Редактор программ на блочных языках AgavaSCADA/AgavaPLC

Редакция от 19.10.2026 · docs.kb-agava.ru

Содержание

1 Назначение 2
2 Открытие редактора и рабочее пространство 2
3 Сети и порядок выполнения 3
  3.1 Создание и организация сетей 3
  3.2 Зависимости и явный порядок 3
  3.3 Передача данных между сетями 3
4 Добавление блоков 4
  4.1 Выбор из каталога 4
  4.2 Добавление из незавершённой связи 4
  4.3 Количество входов 5
5 Входы, выходы и связи 5
  5.1 Создание связи 5
  5.2 Значение неподключённого входа 5
  5.3 Инверсия 5
  5.4 Удаление 6
6 Переменные 6
  6.1 Локальные переменные 6
  6.2 Глобальные переменные 6
  6.3 Чтение и запись 7
  6.4 Таблица переменных 7
7 Функции, параметры и возврат результата 7
8 Функциональные блоки и EN/ENO 8
  8.1 Функциональные блоки 8
  8.2 Разрешение выполнения 8
9 Размещение, выделение и копирование 8
10 Проверка и сохранение 9
  10.1 Диагностика схемы 9
  10.2 Сохранение 9
11 Отладка 10
12 Пример: вычисление результата функции 10
13 Типовые затруднения 10
14 Рекомендуемый порядок работы 11

1 Назначение

Редактор программ на блочных языках предназначен для создания и изменения программ и функций на языках **FBD** и **CFC**. Алгоритм представляется схемой, состоящей из блоков, связей, переменных и параметров. Блоки выполняют вычисления, вызывают функции и функциональные блоки, записывают результаты в переменные.

В проекте используются узлы **«Программа (FBD)»**, **«Программа (CFC)»**, **«Функция (FBD)»** и **«Функция (CFC)»**. Редактор предоставляет общие средства работы с блоками, но способы их размещения различаются.

Представление	Организация схемы
FBD	Последовательность сетей с автоматическим размещением блоков и связей. Блоки не перемещаются произвольно по рабочему полю.
CFC	Свободное размещение блоков на рабочем поле с сохранением их координат. Сети используются для группирования и определения порядка независимых фрагментов.

Выбор языка не задаёт период выполнения программы. Разовое или периодическое выполнение настраивается в задаче проекта. Функция выполняется при её вызове другой программой или функцией.

2 Открытие редактора и рабочее пространство

Для работы необходимо создать программный узел нужного типа и открыть его редактор из дерева проекта. Диаграмма отображается на отдельной вкладке рабочей области. Для нескольких узлов могут быть открыты отдельные вкладки.

Область	Назначение
Рабочее поле	Размещение блоков, соединение входов и выходов, отображение порядка выполнения и диагностических отметок.
Сети	Выбор, переименование и изменение порядка сетей.
Переменные	Просмотр доступных переменных, создание и изменение локальных объявлений, перетаскивание переменных в схему.
Таблица переменных	Обзор имён, типов, областей видимости, начальных значений, значений отладчика и использований.
Параметры	Настройка входных параметров функции и привязка их к входам блоков. Для программы эта панель отсутствует.
Сгенерированный код	Дополнительное текстовое представление текущей схемы, доступное только для просмотра.
Панель масштаба	Изменение масштаба, восстановление 100 % и показ выбранной сети.

Панели **«Таблица переменных»** и **«Сгенерированный код»** первоначально скрыты. Их отображение переключается средствами управления панелями окна редактора. Размеры панелей можно изменять перетаскиванием границ.

Для узлов подключённой библиотеки используется режим только для чтения. Доступность просмотра не означает возможности сохранить изменения библиотечного узла в текущий проект.

3 Сети и порядок выполнения

3.1 Создание и организация сетей

Сеть объединяет часть алгоритма. В панели **«Сети»** доступны команды контекстного меню:

- **«Добавить сеть»**;
- **«Удалить сеть»**;
- **«Переместить раньше»**;
- **«Переместить позже»**.

Названия редактируются в списке сетей. Порядок также изменяется перетаскиванием строк. Выбор строки делает сеть текущей и переводит рабочее поле к её расположению.

В FBD на свободной области схемы предусмотрена команда **«Добавить сеть ниже»**. Новые блоки добавляются с учётом выбранной сети и места добавления. Перед добавлением следует проверить текущую сеть.

Удаление допускается для пустой сети. Если она содержит блоки, необходимо сначала удалить их либо перенести в другую сеть командой блока **«Переместить в сеть»**.

3.2 Зависимости и явный порядок

Порядок выполнения учитывает связи данных: блок, формирующий значение, выполняется раньше блока, использующего этот результат. Для независимых ветвей учитываются порядок сетей и заданный порядок блоков внутри сети. Номер выполнения отображается на блоке.

Для изменения порядка выделенного блока используются команды **«Выполнить раньше»** и **«Выполнить позже»**. По умолчанию им назначены клавиши **−** и **+** при фокусе на рабочем поле. Сочетания могут быть изменены в настройках DevEnv.

Команда недоступна, если перестановка нарушает зависимость по данным либо блок уже занимает крайнее положение. Для нескольких выделенных блоков перестановка выполняется только тогда, когда она допустима для всей группы.

В CFC изменение координат блока служит организации изображения и само по себе не заменяет команды изменения порядка. При наличии нескольких независимых записей или вызовов следует проверять отображаемые номера выполнения.

3.3 Передача данных между сетями

Прямые связи между блоками разных сетей в текущей версии не создаются. Для передачи результата используется переменная: в одной сети значение записывается, в другой — читается. Сеть записи должна выполняться раньше сети чтения, если требуется результат текущего вызова.

Замкнутые цепочки связей также не поддерживаются ни в FBD, ни в CFC. Для алгоритмов с состоянием используются переменные и функциональные блоки с явно определённой последовательностью чтения и записи.

4 Добавление блоков

4.1 Выбор из каталога

Для добавления следует открыть контекстное меню свободного участка рабочего поля. Каталог содержит категории и строку **«Фильтр»**. Поиск выполняется по названию и доступным сведениям о блоке. Результат выбирается мышью либо клавишами ↑, ↓ и **Enter**.

Основные группы элементов:

Группа	Применение
Бинарный оператор	Арифметические, логические и битовые операции, в том числе ADD, SUB, MUL и DIV.
Унарный оператор	Одноместные операции, в том числе NOT.
Сравнение	Проверка равенства, неравенства и отношений между значениями.
IEC: Математика	Доступные математические функции.
IEC: Преобразование типов	Явное преобразование между поддерживаемыми типами.
IEC: Битовые операции	Сдвиги, выбор значений и другие операции группы.
IEC: Строки	Доступные операции над строками.
IEC: Функциональные блоки	Таймеры, счётчики, триггеры и другие доступные стандартные ФБ.
Приложение и категории вызовов	Доступные функции и методы проекта.
Функциональные блоки (ST), экземпляры ФБ	Создание вызова доступного ФБ или обращение к существующему экземпляру.
Переменная	Запись результата в выбранную переменную.
Возврат	Блок Return в редакторе функции.

Фактический состав каталога зависит от приложения, подключённых библиотек, доступных объявлений и контекста добавления. При наличии перегрузок следует учитывать тип результата и аргументов выбранной функции.

4.2 Добавление из незавершённой связи

Блок можно добавить в процессе соединения:

1. Начать протягивание связи от входа или выхода блока.
2. Отпустить её на свободном участке схемы.
3. Выбрать элемент в открывшемся каталоге.

В таком каталоге учитываются направление соединения и тип исходного контакта. Новый блок добавляется с попыткой автоматического подключения к подходящему контакту.

Если связь протягивается от входа, каталог также может предложить подходящую переменную или параметр. В этом случае создаётся привязка входа к значению, без отдельного блока чтения.

4.3 Количество входов

Для поддерживаемых блоков в контекстном меню доступны **«Добавить вход»** и **«Удалить вход»**. Например, число входов может изменяться у операций сложения, умножения и некоторых логических операций.

Для функции допустимые варианты зависят от имеющихся объявлений. Удаление последнего входа или группы входов недоступно, пока к удаляемым контактам подключены связи.

5 Входы, выходы и связи

5.1 Создание связи

Входы расположены с принимающей стороны блока, выходы — с передающей. Для соединения следует протянуть линию от выхода одного блока к входу другого. Один выход может использоваться несколькими входами.

При настройке необходимо учитывать тип контакта, назначение аргумента и направление передачи. Цвет контактов и связей помогает различать типы. Тип также отображается в подсказках и подписях; цвет не заменяет проверку совместимости.

Тип некоторых операторов определяется по подключённым данным. Если необходимое преобразование не поддерживается непосредственно, следует добавить блок явного преобразования типа.

Связь может не создаваться при несовместимости контактов, попытке соединения разных сетей или образовании замкнутой цепочки. Перемещение блоков в CFC не устраняет такие ограничения.

5.2 Значение неподключённого входа

Вход может получать значение из связи, константы, локальной или глобальной переменной, а в функции — из параметра.

Для редактирования подписи свободного входа используются двойной щелчок, **F2** или **Enter**. В поле можно задать константу либо выбрать имя переменной из подсказки. Например, для числового входа задаётся `2.0`, для логического — `true` или `false`.

Для известных скалярных типов предусмотрены начальные значения, соответствующие нулю, `false` или пустой строке. Перед использованием входа следует проверить фактически показанное значение. Если тип ещё не определён или для него требуется явное назначение, вход может обозначаться **???**.

Подпись отсутствующей переменной отмечается как ошибочная. После переименования или удаления внешнего объявления необходимо проверить связанные входы и сообщения диагностики.

5.3 Инверсия

Для контакта типа `bool` предусмотрена инверсия. В его контекстном меню используются команды **«Инвертировать»** и **«Убрать инверсию»** с именем контакта.

Инверсия изменяет логическое значение на противоположное в соответствующей точке схемы. Её следует учитывать отдельно от наличия блока NOT и не применять повторно без необходимости.

5.4 Удаление

Выделенную связь можно удалить клавишей **Delete** или командой **«Удалить связь»** её контекстного меню. Удаление блока удаляет и связанные с ним соединения. После удаления следует проверить свободные входы оставшихся блоков.

6 Переменные

6.1 Локальные переменные

Локальная переменная создаётся в панели **«Переменные»** командой **«Добавить переменную»** либо кнопкой **«+ Добавить»** в таблице переменных. В диалоге задаются:

- **«Имя»**;
- **«Тип»**;
- **«Начальное значение»**.

В списке предложены, в частности, `bool`, `int`, `int32`, `int64`, `uint32`, `uint64`, `float`, `double` и `string`. Поле типа допускает ввод другого типа, доступного приложению. Корректность объявления проверяется при подготовке программы.

Локальные переменные создаются заново при каждом вызове программы или функции; начальное значение применяется при каждом таком вызове. Они не предназначены для хранения результата между циклами задачи.

Для изменения используются **«Изменить…»** или редактирование соответствующих ячеек таблицы. Удаление переменной, которая ещё используется схемой, блокируется с сообщением **«Переменная используется»**. Сначала необходимо убрать или переназначить её использования.

6.2 Глобальные переменные

Панель также показывает доступные глобальные переменные приложения, в том числе объявления из текстовых программных элементов. Для глобального элемента предусмотрена команда **«Перейти к объявлению»**, открывающая место его определения.

Глобальные объявления изменяются в содержащем их программном элементе. Локальные переменные и параметры функции могут скрывать одноимённые глобальные объявления. При совпадении имён следует проверить область видимости и выбранное использование.

Для хранения состояния между вызовами можно использовать глобальную переменную или соответствующий экземпляр функционального блока. Это не означает автоматического сохранения значения при остановке среды исполнения или отключении питания; такая задача настраивается отдельно.

6.3 Чтение и запись

Переменную можно перетащить:

- **на вход блока** — для чтения её значения;
- **на свободный участок схемы** — для добавления блока записи, если переменная допускает запись.

Чтение отображается подписью у входа. Запись выполняется отдельным блоком **`set`**, к которому подаётся записываемое значение. Константные и другие недоступные для записи элементы не используются как цель записи.

Если в нескольких местах схемы записывается одна переменная, необходимо явно проверить порядок этих операций. Для передачи значения из одной сети в другую используется та же переменная с согласованным порядком сетей.

6.4 Таблица переменных

Таблица содержит столбцы **«Имя»**, **«Тип»**, **«Область»**, **«Начальное значение»**, **«Значение отладчика»** и **«Использований»**.

Для отбора предусмотрены поле поиска и варианты **«Все»**, **«Локальные»**, **«Глобальные»**, **«Неиспользуемые»**. Команда **«Перейти к использованиям»** позволяет выбрать блок, в котором встречается переменная, и перейти к нему на схеме.

В таблице используются **Enter/F2** для редактирования, **Tab/Shift+Tab** для перехода между ячейками, **Esc** для отмены ввода, **Insert** для добавления, **Delete** для удаления и **Ctrl+F** для поиска.

7 Функции, параметры и возврат результата

У функции входные параметры настраиваются в панели **«Параметры»** командами **«Добавить параметр»**, **«Изменить…»** и **«Удалить параметр»**. Для параметра задаются имя и тип.

Параметр перетаскивается на вход блока и используется как его значение. Перетаскивание параметра на свободное место не создаёт блока записи. Удаление используемого параметра требует предварительного удаления его привязок.

Свойство функции **«Тип результата»** задаётся в свойствах узла. Для возврата используется блок **Return** из категории **«Возврат»**. В функции с результатом к входу **`value`** подключается вычисленное значение требуемого типа.

Выполнение Return завершает текущий вызов функции. Поэтому положение этого блока в порядке выполнения должно соответствовать логике алгоритма. При условном выполнении необходимо предусмотреть возврат результата для всех требуемых ветвей.

Функции и методы добавляются из доступного каталога. Для метода следует учитывать требуемый объект, а для вызова функционального блока — выбранный экземпляр. Два вызова одного существующего экземпляра используют его общее состояние.

8 Функциональные блоки и EN/ENO

8.1 Функциональные блоки

Каталог содержит стандартные и доступные в приложении ФБ. К типовым стандартным элементам относятся **RS**, **SR**, **R_TRIG**, **F_TRIG**, **TON**, **TOF**, **TP**, а также варианты счётчиков **CTU**, **CTD** и **CTUD**.

Входы и выходы ФБ определяются его назначением. Например, у таймера используются **IN**, **PT**, **Q** и **ET**. Для стандартных таймеров значения времени PT и ET представлены в миллисекундах.

При использовании блока с состоянием следует учитывать периодичность его вызова и назначение входов сброса или разрешения. Простое размещение таймера на схеме не задаёт отдельную задачу его исполнения.

8.2 Разрешение выполнения

Команды контекстного меню **«Добавить пины EN/ENO»** и **«Убрать пины EN/ENO»** включают или убирают дополнительные логические контакты.

Контакт	Назначение
EN	Разрешение выполнения блока. При `false` операция пропускается.
ENO	Признак прохождения разрешённого выполнения: при обычном выполнении с EN = `true` устанавливается `true`, при пропуске — `false`. Инверсия контактов учитывается отдельно.

EN/ENO позволяют связать порядок условных действий, например разрешать запись результата только после выполнения предыдущего блока. ENO не следует считать универсальным сообщением об отсутствии ошибок всего алгоритма.

При EN = `false` значение обычного выхода не является результатом нового вычисления блока. Сохранение предыдущего значения следует задавать логикой переменных или ФБ, а не предполагать по наличию EN.

9 Размещение, выделение и копирование

В FBD компоновка выполняется автоматически. Для изменения структуры используются связи, перенос между сетями и команды порядка. В CFC блоки перетаскиваются мышью; рамку сети можно использовать для перемещения содержащихся в ней блоков.

Допускается выделение нескольких элементов. При нажатии правой кнопки на уже выделенном блоке контекстное меню применяется к группе. Для добавления элемента к выделению используется **Ctrl**; для рамочного выделения — **Shift** с выделением области мышью.

Копирование выполняется командой **«Копировать»** или **Ctrl+C**, вставка — **«Вставить»** либо **Ctrl+V**. При вставке с клавиатуры используется положение указателя в рабочем поле, а если он находится вне него — центр видимой области.

Копируются выделенные блоки и связи между ними. Все вставленные блоки помещаются в одну текущую сеть; исходное распределение по сетям не восстанавливается. При переносе фрагмента в другой узел следует проверить доступность переменных, параметров и вызываемых функций.

Изменения диаграммы можно отменять и повторять общими командами DevEnv; стандартные сочетания — **Ctrl+Z** и **Ctrl+Y**.

Для масштаба предусмотрены кнопки **«−»**, **«100 %»**, **«+»** и **«Показать сеть»**. В FBD колесо мыши прокручивает схему, а **Ctrl + колесо** изменяет масштаб. В CFC колесо используется для масштабирования рабочего поля.

10 Проверка и сохранение

10.1 Диагностика схемы

Проверка выполняется общими средствами подготовки и проверки проекта. Команде подготовки конфигурации по умолчанию соответствует **Ctrl+B**.

Ошибки и предупреждения отображаются в окне **«Ошибки»**. Сообщения, относящиеся к конкретному блоку или связи, дополнительно обозначаются на схеме; подробности доступны в подсказке. Сообщения об объявлениях или общих параметрах могут оставаться только в общем списке.

При исправлении следует проверить типы входов и выходов, значения свободных входов, существование переменных и параметров, выбор перегрузки функции и возврат результата.

Панель **«Сгенерированный код»** позволяет дополнительно просмотреть текстовое представление текущего алгоритма. Изменение схемы выполняется в графическом редакторе; эта панель доступна только для чтения.

10.2 Сохранение

Изменения схемы применяются непосредственно к открытому проекту. У графического редактора нет отдельного этапа принятия текста, аналогичного кнопке сохранения в текстовом редакторе.

Для записи на диск используется **«Файл» → «Сохранить проект»**, по умолчанию **Ctrl+Shift+S**. Закрытие вкладки графического редактора не является отменой правок и не предполагает отдельного запроса о сохранении диаграммы. Для отмены изменения следует использовать команду отмены до сохранения требуемого состояния.

Сохранение проекта не загружает новую конфигурацию на устройство. После проверки необходимо отдельно выполнить подготовку конфигурации, загрузку и запуск принятым для проекта способом.

Для выполнения программы в составе приложения её необходимо включить в свойство **«Узлы»** (`ExecutableNodes`) подходящей задачи. Функция должна вызываться из другого программного элемента.

11 Отладка

На блоке можно установить точку останова командой **«Установить точку останова»**, снять её командой **«Снять точку останова»** либо использовать **F9** при выделенном блоке. Сочетание настраивается в DevEnv.

При остановке отладчика текущий блок выделяется на схеме. Для возврата к нему после просмотра другого участка предусмотрена команда **«Показать текущий блок»**.

Доступные значения отображаются у контактов и в панелях переменных. Щелчок по показанному значению контакта открывает инспектор, в котором можно просмотреть содержимое переменной или объекта. Если значение недоступно в текущем контексте, отображение может отсутствовать либо содержать **«нет данных»**.

Изменения диаграммы, влияющие на исполняемый алгоритм, снимают точки останова этого узла. После таких правок необходимо повторно установить требуемые точки. Простое перемещение блока CFC без изменения алгоритма само по себе не требует их снятия.

Во время отладки часть команд изменения переменных и параметров недоступна. Доступность конкретных действий зависит также от режима только для чтения.

Порядок запуска и работы с отладчиком приведён в описании отладки проекта.

12 Пример: вычисление результата функции

Требуется реализовать функцию `ScaleSum(a, b)`, возвращающую `(a + b) × 2`.

1. Создать **«Функция (FBD)»** с именем `ScaleSum` и указать **«Тип результата» = `float`**.
2. Открыть редактор и в панели «Параметры» создать `a` и `b` типа `float`.
3. Добавить блок **ADD(+)**. Перетащить параметры `a` и `b` на его два входа.
4. От выхода ADD протянуть связь на свободное место и выбрать **MUL(*)**.
5. Для второго входа MUL задать константу `2.0`.
6. Добавить блок **Return** и соединить его вход `value` с выходом MUL.
7. Проверить типы и порядок выполнения: ADD, затем MUL, затем Return.
8. Сохранить и проверить проект. При вызове с `a = 2.0` и `b = 3.0` ожидаемый результат равен `10.0`.

Та же последовательность действий применяется к **«Функция (CFC)»**. В CFC расположение блоков выбирается вручную; порядок вычисления по связям остаётся тем же.

13 Типовые затруднения

Ситуация	Что следует проверить
Блок FBD не перемещается мышью	В FBD применяется автоматическая компоновка; использовать изменение структуры и порядка.
Связь не создаётся	Типы и направления контактов, принадлежность одной сети, отсутствие замкнутой цепочки.
Вход показывает ???	Определённость типа и назначение связи, переменной, параметра или константы.

Переменная отмечена как отсутствующая	Наличие объявления, имя, область видимости и актуальность глобальных объявлений.
Не удаётся удалить переменную или параметр	Использования в блоках и привязках входов.
Нельзя удалить сеть	Наличие блоков; сначала перенести или удалить их.
Команда «Выполнить раньше» недоступна	Зависимости по данным, крайнее положение и допустимость перестановки всей группы.
Значение не сохраняется между вызовами	Использование локальной переменной вместо глобального состояния или экземпляра ФБ.
Таймер работает не так, как ожидается	Вызов программы задачей, период выполнения, единицы времени и входы таймера.
После копирования нет ожидаемой привязки	Объявления в принимающем узле, параметры функции и текущую сеть вставки.
Точка останова исчезла после правки	Изменение алгоритма; повторно установить точку перед отладкой.
На схеме нет отметки ошибки из списка	Сообщение может относиться к объявлению или общим параметрам узла.
Закрытие вкладки не отменило правки	Диаграмма изменяет открытый проект; использовать отмену и сохранение проекта.

14 Рекомендуемый порядок работы

1. Выбрать FBD с автоматической компоновкой или CFC со свободным размещением.
2. Определить назначение программного узла, источники данных и требуемое состояние между вызовами.
3. Создать сети, локальные переменные и параметры функции.
4. Добавить блоки, назначить значения входов и соединения.
5. Проверить порядок независимых ветвей, записи переменных и условного выполнения.
6. Выполнить проверку проекта и устранить диагностику.
7. Сохранить проект, настроить вызов программы или функции и проверить результат при отладке.