



26.51.70.190

УТВЕРЖДЕН

АДИГ.421457.005 РЭ8-ЛУ

КОМПЛЕКСЫ ПРОГРАММНО-ТЕХНИЧЕСКИЕ «СУРА»

Руководство по эксплуатации

Средства редактирования технологических программ

АДИГ.421457.005 РЭ8

Вер.3.2

Листов 35

Содержание

1	НАЗНАЧЕНИЕ ПОЛИСА	5
1.1	ТЕРМИНОЛОГИЯ.....	5
1.2	ДОПОЛНИТЕЛЬНЫЕ МАТЕРИАЛЫ ДЛЯ РАБОТЫ С ПОЛИСОМ.....	6
2	ОБЩИЕ СВЕДЕНИЯ О ПОЛИСЕ	7
2.1	ПРИСТУПАЯ К РАБОТЕ	7
2.2	КАК СТРУКТУРИРОВАТЬ ПРОЕКТ.....	7
2.3	ЧТО ДЕЛАЕТ ПОЛИС	7
2.4	ЧТО ЕЩЕ МОЖЕТ ДЕЛАТЬ ПОЛИС.....	8
2.4.1	Компиляция и загрузка	8
2.4.2	Обзор выполнения технологической программы	8
2.4.3	Диагностика и настройка.....	8
2.4.4	Блокировка.....	8
2.4.5	Перестановка	9
3	РАБОТА С ПОЛИСОМ.....	10
3.1	ГЛАВНОЕ ОКНО ПОЛИСА.....	10
3.2	ШКАЛА ИНСТРУМЕНТОВ	10
3.3	МЕНЮ ВИД	13
3.3.1	Состав меню вид.....	13
3.3.2	Результаты поиска.....	13
3.3.3	Обзор проекта	13
3.3.4	Объекты.....	15
3.3.5	Точки останова	16
3.3.6	Выбор алгоритма.....	16
3.3.7	Стек вызовов.....	17
3.3.8	Порядок выполнения	17
3.3.9	Обзор 1, Обзор 2, Обзор 3	17
3.3.10	Локальный обзор.....	17
3.3.11	Список ошибок	17
3.4	СВОЙСТВА ЗАДАЧ	17
3.4.1	Список задач	17
3.4.2	Поле задачи	18
3.4.3	Размещение алгоблоков.....	19
3.4.4	Реквизиты алгоблоков.....	19
3.4.5	Входы-выходы алгоблока.....	21
3.4.6	Копирование алгоблоков.....	21
3.4.7	Свойства входов алгоблоков	21
3.4.8	Общий случай установки констант	22
3.4.9	Выбор типа данных.....	22
3.5	КОНФИГУРИРОВАНИЕ АЛГОБЛОКОВ	23
3.5.1	Техника конфигурирования	23
3.5.2	Ссылки.....	23
3.5.3	Дополнительные возможности конфигурирования	24
3.5.4	Пересечение линий связи	24
3.6	ПЕРЕМЕЩЕНИЕ И ПЕРЕСТАНОВКА АЛГОБЛОКОВ	24
3.6.1	Возможности перемещения.....	24
3.6.2	Наложение алгоблоков и ссылок	24
3.6.3	Перестановка алгоблоков	25
3.7	КОММЕНТАРИИ.....	25
3.7.1	Виды комментариев	25
3.7.2	Комментарии к задаче.....	25
3.7.3	Другие виды комментариев.....	25
3.8	ПРИВЯЗКА	26
3.8.1	Назначение привязки	26

3.8.2	Отображение привязки	26
3.8.3	Технология привязки	26
3.9	ПРАВА ДОСТУПА И КОЛЛЕКТИВНАЯ РАБОТА С ПОЛИСОМ.....	27
3.9.1	Права доступа	27
3.9.2	Коллективная работа.....	27
4	ОТЛАДКА ТЕХПРОГРАММ	29
4.1	КОМПИЛЯЦИЯ И ЗАГРУЗКА ТЕХПРОГРАММЫ	29
4.2	ОБЗОР	30
4.2.1	Возможности обзора	30
4.2.2	Доступ к задаче во время обзора	30
5	РЕКОМЕНДАЦИИ ПО ПОДГОТОВКЕ ТЕХПРОГРАММ	32
5.1	ИНФОРМАЦИОННЫЕ РЕСУРСЫ	32
5.2	ВВОД/ВЫВОД ИНФОРМАЦИИ ЧЕРЕЗ МОДУЛИ УСО	33
5.3	МЕЖКОНТРОЛЛЕРНЫЙ ОБМЕН	33
5.4	ОБЪЕКТНЫЕ АЛГОРИТМЫ	33
5.4.1	Подробнее об объектах	33
5.4.2	Общие свойства объектных алгоритмов	33
5.5	ВЫЧИСЛИТЕЛЬНЫЕ АЛГОРИТМЫ	34
5.5.1	Состав вычислительных алгоритмов.....	34
5.6	АРХИВИРОВАНИЕ СИГНАЛОВ.....	34
5.6.1	Варианты архивирования	34
5.6.2	Периодическое архивирование сигналов.....	34
5.6.3	Архивирование аналоговых сигналов по апертуре.....	34
5.6.4	Объектное архивирование	34
	ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ	35

Содержание данного документа может изменяться без предварительного уведомления.

В связи с непрерывной работой по улучшению характеристик изделия программное обеспечение может также меняться. Тем не менее, содержание документации регулярно отслеживается и все корректировки вносятся в последующие издания.

Настоящее руководство содержит правила применения набора инструментов **Полиса** при создании проекта АСУ ТП на базе ПТК «СУРА».

Инструменты реализованы средствами **Станции проектирования** на базе Рабочей станции с программными приложениями САПР из состава ПО верхнего уровня ПТК «СУРА» (менеджер приложений «СФЕРА», в дальнейшем по тексту – **СФЕРА**). Инструменты предназначены для технологического программирования проекта АСУ ТП на базе ПТК «СУРА» (в дальнейшем по тексту - СУРА).

Перед началом работы с **Полисом** необходимо также ознакомиться с документом «Комплексы программно-технические «СУРА». Руководство по эксплуатации. Администрирование проектов АСУ ТП. АДИГ.421457.005 РЭ1».

1 Назначение Полиса

Система технологического программирования **Полис** (в дальнейшем по тексту – **Полис**) предназначена для подготовки пользовательских технологических программ контроллеров (фирменное название контроллера - Эликонт), входящих в программно-технический комплекс «СУРА» (в дальнейшем по тексту - СУРА).

Полис - это программный пакет, входящий в группу САПР фирменного программного обеспечения СУРЫ. Все фирменное программное обеспечение СУРЫ объединено менеджером приложений **СФЕРА**, из которой загружаются различные программные компоненты, в том числе и **Полис**.

В **СФЕРЕ** Полис входит в раздел САПР.

При работе с **Полисом** используется библиотека алгоритмов «Библио», описание которой «встроено» в **Полис** и может быть вызвано на экран монитора станции проектирования.

1.1 Терминология

Ниже перечислены основные термины, используемые в тексте и специфичные для **Полиса**.

База данных (БД) – совокупность всех проектных данных об АСУ ТП, формируемая пользователем и необходимая для работы СУРЫ. В состав БД входят и программы, созданные средствами Полиса.

Базис – система управления БД проекта АСУ ТП. С помощью **Базиса** формируется полный перечень объектов, задействованных в АСУ ТП, причем в БД вносится лишь запись о самом объекте, - все сопутствующие признаки «прописываются» в базе данных автоматически. Входит в состав папки САПР менеджера **СФЕРА**.

Администратор БД - программа, обеспечивающая работу со всей БД проекта АСУ ТП. Входит в состав папки Администрирование проекта менеджера **СФЕРА**.

Технологическая программа – совокупность формируемых пользователем данных (задач, алгоблоков, конфигурационных связей между ними и т.д.), загружаемых в Эликонты.

Информационный объект (в дальнейшем, там, где это не приводит к неоднозначности – просто **объект**) – совокупность информации, характеризующая свойства информационного элемента АСУ ТП как единого целого. Информационные объекты могут отражать свойства реальных (датчики, механизмы и т.п.) или виртуальных (регуляторы, шаговые программы и т.п.) элементов.

Задача – совокупность алгоблоков, имеющая имя и размещаемая в одном непрерывном поле задачи. Обычно задача Полиса соответствует сравнительно автономной технологической задаче управления.

Подзадача – задача, иерархически входящая в другую задачу.

Поле задачи – непрерывная область размещения алгоблоков, которая, в частности, может быть больше экрана компьютера. В последнем случае для просмотра алгоблоков, входящих в задачу, используется прокрутка поля задачи.

Алгоблок – алгоритм, введенный в поле задачи.

Алгоритм – созданная разработчиками СУРЫ фиксированная совокупность функций, имеющая имя и входящая в библиотеку алгоритмов Эликонта.

Объектный алгоритм – алгоритм, формирующий совокупную информацию об информационном объекте.

Конфигурация – система связей между алгоблоками.

Свободный вход – не законфигурированный (т.е. ни с чем не связанный) вход алгоблока, на котором задается константа.

Константа – значение сигнала на свободном входе.

Связанный вход – законфигурированный вход алгоблока, т.е. вход, соединенный с каким-либо выходом какого-либо алгоблока.

Ссылка – мнемоническое обозначение адреса источника или приемника, заменяющее линию связи между алгоблоками.

Перестановка – изменение порядка выполнения задач в пределах контроллера или алгоритмов в пределах задачи.

Привязка – связь между алгоблоками или отдельными его элементами (каналами, входами, выходами) с объектами базы данных.

Компиляция – процесс преобразования подготовленной в **Полисе** технологической программы в специальный формат, используемый для загрузки этой программы в Эликонт.

Обзор – наблюдение средствами **Полиса** за реальными сигналами, формируемыми контроллером на входах и выходах алгоблоков. Обзор используется для отладки подготовленной технологической программы.

1.2 Дополнительные материалы для работы с Полисом

Для эффективной работы с **Полисом** необходимо ознакомиться с общей концепцией СУРЫ, составом его компонентов и их общими свойствами. Для этого необходимо изучить следующие документы:

- «Комплексы программно-технические «СУРА». Руководство по эксплуатации. Администрирование проектов АСУ ТП. АДИГ.421457.005 РЭ1»;
- «Комплексы программно-технические «СУРА». Руководство по эксплуатации. Средства управления технологической базой данных. АДИГ.421457.005 РЭ6».

Перед тем, как приступить к подготовке технологической программы в **Полисе**, следует продумать техническую и функциональную структуру АСУ ТП, особенно в той ее части, которая касается контроллеров.

2 Общие сведения о Полисе

2.1 Приступая к работе

Перед тем, как начать работать с **Полисом**, необходимо:

– проинсталлировать на компьютере требуемые компоненты **СФЕРЫ**. Если база данных (БД) хранится на отдельной рабочей станции (сервере БД), то компьютер должен быть объединен в общую сеть с сервером БД и должен быть соответствующим образом «прописан» в **Администраторе БД**;

– на один из компьютеров в общей сети установить ключ защиты (обычно это сервер БД), соответствующий приобретенной лицензии на использование СУРБ. Если компьютер не включен в сеть, этот ключ устанавливается непосредственно на компьютер. Если работа ведется в автономном режиме (без контроллеров), ключ не требуется;

– Средствами **Администратора БД** ввести описание всех рабочих станций, контроллеров и коммутаторов, относящихся к данному проекту (т.е. входящих в БД проекта), а также всех рабочих станций, на которых ведется работа с СУРОЙ;

– через **Базис** ввести описание всех информационных объектов (датчиков, механизмов и т.д.), относящихся к данному проекту. Это описание необходимо для привязки алгоблоков технологической программы к объектам БД. Начать работу с **Полисом** можно и без заполнения БД объектами (или с неполным заполнением БД), но тогда к **Полису** нужно будет вернуться после подготовки БД с тем, чтобы выполнить указанную привязку.

2.2 Как структурировать проект

АСУ ТП может содержать сотни и даже тысячи различных компонент, и чтобы в них можно было легко ориентироваться, им необходимо придать определенную иерархическую структуру. В СУРЕ можно создавать в большой степени произвольную структуру, но лучше всего, если она будет соответствовать внутренней связи отдельных компонент технологического процесса.

Все логические элементы АСУ ТП, связанные с контроллерами, структурируются в СУРЕ следующим образом:

Общие управляющие функции распределяются по контроллерам. Каждому контроллеру присваивается технологическое имя и сетевой номер.

Функции каждого контроллера разбиваются на задачи. Задачам даются технологические имена.

Каждая задача может (но не обязательно) разбиваться на подзадачи, каждая подзадача – на свои подзадачи, те – на еще более мелкие подзадачи и т.д. Иерархия и глубина вложенности задач и подзадач определяется пользователем.

Состав контроллеров задается в **Администраторе БД**, а состав и структура задач и подзадач – непосредственно в **Полисе**.

Свойства задач и подзадач в большинстве своем совпадают, поэтому в дальнейшем при их описании будет использоваться общий термин – задача. Термин подзадача будет использоваться только там, где нужно подчеркнуть соподчиненность этих понятий.

2.3 Что делает Полис

После того, как определена структура проекта, можно приступать к подготовке технологических программ. В процессе этой подготовки в **Полисе** выполняются следующие действия:

- выбирается требуемый контроллер;
- для выбранного контроллера создается (возможно иерархический) перечень задач. Выбирается требуемая задача, - на экране открывается поле задачи;
- из библиотеки алгоритмов Эликонта выбираются требуемые алгоритмы, и они расставляются наглядным образом в поле задачи по указанию пользователя;

- между входами и выходами отдельных алгоритмов проводятся требуемые связи (эта процедура называется конфигурированием);
- на свободных (не связанных конфигурацией) входах алгоритмов задаются требуемые константы;
- некоторые (но не все) элементы полученной алгоритмической структуры (отдельные алгоритмы, отдельные каналы алгоритма, отдельные входы или выходы) привязываются к объектам БД. Такая привязка необходима для отображения информации, формируемой контроллерами, на рабочих станциях ИВК.

Таким образом, технологическая программа готовится для конкретного выбранного контроллера.

2.4 Что еще может делать Полис

2.4.1 Компиляция и загрузка

После того, как контроллер с помощью контроллера домена подключился к информационному пространству, подготовленную для него средствами **Полиса** пользовательскую технологическую программу можно откомпилировать и в контроллер загрузить.

В процессе компиляции технологическая программа, имеющая вид функциональной схемы, преобразуется в специальный формат, используемый для загрузки этой программы в контроллер.

В одной процедуре загрузки программа загружается в один выбранный контроллер, при этом загружаются все задачи технологической программы.

2.4.2 Обзор выполнения технологической программы

Редко когда подготовленная технологическая программа (как и любая другая программа) не содержит ошибок. Полис предоставляет средства для отладки, загруженной в Эликонт программы, при этом могут использоваться модели объекта или, в более простом случае, имитаторы сигналов.

В специальном режиме, называемым **Обзор**, непосредственно в **Полисе** можно наблюдать реальные значения сигналов на входах и выходах алгоблоков. Изменяя сигналы с помощью имитаторов или изменяя значения констант на входах алгоритмов, можно наблюдать реакцию на выходах выбранных алгоритмов и делать соответствующие выводы.

Имеется еще одна полезная возможность – прямо из **Полиса** можно послать Эликонту так называемую объектную команду, имитирующую действие оператора. Например, можно перевести регулятор на ручной или автоматический режим, включить или отключить двигатель, открыть или закрыть задвижку и т.д. Результат посланной команды можно «вживую» наблюдать по изменению сигналов на входах-выходах, выбранных алгоблоков или на экране операторской станции.

Очевидно, что обзор будет работать, только если рабочая станция с **Полисом** подключена к общей сети, к которой, в свою очередь, подключен соответствующий контроллер с загруженной технологической программой.

2.4.3 Диагностика и настройка

Для выбранного контроллера **Полис** позволяет посмотреть использованные технологической программой ресурсы, запросить приборные параметры контроллера и сравнить их с данными БД, посмотреть содержимое буфера ошибок, как результат работы системы встроенной самодиагностики.

2.4.4 Блокировка

При коллективной работе над проектом могут возникнуть информационные конфликты между различными пользователями. Например, два пользователя могут открыть одну и ту же задачу, при этом один из них будет «вмешиваться» в работу другого. Другой пример – один пользователь начал загружать программу в контроллер, в то время как другой продолжает ее редактировать.

Чтобы исключить подобные конфликты, которые могут нарушить целостность базы данных и привести к некорректной работе программы, в Полисе предусмотрен механизм блокировок,

который, в зависимости от ситуации, распространяется на весь контроллер или отдельные входящие в него задачи.

2.4.5 Перестановка

Для работы в реальном времени важен не только состав и конфигурация задач и алгоритмов, но и последовательность их выполнения. После того, как подготовленная технологическая программа будет загружена в контроллер, сначала выполнятся все алгоритмы задачи, стоящей в **Полисе** выше по списку, затем все алгоритмы первой по списку подзадачи, входящей в данную задачу и т.д. В пределах каждой задачи алгоблоки выполняются в той последовательности, в которой они вводились в **Полисе**.

Однако, не всегда порядок ввода задач и алгоблоков отражает желаемую последовательность их выполнения. Здесь на помощь приходит процедура, называемая **перестановкой**. Она позволяет переназначить порядок выполнения уже введенных задач и алгоблоков, и, тем самым, избежать утомительной перекройки технологической программы.

3 Работа с Полисом

3.1 Главное окно Полиса

После запуска приложения открывается главное окно **Полиса** (Рисунок 1).

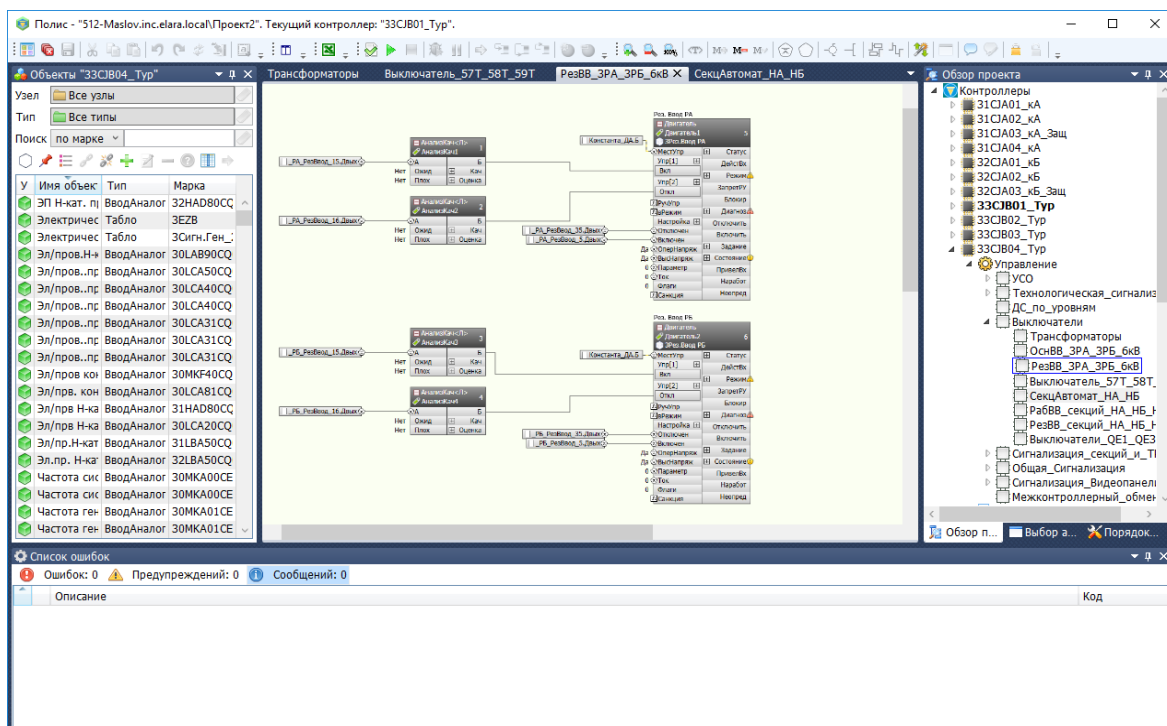


Рисунок 1 – Главное окно Полиса

На рисунке представлена уже заданная конфигурация окон, которая задаётся при помощи меню **Вид**. В шапке окна указывается название приложения, путь (локальный или сетевой) к БД и имя выбранного контроллера. Основное место главного окна **Полиса** занимает поле задачи с алгоблоками. Одновременно можно открыть несколько задач и расположить в различных вариациях при помощи панели привязки (Рисунок 2).




Рисунок 2 – Панель привязки

Центральное расположение основное и единственное для первой открытой задачи. Остальные задачи можно располагать в удобном для пользователя положении. По умолчанию все задачи открываются с центральной привязкой в виде вкладок в главном окне (рисунок 1).


Окна меню вид привязываются по аналогичному принципу. По умолчанию справа открываются окна: **Обзор проекта**, **Выбор алгоритма** и **Порядок выполнения**. Слева открывается окно **объекты**. Снизу все остальные окна меню **Вид**.


3.2 Шкала инструментов


Заккрыть все вкладки . Закрываются все открытые ранее вкладки в поле задач;


Сохранить (Ctrl+S). Сохранить макрос на текстовом языке ST;


Вырезать (Ctrl+X) . Вырезать один или несколько выделенных алгоблоков в поле задач;

Копировать (Ctrl+C)  . Копировать один или несколько алгоблоков в поле задач, в другую задачу текущего контроллера, в задачу другого контроллера;


Вставить (Ctrl+V)  . Вставка одного или нескольких алгоблоков в поле задач, в другую задачу текущего контроллера, в задачу другого контроллера;


Отменить (Ctrl+Z)  . Отменяет последнее действие пользователя при редактировании техпрограммы. В случае если надо отменить несколько последних изменений техпрограммы, то данная кнопка нажимается необходимое количество раз;

Повторить (Ctrl+Y)  . Восстанавливает при редактировании техпрограммы последнее отмененное действие пользователя;

Обновить (Ctrl+R)  . Обновить (актуализировать) информацию;

Экспорт. Экспорт содержимого поля задачи в виде изображения в разных вариантах: в один файл, разбивка на файлы определённого размера, в Microsoft World.

Выделить все (Ctrl+A)  . Выделить все элементы (алгоблоки, связи) в поле задачи или макроса.

Создать спецификацию  . Создание файла спецификации оборудования в формате файл.xlsx.

Проверить код техпрограммы на наличие ошибок  . Проверяет код техпрограммы на наличие ошибок.


Войти в режим обзора (F9)  . Запускает режим Обзора.


Выйти из режима обзора (Ctrl+F2)  . Прекращает работу обзора.


Начать сеанс отладки (Ctrl+D)  . Запускает режим отладки в обзоре.


Приостановить выполнение техпрограммы (Pause)  . Работает в режиме отладки. Прекращает выполнение программы.

Перейти к текущему элементу (Alt+Multiply)  . Обзор в режиме отладки. Переход к активному алгоблоку.

Зайти в (F11)  . Обзор в режиме отладки. Выполнение всех алгоблоков, в том числе внутри макросов.

Перешагнуть (F10)  . Обзор в режиме отладки. Выполнение всех алгоблоков, без захода внутрь макросов.

Выйти из (Shift+F11)  . Обзор в режиме отладки. Выйти из макроса.


Установить точку останова  . В режиме отладки устанавливаются точки останова. Вторым вариантом установить точку останова с помощью контекстного меню **Точка останова\Установить точку останова**.

Сбросить точку останова  . Сбрасывает точку останова.

Увеличить масштаб  . Увеличить масштаб изображения в поле задач.

Уменьшить масштаб  . Уменьшить масштаб изображения в поле задач.

Установить масштаб 1:1  . Устанавливается 1:1, если ранее был изменен.

Настройка типа алгоблока (Ctrl+T)  . Выбор типа данных (Рисунок 3).

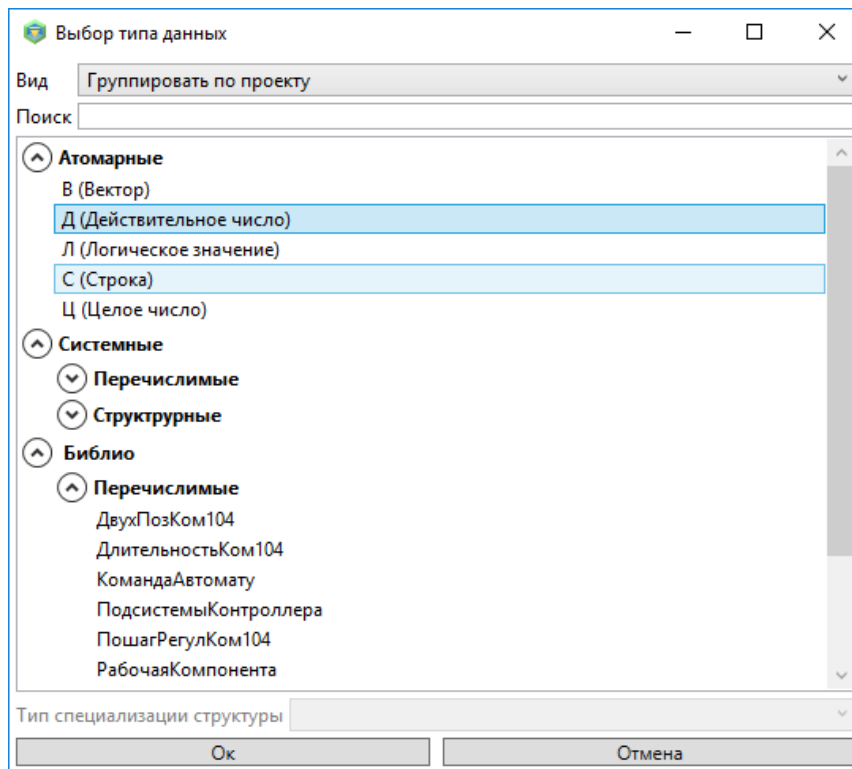


Рисунок 3 - Окно для выбора типа данных

Увеличение модификатора . Увеличивает модификатор для выбранного алгоблока.

Уменьшение модификатора . Уменьшает модификатор для выбранного алгоблока.

Задать значение модификатора (Ctrl+M) . Окно для задания значения (Рисунок 4).

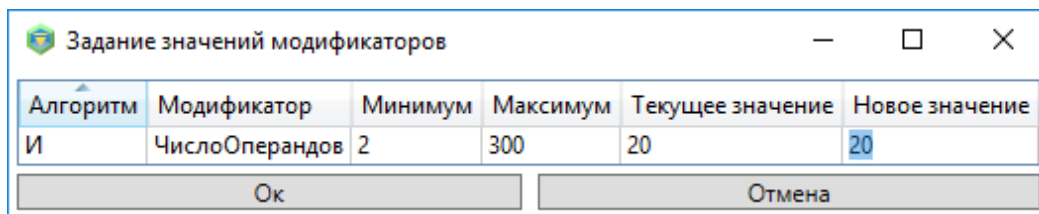


Рисунок 4 - Выбор значений модификатора

Включить качество . Включает качество сигналов.

Отключить качество . Отключает качество.

Инверсия . Добавляет знак инверсии на входе алгоблока.

Без инверсии . Удаляет знак инверсии на входе алгоблока.

Заменить на ссылки . Заменяет линию связи, между алгоблоками, на ссылки.

Заменить на линию . Ссылки на соответствующих входах и выходах алгоблоков заменяются на линию.

Автоматический порядок выполнения . Включает алгоритм автоматической расстановки порядка выполнения алгоблоков по результатам минимизации веса обратных связей.

Вынести в макрос . Выделенные объекты (алгоблоки со связями) в поле задачи выносятся в макрос.

Добавить комментарий . Добавляет комментарий на поле задачи или макроса (Рисунок 5).

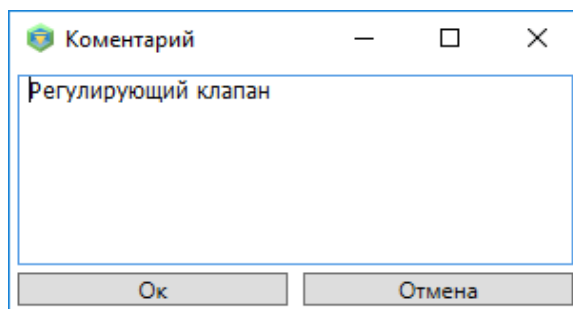


Рисунок 5 - Окно для добавления комментариев

Редактировать комментарий . Редактирование комментариев (Рисунок 6).

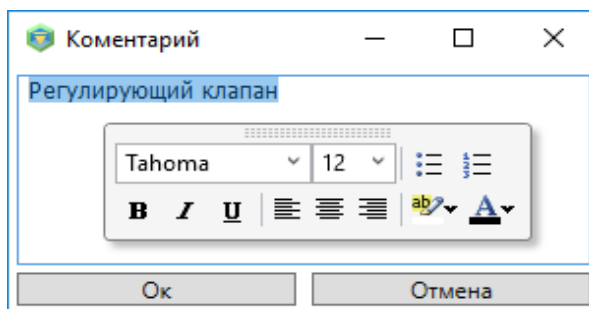


Рисунок 6 - Окно для редактирования комментариев

Заблокировать . Запрещает редактировать задачу другим пользователям.

Разблокировать . Разрешает редактировать задачу другим пользователям.

3.3 Меню Вид

3.3.1 Состав меню вид

Меню Вид Полиса имеет следующие разделы:

- Результаты поиска
- Обзор проекта
- Объекты
- Точки останова
- Стек вызовов
- Выбор алгоритмов
- Порядок выполнения
- Обзор 1
- Обзор 2
- Обзор 3
- Локальный обзор
- Список ошибок

Ниже описывается более подробно каждый пункт меню **Вид**.

3.3.2 Результаты поиска

Открывает окно с результатами поиска в **Полисе**. Окно открывается автоматически при использовании поиска.

3.3.3 Обзор проекта

Пункт меню, открывает окно с набором контроллеров, созданных в **Администраторе БД** (Рисунок 7). При помощи контекстного меню, вызываемого нажатием правой кнопкой мыши по контроллеру (Рисунок 8), создаётся иерархическая структура техпрограммы для выбранного контроллера с переменными, потоками, задачами и подзадачами.

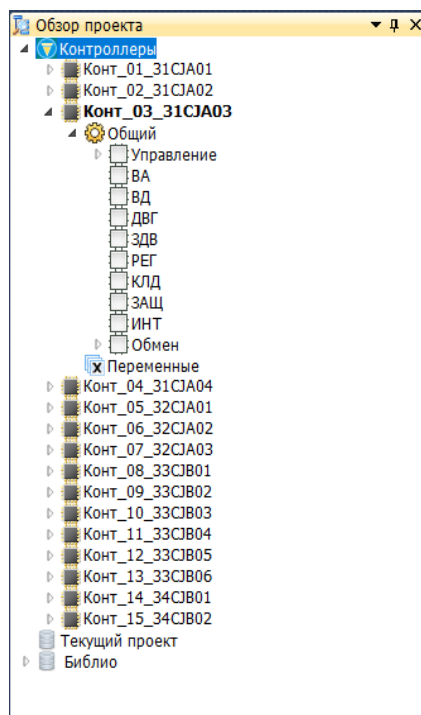


Рисунок 7 – Окно обзор проекта

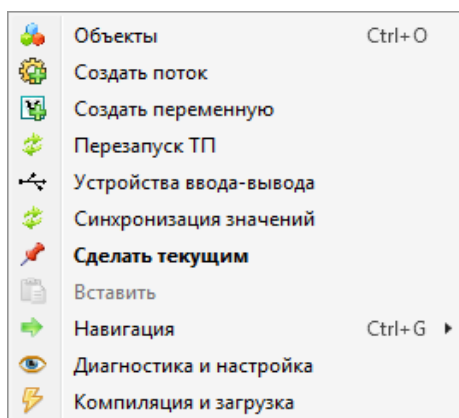


Рисунок 8 – Контекстное меню

Остановимся и рассмотрим более подробно контекстное меню:

- **Объекты** – открывает окно с объектами, относящимися к данному контроллеру.
- **Создать поток** – комплект задач, выполняемых с определённым циклом.
- **Создать переменную** – открывает окно для создания переменной, в котором необходимо указать набор свойств для новой переменной.
- **Перезапуск ТП**- перезапуск техпрограммы.
- **Устройства ввода-вывода** – открывает окно, где необходимо будет создать набор модулей УСО для выбранного контроллера, который будет соответствовать физическому расположению модулей УСО в шкафу контроллера.
- **Синхронизация значений** – синхронизирует значения между базой данных и контроллером.
- **Сделать текущим** – позволяет сделать выбранный контроллер текущим. Это позволяет войти в режим обзора, проверить код техпрограммы на наличие ошибок, начать сеанс отладки, приостановить или восстановить выполнение техпрограммы. Иначе эти функции недоступны.
- **Вставить** – вставляет находящиеся в буфере обмена задачи и подзадачи.
- **Навигация** – позволяет перейти в окно устройств ввода-вывода, а также открыть контроллер в Администраторе БД.

– **Диагностика и настройка** – открывает окно с полной информацией по контроллеру. Здесь можно посмотреть загрузку ЦП, информацию по загрузке техпрограммы, текущие ошибки, выставленные подмены, межконтроллерный обмен.

– **Компиляция и загрузка** – окно для компиляции и загрузке (см. п.3.1).

3.3.4 Объекты

Пункт меню, открывает окно со списком объектов указанного контроллера. Объекты для контроллера уже могут быть созданы в Базисе или же воспользовавшись панелью инструментов их можно создать в поле.

Для удобства все объекты можно отфильтровать по узлу и по типу (Рисунок 9). При необходимости найти ранее созданный объект можно воспользоваться строкой поиска, в которой предусмотрен поиск по марке и по имени объекта.

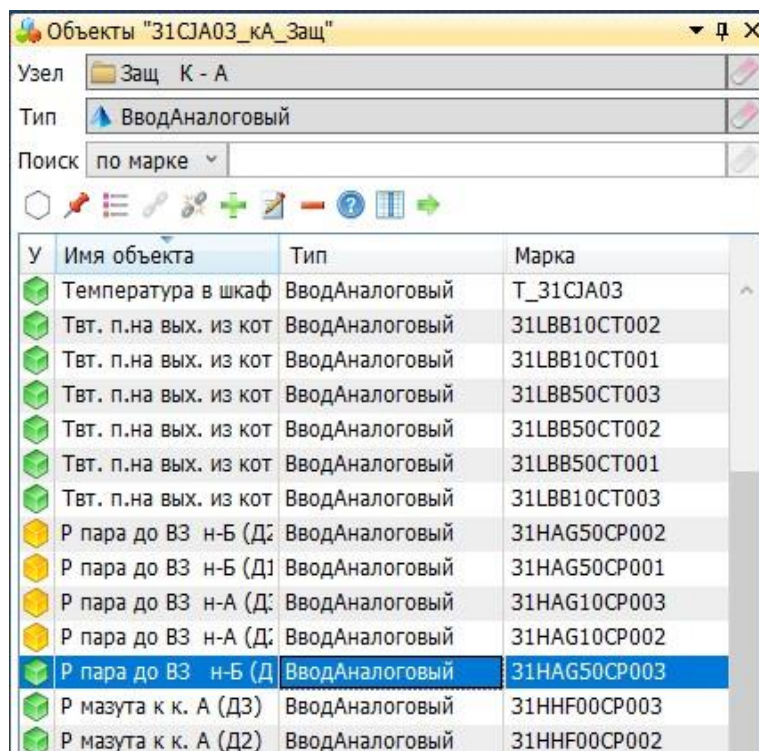


Рисунок 9 – Окно Объекты

3.3.4.1 Панель инструментов

Панель инструментов представлена на рисунке 10.



Рисунок 10 – Панель инструментов окна Объекты

– **показать непривязанные объекты.** Показывает объекты, которые не привязаны ни к алгоблоку ни к котроллеру.

– **зафиксировать контроллер** (не реагировать на переключение закладки).

– **показывать сигналы** – показывает все сигналы для выбранного объекта. Работает только для уже привязанных объектов.




– **привязать объекты к контроллеру (полупривязать).** Кнопка становится активной, после открытия окна задачи выбранного контроллера.

– **отвязать объекты.** Позволяет отвязать объект от алгоритма или контроллера.


– **создать объект.** Окрывает окно добавления объекта, аналогичное окну добавления объектов в Базисе.

– **изменить.** Возможность отредактировать марку объекта, добавить описание, а также изменить узел для объекта.

- **удалить выделенные объекты.** Удаляет выбранные объекты из базы данных
- **справка по типу объекта.** Открывает библиотеку алгоритмов с описанием выбранного алгоритма.
- **Колонки.** Настройка отображения колонок в окне.

Объект в списке может быть в трёх вариациях: **без привязки**, **с привязкой к контроллеру (полупривязан)** и **с привязкой к алгоблоку**. Состояние привязки можно определить по колонке У, где  - без привязки,  - привязка к контроллеру,  - привязка к алгоблоку.

3.3.4.2 Привязка объектов

Для того, чтобы привязать объект к контроллеру (полупривязать), необходимо либо явно указывать контроллер при создании нового объекта в **Базисе**, либо воспользоваться панелью инструментов **Полиса**. Для выбора контроллера в **Полисе**, в окне **Обзор проекта**, необходимо открыть окно задачи или выделить контроллер и при помощи контекстного меню выбрать пункт **Объекты**. После чего выбранный контроллер отобразится в шапке окна **Объекты** и станет активной кнопка  (привязать объекты к контроллеру). При нажатии на кнопку объект привяжется к выбранному контроллеру.

Примечание. При создании нового объекта в **Полисе**, он будет автоматически привязываться к выбранному контроллеру.

Для того, чтобы привязать объект к алгоблоку, открываем окно задачи с алгоблоком. Выбираем непривязанный или полупривязанный объект из списка в окне **Объекты**, зажимаем левой кнопкой мыши объект и перетаскиваем его в шапку алгоблока, либо если требуется привязать объект к входным или выходным сигналам, на соответствующий вход или выход алгоблока. Более подробно технология привязки объектов к алгоблокам описаны в пункте 2.8.

Примечание. Тип объекта должен соответствовать алгоблоку, входному или выходному сигналу. Например, невозможно привязать объект **ВводАналог** к алгоблоку **ВводДискр**.

3.3.5 Точки останова

Точки останова необходимы для отладки техпрограммы. В данном окне отображается список всех установленных точек останова.

3.3.6 Выбор алгоритма

В ПТК «СУРА» есть встроенная библиотека алгоритмов. Данный пункт меню открывает окно со списком всех алгоритмов (Рисунок 11). Алгоритмы представлены в виде списка по алфавиту и списка с сортировкой по группам. По умолчанию окно открывается справа в приложении **Полиса**.

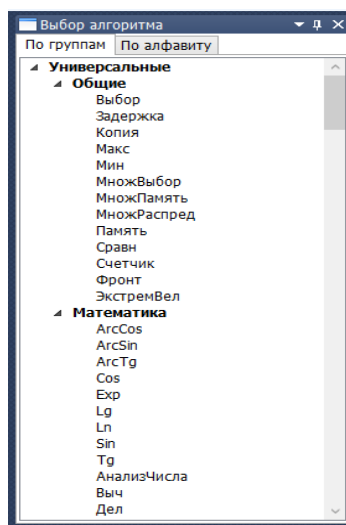


Рисунок 11 – Окно Выбор алгоритма

Для того, чтобы добавить алгоритм, открываем окно задачи. Выбираем алгоритм из списка, зажимаем левой кнопкой мыши алгоритм и перетаскиваем его в окно задачи.

3.3.7 Стек вызовов

Работает в режиме отладки. Выводит информацию о шагах в процессе отладки.

3.3.8 Порядок выполнения

Данный пункт открывает окно, в котором представлен порядок выполнения алгоритмов. Список формируется автоматически по мере добавления алгоритмов в поле задач.

В процессе составления техпрограммы может возникнуть необходимость изменить порядок выполнения алгоритмов. Чтобы полностью не менять программу, удаляя алгоритмы и выставляя их в нужном порядке, открываем данное окно и выставляем алгоритмы в том порядке, который нам необходим. Для этого выделяем алгоритм либо в поле задач, при этом алгоритм подсвечивается в списке, либо непосредственно в самом списке. После этого стрелочками, находящимися сверху окна, выставляем порядок выполнения.

3.3.9 Обзор 1, Обзор 2, Обзор 3

Значения переменных и сигналов от различных оперативных серверов OPCUA.

3.3.10 Локальный обзор

Открывает обзор значений локальных переменных текущего программного модуля.

3.3.11 Список ошибок

Пункт меню открывает окно с сообщениями от компилятора техпрограммы (Рисунок 12). Сообщения бывают трёх видов: **ошибки**, **предупреждения**, **сообщения**. Путём нажатия левой кнопкой мыши по названию сообщений можно скрывать или показывать сообщения в окне. Также в окне отображается полоса состояния компиляции с соответствующим описанием. Кнопка **Прервать** прерывает выполнение компиляции.

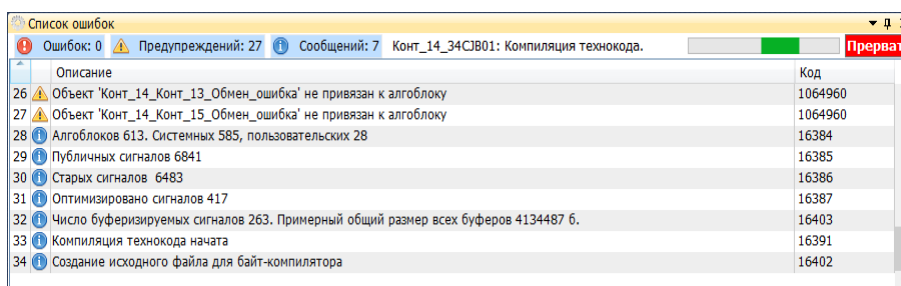


Рисунок 12 – Окно Список ошибок

При нажатии правой кнопкой мыши по одному из сообщений вызывается контекстное меню, которое позволяет выделять сообщения, а также перейти к источнику сообщения или открыть окно, где будет указан путь и возможность перейти к источнику сообщения.

3.4 Свойства задач

Первое, что нужно сделать после запуска **Полиса**, это открыть окно **Обзор проекта** и выбрать нужный контроллер.

Следует помнить, что список системных модулей и контроллеров создается не в **Полисе**, а в **Администраторе БД**. В **Полисе** можно лишь выбрать нужный контроллер.

3.4.1 Список задач

После того, как контроллер выбран, можно создать список задач и подзадач, дополнить этот список новыми задачами или выбрать нужную задачу из готового списка.

Чтобы создать задачу для выбранного контроллера необходимо создать поток для контроллера, а уже в потоке при помощи контекстного меню добавить задачу. Контекстное меню вызывается нажатием правой кнопкой мыши по потоку или в дальнейшем по задаче. При помощи контекстного меню можно выполнить следующие действия:

- создать новую задачу или подзадачу;
- создать переменную для задачи;

- скопировать задачу;
- вставить задачу;
- удалить задачу;
- переместить задачу;
- изменить свойства задачи.

Когда создается новая задача, она помещается в конец списка задач с заданным именем. Аналогичные правила действуют относительно создания подзадач. Разница лишь в том, что в этом случае следует предварительно выбрать родительскую задачу.

Любую задачу можно удалить. Когда задача удаляется, удаляются также все относящиеся к ней подзадачи и все входящие в них алгоблоки. **При удалении задачи следует проявлять осторожность, поскольку вернуть удаленную задачу уже не удастся.**

Любую имеющуюся задачу можно скопировать и затем вставить в данный или любой другой контроллер.

Примечание. При копировании создается точная копия алгоритмической структуры исходной задачи, но все привязанные марки в копии отвязываются (естественно, в исходной задаче все остается без изменения).

С помощью стрелок контекстного меню любую задачу можно переместить, но только в пределах своего уровня иерархии. Перемещение можно выполнить при помощи зажатой клавиши **Ctrl** и стрелками клавиатуры $\uparrow\downarrow$. Для того, чтобы имеющуюся задачу переместить в другую задачу (т.е. сделать ее подзадачей другой задачи), ее следует перетащить в нужное место левой клавишей мыши. Порядок расположения задач определяет порядок их выполнения в контроллере. Правило здесь такое: раньше выполняются алгоблоки задачи, расположенной выше по списку.

3.4.2 Поле задачи

Используя панель инструментов, можно менять масштаб изображения текущей задачи. Его можно на несколько ступеней уменьшить, при этом в пределах экрана разместится большее число алгоблоков. Масштаб уменьшенного изображения можно ступенями увеличивать или сразу вернуться к исходному размеру.

Для удобного размещения алгоритмов в поле задач в **Полисе** созданы линии выравнивания относительно уже размещённых алгоритмов.

Если алгоблок перемещать за пределы видимой части экрана, поле задачи автоматически расширяется и становится больше поля экрана. Поле задачи можно сделать сколь угодно большим (как по вертикали, так и по горизонтали), при этом имеется одно общее правило: **одна задача занимает одно (и только одно) сплошное поле.**

Прокручивать поле задачи можно либо обычными средствами Windows. Прокручивать поле задачи можно также с помощью клавиш:

- управление колёсиком мыши - плавно в соответствующем направлении;
- **PageUp, PageDown** - один экран вверх и вниз;
- **Ctrl+ (PageUp, PageDown)** – крайние положения вверх и вниз;
- **Ctrl+(←,→)** – один экран влево и вправо;
- **Home, End** – крайние положения влево и вправо;
- **Ctrl+ (PageUp, PageDown)** - верхний левый и нижний правый угол.

Прокручивать изображение удобно, используя мышь с колесиком. При вращении колесика поле задачи прокручивается вверх-вниз. Шаг и скорость прокрутки задаются стандартным образом в диалоговом окне Windows при настройке параметров мыши.

В **Полисе** имеется функция отката, которая после удаления и перемещения алгоблоков, связей и констант позволяет вернуться к предыдущему, а после отката – к последующему состоянию (два значка с изогнутыми стрелками). Число ступеней отката логически не ограничено (физическое ограничение связано с размером памяти компьютера).

3.4.3 Размещение алгоблоков

Эликонт содержит развитую библиотеку алгоритмов, каждый из которых можно использовать многократно в одной или разных задачах. Алгоритм, «введенный» в задачу, называется алгоблоком (алгоритмическим блоком). Таким образом, алгоблок - это работающий алгоритм.

Когда открывается новая задача, первое, что надо сделать, это разместить в поле задачи, требуемые алгоблоки. Для этого выбирается значок «+» контекстного меню поля задач (рисунок 13) или открывается окно выбор алгоритма

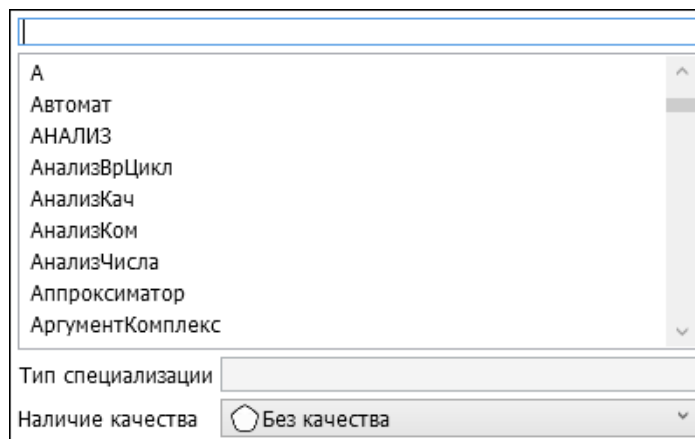


Рисунок 13 – Список библиотечных алгоритмов.

Ряд алгоритмов имеет специальный атрибут, называемый модификатором. Обычно он задает число однотипных элементов – число каналов алгоритма, обслуживающего УСО, число входов сумматора и т.п. Требуемое значение модификатора устанавливается в отдельном поле **Модификатор**.

Выбранный алгоритм превращается в алгоблок, который первоначально размещается в любом месте поля задач. Захватив алгоблок мышью за верхний заголовок (шапку), его можно переместить в любую другую позицию.

Чтобы алгоблок сразу разместить на требуемое место, можно в этом месте щелкнуть правой клавишей мыши и из открывшегося меню выбрать команду **Добавить алгоблок**, - позиция верхнего левого угла алгоблока совпадет с точкой, отмеченной мышью.

Установленный алгоблок можно удалить или заменить в нем модификатор. Для этого правой клавишей мыши щелкается по шапке алгоблока и из открывающегося контекстного меню выбирается команда соответственно **Удалить** или **Изменить модификатор**. Естественно, что все связи, относящиеся к удаленному алгоблоку, ликвидируются. При изменении модификатора связи «отрезанных» каналов ликвидируются, а оставшихся – сохраняются.

3.4.4 Реквизиты алгоблоков

Алгоблок всегда имеет вид прямоугольника. На рисунке 14 представлен алгоблок с реквизитами. Его ширина и высота зависит от вида алгоритма, а также от значения модификатора.

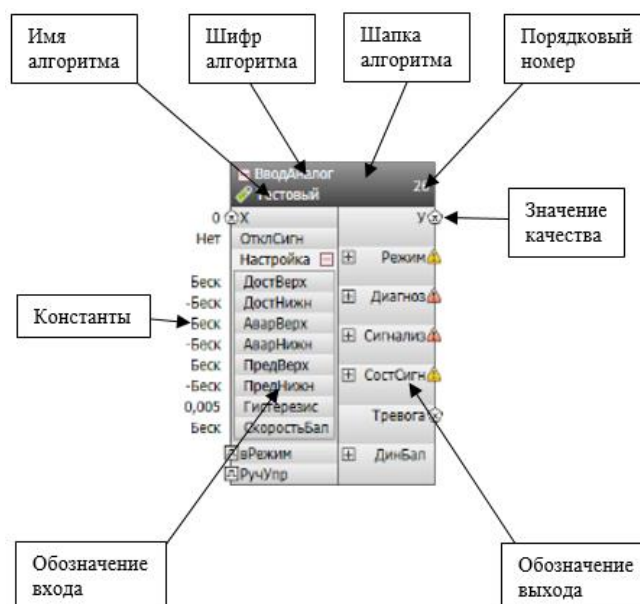


Рисунок 14 – Реквизиты алгоблоков

Есть алгоритмы, для которых вручную настраивается **Тип данных** и значение **Качества**. В первом случае возле шифра алгоритма в \diamond будет указан тип данных алгоритма, во втором возле порядкового номера будет стоять значок качества.

В верхней части (шапке) алгоблока указываются:

- шифр алгоритма;
- имя алгоритма;
- порядковый номер, определяющий порядок выполнения программы алгоритма;
- значение качества;
- тип данных.

Имя алгоблока (не путать с именем алгоритма) присваивается пользователем. Для этого в области шапки нажимается правая клавиша мыши и в открывшемся контекстном меню выбирается строка **Имя алгоблока**. В поле **Имя алгоблока** можно впечатать любое имя, содержащее не более 40 знаков, после чего оно появится в нижней части шапки алгоблока.

Что касается порядковых номеров, то по мере ввода алгоблоков они первоначально назначаются последовательно – первый введенный алгоблок получает первый порядковый номер, второй – второй порядковый номер и т.д. Однако затем порядок выполнения алгоблоков можно изменить (более подробно об этом см. ниже Перемещение и перестановка алгоблоков).

В отличие от порядкового номера, который назначается в пределах одной задачи, идентификатор алгоблока является глобальным относительно всего контроллера. Это означает, что в одном контроллере идентификатор каждого алгоблока уникален и не может повторяться. Впрочем, пользователю об этой уникальности заботиться не требуется, - **Полис** сам назначает адреса нужным образом.

Тем не менее следует иметь ввиду следующее:

- в пределах одного контроллера имеются ограничения на число алгоблоков и ряд других параметров; подробней об этом см. ниже;
- когда отдельные алгоблоки удаляются, порядковый номер оставшихся алгоблоков изменяются следующим образом: если порядковый номер алгоблока меньше адреса удаленного алгоблока, то его порядковый номер не меняется, а если больше, то его порядковый номер уменьшается на единицу;
- когда алгоблок добавляется, он получает порядковый, следующий по порядку.

Шапка алгоблока служит не только для того, чтобы в ней обозначить реквизиты. Если щелкнуть по шапке правой клавишей мыши, можно увидеть контекстное меню, через которое

вызываются различные команды изменения параметров или режимов (подробней назначение этих команд описывается ниже в соответствующих разделах).

3.4.5 Входы-выходы алгоблока

Ниже шапки размещаются входы-выходы алгоблока, которые имеют обозначения, соответствующие их описанию в библиотеке алгоритмов Эликонта.

Помимо обозначения для каждого входа указывается формат, в котором в **Полисе** представляются значения входных сигналов. Этот формат можно изменять, однако в большинстве случаев его можно оставить тем, который в **Полисе** принят по умолчанию.

3.4.6 Копирование алгоблоков

В технологической программе контроллера можно копировать выделенный алгоблок и затем вставлять его внутри исходной задачи или в другие задачи контроллеров в рамках проекта. Эта операция осуществляется через буфер обмена.

Выполнить копирование можно следующими способами:

– Выделить алгоблок в поле задач щелкнуть правой клавишей на шапке алгоблока и из контекстного меню выбрать команду **Копировать**, затем команду **Вставить**.

– Выделить алгоблок в поле задач и нажать комбинацию клавиш **Ctrl+C** для копирования и комбинацию клавиш **Ctrl+V** для вставки.

Примечание. Необходимо помнить, что во вставленном алгоблоке все привязанные марки отвязываются.

Описанными выше способами возможно также проведение групповых операций с алгоблоками. Например, можно скопировать или удалить сразу несколько алгоблоков. Для того, чтобы скопировать несколько алгоблоков нужно в окне задачи мышью выделить их. Для этого нужно удерживать левую клавишу мыши и нужным образом растянуть рамку, либо последовательно щелкать левой клавишей мыши по шапкам алгоблоков, удерживая нажатой клавишу **Shift**. Если уже выделено несколько алгоблоков и какие-то алгоблоки нужно исключить из выделения, следует по ним щелкнуть левой клавишей мыши при нажатой клавише **Shift**. Для того, чтобы отменить выделение всех алгоблоков нужно щелкнуть левой клавишей мыши по свободному полю.

Выделенную группу можно перемещать, удалять, вырезать, копировать и заменять в ней все ссылки на линии конфигурации.

3.4.7 Свойства входов алгоблоков

Входы алгоблоков имеют следующие общие свойства:

На большинстве входов любого алгоблока можно задать константу либо связать этот вход с каким-либо источником. В первом случае вход называется свободным, во втором – связанным. В отдельных случаях вход нельзя связывать и на нем всегда задается константа. Напротив, некоторые входы предназначены исключительно для связывания и на них нельзя задавать константу.

Конфигурируемый вход любого алгоблока можно соединить с любым выходом любого алгоблока (в том числе со своим собственным выходом). Связывать между собой можно как алгоблоки одной задачи, так и алгоблоки, принадлежащие разным задачам.

С одним входом можно связать только один выход. Это означает, что у одного приемника может быть только один источник. В то же время, к одному выходу можно присоединить любое число разных входов одного или разных алгоблоков.


На сконфигурированном входе можно установить инверсию. Для дискретного входа инверсия означает замену логического нуля на единицу, а логической единицы – на ноль. Для аналогового и других непрерывных сигналов инверсия означает изменение знака на противоположный, а для векторного входа – побитную инверсию. На входе, имеющем формат времени, инверсию устанавливать не следует. На свободном входе (с константой) инверсия не задается.

3.4.8 Общий случай установки констант

В исходном состоянии на входах алгоблоков константы равны значениям, указанным в описании библиотеки алгоритмов. На любом свободном (не связанном конфигурацией) входе алгоблока эти константы можно изменять. Совокупность установленных констант определяет параметры настройки алгоблока. Значения констант выводятся слева от полей соответствующих входов алгоблока.

Чтобы изменить значение константы, щелчком мыши на значении константы соответствующее поле переводится в режим редактирования, после чего с клавиатуры вводится новое значение. Используя клавиши «↓» или «↑», можно перейти к редактированию соседнего свободного входа. **Примечание:** константа, равная бесконечности, отображается в Полице надписью: «Беск».

3.4.9 Выбор типа данных

Существует два вида алгоблоков: **типизированные** и **нетипизированные**. Для типизированных алгоблоков, по умолчанию, заложен определённый тип данных для всех его переменных, который нельзя изменить, например, **ВводАналоговый**. Нетипизированные не имеют определенного типа данных, например, **Выбор**, но для них его можно установить вручную при помощи панели инструментов . Тип данных автоматически подхватывается при проведении связи между типизированным и нетипизированным алгоблоком. Если же необходимо провести связь между двумя нетипизированными алгоблоками в таком случае необходима ручная установка типа данных.

Примечание. Переменная – это именованный элемент данных в памяти программируемого сервера. Значение переменной может меняться во времени и это значение может быть наблюдаемо извне сервера. Каждый вход и выход алгоблока является примером переменной. Кроме того, проектант может создавать самостоятельные переменные, не являющиеся входами или выходами.

Типы данных бывают нескольких видов (см. Таблицу 1).

Вид	Описание	Пример
Атомарные	Элементарное значение.	Ток двигателя
Перечислимые	Значение имеет некоторый набор именованных допустимых состояний (состояния являются взаимоисключающими).	Состояние задвижки (открыта, закрыта и т.д.)
Интерналы	Внутренняя величина. В отличие от других типов, значения интерналов не могут передаваться между контроллерами и между разными потоками одного контроллера. Значения интерналов не отображаются на станциях верхнего уровня и не архивируются.	Дескриптор файла
Структурные	Объединяет в себе несколько именованных элементов, каждый из которых является атомом, перечислением или набором. Структура не может содержать в качестве элементов другие структуры.	Настройки задвижки

Атомарные типы являются основными. В таблица 2 представлено описание для каждого атомарного типа данных.

Таблица 2 - Атомарные типы данных

Тип		Описание	Пример
Д	Действительное число	Действительное число (со знаком) в формате с плавающей запятой. Диапазон: от -1.7308 до 1.7308, точность – не менее 15 знаков.	Значение дискретного датчика
Ц	Целое число	Целое число со знаком. Диапазон: от -2 147 483 648 до 2 147 483 647.	Показания электросчетчика
Л	Логическое значение	Имеет 2 значения: «Да» и «Нет».	Значение дискретного датчика
В	Вектор	Набор из 32 бит. В основном, векторы применяются в алгоритмах, реализующих взаимодействие с сторонними системами.	Регистр
С	Строка	Строка, не превышающая 256 символов.	Имя файла

Остальные виды являются системными и имеют большой список типов данных. Описание каждого из них представлено в **Полисе** при наведении курсора на нужный тип данных, в открывающемся окне.

3.5 Конфигурирование алгоблоков

3.5.1 Техника конфигурирования

При конфигурировании, чтобы соединить вход с каким-нибудь выходом, следует подвести мышь к требуемому выходному полю (источнику) - курсор примет вид руки - и затем щелкнуть мышью - рука повернется в другую сторону. Теперь следует щелкнуть мышью на требуемом входном поле (приемнике), - выбранный выход соединится с выбранным входом, а курсор вернется к стандартному изображению. Такой способ, называемый автоматическим, проводит связь по оптимальному маршруту.

Маршрут можно изменить и после того, как связь проведена. Все отрезки линии связи, кроме первого и последнего, можно перемещать с помощью мыши. Для этого к перемещаемому отрезку подводится курсор, - он примет вид двойной стрелки. Теперь, «зацепив» вертикальный отрезок мышью, его можно переместить влево или вправо, а горизонтальный отрезок - вверх или вниз. Еще одна возможность - подвести курсор к первому или последнему отрезку, - он примет вид перекрестья.

Чтобы связать алгоблоки, принадлежащие разным задачам, следует вызвать на экран обе задачи и разместить окна обеих задач так, чтобы был виден как алгоблок-источник, так и алгоблок-приемник. После этого алгоблоки связываются так же, как если бы они находились в одной задаче. Возможен и другой вариант. Можно «зацепить» поле источника и затем щелкнуть левой клавишей мыши на значке задач в шкале инструментов – откроется диалоговое окно со списком задач. Теперь нужно выбрать требуемую задачу и после того, как она появится на экране, действовать описанным выше способом. **Примечание:** при связи алгоблоков, находящихся в разных задачах, линия конфигурации всегда заменяется на ссылку (см. ниже Ссылки).

При конфигурировании можно задать еще один параметр – инверсию. Если сигнал на входе должен инвертироваться, следует подвести курсор к входному полю, нажать правую клавишу мыши и в открывшемся диалоговом окне установить галочку **Инверсия**. Признак инверсии отображается на схеме маленьким кружком слева от входного поля.

3.5.2 Ссылки

Если связываемые алгоблоки находятся далеко друг от друга, линии связи получаются длинными и вся структура становится плохо читаемой. В таком случае явную связь целесообразно заменить на ссылку. Для этого нужно поместить курсор на любую точку связи, нажать правую клавишу мыши и из контекстного меню выбрать команду **Заменить на ссылку**, - линия,

соединяющая алгоблоки, исчезнет и вместо нее появятся две ссылки: одна рядом с выходом источника, другая – рядом со входом приемника. **Внимание:** если связываются алгоблоки разных задач, связь между ними всегда имеет вид ссылки.

Ссылка, «прикрепленная» к источнику, содержит сведения об алгоблоке-приемнике, ссылка, «прикрепленная» к приемнику, содержит аналогичные сведения об алгоблоке-источнике.

Ссылку на входе и выходе алгоблока можно перемещать. Это особенно удобно в том случае, когда с одним выходом связано несколько ссылок на разные приемники, - чтобы эти ссылки не накладывались друг на друга, их следует «расташить».

В **Полисе** можно быстро найти источник или приемник, на которые указывают ссылки. Для этого следует дважды щёлкнуть левой кнопкой мыши по ссылке, - произойдет автоматическая прокрутка поля задачи, так что искомый алгоблок переместиться в видимую часть экрана и будет выделен синим цветом. Эта команда работает также и в случае, когда источник и приемник находятся в разных задачах, - нужная задача автоматически вызовется на экран и в ней будет найден искомый адресат.

3.5.3 Дополнительные возможности конфигурирования

Если подвести курсор к линии связи и по правой клавише мыши открыть контекстное меню, то по отношению к выбранной связи можно выполнить ряд дополнительных действий:

– **Удалить связь.** Если выбрать эту операцию, проведенная связь будет удалена, а на входе алгоблока, к которому подходила эта связь, будет установлена константа по умолчанию.

– **Оптимизировать связь.** По этой команде текущий маршрут заменяется на маршрут, выбираемый самим **Полисом**.

Пункты меню: **инверсия, без инверсии, заменить на ссылку, заменить на линию;** дублируют кнопки панели инструментов.

При перемещении алгоблока вслед за ним будут перемещаться все связанные с ним линии конфигурации, так что все время будет сохраняться оптимальный маршрут связи.

3.5.4 Пересечение линий связи

В задаче с большим числом алгоблоков и линий связи последние могут пересекаться. Если пересечение линий связи означает их соединение, такое пересечение обозначается точкой. Если линии пересекаются, но не соединяются, точка отсутствует.

3.6 Перемещение и перестановка алгоблоков

3.6.1 Возможности перемещения

После того как алгоблок введен в поле задач, его можно перетащить в любое другое место поля задачи (но нельзя переместить в другую задачу). Перемещать можно и законфигурированный алгоблок – подходящие к нему связи будут следовать за алгоблоком, куда бы его ни двигать.

Чтобы переместить алгоблок, следует захватить левой клавишей мыши шапку алгоблока и, не отпуская клавишу, двигать курсор мыши в нужном направлении. Когда место выбрано, клавиша мыши отпускается. Перемещать можно не только алгоблоки, но и ссылки, как на источнике, так и на приёмнике.

Техника перемещения та же – ссылка захватывается левой клавишей мыши и переносится в нужное место. Образованная линия связи между ссылкой и алгоблоком имеет особенность, когда перемещается алгоблок – за ним следует и ссылка, при этом конфигурация связи между ссылкой и выходом алгоблока остается неизменной.

3.6.2 Наложение алгоблоков и ссылок

При перемещении алгоблоков может произойти их наложение – один алгоблок закроет изображение другого алгоблока. В этом случае перекрытый элемент (алгоблок или ссылка) будет закрашен фиолетовым цветом. На рисунке 15 приведен пример наложения алгоблока на другие элементы в поле задач.

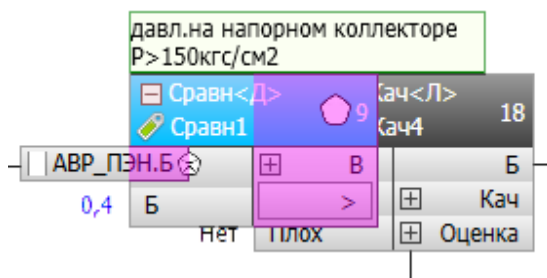


Рисунок 15 – Наложение алгоблоков

Во всех случаях рекомендуется растянуть наложенные элементы.

3.6.3 Перестановка алгоблоков

Размещение алгоблоков в поле задачи никак не определяет порядок их выполнения. Этот порядок вначале целиком определяется последовательностью ввода алгоблоков – вначале выполняется первый введенный алгоблок, затем второй и т.д. Порядковый номер алгоблока (не путать с его адресом) указывается в середине справа шапки алгоблока.

В любое время порядок выполнения алгоблоков можно изменить. Для этого выбирается пункт меню Вид порядок выполнения см. п.... Это окно показывает порядок выполнения алгоблоков – первым выполняется самый верхний в списке алгоблок, затем следующий по списку и т.д. Изменить порядок выполнения алгоблоков можно с помощью экранных клавиш, расположенных в верхней части диалогового окна. Для этого выбирается нужный алгоблок, и он перемещается выше или ниже по списку.

Алгоблок, который нужно переместить, можно выбрать двумя способами. Можно указать на требуемый алгоблок в диалоговом окне, - одновременно этот алгоблок будет выделен в поле задачи, причем если он находился вне видимой части экрана, поле задачи автоматически прокрутится так, чтобы он стал виден. Другой вариант – выделить нужный алгоблок в поле задачи. Тогда он автоматически будет выбран в диалоговом окне перестановки.

3.7 Комментарии

3.7.1 Виды комментариев

Для того, чтобы в техпрограмме легче было ориентироваться, ее рекомендуется снабжать комментариями. Работая с **Полисом**, пользователь может использовать три типа комментариев:

- к задаче;
- к алгоблоку;
- к линии связи в виде ссылок для источника и приёмника;

3.7.2 Комментарии к задаче

Комментарии к задаче помещаются в любом свободном месте поля задачи, причем число таких комментариев в каждой задаче не ограничено. При наборе текста можно использовать любые символы, имеющиеся на стандартной клавиатуре. Чтобы начать текст с новой строки, можно воспользоваться клавишей Enter.

Для того, чтобы ввести комментарий, следует щелкнуть правой клавишей мыши в нужном месте свободного поля задачи и в открывшемся диалоговом окне выбрать строку **Добавить комментарий**. После этого в выбранном месте поля задачи появится окно **Комментарий**, в которое можно ввести текст. Когда текст введен, следует нажать кнопку ок – после чего текст помещается в поле задачи. Затем этот текст с помощью мыши можно перетащить в любое другое место.

3.7.3 Другие виды комментариев

Аналогично комментариям к задаче можно вводить комментарии к каждому алгоблоку и к линии связи в виде ссылок между алгоблоками, причем в последнем случае можно вводить два разных (или одинаковых) комментария к источнику и приемнику сигнала. Введенный текст можно переместить в другое место, для текста можно выбрать требуемый шрифт. Особенность этих комментариев заключается в том, что текст «привязывается» к месту расположения алгоблока – при

перемещении алгоблока соответствующим образом перемещаются как комментарии к данному алгоблоку, так и комментарии к ссылкам, подходящим к данному алгоблоку.

3.8 Привязка

3.8.1 Назначение привязки

Термином привязка обозначаются процедуры, связывающие алгоблок или его отдельные элементы с объектами базы данных. Техника привязки зависит от свойств алгоблока – является ли он объектным или обычным, одноканальным или многоканальным.


В библиотеке имеется группа алгоритмов, называемых объектными. Эти алгоритмы формируют обобщенную информацию о свойствах реальных или виртуальных объектов (датчиков, механизмов, регуляторов и т.п.). Чтобы информацию о таких объектах передавать оперативным станциям (операторской станции, архиву и т.д.), соответствующие алгоблоки необходимо привязать к тем или иным объектам в базе данных.

Объектные алгоритмы могут быть многоканальными. В этом случае к различным объектам базы данных привязывается индивидуально каждый канал алгоблока.

Наконец, в качестве объекта может выступать отдельный аналоговый, дискретный, числовой и временной сигнал, вычисленный в контроллере. В таком случае к объектам базы данных привязывается отдельный выход или вход алгоблока, отражающий требуемый сигнал, а если сигнал векторный, то и отдельный бит, который привязывается как дискретный сигнал.

Составлять технологическую программу в **Полисе** можно до ввода объектов в **Базисе**, но, чтобы выполнить привязку, требуемый объект обязательно должен быть введен в **Базисе**.

3.8.2 Отображение привязки

В **Полисе** предусмотрена специальная метка , которая указывает на привязанные элементы. Рядом с меткой добавляется название марки. На рисунке 16 показана привязка марки для алгоблока и для его входа и выхода.

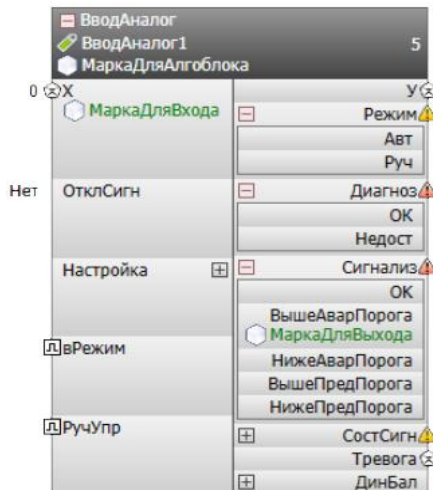


Рисунок 16 – Отображение привязки

3.8.3 Технология привязки

Рассмотрим технологию привязки на примере алгоритма **ВводАналоговый**. Осуществить привязку можно двумя способами. Если объект был уже создан в **Базисе** тогда выбираем непривязанный или полупривязанный объект из списка в окне **Объекты**, зажимаем левой кнопкой мыши объект и перетаскиваем его в шапку алгоблока, либо если требуется привязать объект к входным или выходным сигналам, на соответствующий вход или выход алгоблока. В нашем случае это следующие объекты: **МаркаДляАлгоблока**, **МаркаДляВхода**, **МаркаДляВыхода** (см. Рисунок 16). Второй вариант применим, если объект не был создан в **Базисе**. В этом случае щелкаем правой клавишей мыши в области шапки алгоблока из открывшегося контекстного меню выбираем команду «Создать и привязать объект», - открывается диалоговое окно создания объекта, аналогичное окну добавления объекта в **Базисе**, в котором уже указаны тип и контроллер, нужно

заполнить обязательные поля, выбрать узел и нажать кнопку добавить (см. Рисунок 17) Теперь выбранный канал привязан к выбранной марке объекта. Если требуется привязать отдельный канал или отдельный вход, или выход, можно подвести курсор мыши к выбранному полю и дальше действовать так же, как при привязке через шапку алгоблока.

Чтобы отвязать привязанную марку нужно щёлкнуть правой кнопкой мыши по шапке алгоблока, входу или выходу и из выпадающего контекстного меню выбрать один из пунктов: отвязать объект от алгоблока, отвязать объект от контроллера и удалить объект. При отвязывании нужно быть осторожным – вернуть привязку можно, только заново повторив все описанные выше действия.

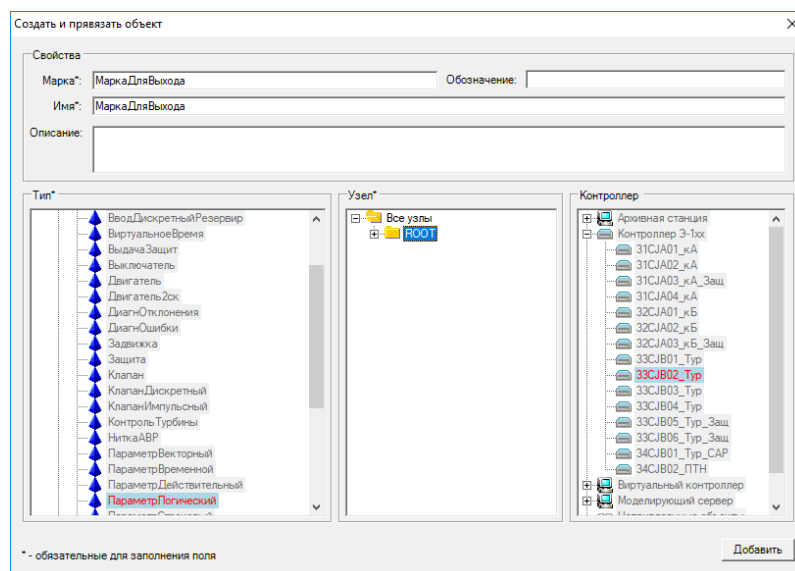


Рисунок 17 – Добавление объекта привязки

Какие бы элементы алгоблока ни привязывались к объекту, существует общее правило: один объект базы данных в качестве элемента привязки можно использовать только один раз. Если объект уже привязан, он исключается из списка свободных объектов, показываемых в поле выбора объектов окна привязки.

3.9 Права доступа и коллективная работа с Полисом

3.9.1 Права доступа

Права доступа определяют, что может и что не может делать конкретный пользователь. Права доступа к **Полису** (как и к другим приложениям) прописываются в программе **Администратор БД**. В ней для каждого пользователя задаются индивидуальные права на работу со следующими процедурами Полиса:

- редактирование техпрограммы;
- редактирование дерева задач;
- компиляция техпрограммы;
- загрузка техпрограммы.

Каждое из этих прав можно задать в пределах всего проекта или в пределах какого-то определенного узла (либо нескольких узлов), - например, для котла, турбины и т.д. (дерево узлов определяется в **Администраторе БД**). Если пользователь не имеет прав, он не сможет выполнить соответствующую процедуру, однако отсутствие прав не препятствует просмотру задач. Если права отсутствуют, и задача находится в режиме просмотра, то при первой попытке ее редактирования возникает сообщение с указанием на ключ защиты Sentinel.

3.9.2 Коллективная работа

В разработке одного проекта обычно участвует несколько пользователей. Чтобы сохранить целостность БД, используется механизм блокировок, которые в большинстве случаев ставятся автоматически. В Полисе реализуются следующие возможности коллективной работы:

Несколько пользователей могут одновременно редактировать разные задачи как одного и того же, так и разных контроллеров.

Несколько пользователей могут просматривать одну и ту же задачу, но только один (тот, который начал редактирование) может продолжать ее редактировать. Как только в задаче что-то изменилось, она блокируется и пока не будет закрыта, никто другой ничего не может в ней изменить.

На время редактирования дерева задач в данном контроллере в режиме **Обзор** блокируется возможность такого редактирования для других пользователей. Эта возможность появляется по окончании редактирования дерева задач.

Если один пользователь запустил компиляцию, загрузку или блокирующий обзор контроллера (см. ниже Отладка техпрограмм. Обзор), то пока не завершились эти процессы, другие пользователи не могут запустить эти операции в том же контроллере и не могут редактировать ни одну из задач данного контроллера (но могут просматривать задачи).

4 Отладка техпрограмм

Для отладки созданной средствами **Полиса** техпрограммы в реальном времени при ее работе в контроллере, программу необходимо скомпилировать, загрузить в контроллер и включить режим работы **Обзор**, при котором **Полис** периодически запрашивает состояния алгоблоков программы и выводит их значения на экран.

4.1 Компиляция и загрузка техпрограммы

Компиляция техпрограммы – это процедура, которая исходную графическую форму техпрограммы, представленную в виде задач, алгоблоков, связей и констант, преобразует в единый файл, «пригодный» для загрузки в контроллер. Этот файл сохраняется в базе данных и в любое время может быть загружен в контроллер.

Окно компиляции и загрузки представлено на рисунке 18

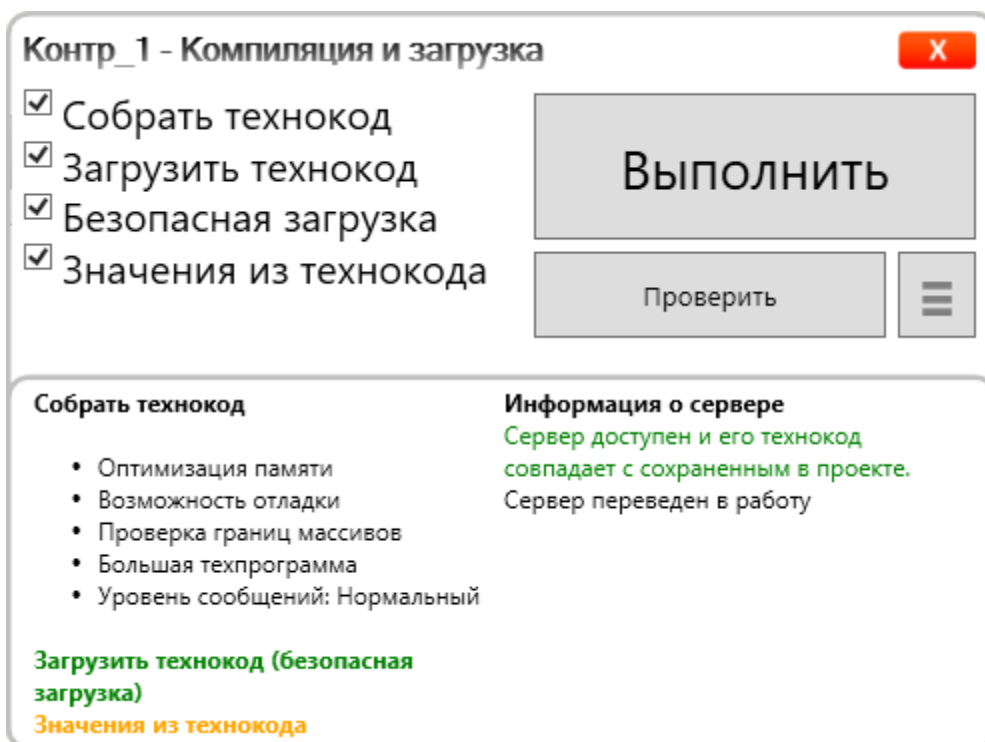


Рисунок 18 – Окно компиляции и загрузки

Для того чтобы проверить техпрограмму на наличие ошибок без компиляции и загрузки, в окне необходимо нажать кнопку проверить.

Перед компиляцией необходимо произвести настройку компилятора (рисунок 19).

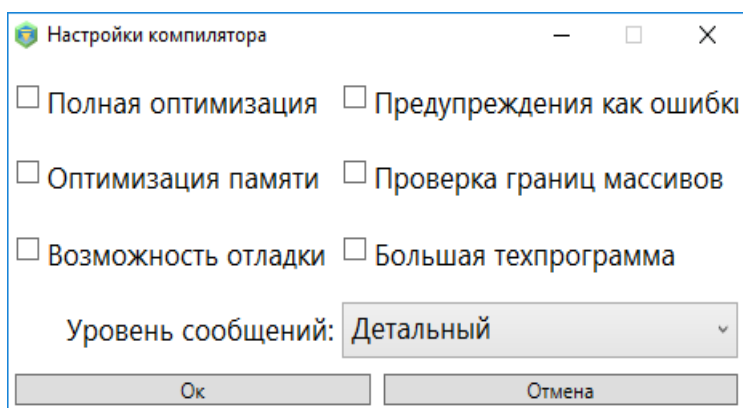


Рисунок 19 – Настройки компилятора

Устанавливаем галочки на опциях, где:

- **Полная оптимизация.** Полная оптимизация технокода по памяти переменных, размеру и быстрдействию технокода. Подходит для релиза технокода для работы на объекте.
- **Оптимизация памяти.** Оптимизация памяти доступных переменных.
- **Возможность отладки.** Добавляет в технокод отладочную информацию, что позволяет отладить технокод на сервере выполнения.
- **Предупреждения как ошибки.** Позволяет компилятору интерпретировать предупреждения как ошибки. Ужесточает требования к чистоте технокода и позволяет избежать скрытых проблем.
- **Проверка границ массивов.** Указывает компилятору выводить код, проверяющий границы при динамическом индексировании массивов. Рекомендуется, если программа содержит модули на ST.

Большая техпрограмма. Позволяет скомпилировать очень большие техпрограммы, которые в обычном режиме не могут быть скомпилированы из-за нехватки оперативной памяти.

- **Уровень сообщений.** Позволяет выводить сообщения с различным уровнем информации.

После настройки компилятора можно приступить к компиляции. Установив галочку «**Собрать технокод**» и нажав кнопку выполнить **Полис** скомпилирует техпрограмму с учётом установленных настроек компилятора. Чтобы загрузить подготовленную технологическую программу в контроллер, рабочая станция с **Полисом** и Эликонт должны находиться в общей сети. Технологическая программа загружается в контроллер целиком, при этом загружаются все подготовленные задачи, не зависимо от того, включены они в **Полисе** или отключены. Чтобы загрузить программу в Эликонт, в окне компиляции и загрузки устанавливается галочка «**Загрузить технокод**» и нажимается кнопка **Выполнить**.

Безопасная загрузка позволяет загружать техпрограмму в оперативном режиме. Если установлена опция «**Значения из технокода**», то значения на входах алгоблоков и значения ручных переменных будут взяты из технокода, развернутого на сервере. Иначе – они будут браться из сохраненного кэша данных.


Примечание. Не рекомендуется использовать опцию «**Значения из технокода**» при загрузке техпрограммы в контроллер, управляющий технологическим процессом.

4.2 Обзор

4.2.1 Возможности обзора

Обзор позволяет просмотреть непосредственно в **Полисе** значения «живых» сигналов, формируемых контроллером. Как и при загрузке, при обзоре рабочая станция с **Полисом** должна находиться в общей сети с контроллером.

Чтобы перевести в режим обзора или вывести из обзора всю задачу, нужно из главного меню выбрать команду соответственно «**Войти в режим обзора**» или «**Выйти из режима обзора**». Обзор задачи означает обзор всех входов и выходов всех алгоблоков выбранной задачи. Отмена обзора задачи означает отмену обзора всех входов-выходов всех алгоблоков данной задачи.

После запуска обзора необходимо установить дополнительный параметр «**Блокировать контроллер во время обзора**» . Если соответствующая опция активирована, другие пользователи не смогут редактировать текущую задачу, пока обзор не будет отменен. Если опция не активна, задача не блокируется.

В режиме обзора рядом с выбранным полем входа (выхода) появится обозначенное зеленым цветом «живое» значение сигнала, полученное непосредственно из контроллера.

4.2.2 Доступ к задаче во время обзора

Прежде чем перейти в режим Обзор, необходимо определить является ли данный обзор блокирующим. Чтобы сделать текущий обзор блокирующим, нужно активировать кнопку «**Блокировать контроллер во время обзора**».

При работе в режиме **Обзор** имеются следующие возможности работы с текущей задачей:

– Если обзор не блокирующий, то в текущей задаче можно выполнять только те операции, которые не влияют на содержание техпрограммы (например, можно перемещать алгоблоки и линии конфигурации, но нельзя изменять алгоритмы, конфигурацию и константы).

– Если обзор блокирующий, то можно выполнять операции, указанные в п.1 и, кроме того, можно выполнять операции, которые вводятся в контроллер без его перезагрузки. К ним относится изменение конфигурации и значений констант (но не изменение алгоблоков).

На п.2 (см. выше) следует обратить особое внимание. Из него следует, что для того, чтобы «на лету» изменять параметры в контроллере, следует перейти в режим обзора и обязательно сделать этот обзор блокирующим.

5 Рекомендации по подготовке техпрограмм

5.1 Информационные ресурсы

При подготовке техпрограмм не накладывается ограничений на максимальное число задач и алгоблоков в одном контроллере. Однако следует учитывать рекомендации по загрузке процессора контроллера, где для оптимальной работы загрузка не должна превышать 80%.

Имеются четыре параметра настройки контроллера, влияющие на его загрузку и быстродействие:

- время цикла техпрограммы;
- частота контроллерной шины;
- период опроса со стороны рабочих станций;
- принцип архивирования.

Правило 1. Даже если загрузка процессора ниже 80 %, время цикла техпрограммы рекомендуется устанавливать максимально возможным, исходя из требований к быстродействию. На рисунке 20 представлена зависимость загрузки процессора от времени цикла техпрограммы.

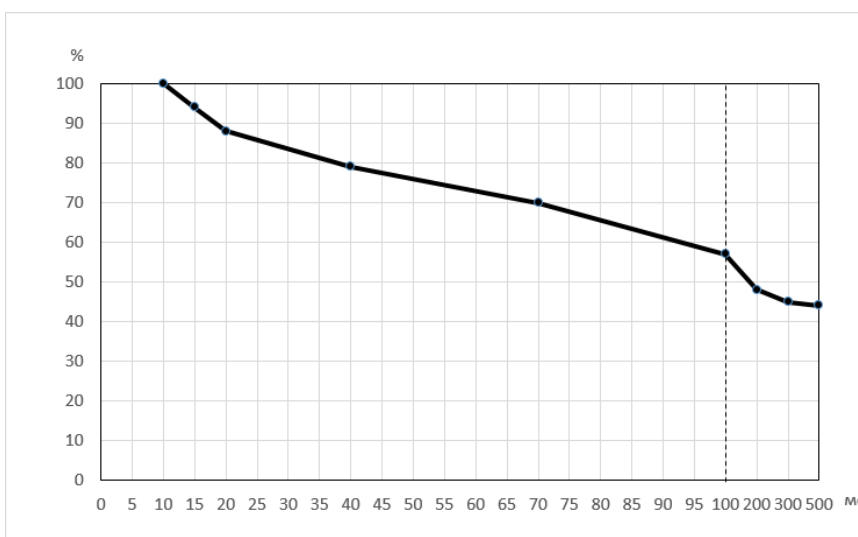


Рисунок 20 – Зависимость загрузки процессора от времени цикла техпрограммы

Правило 2. Если шина не выходит за пределы одного шкафа, частоту шины следует устанавливать равной 5 МГц.

Для удаленных УСО частота шины снижается, при ее выборе следует руководствоваться требованиями стандарта RS-485. Это следует учитывать при оценке быстродействия (чем больше протяженность шины, тем ниже частота и, соответственно, ниже быстродействие).

Правило 3. Период опроса со стороны рабочих станций следует устанавливать максимально возможным, исходя из требований эргономики.

С периодом опроса связано время обновления информации на экране рабочих станций. В нормативных документах для электростанций указано, что это время не должно превышать 0,5 с.

Тем не менее рекомендуется:

- для операторских станций с ручным управлением устанавливать время 0,5 с.
- для обзорного экрана коллективного пользования с большим числом объектов, а также для сервисных рабочих станций устанавливать время 1 с.

Правило 4. Всюду, где это возможно, следует выбирать архивирование по апертуре и использовать максимально допустимое значение апертуры.

Правило 5. Следует избегать ситуации, когда в контроллере фиксируется большое число быстроменяющихся ошибок, – они заметно увеличивают загрузку процессора.

Такое может произойти из-за массовых мерцающих аппаратных отказов или, когда большое число аналоговых технологических параметры постоянно и с высокой частотой флуктуирует в районе предупредительного или аварийного порога (все это маловероятно). В то же время постоянно «висящие» (не меняющиеся) ошибки не влияют на загрузку процессора (если их число меньше 1 тыс.).

5.2 Ввод/вывод информации через модули УСО

Для физического ввода-вывода сигналов в контроллере используются специальные модули – устройства связи с объектом (УСО).

Ограничения по модулям УСО:

- Число блоков УСО в одном контроллере – не более 25 (с учетом расширения).
- Число модулей УСО в пределах одного блока – не более 15 (для АЦП не более 12).

5.3 Межконтроллерный обмен

Осуществляется путём проведения связи из одного контроллера в другой.

Ограничения по межконтроллерному обмену:

– число подключенных к данному контроллеру по сети других контроллеров – не более 10 одиночных или не более 5 дублированных.

5.4 Объектные алгоритмы

5.4.1 Подробнее об объектах

В СУРЕ объект – это совокупность параметров, характеризующих некоторый физический или виртуальный элемент АСУ ТП как единое целое. Эти параметры могут быть представлены на экране операторской станции и/или записаны в архив. Объект обязательно должен иметь марку, уникальную (не повторяющуюся) в пределах одной интегрированной АСУ ТП.

В СУРЕ существует классификация объектов по так называемым оперативным типам. Имеется предопределенный перечень оперативных типов объектов (**Ввод Аналоговый, Регулятор, Задвижка** и т.д.), - всего несколько десятков типов.

В большинстве случаев при работе с **Полисом** определенному типу объекта соответствует какой-либо объектный алгоритм такие объекты называются простыми. Однако, не все типы объектов «покрываются» одним алгоритмом, - для того, чтобы сформировать объектную информацию некоторого числа специфических объектов, необходимо привлечь несколько библиотечных (как объектных, так и не объектных) алгоритмов, определенным образом сконфигурированных. Такие объекты называются составными. Ниже рассматриваются общие свойства объектных алгоритмов, а также приводятся конфигурации составных объектов.

5.4.2 Общие свойства объектных алгоритмов

Помимо своих «прямых» функций, связанных с работой технологической программы, объектные алгоритмы выполняют ряд информационных функций:

Они формируют совокупную информацию о значениях параметров и текущем состоянии объекта (открыто/закрыто, автоматический/ручной, достоверность и т.д.). Эти сведения передаются по запросам операторской станции и отображаются на мнемосхемах или оперативных окнах.

Они направляют в архив информацию о появлении или исчезновении штатных событий (изменении состояний, режимов и т.п.) и появлении или исчезновении нештатных событий, т.е. ошибок.

Они направляют рабочим станциям информацию о текущих нештатных событиях, т.е. ошибках (недостоверности параметров, невыполнении условий и т.п.). Эту информацию может принять любая станция, в которой предусмотрена процедура представления ошибок (к таким станциям, в частности, относится операторская и событийная станции).

Специально подготавливать в Эликонте оперативную информацию о таких объектах не требуется, - после того, как алгоритм с помощью Полиса привязан к марке объекта, вся необходимая информация готова для передачи станциям (это утверждение не касается периодической записи в архив).

5.5 Вычислительные алгоритмы

5.5.1 Состав вычислительных алгоритмов

Библиотека Эликонтов содержит целый ряд алгоритмов, с помощью которых можно выполнять аналоговые и дискретные вычисления. Для аналоговых статических и динамических вычислений используются алгоритмы суммирования, масштабирования, извлечения корня, интегрирования и т.п. Дискретные вычисления выполняются с помощью алгоритмов логического И, ИЛИ, различного рода триггеров и т.д.

5.6 Архивирование сигналов

5.6.1 Варианты архивирования

В СУРЕ реализованы 2 метода архивирования сигналов:

- периодическое
- апертурное

Периодическое архивирование сигналов выполняется регулярно с заданным периодом времени. При апертурном архивировании сигналы записываются в архив только при их изменении на величину больше апертурности.

5.6.2 Периодическое архивирование сигналов

При периодическом архивировании значения сигналов записываются в архив с заданным периодом не зависимо от того, насколько изменился сигнал в течение этого периода.

По умолчанию периодическое архивирование отключено. Для того, чтобы выбранный сигнал записывался в архив, необходимо принудительно изменить значение атрибута **Период** в **Базисе**.

Архивироваться могут сигналы, сформированные любым как объектным, так и не объектным алгоритмом. При этом следует иметь ввиду, что периодически архивировать имеет смысл в основном аналоговые сигналы. Дискретные признаки объектных алгоритмов «автоматически» архивируются в виде событий, а для дискретных сигналов не объектных алгоритмов можно такие события запрограммировать

5.6.3 Архивирование аналоговых сигналов по апертуре

При апертурном архивировании аналоговый сигнал записывается в архив, только если он изменился на определенную величину (аперттуру) относительно значения, ранее записанного в архив.

Архивирование по апертуре уменьшает объем архива и снижает нагрузку на информационную сеть. В то же время, если архивируемые по апертуре сигналы редко выходят за установленные апертурные значения, в архиве будет слишком мало записей и их неудобно будет анализировать. Поэтому апертурный метод рекомендуется сочетать с периодической записью в архив тех же сигналов, при этом период записи можно выбрать достаточно большим (однако, не настолько, чтобы потерять наглядность представления трендов).

5.6.4 Объектное архивирование

Большинство объектных алгоритмов формируют события, которые «автоматически» (без специального программирования) направляются в архив. Например, в архив автоматически направляются события **Задвижка открылась (закрылась)**, **Двигатель включился (отключился)** и т.д. Аналогично в архив записываются и все ошибки, фиксируемые объектными алгоритмами. Подобные записи в архив инициирует сам контроллер.

Лист регистрации изменений

Изм.	Номера листов (страниц)				Всего листов (страниц) в докум.	№ докум.	Вх. № сопроводительного документа и дата	Подпись	Дата
	измененных	замененных	новых	аннулированных					