CODESYS V3.5

Визуализация

Руководство пользователя

24.05.2022
версия 3.0

Оглавление

Глоссарий **2**

1 Цель документа **3**

2 Добавление визуализации в проект **4**

2.1 Добавление экрана визуализации в проект **4**

2.2 Свойства экрана визуализации **6**

2.3 Добавление диалога в проект **8**

3 Менеджер визуализации **9**

3.1 Настройки Менеджера визуализации **9**

3.1.1 Вкладка «Установки» 9

3.1.2 Вкладка «Установки диалога» 11

3.1.3 Вкладка «Горячие клавиши по умолчанию» 12

3.1.4 Вкладка «Визуализации» 12

3.1.5 Вкладка «Управление пользователями» 13

3.1.6 Вкладка «Установки списка текстов» 14

3.1.7 Вкладка «Advanced Settings» 14

3.2 Типы визуализаций **17**

3.2.1 Настройки таргет-визуализации 18

3.2.2 Настройки web-визуализации 19

3.2.3 Настройки удаленной таргет-визуализации 20

4 Стили и профили визуализации **21**

4.1 Стили визуализации **21**

4.2 Профили визуализации **23**

5 Редактор визуализации **24**

5.1 Интерфейс редактора визуализации **24**

5.2 Добавление элемента на экран визуализации. Взаимодействие с элементами **25**

5.3 Команды редактора визуализации **26**

5.4 Вспомогательные редакторы экранов визуализации **30**

5.4.1 Редактор интерфейсов 30

5.4.2 Редактор горячих клавиш 31

5.4.3 Редактор списка визуальных элементов 31

5.5 Включение и отключение сетки **32**

6 Вспомогательные компоненты визуализации **33**

6.1 Пул изображений **33**

6.2 Список текстов **35**

7 Описание графических примитивов **37**

7.1 Структура главы 37
7.2 Базовая группа элементов 38
7.2.1 Прямоугольник, Скругленный прямоугольник, Эллипс. Типичные параметры графических примитивов 38
7.2.2 Линия 44
7.2.3 Полигон, Ломаная, Кривая Безье 45
7.2.4 Секторная диаграмма 46
7.2.5 Изображение 47
7.2.6 Фрейм элемента 49
7.3 Стандартная группа элементов 51
7.3.1 Метка 51
7.3.2 Комбинированное окно (Целочисленное) 51
7.3.3 Комбинированное окно (Массив) 53
7.3.4 Набор вкладок 55
7.3.5 Кнопка 57
7.3.6 Группа 59
7.3.7 Таблица 60
7.3.8 Текстовое поле 63
7.3.9 Полоса прокрутки 64
7.3.10 Бегунок 65
7.3.11 Управление вращением 67
7.3.12 Невидимый ввод 68
7.3.13 Радио-кнопка 68
7.3.14 Кнопка-флажок 70
7.3.15 Индикатор выполнения 71
7.4 Элементы управления измерением 72
7.4.1 Отображение линейки 72
7.4.2 Стрелочный индикатор (90°/180°/360°) 75
7.4.3 Потенциометр 80
7.4.4 Гистограмма 81
7.5 Индикаторы/Переключатели/Изображения 85
7.5.1 Переключатель изображения 85
7.5.2 Индикатор 87
7.5.3 Переключатели/Выключатели 89
7.6 Специальные элементы управления 92
7.6.1 Трассировка 92
7.6.2 Тренд 101
7.6.3 Легенда 106
7.6.4 Элемент ActiveX 107
7.6.5 WebBrowser 108

7.6.6 Символ ожидания Кубик/Цветок 109
7.6.7 Текстовый редактор 110
7.6.8 Путь 3D/ControlPanel 113
7.7 Элементы управления датой/временем 114
7.7.1 Селектор диапазона дат 114
7.7.2 Селектор времени 115
7.7.3 Аналоговые часы 117
7.7.4 Элемент выбора даты 118
7.7.5 Элемент выбора даты и времени 119
7.8 Менеджер тревог 121
7.8.1 Компонент Конфигуратор тревог 121
7.8.2 Таблица тревог 129
7.8.3 Баннер тревог 134
7.9 Остальные элементы 135
7.9.1 Декартовый график XY 135
8 Отображение значений переменных. Форматы вывода и спецификаторы 142
8.1 Отображение значений переменных 142
8.2 Спецификаторы формата вывода переменных 143
8.3 Управляющие последовательности для STRING переменных 144
8.4 Системное время в визуализации 145
8.5 Пример отображения значения переменной 146
9 Привязка действий к элементам визуализации 148
9.1 Вкладка InputConfiguration 148
9.2 Действия для OnMouse<Something> 151
9.2.1 Управление пользователями 152
9.2.2 Закрыть диалог 154
9.2.3 Открыть диалог 155
9.2.4 Изменить язык 156
9.2.5 Изменить отображаемую визуализацию 157
9.2.6 Выполнить команду 158
9.2.7 Переключить визуализацию в фрейме 160
9.2.8 Записать переменную 161
9.2.9 Выполнить ST-код 163
9.2.10 Переключить переменную 164
9.2.11 Передача файлов 165
10 Примеры 166
10.1 Структура главы. Запуск примеров на виртуальном контроллере 166
10.2 Примеры работы с графическими примитивами 168

10.2.1 Типичные параметры графических примитивов ... 168
10.2.2 Фрейм ... 200
10.2.3 Комбинированное окно ... 209
10.2.4 Кнопка ... 216
10.2.5 Таблица ... 223
10.2.6 Отображение линейки и Потенциометр ... 235
10.2.7 Гистограмма ... 241
10.2.8 Индикаторы/Переключатели/Изображения ... 248
10.2.9 Трассировка ... 254
10.2.10 Тренд ... 262
10.2.11 WebBrowser ... 275
10.2.12 Текстовый редактор ... 281
10.2.13 Таблица тревог ... 304
10.3 Примеры решения типовых задач ... 321
10.3.1 Переключение экранов в проекте ... 321
10.3.2 Использование диалогов ... 330
10.3.3 Использование интерфейса фрейма ... 337
10.3.4 Создание анимации (элемент Изображение) ... 346
10.3.5 Создание мультиязычного проекта ... 355
10.3.6 Тиражирование элементов ... 367
10.3.7 Управление пользователями. Парольный доступ ... 370
10.3.8 Менеджер рецептов ... 379
10.3.9 Конвертация значений (Unit Conversion) ... 403
10.4 Дополнительные примеры ... 412
10.4.1 Использование русскоязычной клавиатуры (библиотека OwenVisuDialogs) ... 412
10.4.2 Работа с динамическими точками (элемент Ломаная) ... 413
10.4.3 Считывание координат курсора ... 420
10.4.4 Многопользовательский доступ. Независимая обработка клиентов визуализации ... 426
10.4.5 Дополнительные графические библиотеки ... 431
10.4.6 Кастомизация диалогов ввода ... 432
10.4.7 Дополнительные примеры от компании CODESYS Group ... 435
10.4.8 Очистка истории трендов и тревог из кода программы ... 436
11 Визуализация контроллеров ОВЕН: ограничения и рекомендации ... 437
11.1 Размер экранов визуализации ... 437
11.2 Отображение кириллического текста в визуализации ... 437
11.3 Тип ввода по умолчанию ... 438
11.4 Ограничения ... 439
11.5 Поддерживаемые форматы графических файлов ... 439
11.6 Время цикла визуализации ... 439

11.7 Поддерживаемые шрифты для контроллеров СПК 440
11.8 Управление яркостью подсветки для контроллеров СПК.....443
11.9 Изменение загрузочного логотипа для контроллеров СПК.....443
11.10 Сохранение истории трендов и трендов на USB/SD-накопитель 444

Глоссарий

POU – обобщенное название функций, функциональных блоков и программ.

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 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 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 кнопка мыши/правая кнопка мыши.

1 Цель документа

Настоящее руководство описывает создание экранов визуализации для контроллеров, программируемых в среде **CODESYS V3.5,** с подробным описанием характеристик и настроек всех графических примитивов, а также примерами работы с ними.

ПРИМЕЧАНИЕ
В зависимости от конкретного устройства часть описанного в документе функционала может не поддерживаться.

Руководство предназначено для начинающих и опытных пользователей, подразумевается, что читатель обладает базовыми навыками работы с **CODESYS**.

Во время написания руководства использовалась среда программирования **CODESYS V3.5 SP17 Patch 3**; в более ранних или поздних версиях некоторые компоненты могут отсутствовать или отличаться от приведенных на скриншотах.

2 Добавление визуализации в проект

2.1 Добавление экрана визуализации в проект

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

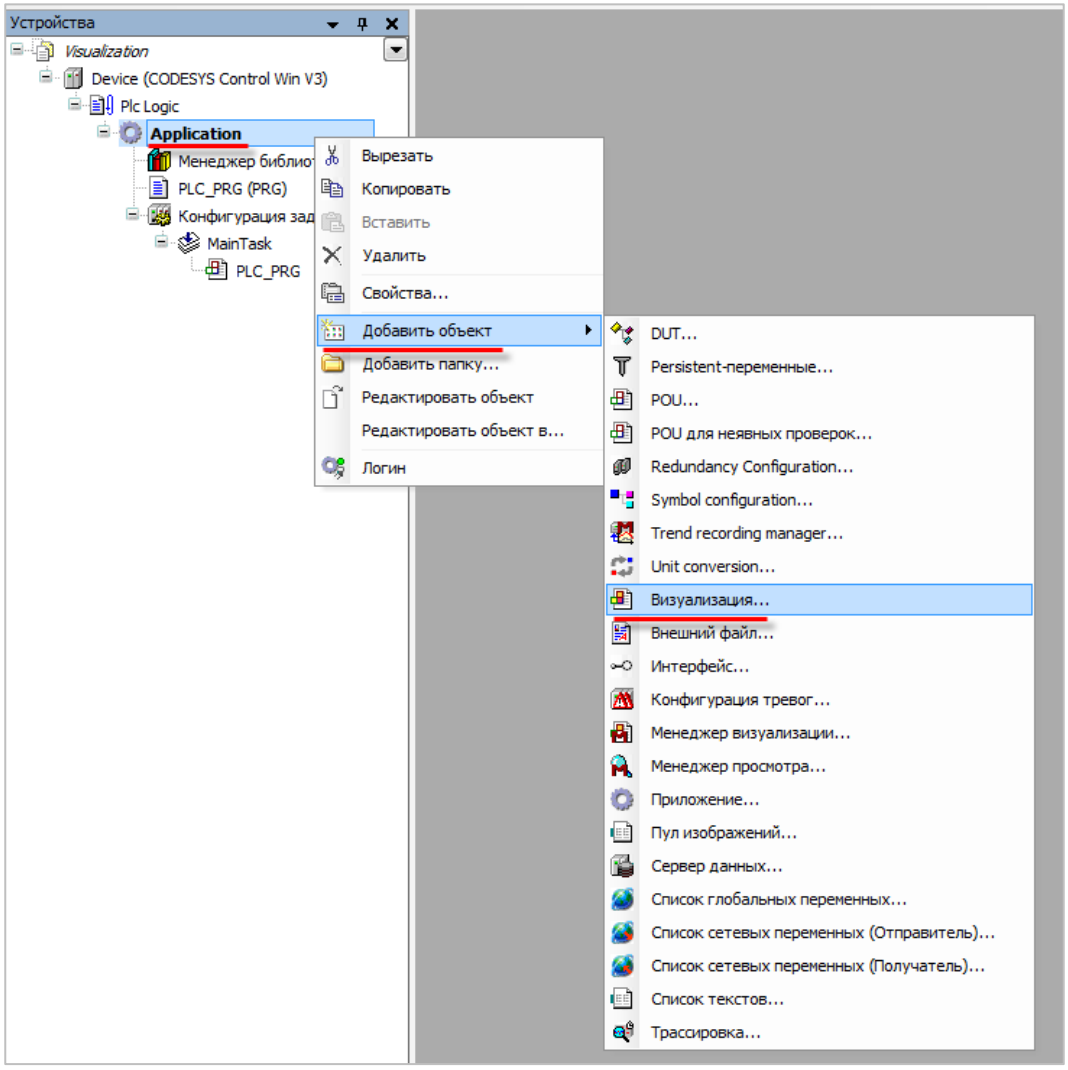


Рисунок 2.1 – Добавление экрана визуализации в проект

Во время создания экрана визуализации следует 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:

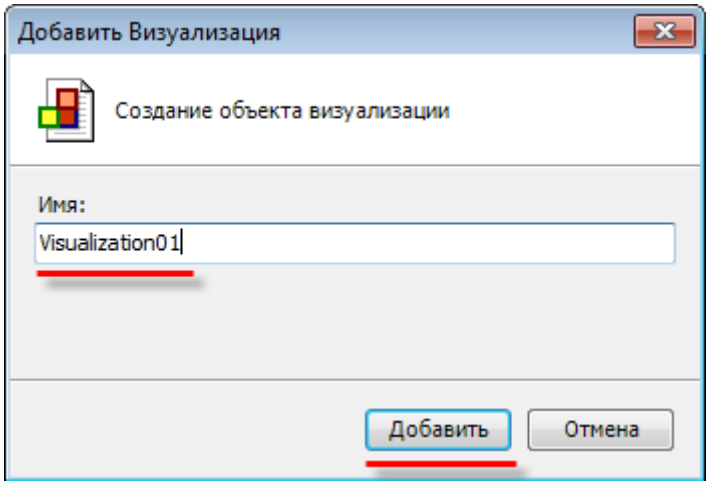


Рисунок 2.2 – Ввод имени экрана визуализации

Одновременно с созданием в проекте первого экрана визуализации, будет автоматически добавлен компонент Менеджер визуализации, содержащий подкомпоненты Таргет-визуализация и Web-визуализация.

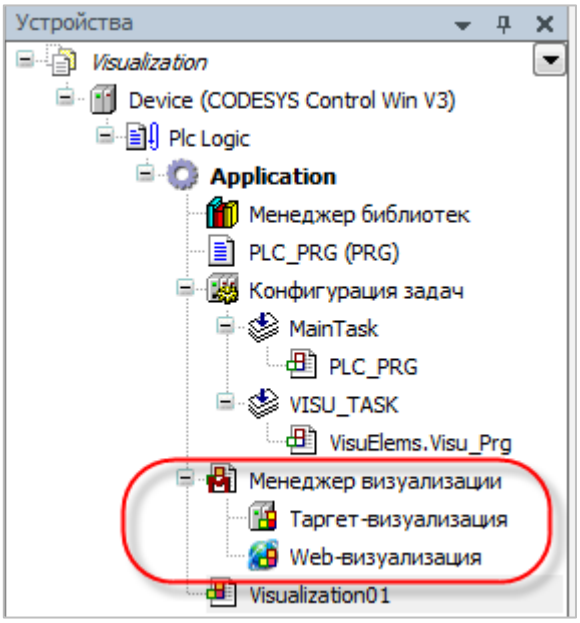


Рисунок 2.3 – Компонент Менеджер визуализации в Панели устройств проекта

2.2 Свойства экрана визуализации

Чтобы настроить тип и размер экрана визуализации, следует нажать **ПКМ** на его название в **Панели устройств** и в появившемся контекстном меню выбрать пункт **Свойства**:

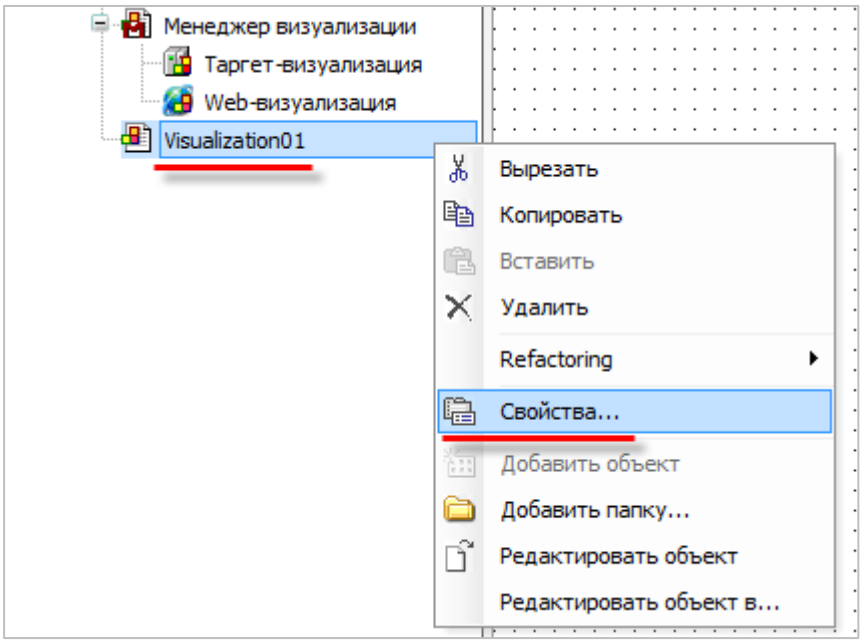


Рисунок 2.4 – Контекстное меню экрана визуализации

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

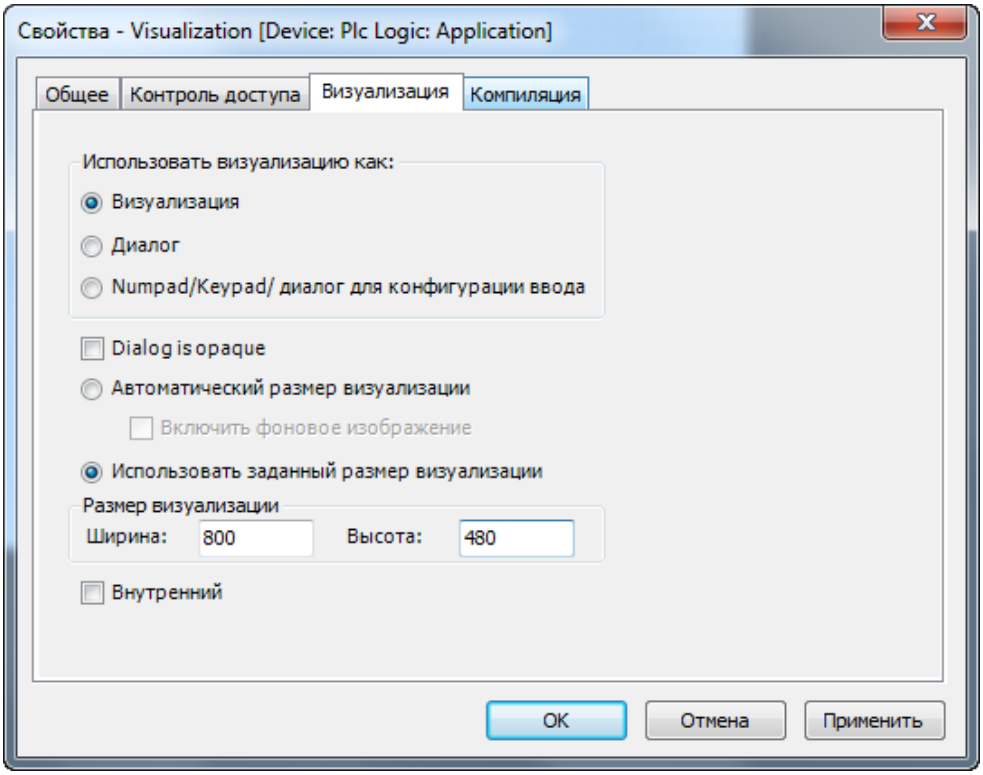


Рисунок 2.5 – Свойства экрана визуализации

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. Программный интерфейс диалога должен быть идентичен интерфейсам стандартных диалогов **Numpad/Keypad** из системной библиотеки **VisuDialogs.library** (расположенной в папке установки CODESYS по пути **…\CODESYS\Projects\Visu\Dialogs**).

Dialog is opaque – если тип визуализации – диалог и установлена данная галочка, то область визуализации, закрытая диалогом, не обновляется. Это повышает скорость отклика в данном диалоге.

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 следует установить галочку **Включить фоновое изображение**.

Использовать заданный размер визуализации – в данном случае пользователь сам определяет ширину и высоту экрана визуализации (в пикселях). В Менеджере визуализации можно указать, должен ли размер экрана визуализации подгоняться под размер экрана клиента визуализации.

Для контроллеров **СПК1xx, СПК1xx [M01]** и **СПК207** разрешение экрана составляет **800 × 480**.

Для контроллеров **СПК110 [M01]**, выпускаемых начиная с **декабря 2022 года** (с гравировкой **Версия 2.6 WSVGA** на задней крышке) – **1024x600**.

Для контроллеров **СПК105** – **480 × 272**.

3. Галочка **Внутренний** может использоваться для экранов визуализации, создаваемых в библиотеках. В случае подключения такой библиотеки к проекту, данный экран визуализации не будет доступен пользователю. Данная возможность применяется для создания вспомогательных экранов, которые не должны быть доступны пользователю напрямую.

2.3 Добавление диалога в проект

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

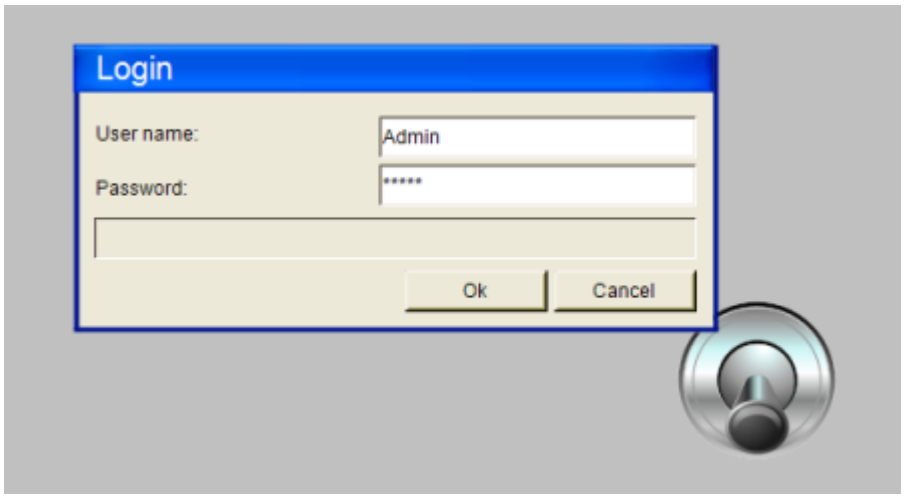


Рисунок 2.6 – Диалоговое окно ввода пароля

Для использования экрана визуализации в качестве диалога, следует в его **Свойствах** в пункте **Использовать визуализацию как** выбрать значение **Диалог** (см. рисунок 2.5).

3 Менеджер визуализации

3.1 Настройки Менеджера визуализации

Компонент **Менеджер визуализации** содержит настройки, применяемые ко всем экранам визуализации в проекте. Компонент добавляется в проект автоматически во время создания первого экрана визуализации.

3.1.1 Вкладка «Установки»

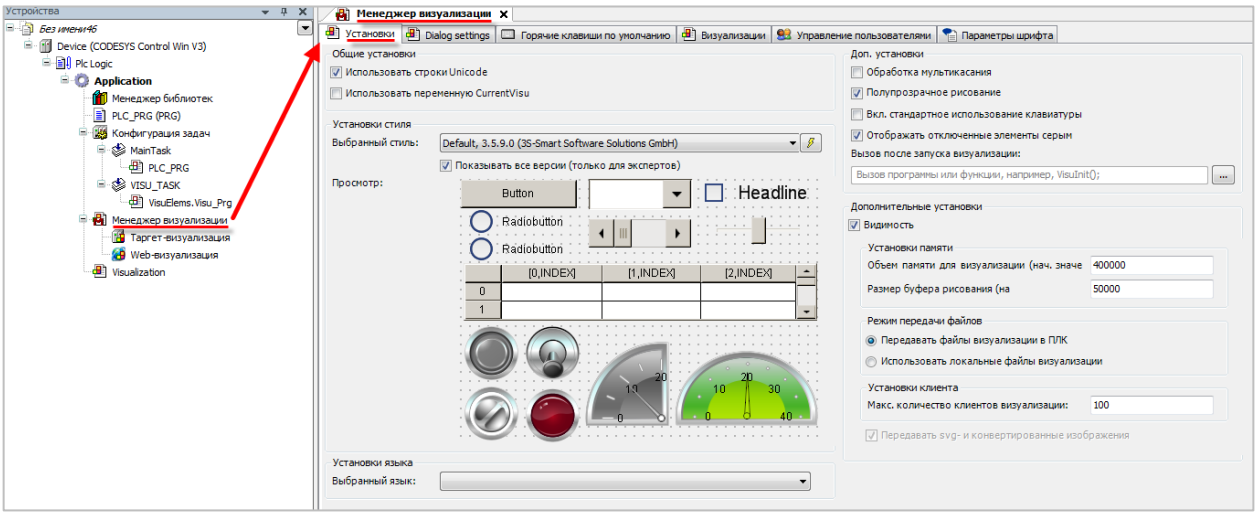


Рисунок 3.1 – Менеджер визуализации, вкладка Установки

Описание настроек приведено в таблице 3.1.

Таблица 3.1 – Установки Менеджера визуализации

Общие установки	
Использовать строки Unicode	Если установлена галочка, то статический текст, отображаемый в визуализации, преобразуется в формат **Юникода**, что позволяет отображать статические кириллические и иероглифические надписи
Использовать переменную CurrentVisu	Если установлена галочка, то появляется возможность использования системной строковой (**STRING**) переменной **VisuElems.CurrentVisu,** которая определяет, какой из экранов визуализации отображается в данный момент для всех клиентов визуализации (в случае установки галочки визуализация всех клиентов синхронизируется). Соответственно, записывая в нее названия экранов визуализации, можно осуществлять переключения между ними. Пример использования приведен в п. 10.3.1
Поддержка клиентских анимаций и наложение исходных элементов	Данная галочка доступна только в случае отсутствия в проекте таргет-визуализации. При установке галочки включается поддержка технологии **overlay**. Эти приводит к следующим изменениям: **1**. В настройках всех элементов визуализации появляется вкладка параметров **Абсолютное перемещение** (без установки галочки у некоторых элементов она отсутствует) **2**. В настройках всех элементов визуализации появляются параметры **Длительность анимации** и **Переместить на передний план** **3**. Открытие/закрытие диалогов, смена экранов фреймов, изменение видимости элементов и их перемещение с помощью параметров вкладки

	Абсолютное перемещение происходит не «мгновенно», а «плавно» (см. ниже ссылку на видео). Анимация обрабатывается на стороне клиента визуализации независимо от цикла задачи **VISU_TASK** **4**. В элементе Изображение появляется возможность использования анимированных изображений (.gif и .svg) См. видео с демонстрацией описанного функционала
Установки стиля	
Выбранный стиль	Позволяет указать стиль визуализации (см. п. 5.1)
Установки языка	
Выбранный язык	Позволяет указать язык динамических текстов, отображаемый по умолчанию во время запуска проекта (для мультиязычных проектов)
Доп. установки	
Мультикасание	Если клиент визуализации поддерживает возможность мультикасания, то для его использования следует активировать данную настройку
Scroll elements with scrollbar	Данная настройка доступна только если установлена галочка **Мультикасание**. В случае установки этой галочки для элементов, поддерживающих прокрутку (например, таблица и Combobox), будет отображаться полоса прокрутки. В случае отсутствия галочки полоса прокрутки у таких элементов отображаться не будет – прокрутка должна будет осуществляться жестами
Полупрозрачное рисование	В случае наличия галочки поддерживается управление прозрачностью элементов визуализации
Вкл. стандартное использование клавиатуры	В случае наличия галочки для переключения между элементами и работе с ними поддерживается стандартный набор клавиш (Tab, Shift+Tab, Enter, Up arrow, Down arrow, Right arrow, Left arrow)
Отображать отключенные элементы серым	В случае наличия галочки деактивированные через параметр **Отключение ввода** элементы отображаются серым
Вызов после запуска визуализации	В данной вкладке можно выбрать POU, который будет однократно выполнен при старте задачи визуализации

3.1.2 Вкладка «Установки диалога»

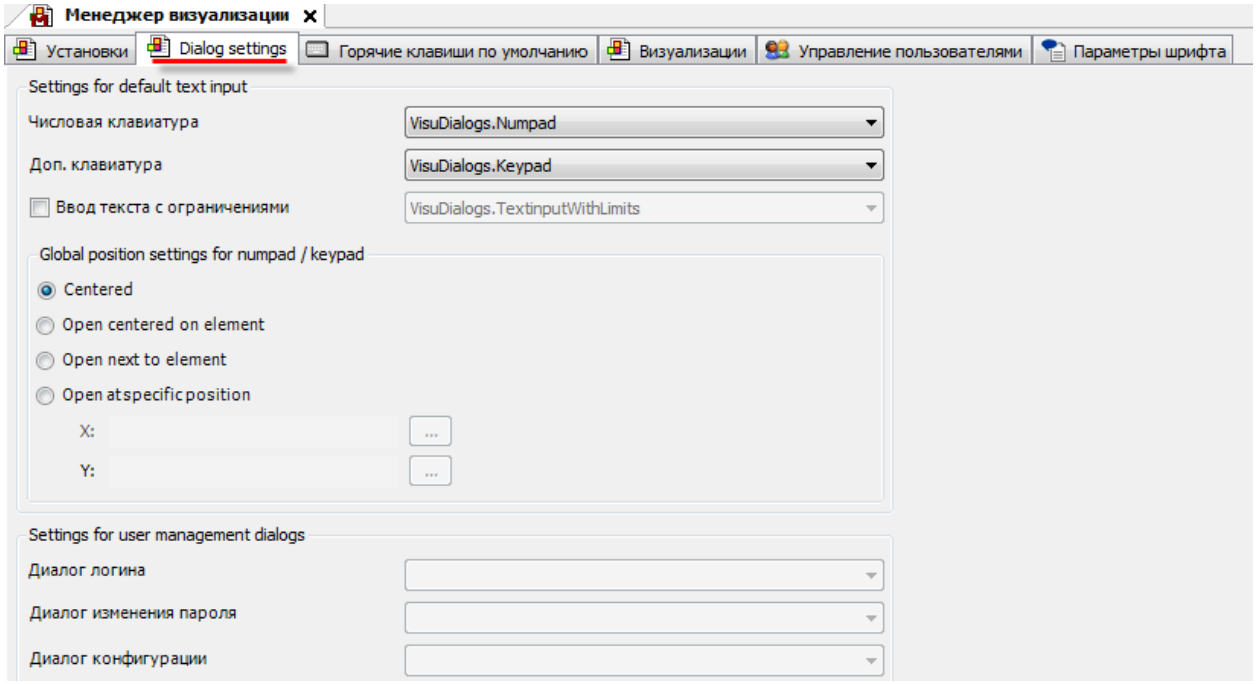


Рисунок 3.2 – Менеджер визуализации, вкладка Dialog Settings

В данной вкладке можно указать диалоги ввода по умолчанию. Для создания библиотек пользовательских диалогов следует использовать исходные коды библиотек **VisuDialogs.library** и **VisuUserMgmtDialogs.library,** расположенные в папке установки CODESYS по пути **…\CODESYS\Projects\Visu\Dialogs**). Для выбора пользовательских диалогов данные библиотеки следует установить в репозиторий и добавить в проект CODESYS. См. также библиотеку диалогов OwenVisuDialogs.

Также можно указать позицию по умолчанию для открытия диалогов **Numpad/Keypad**:

- **Centered** – диалоги открываются по центру экрана визуализации;
- **Open centered on element** – диалоги открываются поверх вызвавшего их элемента;
- **Open next to element** – диалоги открываются рядом с вызвавшим их элементом;
- Open at specific position – диалоги открываются по заданным координатам X/Y. Координаты задаются с помощью переменных типа **VisuElems.CmpVisuHandler.VisuStructPoint.iX**, **VisuElems.CmpVisuHandler.VisuStructPoint.iY**.

3.1.3 Вкладка «Горячие клавиши по умолчанию»

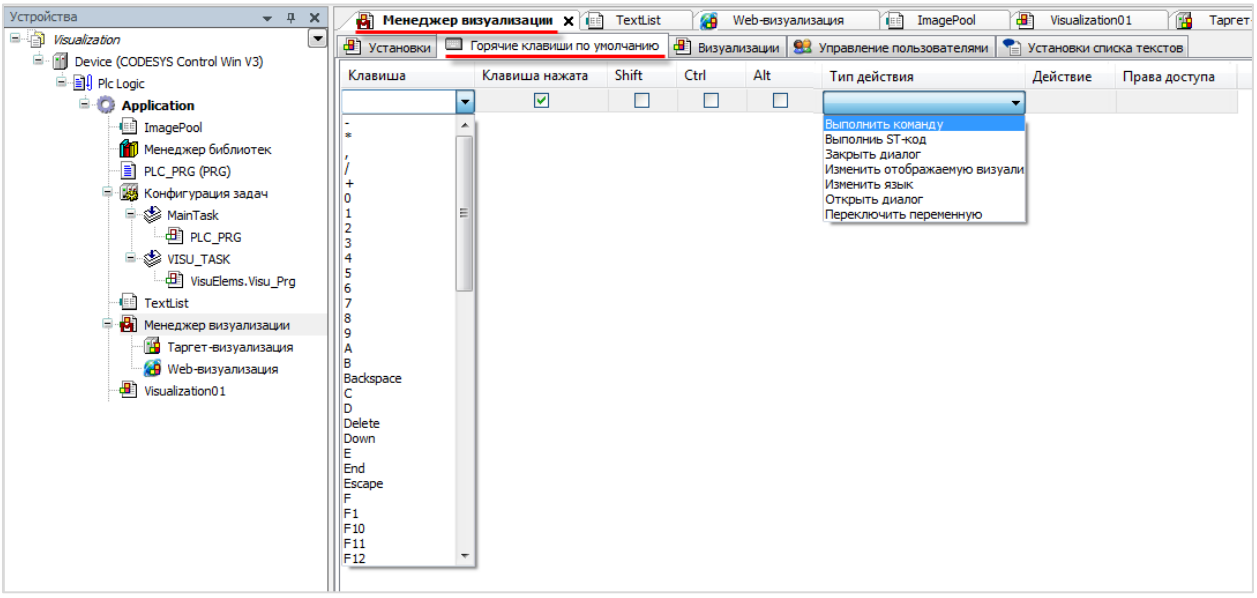


Рисунок 3.3 – Менеджер визуализации, вкладка Горячие клавиши по умолчанию

В данной вкладке можно привязать к нажатию клавиш выполнение действий. Для этого целевое устройство должно иметь соответствующие кнопки. Если галочка **Клавиша нажата** не установлена, то действие будет выполняться в случае отпускания зажатой клавиши. Также можно создавать комбинации клавиш с использованием **Shift/Ctrl/Alt**. Для клавиш и их комбинаций можно настраивать права доступа (в случае наличия в проекте Управления пользователями) – т. е. разрешить или запретить определенным пользователям использование определенных клавиш.

3.1.4 Вкладка «Визуализации»

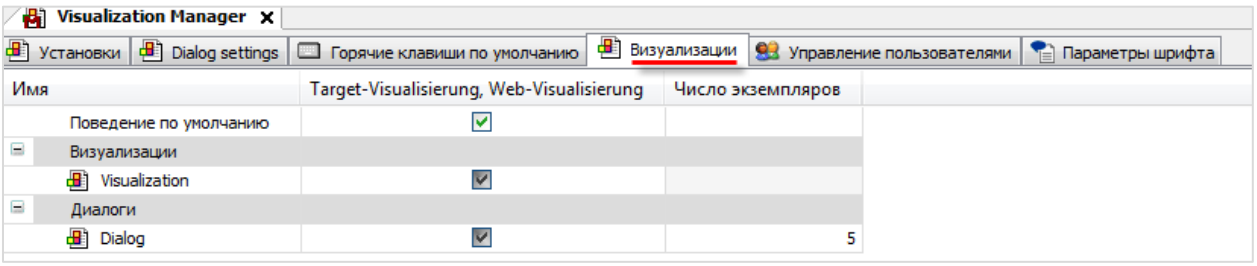


Рисунок 3.4 – Менеджер визуализации, вкладка Визуализации

В данной вкладке расположена таблица всех экранов визуализации для всех типов визуализаций, используемых в проекте. Галочка позволяет отключить из компиляции конкретные экраны (например, если они были нужны только на этапе отладки). При нажатии **ПКМ** на любое место таблицы можно использовать команду **Deactivate Unused Visualizations** для автоматического отключения всех визуализаций, не используемых в проекте (т. е. экранов, на которых нет перехода, и диалогов, которые не вызываются в визуализации). Для диалогов можно указать максимальное число экземпляров, которое может быть открыто одновременно.

3.1.5 Вкладка «Управление пользователями»

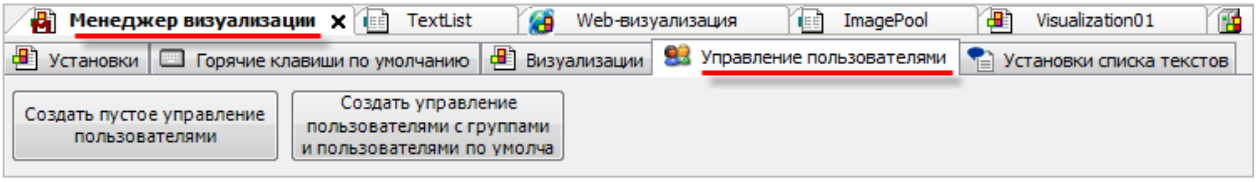


Рисунок 3.5 – Менеджер визуализации, вкладка Управление пользователями

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

Шаблон управления пользователям имеет следующий вид:

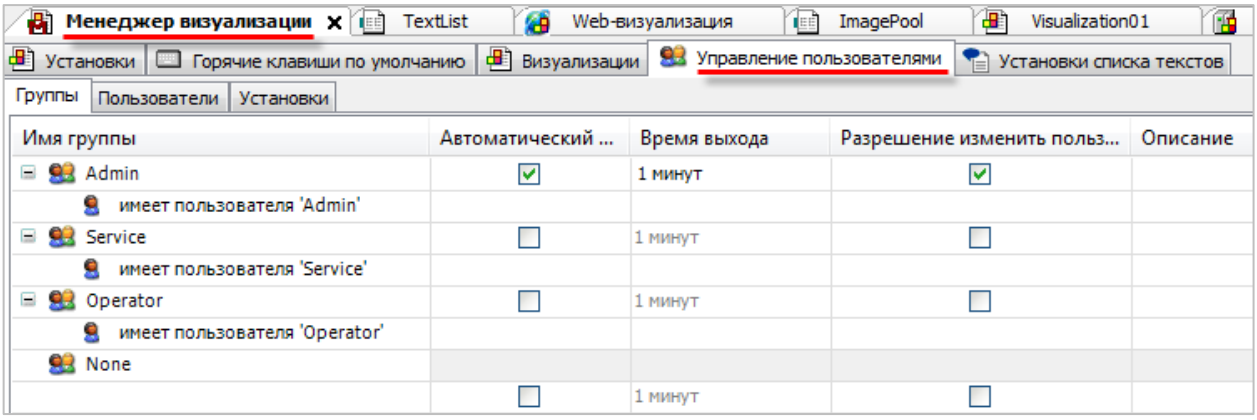


Рисунок 3.6 – Менеджер визуализации, шаблон Управления пользователями

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

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

Списки групп и пользователей могут быть экспортированы и импортированы в виде файлов формата **.csv**.

Во вкладке **Установки** задаются параметры загрузки информации пользователей во время загрузки проекта на целевое устройство (**Всегда/Никогда/Решение при каждой загрузке**), также можно настроить ограничение на число попыток ввода, режим блокировки в случае превышения данного числа (временно/постоянно) и переключение на стартовый экран визуализации в случае выхода пользователя.

Пример создания проекта с **Управлением пользователями** приведен в п. 10.3.7.

3.1.6 Вкладка «Установки списка текстов»

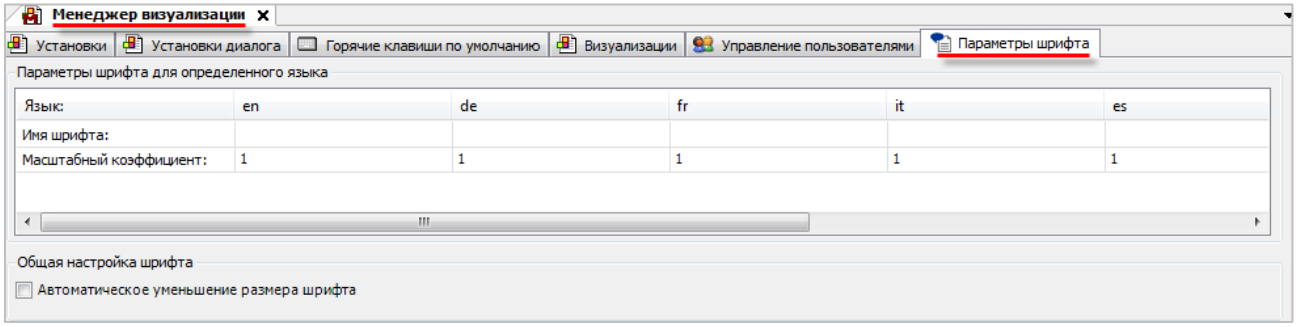


Рисунок 3.7 – Менеджер визуализации, вкладка Установки списков текстов

На данной вкладке можно настроить шрифт и масштабный коэффициент для каждого языка, используемого в визуализации (для мультиязычных визуализаций). Пример создания мультиязычного проекта приведен в п. 10.3.5.

Если установлена галочка **Автоматическое уменьшение размера шрифта**, то шрифт текстов, которые выходят за границы элементов визуализации, будет автоматически уменьшен.

3.1.7 Вкладка «Advanced Settings»

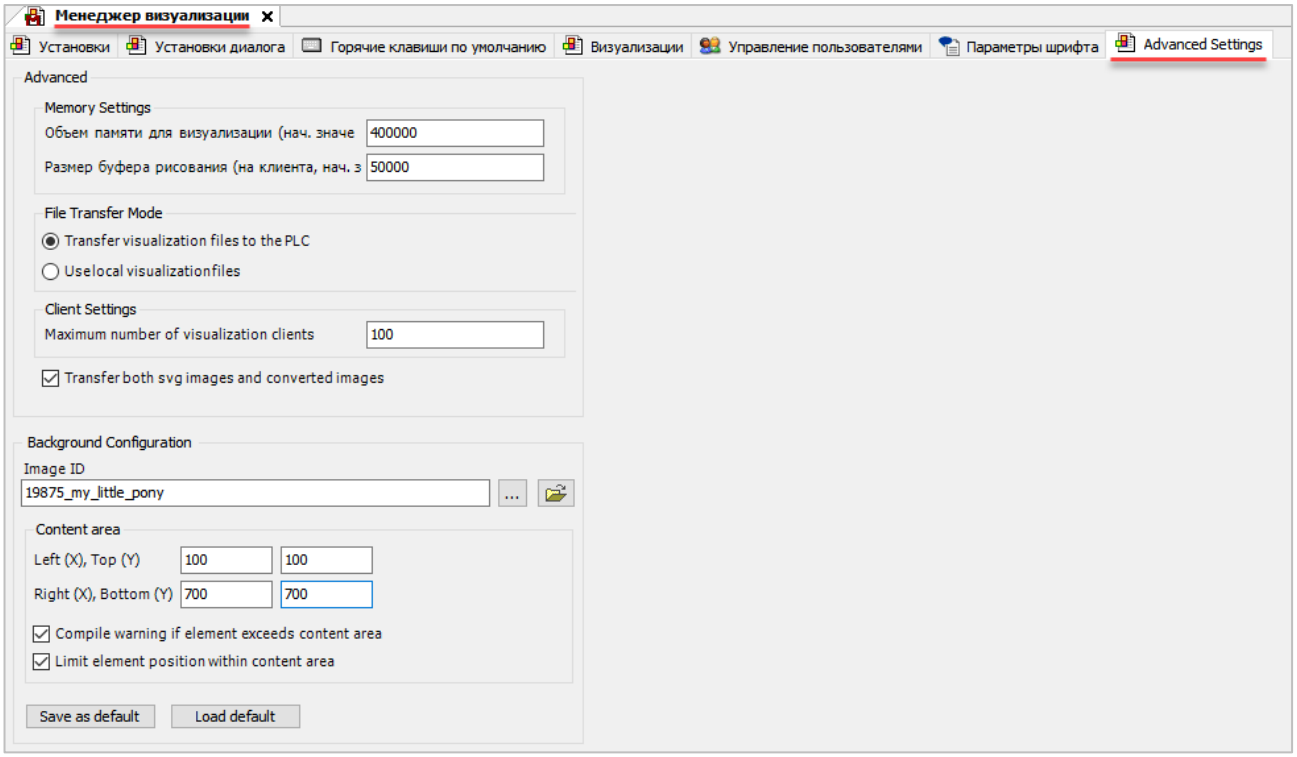


Рисунок 3.8 – Advanced Settings

На данной вкладке задаются специфические настройки визуализации и настройки фонового изображения.

Таблица 3.2 – Параметры вкладки Advanced Settings

Дополнительные установки	
Объем памяти визуализации (нач. значение)	Начальный объем сегмента памяти (в байтах), выделяемый для визуализации. Если значение будет превышено – то динамически будет выделено в два раза больше памяти. Значение по умолчанию – 400000 байт
Буфер рисования (на клиента, нач. значение)	Начальный объем сегмента памяти (в байтах), выделяемый для операций отрисовки одного клиента визуализации. Если значение будет превышено – то динамически будет выделено в два раза больше памяти. 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** на вкладке **Визуализация**
Макс. количество клиентов	Позволяет указать максимальное количество клиентов визуализации (все последующие клиенты не смогут подключиться к визуализации). Значение по умолчанию – **100**. Это значение может быть считано в коде из глобальной константы **VISU_MAX_NUMBER_OF_CLIENTS**
Передавать svg- и конвертированные изображения	В случае наличия галочки изображения формата **.svg** с неподдерживаемыми настройки при загрузке проекта автоматически конвертируются в **.png**. В web-визуализации отображаются исходные .**svg** изображения, в таргет-визуализации – **.png** изображения
Background Configuration	
Image ID	Выбор картинки, которая будет использоваться в качестве фонового изображения для всех экранов визуализации проекта
Content Area	Координаты рабочей области (относительно фонового изображения), в которой можно будет располагать элементы визуализации. Пример: фоновое изображение имеет размер **800x480**, настройки content area – **100/100/500/500**. В результате (см. рис. 3.9): • изображение будет смещено относительно рабочего поля редактора визуализации. Верхняя левая точка изображения в редакторе фактически имеет координаты (**-100, -100**); • рабочая область, в которой можно размещать элементы, представлена на скриншоте белым прямоугольником (только для наглядности) и имеет размеры **400x400** (**500 – 100 = 400**)
Compile warning if element exceeds content area	Включение генерации предупреждений компилятора для тех случаев, когда элементы визуализации выходят за пределы рабочей области.
Limit element position within content area	Запрет размещения элементов визуализации за пределами рабочей области (редактор не позволит пользователю это сделать с помощью невидимых границ при размещении элементов)
Save as default	Сохранение текущих настроек фонового изображения в качестве настроек по умолчанию
Load Default	Сброс настроек фонового изображения в значения по умолчанию

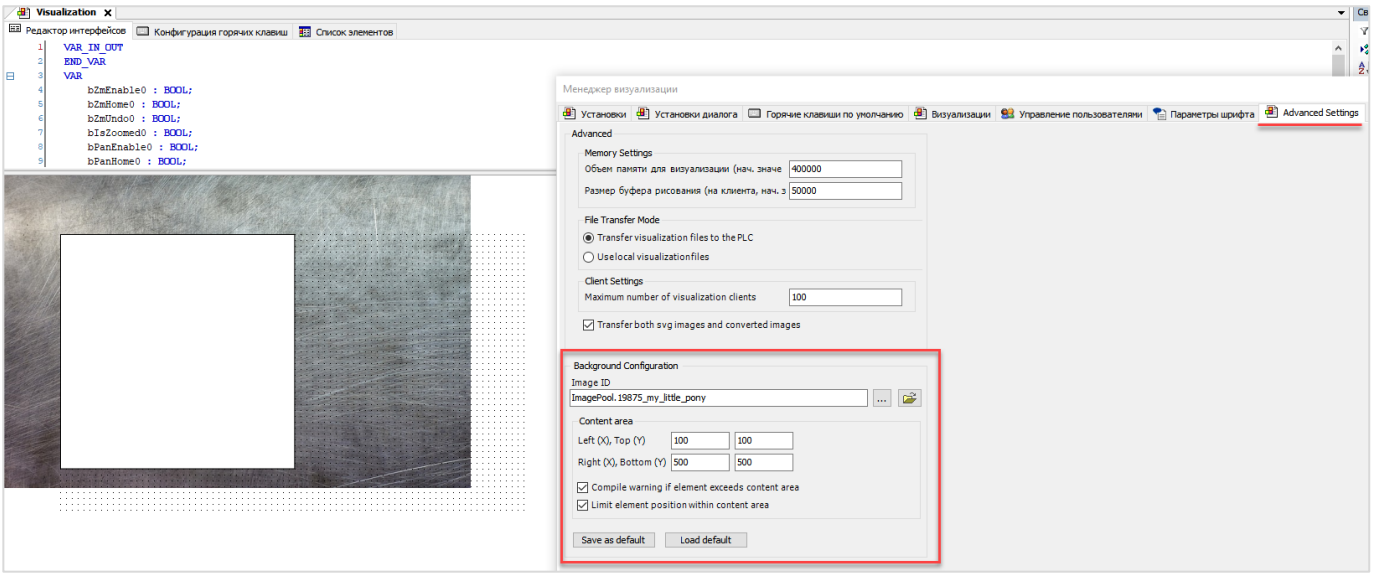


Рисунок 3.9 – Влияние настроек content area на экран визуализации

3.2 Типы визуализаций

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

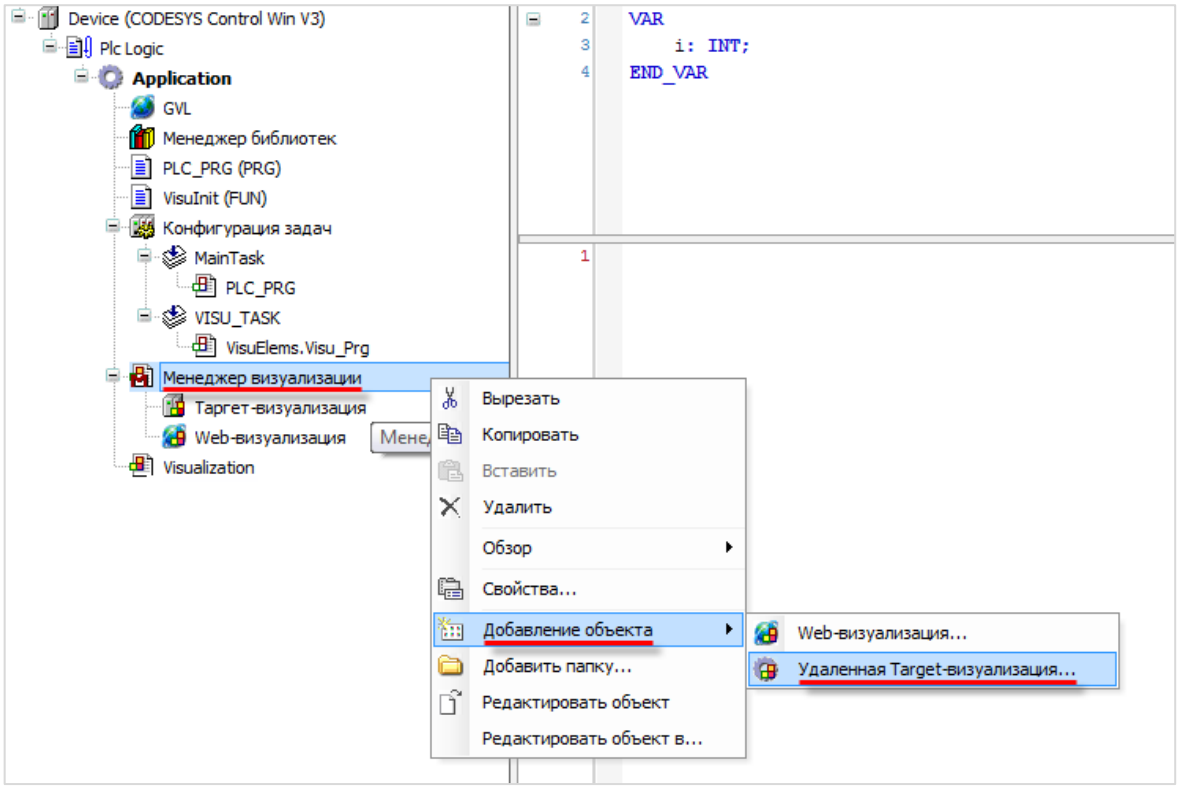


Рисунок 3.10 – Добавление новых типов визуализации в проект

Таргет-визуализация – это визуализация, отображаемая на целевом устройстве (например, панельном контроллере). Каждый проект может содержать только одну таргет-визуализацию. Настройки таргет-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.

Удаленная таргет-визуализация – это визуализация, отображаемая на ПК, подключенном к целевому устройству. Целевое устройство должно иметь таргет-визуализацию и лицензию на удаленную таргет-визуализацию (контроллеры ОВЕН такой лицензии не имеют).

Также существует **сервисная визуализация**, отображаемая в **Редакторе визуализации CODESYS** во время подключения к целевому устройству с запущенным проектом.

3.2.1 Настройки таргет-визуализации

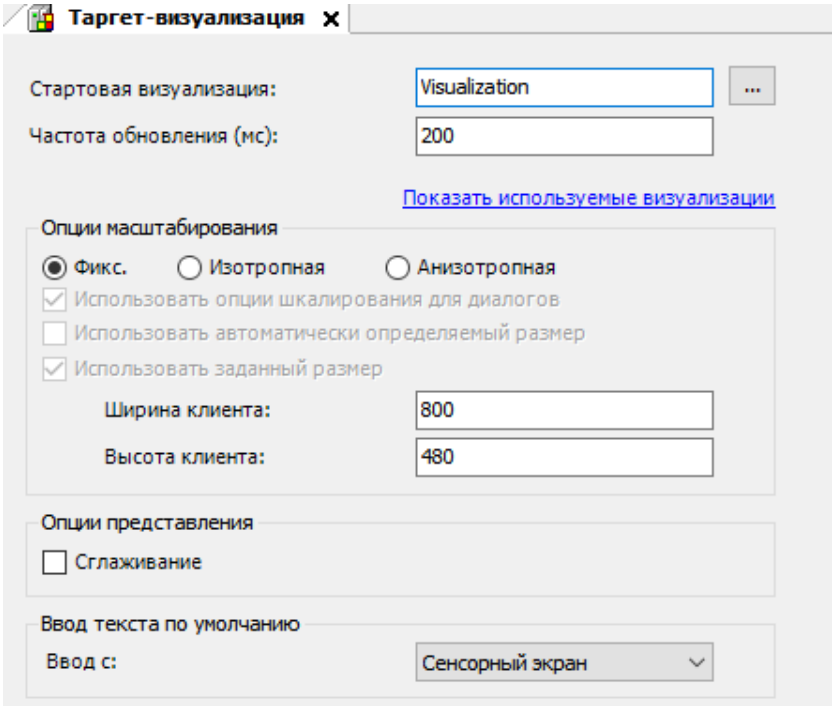


Рисунок 3.11 – Настройки таргет-визуализации

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.

Опции масштабирования – позволяет выбрать режим масштабирования экранов визуализации на целевом устройстве:

- **Фикс.** – размер экрана визуализации на целевом устройстве будет соответствовать оригинальному размеру экрану визуализации (указанному в свойствах экрана визуализации);
- **Изотропная** – позволяет задавать размер визуализации в пикселях. Если размеры экрана не соответствуют дисплею целевого устройства, то экран визуализации будет ***масштабироваться с сохранением*** соотношения сторон;
- **Анизотропная** – позволяет задавать размер визуализации в пикселях. Если размеры экрана не соответствуют дисплею целевого устройства, то экран визуализации будет ***масштабироваться без сохранения*** соотношения сторон.

В случае наличия галочки **Use scaling options for dialog** диалоги таргет-визуализации будут масштабироваться с заданными опциями.

- **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.

3.2.2 Настройки web-визуализации

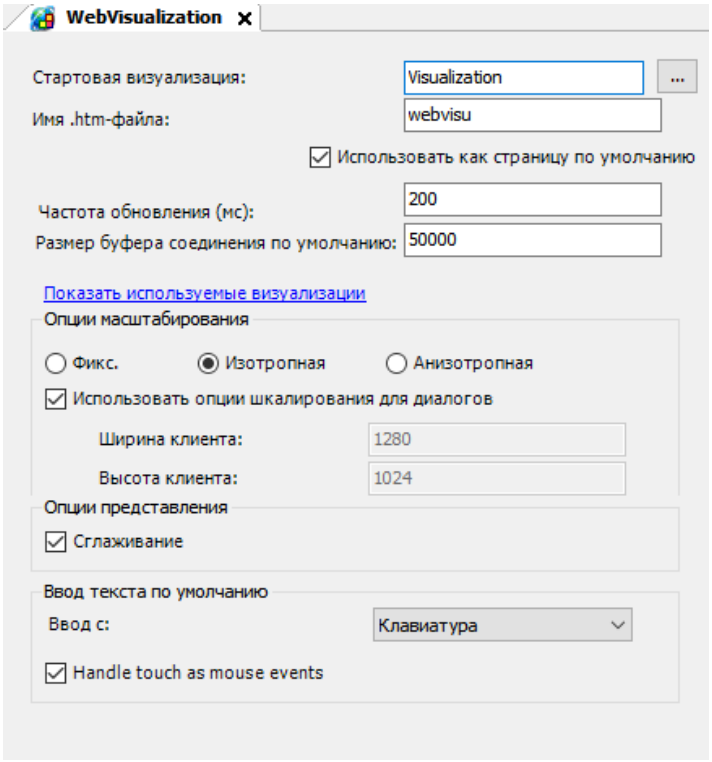


Рисунок 3.12 – Настройки web-визуализации

Стартовая визуализация – позволяет выбрать экран визуализации, который будет отображен на клиенте web-визуализации в случае открытия данной веб-страницы.

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

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

В случае наличия галочки **Use as default page** галочки в адресе может отсутствовать имя web-страницы. Галочка может быть установлена только в одном из компонентов web-визуализации.

ПРИМЕЧАНИЕ
Информация о подключении к web-визуализации без указания порта и информацию по подключению через **HTTPS** приведена в руководстве **CODESYS V3.5. FAQ**.

Частота обновления – позволяет 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:

Фикс. – позволяет задавать фиксированный размер визуализации в пикселях;

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

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

В случае установки галочки **Использовать опции шкалирования для диалогов** диалоги web-визуализации будут масштабироваться с заданными опциями.

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.

Handle touch as mouse events – в случае установки галочки на устройствах с сенсорным экраном жесты обрабатываются как нажатия мыши. Эта опция требуется, например, для управления ползунком или полосой прокрутки на сенсорном устройстве.

3.2.3 Настройки удаленной таргет-визуализации

Стартовая визуализация – позволяет выбрать экран визуализации, который будет отображен клиентом при запуске удаленной таргет-визуализации.

Остальные настройки хранятся на ПК в файле **targetvisuremote.cfg**, расположенном в папке установки CODESYS по пути **…\GatewayPLC**.

ПРИМЕЧАНИЕ

Удаленная таргет-визуализация является дополнительным платным компонентом **CODESYS**. Контроллеры ОВЕН не содержат лицензию на этот компонент.

4 Стили и профили визуализации

4.1 Стили визуализации

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

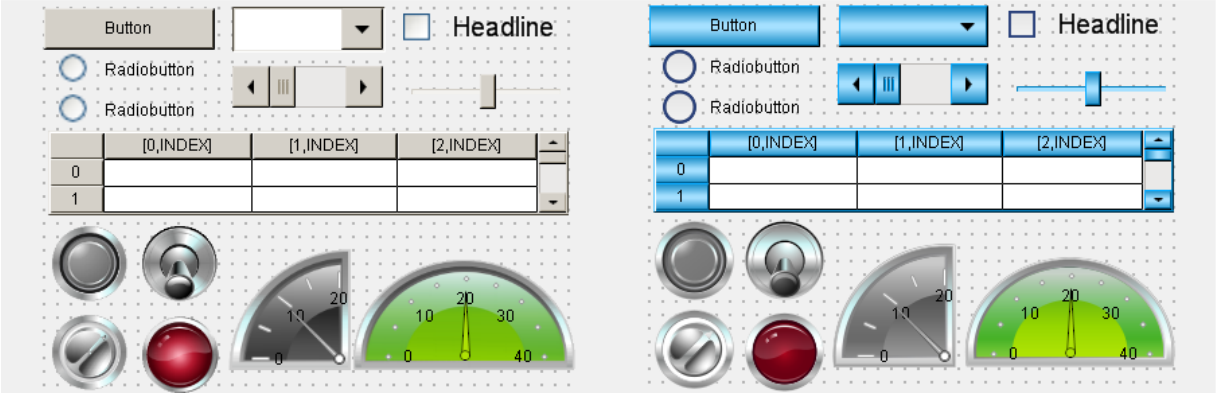


Рисунок 4.1 – Разница между двумя стилями визуализации

Стиль визуализации можно выбрать в установках **Менеджера визуализации**:

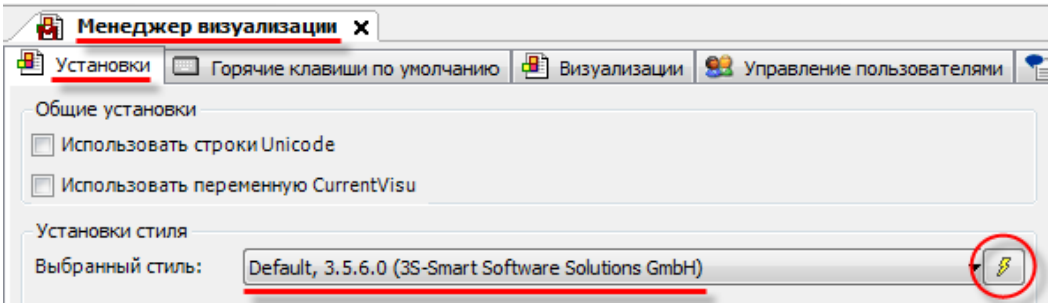


Рисунок 4.2 – Выбор стиля визуализации

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

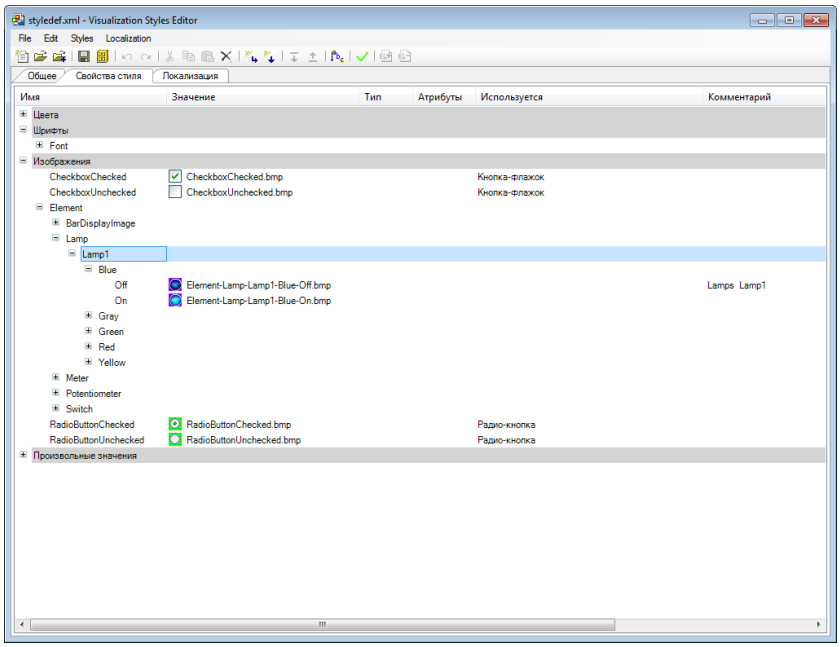


Рисунок 4.3 – Окно редактора стилей визуализации

Файлы стиля визуализации по умолчанию хранятся в формате **.xml** по адресу

C:\ProgramData\CODESYS\Visualization Styles

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

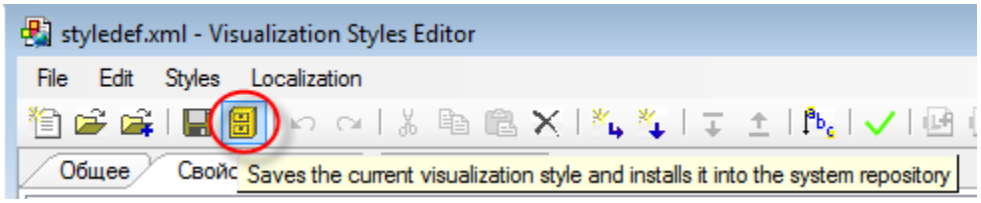


Рисунок 4.4 – Сохранение стиля визуализации и добавления его в редактор визуализации

После выбора стиля следует перезапустить **CODESYS**. Сохраненный стиль можно будет выбрать в установках **Менеджера визуализации** (см. п. 4.1).

См. более подробную информацию о стилях визуализации в этой статье.

4.2 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):

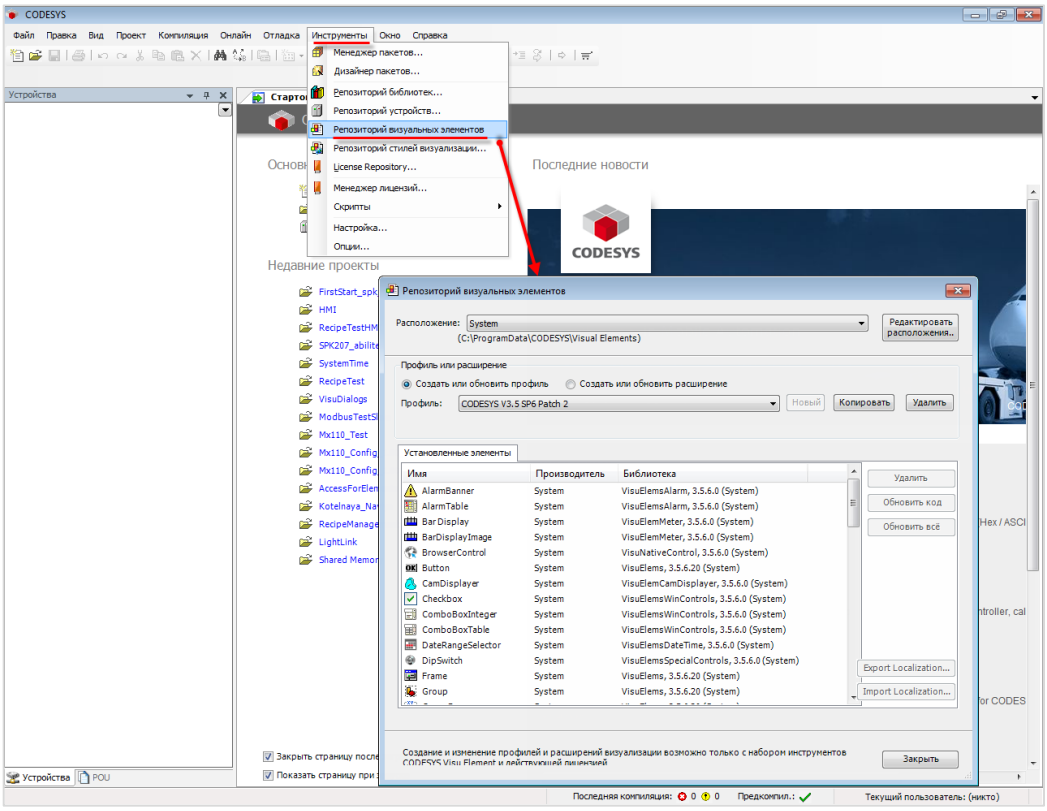


Рисунок 4.5 – Репозиторий визуальных элементов

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

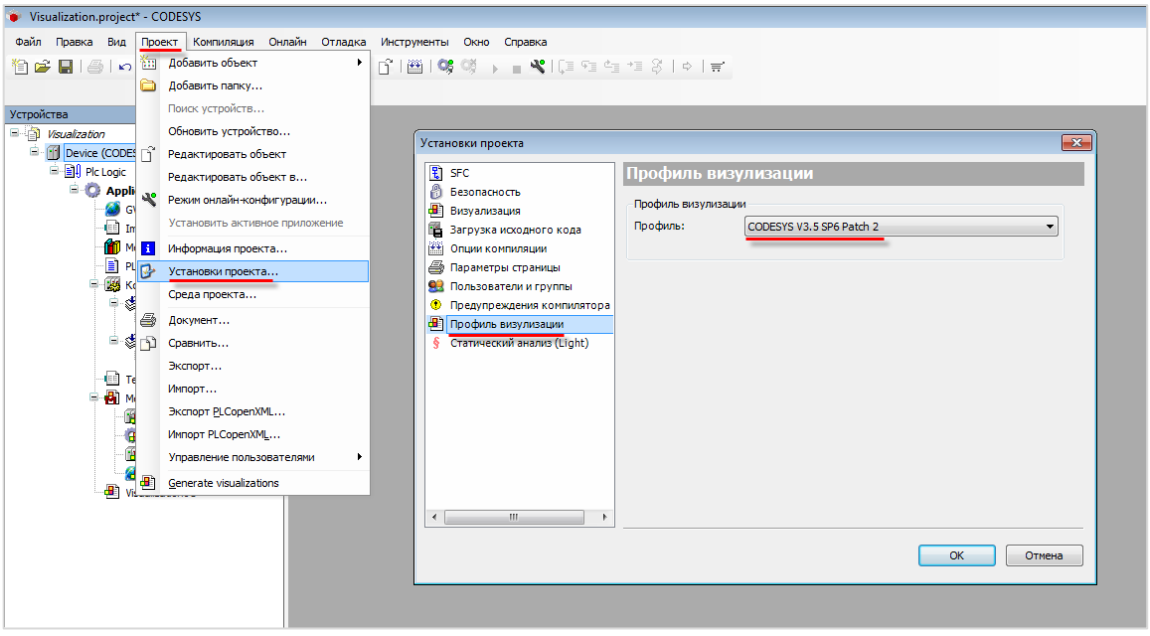


Рисунок 4.6 – Выбор профиля визуализации, используемого в проекте

5 Редактор визуализации

5.1 Интерфейс редактора визуализации

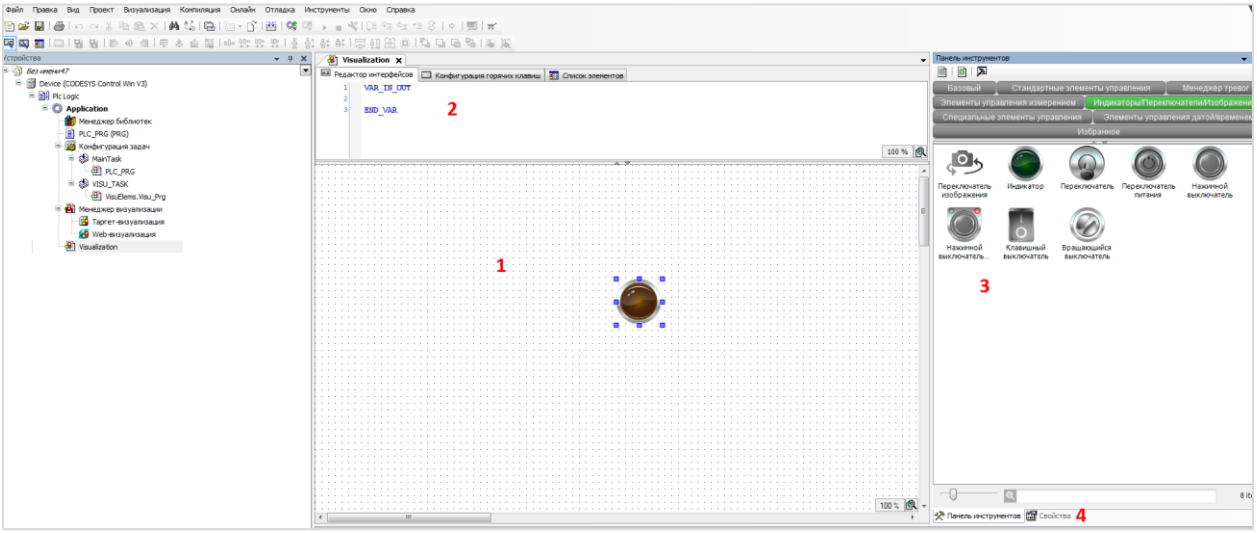


Рисунок 5.1 – Интерфейс редактора визуализации

Редактор визуализации открывается в случае выбора любого экрана визуализации проекта. Редактор включает в себя:

1. **Рабочую область**, в которой создается экран визуализации.
2. **Область объявления переменных**.
3. **Панель инструментов редактора**, содержащую набор графических примитивов.
4. **Панель свойств**, содержащую параметры графического элемента (открывается на месте **Панели инструментов редактора** в случае выбора соответствующего элемента).
5. 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 данные команды присутствуют в контекстном меню **Редактора визуализации**, открываемом с помощью нажатия **ПКМ** на любое место рабочей области. Описание команд приведено в п. 6.3.

5.2 Добавление элемента на экран визуализации. Взаимодействие с элементами

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

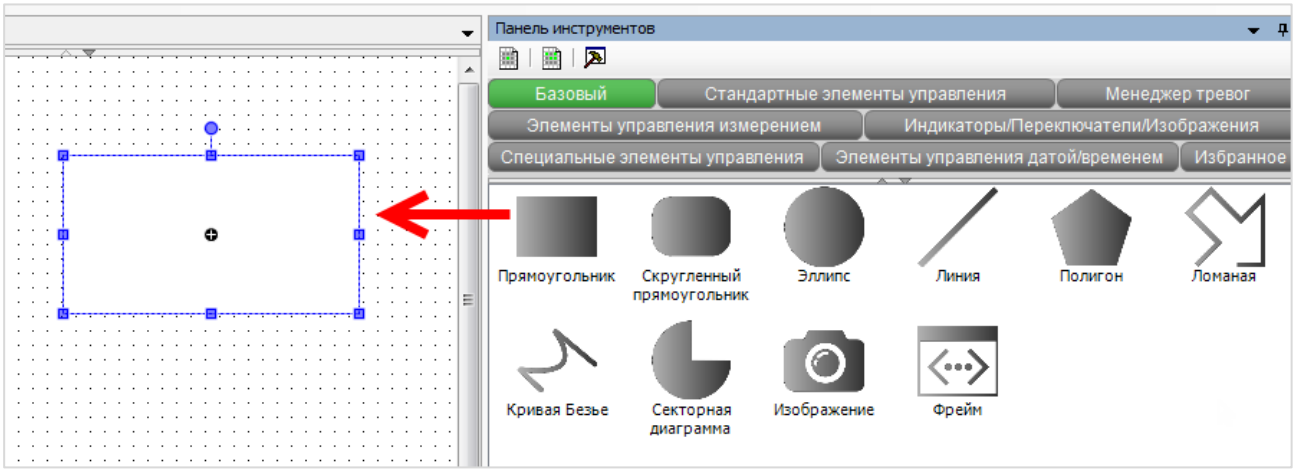


Рисунок 5.2 – Область рабочего поля с добавленным элементом Прямоугольник

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

1. Для **перетаскивания** следует выделить элемент, навести на него мышью для появления курсора перетаскивания ( ) и, зажав **ЛКМ**, перетащить элемент в нужное место экрана.
2. Для **вращения** следует выделить элемент, навести мышью на точку вращения ( ) для появления курсора вращения ( ) и, зажав ЛКМ, повернуть элемент на нужный угол.
3. Для **изменения размера** следует выделить элемент, навести мышью на одну из его опорных точек ( ) для появления курсора деформации ( ) и, зажав **ЛКМ**, растянуть элемент до нужных размеров.
4. Для **изменения текста элемента** следует на текстовое поле элемента мышью для появления курсора ввода ( ), после чего однократным нажатием **ЛКМ** перейти к вводу/редактированию текста. Данная операция возможна только для элементов, которые могут отображать текст. Введенный текст попадет в глобальный список текстов (см. рисунок 5.5).
5. Для **перемещения центра элемента** следует выделить элемент, навести мышью на его центр ( ) до появления курсора перетаскивания ( )и, зажав **ЛКМ**, перетащить 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 следует зажав **ЛКМ**, обвести их рамкой, после чего отпустить кнопку мыши, либо зажать клавишу **Shift** и последовательно нажимать на элементы **ЛКМ**. Во втором случае опорные точки элемента, выделенного первым, сохранят синий цвет:

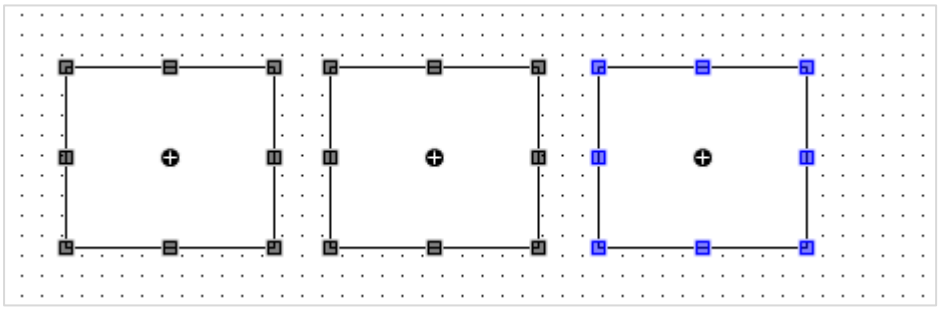


Рисунок 5.3 – Одновременное выделение нескольких элементов

5.3 Команды редактора визуализации

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

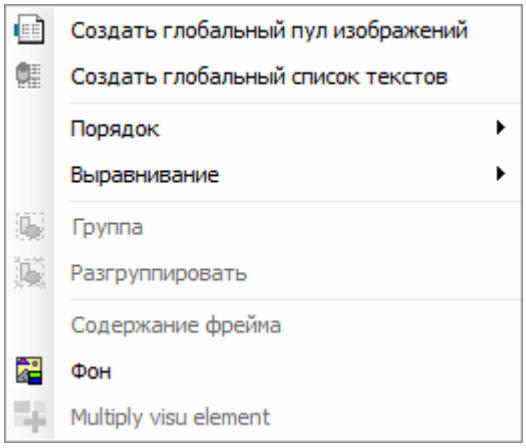


Рисунок 5.4 – Команды контекстного меню редактора визуализации

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 данных компонентов распространяется на все приложения проекта.

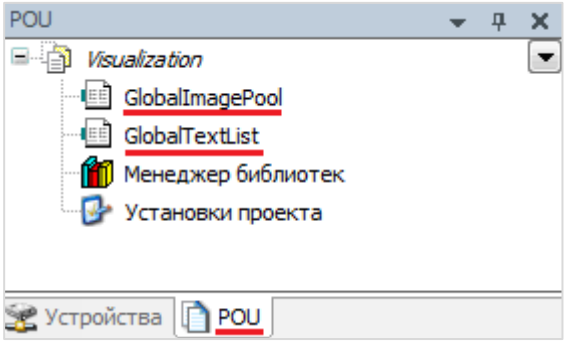


Рисунок 5.5 – Глобальные пул изображений и список текстов

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

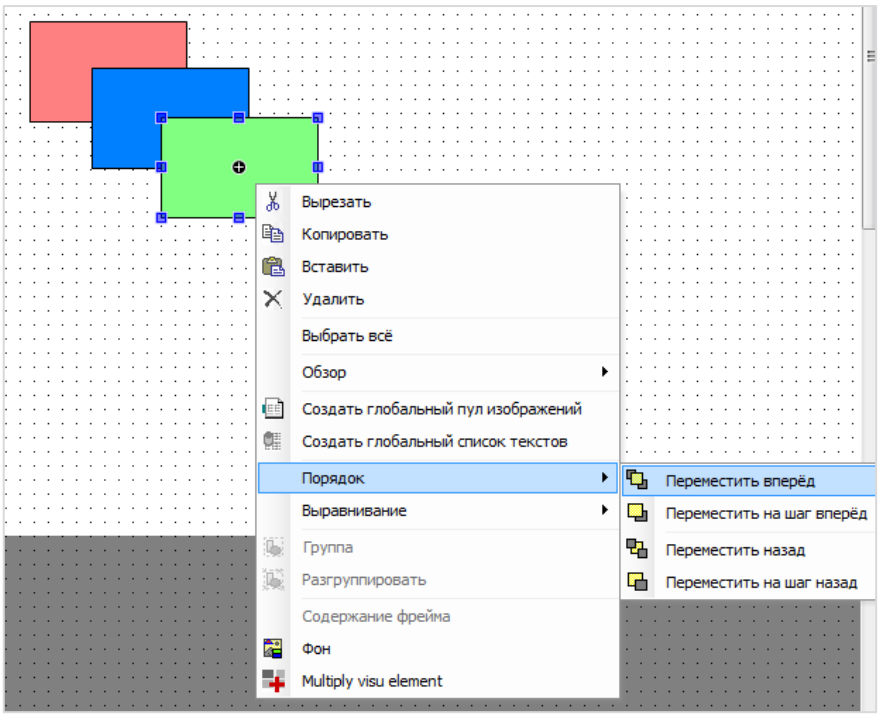


Рисунок 5.6 – Вкладка настройки порядка отображения элементов

3. Команды вкладки **Выравнивание** (становится доступной только в случае одновременного выделения нескольких элементов) позволяет выравнивать элементы и настраивать их отступы и междустрочные интервалы:

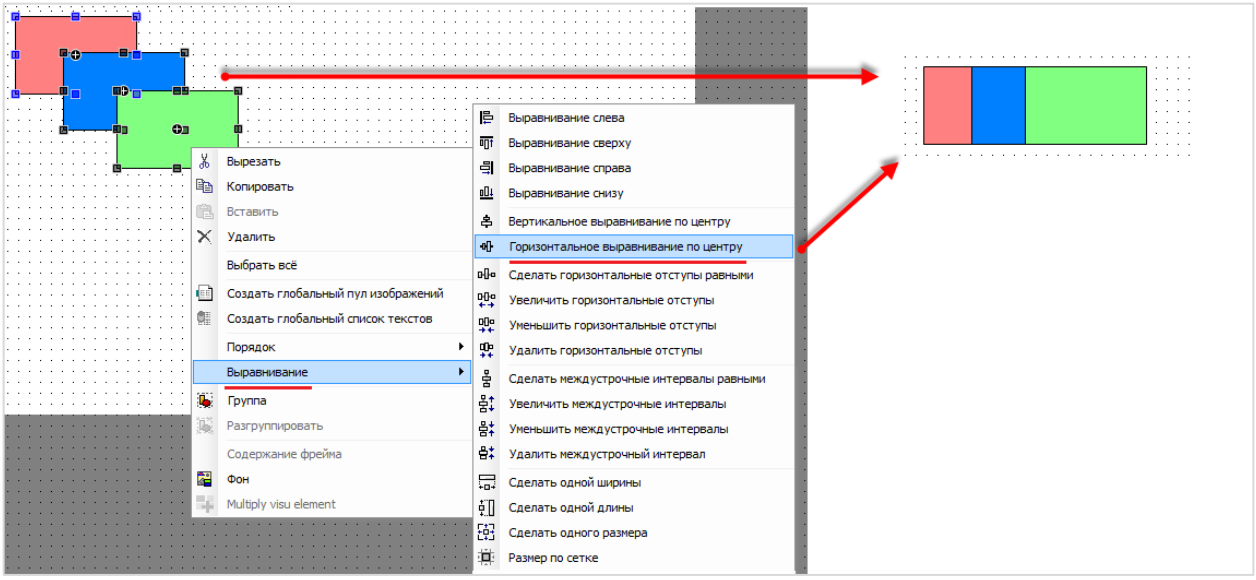


Рисунок 5.7 – Вкладка выравнивания элементов

4. Команда **Группа** позволяет сгруппировать элементы – т. е. все взаимодействие с ними (выделение, перемещение и т. д.) будет происходить как с единым целым. 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** разгруппировывает элементы.

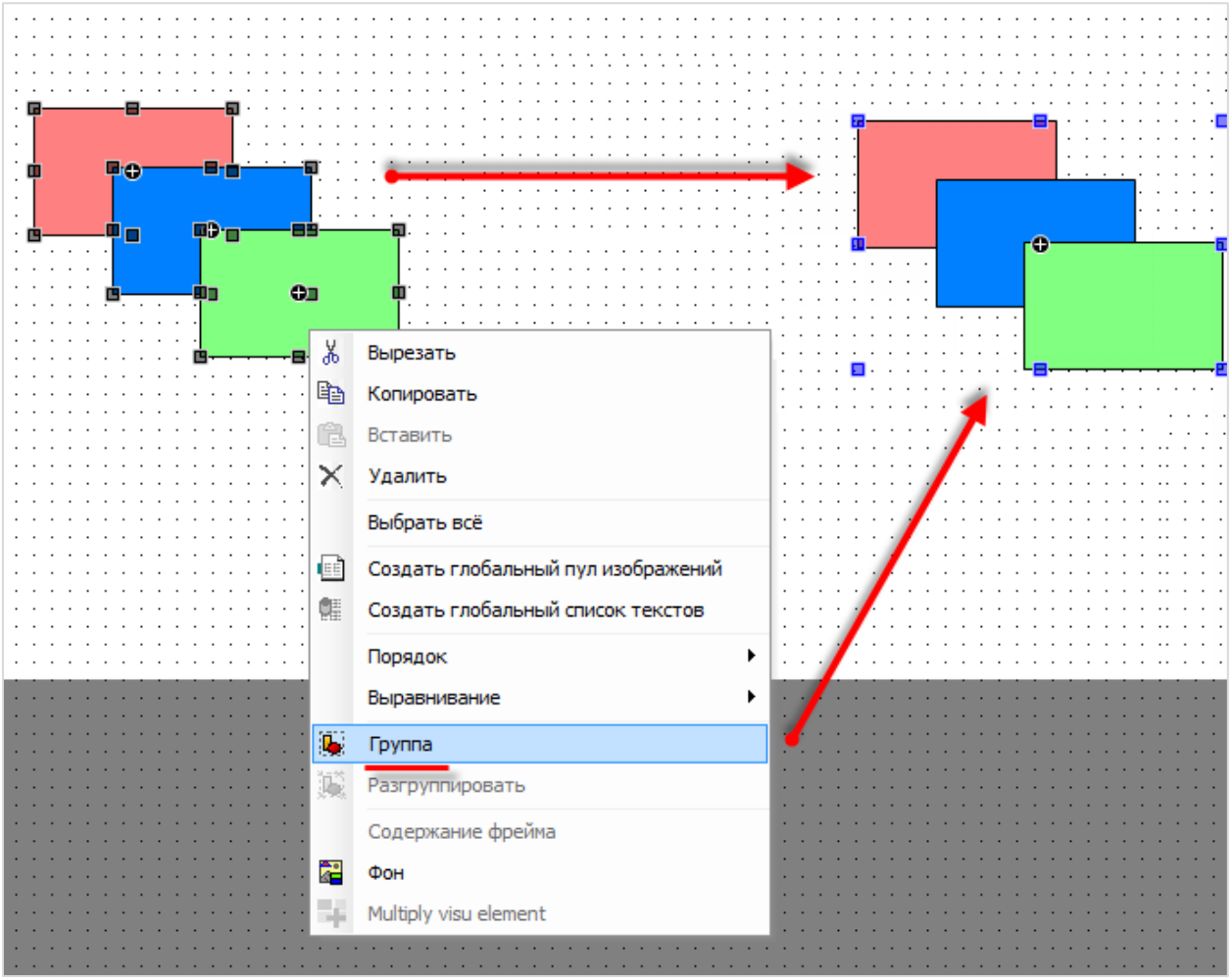


Рисунок 5.8 – Группировка элементов

5. Команда **Содержимое фрейма** доступна для элементов Фрейм и Набор вкладок. Команда позволяет выбрать экраны визуализации, которые будут отображаться в данных элементах.

6. Команда **Фон** позволяет выбрать цвет фона экрана визуализации или фоновое изображение. Предварительно фоновое изображение должно быть загружено в Пул изображений.

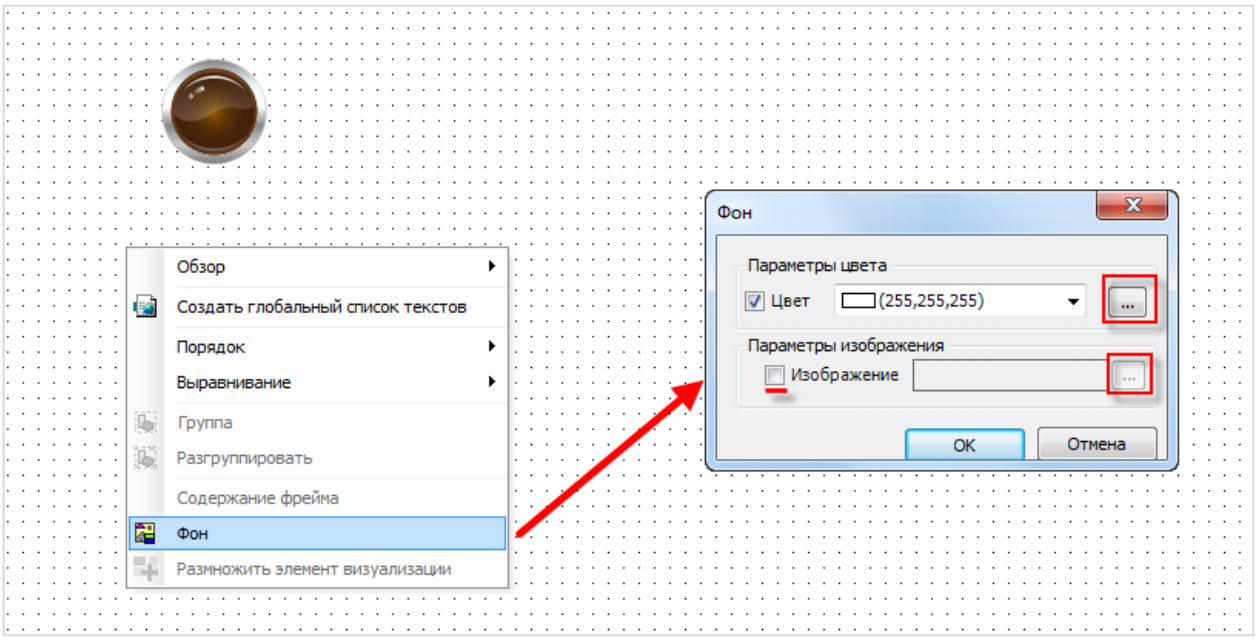


Рисунок 5.9 – Выбор фонового изображения экрана визуализации

7. Команда **Размножить элемент визуализации** позволяет создавать группы однотипных элементов (например, ряды переключателей). Пример ее использования приведен в п. 10.3.6.
8. Команда **Move To Content Area** доступна только в случае установки фонового изображения на вкладке Advanced Settings менеджера визуализации. Эта команда автоматически переносит все элементы экрана, расположенные за пределами рабочей области, в рабочую область.

5.4 Вспомогательные редакторы экранов визуализации

5.4.1 Редактор интерфейсов

Редактор интерфейсов предназначен для объявления переменных, которые используются на данном экране визуализации. Данные переменные обычно применяются в диалогах и экранах, открываемых во фреймах. Подробную информацию о редакторе интерфейсов можно найти в справке CODESYS.

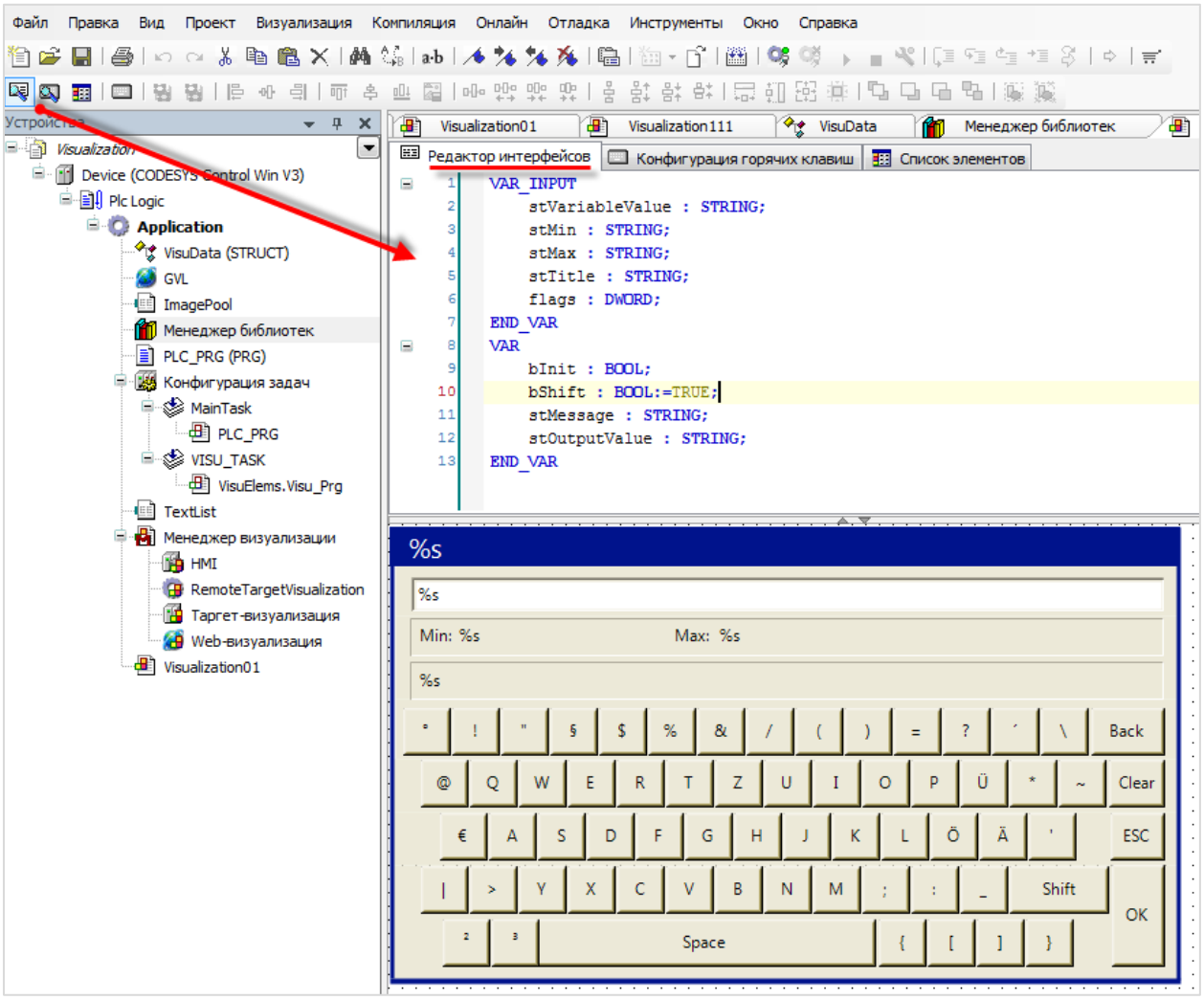


Рисунок 5.10 – Редактор интерфейсов

5.4.2 Редактор горячих клавиш

В **Редакторе горячих клавиш** можно привязать к нажатию клавиш выполнение действий. Для этого целевое устройств должно иметь соответствующие кнопки. Если галочка **Клавиша нажата** не установлена, то действие будет выполняться при отпускании зажатой клавиши. Также 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.

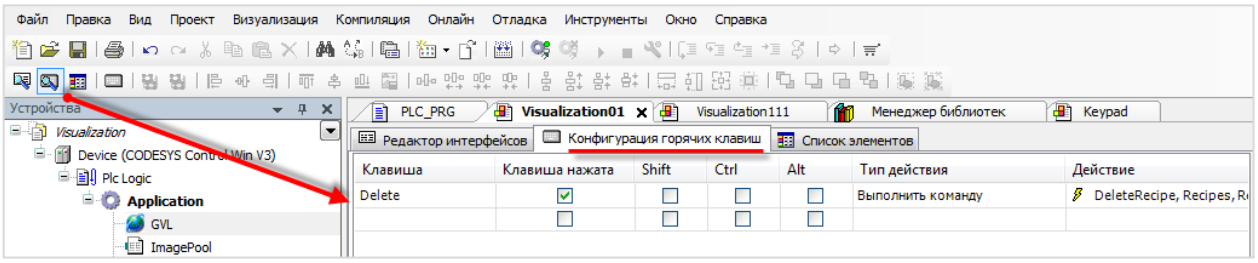


Рисунок 5.11 – Редактор горячих клавиш экрана визуализации

5.4.3 Редактор списка визуальных элементов

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

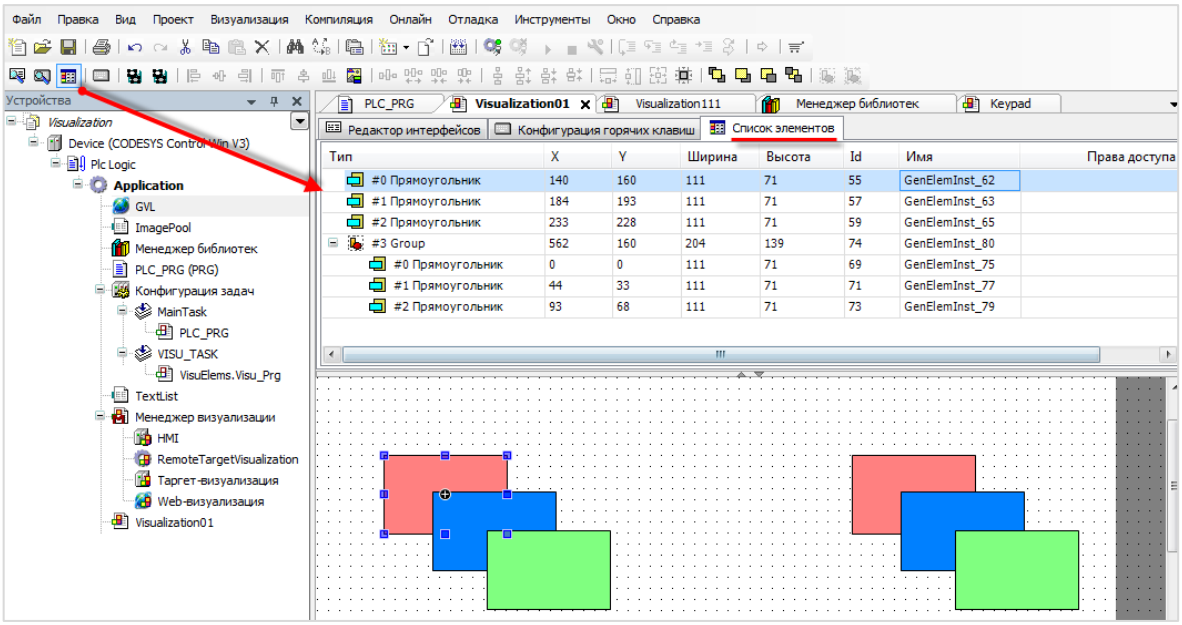


Рисунок 5.12 – Редактор списка визуальных элементов

Столбец **Порядок вкладок** определяет порядок переключения элементов клавишей **Tab** при активации галочки **Вкл. стандартное использование клавиатуры** в Менеджере визуализации.Если оставить для элемента это поле пустым, то элемент нельзя будет выбрать с помощью клавиши **Tab**.

5.5 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**:

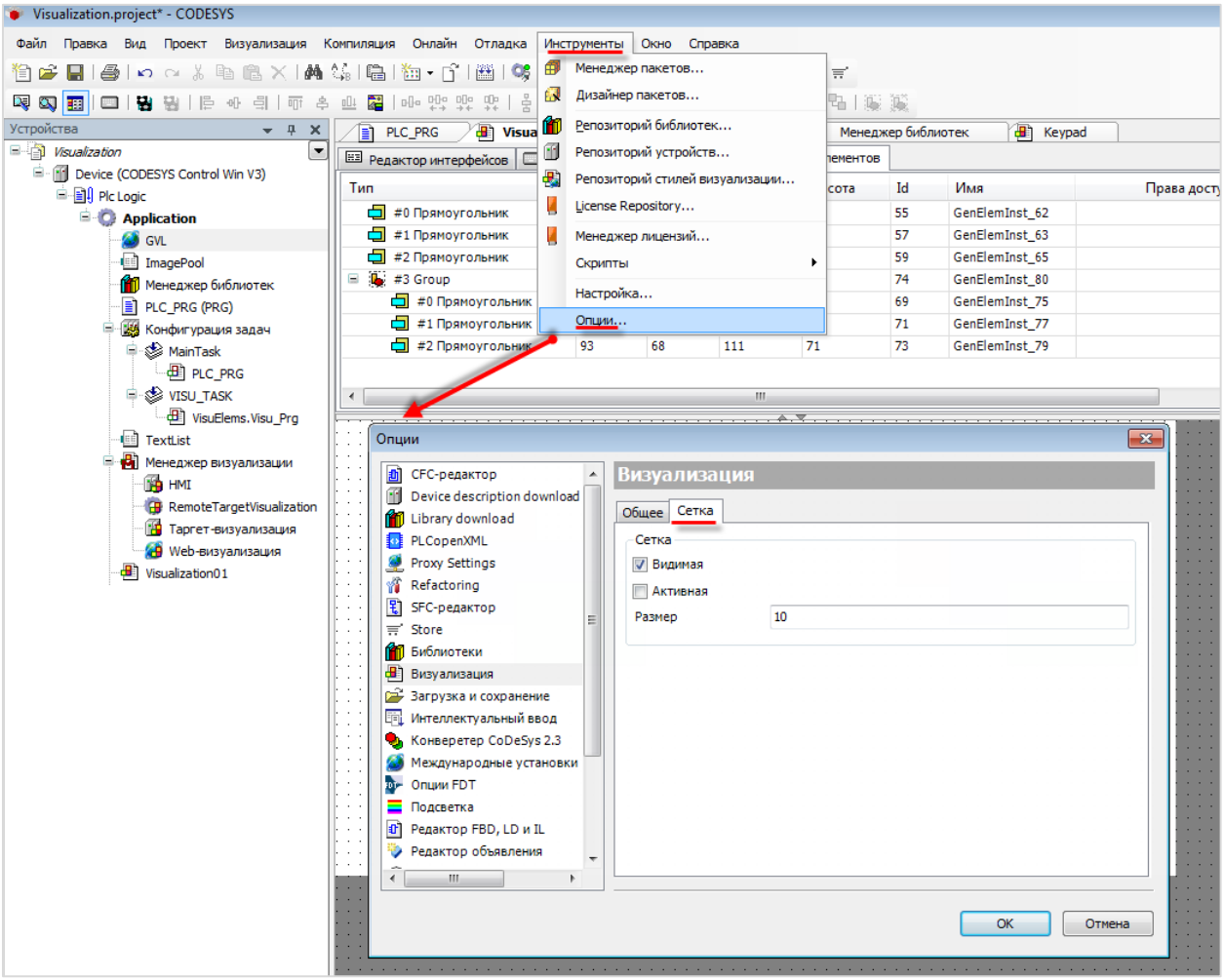


Рисунок 5.13 – Настройки сетки редактора визуализации

6 Вспомогательные компоненты визуализации

6.1 Пул изображений

Пул изображений представляет собой хранилище графических файлов, используемых в проекте. Поддерживается большинство популярных форматов графических файлов, например, .jpg, .png, .bmp, .svg[1] и т. д.).

ПРИМЕЧАНИЕ
Названия файлов не должны содержать кириллические символы, спецсимволы и точки. Поддерживаются только цифры и символы латиницы.

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

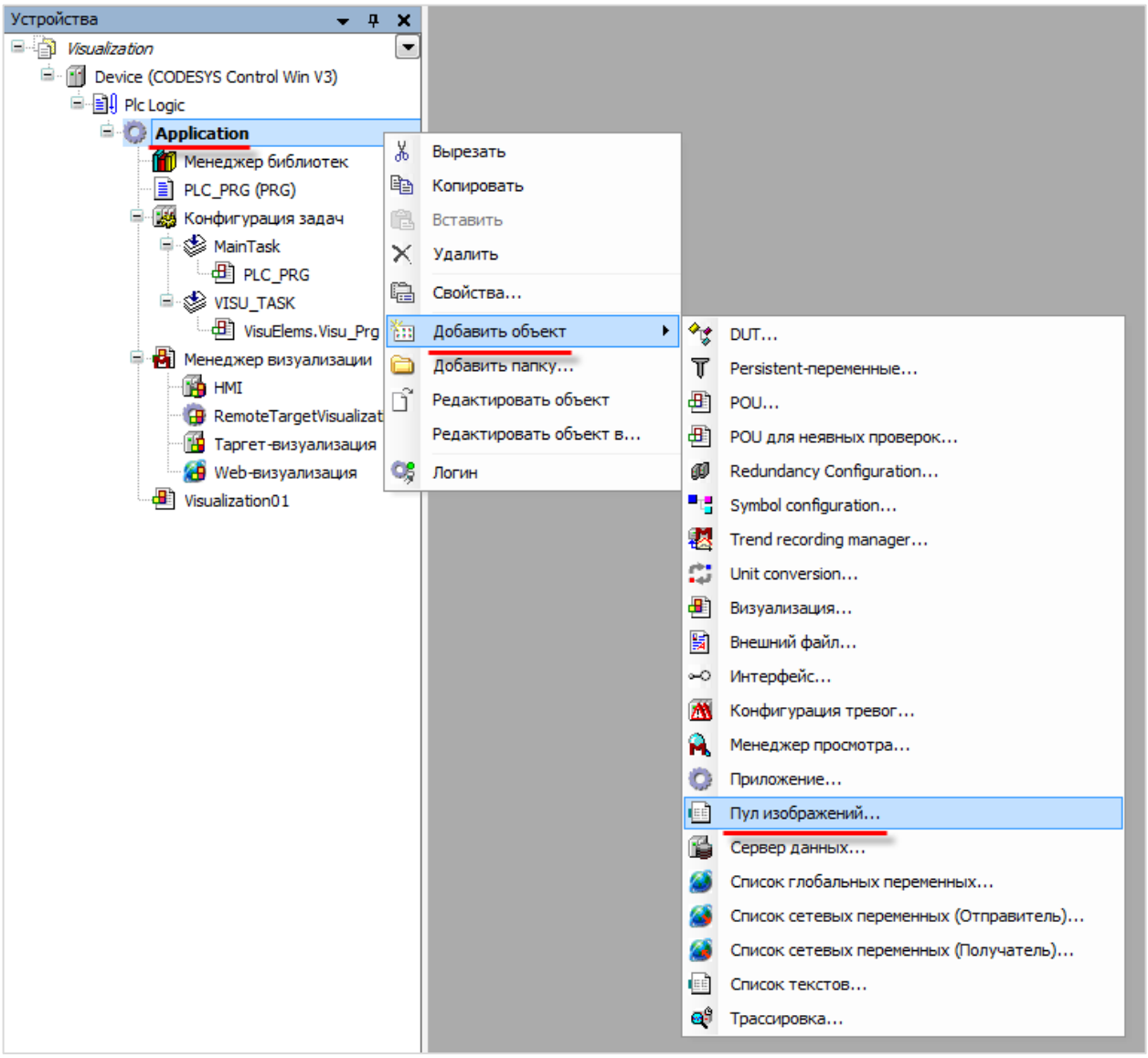


Рисунок 6.1 – Добавление Пула изображений в проект

[1] Требования: svg version="1.2", baseProfile="tiny", правило **Только использованные глифы**

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

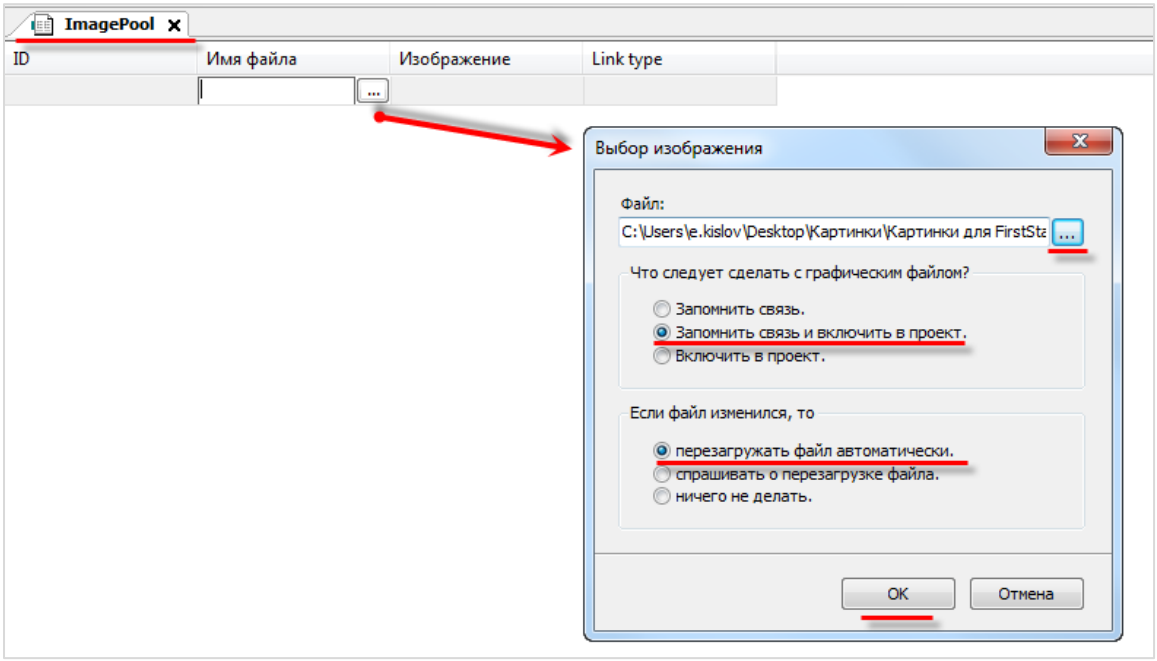


Рисунок 6.2 – Выбор изображения для загрузки

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 в случае изменения изображения – оно будет автоматически меняться в проекте.

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

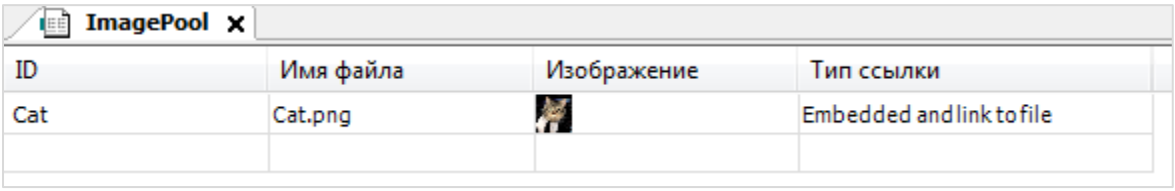


Рисунок 6.3 – Пул изображений после добавления файла

Области использования графических файлов:

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

6.2 Список текстов

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

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

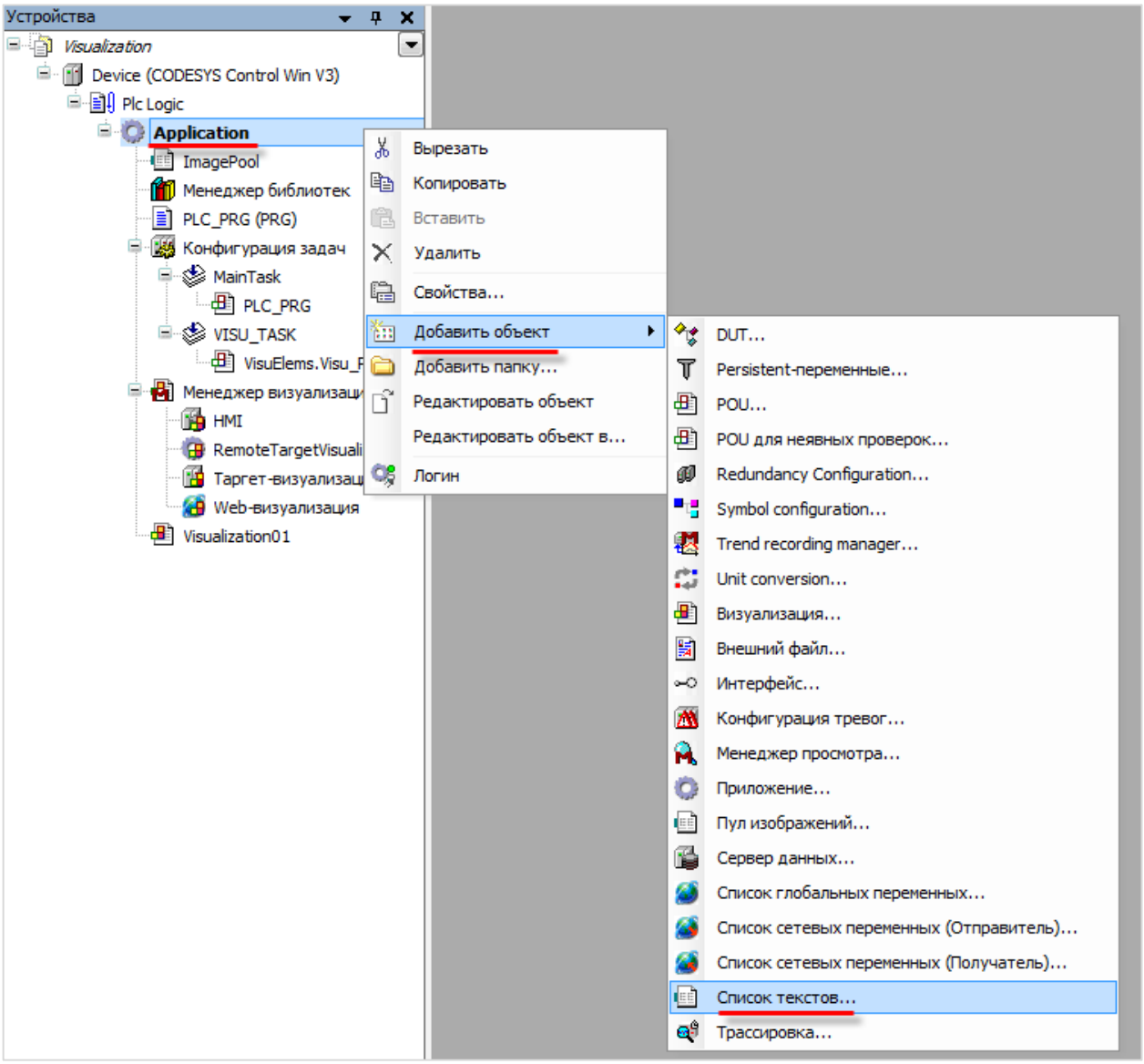


Рисунок 6.4 – Добавление Списка текстов в проект

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

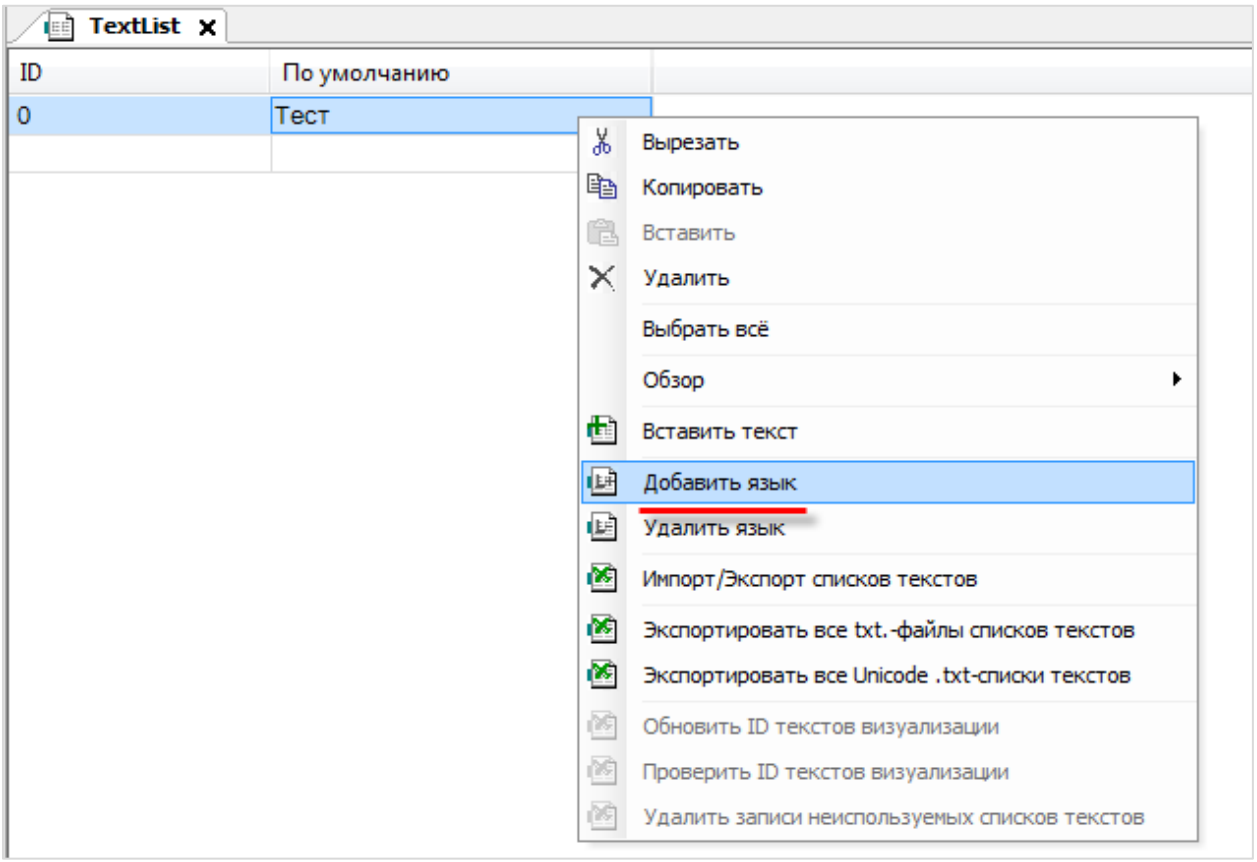


Рисунок 6.5 – Добавление нового языка в список текстов

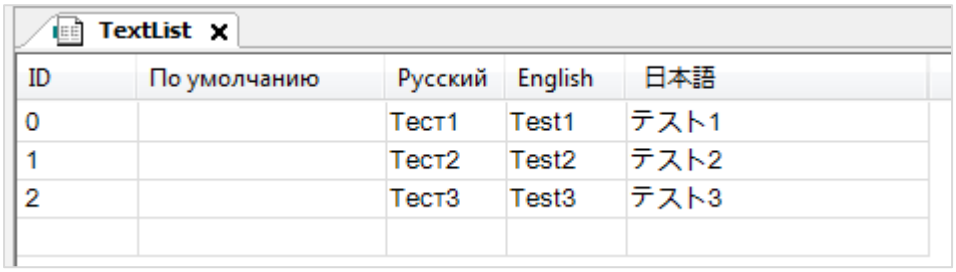


Рисунок 6.6 – Мультиязычный список текстов

Пример реализации **мультиязычного проекта** приведен в п. 10.3.5.

ПРИМЕЧАНИЕ
Для работы со списками текстов в коде программы можно использовать библиотеку **CmpDynamicText**. См. более подробную информацию в этой статье.

7 Описание графических примитивов

7.1 Структура главы

В главе приводится описание параметров всех графических примитивов, присутствующих в **CODESYS** версии **V3.5 SP17 Patch 3**. Так как 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 в случае выделения элемента нажатием **ЛКМ**.

ПРИМЕЧАНИЕ
Для отображения всех параметров элемента, следует поставить галочку **Эксперт**:

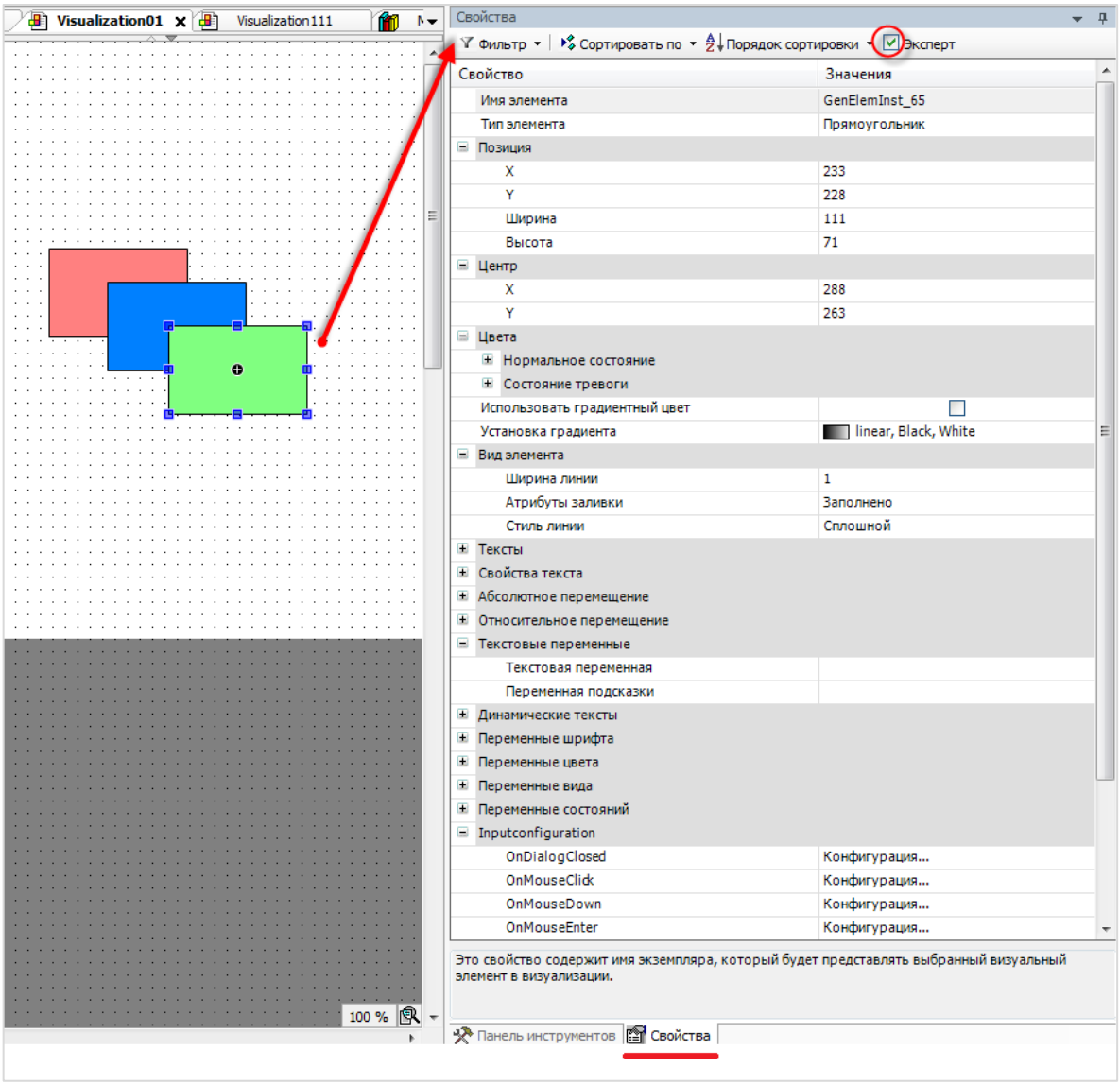


Рисунок 7.1 – Настройки сетки редактора визуализации

7.2 Базовая группа элементов

7.2.1 Прямоугольник, Скругленный прямоугольник, Эллипс. Типичные параметры графических примитивов

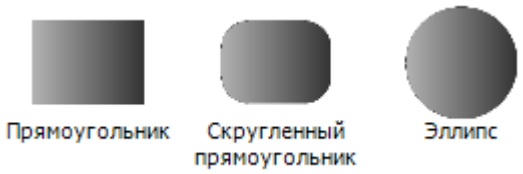


Рисунок 7.2 – Пиктограммы графических примитивов Прямоугольник, Скругленный прямоугольник, Эллипс

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

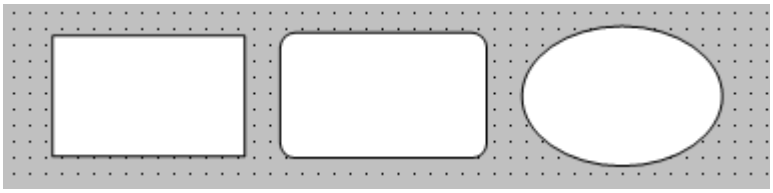


Рисунок 7.3 – Внешний вид элементов Прямоугольник, Скругленный прямоугольник, Эллипс

Элементы могут преобразоваться друг в друга с помощью изменения **Типа элемента**:

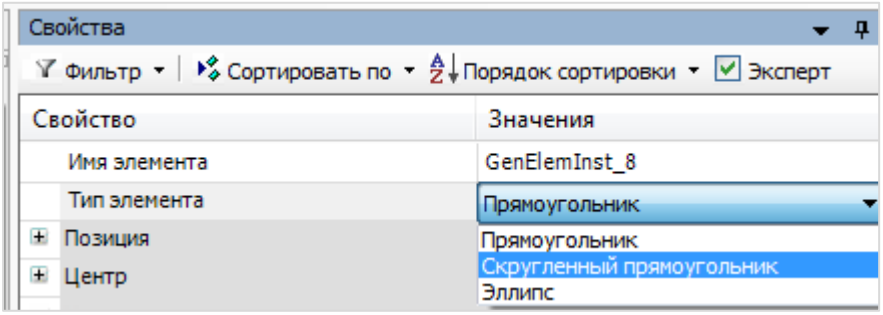


Рисунок 7.4 – Изменение типа элемента

Описание параметров данных графических примитивов приведено в таблице 7.1. Так как значительная часть данных 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. 10.2.1.

Таблица 7.1 – 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**
1.5.	Угол	Угол поворота элемента в градусах.
2.	**Центр**	
2.1.	X	Координата опорной точки центра (⊕) 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 (⊕) 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	Если установлена галочка в данном параметре, для заливки элемента используется градиентный цвет
3.4.	Установка градиента	Настройки градиентного цвета

Продолжение Табл. 7.1

№	Название параметра	Описание параметра
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	Формат текста	Позволяет выбрать тип переноса текста элемента, если он не умещается в одну строку. Возможные значения: **По умолчанию** – текст не переносится, **Разрыв строки** – текст переносится, **Многоточие** – текст переносится, последний отображенный фрагмент заканчивается троеточием
6.4.	Шрифт	Позволяет выбрать шрифт текста элемента, а также определить его размер и стиль (**жирный**, *курсив* и т. д.)
6.5.	Цвет шрифта	Цвет шрифта текста элемента
6.5.1.	Цвет шрифта/ Прозрачность	Прозрачность текста элемента. Возможные значения: **0…255** **0** – полностью прозрачный текст, **255** – полностью непрозрачный

Продолжение Табл. 7.1

№	Название параметра	Описание параметра
7.	**Абсолютное перемещение**	
7.1.	***Перемещение***	
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)
7.5.	Использовать действ. значения	В случае наличия галочки для параметров пп. 7 можно применять переменные типа **REAL** (для точного позиционирования)
8.	**Относительное перемещение**	
8.1.	***Перемещение сверху налево***	
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.	***Перемещение снизу вверх***	
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 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

Продолжение Табл. 7.1

№	Название параметра	Описание параметра
11.	**Переменные шрифта**	
11.1.	Имя шрифта	Строковая переменная, содержащая имя шрифта текста элемента. ***Обратите внимание***, что шрифт должен поддерживаться целевым устройством (см. п. 11.7)
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 – курсив 4 – подчеркнутый 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 (CDS-57944)
11.5.	Цвет	Переменная типа **DWORD**, определяющая цвет текста по модели ARGB
11.6.	Флаги для выравнивания текста	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 ARGB
12.2.2.	Цвет заливки	Переменная типа **DWORD**, определяющая цвет и прозрачность заливки элемента в нормальном состоянии по модели ARGB
12.3.	***Состояние тревоги***	
12.3.1.	Цвет фрейма	Переменная типа **DWORD**, определяющая цвет и прозрачность контура элемента в состоянии тревоги по модели ARGB
12.3.2.	Цвет заливки	Переменная типа **DWORD**, определяющая цвет и прозрачность заливки элемента в состоянии тревоги по модели ARGB

Продолжение Табл. 7.1

№	Название параметра	Описание параметра
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, **1** – 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) **5** – 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
A.1.	Длительность анимации	Переменная типа **DINT**, определяющая время анимации элемента в миллисекундах. См. демонстрацию эффекта анимации в этом видео
A.2.	Переместить на передний план	Переменная типа **BOOL**, позволяющая переместить данный элемент на передний план (если несколько элементов перекрывают друг друга). См. демонстрацию в этом видео
15.	**InputConfiguration** Данная вкладка содержит настройки действий, выполняемых при нажатии элемент. Ее подробное описание приведено в п. 9	
16.	**Права доступа** Данная вкладка доступна только в случае наличия 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	
17.	**Настройки радиуса скругления (только для элемента Скругленный прямоугольник)**	
17.1.	Радиус	Способ определения радиуса скругления. Возможные значения: **Из стиля/В соответствие с размером элемента/Заданный**
17.2.	Значение	Значение радиуса скругления (для режима **Заданный)**

7.2.2 Линия

Рисунок 7.5 – Пиктограмма графического примитива Линия

Графический примитив **Линия** представляет собой отрезок.

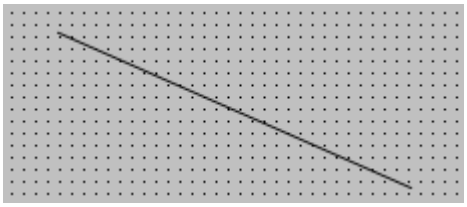


Рисунок 7.6 – Внешний вид элемента Линия

Описание типичных параметров графического примитива приведено в таблице 7.1.

Уникальными параметрами графического примитива являются только координаты точек начала и конца отрезка в пикселях (к параметрам могут быть привязаны целочисленные переменные):

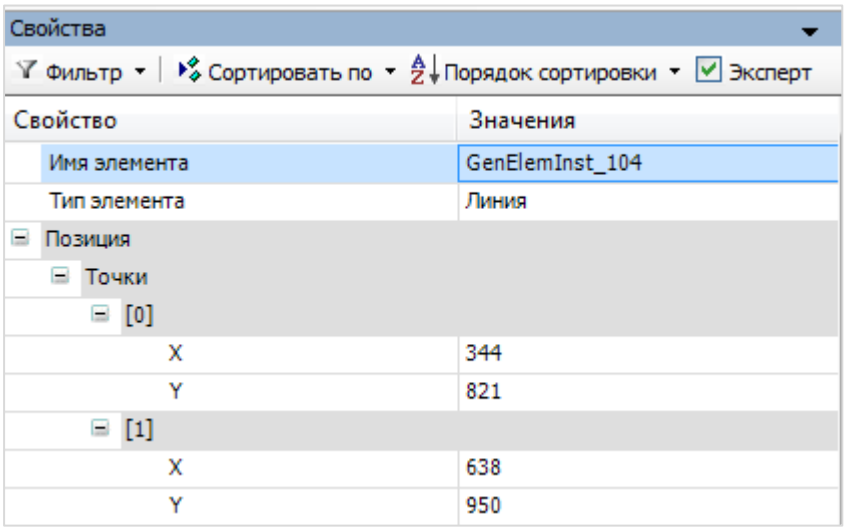


Рисунок 7.7 – Уникальные параметры графического примитива Линия

7.2.3 Полигон, Ломаная, Кривая Безье

Рисунок 7.8 – 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**:

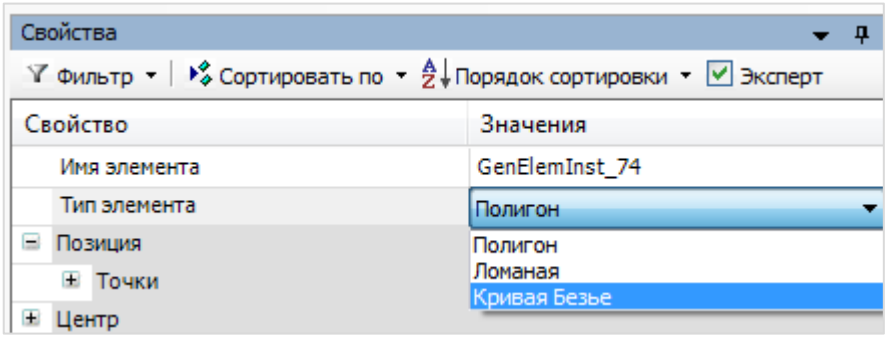


Рисунок 7.9 – Изменение типа элемента

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 следует 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 создастся поверх существующей). Удалить точку можно нажатием на нее **ЛКМ** при зажатых клавишах **Ctrl** и **Shift**.

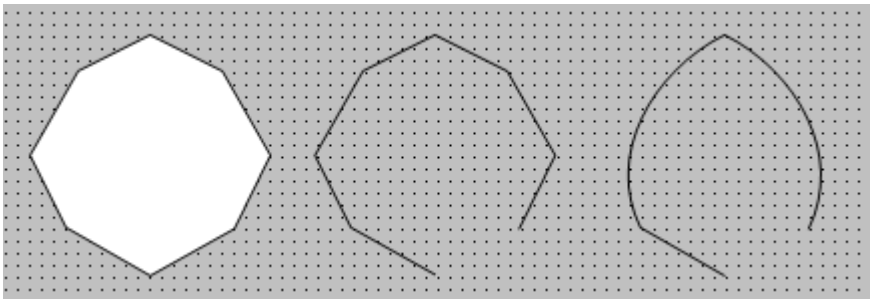
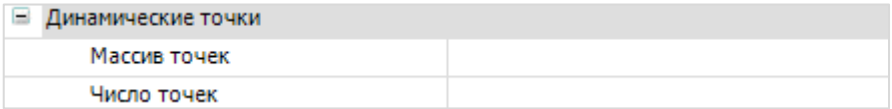


Рисунок 7.10 – Внешний вид графических примитивов Полигон, Ломаная, Кривая Безье

Описание типичных параметров графического примитива приведено в таблице 7.1.

Уникальными параметрами графического примитива являются параметры вкладки **Динамические точки**:



Массив точек – это адрес (ADR) массива типа **VisuElems.CmpVisuHandler.VisuStructPoint,** содержащий текущие координаты точек элемента.

Число точек – целочисленная переменная, содержащая число точек элемента.

Изменение данных параметров в процессе работы программы позволяет менять внешний вид элемента. Пример их использования приведен в п. 10.4.2.

7.2.4 Секторная диаграмма

Рисунок 7.11 – Пиктограмма графического примитива Секторная диаграмма

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

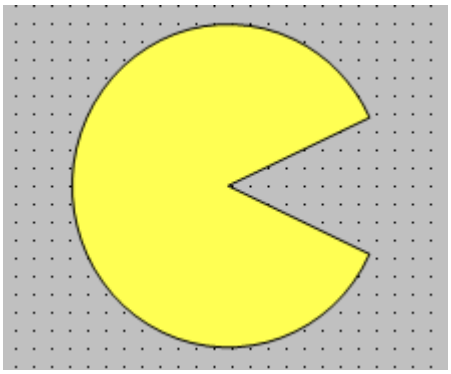
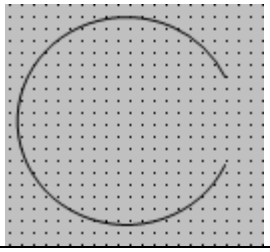


Рисунок 7.12 – Внешний вид графического примитива Секторная диаграмма

Описание типичных параметров графического примитива приведено в таблице 7.1. Уникальные параметры примитива описаны в таблице 7.2:

Таблица 7.2 – 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: 

7.2.5 Изображение

Рисунок 7.13 – 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 одно из системных изображений (используемых для отображения графических примитивов), либо пользовательское изображение, добавленное в Пул изображений.

Рисунок 7.14 – Внешний вид элемента Изображение

Описание типичных параметров графического примитива приведено в таблице 7.1.
Уникальные параметры примитива описаны в таблице 7.3:

Таблица 7.3 – Уникальные параметры графического примитива Изображение

№	Название параметра	Описание параметра
1.	Показать фрейм	Если установлена галочка, то вокруг изображения отображается контур
2.	Кадрирование	Если установлена галочка и выбран тип шкалы **Фиксированный** (см. пп. 5), то в элементе отображается только соответствующая его размеру часть изображения
3.	Прозрачный	Если установлена галочка, то 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	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. Если установлена галочка, то в параметре **Кадрирование** в элементе отображается только соответствующая размеру элемента часть изображения
6.	Горизонтальное выравнивание (только для типа шкалы **Изотропия** или **Фиксированный**)	Выравнивание изображения по горизонтальной оси. Возможные значения: **Лево/По центру/Право** Можно привязать константу или переменную типа **VisuElemBase.VisuEnumHorizontalAlignment**
7.	Вертикальное выравнивание (только для типа шкалы **Изотропия** или **Фиксированный**)	Выравнивание изображения по вертикальной оси. Возможные значения: **Верх/По центру/Низ** Можно привязать константу или переменную типа **VisuElemBase.VisuEnumVerticalAlignment**
8.	Переменная ID изображения/ ID изображения	Строковая переменная, определяющая имя пула изображений и **ID** файла из него. Пример: **MyImagePool.MyID** В случае обращения к пулу изображений, добавленному в проект первым, имя пула можно опускать. Меняя значение этой переменной в цикле программы, можно реализовать анимацию
9.	Динамическое изображения/Версия изображения	Целочисленная переменная, характеризующая версию изображения. При изменении переменной выполняется перерисовка элемента. Возможные случаи использования: • изображение генерируется программно; • изображение меняется в процессе работы проекта (например, загружается с накопителя или циклически поступает с камеры)

7.2.6 Фрейм элемента

Рисунок 7.15 – 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.

Если у экрана, открываемого во фрейме, объявлены переменные интерфейса, то в конфигурации фрейма к ним можно привязать переменные программы. То есть с помощью фрейма можно открывать разные экземпляры одного экрана визуализации, которые будут отличаться только привязанными переменными. Пример использования интерфейса фрейма приведен в п. 10.3.3.

Рисунок 7.16 – Внешний вид элемента Фрейм

Описание типичных параметров графического примитива приведено в таблице 7.1.

Уникальные параметры примитива описаны в таблице 7.4.

Пример работы с элементом приведен в п. 10.2.2.

Таблица 7.4 – Уникальные параметры графического примитива Фрейм

№	Название параметра	Описание параметра
1.	Кадрирование	Если установлена галочка и выбран тип шкалы **Фиксировано** (пп. 3), то во фрейме отображается только соответствующая его размеру часть экрана визуализации
2.	Показать фрейм	Если установлена галочка, то вокруг фрейма отображается контур
3.	Тип шкалы	Тип масштабирования экрана визуализации в фрейме при изменении его (фрейма) размера. Возможные значения: **Изотропия** – в случае изменения размера фрейма, открытый в нем экран визуализации масштабируется с сохранением соотношения сторон; **Анизотропный** – в случае изменения размера фрейма, открытый в нем экран визуализации масштабируется без сохранения соотношения сторон; **Фиксировано** – в случае изменения 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; **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
4.	Параметры полосы прокрутки	(данный параметр доступен только при выборе в пп. 3 типа шкалы **Фиксирована и прокручивается**) 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
5.	Отключить нанесение фона	Для оптимизации производительности, статические элементы фрейма по умолчанию отображаются как фоновое изображение. Это может привести к неожиданным графическим эффектам. Если установлена галочка 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. 10.3.3)
7.	Переменная-переключатель фрейма	Целочисленная переменная, содержащая индекс отображаемого во фрейме в данный момент экрана визуализации. Меняя значение переменной, можно переключать экраны фрейма. Индекс первого экрана визуализации фрейма – **0**, второго – **1** и т. д. Порядок экранов определяется в настройках ссылок (пп. 6)

7.3 Стандартная группа элементов

7.3.1 Метка

Рисунок 7.17 – 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**.

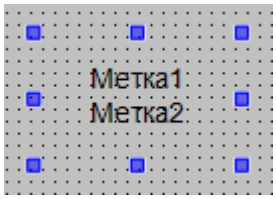


Рисунок 7.18 – Внешний вид элемента Метка

Описание типичных параметров графического примитива приведено в таблице 7.1.

7.3.2 Комбинированное окно (Целочисленное)

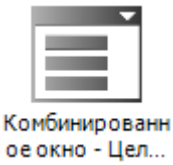


Рисунок 7.19 – Пиктограмма графического примитива Комбинированного окно (Целочисленное)

Графический примитив **Комбинированное окно (Целочисленное)** используется для реализации выбора из списка строк, состоящих из изображений (добавленных в Пул изображений) и текстов (добавленных в Список текстов).

ПРИМЕЧАНИЕ
К элементу может быть привязана переменная типа **Перечисление** с включенной поддержкой списка текстов (см. п. 5 в этой статье).

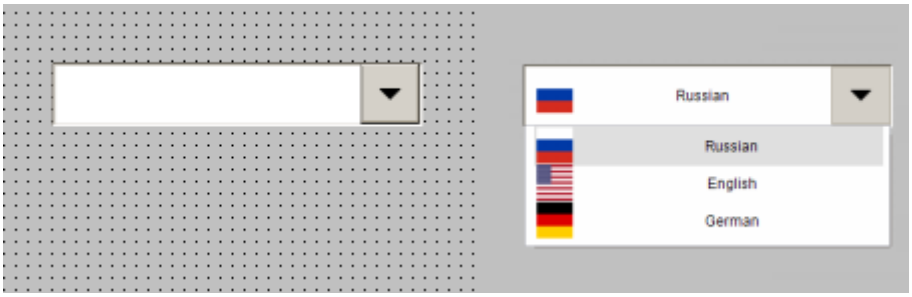


Рисунок 7.20 – Внешний вид элемента Комбинированного окно (Целочисленное)

Описание типичных параметров графического примитива приведено в таблице 7.1.

Уникальные параметры примитива описаны в таблице 7.5.

Пример работы с элементом приведен в п. 10.2.3.

Таблица 7.5 – Уникальные параметры графического примитива Комбинированное окно (Целочисленное)

№	Название параметра	Описание параметра
1.	Переменная	**Целочисленная** переменная, содержащая индекс (**ID**) 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. Вместо выбора конкретного списка можно привязать к параметру переменную типа **STRING** для переключения списков текстов элемента в процессе работы контроллера.
3.	Пул изображений	Пул изображений, содержимое которого отображается в списке элемента. **ID** из пула изображений должны соответствовать возможным значениям переменной из пп. 1
4.	**Параметры списка** При значениях **From Style** будут использованы настройки выбранного стиля визуализации	
4.1.	Высота строки	Высота строки списка в пикселях
4.2.	Число видимых строк	Число видимых строк списка. Если значение меньше, чем число элементов списка текстов из п. 2, то отображается полоса прокрутки. Можно использовать константу или целочисленную переменную
4.3.	Высота изображения	Высота изображений в списке в пикселях
4.4.	Ширина изображения	Ширина изображений в списке в пикселях
4.5.	Сдвиг изображения	Сдвиг изображений относительно левого края списка в пикселях
4.6.	Размер полосы прокрутки	Ширина полосы прокрутки в пикселях
5.	**Поддиапазон**	
5.1.	Использовать поддиапазон	Если установлена галочка, то для переменной из пп. 1 определяется диапазон возможных значений
5.2.	Начальный индекс	Начальный индекс диапазона переменной из пп. 1. Можно использовать константу или целочисленную переменную
5.3.	Конечный индекс	Конечный индекс переменной диапазона из пп. 1. Можно использовать константу или целочисленную переменную
5.4.	Отбрасывать недостающие тексты	Если установлена галочка, то пустые строки в списке не отображаются
6.	**Свойства текста**	
6.1.	Использование	Позволяет использовать для текста элементов списка настройки стиля визуализации, либо индивидуальные установки (см. пп. 6.2)
6.2.	Индивидуальные свойства текста	(данные параметры доступны только при выборе значения **Индивидуальные установки** в пп. 6.1). Свойства текста элементов списка. Описание параметров приведено в таблице 7.1, пп. 6

7.3.3 Комбинированное окно (Массив)

Рисунок 7.21 – Пиктограмма графического примитива Комбинированного окно (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.

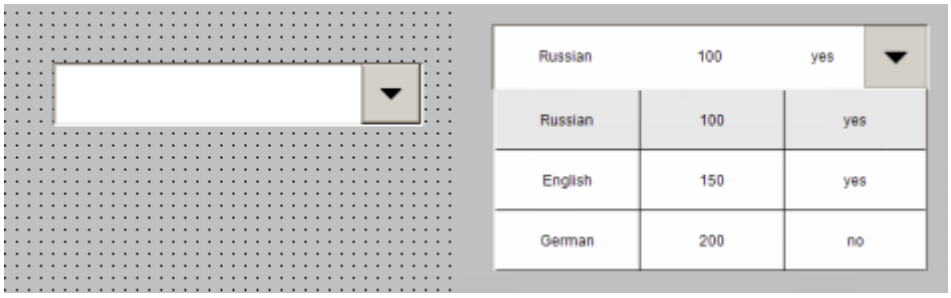


Рисунок 7.22 – Внешний вид элемента Комбинированного окно (Массив)

Описание типичных параметров графического примитива приведено в таблице 7.1.

Уникальные параметры примитива описаны в таблице 7.6.

Пример работы с элементом приведен в п. 10.2.3.

Таблица 7.6 – Уникальные параметры графического примитива Комбинированное окно (Массив)

№	Название параметра	Описание параметра
1.	Переменная	Целочисленная переменная, содержащая номер выбранного элемента списка (нумерация с **0**)
2.	Массив данных	Массив данных, отображаемых элементом. Обычно задается в виде двухмерного массива **ARRAY [0...x, 0…y]**, где x – число столбцов, y – число строк
3.	**Столбцы**	
3.1.	***Столбец*** Число столбцов определяется массивом, выбранным в пп. 2.	
3.1.1.	[№]	Если установлена галочка, то данный массив отображается в таблице элемента
3.1.2.	Ширина	Ширина данного столбца в пикселях
3.1.3.	Столбец изображения	Если установлена галочка, то 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**
3.1.6.	Использовать шаблон	Если установлена галочка, то ячейки данного столбца становятся элементами типа Прямоугольник, и появляется возможность использовать для них настройки соответствующего графического примитива. Стоит отметить, что становится доступна вкладка InputConfiguration, с помощью которой можно реализовать возможность ввода данных в таблицу. К параметрам следует привязать массивы с индексом **INDEX**. Начальный элемент массива будет определять значение параметра первой ячейки и т. д.
3.2.	Максимальный индекс массива	Целочисленная переменная, определяющая число строк, отображаемых в таблице элемента (индекс последний 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 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 таблице 7.1, пп. 6

7.3.4 Набор вкладок

Рисунок 7.23 – Пиктограмма графического примитива Набор вкладок

Графический примитив **Набор вкладок** представляет собой частный случай графического примитива Фрейм с настроенным интерфейсом переключения экранов визуализаций:

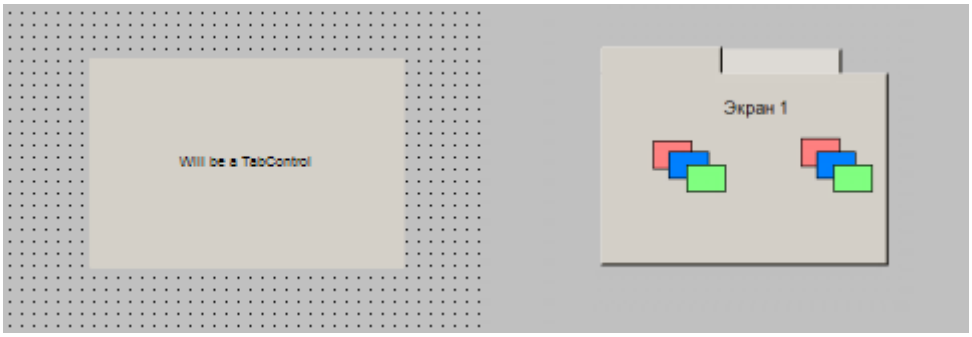


Рисунок 7.24 – Внешний вид элемента Набор вкладок

Описание типичных параметров графического примитива приведено в таблице 7.1.
Уникальные параметры примитива описаны в таблице 7:7

Таблица 7.7 – Уникальные параметры графического примитива Набор вкладок

№	Название параметра	Описание параметра
1.	Ширина вкладки	Ширина строки кнопок переключения экранов элемента в пикселях
2.	Высота вкладки	Высота кнопок переключения экранов элементов в пикселях
2.	Тип масштабирования	Тип масштабирования экрана визуализации при изменении размера его фрейма. Возможные значения: **Изотропия** – в случае изменения размера фрейма, открытый в нем экран визуализации масштабируется с сохранением соотношения сторон; **Анизотропный** – в случае изменения размера фрейма, открытый в нем экран визуализации масштабируется без сохранения соотношения сторон; **Фиксировано** – в случае изменения размера фрейма, открытый в нем экран визуализации сохраняет свой размер. Если установлена галочка в параметре **Кадрирование**, то во фрейме отображается только соответствующую размеру фрейма часть экрана визуализации; **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. В случае выбора данного типа масштабирования параметр **Кадрирование** не отображается
4.	**Параметры полосы прокрутки**	
4.1	Переменная горизонтальной позиции прокрутки	Целочисленная переменная, которая позволяет управлять прокруткой экрана фрейма по горизонтали (настройка доступна только для типа масштабирования **Фиксирована и прокручивается**).
4.2	Переменная вертикальной позиции прокрутки	Целочисленная переменная, которая позволяет управлять прокруткой экрана фрейма по вертикали (настройка доступна только для типа масштабирования **Фиксирована и прокручивается**).
5.	Отключить нанесение фона	Для оптимизации производительности, статические элементы фрейма по умолчанию отображаются как фоновое изображение. Это может привести к неожиданным графическим эффектам. Если установлена галочка у данного параметра, то оптимизация отключается
6.	Ссылки	Данный параметр определяет экраны визуализации, которые могут отображаться в данном фрейме, а также их подписи, и изображения кнопок и переменные невидимости (параметр **Invisible** типа **BOOL**). Если переменная невидимости принимает значение **TRUE**, но кнопка данной вкладки не будет отображаться в элементе. Если отключить видимость кнопок всех вкладок, то сам элемент тоже перестает отображаться
7.	Переключить переменную фрейма	Целочисленная переменная, содержащая индекс отображаемого во фрейме в данный момент экрана визуализации. Меняя значение переменной, можно переключать экраны во фрейме. Индекс первого экрана визуализации фрейма – **0**, второго – **1** и т. д. Порядок экранов определяется в настройках ссылок (пп. 6)

7.3.5 Кнопка

Рисунок 7.25 – 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**.

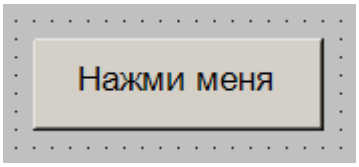


Рисунок 7.26 – Внешний вид элемента Кнопка

Описание типичных параметров графического примитива приведено в таблице 7.1.

Уникальные параметры примитива описаны в таблице 7.8.

Пример работы с элементом приведен в п. 10.2.4.

Таблица 7.8 – 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 масштабирования	Тип масштабирования изображения кнопки в случае изменения ее размера. Возможные значения: **Изотропия** – при изменении размера кнопки, ее изображение масштабируется с сохранением соотношения сторон; **Анизотропный** – при изменении размера кнопки, ее изображение масштабируется без сохранения соотношения сторон; **Фиксировано** – при изменении размера кнопки, его изображение сохраняет свой размер
2.3.	Прозрачный	Если установлена галочка, то 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 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	Переменная типа **BOOL**, определяющая текущее состояние кнопки. Возможные значения: **TRUE** – кнопка нажата **FALSE** – кнопка не нажата
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

7.3.6 Группа

Рисунок 7.27 – 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 в случае перемещения рамки), но у пользователя остается возможность настраивать параметры каждого из них.

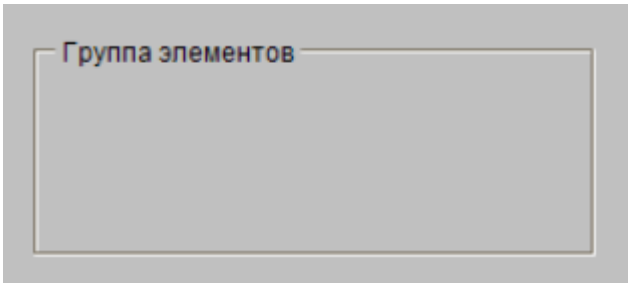


Рисунок 7.28 – Внешний вид элемента Группа

Описание типичных параметров графического примитива приведено в таблице 7.1.

7.3.7 Таблица

Рисунок 7.29 – 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 таблице 7.9). Для снятия выделения следует 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**.

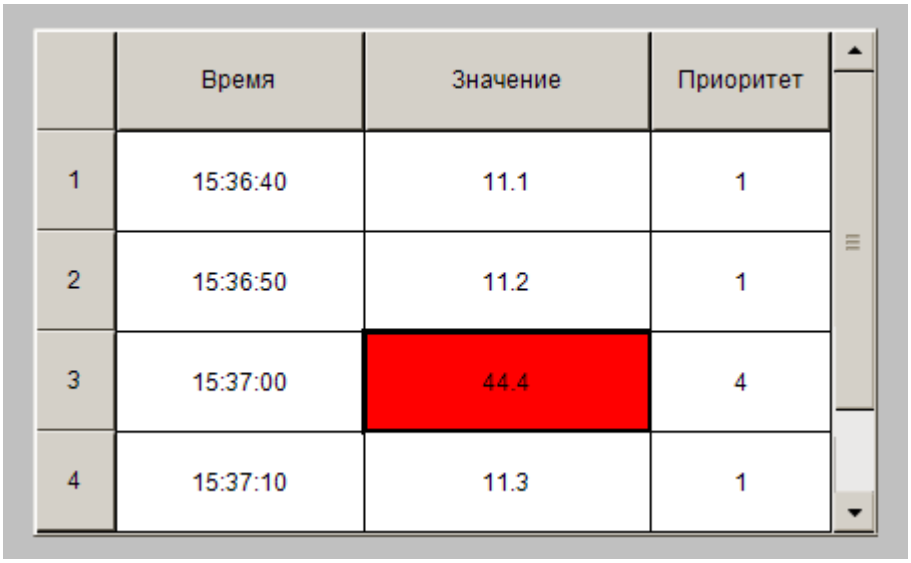


Рисунок 7.30 – Внешний вид элемента Таблица

Описание типичных параметров графического примитива приведено в таблице 7.1.

Уникальные параметры примитива описаны в таблице 7.9.

Пример работы с элементом приведен в п. 10.2.5.

Таблица 7.9 – 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 (индекс последний строки из массива, указанного в пп. 1)
3.	**Столбцы**	
3.1.	***Столбец*** Чисто столбцов определяется массивом, выбранным в пп. 1.	
3.1.1.	[№]	Если установлена галочка, то данный столбец отображается элементом
3.1.2.	Ширина	Ширина данного столбца в пикселях
3.1.3.	Столбец изображения	Если установлена галочка, то данный столбец используется для отображения изображений. Значения столбца должны иметь тип **STRING** содержать **ID** изображений из пула изображений. Если в проекте добавлено несколько пулов, то перед **ID** должно быть указано название пула, например **MyImagePool.MyImageId**
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	Если установлена галочка, то ячейки данного столбца становятся элементами типа Прямоугольник, и появляется возможность использовать для них настройки соответствующего графического примитива. Стоит отметить, что становится доступна вкладка InputConfiguration, с помощью которой можно реализовать возможность ввода данных в таблицу. К параметрам следует привязать массивы с индексом **INDEX**. Начальный элемент массива будет определять значение параметра первой ячейки и т. д.
3.1.7.	Выравнивание текста заголовка из шаблона	Если установлена галочка, то для заголовка используется выравнивание из шаблона (см. пп. 3.1.6)
3.2.	Показать заголовок строки	Если установлена галочка, то в таблице отображается строка заголовков
3.3.	Показать заголовок столбца	Если установлена галочка, то 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

Продолжение Табл.7.9

№	Название параметра	Описание параметра
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	Если установлена галочка, то вокруг выделенных элементов таблицы отображается контур
4.4.	Переменная для выбранного столбца	Переменная типа **INT,** в которую записывается номер выделенного столбца. Может использоваться для выделения столбца из кода программы
4.5.	Переменная для выбранной строки	Переменная типа **INT**, в которую записывается номер выделенной строки. Может использоваться прокрутки таблицы из кода программы. **Для корректной работы высота строки таблицы (пп. 3.4) должна быть кратна высоте самой таблицы**
4.6.	Переменная для выбора действительного столбца	Переменная типа **BOOL**, принимающая значение **TRUE** в случае выделения любого столбца таблицы (в том числе при выделении любой ячейки)
4.7.	Переменная для выбора действительной строки	Переменная типа **BOOL**, принимающая значение **TRUE** в случае выделения любой строки таблицы (в том числе при выделении любой ячейки)

7.3.8 Текстовое поле

Рисунок 7.31 – Пиктограмма графического примитива Текстовое поле

Графический примитив **Текстовое поле** используется для отображения информации. В отличие от графического примитива Метка, данный примитив способен отображать переменные и имеет фон.

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

Рисунок 7.32 – Внешний вид элемента Текстовое поле

Описание типичных параметров графического примитива приведено в таблице 7.1.

Уникальные параметры примитива описаны в таблице 7.10.

Таблица 7.10 – 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
2.	**Конфигурация выделения и знака вставки**	
2.1.	Позиция каретки	Целочисленная переменная, которая соответствует позиции курсора в элементе. **0** – курсор не установлен, **1** – курсор установлен перед первым символом и т. д. Переменная доступна для чтения и записи
2.2.	Начало выделения	Целочисленная переменная, которая соответствует номеру символа, с которого начинается выделение текста. Переменная доступна для чтения и записи
2.3.	Конец выделения	Целочисленная переменная, которая соответствует номеру символа, перед которым заканчивается выделение текста. Переменная доступна для чтения и записи
2.4.	Все выбранные	Переменная типа **BOOL**, которая определяет режим выделения. Возможные значения: **FALSE** – для выделения используются значения параметров пп. 2.2 и 2.3

		TRUE – выделяется весь текст элемента

7.3.9 Полоса прокрутки

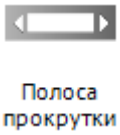


Рисунок 7.33 – Пиктограмма графического примитива Полоса прокрутки

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.

Рисунок 7.34 – Внешний вид элемента Полоса прокрутки

Описание типичных параметров графического примитива приведено в таблице 7.1.
Уникальные параметры примитива описаны в таблице 7.11.

Таблица 7.11 – Уникальные параметры графического примитива Полоса прокрутки

№	Название параметра	Описание параметра
1.	Значение	Целочисленная переменная, значение которой 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.	Дискретность перемещения	Числовая переменная, определяющая шаг изменения переменной, управляемой элементом, при нажатии на полосу прокрутки) (не влияет на кнопки **«Больше»/«Меньше»**)
5.	Прокрутка выполнена	Если установлена галочка, то параметр **Размер страницы** игнорируется. В случае нажатия 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**

7.3.10 Бегунок

Рисунок 7.35 – Пиктограмма графического примитива Бегунок

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

Рисунок 7.36 – Внешний вид элемента Бегунок

Описание типичных параметров графического примитива приведено в таблице 7.1.
Уникальные параметры примитива описаны в таблице 7.12.

Таблица 7.12 – Уникальные параметры графического примитива Бегунок

№	Название параметра	Описание параметра
1.	Переменная	Числовая переменная, чье значение управляется (изменяется) элементом
2.	Дискретность перемещения	Числовая переменная, определяющая шаг изменения переменной, управляемой элементом (при нажатии на элемент). **Примечание**: нажатие в области самого «ползунка» элемента (в зоне нескольких пикселей рядом с ним) будут обработаны – то есть таким образом получится изменить значение переменной (пп. 1) на несколько единиц независимо от значения дискретности перемещения
3.	Прокрутка выполнена	Если установлена галочка, то параметр **Размер страницы** игнорируется. В случае нажатия на любую точку бегунка, переменная элемента принимает соответствующее значение
4.	**Шкала**	
4.1.	Показать шкалу	Если установлена галочка, то рядом с элементом отображается шкала с делениями
4.2.	Начало шкалы	Точка начала шкалы (минимальное отображаемое значение переменной элемента). Можно использовать константу или целочисленную переменную
4.3.	Конец шкалы	Точка конца шкалы (максимальное отображаемое значение переменной элемента). Можно использовать константу или целочисленную переменную
4.4.	Основная шкала	Цена деления основной шкалы. Можно использовать константу или целочисленную переменную
4.5.	Подшкала	Цена деления дополнительной шкалы. Можно использовать константу или целочисленную переменную
4.6.	Формат шкалы	Спецификатор формата вывода делений шкалы
4.9.	Высота шкалы	Расположение шкалы в процентах от нижней границы элемента (**0** – на нижней границе, **100** – на верхней границе). Настройка влияет размер бегунка

Продолжение Табл. 7.12

№	Название параметра	Описание параметра
5.	**Линейка**	
5.1.	Тип диаграммы	Место отображения шкалы элемента. Шкала отображается только если установлена галочка в пп. 4.1. Зависит от ориентации элемента. Возможные значения при горизонтальной ориентации: **Слева/Справа/Слева и справа** Возможные значения при горизонтальной 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**

7.3.11 Управление вращением

Рисунок 7.37 – Пиктограмма графического примитива Управление вращением

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

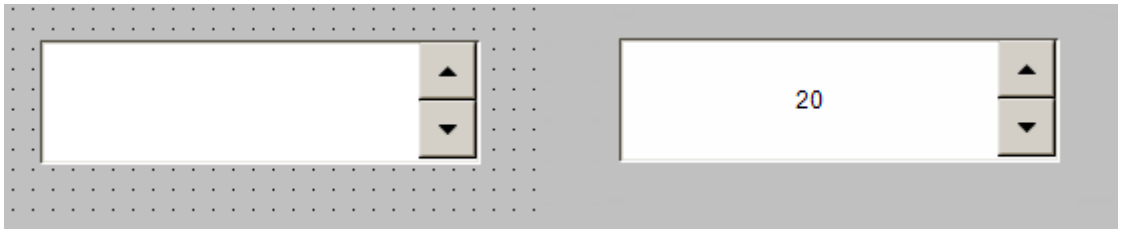


Рисунок 7.38 – Внешний вид элемента Управление вращением

Описание типичных параметров графического примитива приведено в таблице 7.1.
Уникальные параметры примитива описаны в таблице 7.13.

Таблица 7.13 – Уникальные параметры графического примитива Управление вращением

№	Название параметра	Описание параметра
1.	Переменная	Числовая переменная, значение которой 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.	**Поддиапазон**	
4.1.	Минимальное значение	Переменная, определяющая нижнюю границу значений переменной, управляемой элементом
4.2.	Конечный индекс	Переменная, определяющая верхнюю границу значений переменной, управляемой элементом

7.3.12 Невидимый ввод

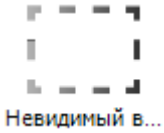


Рисунок 7.39 – Пиктограмма графического примитива Невидимый ввод

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

Рисунок 7.40 – Внешний вид элемента Невидимый ввод

Элемент имеет один уникальный параметр – **Used as pointing area**. Если элемент размещен в диалоге и эта галочка установлена – то диалог можно будет перемещать по экрану, зажав **ЛКМ** в области элемента.

Описание типичных параметров графического примитива приведено в таблице 7.1.

7.3.13 Радио-кнопка

Рисунок 7.41 – 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.

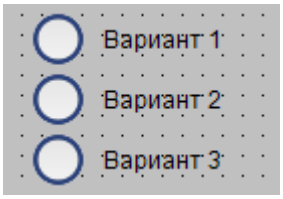
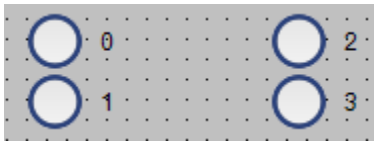
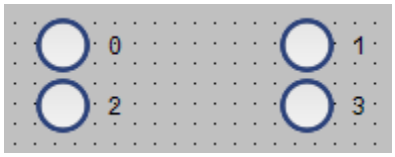


Рисунок 7.42 – Внешний вид элемента Радио-кнопка

Описание типичных параметров графического примитива приведено в таблице 7.1.

Уникальные параметры примитива описаны в таблице 7.14.

Таблица 7.14 – 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	Выбору каждой из кнопок соответствует значение целочисленной переменной. Значение «**0**» всегда соответствует выбору верхней левой кнопки. Остальные значения определяются данным параметром: **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.)  **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.) 
4.	Размер рамки	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 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 таблице 7.1, пп. 6
7.	**Параметры радио-кнопки/Радио-кнопка** В данной 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

7.3.14 Кнопка-флажок

Рисунок 7.43 – Пиктограмма графического примитива Кнопка-флажок

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

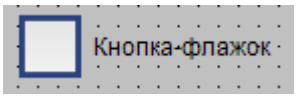


Рисунок 7.44 – Внешний вид элемента Кнопка-флажок

Описание типичных параметров графического примитива приведено в таблице 7.1.

Уникальные параметры примитива описаны в таблице 7.15.

Таблица 7.15 – Уникальные параметры графического примитива Кнопка-флажок

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

7.3.15 Индикатор выполнения

Рисунок 7.45 – Пиктограмма графического примитива Индикатор выполнения

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

Рисунок 7.46 – Внешний вид элемента Индикатор выполнения

Описание типичных параметров графического примитива приведено в таблице 7.1.
Уникальные параметры примитива описаны в таблице 7.16.

Таблица 7.16 – Уникальные параметры графического примитива Индикатор выполнения

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

7.4 Элементы управления измерением

7.4.1 Отображение линейки

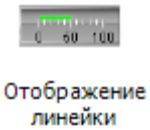


Рисунок 7.47 – Пиктограмма графического примитива Отображение линейки

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

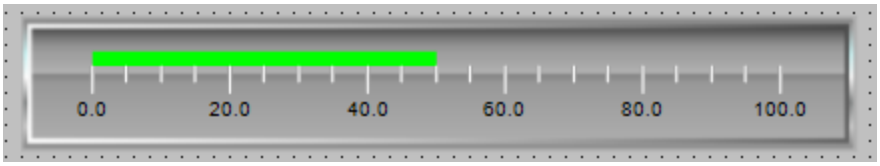


Рисунок 7.48 – Внешний вид элемента Отображение линейки

Описание типичных параметров графического примитива приведено в таблице 7.1.

Уникальные параметры примитива описаны в таблице 7.17.

Пример работы с элементом приведен в п. 10.2.6.

Таблица 7.17 – Уникальные параметры графического примитива Отображение линейки

8	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	Если установлена галочка, то 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**
3.4	Оптимальный размер полоски	В случае наличия галочки основную часть элемента будет составлять полоса заполнения
4.	**Шкала**	
4.1.	Начало шкалы	Точка начала шкалы (минимальное отображаемое значение переменной элемента). Можно использовать константу или целочисленную переменную
4.2.	Конец шкалы	Точка конца шкалы (максимальное отображаемое значение переменной элемента). Можно использовать константу или целочисленную переменную
4.3.	Основная шкала	Цена деления основной шкалы. Можно использовать константу или целочисленную переменную
4.4.	Подшкала	Цена деления дополнительной шкалы. Можно использовать константу или целочисленную переменную
4.5.	Ширина линии шкалы	Ширина делений шкалы в пикселях
4.6.	Цвет шкалы	Цвет делений шкалы элемента
4.7.	Шкала в 3D	Если установлена галочка, то шкала отображается с тенью
4.8.	Фрейм	Если установлена галочка, то вокруг элемента отображается контур

Продолжение Табл. 7.17

№	Название параметра	Описание параметра
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.	Горизонтальное масштабирование	Коэффициент масштабирования элемента по горизонтали. Положительным значениям соответствует сжатие, отрицательным – растягивание
6.4.	Вертикальное масштабирование	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	Если установлена галочка, то вокруг полосы заполнения отображается фоновый прямоугольник
7.3.	Цвет фрейма	Цвет контура элемента. Используется в случае наличия 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	Если установлена галочка, то цвет **всей** полосы заполнения определяется цветом цветовой области, в которой находится текущее значение переменной элемента (см. пп. 7.7)
7.5.	Использовать градиентный цвет	Если установлена галочка, то 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 представляют 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). ***Следует обратить внимание*** на настройку **Переключить весь цвет** (см. пп. 7.4)	
7.7.1.	Начало области	Точка начала цветовой области на полосе заполнения. Можно использовать константу или целочисленную переменную
7.7.2.	Конец области	Точка конца цветовой области на полосе заполнения. Можно использовать константу или целочисленную переменную
7.7.3.	Цвет	Цвет области

7.4.2 Стрелочный индикатор (90°/180°/360°)

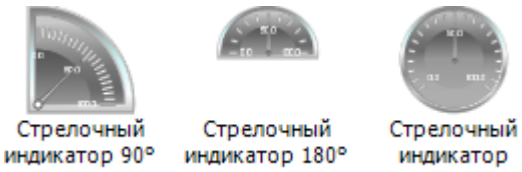


Рисунок 7.49 – Пиктограммы графического примитива Стрелочный индикатор

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

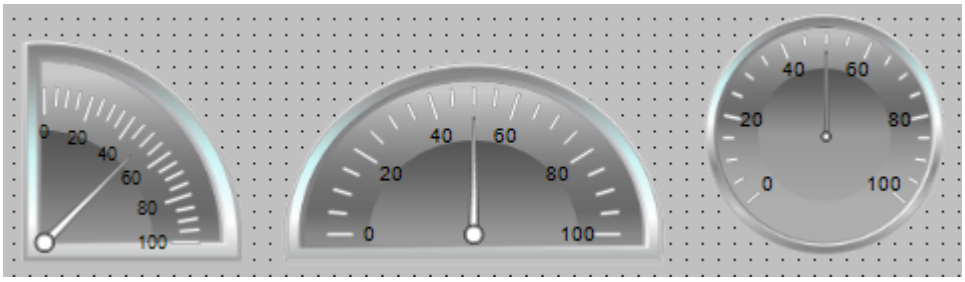


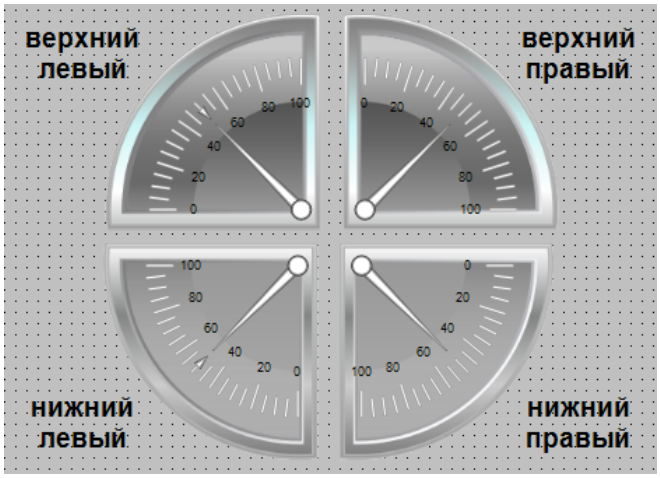
Рисунок 7.50 – Внешний вид элемента Стрелочный индикатор

Описание типичных параметров графического примитива приведено в таблице 7.1.
Уникальные параметры примитива описаны в таблице 7.18.

Таблица 7.18 – 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	Если установлена галочка, то фоновое изображение не перерисовывается при изменении динамических частей элемента

Продолжение Табл. 7.18

№	Название параметра	Описание параметра
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	Данный параметр доступен только для графических примитивов **Стрелочный индикатор 90° и Стрелочный индикатор 180°** и определяет их внешний вид. Возможные значения для графического примитива **Стрелочный индикатор 90°:** 

Продолжение Табл. 7.18

№	Название параметра	Описание параметра
3.3. (90°/ 180°)	Угловой диапазон	Возможные значения для графического примитива **Стрелочный индикатор 180°:**
3.4. (360°)	Начало стрелки	Данный параметр доступен только для графического примитива **Стрелочный индикатор 360°.** Угол между положительным направлением горизонтальной оси и началом шкалы элемента в градусах. Положительное значение соответствует движению против часовой стрелки, отрицательное – по часовой
3.5. (360°)	Конец стрелки	Данный параметр доступен только для графического примитива **Стрелочный индикатор 360°.** Угол между положительным направлением горизонтальной оси и концом шкалы элемента в градусах. Положительное значение соответствует движению против часовой стрелки, отрицательное – по часовой
3.6.	Дополнительная стрелка	Если установлена галочка, то конец стрелки дополнительно выделяется на шкале

Продолжение Табл. 7.18

№	Название параметра	Описание параметра
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). Можно использовать константу или целочисленную переменную
4.4.	Конец шкалы	Точка конца шкалы (максимальное отображаемое значение переменной элемента). Можно использовать константу или целочисленную переменную
4.5.	Основная шкала	Цена деления основной шкалы. Можно использовать константу или целочисленную переменную
4.6.	Подшкала	Цена деления дополнительной шкалы. Можно использовать константу или целочисленную переменную
4.7.	Ширина линии шкалы	Ширина делений шкалы в пикселях
4.8.	Цвет шкалы	Цвет делений шкалы элемента
4.9.	Шкала в 3D	Если установлена галочка, то шкала отображается с тенью
4.10.	Показать шкалу	Если галочка не установлена, то шкала элемента не отображается
4.11.	Рамка внутри	Если галочка установлена, то нижняя сторона шкалы элемента выделяется контуром
4.12.	Рамка снаружи	Если галочка установлена, то 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** В случае выбора **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	Данный параметр доступен только в случае выбора фонового изображения (см. пп. 2.2.1). Расстояние между шкалой и центром элемента в пикселях

Продолжение Табл. 7.18

№	Название параметра	Описание параметра
6.6.	Длина шкалы	Данный параметр доступен только в случае выбора фонового изображения (см. пп. 2.2.1). Длина (в проекции на горизонтальную ось) шкалы элемента в пикселях
7.	**Цвета**	
7.1.	Долговременные цветовые области	Если установлена галочка, то цветовые области (см. пп. 7.3) отображаются независимо от значения переменной элемента
7.2.	Использовать цветовые области для шкалы	Если установлена галочка, то цветовая область формируется путем заливки делений элемента заданным цветом
7.3.	***Цветовые области*** Цветовые области представляют 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 **Долговременные цветовые области** (см. пп. 7.1).	
7.3.1	Начало области	Точка начала цветовой области на шкале элемента. Можно использовать константу или целочисленную переменную
7.3.2.	Конец области	Точка конца цветовой области на шкале элемента. Можно использовать константу или целочисленную переменную
7.3.3.	Цвет	Цвет области

7.4.3 Потенциометр

Рисунок 7.51 – Пиктограмма графического примитива Потенциометр

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

Рисунок 7.52 – Внешний вид элемента Потенциометр

Описание типичных параметров графического примитива приведено в таблице 7.1.

Уникальные параметры примитива описаны в таблице 7.18 (так как 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. 10.2.6.

7.4.4 Гистограмма

Рисунок 7.53 – Пиктограмма графического примитива Гистограмма

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

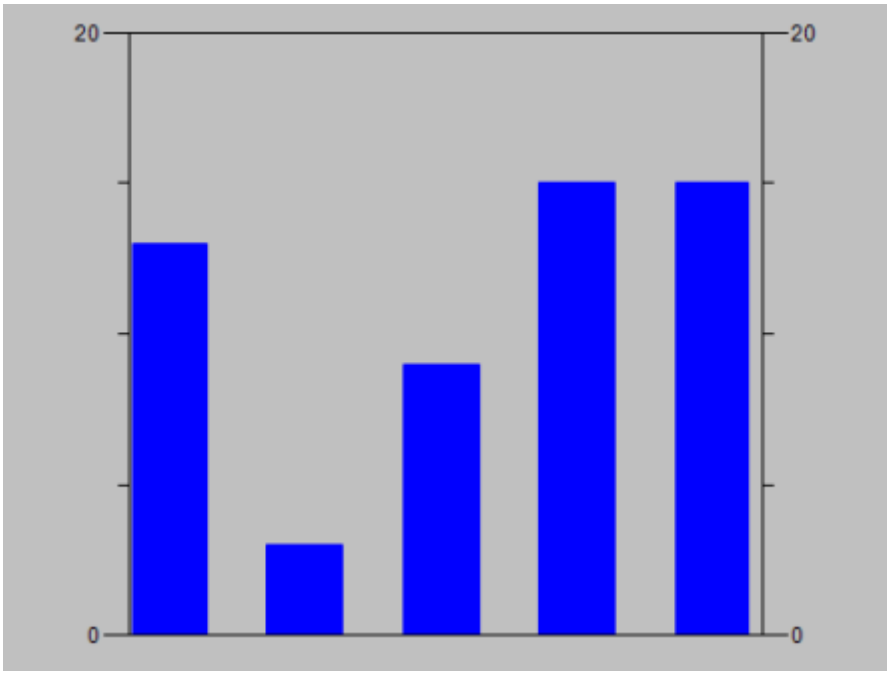
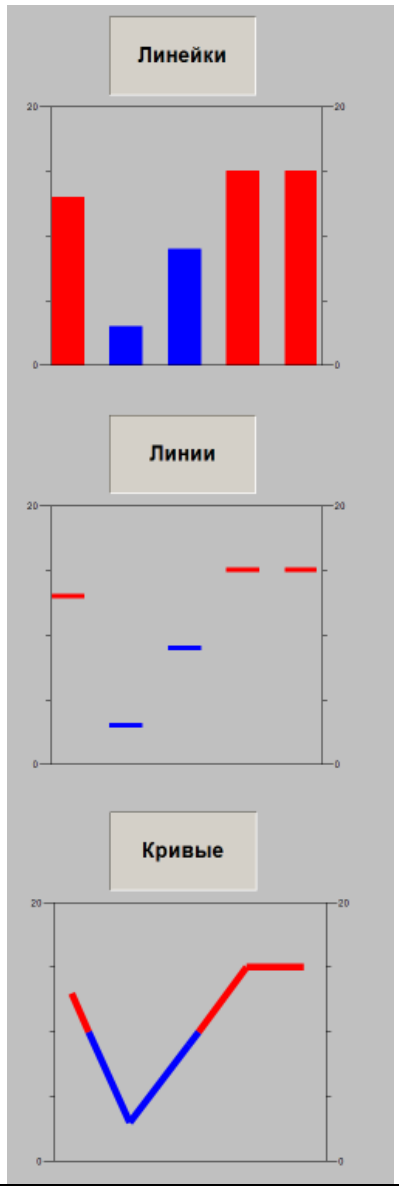


Рисунок 7.54 – Внешний вид элемента Гистограмма

Описание типичных параметров графического примитива приведено в таблице 7.1.
Уникальные параметры примитива описаны в таблице 7.19.
Пример работы с элементом приведен в п. 10.2.7.

Таблица 7.19 – 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	Тип отображения значений массива. Возможные значения: **Линейки/Линии/Кривые** 
4.	Ширина линии	Данный параметр доступен только для **Типа отображения** (пп. 3) **Кривые** и определяет ширину линий в пикселях
5.	Показывать горизонтальные линии	Если установлена галочка, то 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**

Продолжение Табл. 7.19

№	Название параметра	Описание параметра
7.	**Шкала**	
7.1.	Начало шкалы	Точка начала шкалы (минимальное отображаемое значение переменной элемента). Можно использовать константу или целочисленную переменную
7.2.	Конец шкалы	Точка конца шкалы (максимальное отображаемое значение переменной элемента). Можно использовать константу или целочисленную переменную
7.3.	Основная шкала	Цена деления основной шкалы. Можно использовать константу или целочисленную переменную
7.4.	Подшкала	Цена деления дополнительной шкалы. Можно использовать константу или целочисленную переменную
7.5.	Цвет шкалы	Точка начала шкалы (минимальное отображаемое значение переменной элемента)
7.6.	Базовая линия	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 базовой линии. 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	Если установлена галочка, то на гистограмме отображаются цветовые области (см. пп. 9.4)

Продолжение Табл. 7.19

№	Название параметра	Описание параметра
9.4.	***Цветовые области*** 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. Можно использовать константу или целочисленную переменную
9.4.2.	Конец области	Точка конца цветовой области по вертикальной оси гистограммы. Можно использовать константу или целочисленную переменную
9.4.3.	Цвет	Цвет области

7.5 Индикаторы/Переключатели/Изображения

7.5.1 Переключатель изображения

Рисунок 7.55 – Пиктограмма графического примитива Переключатель изображения

Графический примитив **Переключатель изображения** представляет собой переключатель с настраиваемыми изображениями для состояний **TRUE/FALSE** привязанной к элементу переменной типа **BOOL**.

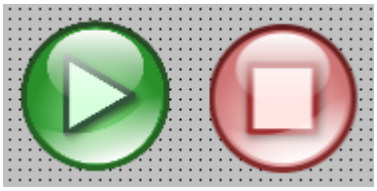


Рисунок 7.56 – Внешний вид элемента Переключатель изображения (состояния TRUE/FALSE)

Описание типичных параметров графического примитива приведено в таблице 7.1.

Уникальные параметры примитива описаны в таблице 7.20.

Пример работы с элементом приведен в п. 10.2.8.

Таблица 7.20 – Уникальные параметры графического примитива Переключатель изображения

№	Название параметра	Описание параметра
1.	Переменная	Переменная типа **BOOL**, которая меняет свое значение при нажатии на элемент и 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	Если установлена галочка, то 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.	Изображение при нажатии	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 в случае нажатия на него до момента отпускания. Может использоваться при необходимости отображать промежуточное состояние элемента.

Продолжение Табл. 7.20

№	Название параметра	Описание параметра
2.5.	Изотропный тип	Тип масштабирования изображения при изменении размеров элемента. Возможные значения: **Изотропия** – в случае изменения размеров элемента, изображение масштабируется с сохранением соотношения сторон; **Анизотропный** – в случае изменения размеров элемента, изображение масштабируется без сохранения соотношения сторон; **Фиксированный** – в случае изменения размеров элемента, изображение сохраняет свой размер
2.6.	Горизонтальное выравнивание	Выравнивание изображения относительно элемента по горизонтали. Возможные значения: **Лево/По центру/Право**
2.7.	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 (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**. Если в данном параметре установлена галочка, то 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**

7.5.2 Индикатор

Рисунок 7.57 – Пиктограмма графического примитива Индикатор

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

Рисунок 7.58 – Внешний вид элемента Индикатор (состояния TRUE/FALSE)

Описание типичных параметров графического примитива приведено в таблице 7.1.
Уникальные параметры примитива описаны в таблице 7.21.
Пример работы с элементом приведен в п. 10.2.8.

Таблица 7.21 – 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	Если установлена галочка, то 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	Если установлена галочка в параметре **Прозрачный**, то указанный в данном параметре цвет не будет отображаться
2.3.	Изотропный тип	Тип масштабирования изображения в случае изменения размеров элемента. Возможные значения: **Изотропия** – в случае изменения элемента, изображение масштабируется с сохранением соотношения сторон; **Анизотропный** – в случае изменения размеров элемента, изображение масштабируется без сохранения соотношения сторон; **Фиксированный** – в случае изменения элемента, изображение сохраняет свой размер
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**

7.5.3 Переключатели/Выключатели

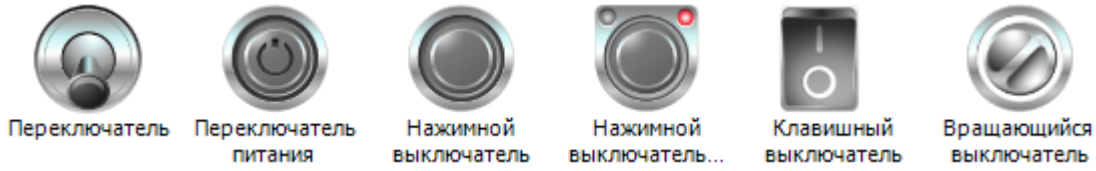


Рисунок 7.59 – Пиктограммы графических примитивов Переключатели/Выключатели

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

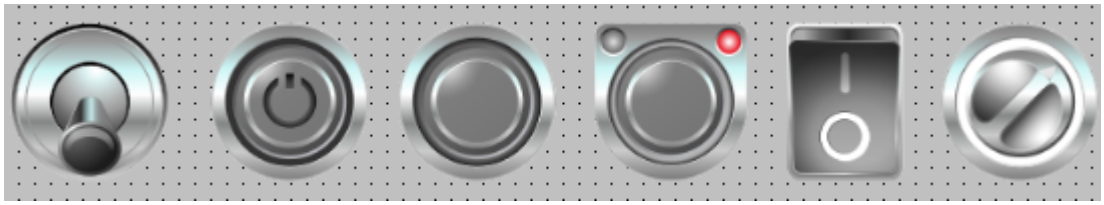


Рисунок 7.60 – Внешний вид элементов Переключатели/Выключатели

Описание типичных параметров графического примитива приведено в таблице 7.1.

Уникальные параметры примитива описаны в таблице 7.22.

Пример работы с элементом приведен в п. 10.2.8.

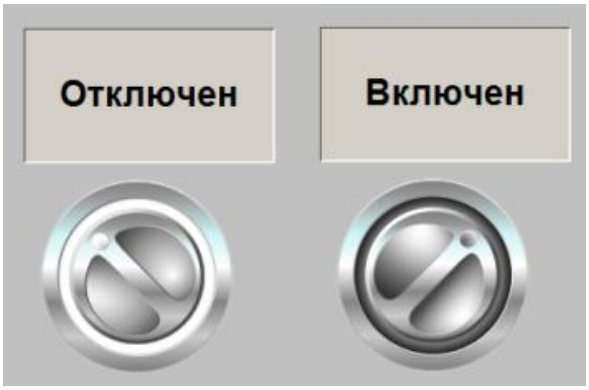
Таблица 7.22 – Уникальные параметры графических примитивов Переключатели/Выключатели

№	Название параметра	Описание параметра
1.	Переменная	Переменная типа **BOOL**, которая управляется элементом
2.	**Параметры изображения**	
2.1.	Прозрачный	Если установлена галочка, то 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 **Прозрачны**й не будет отображаться
2.3.	Изотропный тип	Тип масштабирования изображения в случае изменения размеров элемента. Возможные значения: **Изотропия** – в случае изменения элемента, изображение масштабируется с сохранением соотношения сторон; **Анизотропный** – в случае изменения размеров элемента, изображение масштабируется без сохранения соотношения сторон; **Фиксированный** – в случае изменения размеров элемента, изображение сохраняет свой размер
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**

Продолжение Табл. 7.22

№	Название параметра	Описание параметра
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 уже имеет значение **TRUE** (например, это ее начальное значение), то оно будет переключено на **FALSE**. Если установлена галочка 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**
5.	Ориентация	Данный параметр доступен только для примитива **Вращающийся выключатель**. Определяет внешний вид элемента. Возможные значения: **Наверху/Сбоку**

Продолжение Табл. 7.22

№	Название параметра	Описание параметра
6.	Изменение цвета	Данный параметр доступен только для примитива **Вращающийся выключатель**. Если установлена галочка, то в состоянии включен (**TRUE**) элемент меняет свой цвет 
7.	**Фон**	
7.1.	Изображение	Цвет элемента. Возможные значения: **Yellow/Red/Green/Blue/Gray**

7.6 Специальные элементы управления

7.6.1 Трассировка

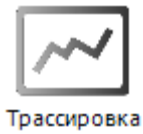


Рисунок 7.61 – Пиктограмма графического примитива Трассировка

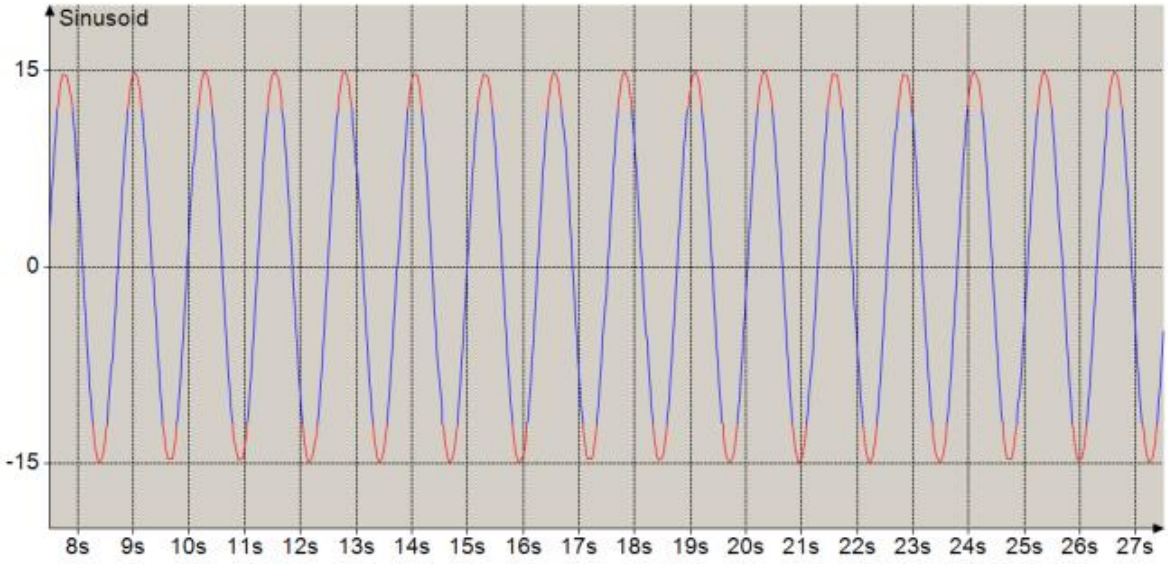


Рисунок 7.62 – Внешний вид элемента Трассировка

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 можно отобразить на графике значения за последние 19 секунд.

Во время добавления графического элемента **Трассировка** на экран визуализации, автоматически создается компонент **Трассировка** с названием:

<Имя экрана визуализации>_Trace<Номер трассировки>

Компонент **Трассировка** доступен только в случае выделения графического элемента и содержит настройки его записи. Компонент также может быть добавлен в проект независимо от графического элемента (**Application – Добавить объект – Трассировка**) – и 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.

ПРИМЕЧАНИЕ
Примитив **Трассировка** не рассчитан на хранение истории значений переменных – для решения данной задачи рекомендуется использовать графический примитив Тренд.

Описание типичных параметров графического примитива приведено в таблице 7.1.

Уникальные параметры примитива описаны в таблице 7.23.

Настройки компонента **Трассировка** приведены в таблице 7.24.

Пример работы с элементом приведен в п. 10.2.9.

Таблица 7.23 – Уникальные параметры графического примитива Трассировка

№	Название параметра	Описание параметра
1.	Показать курсор	Если установлена галочка, то после остановки трассировки по триггеру отображается курсор (метка со значением переменной)
2.	Заменить существующую трассировку	Если установлена галочка, то в случае наличия на целевом устройстве файла трассировки с названием, идентичным названию данной трассировки, при загрузке текущего проекта он будет перезаписан
3.	Числовой формат	Спецификатор формата вывода переменных, отображаемых в подсказке элемента
4.	**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	
4.1.	Сброс триггера	Переменная типа **BOOL**. Когда она принимает значение **TRUE**, триггер трассировки (см. таблицу 7.24, пп. 1) сбрасывается
4.2.	Запустить трассировку	Переменная типа **BOOL**. Когда она принимает значение **TRUE**, остановленная трассировка запускается. Параметры 4.2–4.3 являются частным случаем триггера (см. таблицу 7.24, пп. 1)
4.3.	Остановить трассировку	Переменная типа **BOOL**. Когда она принимает значение **TRUE**, трассировка останавливается
4.4.	***Сохранить трассировку в файл***	
4.4.1.	Сохранить трассировку	Переменная типа **BOOL**. Когда она принимает значение **TRUE**, конфигурация трассировки сохраняется в файл
4.4.2.	Имя файла	Переменная типа **STRING**, определяющая путь к файлу, в который сохраняется конфигурация трассировки
4.5.	***Загрузить трассировку из файла***	
4.5.1.	Загрузить трассировку	Переменная типа **BOOL**. Когда она принимает значение **TRUE**, конфигурация трассировки загружается из файла
4.5.2.	Имя файла	Переменная типа **STRING**, определяющая путь к файлу, из которого загружается конфигурации трассировки

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

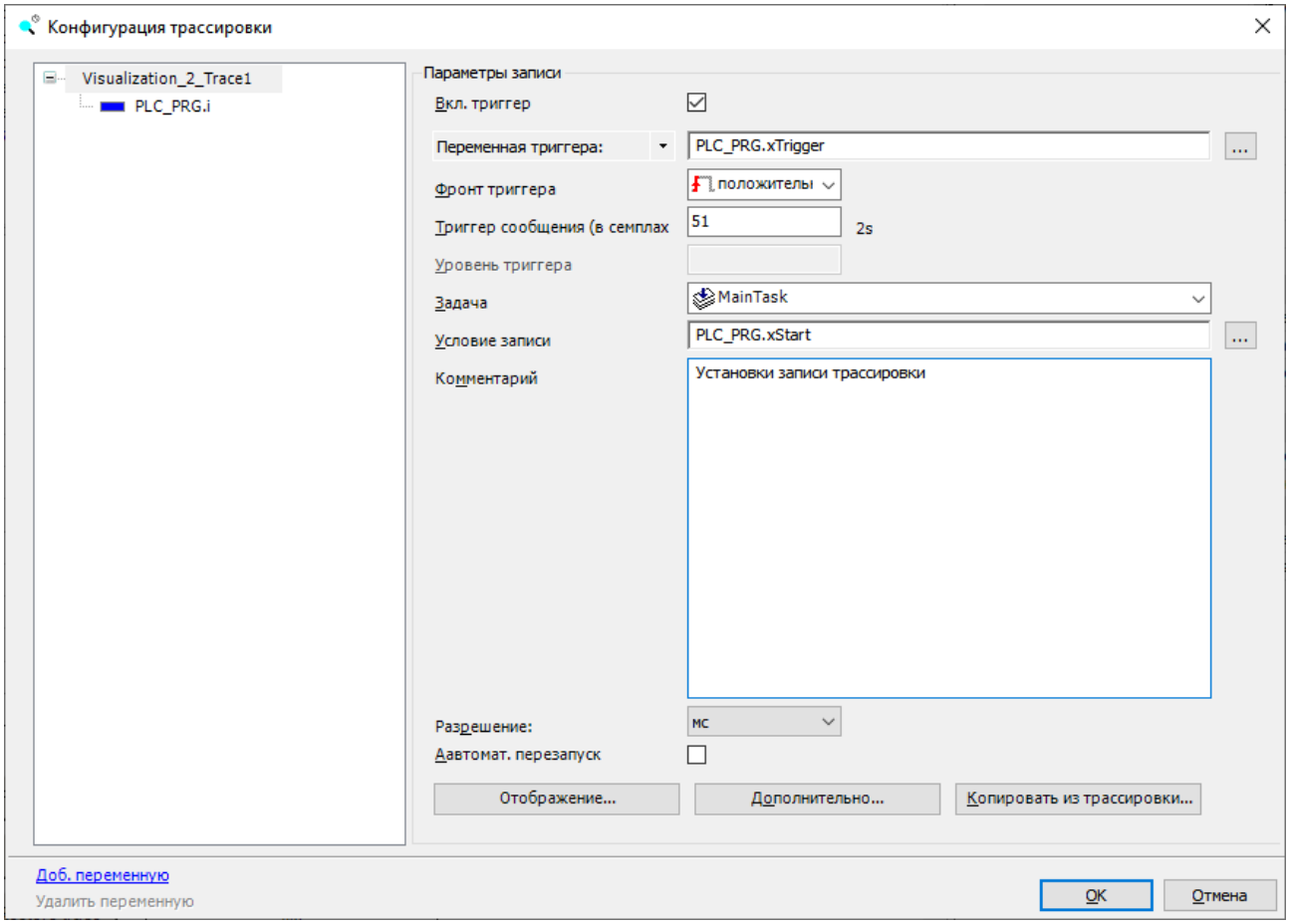


Рисунок 7.63 – Конфигурация трассировки – Параметры записи

Таблица 7.24 – Конфигурация элемента Трассировка

№	Название параметра	Описание параметра
1.	**Параметры записи**	
1.1.	Вкл. триггер	Если установлена галочка, то 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 Отрицательный – трассировка останавливается, когда триггер типа **BOOL** принимает значение **FALSE** или целочисленный триггер в процессе уменьшения достигает значения, указанного в пп. 1.5. Оба – трассировка останавливается, когда триггер типа **BOOL** меняет свое значение или целочисленный триггер в процессе увеличения/уменьшения достигает значения, указанного в пп. 1.5
1.4.	Триггер сообщения	Время после срабатывания триггера (в секундах, пересчитанное из количества семплов), через которое трассировка останавливается
1.5.	Уровень триггера	Данный параметр доступен только в случае указания в пп. 1.2. целочисленной переменной. Значение, при достижении которого срабатывает целочисленный триггер (см. пп. 1.3)
1.6.	Задача	Задача, цикл которой определяет частоту записи значений переменных трассировки
1.7.	Условие записи	Переменная типа **BOOL**. Пока она имеет значение **TRUE** – то трассировка находится в работе и на график добавляются данные. **FALSE** – трассировка останавливается. Если переменная не привязана – то трассировка всегда находится в работе (поведение по умолчанию)
1.8.	Комментарий	Комментарий к данной трассировке
1.9.	Разрешение	Минимальная единица отображения времени на трассировке. Возможные значения: **миллисекунды/микросекунды** См. пример правильной настройки параметра.
1.10.	Автомат. перезапуск	Если установлена галочка, то конфигурация трассировки сохраняется в файл при старте приложения

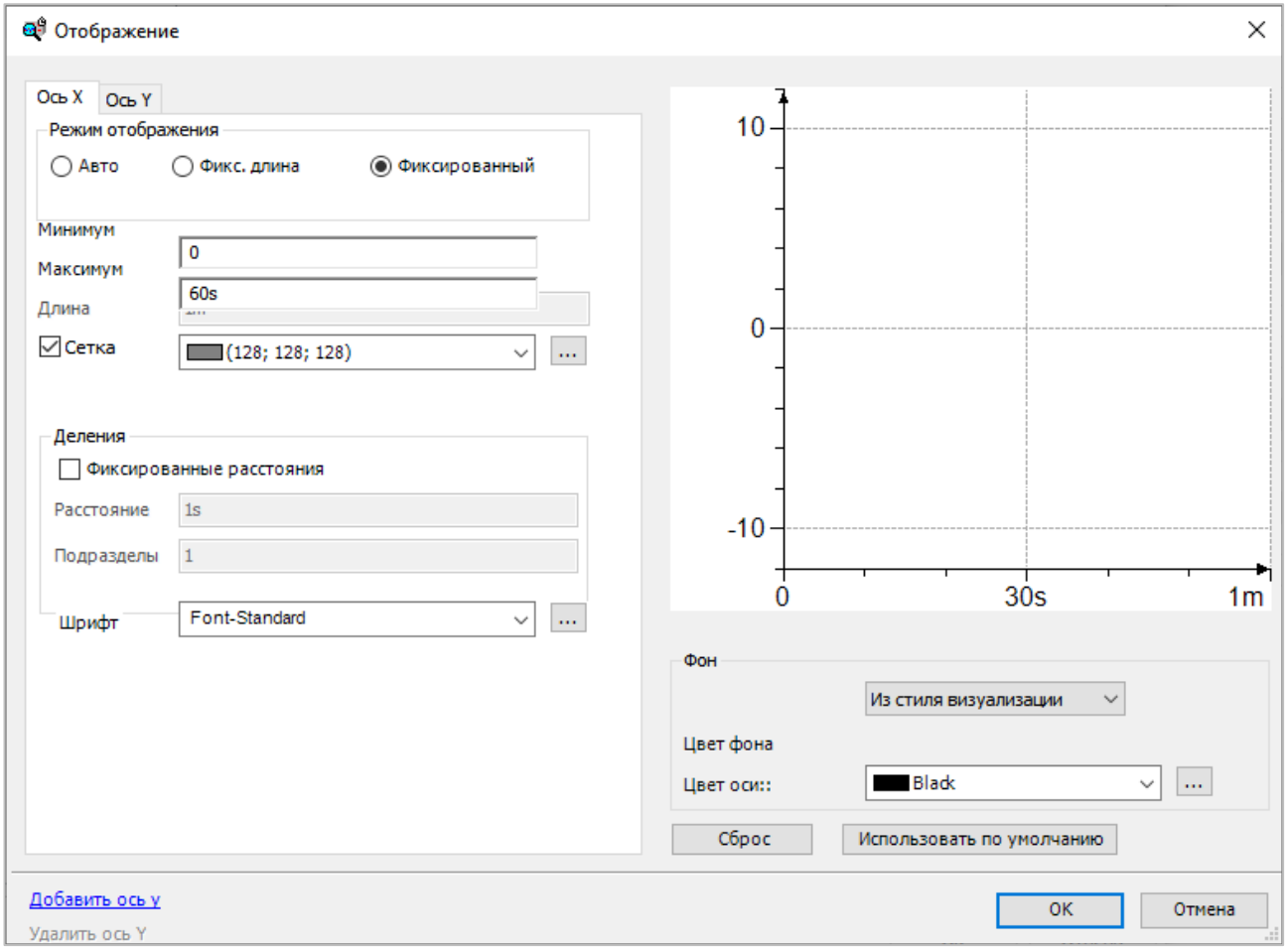


Рисунок 7.64 – Конфигурация трассировки – Отображение

ПРИМЕЧАНИЕ

Данная вкладка отображается некорректно в русскоязычной версии **CODESYS V3.5 SP11 Patch 5**. Для изменения языка интерфейса следует в меню **Инструменты** выбрать пункт **Опции**, перейти во вкладку пункт **Международные установки** и изменить язык интерфейса на английский.

Продолжение Табл. 7.24

№	Название параметра	Описание параметра
2.	**Отображение** 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** (только для оси X) – позволяет указать длину шкалы трассировки по оси X. ***Следует обратить внимание*** на пп. 1.10. **Фиксированный** – позволяет указать начало и конец шкалы трассировки
2.2.	Минимум	Данный параметр доступен только в случае выбора режима отображения **Фиксированный**. Минимум шкалы трассировки. Для оси Y можно вместо ввода числа привязать целочисленную переменную
2.3.	Максимум	Данный параметр доступен только в случае выбора режима отображения **Фиксированный**. Максимум шкалы трассировки. Для оси Y можно вместо ввода числа привязать целочисленную переменную
2.4.	Длина	Данный параметр доступен только для **оси X** при выборе режима отображения **Фиксированная длина**. Длина шкалы трассировки. ***Следует обратить внимание*** на пп. 1.10
2.5.	Сетка	Если установлена галочка, то на трассировке отображается сетка
2.6.	Описание	Данный параметр доступен только для **оси Y**. Название оси трассировки
2.7.	Фиксированные расстояния	Если установлена галочка, то появляется возможность настроить деления шкалы трассировки
2.8.	Расстояние	Данный параметр доступен только, если установлена галочка 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
2.11.	Фон	Тип фона элемента (**Не отображается/Выбрать фон/Использовать фон из стиля визуализации**)
2.12.	Цвет фона	Цвет фона элемента. Данный параметр доступен только в том случае, если в пп. 2.11 выбран режим **Выбрать фон**
2.13.	Цвет оси	Цвет осей элемента
2.14.	Сброс	Сброс настроек отображения трассировки к значениям по умолчанию
2.15.	Использовать по умолчанию	Текущие настройки отображения трассировки становятся настройками по умолчанию, и нажатие кнопки **Сброс** присваивает их любой трассировке проекта
2.16.	Добавить ось Y	Команда добавляет на трассировку дополнительную ось Y. Каждая ось должна иметь уникальное описание (пп. 2.6)
2.17.	Параметры переменной	Данная вкладка доступна только для элемента Тренд, который имеет аналогичные с элементом **Трассировка** параметра отображения. На данной вкладке можно для каждой переменной, отображаемой на тренде, указать переменную типа **BOOL**, которая будет определять ее видимость (**TRUE** – переменная отображается, **FALSE** – переменная не отображается)

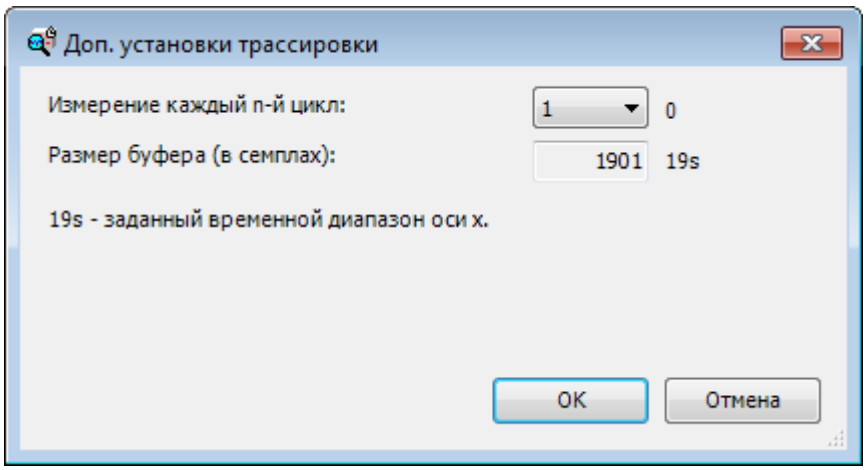


Рисунок 7.65 – Настройки трассировки – Дополнительно

Продолжение Табл. 7.24

№	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)

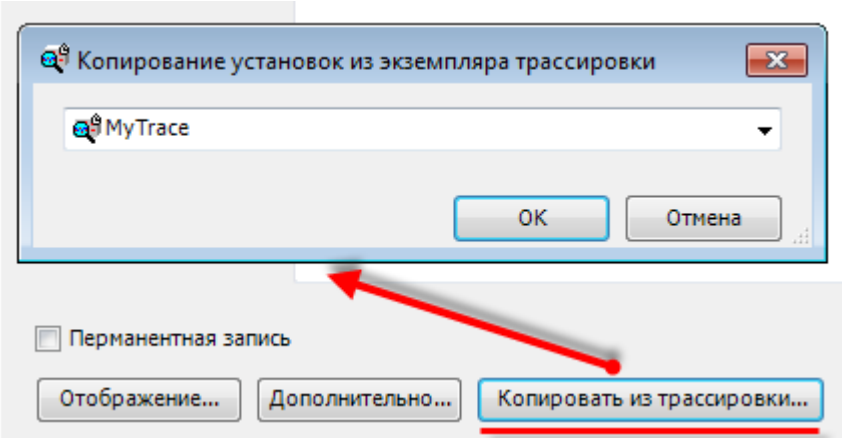


Рисунок 7.66 – Конфигурация трассировки – Копировать из трассировки

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

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

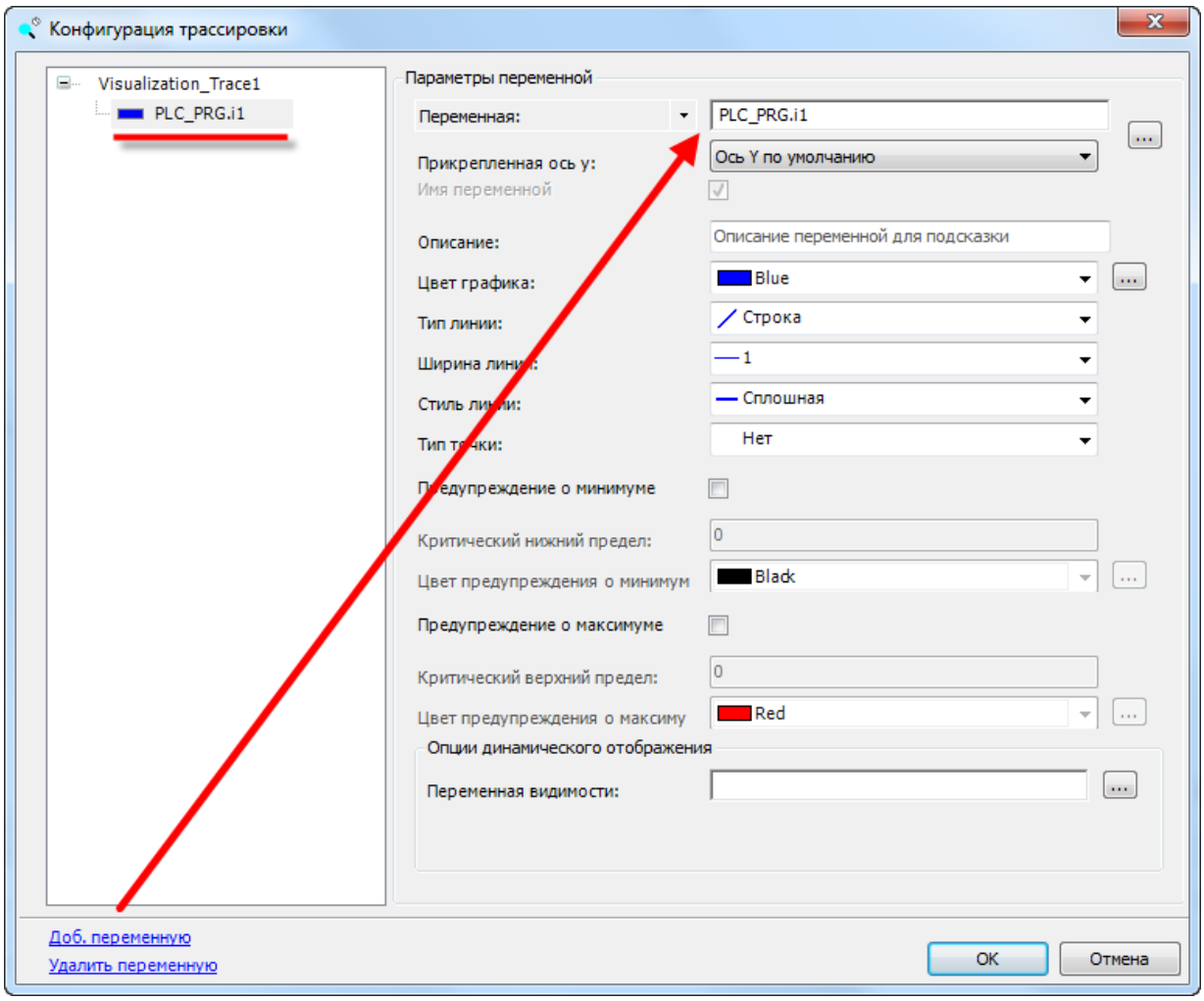


Рисунок 7.67 – Конфигурация трассировки – Установки переменной

Продолжение Табл. 7.24

№	Название параметра	Описание параметра
5.	**Установки переменной**	
5.1.	Переменная	Числовая переменная, отображаемая трассировкой
5.2.	Прикрепленная ось Y	Ось Y трассировки, к которой привязана переменная
5.3.	Имя переменной	В случае наличия галочки в подсказке и легенде элемента помимо имени переменной отображается ее описание (пп. 5.4)
5.4.	Описание	Описание переменной для подсказки легенды
5.5.	Тип кривой	Тип кривой переменной: **Строка** (линия) или **Область**
5.6.	Цвет графика	Цвет графика данной переменной
5.7.	Тип линии	Тип соединения точек на графике. Возможные значения: **Строка** – точки соединяются линиями **Шаг** – точки соединяются прямоугольными импульсами **Нет** – точки не соединяются
5.8.	Ширина линии	Ширина линии. Возможные значения: **1…255**
5.9.	Стиль линии	Стиль линии. Возможные значения: **Сплошная/Тире/Точки/Тире точка/Тире точка точка**
5.10.	Тип точки	Тип точек на графике. Возможные значения: **Точка/Крест/Нет**

Продолжение Табл. 7.24

№	Название параметра	Описание параметра
5.11.	Предупреждение о минимуме	Если установлена галочка, то появляется возможность окрашивать часть графика, выходящую за критический нижний предел (пп. 5.12), в заданный цвет (пп. 5.13)
5.12.	Критический нижний предел	Данный параметр доступен только, если установлена галочка в пп. 5.11. Если значение переменной меньше, чем указанное здесь, то цвет графика ниже этого значения окрашивается в заданный цвет
5.13.	Цвет мин. предупреждения	Данный параметр доступен только, если установлена галочка в пп. 5.11. Цвет, в который окрашивается часть графика, выходящая за критический нижний предел
5.14.	Предупреждение о максимуме	Если установлена галочка, то появляется возможность окрашивать часть графика, выходящую за критический верхний предел (пп. 5.15), в заданный цвет (пп. 5.16)
5.15.	Критический верхний предел	Данный параметр доступен только, если установлена галочка в пп. 5.14. Если значение переменной больше, чем указанное здесь, то цвет графика выше этого значения окрашивается в заданный цвет
5.16.	Цвет макс. предупреждения	Данный параметр доступен только, если установлена галочка в пп. 5.14. Цвет, в который окрашивается часть графика, выходящая за критический верхний предел
5.17.	Переменная видимости	Переменная типа **BOOL**. Когда она принимает значение **FALSE**, данная переменная перестает отображаться на трассировке (но продолжает записываться)

7.6.2 Тренд

Рисунок 7.68 – Пиктограмма графического примитива Тренд

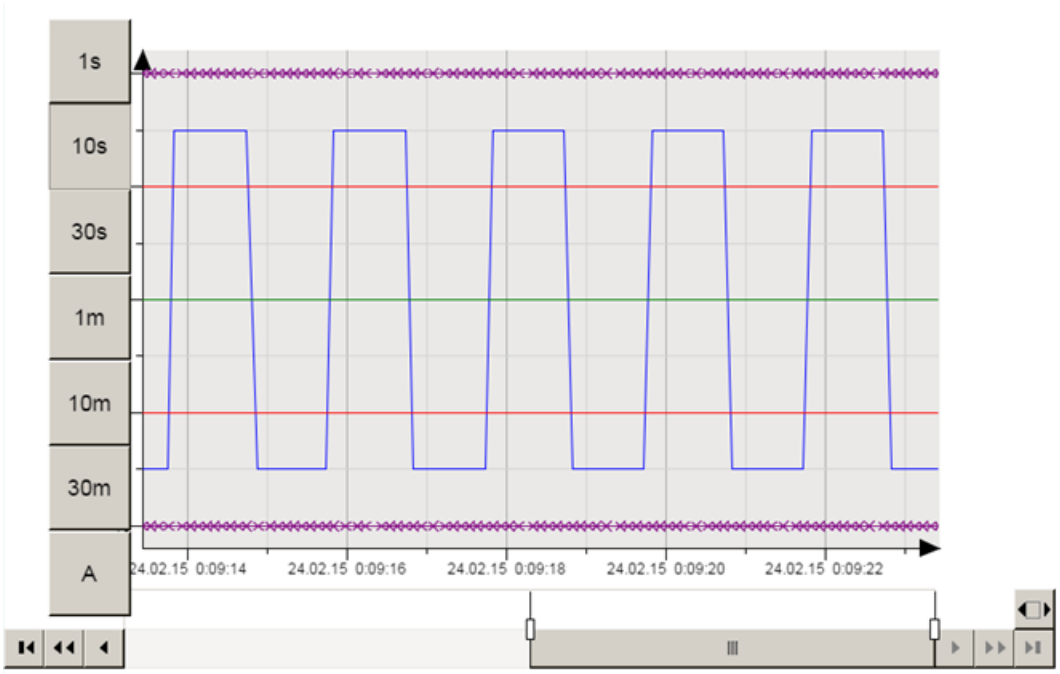


Рисунок 7.69 – Внешний вид элемента Тренд с вспомогательными элементами Селектор диапазона дат и Селектор времени

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

В случае добавления 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 или на подключенном к нему накопителе 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.

ПРИМЕЧАНИЕ

Без привязанного Селектора диапазона дат на тренде не будут отображаться значения переменных.

Описание типичных параметров графического примитива приведено в таблице 7.1.

Уникальные параметры примитива описаны в таблице 7.25.

Настройки компонента **Trend Recording Manager** приведены в таблице 7.26.

Пример работы с элементом приведен в п. 10.2.10.

Таблица 7.25 – 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. таблицу 7.24, пп. 2
3.	Показать курсор	Если установлена галочка, то во время просмотра истории тренда отображается курсор (метка со значением)
4.	Показать подсказку	Данный параметр доступен только, если установлена галочка в пп. 2. Если установлена галочка, то курсор тренда отображается информация о значениях переменных
5.	Показать рамку	Если установлена галочка, то вокруг элемента отображается контур
6.	Числовой формат	Спецификатор формата вывода переменных, отображаемых в подсказке элемента
7.	**Обозначения шкалы**	
7.1.	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)
7.2.	Обозначения в две строки	Если установлена галочка, то временная отметка отображается в две строки: на первой – дата, на второй – время
7.3.	Опустить незначимую информацию	Если установлена галочка, то крайняя левая временная отметка тренда всегда отображается в формате **День-Месяц-Год Часы-Минуты-Секунды**, а все остальные – в формате **Минуты-Секунды**. Пп. 7.4. в данном случае неактивен
7.4.	***Интернационализация***	
7.4.1.	Дата	Данный параметр доступен только при отсутствии галочки в пп. 7.3. Формат вывода даты (см. п. 8.2)
7.4.2.	Время	Данный параметр доступен только при отсутствии галочки в пп. 7.3. Формат вывода времени (см. п. 8.2)
8.	**Прикрепленные экземпляры элементов управления** В данном 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**	
8.1.	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. 7.7.1
8.2.	Селектор времени	Селектор времени данного тренда. Используется для масштабирования графиков тренда. Настройки элемента описаны в п. 7.7.2
8.3.	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. 7.6.3

9.	**Управляющие переменные** В данном 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 (галочка Zoom/Pan)**	
9.1.	***Масштабирование***	
9.1.1.	Включить	Переменная типа **BOOL**, которая используется для управления режимом масштабирования (**TRUE** – масштабирование включено, **FALSE** – масштабирование отключено). В режиме масштабирования при зажатой **ЛКМ** можно выделить на графике прямоугольную область. При отпускании мыши произойдет приближение к этой области. Для устройств с multitouch-экраном для масштабирования можно использовать жесты pinch и spread
9.1.2.	Домой	Переменная типа **BOOL**, по переднему фронту которой происходит возвращение к исходному масштабу
9.1.3.	Отменить	Переменная типа **BOOL**, по переднему фронту которой происходит переход к предыдущему отображаемому масштабу
9.1.4.	Масштабируется	Переменная типа **BOOL**, которая имеет значение **TRUE**, если масштаб графика отличается от исходного
9.2.	***Панорамирование***	
9.2.1.	Включить	Переменная типа **BOOL**, которая используется для управления режимом панорамирования (**TRUE** – панорамирование включено, **FALSE** – панорамирование отключено). В режиме панорамирования при зажатой **ЛКМ** можно перемещать координатные оси в пределах элемента. Это позволяет «пролистывать» график по вертикали и горизонтали. Для устройств с multitouch-экраном для панорамирования можно использовать жест pinch
9.2.2	Домой	Переменная типа **BOOL**, по переднему фронту которой происходит возвращение координатных осей к исходному положению
9.2.3.	Панорамируется	Переменная типа **BOOL**, которая имеет значение **TRUE**, если текущее положение координатных осей отличается от исходного
9.3.	Enable for Y axis	Переменная типа **BOOL**, которая определяет, разрешено ли масштабирование и панорамирование графика по оси Y (**TRUE** – разрешено)

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

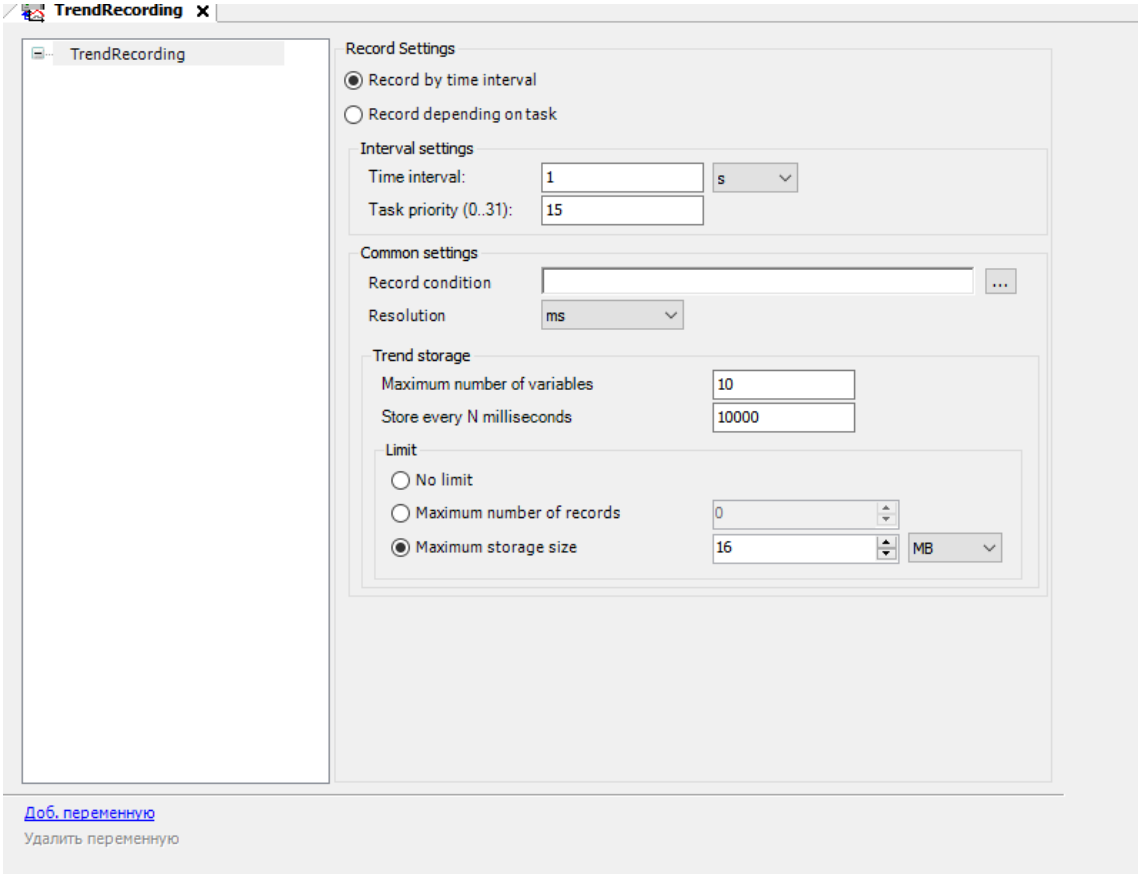


Рисунок 7.70 – Конфигурация тренда

Таблица 7.26 – Конфигурация элемента Тренд

№	Название параметра	Описание параметра
1.	**Record Settings**	
1.1.	Record by time interval	Вызов тренда будет производиться в рамках неявно созданной задачи
1.2.	Time interval	Период вызова неявно создаваемой задачи тренда (для режима **Record by time interval**)
1.3.	Task priority	Приоритет неявно создаваемой задачи тренда (для режима **Record by time interval**). Не рекомендуется устанавливать приоритет ниже **15**, так как в этом случае задача будет выполняться в реальном времени с возможностью вытеснения других задач
1.4.	Record depending on task	Вызов тренда будет производиться в рамках одной из задач конфигурации задач проекта
1.5.	Task	Задача, в которой будет производится вызов тренда (для режима **Record depending on task**)
1.6.	Measure in every n-th cycle	Частота обновления данных на тренде, выраженная в циклах задачи тренда. С этой частотой происходит отрисовка новых точек на тренде и копирование данных в буфер тренда. Этот буфер размещается в оперативной памяти. Возможные значения: **1/2/3/…/20/100/200/500/1000** (для режима **Record depending on task**)

1.7.	Additional runtime buffer for	Буфер для хранения значений, не записанных трендом из-за непредвиденных задержек. Используется для предотвращения потери данных (для режима **Record depending on task**)
2.	**Common Settings**	
2.1.	Record condition	Переменная типа **BOOL**. Пока она имеет значение **TRUE** – то тренд находится в работе и на график добавляются данные (а также происходит сохранение истории). **FALSE** – тренд останавливается. Если переменная не привязана – то тренд всегда находится в работе (поведение по умолчанию)
2.2.	Resolution	Минимальная единица отображения времени на трассировке. Возможные значения: **миллисекунды/микросекунды**
3.	**Trend Storage**	
3.1.	Maximum number of variables	Максимальное число переменных, отображаемых на тренде
3.2.	Store every N milliseconds	Периодичность записи данных из буфера тренда в файл
3.3.	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

Добавление переменных, значения которых будут отображаться на тренде, и их настройка аналогичны элементу Трассировка (см. рисунок 7.67 и таблицу 7.24, пп. 5).

Отличия:

- Параметр **Record Condition** – переменная типа **BOO**L, которая позволяет управлять записью конкретной переменной тренда (**TRUE** – переменная записывается, **FALSE** – переменная не записывается);
- Отсутствует параметр **Переменная видимости**. Вместо этого управление видимостью производится в меню **Отображение – Параметры переменной** (см. таблицу 7.24, пп. 2.17).

Пример очистки истории тренда из кода программы приведен. в п. 10.4.8.

7.6.3 Легенда

Рисунок 7.71 – Пиктограмма графического примитива Легенда

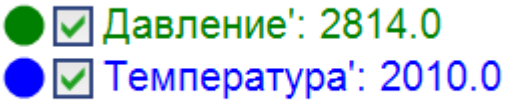


Рисунок 7.72 – Внешний вид элемента Легенда

Графический примитив **Легенда** используется совместно с элементом Тренд для описания соответствия переменных и цветов/типов графиков, которыми отображаются их значения. Также легенда отображает значения переменных в точке курсора и позволяет управлять видимостью перьев графика.

Легенда должна быть привязана к одному из трендов (см. таблицу 7.25. пп. 7.3).

Описание типичных параметров графического примитива приведено в таблице 7.1.

Уникальные параметры примитива описаны в таблице 7.27.

Таблица 7.27 – 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 рамку	Если установлена галочка, то у элемента отображается контур и фон
3.	Числовой формат	Спецификатор формата вывода переменных, отображаемых элементом
4.	**Размещение**	
4.1.	Макс. число строк	Число строк легенды
4.2.	Макс. число столбцов	Число столбцов легенды
4.3.	Curve display type	Тип пиктограммы в легенде. Возможные значения: **Line type** – линия **Bullet** – точка **None** – пиктограмма не отображается
4.4.	Name in curve color	В случае установки галочки цвет текста в легенде совпадает с цветом соответствующего пера тренда
4.5.	Visibility check boxes	В случае установи галочки в легенде отображаются чекбоксы, позволяющие управлять видимостью перьев

7.6.4 Элемент ActiveX

Рисунок 7.73 – Пиктограмма графического примитива ActiveX

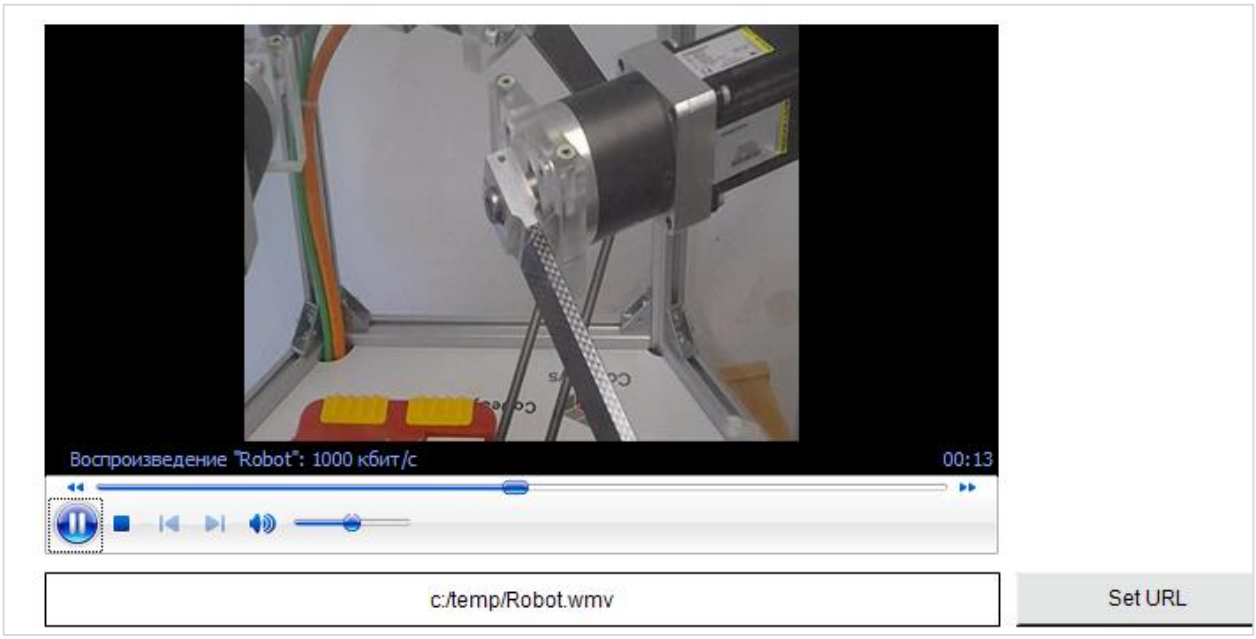


Рисунок 7.74 – Пример использования графического примитива ActiveX – проигрывание видеофайлов с помощью Windows Media Player на виртуальном контроллере CODESYS Control Win V3

Графический примитив **ActiveX** используется для интеграции внешних элементов **ActiveX/Native Control** в проект **CODESYS**.

ПРИМЕЧАНИЕ
Внешние элементы должны входить в состав прошивки целевого устройства.

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

Так как данный функционал в данный момент не поддерживается устройствами компании **ОВЕН**, а также является достаточно сложным в использовании, то его описание не приводится. В случае необходимости можно воспользоваться справкой **CODESYS** и примером использования элемента от компании CODESYS Group (разработчик **CODESYS**): ActiveXExample.package

Пример устанавливается с помощью **Менеджера пакетов** (вкладка **Инструменты**) или (начиная с версии **CODESYS V3.5 SP17**) с помощью утилиты **CODESYS Installer**.

7.6.5 WebBrowser

Рисунок 7.75 – Пиктограмма графического примитива WebBrowser

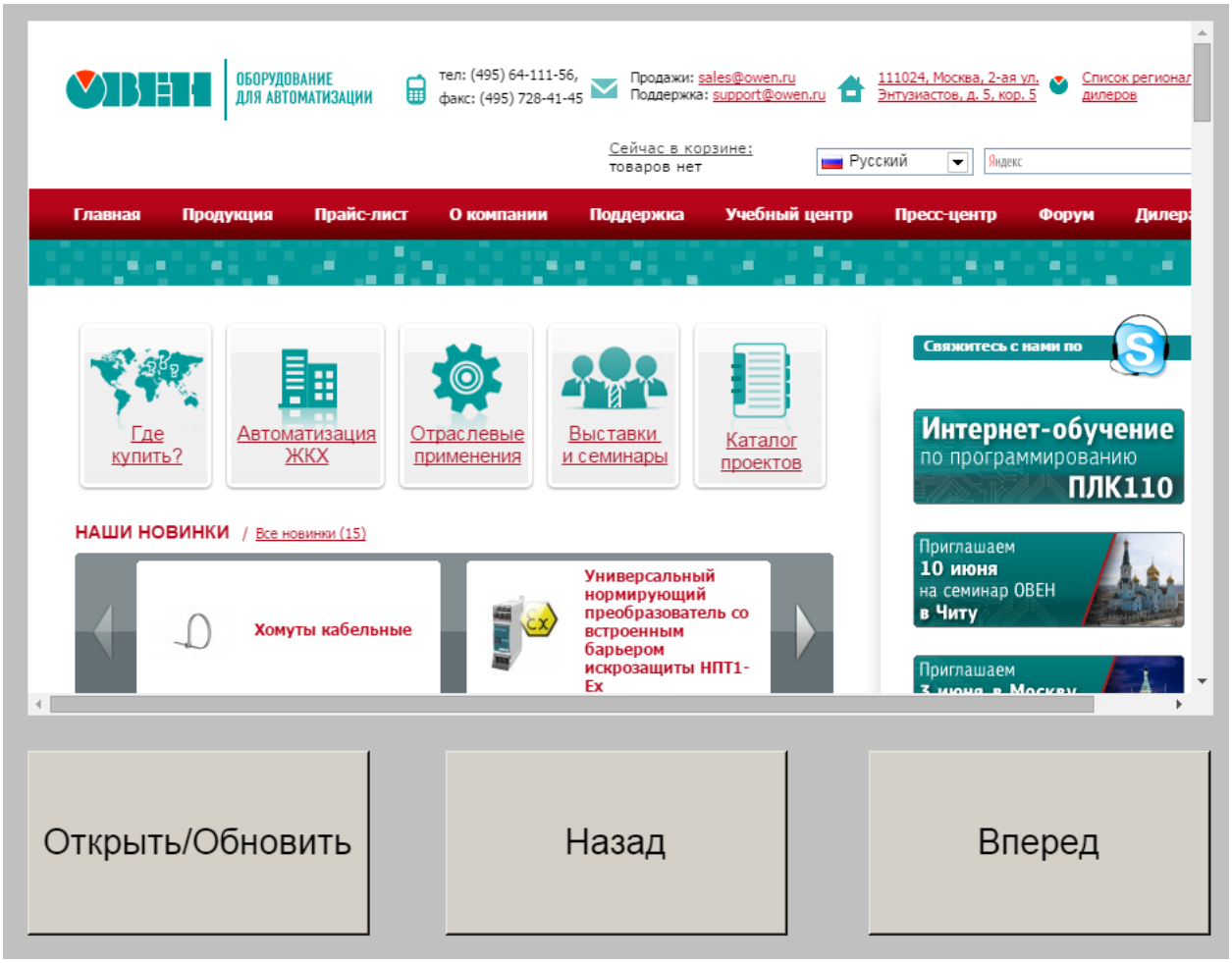


Рисунок 7.76 – Внешний вид элемента WebBrowser

Графический примитив **WebBrowser** используется для отображения веб-страниц.

ПРИМЕЧАНИЕ

Использование примитива возможно только на целевых устройствах с поддержкой **Java** и **HTML5**. Не гарантируется открытие в элементе любой веб-страницы (на стороне сайта требуется разрешение открытия страниц в **IFrame**). В контроллерах ОВЕН СПК не поддерживается работа элемента в таргет-визуализации.

Описание типичных параметров графического примитива приведено в таблице 7.1.

Уникальные параметры примитива описаны в таблице 7.28.

Пример работы с элементов приведен в п. 10.2.11.

Таблица 7.28 - 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.	Показать	Переменная типа **BOOL**, использующаяся для открытия веб-страниц в элементе. Если она принимает значение **TRUE**, то веб-страница, указанная в пп. 1, открывается элементом. Следует обратите внимание, что значение **FALSE** не приводит к прекращению отображения страницы
1.3.	Назад	Переменная типа **BOOL**, использующаяся для возвращения к просматриваемой ранее странице (по значению **TRUE**)
1.4.	Вперед	Переменная типа **BOOL**, использующаяся для возвращения к странице, с которой был сделан переход по кнопке **Back** (по значению **TRUE**)

7.6.6 Символ ожидания Кубик/Цветок

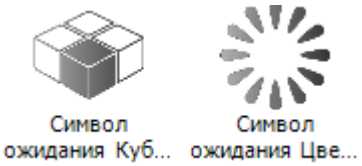


Рисунок 7.77 – Пиктограмма графического примитива Символ ожидания

Рисунок 7.78 – Внешний вид элемента Символ ожидания

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

Описание типичных параметров графического примитива приведено в таблице 7.1.

7.6.7 Текстовый редактор

Рисунок 7.79 – Пиктограмма графического примитива Текстовый редактор

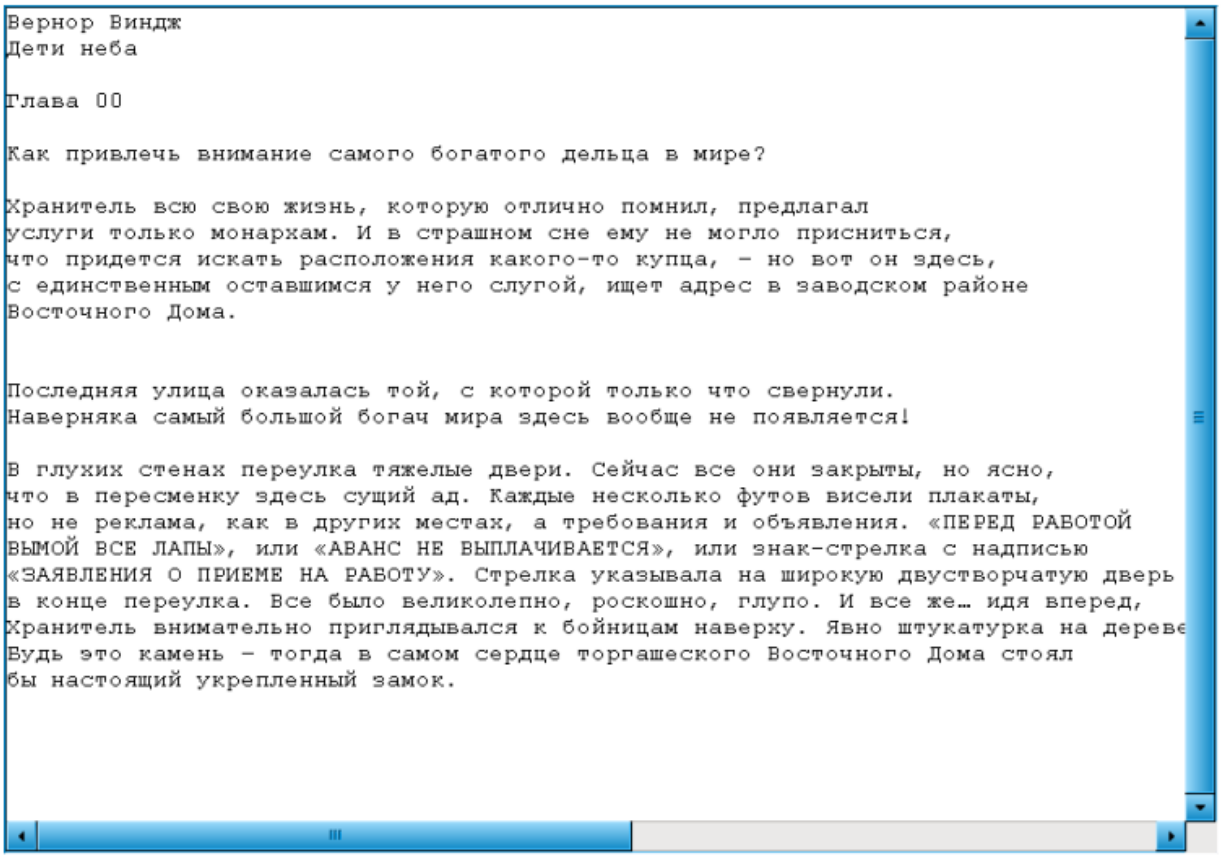


Рисунок 7.80 – Внешний вид элемента Текстовый редактор

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

Описание типичных параметров графического примитива приведено в таблице 7.1.

Уникальные параметры примитива описаны в таблице 7.29.

Пример работы с элементом приведен в п. 10.2.12.

Таблица 7.29 – 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.	Переменная	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 значение **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 переменная принимает значение **TRUE**, то курсор перемещается в положение, определяемое значениями переменных п. 1.3.1 – 1.3.2

Продолжение Табл. 7.29

№	Название параметра	Описание параметра
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 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 во время использования элемента. Коды ошибок приведены в списке **GVL_ErrorCodes** библиотеки **VisuElemTextEditor**
1.6.	Переменная для содержимого изменилась	Переменная типа **BOOL**, которая принимает значение **TRUE** в случае изменения 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 (и схожих с ней)

7.6.8 Путь 3D/ControlPanel

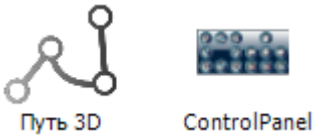


Рисунок 7.81 – Пиктограмма графического примитива Путь 3D

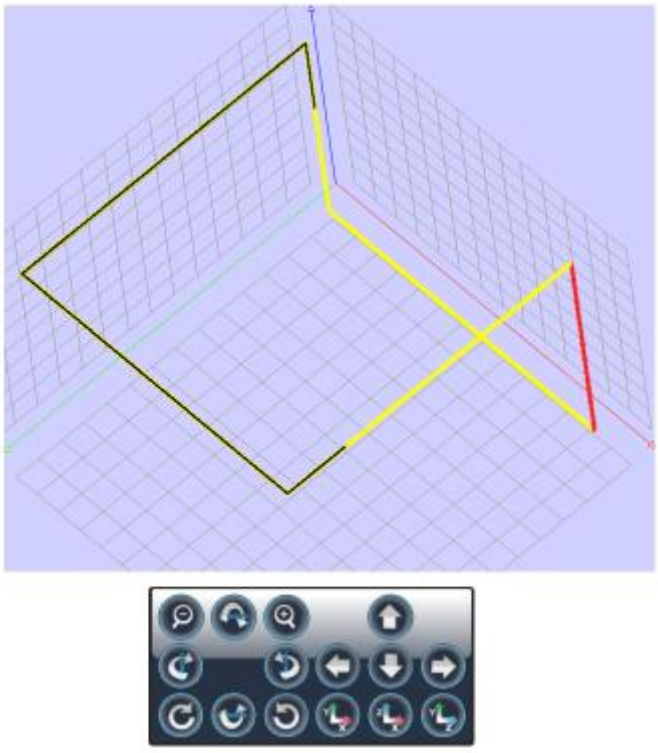


Рисунок 7.82 – Внешний вид элемента Путь 3D

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

Так как функционал в данный момент не поддерживается устройствами компании **ОВЕН**, а также является достаточно сложным в использовании, то его описание не приводится. В случае необходимости можно воспользоваться справкой **CODESYS** и примером использования элемента от компании CODESYS Group (разработчик **CODESYS**): 3DPathGenerator.package

Пример устанавливается с помощью **Менеджера пакетов** (вкладка **Инструменты**) или (начиная с версии **CODESYS V3.5 SP17**) с помощью утилиты **CODESYS Installer**.

7.7 Элементы управления датой/временем

7.7.1 Селектор диапазона дат

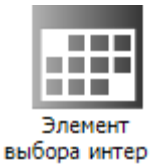


Рисунок 7.83 – Пиктограмма графического примитива Селектор диапазона дат

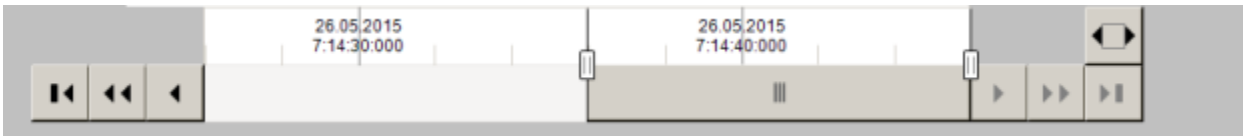


Рисунок 7.84 – Внешний вид элемента Селектор диапазона дат

Графический примитив **Селектор диапазона дат** используется для просмотра истории элемента Тренд. Селектор должен быть привязан к одному из трендов (см. таблицу 7.25, пп. 7.3).

Описание типичных параметров графического примитива приведено в таблице 7.1.

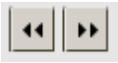
Уникальные параметры примитива описаны в таблице 7.30.

Функции кнопок элемента описаны в таблице 7.31.

Таблица 7.30 – Уникальные параметры графического примитива Селектор диапазона дат

№	Название параметра	Описание параметра
1.	Показать рамку	В случае наличия галочки вокруг элемента отображается рамка
2.	Разрешение	Минимальная единица отображения времени элемента. Возможные значения: миллисекунды/микросекунды
3.	Прикрепленный экземпляр элемента	Элемент Тренд, к которому прикреплен данный селектор. См. также таблицу 7.25, пп. 7.3
4.	**Обозначения шкалы**	
4.1.	Обозначения в две строки	Если установлена галочка, то временная отметка отображается в две строки: на первой – дата, на второй – время
4.2.	Опустить незначимую информацию	Если установлена галочка, то 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
4.3.	**Интернационализация**	
4.3.1.	Дата	Данный параметр доступен только, если не установлена галочка в пп. 2.2. Определяет формат вывода даты (см. п. 9.4)
4.3.2.	Время	Данный параметр доступен только, если не установлена галочка в пп. 2.2. Определяет формат вывода времени (см. п. 9.4)
5.	**Дополнительные кнопки**	
5.1.	Переход к максимальной временной отметке	Если установлена галочка, то у элемента присутствует кнопка **Переход к максимальной временной отметке** (см. таблицу 7.31, пп. 5)
5.2.	Переход к минимальной временной отметке	Если установлена галочка, то у элемента присутствует кнопка **Переход к минимальной временной отметке** (см. таблицу 7.31, пп. 4)
5.3.	Уменьшить масштаб	Если установлена галочка, то у элемента присутствует кнопка **Уменьшить масштаб** (см. таблицу 7.31, пп. 6)

Таблица 7.31 – 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 истории
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

7.7.2 Селектор времени

Рисунок 7.85 – Пиктограмма графического примитива Селектор времени

Рисунок 7.86 – 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. таблицу 7.25, пп. 7.3).

Описание типичных параметров графического примитива приведено в таблице 7.1.

Уникальные параметры примитива описаны в таблице 7.32.

Таблица 7.32 – Уникальные параметры графического примитива Селектор времени

№	Название параметра	Описание параметра
1.	Прикрепленный экземпляр элемента	Элемент **Тренд**, к которому прикреплен данный селектор. См. также таблицу 7.25, 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»	Если установлена галочка, то у элемента присутствует кнопка «**All**», временная область которой соответствует всей истории тренда
4.	**Управляющие переменные**	
4.1.	Время	Переменная типа **LINT**, принимающая значение отображаемой на тренде временной области (в микросекундах). С помощью переменной можно изменять диапазон временной области тренда из кода программы. В случае выбора всей истории (см. пп. 3.1), в переменную не записывается значение, но переменная, привязанная к параметру **Все выбранные** (пп. 4.2) принимает значение **TRUE**
4.2.	Все выбранные	Переменная типа **BOOL**, принимающая значение **TRUE** в случае выбора в качестве временной области всей истории тренда (см. пп. 3.1)

7.7.3 Аналоговые часы

Рисунок 7.87 – Пиктограмма графического примитива Аналоговые часы

Рисунок 7.88 – Внешний вид элемента Аналоговые часы

Графический примитив **Аналоговые часы** используется для отображения переменной типа **TOD** или системного времени в виде часов с аналоговым циферблатом.

Описание типичных параметров графического примитива приведено в таблице 7.1.

Уникальные параметры примитива описаны в таблице 7.33.

Таблица 7.33 – Уникальные параметры графического примитива Аналоговые часы

№	Название параметра	Описание параметра
1.	**Время**	
1.1.	Исп. системное время	В случае наличия галочки элемент отображает системное время. В случае отсутствия – значения переменной типа **TOD** из пп. 1.2.
1.2.	Переменная	Переменная типа **TOD**, значение которой отображается элементом.
2.	**Дизайн** Если параметр имеет значение **Из стиля**, то внешний вид элемента определяется стилем визуализации. Если параметр имеет значение **Явно**, то пользователь может самостоятельно настроить внешний вид элемента (цвет, шрифт и т. д.).	

7.7.4 Элемент выбора даты

Рисунок 7.89 – Пиктограмма графического примитива Элемент выбора даты

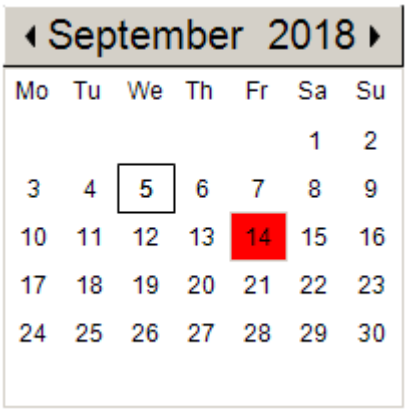


Рисунок 7.90 – Внешний вид элемента Элемент выбора даты

Графический примитив **Элемент выбора даты** используется для задания значения типа **DATE** с помощью календаря.

Описание типичных параметров графического примитива приведено в таблице 7.1.

Уникальные параметры примитива описаны в таблице 7.34.

Таблица 7.34 – Уникальные параметры графического примитива Элемент выбора даты

№	Название параметра	Описание параметра
1.	Переменная	Переменная типа **DATE**, в которую будет записано значение даты, выбранное с помощью элемента.
2.	**Дизайн** Если параметр имеет значение **Из стиля**, то внешний вид элемента определяется стилем визуализации. Если параметр имеет значение **Явно**, то пользователь может самостоятельно настроить внешний вид элемента (цвета, фон, шрифт и т. д.).	

ПРИМЕЧАНИЕ
Названия месяцев и дней недели хранятся в списке текстов **System** на панели POU и могут быть отредактированы пользователем.

7.7.5 Элемент выбора даты и времени

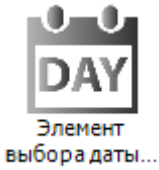


Рисунок 7.91 – Пиктограмма графического примитива Элемент выбора даты и времени

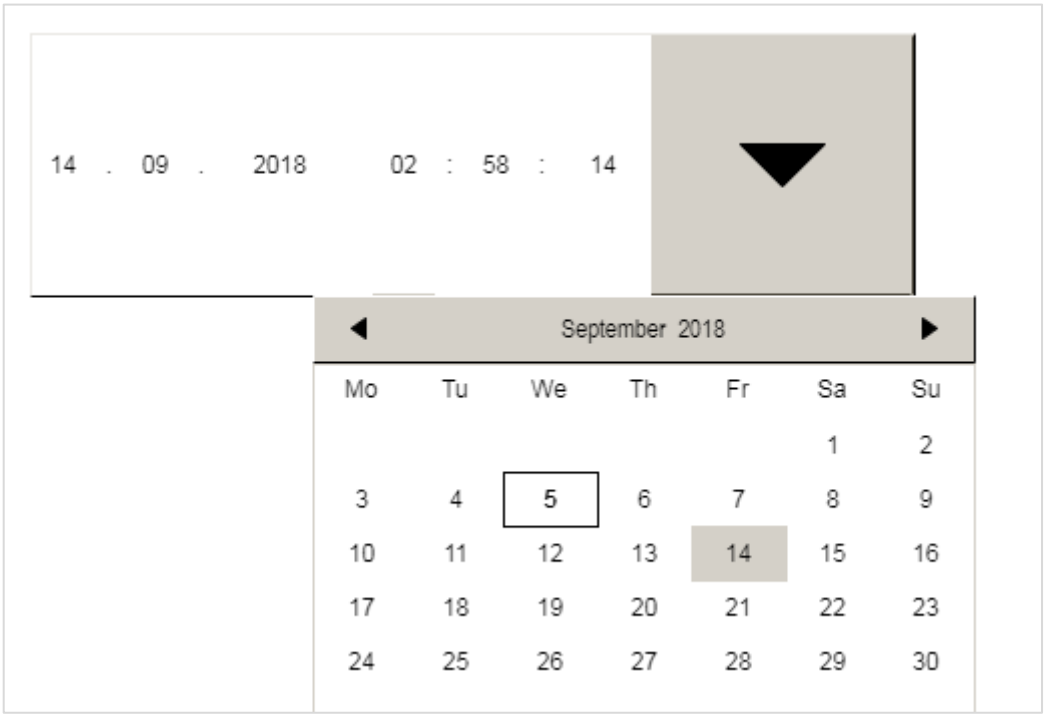


Рисунок 7.92 – Внешний вид элемента Элемент выбора даты и времени

Графический примитив **Элемент выбора даты** используется для задания значения типа **DT/DATE/TIME/LTIME/TOD.** Выбор даты производится с помощью календаря, открываемого нажатием на пиктограмму «Стрелка вниз», или с помощью диалога ввода, открываемого по нажатию на соответствующий разряд. Выбор времени производится с помощью диалога ввода.

Описание типичных параметров графического примитива приведено в таблице 7.1.

Уникальные параметры примитива описаны в таблице 7.35.

Таблица 7.35 – Уникальные параметры графического примитива Элемент выбора даты и времени

№	Название параметра	Описание параметра
1.	Переменная	Переменная типа **DT/DATE/TIME/LTIME/TOD**, в которую будет записано значение, выбранное с помощью элемента
2.	Строка формата	Спецификатор формата вывода переменной
3.	**Дизайн элемента выбора времени-даты** Если параметр имеет значение **Из стиля**, то внешний вид элемента определяется стилем визуализации. Если параметр имеет значение **Явно**, то пользователь может самостоятельно настроить внешний вид элемента (цвета, фон, шрифт и т. д.)	
4.	**Дизайн элемента выбора даты** Если параметр имеет значение **Из стиля**, то внешний вид элемента определяется стилем визуализации. Если параметр имеет значение **Явно**, то пользователь может самостоятельно настроить внешний вид элемента (цвета, фон, шрифт и т. д.)	
5.	**Элемент выбора даты расположения** Если параметр имеет значение **Динамически**, то размеры и положение календаря подбираются автоматически. Если параметр имеет значение **Вручную**, то пользователь может задать эти параметры самостоятельно	

ПРИМЕЧАНИЕ

Названия месяцев и дней недели хранятся в списке текстов **System** на панели POU и могут быть отредактированы пользователем.

7.8 Менеджер тревог

7.8.1 Компонент Конфигуратор тревог

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

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

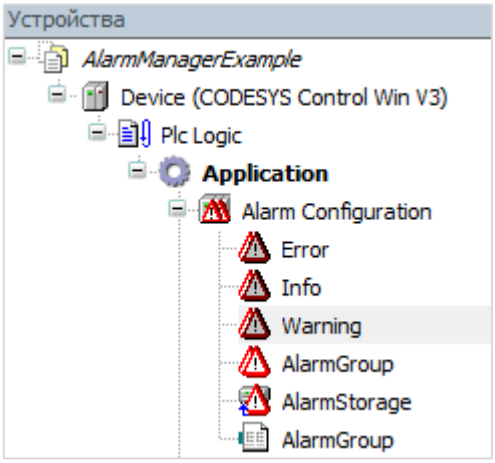


Рисунок 7.93 – Компонент Конфигуратор тревог и его дочерние компоненты

Компонент включает в себя четыре дочерних компонента:

1. **Классы тревог** предназначены для разделения тревог на отдельные типы. То есть 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, как 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**.

Архивы тревог сохраняются в памяти целевого устройства или на подключенном к нему накопителе в формате .sqlite.

Пример очистки истории тревог из кода программы приведен в п. 10.4.8.

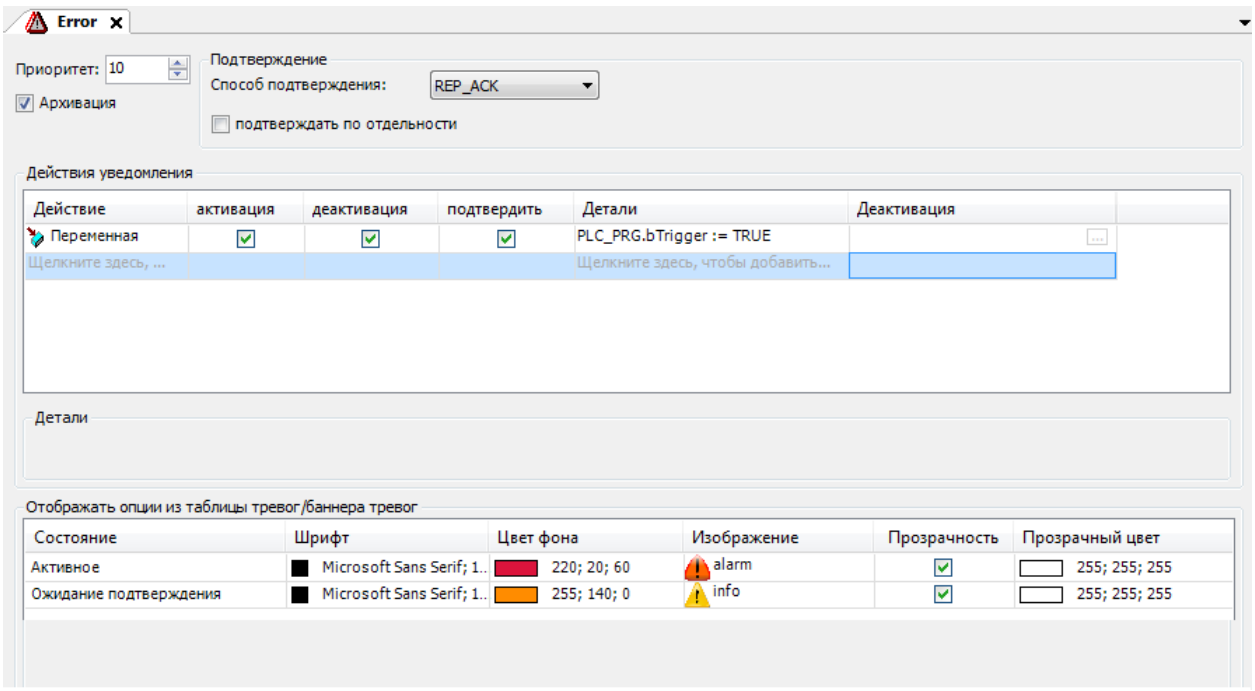


Рисунок 7.94 – Настройки класса тревог Error

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, но тревога квитирована пользователем.

Существуют следующие варианты смены состояний:

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

Таблица 7.36 – Настройки классов тревог

№	Название параметра	Описание параметра
1.	Приоритет	Приоритет тревог, привязанных к данному классу. Высший приоритет – **0**, низший – **255**
2.	Архивация	Если установлена галочка, то история тревог данного класса будет сохраняться в файл. См. также таблицу 7.37, пп. 2
3.	**Подтверждение**	
3.1.	Способ подтверждения	Условие возвращения тревоги данного класса в нормальное состояние. Возможные значения: **ACK** – квитирование тревоги; **REP** – деактивация тревоги; **ACK_REP** – подтверждение деактивированной или деактивация квитированной тревоги; **REP_ACK** – подтверждение деактивированной тревоги; **ACK_REP_ACK** – подтверждение деактивированной тревоги, причем перед деактивацией тревога опционально могла быть сквитирована
3.2.	Подтверждать по отдельности	Если установлена галочка, то тревоги данного класса можно будет только по отдельности (при отсутствии – можно все сразу). См. описание команд подтверждения в пп. 5 в табл. 7.40

Продолжение Табл. 7.36

№	Название параметра	Описание параметра
4.	**Действие уведомления**	
4.1.	Действие	Действие, выполняемое при активации/ деактивации/подтверждении (см. пп. 4.2–4.4) тревог данного класса. Возможные значения: **Переменная** – изменение значения переменной **Выполнение** – запуск файла **Вызов** – вызов функционального блока. Блок должен поддерживать интерфейс **IAlarmNotifiable** библиотеки **AlarmManager**. См. видеопример
4.2.	Активация	Если установлена галочка, то действие выполняется при активации тревог данного класса
4.3.	Деактивация	Если установлена галочка, то действие выполняется при деактивации тревог данного класса
4.4.	Подтвердить	Если установлена галочка, то 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**	
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	Если установлена галочка, то 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

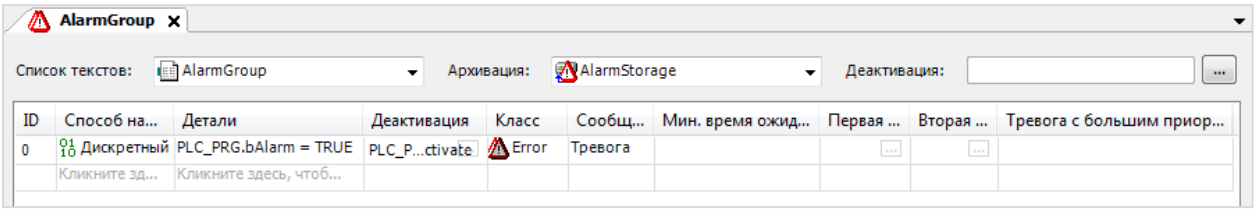


Рисунок 7.95 – Настройки группы тревог AlarmGroup

Таблица 7.37 – 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 Деактивация возможна только для тревог, находящемся в состоянии «нормальное» (условие возникновения тревоги не выполняется)
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 таблице 7.38
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** в случае активации тревоги. Этот текст автоматически добавляется в **Список текстов тревог**, заданный для группы. Помимо фиксированного текста, можно использовать заполнители. Список заполнителей приведен в таблице 7.39
4.7.	Мин. время ожидания	Время задержки появления тревоги при случае выполнения условия активации
4.8.	Первая триггерная переменная	Переменная, которые используются в случае необходимости записи дополнительной информации при активации тревоги (например, ее ID). Переменная должна принадлежать целочисленному типу или типу **STRING/WSTRING**
4.9.	<…> триггерная переменная	
4.10.	Тревога с большим приоритетом	Здесь можно указать существующую **тревогу с более высоким приоритетом** по отношению к создаваемой. В случае одновременного наступления двух тревог будет автоматически подтверждаться текущая. Это позволяет выполнять двухэтапную обработку тревоги, так как тревога с меньшим приоритетом перекрывается тревогой с большим приоритетом. **Пример** для системы контроля температуры: задается тревога с меньшим приоритетом (например, 10) для ***предупреждения*** о температуре, превышающей 30 °C. Предварительно уже задана другая тревога с более высоким приоритетом (например, 1), которая срабатывает при достижении температуры 50 °C (***критическое состояние***). Эту критическую тревогу можно ввести здесь, в конфигурации ***предупреждающей*** тревоги как «Тревогу с большим приоритетом». Существующая ***предупреждающая*** тревога будет автоматически подтверждаться в случае срабатывания ***критической*** тревоги

Таблица 7.38 – Способы наблюдения тревог (см. таблицу 7.37, пп. 4.2)

Способ наблюдения тревоги	Возможные настройки
Дискретный	**Выражение**: В левой части вводится наблюдаемое выражение, в правой части – выражение для сравнения (предел), посередине выберается 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**: вводится наблюдаемое выражение. **Область**: тревожная ситуация возникнет, как только наблюдаемое выражение примет значение из заданного диапазона. Слева вводится выражение, определяющее нижнюю границу области, справа – верхнюю границу. Наблюдаемое выражение будет отображено в поле посередине. Выбираются нужные операторы сравнения, а по желанию задается **Отклонение в %**[1]
Внешний диапазон	**Выражение**: вводится наблюдаемое выражение. **Область**: тревожная ситуация возникнет, как только наблюдаемое выражение примет значение вне заданного диапазона. Слева вводится выражение, определяющее нижнюю границу области, справа – верхнюю границу. Наблюдаемое выражение будет отображено в поле посередине. Выбираются нужные операторы сравнения, а также по желанию задается **Отклонение в %**[1]
Изменение	**Выражение**: вводится наблюдаемое выражение. Тревожная ситуация возникнет, как только его значение изменится
Событие	В данном случае состояние тревоги переключается через приложение с помощью функций библиотеки **AlarmManager.library.** См. справку CODESYS и видеопример

[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.

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.

ПРИМЕЧАНИЕ

Настройки группы тревог могут быть экспортированы/импортированы в формате **.csv**. Для этого следует нажать **ПКМ** на таблицу группы тревог и выбрать соответствующую команду в контекстном меню.

Таблица 7.39 – Заполнители сообщений тревог (см. таблицу 7.37, 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. таблицу 7.36, пп. 1)
TRIGGERVALUE[1]	Значение, вызвавшее тревогу
ALARMID	**ID** тревоги (см. таблицу 7.37, пп. 4.1)
CLASS	Имя класса тревог (см. таблицу 7.37, пп. 4.5)
CURRENTVALUE[1]	Текущее значение наблюдаемой переменной
LATCH1, LATCH2[1]	Значение первой и второй триггерной переменной (см. таблицу 7.37, 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**.

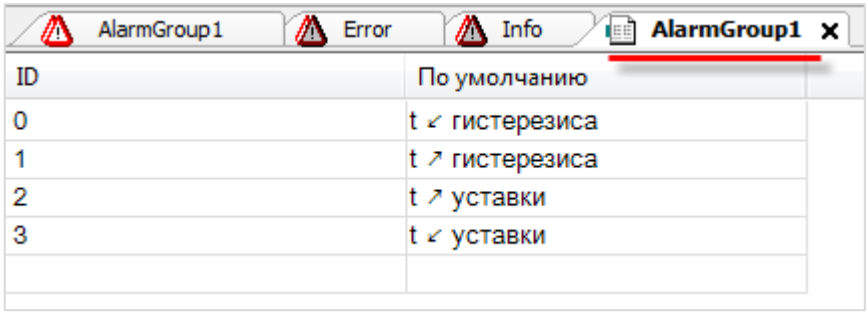


Рисунок 7.96 – Внешний вид Списка текстов тревог

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, так как 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**).

Контроллеры ОВЕН позволяют размещать файлы тревог на подключенном накопителе.

Хранилище тревог имеет следующие настройки:

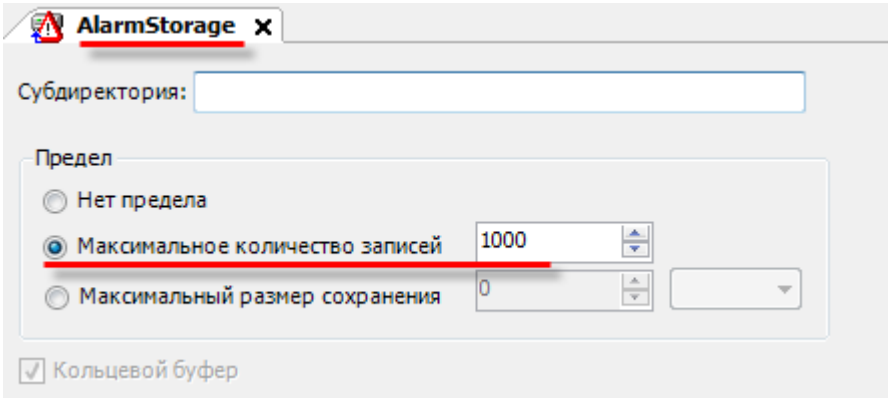


Рисунок 7.97 – Внешний вид Хранилища тревог

1. **Субдиректория** – это директория в памяти целевого устройства, в которой буду сохраняться файлы истории тревог.
2. **Предел** позволяет ограничить количество записей в базе данных – по максимальному количеству, либо по размеру файла записи. В случае превышения ограничения старые записи начнут удаляться (**кольцевой буфер**).

В случае выполнения **заводского сброса** (**Онлайн – Сброс заводской**) файл истории тревог удаляется.

ПРИМЕЧАНИЕ

Данные **AlarmStorage** могут быть экспортированы в формате **.csv** при подключении к контроллеру. Для экспорта следует нажать **ПКМ** на **AlarmStorage** и выбрать команду **CSV-экспорт тревог**.

7.8.2 Таблица тревог

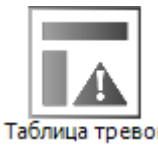


Рисунок 7.98 – Пиктограмма графического примитива Таблица тревог

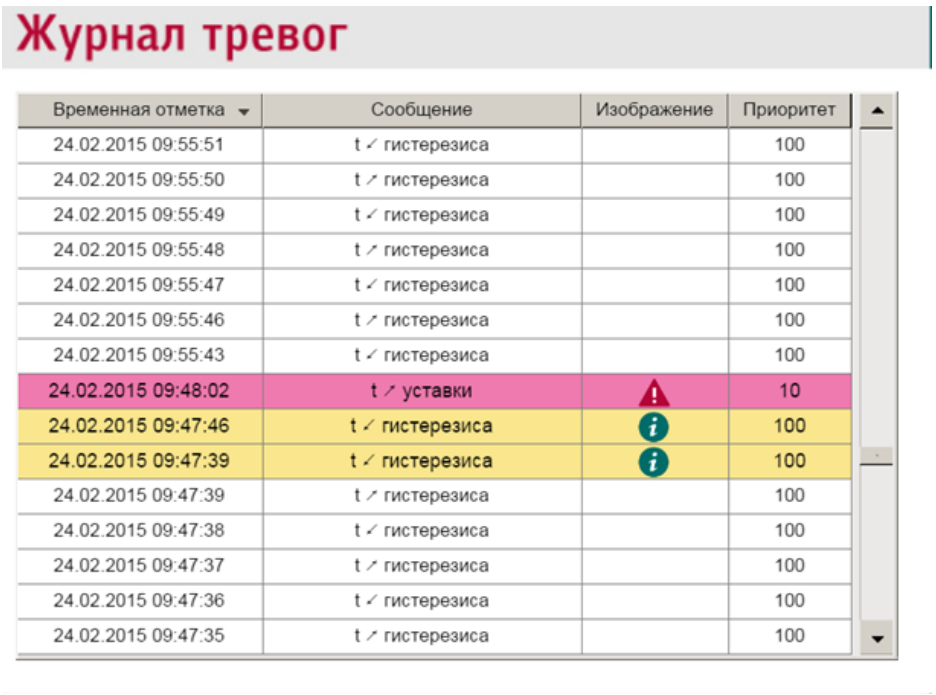


Рисунок 7.199 – Внешний вид элемента Таблица тревог (режим просмотра истории)

Графический примитив **Таблица тревог** используется для отображения состояний аварийных сигналов (тревог) с возможностью просмотра истории их срабатываний.

ПРИМЕЧАНИЕ
В обычном режиме в таблице отображаются только активные тревоги. У пользователя имеется возможность переключить 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. 7.8.1.

Описание типичных параметров графического примитива приведено в таблице 7.1.

Уникальные параметры примитива описаны в таблице 7.40.

Пример работы с элементом приведен в п. 10.2.13.

Пример очистки истории тревог из кода программы приведен в п. 10.4.8.

Таблица 7.40 – Уникальные параметры графического примитива Таблица тревог

№	Название параметра	Описание параметра
1.	**Конфигурация тревоги**	
1.1.	Группы тревог	Группы тревог (см. таблицу 7.36), 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. таблицу 7.37), тревоги которых будут отображаться данным элементом
1.5.	***Фильтр по latch 1***	
1.5.1.	Переменная фильтра	Переменная типа **STRING**, содержащая значение, по которому будет выполнена фильтрация тревог по столбцу первой триггерной переменной (см. таблицу 7.37)
1.5.2.	Тип фильтра	Переменная типа **INT** (или **AlarmManager.EFilterLatchContent**), определяющая тип фильтра. Возможные значения: **0** – фильтрация отключена; **1** – фильтрация по строковому значение (‘alarm_text’); **2** – фильтрация по значению в МЭК-формате (‘T#10s’); **3** – фильтрация по числовому значение (‘123’, ’11.22’)
1.6.	***Filter by time range***	
1.6.1.	Filter variable, from	Переменная типа **DT**, определяющая начальную метку времени для выборки по диапазону времени
1.6.2.	Filter variable, to	Переменная типа **DT**, определяющая конечную метку времени для выборки по диапазону времени
1.6.3.	Filter type	Переменная типа **INT** (или **AlarmManager. EFilterTimeRangeType**), определяющая тип фильтра. Возможные значения: **0** – фильтрация отключена; **1** – фильтрация включена
2.	**Общая конфигурация элемента**	
2.1.	Показать заголовок строки	Если установлена галочка, то у элемента присутствует столбец с номерами строк
2.2.	Показать заголовок столбца	Если установлена галочка, то 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	Если установлена галочка, то сообщение тревоги, которое не помещается в одну строку ячейки, переносится на следующую строку
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	Если установлена галочка, то заголовок столбца выравнивается в соответствии со значением, указанным в пп. 3.5
3.3.	***Ширина***	***Ширина столбца в пикселях***

3.4.	Тип данных	Тип данных, отображаемых в столбце. Возможные значения: **Изображение** – пиктограмма тревоги (см. таблицу 7.33, пп. 5.4); **Временная отметка** – дата и время последнего изменения состояния тревоги; **Временная отметка неактивна** – дата и время последней активации тревоги; **Временная отметка активна** – дата и время последней деактивации тревоги; **Подтверждение временной отметки** – дата и время последнего подтверждения тревоги; Формат временных отметок определяется глобальными строковыми константами **AlarmManager.AlarmGlobals.g_sDateFormat** и **AlarmManager.AlarmGlobals.g_sTimeFormat** **Значение** – значение переменной тревоги (см. таблицу 7.34 пп. 4.3); **Сообщение** – сообщение тревоги (см. таблицу 7.34 пп. 4.6); **Приоритет** – приоритет тревоги (см. таблицу 7.33 пп. 1); **Класс** – класс тревоги (см. таблицу 7.34 пп. 4.5); **Состояние** – состояние тревоги (см. рисунок 7.96)

Продолжение Табл. 7.40

№	Название параметра	Описание параметра
3.5.	Выравнивание текста	Тип выравнивания текста в столбце. Применяется в случае наличия галочки в пп. 3.2. Возможные значения: **Лево/Право/По центру**
3.6.	***Параметры цвета***	
3.6.1.	Включить параметры цвета	Переменная типа **BOOL**. **TRUE** – используются параметры пп. 3.6.2–3.6.3, **FALSE** – используются настройки по умолчанию
3.6.2	Цвет заливки ячейки	
3.6.2.1.	Переменная цвета	Переменная типа **DWORD**, определяющая цвет заливки по модели ARGB
3.6.2.	Использовать цвет для заголовка столбца	Если установлена галочка, то п. 3.6.2.1 влияет на цвет заливки заголовка столбца
3.6.3.	***Цвет текста***	
3.6.3.1	Переменная цвета	Переменная типа **DWORD**, определяющая цвет текста по модели ARGB
3.6.3.2	Использовать цвет для заголовка столбца	Если установлена галочка, то п. 3.6.3.1 влияет на цвет текста заголовка столбца
4.	**Выбор**	
4.1.	Цвет выбора	Цвет выделения ячеек элемента
4.2.	Цвет шрифта выделения	Цвет шрифта выделенных ячеек элемента
4.3.	Рамка вокруг выбранных ячеек	Если установлена галочка, то вокруг выделенных ячеек элемента отрисовывается контур
4.4.	Переменная для выбранной группы тревог	Переменная типа **STRING**, содержащая имя группы тревоги выделенной ячейки
4.5.	Переменная для выбранного ID тревог	Переменная типа **STRING**, содержащая имя списка текстов и **ID** выделенной тревоги
4.6.	Переменная для выбранной строки	Целочисленная переменная, содержащая номер выделенной строки (**−1** – не выделено ни одной строки, **0** – выделена первая строка и т. д.)
4.7.	Переменная для выбора действ. строки	Переменная типа **BOOL**. **TRUE** – одна из строк таблицы выделена
4.8.	Variable for selected alarm information	Переменная для получения информации о тревоге, которая отображается в выделенной строке таблицы. К переменной нужно привязать выход **AlarmSelectionInfo** экземпляра функционального блока **VisuElemsAlarm.AlarmSelectionInfoDefault**. Экземпляр будет содержать информацию о 10 последних выделенных тревогах (см. локальные переменные ФБ)

Продолжение Табл. 7.40

№	Название параметра	Описание параметра
5.	**Управляющие переменные** Данная вкладка содержит переменные типа **BOOL**, с помощью изменения значений которых можно управлять таблицей тревог. Переменные пп. 5.1–5.4 можно сформировать автоматически, выбрав в контекстном меню элемента (открывается по нажатию **ПКМ**) команду **Вставить элементы для подтверждения тревог**. Вместе с переменными также будут созданы кнопки для управления ими	
5.1.	Подтвердить выбранное	Переменная типа **BOOL**, использующаяся для подтверждения выделенной тревоги (по значению **TRUE**)
5.2.	Подтвердить все видимое	Переменная типа **BOOL**, использующаяся для подтверждения всех тревог, отображаемых элементом (по значению **TRUE**)
5.3.	История	Переменная типа **BOOL**, определяющая режим работы элемента: **FALSE** – режим текущих значений (отображаются только активные тревоги) **TRUE** – режим истории (отображаются все когда-либо произошедшие тревоги)
5.4.	Фиксировать позицию прокрутки	Переменная типа **BOOL**, использующаяся для отмены автоматической прокрутки таблицы к первой строке в режиме истории при появлении новой тревоги (по значению **TRUE**)
5.5	Счетчик тревог	Целочисленная переменная, содержащая число тревог, отображаемых в таблице. Доступна только для чтения
5.6.	Счетчик видимых строк	Целочисленная переменная, содержащая число строк, которое может быть одновременно отображено в таблице (без учета прокрутки). Доступна только для чтения
5.7.	Индекс прокрутки	Целочисленная переменная, характеризующая текущую позицию полосы прокрутки. Пока в таблице не отображается полоса прокрутки, переменная имеет значение **0**. После появления полосы прокрутки: перемещение на строку вниз/вверх приводит к увеличению/уменьшению значения переменной на **1**. Меняя значение переменной из программы можно листать таблицу тревог без использования полосы прокрутки и ее кнопок.
5.8.	Текущий столбец сортировки	Переменная типа **VisuElemsAlarm.VisuEnumAlarmDataType,** определяющая столбец, по которому осуществляется сортировка
5.9.	Переменная сортировки	Переменная типа **BOOL**, определяющая режим сортировки: **FALSE** – по убыванию **TRUE** – по возрастанию

7.8.3 Баннер тревог

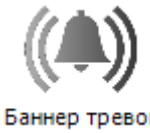


Рисунок 7.100 – Пиктограмма графического примитива Баннер тревог

Рисунок 7.101 – 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 параметров **Критерий фильтра**, **Подтвердить переменную** и вкладки **Обработка нескольких активных тревог**.

Критерий фильтра определяет, какая тревога будет отображаться элементом. Возможные значения: **Наиболее важный/Новейший**.

К параметру **Подтвердить переменную** привязывается переменная типа **BOOL**. Если переменная принимает значение **TRUE**, то происходит квитирование тревоги баннера.

Во вкладке **Обработка нескольких активных тревог** определяется режим обработки тревог, если несколько из них одновременно являются активными (в этом случае параметр **Активно несколько тревог** типа **BOOL** имеет значение **TRUE**). Если установлена галочка **Автоматическое переключение**, то переключение информации об активной тревоге происходит с периодом, определяемым параметром **Каждые N секунд** (тип **UINT**).

Если галочка не установлена, то переключение информации происходит с помощью параметров **Следующая тревога** и **Предыдущая тревога** (тип **BOOL**).

7.9 Остальные элементы

7.9.1 Декартовый график XY

Рисунок 7.102 – Пиктограмма графического примитива Декартовый график XY

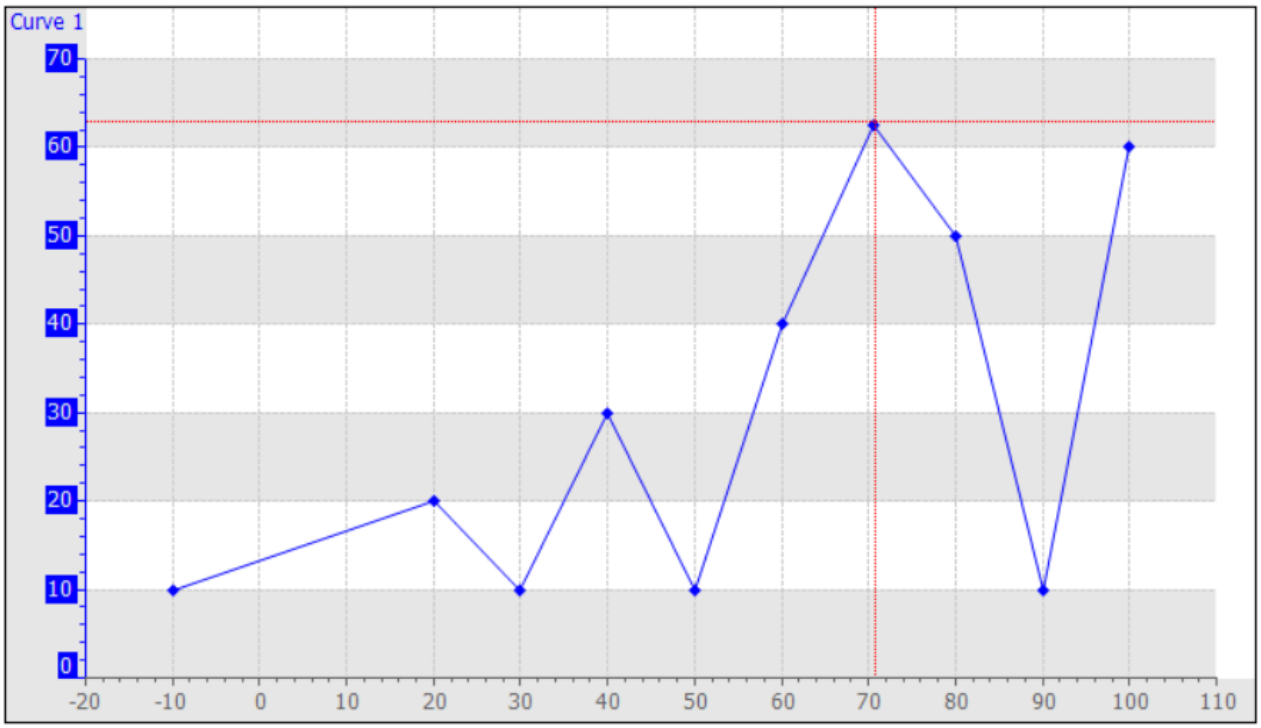


Рисунок 7.103 – Внешний вид элемента Декартовый график XY

Элемент **Декартовый график XY** (вкладка **Специальные элементы управления**) используется для отображения зависимости одной величины от другой. Элемент поддерживает отображение нескольких графиков в одном или разных масштабах.

Описание типичных параметров графического примитива приведено в таблице 7.1.

Уникальные параметры примитива описаны в таблицах 7.41 и 7.42.

Продвинутый пример от компании **CODESYS Group** по работе с библиотекой XY графика:

Visu Element XY Chart Example.package

Файлы формата **.package** устанавливаются с помощью **Менеджера пакетов** (вкладка **Инструменты**) или (начиная с версии **CODESYS V3.5 SP17**) с помощью утилиты **CODESYS Installer**.

Таблица 7.41 – Уникальные параметры графического примитива Декартовый график XY

№	Название параметра	Описание параметра
1.	Декартовый график XY	При нажатии на значение этого параметра открывается окно конфигурации элемента (см. таблицу 7.42)
2.	**Управляющие переменные**	
2.1.	***Масштабирование***	
2.1.1.	Включить	Переменная типа **BOOL**, которая используется для управления режимом масштабирования (**TRUE** – масштабирование включено, **FALSE** – масштабирование отключено). В режиме масштабирования при зажатой **ЛКМ** можно выделить на графике прямоугольную область. При отпускании мыши произойдет приближение к этой области. Для устройств с multitouch-экраном для масштабирования можно использовать жесты pinch и spread
2.1.2.	Домой	Переменная типа **BOOL**, по переднему фронту которой происходит возвращение к исходному масштабу
2.1.3	Отменить	Переменная типа **BOOL**, по переднему фронту которой происходит переход к предыдущему отображаемому масштабу
2.1.4.	Масштабируется	Переменная типа **BOOL**, которая имеет значение **TRUE**, если масштаб графика отличается от исходного
2.2.	***Панорамирование***	
2.2.1.	Включить	Переменная типа **BOOL**, которая используется для управления режимом панорамирования (**TRUE** – панорамирование включено, **FALSE** – панорамирование отключено). В режиме панорамирования при зажатой **ЛКМ** можно перемещать координатные оси в пределах элемента. Это позволяет «пролистывать» график по вертикали и горизонтали. Для устройств с multitouch-экраном для панорамирования можно использовать жест pinch
2.2.2	Домой	Переменная типа **BOOL**, по переднему фронту которой происходит возвращение координатных осей к исходному положению
2.2.3.	Панорамируется	Переменная типа **BOOL**, которая имеет значение **TRUE**, если текущее положение координатных осей отличается от исходного

ПРИМЕЧАНИЕ

При нажатии **ПКМ** на элемент можно использовать команду контекстного меню **Insert elements for controlling XYChart** для добавления элементов управления масштабированием и панорамированием, к которым будут привязаны переменные, автоматически объявленные в интерфейсе экрана после выполнения команды.

Для конфигурации элемента следует два раза нажать **ЛКМ** на значение параметра **Декартовый график XY** (см. пп. 1 в таблице 7.41). В результате откроется следующее окно:

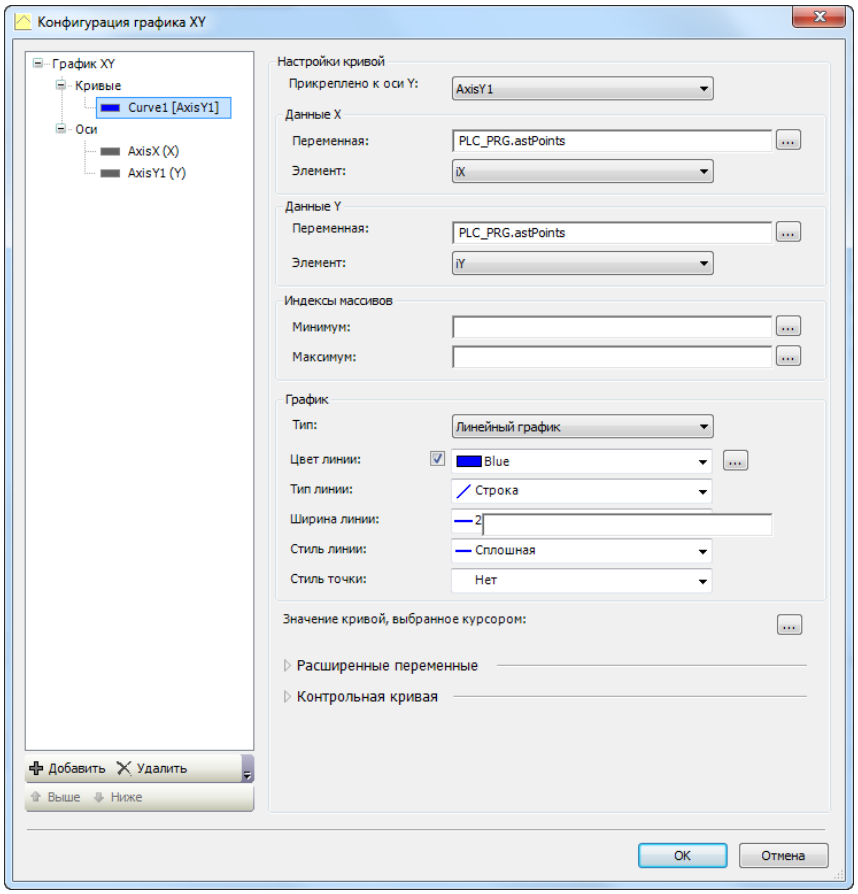


Рисунок 7.104 – Окно конфигурации элемента Декартовый график XY

Узел **График XY** содержит общие параметры отображения графика.

Узел **Кривые** содержит набор графиков, отображаемых элементом. Для добавления нового графика следует нажать кнопку **Добавить**, для удаления выделенного – нажать кнопку **Удалить**. Для перемещения графиков в дереве следует использовать кнопки **Выше** и **Ниже**.

Узел **Оси** содержит набор осей элемента. Элемент имеет одну ось X и произвольное количество осей Y. Для добавления новой оси следует нажать кнопку **Добавить**, для удаления выделенной – нажать кнопку **Удалить**. Для перемещения осей в дереве следует использовать кнопки **Выше** и **Ниже**.

Доступные параметры описаны в таблице 7.42.

Таблица 7.42 – Параметры конфигурации графического примитива Декартовый график XY

№	Название параметра	Описание параметра
1.	**График XY**	
1.1.	Обновить	Тип обновления (перерисовки) графика: автоматически (при изменении любого отображаемых значений) или с помощью целочисленной переменной. Во втором случае перерисовка графика происходит при каждом изменении переменной
1.2.	Фон	Режим отображения фона элемента. Возможные значения: **Без фона** – фон элемента отсутствует; **Простой цвет** – можно выбрать цвет фона графика и степень прозрачности; **Изображение** – в качестве фона используется выбранное изображение из пула изображений; **Из стиля** – фон элемента определяется используемым стилем визуализации; **Градиент** – можно выбрать цвет фона графика с настройками градиента
1.3.	***Главный курсор***	
1.3.1.	Тип	Тип отображения главного курсора. Возможные значения: **Не используется/Вертикально/Большой крестик/Малый крестик/Пересечение**
1.3.2	Стиль линии	Стиль линии главного курсора. Возможные значения: **Сплошная/Тире/Точки/Тире Точка/Тире Точка Точка**
1.3.3.	Цвет	Цвет линии главного курсора
1.3.4.	Переменная позиции X	Целочисленная переменная, в которую записывается текущее значение позиции главного курсора по оси X. Управлять курсором с помощью этой переменной нельзя
1.4.	***Расширенные переменные***	
1.4.1.	Переменная элемента	Переменная типа **VisuElemXYChart.VisuStructXYChart** (структура), которая позволяет менять настройки данного узла в процессе работы программы
1.4.2.	Автоматическое обновление включено	Переменная типа **BOOL**, которая позволяет управлять режимом обновления графика, если для параметра **Обновить** (см. п. 1.1) установлено значение **Автоматически**
1.5.	***Вспомогательный курсор***	
1.5.1.	Стиль линии	Стиль линии вспомогательного курсора. Возможные значения: **Сплошная/Тире/Точки/Тире Точка/Тире Точка Точка**
1.5.2.	Цвет	Цвет линии вспомогательного курсора
1.6.	***Сетка***	
1.6.1.	Тип	Тип сетки элемента.
1.6.2.	Стиль линии	Стиль линий сетки Возможные значения: **Сплошная/Тире/Точки/Тире Точка/Тире Точка Точка**
1.6.3.	Цвет	Цвет линий сетки
1.6.4.	Сетка позади графиков	В случае наличия галочки линии сетки не перекрывают график, в случая отсутствия – перекрывают
1.7.	***Нулевая линия***	
1.7.1.	Позиция	Ось, для которой отображается нулевая линия. Возможные значения: **Не используется**/**Отображение для оси X** (нулевая линия параллельна оси Y)/**Отображение для оси Y** (нулевая линия параллельна оси X)/**Отображение для обеих осей**
1.7.2.	Стиль линии	Стиль нулевой линии. Возможные значения: **Сплошная/Тире/Точки/Тире Точка/Тире Точка Точка**
1.7.3.	Ширина линии	Ширина линии в пикселях (1…20)
1.7.4.	Цвет	Цвет нулевой линии

2.	**Кривые**	
2.1.	***Настройки кривой***	
2.1.1.	Прикреплено к оси	Выбор оси, в масштабе которой отображается данная кривая
2.2.	***Данные X/Y***	
2.2.1.	Переменная	Переменные, содержащие X/Y координаты точек, отображаемых кривой. Возможные типы переменных: одно- или многомерный массив численного типа; одномерный массив структуры, включающей переменные численного типа
2.2.2.	Элемент	Номер измерения массива или переменная структуры для переменной из пп. 2.2.1
2.3.	***Индексы массивов***	
2.3.1.	Минимум	Целочисленная переменная или константа, определяющая минимальный индекс массива, отображаемого кривой
2.3.2.	Максимум	Целочисленная переменная или константа, определяющая максимальный индекс массива, отображаемого кривой
2.4.	***График***	
2.4.0.	Тип	Тип отображения кривой. Возможные значения: **Линейный график** – кривая отображается в виде набора точек или набора точек, соединенных линиями; **Столбчатая гистограмма** – кривая отображается в виде набора столбцов; **Комбинированная гистограмма** - кривая отображается в виде набора точек или набора точек, соединенных линиями, область под которыми заливается выбранным цветом
2.4.1.	***Линейный график***	
2.4.1.1.	Цвет линии	Если установлена галочка, то можно выбрать цвет кривой. Если галочка не установлена, то в качестве цвета кривой используется цвет оси
2.4.1.2.	Тип линии	Тип линии кривой. Возможные значения: **Строка** (линия)/**Низкий шаг** (ступенчатый график)/**Высокий шаг** (ступенчатый график)/**Нет** (только точки)
2.4.1.3.	Ширина линии	Ширина кривой в пикселях (1…20)
2.4.1.4.	Стиль линии	Стиль линии кривой. Возможные значения: **Сплошная/Тире/Точки/Тире Точка/Тире Точка Точка**
2.4.1.5.	Стиль точки	Стиль точек кривой. Возможные значения: **Нет/Точка/Крестик/Малый круг/Большой круг**
2.4.2.	***Столбчатая гистограмма***	
2.4.2.1.	Форма полосы	Тип ширины столбца гистограммы. **Фиксированная ширина** – ширина столбца фиксирована и составляет несколько пикселей; **Переменная ширина** – ширина столбца равна разнице между соседними точками кривой
2.4.2.2.	Основной цвет	Цвет нечетных столбцов гистограммы
2.4.2.3.	Второй цвет	Цвет четных столбцов гистограммы
2.4.2.4.	Цвет рамки	Цвет контура столбцов гистограммы
2.4.3.	***Комбинированная гистограмма***	
2.4.3.1.	Цвет линии	Если установлена галочка, то можно выбрать цвет кривой. Если галочка не установлена, то в качестве цвета кривой используется цвет оси
2.4.3.2.	Тип заливки	Тип заливки области под кривой. Возможные значения: **Нет заливки/Простой цвет/Градиент**
2.4.3.3.	Цвет заливки	Цвет заливки области под кривой
2.4.3.4.	Прозрачность	Степень прозрачности цвета области под кривой
2.4.3.5.	Ширина линии	Ширина кривой в пикселях (1…20)

2.5.	Значение кривой, выбранное курсором	Целочисленная переменная, в которую записывается текущее значение позиции главного курсора для данной кривой по оси Y. Управлять курсором с помощью этой переменной нельзя
2.6.	***Расширенные переменные***	
2.6.1.	Переменная кривой	Переменная типа **VisuElemXYChart.VisuStructXYChartCurve** (структура), которая позволяет менять настройки данной кривой в процессе работы программы
2.7.	***Контрольная кривая***	
2.7.1.	Кривая	Контрольная кривая для данной кривой. Область между данной кривой и контрольной кривой выделяется выбранным цветом
2.7.2.	Цвет	Цвет заливки области между данной кривой и контрольной кривой
3.	**Оси**	
3.1.	***Режим отображения***	
3.1.1.	Позиция	Позиция отображения оси. Возможные значения: **Ось X/Слева/Справа**
3.1.2.	Минимум	Целочисленная константа или переменная, которая определяет минимальное значение диапазона данной оси
3.1.3.	Максимум	Целочисленная константа или переменная, которая определяет максимальное значение диапазона данной оси
3.1.4.	Автомасштаб	В случае наличия галочки минимальное и максимальное значение рассчитывается автоматически в зависимости от отображаемых данных. В этом случае параметры 3.1.2 и 3.1.3 недоступны
3.1.5.	Цвет оси	Цвет оси
3.2.	***Фон***	
3.2.1.	Тип	Тип фона шкалы оси. Возможные значения: **Нет заливки/Простой цвет/Градиент/Изображение** (из пула изображений)
3.3.	***Расширенные переменные***	
3.3.1.	Переменная оси	Переменная типа **VisuElemXYChart.VisuStructXYChartAxis** (структура), которая позволяет менять настройки данной оси в процессе работы программы
3.4.	***Дополнительные параметры отображения***	
3.4.1.	Линейка отображения	В случае наличия галочки на оси отображается шкала с делениями
3.4.2.	Дополнительное место для оси X	В случае наличия галочки шкала делений оси X сжимается, чтобы сэкономить немного места на графике
3.4.3.	Оси пересекаются в начале координат	В случае наличия галочки на шкале делений отображаются только крайние значения
3.4.4	Номера отображения	В случае наличия галочки на шкале делений отображаются значения
3.4.5.	Отображение минимального уровня	В случае наличия галочки на шкале делений отображается минимальное значение для диапазона данной оси (см. пп. 3.1.2)
3.4.6.	Отображение максимального уровня	В случае наличия галочки на шкале делений отображается максимальное значение для диапазона данной оси (см. пп. 3.1.3)
3.4.7.	Отображение номеров в середине	В случае отсутствия галочки на шкале делений отображаются только минимальное и максимальное значение для диапазона данной оси
3.4.8.	Переписать номера	
3.4.9.	Number rotation angle	Угол поворота подписей
3.4.10.	Rotation aligned to mark	В случае установки галочки значения шкалы делений при повороте отображаются правее отметок делений
3.4.11	Text for numbering replacing	Список текстов, тексты из которого будут подставляться вместо цифр шкал делений

3.5.	***Метка***	
3.5.1.	ID текста	Текст из списка текстов, который будет использоваться в качестве названий данной оси
3.5.2.	Цвет	Цвет названия оси
3.5.3.	Вращение	Расположение названия оси. Возможные значения: **0°/90°/270°**
3.5.4.	Шрифт	Шрифт названия оси
3.6.	***Линии уровня*** Линии уровня перпендикулярны настраиваемой оси и используется для отображения отметок на графике	
3.6.1.	Номер линии	Номер линии уровня, для которой настраиваются параметры пп. 3.6.2-3.6.9. Поддерживается до 10 линий для каждой оси
3.6.2.	Значение	Значение, по которому проходит линия уровня
3.6.3.	Стиль	Стиль линии уровня. Возможные значения: **Сплошная/Тире/Точки/Тире Точка/Тире Точка Точка**
3.6.4.	Ширина линии	Ширина линии уровня в пикселях (1…20)
3.6.5.	Цвет	Цвет линии уровня
3.6.6.	ID текста метки	Текст из списка текстов, который будет использоваться в качестве названий данной линии уровня
3.6.7.	Позиция метки	Положение названия относительно линии уровня. Возможные значения: **Встроенный** (на одном уровне с линией уровня)/**Выше/Ниже/Слева/Справа**
3.6.8.	Вращение метки	Расположение названия линии уровня. Возможные значения: **0°/90°/270°**
3.6.9.	Шрифт	Шрифт названия линии уровня

8 Отображение значений переменных. Форматы вывода и спецификаторы

8.1 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**:

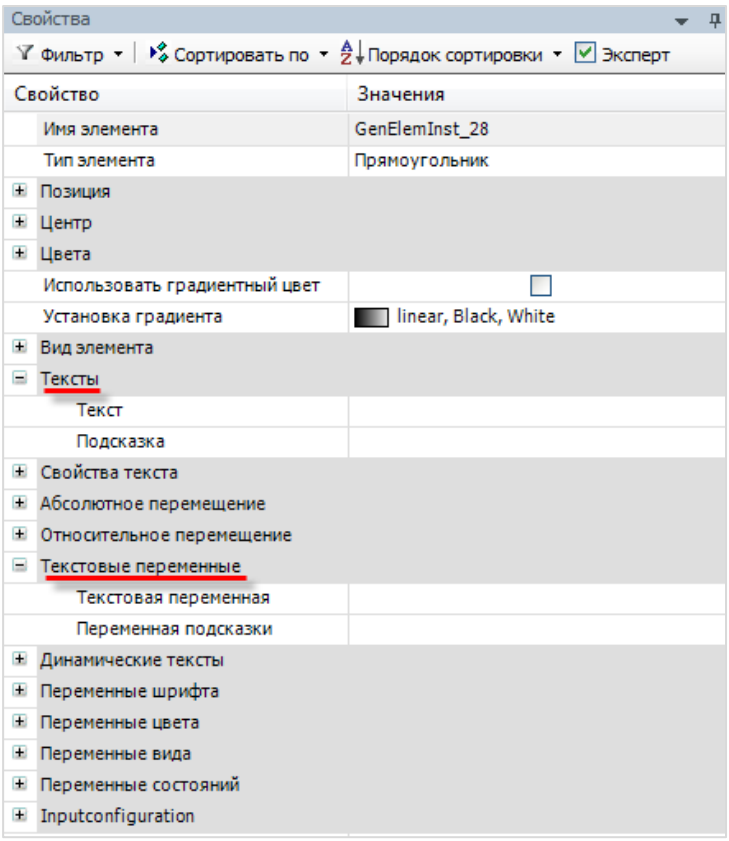


Рисунок 8.1 – Свойства элемента, вкладки Текстовые переменные и Текст

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 строчке текста элемента отображать значение переменной в представлении с плавающей точкой, а на другой – в экспоненциальной форме).

8.2 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 переменных с плавающей точкой), количество выводимых символов (для строковых переменных).

<спецификатор формата вывода переменной> – символ формата вывода отображаемой переменной. Список символов приведен в таблице 8.1:

Таблица 8.1 – Спецификаторы типов переменных

Символ	Тип отображаемых данных	Пример использования		
		Фактическое значение	Форматирование	Отображаемое значение
%d, %i	Десятичное число со знаком	10	%d	10
		7	%03d	007
%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) для переменной с размером до 32 бит	10	%x	a
%llx	Шестнадцатеричное число без знака (без ведущего нуля) для переменной с размером до 64 бит	16#4FFF_3FFF_2FFF_1FFF	%llx	4fff3fff2fff1fff
%u	Десятичное число без знака	-10	%u	10
%c	Один символ таблицы ASCII	16#21	%c	!
%s	Строка	abcdef	%s %.3s	abcdef abc
%f	Действительное число с плавающей точкой	10.1111	%f %.2f	10.111100 10.11
%e, %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. 8.4*.	Время	12:48:52	%t[HH:mm:ss]	12:48:52
%%	Символ процента	-	%%	%

8.3 Управляющие последовательности для STRING переменных

Список доступных управляющих последовательностей для переменных типа **STRING/WSTRING**, которые используются во время работы с текстом (переход на новую строку, возврат каретки и т. д.), приведен в таблице 8.2:

Таблица 8.2 – Управляющие последовательности для строковых переменных

Символ	Результат использования/Отображаемое значение
$$	$ (символ доллара)
$’	‘ (апостроф)
$L	Перевод строки
$N	Новая строка
$R	Возврат каретки
$P	Новая страница
$T	Табуляция
$xx (xx – код символа в **HEX**)	Символ таблицы ASCII (только для STRING)

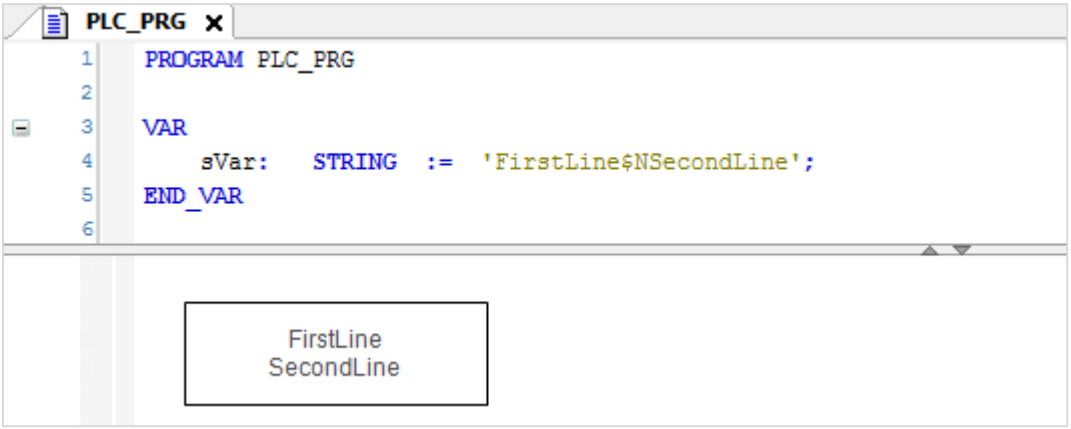


Рисунок 8.2 – Использование управляющих последовательностей

8.4 Системное время в визуализации

В случае использования спецификатора **%t[формат времени**] можно отображать значения переменных типа **TIME**, **DATE_AND_TIME (DT)** и т. д.

ПРИМЕЧАНИЕ
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 таблице 9.3:

Таблица 8.3 – Заполнители формата времени

Заполнитель	Отображаемое значение	Пример отображения
ddd	Сокращенное название дня недели	Fri (пятница)
dddd	Полное имя дня недели	Monday (понедельник)
ddddd	День недели в виде числа	1 (понедельник), 7 (воскресение),
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 0–24	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
us	Микросекунды 0–999	888
ns	Наносекунды 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’]**).

ПРИМЕЧАНИЕ
Название месяцев хранятся в списке текстов **System** на панели POU и могут быть отредактированы пользователем.

В прошлых версиях CODESYS для ввода в визуализации переменной типа **TIME/LTIME** нужно было использовать МЭК-нотацию (например, **T#22h**). В **CODESYS V3.5 SP17** добавили возможность вводить такие значение в более простой форме. Для этого добавлена возможность для спецификаторов формата времени указывать «число знаков после запятой». Например:

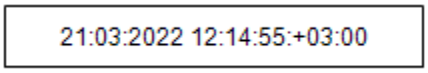
- **%t[HH2]** – ввод часа, число знаков после запятой – **2**. Если ввести **6.25**, то в переменную будет записано **T#6h15m** («6 с четвертью часов»);

- **%t[HH4]** – ввод часа, число знаков после запятой – **4**. Если ввести **6.1250**, то в переменную будет **записано T#6h7m30s** («6 с одной восьмой часов»).

Функционал не позволяет в рамках одного элемента ввести значения для нескольких разрядов времени (т. е. нельзя настроить элемент так, чтобы ввод значения **6.25** означал «6 часов 25 минут»).

Начиная с версии **CODESYS V3.5 SP17** поддерживается:

- возможность использовать в спецификаторах формата времени часовой пояс (Z). Пример: **%t[dd:MM:yyyy HH:mm:ss:ms:Z]**. Спецификатор можно использовать в:
  - элементах визуализации, отображающих системное время (т. е. у которых в параметре **Тексты** указаны спецификаторы формата времени, но к которым не привязано текстовой переменной);
  - тренде (параметр **Интернационализация/Время**);
  - настройках таблицы тревог и баннера тревог (в системной переменной: **AlarmManager.AlarmGlobals.g_sTimeFormat := 'HH:mm:ss:Z'**).
- переключение времени элементов визуализации между локальным и всемирным (UTC) с помощью системной переменной **VisuElems.Visu_DateTime.DisplayUTC** (тип **BOOL**). Переменная влияет на элементы:
  - отображающие системное время (т. е. у которых в параметре **Тексты** указаны спецификаторы формата времени, но к которым не привязано текстовой переменной);
  - тренд;
  - таблицу тревог и баннер тревог;
  - аналоговые часы;
  - элемент выбора даты и времени (если к нему не привязана переменная).



8.5 Пример отображения значения переменной

Сначала следует объявить в **Глобальном списке переменных** переменную **iVar** типа **REAL** с начальным значением **11.2233**:

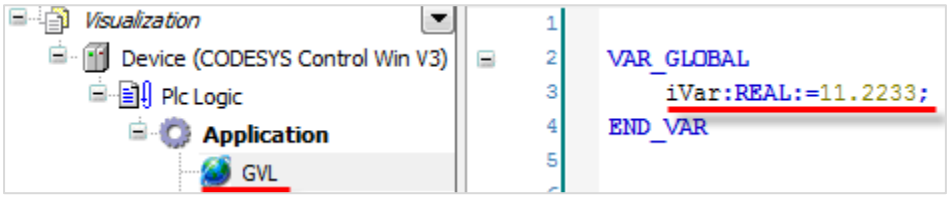


Рисунок 8.3 – Объявление переменной в Глобальном списке переменных

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

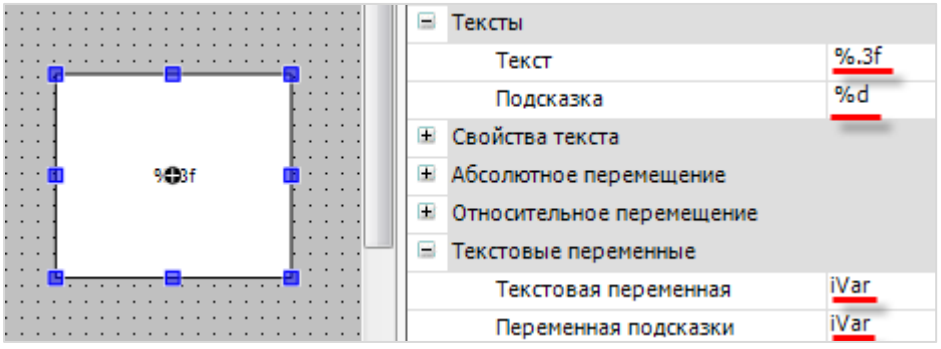


Рисунок 8.4 – Настройка параметров элемента

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

Рисунок 8.5 – Отображение значения переменной в визуализации

9 Привязка действий к элементам визуализации

9.1 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. 9.2) с помощью настройки вкладки **InputConfigutation**. Вкладка **InputConfigutation** присутствует у всех Базовых графических примитивов и Стандартных графических примитивов Кнопка, Текстовое поле, Невидимый ввод.

ПРИМЕЧАНИЕ
Вкладка **InputConfigutation** также присутствует у любого сгруппированного элемента.

Конфигурация ввода	
OnDialogClosed	Конфигурация...
OnMouseClick	Конфигурация...
OnMouseDown	Конфигурация...
OnMouseEnter	Конфигурация...
OnMouseLeave	Конфигурация...
OnMouseMove	Конфигурация...
OnMouseUp	Конфигурация...
OnValueChanged	Конфигурация...
Нажатие	
Переменная	
Переключить на FALSE	☐
Изменение при возвращении	☐
Переключение	
Переменная	
Переключение вне элемента	☐
Горячая клавиша	
Клавиша	

Рисунок 9.1 – Параметры вкладки InputConfiguration

Описание вкладки **InputConfiguration** приведено в таблице 9.1:

Таблица 9.1 – Параметры вкладки InputConfiguration

№	Название параметра	Описание параметра
1.	OnDialogClosed	Действие, привязанное к элементу, выполняется после закрытия диалога, открытого в результате нажатия на элемент
2.	OnMouseClick	Действие, привязанное к элементу, выполняется после нажатия и последующего отпускания 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 отпускания курсора, причем, в отличие от пп. 2, курсор может быть отпущен за пределами элемента
8.	OnValueChanged	Действие, привязанное к элементу, выполняется при изменении значения переменной, привязанной к элементу, с помощью самого элемента (например, с помощью экранной клавиатуры). Это, в частности, позволяет произвести валидацию нового значения после его ввода. Действие не выполняется в том случае, если значение переменной изменяется из кода программы
9.	**Нажать**	
9.1.	Переменная	Переменная типа **BOOL**, которая переключает свое значение на **TRUE** на время нажатия курсора на элемент. После отпускания курсора переменной возвращается значение **FALSE**
9.2.	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**. Если установлена галочка 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**
9.3.	Изменение при возвращении	Если установлена галочка, то переменная, указанная в п. 8.1, будет менять свое состояние, если нажатый на элементе курсор без отпускания будет перемещаться за границы элемента и обратно. То есть, 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 отпускания курсора (неважно, в пределах элемента это произойдет, или нет), переменной будет присвоено начальное значение – т. е. **FALSE**
10.	**Переключить**	
10.1.	Переменная	Переменная типа **BOOL**, которая будет переключать свое значение при нажатии и последующем отпускании курсора на элементе (т. е. клике)
10.2.	Переключение вне элемента	Если установлена галочка, то указанная переменная в п. 9.1, будет переключаться, даже если курсор отпущен за пределами элемента

Продолжение Табл. 9.1

№	Название параметра	Описание параметра
11.	**Горячая клавиша** Позволяет привязать клавишу клиента визуализации к элементу. То есть нажатие данной клавиши будет эквивалентно нажатию курсора на элемент. Данная вкладка представляет собой часть настроек Редактора горячих клавиш	
11.1.	Клавиша	Клавиша клиента визуализации
11.2.	События	Событие, вызываемое нажатием на клавишу. Возможные значения: **Нет** – событие отсутствует; **Mouse Down** – нажатие курсора на элемент (см. п. 3); **Mouse Up** – отпускание зажатого курсора с элемента (см. п. 7); **Mouse Down/Up** – нажатие и последующее отпускание курсора с элемента, «**клик**» (см. п. 2)
11.3.	Shift	Если установлена галочка, то клавиша должна нажиматься с зажатой клавишей **Shift**
11.4.	Control	Если установлена галочка, то клавиша должна нажиматься с зажатой клавишей **Control**
11.5.	Alt	Если установлена галочка, то клавиша должна нажиматься с зажатой клавишей **Alt**

9.2 Действия для OnMouse<Something>

При нажатии в **Свойствах** элемента на один из основных параметров вкладки **InputConfiguration** (см. пп. 1–8 в таблице 9.1), открывается окно конфигурации элемента, в котором содержится список действий, которые будут выполняться при нажатии (или других условиях) на элемент. Для этого следует присвоить действие элемента с помощью нажатия кнопки «**>**» и настроить его параметры:

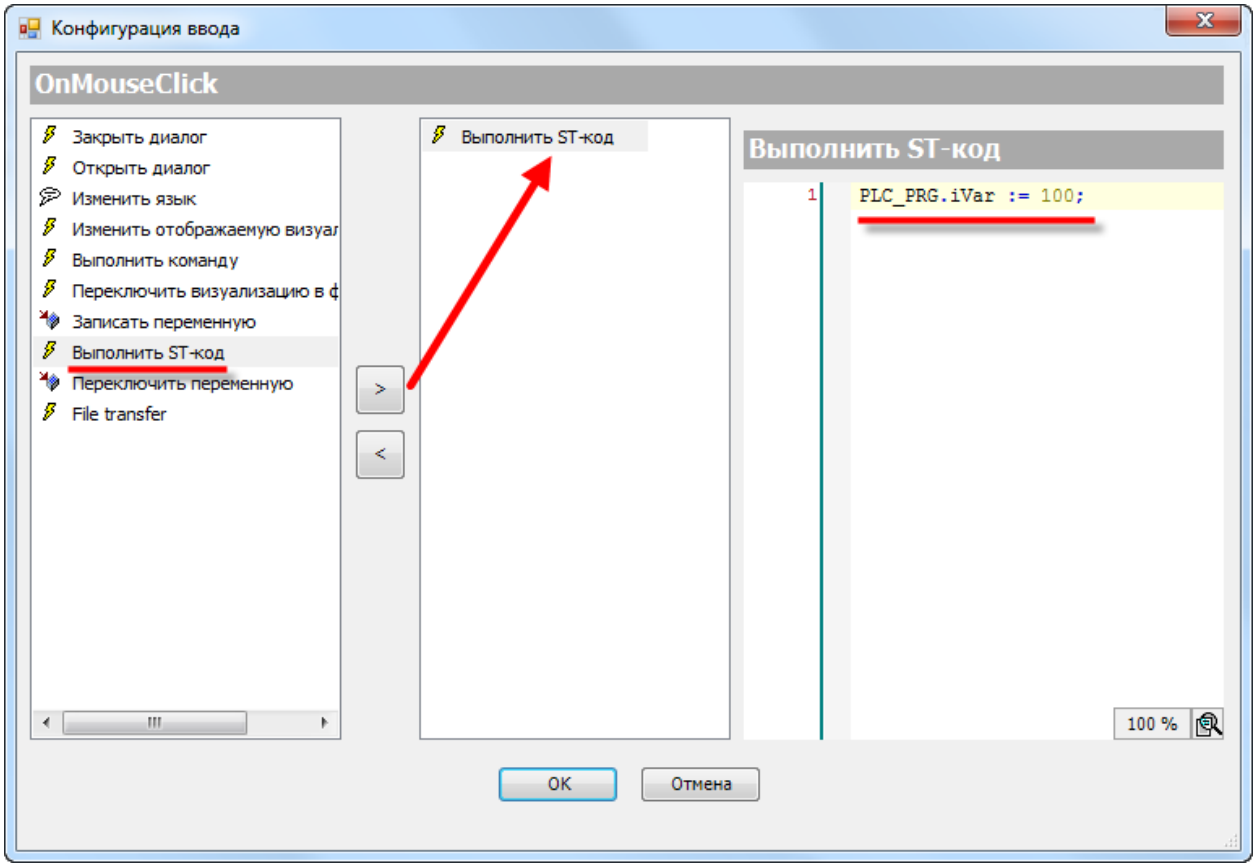


Рисунок 9.2 – Настройка действий элемента

9.2.1 Управление пользователями

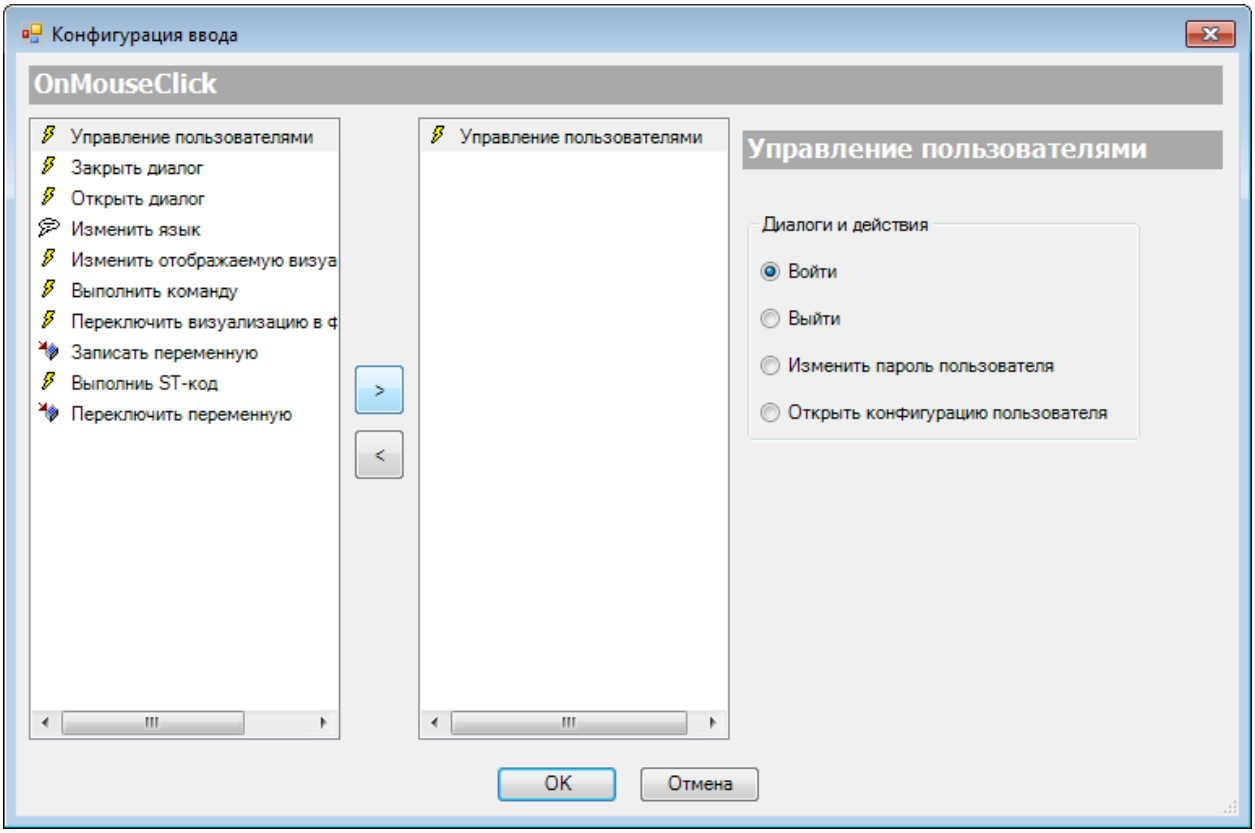


Рисунок 9.3 – Параметры действия Управление пользователями

Данная вкладка доступна только в случае наличия в проекте Управления пользователями.
Пример создания многопользовательского проекта приведен в п. 10.3.7.

Действие **Управления пользователями** используется для привязки к элементу одного из диалогов библиотеки **VisuUserManagement** (или другой библиотеки, диалоги которой выбраны в менеджере визуализации на вкладке Установки диалога в качестве диалогов управления пользователями):

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

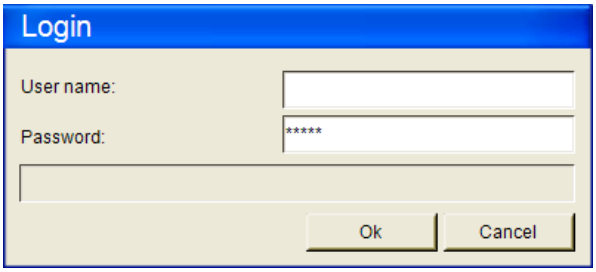


Рисунок 10.4 – Диалог Войти

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

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

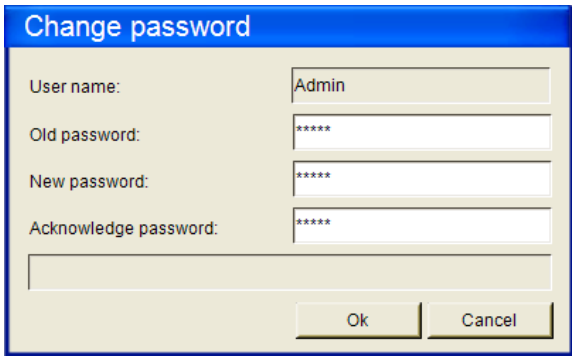


Рисунок 9.5 – Диалог Выйти

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

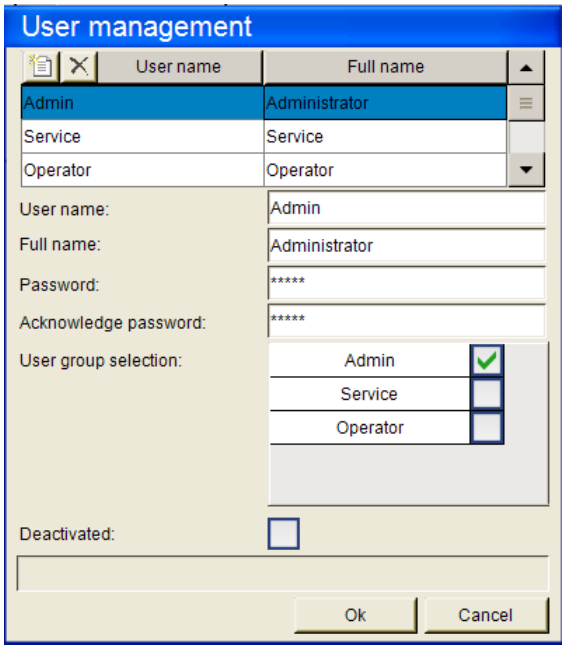


Рисунок 9.6 – Диалог Изменение конфигурации пользователей

9.2.2 Закрыть диалог

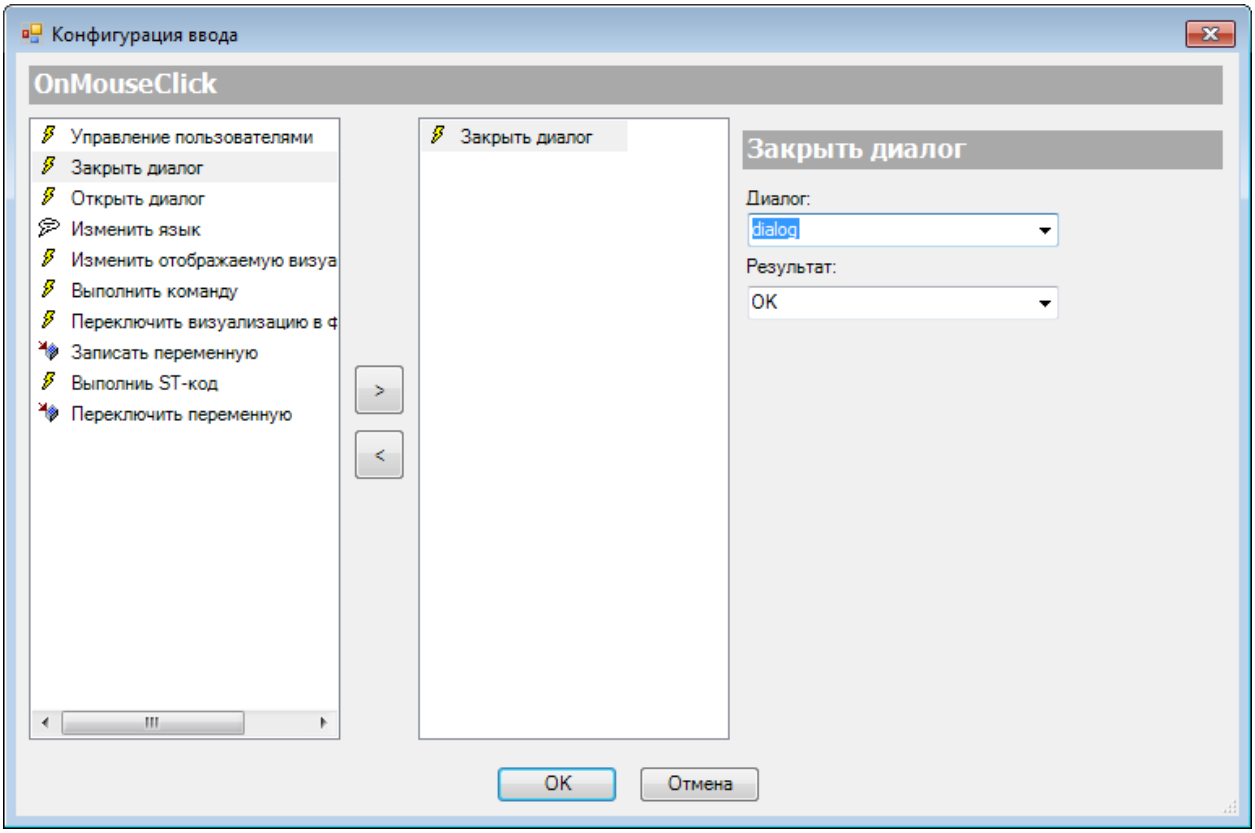


Рисунок 9.7 – Параметры действия Закрыть диалог

Действие **Закрыть 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. 9.2.3). Если действие настраивается в элементе визуализации, который размещен в диалоге, то по умолчанию будет подставлен заполнитель **Current Dialog** для закрытия данного диалога.

Пример работы с диалогами приведен в п. 10.3.2.

9.2.3 Открыть диалог

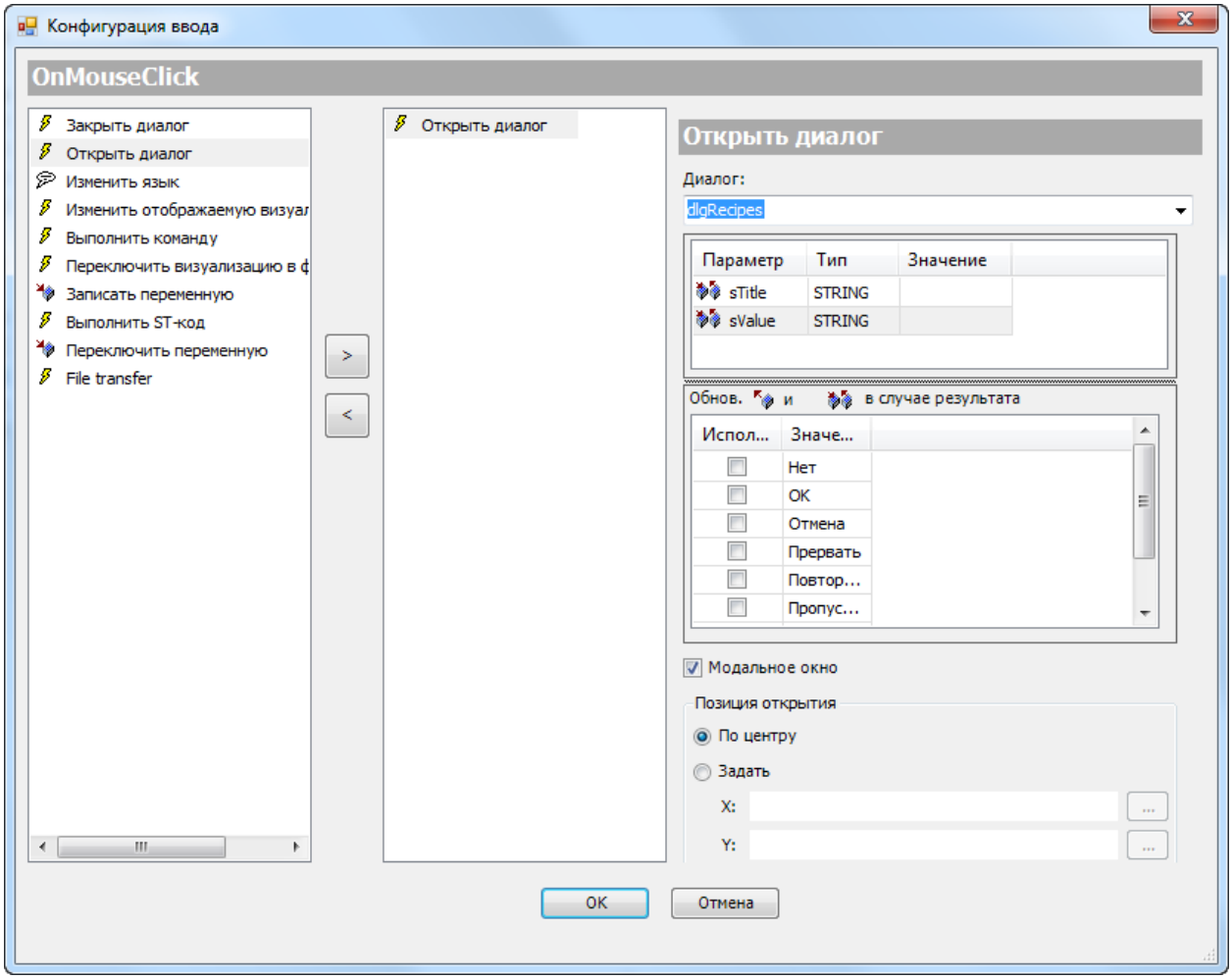


Рисунок 9.8 – Параметры действия Открыть диалог

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. 9.2.2).
3. Галочка модальности диалога (**Open Dialog Modal**) – если установлена галочка, то 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.

Пример работы с диалогами приведен в п. 10.3.2.

9.2.4 Изменить язык

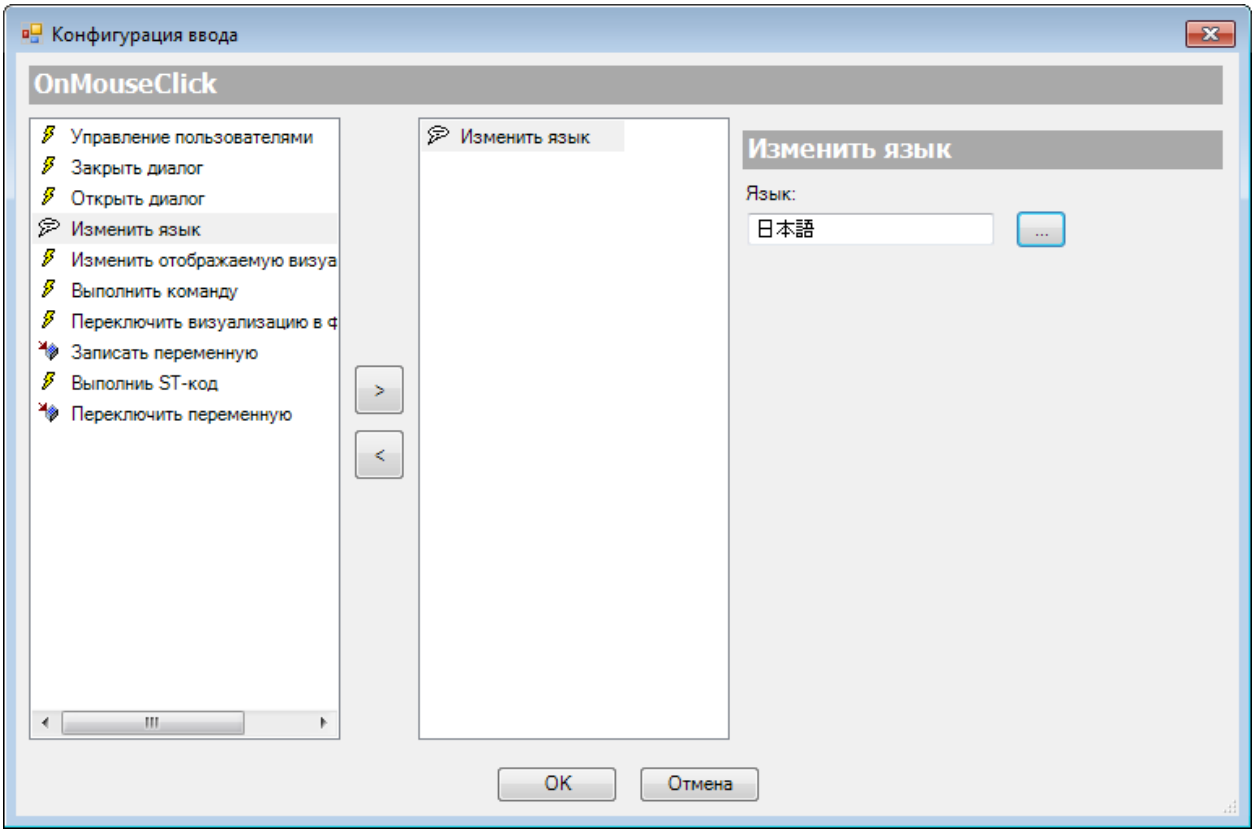


Рисунок 9.9 – Параметры действия Изменить язык

Действие **Изменить язык** приводит к смене текущего языка визуализации. Предварительно язык должен быть добавлен в Список текстов.

Пример создания мультиязычного проекта приведен в п. 10.3.5.

9.2.5 Изменить отображаемую визуализацию

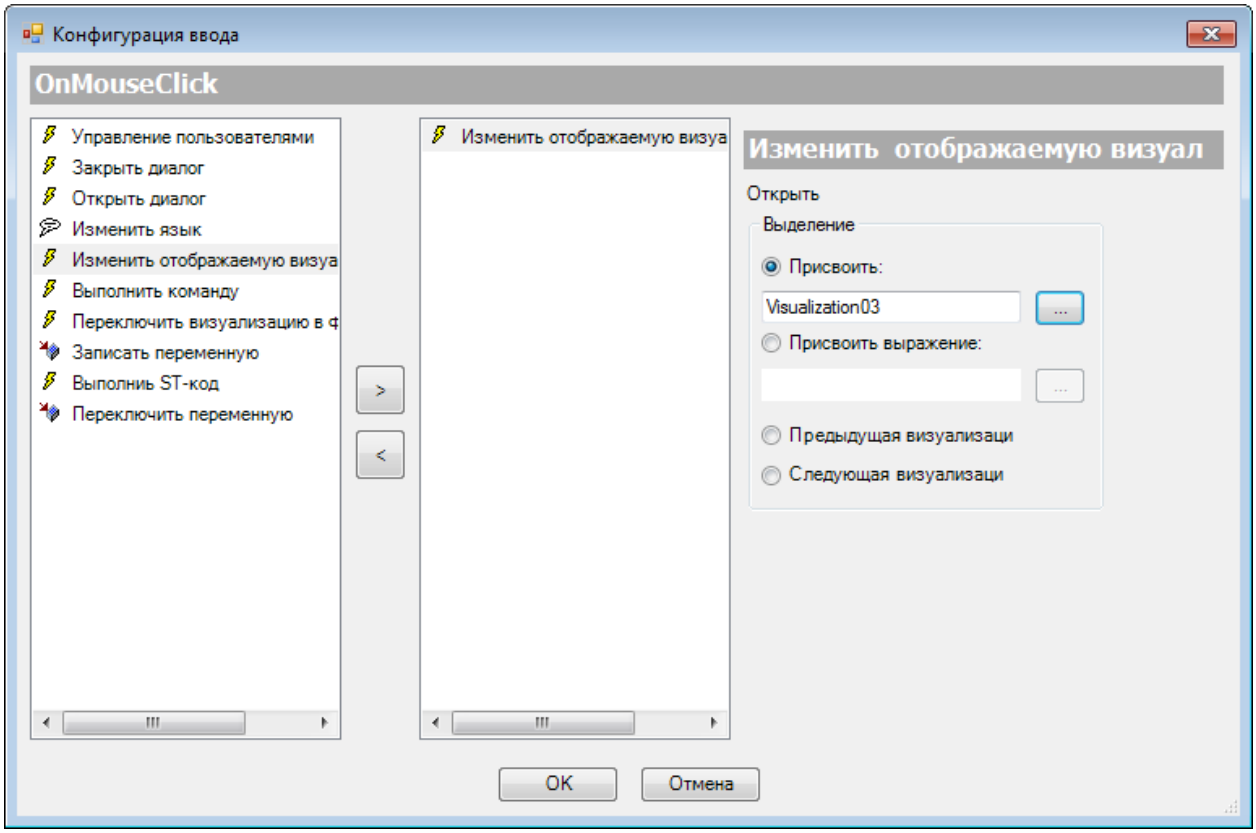


Рисунок 9.10 – Параметры действия Изменить отображаемую визуализацию

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. 10.3.1.

9.2.6 Выполнить команду

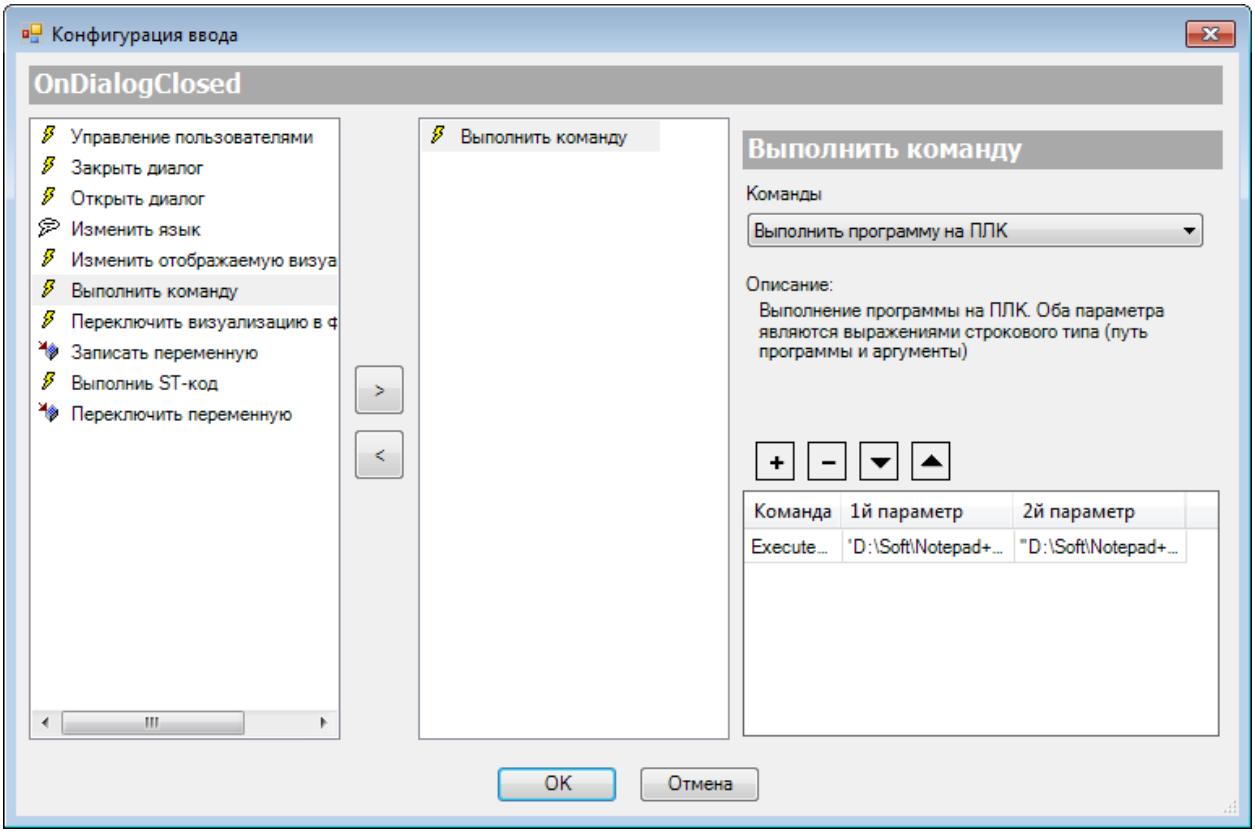


Рисунок 9.11 – Параметры действия Выполнить команду

Действие **Выполнить команду** приводит к выполнению команды (или группы команд) на целевом устройстве. Добавление команды в список выполняемых осуществляется с помощью кнопки «**+**», удаление с помощью кнопки «**–**», изменение порядка выполнения команд – с помощью кнопок «**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 таблице 9.2:

Таблица 9.2 – Список команд действия Выполнить команду

Команда	Описание команды	Первый параметр	Второй параметр
Выполнить программу на ПЛК/клиенте (ExecutePlcProgram/ ExecuteClientProgram)	Запуск приложения/открытие файла с помощью приложения	Путь к приложению	Путь к файлу
Пример	Открытие текстового файла **readme.txt** с помощью приложения **Notepad++.** Первый параметр: 'D:\Soft\Notepad++\notepad++.exe' Второй параметр: 'D:\Soft\Notepad++\readme.txt' **Данный функционал работает только на ПЛК с ОС Windows и не поддерживается в web-визуализации**		
Печать (Print)	Печать содержимого экрана визуализации	Отсутствует	Отсутствует
Комментарий	**Данный функционал работает только на ПЛК с ОС Windows и не поддерживается в web-визуализации**		
Navigate to URL	Переход по ссылке в веб-клиенте визуализации	Адрес веб-страницы	При отсутствии значения, веб-страница открывается в новом окне. При значении **‘replace’**, веб-страница открывается вместо веб-визуализации
Пример	Открытие в новом окне страницы www.owen.ru Первый параметр: ‘https://owen.ru’ Второй параметр: ‘’ **Данный функционал работает только на клиентах web-визуализации**		
Команды работы с **Менеджером рецептов** (см. п. 10.3.8)			
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		

9.2.7 Переключить визуализацию в фрейме

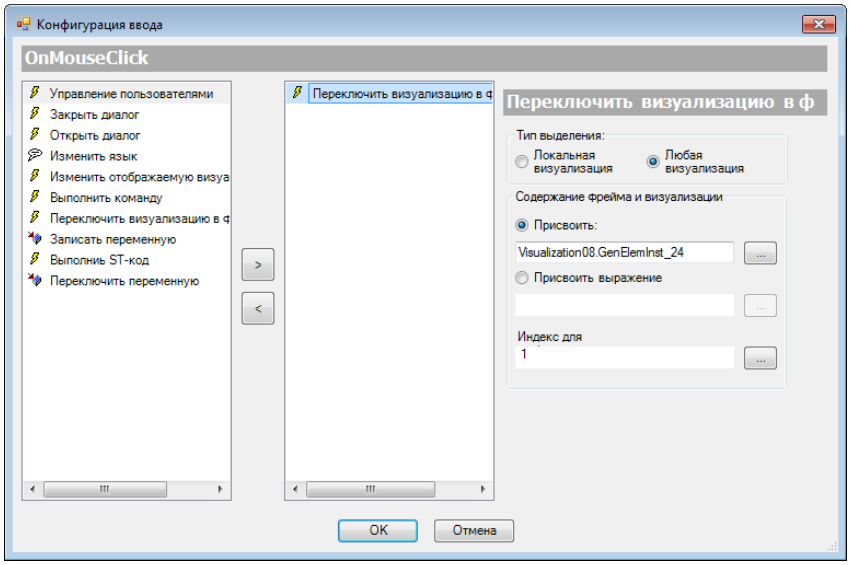


Рисунок 9.12 – Параметры действия Переключить визуализацию во фрейме

Действие **Переключить визуализацию во фрейме** приводит к смене экрана визуализации в элементе типа Фрейм.

Тип выделения характеризует расположение фрейма в проекте – режимы **Локальная визуализация** и **Пошаговая локальная визуализация** позволяют переключать только визуализации фреймов, расположенных на данном экране визуализации, в то время как режим **Любая визуализация** позволяет делать это с любым фреймом проекта. Для режима **Пошаговая локальная визуализация** пользователь выбирает не конкретный экран фрейма, а одну из следующих опций: **Переключить на предыдущий/Переключить на следующий/Переключить на первый/Переключить на последний.**

В случае использования **Любой визуализации** пользователь может 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), первому из экранов соответствует индекс **0**, второму – **1**, и т. д.

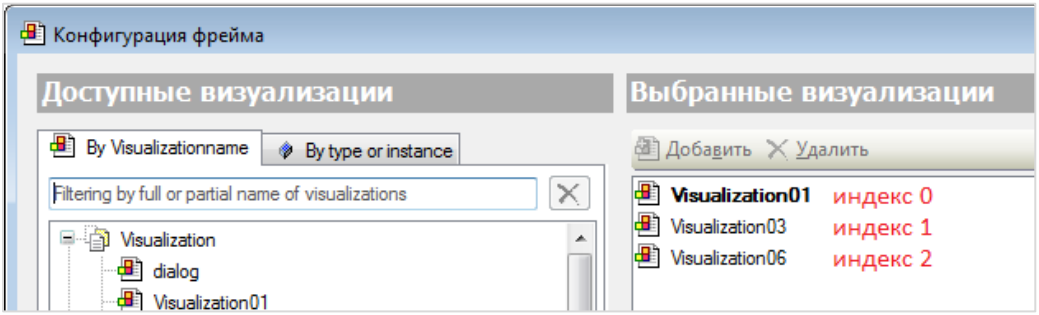


Рисунок 9.13 – Конфигурация фрейма

Примеры реализации переключения экранов во фрейме приведены в п. 10.2.2 и п. 10.3.3.

9.2.8 Записать переменную

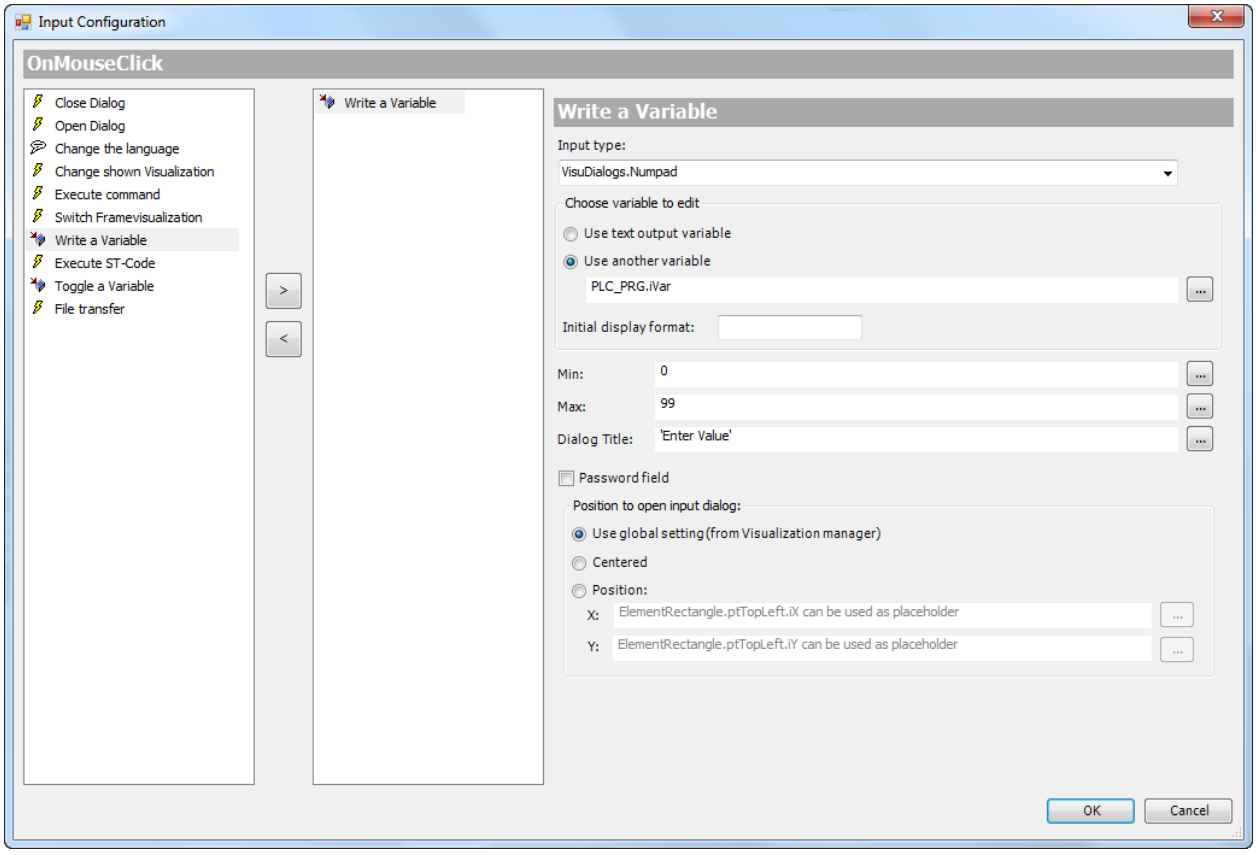


Рисунок 9.14 – Параметры действия Записать переменную

ПРИМЕЧАНИЕ
Данная вкладка отображается некорректно в русскоязычной версии **CODESYS V3.5 SP11 Patch 5**. Для изменения языка интерфейса следует в меню **Инструменты** выбрать пункт **Опции**, перейти в во вкладку пункт **Международные установки** и изменить язык интерфейса на английский.

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**);
- **VisuDialogs.NumpadExtended** – используется цифровая экранная клавиатура с возможностью ввода шестнадцатеричных (HEX) значений (диалог **NumpadExtended** из библиотеки **VisuDialogs**);

В качестве изменяемой переменной можно выбрать текстовую переменную, привязанную к элементу (**Использовать текстовую выходную переменную**), либо указать необходимую переменную вручную (**Использовать другую переменную**). Параметр **Initial display format** определяет спецификатор формата вывода переменной диалога и ее пределов.

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 рисунке 10.14).

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

Если установлена галочка **Поле пароля**, то вводимое значение будет отображаться звездочками (*****).

Во вкладке **Position to open input dialog** можно выбрать координаты открытия диалога:

- **Use global settings** – используются настройки из Менеджера визуализации;
- **Centered** – диалог открывается по центру экрана;
- **Position** – координаты определяются целочисленным переменными. Заполнители **ElementRectangle.ptTopLeft.iX** и **ElementRectangle.ptBottomRight.iY** соответствуют координатам элемента, нажатие на который открывает диалог.

9.2.9 Выполнить ST-код

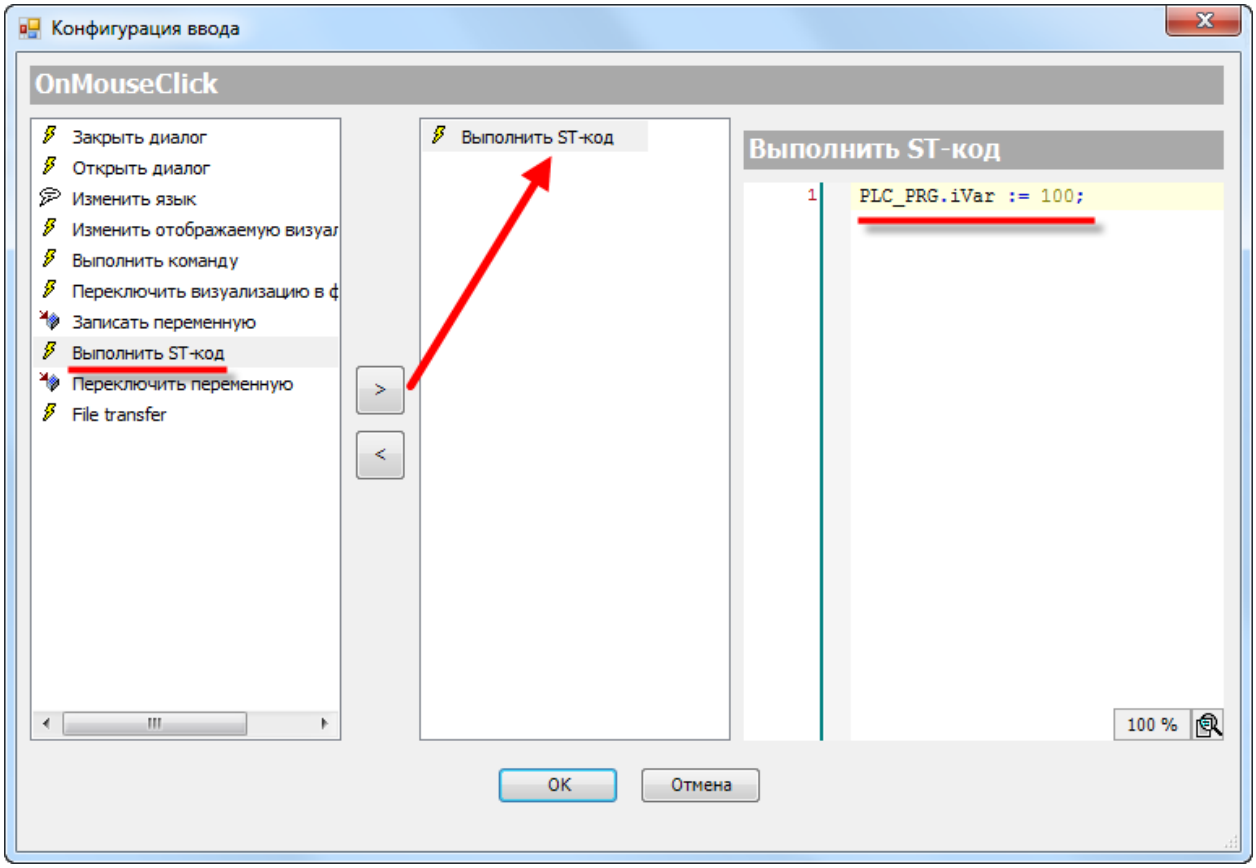


Рисунок 9.15 – Параметры действия Выполнить ST-код

Действие **Выполнить ST-код** используется для выполнения кода на языке **ST**. Используемые переменные должны быть объявлены в проекте.

9.2.10 Переключить переменную

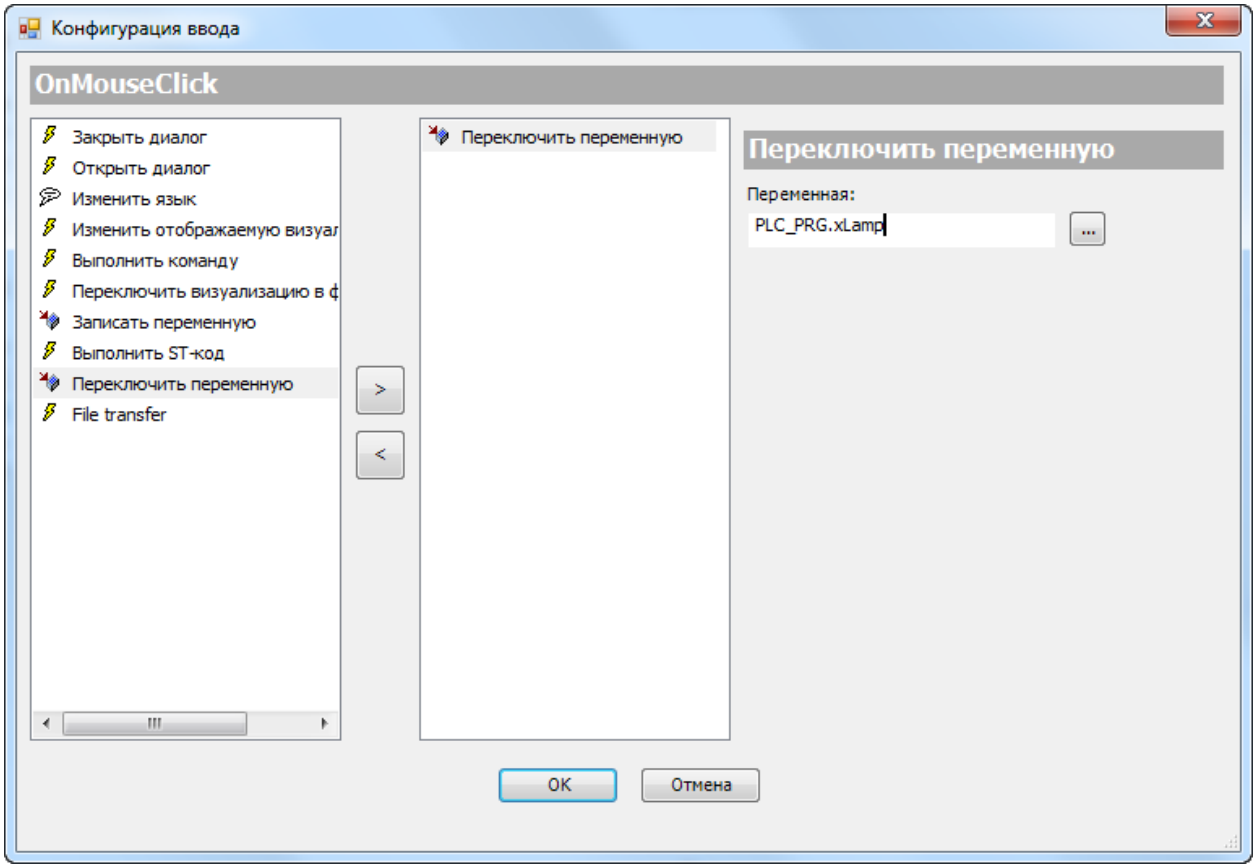


Рисунок 9.16 – Параметры действия Переключить переменную

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

9.2.11 Передача файлов

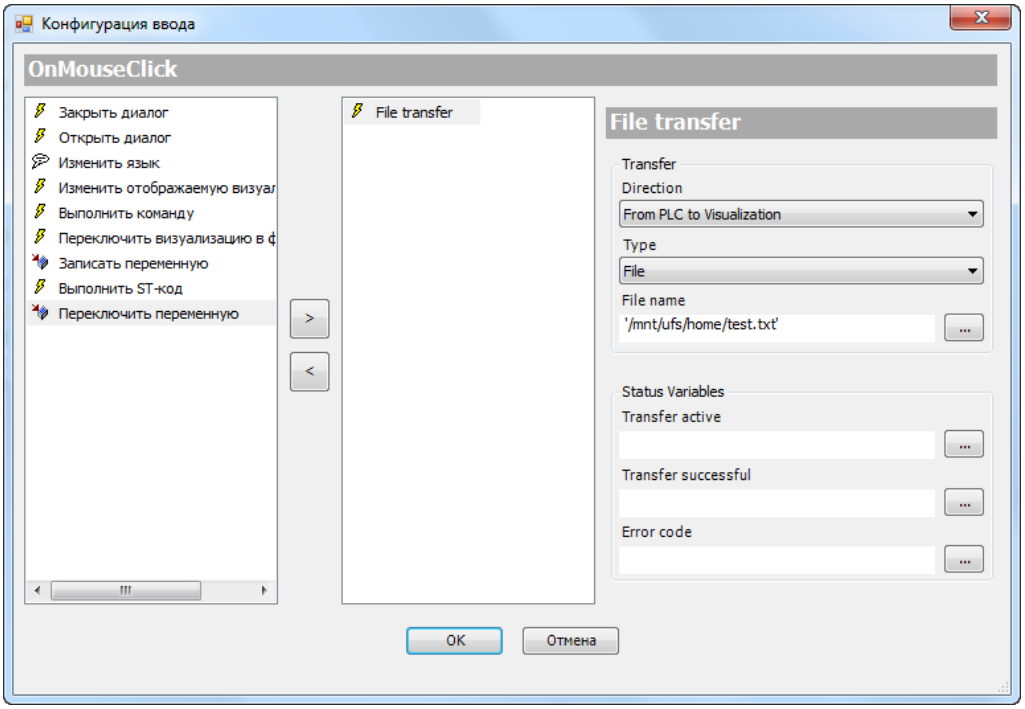


Рисунок 9.17 – Параметры действия Передача файлов

Действие **Передача файлов** используется для передачи файлов между контроллером и клиентами web-визуализации.

Параметр **Direction** определяет направление передачи файлов:

- **From PLC to Visualization** – web-клиент загружает файл из контроллера;
- **From Visualization to PLC** – web-клиент загружает файл в контроллер.

Параметр **Type** определяет тип передачи: передача файла (**File**) или передача потока байт (**Streaming**).

Параметр Name содержит полный путь к файлу (**File**) или имя объекта с интерфейсом **IVisuStreamReader/IVisuStreamWriter** (**Streaming**).

Во вкладке **Status Variables** можно привязать переменные для контроля передачи:

- **Transfer Active** – переменная типа **BOOL**, имеет значение TRUE, пока осуществляется передача данных;
- **Transfer Successful** – переменная типа **BOOL**, принимает значение TRUE после окончания передачи данных;
- **ErrorCode** – переменная типа **VU.ERROR**, содержит код ошибки.

Внешний вид диалога загрузки/выгрузки файлов может быть изменен путем редактирования CSS в файле в памяти ПЛК: **/<рабочая директория CODESYS>/PlcLogic/Visu/webvisu.htm**

10 Примеры

10.1 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. Так как число примитивов достаточно велико и 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. Так как 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.

В случае необходимости большинство примеров можно запустить на контроллерах ОВЕН, выбрав соответствующий таргет-файл. Исключение составляют только примеры, затрагивающие обширные предметные области (например, Типичные параметры графических примитивов, Текстовый редактор, Таблица тревог, Менеджер рецептов) – размеры их экранов визуализации превышают **800 × 480,** в связи с чем они будут некорректно отображаться на дисплее контроллеров **СПК**.

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

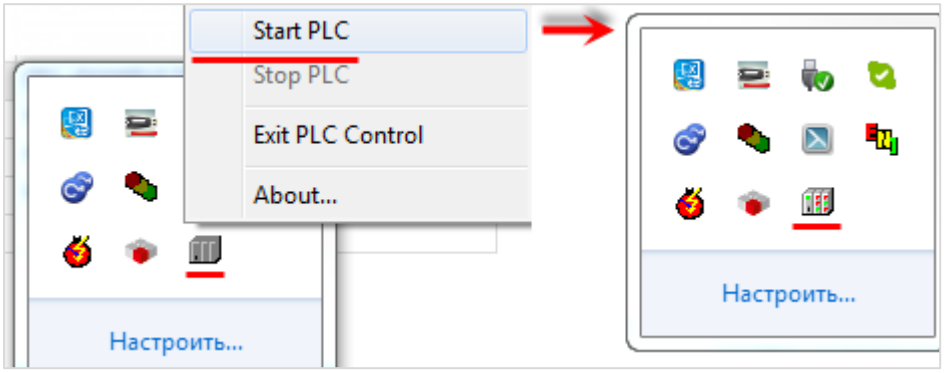


Рисунок 10.1 – Запуск виртуального контроллера

В установках соединения **CODESYS** следует просканировать сеть и выбрать устройство, имя которого совпадает с именем ПК:

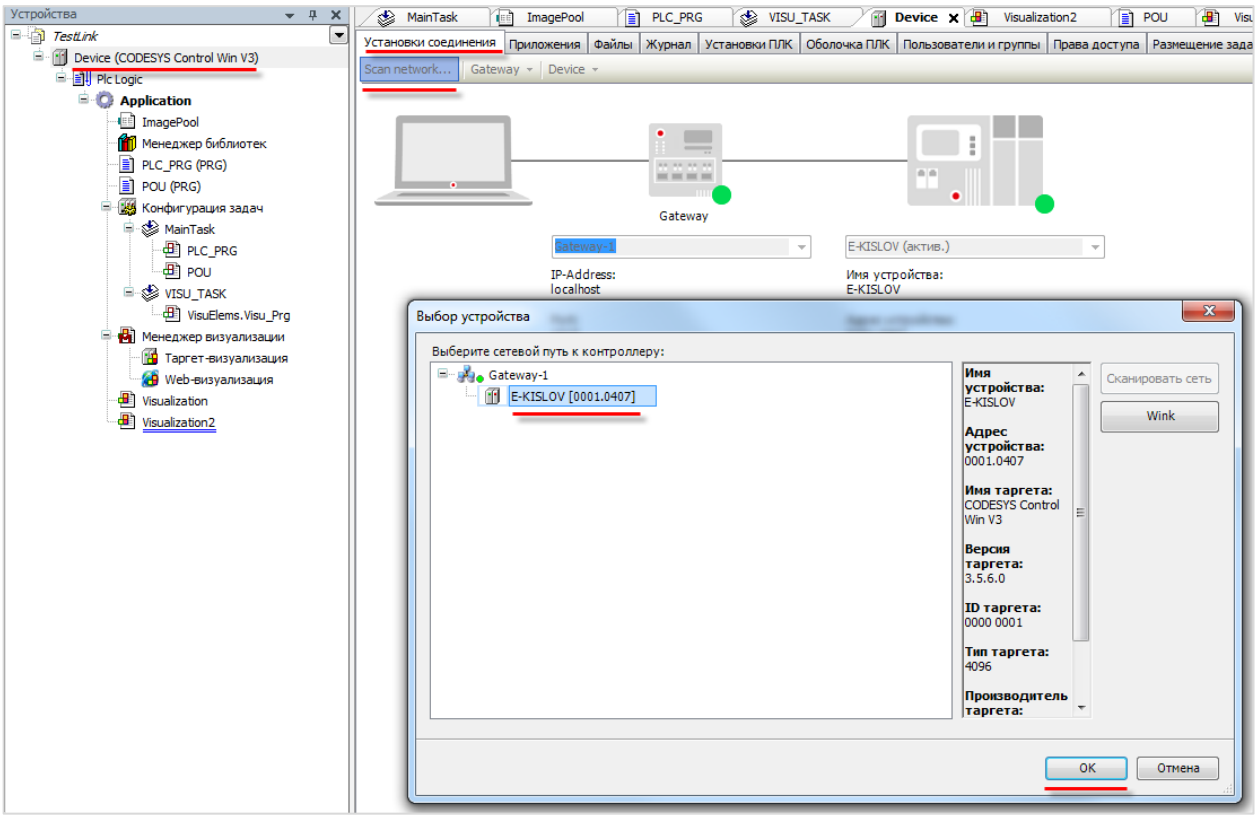


Рисунок 10.2 – Окно сканирования сети. Подключение к виртуальному контроллеру

ПРИМЕЧАНИЕ

Перед загрузкой проекта в виртуальный контроллер рекомендуется выполнить команды вкладки **Компиляция** – **Очистить все** и **Перекомпиляция**.

ПРИМЕЧАНИЕ

Вместе с **CODESYS** распространяется версия виртуального контроллера с **ограничением времени** непрерывной работы (около получаса). После этого контроллер можно перезапустить. В случае необходимости лицензию на виртуальный контроллер можно приобрести у компании CODESYS Group (разработчик **CODESYS**).

10.2 Примеры работы с графическими примитивами

10.2.1 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). 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.

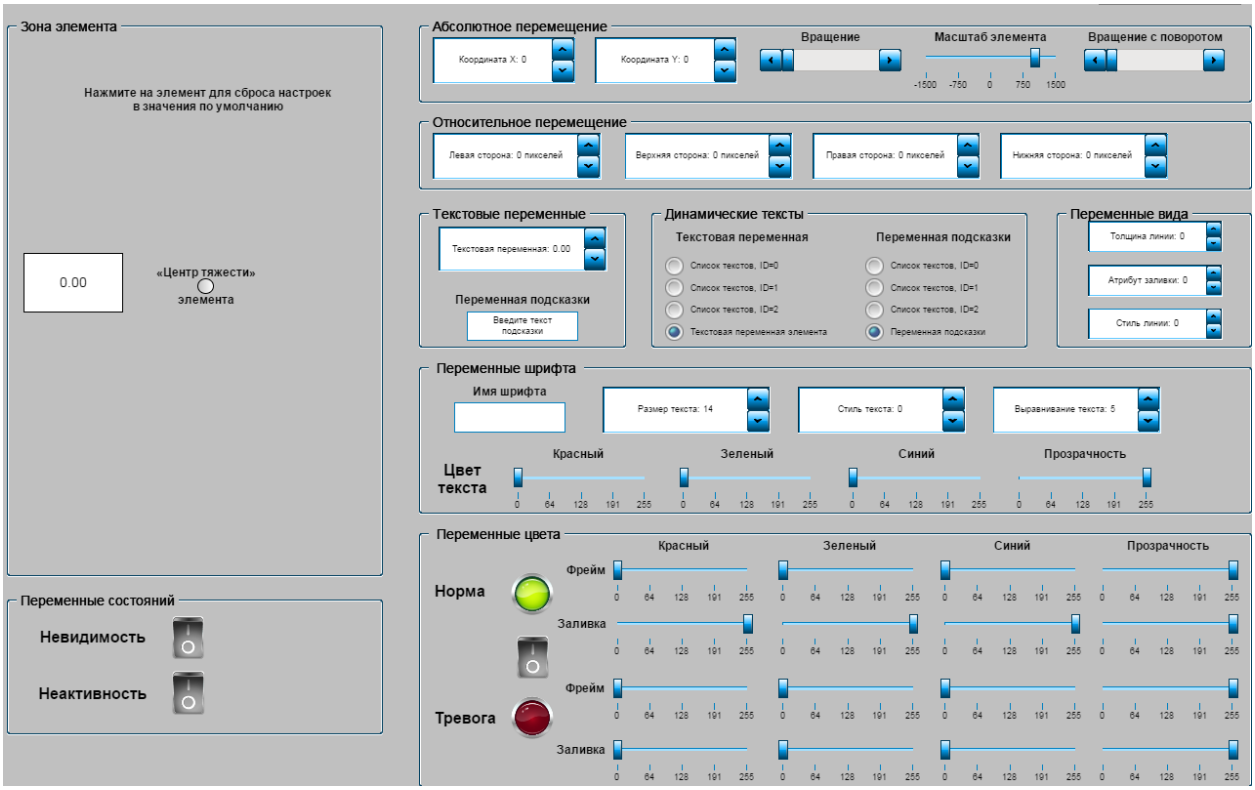


Рисунок 10.3 – Внешний вид примера Типичные свойства графических примитивов

Пример создан в среде **CODESYS V3.5 SP11 Patch 5** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с таргет-файлом **3.5.11.5**.

Пример доступен для скачивания: Example_Rectangle.projectarchive

Для создания примера следует:

1. Создается новый ***стандартный*** проект **CODESYS** с названием **Example_Rectangle** и настройками по умолчанию: таргет – CODESYS Control Win V3, язык PLC_PRG – **ST**.
2. В программе **PLC_PRG** объявляются следующие переменные:

PLC_PRG

```
PROGRAM PLC_PRG
VAR
    //Абсолютное перемещение                                    // Название параметра в CODESYS
    iCoord_X:                   INT;                            // X
    iCoord_Y:                   INT;                            // Y
    iRotation:                  INT;                            // Вращение
    iScale:                     INT :=  1000;                   // Масштабирование
    iSelfRotation:              INT;                            // Внутреннее вращение

    // Относительное перемещение                                // Название параметров в CODESYS некорректны
    iLeftSideMovement:          INT;                            // Перемещение сверху налево: X
    iUpSideMovement:            INT;                            // Перемещение сверху налево: Y
    iRightSideMovement:         INT;                            // Перемещение снизу вверх: X
    iDownSideMovement:          INT;                            // Перемещение снизу вверх: Y

    // Текстовые переменные
    rElementVariable:           REAL;                           // Текстовая переменная
    wsTooltip:                  WSTRING :=  "Введите текст$Nподсказки";     // Переменная подсказки

    // Динамические тексты
    sTextIndex:                 STRING;                         // Индекс текста
    sTooltipIndex:              STRING;                         // Индекс подсказки
    iTextIndex:                 INT :=  3;                      // дополнительные переменные...
    iTooltipIndex:              INT :=  3;                      // ...для радио-кнопок

    // Переменные шрифта
    sFontName:                  STRING;                         // Имя шрифта
    iTextSize:                  INT :=  14;                     // Размер
    dwTextFlag:                 DWORD;                          // Флаги
    // dwTextCharset:           DWORD;                          // Набор символов. Не используется в примере
    dwTextColor:                DWORD;                          // Цвет
    iTextAligment:              INT :=  5;                      // Flags for text aligment

    // Переменные цвета
    bSwitchStateColor:          BOOL;                           // Переключить цвет
    dwNormalState_FrameColor:   DWORD;                          // Нормальное состояние: Цвет фрейма
    dwNormalState_FillColor:    DWORD;                          // Нормальное состояние: Цвет заливки
    dwAlarmState_FrameColor:    DWORD;                          // Состояние тревоги: Цвет фрейма
    dwAlarmlState_FillColor:    DWORD;                          // Состояние тревоги: Цвет заливки

    // Переменные вида
    iLineThickness:             INT;                            // Ширина линии
    dwFillEnable:               DWORD;                          // Атрибуты заливки
    dwLineStyle:                DWORD;                          // Стиль линии

    //Переменные состояний
    bInvisible:                 BOOL;                           // Невидимый
    bNoTouchReaction:           BOOL;                           // Отключение ввода
```

Рисунок 10.4 – 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;
61      byNormalState_FrameColorGreen:  BYTE;
62      byNormalState_FrameColorBlue:   BYTE;
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;
```

Рисунок 10.5 – Объявление переменных программы PLC_PRG (окончание)

3. У прямоугольника есть параметры, определяющие цвет его текста, контура и заливки. К данным параметрам привязываются переменные типа **DWORD**. Переменная определяет цвет по модели ARGB. Но управлять цветом с помощью только одной переменной крайне неудобно. Поэтому предусматривается 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, потребуются четыре переменные типа **BYTE**, которые будут привязаны к управляющим элементам визуализации (см. рисунок 10.5).

   Чтобы «собирать» цвет из оттенков, следует создать функцию **ARGB_TO_DWORD** на языке **ST**:

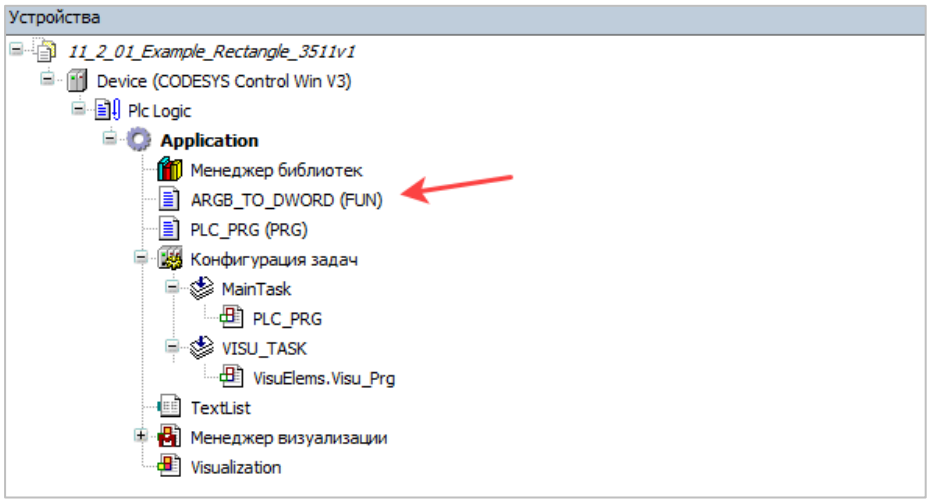


Рисунок 10.6 – Создание функции «сборки» цвета

4. Код функции **ARBG_TO_DWORD** будет выглядеть следующим образом:

ARGB_TO_DWORD

```
FUNCTION ARGB_TO_DWORD : DWORD
VAR_INPUT
    byAlpha:        BYTE;
    byRed:          BYTE;
    byGreen:        BYTE;
    byBlue:         BYTE;
END_VAR
```

```
ARGB_TO_DWORD := BYTE_TO_DWORD(byBlue) + SHL(BYTE_TO_DWORD(byGreen), 8)
        + SHL(BYTE_TO_DWORD(byRed), 16) + SHL(BYTE_TO_DWORD(byAlpha), 24);
```

Рисунок 10.7 – Код функции ARGB_TO_DWORD

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

```
// для управления динамическими текстами
sTextIndex      :=  INT_TO_STRING(iTextIndex);
sTooltipIndex   :=  INT_TO_STRING(iTooltipIndex);

// для управления цветами

// собираем цвет текста
dwTextColor := ARGB_TO_DWORD(byTextColorAlpha, byTextColorRed, byTextColorGreen, byTextColorBlue);
// собираем цвет фрейма в нормальном состоянии
dwNormalState_FrameColor := ARGB_TO_DWORD(byNormalState_FrameColorAlpha, byNormalState_FrameColorRed, byNormalState_FrameColorGreen, byNormalState_FrameColorBlue);
// собираем цвет заливки в нормальном состоянии
dwNormalState_FillColor := ARGB_TO_DWORD(byNormalState_FillColorAlpha, byNormalState_FillColorRed, byNormalState_FillColorGreen, byNormalState_FillColorBlue);
// собираем цвет фрейма в состоянии
dwAlarmState_FrameColor := ARGB_TO_DWORD(byAlarmState_FrameColorAlpha, byAlarmState_FrameColorRed, byAlarmState_FrameColorGreen, byAlarmState_FrameColorBlue);
// собираем цвет заливки в состоянии тревоги
dwAlarmlState_FillColor := ARGB_TO_DWORD(byAlarmState_FillColorAlpha, byAlarmState_FillColorRed, byAlarmState_FillColorGreen, byAlarmState_FillColorBlue);
```

Рисунок 10.8 – Код программы PLC_PRG

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

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

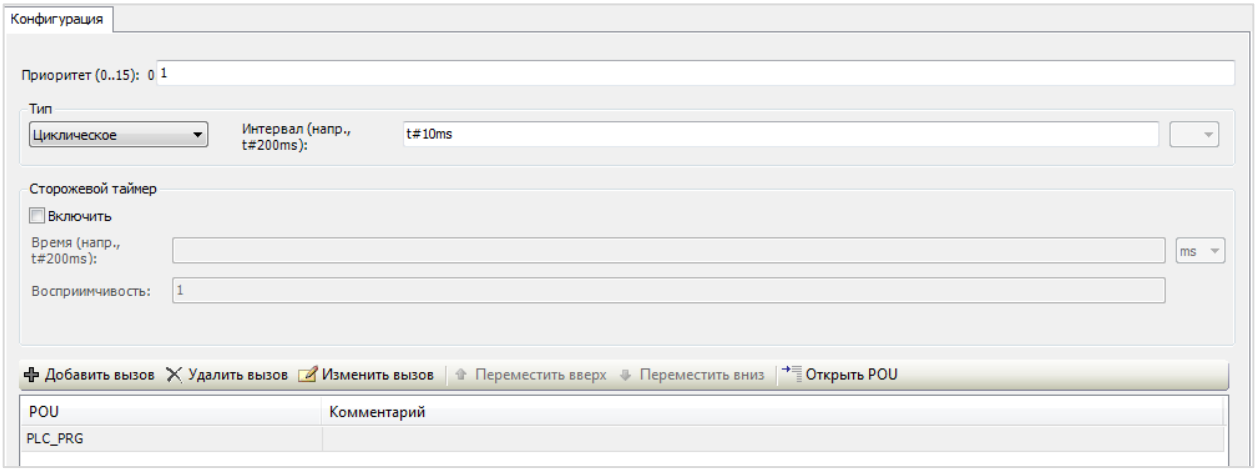


Рисунок 10.10 – Настройки частоты вызова задачи MainTask

7. Добавить в проект экран визуализации с названием **Visualization**. В его свойствах выбрать размер **1760 × 1080.** Подразумевается, что проект будет запускаться на виртуальном контроллере с отображением в веб-браузере.
   Создание экрана визуализации приведет к автоматическому добавлению в проект Менеджера визуализации.

   Ему следует задать следующие настройки:

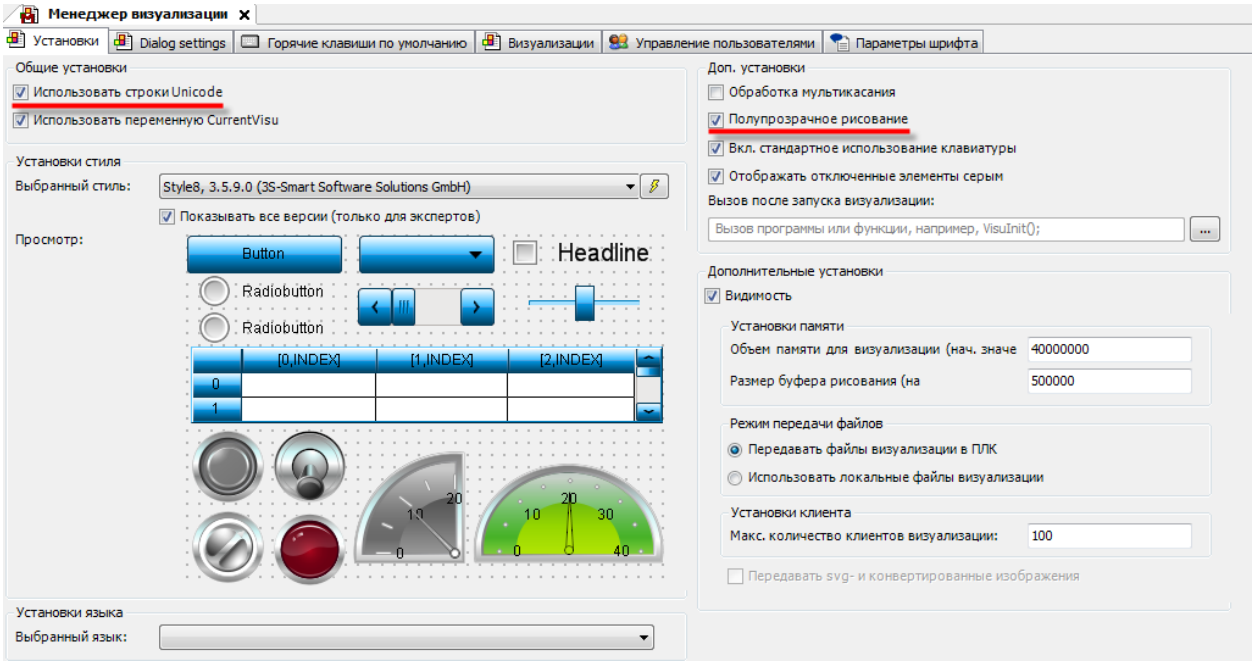


Рисунок 10.11 – Настройки Менеджера визуализации

Настройки таргет- и web-визуализации будут следующими:

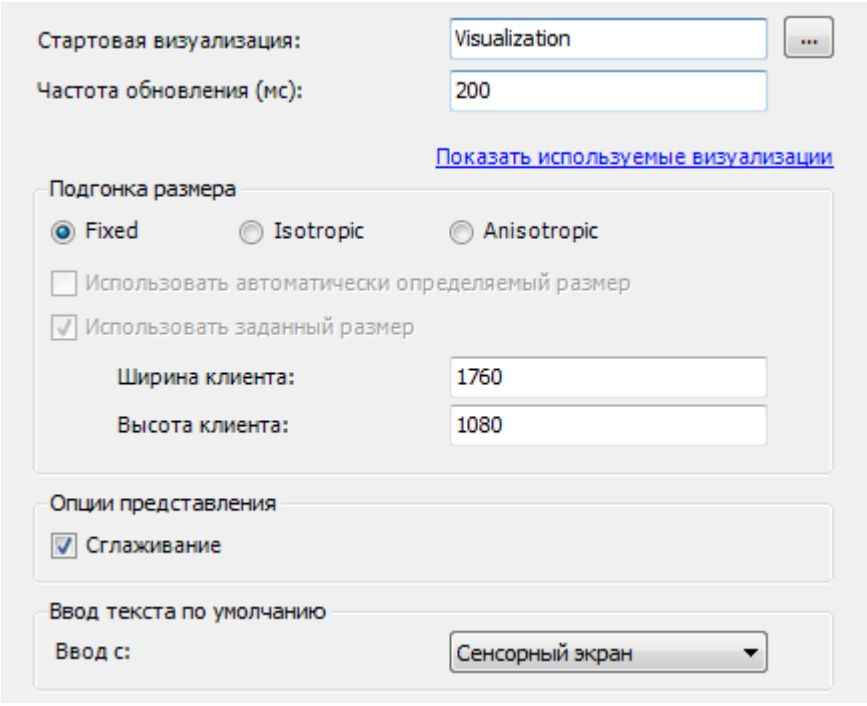


Рисунок 10.12 – Настройки таргет-визуализации

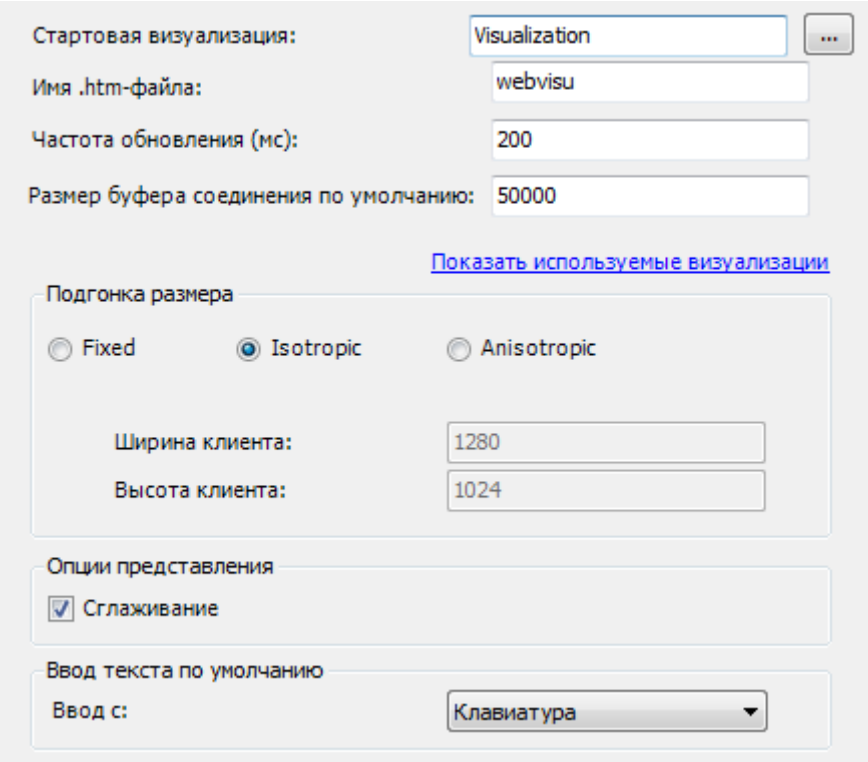


Рисунок 10.13 – Настройки web-визуализации

8. Добавить в проект Список текстов с названием **TextList**. Данный компонент потребуется для управления динамическими текстами прямоугольника:

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

Рисунок 10.14 – Содержимое списка текстов TextList

9. Приступить к наполнению экрана **Visualization**. Внешний вид готового экрана был показан на рисунке 10.3.

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

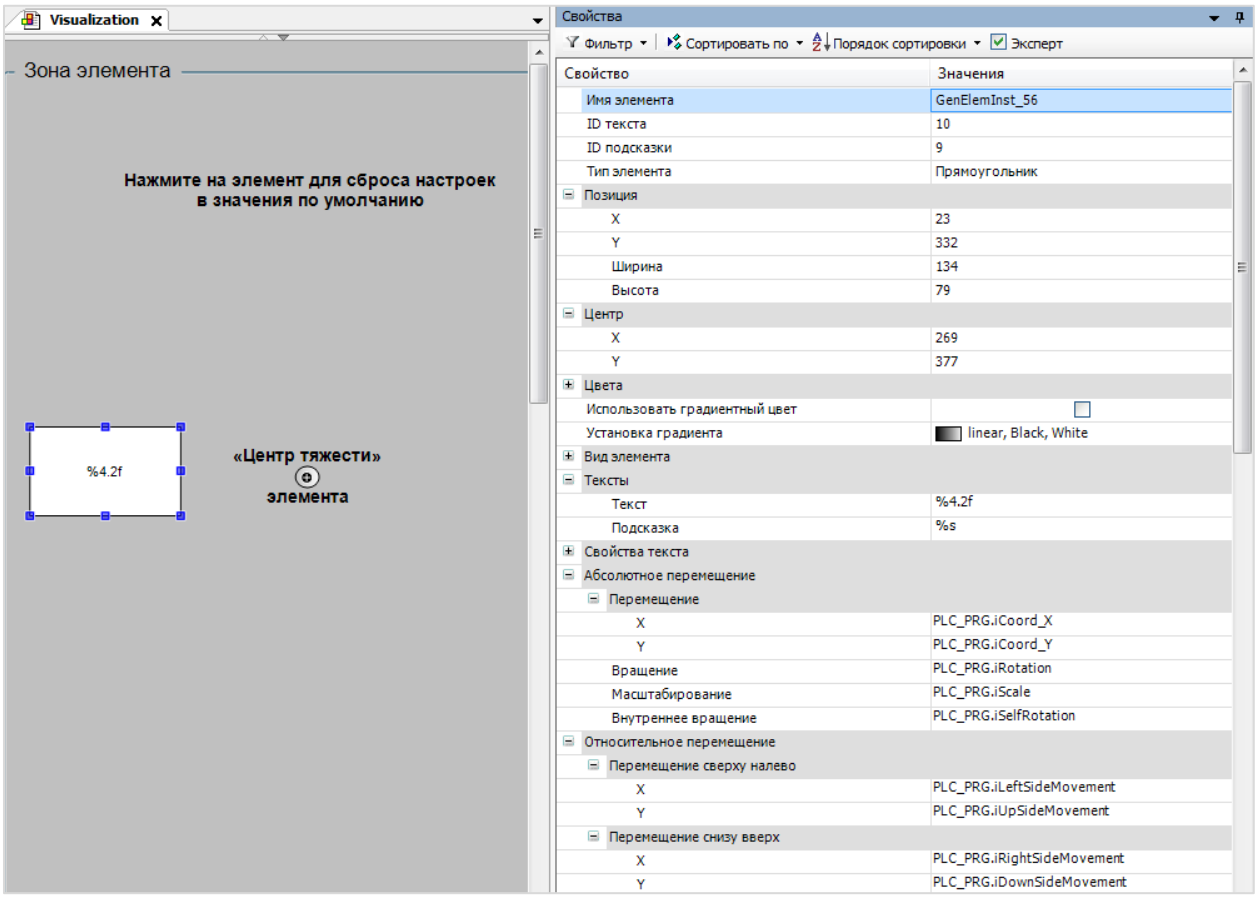


Рисунок 10.15 – Настройки элемента Прямоугольник (начало)

Свойства

Фильтр | Сортировать по | Порядок сортировки | Эксперт

Свойство	Значения
⊞ Относительное пере...	
⊟ Текстовые переменные	
Текстовая перем...	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...
⊟ OnMouseClick	Конфигурация...
Выполнить ST...	PLC_PRG.iCoord_X:=0;

Рисунок 10.16 – Настройки элемента Прямоугольник (окончание)

На центр элемента накладывается элемент **Эллипс** (чтобы можно было видеть, вокруг чего вращается элемент). Во вкладке **Абсолютное перемещение/Перемещение** привязываются те же переменные, что и у прямоугольника (чтобы в случае перемещения элемента, центр его тяжести также смещался).

⊟ Абсолютное перемещение	
⊟ Перемещение	
X	PLC_PRG.iCoord_X
Y	PLC_PRG.iCoord_Y

Рисунок 10.17 – Настройка параметра Абсолютное перемещение у элемента Прямоугольник

Текстовые надписи создаются с помощью элементов Метка. Границы области элемента (зона элемента) представляет собой элемент Группа.
Во вкладке InputConfiguration к параметру **OnMouseClick** привязывается **действие** Выполнить ST-код. ST-код содержит начальные значения переменных элемента. По нажатию на элемент, его параметры будут возвращены к этим значениям.

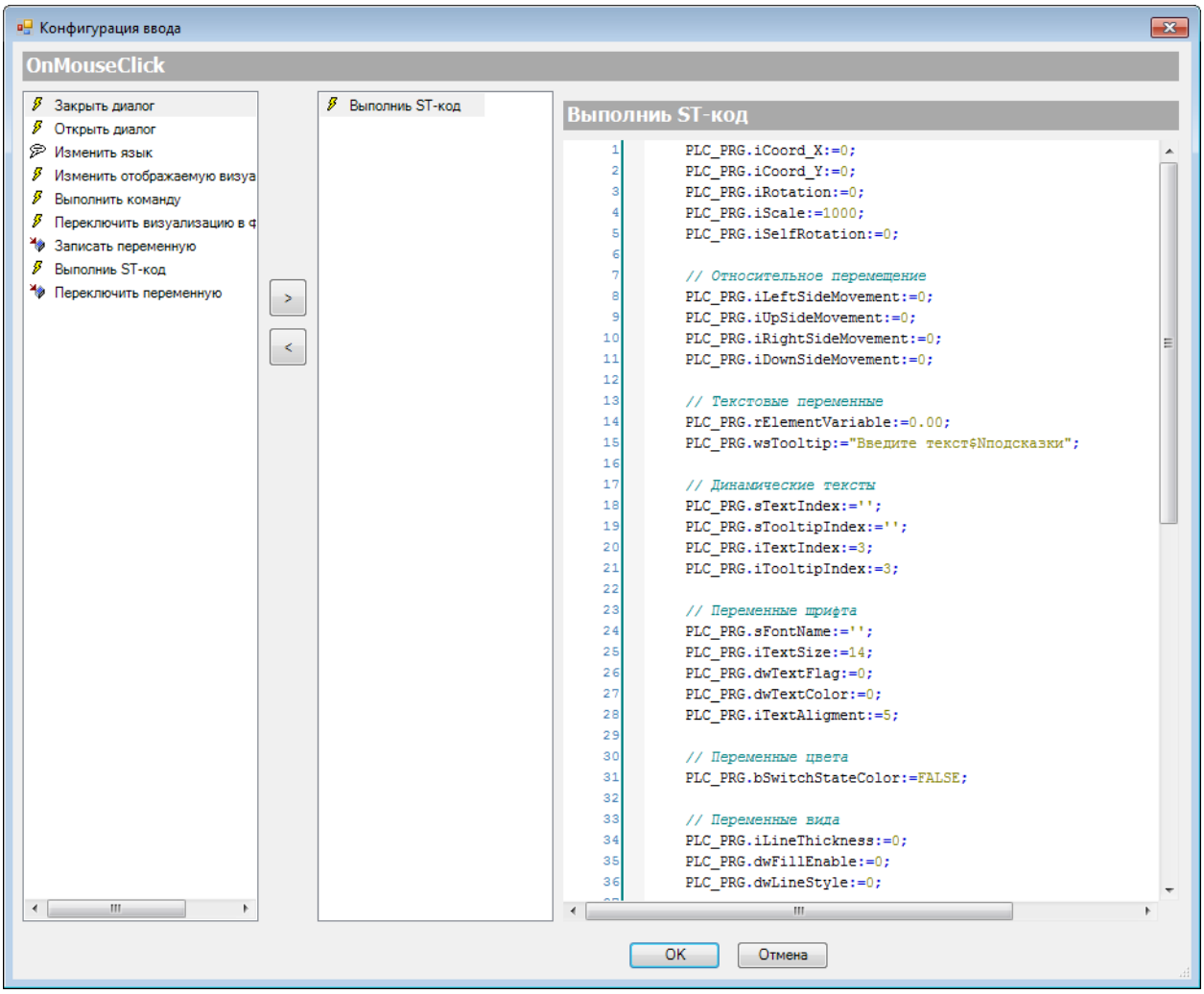


Рисунок 10.18 – Настройки параметра OnMouseClick вкладки InputConfiguration элемента Прямоугольник (начало)

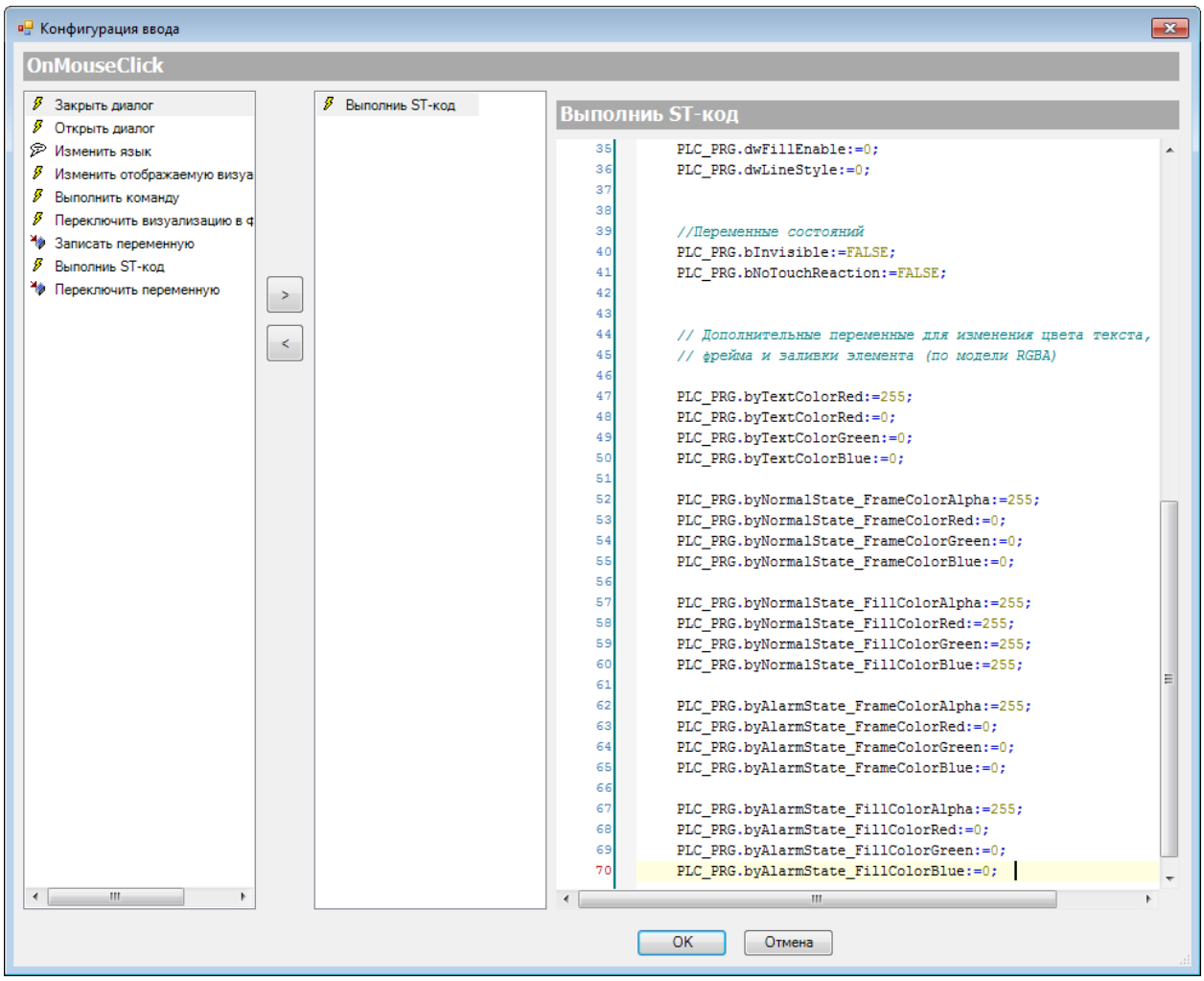


Рисунок 10.19 – Настройки параметра OnMouseClick вкладки InputConfiguration элемента Прямоугольник (окончание)

11. Добавить на экран панель управления параметрами вкладки **Абсолютное перемещение**. Данные параметры будут использоваться для перемещения и вращения элемента и изменения его масштаба.

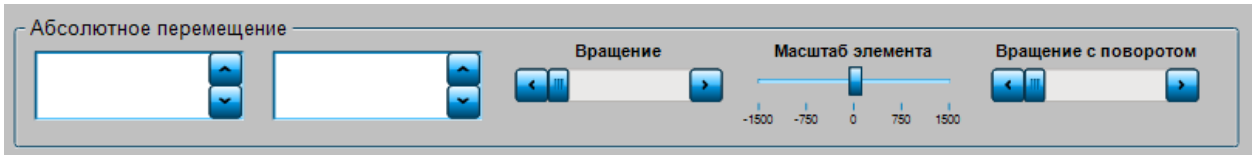


Рисунок 10.20 – Панель управления параметрами вкладки Абсолютное перемещение

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).

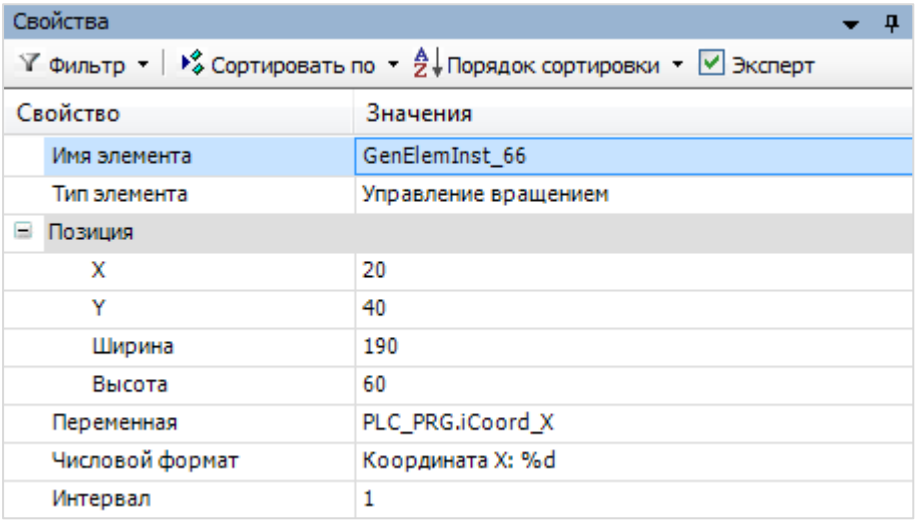


Рисунок 10.21 – Настройки элемента Управление вращением 1

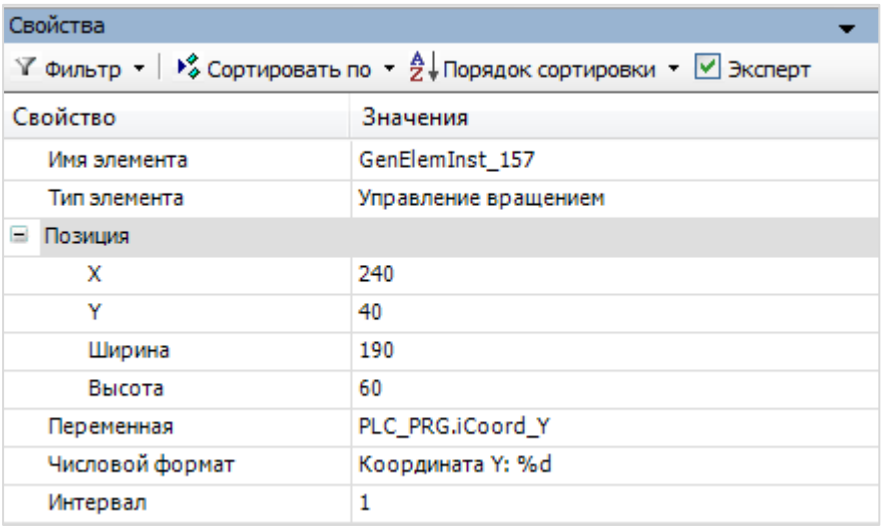


Рисунок 10.22 – Настройки элемента Управление вращением 2

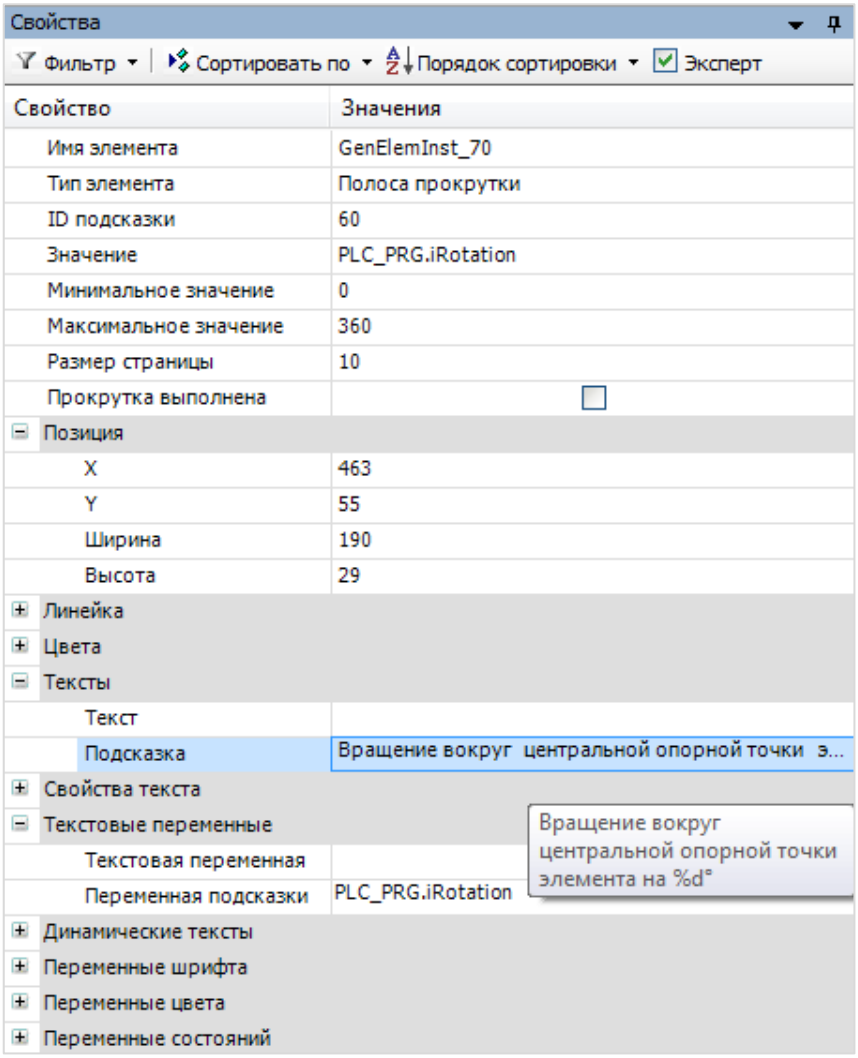


Рисунок 10.23 – Настройки элемента Полоса прокрутки 1

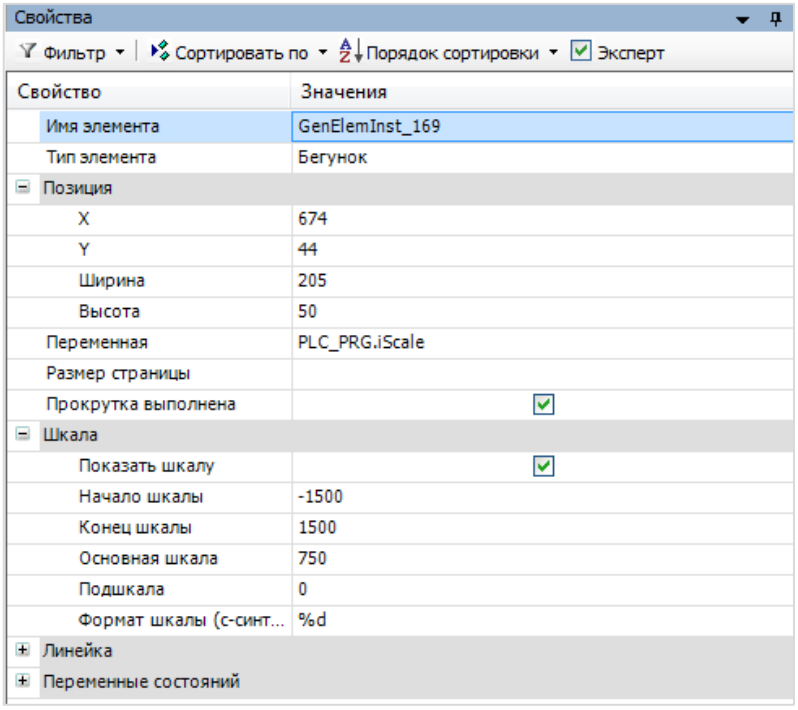


Рисунок 10.24 – Настройки элемента Бегунок

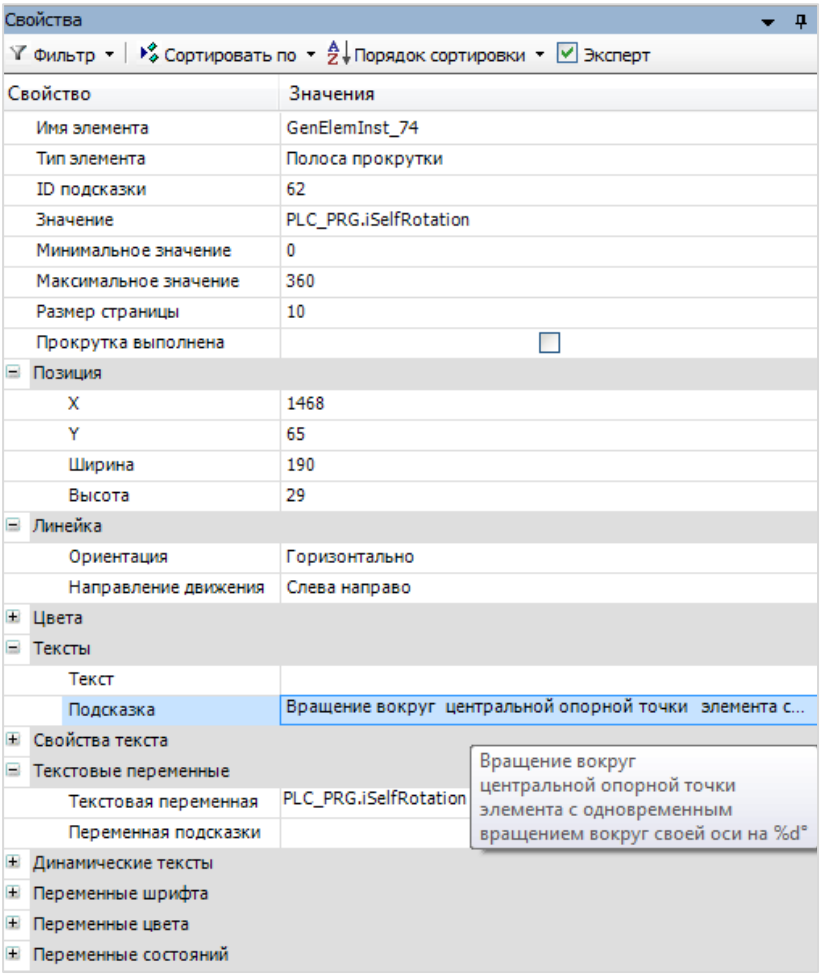


Рисунок 10.25 – Настройки элемента Полоса прокрутки 2

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

Рисунок 10.26 – 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

Рисунок 10.27 – 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

Рисунок 10.28 – 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

Рисунок 10.29 – 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

Рисунок 10.30 – Настройки элемента Управление вращением 4

13. Добавить на экран панель управления параметрами вкладки **Текстовые переменные**. Данные параметры будут отображаться элементом (**Текстовая переменная** – число с плавающей точкой, **Переменная подсказки** – строковая переменная).

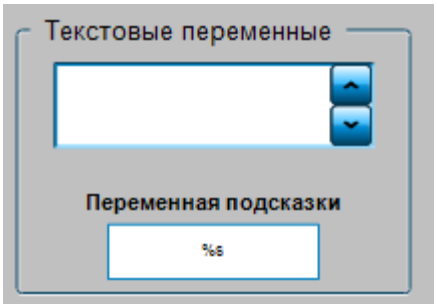


Рисунок 10.31 – Панель управления параметрами вкладки Текстовые переменные

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

Фильтр | Сортировать по | Порядок сортировки | Эксперт

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

Рисунок 10.32 – Настройки элемента Управление вращением

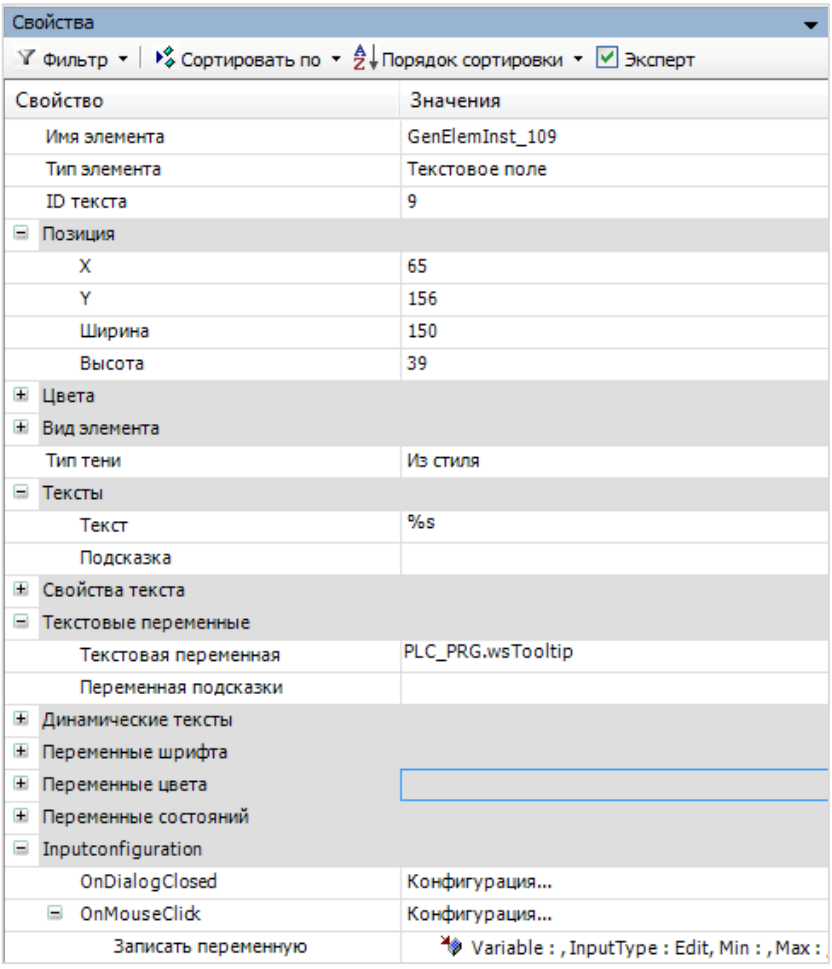


Рисунок 10.33 – Настройки элемента Текстовое поле

У элемента **Текстовое поле** во вкладке InputConfiguration к параметру **OnMouseClick** привяжем действие Записать переменную.

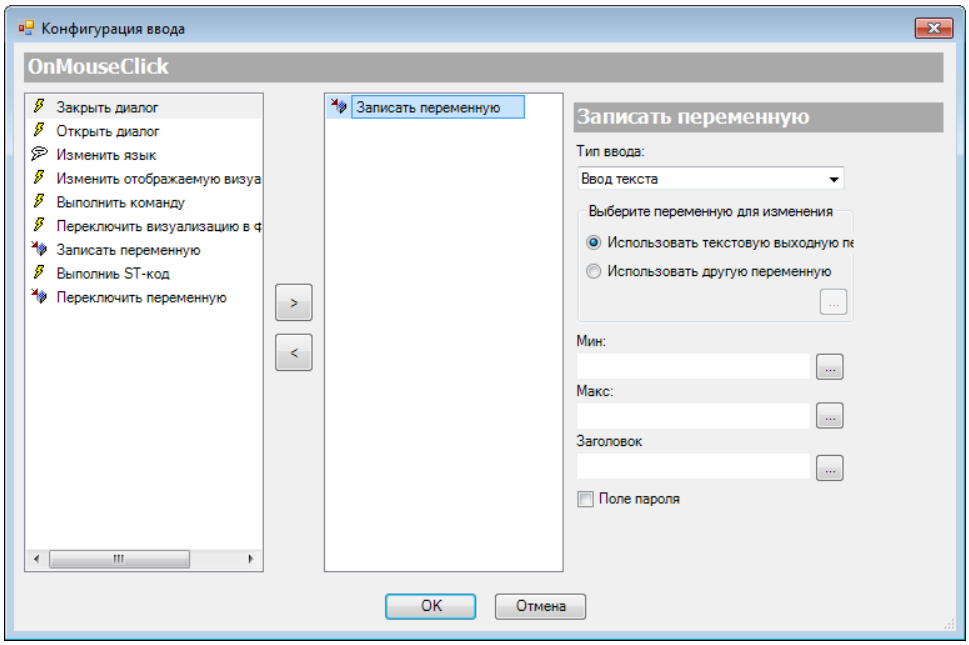


Рисунок 10.34 – Настройки действия элемента Текстовое поле

14. Добавить на экран панель управления параметрами вкладки **Динамические тексты**. Данные параметры позволяют выбирать тексты из Списка текстов, которые будут отображаться элементом и его подсказкой (вместо переменных из вкладки **Текстовые переменные**).

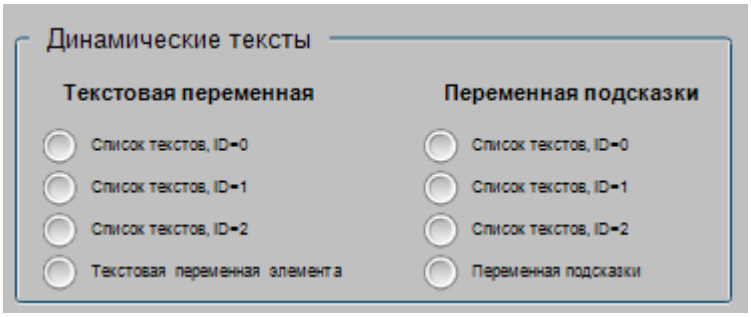


Рисунок 10.35 – Панель управления параметрами вкладки Динамические тексты

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. рисунок 10.8).

Данное преобразование возможно потому, что **ID** представляют собой числа (см. рисунок 10.14).

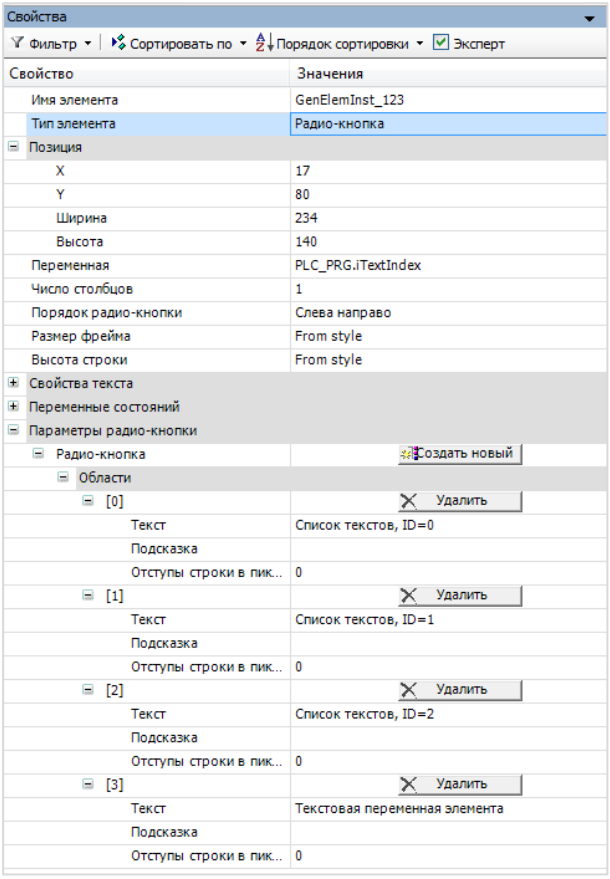


Рисунок 10.36 – Настройки элемента Радио-кнопка 1

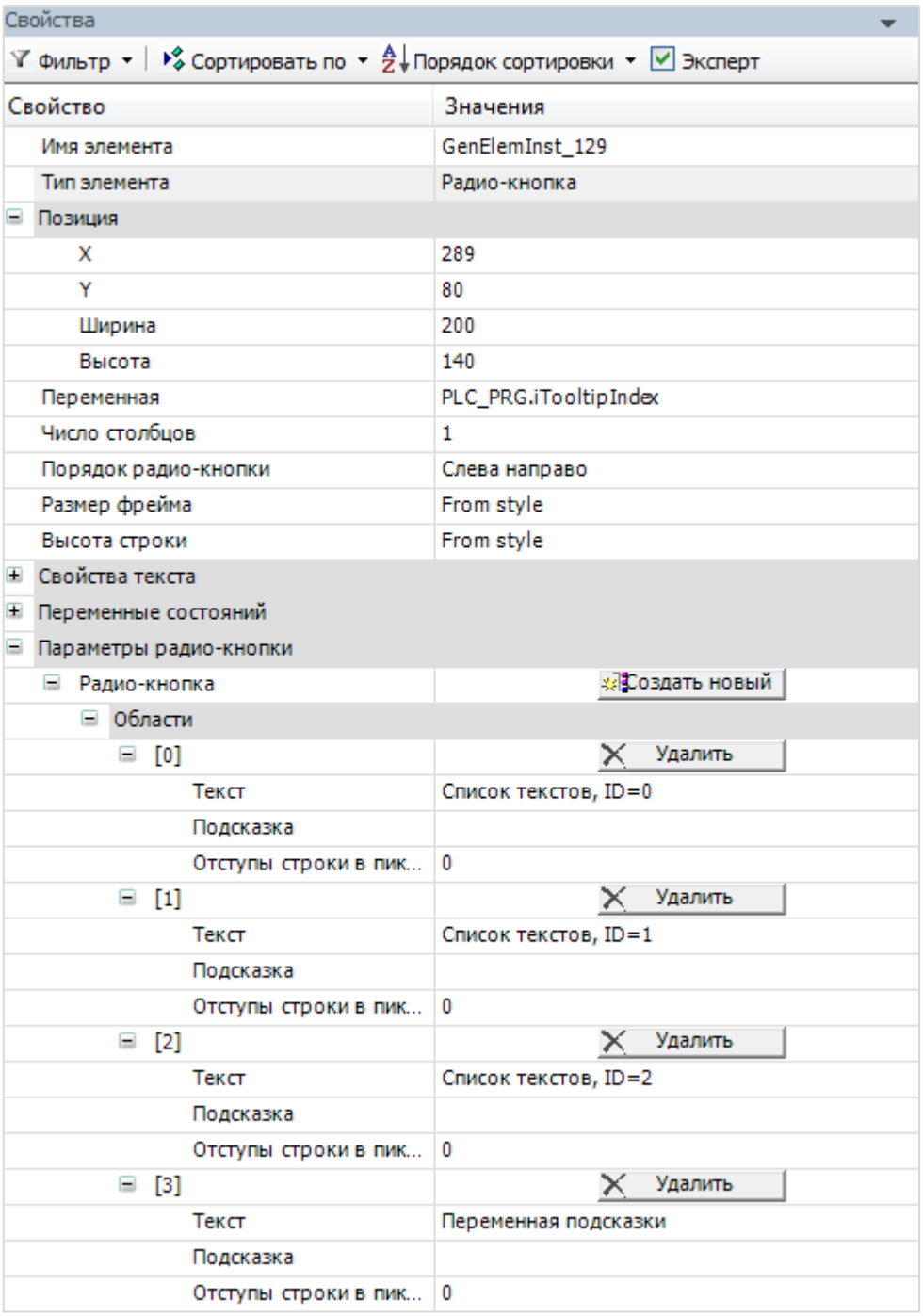


Рисунок 10.37 – Настройки элемента Радио-кнопка 2

15. Добавить 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.

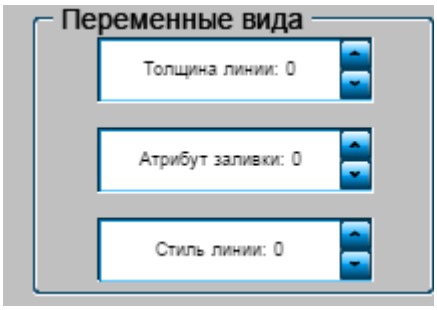


Рисунок 10.38 – Панель управления параметрами вкладки Переменные вида

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

Фильтр | Сортировать по | Порядок сортировки | Эксперт

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

Рисунок 10.39 – Настройки элемента Управление вращением 1

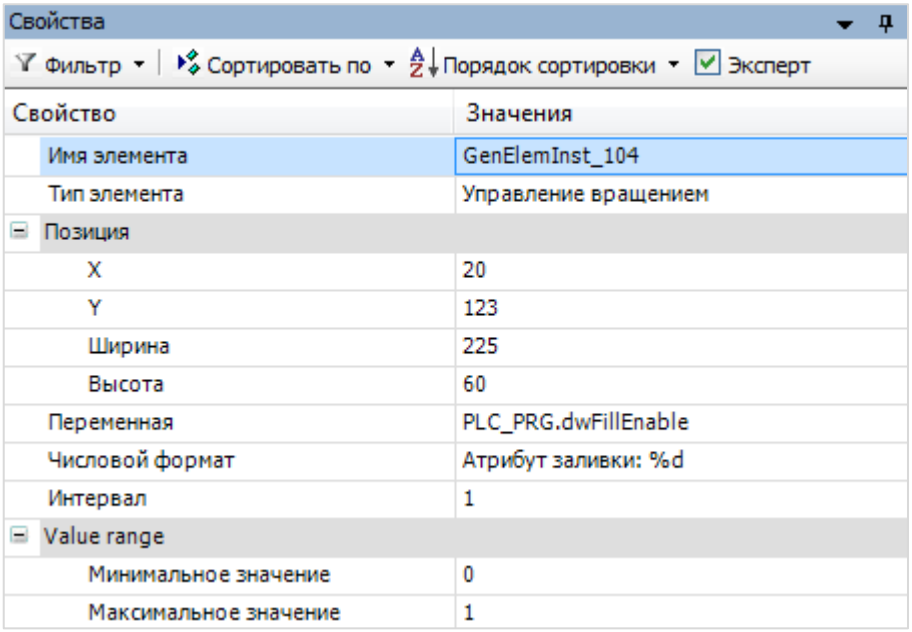


Рисунок 10.40 – Настройки элемента Управление вращением 2

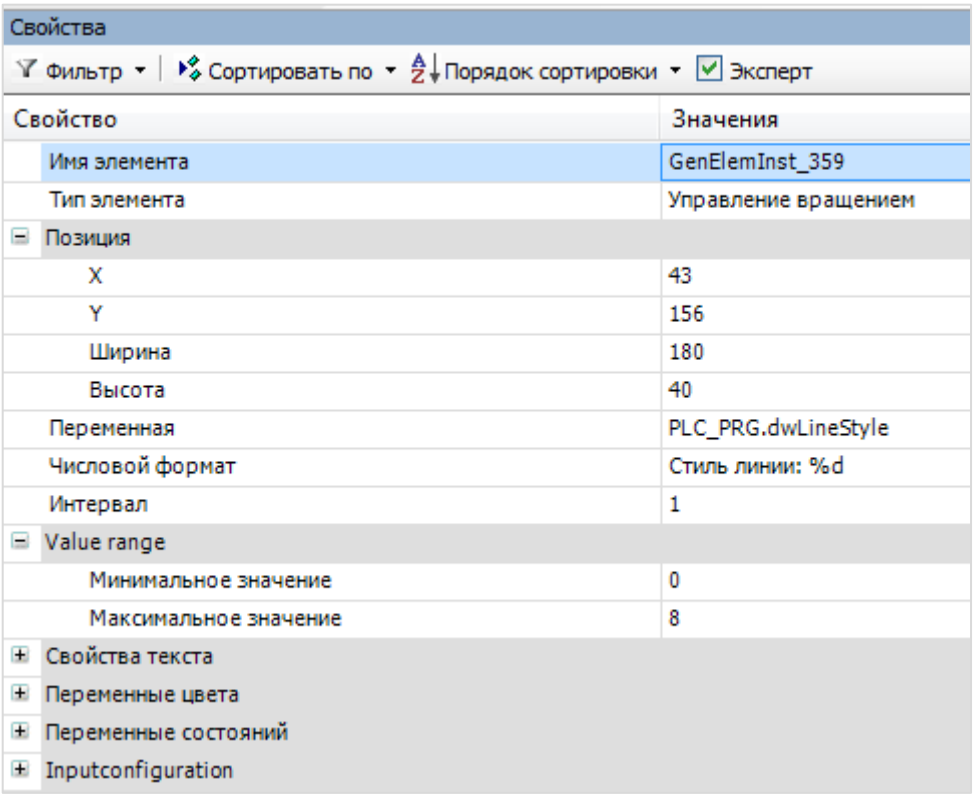


Рисунок 10.41 – Настройки элемента Управление вращением 3

16. Добавить на экран панель управления параметрами вкладки **Переменные шрифта**. Данные параметры позволяют настраивать шрифт, размер стиль и цвет текста элемента. Как упоминалось ранее (пп. 3), цвет будет собираться из значений задаваемых пользователем оттенков. Параметр **Набор символов** не работает в текущей (**3.5 SP11**) версии **CODESYS**, и поэтому в данном примере не рассматривается.

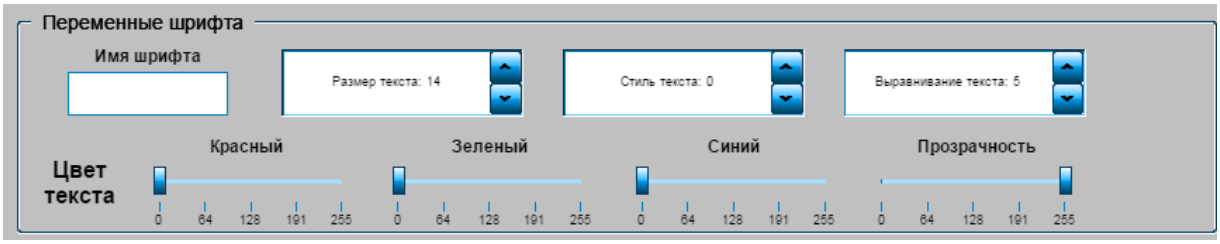


Рисунок 10.42 – Панель управления параметрами вкладки Переменные шрифта

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).

Свойства

Фильтр | Сортировать по | Порядок сортировки

Эксперт

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...
OnMouseClick	Конфигурация...
Записать переменную	Variable:, InputType:..

Рисунок 10.43 – Настройки элемента Текстовое поле

У элемента Текстовое поле во вкладке InputConfiguration к параметру **OnMouseClick** привязывается действие Записать переменную.

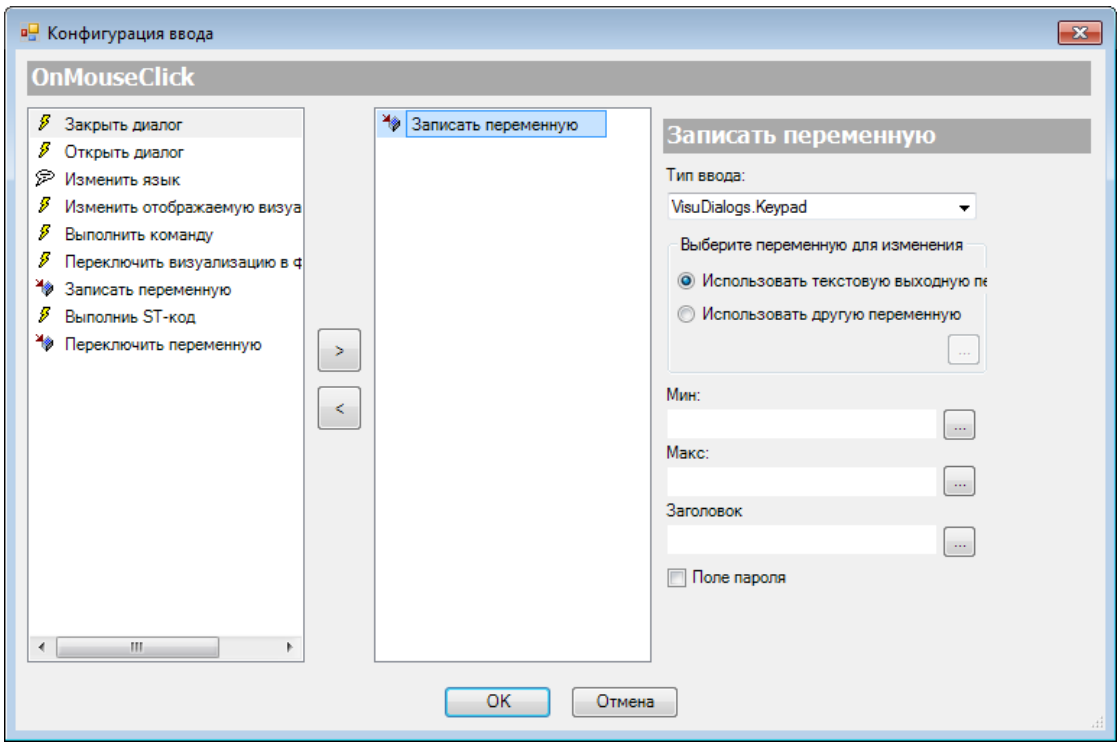


Рисунок 10.44 – Настройки действия элемента Текстовое поле

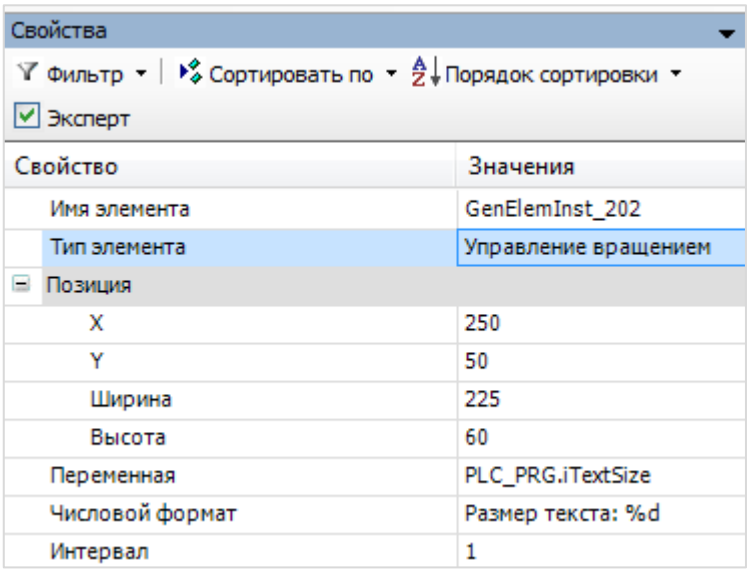
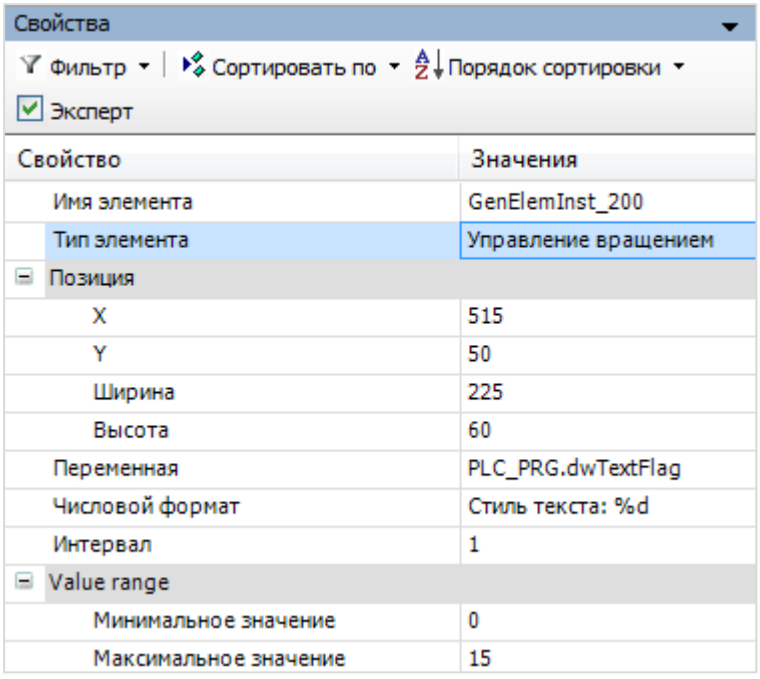
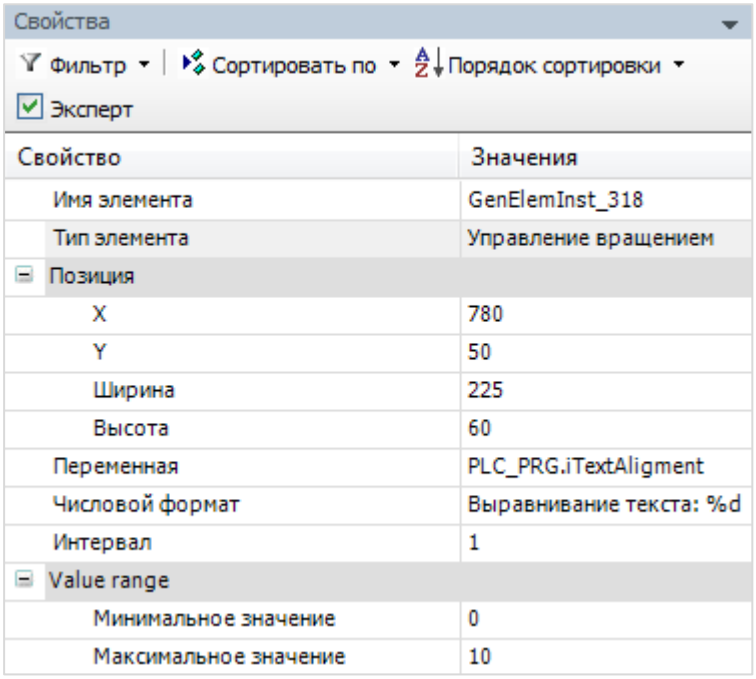


Рисунок 10.45 – Настройки элемента Управление вращением 1



Свойство	Значения
Имя элемента	GenElemInst_200
Тип элемента	Управление вращением
Позиция	
X	515
Y	50
Ширина	225
Высота	60
Переменная	PLC_PRG.dwTextFlag
Числовой формат	Стиль текста: %d
Интервал	1
Value range	
Минимальное значение	0
Максимальное значение	15

Рисунок 10.46 – Настройки элемента Управление вращением 2



Свойство	Значения
Имя элемента	GenElemInst_318
Тип элемента	Управление вращением
Позиция	
X	780
Y	50
Ширина	225
Высота	60
Переменная	PLC_PRG.iTextAligment
Числовой формат	Выравнивание текста: %d
Интервал	1
Value range	
Минимальное значение	0
Максимальное значение	10

Рисунок 10.47 – Настройки элемента Управление вращением 3

Свойства

Фильтр | Сортировать по | Порядок сортировки

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	☑
Шкала	
Показать шкалу	☑
Начало шкалы	0
Конец шкалы	255
Основная шкала	63
Подшкала	0
Формат шкалы (c-синтаксис)	%d

Рисунок 10.48 – Настройки элемента Бегунок 1

Свойства

Фильтр | Сортировать по | Порядок сортировки

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	☑
Шкала	
Показать шкалу	☑
Начало шкалы	0
Конец шкалы	255
Основная шкала	63
Подшкала	0
Формат шкалы (c-синтаксис)	%d

Рисунок 10.49 – Настройки элемента Бегунок 2

Свойства

Фильтр | Сортировать по | Порядок сортировки

Эксперт

Свойство	Значения
Имя элемента	GenElemInst_196
Тип элемента	Бегунок
Позиция	
X	575
Y	150
Ширина	200
Высота	50
Переменная	PLC_PRG.byTextColorBlue
Размер страницы	
Прокрутка выполнена	☑
Шкала	
Показать шкалу	☑
Начало шкалы	0
Конец шкалы	255
Основная шкала	63
Подшкала	0
Формат шкалы (c-синтаксис)	%d

Рисунок 10.50 – Настройки элемента Бегунок 3

17. Добавить на экран панель управления параметрами вкладки **Переменные цвета**. Данные 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.

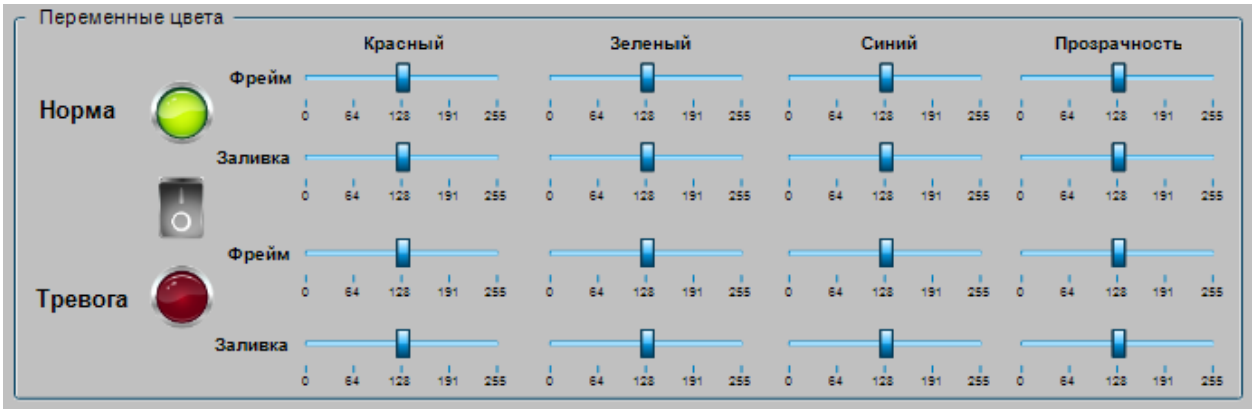
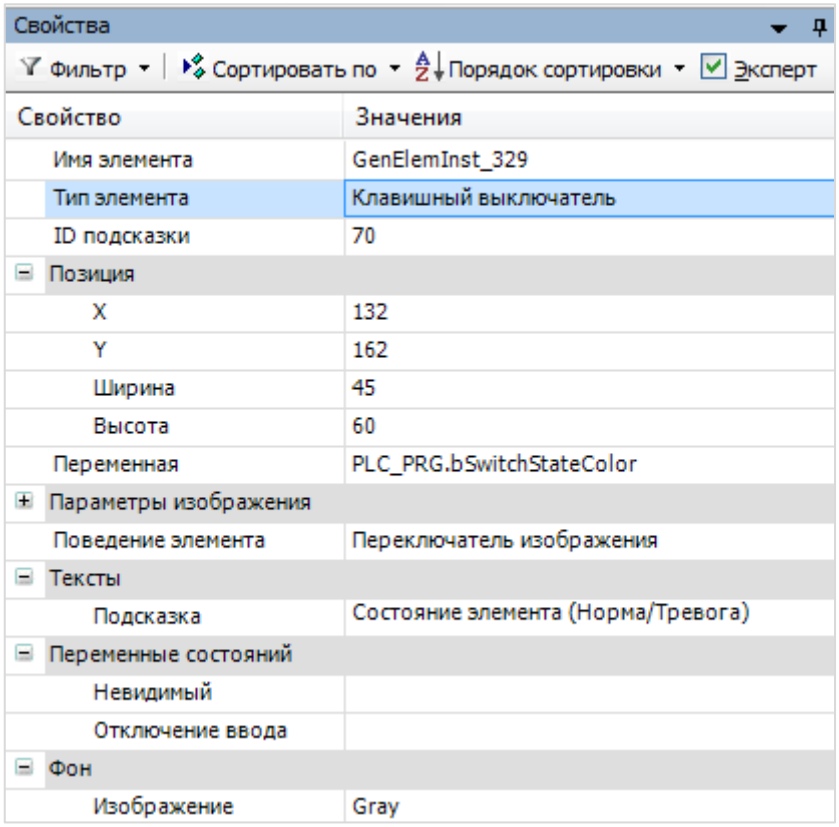


Рисунок 10.51 – Панель управления параметрами вкладки Переменные цвета

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

Рисунок 10.52 – Настройки элемента Клавишный выключатель

Элемент Клавишный переключатель будет определять состояние цвета элемента (**TRUE** – тревога, **FALSE** – норма). Для визуализации состояния необходимы два индикатора – зеленый (**FALSE**, норма) и красный (**TRUE**, тревога). Но элемент Индикатор в состоянии **FALSE** всегда отображается «потухшим». Поэтому вместо индикатора, для состояния «Норма» используется элемент Переключатель изображения, в котором приводятся ссылки на картинки индикатора из пула изображений стиля визуализации:

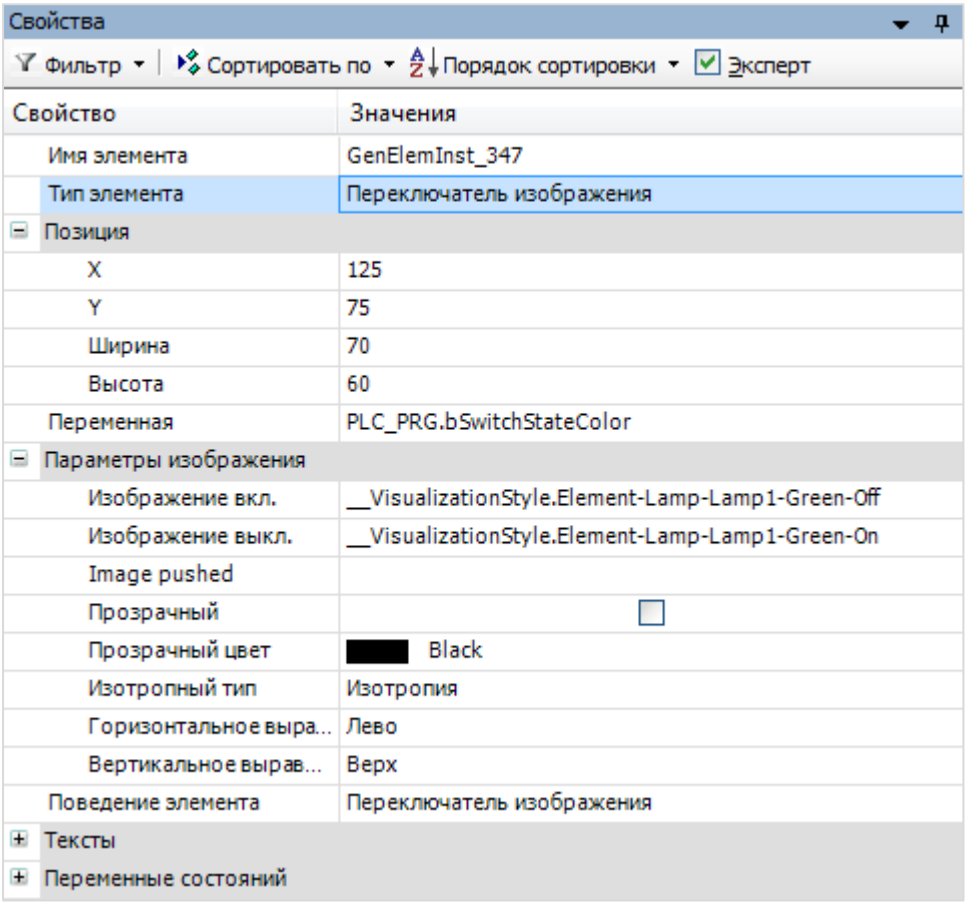


Рисунок 10.53 – Настройки элемента Переключатель изображения («Норма»)

Индикатор «Тревога» будет представлять собой стандартный элемент Индикатор:

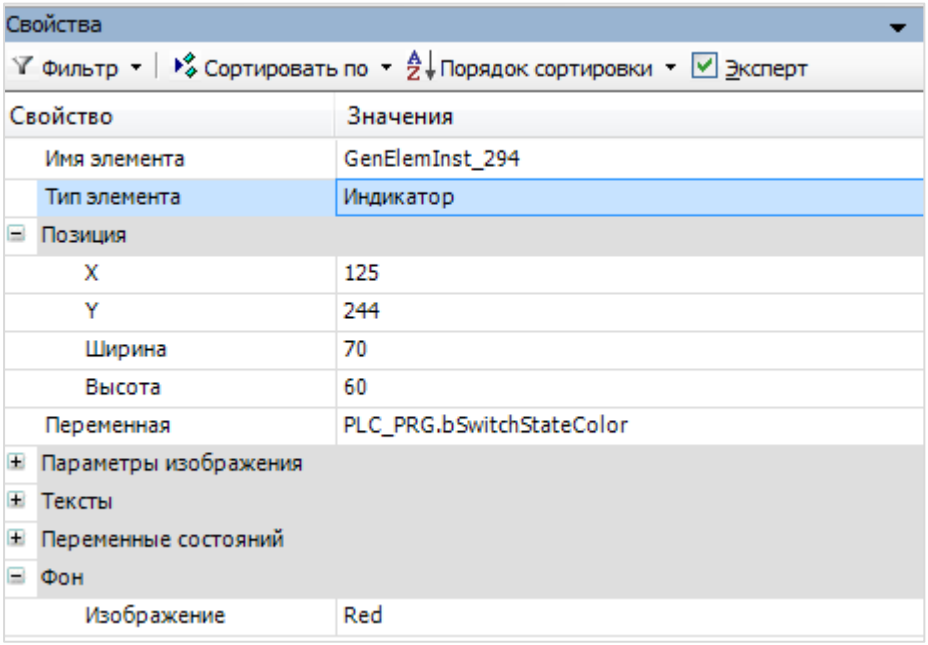


Рисунок 10.54 – Настройки элемента Индикатор («Тревога»)

Элементы Бегунок отличаются только привязываемыми к ним переменными (см. таблицу 10.1):

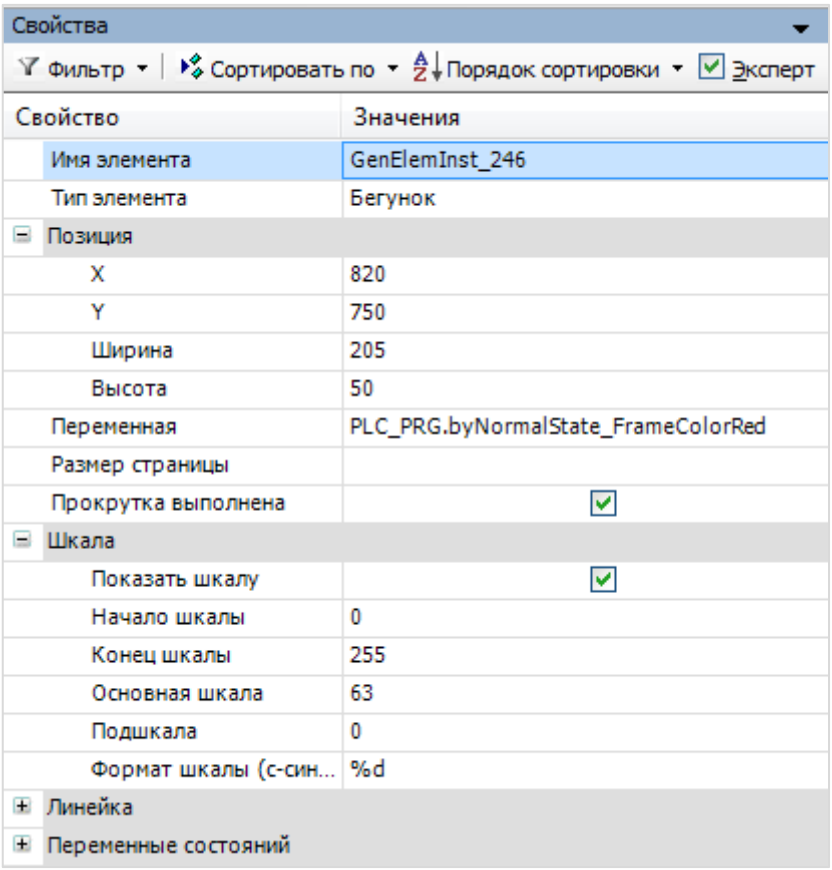


Рисунок 10.55 – Настройки элемента Бегунок

Таблица 10.1 – 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

18. Добавить на экран панель управления параметрами вкладки **Переменные состояний**. Данные параметры позволяют управлять видимостью и активностью (реакцией на касание) элемента.

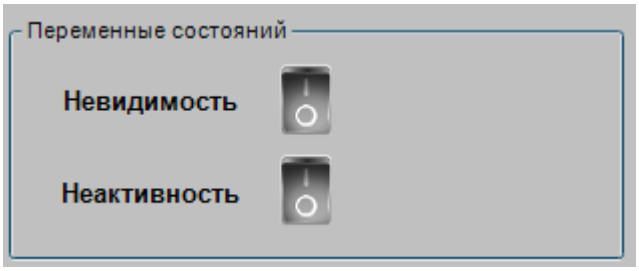


Рисунок 10.56 – Панель управления параметрами вкладки Переменные состояний

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).

Свойства

Фильтр | Сортировать по | Порядок сортировки | Эксперт

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

Рисунок 10.57 – 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

Рисунок 10.58 – Клавишный выключатель 2

19. Запустить проект на виртуальном контроллере.
Открыть в веб-браузере страницу http://localhost:8080/webvisu.htm, чтобы увидеть web-визуализацию проекта. Браузер должен поддерживать **HTML5**.
Затем проверить функционал проекта.

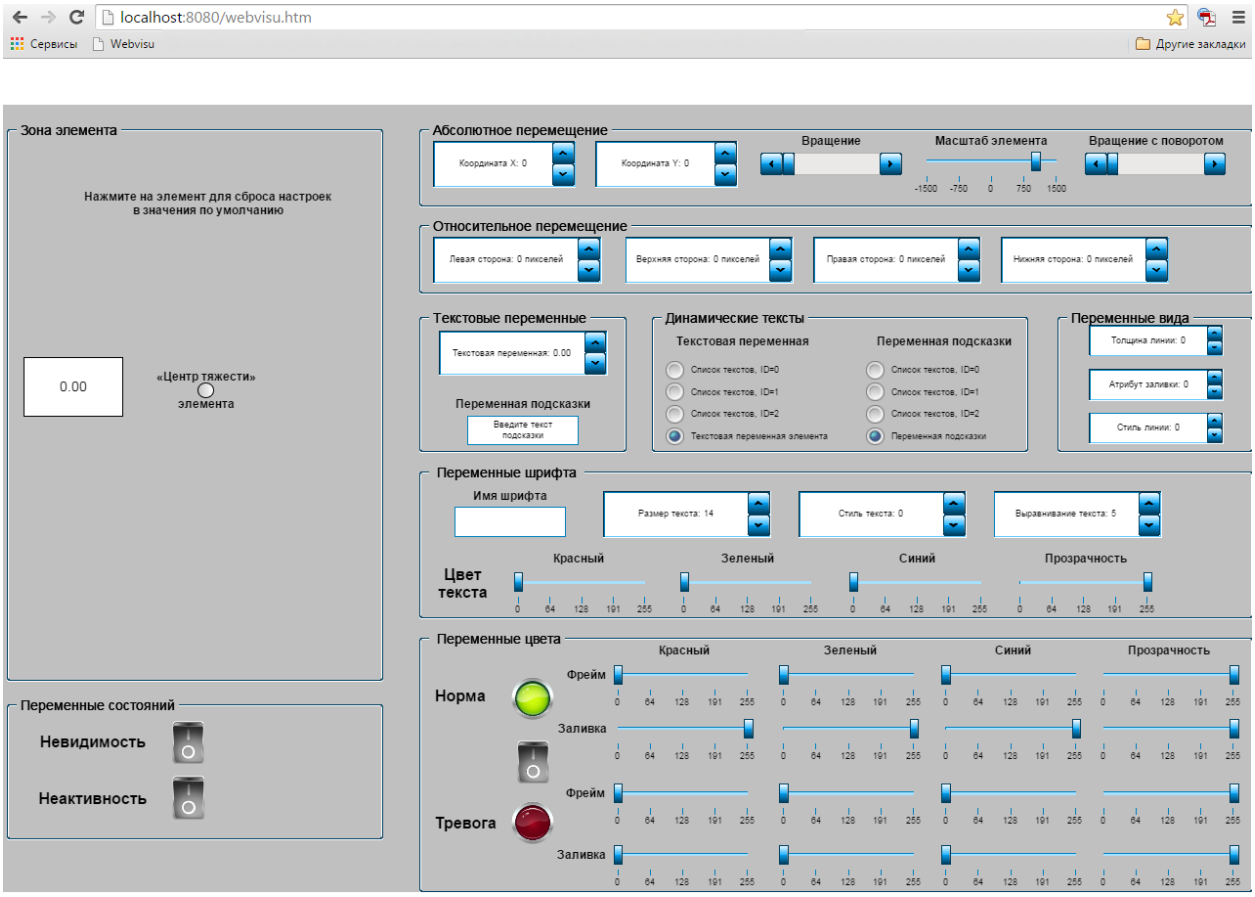


Рисунок 10.59 – Web-визуализация проекта в браузере Chrome

20. Проверить 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, одновременно элемент поворачивается вокруг своего геометрического центра.

21. Проверить функционал вкладки **Относительное перемещение**:
- изменение значений приводит к соответствующему изменению размеров элемента.

22. Проверить функционал вкладки **Текстовые переменные**:
- изменение текстовой переменной приводит к изменению значения, отображаемого элементом;
- после ввода подсказки переменная будет отображена элементом.

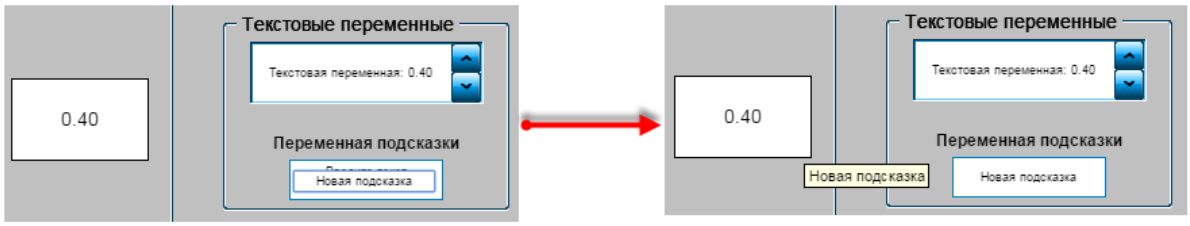


Рисунок 10.60 – Ввод и отображение подсказки элемента

23. Проверить функционал вкладки **Динамические тексты**:
- с помощью радиокнопок можно менять отображаемое значение и подсказку элемента на тексты из Списка текстов.

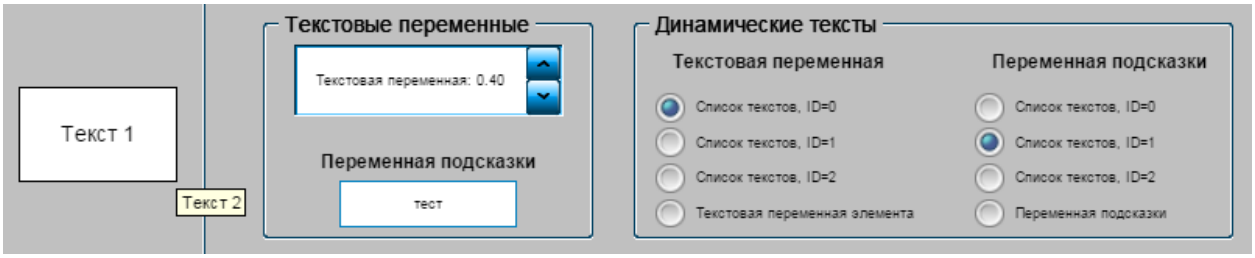


Рисунок 10.61 – Использование динамических текстов

23. Проверить 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 работает корректно.

Рисунок 10.62 – Изменение стиля линий

24. Проверить функционал вкладки **Переменные шрифта**:

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

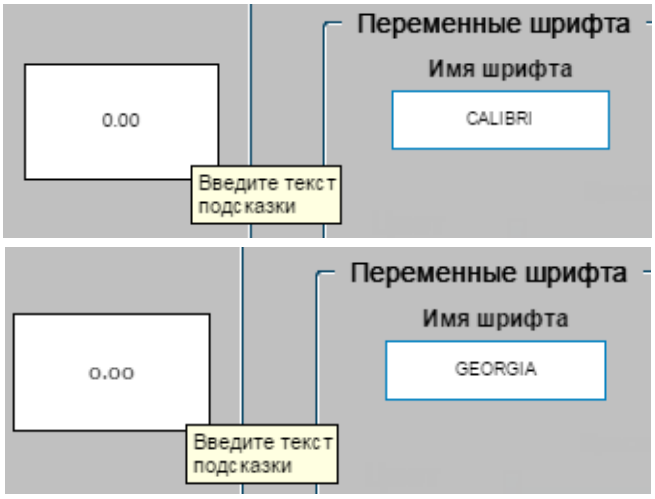


Рисунок 10.63 – Изменение шрифта текста элемента

Изменение размера текста работает корректно;

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

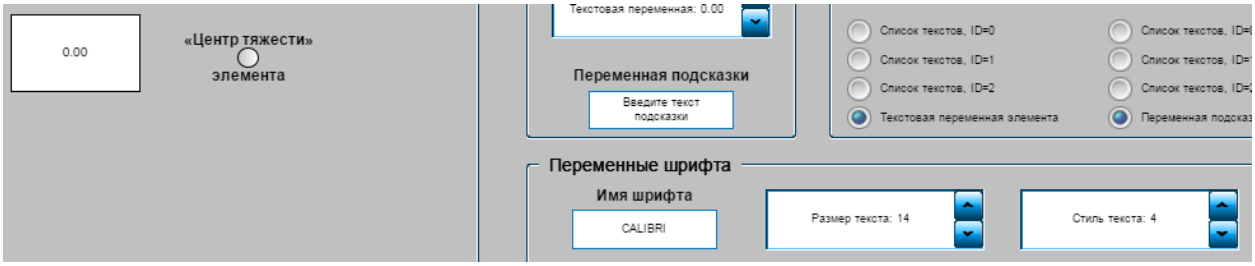


Рисунок 10.64 – Стиль текста в web-визуализации

Рисунок 10.65 – Стиль текста в визуализации редактора CODESYS

Изменение выравнивания текста работает корректно;

Изменение цвета текста работает корректно.

25. Проверить функционал вкладки **Переменные цвета**:

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

26. Проверить 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.

10.2.2 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. 10.3.3.

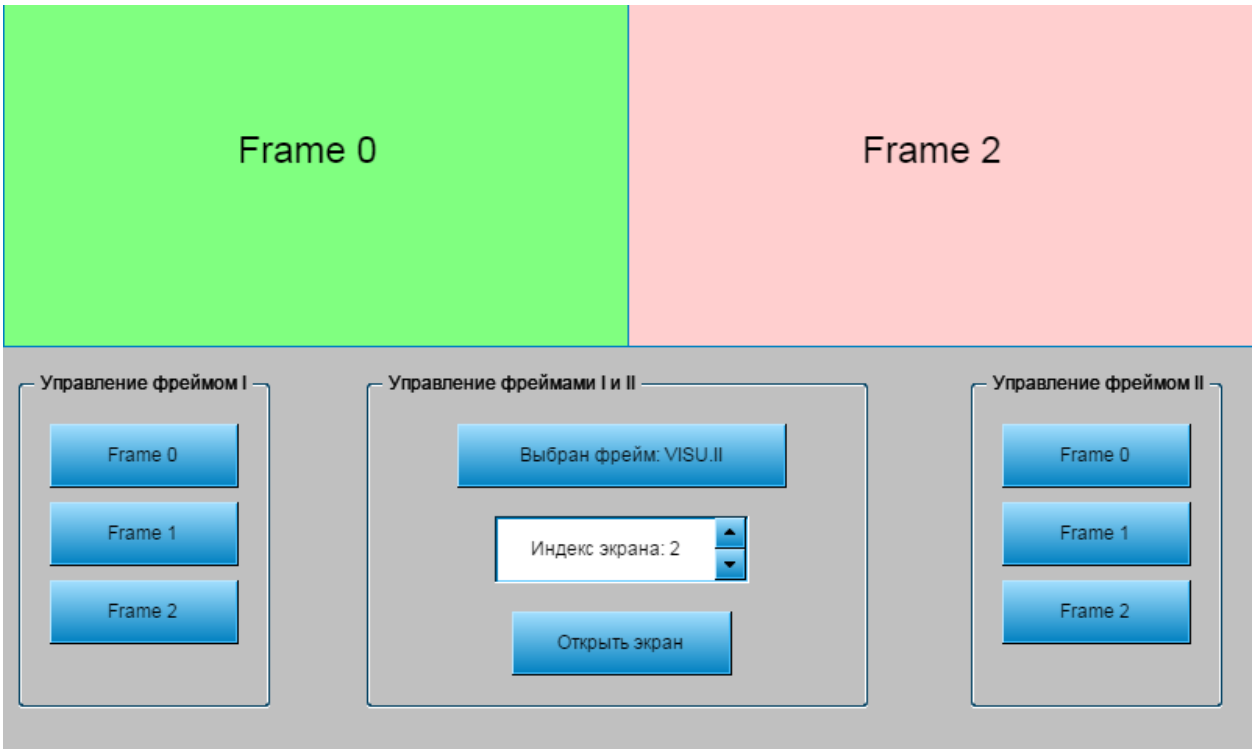


Рисунок 10.66 – Внешний вид примера Фрейм

Пример создан в среде **CODESYS V3.5 SP11 Patch 5** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с таргет-файлом **3.5.11.5**.

Пример доступен для скачивания: Example_Frame.projectarchive

Продвинутый пример от компании **CODESYS Group** по работе с фреймами в коде программы:
VisualizationSwitching.package
Файлы формата **.package** устанавливаются с помощью **Менеджера пакетов** (вкладка **Инструменты**) или (начиная с версии **CODESYS V3.5 SP17**) с помощью утилиты **CODESYS Installer**.

Для создания примера с использованием элемента **Фрейм** следует:

1. Создать новый ***стандартный*** проект **CODESYS** с названием **Example_Frame** и настройками по умолчанию: таргет – CODESYS Control Win V3, язык PLC_PRG – **ST**.
2. В программе **PLC_PRG** объявить следующие переменные:

```
PLC_PRG
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
```

Рисунок 10.67 – Объявление переменных программы PLC_PRG

3. Добавить в проект экран визуализации с названием **Frame_0**. В его свойствах выбрать размер **400 × 220.** Данный экран будет одним из экранов, отображаемых элементом Фрейм. Затем следует добавить на него метку с названием экрана и выбрать фон (с помощью вкладки Фон контекстного меню, открываемого по нажатию **ПКМ** на рабочую область).

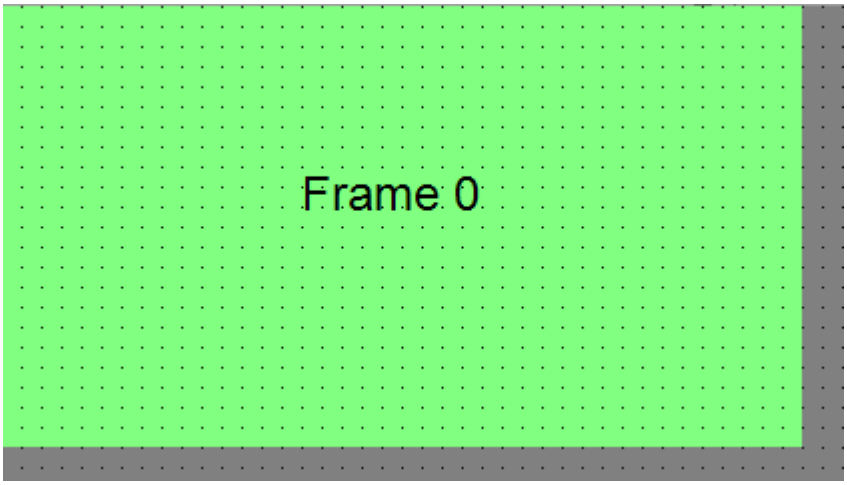


Рисунок 10.68 – Экран визуализации Frame_0

4. Добавить 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. Добавить в проект экран визуализации с названием **Visu.** В его свойствах следует выбрать размер **800 × 480**. Данный экран будет основным экраном проекта.

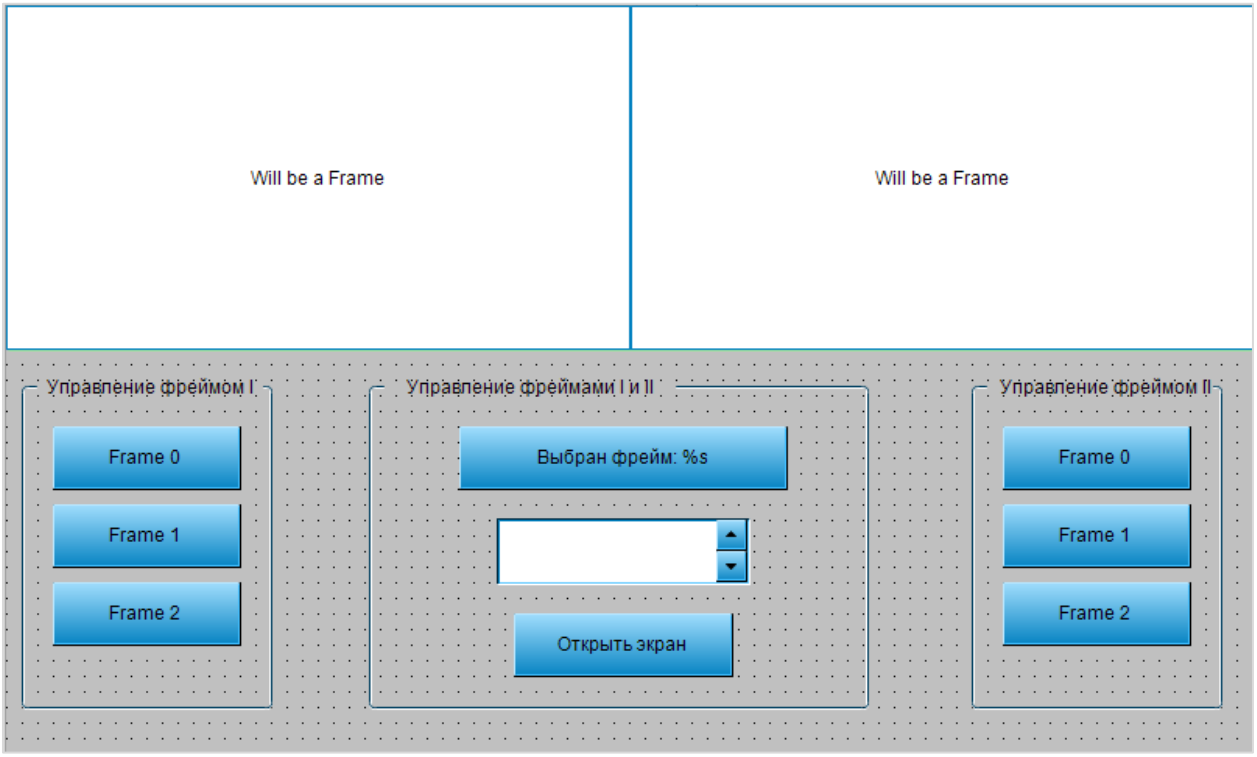


Рисунок 10.69 – Экран визуализации Visu (фреймы без конфигурации)

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.

6. Настроить 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 следует нажать кнопку **Конфигурация**:

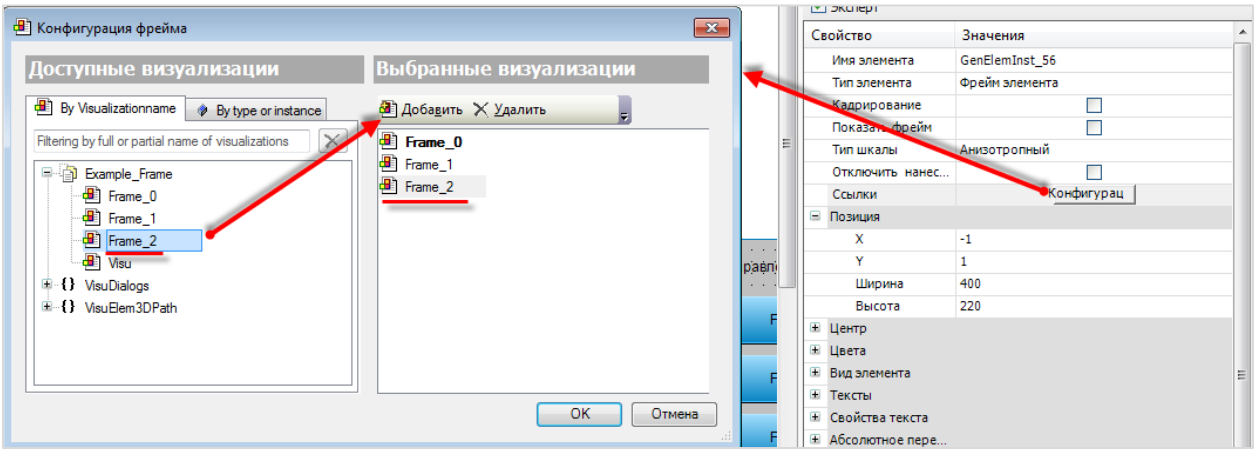


Рисунок 10.70 – Конфигурация фрейма

Затем следует привязать 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.

Фрейм настраивается следующим образом:

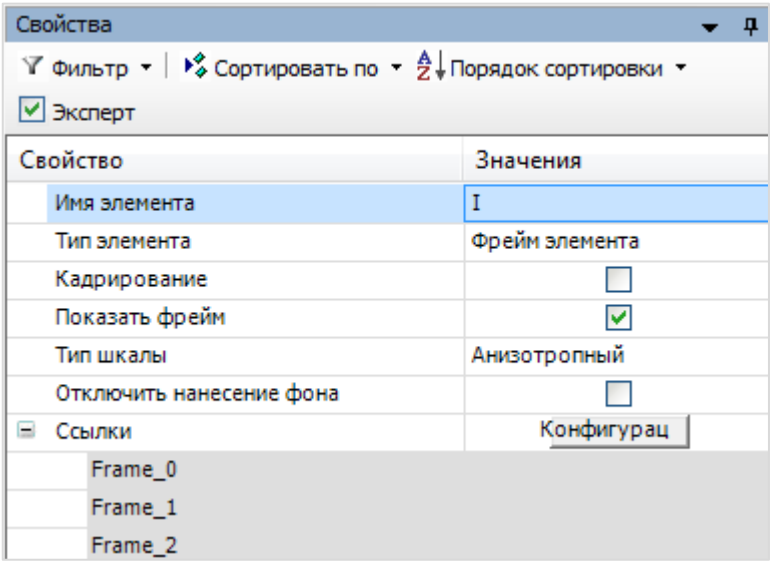


Рисунок 10.71 – Параметры элемента Фрейм

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

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

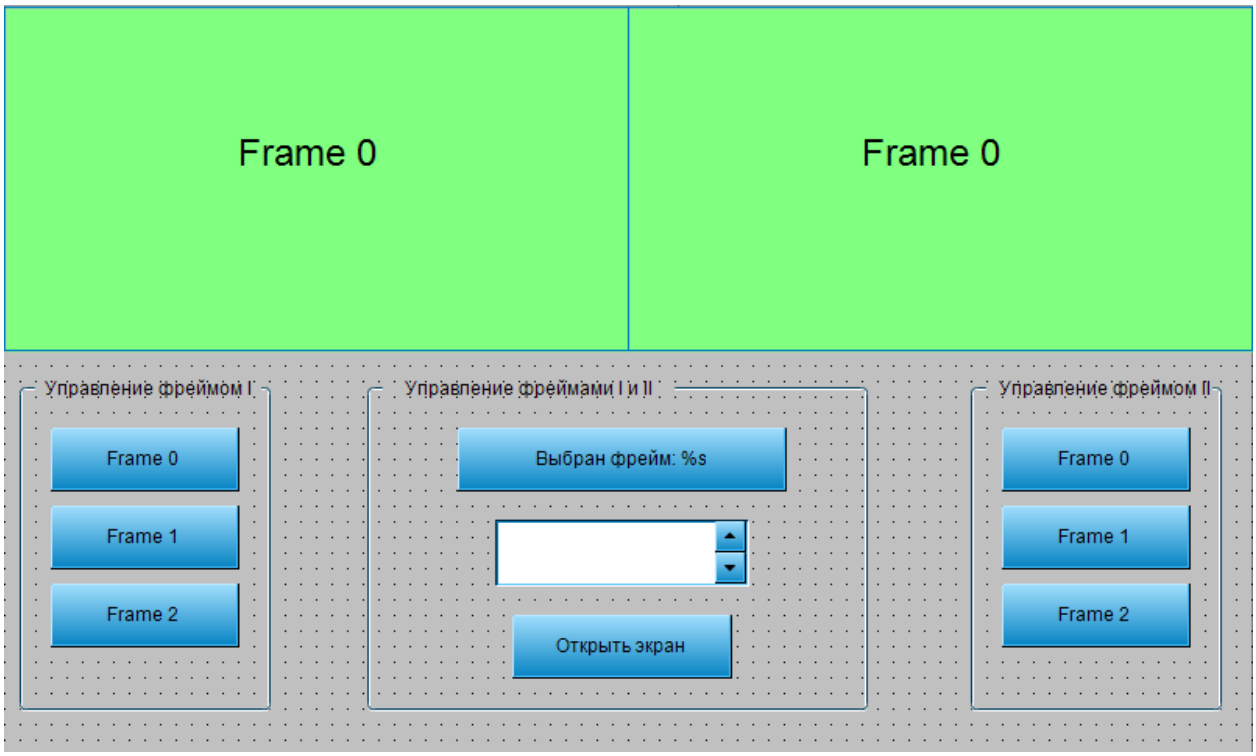


Рисунок 10.72 – Экран визуализации Visu (фреймы сконфигурированы)

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

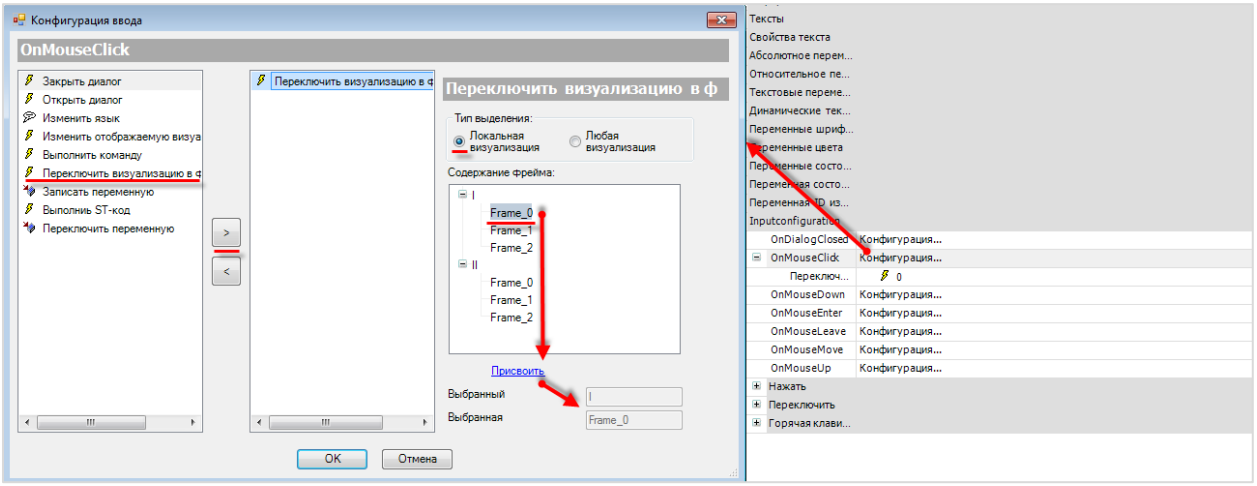


Рисунок 10.73 – Настройки действия кнопки Frame_0

Во вкладке InputConfiguration к параметру **OnMouseClick** следует привязать действие Переключить визуализацию во фрейме. Затем следует выбрать тип выделения **Локальная визуализация** и присвоить 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 настраиваются все кнопки панелей **Управление фреймом I** и **Управление фреймом II**. Фреймы будут отличаться только привязанными экранами (см. рисунок 10.73).

9. Настроить панель **Управление фреймом I и II**. С помощью этой панели можно будет переключать экраны в любом из фреймов, предварительно введя его название. Настройки кнопки **Выбран фрейм**:

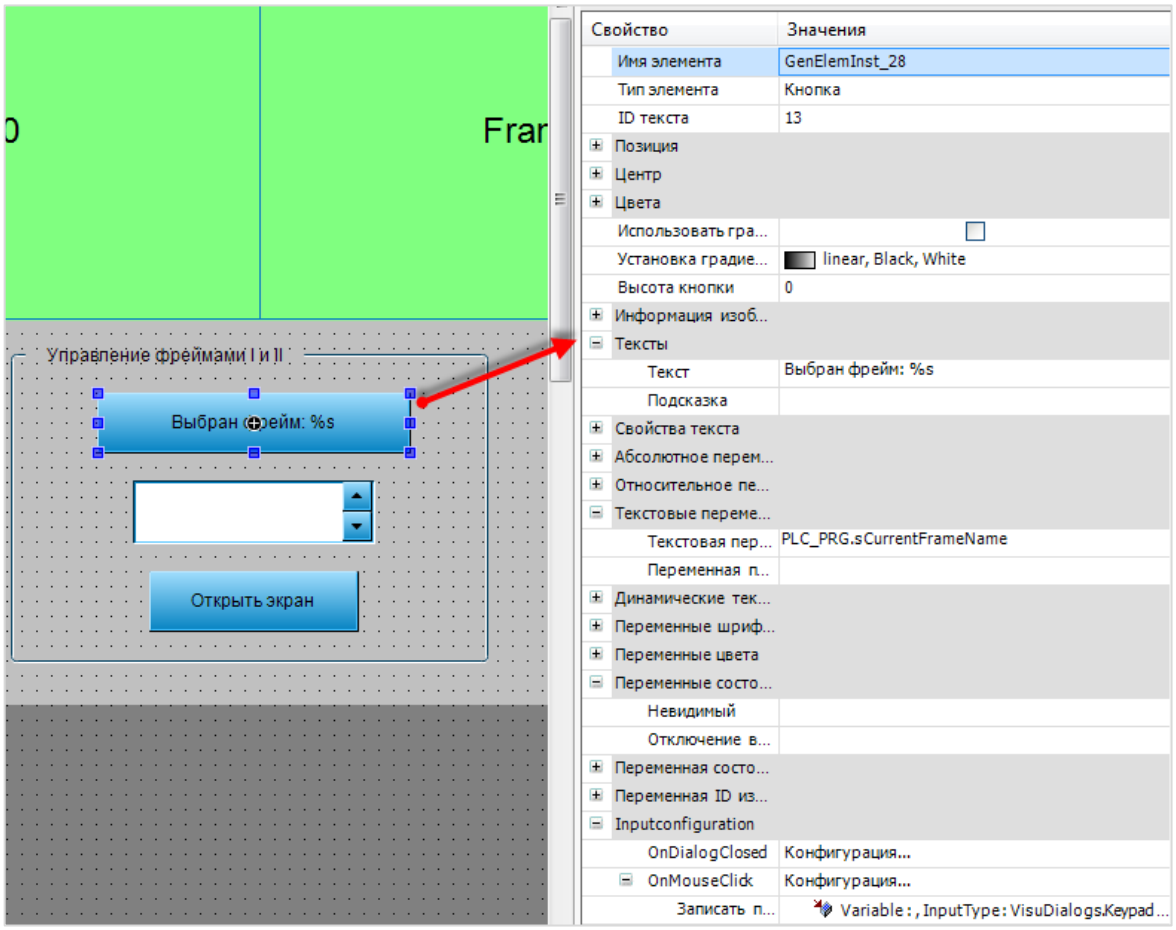


Рисунок 10.74 – Настройки кнопки Выбран фрейм

Во вкладке InputConfiguration к параметру **OnMouseClick** привязывается действие Записать переменную:

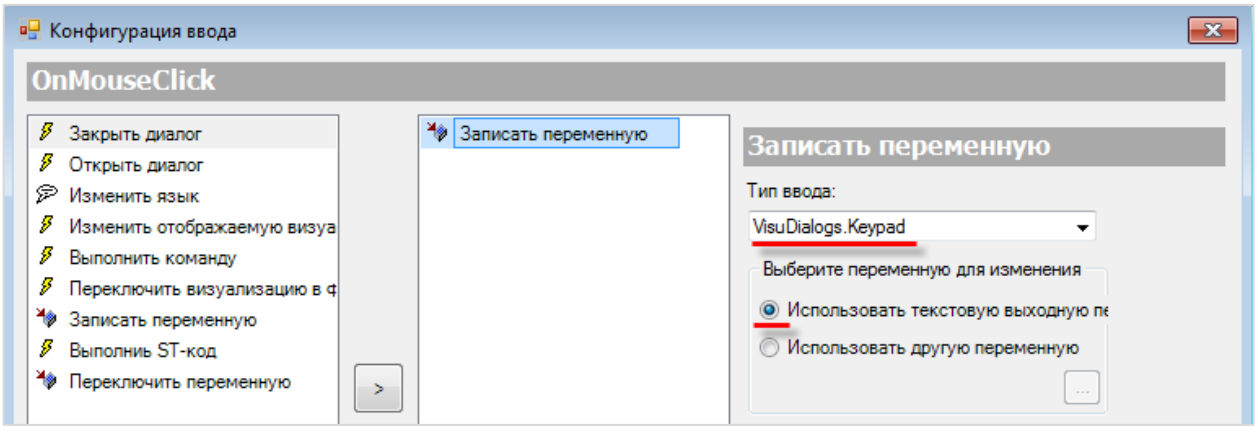


Рисунок 10.75 – Настройки действия кнопки Выбран фрейм

Затем настраивается элемент Управление вращением, который будет использоваться для выбора индекса экрана, открываемого в фрейме:

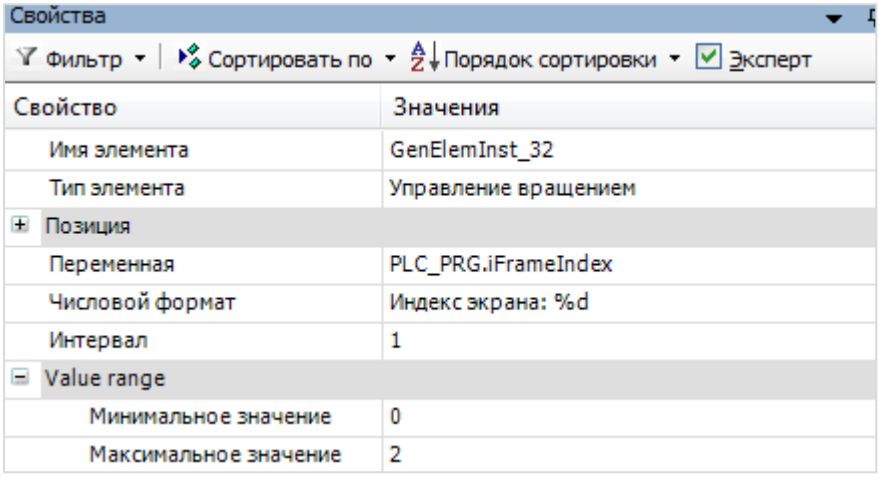


Рисунок 10.76 – Настройки элемента Управление вращением

Для настройки кнопки **Открыть экран** следует во вкладке InputConfiguration к параметру **OnMouseClick** привязать действие Переключить визуализацию во фрейме. Затем выбрать тип выделения **Любая визуализация** и присвоить ей переменные, объявленные в программе **PLC_PRG** (пп. 2).

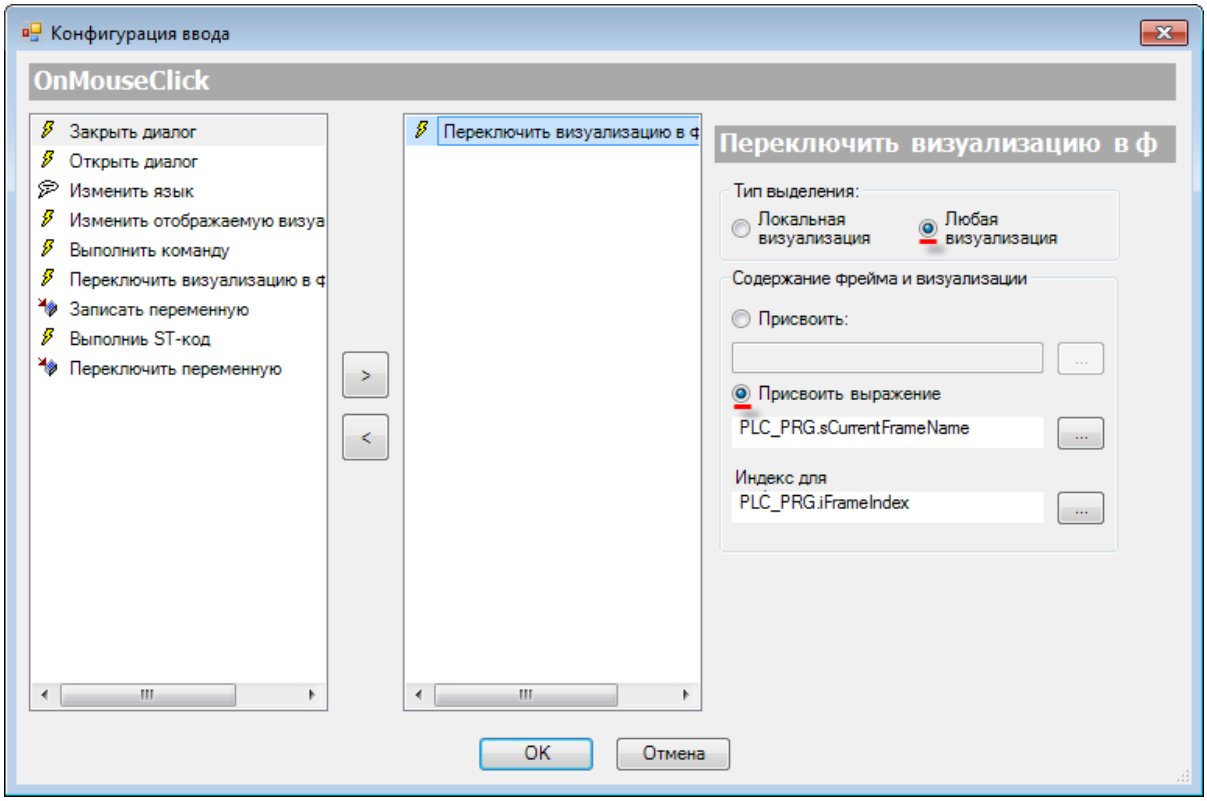
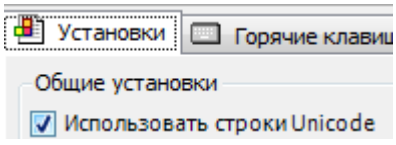


Рисунок 10.77 – Настройки действия кнопки Открыть экран

10. Настроить компонент Менеджер визуализации. В его установках следует поставить галочку **Использовать строки Unicode**:



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

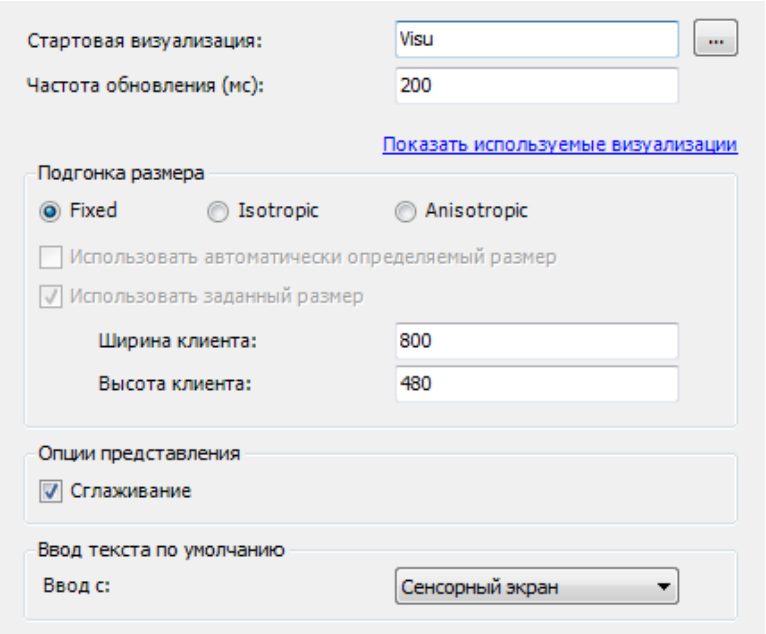


Рисунок 10.78 – Настройки таргет-визуализации

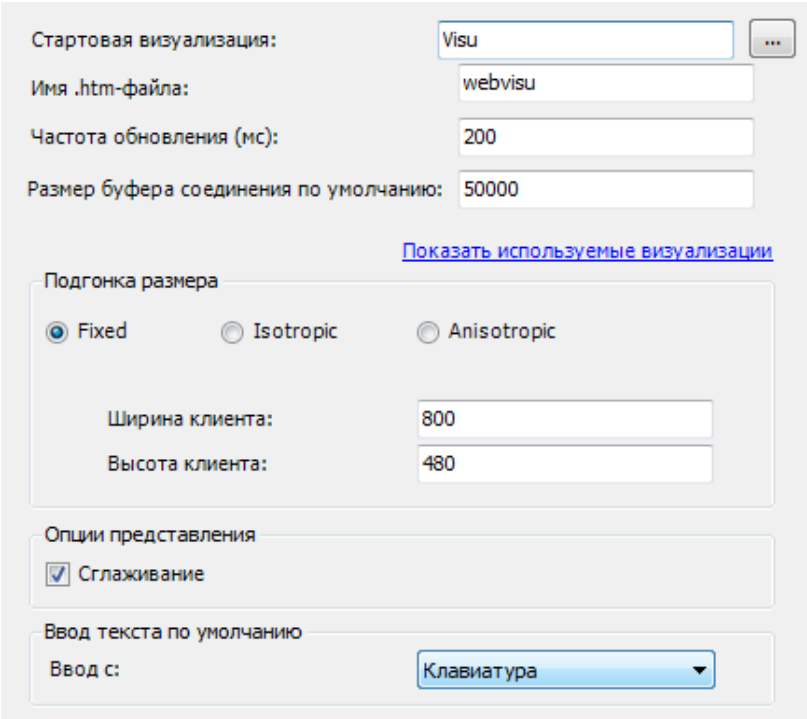


Рисунок 10.79 – Настройки web-визуализации

11. Запустить проект на виртуальном контроллере.
Открыть в веб-браузере страницу http://localhost:8080/webvisu.htm, чтобы увидеть web-визуализацию проекта. ***Следует обратить внимание***, что браузер должен поддерживать **HTML5**.
Проверить функционал проекта.

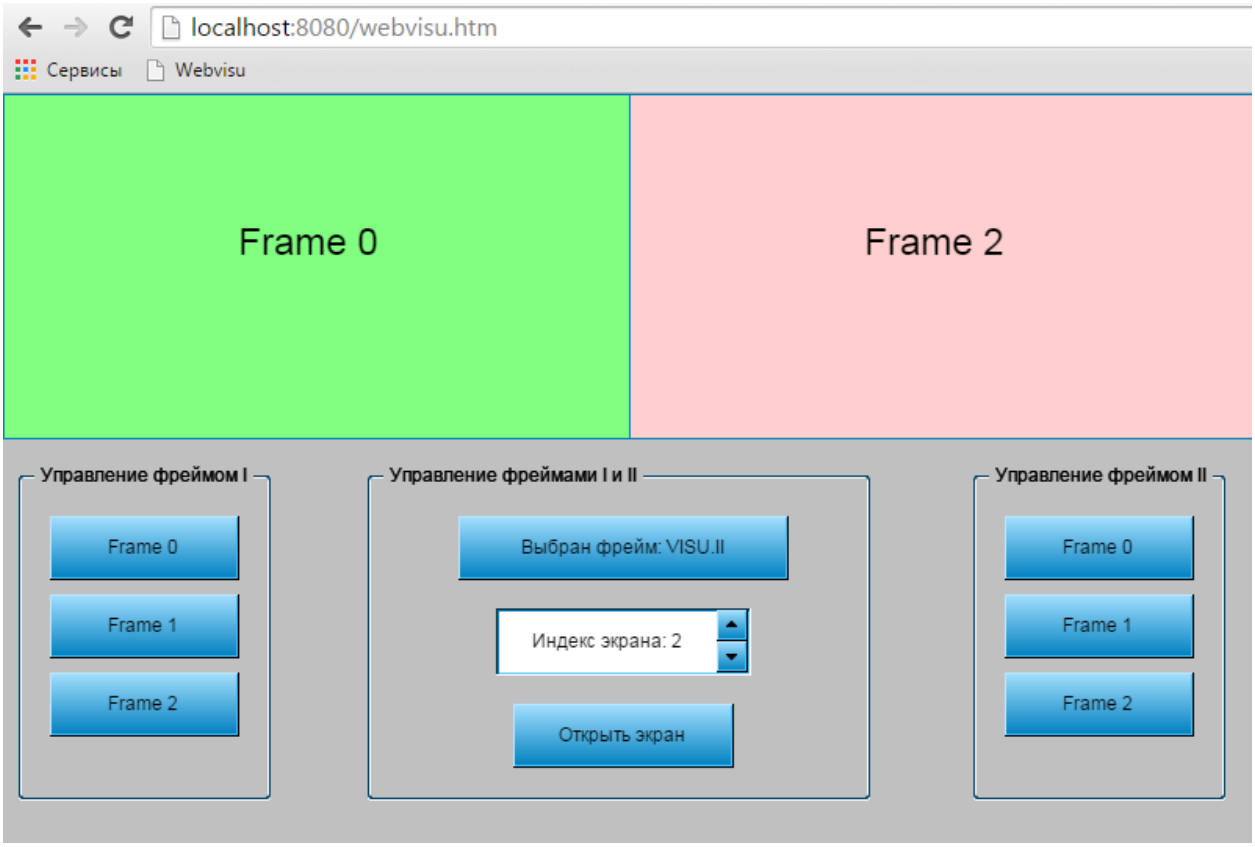


Рисунок 10.80 – Web-визуализация проекта в браузере Chrome

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. Сначала следует нажать на кнопку **Выбран элемент** и ввести 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. Выбрать индекс фрейма.
3. Нажать на кнопку **Открыть этот фрейм**, чтобы открыть экран с данным индексом в выбранном фрейме.

10.2.3 Комбинированное окно

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

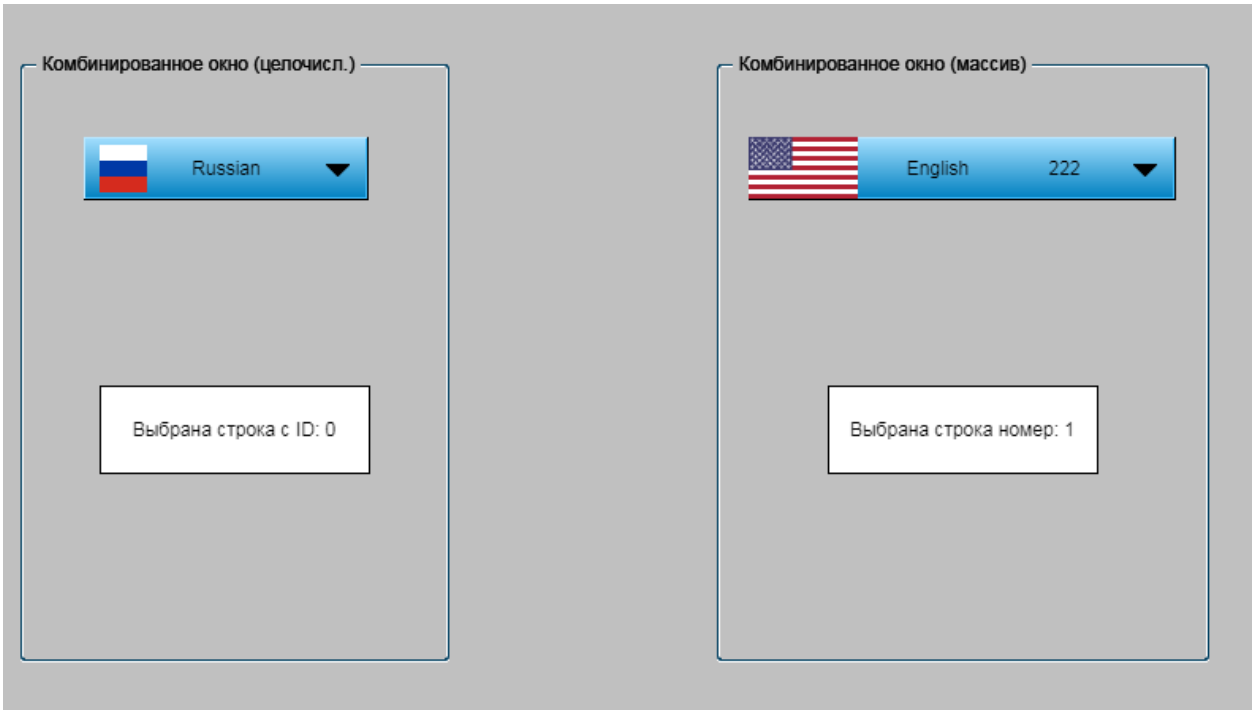


Рисунок 10.81 – Внешний вид примера Комбинированное окно

Пример создан в среде **CODESYS V3.5 SP11 Patch 5** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с таргет-файлом **3.5.11.5**.

Пример доступен для скачивания: Example_Combobox.projectarchive

Для создания примера с использованием элемента **Комбинированное окно** следует:

1. Создать новый ***стандартный*** проект **CODESYS** с названием **Example_Combobox** и настройками по умолчанию: таргет – CODESYS Control Win V3, язык PLC_PRG – **ST**.
2. В программе **PLC_PRG** объявить следующие переменные:

```
PLC_PRG
1  PROGRAM PLC_PRG
2  VAR
3      iVar1:INT; // переменная, содержащая ID строки, выбранной элементом Комбинированное окно - Целочисленное
4      iVar2:INT; // переменная, содержащая номер строки, выбранной элементом Комбинированное окно - Массив
5
6      // Массив для элемента Комбинированное окно - Массив
7      arrTable : ARRAY [0..2, 0..2] OF STRING := ['0','4','8',
8                                                  'Russian','English','Japan',
9                                                  '111','222', '333'];
10 END_VAR
11
12
```

Рисунок 10.82 – Объявление переменных программы PLC_PRG

3. Добавить в проект Пул изображений с названием **ImagePool**. Загрузить в него три картинки с ID **0**, **4** и **8**:

ImagePool

ID	Имя файла	Изображение	Link type
0	flag_of_russia.svg		Embedded and link to file
4	flag_of_the_united_states.svg		Embedded and link to file
8	flag_of_japan.svg		Embedded and link to file

Рисунок 10.83 – Содержимое пула изображений ImagePool

4. Добавить в проект список текстов с названием **TextList**. Его **ID** будут совпадать с **ID** из **Пула изображений**:

TextList

ID	По умолчанию
0	Russian
4	English
8	Japanese

Рисунок 10.84 – Содержимое списка текстов TextList

5. Добавить в проект экран визуализации с названием **Visualization**. В его свойствах выбрать размер **800 × 480.** Добавить на экран два элемента Группа, элемент Комбинированное окно – Целочисленное, элемент Комбинированное окно – Массив и два элемента Прямоугольник.

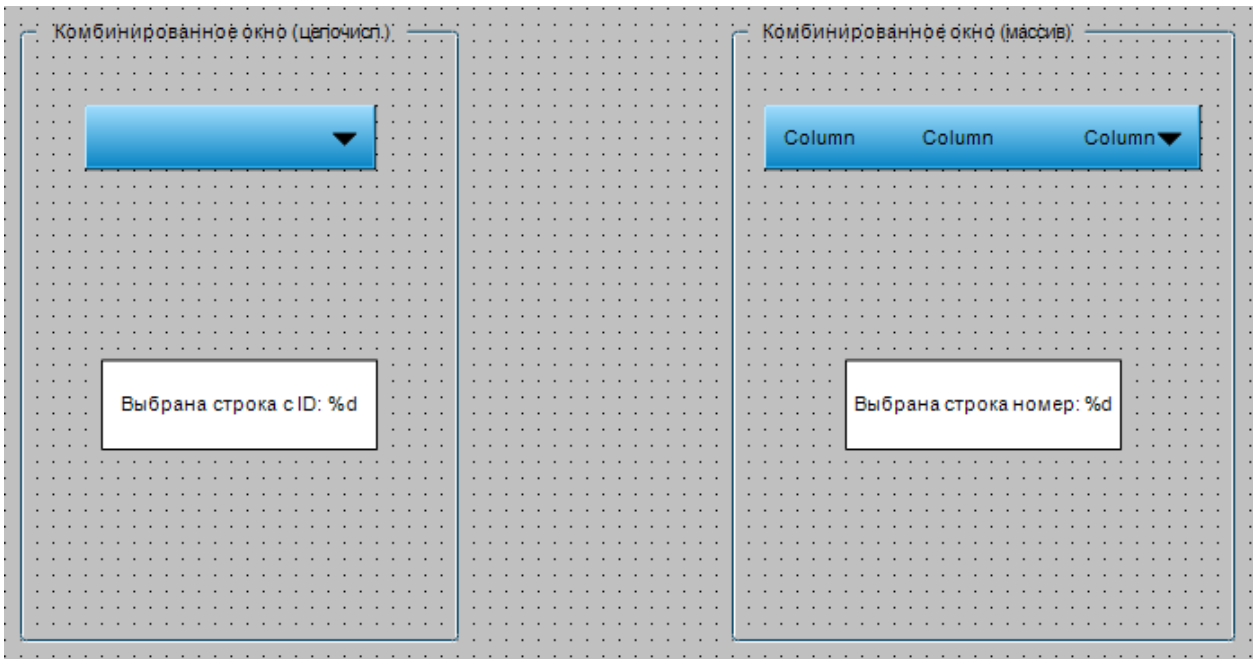


Рисунок 10.85 – Содержимое экрана Visualization

6. Настроить элемент Комбинированное окно – Целочисленное:

Свойства

Фильтр | Сортировать по | Порядок сортировки | Эксперт

Свойство	Значения
Имя элемента	GenElemInst_8
Тип элемента	Комбинированное окно - Целочисленн
⊟ Позиция	
X	60
Y	110
Ширина	180
Высота	40
Переменная	PLC_PRG.iVar1
Список текстов	'TextList'
Пул изображений	'ImagePool'
⊞ Параметры списка	
⊞ Тексты	
⊟ Поддиапазон	
Использовать поддиапазон	☐
Минимальное значение	0
Конечный индекс	8
Отбрасывать недостающие тексты	☑
⊞ Свойства текста	
⊞ Переменные состояний	

Рисунок 10.86 – Параметры элемента Комбинированное окно – Целочисленное

7. Настроить элемент Комбинированное окно – Массив:

Свойства

Фильтр | Сортировать по | Порядок сортировки | Эксперт

Свойство	Значения
Имя элемента	GenElemInst_14
Тип элемента	Комбинированное окно - Массив
⊟ Позиция	
X	480
Y	110
Ширина	270
Высота	40
Переменная	PLC_PRG.iVar2
Массив данных	PLC_PRG.arrTable
⊟ Столбцы	
⊟ Столбец	
⊟ [0]	☑
Ширина	70
Столбец изображения	☑
⊞ Конфигурация изобра...	
Выравнивание текста...	По центру
⊟ [1]	☑
Ширина	100
Столбец изображения	☐
⊞ Конфигурация изобра...	
Выравнивание текста...	По центру
⊟ [2]	☑
Ширина	100
Столбец изображения	☐
⊞ Конфигурация изобра...	
Выравнивание текста...	По центру
Макс. индекс массива	2
Высота строки	30
Число видимых строк	3
Размер полосы прокрутки	20
⊞ Тексты	
⊞ Свойства текста	
⊞ Переменные состояний	

Рисунок 10.87 – Параметры элемента Комбинированное окно – Массив

8. Настроить элементы Прямоугольник. Они будут отличаться только привязываемыми переменными – **iVar1** для первого и **iVar2** для второго соответственно.

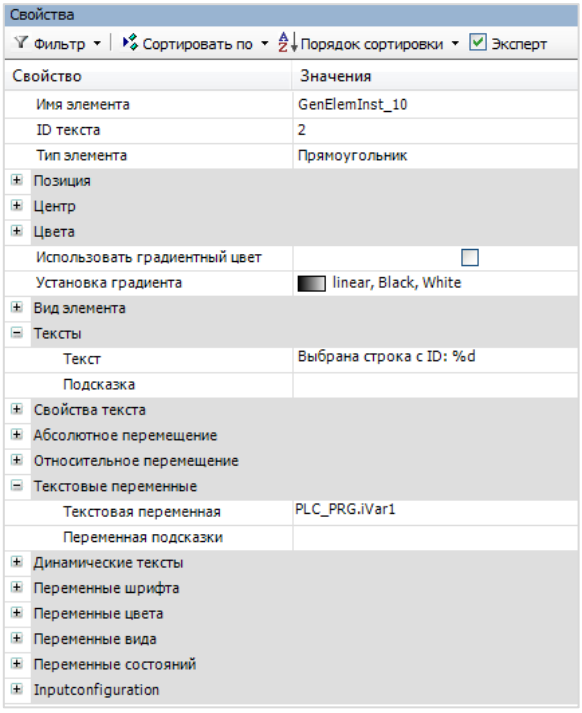
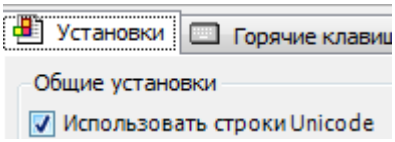


Рисунок 10.88 – Параметры элемента Прямоугольник 1

9. Настроить Менеджер визуализации. В его установках поставить галочку **Использовать строки Unicode**:



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

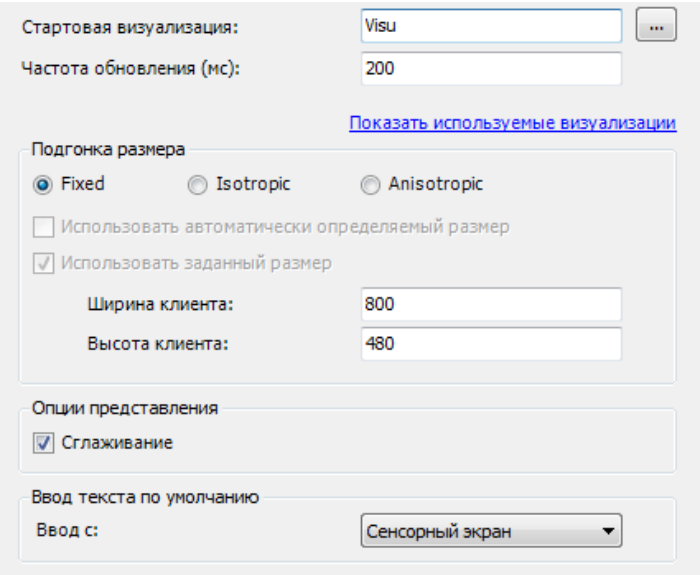


Рисунок 10.89 – Настройки таргет-визуализации

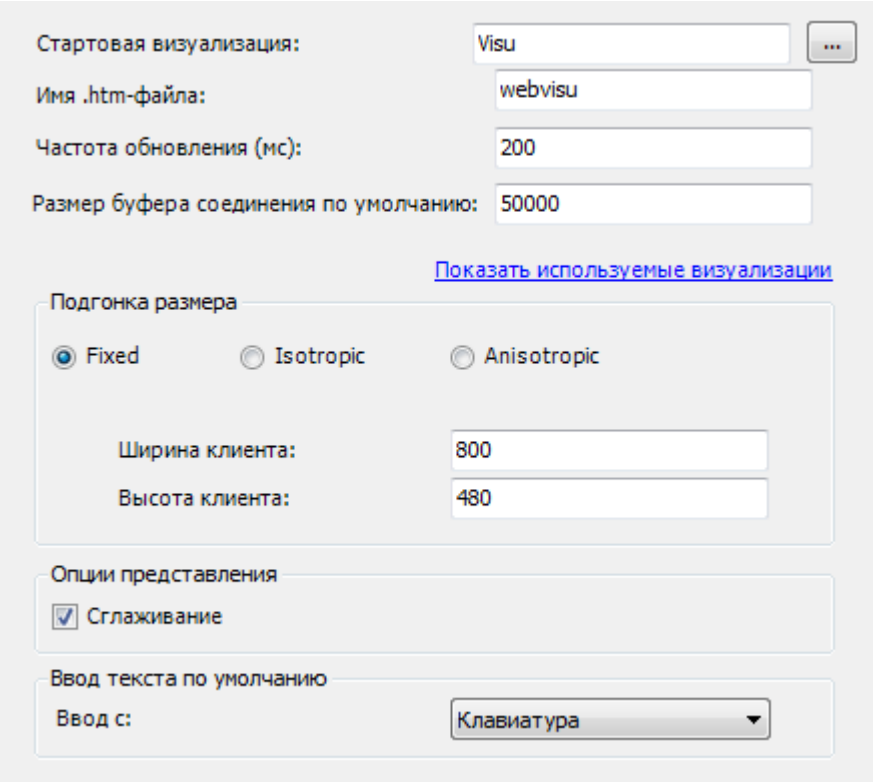


Рисунок 10.90 – Настройки web-визуализации

10. Запустить проект на виртуальном контроллере.
    Открыть в веб-браузере страницу http://localhost:8080/webvisu.htm, чтобы увидеть web-визуализацию проекта. ***Следует обратить внимание***, что браузер должен поддерживать **HTML5**.
    Проверить функционал проекта.

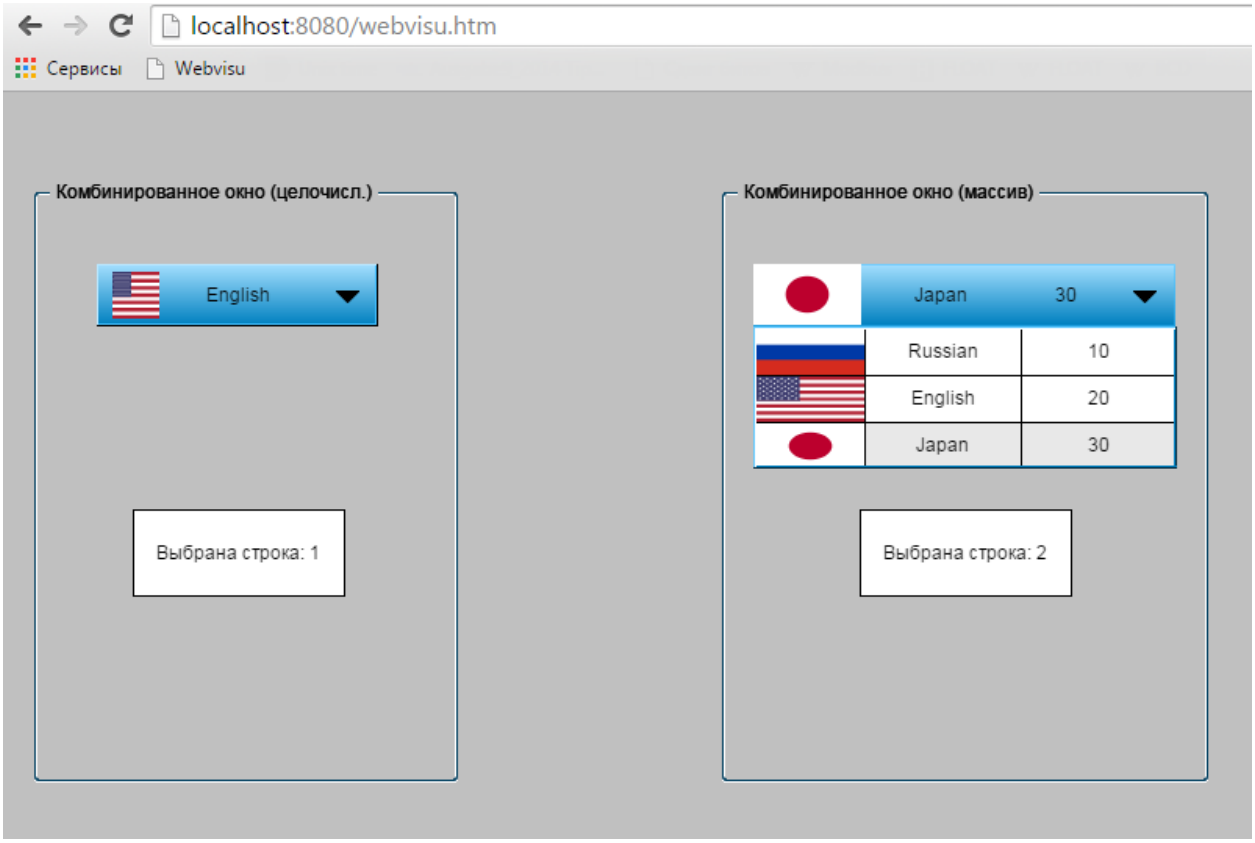


Рисунок 10.91 – Web-визуализация проекта в браузере Chrome

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** идентификатора (**ID**) 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.

ПРИМЕЧАНИЕ
В версии **CODESYS V3.5 SP11** могут возникнуть проблемы в случае использования в элементе **Комбинированное окно – Массив** столбца изображений, если в **Менеджере визуализации** установлена галочка **Использовать строки Unicode**. В данном случае изображения, **ID** которых превышают один символ, не будут отображаться. В рамках данного примера используются односимвольные ID (‘0’, ‘4’, 8’).

10.2.4 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 данного примера будет рассмотрена только первая функция кнопки.

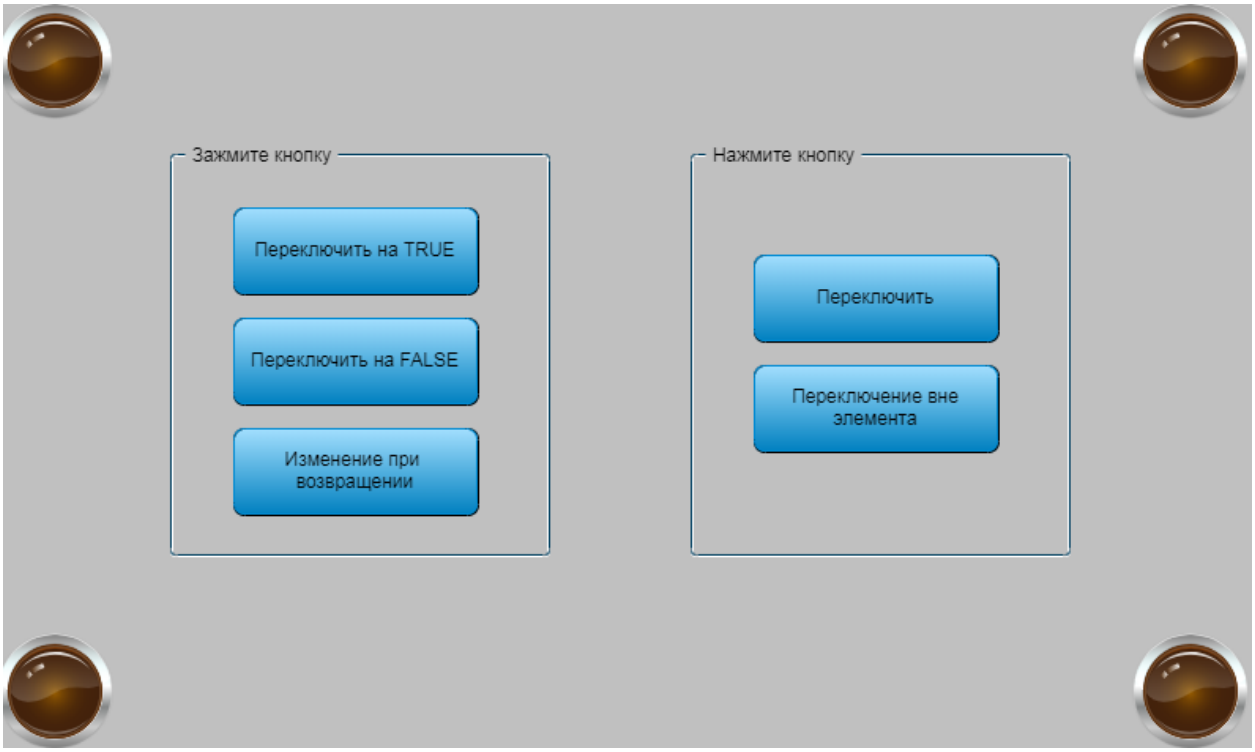


Рисунок 10.92 – Внешний вид примера Кнопка

Пример создан в среде **CODESYS V3.5 SP11 Patch 5** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с таргет-файлом **3.5.11.5**.

Пример доступен для скачивания: Example_Button.projectarchive

Для создания примера с использованием элемента **Кнопка** следует:

1. Создать новый ***стандартный*** проект **CODESYS** с названием **Example_Button** и настройками по умолчанию: таргет – CODESYS Control Win V3, язык PLC_PRG – **ST**.
2. В программе **PLC_PRG** объявить переменную **bVar** типа **BOOL**:

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

Рисунок 10.93 – Объявление переменных программы PLC_PRG

3. Добавить в проект экран визуализации **Visualization**
   В его свойствах выбрать размер **800 × 480.** Экран будет содержать четыре элемента Индикатор, два элемента Группа и пять элементов Кнопка.

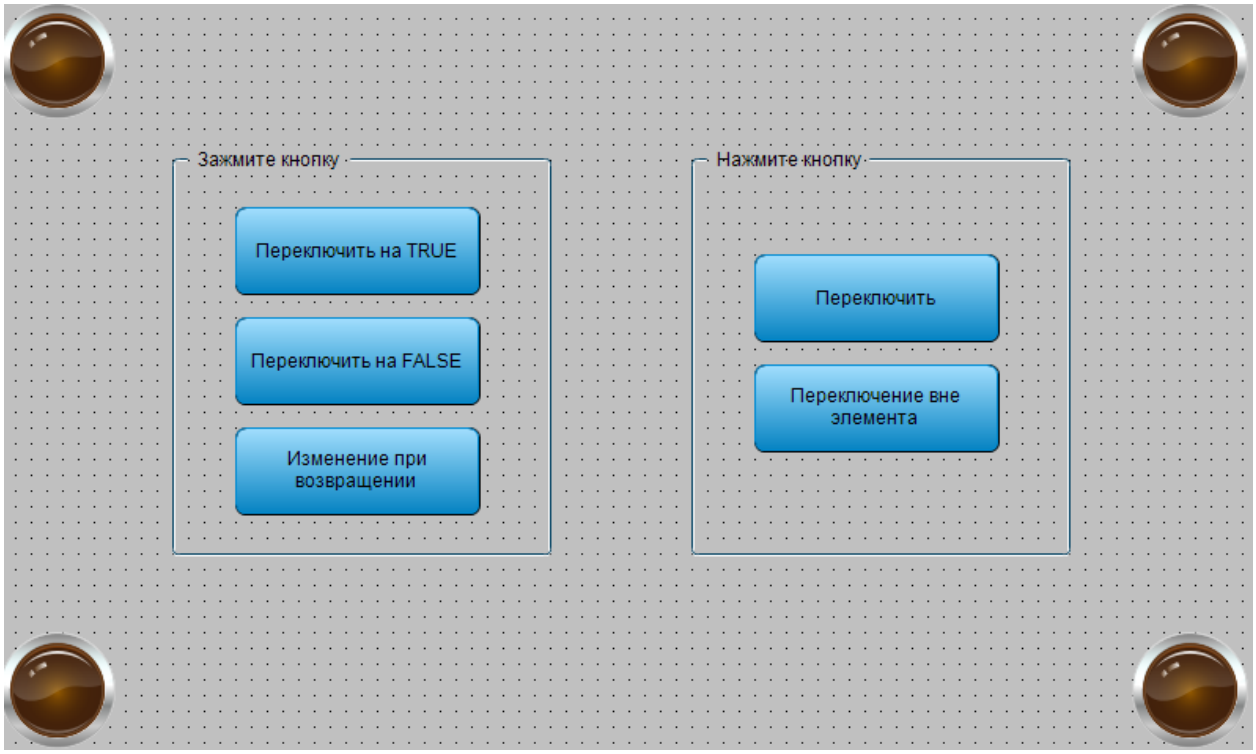


Рисунок 10.94 – Содержимое экрана Visualization

4. К каждому из четырех индикаторов привязать переменную **bVar**:

Свойства

Фильтр | Сортировать по | Порядок сортировки | Эксперт

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

Рисунок 10.95 – Настройки элемента Индикатор

5. Настроить кнопку **Переключить на TRUE**:

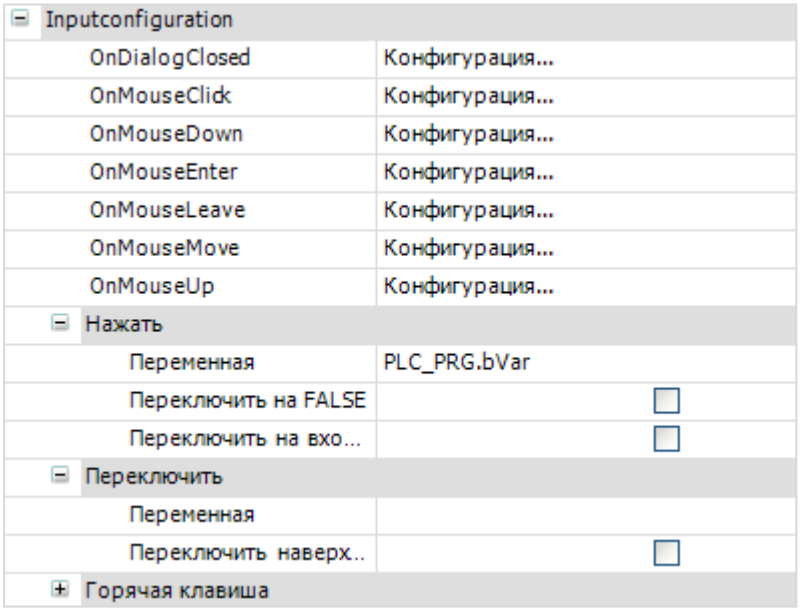


Рисунок 10.96 – Настройки кнопки Переключить на TRUE

6. Настроить кнопку **Переключить на FALSE**:

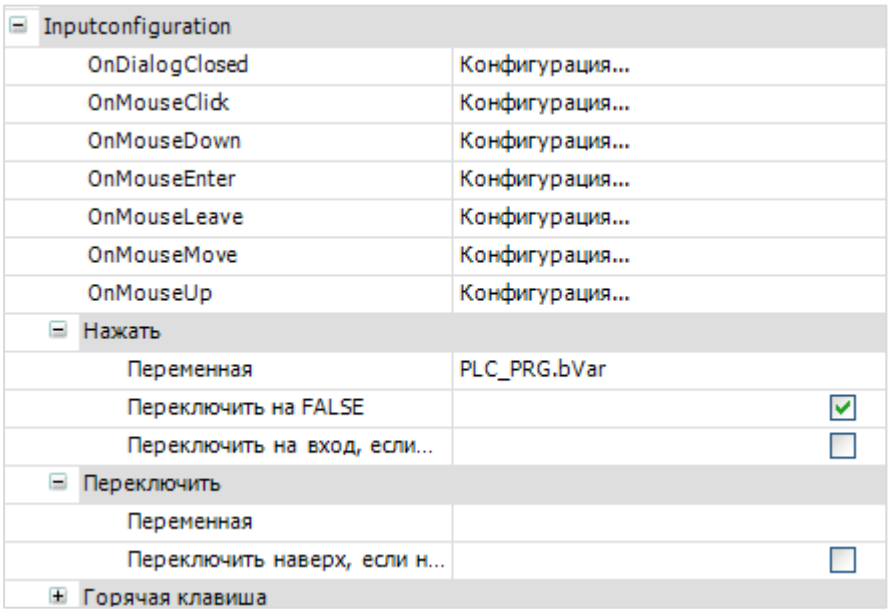


Рисунок 10.97 – Настройки кнопки Переключить на FALSE

7. Настроить кнопку **Изменение при возвращении**:

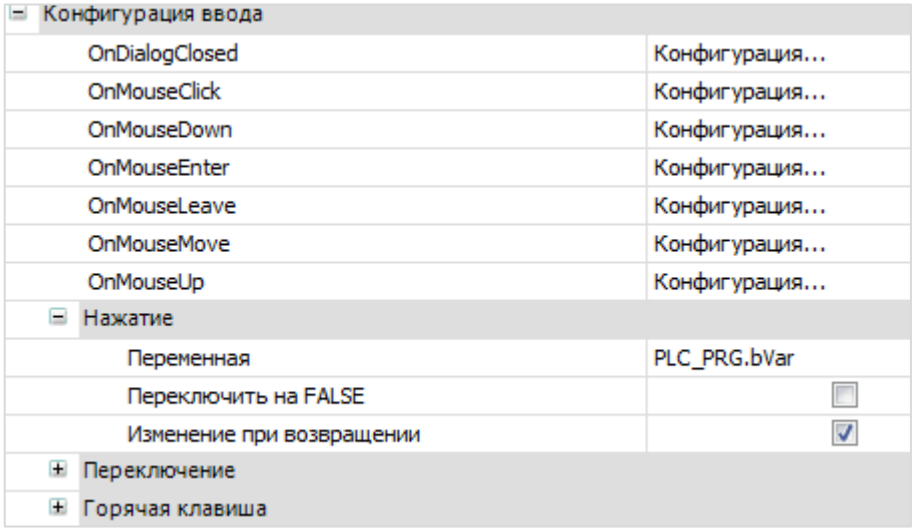


Рисунок 10.98 – Настройки кнопки Изменение при возвращении

8. Настроить кнопку **Переключить**:

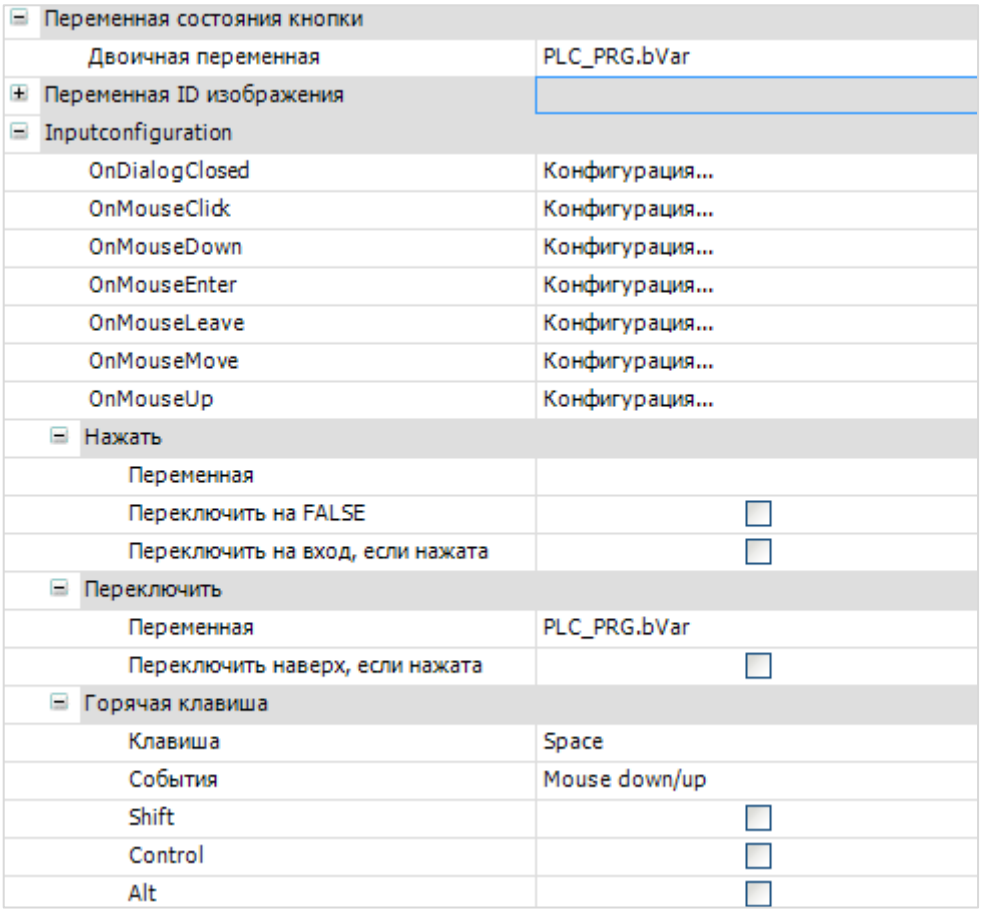


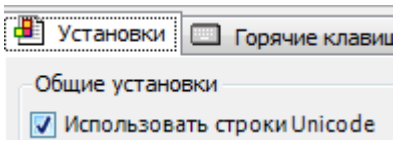
Рисунок 10.99 – Настройки кнопки Переключить

9. Настроим кнопку **Переключение вне элемента:**

Конфигурация ввода	
OnDialogClosed	Конфигурация...
OnMouseClick	Конфигурация...
OnMouseDown	Конфигурация...
OnMouseEnter	Конфигурация...
OnMouseLeave	Конфигурация...
OnMouseMove	Конфигурация...
OnMouseUp	Конфигурация...
Нажатие	
Переключение	
Переменная	PLC_PRG.bVar
Переключение вне элемента	☑

Рисунок 10.100 – Настройки кнопки Переключение вне элемента

10. Настроить компонент Менеджер визуализации. В его установках следует поставить галочку **Использовать строки Unicode**:



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

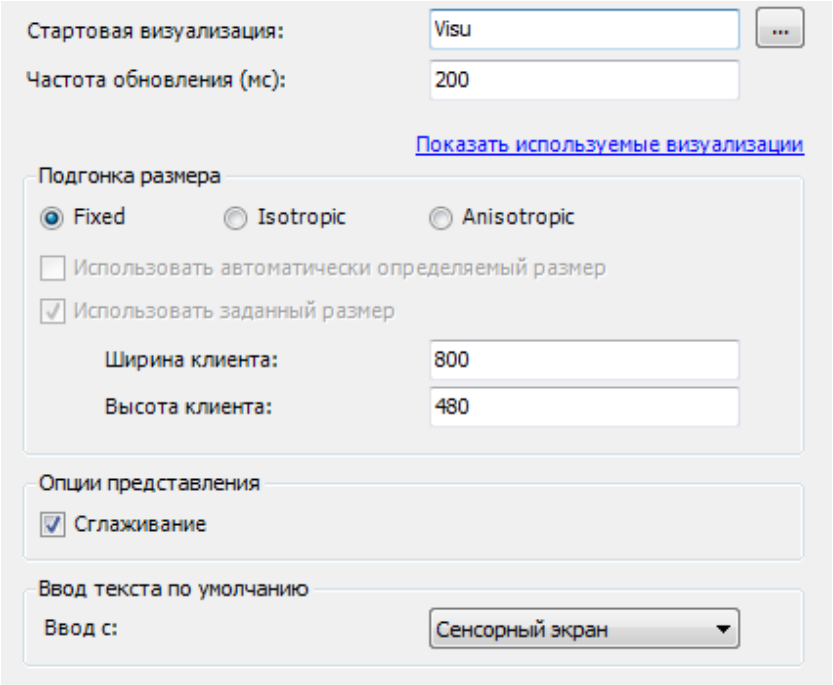


Рисунок 10.101 – Настройки таргет-визуализации

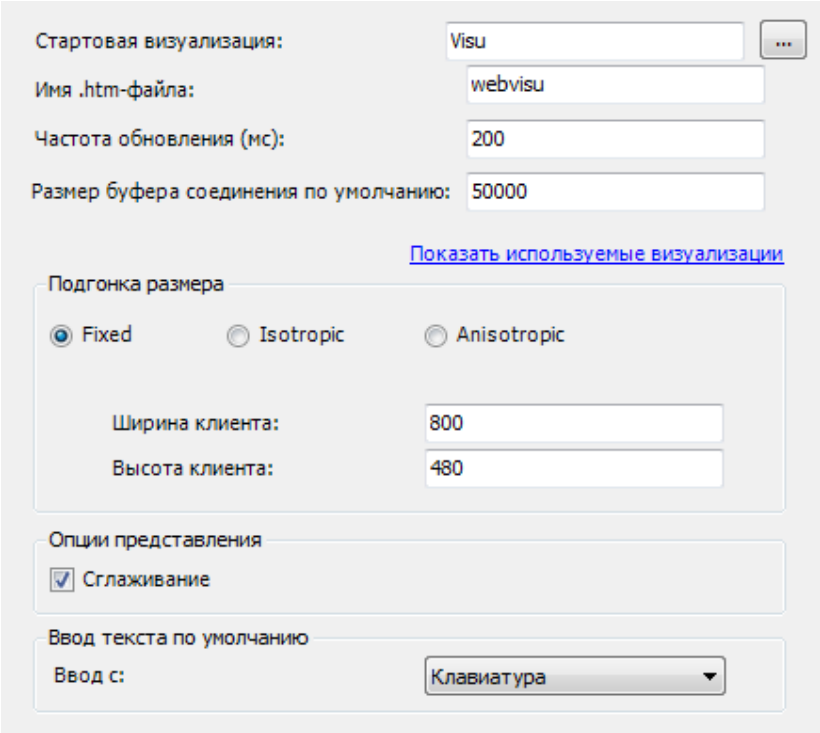


Рисунок 10.102 – Настройки web-визуализации

11. Запустить проект на виртуальном контроллере. Открыть в веб-браузере страницу http://localhost:8080/webvisu.htm, чтобы увидеть web-визуализацию проекта. ***Следует обратить внимание***, что браузер должен поддерживать **HTML5**. Проверить функционал проекта (пункты выполняются последовательно).

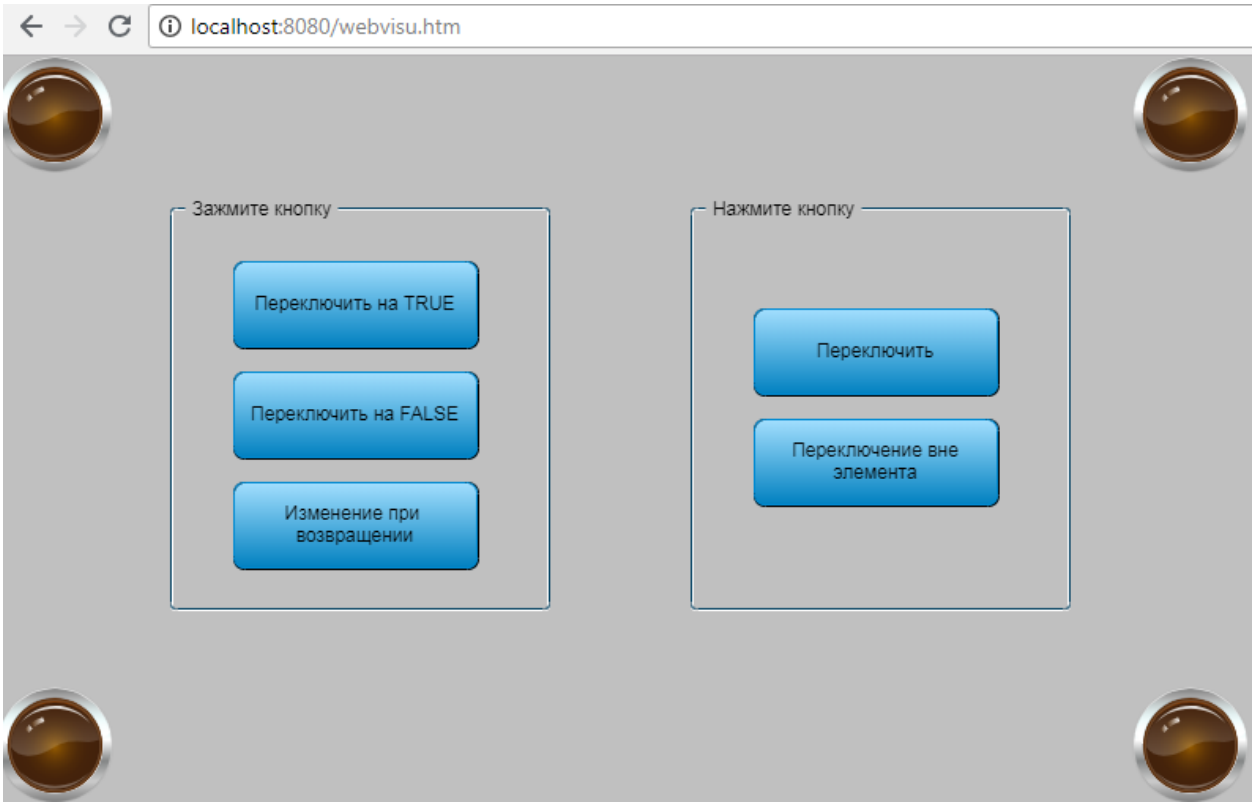


Рисунок 10.103 – Web-визуализация проекта в браузере Chrome

Порядок работы примера с использованием элемента **Кнопка**:

1. При зажатии кнопки **Переключить на TRUE**, переменная **bVar** переключается в **TRUE** – лампы загораются. При отпускании 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 происходит попытка переключить в **FALSE** переменную, которая и так находится в этом состоянии). Последующее зажатие кнопки приводит к переключению переменной в **FALSE**, отпускание – к возвращению ее в **TRUE**.
3. После первого нажатия на кнопку **Изменение при возвращении**, переменная переключатся в состояние **FALSE** (этот эффект связан с тем, что происходит попытка переключить в **TRUE** переменную, которая и так находится в этом состоянии). Последующее зажатие кнопки приводит к переключению переменной в **TRUE**. Не отпуская кнопку, следует перетащить курсор за ее пределы – переменная переключится в **FALSE**. После возвращения курсор в пределы элемента – переменная переключится в **TRUE**. С отпусканием кнопки – 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**). В состоянии **TRUE** кнопка будет отображаться «зажатой». В web-визуализации вместо нажатия на саму кнопку, можно нажимать на кнопку **Пробел** на клавиатуре.
5. С нажатием кнопки **Переключение вне элемента** и, не отпуская ее, следует переместить курсор за пределы элемента, после чего отпустить кнопку. Переменная изменит свое состояние на противоположное (с **FALSE** на **TRUE** или с **TRUE** на **FALSE**).

10.2.5 Таблица

Данный пример посвящен работе с графическим примитивом Таблица. В рамках примера рассмотрена запись в таблицу структуры данных по команде пользователя.

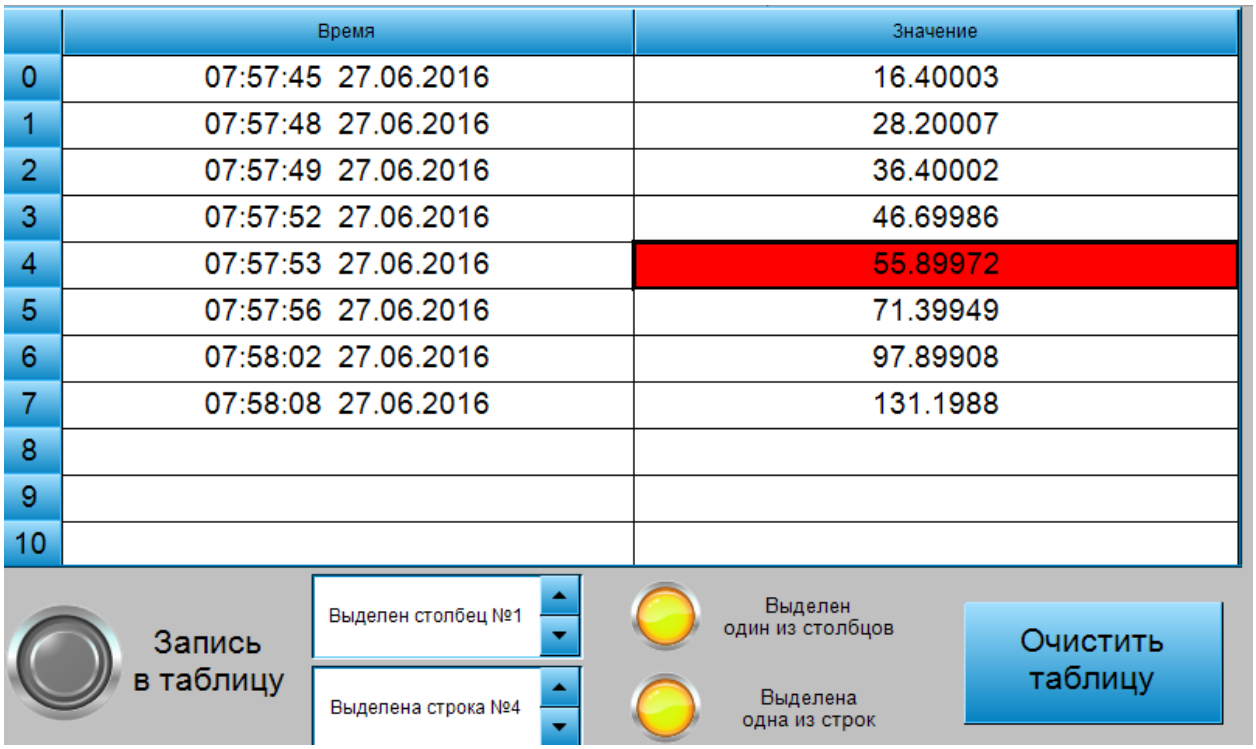


Рисунок 10.104 – Внешний вид примера Таблица

Пример создан в среде **CODESYS V3.5 SP11 Patch 5** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с таргет-файлом **3.5.11.5**.

Пример доступен для скачивания: Example_Table.projectarchive

Для создания примера с использованием элемента **Таблица** следует:

1. Создать новый ***стандартный*** проект **CODESYS** с названием **Example_Table** и настройками по умолчанию: таргет – CODESYS Control Win V3, язык PLC_PRG – **ST**.
2. Добавить в проект компонент **DUT** типа **Структура** с названием **TableStruct**:

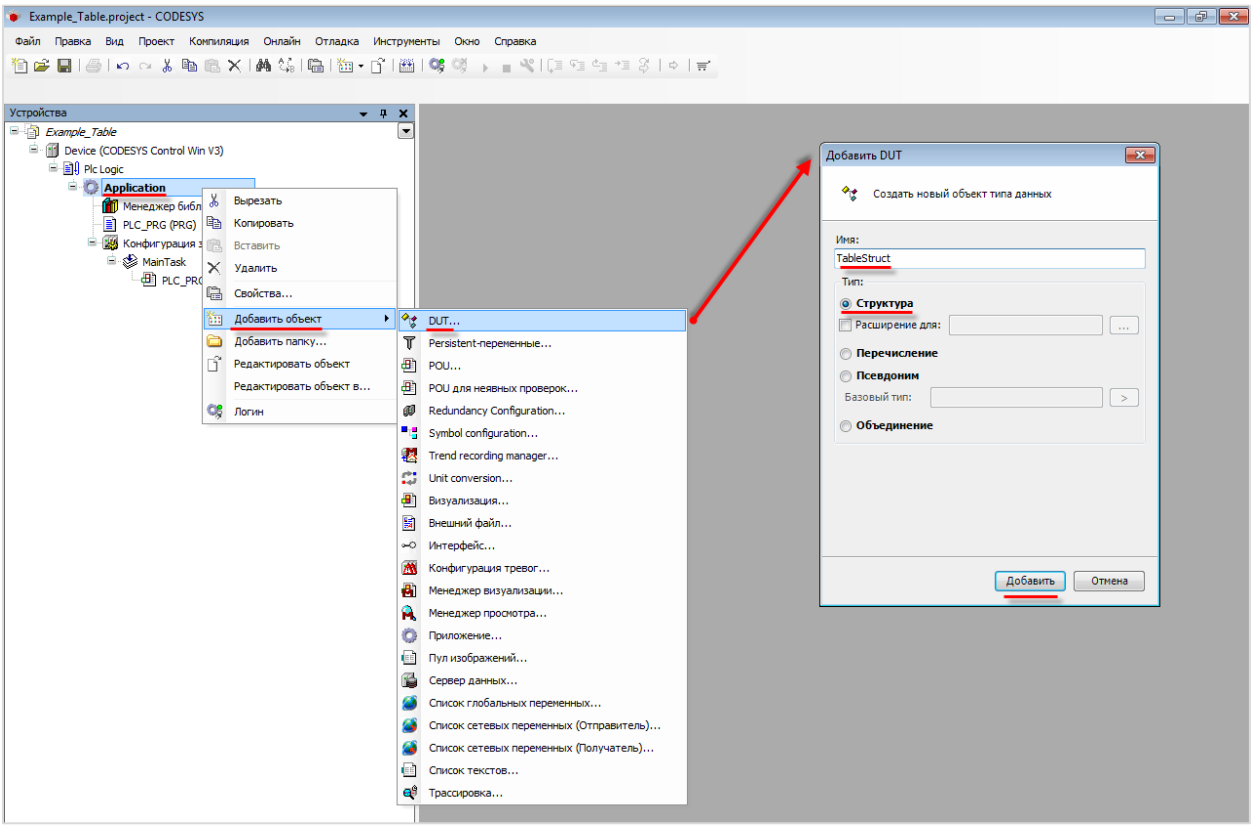


Рисунок 10.105 – Добавление в проект структуры TableStruct

3. Структура представляет собой пользовательский тип данных, который содержит набор переменных разных типов. Структура будет содержать две переменные типа **STRING**:

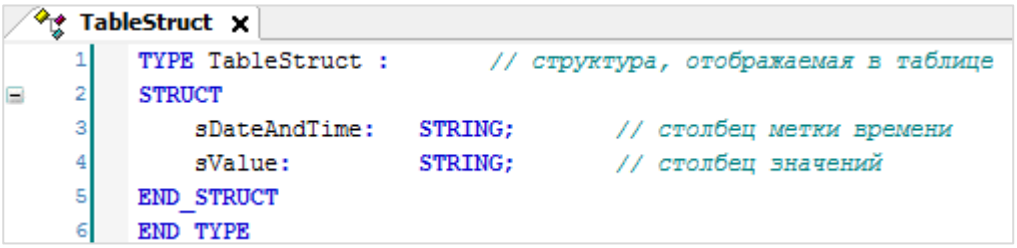


Рисунок 10.106 – Объявление переменных структуры TableStuct

Использование переменных типа **STRING** связано с наличием у переменных начальных значений (по умолчанию – «нулевых»). При использовании типов **DT** и **REAL**, в начальный момент времени таблица имела бы приблизительно такой вид:

dt#1970-01-01-00:00:00	0.0
dt#1970-01-01-00:00:00	0.0
dt#1970-01-01-00:00:00	0.0
dt#1970-01-01-00:00:00	0.0
dt#1970-01-01-00:00:00	0.0

Чтобы избежать этого, используются переменные типа **STRING**, начальные значения которых (‘’) в визуализации представляются пустой строкой.

4. В **Менеджере библиотек** добавить библиотеку **CAA DTUtil** (библиотека требуется нам, чтобы работать с системным временем контроллера):

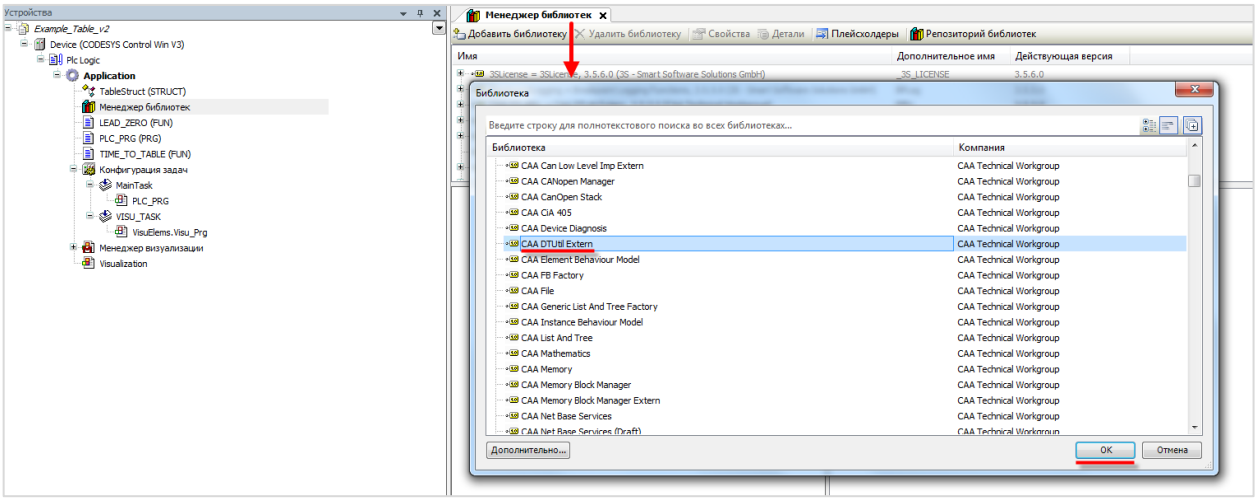


Рисунок 10.107 – Добавление в проект библиотеки CAA DTUtil

ПРИМЕЧАНИЕ

В контроллерах ОВЕН системное время доступно в узле **OwenRTC** таргет-файла (в том числе в виде форматированной строки).

5. В программе **PLC_PRG** объявить следующие переменные:

PLC_PRG

```
PROGRAM PLC_PRG
VAR
    arrTable:              ARRAY [0..10] OF TableStruct;     // массив для структуры

    MyGetDateAndTime:      DTU.GetDateAndTime;               // экземпляр функционального блока чтения системного времени
    dtCurrentDateAndTime:  DT;                               // системное время в формате DT

    bTrigger:              BOOL;                             // триггер записи

    rValue:                REAL;                             // переменная, значение которой отображается в таблице
    iIndex:                INT;                              // номер строки таблицы, в которую осуществляется запись

    iCounter:              INT;                              // счетчик (для очистки таблицы по нажатию на кнопку)

    iColumnSelection:      INT;                              // номер выделенного столбца
    iLineSelection:        INT;                              // номер выделенной строки
    bColumnSelection:      BOOL;                             // флаг выделения любого из столбцов таблицы
    bLineSelection:        BOOL;                             // флаг выделения любой из строк таблицы
END_VAR
```

Рисунок 10.108 – Объявление переменных программы PLC_PRG

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

```
// эмуляция изменения значений переменной
rValue := rValue + 0.1;

 // вызов ФБ считывания системного времени
MyGetDateAndTime(xExecute := NOT(MyGetDateAndTime.xDone) );

// если ФБ успешно завершил работу...
IF MyGetDateAndTime.xDone THEN
    // если ФБ успешно завершил работу...
    dtCurrentDateAndTime := MyGetDateAndTime.dtDateAndTime;
END_IF

 // запись строки в таблицу по триггеру
IF bTrigger THEN

    arrTable[iIndex].sDateAndTime := TIME_TO_TABLE(dtCurrentDateAndTime); // запись текущего времени, преобразованного...
                                                                          // ...с помощью функции TIME_TO_TABLE
    arrTable[iIndex].sValue := REAL_TO_STRING(rValue);                    // запись текущего значения переменной
    iIndex   := iIndex + 1;                                               // переход на следующую строку таблицы
    bTrigger := FALSE;                                                    // сброс триггера
END_IF

// после заполнения таблицы - переход на первую строку
// (запись в режиме циклического буфера)
IF iIndex > 10 THEN
    iIndex := 0;
END_IF
```

Рисунок 10.109 – 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**.

Код функции выглядит следующим образом:

TIME_TO_TABLE

```
// функция преобразования системного времени из формата DT в формат STRING (<часы:минуты:секунды день.месяц.год>)

FUNCTION TIME_TO_TABLE : STRING
VAR_INPUT
    dtCurrentDateAndTime:   DT;
END_VAR
VAR
    uiYear:                 UINT;
    uiMonth:                UINT;
    uiDay:                  UINT;
    uiHour:                 UINT;
    uiMinute:               UINT;
    uiSecond:               UINT;

    sYear:                  STRING;
    sMonth:                 STRING;
    sDay:                   STRING;
    sHour:                  STRING;
    sMinute:                STRING;
    sSecond:                STRING;

    sConcat:                STRING;
END_VAR
```

```
DTU.DTSplit(dtCurrentDateAndTime, ADR(uiYear), ADR(uiMonth), ADR(uiDay), ADR(uiHour), ADR(uiMinute), ADR(uiSecond));

sYear   :=  UINT_TO_STRING(uiYear);
sMonth  :=  LEAD_ZERO(uiMonth);
sDay    :=  LEAD_ZERO(uiDay);
sHour   :=  LEAD_ZERO(uiHour);
sMinute :=  LEAD_ZERO(uiMinute);
sSecond :=  LEAD_ZERO(uiSecond);

sConcat :=  CONCAT(sHour,':');
sConcat :=  CONCAT(sConcat,sMinute);
sConcat :=  CONCAT(sConcat,':');
sConcat :=  CONCAT(sConcat,sSecond);
sConcat :=  CONCAT(sConcat,'  ');
sConcat :=  CONCAT(sConcat,sDay);
sConcat :=  CONCAT(sConcat,'.');
sConcat :=  CONCAT(sConcat,sMonth);
sConcat :=  CONCAT(sConcat,'.');
sConcat :=  CONCAT(sConcat,sYear);

TIME_TO_TABLE   :=  sConcat;
```

Рисунок 10.110 – Код функции TIME_TO_TABLE

Функция **LEAD_ZERO** используется для добавления ведущих нулей к разрядам времени.

LEAD_ZERO

```
// функция добавления ведущего нуля к разрядам времени

FUNCTION LEAD_ZERO : STRING
VAR_INPUT
    uiInput:    UINT;
END_VAR
VAR
END_VAR
```

```
IF uiInput > 9 THEN
    LEAD_ZERO := UINT_TO_STRING(uiInput);
ELSE
    LEAD_ZERO := CONCAT('0', UINT_TO_STRING(uiInput));
END_IF
```

Рисунок 10.111 – Код функции LEAD_ZERO

7. Добавить в проект экран визуализации **Visualization**. В его свойствах выбрать размер **800 × 480.** Экран будет содержать элемент Таблица, элемент Нажимной выключатель, два элемента Управление вращением, два элемента Индикатор и элемент Кнопка. Три текстовые надписи сделаны с помощью элементов Метка.

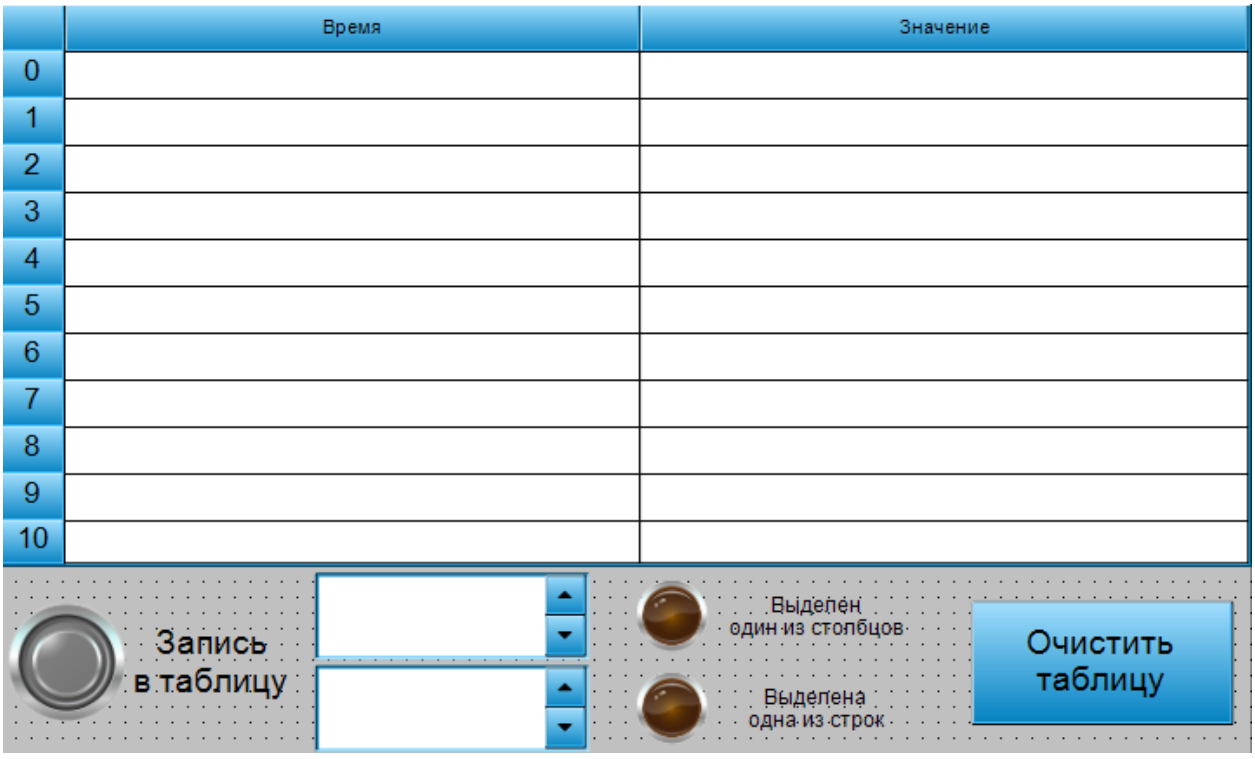


Рисунок 10.112 – Содержимое экрана Visualization

8. Настроить элемент Таблица:

Свойства

Фильтр | Сортировать по | Порядок сортировки | Эксперт

Свойство	Значения
Имя элемента	GenElemInst_24
Тип элемента	Таблица
Массив данных	PLC_PRG.arrTable
Макс. индекс массива	10
Столбцы	
Столбец	
[0]	☑
Заголовок столбца	Время
Ширина	368
Столбец изображения	☐
Конфигурация изобра...	
Выравнивание текста...	По центру
Использовать шаблон	☐
Выравнивание текста...	☐
[1]	☑
Заголовок столбца	Значение
Ширина	389
Столбец изображения	☐
Конфигурация изобра...	
Выравнивание текста...	По центру
Использовать шаблон	☐
Выравнивание текста...	☐
Показать заголовок строки	☑
Показать заголовок столбца	☑
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...	☑
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

Рисунок 10.113 – Параметры элемента Таблица

9. К элементу Нажимной выключатель привязать 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

Рисунок 10.114 – Параметры элемента Нажимной выключатель

10. Настроить 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

Рисунок 10.115 – 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

Рисунок 10.116 – Параметры элемента Управление вращением 2

11. Настроить 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

Рисунок 10.117 – Параметры элемента Индикатор (выделен один из столбцов)

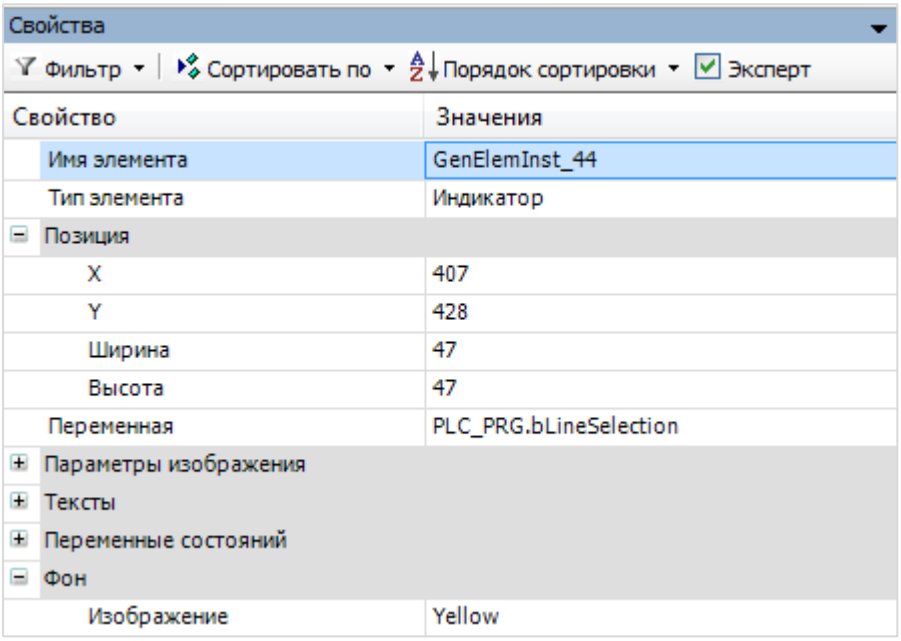


Рисунок 10.118 – Параметры элемента Индикатор (выделена одна из строк)

12. К элементу Кнопка (**Очистить таблицу**) во вкладке InputConfiguration к параметру **OnMouseClick** привязать действие Выполнить ST-код:

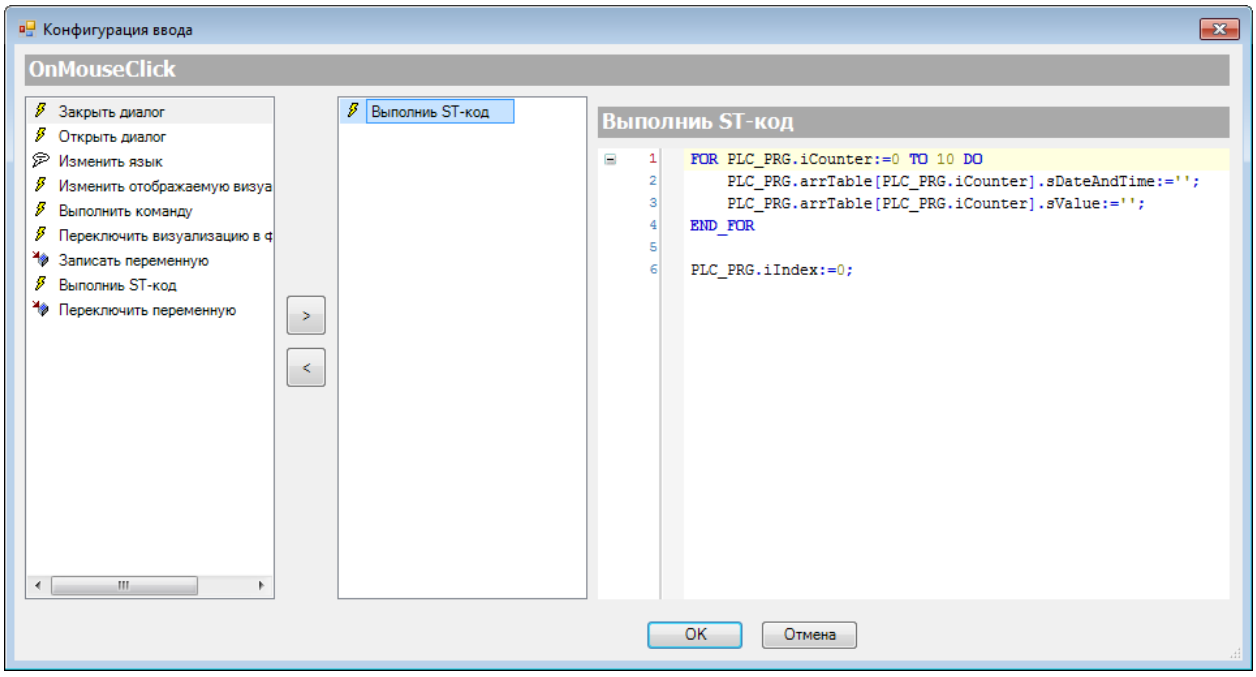
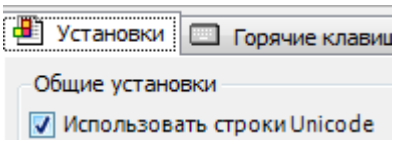


Рисунок 10.119 – ST-код элемента Кнопка (Очистить таблицу)

13. Настроить Менеджер визуализации. В его установках поставить галочку **Использовать строки Unicode**:



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

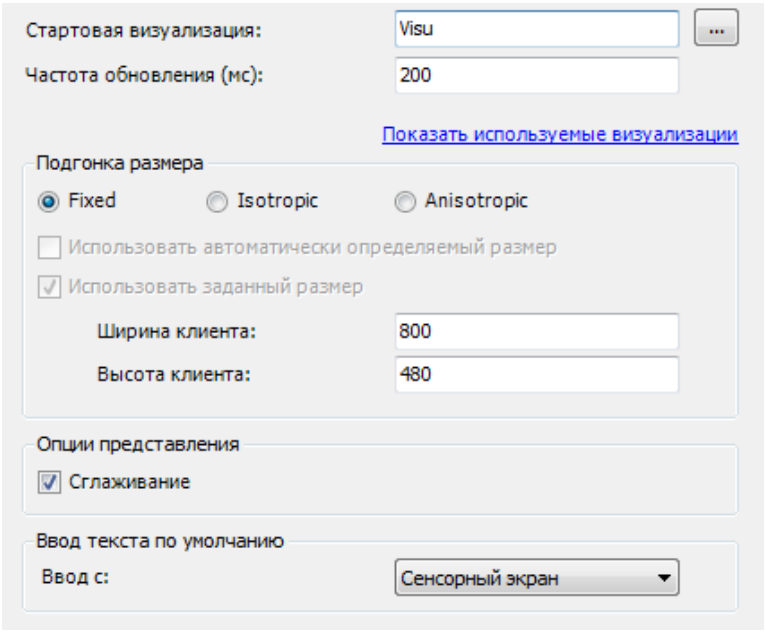


Рисунок 10.120 – Настройки таргет-визуализации

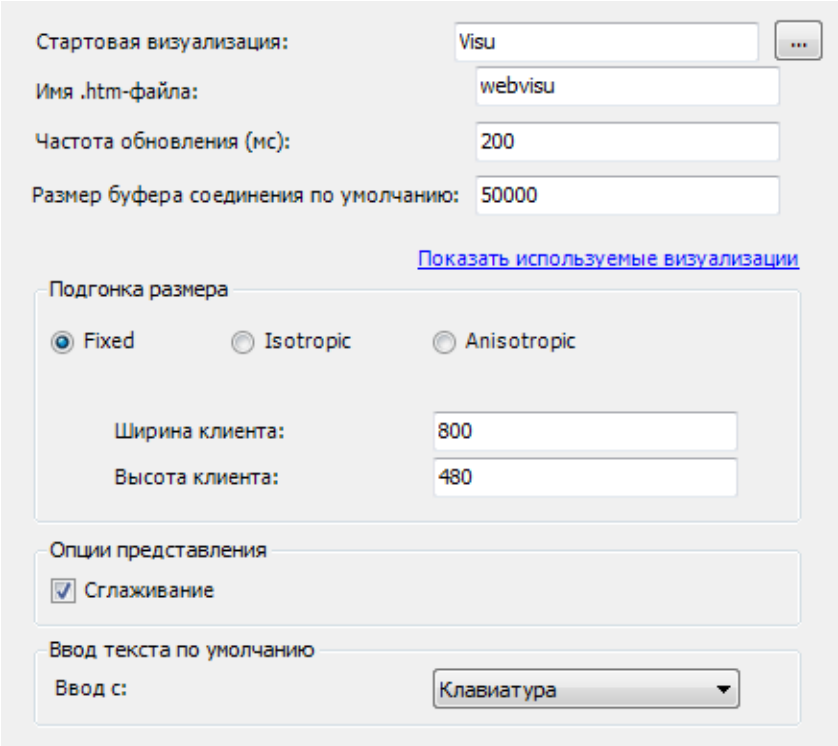


Рисунок 10.121 – Настройки web-визуализации

14. Запустить проект на виртуальном контроллере. Открыть в веб-браузере страницу http://localhost:8080/webvisu.htm, чтобы увидеть web-визуализацию проекта. ***Следует обратить внимание***, что браузер должен поддерживать **HTML5**. Проверить функционал проекта.

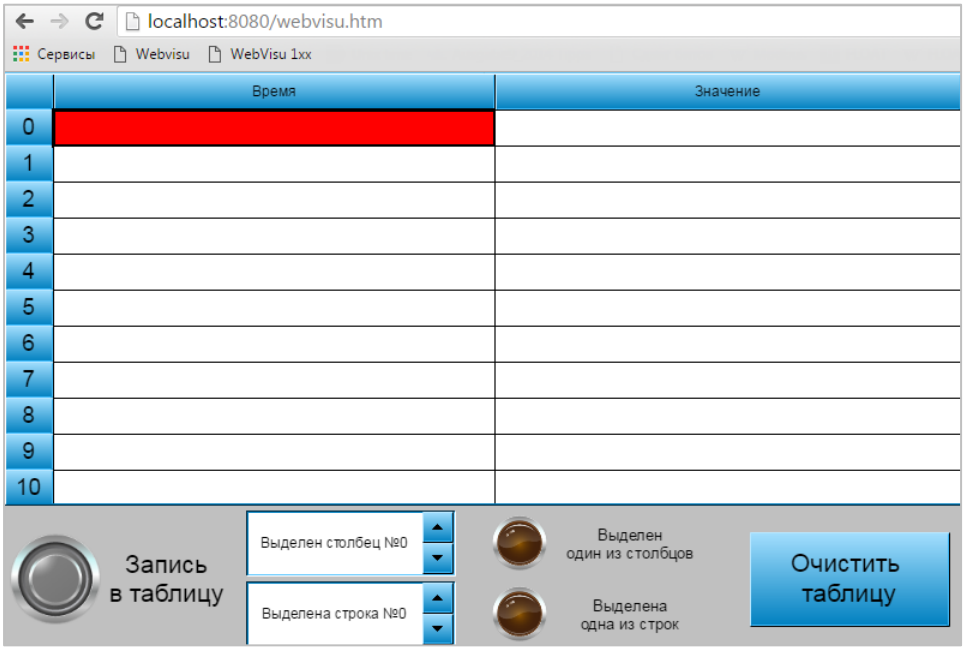


Рисунок 10.122 – Web-визуализация проекта в браузере Chrome

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

Рисунок 10.123 – Отображение структуры в таблице

Следует нажать на десятую ячейку нулевого столбца таблицы.

В элементах **Управление вращением** отобразятся номера строки и столбца ячейки, **Индикаторы** выделения загорятся:

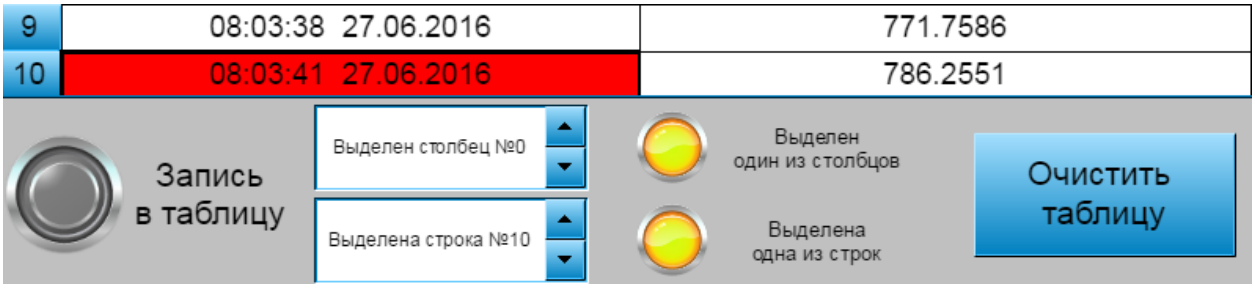


Рисунок 10.124 – Информация о выделенной ячейке таблицы

Затем следует нажать на верхнюю левую клетку таблицу (пустую), чтобы сбросить выделение. Далее нажать на кнопку **Очистить таблицу**, чтобы стереть отображаемые в ней значения.

10.2.6 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).

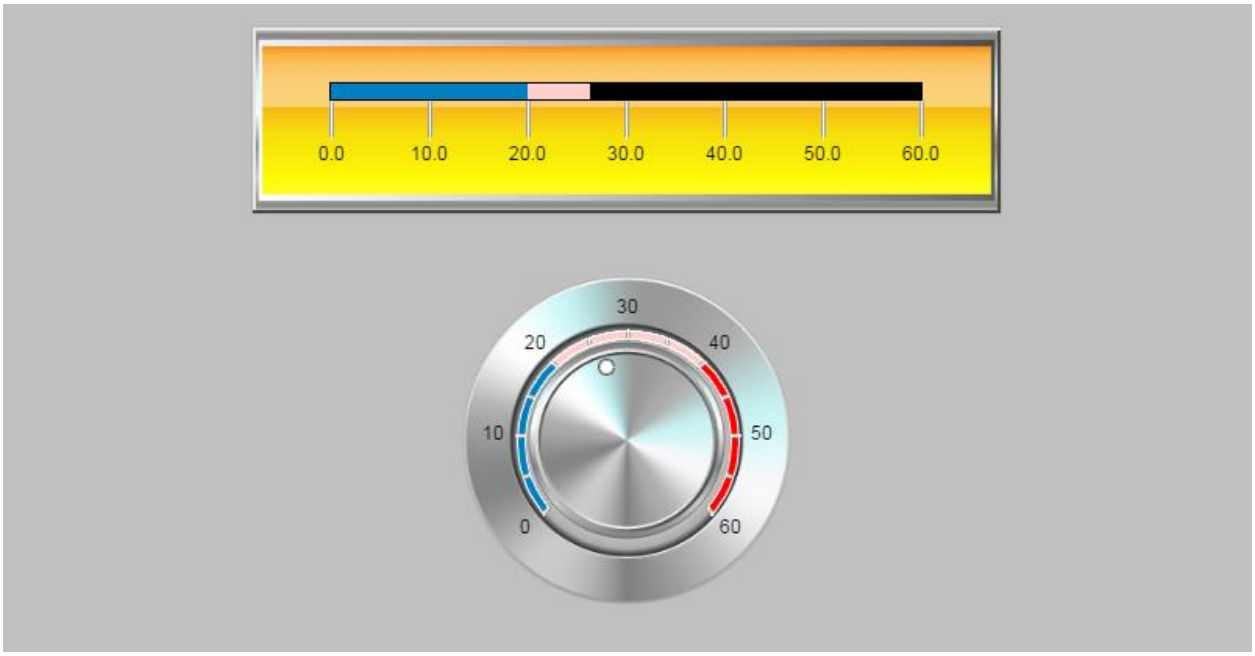


Рисунок 10.125 – Внешний вид примера Отображение линейки и Потенциометр

Пример создан в среде **CODESYS V3.5 SP11 Patch 5** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с таргет-файлом **3.5.11.5**.

Пример доступен для скачивания: Example_BarDisplay.projectarchive

Для создания примера с использованием элементов **Потенциометр** и **Отображение линейки** следует:

1. Создать новый ***стандартный*** проект **CODESYS** с названием **Example_BarDisplay** и настройками по умолчанию: таргет – CODESYS Control Win V3, язык PLC_PRG – **ST**.
2. В программе **PLC_PRG** объявить переменную **rValue** типа **REAL**:

```
1  PROGRAM PLC_PRG
2  VAR
3      rValue:REAL;
4  END_VAR
```

Рисунок 10.126 – Объявление переменных программы PLC_PRG

3. Добавить в проект экран визуализации Visualization. В его свойствах выбрать размер **800 × 480.** Экран будет содержать элемент Отображение линейки и элемент Потенциометр.

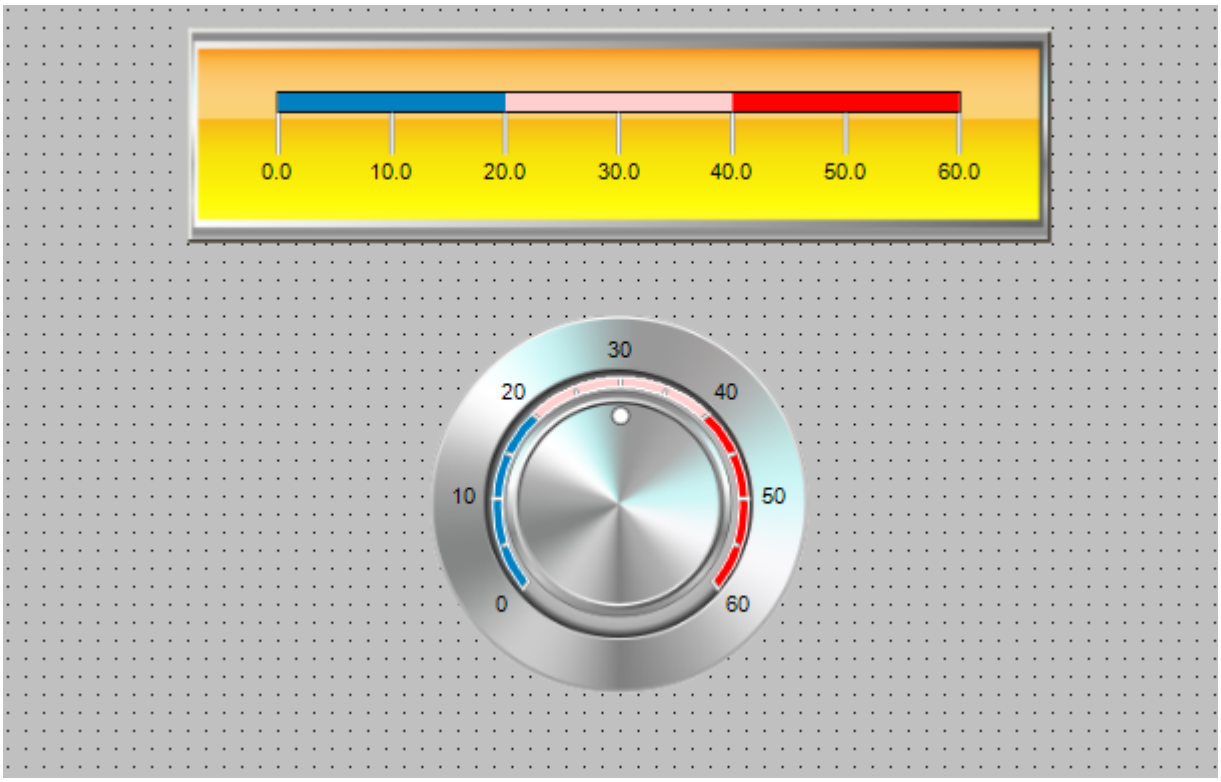


Рисунок 10.127 – Содержание экрана Visualization

4. Настроить 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	☑
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	White
Шкала в 3D	☑
Фрейм элемента	☑
Метка	
Единица	
Шрифт	Font-Standard
Формат шкалы (c-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	White
Фон линейки	☑
Цвет фрейма	212; 208; 200
Переключить весь цвет	☐
Использовать градиентный цвет для лин...	☐
Маркеры цветовых диапазонов	Нет маркеров
Цветовые области	Создать новый
Области	
[0]	Удалить
Начало области	0
Конец области	20
Цвет	Element-Control-Color
[1]	Удалить
Начало области	20
Конец области	40
Цвет	Element-Alarm-Fill-Color
[2]	Удалить
Начало области	40
Конец области	60
Цвет	Element-Alarm-Frame-Color

Рисунок 10.128 – Параметры элемента Отображение линейки

5. Настроить элемент Потенциометр:

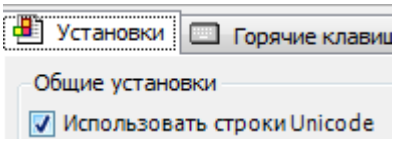
Свойства

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	☑
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	☑
Показать шкалу	☑
Фрейм внутри	☑
Фрейм снаружи	☑
Метка	
Метка	Снаружи
Единица	
Шрифт	Font-Standard
Формат шкалы (c-синтаксис)	%d
Макс. ширина текста меток	28
Высота текста меток	15
Цвет шрифта	Element-Potentiometer-LabelFontColor
Расположение	
Цвета	
Цветовые области	
Долговременные цветовые обл...	☑
Цветовые области	Создать новый
Области	
[0]	Удалить
Начало области	0
Конец области	20
Цвет	Element-Control-Color
[1]	Удалить
Начало области	20
Конец области	40
Цвет	BasicElement-Alarm-Fill-Color
[2]	Удалить
Начало области	40
Конец области	60
Цвет	Element-Alarm-Frame-Color

Рисунок 10.129 – Параметры элемента Потенциометр

6. Настроить компонент Менеджер визуализации. В его установках поставить галочку **Использовать строки Unicode**:



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

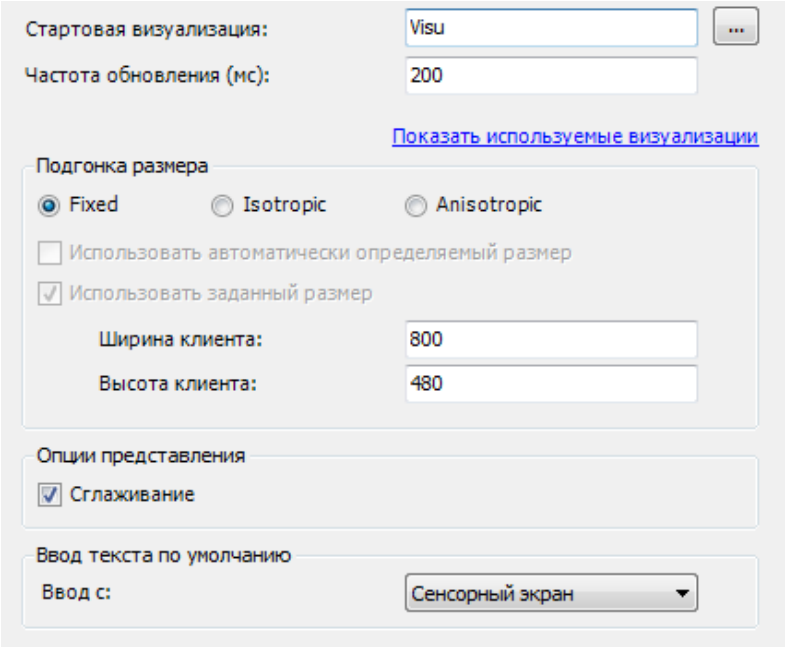


Рисунок 10.130 – Настройки таргет-визуализации

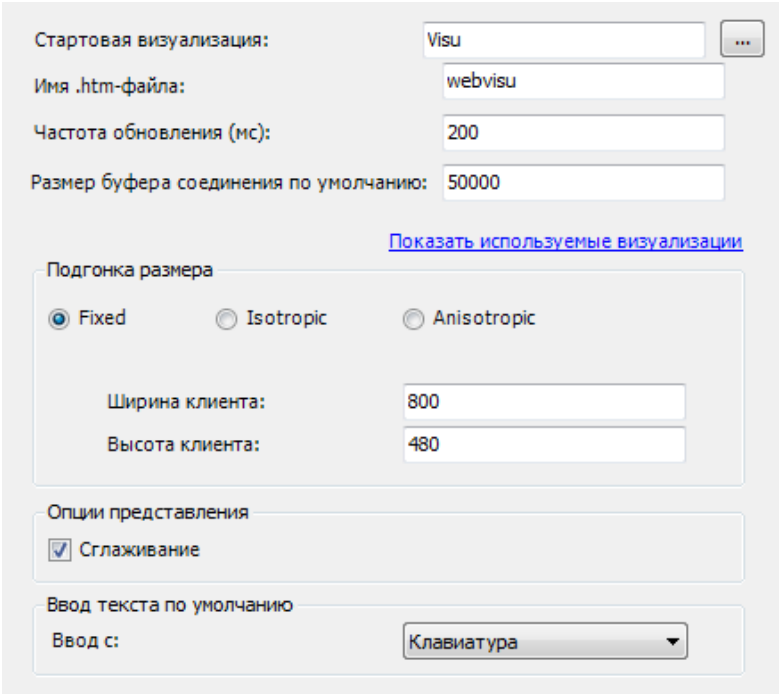


Рисунок 10.131 – Настройки web-визуализации

7. Запустить проект на виртуальном контроллере. Открыть в веб-браузере страницу http://localhost:8080/webvisu.htm, чтобы увидеть web-визуализацию проекта. ***Следует обратить внимание***, что браузер должен поддерживать **HTML5**. Проверить функционал проекта.

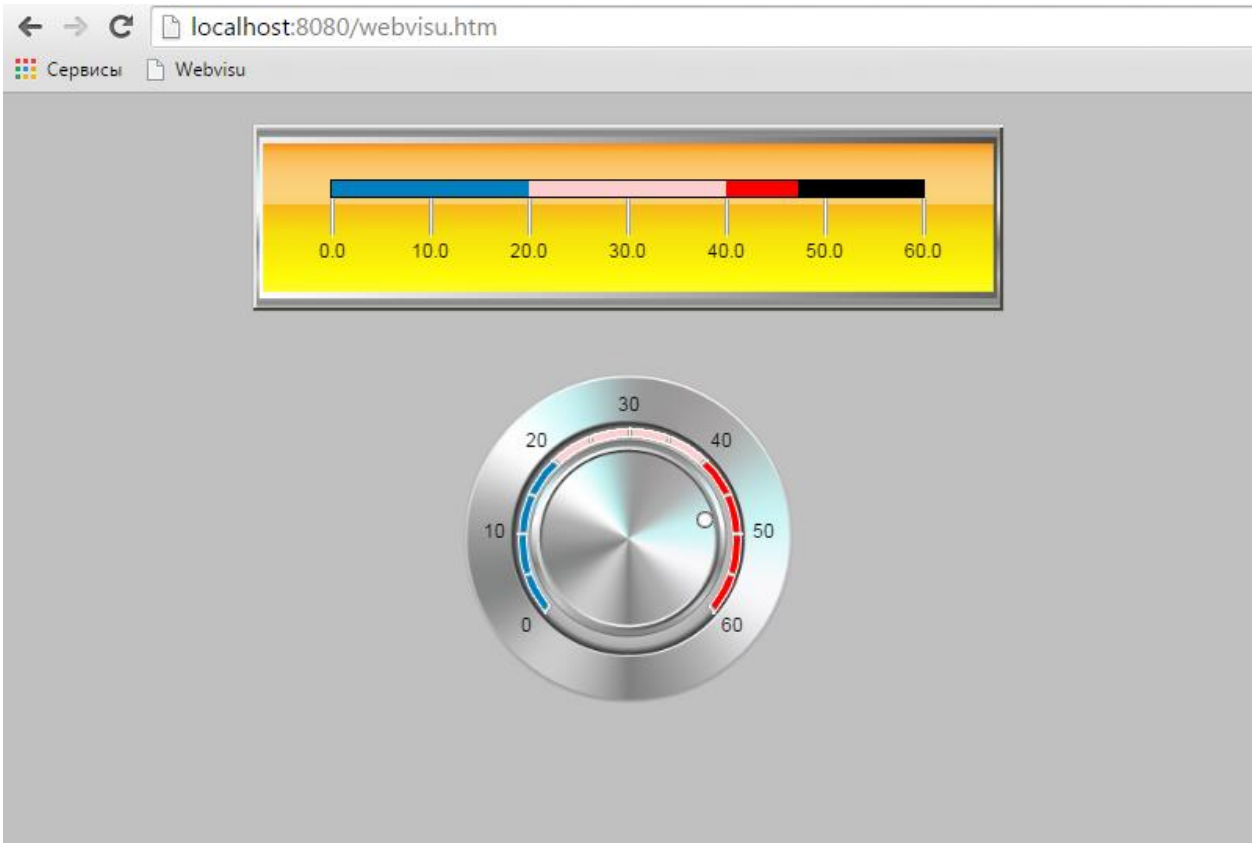


Рисунок 10.132 – Web-визуализация проекта в браузере Chrome

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**.

10.2.7 Гистограмма

Данный пример посвящен работе с графическим примитивом Гистограмма, используемым для отображения в графическом виде одномерных массивов значений.

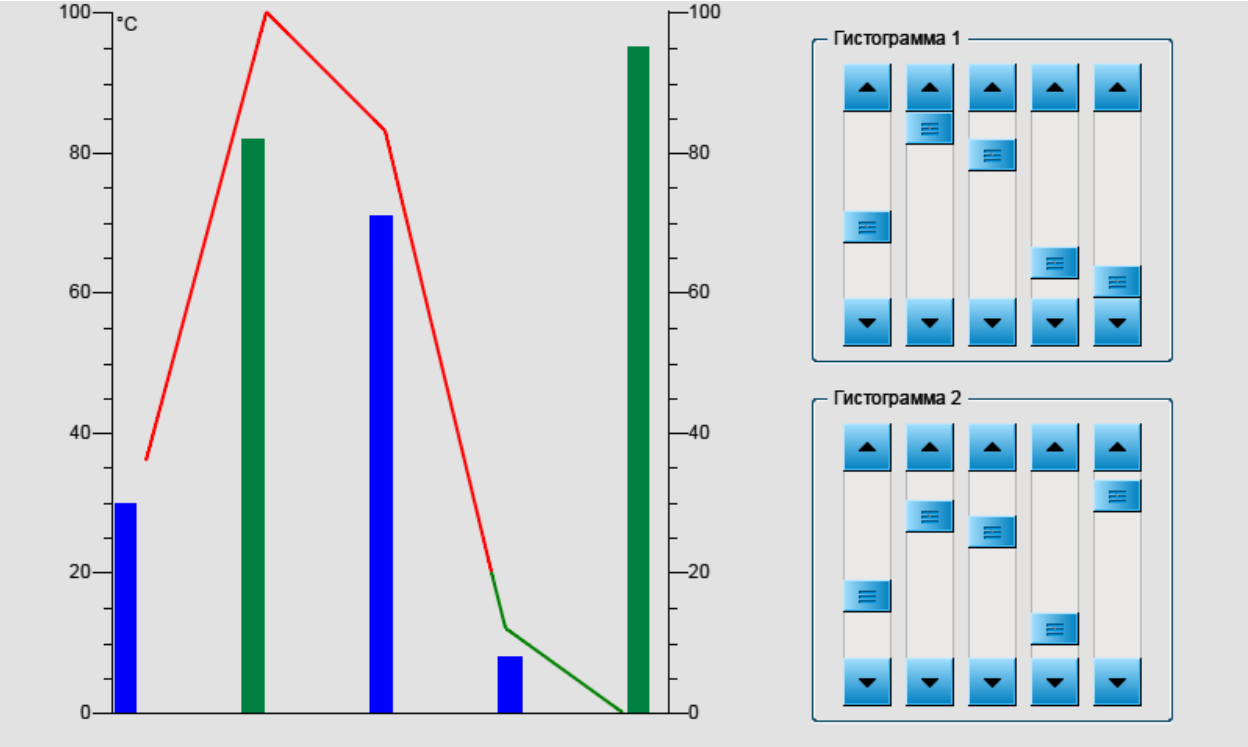


Рисунок 10.133 – Внешний вид примера Гистограмма

Пример создан в среде **CODESYS V3.5 SP11 Patch 5** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с таргет-файлом **3.5.11.5**.

Пример доступен для скачивания: Example_Histogram.projectarchive

Для создания примера с использованием элемента **Гистограмма** следует:

1. Создать новый ***стандартный*** проект **CODESYS** с названием **Example_Histogram** и настройками по умолчанию: таргет – CODESYS Control Win V3, язык PLC_PRG – **ST**.
2. В программе **PLC_PRG** объявить 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
```

Рисунок 10.134 – Объявление переменных программы PLC_PRG

3. Добавить в проект экран визуализации **Visualization**. В его свойствах выбрать размер **800 × 480.** Экран будет содержать два элемента Гистограмма (наложенных друг на друга), два элемента Группа и десять элементов Полоса прокрутки.

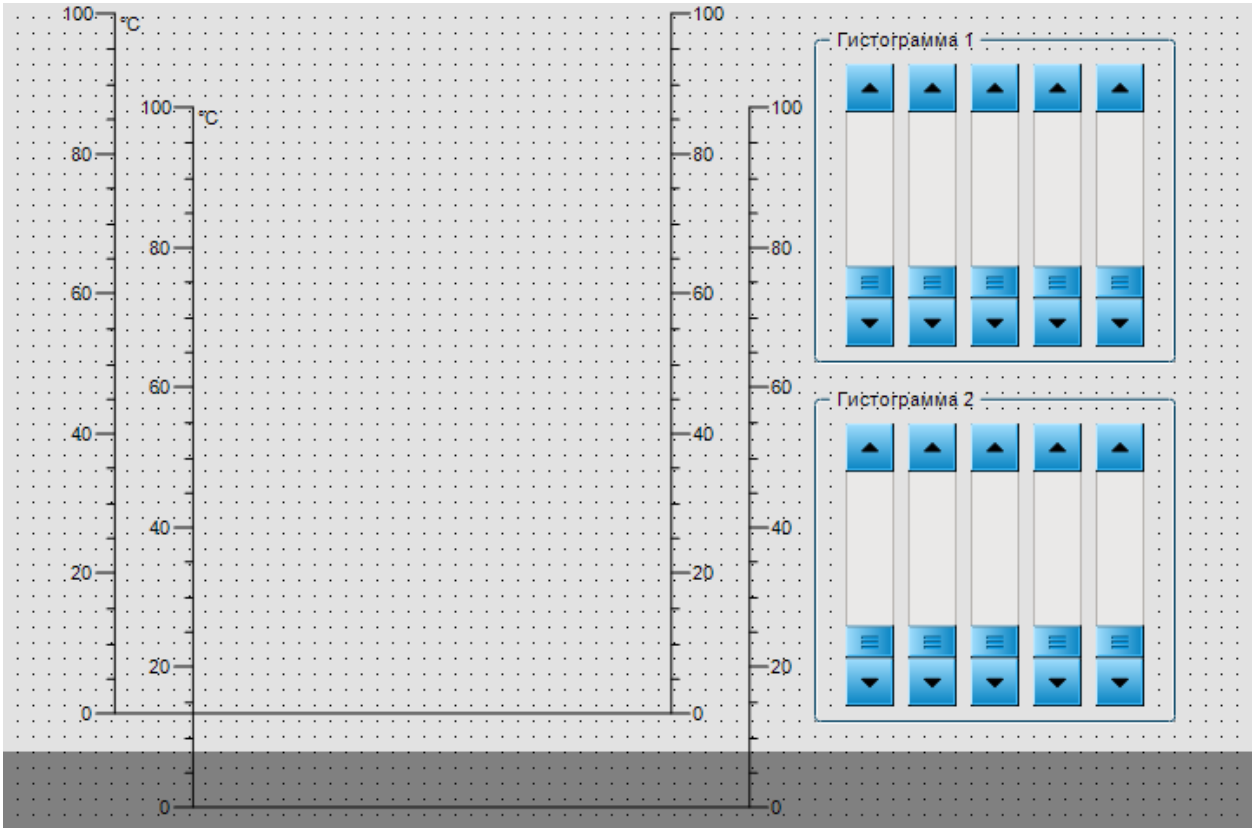


Рисунок 10.135 – Содержание экрана Visualization (гистограммы еще не наложены друг на друга)

4. Настроить элементы Гистограмма **1** и Гистограмма **2**:

Свойства

Фильтр | Сортировать по | Порядок сортировки

Эксперт

Свойство	Значения
Тип элемента	Гистограмма
Массив данных	PLC_PRG.arrHist1
⊞ Поддиапазон массива	
Тип отображения	Кривая
Ширина линии	3
Показывать горизонтальные линии	☐
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 (c-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...	☑
⊞ Цветовые области	Создать новый

Рисунок 10.136 – Параметры элемента Гистограмма 1

Вторая гистограмма отличается от первой только привязываемой переменной, типом отображения и условием тревоги:

Свойства

Фильтр | Сортировать по | Порядок сортировки

Эксперт

Свойство	Значения
Имя элемента	GenElemInst_137
Тип элемента	Гистограмма
Массив данных	PLC_PRG.arrHist2
⊞ Поддиапазон массива	
Тип отображения	Линейки
Показывать горизонтальные линии	☐
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...	☑
⊞ Цветовые области	Создать новый

Рисунок 10.137 – Параметры элемента Гистограмма 2

После окончания настройки элементов, их следует наложить друг на друга.

5. Настроить элементы Полоса прокрутки:

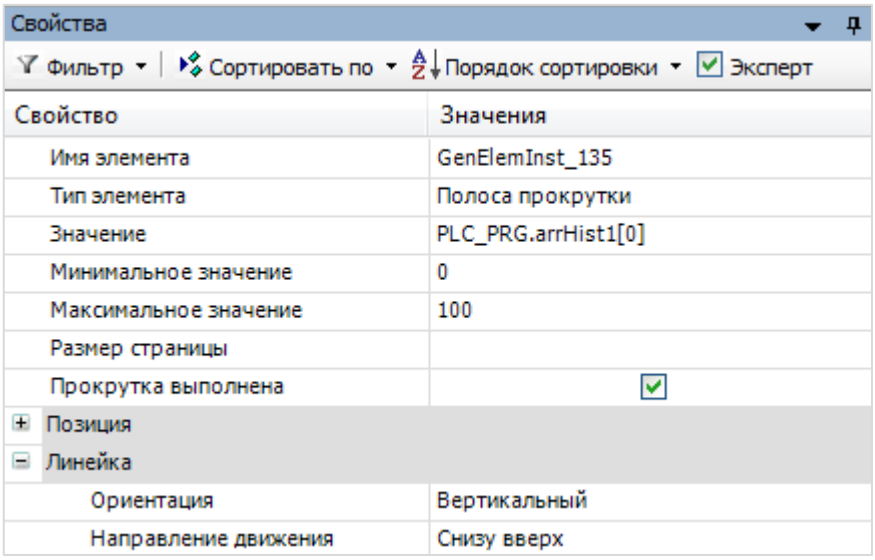


Рисунок 10.138 – Параметры элемента Полоса прокрутки 1 (группа Гистограмма 1)

Остальные элементы будут отличаться только привязываемыми переменными.

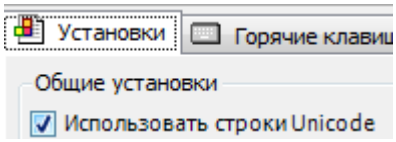
Для группы **Гистограмма 1** это, соответственно, будут:

PLC_PRG.arrHist1[0]
PLC_PRG.arrHist1[1]
PLC_PRG.arrHist1[2]
PLC_PRG.arrHist1[3]
PLC_PRG.arrHist1[4]

Для группы Гистограмма 2:

PLC_PRG.arrHist2[0]
PLC_PRG.arrHist2[1]
PLC_PRG.arrHist2[2]
PLC_PRG.arrHist2[3]
PLC_PRG.arrHist2[4]

6. Настроить компонент Менеджер визуализации. В его установках поставить галочку **Использовать строки Unicode**:



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

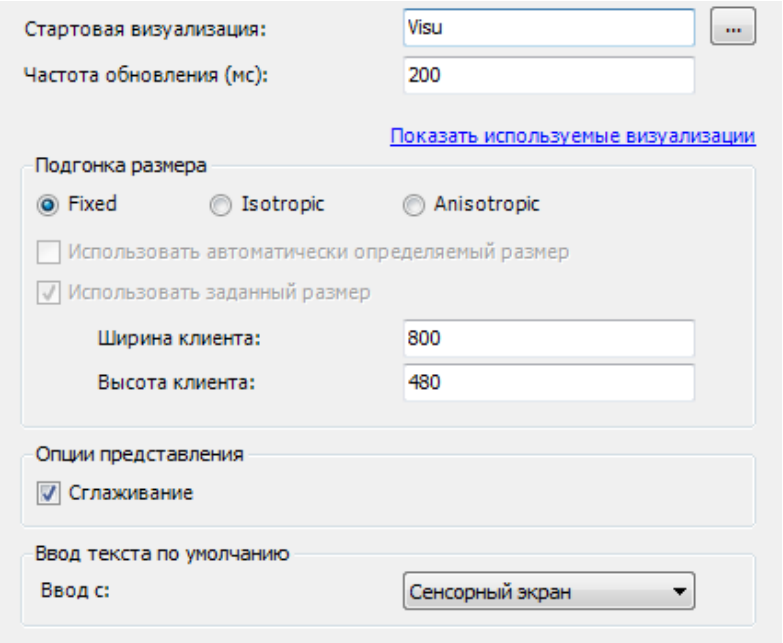


Рисунок 10.139 – Настройки таргет-визуализации

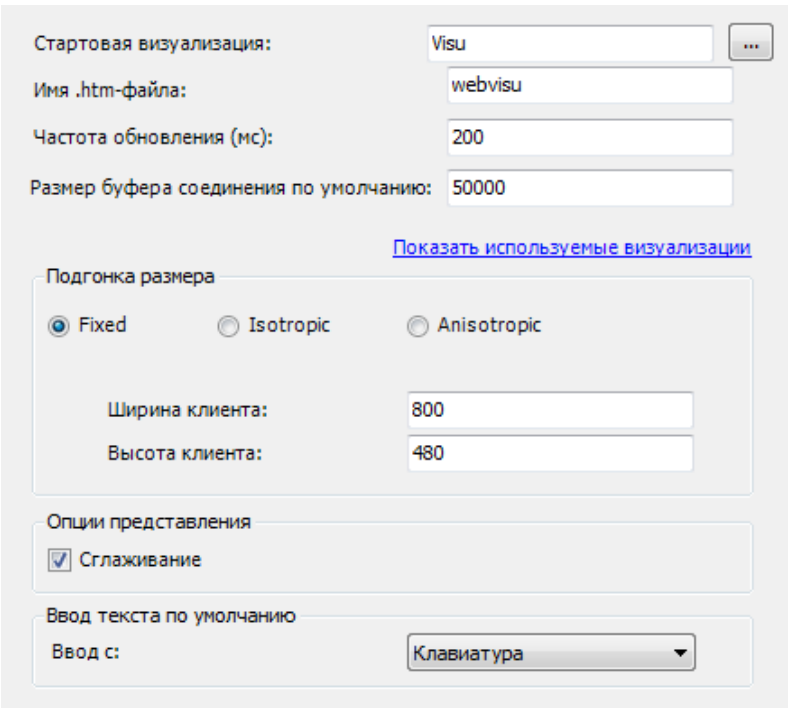


Рисунок 10.140 – Настройки web-визуализации

7. Запустить проект на виртуальном контроллере. Открыть в веб-браузере страницу http://localhost:8080/webvisu.htm, чтобы увидеть web-визуализацию проекта. ***Следует обратить внимание***, что браузер должен поддерживать **HTML5**. Проверить функционал проекта.

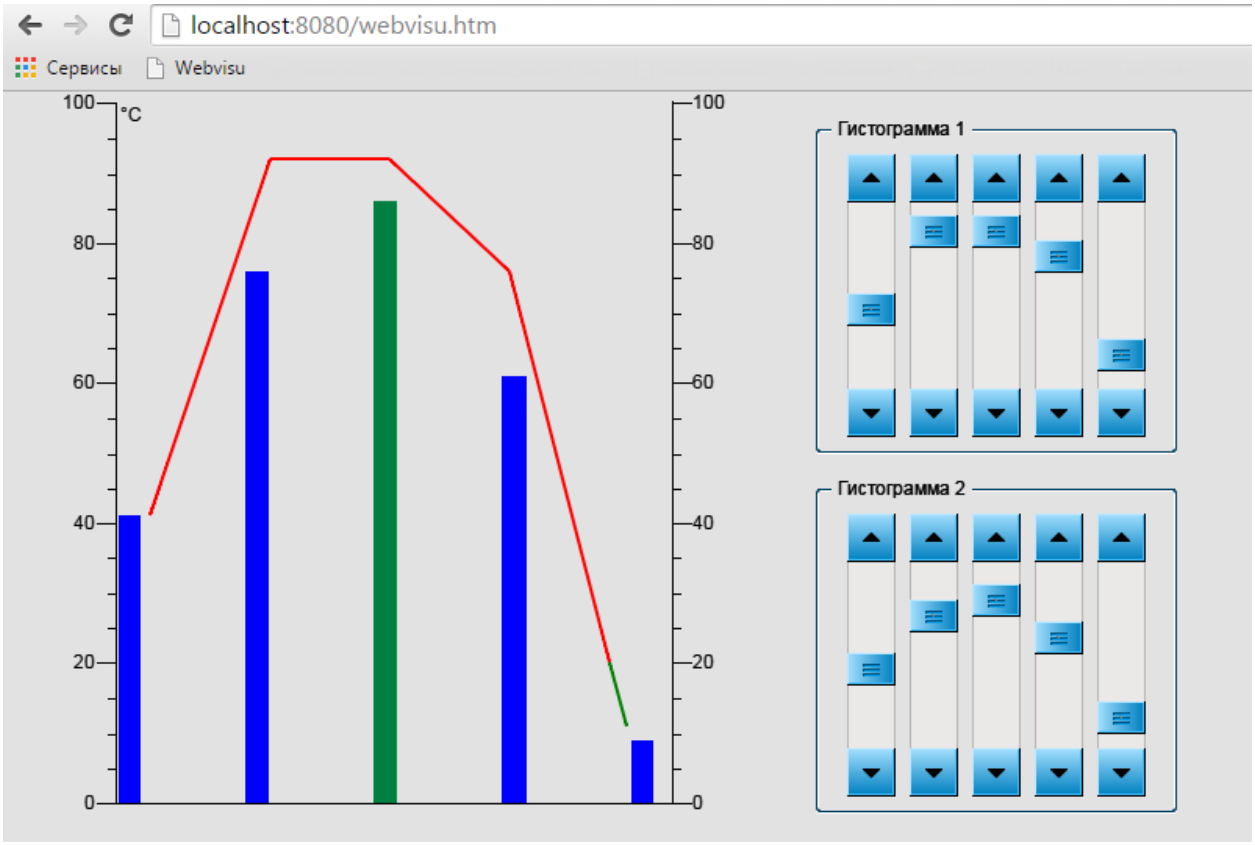


Рисунок 10.141 – Web-визуализация проекта в браузере Chrome

Меняя значение элементов массива с помощью элементов **Полоса прокрутки**, можно наблюдать соответствующие изменения в элементе **Гистограмма**.

Для **Гистограммы 1** значения, превышающие 80, являются тревожными – столбец окрашивается в зеленый цвет.

Для **Гистограммы 2** тревожными являются значения менее 20-ти – линия графика окрашивается в зеленый.

10.2.8 Индикаторы/Переключатели/Изображения

Данный пример посвящен работе с элементами Индикатор, Переключатель и Переключатель изображения, предназначенными для отображения и переключения переменных типа **BOOL**.

Рисунок 10.142 – Внешний вид примера Индикаторы/Переключатели/Изображения

Пример создан в среде **CODESYS V3.5 SP11 Patch 5** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с таргет-файлом **3.5.11.5**.

Пример доступен для скачивания: Example_Lamps.projectarchive

Для работы примера с элементами **Индикаторы/Переключатели/Изображения** следует:

1. Создать новый ***стандартный*** проект **CODESYS** с названием **Example_Lamps** и настройками по умолчанию: таргет – CODESYS Control Win V3, язык PLC_PRG – **ST**.

2. В программе **PLC_PRG** объявить 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
```

Рисунок 10.143 – Объявление переменных программы PLC_PRG

3. Добавить в проект экран визуализации **Visualization**. В его свойствах выбрать размер **800 × 480.** Экран будет содержать пять элементов Индикатор, один элемент Переключатель изображения и шесть элементов Переключатель/Выключатель (разных типов).

Рисунок 10.144 – Содержание экрана Visualization

4. Настроить элементы Индикаторы и Переключатели согласно таблице 11.2:

Таблица 10.2 – 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

Продолжение Табл. 10.2

№ пп.	Тип элемента	Внешний вид элемента	Переменная	Фон
9.	Нажимной выключатель LED		PLC_PRG.bVar4	Blue
10.	Клавишный выключатель		PLC_PRG.bVar5	Gray
11.	Вращающийся выключатель		PLC_PRG.bVar6	Gray

5. Настроить элемент Переключатель изображения:

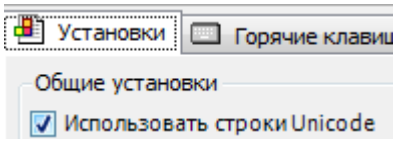
Свойства

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	☐
Прозрачный цвет	Black
Изотропный тип	Изотропия
Горизонтальное выравнивание	Лево
Вертикальное выравнивание	Верх
Поведение элемента	Переключатель изображения
Тексты	
Подсказка	
Переменные состояний	
Невидимый	
Отключение ввода	

Рисунок 10.145 – Параметры элемента Переключатель изображения

6. Настроить компонент Менеджер визуализации. В его установках поставить галочку **Использовать строки Unicode**:



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

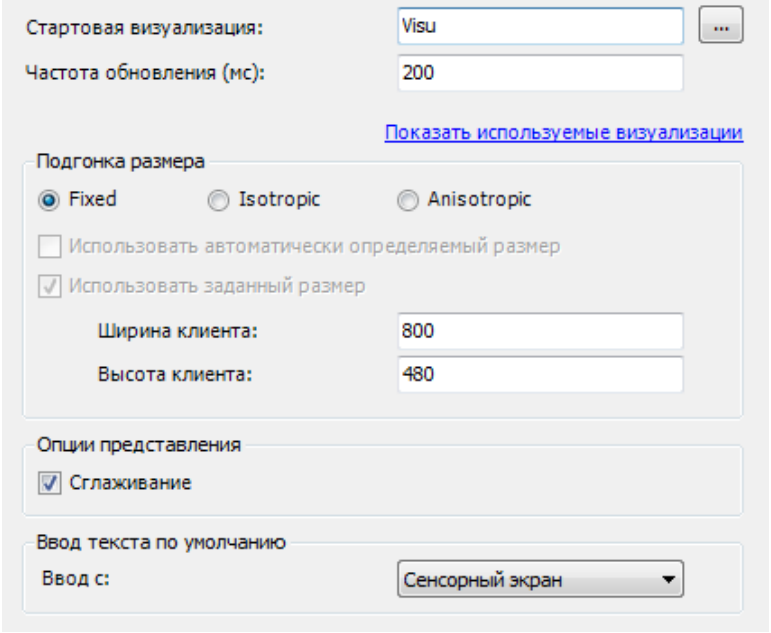


Рисунок 10.146 – Настройки таргет-визуализации

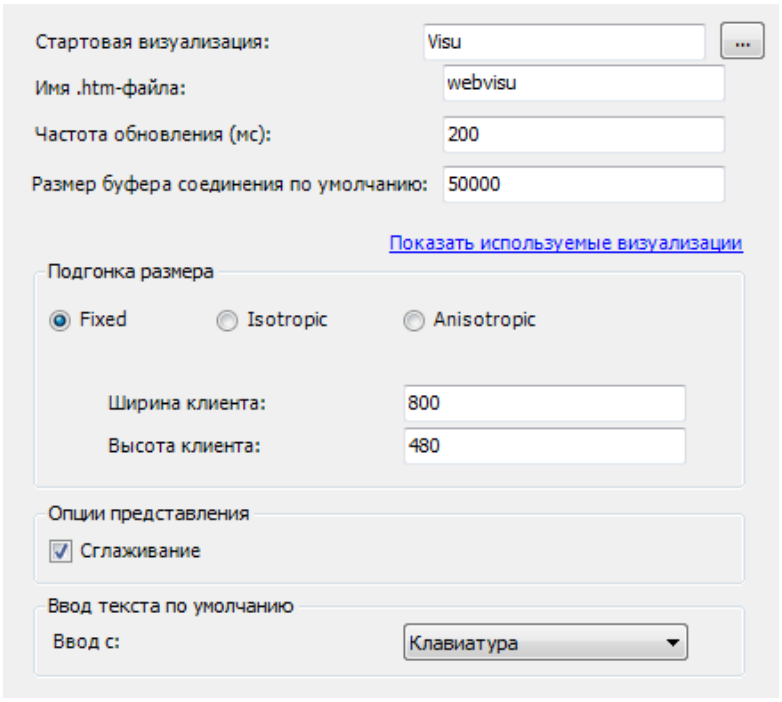


Рисунок 10.147 – Настройки web-визуализации

7. Запустить проект на виртуальном контроллере. Открыть в веб-браузере страницу http://localhost:8080/webvisu.htm, чтобы увидеть web-визуализацию проекта. ***Следует обратить внимание***, что браузер должен поддерживать **HTML5**. Проверить функционал проекта.

Рисунок 10.148 – Web-визуализация проекта в браузере Chrome

Нажатие на переключатели будет включать/отключать соответствующие лампы.

Лампа, сделанная на основе элемента **Переключатель изображения**, доступна для нажатия. На время зажатия она принимает третье («промежуточное») изображение.

10.2.9 Трассировка

Данный пример посвящен работе с элементом Трассировка, который используется для построения графика значения переменной в реальном времени.

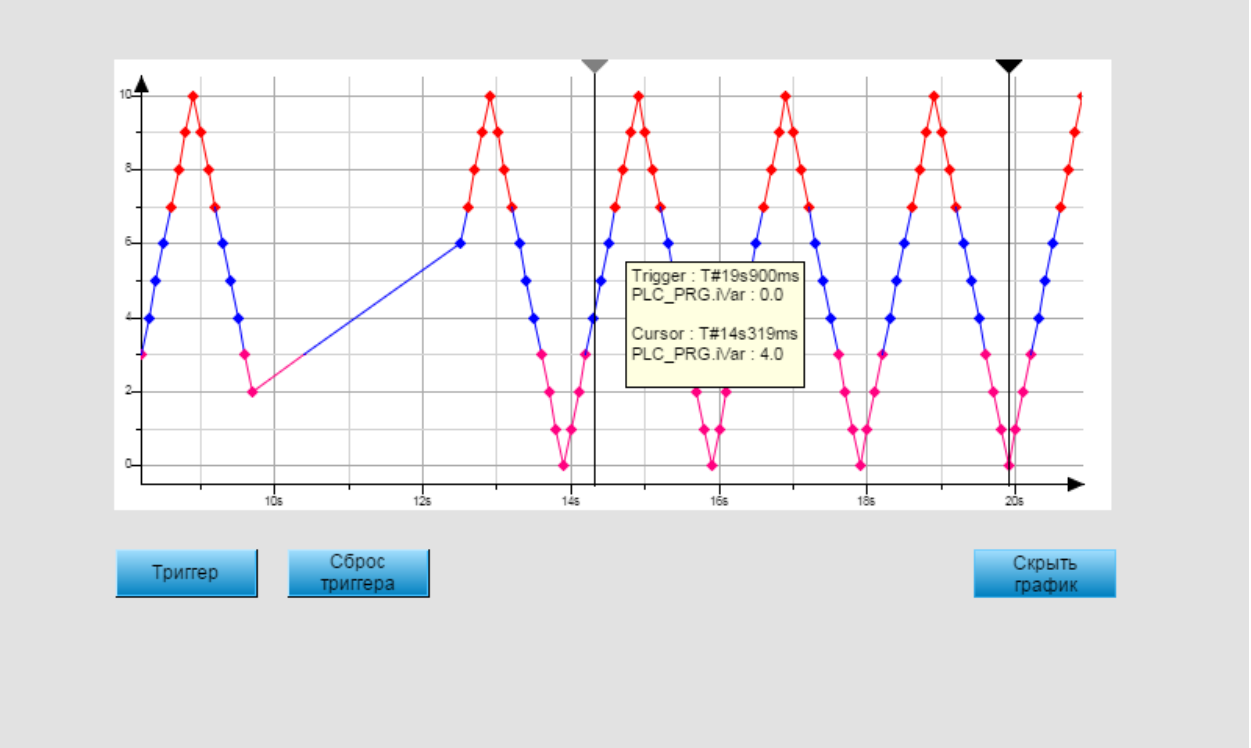


Рисунок 10.149 – Внешний вид примера Трассировка

Пример создан в среде **CODESYS V3.5 SP11 Patch 5** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с таргет-файлом **3.5.11.5**.

Пример доступен для скачивания: Example_Trace.projectarchive

Для создания примера с использованием элемента **Трассировка** следует:

1. Создать новый ***стандартный*** проект **CODESYS** с названием **Example_Trace** и настройками по умолчанию: таргет – CODESYS Control Win V3, язык PLC_PRG – **ST**.
2. В программе **PLC_PRG** объявить следующие переменные:

PLC_PRG

```
PROGRAM PLC_PRG
VAR
    bDir:           BOOL;           // направление изменения iVar (FALSE - приращение, TRUE - убывание)
    iVar:           INT;            // переменная, отображаемая трассировкой

    bTrigger:       BOOL;           // триггер остановки трассировки
    bResetTrigger:  BOOL;           // переменная сброса триггера
    bVisible:       BOOL    :=  TRUE;  // переменная видимости графика
END_VAR
```

Рисунок 10.150 – Объявление переменных программы PLC_PRG

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

```
// эмуляция изменения значений

IF  NOT(bDir) THEN
    iVar := iVar + 1;
ELSE
    iVar := iVar - 1;
END_IF

IF iVar = 0  THEN
    bDir := FALSE;
ELSIF iVar = 10 THEN
    bDir := TRUE;
END_IF
```

Рисунок 10.151 – Код программы PLC_PRG

4. Добавить в проект экран визуализации **Visualization**. В его свойствах выбрать размер **800 × 480.** Экран будет содержать элемент Трассировка и три элемента Кнопка.

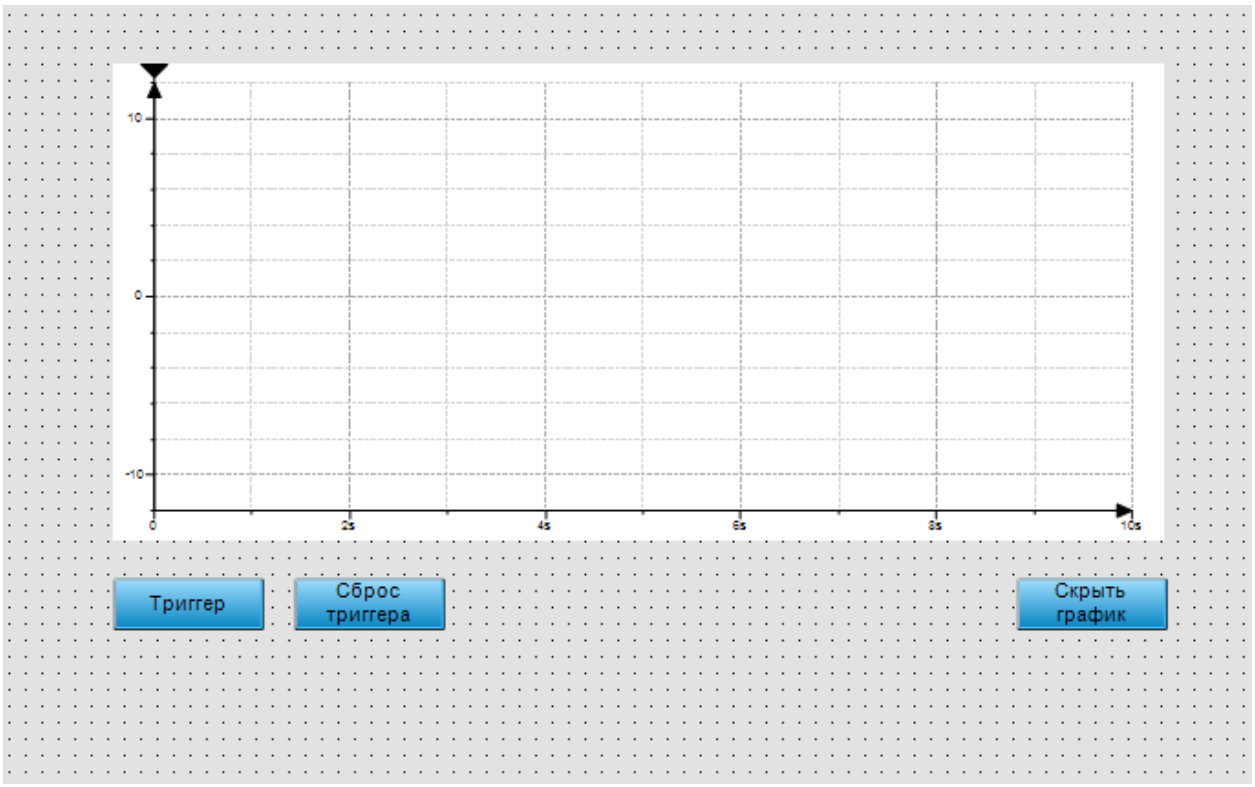


Рисунок 10.152 – Содержания экрана Visualization

5. Настроить элемент Трассировка:

Свойства

Фильтр | Сортировать по | Порядок сортировки

Эксперт

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	☑
Заменить существующую т...	☑
Управляющие переменные	
Сброс триггера	PLC_PRG.bResetTrigger
Запустить трассировку	
Остановить трассировку	
Сохранить трассировку...	
Загрузить трассировку...	

Рисунок 10.153 – Параметры элемента Трассировка

Двойным нажатием на значение параметра **Трассировка** (**Visualization_Trace1**) открыть меню конфигурации трассировки и настроить конфигурацию следующим образом:

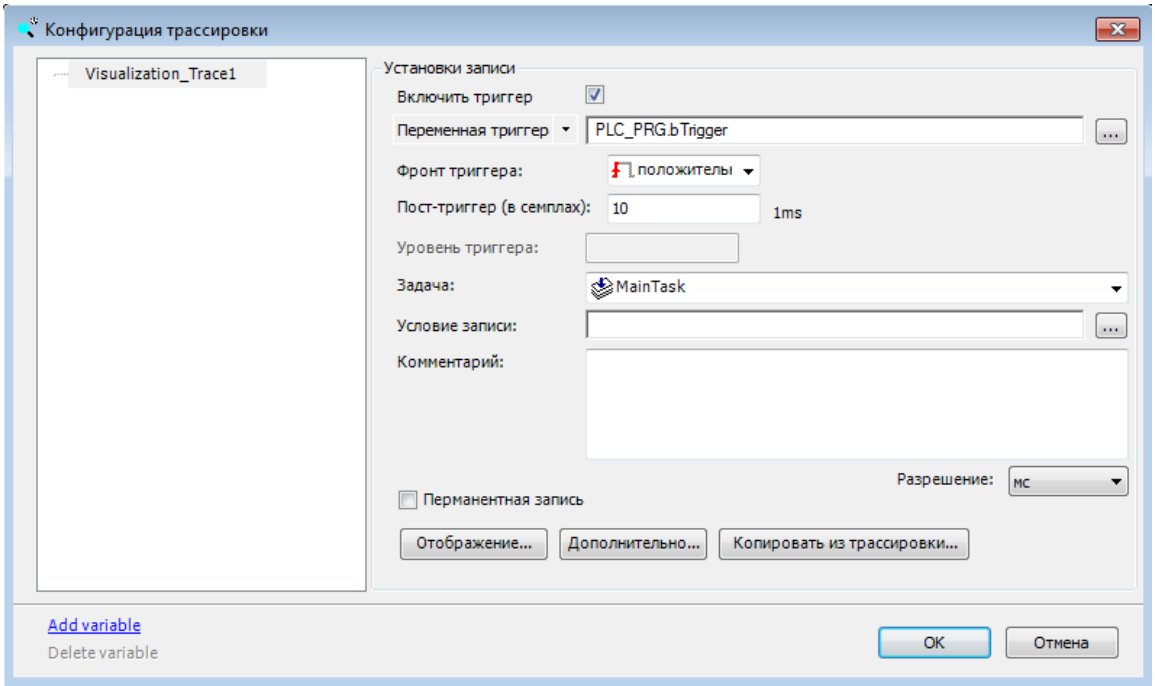


Рисунок 10.154 – Конфигурация трассировки

Затем следует нажать на кнопку **Доб. переменную**, чтобы привязать к элементу переменную, которая будет отображаться на графике:

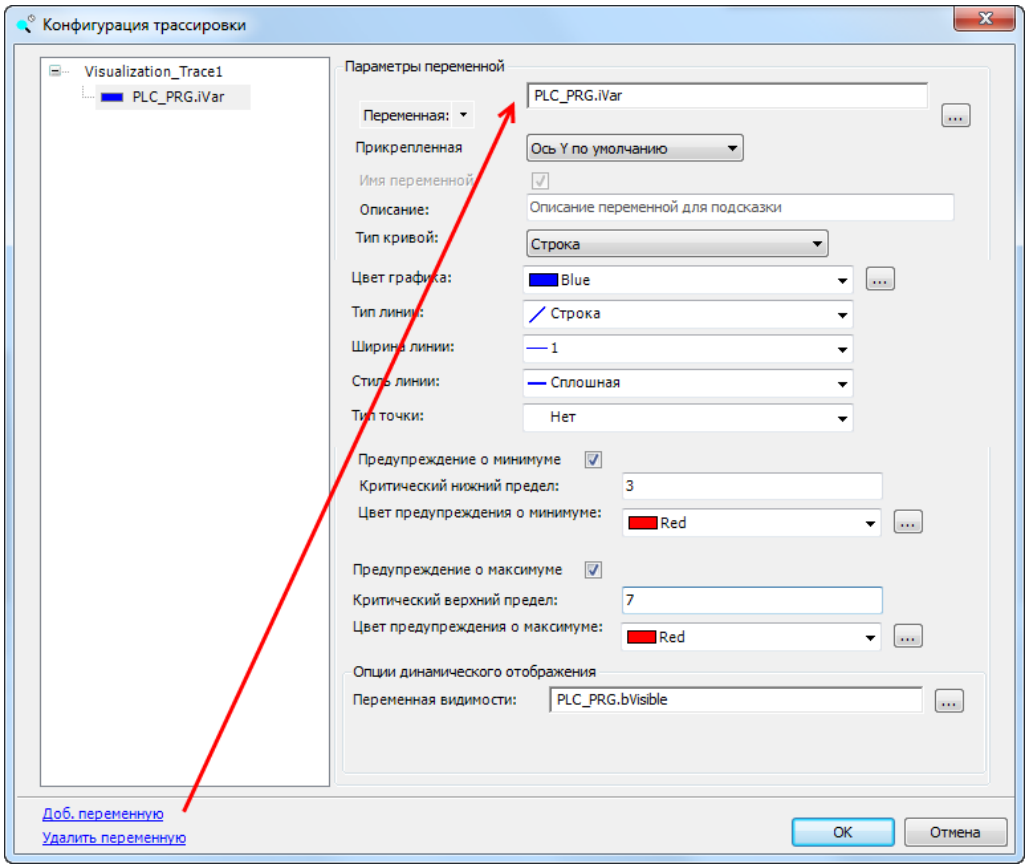


Рисунок 10.155 – Привязка переменной к трассировке

6. К кнопке **Триггер** во вкладке InputConfiguration к параметру **OnMouseClick** привязать действие Выполнить ST-код:

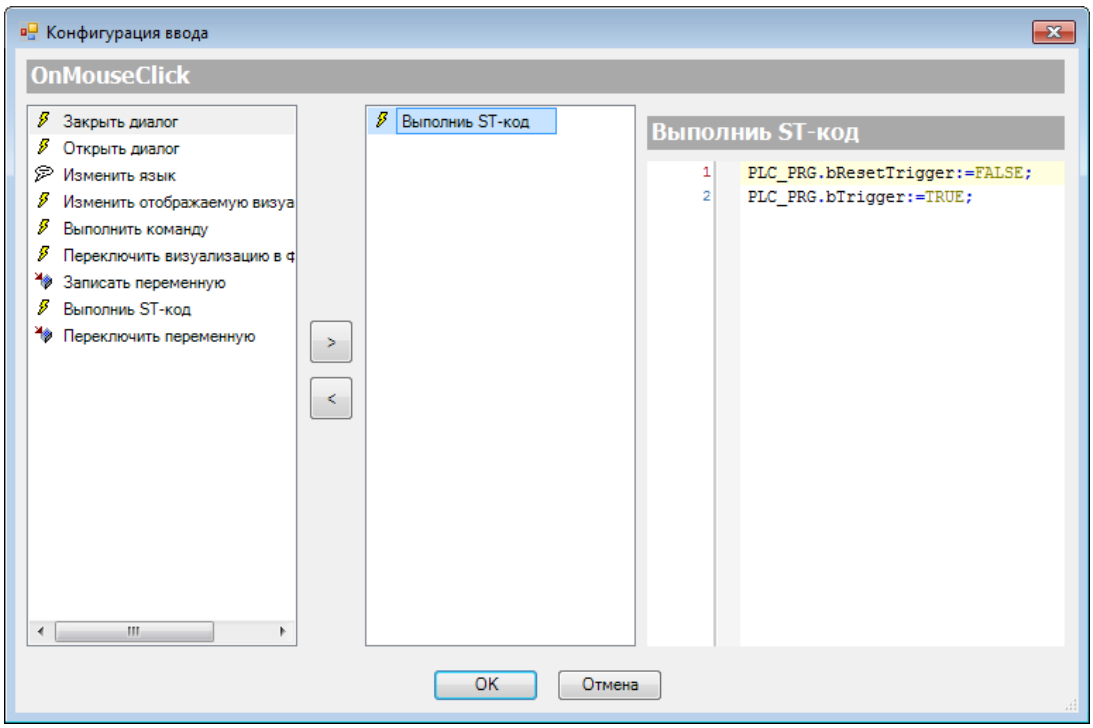


Рисунок 10.156 – ST-код кнопки Триггер

7. К кнопке **Сброс триггера** во вкладке InputConfiguration к параметру **OnMouseClick** привязать действие Выполнить ST-код:

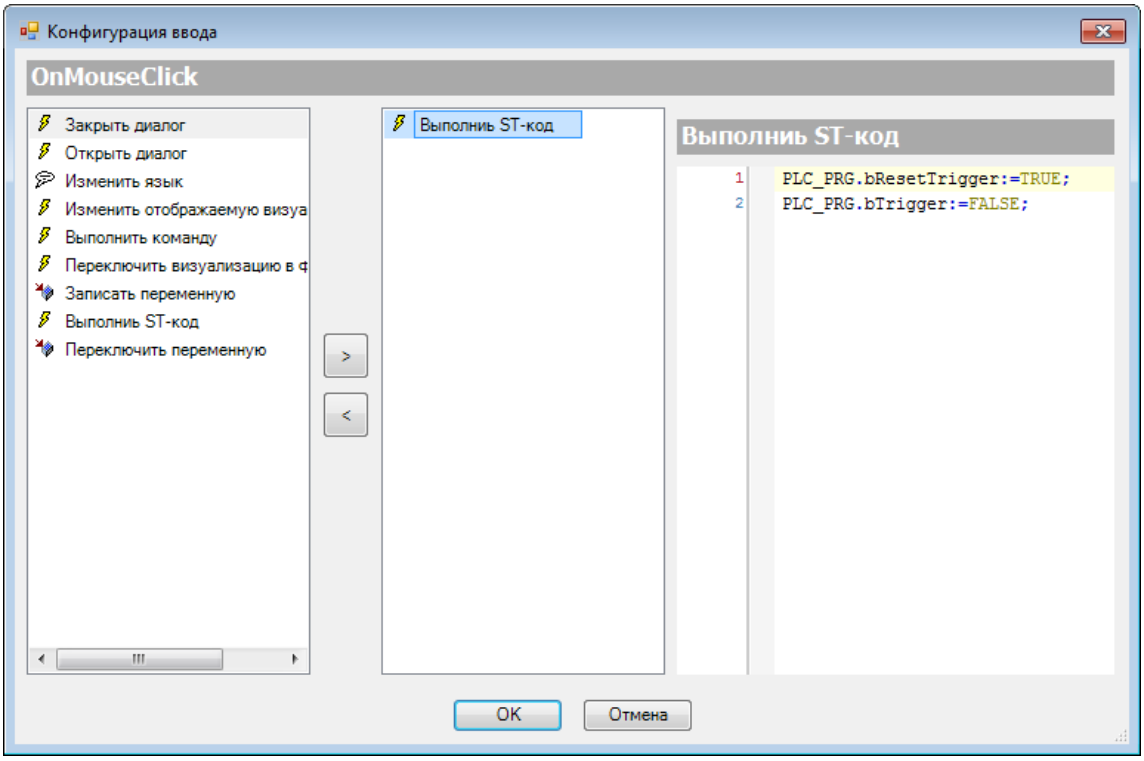


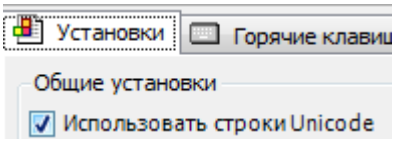
Рисунок 10.157 – ST-код кнопки Сброс триггера

8. К кнопке **Скрыть график** привязать переменную **bVisible**:

Inputconfiguration	
OnDialogClosed	Конфигурация...
OnMouseClick	Конфигурация...
OnMouseDown	Конфигурация...
OnMouseEnter	Конфигурация...
OnMouseLeave	Конфигурация...
OnMouseMove	Конфигурация...
OnMouseUp	Конфигурация...
Нажать	
Переключить	
Переменная	PLC_PRG.bVisible
Переключить навер...	☐
Горячая клавиша	

Рисунок 10.158 – Параметры кнопки Скрыть график

9. Настроить компонент Менеджер визуализации. В его установках поставить галочку **Использовать строки Unicode**:



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

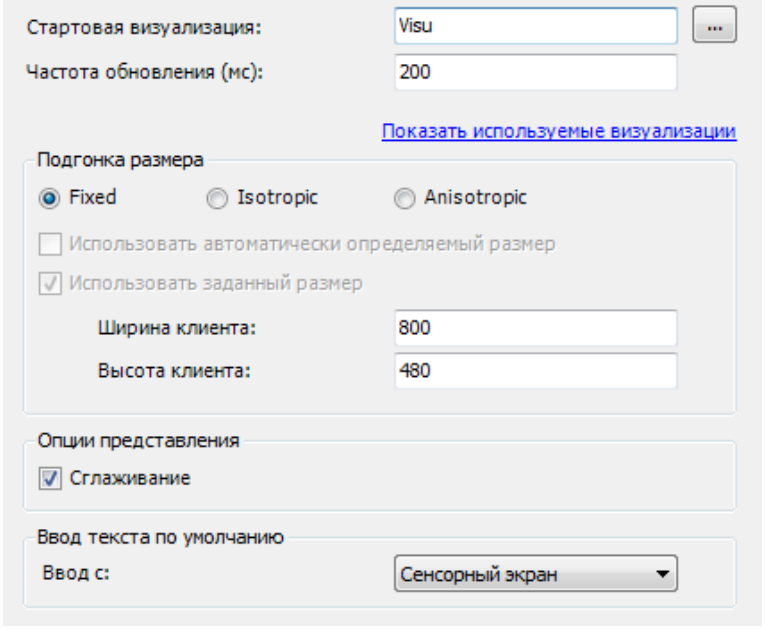


Рисунок 10.159 – Настройки таргет-визуализации

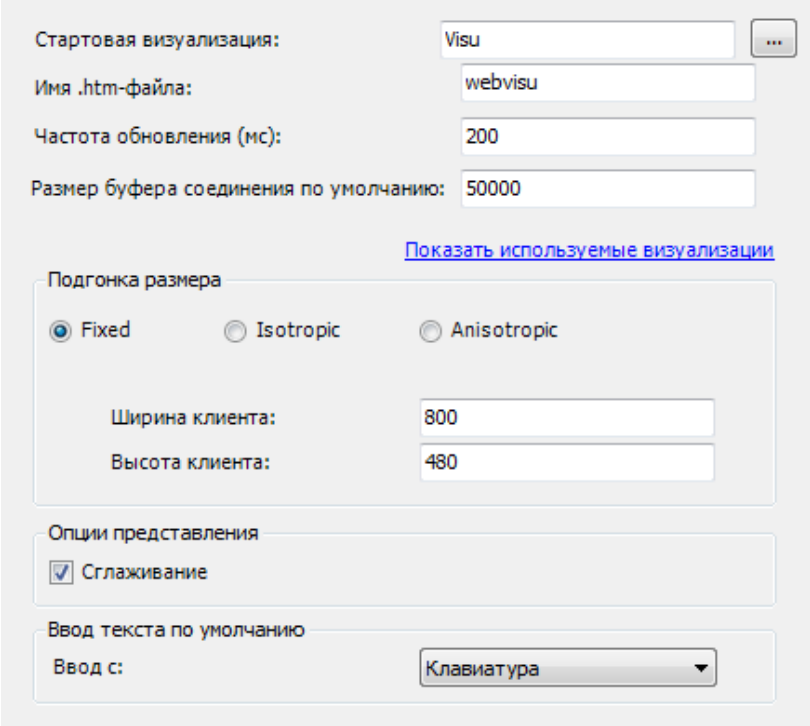


Рисунок 10.160 – Настройки web-визуализации

10. Запустить проект на виртуальном контроллере. Открыть в веб-браузере страницу http://localhost:8080/webvisu.htm, чтобы увидеть web-визуализацию проекта. ***Следует обратить внимание***, что браузер должен поддерживать **HTML5**. Проверить функционал проекта.

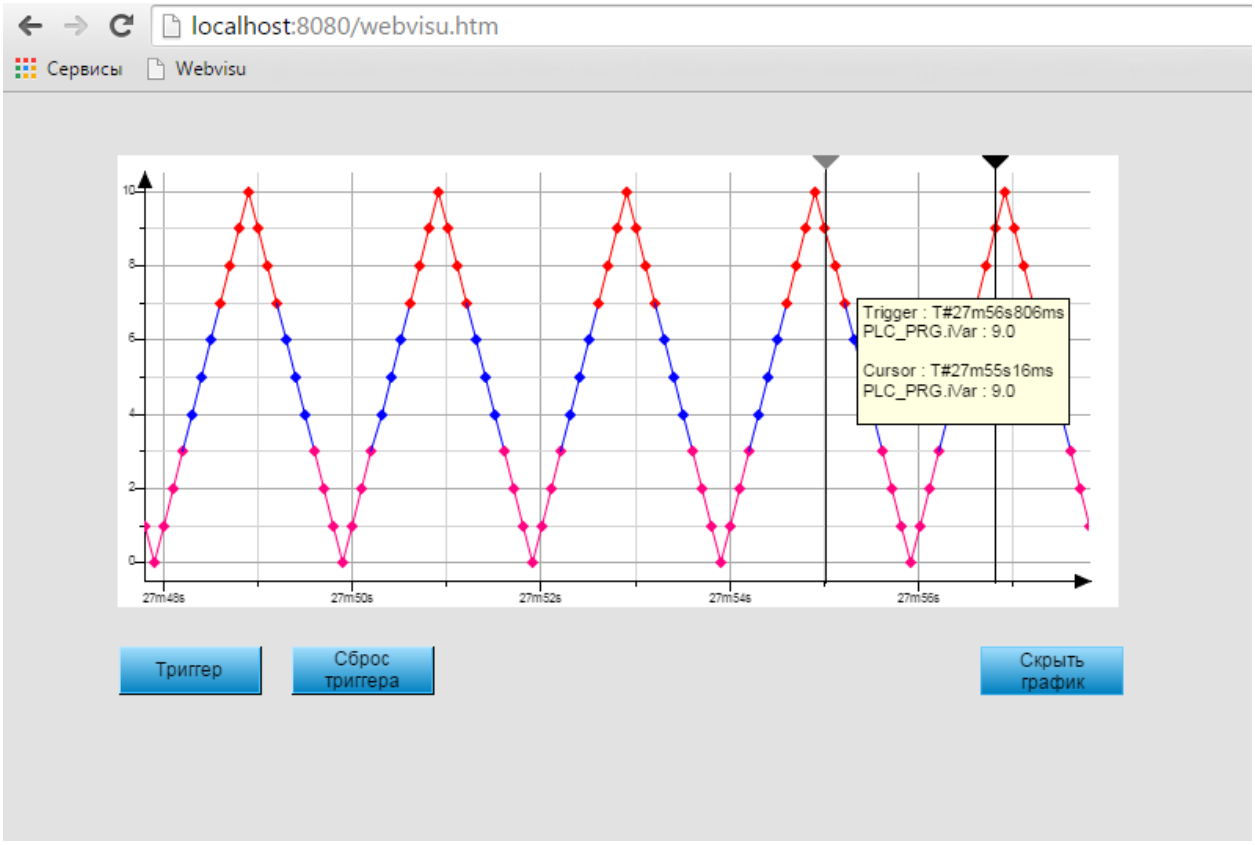


Рисунок 10.161 – Web-визуализация проекта в браузере Chrome

Нажатие на кнопку **Триггер** останавливает график. 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.

10.2.10 Тренд

Данный пример посвящен работе с графическим примитивом Тренд, который используется для построения графика значения переменной с возможностью просмотра истории.

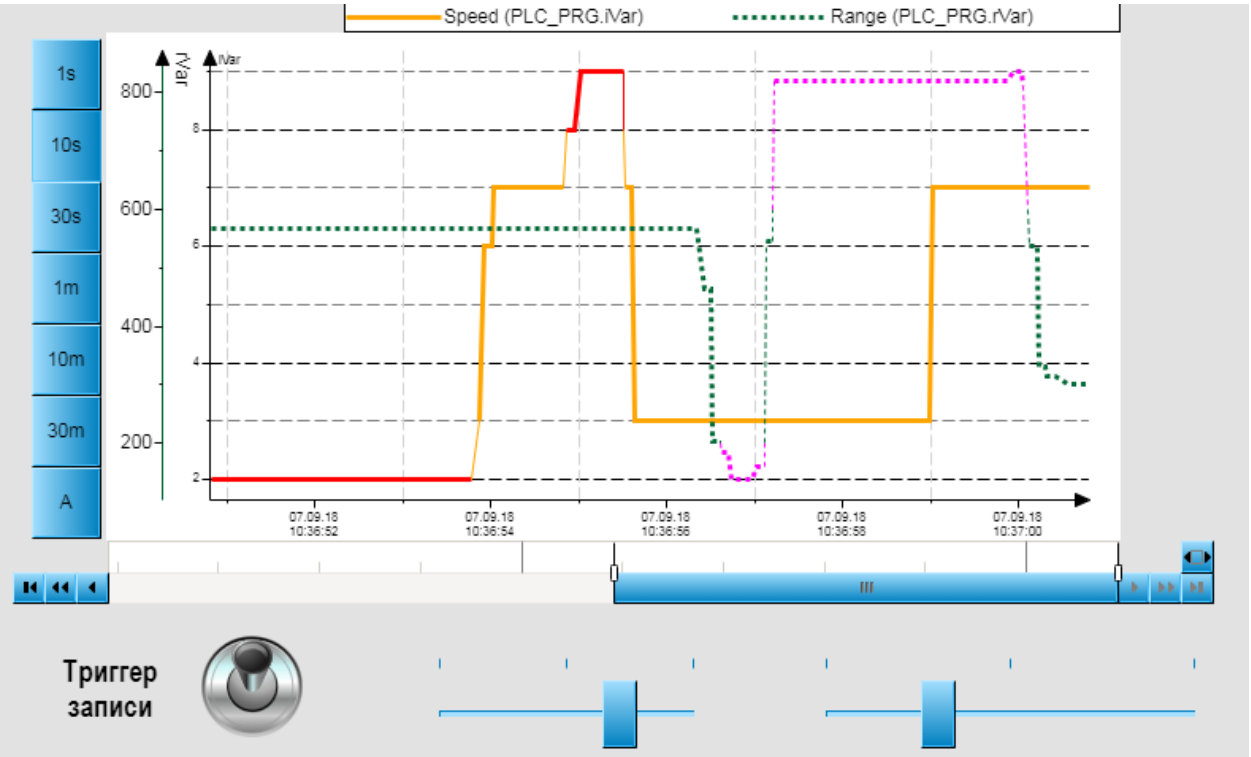


Рисунок 10.162 – Внешний вид примера Тренд

Пример создан в среде **CODESYS V3.5 SP11 Patch 5** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с таргет-файлом **3.5.11.5**.

Пример доступен для скачивания: Example_Trend.projectarchive

Продвинутый пример от компании **CODESYS Group** по работе с библиотекой тренда: TrendAPIExample.package
Файлы формата **.package** устанавливаются с помощью **Менеджера пакетов** (вкладка **Инструменты**) или (начиная с версии **CODESYS V3.5 SP17**) с помощью утилиты **CODESYS Installer**.

Для создания примера с использованием элемента **Тренд** следует:

1. Создать новый ***стандартный*** проект **CODESYS** с названием **Example_Trend** и настройками по умолчанию: таргет – CODESYS Control Win V3, язык PLC_PRG – **ST**.
2. В программе **PLC_PRG** объявить следующие переменные:

```
PLC_PRG
1  PROGRAM PLC_PRG
2  VAR
3
4      iVar:       INT;                // переменная 1, отображаемая трендом
5      rVar:       REAL;               // переменная 2, отображаемая трендом
6      bTrigger:   BOOL    :=  TRUE;   // триггер запуска/остановки записи
7
8  END_VAR
```

Рисунок 10.163 – Объявление переменных программы PLC_PRG

3. Добавить в проект экран визуализации **Visualization**. В его свойствах выбрать размер **800 × 480.** Добавить на экран элемент Тренд. Открывшееся окно **Конфигурация тренда** можно закрыть (его настройка будет проведена далее). Нажать **ПКМ** на элемент и выбрать команду **Параметры отображения тренда**. Переименовать существующую ось Y в **iVar**. Добавить вторую ось Y с названием **rVar**.

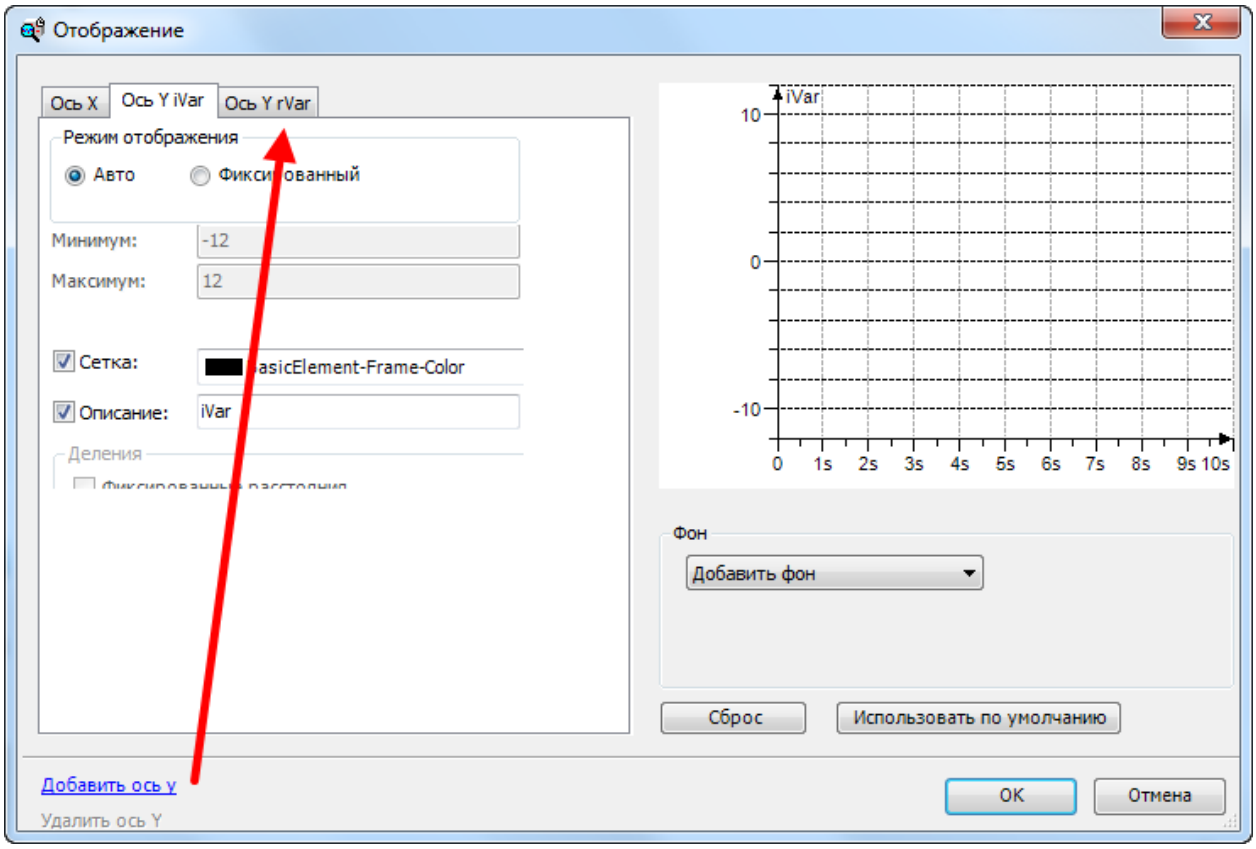


Рисунок 10.164 – Добавление дополнительной оси тренда

Нажать **ПКМ** на тренд и выбрать команду **Изменить запись тренда.** Откроется окно конфигурации тренда. Конфигурацию следует настроить в соответствие с рисунком 10.165. Затем следует нажать на кнопку **Доб. переменную**, чтобы привязать к элементу переменные, которые будут отображаться на графике:

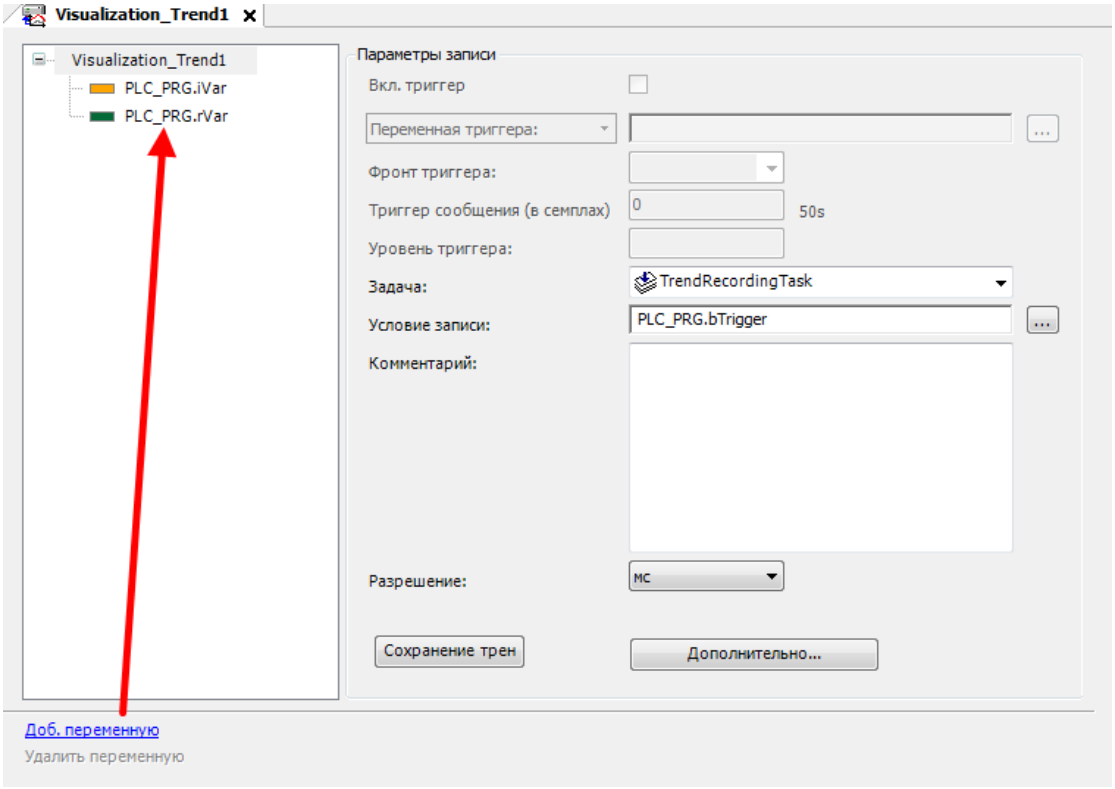


Рисунок 10.165 – Конфигурация тренда

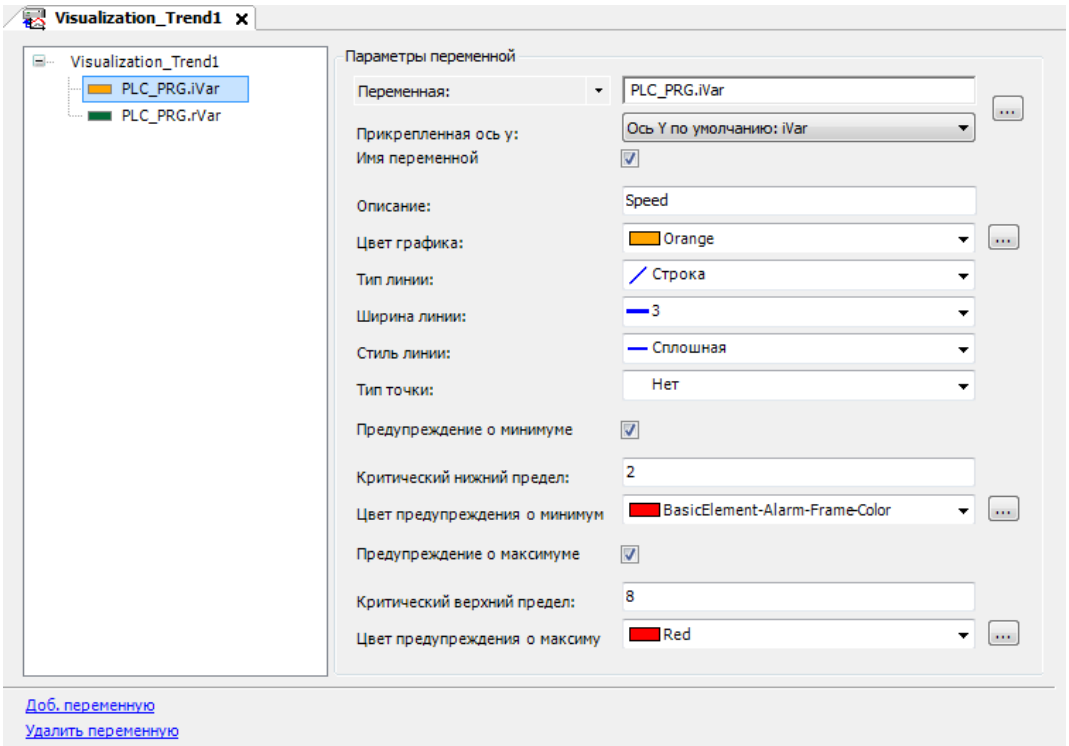


Рисунок 10.166 – Настройки переменной iVar

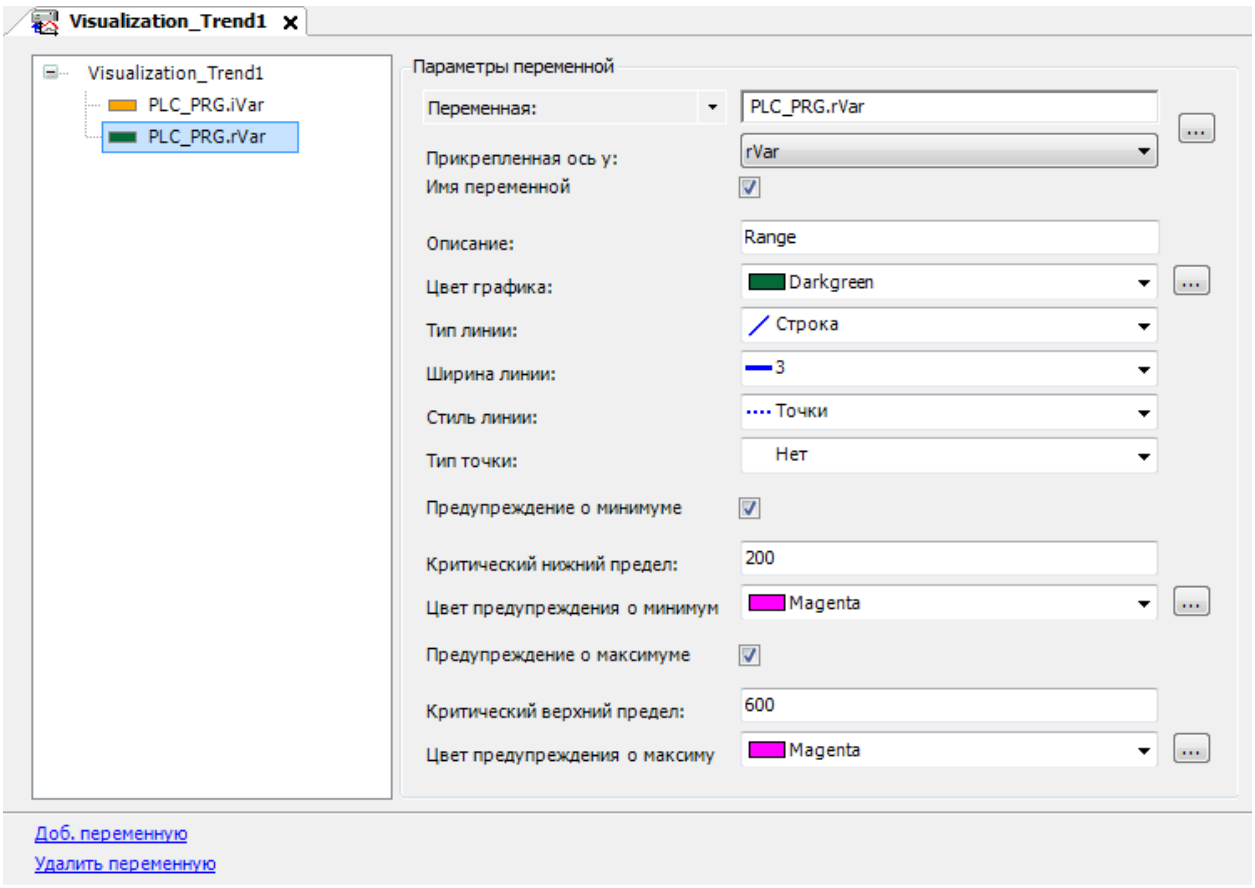


Рисунок 10.167 – Настройки переменной rVar

Тренд следует растянуть с помощью перетаскивания его опорных точек:

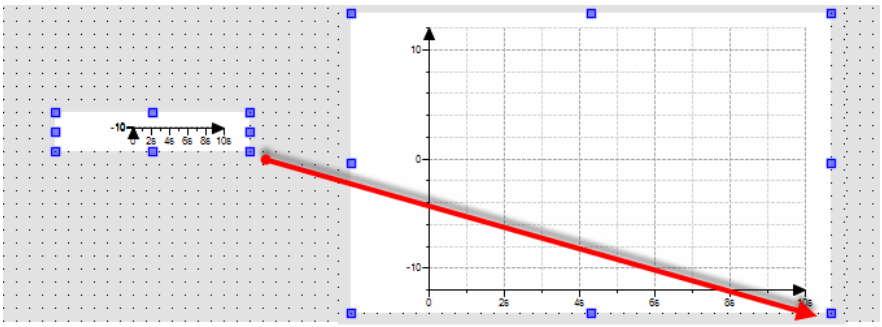


Рисунок 10.168 – Изменение размера тренда

Затем следует нажать **ПКМ** на элемент и выбрать команду **Вставить элементы управления трендом**:

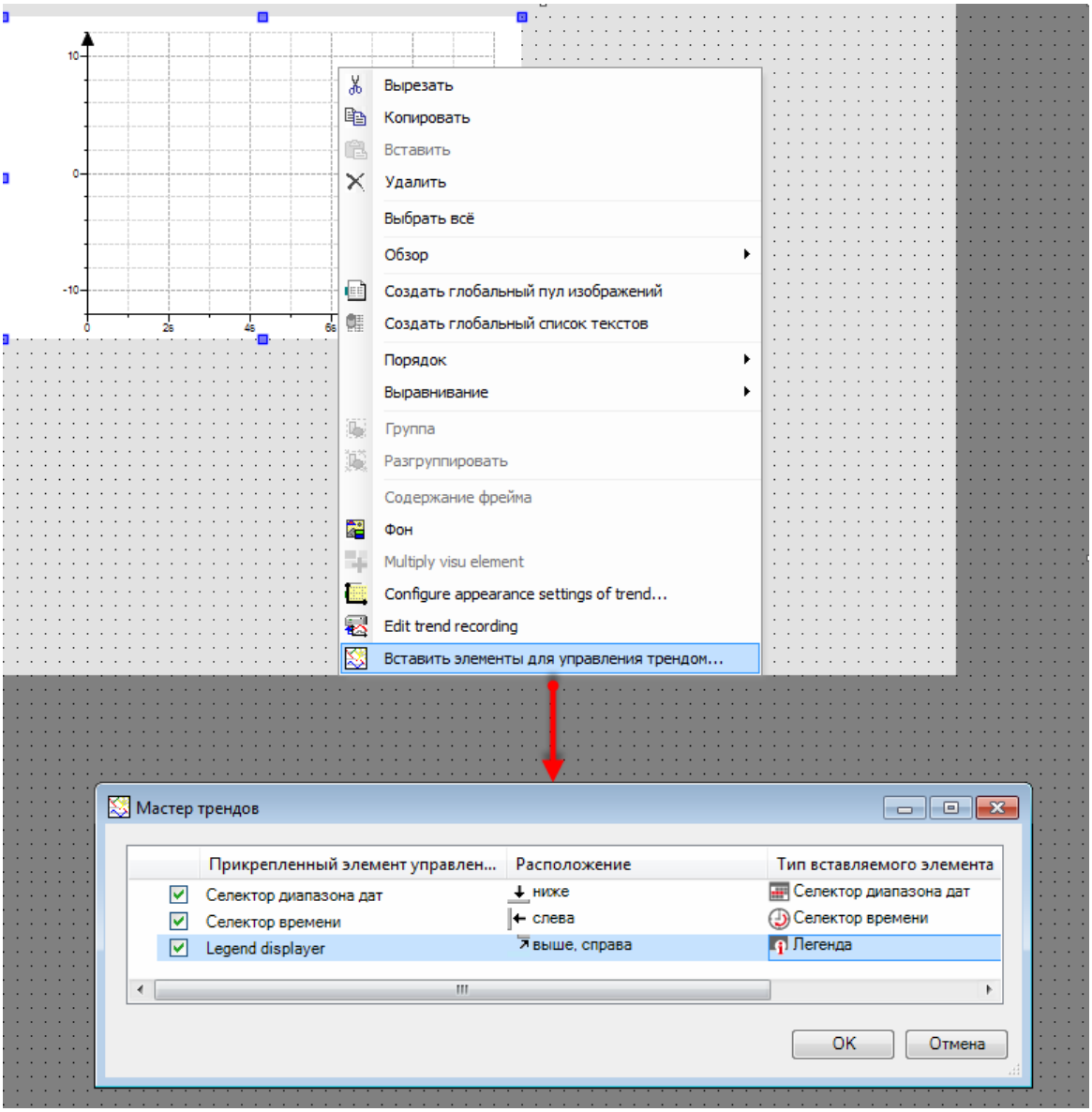


Рисунок 10.169 – Добавление элементов управления трендом

Это приведет к автоматическому добавлению и привязке к тренду элементов Селектор диапазона дат, Селектор времени и Легенда. Настройки элементов приведены ниже.

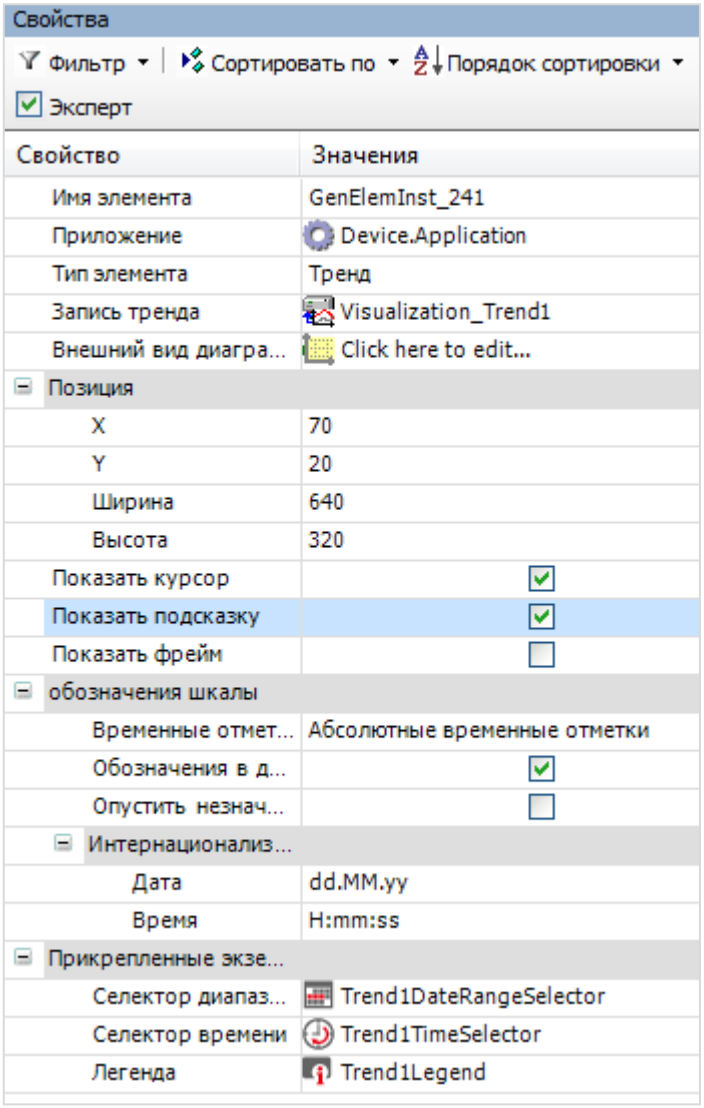


Рисунок 10.170 – Параметры элемента Тренд

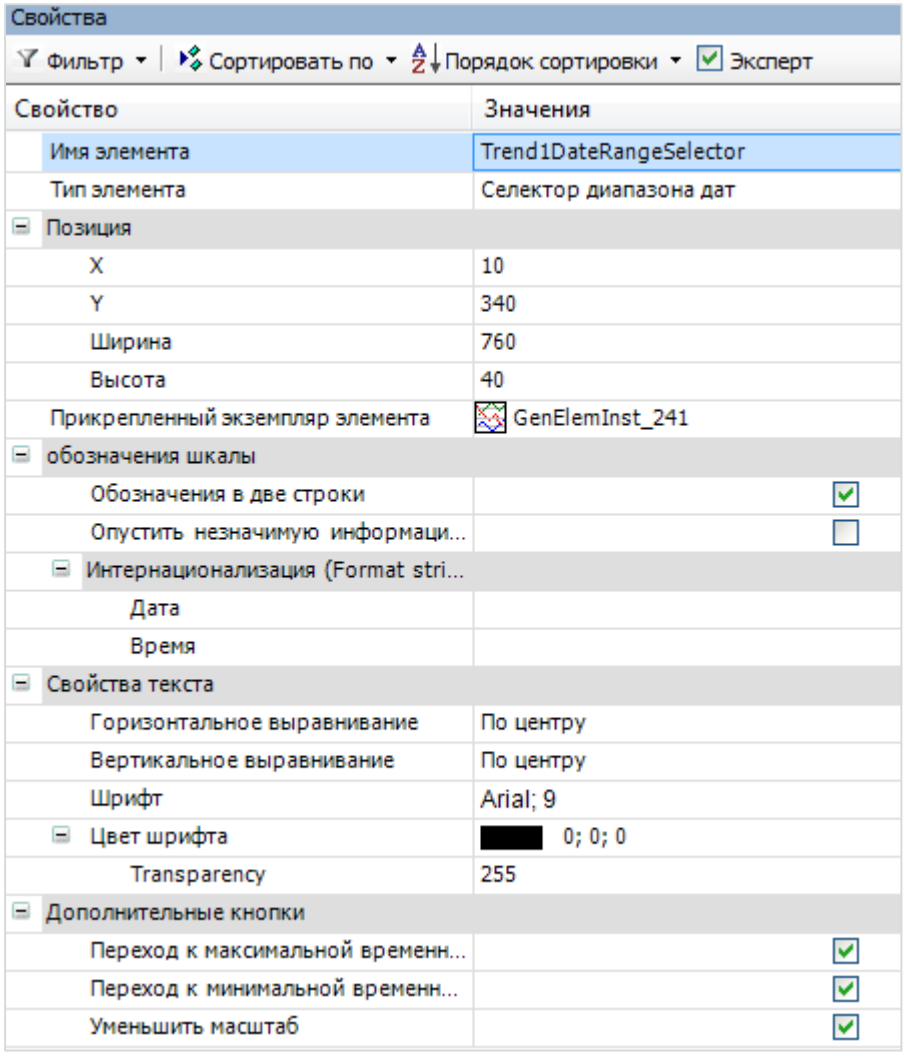


Рисунок 10.171 – Параметры элемента Селектор диапазона дат

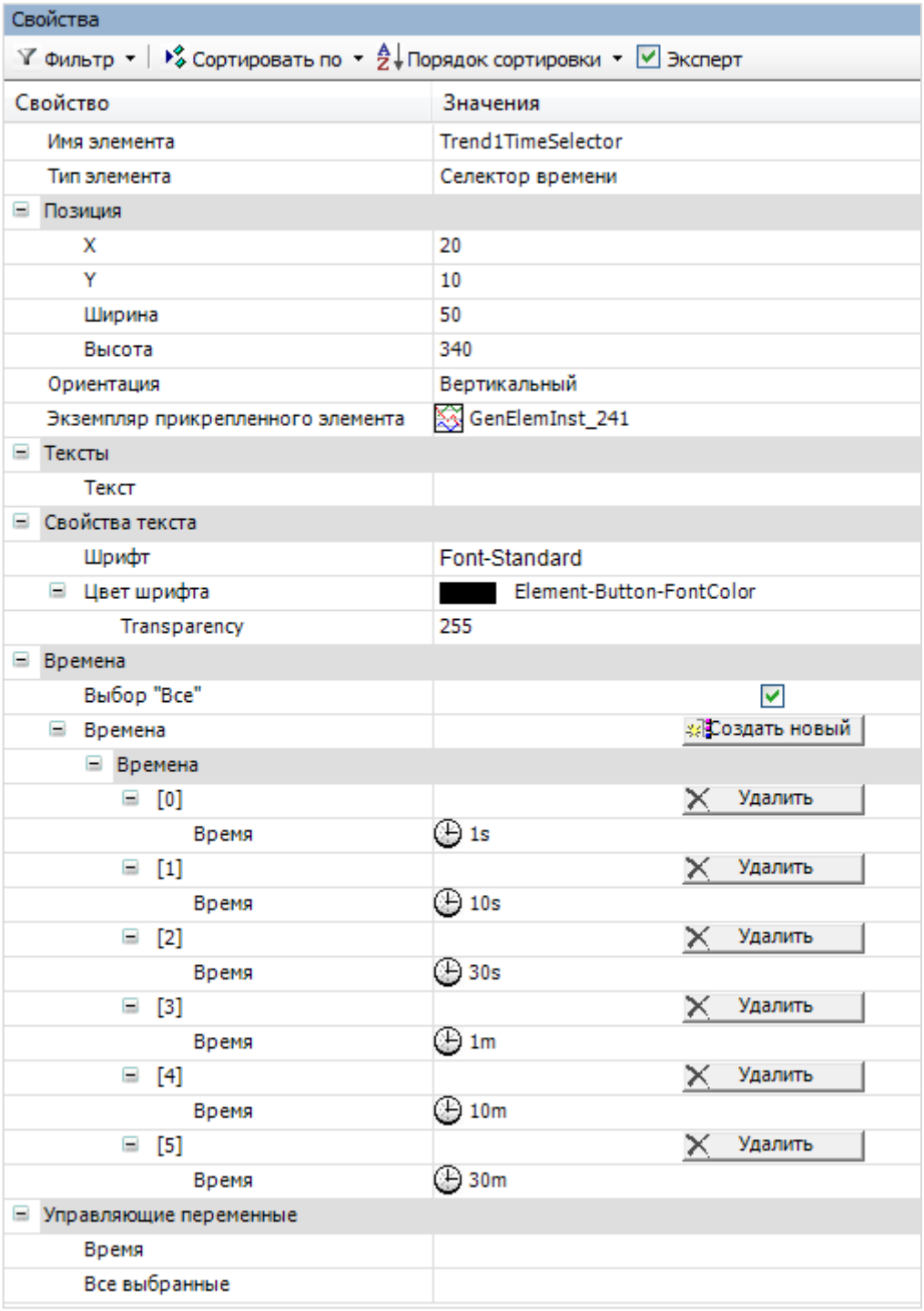


Рисунок 10.172 – Параметры элемента Селектор времени

Свойства

Фильтр | Сортировать по | Порядок сортировки

Эксперт

Свойство	Значения
Имя элемента	Trend1Legend
Тип элемента	Легенда
Позиция	
X	420
Y	0
Ширина	290
Высота	20
Ориентация	Горизонтально
Показать фрейм	☑
Размещение	
Макс. число строк	1
Макс. число столбцов	2
Свойства текста	
Шрифт	Font-Standard
Цвет шрифта	Font-Default-Color
Transparency	255

Рисунок 10.173 – Параметры элемента Легенда

Далее следует добавить на экран элемент Переключатель и привязать к нему переменную **bTrigger**. Также следует добавить поясняющую надпись (**Триггер записи**) с помощью элемента Метка.

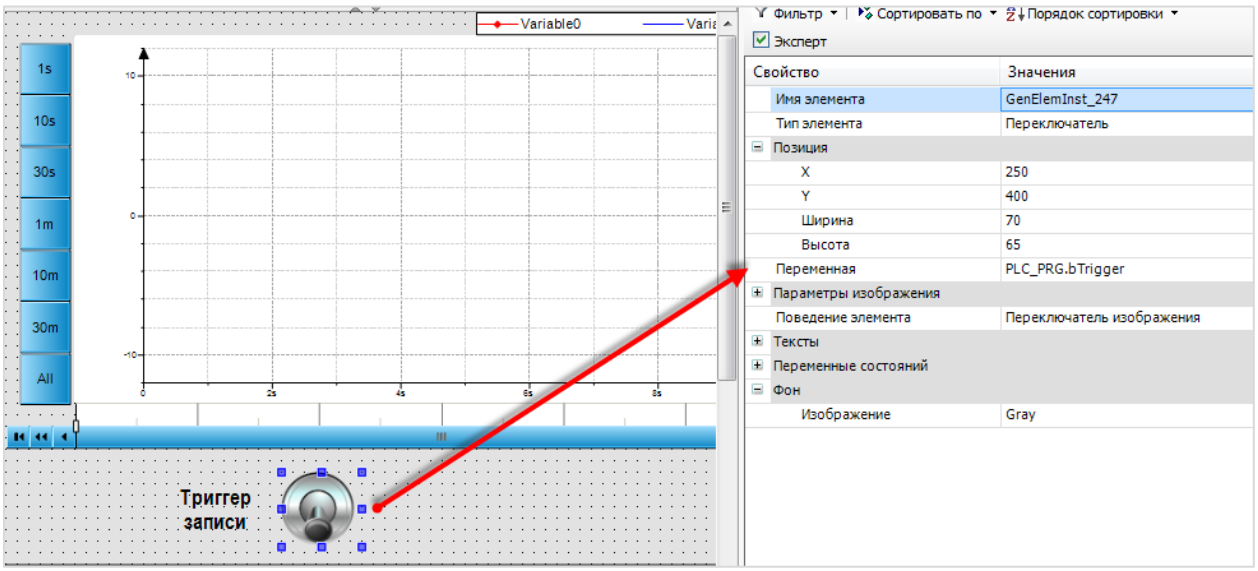


Рисунок 10.174 – Параметры элемента Переключатель

Затем следует добавить два элемента Бегунок, к которым будут привязаны переменные **iVar** и **rVar**.

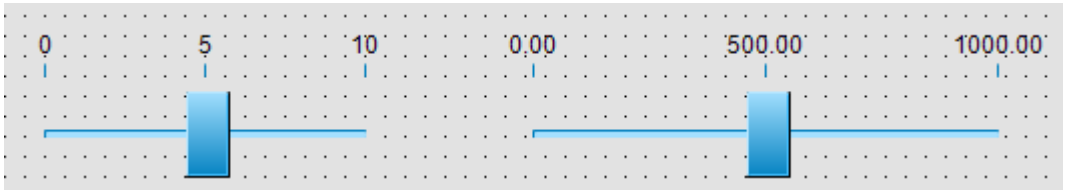


Рисунок 10.175 – Элементы Бегунок для изменения значений переменных тренда

Свойства

Фильтр | Сортировать по | Порядок сортировки | Эксперт

Свойство	Значения
Имя элемента	GenElemInst_250
Тип элемента	Бегунок
Позиция	
X	260
Y	420
Ширина	201
Высота	61
Переменная	PLC_PRG.iVar
Дискретность перемещения	
Прокрутка выполнена	☐
Шкала	
Показать шкалу	☑
Начало шкалы	0
Конец шкалы	10
Основная шкала	5
Подшкала	5
Формат шкалы (с-синтаксис)	%i
Высота шкалы	30
Линейка	
Тип диаграммы	Верх
Ориентация	Горизонтально
Направление движения	Слева направо
Переменные состояний	
Невидимый	
Отключение ввода	

Свойства

Фильтр | Сортировать по | Порядок сортировки | Эксперт

Свойство	Значения
Имя элемента	GenElemInst_252
Тип элемента	Бегунок
Позиция	
X	500
Y	420
Ширина	281
Высота	61
Переменная	PLC_PRG.rVar
Дискретность перемещения	
Прокрутка выполнена	☐
Шкала	
Показать шкалу	☑
Начало шкалы	0
Конец шкалы	1000
Основная шкала	500
Подшкала	0
Формат шкалы (с-синтаксис)	%.2f
Высота шкалы	30
Линейка	
Тип диаграммы	Верх
Ориентация	Горизонтально
Направление движения	Слева направо
Переменные состояний	
Невидимый	
Отключение ввода	

Рисунок 10.176 – Параметры элементов Бегунок

4. ***Следует обратить внимание***, что после добавления на экран элемента Тренд, в проект автоматически добавляется компонент **Trend Recording Manager** с подкомпонентом **Visualization_Trend1**, в котором хранится конфигурация тренда, а также системная задача **TrendRecordingTask**:

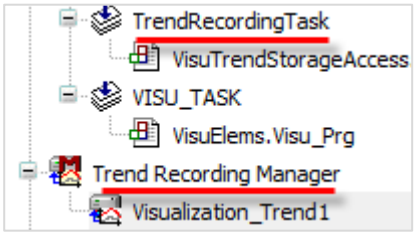
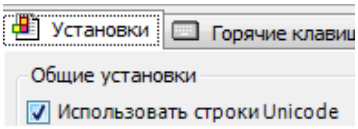


Рисунок 10.177 – Автоматически созданные компоненты тренда

В данном примере время цикла задачи тренда (**TrendRecordingTask**) составляет **T#1s**.

5. Настроить компонент Менеджер визуализации. В его установках поставить галочку **Использовать строки Unicode**:



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

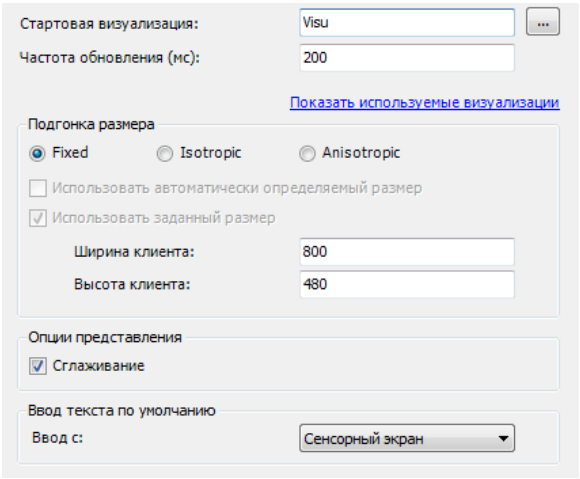


Рисунок 10.178 – Настройки таргет-визуализации

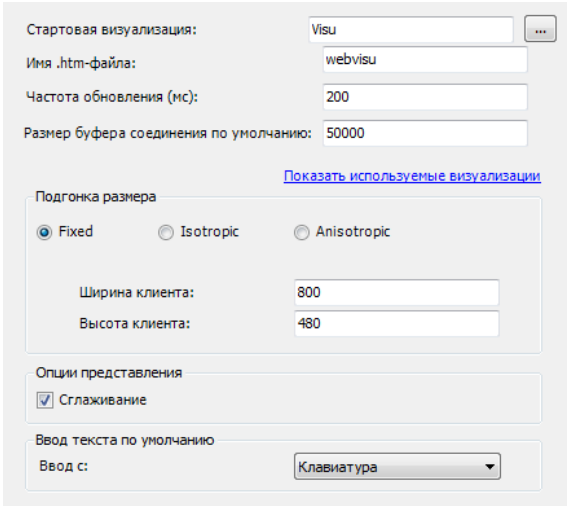


Рисунок 10.179 – Настройки web-визуализации

6. Запустить проект на виртуальном контроллере. Открыть в веб-браузере страницу http://localhost:8080/webvisu.htm, чтобы увидеть web-визуализацию проекта. ***Следует обратить внимание***, что браузер должен поддерживать **HTML5**. Проверить функционал проекта.

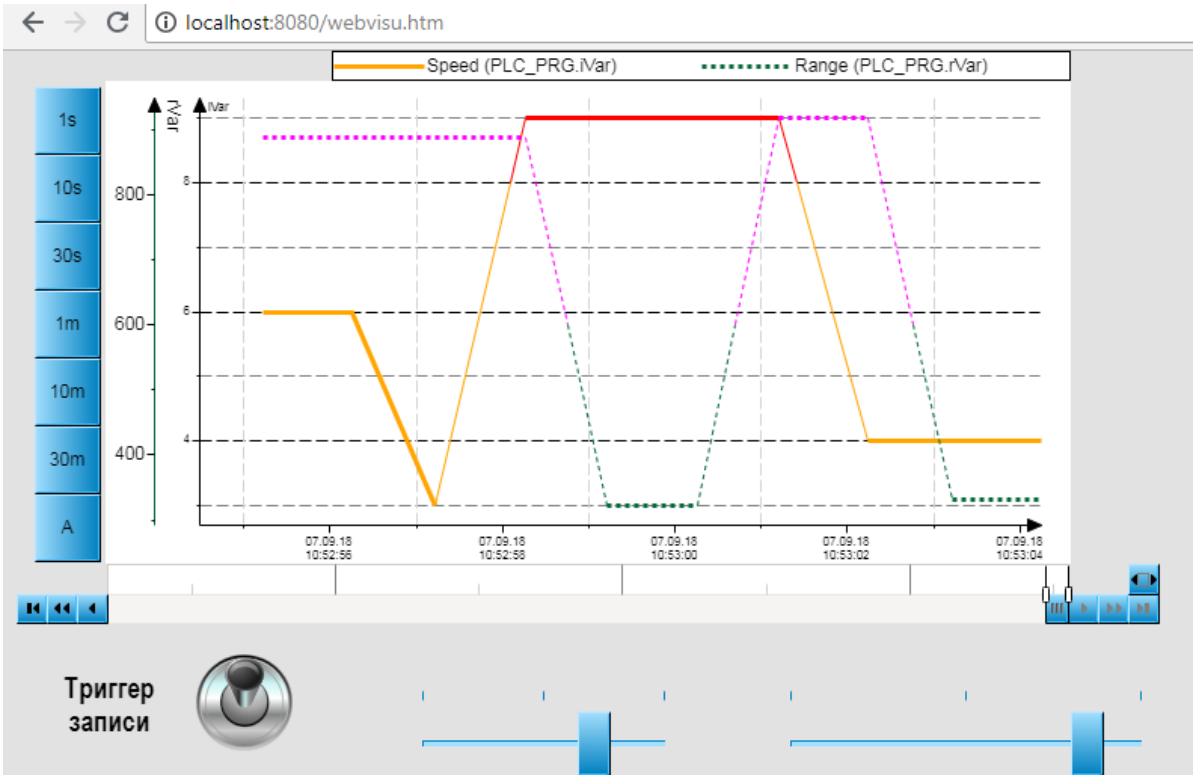


Рисунок 10.180 – Web-визуализация проекта в браузере Chrome

Нажатие на переключатель **Триггер записи** останавливает запись тренда. Повторное нажатие запускает запись заново, последняя точка предыдущей записи будет соединяется с первой точкой новой:

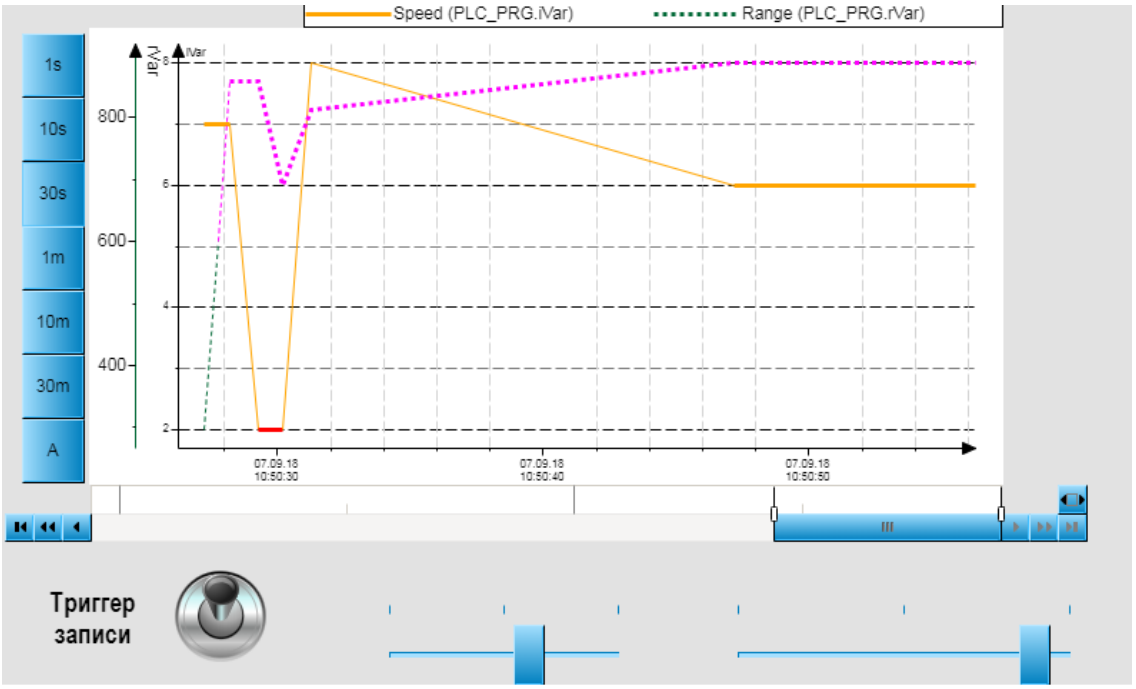


Рисунок 10.181 – Внешний вид тренда после включения и отключения триггера записи

С помощью кнопок **Селектора времени** можно менять масштаб тренда по оси времени:

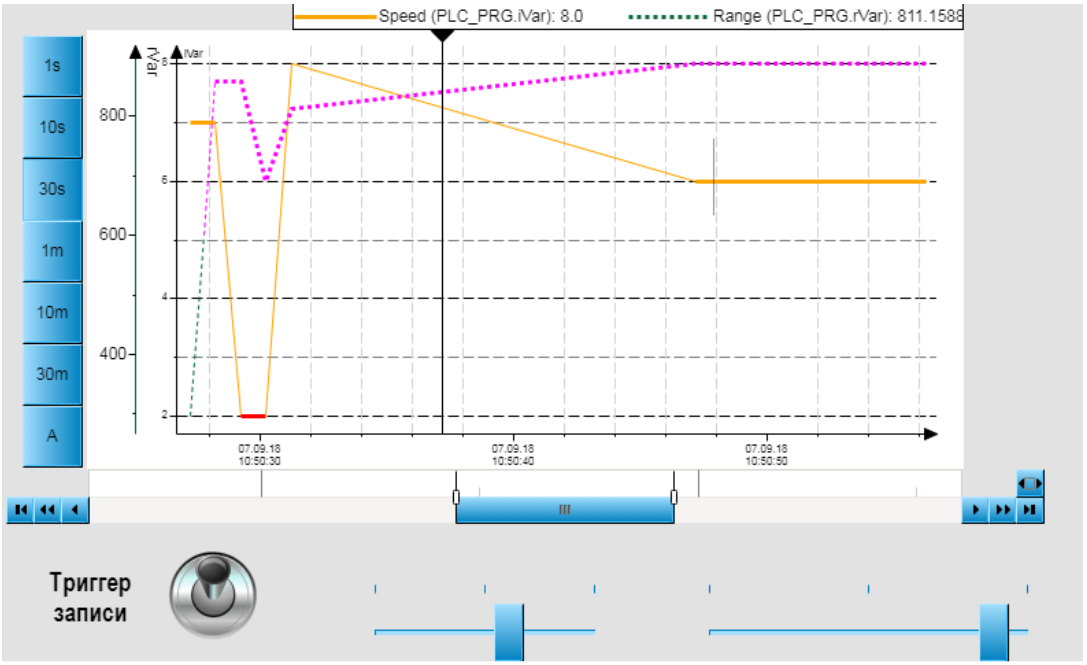


Рисунок 10.182 – Внешний вид тренда (масштаб времени = 30 с)

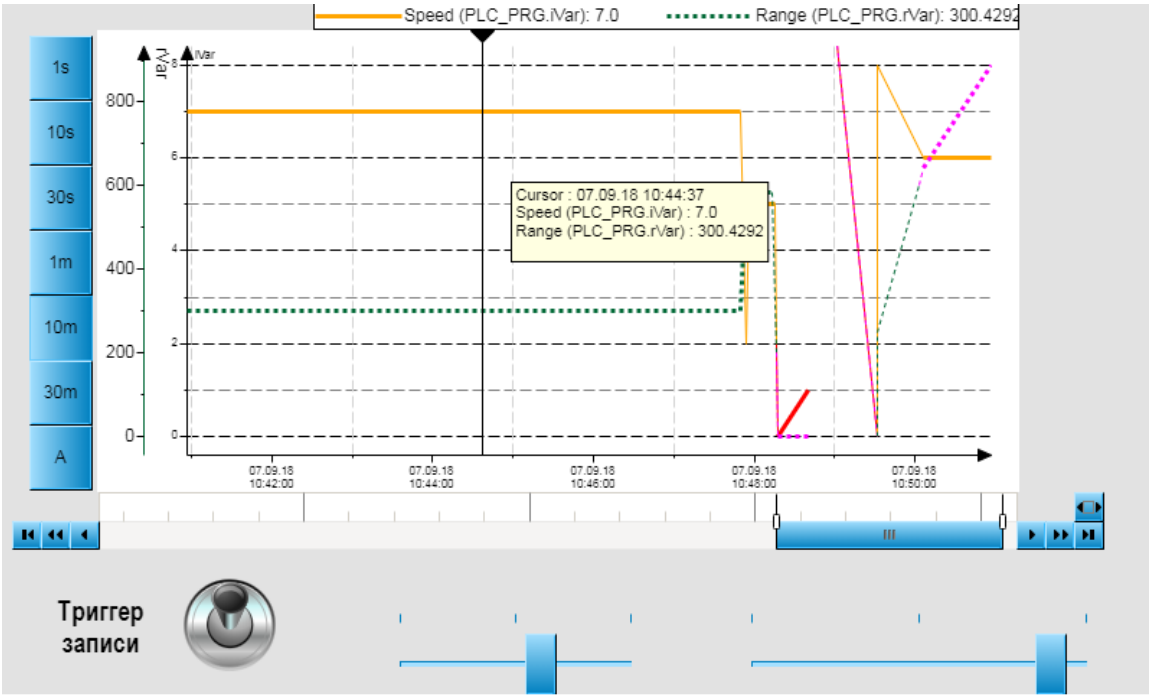


Рисунок 10.183 – Внешний вид тренда (масштаб времени = 10 мин.)

С помощью кнопок **Селектора диапазона дат** можно просматривать историю тренда. Функции кнопок описаны в [таблице 7.31](). В режиме просмотра истории при наведении курсора на график, текущее значение переменной отображается легендой, а также в метке курсора.

Для возвращения тренда в режим обновления в реальном времени следует нажать кнопку .

10.2.11 WebBrowser

Данный пример посвящен работе с графическим примитивом WebBrowser, используемом для открытия в плоскости экрана визуализации веб-страниц. ***Следует обратить внимание***, что использование примитива возможно только на целевых устройствах с поддержкой **Java** и **HTML5**. Не гарантируется открытие в элементе любой веб-страницы.

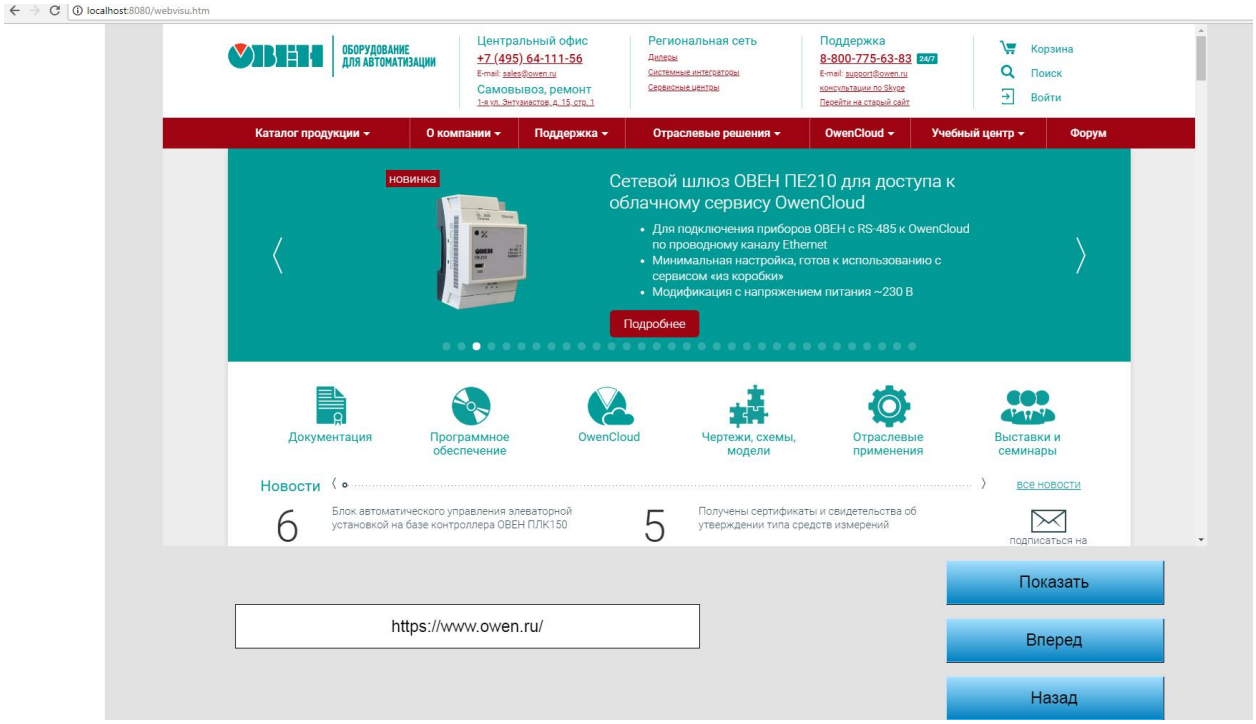


Рисунок 10.184 – Внешний вид примера WebBrowser

Пример создан в среде **CODESYS V3.5 SP11 Patch 5** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с таргет-файлом **3.5.11.5**.

Пример доступен для скачивания: Example_WebBrowser.projectarchive

Для создания примера с использованием элемента **WebBrowser** следует:

1. Создать новый ***стандартный*** проект **CODESYS** с названием **Example_WebBrowser** и настройками по умолчанию: таргет – CODESYS Control Win V3, язык PLC_PRG – **ST**.
2. В программе **PLC_PRG** объявить 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
```

Рисунок 10.185 – Объявление переменных программы PLC_PRG

3. Добавить в проект экран визуализации **Visualization**. В его свойствах выбрать размер **800 × 480.** Экран будет содержать элемент WebBrowser, элемент Текстовое поле и три элемента Кнопка.

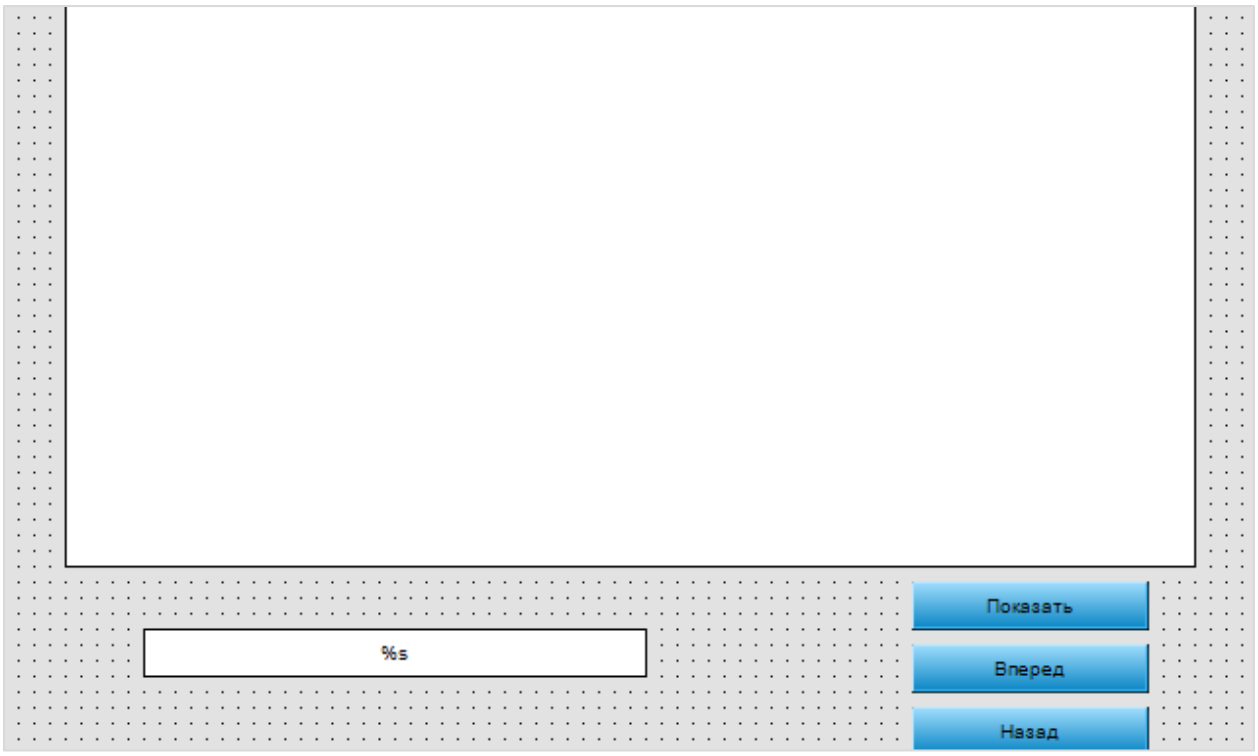


Рисунок 10.186 – Содержание экрана Visualization

4. Настроить элемент WebBrowser:

Свойства

Фильтр | Сортировать по | Порядок сортировки

Эксперт

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

Рисунок 10.187 – Параметры элемента WebBrowser

5. Настроить элемент Текстовое поле:

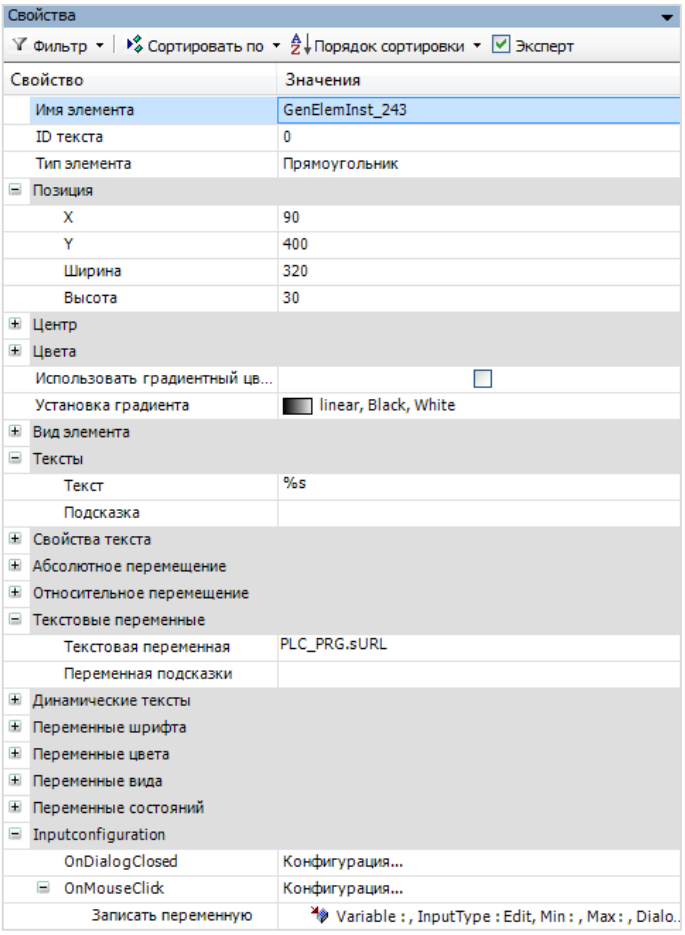


Рисунок 10.188 – Параметры элемента Текстовое поле

Во вкладке InputConfiguration к параметру **OnMouseClick** привязать действие Записать переменную:

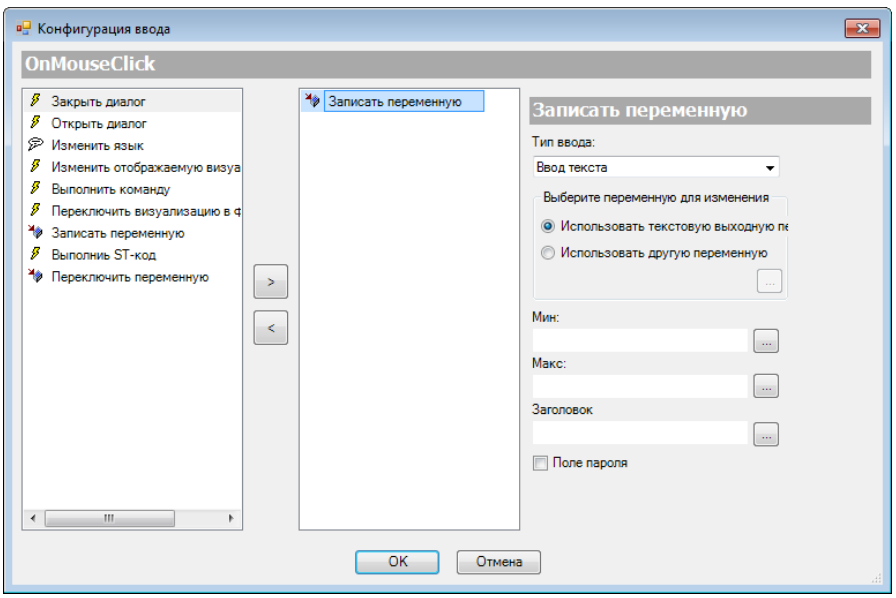


Рисунок 10.189 – Настройки действия элемента Текстовое поле

6. К кнопке **Показать** привязать переменную **bShow**:

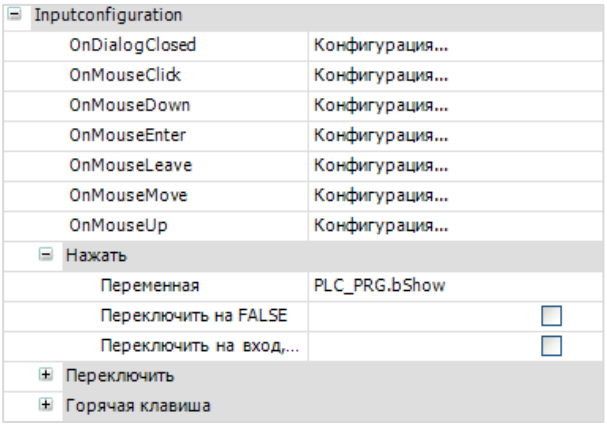


Рисунок 10.190 – Параметры кнопки Показать

К кнопке **Вперед** привязать переменную **bForward**:

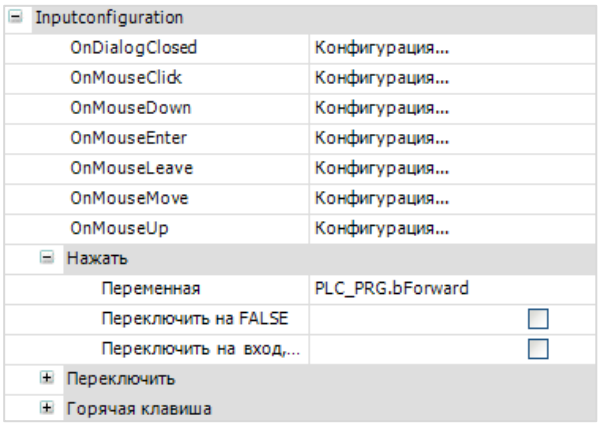


Рисунок 10.191 – Параметры кнопки Вперед

К кнопке **Назад** привязать переменную **bBack**:

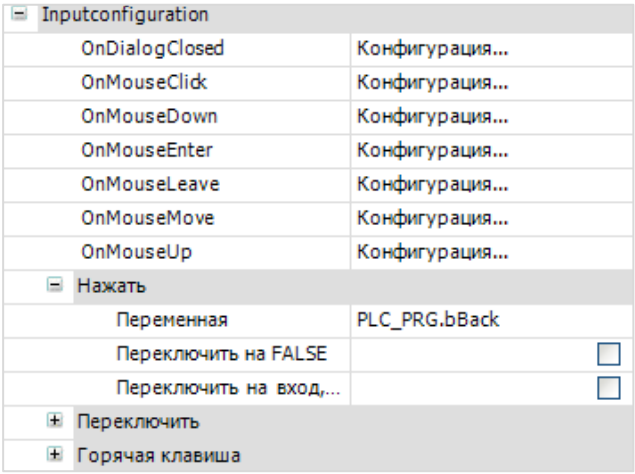
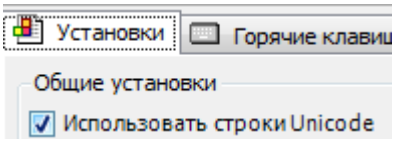


Рисунок 10.192 – Параметры кнопки Назад

7. Настроить компонент Менеджер визуализации. В его установках поставить галочку **Использовать строки Unicode**:



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

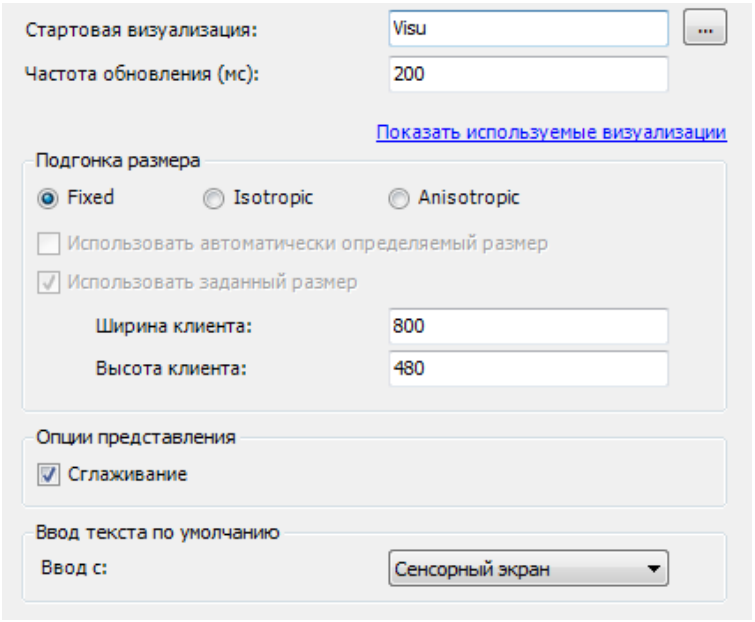


Рисунок 10.193 – Настройки таргет-визуализации

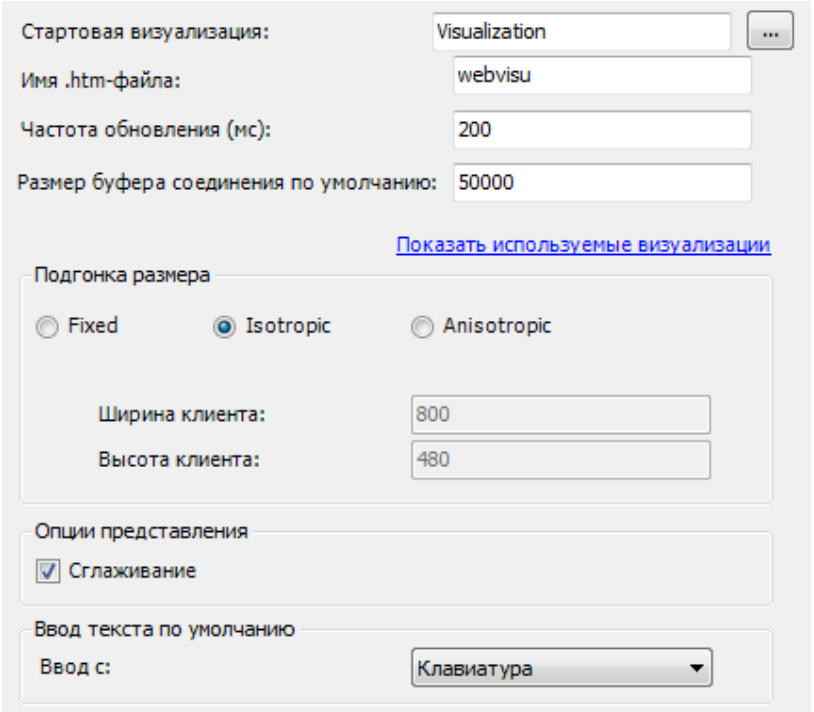


Рисунок 10.194 – Настройки web-визуализации

8. Запустить проект на виртуальном контроллере. Открыть в веб-браузере страницу http://localhost:8080/webvisu.htm, чтобы увидеть web-визуализацию проекта. ***Следует обратить внимание***, что браузер должен поддерживать **HTML5**. Проверить функционал проекта.

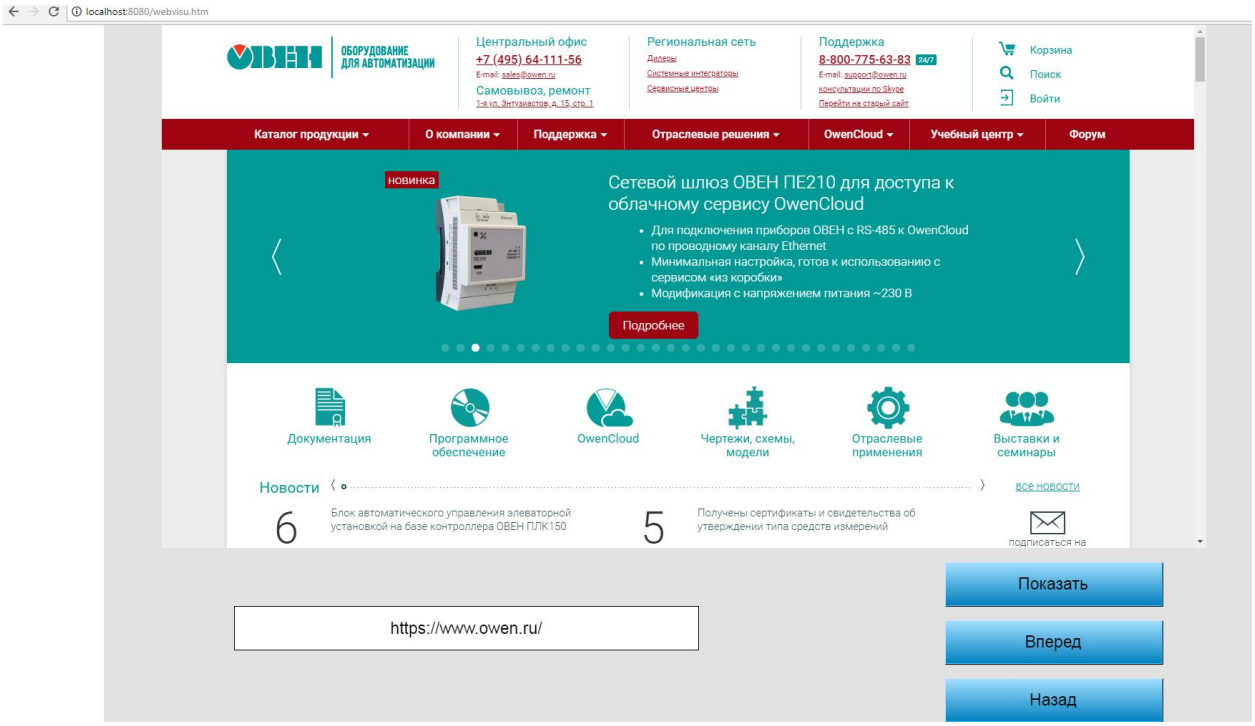


Рисунок 10.195 – Web-визуализация проекта в браузере Chrome

Сначала следует ввести в текстовом поле адрес веб-страницы (в формате https://owen.ru) и нажать кнопку **Показать**. Это приведет к открытию соответствующей веб-страницы в элементе.

Затем открыть в элементе другую страницу (с помощью ввода ее адреса). Нажать кнопку **Назад**, чтобы вернуться на предыдущую страницу. Далее нажать кнопку **Вперед**, чтобы перейти обратно.

10.2.12 Текстовый редактор

Данный пример посвящен работе с графическим примитивом Текстовый редактор, используемым для просмотра и редактирования текстовых файлов (в формате **.txt**).

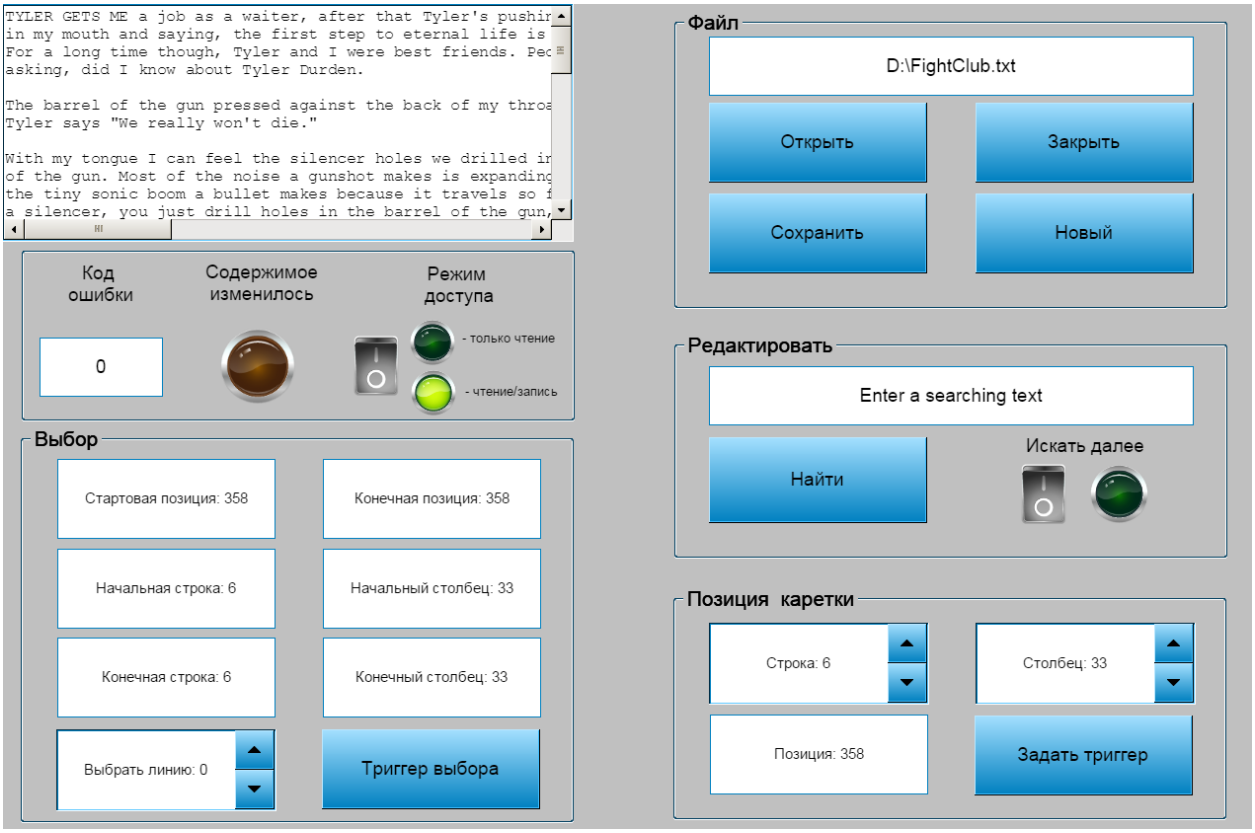


Рисунок 10.196 – Внешний вид примера Текстовый редактор

Пример создан в среде **CODESYS V3.5 SP11 Patch 5** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с таргет-файлом **3.5.11.5**.

Пример доступен для скачивания: Example_TextEditor.projectarchive

Для создания примера с использованием элемента **Текстовый редактор** следует:

1. Создать новый ***стандартный*** проект **CODESYS** с названием **Example_TextEditor** и настройками по умолчанию: таргет – CODESYS Control Win V3, язык PLC_PRG – **ST**.
2. В программе **PLC_PRG** объявить следующие переменные:

```
PROGRAM PLC_PRG
VAR
    // Файл
    sFileName:          STRING:='Choose a path to file (D:\1.txt)';     // Имя файла
    bOpen:              BOOL;                                          // Открыть
    bClose:             BOOL;                                          // Закрыть
    bSave:              BOOL;                                          // Сохранить
    bNew:               BOOL;                                          // Новый

    // Редактировать
    sSearch:            STRING:='Enter a searching text';              // Искать
    bFind:              BOOL;                                          // Найти
    bFindNext:          BOOL;                                          // Найти далее

    // Позиция каретки
    iLine:              INT;                                           // Линия
    iColumn:            INT;                                           // Столбец
    iPosition:          INT;                                           // Позиция
    bTriggerSet:        BOOL;                                          // Задать триггер

    // Выбор
    iStartPosition:     INT;                                           // Стартовая позиция
    iEndPosition:       INT;                                           // Конечная позиция
    iStartLineNumber:   INT;                                           // Начальный номер строки
    iStartColumnIndex:  INT;                                           // Начальный индекс столбца
    iEndLineNumber:     INT;                                           // Конечный номер строки
    iEndColumnIndex:    INT;                                           // Конечный индекс столбца
    iLineToSelect:      INT;                                           // Выбираемая линия
    bTriggerSelect:     BOOL;                                          // Выбор триггера

    // Обработка кода ошибок
    iErrorCode:         INT;                                           // Переменная для кода ошибки

    //
    bContentChanged:    BOOL;                                          // Переменная для содержимого изменилась
    bAccessMode:        BOOL;                                          // Переменная для режима доступа
    bAccessModeRev:     BOOL;                                          // Инвертированное значение режима доступа к файлу (для лампы)
END_VAR
```

Рисунок 10.198 – Объявление переменных программы PLC_PRG

3. Код программы будет выглядеть следующим образом:

```
bAccessModeRev:=NOT bAccessMode;  // инвертированное значение режима доступа к файлу (для лампы)
```

Рисунок 10.198 – Код программы PLC_PRG

4. Добавить в проект экран визуализации **Visualization**. В его свойствах выбрать размер **1280 × 860.**

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.

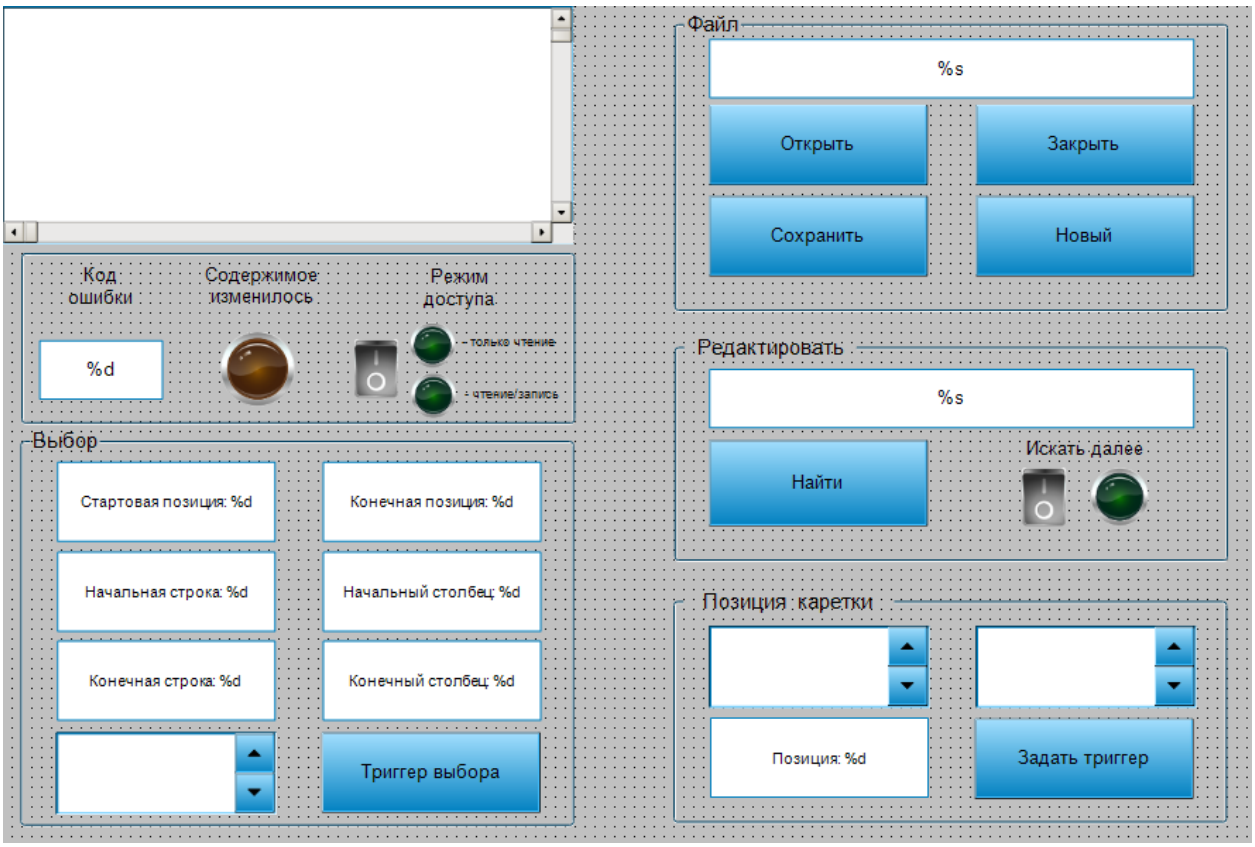


Рисунок 10.199 – Содержание экрана Visualization

5. Настроить 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

Рисунок 10.200 – Параметры элемента Текстовый редактор

6. Добавить на экран панель управления параметрами вкладки **Файл**. Данные параметры позволяют открывать, закрывать, сохранять и создавать текстовые файлы.

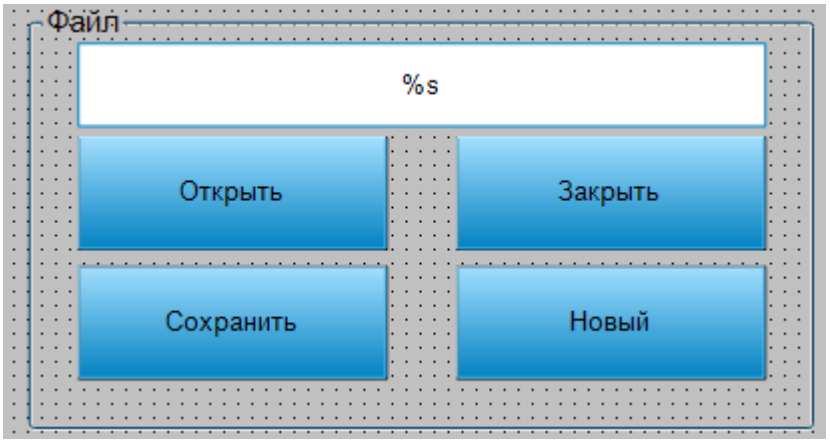


Рисунок 10.201 – Панель управления параметрами вкладки Файл

Панель представляет собой элемент Группа, в которой расположен элемент Текстовое поле и четыре элемента Кнопка. Параметры элементов приведены ниже (неотображенные 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...
⊟ OnMouseClick	Конфигурация...
Записать переменную	Variable :, InputType : Edit, Min : , Max : ,
OnMouseDown	Конфигурация...

Рисунок 10.202 – Параметры элемента Текстовое поле

Во вкладке InputConfiguration к параметру **OnMouseClick** привязать действие Записать переменную:

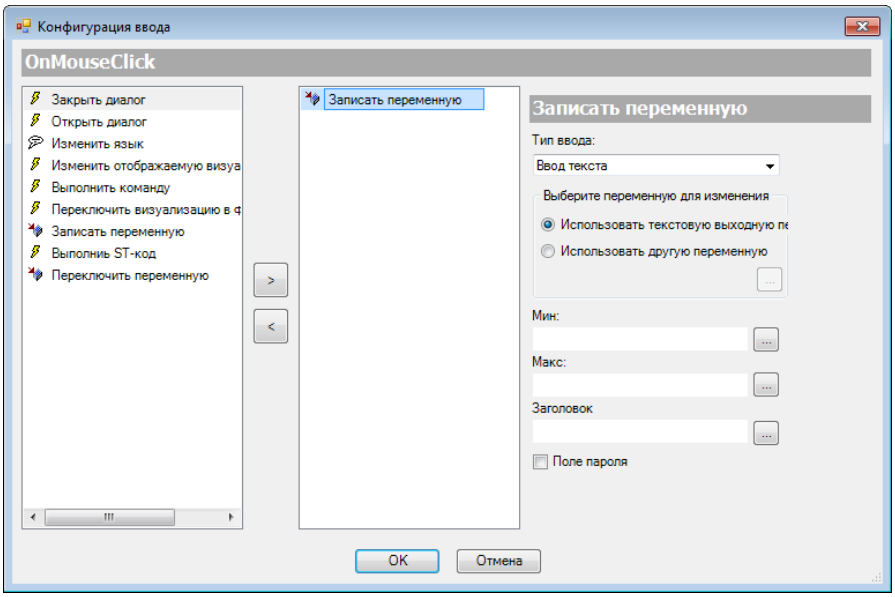


Рисунок 10.203 – Настройки действия элемента Текстовое поле

К кнопкам следует привязать соответствующие переменные:

Inputconfiguration	
OnDialogClosed	Конфигурация...
OnMouseClick	Конфигурация...
OnMouseDown	Конфигурация...
OnMouseEnter	Конфигурация...
OnMouseLeave	Конфигурация...
OnMouseMove	Конфигурация...
OnMouseUp	Конфигурация...
Нажать	
Переключить	
Переменная	PLC_PRG.bOpen
Переключить наверх, есл...	☐
Горячая клавиша	

Рисунок 10.204 – Параметры кнопки Открыть

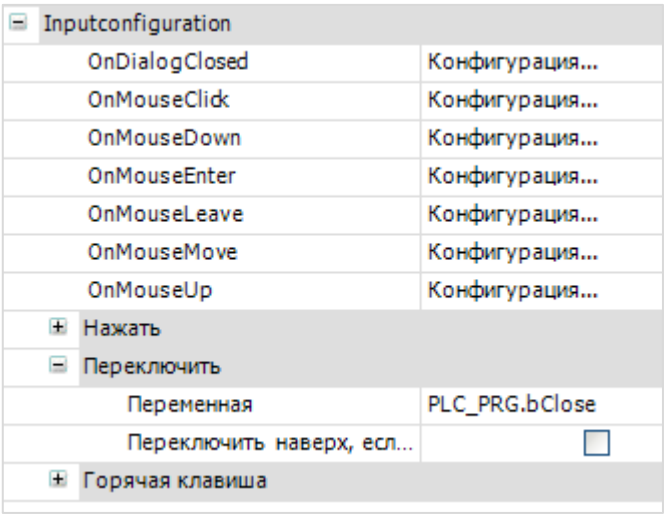


Рисунок 10.205 – Параметры кнопки Закрыть

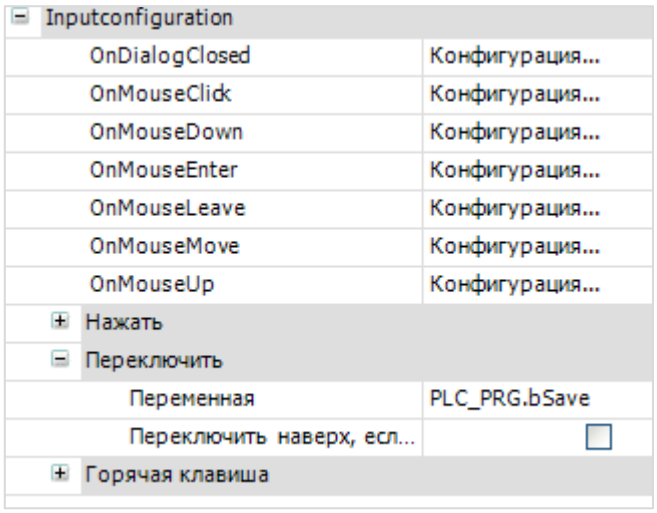


Рисунок 10.206 – Параметры кнопки Сохранить

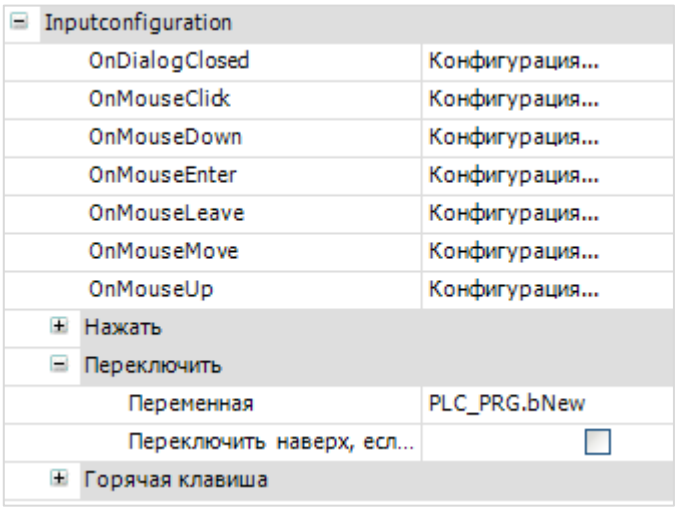


Рисунок 10.207 – Параметры кнопки Новый

7. Добавить на экран панель управления параметрами вкладки **Редактировать**. Данные параметры позволяют осуществлять поиск по текстовому файлу.

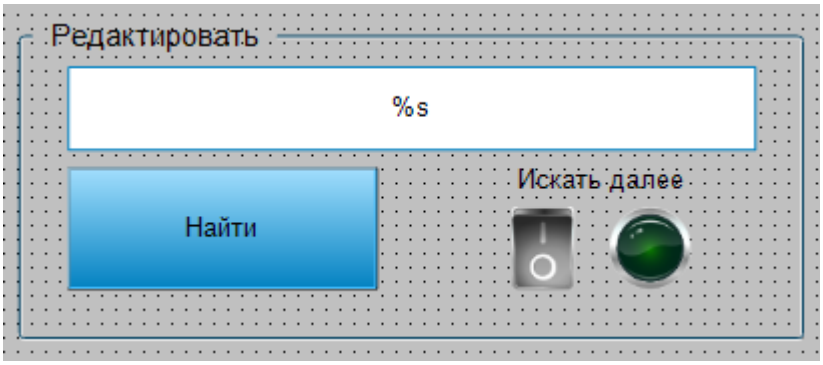


Рисунок 10.208 – Панель управления параметрами вкладки Файл

Панель представляет собой элемент Группа, в которой расположен элемент Текстовое поле, элемент Кнопка, элемент Клавишный выключатель и элемент Индикатор. Параметры элементов приведены ниже (неотображенные параметры находятся в значениях по умолчанию).

Свойства

Фильтр | Сортировать по | Порядок сортировки | Эксперт

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...
⊟ OnMouseClick	Конфигурация...
Записать переменную	Variable :, InputType : Edit, Min : , Max :
OnMouseDown	Конфигурация...

Рисунок 10.209 – Параметры элемента Текстовое поле

Во вкладке InputConfiguration к параметру **OnMouseClick** следует привязать действие Записать переменную:

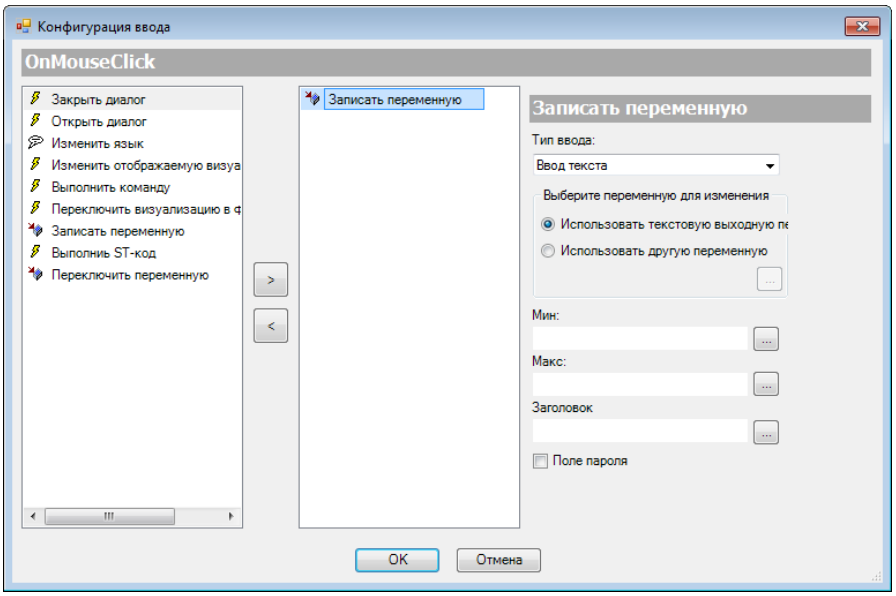


Рисунок 10.210 – Настройки действия элемента Текстовое поле

К кнопке **Найти** следует привязать переменную **bFind**:

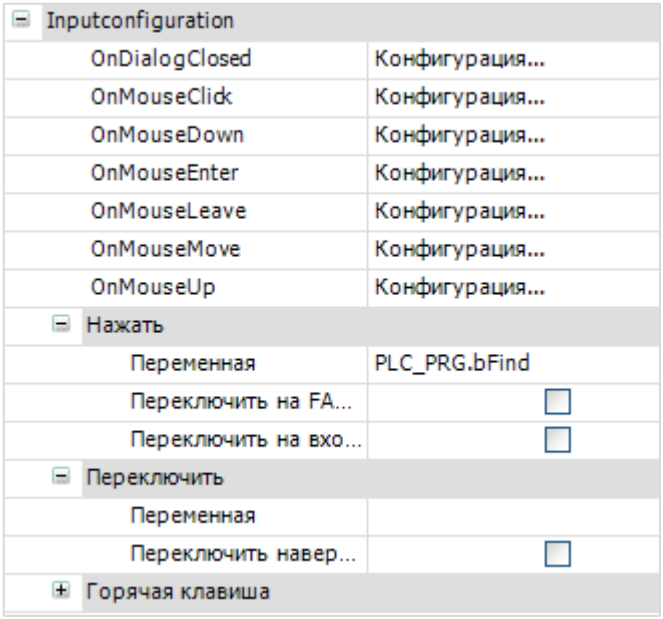


Рисунок 10.211 – Параметры кнопки Найти

К клавишному выключателю и индикатору (**Искать далее**) следует привязать переменную **bFindNext**:

Позиция	
X	351
Y	139
Ширина	52
Высота	62
Переменная	PLC_PRG.bFindNext

Рисунок 10.212 – Параметры элемента Клавишный выключатель (к индикатору привязана та же переменная)

8. Добавить на экран панель управления параметрами вкладки **Позиция каретки**. Данные параметры управляют перемещением курсора по текстовому файлу.

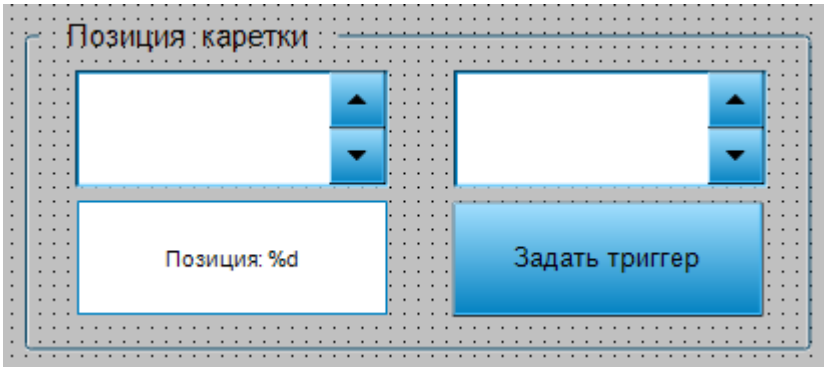


Рисунок 10.213 – Панель управления параметрами вкладки Позиция каретки

Панель представляет собой элемент Группа, в которой расположены два элемента Управление вращением, элемент Текстовое поле и элемент Кнопка. Параметры элементов приведены ниже (неотображенные параметры находятся в значениях по умолчанию).

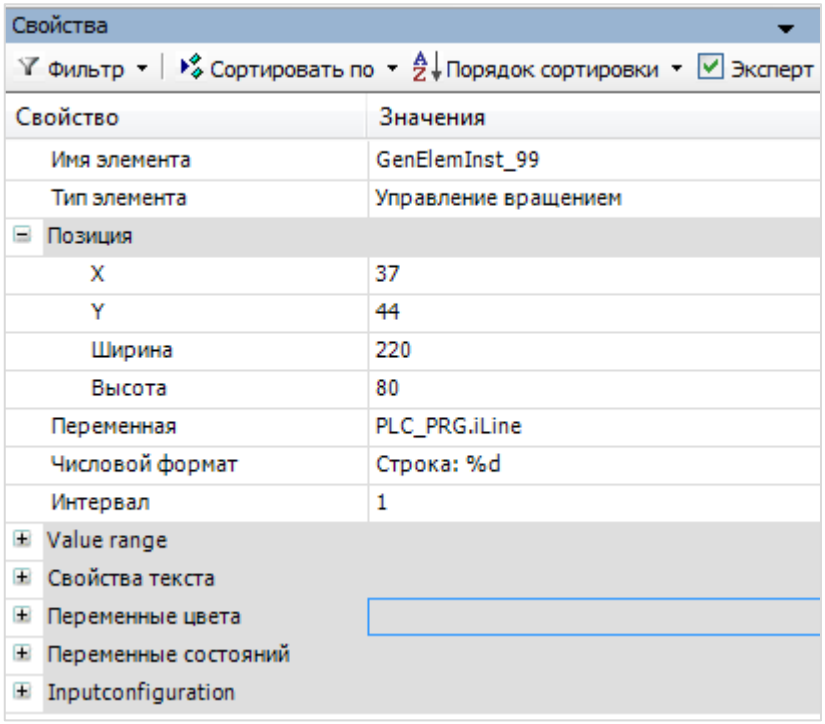


Рисунок 10.214 – Параметры элемента Управление вращением 1

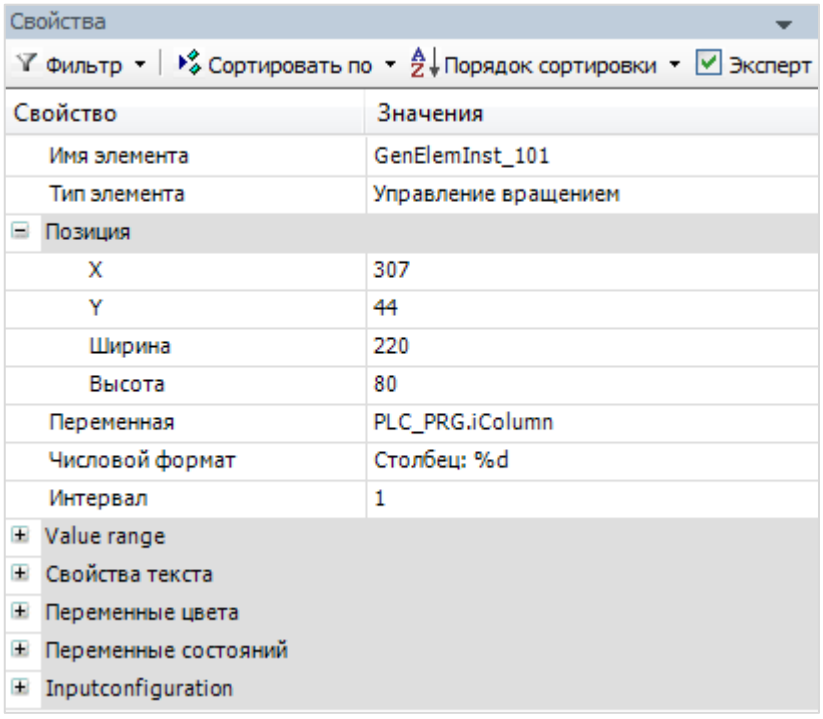


Рисунок 10.215 – Параметры элемента Управление вращением 2

Свойства

Фильтр | Сортировать по | Порядок сортировки

Эксперт

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	

Рисунок 10.216 – Параметры элемента Текстовое поле

К кнопке **Задать триггер** следует привязать переменную **bTriggerSet**:

Inputconfiguration	
OnDialogClosed	Конфигурация...
OnMouseClick	Конфигурация...
OnMouseDown	Конфигурация...
OnMouseEnter	Конфигурация...
OnMouseLeave	Конфигурация...
OnMouseMove	Конфигурация...
OnMouseUp	Конфигурация...
Нажать	
Переключить	
Переменная	PLC_PRG.bTriggerSet
Переключить навер...	☐
Горячая клавиша	

Рисунок 10.217 – Параметры элемента Кнопка (Задать триггер)

9. Добавить на экран панель управления параметрами вкладки **Выбор**. Данные параметры характеризуют выделенный фрагмент в текстовом файле.

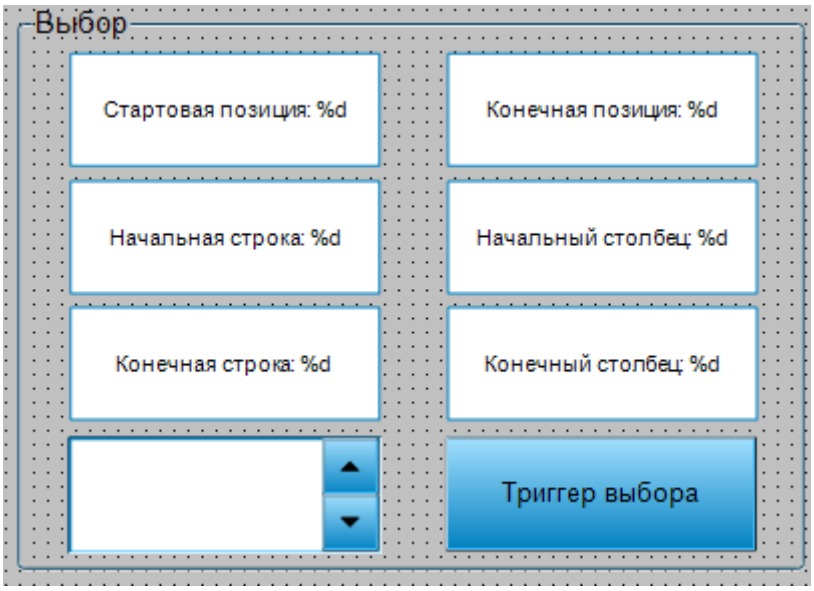


Рисунок 10.218 – Панель управления параметрами вкладки Выбор

Панель представляет собой элемент Группа, в которой расположены 6 элементов Текстовое поле, элемент Управление вращением и элемент Кнопка. Параметры элементов приведены ниже (неотображенные параметры находятся в значениях по умолчанию).

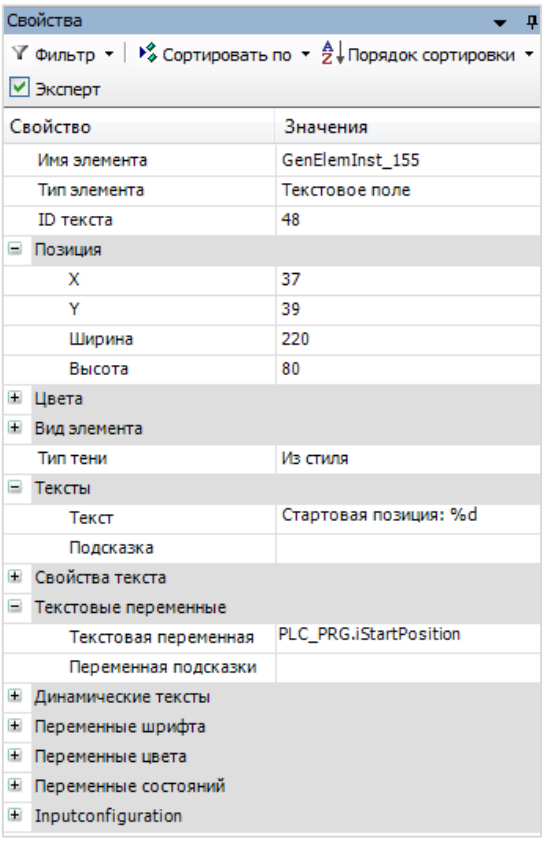


Рисунок 10.219 – Параметры элемента Текстовое поле 1

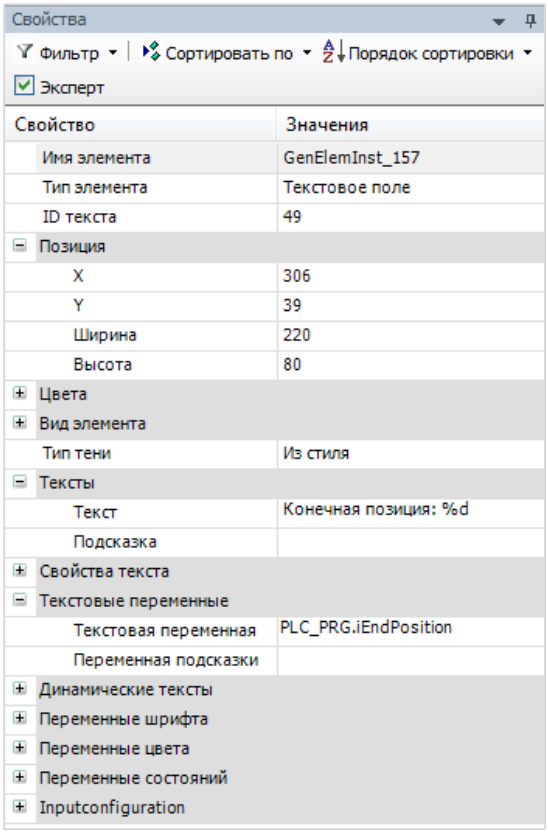


Рисунок 10.220 – Параметры элемента Текстовое поле 2

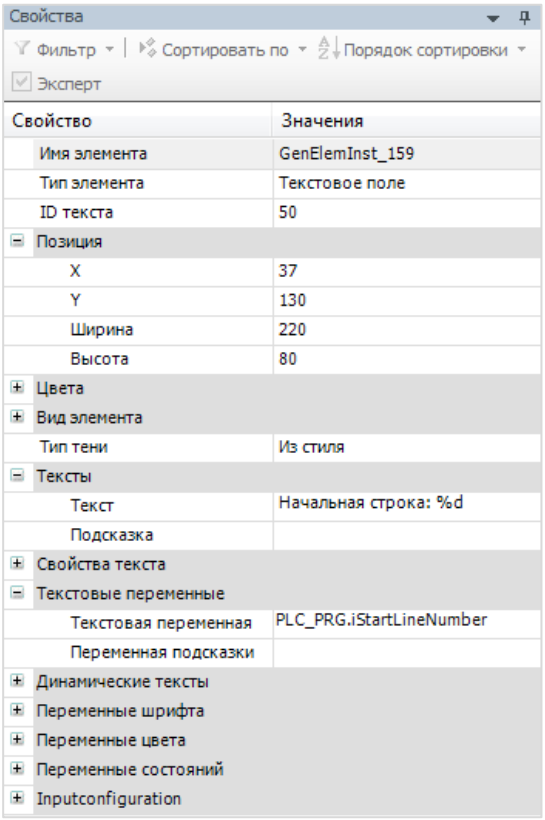


Рисунок 10.221 – Параметры элемента Текстовое поле 3

Свойства

Фильтр | Сортировать по | Порядок сортировки

Эксперт

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	

Рисунок 10.222 – Параметры элемента Текстовое поле 4

Свойства

Фильтр | Сортировать по | Порядок сортировки

Эксперт

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	

Рисунок 10.223 – Параметры элемента Текстовое поле 5

Свойства

Фильтр | Сортировать по | Порядок сортировки

Эксперт

Свойство	Значения
Имя элемента	GenElemInst_165
Тип элемента	Текстовое поле
ID текста	53
⊟ Позиция	
X	306
Y	220
Ширина	220
Высота	80
⊞ Цвета	
⊞ Вид элемента	
Тип тени	Из стиля
⊟ Тексты	
Текст	Конечный столбец: %d
Подсказка	
⊞ Свойства текста	
⊟ Текстовые переменные	
Текстовая переменная	PLC_PRG.iEndColumnIndex
Переменная подсказки	
⊞ Динамические тексты	
⊞ Переменные шрифта	
⊞ Переменные цвета	
⊞ Переменные состояний	
⊞ Inputconfiguration	

Рисунок 10.224 – Параметры элемента Текстовое поле 6

Свойства

Фильтр | Сортировать по | Порядок сортировки | Эксперт

Свойство	Значения
Имя элемента	GenElemInst_103
Тип элемента	Управление вращением
⊟ Позиция	
X	37
Y	314
Ширина	220
Высота	80
Переменная	PLC_PRG.iLineToSelect
Числовой формат	Выбрать линию: %d
Интервал	1
⊞ Value range	
⊞ Свойства текста	
⊞ Переменные цвета	
⊞ Переменные состояний	
⊞ Inputconfiguration	

Рисунок 10.225 – Параметры элемента Управление вращением

К кнопке **Триггер выбора** следует привязать переменную **bTriggerSelect**:

Inputconfiguration	
OnDialogClosed	Конфигурация...
OnMouseClick	Конфигурация...
OnMouseDown	Конфигурация...
OnMouseEnter	Конфигурация...
OnMouseLeave	Конфигурация...
OnMouseMove	Конфигурация...
OnMouseUp	Конфигурация...
Нажать	
Переменная	
Переключить на FA...	☐
Переключить на вхо...	☐
Переключить	
Переменная	PLC_PRG.bTriggerSelect
Переключить навер...	☐
Горячая клавиша	

Рисунок 10.226 – Параметры элемента Кнопка (Триггер выбора)

10. Добавить на экран панель управления остальными параметрами элемента.

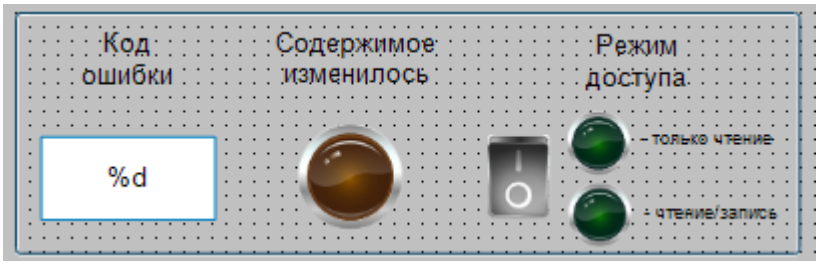


Рисунок 10.227 – Панель управления остальными параметрами элемента

Панель представляет собой элемент Группа, в которой расположен элемент Текстовое поле, три элемента Индикатор и элемент Клавишный переключатель. Пять поясняющих надписей сделаны с помощью элемента Метка. Параметры элементов приведены ниже (неотображенные 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	

Рисунок 10.228 – 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	☐
Прозрачный цвет	Black
Изотропный тип	Изотропия
Горизонтальное выравн...	Лево
Вертикальное выравнив...	Верх
⊞ Тексты	
⊞ Переменные состояний	
⊟ Фон	
Изображение	Yellow

Рисунок 10.229 – Параметры индикатора Содержимое изменилось

Свойства

Фильтр | Сортировать по | 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	☐
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

Рисунок 10.230 – 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	☐
Прозрачный цвет	Black
Изотропный тип	Изотропия
Горизонтальное выравн...	Лево
Вертикальное выравнив...	Верх
Тексты	
Переменные состояний	
Фон	
Изображение	Green

Рисунок 10.231 – Параметры индикатора Только чтение

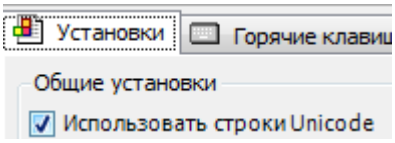
Свойства

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	☐
Прозрачный цвет	Black
Изотропный тип	Изотропия
Горизонтальное выравн...	Лево
Вертикальное выравнив...	Верх
Тексты	
Переменные состояний	
Фон	
Изображение	Green

Рисунок 10.232 – Параметры индикатора Чтение/запись

11. Настроить Менеджер визуализации. В его установках следует поставить галочку **Использовать строки Unicode**:



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

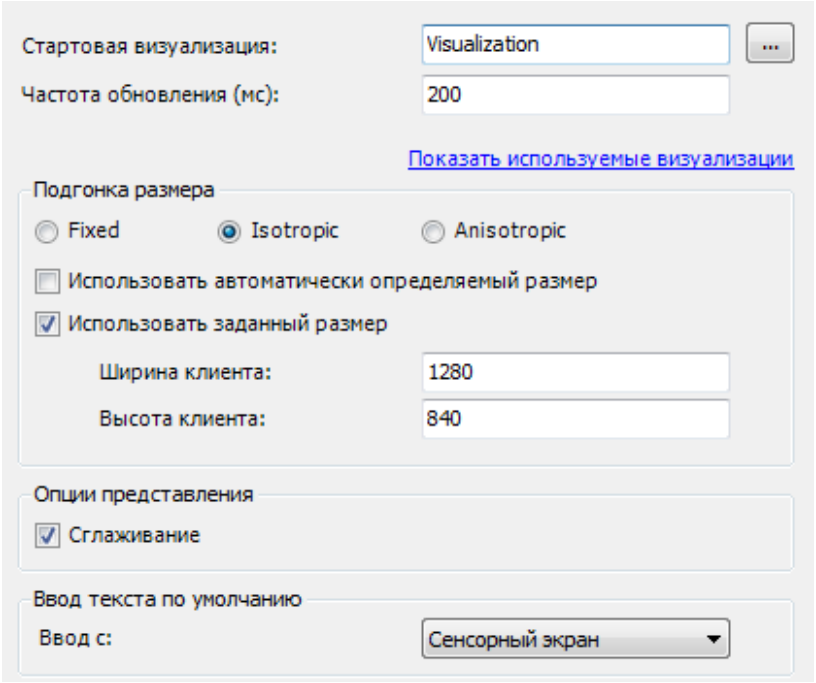


Рисунок 10.233 – Настройки таргет-визуализации

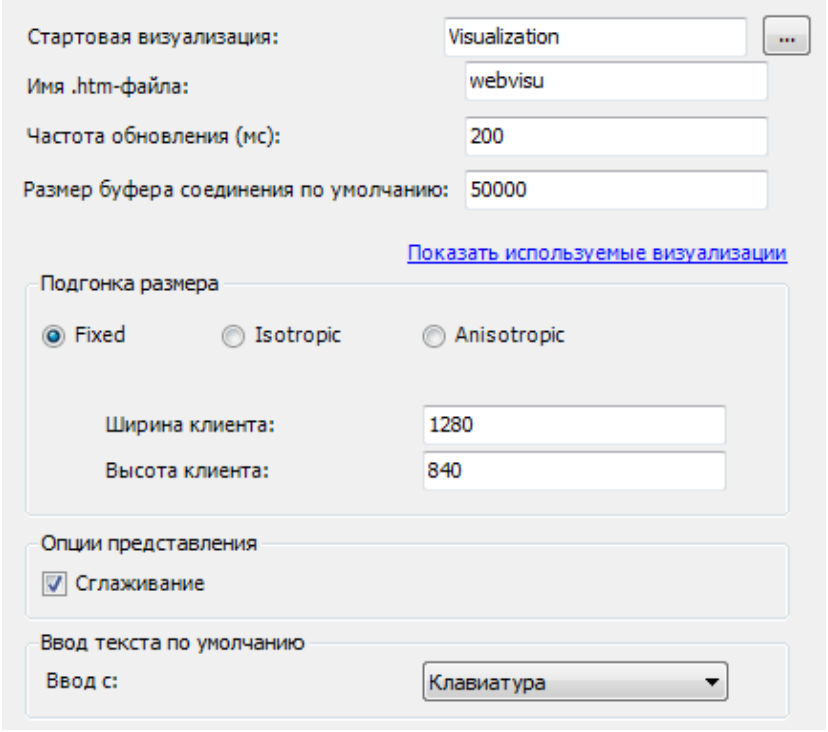


Рисунок 10.234 – Настройки web-визуализации

12. Запустить проект на виртуальном контроллере. Открыть в веб-браузере страницу http://localhost:8080/webvisu.htm, чтобы увидеть web-визуализацию проекта. ***Следует обратить внимание***, что браузер должен поддерживать **HTML5**. Проверить функционал проекта.

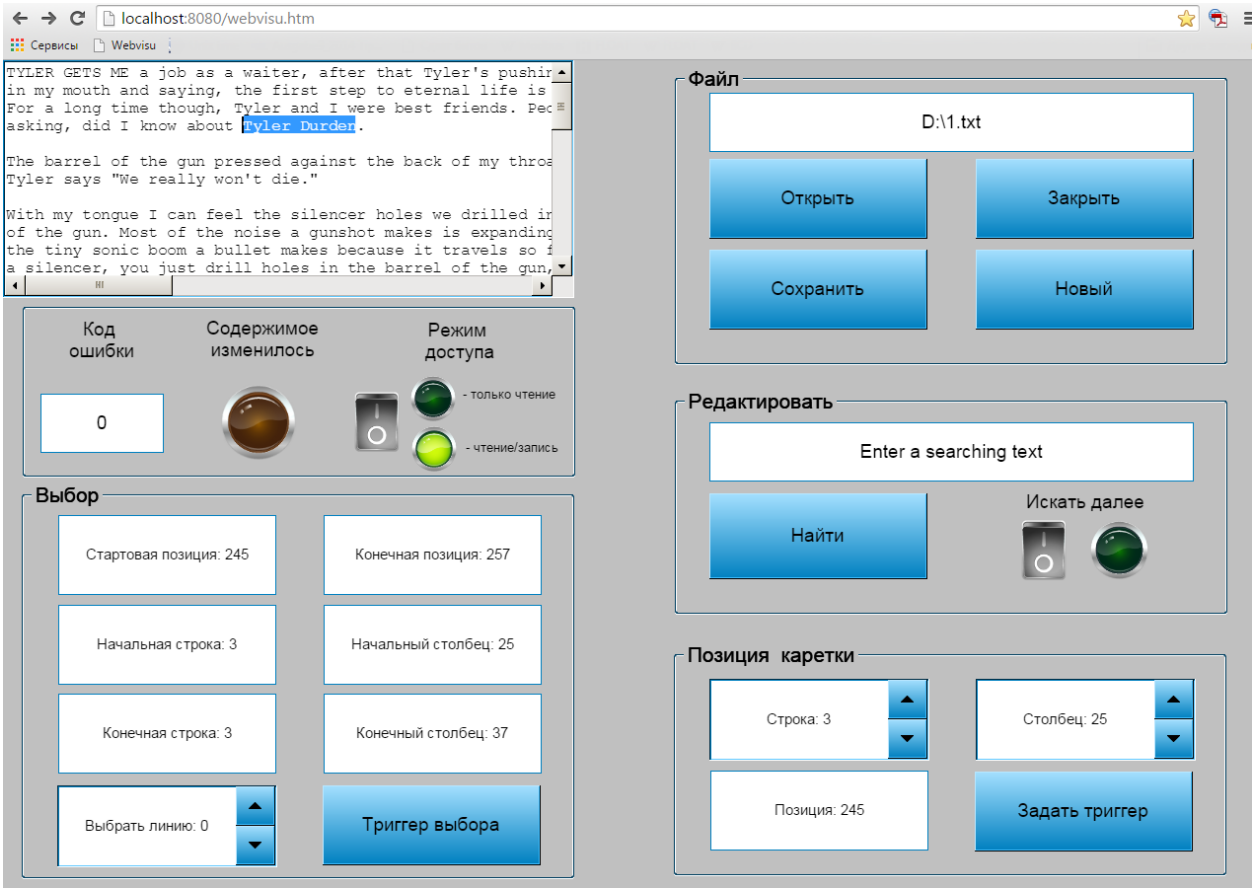


Рисунок 10.235 – Web-визуализация проекта в браузере Chrome

На панели **Файл** следует указать путь к файлу. Путь не должен содержать кириллических символов. Если файл содержит текст на русском языке, то должен быть сохранен в кодировке **Юникод**. ***Следует обратить внимание*** на ограничение длины строк в файле (таблица 8.29, пп. 2).

Нажатие на кнопку **Открыть** открывает выбранный файл в текстовом редакторе. Нажатие на кнопку **Закрыть** – закрывает. Нажатие на кнопку **Сохранить** – сохраняет изменения в данном файле. Нажатие на кнопку **Новый** приведет к созданию нового пустого текстового файла.

Во вкладке **Редактировать** следует ввести текстовый фрагмент, поиск которого будет произведен 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. Чтобы выделить одну из строк файла, следует выбрать номер линии и нажать кнопку **Триггер выбора.**

В поле **Код ошибки** отображается код последний ошибки, возникшей при работе с файлом. Например, в случае попытки 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 загорается 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.

10.2.13 Таблица тревог

Данный пример посвящен работе с графическим примитивом Таблица тревог, используемым для отображения состояний аварийных сигналов (тревог) с возможностью просмотра истории их срабатываний.

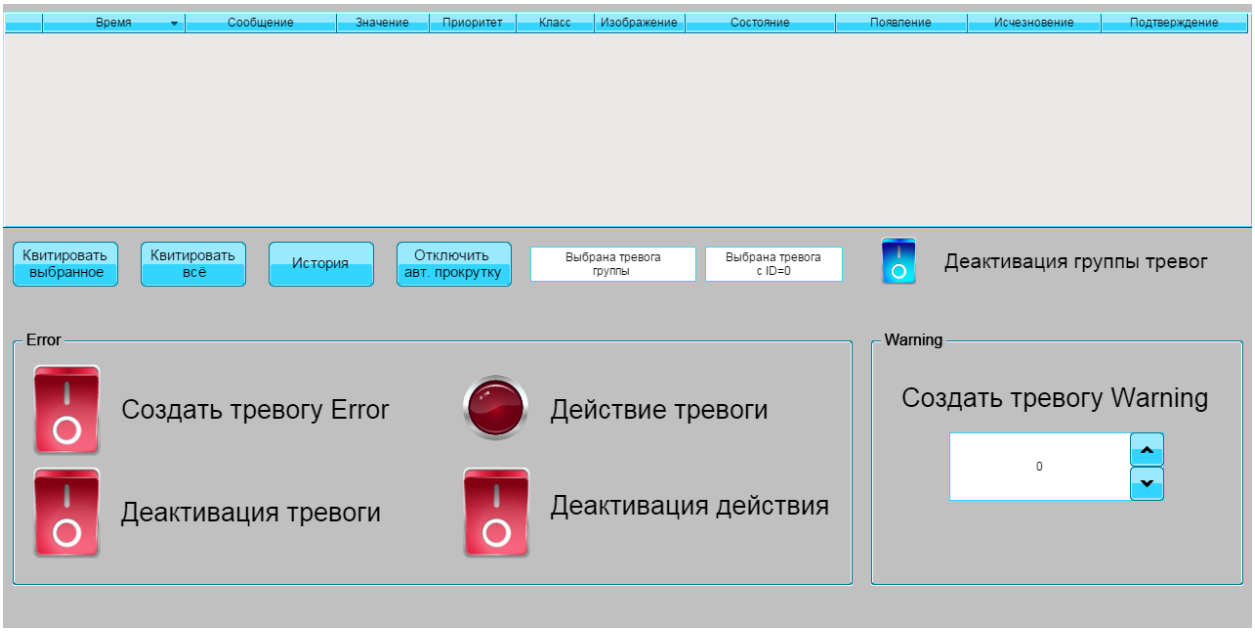


Рисунок 10.236 – Внешний вид примера Таблица тревог

Пример создан в среде **CODESYS V3.5 SP11 Patch 5** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с таргет-файлом **3.5.11.5**.

Расширенный пример создан в среде **CODESYS V3.5 SP16 Patch 3** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с таргет-файлом **3.5.16.30**.

Пример доступен для скачивания: Example_AlarmTable.projectarchive

Расширенная версия примера: Example_AlarmTableExtended.projectarchive

Продвинутый пример от компании **CODESYS Group** по работе с библиотекой менеджера тревог:

AlarmManager.package + видеопример

Файлы формата **.package** устанавливаются с помощью **Менеджера пакетов** (вкладка **Инструменты**) или (начиная с версии **CODESYS V3.5 SP17**) с помощью утилиты **CODESYS Installer**.

Для создания примера с использованием элемента **Таблица тревог** следует:

1. Создать новый ***стандартный*** проект **CODESYS** с названием **Example_AlarmTable** и настройками по умолчанию: таргет – CODESYS Control Win V3, язык PLC_PRG – **ST**.
2. В программе **PLC_PRG** объявить следующие переменные:

```
PROGRAM PLC_PRG
VAR
    bAlarm:                 BOOL;       // переменная тревоги типа BOOL (класс Error)
    bAlarmDeactivate:       BOOL;       // деактивации тревоги

    bAlarmLamp:             BOOL;       // переменная действия тревоги
    bAlarmLampOff:          BOOL;       // деактивация действия тревоги

    bAlarmGroupDeactivate:  BOOL;       // деактивация группы тревог

    iAlarm:                 INT;        // переменная тревоги типа INT (класс Warning)

    sSelectedAlarmGroupName:  STRING;   // имя группы, тревога которой выбрана в таблице в данный момент
    uiSelectedAlarmId:        UINT;     // ID тревоги, выбранной в таблице в данный момент

END_VAR
```

Рисунок 10.237 – Объявление переменных программы PLC_PRG

3. Добавить в проект компонент Alarm Configuration:

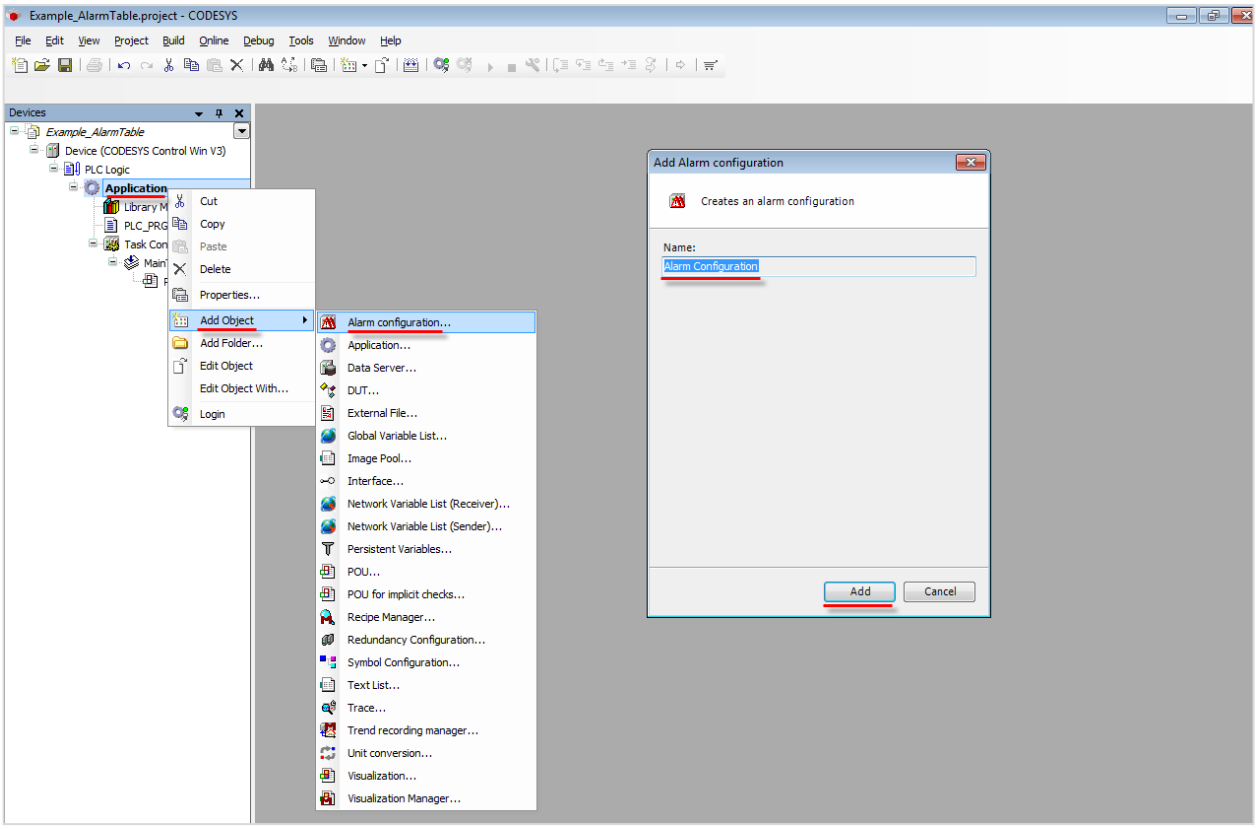


Рисунок 10.238 – Добавление компонента Alarm Configuration

Панель устройств после добавления компонента будет выглядеть следующим образом:

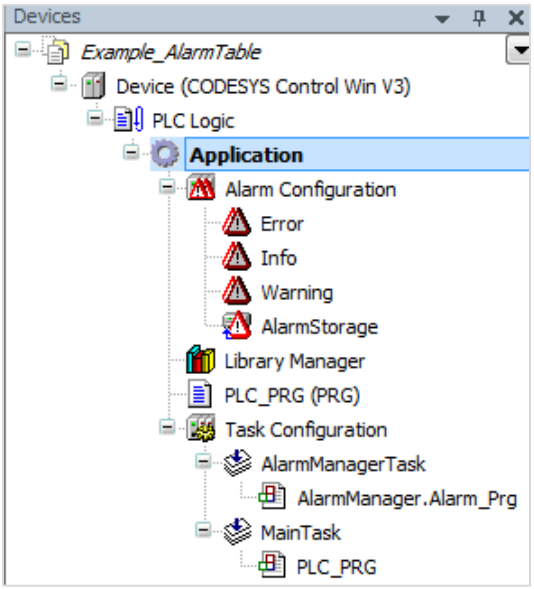


Рисунок 10.239 – Панель устройств после добавление компонента Alarm Configuration

4. Создать в Alarm Configuration группу тревог с названием **Alarm Group**. Автоматически будет создан список текстов с тем же названием.

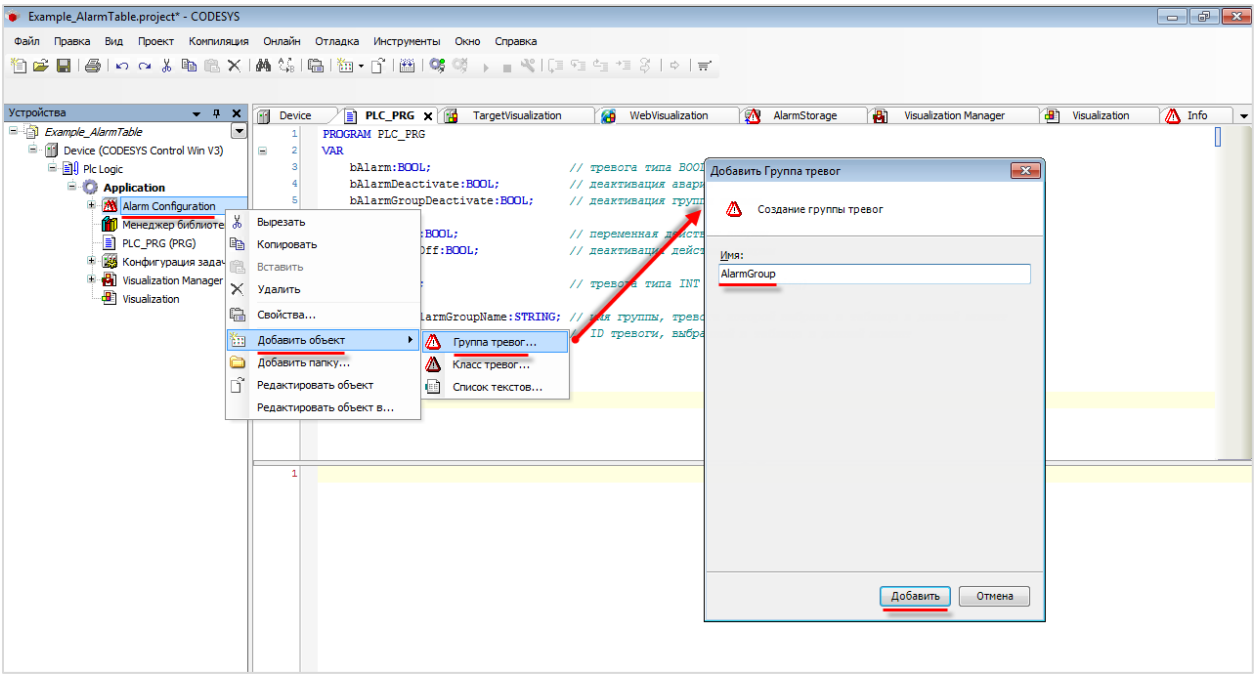


Рисунок 10.240 – Создание группы тревог AlarmGroup

Затем следует удалить класс тревог **Info**, так как он не используется в примере:

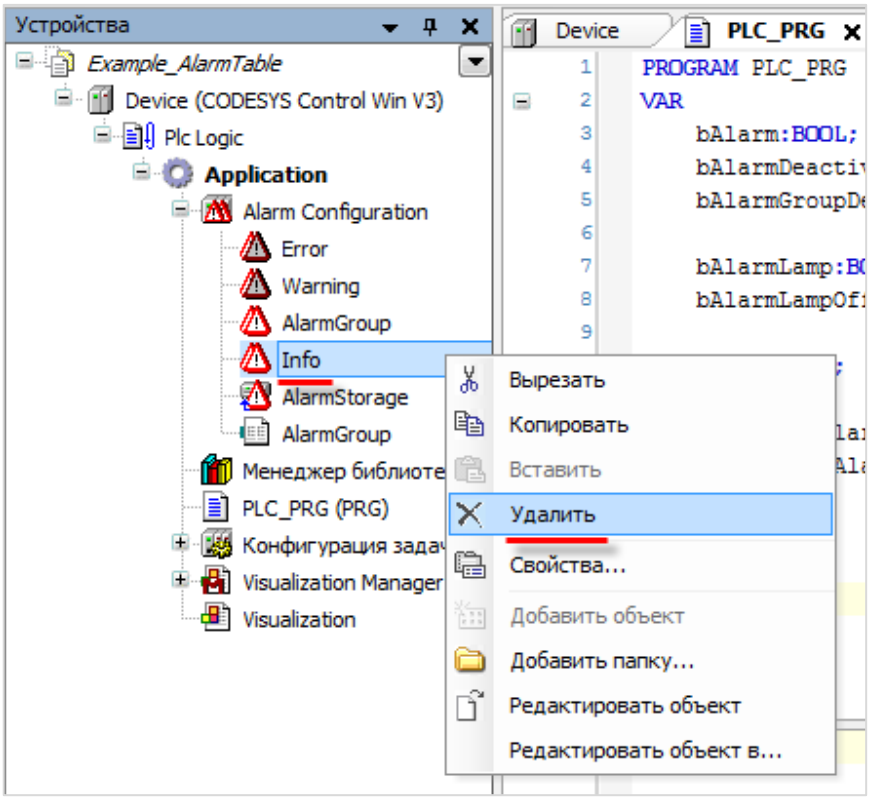


Рисунок 10.241 – Удаление класса тревог Info

5. Настроить класс тревог **Error**:

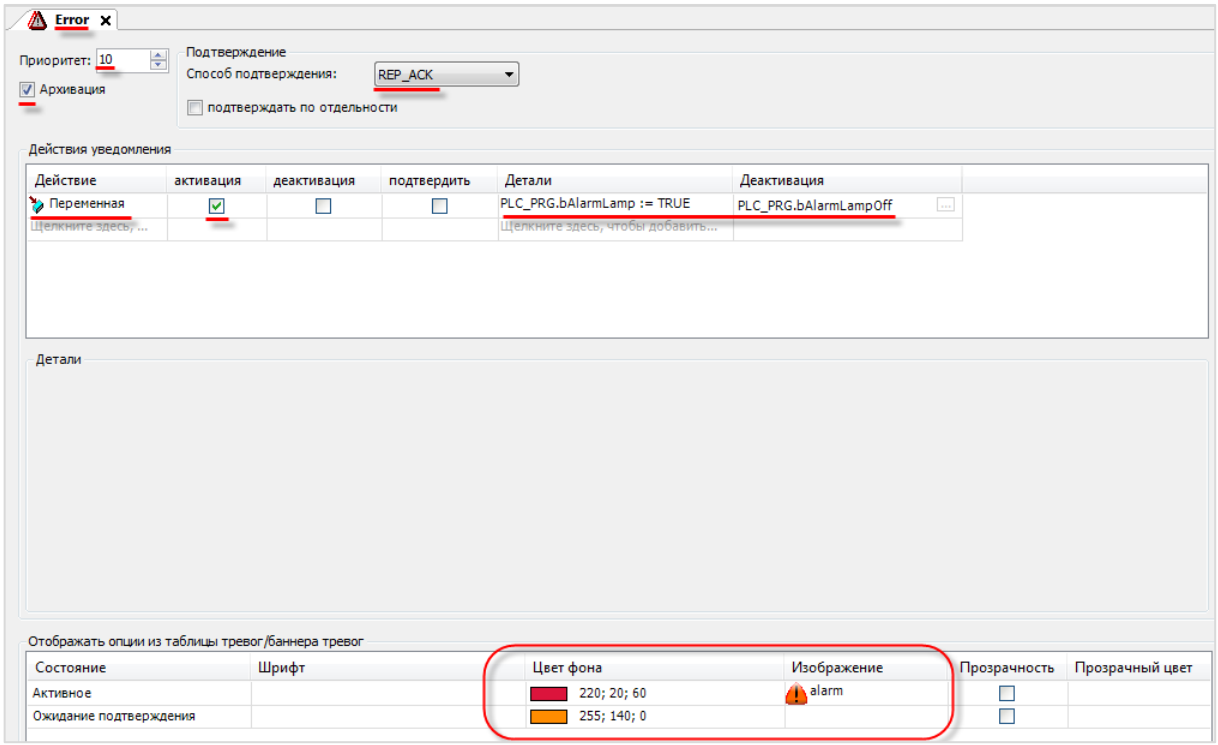


Рисунок 10.242 – Настройки класса тревог Error

Для добавления изображения следует ввести любое название, нажать **Enter** и указать путь к графическому файлу.

6. Настроить класс тревог **Warning**:

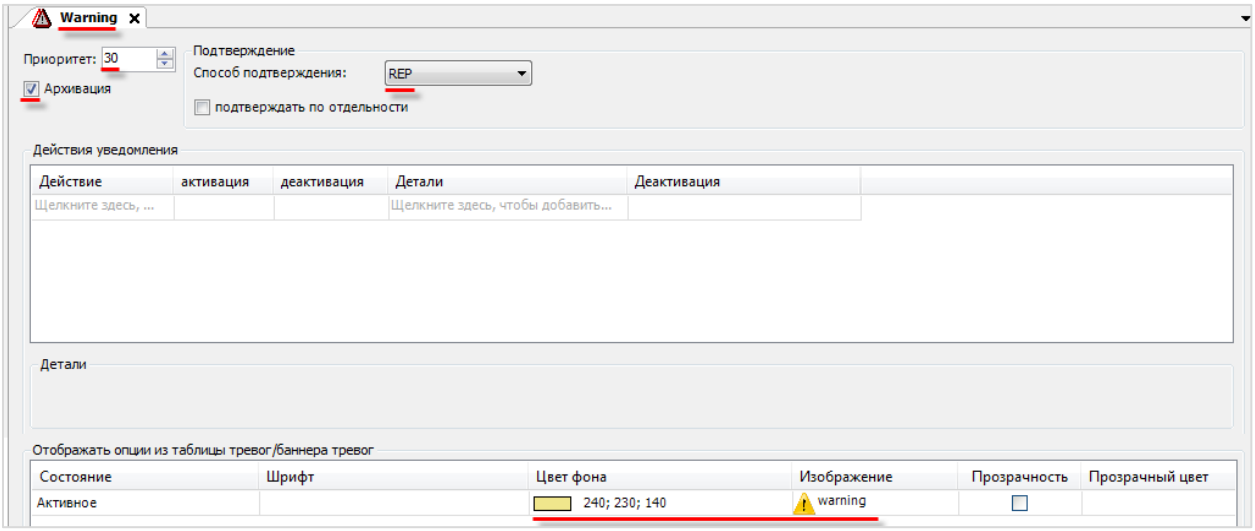


Рисунок 10.243 – Настройки класса тревог Warning

7. Настроить группу тревог **Alarm Group**:

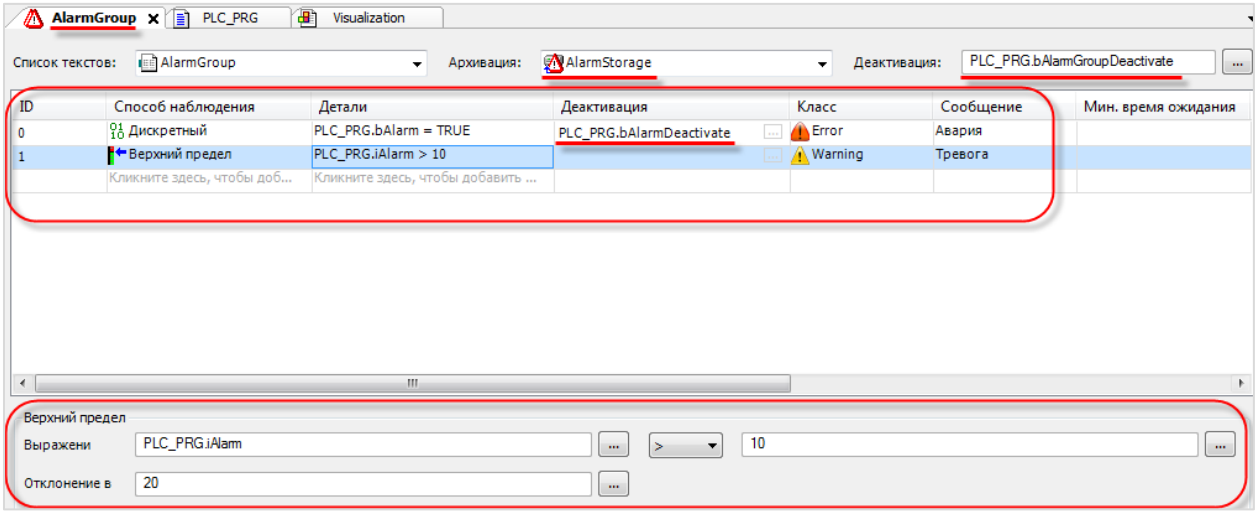


Рисунок 10.244 – Настройки группы тревог AlarmGroup

Настройка деталей каждой тревоги происходит на панели, расположенной под компонентом.

8. Добавить в проект экран визуализации **Visualization**. В его свойствах выбрать размер **1280 × 640.** 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.

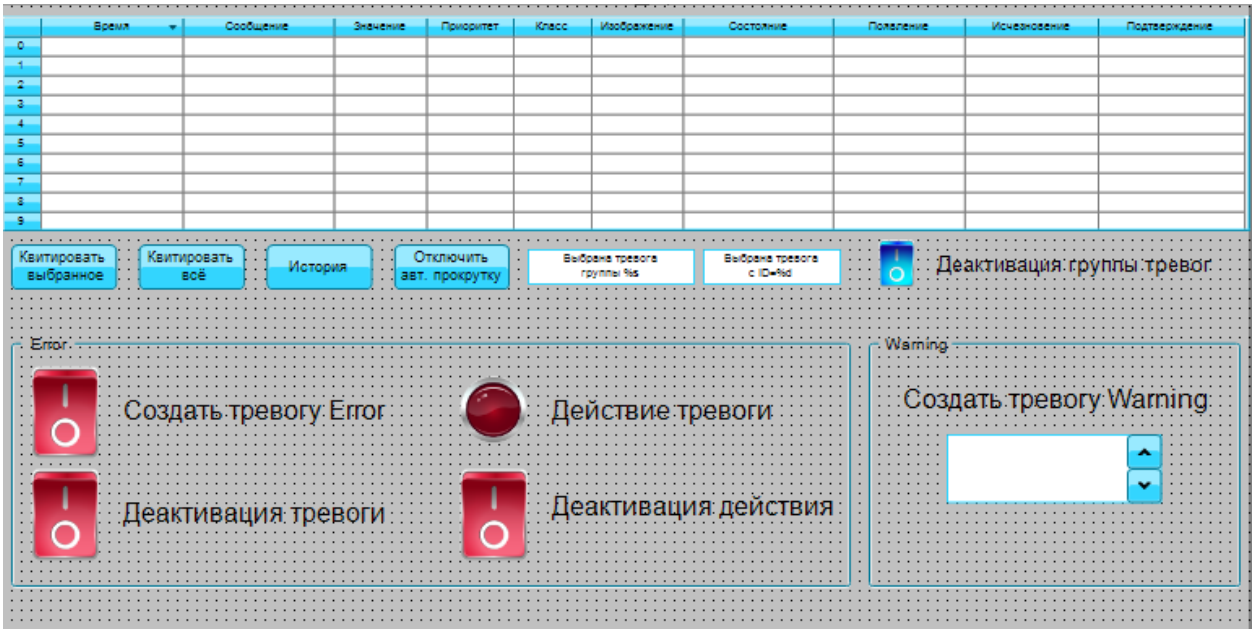


Рисунок 10.245 – Содержание экрана Visualization

9. Настроить 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	☑
Показать заголовок столбца	☑
Высота строки	20
Ширина строки заголовка	40
Размер полосы прокрутки	20
Автоматический разрыв строки в сообщениях тревог	☑

Рисунок 10.246 – Настройки Таблицы тревог (начало)

Столбцы таблицы создаются с помощью кнопки **Создать новый** во вкладке **Столбцы**:

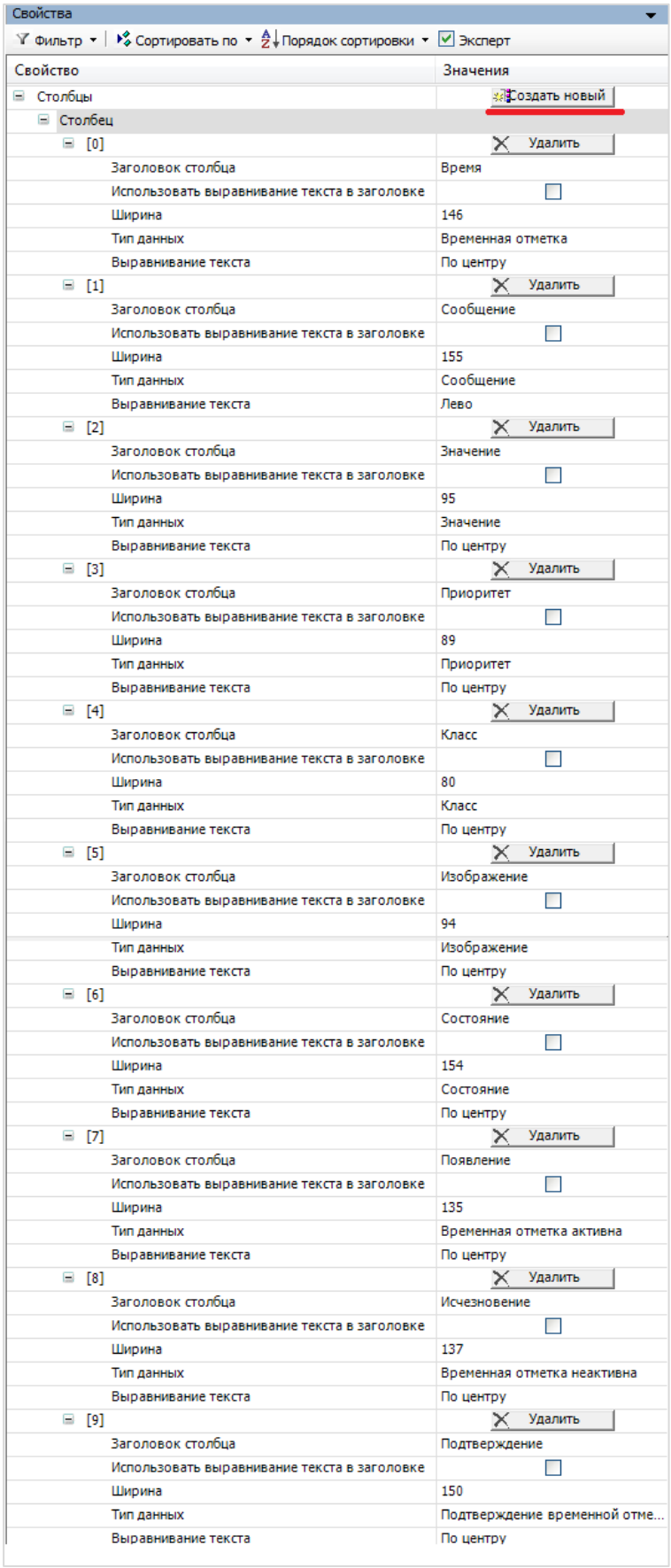


Рисунок 10.247 – Настройки Таблицы тревог (продолжение)

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	☑
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

Рисунок 10.248 – Настройки Таблицы тревог (окончание)

Управляющие переменные таблицы и их кнопки можно создать автоматически – для этого следует нажать на таблицу **ПКМ** и в контекстном меню выбрать пункт **Вставить элементы для подтверждения тревог**:

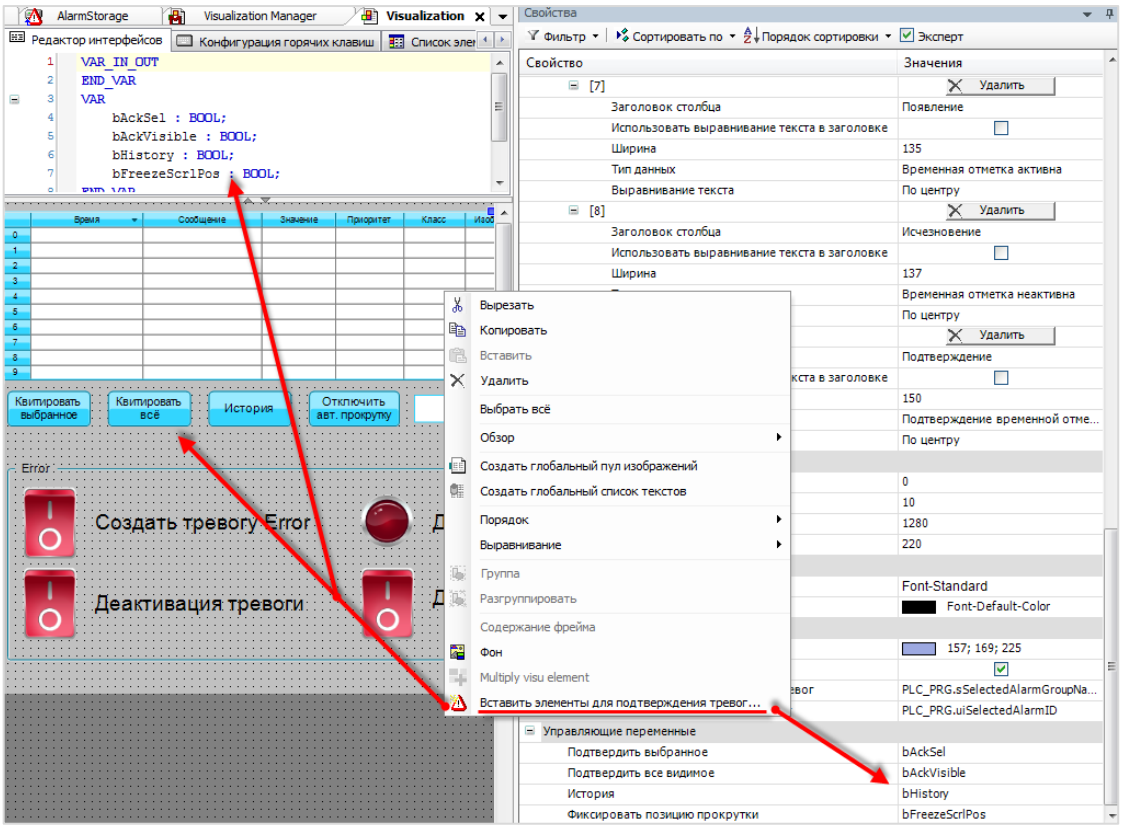


Рисунок 10.249 – Создание кнопок подтверждения тревог

10. У кнопок **История** и **Отключить авт.прокрутку** следует привязать переменные состояния (они будут соответствовать переключаемым переменным), чтобы видеть, когда активны соответствующие режимы:

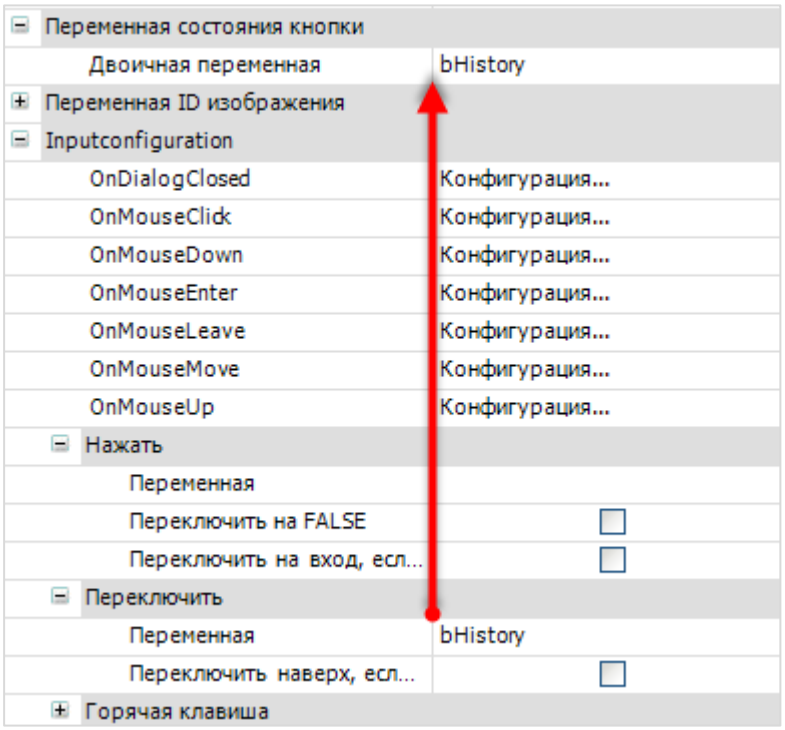


Рисунок 10.250 – Настройки кнопки История

Для кнопки **Отключить авт. прокрутку** двоичной переменной будет **bFreezeScrlPos**.

11. Настроить элементы Текстовое поле, используемые для вывода информации о выделенной строке таблицы:

Рисунок 10.253 – Элементы Текстовое поле 1 и 2

Свойства

Фильтр | Сортировать по | Порядок сортировки | Эксперт

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	

Рисунок 10.251 – Настройки элемента Текстовое поле 1

Свойства

Фильтр | Сортировать по | Порядок сортировки | Экспер

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	

Рисунок 10.252 – Настройки элемента Текстовое поле 2

12. К клавишному переключателю **Деактивация таблицы** привязывается переменная **bAlarmGroupDeactivate**:

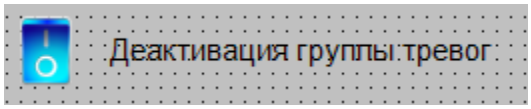


Рисунок 10.253 – Переключатель деактивации группы тревог

Свойства

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

Рисунок 10.254 – Настройки элемента Клавишный переключатель

13. Настроить элементы группы **Error**, используемые для управления тревогой одноименного класса:

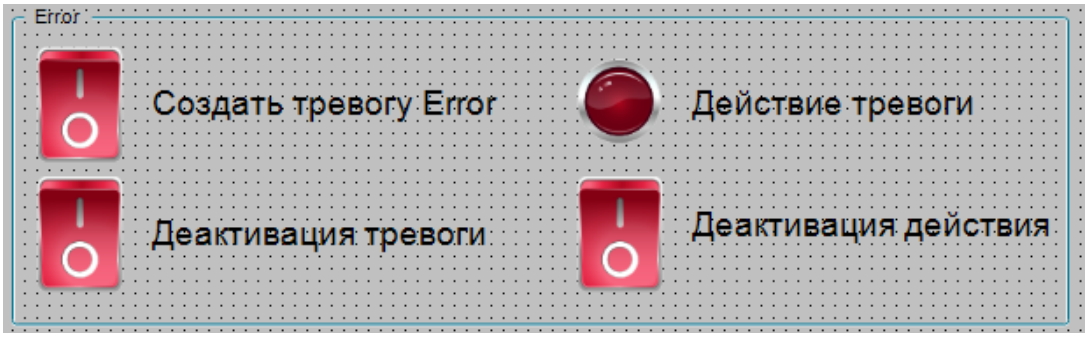


Рисунок 10.255 – Элементы группы Error

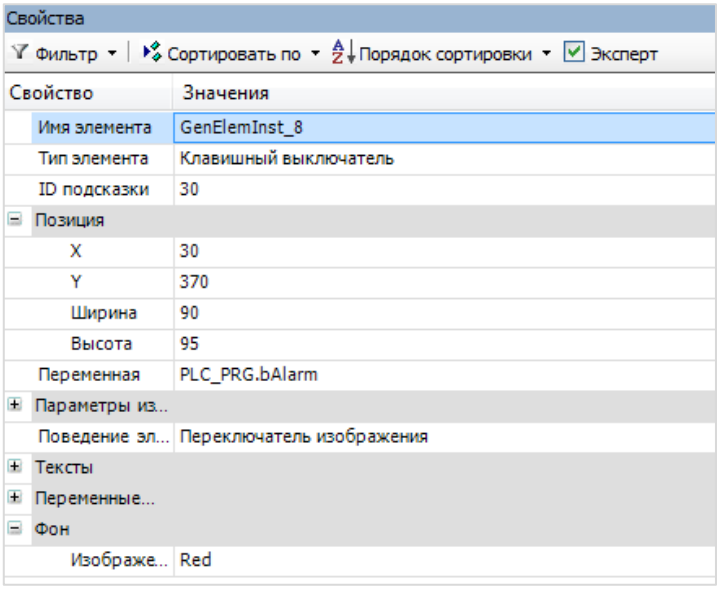


Рисунок 10.256 – Настройки элемента Клавишный выключатель 1

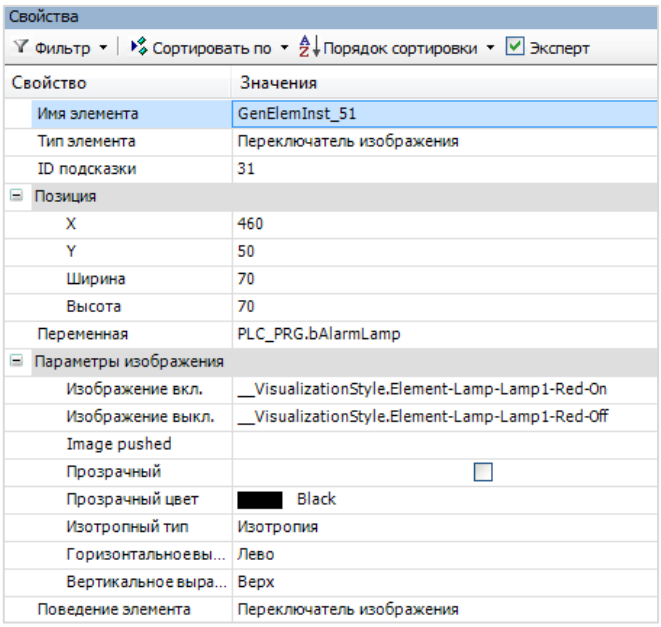


Рисунок 10.257 – Настройки элемента Переключатель изображения

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

Рисунок 10.258 – 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

Рисунок 10.259 – Настройки элемента Клавишный выключатель 3

14. Настроить элементы группы **Warning**, используемые для управления тревогой одноименного класса:

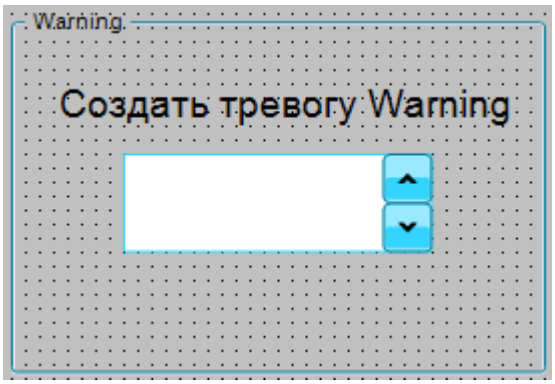


Рисунок 10.260 – Элементы группы Warning

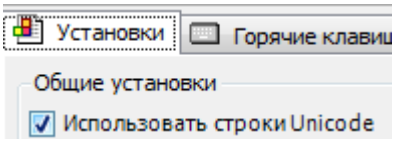
Свойства

Фильтр | Сортировать по | Порядок сортировки | Эксперт

Свойство	Значения
Имя элемента	GenElemInst_47
Тип элемента	Управление вращением
Позиция	
X	970
Y	440
Ширина	220
Высота	70
Переменная	PLC_PRG.iAlarm
Числовой формат	%d
Интервал	1

Рисунок 10.261 – Настройки элемента Управление вращением

15. Настроить Менеджер визуализации. В его установках следует поставить галочку **Использовать строки Unicode**:



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

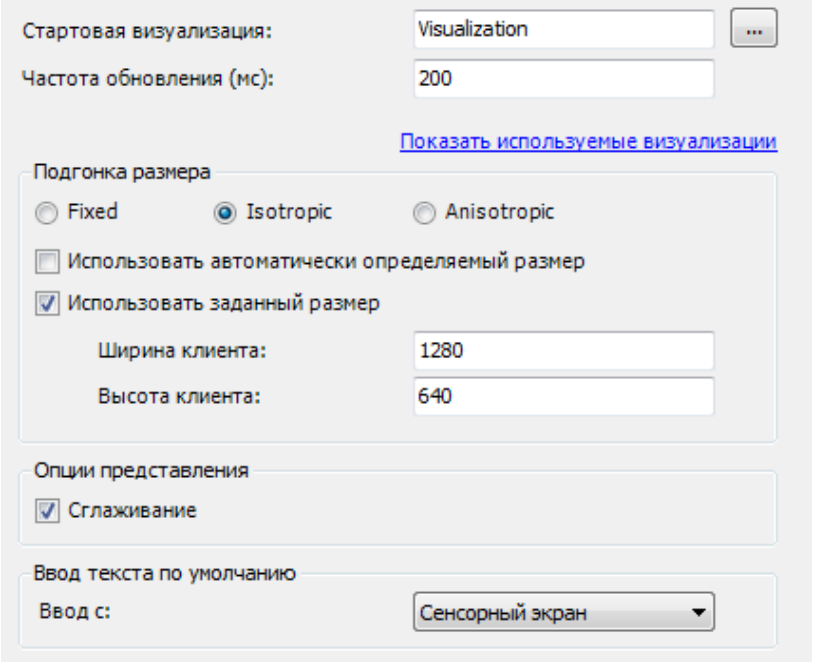


Рисунок 10.262 – Настройки таргет-визуализации

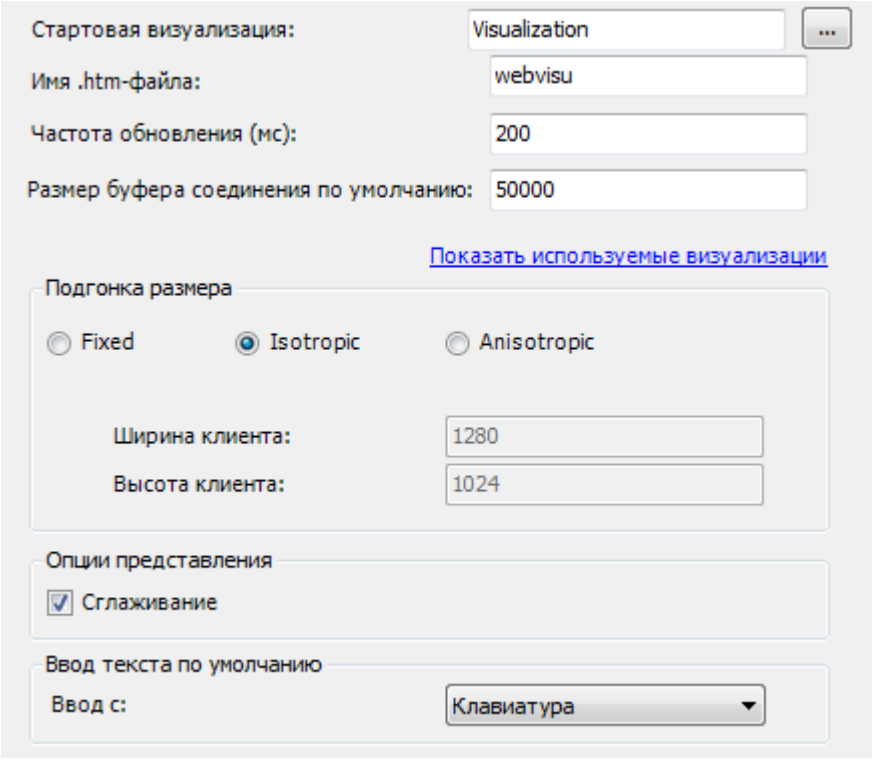


Рисунок 10.263 – Настройки web-визуализации

16. Запустить проект на виртуальном контроллере. Открыть в веб-браузере страницу http://localhost:8080/webvisu.htm, чтобы увидеть web-визуализацию проекта. ***Следует обратить внимание***, что браузер должен поддерживать **HTML5**. Проверить функционал проекта.

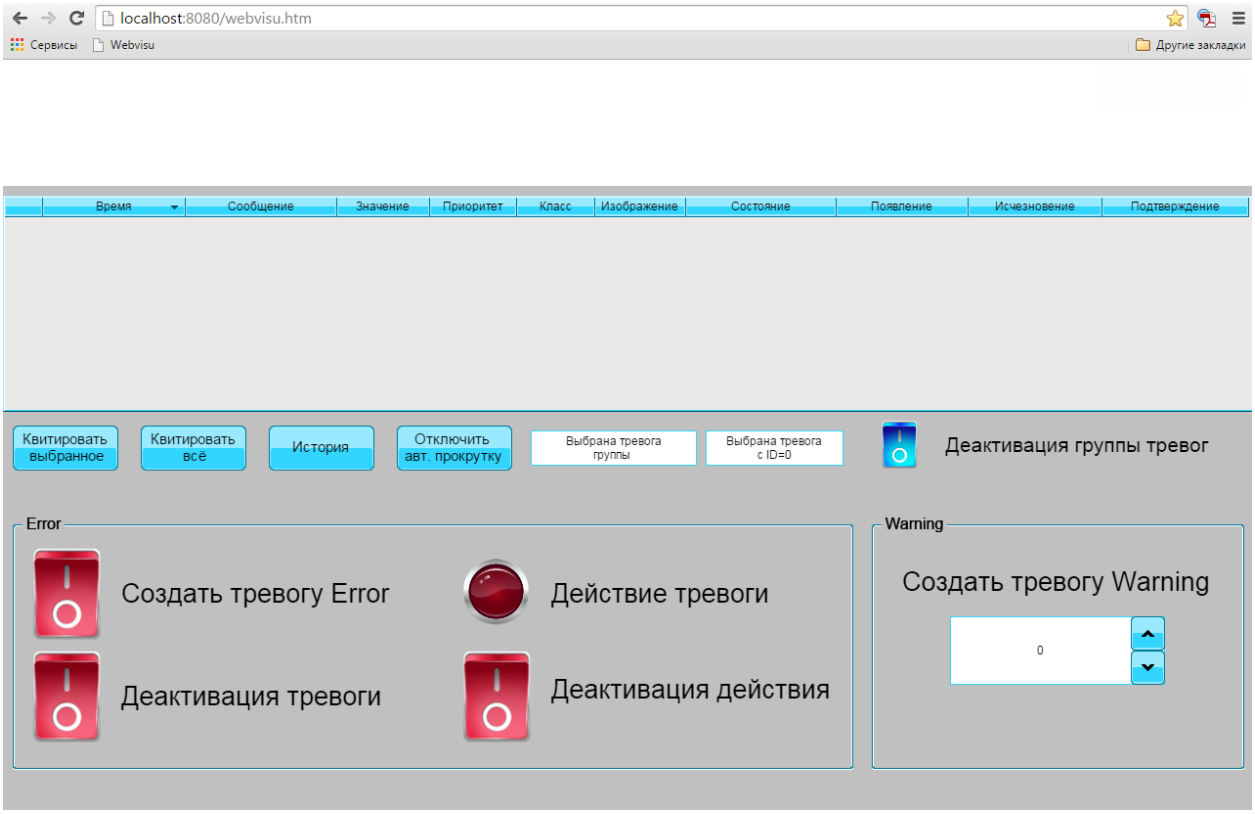


Рисунок 10.264 – Web-визуализация проекта в браузере Chrome

Сначала следует нажать переключатель **Создать тревогу Error**. Элемент **Действие тревоги** загорится красным, 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		

Рисунок 10.268 – Отображение в таблице активной тревоги класса Error

Затем следует нажать на элемент **Действие тревоги**, чтобы отключить его. Повторно нажать 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	

Рисунок 10.265 – Отображение в таблице тревоги, ожидающей подтверждения

Нажать на строку таблицы, чтобы выделить ее. В элементах **Текстовое поле** отобразится информация о выделенной тревоге:

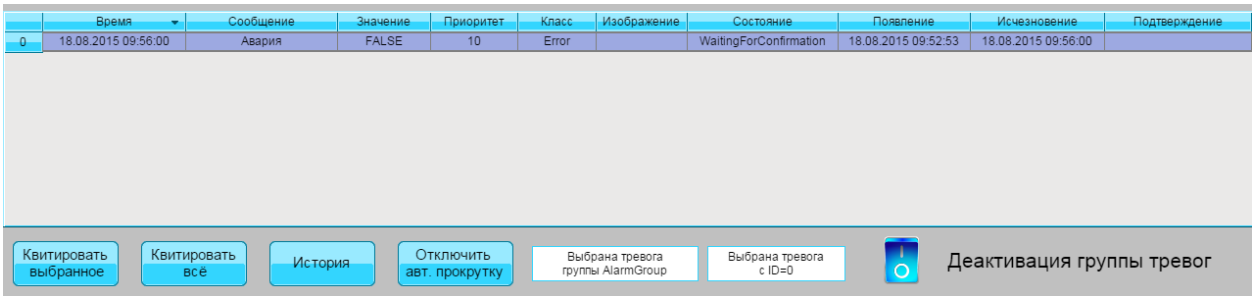


Рисунок 10.266 – Отображение в таблице выделенной тревоги

Нажать 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:

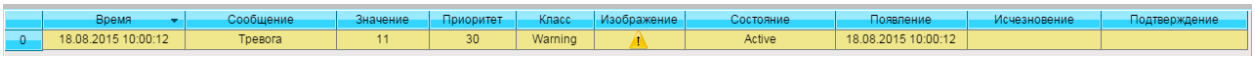


Рисунок 10.267 – Отображение в таблице активной тревоги класса Warning

Затем следует ввести значение **9**. Тревога не пропадет, но значение переменной теперь не превышает **10**. Это связано с тем, для тревоги был задан гистерезис в **20 %** от значения срабатывания (см. рисунок 10.244). Соответственно, чтобы отключить тревогу, значение переменной должно уменьшиться как минимум до **80 %** от значения срабатывания. Чтобы тревога исчезла из таблицы следует ввести значение **8**.

Чтобы посмотреть историю тревог, следует нажать кнопку **История**:

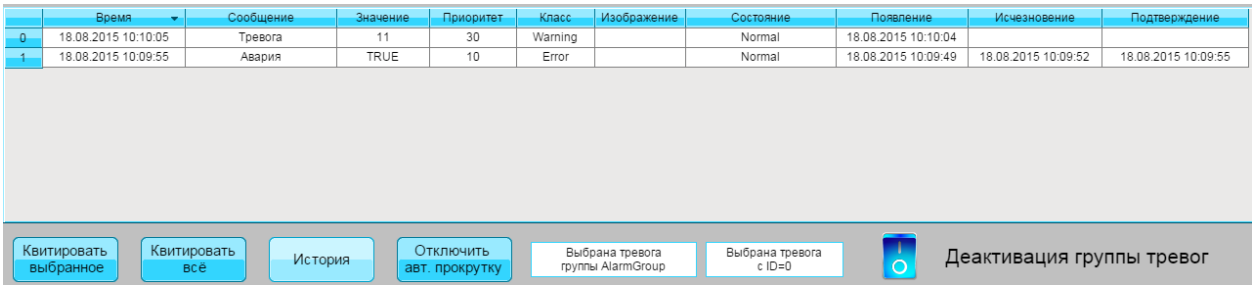


Рисунок 10.268 – Таблица тревог в режиме просмотра истории

В столбце **Время** отображается время последнего изменения состояния тревоги. Для тревоги класса **Warning** столбец **Исчезновение** остается пустым, время исчезновения отображается в столбце Время – это связано с тем, что для нее выбран способ подтверждения **REP** (см. рисунок 10.243), который подразумевает только одно состояние тревоги (активна).

Переключатели деактивации работают следующим образом:

1. Во время деактивации группы тревог, сообщения данной группы тревог не отображаются в таблице и ее истории.
2. Во время деактивации тревоги, сообщения данной тревоги не отображаются в таблице и ее истории.
3. Во время деактивации действия, действия данной тревоги не выполняются при ее активации.

10.3 Примеры решения типовых задач

10.3.1 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**.

Следует обратить внимание, что в случае использования в проекте системной переменной **CurrentVisu**, переключение экранов будет затрагивать **всех клиентов визуализации**. Соответственно, для многопользовательского доступа использование этой переменной крайне нежелательно. См. более подробную информацию в п. 10.4.4.

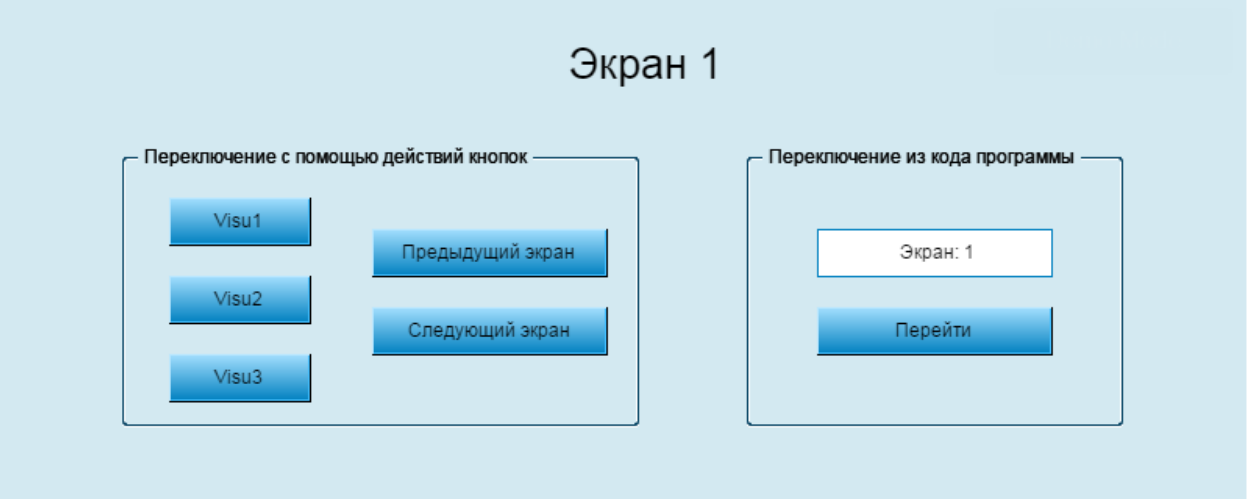


Рисунок 10.269 – Внешний вид примера Переключение экранов

Пример создан в среде **CODESYS V3.5 SP11 Patch 5** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с таргет-файлом **3.5.11.5**.

Пример доступен для скачивания: Example_VisuSwitch.projectarchive

Для создания примера с функцией переключения экранов следует:

1. Создать новый ***стандартный*** проект **CODESYS** с названием **Example_VisuSwitch** и настройками по умолчанию: таргет – CODESYS Control Win V3, язык PLC_PRG – **ST**.
2. В программе **PLC_PRG** объявить переменную **iVisuIndex** типа **INT**:

```
1  PROGRAM PLC_PRG
2  VAR
3      iVisuIndex:INT;  // индекс экрана визуализации, на который осуществляется переход
4  END_VAR
```

Рисунок 10.270 – Объявление переменных программы PLC_PRG

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

```
// в зависимости от числа, введенного пользователем...
// ...происходит переход на соответствующий экран визуализации

CASE iVisuIndex OF
    1: VisuElems.CURRENTVISU:='Visu1';
    2: VisuElems.CURRENTVISU:='Visu2';
    3: VisuElems.CURRENTVISU:='Visu3';
END_CASE

// сброс индекса экрана после перехода для возможности...
// ...переключения экранов действиями кнопок

iVisuIndex:=0;
```

Рисунок 10.271 – Код программы PLC_PRG

ПРИМЕЧАНИЕ

Использование системной переменной **VisuElems.CurrentVisu** возможно только если установлена соответствующая галочка в установках Менеджера визуализации – переключение экранов визуализации затрагивает всех клиентов, что делает нежелательным использование данной переменной в проектах с возможностью многопользовательского доступа.

4. Добавить в проект три экрана визуализации с названиями **Visu1, Visu2** и **Visu3**. В их свойствах выберем размер **800 × 480.** На каждом из экранов добавить элемент Метка с названием экрана (**Экран 1**, **Экран 2**, **Экран 3**). Выбрать для каждого из экранов свой цвет фона.

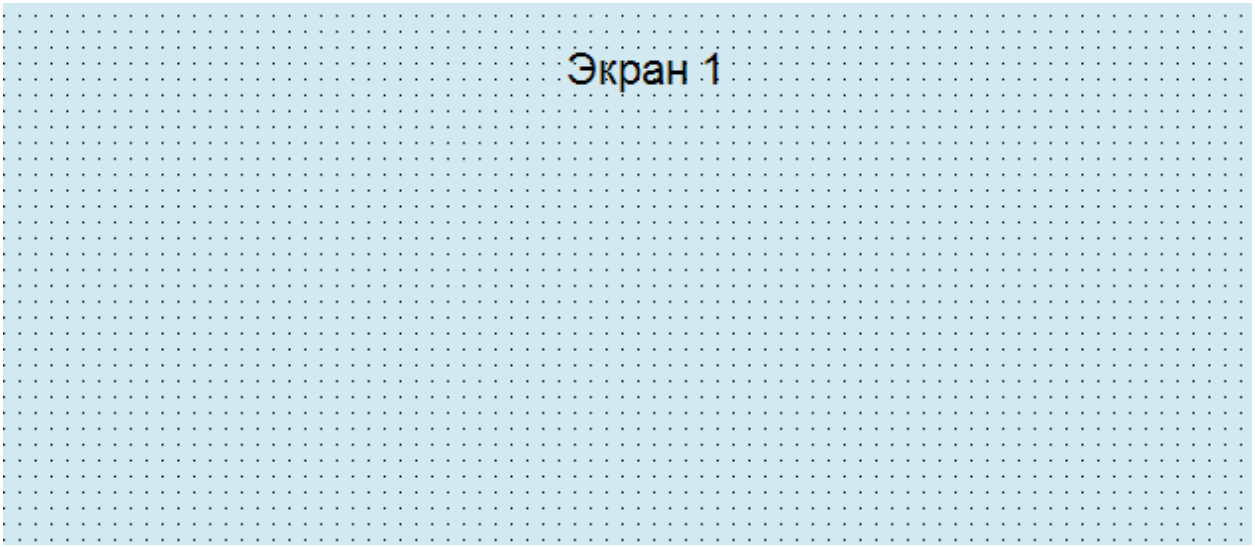


Рисунок 10.272 – Экран визуализации Visu1

5. В интерфейсе каждого из экранов объявить локальную переменную **iSelect** типа **INT**, начальное значение которой будет совпадать с номером экрана:

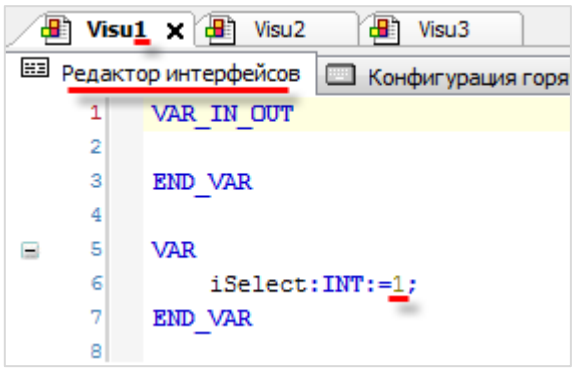


Рисунок 10.273 – Интерфейс экрана визуализации Visu1

Соответственно, для экрана **Visu2** переменная **iSelect** будет иметь начальное значение **2**, для **Visu3** – **3**.

6. Экран **Visu1** будет содержать два элемента Группа, шесть элементов Кнопка и элемент Текстовое поле:

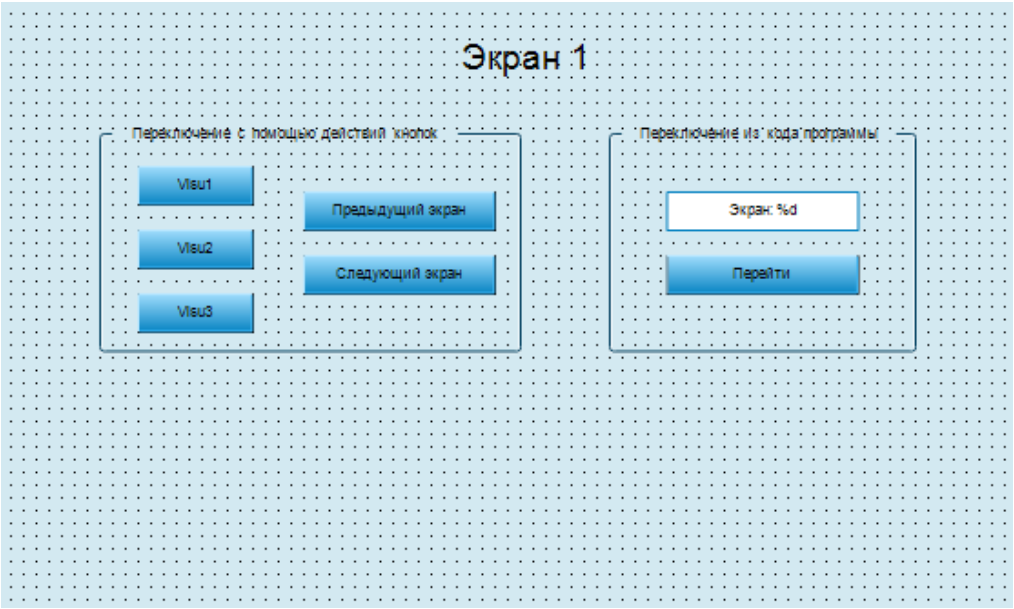


Рисунок 10.274 – Содержимое экрана визуализации Visu1

Сначала следует настроить кнопки **Visu1**, **Visu2**, **Visu3**. В свойствах элементов во вкладке InputConfiguration выбрать параметр **OnMouseClick** и привязать к нему действие Изменить отображаемую визуализацию, которому присваивается название соответствующего экрана:

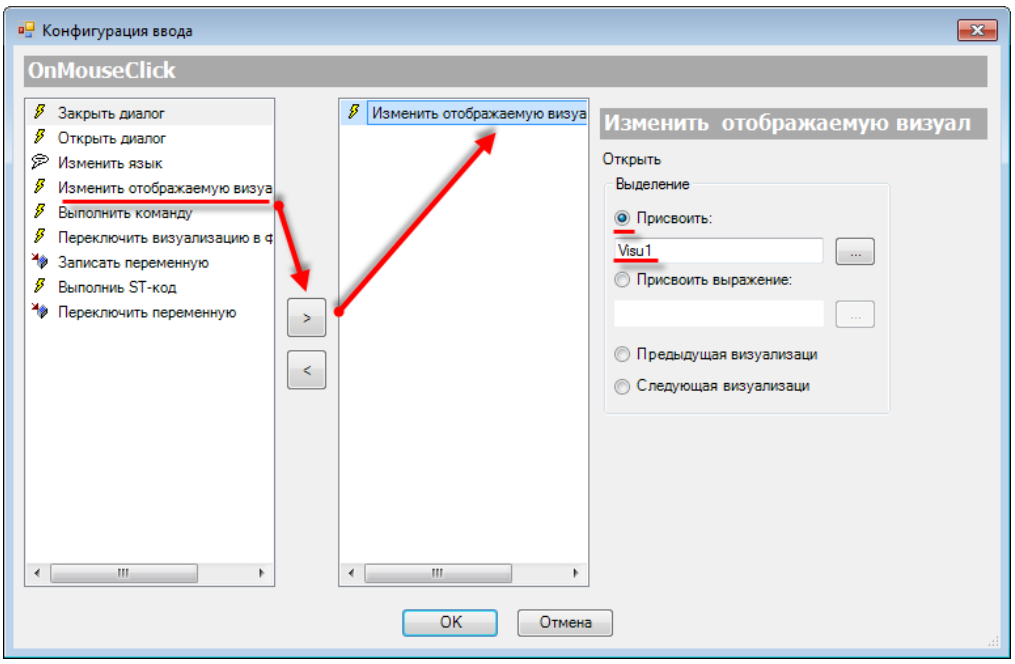


Рисунок 10.275 – Настройка действия кнопки Visu1

Настройки кнопок **Visu2** и **Visu3** аналогичны.

Действия кнопок **Предыдущая визуализация** и **Следующая визуализация**, соответственно, будут выглядеть следующим образом:

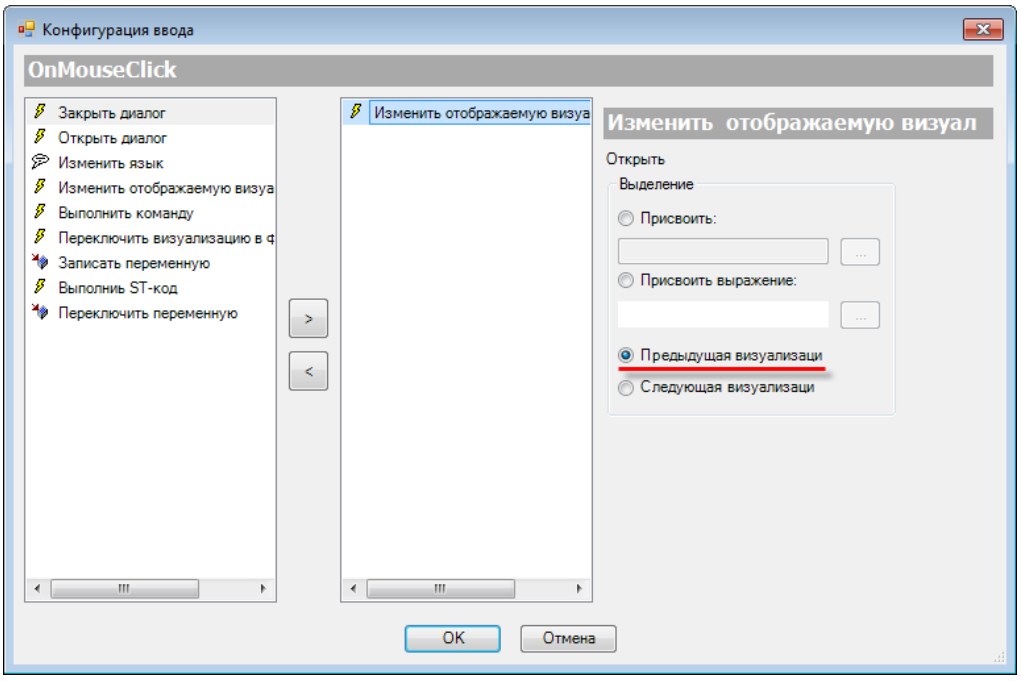


Рисунок 10.276 – Настройки действия кнопки Предыдущая визуализация

Для кнопки **Следующая визуализация** следует выбрать пункт **Следующая визуализация**.

Настройки элемента Текстовое поле будут выглядеть следующим образом:

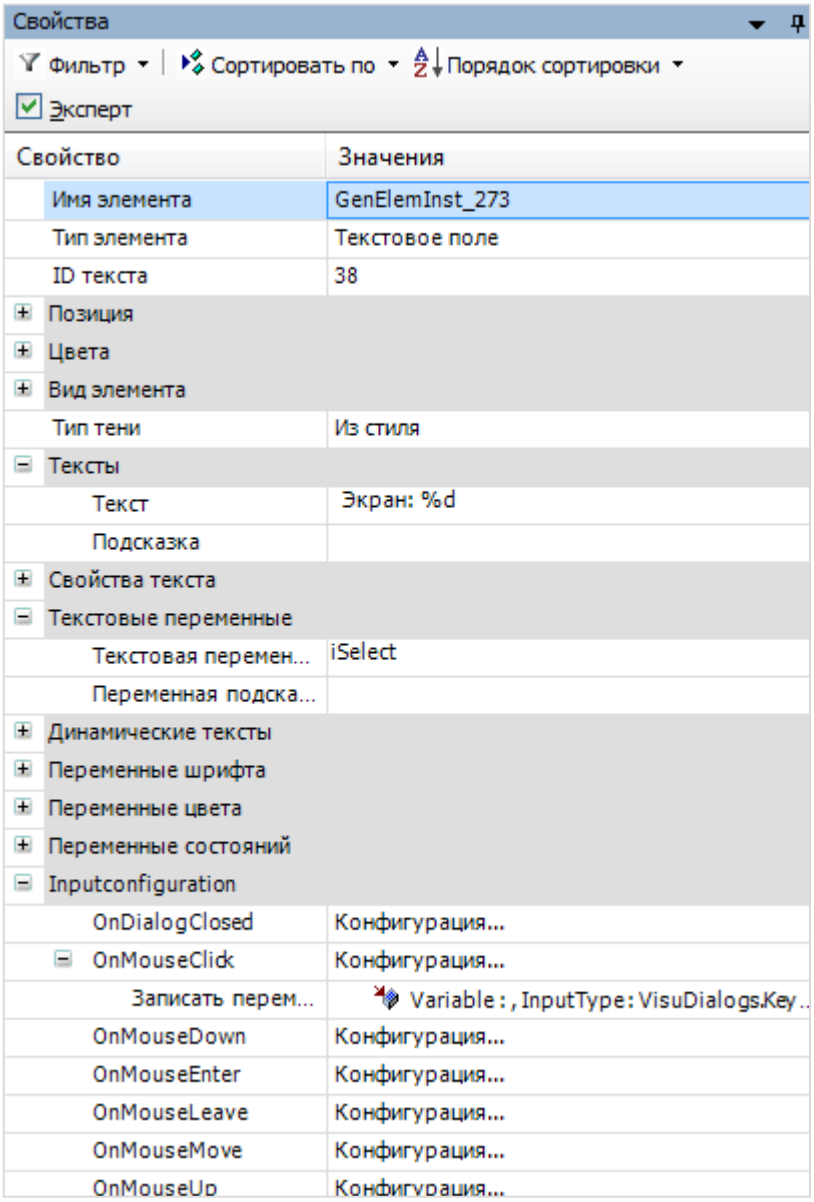


Рисунок 10.277 – Параметры элемента Текстовое поле

В свойствах Текстового поля во вкладке InputConfiguration к параметру **OnMouseClick** следует привязать действие Записать переменную со следующими настройками:

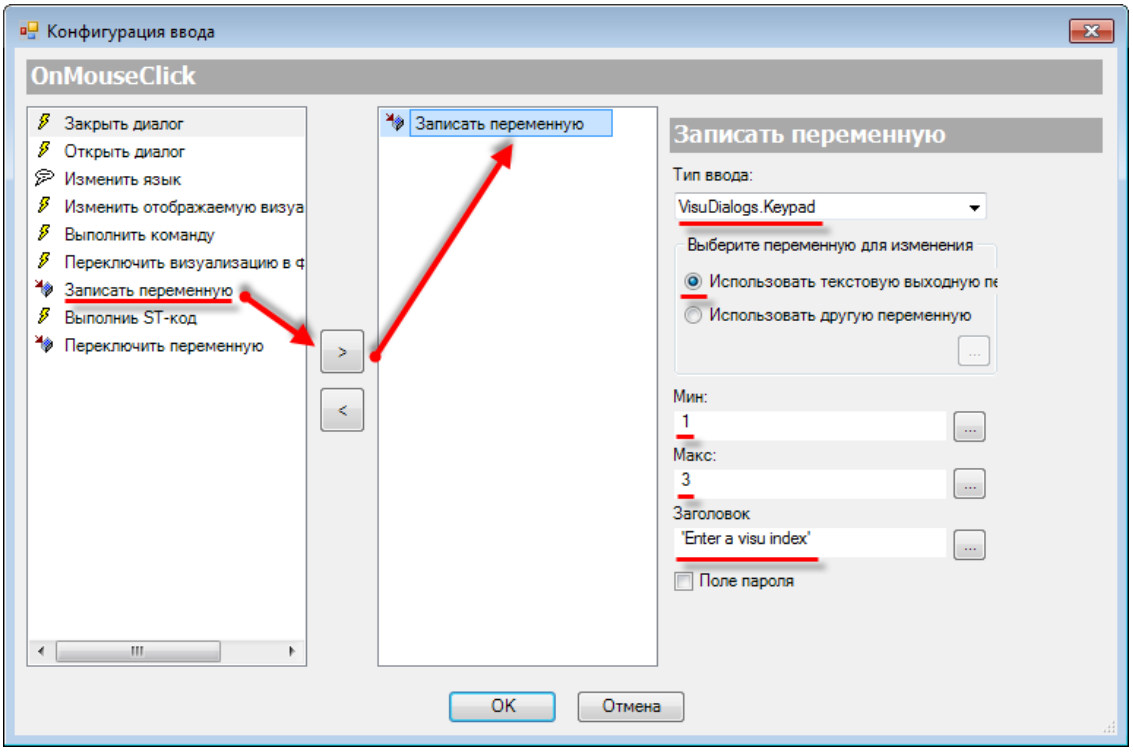


Рисунок 10.278 – Настройки действия элемента Текстовое поле

В свойствах кнопки **Применить** во вкладке InputConfiguration к параметру **OnMouseClick** следует привязать действие Выполнить ST-код со следующими настройками:

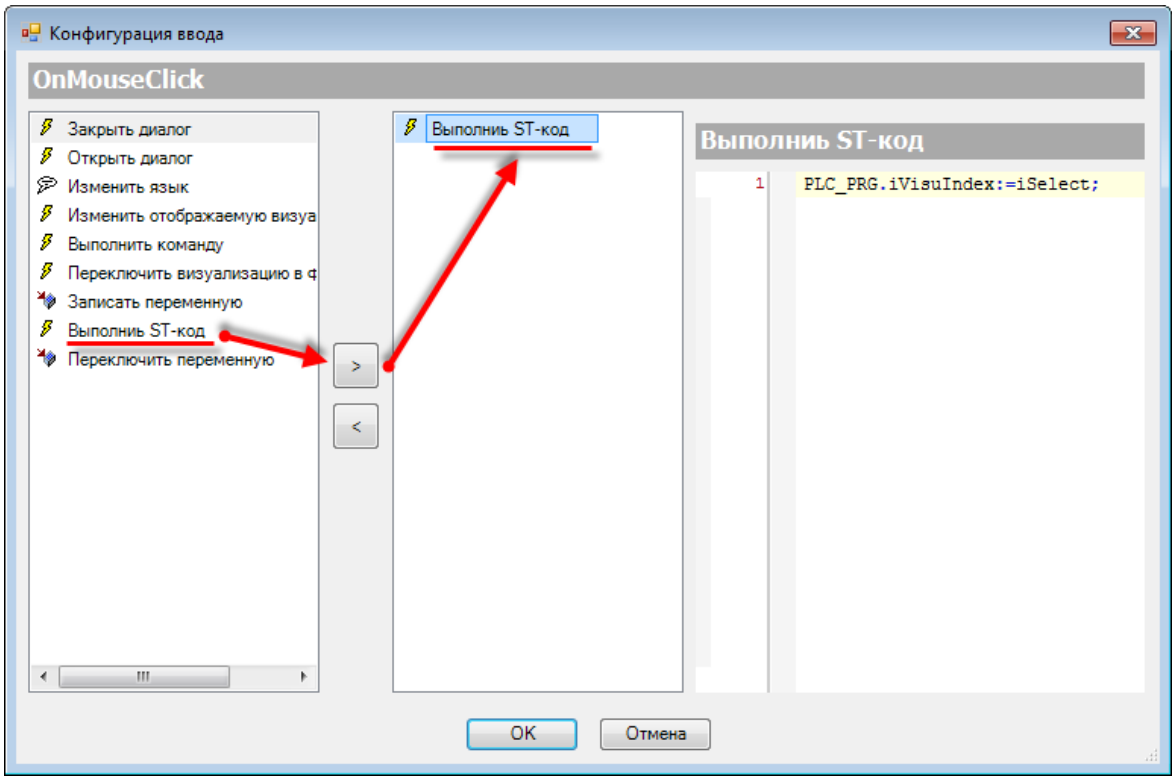


Рисунок 10.279 – Настройки действия кнопки Применить

7. Скопировать содержимое экрана **Visu1** на экраны **Visu2** и **Visu3**:

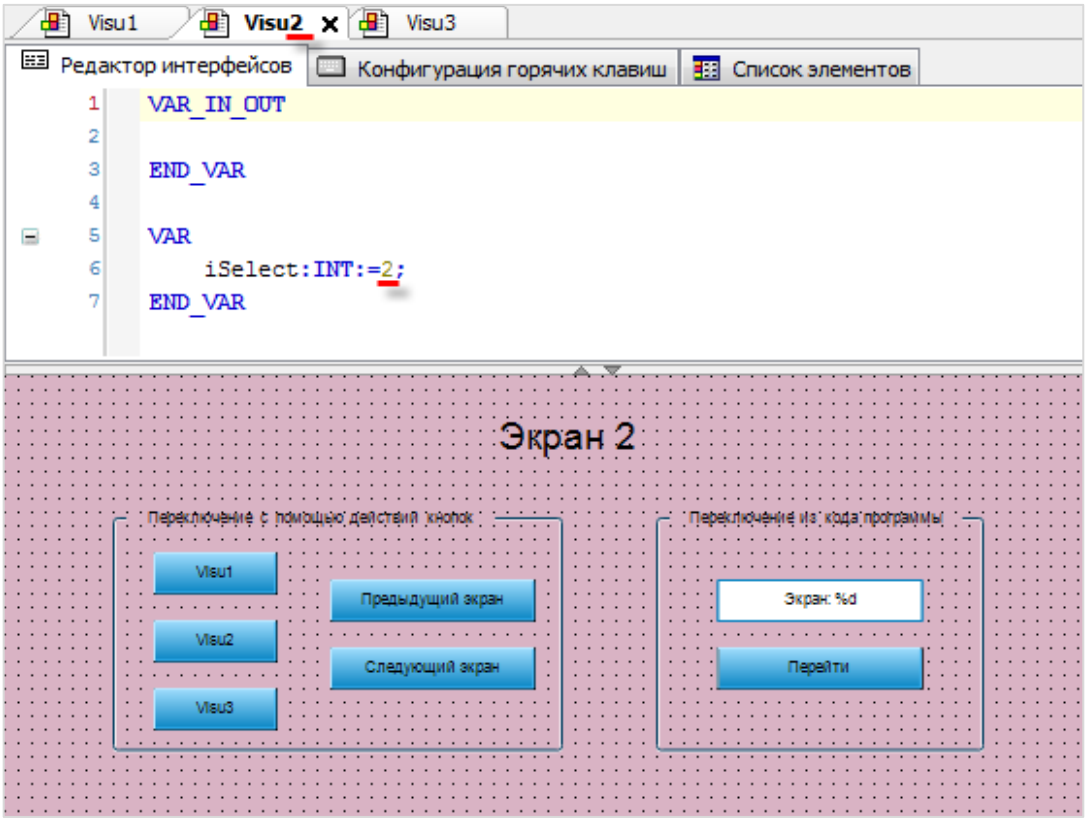


Рисунок 10.280 – Содержимое экрана визуализации Visu2

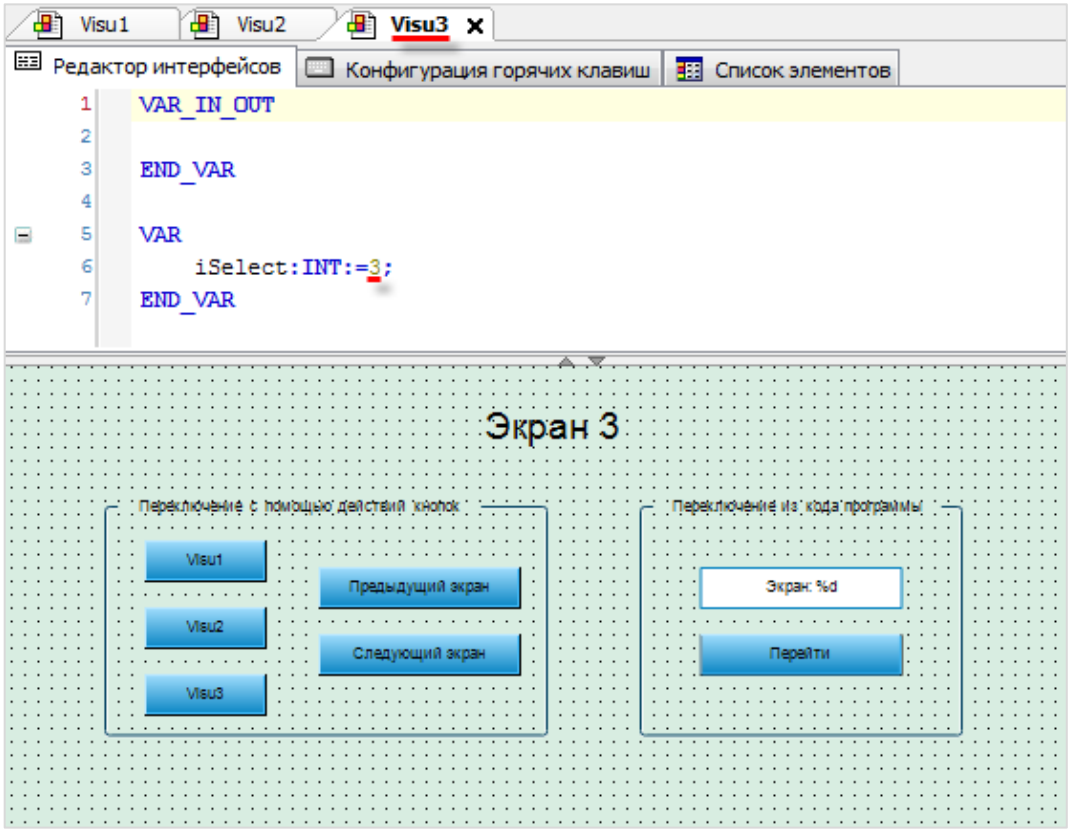
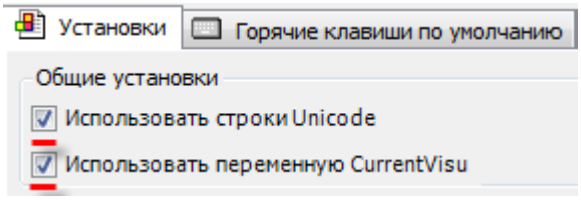


Рисунок 10.281 – Содержимое экрана визуализации Visu3

8. Настроить Менеджер визуализации. В его установках поставить галочки **Использовать строки Unicode** и **Использовать переменную CurrentVisu**:



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

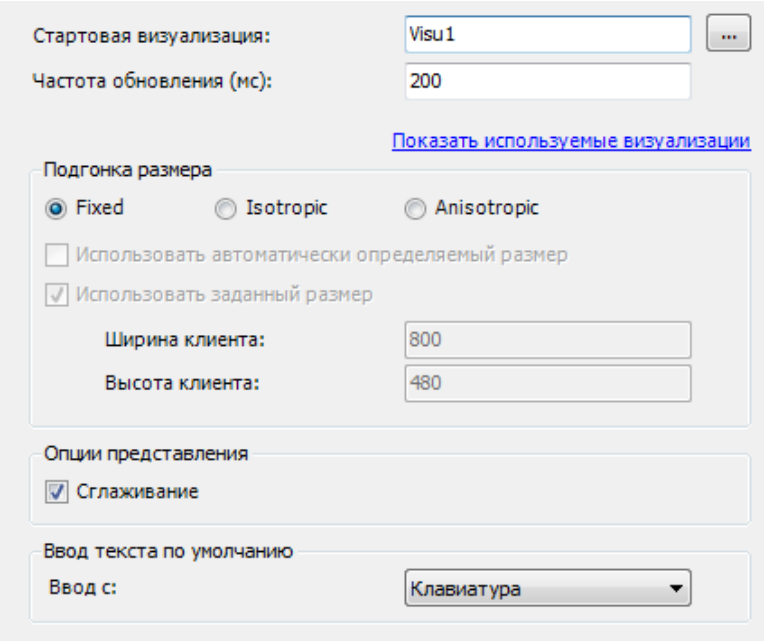


Рисунок 10.282 – Настройки таргет-визуализации

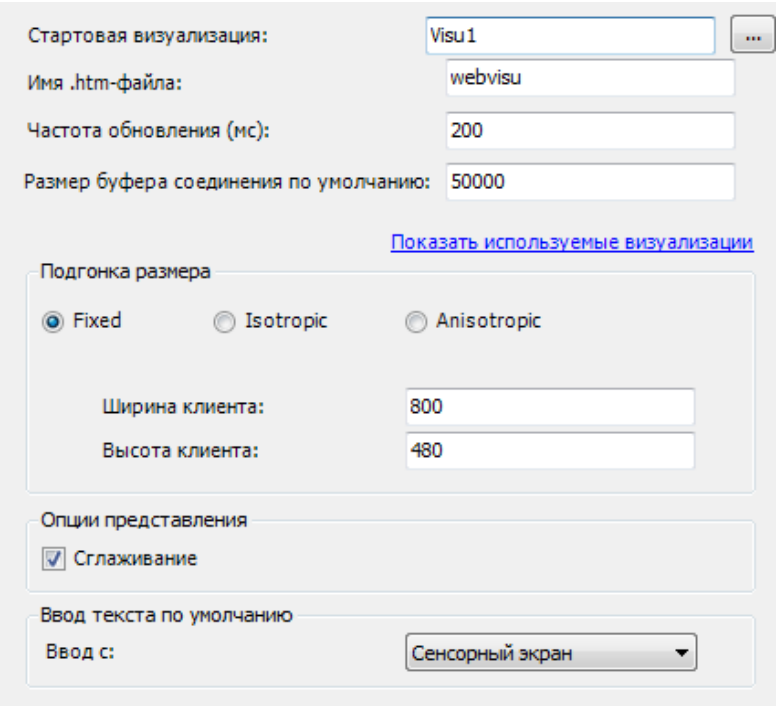


Рисунок 10.283 – Настройки web-визуализации

9. Запустить проект на виртуальном контроллере. Открыть в веб-браузере страницу http://localhost:8080/webvisu.htm, чтобы увидеть web-визуализацию проекта. ***Следует обратить внимание***, что браузер должен поддерживать **HTML5**. Проверить функционал проекта.

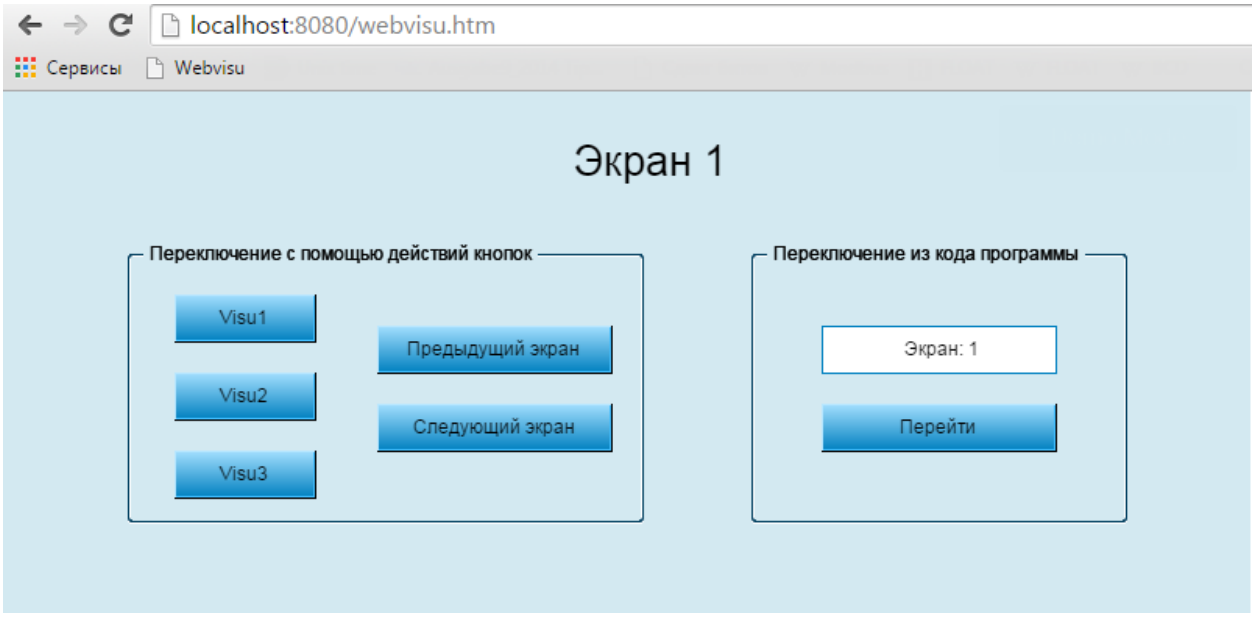


Рисунок 10.284 – Web-визуализация проекта в браузере Chrome

Во время запуска проекта пользователь находится на экране 1.

Во вкладке **Переключение** с помощью действий кнопок следует нажать кнопку **Visu2**, чтобы перейти на экран 2. Затем нажать кнопку **Visu3**, чтобы перейти на экран 3. Нажатие на кнопку **Предыдущий экран** приведет к переходу на экран 2. Нажатие на кнопку **Следующий экран** приведет к возвращению на экран 3.

Во вкладке **Переключение из кода программы** следует нажать на текстовое поле **Экран** и ввести 2. Для переключения раскладки экранной клавиатуры следует воспользоваться клавишей **Shift.**

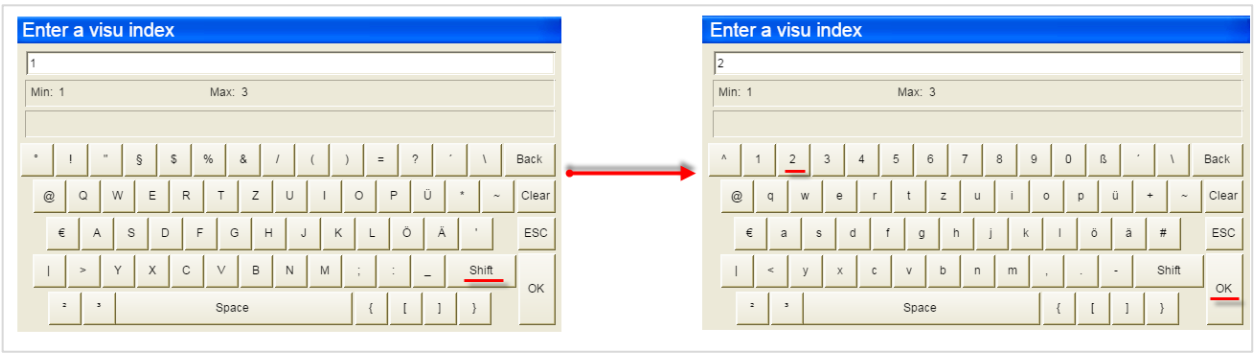


Рисунок 10.285 – Переключение раскладки экранной клавиатуры

Следует нажать кнопку **Перейти**, чтобы перейти на экран с выбранным номером.

ПРИМЕЧАНИЕ

Переключение экранов в этом случае происходит в коде программы с помощью системной переменной **CurrentVisu**.

10.3.2 Использование диалогов

Данный пример посвящен использованию диалоговых окон в пользовательском проекте.

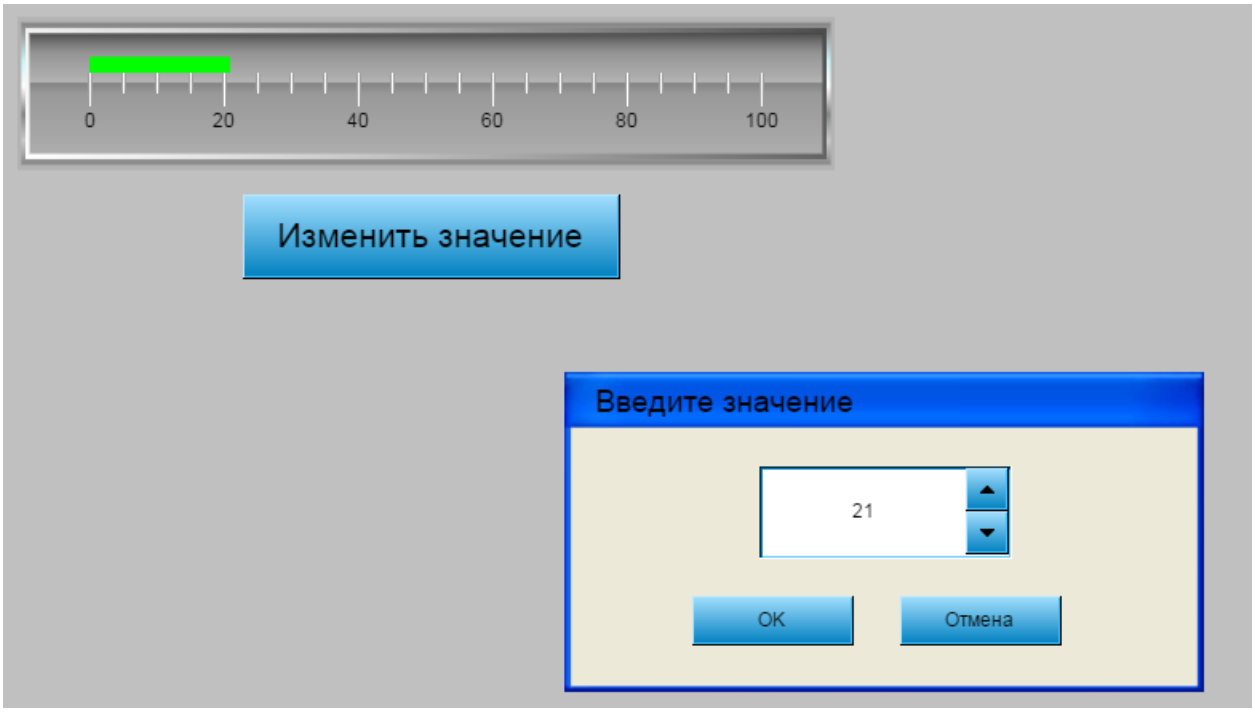


Рисунок 10.286 – Внешний вид примера Использование диалогов

Пример создан в среде **CODESYS V3.5 SP11 Patch 5** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с таргет-файлом **3.5.11.5**.

Пример доступен для скачивания: Example_VisuDialogs.projectarchive

Продвинутый пример от компании **CODESYS Group** по работе с диалогами в коде программы:

VisuDialogST.package

Файлы формата **.package** устанавливаются с помощью **Менеджера пакетов** (вкладка **Инструменты**) или (начиная с версии **CODESYS V3.5 SP17**) с помощью утилиты **CODESYS Installer**.

ПРИМЕЧАНИЕ
Работа с диалогами из кода программы возможна с помощью библиотеки **Visu Utils.**
См. п. 10.4.4.

Для создания примера с использованием диалоговых окон следует:

1. Создать новый ***стандартный*** проект **CODESYS** с названием **Example_VisuDialogs** и настройками по умолчанию: таргет – CODESYS Control Win V3, язык PLC_PRG – **ST**.
2. В программе **PLC_PRG** объявить 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
```

Рисунок 10.287 – Объявление переменных программы PLC_PRG

3. Добавить в проект экран визуализации **ChangeValue**, который будет использоваться в качестве диалога. В его свойствах выбрать тип **Диалог** и размер **400 × 200**:

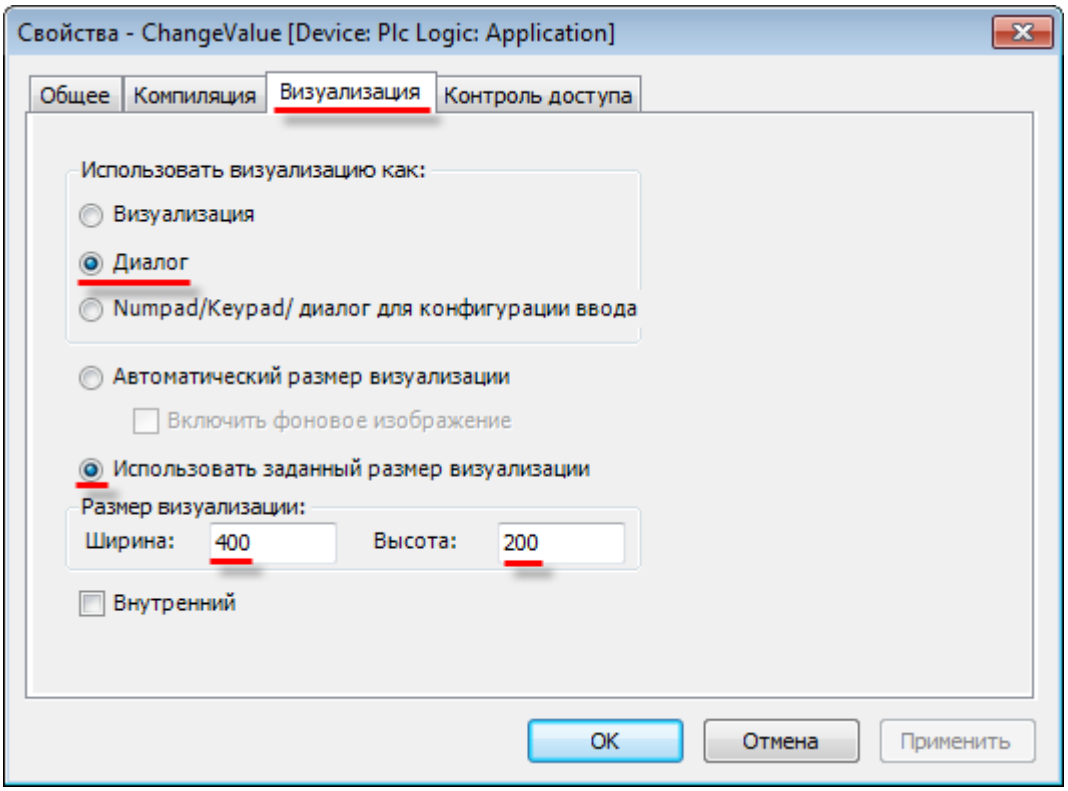


Рисунок 10.288 – Свойства экрана ChangeValue

4. Диалоговое окно **ChangeValue** содержит элемент Изображение, элемент Метка, элемент Управление вращением и два элемента Кнопка, а также выходную переменную интерфейса **iDialogVar** типа **INT**:

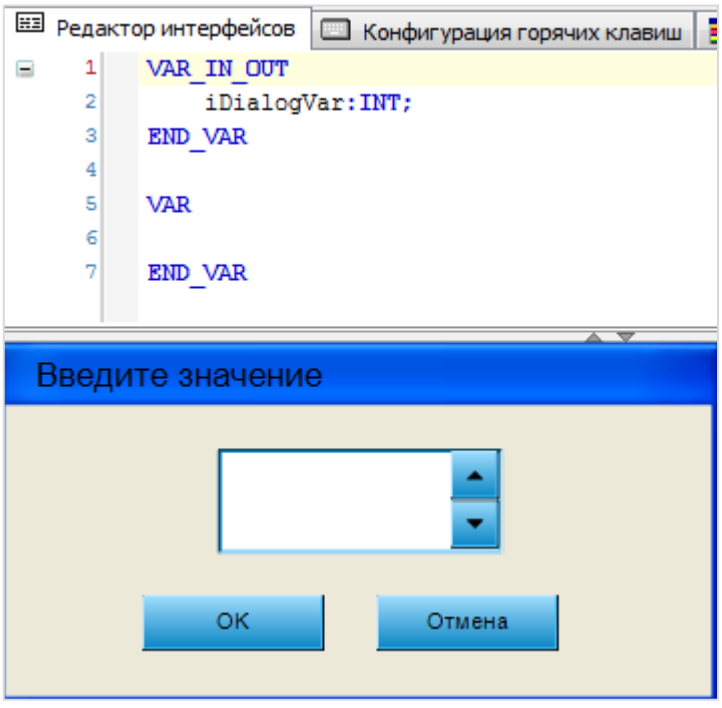


Рисунок 10.289 – Интерфейс и содержимое экрана ChangeValue

Настройки элементов приведены ниже.

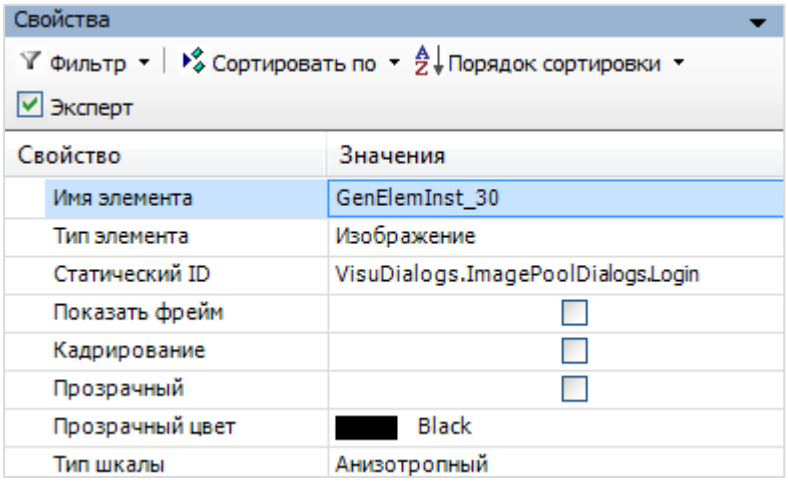


Рисунок 10.290 – Параметры элемента Изображение

Элемент Изображение расположен в нижнем слое (см. п. 5.2, пп. 2) по отношению к остальным элементам, чтобы не перекрывать их.

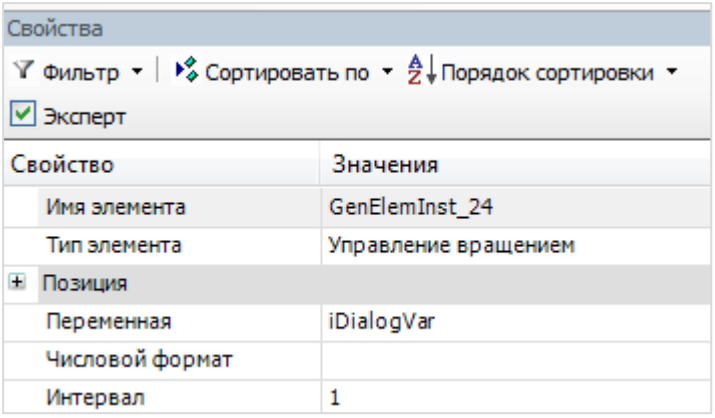


Рисунок 10.291 – Параметры элемента Управление вращением

В свойствах кнопки **OK** во вкладке InputConfiguration к параметру **OnMouseClick** следует привязать действие Закрыть диалог:

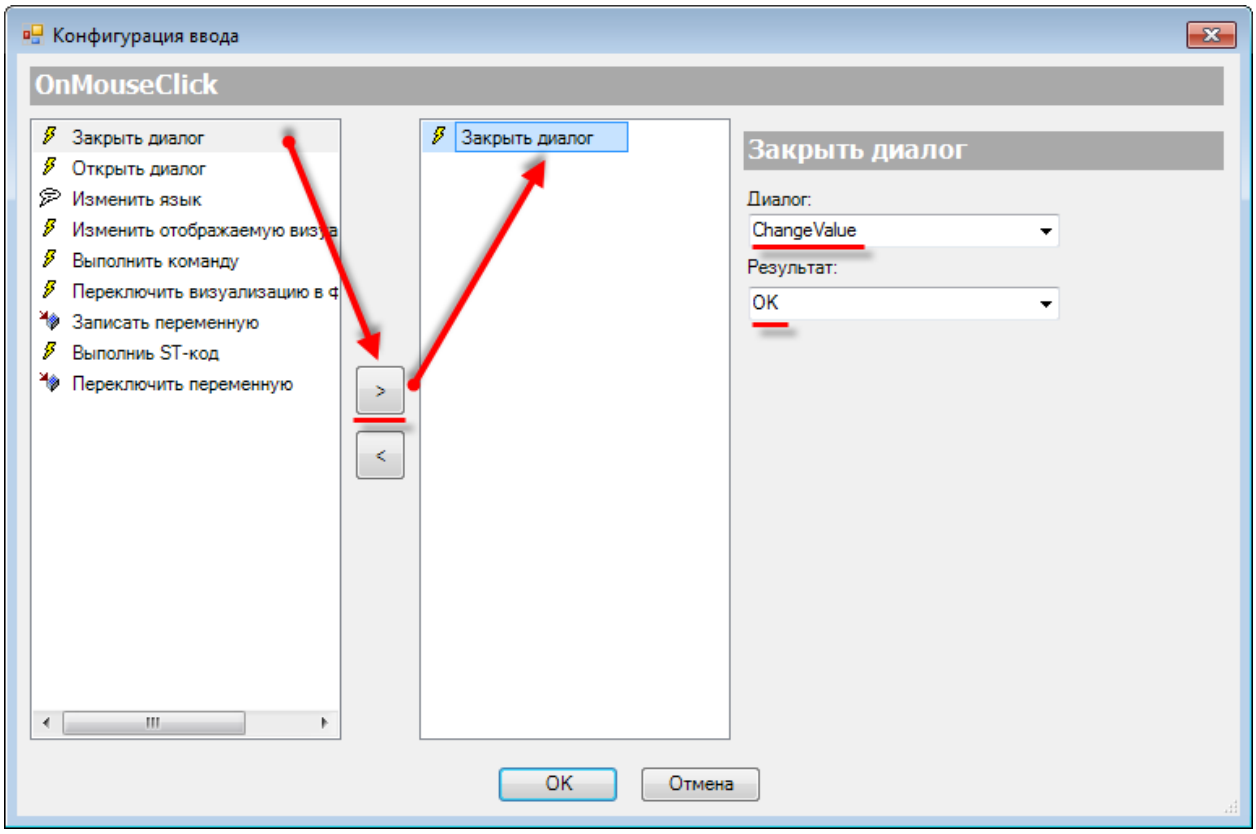


Рисунок 10.292 – Настройки действия кнопки OK

Настройки действия кнопки **Отмена** будут отличаться только результатом (результат: **Отмена**).

5. Добавить в проект экран визуализации **Visualization**. В его свойствах выбрать размер **800 × 480**. Данный экран будет являться основным экраном проекта, и именно на нем будет располагаться кнопка вызова диалога.

   Экран содержит элемент Отображение линейки и элемент Кнопка:

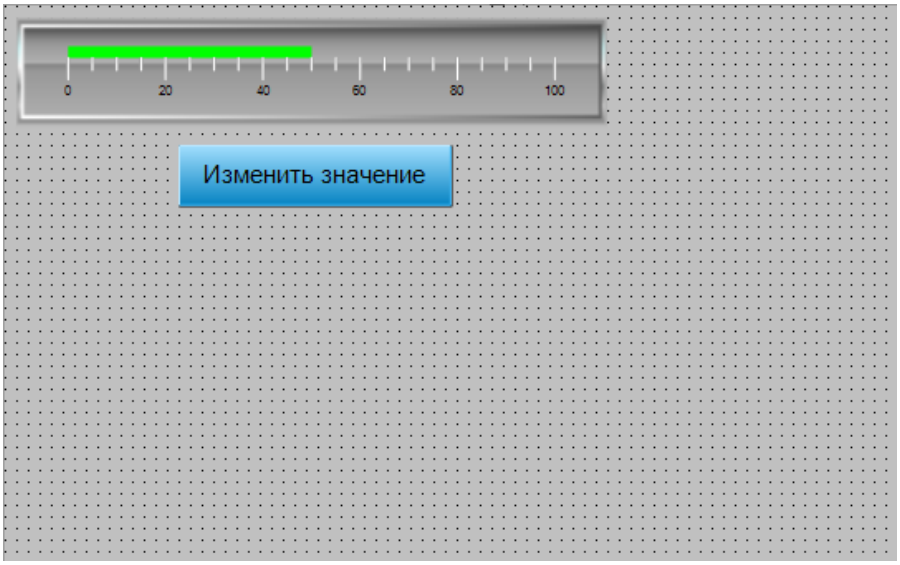


Рисунок 10.293 – Содержимое экрана Visualization

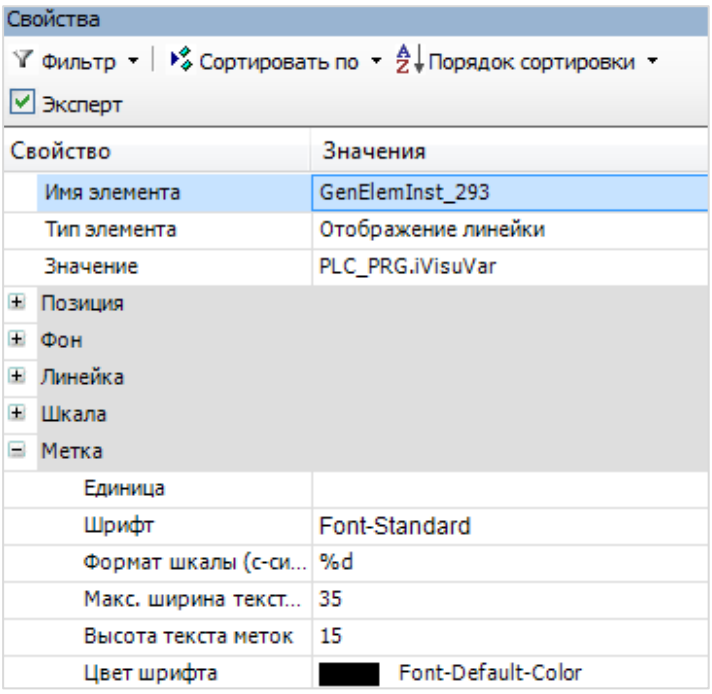


Рисунок 10.294 – Параметры элемента Отображение линейки

Затем следует настроить действие кнопки **Изменить значение**.

В свойствах кнопки **Изменить значение** во вкладке InputConfiguration к параметру **OnMouseClick** следует привязать действие Открыть диалог:

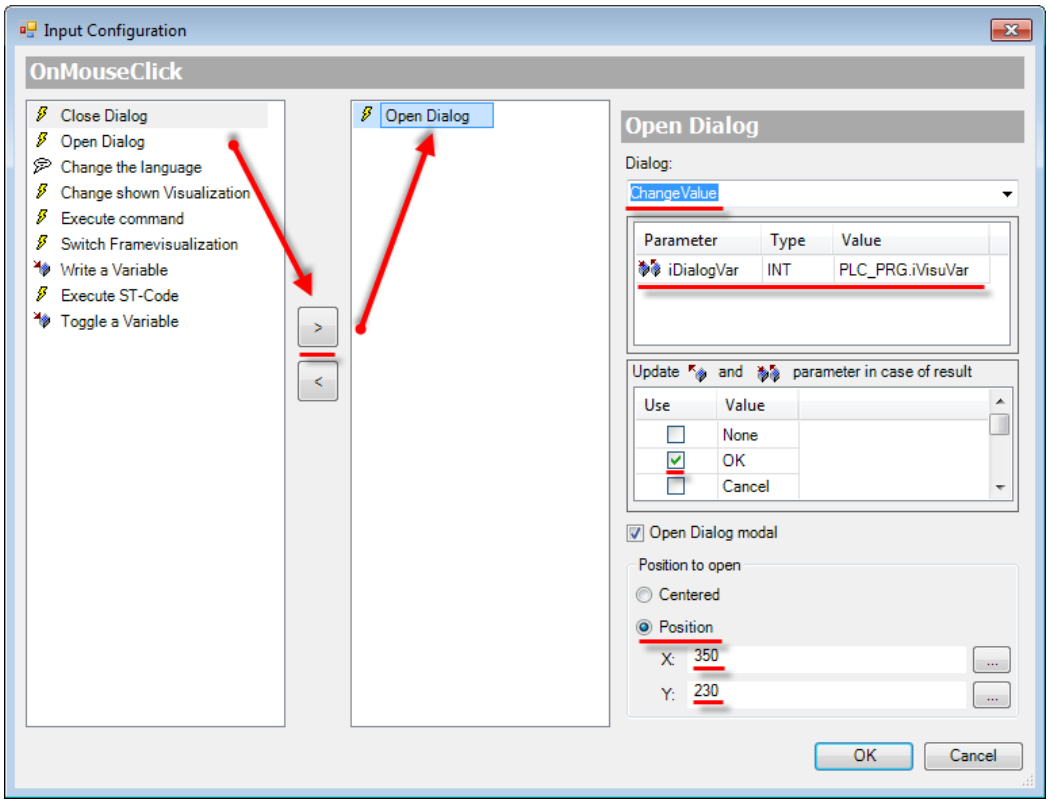
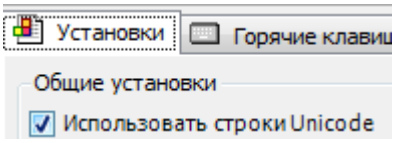


Рисунок 10.295 – Настройки действия кнопки Изменить значение

После 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**.

6. Настроить Менеджер визуализации. В его установках следует поставить галочку **Использовать строки Unicode**:



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

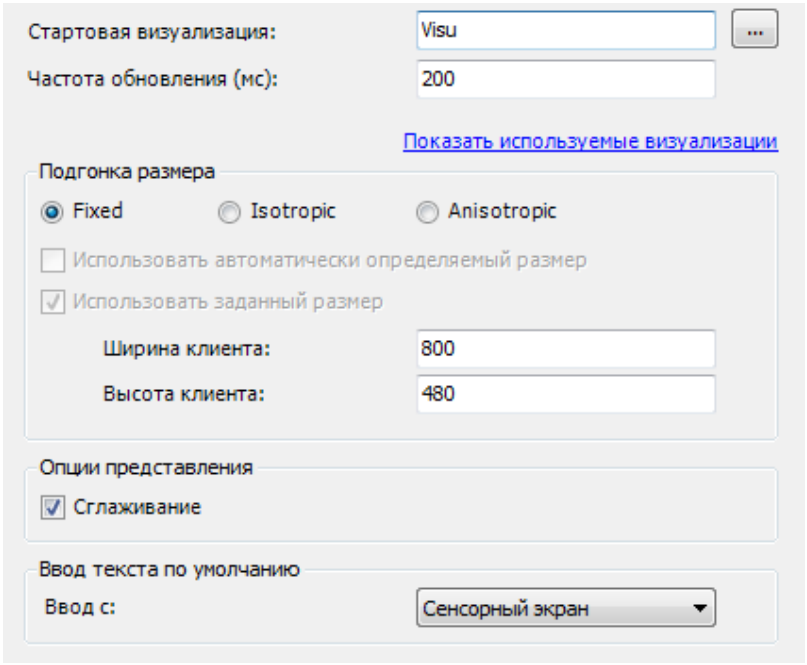


Рисунок 10.296 – Настройки таргет-визуализации

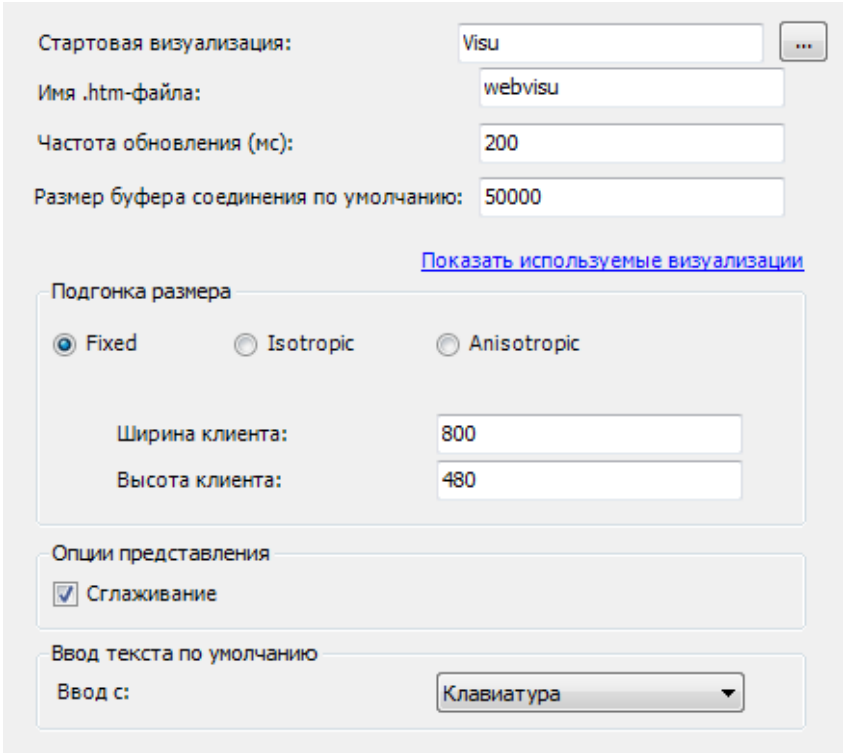


Рисунок 10.297 – Настройки web-визуализации

7. Запустить проект на виртуальном контроллере. Открыть в веб-браузере страницу http://localhost:8080/webvisu.htm, чтобы увидеть web-визуализацию проекта. ***Следует обратить внимание***, что браузер должен поддерживать **HTML5**. Проверить функционал проекта.

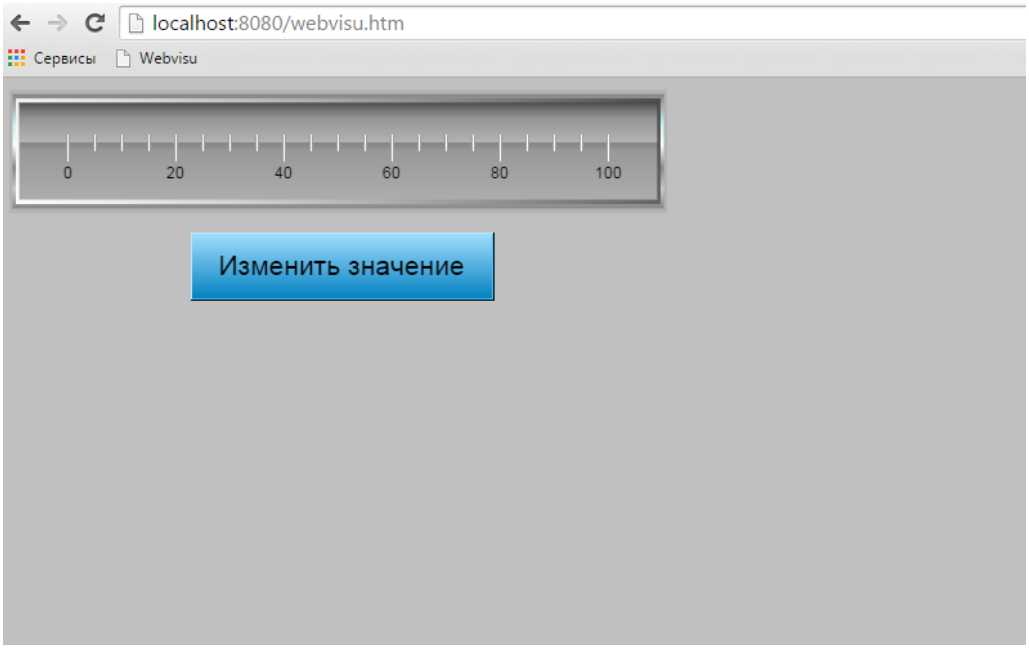


Рисунок 10.298 – Web-визуализация проекта в браузере Chrome

Затем следует нажать кнопку Изменить значение для открытия диалога **ChangeValue**:

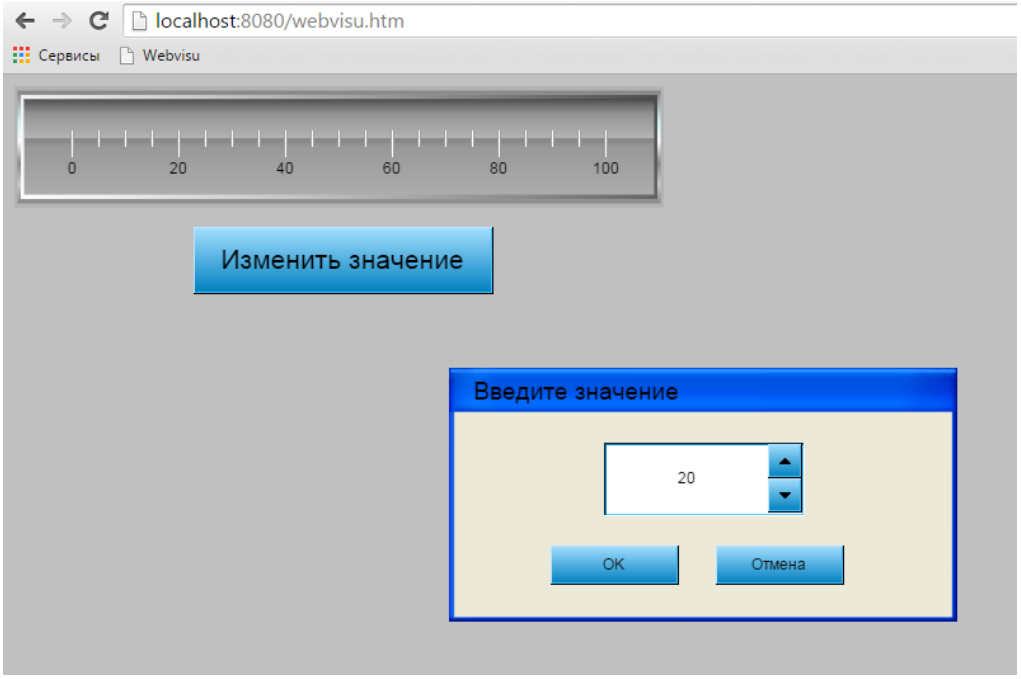


Рисунок 10.299 – Изменение значения с помощью диалога

Затем следует ввести число и нажать **ОК**. Введение значение отобразится элементом **Отображение линейки**. Нажатие на кнопку **Отмена** закроет диалог без записи значения в программу.

10.3.3 Использование интерфейса фрейма

Данный пример посвящен работе с интерфейсом элемента Фрейм.

Довольно часто в системах автоматизации присутствуют группы типовых устройств, для которых следует 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:

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.

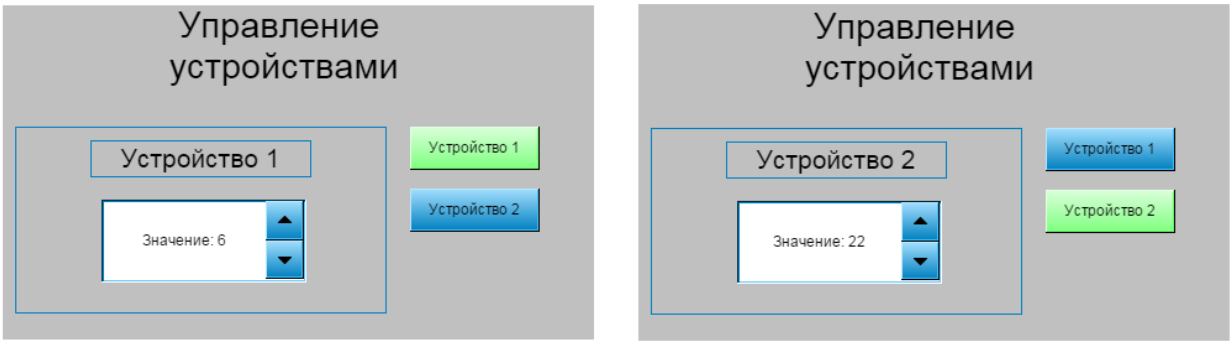


Рисунок 10.300 – Внешний вид примера Интерфейс фрейма
На обоих рисунках в фрейме открыты разные экземпляры одного экрана визуализации

Пример создан в среде **CODESYS V3.5 SP11 Patch 5** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с таргет-файлом **3.5.11.5**.

Пример доступен для скачивания: Example_FrameInterface.projectarchive

Для создания примера с использованием элемента **Фрейм** следует:

1. Создать новый ***стандартный*** проект **CODESYS** с названием **Example_FrameInterface** и настройками по умолчанию: таргет – CODESYS Control Win V3, язык PLC_PRG – **ST**.
2. В программе **PLC_PRG** объявить следующие переменные:

```
PROGRAM PLC_PRG
VAR
    // Переменные устройства 1
    bButtonDevice1:BOOL:=TRUE;              // переменная для изменения цвета кнопки при выборе устройства
    wsNameDevice1:WSTRING:="Устройство 1";  // название устройства
    iValueDevice1:INT;                      // параметр устройства

    // Переменные устройства 2
    bButtonDevice2:BOOL;
    wsNameDevice2:WSTRING:="Устройство 2";
    iValueDevice2:INT;
END_VAR
```

Рисунок 10.301 – Объявление переменных программы PLC_PRG

3. Добавить в проект экран визуализации **Frame**. В его свойствах выбрать размер **300 × 150**. Экран будет содержать элемент Текстовое поле, элемент Управление вращением и две входных переменных интерфейса:

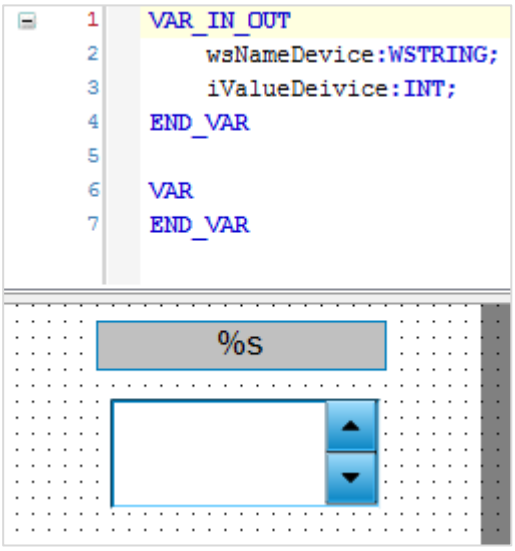


Рисунок 10.302 – Интерфейс и содержимое экрана Frame

Свойства

Фильтр | Сортировать по | Порядок сортировки | 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	

Рисунок 10.303 – Параметры элемента Текстовое поле

Свойства

Фильтр | Сортировать по | Порядок сортировки | 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

Рисунок 10.304 – Параметры элемента Управление вращением

4. Добавить в проект экран визуализации **Visualization**. В его свойствах выбрать размер **480 × 272**. Экран будет содержать элемент Фрейм и два элемента Кнопка. Название экрана создано с помощью элемента Метка.

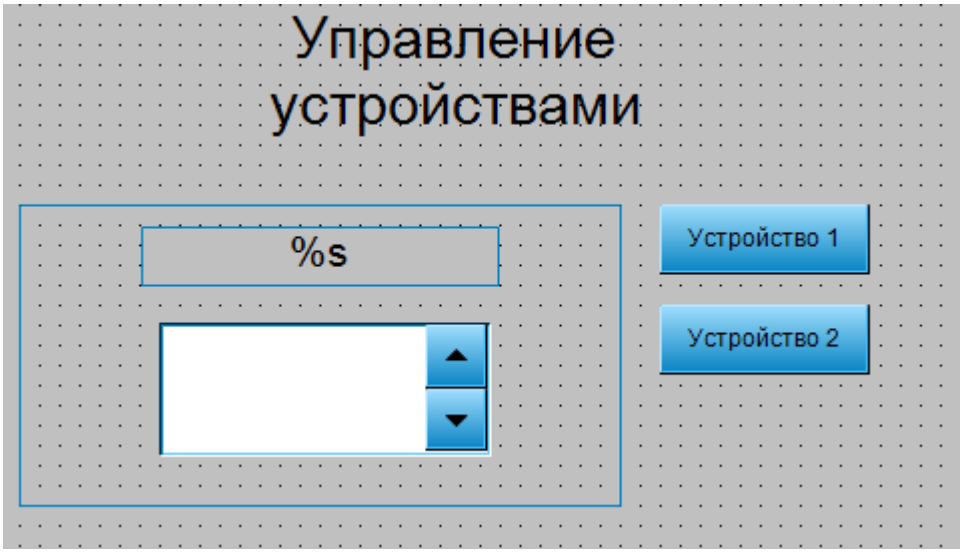


Рисунок 10.305 – Содержимое экрана Visualization

Сначала следует привязать к элементу Фрейм два экземпляра экрана **Frame**, и к интерфейсу каждого из них привязать соответствующие переменные программы (вкладка **Ссылки**):

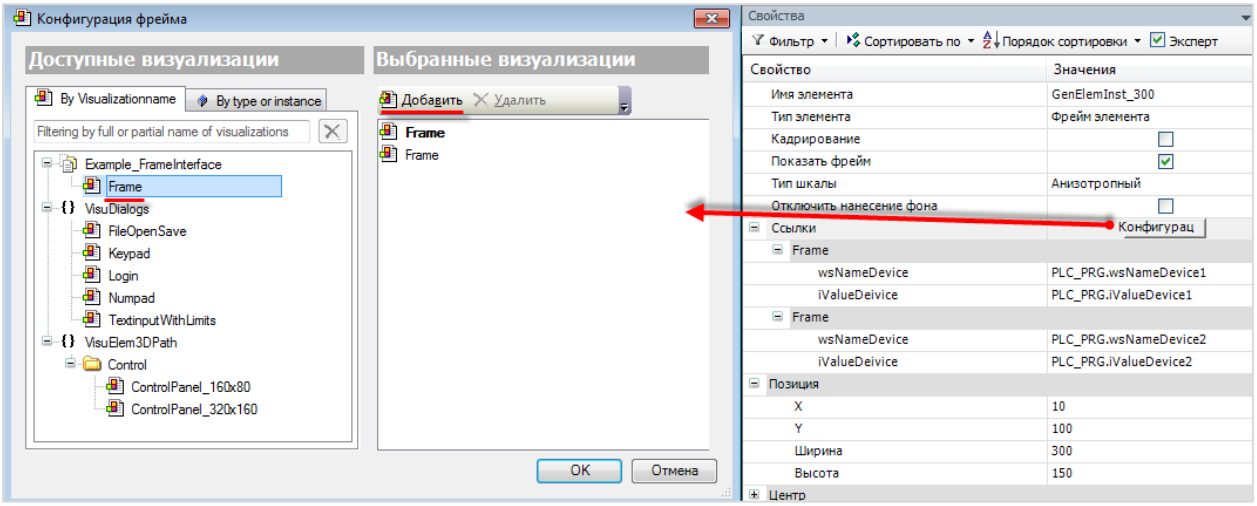


Рисунок 10.306 – Параметры и конфигурация элемента Фрейм

Настроить кнопку **Устройство 1**.

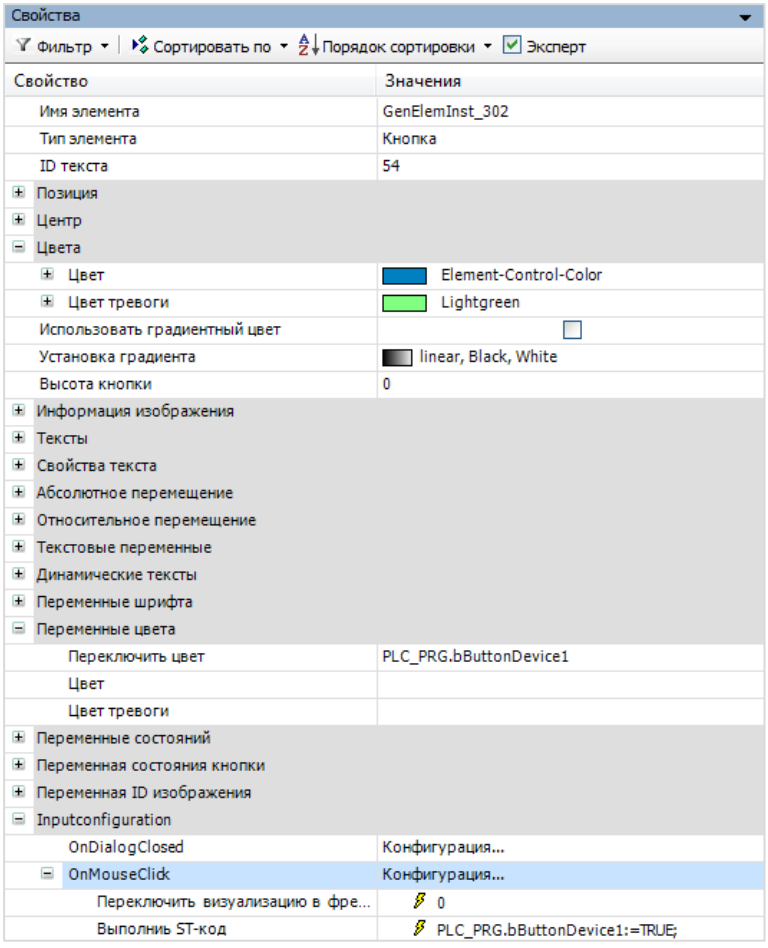


Рисунок 10.307 – Параметры кнопки Устройство 1

Во вкладке InputConfiguration к параметру **OnMouseClick** следует привязать действия Переключить визуализацию во фрейме и Выполнить ST-код (для изменения цвета кнопки при выборе экрана устройства):

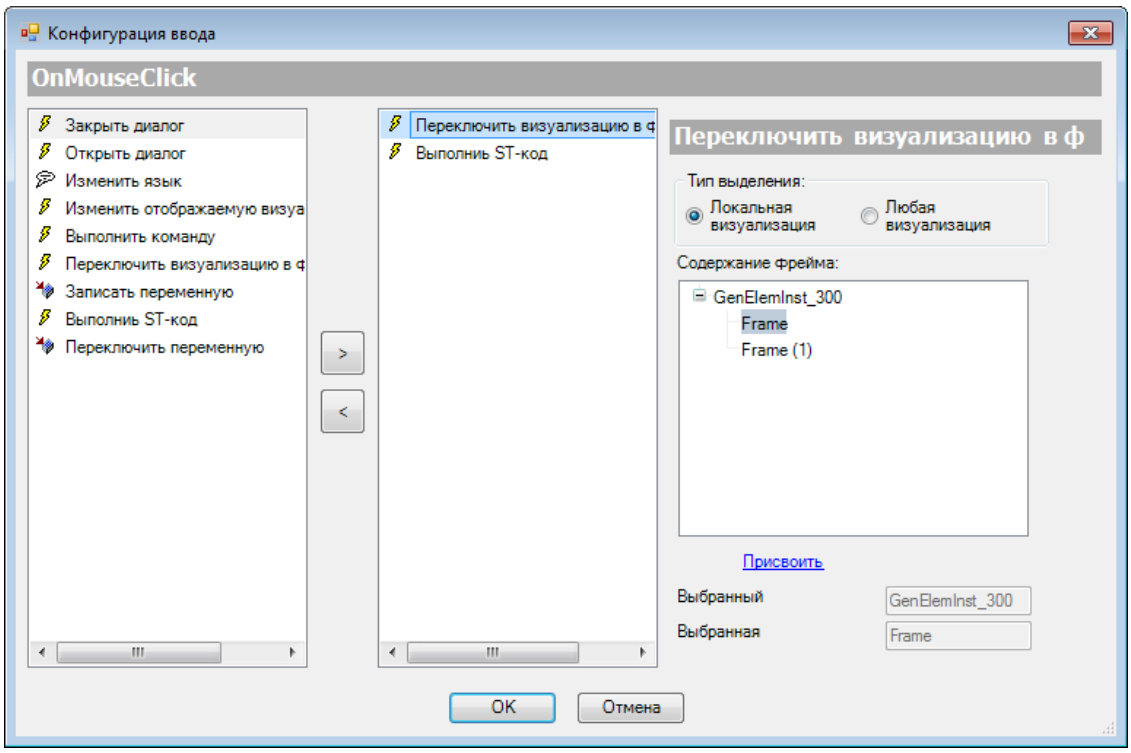


Рисунок 10.308– Настройки действия кнопки Устройство 1 (Переключить визуализацию в фрейме)

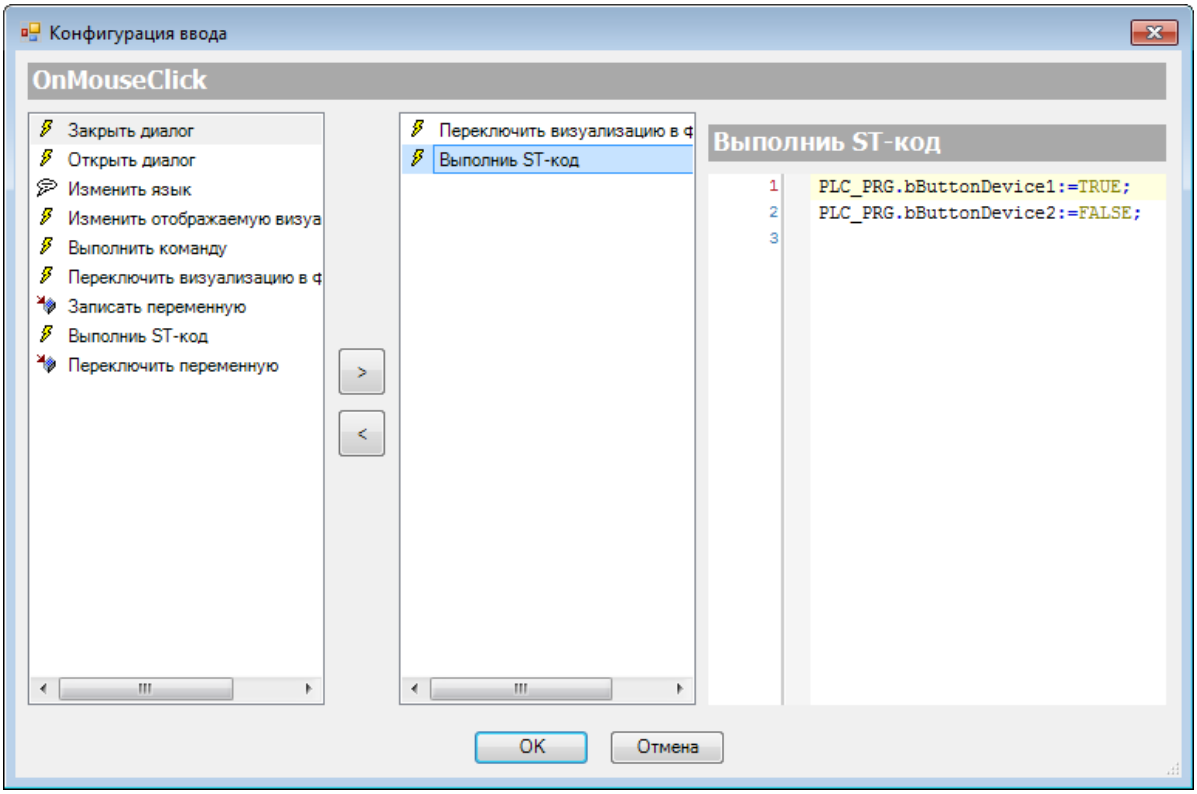


Рисунок 10.309 – Настройки действия кнопки Устройство 1 (Выполнить ST-код)

Затем следует настроить кнопку **Устройство 2**.

Свойства

Фильтр | Сортировать по | Порядок сортировки | Эксперт

Свойство	Значения
Имя элемента	GenElemInst_304
Тип элемента	Кнопка
ID текста	55
⊞ Позиция	
⊞ Центр	
⊟ Цвета	
⊞ Цвет	Element-Control-Color
⊞ Цвет тревоги	Lightgreen
Использовать градиентный цвет	☐
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...
⊟ OnMouseClick	Конфигурация...
Переключить визуализацию в фре...	1
Выполнить ST-код	PLC_PRG.bButtonDevice1:=FALSE;

Рисунок 10.310 – Параметры элемента Устройство 2

Во вкладке InputConfiguration к параметру **OnMouseClick** следует привязать действия Переключить визуализацию во фрейме и Выполнить ST-код (для изменения цвета кнопки во время выбора экрана устройства):

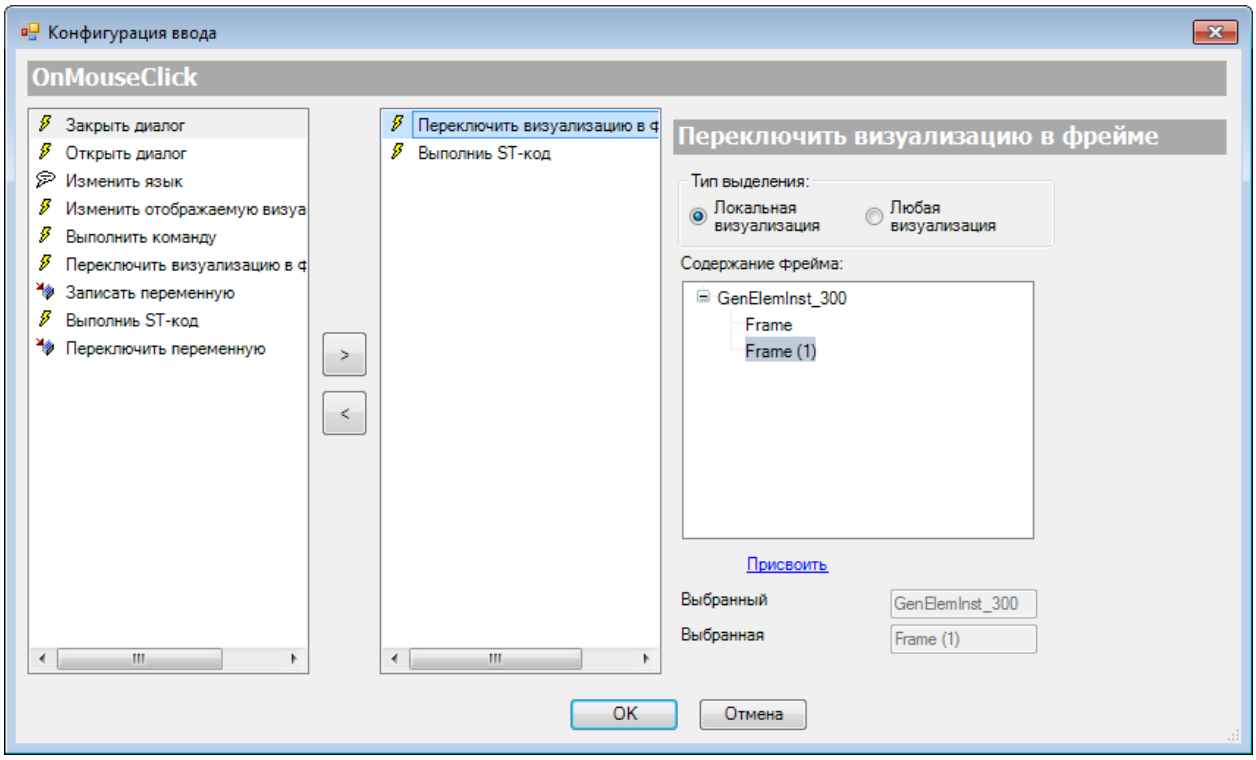


Рисунок 10.311 – Настройки действия кнопки Устройство 2 (Переключить визуализацию в фрейме)

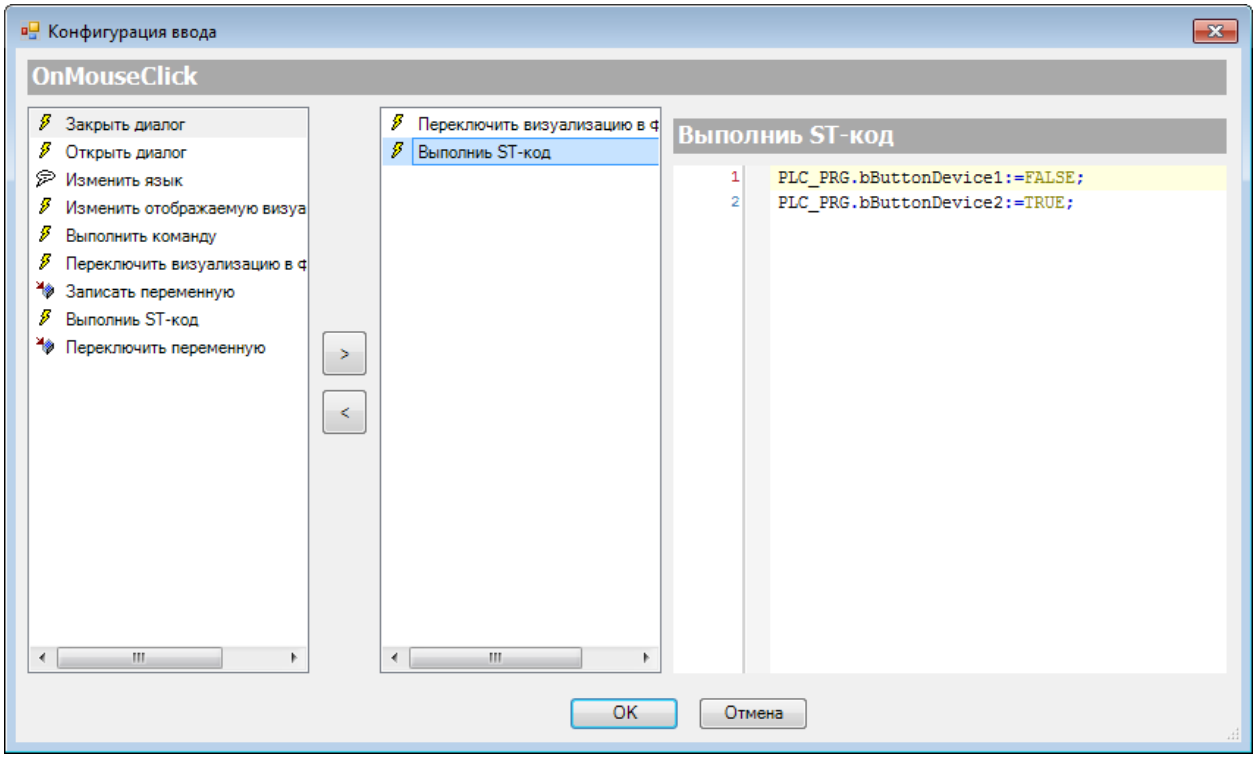
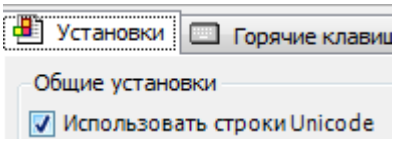


Рисунок 10.312 – Настройки действия кнопки Устройство 2 (Выполнить ST-код)

5. Настроить Менеджер визуализации. В его установках следует поставить галочку **Использовать строки Unicode**:



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

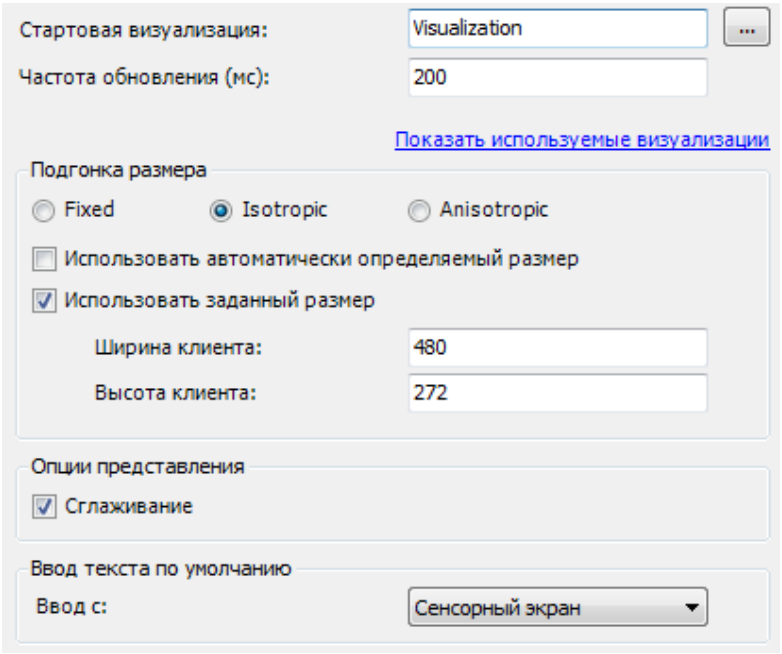


Рисунок 10.313 – Настройки таргет-визуализации

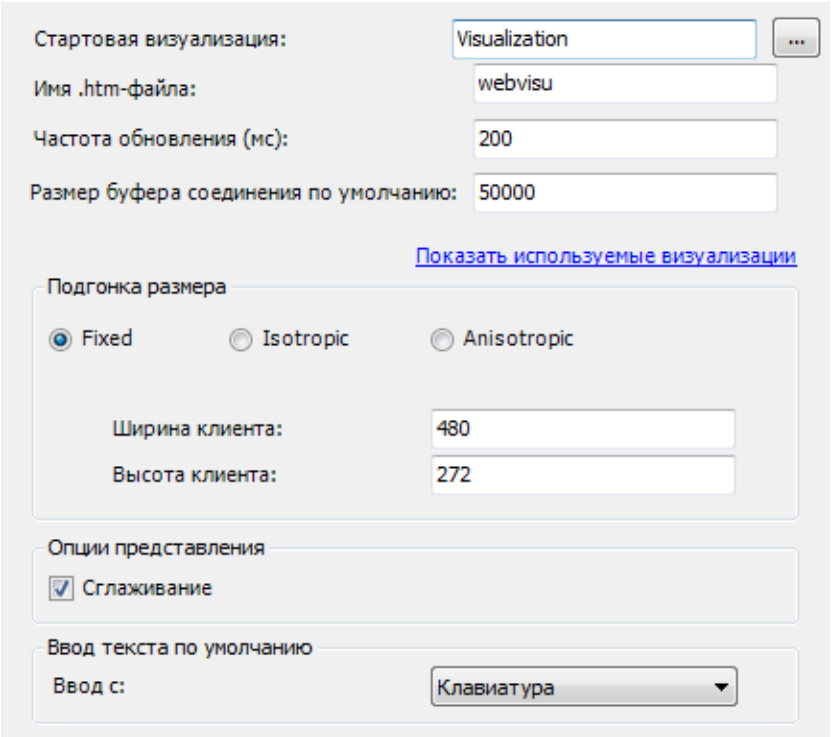


Рисунок 10.314 – Настройки web-визуализации

6. Запустить проект на виртуальном контроллере. Открыть в веб-браузере страницу http://localhost:8080/webvisu.htm, чтобы увидеть web-визуализацию проекта. ***Следует обратить внимание***, что браузер должен поддерживать **HTML5**. Проверить функционал проекта.

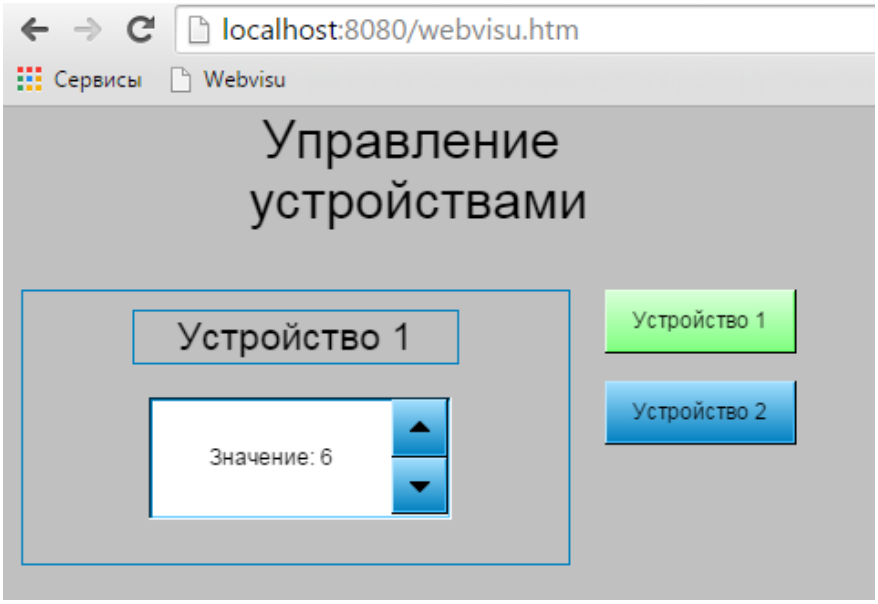


Рисунок 10.315 – Web-визуализация проекта в браузере Chrome

По умолчанию во фрейме отображается экран устройства **1**. Следует ввести значение (например, 6). Чтобы открыть в фрейме другой экземпляр экрана визуализации следует нажать кнопку **Устройство 2**.

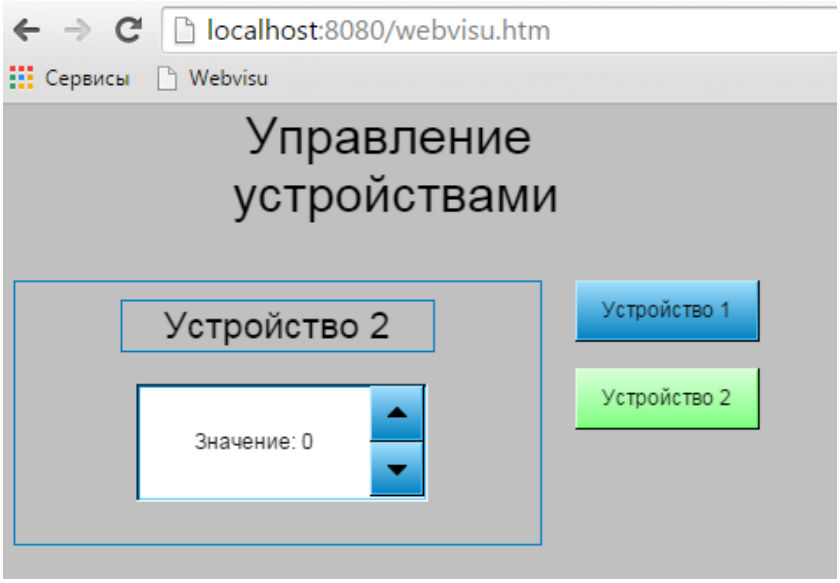


Рисунок 10.316 – Переключение экземпляра экрана во фрейме

Чтобы увидеть, что ранее введенное значение сохранилось следует переключиться на **Устройство 1**.

10.3.4 Создание анимации (элемент Изображение)

Данный пример посвящен созданию анимации с помощью элемента Изображение.

Рисунок 10.317 – Внешний вид примера Анимация

Пример создан в среде **CODESYS V3.5 SP11 Patch 5** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с таргет-файлом **3.5.11.5**.

Пример доступен для скачивания: Example_Animation.projectarchive

Для создания примера с использованием анимации следует:

1. Создать новый ***стандартный*** проект **CODESYS** с названием **Example_Animation** и настройками по умолчанию: таргет – CODESYS Control Win V3, язык PLC_PRG – **ST**.
2. Добавить в проект компонент Пул изображений с названием по умолчанию **ImagePool**:

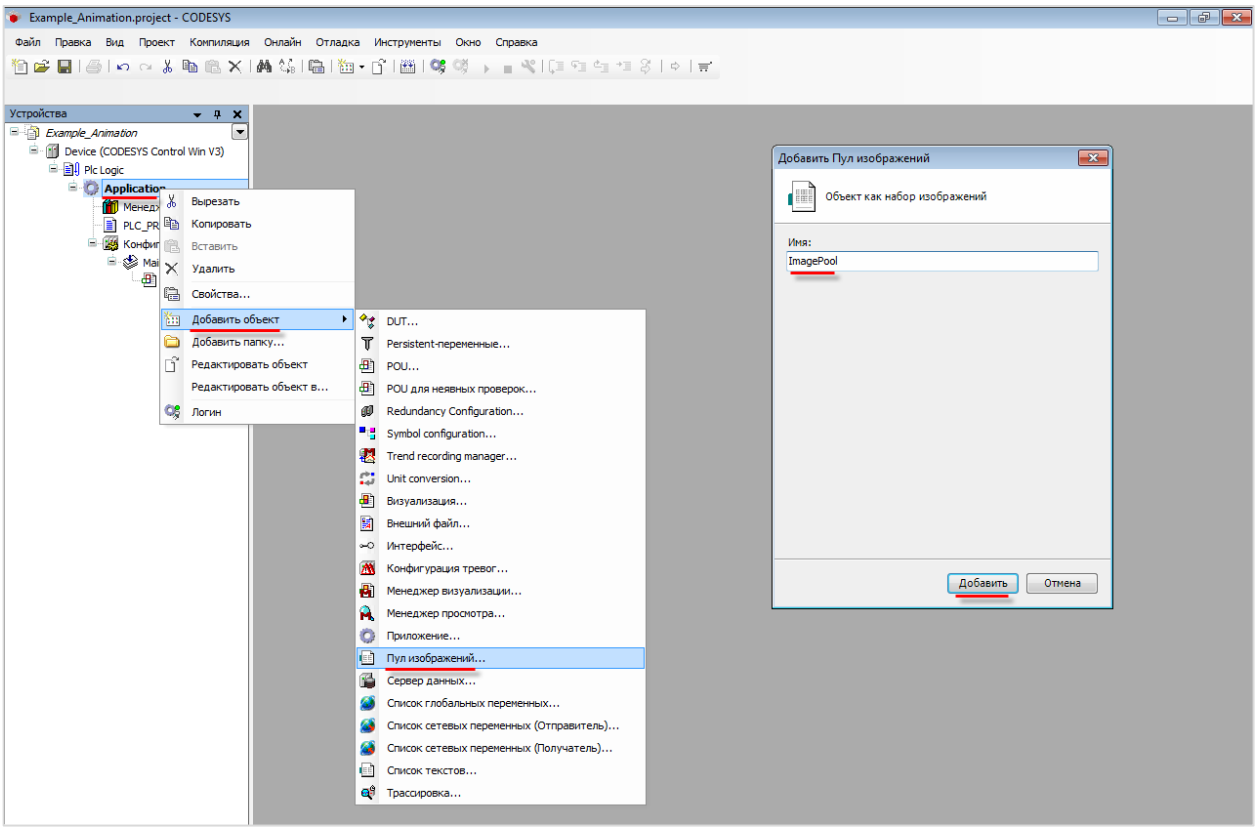


Рисунок 10.318 – Добавление в проект пула изображений

3. Добавить в пул изображений графический файлы, которые будут использоваться в качестве кадров анимации:

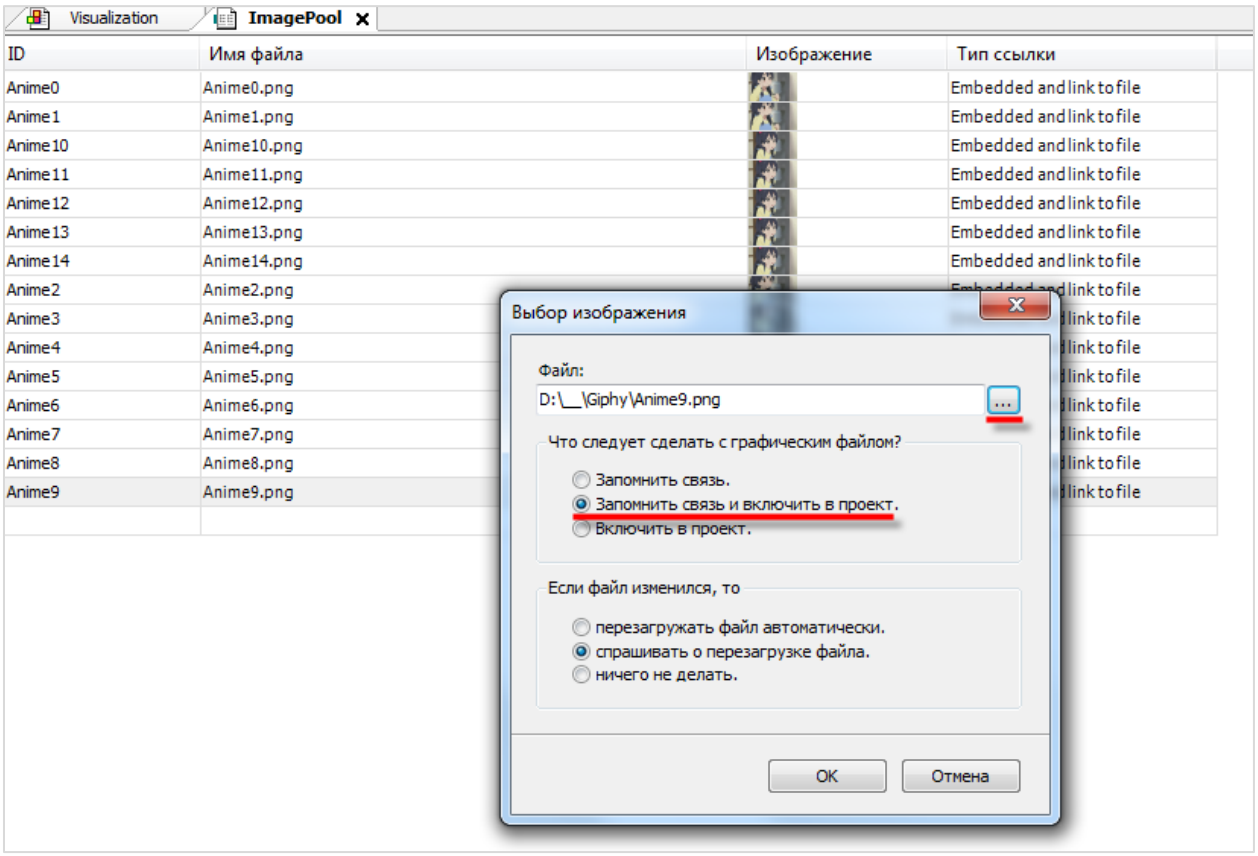


Рисунок 10.319 – Добавление графических файл в пул изображений

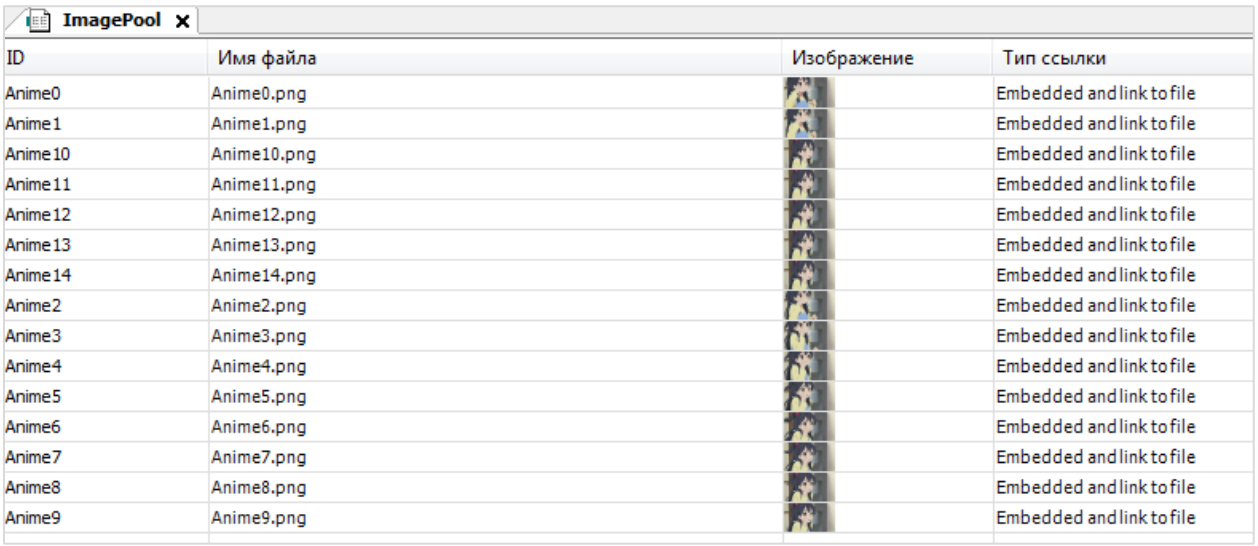


Рисунок 10.320 – Внешний вид пула изображений после добавления графических файлов

4. В программе **PLC_PRG** объявить следующие переменные:

```
PLC_PRG
1  PROGRAM PLC_PRG
2
3  VAR
4      xEnable:              BOOL;                 // сигнал включения анимации (TRUE - включена)
5      xStop:                BOOL;                 // сигнал остановки анимации (TRUE - остановлена)
6      xReset:               BOOL;                 // сигнал сброса анимации на начальный кадр
7      sPictureMainName:     STRING :=   'Anime';  // общая часть названия картинок из Пула изображений
8      uiCountPictures:      UINT   :=   15;       // число картинок в пуле изображений
9
10     xActive:              BOOL;                 // флаг "анимация запущена"
11     xStopped:             BOOL;                 // флаг "анимация остановлена"
12     uiCurrentFrame:       UINT;                 // номер отображаемой картинки
13     sCurrentFrame:        STRING :=   'Anime0'; // название картинки начального кадра
14
15     i:                    INT;                  // счетчик цикла
16     fbRtrigReset:         R_TRIG;               // ФБ определения сигнала сброса анимации
17 END_VAR
```

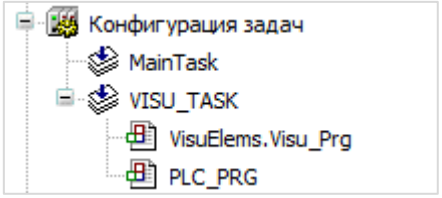
Рисунок 10.321 – Объявление переменных программы PLC_PRG

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

```
1
2   // по сигналу сброса переключаем анимацию на начальный кадр
3   fbRtrigReset(CLK := xReset);
4
5   IF fbRtrigReset.Q THEN
6       uiCurrentFrame := 0;
7       sCurrentFrame  := 'Anime0';
8   END_IF
9
10  // сигнал остановки приоритетнее сигнала запуска
11  IF xStop THEN
12      xActive := FALSE;
13      xStopped   := TRUE;
14  ELSIF xEnable THEN
15      xActive  := TRUE;
16      xStopped := FALSE;
17
18      IF uiCurrentFrame >= uiCountPictures THEN
19          uiCurrentFrame := 0;
20      END_IF
21
22      sCurrentFrame  := CONCAT(sPictureMainName, UINT_TO_STRING(uiCurrentFrame) );
23      uiCurrentFrame := uiCurrentFrame + 1;
24
25  END_IF
```

Рисунок 10.322 – Код программы PLC_PRG

Программу **PLC_PRG** следует привязать к задаче **VISU_TASK**. Период переключения кадров анимации будет соответствовать времени цикла этой задачи.



6. Добавить в проект экран визуализации **Visualization**. В его свойствах выбрать размер **800 × 480**. Экран будет содержать элемент Изображение, два элемента Прямоугольник и три элемента Кнопка.

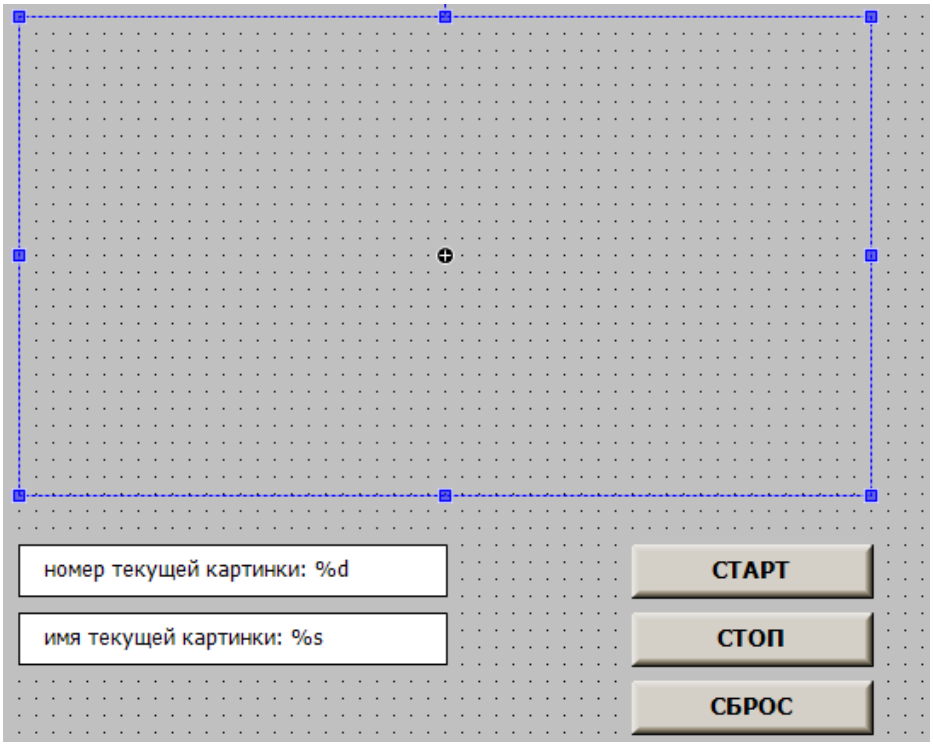


Рисунок 10.323 – Содержимое экрана Visualization

Сначала следует настроить элемент Изображение. К параметру **ID изображения** привязывается переменная **sCurrentFrame**:

Свойства

Фильтр | Сортировать по | Порядок сортировки | Эксперт

Свойство	Значения
Тип масштабирования	Изотропия
Горизонтальное выравнив...	Лево
Вертикальное выравнивание	Верх
⊟ Позиция	
X	140
Y	10
Ширина	500
Высота	281
Угол	0
⊟ Центр	
X	390
Y	150
⊞ Цвета	
⊞ Вид элемента	
⊟ Тексты	
Текст	
Подсказка	
⊞ Свойства текста	
⊟ Переменная ID изображения	
ID изображения	PLC_PRG.sCurrentFrame

Рисунок 10.324 – Параметры элемента Изображение

К кнопке **Старт** следует привязать переменную **xEnable**, к кнопке **Стоп** – **xStop**, к кнопке **Сброс** – **xReset**:

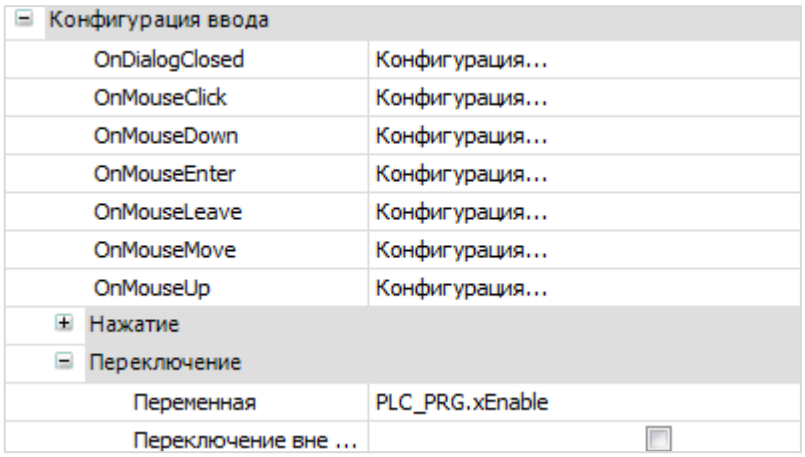


Рисунок 10.325 – Параметры кнопки Старт

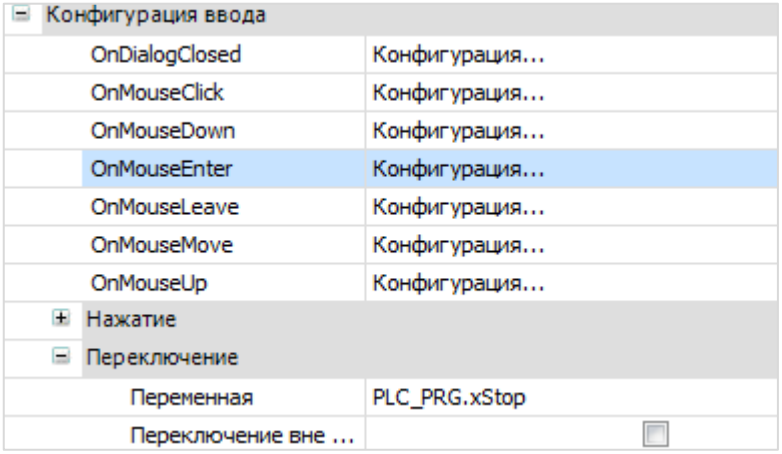


Рисунок 10.326 – Параметры переключателя Стоп

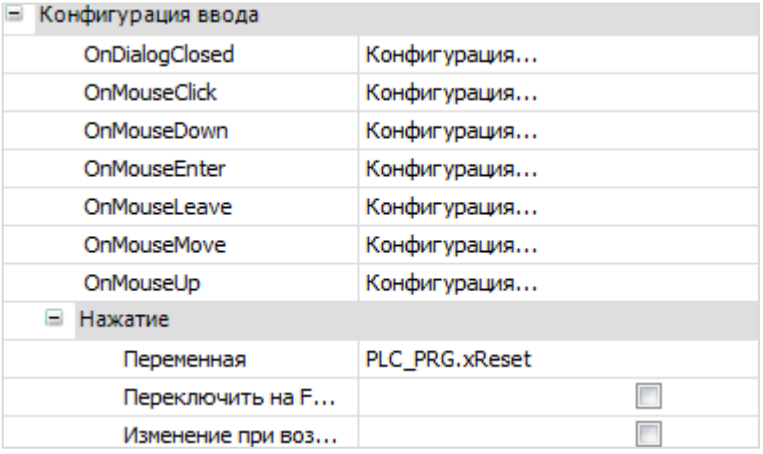


Рисунок 10.327 – Параметры переключателя Сброс

К прямоугольникам следует привязать переменные **uiCurrentFrame** и **sCurrentFrame:**

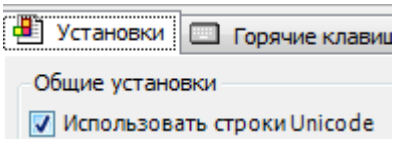
Свойство	Значения
Имя элемента	GenElemInst_9
ID текста	160
Тип элемента	Прямоугольник
⊟ Позиция	
X	140
Y	320
Ширина	251
Высота	30
Угол	0
⊟ Центр	
X	265
Y	335
⊞ Цвета	
Использовать градиентны...	☐
Установка градиента	linear, Black, White
⊞ Вид элемента	
⊟ Тексты	
Текст	номер текущей картинки: %d
Подсказка	
⊞ Свойства текста	
⊞ Абсолютное перемещение	
⊞ Относительное перемещение	
⊟ Текстовые переменные	
Текстовая переменная	PLC_PRG.uiCurrentFrame
Переменная подсказки	

Рисунок 10.328 – Параметры прямоугольника Номер текущей картинки

Свойство	Значения
Имя элемента	GenElemInst_11
ID текста	136
Тип элемента	Прямоугольник
⊟ Позиция	
X	140
Y	360
Ширина	251
Высота	30
Угол	0
⊟ Центр	
X	265
Y	375
⊞ Цвета	
Использовать градиентны...	☐
Установка градиента	linear, Black, White
⊞ Вид элемента	
⊟ Тексты	
Текст	имя текущей картинки: %s
Подсказка	
⊞ Свойства текста	
⊞ Абсолютное перемещение	
⊞ Относительное перемещение	
⊟ Текстовые переменные	
Текстовая переменная	PLC_PRG.sCurrentFrame
Переменная подсказки	

Рисунок 10.329 – Параметры прямоугольника Имя текущей картинки

7. Настроить Менеджер визуализации. В его установках следует поставить галочку **Использовать строки Unicode**:



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

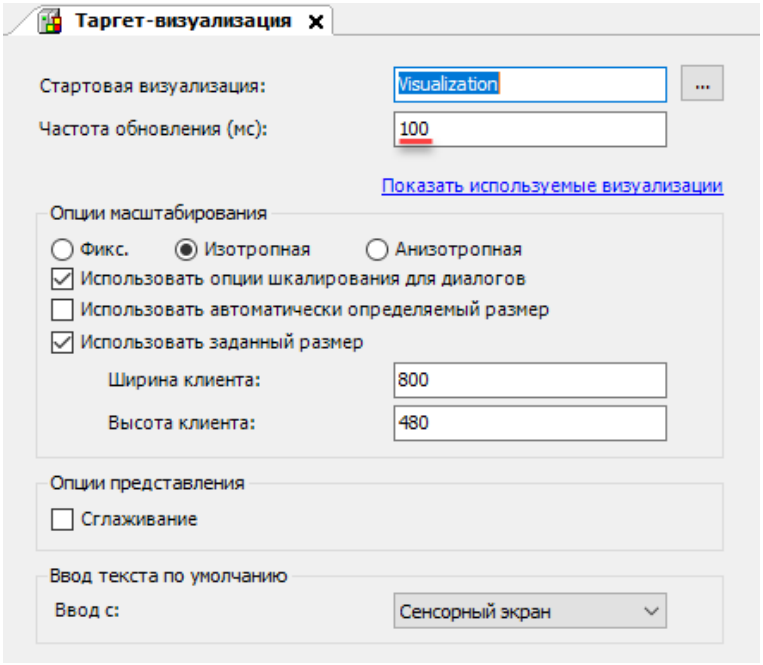


Рисунок 10.330 – Настройки таргет-визуализации

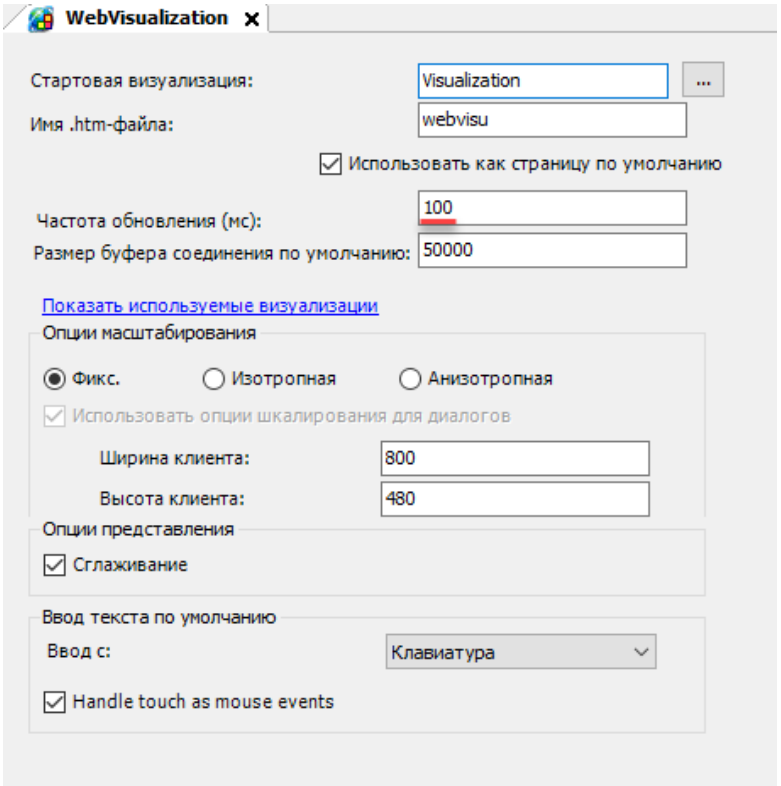


Рисунок 10.331 – Настройки web-визуализации

8. Запустить проект на виртуальном контроллере. Открыть в веб-браузере страницу http://localhost:8080/webvisu.htm, чтобы увидеть web-визуализацию проекта. ***Следует обратить внимание***, что браузер должен поддерживать **HTML5**. Проверить функционал проекта.

Рисунок 10.332 – Web-визуализация проекта в браузере Chrome

Чтобы запустить анимацию следует нажать переключатель **Старт**. Нажатие на кнопку **Стоп** остановит воспроизведение анимации.

Нажатие на кнопку **Сброс** приведет к сбросу анимации на начальный кадр.

10.3.5 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.

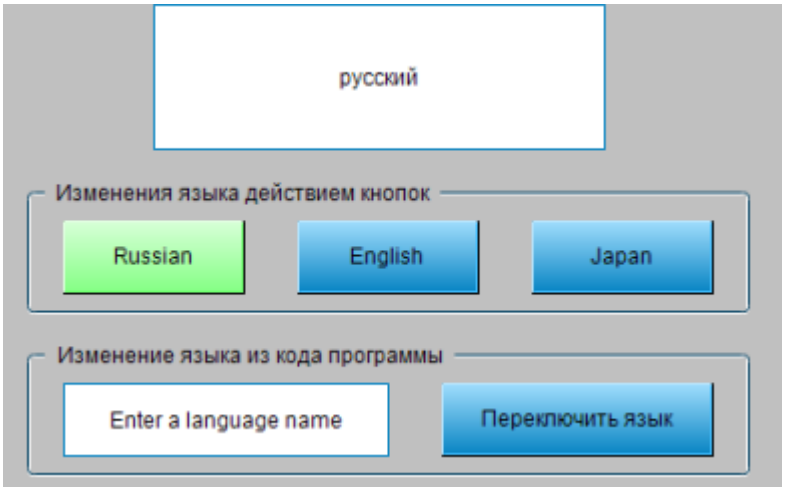


Рисунок 10.333 – Внешний вид примера Мультиязычность

Пример создан в среде **CODESYS V3.5 SP11 Patch 5** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с таргет-файлом **3.5.11.5**.

Пример доступен для скачивания: Example_Multilanguage.projectarchive

Для создания примера с использованием мультиязычности следует:

1. Создать новый ***стандартный*** проект **CODESYS** с названием **Example_Multilanguage** и настройками по умолчанию: таргет – CODESYS Control Win V3, язык PLC_PRG – **ST**.
2. Добавить в проект компонент Список текстов с названием по умолчанию **TextList**:

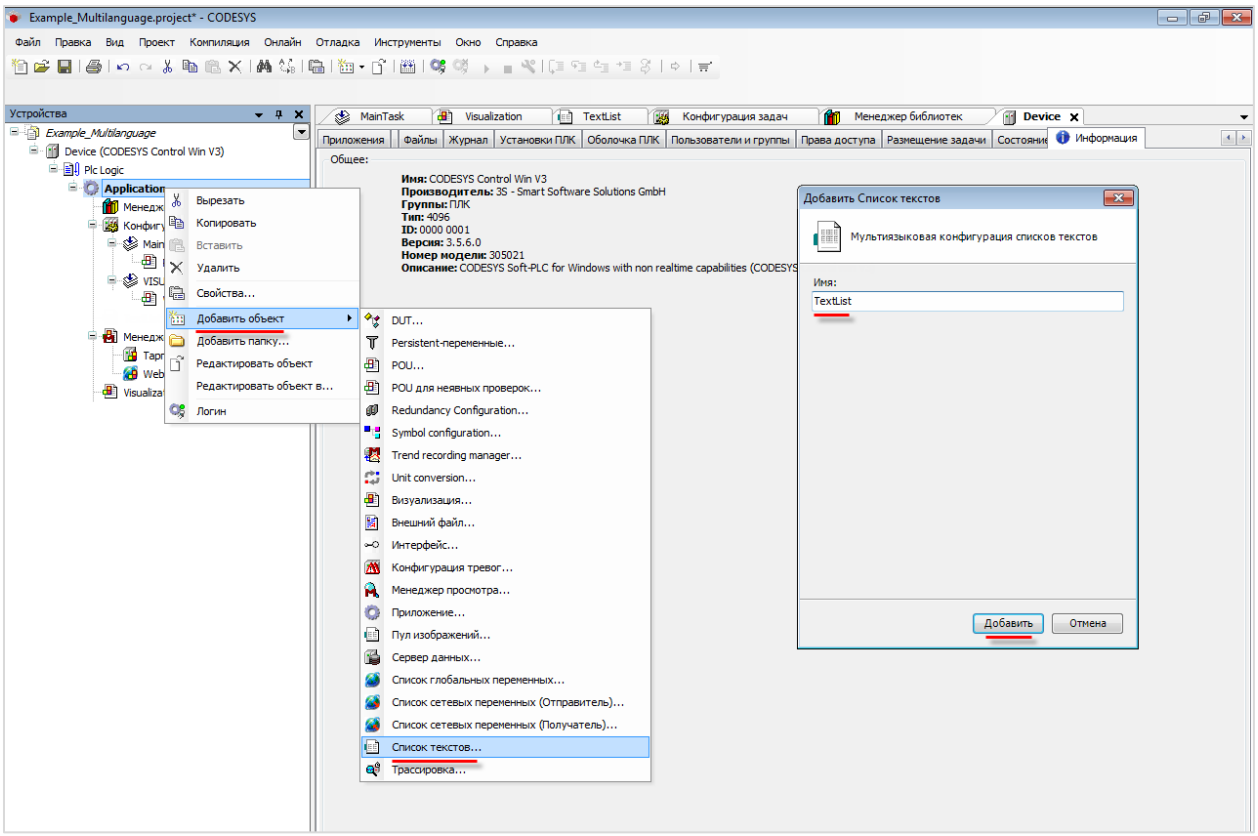


Рисунок 10.334 – Добавление в проект списка текстов

3. Нажать на любую область списка текстов **ПКМ** и выбрать команду **Добавить язык**:

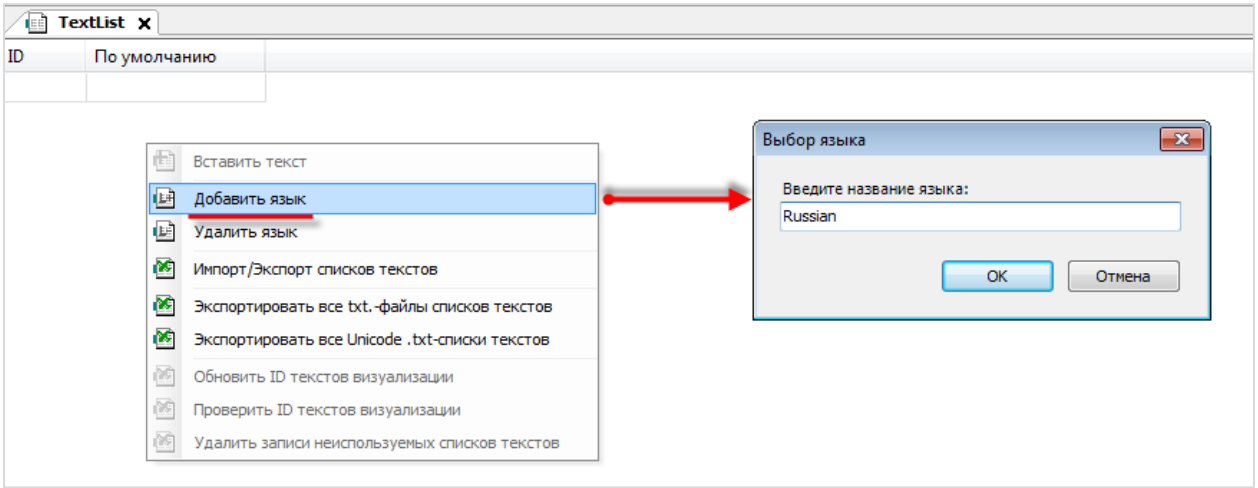


Рисунок 10.335 – Добавление языка в список текстов

Далее следует добавить в список три языка с названиями **Russian**, **English** и **Japan**.

4. Добавить в список текст со следующими настройками:

ID	По умолчанию	Russian	English	Japan
0		русский	english	日本語

Рисунок 10.336 – Содержимое списка текстов

5. В программе **PLC_PRG** объявить следующие переменные:

```
PROGRAM PLC_PRG
VAR
    bRussian:BOOL:=TRUE;                                    // переменная цвета кнопки Russian
    bEnglish:BOOL;                                          // переменная цвета кнопки English
    bJapan:BOOL;                                            // переменная цвета кнопки Japan

    sSetLanguageName:STRING:='Enter a language name'; // переменная, в которую записывается...
                                                            // ...имя языка, выбранного пользователем
    bSetLanguage:BOOL;                                      // переменная выбора языка
END_VAR
```

Рисунок 10.337 – Объявление переменных программы PLC_PRG

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

```
// переключение языка проект на язык, выбранный пользоватaлем

IF bSetLanguage=TRUE THEN
    VisuElems.CURRENTLANGUAGE:=sSetLanguageName;
END_IF

// окраска кнопки выбранного языка

IF VisuElems.CURRENTLANGUAGE='russian' OR VisuElems.CURRENTLANGUAGE='RUSSIAN' THEN
                bRussian:=TRUE;
                bEnglish:=FALSE;
                bJapan:=FALSE;
        ELSIF VisuElems.CURRENTLANGUAGE='english' OR VisuElems.CURRENTLANGUAGE='ENGLISH' THEN
                bEnglish:=TRUE;
                bRussian:=FALSE;
                bJapan:=FALSE;
        ELSIF VisuElems.CURRENTLANGUAGE='japan' OR VisuElems.CURRENTLANGUAGE='JAPAN' THEN
                bJapan:=TRUE;
                bRussian:=FALSE;
                bEnglish:=FALSE;
END_IF
```

Рисунок 10.338 – Код программы PLC_PRG

7. Добавить в проект экран визуализации **Visualization**. В его свойствах выбрать размер **480 × 272**. Экран будет содержать два элемента Текстовое поле, четыре элемента Кнопка и два элемента Группа.

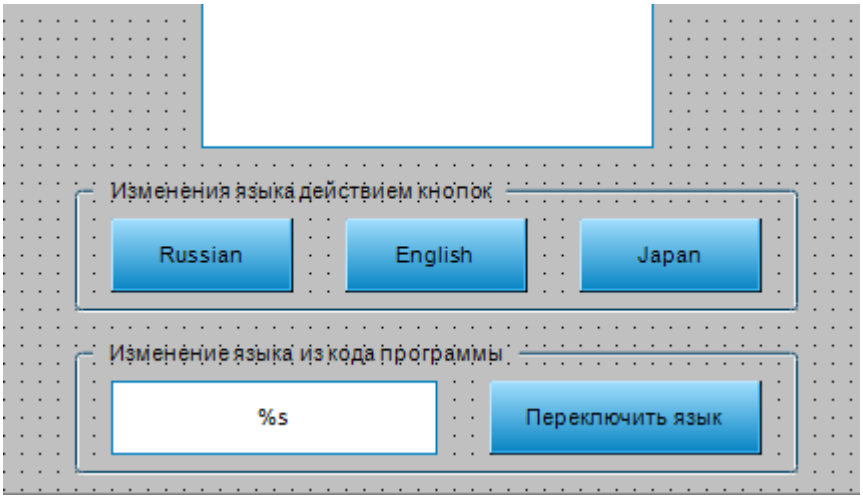


Рисунок 10.339– Содержимое экрана Visualization

Настроить текстовое поле (верхнее), в котором будет отображаться мультиязычный 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	

Рисунок 10.340 – Параметры элемента Текстовое поле

Затем следует настроить кнопки группы **Изменение языка действием кнопок**.

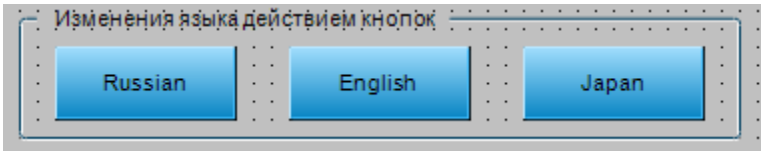


Рисунок 10.341 – Элементы группы Изменение языка действием кнопок

Во вкладке **Переменные состояния** к параметру **Переключить цвет** следует привязать переменные **bRussian**, **bEnglish**, **bJapan** соответственно:

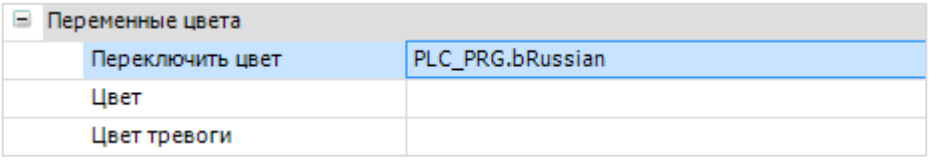


Рисунок 10.342 – Параметры кнопки Russian

Во вкладке InputConfiguration к параметру **OnMouseClick** следует привязать действия Изменить язык и Выполнить ST-код:

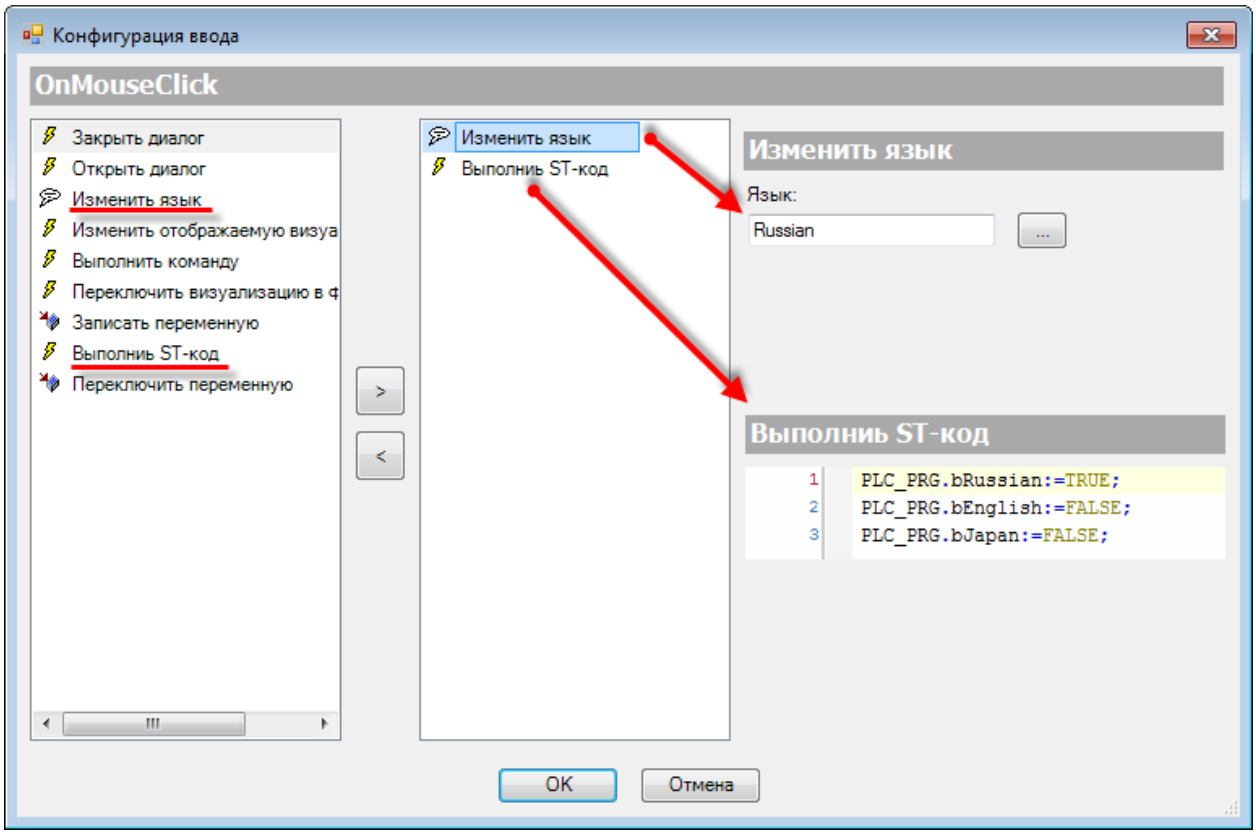


Рисунок 10.343 – Настройки действия кнопки Russian

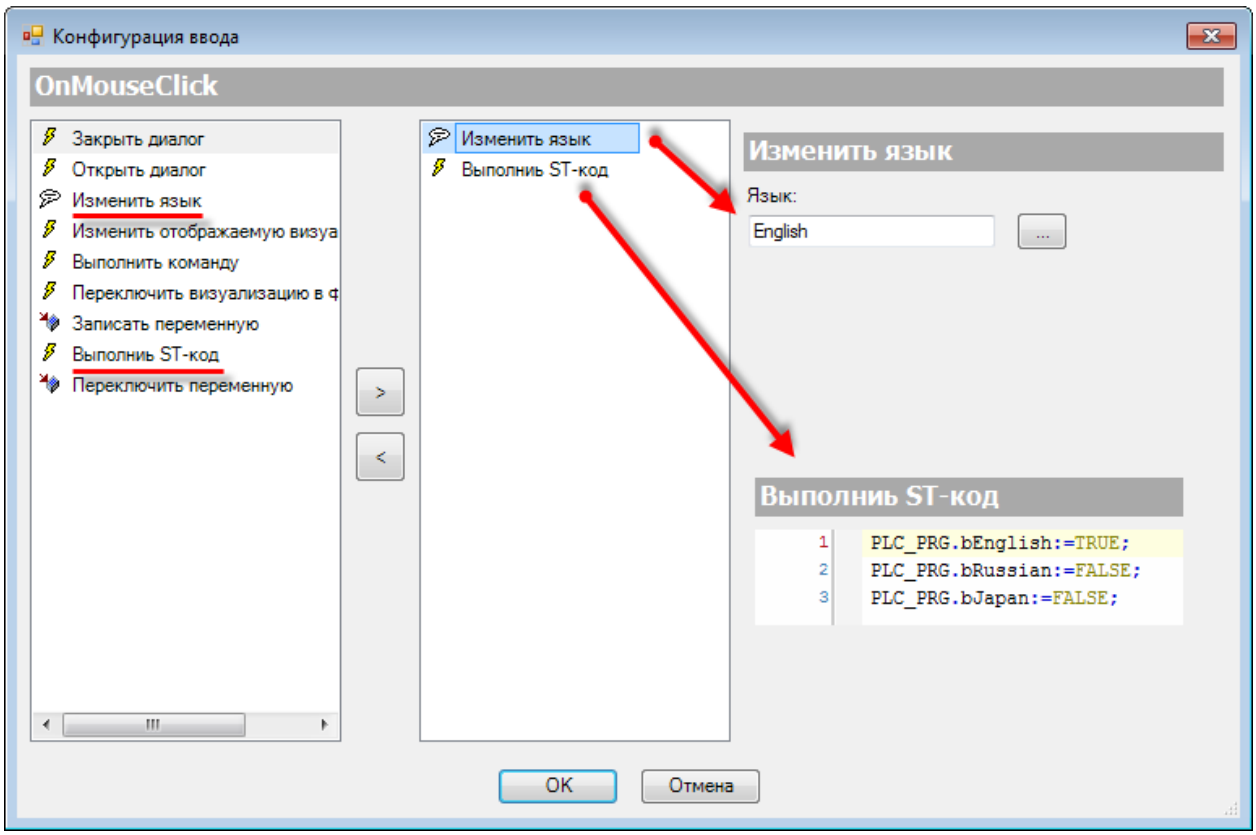


Рисунок 10.344 – Настройки действия кнопки English

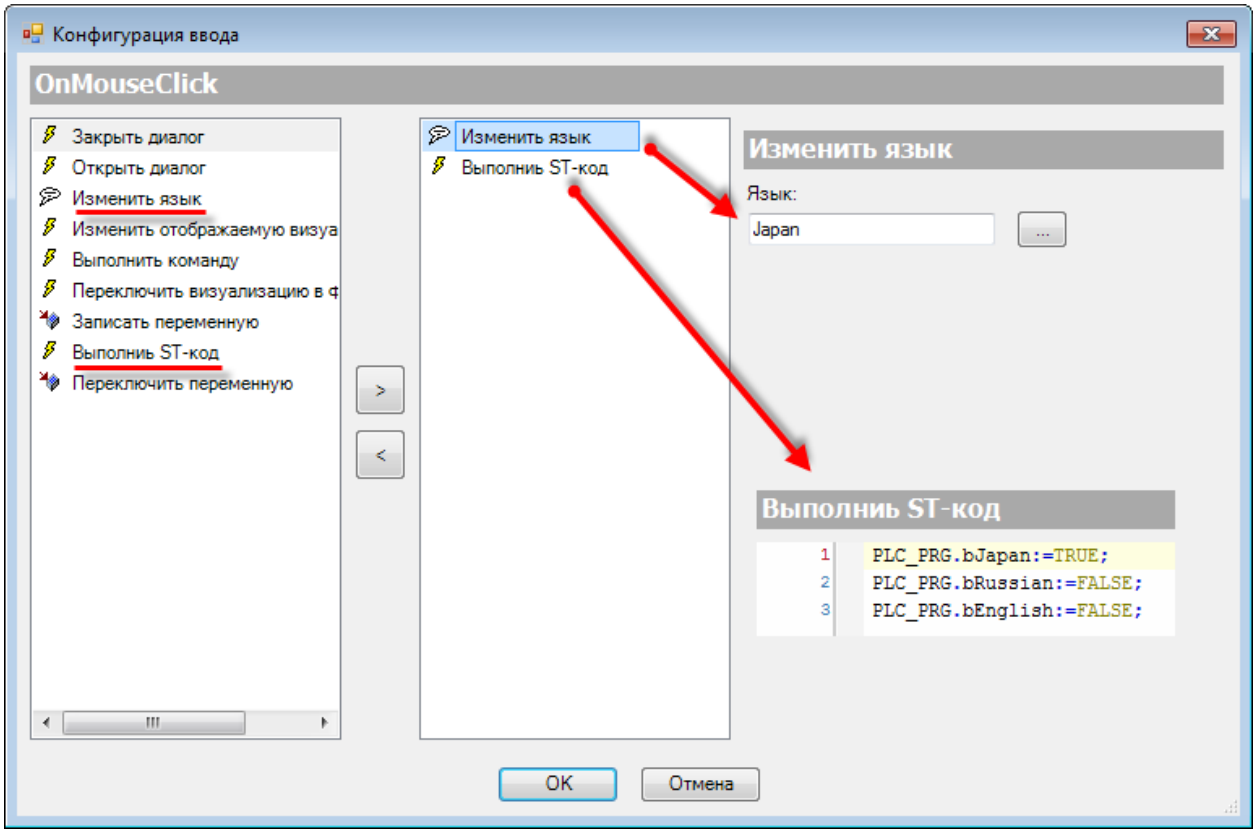


Рисунок 10.345 – Настройки действия кнопки Japan

Затем следует настроить элементы группы **Изменение языка из кода программы**.

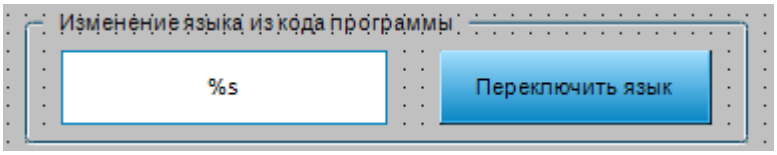


Рисунок 10.346 – Элементы группы Изменение языка из кода программы

К полю ввода языка следует привязать переменную **sSetLanguageName**:

Свойства

Фильтр | Сортировать по | Порядок сортировки | Эксперт

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...
⊟ OnMouseClick	Конфигурация...
Записать переменную	Variable:, InputType: VisuDialogs.Keypa

Рисунок 10.347 – Параметры элемента Текстовое поле

У элемента Текстовое поле во вкладке InputConfiguration к параметру **OnMouseClick** следует привязать действие Записать переменную:

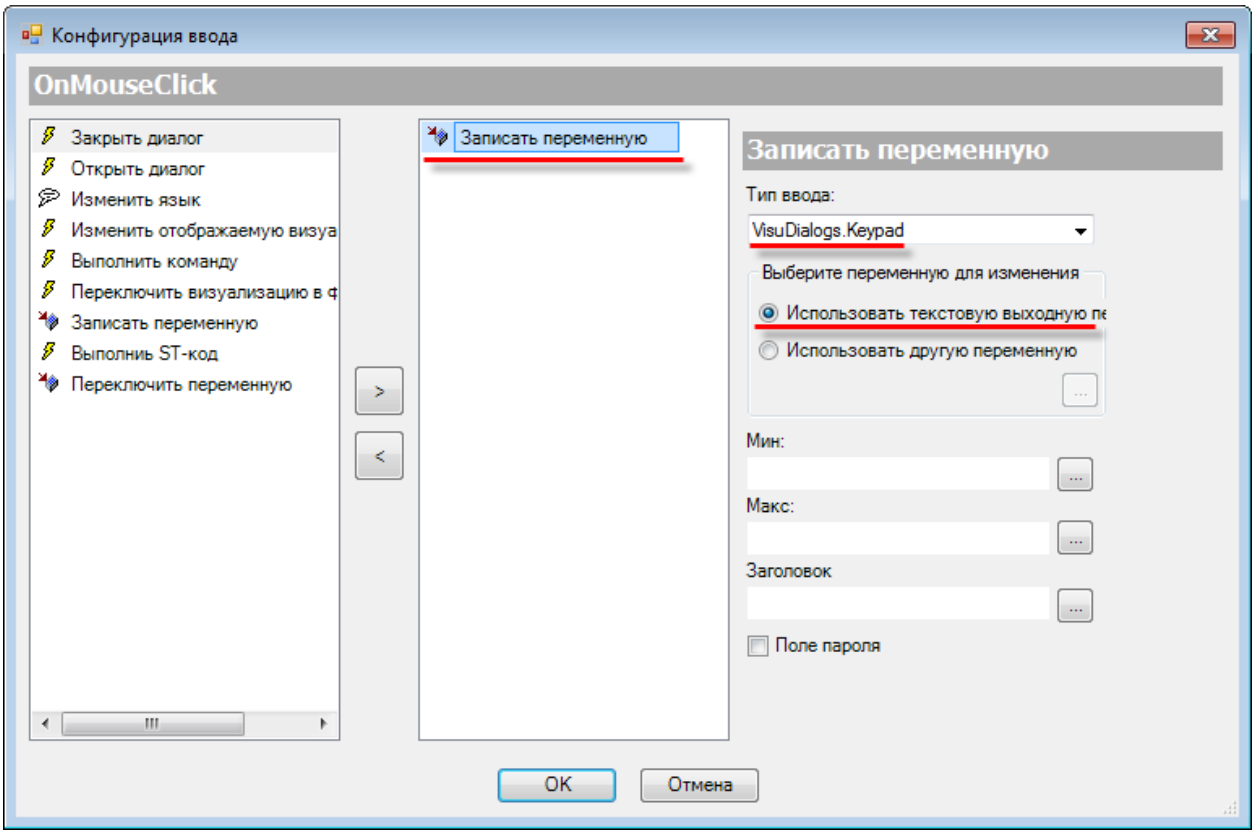


Рисунок 10.348 – Настройки действия элемента Текстовое поле

К кнопке **Переключить язык** следует привязать переменную **bSetLanguage**:

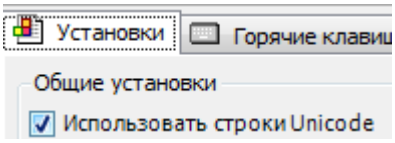
Свойства

Фильтр | Сортировать по | Порядок сортировки | Эксперт

Свойство	Значения
Имя элемента	GenElemInst_326
Тип элемента	Кнопка
ID текста	7
⊞ Позиция	
⊞ Центр	
⊞ Цвета	
Использовать градиентный цвет	☐
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...
OnMouseClick	Конфигурация...
OnMouseDown	Конфигурация...
OnMouseEnter	Конфигурация...
OnMouseLeave	Конфигурация...
OnMouseMove	Конфигурация...
OnMouseUp	Конфигурация...
⊟ Нажать	
Переменная	PLC_PRG.bSetLanguage
Переключить на FALSE	☐

Рисунок 10.349 – Параметры кнопки Переключить язык

8. Настроить Менеджер визуализации. В его установках следует поставить галочку **Использовать строки Unicode**:



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

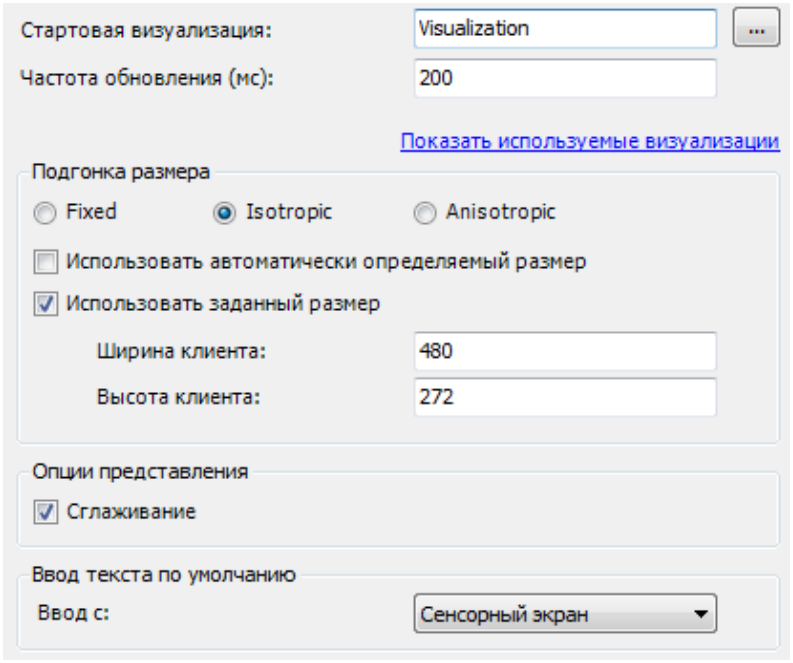


Рисунок 10.350 – Настройки таргет-визуализации

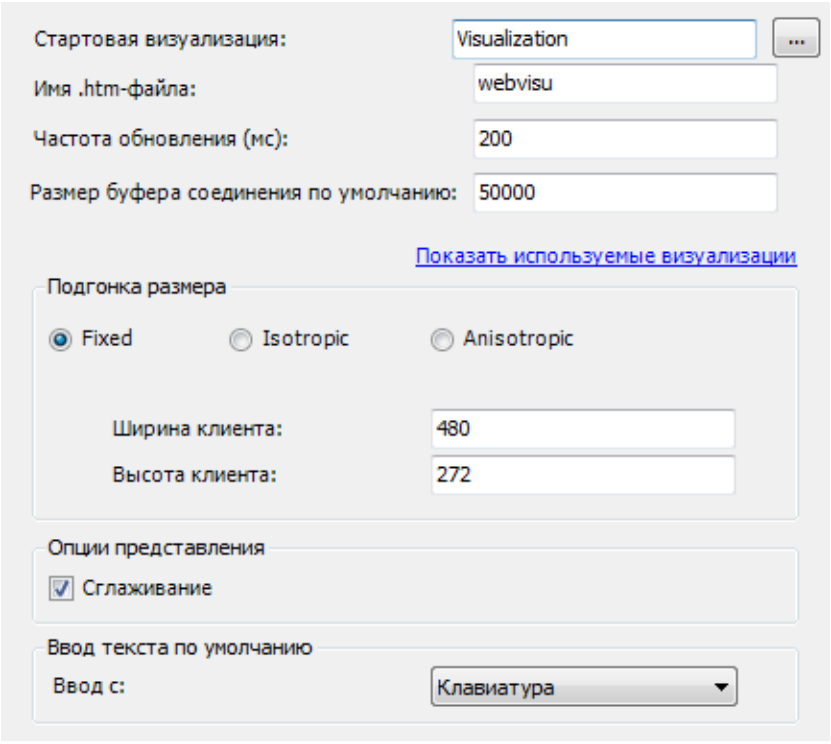


Рисунок 10.351 – Настройки web-визуализации

9. Запустить проект на виртуальном контроллере. Открыть в веб-браузере страницу http://localhost:8080/webvisu.htm, чтобы увидеть web-визуализацию проекта. ***Следует обратить внимание***, что браузер должен поддерживать **HTML5**. Проверить функционал проекта.

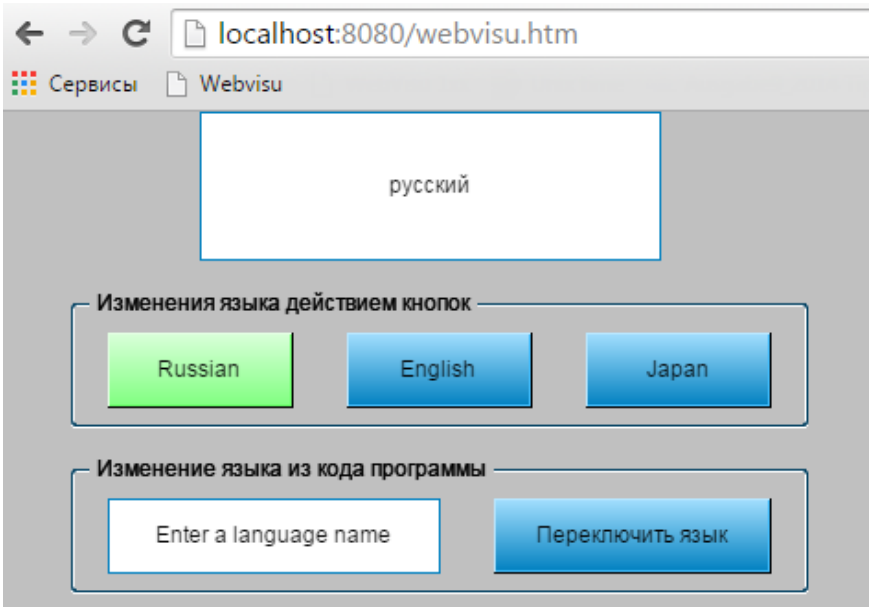


Рисунок 10.352 – Web-визуализация проекта в браузере Chrome

По умолчанию в проекте выбран русский язык. Чтобы переключить язык текста на английский следует нажать кнопку **English**. Чтобы переключить язык текста на японский, следует нажать кнопку **Japan**.

Далее следует нажать на поле ввода имени язык и ввести **ENGLISH**. Затем нажать кнопку **Переключить язык**.

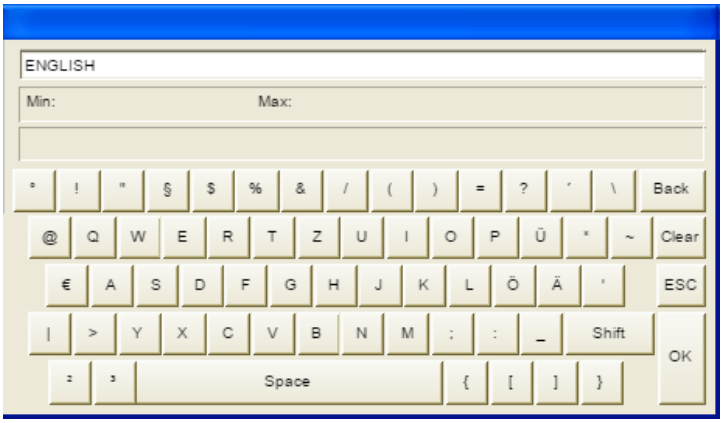
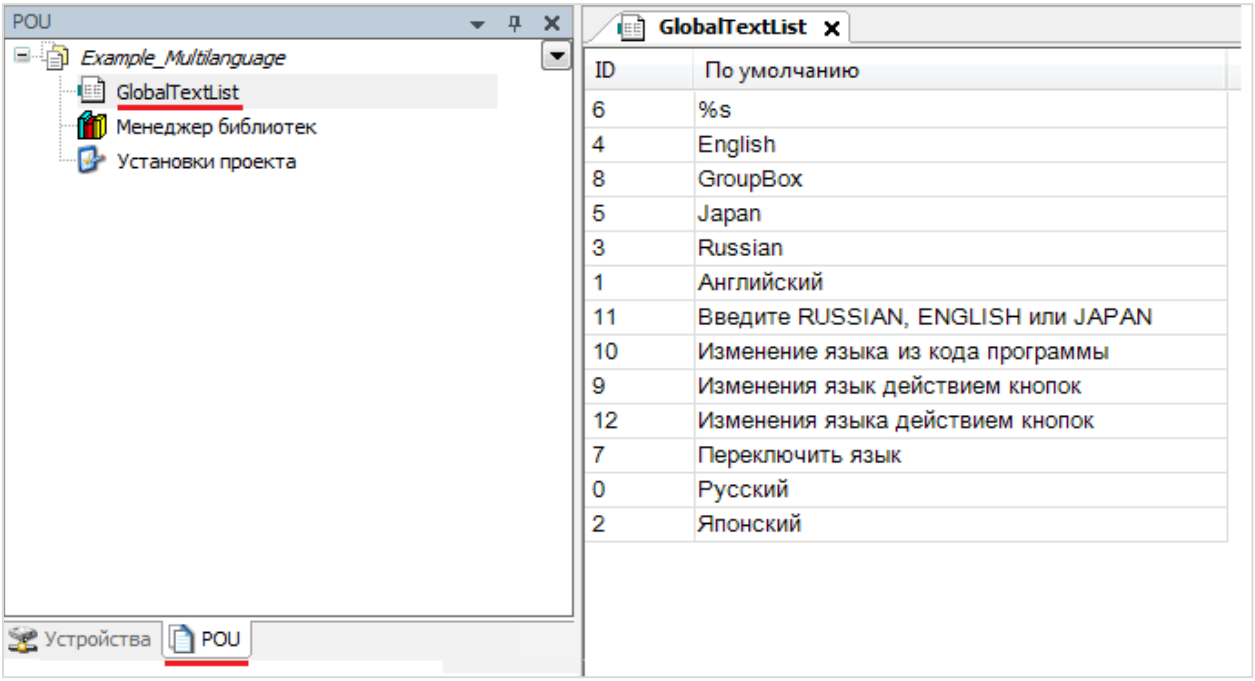


Рисунок 10.353 – Ввод имени языка с помощью экранной клавиатуры

Это приведет к переключению языка текста на английский. ***Следует обратить 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.

Следует обратить внимание, что создаваемый в визуализации текст автоматически попадает в список текстов **GlobalTextList**, расположенный на панели устройств во вкладке **POU**:



ID	По умолчанию
6	%s
4	English
8	GroupBox
5	Japan
3	Russian
1	Английский
11	Введите RUSSIAN, ENGLISH или JAPAN
10	Изменение языка из кода программы
9	Изменения язык действием кнопок
12	Изменения языка действием кнопок
7	Переключить язык
0	Русский
2	Японский

Рисунок 10.354 – Глобальный список текстов проекта

10.3.6 Тиражирование элементов

Данный пример посвящен тиражированию элементов.

Во время разработки 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 данном случае, вместо создания элементов вручную, можно воспользоваться командой тиражирования элементов (**Размножить элемент визуализации**). Для этого переменные элементов, создаваемых тиражированием, предварительно должны быть собраны в массив.

Тиражировать можно следующие элементы:

1. Группы Базовые;
2. Группы Индикаторы/Переключатели/Изображения;
3. Кнопка, Текстовое поле, Кнопка-флажок, Управление вращением.

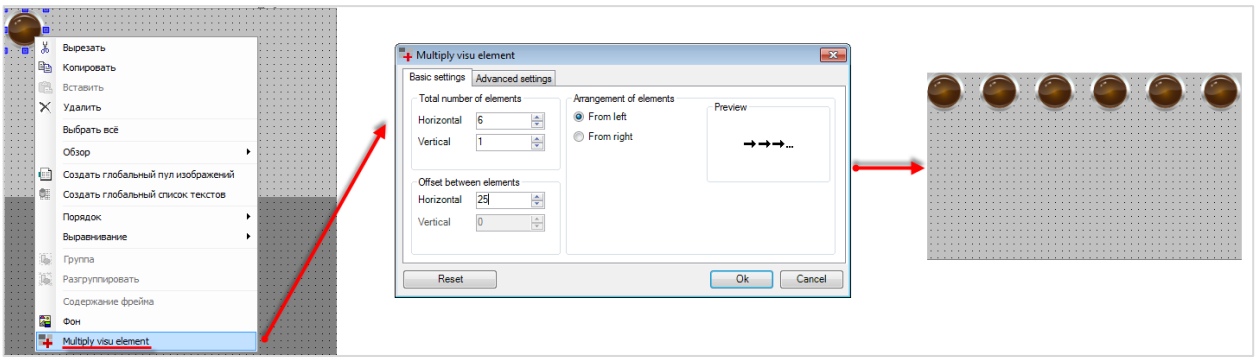


Рисунок 10.355 – Использование тиражирования элементов в проекте

Для создания примера с использованием функции тиражирования элементов следует:

1. Создать новый ***стандартный*** проект **CODESYS** с названием **Example_MultiplyVisuElements** и настройками по умолчанию: таргет – CODESYS Control Win V3, язык PLC_PRG – **ST**.
2. В программе **PLC_PRG** объявить массив переменных типа **BOOL**:

```
1  PROGRAM PLC_PRG
2  VAR
3      arrLamps:ARRAY [1..6] OF BOOL;  // массив переменных тиражируемых элементов
4  END_VAR
```

Рисунок 10.356 – Объявление переменных программы PLC_PRG

3. Создать в проекте экран визуализации **Visualization**. Добавить на экран элемент Индикатор и привязать к нему переменную-заполнитель **arrLamps[$FIRSTDIM$]:**

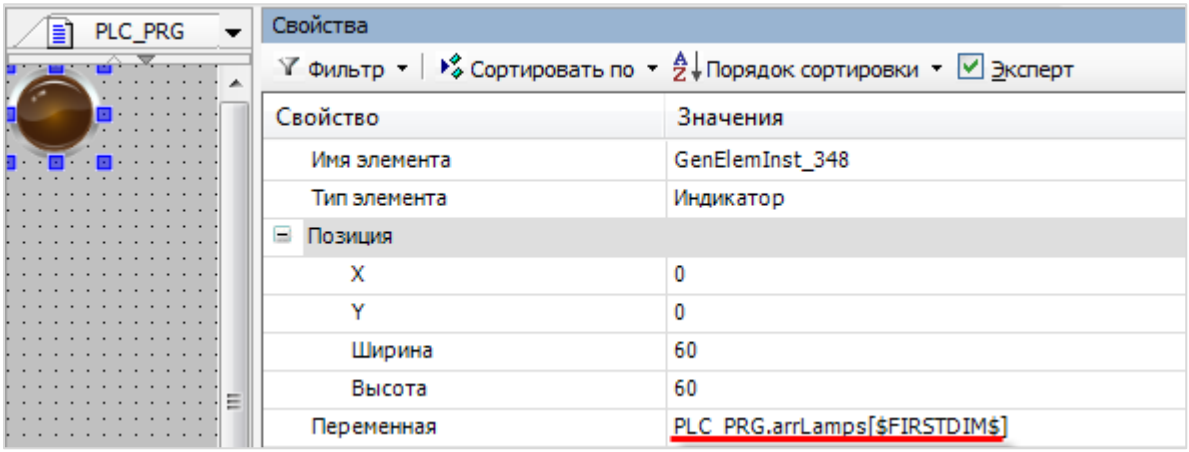


Рисунок 10.357 – Параметры элемента Индикатор

Затем следует нажать на элемент **ПКМ** и в контекстном меню выбрать команду **Размножить элемент визуализации**:

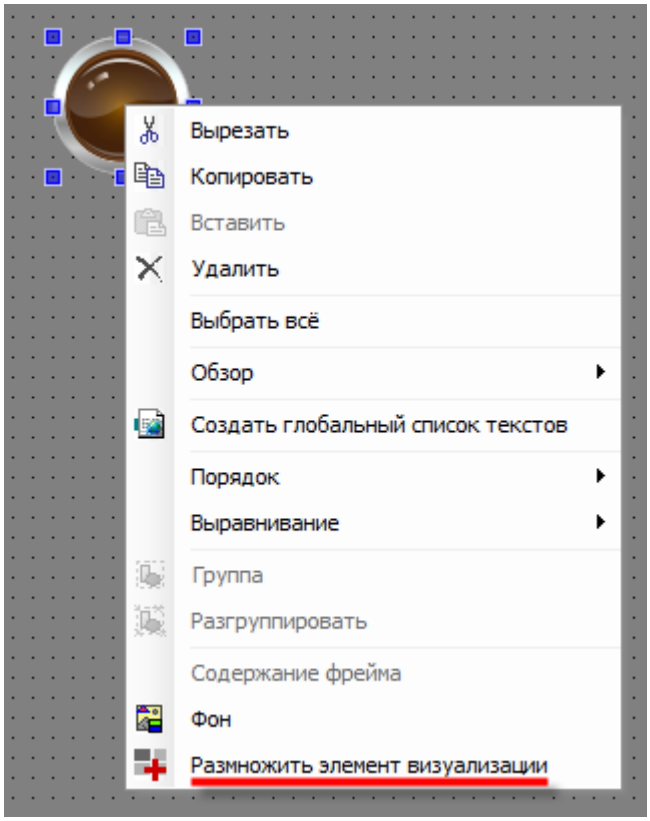


Рисунок 10.358 – Команда тиражирования элементов

В открывшемся окне конфигурации тиражирования следует задать следующие настройки:

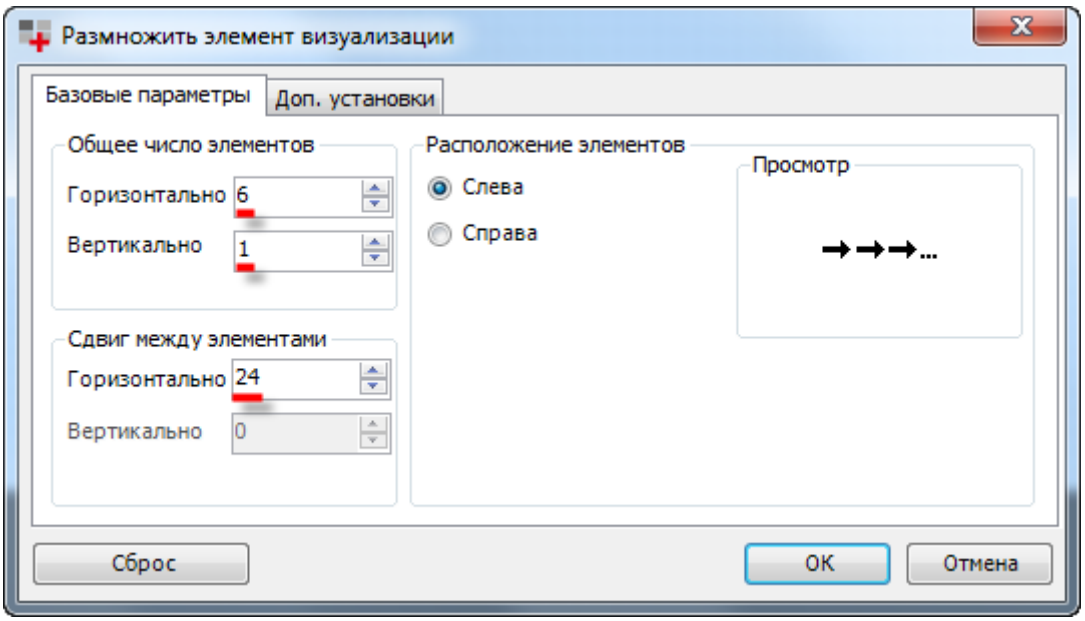


Рисунок 10.359 – Окно конфигурации тиражирования

В результате будут созданы шесть индикаторов, расположенные в один ряд, с зазором в 24 пикселя между соседними элементами:

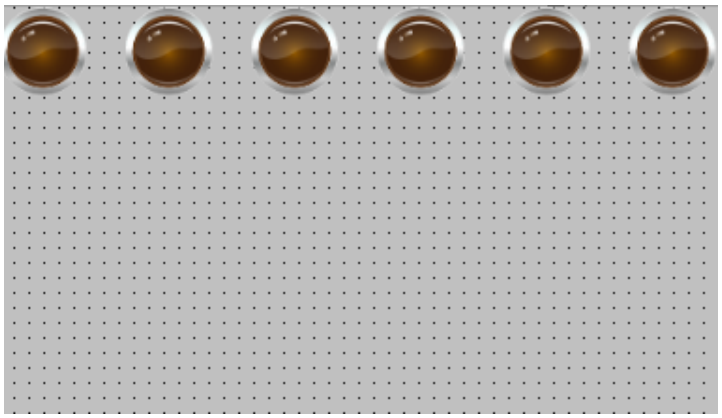


Рисунок 10.360 – Результат тиражирования элемент Индикатор

К первому индикатору привязана переменная **arrLamps[1],** ко второму – **arrLamps[2]** и т. д. В случае необходимости номер первого элемента массива и шаг между ними может быть задан на вкладке **Доп. установки** окна тиражирования элементов.

10.3.7 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. 10.3.1), изменения языка (см. п. 10.3.5) и т. д.

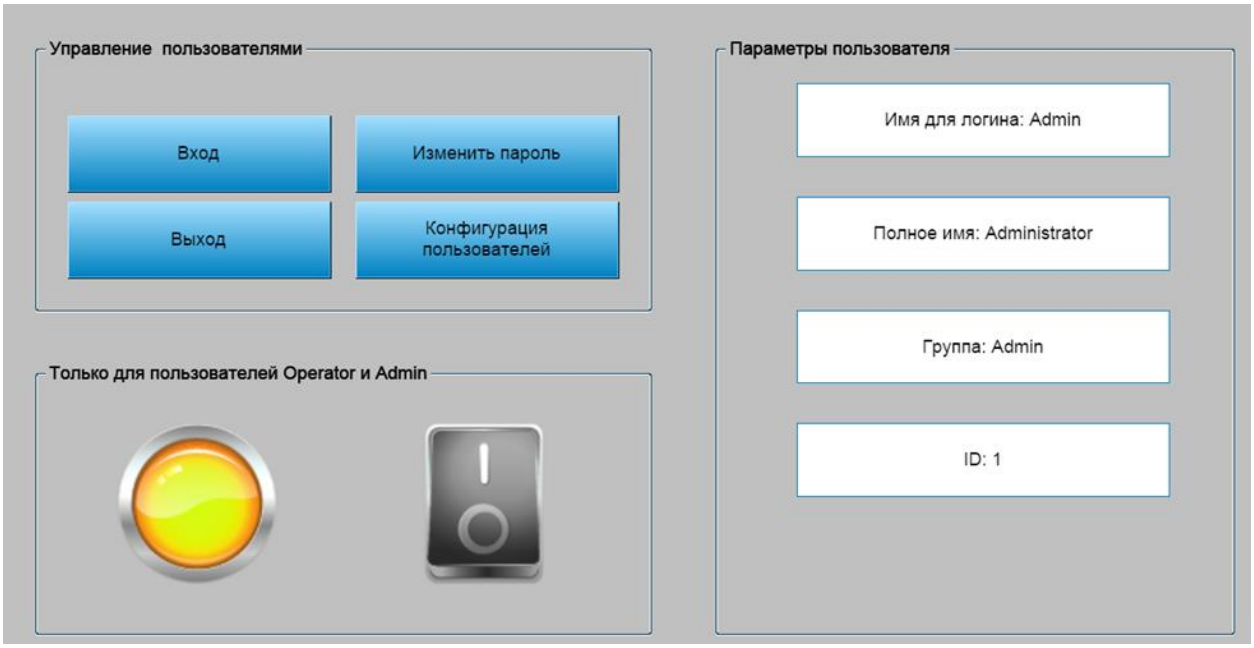


Рисунок 10.361 – Внешний вид примера Управление пользователями

Пример создан в среде **CODESYS V3.5 SP11 Patch 5** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с таргет-файлом **3.5.11.5**.

Пример доступен для скачивания: Example_UserManagment.projectarchive

Для создания примера с использованием парольного доступа следует:

1. Создать новый ***стандартный*** проект **CODESYS** с названием **Example_UserManagment** и настройками по умолчанию: таргет – CODESYS Control Win V3, язык PLC_PRG – **ST**.
2. В программе **PLC_PRG** объявить переменную **bLamp** типа **BOOL**:

```
1  PROGRAM PLC_PRG
2  VAR
3      bLamp:BOOL;          // переменная индикатора
4  END_VAR
```

Рисунок 10.362 – Объявление переменных программы PLC_PRG

3. Добавить в проект экран визуализации **Visualization**. В его свойствах выбрать размер **800 × 480**.
4. В Менеджере визуализации открыть вкладку Управление пользователями и нажать кнопку **Создать управление пользователями с группами и пользователями по умолчанию**:

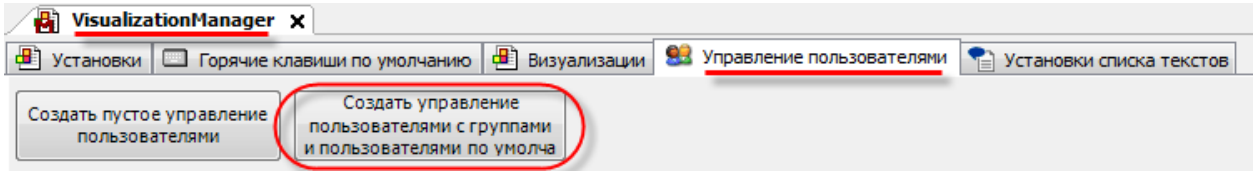


Рисунок 10.363 – Создание управления пользователями

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.

Настройки группы пользователей:

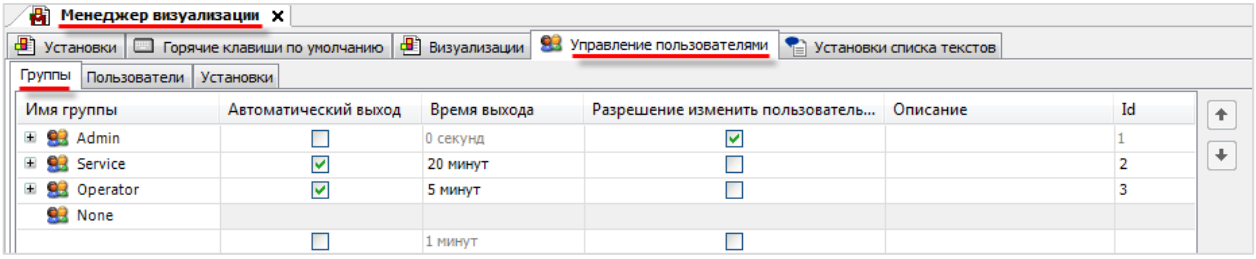


Рисунок 10.364 – Настройки групп пользователей

Для групп **Service** и **Operator** следует установить галочку **Автоматический выход** и указать время бездействия, через которое будет произведен принудительный выход пользователя из системы. Для группы **Admin** следует установить галочку **Разрешение изменить пользовательские данные** – это позволит членам этой группы открыть соответствующее диалоговое окно в процессе работы проекта.

Настройка пользователей:

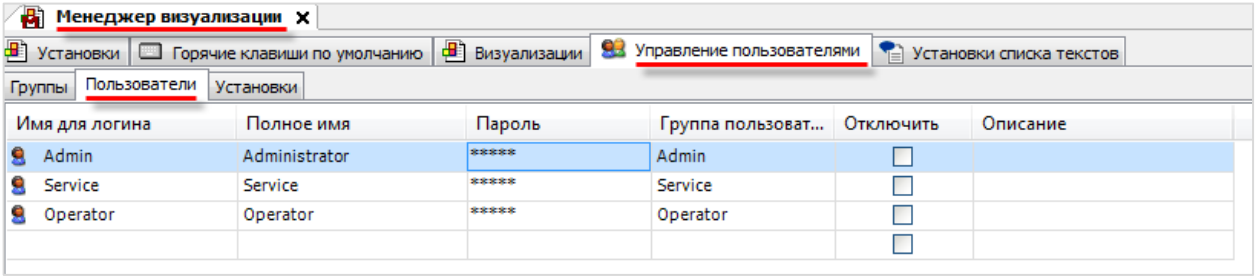


Рисунок 10.365 – Настройки пользователей

Пользователю **Admin** следует задать пароль **1**, пользователю **Service** – пароль **2**, пользователю **Operator** – пароль **3**. После изменения числа символов, пароль в столбце будет отображаться в виде пяти «звездочек» (*****).

На вкладке **Установки** присутствует настройка **Поведение при выходе**, которая позволяет в случае выхода 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:

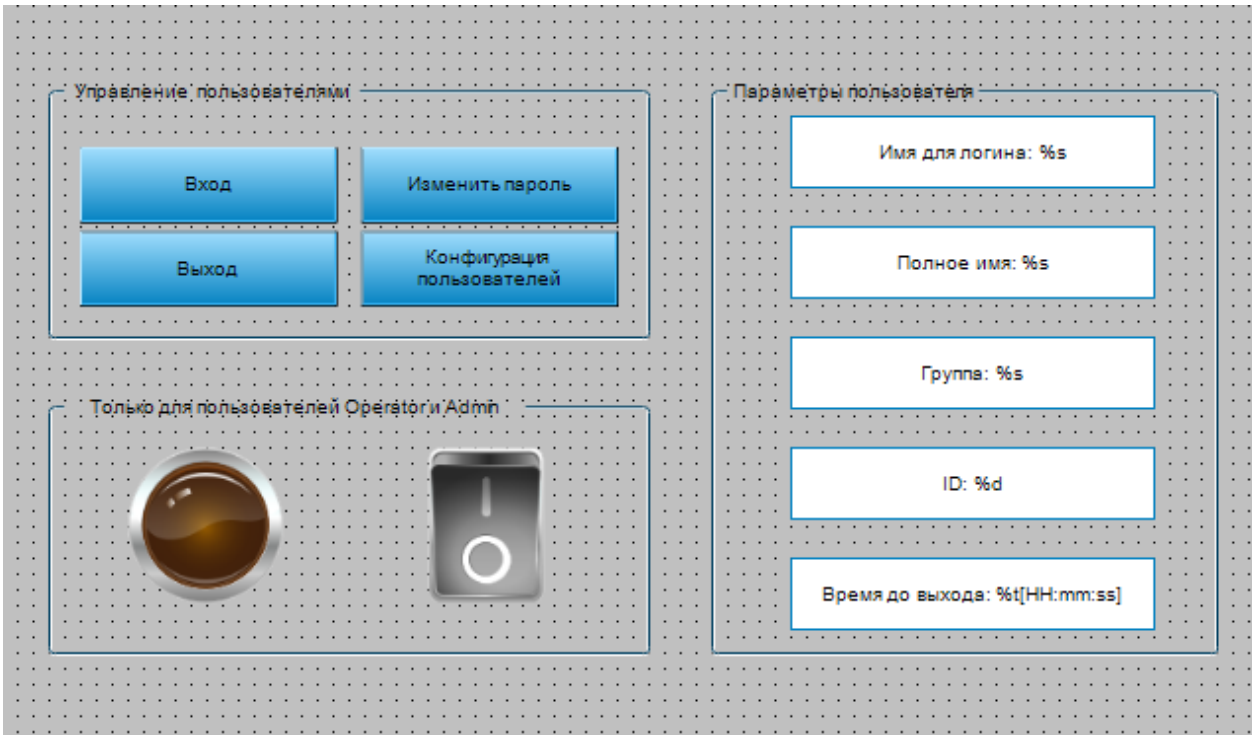


Рисунок 10.366 – Содержание экрана Visualization

Сначала следует настроить кнопки вкладки **Управление пользователями**.

У кнопки **Вход** во вкладке InputConfiguration к параметру **OnMouseClick** привязывается действие Управление пользователями с диалогом **Войти**. Действие становится доступным только после настройки одноименной вкладки **в Менеджере визуализации**.

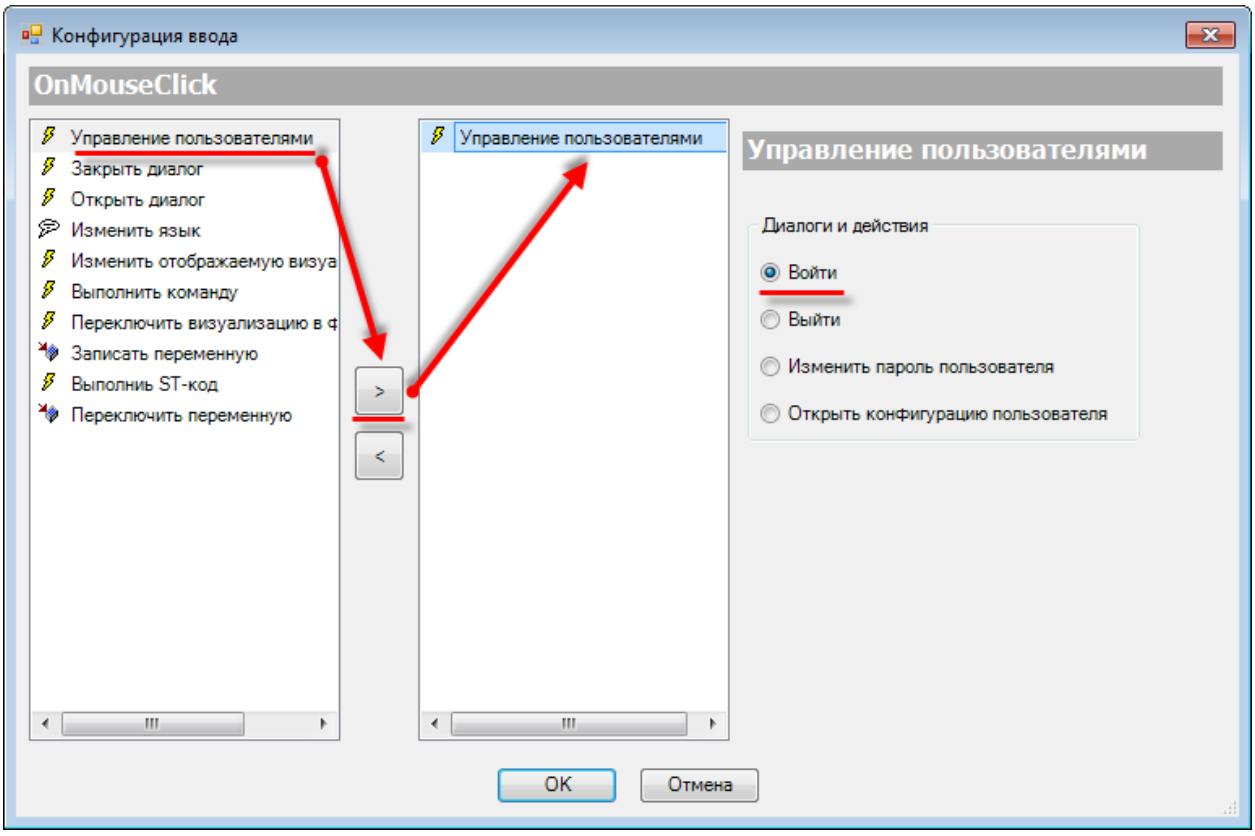


Рисунок 10.367 – Настройки действия Управление пользователями

К кнопке **Выход** привязывается действие **Управление пользователями** с диалогом **Выйти**.

К кнопке Изменить пароль привязывается действие Управление пользователями с диалогом Изменить пароль пользователя.

К кнопке Конфигурация пользователей привязывается действие Управление пользователями с диалогом Открыть конфигурацию пользователей.

Затем настраиваются элементы вкладки Только для пользователей Operator и Admin.

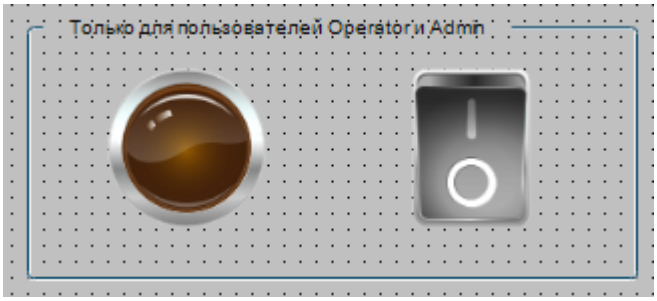


Рисунок 10.368 – Элементы вкладки Только для пользователей Operator и Admin

Вкладка будет содержать элемент Индикатор и Клавишный переключатель, оба элемента будут видны только пользователям групп **Operator** и **Admin**, **Клавишный переключатель** будет реагировать на нажатие только пользователей группы **Admin**.

Для этого в настройках элемента **Индикатор** следует дважды нажать **ЛКМ** на параметр **Права доступа**. Параметр становится доступным только после настройки вкладки Управление пользователями в **Менеджере визуализации**.

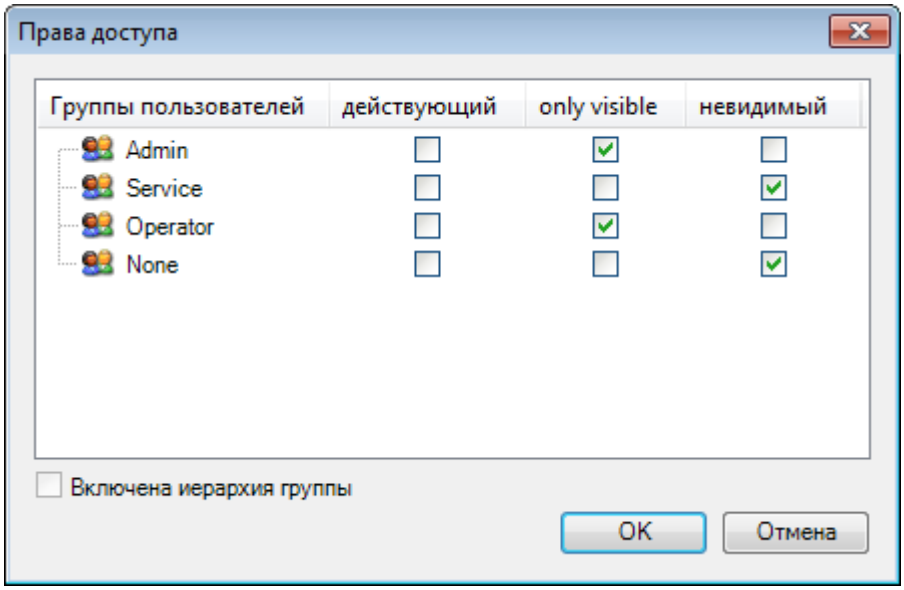


Рисунок 10.369 – Настройка прав доступа для элемента Индикатор

Для пользователей **None** и **Service** следует установить галочку **Невидимый**, для пользователей **Operator** и **Admin** – галочку **Only Visible**.

Настройки прав доступа для элемента Клавишный переключатель будут выглядеть следующим образом:

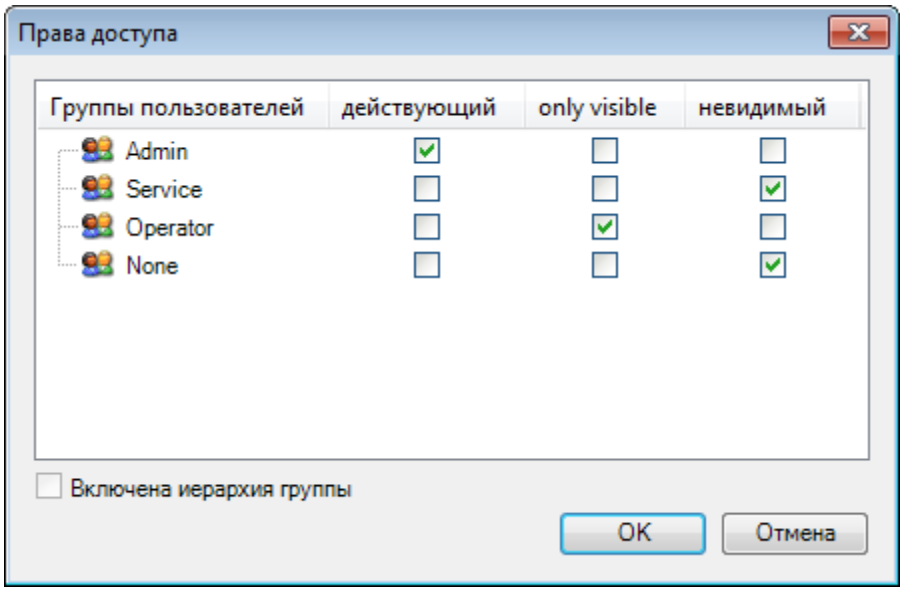


Рисунок 10.370 – Настройка прав доступа для элемента Клавишный переключатель

Далее следует настроить элементы вкладки **Параметры пользователя**. Вкладка содержит шесть элементов Текстовое поле, которые будут использоваться для отображения информации о текущем пользователе с помощью системных переменных.

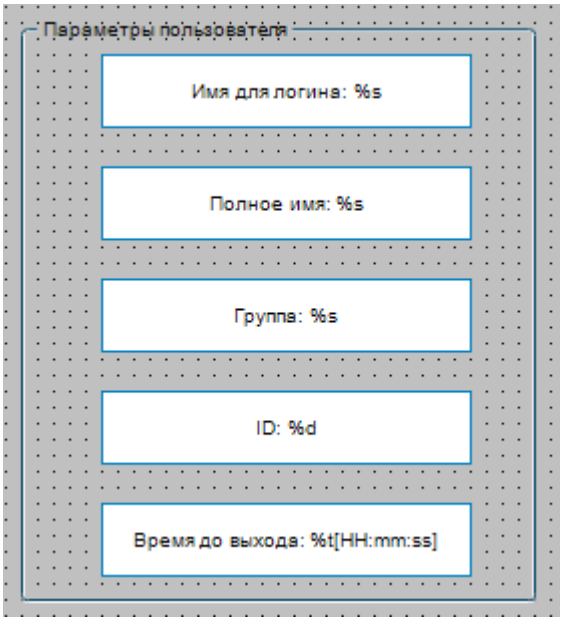


Рисунок 10.371 – Элементы вкладки Параметры пользователя

Настройки вкладки приведены в таблице 10.3.

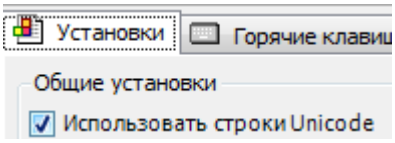
Таблица 10.3 – Переменные элементов вкладки Параметры пользователя

№ пп.	Текст элемента	Текстовая переменная	Тип переменной
1	Имя для логина: %s	VisuElems.CurrentUserName	WSTRING
2	Полное имя: %s	VisuElems.CurrentFullUserName	WSTRING
3	Группа: %s	VisuElems.CurrentUserGroupName	WSTRING
4	ID: %d	VisuElems.CurrentUserGroupId	DWORD
5	Время до выхода: %t[HH:mm:ss]	VisuElems.CurrentRemainingAutoLogoutTime	TIME
Переменная невидимости			
6	Время до выхода: %t[HH:mm:ss]	NOT(VisuElems.CurrentUseAutoLogoutTime)	BOOL

ПРИМЕЧАНИЕ

Системные переменные являются контекстно-зависимыми – то есть их значения будут отличаться для разных клиентов визуализации. Эти переменные не подходят для обработки в коде программы – для этого следует использовать другие средства (см. пример).

6. Настроить Менеджер визуализации. В его установках следует поставить галочку **Использовать строки Unicode**:



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

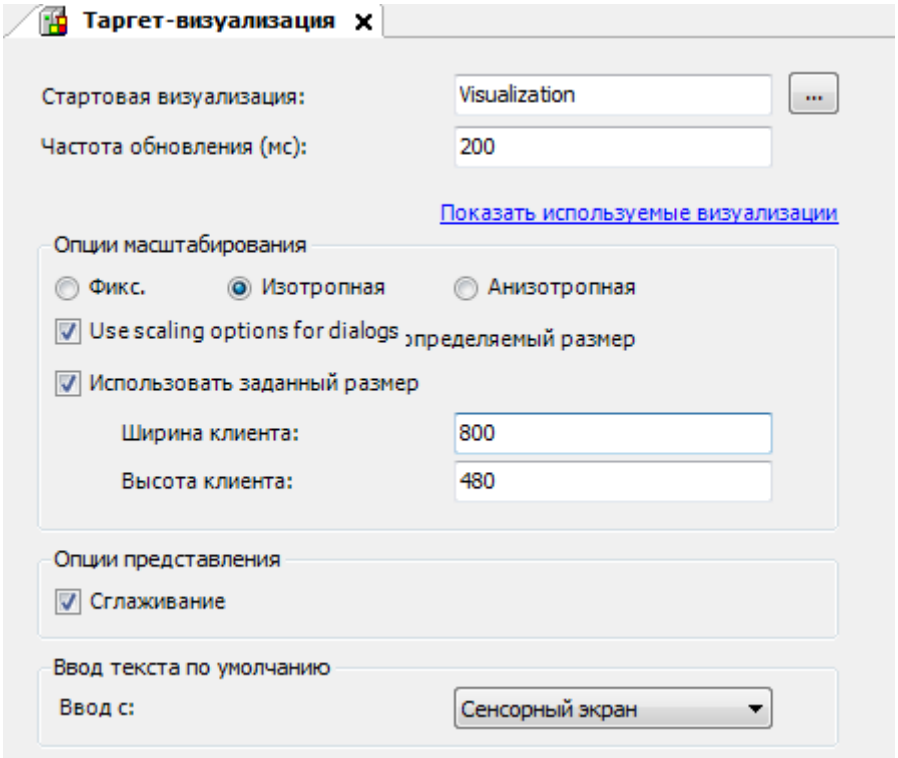


Рисунок 10.372 – Настройки таргет-визуализации

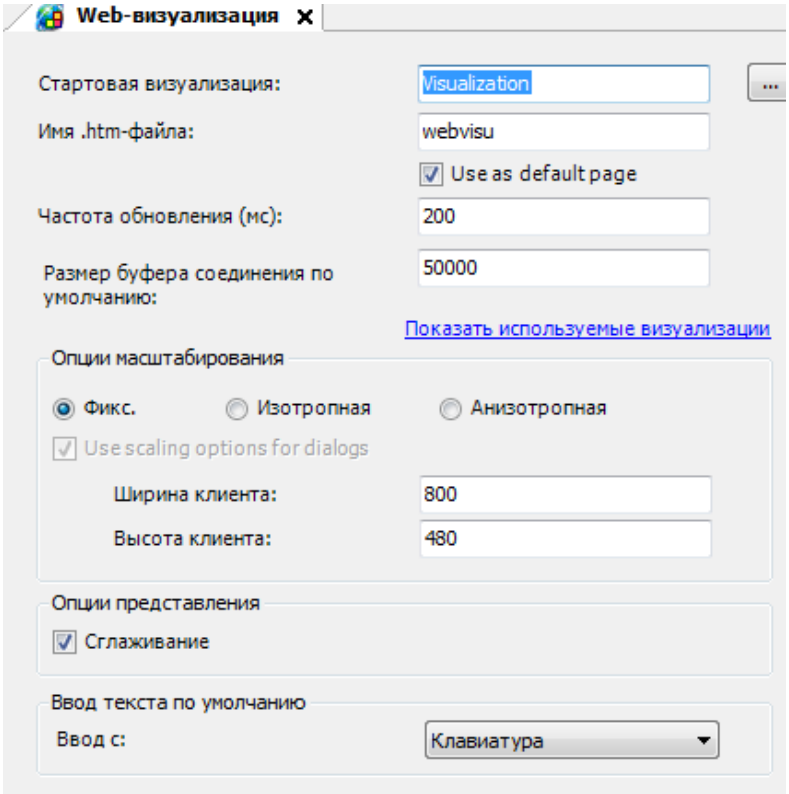


Рисунок 10.373 – Настройки web-визуализации

7. Запустить проект на виртуальном контроллере. Открыть в веб-браузере страницу http://localhost:8080/webvisu.htm, чтобы увидеть web-визуализацию проекта. ***Следует обратить внимание***, что браузер должен поддерживать **HTML5**. Проверить функционал проекта.

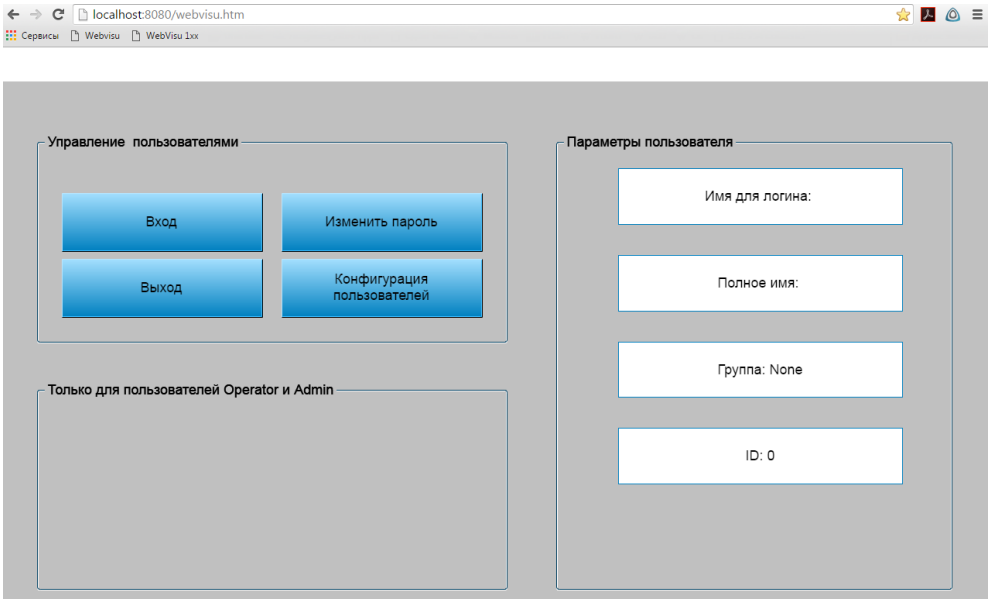


Рисунок 10.374 – Web-визуализация проекта в браузере Chrome

После загрузки проекта, в системе еще нет авторизированного 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.

Сначала следует нажать кнопку **Вход** и ввести логин **Operator** (регистр имеет значение), пароль – **3**.

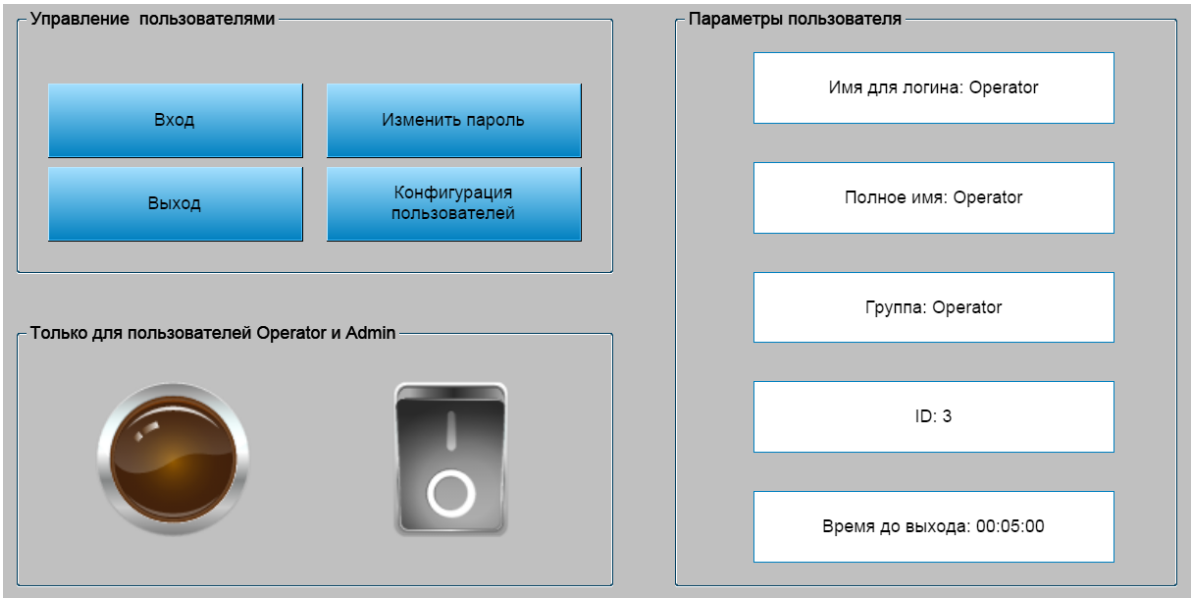


Рисунок 10.375 – Авторизация пользователя Operator

Во вкладке **Только для пользователей Operator и Admin** отобразятся индикатор и переключатель, во вкладке **Параметры пользователя** – информация о пользователе **Operator**. Спустя пять минут бездействия произойдет принудительный выход пользователя из системы.

ПРИМЕЧАНИЕ
Переключатель и кнопка **Конфигурация пользователей** являются неактивными для пользователя **Operator**.

Затем нажать кнопку **Изменить пароль**:

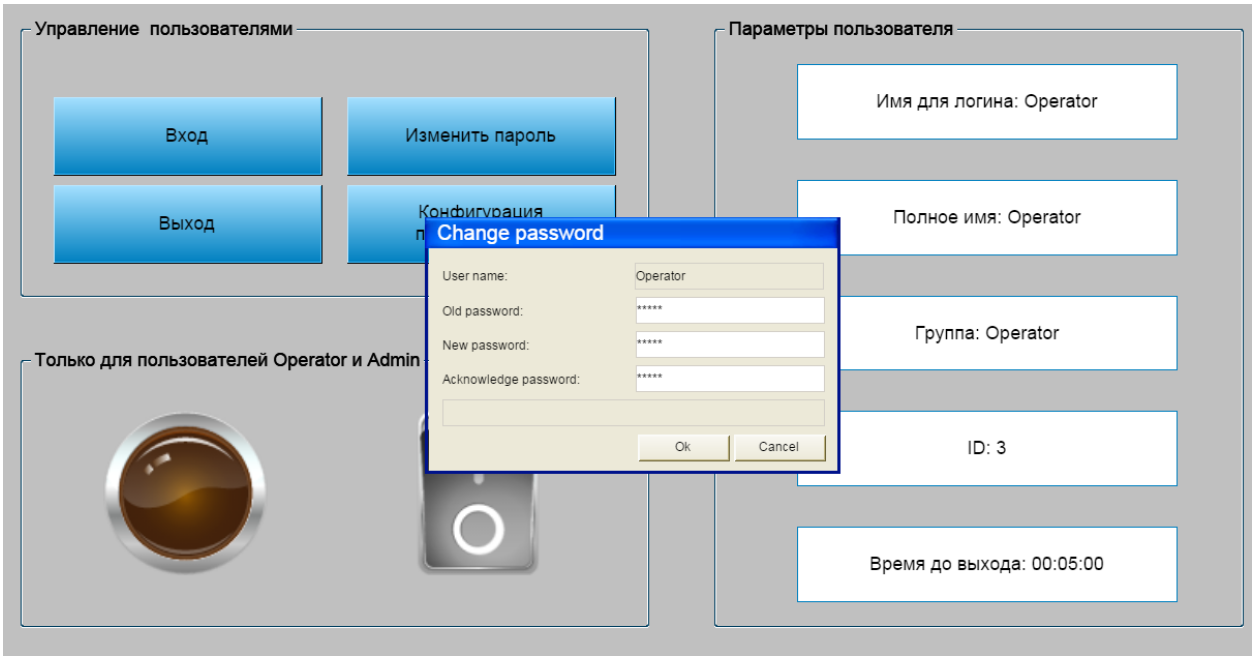


Рисунок 10.376 – Диалоговое окно изменения пароля

Для изменения пароля, следует один раз ввести старый пароль (**Old password**) и два раза – новый (**New password**, **Acknowledge password**).

Затем нажать кнопку **Вход** и ввести логин **Admin** (регистр имеет значение), пароль – **1**.

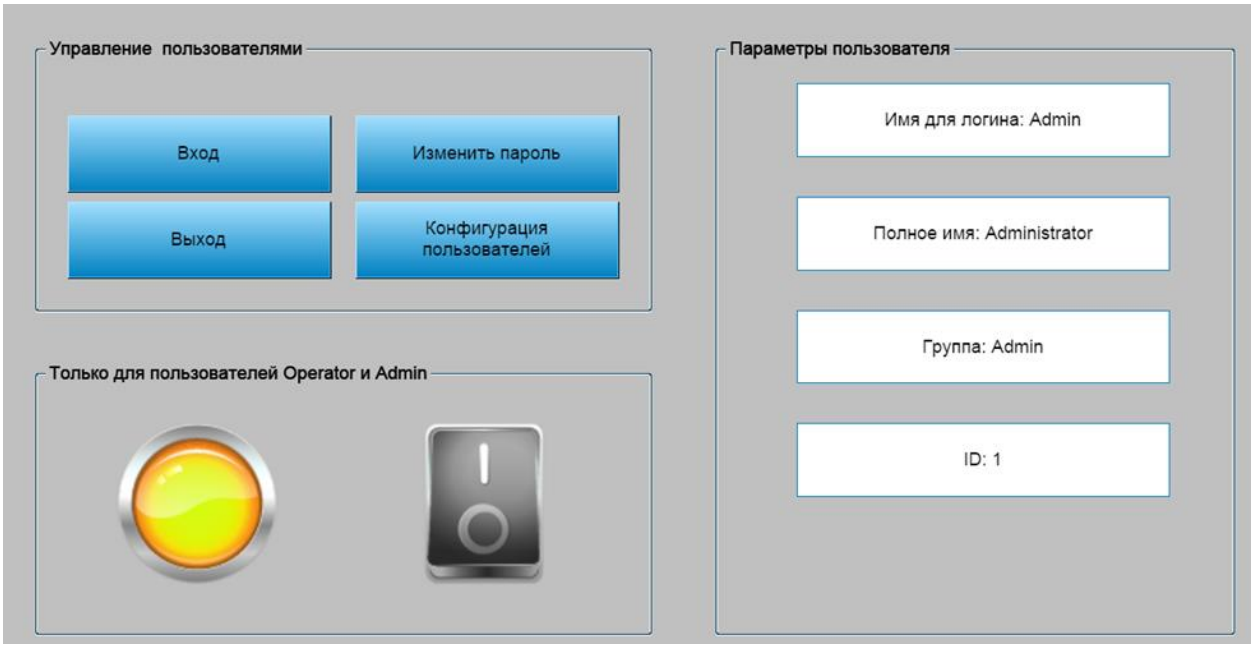


Рисунок 10.377 – Авторизация пользователя Admin

Для пользователя **Admin** доступно управление индикатором с помощью нажатия переключателя. Поле **Время до выхода** показывает некорректное значение, так как для данной группы пользователей отключен автоматический выход из системы.

Сначала следует нажать кнопку **Конфигурация пользователей**:

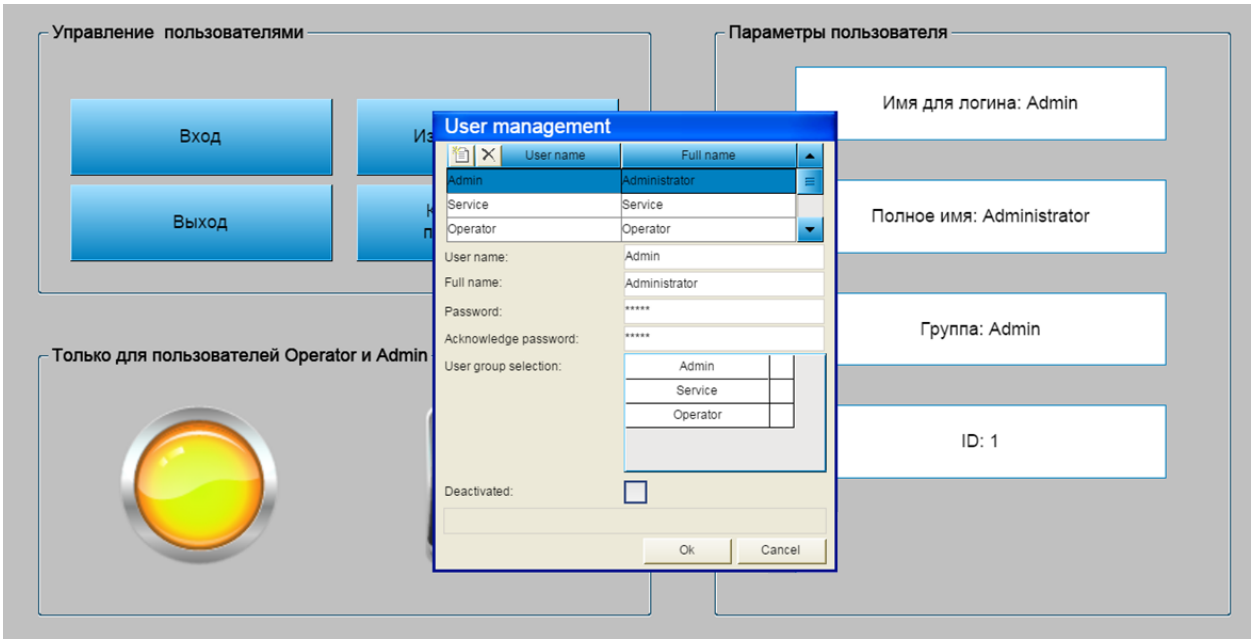


Рисунок 10.378 – Диалоговое окно изменения пароля

С помощью данного диалогового окна можно менять логины и пароли пользователей, удалять пользователей, создавать новых и блокировать существующих.

10.3.8 Менеджер рецептов

Данный пример посвящен работе с компонентом **Менеджер рецептов**.

Компонент **Менеджер рецептов** позволяет создавать списки переменных (**группы рецептов**) и составлять для них наборы значений (**рецепты**), которые будут присваиваться переменным соответствующей группы по команде пользователя.

В случае использования в своей программе **Менеджера рецептов** пользователю не требуется вникать в технические особенности работы с файлами (чтение/запись) – достаточно 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.

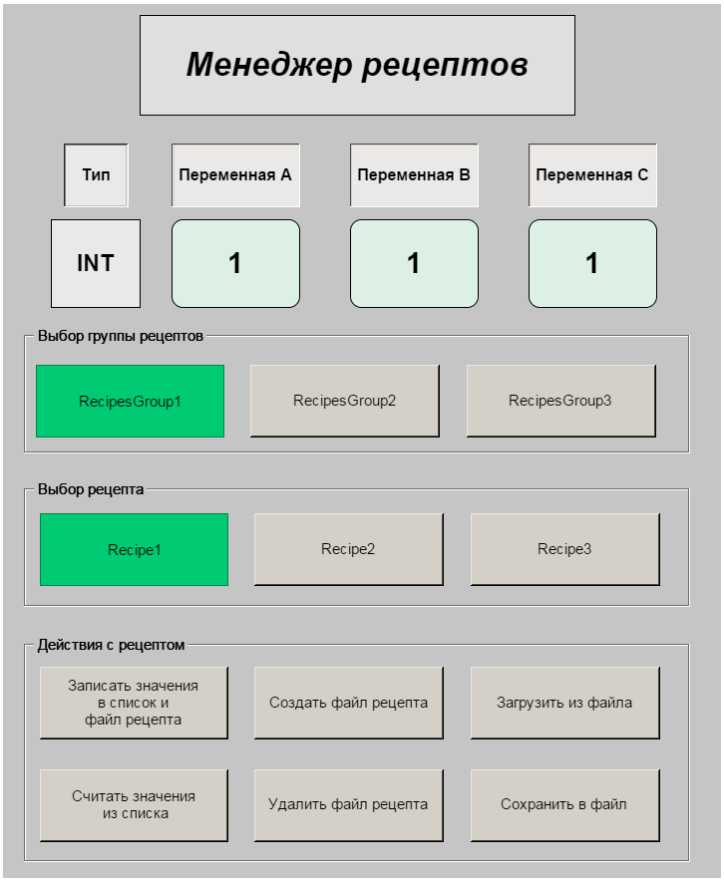


Рисунок 10.379 – Внешний вид примера Менеджер рецептов

Пример создан в среде **CODESYS V3.5 SP11 Patch 5** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с таргет-файлом **3.5.11.5**.

Пример доступен для скачивания: Example_RecipeManager.projectarchive

См. видеокурс по работе с рецептами.

Продвинутый пример от компании **CODEYS Group** по работе с библиотекой менеджера рецептов:

RecipeManagment.package

Файлы формата **.package** устанавливаются с помощью **Менеджера пакетов** (вкладка **Инструменты**) или (начиная с версии **CODESYS V3.5 SP17**) с помощью утилиты **CODESYS Installer**.

В качестве примера рассматривается создание проекта с использованием **Менеджера рецептов**.

В этом проекте пользователь будет оперировать тремя группами рецептов (**RecipesGroup1**, **RecipesGroup2**, **RecipesGroup3**), каждая из которых состоит из трех рецептов (**Recipe1**, **Recipe2**, **Recipe3**), определяемых наборами из трех переменных:

Таблица 10.4 – Структура примера проекта с использованием Менеджера рецептов

Группа рецептов	Тип переменных	Переменные группы	Рецепты		
			Recipe1	Recipe2	Recipe3
RecipesGroup1	INT	iA	1	2	3
		IB	1	2	3
		iC	1	2	3
RecipesGroup2	REAL	rA	1.111	2.222	3.333
		rB	1.111	2.222	3.333
		rC	1.111	2.222	3.333
RecipesGroup3	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 **RecipesGroup1** 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 **RecipesGroup1** 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 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***.

Для создания примера с использованием рецептов следует:

1. Создать новый ***стандартный*** проект **CODESYS** с названием **ExampleRecipeManager** и настройками по умолчанию: таргет – CODESYS Control Win V3, язык PLC_PRG – **ST**.
2. В программе **PLC_PRG** объявить следующие переменные:

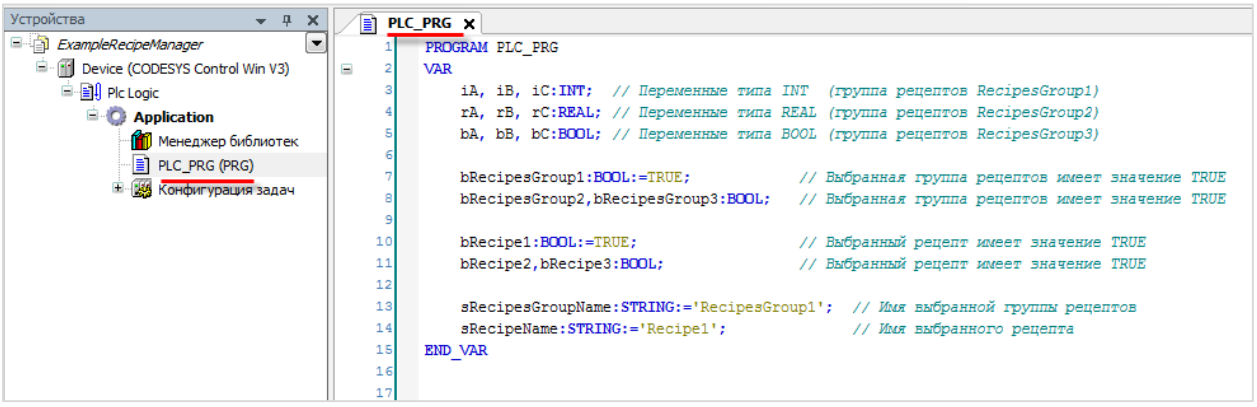


Рисунок 10.380 – Объявление переменных в программе PLC_PRG

3. Добавить компонент **Менеджер рецептов** (Recipe Manager):

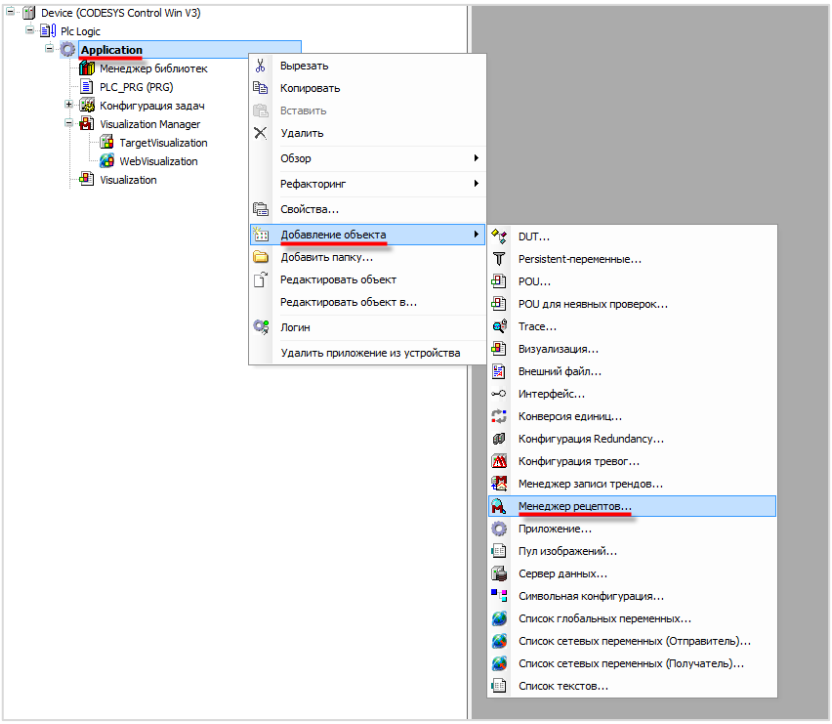


Рисунок 10.381 – Добавление в проект Менеджера рецептов

На вкладке **Сохранение** следует выбрать тип сохранения **Текстовый** (можно также сохранять файл в двоичном виде), указать путь к папке, в которой будут сохраняться рецепты и их расширение. Так как проект будет запущен на виртуальном контроллере, то в качестве места сохранения рецептов будет использоваться папка на **логическом диске D** компьютера, на котором запускается виртуальный контроллер.

ПРИМЕЧАНИЕ

Для контроллеров ОВЕН (и других целевых устройств с ОС **Linux)** должен быть указан корректный путь к папке рецептов (например, **/mnt/ufs/home**). Если путь не указан – то файлы рецептов будут сохранены в рабочей директории.

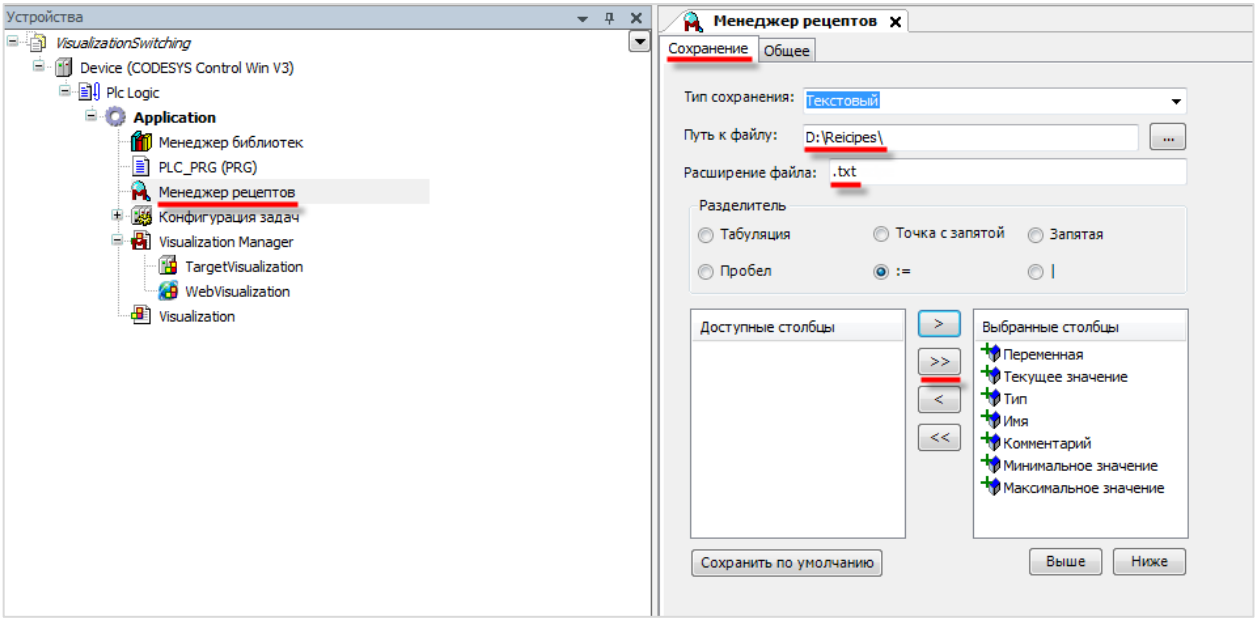


Рисунок 10.382 – Настройки Менеджера рецептов, вкладка Сохранение

Во вкладке **Общее** присутствует параметр **Управление рецептами**, который отвечает за возможность изменения рецептов в процессе работы проекта. Если установлена галочка **Сохранять изменения рецептов в файлы рецептов** автоматически, то будет происходить автоматическое изменение файлов рецептов в случае изменения их списков.

Опция **Загрузить рецепт** позволяет выбрать, что будет происходить, если загружаемый рецепт не полностью соответствует своему списку – либо все его значения не будут загружены и системная переменная **g_RecipeManager.LastError** примет значение **ERR_RECIPE_MISMATCH** (вариант **Загрузка при полном соответствии списку переменных**), либо будут загружены значения только тех параметров, имена которых соответствуют именам из списка (вариант **Загрузка переменных, подходящих по имени**).

Если установлена галочка **Overwrite existing recipes on download**, то при загрузке проекта все уже существующие в ПЛК файлы рецептов будут перезаписаны.

Опция **Записать рецепт** позволяет выбрать, что будет происходить, если записываемое из рецепта значение выходит за диапазон допустимых для параметра значений – либо оно будет ограничено минимальным или максимальных значения (вариант **Ограничить переменную**), либо сохранит свое последнее значение (вариант **Не записывать переменную**).

Если установлена галочка **Проверить рецепт на изменения**, то сохранение данных в рецепт будет произведено только в том случае, если значения его параметров изменились.

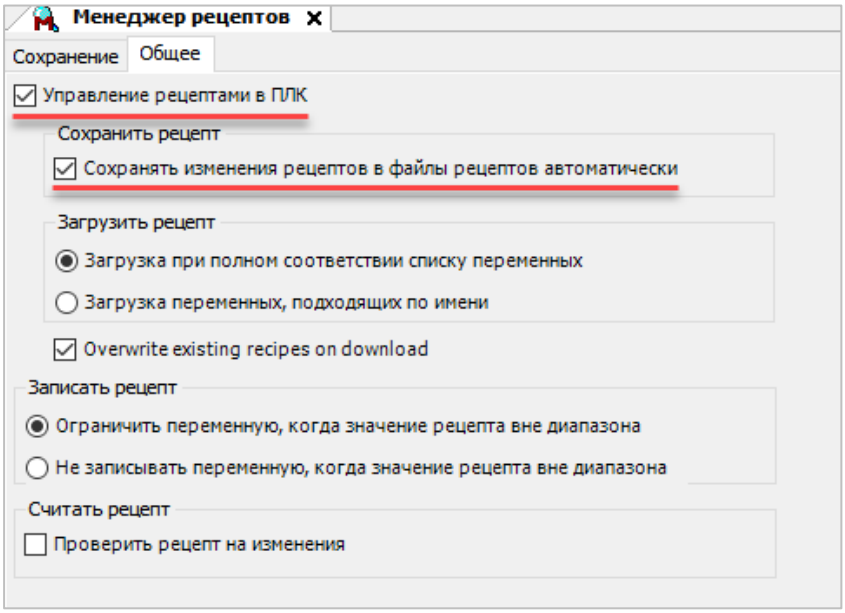


Рисунок 10.383 – Настройки Менеджера рецептов, вкладка Общее

Сначала следует создать три **группы рецептов** (определений списков) с названиями **RecipesGroup1**, **RecipesGroup2**, **RecipesGroup3**.

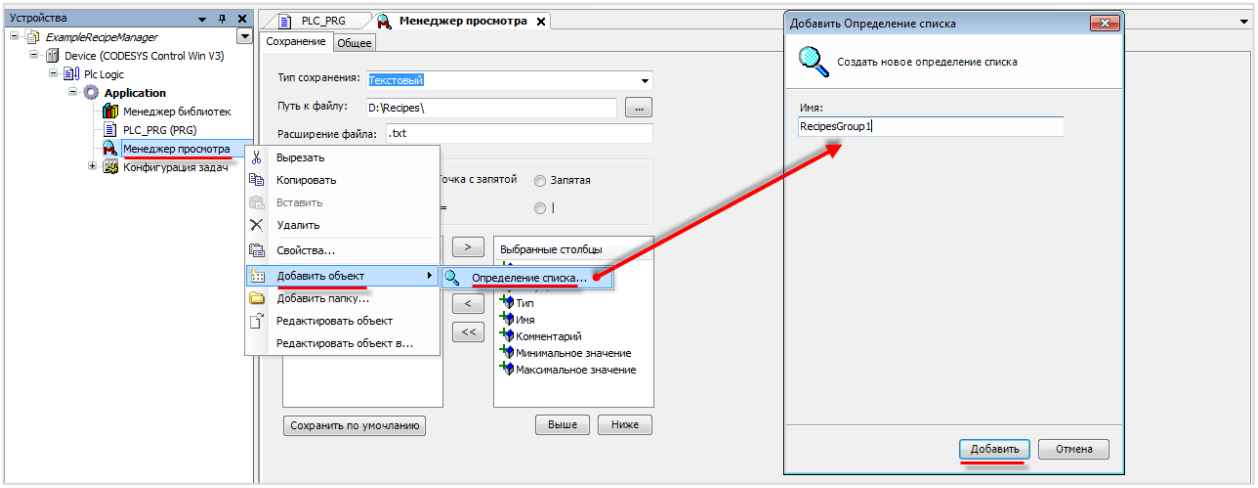


Рисунок 10.384 – Добавление группы рецептов

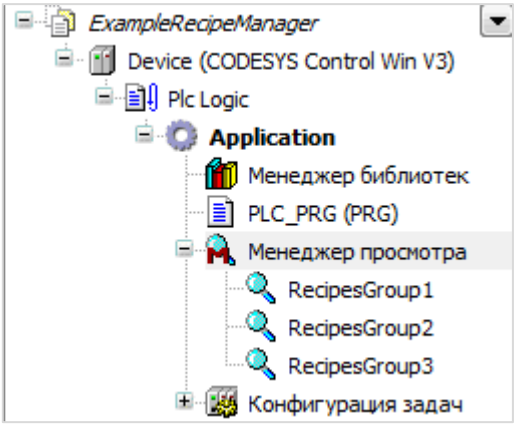


Рисунок 10.385 – Три группы рецептов на Панели устройств

В группе **RecipesGroup1** следует выбрать переменные **iA**, **iB**, **iC**, объявленные в программе **PLC_PRG**. В столбце **Имя** задать переменным имена **Переменная A**, **Переменная B**, **Переменная C**. Эти имена, как и комментарии, являются вспомогательной информацией и через них не производится обращение к переменным программы.

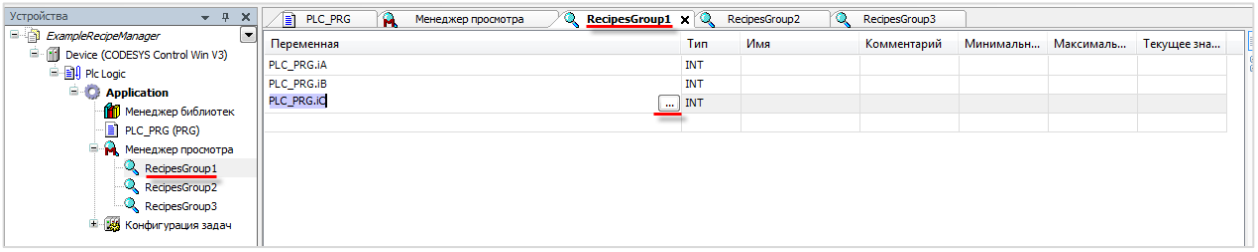


Рисунок 10.386 – Окно группы рецептов

Затем следует создать в группе рецептов **RecipesGroup1** три рецепта с названиями **Recipe1**, **Recipe2**, **Recipe3** и наполнить их значениями:

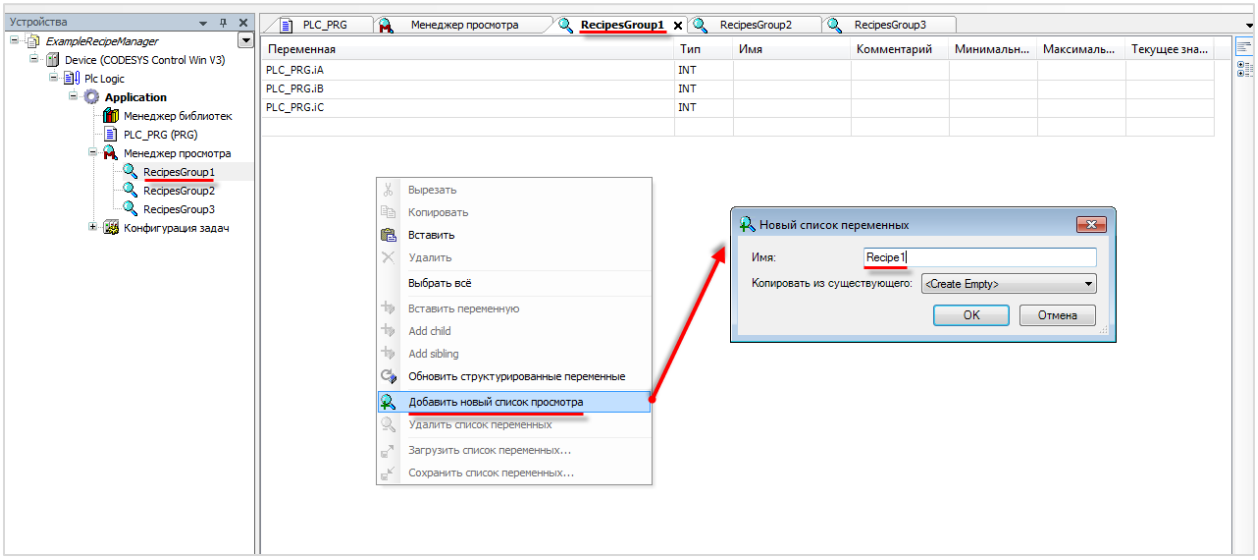


Рисунок 10.387 – Создание рецепта в группе рецептов

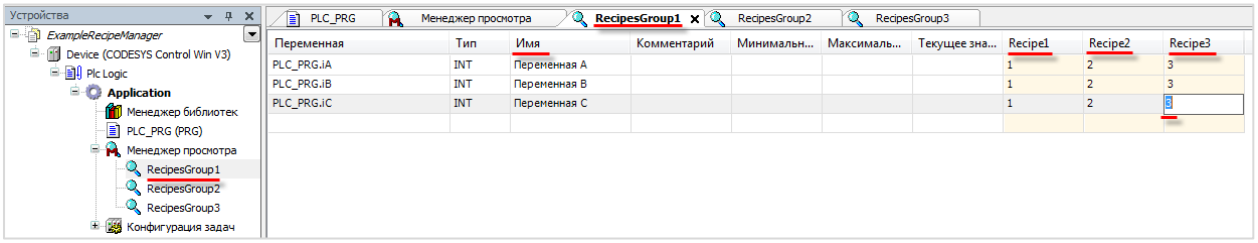


Рисунок 10.388 – Группа рецептов RecipesGroup1

По аналогии настраивается группа **RecipesGroup2** (три переменных типа **REAL**) и группа **RecipesGroup3** (три переменных типа **BOOL**):

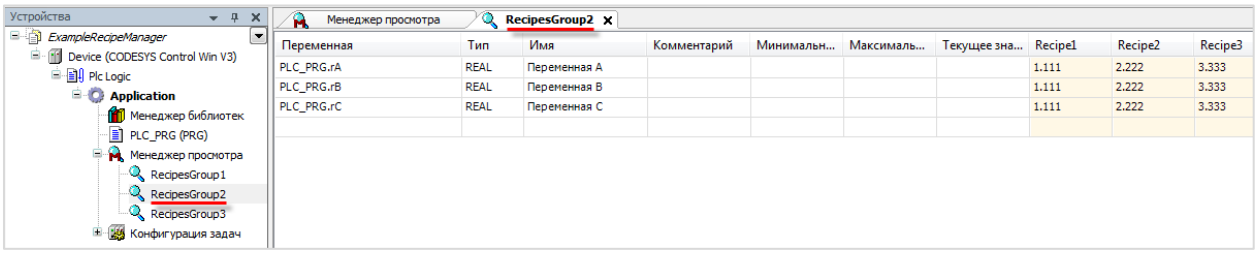


Рисунок 10.389 – Группа рецептов RecipesGroup2

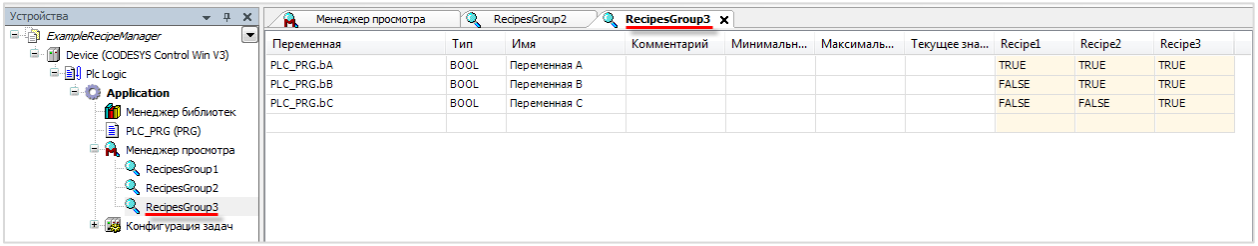


Рисунок 10.390 – Группа рецептов RecipesGroup3

4. Создание визуализации

4.1. Экран Visualization

Сначала в проект следует добавить экран визуализации с названием **Visualization**. В его свойствах выбрать размер **600 × 700**. Затем добавить 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.

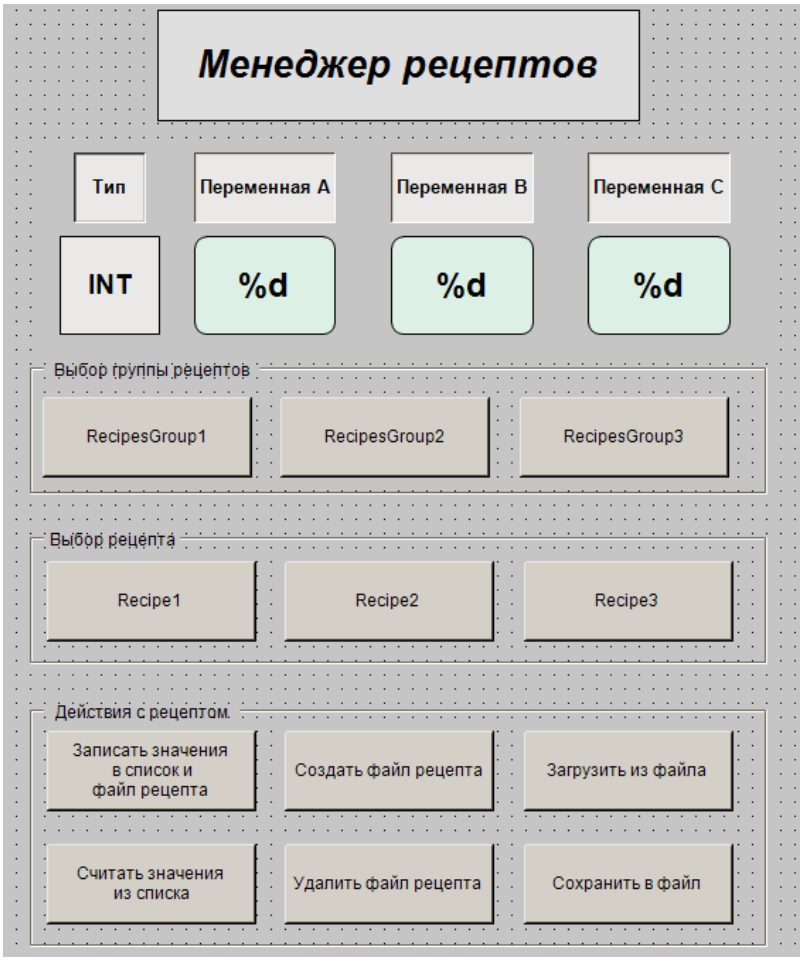


Рисунок 10.391 – Содержимое экрана визуализации Visualization

4.2. Экран Frame1

Для отображения в элементе Фрейм экрана **Visualization** значений переменных трех групп рецептов, следует создать еще три экрана визуализации с названиями **Frame1**, **Frame2**, **Frame3**. В их свойствах выбрать размер **520 × 150**.

ПРИМЕЧАНИЕ
Число фреймов совпадает с количеством отображаемых групп рецептов.

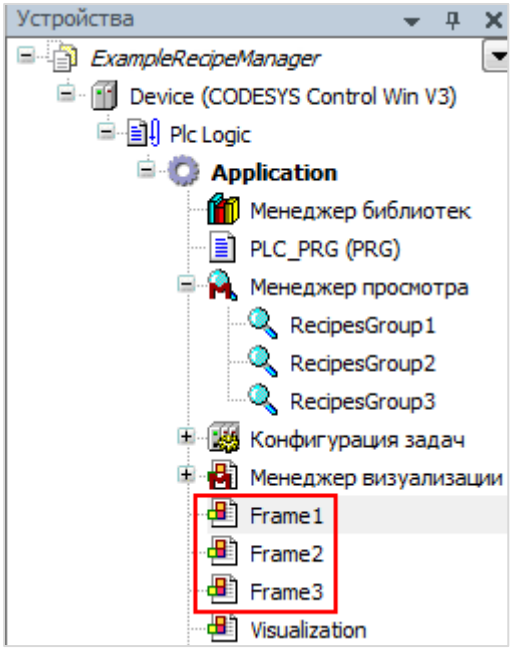


Рисунок 10.392 – Экраны фрейма на Панели устройств

Затем следует добавить на экран **Frame1** элементы типа Текстовое поле (верхний ряд) и Прямоугольник/Скругленный прямоугольник (нижний ряд). Последние будут использоваться для отображения значений переменных рецептов:

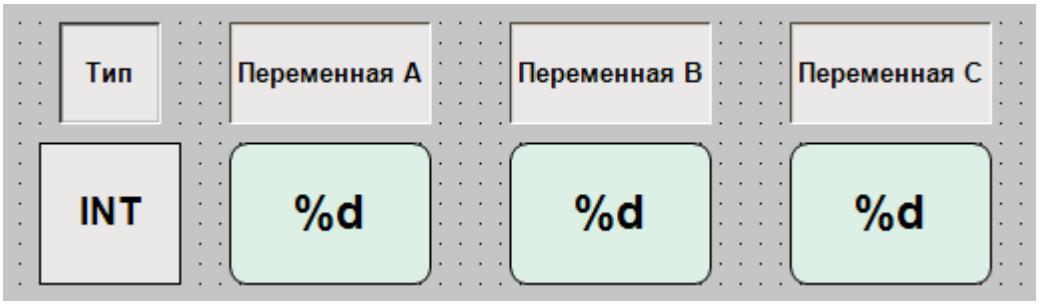


Рисунок 10.393 – Экран визуализации Frame1

К полю вывода **переменной A** привязывается переменная **PLC_PRG.iA** (параметр **Текстовая переменная**) и задается ей спецификатор формата вывода **%d**, используемый для отображения целочисленных значений (параметр **Текст**). Во вкладке InputConfiguration следует выбрать параметр **OnMouseClick** и привязать к нему действие Записать переменную, что позволит изменять значения переменных в процессе работы проекта.

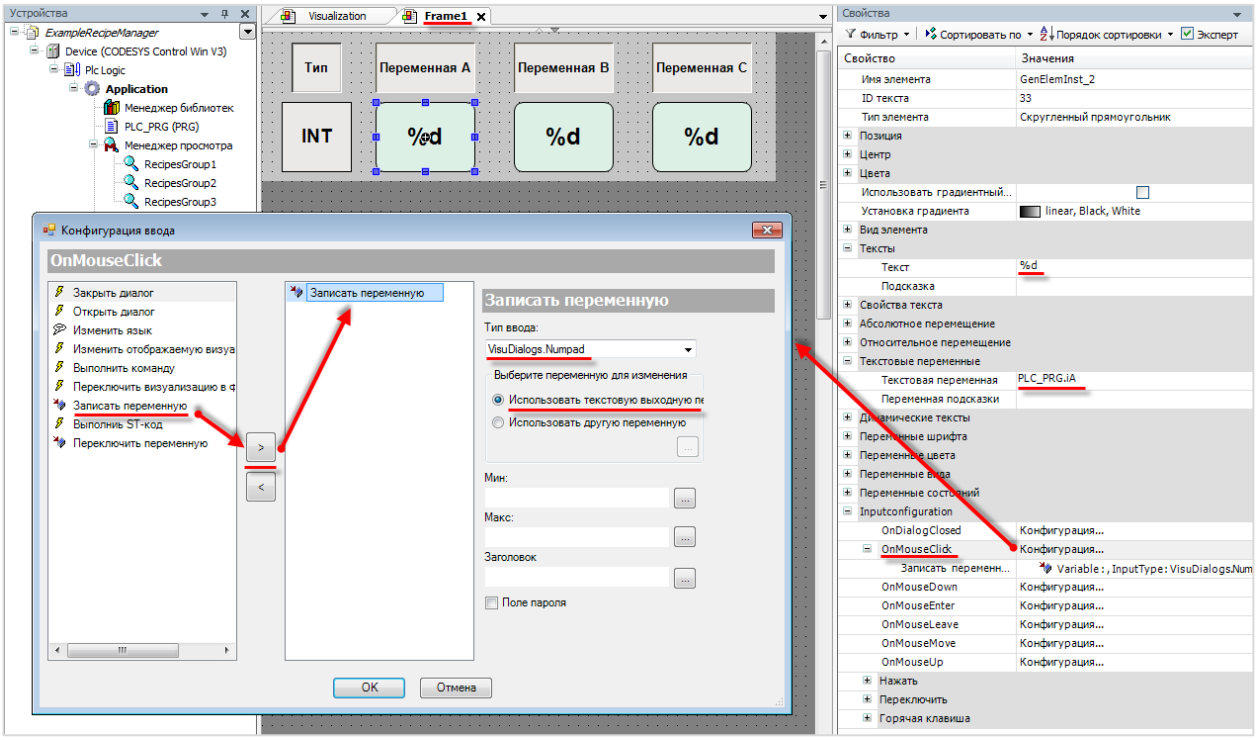


Рисунок 10.394 – Настройка полей вывода переменных типа INT

Аналогично настраиваются поля вывода **переменной B** и **C**. Поля будут отличаться только привязываемыми к элементам переменными – **PLC_PRG.iB** и **PLC_PRG.iC** соответственно.

4.3. Экраны Frame2 и Frame3

Аналогично предыдущему пункту настраиваются экраны **Frame2** и **Frame3**. Так как их содержимое аналогично экрану **Frame1**, можно перенести все элементы с экрана на экран с помощью команд **Копировать/Вставить**.

К экрану **Frame2** привязываются переменные типа REAL **PLC_PRG.rA**, **PLC_PRG.rB** и **PLC_PRG.rC** с форматом вывода **%.3f** (три знака после запятой), 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**).

Так как переменные типа **BOOL** принимают значения **TRUE/FALSE**, то в настройках их записи выбирается диалог **VisuDialogs.Keypad**, который представляет собой экранную клавиатуру.

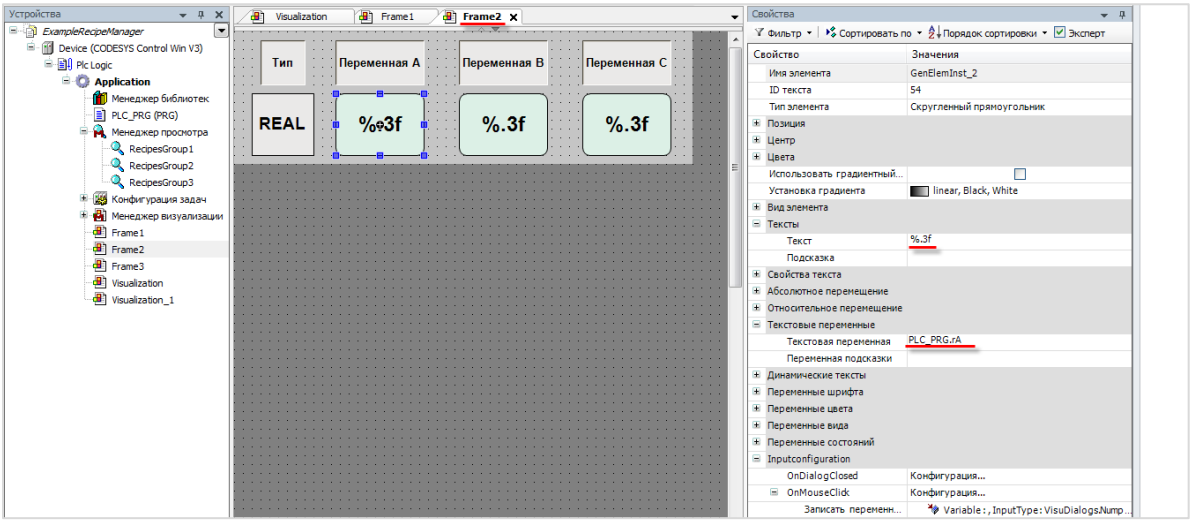


Рисунок 10.395 – Настройка полей вывода переменных типа REAL

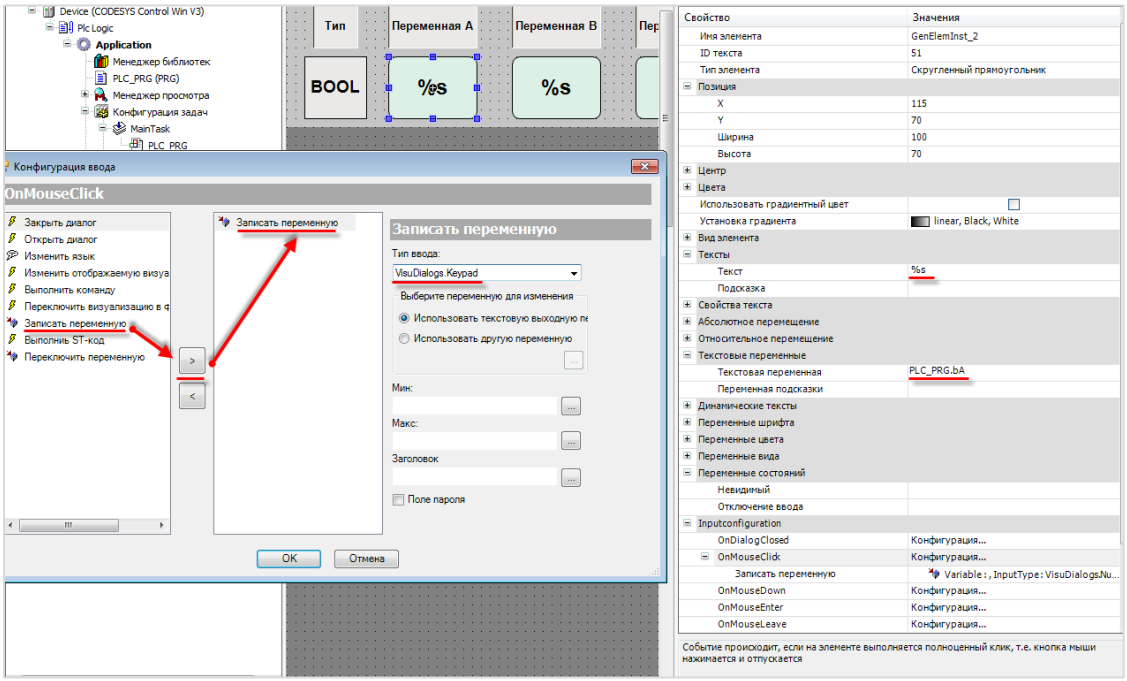


Рисунок 10.396 – Настройка полей вывода переменных типа BOOL

5. Экран Visualization (настройка действий

5.1. Настройка элемента Фрейм

После создания и настройки экранов фрейма следует вернуться на экран **Visualization**. В конфигурации элемента Фрейм выбрать экраны **Frame1**, **Frame2**, **Frame3**.

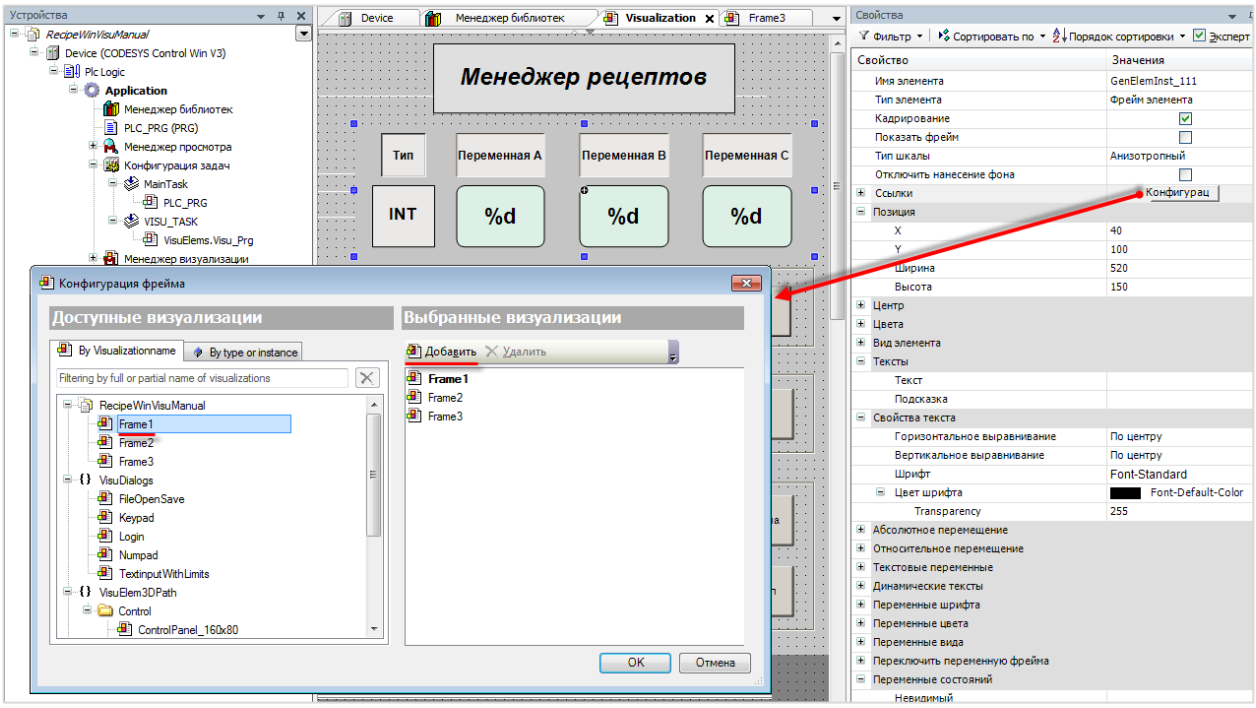


Рисунок 10.397 – Настройка содержимого элемента Фрейм

5.2. Настройка кнопок выбора групп рецептов

Далее следует настроить кнопку выбора группы рецептов **RecipesGroup1**. Выбирается цвет тревоги (такой цвет кнопка будет принимать в случае выбора привязанной к ней группы рецептов), задается ее текст, привязывается к параметрам **Переключить цвет** и **Двоичная переменная** переменную **PLC_PRG.bRecipesGroup1**. В случае нажатия на кнопку, переменные будут принимать значение **TRUE**, что приведет к изменению цвета кнопки и отображению ее нажатия.

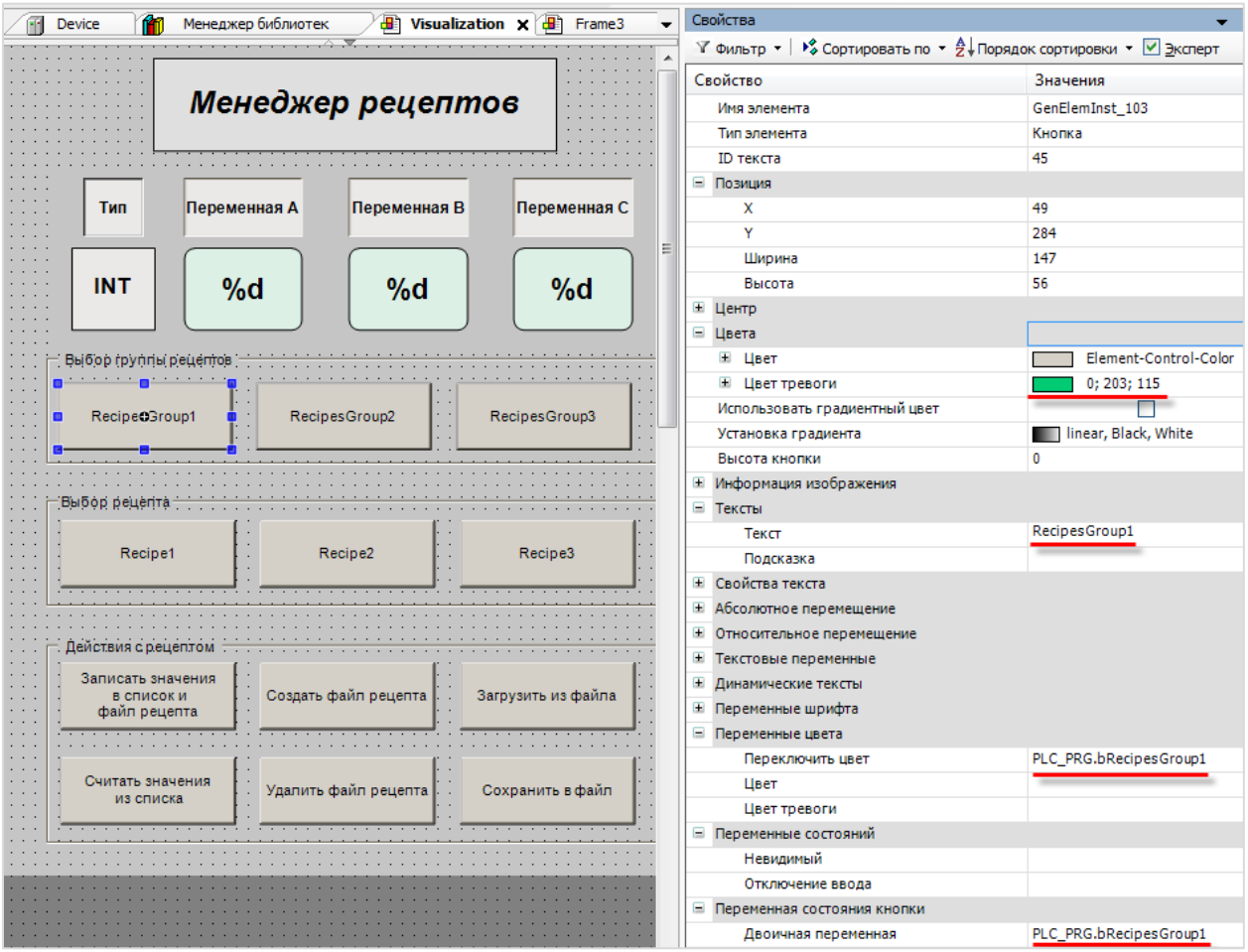


Рисунок 10.398 – Настройки кнопки выбора групп рецептов RecipesGroup1

Во вкладке InputConfiguration кнопки **RecipesGroup1** следует выбрать параметр **OnMouseClick** и привязать к нему два действия: Выполнить ST-код и Переключить визуализацию во фрейме. Для первого действия добавить код на языке **ST** (Рисунок 10.399), для второго – **Frame1 (**Рисунок 10.400).

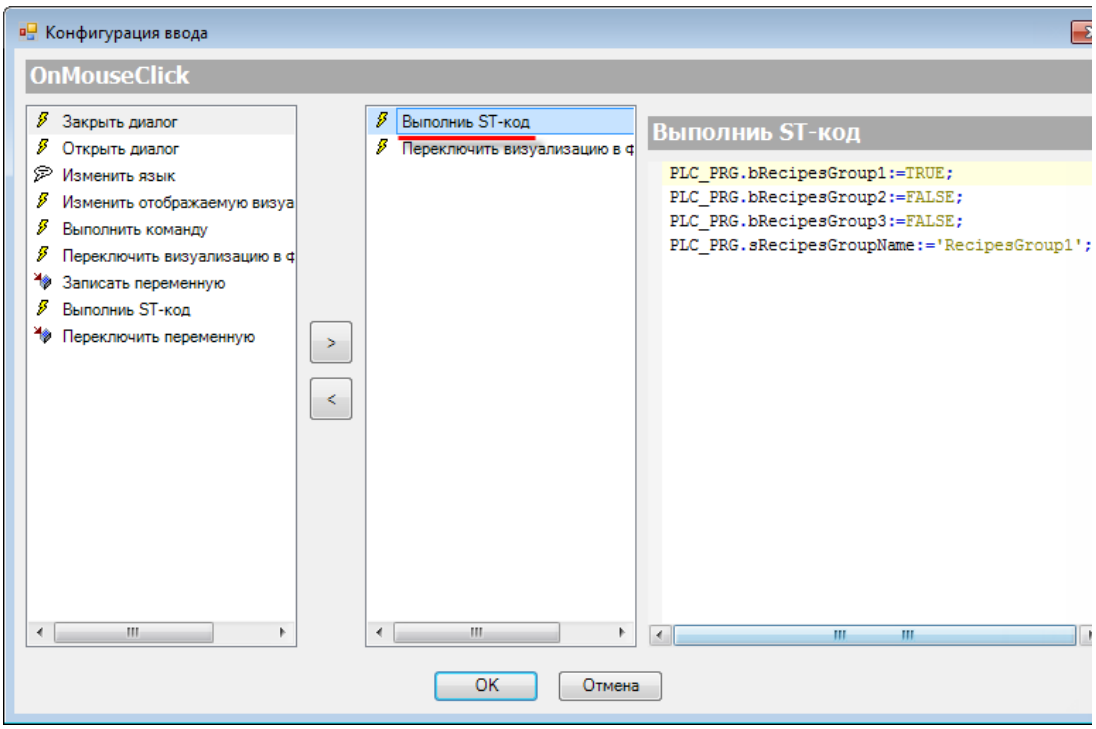


Рисунок 10.399 – Настройка параметра OnMouseClick кнопки RecipesGroup1, действие Выполнить ST-код

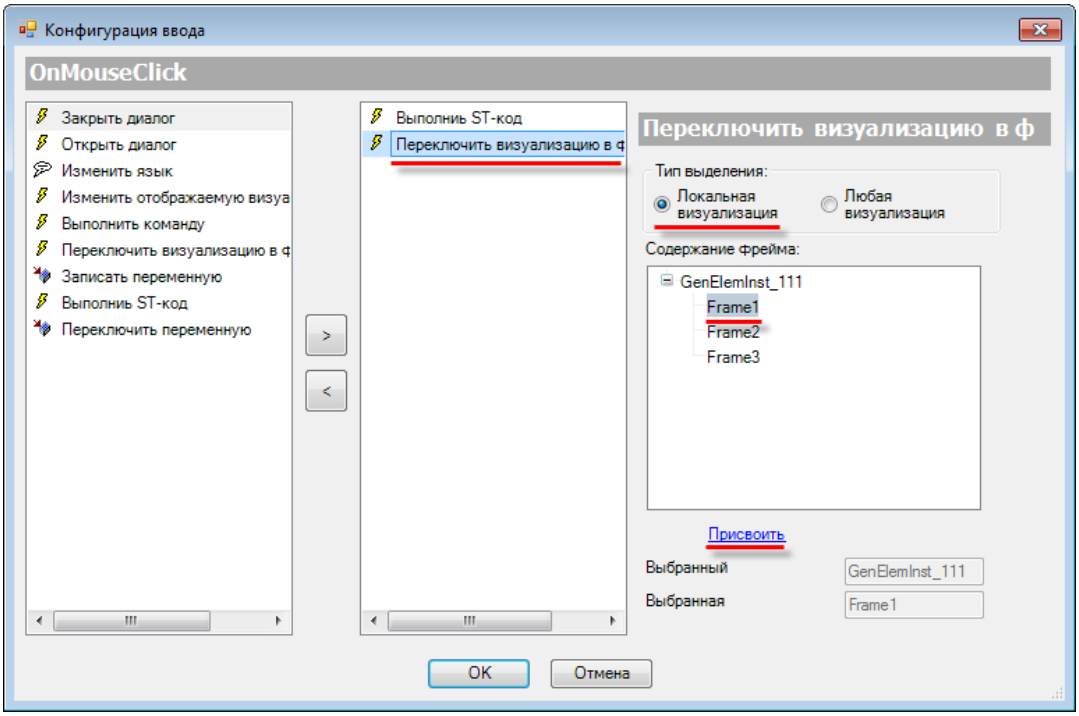


Рисунок 10.400 – Настройка параметра OnMouseClick кнопки RecipesGroup1, действие Переключить визуализацию во фрейме

В случае нажатия на кнопку выбора **RecipesGropup1** будут «отжиматься» все остальные кнопки выбора групп, в переменную **PLC_PRG.sRecipesGroupName** – записываться имя данной группы, во фрейме будут отображаться поля вывода ее переменных.

Аналогично настраиваются кнопки выбора группы **RecipesGroup2** и **RecipesGroup3**. ST-код для них приведен на рисунках 10.401 и 10.402:

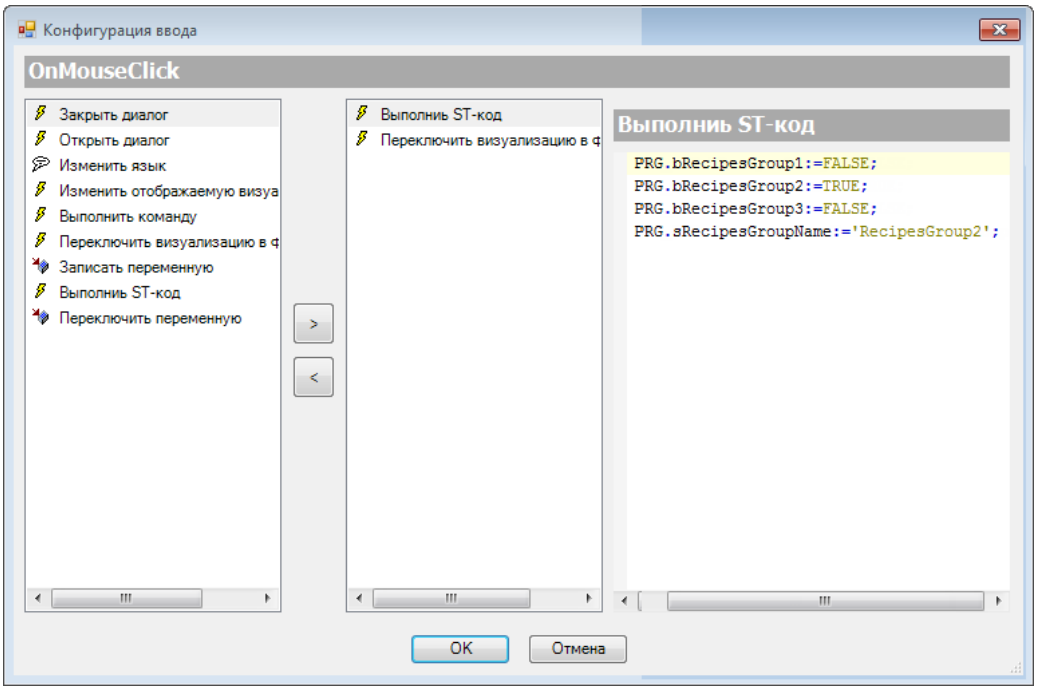


Рисунок 10.401 – Настройка параметра OnMouseClick кнопки RecipesGroup2, действие Выполнить ST-код

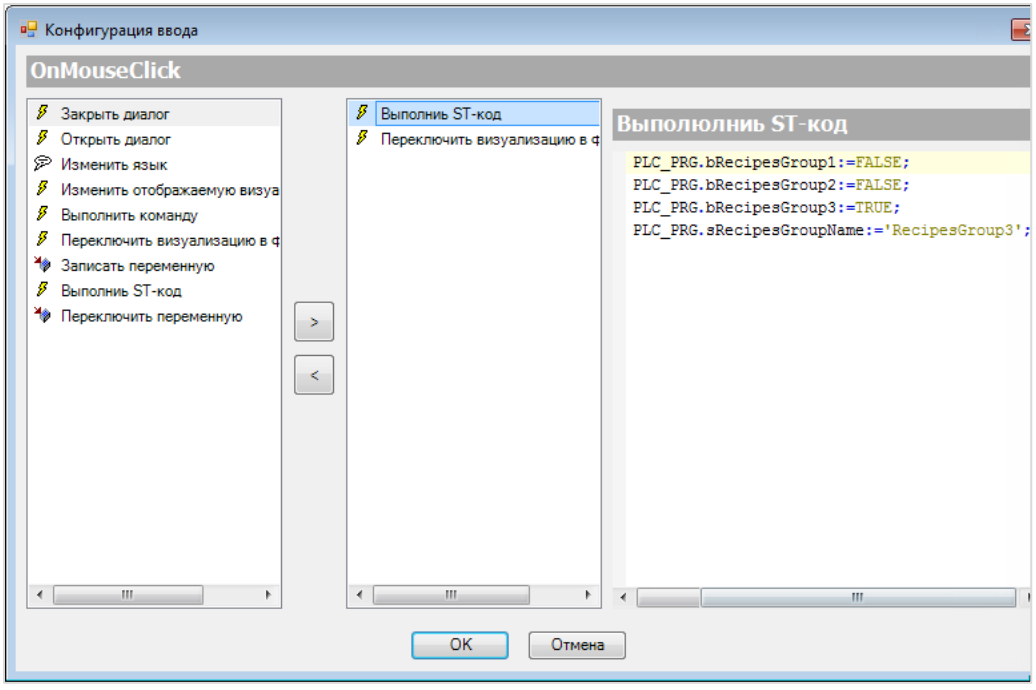


Рисунок 10.402 – Настройка параметра OnMouseClick кнопки RecipesGroup3, действие Выполнить ST-код

5.3. Настройка кнопок выбора рецептов

Сначала настраивается кнопка выбора рецепта **Recipe1**. Далее следует выбрать цвет тревоги (такой цвет кнопка будет принимать в случае выбора привязанного к ней рецепта), задается ее текст, привязывается к параметрам **Переключить цвет** и **Двоичная переменная** переменную **PLC_PRG.bRecipe1**. В случае нажатия на кнопку, переменные будут принимать значение **TRUE**, что приведет к изменению цвета кнопки и отображению ее нажатой.

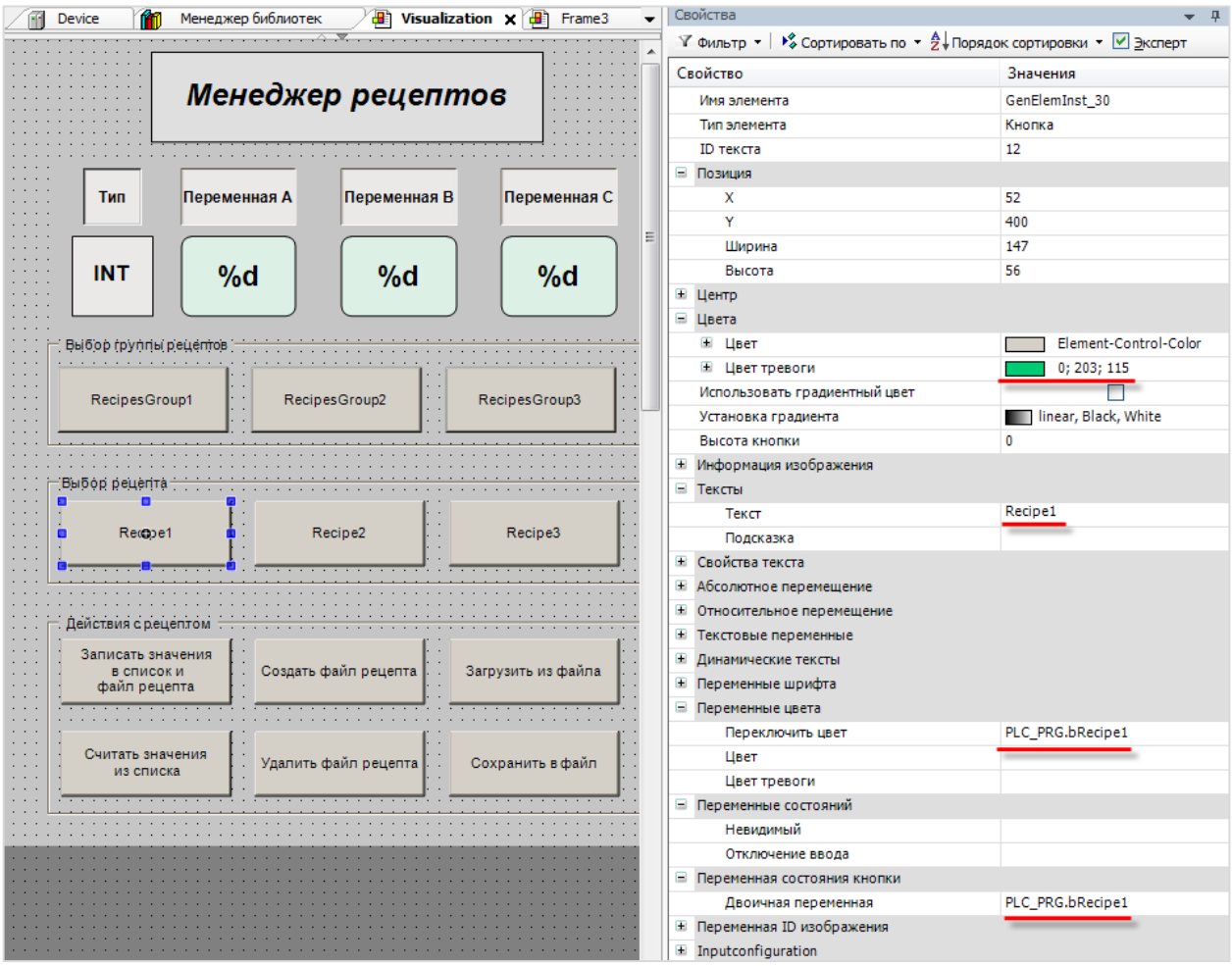


Рисунок 10.403 – Настройки кнопки выбора рецепта Recipe1

Во вкладке InputConfiguration кнопки **Recipe1** следует выбрать параметр **OnMouseClick** и привязать к нему действие Выполнить ST-код:

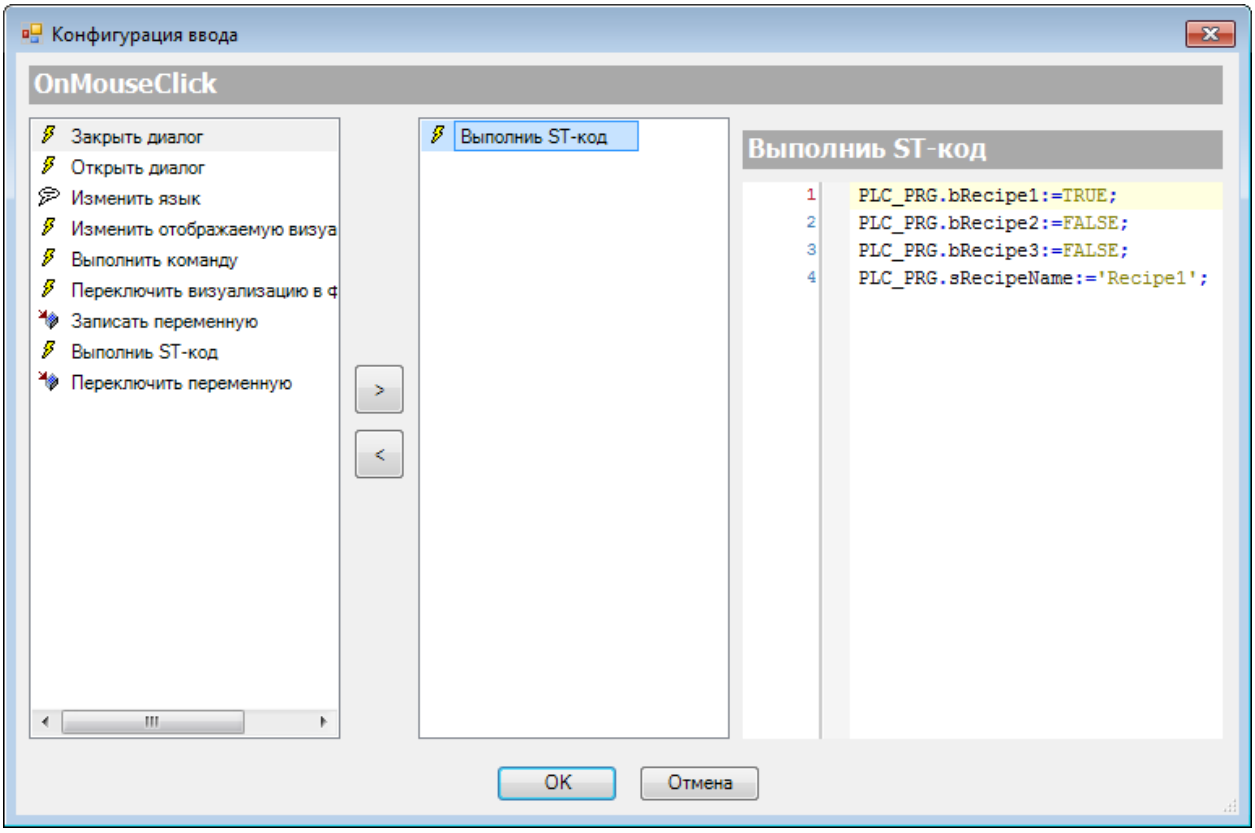


Рисунок 10.404 – Настройка параметра OnMouseClick кнопки Recipe1, действие Выполнить ST-код

Соответственно, по нажатию кнопки выбора **Recipe1** будут «отжиматься» все остальные кнопки выбора рецептов, в переменную **PLC_PRG.sRecipeName** записываться имя данного рецепта.

Аналогично настраиваются кнопки выбора рецептов **Recipe2** и **Recipe3**. ST-код для них приведен на рисунках 10.405 и 10.406:

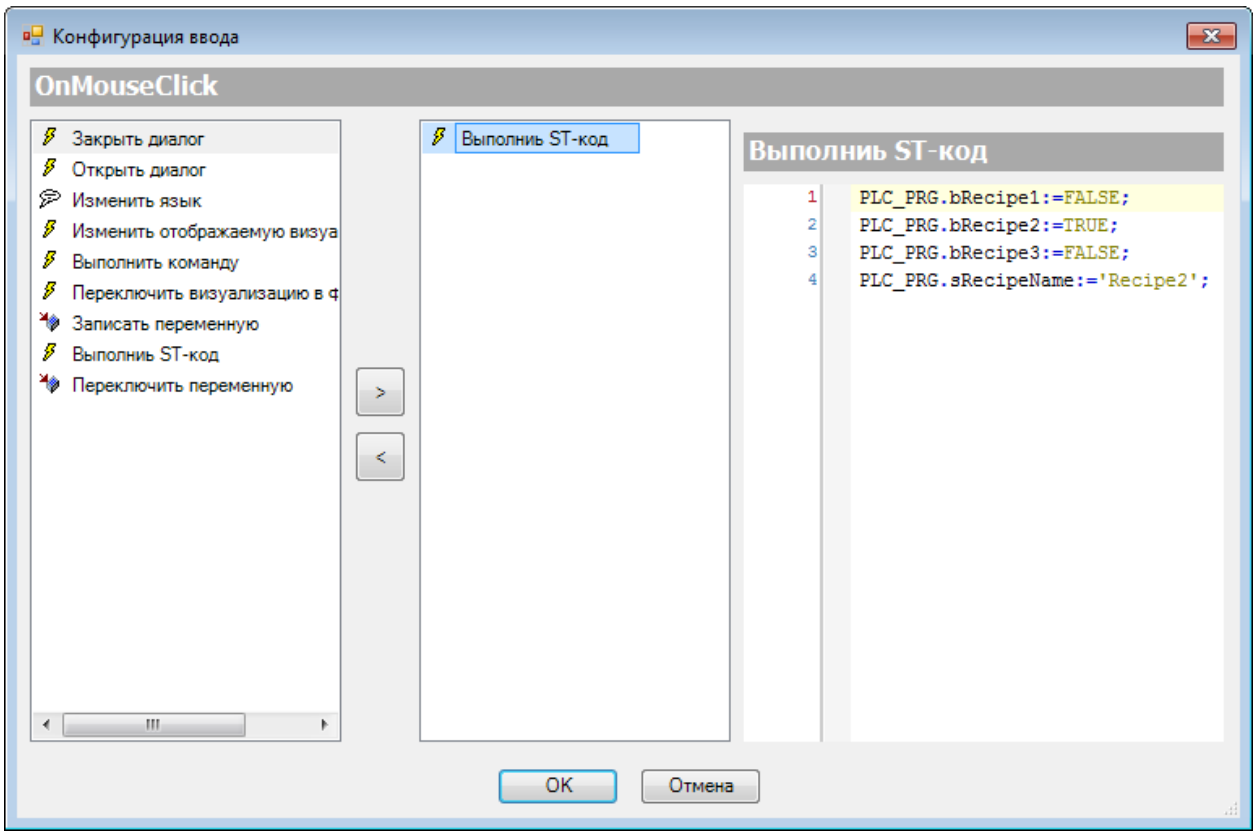


Рисунок 10.405 – Настройка параметра OnMouseClick кнопки Recipe2, действие Выполнить ST-код

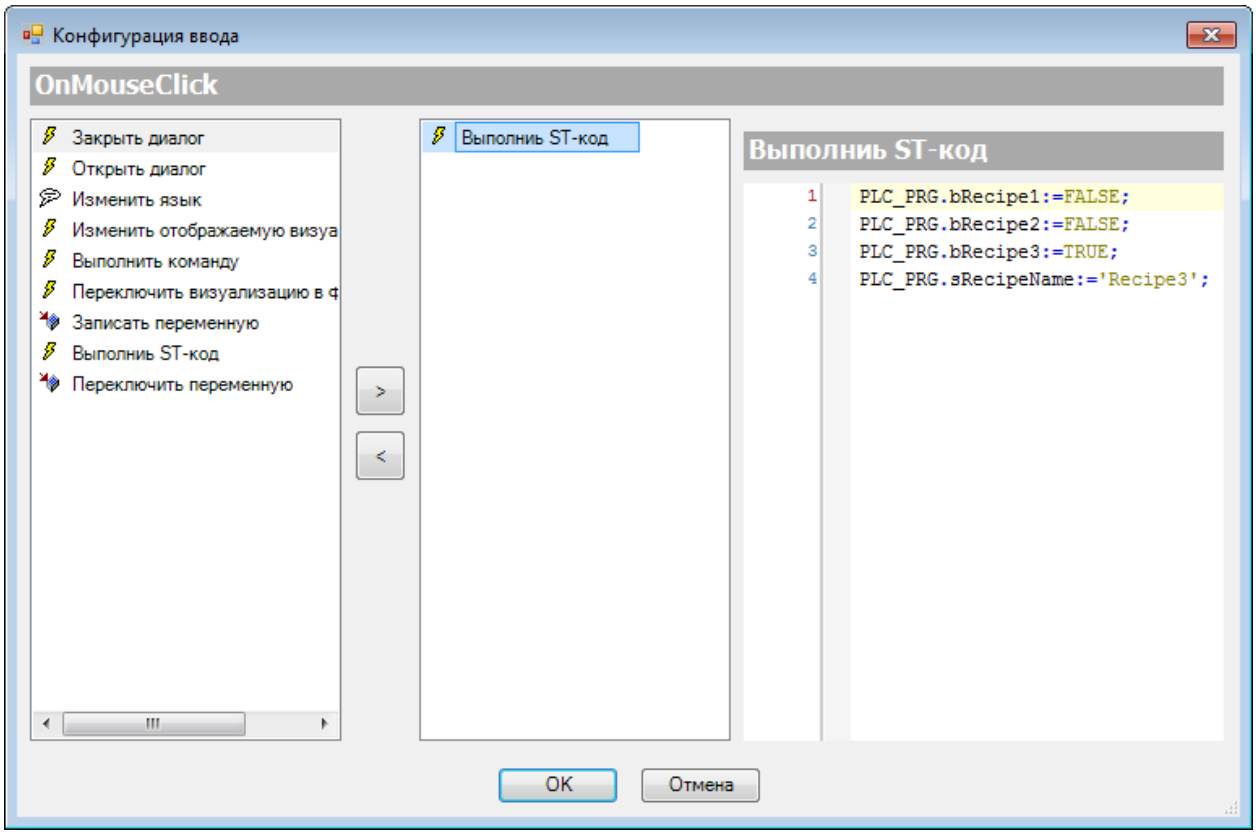


Рисунок 10.406 – Настройка параметра OnMouseClick кнопки Recipe3, действие Выполнить ST-код

5.4. Настройка кнопок управления рецептами

Сначала настраиваются кнопки управления рецептами. К этим кнопкам не привязывается никаких переменных, поэтому настройка ограничивается выбором на вкладке InputConfiguration параметра **OnMouseClick** и привязкой к нему действия Выполнить команду.

Для кнопки **Записать значения в список и файл рецепта** следует выбрать команду **ReadRecipe** и добавить ее с помощью кнопки «**+**». Не следует понимать англоязычное название команды буквально – она читает текущие значения переменных программы и записывает их в список рецепта, и с помощью установленной галочке во вкладке **Общее** (Рисунок 10.386) после этого автоматически обновляется и файл рецепта.

Команда **WriteRecipe** записывает значения рецепта в переменные программы, поэтому её следует привязать 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 данном примере это переменные **PLC_PRG.sRecipesGroupName** и **PLC_PRG.sRecipeName**.

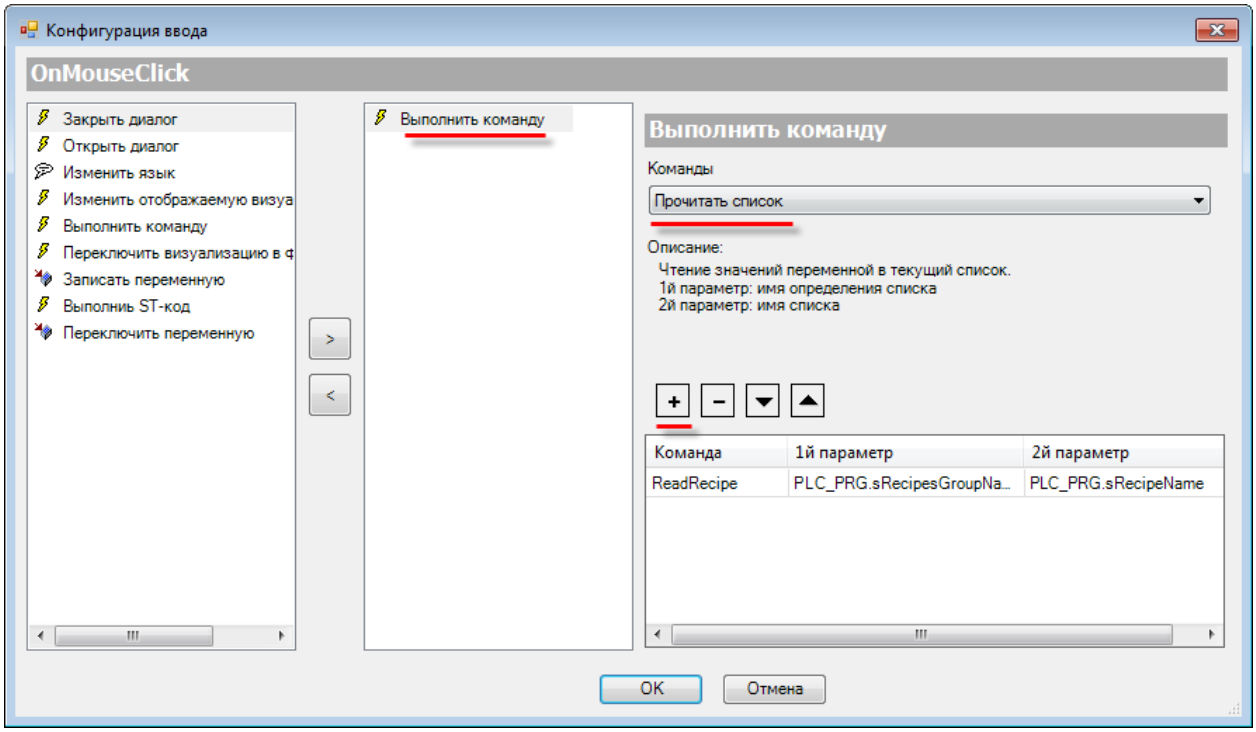


Рисунок 10.407 – Настройка параметра OnMouseClick кнопки Записать значения в список и файл рецепта, команда Прочитать список (ReadRecipe)

Аналогично настраиваются остальные кнопки (отличие будет только в выбираемых командах).

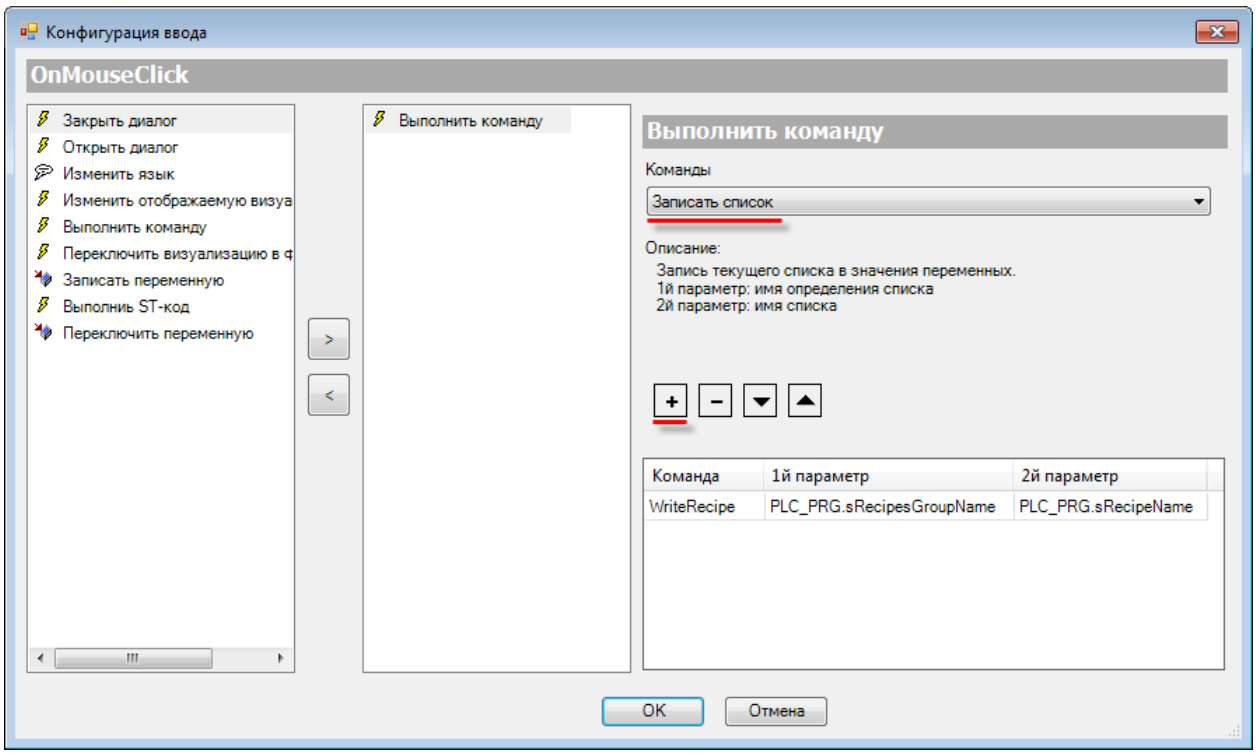


Рисунок 10.408 – Настройка параметра OnMouseClick кнопки Считать значения из списка, команда Записать список (WriteRecipe)

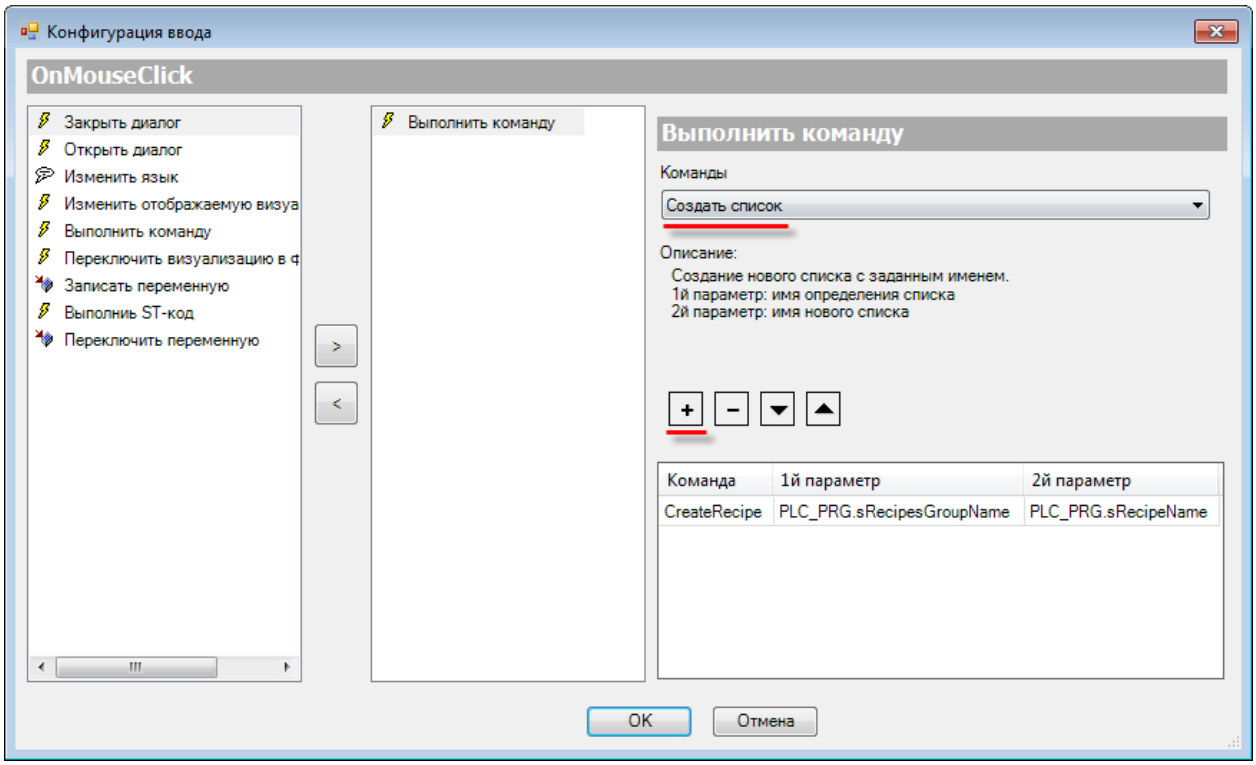


Рисунок 10.409 – Настройка параметра OnMouseClick кнопки Создать рецепт, команда Создать список (CreateRecipe)

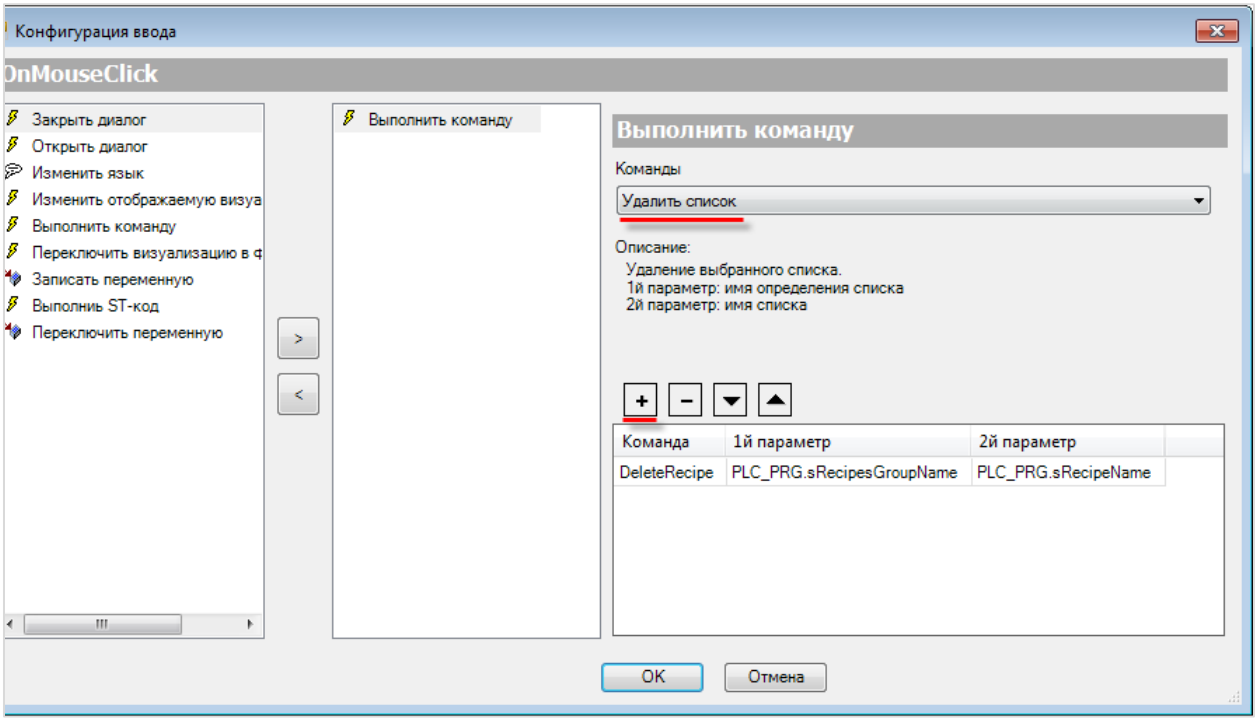


Рисунок 10.410 – Настройка параметра OnMouseClick кнопки Удалить рецепт, команда Удалить список (DeleteRecipe)

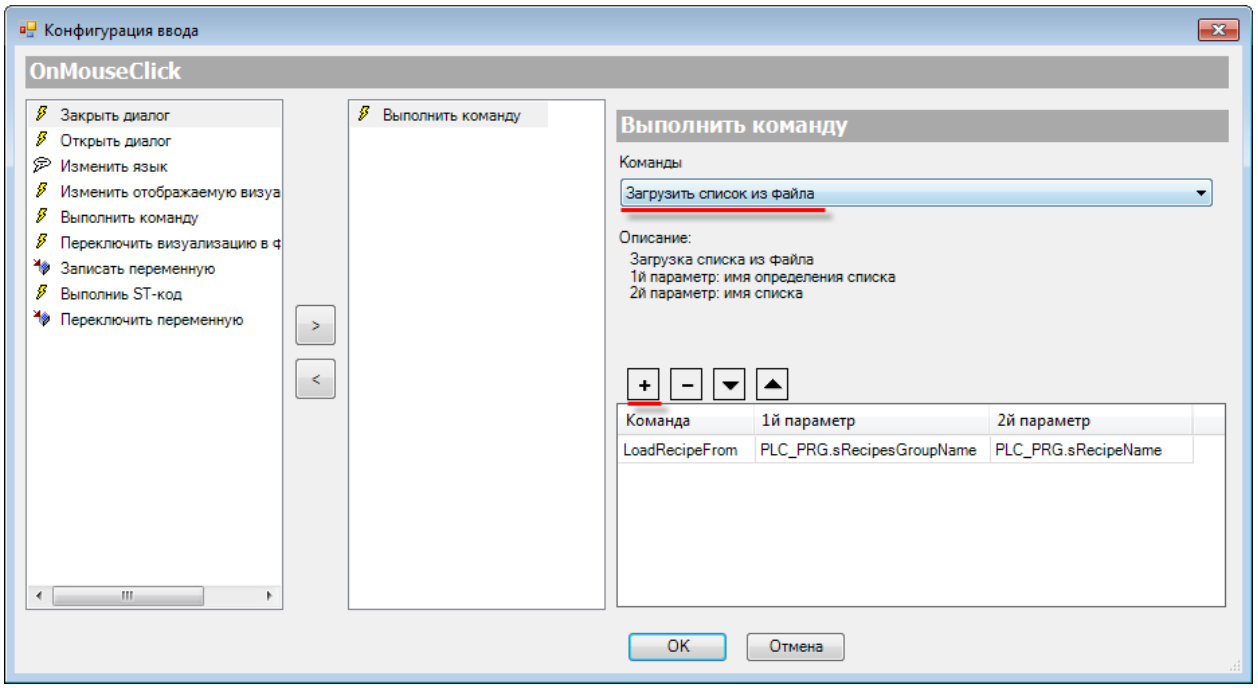


Рисунок 10.411 – Настройка параметра OnMouseClick кнопки Загрузить из файла, команда Загрузить список из файла (LoadRecipeFrom)

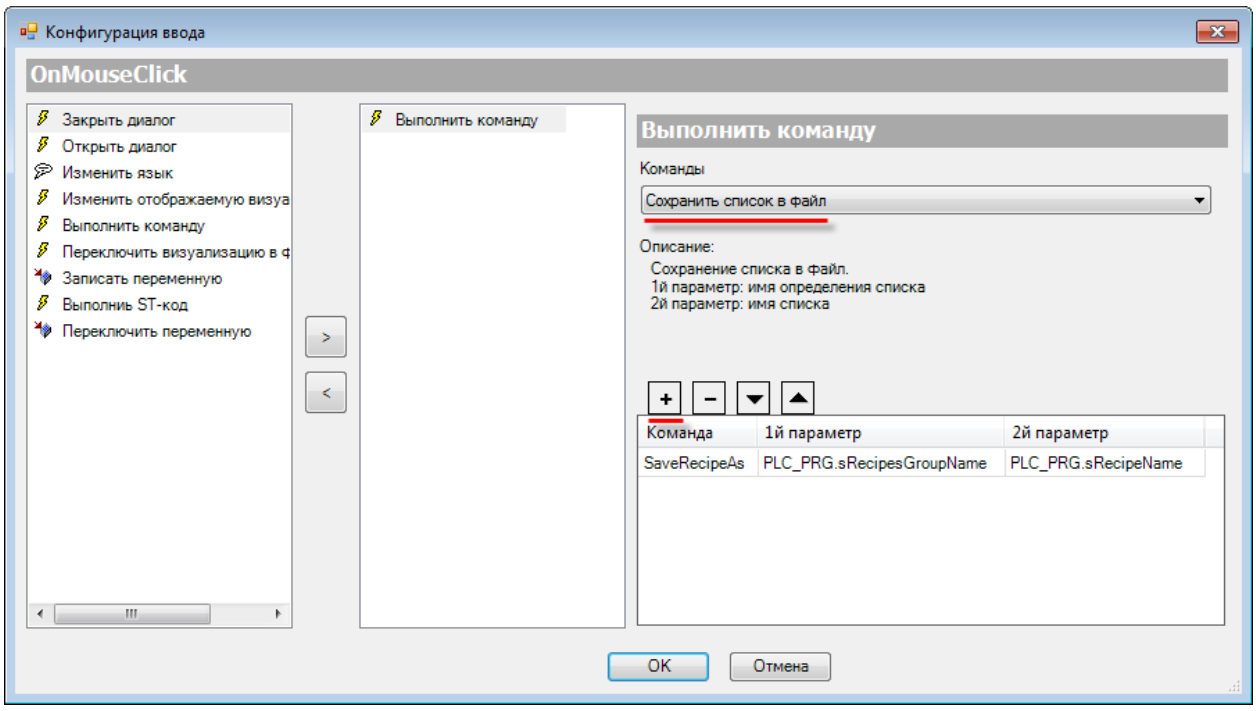
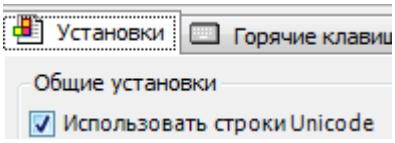


Рисунок 10.412 – Настройка параметра OnMouseClick кнопки Сохранить в файл, команда Сохранить список в файл (SaveRecipeAs)

6. Настроить Менеджер визуализации. В его установках следует поставить галочку **Использовать строки Unicode**:



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

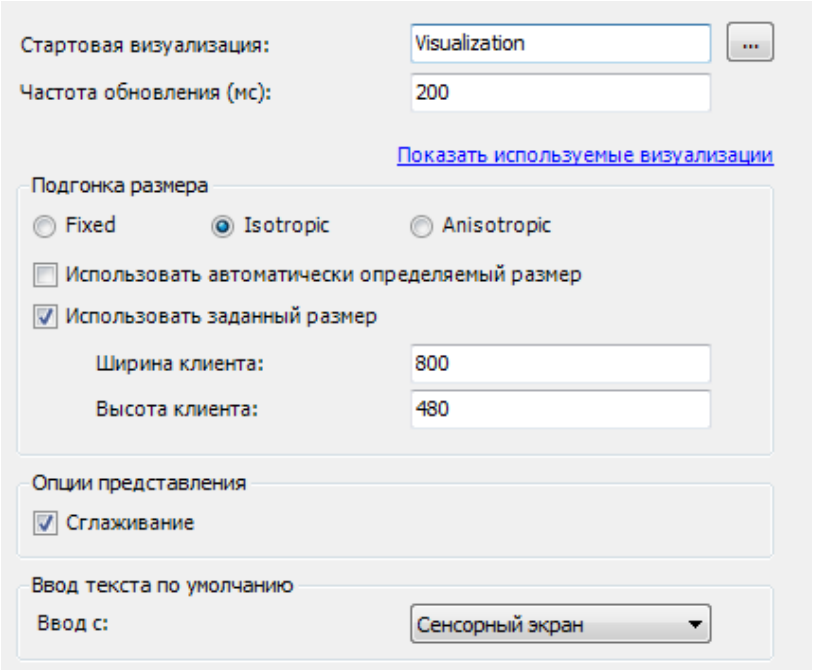


Рисунок 10.413 – Настройки таргет-визуализации

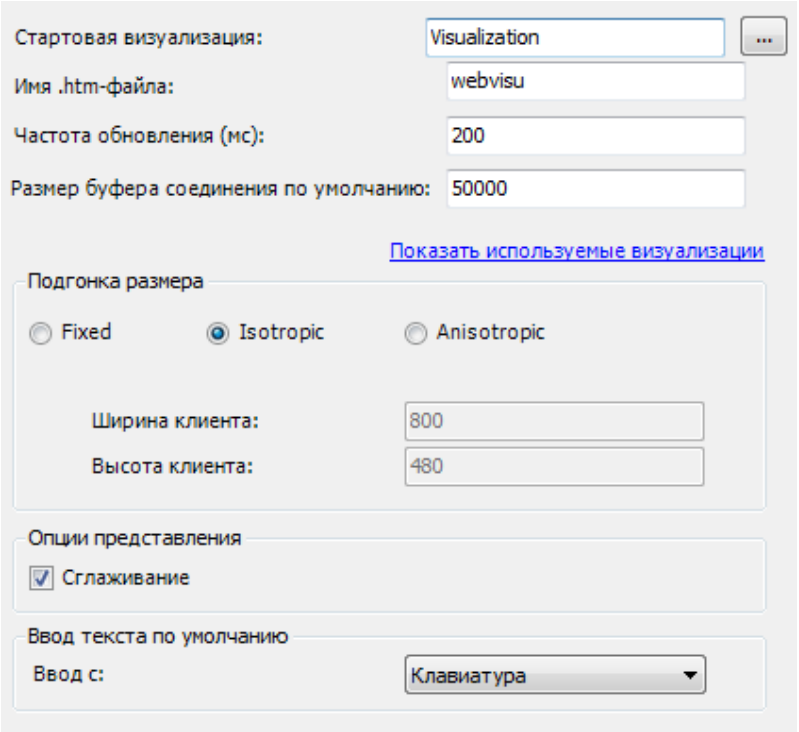


Рисунок 10.414 – Настройки web-визуализации

7. Запустить проект на виртуальном контроллере. Открыть в веб-браузере страницу http://localhost:8080/webvisu.htm, чтобы увидеть web-визуализацию проекта. ***Следует обратить внимание***, что браузер должен поддерживать **HTML5**. Проверить функционал проекта.

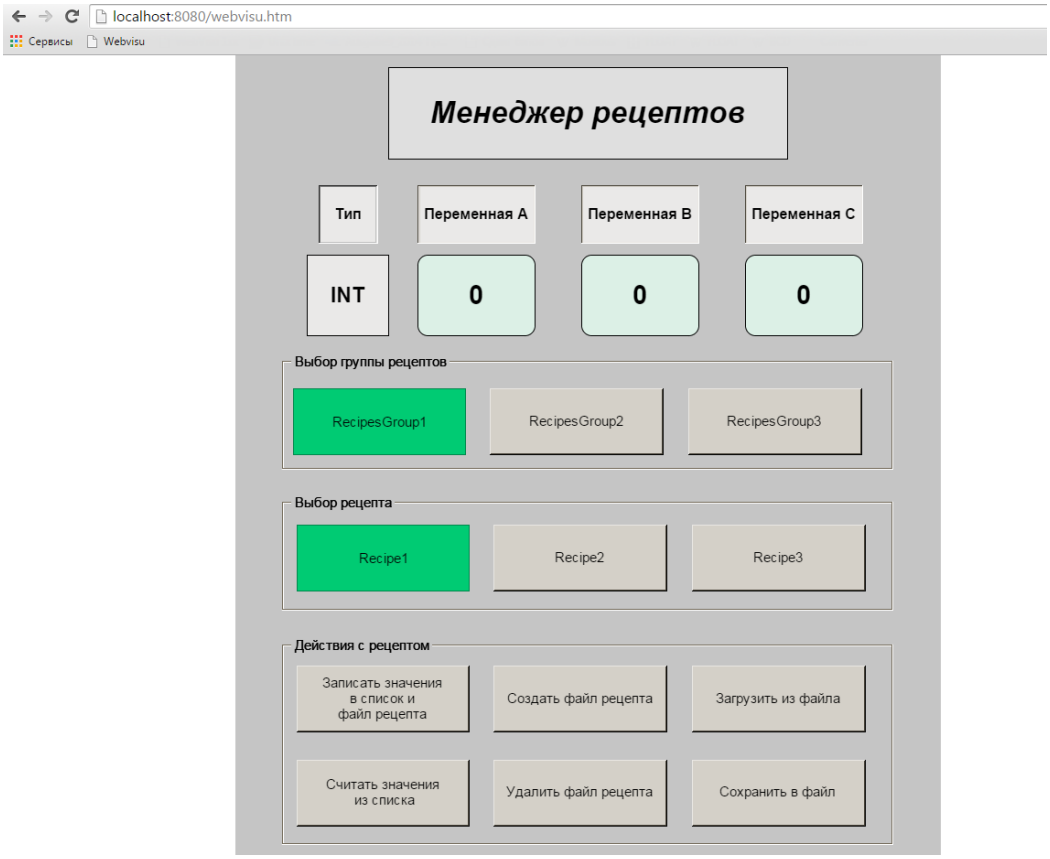


Рисунок 10.415 – Web-визуализация проекта в браузере Chrome

Следует обратить внимание, что после запуска проекта, в папке **D:\recipes** (указанной в **Менеджере рецептов**, см. рисунок 10.385) были созданы **файлы рецептов**.

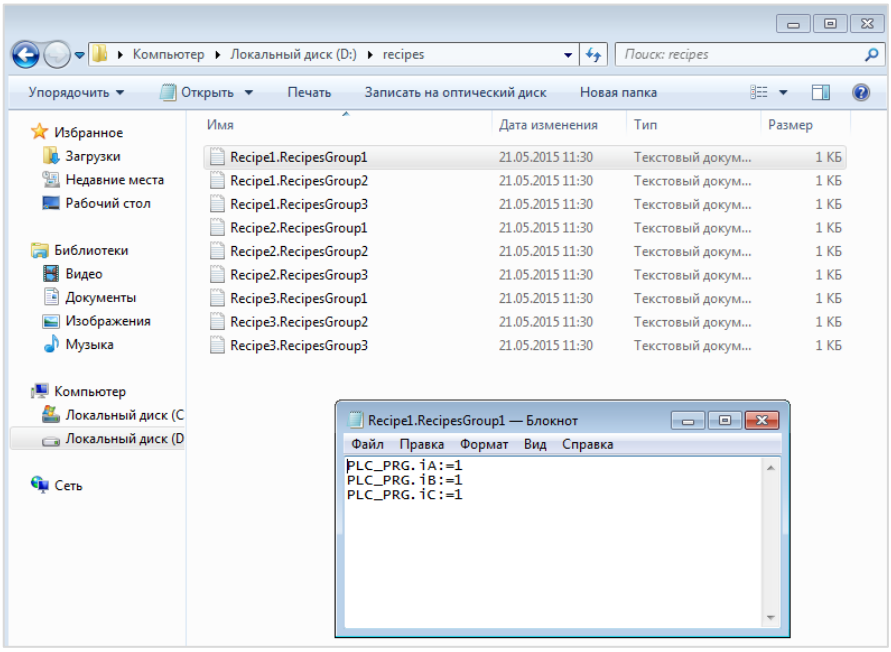


Рисунок 10.416 – Файлы рецептов

Проверка работы 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. ***Следует обратить 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 необязательно – оно будет сформировано автоматически.

ПРИМЕЧАНИЕ

Для контроллеров ОВЕН (и других целевых устройств 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.

10.3.9 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.

Подобные преобразования можно проводить непосредственно в программе, но это приведет к увеличению количества кода и используемых переменных.

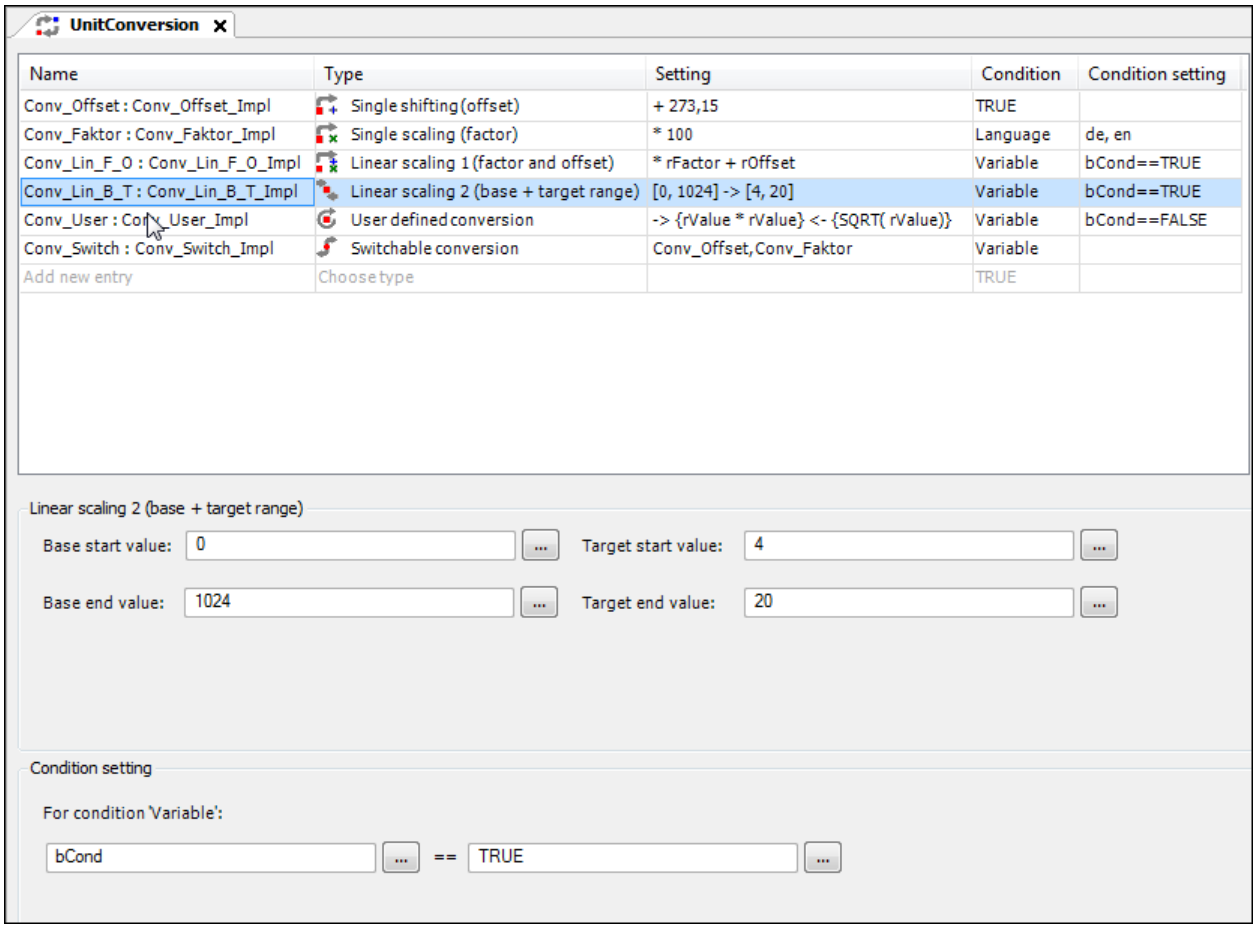


Рисунок 10.417 – Внешний вид компонента Unit Conversion

В качестве примера рассматривается преобразование токового сигнала **4…20 мА** в температуру из диапазона **−50…+150 °C** (при допущении линейной зависимости между величинами).

Пример создан в среде **CODESYS V3.5 SP11 Patch 5** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с таргет-файлом **3.5.11.5**.

Пример доступен для скачивания: Example_UnitConversion.projectarchive

Для создания примера с использованием конвертации значений следует:

1. Создать новый ***стандартный*** проект **CODESYS** с названием **Example_UnitConversion** и настройками по умолчанию: таргет – CODESYS Control Win V3, язык PLC_PRG – **ST**.
2. В программе **PLC_PRG** объявить переменную **rCurrentLoop** типа **REAL**:

```
PROGRAM PLC_PRG
VAR
    rCurrentLoop:REAL:=4; // токовый сигнал из диапазона 4..20 мА
END_VAR
```

Рисунок 10.418 – Объявление переменных программы PLC_PRG

3. Добавить в проект компонент **Unit Conversion** с названием по умолчанию:

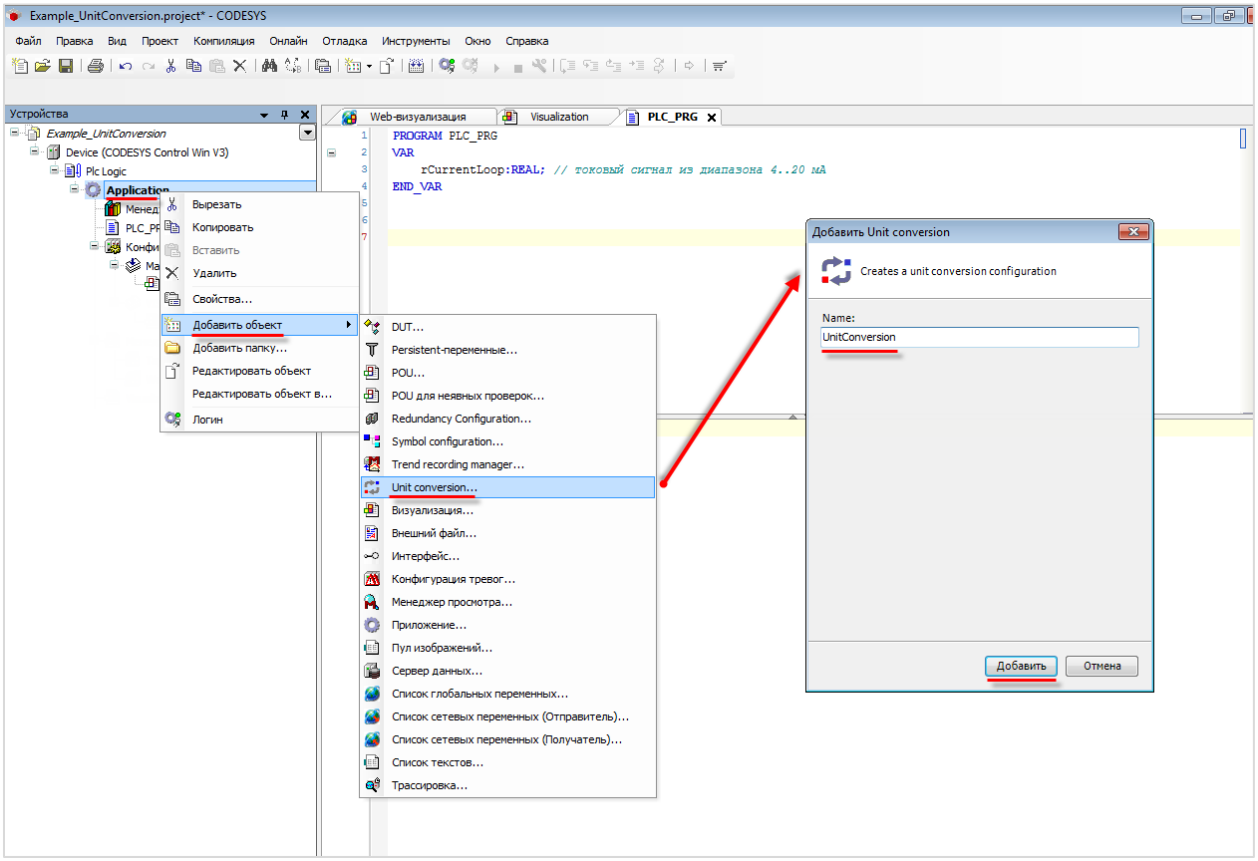


Рисунок 10.419 – Добавление в проект компонента Unit Conversion

4. Добавить новый шаблон преобразования **CurrentLoopToTemperature** типа **Linear scalling 1 (factor and offset)**:

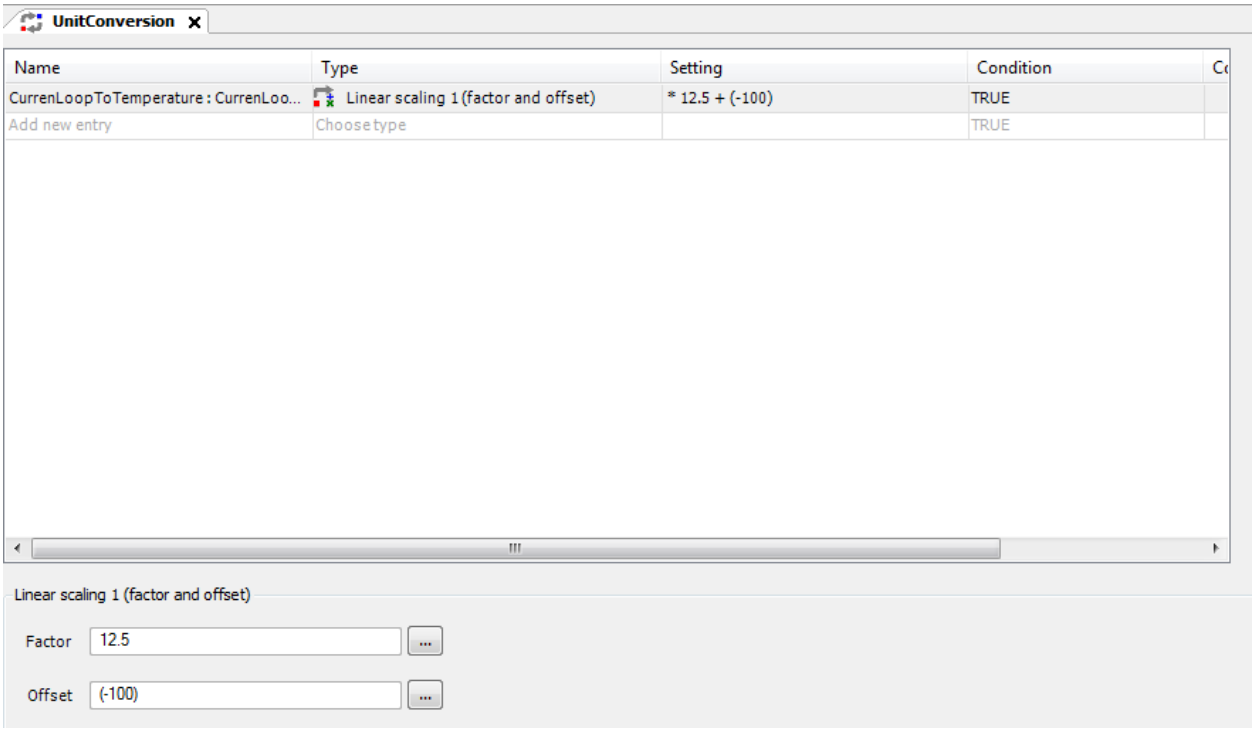


Рисунок 10.420 – Настройка компонента Unit Conversion

Для решения данной задачи следует составить два линейных уравнения.

После решения системы из двух линейных уравнений, получаем зависимость температуры от значения тока:

$$t = 12.5 \cdot I - 100,$$

где **I** – значение токового сигнала, мА.

Соответственно, в столбце **Setting** для шаблона задается **Factor = 12.5 и Offet = (−100)**.

В столбце **Condition** по умолчанию стоит значение **TRUE** – это означает, что преобразование будет выполняться всегда.

Подробное описание компонента приведено в пп. 7.

5. Добавить в проект экран визуализации **Visualization**. В его свойствах выбрать размер **480 × 272.** Экран будет содержать элемент Бегунок и элемент Текстовое поле. Поясняющая надпись сделана с помощью элемента Метка.

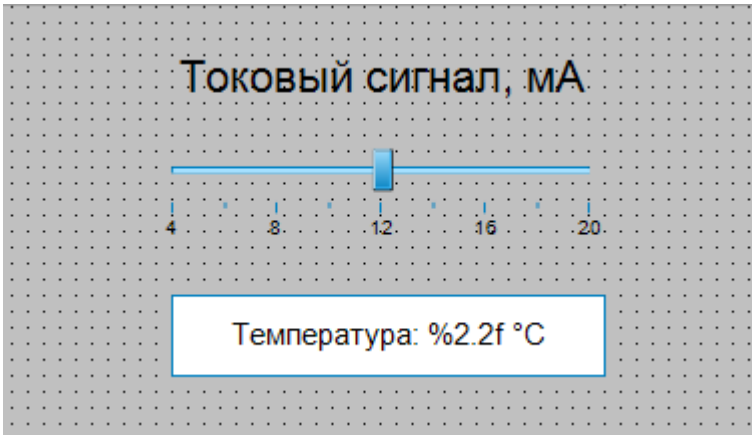


Рисунок 10.421 – Содержание экрана Visualization

Затем следует настроить 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	☑
Шкала	
Показать шкалу	☑
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	

Рисунок 10.422 – Настройки элемента Бегунок

Элемент Текстовое поле будет использоваться для отображения преобразованного значения. Для этого к нему следует привязать переменную **rCurrentLoop**, после чего в строке параметра **Текстовая переменная** нажать на ярлык **Unit Conversion** и в появившемся окне выбрать шаблон преобразования:

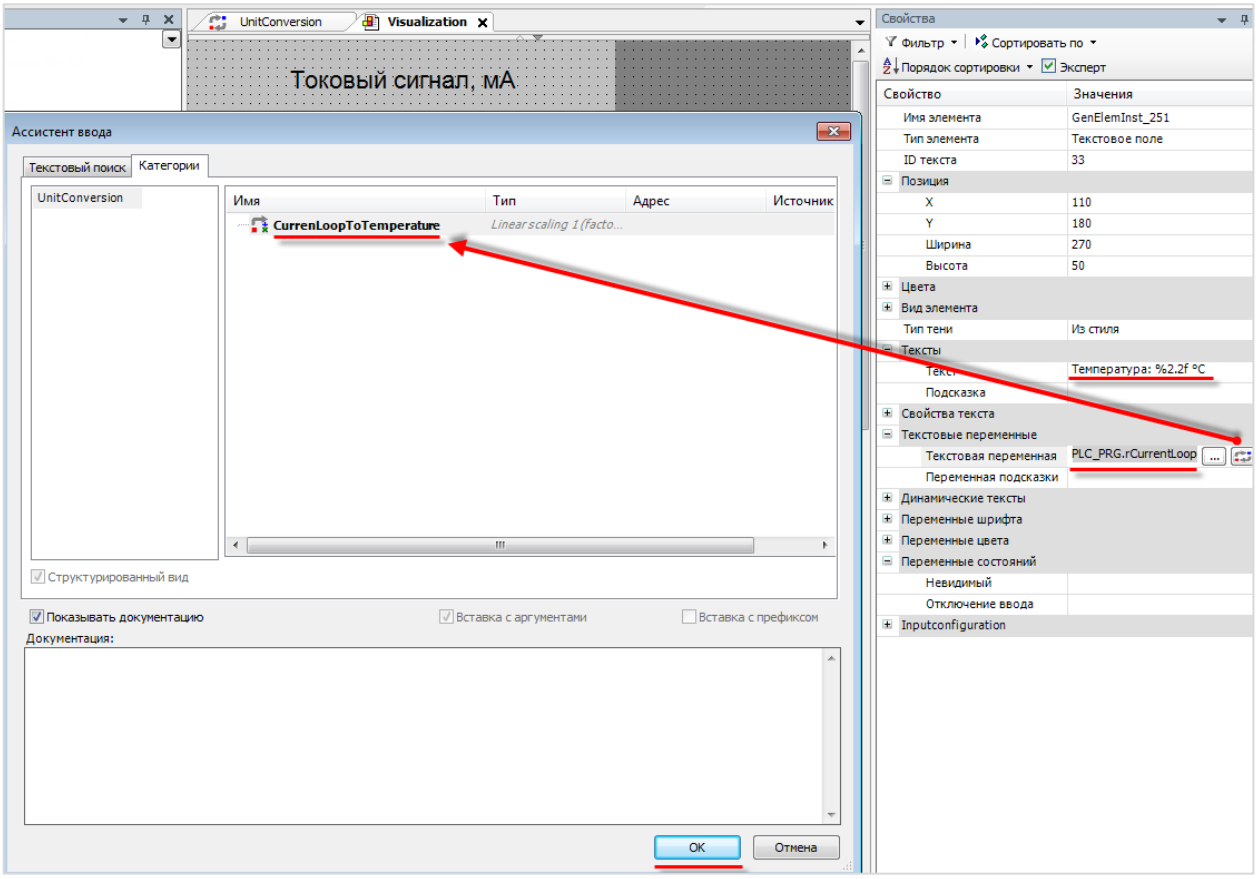
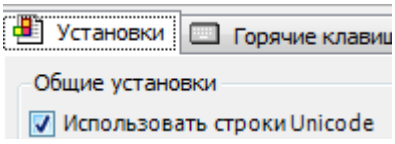


Рисунок 10.423 – Выбор шаблона преобразования для элемента Текстовое поле

ПРИМЕЧАНИЕ
Возможность выбора шаблона появляется только в случае добавления в проект компонента **Unit Conversion**.

6. Настроить Менеджер визуализации. В его установках следует поставить галочку **Использовать строки Unicode**:



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

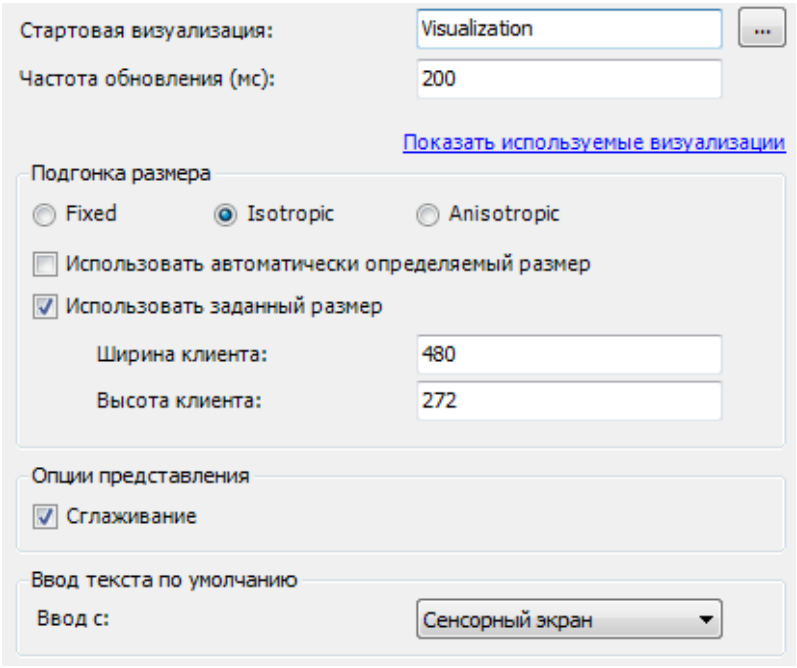


Рисунок 10.424 – Настройки таргет-визуализации

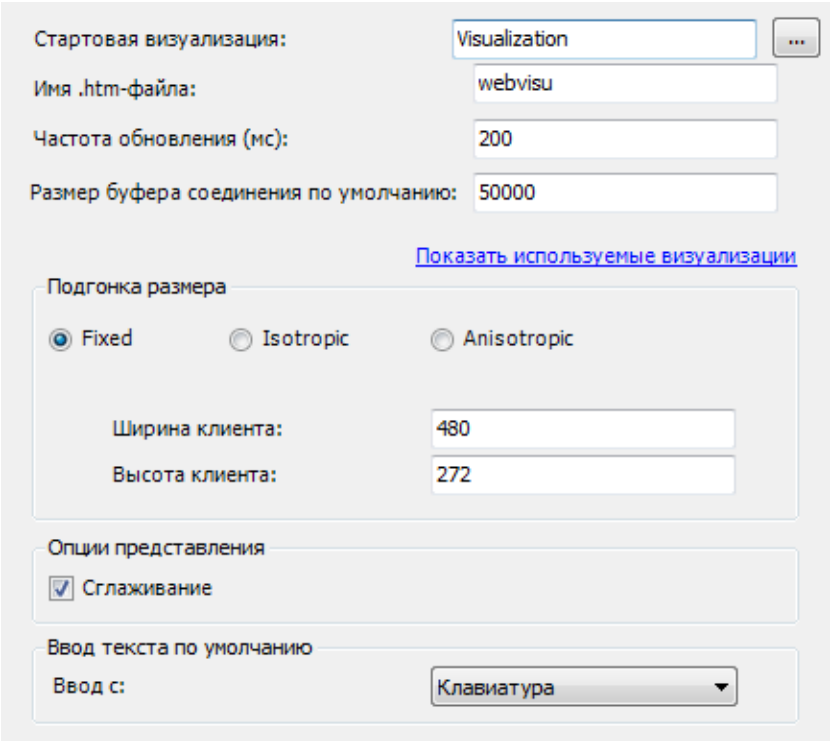


Рисунок 10.425 – Настройки web-визуализации

7. Запустить проект на виртуальном контроллере. Открыть в веб-браузере страницу http://localhost:8080/webvisu.htm, чтобы увидеть web-визуализацию проекта. ***Следует обратить внимание***, что браузер должен поддерживать **HTML5**. Проверить функционал проекта.

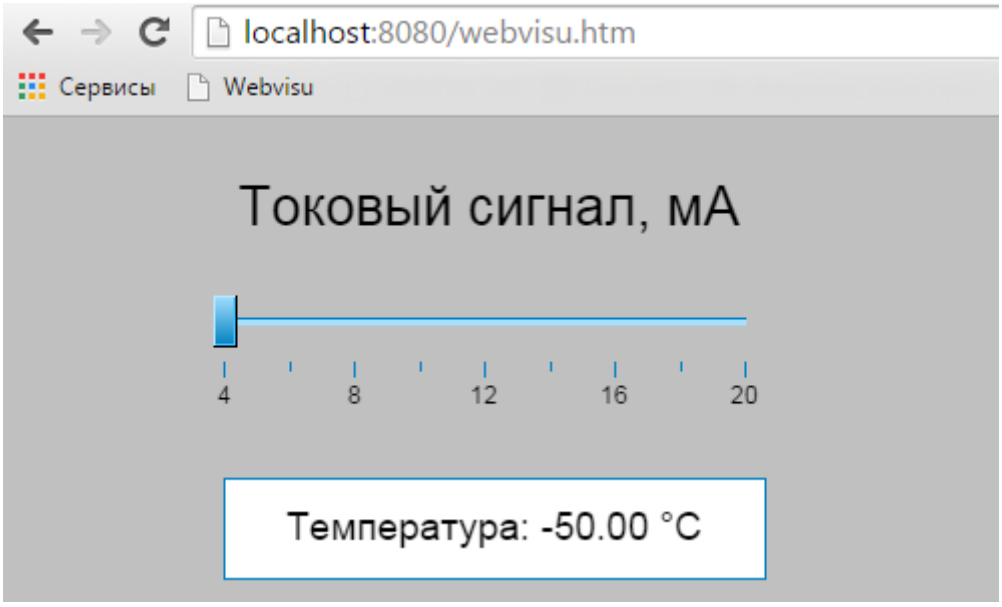


Рисунок 10.426 – Web-визуализация проекта в браузере Chrome

По умолчанию токовый сигнал равен **4 мА**, что соответствует нижней границе температурного диапазона **(−50 °C**). Меняя значение токового сигнала, можно наблюдать за соответствующим изменением температуры. К обоим элементам привязана одна и та же переменная.

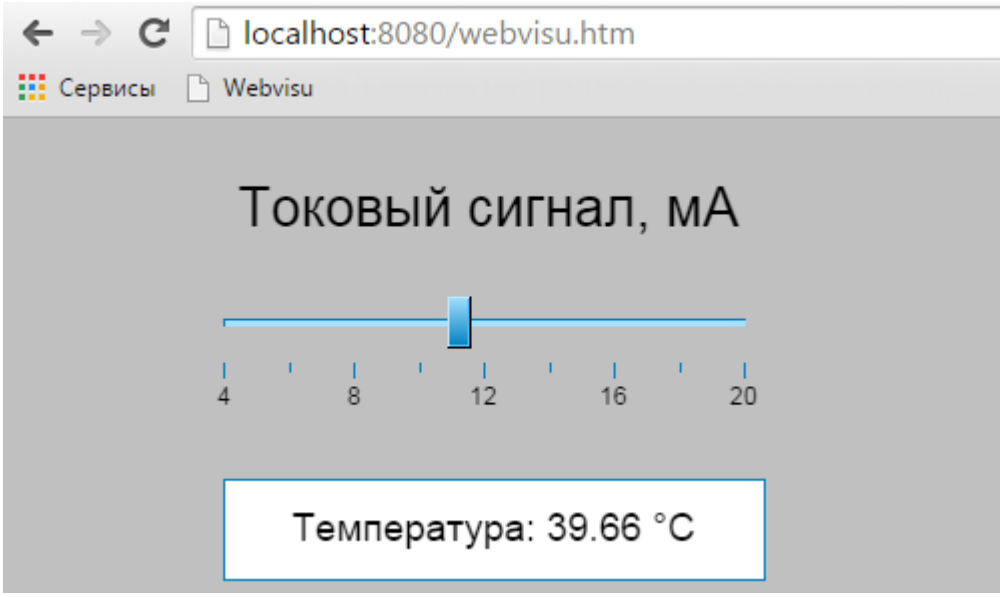


Рисунок 10.427 – Конвертация токового сигнала в температуру

8. Описание компонента **Unit Conversion:**

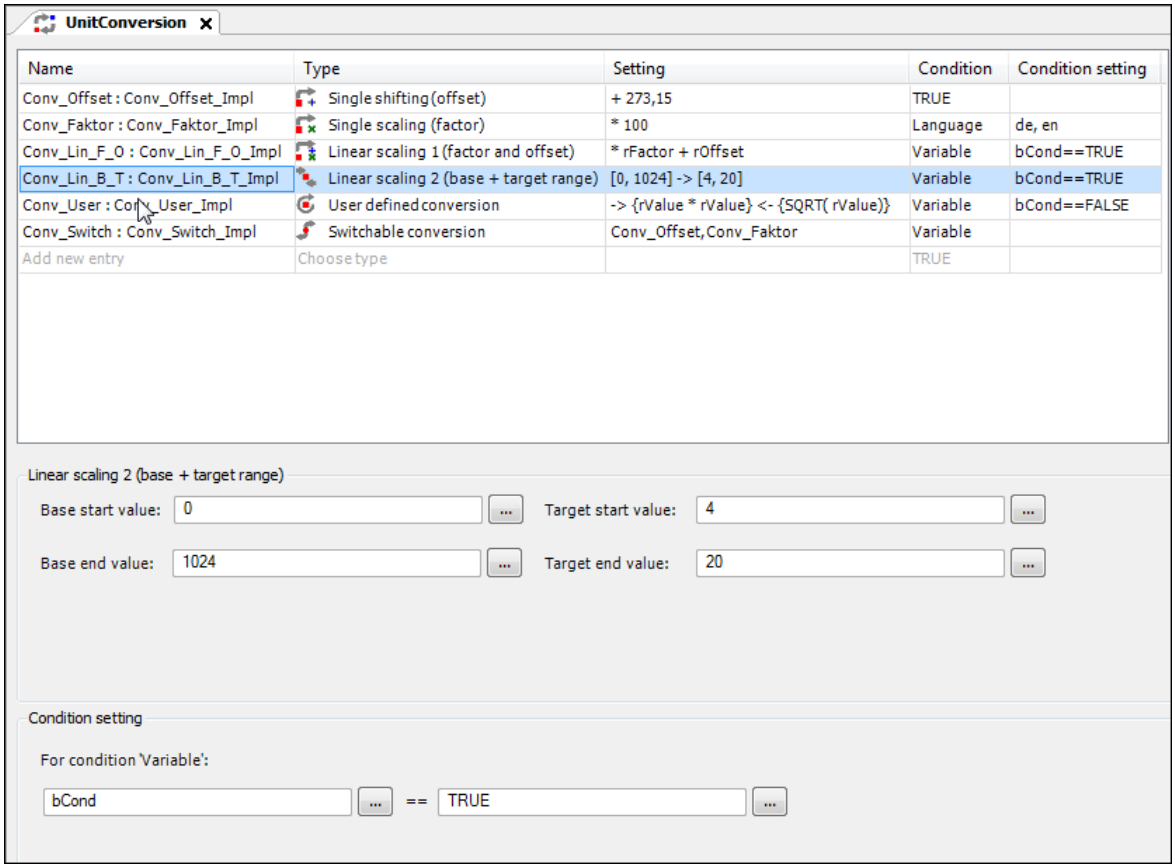


Рисунок 10.428 – Внешний вид компонента Unit Conversion

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 + смешение значения;
- **Linear Scaling 2 (base + target range)** – масштабирование значения из заданного начального диапазона в заданный конечный. Данный тип может использоваться только в случае нулевой нижней границы начального диапазона – именно поэтому он не используется 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 в случае выбора в проекте определенного языка (см. п. 10.3.5);
- **Variable** – шаблон работает только, 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:

```
PROGRAM PLC_PRG
VAR
    rValue:REAL;
    rConvertedValue:REAL;
END_VAR
```

```
// Прямое преобразование типа CurrentLoopToTemperature
// (из мА в градусы Цельсия)

rConvertedValue:=CurrentLoopToTemperature.Convert(rValue);


// Обратное преобразование типа CurrentLoopToTemperature
// (из градусов Цельсия в мА)

rValue:=CurrentLoopToTemperature.Reverse(rConvertedValue);
```

Рисунок 10.429 – Использование шаблонов преобразований в коде программы

10.4 Дополнительные примеры

10.4.1 Использование русскоязычной клавиатуры (библиотека OwenVisuDialogs)

Русскоязычная клавиатура входит в состав библиотеки **OwenVisuDialogs**. Также эта библиотека включает в себя кастомизированные версии других диалогов (например, Numpad).

Библиотека доступна для загрузки на сайте ОВЕН в разделе **CODESYS V3/Библиотеки и компоненты**. Библиотека распространяется в открытом виде (.library) и доступна для редактирования.

Информация о библиотеке приведена в этой статье.

Пример использования библиотеки доступен для скачивания:
Example_OwenVisuDialogs_3514v1.projectarchive

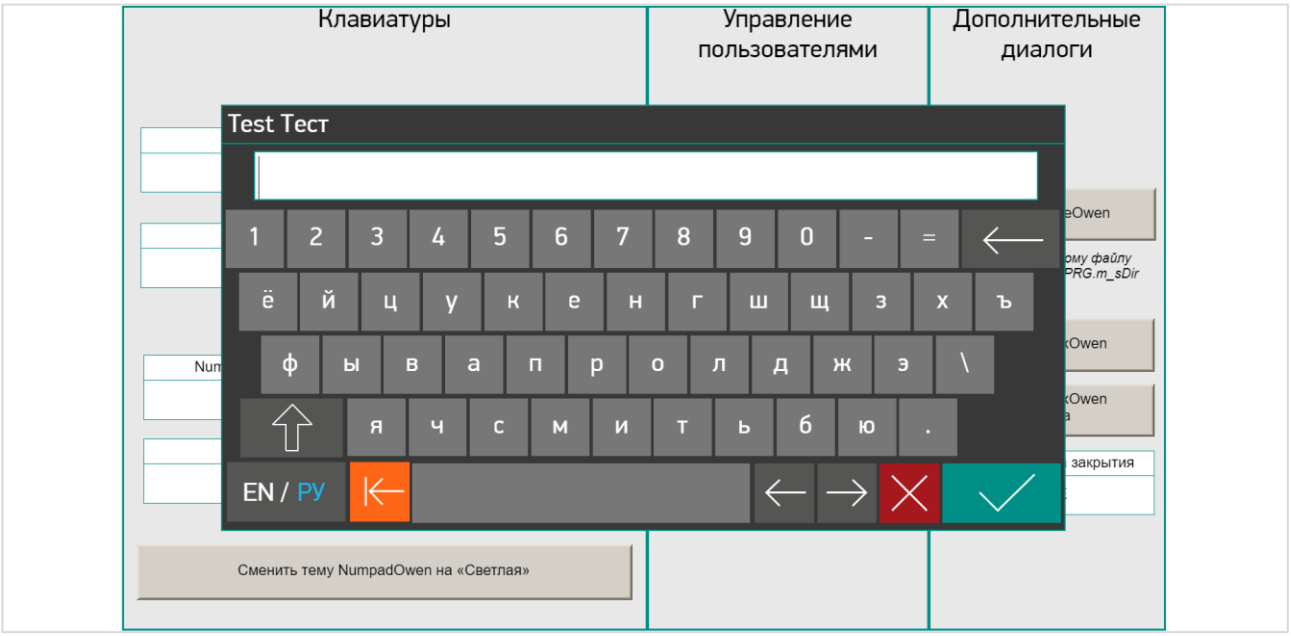


Рисунок 10.430 – Внешний вид русскоязычной клавиатуры из библиотеки OwenVisuDialogs

10.4.2 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.

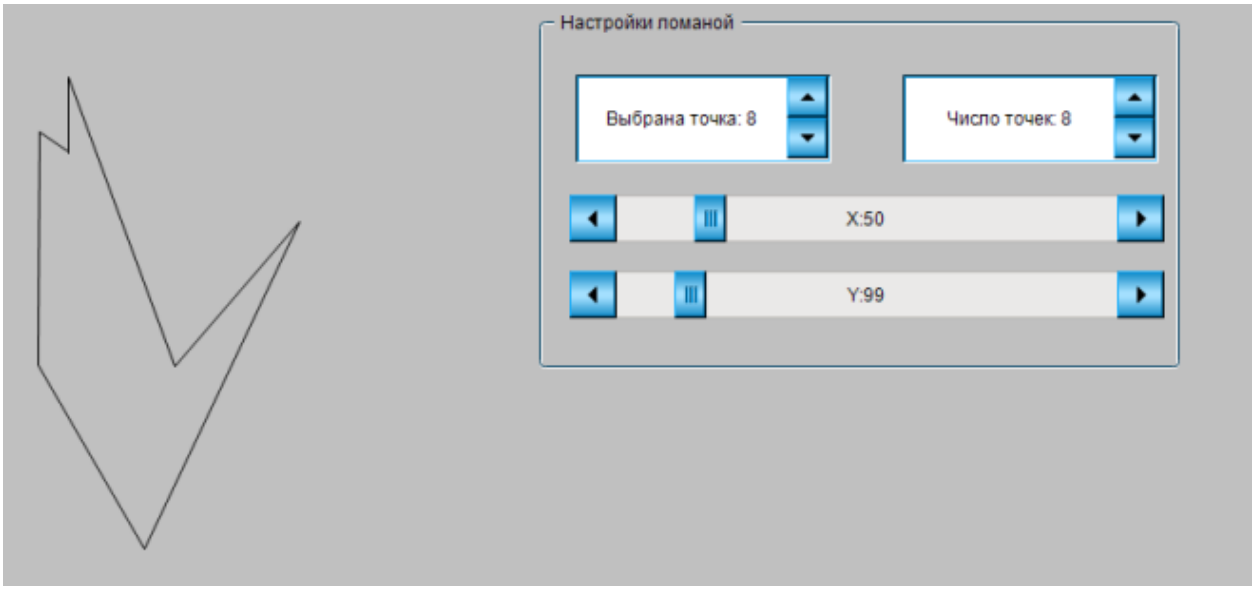


Рисунок 10.431 – Внешний вид примера Ломаная

Пример создан в среде **CODESYS V3.5 SP11 Patch 5** и подразумевает запуск на виртуальном контроллере **CODESYS Control Win V3** с таргет-файлом **3.5.11.5**.

Пример доступен для скачивания: Example_Polyline_3511v1.projectarchive

Для создания примера с использованием динамических точек следует:

1. Создать новый ***стандартный*** проект **CODESYS** с названием **Example_Polyline** и настройками по умолчанию: таргет – CODESYS Control Win V3, язык PLC_PRG – **ST**.
2. В **Менеджере библиотек** добавить библиотеку **CmpVisuHandler** версии **3.5.6.0** или выше (требуется для работы с динамическими точками):

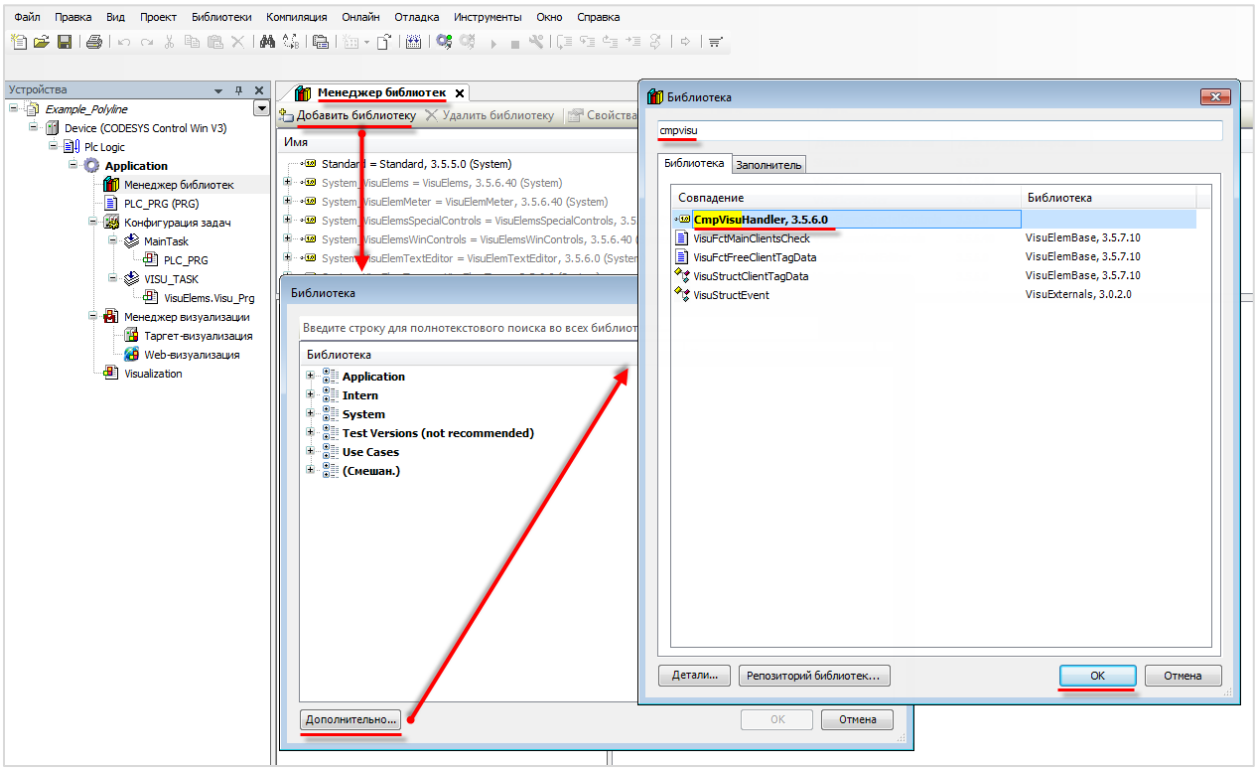


Рисунок 10.432 – Добавление библиотеки CmpVisuHandler

3. В программе **PLC_PRG** объявить 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
```

Рисунок 10.433 – Объявление переменных программы PLC_PRG

4. Добавить в проект экран визуализации **Visualization**. В его свойствах выбрать размер **800 × 480.** Экран будет содержать элемент Ломаная, элемент Группа, два элемента Управление вращением и два элемента Полоса прокрутки.

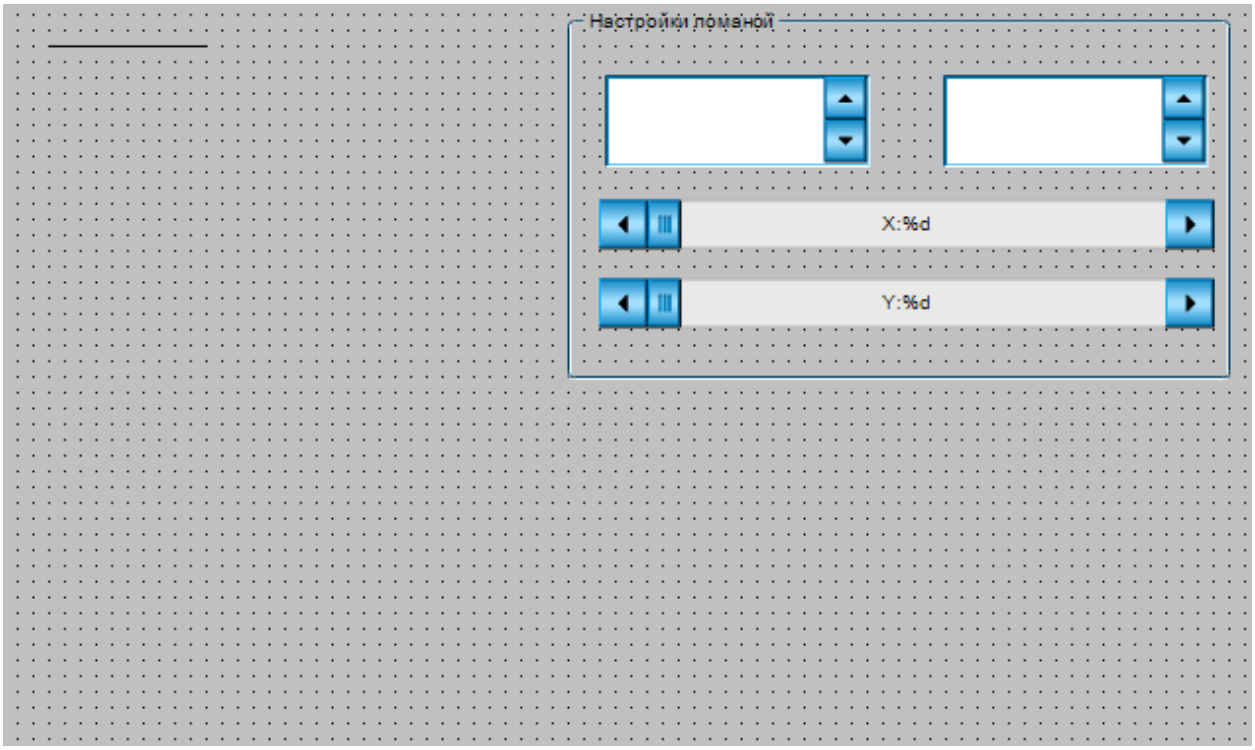


Рисунок 10.434 – Содержимое экрана визуализации Visualization

5. Настроить элемент Ломаная. ***Следует обратить 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

Рисунок 10.435 – Настройки элемента Ломаная

6. Настроить 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

Рисунок 10.436 – 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...	

Рисунок 10.437 – Настройки элемента Управление вращением 2

7. Настроить элементы Полоса прокрутки. Данные 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].X
Минимальное значение	0
Максимальное значение	300
Размер страницы	
Прокрутка выполнена	☐
Позиция	
X	20
Y	128
Ширина	390
Высота	30
Линейка	
Цвета	
Тексты	
Текст	X:%d
Подсказка	

Рисунок 10.438 – Настройки элемента Полоса прокрутки 1

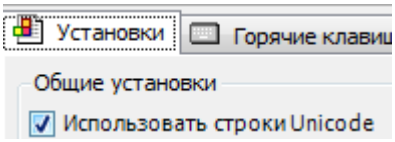
Свойства

Фильтр | Сортировать по | Порядок сортировки | Эксперт

Свойство	Значения
Имя элемента	GenElemInst_32
Тип элемента	Полоса прокрутки
ID текста	6
Значение	PLC_PRG.arrPolylineCoordinates[PLC_PRG.iIndex].Y
Минимальное значение	0
Максимальное значение	800
Размер страницы	
Прокрутка выполнена	☐
⊟ Позиция	
X	20
Y	178
Ширина	390
Высота	30
⊞ Линейка	
⊞ Цвета	
⊟ Тексты	
Текст	Y:%d
Подсказка	

Рисунок 10.439 – Настройки элемента Полоса прокрутки 2

8. Настроить компонент Менеджер визуализации. В его установках следует поставить галочку **Использовать строки Unicode**:



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

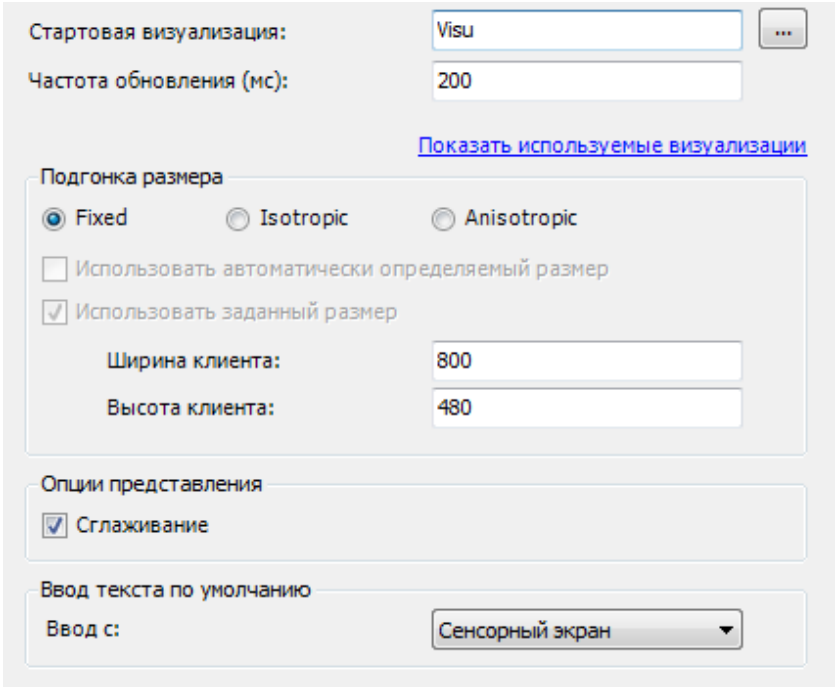


Рисунок 10.440 – Настройки таргет-визуализации

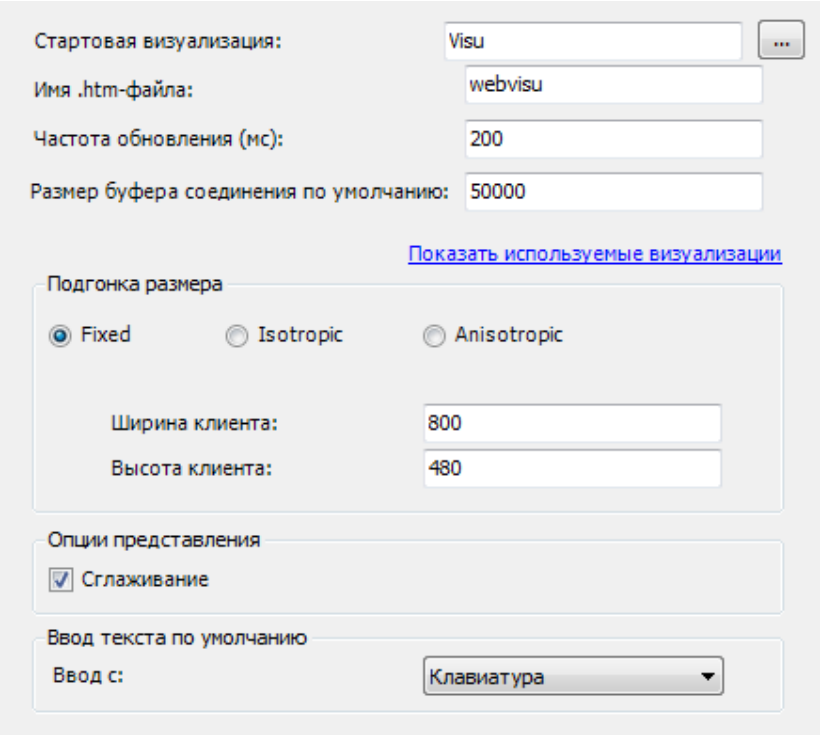


Рисунок 10.441 – Настройки web-визуализации

9. Запустить проект на виртуальном контроллере. Открыть в веб-браузере страницу http://localhost:8080/webvisu.htm, чтобы увидеть web-визуализацию проекта. ***Следует обратить внимание***, что браузер должен поддерживать **HTML5**. Проверить функционал проекта.

Рисунок 10.442 – Web-визуализация проекта в браузере Chrome

В качестве примера следует задать число точек фигуры (например, **10**). С помощью полос прокрутки выбрать координаты первой точки (например, **100, 100**). Затем выбрать вторую точку, после чего задать ее координаты. Каждая последующая точка будет автоматически соединяться с предыдущей. В результате, по заданным пользователем координатам будет построена некая геометрическая фигура:

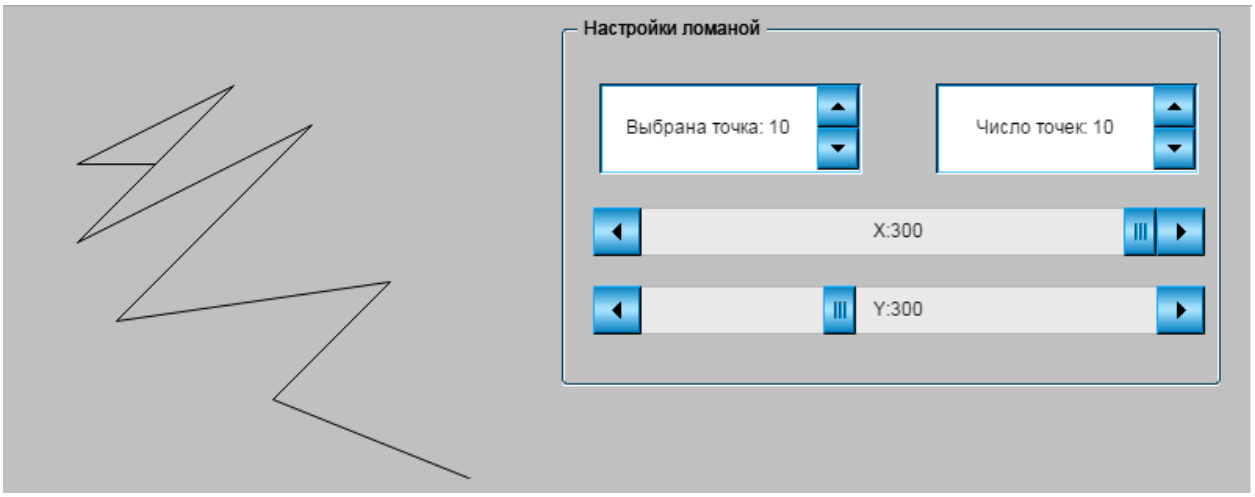


Рисунок 10.443 – Построение фигуры по координатам точек

10.4.3 Считывание координат курсора

Данный пример посвящен работе с курсором в визуализации **CODESYS**. Системные переменные контроллеров ОВЕН СПК позволяют получить данные о текущих координатах курсора и числе нажатий на дисплей. В примере пользователь будет перемещать элемент с помощью движения курсора.

Рисунок 10.444 – Внешний вид примера Считывание координат курсора

Пример создан в среде **CODESYS V3.5 SP11 Patch 5** и подразумевает запуск на контроллере **СПК1xx [M01]** с таргет-файлом **3.5.11.5**.

Пример доступен для скачивания: Example_Cursor.projectarchive

ПРИМЕЧАНИЕ
Данный функционал поддерживают только контроллеры ОВЕН СПК.

ПРИМЕЧАНИЕ
Данный функционал работает только в таргет-визуализации.

Для работы примера со считыванием координат курсора следует:

1. Создать новый ***стандартный*** проект **CODESYS** с названием **Example_Cursor** и настройками по умолчанию: таргет – CODESYS Control Win V3, язык PLC_PRG – **ST**.
2. В программе **PLC_PRG** объявить следующие переменные:

```
PLC_PRG
1  PROGRAM PLC_PRG
2  VAR
3       iCurrentX:          INT;
4       iCurrentY:          INT;
5       udiTouchCount:      UDINT;
6       xIsSelected:        BOOL;
7  END_VAR
8
9  // параметры элемента визуализации
10 VAR CONSTANT
11      c_iElementCoordX:   INT := 30;
12      c_iElementCoordY:   INT := 30;
13      c_iElementWidth:    INT := 115;
14      c_iElementHeight:   INT := 76;
15 END_VAR
```

Рисунок 10.445 – Объявление переменных программы PLC_PRG

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

```
1  udiTouchCount := Screen.countTouch;
2
3  IF xIsSelected THEN
4      iCurrentX := Screen.cursorX - c_iElementCoordX - c_iElementWidth/2;
5      iCurrentY := Screen.cursorY - c_iElementCoordY - c_iElementHeight/2;
6  END_IF
```

Рисунок 10.446 – Код программы PLC_PRG

4. Добавить в проект экран визуализации **Visualization**. В его свойствах выбрать размер **800 × 480.** Экран будет содержать элемент Эллипс, элемент Группа и три элемента Прямоугольник.

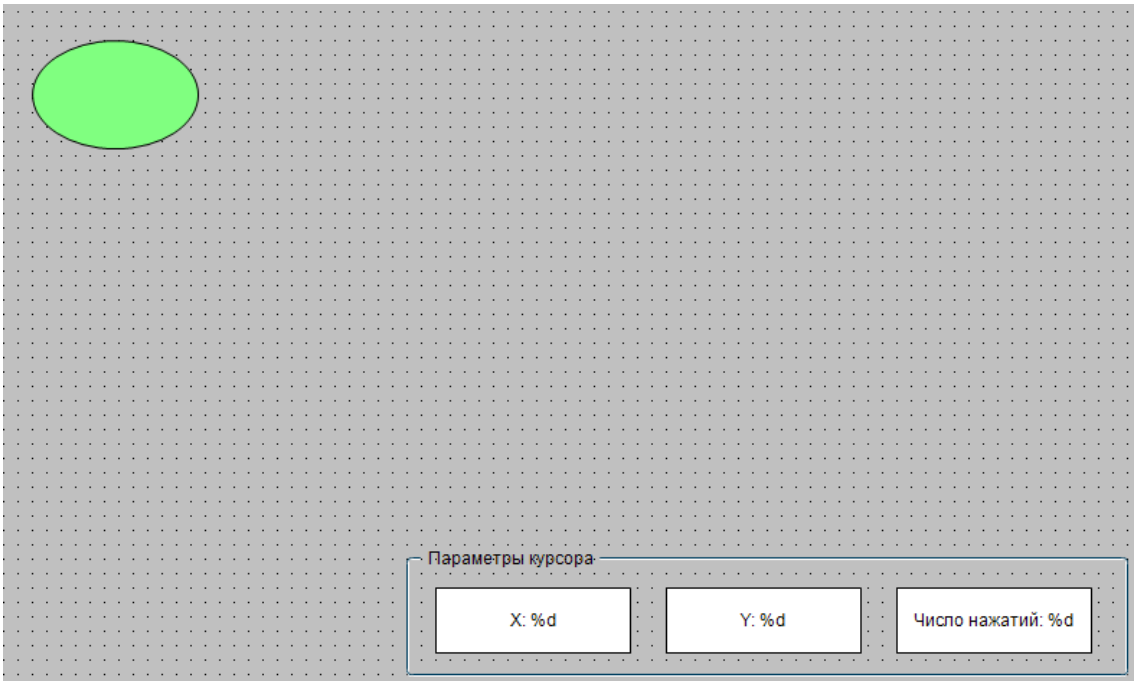


Рисунок 10.447 – Содержимое экрана визуализации Visualization

5. Настроить элемент Эллипс. После нажатия и отпускания (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.

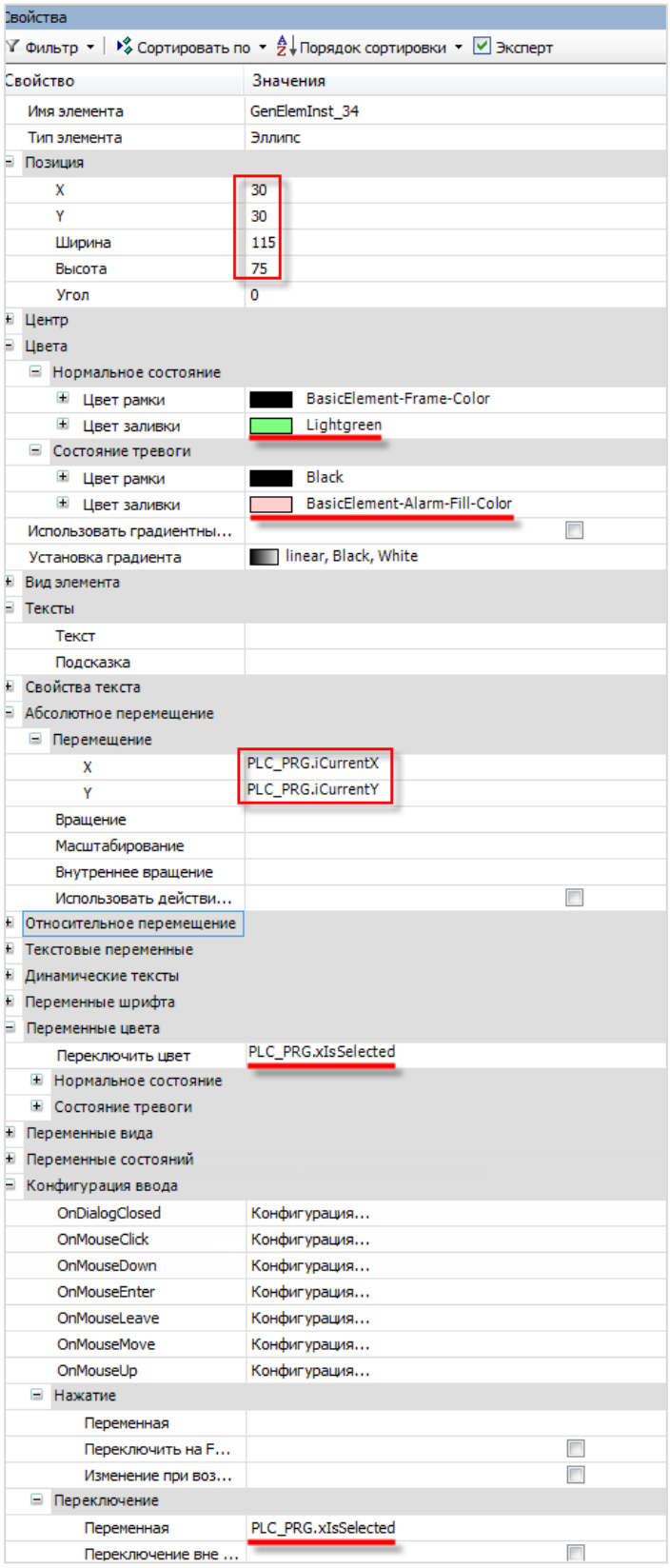


Рисунок 10.448 – Настройки элемента Эллипс

6. Настроить элементы вкладки **Параметры курсора**. Элементы данной вкладки будут отображать координаты текущего положения курсора и общее число нажатий на элемент.

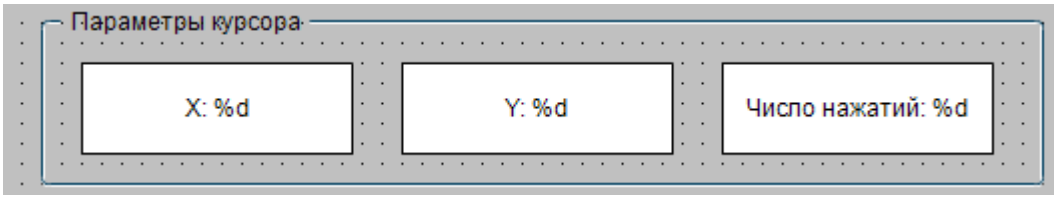


Рисунок 10.456 – Внешний вид вкладки Нажатие

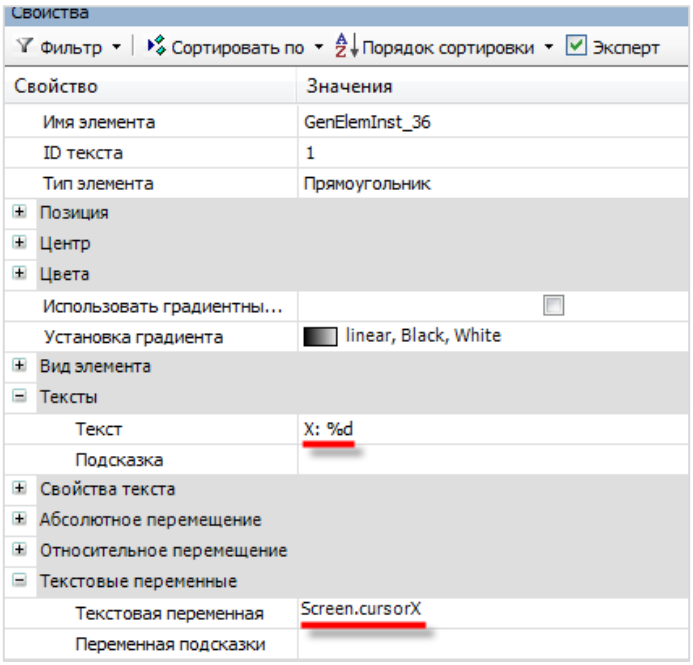


Рисунок 10.449 – Настройки элемента Прямоугольник 1

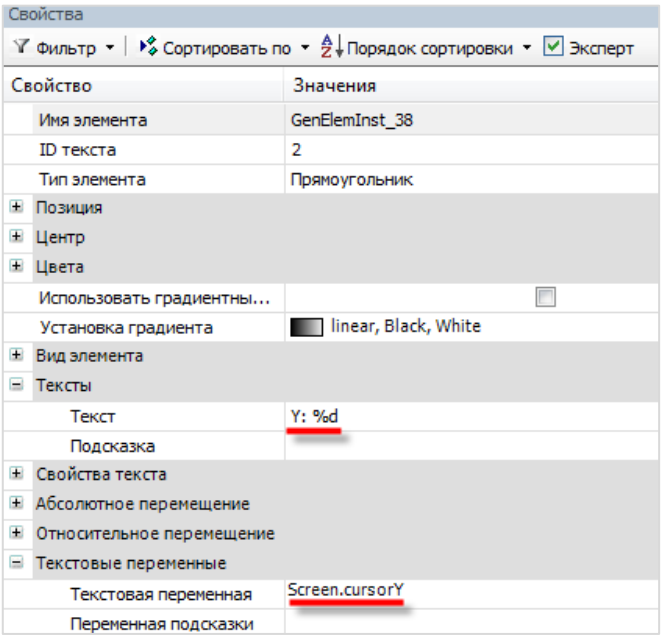


Рисунок 10.450 – Настройки элемента Прямоугольник 2

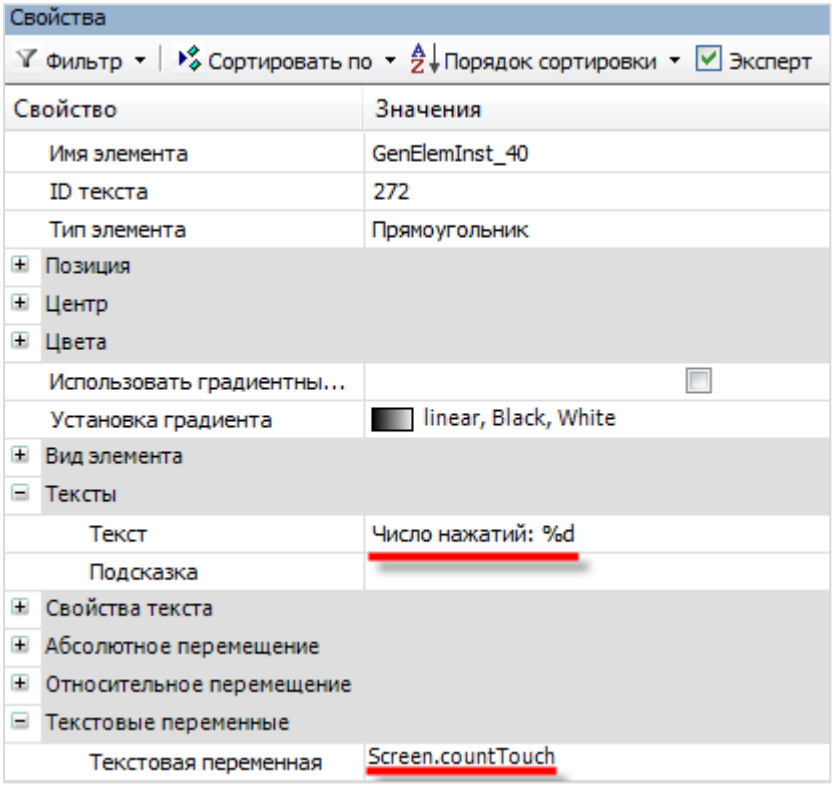
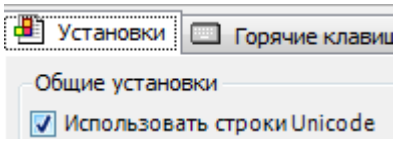


Рисунок 10.451 – Настройки элемента Прямоугольник 3

7. Настроить компонент Менеджер визуализации. В его установках следует поставить галочку **Использовать строки Unicode**:



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

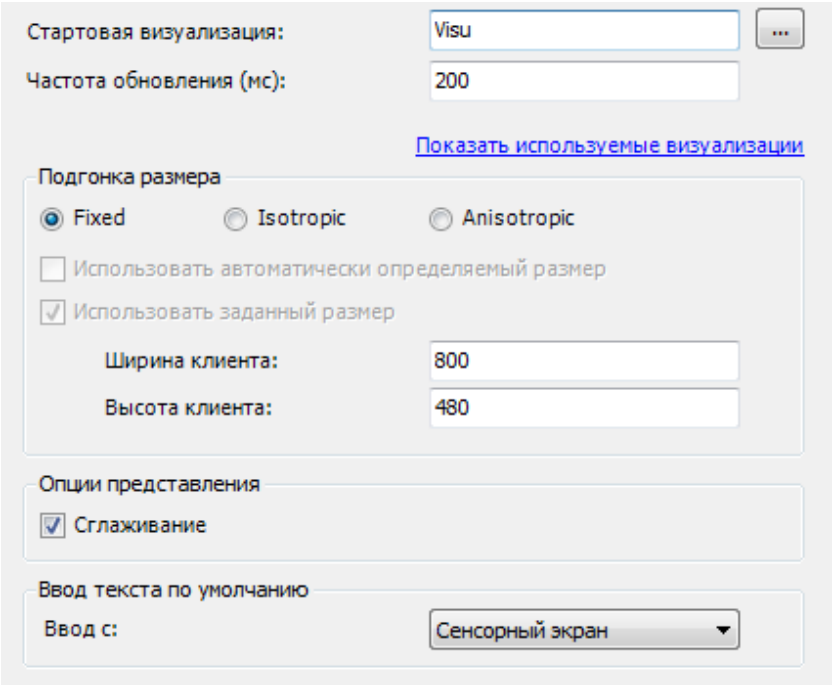


Рисунок 10.452 – Настройки таргет-визуализации

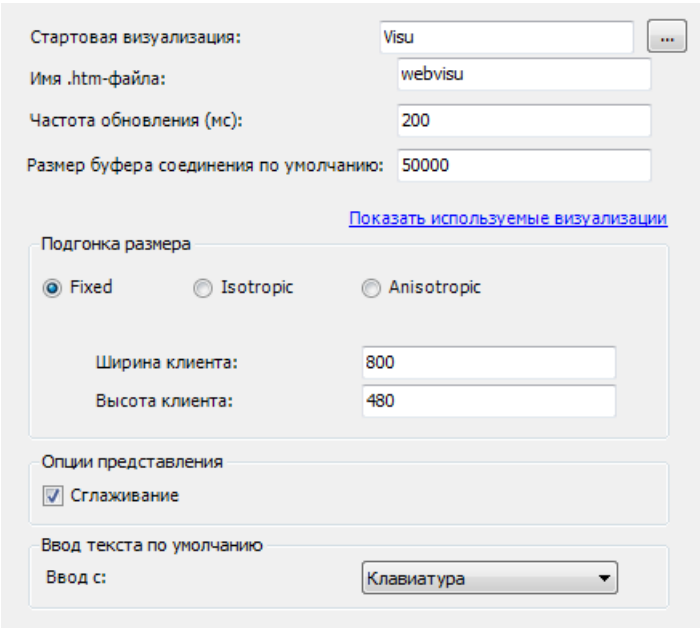


Рисунок 10.453 – Настройки web-визуализации

8. Запустить проект на контроллере. Проверить функционал проекта в таргет-визуализации контроллера.

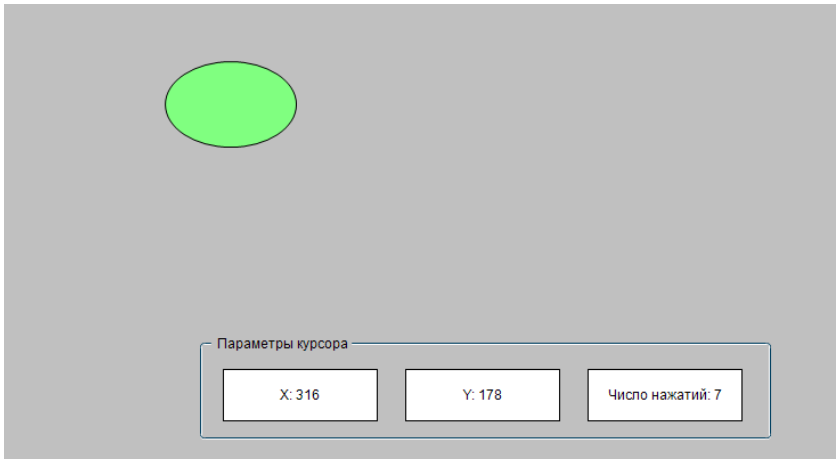


Рисунок 10.454 – Таргет-визуализация контроллера

Сначала следует нажать на эллипс, после чего перемещать курсор по экрану. Эллипс изменит свой цвет и будет перемещаться вслед за курсором. Информация во вкладке **Параметры курсора** будет обновляться по мере взаимодействия с элементом. Чтобы зафиксировать элемент на новом месте, следует нажать на этом месте курсором.

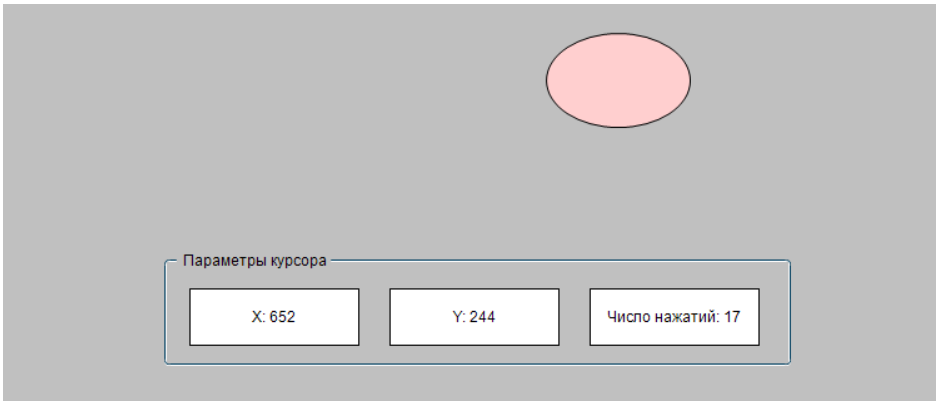


Рисунок 10.455 – Перемещение элемента с помощью курсора

10.4.4 Многопользовательский доступ. Независимая обработка клиентов визуализации

При использовании контроллеров с web-визуализацией 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 – так как в случае установки соответствующей галочки в Менеджере визуализации визуализация становится общей для всех клиентов, и переключение экранов затрагивает всех пользователей.

Простого решения данной задачи (программной обработки визуализаций в случае многопользовательского доступа) не существует.

В данном пункте приводятся 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 следует воспользоваться 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 таргет-визуализации.

sVisuName – переменная типа **STRING**, содержащая имя экрана визуализации, который необходимо открыть.

2. Независимая работа клиентов на одном экране

В пункте 10.3.3 описывается работа с **интерфейсом фрейма**.

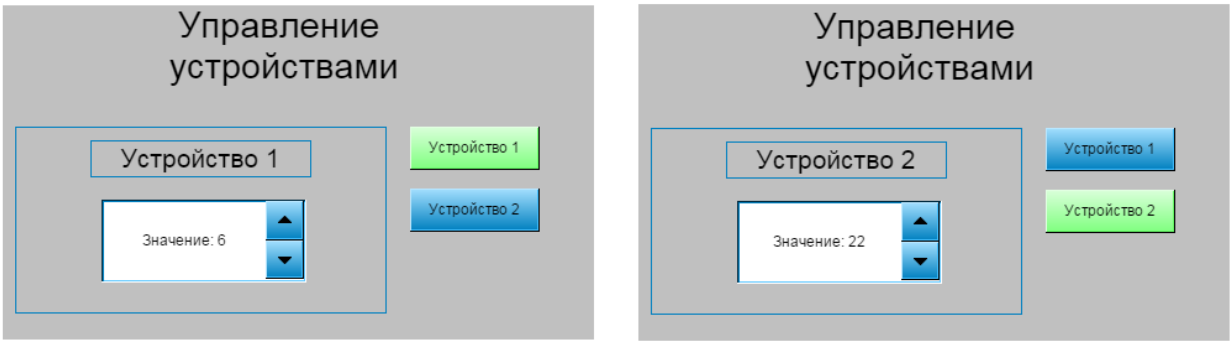


Рисунок 10.456 – Внешний вид примера Интерфейс фрейма.
На обоих рисунках в фрейме открыт один и тот же экран визуализации

Данный пример не подразумевает многопользовательский доступ (впрочем, как практически все рассмотренные в документе примеры) – так как все пользователи взаимодействует с одними и теми же переменными, привязанными к кнопкам.

Пример: пусть **первый веб-клиент** визуализации должен нажать на кнопку **Устройство 2**. В результате экран будет выглядеть в соответствии с правой половиной рисунка 11.468.

Тогда **второй веб-клиент** во время подключения увидит следующее:

Рисунок 10.457 – Выделенная кнопка не соответствует отображаемому устройству

Т. е. несмотря на то, что пользователь видит в фрейме экран **Устройство 1**, активна кнопка **Устройство 2**, так как ее нажал первый пользователь.

Это можно избежать с помощью использования системной переменной **CURRENTCLIENTID**, которая содержит индекс текущего клиента визуализации. Для ее использования следует добавить в проект библиотеку **VisuGlobalClientManager**.

Сначала следует открыть проект Example_FrameInteface.

В переменных программы вместо **BOOL** переменных **bButtonDevice1** и **bButtonDevice2** следует объявить массивы аналогичного типа, каждый из которых содержит по **10** значений. Число **10** соответствует количеству клиентов визуализации, которые могут единовременно работать с проектом.

```
VAR
    // Переменные устройства 1
    arrButtonDevice1:ARRAY [1..10] OF BOOL:=[10(TRUE)];        // переменная для изменения цвета кнопки при выборе устройства
    wsNameDevice1:WSTRING:="Устройство 1";  // название устройства
    iValueDevice1:INT;                      // параметр устройства

    // Переменные устройства 2
    arrButtonDevice2:ARRAY [1..10] OF BOOL;
    wsNameDevice2:WSTRING:="Устройство 2";
    iValueDevice2:INT;
END VAR
```

Рисунок 10.458 – Объявление переменных программы PLC_PRG

На экране **Visualization** следует отредактировать ST-код действий кнопок **Устройство 1** и **Устройство 2**:

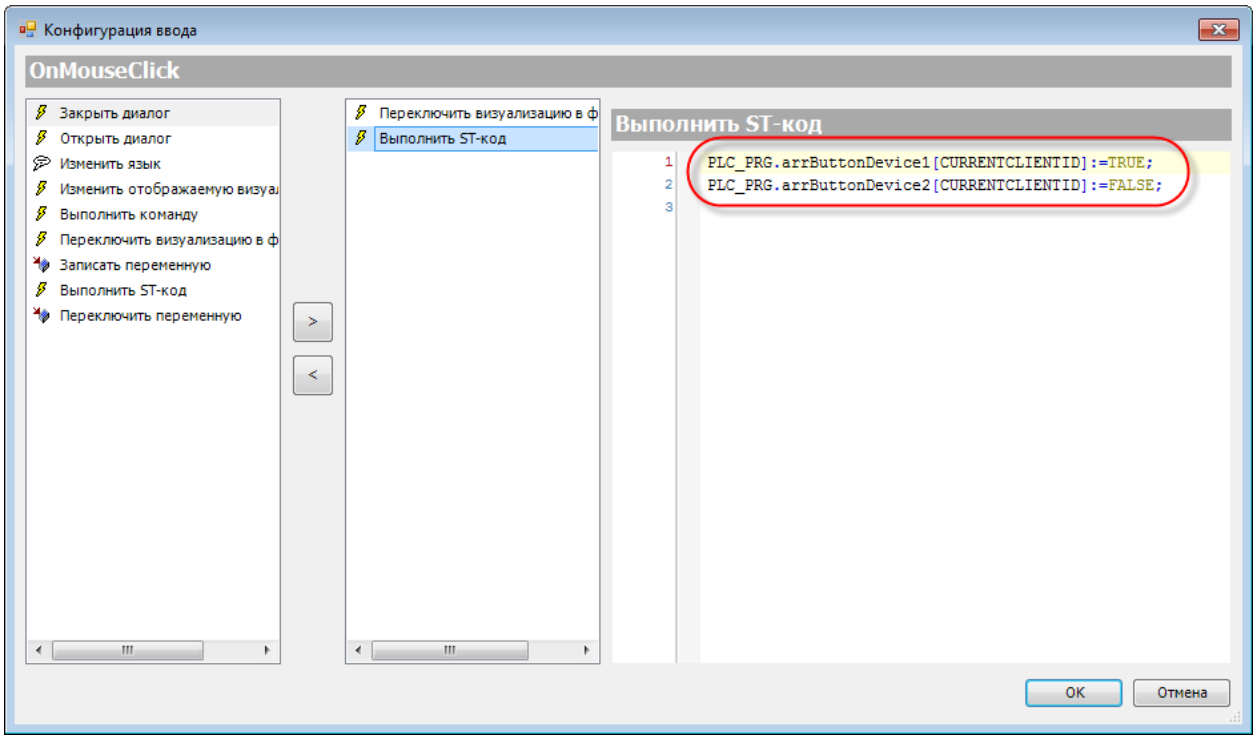


Рисунок 10.459 – Изменение ST-кода кнопки Устройство 1

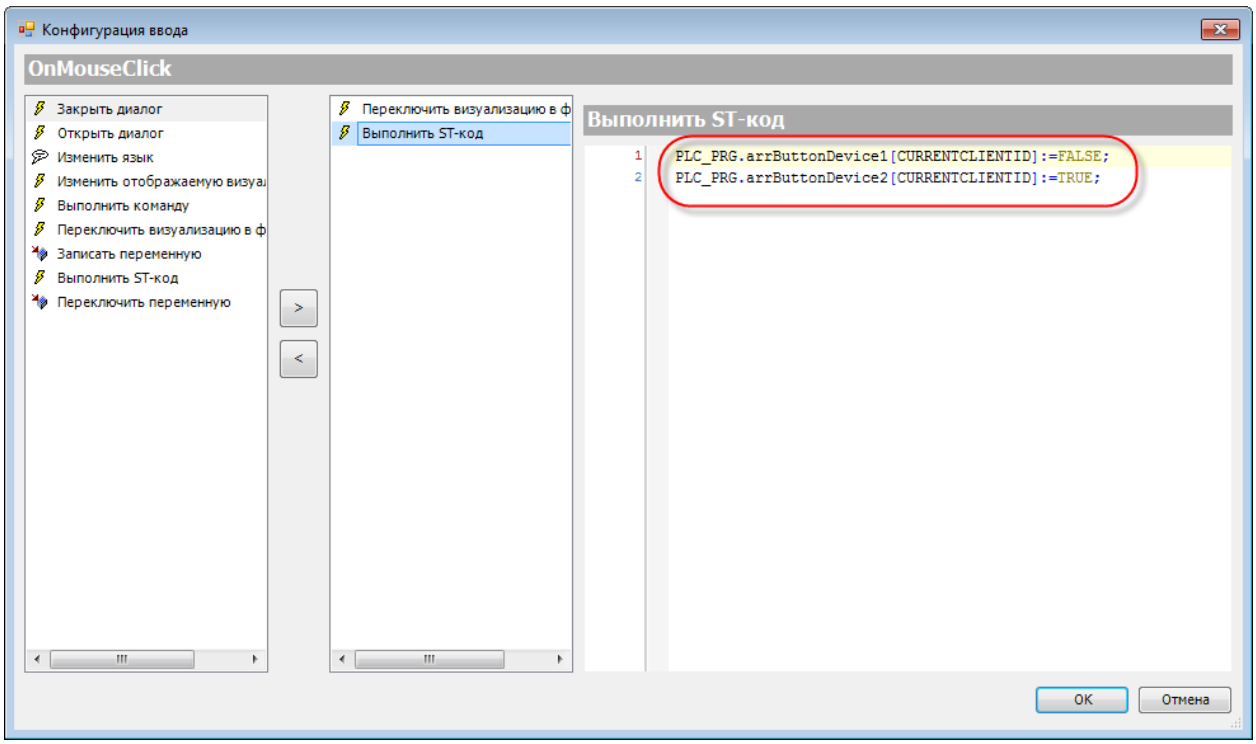


Рисунок 10.460 – Изменение ST-кода кнопки Устройство 2

К параметрам **Переключить цвет** обоих кнопок следует привязать соответствующие переменные:

Переменные цвета	
Переключить цвет	PLC_PRG.arrButtonDevice1[CURRENTCLIENTID]
Цвет	
Цвет тревоги	

Рисунок 10.461 – Параметры элемента Устройство 1

Переменные цвета	
Переключить цвет	PLC_PRG.arrButtonDevice2[CURRENTCLIENTID]
Цвет	
Цвет тревоги	

Рисунок 10.462 – Параметры элемента Устройство 2

В результате, каждый из пользователей будет работать со своими переменными, характеризующими активность кнопок, что позволит избежать ситуации, приведенной на рисунке 10.457.

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 CODESYS Group (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. Это рекомендуется только ***опытным*** пользователям.

Примеры доступны для скачивания:

ClientIndependentVisuSwitching.package

VisualizationSwitching.package

ResponsiveDesignExample.package

AdvancedVisu.rar

Файлы формата **.package** устанавливаются с помощью **Менеджера пакетов** (вкладка **Инструменты**) или (начиная с версии **CODESYS V3.5 SP17**) с помощью утилиты **CODESYS Installer**.

4. Библиотека **Visu Utils**

Библиотека **Visu Utils** позволяет получить информацию о клиентах визуализации, переключать экраны и открыть диалоги для конкретных группы клиентов. Описание библиотеки приведено в справке CODESYS.

5. Пример менеджера клиентов

Данный пример демонстрирует принцип получения информации о клиентах визуализации с помощью системных библиотеки **Visu Utils**, а также открытия и закрытия диалогов для конкретных групп клиентов.

Пример создан в среде **CODESYS V3.5 SP17 Patch 3** и подразумевает запуск на контроллере **СПК1xx [M01]** с таргет-файлом **3.5.17.31**.

Пример доступен для скачивания: Example_ClientManager.projectarchive

10.4.5 Дополнительные графические библиотеки

Во время добавления в проект первого экрана визуализации пользователь может включить поддержку библиотеки **VisuSymbols:**

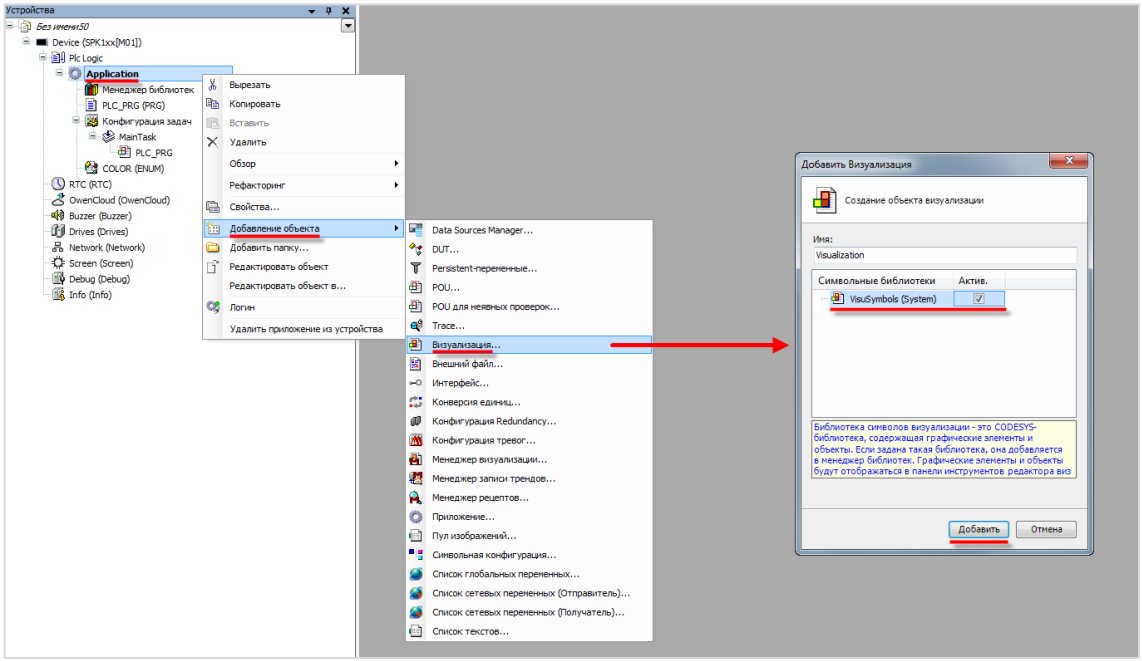


Рисунок 10.463 – Включение поддержки библиотеки VisuSymbols

В результате на панели инструментов редактора визуализации появится вкладка **Symbols**, содержащая различные графические пиктограммы:

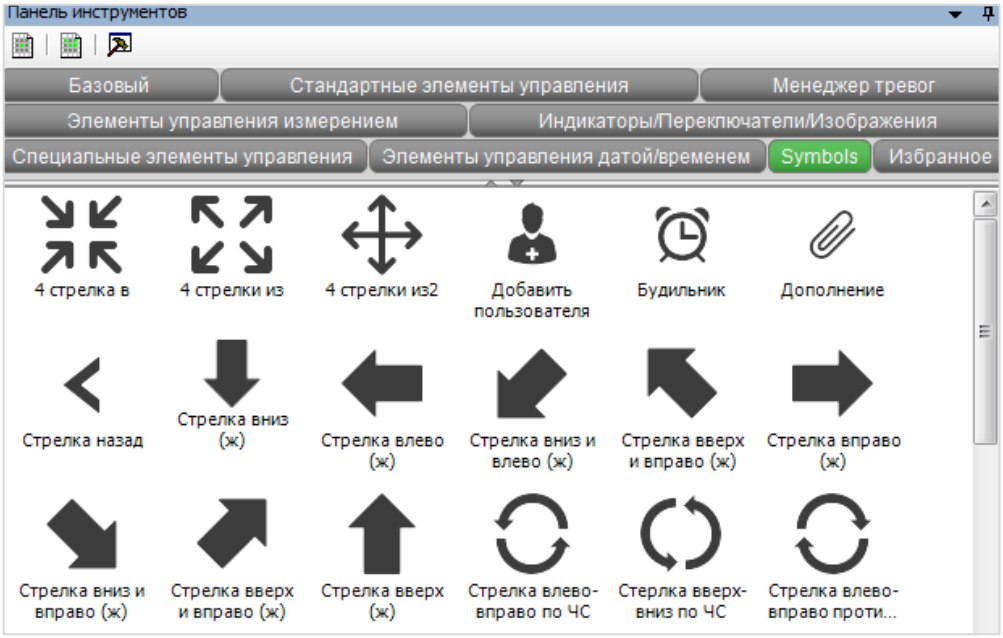


Рисунок 10.464 – Вкладка Symbols в редакторе визуализации

ПРИМЕЧАНИЕ
Библиотека **VisuSymbols** может быть добавлена в проект через **Менеджер библиотек.**

10.4.6 Кастомизация диалогов ввода

Стандартные диалоги ввода (**Numpad**, **Keypad**, **Login**, **Change Password**, **User Config**) могут быть адаптированы под потребности пользователя. Для этого следует открыть библиотеку, содержащую исходники этих диалогов. Диалоги **Numpad** и **Keypad** содержатся в библиотеке **VisuDialogs**. Диалоги **Login**, **Change Password** и **User Config** содержатся в библиотеке **VisuUserMgmtDialogs**. Обе библиотеки расположены в папке установки CODESYS по пути **…\CODESYS\Projects\Visu\Dialogs**).

Для редактирования диалога следует:

1. Отрыть соответствующую библиотеку в CODESYS.
2. Отредактировать диалога нужным образом.

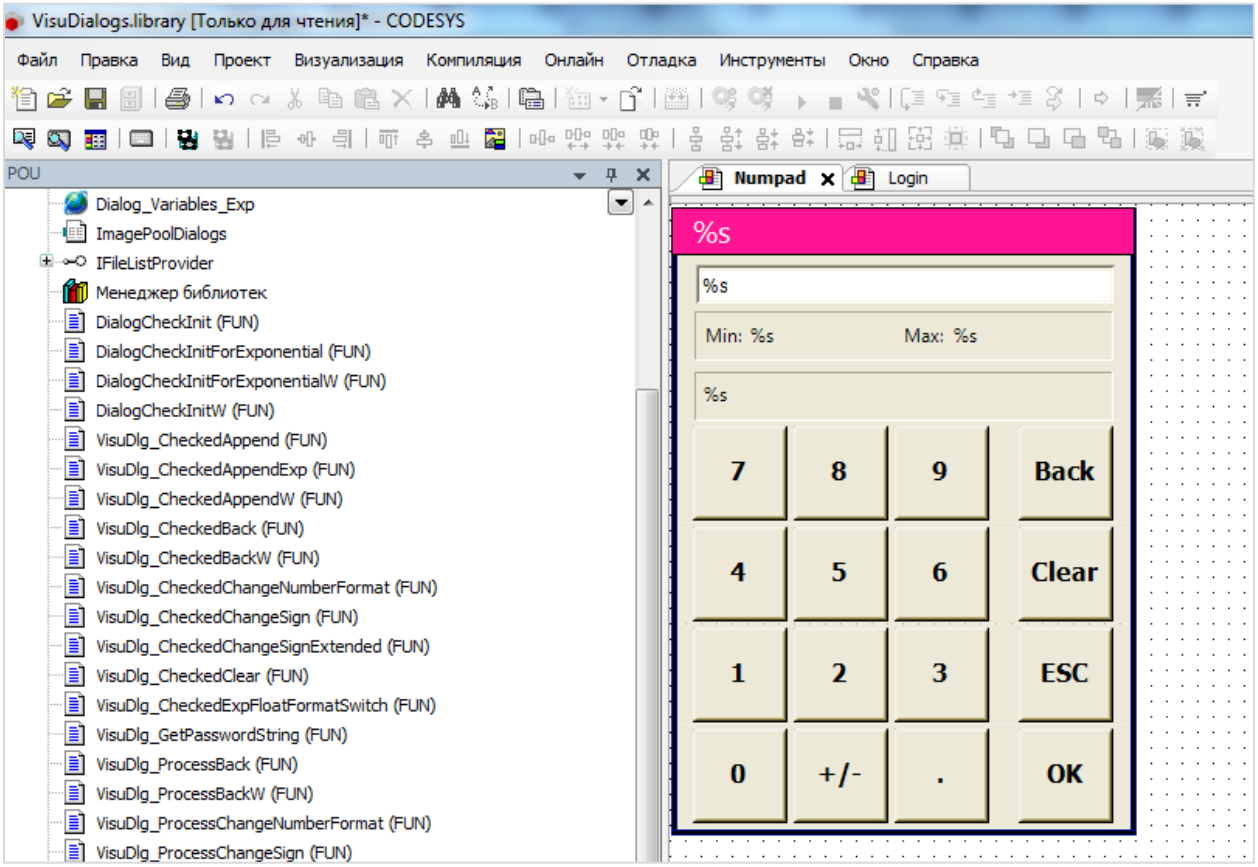


Рисунок 10.465 – Редактирование библиотеки диалогов

ПРИМЕЧАНИЕ

Рекомендуется не редактировать код библиотеки – это может привести к некорректной работе диалогов.

3. В узле **Информация о проекте** на вкладке **Общее** изменить заголовок и снять галочку **Выпущен**.

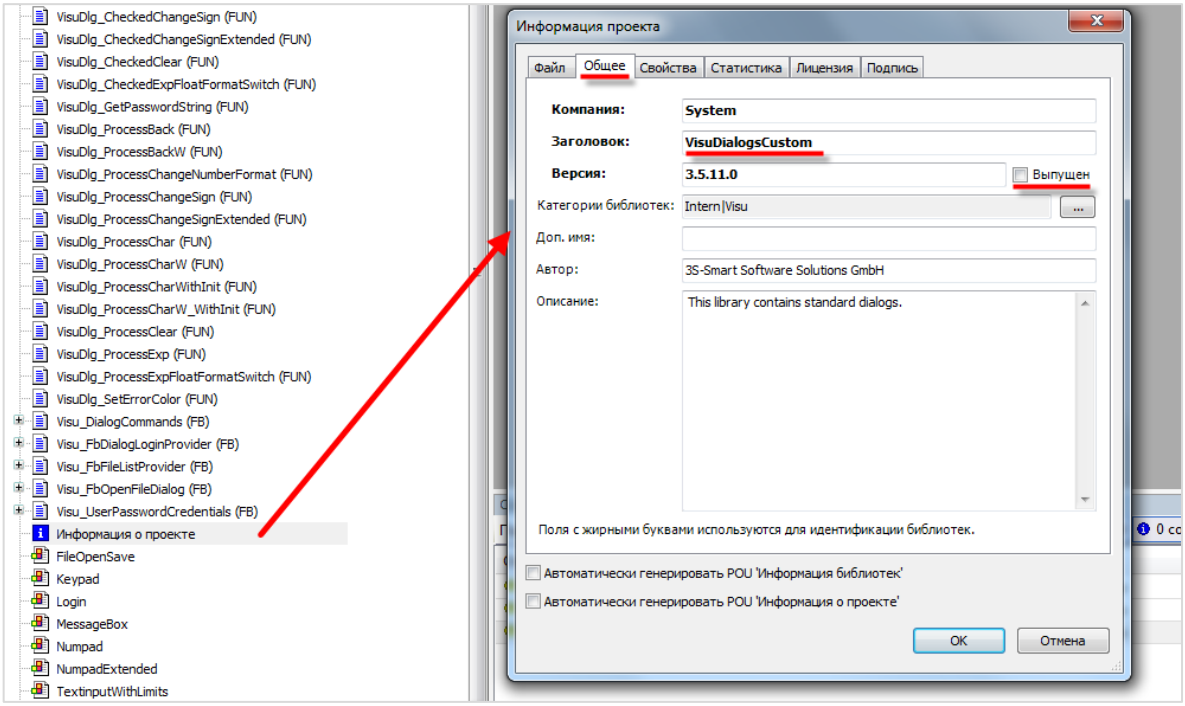


Рисунок 10.466 – Изменение заголовка библиотеки

На вкладке **Свойства** изменить ключи **Placeholder** и **Title**.

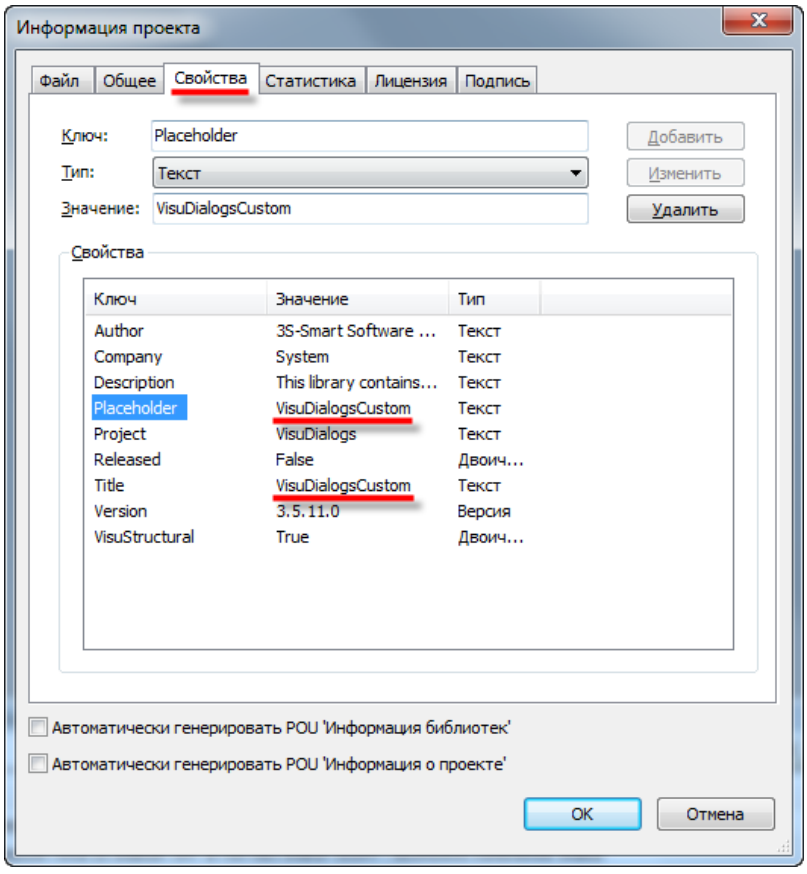


Рисунок 10.467 – Изменение ключей библиотеки

4. Пересохранить библиотеку под новым названием.
5. Установить библиотеку в репозиторий.

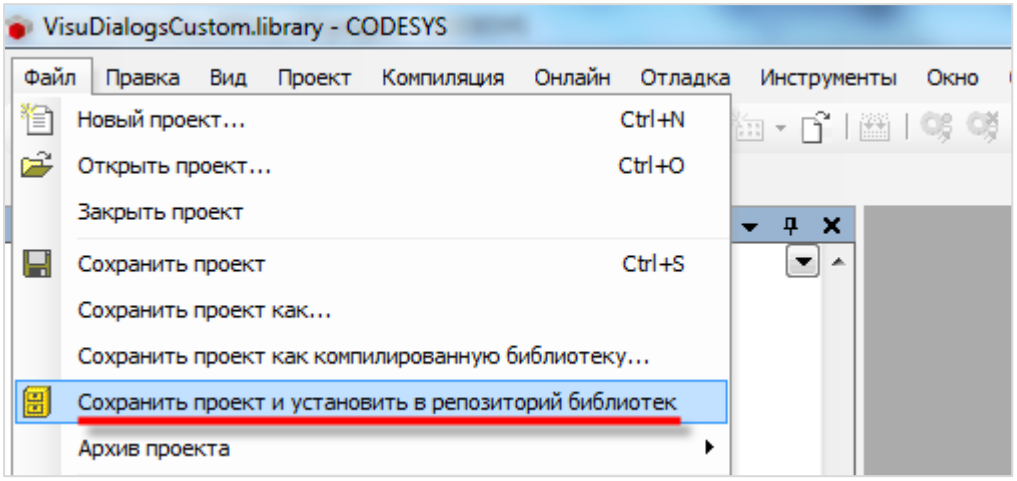


Рисунок 10.468 – Установка библиотеки в репозиторий

6. Создать новый проект **CODESYS**.

7. Добавить в проект созданную библиотеку.

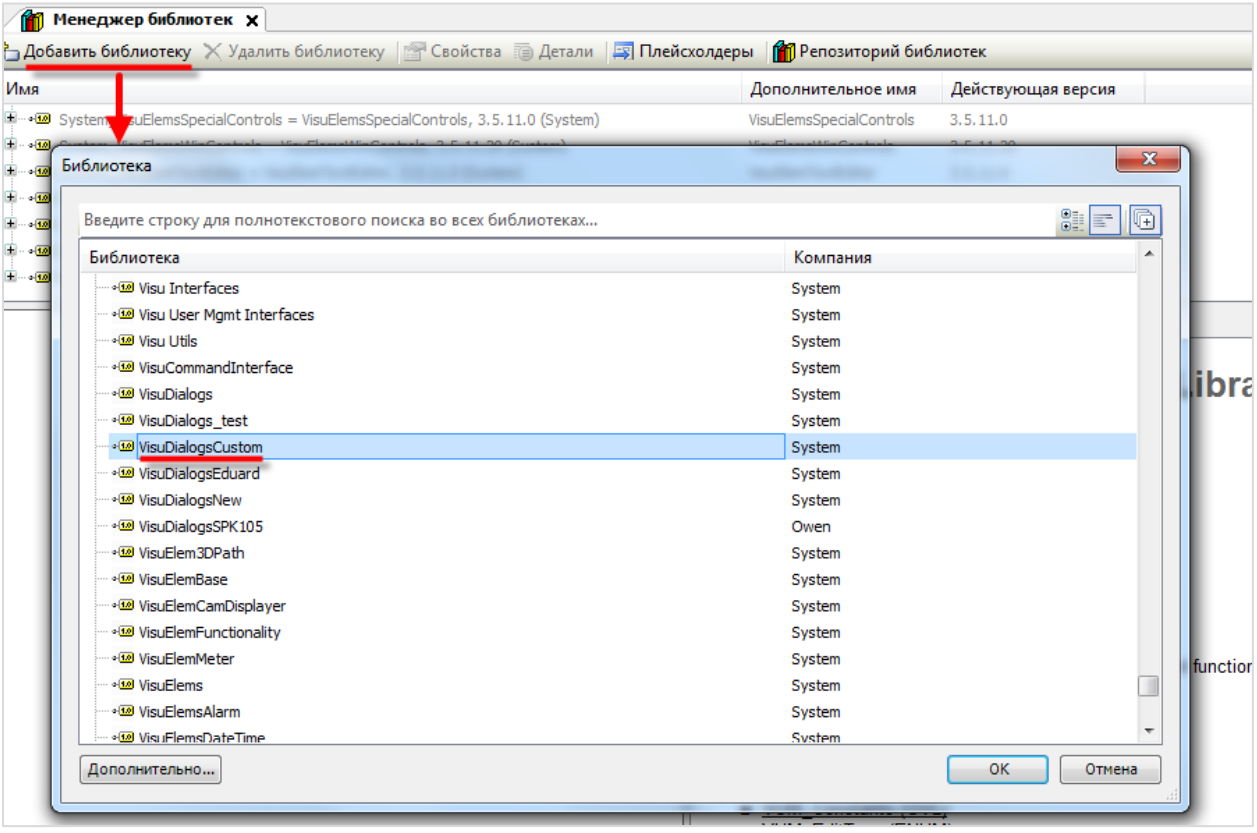


Рисунок 10.469 – Добавление библиотеки в проект CODESYS

8. Выбрать диалог библиотеки в Менеджере визуализации или настройках действия Записать переменную.

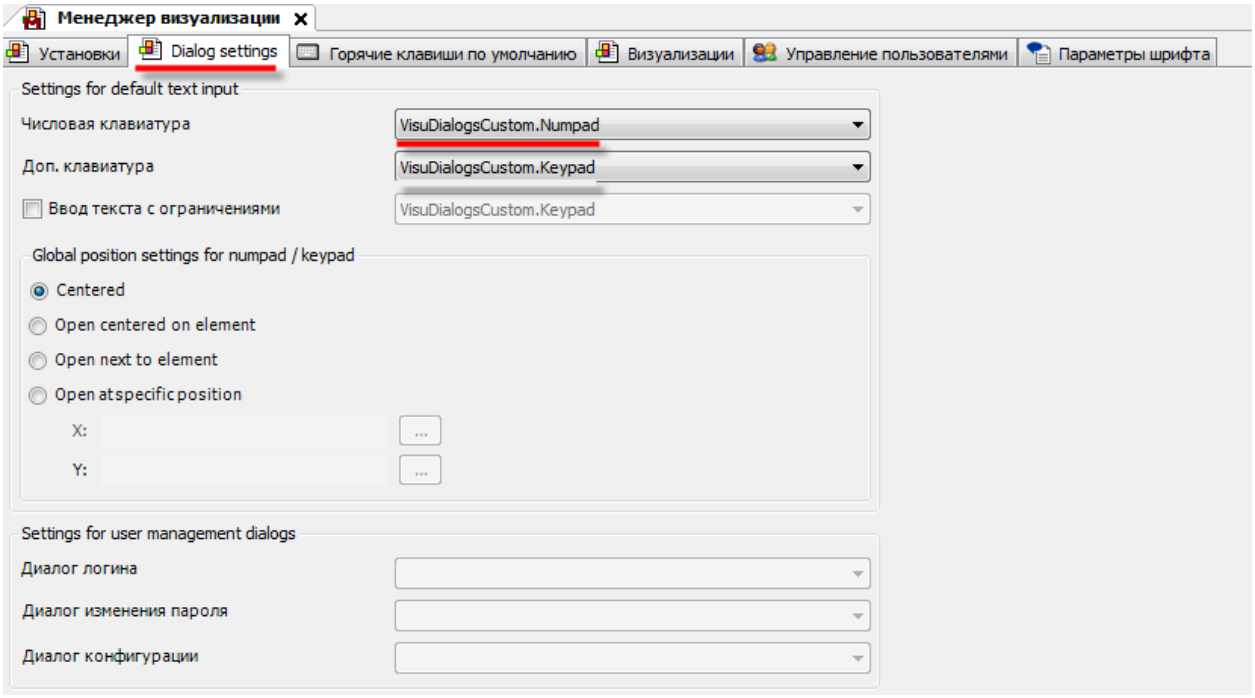


Рисунок 10.470 – Выбор диалога в проекте CODESYS

Созданная в примере библиотека доступна для скачивания: VisuDialogsCustom.library

10.4.7 Дополнительные примеры от компании CODESYS Group

Компания **CODESYS Group** (разработчик CODESYS) предоставляет примеры, демонстрирующие работу со специфичным функционалом визуализации:

1. VisuEventHandler.package – программная обработка действий пользователя в визуализации.
2. VisuValueChangeHandler.package – программная обработка ввода новых значений в визуализации.
3. EscImageButton.package – дополнительный графический примитив (кнопка с 5 состояниями).
4. MnControlsLight.package – дополнительный графический примитив (джойстик).

Примеры работы со специфическими возможностями интерфейсов экранов визуализации приведены в справке CODESYS.

10.4.8 Очистка истории трендов и тревог из кода программы

Начиная с версии **CODESYS V3.5 SP17 Patch 3** поддерживается очистка трендов и тревог из кода программы. Ниже приведены соответствующие примеры кода.

```
// очистка истории тренда из кода программы
VAR
   itfTrendRecording         : ITrendRecording;
   itfTrendStorageWriter     : ITrendStorageWriter;
   itfTrendStorageWriter3    : ITrendStorageWriter3;
   // название записи тренда в дереве проекта
   sTrendRecordingName       : STRING := 'Visualization_Trend1';

   xClearTrendHistory        : BOOL;
ENV_VAR

IF xClearTrendHistory THEN

    itfTrendRecording :=
GlobalInstances.g_TrendRecordingManager.FindTrendRecording(ADR(sTrendRecordingName));
    IF itfTrendRecording <> 0 THEN
        itfTrendStorageWriter := itfTrendRecording.GetTrendStorageWriter();
        IF __QUERYINTERFACE(itfTrendStorageWriter, itfTrendStorageWriter3) THEN
            itfTrendStorageWriter3.ClearHistory();
        END_IF
    END_IF

        xClearTrendHistory := FALSE;

END_IF
```

```
// очистка истории тревог из кода программы
VAR
   itfAlarmConfiguration  : AlarmManager.IAlarmConfiguration;
   itfAlarmConfiguration7 : AlarmManager.IAlarmConfiguration7;
END_VAR

FOR i := 0 TO AlarmGlobals.g_AlarmHandler.AlarmConfigurationCount - 1 DO
        itfAlarmConfiguration := AlarmGlobals.g_AlarmHandler.GetAlarmConfiguration(i);
        IF __QUERYINTERFACE(itfAlarmConfiguration, itfAlarmConfiguration7) THEN
                // сохраняем в журнале текущие тревоги
                itfAlarmConfiguration7.ClearHistory(xKeepActiveAlarms := TRUE);
        END_IF
END_FOR
```

11 Визуализация контроллеров ОВЕН: ограничения и рекомендации

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 контроллеров ОВЕН.

11.1 Размер экранов визуализации

Размер экранов визуализации должен соответствовать разрешению контроллеров.

Для контроллеров **СПК1xx, СПК1xx [M01]** и **СПК207** разрешение экрана составляет **800 × 480**.

Для контроллеров **СПК110 [M01]**, выпускаемых начиная с **декабря 2022 года** (с гравировкой **Версия 2.6 WSVGA** на задней крышке) – **1024x600**.

Для контроллеров **СПК105** – **480 × 272**.

Размер задается для таргет-визуализации и в свойствах каждого экрана визуализации проекта.

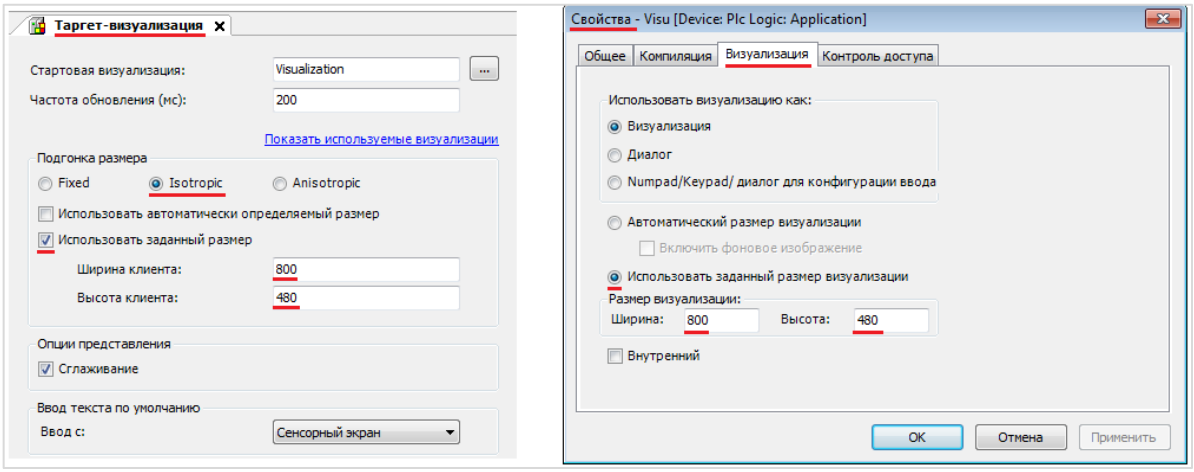


Рисунок 11.1 – Настройка размеров таргет-визуализации и экрана визуализации Visu

11.2 Отображение кириллического текста в визуализации

Для отображения кириллического текста следует использовать переменные типа **WSTRING**, предварительно поставив в Менеджере визуализации галочку **Использовать строки Unicode**:

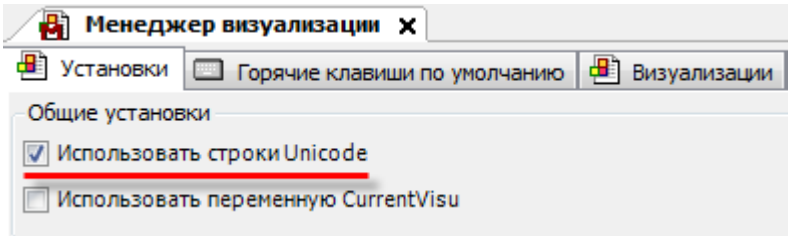


Рисунок 11.2 – Установка галочки Использовать строки Unicode в Менеджере визуализации

11.3 Тип ввода по умолчанию

Так как контроллеры **СПК** не имеют встроенной аппаратной клавиатуры, то пользовательский ввод в большинстве случаев должен осуществляться с помощью экранной клавиатуры. Для этого следует в настройках таргет-визуализации установить по умолчанию тип ввода **Сенсорной экран**:

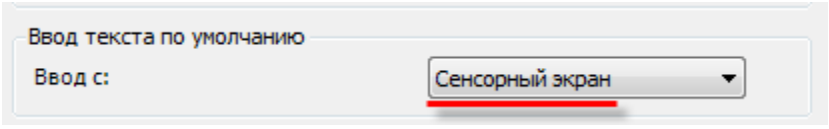


Рисунок 11.3 – Настройка типа ввода по умолчанию для target-визуализации

Настройка не требуется, если к СПК подключен USB HID клавиатура.

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):

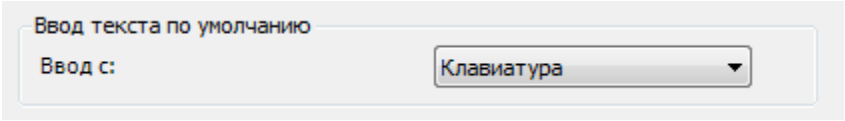


Рисунок 11.4 – Настройка типа ввода по умолчанию для web-визуализации

Также для каждого элемента во время настройки действия Записать переменную можно выбрать произвольный тип ввода:

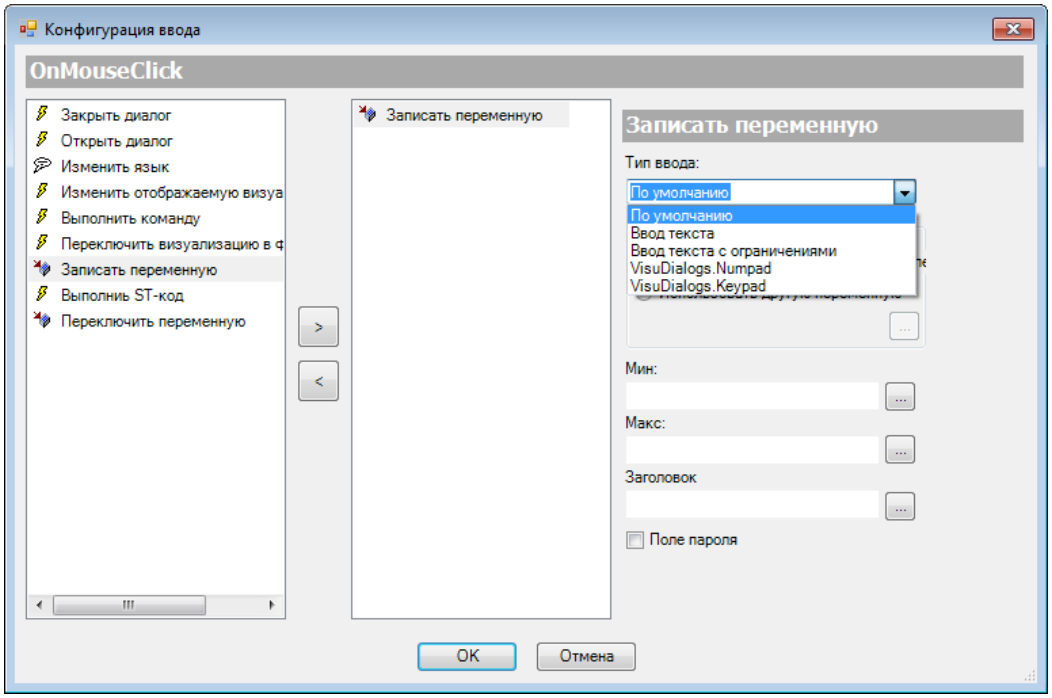


Рисунок 11.5 – Настройка типа ввода для действия Запись переменную

11.4 Ограничения

Элементы ActiveX и Путь 3D не поддерживаются контроллерами ОВЕН.

Элемент WebBrowser не поддерживается для таргет-визуализации контроллеров ОВЕН СПК (но поддерживается для web-визуализации).

Для контроллеров ОВЕН СПК не поддерживается опция Поддержка клиентских анимаций и наложение исходных элементов.

11.5 Поддерживаемые форматы графических файлов

Не рекомендуется использовать графические файлы с разрешением, которое превышает размер элемента в **CODESYS** (например, если ваша кнопка имеет размер **40 × 15**, то не стоит использовать в качестве подложки графический файл с разрешением **400 × 150**).

Поддерживаются графические файлы форматов .bmp, .jpg, jpeg, .png, .svg[2].

Обратите внимание, что формат .gif не поддерживается в таргет-визуализации.

11.6 Время цикла визуализации

Скорость отрисовки и отклика визуализации напрямую связана с временем цикла задачи **VISU_TASK**. Рекомендуется выставлять время цикла визуализации в пределах **100…200 мс**.

См. также эту статью.

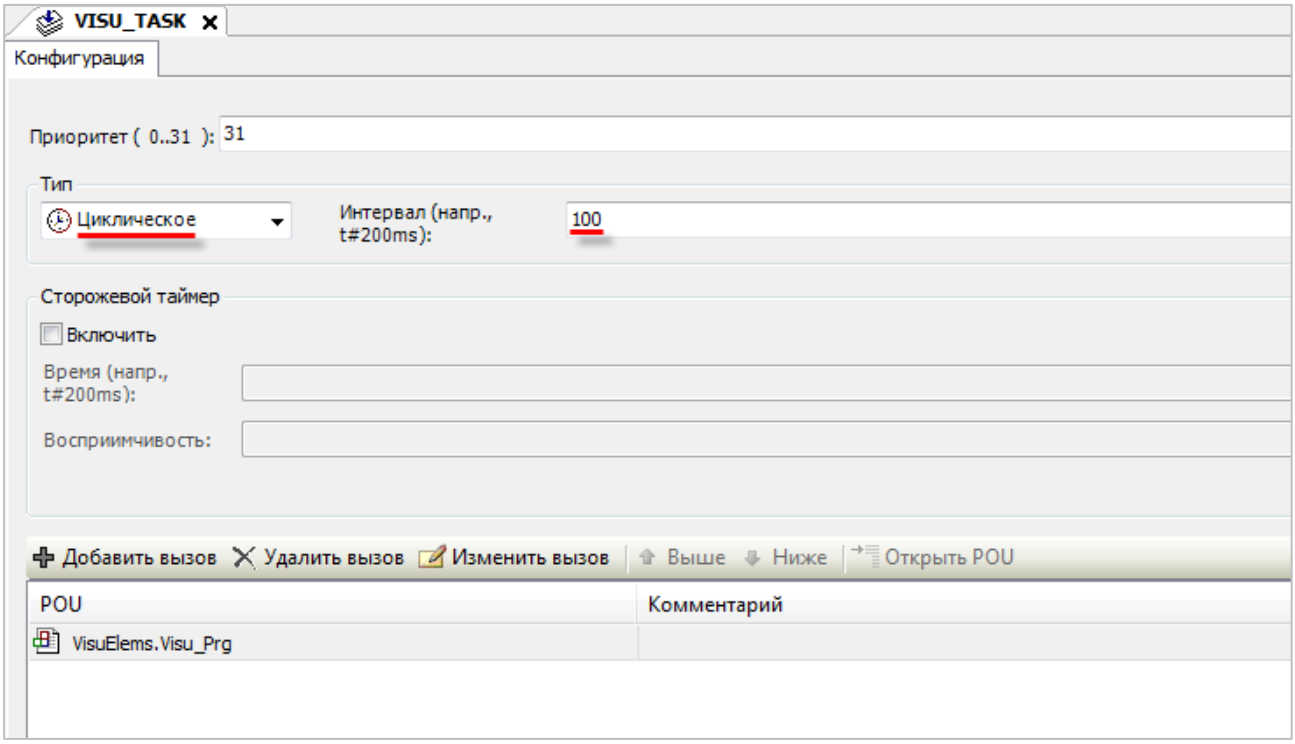


Рисунок 11.7 – Настройки времени цикла задачи визуализации

[2] Требования: svg version="1.2", baseProfile="tiny", правило **Только использованные глифы**

11.7 Поддерживаемые шрифты для контроллеров СПК

Контроллеры ОВЕН СПК поддерживают **исключительно** следующий набор шрифтов, для которых гарантируется соответствие отображения на дисплее контроллера/в редакторе **CODESYS**/в web-визуализации и кириллического начертания:

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
Segoe UI	Обычный, наклонный, полужирный, полужирный наклонный

Для установки шрифтов на ПК достаточно установить пакет таргет-файлов ОВЕН.

Начиная с прошивки **2.4.xxxx.xxxx** поддерживаются дополнительные шрифты (например, различные варианты шрифтов **Roboto** и **Open Sans**). Они не входят в пакет таргет-файлов; требуется установить их с помощью отдельного инсталлятора, загрузив его по ссылке или из web-конфигуратора контроллера на вкладке **ПЛК/Загрузки**.

Чтобы изменить шрифт элемента в визуализации, во вкладке **Свойства текста** следует выбрать параметр **Шрифт**:

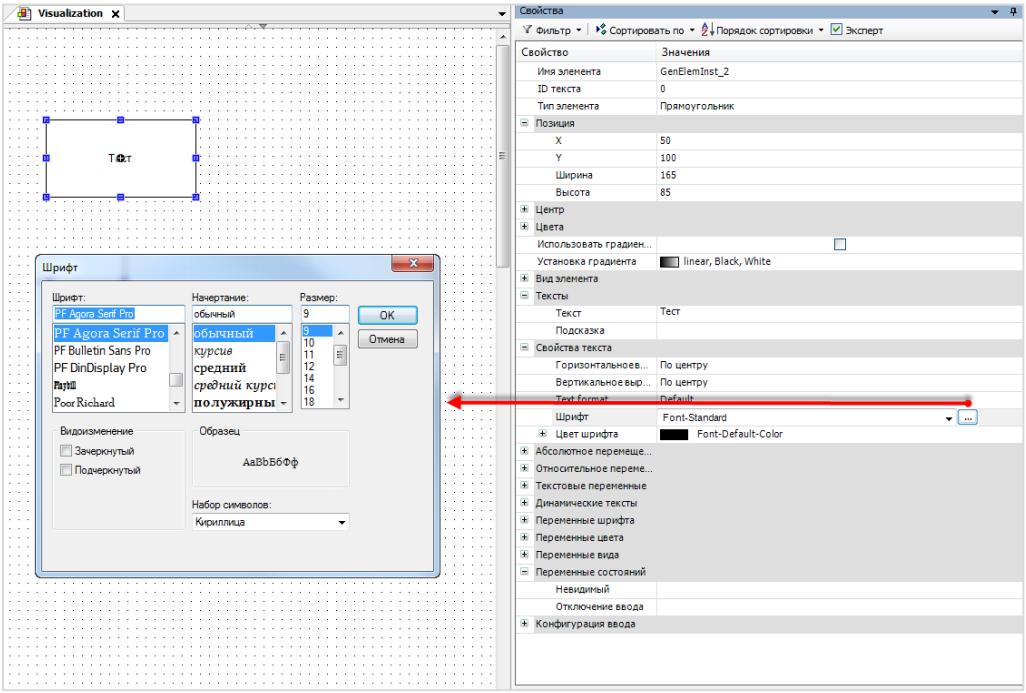


Рисунок 11.8 – Выбор шрифта элемента

Чтобы не выбирать шрифт для каждого элемента визуализации, можно создать свой стиль с выбранными шрифтами по умолчанию. Для этого следует перейти в Менеджер визуализации и открыть **Редактор стилей**:

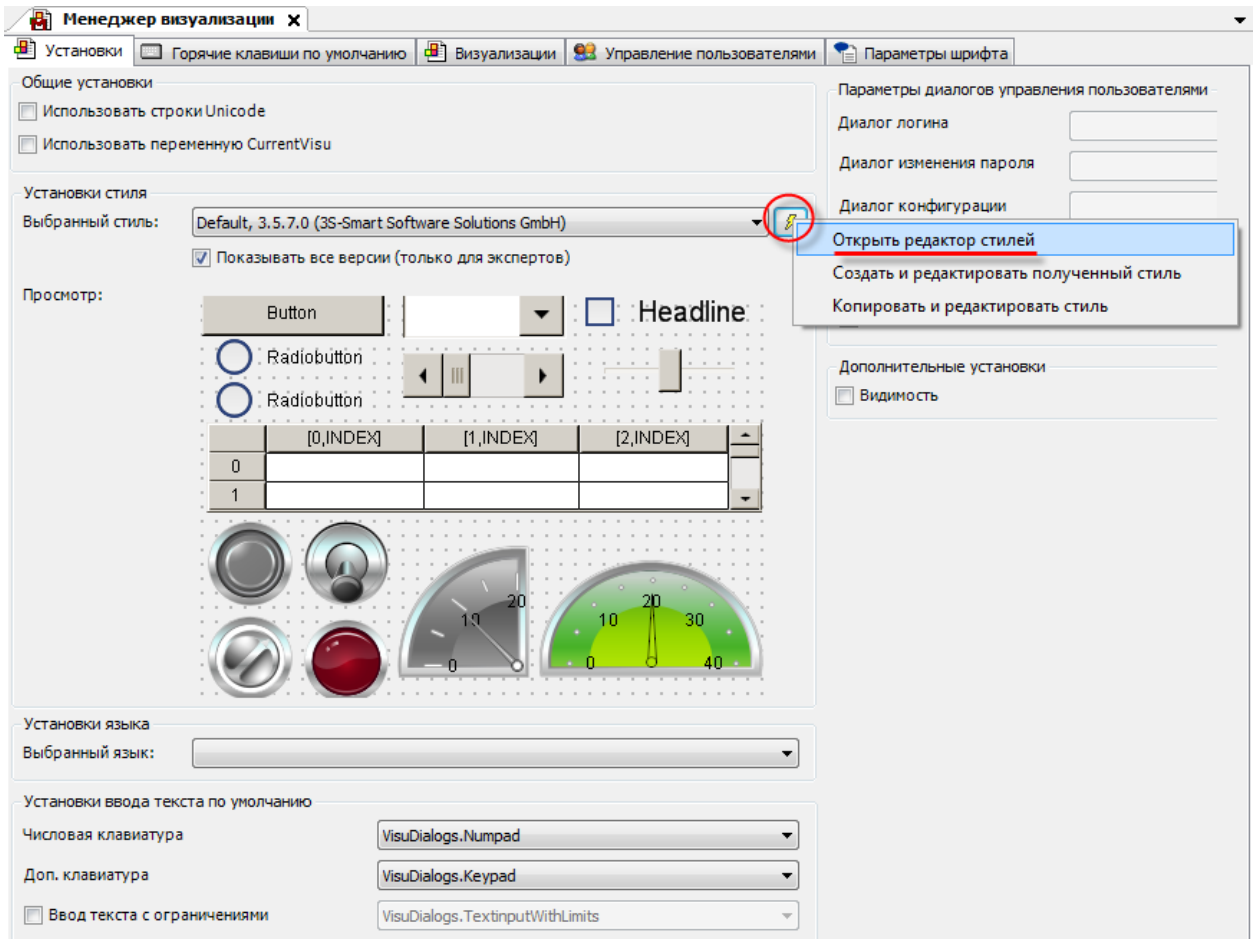


Рисунок 11.9 – Запуск редактора стилей

В меню **File** следует выбрать команду **Open as copy**, чтобы создать свой стиль на основе одного из существующих. Затем указать папку, в которой он будет сохранен (***рекомендуется*** сохранять каждый стиль в отдельную папку).

ПРИМЕЧАНИЕ

Данная папка должна существовать в ОС, и у пользователя должны быть права на ее изменение.

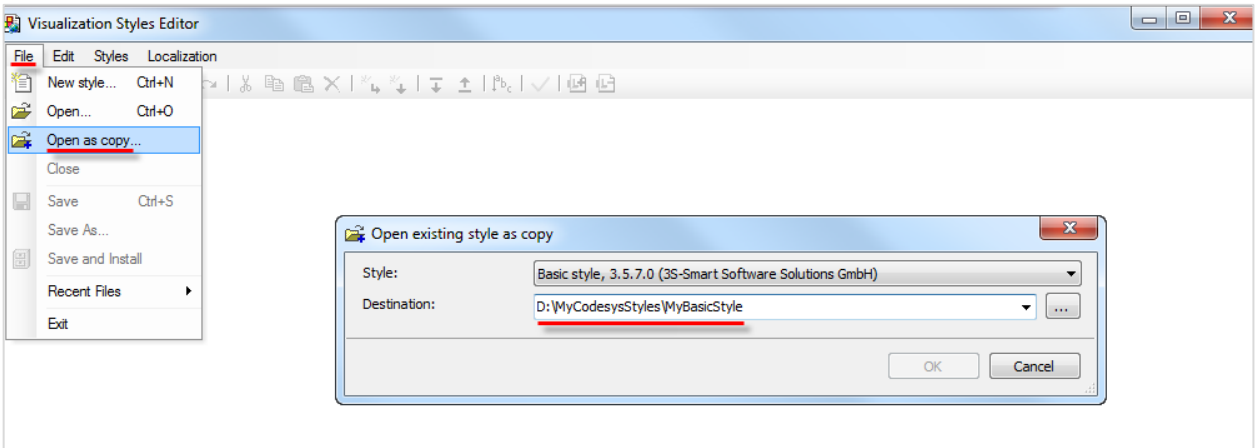


Рисунок 11.10 – Создание нового стиля на базе существующего

На вкладке **General** следует указать название стиля и его версию:

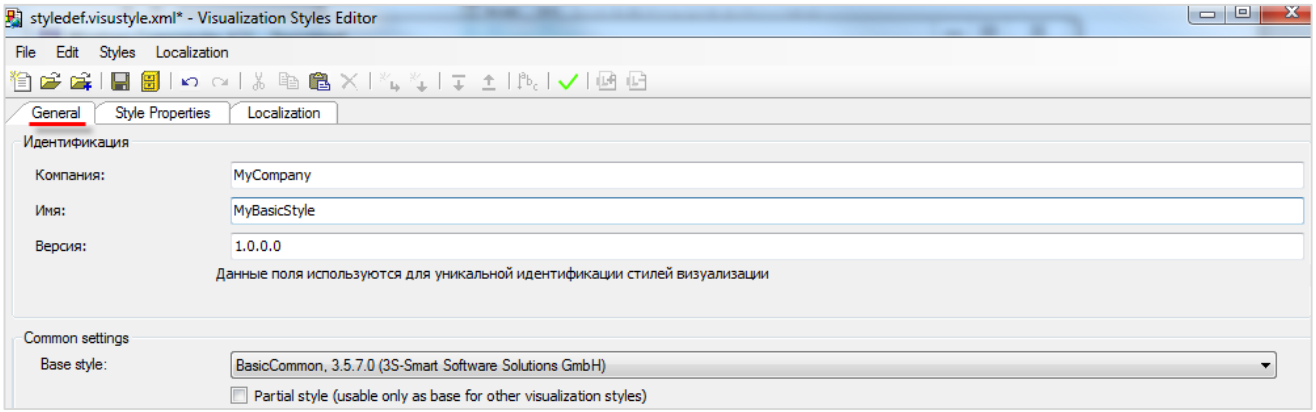


Рисунок 11.11 – Настройки вкладки General

На вкладке **Style Properties** следует выбрать шрифты по умолчанию (из числа тех, что поддерживаются СПК):

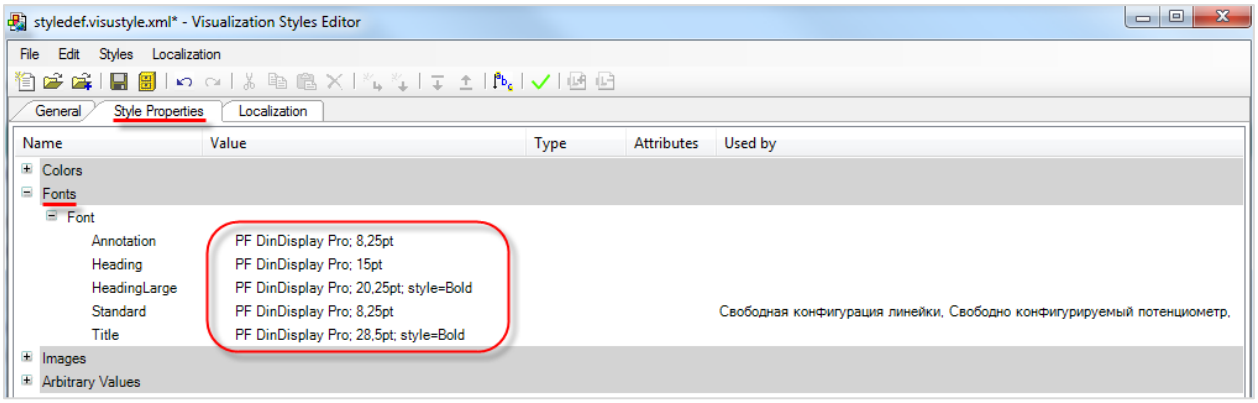


Рисунок 11.12 – Настройки вкладки Style Properties

В меню **File** следует выбрать команду **Save and Install**, чтобы сохранить стиль в репозитории.

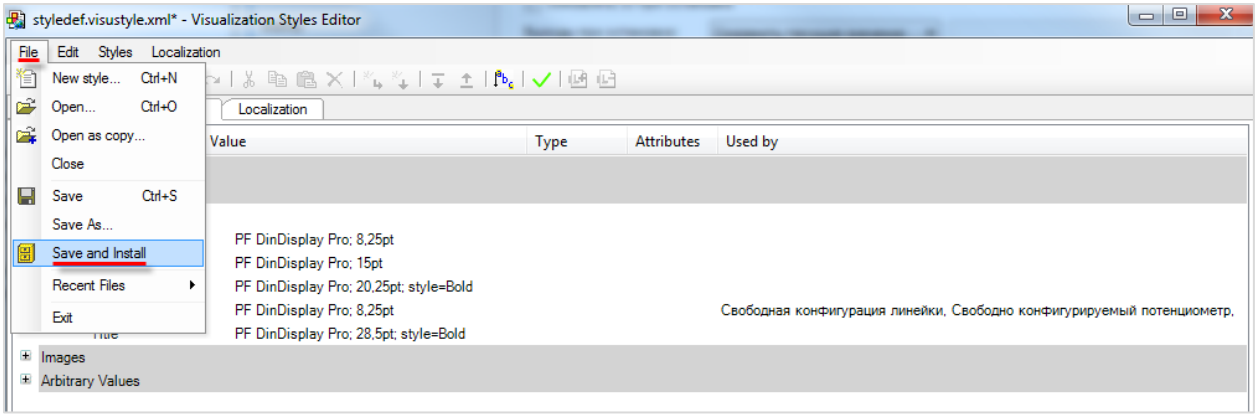


Рисунок 11.13 – Сохранение стиля визуализации

Затем следует перезапустить **CODESYS**. После этого можно будет выбрать сохраненный стиль в **Менеджере визуализации**:

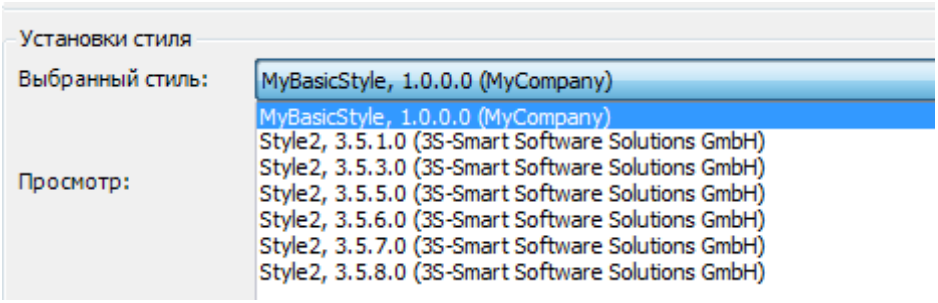


Рисунок 11.14 – Выбор стиля в Менеджере визуализации

11.8 Управление яркостью подсветки для контроллеров СПК

Управление яркостью подсветки дисплея контроллеров СПК осуществляется через узел **Screen** таргет-файла. См. описание узла в руководстве **CODESYS V3.5. Описание таргет-файлов**, которое доступно на сайте ОВЕН в разделе CODESYS V3/Документация.

На вкладке **Конфигурация** можно задать настройки управления подсветкой – их уже нельзя будет изменить в процессе работы проекта. Если же требуется изменять их в процессе работы – то следует установить для параметра **Parameters are used** значение **FALSE** и привязать переменные к нужным каналам вкладки **Соотнесение входов-выходов**.

Рисунок 11.15 – Настройки конфигурации в узле Screen

Кроме того, узел предоставляет системные переменные для обработки нажатий в таргет-визуализации:

- **Screen.cursorX** (INT) – координата последнего нажатия курсора по оси X;
- **Screen.cursorY** (INT) – координата последнего нажатия курсора по оси Y;
- **Screen.countTouch** (UDINT) – число нажатий на экран с момента включения;
- **Screen.uiRotateAngle** (UINT) – угол поворота экрана (0/90/180/270).

11.9 Изменение загрузочного логотипа для контроллеров СПК

Изменить загрузочный логотип для контроллеров СПК можно в web-конфигураторе на вкладке **ПЛК/Заставка**. Описание вкладки приведено в документе **Изменение логотипов СПК1xx [M01]**, доступном на странице прибора на сайте ОВЕН.

11.10 Сохранение истории трендов и трендов на USB/SD-накопитель

Начиная с версии прошивки **2.x.xxxx.xxxx** в web-конфигураторе на вкладке **ПЛК/Настройки** доступна группа параметров **Тренды и тревоги**:

- **Каталог трендов** – директория USB- или SD-накопителя, в которой будут сохраняться файлы трендов;
- **Каталог тревог** – директория USB- или SD-накопителя, в которой будут сохраняться файлы тревог;
- **Объем памяти, используемый memsys5, Мб** – начальный объем памяти, выделенный приложению **sqlite** (которое используется для реализации сохранения трендов и тревог). Если в проекте множество трендов и/или таблиц тревог, то в некоторых из них из-за недостатка памяти могут не отображаться данлогы. В этом случае значение данного параметра следует увеличить;
- **Период диагностики, мс** – период вывода в журнал контроллера диагностической информации (класса **Debug**) о памяти, используемой приложением **sqlite** (см. рисунок 7.30).

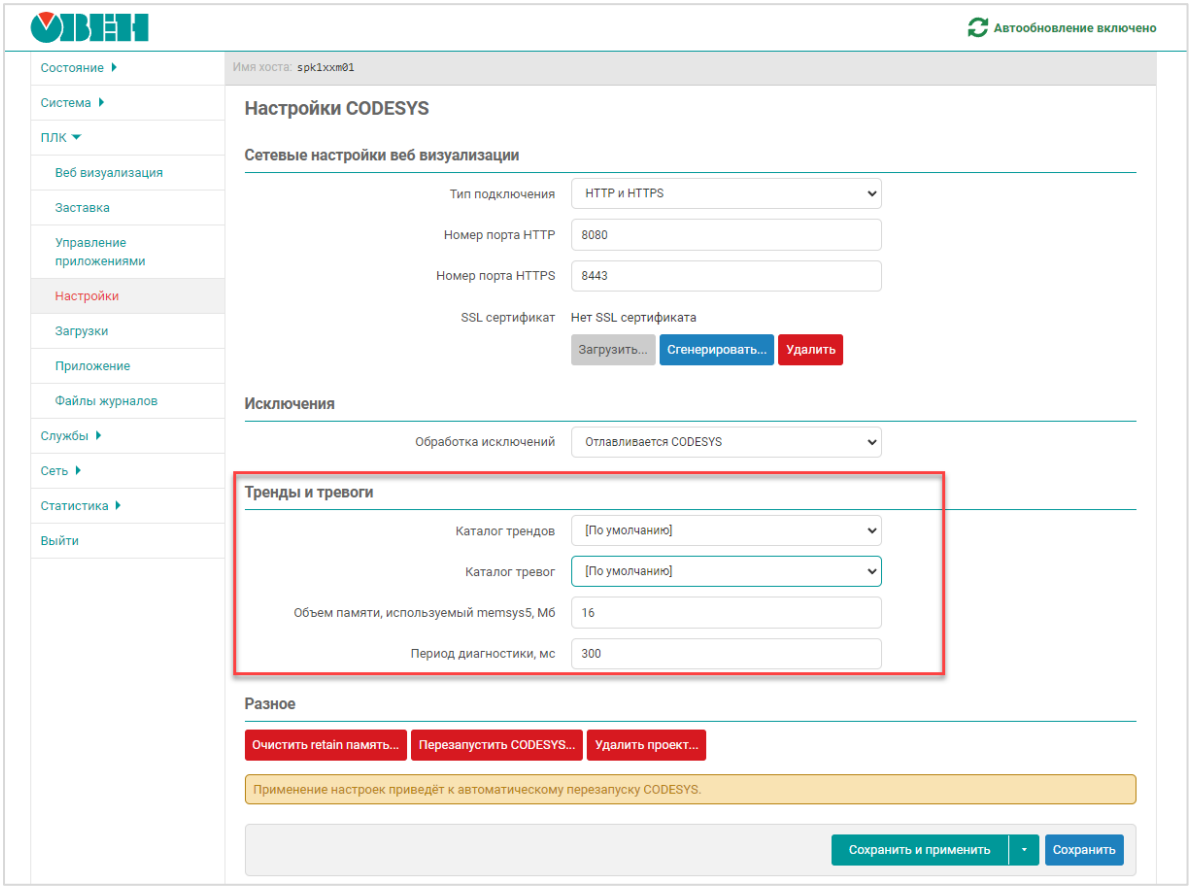


Рисунок 11.16 – Настройки трендов и тревог в web-конфигураторе

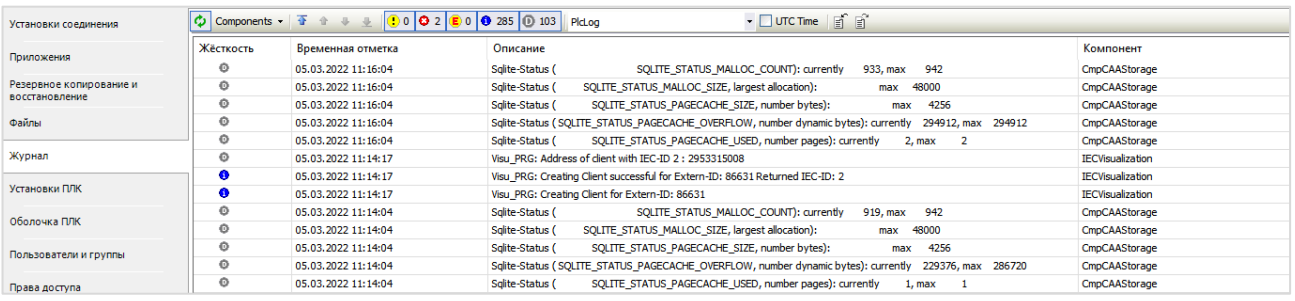


Рисунок 11.17 – Диагностика памяти sqlite в журнале CODESYS

Сохранение файлов трендов и тревог на накопителе позволяет снизить использование ресурса перезаписи внутренней flash-памяти контроллера, но увеличивает время доступа к этим файлам (например, загрузка исторических данных тренда в этом случае будет происходить медленнее).

В случае извлечения накопителя в процессе работы контроллера в элементе визуализации будет отображаться ошибка **Some kind of disk I/O error occurred** (см. рисунок 7.31). Для восстановления работы элемента следует подключить накопитель и перезагрузить контроллер.

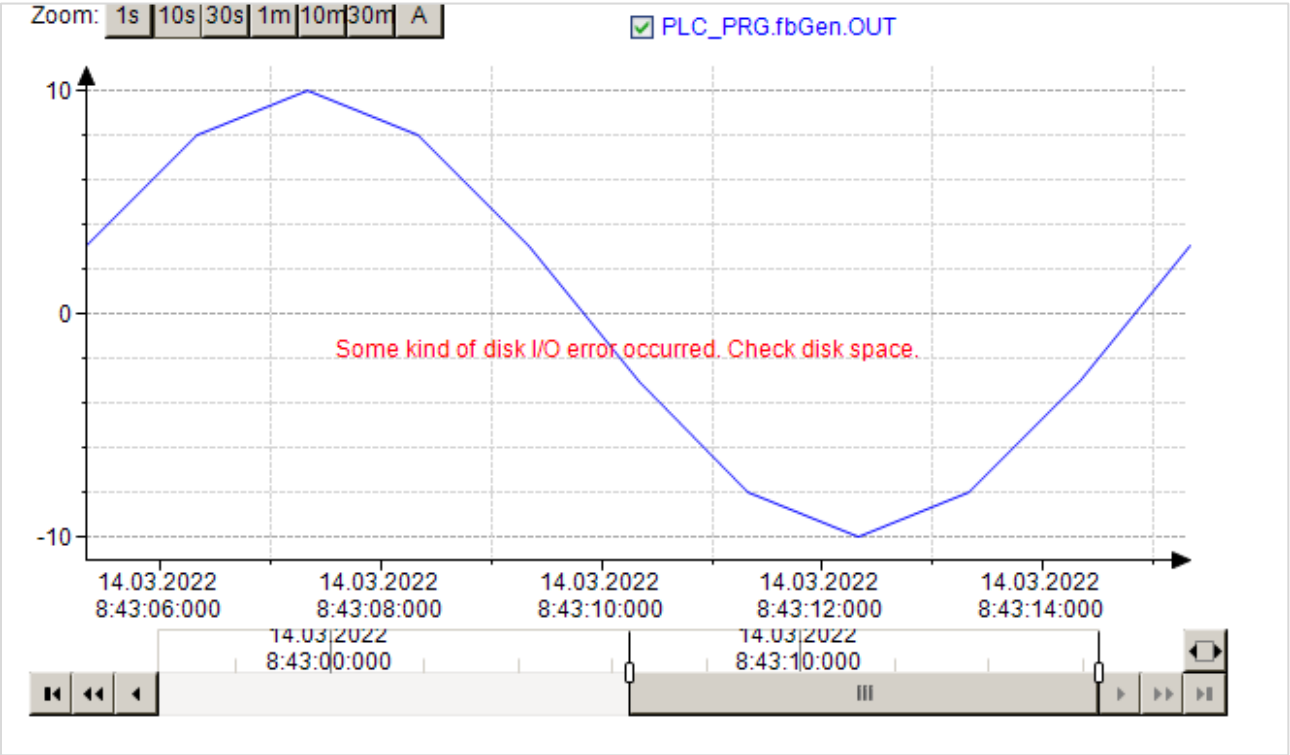


Рисунок 11.18 – Ошибка извлечения накопителя, на который производится сохранение файлов трендов