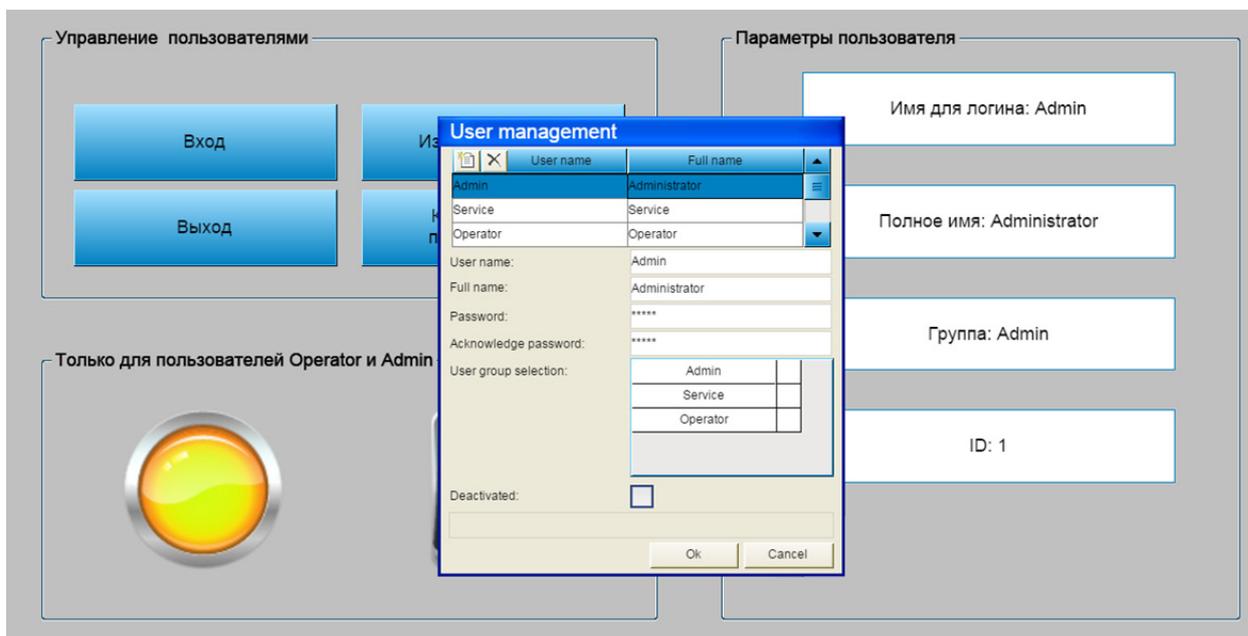


Рис. 11.3.7.18. Диалоговое окно изменения пароля

С помощью этого диалога можно менять логины и пароли пользователей, удалять пользователей, создавать новых и блокировать существующих.

11.3.8. Менеджер рецептов

Данный пример посвящен работе с компонентом **Менеджер просмотра (Менеджер рецептов)**.

Компонент **Менеджер просмотра (Recipe Manager; далее – Менеджер рецептов)** позволяет создавать списки переменных (**группы рецептов**) и составлять для них наборы значений (**рецепты**), которые будут присваиваться переменным соответствующей группы по команде пользователя.

При использовании в своей программе **Менеджера рецептов** пользователю нет необходимости вникать в технические особенности работы с файлами (чтение/запись) – достаточно лишь создать экраны графического интерфейса оператора и привязать к ним переменные рецептов. Фактически рецепты представляют собой текстовые файлы, хранящиеся в памяти контроллера, которые подгружаются в программу по команде пользователя.

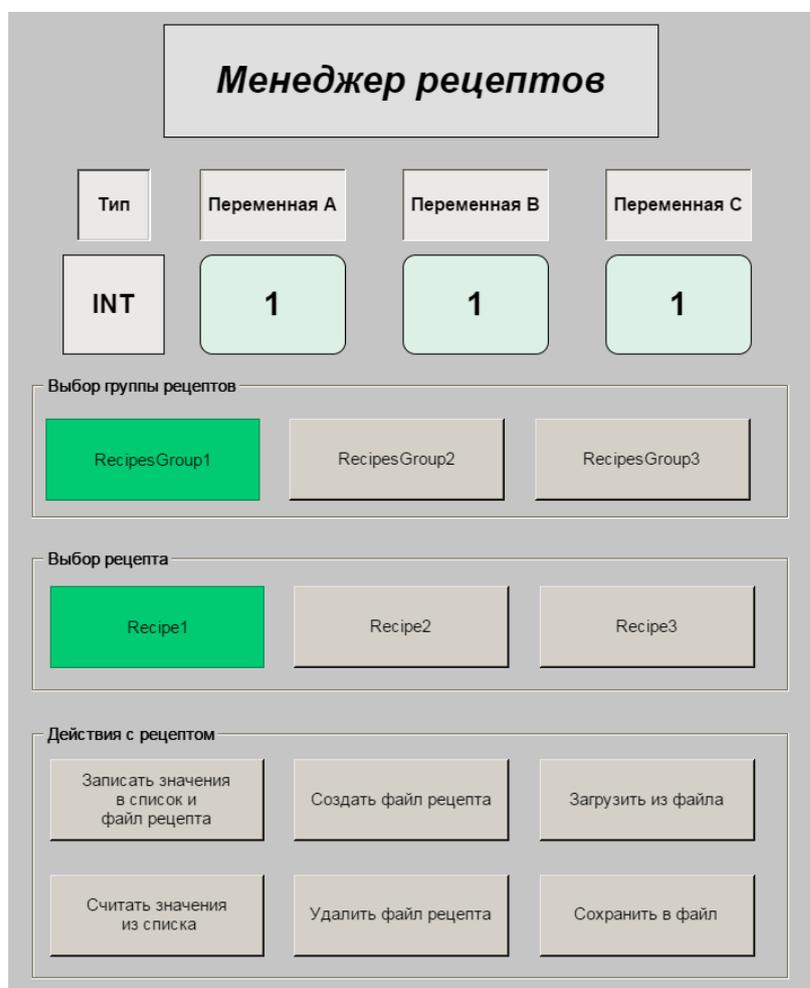


Рис. 11.3.8.1. Внешний вид примера **Менеджер рецептов**

Пример создан в среде **CODESYS 3.5 SP6 Patch 2** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с target-файлом **3.5.6.0**.

Пример доступен для скачивания: [Example RecipeManager.projectarchive](#)

Рассмотрим **пример** создания проекта с использованием **Менеджера рецептов**. В этом проекте пользователь будет оперировать тремя группами рецептов (**RecipeGroup1**, **RecipeGroup2**, **RecipeGroup3**), каждая из которых состоит из трех рецептов (**Recipe1**, **Recipe2**, **Recipe3**), определяемых наборами из трех переменных:

Табл. 46. Структура примера проекта с использованием **Менеджера рецептов**

Группа рецептов	Тип переменных	Переменные группы	Рецепты		
			Recipe1	Recipe2	Recipe3
RecipeGroup1	INT	iA	1	2	3
		iB	1	2	3
		iC	1	2	3
RecipeGroup2	REAL	rA	1.111	2.222	3.333
		rB	1.111	2.222	3.333
		rC	1.111	2.222	3.333
RecipeGroup3	BOOL	bA	TRUE	TRUE	TRUE
		bB	FALSE	TRUE	TRUE
		bC	FALSE	FALSE	TRUE

Группа рецептов – это набор рецептов, которые определяются одними и теми же переменными. Например, группа **RecipeGroup1** содержит три переменные (**iA**, **iB**, **iC**) типа **INT** и состоит из трех рецептов (**Recipe1**, **Recipe2**, **Recipe3**).

Рецепт – это набор конкретных значений переменных, входящих в группу рецептов. Например, рецепт **Recipe1** группы **RecipeGroup1** содержит значения (**1,1,1**).

Список рецепта – это буфер памяти, в котором хранятся значения переменных рецепта. Буфер является промежуточным звеном между проектом **CODESYS** и файлом рецепта.

Файл рецепта – это текстовый файл, который содержит значения переменных рецепта и хранится в файловой системе целевого устройства.

Рассматриваемый нами пример будет содержать **три группы рецептов**, каждая из которых будет состоять из **трех переменных** и содержать **три рецепта**. Каждая группа будет содержать переменные *только одного* типа (**INT**, **REAL**, **BOOL**). Имена рецептов для каждой из групп будут **совпадать**.

1. Создадим новый **стандартный** проект **CODESYS** с названием **ExampleRecipeManager** и настройками по умолчанию: target – **CODESYS Control Win V3**, язык PLC_PRG – **ST**.

2. В программе **PLC_PRG** объявим следующие переменные:

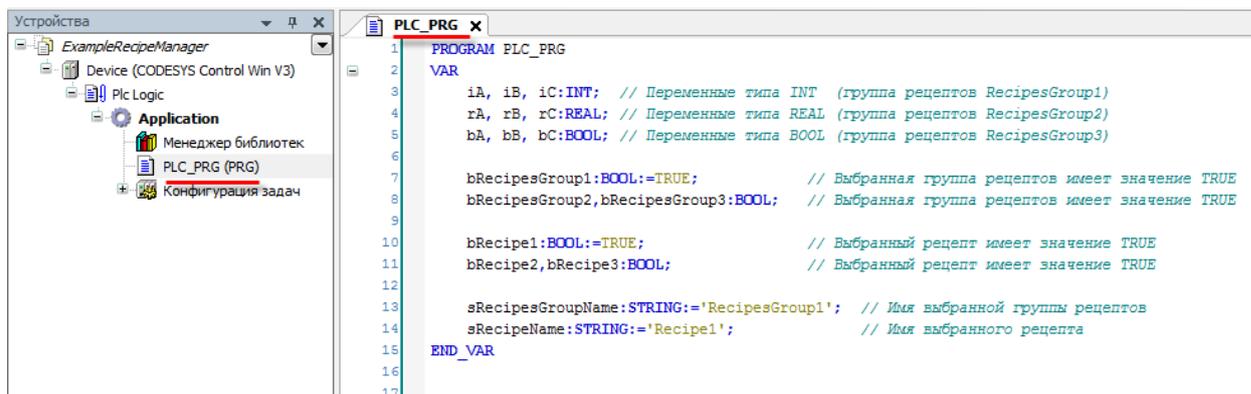


Рис. 11.3.8.2. Объявление переменных в программе **PLC_PRG**

3. Добавим компонент **Менеджер рецептов** (Recipe Manager):

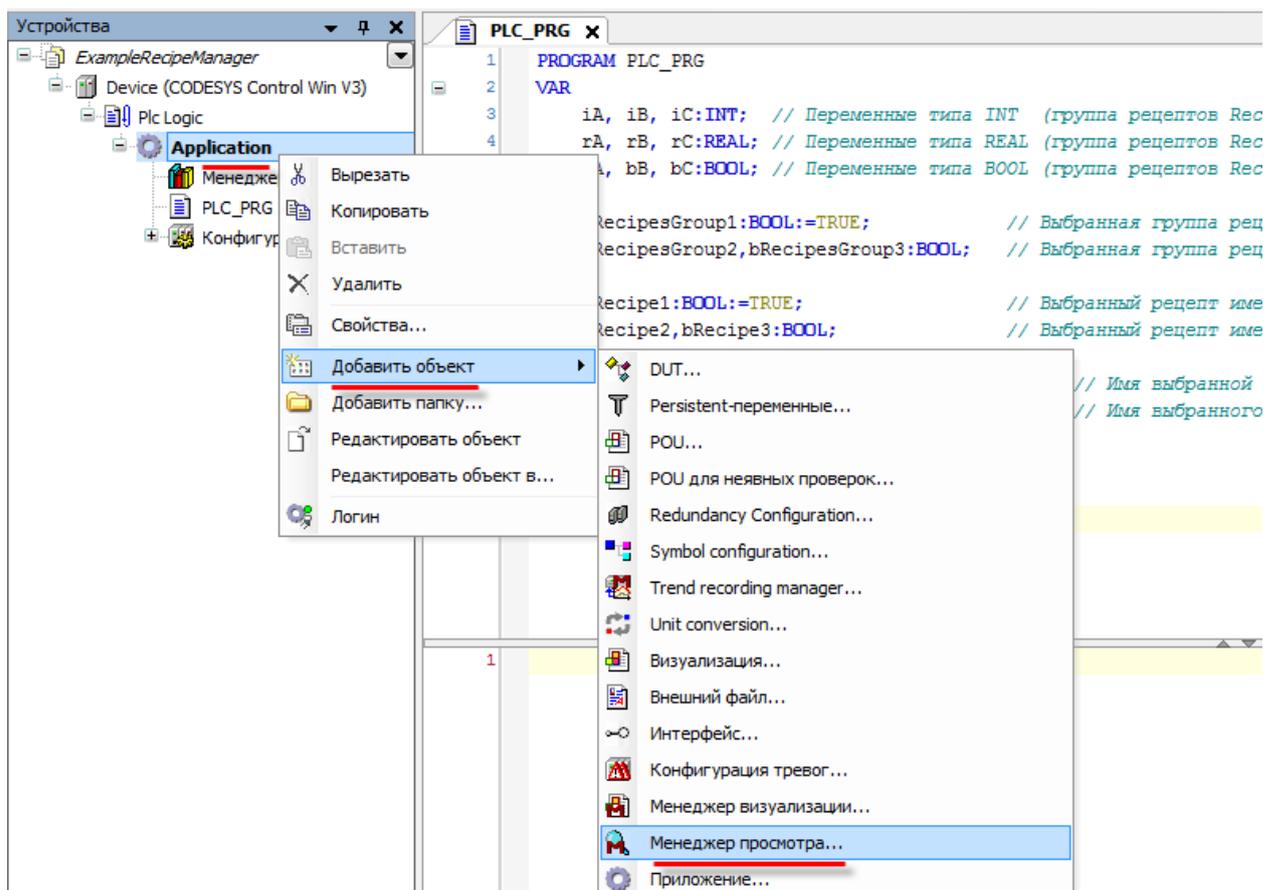


Рис. 11.3.8.3. Добавление в проект **Менеджера рецептов**

На вкладке **Сохранение** выберем тип сохранения **Текстовый** (имеется возможность также сохранять файл в двоичном виде), укажем путь к папке, в которой будут сохраняться рецепты и их расширение. Поскольку мы будем запускать проект на [виртуальном контроллере](#), то в качестве места сохранения рецептов будет использоваться папка на **логическом диске D** компьютера, на котором запускается виртуальный контроллер.

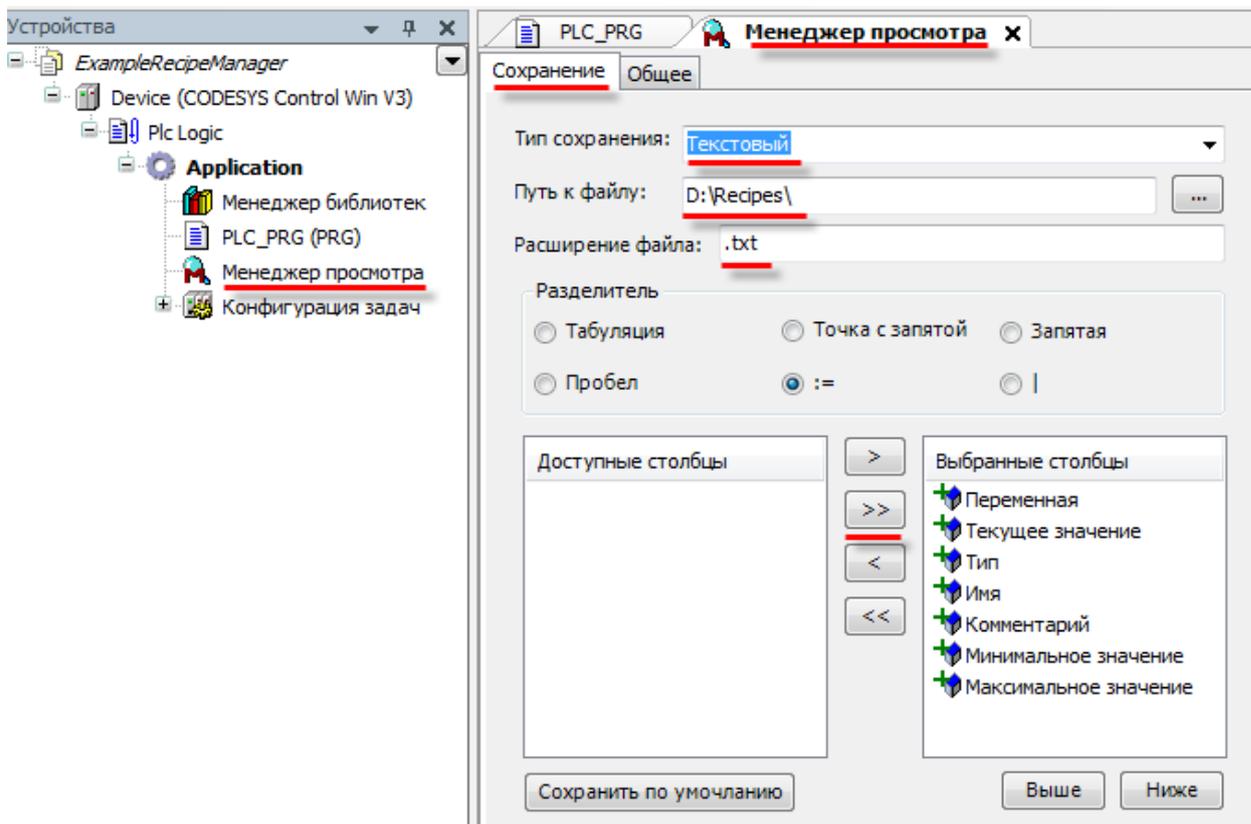


Рис. 11.3.8.4. Настройки **Менеджера рецептов**, вкладка **Сохранение**

На вкладке **Общее** присутствуют два параметра: первый из них отвечает за возможность изменения рецептов в процессе работы проекта, второй – за автоматическое изменение файлов рецептов при изменении их списков. По умолчанию оба параметра активны, и **рекомендуется не изменять** их значения.

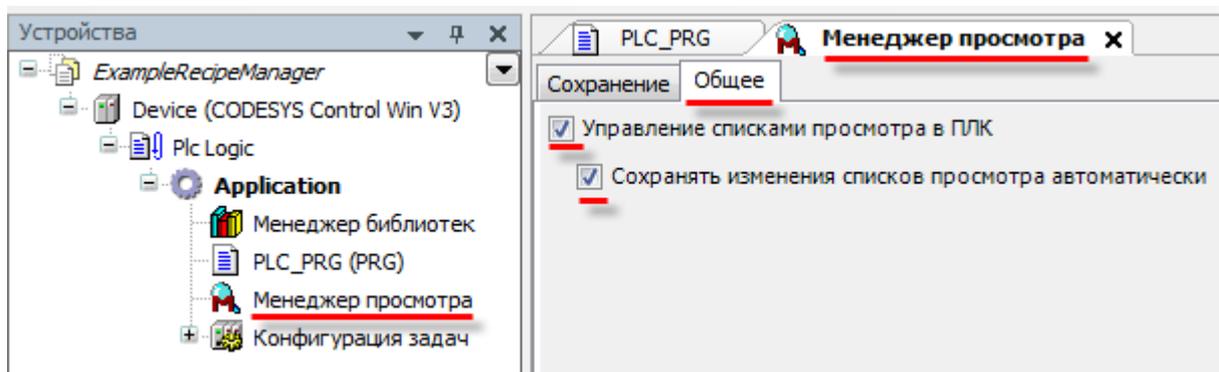


Рис. 11.3.8.5. Настройки **Менеджера рецептов**, вкладка **Общее**

Создадим три группы рецептов (определений списков) с названиями **RecipesGroup1**, **RecipesGroup2**, **RecipesGroup3**.

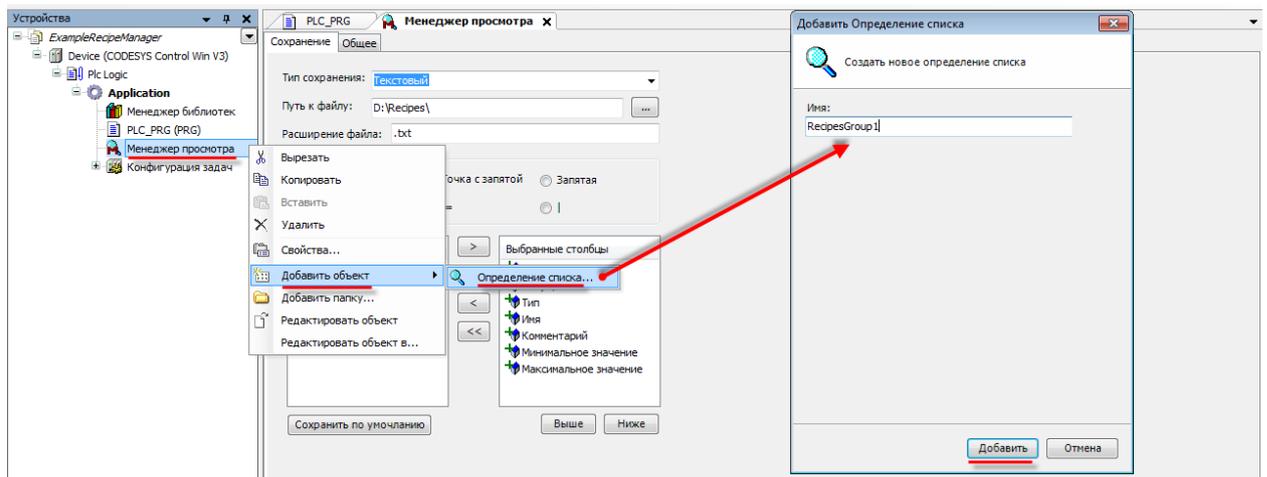


Рис. 11.3.8.6. Добавление группы рецептов

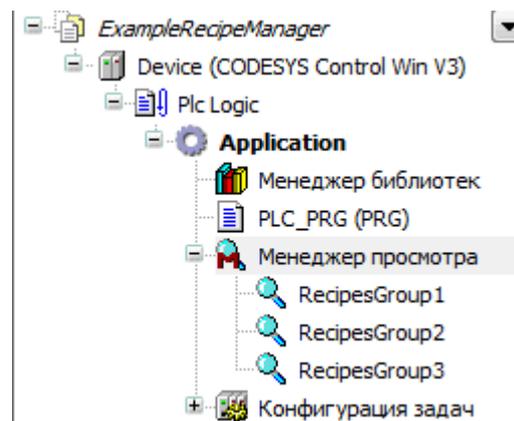


Рис. 11.3.8.7. Три группы рецептов на Панели устройств

В группе **RecipesGroup1** выберем переменные **iA**, **iB**, **iC**, объявленные в программе **PLC_PRG**. В столбце **Имя** зададим переменным имена **Переменная А**, **Переменная В**, **Переменная С**. Эти имена, как и комментарии, являются вспомогательной информацией и через них не производится обращение к переменным программы.

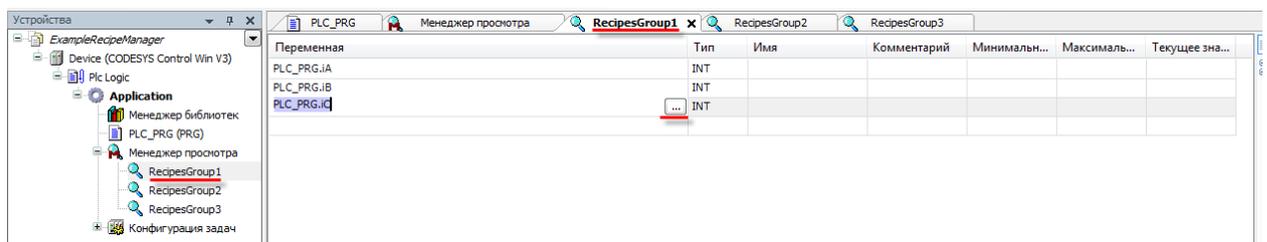


Рис. 11.3.8.8. Окно группы рецептов

Создадим в группе рецептов **RecipesGroup1** три рецепта с названиями **Recipe1**, **Recipe2**, **Recipe3** и наполним их значениями:

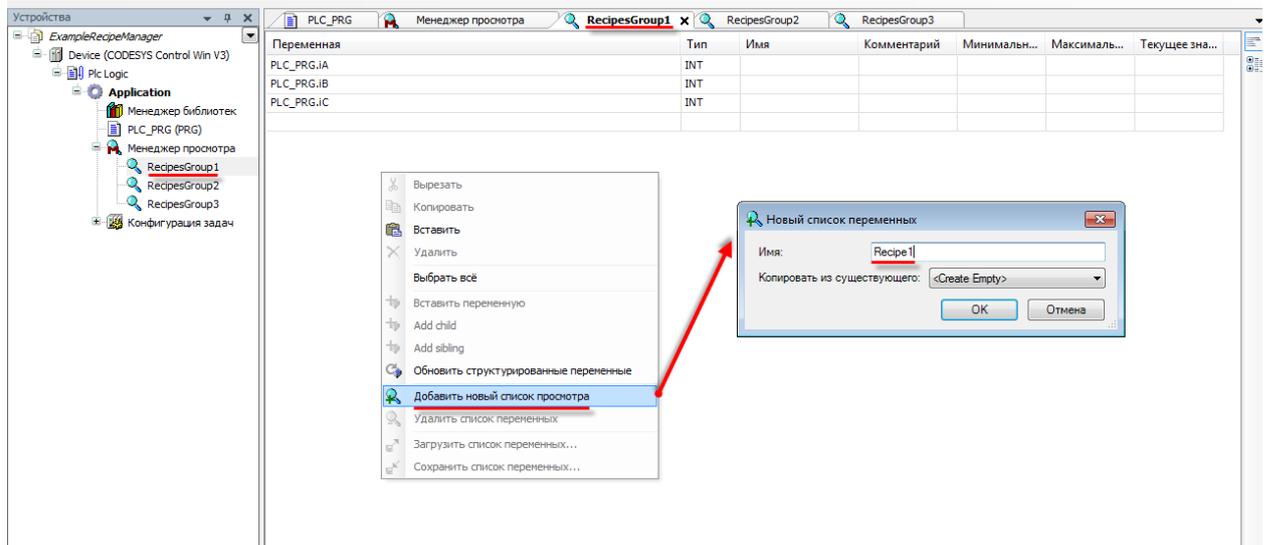


Рис. 11.3.8.9. Создание рецепта в группе рецептов

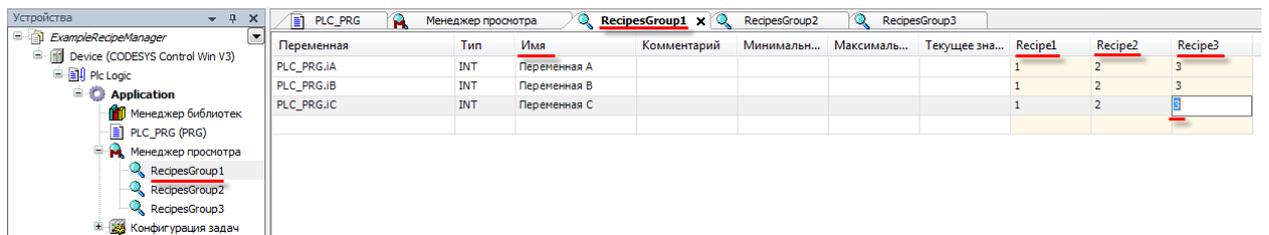


Рис. 11.3.8.10. Группа рецептов **RecipesGroup1**

По аналогии настроим группу **RecipesGroup2** (три переменных типа **REAL**) и группу **RecipesGroup3** (три переменных типа **BOOL**):

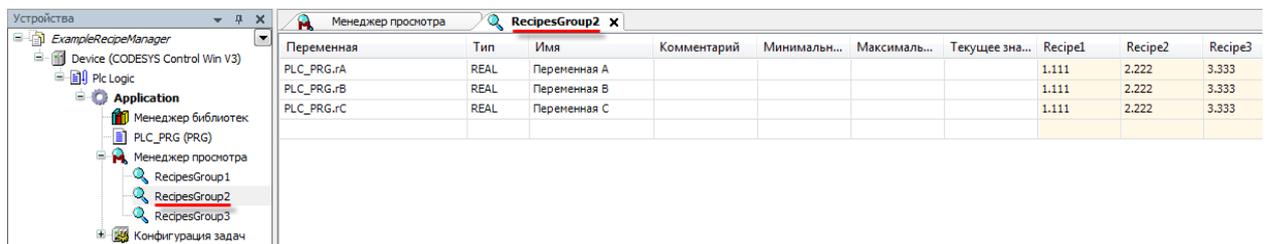


Рис. 11.3.8.11. Группа рецептов **RecipesGroup2**

Переменная	Тип	Имя	Комментарий	Минимальн...	Максимальн...	Текущее зна...	Recipe1	Recipe2	Recipe3
PLC_PRG.bA	BOOL	Переменная А					TRUE	TRUE	TRUE
PLC_PRG.bB	BOOL	Переменная В					FALSE	TRUE	TRUE
PLC_PRG.bC	BOOL	Переменная С					FALSE	FALSE	TRUE

Рис. 11.3.8.12. Группа рецептов **RecipesGroup3**

4. Создание визуализации

4.1. Экран Visualization

[Добавим в проект](#) экран визуализации с названием **Visualization**. В его [свойствах](#) выберем размер **600x700**. Добавим на экран визуализации его название (элемент [Прямоугольник](#)), фрейм, в котором будут отображаться значения переменных выбранного рецепта (элемент [Фрейм](#)), кнопки выбора групп рецептов, кнопки выбора рецептов, кнопки управления рецептами (элементы [Кнопка](#)), которые будут визуально отделены друг от друга с помощью элементов [Группа](#).

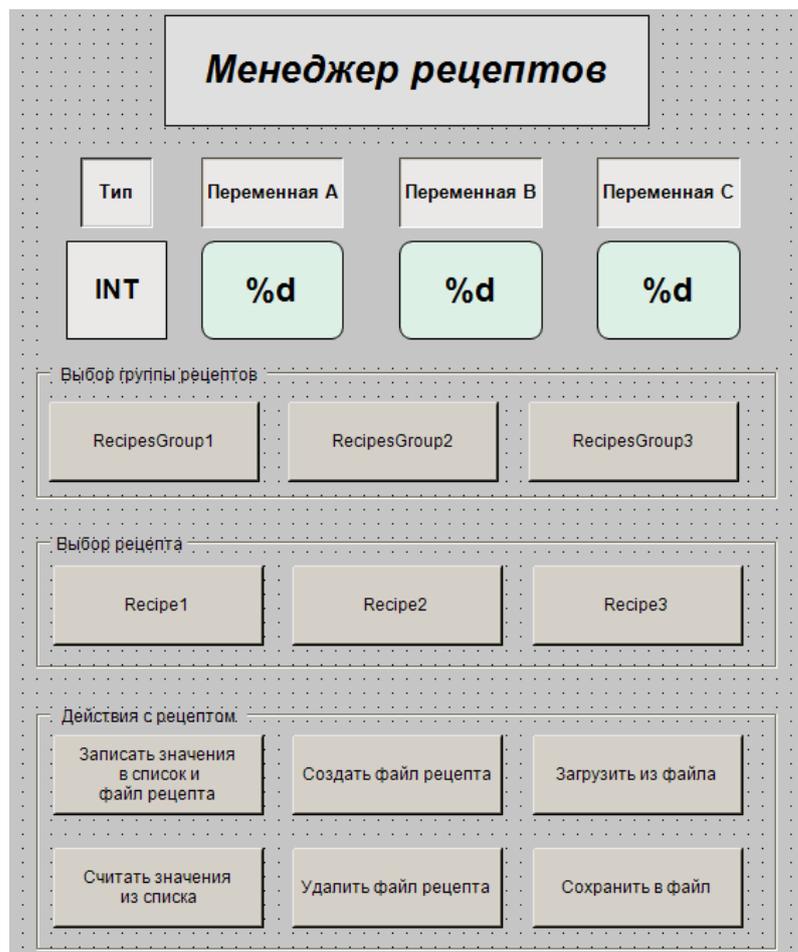


Рис. 11.3.8.13. Содержимое экрана визуализации **Visualization**

4.2. Экран Frame1

Для отображения в элементе [Фрейм](#) экрана **Visualization** значений переменных трех групп рецептов, [создадим еще три экрана визуализации](#) с названиями **Frame1**, **Frame2**, **Frame3**. В их [свойствах](#) выберем размер **520x150**. **Обратите внимание**, что число фреймов совпадает с количеством отображаемых групп рецептов.

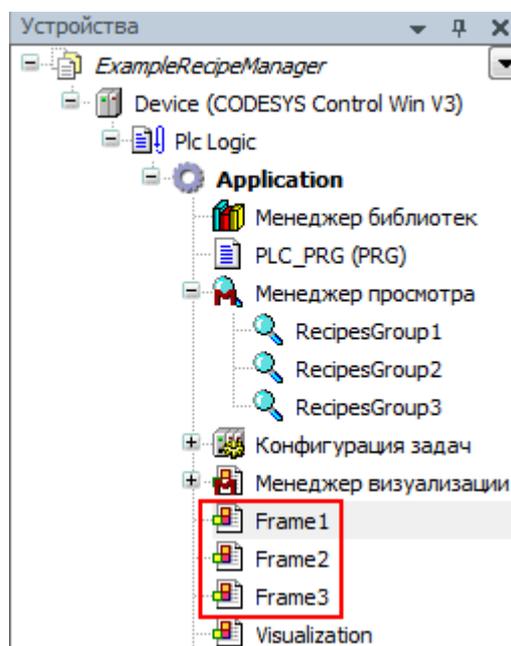


Рис. 11.3.8.14. Экраны фрейма на **Панели устройств**

Добавим на экран **Frame1** элементы типа [Текстовое поле](#) (верхний ряд) и [Прямоугольник/Скругленный прямоугольник](#) (нижний ряд). Последние будут использоваться для отображения значений переменных рецептов:



Рис. 11.3.8.15. Экран визуализации **Frame1**

Привяжем к полю вывода **переменной А** переменную **PLC_PRG.iA** (параметр **Текстовая переменная**) и зададим ей спецификатор формата вывода **%d**, используемый для отображения целочисленных значений (параметр **Текст**). Во вкладке **InputConfiguration** выберем параметр **OnClick** и привяжем к нему действие **Записать переменную**. Это позволит нам изменять значения переменных в процессе работы проекта.

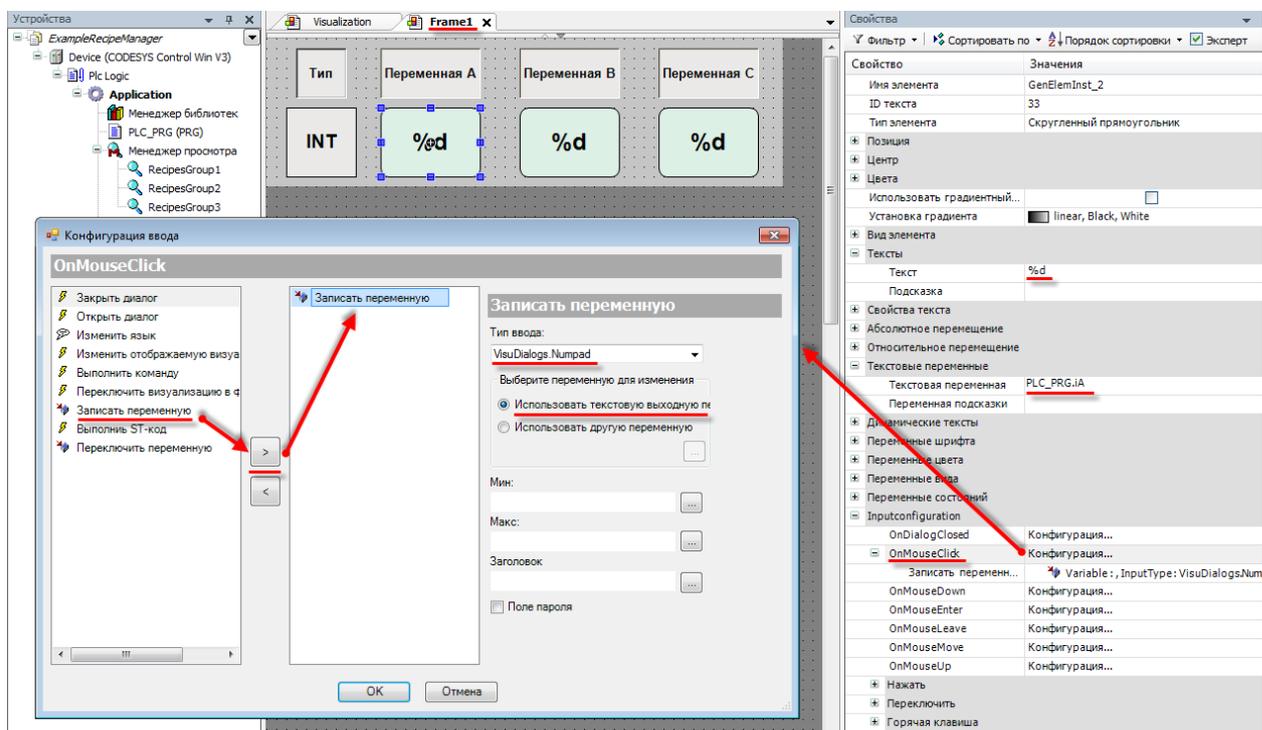


Рис. 11.3.8.16. Настройка полей вывода переменных типа INT

Аналогично настроим поля вывода **переменной В** и **С**. Очевидно, что они будут отличаться только привязываемыми к элементам переменными – **PLC_PRG.iB** и **PLC_PRG.iC** соответственно.

4.3. Экраны Frame2 и Frame3

Аналогично предыдущему пункту настроим экраны **Frame2** и **Frame3**. Поскольку их содержимое совершенно аналогично экрану **Frame1**, можно просто перенести все элементы с экрана на экран при помощи команд **Копировать/Вставить**.

Привяжем к экрану **Frame2** переменные типа REAL **PLC_PRG.rA**, **PLC_PRG.rB** и **PLC_PRG.rC** с форматом вывода **%.3f** (три знака после запятой), а к экрану **Frame3** - переменные типа BOOL **PLC_PRG.bA**, **PLC_PRG.bB** и **PLC_PRG.bC** с форматом вывода **%s** (для отображения в виде TRUE/FALSE).

Поскольку переменные типа **BOOL** принимают значения **TRUE/FALSE**, то в настройках их записи выберем диалог **VisuDialogs.Keypad**, который представляет собой экранную клавиатуру.

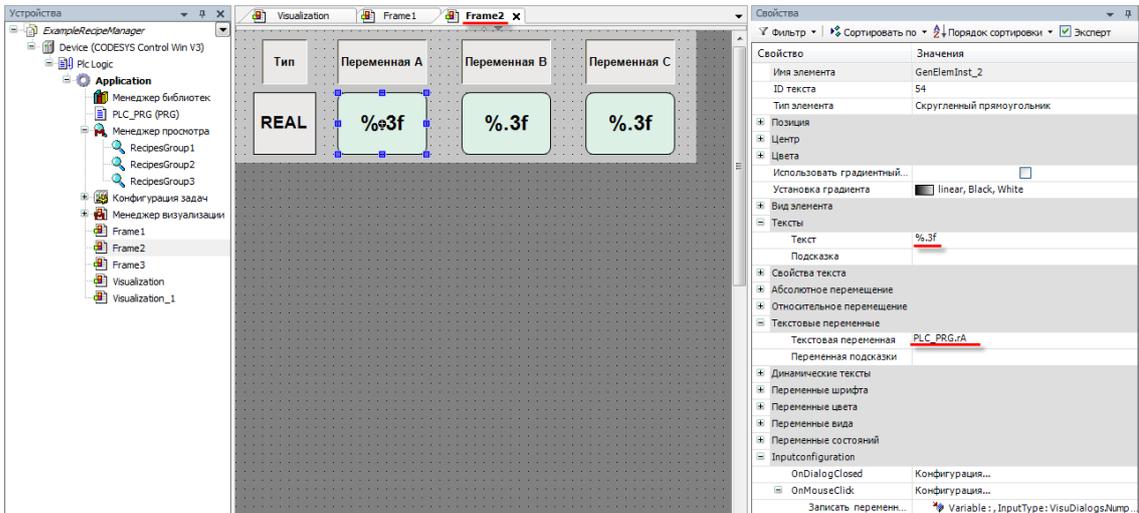


Рис. 11.3.8.17. Настройка полей вывода переменных типа REAL

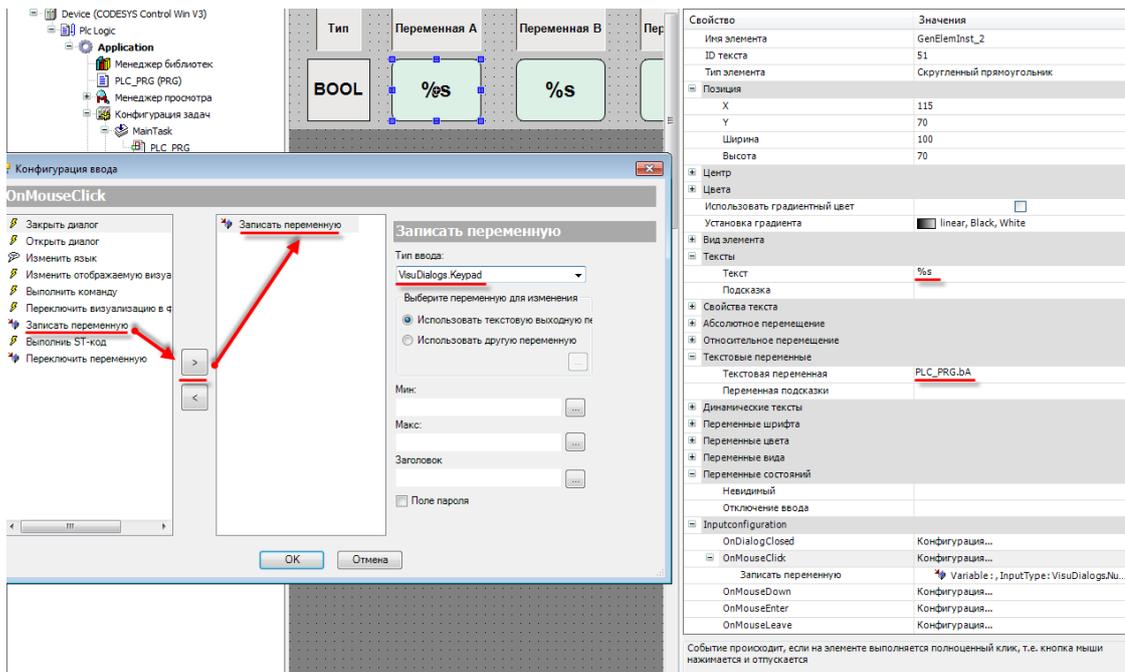


Рис. 11.3.8.18. Настройка полей вывода переменных типа BOOL

5. Экран Visualization (настройка действий)

5.1. Настройка элемента Фрейм

Теперь, после создания и настройки экранов фрейма, вернемся на экран **Visualization**. В конфигурации элемента **Фрейм** выберем экраны **Frame1**, **Frame2**, **Frame3**.

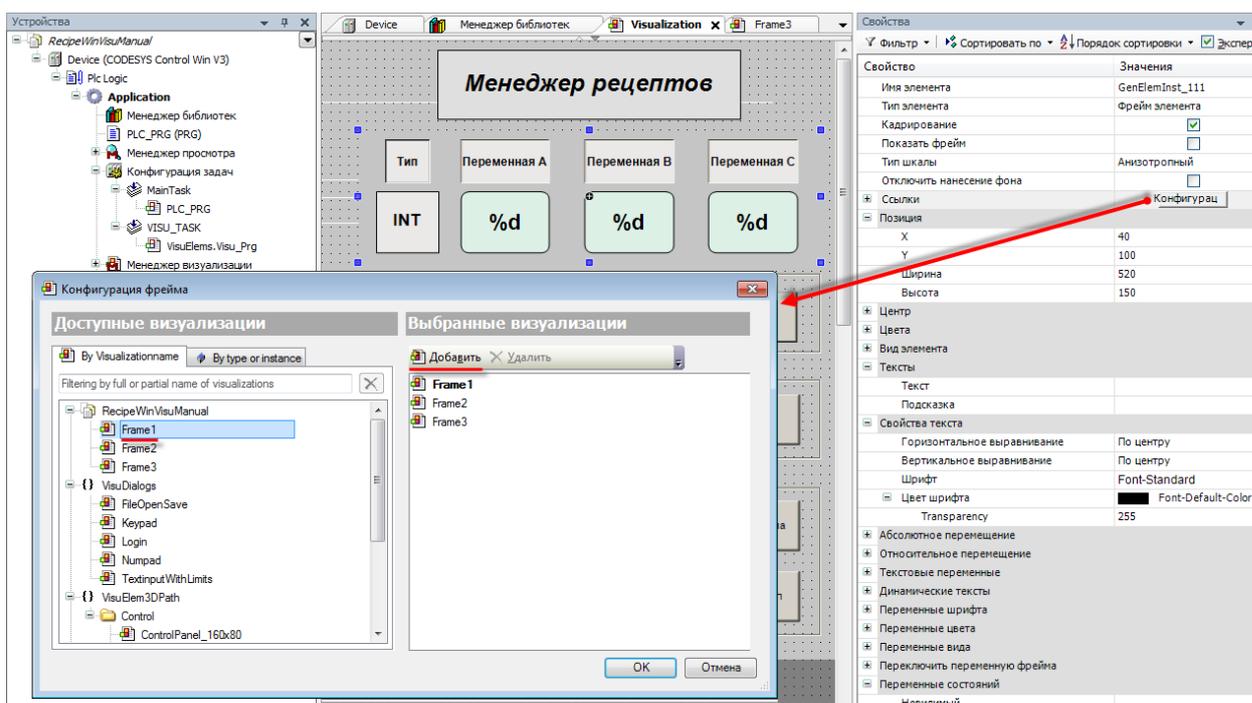


Рис. 11.3.8.19. Настройка содержимого элемента **Фрейм**

5.2. Настройка кнопок выбора групп рецептов

Настроим кнопку выбора группы рецептов **RecipesGroup1**. Выберем цвет тревоги (такой цвет кнопка будет принимать при выборе привязанной к ней группы рецептов), зададим ее текст, привяжем к параметрам **Переключить цвет** и **Двоичная переменная** переменную **PLC_PRG.bRecipesGroup1**. При нажатии на кнопку, эти переменные будут принимать значение **TRUE**, что приведет к изменению цвета кнопки и отображению ее нажатой.

Свойство	Значения
Имя элемента	GenElemInst_103
Тип элемента	Кнопка
ID текста	45
Позиция	
X	49
Y	284
Ширина	147
Высота	56
Центр	
Цвета	
Цвет	Element-Control-Color
Цвет тревоги	0; 203; 115
Использовать градиентный цвет	<input type="checkbox"/>
Установка градиента	linear, Black, White
Высота кнопки	0
Информация изображения	
Тексты	
Текст	RecipesGroup1
Подсказка	
Свойства текста	
Абсолютное перемещение	
Относительное перемещение	
Текстовые переменные	
Динамические тексты	
Переменные шрифта	
Переменные цвета	
Переключить цвет	PLC_PRG.bRecipesGroup1
Цвет	
Цвет тревоги	
Переменные состояний	
Невидимый	
Отключение ввода	
Переменная состояния кнопки	
Двоичная переменная	PLC_PRG.bRecipesGroup1

Рис. 11.3.8.20. Настройки кнопки выбора групп рецептов **RecipesGroup1**

Во вкладке [InputConfiguration](#) кнопки **RecipesGroup1** выберем параметр **OnClick** и привяжем к нему два действия: [Выполнить ST-код](#) и [Переключить визуализацию во фрейме](#). Для первого действия напишем код на языке **ST** (рис. 11.3.8.21), для второго - выберем **Frame1** (рис. 11.3.8.22).

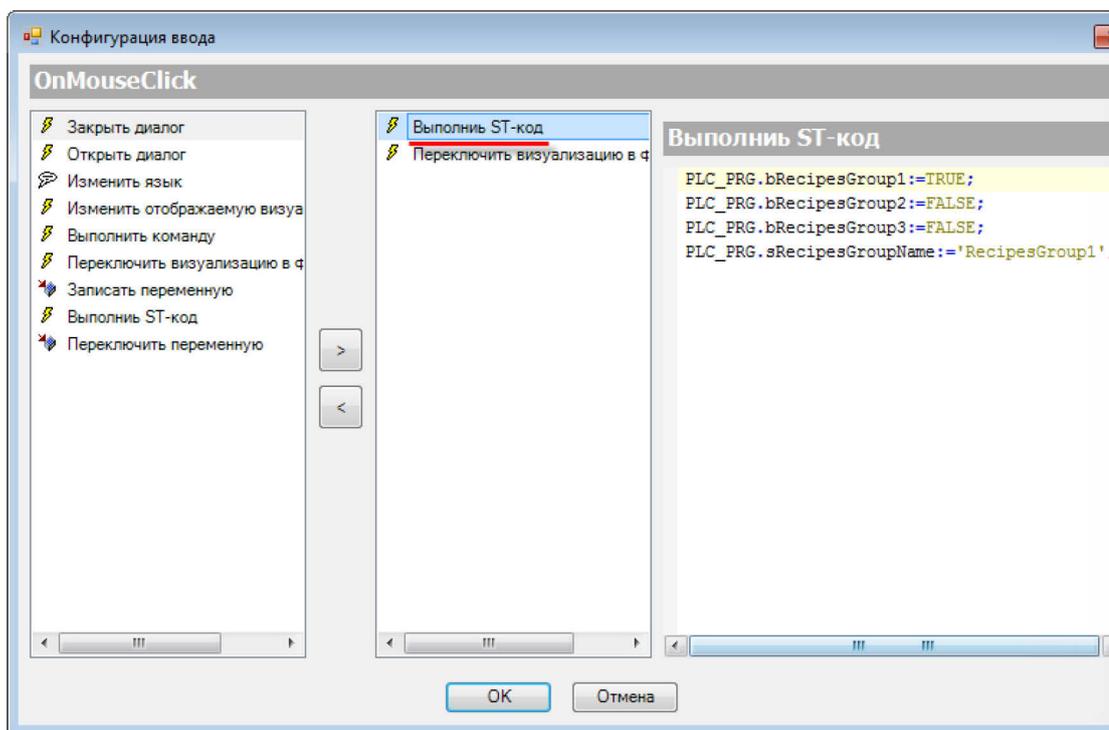


Рис. 11.3.8.21. Настройка параметра **OnClick** кнопки **RecipesGroup1**, действие **Выполнить ST-код**

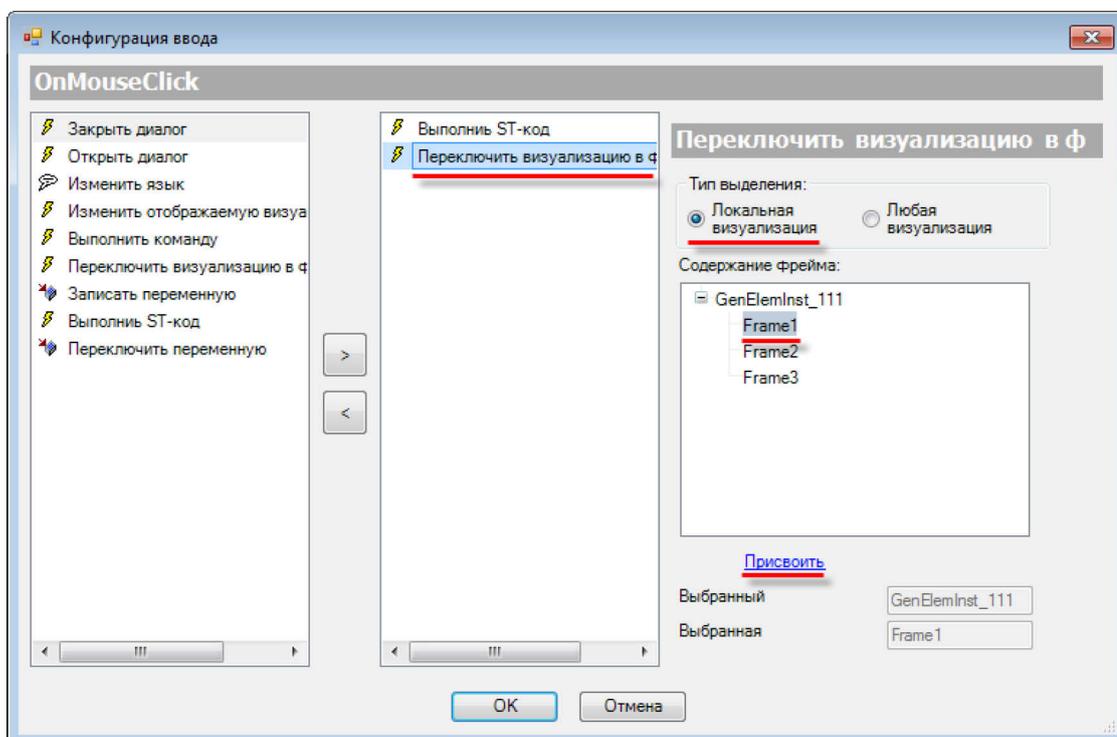


Рис. 11.3.8.22. Настройка параметра **OnClick** кнопки **RecipesGroup1**, действие **Переключить визуализацию во фрейме**

Соответственно, при нажатии кнопки выбора **RecipesGroup1** будут «отжиматься» все остальные кнопки выбора групп, в переменную **PLC_PRG.sRecipesGroupName** – записываться имя данной группы, а во фрейме будут отображаться поля вывода ее переменных.

Аналогично настроим кнопки выбора группы **RecipesGroup2** и **RecipesGroup3**. ST-код для них приведен на рис. 11.3.8.23. и рис. 11.3.8.24:

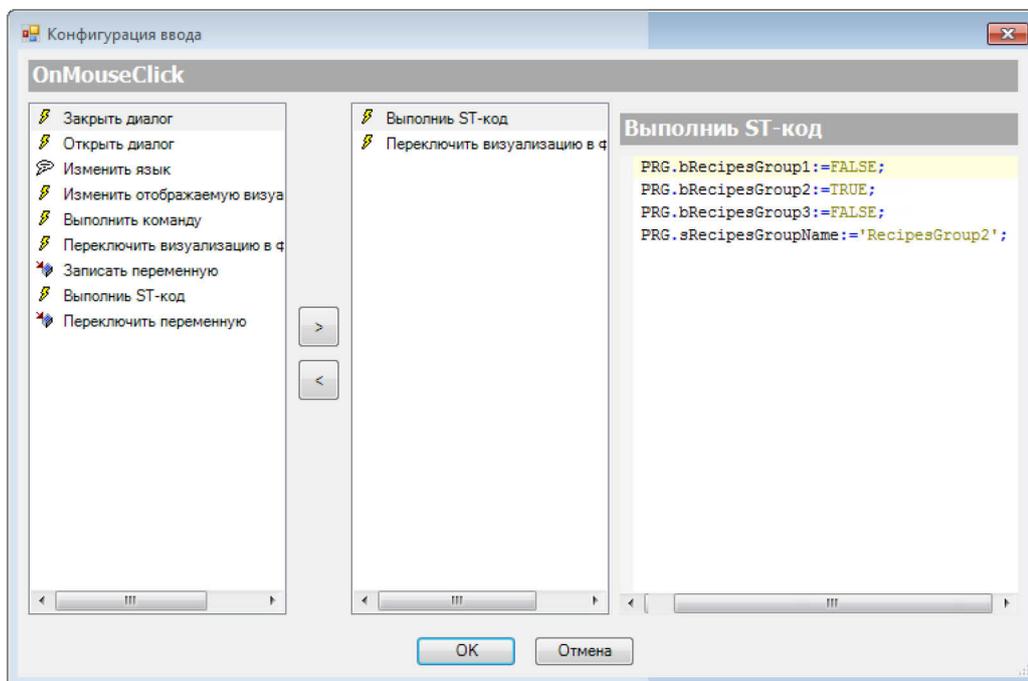


Рис. 11.3.8.23. Настройка параметра **OnMouseClicked** кнопки **RecipesGroup2**, действие **Выполнить ST-код**

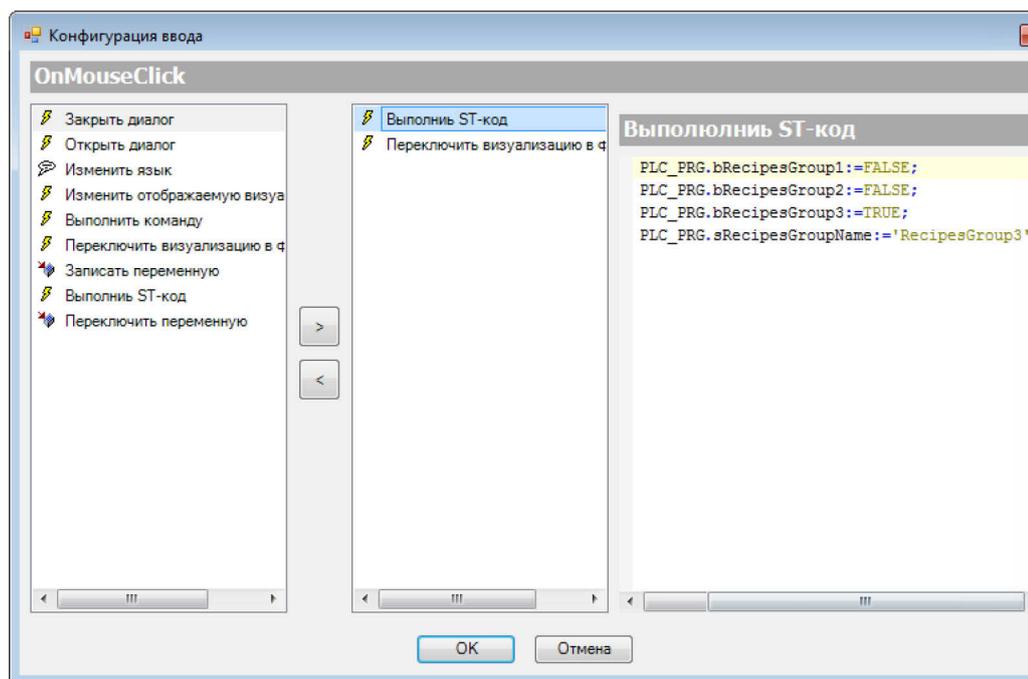


Рис. 11.3.8.24. Настройка параметра **OnMouseClicked** кнопки **RecipesGroup3**, действие **Выполнить ST-код**

5.3. Настройка кнопок выбора рецептов

Настроим кнопку выбора рецепта **Recipe1**. Выберем цвет тревоги (такой цвет кнопка будет принимать при выборе привязанного к ней рецепта), зададим ее текст, привяжем к параметрам **Переключить цвет** и **Двоичная переменная** переменную **PLC_PRG.bRecipe1**. При нажатии на кнопку, это переменные будут принимать значение **TRUE**, что приведет к изменению цвета кнопки и отображению ее нажатой.

Свойство	Значения
Имя элемента	GenElemInst_30
Тип элемента	Кнопка
ID текста	12
Позиция	
X	52
Y	400
Ширина	147
Высота	56
Центр	
Цвета	
Цвет	Element-Control-Color
Цвет тревоги	0; 203; 115
Использовать градиентный цвет	<input type="checkbox"/>
Установка градиента	linear, Black, White
Высота кнопки	0
Информация изображения	
Тексты	
Текст	Recipe1
Подсказка	
Свойства текста	
Абсолютное перемещение	
Относительное перемещение	
Текстовые переменные	
Динамические тексты	
Переменные шрифта	
Переменные цвета	
Переключить цвет	PLC_PRG.bRecipe1
Цвет	
Цвет тревоги	
Переменные состояний	
Невидимый	
Отключение ввода	
Переменная состояния кнопки	
Двоичная переменная	PLC_PRG.bRecipe1
Переменная ID изображения	
Inputconfiguration	

Рис. 11.3.8.25. Настройки кнопки выбора рецепта **Recipe1**

Во вкладке [InputConfiguration](#) кнопки **Recipe1** выберем параметр **OnClick** и привяжем к нему действие [Выполнить ST-код](#):

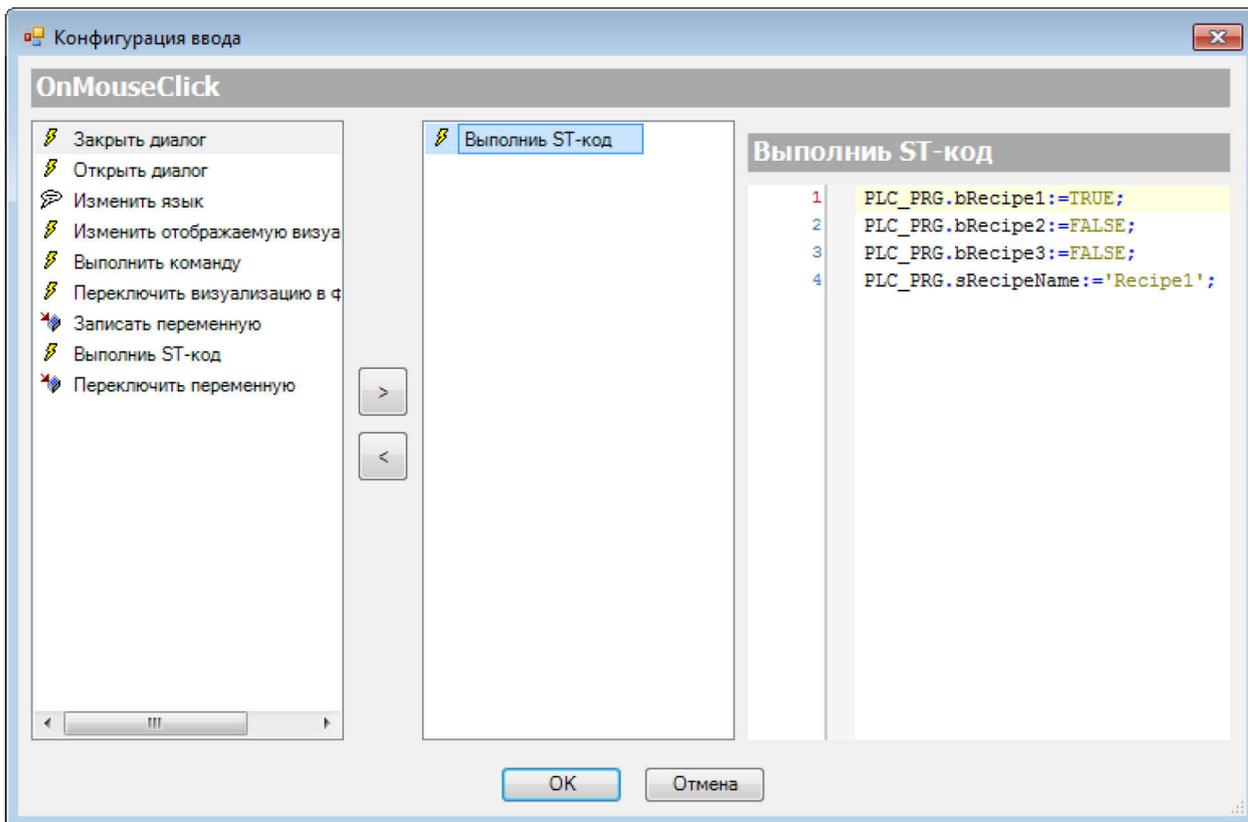


Рис. 11.3.8.26. Настройка параметра **OnClick** кнопки **Recipe1**, действие **Выполнить ST-код**

Соответственно, при нажатии кнопки выбора **Recipe1** будут «отжиматься» все остальные кнопки выбора рецептов, в переменную **PLC_PRG.sRecipeName** записываться имя данного рецепта.

Аналогично настроим кнопки выбора рецептов **Recipe2** и **Recipe3**. ST-код для них приведен на рис. 11.3.8.27 и рис. 11.3.8.28.

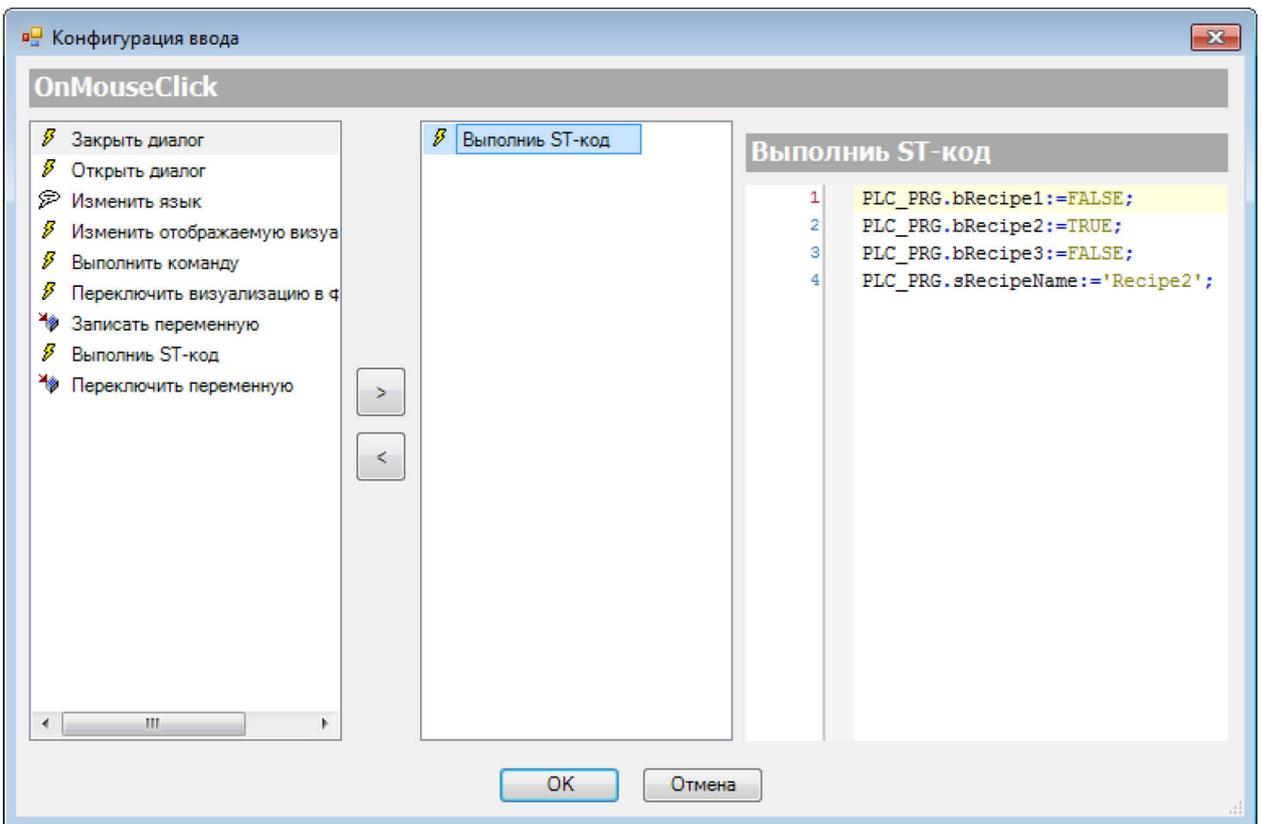


Рис. 11.3.8.27. Настройка параметра **OnMouseClicked** кнопки **Recipe2**, действие **Выполнить ST-код**

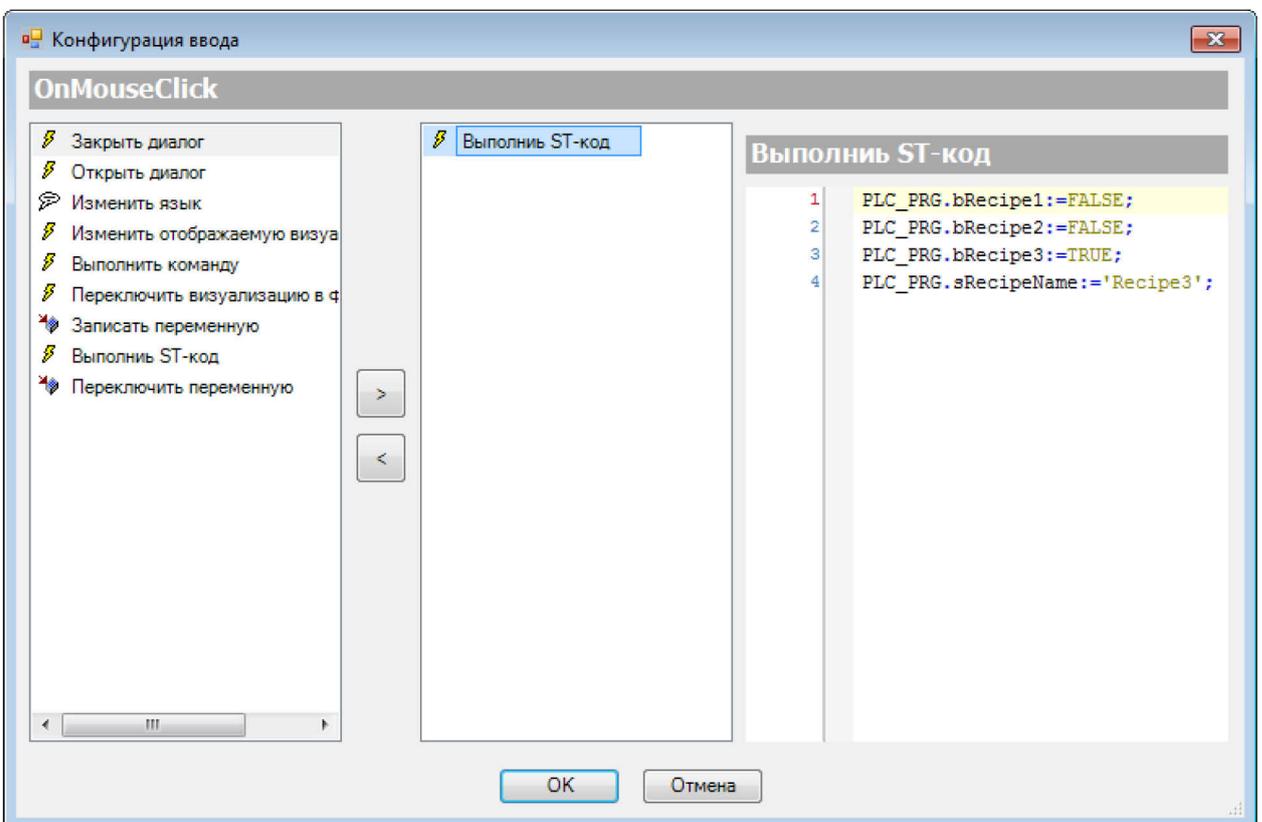


Рис. 11.3.8.28. Настройка параметра **OnMouseClicked** кнопки **Recipe3**, действие **Выполнить ST-код**

5.4. Настройка кнопок управления рецептами

Настроим кнопки управления рецептами. К этим кнопкам не привязывается никаких переменных, поэтому настройка ограничивается выбором на вкладке [InputConfiguration](#) параметра **OnClick** и привязкой ему действия [Выполнить команду](#).

Для кнопки **Записать значения в список и файл рецепта** выберем команду **ReadRecipe** и добавим ее с помощью кнопки «+». Не следует понимать англоязычное название команды буквально – она читает текущие значения переменных программы и записывает их в список рецепта, а благодаря галочке вкладки **Общее** (рис. 11.3.8.5) после этого автоматически обновляется и файл рецепта.

В свою очередь, команда **WriteRecipe** записывает значения рецепта в переменные программы, поэтому привяжем ее к кнопке **Считать значения из списка**.

Аргументами любой из команд, связанной с рецептами, являются строковые переменные, содержащие имя группы рецептов и имя рецепта, с которым производится действие. В нашем случае это переменные **PLC_PRG.sRecipesGroupName** и **PLC_PRG.sRecipeName**.

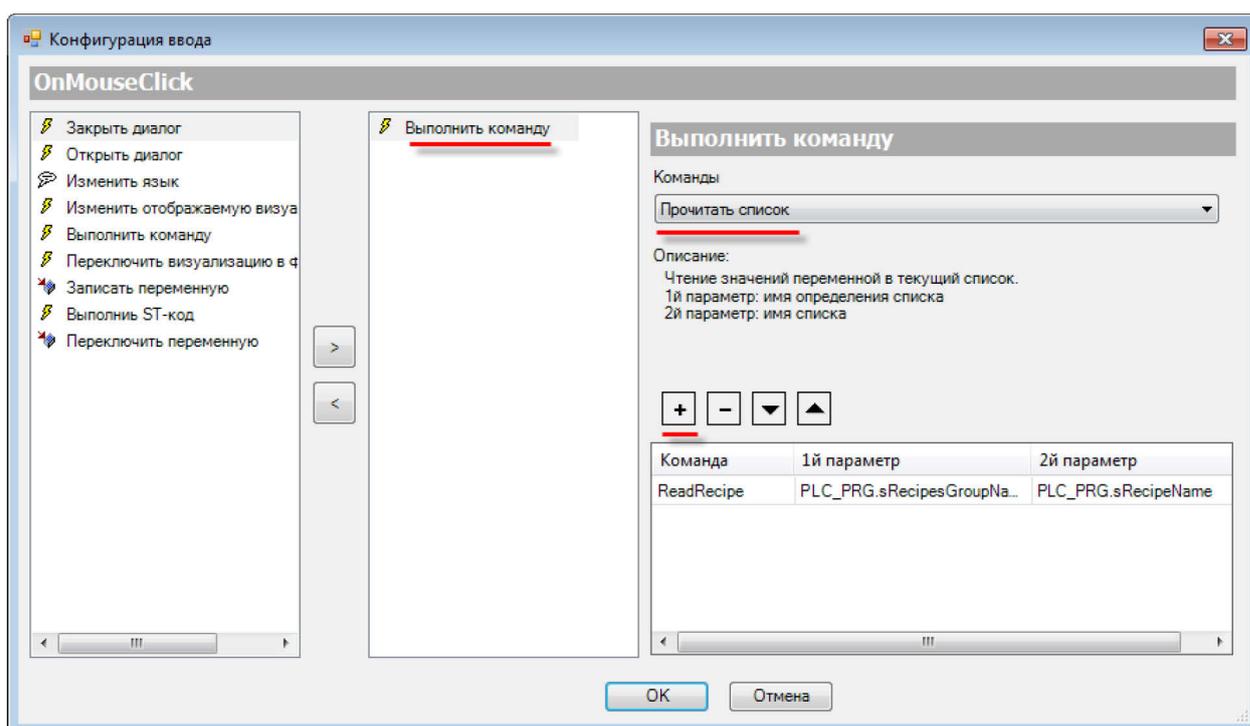


Рис. 11.3.8.29. Настройка параметра **OnClick** кнопки **Записать значения в список и файл рецепта**, команда **Прочитать список (ReadRecipe)**

Аналогично настроим остальные кнопки (отличие будет только в выбираемых командах).

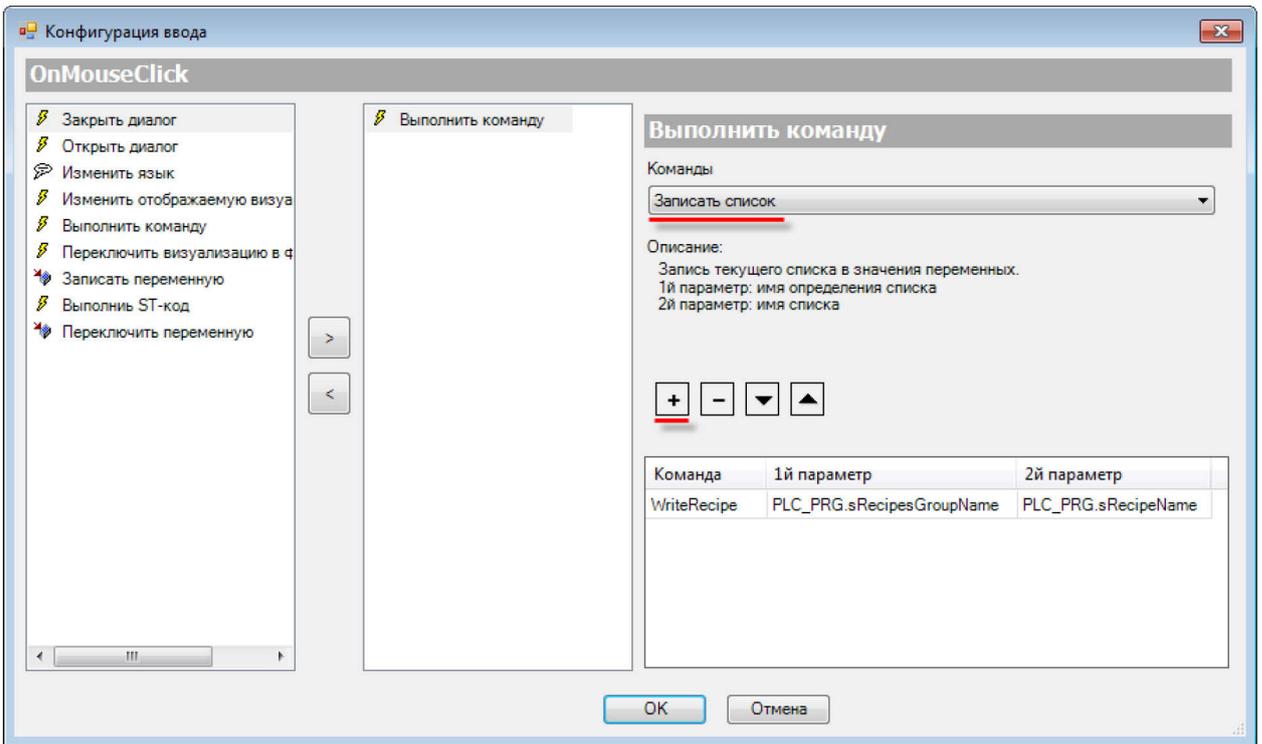


Рис. 11.3.8.30. Настройка параметра **OnMouseClicked** кнопки **Считать значения из списка**, команда **Записать список (WriteRecipe)**

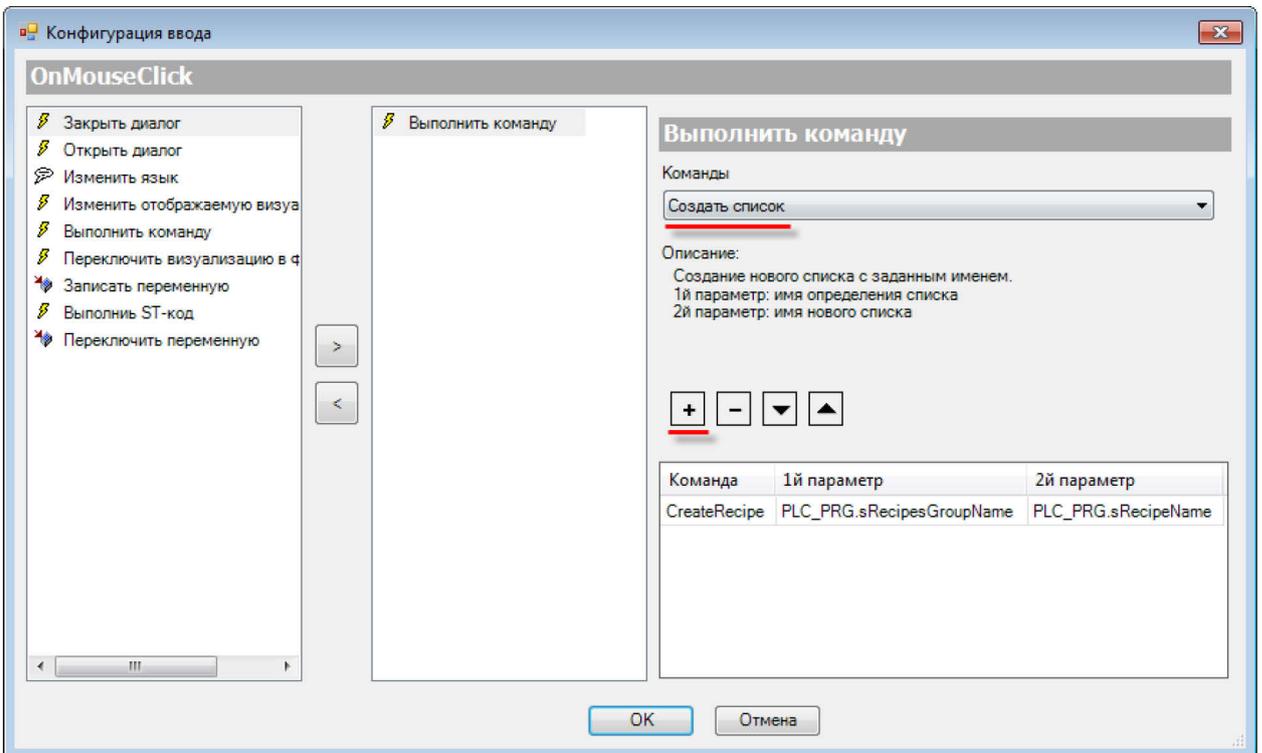


Рис. 11.3.8.31. Настройка параметра **OnMouseClicked** кнопки **Создать рецепт**, команда **Создать список (CreateRecipe)**

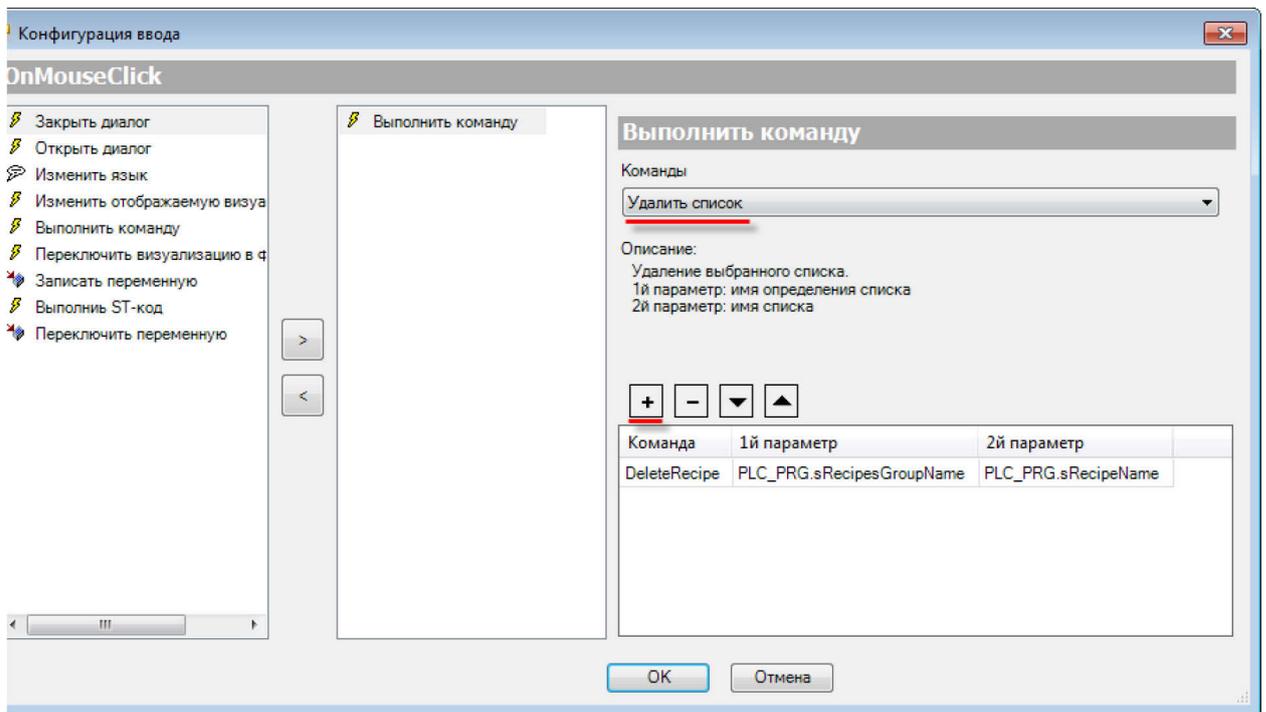


Рис. 11.3.8.32. Настройка параметра **OnMouseClicked** кнопки **Удалить рецепт**, команда **Удалить список (DeleteRecipe)**

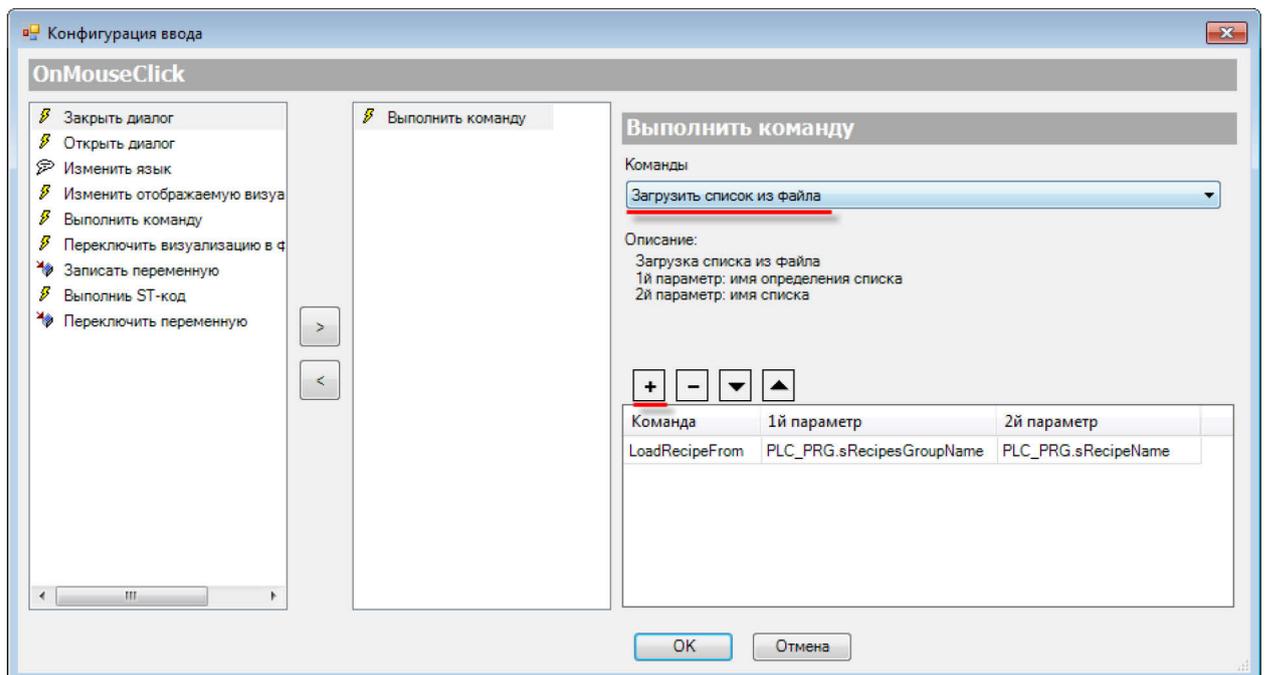


Рис. 11.3.8.33. Настройка параметра **OnMouseClicked** кнопки **Загрузить из файла**, команда **Загрузить список из файла (LoadRecipeFrom)**

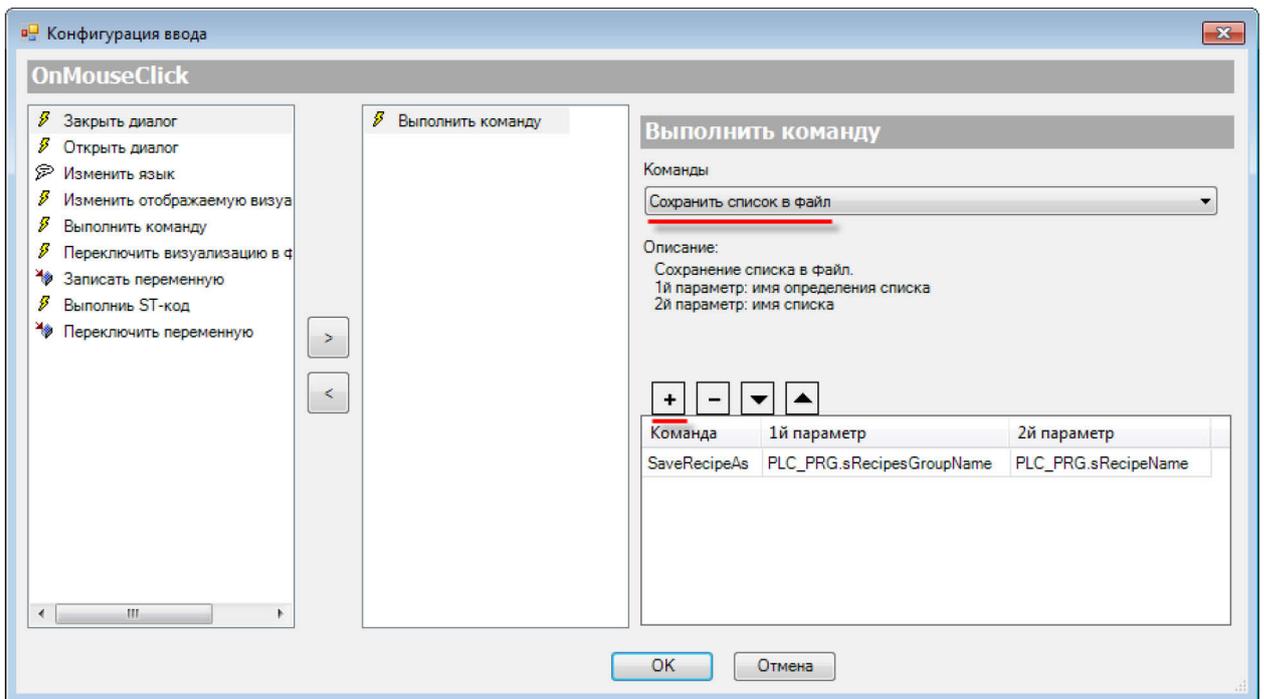
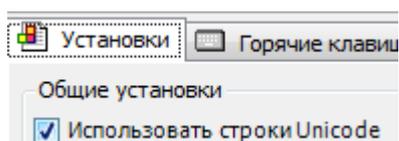


Рис. 11.3.8.34. Настройка параметра **OnMouseClicked** кнопки **Сохранить в файл**, команда **Сохранить список в файл (SaveRecipeAs)**

6. Настроим [Менеджер визуализации](#). В его установках поставим галочку **Использовать строки Unicode**:



Настройки **target**- и **web**-визуализации будут выглядеть следующим образом:

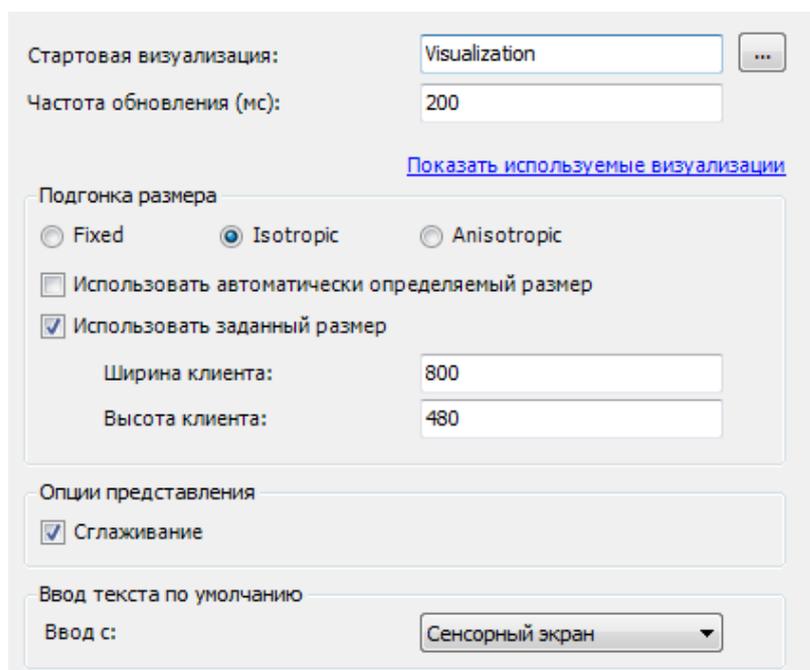


Рис. 11.3.8.35. Настройки **target**-визуализации

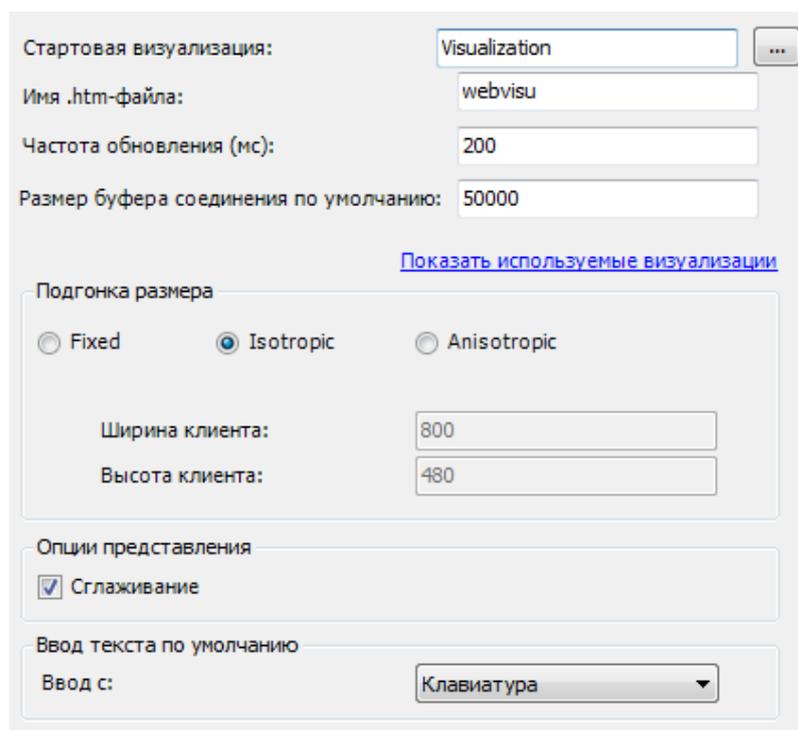


Рис. 11.3.8.36. Настройки **web**-визуализации

7. [Запустим проект на виртуальном контроллере](http://localhost:8080/webvisu.htm). Откроем в веб-браузере страницу <http://localhost:8080/webvisu.htm>, чтобы увидеть web-визуализацию проекта. **Обратите внимание**, что браузер должен поддерживать **HTML5**. Проверим функционал проекта.

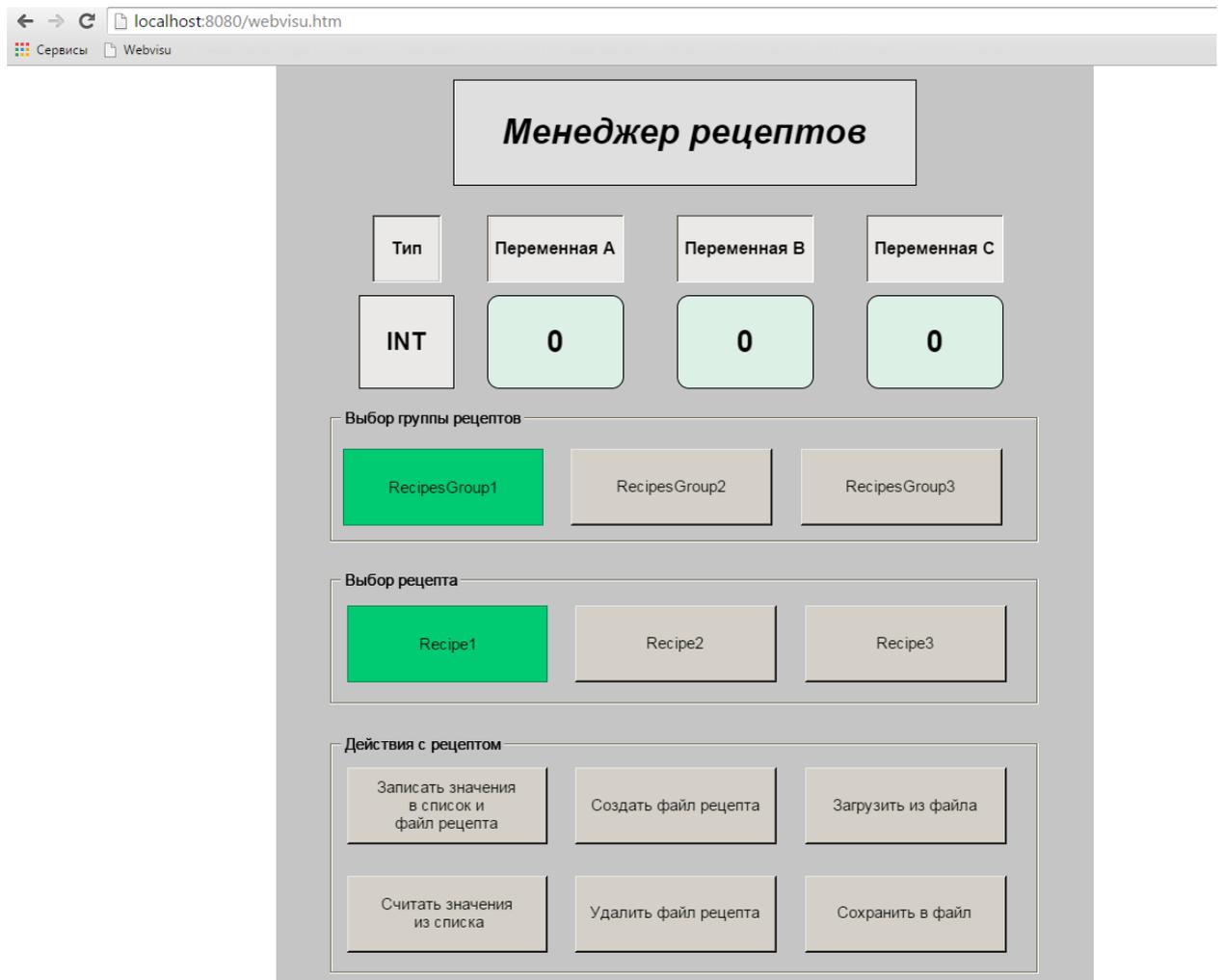


Рис. 11.3.8.37. Web-визуализация проекта в браузере **Chrome**

Обратите внимание, что после запуска проекта, в папке **D:\recipes** (указанной в Менеджере рецептов, см. рис. 11.3.8.4) были созданы **файлы рецептов**.

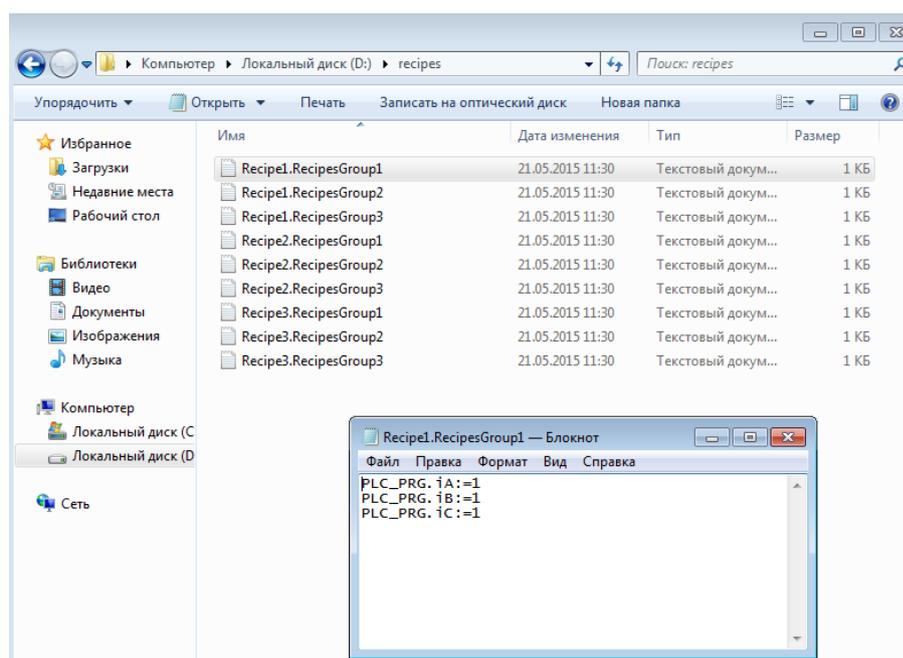


Рис. 11.3.8.38. Файлы рецептов

Проверим работу кнопок действий с рецептами:

Нажатие на кнопку **Записать значения в список и файл рецепта** приводит к записи текущих значений переменных программы в список рецепта и файл рецепта;

Нажатие на кнопку **Считать значения из списка** приводит к записи текущих значений из рецепта в переменные программы и, соответственно, их отображению на экране;

Нажатие на кнопку **Удалить файл рецепта** приводит к удалению файла рецепта;

Нажатие на кнопку **Создать файл рецепта** приводит к созданию файла рецепта со значениями текущих переменных программы. Название файла определяется выбранным рецептом и его группой. **Обратите внимание**, что таким образом нельзя перезаписать существующий файл – только создать новый;

Нажатие на кнопку **Загрузить из файла** позволяет выбрать файл рецепта, значения из которого будут записаны в переменные программы. Конфигурация файла рецепта должна совпадать с конфигурацией группы рецептов, в рецепт которой загружаются значения из файла (нельзя загрузить значения типа **INT** в рецепт, состоящий из переменных типа **BOOL**);

Нажатие на кнопку **Сохранить в файл** позволяет сохранить текущие значения переменных программы в файл рецепта с произвольным названием. Расширение указывать не обязательно – оно будет сформировано автоматически.

Обратите внимание, что на целевых устройствах с ОС **Linux** следует строго соблюдать регистр названий рецептов и их групп, т.к. в данном случае **Recipe1.RecipeGroup1.txt** и **recipe1.recipegroup1.txt** – это два разных файла

11.3.9. Конвертация значений (Unit Conversion)

Данный пример посвящен работе с компонентом **Unit Conversion**, который позволяет создавать **шаблоны преобразований** переменных (сдвиг, масштабирование и т.д.) и привязывать их к элементам визуализации – т.е. можно с помощью двух элементов, к которым привязана одна и та же переменная, отобразить разные значения, полученные путем использования различных шаблонов конвертации.

Очевидно, что подобные преобразования можно проводить непосредственно в программе, но это приведет к увеличению количества кода и используемых переменных.

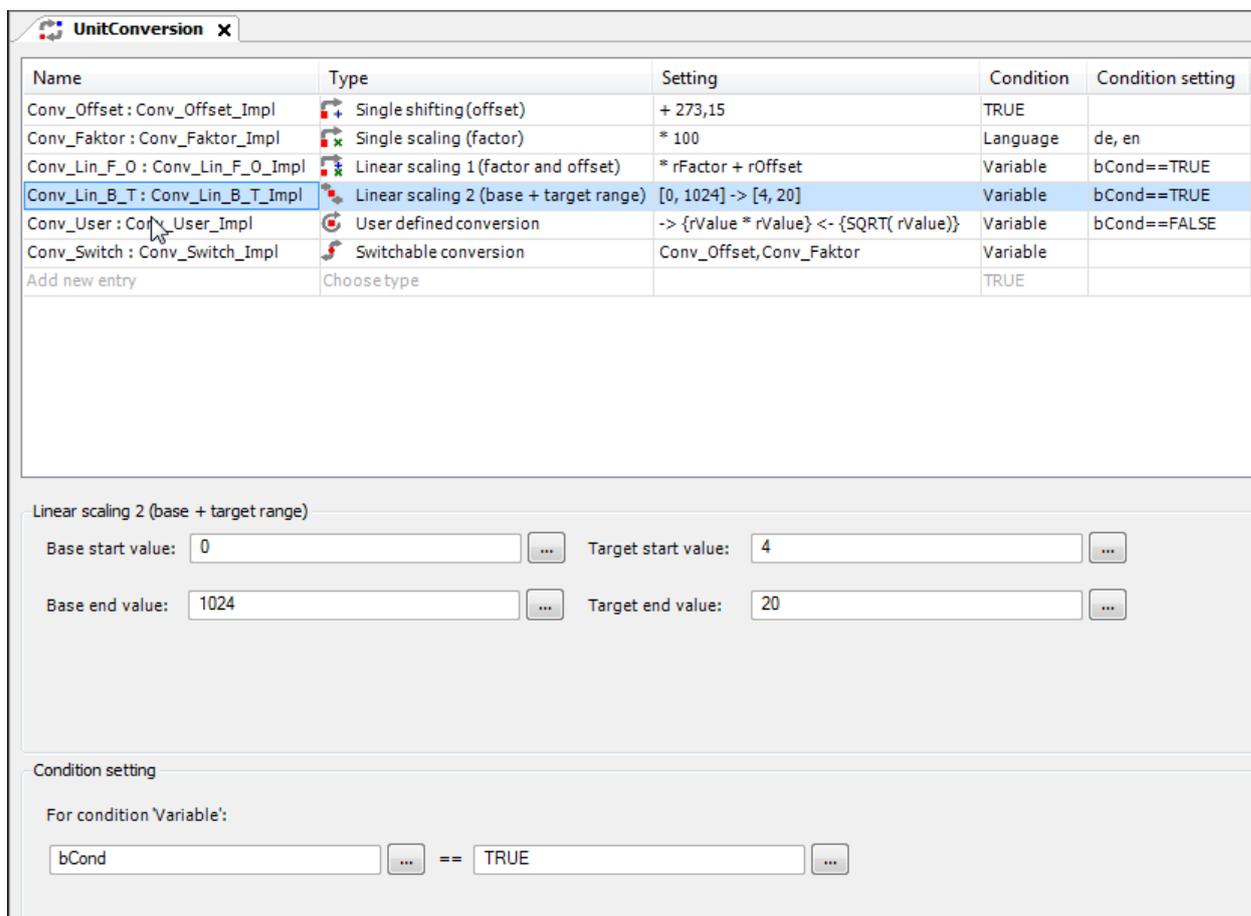


Рис. 11.3.9.1. Внешний вид компонента **Unit Conversion**

Ниже рассмотрен пример преобразования токового сигнала **4...20 mA** в температуру из диапазона **-50...150 °C** (при допущении о линейной зависимости между величинами).

Пример создан в среде **CODESYS 3.5 SP6 Patch 2** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с target-файлом **3.5.6.0**.

Пример доступен для скачивания: [Example_UnitConversion.projectarchive](#)

1. Создадим новый **стандартный** проект **CODESYS** с названием **Example_UnitConversion** и настройками по умолчанию: target – **CODESYS Control Win V3**, язык PLC_PRG – **ST**.

2. В программе **PLC_PRG** объявим переменную **rCurrentLoop** типа **REAL**:

```
PLC_PRG x
1 PROGRAM PLC_PRG
2 VAR
3     rCurrentLoop:REAL:=4; // ТОКОВЫЙ СИГНАЛ ИЗ ДИАПАЗОНА 4..20 мА
4 END_VAR
```

Рис. 11.2.9.2. Объявление переменных программы **PLC_PRG**

3. Добавим в проект компонент **Unit Conversion** с названием по умолчанию:

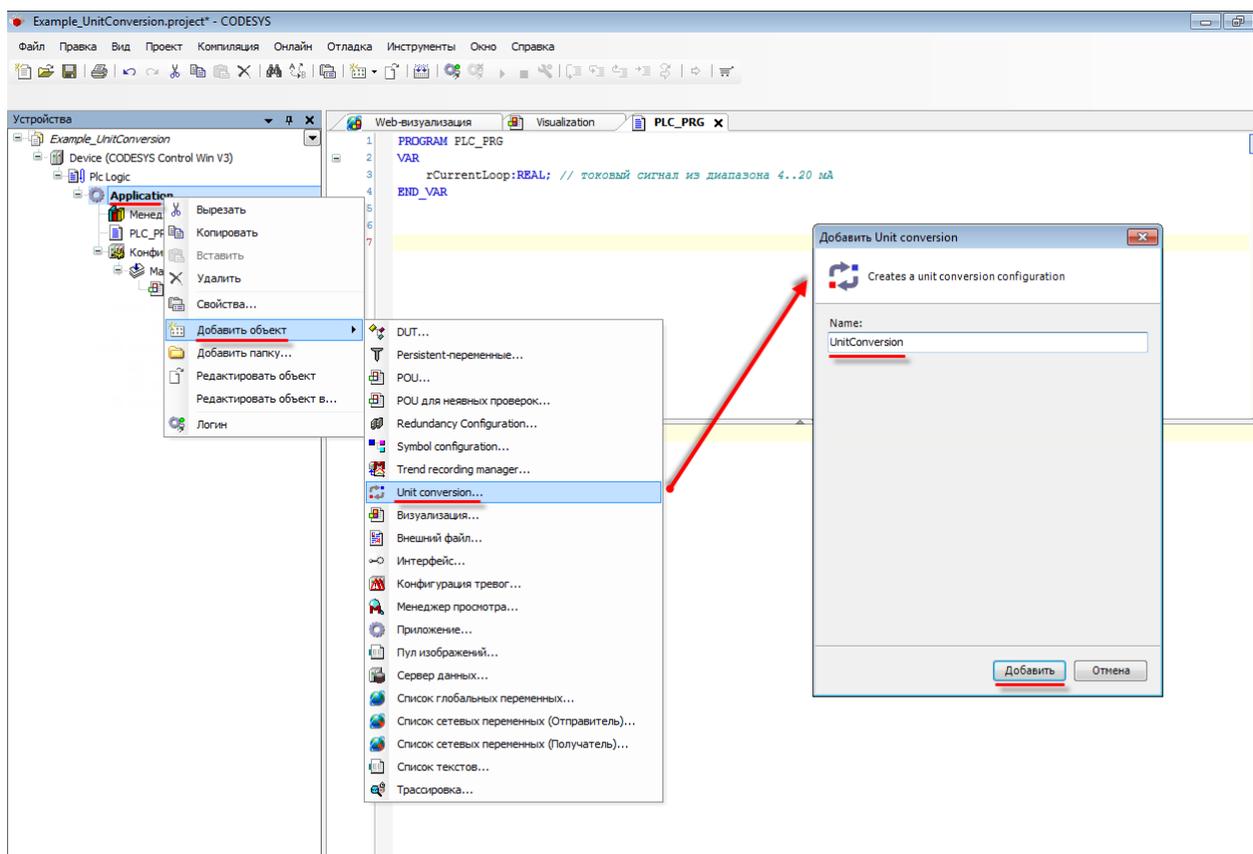


Рис. 11.2.9.3. Добавление в проект компонента **Unit Conversion**

4. Добавим новый шаблон преобразования **CurrentLoopToTemperature** типа **Linear scaling 1 (factor and offset)** **1 (factor and offset)**:

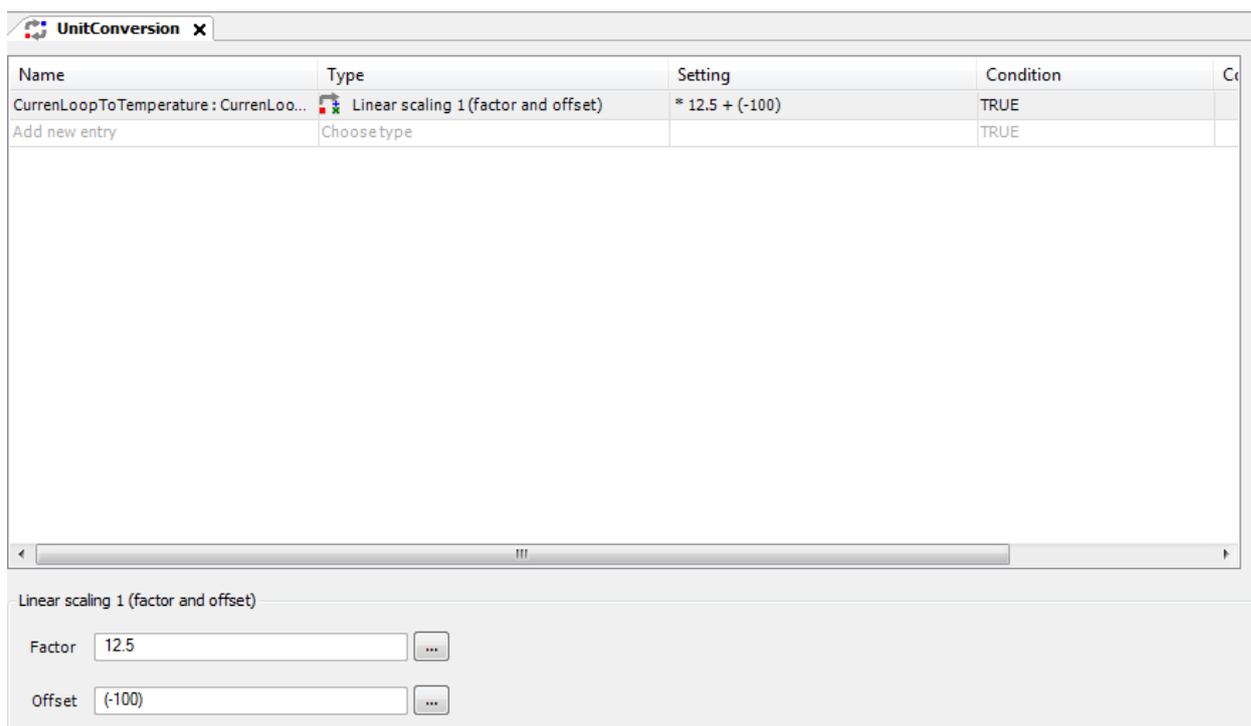


Рис. 11.2.9.4. Настройка компонента **Unit Conversion**

Напомним, сформулированная нами задача – преобразовать значение токового сигнала **4...20 мА** в соответствующее ему значение температуры из диапазона **-50...150 °С** (при допущении о линейной зависимости между величинами).

После решения системы из двух линейных уравнений, получаем зависимость температуры от значения тока:

$$t = 12.5 \cdot I - 100, \text{ где}$$

I – значение токового сигнала, мА

Соответственно, в столбце **Setting** для нашего шаблона зададим **Factor = 12.5** и **Offset=(-100)**.

В столбце **Condition** по умолчанию стоит значение **TRUE** – это означает, что преобразование будет выполняться всегда.

Подробное описание компонента приведено в пп. 8.

5. [Добавим в проект](#) экран визуализации **Visualization**. В его [свойствах](#) выберем размер **480x272**. Экран будет содержать элемент [Бегунок](#) и элемент [Текстовое поле](#). Поясняющая надпись сделана с помощью элемента [Метка](#).

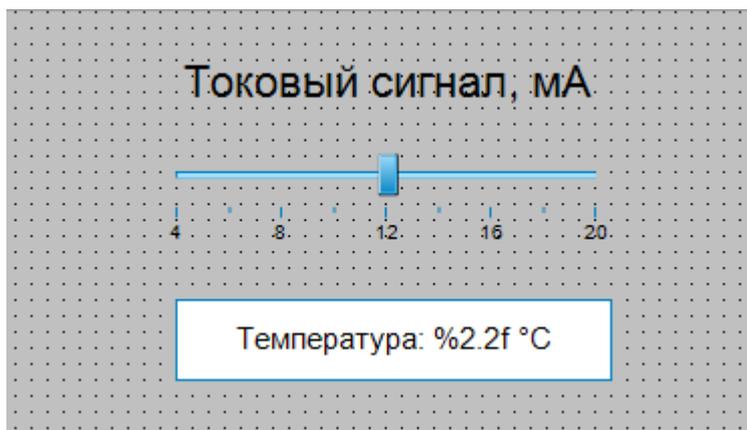


Рис. 11.2.9.5. Содержание экрана **Visualization**

Настроим элемент [Бегунок](#), который будет использоваться для изменения значения токового сигнала:

Свойства	
Фильтр	Сортировать по
Порядок сортировки	Эксперт
Свойство	Значения
Имя элемента	GenElemInst_249
Тип элемента	Бегунок
Позиция	
X	90
Y	80
Ширина	300
Высота	80
Переменная	PLC_PRG.rCurrentLoop
Размер страницы	
Прокрутка выполнена	<input checked="" type="checkbox"/>
Шкала	
Показать шкалу	<input checked="" type="checkbox"/>
Начало шкалы	4
Конец шкалы	20
Основная шкала	4
Подшкала	2
Формат шкалы (с-син...	%d
Линейка	
Тип диаграммы	Низ
Ориентация	Горизонтально
Направление движения	Слева направо
Переменные состояний	
Невидимый	
Отключение ввода	

Рис. 11.2.9.6. Настройки элемента **Бегунок**

Элемент **Текстовое поле** будет использоваться для отображения преобразованного значения. Для этого привяжем к нему переменную **rCurrentLoop**, после чего в строке параметра **Текстовая переменная** нажмем на ярлык **Unit Conversion** и в появившемся окне выберем шаблон нашего преобразования:

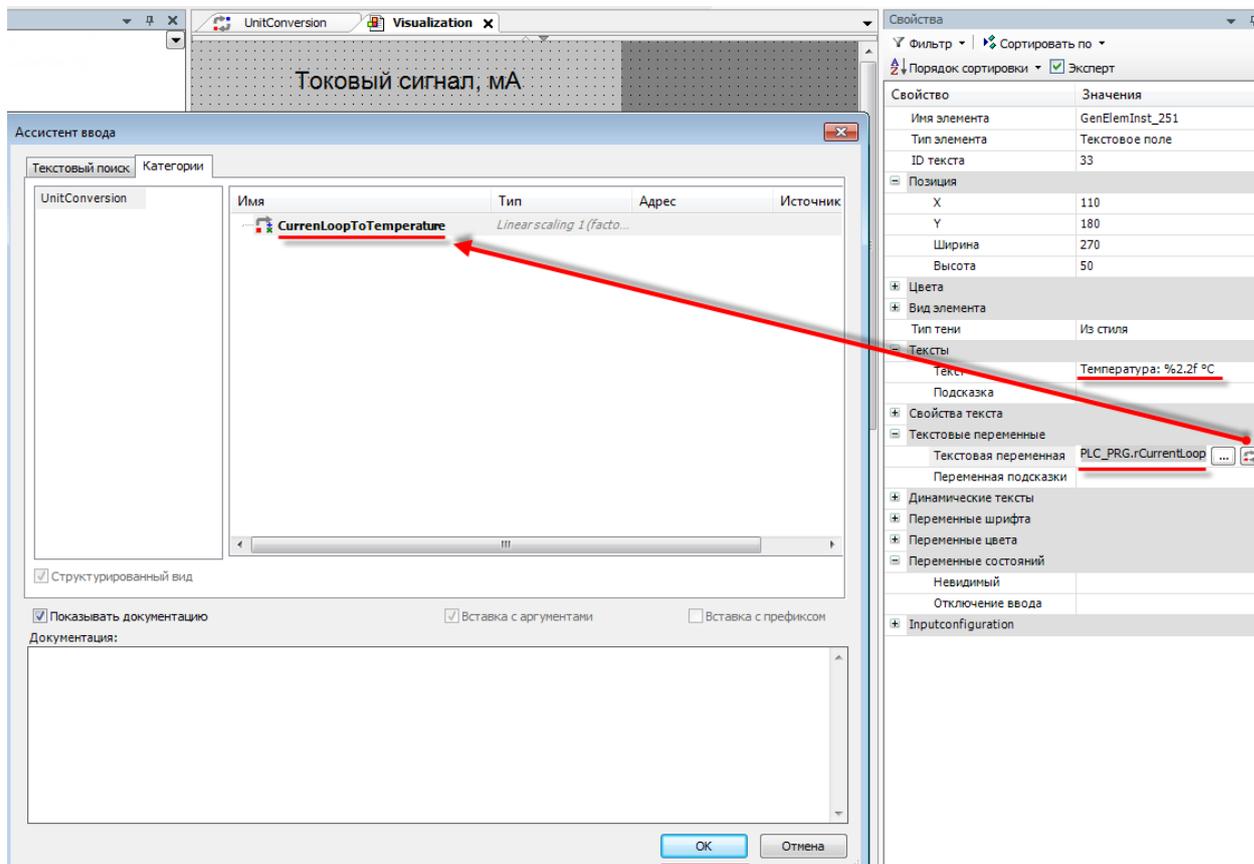
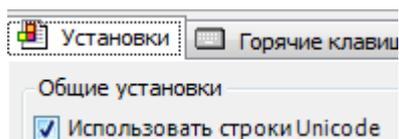


Рис. 11.2.9.7. Выбор шаблона преобразования для элемента **Текстовое поле**

Обратите внимание, что возможность выбора шаблона появляется только после добавления в проект компонента **Unit Conversion**.

6. Настроим [Менеджер визуализации](#). В его установках поставим галочку Использовать строки Unicode:



Настройки **target**- и **web**-визуализации будут выглядеть следующим образом:

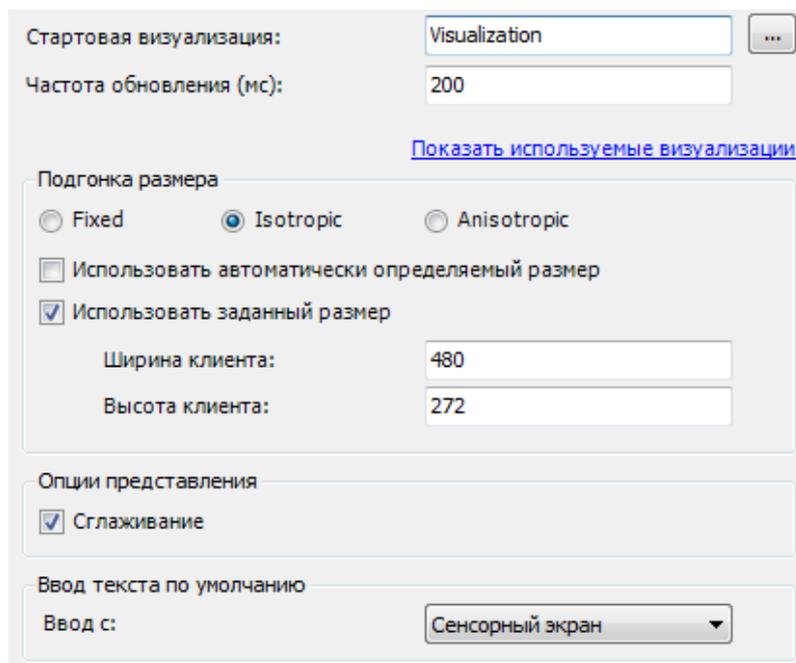


Рис. 11.3.9.8. Настройки **target**-визуализации

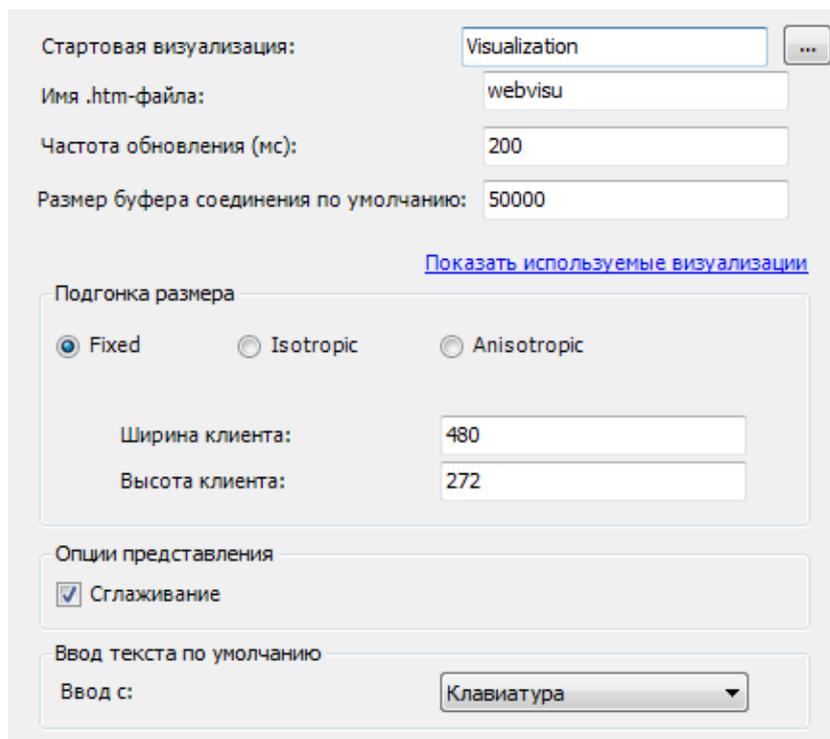


Рис. 11.3.9.9. Настройки **web**-визуализации

7. [Запустим проект на виртуальном контроллере](#). Откроем в веб-браузере страницу <http://localhost:8080/webvisu.htm>, чтобы увидеть web-визуализацию проекта. **Обратите внимание**, что браузер должен поддерживать **HTML5**. Проверим функционал проекта.

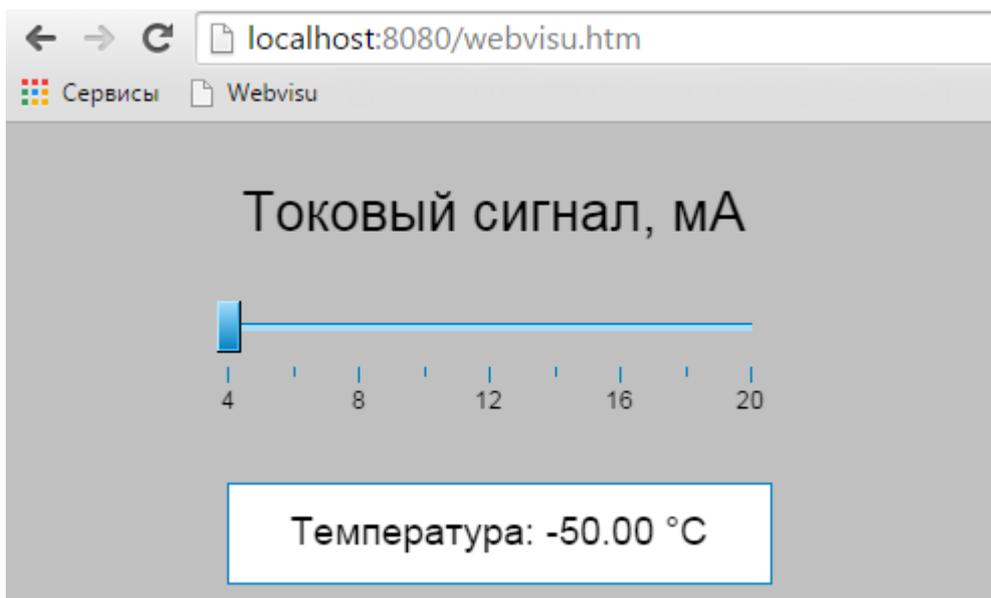


Рис. 11.3.9.10. Web-визуализация проекта в браузере **Chrome**

По умолчанию токовый сигнал равен **4 мА**, что соответствует нижней границе температурного диапазона (**-50 °C**). Меняя значение токового сигнала, наблюдайте за соответствующим изменением температуры. Напоминаем, что к обоим элементам привязана одна и та же переменная.

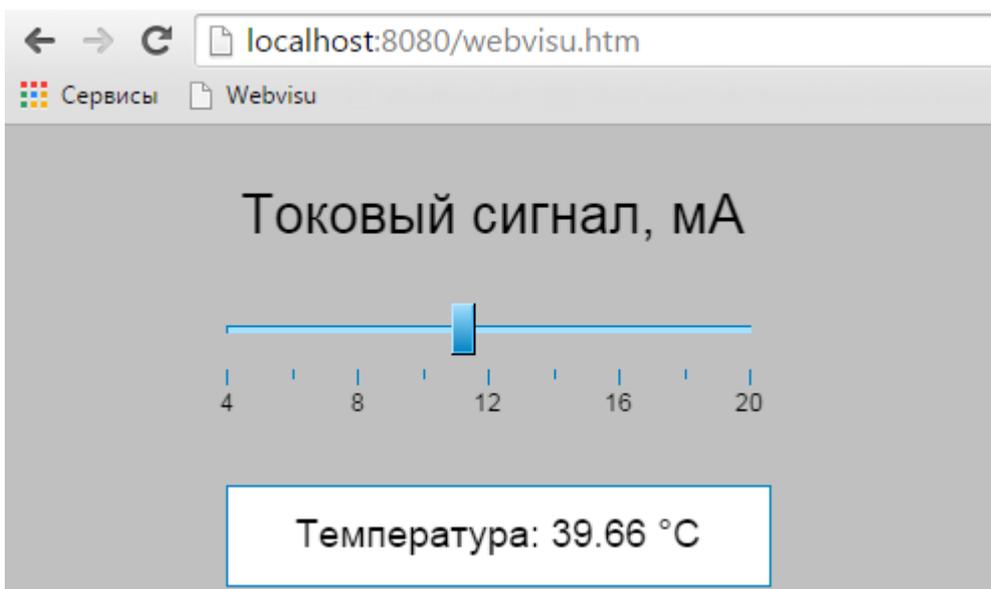


Рис. 11.3.9.11. Конвертация токового сигнала в температуру

Полное описание компонента **Unit Conversion** приведено ниже.

8. Описание компонента Unit Conversion

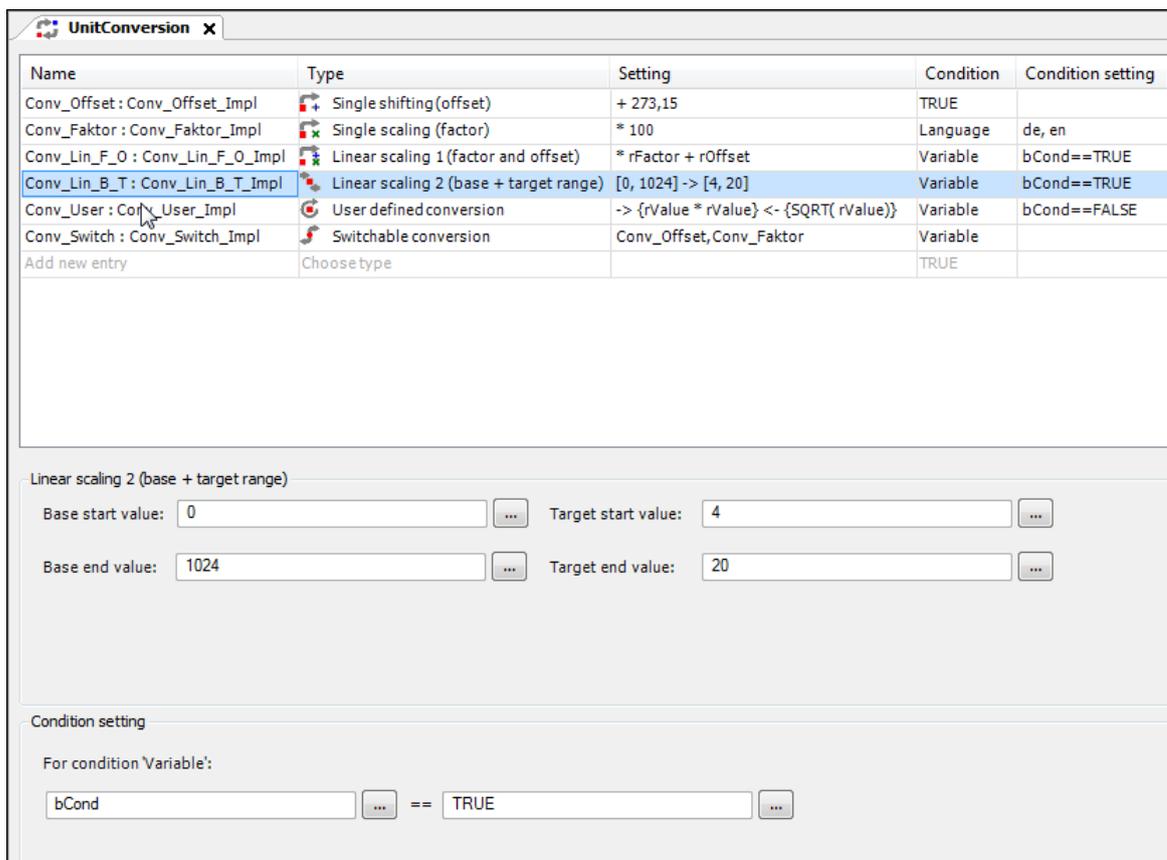


Рис. 11.3.9.12. Внешний вид компонента Unit Conversion

В столбце **Name** задается название шаблона преобразования.

В столбце **Type** выбирается тип преобразования. Доступны следующие типы:

Single Shifting (offset) – смещение значения;

Single Scaling (factor) – масштабирование (умножение) значения;

Linear Scaling 1 (factor and offset) – масштабирование + смещение значения;

Linear Scaling 2 (base + target range) – масштабирование значения из заданного начального диапазона в заданный конечный. **Обратите внимание**, что данный тип может использоваться только в случае нулевой нижней границы начального диапазона – именно поэтому он не использовался в данном примере;

User Defined Conversion – преобразование по формуле, заданной пользователем. Пользователь также может задать формулу обратного преобразования;

Switchblade Conversion – совместное использование нескольких шаблонов преобразований, переключение между которыми происходит по заданным пользователем условиям (см. ниже).

В столбце **Setting** отображаются настройки данного шаблона (коэффициент смещения, масштабирования и т.д.).

В столбце **Condition** отображается условие работы данного шаблона:

TRUE – шаблон работает всегда;

Language – шаблон работает только при выборе в проекте определенного языка (см. [п. 11.3.5](#));

Variable – шаблон работает только в том случае, если выбранная переменная принимает заданное значение.

В столбце **Condition Setting** отображаются настройки условия работы данного шаблона.

В нижней области компонента происходит задание настроек и условий работы шаблона. В большинстве случаев помимо фиксированных значений можно использовать переменные, что позволяет менять коэффициенты шаблона преобразования из кода программы.

Помимо визуализации, шаблоны можно использовать в коде программ как функции:

```
1 PROGRAM PLC_PRG
2 VAR
3   rValue:REAL;
4   rConvertedValue:REAL;
5 END_VAR
6
7 // Прямое преобразование типа CurrentLoopToTemperature
8 // (из mA в градусы Цельсия)
9
10 rConvertedValue:=CurrentLoopToTemperature.Convert(rValue);
11
12 // Обратное преобразование типа CurrentLoopToTemperature
13 // (из градусов Цельсия в mA)
14
15 rValue:=CurrentLoopToTemperature.Reverse(rConvertedValue);
```

Рис. 11.3.9.13. Использование шаблонов преобразований в коде программы

11.4. Примеры, созданные по запросам пользователей

11.4.1. CODESYS HMI

Данный пример посвящен работе с **CODESYS HMI**.

CODESYS HMI – это система исполнения визуализаций, созданных в среде программирования **CODESYS**, на персональном компьютере пользователя. Она может быть использована в случае отсутствия у целевого устройства поддержки web-визуализации. Типичные применения **CODESYS HMI** – это встроенные пульта управления станками, погрузчиками, кранами, трамваями и т.д.

Вместе с **CODESYS** распространяется версия **CODESYS HMI** с **ограничением времени** непрерывной работы. При необходимости полноценного использования, следует приобрести лицензию у компании [3S](#) (разработчик **CODESYS**).

В данном примере будет рассмотрена совместная работа панельного контроллера **СПК** и **CODESYS HMI** – нажатие кнопки на дисплее контроллера будет приводить к включению лампы в HMI-визуализации, запущенной на **ПК**.

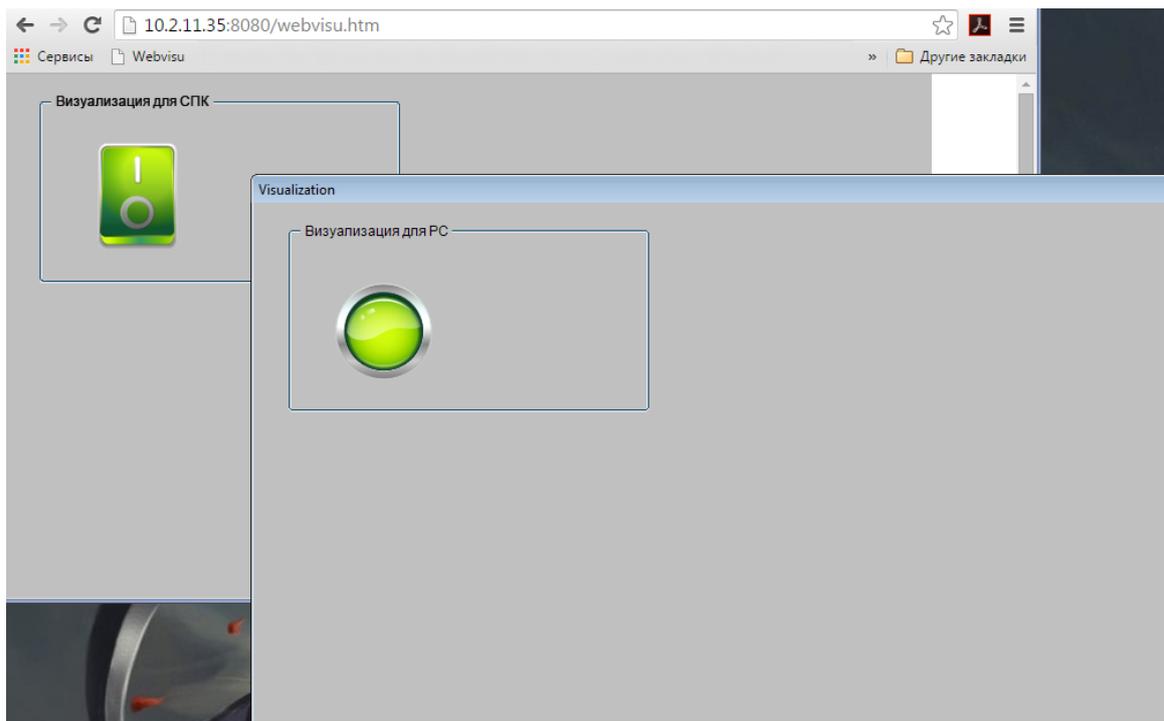


Рис. 11.4.1.1. Внешний вид примера **CODESYS HMI**

Пример создан в среде **CODESYS 3.5 SP6 Patch 2** и подразумевает запуск на СПК2xx с target-файлом **Owen SPK2xx 03.CS.WEB (3.5.4.20 [023])** при использовании на ПК **CODESYS HMI** с target-файлом **3.5.6.20**.

Пример доступен для скачивания: [Example_HMI.projectarchive](#)

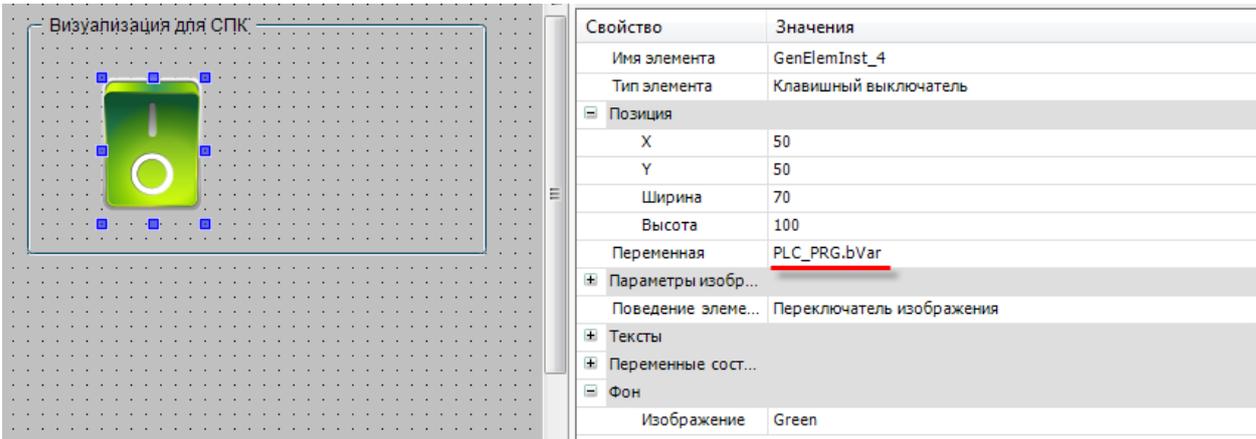
1. Создадим новый *стандартный* проект **CODESYS** с названием **Example_HMI** и следующими настройками: target – **Owen SPK2xx 03.CS.WEB (3.5.4.20 [023])**, язык PLC_PRG – **ST**.

2. В программе **PLC_PRG** объявим переменную **bVar** типа **BOOL**:

```
1 PROGRAM PLC_PRG
2 VAR
3     bVar:BOOL;
4 END_VAR
5
```

Рис. 11.4.1.2. Объявление переменных программы **PLC_PRG**

3. Добавим в проект экран визуализации **Visualization**. В его свойствах выберем размер **800x480**. Экран будет содержать элемент Клавишный выключатель, к которому привязана переменная **bVar**, и элемент Группа:

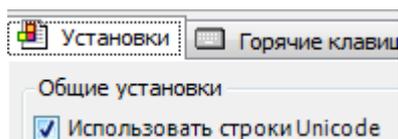


The screenshot shows a visualization editor interface. On the left, a visualization screen titled "Визуализация для СПК" contains a green button with a white circle and a vertical line. On the right, a properties table is displayed.

Свойство	Значения
Имя элемента	GenElemInst_4
Тип элемента	Клавишный выключатель
Позиция	
X	50
Y	50
Ширина	70
Высота	100
Переменная	<u>PLC_PRG.bVar</u>
Параметры изобр...	
Поведение элеме...	Переключатель изображения
Тексты	
Переменные сост...	
Фон	
Изображение	Green

Рис. 11.4.1.3. Содержимое экрана **Visualization**

4. Настроим [Менеджер визуализации](#). В его установках поставим галочку **Использовать строки Unicode**:



Настройки **target**- и **web**-визуализации будут выглядеть следующим образом:

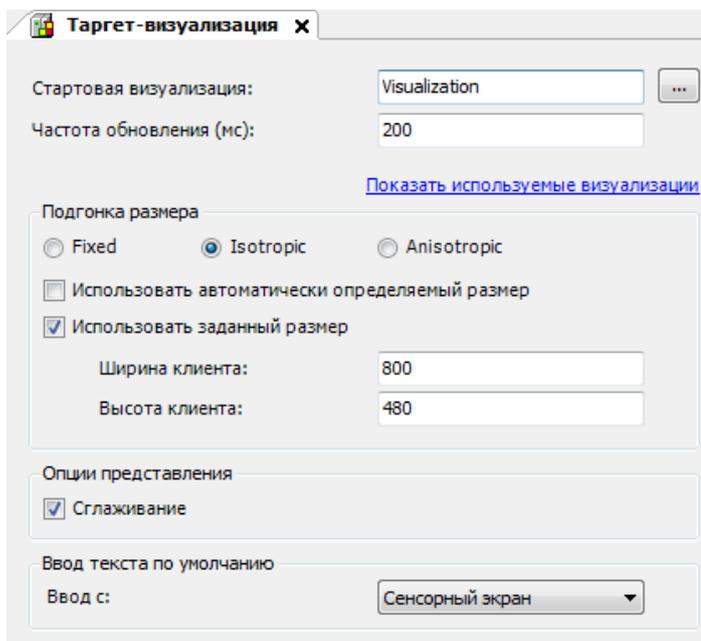


Рис. 11.4.1.4. Настройки **target**-визуализации

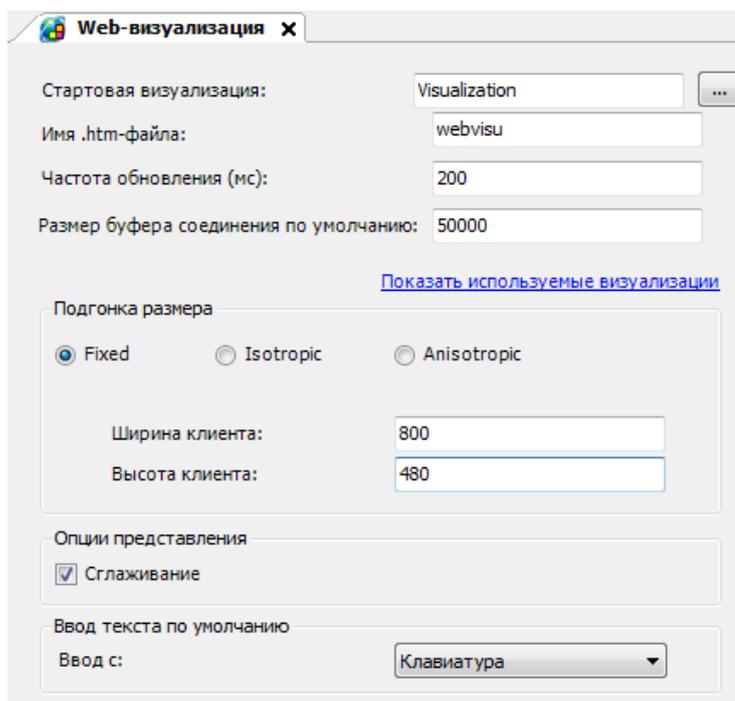


Рис. 11.4.1.5. Настройки **web**-визуализации

5. В [Менеджере визуализации](#) добавим компонент [HMI-визуализация](#):

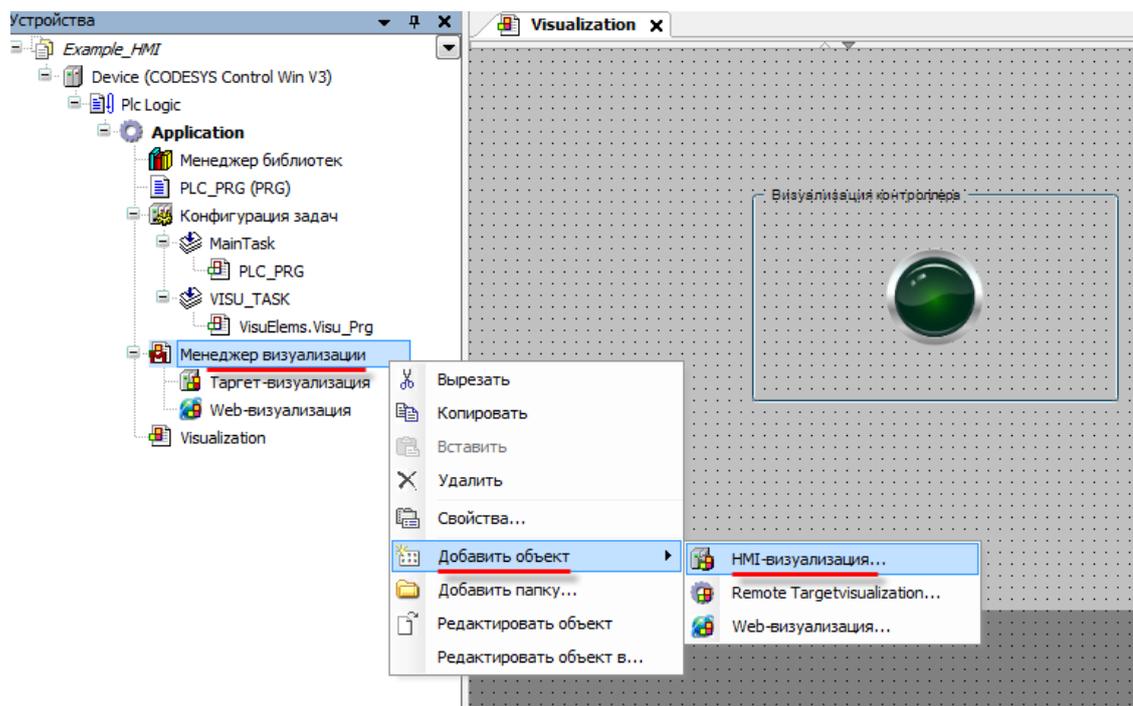


Рис. 11.4.1.6. Добавление HMI-визуализации

На вкладке **Устройство и соединение** выберем устройство **CODESYS HMI** версии **3.5.6.20**:

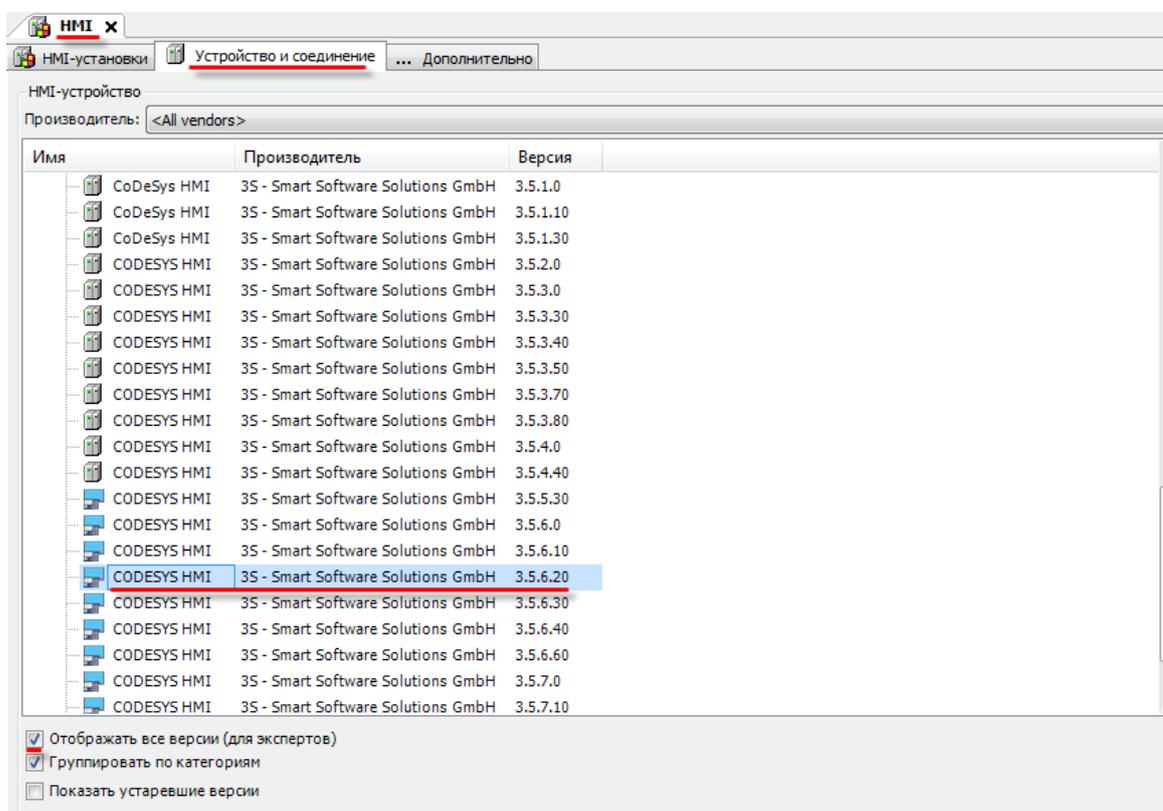


Рис. 11.4.1.7. Выбор версии компонента **CODESYS HMI**

6. Установим связь с СПК:

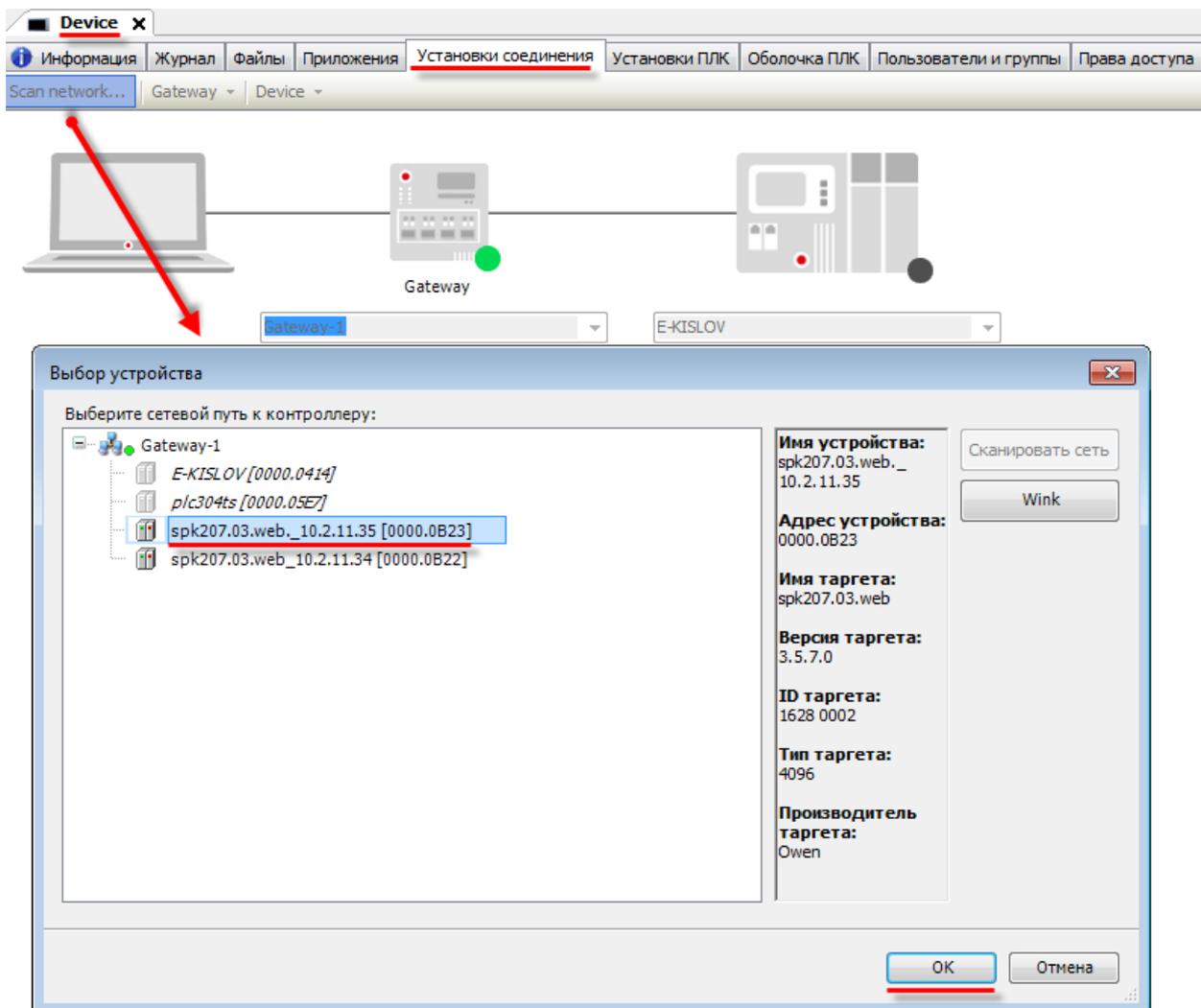


Рис. 11.4.1.8. Окно сканирования сети. Подключение к СПК

7. В компоненте **HMI** на вкладке **HMI-установки** нажмем кнопку **Создать HMI**. В появившемся диалоговом окне нажмем **ОК**.

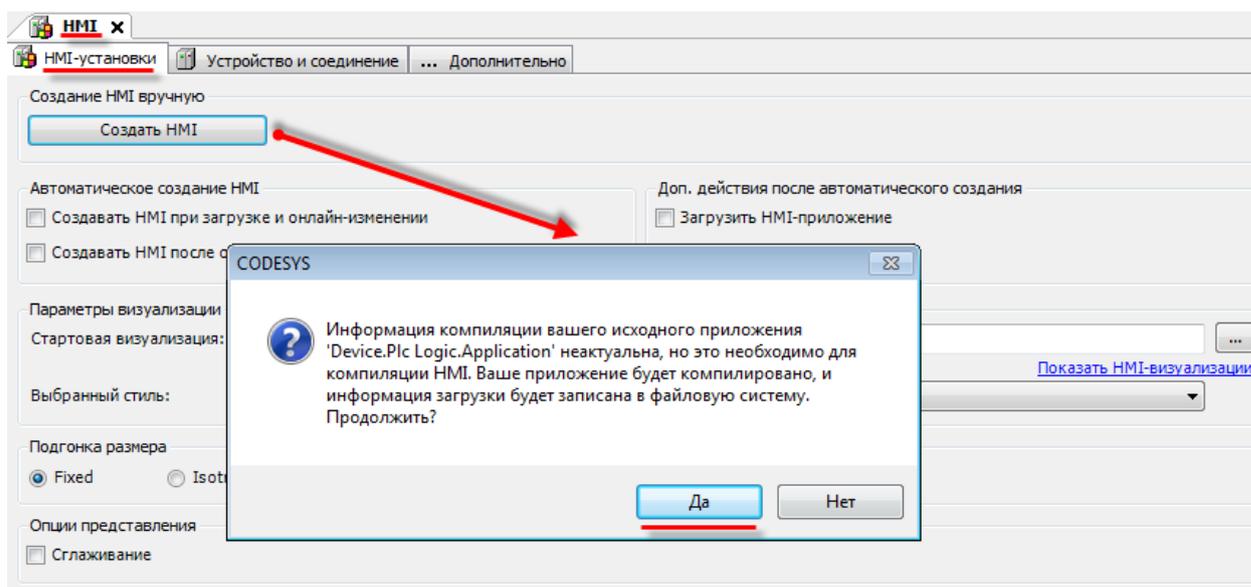


Рис. 11.4.1.9. Создание HMI-визуализации в проекте

Это приведет к добавлению в проект устройства **HMI**:

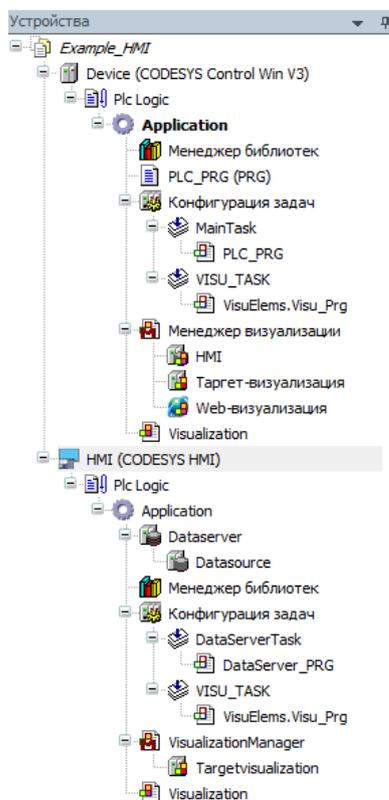


Рис. 11.4.1.10. Панель устройств после добавления в проект HMI-визуализации

Откроем экран **Visualization** устройства **CODESYS HMI**. По умолчанию он полностью идентичен экрану **Visualization** исходного устройства (СПК):

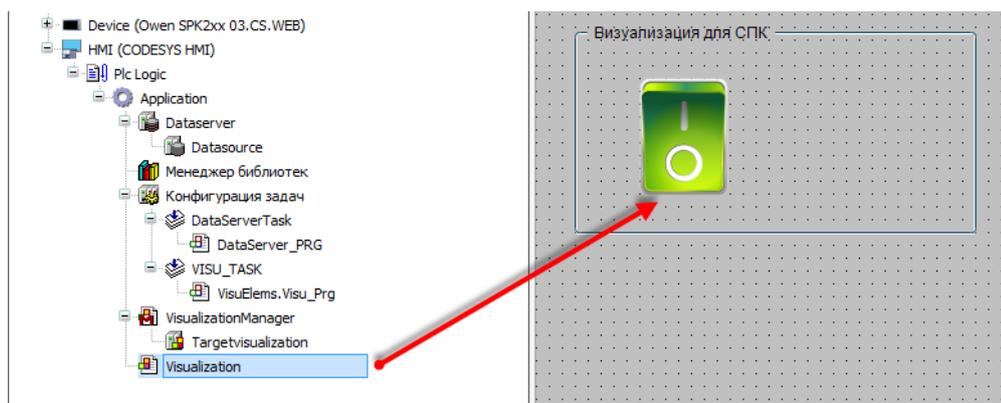


Рис. 11.4.1.11. Внешний вид экрана **Visualization** устройства **HMI**

Изменим надпись элемента **Группа** и заменим выключатель на элемент **Индикатор**, к которому привяжем переменную **bVar**. В появившемся при этом диалоговом окне нажмем **ОК**:

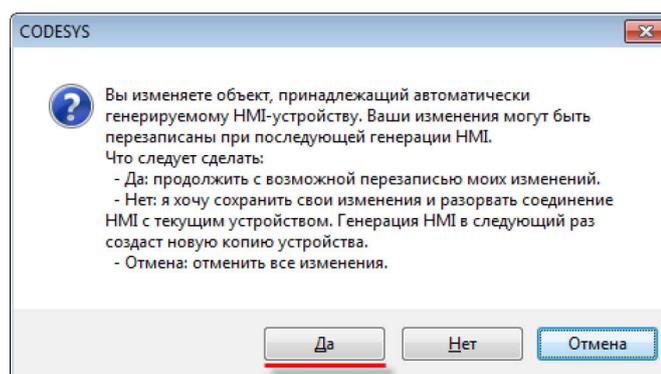


Рис. 11.4.1.12. Диалоговое окно изменения HMI-визуализации

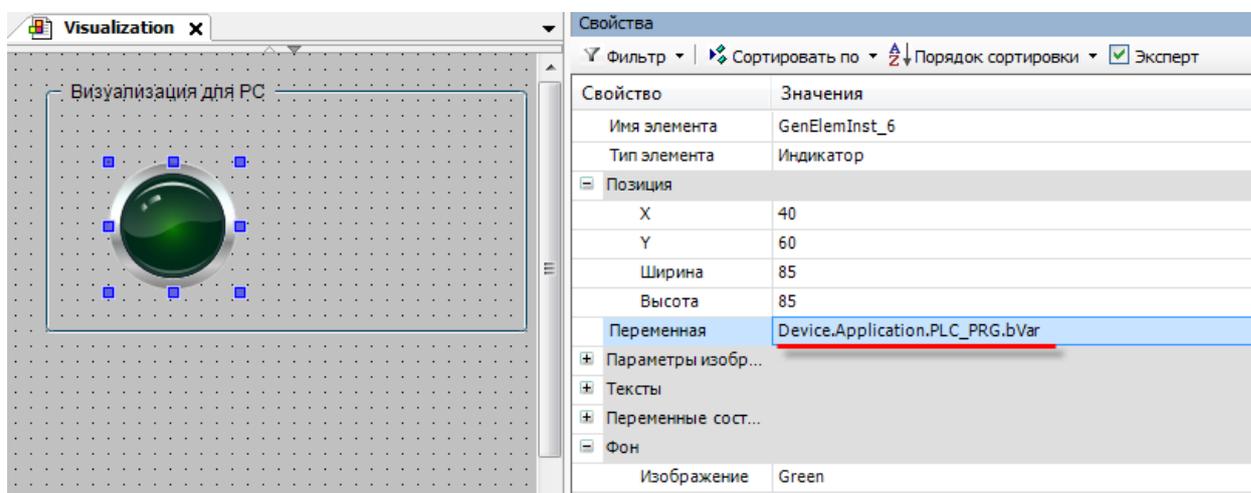


Рис. 11.4.1.13. Внешний вид экрана **Visualization** устройства **HMI** после доработки

8. Загрузим проект для СПК в контроллер с помощью команды **Логин** и запустим его:

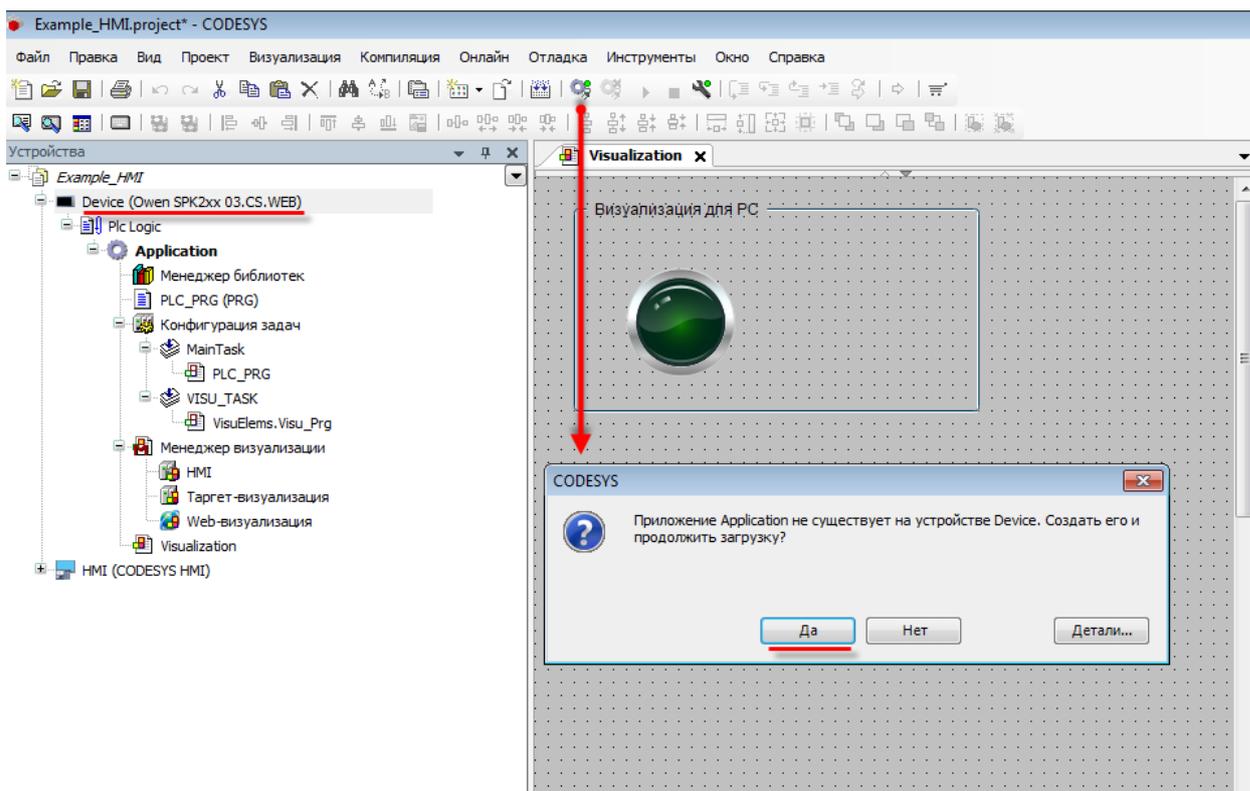


Рис. 11.4.1.14. Загрузка проекта в СПК

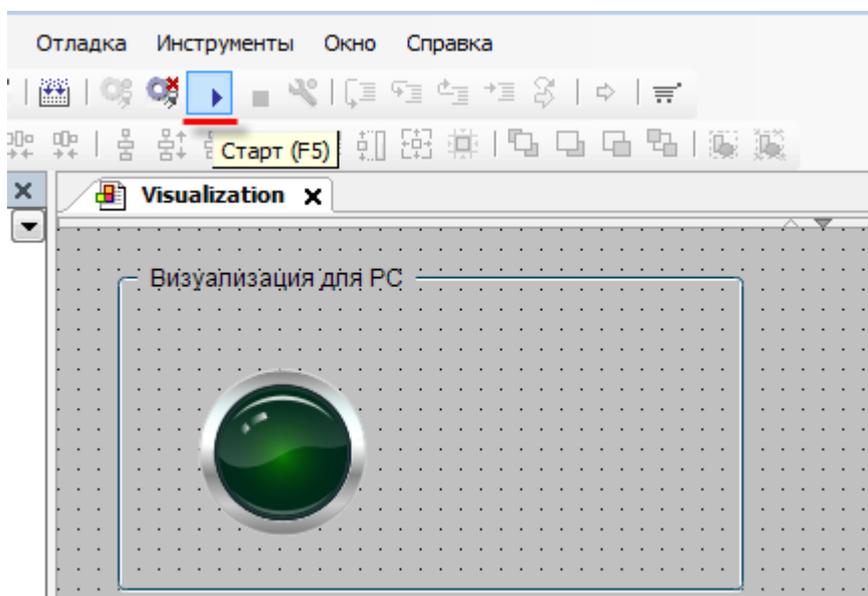


Рис. 11.4.1.15. Запуск приложения

Запустим приложение **CODESYS HMI**. Это можно сделать из **меню Пуск** или непосредственно из папки **GatewayPLC**, расположенной в папке установки **CODESYS**:

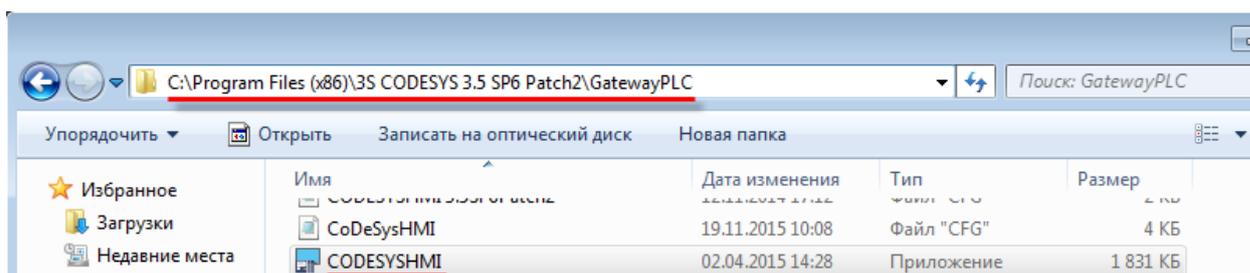


Рис. 11.4.1.16. Запуск приложения **CODESYS HMI**

На панели задач Windows появится соответствующая пиктограмма:

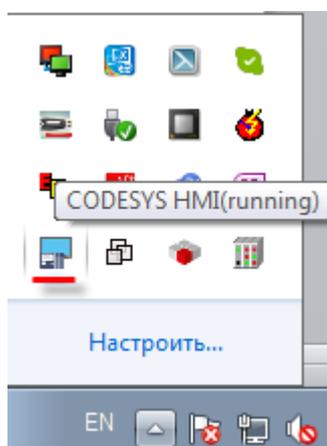


Рис. 11.4.1.17. Пиктограмма приложения **CODESYS HMI**

Для **CODSEYS HMI** произведем сканирование сети и выберем устройство, чье имя совпадает с именем ПК:

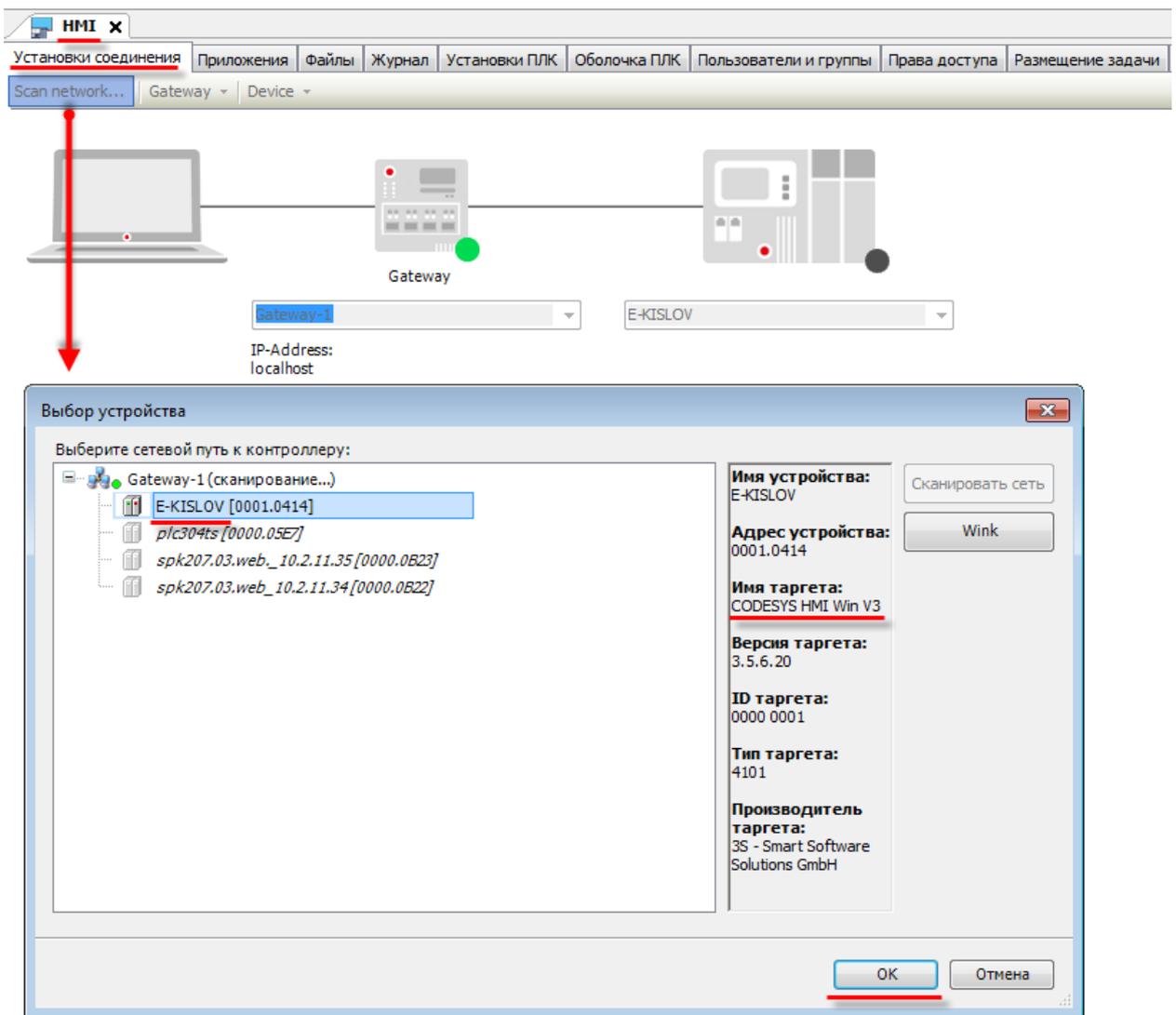


Рис. 11.4.1.18. Окно сканирования сети

Нажмем **ПКМ** на приложении **Application** устройства **HMI** и выполним команду **Логин**:

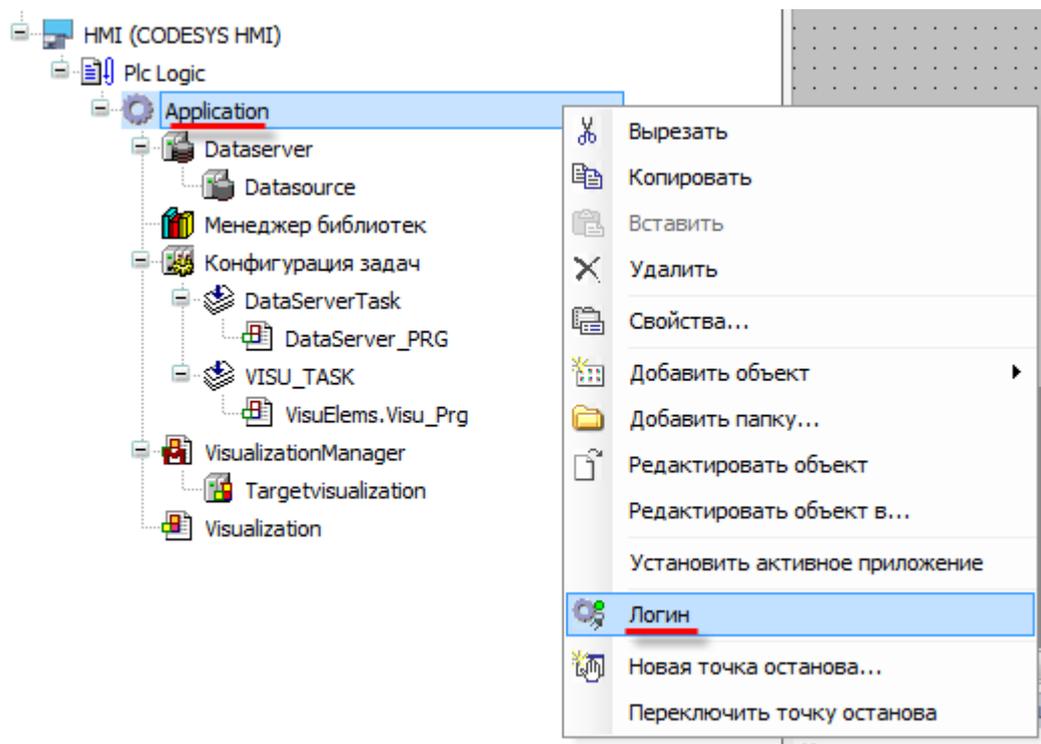


Рис. 11.4.1.19. Окно сканирования сети

Запустим приложение с помощью команды **Старт**:

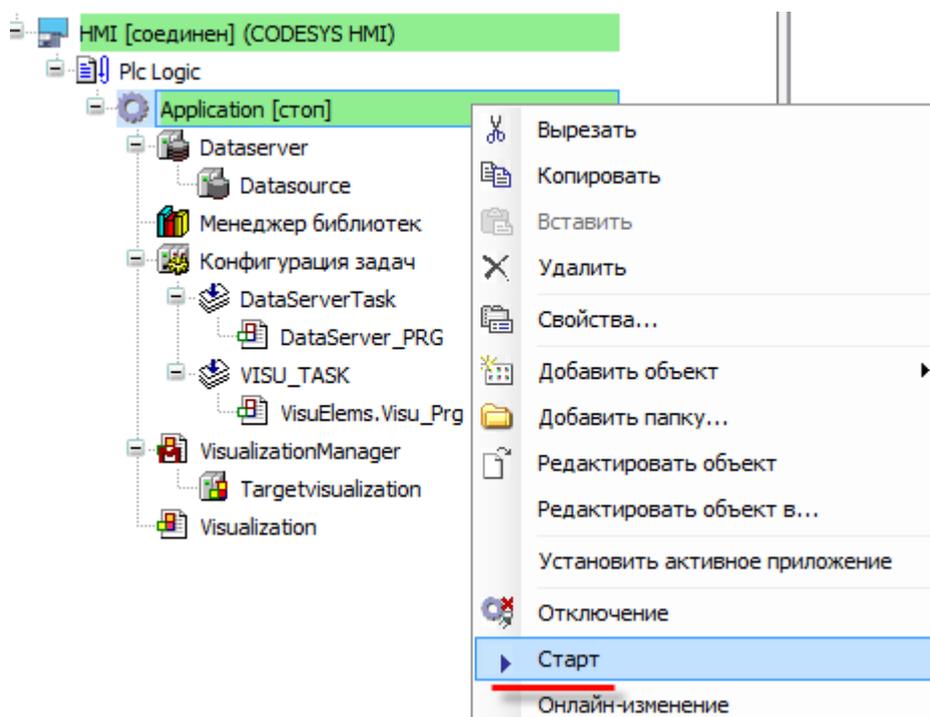


Рис. 11.4.1.20. Запуск приложения

При этом в программе **CODESYS HMI** будет автоматически открыт экран визуализации **Visualization**:

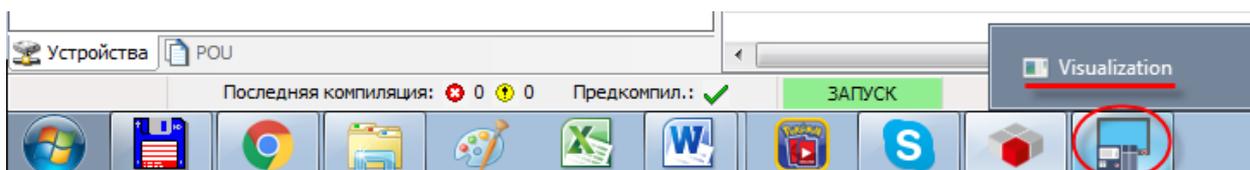


Рис. 11.4.1.21. Программа **CODESYS HMI** с открытым экраном **Visualization**

9. Меняя положение выключателя с дисплея СПК (или через web-визуализацию), наблюдайте за включением/отключением индикатора в HMI-визуализации на ПК. **Обратите внимание**, что по умолчанию HMI-визуализация занимает весь экран; настройки ее размера описаны ниже.

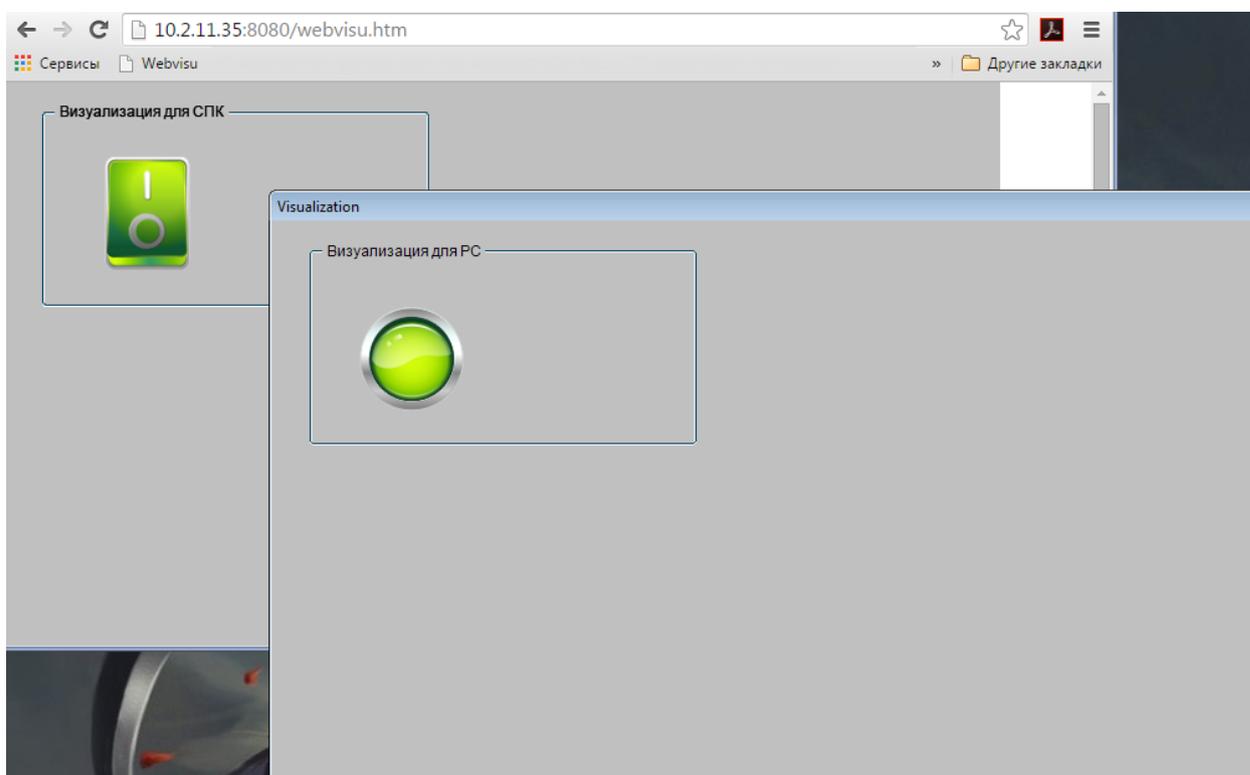


Рис. 11.4.1.22. Проект **Example_HMI** в процессе работы. На ПК одновременно запущены web-визуализация контроллера и HMI-визуализация

10. Как изменить разрешение HMI-визуализации?

По умолчанию, HMI-визуализация открывается в режиме **FullScreen**, занимая весь дисплей ПК. Чтобы настроить разрешение HMI-визуализации, откройте файл **CODESYSHMI.cfg**, расположенный в папке **GatewayPLC** (она находится в папке установки **CODESYS**) и добавьте в него следующие строки:

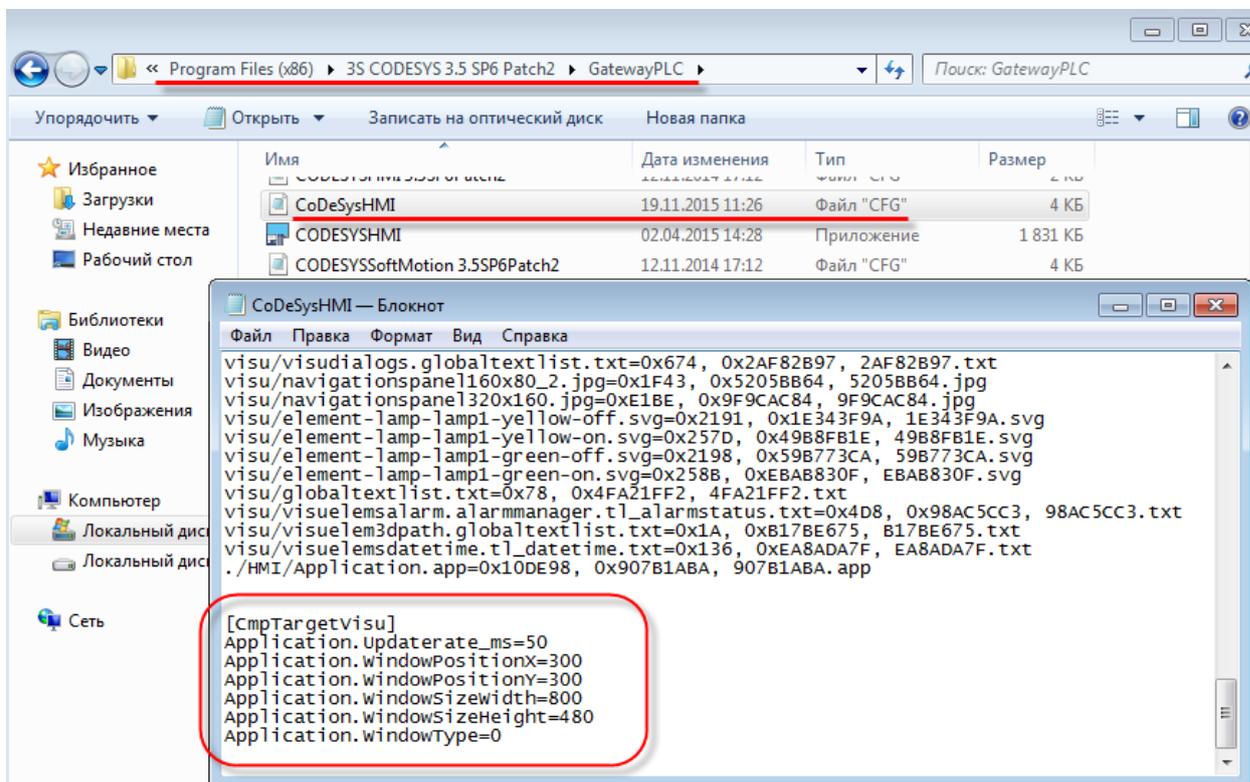


Рис. 11.4.1.23. Редактирование файла **CODESYSHMI.cfg**

Updaterate_ms – частота обновления HMI-визуализации в миллисекундах;

WindowPositionX и **WindowPositionY** – координаты открытия окна HMI-визуализации в пикселях по оси X и Y соответственно (координаты левой верхней точки);

WindowSizeWidth – размер окна HMI-визуализации по горизонтали в пикселях;

WindowSizeHeight – размер окна HMI-визуализации по вертикали в пикселях;

WindowType – тип открытия окна: **0** – с рамкой и заголовком, **1** – с рамкой и без заголовка, **2** – без рамки и без заголовка.

11. Как открывать и закрывать HMI-визуализацию при помощи ярлыков?

В нашем примере запуск окна HMI-визуализации был произведен при старте приложения в **CODESYS**; кроме того, закрыть его можно только с помощью команды **Закреть окно** в Windows (или комбинацией клавиш **Alt+F4**). Очевидно, что в большинстве случаев требуется более традиционный механизм взаимодействия с приложением – например, открытие с помощью ярлыка и закрытие при нажатии на кнопку.

Для закрытия окна HMI добавьте на экран визуализации любой элемент, содержащий вкладку **InputConfiguration** (например, [Кнопка](#)), и к параметру **OnMouseClicked** привяжите действие [Выполнить команду](#) с 1-ым параметром

```
"C:\Program Files (x86)\3S CODESYS 3.5 SP6 Patch2\GatewayPLC\VisualClientController.exe" --application=Application --flags=2'
```

2-й параметр оставьте пустым. **Обратите внимание**, что путь к файлу **VisualClientController.exe** может отличаться в зависимости от папки установки **CODESYS**, а имя приложения (**Application**) должно совпадать с именем приложения в проекте **CODESYS**.

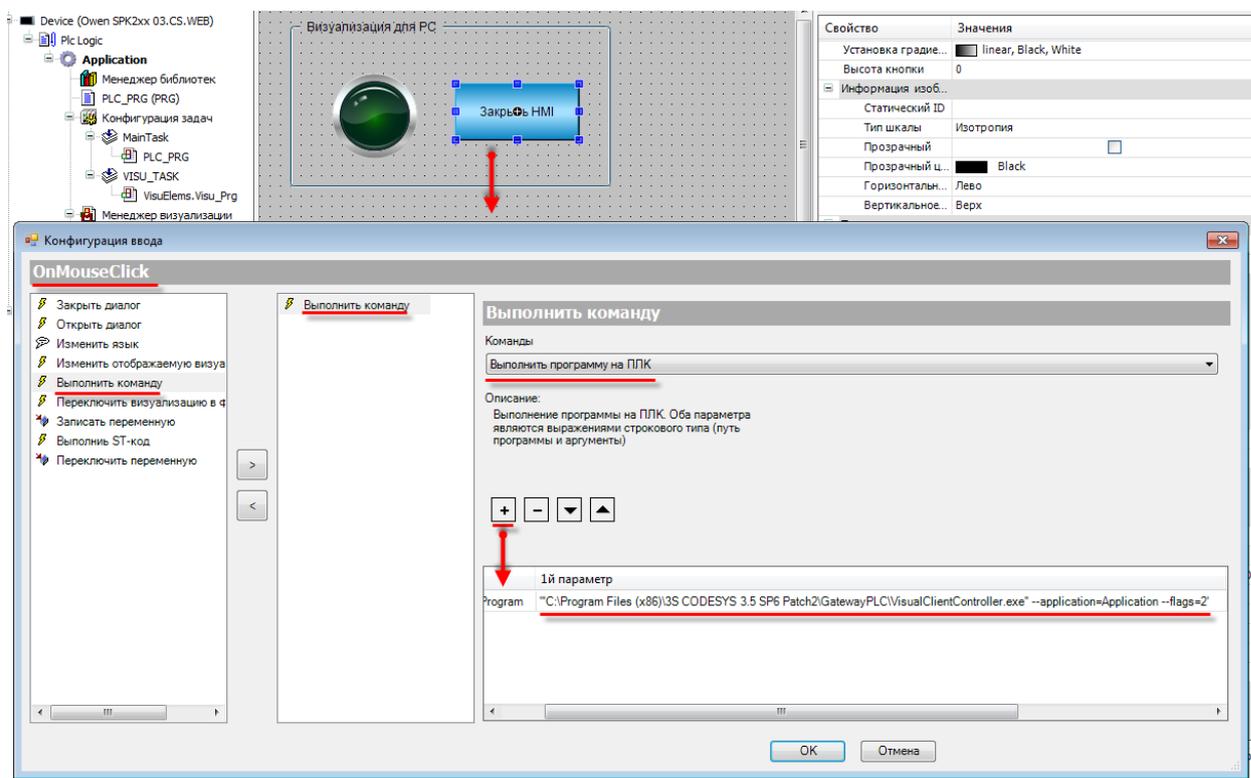


Рис. 11.4.1.24. Привязка к элементу команды закрытия окна HMI-визуализации

Для запуска окна HMI-визуализации создайте на рабочем столе ярлык, указав для него следующий путь:

"C:\Program Files (x86)\3S CODESYS 3.5 SP6 Patch2\GatewayPLC\VisualClientController.exe" --application=Application --flags=1

Обратите внимание, что путь к файлу **VisualClientController.exe** может отличаться в зависимости от папки установки **CODESYS**, а имя приложения (**Application**) должно совпадать с именем приложения в проекте **CODESYS**.

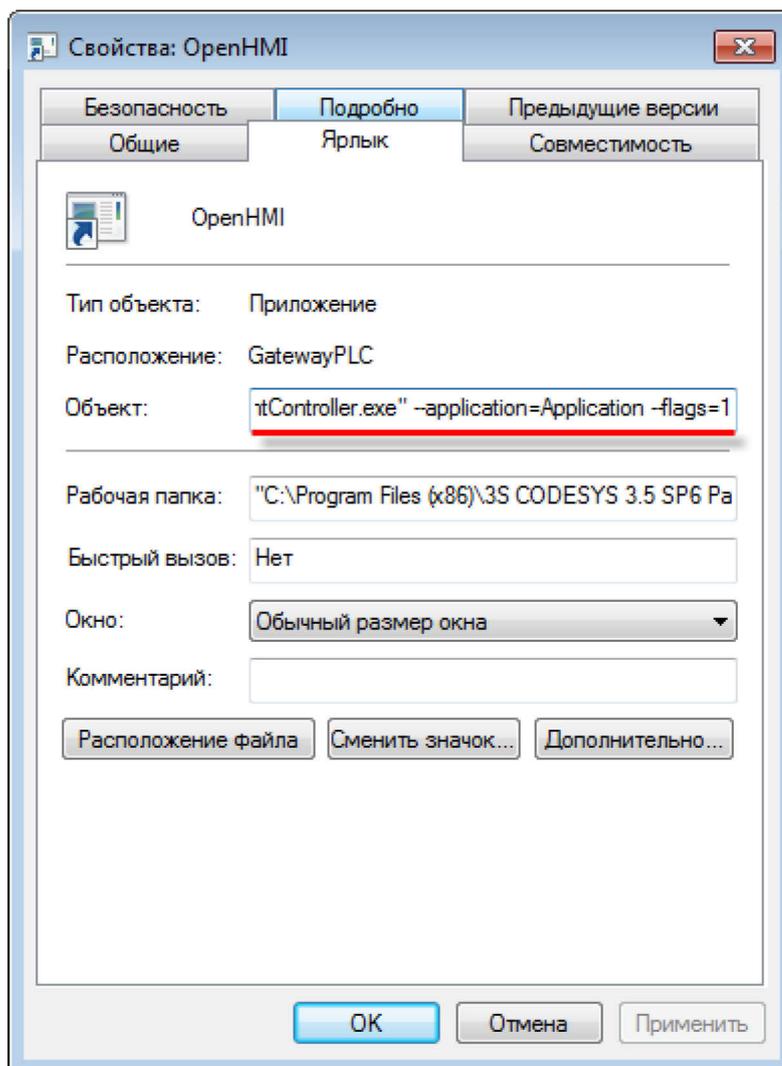


Рис. 11.4.1.25. Создание ярлыка для запуска окна HMI-визуализации

Обратите внимание, что ярлык будет активен только при запущенном приложении **CODESYS HMI** (см. рис. 11.4.1.16).

Если приведенный выше способ не работает (или срабатывает не всегда), то можно перезапустить приложение **CODESYS HMI** – по умолчанию оно откроется со стартовым экраном последнего загруженного проекта, при этом на восстановление связи с СПК/ПЛК уйдет около 10 секунд.

11.4.2. Работа с динамическими точками (элемент Ломаная)

Данный пример посвящен работе с динамическими точками. С помощью элементов, которые содержат одноименную вкладку параметров ([Полигон](#), [Ломаная](#), [Кривая Безье](#)), пользователь может рисовать свои геометрические фигуры в визуализации в процессе работы проекта.



Рис. 11.4.2.1. Внешний вид примера **Ломаная**

Пример создан в среде **CODESYS 3.5 SP6 Patch 2** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с target-файлом **3.5.6.0**.

Пример доступен для скачивания: [Example_Polyline.projectarchive](#)

1. Создадим новый **стандартный** проект **CODESYS** с названием **Example_Polyline** и настройками по умолчанию: target – **CODESYS Control Win V3**, язык **PLC_PRG – ST**.

2. В **Менеджере библиотек** добавим библиотеку **CmpVisuHandler** версии **3.5.6.0** (она понадобится нам, чтобы работать с динамическими точками):

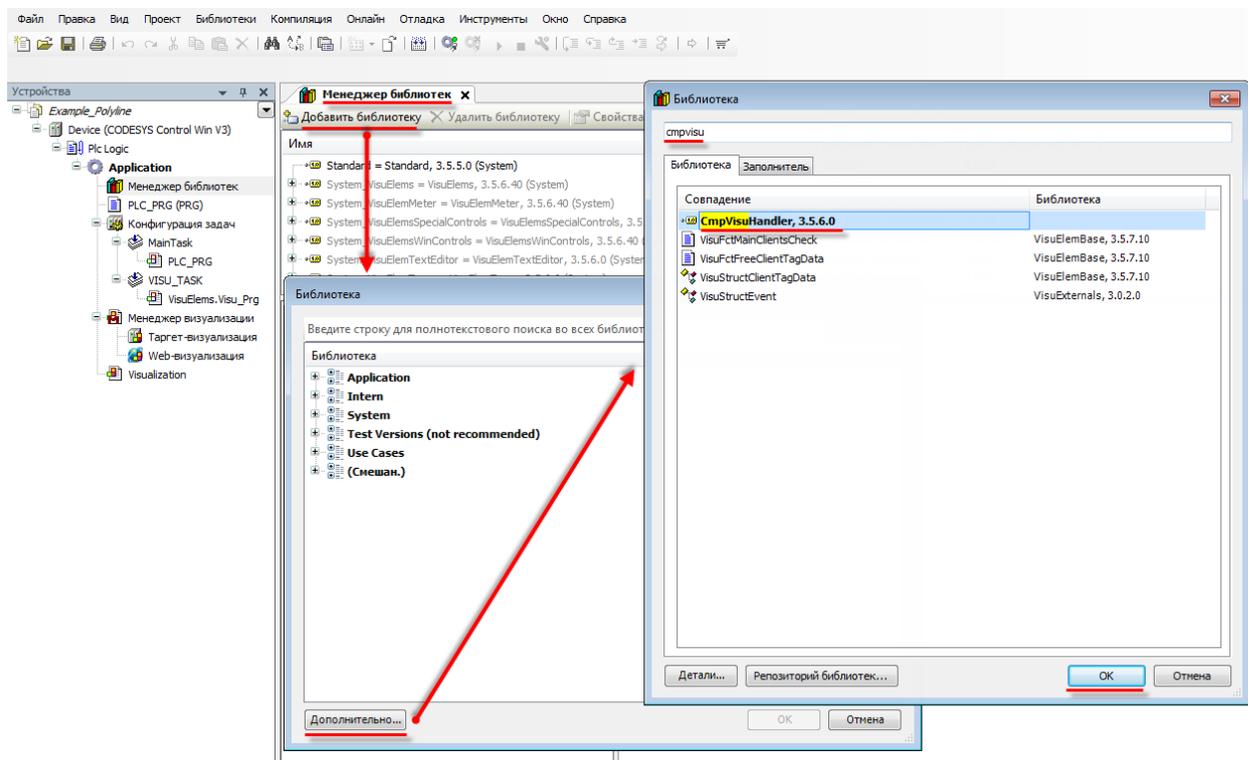


Рис. 11.4.2.2. Добавление библиотеки **CmpVisuHandler**

3. В программе **PLC_PRG** объявим следующие переменные:

```
1 PROGRAM PLC_PRG
2 VAR
3   arrPolylineCoordinates: ARRAY [1..100] OF VisuStructPoint; // массив координат точек ломаной
4   iPolylinePoints: INT:=2; // кол-во точек ломаной (по умолчанию - две)
5   iIndex: INT:=1; // индекс выбранной точки
6 END_VAR
```

Рис. 11.4.2.3. Объявление переменных программы **PLC_PRG**

4. [Добавим в проект](#) экран визуализации **Visualization**. В его [свойствах](#) выберем размер **800x480**. Экран будет содержать элемент [Ломаная](#), элемент [Группа](#), два элемента [Управление вращением](#) и два элемента [Полоса прокрутки](#).

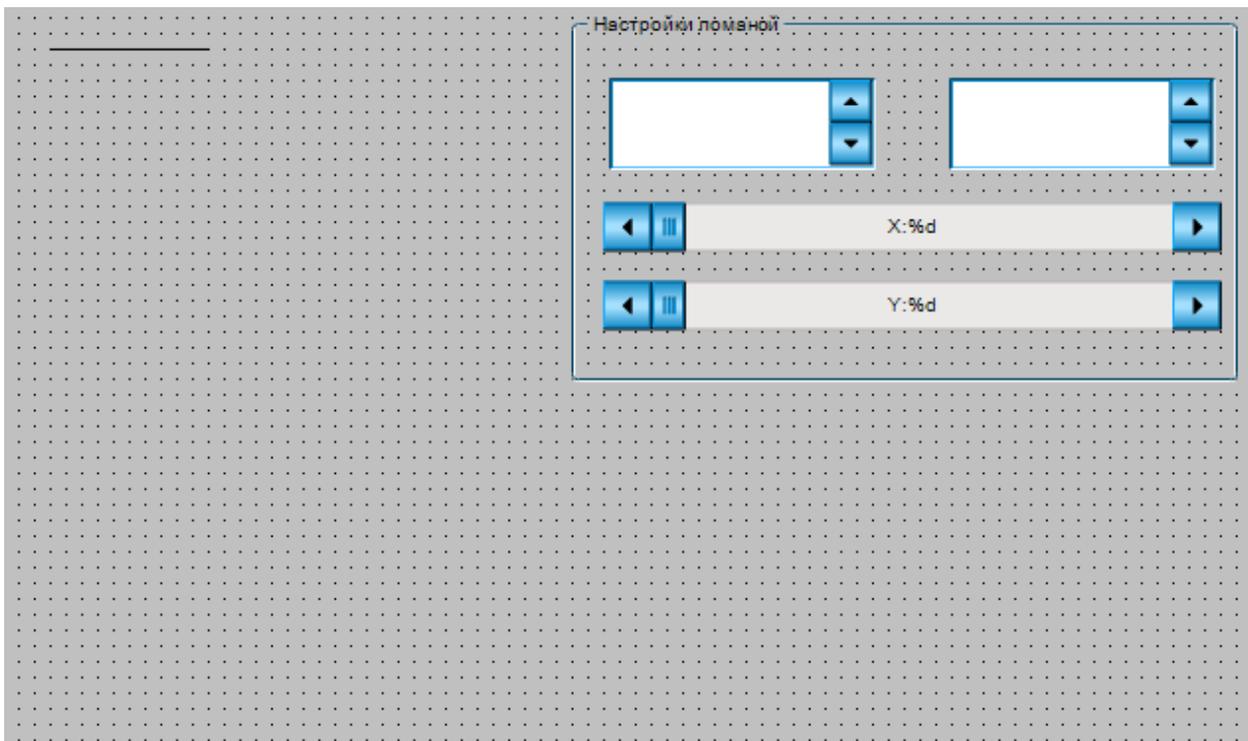


Рис. 11.4.2.4. Содержимое экрана визуализации **Visualization**

Настройки элементов приведены ниже.

5. Настроим элемент [Ломаная](#). **Обратите внимание**, что к параметру **Массив точек** привязывается не переменная типа **ARRAY**, а ее [адрес](#).

Динамические точки	
Массив точек	ADR(PLC_PRG.arrPolylineCoordinates)
Число точек	PLC_PRG.iPolylinePoints

Рис. 11.4.2.5. Настройки элемента **Ломаная**

6. Настроим элементы [Управление вращением](#). Один из этих элементов будет использоваться для выбора точки ломаной, координаты которой будут в данный момент меняться пользователем, другой - для настройки количества точек ломаной.

Свойство	Значения
Имя элемента	GenElemInst_26
Тип элемента	Управление вращением
[-] Позиция	
X	25
Y	50
Ширина	165
Высота	55
Переменная	PLC_PRG.iIndex
Числовой формат	Выбрана точка: %d
Интервал	1
[-] Value range	
Минимальное значе...	1
Максимальное значе...	PLC_PRG.iPolylinePoints

Рис. 11.4.2.6. Настройки элемента **Управление вращением 1**

Свойство	Значения
Имя элемента	GenElemInst_28
Тип элемента	Управление вращением
[-] Позиция	
X	240
Y	50
Ширина	165
Высота	55
Переменная	PLC_PRG.iPolylinePoints
Числовой формат	Число точек: %d
Интервал	1
[-] Value range	
Минимальное значе...	1
Максимальное значе...	

Рис. 11.4.2.7. Настройки элемента **Управление вращением 2**

7. Настроим элементы [Полоса прокрутки](#). Эти элементы будут использоваться для изменения координат выбранной точки ломаной.

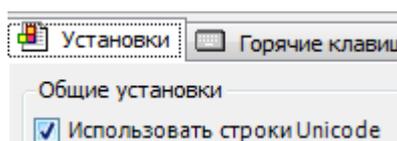
Свойство	Значения
Имя элемента	GenElemInst_30
Тип элемента	Полоса прокрутки
ID текста	5
Значение	PLC_PRG.arrPolylineCoordinates[PLC_PRG.iIndex].iX
Минимальное значение	0
Максимальное значение	300
Размер страницы	
Прокрутка выполнена	<input type="checkbox"/>
[-] Позиция	
X	20
Y	128
Ширина	390
Высота	30
[+] Линейка	
[+] Цвета	
[-] Тексты	
Текст	X:%d
Подсказка	

Рис. 11.4.2.8. Настройки элемента **Полоса прокрутки 1**

Свойства	
Ƴ Фильтр ⚡ Сортировать по 2/2 Порядок сортировки <input checked="" type="checkbox"/> Эксперт	
Свойство	Значения
Имя элемента	GenElemInst_32
Тип элемента	Полоса прокрутки
ID текста	6
Значение	PLC_PRG.arrPolylineCoordinates[PLC_PRG.iIndex].iY
Минимальное значение	0
Максимальное значение	800
Размер страницы	
Прокрутка выполнена	<input type="checkbox"/>
[-] Позиция	
X	20
Y	178
Ширина	390
Высота	30
[+] Линейка	
[+] Цвета	
[-] Тексты	
Текст	Y:%d
Подсказка	

Рис. 11.4.2.9. Настройки элемента **Полоса прокрутки 2**

8. Настроим компонент [Менеджер визуализации](#). В его установках поставим галочку **Использовать строки Unicode**:



Настройки **target**- и **web**-визуализации будут выглядеть следующим образом:

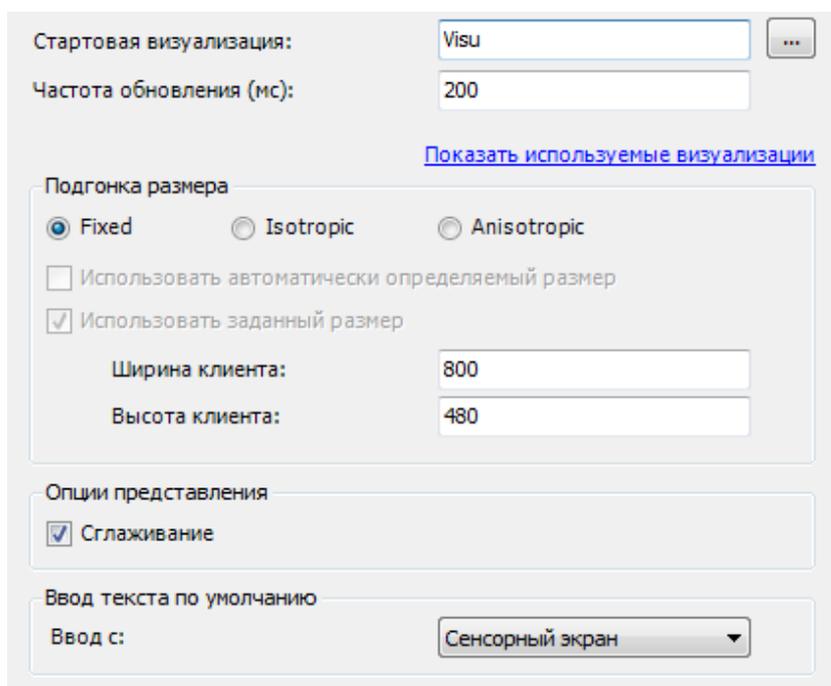


Рис. 11.4.2.10. Настройки **target**-визуализации

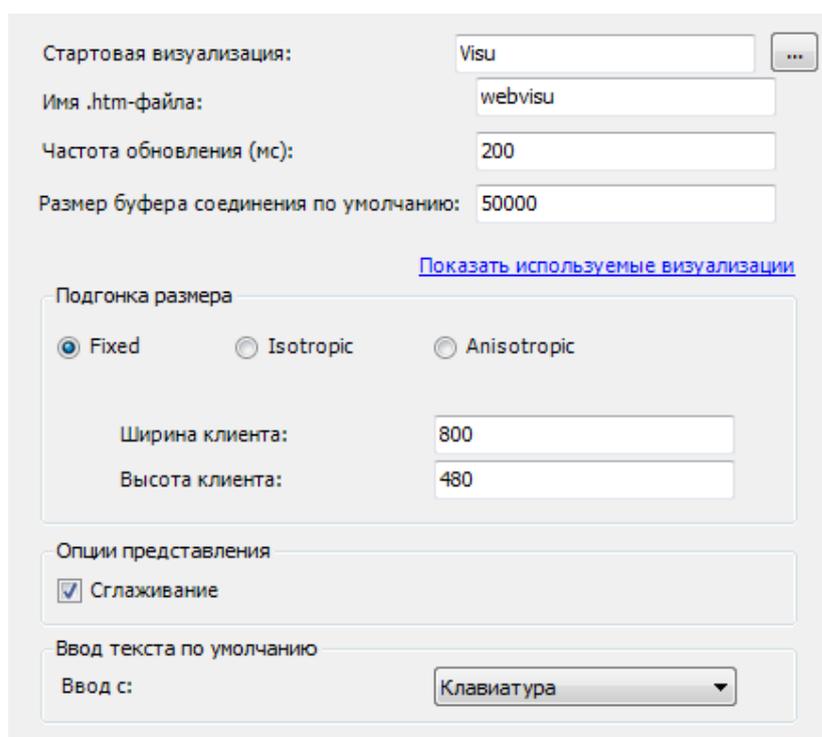


Рис. 11.4.2.11. Настройки **web**-визуализации

9. [Запустим проект на виртуальном контроллере](#). Откроем в веб-браузере страницу <http://localhost:8080/webvisu.htm>, чтобы увидеть web-визуализацию проекта. **Обратите внимание**, что браузер должен поддерживать **HTML5**. Проверим функционал проекта.

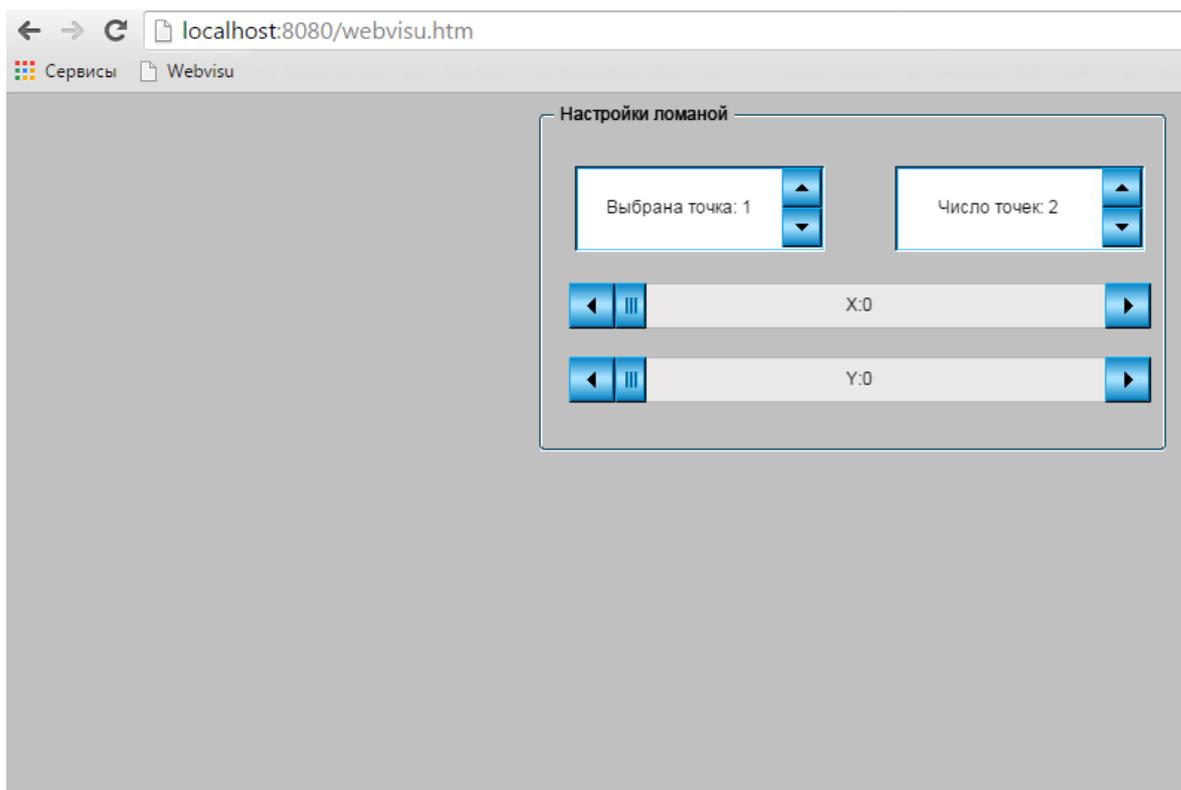


Рис. 11.4.2.12. Web-визуализация проекта в браузере **Chrome**

Зададим число точек фигуры (например, **10**). С помощью полос прокрутки выберем координаты первой точки (например, **100,100**). Выберем вторую точку, после чего зададим ее координаты. Каждая последующая точка будет автоматически соединяться с предыдущей. В результате, по заданным пользователем координатам будет построена некая геометрическая фигура:

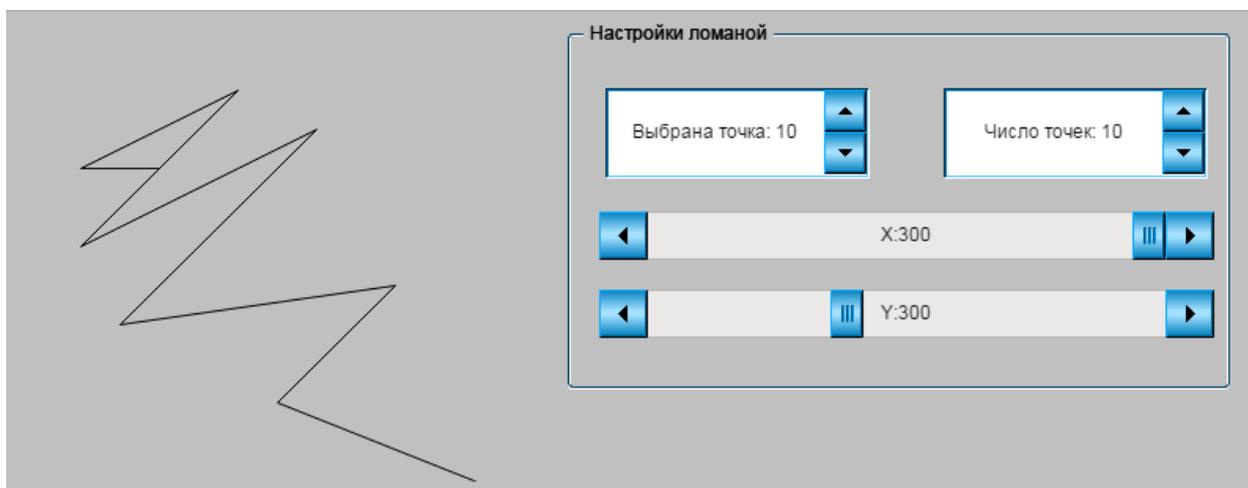


Рис. 11.4.2.13. Построение фигуры по координатам точек

11.4.3. Считывание координат курсора

Данный пример посвящен работе с курсором в визуализации **CODESYS**. Библиотека **SysTargetVisu23** позволяет получить данные о количестве и координатах последнего нажатия, отпускания и перемещения курсора. В примере пользователь будет перемещать элемент с помощью движения курсора.

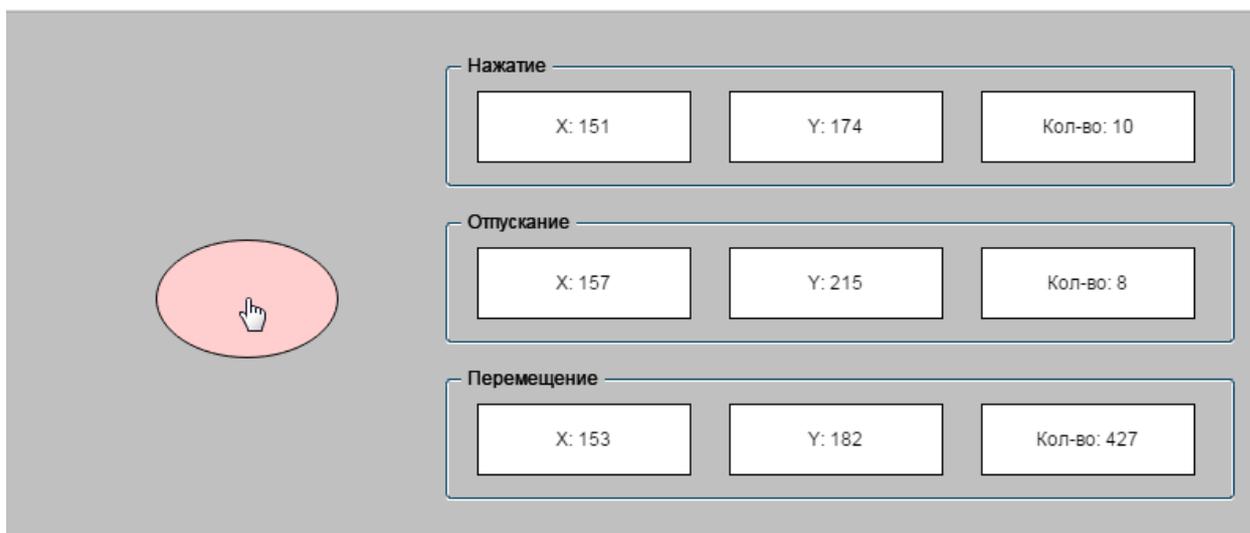


Рис. 11.4.3.1. Внешний вид примера **Считывание координат курсора**

Пример создан в среде **CODESYS 3.5 SP6 Patch 2** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с target-файлом **3.5.6.0**.

Пример доступен для скачивания: [Example_Cursor.projectarchive](#)

1. Создадим новый **стандартный** проект **CODESYS** с названием **Example_Cursor** и настройками по умолчанию: target – **CODESYS Control Win V3**, язык PLC_PRG – **ST**.

2. В **Менеджере библиотек** добавим библиотеку **SysTargetVisu23** версии **3.5.6.0** (она понадобится нам, чтобы считывать информацию о курсоре):

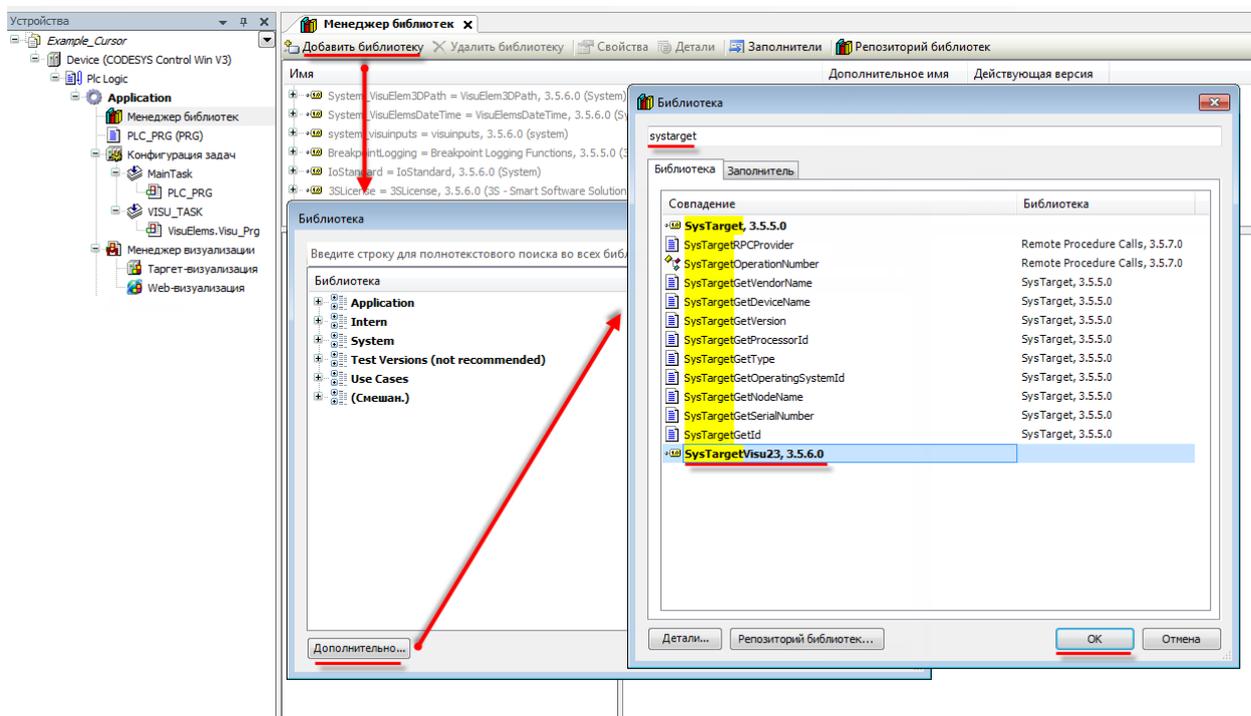


Рис. 11.4.3.2. Добавление библиотеки **SysTargetVisu23**

3. В программе **PLC_PRG** объявим следующие переменные:

```
1 PROGRAM PLC_PRG
2 VAR
3     MouseDownEvent:MouseEvent; // структура данных нажатия курсора
4     MouseUpEvent:MouseEvent; // структура данных отпущения курсора
5     MouseMoveEvent:MouseEvent; // структура данных перемещения курсора
6
7     bSelect:BOOL; // переменная выбора элемента
8     iCurrentX:INT:=100; // начальная координата элемента по горизонтали
9     iCurrentY:INT:=80; // начальная координата элемента по вертикали
10 END_VAR
```

Рис. 11.4.3.3. Объявление переменных программы **PLC_PRG**

4. Код программы **PLC_PRG** будет выглядеть следующим образом:

```
1 IF bSelect=TRUE THEN // если пользователь нажал на элемент...
2   GetLastMouseDownEvent (ADR(MouseDownEvent)); // ...то начинаем считывать информацию о курсоре...
3   GetLastMouseUpEvent (ADR(MouseUpEvent));
4   GetLastMouseMoveEvent (ADR(MouseMoveEvent));
5 // ...и присваиваем элементу координаты курсора
6   iCurrentX:=MouseMoveEvent.nXPos-30-115/2; // 30 - Позиция/X, 115 - Позиция/Ширина
7   iCurrentY:=MouseMoveEvent.nYPos-30-75/2; // 30 - Позиция/Y, 75 - Позиция/Высота
8 END_IF
9
```

Рис. 11.4.3.4. Код программы **PLC_PRG**

5. [Добавим в проект](#) экран визуализации **Visualization**. В его [свойствах](#) выберем размер **800x480**. Экран будет содержать элемент [Эллипс](#), три элемента [Группа](#) и девять элементов [Прямоугольник](#).

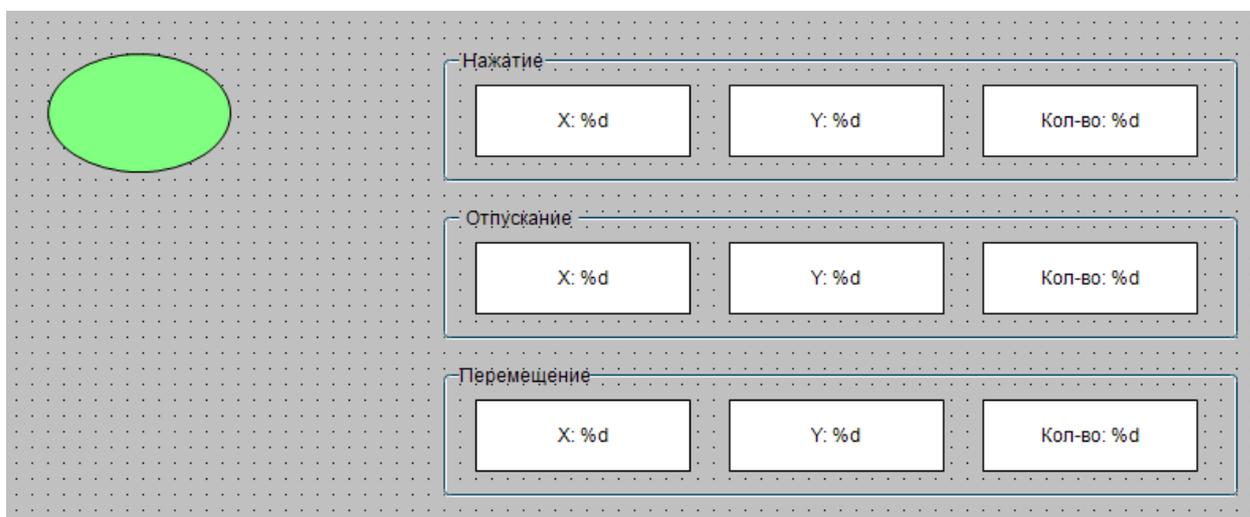


Рис. 11.4.3.4. Содержимое экрана визуализации **Visualization**

Настройки элементов приведены ниже.

6. Настроим элемент Эллипс. После нажатия и отпущания (клика) пользователем, элемент будет выделяться (менять цвет), после чего пользователь сможет перемещать элемент с помощью движения курсора. Еще один клик отпустит элемент, после чего он будет зафиксирован на месте положения курсора.

[-] Позиция	
X	30
Y	30
Ширина	115
Высота	75
[+] Центр	
[-] Цвета	
[-] Нормальное состояние	
[+] Цвет фрейма	BasicElement-Frame-Color
[+] Цвет заливки	Lightgreen
[-] Состояние тревоги	
[+] Цвет фрейма	Black
[+] Цвет заливки	BasicElement-Alarm-Fill-Color
Использовать градиентный цвет	<input type="checkbox"/>
Установка градиента	linear, Black, White
[+] Вид элемента	
[+] Тексты	
[+] Свойства текста	
[-] Абсолютное перемещение	
[-] Перемещение	
X	PLC_PRG.iCurrentX
Y	PLC_PRG.iCurrentY
Вращение	
Масштабирование	
Внутреннее вращение	
[+] Относительное перемещение	
[+] Текстовые переменные	
[+] Динамические тексты	
[+] Переменные шрифта	
[-] Переменные цвета	
Переключить цвет	PLC_PRG.bSelect
[+] Нормальное состояние	
[+] Состояние тревоги	
[+] Переменные вида	
[-] Переменные состояний	
Невидимый	
Отключение ввода	
[-] Inputconfiguration	
OnDialogClosed	Конфигурация...
OnMouseClicked	Конфигурация...
OnMouseDown	Конфигурация...
OnMouseEnter	Конфигурация...
OnMouseLeave	Конфигурация...
OnMouseMove	Конфигурация...
OnMouseUp	Конфигурация...
[+] Нажать	
[-] Переключить	
Переменная	PLC_PRG.bSelect
Переключить вверх, если наж...	<input type="checkbox"/>

Рис. 11.4.3.5. Настройки элемента Эллипс

7. Настроим элементы вкладки **Нажатие**. Элементы этой вкладки будут отображать координаты последнего нажатия курсора на элемент, а также общее число нажатий на элемент.

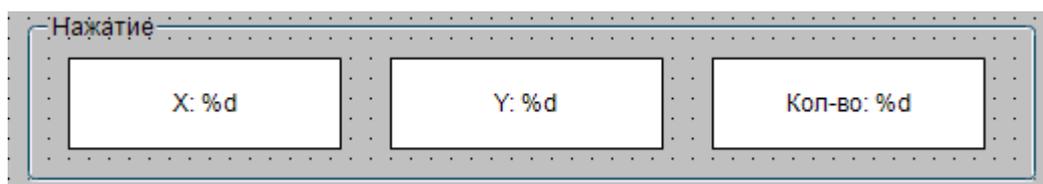


Рис. 11.4.3.6. Внешний вид вкладки **Нажатие**

[-] Тексты	
Текст	X: %d
Подсказка	
[+] Свойства текста	
[+] Абсолютное перемещение	
[+] Относительное перемещение	
[-] Текстовые переменные	
Текстовая переменная	PLC_PRG.MouseDownEvent.nXPos
Переменная подсказки	

Рис. 11.4.3.7. Настройки элемента **Прямоугольник 1**

[-] Тексты	
Текст	Y: %d
Подсказка	
[+] Свойства текста	
[+] Абсолютное перемещение	
[+] Относительное перемещение	
[-] Текстовые переменные	
Текстовая переменная	PLC_PRG.MouseDownEvent.nYPos
Переменная подсказки	

Рис. 11.4.3.8. Настройки элемента **Прямоугольник 2**

[-] Тексты	
Текст	Кол-во: %d
Подсказка	
[+] Свойства текста	
[+] Абсолютное перемещение	
[+] Относительное перемещение	
[-] Текстовые переменные	
Текстовая переменная	PLC_PRG.MouseDownEvent.dwCounter

Рис. 11.4.3.9. Настройки элемента **Прямоугольник 3**

8. Настроим элементы вкладки **Отпускание**. Элементы этой вкладки будут отображать координаты последнего отпускания курсором элемента, а также общее число отпусканий.

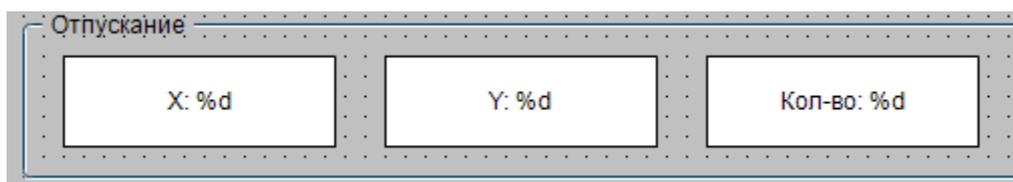


Рис. 11.4.3.10. Внешний вид вкладки **Отпускание**

[-] Тексты	
Текст	X: %d
Подсказка	
+ Свойства текста	
+ Абсолютное перемещение	
+ Относительное перемещение	
[-] Текстовые переменные	
Текстовая переменная	PLC_PRG.MouseUpEvent.nXPos
Переменная подсказки	

Рис. 11.4.3.11. Настройки элемента **Прямоугольник 4**

[-] Тексты	
Текст	Y: %d
Подсказка	
+ Свойства текста	
+ Абсолютное перемещение	
+ Относительное перемещение	
[-] Текстовые переменные	
Текстовая переменная	PLC_PRG.MouseUpEvent.nYPos
Переменная подсказки	

Рис. 11.4.3.12. Настройки элемента **Прямоугольник 5**

[-] Тексты	
Текст	Кол-во: %d
Подсказка	
+ Свойства текста	
+ Абсолютное перемещение	
+ Относительное перемещение	
[-] Текстовые переменные	
Текстовая переменная	PLC_PRG.MouseUpEvent.dwCounter
Переменная подсказки	

Рис. 11.4.3.13. Настройки элемента **Прямоугольник 6**

9. Настроим элементы вкладки **Перемещение**. Элементы этой вкладки будут отображать текущие координаты курсора, а также общее число его перемещений (т.е. изменений координат).

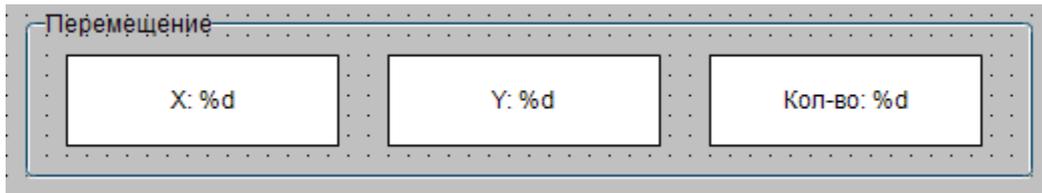


Рис. 11.4.3.14. Внешний вид вкладки **Отпускание**

[-] Тексты	
Текст	X: %d
Подсказка	
[+] Свойства текста	
[+] Абсолютное перемещение	
[+] Относительное перемещение	
[-] Текстовые переменные	
Текстовая переменная	PLC_PRG.MouseMoveEvent.nXPos
Переменная подсказки	

Рис. 11.4.3.15. Настройки элемента **Прямоугольник 7**

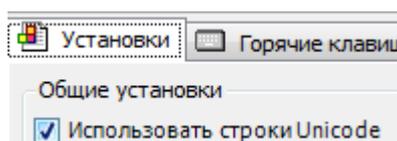
[-] Тексты	
Текст	Y: %d
Подсказка	
[+] Свойства текста	
[+] Абсолютное перемещение	
[+] Относительное перемещение	
[-] Текстовые переменные	
Текстовая переменная	PLC_PRG.MouseMoveEvent.nYPos
Переменная подсказки	

Рис. 11.4.3.16. Настройки элемента **Прямоугольник 8**

[-] Тексты	
Текст	Кол-во: %d
Подсказка	
[+] Свойства текста	
[+] Абсолютное перемещение	
[+] Относительное перемещение	
[-] Текстовые переменные	
Текстовая переменная	PLC_PRG.MouseMoveEvent.dwCounter
Переменная подсказки	

Рис. 11.4.3.17. Настройки элемента **Прямоугольник 9**

10. Настроим компонент [Менеджер визуализации](#). В его установках поставим галочку **Использовать строки Unicode**:



Настройки **target**- и **web**-визуализации будут выглядеть следующим образом:

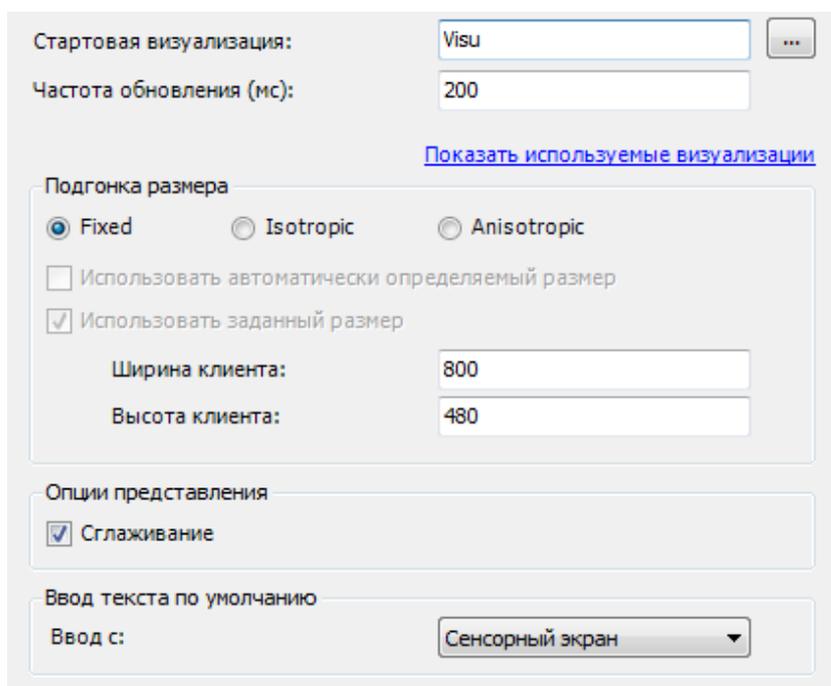


Рис. 11.4.3.18. Настройки **target**-визуализации

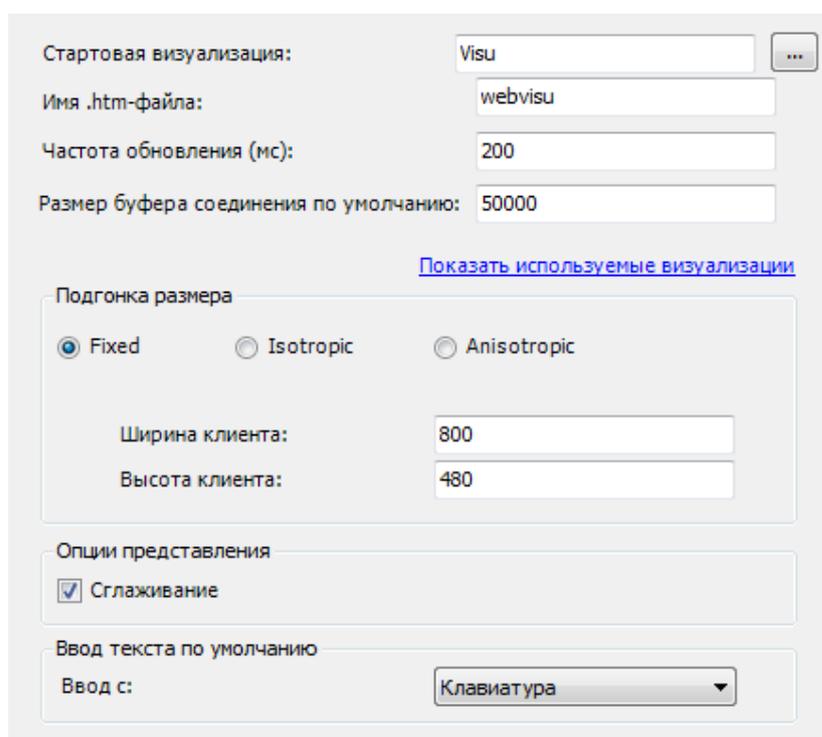


Рис. 11.4.3.19. Настройки **web**-визуализации

11. [Запустим проект на виртуальном контроллере](#). Откроем в веб-браузере страницу <http://localhost:8080/webvisu.htm>, чтобы увидеть web-визуализацию проекта. **Обратите внимание**, что браузер должен поддерживать **HTML5**. Проверим функционал проекта.

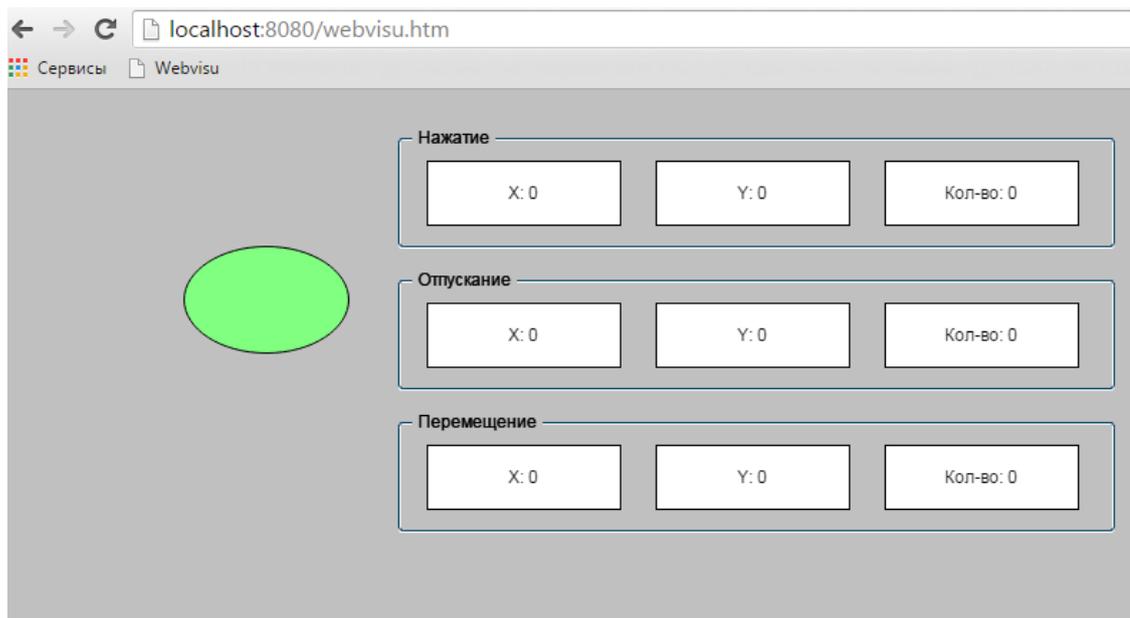


Рис. 11.4.3.20. Web-визуализация проекта в браузере **Chrome**

Нажмите на эллипс курсором, после чего перемещайте курсор по экрану. Эллипс изменит свой цвет и будет перемещаться вслед за курсором. Информация во вкладках будет обновляться по мере взаимодействия с элементом. Для того чтобы зафиксировать элемент на новом месте, нажмите на этом месте курсором.

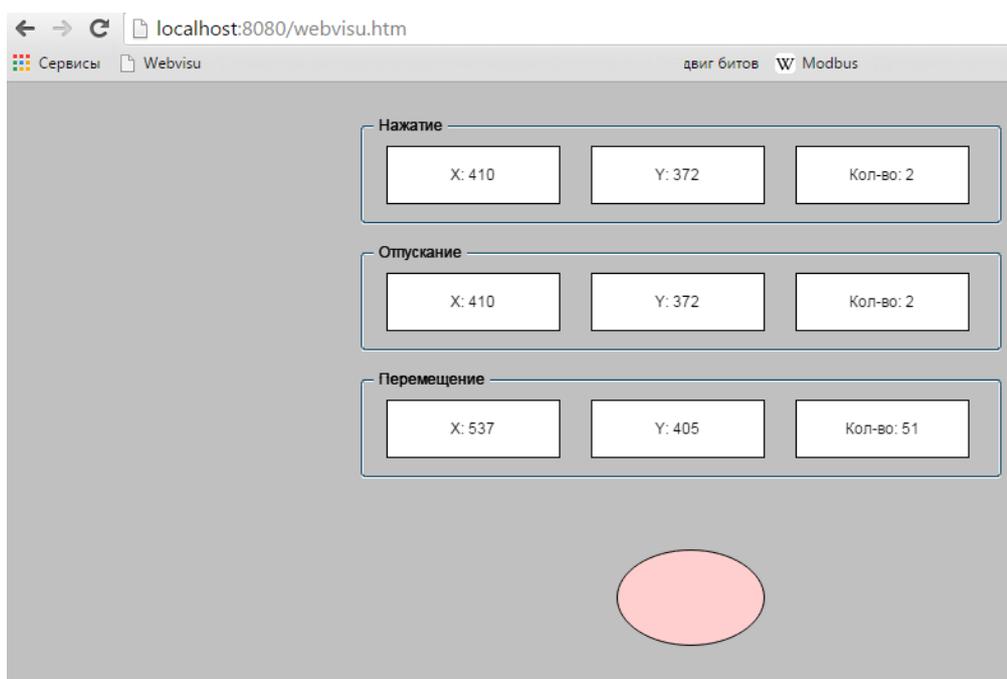


Рис. 11.4.3.21. Перемещение элемента с помощью курсора

11.4.4. Многопользовательский доступ. Независимая обработка клиентов визуализации

При использовании контроллеров с [web-визуализацией](#) (например, **СПК2хх**) может возникнуть ситуация, когда с визуализацией контроллера одновременно работает несколько клиентов. В простом случае это не создает проблем – стандартные действия вкладки [InputConfiguration](#), такие, как [Изменить отображаемую визуализацию](#) и [Переключить визуализацию в фрейме](#), работают независимо для клиентов – т.е. переключение визуализации для одного клиента не повлияет на то, что видят в данный момент остальные.

Ситуация усложняется при необходимости программной обработки визуализации (например, переключения экранов из кода программы). Использование системной переменной [CurrentVisu](#), содержащей имя открытого экрана визуализации, уже неприемлемо – поскольку при наличии соответствующей галочки в [Менеджере визуализации](#) визуализация становится общей для всех клиентов, и переключение экранов затрагивает всех пользователей.

В целом, какого-то *простого* решения данной задачи (программной обработки визуализаций в случае многопользовательского доступа) не существует. В данном пункте мы приведем несколько способов, которые могут частично решить некоторые аспекты данной задачи.

1. Переключение визуализации для всех пользователей без **CurrentVisu**

Такая задача может возникнуть, когда клиенты должны независимо переключать визуализации с помощью кнопок (поэтому **CurrentVisu** не подходит), но в каких-то ситуациях необходимо изменить отображаемую визуализацию для всех клиентов из кода программы (например, показать экран, на котором сработала тревога).

В этом случае воспользуйтесь следующим кодом:

```
VisuElems.g_VisuManager.SetMainVisuGlobally(VisuElems.Visu_ClientType.Webvisualization,sVisuName);  
VisuElems.g_VisuManager.SetMainVisuGlobally(VisuElems.Visu_ClientType.Targetvisualization,sVisuName);
```

Первая функция переключает визуализацию для всех клиентов web-визуализации, вторая – для клиента target-визуализации.

sVisuName – переменная типа **STRING**, содержащая имя экрана визуализации, который необходимо открыть.

2. Независимая работа клиентов на одном экране

В пункте [11.3.3](#) мы показывали, как работать с **интерфейсом фрейма**.

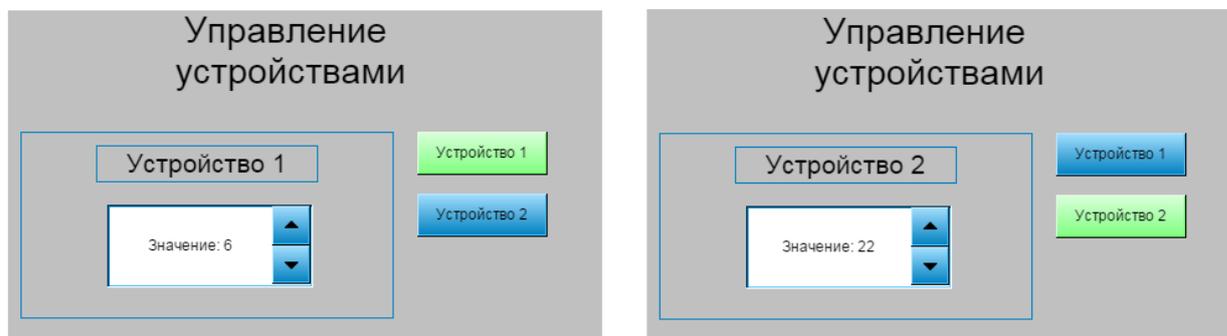


Рис. 11.4.4.1. Внешний вид примера **Интерфейс фрейма**.

На обоих рисунках в фрейме открыт один и тот же экран визуализации

Как справедливо можно заметить, этот пример не подразумевает многопользовательский доступ (впрочем, как практически все рассмотренные в документе примеры) – поскольку все пользователи взаимодействует с одними и теми же переменными, привязанными к кнопкам.

Поясним вышесказанное на примере. Пусть **первый веб-клиент** визуализации нажмет на кнопку **Устройство 2**. В результате экран будет выглядеть в соответствии с правой половиной рис. 11.4.4.1.

Но **второй веб-клиент**, подключившись, увидит следующее:



Рис. 11.4.4.2. Выделенная кнопка не соответствует отображаемому устройству

Т.е. несмотря на то, что пользователь видит в фрейме экран **Устройство 1**, активна кнопка **Устройство 2**, поскольку ее нажал первый пользователь.

Покажем, как можно избежать этого с помощью использования системной переменной **CURRENTCLIENTID**, которая содержит индекс текущего клиента визуализации. Для ее использования необходимо добавить в проект библиотеку **VisuGlobalClientManager**.

Откроем проект [Example_FrameInterface](#). В переменных программы вместо **BOOL** переменных **bButtonDevice1** и **bButtonDevice2** объявим массивы аналогичного типа, каждый из которых содержит по **10** значений. Число **10** соответствует количеству клиентов визуализации, которые могут одновременно работать с проектом.

```
2  VAR
3  // Переменные устройства 1
4  arrButtonDevice1:ARRAY [1..10] OF BOOL:= [10(TRUE)]; // переменная для изменения цвета кнопки при выборе устройства
5  wNameDevice1:WSTRING:="Устройство 1"; // название устройства
6  iValueDevice1:INT; // параметр устройства
7
8  // Переменные устройства 2
9  arrButtonDevice2:ARRAY [1..10] OF BOOL;
10 wNameDevice2:WSTRING:="Устройство 2";
11 iValueDevice2:INT;
12 END VAR
```

Рис. 11.4.4.3. Объявление переменных программы **PLC_PRG**

На экране **Visualization** отредактируем **ST-код** действий кнопок **Устройство 1** и **Устройство 2**:

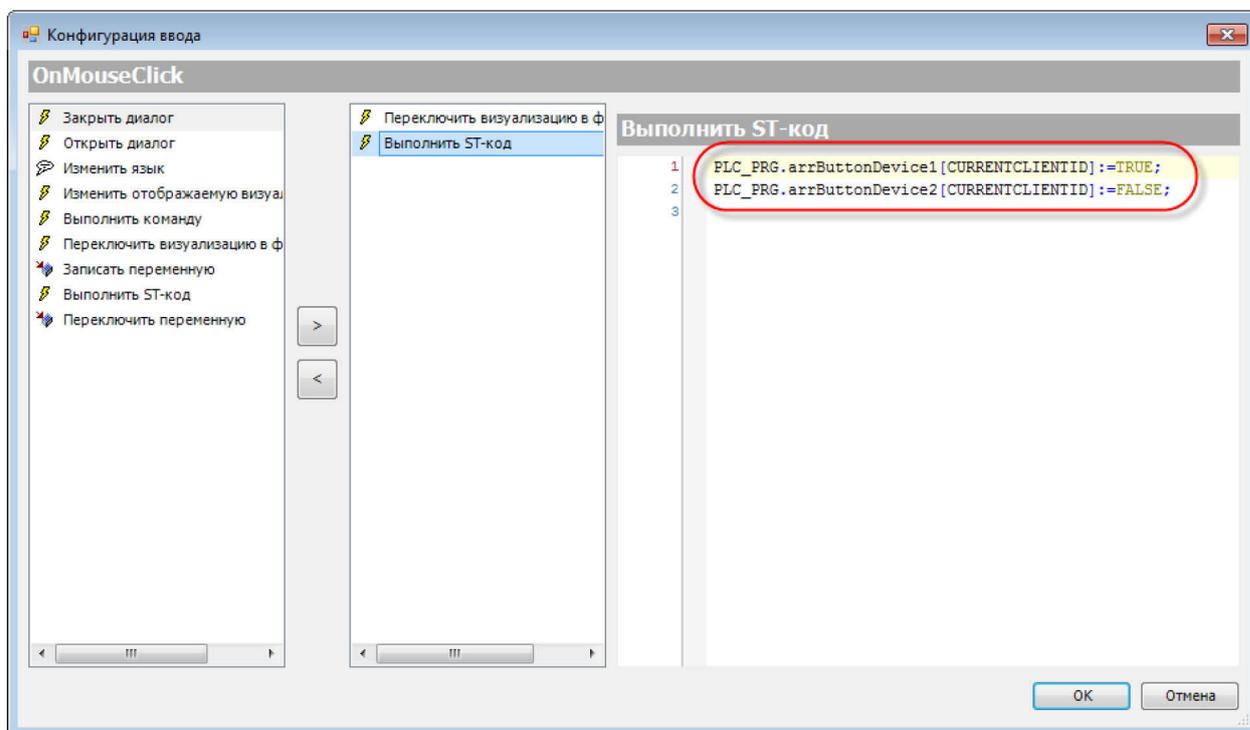


Рис. 11.4.4.4. Изменение ST-кода кнопки **Устройство 1**

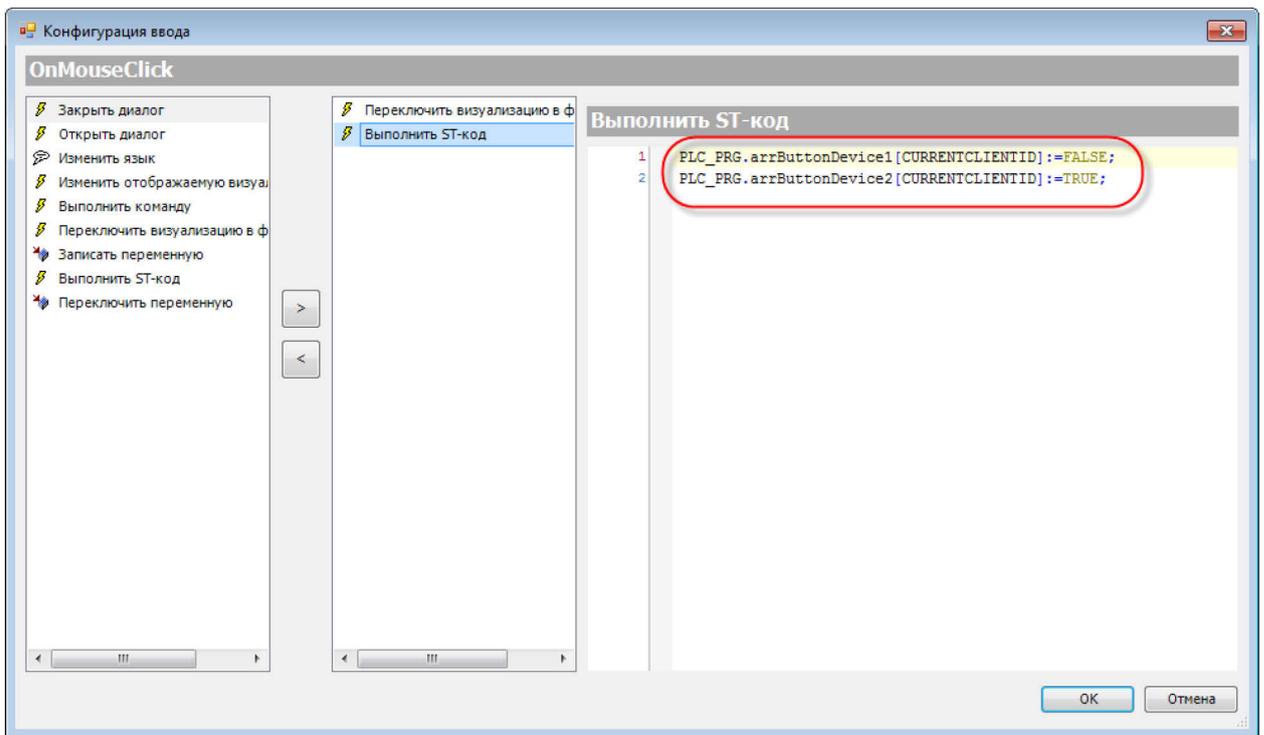


Рис. 11.4.4.5. Изменение ST-кода кнопки **Устройство 2**

К параметрам **Переключить цвет** обоих кнопок привяжем соответствующие переменные:

Переменные цвета	
Переключить цвет	PLC_PRG.arrButtonDevice1 [CURRENTCLIENTID]
Цвет	
Цвет тревоги	

Рис. 11.4.4.6. Параметры элемента **Устройство 1**

Переменные цвета	
Переключить цвет	PLC_PRG.arrButtonDevice2 [CURRENTCLIENTID]
Цвет	
Цвет тревоги	

Рис. 11.4.4.7. Параметры элемента **Устройство 2**

В результате, каждый из пользователей будет работать со своими переменными, характеризующими активность кнопок, что позволит избежать ситуации, приведенной на рис. 11.4.4.2

3. Работа с системными библиотеками

Использование системных библиотек визуализации (**VisuElems** и др.) теоретически позволяет решать сложные задачи, связанные с программной обработкой визуализации при условии многопользовательского доступа. В то же время, это сопряжено с рядом сложностей, основной из которых является отсутствие документации на данные библиотеки.

Компания [3S](#) (разработчик **CODESYS**) предоставляет несколько примеров, демонстрирующих работу с этими библиотеками; их можно использовать для ознакомления с базовыми принципами и в дальнейшем самостоятельно заниматься изучением функционала библиотек. В целом, это рекомендуется только **опытным** пользователям.

Примеры доступны для скачивания:

[ClientIndependentVisuSwitching.package](#)

[VisualizationSwitching.package](#)

[AdvancedVisu.rar](#)

Файлы формата **.package** устанавливаются с помощью **Менеджера пакетов** (вкладка **Инструменты**).

Обратите внимание, что компания [ОВЕН](#) не принимает вопросы по работе с системными библиотеками и не несет никакой ответственности за последствия их использования.

11.4.5. Подсчет количества клиентов web-визуализации

Данный пример показывает, как подсчитать текущее число клиентов web-визуализации. В его основе лежит программный код, предоставленный компанией [3S](#) (разработчик **CODESYS**), который включает в себя с работу системными библиотеками. Эти библиотеки не имеют сопроводительной документации, поэтому пользователям не рекомендуется их использование – за исключением дословного повторения примеров, предоставленных компанией **3S**.

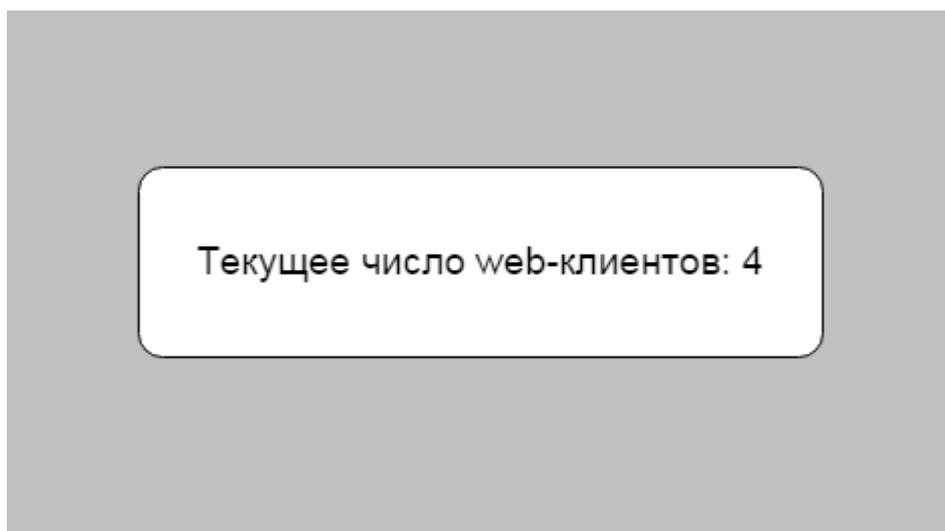


Рис. 11.4.5.1. Внешний вид примера Подсчет количества клиентов web-визуализации

Пример создан в среде **CODESYS 3.5 SP6 Patch 2** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с target-файлом **3.5.6.0**.

Пример доступен для скачивания: [Example_CountingWebClients.projectarchive](#)

1. Создадим новый **стандартный** проект **CODESYS** с названием **Example_CountingWebClients** и настройками по умолчанию: target – **CODESYS Control Win V3**, язык **PLC_PRG – ST**.

2. В программе **PLC_PRG** объявим следующие переменные:

```

1 PROGRAM PLC_PRG
2 // ВНИМАНИЕ: При выполнении итераций или других операций для элементов визуализации, используемый для этого POU
3 // ДОЛЖЕН выполняться в задаче VISU_TASK
4 VAR
5     pClient : POINTER TO VisuElems.VisuStructClientData; // указатель на данные клиента визуализации
6
7     iCountActivWebVisus :INT; // число клиентов web-визуализации
8 END_VAR

```

Рис. 11.4.5.2. Объявление переменных программы **PLC_PRG**

3. Код программы **PLC_PRG** будет выглядеть следующим образом:

```

1 //-----//
2 // Пример: Подсчет числа клиентов web-визуализации//
3 //-----//
4 iCountActivWebVisus := 0; // обнуляем счетчик в начале цикла
5
6
7 VisuElems.VisuElemBase.g_ClientManager.BeginIteration(); // Начинаем итерацию по клиентам визуализации - ...
8 // ...это базовая операция для большинства действий
9
10 WHILE (pClient := VisuElems.VisuElemBase.g_ClientManager.GetNextClient()) <> 0 DO // проходимся по всем клиентам визуализации...
11     IF pClient^.GlobalData.ClientType = VisuElems.Visu_ClientType.Webvisualization THEN // ... и когда видим клиента web-визуализации...
12         iCountActivWebVisus := iCountActivWebVisus + 1; // ...увеличиваем счетчик на +1
13     END_IF
14 END WHILE

```

Рис. 11.4.5.3. Код программы **PLC_PRG**

4. Привяжем программу **PLC_PRG** к задаче **VISU_TASK**. Задачу **MainTask** следует удалить.

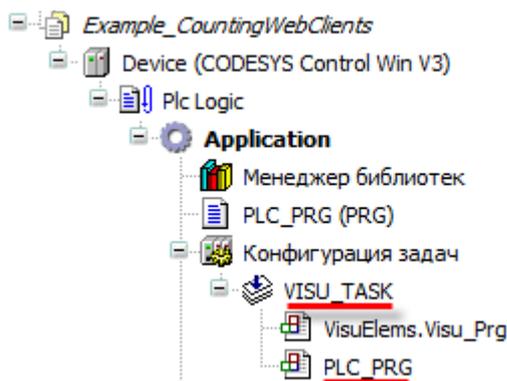


Рис. 11.4.5.4. Привязка программы **PLC_PRG** к задаче **VISU_TASK**

5. [Добавим в проект](#) экран визуализации **Visualization**. В его [свойствах](#) выберем размер **480x272**. Экран будет содержать элемент [Скругленный прямоугольник](#).

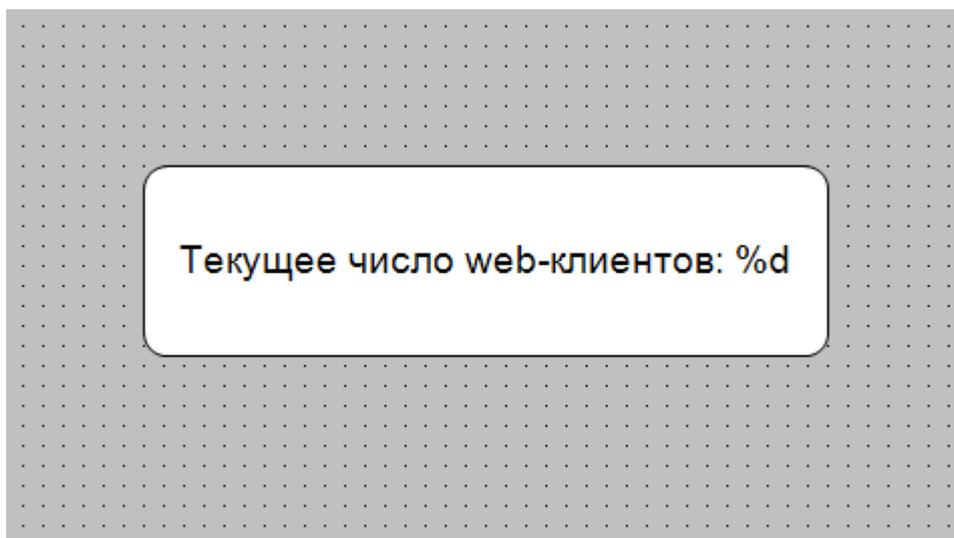


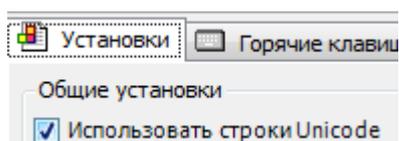
Рис. 11.4.5.5. Содержимое экрана визуализации **Visualization**

Настроим элемент следующим образом:

Свойство	Значения
Имя элемента	GenElemInst_2
ID текста	0
Тип элемента	Скругленный прямоугольник
⊕ Позиция	
⊕ Центр	
⊕ Цвета	
Использовать градиенты...	<input type="checkbox"/>
Установка градиента	linear, Black, White
⊕ Вид элемента	
⊖ Тексты	
Текст	Текущее число web-клиентов: %d
Подсказка	
⊖ Свойства текста	
Горизонтальное вырав...	По центру
Вертикальное выравни...	По центру
Шрифт	Arial; 14
⊕ Цвет шрифта	Font-Default-Color
⊕ Абсолютное перемещение	
⊕ Относительное перемещен...	
⊖ Текстовые переменные	
Текстовая переменная	PLC_PRG.iCountActivWebVisus
Переменная подсказки	

Рис. 11.4.5.6. Параметры элемента **Скругленный прямоугольник**

6. Настроим [Менеджер визуализации](#). В его установках поставим галочку **Использовать строки Unicode**:



Настройки **target**- и **web**-визуализации будут выглядеть следующим образом:

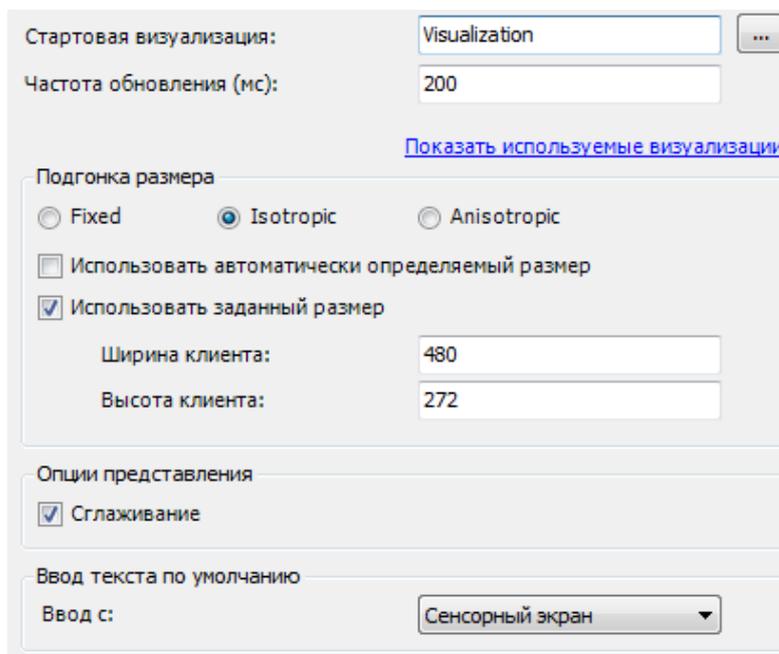


Рис. 11.4.5.7. Настройки **target**-визуализации

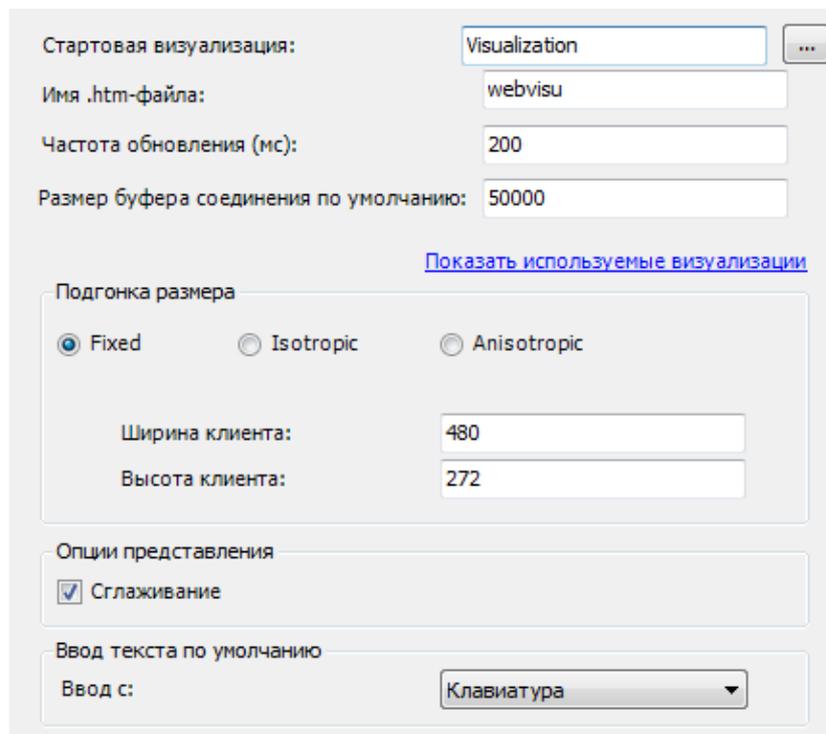


Рис. 11.4.5.8. Настройки **web**-визуализации

7. [Запустим проект на виртуальном контроллере](#). Откроем в веб-браузере страницу <http://localhost:8080/webvisu.htm>, чтобы увидеть web-визуализацию проекта. **Обратите внимание**, что браузер должен поддерживать **HTML5**. Проверим функционал проекта.

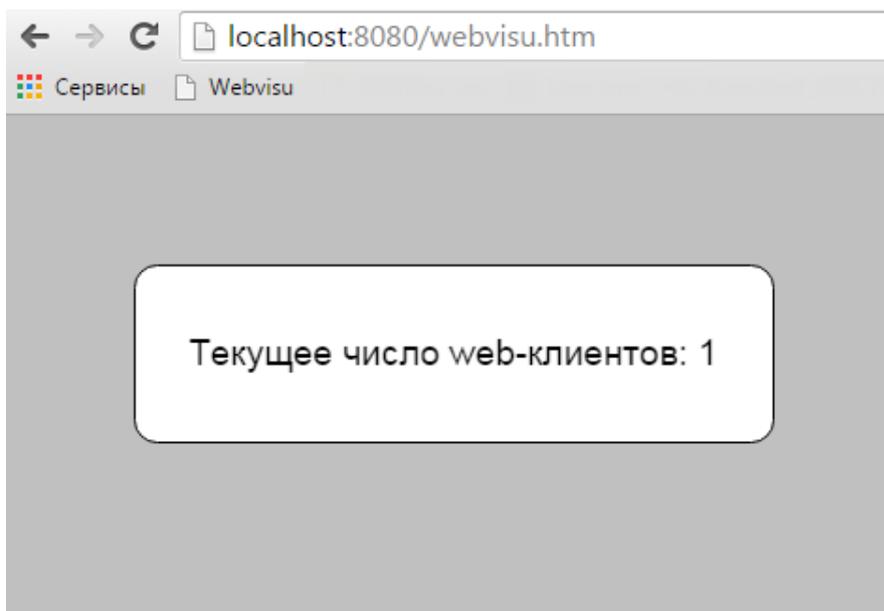


Рис. 11.4.5.9. Web-визуализация проекта в браузере **Chrome**

Текущее число web-клиентов – 1. Откройте страницу <http://localhost:8080/webvisu.htm> в еще нескольких вкладках браузера, чтобы увидеть, как изменяется число клиентов:

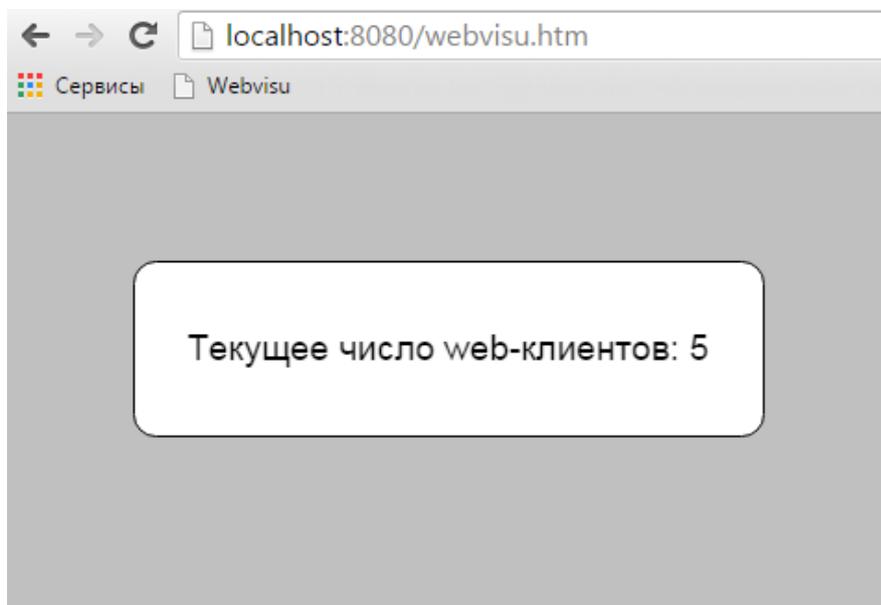


Рис. 11.4.5.10. Отображение числа клиентов web-визуализации

Обратите внимание, что максимальное число клиентов web-визуализации виртуального контроллера равно **5**. Контроллеры СПК не имеют таких ограничений, но следует понимать, что каждый из клиентов занимает память и нагружает встроенный web-сервер. На число возможных клиентов влияют **дополнительные установки Менеджера визуализации**.

11.4.6. Обработка нажатий аппаратных кнопок

Контроллеры **СПК207** имеют аппаратные кнопки (**SYS, F1, F2, F3, F4, F5**), к которым могут быть привязаны различные функции (например, переключение экранов визуализации или изменение значения логической переменной). В данном пункте приведено описание различных способов работы с аппаратными кнопками. Считать текущие состояния аппаратных кнопок в программе можно с помощью библиотеки **SPKtools**.

1. Параметр **Горячая клавиша** в настройках элемента

[Базовые](#) и часть [стандартных](#) графических примитивов содержат вкладку [InputConfiguration](#), которая в свою очередь содержит вкладку **Горячая клавиша**. Вкладка включает два основных параметра – **клавишу**, при нажатии на которую происходит **событие**.

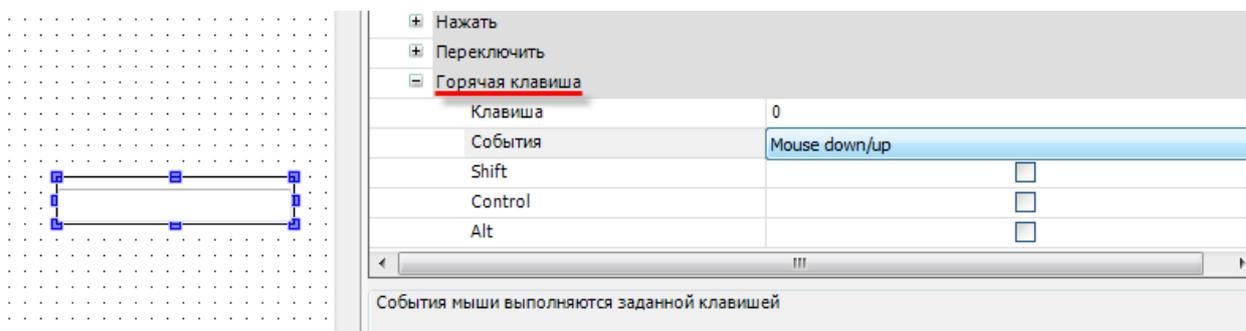


Рис. 11.4.6.1. Параметры вкладки **Горячая клавиша**

Соответствие между аппаратными кнопками СПК и клавишами, выбираемыми в данном пункте, приведено ниже:

СПК	SYS	F1	F2	F3	F4	F5
CODESYS	0	1	2	3	4	5

Пользователь может выбрать одно из следующих событий:

Mouse Down – нажатие курсора на элемент;

Mouse Up – отпускание зажатого курсора с элемента;

Mouse Down/Up – нажатие и последующее отпускание курсора с элемента («клик»).

Иными словами, нажатие аппаратной кнопки СПК будет эквивалентно нажатию на элемент. На рис. 11.6.4.1 приведен пример настройки элемента [Прямоугольник](#), при котором нажатие на клавишу **SYS** будет восприниматься СПК как нажатие на данный прямоугольник.

С помощью этой же вкладки можно настроить другие клавиши (например, **Space**, **Delete** и т.д.), которые будут активны в web-визуализации. Для них можно использовать галочки **Shift**, **Control**, **Alt** – тогда событие будет происходить при нажатии соответствующей комбинации клавиш.

2. Конфигурация горячих клавиш в настройках экрана визуализации

После настройки элемента в соответствии с рис. 11.6.4.1, на вкладке [Конфигурация горячих клавиш](#) данного экрана визуализации появятся две строки – первая из них соответствует нажатию на элемент, вторая – отпусканью.

Клавиша	Клавиша нажата	Shift	Ctrl	Alt	Тип действия	Действие	ID элемента	Права доступа
0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Выполнить событие мыши		#0	Будут использованы...
0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Выполнить событие мыши		#0	Будут использованы...

Рис. 11.4.6.2. Редактор горячих клавиш экрана **Visualization**

С помощью этого редактора можно не только настраивать горячие клавиши элементов данного экрана, но и привязать действие не к элементу, а напрямую к нажатию клавиши:

Клавиша	Клавиша нажата	Shift	Ctrl	Alt	Тип действия	Действие	ID элемента	Права доступа
0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Выполнить событие мыши		#0	Будут использованы...
0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Выполнить событие мыши		#0	Будут использованы...
1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Переключить переменную	bVar		Не определено. Пол...

Рис. 11.4.6.3. Добавление действия для кнопки СПК **F1** (для экрана **Visualization**)

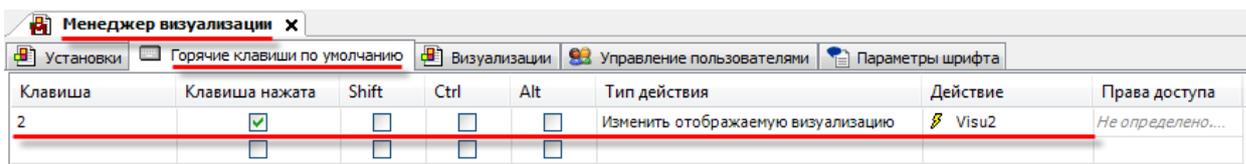
В данном случае, при нажатии на кнопку **F1** значение логической переменной **bVar** будет изменено на противоположное.

Обратите внимание, что настроенные здесь клавиши будут работать только на данном экране визуализации. К каждой клавише можно привязать несколько действий.

Список доступных действий приведен в [п. 10.2](#).

3. Конфигурация горячих клавиш в Менеджере визуализации

В предыдущем пункте показано, как привязать к клавише СПК выполнение действия – но, как упоминалось, клавиша будет работать только при отображении данного экрана визуализации. Если необходимо выполнять действие по нажатию аппаратной кнопки вне зависимости от текущего экрана визуализации, то следует настраивать его на вкладке **Горячие клавиши по умолчанию** [Менеджера визуализации](#).



Клавиша	Клавиша нажата	Shift	Ctrl	Alt	Тип действия	Действие	Права доступа
2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Изменить отображаемую визуализацию	⚡ Visu2	Не определено....
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			

Рис. 11.4.6.4. Добавление действия для кнопки СПК **F2** (для всех экранов проекта)

В данном случае, при нажатии на кнопку **F2** будет осуществлен переход на экран визуализации **Visu2**.

Обратите внимание, что действия, заданные на данной вкладке и [в Редакторе горячих клавиш](#) экранов визуализации, не перекрывают, а дополняют друг друга. К каждой клавише можно привязать несколько действий.

12. Визуализация на СПК: ограничения и рекомендации

Среда программирования **CODESYS** предоставляет широкие возможности по созданию экранов визуализации для операторского интерфейса; в то же время, аппаратные и программные возможности устройств, для которых разрабатывается визуализация, накладывают на нее определенные ограничения. В этой главе описаны ограничения и приведены рекомендации по разработке визуализации для сенсорных панельных контроллеров **СПК1xx** и **СПК2xx**.

12.1. Размер экранов визуализации

Размер экранов визуализации должен соответствовать разрешению СПК. Для СПК107, СПК110 и СПК2xx оно составляет **800x480**, для СПК105 – **480x272**. Размер задается для [target-визуализации](#) и в [свойствах](#) каждого экрана визуализации проекта.

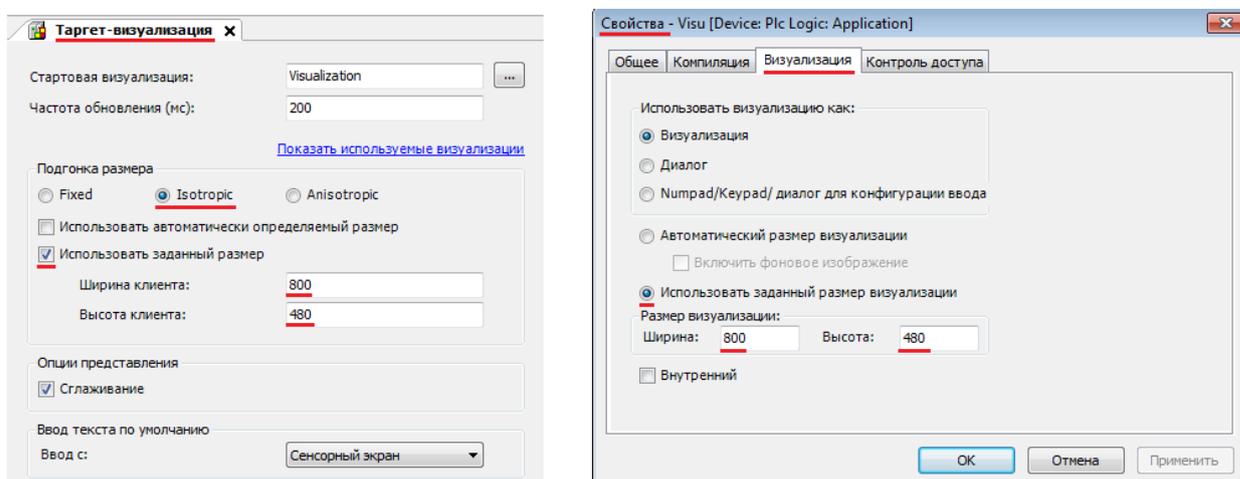


Рис. 12.1. Настройка размеров **target-визуализации** и экрана визуализации **Visu**

12.2. Отображение кириллического текста в визуализации

Для отображения кириллического текста на дисплее СПК, необходимо использовать переменные типа **WSTRING**, предварительно поставив в [Менеджере визуализации](#) галочку **Использовать строки Unicode**:

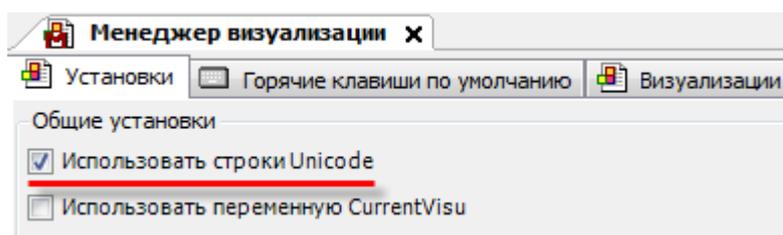


Рис. 12.2. Установка галочки **Использовать строки Unicode** в **Менеджере визуализации**

12.3. Тип ввода по умолчанию

Поскольку СПК не имеет аппаратной клавиатуры (возможность ее подключения по USB также отсутствует), то пользовательский ввод должен осуществляться с помощью экранной клавиатуры. Для этого необходимо в настройках [target-визуализации](#) установить по умолчанию тип ввода **Сенсорный экран**:



Рис. 12.3. Настройка типа ввода по умолчанию для **target-визуализации**

В то же время, при работе с [web-визуализацией](#) удобно использовать клавиатуру ПК (при условии, что web-визуализация запускается именно на ПК, а не на планшетах и подобных устройствах):

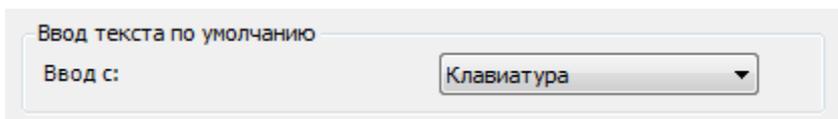


Рис. 12.4. Настройка типа ввода по умолчанию для **web-визуализации**

Помимо этого, для каждого элемента при настройке действия [Записать переменную](#) можно выбрать произвольный тип ввода:

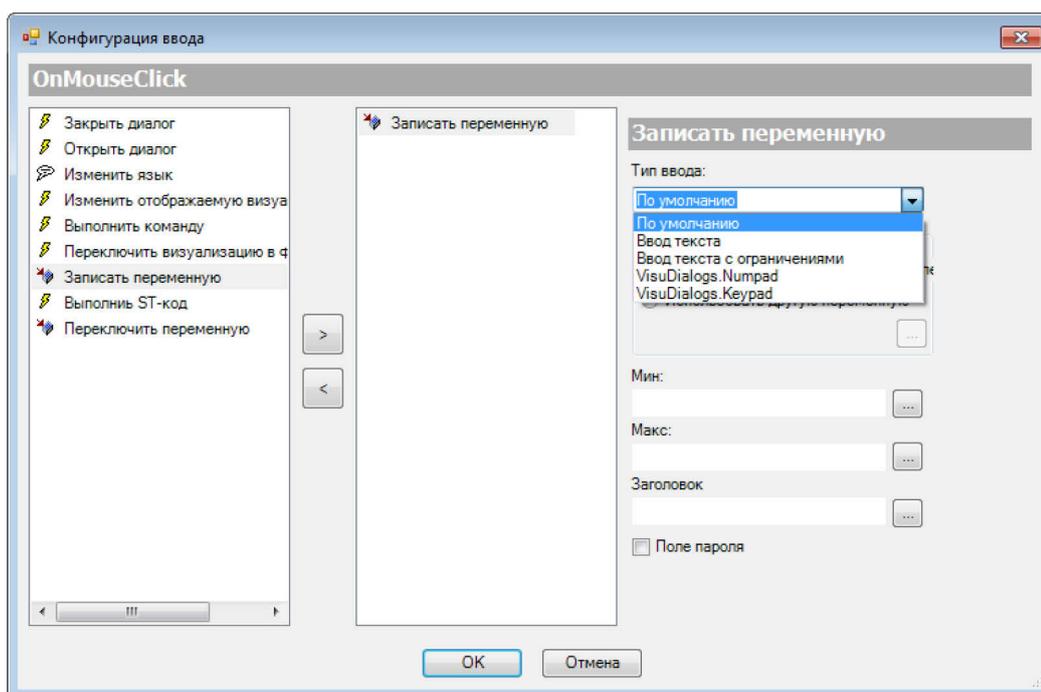


Рис. 12.5. Настройка типа ввода для действия **Запись переменную**

12.4. Отсутствие поддержки элементов ActiveX и Путь3D

Элементы [ActiveX](#) и [Путь 3D](#) не поддерживаются СПК.

12.5. Поддержка трендов

Элементы [Тренд](#) и [WebBrowser](#) не поддерживаются в прошивках **3.xxx**.

12.6. Изображения

Не рекомендуется использовать [графические файлы](#) с разрешением, которое превышает размер элемента в **CODESYS** (например, если ваша кнопка имеет размер **40x15**, то не стоит использовать в качестве подложки графический файл с разрешением **400x150**).

Поддерживаются графические файлы форматов .bmp, .jpg, jpeg, .png, .svg (спецификации **1.0** и **1.1**).

12.7. Градиенты

Для **СПК1xx** не рекомендуется использование градиентных цветов, поскольку это может привести к резкому снижению скорости отклика визуализации.

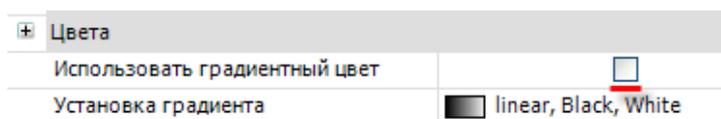


Рис. 12.6. Параметр **Использовать градиентный цвет**

Эти ограничения касаются только параметра **CODESYS** и не распространяются на [графические файлы](#), используемые в проекте.

12.8. Время цикла визуализации

Скорость отрисовки и отклика визуализации напрямую связана с временем цикла задачи VISU_TASK.

Для СПК2хх рекомендуется выставлять время цикла визуализации в пределах 100...200 мс, для СПК1хх – 200...300 мс.



Рис. 12.7. Настройки времени цикла задачи визуализации

12.9. Поддерживаемые шрифты

Начиная с прошивки версии **4.800** и таргет-файлов **версии 3.5.4.22**, на СПК поддержан следующий набор шрифтов, для которых гарантируется соответствие отображения на дисплее СПК/в редакторе **CODESYS**/в web-визуализации:

Шрифт	Доступные начертания
DS Cristal	Обычный, наклонный, полужирный, полужирный наклонный
PF Agora Serif Pro	Обычный, курсив, средний, средний курсив, полужирный, полужирный курсив
PF DinDisplay Pro	Обычный, курсив, средний, средний курсив, полужирный, полужирный курсив
Tahoma	Обычный, наклонный, полужирный, полужирный наклонный
Times New Roman	Обычный, курсив, полужирный, полужирный курсив

Для установки шрифтов на ПК достаточно установить пакет таргет-файлов для СПК версии **3.5.4.22** или выше.

Для того чтобы изменить шрифт элемента в визуализации, во вкладке **Свойства текста** выберите параметр **Шрифт**:

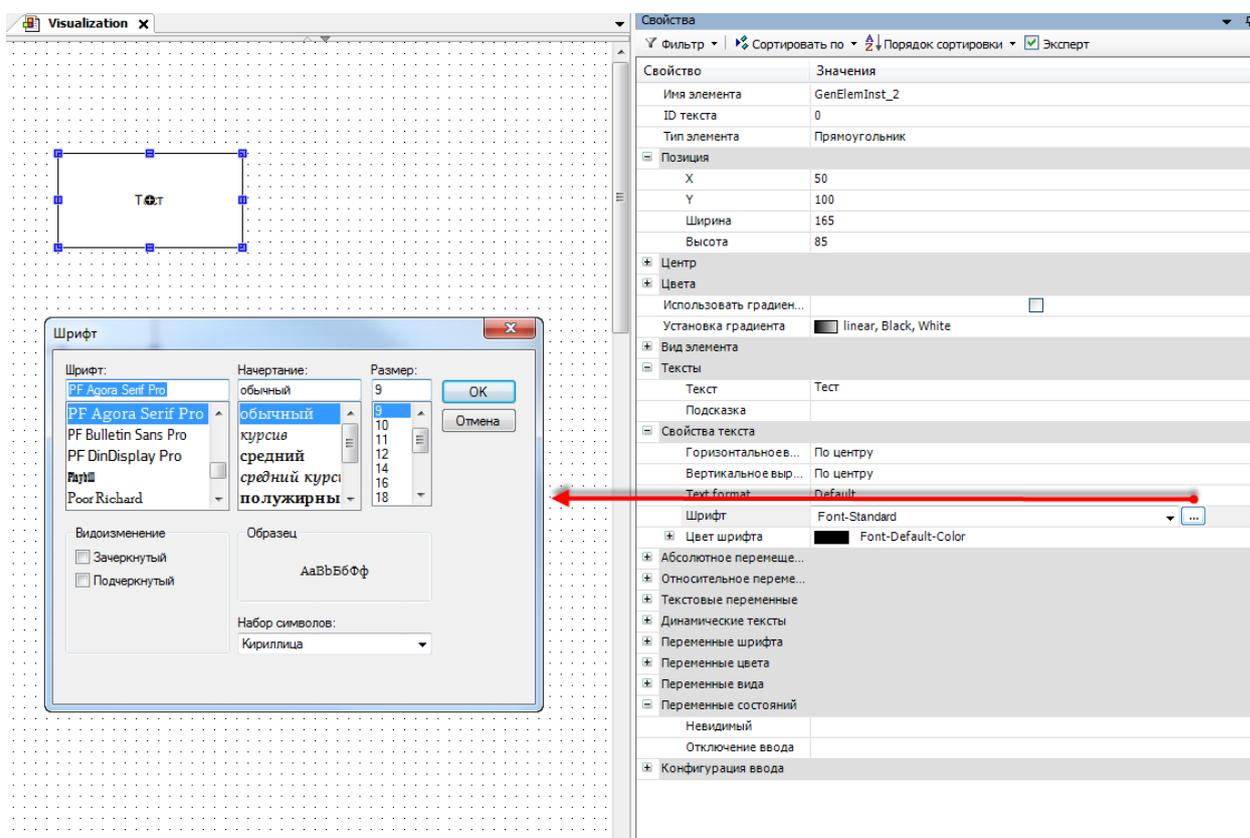


Рис. 12.8. Выбор шрифта элемента

Чтобы не выбирать шрифт для каждого элемента визуализации, можно создать свой стиль с выбранными шрифтами по умолчанию. Для этого перейдите в [Менеджер визуализации](#) и откройте **Редактор стилей**:

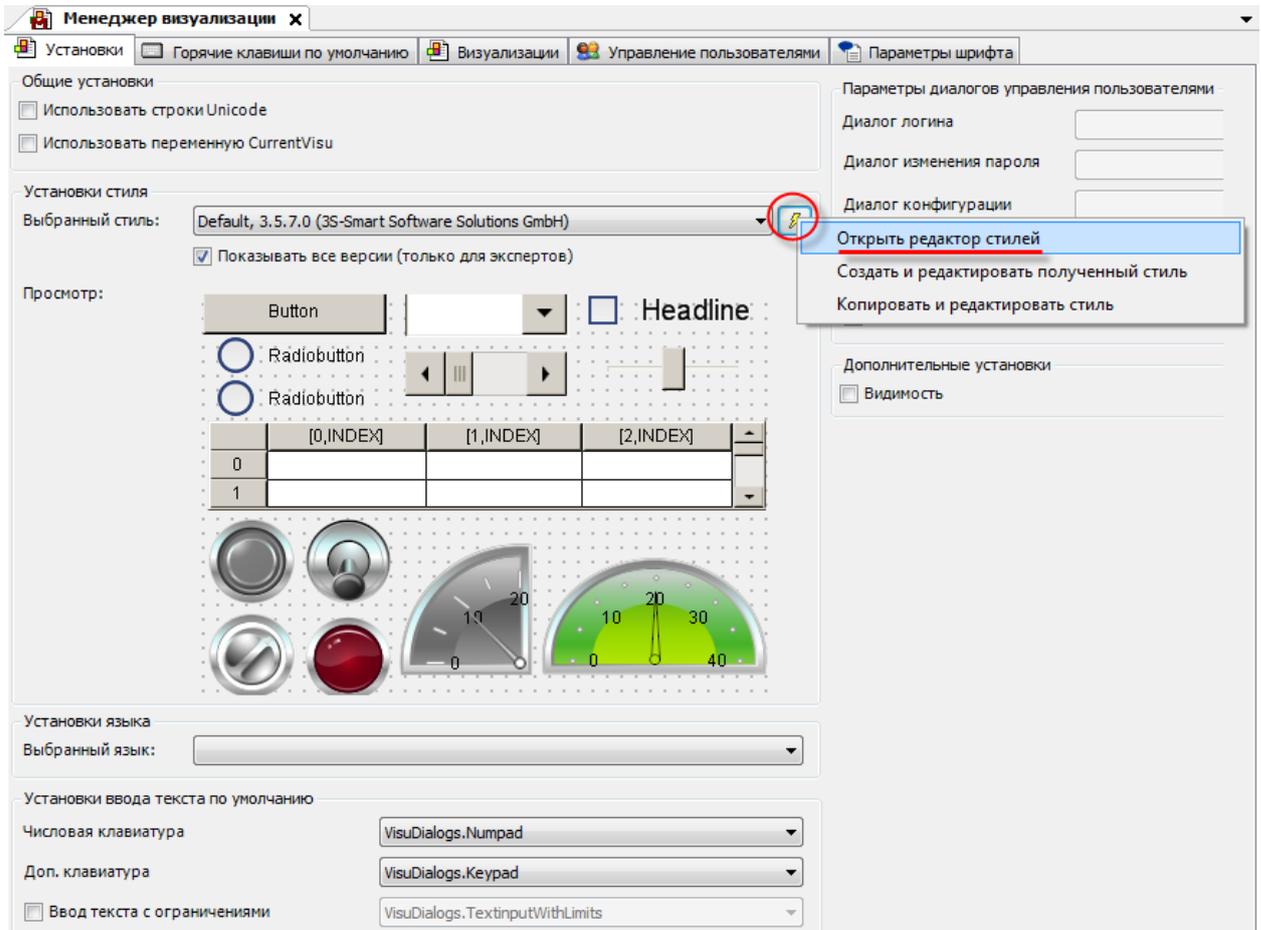


Рис. 12.9. Запуск редактора стилей

В меню **File** выберите команду **Open as copy**, чтобы создать свой стиль на основе одного из существующих. Укажите папку, в которой он будет сохранен (*рекомендуется* сохранять каждый стиль в отдельную папку). **Обратите внимание**, что данная папка должна существовать в ОС и у пользователя должны быть права на ее изменение.

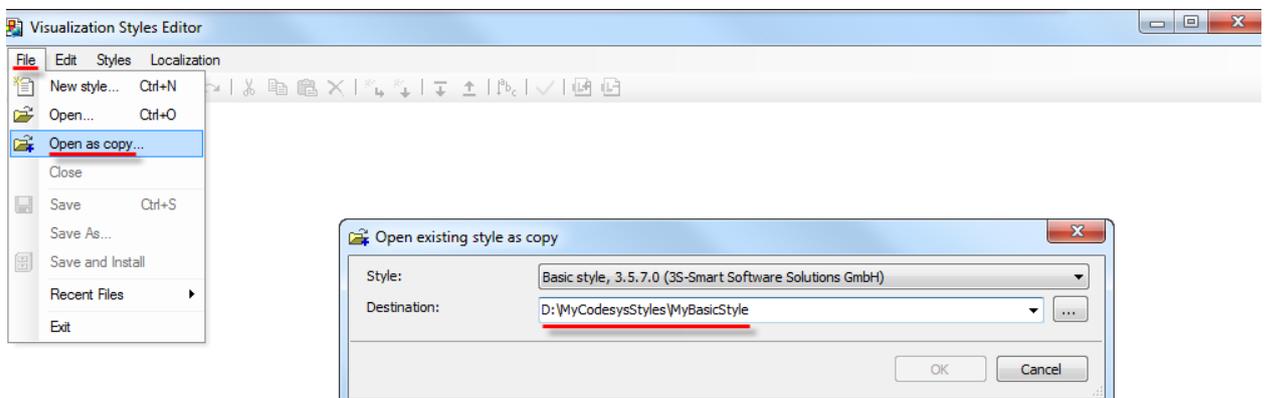


Рис. 12.10. Создание нового стиля на базе существующего

На вкладке **General** укажите название стиля и его версию:

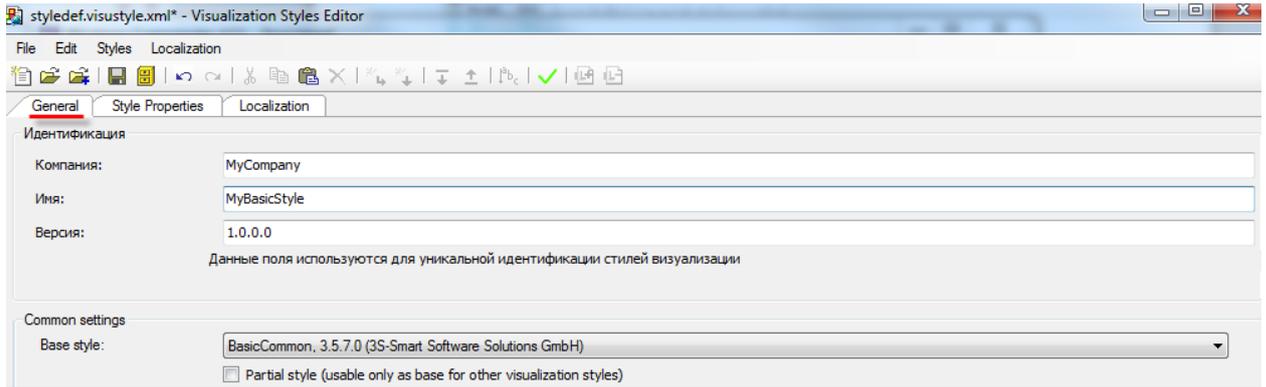


Рис. 12.11. Настройки вкладки **General**

На вкладке **Style Properties** выберите шрифты по умолчанию (из числа тех, что поддерживаются СПК):

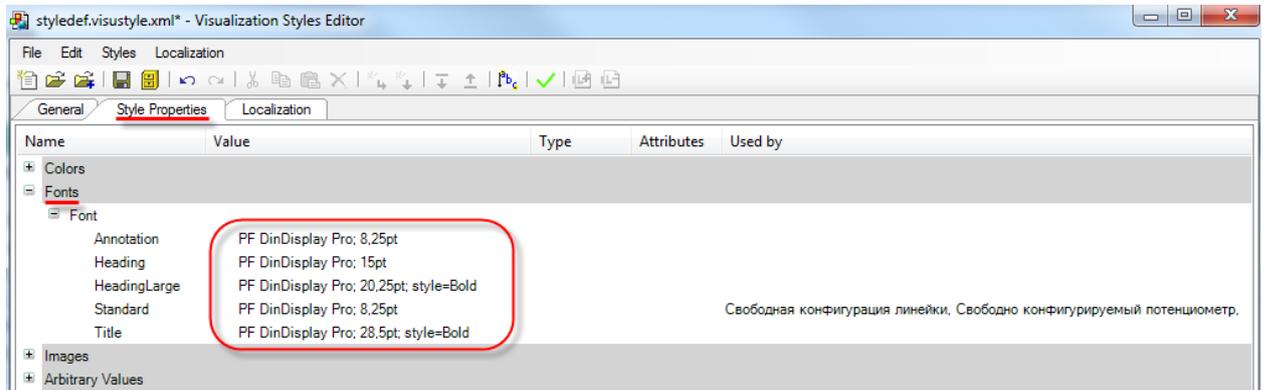


Рис. 12.12. Настройки вкладки **Style Properties**

В меню **File** выберите команду **Save and Install**, чтобы сохранить стиль в репозитории.

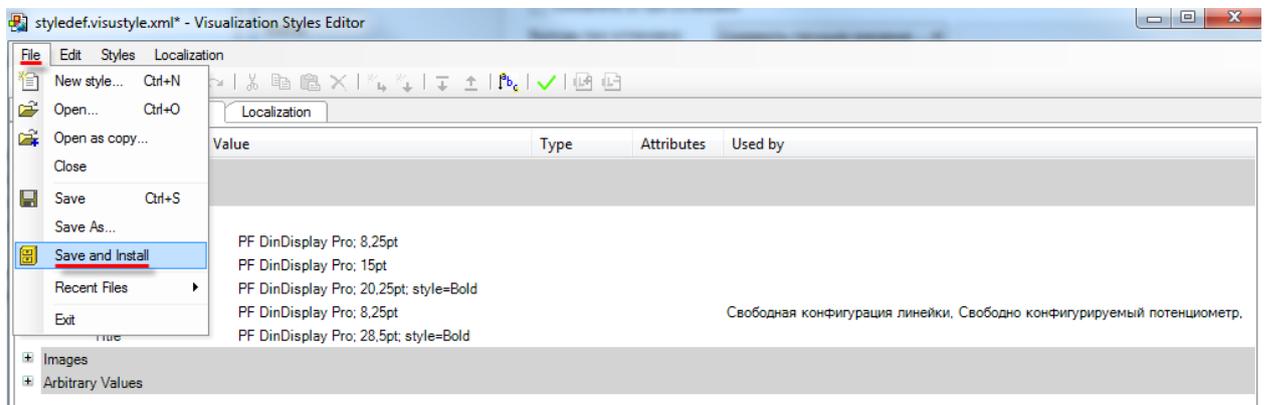


Рис. 12.13. Сохранение стиля визуализации

Перезапустите **CODESYS**. После этого можно будет выбрать сохраненный стиль в **Менеджере визуализации**:

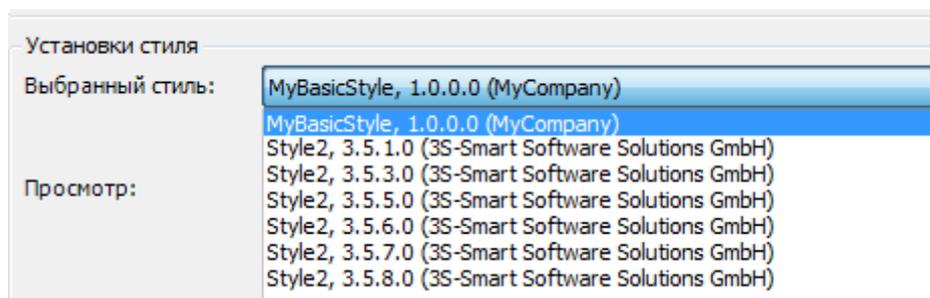


Рис. 12.14. Выбор стиля в **Менеджере визуализации**

13. Список изменений в новых сервис-паках (SP) CODESYS

Среда программирования CODESYS находится в процессе постоянного развития; приблизительно раз в полгода выходят новые **сервис-паки (SP)**, добавляющие новый функционал. Данный документ основан на версии **CODESYS 3.5 SP6 Patch 2**. В этой главе рассмотрены изменения компонентов визуализации, которые внесли следующие сервис-паки (**SP7** и далее). Скачать новые версии **CODESYS** можно на [сайте продукта](#) (требуется пройти регистрацию) и на сайте компании [OBEH](#) в разделе **CODESYS V3**. **Обратите внимание**, что некоторые возможности новых сервис-паков могут не поддерживаться **СПК** до момента выхода новой прошивки.

13.1. CODESYS 3.5 SP7

13.1.1. Внешний вид Панели инструментов

В SP6 **Панель инструментов** редактора визуализации состояла из раскрывающихся вкладок; в SP7 на смену им пришли категории элементов. Пользователь может одновременно работать с элементами одной или нескольких категорий, а также создавать свои собственные, наполняя их необходимыми стандартными элементами.

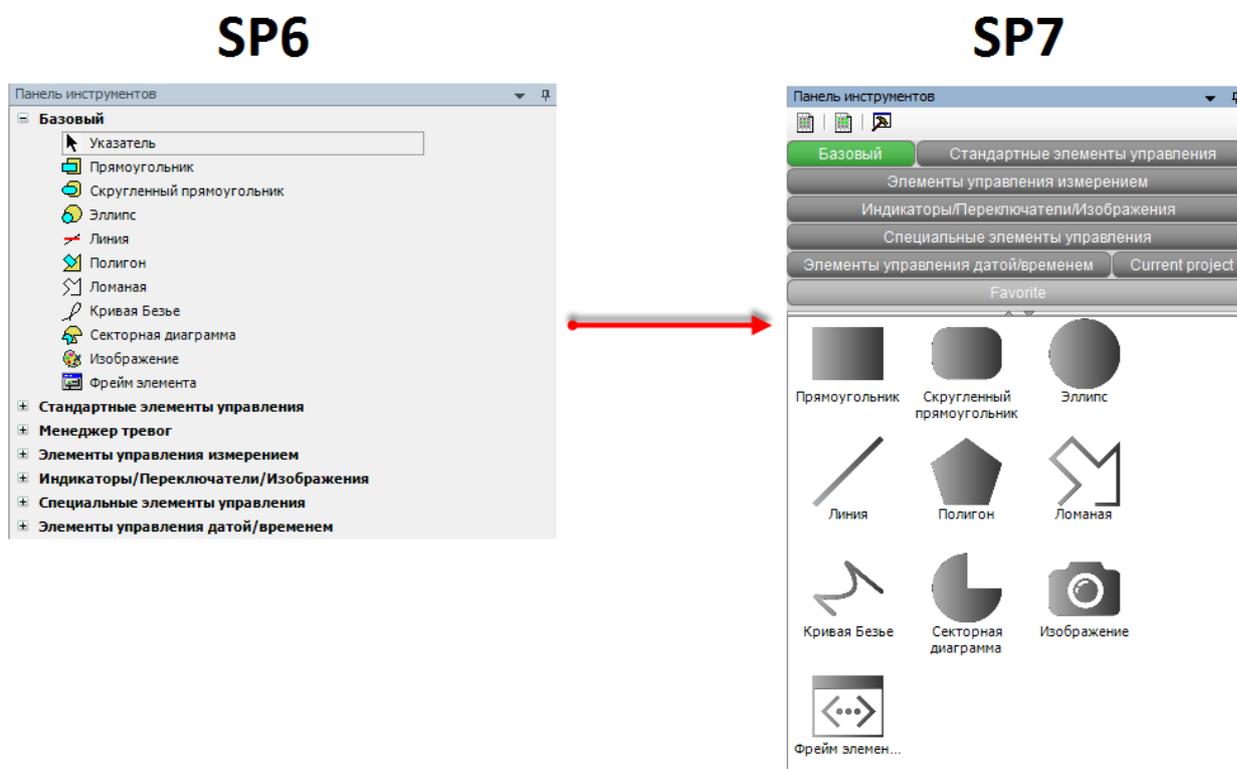


Рис. 13.1. Внешний вид Панели инструментов в SP6 и SP7

13.1.2. Библиотека VisuSymbols

Библиотека **VisuSymbols**, появившаяся в **SP7**, содержит набор пиктограмм, которые расположены на **Панели инструментов** в категории **Symbols**:

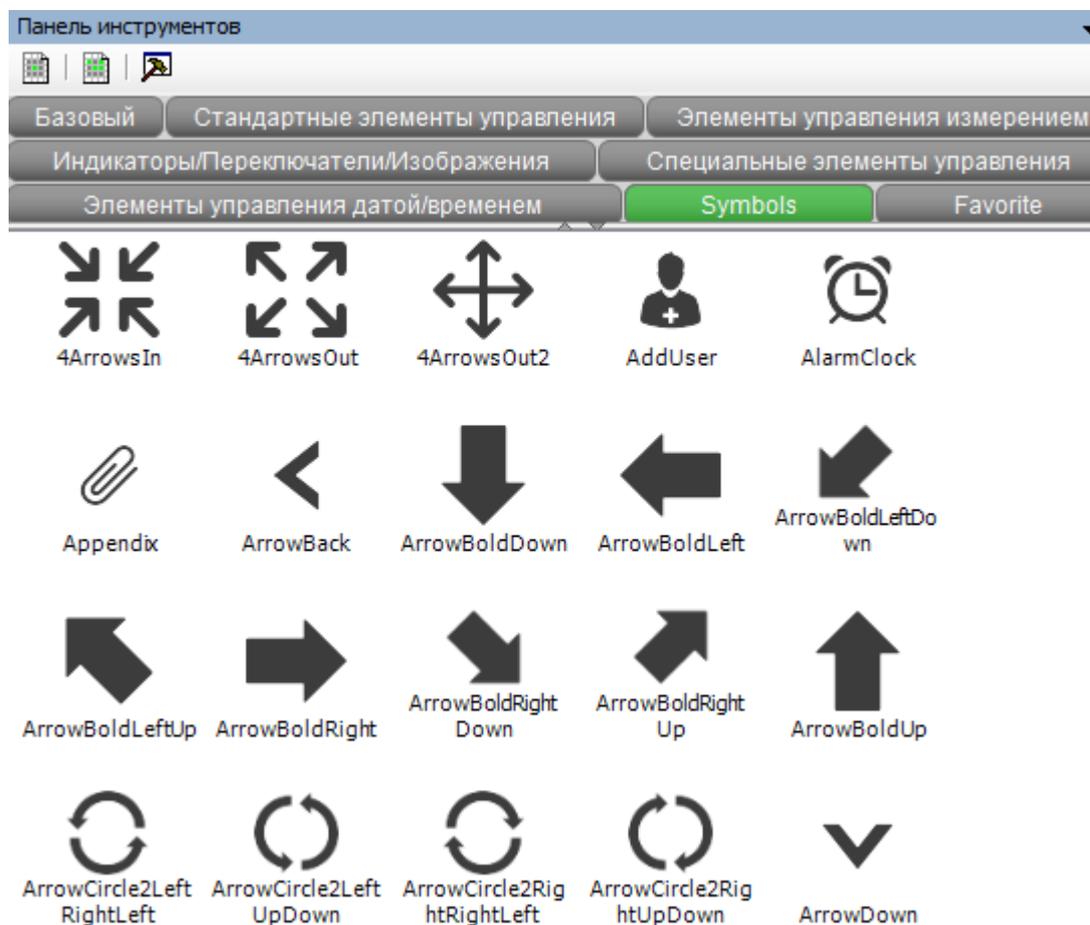


Рис. 13.2. Содержимое тега **Symbols**

13.1.3. Дополнительные оси для трендов

Для элемента [Тренд](#) теперь можно создавать дополнительные оси – например, для отображения переменных, чьи значения отличаются друг от друга на порядок и более.

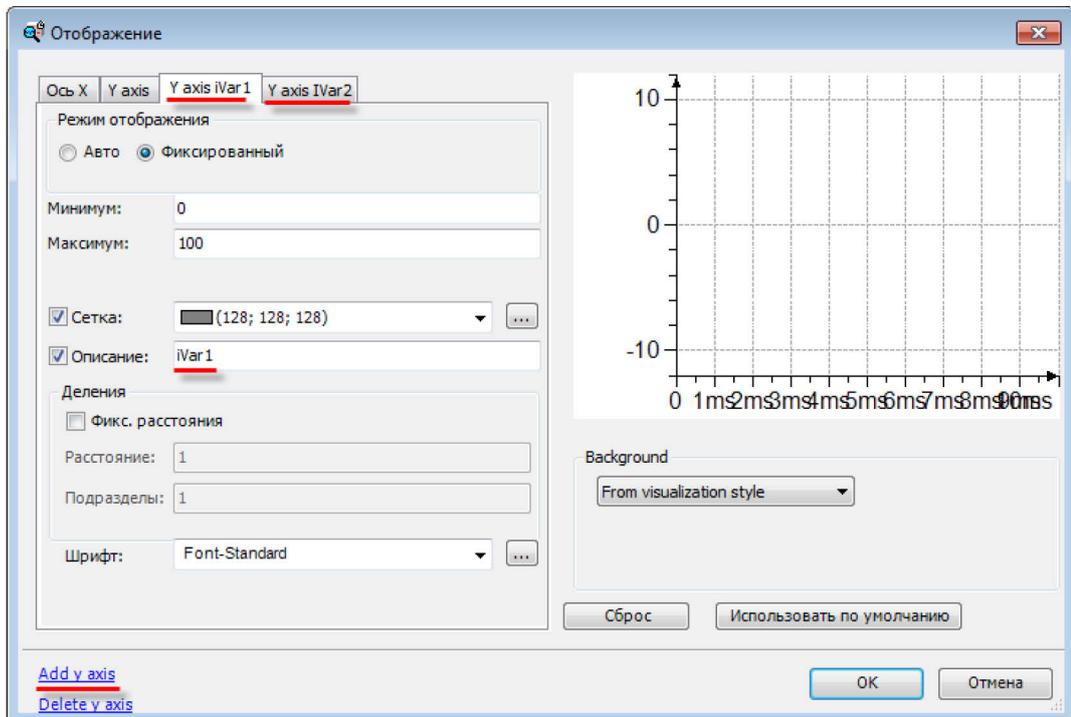


Рис. 13.3. Создание дополнительных осей

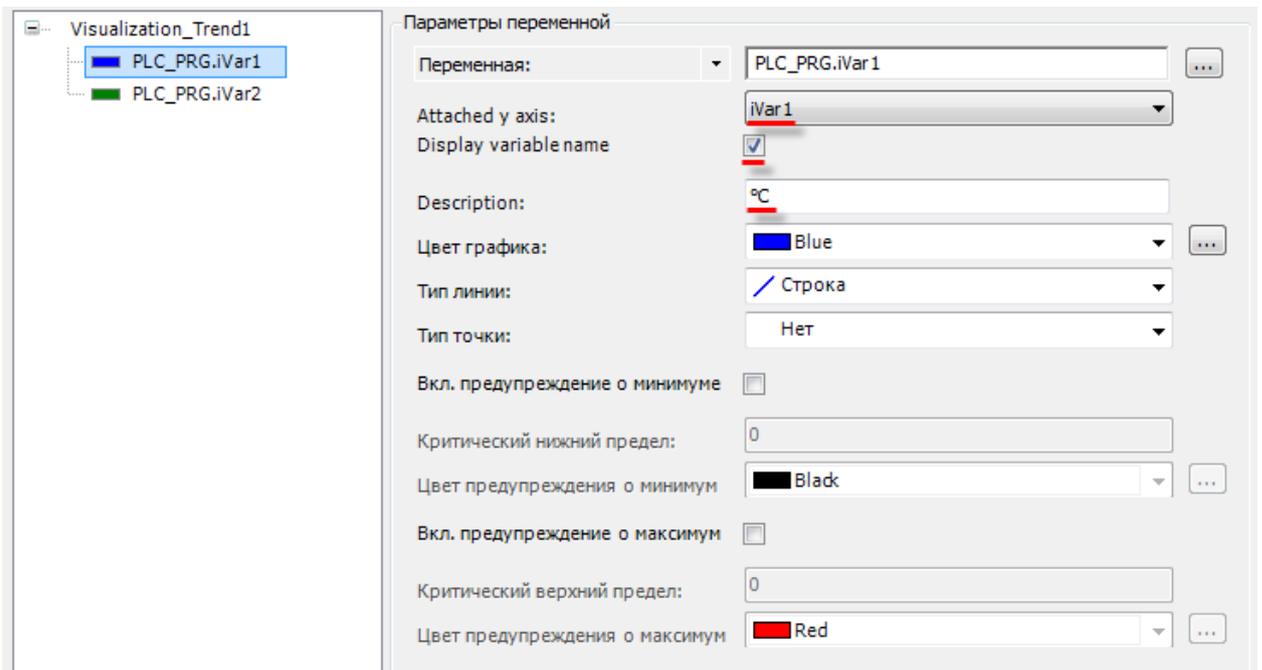


Рис. 13.4. Выбор осей для переменных



Рис. 13.5. Тренд с двумя осями в визуализации **CODESYS**

13.1.4. Экспорт/импорт групп тревог

В компоненте [Конфигуратор тревог](#) теперь можно экспортировать и импортировать списки тревог из **.csv** файлов.

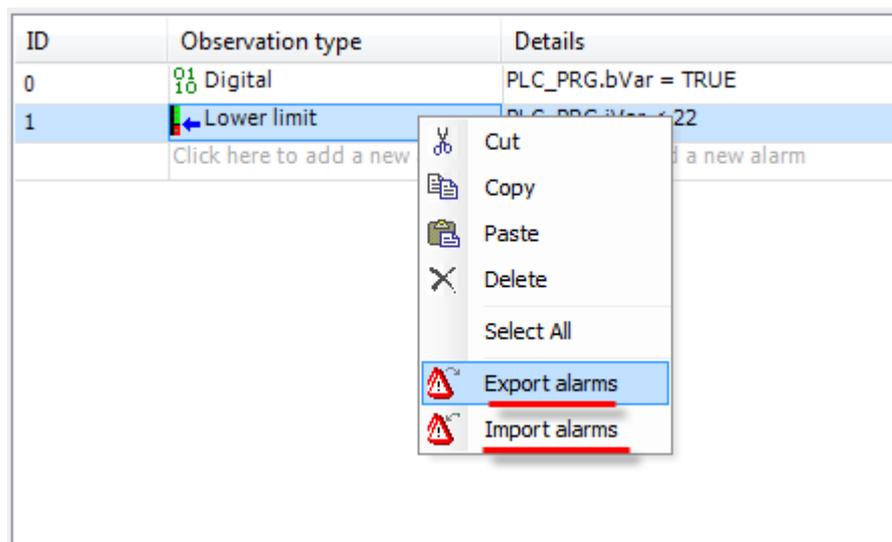


Рис. 13.6. Команды экспорта/импорта тревог

13.1.5. Параметр Text Format

К [типичным параметрам](#) графических примитивов добавился параметр **Text Format** (вкладка **Свойства текста**). Параметр определяет тип переноса текста в элементе, если он не умещается в одну строку.

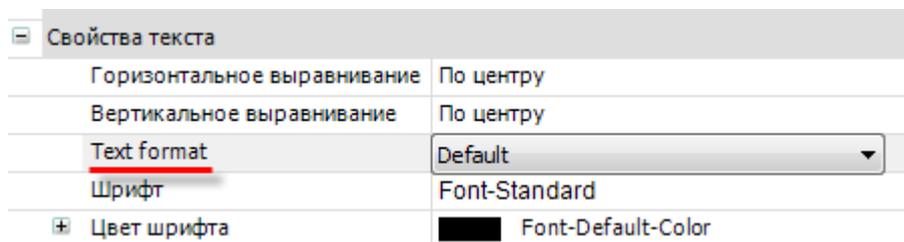


Рис. 13.7. Параметр **Text Format**

Default – текст не переносится;

Linebreak – текст переносится;

Elipsis – текст не переносится, последний отображенный фрагмент заканчивается троеточием.

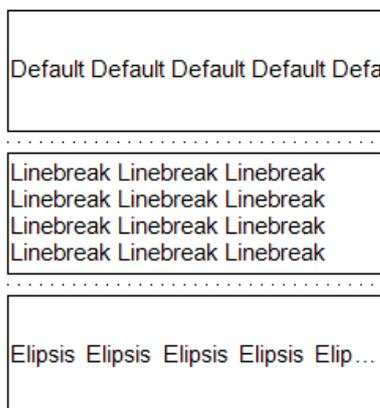


Рис. 13.8. Перенос текста при различных значениях параметра **Text Format**

13.1.6. Изменения в локализации

В **CODESYS 3.5 SP7 Patch 3** внесены серьезные изменения в русскоязычную локализацию; в частности, исправлены ошибки при добавлении в проект [Конфигуратора тревог](#), настройки действия [Открыть диалог](#), [меню выбора фона](#) экрана визуализации, которые ранее проявлялись при использовании русскоязычного интерфейса среды программирования.

13.2. CODESYS 3.5 SP8

13.2.1. Графический примитив Часы

В SP8 добавлен новый графический примитив – **Часы** (категория **Элементы управления датой/временем**). Он может использоваться как для отображения системного времени, так и для визуализации значения переменной типа **TIME_OF_DAY**. Дизайн элемента может быть настроен пользователем (фоновое изображение, толщина стрелок и т.д.).



Рис. 13.9. Внешний вид элемента **Часы**

13.2.2. Толщина линий Трассировок/Трендов

У элементов [Трассировка](#) и [Тренд](#) появилась возможность настроить ширину и стиль линии каждой отображаемой переменной:

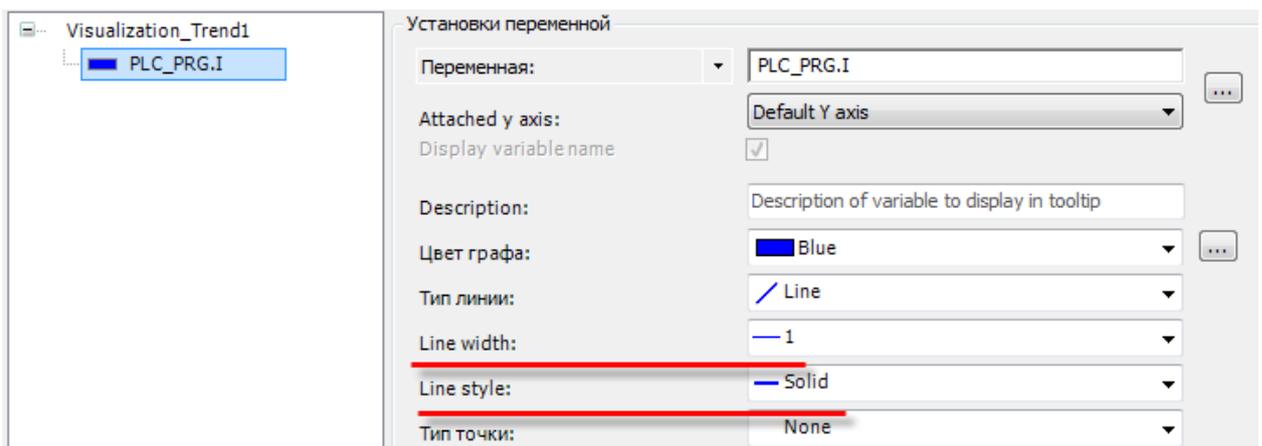


Рис. 13.10. Настройки отображения переменной тренда

13.2.3. Таблица тревог

Для элементов [Таблица тревог](#) и [Баннер тревог](#) появилась возможность создавать несколько столбцов сообщений:

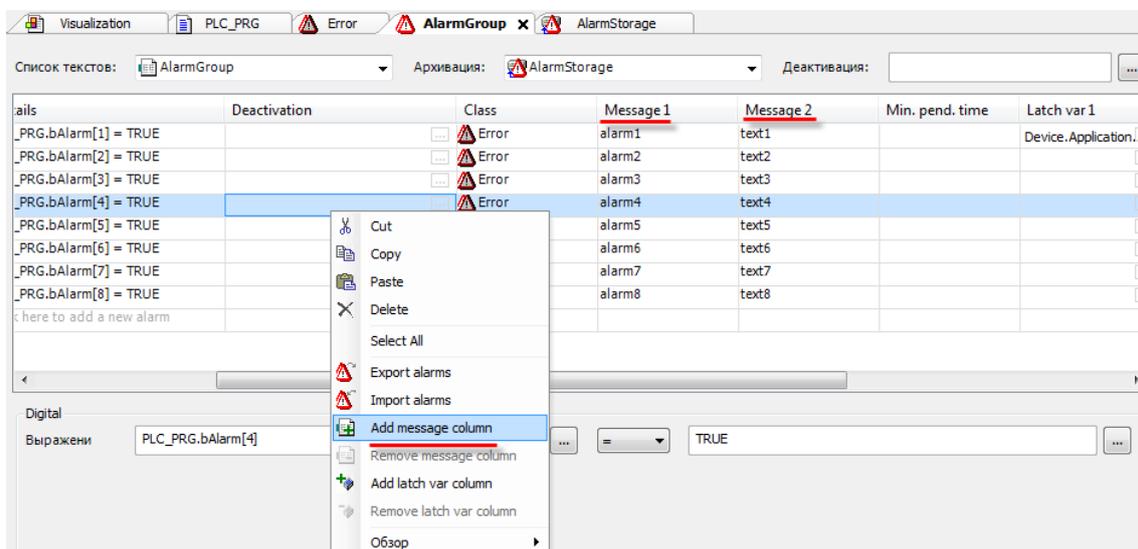


Рис. 13.11. Добавление второго столбца сообщений для группы тревог **AlarmGroup**

В **Таблице тревог** появились дополнительные управляющие переменные:

Управляющие переменные	
Подтвердить выбранное	bAckSel
Подтвердить все видимое	bAckVisible
История	bHistory
Фиксировать позицию прокрутки	bFreezeScrollPos
<u>Count alarms</u>	PLC_PRG.dwCountAlarms
<u>Count visible rows</u>	PLC_PRG.dwCountVisibleRows
<u>Current scroll index</u>	PLC_PRG.dwCurrentScrollIndex

Рис. 13.12. Управляющие переменные **Таблицы тревог**

Имя параметра	Тип	Описание
Count alarms	DWORD	Число тревог в таблице. Только для чтения.
Count visible rows	DWORD	Число строк, которое может быть одновременно отображено в таблице (без учета прокрутки). Только для чтения.
Current scroll index	DWORD	Текущая позиция полосы прокрутки. Пока в таблице не отображается полоса прокрутки, переменная имеет значение 0 . После появления полосы прокрутки, перемещение на строку вниз/вверх увеличивает/уменьшает значение переменной на 1 . Меняя ее значение из программы, можно листать таблицу тревог без использования полосы прокрутки и ее кнопок.

13.2.4. Диалог ввода NumpadExtended

К диалогам ввода **Numpad** и **Keypad** добавился **NumpadExtended**, позволяющий вводить значения в шестнадцатеричном формате. Для отображения таких значений в визуализации используется [спецификатор %x](#).

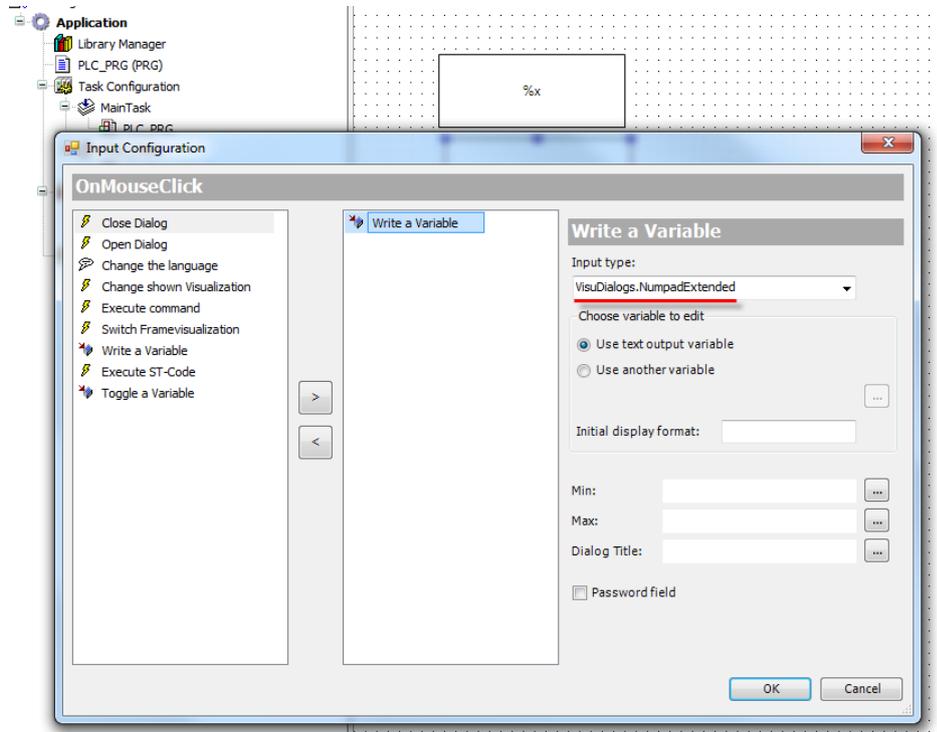


Рис. 13.13. Выбор диалога ввода **NumpadExtended**

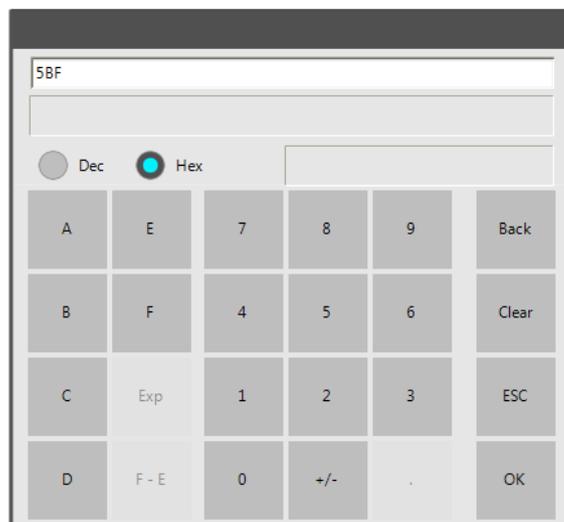


Рис. 13.14. Внешний вид диалога ввода **NumpadExtended**

13.3. CODESYS 3.5 SP9

13.3.1. Графический примитив Календарь

В SP9 добавлен новый графический примитив – **Календарь** (категория **Элементы управления датой/временем**). Он используется для установки значения переменной типа **DATE**. Дизайн элемента может быть настроен пользователем (фонное изображение, цвет выделения и т.д.).

< June 2016 >							< July 2016 >							< August 2016 >							< September 2016 >							
Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	
		1	2	3	4	5					1	2	3	1	2	3	4	5	6	7					1	2	3	4
6	7	8	9	10	11	12	4	5	6	7	8	9	10	8	9	10	11	12	13	14	5	6	7	8	9	10	11	
13	14	15	16	17	18	19	11	12	13	14	15	16	17	15	16	17	18	19	20	21	12	13	14	15	16	17	18	
20	21	22	23	24	25	26	18	19	20	21	22	23	24	22	23	24	25	26	27	28	19	20	21	22	23	24	25	
27	28	29	30				25	26	27	28	29	30	31	29	30	31					26	27	28	29	30			
< October 2016 >							< November 2016 >							< December 2016 >							< January 2017 >							
Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	
					1	2		1	2	3	4	5	6					1	2	3	4							1
3	4	5	6	7	8	9	7	8	9	10	11	12	13	5	6	7	8	9	10	11	2	3	4	5	6	7	8	
10	11	12	13	14	15	16	14	15	16	17	18	19	20	12	13	14	15	16	17	18	9	10	11	12	13	14	15	
17	18	19	20	21	22	23	21	22	23	24	25	26	27	19	20	21	22	23	24	25	16	17	18	19	20	21	22	
24	25	26	27	28	29	30	28	29	30					26	27	28	29	30	31	23	24	25	26	27	28	29		
31																					30	31						
< February 2017 >							< March 2017 >							< April 2017 >							< May 2017 >							
Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	
		1	2	3	4	5			1	2	3	4	5						1	2	1	2	3	4	5	6	7	
6	7	8	9	10	11	12	6	7	8	9	10	11	12	3	4	5	6	7	8	9	8	9	10	11	12	13	14	
13	14	15	16	17	18	19	13	14	15	16	17	18	19	10	11	12	13	14	15	16	15	16	17	18	19	20	21	
20	21	22	23	24	25	26	20	21	22	23	24	25	26	17	18	19	20	21	22	23	22	23	24	25	26	27	28	
27	28						27	28	29	30	31			24	25	26	27	28	29	30	29	30	31					

Рис. 13.14. Внешний вид элемента **Календарь**

13.3.2. Графический примитив Установка даты-времени

В SP9 добавлен новый графический примитив – **Установки даты-времени** (категория **Элементы управления датой/временем**). Он используется для установки значения переменной типа **DATE**, **DATE_AND_TIME**, **TIME**, **LTIME**, **TIME_OF_DAY**. Каждый разряд времени устанавливается с помощью диалога ввода, дата – с помощью [календаря](#). Дизайн элемента может быть настроен пользователем (фоновое изображение, цвет выделения и т.д.).

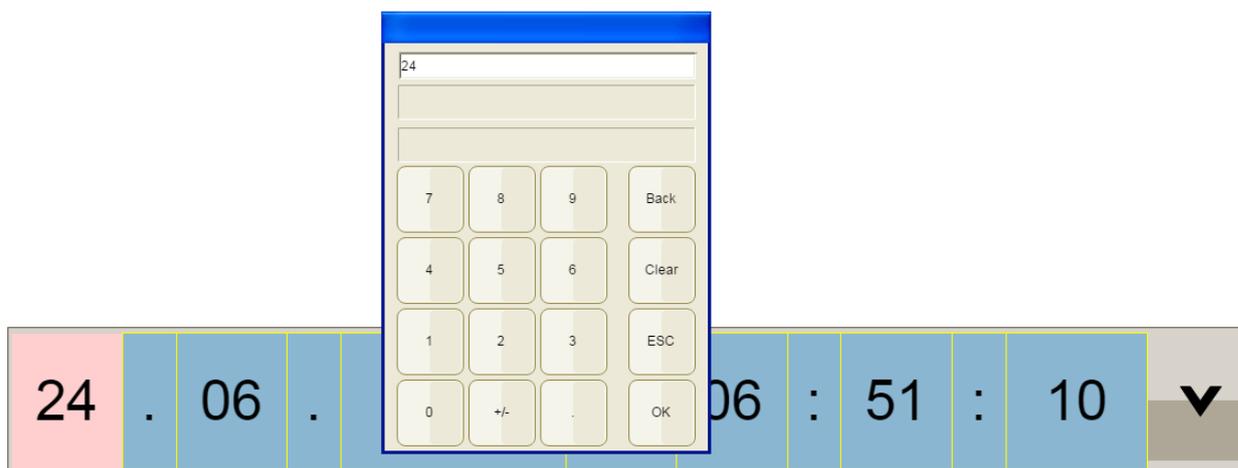


Рис. 13.15. Внешний вид элемента **Установка даты-времени**

13.3.3. Отображение текста перечислений

При создании перечислений появилась возможность поставить галочку **Поддержка списка текстов**, в результате чего при привязке экземпляров перечисления к графическим элементам, в них будут отображаться текстовые значения перечисления (без галочки – числовые).

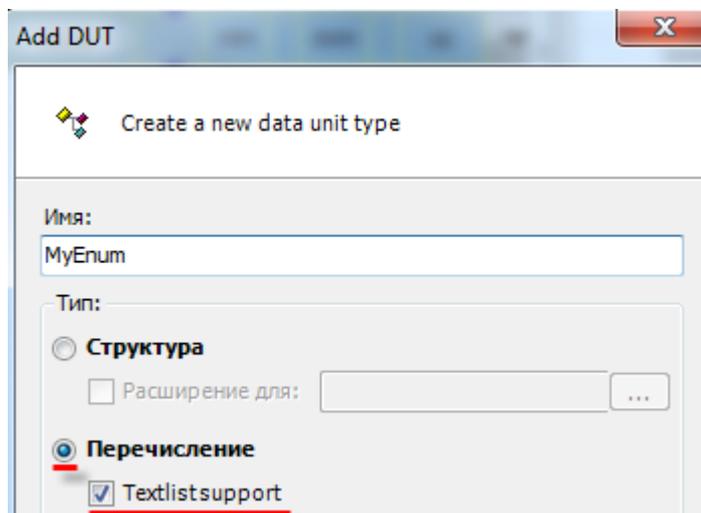


Рис. 13.16. Выбор поддержки списка текстов при создании перечисления

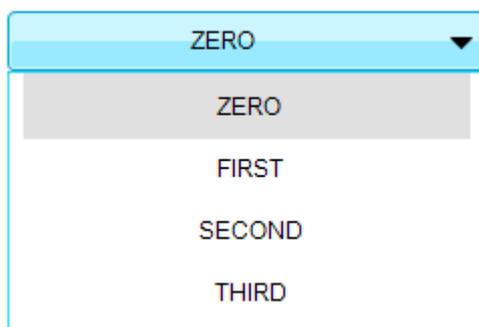


Рис. 13.17. Отображение перечисления в элементе [Комбинированное окно \(Целочисленное\)](#)

13.3.4. Выбор позиции и выделение в диалогах ввода

При работе с диалогами ввода появилась возможность установки курсора в произвольную позицию (в web-визуализации можно двигать курсор стрелками клавиатуры) и выделения части строки. Это может понадобиться при редактировании введенного текста.

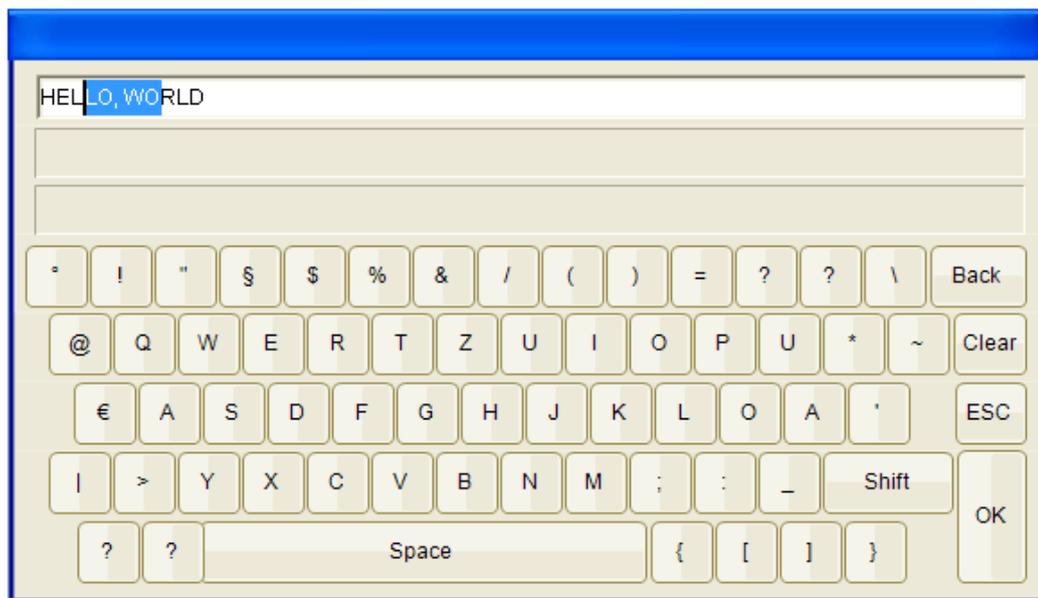


Рис. 13.18. Выделение части строки в диалоге ввода

13.3.5. Параметр Use real values

К [типичным параметрам](#) графических примитивов добавился параметр **Use real values** (вкладка **Абсолютное перемещение**). При наличии галочки появляется возможность привязать к параметрам данной вкладки переменные типа **REAL** – это может понадобиться при необходимости очень точного перемещения и вращения элементов.

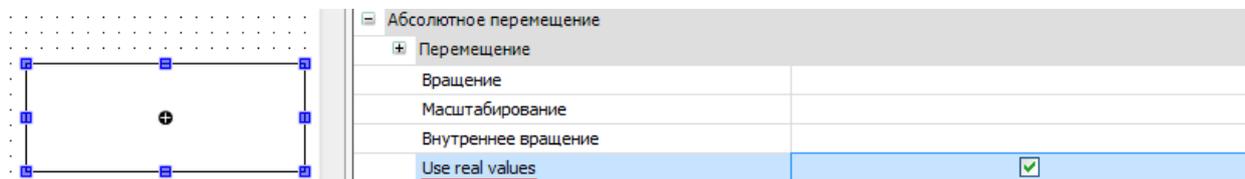


Рис. 13.19. Параметр **Use real values**

13.3.6. Локализация названий месяцев и дней

При добавлении в проект элемента [Календарь](#) или [Установка даты-времени](#), а также при использовании [спецификатора %t\[формат времени\]](#) в проект автоматически добавляется список текстов **System** (на вкладке **POU**), который содержит названия месяцев и дней. По умолчанию используются англоязычные термины, но пользователь может поменять их на свои – в т.ч., и на русскоязычные.



The screenshot shows the POU application interface. On the left, a tree view under 'POU' contains 'GlobalTextList', 'Менеджер библиотек', 'System' (highlighted), and 'Установки проекта'. On the right, the 'System' tab displays a table with two columns: 'ID' and 'Default'. The table lists 24 entries, each representing a month or day of the week in English, with the same English word in the 'Default' column.

ID	Default
Apr	Apr
April	April
Aug	Aug
August	August
Dec	Dec
December	December
Feb	Feb
February	February
Fri	Fri
Friday	Friday
Jan	Jan
January	January
Jul	Jul
July	July
Jun	Jun
June	June
Mar	Mar
March	March
May	May
May_3Letter	May
Mon	Mon
Monday	Monday
Nov	Nov
November	November
Oct	Oct
October	October
Sat	Sat
Saturday	Saturday
Sep	Sep
September	September
Sun	Sun
Sunday	Sunday
Thu	Thu
Thursday	Thursday
Tue	Tue
Tuesday	Tuesday
Wed	Wed
Wednesday	Wednesday

Рис. 13.20. Список текстов **System**

13.3.7. Текстовое поле

У элемента [Текстовое поле](#) появилась вкладка параметров **Selection and caret configuration**, которая позволяет определить параметры текущего выделения сегмента текста элемента, а также управлять этим выделением:

Selection and caret configuration	
Caret position	PLC_PRG.iCaretPosition
Selection start	PLC_PRG.iSelectionStart
Selection end	PLC_PRG.iSelectionEnd
All selected	PLC_PRG.xAllSelected

Рис. 13.21. Вкладка параметров **Selection and caret configuration** элемента **Текстовое поле**

Имя параметра	Тип	Описание
Caret position	INT	Позиция курсора. 0 – текст не выделен, 1 – курсор перед первым символом, 2 – курсор перед вторым символом и т.д.
Selection Start	INT	Номер первого символа выделенного сегмента текста. При значении 0 выделение отсутствует.
Selection End	INT	Номер последнего символа выделенного сегмента текста. При значении 0 выделение отсутствует.
All Selected	BOOL	При значении TRUE выделяется весь текст элемента независимо от значений предыдущих параметров. Во всех остальных случаях принимает значение FALSE .



Watch 1			
Expression	Application	Type	Value
PLC_PRG	Device.Application	PLC_PRG	
iCaretPosition		INT	3
iSelectionStart		INT	3
iSelectionEnd		INT	7
xAllSelected		BOOL	FALSE
str		STRING	'This is test'

Рис. 13.22. Использование параметров вкладки **Selection and caret configuration**