

ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ  
"ТСА-Сервис"



ОКПД 2 26.51.70.190



УТВЕРЖДАЮ  
Генеральный директор  
ООО «ТСА-Сервис»  
\_\_\_\_\_ Петров С.В.  
«01» ноября 2018 г.

**Комплекс программно-технический Квинт-6**

Система технологического программирования контроллеров  
Пилон.

Руководство пользователя

ПФДИ.421457.009 И3.2

Москва

2018

Инв. № подл.	Подп. и дата	Взаим. инв. №	Инв. №	Подп. и дата

# Содержание

<b>1</b>	<b>Введение.....</b>	<b>6</b>
1.1	Терминология .....	6
1.2	Дополнительные материалы для работы с Пилоном.....	7
<b>2</b>	<b>Общие сведения о Пилоне.....</b>	<b>7</b>
2.1	Приступая к работе .....	7
2.2	Как структурировать проект .....	8
2.3	Что делает Пилон.....	8
2.4	Что еще может делать Пилон .....	9
2.4.1	Компиляция и загрузка .....	9
2.4.2	Обзор выполнения технологической программы.....	9
2.4.3	Обзор приборных параметров контроллера и буфера ошибок.....	10
2.4.4	Блокировка.....	10
2.4.5	Перестановка.....	10
2.4.6	Отключение .....	10
<b>3</b>	<b>Работа с Пилоном .....</b>	<b>11</b>
3.1	Главное меню .....	11
3.1.1	Состав главного меню .....	11
3.1.2	Меню Файл.....	11
3.1.3	Меню Правка .....	16
3.1.4	Меню Поиск.....	16
3.1.5	Меню Вид.....	17
3.1.6	Меню Техпрограмма .....	17
3.1.7	Меню Окна .....	18
3.1.8	Меню «?».....	18
3.2	Шкала инструментов .....	18
3.3	Настройки Пилона .....	19
3.3.1	Настройки «Выделение при движении мыши».....	19
3.3.2	Настройка «Блокировать контроллер во время обзора» .....	20
3.3.3	Настройки «Атрибуты алгоблока по умолчанию» .....	20
3.3.4	Настройки «Комментарии».....	20
3.3.5	Настройки «Параметры копирования алгоблоков» .....	20
3.3.6	Настройки «Цвет выбранного алгоблока и цвет фона» .....	20

Подп. и дата	
Инв. №	
Взаим. инв. №	
Подп. и дата	
Инв. № подл.	

					ПФДИ.421457.009 ИЗ.2			
Изм.	Лист	№ докум	Подп.	Дата				
Разраб.	Туркин				<b>Комплекс программно-технический Квинт-6.</b> Система технологического программирования контроллеров Пилон. Руководство пользователя.	Лит	Лист	Листов
Пров.	Зарипов						2	76
Н.контр	Бочаров					ООО «ТСА-Сервис»		
Утверд.	Петров							

3.3.7	Настройка «Запрашивать подтверждения»	21
3.3.8	Настройка «Сохранять состояния окон»	21
3.3.9	Настройка «Однооконный режим»	21
3.3.10	Настройка «Полис»	21
3.3.11	Настройка «Шрифт Полиса»	21
3.3.12	Настройки «По умолчанию»	21
3.4	Свойства задач	21
3.4.1	Список задач	21
3.4.2	Главное окно Пилона	22
3.4.3	Поле задачи	23
3.4.4	Размещение алгоблоков	24
3.4.5	Реквизиты алгоблоков	25
3.4.6	Входы-выходы алгоблока	26
3.4.7	Дополнительные атрибуты	26
3.4.8	Копирование алгоблоков	26
3.4.9	Свойства входов алгоблоков	28
3.5	Константы	29
3.5.1	Общий случай установки констант	29
3.5.2	Установка векторных констант	30
3.5.3	Установка констант в физических единицах	30
3.5.4	Уникальность констант	30
3.5.5	Особенности установки констант	31
3.6	Конфигурирование алгоблоков	33
3.6.1	Техника конфигурирования	33
3.6.2	Ссылки	33
3.6.3	Дополнительные возможности конфигурирования	34
3.6.4	Атрибуты линии связи	35
3.6.5	Пересечение линий связи	36
3.7	Перемещение и перестановка алгоблоков	36
3.7.1	Возможности перемещения	36
3.7.2	Наложение алгоблоков и ссылок	37
3.7.3	Перестановка алгоблоков	37
3.8	Комментарии	37
3.8.1	Виды комментариев	37
3.8.2	Комментарии к задаче	38
3.8.3	Другие виды комментариев	38

Инв. № подл.	Подп. и дата	Взаим. инв. №	Инв. №	Подп. и дата

3.9	Привязка .....	38
3.9.1	Назначение привязки .....	38
3.9.2	Отображение привязки .....	38
3.9.3	Соответствие алгоритмов оперативным типам объектов .....	39
3.9.4	Технология привязки .....	40
3.10	Права доступа и коллективная работа с Пилоном .....	41
3.10.1	Права доступа .....	41
3.10.2	Коллективная работа .....	42
<b>4</b>	<b>Отладка технологических программ.....</b>	<b>42</b>
4.1	Компиляция и загрузка технологической программы .....	42
4.1.1	Виды компиляции .....	42
4.1.2	Обычная компиляция и загрузка.....	43
4.1.3	Компиляция с обновлением .....	43
4.1.4	Загрузка .....	44
4.2	Обзор .....	44
4.2.1	Возможности обзора .....	44
4.2.2	Доступ к задаче во время обзора .....	45
<b>5</b>	<b>Рекомендации по подготовке технологических программ .....</b>	<b>45</b>
5.1	Информационные ресурсы .....	45
5.2	Ввод/вывод информации через модули УСО .....	46
5.3	Связь между контроллерами .....	46
5.3.1	Источник и приемник принадлежат одному проекту АСУ ТП и находятся внутри одного системного модуля .....	49
5.3.2	Источник и приемник принадлежат разным проектам АСУ ТП .....	49
5.3.3	Источник – одиночный или резервированный контроллер, приемник – кластер.....	49
5.3.4	Источник – кластер, приемник – одиночный или резервированный контроллер.....	49
5.3.5	Источник и приемник – кластеры .....	49
5.4	Каскадные связи .....	50
5.5	Объектные алгоритмы .....	50
5.5.1	Подробнее об объектах .....	50
5.5.2	Общие свойства объектных алгоритмов.....	51
5.5.3	Неявные входы объектных алгоритмов .....	51
5.6	Вычислительные алгоритмы .....	52
5.6.1	Состав вычислительных алгоритмов .....	52
5.6.2	Применение алгоритма УНО (Универсальные операции) .....	52
5.6.3	Применение алгоритма ПКП .....	53

Инв. № подл.	Подп. и дата
Взаим. инв. №	Подп. и дата
Инв. №	Подп. и дата
Подп. и дата	Подп. и дата

5.7	Архивирование сигналов .....	54
5.7.1	Варианты архивирования.....	54
5.7.2	Периодическое архивирование сигналов .....	54
5.7.3	Архивирование аналоговых сигналов по апертуре.....	56
5.7.4	Объектное архивирование .....	56
5.7.5	Программирование событий .....	57
5.7.6	Программирование ошибок.....	58
5.8	Работа с кластерными контроллерами .....	59
5.8.1	Подготовка технологических программ для кластерных контроллеров .....	59
5.8.2	Режим обзора для кластеров.....	59
<b>6</b>	<b>Подготовка логических шаговых программ .....</b>	<b>60</b>
6.1	Шкала инструментов Полиса.....	62
6.2	Структура логической программы.....	62
6.3	Фрагменты шага .....	63
6.3.1	Команда.....	63
6.4	Таймер.....	64
6.4.1	Условия .....	64
6.4.2	Переходы .....	64
6.5	Последовательность действий до начала программирования .....	65
6.6	Техника программирования логических программ .....	65
6.6.1	Конфигурирование программных выходов.....	66
6.6.2	Запись текста программы без условий .....	67
6.6.3	Запись текста программы с условиями.....	69
6.7	Общие ограничения логических программ.....	71
6.8	Анализ и редактирование программы .....	72
6.9	Отладка логической программы.....	73
6.9.1	Режимы работы программы .....	73
6.9.2	Обзор программы.....	73
<b>7</b>	<b>Подготовка отчета.....</b>	<b>74</b>
7.1	Проектирование форматов.....	74
7.2	Предварительный просмотр форматов и печать.....	75
	<b>Лист регистрации изменений .....</b>	<b>76</b>

Инв. № подл.	Подп. и дата	Взаим. инв. №	Инв. №	Подп. и дата

# 1 Введение

Система технологического программирования Пилон (в дальнейшем по тексту – Пилон) предназначена для подготовки пользовательских технологических программ контроллеров (фирменное название контроллера - Ремиконт), входящих в программно-технический комплекс Квинт-6 (в дальнейшем по тексту - Квинт).

Пилон - это программный пакет, входящий в группу САПР фирменного программного обеспечения Квинта. Все фирменное программное обеспечение Квинта объединено программной оболочкой Квинтегратор, из которой загружаются различные программные компоненты, в том числе и Пилон

В Квинтеграторе Пилон входит в папку **Проектирование**.

При работе с Пилоном используется библиотека алгоритмов Ремиконтов, описание которой «встроено» в Пилон и может быть вызвано на экран монитора станции проектирования.

## 1.1 Терминология

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

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

**Алгоблок** – алгоритм, введенный в поле задачи.

**Алгоритм** – созданная разработчиками Квинта фиксированная совокупность функций, имеющая имя и входящая в библиотеку алгоритмов Ремиконта.

**Аркада** – система управления БД проекта АСУ ТП. С помощью Аркады 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 *Проектирование* оболочки Квинтегратор.

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

**Библиотека алгоритмов** - Фирменная **библиотека алгоритмов** является основой для создания технологических программ. В состав библиотеки входят специальные алгоритмы и алгоритмы универсального назначения. Описание библиотеки встроено в Квинтегратор (см. последний раздел дерева приложений «**Справка/Подсистема управление (Ремиконты)**» ).

**Блокировка** – динамически устанавливаемый признак, временно запрещающий редактирование отдельных элементов базы данных определенным пользователям. Блокировка сохраняет целостность базы данных при многопользовательском доступе и обеспечивает возможность коллективной работы над одним проектом.

**Системный модуль** – сетевая группа, состоящая из Шлюза и связанных с ним по сети контроллеров. Обычно проект управляющей части АСУ ТП содержит несколько системных модулей.

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

**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 Пилона соответствует сравнительно автономной технологической задаче управления.

**Кластер** – организация резервирования контроллеров для создания в АСУ ТП подсистем технологических защит

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

**Константа** – значение сигнала на свободном входе.

Инва. №				
Подп. и дата				
Инва. № подл.				

**Конфигурация** – система связей между алглоблоками.

**Многоканальный алгоритм** – алгоритм, имеющий несколько независимых каналов, каждый из которых выполняет однотипные функции. Некоторые многоканальные алгоритмы имеют общую часть, относящуюся одновременно ко всем каналам.

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

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

**Отключенная задача** – задача, у которой все выходы всех алглоблоков заморожены, а в некоторых случаях обнулены.

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

**Подзадача** – задача, иерархически входящая в другую задачу.

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

**Полис** – компонента Пилона для программирования логических шаговых программ

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

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

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

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

## 1.2 Дополнительные материалы для работы с Пилоном

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

- «Комплексы программно-технические Квинт-6. Часть 1 Структура и системная интеграция. Руководство по эксплуатации ПФДИ.421457.003 РЭ1»;
- «Комплексы программно-технические Квинт-6. Система управления технологической базой данных Аркада. Руководство пользователя ПФДИ.421457.009 ИЗ. 4»;
- «Комплексы программно-технические Квинт-6. Администрирование проектов АСУ ТП. Руководство пользователя ПФДИ.421457.009 ИЗ. 3».

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

## 2 Общие сведения о Пилоне

### 2.1 Приступая к работе

Перед тем, как начать работать с Пилоном, необходимо произвести последовательно следующие действия:

- 1 Проинсталлировать на компьютере требуемые компоненты Квинтегратора (по крайней мере Пилон). Если работа ведется на автономной рабочей станции, то помимо Пилона следует проинсталлировать Администратор и Аркаду. Если база данных (БД) хранится на отдельной рабочей станции (сервере БД), то компьютер с Пилоном должен быть объединен в общую сеть с сервером БД и в Администраторе БД должен быть соответствующим образом «прописан» компьютер с Пилоном.

Инв. № подл.	Подп. и дата
Взаим. инв. №	Инв. №
Подп. и дата	Подп. и дата

- 2 На один из компьютеров в общей сети установить ключ защиты, соответствующий приобретенной лицензии на использование Квинта. Если компьютер с Пилоном не включен в сеть, этот ключ устанавливается непосредственно на компьютер с Пилоном. Если работа с Пилоном ведется в автономном режиме (без контроллеров), ключ не требуется.
- 3 Средствами Администратора ввести описание всех Шлюзов и связанных с ними Ремиконтов (Системных модулей), относящихся к данному проекту (т.е. входящих в БД проекта), а также всех рабочих станций, на которых ведется работа с Пилоном.
- 4 Средствами Администратора ввести описание всех системных модулей и всех Ремиконтов, относящихся к другим проектам, с которыми необходим информационный обмен Ремиконтов данного проекта. Аналогично в БД других проектов надо ввести описание Ремиконтов данного проекта.
- 5 Через Аркаду 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 Пилоном можно и без заполнения БД объектами (или с неполным заполнением БД), но тогда к Пилону 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 Квинте 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 Квинте следующим образом:

- общие управляющие функции распределяются по системным модулям, при этом каждому системному модулю дается технологическое имя и сетевой номер. Реквизиты системного модуля соответствуют реквизитам Шлюза;
- функции каждого системного модуля распределяются по контроллерам, входящим в данный системный модуль. Каждому контроллеру присваивается технологическое имя и сетевой номер. Если проект включает защитную подсистему, то в состав системного модуля включается защитный кластер со своим именем и сетевым номером, а в кластер включаются входящие в него защитные контроллеры со своими именами и сетевыми номерами;
- 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 с учетом единственного ограничения: **суммарное число всех задач и подзадач одного контроллера не должна превышать 256**;
- состав системных модулей и контроллеров задается в Администраторе, а состав и структура задач и подзадач – непосредственно в Пилоне.

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 Пилон

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

- выбирается требуемый системный модуль, а в нем - требуемый контроллер. Для подсистемы защиты вместо контроллера выбирается кластер;
- для выбранного контроллера создается (возможно иерархический) перечень задач. Выбирается требуемая задача, - на экране открывается поле задачи;
- из библиотеки алгоритмов Ремиконта 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);

Инва. № подл.	Подп. и дата	Взаим. инв. №	Инва. №	Подп. и дата

Изм	Лист	№ докум	Подп.	Дата	ПФДИ.421457.009 И3.2	Лист
						8

- 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, в рабочих станциях ИВК (операторской, архивной, событийной и т.д.).

Таким образом, технологическая программа готовится для конкретного выбранного контроллера, а применительно к подсистеме защиты – для выбранного кластера.

## 2.4 Что еще может делать Пилон

### 2.4.1 Компиляция и загрузка

После того, как контроллер с помощью Шлюза Ш-81 подключился к информационному пространству подготовленную для него средствами Пилона 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.

Пилон позволяет использовать два варианта компиляции:

- 1 **С обновлением.** При загрузке техпрограммы, скомпилированной этим вариантом, состояние всех выходов и всех режимов всех алгоблоков данного контроллера принимают значения по умолчанию (т.е. значения, указанные в описании библиотечных алгоритмов).
- 2 **Без обновления.** В этом случае там, где это возможно, текущее (накопленное) состояние алгоблоков данного контроллера сохраняется. Слово «возможно» в этом контексте означает, что не всегда состояние алгоблока может быть сохранено. Например, если в процессе редактирования технологической программы алгоблок был удален, то очевидно, что говорить о сохранении его параметров не приходится.

Общие правила здесь такие:

- состояние может быть сохранено, если алгоблок не удалялся или в нем не изменялся модификатор. Если на входе алгоритма изменялась константа или конфигурация, то после загрузки в первый момент состояние выходов будет сохранено, но затем оно примет значение, соответствующее новым параметрам на входе;
- загрузка с обновлением обычно используется, когда нужно «погасить» всю предысторию работы контроллера. И наоборот, загрузка без обновления используется при частичной модификации проекта, когда требуется сохранить текущее состояние алгоритмов контроллера;
- в одной процедуре загрузки программа загружается в один выбранный контроллер, при этом *загружаются все задачи технологической программы (как включенные, так и отключенные)*. Для дублированного и защитного контроллера одна и та же технологическая программа с помощью одной команды загружается сразу соответственно в основной и резервный контроллер или во все контроллеры одного выбранного кластера.

### 2.4.2 Обзор выполнения технологической программы

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

В специальном режиме, называемом **Обзор**, непосредственно в Пилоне 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 Пилона можно послать Ремиконту 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.

Инва. № подл.	Подп. и дата	Инва. №	Подп. и дата
Взаим. инв. №			

Изм	Лист	№ докум	Подп.	Дата	ПФДИ.421457.009 И3.2	Лист
						9

Очевидно, что обзор будет работать, только если рабочая станция с Пилоном подключена к общей сети, к которой, в свою очередь, подключен соответствующий контроллер с загруженной технологической программой. Более подробно возможности обзора описаны ниже в разделе 4.

### 2.4.3 Обзор приборных параметров контроллера и буфера ошибок

Для выбранного контроллера Пилон 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 Пилоне предусмотрен механизм блокировок, который, в зависимости от ситуации, распространяется на весь контроллер или отдельные входящие в него задачи.

Признаки блокировки контроллера и задачи отображаются в нижней части главного окна Пилона (см. ниже).

Подробнее свойства блокировок описаны в 3.10.

### 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 Пилоне 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 Пилоне.

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.

Более того, Пилон позволяет не только изменять порядок выполнения, но и модифицировать созданную структуру задач, - перемещать подзадачи из одних задач в другие, а отдельные задачи делать подзадачами других задач. Это означает, что в принципе не обязательно сразу создавать требуемую иерархию задач, - можно последовательно создать и программировать одну задачу за другой, формируя «линейный» (не иерархический) список всех задач, а потом «вставить» мелкие задачи в более крупные, те – в еще более крупные и т.д. В процессе такой перестановки вся структура алгоблоков, относящаяся к тем или иным задачам, остается без изменения.

### 2.4.6 Отключение

В процессе отладки часто полезно отключить отдельные задачи технологической программы. Отключенные задачи в контроллере не выполняются, а выходные сигналы всех входящих в них алгоблоков замораживаются (в некоторых случаях обнуляются). Пилон позволяет избирательно отключать как отдельную подзадачу самого нижнего уровня, так и задачу более высокого уровня со всеми входящими в нее подзадачами.

Правило здесь такое: если задача любого уровня отключена, то отключенными считаются все входящие в нее подзадачи.

Однако, обратное не верно, - если включается какая-либо задача, это не означает «автоматического» включения всех входящих в нее подзадач. Если они были индивидуально отключены, они останутся отключенными, если индивидуально не отключались, то включатся вместе с вышестоящей задачей.

Инд. № подл.	Подп. и дата	Взаим. инв. №	Инд. №	Подп. и дата

## 3 Работа с Пилоном

### 3.1 Главное меню

#### 3.1.1 Состав главного меню

Главное меню Пилона имеет следующие разделы:

- **Файл;**
- **Правка;**
- **Поиск;**
- **Вид;**
- **Технологическая программа;**
- **Окна;**
- **«?».**

При выборе любого из этих разделов открывается ниспадающее меню, в котором расположены команды второго уровня. Ниже состав главного меню описывается более подробно.

#### 3.1.2 Меню Файл

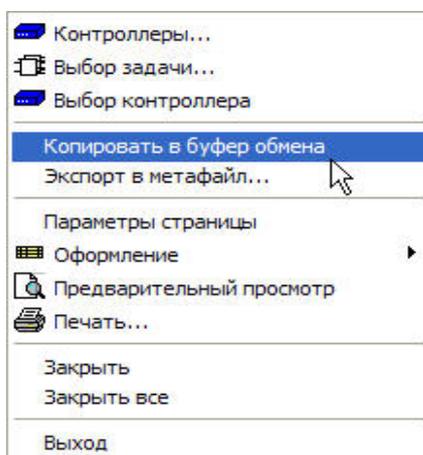


Рисунок 1 – Меню Файл

Команда «**Контроллеры...**» (рисунок 1) вызывает окно, в левую часть которого выводится структура Шлюзов и контроллеров проекта в соответствии с их описанием в **Администраторе**, а в закладке **Оперативные** выводится информация о параметрах контроллера, установленных в **Базе данных** (рисунок 2).

После включения режима **Обзор параметров и буфера ошибок** программа запрашивает эти параметры у контроллера (или контроллеров при резервированном или кластерном исполнении Ремиконта) и выводит их на экран (поля **Основной** и **Резервный** на рисунке 2) с индикацией красным шрифтом параметров, несовпадающих по значениям с данными БД.

При работе в обзоре оперативных параметров могут возникнуть следующие нештатные ситуации:

- **Контроллер не отвечает.** В этом случае заголовок соответствующего контроллера мигает красным цветом. При наведении мыши на заголовок появляется текст с уточняющей причиной;
- **Текущий уровень отказа.** Если в контроллере имеется отказ любого уровня (в том числе нулевого), текст соответствующего поля становится красным;
- **Время выполнения.** Если текущее время выполнения техпрограммы становится больше заданного в БД, текст соответствующего поля становится красным;
- **Температура в шкафу.** Если температура в шкафу выше допустимой, текст соответствующего поля становится красным. Если эта температура недостоверна, перед значением ставится символ @ и значение также выводится красным цветом;
- **Часы.** Если часы в контроллере ни разу не устанавливались, в поле красным цветом выводится текст «Не устанавливались»;

Инв. № подл.	Подп. и дата
Взаим. инв. №	Инв. №
Подп. и дата	Подп. и дата

- **Параметры БД и контроллера не совпадают.** Если между параметрами, прописанными в БД и полученными из контроллера, имеются отличия, текст соответствующего поля контроллера становится красным.

При выборе какого либо контроллера, закладка **Ресурсы** вызывает окно с информацией об использованных в технологической программе ресурсов контроллера (рисунок 3).

При выборе какого-либо контроллера (кроме Рем.-390) закладка УСО вызывает окно с перечнем типов модулей УСО, прописанных в базе данных выбранного контроллера (рисунок 4).

При выборе какого либо контроллера закладка **Ошибки** вызывает окно с перечнем технологических и приборных ошибок, зарегистрированных системой диагностики контроллера и хранящихся в его буфере ошибок (рисунок 5). Необходимо помнить, что буфер рассчитан на 44 ошибки. Если их количество превышает его объем, то часть ошибок не выводится и выдается сообщение: **«Буфер ошибок переполнен»**.

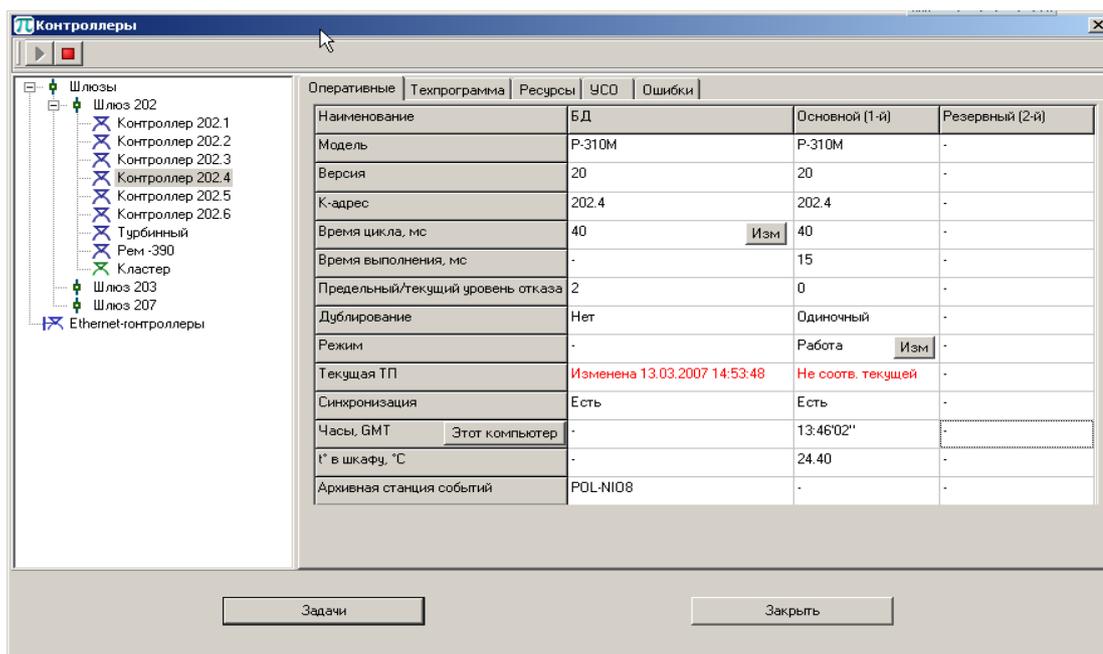


Рисунок 2 - Окно Контроллеры. Закладка Оперативные

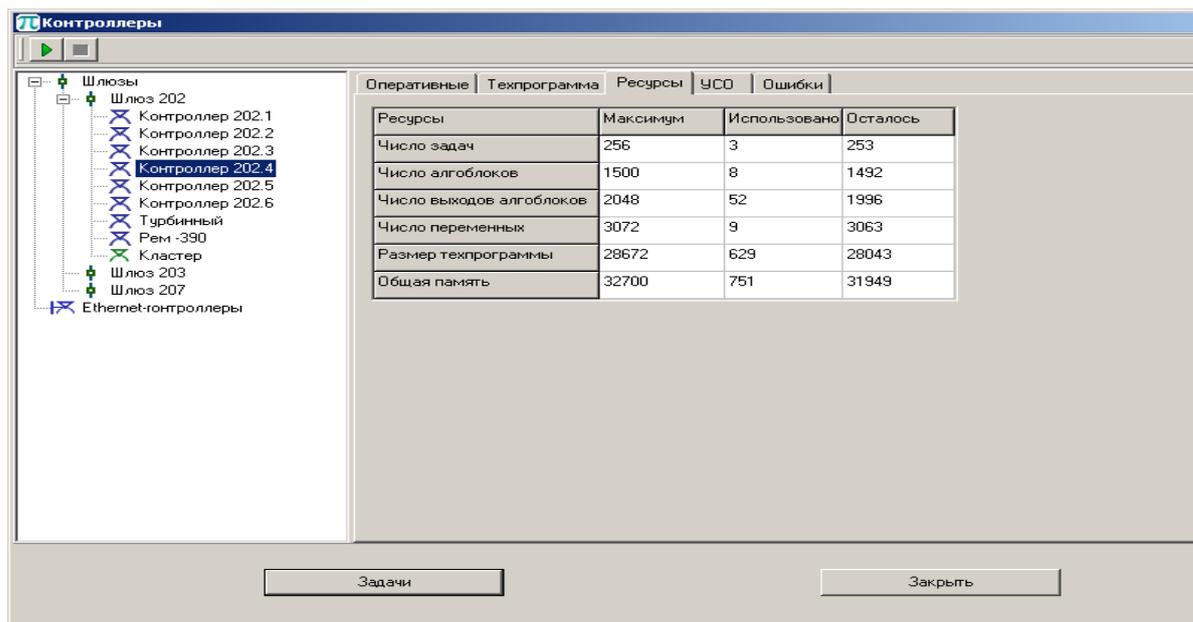
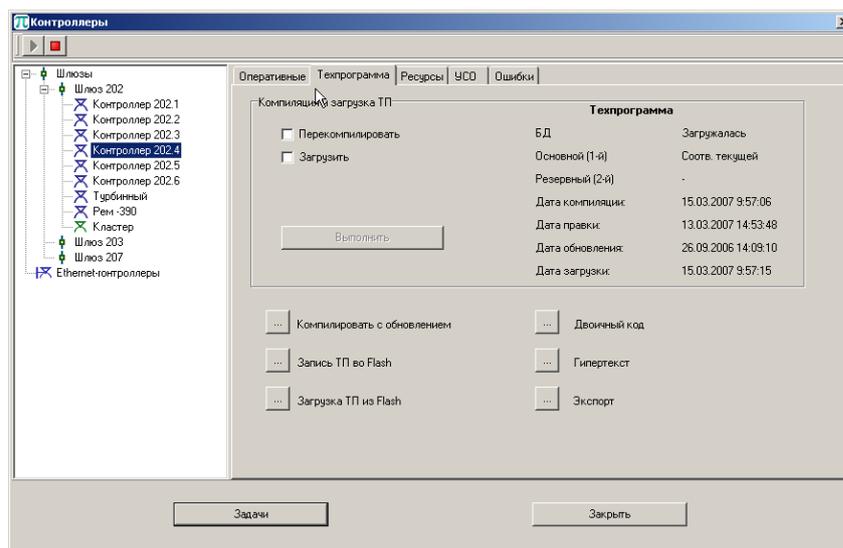


Рисунок 3 – Окно Контроллеры. Закладка Ресурсы

Инв. №	Подп. и дата
Взаим. инв. №	
Подп. и дата	
Инв. № подл.	

Изм	Лист	№ докум	Подп.	Дата
-----	------	---------	-------	------





**Рисунок 6 - Окно Контроллеры. Зкладка Технологическая программа**

Во всех окнах нажатие клавиши **Задачи** вызывает окно выбора задач технологических программ.

Закладка **Техпрограмма** открывает в окне дополнительное поле (рисунок 6) для действий с подготовленной технологической программой.

По указателю **«Перекompиллировать»** и команде **«Выполнить»** формируется файл технологической программы для последующей загрузки в контроллер (загрузочный файл). В процессе этой операции технологическая программа, подготовленная в Пилоне, преобразуется в специальный формат, «понимаемый» Ремиконтом. По этой команде выполняется компиляция без обновления состояния алгоритмов в технологической программе.

По указателю **«Компилировать с обновлением»** и команде **«Выполнить»** так же формируется файл техпрограммы для последующей загрузки в контроллер. Отличие этой команды от указателя **«Перекompиллировать»** заключается в том, что в процессе загрузки выполняется обновление состояния алгоритмов техпрограммы до исходного.

По указателю **«Загрузить»** и команде **«Выполнить»** технологическая программа загружается в контроллер. Для кластера предусмотрена функция синхронизации данных в элементах кластера. Более подробно процедуры компиляции и загрузки рассмотрены в разделе 4.

Указатели **«Двоичный код»** и **«Гипертекст»** позволяют просмотреть подготовленную технологическую программу в виде двоичного файла или в HTML-формате соответственно. Этот просмотр используется разработчиками Пилона.

По указателю **«Экспорт»** загрузочный файл копируется на диск компьютера с расширением *tps*. Это так же используется разработчиками Пилона.

В это окно выводятся также календарные даты последних действий с технологической программой.

Указатель **«Синхронизировать данные»** появляется только для кластерных Ремиконтов. По нему технологическая программа и состояния алгоритмов, которые они приобретают в процессе выполнения программы, передаются из одного контроллера кластера в другой. Выбор контроллера-источника и контроллера-приемника осуществляется в дополнительном диалоговом окне. Синхронизация необходима для случая ремонта кластера при работающем объекте управления. Более подробно об этом см. в документе «Комплексы программно-технические Квинт-6. Многоцелевой контроллер Ремиконт Р-380. Руководство по эксплуатации ПФДИ.421457.001 РЭ».

Команда **«Выбор контроллера»** отличается от вышеописанной команды **«Контроллеры»** тем, что в дополнительное окно выводится только структура Шлюзов и контроллеров проекта в соответствии с их описанием в Администраторе. Нажатие клавиши **Задачи** вызывает окно выбора в технологической программе задач

Команда **«Выбор задачи»** позволяет сразу перейти к окну выбора задач технологической программы того контроллера, который был активным при предыдущем закрытии Пилона.

Инд. № подл.	Подп. и дата
Взаим. инв. №	Инд. №
Подп. и дата	Подп. и дата

Команда **«Копировать в буфер обмена»** позволяет скопировать активную задачу в буфер обмена Windows. После этого она может быть вставлена в любое приложение, поддерживающее работу с буфером обмена (графический редактор, Word и т.д.).

Команда **«Экспорт в метафайл»** сохраняет активную задачу в виде отдельного файла формата emf. После выбора этой команды открывается диалоговое окно, в котором указывается место размещения сохраняемой задачи. Сохраненный файл можно затем загрузить в любой графический редактор, работающий с векторным форматом (Illustrator, Corel Draw, ASDSee и т.п.).

Команда **«Параметры страницы»** активна при условии открытой какой либо задачи. При выборе данной опции появляется диалоговое окно настройки форматов, определяющее размеры «листа» задачи а так же ориентацию: книжная или альбомная.

Команда **«Оформление»** открывает подменю выбора редактируемого шаблона (Титульный лист проекта \\  
Титульный лист задачи \\  
страница задачи. Более подробно см. в разделе 7).

Команда **«Предварительный просмотр»** позволяет посмотреть форматки задачи в том виде, в каком они будут напечатаны принтером.

Команда **«Печать»** включает распечатку форматок задачи на принтере.

Команды **«Закреть»** и **«Закреть все»** закрывают соответственно текущую задачу и все открытые задачи (их может быть несколько).

Команда **«Выход»** закрывает Пилон.

Инд. № подл.	Подп. и дата	Взаим. инв. №	Инв. №	Подп. и дата

Изм	Лист	№ докум	Подп.	Дата	ПФДИ.421457.009 И3.2	Лист
						15

### 3.1.3 Меню Правка

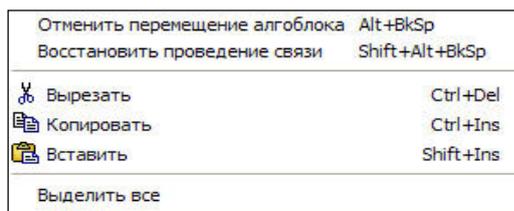


Рисунок 7 – Меню Правка

При выборе меню **Правка**, открывается окно, изображенное на рисунке 7.

Команда **«Отменить перемещение алглобка»** отменяет последнее действие пользователя, связанное с перемещением алглобка в задаче.

Команда **«Восстановить проведение связи»** восстанавливает удаленную линию связи.

Команда **«Вырезать»** удаляет выделенный алглобок и помещает его в буфер обмена.

Команда **«Копировать»** копирует выделенный алглобок в буфер обмена.

Команда **«Вставить»** вставляет алглобок из буфера обмена в задачу.

Команда **«Выделить все»** выделяет все алглобки задачи.

### 3.1.4 Меню Поиск

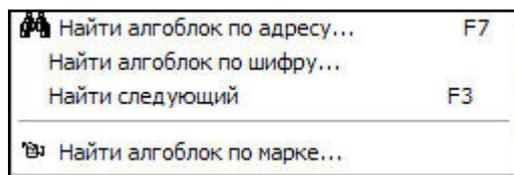


Рисунок 8 – Меню Поиск

При выборе меню **Поиск**, открывается окно, изображенное на рисунке 8.

Команда **«Найти алглобок по адресу...»** обеспечивает поиск алглобка во всех задачах техпрограммы. Адрес искомого алглобка указывается в дополнительном окне, вызываемом по этой команде. По результатам поиска открывается задача с найденным алглобком, а «шапка» алглобка выделена синим цветом.

Команда **«Найти алглобок по шифру...»** обеспечивает поиск алглобка во всех задачах техпрограммы. Библиотечный шифр или порядковый номер в библиотеке искомого алглобка указывается в дополнительном окне, вызываемом по этой команде. По результатам поиска открывается задача, в которой присутствует искомым алглобок с наименьшим адресом. а «шапка» алглобка выделяется цветом, установленным в настройках (о настройках см. ниже).

Команда **«Найти следующий»** обеспечивает поиск алглобка, номер которого на единицу больше последнего найденного алглобка при поиске по адресу или по шифру (но не по марке).

Команда **«Найти алглобок по марке...»** обеспечивает поиск алглобка во всех контроллерах проекта. Все привязанные марки данного контроллера для выбора нужной показываются в дополнительном окне. В этом же окне марку для поиска можно ввести с помощью клавиатуры. По результатам поиска открывается задача, в которой присутствует искомым алглобок, привязанный к этой марке, «шапка» алглобка выделяется цветом, установленным в настройках. Команда доступна сразу после запуска Пилона. В случае, если искомым объект находится не в выбранном контроллере, все открытые задачи закрываются, текущим становится контроллер, в котором находится этот объект, открывается нужная задача и выделяется соответствующий алглобок.

Инва. № подл.	Подп. и дата
Взаим. инв. №	Инва. №
Подп. и дата	Подп. и дата

### 3.1.5 Меню Вид

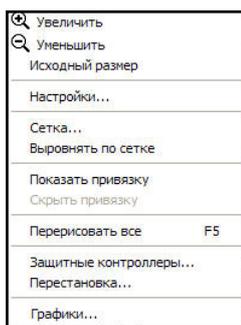


Рисунок 9 – Меню Вид

При выборе меню **Правка**, открывается окно, изображенное на рисунке 9.

Команды **«Увеличить» «Уменьшить», «Исходный размер»** соответственно увеличивают или уменьшают или возвращают в исходный размер алгоблоков в поле задач. Увеличить размер сверх исходного нельзя, т.е. исходный размер является максимальным. В уменьшенном масштабе возможно редактирование.

Команда **«Настройки»** открывает диалоговое окно, в котором устанавливаются основные настройки Пилона. Содержание этого окна описывается ниже в 3.3.

Команды **«Показать привязку», «Скрыть привязку»**. Если алгоблок или его канал привязаны к марке какого-либо объекта, в верхнем правом углу изображения алгоблока или канала показывается метка в виде небольшого черного квадратика. Данные команды позволяют показать или скрыть эту метку.

Команды **«Сетка», «Выровнять по сетке»**. Поле Пилона может быть «разграфлено» с помощью сетки, изображаемой в виде точек. По команде **«Сетка»** открывается дополнительное окно, в котором можно указать шаг сетки в пикселях, а также определить, показывать или не показывать сетку. Когда алгоблок устанавливается в поле задачи, его верхний левый угол привязывается к сетке. Однако, если после установки алгоблоков шаг сетки изменить, привязка к сетке может нарушиться. Команда **«Выровнять по сетке»** привязывает к сетке все алгоблоки текущей задачи.

Команда **«Перерисовать все»** используется, если изображение задачи нарушилось или в нем появились посторонние элементы.

Команда **«Защитные контроллеры»** используется только для работы с защитным кластером. По этой команде открывается дополнительное окно, в котором при работе Пилона в режиме **Обзор** можно установить приоритет одного из контроллеров кластера. При этом информация о значениях сигналов на входах/выходах алгоблоков будет выводиться на экран Пилона от приоритетного контроллера.

Команда **«Перестановка»** открывает дополнительное окно, в котором алгоблоки расставлены в порядке их выполнения. В этом окне порядок выполнения можно изменить (более подробно см. 3.7).

Команда **«Графики...»**. По этой команде в окне Пилона на рабочее поле задачи можно поместить график для наглядного просмотра изменения во времени одного или нескольких выходных сигналов алгоблоков.

### 3.1.6 Меню Техпрограмма

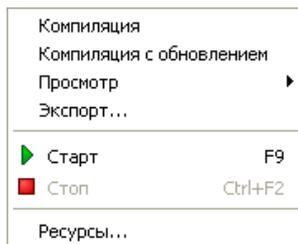


Рисунок 10 – Меню Техпрограмма

Инд. № подл.	Подп. и дата
Взаим. инв. №	Инд. №
Подп. и дата	Подп. и дата

При выборе меню **Техпрограмма**, открывается окно, изображенное на рисунке 10.

Команды **«Компиляция»**, **«Компиляция с обновлением»**, **«Просмотр»**, **«Экспорт»**, **«Ресурсы»** дублируют одноименные указания вышеописанной команды **«Контроллеры»** в меню **Файл** (рисунок 6).

Команда **«Старт»** переводит отдельные (выделенные) элементы Пилона (входы, выходы, алгоблоки или задача в целом) в режим работы **Обзор**, при котором в Пилоне отображаются реальные значения сигналов, полученные из контроллера. В этом режиме можно также прямо в Пилоне менять значения констант на входах алгоблоков и выполнять объектные команды. Более подробно работа Пилона в режиме **Обзор** описана в разделе 4.

Команда **«Стоп»** возвращает **Пилон** в режим обычного редактирования.

### 3.1.7 Меню Окна

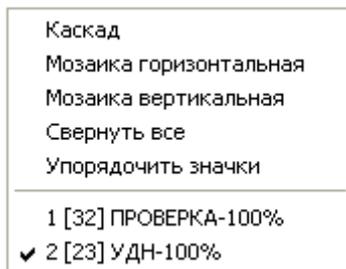


Рисунок 11 – Меню Окна

При выборе меню **Окна**, открывается окно, изображенное на рисунке 11.

Команды **«Каскад»**, **«Мозаика горизонтальная»**, **«Мозаика вертикальная»**. размещают открытые задачи соответствующим образом (в соответствии со стандартами Windows).

Команда **«Свернуть все»** сворачивает все открытые задачи и размещает их в нижнем правом углу окна Пилона.

Команда **«Упорядочить значки»** «прижимает» значки свернутых задач к левому нижнему углу окна Пилона и размещает их рядом. Команда действует, только если какие-то задачи свернуты.

**«Список задач»**. В нижней части ниспадающего меню размещается перечень всех открытых задач и указывается масштаб, в котором открыты задачи. Галочкой отмечается активная (текущая) задача.

### 3.1.8 Меню «?»

В этом пункте меню содержатся:

- этот документ;
- функциональное описание библиотечных алгоритмов Ремиконта Р-380;
- функциональное описание библиотечных алгоритмов Ремиконта Р-390.

## 3.2 Шкала инструментов

В режиме редактирования значки шкалы инструментов и их назначение приведены на рисунке 12.

Значки в основном повторяют команды диалоговых окон главного меню. Дополнительно имеется лишь значок команды **«Добавить алгоблок»**, нажатие которого вызывает дополнительное окно со списком библиотечных алгоритмов контроллера. Выбранный в списке алгоритм вставляется в поле активной задачи в качестве алгоблока технологической программы.

Кроме того, на шкале инструментов указывается имя открытой текущей задачи.

Инд. № подл.	Подп. и дата	Взаим. инв. №	Инд. №	Подп. и дата

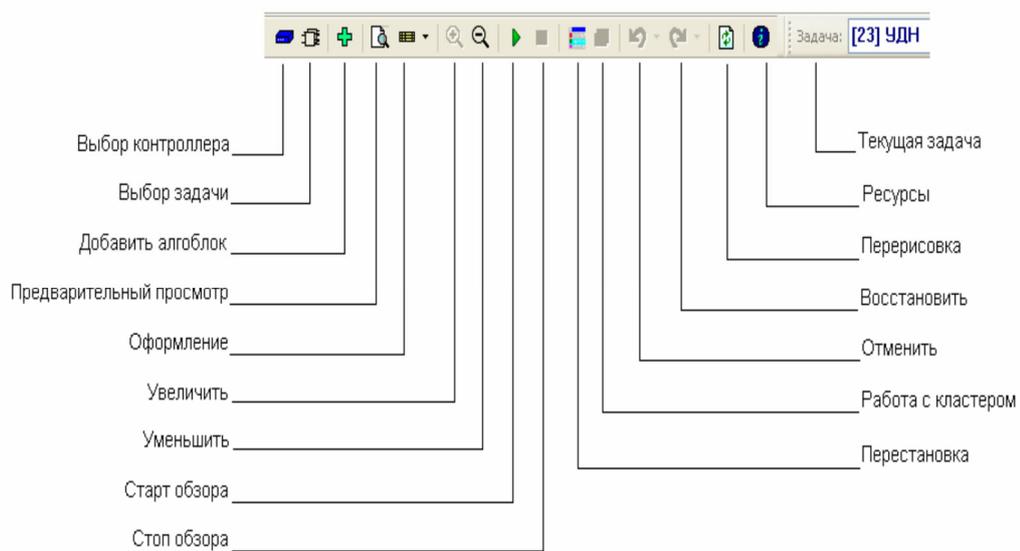


Рисунок 12 - Шкала инструментов в режиме редактирования

### 3.3 Настройки Пилона

Выбор в главном меню пункт **Вид** и в нем команду **Настройки...**, открывает доступ к диалоговому окну для установки настроек Пилона (Рисунок 13). Параметры настроек имеют несколько категорий.

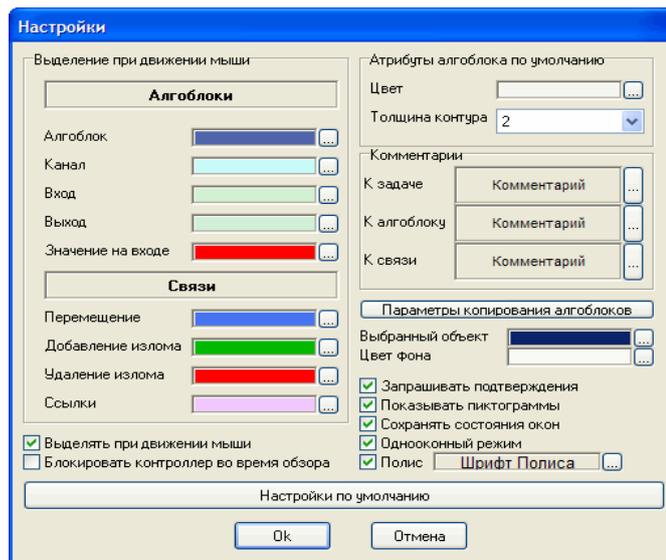


Рисунок 13 - Диалоговое окно настроек Пилона

#### 3.3.1 Настройки «Выделение при движении мыши»

Когда курсор мыши проходит над элементами изображения, цвет этих элементов может меняться. Элементами изображения являются алглобки и их линии связи.

Для алглобков можно отдельно задать цвета:

- для шапки алглобков;
- для каналов;
- для полей входов;
- для полей выходов;
- для значений констант.

Для линий связи можно задать изменение цвета:

- для режима перемещения отрезка линии связи;

Инв. №	Подп. и дата
Взаим. инв. №	
Подп. и дата	
Инв. № подл.	

Изм	Лист	№ докум	Подп.	Дата
-----	------	---------	-------	------

- для режима добавления излома линии;
- для режима удаления излома;
- для поля ссылки, заменяющей линию связи.

Более подробно смысл указанных настроек будет описан далее.

Установленные параметры действуют на все алгоблоки и линии связи всех задач и всех контроллеров, причем они «приписываются» конкретному пользователю Пилона. Это означает, что разные пользователи могут индивидуально выбирать для себя палитру, даже если они работают над одним проектом. Чтобы для всех пользователей алгоблоки выглядели одинаково, следует на всех рабочих станциях с Пилоном (и для всех пользователей одной рабочей станции) установить однотипную палитру.

### 3.3.2 Настройка «Блокировать контроллер во время обзора»

Эта категория касается возможностей коллективной работы над проектом. Если установлена галочка против этого поля, то выбранный контроллер на какой либо станции проектирования Пилон блокирует доступ к нему других станций проектирования. Если галочку снять, то режим **Обзор** становится неблокирующим (блокировка включается лишь на короткое время изменения параметров), при этом разные пользователи могут одновременно проектировать и отлаживать в режиме **Обзор** «свои» программы.

### 3.3.3 Настройки «Атрибуты алгоблока по умолчанию»

Задается цвет алгоблока и толщина его контура. Рекомендуется цвет алгоблока выбирать равным цвету поля или делать его несколько темнее, а толщину контура назначать равной 2 (возможны и любые другие настройки). Новые настройки действуют только на вновь введенные алгоблоки, атрибуты ранее введенных алгоблоков останутся без изменения.

### 3.3.4 Настройки «Комментарии»

Устанавливаются индивидуальные виды шрифтов для комментариев:

- к задачам;
- к алгоблокам;
- к связям между алгоблоками.

### 3.3.5 Настройки «Параметры копирования алгоблоков»

При нажатии этой клавиши открывается дополнительное диалоговое окно (рисунок 14), в котором указываются параметры, которые копируются вместе с копируемым алгоблоком. Если указатель **Больше не показывать это окно при копировании** не отмечен, то это окно будет появляться всякий раз, когда скопированный алгоблок вставляется в задачу. Если выбранные параметры применяются всегда при копировании, то этот указатель необходимо отметить и тем самым сократить число шагов в процедуре вставки алгоблоков (об указателях параметров см. рисунок 14).

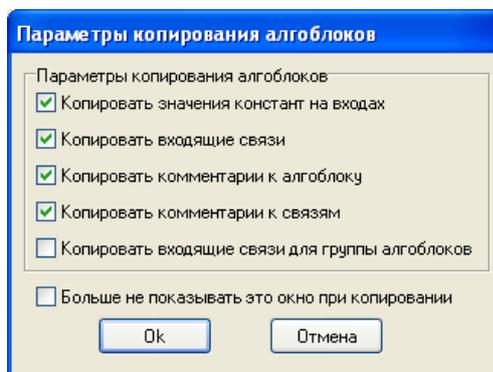


Рисунок 14 - Окно параметров копирования алгоблоков

### 3.3.6 Настройки «Цвет выбранного алгоблока и цвет фона»

Помимо единого цвета шапки алгоблоков можно установить отдельно цвет шапки выбранного алгоблока при его выборе указателем мыши. Кроме того, можно задать цвет фона для поля задачи в целом.

Инд. № подл.	Подп. и дата	Взаим. инв. №	Инд. №	Подп. и дата
--------------	--------------	---------------	--------	--------------

### 3.3.7 Настройка «Запрашивать подтверждения»

Если установить этот указатель, то при любом удалении будет запрашиваться подтверждение, в противном случае выбранный элемент будет удален без подтверждения.

### 3.3.8 Настройка «Сохранять состояния окон»

Если установить этот указатель, то при следующем открытии Пилона окна будут приведены в то же состояние, в каком они были на момент закрытия Пилона.

### 3.3.9 Настройка «Однооконный режим»

Если установить этот указатель, то при открытии новой задачи предыдущая будет автоматически закрываться. Таким образом, всегда будет открыто только одно окно. В противном случае (если эта опция не отмечена) можно будет открывать одновременно несколько окон.

### 3.3.10 Настройка «Полис»

Если установить этот указатель, то составление логической программы будет осуществляться при помощи специальной компоненты Пилона – Полис (см. ниже описание Полиса в 6.1). Если эта опция не указана, то создание логических программ обеспечивается только непосредственным конфигурированием алгоритмов УЛП и ЭТА.

### 3.3.11 Настройка «Шрифт Полиса»

При помощи этой опции устанавливаются стили шрифта текстов Полиса, отличные от заданных по умолчанию.

### 3.3.12 Настройки «По умолчанию»

Если нажать на эту клавишу, все вышеописанные настройки примут значения, принятые в Пилоне по умолчанию.

## 3.4 Свойства задач

Первое, что нужно сделать после запуска Пилона, это выбрать нужный контроллер. Для этого через меню **Файл/Выбор контроллера** или с помощью соответствующего значка открывается диалоговое окно с иерархическим списком контроллеров (рисунок 2), из которого сначала выбирается требуемый системный модуль и затем требуемый контроллер.

Следует помнить, что список системных модулей и контроллеров создается не в Пилоне, а в Администраторе, - в Пилоне можно лишь выбрать нужный контроллер. Кроме того, для подсистемы технологических защит выбирается не контроллер, а кластер, но далее работать с кластером можно так же, как с обычным контроллером.

### 3.4.1 Список задач

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

Чтобы создать или выбрать задачу для выбранного контроллера необходимо в соответствующем окне нажать клавишу **Задачи** или воспользоваться меню **Файл/Выбор задачи** или нажать значок **Выбор задачи** на панели инструментов. В результате откроется диалоговое окно (рисунок 15), в котором можно выполнить следующие действия:

- создать новую задачу или подзадачу;
- удалить задачу;
- скопировать задачу;
- отключить задачу;
- перенумеровать задачу;
- обновить список задач;
- переместить задачу.

Когда создается новая задача, она помещается в конец списка задач с первоначальным именем **Новая задача**. Если выделить имя задачи и затем щелкнуть по нему мышью, это имя можно изменить. Аналогичные правила действуют относительно создания подзадач, - разница лишь в том, что в этом случае следует предварительно выбрать родительскую задачу.

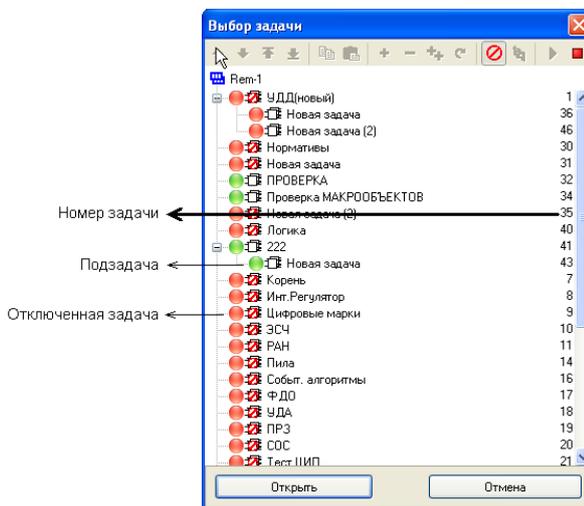
Любую задачу можно удалить. Когда задача удаляется, удаляются также все относящиеся к ней подзадачи и все входящие в них алгоблоки.

Инд. №				
Инд. №				
Инд. №				
Инд. №				
Инд. №				

**Примечание** - При удалении задачи следует проявлять осторожность, поскольку вернуть удаленную задачу невозможно.

Любую имеющуюся задачу можно скопировать и затем вставить в данный или любой другой контроллер, который выбирается из открывающегося диалогового окна. В этом же окне можно определить, следует ли копировать только выбранную задачу или также все входящие в нее подзадачи.

**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).



**Рисунок 15 - Окно со списком задач**

Любую задачу можно отключить - в этом случае в списке задач она помечается знаком запрета (перечеркнутый кружок).

**Примечание** - При отключении задачи все входящие в нее подзадачи также считаются отключенными, хоть против них знак запрета может и не стоять.

Если задача включается, то все входящие в нее подзадачи, которые индивидуально не были отключены, также включаются, но все ранее отключенные подзадачи остаются отключенными.

Когда создается новая задача, она получает очередной номер. Этот номер никак не указывает на порядок выполнения задачи, - он используется лишь в ссылках, определяющих связь между алгоблоками, которые находятся в разных задачах. Если задача перемещается, ее номер «следует» за ней, если удаляется, номер исключается из общего списка. Если в нумерации задач имеются неиспользованные номера, то при добавлении новой задачи она получает младший неиспользованный номер, если «дырок» в нумерации нет, новая задача получает очередной номер. С помощью клавиши **Перенумеровать** можно упорядочить все номера так, что они будут идти последовательно (сверху вниз), при этом соответствующим образом будут скорректированы все ссылки в конфигурации алгоблоков.

Если в списке задач не обнаруживаются какие-то из имеющихся задач или, наоборот, имеются лишние задачи, следует нажать клавишу **Обновить**.

С помощью стрелок в диалоговом окне любую задачу можно переместить, но только в пределах своего уровня иерархии. 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*.

Часть перечисленных команд доступна также из контекстного меню, которое можно открыть, если в поле диалогового окна нажать правую клавишу мыши.

### 3.4.2 Главное окно Пилона

После выбора задачи открывается главное окно Пилона (рисунок 16), в верхней строке которого указывается заголовок окна, путь (локальный или сетевой) к БД, имя выбранного контроллера, имя выбранной задачи и масштаб изображения. Ниже заголовка расположено главное меню, а под ним

Инва. №	Подп. и дата
Взаим. инв. №	
Подп. и дата	
Инва. № подл.	

– шкала инструментов со значками. Справа от шкалы инструментов находится окно, в котором дублируется номер и имя текущей (активной) задачи. В нижней части окна расположена строка состояния, в которую выводится информация о выбранном поле алглобка, о состоянии контроллера и состоянии текущей задачи. Основное место главного окна Пилона занимает поле задачи с алглобками.

Одновременно можно открыть несколько задач и расположить их в полный экран, каскадом, вертикальной или горизонтальной мозаикой. Список открытых задач можно увидеть через меню **Окна**. Если из этого меню выбрать одну из задач, она расположится поверх других открытых задач и станет активной.

Через меню **Файл** можно закрыть активную открытую задачу или закрыть сразу все открытые задачи. Как это принято в Windows, окно каждой задачи можно минимизировать, максимизировать, разместить каскадно или закрыть с помощью соответствующих значков в правом углу заголовка окна. Размеры окна можно менять, перетаскивая левой клавишей мыши его границы, при этом размер изображения алглобков не меняется.

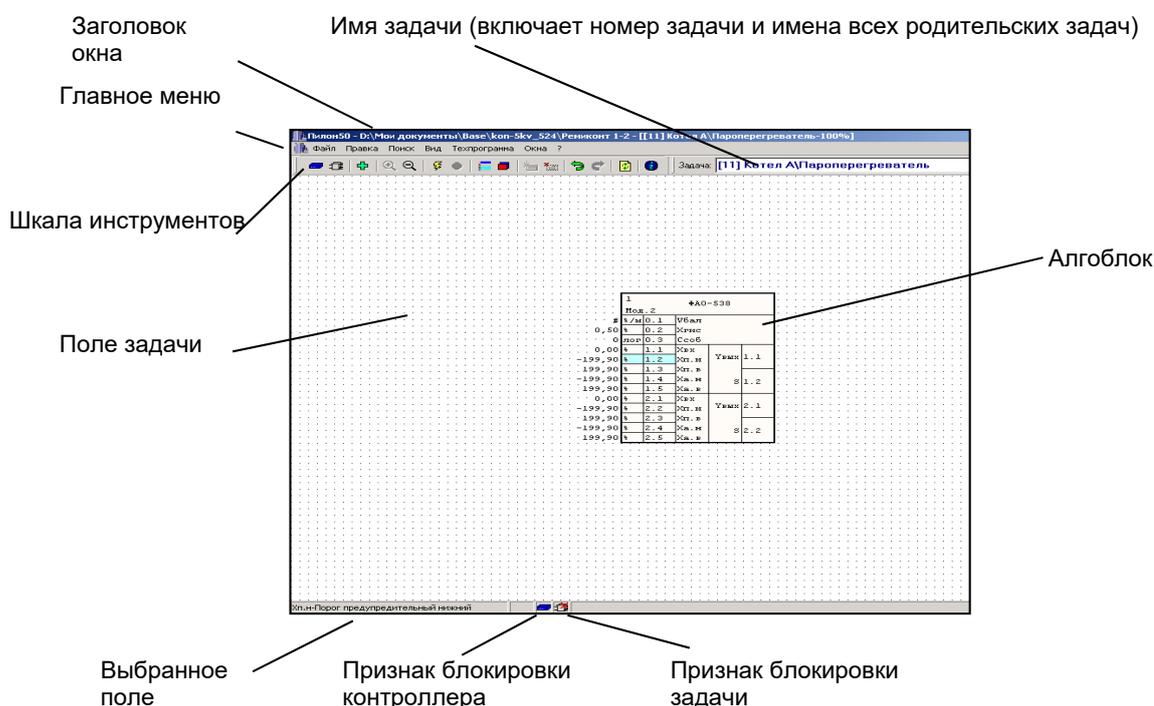


Рисунок 16 -Главное окно Пилона

### 3.4.3 Поле задачи

Используя меню *Вид*, можно менять масштаб изображения текущей задачи. Его можно на несколько ступеней уменьшить, при этом в пределах экрана разместится большее число алглобков. Масштаб уменьшенного изображения можно ступенями увеличивать или сразу вернуться к исходному размеру. Увеличить исходный размер нельзя.

Аккуратному размещению алглобков в окне задачи помогает сетка, которая точками покрывает поле задачи. Через меню **Вид/Сетка** вызывается диалоговое окно, в котором можно установить или отменить сетку, а для имеющейся сетки указать ее шаг (в пикселях). Сетка видна только на экране, при распечатке она отсутствует. Не зависимо от того, видна ли сетка на экране, рамки алглобков размещаются вдоль заданной сетки. Если шаг сетки изменить, то эти границы могут не совпасть с новой сеткой. Чтобы привести их в соответствие с новой сеткой, следует выполнить команду **Вид/Выровнять по сетке**.

Инв. №	Подп. и дата
Взаим. инв. №	
Подп. и дата	
Инв. № подл.	

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, либо поместив курсор мыши на край экрана – курсор примет изображение стрелки. Если затем нажать и удерживать левую клавишу мыши, поле задачи начнет перемещаться. Прокручивать поле задачи можно также с помощью клавиш:

- *управление курсором* - 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.

Для поля задачи можно выбрать нужный фон (рисунок 12). Установленная настройка будет действовать для всех задач, которые проектируются на данном компьютере.

В Пилоне 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).

### 3.4.4 Размещение алгоблоков

Ремиконт 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 «+», - сразу же открывается диалоговое окно со списком всех библиотечных алгоритмов (рисунок 17). Каждый алгоритм имеет библиотечный номер, шифр и наименование. Список алгоритмов можно упорядочить по алфавиту относительно любого из этих атрибутов, - для этого нужно щелкнуть мышью на заголовке соответствующего атрибута (считается, что латинские буквы по алфавиту стоят впереди русских, а цифры – впереди всех букв).

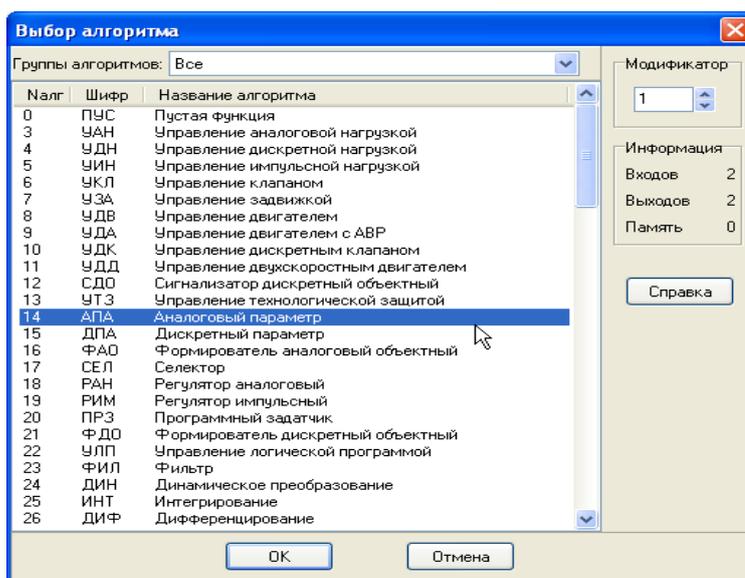


Рисунок 17 - Список библиотечных алгоритмов

Инва. №	Подп. и дата
Инва. №	Подп. и дата
Взаим. инв. №	Подп. и дата
Инва. № подл.	Подп. и дата

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**. Если в выбранном алгоритме модификатор отсутствует, в это поле блеклым цветом выводится обозначение «0».

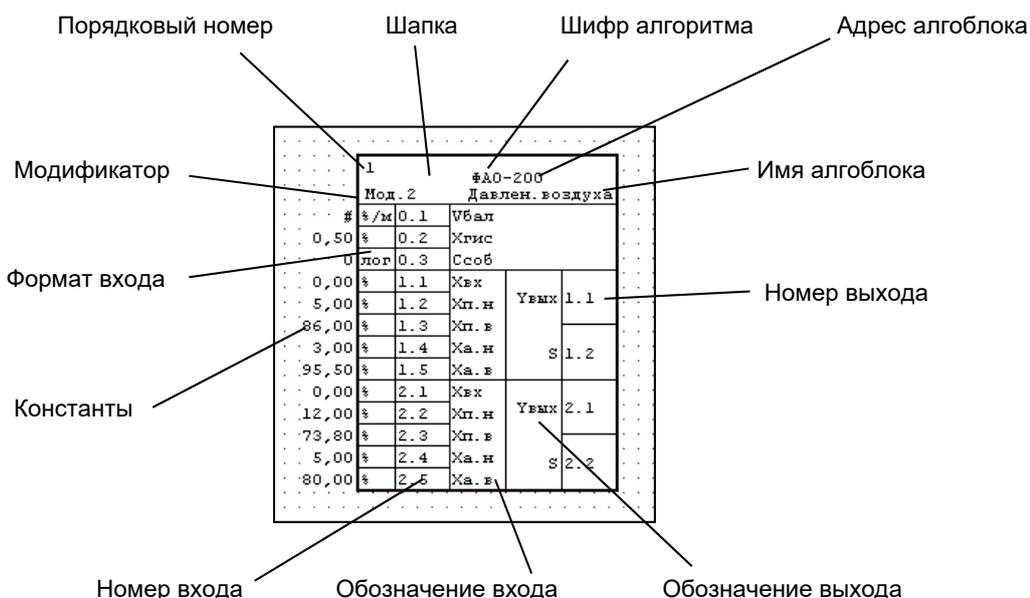
Выбранный алгоритм превращается в алгоблок, который первоначально размещается в середине экрана. После этого, захватив алгоблок мышью за верхний заголовок (шапку), его можно переместить в любую другую позицию.

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.

### 3.4.5 Реквизиты алгоблоков

Алгоблок всегда имеет вид прямоугольника. Его ширина всегда постоянна и не зависит от вида алгоритма, а высота зависит от вида алгоритма и от значения модификатора.



**Рисунок 18 - Реквизиты алгоблока**

В верхней части (шапке) алгоблока (рисунок 18) указываются:

- шифр алгоритма (трехбуквенное обозначение);
- значение модификатора;
- имя алгоблока;
- порядковый номер, определяющий порядок выполнения программы алгоблока;
- адрес, согласно которому алгоблок прописывается в базе данных.

Шифр и модификатор определяются выбранным алгоритмом и значением выбранного модификатора.

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 **Свойства**, а затем в новом окне – закладка **Общие**. В поле **Имя алгоблока** можно впечатать

Инва. №	Подп. и дата
Взаим. инв. №	
Инва. № подл.	

любое имя, содержащее не более 15 знаков, после чего оно появится в нижней части шапки алглобла. В закладке **Общие** можно также увидеть дополнительные параметры, связанные с данным алглоблом (идентификатор, дату и время модификации и т.д.).

Что касается порядковых номеров, то по мере ввода алглоблов они первоначально назначаются последовательно – первый введенный алглобок получает первый порядковый номер, второй – второй порядковый номер и т.д. Однако затем порядок выполнения алглоблов можно изменить (подробно об этом изложено в 3.7).

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

Тем не менее, следует иметь в виду следующее:

- в пределах одного контроллера имеются ограничения на число алглоблов и ряд других параметров;
- когда отдельные алглобки удаляются, адреса оставшихся алглоблов не изменяются следующим образом: если адрес алглобла меньше адреса удаленного алглобла, то его адрес не меняется, а если больше, то его адрес уменьшается на единицу;
- когда алглобок добавляется, он получает адрес, следующий по порядку адресов.

Шапка алглобла 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). В открывающемся списке одна строка выделена жирным шрифтом – это команда, которую можно выполнить, если дважды щелкнуть левой клавишей мыши по шапке алглобла.

### 3.4.6 Входы-выходы алглобла

Ниже шапки размещаются входы-выходы алглобла, которые имеют номера и обозначения, соответствующие их описанию в библиотеке алгоритмов Ремиконта.

Каждый вход и выход имеет двухзначный порядковый номер. Первый знак (перед точкой) – это номер канала, второй знак (после точки) – номер входов или выхода в пределах канала.

Ряд канальных алгоритмов имеют общую часть, не связанную с конкретными каналами. Для входов-выходов общей части указывается нулевой номер канала. Нулевой номер канала имеют также все входы-выходы алгоритмов, в которых каналы отсутствуют (например РАН, РИМ и т.п.).

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

### 3.4.7 Дополнительные атрибуты

Алглобок имеет пять полей – шапку, канал, вход, выход и значение на входе. Когда курсор мыши проходит над соответствующими полями алглобла или над отдельными участками линий связи между алглоблами, они подсвечиваются, причем цвет подсветки можно задавать индивидуально для каждого поля и отрезков линий связи через главное меню **Вид/Палитра** (рисунок 13).

Если подвести курсор мыши к шапке алглобла, нажать правую клавишу, в контекстном меню выбрать **Свойства** и затем выбрать закладку **Отрисовка**, то индивидуально для каждого алглобла можно установить цвет алглобла и толщину его контура.

Цвет алглобла не должен совпадать с цветом подсветки отдельных полей алглобла при перемещении курсора, - в противном случае эти поля не будут выделяться. Индивидуальную палитру удобно использовать при редактировании проекта, - цветовое выделение позволяет легко отличить вновь введенные или отредактированные элементы технологической программы.

### 3.4.8 Копирование алглоблов

В технологической программе контроллера можно копировать выделенный алглобок и затем вставлять его внутри исходной задачи или в другие задачи того же контроллера. Эта операция осуществляется через буфер обмена.

Чтобы копировать алглобок его надо выделить и выполнить в меню команду **Правка/Копировать**, чтобы вставить - выполнить команду меню **Правка/Вставить**.

Инд. № подл.	Подп. и дата	Взаим. инв. №	Инд. №	Подп. и дата

Другой способ – щелкнуть правой клавишей на шапке алглобка и из контекстного меню выбрать команду **Копировать**, затем команду **Вставить**

Еще один способ - выделить алглобок и нажать комбинацию клавиш **Ctrl+Ins** для копирования и комбинацию клавиш **Shift+Ins** для вставки...

При всех способах вставки открывается дополнительное окно для указания параметров копирования (рисунок 14).

Копирование также можно осуществлять через промежуточный файл в формате **XML**.

Только этот способ следует применять для копирования алглобка из одного контроллера в другой или из одного проекта АСУ ТП в другой проект.

Для этого способа необходимо щелкнуть правой клавишей на шапке алглобка и из контекстного меню выбрать команду **Копировать в XML**, для вставки – команду **Вставить из XML**. После выбора команды **Копировать в XML** в появившемся дополнительном окне необходимо выбрать указатель **В файл**. (Выбор указателя **В буфер обмена** предполагает действия, описанные выше). Указатель **В файл** открывает окно, в котором надо указать путь к файлу, где будет храниться копия алглобка. Для вставки после копирования необходимо щелкнуть правой клавишей мыши в любом свободном месте поля задачи и в контекстном меню выбрать команду **Вставить из XML**. Далее в возникшем окне установить указатель **Из файла**, затем выбрать файл, в который был скопирован алглобок и выполнить команду **Открыть**. Необходимо помнить, что во вставленном алглобке все привязанные марки отвязываются.

Описанными выше способами возможно также проведение групповых операций с алглобками. Например, можно скопировать или удалить сразу несколько алглобков. Для того, чтобы скопировать несколько алглобков 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 алглобков, удерживая нажатой клавишу **Shift**. Если уже выделено несколько алглобков и какие-то алглобки нужно исключить из выделения, следует по ним щелкнуть левой клавишей мыши при нажатой клавише **Shift**. Для того, чтобы отменить выделение всех алглобков нужно щелкнуть левой клавишей мыши по свободному полю.

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

Все указанные операции (кроме перемещения) можно выполнять также в окне перестановки (рисунок 19). Для того чтобы выделить несколько подряд идущих алглобков нужно нажать клавишу **Shift** и удерживая ее щелкнуть левой клавишей мыши по первому и последнему алглобку в выделяемой группе. Если выделяемые алглобки стоят не подряд, то следует нажать клавишу **Control (Ctrl)** и, удерживая ее, щелкнуть по всем алглобкам, которые необходимо выделить.

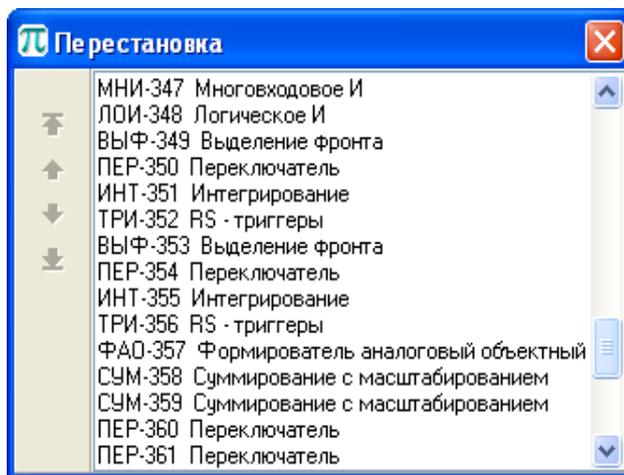


Рисунок 19 - Окно перестановки

Инв. № подл.	Подп. и дата	Взаим. инв. №	Инв. №	Подп. и дата

Копировать и вставлять группу (впрочем, как и отдельный алгоблок) можно лишь в пределах одного контроллера. Когда вставляется группа алгоблоков, то в диалоговом окне можно дополнительно определить, требуется ли копировать входящие связи для этой группы. Если соответствующая галочка не стоит, то связи сохраняются только внутри группы (в этой операции одиночный алгоблок рассматривается как группа, состоящая из одного алгоблока).

Параметры копирования алгоблоков в виде указателей устанавливаются в окне настроек (рисунок 14). Значения параметров следующие:

- копировать значения констант на входах – если эта опция выделена, то при вставке алгоблоков на их входах будут заданы те же значения (константы), что и у исходных. Если эта опция не указана, то на входах будут выставлены значения констант по умолчанию;
- копировать входящие связи - если эта опция выделена, то при вставке алгоблоков связи внутри группы сохраняются. Если эта опция не указана, то все связи будут удалены, а на ранее связанных входах будут выставлены значения констант по умолчанию;
- копировать комментарии к алгоблоку - если эта опция выделена, то при вставке алгоблоков будут также вставлены комментарии к ним. Если эта опция не указана, то у скопированных алгоблоков комментарии будут отсутствовать;
- копировать комментарии к связям - если эта опция выделена, то при вставке алгоблоков будут также вставлены комментарии к их связям. Если эта опция не указана, то у скопированных алгоблоков комментарии к связям будут отсутствовать;
- копировать входящие связи для группы алгоблоков - если эта опция выделена, то при вставке алгоблоков связи с алгоблоками, не входящими в копируемую группу сохраняются. Если эта опция не указана, то все связи с алгоблоками, не входящими в копируемую группу будут удалены, а на ранее связанных входах будут выставлены значения констант по умолчанию.

### 3.4.9 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. Единственное ограничение: *конфигурируемые алгоблоки должны принадлежать одному контроллеру*. Для связи алгоблоков, находящихся в разных контроллерах, используется другая процедура - информационного вывода (см. 5.3);
- 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;
- дискретный вход связывается не просто с выходом, но с определенным битом выбранного выхода. Для упакованного векторного выхода назначение битов определяется видом алгоритма, для дискретного неупакованного сигнала его значение содержится в старшем (16-м) бите, причем этот бит выбирается автоматически. Для аналогового выхода в младшем (первом) бите содержится признак достоверности, а в старшем (шестнадцатом) – знак сигнала. Более подробно форматы различного типа сигналов описаны в **Библиотеке алгоритмов**.
- на законфигурированном 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.

Связь может быть скопирована. Выделив связь, в контекстном меню нужно выбрать команду **Копировать**. При этом связь копируется в буфер обмена. Для вставки в контекстном меню свободного входа следует выбрать команду **Вставить**. Таким образом, можно провести много

Инв. № подл.	Подп. и дата
	Инв. №
Взаим. инв. №	Подп. и дата
	Инв. №

связей из одного источника. При выборе другого контроллера содержимое буфера обмена очищается.

### 3.5 Константы

#### 3.5.1 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 (некоторые исключения описаны ниже). 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. *Замечание: константа, равная бесконечности, отображается в Пилоне знаком «#».*

Возможен и другой способ. Двойным щелчком мыши по требуемому входному полю или через правую клавишу мыши и затем через меню **Значение на входе** открывается диалоговое окно с несколькими полями (рисунок 20):

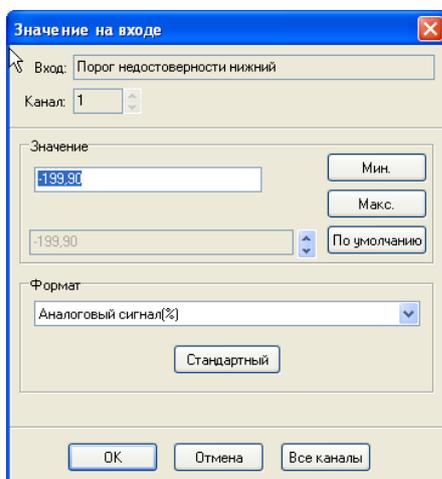


Рисунок 20 - Окно установки значения на входе

В поле **Вход** выводится название выбранного входа.

В области **Значение** в верхнее (белое) поле с клавиатуры вводится требуемое значение константы. В следующей строке показывается значение, которое реально будет воспринято контроллером. Дело в том, что константы в контроллере имеют определенную дискретность и определенный диапазон изменения. Если в верхнем поле указать значение вне требуемой дискретности или вне допустимого диапазона, то значение константы округляется до ближайшего возможного и результат показывается в нижнем поле. С помощью экранных клавиш **Мин**, **Макс**, **По умолчанию** можно сразу установить минимальное значение, максимальное значение или вернуться к начальному значению, указанному в **Библиотеке алгоритмов**.

В некоторых случаях значение константы удобно видеть в разных форматах. Для этого используется поле **Формат** и из раскрывающегося списка форматов выбирается нужный. С помощью экранной клавиши **Стандартный** всегда можно вернуться к стандартному формату, указанному в библиотеке алгоритмов. Реально потребность в изменении формата может возникнуть лишь для векторных сигналов, которые можно просматривать в двоичном или 16-ричном виде, и для временных сигналов – их можно задавать в секундах или минутах.

Диалоговое окно предоставляет дополнительные возможности, повышающие удобство и оперативность установки требуемых констант. Во-первых, не выходя из этого окна, можно задать значения констант на всех одноименных входах многоканального алгоритма, - для этого предварительно выбирается номер канала в поле **Канал**. Во-вторых, с помощью экранной клавиши **Все каналы** можно задать одинаковое значение константы на одноименных входах всех каналов выбранного многоканального алгоритма.

Инв. № подл.	Подп. и дата
Взаим. инв. №	Инв. №
Подп. и дата	Подп. и дата

### 3.5.2 Установка векторных констант

Ряд алгоритмов имеет входы, на которых устанавливается векторная константа. Обычно через вектор задается совокупность параметров, определяющих работу алгоритма. Диалоговое окно **Значение на входе** помогает выбрать требуемые параметры на векторном входе. В качестве примера на рисунке 21 показано содержание этого окна для векторного входа  $W_{кан}$  алгоритма АЦП. Окно содержит два поля, определяющих тип датчика и время преобразования. Если левой клавишей мыши выбрать поле **Тип датчика**, то в правом окне откроется иерархический список всех типов датчиков, которые «совместимы» с данным типом АЦП. Код выбранного датчика переносится в левое поле **Тип датчика**. Аналогично можно выбрать время преобразования. Итоговый код векторной константы выводится в нижнее поле диалогового окна, а после того, как будет нажата экранная клавиша **ОК**, этот код будет показан слева от соответствующего входа алгоблока. Как и для других констант, в данном случае, не выходя из диалогового окна, можно установить требуемые параметры во всех каналах данного алгоритма либо установить одни и те же значения сразу на всех одноименных входах во всех каналах алгоритма.

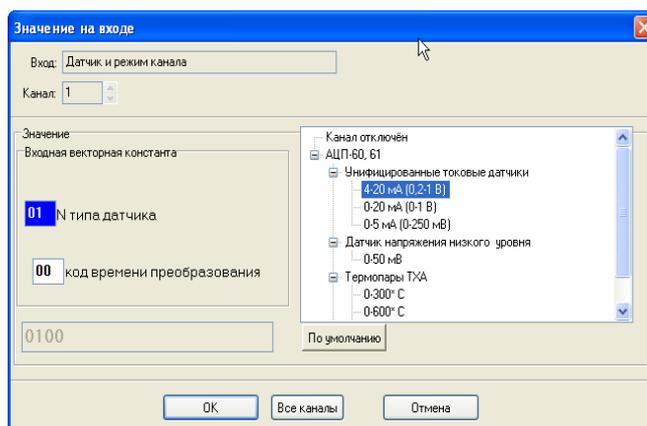


Рисунок 21 - Выбор параметров на векторном входе (пример)

### 3.5.3 Установка констант в физических единицах

Для алгоритмов ФАО и ФАР константы, определяющие пороги аварийной и предупредительной уставок, можно задавать не только в процентах, но и в физических единицах.

Внимание: константы в физических единицах можно задавать только в том случае, если соответствующий канал привязан к базе данных.

В этих же алгоритмах и для этих же констант в диалоговом окне добавлена экранная клавиша **Не используется**. Если нажать эту клавишу, то для верхнего и нижнего значения уставки будет задано соответственно максимальное и минимальное значение.

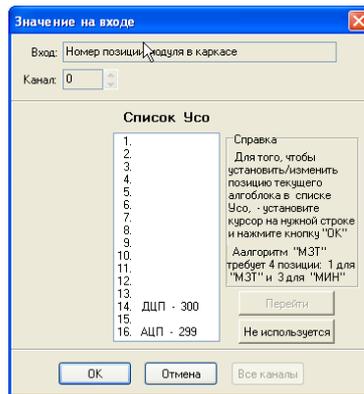
### 3.5.4 Уникальность констант

В некоторых типах алгоритмов определенные константы должны иметь уникальные (не повторяющиеся) значения в пределах одного контроллера (пример – номер позиции модуля УСО для всех алгоритмов, обслуживающих УСО). В таких случаях Пилон автоматически устанавливает очередное допустимое значение константы. Его можно изменить, но если новое значение повторит уже использованное, возникнет сообщение об ошибке. Подобный контроль уникальности значения констант имеется также в ряде алгоритмов, в которых автоматический ввод значений отсутствует, но уникальность требуется.

В качестве примера показаны диалоговые окна, которые используются для ввода подобных констант. При установке номера позиции модуля УСО в диалоговом окне (рисунок 22) видны все уже задействованные номера, - остается лишь выбрать «пустой» номер.

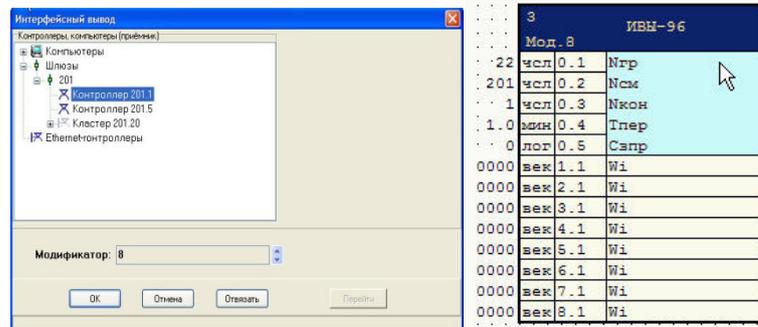
Здесь же можно выбрать определенный номер и затем нажать клавишу **Найти**, - будет найден алгоблок с соответствующим алгоритмом, обслуживающий выбранное УСО.

Инва. №	Подп. и дата
Взаим. инв. №	
Инва. № подл.	

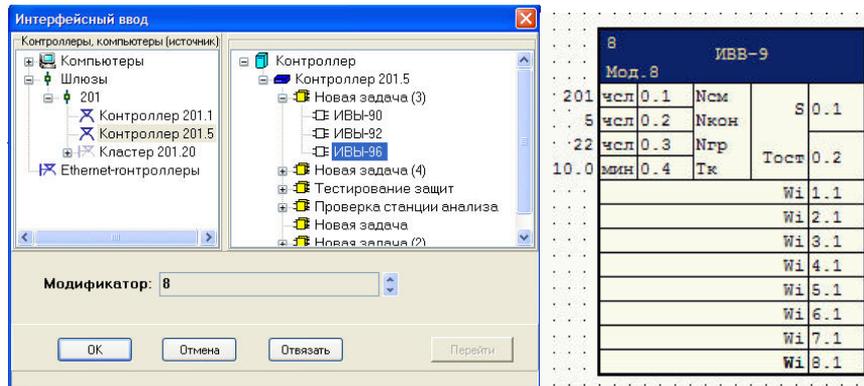


**Рисунок 22 - Установка уникальной константы (пример 1)**

При интерфейсной связи между контроллерами для алгоритма интерфейсного вывода ИВЫ в диалоговом окне показываются все имеющиеся системные модули и контроллеры и в нем можно выбрать нужный контроллер-приемник, - на входах Nсм и Nкон алглобка сразу же установятся требуемые константы (рисунок 23). Аналогично для алгоритма интерфейсного ввода ИВВ в диалоговом окне можно выбрать не только системный модуль и контроллер-источник, но и определить в нем конкретную группу интерфейсного вывода, с которой нужно связаться (рисунок 24).



**Рисунок 23 - Установка уникальной константы (пример 2а)**



**Рисунок 24 - Установка уникальной константы (пример 2б)**

В ряде алгоритмов некоторые входы предназначены только для связи с другими алгоритмами и на них константы не устанавливаются. Если такой вход не законфигурирован, значение константы на нем отсутствует.

В следующем разделе приведен список алгоритмов, на отдельных входах которых константы устанавливаются автоматически, контролируется уникальность констант, имеется возможность устанавливать константы в физических единицах и имеется запрет на установку констант.

### 3.5.5 Особенности установки констант

Ниже приводятся таблицы, в которых отмечаются различные особенности в установке констант:

ИВ. № подл.	Подп. и дата
Взаим. инв. №	ИВ. №
ИВ. № подл.	Подп. и дата

Изм	Лист	№ докум	Подп.	Дата	ПФДИ.421457.009 И3.2	Лист 31
-----	------	---------	-------	------	----------------------	------------

Автоматическая установка значения констант

Алгоритмы	Входы
УЛП	Нлп
ИВЫ	Нгр

Контроль уникальности значения констант

Алгоритмы	Входы
УЛП	Нлп
ИВВ	Комбинация значений Nсм, Nкон и Nгр
ИВЫ	Nгр для обычных контроллеров и комбинация значений Nсм, Nкон и Nгр для кластера
Все алгоритмы связи с УСО	Нпоз

Возможность установки констант в физических единицах

Алгоритмы	Входы
ФАО	X ан
	X ав
	X пн
	X пв
ФАР	X ан
	X ав
	X пн
	X пв

**Примечание** - Физические единицы можно задавать только для каналов, привязанных к базе данных.

Запрет установки констант

Алгоритм	Вход
РАН	Wсл
РИМ	Wсл
ПРЗ	Ni пр
УЛП	Ni эт
ДИН	Wсл
ИНТ	Wсл
СУМ	Wсл
УМД	Wсл
ПЕК	Wсл
ИМП	W
ЦИП	W

Инв. № подл.	Подп. и дата
Взаим. инв. №	Инв. №
Подп. и дата	Подп. и дата

## 3.6 Конфигурирование алгоблоков

### 3.6.1 Техника конфигурирования

При конфигурировании следует различать формат входа, с которым устанавливается связь, причем все форматы с точки зрения конфигурирования делятся на две категории – дискретные и все остальные (аналоговые, числовые, временные, векторные и т.п.).

Вначале рассмотрим конфигурирование входа с не дискретным форматом. Чтобы соединить такой вход с каким-нибудь выходом, следует подвести мышью 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.

Если автоматический маршрут неприемлем, можно поступить по-другому. «Зацепив» требуемый выход, следует щелкнуть левой клавишей мыши в любой точке экрана, где направление линии нужно изменить на 90 градусов. Последовательно щелкая в других точках экрана, можно провести линию связи по любому маршруту. Здесь имеется лишь одно ограничение: *число поворотов линии связи не может быть больше семи*. Если в процессе такой «прокладки» связь еще не установлена и нужно отменить все предыдущие действия, достаточно щелкнуть правой клавишей мыши.

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 двух коротких параллельных линий. 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. Если теперь щелкнуть левой клавишей мыши, образуется новый узел, который разделит отрезок на две части, одну из которых можно перетащить описанным выше способом. Часть узлов (но не все) сложной линии связи можно удалить и, тем самым, «спрямить» маршрут. Для этого к узлу подводится курсор мыши и, если курсор принял вид зачеркнутого круга, нажимается левая клавиша мыши, - узел удаляется.

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.3.6.2).

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 (рисунок 25 а).

Теперь рассмотрим конфигурирование дискретного входа. Здесь действуют те же правила, - отличие заключается лишь в том, что на заключительном этапе связь проводится не сразу, а вначале появляется диалоговое окно, - в нем следует задать номер бита выходного сигнала, с которым устанавливается связь. Этот номер, который может находиться в диапазоне 1-16, отображается на схеме слева от входного поля.

### 3.6.2 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 (рисунок 25 б). 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.

**Примечание** - Если связываются алгоблоки разных задач, связь между ними всегда имеет вид ссылки.

Инва. № подл.	Подп. и дата	Взаим. инв. №	Инва. №	Подп. и дата
---------------	--------------	---------------	---------	--------------

Изм	Лист	№ докум	Подп.	Дата	ПФДИ.421457.009 И3.2	Лист
						33

Ссылка, «прикрепленная» к источнику, содержит сведения об алглоблке-приемнике, ссылка, «прикрепленная» к приемнику, содержит аналогичные сведения об алглоблке-источнике. Если ссылки относятся к алглоблкам, находящимся в разных задачах, впереди указывается номер задачи, если ссылки связывают алглобки одной задачи, номер задачи опускается.

Ссылку на выходе (но не на входе) алглобка 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 Пилоне можно быстро найти источник или приемник, на которые указывают ссылки. Для этого следует поместить курсор на ссылку, нажать правую клавишу мыши и в диалоговом окне выбрать команду **Перейти по ссылке**, - произойдет автоматическая прокрутка поля задачи, так что искомый алглобок переместится в видимую часть экрана и будет выделен 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.

### 3.6.3 Дополнительные возможности конфигурирования

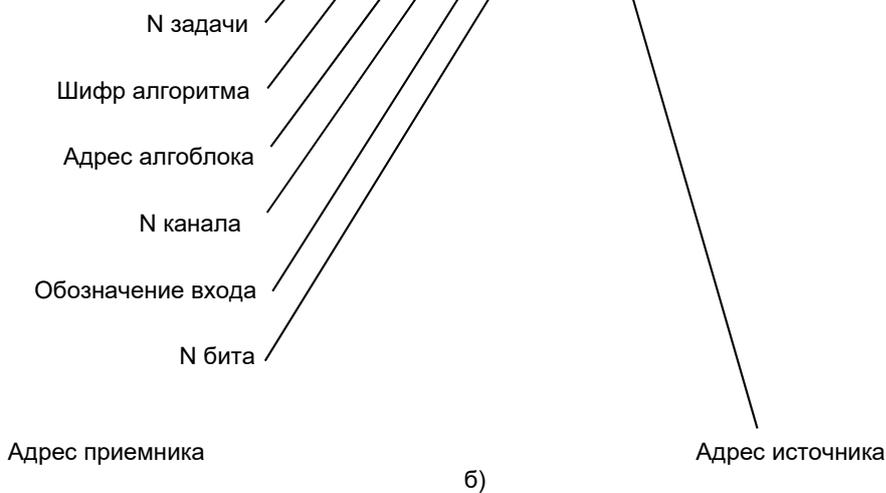
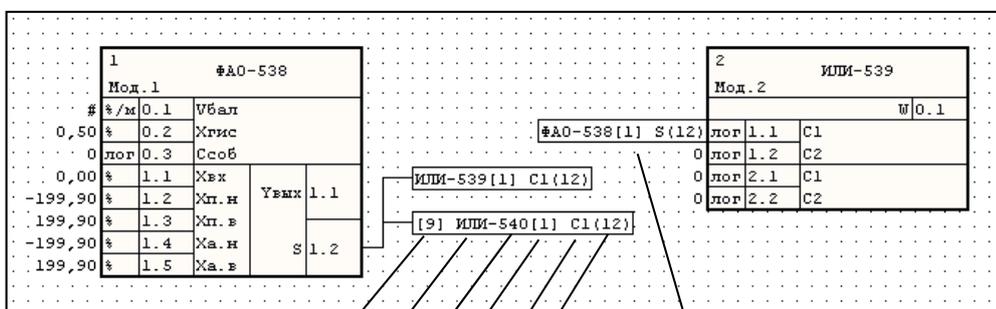
Если подвести курсор к линии связи и по правой клавише мыши открыть контекстное меню, то по отношению к выбранной связи можно выполнить ряд дополнительных действий:

- **Удалить связь.** Если выбрать эту операцию, проведенная связь будет удалена, а на входе алглобка, к которому подходила эта связь, будет установлена константа по умолчанию;
- **Провести автоматическую связь.** Эта функция (команда **Авто**) активна, только если связь проводилась по индивидуальному маршруту или один из отрезков ранее установленной автоматической связи перемещался. По команде **Авто** текущий маршрут заменяется на маршрут, выбираемый самим Пилоном. Имеется возможность заменить на автоматическую не только выбранную связь, но и все связи, относящиеся к выбранному алглобку. Для этого правой клавишей мыши следует щелкнуть на шапке алглобка и в открывшемся контекстном меню выбрать команду **Сделать все связи автоматическими**. Если связь проведена в автоматическом режиме, то при перемещении алглобка вслед за ним будут перемещаться все связанные с ним линии конфигурации, так что все время будет сохраняться оптимальный маршрут связи. Если отдельные отрезки линии связи перемещались вручную, то автоматический режим.

Инв. № подл.	Подп. и дата	Взаим. инв. №	Инв. №	Подп. и дата



а)



б)

**Рисунок 25 - Конфигурирование алгоблоков**

а) – прямая связь между алгоблоками

б) – связь в виде ссылки

### 3.6.4 Атрибуты линии связи

В 3.3 указывалось, что через главное меню **Вид/Настройка** можно определить цвета отдельных участков линии связи (рисунок 13), над которыми проходит курсор мыши. Помимо этого, для каждой конкретной линии можно задать индивидуальные параметры, - нужно щелкнуть правой клавишей мыши на линии связи и в открывшемся диалоговом окне выбрать строку **Свойства**, а в нем – закладку **Отрисовка**. Теперь для выбранной линии можно задать ее цвет, толщину и стиль.

Инв. № подл.	Подп. и дата	Инв. №	Подп. и дата
Взаим. инв. №	Инв. №	Взаим. инв. №	Инв. №
Инв. № подл.	Подп. и дата	Инв. №	Подп. и дата

Изм	Лист	№ докум	Подп.	Дата
-----	------	---------	-------	------

Замечание: стиль, отличный от непрерывной линии, можно установить только при толщине линии 1 пункт. Для больших, чем 1 пункт, значений линия всегда непрерывна не зависимо от выбранного стиля.

### 3.6.5 Пересечение линий связи

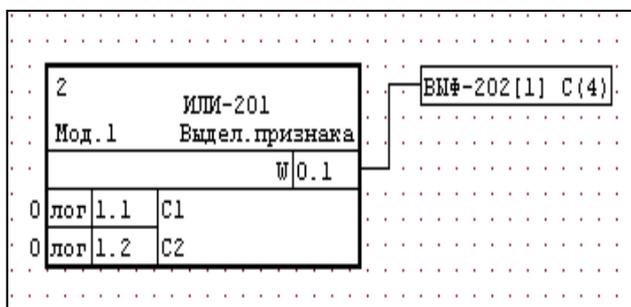
В задаче с большим числом алгоблоков и линий связи последние могут пересекаться. Если пересечение линий связи означает их соединение, такое пересечение обозначается точкой. Если линии пересекаются, но не соединяются, точка отсутствует.

## 3.7 Перемещение и перестановка алгоблоков

### 3.7.1 Возможности перемещения

Когда вводится очередной алгоблок, он вначале помещается в середину экрана. Дальше 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. Следует помнить, что отслеживание связей имеет нюансы – оно выполняется по разному для автоматической и не автоматической связи (см. 3.6 и 3.6.3).

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, относящиеся к выходам алгоблока (но не входные ссылки).



Техника перемещения та же – ссылка захватывается левой клавишей мыши и переносится в нужное место. Между ссылкой и выходом образуется связь, похожая на связь между алгоблоками (рисунок 26). Особенность этой связи – ее нельзя редактировать (т.е. перемещать отдельные линии, удалять или добавлять узлы). Когда перемещается алгоблок – за ним следует и ссылка, при этом конфигурация связи между ссылкой и выходом алгоблока остается неизменной.

Рисунок 26 – Ссылка на выходе алгоблока

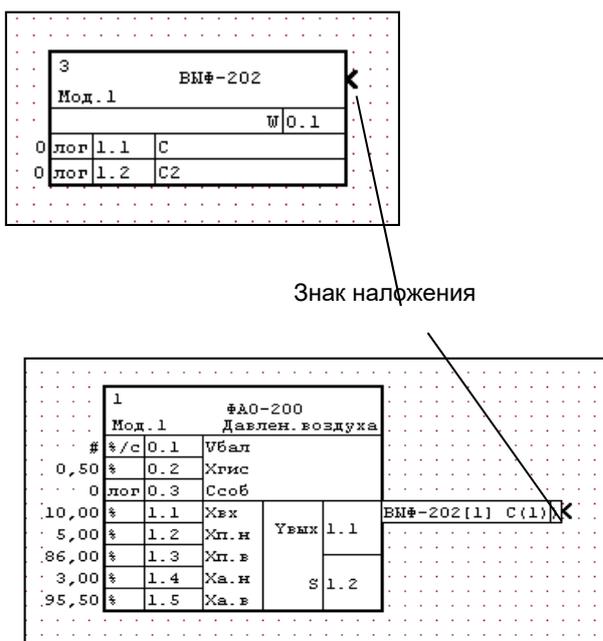


Рисунок 27 – Графическое наложение ссылок

Инв. № подл.	
Подп. и дата	
Взаим. инв. №	
Инв. №	
Подп. и дата	

Изм	Лист	№ докум	Подп.	Дата
-----	------	---------	-------	------

### 3.7.2 Наложение алгоблоков и ссылок

При перемещении алгоблоков может произойти их наложение – один алгоблок закроет изображение другого алгоблока. В этом случае справа от шапки алгоблока ставится специальный значок в виде галочки (рисунок 27).

Аналогичный знак ставится и справа от ссылки – он означает, что под данной ссылкой находится другая ссылка. В свою очередь, это говорит о том, что с данным выходом связан больше чем один вход.

Как в первом, так и во втором случаях рекомендуется «растачить» наложенные элементы.

### 3.7.3 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 верхнем левом углу шапки алгоблока.

В любое время порядок выполнения алгоблоков можно изменить. Для этого выбирается команда **Вид/Перестановка** или на шкале инструментов нажимается значок, содержащий цветные горизонтальные полосы, - открывается диалоговое окно со списком всех алгоблоков данной задачи (рисунок 28). Это окно, размеры которого можно менять, 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 левой 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.

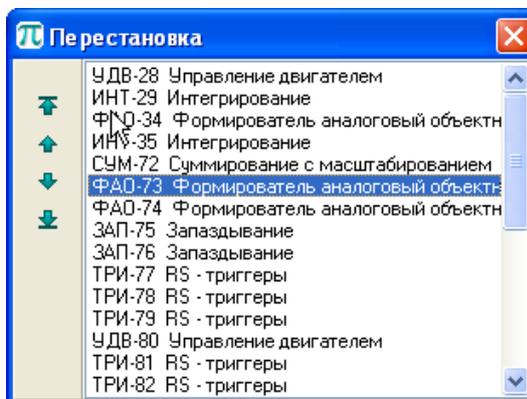


Рисунок 28 - Диалоговое окно перестановки алгоблоков

## 3.8 Комментарии

### 3.8.1 Виды комментариев

Для того, чтобы в технологической программе легче было ориентироваться, ее рекомендуется снабжать комментариями. Работая с Пилоном, пользователь может использовать следующие комментарии:

- к задаче;
- к алгоблоку;
- к линии связи для источника;
- к линии связи для приемника.

Инв. №	Подп. и дата
Взаим. инв. №	
Подп. и дата	
Инв. № подл.	

### 3.8.2 Комментарии к задаче

Комментарии к задаче помещаются в любом свободном месте поля задачи, причем число таких комментариев в каждой задаче не ограничено. Число символов в каждом комментарии – не больше 100. 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 «вдавленное» прямоугольное поле, в которое можно вводить текст. Когда текст введен, следует щелкнуть мышью в любом свободном месте поля задачи вне «вдавленного» прямоугольника, - текст помещается в поле задачи. Затем этот текст с помощью мыши можно перетащить в любое другое место. Выбрав соответствующие команды в диалоговом окне, введенный текст можно отредактировать или удалить.

Через главное меню **Вид/Настройка** можно выбрать шрифт, используемый для комментария (см. раздел 3, рисунок 12).

### 3.8.3 Другие виды комментариев

Аналогично комментариям к задаче можно вводить комментарии к каждому алгоблоку и к линии связи 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 связям, подходящим к данному алгоблоку.

## 3.9 Привязка

### 3.9.1 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 (рисунок 29).

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 Пилоне можно до ввода объектов в Аркаде, но, чтобы выполнить привязку, требуемый объект обязательно должен быть введен в базу данных.

### 3.9.2 Отображение привязки

В Пилоне предусмотрена специальная метка, которая указывает на привязанные элементы. Если привязывается объектный одноканальный алгоритм, эта метка ставится в верхнем правом углу канальной области. Если алгоритм объектный многоканальный, метка ставится в области каждого привязанного канала. Если привязываются отдельные входные или выходные сигналы, метка ставится в поле соответствующего входа или выхода.

Через меню **Вид** можно вывести отображение меток привязки или скрыть эти метки.

Инд. № подл.	Подп. и дата
Взаим. инв. №	Инд. №
Подп. и дата	Подп. и дата

Изм	Лист	№ докум	Подп.	Дата	ПФДИ.421457.009 И3.2	Лист
						38

6 УЛН-205 Розжиг горелок				4 #А0-203 Температура				5 СУН-204 Суммирование			
Мод. 3				Мод. 2				Мод. 2			
0 лог	0.1	Сбл	W1	0.1	0.50	%	0.1	Убал	0.000	0.1	У
0 числ	0.2	Нэт	W2	0.2	0,50	%	0.2	Хрис	1,0000	0.1	У
0 лог	0.3	Сожк	W3	0.3	0,00	%	0.3	Ссоб	0,00	0.2	У
0000	век	0.4	Wпу	0.4	5,01	%	1.1	Хжк	1,0000	0.1	У
0 лог	0.5	Спе	W5	0.5	85,00	%	1.2	Хп.в	0,00	0.2	У
0 лог	0.6	Ссв	W6	0.6	3,00	%	1.3	Хп.в	1,0000	0.1	У
0 лог	0.7	Ссвр	W7	0.7	95,00	%	1.4	Ха.в	0,00	0.2	У
0 числ	0.8	Нэта	W8	0.8	0,00	%	1.5	Ха.в	1,0000	0.1	У
0 лог	0.9	Тост	W9	0.9	6,01	%	2.1	Хжк	0,00	0.2	У
0 числ	0.10	Нлтп	Wраб	0.10	90,00	%	2.2	Хп.в	1,0000	0.1	У
0 числ	1	Нэт	S	0.11	2,00	%	2.3	Хп.в	0,00	0.2	У
0 числ	2.1	Нэт	W1с	0.12	95,00	%	2.4	Ха.в	1,0000	0.1	У
0 числ	3.1	Нэт	W2yc	0.14		%	2.5	Ха.в		0.2	У

Привязка одноканального  
объектного алгоритма

Привязка каналов  
многоканального  
объектного алгоритма

Привязка отдельных  
входных и выходных сигналов

Рисунок 29 - Варианты привязки алгоритмов

### 3.9.3 Соответствие алгоритмов оперативным типам объектов

При подготовке базы данных следует учитывать, что в процессе привязки в Пилоне показываются не все марки, введенные в базу данных, а лишь те из них, которые соответствуют оперативному типу привязываемого алгоритма. В Таблице 1 приведен перечень соответствий алгоритмов и оперативных типов. Если привязывается объектный алгоритм (например, РИМ) или отдельный канал объектного алгоритма (например, ФАО), то будут показаны все еще не привязанные марки, но только соответствующего оперативного типа объекта (например, **Регулятор импульсный** или **Датчик аналоговый**). Если привязывается какой-либо вход или выход алгоритма (неважно, объектного или объектного, не важен и формат входа или выхода), то будут показаны марки всех объектов, записанных в базе данных как аналоговые, дискретные, числовые и временные сигналы.

Таблица 1 – Соответствие алгоритмов и оперативных типов

№	Объектный алгоритм		Оперативный тип
	Шифр	Наименование	
1.	АПА	Аналоговый параметр	Параметр аналоговый
2.	ВЫК	Выключатель	Выключатель
3.	ГУЗ	Групповое управление защитой	Защита групповая
4.	ДПА	Дискретный параметр	Параметр дискретный
5.	ПРЗ	Программный задатчик	Задатчик программный
6.	РАН	Регулятор аналоговый	Регулятор аналоговый
7.	РИМ	Регулятор импульсный	Регулятор импульсный
8.	РПН	Регулятор напряжения под нагрузкой	Регулятор напряжения
9.	РУБ	Ручная блокировка	Блокировка ручная
10.	СДО	Сигнализатор дискретный объектный	Сигнализатор дискретный
11.	СЕЛ	Селектор	Селектор
12.	УАН	Управление аналоговой нагрузкой	Нагрузка аналоговая
13.	УДА	Управление двигателем с АВР	АВР
14.	УДВ	Управление двигателем	Двигатель двухскоростной
15.	УДВ	Управление двигателем	Двигатель односкоростной
16.	УДК	Управление дискретным клапаном	Клапан дискретный

Инв. № подл.	Подп. и дата
Взаим. инв. №	Инв. №
Подп. и дата	Подп. и дата

Изм	Лист	№ докум	Подп.	Дата
-----	------	---------	-------	------

Продолжение таблицы 1

17.	УДН	Управление дискретной нагрузкой	Нагрузка дискретная
18.	УЗА	Управление задвижкой	Задвижка универсальная
19.	УИН	Управление импульсной нагрузкой	Катушка дугогасящая
20.	УИН	Управление импульсной нагрузкой	Нагрузка импульсная
21.	УКЛ	Управление клапаном	Клапан
22.	УЛП	Управление логической программой	Логическая программа
23.	УТЗ	Управление технологической защитой	Защита локальная
24.	ФАО	Формирователь аналоговый объектный	Датчик аналоговый
25.	ФАР	Формирователь аналоговый резервированный	Датчик аналоговый резервированный
26.	ФДО	Формирователь дискретный объектный	Датчик дискретный
27.	ФДР	Формирователь дискретный резервированный	Датчик дискретный резервированный
28.	ХДС	Хроника дискретных событий	Хроника дискретных событий
29.	ЭСЧ	Электросчетчик	Электросчетчик
30.	Входы или выходы любых алгоритмов		Сигнал аналоговый
31.			Сигнал дискретный
32.			Сигнал числовой
33.			Сигнал временной

### 3.9.4 Технология привязки

Рассмотрим технологию привязки на примере многоканального объектного алгоритма **ФАО – формирователь аналоговый объектный**. При щелчке правой клавишей мыши в области шапки алгоблока из открывшегося контекстного меню выбирается команда **Привязка**, - открывается диалоговое окно привязки (рисунок 30). В нем следует нажать на значок «+» - откроется следующее диалоговое окно (рисунок 31), в котором предлагается выбрать узел, тип объекта и марку объекта. Марку можно ввести непосредственно с клавиатуры и без предварительного выбора узла и типа объекта. Для этого в диалоговом окне в поле **Отбор по марке** – все имеющиеся марки, соответствующие типу привязываемого алгоритма, появляются в поле **Объект**. Теперь нужную марку можно впечатать в поле **Марка**. Выбрав нужные записи или впечатав марку, нажимается экранная клавиша **Добавить**, - соответствующая строка появляется в окне привязки. Теперь выбранный канал привязан к выбранной марке объекта.

Не закрывая открытые диалоговые окна и выбирая нужный канал, требуемую марку и каждый раз нажимая клавишу **добавить**, можно привязать все каналы выбранного алгоритма.

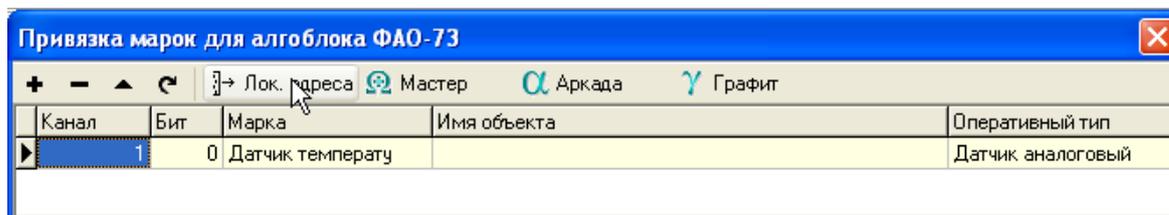
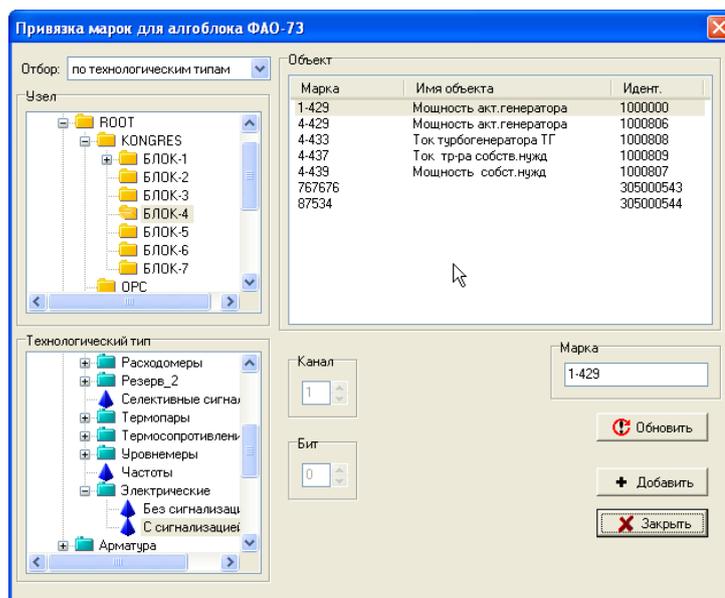


Рисунок 30 - Диалоговое окно привязки

Инд. № подл.	Подп. и дата
Взаим. инв. №	Инд. №
Подп. и дата	Подп. и дата



**Рисунок 31 - Выбор объекта привязки**

Если привязан один из каналов алгоритма, это не означает, что в нем нельзя дополнительно привязать отдельные входы и выходы, а для векторных сигналов – отдельные биты. Чтобы это сделать, нужно использовать закладки **Вход**, **Выход** диалогового окна привязки.

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. Следует помнить (см. 3.9.3), что, если привязываются отдельные входы или выходы, в диалоговом окне выбора объектов будут помещены только те марки объектов, которые имеют оперативный тип **Сигнал аналоговый**, **Сигнал дискретный**, **Сигнал числовой** и **Сигнал временной**. При привязке одиночного дискретного сигнала следует дополнительно выбрать номер бита.

Чтобы отвязать привязанную марку, в диалоговом окне привязки марок следует выделить нужную запись и на панели инструментов нажать значок «-». При удалении привязки нужно быть осторожным – вернуть привязку можно, только заново повторив все описанные выше действия.

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 марки окна привязки.

### 3.10 Права доступа и коллективная работа с Пилоном

#### 3.10.1 Права доступа

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

- 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). 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.

Инв. №	Подп. и дата
Взаим. инв. №	
Подп. и дата	
Инв. № подл.	

### 3.10.2 Коллективная работа

В разработке одного проекта обычно участвует несколько пользователей. Чтобы сохранить целостность БД, используется механизм блокировок, которые в большинстве случаев ставятся автоматически. В Пилоне 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. 4.2), 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).

Признаки блокировки (отдельно для контроллера и текущей задачи) показываются в строке состояния в виде перечеркнутого символа.

## 4 Отладка технологических программ

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

Для информационной связи с контроллером необходимо произвести настройку его Шлюза Ш-81 в информационной сети по правилам, указанным в документе «Комплексы программно-технические Квинт-6. Администрирование проектов АСУ ТП. Руководство пользователя ПФДИ.421457.009 Из. 3».

### 4.1 Компиляция и загрузка технологической программы

#### 4.1.1 Виды компиляции

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.

В Пилоне имеется два варианта компиляции:

- обычная (без обновления);
- с обновлением.

Понятие обновления трактуется следующим образом. В процессе работы состояние различных параметров или значения сигналов ряда алгоритмов не только меняются, но и после изменения запоминаются. К таким параметрам относятся, например, режимы управления (автоматический или ручной), величина задания и т.п., а к накопленным сигналам – выходы интеграторов, триггеров, счетчиков и т.п.). При компиляции в загрузочном файле для каждого алгоритма делается пометка – в процессе последующей загрузки техпрограммы в контроллер оставлять измененные параметры или накопленные значения без изменения либо сбросить их и привести к начальным значениям, указанным в описании библиотеки алгоритмов. Первый вариант является обычной компиляцией (без обновления), второй вариант – компиляцией с обновлением.

Следует учитывать, что если после последней компиляции добавлялись какие-либо алгоблоки, то при последующей даже обычной компиляции они в загрузочном файле будут помечены как новые и в процессе последующей загрузки их параметры (но только этих алгоблоков) будут сброшены к начальным значениям.

Инд. № подл.	Подп. и дата	Взаим. инв. №	Инд. №	Подп. и дата
--------------	--------------	---------------	--------	--------------

Изм	Лист	№ докум	Подп.	Дата	ПФДИ.421457.009 Из.2	Лист
						42

## 4.1.2 Обычная компиляция и загрузка

В разделе **Файл** главного меню по команде **Контроллеры / Технологическая программа** открывается окно (рисунок 32), в котором с помощью галочек и команды **Выполнить** выбирается вариант последующих действий.

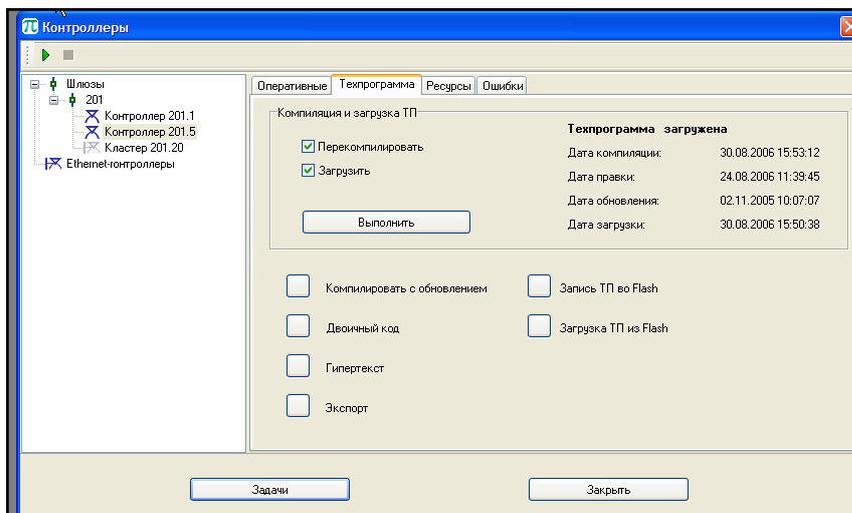


Рисунок 32 - Окно команды «Контроллеры / Технологическая программа»

Если при открытии окна галочка **Перекompиллировать** не стоит, значит с момента последней компиляции технологическая программа не менялась. Однако, это не мешает поставить галочку и, повторно откомпилировать технологическую программу. Если не стоит галочка **Загрузить**, значит после последней загрузки загрузочный файл не менялся. Однако, это не мешает поставить галочку и, повторно загрузить ту же программу. После загрузки состояние не измененной части техпрограммы сохранится.

Если при открытии окна галочки уже стоят, это означает, что в технологическую программу были внесены изменения. Однако, это не мешает снять галочку перекомпиляции, а галочку перезагрузить оставить – в этом случае будет загружена «старая» технологическая программа. С другой стороны, можно галочку перекомпиляции оставить, а галочку перезагрузить снять. В этом случае программа будет откомпилирована, но не загружена, - к загрузке можно будет вернуться позже.

В правой части окна указаны даты соответствующих действий. Дата правки показывает, когда программа последний раз корректировалась в Пилоне. Дата компиляции – когда она последний раз компилировалась обычным образом (без обновления). Дата обновления – когда программа последний раз компилировалась с обновлением. Дата загрузки – когда программа последний раз загружалась в контроллер.

## 4.1.3 Компиляция с обновлением

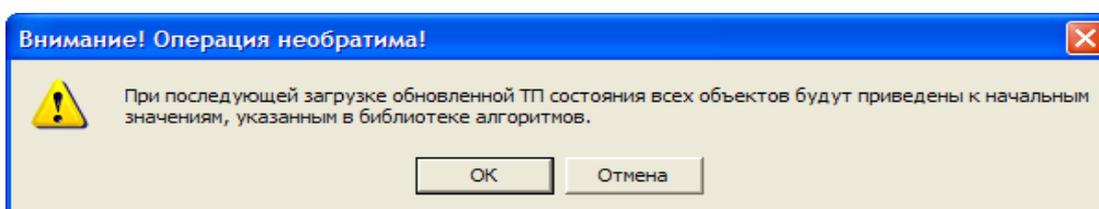


Рисунок 33 – Окно предупреждения о необратимости процесса

По команде **Компиляция с обновлением** открывается диалоговое окно с предупреждением о том, что эта процедура необратима (рисунок 33). Если ответить согласием, будет выполнена только компиляция с обновлением, загрузка в этой процедуре не предусмотрена.

Если после этого указать команду **Загрузить/Выполнить**, откроется диалоговое окно с напоминанием о том, что после загрузки состояние всех алгоритмов технологической программы будет приведено в начальное состояние.

Ив. № подл.	Подп. и дата
Взаим. ив. №	Ив. №
Подп. и дата	Подп. и дата

#### 4.1.4 Загрузка

Чтобы загрузить подготовленную технологическую программу в Ремиконт, рабочая станция с Пилоном и Ремиконт должны находиться в общей сети. Технологическая программа загружается в контроллер целиком, при этом загружаются все подготовленные задачи, не зависимо от того, включены они в Пилоне или отключены.

Чтобы загрузить программу в Ремиконт, в окне (рисунок 6) выбирается команда **Загрузить /Выполнить**, после чего появляется окно, изображенное на рисунке 34.

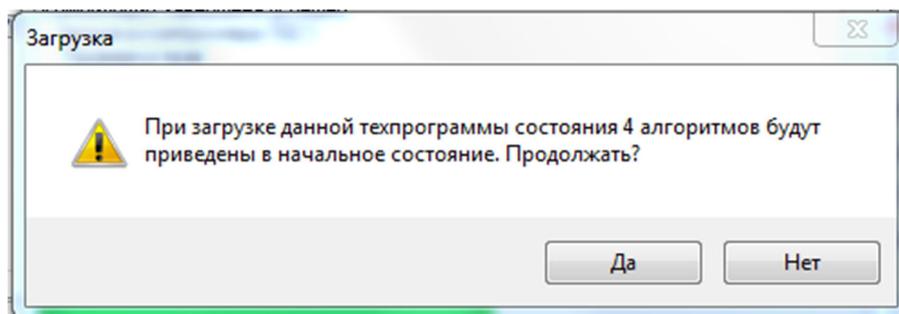


Рисунок 34 – Диалоговое окно процесса загрузки

В нем сообщается, сколько алгоритмов будет приведено в начальное состояние – это означает, что данное число алгоритмов было добавлено после последней загрузки (в данном случае четыре)

В ответ на согласие и в случае, если контроллер находится в режиме **Работа**, появляется новое окно (рисунок 35), в котором предлагается перевести его в режим **Наладка**.

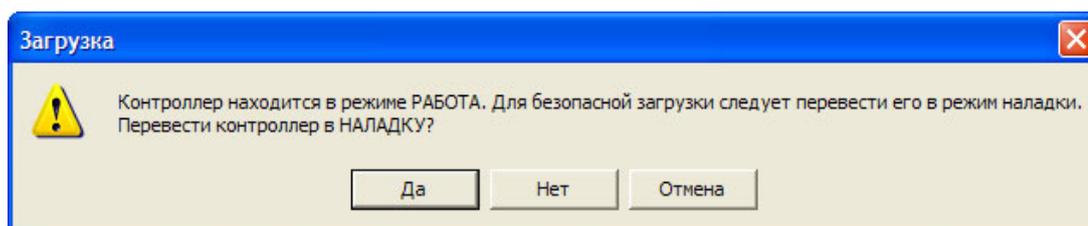


Рисунок 35 - Диалоговое окно продолжения процесса загрузки

После соответствующего ответа начинается процедура загрузки, которая заканчивается следующим сообщением в открывающемся окне (рисунок 36).

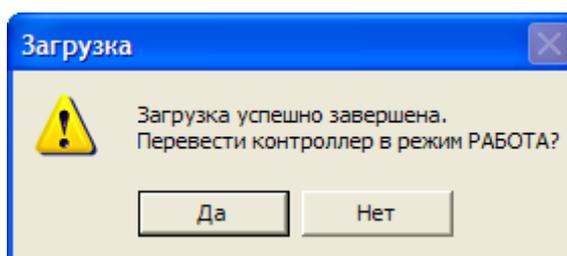


Рисунок 36 – финальное диалоговое окно загрузки

Здесь можно вернуться в режим контроллера **Работа** или остаться в режиме **Наладка**.

## 4.2 Обзор

### 4.2.1 Возможности обзора

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

Перевести в режим обзора можно следующие элементы техпрограммы:

- отдельный вход или выход алгоблока;
- отдельный канал многоканального алгоблока;
- целиком алгоблок;

Инв. № подл.	
Подп. и дата	
Взаим. инв. №	
Инв. №	
Подп. и дата	

Изм	Лист	№ докум	Подп.	Дата

– целиком задачу.

Чтобы перевести в режим обзора отдельный вход или выход, следует нажать правой клавишей мыши соответствующее поле и из открывшегося контекстного меню выбрать команду **Старт**. В открывшемся диалоговом окне выбираются те же параметры, что и при загрузке программы. Кроме того выбрав в главном меню пункт **Вид** и в нем команду **Настройки...**, необходимо установить дополнительный параметр **Блокировать контроллер во время обзора**. Если соответствующая галочка установлена, другие пользователи не смогут редактировать текущую задачу, пока обзор не будет отменен. Если галочка не стоит, задача не блокируется.

В режиме обзора рядом с выбранным полем входа (выхода) появится обозначенное зеленым цветом «живое» значение сигнала, полученное непосредственно из контроллера. Если на входе имеется адресная ссылка, то чтобы увидеть значение сигнала в обзоре, нужно подвести курсор мыши к полю соответствующего входа, - значение сигнала будет выведено поверх ссылки.

Аналогично можно запустить или остановить обзор отдельного канала многоканального алгоблока или всего алгоблока в целом. Разница лишь в том, что в первом случае обзор запускается и останавливается через поле канала, а во втором – через шапку алгоблока. Обзор канала означает обзор всех входов и выходов выбранного канала, а обзор алгоблока – всех входов и выходов выбранного алгоблока. При отмене обзора канала (алгоблока) отменяется обзор всех входов и выходов канала (алгоблока) не зависимо от того, устанавливался ли перед этим обзор групповым или индивидуальным методом.

Чтобы перевести в режим обзора или вывести из обзора всю задачу, нужно из главного меню выбрать команду соответственно **Технологическая программа/Старт** или **Технологическая программа/Стоп**. 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, не зависимо от того, устанавливался ли перед этим обзор групповым или индивидуальным методом.

Все это можно выполнять последовательно для разных входов, каналов, алгоблоков и задач. Тем самым, можно создать любую комбинацию одновременно наблюдаемых «живых» сигналов.

Обзор защитных контроллеров имеет ряд особенностей, которые обсуждаются ниже в 5.8.

#### 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 поставить галочку **Блокировать контроллер во время обзора** в команде **Настройки** пункта **Вид** главного меню (см. 3.3).

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

- 1 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).
- 2 Если обзор блокирующий, то можно выполнять операции, указанные в поз.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 поз. 2 следует обратить особое внимание. Из него следует, что для того, чтобы «на ходу» изменять параметры в контроллере, следует перейти в режим обзора и обязательно сделать этот обзор блокирующим.

### 5 Рекомендации по подготовке технологических программ

#### 5.1 Информационные ресурсы

При подготовке техпрограмм следует учитывать следующие ограничения:

Максимальное число задач в одном контроллере – 256. На общее число задач в одной АСУ ТП ограничений не накладывается (они могут быть связаны лишь с ограничениями на общее число контроллеров в системе).

Максимальное число алгоблоков в одном контроллере – 1500. На распределение алгоблоков по задачам ограничений не накладывается. Это означает, что 1500 алгоблоков с любыми модификаторами могут находиться как в одной, так и в разных задачах, - лишь бы сумма всех алгоблоков по всем задачам не превысила 1500.

Инв. № подл.	Подп. и дата	Взаим. инв. №	Инв. №	Подп. и дата

При работе с Пилоном уже использованные и оставшиеся информационные ресурсы по выбранному контроллеру можно увидеть, выбрав в главном меню раздел **Технологическая программа** и в нем строку **Ресурсы**, либо нажав на соответствующий значок на шкале инструментов, - откроется диалоговое окно (рисунок 1) с информацией о максимально возможных, использованных и оставшихся информационных ресурсах. Следует учитывать, что информация в этом окне обновляется только в момент его открытия, поэтому, если изменять состав алглоблоков при открытом диалоговом окне, информация в нем обновляться не будет.

Пилон контролирует оставшиеся ресурсы и, если они окажутся исчерпанными, информирует об этом пользователя (с указанием на то, какой ресурс исчерпан).

## 5.2 Ввод/вывод информации через модули УСО

Для физического ввода-вывода сигналов в контроллере используются специальные модули – устройства связи с объектом (УСО). В библиотеке алгоритмов имеется специальная группа алгоритмов, организующая связь с этими модулями. Эти алгоритмы называются так же, как и модули УСО, которые они обслуживают. В таблице 2 приведен перечень алгоритмов ввода/вывода.

Таблица 2 – Модули УСО и их алгоритмы

Алгоритмы ввода/вывода		Обслуживаемые модули УСО
Шифр	Наименование	
АЦП	Аналого-цифровой ввод	АЦП-80, АЦП-83, АЦП-84, АЦП-90, АЦП-93, АЦП-94
ДЦП	Дискретно-цифровой ввод	ДЦП-80, ДЦП-81, ДЦП-90
АВВ	Аналого-цифровой ввод/вывод	АВВ-81
ДВВ	Дискретный ввод/вывод	ДВВ-80, ДВВ-81, ДВВ-90, ДВВ-91
ИЦП	Импульсно-цифровой ввод	ИЦП-80, ИЦП-90
ЦАП	Цифро-аналоговый вывод	ЦАП-80, ЦАП-90
ЦДП	Цифро-дискретный вывод	ЦДП-80, ЦДП-81, ЦДП-90, ЦДП-92
ЦИП	Цифро-импульсный вывод	ЦИП-80
КПП	Контроль приборных параметров	КПП-90
МЗТ	Мониторинг защиты турбины	МЗТ-81
МЧТ	Мониторинг частоты турбины	МЧТ-81

В Пилоне каждый алгоритм ввода/вывода обслуживает один (и только один) модуль УСО, размещенный в определенном посадочном месте каркаса. Номер этого места задается на специальном входе алгоритмов ввода/вывода, как это показано ниже на примере алгоритма АЦП. Здесь значение Nпоз=5 означает, что данный алгоритм обслуживает модуль АЦП, находящийся на 5-й позиции. Если установить Nпоз=0, то данный алгоритм не будет обслуживать ни один из модулей УСО

Вход Nпоз имеется у всех алгоритмов ввода-вывода. Алгоритм АЦП имеет дополнительный вход Стип, на котором указывается тип модуля АЦП, с которым связан данный алгоритм.

Алгоритмы ввода/вывода, чьи сигналы используются в разных задачах, рекомендуется группировать в отдельной задаче (или задачах).

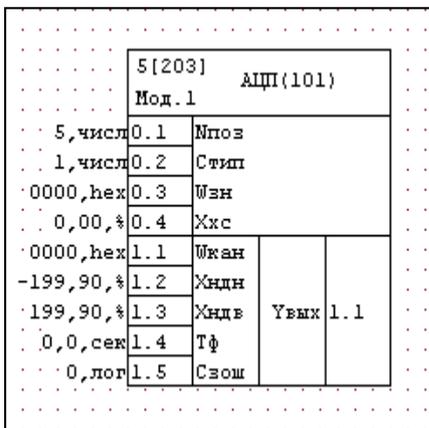
В противном случае эти алгоритмы размещаются в одной задаче вместе с другими алгоритмами.

## 5.3 Связь между контроллерами

Связь между двумя контроллерами организуется с помощью двух алгоритмов – информационного вывода (ИВЫ) и информационного ввода (ИВВ). Контроллер с алгоритмом ИВЫ является источником информации, а контроллер с алгоритмом ИВВ – ее приемником. Размещение в информационной сети источника и приемника может иметь различные варианты:

- источник и приемник принадлежат одному проекту АСУ ТП и находятся внутри одного системного модуля;
- источник и приемник принадлежат одному проекту АСУ ТП и находятся в разных системных модулях;

Ив. № подл.	Подп. и дата
Взаим. ив. №	Ив. №
Подп. и дата	Подп. и дата



- источник и приемник принадлежат разным проектам АСУ ТП. Каждый из перечисленных выше вариантов может иметь еще две особенности:
  - а) один контроллер может быть обычным, другой – элементом защитного кластера;
  - б) оба контроллера могут быть кластерами.

Для всех случаев конфигурация осуществляется с помощью настройки алгоритма ИВЫ в источнике и алгоритма ИВВ в приемнике.

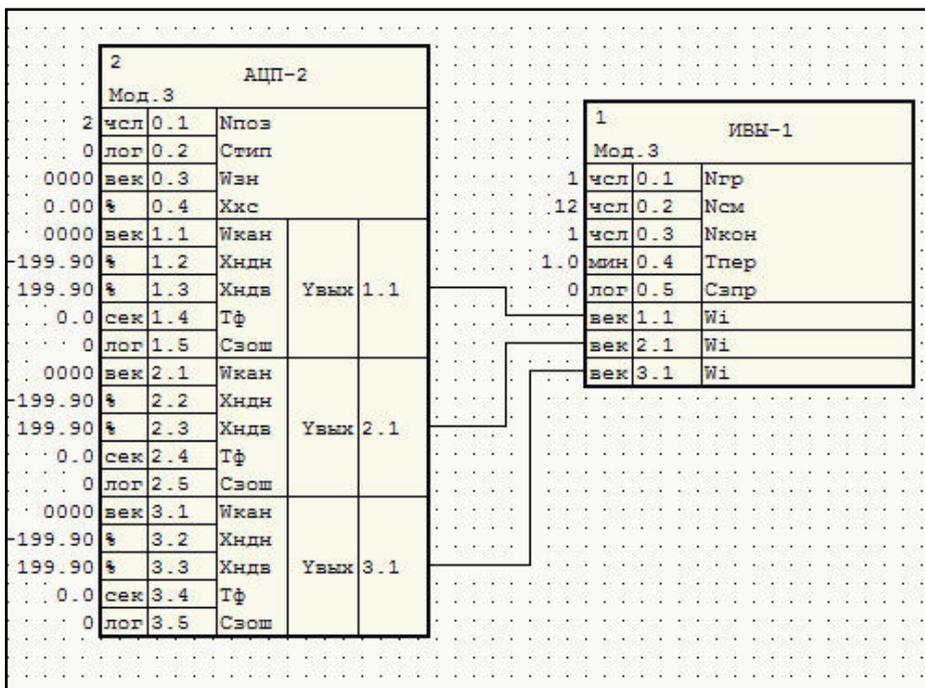
На рисунке 37 показан пример организации такой конфигурации.

В источнике передаваемые сигналы подключаются к алгоритму ИВЫ. Настройка этого алгоритма включает в себя установку следующих параметров:

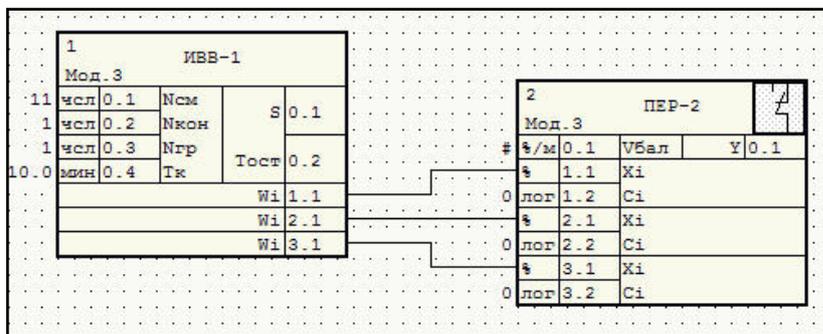
- номер группы (Nгр);
- сетевой номер системного модуля контроллера-приемника (Nсм);
- сетевой номер контроллера-приемника (Nк);
- период передачи информации (Тпер.).

В пределах техпрограммы одного контроллера параметры Nгр, Nсм, Nк совместно должны быть уникальны для каждого канала связи.

В приемнике сигналы принимаются через алгоритм ИВВ и затем могут использоваться в других алгоритмах. Настройка алгоритма ИВВ заключается в указании адреса источника (Nсм и Nк), номера группы алгоритма ИВЫ в источнике, величины тайм-аута (Тк) для контроля целостность информационного канала связи.



а) Контроллер источник



б) Контроллер приемник

Рисунок 37 - Пример интерфейсной связи между контроллерами

Инв. № подл.	Подп. и дата
Взаим. инв. №	Инв. №
Подп. и дата	Подп. и дата

Изм	Лист	№ докум	Подп.	Дата
-----	------	---------	-------	------

Для источника параметр Ngr устанавливается автоматически при вызове алгоритма ИВЫ.

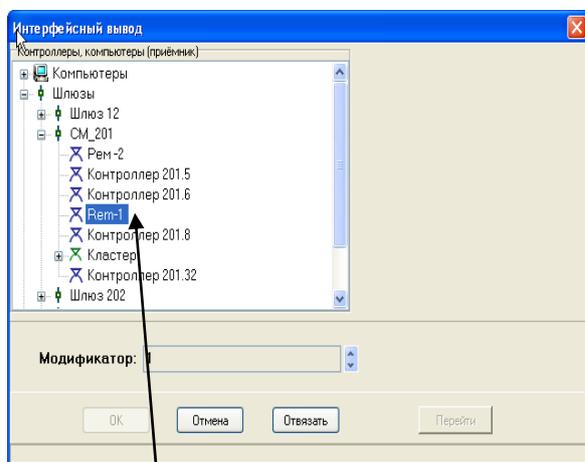
Параметр Tпер для ИВЫ и Tk для ИВВ устанавливаются непосредственно с клавиатуры или в дополнительном диалоговом окне, вызываемом щелчком мыши в поле алгоритма.

Параметры Nсм и Nкон для источника необходимо выбрать из диалогового окна (рисунок 38а). Для этого в поле любого из этих входов алгоритма ИВЫ следует дважды щелкнуть левой клавишей мыши или щелкнуть правой клавишей и затем выбрать команду **Значение на входе**, - откроется иерархический список имеющихся в проекте архивных станций и контроллеров. Если информация должна записываться в архив, выбирается одна из архивных станций, если информацию нужно передать в другой контроллер, выбирается системный модуль и требуемый контроллер

Аналогично можно действовать и для приемника, - разница лишь в том, что здесь в диалоговом окне (рисунок 38б) выбирается не только контроллер-источник, но также и задача, а в ней конкретный алгоблок ИВЫ, с которым нужно установить связь.

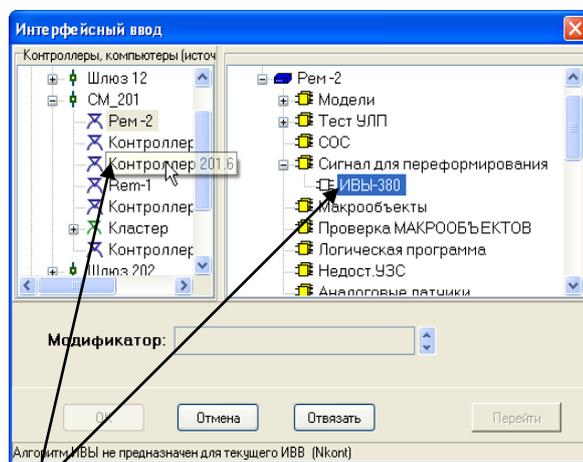
После этих операций слева от соответствующих входов алгоритмов ИВЫ и ИВВ будут проставлены необходимые константы, указывающие на сетевые номера приемника и источника.

Более подробно свойства алгоритмов интерфейсного ввода-вывода описаны в библиотеке алгоритмов



Для источника из списка выбирается приемник (системный модуль и контроллер)

а)



Для приемника из списка выбирается источник (системный модуль, контроллер, задача, алгоблок ИВЫ)

б)

**Рисунок 38 - Выбор приемника и источника через диалоговые окна**

При организации связей между контроллерами с целью минимизации нагрузки на информационную сеть следует учитывать следующие рекомендации:

- дискретные сигналы, направляемые одному контроллеру, предварительно упаковывать с помощью алгоритма шифрации;
- использовать возможно меньшее число групп, передавая в одной группе большее число сигналов;
- устанавливать максимально допустимый (с точки зрения задач управления) период передачи;
- чтобы избежать резкого увеличения пиковой нагрузки на Шлюзы, рекомендуется в разных группах и разных контроллерах разносить период передачи, так чтобы эти периоды не были кратными друг другу (это минимизирует вероятность появления моментов времени, когда одновременно в сеть посылается информация от большого числа групп);
- если в обмене участвует множество контроллеров, для уменьшения нагрузки на Шлюз следует по возможности большую часть сигналов от разных контроллеров-источников одного системного модуля переслать сначала одному из контроллеров этого системного модуля, там объединить их в одну группу, передать в один из контроллеров другого

Инв. № подл.	Подп. и дата
Взаим. инв. №	Инв. №
Подп. и дата	Подп. и дата

системного модуля; затем на стороне приемника распределить эти сигналы по отдельным контроллерам-приемникам;

- поскольку для связи контроллеров, относящихся к разным системным модулям, используется сообщение в виде датаграммы (при этом отсутствует подтверждение о доставке сообщения), следует избегать пересылки между системными модулями коротких неповторяющихся сигналов.

Ниже описывается специфика конфигурации для различных вариантов размещения источника и приемника в информационной сети.

Особенности связи «обычных» контроллеров с защитными контроллерами, входящими в кластер, рассматриваются в 5.8.

### 5.3.1 Источник и приемник принадлежат одному проекту АСУ ТП и находятся внутри одного системного модуля

При установке настроек алгоритма ИВЫ источника в диалоге предлагается выбрать тип информационного обмена:

- по сеансам;
- по датаграммам.

Необходимо указать тип **по сеансам**.

При выборе типа обмена **по датаграммам** параметр Ncm должен быть равен 0.

В алгоритме ИВВ приемника, если он находится в одном системном модуле с источником, параметр Ncm всегда равен 0, независимо от типа обмена.

### 5.3.2 Источник и приемник принадлежат разным проектам АСУ ТП

Для этого варианта разные АСУ ТП должны быть объединены единой системной сетью Ethernet. В Администраторе проекта (подробно описано в 2.2.7 документа «Комплексы программно-технические Квинт-6. Администрирование проектов АСУ ТП. Руководство пользователя ПФДИ.421457.009 Из. 3»), где находится контроллер-источник, должны быть прописан контроллер-приемник и его Шлюз с указателями **Принадлежит другому проекту**.

Аналогично в Администраторе проекта, где находится контроллер-приемник, должны быть указаны Шлюз и контроллер источника.

В этом случае конфигурацию связей можно организовать с помощью диалоговых окон (рисунок 38).

### 5.3.3 Источник – одиночный или резервированный контроллер, приемник – кластер

Независимо от того, находятся ли источник и приемник в одном системном модуле или в разных модулях или в разных проектах, источник должен иметь два алгоритма ИВЫ с одинаковой информацией на входах, у которых параметр Nкон должен соответствовать сетевому номеру соответствующего контроллера кластера в системном модуле. При этом параметры Ncm, Nгр у обоих ИВЫ должны быть одинаковыми. В технологической программе приемника должен быть один алгоритм ИВВ с соответствующими параметрами Ncm, Nкон, Nгр.

### 5.3.4 Источник – кластер, приемник – одиночный или резервированный контроллер

Независимо от того, находятся ли источник и приемник в одном системном модуле или в разных модулях или в разных проектах, приемник должен иметь два алгоритма ИВВ, у которых параметр Nкон должен соответствовать сетевому номеру соответствующего контроллера кластера в системном модуле. При этом параметры Ncm, Nгр у обоих ИВВ должны быть одинаковыми.

В технологической программе источника должен быть один алгоритм ИВЫ с соответствующими параметрами Ncm, Nкон, Nгр.

### 5.3.5 Источник и приемник – кластеры

Независимо от того, находятся ли источник и приемник в одном системном модуле или в разных модулях или в разных проектах, источник должен иметь два алгоритма ИВЫ с одинаковой информацией на входах, у которых параметр Nкон должен соответствовать сетевому номеру соответствующего контроллера кластера в системном модуле. При этом параметры Ncm, Nгр у

ИВ. № подл.	Подп. и дата
Взаим. инв. №	ИВ. №
Подп. и дата	Подп. и дата

Изм	Лист	№ докум	Подп.	Дата	ПФДИ.421457.009 Из.2	Лист
						49

обоих ИВЫ должны быть одинаковыми. Для этого Ngr, который в диалоге устанавливается автоматически

В технологической программе приемника должны быть два алгоритма ИВВ с соответствующими параметрами Ncm, Nкон, Nгр контроллеров источника.

## 5.4 Каскадные связи

В цепочке последовательно включенных алгоритмов часто необходимо, чтобы предварительно включенные алгоритмы, отслеживали состояние последующих алгоритмов и, тем самым, обеспечивали безударное изменение режимов управления.

В библиотеке имеется специальная группа так называемых каскадных алгоритмов, которые обеспечивают эту возможность. К ним относятся:

- РАН (18) - Регулятор аналоговый;
- РИМ (19) - Регулятор импульсный;
- ДИН (24) - Динамическое преобразование;
- ИНТ (25) – Интегрирование;
- СУМ (32) - Суммирование с масштабированием;
- УМД (33) - Умножение-деление;
- ПЕК (64) - Переключатель каскадный;

Общим для этих алгоритмов является то, что они имеют специальный вход слежения Wсл и специальный выход обратного счета Wос. Чтобы организовать каскадную связь, выход Wос последующего алгоритма соединяется со входом Wсл предыдущего алгоритма так, как это показано (рисунок 39) на примере каскадного регулятора. В единую каскадную цепочку можно выстроить любое число каскадных алгоритмов, при этом каждая соседняя пара алгоритмов соединяется указанным выше способом.

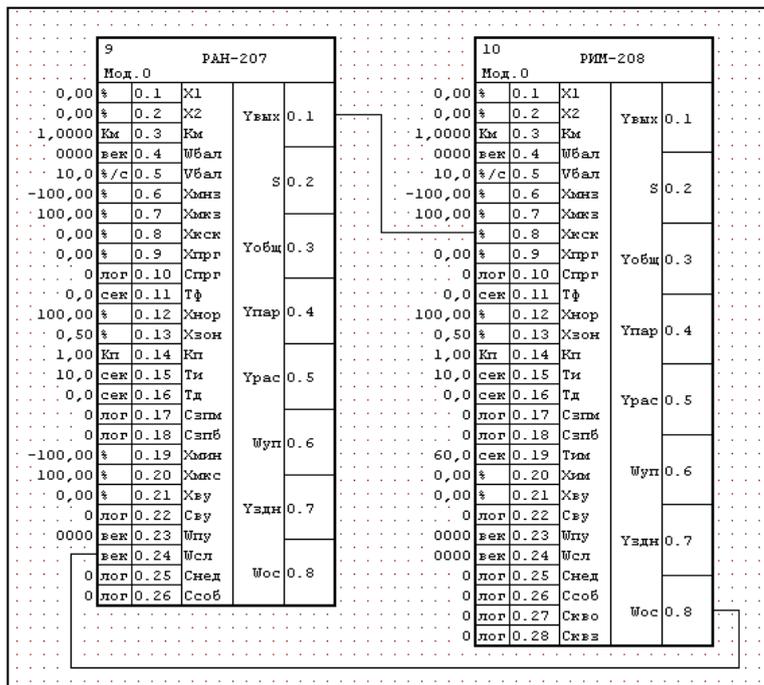


Рисунок 39 - Пример каскадного регулятора

## 5.5 Объектные алгоритмы

### 5.5.1 Подробнее об объектах

Ранее уже говорилось, что в Квинте имеется понятие информационного объекта. В ряде случаев такой объект отражает свойства реального (технологического) объекта (например, датчика, клапана, задвижки и т.д.). В других случаях информационный объект является виртуальным и не

Инв. № подл.	Взаим. инв. №	Инв. №	Подп. и дата

связан с реальным технологическим объектом (например, регулятор, шаговая программа и т.д.). Далее, там, где это не приводит к неоднозначности, вместо термина **информационный объект** будет использоваться термин **объект**.

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

В большинстве случаев при работе с Пилоном определенному типу объекта соответствует какой-либо объектный алгоритм (например, объекту **Аналоговый датчик** соответствует объектный алгоритм **ФАО - формирователь аналоговый объектный**). 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.

### 5.5.2 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, недостоверность** 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 – для технологических ошибок, и станция приборных ошибок – для ошибок технических средств Квинта).

Специально подготавливать в Ремиконте оперативную информацию о таких объектах не требуется, - после того, как алгоритм с помощью Пилона привязан к марке объекта, вся необходимая информация готова для передачи станциям (это утверждение не касается периодической записи в архив, - перечень архивируемой информации и период архивирования задаются отдельно с помощью специального алгоритма информационного вывода ИВЫ (см. 5.7).

Важное замечание: на входе каждого объектного алгоритма, чья информация о событиях должна записываться в архив, следует установить значение константы  $S_{соб}=1$ ; при  $S_{соб}=0$  события в архив записываться не будут. Кроме того в Администраторе БД следует указать архивную станцию для данного контроллера.

### 5.5.3 Неявные входы объектных алгоритмов

Несколько объектных алгоритмов имеют так называемые неявные входы. Основное назначение неявных входов – логически связать с объектом, представленным объектным алгоритмом, состояние каких-то других сопутствующих сигналов, с тем чтобы отобразить на мнемосхеме состояние этих сигналов вместе с состоянием объекта. Часто (но не всегда) такими сопутствующими сигналами являются признаки срабатывания защит и блокировок, которые приводят к изменению состояния объектного алгоритма (отключению двигателя, закрытию задвижки и т.д.). Дискретные неявные входы (входы условий) позволяют увидеть на мнемосхеме первопричину изменения состояния таких объектов. Как правило, эта первопричина представляется на мнемосхеме в виде соответствующего текста, предварительно задаваемого в Аркаде. К объектам, имеющим по 10 неявных входов условий, относятся задвижки, клапаны, двигатели и АВР.

Инд. № подл.	Подп. и дата
Взаим. инв. №	Инд. №
Подп. и дата	Подп. и дата

Неявные входы не видны на функциональной схеме Пилона. Адреса источников, связанных с неявными входами, которые называются локальными адресами, задаются в диалоговом окне привязки. Эти адреса можно задать не только в Пилоне, но и непосредственно в Аркаде.

Чтобы задать локальные адреса, следует воспользоваться экранной клавишей **Лок.адреса** в окне привязки. При нажатии этой клавиши открывается диалоговое окно с перечнем локальных адресов, которые потенциально имеются у выбранного алгоритма (рисунок 40). Затем следует воспользоваться значком **Привязать вход или выход алгоблока**, после чего из открывающейся последовательности диалоговых окон выбрать контроллер, задачу, алгоблок, вход или выход и, наконец, конкретный сигнал привязки. С помощью экранной клавиши **Отвязать вход или выход алгоблока** любой привязанный локальный адрес можно отвязать.

Замечание: прежде чем привязывать локальные адреса, следует привязать сам алгоблок или, если он каналный, - канал алгоблока.

Видимые поля в таблице                      Привязать вход или выход алгоблока                      Отвязать вход или выход алгоблока

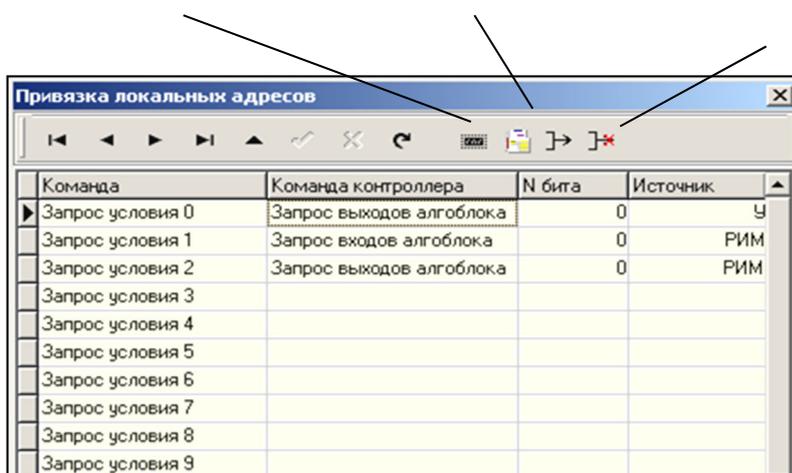


Рисунок 40 - Работа с локальными адресами

## 5.6 Вычислительные алгоритмы

### 5.6.1 Состав вычислительных алгоритмов

Библиотека Ремиконтов 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.

Наряду с этим имеется два специализированных алгоритма, с помощью которых можно выполнять достаточно сложные вычисления – это УНО (Универсальные операции) и ПКП (Программа комплексного параметра). Алгоритм УНО работает самостоятельно, алгоритм ПКП – совместно с алгоритмом ВКП (Вычисление комплексного параметра).

### 5.6.2 Применение алгоритма УНО (Универсальные операции)

Этот алгоритм используется в тех случаях, когда нужно выполнить большое число сравнительно простых арифметических и/или логических операций. Алгоритм является многоканальным, причем в каждом канале выполняется одна элементарная операция.

На входах  $W_i$  алгоритма УНО задается вид функции в каждом канале (рисунок 41). Можно устанавливать требуемые векторные константы непосредственно на входе алгоритма, но можно воспользоваться диалоговым окном, которое открывается по двойному щелчку левой клавиши мыши в области этих входов. Затем можно выбрать вид функции, аргумент и адрес, по которому засылается результат.

Инв. №	Подп. и дата
Взаим. инв. №	
Подп. и дата	
Инв. № подл.	

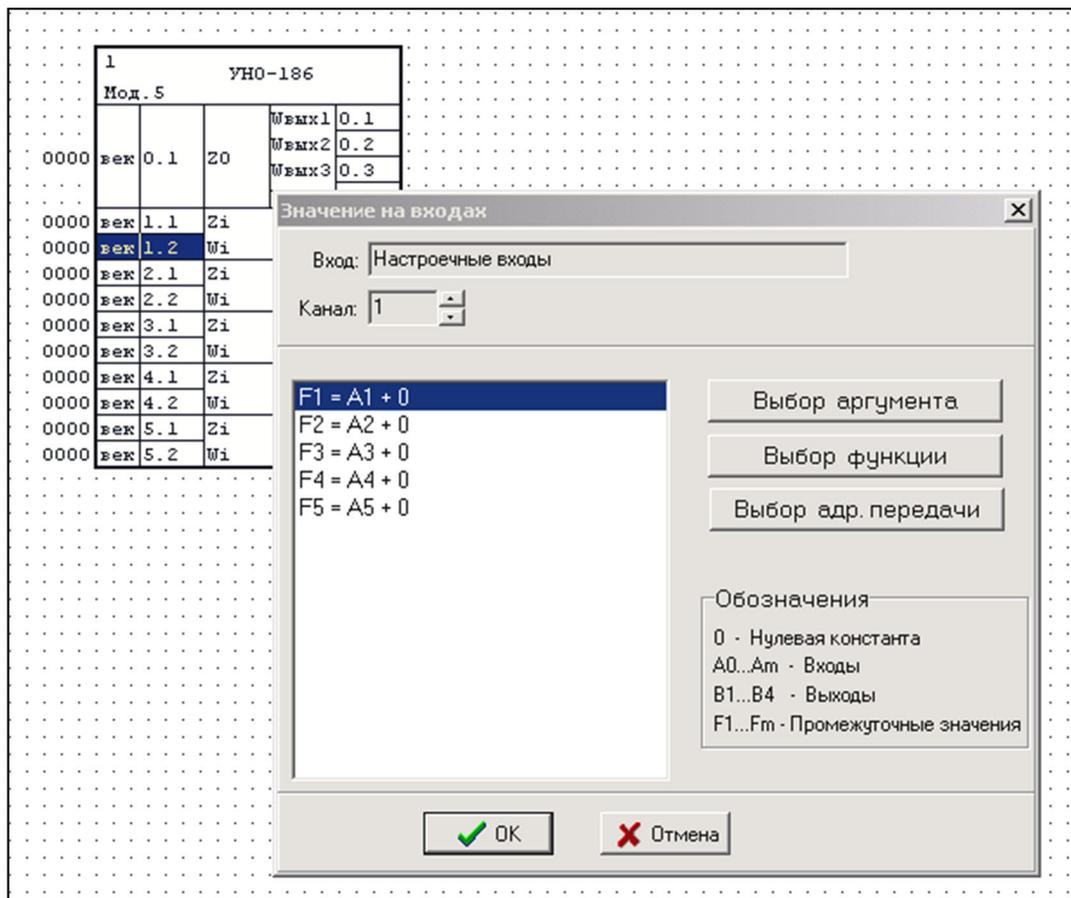


Рисунок 41 - Диалоговое окно программирования алгоритма УНО

### 5.6.3 Применение алгоритма ПКП

Этот алгоритм целесообразно использовать, если нужно выполнять достаточно сложные алгебраические вычисления. Алгоритм работает совместно с алгоритмом ВКП, правила их соединения приведены в описании библиотеки алгоритмов.

Вычислительная функция определяется последовательностью констант, заданных на входах Li алгоритма ПКП. Эти константы можно вводить непосредственно на входе алгоритма, но можно воспользоваться диалоговым окном, которое открывается сразу же после того, как алгоритм ПКП вводится в поле задачи, либо по двойному щелчку левой клавиши мыши в шапке алгоблока. Если в этом окне дважды щелкнуть левой клавишей мыши в области соответствующего поля, откроется еще одно окно, через которое можно ввести требуемое значение, а если щелкнуть правой клавишей – выбрать требуемую инструкцию.

**Примечание** - При выборе алгоритма ПКП можно оставлять модификатор, равный 1. После того, как в диалоговом окне будут введены необходимые инструкции, модификатор автоматически примет требуемое значение.

Если в открывшемся окне **Программа комплексного параметра** (рисунок 42) нажать клавишу **Формула**, откроется следующее окно **Редактор формул для алгоритма ПКП**, в котором можно впечатать с клавиатуры или выбрать из списка требуемые функции, получив, тем самым, требуемую формулу (рисунок 42). Чтобы проверить, правильно ли введена формула, нужно нажать экранную клавишу **Проверка**. Если ошибок нет, следует нажать клавишу **ОК**, - формульное окно закроется, в окне со списком инструкций установятся нужные инструкции, а на входах алгоритма ПКП - нужные константы.

Инв. № подл.	Подп. и дата
Взаим. инв. №	Инв. №
Подп. и дата	Подп. и дата

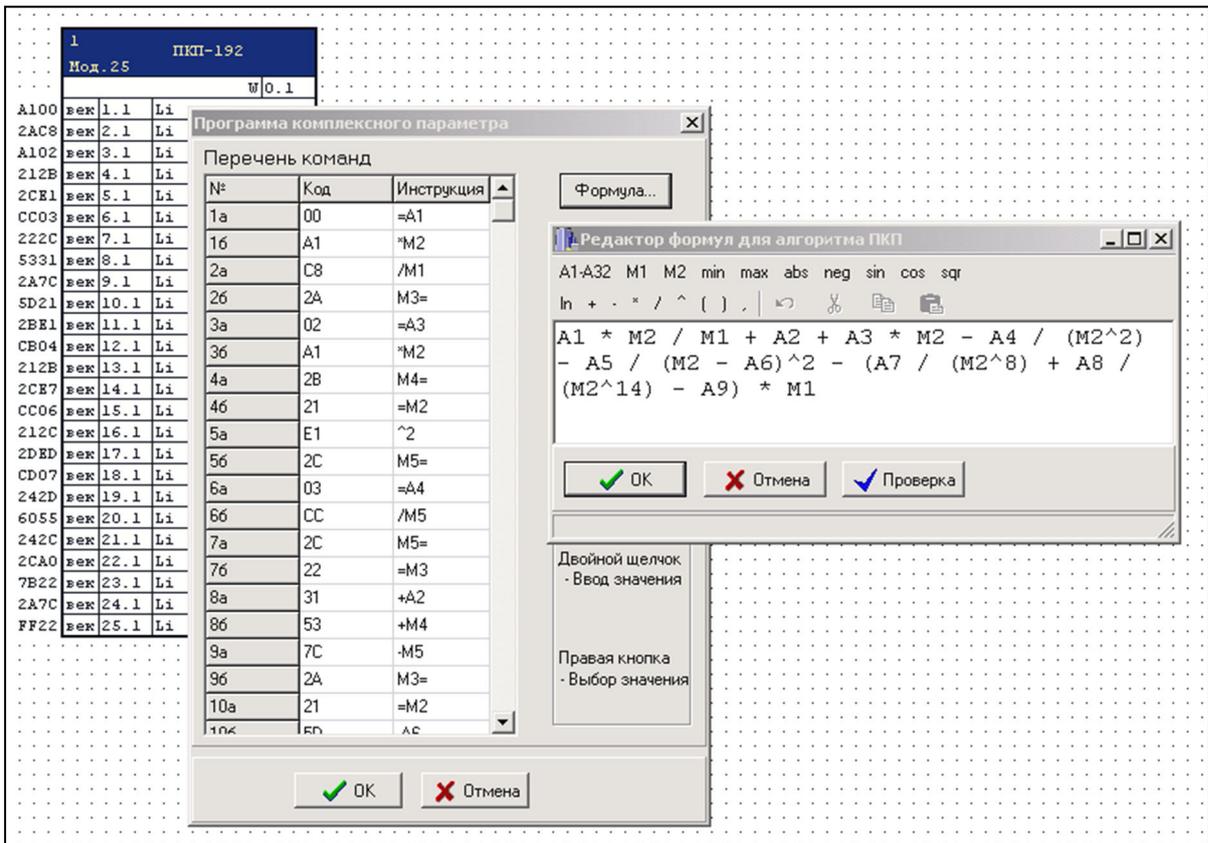


Рисунок 42 - Диалоговое окно программирования алгоритма ПКП

## 5.7 Архивирование сигналов

### 5.7.1 Варианты архивирования

В Квинте реализованы 4 метода архивирования сигналов:

- периодическое;
- апертурное;
- объектное;
- событийное.

Периодическое архивирование сигналов выполняется регулярно с заданным периодом времени. При апертурном архивировании сигналы записываются в архив только при их изменении. Объектное архивирование предполагает, что в архив записываются не значения одиночных сигналов, а вся информация, относящаяся к архивируемым объектам. Событийное архивирование фиксирует в архиве все происходящие события – как штатные, так и не штатные (ошибки).

### 5.7.2 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.

В Пилоне для периодического архивирования используется алгоритм ИВЫ - информационный вывод. Каждый вход этого алгоритма «отвечает» за один не упакованный (например, аналоговый) архивируемый сигнал или одну упаковку упакованных сигналов (например, за 16 дискретных признаков). В качестве параметра настройки для алгоритма задается период архивирования, общий для всех сигналов одного алгоритма. Отсюда вывод: *с одним алгоритмом ИВЫ связываются сигналы, имеющие одинаковый период архивирования; если период архивирования должен быть разным, нужно использовать несколько алгоритмов ИВЫ.*

Если на входе Сзпр алгоритма ИВЫ установить не константу, а связать этот вход с другим алгоритмом, можно динамически разрешать или запрещать запись всей группы сигналов в архив. Аналогично, управляя входом Тпер, можно динамически менять период записи в архив для всей группы сигналов.

Инв. №	Подп. и дата
Взаим. инв. №	
Подп. и дата	
Инв. № подл.	

Архивироваться могут сигналы, сформированные любым как объектным, так и не объектным алгоритмом. При этом могут встретиться следующие случаи:

Архивируются выходные сигналы объектного алгоритма. В этом случае каждый его выход (каждый аналоговый сигнал или каждая упаковка, содержащая совокупность дискретных признаков) напрямую (без промежуточных алгоритмов) соединяются с одним из входов  $W_i$  алгоритма ИВЫ. В архив при этом будет записываться сконфигурированный аналоговый сигнал и все признаки, содержащиеся в упаковке. Каждый архивируемый сигнал привязывать к марке объекта не требуется, - достаточно привязать к марке сам объектный алгоритм или отдельный его канал.

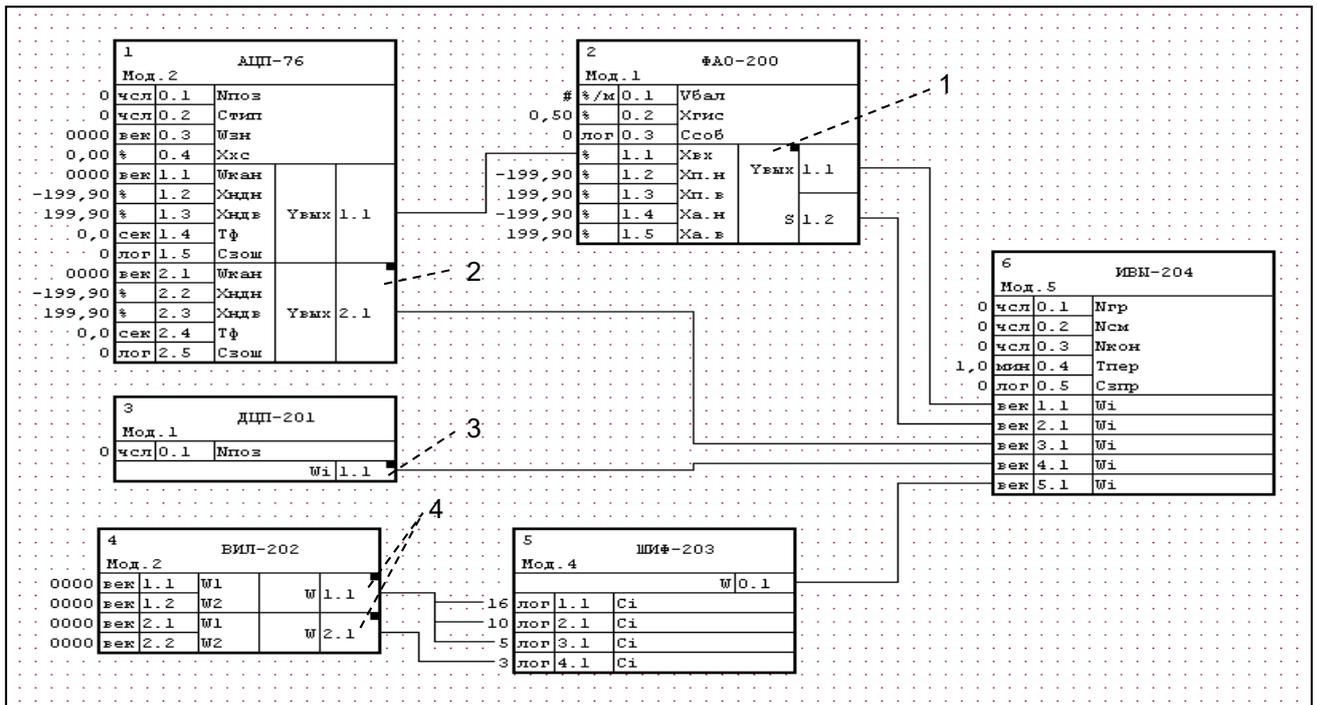
Архивируется аналоговый выходной сигнал не объектного алгоритма. Такой выход напрямую (без промежуточных алгоритмов) соединяется с одним из входов  $W_i$  алгоритма ИВЫ и привязывается к марке объекта.

Архивируются одиночные дискретные выходные сигналы (биты) не объектных алгоритмов. В принципе, каждый такой сигнал можно индивидуально подключить ко входу алгоритма ИВЫ, но это весьма не экономный способ, так как каждое достаточно длинное сообщение, отправляемое в архив, будет содержать всего один бит технологической информации. Предпочтительно вначале упаковать отдельные биты с помощью алгоритма шифрации ШИФ (максимум по 16 сигналов) и уже выход этого алгоритма напрямую (без промежуточных алгоритмов) связать с одним из входов алгоритма ИВЫ. В таком варианте к марке привязывается каждый выходной архивируемый бит.

Примечание: Для того, чтобы выбранные сигналы записывались в архив, в алгоритме ИВЫ на входе  $N_{см}$  должен быть задан сетевой номер архивной станции, а на входе  $N_{кон}$  всегда  $N_{кон}=0$ .

На рисунке 43 иллюстрируются различные варианты периодического архивирования сигналов. 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 (5.7.5).

Инв. № подл.	Подп. и дата	Взаим. инв. №	Инв. №	Подп. и дата	ПФДИ.421457.009 И3.2					Лист
										55
					Изм	Лист	№ докум	Подп.	Дата	



1 – архивируются аналоговый сигнал (Yвых) и все его признаки (S), сформированные объектным алгоритмом. К марке объекта привязывается канал объектного алгоритма (отдельно выходы привязывать не требуется).  
 2 – архивируется аналоговый сигнал, сформированный на выходе необъектного алгоритма. К марке привязывается выход этого алгоритма.  
 3- архивируются упакованные дискретные сигналы, сформированные на выходе необъектного алгоритма. К марке привязывается каждый архивируемый бит выходного сигнала.  
 4 – архивируются отдельные дискретные сигналы, сформированные на выходах необъектного алгоритма.  
 Перед тем, как поступить на вход алгоритма ИВЫ, все четыре дискретных сигнала упаковываются шифратором ШИФ. К марке привязывается каждый архивируемый бит.

**Рисунок 43 - Примеры периодического архивирования сигналов**

### 5.7.3 Архивирование аналоговых сигналов по апертуре

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

Для апертурного архивирования схема и правила такие же, как для периодического, - разница лишь в том, что вместо алгоритма информационного вывода ИВЫ используется алгоритм выделения аналоговых событий ВАС.

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).

### 5.7.4 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.

Инв. № подл.	Подп. и дата	Взаим. инв. №	Инв. №	Подп. и дата
--------------	--------------	---------------	--------	--------------

Имеется еще один вид объектного архивирования, которое инициирует архивная станция. Эта станция периодически запрашивает состояние всех объектов, формируемых всеми контроллерами, описанными в базе данных. Если состояние какого-либо объекта изменилось, вся объектная информация, относящаяся к данному объекту, записывается в архив. Например, если регулятор переведен на ручной режим или зафиксировано, что исполнительный механизм дошел до конечного выключателя, вся объектная информация об этом регуляторе или задвижке будет записана в архив.

Таким образом, оба вида объектного архивирования не требуют программирования. Разница между ними заключается в том, что в первом случае в архив записывается информация об одном конкретном событии, причем она записывается сразу же, как только событие возникло. Во втором случае записывается вся совокупность параметров, характеризующих объект, причем запись ведется периодически с равными интервалами времени, - в каждом контроллере архивная станция по кругу 1 раз в 1 сек запрашивает эту информацию у одного очередного объекта. Например, если не зависимо от числа контроллеров, в системе в каждом из них с помощью Пилона сформировано по 60 объектов, то среднее время опроса каждого объекта во всех контроллерах составит 1 мин. Такая «фооновая» архивация позволяет воспроизвести поведение всех объектов за длительный промежуток времени. Особенно полезна она при «проигрывании» архива и при работе станции руководителя.

### 5.7.5 Программирование событий

Выше отмечалось, что большинство объектных алгоритмов формируют события, которые автоматически (без специального программирования) направляются в архив. Тем не менее, может возникнуть необходимость зафиксировать в архиве события, связанные с изменением состояния отдельных признаков, не относящихся к объектным алгоритмам. Для этих целей используется алгоритм выделения дискретных событий ВДС.

Для того, чтобы запрограммировать набор архивируемых событий, каждый дискретный вход алгоритма ВДС соединяется с дискретным признаком, изменение состояния которого должно фиксироваться в архиве как событие. На данном примере такими событиями являются изменение знака выходного сигнала сумматора СУМ и изменение признака недостоверности этого сигнала (соответственно биты 16 и 1), а также изменение выходного дискретного сигнала вибратора ВИБ.

Очевидно, что все дискретные сигналы, архивируемые в качестве событий, должны быть описаны в базе данных. При использовании ВДС (рисунок 44) можно воспользоваться одним из двух вариантов привязки к базе данных:

- привязать источники событийной информации,
- привязать адекватные им выходы статуса S алгоритма ВДС (результат будет одинаковым).

Инв. № подл.	
Подп. и дата	
Взаим. инв. №	
Инв. №	
Подп. и дата	

**Рисунок 44 - Пример программирования событий**

**5.7.6 Программирование ошибок**

По аналогии с событиями, большинство объектных алгоритмов формируют ошибки, которые автоматически (без специального программирования) направляются всем станциям ИВК, способным распознавать ошибки (в частности, они записываются в архив). Тем не менее, может возникнуть необходимость связать ошибку с изменением состояния отдельных признаков, не относящихся к объектным алгоритмам. Для этих целей используются алгоритмы фиксации ошибочных ситуаций ФОС или сигнализатор дискретный объектный СДО.

В обоих случаях входы этих алгоритмов связываются с сигналами, изменение состояния которых должно рассматриваться как ошибка. Как обычно, архивируемые сигналы предварительно должны быть описаны в базе данных.

С помощью алгоритма ФОС не только фиксируются сами ошибки, но и определяется их категория и код (вход Nкод). Это позволяет различным образом интерпретировать ошибки при их представлении в рабочих станциях ИВК (см. описание алгоритма ФОС в библиотеке алгоритмов). *Когда событийное архивирование организуется с помощью алгоритма ФОС, к маркам привязываются не источники сигнала, а входы алгоритма ФОС (рисунок 45).*

Алгоритм СДО является обычным объектным многоканальным алгоритмом, который обрабатывает входные сигналы или команды дисплейного управления и интерпретирует их как ошибки. В алгоритме можно задать вид ошибки – предупредительная или аварийная (см. описание алгоритма СДО в **Библиотеке алгоритмов**).

**Примечание** - когда событийное архивирование организуется с помощью алгоритма СДО, к маркам, имеющим оперативный тип Сигнализатор дискретный, привязываются не источники сигнала и не входы алгоритма СДО, а его каналы.

Инв. № подл.	Подп. и дата	Взаим. инв. №	Инв. №	Подп. и дата	<p align="center">ПФДИ.421457.009 И3.2</p>					Лист
										58
					Изм	Лист	№ докум	Подп.	Дата	

К марке привязывается  
каждый вход ФОС

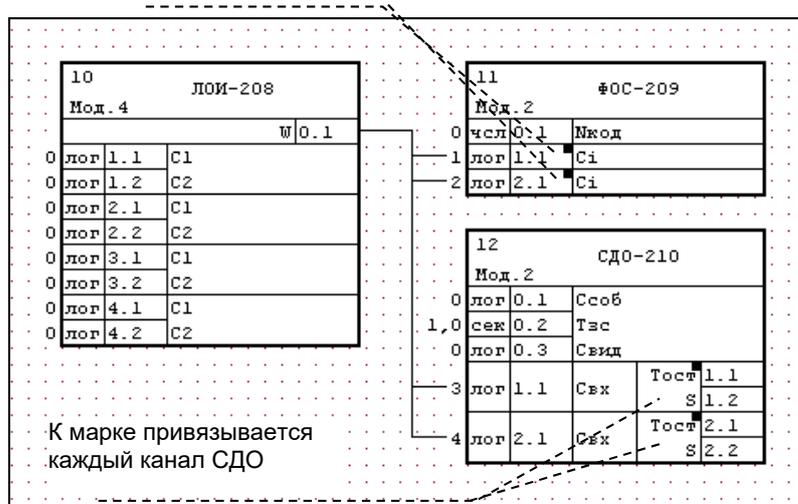


Рисунок 45 - Пример программирования ошибок

## 5.8 Работа с кластерными контроллерами

### 5.8.1 Подготовка технологических программ для кластерных контроллеров

Кластеры используются в составе АСУ ТП для создания подсистем защит технологического оборудования. Кластер состоит из двух одинаковых контроллеров. Для них готовится одна технологическая программа.

Предварительно из списка контроллеров выбирается имя кластера, прописанного в Администраторе, после чего для него выполняются все процедуры по подготовке технологической программы, как для обычного контроллера.

Подготовленная технологическая программа обычным образом компилируется и загружается во все контроллеры кластера по одной команде.

### 5.8.2 Режим обзора для кластеров

Отличие режима **Обзор** при наладке технологических программ кластеров от этого же режима при работе с обычными контроллерами состоит в том, что информация на экран Пилона выводится только от одного контроллера, который называется **Приоритетный**.

По умолчанию приоритет устанавливается автоматически для контроллера кластера с минимальным уровнем ошибок, а если уровень ошибок одинаков, то с минимальным сетевым номером внутри кластера.

Принудительно отдать приоритет любому контроллеру можно с помощью команды Главного меню **Вид/Защитные контроллеры** или с помощью соответствующего значка на панели инструментов (рисунок 9). В результате откроется окно (рисунок 46), в котором необходимо в левом поле щелкнуть по нужному контроллеру правой клавишей мыши и из дополнительного диалогового окна выбрать команду **Сделать приоритетным**, в результате зеленая точка переместится на выбранный контроллер. Следует учитывать, что принудительно установленный приоритет действует только на время вызова окна (рисунок 46), при его закрытии устанавливается приоритет по умолчанию.

На выполнение защитных функций изменение приоритета никак не влияет – в этом отношении все контроллеры кластера всегда активны.

Подп. и дата
Инв. №
Взаим. инв. №
Подп. и дата
Инв. № подл.

Если в режиме **Обзор** кластера изменяется значение констант или конфигурация, новые значения автоматически изменяются в обоих контроллерах независимо от того, какой контроллер в данный момент был приоритетен.

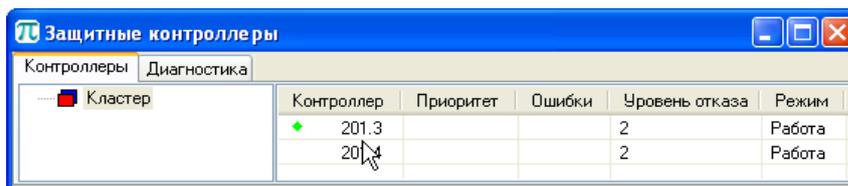


Рисунок 46 - Выбор контроллера для установки принудительного приоритета

## 6 Подготовка логических шаговых программ

Для создания логических шаговых программ используются два специальных алгоритма УЛП (управление логической программой) и ЭТА (этап). Объектный алгоритм УЛП координирует ход выполнения шаговой программой и содержит набор программных битовых выходов, состояния которых могут изменять команды алгоритма ЭТА. Эти выходы конфигурируются с приемниками, обрабатывающими указанные команды.

Входы алгоритмов ЭТА являются последовательными шагами программы. На входах задаются команды или условия хода программы.

Для того, чтобы логическая программа правильно функционировала, должны соблюдаться следующие правила:

Каждый из входов Nэт (номер этапа) алгоритма УЛП должен напрямую (без промежуточных алгоритмов) подключаться к выходу Wвых «своего» этапа ЭТА.

К марке привязывается объектный алгоритм УЛП, привязывать к маркам алгоритмы ЭТА не требуется.

В одной логической программе алгоритм УЛП всегда один, алгоритмов ЭТА может быть до 31. Пример сконфигурированной логической программы из 3-х этапов (параметры шагов не заданы) приведен на (рисунок 47).

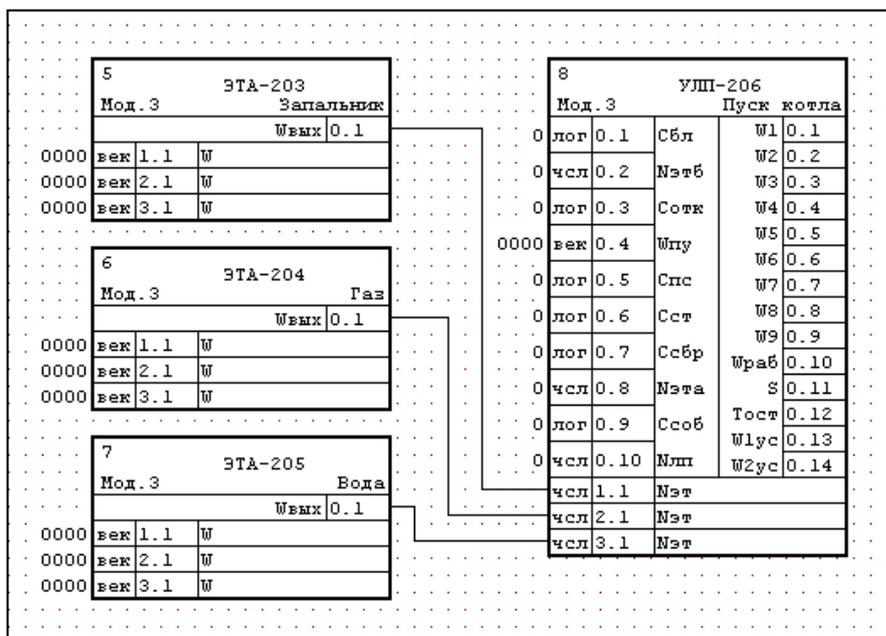


Рисунок 47 - Конфигурирование логической программы (пример)

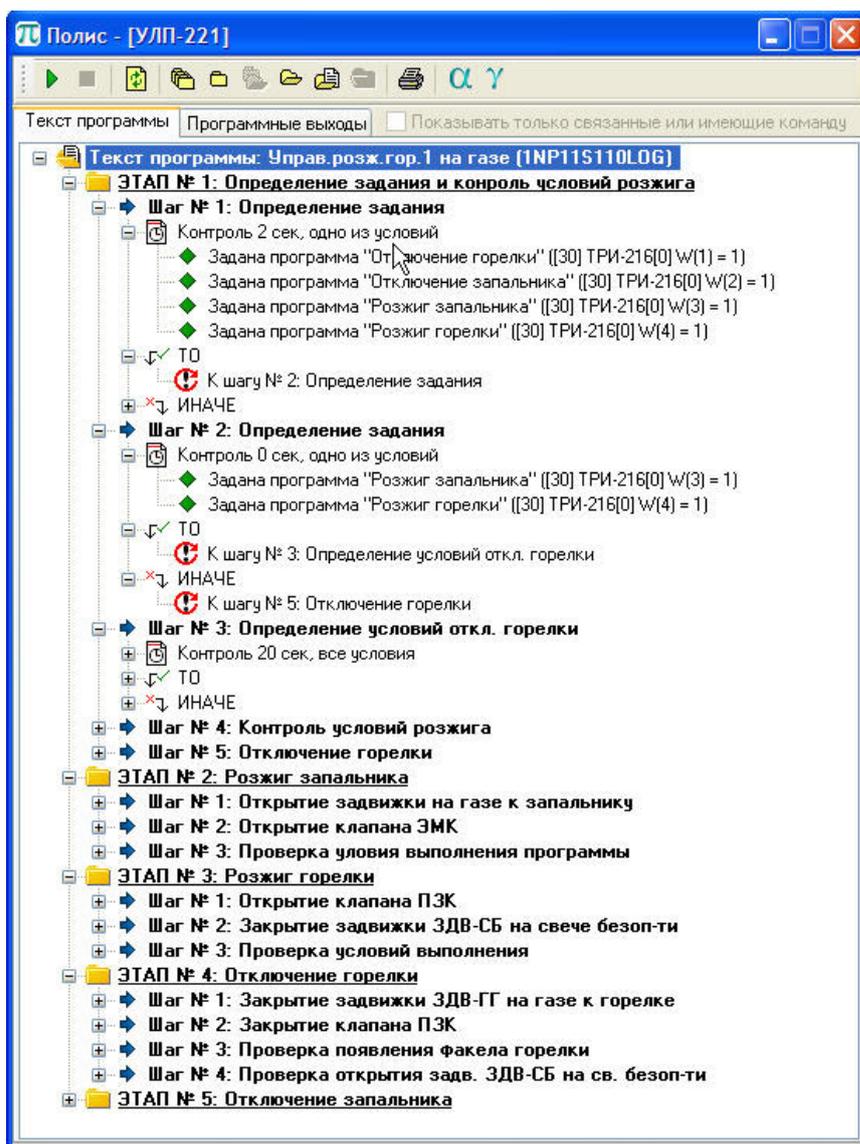
Для создания логической программы Пилон имеет специальную программную компоненту **Полис**, обеспечивающую представление логики шаговой программы в виде записи последовательности текстовых строк.

Инв. №	Подп. и дата
Взаим. инв. №	Подп. и дата
Инв. № подл.	

Вид, в котором шаговая программа представляется на экране Пилона, задается в диалоговом окне настроек, которое открывается по команде **Вид/Настройки** (рисунок 13). Если галочка против указателя **Полис** снята, шаговая программа представляется в виде алглоблоков с алгоритмами УЛП и ЭТА, как на рисунке 47. Если галочка стоит, то на экране Пилона остается только алгоритм УЛП со скрытыми программными выходами, а вся логика программы задается в текстовом виде.

Дополнительно, открыв диалоговое окно **Шрифт Полиса**, можно выбрать вид и размер шрифта, которым будет представляться программа.

В качестве примера на рисунке 48 показан общий вид логической программы в записях Полиса. В тексте программы видна сама логическая программа, ее этапы, шаги, а также команды, условия и переходы шагов. Этапы, шаги, команды могут быть поименованы. Этапы и шаги можно в тексте свернуть или развернуть.



**Рисунок 48 - Вид шаговой программы в окне закладки Текст программы (пример)**

Закладка **Программные выходы** открывает другое окно (рисунок 49). В нем указаны 144 дискретных выхода (9 упакованных выходов W1-W9 по 16 бит в каждом), принадлежащих алгоритму УЛП данной программы. Каждый из 144 выходных битов можно связать с любым приемником. В показанном ниже примере первый бит выхода W1 связан со входом С1 первого канала алгоритма УЗА, а второй бит – со входом С2 того же алгоритма. Как сами выходы, так и назначение связей с приемником могут быть поименованы.

Таким образом, для подготовки логической программы в Полисе имеются две составляющие: запись самой программы в виде последовательности этапов и шагов, и определение связей между командами, сформированными программой, и приемниками этих команд.

Инд. № подл.	Подп. и дата	Взаим. инв. №	Инд. №	Подп. и дата

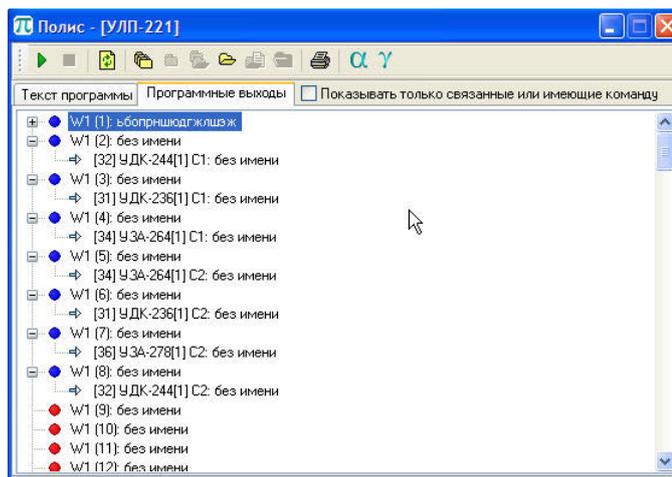


Рисунок 49 - Вид программы в окне закладки Программные выходы (пример)

### 6.1 Шкала инструментов Полиса

Окна Полиса имеют шкалу инструментов (рисунок 50) следующего назначения (здесь значки инструментов обозначены цифрами):

- 1 – запуск режима Обзор.
- 2 – выход из режима Обзор.
- 3 – обновление окна Полиса.
- 4 – свернуть все. Сворачиваются все шаги и этапы и остаются видны только заголовки этапов логической программы.
- 5 - свернуть все шаги. Сворачиваются все шаги и остаются видны все этапы логической программы и все заголовки шагов.
- 6 – свернуть все, кроме текущего шага. Сворачиваются все шаги, кроме выделенного. Данный шаг остается в том виде, в котором он был представлен до нажатия этой кнопки.
- 7 – развернуть все. Разворачиваются все этапы, шаги, команды и условия всех шагов.
- 8 – показать все шаги. Разворачиваются все этапы и становятся видны все заголовки шагов.
- 9 – развернуть текущий шаг. Выделенный шаг полностью разворачивается.
- 10 – экспорт программы в Microsoft Word. Программа в полностью развернутом виде с комментариями к программным выходам переносится в *Microsoft Word* для дальнейшей распечатки или хранения копии.
- 11 – переход в Аркаду. При нажатии на эту кнопку производится открытие приложения Аркада (в другом окне).
- 12 – переход в Графит. При нажатии на эту кнопку производится открытие приложения Графит (в другом окне).



Рисунок 50 - Шкала инструментов Полиса

### 6.2 Структура логической программы

Логическая программа имеет иерархическую структуру и состоит из следующих элементов:

- управление логической программой (алгоритм УЛП);
- этапы;

Инд. № подл.	Подп. и дата	Взаим. инв. №	Инд. №	Подп. и дата

- шаги;
- операторы.

Алгоритм УЛП, описание которого приведено в библиотеке алгоритмов, координирует ход выполнения логической программой. Одна логическая программа всегда содержит только один алгоритм УЛП.

Логическая программа состоит из этапов. Одна логическая программа может содержать до 31 этапа. Сами этапы как таковые никаких действий не выполняют и являются лишь «оболочкой», позволяющей организовывать необходимые переходы и остановки, а также сделать программу более наглядной.

Каждый этап состоит из шагов. Один этап может содержать до 31 шага. Шаги состоят из операторов, в которых анализируются заданные условия выполнения команд и переходов, а также выполняются сами команды. В общем случае шаг состоит из нескольких фрагментов:

- **предварительные команды** – выполняются, безусловно, в начале шага. Один фрагмент может содержать несколько предварительных команд, в частном случае предварительные команды могут отсутствовать;
- **таймер** - задает выдержку времени или контрольное время выполнения условий
- **условия** - фрагмент существует только в условном шаге, в нем анализируются; условия выполнения последующих фрагментов шага. Условия могут задаваться по совокупному состоянию различных дискретных сигналов, существующих в данном контроллере;
- **заключительные команды** - выполняются в конце шага. Условный шаг в зависимости от выполнения условий может иметь разный набор команд. В частном случае заключительные команды могут отсутствовать
- **переходы** - безусловный шаг имеет одну ветку перехода, условный шаг - две ветки, одна ветка соответствует выполненным условиям, другая – невыполненным.

### 6.3 Фрагменты шага

#### 6.3.1 Команда

Любая выполняемая в шаге команда (предварительная или заключительная) изменяет состояние одного бита, принадлежащего одному из 9 выходов алгоритма УЛП данной (и только данной) логической программы. Имеется также групповая команда, которая соответствующим образом одновременно меняет состояние всех 16 бит одного выхода.

При работе с текстом программы эти выходы скрыты и отображаются в специальной форме Полиса (рисунок 49).

Каждая команда может следующим образом изменять состояние выходного бита (или битов одного выхода):

- установить в 0;
- установить в 1;
- установить в 1 со сбросом.

В первых двух случаях состояние бита сохраняется до тех пор, пока оно не будет изменено другой командой. В последнем случае на выходе «1» устанавливается на время выполнения шага, после чего автоматически сбрасывается в «0».

Один и тот же бит может изменять свое состояние под действием разных команд, принадлежащих разным этапам и разным этапам данной программы.

Для того, чтобы команды логической программы воздействовали на исполнительные устройства, выходные биты программы конфигурируются с соответствующими алгоритмами (рисунок 51). Как обычно в Пилоне, один программный выход Полиса может быть связан с любым числом входов разных алгоритмов.

Инд. № подл.	Подп. и дата	Взаим. инв. №	Инв. №	Подп. и дата
--------------	--------------	---------------	--------	--------------

Изм	Лист	№ докум	Подп.	Дата	ПФДИ.421457.009 И3.2	Лист
						63

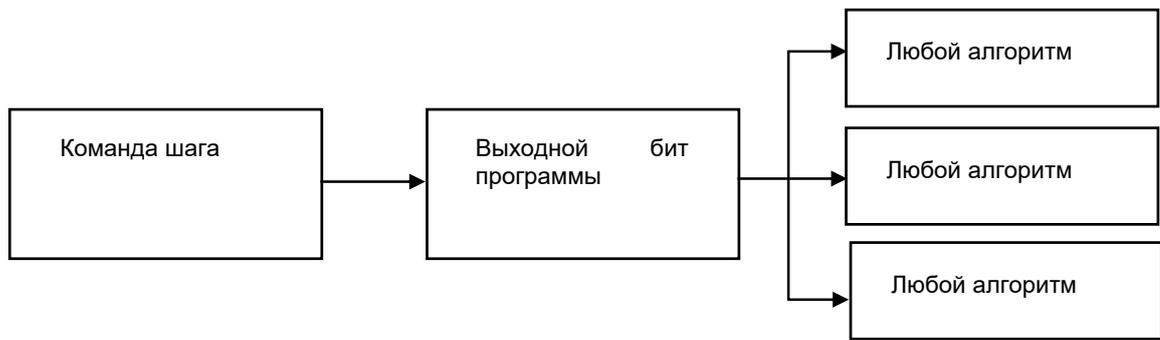


Рисунок 51 - Цепочка выполнения команды

## 6.4 Таймер

Каждый шаг содержит один таймер, в котором можно задать следующие варианты времени (как для выдержки времени, так и для контрольного времени ожидания выполнения условий):

- абсолютное время в минутах;
- абсолютное время в секундах;
- время, равное одному циклу контроллера;
- время, равное остатку времени от предыдущего шага;
- бесконечное время;
- нулевое время (т.е. не ждать, что равносильно отсутствию таймера).

Имеются следующие типы ожидания:

- ждать выполнения всех условий, но не дольше, чем заданное контрольное время;
- ждать выполнения хотя бы одного условия, но не дольше, чем заданное контрольное время;
- ждать заданное время, после чего проверить выполнение всех условий;
- ждать заданное время, после чего проверить выполнение хотя бы одного условия;
- безусловное ожидание.

Первые четыре типа ожидания используются в условном шаге, последний тип – это обычная выдержка времени в безусловном шаге.

### 6.4.1 Условия

Условие в шаге – это состояние какого-либо дискретного сигнала, формируемого в данном контроллере. Условие считается выполненным, если значение соответствующего дискретного сигнала равно 1. Чтобы связать выполнение условия с нулевым значением сигнала, в условии следует установить инверсию. В одном шаге можно определить несколько условий, объединенных либо по схеме И либо по схеме ИЛИ.

### 6.4.2 Переходы

Переходы задают дальнейшее (после выполнения текущего шага) «движение» программы. В каждом шаге можно задать следующие переходы:

- к заданному шагу в пределах текущего этапа;
- к заданному этапу в пределах текущей программы (всегда к первому шагу);
- в состояние Сброс программы;
- в состояние мягкого зависания;
- в состояние Конец программы;
- к следующему шагу текущего этапа с переходом в состояние **Стоп**;
- к началу текущего шага с переходом в состояние **Стоп**.

В условном шаге каждый вариант переходов можно задать индивидуально для ветки **То** и **Иначе** (о состояниях программы см. описание алгоритма УЛП).

Если переход явно не указан, то подразумевается следующее:

- если шаг не последний, то программа переходит к следующему шагу текущего этапа;
- если шаг последний, но этап не последний, то программа переходит к началу следующего этапа;
- если шаг и этап последние, то программа переходит в состояние **Конец программы**.

Инв. № подл.	Подп. и дата
Взаим. инв. №	Инв. №
Подп. и дата	Подп. и дата

## 6.5 Последовательность действий до начала программирования

До начала программирования новой логической шаговой программы необходимо выполнить в строгой последовательности следующие действия:

- 1 В Аркаде прописать очередной объект - логическая программа.
- 2 В Пилоне в одну из задач ввести алгоритм УЛП и стандартным образом привязать его к соответствующей марке.
- 3 В окне настроек Пилона (рисунок 13) установить указатель **Полис**. При необходимости там же выбрать тип и размер шрифта, которые будут использоваться в тексте программы Полиса (шрифтовые выделения типа **жирный**, **курсив** и т.д. не действуют, но можно использовать строчные и прописные буквы).
- 4 Двойным щелчком мыши по шапке алгоритма УЛП открыть окно Полиса с двумя закладками. В окне закладки **Текст программы** (рисунок 52) будет размещаться сама логическая программа. Исходно автоматически создается строка с именем программы, заданном в Аркаде, строка с первым этапом (без имени), строка с первым шагом в первом этапе (без имени) и один фрагмент шага **Таймер** с временем **Ждать 0 с**. В окне закладки **Программные Выходы** (рисунок 53) указаны 144 выхода (бита) программы для связи с исполнительными устройствами. Если в этом окне отметить галочку **Показывать только связанные или имеющие команду**, то незадействованные выходы показываться не будут (при этом легче ориентироваться в общем списке выходов).

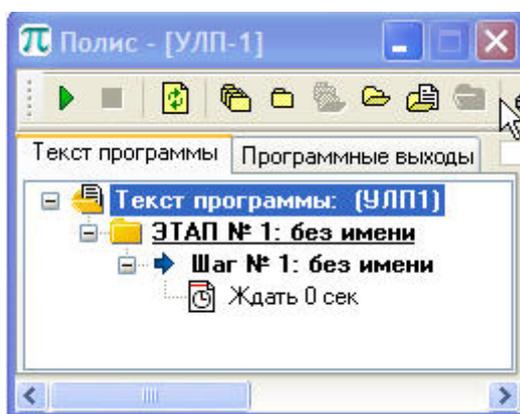


Рисунок 52 - Исходное состояние окна закладки Текст программы

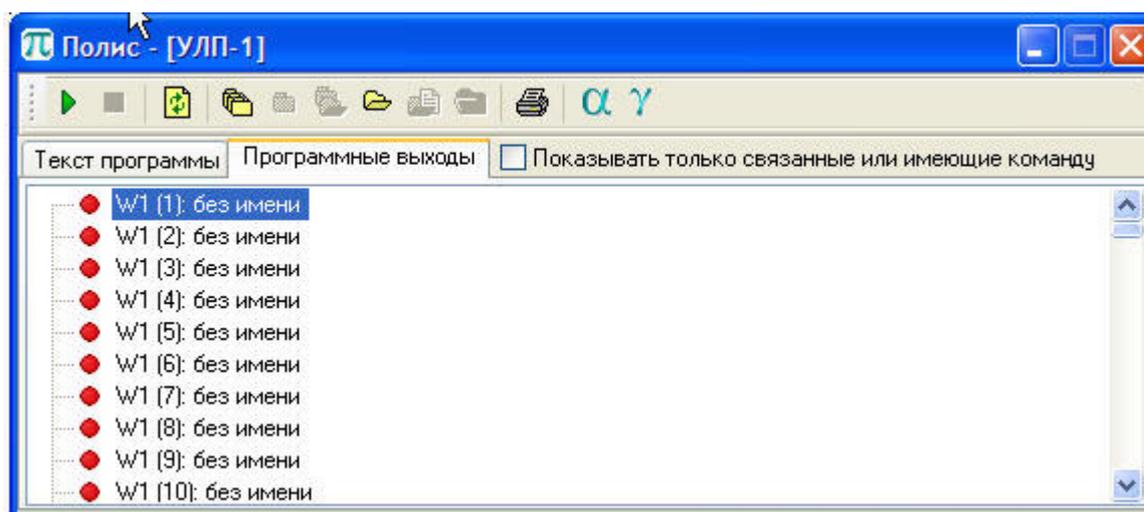


Рисунок 53 - Исходное состояние окна закладки Программные выходы

## 6.6 Техника программирования логических программ

При программировании возможны три варианта:

Инв. № подл.	Подп. и дата
Взаим. инв. №	Инв. №
Подп. и дата	Подп. и дата

Изм	Лист	№ докум	Подп.	Дата
-----	------	---------	-------	------

- 1 Вначале сконфигурировать программные выходы с приемниками, а затем подготовить текст программы, выбирая для команд уже подготовленные выходы.
- 2 Вначале подготовить текст программы, выбирая для команд любые из 144 программных выходов, после чего «активные» (находящиеся под управлением команд) выходы сконфигурировать с нужными приемниками.
- 3 Смешанный вариант, когда вначале конфигурируется группа программных выходов, затем подготавливается часть программы, в которой используются эти выходы, после чего процесс повторяется.

Во всех вариантах программным выходам и командам можно присвоить технологические имена.

### 6.6.1 Конфигурирование программных выходов

В качестве примера определим 2 программных выхода, которые будут формировать команды управления насосом №1

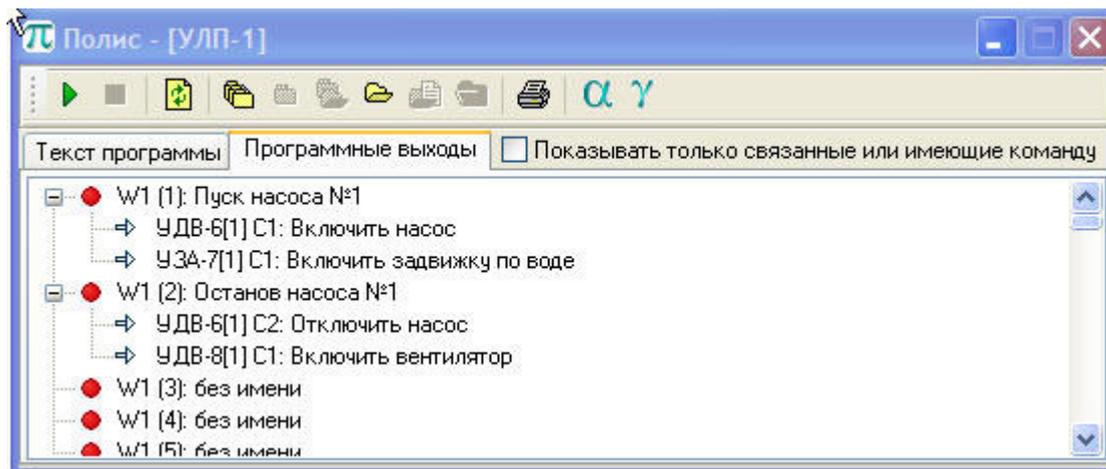


Рисунок 54 - Конфигурирование программных выходов (пример)

Вначале дадим наименования этим выходам, для чего выделим строку W1(1), щелкнем по ней левой клавишей мыши и с клавиатуры введем технологическое имя, - в нашем случае это **Пуск насоса №1** (рисунок 54). Затем определим, с какими алгоблоками - приемниками этот выход связан. Для этого из диалогового окна, которое открывается по правой клавише на выходе W1(1), выбираем **Добавить связь**, - курсор принимает форму руки, перемещаем курсор к входу нужного алгоритма, щелкаем по нему левой клавишей и на этом входе получаем ссылку на источник, а в источнике – указание на вид и номер алгоблока - приемника, номер канала (в скобках) и обозначение входа-приемника. Теперь этой ссылке в окне Полиса остается присвоить технологическое имя – в нашем случае это **Включить насос** (общее правило для всех технологических имен – вначале выделяем нужную строчку, затем щелкаем на ней левой клавишей мыши и с клавиатуры вводим требуемое имя).

Один программный выход может быть связан с любым числом алгоблоков - приемников. В нашем случае первый программный выход одновременно связан с алгоритмами управления насосом и задвижкой.

Для формирования команды отключения насоса аналогично поступаем со вторым программным выходом.

**Примечание** - Последовательность записанных ссылок, относящихся к одному программному выходу, значения не имеет, - все они реализуются одновременно.

Если по ссылке, связанной с выходом, щелкнуть правой клавишей мыши и выбрать **Перейти по ссылке**, то курсор перейдет к соответствующему входу алгоблока – приемника, а окно Полиса свернется. И наоборот, если ту же операцию выполнить у входа алгоблока - приемника, то курсор мыши перейдет к соответствующей ссылке окна Полиса.

Сигнал на входе алгоблока - приемника можно инвертировать по обычному правилу Полиса, при этом инверсию можно установить, щелкая правой клавишей мыши либо на строке ссылки в Полисе, либо на входе алгоблока - приемника. В любом случае признак инверсии будет отражен в двух местах – на входе алгоблока - приемника и перед строкой соответствующей ссылке (непосредственно после стрелки) в Полисе.

Инва. №	Подп. и дата
Взаим. инв. №	
Подп. и дата	
Инва. № подл.	

Любая ссылка, связывающая программный выход с алгоблоком - приемником, может быть удалена. Для этого по ссылке следует щелкнуть правой клавишей мыши и в диалоговом окне выбрать **Удалить**. Для удаления можно использовать как ссылку, прописанную в Полисе, так и ссылку на входе алгоблока - приемника, - в любом случае связь разорвется и обе ссылки будут удалены.

На рисунке 54 видно, что каждая строка, определяющая назначение программных выходов и ссылок, имеет два поля: системное и технологическое. Системное поле формируется автоматически, - оно либо предопределено (например, номер выхода), либо появляется после выполненной той или иной операции (например, после добавления связи с алгоблоком - приемником). Технологическое поле по умолчанию отсутствует и появляется лишь после того, как оно будет введено с клавиатуры. Когда поле находится в режиме редактирования, системная часть поля скрывается, однако после выхода из режима редактирования системное поле вновь появляется и размещается перед технологическим полем.

Для того, чтобы одним взглядом охватить все задействованные программные выходы и выделить их из общего перечня, содержащего 144 выхода, можно отметить галочкой поле **Показывать только связанные или имеющие команду**. Связанные – это сконфигурированные программные выходы, имеющие команду – это программные выходы, находящиеся под управлением команд логической программы.

Слева от программных выходов имеется кружок красного или синего цвета. Красный цвет означает, что в тексте программы нет команды, связанной с данным выходом, синий кружок свидетельствует о том, что данный выход находится под управлением какой-либо команды. По правой клавише мыши на выбранном выходе и выборе **Найти первую команду для данного выхода** можно перейти к тексту программы, при этом соответствующая команда будет подсвечена. С одним программным выходом может быть связано несколько команд, находящихся в разных фрагментах шага, в разных шагах или разных этапах. Просмотреть все эти выходы можно, если по правой клавише мыши выбрать **Следующая ссылка** или **Предыдущая ссылка**.

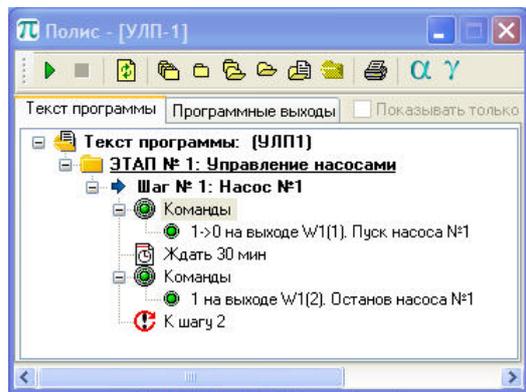
### 6.6.2 Запись текста программы без условий

В качестве примера рассмотрим простейший случай записи следующей программы:

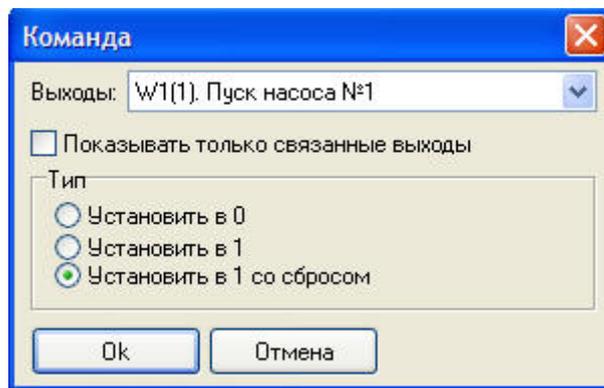
- 1 Включить насос №1.
- 2 Выдержать насос во включенном состоянии 30 мин.
- 3 Выключить насос.

Открываем окно закладки **Текст программы**. Исходно оно имеет вид как на **Ошибка! Источник ссылки не найден.**

С клавиатуры в печатаем имя первому этапу и первому шагу: **Управление насосами, Насос №1** (рисунок 55 а). На записи шага щелкаем правой клавишей мыши и из диалогового окна выбираем **Добавить команду**. В открывшемся дополнительном окне (рисунок 55 б) выбираем один из программных выходов, на который должна действовать команда, и назначаем тип команды.



а)



б)

**Рисунок 55 - Пример записи текста программы без условий**

Если выход, на который должна действовать команда, уже описан (т.е. связан с алгоблоком - источником), то предпочтительно отметить галочку в поле **Показывать только связанные выходы**. Тогда в поле **Выходы** будут перечислены не все 144, а только «авторизованные»

Инв. №	Подп. и дата
Взаим. инв. №	
Подп. и дата	
Инв. № подл.	

программных выходы. Если нужный программный выход еще не связан, то галочку следует снять, тогда из выпадающего списка можно выбрать любой из 144 программных выходов.

Для выбора типа команды в окне (рисунок 55 б) имеется три указателя:

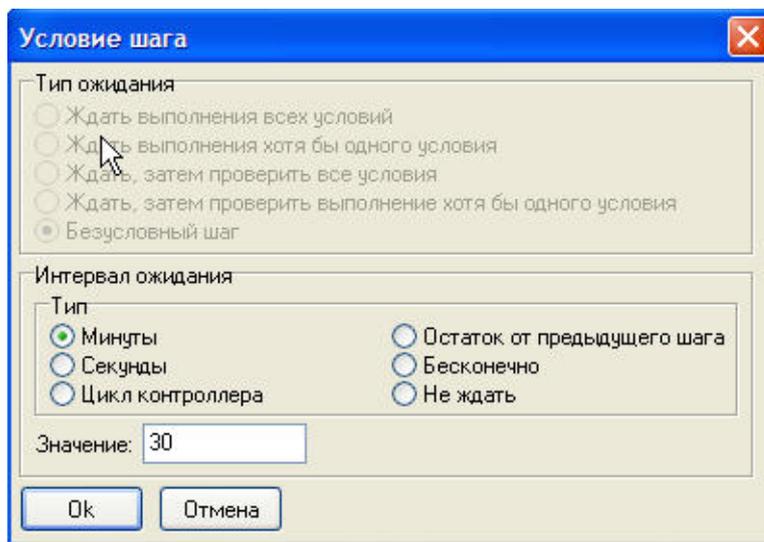
- установить в «0»;
- установить в «1»;
- установить в 1 со сбросом «1->0».

Назначение первых двух типов команды очевидно, при выборе третьего типа на программном выходе на **время выполнения шага** будет установлена 1, после чего состояние выхода автоматически сбросится в 0.

Имеется возможность с помощью одной команды установить значение не только на одном из 144 дискретных выходов, но и сразу на 16-и битах одного из упакованных выходов W1-W9.

После выбора типа команды нажать экранную клавишу **ОК**, в результате диалоговое окно закроется а в тексте программы под заголовком шага появится строка **Команда** с записью, характеризующей эту команду. В нашем случае на выходе W1, бит 1 будет установлена «1->0», причем этому программному выходу автоматически будет присвоено имя, ранее заданное при определении программного выхода.

Теперь запрограммируем таймер для нашего примера без условий. Для этого щелкнем правой клавишей мыши по строке **Ждать** и из открывшегося диалогового окна выбираем строку **Свойства**, в результате открывается новое диалоговое окно **Условие шага** (рисунок 56), в котором можно задать параметры таймера.



**Рисунок 56 - Окно заданий параметров таймера**

На рисунке 56 видно, что в данном окне имеется возможность задать:

- время в минутах;
- время в секундах;
- значение, равное одному циклу контроллера;
- значение, равное остатку времени от предыдущего цикла;
- бесконечное время;
- нулевое время (не ждать);

В первых двух случаях дополнительно задается значение времени в поле **Значение**, в остальных это значение предопределено выбранным вариантом.

После того, как будет нажата экранная клавиша **ОК**, диалоговое окно закроется, а в программе в строке таймера появится установленное значение.

Далее для нашего примера следует сформировать заключительную команду. Для этого щелкаем правой клавишей мыши по строке таймера и определяем заключительную команду так же, как и предварительную команду.

Инв. № подл.	Подп. и дата
Инв. №	Подп. и дата
Взаим. инв. №	Подп. и дата
Инв. № подл.	Подп. и дата

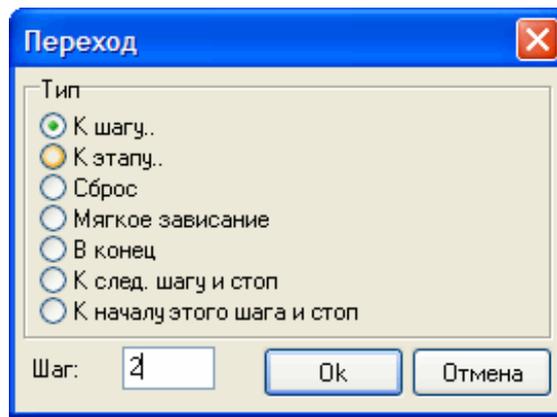


Рисунок 57 - Окно выбора переходов без условий

Осталось назначить переход, определяющий дальнейший путь прохождения программы. Для этого щелкаем правой клавишей мыши по строке таймера и выбираем **Добавить переход**, - открывается диалоговое окно (рисунок 57), в котором необходимо указать тип перехода.

Для выбора типа перехода имеются следующие указатели:

- **К шагу.** Переход к указанному в нижнем поле шагу текущего этапа;
- **К этапу.** Переход к первому шагу указанного в нижнем поле этапа текущей программы;
- **Сброс.** Переход в начальное состояние со сбросом всех текущих значений программы;
- **В конец.** Переход в конец программы и останов;
- **К следующему шагу и стоп.** Останов программы на начале следующего шага;
- **К началу этого шага и стоп.** Останов программы с возвратом на начало текущего шага.

В программе переход к шагу или этапу может явно не задаваться. В этом случае программа перейдет к следующему шагу текущего этапа либо, если шаг последний, к началу следующего этапа текущей программы, либо, если этап последний, в состояние конца программы.

### 6.6.3 Запись текста программы с условиями

В качестве примера рассмотрим простой случай записи следующей программы:

- В шаге №1 включить насос №1;
- Через 10 сек. проверить условия о состоянии битов 9 и 10 на выходе S алгоритма УДВ-6. Условия соответствуют включенному двигателю;
- Если предыдущее условие выполняется, то перейти к шагу №2 программы, иначе – мягкое зависание;
- В шаге №2 включить насос №2;
- Через 10 сек. проверить условия о состоянии битов 9 и 10 на выходе S алгоритма УДВ-6. Условия соответствуют включенному двигателю;
- Если предыдущее условие выполняется, то перейти к концу программы, иначе – к шагу №3;
- В шаге №3 выключить насос №1.

Текст программы указанного выше примера приведен на рисунке 58.

Предварительные команды для условного шага задаются так же, как и для безусловного. Отличия заключаются в записях шагов с условиями. В качестве условий записываются predetermined состояния дискретных выходов алгоблоков.

Для программирования условного шага необходимо мышью выбрать требуемый выход алгоблока - источника условия (при этом курсор мыши приобретает вид руки). Затем щелкнуть по строке таймера и в появившемся диалоговом окне выбрать номер бита условия и, при необходимости, признак инверсии (при отсутствии инверсии условие считается выполненным, если соответствующий бит равен 1, при наличии инверсии – если бит равен 0).

Инв. № подл.	Подп. и дата
Взаим. инв. №	Инв. №
Подп. и дата	Подп. и дата

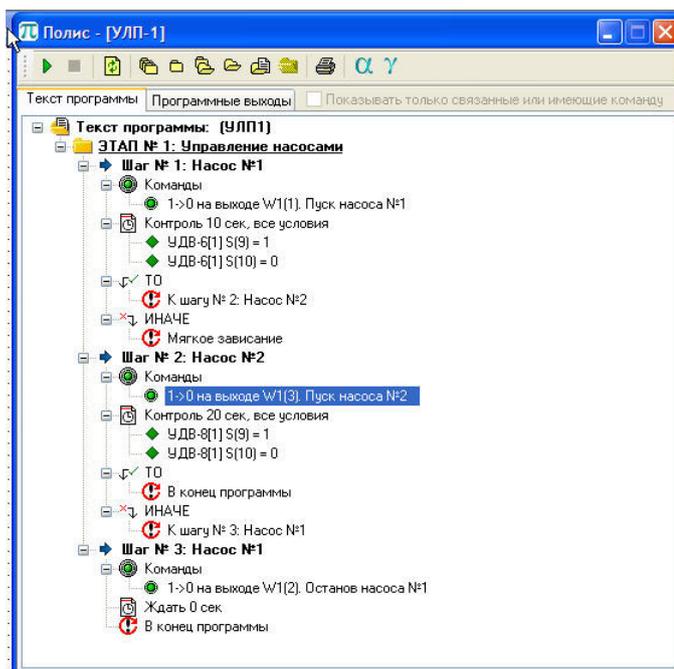


Рисунок 58 - Пример записи текста программы с условиями

Затем следует определить функции таймера. Для этого щелчком правой клавишей мыши по строке таймера вызываем дополнительное диалоговое окно (рисунок 56) и в нем устанавливаем параметры таймера: время (как в безусловном шаге) и назначение.

Назначения таймера (рисунок 59) предусмотрены следующие:

- **Ждать выполнения всех условий.** Это контрольное время, в течение которого ожидается выполнение всех заданных далее условий (схема И). Если до истечения этого времени условия будут выполнены, программа перейдет по ветке **То** (рисунок 60), в противном случае она перейдет по ветке **Иначе** (рисунок 61).
- **Ждать выполнения хотя бы одного условия.** То же, что и в предыдущем пункте, но для выполнения условий достаточно, чтобы было выполнено хотя бы одно из них (схема ИЛИ).
- **Ждать, затем проверить все условия.** То же, что **Ждать выполнения всех условий**, но таймер обрабатывается полностью и лишь затем проверяются условия.
- **Ждать, затем проверить хотя бы одно условие.** То же, что **Ждать выполнения хотя бы одного условия**, но таймер обрабатывается полностью и лишь затем проверяются условия.
- **Безусловный шаг.** Используется для замены условного шага на безусловный. При этом перечень всех условий и сама ветка **Иначе** удаляются.

В тексте программы в строке таймера появляется запись **Контроль** и далее следует значение времени и вид контроля условий в соответствии с установленными параметрами таймера.

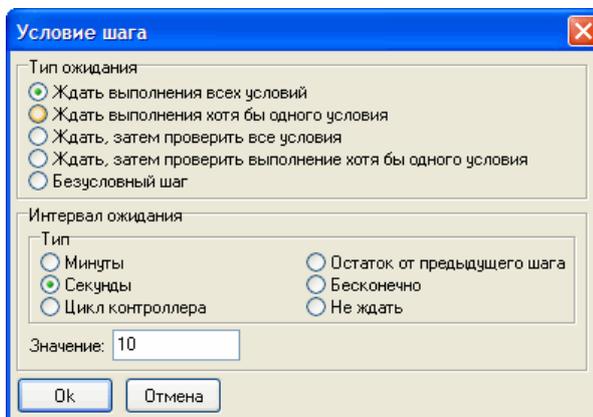
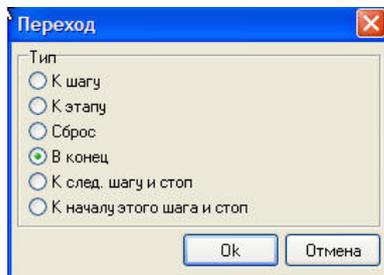


Рисунок 59 - Окно установки параметров таймера

Осталось определить поведение программы в случае, когда условия будут выполнены и не выполнены.

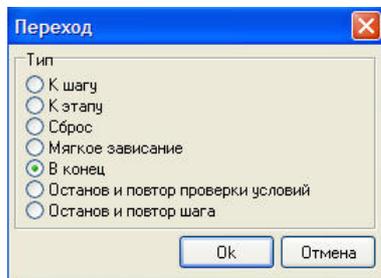
Инва. №	Подп. и дата
Взаим. инв. №	
Подп. и дата	
Инва. № подл.	

Для выполненных условий щелчком правой клавиши мыши по строке таймера (или строке с заголовком шага) открываем дополнительное окно и в открывшемся меню выбираем **Добавить переход по ТО**, в результате открывается то же диалоговое окно, что и для безусловного шага (рисунок 60). В окне можно выбрать требуемый переход. Перед тем, как будет выполнен указанный переход, можно выполнить любой набор команд. Для этого щелчком правой клавишей мыши по строке перехода **ТО** и в открывшемся диалоговом окне выбираем **Добавить команду**. Далее действуем по аналогии с процедурой, описанной в безусловном шаге.



**Рисунок 60 - Окно выбора перехода по условиям ветки То**

Для невыполненных условий таким же способом добавляем в программу ветку **Иначе** (рисунок 61) набор команд:



**Рисунок 61 - Окно выбора перехода по условиям ветки Иначе**

Отличие ветки **Иначе** в вариантах перехода в том что здесь имеются следующие дополнительные возможности (рисунок 61), связанные с поведением программы при невыполнении условий:

- **Мягкое зависание.** Программа остается в режиме пуска и проверка условий продолжается;
- **Останов и повтор проверки условий.** Программа останавливается и когда она снова будет пущена оператором, фрагмент программы с таймером и проверкой условий повторится;
- **Останов и повтор шага.** То же, что и в предыдущем пункте, но текущий шаг повторяется полностью (вместе с предварительными командами).

Во всех указанных случаях в статусе алгоритма УЛП формируются соответствующие признаки.

## 6.7 Общие ограничения логических программ

Ниже перечислены общие ограничения, влияющие на логические программы:

- каждая логическая программа резервирует в памяти контроллера место под один алгоблок для алгоритма УЛП и по одному алгоблоку на каждый этап. Поэтому при добавлении логической программы свободные ресурсы, контролируемые в Пилоне, уменьшаются на соответствующую величину;
- число этапов в одной логической программе – до 31. Если задача требует большего числа этапов, ее следует разбить на несколько программ и при завершении текущей программы автоматически (через алгоритм УЛП) запускать следующую программу;
- число шагов в одном этапе – до 31. Если задача требует большего числа шагов, ее следует разбить на несколько этапов;
- суммарное число операторов (строк программы) для всех шагов одного этапа – до 255;
- число дискретных программных выходов одной программы – до 144;
- число условий в одном шаге – ограничений нет. Действуют только перечисленные выше общие ограничения.

Инв. № подл.	Подп. и дата
Взаим. инв. №	Инв. №
Подп. и дата	Подп. и дата

## 6.8 Анализ и редактирование программы

При создании программы могут быть случаи, что отдельные ее строки окажутся некорректными. Типичный случай – запись перехода на несуществующий шаг или этап. Когда такая программа открывается в окне Полиса, вначале появляется окно **Анализ логической программы** (рисунок 62). Вызвать это окно можно и принудительно, щелкнув правой клавишей мыши по шапке алгоритма УЛП и выбрав из диалогового окна строку **Анализ логической программы**.

В нижней части окна (выводится список ошибок и предупреждений. Ошибки – это некорректные записи программы, при которых она не может правильно выполняться. Предупреждения информируют о том, что отдельные операторы программы в явном виде не заданы – программа в таких случаях будет выполняться согласно оговоренным правилам (например, если явно не задан переход, то программа перейдет к следующему шагу и т.п.).

Опера...	№...	Тип оператора	Комментарий	№...	Кол-во операторов	Кол-во шагов	Кол-во исп. команд
****		Этап № 1	Управление насосами	1	20	4	3
A001	1	Шаг № 1- усл. с контр.вре...	Насос №1				
2101	2	Команда	1->0 на выходе W1(1) Пуск насоса №1				
B010	3	Время	010 сек				
0000	4	Условие	УДВ-6(1) W(9) = 1				
0000	5	Условие	УДВ-6(1) W(10) = 0				
A002	6	Шаг № 2- усл. с контр.вре...	Насос №2				
2103	7	Команда	1->0 на выходе W1(3) Пуск насоса №2				
B020	8	Время	020 сек				
0000	9	Условие	УДВ-8(1) W(9) = 1				
0000	10	Условие	УДВ-8(1) W(10) = 0				
0000	11	Условие	УДВ-6(1) W(1) = 1				
CFFF	12	Переход по "То"	В конец программы				
D003	13	Переход по "Иначе"	К шагу №3				
A003	14	Шаг № 3- безусловный	Насос №1				
2102	15	Команда	1->0 на выходе W1(2) Останов насоса №1				
CFFF	16	Безусл. переход	В конец программы				
A004	17	Шаг № 4- усл. с контр.вре...					
0000	18	Условие					
CF01	19	Переход по "То"	К этапу №1				
DF05	20	Переход по "Иначе"	К этапу №5				

**Ошибки и предупреждения**

- Этап №1, Шаг №1 переход по ветке "То" не задан (будет выполняться переход к след. шагу)
- Этап №1, Шаг №1 переход по ветке "Иначе" не задан (будет выполняться переход в состояние мягкого зависания)
- Этап №1, Шаг №4 время не задано (будет приравнено остатку времени от предыдущего шага)
- Этап №1, Шаг №4 : переход к несуществующему этапу № 5 или к несуществующему шагу № 1 этапа № 5

Ошибок : 1  
Предупреждений : 3

Рисунок 62 - Окно анализа логической программы

В верхней части окна программа представлена в табличной форме: 16-ричные операторы, записанные в формате алгоритма ЭТА, номер оператора, его тип, комментарий, общие количества операторов, шагов, исполнительных команд для каждого этапа. Здесь можно исправить неправильный оператор (см. описание алгоритма ЭТА в **Библиотеке алгоритмов**).

Если закрыть окно Анализа, откроется окно Полиса, в котором можно ввести необходимую коррекцию программы. В окне Полиса можно редактировать программу не только для того, чтобы исправить ошибки, но и для изменения ее логики. Щелкнув правой клавишей мыши на соответствующем операторе, можно указать **Свойства** и затем выбрать новый оператор или новое значение параметра, либо указать **Удалить** – тогда соответствующий фрагмент будет удален.

Корректировке программы помогает возможность быстрого перехода из операторов условий, команд и переходов к соответствующим алгоритмам Пилона. Для этого из диалогового окна, открывающегося по правой клавише на соответствующем операторе, следует выбрать указатели **Найти выход** (для команд) или **Перейти по ссылке** (для условий и переходов).

Программа Полиса может находиться в полностью развернутом или свернутом (полностью или частично) виде. При открытии она находится в свернутом состоянии, - видны только заголовок программы и перечень ее этапов. Развернуть или свернуть отдельные элементы программы можно непосредственно в ее тексте, либо воспользоваться командами группового свертывания и развертывания. Для управления свертыванием/развертыванием применяется правая клавиша мыши или соответствующие значки шкалы инструментов.

Инд. №	Инд. №	Инд. №	Инд. №
Взаим. инв. №	Взаим. инв. №	Взаим. инв. №	Взаим. инв. №
Подп. и дата	Подп. и дата	Подп. и дата	Подп. и дата
Инд. № подл.	Инд. № подл.	Инд. № подл.	Инд. № подл.

## 6.9 Отладка логической программы

### 6.9.1 Режимы работы программы

Для отладки логическая программа может переводиться в различные режимы работы:

- рабочий;
- информационный;
- ручной;
- автоматический;
- штатный;
- поэтапный;
- пошаговый.

В рабочем режиме выполняются все команды текущей программы, т.е. формируются ее программные выходы.

В информационном режиме команды не выполняются и программные выходы не формируются.

В ручном режиме программой управляют ручные команды, формируемые оператором. При этом возможны все варианты выполнения программы.

В автоматическом режиме команды поступают на входы алгоритма УЛП от автоматических устройств. При этом возможно только штатное выполнение программы

В штатном режиме после пуска программа выполняется полностью до конца.

В поэтапном режиме программа останавливается после выполнения каждого очередного этапа.

В пошаговом режиме программа останавливается после выполнения каждого очередного шага.

Все указанные возможности позволяют отладить даже очень сложные программы, имеющие большое число этапов, шагов, условий, команд и переходов.

**Примечание** - первый запуск программы после изменения (редактирования или просто перезагрузки) осуществляется в информационном режиме. Поэтому для проверки работоспособности программы в рабочем режиме необходимо остановить выполнение программы, нажать правой клавишей мыши в окне Полиса и перевести программу в рабочий режим и вновь запустить программу.

### 6.9.2 Обзор программы

Логическая программа может работать в различных режимах, быть достаточно разветвленной, иметь большое число условных переходов и циклов. Для отладки программы в Полисе предусмотрен режим **Обзор**, используемый на этапе проектирования. Как и алгоблоки Пилона, текст программы на Полисе в этом режиме начинает жить «реальной жизнью», - запрашивать требуемые параметры из контроллера, формировать команды, поступающие в контроллер и т.д.

Как и в работе с Пилоном, в режиме **Обзор** Полиса программа загружается в контроллер и там выполняется. Непосредственно с экрана можно управлять режимом работы программы – ее пускать и останавливать, изменять режим и т.д. Для этого предусмотрены соответствующие значки на панели инструментов Полиса.

Вначале программу следует перевести в режим **Обзор** и выполнить процедуру, обычную для Пилона (компиляция, загрузка и т.д.). Затем слева от строки с именем первого этапа программ появится зеленый треугольник, свидетельствующий о том, что программа готова к пуску.

После того, как на шкале инструментов будет нажата клавиша **Пуск**, начнется выполнение всех операторов. Когда программа дойдет до первого таймера, начнется отсчет времени и значение таймера будет изменяться, показывая оставшееся время. По мере продвижения программы слева от выполняемых в данный момент операторов будут появляться новые зеленые треугольники, так что всегда можно видеть, какой оператор является текущим. Если в шаге задано несколько условий, то как только какое-то условие выполняется, слева от него появляется зелёная галочка, так что всегда можно не только увидеть, на каком этапе и шаге находится программа, но какие условия уже выполнены, а какие нет.

Наблюдать можно не только за ходом программы и состоянием условий, но и за состоянием программных выходов. Если в запущенной программе перейти на закладку **Программные выходы**, то можно увидеть, какие выходы находятся в состоянии 0, а какие в состоянии 1.

Инв. № подл.	Подп. и дата	Взаим. инв. №	Инв. №	Подп. и дата
--------------	--------------	---------------	--------	--------------

В любой момент с помощью соответствующих значков на шкале инструментов программу можно остановить, сбросить или пустить заново. Также можно перейти в пошаговый, поэтапный информационный или рабочий режим выполнения.

**Примечание** - Чтобы протестировать работу программы в Полисе в сочетании с Пилоном, следует последовательно перевести в **Обзор** как Полис, так и Пилон. Если нужно проверить программу, начиная с первого этапа, следует сбросить программу, а затем ее пустить.

## 7 Подготовка отчета

Чтобы подготовить отчет, следует открыть Пилон и выбрать в меню пункт **Файл/Оформление**, в результате откроется список с выбором типа форматки:

- титульный лист проекта;
- титульный лист контроллера;
- страница задачи.

Таким образом, можно создать вариант оформления одного титульного листа для всего проекта, индивидуального титульного листа для каждого контроллера в данном проекте и типовой страницы для всех задач в пределах данного проекта. Аналогичный список доступен также по значку на панели инструментов Пилона. Правила подготовки одинаковы для всех видов форматок.

### 7.1 Проектирование форматок

После открытия Пилона, до выбора контроллера и задачи, имеется возможность определить вид форматок для титульного листа проекта, всех контроллеров данного проекта и всех задач данного проекта.

Рассмотрим методику проектирования форматки на примере титульного листа проекта. Открываем **Файл/Оформление/Титульный лист** (или используем раскрывающееся меню соответствующего значка на шкале инструментов), - на экран выводится форматка, на которой обозначены ее границы (по отношению к краям листа), границы полей (по отношению к краям форматки) и элементы оформления форматки в виде прямоугольников.

Далее необходимо задать размер страницы. Для этого выбирается **Файл/Параметры страницы** или нажимается соответствующий значок на шкале инструментов. В открывшемся окне (рисунок 63) устанавливаются размеры и ориентация страницы, размеры отступов от края форматки до ее рабочего поля, в котором будут размещаться алгоблоки, толщина рамки и параметры принтера. Размер форматки (но не страницы) и отступы можно задать непосредственно мышью, перемещая правую и/или нижнюю границу соответствующего поля.

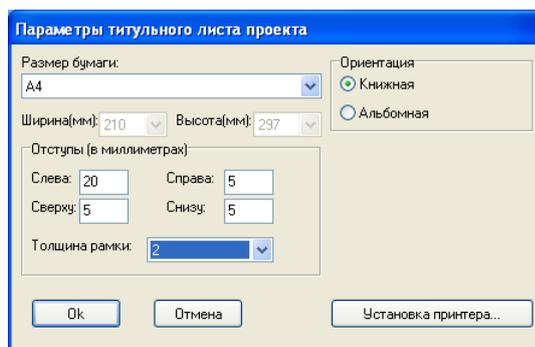
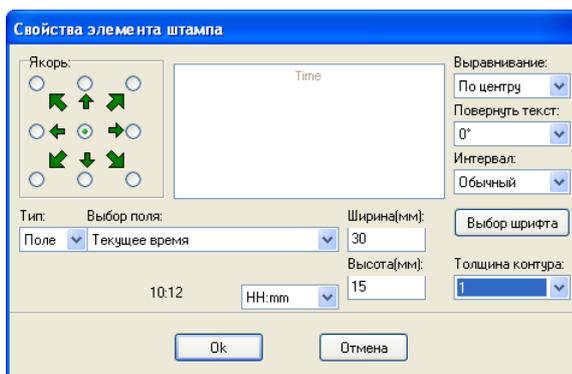


Рисунок 63 - Параметры титульного листа проекта.

Следующий шаг – определение элементов оформления страницы (штампов, надписей и т.п.).

Чтобы добавить какой-либо элемент необходимо нажать значок  на шкале инструментов и в открывшемся диалоговом окне (рисунок 64) задать параметры данного элемента оформления (размер прямоугольника, шрифт и т.п.). При этом поле **Якорь** определяет привязку элемента к границам форматки при изменении ее размеров (например, для того чтобы элемент всегда размещался по центру форматки при изменении ее ширины, в поле **Якоря** следует поставить точку по центру).

Инв. № подл.	Подп. и дата	Взаим. инв. №	Инв. №	Подп. и дата
--------------	--------------	---------------	--------	--------------



**Рисунок 64 - Свойства элемента штампа**

В диалоговом окне (рисунок 64) задается также тип выбранного элемента, - это может быть либо обычный текст, задаваемый пользователем, либо поле. Если выбирается поле, то появляется дополнительная возможность: из выпадающего списка можно выбрать тип поля, причем перечень типов зависит от вида форматки. По существу поле является неким шаблоном. В процессе проектирования на экране обозначается только тип поля для выбранного элемента в английской транскрипции (например, *Date* при выборе типа поля **Текущая дата**). При распечатке в указанное поле будет подставляться реальное значение (например, реальная дата).

На форматке можно разместить любое число элементов оформления и с помощью мыши разместить их произвольным образом (например, из нескольких элементов можно сконструировать штамп). Любой элемент можно удалить или переопределить его параметры, щелкнув по нему правой клавишей мыши и выбрав соответствующий пункт меню. Для элемента типа **Текст** в этом окне можно выбрать также **Редактировать**, после чего в соответствующий прямоугольник ввести требуемый текст.

Для любого прямоугольника можно задать толщину рамки. Если рамки не требуется, следует задать ее нулевую толщину.

## 7.2 Предварительный просмотр форматок и печать

Границы форматок обозначаются при просмотре на экране двойной пунктирной линией. Если размещение какой-либо задачи выходит за пределы одной форматки, она будет напечатана на нескольких форматках. При этом граница может разрезать некоторые алгоблоки и часть алгоблоков попадет на одну форматку, а часть – на другую. Чтобы этого не произошло, рекомендуется размещать алгоблоки таким образом, чтобы они укладывались в пределах одной форматки.

Если задача разделяется на несколько форматок и граница форматок разрезает конфигурационные линии, связывающие алгоблоки, то на примыкающих полях смежных форматок проставляются маркерные порядковые номера, по которым можно проследить связи между алгоблоками, размещенными на разных форматках.

Чтобы посмотреть, как будут выглядеть форматки с выбранной задачей после печати, необходимо выбрать **Файл/Предварительный просмотр** или нажать значок  на шкале инструментов, - откроется обзор форматок данной задачи перед печатью. В обзоре можно выбрать масштаб просмотра.

Чтобы вывести задачу на печать, необходимо в окне задачи (но не просмотра) выбрать **Файл/Печать**.

Индивидуально для каждой открытой задачи можно переопределить отдельные элементы ее оформления по отношению к тому, которое было задано для проекта в целом, а именно размер форматки и ее ориентацию. Если такого переопределения не делается, задача оформляется стандартным для данного проекта образом.

Инва. № подл.	Подп. и дата
Взаим. инв. №	Инва. №
Подп. и дата	Подп. и дата

