Узлы проекта AgavaSCADA/AgavaPLC

□

Содержание

- Узлы проекта
  - Проект
  - Станция
  - Транспортная система
    - Транспорт
  - Система визуализации
    - Фрейм
    - Окно
    - Композиция
    - Слой
    - Виджет
    - Меню
    - Создание многомониторных систем визуализации
  - Виджеты
    - Виджет «Текстовая метка»
      - Отображение нескольких значений
      - Модификаторы спецификаторов формата
        - Модификатор минимальной ширины поля
        - Модификатор точности
    - Виджет «Условная метка»
    - Виджет «Изображение»
    - Виджет «Условное изображение»
    - Виджет «Движущееся изображение»
    - Виджет «Кнопка»
    - Виджет «Гистограмма»
    - Виджет «Исторический график»
      - Легенда
      - Режимы просмотра графиков
      - Визирная линия
    - Виджет «График кусочно-линейной функции»
    - Виджет «Список событий»
    - Виджет «Ввод значения»
    - Виджет «Выбор значения из списка»

- Виджет «Флаг»
- Виджет «Отчет»

Действия
- Действие «Воспроизведение звукового файла»
- Действие «Смена композиции»
- Действие «Отображение окна»
- Действие «Закрытие окна»
- Действие «Запись значения в узел»
- Действие «Создание события»
- Действие «Команда узлу»
- Действие «Запуск процесса»
- Действие «Авторизация пользователя»
- Действие «Переключение видимости окна»
- Действие «Выполнение скрипта»
- Действие «Отображение меню»

Система тревог
- Группа событий
- Контроллер событий
- Порядок настройки системы

Система архивации
- Порядок настройки системы
- Архиватор SQLite
  - Свойства узла "Архиватор SQLite"
- Архиватор PostgreSQL

Система журналирования
- Порядок настройки системы
- Логгер SQLite
- Логгер файловый
- Логгер SMS

Система проверки прав пользователей
- Роли и пользователи
- Привилегии
- Порядок настройки системы

Сигналы
- Сигнал
  - Сигнал простой
  - Постоянная
  - Массив
  - Сигнал комплексный

- Источник
  - Регистр Modbus
  - Источник индексный
- Операция
  - Операция «Сложение»
  - Операция «Вычитание»
  - Операция «Умножение»
  - Операция «Деление»
  - Операция «Условие»
  - Операция «Пара условий»
  - Операция «Логическое ИЛИ»
  - Операция «Логическое И»
  - Операция «Логическое НЕ»
  - Операция «Таймер»
  - Операция «Мультиплексор»
  - Операция «Демультиплексор»
  - Операция «Генератор случайного числа»
  - Операция «Скрипт C++»
- Объект

Задачи
- Задача ПЛК
  - Блоки задачи ПЛК

Устройства

Отчеты
- Создание нового отчета
- Редактирование макета отчета
  - Окно «Инспектор объектов»
  - Окно «Инспектор данных»
  - Окно «Инспектор скриптов»
  - Окно «Структура отчета»
- Элементы отчета
  - Текст
  - Диаграмма
  - Изображение
  - Фигура
  - Горизонтальная компоновка (Layout)
- Разделы отчета (бэнды)
- Функции
  - Агрегатные функции

AVG - вычисление среднего значения.

SUM - вычисление суммы значений.

MIN

MAX

COUNT

Числовые функции

numberFormat - форматирование числа

1 Узлы проекта

Узел - элементарная сущность проекта, осуществляющая хранение информации или реализующая определенный функционал. Каждый узел обладает набором свойств. Все узлы, объединенные в иерархическую структуру, образуют проект.

1.1 Проект

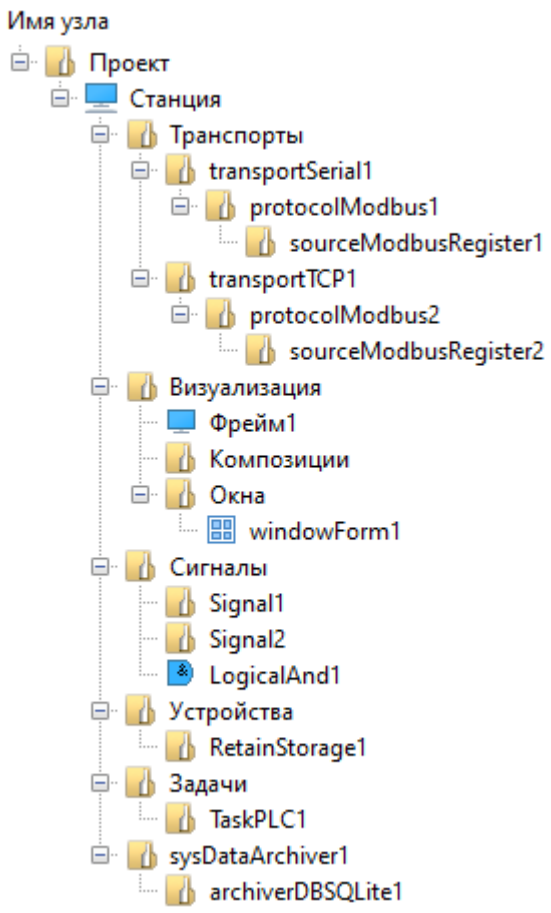


Рисунок 21 – Дерево проекта

Узел "Проект" ("Корневой узел", "Корень") предназначен для хранения общей информации о проекте и связывания других узлов проекта.

1.2 Станция

Станция – важнейший узел, управляющий работой собственных систем и выполняющий обмен данными с другими станциями. Физически станция представляет собой устройство, на котором исполняется проект (компьютер, панель оператора, МПР-60 или другое устройство).

Для использования в проекте доступны несколько типов станций. например:

- **Станция** - общий тип станции. Проекты с такой станцией могут выполняться на большинстве устройств - контроллеры, панели, персональные компьютеры. Чаще всего используется для проектов, предназначенных для выполнения на персональном компьютере.
- **Панель ПО-40** - специализированный тип для проектов на панелях оператора ПО-40. Станция такого типа по отношению к общему дополнительно предоставляет возможность управления оборудованием панели оператора (светодиоды индикации и др.), использования корзины субмодулей.
- **Панель ПО-50** - специализированный тип для проектов на панелях оператора ПО-50. Станция такого типа по отношению к общему дополнительно предоставляет возможность управления оборудованием панели оператора (светодиоды индикации и др.).
- **Контроллер МПР-60** - специализированный тип для проектов на программируемых реле МПР-60. Станция такого типа по отношению к общему дополнительно предоставляет возможность управления оборудованием панели оператора (светодиоды индикации и др.), использования корзины субмодулей.

При построении больших проектов станций в проекте может быть несколько, и они могут выполнять разные задачи. Например, станция-сервер выполняет сбор и регистрацию данных, клиентские станции выполняют визуализацию данных на рабочих местах и т. д.

Станции подчиняются узлы типа "Система" - Транспортная система, Система архивирования, Система визуализации и т. д.

1.3 Транспортная система

Транспортная система управляет передачей данных по различным каналам связи. Система управляет работой "Транспортов" - узлов, выполняющих обмен данными с устройствами по определенному протоколу.

С одной стороны, транспортная система обращается непосредственно к сигналам и каналам устройства, а с другой стороны, представляет их в виде неких универсальных единиц – узлов-источников. Каждый источник однозначно соответствует определенному сигналу или параметру устройства. При изменении сигнала меняется и значение источника. При записи данных в источник производится подача команды устройству на установку определенного значения конкретного параметра.

Транспортная система допускает использование нескольких транспортов, что дает возможность одновременного обмена данными с большим количеством устройств по

разным протоколам.

1.3.1 Транспорт

Транспорт предназначен для обмена данными с устройствами по определенному протоколу. Один транспорт, в зависимости от типа, позволяет обмениваться данными с одним или несколькими устройствами.

Доступны для использования следующие виды транспортов:

- Транспорт Modbus-TCP;
- Транспорт Modbus-RTU;
- Транспорт SNMP;
- Транспорт OPC UA;
- Транспорт OPC DA (только для Windows);
- Транспорт MQTT;
- Последовательный транспорт (устаревший);
- TCP/IP транспорт (устаревший).

Подробное описание транспортов приведено в отдельном документе "Транспорты AgavaSCADA/AgavaPLC".

1.4 Система визуализации

Система визуализации недоступна в проектах для контроллеров МПР

Задачей этой системы является отображение данных в виде набора мнемонических схем (экранов, мнемосхем), а также предоставление пользователю возможности управления.

Каждый экран позволяет в схематичном, понятном пользователю виде представить участок технологической схемы с отображением необходимых параметров технологического процесса.

Изображение на экране строится с помощью окон, на которых размещаются объекты визуализации – виджеты. Виджетами могут быть текстовые метки, изображения, в том числе анимированные, элементы управления, графики и т. д. Виджеты можно объединять в слои для удобного управления отображением виджетов.

Виджеты могут быть статическими и динамическими. Динамические виджеты привязаны к одному или нескольким сигналам и их отображение зависит от значений этих сигналов. Статические виджеты не имеют привязки к сигналам и их изображение всегда постоянно.

Для перехода по экранам, изменения значений параметров и иного управления служат специальные виджеты: кнопки управления, поля ввода и другие.

Для повышения быстродействия и увеличения комфортности работы с системой система визуализации хранит в памяти кэш из нескольких последних открытых экранов. Когда при последовательном открытии экранов этот кэш заполняется, наиболее старые экраны удаляются из него. Это связано с тем, что, как правило, в работе используются несколько одних и тех же экранов, которые отображают сводную информацию о техпроцессе и оператор в штатном режиме переключается только между ними. Сохраняя их в памяти, система ускоряет переключение между ними, поскольку при этом не требуется каждый раз создавать экран, все находящиеся на нем объекты и запускать процессы обновления параметров.

Когда экран переходит в кэш, обновление параметров на нем прекращается для снижения нагрузки на центральный процессор. Если все же необходимо, чтобы экран всегда производил обновление данных (по разным причинам), даже находясь в кэше, необходимо в его настройках включить опцию «Постоянно обновляемый».

Подчиненные типы узлов:

- фрейм;
- окно;
- композиция.

1.4.1 Фрейм

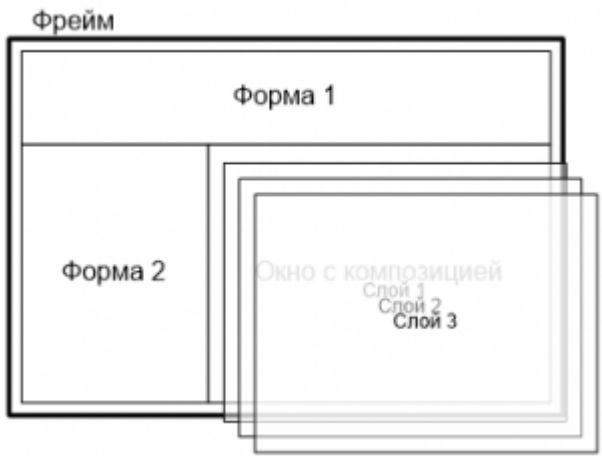


Рисунок 23 – Пример структуры сложного фрейма, состоящего из двух форм и одного окна с многослойной композицией

Фрейм – узел, предназначенный для группировки окон, отображаемых на определенном дисплее. Если дисплеев несколько, то для каждого из них можно использовать отдельный фрейм. Окон во фрейме может быть несколько, однако они не могут накладываться друг на друга.

1.4.2 Окно

Окно – узел, осуществляющий отображение видеокадра (мнемонической схемы).

Существуют два вида окон:

1. «Форма», предназначенная для вывода простого набора данных без необходимости

управления видимостью или внешним видом его элементов. Для вывода информации в такое окно кроме создания окна и добавления к нему виджетов ничего более не требуется.

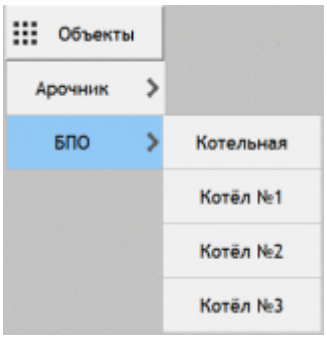


Рисунок 24 — Внешний вид меню в среде исполнения

2. «Окно с композицией», предназначенное для вывода большого количества информации и позволяющее организовать группировку элементов по слоям, управлять внешним видом элементов, видимостью слоев и отдельных элементов, а также дающее возможность регулирования работы пользователя с элементами. Для вывода информации в такое окно нужно к окну привязать композицию, содержащую минимум один слой, в который уже добавляются виджеты.

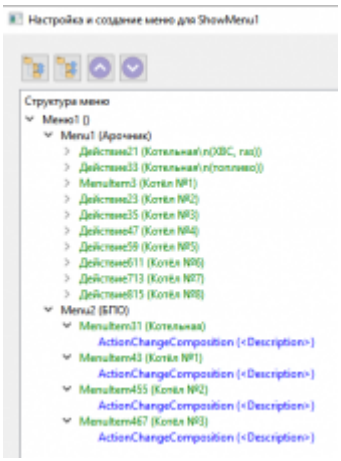


Рисунок 25 — Настройка меню в среде разработки

1.4.3 Композиция

Композиция – узел, представляющий составленную при разработке проекта мнемоническую схему в виде набора элементов визуализации (виджетов), сгруппированных в один или несколько слоев.

Композиция дает возможность организовать виджеты в слои, видимостью которых можно управлять во время работы проекта.

1.4.4 Слой

Слой – группа элементов визуализации, объединенных каким-то общим признаком.

1.4.5 Виджет

Виджет – элемент визуализации, предназначенный для отображения информации в заданном виде.

1.4.6 Меню

Меню – элемент визуализации, сочетающий в себе несколько виджетов «Кнопка». Используется для более удобного управления проектом (см. Рисунок 24).

Для добавления меню, необходимо в среде разработки назначить действие «Отображение меню» для кнопки (или любого другого виджета). После добавления действия необходимо выделить его и нажать на кнопку настроек на панели вверху окна действий. Открывшееся окно (см. Рисунок 25) содержит дерево со структурой меню, в который можно добавлять элементы двух типов:

- меню – элемент, который позволяет группировать элементы меню, выводя их в виде раскрывающегося списка. На рисунках 24 и 25 этим элементом являются кнопки «Арочник» и «БПО».
- элемент меню – элемент, который предоставляет функционал, аналогичный виджету «Кнопка». Для добавления действия, выполняющегося при нажатии на элемент меню в среде исполнения, необходимо нажать на него ПКМ в дереве структуры меню и выбрать "Создать действие". На рисунках 24 и 25 этим элементом являются кнопки «Котельная», «Котёл №1», «Котёл №2» и «Котёл №3».

1.4.7 Создание многомониторных систем визуализации

AgavaSCADA предоставляет возможность реализации систем визуализации в составе проектов, использующих для отображения информации более одного дисплея.

Для вывода информации в окнах на нескольких дисплеях в проект необходимо внести следующие корректировки относительно проекта, использующего только один дисплей:

1. Добавить дополнительные фреймы, чтобы их количество в проекте соответствовало количеству дисплеев. Каждому фрейму назначить порядковый номер, начиная с 1. Фреймы связываются с дисплеями по номеру.
2. Добавить дополнительное окно (или окна, если в исходном фрейме их несколько или дополнительных фреймов больше одного).
3. Добавить в дополнительные фреймы созданные на этапе 2 окна, чтобы повторить структуру основоного фрейма.

Базовая настройка на этом закончена. После запуска такого проекта на двух или более дисплеях должны отображаться созданные окна.

Если в проекте используется навигация по окнам, то необходимо в каждом окне, включенном в фрейм, скорректировать действия, размещаемые на кнопках, меню или

друигх виджетах таким образом, чтобы действия открывали окна или загружали композиции в нужном окне.

1.5 Виджеты

1.5.1 Виджет «Текстовая метка»

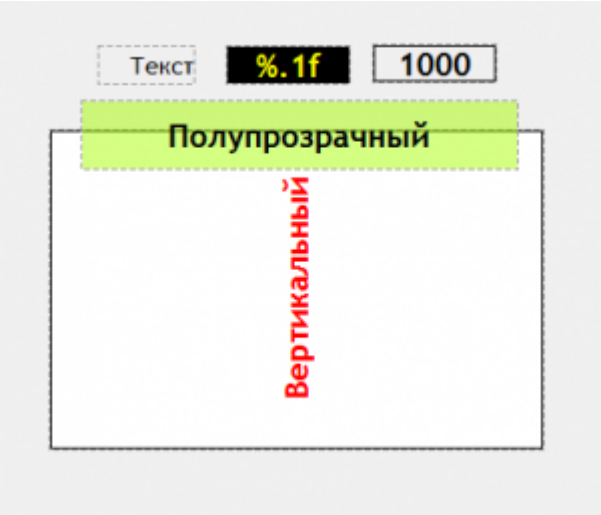


Рисунок 26 – Примеры оформления виджетов типа "Текстовая метка"

Виджет «Текстовая метка» – наиболее часто применяемый виджет, с помощью которого можно реализовать разные способы отображения текстовой информации.

Виджет «Текстовая метка» может использоваться для отображения статичного текста, значений параметров (одного или нескольких) и их комбинаций.

При необходимости отображения значений параметров они указываются в свойстве «Источник». Для задания определенного формата отображения используются спецификаторы формата в стиле языка «Си»:

- %f – значение в виде числа с плавающей точкой;
- %s – строка;
- %i – целое число со знаком;
- %u – целое число без знака;
- %d – целое число со знаком в десятичной системе исчисления.

Для разных типов свойств необходимо использовать строго определенные для них спецификаторы, с другими спецификаторами значения выводиться не будут.

Для отображения знака «%» необходимо указать его дважды: «%%».

1.5.1.1 Отображение нескольких значений

Одна текстовая метка может использоваться для отображения нескольких значений. Для этого все значения, которые нужно отобразить, указываются в свойстве «Источник», а в свойстве «Текст» указываются последовательно несколько

спецификаторов формата соответственно порядку перечисления значений в свойстве «Источник».

Пример: текстовая метка, отображающая текущую позицию в очереди:

- Текст: `Позиция %i из %i`
- Источник: два узла из проекта.

1.5.1.2 Модификаторы спецификаторов формата

1.5.1.2.1 Модификатор минимальной ширины поля

Целое число, расположенное между знаком % и кодом формата, играет роль модификатора минимальной ширины поля. Если указан модификатор минимальной ширины поля, то, чтобы ширина поля вывода была не меньше указанной минимальной величины, вывод при необходимости будет дополнен пробелами. Если же выводятся строки или числа, которые длиннее указанного минимума, то они все равно будут отображаться полностью. По умолчанию для дополнения используются пробелы. А если для этого надо использовать нули, то перед модификатором ширины поля следует поместить 0. Например, %05i означает, что любое число, количество цифр которого меньше пяти, будет дополнено таким количеством нулей, чтобы состоять из пяти цифр.

Примеры

Значение	Спецификатор формата	Выводимое значение
10.123	%i	10
10.123	%d	10
10.123	%05i	00010
10.123	%f	10.123000
10.123	%.1f	10.1
10.123	%10f	<пробел>10.123000
10.123	%012f	00010.123000

1.5.1.2.2 Модификатор точности

Модификатор точности следует за модификатором минимальной ширины поля (если таковой имеется). Он состоит из точки и расположенного за ней целого числа. Значение этого модификатора зависит от типа данных, к которым его применяют.

Когда модификатор точности применяется к данным с плавающей точкой, для преобразования которых используются спецификаторы преобразования %f, %e или %E, то он определяет количество выводимых десятичных разрядов. Например, %10.4f означает, что ширина поля вывода будет не менее 10 символов, причем для десятичных разрядов будет отведено четыре позиции.

Если модификатор точности применяется к %g или %G, то он определяет количество значащих цифр.

Примененный к строкам, модификатор точности определяет максимальную длину поля. Например, %5.7s означает, что длина выводимой строки будет составлять минимум пять и максимум семь символов. Если строка окажется длиннее, чем

максимальная длина поля, то конечные символы выводиться не будут.

Если модификатор точности применяется к целым типам, то он определяет минимальное количество цифр, которые будут выведены для каждого из чисел. Чтобы получилось требуемое количество цифр, добавляется некоторое количество ведущих нулей.

Примеры:

Значение	Спецификатор формата	Выводимое значение
10.123	%f	10.123000
10.123	%.1f	10.1
10.123	%10.4f	<пробел><пробел><пробел>10.1230
10.123	%010.3f	000010.123
Это простая проверка	%10.15s	Это простая про

1.5.2 Виджет «Условная метка»

ссылка=Файл:AgavaPLC–Список_сопоставлений_виджета_%22Условная_метка%22.png

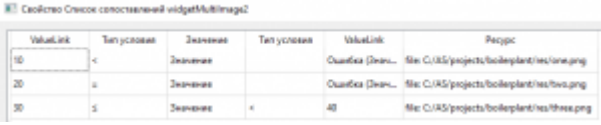
Виджет «Условная метка» позволяет изменять выводимую строку в соответствии с изменениями привязанной к ней переменной.

Задать соответствие текста необходимому условию можно в свойстве «Список сопоставлений».

1.5.3 Виджет «Изображение»

Виджет «Изображение» предназначен для отображения на окнах статичных изображений форматов png, gif, jpg, bmp или анимированных изображений в формате gif. Отображаемая картинка выбирается в разделе «Оформление» - «Изображение». Она должна находиться в папке проекта.

1.5.4 Виджет «Условное изображение»



ссылка=Файл:AgavaPLC–Список_сопоставлений_виджета_%22Условное_изображение%22.png

Виджет «Условное изображение» позволяет изменять отображаемое изображение в соответствии с изменениями привязанной к нему переменной.

Задать соответствие изображения необходимому условию можно в свойстве «Список сопоставлений».

1.5.5 Виджет «Движущееся изображение»

Виджет предназначен для отображения на окнах изображений, перемещающихся в пределах области виджета. В основном используются на экране спящего режима.

Отображаемая картинка выбирается в разделе «Оформление» - «Изображение». Она должна находиться в папке проекта.

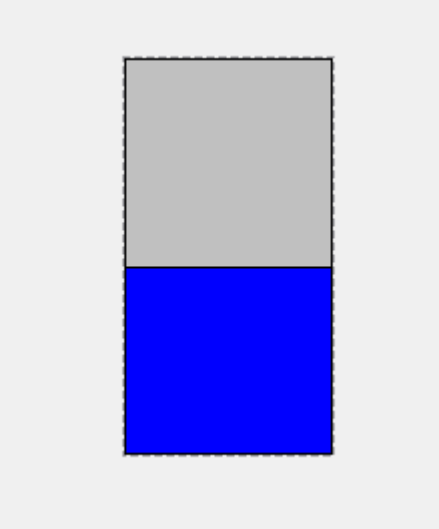
1.5.6 Виджет «Кнопка»

Виджет «Кнопка» предназначен для реализации функций управления:

- ввод данных;
- передача управляющих воздействий;
- смена режима работы АРМ или иного устройства, исполняющего проект.

Выполнение всех функций виджета обеспечивается с помощью узлов типа «Действие», добавляемых в соответствующее свойство (см. раздел #Действия).

1.5.7 Виджет «Гистограмма»



ссылка=Файл:AgavaPLC–Виджет_%22Гистограмма%22.png

Виджет предназначен для отображения значений сигналов в виде столбчатой диаграммы.

Виджету необходимо задать источник (регистр/сигнал, с которого он будет получать данные), а также минимальное и максимальное значения.

1.5.8 Виджет «Исторический график»

Виджет «Исторический график» предназначен для отображения графиков параметров на двухкоординатной области изображения с привязкой к осям времени и значений.

Виджет может работать с данными, архивируемыми в БД, и напрямую с сигналами,

значения которых вычисляются в ходе работы.

При работе с архивируемыми данными интервал обновления графика зависит от режима выдачи значения у базы данных: при выборе сохраняемых в БД он будет соответствовать интервалу архивирования, а при сохраненных в БД - интервалу сохранения БД на диск. Независимо от режима выдачи, график не может обновляться меньше чем за 300 мс.

При работе напрямую с сигналами график обновляется при изменении значения сигнала, но также не чаще чем раз в 300 мс.

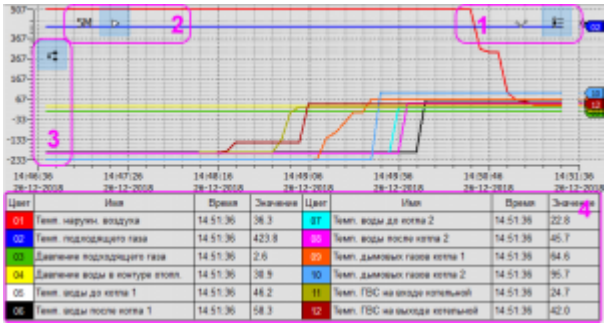


Рисунок 30 – Виджет «Исторический график». Области кнопок-пиктограмм. В нижней части отображена легенда

Виджет «Исторический график» состоит из нескольких областей:

- область осей X, Y;
- область легенды, содержащей список кривых и их параметры (цвет кривой, наименование кривой, текущее значение, время последнего зарегистрированного значения);
- область вывода данных.

На виджете имеются несколько областей, содержащих кнопки-пиктограммы управления внешним видом графика (см. Рисунок 28):

1. область пиктограмм графика;
2. область пиктограмм оси времени (X);
3. область пиктограмм оси значений (Y);
4. область легенды.

В области пиктограмм графика слева направо расположены следующие кнопки-пиктограммы:

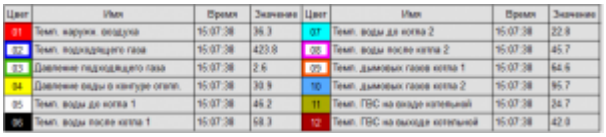
- кнопка переключения режима отображения точек на всех кривых;
- кнопка переключения видимости легенды;
- кнопка включения режима визирной линии.

В области пиктограмм оси времени (X) слева направо расположены следующие кнопки-пиктограммы:

- кнопка открытия окна выбора интервала времени для оси X, текст кнопки отображает текущий выбранный интервал, например: 5М – 5 минут, 6Ч – 6 часов, 30М – 30 минут;
- кнопка-индикатор включения следящего режима.

В области пиктограмм оси значений (Y) расположена кнопка-пиктограмма «А» – кнопка переключения режима автоматического масштабирования значений.

1.5.8.1 Легенда



Цвет	Имя	Время	Значение	Цвет	Имя	Время	Значение
01	Темп. наружн. воздуха	15:07:38	36.3	07	Темп. воды до котла 2	15:07:38	22.8
02	Темп. подходящего газа	15:07:38	423.8	08	Темп. воды после котла 2	15:07:38	45.7
03	Давление подходящего газа	15:07:38	2.6	09	Темп. дымовых газов котла 1	15:07:38	64.6
04	Давление воды в контуре отопл.	15:07:38	39.9	10	Темп. дымовых газов котла 2	15:07:38	95.7
05	Темп. воды до котла 1	15:07:38	46.2	11	Темп. ПВС на входе котельной	15:07:38	24.7
06	Темп. воды после котла 1	15:07:38	68.3	12	Темп. ПВС на выходе котельной	15:07:38	42.0

Рисунок 31 – Легенда с четырьмя выключенными параметрами

Легенда – специальная область в нижней части экрана архива, предназначенная для отображения параметров (наименование, текущее значение и т. д.) выводимых графиков в виде таблицы. Для ознакомления с примерным видом легенды см. Рисунок 31. Информация о каждом сигнале в легенде выводится в несколько колонок:

- «Цвет» – индицирует цвет, которым выводится соответствующая кривая на графике, колонка «Цвет» позволяет нажатием переключать видимость соответствующей кривой на графике (см. Рисунок 29). Также в колонке "Цвет" отображается номер пера, которым выводится кривая параметра на графике;
- «Имя» («Наименование») – содержит наименование параметра;
- «Время» – содержит время последнего зарегистрированного значения параметра;
- «Значение» – содержит последнее зарегистрированное значение параметра.

1.5.8.2 Режимы просмотра графиков

Пользователю доступно два режима просмотра графиков:

- следящий режим;
- режим просмотра истории.

В следящем режиме на графике всегда отображаются актуальные значения параметров, тренд периодически дочитывает поступившие данные из базы и обновляет графики.

В режиме просмотра истории на графике отображаются исторические значения параметров. Просмотр истории технологического процесса осуществляется с помощью кнопок перемещения по осям и кнопок изменения масштаба.

Режим просмотра истории включается автоматически при прокрутке графика назад по времени.

1.5.8.3 Визирная линия

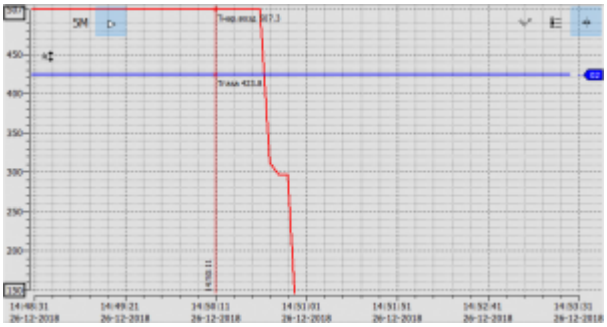


Рисунок 32 – График в режиме просмотра истории с установленной визирной линией

Для просмотра значений параметров в нужный момент времени служит визирная линия, отображаемая на графике как вертикальная линия.

При щелчке левой кнопкой мыши (ЛКМ) в области графиков визирная линия будет отображена в выбранном месте графика.

В точке пересечения визирной линии и кривых параметров отображаются значения этих параметров. Также в нижней части области графиков, возле оси времени, визирная линия отображает время.

1.5.9 Виджет «График кусочно-линейной функции»

Виджет предназначен для отображения графика зависимости двух параметров, расположенном на осях X и Y.

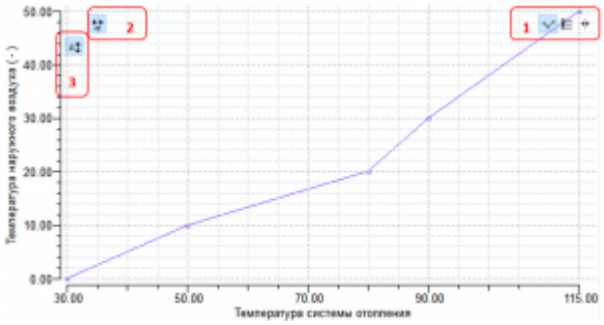


Рисунок 33 – Виджет «График кусочно-линейной функции». Области кнопок-пиктограмм.

Виджет состоит из нескольких областей:

- область осей X, Y;
- область легенды, содержащей список кривых и их параметры (цвет кривой, наименование кривой, текущее значение, время последнего зарегистрированного значения);
- область вывода графика.

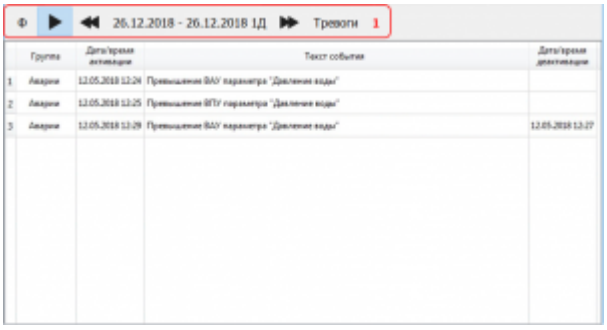
На области вывода графика виджета имеются несколько областей, содержащих кнопки-пиктограммы управления внешним видом графика (см. Рисунок 33):

1. Область пиктограмм графика.
2. Область пиктограмм оси первого параметра (X).
3. Область пиктограмм оси второго параметра (Y).

В области пиктограмм графика расположены следующие кнопки-пиктограммы:

- кнопка переключения режима отображения точек на всех кривых;
- кнопка переключения видимости легенды;
- кнопка включения режима визирной линии.

В области пиктограмм оси первого (X) и второго (Y) параметров расположены кнопки-пиктограммы «А» – кнопка переключения режима автоматического масштабирования значений.



ссылка=Файл:AgavaPLC–Виджет_%22Список_событий%22.png

1.5.10 Виджет «Список событий»

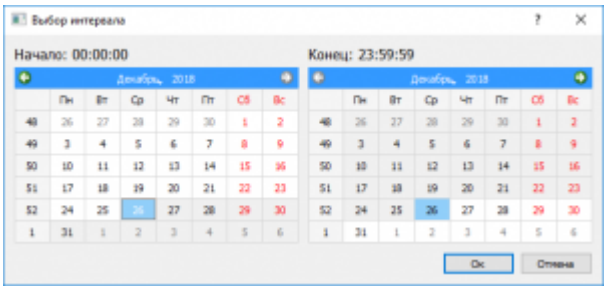


Рисунок 35 – Окно выбора интервала

Виджет предназначен для отображения регистрируемых событий в работе.

В верхней части виджета расположена панель управления (1).

Панель управления содержит органы управления журналом:

- кнопка отображения фильтра событий;
- кнопка включения следящего режима журнала;
- кнопка изменения интервала на один день назад;
- кнопка-индикатор выбора интервала, за который отображаются события;
- кнопка изменения интервала на один день вперед;

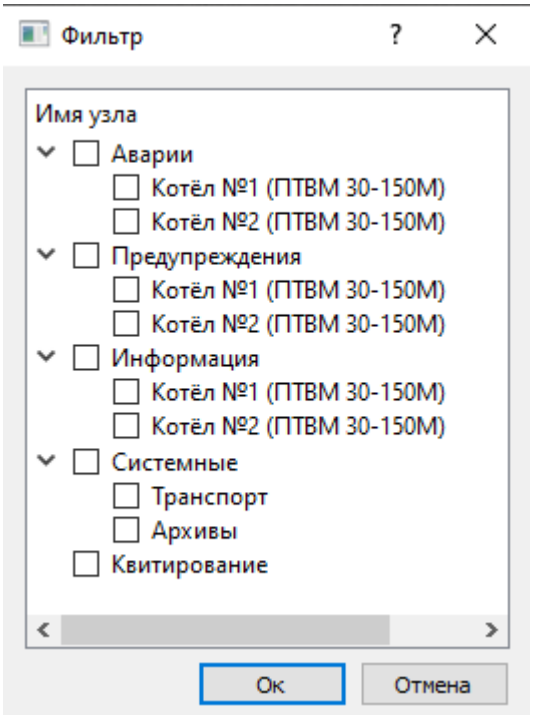
- кнопка вызова окна со списком активных тревог.

Ниже панели управления, в основном поле экрана, расположена таблица, содержащая зарегистрированные события.

Заголовок таблицы позволяет изменять порядок столбцов и включать сортировку по столбцам. Например, возможно включение сортировки событий сначала по времени, затем по группе. Заголовок таблицы имеет возможность отключения / включения отображения любого столбца.

При большом количестве записей в правой части таблицы появляется полоса прокрутки, позволяющая пролистывать список событий.

Нажатие на кнопку-индикатор выбора интервала вызывает окно выбора интервала (см. Рисунок 35):



ссылка=Файл:AgavaPLC-Окно_%22Фильтр%22.png

После выбора нужного интервала для сохранения выбора необходимо нажать кнопку «ОК». После сохранения будут отображены зарегистрированные события за выбранный интервал.

Нажатие на кнопку отображения окна фильтра вызывает соответствующее окно. Окно фильтра позволяет изменить содержимое основного окна журнала событий путем выбора групп и подгрупп событий, перечисленных в окне фильтра (см. Рисунок 36).

После выбора нужных групп и подгрупп для сохранения выбора необходимо нажать кнопку «ОК». После сохранения будут отображены зарегистрированные события, принадлежащих выбранным группам.

1.5.11 Виджет «Ввод значения»

Виджет «Ввод значения» предназначен для отображения значения привязанного сигнала и его редактирования.

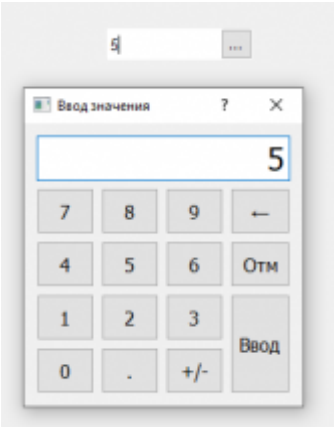


Рисунок 37 – Виджет «Ввод значения» с отображением экранной цифровой клавиатуры

После выбора узла для чтения/записи и соответствующего типа значения необходимо выбрать нужный тип ввода в разделе «Оформление».

Доступны следующие типы ввода:

- кнопки изменения – при нажатии на виджет появляются кнопки увеличения и уменьшения на заданный шаг изменения;
- кнопка редактирования – при нажатии на виджет появляется кнопка, при клике на которую открывается экранная цифровая клавиатура;

Виджет имеет возможность проверки вводимых значений, при активации которой запись будет осуществляться только тогда, когда записываемое значение будет находиться в диапазоне между заданными минимальным и максимальными значениями.

1.5.12 Виджет «Выбор значения из списка»

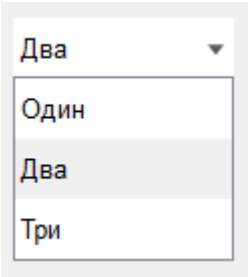


Рисунок 38 – Виджет «Выбор значения из списка»

Виджет «Выбор значения из списка» предназначен для отображения значения

привязанного сигнала и его изменения в удобном для пользователя текстовом виде (см. Рисунок 37).

С помощью свойства «Список сопоставлений» задается соответствие между значением сигнала и выводимой в виджете строкой.

При нажатии на кнопку в правой части виджета отображается список всех сопоставлений. При выборе нужного элемента в списке в установленный узел производится запись нового значения.

1.5.13 Виджет «Флаг»

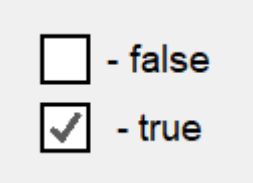


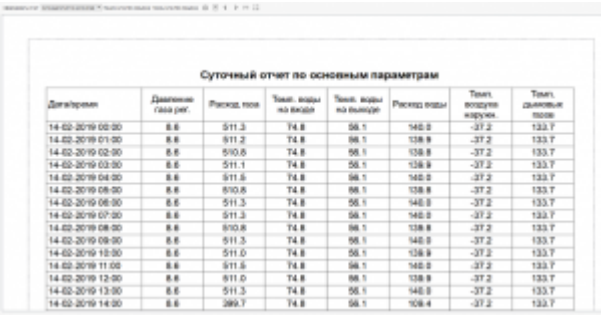
Рисунок 39 – Варианты внешнего вида виджета «Флаг»

Виджет «Флаг» предназначен для отображения значения и изменения булевой переменной (true/false). Состояние true отображается в виде галочки (см. Рисунок 38).

При нажатии на виджет значение переменной изменяется на противоположное.

В случае, если флаг необходимо использовать только для отображения информации, поставьте галочку в свойстве «Отключен»

1.5.14 Виджет «Отчет»



Суточный отчет по основным параметрам

Дата/время	Давление газа рег.	Расход газа	Темп. воды на входе	Темп. воды на выходе	Расход воды	Темп. воздуха наружн.	Темп. дымовых газов
14-02-2019 00:00	8.6	511.3	74.8	56.1	140.0	-37.2	133.7
14-02-2019 01:00	8.6	511.2	74.8	56.1	138.9	-37.2	133.7
14-02-2019 02:00	8.6	510.6	74.8	56.1	138.8	-37.2	133.7
14-02-2019 03:00	8.6	511.1	74.8	56.1	138.9	-37.2	133.7
14-02-2019 04:00	8.6	511.5	74.8	56.1	140.0	-37.2	133.7
14-02-2019 05:00	8.6	510.8	74.8	56.1	138.8	-37.2	133.7
14-02-2019 06:00	8.6	511.3	74.8	56.1	140.0	-37.2	133.7
14-02-2019 07:00	8.6	511.3	74.8	56.1	140.0	-37.2	133.7
14-02-2019 08:00	8.6	510.8	74.8	56.1	138.8	-37.2	133.7
14-02-2019 09:00	8.6	511.3	74.8	56.1	140.0	-37.2	133.7
14-02-2019 10:00	8.6	511.0	74.8	56.1	138.9	-37.2	133.7
14-02-2019 11:00	8.6	511.5	74.8	56.1	140.0	-37.2	133.7
14-02-2019 12:00	8.6	511.0	74.8	56.1	138.9	-37.2	133.7
14-02-2019 13:00	8.6	511.3	74.8	56.1	140.0	-37.2	133.7
14-02-2019 14:00	8.6	399.7	74.8	56.1	109.4	-37.2	133.7

Рисунок 40 – Виджет «Отчет» в режиме отображения сформированного отчета

Виджет «Отчет» предназначен для формирования отчетов по заранее спроектированному макету.

Подробный порядок создания и редактирования отчетов приведен в разделе #Отчеты.

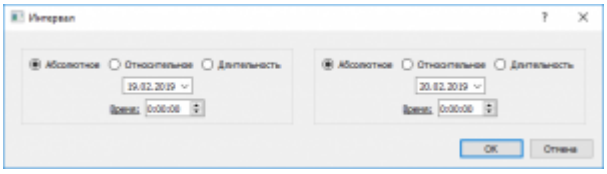


Рисунок 41 – Окно выбора интервала отчета

В верхней части виджета размещена панель управления, содержащая кнопки вызова команд:

- печать сформированного отчета на установленном в системе принтере;
- сохранение сформированного отчета в файле формата PDF;
- переход к предыдущей странице отчета;
- переход к следующей странице отчета;
- масштабирование экрана по ширине листа отчета;
- масштабирование экрана по высоте листа отчета.

При формировании отчета для изменения интервала времени, за который будет формироваться отчет, необходимо на панели управления щелкнуть кнопку выбора интервала (на ней отображается текущий выбранный интервал, например, «Начало суток без смещения – Конец суток без смещения»). Далее появится окно «Интервал», в котором можно выбрать начало и конец интервала (см. Рисунок 41).

В качестве начала и конца интервала можно выбрать как конкретную дату по календарю, так относительную, например, «Начало дня», "Конец недели". После выбора желаемого интервала необходимо нажать кнопку «OK».

Выбранный интервал отобразится на панели управления отчетом. Для формирования отчета за выбранный интервал необходимо нажать кнопку «Сформировать отчет». Через некоторое время, после загрузки данных из БД и их обработки, экран обновится и сформированный отчет за выбранный интервал будет отображен.

1.6 Действия

В свойствах виджетов «Обработка событий» создаются узлы типа «Действие», которые активируются при возникновении соответствующих событий:

- OnLeftButtonDown – нажатие левой кнопки мыши при наличии фокуса ввода на виджете.
- OnLeftButtonUp – отпускание левой кнопки мыши при наличии фокуса ввода на виджете.
- OnFocusOut – потеря фокуса ввода.
- OnButtonPress – нажатие левой кнопки мыши или кнопок "Пробел", "Enter" на виджете типа «Кнопка».
- OnButtonRelease – отпускание левой кнопки мыши или кнопок "Пробел", "Enter" на виджете типа «Кнопка».

1.6.1 Действие «Воспроизведение звукового файла»

Действие «Воспроизведение звукового файла» предназначено для воспроизведения указанного звукового файла через подключенный громкоговоритель или встроенную аудиосистему. Выбранный файл должен располагаться в папке проекта и быть загруженным в контроллер.

1.6.2 Действие «Смена композиции»

Действие «Смена композиции» предназначено для смены текущей отображаемой композиции в указанном окне на другую.

1.6.3 Действие «Отображение окна»

Действие «Отображение окна» предназначено для создания и отображения окна. Окно можно отобразить по центру дисплея либо по заданным координатам.

1.6.4 Действие «Закрытие окна»

Действие «Закрытие окна» предназначено для закрытия окна.

1.6.5 Действие «Запись значения в узел»

Действие «Запись значения в узел» предназначено для записи определенного значения в указанный узел.

1.6.6 Действие «Создание события»

Действие «Создание события» предназначено для создания нового события. События используются для вывода информационных сообщений или активации тревог, которые передаются в подсистемы тревог и журналирования.

1.6.7 Действие «Команда узлу»

Действие «Команда узлу» предназначено для передачи определенной команды («Запуск», «Останов» и др.) указанному узлу.

1.6.8 Действие «Запуск процесса»

Действие «Запуск процесса» предназначено для запуска указанного приложения ОС.

1.6.9 Действие «Авторизация пользователя»

Действие «Авторизация пользователя» предназначено для взаимодействия с системой проверки прав пользователей. При выполнении действия производится регистрация в системе указанного пользователя как текущего.

1.6.10 Действие «Переключение видимости окна»

Действие «Переключение видимости окна» сочетает в себе действия по открытию и закрытию окна. Если окно было изначально закрыто, оно будет отображено, если окно было открыто – оно будет закрыто.

1.6.11 Действие «Выполнение скрипта»

Действие «Выполнение скрипта» предназначено для запуска исполнения скрипта C++. Скрипт создается с помощью операции «Скрипт C++», описание которой приведено в разделе #Операция «Скрипт C++».

1.6.12 Действие «Отображение меню»

Действие «Отображение меню» предназначено для отображения всплывающего или выпадающего меню по команде пользователя.

1.7 Система тревог

Система тревог предназначена для оповещения пользователя об обнаруженных нарушениях.

Система содержит два типа узлов:

- Группа событий.
- Контроллер событий.

1.7.1 Группа событий

Группа событий – узел, предназначенный для взаимодействия с виджетом «Список событий». Группы событий позволяют группировать события по какому-то признаку, например по уровню тревоги (авария, предупреждение, квитирование и т.д.), технологическому участку и т.д.

Для каждой группы можно выбрать свой цвет шрифта и фона, которые будут использоваться для вывода событий, принадлежащих этой группе в виджете "Список событий".

В свойстве «Описание» группы событий указывается текст, который будет отображаться в фильтре списка событий.

1.7.2 Контроллер событий

Контроллер событий – узел, реализующий логику проверки соответствия выбранных сигналов контролируемым условиям. Контроллеры событий выполняют всю работу по проверке значений контролируемых параметров и выполнению соответствующих действий.

Контроллер событий проверяет значения заданных узлов на соответствие заданным

условиям и при их совпадении выполняет заданные действия.

В качестве действий можно указать не только действие "Создание события", но и другие (см. раздел #Действия).

Система тревог

1.7.3 Порядок настройки системы

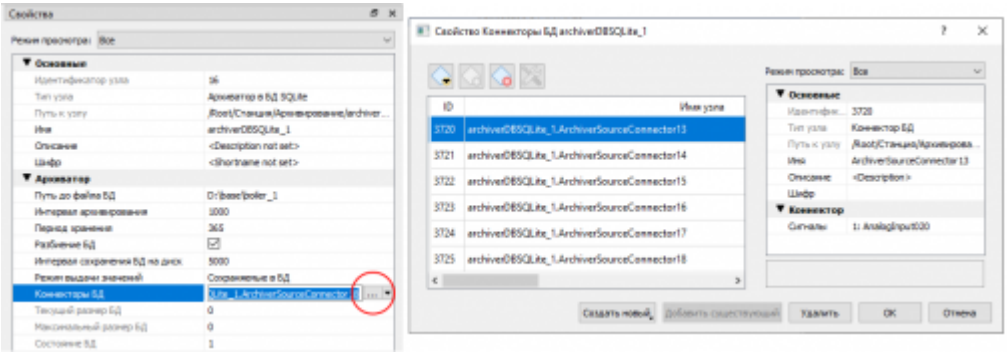
Порядок настройки системы тревог следующий:

1. Добавить в проект систему тревог. Для этого нажать ПКМ по узлу «Станция» в дереве проекта, затем выбрать «Добавить узел» - «Система» - «Система тревог».
2. Добавить узел «События» в систему тревог. Для этого нажать ПКМ по системе тревог в дереве проекта, затем выбрать «Добавить узел» - «Группа» - «События». При необходимости настроить цвет шрифта и фона, которые будут отображаться у события в окне тревог или журнале.
3. Добавить нужные группы событий в узел «События». Для этого нажать ПКМ по узлу «События» в дереве проекта, затем выбрать «Добавить узел» - «Группа событий».
4. Добавить контроллеры событий в систему тревог. Для этого нажать ПКМ по системе тревог в дереве проекта, затем выбрать «Добавить узел» - «Контроллер событий».
5. Добавить в свойство «Сигналы» контроллеров событий узлы, состояние которых необходимо контролировать.
6. Задать для этих узлов контролируемые условия в соответствующем свойстве контроллера событий.
7. Настроить действия, которые будут выполняться при выполнении контролируемого условия. Для вывода аварий, предупреждений или информационных сообщений необходимо добавить действие «Создание события».
8. Настроить действие «Создание события». В его свойстве «Группа событий» указать группу из созданных в пункте 2. Ввести тексты сообщения активации и деактивации (при необходимости), указать уровень события. Теперь, при срабатывании контролируемого условия, будет появляться окно тревог с сообщением активации.

1.8 Система архивации

Система архивирования предназначена для управления процессом сохранения значений параметров в базы данных. Параметры, сохраненные в базу данных, могут использоваться для построения графиков параметров и генерации отчетов.

Архивация параметров производится с помощью архиваторов.



Настройка архиватора

1.8.1 Порядок настройки системы

Для сохранения истории изменения значений параметров в БД необходимо выполнить следующие действия:

1. Добавить в проект систему архивации. Для этого нажать ПКМ по узлу «Станция» в дереве проекта, затем выбрать «Добавить узел» - «Система» - «Система архивации».
2. Добавить в систему архивации архиватор. Для этого нажать ПКМ по узлу «Система архивации» в дереве проекта, затем выбрать «Добавить узел» - «Архиватор в БД SQLite».
3. Настроить параметры архиватора, такие как: путь до файла БД, интервал архивирования (как часто система архивации будет опрашивать значение параметра), интервал сохранения БД на диск (как часто эти значения будут записываться в постоянную память), разбиение БД (будет ли система архивации создавать новый файл БД каждые сутки), период хранения и режим выдачи значений для построения графика (сохраняемые значения будут обновляться с частотой интервала архивирования, а сохраненные - с частотой интервала сохранения).
4. Добавить параметры в коннекторы база данных. Для этого необходимо открыть свойство архиватора «Коннекторы БД», выбрать «Создать новый» - «Коннектор БД». После этого нужно нажать по добавленному коннектору, открыть его свойство «Сигналы» и добавить параметр, значение которого необходимо сохранять в БД.

1.8.2 Архиватор SQLite

Архиватор SQLite предназначен для сохранения значений параметров в БД формата SQLite.

Для извлечения и просмотра баз данных формата SQLite с устройств может использоваться программа «Архиватор».

1.8.2.1 Свойства узла "Архиватор SQLite"

Наименование свойства	Идентификатор	Тип	Доступ	Описание
Путь до файла БД	FilePath	STRING	Чтение / запись	Путь до файла БД. Абсолютный или относительный.
Интервал архивирования	Interval	INT	Чтение / запись	Интервал архивирования, мс.
Период хранения	Retention	INT	Чтение / запись	Период хранения, дней.
Разбиение БД	DBSplitting	BOOL	Чтение / запись	Разбиение БД на файлы. При включении база данных будет разделяться на отдельныйе файлы, содержащие данные за одни сутки.
Интервал сохранения БД на диск	StoringInterval	INT	Чтение / запись	Интервал сохранения БД на диск, мс.
Режим выдачи значений	ValueReturnMode	ENUM	Чтение / запись	Режим выдачи значений. Режим "Сохраняемые в БД" - данные выдаются сразу после получения архиватором, возможно до фактического сохранения в БД. Режим "Сохраненные в БД" - данные выдаются только после фактической записи в БД на диск.
Коннекторы БД	Signals	NODESARRAY	Чтение / запись	Список коннекторов, содержащих сигналы для архивирования.
Текущий размер БД	DBCurrentSize	INT	Только чтение	Текущий размер БД, Мб.
Максимальный размер БД	DBMaxSize	INT	Только чтение	Максимальный размер БД, Мб.
Состояние БД	DBStatus	INT	Только чтение	Состояние БД.

1.8.3 Архиватор PostgreSQL

Архиватор PostgreSQL предназначен для сохранения значений параметров в базы данных под управлением сервера PostgreSQL.

1.9 Система журналирования

Система ведения журналов событий предназначена для управления регистрацией и хранением событий. В базу данных записываются все системные события и события, генерируемые с помощью системы тревог, которые соответствуют выбранному уровню сообщений.

Для ведения журналов используются узлы типа "Логгер".

1.9.1 Порядок настройки системы

Порядок настройки системы журналирования следующий:

1. Добавить и настроить систему тревог для генерации событий.
2. Добавить в проект систему журналирования. Для этого нажать ПКМ по узлу «Станция» в дереве проекта, затем выбрать «Добавить узел» - «Система» - «Система журналирования».

3. Добавить в систему журналирования логгер. Для этого нажать ПКМ по узлу «Система архивации» в дереве проекта, затем нажать «Добавить узел» и выбрать нужный логгер.
4. Настроить параметры логгера (см. разделы ниже).

1.9.2 Логгер SQLite

Логгер SQLite предназначен для сохранения событий, возникающих в ходе исполнения прикладной программы в СУБД SQLite.

Логгер позволяет настраивать срок хранения событий, уровень сохраняемых событий и включать режим разбиения базы данных на отдельные файлы, содержащие данные только за одни сутки.

Информацию о событиях в базе данных, которую ведет логгер, можно просмотреть с помощью виджета "Список событий".

1.9.3 Логгер файловый

Логгер файловый предназначен для сохранения журналов событий в текстовых файлах на накопителе. Логгер имеет минимальное количество настроек, и обычно используется для сохранения журнала событий в случае необходимости отладки работы прикладной программы без среды разработки.

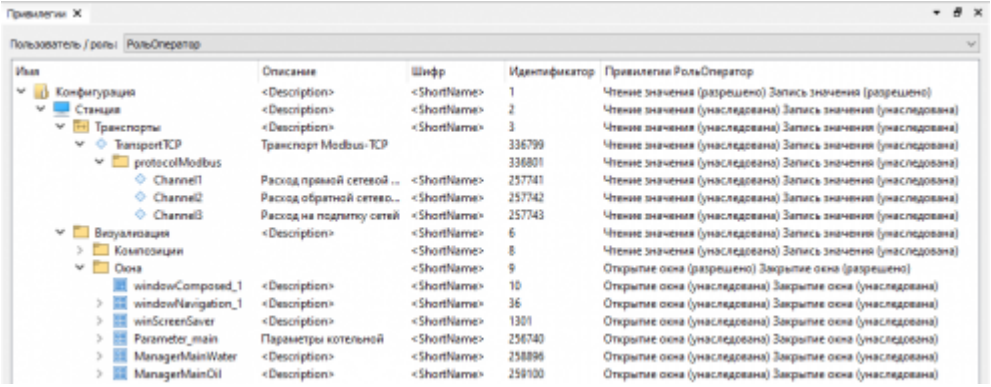
1.9.4 Логгер SMS

Логгер SMS предназначен для SMS информирования с помощью субмодуля GRPS или внешнего GPRS модема.

Принцип работы логгера следующий:

1. Логгер принимает возникающие события, фильтрует их согласно установленного уровня событий и выбранных групп.
2. События, прошедшие фильтрацию, форматируются согласно установленного шаблона.
3. Отформатированные сообщения отправляются через подключенный модем в виде SMS сообщения. Отправка сообщений производится на телефонные номера, записанные в телефонной книге на SIM-карте.

1.10 Система проверки прав пользователей



Окно привилегий

Система реализует механизм контроля доступа к узлам проекта и их функциях на основе ролей. Узлы при выполнении действий проверяют наличие установленных привилегий у авторизованного в данный момент пользователя.

1.10.1 Роли и пользователи

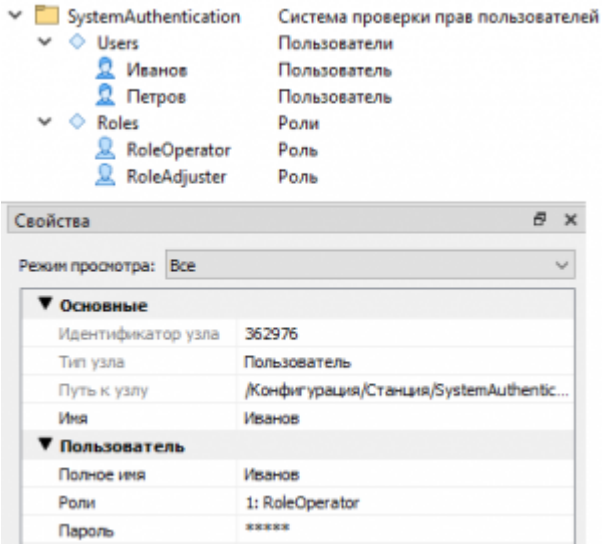
Система содержит в себе две подсистемы:

- Пользователи. Подсистема определяет пользователей, каждому из которых назначается логин/пароль и одна или несколько ролей.
- Роли. Роли хранят наборы доступных привилегий.

Отношения между пользователями, ролями и привилегиями можно описать следующим образом:

- Один пользователь может иметь несколько ролей.
- Одну роль могут иметь несколько пользователей.
- Одна роль может иметь несколько привилегий.
- Одна привилегия может принадлежать нескольким ролям.

1.10.2 Привилегии



Система проверки прав пользователей

Для узлов системы доступна установка следующих привилегий:

Все узлы:

- Чтение значения.
- Запись значения.

Окна и фреймы:

- Открытие окна.
- Закрытие окна.

При установке привилегий на какой-либо узел, все подчиненные узлы получают те же привилегии с флагом "унаследована".

Если привилегия не установлена, то считается что действие, регулируемое привилегией, запрещено.

Значение унаследованных привилегий можно переопределить с помощью явного задания привилегии.

1.10.3 Порядок настройки системы

Привилегии настраиваются в отдельном окне привилегий, доступном в меню "Вид".

Порядок настройки системы проверки прав в версии AgavaSCADA 1.2 следующий:

1. Добавить в проект систему проверки прав пользователей. Для этого нажать ПКМ по узлу «Станция» в дереве проекта, затем выбрать «Добавить узел» - «Система» - «Система проверки прав пользователей».
2. Добавить в систему проверки прав пользователей узел «Роль». Для этого нажать ПКМ по узлу «Система проверки прав пользователей» в дереве проекта, затем выбрать «Добавить узел» - «Роли».
3. Добавить в узел «Роли» необходимые роли. Для этого нажать ПКМ по узлу «Роли» в дереве проекта, затем выбрать «Добавить узел» - «Роль».
4. Для каждой роли настроить привилегии в окне привилегий. Для этого необходимо нажать «Вид» - «Открыть привилегии». В открывшемся окне в выпадающем списке «Пользователь / роль» выбрать нужную роль. Затем, нажимая ПКМ по узлам и выбирая «Добавить привилегию» настроить все необходимые ограничения для данной роли.
5. Добавить в систему проверки прав пользователей узел «Пользователи». Для этого нажать ПКМ по узлу «Система проверки прав пользователей» в дереве проекта, затем выбрать «Добавить узел» - «Пользователи».
6. Добавить в узел «Пользователи» необходимых пользователей. Для этого нажать ПКМ по узлу «Пользователи» в дереве проекта, затем выбрать «Добавить узел» - «Пользователи».
7. Для каждого пользователя указать имя и пароль, которые будут использоваться при авторизации, и выбрать одну или несколько ролей.

1.11 Сигналы

Регистры и другие узлы, добавленные в группу «Сигналы», являются внутренними для проекта и могут менять свои значения только с помощь скриптов или других узлов, с которыми они связаны.

Сигналы содержат в себе несколько подгрупп.

1.11.1 Сигнал

Сигнал - специальный тип узла, хранящий значение узла типа "источник". Сигналы могут использоваться для хранения значений нескольких источников, модифицировать их, обладать дополнительными полями и т. д.

Сигналы некоторых типов позволяют модифицировать исходное значение источников. Принцип обработки значения источника или сигнала с помощью операций:

Первичное значение поступает на вход операции, производится вычисление. Если есть еще операции, то вычисленное значение после первой операции передается на вход второй. И так далее по всем операциям. Если операций больше нет, вычисленное значение устанавливается в качестве значения узла.

Все операции выполняются в порядке, в котором они заданы в соответствующих свойствах.

1.11.1.1 Сигнал простой

Узел типа "Сигнал простой" предназначен для хранения и обработки значения источника, или другого сигнала или операции. Позволяет произвести какие-либо действия с этим значением с помощью добавления нужных операций в свойство «Операции после чтения».

1.11.1.2 Постоянная

Узел типа "Постоянная" предназначен для хранения данных выбранного формата и позволяет изменять их в ходе выполнения программы. Связывание с источником не предусмотрено. Изменение осуществляется с помощью специализированных функций или блока установки значения в задаче ПЛК.

1.11.1.3 Массив

Массив - совокупность каких-либо объектов, указанных в свойстве «Входные значения». Нумерация объектов в массиве начинается с 0. Может использоваться для взаимодействия с демультиплексором.

1.11.1.4 Сигнал комплексный

Сигнал используется для взаимодействия с виджетом «Цифровой индикатор». Позволяет контролировать входное значение и генерировать события переходу входного значения через уставки.

Содержит в себе свойства для настройки следующих параметров:

- ВАУ – верхняя аварийная уставка;
- ВПУ - верхняя предаварийная уставка;
- НПУ - нижняя предаварийная уставка;
- НАУ – нижняя аварийная уставка;
- ФНЧ – фильтр низких частот.

1.11.2 Источник

Источник - специальный тип узлов, предназначенных для передачи данных от внешних устройств.

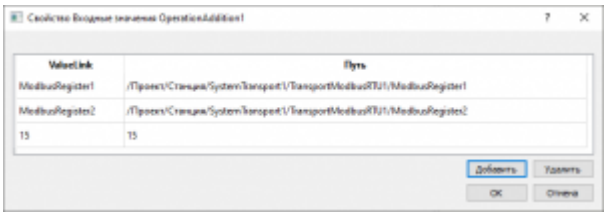
1.11.2.1 Регистр Modbus

Регистр Modbus используется для приема и передачи данных по протоколам Modbus-RTU, Modbus-TCP. Данный тип узов также может использоваться для организации работы некоторых алгоритмов программы (например для взаимодействия с контроллером событий).

1.11.2.2 Источник индексный

Узел типа "Источник индексный" используется для обработки данных, полученных по протоколу Upd, а так же в массивах для индексации значений.

1.11.3 Операция



Входные значения операций

Узлы типа "Операция" используются для реализации вычислительных алгоритмов в задаче ПЛК или отдельно, при их размещении в группе "Сигналы".

В качестве входных значений операций выбираются константные числа, другие узлы группы «Сигналы», регистры или блоки задачи ПЛК.

1.11.3.1 Операция «Сложение»

Выходным значением является сумма входных значений. В качестве входных значений можно выбирать как узлы, так и конкретные числа формата int или double.

1.11.3.2 Операция «Вычитание»

Выходным значением является разность первого входного значения и суммы всех последующих значений. В качестве входных значений можно выбирать как узлы, так и конкретные числа формата int или double.

1.11.3.3 Операция «Умножение»

Выходным значением является произведение значений всех входных значений. В качестве входных значений можно выбирать как узлы, так и конкретные числа формата int или double.

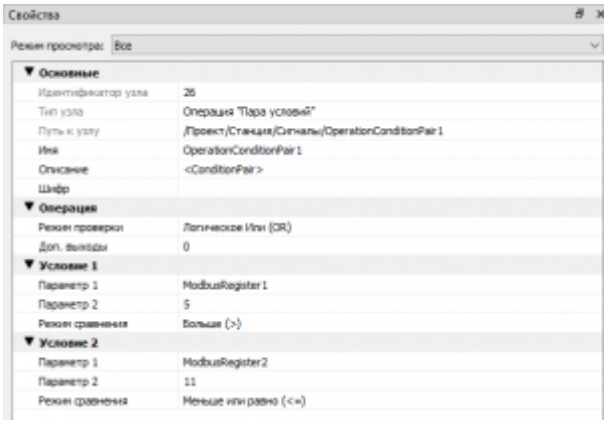
1.11.3.4 Операция «Деление»

Выходным значением является результат от деления первого входного значения на произведение всех последующих значений. В качестве входных значений можно выбирать как узлы, так и конкретные числа формата int или double.

1.11.3.5 Операция «Условие»

Узел типа "Операция условие" предназначен для сравнения значений двух узлов. Для условия задаются сравниваемые параметры и выбирается режим сравнения. В качестве входных значений можно выбирать как узлы, так и конкретные числа формата int или double.

1.11.3.6 Операция «Пара условий»



Операция «Пара условий»

Операция содержит в себе два условия. Для каждого условия задаются индивидуальные сравнимаемые параметры и выбирается режим сравнения.

Выходное значение операции зависит от ее режима проверки – при выбранном режиме «Логическое ИЛИ» выходное значение равно логической «1» если одно из условий равно «1», при «Логическом И»- если оба условия равны «1».

1.11.3.7 Операция «Логическое ИЛИ»

Если одно из входных значений становится равно логической «1» (контакты замкнуты), то выходное значение становится логической «1» (выход включен).

1.11.3.8 Операция «Логическое И»

Если все входные значения становятся равны логической «1», выходное значение становится логической «1» (выход включен).

1.11.3.9 Операция «Логическое НЕ»

Если входное значение блока становится равно логическому «0», то выходное значение становится логической «1», и наоборот.

1.11.3.10 Операция «Таймер»

Таймер предназначен для выполнения каких-либо действий через одинаковые промежутки времени (например запись значения в узел). Промежуток между исполнениями действий задается в свойстве «Интервал срабатывания», а сами действия в свойстве «Действия OnTimer». Для работы операции ее необходимо добавить в свойство «Узлы» задачи с периодическим или разовым режимом выполнения.

1.11.3.11 Операция «Мультиплексор»

Мультплексор осуществляет перевод нескольких булевых переменных в целочисленное значение по формуле: «выход = вход1 + вход2*2^1 + вход3*2^2 + ...».

Например если в качестве входных значений указаны два регистра, то выходное значение мультиплексора будет формироваться следующим образом:

Значение регистра 1	Значение регистра 2	Выходное значение
0	0	0
1	0	1
0	1	2
1	1	3

1.11.3.12 Операция «Демультиплексор»

Демультиплексор осуществляет перевод целочисленных значений из массива в несколько булевых переменных, при этом включается выход, номер которого соответствует числу из массива (например первый элемент массива равен 5, на пятом выходе появится значение true). Нумерация выходов начинается с нуля.

1.11.3.13 Операция «Генератор случайного числа»

Выходным значением является случайное число в диапазоне между указанным минимальным и максимальным значениями. Для работы операции ее необходимо добавить в свойство «Узлы» задачи с периодическим или разовым режимом выполнения.

1.11.3.14 Операция «Скрипт C++»

Операция «Скрипт C++» предназначена для реализации разнообразных алгоритмов на языке C++. Операция предусматривает реализацию одной или нескольких функций, выполняющих необходимые пользователю действия.

Для выполнения скрипта, можно добавить в свойство «Узлы» задачи с периодическим или разовым режимом выполнения. Другим способом вызова срабатывания скрипта является добавление его в действие «Выполнение скрипта».

Пример скрипта, выполняющего получение значения определенного узла и запись измененного значения в другой узел:

```
float val()
{
    float f = GetNodeValueAsFloat("/Конфигурация1/stationSimple1/Signals1/Random2");
    float f1 = f*100.0;
    SetNodeValueAsFloat("/Конфигурация1/stationSimple1/Signals1/Constant3", f1);
    return f*10.0;
}
```

Приведенный выше скрипт возвращает исходное значение, умноженное на 10. Возвращаемое таким образом значение записывается в выход операции для возможности считывания в других операциях, скриптах и т. д.

Скрипты могут возвращать значения следующих типов:

- void;
- float;
- double;
- int.

Подробное описание реализации языка C++ в AgavaSCADA/AgavaPLC приведено на странице "Описание языка C++ в AgavaPLC".

1.11.4 Объект

Узлы типа "Объект" предназначены для взаимодействия с аппаратными компонентами контроллера. Для всех объектов, являющихся субмодулями, необходимо указать слот, в котором они установлены. Доступ к значениям входов/выходов субмодуля осуществляется через его вложенные узлы.

ПЛК процессорный модуль

Предоставляет возможность регулирования подсветки экрана; включения и отключения индикаторов работы, аварии и программы.

ПЛК субмодуль DI

Позволяет получать доступ к значениям входов субмодуля DI.

ПЛК субмодуль R

Позволяет управлять значениями выходов субмодуля R.

ПЛК субмодуль SIM

Позволяет управлять значениями выходов субмодуля SIM.

ПЛК субмодуль DO6

Позволяет управлять значениями выходов субмодуля DO6.

ПЛК субмодуль AI

Позволяет получать доступ к значениям входов субмодуля AI и задавать их тип.

ПЛК субмодуль AIO

Позволяет получать доступ к значениям входов субмодуля AIO, управлять значениями выходов и задавать тип входов и выходов.

ПЛК субмодуль TMP

Позволяет получать доступ к значениям входов субмодуля TMP и задавать их тип.

1.12 Задачи

Группа «Задачи» предназначена для группировки узлов типа «Задача», которые используются для циклического или разового выполнения созданных пользователем алгоритмов.

1.12.1 Задача ПЛК

Данный узел предоставляет возможности для связывания между собой других узлов (таких как сигналы, субмодули ПЛК, постоянные и т. д.) в удобной графической форме (см. Рисунок 64). Для перехода к редактору необходимо щелкнуть правой кнопкой мыши на задачу ПЛК и выбрать в открывшемся контекстном меню команду «Открыть графическое представление» или дважды щелкнуть левой кнопкой мыши на задаче ПЛК в дереве проекта.

Рисунок 64 – Окно графического редактирования задачи «ПЛК-60»

Панель инструментов задачи ПЛК, расположенная в верхней части окна, содержит следующие элементы (см. Рисунок 16):

- Выпадающий список выбора режима подписи портов.
- Выпадающий список выбора режима подписи блоков.
- Кнопка запуска режима симулятора.
- Кнопка запуска онлайн отладки.
- Кнопка печати задачи.
- Инструменты управления симуляцией.

Добавлять узлы в задачу можно перетаскиванием мышью из окна функциональных блоков или из окна дерева проекта. Для связывания узлов между собой необходимо, зажав левую кнопку мыши на выходе одного узла, протянуть линию ко входу другого и

затем отпустить левую кнопку мыши для соединения блоков.

Созданную связь можно удалить, выделив ее левой кнопкой мыши и нажав клавишу "Del". Связь можно "разорвать", то есть добавить разрыв, нажав по ней правой кнопкой мыши и выбрав в контекстном меню команду «Добавить разрыв».

Для выделения нескольких блоков рамкой зажмите клавишу "Shift" и выделяйте блоки мышью с нажатой левой кнопкой.

Задача ПЛК предоставляет возможности для проведения тестирования и отладки проекта, описанные в разделе #Проверка и отладка проекта.

1.12.1.1 Блоки задачи ПЛК

Помимо операций, описанных в разделе #Операция, для реализации алгоритмов в задачу ПЛК можно добавлять блоки, описанные ниже.

Функциональный блок

Используется для выделения каких-либо действий в отдельный блок, для упрощения понимания работы алгоритма программы. Содержит входы, выходы и блоки, обеспечивающие работу внутренней логики. Позволяет проводить изолированную отладку, при которой значения на входах функционального блока вручную задаются пользователем. Блок, работающий в данном режиме выделяется черной рамкой. Для включения изолированной отладки необходимо открыть блок и нажать кнопку «Изолированная отладка» на панели управления задачей.

Блок установки значения

Блок осуществляет запись значения, поданного на вход in0, в узел, указанный в свойстве «Узел для чтения/записи»

Блок задержки

Блок задержки используется для создания задержки передачи значения с входа блока на выход на один цикл обработки задачи.

Блок «Битовое OR»

Для вычисления значения будет произведена операция "ИЛИ" над каждым битом значения в отдельности. Например: 1010 OR 0110 = 1110

Блок «Логическое И»

Если все входные значения становятся равны логической «1», выходное значение становится логической «1» (выход включен).

Блок «Битовое AND»

Для вычисления значения будет произведена операция "И" над каждым битом значения в отдельности. Если все биты, стоящие на одинаковых позициях в двоичных представлениях значений, поступающих на входы блока, равны логической «1», то на выходе элемента в этой позиции появляется логическая «1» (выход включен). Например: 1010 AND 0110 = 0010

Блок «Логическое НЕ»

Если входное значение блока становится равно логическому «0», то выходное значение становится логической «1», и наоборот.

Блок «Битовое XOR»

Если только на один из входов блока функции поступает логическая «1», то на выходе элемента появляется логическая «1». Если на входы блока функции поступают целочисленные значения, то операция будет произведена над каждым битом значения в отдельности. Например: 1010 XOR 0110 = 1100

Блок «Битовое SHR»

Побитовый логический сдвиг вправо используется для выполнения операции побитового логического сдвига операнда X вправо на N бит с дополнением нулями слева.

Блок «Битовое SHL»

Побитовый логический сдвиг влево используется для выполнения операции побитового логического сдвига операнда X влево на N бит с дополнением нулями справа.

Блок «Таймер»

Таймер предназначен для выполнения каких-либо действий (см. раздел 4.6) через одинаковые промежутки времени (например запись значения в узел).

Блок «Мультиплексор»

Мультплексор осуществляет перевод нескольких булевых переменных в целочисленное значение по формуле: «выход = вход1 + вход2*2^1 + вход3*2^2 + ...»

Блок «Демультиплексор»

Демультиплексор осуществляет перевод целочисленных значений из массива в несколько булевых переменных, при этом включается выход, номер которого соответствует числу из массива (например первый элемент массива равен 5, на пятом выходе появится значение true). Нумерация выходов начинается с нуля.

Блок «Генератор случайного числа»

Выходным значением является случайное число в диапазоне между указанным минимальным и максимальным значениями.

Блок «Скрипт C++»

Операция «Скрипт C++» предназначена для реализации разнообразных алгоритмов на языке C++. Операция предусматривает реализацию одной или нескольких функций, выполняющих необходимые пользователю действия. Для выполнения скрипта, необходимо добавить его в качестве узла для задачи или задачи ПЛК.

Пример скрипта, выполняющего получение значения определенного узла и запись измененного значения в другой узел:

```
float val()
{
    float f = GetNodeValueAsFloat("/Конфигурация1/stationSimple1/Signals1/Random2");
    float f1 = f*100.0;
    SetNodeValueAsFloat("/Конфигурация1/stationSimple1/Signals1/Constant3", f1);
    return f*10.0;
}
```

Приведенный выше скрипт возвращает исходное значение, умноженное на 10. Возвращаемое таким образом значение записывается в выход операции для возможности считывания в других операциях, скриптах и т. д.

Скрипты могут возвращать значения следующих типов:

- void;
- float;
- double;
- int.

Подробное описание реализации языка C++ в AgavaSCADA/AgavaPLC приведено на странице "Описание языка C++ в AgavaPLC".

Блок «Таймер с задержкой»

Таймер с задержкой используется для операции задержки передачи и/или отключения сигнала. На выходе таймера появится сигнал логической «1» с задержкой относительно фронта входного сигнала продолжительностью равной интервалу включения и выключится по спаду входного сигнала с задержкой продолжительностью равной интервалу выключения. В том случае, если необходимо использовать блок только как таймер с задержкой включения (отключения) интервал выключения (включения) устанавливается равным 0.

Блок «Генератор прямоугольных импульсов»

Генератор прямоугольных импульсов используется для формирования прямоугольных импульсов пульсации. На выходе генератора формируются импульсы с заданными параметрами длительности включенного и отключенного импульса на время действия управляющего сигнала на входе (сигнал логической «1»).

Блок «Триггер RS»

RS-триггер с приоритетом выключения используется для переключения с фиксацией состояния во время поступления коротких импульсов на соответствующий вход. На выходе появится логическая «1» по фронту сигнала на входе Set, которая будет сброшена в «0», при поступлении импульса на вход Reset.

Блок «Триггер SR»

SR-триггер с приоритетом включения используется для переключения с фиксацией состояния в случае поступления коротких импульсов на соответствующий вход. На выходе появится логическая «1» по фронту сигнала на входе Set, которая будет сброшена в «0», при поступлении импульса на вход Reset.

Блок «Детектор переднего фронта импульса»

Детектор переднего фронта импульса используется в случае необходимости иметь реакцию на изменение состояния дискретного входного сигнала. На выходе генерируется единичный импульс при изменении состояния входа из «0» в «1».

Блок «Детектор заднего фронта импульса»

Детектор заднего фронта импульса используется в случае необходимости иметь реакцию на изменение состояния дискретного входного сигнала. На выходе генерируется единичный импульс при изменении состояния входа из «1» в «0».

Блок «D-триггер»

D-триггер используется для формирования импульса включения выхода на интервал времени импульса на входе D, выходной интервал будет синхронизирован с тактовой частотой на входе C. На выходе Q триггера появится сигнал логической «1» по фронту тактовых импульсов на входе C при наличии сигнала логической «1» на входе D. Возврат выхода в сигнал логического «0» произойдет по фронту тактовых импульсов на входе C при наличии сигнала логического «0» на входе D. Вход Set принудительно устанавливает выход в состояние логической «1». Вход Reset является приоритетным и устанавливает выход в состояние логического «0».

Блок «Селектор»

Если на вход Value подается логическая «1», то результатом выполнения операции блока на выходе является входной сигнал ValueT. Если же Value равен логической «0», то результатом выполнения операции блока на выходе является входной сигнал ValueF.

Блок «Больше чем»

На выход подается логическая «1», если значение in0 больше чем значение in1.

Блок «Равенство»

На выход подается логическая «1», если значение in0 равно значению in1.

Блок «Абсолютное значение числа»

Результатом операции функции на выходе является модуль подаваемого на вход значения. Если число на входе положительное, то на выход подается само число, если отрицательное – положительное число, получаемое от перемены его знака с «-» на «+».

Блок «Возведение в степень»

Результатом операции функции на выходе является значение, равное значению in0, возведенному в степень значения N.

Блок «PID-регулятор»

Регулятор PID применяется для поддержания заданного значения параметра. На вход Pv подается текущее значение регулируемой величины, на Sp – ее заданное значение. С помощью изменения коэффициентов ПИД регулятора kP, kI и kD достигается оптимальный режим работы. Min задает минимальное значение на выходе, Max –

максимальное. time - период регулятора (в мс.).

Блок «PWM-регулятор»

Применяется для преобразования аналогового сигнала в дискретный с помощью ШИМ. На вход in подается текущее значение регулируемой величины, Min задает минимальное значение на входе, Max – максимальное. time - период регулятора (в мс.)

1.13 Устройства

Группа «Устройства» предназначена для группировки узлов типа "Устройство", выполняющие разнообразные задачи, например работа с внешними накопителями и ПЗУ.

Retain-накопитель дает возможность хранить значения регистров, постоянных и сигналов в постоянной памяти контроллера, что позволяет им сохранять свои значения после отключения питания.

Для сохранения переменных необходимо перечислить их в свойстве «Узлы» и указать путь сохранения на выбранном накопителе.

1.14 Отчеты

Данный узел доступен только при исполнении проекта на настольном компьютере. На панелях оператора и контроллерах МПР этот узел не доступен.

Отчет – технологический документ, содержащий информацию о ходе производственного процесса, обработанную по заложенным в отчет алгоритмам. Отчеты формируются на основе макетов (статической части отчетов) путем добавления в них свойств и значений заданных параметров технологического процесса, которые формируются (выбираются из базы данных технологических параметров и проходят соответствующую обработку) по алгоритмам, размещенным в этих шаблонах.

Данные для отчета предоставляет выборка, осуществляющая получение данных и их обработку из базы данных.

1.14.1 Создание нового отчета

Создание любого отчета производится в два этапа.

1. Создание необходимых узлов в конфигурации: сам отчет и необходимое количество выборок.
2. Составление макета отчета.

Создание узла типа «Отчет» в конфигурации и выборок для него производится аналогично созданию узлов других типов. К станции добавляется узел типа «Отчеты» с помощью команды контекстного меню «Добавить узел», далее к узлу типа «Отчеты» добавляется отчет и к отчету добавляется необходимое количество выборок.

Выборка – специальный узел, предназначенный для выборки данных из архива и их обработки для последующего отображения в отчете.

Перед переходом к редактированию макета желательно проделать следующие операции:

- в свойстве отчета «Путь к файлу отчета» установить имя файла, в котором будет храниться макет отчета;
- в свойстве «Источники» выборки (выборок) отчета установить перечень архивных сигналов (коннекторов), которые будут выводиться в отчете.

1.14.2 Редактирование макета отчета

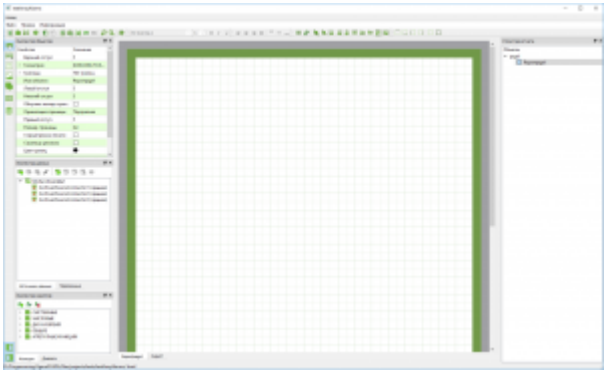


Рисунок 42 – Редактор макета отчета

Для редактирования макета отчета необходимо переключиться из редактора конфигурации в редактор макета отчета с помощью команды «Открыть редактор макета отчета», доступной из контекстного меню дерева конфигурации, отображаемого при щелчке правой кнопкой мыши на узле типа «Отчет».

В окне редактора расположены:

- панели инструментов;
- панель объектов отчета;
- окно «Инспектор объектов»;
- окно «Инспектор данных»;
- окно «Инспектор скриптов»;
- область построения макета отчета;
- окно «Структура отчета».

Редактор макета позволяет вызывать команды сочетаниями клавиш:

- Ctrl + N – новый макет;
- Ctrl + O – загрузить макет;
- Ctrl + S – сохранить макет;
- Ctrl + Shift + S – сохранить макет под новым именем;
- Ctrl + P – предварительный просмотр;
- Ctrl + Z – отмена изменений;

- Ctrl + Shift + Z – возврат отмененных изменений;
- Ctrl + C – копировать объект;
- Ctrl + X – вырезать объект;
- Ctrl + V – вставить объект;
- Ctrl + Стрелки – переместить выделенный объект;
- Shift + Стрелки – изменить размер выделенных объектов;
- Del – удалить выделенные объекты;
- Shift + левая кнопка мыши – создать область выделения.

1.14.2.1 Окно «Инспектор объектов»

Окно «Инспектор объектов» содержит список свойств выделенного объекта и предназначено для их редактирования.

1.14.2.2 Окно «Инспектор данных»

Окно «Инспектор данных» содержит список доступных для использования источников данных (выборок) и предназначено для заполнения блока «Данные» элементами соответствующего типа.

1.14.2.3 Окно «Инспектор скриптов»

Окно «Инспектор скриптов» содержит список доступных для использования в отчете скриптов.

1.14.2.4 Окно «Структура отчета»

Окно «Структура отчет» отображает структуру отчета в виде дерева связанных элементов.

1.14.3 Элементы отчета

Для составления макета отчета доступны следующие элементы:

- текст;
- данные;
- изображение;
- фигура;
- компоновка (Layout).

1.14.3.1 Текст

Элемент «Текст» служит для вывода надписей или содержимого полей источников данных.

Объект «Текст» обладает очень широкими возможностями. Он умеет отображать текст, рамку, заливку. Текст может быть отображен любым шрифтом любого размера, цвета и стиля. Большинство настроек делаются визуально с помощью панелей инструментов.

Одна из самых главных особенностей этого универсального объекта – это возможность отображения не только статичного текста, но и текста с применением вычисляемых выражений. Причем, выражения могут располагаться в объекте вперемешку с текстом. Рассмотрим простой пример – поместим в объект «Текст» следующую строку: «Привет! Сегодня $S{now()}».

Если запустить отчет на построение, мы увидим приблизительно следующее: «Привет! Сегодня 20180803». В процессе построения отчета выражение в тексте, заключенное $S{}, было вычислено и полученное значение вставлено обратно в текст. Объект «Текст» может содержать любое количество выражений, смешанных с обычным текстом. В скобки можно заключать и одиночные переменные, и выражения, например, $S{1+2*(3+4)}. В выражениях можно использовать константы, переменные ($V{}), функции, поля БД ($D{}).

Свойства элемента «Текст»:

Наименование свойства	**Описание**
alignment	Выравнивание текста по вертикали и горизонтали
angle	Поворот надписи
autoHeight	Автоматический подбор высоты
autoWidth	Автоматический подбор ширины
backgroundColor	Цвет заливки
backgroundMode	Режим заливки
backgroundOpacity	Прозрачность заливки
borders	Границы
content	Содержимое объекта «Текст»
font	Шрифт
fontColor	Цвет шрифта
foregroundOpacity	Прозрачность шрифта
geometry	Размер и расположение объекта
itemLocation	Расположение объекта (страница или бэнд)
margin	Отступы
objectName	Имя объекта
stretchToMaxHeight	Установка значения высоты самого высокого объекта на бэнде
trimValue	Удаление пробелов в начале и в конце надписи

Для редактирования свойства «Content» используется диалоговое окно «Text Item Editor», вызов которого осуществляется двойным щелчком левой кнопки мыши на элементе «Текст».

Свойство «Content» может содержать:

- текст;
- значения переменных;
- значения полей из набора данных;
- значения, формируемые посредством исполнения скрипта.

Для вывода значений переменных используется синтаксис «$V{имяПеременной}» (без

кавычек), для вывода значения поля данных – «$D{имяНабораДанных.ИмяПеременной}», для вывода результата исполнения скрипта – «$S{телоСкрипта}».

1.14.3.2 Диаграмма

1.14.3.3 Изображение

Элемент для вывода изображений:

Наименование свойства	Описание
autoSize	Подгонять размер под размер изображения
borders	Границы
content	
datasource	Источник данных
field	Поле данных
geometry	Размер и расположение
image	Изображение
itemLocation	Расположение объекта (страница или бэнд)
objectName	Имя объекта
stretchToMaxHeight	Устанавливать значение высоты самого высокого объекта на бэнде

1.14.3.4 Фигура

Элемент для вывода фигур:

Наименование свойства	Описание
borders	Границы
geometry	Размер и расположение
itemLocation	Расположение объекта (страница или бэнд)
lineWidth	Толщина линии
objectName	Имя объекта
opacity	Прозрачность
penStyle	Стиль линии
shape	Фигура
shapeBrush	Стиль заливки
shapeBrushColor	Цвет заливки
shapeColor	Цвет фигуры
stretchToMaxHeight	Устанавливать значение высоты самого высокого объекта на бэнде

1.14.3.5 Горизонтальная компоновка (Layout)

Элемент, позволяющий объединять несколько элементов в группу:

Наименование свойства	Описание
borders	Границы
geometry	Размер и расположение
itemLocation	Расположение объекта (страница или бэнд)
objectName	Имя объекта
stretchToMaxHeight	Устанавливать значение высоты самого высокого объекта на бэнде

1.14.4 Разделы отчета (бэнды)

Бэнд предназначен для размещения других элементов отчета. Бэнд может быть нескольких типов:

- Report Header – заголовок отчета;
- Report Footer – завершение отчета;
- Page Header – верхний колонтитул страницы отчета;
- Page Footer – нижний колонтитул страницы отчета;
- Data – данные отчета;
- SubDetail – подчиненные данные отчета;
- SubDetailHeader – заголовок подчиненных данных;
- SubDetailFooter – завершение подчиненных данных;
- GroupHeader – заголовок группы;
- GroupFooter – завершение группы.

Бэнды применяются для логической группировки объектов. К примеру, размещение объекта на элементе «Данные» типа «Page Header», приводит к отображению этого объекта в верхней части каждой страницы готового отчета. Аналогичным образом «Page Footer» выводится внизу каждой страницы со всеми лежащими на нем объектами.

Общие для всех элементов «Данные» свойства:

Наименование свойства	Описание
autoHeight	Автоматический подбор высоты
backgroundColor	Цвет заливки бэнда
borders	Границы
geometry	Размер
keepBottomSpace	Сохранение отступа от нижней границы бэнда
objectName	Имя объекта
printIfEmpty	Отображение пустого бэнда
splittable	Разделение бэнда, если он не влезает на страницу

Report Footer:

Наименование свойства	Описание
maxScalePercent	Максимальный процент, на который можно уменьшить бенд, если он не влезает на страницу. Если бэнд даже после сжатия не влезает на страницу, он будет перенесен полностью или частично – в зависимости от настроек бэнда

Data:

Наименование свойства	Описание
datasource	Источник данных. Data-бэнд будет сформирован для каждой строки в источнике данных
keepFooterTogether	Если Report Footer не влезает на страницу, то он будет перенесен на следующую страницу совместно с последним экземпляром Data-бэнда
sliceLastRow	Указывает генератору отчетов на то, можно ли разрезать последний экземпляр Data-бэнда или его нужно перенести целиком

Бэнды данного типа предназначены для вывода на печать данных из наборов. Набор данных - это заранее неизвестное количество строк (записей), каждая из которых содержит определенное количество колонок (полей).

SubDetail:

Наименование свойства	Описание
datasource	Источник данных. SubDetail-бэнд будет сформирован для каждой строки в источнике данных

SubDetailHeader, SubDetailFooter:

Наименование свойства	Описание
printAlways	Печатать, даже если SubDetail пустой

GroupHeader:

Наименование свойства	Описание
groupFieldName	Поле, по которому осуществляется группировка. Экземпляр GroupHeader будет формироваться при смене значения в этом поле

1.14.5 Функции

1.14.5.1 Агрегатные функции

Агрегатные функции предназначены для обработки значений, содержащихся в резделах Data и SubDetail.

1.14.5.1.1 AVG - вычисление среднего значения.

Пример использования:

```
$S{AVG("TextItem4","DataBand1");}
```

1.14.5.1.2 SUM - вычисление суммы значений.

Пример использования:

```
$S{SUM("TextItem8","DataBand1");}
```

1.14.5.1.3 MIN

1.14.5.1.4 MAX

1.14.5.1.5 COUNT

1.14.5.2 Числовые функции

1.14.5.2.1 numberFormat - форматирование числа

Синтаксис:

```
numberFormat(значение, 'формат', точность)
```

Форматы:

- `'f'` — фиксированная запись (например, `"123.45"`);
- `'g'` — общий формат (автоматически выбирает между `'f'` и `'e'`);
- `'e'` — экспоненциальная запись (например, `"1.2345e+02"`);

Пример использования:

```
$S{numberFormat($D{fetcharchiverdata1.ArchiversSourceConnector0 (среднее)}, 'f', 3)}       -> 123.495
```

Источник — https://docs.kb-agava.ru/index.php?title=Узлы_проекта_AgavaSCADA/AgavaPLC&oldid=3069

Эта страница в последний раз была отредактирована 16 июля 2025 в 17:25.