

АННОТАЦИЯ


Настоящая часть руководства оператора содержит описание редактора пользовательского интерфейса ПП «СКАДА А-СОФТ» (далее по тексту СКАДА) и сведения о библиотеках графических элементов.

СОДЕРЖАНИЕ

1	Редактор пользовательского интерфейса	4
2	События, их обработка и карты событий	18
3	Сигнализация.....	22
3.1	Формирование сигнала и получение его средой визуализации.....	23
3.2	Квитирование.....	23
4	Управление правами	25
5	Библиотеки графических элементов	26
5.1	Библиотека «Базовые виджеты» (originals)	28
5.1.1	Примитив элементарная фигура (ElFigure)	29
5.1.2	Примитив текста (Text)	31
5.1.3	Примитив элементы формы (FormEl)	32
5.1.4	Примитив отображения медиа-материалов (Media).....	37
5.1.5	Примитив построения диаграмм/графиков (Diagram)	39
5.1.6	Примитив формирования протокола (Protocol)	43
5.1.7	Примитив формирования отчётной документации (Document).....	45
5.1.8	Примитив контейнера (Box)	47
5.1.9	Примитив поверхность (Surface).....	48
5.2	Графический редактор для виджетов, основанных на примитиве элементарная фигура.....	51
5.3	Библиотека основных элементов пользовательского интерфейса (Main)	53
5.4	Библиотека «Элементы мнемосхемы» (mnEls)	60
5.4.1	Элементы трубопровода.....	61
5.4.2	Элементы, изображающие различные технологические устройства	62
5.4.3	Другие элементы	65
5.5	Библиотека электроэлементов мнемосхем пользовательского интерфейса (ElectroEls).....	67
5.5.1	Динамические элементы библиотеки	68
5.5.2	Статические элементы библиотеки «Электроэлементы»	70
5.6	Элементы библиотеки «NT-tmp».....	71
	Перечень принятых сокращений	73

1 Редактор пользовательского интерфейса

Разработка пользовательского интерфейса в ПП «СКАДА А-СОФТ» выполняется в одном окне – «Редакторе пользовательского интерфейса», реализующем многодокументный интерфейс (MDI), позволяющий одновременно редактировать несколько кадров различных размеров. Для перехода в данное окно необходимо на панели

инструментов системного конфигуратора нажать на правую иконку  «Рабочий пользовательский интерфейс (Qt)». В появившемся окне доступны следующие механизмы управления разработкой: панели инструментов, пункты меню и контекстное меню. Большинство действий дублируются. Навигационные интерфейсы реализованы присоединяемыми окнами. Конфигурация панелей инструментов и присоединяемых окон сохраняется при выходе и восстанавливается при старте системы.

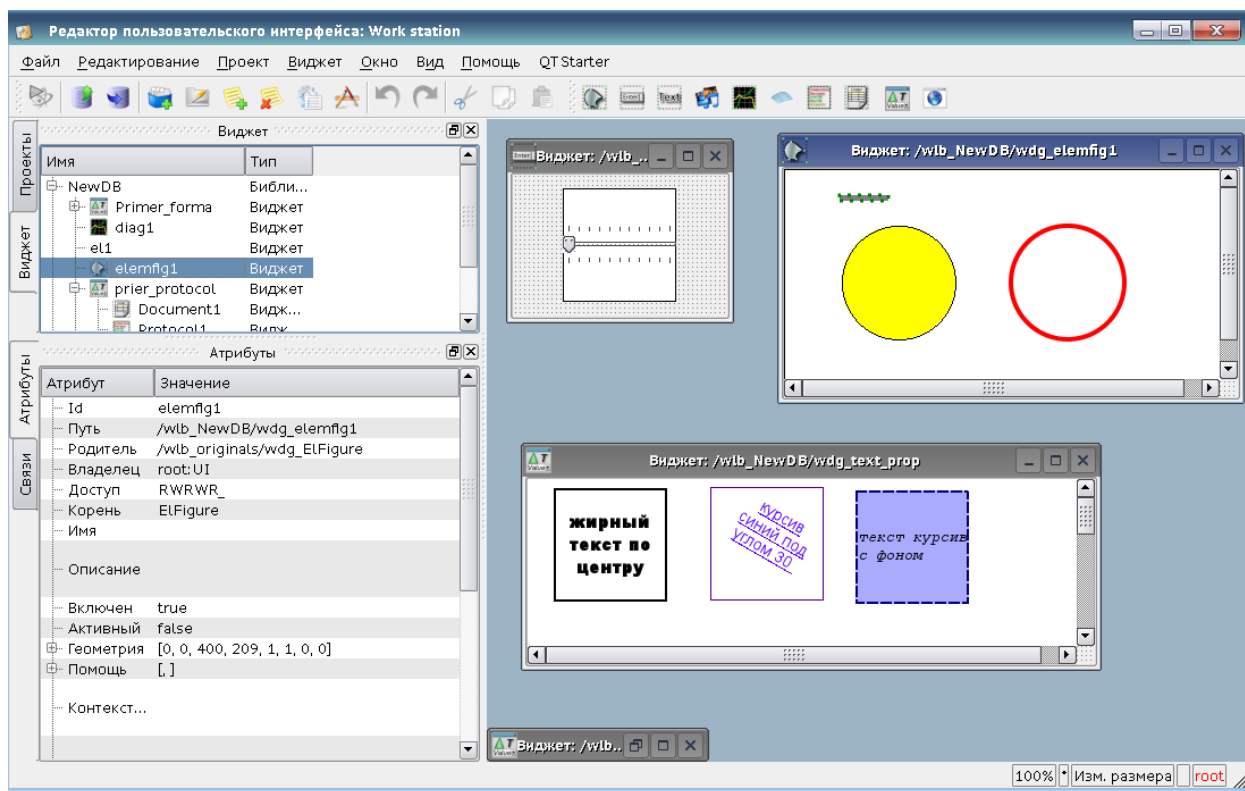


Рисунок 1

Доступ к основным компонентам СВУ производится посредством закрепленных окон:

- *Виджет* – реализовано в виде дерева библиотек виджетов, позволяет быстро найти нужный виджет или библиотеку и провести их редактирование. Содержит следующее контекстное меню:

- "Новая библиотека" – создание новой библиотеки;

- "Добавить визуальный элемент" – добавление визуального элемента в библиотеку;
- "Удалить визуальный элемент" – удаление визуального элемента из библиотеки;
- "Очистить изменения визуального элемента" – очистка визуального элемента с наследованием изменённых свойств или установкой их по умолчанию;
- "Свойства визуального элемента" – конфигурация визуального элемента;
- "Редактировать визуальный элемент" – визуальное редактирование элемента;
- "Найти использования визуального элемента" - в нижней части экрана отображает окно с результатом поиска выделенного визуального элемента во всех библиотеках;
- "Перейти к родительскому визуальному элементу" – переход к элементу, на основании которого он сделан;
- "Вырезать визуальный элемент" – вырезание/перемещение визуального элемента;
- "Копировать визуальный элемент" – копирование визуального элемента;
- "Вставить визуальный элемент" – вставка визуального элемента;
- "Загрузить из БД" – загрузка данных визуального элемента из БД;
- "Сохранить в БД" – сохранение данных визуального элемента в БД;
- "Обновить библиотеки" – выполняет перечитывание конфигурации и состава библиотек из модели данных.

- *Проект* – реализовано в виде дерева страниц проектов. Для формирования пользовательских видеокладов достаточно разместить в окне проекта элементы из библиотек виджетов. Содержит следующее контекстное меню:

- "*Запустить исполнение проекта*" – запуск исполнения выбранного проекта;
- "Новый проект" – создание нового проекта;
- "Добавить визуальный элемент" – добавление визуального элемента в проект/страницу;
- "Удалить визуальный элемент" – удаление визуального элемента из проекта/страницы;

- "Очистить изменения визуального элемента" – очистка визуального элемента с наследованием изменённых свойств или установкой их по умолчанию;
- "Свойства визуального элемента" – конфигурация визуального элемента;
- "Редактировать визуальный элемент" – визуальное редактирование элемента;
- "Найти использования визуального элемента" - в нижней части экрана отображает окно с результатом поиска выделенного визуального элемента во всех библиотеках;
- "Перейти к родительскому визуальному элементу" – переход к элементу, на основании которого он сделан;
- "Библиотека: {Имя библиотеки}" – пункты меню для доступа к видеокадрам/виджетам содержащимся в библиотеке;
- "Вырезать визуальный элемент" – вырезание/перемещение визуального элемента;
- "Копировать визуальный элемент" – копирование визуального элемента;
- "Вставить визуальный элемент" – вставка визуального элемента;
- "Загрузить из БД" – загрузка данных визуального элемента из БД;
- "Сохранить в БД" – сохранение данных визуального элемента в БД;
- "Создание альтернативного представления" – создание альтернативной страницы для отображения на оборудовании, выполняющем разные роли (например, коллективное табло отображения, рабочее место оператора или встроенная сенсорная панель);
- "Удаление альтернативного представления" – удаление выбранного альтернативного представления страницы;
- "Выбор альтернативного представления" – выбирает из существующих альтернативную страницу;
- "Обновить проекты" – выполняет перечитывание конфигурации и состава проектов из модели данных.

В основном пространстве рабочего окна размещаются окна страниц проектов, видеокадров библиотек виджетов, пользовательских элементов и элементов примитивов на момент их визуального редактирования.

В меню рабочего стола размещены все инструменты, необходимые для разработки интерфейсов СВУ. Меню имеет следующую структуру:

- *"Файл"* – общие операции:
 - "Загрузить из БД" – загрузка данных визуального элемента из БД;
 - "Сохранить в БД" – сохранение данных визуального элемента в БД;
 - "Закрыть" – закрыть окно редактора;
 - "Выход" – выход из системы СКАДА;
- *"Редактирование"* – операции редактирования визуальных элементов:
 - "Откат изменений визуального элемента" – осуществление отката последнего изменения визуального элемента;
 - "Повтор изменений визуального элемента" – осуществление повтора изменения визуального элемента;
 - "Вырезать визуальный элемент" – вырезание/перемещение визуального элемента в момент вставки;
 - "Копировать визуальный элемент" – копирование визуального элемента в момент вставки;
 - "Вставить визуальный элемент" – вставка визуального элемента;
- *"Проект"* – операции над проектами:
 - "Запустить исполнение проекта" – запуск исполнения выбранного проекта;
 - "Новый проект" – создание нового проекта;
 - "Добавить визуальный элемент" – добавление визуального элемента в проект;
 - "Удалить визуальный элемент" – удаление визуального элемента из проекта;
 - "Очистить изменения визуального элемента" – очистка визуального элемента с наследованием изменённых свойств или установкой их по умолчанию;
 - "Создание альтернативного представления" – создание альтернативной страницы для отображения на оборудовании, выполняющем разные роли (например, коллективное табло отображения, рабочее место оператора или встроенная сенсорная панель);
 - "Удаление альтернативного представления" – удаление выбранного альтернативного представления страницы;
 - "Выбор альтернативного представления" – выбирает из существующих альтернативную страницу;
 - "Свойства визуального элемента" – конфигурация визуального элемента;
 - "Редактировать визуальный элемент" – визуальное редактирование элемента;

- "Найти использования визуального элемента" - в нижней части экрана отображает окно с результатом поиска выделенного визуального элемента во всех библиотеках;
- "Перейти к родительскому визуальному элементу" – переход к элементу, на основании которого он сделан.
- *"Виджет"* – операции над виджетами и библиотеками виджетов:
 - "Новая библиотека" – создание новой библиотеки;
 - "Добавить визуальный элемент" – добавление визуального элемента в библиотеку;
 - "Удалить визуальный элемент" – удаление визуального элемента из библиотеки;
 - "Очистить изменения визуального элемента" – очистка визуального элемента с наследованием изменённых свойств или установкой их по умолчанию;
 - "Свойства визуального элемента" – конфигурация визуального элемента;
 - "Редактировать визуальный элемент" – визуальное редактирование элемента;
 - "Найти использования визуального элемента" - в нижней части экрана отображает окно с результатом поиска выделенного визуального элемента во всех библиотеках;
 - "Перейти к родительскому визуальному элементу" – переход к элементу, на основании которого он сделан;
 - "Вид" – управление расположением визуальных элементов на ивдеокадрах:
 - "Виджет вверх" – поднятие виджета на самый верх;
 - "Виджет вниз" – опускание виджета на самый низ;
 - "Поднять виджет" – поднять виджет выше;
 - "Опустить виджет" – опустить виджет ниже;
 - "Выравнять слева" – выравнивание виджета слева;
 - "Выравнять по центру вертикально" – выравнивание виджета вертикально по центру;
 - "Выравнять справа" – выравнивание виджета справа;
 - "Выравнять сверху" – выравнивание виджета сверху;
 - "Выравнять по центру горизонтально" – выравнивание виджета горизонтально по центру;
 - "Выравнять снизу" – выравнивание виджета снизу;

- "Библиотека: {Имя библиотеки}" – пункты меню для доступа к видеокадрам/виджетам содержащимся в библиотеке;
- "Окно" – управление окнами MDI-интерфейса:
 - "Закрыть" – закрыть активное окно;
 - "Закрыть все" – закрыть все окна;
 - "Уложить" – уложить все окна для видимости одновременно;
 - "Каскадировать" – расположить все окна каскадом;
 - "Следующее" – активировать следующее окно;
 - "Предыдущее" – активировать предыдущее окно;
- "Вид" – управление отображением рабочего окна и панелей на нём:
 - "Весь экран" – отображение на весь экран;
 - "Панель визуальных элементов" – панель управления визуальными элементами;
 - "Функции видимости виджетов" – панель управления видимостью и расположением виджетов на панелях;
 - "Панель элементарных фигур" – дополнительная панель редактирования примитива элементарных фигур ("ElFigure");
 - "Результаты поиска"- окно отображения результата поиска визуальных элементов в библиотеках;
 - "Проекты" – закрепленное окно управления деревом проектов;
 - "Виджет" – закрепленное окно управления деревом библиотек виджетов;
 - "Атрибуты" – закрепленное окно диспетчера атрибутов;
 - "Связи" – закрепленное окно диспетчера связей;
 - "Библиотека: {Имя библиотеки}" – управление видимостью панелей библиотек виджетов;
- "Помощь" – помощь по СКАДа и модулю Vision:
 - "Справка" – содержит справочную информацию о работе в программной платформе;

- "Про QT" – информация о библиотеке QT, используемой модулем;
- "Что это" – запрос описания элементов интерфейса окна;
- "*QT Starter*" – запуск UI модулей СКАДА системы:
 - системный конфигурактор (QT);
 - рабочий пользовательский интерфейс (QT).

Внизу окна разработки СВУ располагается строка статуса, в которой размещены индикаторы визуального масштаба редактируемого кадра, текущей раскладки клавиатуры, режима изменения размера элементов, режима текущей станции движка СВУ и пользователя, от имени которого ведётся разработка интерфейса СВУ. По двойному клику на индикаторе пользователя можно сменить пользователя, введя новое имя и пароль пользователя. В главное поле строки статуса выводятся различные информационные сообщения и сообщения помощи.

Для редактирования свойств визуальных элементов предусмотрено два диалога: редактирование свойств контейнеров визуальных элементов (библиотек виджетов и проектов) и свойств самих визуальных элементов. Диалоги вызываются из контекстного меню визуального элемента. Изменения, внесённые в диалогах, сразу же попадают в движок СВУ. В части 3 руководства оператора на примере разобран порядок создания, сохранения и редактирования свойств визуальных элементов.

На рисунке 2 представлено окно свойств контейнера визуальных элементов, при вызове его для проекта становятся доступными вкладки «Стили», «Журнал сообщений», «Строка состояния» и «Диагностика».

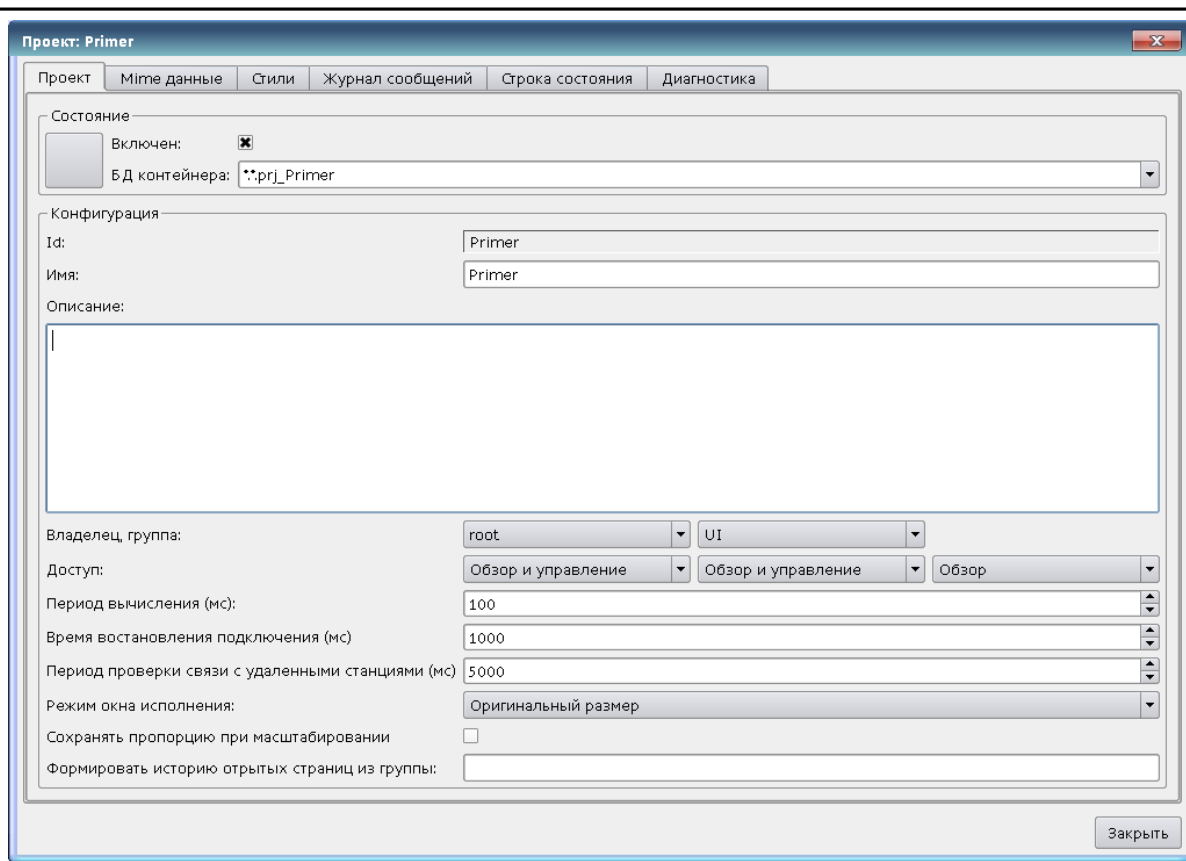


Рисунок 2

С помощью главной вкладки этого окна можно установить:

- состояние контейнера элементов, а именно: "Включен" и БД контейнера;
- идентификатор, имя и описание контейнера;
- пользователя, группу пользователей и доступ пользователей;
- для проекта: период вычисления проекта, время восстановления подключения, период проверки связи с удаленными станциями, режим открытия окна при исполнении, формирование истории открытых страниц из группы и флаг сохранения пропорций при масштабировании.

На вкладке «Строка состояния» можно настроить перечень отображаемых в строке состояния элементов. На вкладке «Диагностика» отображаются текущие сообщения системы.

С помощью вкладки «Стили» может быть создано множество стилей, каждый из которых будет хранить цветовые, шрифтовые и другие свойства элементов кадра. Простая смена стиля позволит быстро преобразить интерфейс ВУ, а возможность назначения индивидуальной стили для пользователя позволит учесть его индивидуальные особенности.

Окно редактирования свойств визуального элемента виджет представлено на рисунке 3 и содержит следующие вкладки: «Виджет», «Атрибуты», «Обработка» и «Связи».

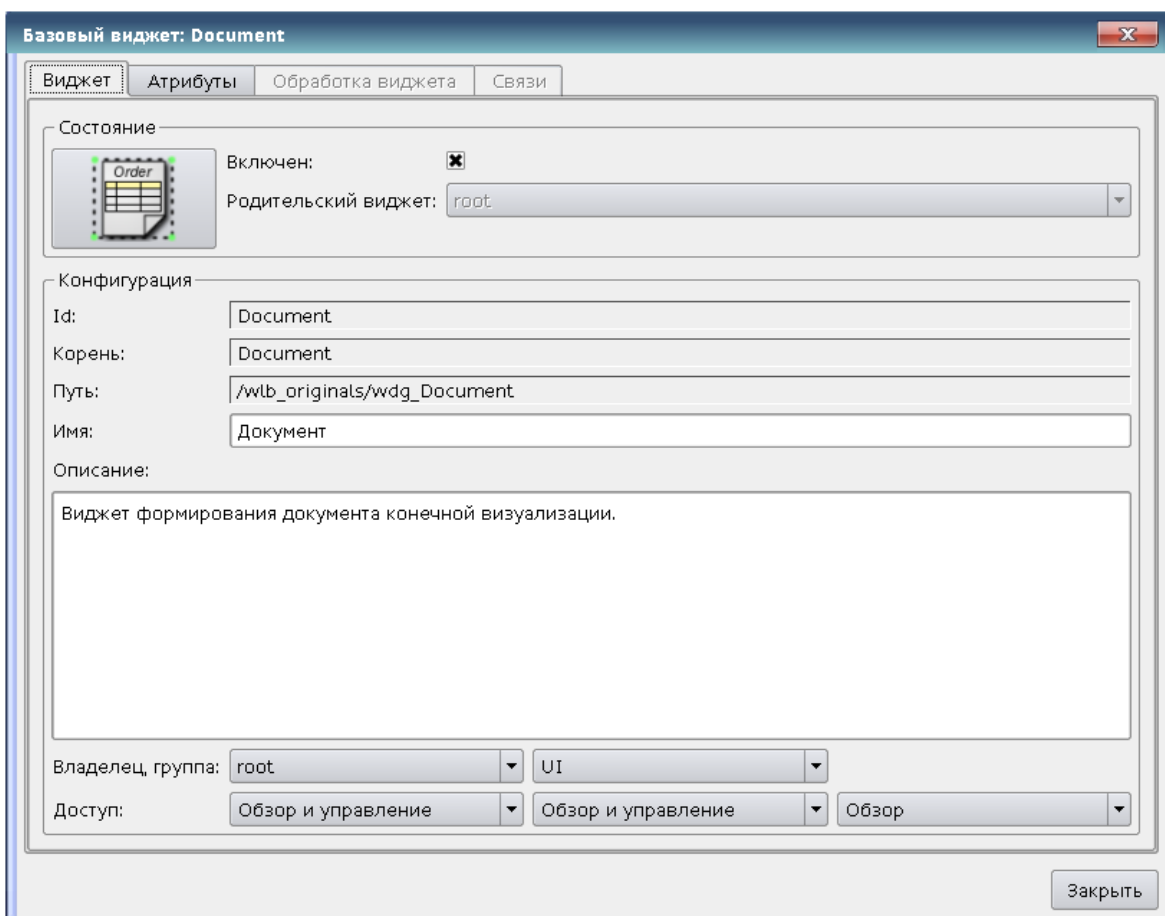


Рисунок 3

На вкладке «Виджет» можно установить:

- состояние элемента, а именно: "Включен" и родительский виджет;
- конфигурацию элемента: идентификатор, корень, путь, имя и описание элемента;
- пользователя, группу пользователей элемента и доступ пользователей.

Вкладка «Атрибуты» содержит набор стандартных и индивидуальных атрибутов элемента и позволяет провести их редактирование (совпадает с содержанием закрепленного окна «Атрибуты»).

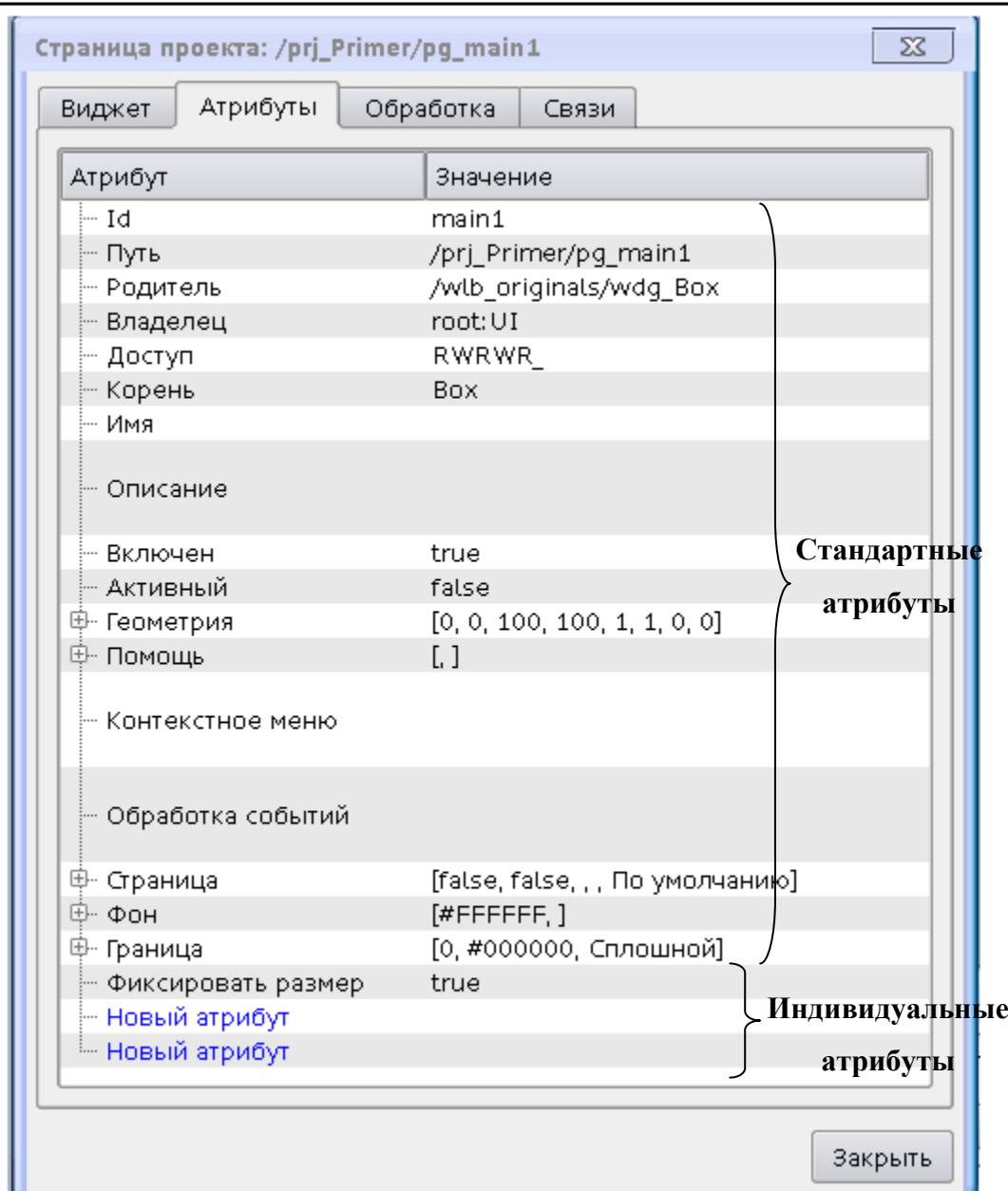


Рисунок 4

Стандартными атрибутами любого виджета являются:

"ID" (*id*) – идентификатор виджета;

"Путь" (*path*) – путь к виджету;

"Родитель" (*parent*) – путь к родительскому виджету;

"Владелец" (*owner*) – владелец и группа виджета в форме "[владелец]:[группа]" (по умолчанию "root:UI");

"Доступ" (*perm*) – права доступа к виджету в форме "[польз.][группа][другие]":

- _ – нет доступа;
- R_ – только чтение;
- RW – чтение и запись.

По умолчанию RWRWR_.

"Корень"(*root*) – идентификатор примитива, лежащего в основе виджета;

"Имя"(*name*) – название виджета;

"Описание" (*dscr*) – текстовое описание виджета;

"Включен"(*en*) – состояние виджета включен, отключенный виджет не отображается при исполнении;

"Активный"(*activ*) – состояние виджета. Активные элементы могут получать фокус при исполнении, а значит получать клавиатурные и иные события с последующей их обработкой;

"Геометрия"(*geom*) – размеры виджета;

"Помощь"(*tipTool,tipStatus*) – подсказка оператору и состояние виджета;

"Контекстное меню"(*contextMenu*) – контекстное меню в формате списка строк: "[ItName]:[Signal]", где ItName – имя элемента; Signal – имя сигнала, результирующее имя сигнала: "usr_[Signal]";

"Обработка событий"(*evProc*) – прямая обработка событий для управления страницами в формате: "[event]:[evSrc]:[com]:[prm]", где:

- event – ожидаемое событие;
- evSrc – источник события;
- com – команда сеанса:
 - open – открытие страницы. Открываемая страница указывается в параметре <prm> как напрямую, так и в виде шаблона (например: /pg_so/1/*/*);
 - next – открытие следующей страницы. Открываемая страница указывается в параметре <prm> в виде шаблона (например: /pg_so/*/*/\$);
 - prev – открытие предыдущей страницы. Открываемая страница указывается в параметре <prm> в виде шаблона (например: /pg_so/*/*/\$);
- prm – параметр команды, где используются:
 - pg_so – прямое имя желаемой страницы с префиксом. Требует обязательного соответствия и используется для идентификации предыдущей открытой страницы;
 - 1 – имя новой страницы в общем пути без префикса. Игнорируется при обнаружении предыдущей открытой страницы;
 - * – имя страницы берется из имени предыдущей страницы или подставляется первая доступная страница, если предыдущая открытая страница отсутствует;

- \$ – указывает на место, относительно которого открывается страница.

"Фон" – установка цвета виджета или изображения

"Граница" – оформление границы виджета (ширина линий, цвет, стиль)

Вкладка «Обработка виджета» обеспечивает описание конфигурации процесса формирования динамических связей и конфигурации динамики. Вкладка содержит таблицу конфигурации свойств атрибутов виджета и поле текста программы, для описания процедуры обработки виджета (рисунок 5).

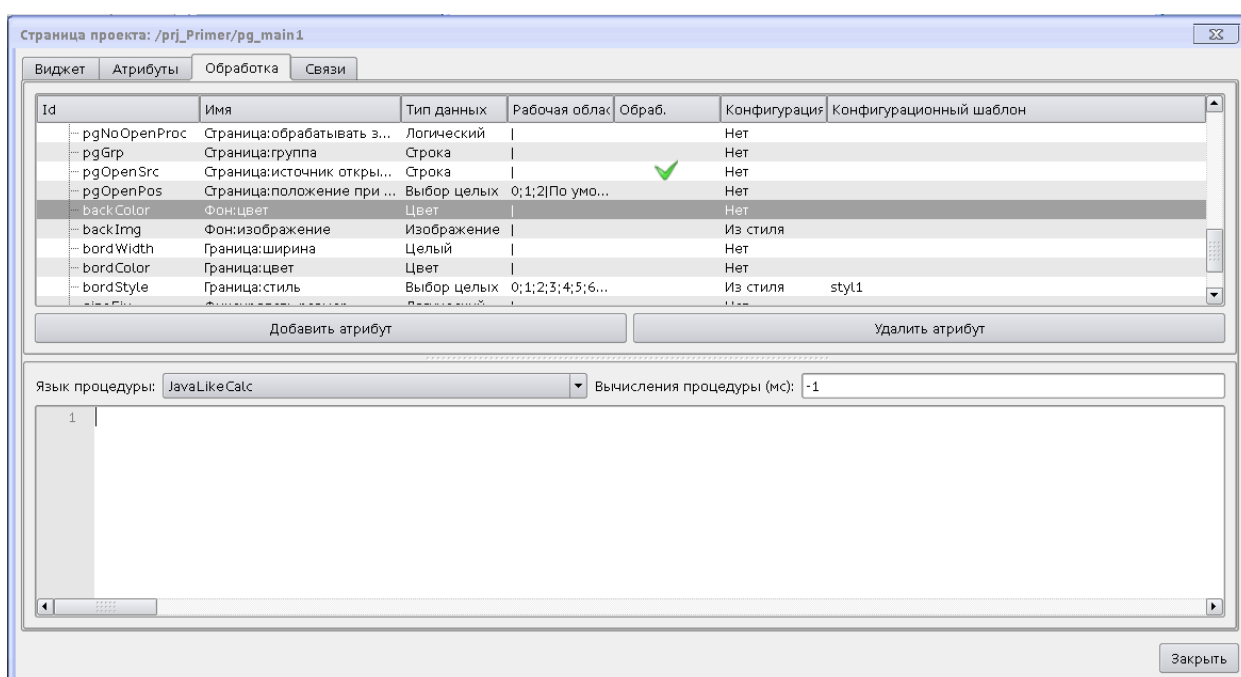


Рисунок 5

В данном окне возможно добавление, редактирование и удаление атрибутов видеокadra. Для каждого пользовательского атрибута можно изменить следующие параметры:

ID – идентификатор атрибута;

Имя – описание атрибута;

Тип данных – может принимать значения: логический, целый, вещественный, строка, объект, выбор целых, выбор вещественных, выбор строк, текст, цвет, изображение, шрифт, ДатаВремя, адрес;

Рабочая область – позволяет задать диапазон значений;

Обработка – установка в значение «True» позволяет использовать атрибут в вычислительной процедуре виджета;

Конфигурация – позволяет указать тип связи для атрибута и принимает значение: постоянная, входная связь, выходная связь, полная связь, из стиля, тревога;

Конфигурационный шаблон – позволяет описать группы динамических атрибутов (разные типы параметров подсистемы "DAQ"). Кроме того, при корректном формировании этого поля работает механизм автоматического назначения атрибутов при указании только параметра подсистемы "DAQ", что упрощает и ускоряет процесс конфигурации. Значение этой колонки имеет следующий формат: <Параметр> | <Идентификатор>, где:

<Параметр>- группа атрибута;

<Идентификатор> – идентификатор атрибута, именно это значение сопоставляется с атрибутами параметров DAQ при автоматическом связывании после указания групповой связи.

Например, для настройки стиля для атрибута цвет необходимо в поле «Конфигурация» этого атрибута выбрать «Из стиля», а в «Конфигурационном шаблоне» указать идентификатор стиля.

При выборе в поле «Конфигурация» для атрибута значения типа связи – постоянная, входная связь (связь с динамикой только для чтения), выходная связь (связь с динамикой только для записи) или полная связь (чтение и запись) – на вкладке «Связи» становится доступной настройка связи с динамикой (рисунок 6). При этом тип связи может принимать значение:

val: – прямая загрузка значения через механизм связей. Например, связь: "val:100" загружает в атрибут виджета значение 100. Часто используется в случае отсутствия конечной точки связи с целью прямой установки значения.

prm: – связь на атрибут параметра или параметр в целом, для группы атрибутов подсистемы "Сбор данных". Знак "(+)", в конце адреса, сигнализирует об успешной линковке и присутствии целевого объекта.

wdg: – связь на атрибут другого виджета или виджет в целом, для группы атрибутов.

Настройка связи осуществляется последовательным выбором значения для выбранного атрибута.

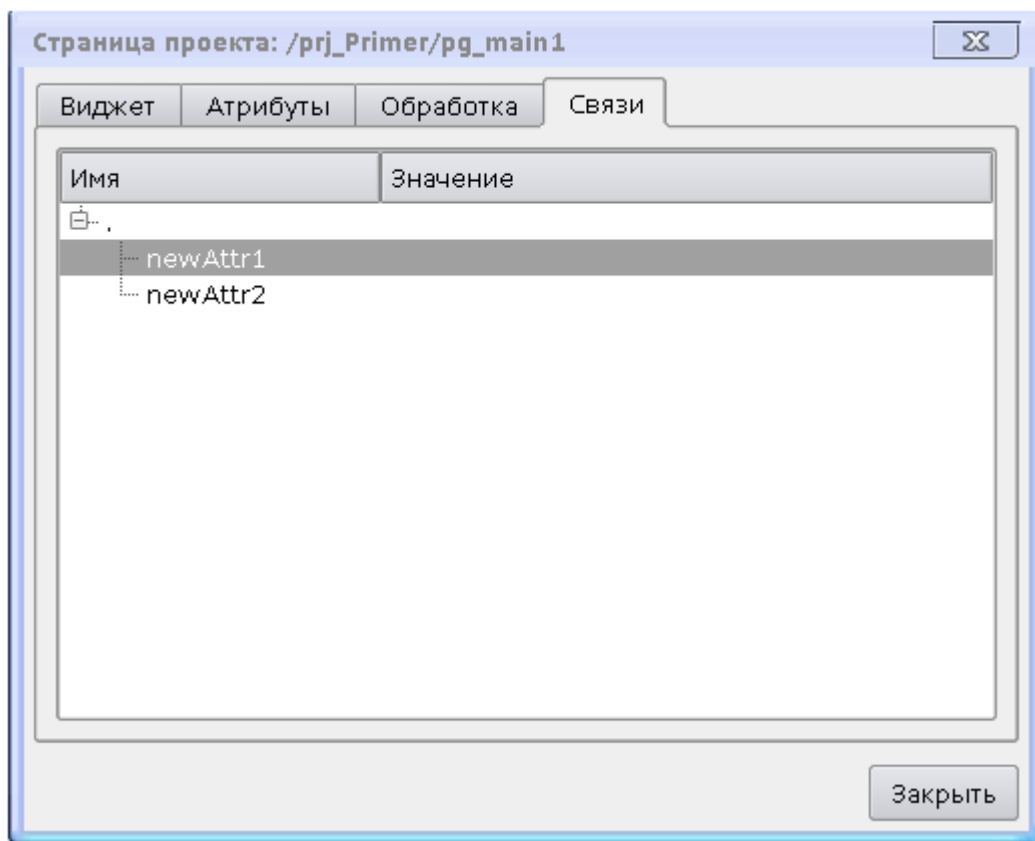


Рисунок 6

Обработка связей происходит с периодичностью вычисления видеокadra в следующем порядке:

- получение данных входных связей;
- выполнение вычисления скрипта;
- передача значений по выходным связям.

При размещении виджета, содержащего конфигурацию связей, в контейнер виджетов все связи исходного виджета добавляются в список результирующих связей контейнера виджетов.

Для создания кадров общего назначения с функцией предоставления детализированных данных разных источников одного типа предусмотрен механизм динамической установки связей посредством зарезервированного ключевого идентификатора "<page>" группы атрибутов связей у кадров общего назначения и динамическое назначение связей с идентификатором "<page>" в процессе открытия кадра общего назначения сигналом от другого виджета.

2 События, их обработка и карты событий

В ПП «СКАДА А-СОФТ» предусмотрен механизм управления интерактивными пользовательскими событиями.

Событие – это сообщение, которое возникает в различных точках исполняемого кода при выполнении определенных условий. События предназначены для того, чтобы иметь возможность предусмотреть реакцию программного обеспечения (например, открытие окна при нажатии пользователем на кнопку).

Менеджер событий должен работать, используя карты событий. Карта событий – это список именованных событий с указанием его происхождения. Происхождением события может быть клавиатура, манипулятор мыши, джойстик и т.д. При возникновении события менеджер событий ищет его в активной карте и сопоставляет с именем события. Сопоставленное имя события помещается в очередь на обработку. Виджеты в этом случае должны обрабатывать полученную очередь событий.

Активная карта событий указывается в профиле каждого пользователя или устанавливается по умолчанию.

В целом предусмотрены четыре типа событий:

- события образов СВУ (префикс: `ws_`), например, событие нажатия кнопки `ws_BtPress`;
- клавишные события (префикс: `key_`) - все события от клавиатуры и мыши в виде `key_presAlt1`;
- пользовательские события (префикс: `usr_`) генерируются пользователем в процедурах обчёта виджетов;
- мапированные события (префикс: `map_`) - события, полученные из карты событий.

Само событие представляет мало информации, особенно если его обработка происходит на уровнях выше. Для однозначной идентификации события и его источника событие в целом записывается следующим образом: "`ws_BtPress:/curtime`". Где:

`ws_BtPress` – событие;

`/curtime` – путь к дочернему элементу, сгенерировавшему событие.

В таблице 1 приведен перечень стандартных событий, поддерживаемых в СКАДА.

Таблица 1

ID	Описание
<i>Клавиатурные события: key_[pres rels][Ctrl Alt Shift]{Key}:</i>	
*SC#3b	скан код клавиши
*#2cd5	код неименованной клавиши
*Esc	"Esc"
*BackSpace	удаления предыдущего символа - "<-"
*Return, *Enter	ввод - "Enter"
*Insert	вставка - "Insert"
*Delete	удаление - "Delete"
*Pause	пауза - "Pause"
*Print	печать экрана - "Print Screen"
*Home	дом - "Home"
*End	конец - "End"
*Left	влево - "<-"
*Up	вверх - '^'
*Right	вправо - "->"
*Down	вниз - 'v'
*PageUp	страницы вверх - "PageUp"
*PageDown	страницы вниз - "PageDown"
*F1 - *F35	функциональная клавиша от "F1" до "F35"
*Space.	пробел - ' '
*Apostrophe	апостроф - ''
Asterisk	звёздочка на дополнительном поле клавиатуры - ''
*Plus	плюс на дополнительном поле клавиатуры - '+'
*Comma	запятая - ','
*Minus	минус - '-'
*Period	точка - '.'
*Slash	наклонная черта - '\'
*0 - *9	цифра от '0' до '9'
*Semicolon	точка с запятой - ';'
*Equal	равно - '='
*A - *Z	клавиши букв латинского алфавита от 'A' до 'Z'
*BracketLeft	левая квадратная скобка - '['
*BackSlash	обратная наклонная линия - '/'
*BracketRight	правая квадратная скобка - ']'
*QuoteLeft	левая кавычка - '"'
<i>События клавиатурного фокуса:</i>	
ws_FocusIn	фокус получен виджетом
ws_FocusOut	фокус утерян виджетом
<i>События от манипулятора мышь:</i>	
key_mouse[Pres Rels][Left Right Midle]	нажата/отпущена кнопка мыши

ID	Описание
key_mouseDbIcIck	двойное нажатие левой кнопки мыши
ws_mouseEnter	наведение курсора на область виджета
ws_mouseLeave	покидание курсором области виджета
<i>События квитирования на стороне среды визуализации:</i>	
ws_alarmLev	квитирование всех нарушений всеми способами уведомления
ws_alarmLight	квитирование всех нарушений уведомления миганием/светом
ws_alarmAlarm	квитирование всех нарушений уведомления гудком
ws_alarmSound	квитирование всех нарушений уведомления звуком/речью
<i>События примитива элементарной фигуры ElFigure:</i>	
ws_Fig[Left Right Midle DbIcIck]	активация фигур (заливок) клавишей мыши
ws_Fig[n][Left Right Midle DbIcIck]	активация фигуры (заливки) [n] клавишей мыши
<i>События примитива элементов формы FormEl:</i>	
ws_LnAccept	установлено новое значение в строке ввода
ws_TxtAccept	изменено значение редактора текста
ws_ChkChange	состояние флажка изменено
ws_BtPress	кнопка нажата
ws_BtRelease.	кнопка отпущена
ws_BtToggleChange	изменена вдавленность кнопки
ws_CombChange	изменено значение поля выбора
ws_ListChange	изменен текущий элемент списка
ws_SliderChange	изменение положения слайдера
<i>События примитива медиа-контента Media:</i>	
ws_MapAct{n}[Left Right Midle]	активирована медиа-область с номером {n} клавишей мыши
ws_MediaFinished	окончание проигрывания Медиа-потока

События являются основным механизмом уведомления и активно используются для осуществления взаимодействия с пользователем. Для обработки событий предусмотрены два механизма: сценарии управления открытием страниц и вычислительная процедура виджета.

Механизм "Сценарии управления открытием страниц" основан на стандартном атрибуте виджета "evProc" и описан в разделе 1.

Механизм "Обработка событий с помощью вычислительной процедуры виджета" основан на атрибуте "event" и пользовательской процедуре вычисления на одном из языков пользовательского программирования СКАДА. События по мере поступления аккумулируются в атрибуте "event" до момента вызова вычислительной процедуры.

Вычислительная процедура вызывается с указанной периодичностью вычисления виджета и получает значение атрибута "event" в виде списка событий.

В процедуре вычисления пользователь может: проанализировать, обработать и исключить обработанные события из списка, а также добавить в список новые события. Оставшиеся после исполнения процедуры события анализируются на предмет соответствия условиям вызова сценарием первого механизма, после чего, оставшиеся события передаются на верхний по иерархии виджет для обработки им, при этом осуществляется коррекция пути событий в соответствии с иерархией проникновения события.

Содержимое атрибута "event" является списком событий формата <event>:<evSrc>, с событием в отдельной строке. Приведём пример процедуры обработки событий на Java-подобном языке пользовательского программирования СКАДА:

```
using Special.FLibSYS;
ev_rez = "";
off = 0;
while(true)
{
sval = strParse(event,0,"\n",off);
if( sval == "" ) break;
else if( sval == "ws_BtPress:/cvt_light" ) alarmSt = 0x1000001;
else if( sval == "ws_BtPress:/cvt_alarm" ) alarmSt = 0x1000002;
else if( sval == "ws_BtPress:/cvt_sound" ) alarmSt = 0x1000004;
else ev_rez+=sval+"\n";
}
event=ev_rez;
```

3 Сигнализация

Важным элементом любого интерфейса визуализации является уведомление пользователя про нарушения – сигнализация. Для упрощения восприятия, а также в виду тесной связности визуализации и уведомления, интерфейс уведомления интегрирован в интерфейс визуализации. Во всех виджетах предусмотрены два дополнительных атрибута (уровня сеанса): "alarm" и "alarmSt". Атрибут "alarm" используется для формирования сигнала виджетом в соответствии с его логикой, а атрибут "alarmSt" используется для контроля за фактом сигнализации ветви дерева сеанса проекта.

Атрибут "alarm" является строкой и имеет следующий формат: {lev|categ|message|type|tp_arg}, где:

lev – уровень сигнализации: число от 0 до 255;

categ - категория сигнала: параметр подсистемы сбора, объект, путь или комбинация;

message - сообщение сигнализации;

type - типы уведомления (визуальное, гудок и речь), формируется в виде целого числа, содержащего флаги способов уведомлений:

0x01 - визуальная;

0x02 - гудок, часто производится через PC-speaker;

0x04 - звуковой сигнал из файла звука или синтез речи; если в <tp_arg> указано имя ресурса звукового файла, то воспроизводится именно он, иначе выполняется синтез речи из текста указанного в <message>.

tp_arg - аргумент типа, используется в случае осуществления звуковой сигнализации для указания ресурса звукового сигнала (файл звукового формата).

Атрибут "alarmSt" является целым числом, которое отражает максимальный уровень сигнала и факт квитирования ветви дерева сеанса проекта. Формат числа имеет следующий вид:

- первый байт (0-255) характеризует уровень сигнала ветви;
- второй байт указывает тип уведомления (также как и в атрибуте "alarm");
- третий байт указывает тип несквитированного уведомления (также как и в атрибуте "alarm");
- первый бит четвёртого байта имеет специальное назначение, установка этого бита является фактом квитирования уведомлений указанных первым байтом.

3.1 Формирование сигнала и получение его средой визуализации

Формирование сигнала производится самим виджетом путём установки собственного атрибута "alarm" нужным образом, и в соответствии с ним устанавливается атрибут "alarmSt" текущего и вышестоящих виджетов. Среда визуализации получает уведомление о сигнале с помощью стандартного механизма уведомления об изменении атрибутов виджетов.

Такой механизм предоставляет возможность формировать интерфейсы сигнализации как на уровне подсистемы "Сбор данных", так и прямо на уровне представления.

Учитывая то, что обработка условий сигнализации осуществляется в виджетах, страницы, содержащие объекты сигнализации, должны исполняться в фоне, не зависимо от открытости их в данный момент. Это осуществляется путём установки флага исполнения страницы в фоне.

Хотя механизм сигнализации и построен в среде визуализации, возможность формирования невидимых элементов сигнализации остаётся, например, путём создания страницы, которая никогда не будет открываться.

3.2 Квитирование

Квитирование производится путём указания корня ветви виджетов и типов уведомления. Это позволяет реализовать квитирование на стороне среды визуализации как по группам, например, по объектам сигнализации, так и индивидуально по объектам. При этом можно независимо квитировать разные типы сигнализаций. Установка квитирования производится простой модификацией атрибута "alarmSt".

Пример скрипта для работы с сигналами приведён ниже:

```
//Выделение факта наличия сигнализаций разных способов уведомления
cvt_light_en = alarmSt&0x100;
cvt_alarm_en = alarmSt&0x200;
cvt_sound_en = alarmSt&0x400;
//Выделение факта наличия несквитированных сигнализаций разных способов
уведомления
cvt_light_active = alarmSt&0x10000;
cvt_alarm_active = alarmSt&0x20000;
cvt_sound_active = alarmSt&0x40000;
//Обработка событий кнопок квитирования и квитирование разных способов
уведомлений
ev_rez = "";
```

```
off = 0;
while(true)
{
    sval = strParse(event,0,"\n",off);
    if(sval == "") break;
    else if(sval == "ws_BtPress:/cvt_light") alarmSt = 0x1000001;
    else if(sval == "ws_BtPress:/cvt_alarm") alarmSt = 0x1000002;
    else if(sval == "ws_BtPress:/cvt_sound") alarmSt = 0x1000004;
    else ev_rez+=sval+"\n";
}
event=ev_rez;
```

4 Управление правами

Для разделения доступа к интерфейсу ВУ и его составляющим каждый виджет содержит информацию о владельце, его группе и правах доступа. Права доступа записываются в форме: <пользователь><группа><остальные>, где каждый элемент состоит из трёх признаков доступа. Для элементов СВУ принята следующая их интерпретация:

- 'r' - право на просмотр виджета;
- 'w' - право на контроль над виджетом.

В режиме разработки используется простая схема доступа "`root.UI:RWRWR_`", что означает - все пользователи могут открывать и просматривать библиотеки, их компоненты и проекты, а редактировать могут все пользователи группы "UI" (пользовательские интерфейсы).

В режиме исполнения работают права, описанные в компонентах интерфейса.


5 Библиотеки графических элементов

Пользовательский интерфейс ПП «СКАДА А-СОФТ» представляет собой набор графических страниц (мнемосхем). Каждая графическая страница располагается в рабочей области окна. Иерархия страниц определяет навигацию пользователя по проекту.

Иерархия страниц разрабатывается в виде многоуровневых мнемосхем, доступ к которым в любой момент времени зависит от прав текущего пользователя.

Мнемосхемы являются средствами визуального контроля за технологическим процессом и предоставляют доступ в режиме реального времени к используемому технологическому оборудованию, отображаемому на них, в части:

- текущего состояния;
- управления (при наличии соответствующих прав пользователя);
- измерения текущих параметров;
- достоверности сигналов;
- состояния интерфейсных каналов связи ПТС, позволяющие в режиме реального времени формировать управляющие воздействия для управления технологическим оборудованием.

Вызов мнемосхем может осуществляться через кнопки вызова мнемосхемы, с помощью кнопок навигации  , с использованием протоколов сигнализации или с помощью функций пользовательского API.

Мнемосхемы могут быть объединены в группы. На рисунке 7 приведен пример вызова мнемосхем с помощью кнопок «Основная схема», «Левая зона», «Правая зона» и т.д.

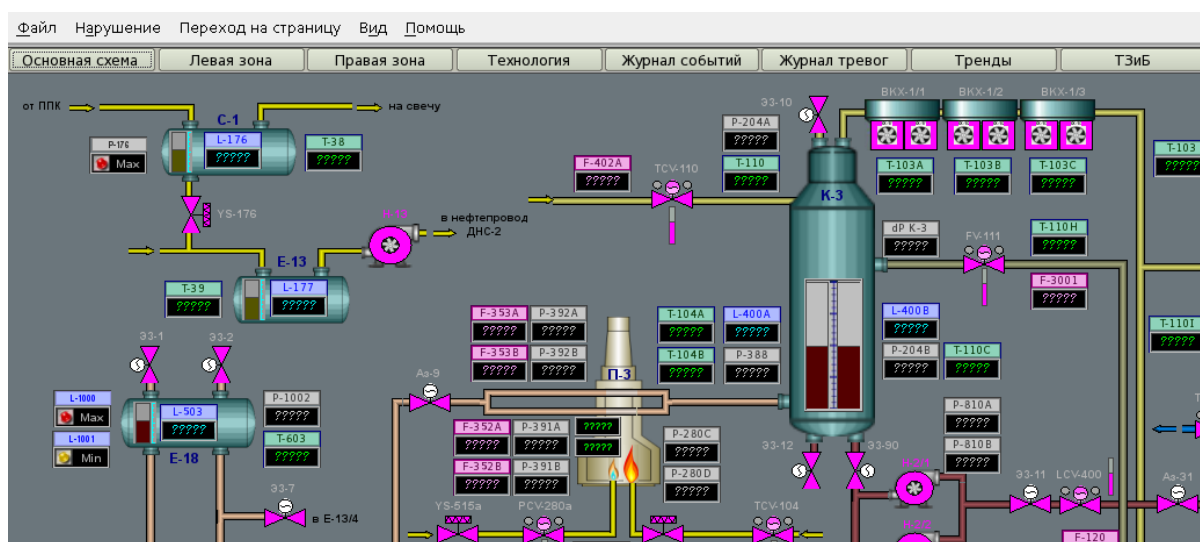


Рисунок 7

В рабочей области мнемосхем могут размещаться статические и динамические объекты.

Статические объекты представляют собой текстовые надписи, рисунки, линии и прочие фигуры, выполненные в однотонной цветовой гамме, либо состоящие из разноцветных элементов.

Динамические объекты представляют собой пиктограммы, отображающие состояние механизмов, процессов, аналоговых и дискретных параметров.

Операция выбора динамических элементов на мнемосхемах позволяет вызывать окна управления данными элементами, либо окна, которые предназначены для более детального отображения информации (графики).

Все виджеты элементов мнемосхем распределены по библиотекам, подключаемым по мере необходимости.

5.1 Библиотека «Базовые виджеты» (originals)

Библиотека «Базовые виджеты» (originals) содержит образы базовых элементов (примитивов) используемых для разработки проектов.

Таблица 2

Id	Наименование	Функция
ElFigure	Элементарные графические фигуры	<p>Примитив является основой для отрисовки элементарных графических фигур со всевозможной комбинацией их в одном объекте. Предусматривается поддержка следующих элементарных фигур: линия, дуга, кривая Безье, заливка замкнутого пространства.</p> <p>Для всех фигур, содержащихся в виджете, устанавливаются единые свойства толщины, цвета и т.д., которые можно изменять при необходимости.</p> <p>Для редактирования элементарных фигур используется специальный графический редактор, описанный в подразделе 5.2</p>
FormEl	Элементы формы	<p>Включает поддержку стандартных компонентов формы:</p> <ul style="list-style-type: none"> редактирование строки; редактирование текста; флажок; кнопка; поле выбора из списка; список; таблица; дерево; слайдер; строка прокрутки
Text	Текстовые поля	Элемент текста (метки). Характеризуется типом шрифта, цветом, ориентацией и выравниванием
Media	Медиа	Элемент отображения растровых и векторных изображений различных форматов, проигрывания анимированных изображений, проигрывания аудио фрагментов, просмотра видеофрагментов и атрибутов СКАДА-системы типа массив
Diagram	Диаграмма	Элемент построения графиков и диаграмм с поддержкой возможности отображения нескольких потоков трендов и различных режимов отображения
Protocol	Протокол	Элемент визуализации данных архива сообщений путём формирования протоколов с различными способами визуализации, начиная от статического сканирующего просмотра и заканчивая динамическим отслеживанием протокола сообщения

Id	Наименование	Функция
Document	Документ	Элемент формирования отчётов, журналов и другой документации на основе указанных данных
Box	Группа элементов (контейнер)	Содержит механизм размещения других виджетов с целью формирования новых, более сложных виджетов и страниц конечной визуализации
Surface	Поверхность	Виджет, позволяющий строить трехмерные поверхности различных типов по заданным пользователем или архивным данным

Более детально рассмотрим индивидуальные атрибуты каждого примитива.

5.1.1 Примитив элементарная фигура (ElFigure)

Реализована поддержка элементарных фигур: линия, эллиптическая дуга, кривая Безье и заливка замкнутых контуров цветом и изображением.

Для элементарных фигур реализованы следующие операции:

- создание/удаление фигур;
- копирование фигур;
- перемещение и изменение размеров фигур с помощью мыши и клавиатуры;
- возможность связывать элементарные фигуры друг с другом, получая более сложные, для которых доступны все свойства исходных элементарных фигур;
- возможность одновременного перемещения нескольких фигур.

Фигуры, лежащие в основе данного виджета, содержат точки (начальная и конечная), которые могут стыковаться с соответствующими точками других фигур, и точки, с помощью которых изменяется геометрия фигуры.

Индивидуальными атрибутами данного примитива являются:

Линия: ширина (lineWdth) – ширина линии;

Линия: цвет (lineClr) – имя цвета в виде "color[-alpha]", где:

- "color" – стандартное имя цвета или числовое представление из трёх шестнадцатеричных чисел цвета "#RRGGBB";
- "alpha" – уровень альфа-канала (0-255).

Примеры:

"red" – сплошной красный цвет;

"#FF0000" – сплошной красный цвет в цифровом коде;

"red-127" – полупрозрачный красный цвет;

Линия: стиль (lineStyle) – стиль линии (сплошная, пунктир, точечная);

Граница:ширина (bordWdth) – ширина бордюра линии. Нулевая ширина указывает на отсутствие бордюра;

Граница:цвет (bordClr) – цвет бордюра;

Заполнение:цвет (fillColor) – цвет заливки;

Заполнение:изображение (fillImg) – имя изображения в форме "[src:]name", где: "src" – источник изображения:

- file – прямо из локального файла по пути;

- res – из таблицы mime ресурсов БД.

"name" – путь файла или идентификатор mime-ресурса.

Примеры:

"res:backLogo" и "backLogo" – из таблицы mime ресурсов БД для идентификатора "backLogo";

"file:/var/tmp/backLogo.png" – из локального файла по пути "/var/tmp/backLogo.png".

Угол поворота (orient) - угол поворота содержимого виджета;

Список элементов (elLst) - список графических примитивов в формате:

Линия:

```
line: (x|y) | {1}: (x|y) | {2}: [width|w{n}]: [color|c{n}]:  
[bord_w|w{n}]: [bord_clr|c{n}]: [line_stl|s{n}]
```

Дуга:

```
arc: (x|y) | {1}: (x|y) | {2}: (x|y) | {3}: (x|y) | {4}: (x|y) | {5}: [width|  
w{n}]: [color|c{n}]: [bord_w|w{n}]: [bord_clr|c{n}]: [line_stl|s{n}]
```

Кривая Безье:

```
bezier: (x|y) | {1}: (x|y) | {2}: (x|y) | {3}: (x|y) | {4}: [width|w{n}]:  
[color|c{n}]: [bord_w|w{n}]: [bord_clr|c{n}]: [line_stl|s{n}]
```

Заливка:

```
fill: (x|y) | {1}, (x|y) | {2}, ..., (x|y) | {n}: [fill_clr|c{n}]:  
fill_img|i{n}]
```

Где:

- (x|y) – прямая точка (x,y) координаты в пикселах с плавающей точкой;
- {1}...{n} – динамические точки 1...n;
- width, bord_w – прямая ширина линии и бордюра в пикселах с плавающей точкой;
- w{n} – динамическая ширина 'n';

- color, bord_clr, fill_clr – прямой цвет линии, бордюра и заполнения в виде имени или 32-битного кода с альфа: {имя}-AAA, #RRGGBB-AAA;
- c{n} – динамический цвет 'n';
- line_stl – прямой стиль линии: 0-сплошная, 1-пунктирная, 2-точечная;
- s{n} – динамический стиль 'n';
- fill_img – прямое изображение заполнения в формате "[src%3Aname]", где:
 - "src" – источник изображения:
 - file – непосредственно из локального файла по пути;
 - res – из таблицы mime-ресурсов БД.
 - "name" – путь файла или идентификатор mime-ресурса;
- i{n} – динамическое изображение заполнения 'n'.

На рисунке 8 представлен пример видеокadra, содержащий элементарные фигуры с атрибутами этих фигур.

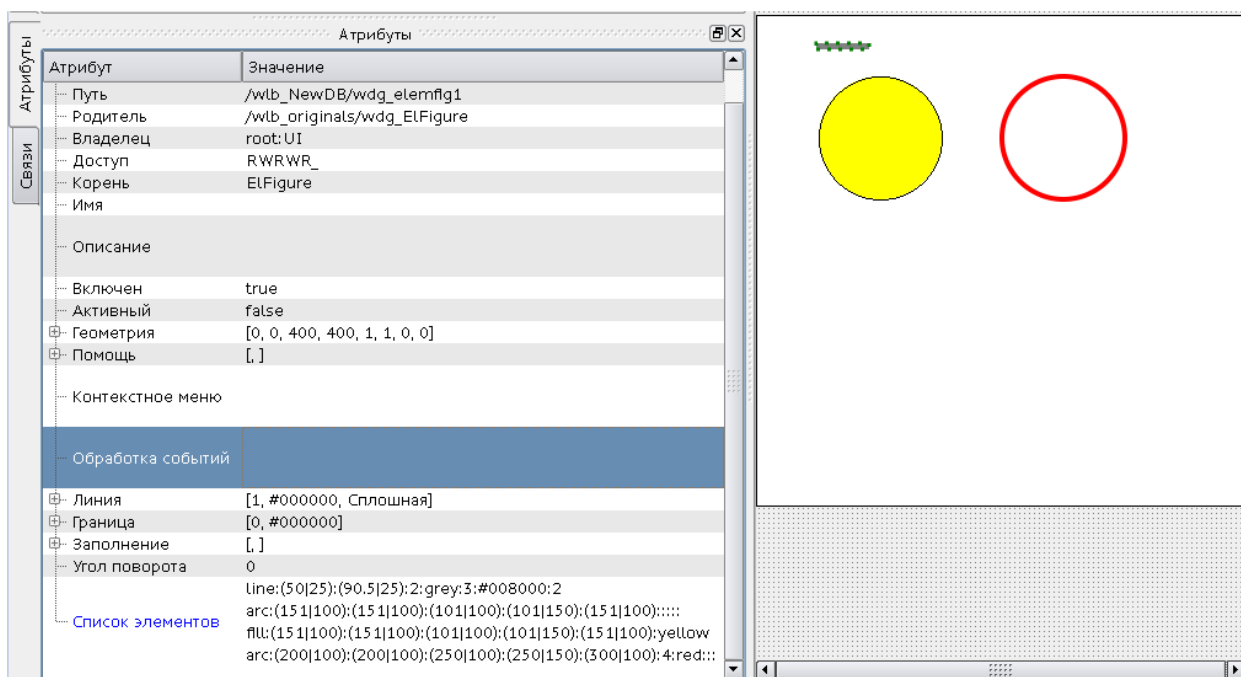


Рисунок 8

5.1.2 Примитив текста (Text)

Индивидуальными атрибутами примитива текст являются:

- шрифт (font) – имя шрифта в виде:

"{family}{size}{bold}{italic}{underline}{strike}", где:

- "family" – семейство шрифта, для пробелов используйте символ ' ', вроде: "Arial", "Courier", "Times_New_Roman";
- "size" – размер шрифта в пикселах;
- "bold" – усиление шрифта (0 или 1);
- "italic" – наклонность шрифта (0 или 1);
- "underline" – подчёркивание шрифта (0 или 1);
- "strike" – перечёркивание шрифта (0 или 1);
- *цвет текста (color)*;
- *ориентация текста (orient)*, поворот на угол;
- *автоматический перенос по словам (wordWrap)*;
- *выравнивание текста по горизонтали и вертикали (align)*;
- *отображение фона в виде цвета и/или изображения (backColor, backImg)*;
- *отображение бордюра* вокруг текста, с указанным цветом, шириной и стилем (bordWight, boardColor, boardStyle);
- *формирование текста из аргументов различного типа и свойств*. Аргумент может быть трех типов: integer, real, string. Конфигурация аргумента: целое – [len] - ширина значения; вещественное – [width];[form];[prec] - ширина значения, форма значения('g', 'e', 'f'); строка-[len] - ширина строки.

На рисунке 9 представлен видеокادر, содержащий примеры текста с использованием различных параметров.

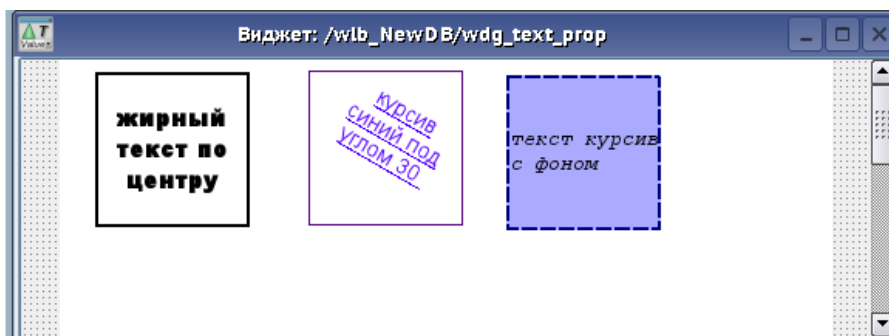


Рисунок 9

5.1.3 Прimitives элементы формы (FormEl)

Реализованы следующие типы элементов формы:

- *Строка редактирования* - представлено следующими видами: «Текст», «Combo», «Целое», «Вещественное», «Время», «Дата», «Время и Дата»;

- *Текстовый редактор* - представляет редактор плоского текста с подтверждением или отказом от ввода;
- *Check Box* - предоставляет поле выбора бинарного флажка;
- *Кнопка* - предоставляет кнопку с поддержкой: цвета кнопки, изображения в кнопке и режима фиксации;
- *Combo Box (выбор из списка)* - предоставляет поле выбора элемента, со списка указанных элементов;
- *Список* - предоставляет поле списка с контролем за текущим элементом;
- *Дерево* - элемент иерархической структуры;
- *Таблица* - предоставляет редактор создания таблиц;
- *Слайдер* - элемент слайдера;
- *Полоса прокрутки*.

Для указанных элементов реализованы режимы: «Включен» и «Активен», а также передача изменений и событий в модель данных СВУ. Все элементы могут быть использованы для создания форм пользовательского ввода.

Индивидуальные атрибуты каждого элемента формы представлены в таблице 3.

Таблица 3

Название (ID)	Значение
<i>Строка редактирования</i>	
Значение (value)	Содержимое строки
Вид (view)	Вид строки редактирования: текст, комбобокс, целое, вещественное, время, дата, время и дата
Конфигурация (cfg)	<p>Конфигурация строки. Формат значения данного поля для различных видов строки:</p> <p><i>Текст</i> – ввод строки текста по шаблону с символьными элементами:</p> <p>A - необходим ASCII алфавитный символ: A-Z, a-z;</p> <p>a - разрешён, но не необходим ASCII алфавитный символ;</p> <p>N - необходим ASCII алфавитно-цифровой символ: A-Z, a-z, 0-9;</p> <p>n - разрешён, но не необходим ASCII алфавитно-цифровой символ;</p> <p>X - необходим любой символ;</p> <p>x - разрешён, но не необходим любой символ;</p> <p>9 - ASCII цифра необходима: 0-9;</p> <p>0 - ASCII цифра разрешена, но не необходима;</p> <p>D - ASCII цифра необходима: 1-9;</p> <p>d - ASCII цифра разрешена, но не необходима (1-9);</p> <p># - ASCII цифра или знаки плюс/минус разрешены, но не необходимы;</p> <p>H - необходим символ шестнадцатиричного числа: A-F, a-f, 0-9;</p> <p>h - разрешён, но не необходим символ шестнадцатиричного числа;</p> <p>B - необходим бинарный символ: 0-1;</p> <p>b - разрешён, но не необходим бинарный символ;</p> <p>> - все следующие алфавитные символы в верхнем регистре;</p>

Название (ID)	Значение
	<p>< - все следующие алфавитные символы в нижнем регистре; ! - выключение преобразования регистра; \ - используйте в разделителях для экранирования специальных символов, которые перечислены. <i>Комбобокс</i> - список значений редактируемого комбо-бокса по строкам; <i>Целое</i> - значение целого числа в форме: "[Минимум]:[Максимум]:[ШагИзменения]:[Префикс]:[Суффикс]"; <i>Вещественное</i> - значение вещественного числа в форме: "[Минимум]:[Максимум]:[ШагИзменения]:[Префикс]:[Суффикс]:[ЗнаковПослеТочки]"; <i>Время, Дата, Дата и время</i> - формировать дату по шаблону с параметрами: d - номер дня (1-31); dd - номер дня (01-31); ddd - сокращённое наименование дня ("Mon" ... "Sun"); dddd - полное наименование дня ("Monday" ... "Sunday"); M - номер месяца (1-12); MM - номер месяца (01-12); MMM - сокращённое название месяца ("Jan" ... "Dec"); MMMM - полное наименование месяца ("January" ... "December"); uu - последние две цифры года; uuuu - год полностью; h - час (0-23); hh - час (00-23); m - минуты (0-59); mm - минуты (00-59); s - секунды (0-59); ss - секунды (00-59); AP,ap - отображать AM/PM или am/pm</p>
Подтверждать (confirm)	Включение режима подтверждения
Шрифт (font)	<p>Имя шрифта в формате: "{family} {size} {bold} {italic} {underline} {strike}", где: - "family" - семейство шрифта, для пробелов используйте символ ' ', вроде: "Arial", "Courier", "Times_New_Roman"; - "size" - размер шрифта в пикселах; - "bold" - усиление шрифта (0 или 1); - "italic" - наклонность шрифта (0 или 1); - "underline" - подчёркивание шрифта (0 или 1); - "strike" - перечёркивание шрифта (0 или 1)</p>
<i>Текстовый редактор</i>	
Значение(value)	Содержимое редактора
Перенос слов (wordWrap)	Автоматический перенос текста по словам
Подтверждать (confirm)	Включение режима подтверждения
Шрифт (font)	Имя шрифта (формат описан выше)

Название (ID)	Значение
<i>Check Box</i>	
Имя (name)	Имя/метка флага
Значение(value)	Значение флага
Шрифт (font)	Имя шрифта (формат описан выше)
<i>Кнопка</i>	
Имя (name)	Имя, надпись на кнопке
Значение(value)	Значение для фиксированной кнопки
Изображение (img)	Изображение на кнопке. Имя изображения в форме "[src:]name", где: "src" - источник изображения: - file - прямо из локального файла по пути; - res - из таблицы mime ресурсов БД. "name" - путь файла или идентификатор mime-ресурса. Примеры: "res:backLogo" или "backLogo" - из таблицы mime ресурсов БД для идентификатора "backLogo"; "file:/var/tmp/backLogo.png" - из локального файла по пути "/var/tmp/backLogo.png"
Цвет (color)	Цвет кнопки. Имя цвета в виде " color[-alpha] ", где: "color" - стандартное имя цвета или числовое представление из трёх шестнадцатеричных чисел цвета "#RRGGBB"; "alpha" - уровень альфа-канала (0-255). Примеры: "red" - сплошной красный цвет; "#FF0000" - сплошной красный цвет в цифровом коде; "red-127" - полупрозрачный красный цвет.
Цвет:текст (colorText)	Цвет текста
Переключатель (checkable)	Признак функционирования как фиксированная кнопка
Шрифт (font)	Имя шрифта (формат описан выше)
<i>Список и выбор из списка</i>	
Значение(value)	Выбранное значение списка
Элементы (items)	Перечень элементов списка в формате "[Текст][ID][Icon]", где "Текст" - элемент списка; "ID" - ID элемента списка; "Icon" - источник, может быть указан путь к файлу - "file" или "res" - из таблицы mime ресурсов БД
Шрифт (font)	Имя шрифта (формат описан выше)
<i>Слайдер и полоса прокрутки</i>	
Значение(value)	Положение слайдера
Конфигурация (cfg)	Конфигурация слайдера в формате: "[ВертОриент]:[Минимум]:[Максимум]:[ОдинШаг]:[СтрШаг]". Где: "ВертОриент" - признак вертикальной ориентации, по умолчанию ориентация горизонтальная; "Минимум" - минимальное значение; "Максимум" - максимальное значение; "ОдинШаг" - размер одного шага; "СтрШаг" - размер страничного шага.

Название (ID)	Значение
<i>Таблица</i>	
Значение(value)	Текущее значение
Элементы	<p>XML тег“tbl” для заполнения таблицы:</p> <pre><tbl> <h><s>{Заголовок1}</s><s>{Заголовок2}</s></h> <r><s>{Ряд1Колонка1Строка}</s><i>{Ряд1Колонка2Целое}</i></r> <r>{Ряд2Колонка1Логическое}<r>{Ряд2Колонка2Вещественно e}</r></tbl></pre> <p>Теги:</p> <p>tbl – Таблица, свойства таблицы в целом:</p> <ul style="list-style-type: none"> sel –режим выбора-выделения элементов таблицы: “row” –по строкам, “col”- по колонкам, “cell”- ячейками по умолчанию; KeyID –номер ключевой строки-колонки, для получения значения выбора; colsWdthFit – подстраивать размер колонок (размер которых не фиксирован)под заполнениевсей ширины таблицы; hHdrVis, vHdrVis – установка видимости горизонтального, вертикального заголовков; sortEn – включение прямой сортировки по колонкам. <p>h- Строка заголовков. Возможные атрибуты тегов ячеек заголовка для колонки в целом:</p> <ul style="list-style-type: none"> width – ширина колонки в пикселах или процентах(10%); edit – возможность редактирования(0 или 1) ячеек колонки, по умолчанию –нет(0); color – цвет колонки в целом, в виде имени цвета или его кода; colorText – цвет текста колонки в целом, в виде имени цвета или его кода; font – шрифт текста колонки в целом, в виде типовой строки SCADA; sort- сортировка по данной колонке [0-по убыванию; 1 –по возрастанию]; <p>r – Ряд значений. Возможные атрибуты тегов ячеек ряда для ряда в целом:</p> <ul style="list-style-type: none"> color – цвет ряда в целом, в виде имени цвета или его кода; colorText – цвет текста ряда в целом, в виде имени цвета или его кода; font – шрифт текста ряда в целом, в виде типовой строки SCADA; <p>s, i, r, b – ячейки типов данных строка, целое, вещественное и логическое. Возможные атрибуты:</p> <ul style="list-style-type: none"> color – цвет фона ячейки, в виде имени цвета или его кода; colorText – цвет текста ячейки, в виде имени цвета или его кода; font – шрифт текста ячейки, в виде типовой строки SCADA; img – изображение ячейки в форме “[{src:}] {name}” edit – возможность редактирования (0 или 1) ячейки колонки, по умолчанию – нет (0)
Шрифт (font)	Имя шрифта (формат описан выше)
<i>Дерево</i>	
Значение(value)	Выбранное значение
Элементы	Список элементов в виде пути: «/кат/кат/элемент» по строкам
Шрифт (font)	Имя шрифта (формат описан выше)

На рисунке 10 представлен видеокадр, содержащий элементы формы.

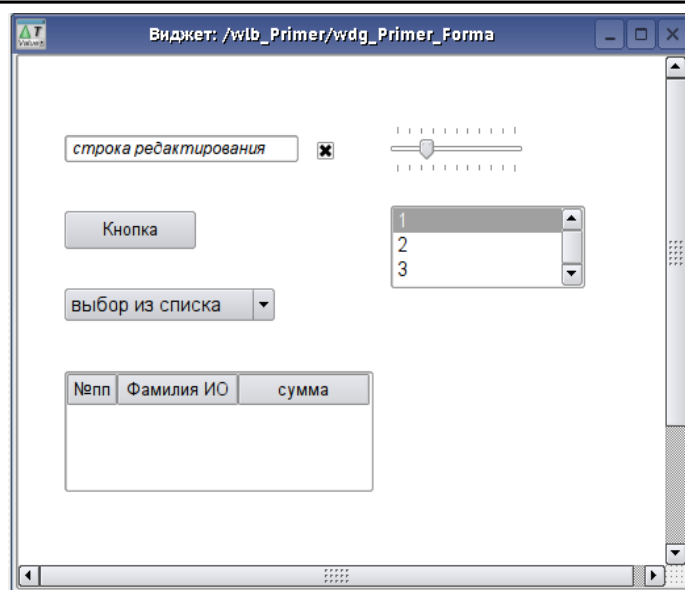


Рисунок 10

5.1.4 ПрIMITив отображения медиа-материалов (Media)

Индивидуальными атрибутами данного примитива являются:

источник медиа данных (изображения или видео-материала) в формате

``[src:]name'', где src - источник:

- file-прямо из файла,
- res - из таблицы mime ресурсов БД,
- stream - URL потока проигрывания видео или аудио,
- prn - из атрибута контроллера(только для типа медиа - массив);
- *тип медиа(type)*: изображение, анимация, полное видео, массив
- *область карты (areas)* – количество активных областей;
- *заполнять виджет(fit)* – согласовать содержимое с размером виджета.

В зависимости от типа медиа-данных становятся доступными дополнительные атрибуты:

для анимации(Movie): скорость проигрывания в процентах от оригинальной скорости.

Если значение меньше 1% - проигрывание прекращается;

для полноформатного видео (Full video):

- играть(play);
- Завер.проигр.(roll) – повторение проигрывания по завершению;
- пауза(pause);

- размер (size) – общий размер видео (в мс);
- положение (seek) – позиция проигрывания видео (в мс);
- громкость (volume) – громкость звука(0..100);

для массива:

- период слежения, с (trcPer) – интервал обновления виджета;
- пауза(pause);
- формат данных – формат отображения данных: нет, RGB, монохромный;
- архиватор значений(arch) – в формате «МодульАрхивов.IDАрхиватора»;
- промежуток времени;
- цвет минимального значения - считывается при незаданном формате данных;
- цвет максимального значения - считывается при незаданном формате данных;
- изображение: ширина и высота отображения.

На рисунке 11 представлена часть экрана с видеокадром, содержащим примеры просмотра/проигрывания медиа-данных.

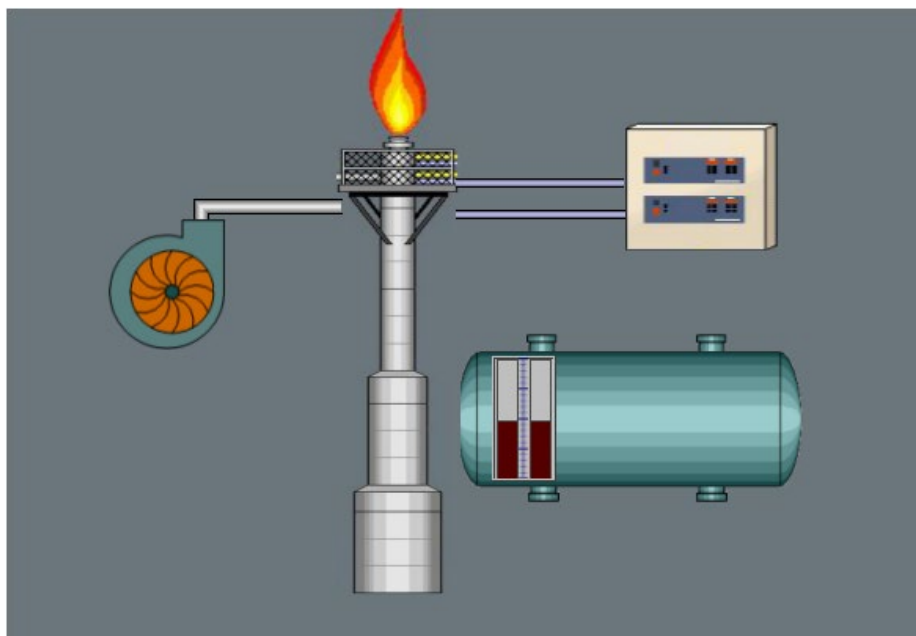


Рисунок 11

5.1.5 Прimitив построения диаграмм/графиков (Diagram)

Содержит следующие типы диаграмм:

"График" – построение зависимости величины какого-то измеряемого параметра от времени;

"Массив" – отображение объектных типов данных. По шкале абсцисс откладываются целые значения – соответствующие номеру элемента в массиве, а по шкале ординат откладываются сами элементы массива. Таким образом, имея некий массив данных из 1500 элементов, данные будут распределены в диапазоне [0:1500];

"Зависимость" – строит график зависимости одного параметра от другого.

Для всех типов диаграмм в качестве источника данных возможно указание:

- параметра контроллера подсистемы "Сбор Данных";
- архива значений.

Индивидуальными атрибутами виджета «Диаграмма» являются:

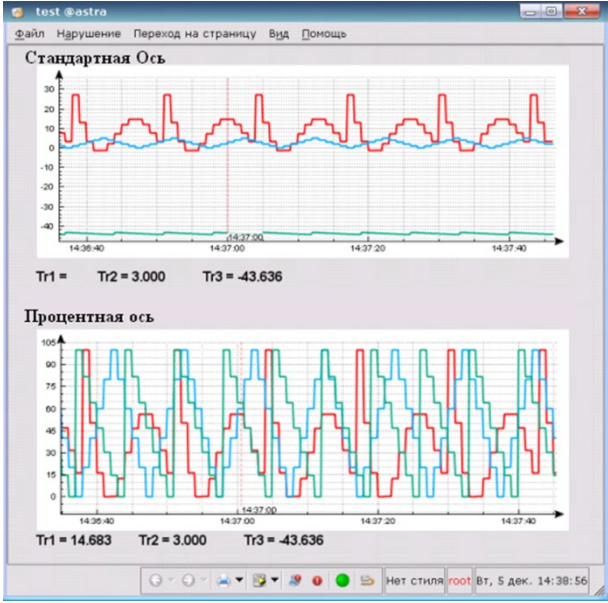
- *период слежения (trcPer)* – задает интервал обновления виджета. При выставлении значения в «0» трассировка прекращается, при этом запускается обновление текущей области отображения с дозагрузкой недостающих данных;
- *тип (type)* – график, массив, график зависимости;
- *время* – задает правую границу временной шкалы графика и состоит из двух вложенных атрибутов:
 - сек (указывается секундная составляющая, например 16.01.2018 11:21:20);
 - микросек (указывается микро секундная составляющая, например 146070);
- *размер* – устанавливает размер временной шкалы в секундах, так например, если мы установим атрибут «время» 16.01.2018 11:21:20, а атрибут «размер» установим в 60 секунд, то временная шкала (ось абсцисс – ось x) будет лежать в диапазоне [11:20:20; 11:21:20];
- *строгий диапазон* – использует данные только из указанного диапазона и в основном используется для графиков типа «Массив»;
- *курсор* – задает левую границу графика (точку отсчета), работает в связке с атрибутом статические оси и состоит из трех вложенных атрибутов:
 - сек (указывается секундная составляющая, например 16.01.2018 11:21:20);
 - микросек (указывается микро секундная составляющая, например 146070);
 - цвет (указывается цвет точки отсчета).
- *архиватор значений* – позволяет указать, откуда брать данные для построения графика:

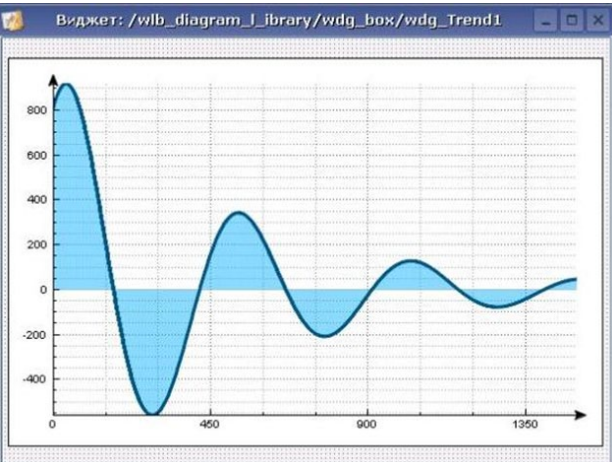
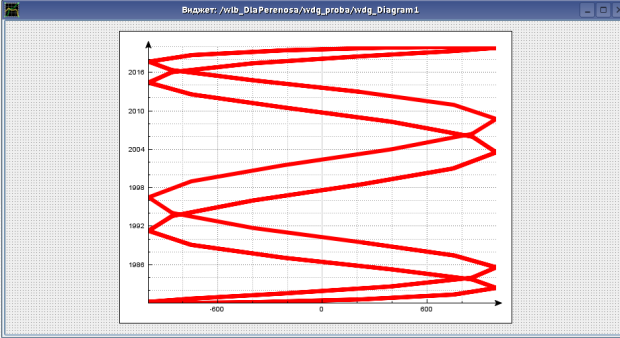
- из буфера – значение <buffer>;
- из архива - в поле ввести архиватор значений в форме:
«Модуль архивов.IdАрхиватора» (например DBArch.postgresArhiv);
- из архива, если в нем данные на единицу времени отсутствуют, брать из буфера (оставить поле пустым);
- *параметры* состоит из динамического числа атрибутов, один из которых является обязательным - «число графиков» и указывает на число поддерживаемых виджетом графиков (максимум 10). Остальные вложенные атрибуты формируются для каждого графика в зависимости от выставленного значения. Вложенными атрибутами являются:
 - адрес – указывает адрес к источнику данных в формате:
 - полный адрес к атрибуту параметра DAQ или Архива в форматах: Модуль/Сбор_данных/Контроллер/Параметр/Атрибут или /Архив/va_адрес к архиву,
 - прямая установка данных по префиксу data:<XMLNodeData>,
 - прямая установка данных по префиксу line:<значение>;
 - ширина линии;
 - цвет линии;
 - цвет заливки графика;
 - заливать график;
 - отображать график;
 - маркеры - ошибка(цвет), норма(цвет), отображать (ошибки, не отражать, все), тип предупредительной линии (нет, сплошная, пунктир, точечная, пунктир точка, пунктир точка точка, жирная);
 - значения;
 - свойства реального архива в виде “BegArh:EndArh:DataPeriod”, где “BegArh” – начало архива, “EndArh” – конец архива, “DataPeriod” – период данных архива в секундах, в реальном представлении вплоть до микросекунд(1e-6);
 - тип линии – нет, линия, ступенька справа, ступенька по центру, импульс, ступенька слева;
- *шкалы* – настройка следующих свойств шкал виджета:
 - цвет – задает цвет шкал;

- статические оси – поддерживается только для типа Диаграммы «График», по умолчанию установлено значение «false» (описание в таблице 4);
- процентные оси (*sclPercent*) - позволяет установить флаг, переводящий оси в процентный масштаб (описание в таблице 4);
- маркеры – позволяет изменять отображение числовых значений на шкалах посредством вложенных атрибутов цвет и шрифт;
- X оси - позволяет настраивать шкалы абсцисс, посредством установки следующих вложенных атрибутов тип и диапазон. Тип шкалы абсцисс позволяет настраивать отображение осей - не рисовать, решетка, маркеры, решетка и маркеры. Диапазон рассмотрен в таблице 4;
- Y оси - состоит из динамического числа атрибутов, один из которых является обязательным «число осей» и указывает на число отображаемых осей ординат. Остальные вложенные атрибуты формируются в зависимости от установленного значения. Каждый из динамических атрибутов представлен вложенным набором параметров, позволяющими настроить каждую из осей:
 - тип – способ отображения осей: не рисовать, глобально, маркеры, решетка и маркеры, маркеры (лог), решетка и маркеры (лог);
 - расположение – положение маркеров внутри или вне области построения графиков;
 - автомасштабирование – при включении игнорируются значения атрибутов мин. значение и макс. значение;
 - мин. значение – устанавливает нижнюю границу оси ординат;
 - макс. значение – устанавливает верхнюю границу оси ординат;
 - множитель – коэффициент, на значение которого умножается каждое значение параметра графика;
 - прикрепленные графики;
- *базовый масштаб* – установка атрибутов «время» и «размер» в формате «Time:значение;Size:значение;» (например: Time:0;Size:60;). В примере в качестве времени устанавливается текущее время, а размер отображаемой области равен 60 секунд. Установка этих значений позволяет указать параметры по умолчанию, к которым пользователь сможет вернуться в случае если изменял размеры графика (двигал, масштабировал и т.д.) в процессе работы.

В таблице 4 описаны индивидуальные атрибуты элемента диаграммы в зависимости от выбранного типа.

Таблица 4

Тип диаграммы	Индивидуальные атрибуты
<p data-bbox="225 461 331 495">График</p> 	<p data-bbox="901 461 1461 600">Вложенный атрибут «диапазон» у атрибута X оси – позволяет выбирать диапазон отображения из трех возможных вариантов:</p> <ul data-bbox="927 613 1267 719" style="list-style-type: none">– s – секунды;– ms – мили секунды;– mks – микро секунды. <p data-bbox="901 723 1366 757">По умолчанию стоят s – секунды.</p> <p data-bbox="901 761 1461 974">Процентные оси - позволяет установить флаг, переводящий оси в процентный масштаб. В этом случае диапазон шкалы Y будет установлен от 0 до 100%, все графики будут вписаны в этот диапазон.</p> <p data-bbox="901 978 1461 1675">Статические оси – при установке флага левая граница шкалы абсцисс устанавливается согласно значению атрибута «Базовое время», а график начинает строиться от этой точки к правой границе шкалы ординат, чье значение высчитывается посредством прибавления к значениям атрибута «Базовое время» значения атрибута «Размер». Разница между использованием этого атрибута и стандартной работой осей заключается в том, что в первом случае значения на оси абсцисс никуда не смещаются при превышении значения области отображения, тогда как во втором случае смещение значений идет справа на лево при превышении области отображения (атрибут «размер»)</p>

Тип диаграммы	Индивидуальные атрибуты
<p>Массивы</p> 	<p>Вложенные атрибуты X оси:</p> <ul style="list-style-type: none"> - «мин. значение»; - «макс. значение»; - «множитель»; - «смещение» устанавливает смещение по оси абсцисс, так например, если взять массив данных размером в 1500 элементов, задать значение атрибута равным 900, то построится тот же график, однако его данные будут распределены в диапазоне [900; 2400]
<p>График зависимости</p> 	<p>Вложенный атрибут «адрес (гориз.)» в атрибуте Параметры; Вложенные атрибуты X оси:</p> <ul style="list-style-type: none"> - «мин. Значение»; - «макс. Значение»; - «множитель».

5.1.6 ПрIMITив формирования протокола (Protocol)

Реализована поддержка элемента формирования протокола со свойствами:

- формирование протокола из архива сообщений за указанное время с заданной глубиной;
- запрос данных из указанных архиваторов сообщений;
- выборка данных из архивов по уровню важности и шаблону категории сообщений;
- поддержка режима слежение за появлением сообщений в архиве сообщений.

Индивидуальными атрибутами элемента являются:

- *заголовок видим (headVis)* – видимость заголовка таблицы;
- *время (time)* – текущее время;
- *размер (tSize)* – размер запроса, секунды. Установите значение в '0' для получения всех нарушений, для "lev" < 0;

- *период слежения, с (trcPer)* – режим и периодичность слежения;
- *архиватор (arch)* – архиватор сообщений в форме "МодульАрхивов.IdАрхиватора";
- *шаблон (tmpl)* - Шаблон категории или регулярное выражение "{re}/".

Для шаблона зарезервированы символы:

- '*' – множество любых, группа символов;
- '?' – любой, один символ;
- '\\' – используйте для экранирования специальных символов.
- *уровень (lev)* – уровень сообщений. Для получения текущих нарушений значение должно быть < 0;
- *порядок отображения (viewOrd)* - порядок отображения: "По времени", "По уровню", "По категории", "По сообщению", "По времени (обратно)", "По уровню (обратно)", "По категории (обратно)", "По сообщению (обратно)";
- *показать колонки (col)* – Список видимых и порядок колонок, разделённый символом ';'. Поддерживаются колонки:
 - "pos" – номер строки;
 - "tm" – дата и время сообщения;
 - "utm" – микросекундная часть времени сообщения;
 - "lev" – уровень сообщения;
 - "cat" – категория сообщения;
 - "mess" – текст сообщения;
- *свойства элемента (ItProp)* – количество свойств элемента.

На рисунке 12 представлен пример сформированного протокола.

	Дата и время	мс	Уровень	Категория	Сообщение
5	25.07.2018 14:56:34	132014	4	\\WorkStation\sub_DAG\mod_OPC_UA\cntr_test\	0x80050000:Ошибка подключения к Internet сокету: Операция выполняется в данный момент!
6	25.07.2018 14:56:40	951804	1	\\WorkStation\UI\VCAEngine\1-3392e6a2-aed5-490d-b901-4e36c9ed00cc\	Включение сеанса.
7	25.07.2018 14:56:40	953414	1	\\WorkStation\UI\VCAEngine\1-3392e6a2-aed5-490d-b901-4e36c9ed00cc\	Запуск сеанса.
8	25.07.2018 14:56:40	956627	1	\\WorkStation\UI\VCAEngine\	Пользователь 'root' подключился к сессии '1-3392e6a2-aed5-490d-b901-4e36c9ed00cc'.
9	25.07.2018 14:56:41	991081	3	\\WorkStation\sub_UI\mod_Vision\	Ошибка открытия: /dev/input/by-path/platform-pcspkr-event-sprk
10	25.07.2018 14:56:41	9218	0	\\WorkStation\sub_UI\mod_Vision\	Масштаб корневой страницы [1.000000:1.000000].
11	25.07.2018 14:56:43	186255	1	\\WorkStation\UI\VCAEngine\	Пользователь 'root' отключился от сессии '1-3392e6a2-aed5-490d-b901-4e36c9ed00cc'.
12	25.07.2018 14:56:43	186342	1	\\WorkStation\UI\VCAEngine\1-3392e6a2-aed5-490d-b901-4e36c9ed00cc\	Останов сеанса.

Рисунок 12

5.1.7 ПрIMITив формирования отчётной документации (Document)

Реализована поддержка элемента формирования отчётной документации со свойствами:

- гибкое формирование структуры документа на основе языка гипертекстовой разметки, что дает возможность форматирования документов;
- формирование документов по команде или по графику - для формирования отчётной документации в архив с последующим просмотром архива;
- формирование документа в режиме реального времени – для формирования документов полностью динамически и на основе архивов за указанное время;
- использование атрибутов виджета для передачи значений и адресов на архивы в документ. Позволяет использовать виджет документа как шаблон при формировании отчётов с другими входными данными.

В основе любого документа лежит XHTML-шаблон. XHTML-шаблон это тег "body" WEB-страницы, содержащий статику документа в стандарте XHTML 1.0 и элементы исполняемых инструкций на одном из языков пользовательского программирования СКАДА в виде `<?dp<procedure>?>`. Результирующий документ формируется путём исполнения процедур и вставки их результата в документ.

Индивидуальными атрибутами виджета формирования отчетности являются:

- *CSS (style)* – правила CSS в строках, например, для изменения цвета документа необходимо набрать: `"body {background-color:#818181;}"`;
- *шаблон(templ)* – шаблон документа в XHTML. Начинается с тега "body" и включает процедурные вставки:

```
<body docProcLang="JavaLikeCalc.JavaScript">  
  <h1>Значение<?dp return wCod+1.314;?></h1>  
</body>
```

- *документ (doc)* – финальный документ в XHTML. Начинается с тега "body";
- *шрифт (font)* – базовый шрифт текста документа;
- *время начала документа(bTime)*;
- *текущее время (time)* – время генерации документа, записать время для генерации документа от этой точки или нуль для его регенерации;
- *размер архива (n)* – глубина архива, при положительном значении становятся активными атрибуты архива:

- архив:курсор:текущий (aCur) – позиция текущего документа в архиве. Запись значения <0 производит архивацию текущего документа;
- архив:курсор:вид (vCur) – текущий визуализируемый документ архива. Запись значения -1 – выбор следующего документа для отображения, -2 – выбор предыдущего документа для отображения;
- архив:текущий документ (aDoc) – текущий документ архива в XHTML. Начинается с тега “body”;
- архив:размер (aSize) – реальный размер архива документа.

Динамика шаблона документа определяется вставками исполняемых инструкций вида `<?dp {procedure}??>`. В процедурах могут использоваться одноимённые атрибуты виджета и функции пользовательского интерфейса программирования. Кроме атрибутов виджета зарезервированы специальные атрибуты, теги и атрибуты тегов (таблица 5).

Таблица 5

Имя	Назначение
<i>Атрибуты</i>	
rez	Атрибут результата исполнения процедуры, содержимое которого помещается в дерево документа
lTime	Время последнего формирования. Если документ формируется впервые, то <code><lTime> = <bTime></code>
rTime	Содержит время для перебираемых значений в секундах. Определяется внутри тегов с атрибутом "docRept"
rTimeU	Содержит время для перебираемых значений в микросекундах. Определяется внутри тегов с атрибутом "docRept"
rPer	Содержит периодичность перебора значений (атрибут "docRept")
mTime, mTimeU, mLev, mCat, mVal	Определяются внутри тегов с атрибутом "docAMess" при разборе сообщений архива сообщений: mTime - время сообщения; mTimeU - время сообщения, микросекунды; mLev - уровень сообщения; mCat - категория сообщения; mVal - значение сообщения
<i>Специальные теги</i>	
<i>Специальные атрибуты стандартных тегов</i>	
body.docProcLang	Язык исполняемых процедур документа. По умолчанию это JavaLikeCalc.JavaScript
*.docRept="1s"	Тег с указанным атрибутом при формировании размножается путём смещения времени в атрибуте "rTime" на значение указанное в данном атрибуте

Имя	Назначение
.docAMess="1:PLC"	Указывает на необходимость размножения тега с атрибутом сообщения из архива сообщений за указанный интервал времени и в соответствии с уровнем (1) и шаблоном запроса (PLC*). В шаблоне запроса может указываться регулярное выражение в виде <code>{re}</code> . Для данного тега, в процессе размножения, определяются атрибуты: mTime, mTimeU, mLev, mCat и mVal
*.docRevers="1"	Указывает на инвертирование порядка размножения, последний сверху
*.docAppend="1"	Признак необходимости добавления результата выполнения процедуры в тег процедуры. Иначе результат исполнения заменяет содержимое тега
body.docTime	Время формирования документа. Используется для установки атрибута <code><Time></code> при следующем формировании документа. Не устанавливается пользователем!
table.export="1"	Включение возможности экспорта содержимого указанной таблицы в CSV-файл и другие табличные форматы

На рисунке 13 представлен пример сформированного документа.

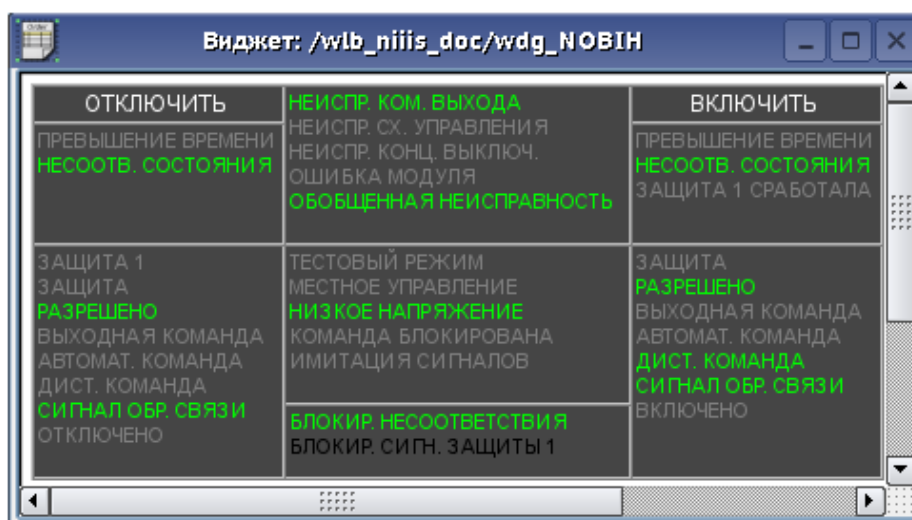


Рисунок 13

5.1.8 Примитив контейнера (Box)

Примитив контейнера используется для формирования составных виджетов и/или страниц пользовательского интерфейса. Данный примитив может включать в себя ссылки на видеокадры из библиотеки, формируя тем самым пользовательские элементы нужной конфигурации. Индивидуальные атрибуты данного примитива:

- *контейнер (box)* – позволяет формировать нужные объекты путём группировки базовых в рамках данного примитива;

- *страница* (*pgOpenSrc*) – элементы, построенные на данном примитиве, могут выполнять роль страницы пользовательского интерфейса. Полный адрес страницы, которая включена внутрь данного контейнера;
- *контейнер страниц* (*pgGrp*) – свойство замещения собственного содержимого другой страницей, в процессе исполнения. Используется для формирования фреймов на страницах пользовательского интерфейса;
- *фон* (*backColor, backImg*) – фон в виде цвета или изображения;
- *бордюр* (*bordWidth, bordColor, bordStyle*) – поддерживает возможность отображения бордюра с указанным цветом, толщиной и стилем.

5.1.9 Примитив поверхность (*Surface*)

Реализована возможность построения трехмерной поверхности по заданным пользователем или архивным данным.

Индивидуальными атрибутами являются:

- *источник* – исходные данные в формате:

`prm:Модуль/контроллер/параметр/атрибут`

Например: `prm:TANGO/tango_test/surf/FourtyEight`). Данные, обрабатываемые виджетом, должны быть представлены в виде объекта типа *Matrix* или *Array*;

- *период слежения* - позволяет задавать интервал обновления виджета;
- *пауза* – позволяет зафиксировать изображение поверхности в момент времени выставления значения этого атрибута в единицу. Так же, если значение данного атрибута установлено в единицу или `true`, то виджет не будет обновлять форму поверхности;
- *архиватор значений* – позволяет отображать поверхность по данным, хранящимся в архиве в формате «Модуль Архивов.Id архиватора». Данный атрибут работает совместно с атрибутом «Промежуток времени» (чтобы поверхность строилась по архивируемым данным, необходимо указывать оба атрибута);
- *промежуток времени* – позволяет указать промежуток времени архивируемых данных, по которым будет строиться поверхность. Значение этого атрибута устанавливаются согласно следующему формату:
«начало (sec) начало (usec) конец (sec) конец (usec)».

В случае если в указанном интервале архивируемые данные различны, то будет построена лишь та поверхность, которая соответствует архивируемым данным с меткой времени наиболее близкой к конечной. Данный атрибут работает совместно с атрибутом «Архиватор значений»;

- *поверхность* – предназначен для изменения визуальных стилей поверхности и цвета рендерной сетки, в случае наличия последней. Атрибут стиль может принимать следующие значения:

- не рисовать;
- каркас – поверхность представлена в виде прозрачной сетки, позволяющей видеть все точки поверхности одновременно (рисунок 14а);
- непрозрачная сетка – поверхность представлена в виде непрозрачной сетки, отображающей только точки поверхности образующие ее передний план (рисунок 14б);
- Цветная – так называемая радужная раскраска, цвета которой распределяются в вертикальном направлении снизу вверх по цветам радуги – от фиолетового до красного (рисунок 14в);
- Цветная + сетка – закрашенная поверхность с накладываемой на неё рендерной сеткой (рисунок 14г).

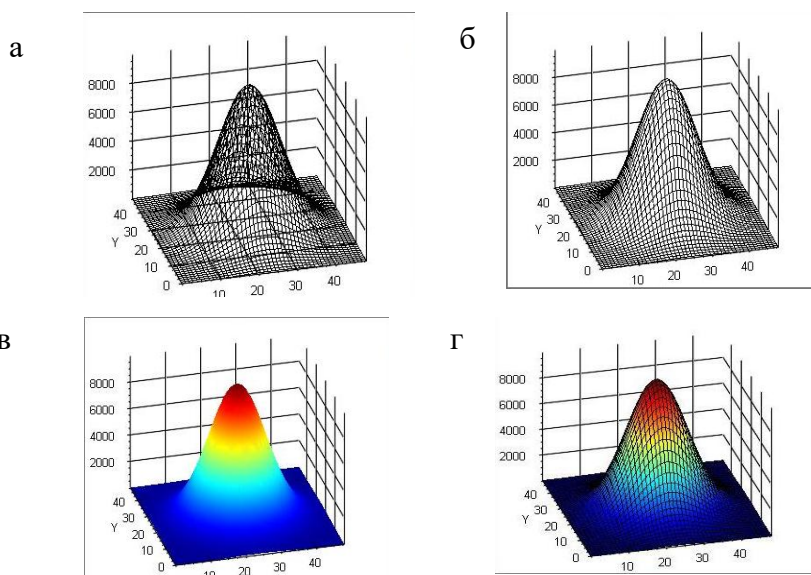


Рисунок 14

- *Шкалы* – предназначен для изменения параметров каждой из трех осей поверхности и представлен вложенным набором атрибутов, позволяющими задать стили отображения осей (такие как не рисовать, каркас, коробка), имен осей, числовых значений;

- *Расположение* – предназначен для детальной настройки положения отображаемой поверхности (угол поворота, смещение, масштаб, зуммирование, расположение по умолчанию);
- *Курсор* – предназначен для установки параметров трассировщика: координаты курсора X и Y в трехмерном пространстве, ширину линии и цвет трассировщика;
- *Значение* – предназначен для хранения Z координаты под курсором мыши.

5.2 Графический редактор для виджетов, основанных на примитиве элементарная фигура

Для редактирования виджетов, основанных на примитиве элементарная фигура, в ПП «СКАДА А-СОФТ» используется векторный графический редактор, позволяющий изображать объекты, характеристики которых могут быть динамически изменены. Вызов графического редактора производится в окне редактирования виджета, основанного на примитиве элементарная фигура, выбором строки «Вход в редактирование виджета» (при нажатии на правую кнопку мыши). Основными элементами графического редактора являются три графических примитива: линия, дуга, кривая Безье (рисунок 15). К динамически изменяющимся характеристикам этих примитивов относятся:

- координаты контрольных точек; используются для задания формы линии, дуги или кривой Безье. При этом линия имеет — 2 контрольные точки, дуга — 5 контрольных точек (1 - начало дуги, 2 - конец дуги, 3 - центр окружности, 4 - середина дуги (для круга - четверть окружности), 5 - контрольная точка (для круга – половина окружности)), кривая Безье — 4;
- ширина линии;
- ширина бордюра (границы);
- цвет бордюра (границы);
- стиль линии (сплошная, пунктирная, точечная).

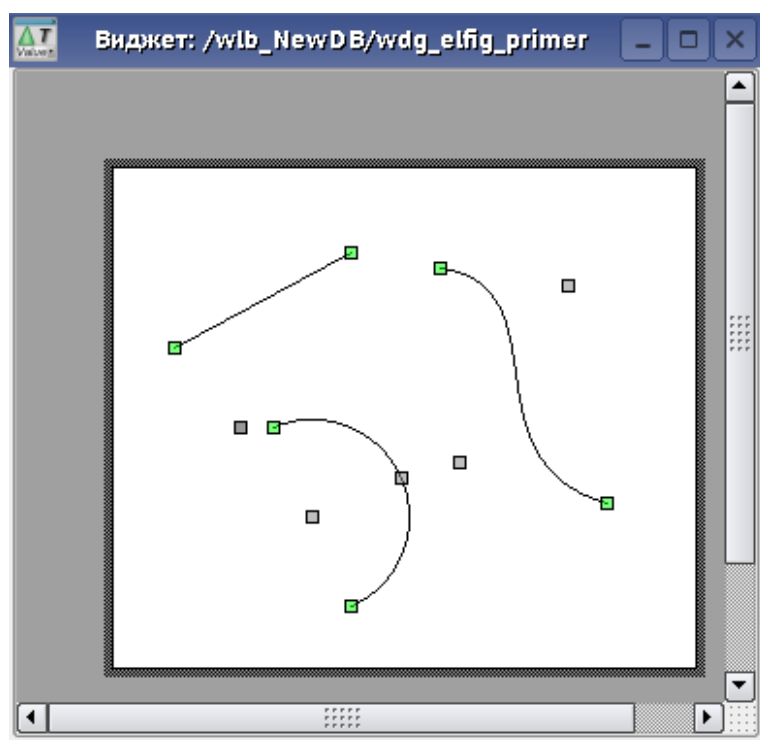



Рисунок 15

Комбинируя указанные элементарные фигуры можно создать более сложные графические объекты. Если связанные примитивы образуют замкнутый контур, то он может быть залит цветом и/или изображением.

К возможностям редактора также относятся выделение, перемещение, копирование и удаление фигур.

Для наглядного управления свойствами элементарной фигуры можно вызвать «Диалог свойств элементарной фигуры» (рисунок 16). Для этого необходимо выделив фигуру нажать правую кнопку мыши и выбрать строку «Показать свойства элементарной фигуры». В результате появится окно диалога, в котором будут отображаться данные выбранной фигуры. Также, предусмотрена возможность включать/исключать отдельные свойства диалога (кнопка ). В случае исключения отдельных свойств они не будут обрабатываться при подтверждении диалога (кнопка «Принять»). При подтверждении диалога все указанные данные для включенных свойств будут применены для всей группы фигур. Диалог для свойств заливки позволяет управлять свойствами отдельной заливки. Кнопки «Дин/Стат» делают соответствующие свойства динамическими либо статическими.

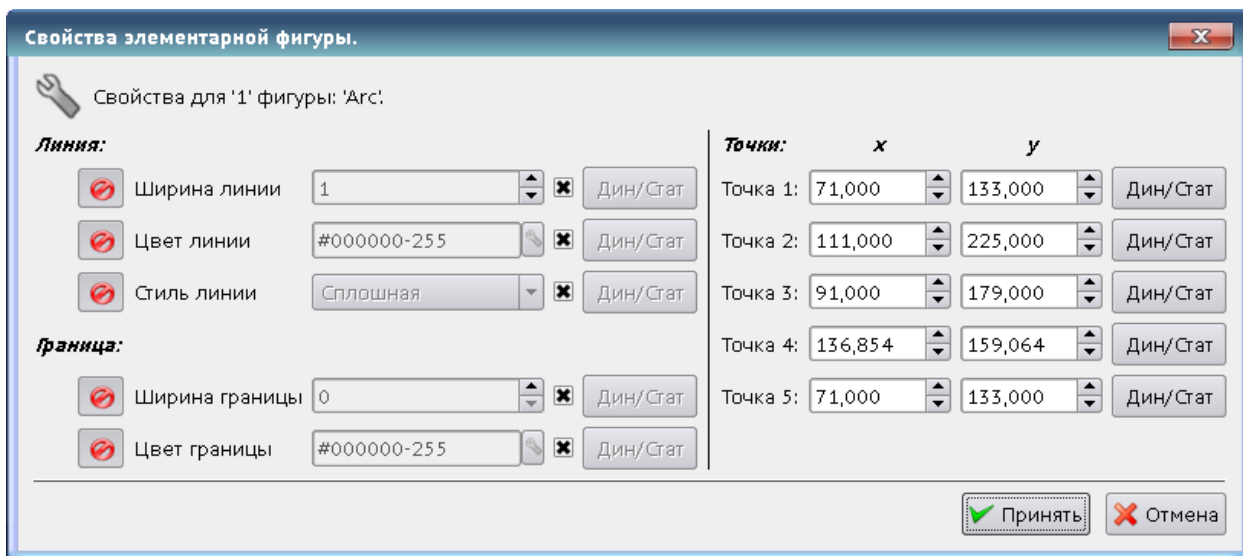


Рисунок 16

5.3 Библиотека основных элементов пользовательского интерфейса (Main)

В своём составе библиотека содержит около двух десятков графических элементов, наиболее востребованных при формировании пользовательского интерфейса управления технологическим процессом (рисунок 17). Для использования большинства элементов библиотеки необходимо добавить виджет на мнемосхему и связать с параметром источника данных (см. часть 3 настоящего руководства).

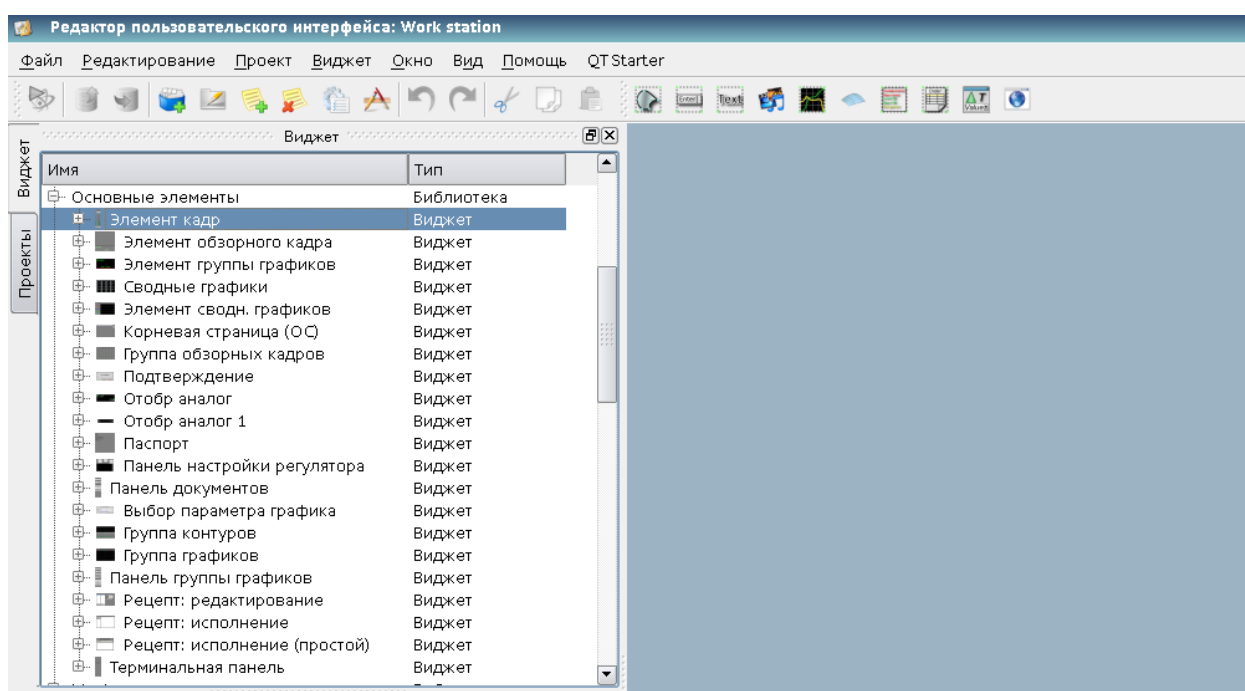
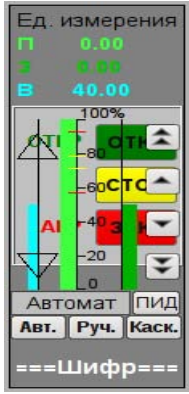
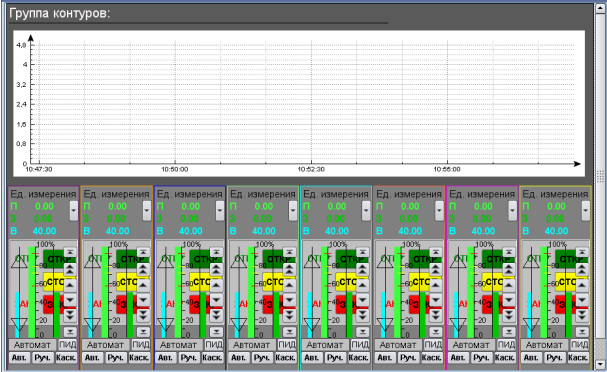


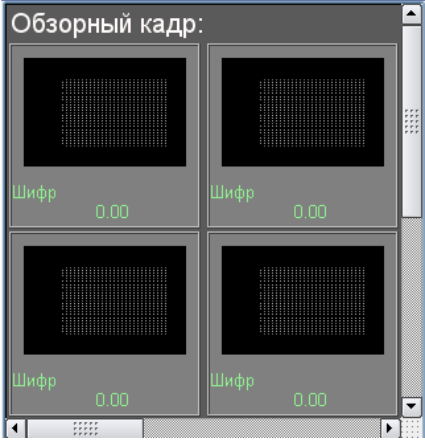
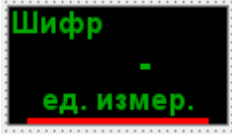
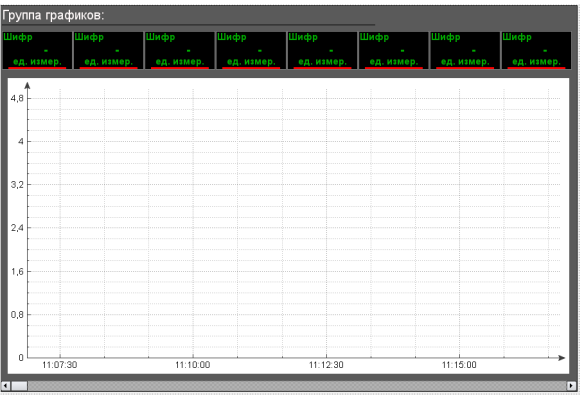
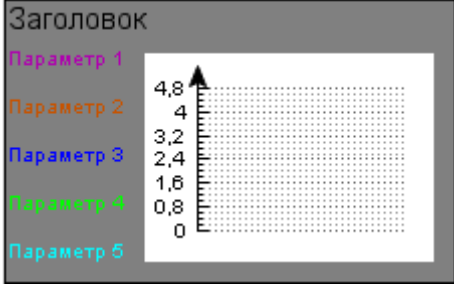
Рисунок 17

В таблице 6 приведено описание элементов библиотеки.

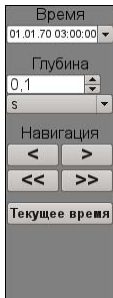
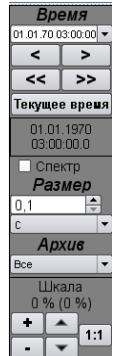
Таблица 6


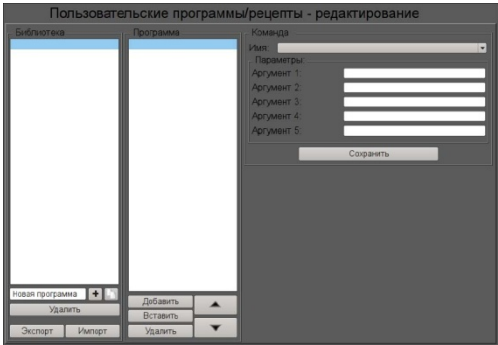
Графическое изображение	Описание
	<p><i>Отобр аналог</i></p> <p>Поле отображения аналоговых параметров, таких как:</p> <ul style="list-style-type: none"> – текущие значения параметров, – наличие сигнализации о превышении уставок, – наличие сигнализации о неисправностях измерительного канала. <p>Текущие значения параметра выводятся в виде чисел с плавающей запятой. Формат чисел определяется при проектировании.</p>
	<p><i>Отобр аналог 1</i></p> <p>Служит для отображения текущего</p>

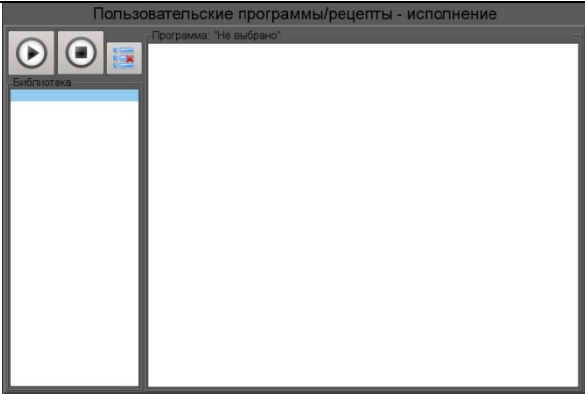
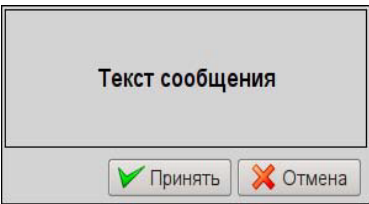
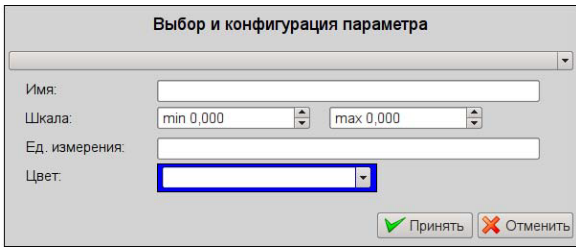
Графическое изображение	Описание
	<p>значения аналогового параметра с односимвольным префиксом типа измеряемой величины.</p> <p><i>Элемент кадр</i></p> <p>Элемент является универсальной панелью управления различными устройствами: аналоговыми: показания, ручные вводы значений и регуляторы (аналоговые и импульсные); дискретными: клапаны, отсекатели, задвижки, двигатели, вентиляторы и всевозможные переключатели.</p> <p>Заложен в шаблоне проекта "Объекты сигнализации" и, если новый проект создаётся на основе этого шаблона, вызов осуществляется автоматически.</p>
	<p><i>Группа контуров</i></p> <p>Элемент служит для одновременного наблюдения и управления несколькими контурами (до восьми), включает в себя как экземпляры виджета "Элемент кадр" для каждого контура, так и виджет "Диаграмма" для наблюдения за трендами контуров и просмотра истории. Предназначен для выполнения роли страницы-шаблона и должен непосредственно помещаться в дерево проекта.</p>
	<p><i>Элемент обзорного кадра</i></p> <p>Служит основой обзорного кадра, отражает текстовую информацию о параметре в виде наименования и значения, а также график (тренд) параметра за небольшой (настраиваемый) промежуток времени для наблюдения за текущей тенденцией поведения параметра с авто-масштабированием по шкале значения. Данный виджет не предназначен для самостоятельного использования, в отрыве от обзорного кадра, но использовать его можно, например, поместив на мнемосхему и установив связь с параметром источника данных.</p>

Графическое изображение	Описание
	<p><i>Группа обзорных кадров</i></p> <p>Служит для отображения текущих трендов по параметрам объекта сигнализации в количестве до 24 штук, поддерживает функцию масштабирования элементов в зависимости от их количества. Состоит из виджетов "Элемент обзорного кадра". Виджет предназначен для выполнения роли страницы-шаблона и должен непосредственно помещаться в дерево проекта.</p>
	<p><i>Элемент группы графиков</i></p> <p>Служит для создания групп графиков. Содержит информацию о параметре, режим регулятора, если параметр является таковым, единицы измерения аналогового параметра, а также цвет, соответствующий параметру тренда.</p>
	<p><i>Группа графиков</i></p> <p>Служит для одновременного наблюдения тренда и управления параметрами объекта сигнализации, включает в себя как экземпляры виджета "Элемент группы графиков" для каждого параметра, так и виджет "Диаграмма" для наблюдения за графиками параметров и просмотра истории, а также горизонтальную полосу прокрутки для быстрой навигации по доступной истории выбранных для отображения параметров. Виджет предназначен для выполнения роли страницы-шаблона и должен непосредственно помещаться в дерево проекта. К каждому кадру может подключаться до восьми параметров путём установки связей.</p>
	<p><i>Элемент сводных графиков</i></p> <p>Элемент позволяет отображать тренды по пяти параметрам за указанный промежуток времени и до текущего времени.</p>

Графическое изображение	Описание
	<p><i>Сводные графики</i></p> <p>Служит для отображения трендов основных параметров по всему проекту визуализации. Предназначен для выполнения роли страницы-шаблона и должен непосредственно помещаться в дерево проекта. К каждому виджету-видеокадру может подключаться до 16*5 параметров путём установки связей. Графики, для которых не будут установлены связи, будут скрыты при исполнении, также возможно масштабирование подключенных графиков для заполнения области всего виджета.</p>
	<p><i>Панель настройки регулятора</i></p> <p>Служит для настройки ПИД регулятора, включает в себя информацию о параметре-регуляторе, поля настроек регулятора, и виджет "Диаграмма" для наблюдения за трендами регулятора и просмотра истории. Может использоваться как в роли панели, вызываемой из панели управления параметрами "EiCadr", так и в роли страницы-шаблона. Виджет должен непосредственно помещаться в контейнер панелей дерева проекта, где будет осуществляться динамическая линковка на параметр регулятора. Для создания статического перечня контуров настроек регуляторов, с возможностью последующего листания по ним, необходимо поместить их в контейнер контуров регуляторов "greg" каждого объекта сигнализации и статически связать с соответствующим параметром, а также обеспечить равенство идентификатора панели и связанного параметра.</p>
	<p><i>Корневая страница</i></p> <p>Служит базой для создания пользовательских интерфейсов управления технологическими процессами, основанными на объектах сигнализации. Корневая страница содержит четыре области:</p> <ol style="list-style-type: none">1 "Область кнопок-индикаторов объектов сигнализации" (вверху) — служит для предоставления информации о наличии аварий в объекте сигнализации, а также для переключения между ними.2 "Область кнопок-режимов отображения" (справа-вверху) — индикация выбора и выбор режима отображения. Содержит также кнопки квитации, которые появляются при возникновении нарушений и кнопки перелистывания страниц мнемосхем.3 "Контейнер мнемосхем и основных кадров интерфейса оператора" (в центре) — область

Графическое изображение	Описание
	<p>контейнера для включения в неё мнемосхем и основных кадров при выборе их кнопками режимов отображения или смене объекта сигнализации.</p> <p>4 "Контейнер панелей управления" (справа внизу) — область контейнера для включения в ней панелей управления различными объектами в области контейнера мнемосхем, например: панель параметра, документа, графика и т.д.</p> <p>Под контейнером панелей управления располагается кнопка запуска демонстрационного режима – режима, при котором осуществляется периодическое переключение показательных кадров, изменение режимов и других операций согласно сценарию.</p> <p>Данный виджет может использоваться только в режиме корневой страницы, которая должна помещаться в дерево проекта как элемент "/*/so".</p>
	<p><i>Паспорт</i></p> <p>Служит для отображения паспорта параметра: детальной информации, включающей шифр, описание, единицы измерения, аварийные границы и т.д. Документ формируется полностью динамически. Данный элемент должен помещаться в логический контейнер панелей дерева проекта. В режиме редактирования этот виджет представляет собой пустой "Документ". Связывание с параметром осуществляется динамически при вызове из элементов представления данных параметра.</p>
	<p><i>Панель документов</i></p> <p>Служит для управления документами и навигации по их истории. Элементом поддерживаются динамические и архивные документы. Данный элемент должен помещаться в логический контейнер панелей дерева проекта. Связывание с параметром осуществляется динамически при вызове из элемента документа.</p>
	<p><i>Панель группы графиков</i></p> <p>Служит для управления виджетом "Диаграмма", позволяет просмотреть историю графиков за определенный период времени в нужном разрешении, поддерживается масштабирование шкалы, выбор архиваторов для отображения, а также представление графиков в виде спектра присутствующих частот.</p> <p>Данный элемент должен помещаться в логический контейнер панелей дерева проекта.</p>

Графическое изображение	Описание
	<p>Связывание с параметром осуществляется динамически при вызове из элемента диаграмма.</p>
	<p><i>Терминальная панель</i> Служит для заполнения пустого места, когда не выбран элемент для управления. Элемент должен помещаться в логический контейнер панелей дерева проекта.</p>
	<p><i>Рецепт: редактирование</i> Служит для пользовательского редактирования программ-рецептов. Программа-рецепт представляет собой последовательный вызов блоков функций - команды (макросы), принимающие до пяти аргументов и возвращающие строку результата, с кодом завершения вначале: "Работа" (0), "Завершен" (> 0) и "Ошибка" (< 0). Кадр "Рецепт: редактирование" содержит слева на право: "Библиотека" — библиотека со списком программ и элементами управления библиотекой. "Программа" — список шагов-команд выбранного в библиотеке рецепта-программы с элементами управления. "Команда" — поле редактирования выбранного шага рецепта в составе выбора команды и установки значений доступных атрибутов, а также кнопки сохранения изменений. Данный кадр должен быть помещен в логический контейнер мнемосхем или панелей дерева проекта.</p>
	<p><i>Рецепт: исполнение</i> Служит для непосредственного исполнения или наблюдения за исполнением во внешнем вычислителе программ-рецептов, ранее сформированных в кадре Рецепт: редактирование. Кадр "Рецепт: исполнение" содержит слева на право: "Запуск/останов/пропуск" — две кнопки запуска и останова выбранной программы, а также кнопка пропуска выполнения текущего шага. "Библиотека" — библиотека со списком программ. "Программа" — документ списка шагов-команд выбранного в библиотеке рецепта-программы. При исполнении в этом поле отслеживается текущее состояние исполнения</p>

Графическое изображение	Описание
	<p>путём соответствующей подсветки шагов.</p> <p>Исполняемый рецепт-программа может быть приостановлен путём нажатия кнопки "Пауза" в месте кнопки "Запуск" или прерван путём нажатия кнопки "Останов". Также возможно пропустить шаг, нажав кнопку "Пропустить", в момент исполнения шага.</p> <p>По любому завершению рецепта-программы происходит генерация сообщения с параметрами сеанса, а также архивирование документа сеанса. По умолчанию архив документов настроен на глубину 10 документов.</p>
	<p><i>Подтверждение</i> - реализует простейший диалог подтверждения операций. Элемент содержит сообщение с вопросом и две кнопки "Принять" и "Отмена". Данный виджет может быть использован разработчиком при создании кадров динамического взаимодействия в операциях, требующих подтверждения у пользователя. Для использования нужно добавить данный элемент в логический контейнер панелей дерева проекта.</p>
	<p><i>Выбор параметра графика</i> реализует диалог выбора источника данных, часто архивных, для формирования графика в кадре "Группа графиков". Выбор предоставляется из перечня указанного в атрибуте "Параметры доступные для выбора" кадра-инициатора. Для выбранного источника можно указать имя, шкалу, единицу измерения и цвет графика.</p>

5.4 Библиотека «Элементы мнемосхемы» (mnEls)

Библиотека строится на основе примитивов виджетов и модуля JavaLikeCalc, позволяющего создавать вычисления на Java-подобном языке.

Для подключения библиотеки «Элементы мнемосхем» пользовательского интерфейса к проекту необходимо в панели управления графического редактора выбрать пункт Виджеты и из всплывающего меню выбрать библиотеку mnEls (рисунок 18).

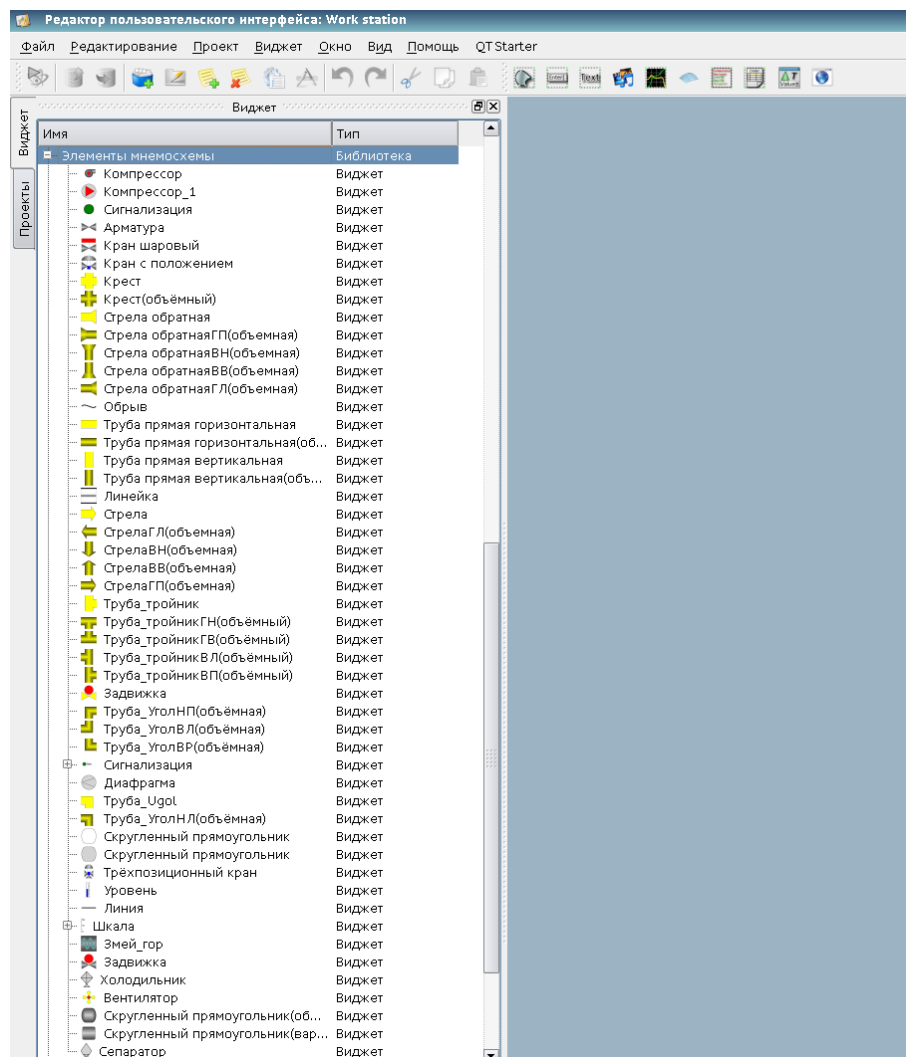


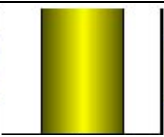
Рисунок 18

Библиотека содержит 50 графических элементов, необходимых при формировании мнемосхем пользовательского интерфейса управления технологическим процессом. Большинство элементов имеют квадратную форму, позволяющую легко поворачивать и масштабировать их при необходимости (угол поворота всех виджетов по умолчанию равен "0"). Некоторые из них содержат скрипт, описывающий их поведение.

5.4.1 Элементы трубопровода

В таблице 7 приведен перечень элементов, при помощи которых можно выстроить трубопровод любой сложности. По умолчанию все элементы залиты желтым цветом (объемные - полупрозрачными изображениями в градациях серого). Соответствуют ГОСТ 21.206-93.

Таблица 7

Условное изображение		Описание
плоское	объемное	
		Крест
		Труба прямая горизонтальная
		Труба прямая вертикальная
		Стрела. Виджеты объемного изображения представлены в четырех вариантах в соответствии с разными углами поворота
		Стрела обратная. Виджеты объемного изображения представлены в четырех вариантах в соответствии с разными углами поворота
		Труба_Угол. Виджеты объемного изображения представлены в четырех вариантах в соответствии с разными углами поворота.
		Труба_тройник. Виджеты объемного изображения представлены в четырех вариантах в соответствии с разными углами поворота

Условное изображение		Описание
плоское	объемное	
		Труба_УголСкруглНЛ(объемная)
		Труба_УголСкруглНП(объемная)
		Труба_УголСкруглВЛ(объемная)
		Труба_УголСкруглВП(объемная)

5.4.2 Элементы, изображающие различные технологические устройства

В таблице 8 приведен перечень элементов библиотеки «Элементы мнемосхемы» - изображений технологических устройств, часто встречающихся при построении мнемосхем различных технологических процессов. В таблице 9 описаны параметры связывания элементов с источниками данных.

Таблица 8

Графическое обозначение	Описание
	Компрессор
	Компрессор_1
	Задвижка

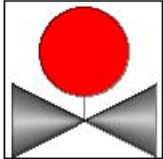

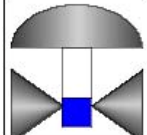
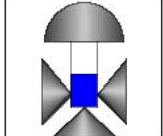
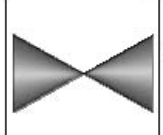
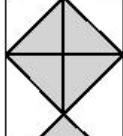
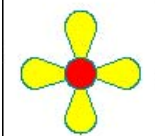
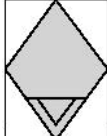
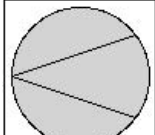
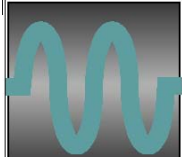
Графическое обозначение	Описание
	Задвижка (объемная)
	Кран шаровый. Включает время хода и время отрыва. В зависимости от состояния меняет цвет.
	Кран с положением. В зависимости от состояния меняет форму.
	Трёхпозиционный кран. В зависимости от состояния меняет форму.
	Арматура
	Холодильник
	Вентилятор
	Сепаратор. Модель сепаратора с двумя фазами, жидкой и газовой
	Диафрагма
	Змей_гор(теплообменник) Модель теплообменника, рассчитывающая теплообмен двух потоков

Таблица 9

ID	Параметр	Тип	Конфигурация	Конфигурационный шаблон	Описание
<i>Виджет "Кран шаровый" (El_Kran_Sh)</i>					
com	Команда	Логический	Полная связь	Parametr com	Команда на закрытие/открытие
shifr	Шифр	Строка	Полная связь	Parametr NAME	Короткое имя, шифр, параметра
st_close	Состояние "Закрыто"	Логический	Полная связь	Parametr st_close	Закрытое состояние крана
st_open	Состояние "Открыто"	Логический	Полная связь	Parametr st_open	Открытое состояние крана
<i>Виджет "Кран с положением" (El_Kran_polozh)</i>					
out	Положение	Вещественный	Входная связь	Parametr out	Степень открытия/закрытия крана
<i>Виджет "Трёхпозиционный кран" (Kran_3_pos)</i>					
out	Положение	Вещественный	Входная связь	Parametr out	Степень открытия/закрытия крана
<i>Виджет "Компрессор" (Compressor)</i>					
com	Команда	Логический	Полная связь	Parametr com	Команда на пуск/останов

Для виджетов "Кран шаровый", "Кран с положением", "Трёхпозиционный кран" описана процедура открытия/закрытия, доступная для просмотра и редактирования на вкладке «Обработка» виджета (рисунок 19).

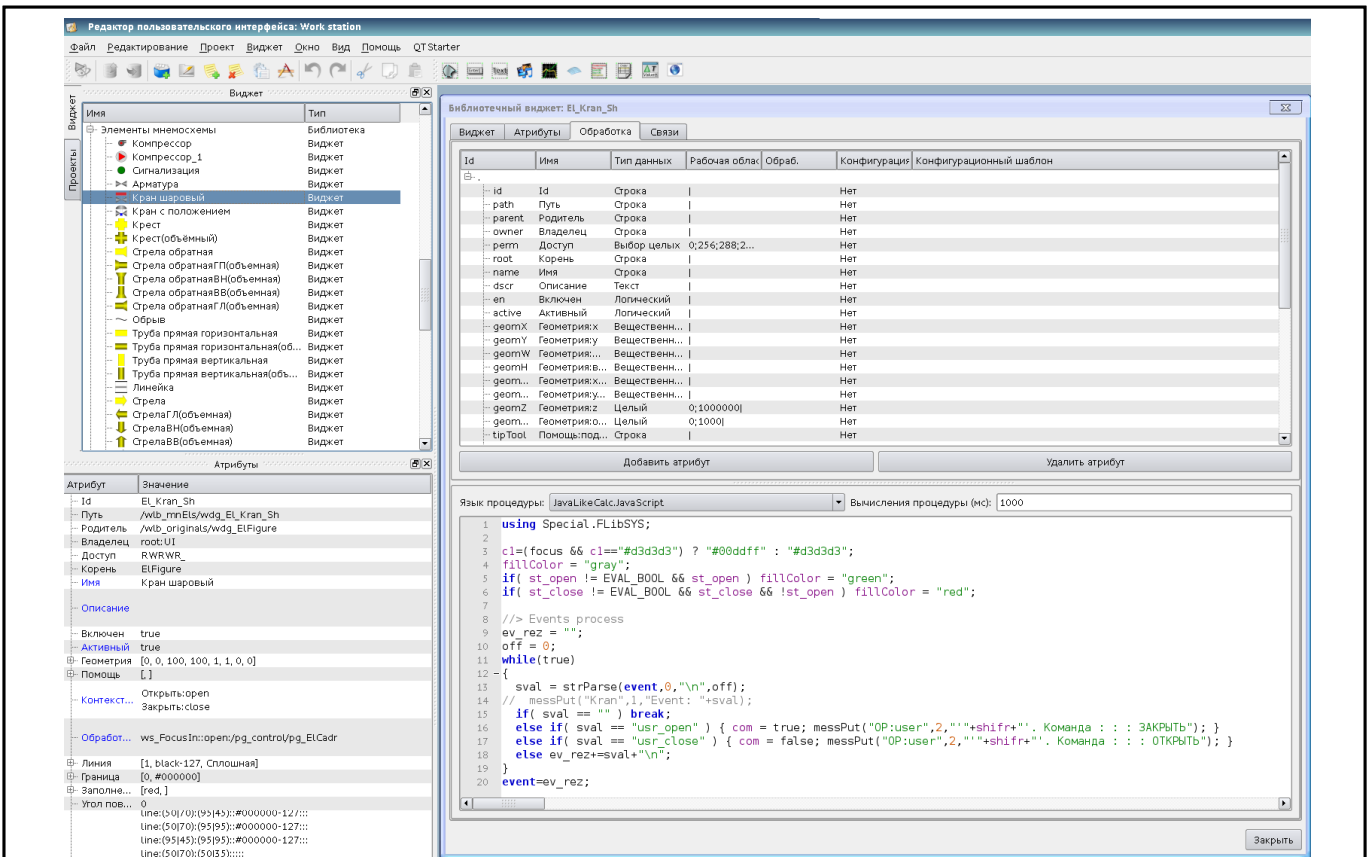
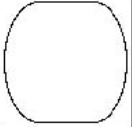




Рисунок 19

5.4.3 Другие элементы

В таблице 10 приведен перечень вспомогательных элементов для построения мнемосхем. Некоторые из них содержат скрипт, описывающий их поведение. В таблице 11 представлены параметры связывания виджетов с источниками данных.

Таблица 10

Условное обозначение	Описание
	Скругленный прямоугольник
	Скругленный прямоугольник (объемный)
	Скругленный прямоугольник (вариант 2)

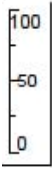
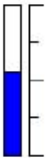


Условное обозначение	Описание
	Шкала
	Уровень
	Линия
	Сигнализация

Таблица 11

ID	Параметр	Тип	Конфигурация	Конфигурационный шаблон	Описание
<i>Виджет "Уровень"</i>					
max	Максимум	Вещественный	Входная связь	Parametr max	Максимум шкалы
min	Минимум	Вещественный	Входная связь	Parametr min	Минимум шкалы
var	Значение	Вещественный	Входная связь	Parametr var	Значение уровня

5.5 Библиотека электроэлементов мнемосхем пользовательского интерфейса (ElectroEls)

Библиотека электроэлементов создана для предоставления электроэлементов мнемосхем пользовательского интерфейса, строится на основе примитивов виджетов и модуля JavaLikeCalc.

Для подключения библиотеки «Электроэлементов» к проекту необходимо в панели управления графического редактора выбрать Виджеты и из всплывающего меню выбрать библиотеку ElectroEls (рисунок 20).

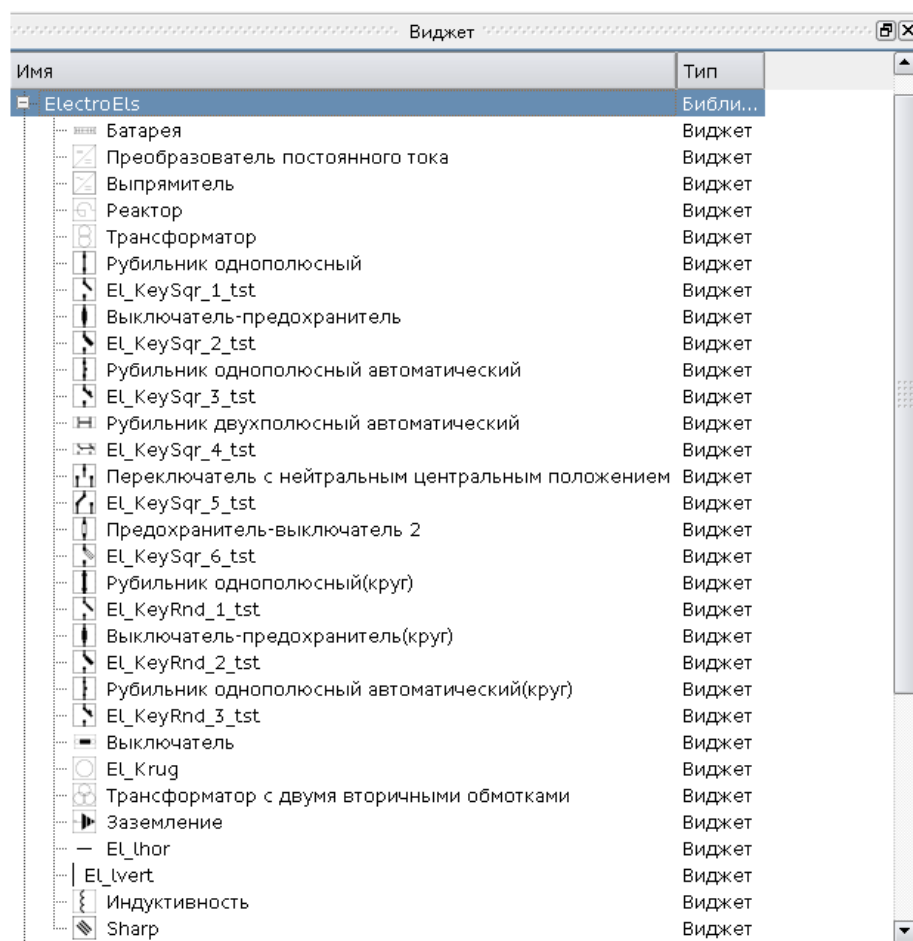


Рисунок 20

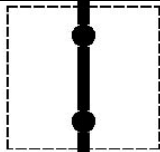
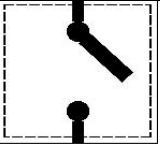
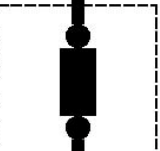
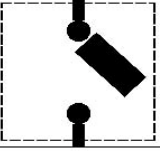
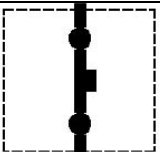
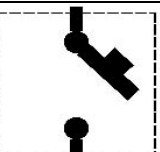
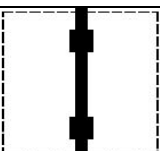
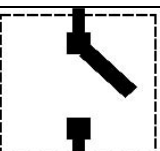
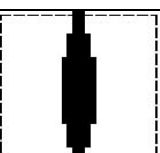
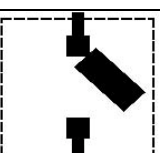
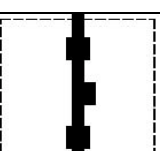
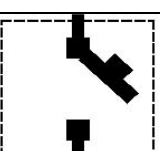

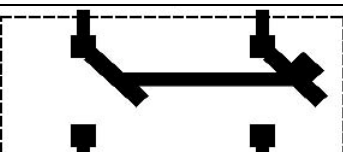
В своём составе библиотека содержит более 20 графических элементов, используемых при формировании мнемосхем пользовательского интерфейса управления технологическим процессом в области электроэнергетики.

По умолчанию все виджеты имеют масштаб по обеим осям, равный "1", а их угол поворота составляет "0" градусов. Возможно изменение значений указанных атрибутов для задания желаемых пропорций.

5.5.1 Динамические элементы библиотеки

В таблице 12 приведен перечень различного вида выключателей и переключателей в различном состоянии. В таблице 13 приведены параметры, посредством которых осуществляется связь элемента с источником данных.

Таблица 12

Графическое изображение элемента		Описание
состояние включен	состояние выключен	
		Рубильник однополюсный(круг)
		Предохранитель- выключатель(круг)
		Рубильник однополюсный автоматический(круг)
		Рубильник однополюсный
		Предохранитель- выключатель
		Рубильник однополюсный автоматический
		Рубильник двухполюсный автоматический

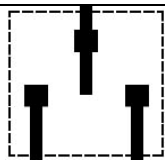
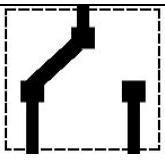
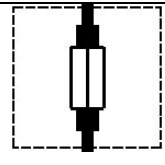
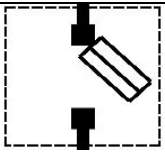
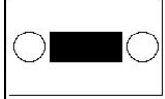
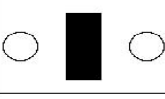
Графическое изображение элемента		Описание
состояние включен	состояние выключен	
		Переключатель с нейтральным центральным положением
		Предохранитель-выключатель 2
		Выключатель

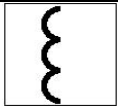
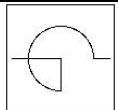
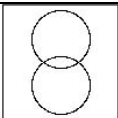
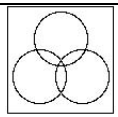
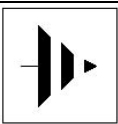



Таблица 13

ID	Параметр	Тип	Конфигурация	Конфигурационный шаблон	Описание
<i>Виджеты: "Рубильник однополюсный(круг)" (El_Key_1), "Предохранитель-выключатель(круг)" (El_Key_2), "Рубильник однополюсный автоматический(круг)" (El_Key_3)</i>					
val	Значение	Логический	Входная связь	Parameter val	
<i>Виджеты: "Рубильник однополюсный" (El_KeySqr_1), "Предохранитель-выключатель" (El_KeySqr_2), "Рубильник однополюсный автоматический" (El_KeySqr_3), "Рубильник двухполюсный автоматический" (El_KeySqr_4), "Предохранитель-выключатель 2" (El_KeySqr_6)</i>					
val	Значение	Логический	Входная связь	Parameter var	
DESCR	Описание	Строка	Входная связь	Parameter DESCR	
st	Статус ошибки	Логический	Входная связь	Parameter st	
<i>Виджет "Переключатель с нейтральным центральным положением" (El_KeySqr_5)</i>					
val	Значение	Логический	Входная связь	Parameter var	
val1	Значение	Логический	Входная связь	Parameter var	
st	Статус ошибки	Логический	Входная связь	Parameter st	

5.5.2 Статические элементы библиотеки «Электроэлементы»

В таблице 14 представлены статические элементы библиотеки. Графическое изображение элементов соответствует требованиям ГОСТ 2.723-68.

Таблица 14


Графическое изображение	Описание
	Индуктивность
	Реактор
	Трансформатор
	Трансформатор с двумя вторичными обмотками
	Заземление
	Батарея
	Выпрямитель
	Преобразователь постоянного тока

5.6 Элементы библиотеки «NT-tmp»

Библиотека «NT-tmp» содержит элементы технологического оборудования, разработанные для АСУ ТП нефтеперерабатывающей промышленности (таблица 15).

Таблица 15

Графическое изображение	Описание
	<p><i>Аналог</i></p> <p>Поле отображения аналоговых параметров. В зависимости от типа параметра окрашен в разные цвета:</p> <ul style="list-style-type: none"> температура (Т) -зеленый; давление (Р) – серый; уровень (L) – синий; расход (F) – розовый.
	<p><i>Гистограмма аналогового параметра</i></p> <p>Справа от шкалы гистограммы находятся блоки, отвечающие за отображение срабатывания дискретных уставок (верхних/нижних аварийных/ предупредительных/физических). Внутри блоков находятся значения уставок. Блоки соединены со шкалой линиями, цвет которых зависит от типа уставки:</p> <ul style="list-style-type: none"> - аварийная уставка – красная линия, - предупредительная уставка – жёлтая линия, - физическая уставка – белая линия
	<p><i>Насос</i></p> <p>Цвет пиктограммы зависит от режима управления и активности насоса.</p>
	<p><i>Электрозадвижка</i></p> <p>Состояние задвижки зависит от ее формы и цвета.</p>
	<p><i>Клапан отсечной</i></p> <p>Изображение зависит от состояния клапана: открыт, закрыт, промежуточное положение, неопределенное положение (одновременно присутствуют сигналы открытого и</p>

Графическое изображение	Описание
	<p>закрытого состояния задвижки).</p> <p><i>Клапан регулирующий</i></p> <p>Рядом с клапаном изображается регулируемый параметр. Состояние задвижки зависит от ее формы и цвета.</p>
	<p><i>Аппарат воздушного охлаждения</i></p> <p>В зависимости от состояния аппарата элемент окрашивается в разные цвета: зеленый - включен, розовый - выключен.</p>

На мнемосхемах состояние технологического оборудования индицируется изменением цветовой гаммы пиктограммы следующим образом:

- зеленый цвет – кран открыт, авария отсутствует;
- красный цвет – кран закрыт, аккумулятор включен, питание датчика – включено, авария питания/ связи/аккумулятора/220;
- желтый цвет – кран в промежуточном состоянии;
- серый цвет – аккумулятор отключен, питание датчика отключено.

По всем элементам таблицы выводится окно управления. Окно управления аналогом представлено на рисунке 21. Реализация окна зависит от протокола связи нижнего уровня (для протокола EN см. часть 4 настоящего руководства).

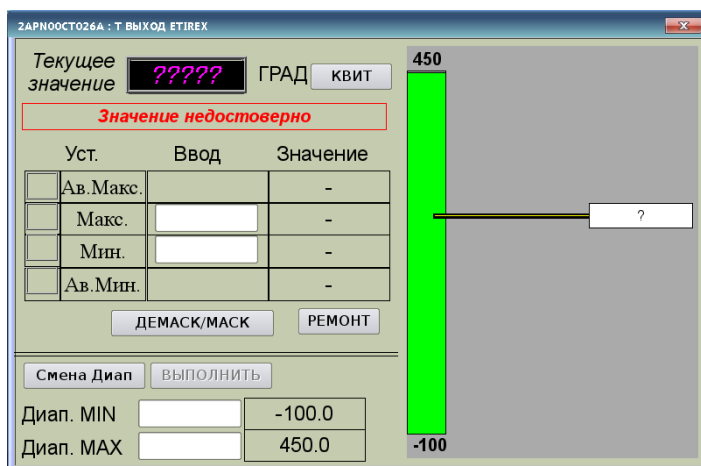


Рисунок 21

Перечень принятых сокращений

АСУ ТП	Автоматизированная система управления технологического процесса
ПИД	Пропорционально-интегрально-дифференцирующий регулятор
ПТС	Программно-технические средства
СВУ	Среда визуализации и управления
ТПТС	Типовое программно-техническое средство
SCADA	Диспетчерское управление и сбор данных (Supervisory Control And Data Acquisition)

