Универсальная среда разработки Agava

Редакция от 19.10.2026 · docs.kb-agava.ru

Содержание

1 Введение ... 3
1.1 Версии программного обеспечения ... 3
1.2 Дополнительные статьи и документы ... 3
2 Разработка проекта ... 4
2.1 Создание нового проекта ... 4
2.2 Редактирование существующего проекта ... 4
3 Структура и состав проекта ... 5
4 Редактирование свойств узлов ... 6
4.1 Редактирование свойств узлов с помощью окна «Свойства» ... 7
4.2 Редактирование свойств группы узлов ... 7
4.3 Групповое редактирование свойств узлов ... 7
4.4 Функция групповой замены источников в узлах ... 8
5 Редактирование графического представления узла ... 9
5.1 Связывание источников и виджетов ... 10
5.2 Редактирование многослойных композиций ... 10
5.3 Использование якорей ... 11
6 Редактирование текстовых программ на языках AgavaScript (AS) и StructuredText (ST) ... 12
7 Редактирование программ на блочных языках (FBD, CFC) ... 12
8 Сохранение проекта ... 12
9 Проверка и отладка проекта ... 13
9.1 Эмуляция проекта ... 13
9.2 Мониторинг сигналов ... 13
9.3 Мониторинг событий ... 14
9.3.1 Использование мониторинга ... 14
9.3.2 Фильтрация событий ... 14
9.4 Отладка приложения ... 15
10 Компиляция проекта ... 15
10.1 Проверка проекта ... 15
10.1.1 Установка пароля на файл конфигурации ... 15
11 Загрузка проекта в устройство ... 16
11.1 Загрузка из среды разработки ... 16
11.2 Загрузка через USB-Flash (для ПО-40 и ПО-50) ... 17
11.3 Загрузка через браузер (для МПР-60) ... 17
12 Отладка проекта ... 17
12.1 Назначение ... 17
12.2 Локальная отладка ... 18
12.2.1 Отладка отдельной функции ... 18
12.2.2 Отладка проекта в локальной среде исполнения ... 18
12.3 Удалённая отладка ... 19
12.3.1 Подготовка к отладочному запуску ... 19
12.4 Просмотр состояния программы и управление выполнением ... 19
12.5 Окна отладчика ... 20
12.5.1 Окно «Локальные переменные» ... 20
12.5.2 Окно «Глобальные переменные» ... 20
12.5.3 Окно «Выражения» ... 21

12.5.4 Окно «Стек вызовов» ... 21
12.5.5 Окно «Точки останова» ... 21
12.6 Изменение значений и завершение сеанса ... 21
13 Использование ИИ-агентов через MCP ... 22
14 Обслуживание контроллеров со средой исполнения AgavaPLC ... 22
14.1 Окно «Информация о среде исполнения» ... 22
14.1.1 Назначение ... 22
14.1.2 Открытие окна ... 23
14.1.3 Отображаемые сведения ... 23
14.1.4 Доступность обновлений ... 24
14.1.5 Подготовка к обновлению ... 24
14.1.6 Порядок обновления ... 25
14.1.7 Выполнение установки на устройстве ... 25
14.1.8 Проверка результата обновления ... 25
14.1.9 Диагностические сообщения ... 25
15 Библиотеки ресурсов AgavaSCADA/AgavaPLC ... 26
15.1 Открытие окна ресурсов ... 26
15.2 Создание библиотеки ... 27
15.3 Подключение существующей библиотеки ... 27
15.4 Формирование структуры библиотеки ... 27
15.5 Добавление файлов ... 28
15.6 Назначение ресурса объекту проекта ... 28
15.7 Копирование и удаление элементов ... 28
15.8 Соблюдение правил адресации ... 29
15.9 Сохранение и перенос проекта ... 29
15.10 Проверка результата ... 29

Универсальная среда разработки предназначена для создания и редактирования программного обеспечения для панелей оператора АГАВА ПО, многоканальных программируемых реле АГАВА МПР-60 и систем диспетчеризации на базе SCADA-системы AgavaSCADA.

1 Введение

Универсальная среда разработки Agava имеет удобный и эргономичный пользовательский интерфейс со всеми инструментами, необходимыми для создания эффективных проектов-приложений, предназначенных для исполнения на ПК, панелях оператора или персональных компьютерах.

Среда разработки совмещает редакторы текста, редакторы визуальных представлений объектов, компоновщики структуры, средства связывания объектов и других частей проекта, анализаторы проекта в одном приложении и дает пользователю возможность ведения эффективной разработки проектов-приложений от начала и до завершения в одной программе, без использования разнородных приложений-утилит и необходимости переключения между ними.

1.1 Версии программного обеспечения



При использовании среды разработки совместно с панелями оператора АГАВА ПО или многоканальными программируемыми реле АГАВА МПР-60 необходимо использовать соответствующую версию среды разработки

Для определения подходящей версии среды разработки необходимо определить версию среды исполнения, установленной в панели оператора или реле. Тип и версия среды исполнения указаны в системной утилите прибора. Например, "AgavaPLC 1.6", "AgavaPLC 1.7" и т.д. Для использования со средой исполнения AgavaPLC 1.6 необходимо использовать среду разработки версии 1.6, для среды исполнения AgavaPLC 1.7 необходима среда разработки версии 1.7 и так далее.

1.2 Дополнительные статьи и документы

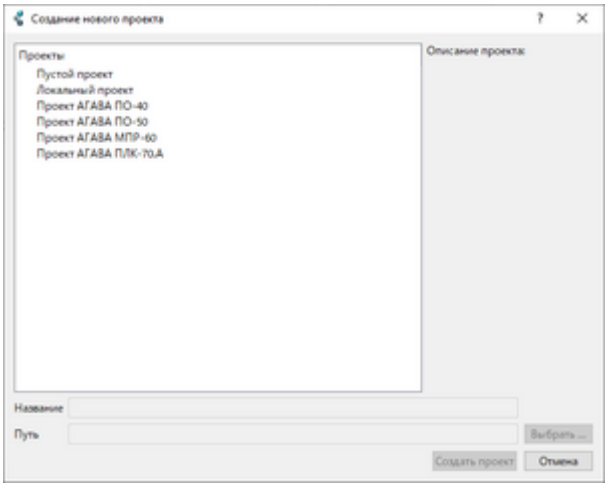
- Описание узлов AgavaSCADA/AgavaPLC.
- Свойства узлов в AgavaSCADA/AgavaPLC.
- Транспорты AgavaSCADA/AgavaPLC.
- Описание языка C++ в AgavaSCADA/AgavaPLC.
- Описание языка ST в AgavaSCADA/AgavaPLC.
- Объектная модель AgavaSCADA/AgavaPLC.
- Описание базовых классов AgavaSCADA/AgavaPLC.
- Коды ошибок и предупреждений AgavaSCADA/AgavaPLC.
- Рекомендации по разработке приложений AgavaSCADA/AgavaPLC.

Полный перечень всех документов, связанных с разработкой приложений на AgavaSCADA/AgavaPLC, приведен на странице категории AgavaSCADA/AgavaPLC.

2 Разработка проекта

2.1 Создание нового проекта

Создание нового проекта в среде разработки осуществляется с помощью команды «Создать проект…» в меню «Файл». После вызова данной команды откроется диалоговое окно создания нового проекта (см. Рисунок 1).



Окно "Создание нового проекта"

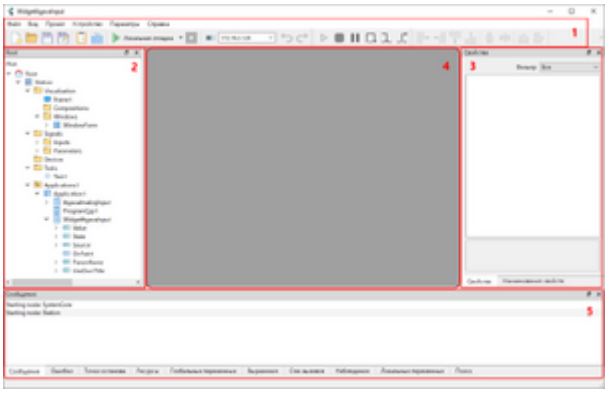
Для создания проекта необходимо в списке типов проектов выбрать необходимый, ввести имя проекта и путь для его сохранения, после чего нажать кнопку «Создать проект».

Доступны несколько типов проектов:

- «Пустой проект» – пустой проект, не содержащий никаких узлов. Данный тип можно использовать для создания произвольного проекта "с нуля".
- «Локальный проект» – предварительно настроенный проект, содержащий необходимые подсистемы и предназначенный для реализации простой системы для одного устройства, при выборе данного типа пользователь имеет возможность выбрать нужные ему подсистемы и получить полностью рабочий проект в минимальной конфигурации.
- «Проект АГАВА ПО-40» - проект, предназначенный для исполнения на панели оператора АГАВА ПО-40.
- «Проект АГАВА ПО-50» - проект, предназначенный для исполнения на панели оператора АГАВА ПО-50.
- «Проект АГАВА ПЛК-60.А (МПР-60)» - проект, предназначенный для исполнения на контроллере АГАВА ПЛК-60.А (ранее МПР-60).
- «Проект АГАВА ПЛК-70.А» - проект, предназначенный для исполнения на контроллере АГАВА ПЛК-70.А.

2.2 Редактирование существующего проекта

Редактирование существующего проекта начинается с открытия файла проекта путем вызова команды «Открыть…» в меню «Файл». После загрузки файла в основном окне среды разработки будут отображены свойства и параметры проекта (см. Рисунок 2).



Основное окно среды разработки AgavaPLC с загруженным проектом

По умолчанию основное окно среды разработки содержит перечисленные ниже части:

1. Строка заголовка приложения, панель меню и панель инструментов.
2. Окно дерева проекта. Предоставляет доступ ко всей структуре проекта (см. раздел Узлы проекта).
3. Окно свойств (см. раздел #Редактирование свойств узлов).
4. Область для отображения содержимого, например графического представления окна, текста программы или задачи ПЛК.
5. Область вспомогательных окон:
    1. Окно "Сообщения".
    2. Окно "Ошибки".
    3. Окно "Точки останова".
    4. Окно "Ресурсы".
    5. Окно "Глобальные переменные".
    6. Окно "Выражения".
    7. Окно "Стек вызовов".
    8. Окно "Наблюдение".
    9. Окно "Локальные переменные".
    10. Окно "Поиск".

После открытия проекта можно приступить к его редактированию.

По завершению всех операций по редактированию необходимо выполнить проверку проекта на корректность с помощью команды «Проверить проект», расположенной в меню «Проект» (см. 6.1).

3 Структура и состав проекта

Типичный проект содержит следующие части:

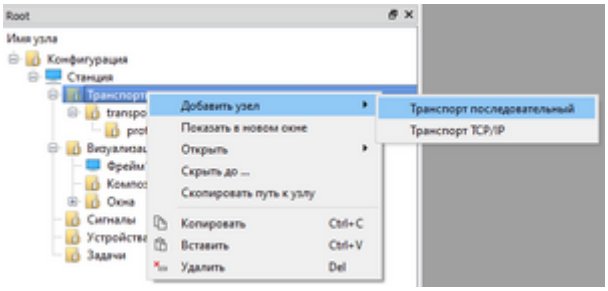
- станция – важнейший объект, управляющий работой собственных подсистем и осуществляющий обмен данными с другими станциями;
- транспортная система, выполняющая сбор данных с УСО и других устройств;
- система визуализации, обеспечивающая отображение информации в различных видах.

Проект также может содержать следующие части:

- система архивирования, сохраняющая данные в БД для последующего извлечения и анализа;

- система проверки прав пользователей, предназначенная для контроля доступа пользователя к объектам в режиме исполнения;
- система тревог, предназначенная для оповещения пользователя об обнаруженных нарушениях;
- система журналирования, выполняющая архивирование зарегистрированных событий в БД для последующего извлечения и анализа.

Проект имеет древовидную структуру, вследствие чего все узлы связаны друг с другом. Узлы нижних уровней подчиняются узлам верхних уровней. К примеру, станции подчинены все подсистемы, а каждой подсистеме подчинены все узлы, которыми она управляет.



Добавление узла в проект AgavaPLC

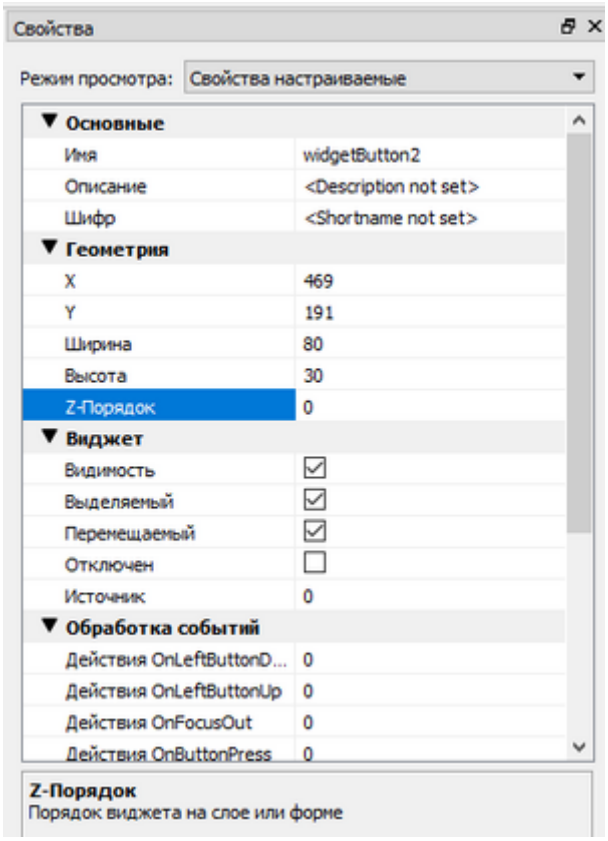
Перед началом разработки проекта необходимо определить системы, необходимые для реализации проекта. Взаимодействие с системами осуществляется через дерево проекта (см. Рисунок 2), в которое добавляются нужные узлы. Для добавления нового узла, необходимо нажать правой кнопкой мыши на узел верхнего уровня, которому он подчинён, и выбрать в контекстном меню команду «Добавить узел».

Описание частей проекта приведено в документе Узлы проекта AgavaSCADA/AgavaPLC.

4 Редактирование свойств узлов

Среда разработки обладает развитыми возможностями по редактированию свойств узлов. Описание доступных типов свойств приведно в статье Свойства узлов в AgavaPLC/AgavaSCADA.

4.1 Редактирование свойств узлов с помощью окна «Свойства»



Окно «Свойства» виджета «Кнопка»

Редактирование свойств всех узлов осуществляется с помощью окна «Свойства».

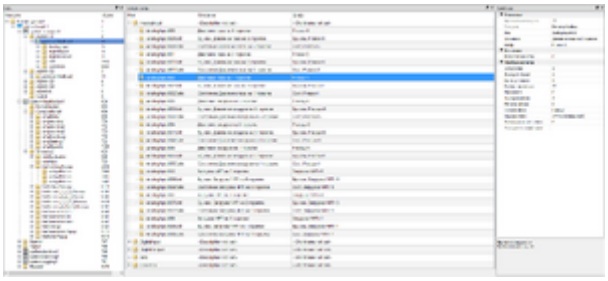
В окне «Свойства» в виде таблицы отображаются все доступные свойства узла, объединенные в группы. В левой колонке отображены наименования свойств, в правой отображены их значения. Свойства, недоступные для изменения, имеют наименование серого цвета. В нижней части окна расположена область подсказки, содержащая пояснение к выделенному свойству.

4.2 Редактирование свойств группы узлов

В случае, когда необходимо изменить свойства у нескольких узлов, можно воспользоваться возможностью группового редактирования свойств. Для этого нужно выделить несколько узлов в дереве конфигурации либо на графическом представлении с помощью левой кнопки мыши и нажатой кнопки «Ctrl». При этом в окне свойств будут отображены только те свойства, что имеются у всех выделенных узлов. Далее можно изменить нужное свойство, при этом после завершения редактирования измененное значение будет установлено во всех выбранных узлах.

4.3 Групповое редактирование свойств узлов

При необходимости редактирования разных свойств у большого количества узлов можно воспользоваться специальным окном «Список узлов».



Окно "Список узлов"

Открытие окна выполняется с помощью вызова команды «Открыть список узлов» в контекстном меню дерева конфигурации. Открытое окно содержит все узлы, вложенные в выбранные при вызове команды.

Окно позволяет редактировать свойства непосредственно в ячейках таблицы, без переключения на окно «Свойства». Групповое редактирование свойств узлов через окно «Свойства» также доступно.

Щелчок правой кнопкой мыши на списке вызывает контекстное меню, содержащее команды:

- “Свойства”;
- “Экспорт в *.txt”;
- “Импорт из *.txt”;
- “Удалить узел”.

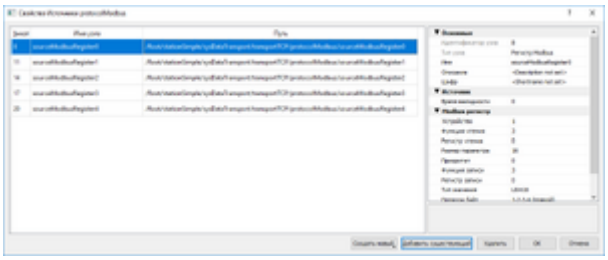
Команда «Свойства» открывает диалоговое окно выбора отображаемых свойств.

Команды «Экспорт», «Импорт» выполняют экспорт списка свойств узлов в текстовый файл и импорт соответственно.

Команда «Удалить узел» производит удаление выделенного узла (узлов).

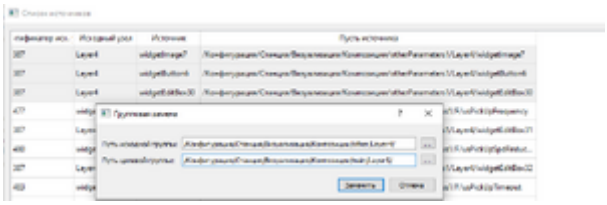
4.4 Функция групповой замены источников в узлах

При разработке сложных проектов может возникнуть необходимость заменить источники сразу в большом количестве узлов. Для этой цели используется функция групповой замены источников. Необходимо перейти в редактирование свойств типа «Список источников», которое производится в соответствующем окне (см. Рисунок 8).



Окно редактирования свойств типа «Список источников»

Затем необходимо использовать функцию групповой замены (см. Рисунок 9), выделив нужные узлы, нажав на них правой кнопкой мыши и выбрав «Групповая замена».



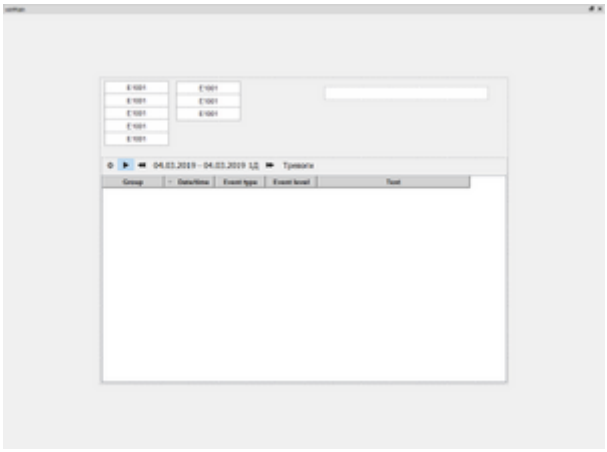
Окно функции «Групповая замена»

5 Редактирование графического представления узла

Редактирование содержимого окон, слоев, форм, композиций и других элементов, предназначенных для отображения на экране, осуществляется в окне графического представления. Открытие окна возможно с помощью команды «Открыть графическое представление» в контекстном меню дерева конфигурации или двойного нажатия левой кнопки мыши для узлов следующих типов:

- фрейм;
- форма;
- окно с композицией;
- слой.

Открытое графическое представление выбранного узла может содержать все уже размещенные на нем элементы визуализации и (или) другие узлы и позволяет добавить новые.



Открытое графическое представление окна «winMain»

На рисунке 10 представлено графическое представление окна «winMain», обведенное пунктирной рамкой серого цвета, обозначающей внешнюю границу открытого узла. Тип рамки пользователь может выбрать по своему усмотрению, выбор производится в окне настроек среды разработки, доступном через команду «Настройки» в меню «Параметры».

Щелчок правой кнопкой мыши на пустом месте в открытом графическом представлении узла вызывает контекстное меню, содержащее команды удаления и вставки виджетов из буфера обмена, а также команды добавления новых виджетов доступных типов.

Добавление новых виджетов также может осуществлять путем их перестаскивания в рабочую область из панели виджетов.

Графическое представление позволяет перемещать выделенные виджеты кнопками перемещения курсора на клавиатуре, а также изменять их размеры кнопками перемещения курсора при нажатой кнопке Shift.

5.1 Связывание источников и виджетов

Связывание виджетов и других узлов с источниками данных необходимо для того, чтобы виджеты или другие узлы могли отображать или обрабатывать значения, хранящиеся в источниках.

Для связи виджетов с источниками данных можно воспользоваться обычным способом связывания через окно «Свойства». К примеру, связывание размещенного на форме виджета «Текстовая метка» с регистром Modbus осуществляется добавлением регистра Modbus в свойство виджета «Источник».

Вместе с обычным способом связывания существует и другой, более быстрый способ. Для быстрого связывания регистра с виджетом достаточно нажать на регистр в дереве конфигурации левой кнопкой мыши и перенести его на виджет в открытом графическом представлении.



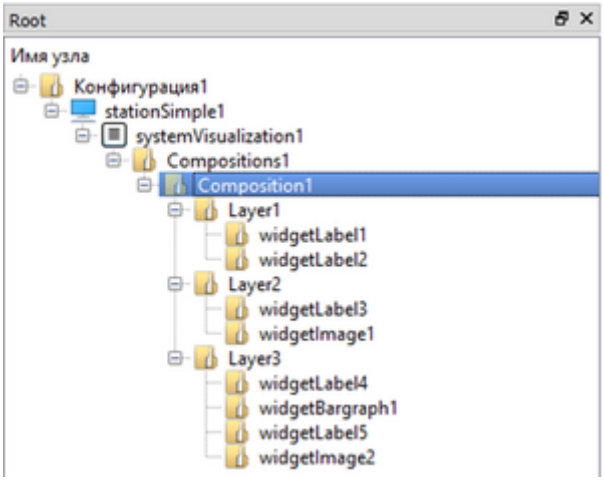
Связывание узла «sourceModbusRegister0» и виджета

При помещении курсора над виджетом последний будет заштрихован сеткой зеленого или красного цвета, а курсор изменит свой вид на стрелку с квадратиком (см. рисунок 11).

Зеленая штриховка виджета означает отсутствие связи выделенного виджета с источником данных, в этом случае связывание будет произведено. Красная штриховка говорит об ошибках связывания виджета и сигнала/источника.

5.2 Редактирование многослойных композиций

Редактирование многослойных композиций принципиально не отличается от редактирования форм, кроме одного нюанса. Так как композиция может состоять из нескольких слоев, и виджеты на слоях могут располагаться один над другим, для использования наиболее удобного способа редактирования такой композиции можно воспользоваться возможностью открытия каждого слоя в отдельном окне.



Пример структуры композиции «Composition1», состоящей из трех слоев

Для открытия графического представления слоя нужно воспользоваться командой «Открыть графическое представление» аналогично открытию графического представления формы или композиции.

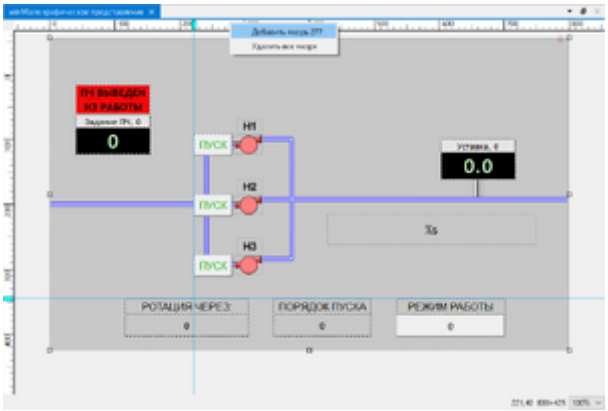
Открыв одновременно композицию и слой, при редактировании слоя можно видеть, что содержимое композиции также изменяется и наоборот, при редактировании композиции изображение слоя будет изменяться.

Слой обладает важным свойством «Прозрачность для действий». Данное свойство позволяет регулировать доступность виджетов, расположенных на нижележащих слоях, для действий пользователя. Например, если у верхнего слоя данное свойство выключено, то виджеты, расположенные на всех нижележащих слоях, будут недоступны для взаимодействия. Если же у верхнего слоя свойство включено, то виджеты нижележащего слоя будут доступны.

5.3 Использование якорей

Данная возможность доступна с версии `AS 1.6.32+`.

При работе с графическими представлениями окон и композиций, содержащих большое количество виджетов, могут быть полезны якоря - создаваемые пользователем метки на размерных линейках.



AS 1.6.32+: Графическое представление формы с установленными якорями

Якоря позволяют контролировать положение групп виджетов благодаря связанным с ними линиям привязки. Линии привязки, позволяют быстрее выстраивать группы виджетов благодаря "прилипанию".

Создание якоря выполняется следующими способами:

- Двойным кликом левой кнопки мыши по размерной линейке.
- Кликом правой кнопки мыши на размерной линейке и выбором команды "Добавить якорь".

Удаление якоря выполняется следующими способами:

- Двойным кликом левой кнопки мыши по якорю на размерной линейке.
- Кликом правой кнопки мыши на размерной линейке и выбором команды "Удалить якорь".

Созданные якоря можно перемещать по размерной линейке с помощью мыши.

6 Редактирование текстовых программ на языках AgavaScript (AS) и StructuredText (ST)

Универсальная среда разработки имеет современный и функциональный встроенный редактор тестовых программ, оснащенный следующими функциями:

- Ввод и редактирование текста.
- Управление объявлениями программных элементов.
- Поиск, замена и перемещение по тексту.
- Подсказки и переход к определению.
- Статический анализ тестов программ.
- Отладка.

Полное описание редактора размещено в документе Редактор текстовых программ AgavaSCADA/AgavaPLC.

7 Редактирование программ на блочных языках (FBD, CFC)

Универсальная среда разработки имеет современный и функциональный встроенный редактор блочных программ, оснащенный следующими функциями:

- Составление и редактирование блок-схемы;
- Управление порядком исполнения блоков;
- Организация блоков с помощью сетей;
- Отладка.

Полное описание редактора размещено в документе Редактор программ на блочных языках AgavaSCADA/AgavaPLC.

8 Сохранение проекта

Для сохранения проекта на диск вызовите команду «Файл->Сохранить проект» или «Файл->Сохранить проект как...», если необходимо сделать копию открытого проекта и продолжить работу в ней после сохранения.

9 Проверка и отладка проекта

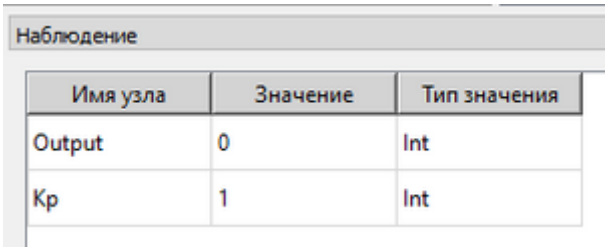
9.1 Эмуляция проекта

Проверить работоспособность проекта можно, не загружая его в контроллер. Для этого существует возможность запустить среду исполнения прямо на компьютере.

Для этого необходимо:

1. Установить текущий проект как проект по умолчанию. Для этого необходимо вызвать команду "Проект->Установить проект по умолчанию..." и выбрать ранее сохраненный файл конфигурации.
2. Запустить среду исполнения. Для этого нужно вызвать команду меню "Проект->Запуск среды исполнения". После этого будет выполнен запуск в новом окне среды исполнения.

9.2 Мониторинг сигналов



Окно "Наблюдение"

Функция мониторинга сигналов позволяет наблюдать значения, хранящиеся в узлах проекта при исполнении проекта на целевом устройстве.

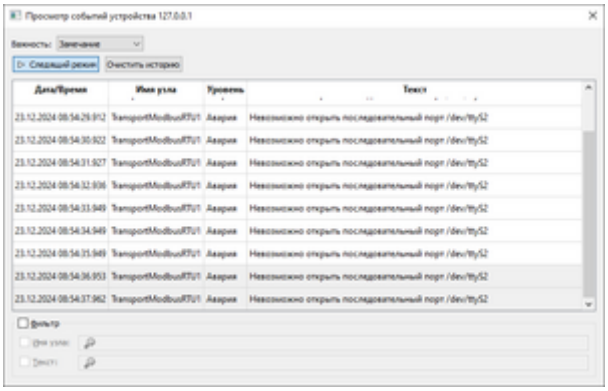
Для запуска данной функции необходимо:

1. Запустить среду исполнения.
2. Вызвать команду меню «Устройство → Подключение/отключение», в появившемся окне выбрать IP-адрес и порт устройства, значения узлов которого необходимо мониторить. При необходимости просмотра значений узлов и сигналов в режиме эмуляции проекта на персональном компьютере укажите адрес 127.0.0.1. Если необходимо просматривать значения узлов проекта, выполняющегося на контроллере или панели оператора, укажите IP-адрес контроллера или панели.
3. Нажать кнопку «Подключение».

Поле подключения к среде исполнения нужно добавить в окно "Наблюдение" узлы, значения которых необходимо наблюдать. Такие узлы должны иметь возможность чтения или записи значений – это регистры Modbus и другие узлы типа "Источник", сигналы.

Для добавления узлов в окно "Наблюдение" необходимо нажать на нужном узле в дереве проекта правой кнопкой мыши и выбрать команду «Добавить в окно наблюдения». Выбранные узлы будут отображаться в окне «Наблюдение» вместе со считанными значениями.

9.3 Мониторинг событий



Окно просмотра событий

Мониторинг событий позволяет наблюдать за исполнением проекта, работающего в устройстве. При включении мониторинга пользователь получает информацию о возникающих нефатальных ошибках и предупреждениях, возникающих в ходе исполнения проекта со следующими признаками:

- метка времени, соответствующая времени возникновения события;
- степень важности события;
- тип события - информационное или события активирования, деактивирования и квитирования тревог;
- имя узла, сформировавшего событие.
- текст события.

9.3.1 Использование мониторинга

Для старта мониторинга необходимо подключиться к нужному устройству способом, рассмотренным в разделе #Мониторинг сигналов, а затем вызвать команду «`Устройство → Просмотр событий`». В открывшемся окне будут выводиться все события, регистрируемые устройством.

С помощью выпадающего списка в левом верхнем углу окна можно выбрать минимальный уровень выводимых событий.

9.3.2 Фильтрация событий

Функция доступна начиная с версии `AgavaSCADA/AgavaPLC 1.2.56+, 1.5.25+`.

В нижнем левом углу окна расположена область настройки фильтра событий, позволяющая фильтровать выводимые события по имени узла и тексту событий. При использовании фильтра возможно задание фильтра в виде:

- простой строки, при этом в список попадают события, содержащие полное совпадение заданной строки; при задании строки возможно в(ы)ключение признака учета регистра символов в строке;
- регулярного выражения, при этом в список попадают события, удовлетворяющие введенное выражение. Правила составления регулярных выражений описаны на странице Использование регулярных выражений.

9.4 Отладка приложения

Подробное описание использования отладчика для отладки программ описано в разделе Отладка проекта.

10 Компиляция проекта

Перед загрузкой проекта в контроллер или запуском эмуляции проекта необходимо откомпилировать проект и получить файл конфигурации, предназначенный для исполнения средой исполнения, установленной в контроллере или на персональном компьютере (в случае разработки системы диспетчеризации). Для этого необходимо вызвать команду меню "Проект->Сохранить конфигурацию", выбрать папку проекта и нажать кнопку "Сохранить".

10.1 Проверка проекта



Окно «Ошибки»

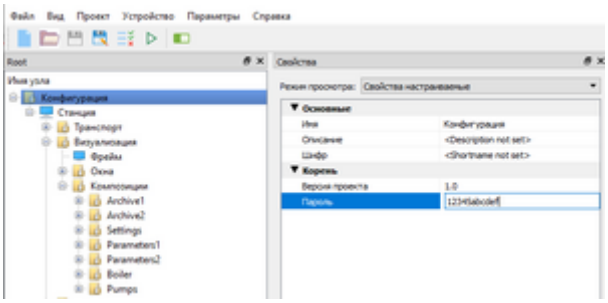
Перед сохранением проекта желательно выполнить его проверку на наличие ошибок с помощью команды «Проект->Проверить проект». После проверки обнаруженные ошибки будут отображены в окне «Ошибки».

Если отображение окна «Ошибки» выключено, его можно включить командой «Вид->Показать окно ошибок».

Для облегчения исправления обнаруженных ошибок окно поддерживает выполнение некоторых команд над обнаруженными ошибками, вызывать которые можно из контекстного меню, отображаемого щелчком правой кнопки мыши на строке с ошибкой в окне. Контекстное меню позволяет вызывать команду «Показать узел в дереве конфигурации», выполняющую соответствующее действие.

Также щелчок левой кнопкой мыши на строке с ошибкой открывает окно «Свойства» и дает возможность изменить в нем все свойства узла, содержащего ошибку.

10.1.1 Установка пароля на файл конфигурации



Установка пароля

Файл конфигурации, загружаемый в контроллер или панель оператора, можно защитить паролем, чтобы предотвратить его несанкционированное изменение.

Для установки пароля на файл конфигурации необходимо выбрать в дереве проекта узел «Конфигурация» и отредактировать его свойство «Пароль». Допускается использование любых цифр, букв и спецсимволов.

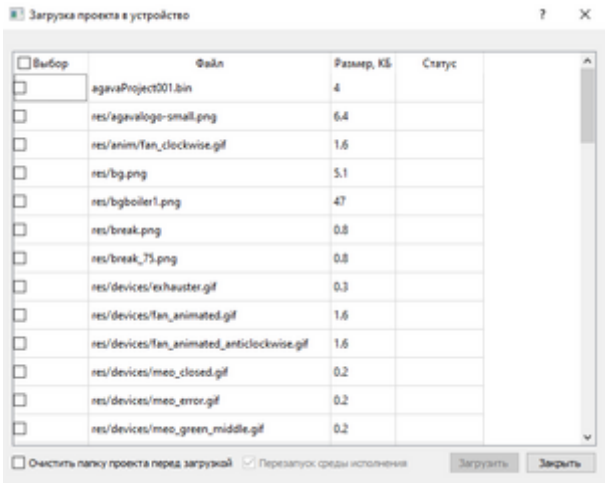
В дальнейшем при открытии файла конфигурации в среде разработки будет появляться окно с запросом пароля.

11 Загрузка проекта в устройство

Загрузка проекта в целевое устройство может быть осуществлена несколькими способами:

- загрузка из среды разработки;
- загрузка с помощью USB-Flash накопителя (для панелей оператора);
- загрузка через браузер (для МПР-60).

11.1 Загрузка из среды разработки



Окно «Загрузка проекта в устройство» после подключения к контроллеру

Для загрузки проекта из среды разработки необходимо произвести следующие действия в указанном порядке:

1. Создайте файл конфигурации.
2. Выполните подключение целевого устройства к ПК по сети Ethernet.
3. После подключения вызовите в среде разработки команду «Проект->Загрузка проекта в устройство».
4. В открывшемся окне будет отображен список файлов проекта, доступных для загрузки в целевое устройство.
5. Выберите необходимые файлы и нажмите кнопку «Загрузить». После нажатия кнопки начнется процесс загрузки файлов, при этом в нижней части окна будет меняться состояние полосы прогресса, а также для каждого файла в таблице будет отображен результат загрузки.
6. После окончания загрузки среда исполнения в целевом устройстве будет перезапущена и загруженный проект будет запущен.

11.2 Загрузка через USB-Flash (для ПО-40 и ПО-50)

Для загрузки проекта в устройство с помощью USB-Flash накопителя необходимо произвести следующие действия в указанном порядке:

1. Подготовьте USB-Flash накопитель, предварительно отформатировав его в файловую систему FAT32.
2. Вызовите в среде разработки команду «Подготовка архива для обновления» или «Сохранить архив проекта» в меню «Проект».
3. Сохраните архив в корень USB-Flash накопителя.
4. Перейдите в системную утилиту, перезагрузив панель оператора, при загрузке во время однократного звукового сигнала нажмите на экран и удерживайте до повторного звукового сигнала. Введите пароль: 111111.
5. Перейдите на вкладку «Обновление», затем подключите USB-Flash накопитель через переходник USB-OTG, выберите из списка загрузочный архив, нажать кнопку «Обновить». Дождитесь сообщения «Обновление успешно» и нажмите кнопку «OK».
6. Перейдите на вкладку «Перезапуск» и нажмите кнопку «Перезапуск».

После перезагрузки панели загруженный проект будет запущен.

11.3 Загрузка через браузер (для МПР-60)

Для загрузки проекта через браузер необходимо произвести следующие действия в указанном порядке.

1. Вызовите в среде разработки команду «Подготовка архива для обновления» или «Сохранить архив проекта» в меню «Проект».
2. Сохраните архив.
3. Откройте системную утилиту, введя IP-адрес контроллера в адресную строку браузера.
4. Перейдите на вкладку «Загрузка ПО». Рисунок 71 - Вкладка системной утилиты «Загрузка ПО»
5. Выберите на компьютере архив с проектом, который необходимо загрузить в контроллер.
6. Нажмите кнопку «Загрузить».
7. Дождитесь окончания загрузки файла и обновления проекта.

12 Отладка проекта

Встроенная отладка проекта доступна в AgavaSCADA (AS) начиная с версии 1.6. Состав команд и средств отображения данных определяется используемой версией среды разработки и среды исполнения; приведённое ниже описание учитывает реализацию версии 1.7.8.

12.1 Назначение

Отладка предназначена для исследования поведения прикладной программы в процессе её выполнения, выявления логических ошибок и проверки соответствия реализованного алгоритма заданным требованиям. Средства отладки обеспечивают установку точек останова, приостановку и возобновление выполнения, пошаговое исполнение, просмотр локальных и глобальных переменных, а также анализ стека вызовов. Для поддерживаемых типов данных предусмотрено изменение значений переменных во время остановки.

Среда разработки поддерживает следующие инструменты отладки:

- локальная отладка на компьютере разработчика с запуском среды исполнения;
- удалённая отладка на контроллере или панели оператора.
- локальный отладочный вызов отдельной функции с заданными входными аргументами.

12.2 Локальная отладка

Локальная отладка применяется для предварительной проверки алгоритмов на компьютере разработчика. Она позволяет исследовать последовательность выполнения программы и преобразование данных до запуска проекта на целевом устройстве. При оценке результатов необходимо учитывать различия аппаратных ресурсов, внешних подключений и временных характеристик компьютера и контроллера.

12.2.1 Отладка отдельной функции

Для локального исследования функции выполняется следующая последовательность действий:

1. Открыть редактор требуемой функции и проверить её исходный текст. Устранить ошибки компиляции.
2. Установить точки останова в исследуемых участках программы с помощью клавиши **F9** либо щелчка левой кнопкой мыши в поле установки точек останова рядом с номером строки.
3. Нажать кнопку **«Начать локальную отладку»** в редакторе функции.
4. Если функция имеет входные аргументы, задать их значения в открывшемся диалоговом окне и подтвердить запуск.
5. После остановки исследовать значения переменных и стек вызовов. Для проверки дальнейшего выполнения использовать команды продолжения и пошагового исполнения, приведённые в подразделе 7.3.2.

При таком запуске подготавливается отдельный контекст вызова выбранной функции. Результат характеризует её выполнение при заданных аргументах и доступном состоянии данных приложения. Проверка взаимодействия задач и внешних устройств требует запуска проекта в соответствующей среде исполнения.

12.2.2 Отладка проекта в локальной среде исполнения

Для запуска проекта под управлением отладчика на компьютере разработчика необходимо:

1. Выполнить сборку командой **«Проект → Сохранить конфигурацию»** и убедиться в отсутствии ошибок в окне **«Ошибки»**.
2. Установить требуемые точки останова.
3. В поле адреса устройства на панели инструментов указать **127.0.0.1** или **localhost**.
4. Выполнить команду **«Отладка среды исполнения»** на панели инструментов.
5. Дождаться запуска локальной среды исполнения и установления соединения с отладчиком.

Среда разработки запускает локальную среду исполнения с подготовленной конфигурацией и включённой отладкой. Для этого на компьютере должна быть доступна программа `rtu.exe`. Просмотр данных и управление выполнением осуществляются теми же средствами, что и при удалённой отладке.

12.3 Удалённая отладка

Удалённая отладка предназначена для исследования программы непосредственно на контроллере или панели оператора. Среда исполнения выполняет программу на целевом устройстве, а среда разработки предоставляет интерфейс управления выполнением и отображения отладочных данных.

12.3.1 Подготовка к отладочному запуску

Перед запуском необходимо обеспечить сетевую доступность целевого устройства и соответствие используемых версий среды разработки и среды исполнения.

Подготовка и запуск выполняются в следующем порядке:

1. Открыть требуемый проект и выполнить сборку командой **«Проект → Сохранить конфигурацию»**. Убедиться в отсутствии ошибок в окне **«Ошибки»**.
2. Установить точки останова в исследуемых участках программы с помощью клавиши **F9** либо щелчка левой кнопкой мыши в поле установки точек останова рядом с номером строки.
3. Указать IP-адрес контроллера или панели оператора в поле выбора адреса устройства на панели инструментов.
4. Выполнить команду **«Отладка среды исполнения»**.
5. Дождаться передачи файлов конфигурации, перезапуска среды исполнения в режиме отладки и установления соединения.

В процессе запуска среда разработки передаёт конфигурацию на устройство, настраивает отладочный режим и инициирует перезапуск среды исполнения. После повторного подключения устанавливается отладочный сеанс. Индикатор **«Подключение/отключение»** отображает состояние соединения с устройством; готовность к управлению выполнением дополнительно определяется состоянием сеанса и доступностью команд на панели отладки.

Отладочный запуск предусматривает перезапуск среды исполнения, а точки останова изменяют временную последовательность выполнения программы. Эти особенности необходимо учитывать при исследовании циклических алгоритмов и взаимодействия с внешним оборудованием.

12.4 Просмотр состояния программы и управление выполнением

Данные переменных и стека вызовов характеризуют состояние программы в момент остановки. После возобновления выполнения эти сведения обновляются при следующей остановке. Текущее состояние отладчика определяет доступность команд управления: во время выполнения доступна приостановка, а во время остановки — продолжение и пошаговое исполнение.

При достижении точки останова выполнение отлаживаемого кода приостанавливается. Среда разработки отображает позицию остановки, значения доступных переменных и стек вызовов. Жёлтым выделением в редакторе обозначается текущая позиция выполнения; её следует интерпретировать как место остановки перед дальнейшим исполнением соответствующего участка программы.

Основные команды управления приведены в таблице. Указаны сочетания клавиш по умолчанию; для настраиваемых команд фактические сочетания определяются параметрами среды разработки.

Команда	Сочетание клавиш	Назначение
Возобновить отладку	F5	Продолжить выполнение до следующего останова или завершения выполнения.
Приостановить отладку	F6	Запросить остановку на ближайшей доступной для отладчика позиции.
Установить или снять точку останова	F9	Изменить наличие точки останова на текущей строке редактора.
Шаг	F10	Продолжить выполнение до следующей доступной позиции на текущем или меньшем уровне вложенности вызовов. Вызов подпрограммы выполняется без пошагового входа в её тело; установленные точки останова сохраняют действие.
Шаг с заходом	F11	Перейти к следующей доступной позиции, включая вход в вызываемую подпрограмму при наличии доступного для отладки кода.
Шаг с выходом	Shift+F11	Продолжить выполнение до выхода из текущей подпрограммы либо до срабатывания точки останова.
Завершить сессию отладки	Кнопка панели отладки	Завершить текущий отладочный сеанс.

Пошаговое выполнение определяется отладочными позициями скомпилированной программы. Поэтому один шаг не во всех случаях соответствует одной строке исходного текста, а вход в системные или встроенные функции может быть недоступен.

Точки останова могут устанавливаться до запуска и изменяться во время сеанса. Если выбранная строка не соответствует исполняемому коду, отладчик может скорректировать положение точки. Фактическое положение следует проверять по отметке в редакторе и данным окна **«Точки останова»**.

12.5 Окна отладчика

Отладочные окна открываются соответствующими командами меню **«Вид»**.

12.5.1 Окно «Локальные переменные»

Данное окно отображает параметры и локальные переменные текущего контекста остановки, а также доступные данные объекта при выполнении метода.

Для переменных приводятся имя, значение и объявление. Состав отображаемых сведений определяется доступностью переменных в текущей позиции программы. Поля составных значений и элементы массивов представляются вложенными строками, если их раскрытие поддерживается отладчиком.

12.5.2 Окно «Глобальные переменные»

Данное окно содержит глобальные переменные модуля, к которому относится выполняемый код.

Глобальные переменные принадлежат модулю и не являются локальными данными отдельного кадра стека. При наличии пространства имён оно учитывается при отображении имени переменной.

12.5.3 Окно «Выражения»

Данное окно предназначено для ведения пользовательского списка наблюдаемых значений.

Добавление выполняется через команду **«Добавить выражение»**, изменение — редактированием записи, удаление — командами контекстного меню.

Значения определяются в контексте текущей остановки. В рассматриваемой реализации поддерживаются имена переменных и квалифицированные имена с указанием области видимости, например `counter`, `::counter` или `Control::counter`.

Вычисление произвольных арифметических выражений, вызов функций, обращение к полям через точку и индексирование массива в строке выражения не поддерживаются.

Недоступное имя или неподдерживаемая запись сопровождаются диагностическим сообщением.

12.5.4 Окно «Стек вызовов»

Данное окно отображает последовательность активных вызовов, приведших к текущей позиции. Для каждого элемента приводятся объявление функции, секция исходного кода, строка и столбец. Двойной щелчок по записи открывает соответствующий участок программы. Такой переход служит навигации по исходному тексту: данные переменных сохраняют контекст фактической остановки.

12.5.5 Окно «Точки останова»

Данное окно содержит перечень установленных точек с указанием строки и программного узла. Контекстное меню позволяет удалить выбранные точки либо весь перечень. Дополнительные сведения о расположении точки доступны во всплывающей подсказке.

При включённом отображении отладочных значений в текстовом редакторе значения переменных также выводятся непосредственно в исходном тексте. Для составного значения может быть открыт отдельный инспектор. Эти сведения относятся к последней остановке и очищаются после продолжения выполнения.

12.6 Изменение значений и завершение сеанса

Во время остановки допускается редактирование значения переменной в столбце **«Значение»**, если её тип и способ представления поддерживают запись. Введённое значение должно соответствовать типу переменной. Изменение передаётся в отлаживаемый процесс и влияет на дальнейшее выполнение текущего запуска. Оно не изменяет исходный текст и само по себе не задаёт начальное значение для последующих запусков.

Для продолжения исследования применяется команда **«Возобновить отладку»**.

Для завершения используется команда **«Завершить сессию отладки»**. При отладке проекта через соединение со средой исполнения эта команда также передаёт запрос на завершение приложения среды исполнения; она не равнозначна продолжению программы без отладчика. Дальнейший запуск определяется режимом работы целевого устройства.

После изменения исходного текста для проверки новой редакции программы необходимо повторить сборку и отладочный запуск. При ошибке подключения следует проверить адрес и доступность устройства, сообщения среды разработки и состояние среды исполнения.

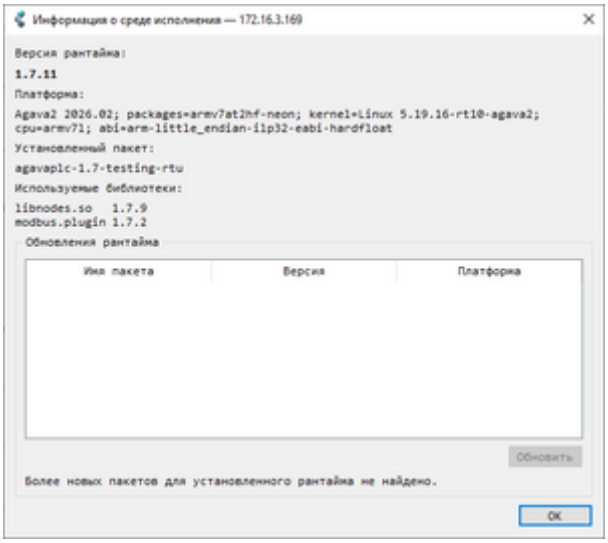
13 Использование ИИ-агентов через MCP

Универсальная среда разработки Agava предоставляет средства взаимодействия с внешними ИИ-агентами через **MCP (Model Context Protocol)**. Подключённый агент может изучать структуру открытого проекта, создавать и изменять узлы, редактировать программные тексты, проверять проект и исследовать его выполнение в локальной среде исполнения.

Полное описание возможностей MCP-интеграции среды разработки, порядка использования и настройки описано в документе "Использование ИИ-агентов через MCP".

14 Обслуживание контроллеров со средой исполнения AgavaPLC

14.1 Окно «Информация о среде исполнения»



Окно "Информация о среде исполнения"

Средой исполнения, или рантаймом, называется программное обеспечение устройства, обеспечивающее выполнение прикладного проекта AgavaSCADA/AgavaPLC.

Данное окно доступно в среде разработки систем версии `AS 1.7.10+`.

14.1.1 Назначение

Окно «Информация о среде исполнения» предназначено для просмотра сведений о программном обеспечении подключённого устройства и выполнения обновления среды исполнения на поддерживаемых платформах.

В окне отображаются:

- версия среды исполнения (рантайма);
- характеристики платформы;
- наименование установленного пакета;
- перечень используемых библиотек с указанием их версий.

Сведения в окне относятся к среде исполнения, с которой установлено текущее соединение.

14.1.2 Открытие окна

Для просмотра сведений необходимо:

1. Указать адрес устройства в среде разработки.
2. Установить соединение командой **«Устройство → Подключение/отключение»** либо соответствующей кнопкой панели инструментов.
3. Выполнить команду **«Устройство → Информация о среде исполнения»**.

Команда открытия окна доступна при установленном соединении. В заголовке окна указывается адрес подключённого устройства. Запрос сведений выполняется автоматически при открытии окна и при восстановлении соединения.

Во время получения данных отображается надпись **«Запрос...»**. Обозначения **«(не указана)»** и **«(не указан)»** означают, что соответствующие сведения не были предоставлены устройством. При потере соединения выводится сообщение **«Нет соединения с устройством»**.

14.1.3 Отображаемые сведения

Элемент окна	Содержание и назначение
«Версия рантайма»	Версия программного обеспечения среды исполнения, сообщённая устройством. Используется для идентификации установленного выпуска и определения доступности обновлений.
«Платформа»	Описание операционной системы и доступных характеристик платформы. Состав сведений зависит от устройства и установленного программного обеспечения.
«Установленный пакет»	Наименование установленного пакета программного обеспечения AgavaSCADA по данным устройства. При наличии нескольких пакетов их наименования отображаются отдельными строками.
«Используемые библиотеки»	Перечень библиотек и соответствующих им версий. Сведения применяются при анализе состава программного обеспечения и обращении в техническую поддержку.
«Обновления рантайма»	Раздел поиска и выбора более новых пакетов для поддерживаемой платформы. Содержит таблицу пакетов, кнопку **«Обновить»** и строку состояния операции.
«OK»	Закрытие окна. На этапе передачи пакета и ожидания ответа устройства закрытие окна блокируется.

В описании платформы могут присутствовать следующие обозначения:

Обозначение	Значение
packages	Архитектура установочных пакетов. Используется при подборе обновления для устройства.

`kernel`	Наименование и версия ядра операционной системы.
`cpu`	Обозначение архитектуры процессора, сообщённое операционной системой.
`abi`	Обозначение двоичного интерфейса сборки среды исполнения.

Архитектура процессора и архитектура пакетов являются самостоятельными характеристиками. Подбор пакетов обновления выполняется по значению архитектуры пакетов.

Текст сведений допускает выделение мышью и копирование для включения в диагностические материалы.

14.1.4 Доступность обновлений

Раздел **«Обновления рантайма»** отображается для устройств платформы ARMv7, обозначаемой в программном обеспечении также как ARM7. Проверка каталога обновлений предусмотрена для среды исполнения версии **1.7.1 и новее**.

Для поиска обновлений компьютер со средой разработки должен иметь сетевой доступ к сайту `files.kb-agava.ru`. Поиск доступных пакетов для обновления начинается автоматически после получения сведений об устройстве.

В таблицу включаются пакеты, удовлетворяющие следующим условиям:

- наименование совпадает с одним из наименований установленных пакетов;
- архитектура пакета соответствует архитектуре пакетов устройства;
- основная числовая версия пакета выше текущей версии рантайма.

Столбец таблицы	**Содержание**
«Имя пакета»	Наименование пакета программного обеспечения.
«Версия»	Версия пакета, включая суффикс ревизии при его наличии.
«Платформа»	Архитектура пакета.

Для выбора устанавливаемого обновления используется выделение одной строки.

Кнопка **«Обновить»** становится доступной после выбора пакета при установленном соединении с устройством. Всплывающая подсказка содержит путь к файлу пакета в каталоге обновлений.

Сообщение **«Более новых пакетов для установленного рантайма не найдено»** означает, что доступные обновления для рантайма отсутствуют. При невозможности определить параметры устройства или получить каталог выводится соответствующее диагностическое сообщение, в этом случае отсутствие строк в таблице не подтверждает отсутствие обновлений.

14.1.5 Подготовка к обновлению

	Обновление предусматривает завершение среды исполнения и прерывание выполнения прикладного проекта. Процедуру следует проводить в период, когда такая остановка допустима условиями эксплуатации устройства. Перед началом сохраняют необходимые изменения проекта и резервные копии используемых настроек.

14.1.6 Порядок обновления

1. Установить соединение с требуемым устройством и открыть окно **«Информация о среде исполнения»**.
2. Проверить адрес устройства в заголовке окна, текущую версию рантайма, наименование установленного пакета и описание платформы.
3. Дождаться завершения поиска в разделе **«Обновления рантайма»**.
4. Выделить строку требуемого пакета и нажать **«Обновить»**. Дополнительное подтверждение запуска операции не запрашивается.
5. Дождаться загрузки пакета на компьютер со средой разработки и его передачи на устройство. Ход операции отображается в строке состояния. На время операции выбор другого пакета блокируется.
6. Дождаться результата предварительной проверки и подтверждения приёма задания на установку. При успешной передаче выводится сообщение **«Пакет передан на установку. Рантайм завершится и будет запущен повторно после обработки пакета»**.
7. Дождаться обработки пакета и повторного запуска среды исполнения. После подтверждения приёма задания соединение с устройством закрывается в рамках штатного завершения рантайма.
8. Восстановить соединение с устройством и повторно открыть окно. Проверить, что поле **«Версия рантайма»** соответствует выбранному выпуску. При наличии ревизии сборки сопоставить и её обозначение с ожидаемым результатом.
9. Проверить состояние прикладного проекта и его запуск в соответствии с конфигурацией устройства.

Продолжительность обновления определяется размером пакета, скоростью соединения и временем выполнения операций на устройстве.

На этапе загрузки пакета на компьютер закрытие окна отменяет загрузку. После передачи пакета на устройство закрытие окна блокируется до получения ответа либо завершения ожидания. После подтверждения приёма задания дальнейшая обработка выполняется на устройстве и продолжается независимо от окна среды разработки.

14.1.7 Выполнение установки на устройстве

После получения пакета среда исполнения выполняет предварительную проверку пакета с обновлением. При отрицательном результате дальнейшая установка не начинается; в окно передаются описание и код ошибки.

При успешной проверке пакета выполняется его установка, соединение со средой разработки прерывается.

14.1.8 Проверка результата обновления

Подтверждение результата обновления выполняется путем проверки версии рантайма после восстановления соединения. Номер версии, отображаемый в верхней части окна должен соответствовать версии выбранного для установки пакета.

14.1.9 Диагностические сообщения

Ситуация	Порядок действий
Отсутствует соединение с устройством	Проверить адрес и доступность устройства, восстановить соединение и повторно открыть окно.

Проверка обновлений недоступна для установленной версии	Уточнить поддерживаемый порядок первоначального обновления у администратора устройства или поставщика программного обеспечения.
Недостаточно данных для поиска обновлений	Проверить наличие версии рантайма, наименования установленного пакета и архитектуры пакетов. Передать отсутствующие сведения и диагностическое сообщение администратору.
Ошибка получения каталога или загрузки пакета	Проверить сетевой доступ компьютера со средой разработки к каталогу обновлений. После устранения причины повторно открыть окно для поиска пакетов.
Пакет отклонён при предварительной проверке	Зафиксировать описание и код ошибки. Проверить соответствие выбранного пакета устройству; при необходимости обратиться к администратору.
Не получен ответ на запрос или соединение прервано до подтверждения приёма задания	Проверить фактическое состояние устройства и результат предыдущей операции до повторной передачи пакета.
После повторного подключения версия не соответствует выбранному выпуску	Проверить версию пакета. Повторить обновление.

При обращении в техническую поддержку целесообразно предоставить:

- наименование модели контроллера;
- версию среды исполнения и версию среды разработки Agava;
- сведения из окна, наименование выбранного пакета;
- описание и код ошибки;
- время выполнения операции.

15 Библиотеки ресурсов AgavaSCADA/AgavaPLC

Библиотеки ресурсов AgavaSCADA/AgavaPLC предназначены для централизованного хранения файлов, используемых объектами проекта, и их повторного применения при разработке. Библиотека сохраняется в отдельном файле с расширением .res и содержит ресурсы, а также структуру папок, определяющую их организацию. Назначение ресурса объекту осуществляется посредством ссылки на библиотеку и путь к соответствующему элементу внутри неё.

15.1 Открытие окна ресурсов



Окно "Ресурсы" с открытой библиотекой

Работа с библиотеками выполняется при открытом проекте в окне «Ресурсы». Для его отображения необходимо выбрать команду «Вид → Показать окно „Ресурсы“». Предусмотрены древовидное и объектное представления. Древовидное представление

используется для управления структурой библиотек; объектное — для просмотра их содержимого. В объектном представлении переход в библиотеку или папку осуществляется двойным щелчком, возврат к предыдущему уровню — кнопкой «Вверх».

15.2 Создание библиотеки

Для создания новой библиотеки следует вызвать контекстное меню корневого узла ресурсов и выбрать «Добавить узел → Библиотека ресурсов». После создания необходимо выделить библиотеку и задать её параметры в окне свойств.

Следует различать наименование библиотеки и значение свойства «Имя файла». Наименование используется при обращении к ресурсам из объектов проекта; свойство «Имя файла» определяет файл .res, в котором сохраняется содержимое библиотеки. Рекомендуется устанавливать содержательные и уникальные наименования библиотек, а для каждой самостоятельной библиотеки использовать отдельный файл.

Свойства «Номер сборки» и «Дата сборки» формируются автоматически при сохранении библиотеки и доступны только для чтения.

15.3 Подключение существующей библиотеки

Для использования ранее подготовленной библиотеки необходимо вызвать контекстное меню корневого узла ресурсов, выбрать «Добавить существующую библиотеку ресурсов» и указать один или несколько файлов .res.

Если выбранный файл находится в каталоге проекта или его подкаталогах, путь к нему сохраняется относительно каталога проекта. Для файла, расположенного вне каталога проекта, сохраняется абсолютный путь. Подключение существующей библиотеки использует выбранный файл по месту его размещения; автоматическое копирование в каталог проекта данной операцией не предусмотрено.

Для обеспечения переносимости рекомендуется предварительно размещать библиотеки в каталоге проекта. При подключении библиотеки необходимо проверить её наименование, путь и состав ресурсов.

15.4 Формирование структуры библиотеки

В библиотеке допускается размещать отдельные ресурсы и папки. Папки могут содержать ресурсы и другие папки, что позволяет формировать многоуровневую структуру.

Для создания папки следует выделить библиотеку либо существующую папку и выбрать в контекстном меню «Добавить узел → Папка». Структуру целесообразно формировать по назначению ресурсов, например выделять папки для изображений оборудования, элементов интерфейса и звуковых сообщений.

Наименования папок и ресурсов участвуют в формировании адресов. Поэтому рекомендуется использовать уникальные имена в пределах одного родительского узла и сохранять расширения файлов ресурсов.

15.5 Добавление файлов

Для включения файлов в библиотеку необходимо выделить библиотеку или требуемую папку, выбрать «Добавить узел → Ресурс», затем указать один или несколько файлов. Наименования создаваемых ресурсов первоначально соответствуют именам выбранных файлов.

При добавлении содержимое файла считывается в ресурс, а при сохранении библиотеки записывается в файл .res. Таким образом, сохранённая библиотека содержит данные включённых в неё файлов. Последующее изменение исходного файла на диске само по себе не обновляет ранее добавленный ресурс.

Диалог добавления допускает выбор файлов любых типов. Однако возможность использования конкретного формата определяется объектом, которому назначается ресурс. Например, включение файла в библиотеку ещё не означает, что он может быть отображён как изображение или воспроизведён как звук.

15.6 Назначение ресурса объекту проекта

Для использования библиотечного ресурса следует выделить объект проекта и открыть соответствующее свойство, поддерживающее выбор ресурса, например «Изображение». В меню кнопки выбора необходимо выполнить команду «Выбрать ресурс», указать библиотеку и требуемый файл. Для перехода во вложенную папку используется двойной щелчок; выбор файла можно завершить двойным щелчком по нему либо командой «Выбрать» его контекстного меню.

Ссылка на ресурс имеет вид Библиотека:Папка/Файл. Например, запись `Equipment:Images/Pump.png` обозначает ресурс `Pump.png`, находящийся в папке `Images` библиотеки `Equipment`. Одному ресурсу могут соответствовать ссылки из нескольких объектов проекта.

Команда «Выбрать файл» задаёт обращение к внешнему файлу. Такой файл сохраняет зависимость от своего расположения в файловой системе и не становится содержимым библиотеки вследствие назначения объекту.

15.7 Копирование и удаление элементов

Для копирования следует выделить элементы в дереве ресурсов, выполнить команду «Копировать», выбрать допустимый родительский узел и выполнить команду «Вставить». Поддерживаются сочетания клавиш `Ctrl+C` и `Ctrl+V`. Папка копируется вместе с её содержимым.

Вставка ограничивается допустимой структурой: библиотеки размещаются на корневом уровне, папки и ресурсы — внутри библиотек или папок. После копирования библиотеки необходимо проверить её наименование и свойство «Имя файла», обеспечив самостоятельное размещение копии.

Удаление выделенных элементов выполняется командой «Удалить» или клавишей `Delete`. Перед удалением ресурса, папки либо библиотеки необходимо проверить их использование объектами проекта и при необходимости изменить соответствующие ссылки. Удаление библиотеки из структуры проекта не следует рассматривать как операцию удаления её файла с диска.

15.8 Соблюдение правил адресации

Поиск библиотечного ресурса выполняется по наименованию библиотеки и пути внутри неё. Поэтому изменение наименования библиотеки, переименование ресурса или папки, а также перенос ресурса в другую папку могут нарушить ранее заданные ссылки.

После изменения структуры необходимо проверить свойства объектов, использующих затронутые ресурсы, и повторно назначить их при необходимости. При выборе одноимённых файлов из разных папок следует контролировать полный адрес ресурса.

Наименование библиотеки и путь к файлу .res имеют различное назначение. Изменение одного параметра не следует считать эквивалентом изменения другого.

15.9 Сохранение и перенос проекта

Сохранение изменённых библиотек выполняется при сохранении проекта. При записи обновляются сведения о сборке библиотеки. Для успешного сохранения каталог размещения файла .res должен существовать и быть доступен для записи.

Существующая подключённая библиотека сохраняется по указанному для неё пути. Если один файл библиотеки используется несколькими проектами, изменение его содержимого может отразиться на каждом из них после повторной загрузки. Для независимого редактирования рекомендуется создавать отдельную копию файла.

При переносе проекта необходимо передавать все используемые файлы .res, сохраняя предусмотренное относительное расположение. Внешние файлы, назначенные непосредственно объектам, также подлежат переносу с последующей проверкой путей.

15.10 Проверка результата

После завершения работы следует сохранить проект, повторно открыть его и проверить наличие библиотек, состав папок и корректность назначенных ресурсов. Затем необходимо проверить использование ресурсов в соответствующих объектах: отображение изображений, применение макетов и воспроизведение звуков, если такие функции задействованы.

Для звуковых ресурсов системы тревог дополнительно требуется проверить настройку разрешения воспроизведения звуков. Наличие звукового файла в библиотеке и назначение ссылки являются отдельными условиями и сами по себе не обеспечивают воспроизведение.