

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 расчётов и моделирования Мезон
Руководство пользователя
ПФДИ.421457.009 ИЗ.6

Москва
2018

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

Содержание

1	Введение	8
1.1	Структура документа	8
1.2	Терминология	8
1.3	Типы данных	9
1.4	Структура проектов Мезона	10
2	Создание проекта. Шаг за шагом	13
2.1	Шаг 1. Постановка задачи	13
2.2	Шаг 2. Декомпозиция	13
2.3	Шаг 3. Запуск Мезон-редактора, создание хранилища исходных кодов	14
2.4	Шаг 4. Создание проекта	14
2.5	Шаг 5. Добавление алглоблов, создание составного алгоритма	14
2.6	Шаг 6. Создание и настройка мнемосхемы	17
2.7	Шаг 7. Компилирование и запуск проекта	19
2.8	Шаг 7. Загрузка техпрограммы в Мезон-станцию	20
2.9	Шаг 8. Настройка и запуск Мезон-клиента	20
2.10	Шаг 9. Трассировка и отладка	21
3	Справочник	22
3.1	Окно запуска Мезон-редактора.....	22
3.2	Главное окно Мезон-редактора	22
3.2.1	Главное меню.....	23
3.2.2	Панель инструментов.....	27
3.2.3	Рабочая область	28
3.2.4	Строка состояния.....	36
3.3	Окно трассировки.....	37
3.4	Диалоги	38
3.4.1	Диалог Проект	38
3.4.2	Диалог Инспектор алглоблов	39
3.4.3	Диалог Копирование алглоблов.....	41
3.4.4	Диалог Управление модулями.....	41
3.4.1	Диалог Создание модуля проекта.....	42
3.4.2	Диалог Настройка хранилища	43
3.4.3	Диалог Выбор алгоритма	44
3.4.4	Диалог Выделение алглоблов, Выделение комментариев	44
3.4.5	Диалог Перейти к алглоблу, Перейти к комментарию	45
3.4.6	Диалог «Точки останова»	45

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

ПФДИ.421457.009 ИЗ.6				
Изм	Лист	№ докум	Подп.	Дата
Разраб.	Туркин			
Пров.	Зарипов			
Н.контр	Бочаров			
Утверд.	Петров			
Комплекс программно-технический Квинт-6. Система расчетов и моделирования Мезон. Руководство пользователя.			Лит	Лист
				2
				166
ООО «ТСА-Сервис»				

3.4.7	Диалог Изменить порядок алгоблоков.....	46
3.4.8	Диалог Добавить алгоблок, диалог Добавить задачу в техпрограмму.....	47
3.4.9	Диалог Управление окнами.....	48
3.4.10	Диалог Обзор плагинов.....	48
3.4.11	Диалог Параметры.....	48
3.5	Приёмы редактирования.....	52
3.5.1	Комментарии.....	52
3.5.2	Работа с группами объектов.....	52
3.6	Приёмы отладки.....	52
3.6.1	Компиляция, загрузка и запуск техпрограммы.....	53
3.6.2	Отладка по шагам.....	53
3.6.3	Отладка по точкам останова.....	53

4 Часто возникающие вопросы.....54

4.1	Вопрос 1. Есть готовая dll, в которой реализована некая функция. Как максимально быстро начать использовать её в Мезоне?.....	54
4.2	Вопрос 2. Как происходит преобразование сигналов различных типов?.....	54
4.2.1	Настраиваемое скрытое преобразование [1].....	55
4.2.2	Настраиваемое скрытое преобразование [2].....	55
4.2.3	Настраиваемое скрытое преобразование [3].....	56
4.3	Вопрос 3. Каков порядок выполнения алгоблоков и как на него влиять?.....	56
4.4	Вопрос 4. Что означают предупреждения компилятора о побочных эффектах связей?.....	57
4.4.1	Побочные эффекты обратных связей.....	57
4.4.2	Связь может привести к побочному эффекту.....	58
4.4.3	Побочные эффекты связей, не отслеживаемые компилятором.....	58
4.5	Вопрос 5. Что нужно сделать, чтобы запустить Мезон-редактор?.....	59
4.6	Вопрос 6. Почему отсутствует нужный Мезон-сервер в диалоге создания и выбора техпрограмм Мезона?.....	59
4.7	Вопрос 7. Как управлять техпрограммами, запущенными на Мезон-станции?.....	59
4.8	Вопрос 8. Как перенести проект из одной базы Квинта в другую?.....	59
4.9	Вопрос 9. Как перенести модуль (задачу, мнемосхему, составной алгоритм) из одного проекта в другой?.....	60
4.10	Вопрос 10. Как на локальной машине открыть архив?.....	60
4.11	Вопрос 11. Как из техпрограммы Мезона записать значения в архив?.....	60
4.12	Вопрос 12. Почему отображается надпись «ошибка» в рабочей области проекта Мезона?.....	60
4.13	Вопрос 13. Как настроить автоматический запуск технологической программы?.....	61
4.14	Вопрос 14. Как контекстно открыть модуль техпрограммы?.....	62
4.15	Вопрос 15. Почему в режиме отладки некоторые модули становятся серыми, а внизу появляется надпись «модуль открыт не контекстно...»?.....	62

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

5	Библиотека алгоритмов Мезона	63
6	Функциональное описание алгоритмов Мезона.....	68
6.1	Группа 'ModBus'.....	68
6.1.1	ModBus-ввод	68
6.1.2	ModBus-вывод.....	68
6.2	Группа 'Биты'	69
6.2.1	БитНе	69
6.2.2	БитИ	69
6.2.3	БитИли	70
6.2.4	Описание алгоритма.....	70
6.2.5	БитИскл.....	70
6.2.6	СдвигВлево	71
6.2.7	СдвигВправо.....	71
6.2.8	БитШифр.....	71
6.2.9	БитДешифр	72
6.3	Группа 'Общие'	73
6.3.1	Память	73
6.3.2	Задержка.....	73
6.3.3	Сравнить.....	73
6.3.4	Выбор.....	74
6.3.5	МножВыбор	74
6.3.6	Размнож.....	75
6.3.7	Инициализация	75
6.3.8	Формула.....	76
6.3.9	Преобразователи.....	79
6.3.10	ТВЫ.....	83
6.3.11	ТВВ.....	84
6.3.12	Переменная.....	84
6.3.13	Переменная 2.....	86
6.4	Группа 'Логика'.....	88
6.4.1	Не	88
6.4.2	И	88
6.4.3	Или	88
6.4.4	ИсклИЛИ	89
6.5	Группа 'Математика'.....	90
6.5.1	Слож.....	90
6.5.2	Выч	90
6.5.3	Умнож.....	90
6.5.4	Делен	91
6.5.5	ДелОст	91
6.5.6	Округл	91
6.5.7	Степень.....	92

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

6.5.8	Корень.....	92
6.5.9	$X = -X$	92
6.5.10	$1/X$	92
6.5.11	10^X	93
6.5.12	E^X	93
6.5.13	Lg	93
6.5.14	Ln	94
6.5.15	Sin	94
6.5.16	Cos	94
6.5.17	Tg	94
6.5.18	$ArcSin$	95
6.5.19	$ArcCos$	95
6.5.20	$ArcTg$	95
6.5.21	Факториал.....	95
6.5.22	Модуль.....	96
6.6	Группа 'Время'.....	97
6.6.1	Сейчас.....	97
6.6.2	УпакВр.....	97
6.6.3	РаспакВр.....	98
6.6.4	УпакДлит.....	98
6.6.5	РаспакДлит.....	99
6.6.6	СмещВр.....	99
6.6.7	ДлитВр.....	100
6.7	Группа 'ТАУ'.....	101
6.7.1	Задер.....	101
6.7.2	Интегр.....	102
6.7.3	Диффер.....	102
6.7.4	ИнЗвено1.....	102
6.7.5	ИнЗвено2.....	103
6.8	Группа 'Моделирование'.....	104
6.8.1	Виконт.....	104
6.8.2	АЦП.....	106
6.8.3	ЦАП.....	107
6.8.4	ДЦП.....	107
6.8.5	ЦДП.....	108
6.8.6	ДВВ.....	108
6.8.7	ЦИП.....	109
6.8.8	АВВ.....	109
6.8.9	МЗТ.....	110
6.8.10	МЧТ.....	111
6.8.11	Задвижка.....	112
6.8.12	Фрейм.....	113
6.8.13	БОК.....	115

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

6.8.14	РегКлапан	116
6.8.15	АД-I	117
6.9	Группа 'Отладка'	119
6.9.1	ОтлПечать	119
6.9.2	ВремяЦикла	119
6.10	Группа 'Excel'	121
6.10.1	КнигаXL	121
6.10.2	ЗаписьXL	122
6.10.3	ЧтениеXL	122
6.10.4	ЛистXL	123
6.10.5	ДиапазонXL	123
6.10.6	ЯчейкаXL	124
6.11	Группа 'Квинт'	125
6.11.1	ЧитатьПараметр	125
6.11.2	ПисатьПараметр	126
6.11.3	История	128
6.11.4	ЧислоВРем	131
6.11.5	РемВЧисло	131
6.11.6	ИВЫ	132
6.11.7	ИВВ	133
6.11.8	ЧитатьВнешнийПараметр	134
6.11.9	История 2	136
6.11.10	ИсторияВнешнегоПараметра	139
6.12	Группа 'Строки'	144
6.12.1	РегСтр	144
6.12.2	ДлСтр	144
6.12.3	Сцепить	144
6.12.4	ПодСтр	145
6.12.5	ПовторСтр	145
6.12.6	ПоискСтр	145
6.12.7	ПреобрСтр	146
6.13	Группа 'События'	147
6.13.1	СчетСоб	147
6.13.2	СобИли	147
6.13.3	СобИ	147
6.13.4	СобВил	148
6.13.5	ЕстьСоб	148
6.14	Группа 'Управление'	149
6.14.1	RS-триггер	149
6.14.2	T-триггер	149
6.14.3	Таймер	149
6.14.4	Цикл	150
6.14.5	Шаг	151

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

6.14.6	Ошибка.....	151
6.14.7	Останов.....	152
6.14.8	Сначала	152
6.15	Группа 'Файлы'	153
6.15.1	Выбор файла.....	153
6.15.2	Запись	153
6.15.3	Чтение.....	154
6.15.4	Процесс.....	155
6.16	Группа 'Таблицы'	156
6.16.1	Писать в таблицу	156
6.16.2	Читать таблицу.....	157
6.17	Группа 'Мнемосимволы'	160
6.17.1	Кнопка	160
6.17.2	Ползунок	160
6.17.3	Редактор	161
6.17.4	Редактор значений	161
6.17.5	Слайд	161
6.17.6	Индикатор	162
6.17.7	Управление задачей.....	162
6.17.8	График	163
6.18	Группа 'Случайные'	164
6.18.1	СлучЧисло	164
6.18.2	СлучЦелое.....	164
6.18.3	СлучЛог.....	164
6.18.4	СлучСоб.....	165
Лист регистрации изменений.....		166

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

1 Введение

Мезон-редактор – это графический редактор, входящий в состав **программно-технического комплекса Квинт-6** (далее в тексте – Квинт). Он предназначен для программирования и отладки широкого круга вычислительных задач.

В Мезон-редакторе в качестве основы языка программирования используется одна из реализаций языка **FBD (Function Block Diagram** — графический язык программирования стандарта **IEC61131-3**). Программирование на нём представляет собой размещение на поле набора логических блоков обработки сигнала и установления связей между ними.

При помощи Мезон-редактора можно создавать вычислительные программы расчёта ТЭП, построения отчётов на основе архивных или «живых» данных, моделирования технологического процесса и т.п.

Программы, составленные в Мезон-редакторе, могут выполняться как на обычном персональном компьютере, так и на **Мезон-контроллере МК-80**, входящем в состав Квинт-6. МК-80 это одноплатный контроллер, работающий под управлением ОС Windows CE. При его применении надёжность выполнения программ значительно возрастает, что позволяет использовать его для вспомогательных вычислений непосредственно в контуре управления технологическим процессом.

1.1 Структура документа

Практика показывает, что научиться чему-либо по справочному пособию чрезвычайно трудно – для этих целей больше подходит пошаговое описание демонстрационного примера.

Однако тем, кто уже знает, как работать с Мезон-редактором, но тем не менее нуждается в справке по тем или иным редко используемым возможностям, использовать такое описание в качестве справочника весьма неудобно.

С другой стороны ни пошаговые примеры, ни справочник не дают прямых ответов на специфические, но часто возникающие в ходе работы вопросы.

В данном документе произведена попытка учесть изложенные соображения. В результате структура документа имеет следующий вид:

- 1 **Создание проекта. Шаг за шагом.** В этой главе описывается создание проекта с нуля, начиная от постановки задачи, ее декомпозиции и заканчивая завершающими действиями, связанными с ее отладкой и выполнением.
- 2 **Справочник.** Глава посвящена описанию элементов управления в Мезон-редакторе и дополнительных возможностей, которые не были рассмотрены в первой главе или которые упоминались в ней вскользь.
- 3 **Часто задаваемые вопросы.** Здесь описываются наиболее частые вопросы, возникающие при работе с Мезон-редактором. Вопросы разбиты по нескольким темам для облегчения поиска. На каждый вопрос даётся максимально подробный ответ, часто сопровождаемый работающим примером.

1.2 Терминология

При описании **Мезон-редактора** используется следующая терминология:

- **Алглоблок** - обобщенное название для простых, составных и фиктивных алглоблоков, а также для других видов экземпляров, которые возможно будут появляться в дальнейшем;
- **Алгоритм** – обобщенное название для простых, составных и фиктивных алгоритмов;
- **Библиотека алгоритмов** - совокупность алгоритмов, которые доступны пользователю для их использования в техпрограмме;
- **Вход** - свойство алгоритма для записи;
- **Выход** - свойство алгоритма для чтения;
- **Задача** - составной алглоблок первого уровня вложенности;
- **Конфигурирование** - тоже, что и связывание, а также установка значения на входе алглоблока;
- **Мезон** – расчетно-моделирующая подсистема в составе АСУ ТП ПТК Квинт-6. Обобщенное название для *Мезон-редактора*, *Мезон-сервера*;
- **Мезон-редактор** – редактор вычислительных и модельных технологических программ;

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

Изм	Лист	№ докум	Подп.	Дата	ПФДИ.421457.009 И3.6	Лист 8
-----	------	---------	-------	------	----------------------	-----------

- **Мезон-сервер** – обобщенное название для *Мезон-станции* и *Мезон-контроллера*. Средство выполнения вычислительных и модельных технологических программ, подготовленных в Мезон-редакторе;
- **Мезон-станция** – сервер выполнения скомпилированных технологических программ Мезона, обычно запускаемый из-под монитора приложений на персональном компьютере под управлением ОС Windows настольной версии;
- **Мезон-контроллер** – сервер выполнения скомпилированных техпрограмм Мезона, работающий на законченном аппаратном устройстве, под управлением Windows CE.
- **Мнемоблок** - простой алгоблок, осуществляющий графический интерфейс с пользователем;
- **Мнемосхема** - задача, объединяющая мнемоблоки. В отличие от простой задачи видна на стадии выполнения проекта (SHOW TIME);
- **Модуль** – обобщенное название для проекта, задачи, составного и фиктивного алгоблоков, открытых в Мезон-редакторе для редактирования;
- **Объект** - участок кода технологической программы, имеющий автономное смысловое значение. Как правило, объект реализуется алгоблоком;
- **Проект** - формат хранения технологической программы в структурированном виде. Содержит в себе сведения о связях между составными алгоблоками первого уровня вложенности, а также дополнительную (проектную) информацию, формально принадлежащую проекту в целом или одному или нескольким алгоблокам в проекте;
- **Простой алгоритм** - функция, реализующая выполнение последовательности команд, призванная решать поставленную задачу (например, алгоритм управления задвижкой). В общем случае функция принимает в качестве входных данных значения входов алгоритма, а результаты их обработки оставляет на выходах алгоритма;
- **Простой алгоблок** - экземпляр простого алгоритма, используемый в технологической программе;
- **Составной алгоритм** - совокупность простых и составных алгоритмов, имеющая интерфейс ввода-вывода;
- **Составной алгоблок** - экземпляр составного алгоритма, используемый в технологической программе;
- **Связь** - обозначение передачи информации с выхода на вход алгоблока;
- **Связывание** - процесс проведения связи;
- **Технологическая программа** - описание взаимодействия алгоблоков, а также результат компиляции *модулей* при помощи компилятора;
- **Уровень вложенности** - условное число, показывающее как далеко от уровня проекта (нулевой уровень) расположен тот или иной алгоблок;

1.3 Типы данных

Любой алгоритм имеет входы и (или) выходы. Значения констант на этих входах, а также информация, передаваемая от одного алгоблока к другому, называется сигналом. Однозначная интерпретация сигнала зависит от его типа. Так сигналы, типа векторный, целый, контейнер данных занимают в памяти по 4 байта, но интерпретируются в системе совершенно по-разному.

Основные типы входов и выходов в системе, соответствующие типам сигналов:

- 1 **Логический** - значения 0 или 1.
- 2 **Целый** - целое знаковое число в диапазоне минус 2147483647.....плюс 2147483648.
- 3 **Векторный** - целое беззнаковое число в диапазоне 04294967295, с которым можно работать как с массивом бит.
- 4 **Действительный** - знаковое число с плавающей запятой в диапазоне $\pm(2.2E-308...1,8E308)$.
- 5 **Модельное время** - относительная или абсолютная (относительно времени задачи) длительность в мсек. Сигнал предназначен в основном для целей моделирования и используется для задания всевозможных длительностей моделей механизмов.
- 6 **Строковый** - строковая константа, содержащая любое число символов.
- 7 **Дата и время** - число, описывающее дату и время, начиная от 1 января 1900 года 00:00:00.000.
- 8 **Длительность** - число, описывающее интервал времени в миллисекундах в диапазоне $\pm(9999 \text{ сут. } 23 \text{ ч. } 59 \text{ м. } 59 \text{ с. } 999 \text{ мсек.})$.
- 9 **Контейнер данных** - произвольный четырехбайтовый указатель. Константа данного типа будет представлена в виде массива байт, произвольной длины (контейнер).

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

Изм	Лист	№ докум	Подп.	Дата	ПФДИ.421457.009 И3.6	Лист
						9

10 **Событийный** - служит только для передачи событий и не может быть константой.

Примечания

1 Тип сигнала **контейнер данных** предоставляет ссылку в области памяти, за которой закрепляется определённый набор байт. Использование данного типа сигналов целесообразно для осуществления передачи произвольного массива данных между алгоблоками по ссылке. Можно также использовать непосредственные данные на входе с типом **контейнер данных**. Для этого следует записать ссылку на файл, содержащий непосредственные данные. Ссылка должна начинаться с ключевого слова `file://<имя файла>`. Если файл расположен в той же директории, что и задача в которой находится алгоблок, то имя файла может не содержать пути, в противном случае имя файла оснащается полным или относительным путём.

2 Некоторые типы сигналов могут быть преобразованы в другие типы (например, логический в целый), однако обратное преобразование иногда невозможно (например, действительное число можно преобразовать в строку, но не наоборот) или может приводить к побочным эффектам. Более подробно об этом сказано в разделе 4.

И ещё два дополнительных типа, не соответствующих типам сигналов:

- **Скрытый** - скрытый вход алгоблока, служит для хранения параметров отображения алгоблока в редакторе, а также для некоторых служебных целей. Hide описывается символьным массивом, произвольной длины;

Примечание - Использовать этот тип не рекомендуется. Он оставлен в системе для совместимости со старыми версиями алгоблоков.

- **Виртуальный** - неопределённый тип, соответствует любому произвольному типу. Удобен для реализации фиктивных алгоритмов. Входы и выходы фиктивных алгоритмов могут иметь тип виртуальный, что позволяет связывать их с любым типом сигналов в системе. При подстановке реального алгоблока вместо шаблона, все виртуальные типы будут заменены на типы выводов замещающего алгоблока (если такая замена окажется возможной).

1.4 Структура проектов Мезона

В Квинте существует понятие проекта. Это понятие описывает совокупность всех ресурсов и метаданных, используемых при проектировании АСУ ТП. Типичным случаем проекта является проект АСУ ТП энергоблока, заключающий все технологические программы, мнемосхемы, марки, объекты и пр. Физически проект Квинта хранится в специализированной Базе данных проекта на диске серверного компьютера.

В Мезоне также введено понятие проекта, но по смыслу оно несколько отличается от выше описанного понятия. Проект Мезона является подчинённым по отношению к проекту Квинта (проект Квинта может содержать несколько проектов Мезона). Он описывает одну технологическую задачу (например, подсчёт ТЭП энергоблока) и заключает в себе информацию о связи между исходными кодами. Далее под словом проект мы будем понимать проекты Мезона, за исключением особо оговорённых случаев.

Иерархия проектов Квинта и Мезона показана на рисунке 1, где видно, что в Базе данных проекта Квинта хранится информация обо всех проектах Мезона, а также о хранилище исходных кодов проектов Мезона. В проектах Мезона хранится информация о связи исходных кодов, расположенных в хранилищах, доступных в данном проекте Квинта. Хранение исходных кодов в хранилищах позволяет использовать их между различными проектами Мезона, которые могут располагаться в разных проектах Квинта. За счёт этого достигается переносимость и повторное использование программных кодов.

Интв. № подл.	Подп. и дата
Взаим. интв. №	Интв. №
Подп. и дата	Подп. и дата

Изм	Лист	№ докум	Подп.	Дата	ПФДИ.421457.009 И3.6	Лист
						10

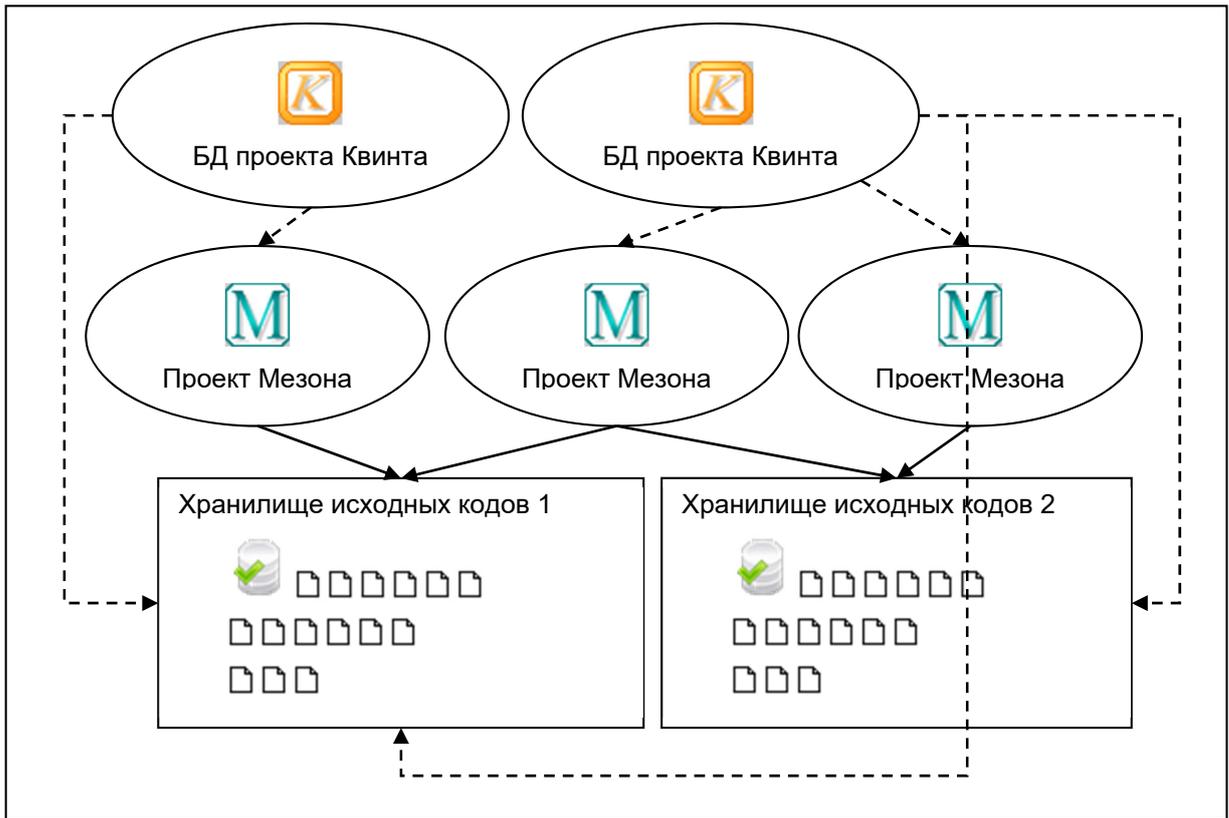


Рисунок 1 – Иерархия проектов Квинта и Мезона

Сам проект Мезона имеет иерархическую структуру, приведённую на рисунке 2.

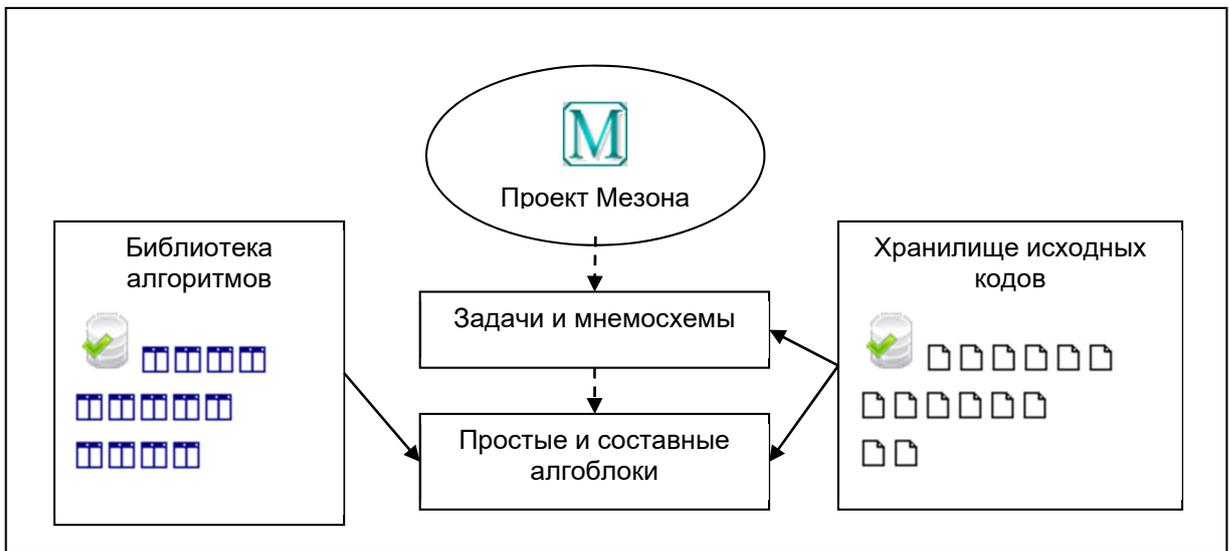


Рисунок 2 – Иерархическая структура проекта Мезона

Как уже было сказано, проект Мезона хранится в проекте Квинта. Физически проект Мезона располагается в одной из пользовательских таблиц БД проекта Квинта. На верхнем уровне проекта расположены задачи и мнемосхемы. Их выполнением можно управлять извне (запускать, останавливать, изменять параметры работы из других задач или мнемосхем). Также можно настроить параметры их автоматического выполнения (например, задать времена циклов выполнения отдельно для каждой задачи). В проекте описываются межзадачные связи и их параметры. Это означает, что одна и та же задача или мнемосхема, будучи использована в разных проектах, может быть по-разному настроена и по-разному управляема извне.

Задачи и мнемосхемы содержат простые и составные алгоблоки. Последние, в свою очередь, также состоят из простых и составных алгоблоков. Уровень вложенности составных алгоритмов практически ничем не ограничивается. Единственное ограничение заключается в том, что не

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

Изм	Лист	№ докум	Подп.	Дата

2 Создание проекта. Шаг за шагом

В данной главе шаг за шагом будут описаны основные действия, выполняя которые, можно, во-первых, познакомиться с возможностями Мезона, во-вторых, научиться создавать проект, начиная идеей проекта и заканчивая развёртыванием готового продукта на конкретном объекте, в-третьих, получить общие навыки обращения с основными инструментами редактора.

В качестве учебного проекта будет рассмотрен процесс создания проекта, выполняющего несложный математический расчёт. Такой выбор учебного примера обусловлен тем, что позволяет наиболее полно раскрыть большинство возможностей Мезона, при этом не требуются какие-либо специфические навыки обращения с ПТК Квинт или глубокие познания в теории автоматического управления процессами.

Практически все шаги описываются следующим образом: вначале идёт подробное описание цели (**выделяется шрифтом**), которую следует достичь на текущем шаге, затем подробно описываются действия по её достижению. Описание логической составляющей приводится лишь в общих словах с соответствующими ссылками на исходные коды (исходники). Так как, во-первых, готовый проект Мезона приведён в Квинте в качестве примера, и при желании можно подробнее ознакомиться с исходниками. А во-вторых, основная цель описания – как можно подробнее раскрыть сам процесс создания программ, а не логику частного примера.

2.1 Шаг 1. Постановка задачи

На данном шаге будет сформулирована и конкретизирована задача. Целью данного шага является постановка задачи и конкретизация (описание вычислений и логики).

Создадим небольшую игру. Сделаем нечто, похожее на «однорукого бандита». По нажатию кнопки случайным образом будут генерироваться три числа. Победа присуждается, если все три числа совпадут.

Итак, с целью мы определились. Теперь попробуем сформулировать задачу более конкретно. Для этого определим ряд требований, которым должна удовлетворять наша программа, это, собственно, и будет являться постановкой задачи.

Программа должна:

- 1 Иметь пользовательский интерфейс.
- 2 Позволять пользователю задать интервал перебора случайных чисел.
- 3 Отображать ход перебора и результаты раунда.
- 4 Запуск поиска решения должен осуществляться по команде, нажатием кнопки.

2.2 Шаг 2. Декомпозиция

На данном шаге будет произведена декомпозиция задачи. Целью данного шага является формализация структуры будущего проекта.

Попробуем определить состав нашей **технологической программы** (далее в тексте – **техпрограммы**).

Согласно требованиям к техпрограмме, сформулированным в п. 1 подраздела 2.1, она должна иметь графический интерфейс, то есть должна содержать мнемосхему с графическими **алглобками (мнемоблоками)**. В принципе, можно всю техпрограмму сделать на мнемосхеме, но в учебных целях мы разделим графический интерфейс и расчётные задачи.

Техпрограмма будет содержать мнемосхему, которая реализует пользовательский интерфейс, задачу, в которой реализуется расчётная часть техпрограммы и задачу, в которой объявляются переменные.

На мнемосхеме будут расположены:

- три поля для отображения чисел;
- поле для отображения сообщений о ходе игры;
- кнопка запуска игры;
- поле для ввода границы интервала перебора чисел.

В расчётной задаче должны быть:

- три блока генерации случайных чисел;
- составной алглобок управления расчётом.

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

Функцией алгоритма управления расчётом является остановка перебора после заданного числа циклов, проверка результатов раунда и выдача результатов.

2.3 Шаг 3. Запуск Мезон-редактора, создание хранилища исходных кодов

На этом шаге проведём необходимую подготовительную работу и создадим новое хранилище для исходных кодов нашего проекта.

В дереве приложений **Квинт-6** раскрываем папку **Проектирование** и запускаем приложение **Мезон-редактор**. После того, как Мезон-редактор загрузит все библиотеки, появится пустое окно (при условии, что Мезон запускается на данном компьютере впервые). Прежде, чем создать проект, следует вначале определить место хранения его исходных кодов.

Для того чтобы добавить хранилище в БД проекта Квинта, следует выбрать пункт главного меню **Файл / Хранилища / Добавить...** В результате появится диалоговое окно, позволяющее ввести символическое имя хранилища (до 40 знаков), а также задать путь к хранилищу. Для этого следует щёлкнуть на кнопку с тремя точками и в появившемся диалоге выбрать локальную или сетевую папку, доступную с данного компьютера. В поле краткого описания хранилища можно ввести произвольный текст подсказки. Флаг **Использовать в библиотеке алгоритмов** позволяет указать, будут ли составные алгоритмы включаться в состав библиотеки алгоритмов. Если такой флаг установлен, то все составные алгоритмы, хранящиеся на первом уровне данного хранилища, попадут в библиотеку алгоритмов и их можно будет выбрать в программе **Мастер добавления алгоритмов**.

Создадим папку с именем **Хранилище пробного проекта** в любом удобном месте и укажем её в качестве хранилища.

Теперь можно приступить к созданию проекта.

2.4 Шаг 4. Создание проекта

На данном шаге будет начато создание проекта. Будут рассмотрены особенности создания проекта Мезона, такие, как выбор сервера выполнения, добавление новой задачи.

Для того чтобы создать новый проект, выберем в главном меню Мезон-редактора пункт **Файл –**

Проект... (или нажмем на кнопку  на панели инструментов). В появившемся мастере работы с проектами нажмем на кнопку **Добавить проект**. Появится диалоговое окно **Добавление проекта**. В поле имя проекта задается символическое название проекта. Введем «Учебный проект». В выпадающем списке **Сервер** выполнения можно выбрать тот сервер, на котором должен выполняться наш проект.

Пока оставим сервер по умолчанию – **Встроенный**. Нажимаем на кнопку **Ок**. Проект появился в списке проектов Мезона. Теперь нажимаем на кнопку **Ок** в мастере **Выбор проекта**. Окно мастера закрывается, а вместо него появится диалог сохранения задачи по умолчанию, с помощью которого можно сразу добавить первую задачу в проект.

Введем имя будущей задачи, к примеру - **Игра**. Нажимаем кнопку **Сохранить**.

Перед нами откроется белое поле нашей задачи.

2.5 Шаг 5. Добавление алгоблоков, создание составного алгоритма

На данном шаге мы научимся использовать диалог выбора алгоритмов для добавления алгоблоков в модуль. Узнаем то такие порядковые номера алгоблоков и для чего они нужны. Познакомимся с принципами работы с группами алгоблоков и узнаем, как изменять их параметры, задавать константы на входах.

Добавить алгоблок в модуль можно одним из способов:

- 1 Выбрать пункт главного меню **Вставка / Простой алгоблок**.

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

- 2 Щелкнуть на свободном поле модуля правой кнопкой мыши и в появившемся контекстном меню выбрать пункт **Добавить алгоблок**.
- 3 Щелкнуть на кнопке  в панели инструментов.

В любом случае результатом станет появление диалога **Выбор алгоритма**.

В разделе **Случайные** выберем алгоблок **СлучЦелое**. Перетащим его мышью на поле задачи. Повторим действия ещё два раза.

После этого добавим из раздела **Общие** четыре алгоблока **Переменная** с типом **«Целое»** и именами **МаксЧисл**, **Число1**, **Число2** и **Число3**. Добавим переменную с именем **Текст** и типом **строковый**. Добавим алгоблок **Останов** из раздела **Управление**.

Если вы человек наблюдательный, то наверняка заметили, что каждый добавленный нами алгоблок автоматически подписывается сверху в виде **Алг. X**, где X – это порядковый номер алгоблока в пределах данного модуля. Этот номер присваивается любому алгоблоку автоматически при добавлении его в модуль. Порядковые номера алгоблоков влияют на порядок их выполнения в техпрограмме. Часто этот порядок должен быть строго определён и никаких вариаций не допускает, реже этот порядок может быть произвольным.

Для примера, представим ситуацию, когда в модуле расположены два алгоблока – сложения и деления. При этом результат сложения используется в качестве делителя в алгоблоке деления. Если алгоблок сложения будет выполняться позже алгоблока деления (т.е. его порядковый номер будет больше), то в самый первый момент значение делителя будет равно значению по умолчанию (т.к. алгоблок сложения ещё не успел ни разу выполниться), т.е. нулю. Таким образом, произойдёт недопустимая операция деления на 0. Проблема решается путём перестановки этих двух алгоблоков таким образом, чтобы алгоблок сложения выполнялся раньше алгоблока деления (т.е. имел меньший порядковый номер).

Если алгоблок удаляется из модуля, в порядковых номерах образуются «дырки». Они ни на что не влияют, т.к. на порядок выполнения одного алгоблока оказывает влияние не конкретный порядковый номер, а знак разности с другим порядковым номером. Если эта разность положительна, то алгоблок с вычитаемым порядковым номером будет выполняться позже алгоблока с уменьшаемым порядковым номером, если отрицательна – будет выполняться раньше. Тем не менее, среда редактора старается заполнять «дырки» в первую очередь. Поэтому при добавлении нового алгоблока в модуль, имеющий «дырки» в порядковой нумерации, ему будет присвоен один из пропущенных порядковых номеров. Так, как это не всегда то, что хотелось бы разработчику, алгоблок следует переместить вверх или вниз по массиву порядковых номеров, так, чтобы он занял нужное место. Более подробную информацию о порядке выполнения алгоблоков и порядковых номерах вы можете почерпнуть в разделе 4.

Изменим порядок алгоблоков, чтобы он был как на рисунке 3.

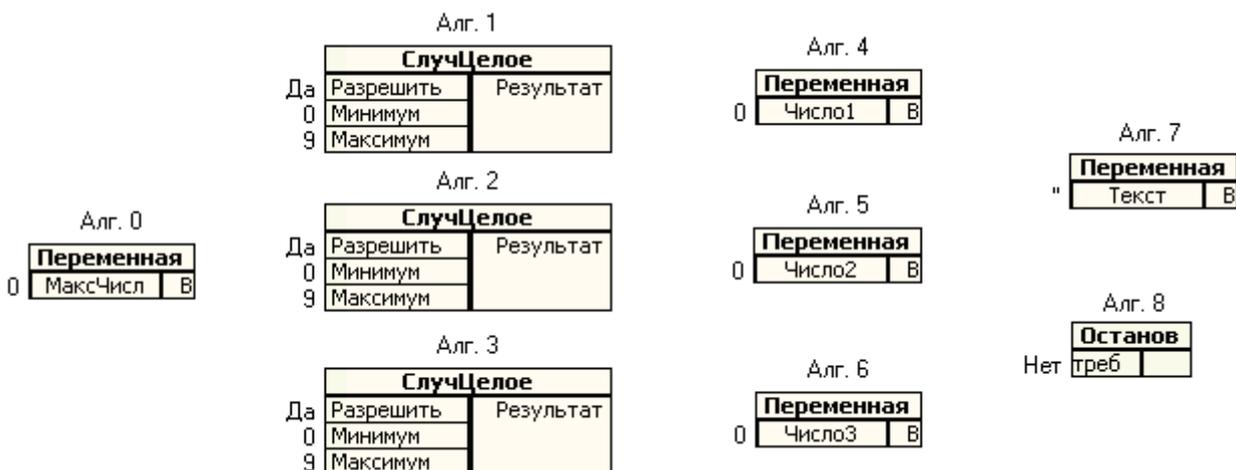


Рисунок 3 – Порядок расстановки алгоблоков в задаче

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

Теперь создадим составной алгоритм.

Для создания любого программного модуля следует воспользоваться пунктом главного меню Мезон-редактора **Файл – Создать....** В появившемся мастере следует выбрать ветвь



Стандартные и указать тип стандартного модуля. В нашем случае, укажем тип **Составной алгоритм** и нажмем **Ок**. Появится уже знакомый нам диалог сохранения модуля проекта. В этом диалоге выберем хранилище. Представляется не очень удобным сохранять модули проекта (задачи, мнемосхемы и составные алгоритмы) все вместе, поэтому заведем в выбранном

хранилище новую папку. Для этого следует щелкнуть на кнопке  и в появившемся диалоговом окне ввести ее имя, например, **Составные алгоритмы**. Откроем вновь созданную папку двойным щелчком, введем новое имя нашего составного алгоритма, например, **Управление задачей**, и нажмем на кнопку **Сохранить**.

После этого должно открыться поле составного алгоритма.

Добавим на него алгоблоки:

- алгоблок **Инициализация** (тип «**действительный**»);
- алгоблок **Сложить**;
- три алгоблока **Сравнить** (тип «**целый**»);
- два алгоблока **Размножить** (тип «**целый**» и «**логический**»);
- три алгоблока **И**;
- алгоблок **Не**;
- два алгоблока **Выбор** (тип «**строковый**»).

Добавленные алгоблоки расположим так, как показано на рисунке 4, изменим их порядок и свяжем их между собой.

Теперь настроим интерфейс составного алгоритма. Добавим группу входов, а в ней три входа с именами **Число1**, **Число2** и **Число3**. Соединим их с входами алгоблоков **Сравнить** и **Размножить** №№ 5, 4, 6 соответственно.

Добавим группу выходов, и в ней выход с именем **Текст**. Соединим его с выходом «**В**» алгоблока **Выбор** № 12.

Добавим канал и в нем вход с именем **Число циклов**, соединим его со входом «**В**» алгоблока **Сравнить** №2 и выход с именем **Конец**, соединим его с выходом «**В1**» алгоблока **Размножить** № 3.

После наших манипуляций составной алгоритм должен выглядеть так, как показано на рисунке 4.

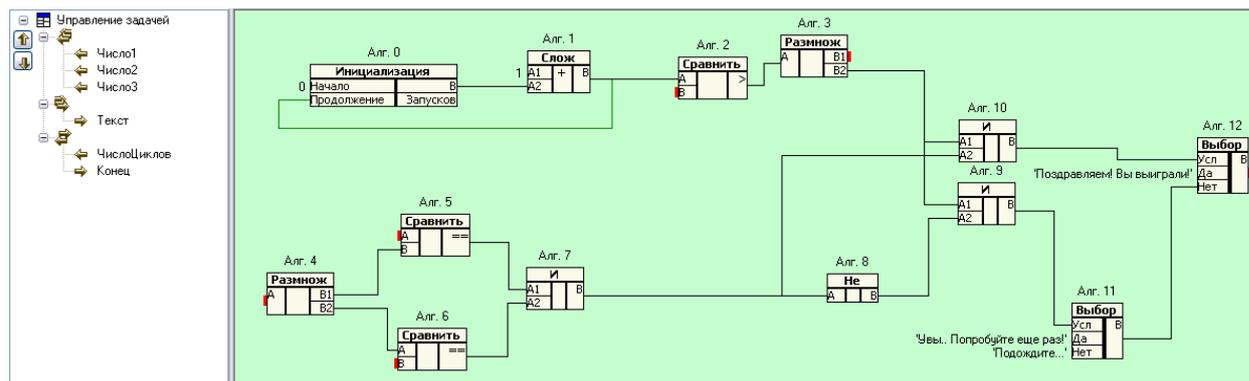


Рисунок 4 – Алгоблоки составного алгоритма Управление задачей

Нажмем кнопку **Сохранить** .

Прейдем на вкладку нашей задачи и добавим созданный составной алгоритм в задачу. Для этого в поле задачи щелкнем ПКМ и в выпадающем списке выберем пункт **Добавить составной алгоблок**. Выберем наш алгоритм и нажмем **Открыть**.

После этого свяжем алгоблоки, изменим их номера и расположим так, как показано на рисунке 5.

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

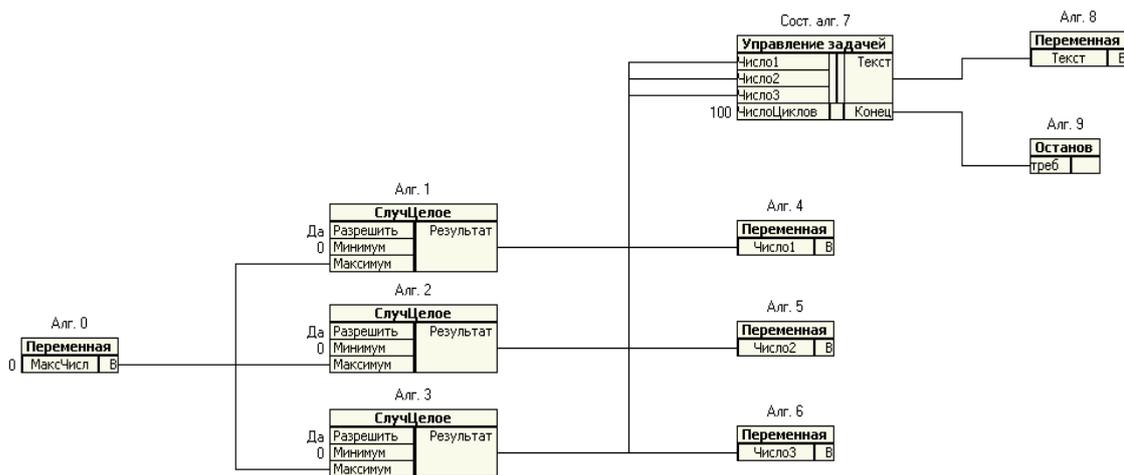


Рисунок 5 – Добавление составного алгоритма в задачу

Теперь можно приступать к созданию мнемосхемы.

2.6 Шаг 6. Создание и настройка мнемосхемы

На этом шаге мы научимся создавать и настраивать мнемосхемы (задачи). Кроме того, узнаем, как просматривать список модулей, входящих в проект.

Для создания мнемосхемы воспользуемся пунктом главного меню Мезон-редактора **Файл / Создать...** В появившемся мастере следует выбрать пункт **Стандартные** и указать тип



мнемосхема. В появившемся диалоге сохранения модуля проекта выберем хранилище, введем имя нашей новой мнемосхемы, например, **Игра**, и нажмем на кнопку **Сохранить**. Мнемосхема будет создана и автоматически добавлена в проект (модули задач и мнемосхем добавляются в проект автоматически при их создании). После этого она будет открыта на отдельной вкладке редактора и готова для редактирования. Если в качестве имени модуля (мнемосхемы или задачи) указать уже имеющийся, и нажать на кнопку **Сохранить**, он будет добавлен в проект и открыт на новой вкладке редактора, при этом информация не будет затерта.

Добавим на мнемосхему алгоблоки: четыре **индикатора**, один **редактор значений** и одну **кнопку**.

После этого скопируем из задачи все переменные и вставим их на мнемосхеме.

Свяжем все алгоблоки так, как показано на рисунке 6.

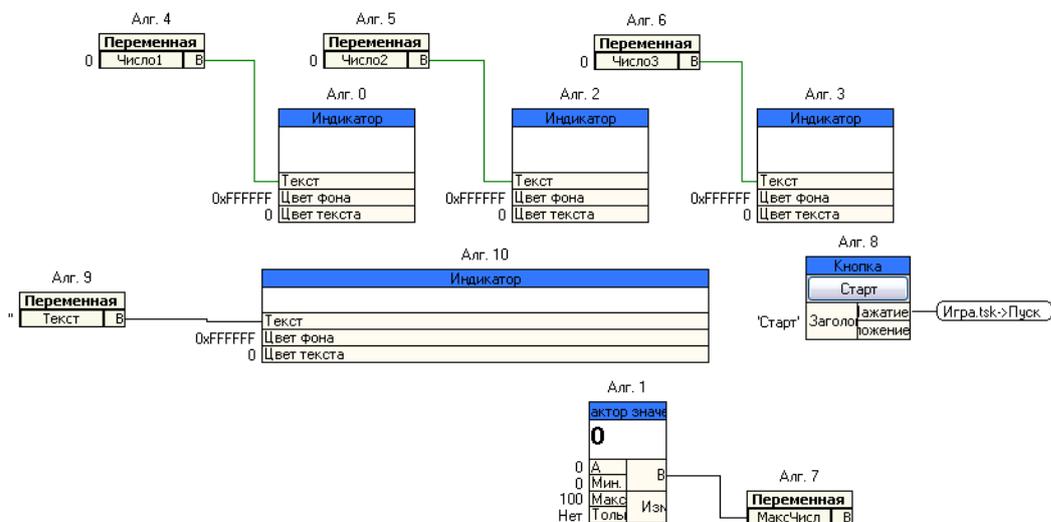


Рисунок 6 - Мнемосхема

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

В свойствах индикаторов и редактора значений можно выбрать желаемый шрифт и его размер.

Теперь необходимо расположить добавленные мнемоблоки нужным образом и придать им требуемый размер.

Вначале придадим одинаковый размер трем **индикаторам**, в которых будут отображаться результаты перебора. Для этого необходимо выделить сразу три этих блока. Это можно сделать двумя способами:

- 1 Щёлкнуть правой кнопкой мыши на свободном месте модуля и не отпуская, переместить курсор так, чтобы в пунктирный прямоугольник попали сразу три индикатора. Если отпустить кнопку мыши, то сразу три мнемоблока окажутся выделенными.
- 2 Удерживая кнопку **Shift** на клавиатуре, последовательно щелкать на заголовках требуемых алгоблоков. Для отмены выделения алгоблока, выделенного любым из способов, следует, удерживая **Shift**, щелкнуть на его заголовке.

Теперь, когда три мнемоблока индикаторов выделены, над ними можно проводить групповые операции.

Придадим индикаторам требуемый размер. Для этого растянем один из них, удерживая и перемещая один из черных маркеров. По завершении операции, когда кнопка мыши будет отпущена, остальные мнемоблоки приобретут точно такие же размеры, что и первый.

Примечание - групповое изменение параметров алгоблоков (в т.ч. и размеров мнемоблоков), возможно только, если в выделенной группе находятся алгоблоки одного типа. Если это не так, изменения коснутся только того алгоблока, к которому они были приложены.

Теперь выровняем мнемоблоки. Для этого выполним пункт меню **Вид / Выравнивание**. Откроется окно с кнопками выбора режимов выравнивания алгоблоков. Нажмем кнопку **Прижать кверху**. Выделенные мнемоблоки окажутся выровненными по горизонтали.

Перемещая любой алгоблок в выделенной группе, мы можем переместить всю группу алгоблоков как единое целое. Разместим, таким образом, мнемоблоки индикаторов в нужном месте мнемосхемы.

Теперь похожим образом разместим остальные мнемоблоки.

Пока что кнопка не содержит никаких поясняющих надписей, а редактор значений не содержит значений по умолчанию. Для того, чтобы это исправить, следует на соответствующих входах этих алгоблоков задать константные значения. Назовем кнопку именем **Старт**. Для этого следует щелкнуть на входе **Заголовок** алгоблока **Кнопка** и в появившейся строке редактирования ввести нужную подпись.

Ввод подтверждается нажатием **Enter**, переходом к редактированию другой константы, либо щелчком на свободном месте модуля. Таким же образом зададим значения по умолчанию для алгоблока Редактор значений. Для этого введем константу на входе **A =5**. Это будет граница диапазона перебора случайных чисел. Сохраним изменения.

При желании, можно задать нашей мнемосхеме фон. В качестве фона можно использовать растровый рисунок, подготовленный в любом графическом редакторе (например, редактор Paint.exe, входящий в комплект поставки Windows). Подготавливаем рисунок, сохраняем его в виде файла с расширением *.bmp. Выбираем пункт главного меню **Вид – Изменить фон...** в появившемся диалоговом окне находим ранее сохраненный файл с фоновым рисунком и нажимаем **Открыть**. В результате фоновый рисунок появляется в качестве подложки мнемосхемы. На этом начальные работы по созданию мнемосхемы закончены.

Чтобы увидеть, как будет выглядеть мнемосхема при работе, выполним пункт меню **Вид / Предпросмотр**.

Выберем пункт **Файл / Модули...** в главном меню Мезон-редактора или нажмем на кнопку . В появившемся диалоговом окне отображаются все модули, входящие в состав данного проекта, в том числе и сам проект. Жирным шрифтом показано полное название модуля (на вкладках редактора оно может сокращаться для экономии места), а обычным – полный путь к данному

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

модулю. Выберем модуль проекта (первая строка). Можно установить флаг **Показывать схему проекта**, тогда в редакторе при открытии проекта будет показана древовидная иерархия его модулей, а между модулями первого уровня (задачами и мнемосхемами) будут показаны межзадачные связи. Однако сейчас этот флаг можно убрать, т.к. наш проект пока слишком прост и его схема нам ничем не интересна. Все – нажимаем на кнопку **Открыть**. Появится вкладка с исходником проекта. На ней в виде алглобка будет изображена наша мнемосхема.

Настроим выполнение задачи и мнемосхемы.

Сначала настроим управление задачи кнопкой **Пуск**. Для этого свяжем выход **Нажатие** кнопки со входом **Пуск** задачи. Таким образом, задача будет запущена, когда придет событие запуска задачи от нажатия кнопки.

Настроим нужным образом выполнение мнемосхемы (см. 2.2).

Щелкнем правой кнопкой мыши на заголовке мнемосхемы, вызовем контекстное меню и выберем пункт **Мастер настройки**. Появится окно мастера, в котором можно задать дополнительные параметры выполнения мнемосхемы. Оставим условие **По умолчанию** без изменений. В списке **Условие выполнения** выберем **Выполнять каждые...мс (без компенс. idle)**, а в цифровом поле введем **100**. Таким образом, мнемосхема будет выполняться каждые 100 мс без компенсации колебаний времени цикла в режиме ожидания. Такой период вполне достаточен для комфортной работы с элементами управления, расположенными на данной мнемосхеме (он определяет время отклика элементов управления). В списке **При исключении** выберем пункт **Идти далее**.

2.7 Шаг 7. Компилирование и запуск проекта

На этом шаге мы научимся компилировать техпрограмму, находить и исправлять ошибки, возникшие во время компиляции. Также мы запустим техпрограмму на выполнение, узнаем, как открывать и закрывать вкладки с исходными кодами проекта.

Нажмем кнопку **Собрать**. В окне отладки появятся сообщения об ошибках (рисунок 7).

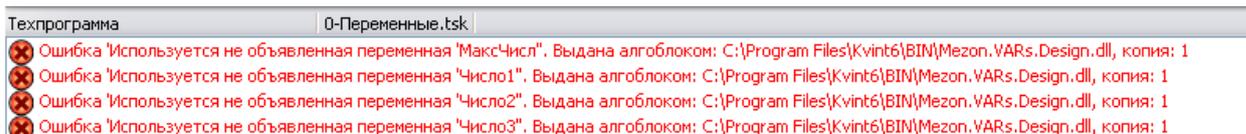


Рисунок 7 – Окно отладки

Это произошло из-за того, что мы забыли объявить переменные, которые использовали в своем проекте.

Можно щелкнуть мышью дважды на строке с ошибкой, при этом откроется задача, в которой она возникла и, при возможности, будет подсвечен алглобок, вызвавший ошибку.

Чтобы исправить ошибки, создадим еще одну задачу, которую назовем **Переменные**. Добавим в нее все использованные нами переменные. Изменим порядок модулей, как показано на рисунке 8. В свойствах задачи поставим **Разрешение = Нет**.

В итоге, проект должен выглядеть, как показано на рисунке 8.



Рисунок 8 – Созданный проект

Соберем проект снова. Теперь ошибок быть не должно.

Нажмем кнопку **F9** или . Наша программа должна запуститься. На входах и выходах алглобков будут видны текущие значения. Можем нажать на кнопку **Старт** и проверить ход игры.

Остановим техпрограмму, нажав на кнопку .

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.

На данном этапе вкладка с проектом нам больше не понадобится, и мы можем ее закрыть. Это можно выполнить одним из следующих способов:

- вызвать контекстное меню на вкладке с проектом и выбрать пункт **Закрыть**;
- выбрать в главном меню Мезон-редактора пункт **Окна / Закрыть / Учебный проект**;
- выбрать пункт **Окна / Выбрать** в появившемся диалоговом окне выделить строку с названием вкладки, на которой открыт **Учебный проект** и нажать на кнопки **Закрыть** и **Отмена**.

2.8 Шаг 7. Загрузка техпрограммы в Мезон-станцию

На этом шаге мы научимся загружать и выгружать техпрограмму из Мезон-станции.

На предыдущем шаге мы уже запускали техпрограмму. Однако, она выполнялась на встроенном сервере (вспомните, что мы его указали на шаге №4). Такой режим используется в основном при отладке техпрограммы. При обычной работе чаще используют не встроенный сервер, а сервер выполнения, работающий на законченном аппаратном устройстве. При этом структура проекта максимально напоминает связку Ремиконт-Операторская станция.

Настроим наш сервер выполнения.

Для начала мы откроем в пакете **Квинт-6** программное приложение **Администратор БД**, а в нем пункт меню **Файл / Абоненты сети**. Из списка компьютеров выберем свой и на вкладке **Компьютер** установим флаг в пункте **Мезон-сервер**.

После этого в свойствах нашего проекта появится возможность указать не встроенный сервер выполнения.

Закроем **Администратор БД** и запустим программу **Конфигуратор**. После этого запустим Мезон-станцию.

В Мезон-редакторе закроем проект, откроем диалог управления проектами и нажмем кнопку **Редактировать проект**. Из списка серверов выберем наш компьютер и нажмем **ОК**.

Откроем наш проект, выберем пункт меню **Сервис / параметры / выполнение** и снимем флаги с пунктов **Всегда создавать новый сервер** и **Всегда разрушать созданный сервер**.

Из раздела **Выполнение** дерева **Квинт-6** запустим Мезон-станцию. После этого запустим задачу на выполнение.

Развернем окно Мезон-станции. В списке техпрограмм должна появиться наша техпрограмма. Это значит, что мы загрузили техпрограмму и она в данный момент выполняется на сервере. Рядом с названием техпрограммы указывается количество подключенных к ней клиентов. У нас оно будет равно 1, т.к. в данном случае клиентом является Мезон-редактор, который подключен к техпрограмме и отображает результаты ее работы.

Остановим техпрограмму, нажав в Мезон-редакторе на кнопку **Стоп** . На вопрос **Ликвидировать сервер?** ответим утвердительно. Посмотрим еще раз в окно Мезон-станции. Список техпрограмм должен быть пуст. Это означает, что мы выгрузили техпрограмму.

2.9 Шаг 8. Настройка и запуск Мезон-клиента

На этом шаге мы научимся настраивать и запускать Мезон-клиент.

Мы создали проект, научились запускать и останавливать техпрограмму.

Теперь узнаем, как настраивать Мезон-клиент для выполнения

Выберем в меню Мезон-редактора пункт **Сервис / командная строка** и скопируем полученный текст в буфер обмена.

Теперь перейдем в дерево **Квинт-6**, выберем пункт меню **Сервис** и в нём команду **Редактировать дерево...**

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

Изм	Лист	№ докум	Подп.	Дата	ПФДИ.421457.009 И3.6	Лист
						20

Добавим приложение **Mezon.exe**. В окне свойств добавляемого приложения укажем в дереве **Квинт-6** имя **Игра**, в поле **Параметры** вставим текст командной строки, полученной в Мезон-редакторе. Нажмем **ОК**. Теперь в дереве **Квинт-6** появился новый пункт с названием **Игра**.

Щелкнем по нему 2 раза. Должна открыться наша мнемосхема, но внизу появится ошибка **Невозможно связаться с сервером**. Это произошло из-за того, что техпрограмма не загружена в Мезон-станцию. Закроем Мезон-клиент и, перейдя в Мезон-редактор, запустим техпрограмму. Нажмем на кнопку **Остановить**, на вопрос **Ликвидировать сервер?** ответим отрицательно.

Посмотрим в окне Мезон-станции информацию о запущенных техпрограммах. Должна выполняться наша техпрограмма с именем **Игра**. Количество клиентов должно быть равно нулю.

Теперь запустим Мезон-клиент из дерева Квинт-6. Теперь ошибок быть не должно, на открывшейся мнемосхеме должны отображаться значения.

Можно играть. Нажмите на кнопку **Старт**.

2.10 Шаг 9. Трассировка и отладка

На этом шаге мы познакомимся с приемами отладки техпрограмм.

Созданная нами техпрограмма простая и все ошибки в ней мы нашли и устранили еще в процессе компиляции. Однако на ней мы познакомимся с приемами отладки и трассировки.

В Мезон-редакторе запустим техпрограмму. На предложение **Выберите сервер** нужно выбрать сервер с именем **Игра**. Окна задач и мнемосхем должны изменить свой вид – на входах и выходах алгоблоков должны появиться текущие значения. Таким образом, мы подключились к техпрограмме, которая сейчас выполняется.

Посмотрим по шагам, как она выполняется.

Поставим техпрограмму на паузу. Нажмем на кнопку **Старт** и будем нажимать на кнопку **Сделать шаг** . При каждом нажатии будет выполняться один шаг, и мы можем увидеть как меняются значения на входах алгоблоков.

Теперь щелкнем правой клавишей мыши на любом алгоблоке и выберем пункт **Дойти до алгоблока** при этом программа выполнит все, что предшествует этому алгоблоку и остановит выполнение на нем. Этим методом мы можем быстро проходить те участки кода, которые нас в данный момент не интересуют.

Теперь научимся работать с точками останова. Выберем любой алгоблок, к примеру, **Индикатор**, расположенный на мнемосхеме. Щелкнем на нем правой кнопкой мыши и выберем пункт **Установить точку останова**. Теперь нажмем кнопку **Запустить**. Программа выполнится один раз и остановится на этом алгоблоке.

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

3 Справочник

3.1 Окно запуска Мезон-редактора

После запуска Мезон-редактора из Квинтегратора, появляется стартовое окно (рисунок 9), отображающее процесс загрузки библиотеки алгоритмов (стандартной и пользовательской, если последняя имеется).



Рисунок 9 – Стартовое окно Мезон-редактора

В окне отображается имя текущей загружаемой библиотеки и название файла плагина, загружаемого в данный момент. Если в процессе загрузки плагинов возникнет ошибка – информация об этом попадет в консоль окна **Квинтегратор**, а также в окно трассировки (рисунок 10) Мезон-редактора.

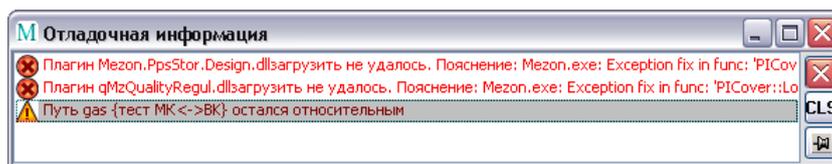


Рисунок 10 – Окно трассировки

3.2 Главное окно Мезон-редактора

После того, как Мезон-редактор загрузит все библиотеки, появится Главное окно, позволяющее начать работу с редактором. Вид окна показан на рисунке 11.

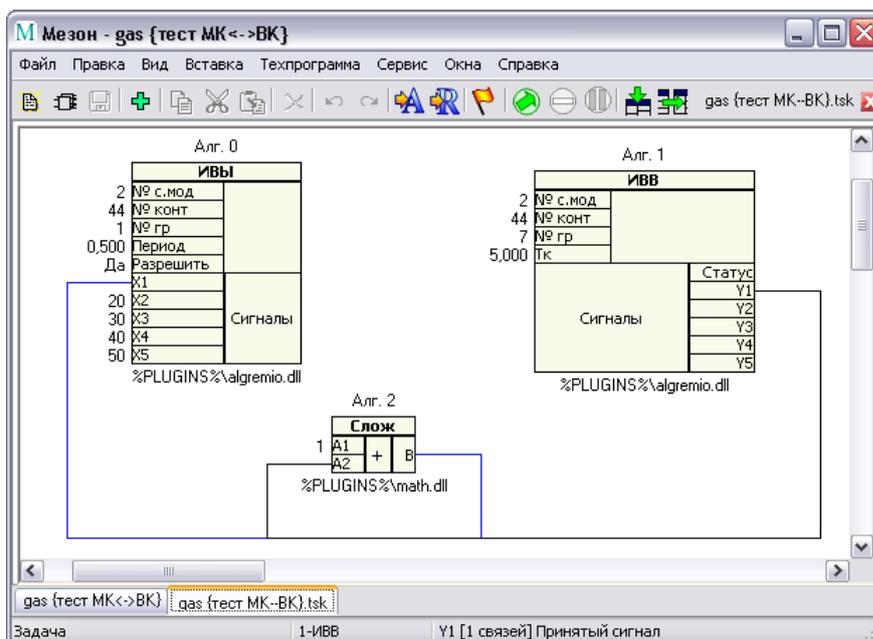


Рисунок 11 – Главное окно Мезон-редактора

В окне содержится **Главное меню Мезон-редактора**, **Панель инструментов**, **Рабочая область** и **Строка состояния**.

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

3.2.1 Главное меню

Главное меню содержит восемь основных пунктов:

- **Файл;**
- **Правка;**
- **Вид;**
- **Вставка;**
- **Техпрограмма;**
- **Сервис;**
- **Окна;**
- **Справка.**

3.2.1.1 Файл

Этот пункт меню позволяет управлять файлами проекта техпрограммы Мезона. В свою очередь содержит следующие подпункты:

- **Техпрограмма.** Открывает **Диалог Проект**, с помощью которого можно открыть, создать, редактировать или дублировать проект техпрограммы Мезона;
- **Заккрыть техпрограмму.** Закрывает текущую открытую техпрограмму Мезона;
- **Импортировать ТП.** Позволяет открывать старые проекты Мезона и конвертировать их в новый формат. После успешного конвертирования новая техпрограмма будет открыта в редакторе;
- **Создать.** Открывает **Диалог Создание модуля проекта**, с помощью которого в существующий проект можно добавить новый проектный модуль (задачу, техпрограмму, составной алгоритм и т.п.);
- **Модули.** Открывает Диалог **Управление модулями** текущего открытого проекта. С его помощью можно добавить в проект имеющийся модуль, удалить модуль из проекта или переименовать модуль в проекте;
- **Хранилища.** Этот пункт содержит три подпункта:
 - а) **Добавить.** Позволяет добавить новое хранилище исходных модулей Мезона в текущий проект Квинта. Выбор этого пункта приведет к появлению Диалог **Настройка хранилища**, с помощью которого можно задать имя, путь и комментарий для нового хранилища;
 - б) **Изменить.** Позволяет изменить свойства уже имеющихся хранилищ исходных модулей Мезона. Вначале появляется диалог выбора хранилища, которое нужно изменить. Если выбрать нужное хранилище из списка и нажать **Ок**, появится Диалог **Настройка хранилища**, который позволит изменить свойства хранилища (например, имя хранилища или путь к нему);
 - в) **Удалить.** Позволяет удалить имеющееся хранилище. При выборе этого пункта будет предложен выбор способа удаления хранилища: виртуальное удаление или перманентное удаление, а также можно отказаться от продолжения данной операции. Рекомендуется использовать виртуальное удаление хранилища.
Примечание - Не удаляйте хранилище, если вы не уверены, что исходные коды, расположенные в нем, нигде больше не используются;
- **Сохранить.** Сохраняет текущий открытый модуль, если в нем сделаны какие-либо изменения. Если в модели не было ни одного изменения с момента его открытия в редакторе, данный пункт будет недоступен;
- **Сохранить все.** Сохраняет все изменения, сделанные во всех открытых в данный момент модулях. Если ни в одном из открытых модулей нет ни одного изменения, пункт будет недоступен;
- **Сохранить как.** Сохраняет текущий модуль под другим именем (при этом тип модуля изменен быть не может). При переименовании модуля создается новый модуль с новым именем, при этом модуль со старым именем остается неизменным. После успешного сохранения, появится диалог, предлагающий поменять старый модуль на только что сохраненный модуль с новым именем в проекте. Если отказаться от этого – в проекте по прежнему останется модуль со старым именем;
- **Выход.** Закрывает Мезон-редактор. Если в одном или нескольких модулях остались несохраненные изменения, появится диалог, позволяющий сохранить все изменения (по одному файлу за раз), закрыть редактор без сохранения изменений или отказаться от закрытия.

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

3.2.1.2 Правка

Этот пункт меню позволяет редактировать технологическую программу. В свою очередь содержит следующие подпункты:

- **Отменить.** Позволяет произвести многошаговую отмену изменений, сделанных в текущем открытом модуле. Для каждого открытого модуля ведется независимый список многошаговых отмен, поэтому команда отмены воздействует на каждый модуль отдельно. Пункт недоступен, если список отмен пуст или достигли его конца;
- **Повторить.** Аналогично по назначению пункту **Правка – Отменить**, с тем исключением, что с его помощью можно повторить любое действие, отмененное ранее. Пункт недоступен, если список отмен пуст или достигли его начала;
- **Вырезать.** Позволяет вырезать в буфер обмена элементы, выделенные в текущем открытом модуле. Выделенные элементы будут удалены из текущего модуля. Пункт недоступен, если ни один элемент не выделен;
- **Копировать.** Позволяет скопировать в буфер обмена элементы, выделенные в текущем открытом модуле. Пункт недоступен, если ни один элемент не выделен;
- **Вставить.** Позволяет вставить из буфера обмена, ранее скопированные в него элементы. Пункт недоступен, если в буфере обмена нет информации, подходящей для вставки в текущий модуль;
- **Удаление.** Выбрав этот пункт можно удалить выделенные элементы. Пункт недоступен, если ни один элемент не выделен;
- **Выделить все.** Позволяет выделить все элементы в текущем открытом модуле;
- **Выделить алгоблоки.** Позволяет выделить простые и составные алгоблоки в текущем открытом модуле. Если текущим является модуль проекта, будут выделены задачи. Выбор этого пункта приведет к появлению списка алгоритмов, которые необходимо выделить. Одновременно можно выделить несколько алгоблоков вручную по названиям или автоматически по типам;
- **Выделить комментарии.** Действие данного пункта аналогично действию пункта **Ошибка!** Источник ссылки не найден. с той разницей, что позволяет выделять комментарии.
- **Перейти к...** Содержит два подпункта:
 - а) **Перейти к алгоблоку.** Позволяет перейти к алгоблоку в текущем открытом модуле. При выборе этого пункта появляется список алгоблоков, в котором следует выбрать нужный алгоблок и нажать **Ок**. Рабочая область модуля будет прокручена так, чтобы алгоблок, к которому произвели переход, оказался в центре окна. Его заголовок будет подсвечен цветом выделения;
 - б) **Перейти к комментарию.** Действие данного пункта аналогично действию пункта **Ошибка!** Источник ссылки не найден. с той разницей, что позволяет переходить к комментариям;
- **Изменить порядок.** Отображает Диалог **Изменить порядок алгоблоков**, с помощью которого можно изменить порядковые номера алгоблоков, тем самым поменять порядок их выполнения на сервере Мезона.

3.2.1.3 Вид

Этот пункт меню позволяет редактировать вид мнемосхемы. В свою очередь содержит следующие подпункты:

- **Изменить фон.** Позволяет добавить или изменить фон мнемосхемы. Пункт доступен, если текущим открытым модулем является мнемосхема;
- **Удалить фон.** Позволяет удалить фон с мнемосхемы. Пункт доступен, если текущим открытым модулем является мнемосхема;
- **Предпросмотр.** Позволяет посмотреть конечный вид мнемосхемы (без элементов редактирования). Для выхода из режима просмотра следует вновь выбрать данный пункт. Пункт доступен, если текущим открытым модулем является мнемосхема;
- **Обновить.** Перерисовывает изображение текущего открытого модуля.

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

3.2.1.4 Вставка

Этот пункт меню позволяет вставлять в технологическую программу алгоблоки, комментарии, задачи и мнемосхемы. В свою очередь содержит следующие подпункты:

- **Простой алгоблок.** Открывает Диалог **Выбор алгоритма**, с помощью которого можно добавить простой или составной алгоритм из библиотеки алгоритмов. Пункт доступен, если текущим открытым модулем является задача, мнемосхема или составной алгоритм;
- **Составной алгоблок.** Позволяет вставить в текущий открытый модуль составной алгоблок, не попадающий в библиотеку алгоритмов. Открывается Диалог **Добавить алгоблок**, диалог **Добавить задачу в техпрограмму**, с помощью которого можно перейти к нужному хранилищу и выбрать требуемый составной алгоблок. Пункт доступен, если текущим открытым модулем является задача, мнемосхема или составной алгоритм;
- **Комментарий.** Позволяет вставить комментарий в текущий открытый модуль. После выбора этого пункта редактор перейдет в режим вставки комментария. За курсором мыши будет следовать изображение пустого комментария, которое можно передвинуть в нужное место модуля и, нажав основную кнопку мыши, установить комментарий на требуемую позицию. После этого комментарий можно отредактировать обычным образом;
- **Задача.** Позволяет вставить задачу в текущий открытый модуль. Открывается Диалог **Добавить алгоблок**, диалог **Добавить задачу в техпрограмму**, при помощи которого можно выбрать ранее сохраненную задачу или мнемосхему (если изменить тип файла). Пункт доступен, если текущим открытым модулем является проект;
- **Мнемосхема.** Позволяет вставить мнемосхему в текущий открытый модуль. Открывается Диалог **Добавить алгоблок**, диалог **Добавить задачу в техпрограмму**, при помощи которого можно выбрать ранее сохраненную мнемосхему или задачу (если изменить тип файла). Пункт доступен, если текущим открытым модулем является проект.

3.2.1.5 Техпрограмма

Этот пункт меню позволяет подготовить технологическую программу к запуску. В свою очередь содержит следующие подпункты:

- **Собрать.** Запускает встроенный компилятор техпрограмм. Если техпрограмма будет собрана без ошибок, она может быть автоматически запущена на выполнение. Для этого в Диалог **Параметры** следует установить опцию **Выполнение / Отладка / Отлаживать** после компиляции. Если во время сборки техпрограммы возникнут ошибки или (и) предупреждения, информация об этом попадет в **Окно трассировки**;
- **Запустить.** Запускает скомпилированную техпрограмму на выполнение. Если в исходных модулях техпрограммы с момента последней компиляции были внесены изменения, техпрограмма вначале будет собрана, а затем запущена. Сервер, на котором будет она будет запущена, задается как опция проекта с помощью Диалог **Проект**. Если по каким-то причинам сервер, на котором должна выполняться техпрограмма, недоступен в течение времени тайм-аута, появится сообщение о невозможности загрузки техпрограммы на сервер. Если техпрограмма уже выполняется, но поставлена на паузу (меню **Техпрограмма / Пауза**), она будет запущена, начиная с первого алгоблока. Пункт доступен, если техпрограмма не запущена или поставлена на паузу;
- **Пауза.** Приостанавливает выполнение запущенной техпрограммы на последнем алгоблоке. Для продолжения работы техпрограммы, следует воспользоваться пунктом меню **Техпрограмма / Запустить**. Пункт доступен, если техпрограмма выполняется на сервере;
- **Остановить.** Прерывает выполнение техпрограммы и выгружает ее из памяти Мезон-сервера. Пункт доступен, если техпрограмма была загружена на Мезон-сервер;
- **Точки останова.** Содержит два подпункта:
 - а) **Перейти.** Позволяет перейти к алгоблоку, на котором установлена точка останова. В Диалог **«Точки останова»** следует выделить нужный алгоритм и нажать **Ок**. Рабочая область модуля будет прокручена так, чтобы алгоблок, к которому произвели переход, оказался в центре окна. Его заголовок будет подсвечен цветом выделения. Пункт доступен, если техпрограмма была загружена на Мезон-сервер. Если техпрограмма загружена, но ни одной точки останова установлено не было – диалог перехода показан не будет;
 - б) **Удалить.** Позволяет удалить одну или несколько точек останова, установленных во время выполнения техпрограммы, с помощью Диалог **«Точки останова»**. Пункт

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

доступен, если техпрограмма была загружена на Мезон-сервер. Если техпрограмма загружена, но ни одной точки останова установлено не было – диалог перехода показан не будет;

- **Выполнить.** Загружает техпрограмму на Мезон-сервер и переводит Мезон-редактор в режим Операторской станции. При этом становятся доступны только мнемосхемы в их законченном виде. Если техпрограмма еще не была загружена на сервер – она будет предварительно загружена. Если техпрограмма была изменена или еще ни разу не компилировалась – она будет скомпилирована и загружена на сервер автоматически;
- **Восстановить данные.** Позволяет восстановить срез данных текущей техпрограммы, выполняющейся на сервере, из файла, сохраненного ранее. При этом восстановятся состояния всех алглоблов, имеющиеся на момент создания среза. Если срез данных соответствует текущей техпрограмме лишь частично (например, с момента создания среза в техпрограмму были внесены изменения), в окне трассировки появится предупреждение о том, что срез восстановлен не полностью. Состояние новых алглоблов, которые появились в техпрограмме после сохранения среза, останется прежним. Состояния алглоблов, которые были удалены, будут отброшены. Если срез более чем на 30 % отличается от текущей техпрограммы вместо предупреждающей надписи появится сообщение о том, что срез восстановить не удалось. Пункт доступен, если техпрограмма была загружена на сервер;
- **Сохранить данные.** Позволяет сохранить мгновенный срез данных на сервере в виде файла с возможностью его восстановления в дальнейшем. Если в момент подготовки среза возникнет ошибка – сообщение об этом будет зафиксировано в окне трассировки. Пункт доступен, если техпрограмма была загружена на сервер.

3.2.1.6 Сервис

Этот пункт меню содержит сервисные функции. В свою очередь содержит следующие подпункты:

- **Командная строка.** С помощью данного пункта можно получить командную строку для запуска Мезона в режиме Операторской станции. Пункт доступен только при открытом проекте Мезона. Для того, чтобы создать ярлык для запуска мезона в режиме Операторской станции в дереве Квинтегратор следует отредактировать состав программ (**Квинтегратор / Сервис / Редактировать дерево**). В поле **Имя файла** написать %installdir%\bin\mazon.exe, в поле **Параметры** полученную командную строку, в поле **В дереве** ввести символическое название ярлыка, а в поле **Описание** – краткое описание того, что будет делать запускаемая Операторская станция. В дереве появится новый ярлык с указанным именем;
- **Объединить в сост. алглобл.** С помощью данного пункта можно объединить алглоблы в составной алглобл. Для этого необходимо выделить объединяемые алглоблы, нажать **Сервис / Объединить в сост. алглобл.** Появится диалог **Создание составного алгоритма**. В нем следует выбрать хранилище, в котором необходимо сохранить файл составного алглобла под любым именем. После этого откроется только что сохраненный модуль и будет предложено вставить в него выделенные ранее алглоблы при помощи Диалог **Копирование алглоблов**. Следует выбрать параметры копирования алглоблов и нажать Ok. В результате алглоблы попадут в составной алгоритм. Затем обычным образом настроить составной алгоритм (создать интерфейс и связать его в выводами алглоблов). Полученный таким образом составной алгоритм можно использовать в задаче (см. 3.2.1.4). Пункт доступен, если в текущем модуле выделены алглоблы;
- **Исправить ссылки.** Позволяет исправить ссылки в техпрограмме, поврежденные в результате переноса исходных модулей или по другим причинам. Если до выбора данного пункта была открыта техпрограмма, она автоматически закроется (будет предложено сохранить изменения, отказаться от сохранения или отменить операцию). После этого появится Диалог **Проект**, с помощью которого следует выбрать проект, в котором необходимо исправить ссылки. После подтверждения выбора, будет запущен механизм проверки ссылок. Если будет найдена ссылка на отсутствующий модуль или алгоритм, появится сообщение, о том какая ссылка была найдена, и будет запрошено подтверждение на ее исправление. Если подтвердить запрос, появится диалог выбора модуля для замены неисправной ссылки. После выбора нужного модуля для замены, все ссылки, ранее указывающие на отсутствующий модуль, будут заменены ссылками на только что выбранный и проверка продолжится. Так будет повторяться до тех пор, пока в проекте не будут исправлены все ссылки или пока операция не будет прервана отрицательным ответом на подтверждение о замене. Если в проекте нет неисправных ссылок или они все

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

были исправлены, появится сообщение о том, что неисправные ссылки не найдены. После этого вновь появится диалог **Проект**, позволяющий выбрать следующий проект для исправления ссылок. Если данная операция больше не требуется, следует в данном диалоге нажать кнопку отмены;

- **Статистика.** Содержит два подпункта:
 - а) **Техпрограммы.** Выводит приблизительную статистику проекта (число алгоритмов, связей, входов – выходов и т.п.). Пункт доступен, если имеется открытый проект;
 - б) **Задачи.** Выводит приблизительную статистику задачи (число алгоритмов, связей, входов – выходов и т.п.). Пункт доступен, если имеется открытый модуль;
- **Обзор плагинов.** Вызывает диалог **Обзор плагинов**. Позволяет просмотреть состав библиотеки алгоритмов на уровне плагинов, а также просмотреть содержание плагинов и их экспортируемые функции;
- **Параметры.** Вызывает Диалог **Параметры**, при помощи которого можно настроить Мезон-редактор.

3.2.1.7 Окна

Закреть. Закрывает окна по одному из списка. Для этого в выпадающем списке следует выбрать нужный пункт с именем окна. Текущее открытое окно в списке помечается флагом. Пункт доступен, если есть хотя бы один открытый модуль;

Закреть все. Закрывает все открытые окна. Проект при этом остается открытым;

Выбрать. Открывает диалог **Управление окнами**, при помощи которого можно активировать или закрыть нужное окно.

3.2.1.8 Справка

- **Руководство.** Открывает справку на основе настоящего документа;
- **Библиотека алгоритмов.** Открывает справку о библиотеке алгоритмов Мезона;
- **О программе.** Вызывает окно **О программе**.

3.2.2 Панель инструментов

Панель инструментов Мезон-редактора в основном дублирует некоторые часто используемые пункты главного меню. Вид панели с комментариями к значкам показан на рисунке 12.

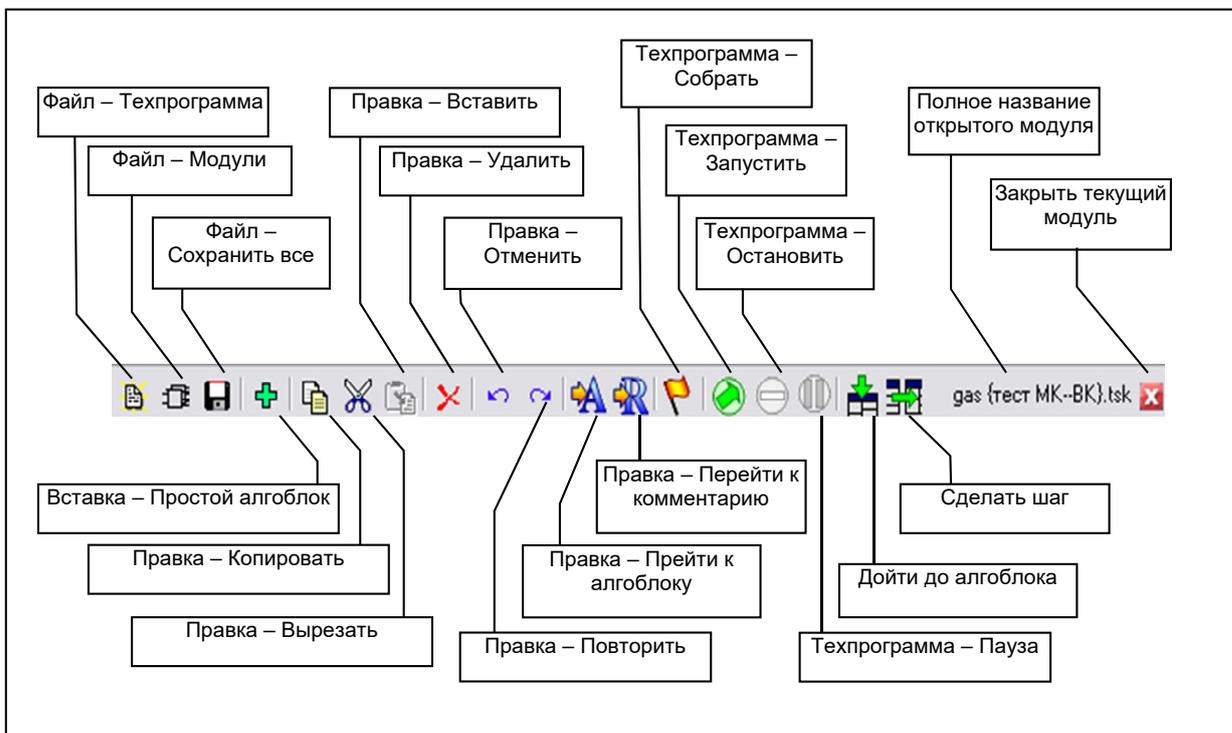


Рисунок 12 – Панель инструментов с комментариями к значкам

Инв. № подл.	Подп. и дата
Взаим. инв. №	Инв. №
Подп. и дата	Подп. и дата
Изм	Лист
№ докум	№ докум
Подп.	Подп.
Дата	Дата

Работа дублирующих кнопок полностью совпадает с работой соответствующих пунктов меню. Две кнопки  (дойти до алглобла) и  (сделать шаг) дублируют пункты контекстного меню алглобла.

Область, в которой отображается полное название модуля, работает следующим образом:

- Если Мезон-редактор работает в режиме редактирования – область названия всегда отображает полное (несокращенное) название модуля с расширением файла (**tsk** – модуль задачи, **msh** – модуль мнемосхемы, **dga** – модуль составного алгоритма). Если расширение отсутствует – значит, текущим модулем является проект;
- Если редактор работает в режиме отладки – область названия будет отображать название модуля в зависимости от контекста его открытия. Для примера пусть в задаче **ЗД1** с порядковым номером в проекте – 0, есть составной алгоритм **СА1** с порядковым номером – 12, содержащий в свою очередь составной алгоритм **СА2** с номером 3. Тогда, если модули **СА1** и **СА2** открыты не контекстно (т.е. при помощи диалога управления модулями или по пункту **Редактировать составной алглобок** контекстного меню составного алглобла, до перехода в режим отладки), их названия будут отображаться по правилу вышеприведенного пункта а). Отладочная информация для модулей, открытых таким образом будет недоступна. Если же модули открыты контекстно (т.е. по пункту **Раскрыть составной алглобок**, контекстного меню составного алглобла или при переходе в данный модуль по срабатыванию точки останова или при отладке по шагам) их название будет построено таким образом: **0-ЗД1 . tsk/12-СА1 . dga** и **0-ЗД1 . tsk/12-СА1 . dga/3-СА2 . dga** соответственно. Запись **0-ЗД1 . tsk/12-СА1 . dga** означает, что составной алглобок **СА1** с порядковым номером 12 контекстно открыт из задачи **ЗД1** с порядковым номером в проекте – 0. Таким образом можно понять контекст открытия составного алглобла. Это удобно, если открывается несколько экземпляров одного алглобла с различными контекстами.

3.2.3 Рабочая область

Рабочей областью принято называть область редактора, в которой отображается состав текущего открытого модуля. Она состоит из окна, в котором располагаются алглобки модуля, областей вертикальной и горизонтальной прокруток окна, а также из вкладок (расположены внизу), определяющих один или более открытых модулей (рисунок 13). Вкладка текущего открытого

модуля выделяется `gas {тест МК-ВК}.tsk`.

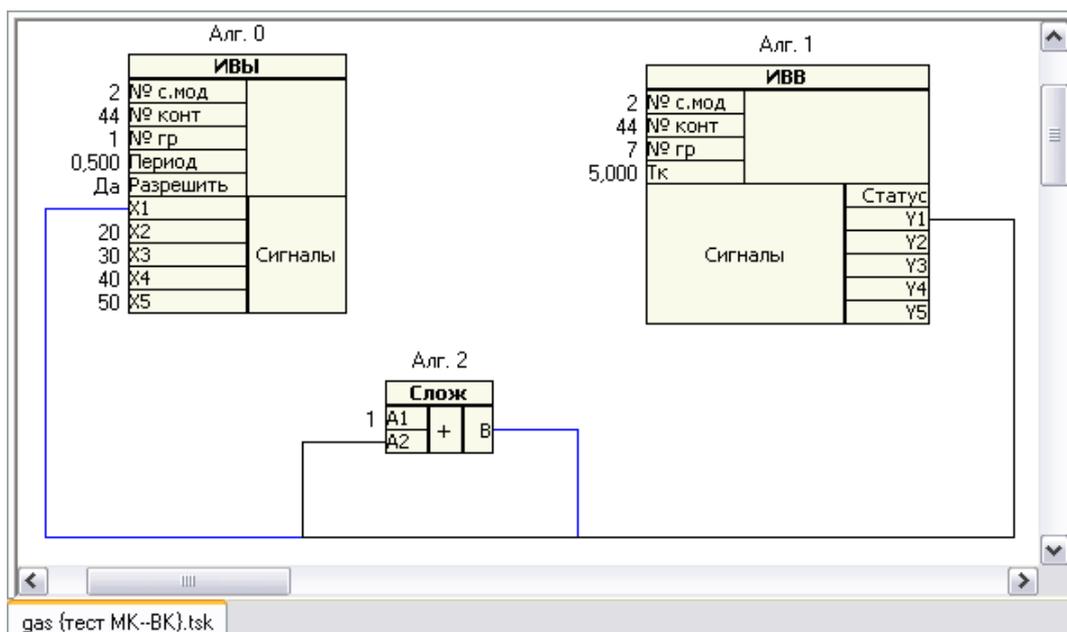


Рисунок 13 – Рабочая область

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

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

Вид рабочей области сильно зависит от того, какой тип модуля в ней открыт в настоящий момент. Всего может быть три различных вида рабочих областей:

- 1 Рабочая область задачи или мнемосхемы.
- 2 Рабочая область составного алгоритма.
- 3 Рабочая область проекта.

3.2.3.1 Рабочая область задачи и мнемосхемы

Вид рабочих областей задачи и мнемосхемы практически совпадает (показан на рисунке 13). Единственное отличие заключается в том, что мнемосхема может иметь фон, отображаемый в рабочей области

3.2.3.2 Рабочая область составного алгоритма

Рабочая область составного алгоритма содержит все элемента рабочей области задачи или мнемосхемы и дополняется окном интерфейса составного алгоблока. При помощи данного окна можно создавать и редактировать интерфейс составного алгоритма. Вид рабочей области показан на рисунке 14.

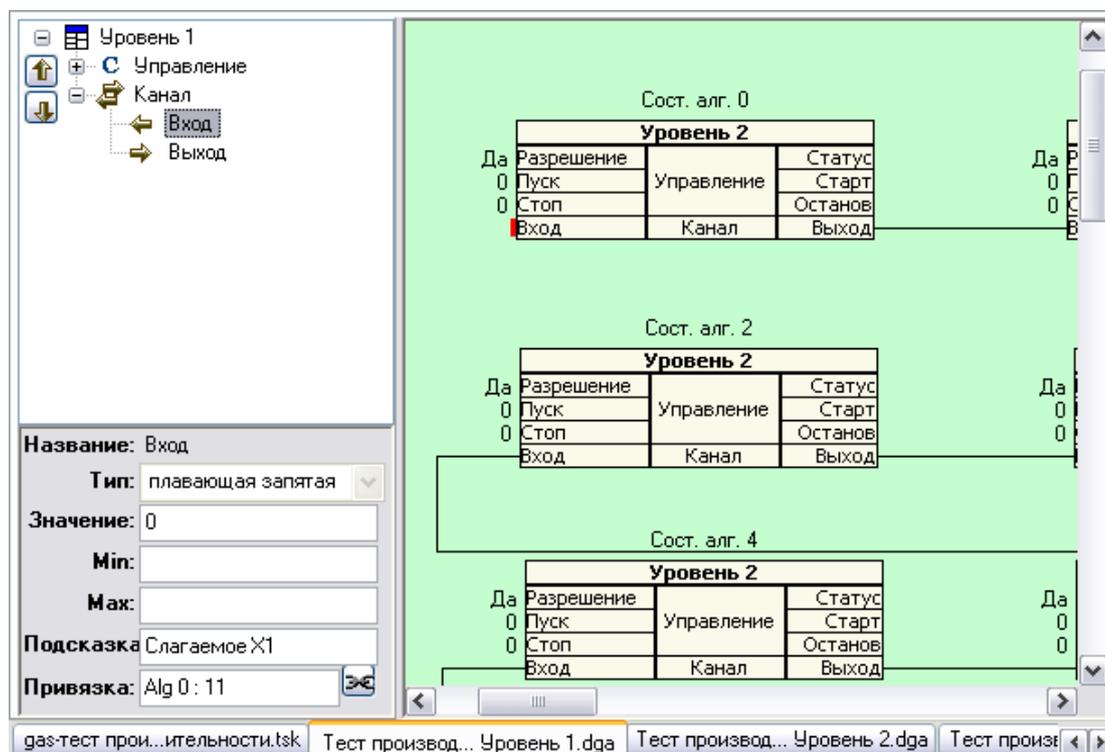


Рисунок 14 – Рабочая область составного алгоритма

Левую сторону рабочей области занимает окно интерфейса составного алгоблока. Интерфейс алгоритма отображается в виде дерева. Корнем дерева является название составного алгоритма, которое будет отображаться в шапке составного алгоблока (на рисунке 14 – «Уровень 1»). Для добавления или удаления элементов в дереве служит контекстное меню окна, содержащее следующие пункты:

- **Добавить управление.** Добавляет в интерфейс составного алгоритма канал управления. С помощью выводов этого канала можно запрещать / разрешать выполнение составного алгоритма по условию на входе **Разрешение** или по событиям на входе **Пуск** и **Стоп**. Выход **Статус** отображает текущий статус составного алгоритма (0 – запущен, 1 – Разрешение = Нет, не был запущен (если вход **Пуск** связан, но события пуска не было) или был остановлен по событию на входе **Стоп**, 8 – остановлен алгоритмом **Останов** или **Ошибка**, 10 – прерван алгоритмом **Сначала**. На выходах **Старт** и **Останов** формируются соответствующие события при старте (запуске алгоблока) по входу **Разрешение=Да** или по событию на входе **Пуск**; и останове по входу **Разрешение=Нет**, по событию на входе **Стоп** или при срабатывании алгоритмов **Останов** или **Ошибка**. Удалить отдельные входы или выходы из канала управления невозможно. Канал можно удалить только целиком. Также выходы этого канала невозможно привязать к выводам алгоблоков, составляющих

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

составной алгоритм. Пункт доступен, если вызвать контекстное меню щелчком на заголовке составного алгоритма (в примере, изображенном на рисунке – это корневая ветвь **Уровень 1**);

- **Добавить группу входов.** Добавляет группу входов. Группа входов визуально объединяет несколько входов составного алгоритма и не несет логической нагрузки. По умолчанию она носит название **Группа входов**, которое можно изменить сразу после добавления группы, либо позже, выделив нужную группу и повторно щелкнув по ней мышью. Пункт доступен, если вызвать контекстное меню щелчком на заголовке составного алгоритма (в примере, изображенном на рисунке – это корневая ветвь **Уровень 1**). При удалении группы входов, все входы, расположенные в ней, также будут удалены;
- **Добавить вход.** Добавляет вход в группу входов или канал. Сразу после добавления входа он получает имя **Вход**. Его можно поменять немедленно или позже, выделив нужный вход и повторно щелкнув по нему мышью. После того, как вход будет добавлен, его можно связать с одним из входов алгоблока, входящего в состав составного алгоритма. При удалении входа привязка к входу алгоблока удаляется автоматически;
- **Добавить группу выходов.** Добавляет группу выходов. Группа выходов визуально объединяет несколько выходов составного алгоритма и не несет логической нагрузки. По умолчанию она носит название **Группа выходов**, которое можно изменить сразу после добавления группы, либо позже, выделив нужную группу и повторно щелкнув по ней мышью. Пункт доступен, если вызвать контекстное меню щелчком на заголовке составного алгоритма (в примере, изображенном на рисунке – это корневая ветвь **Уровень 1**). При удалении группы выходов, все выходы, расположенные в ней, также будут удалены;
- **Добавить выход.** Добавляет выход в группу выходов или канал. Сразу после добавления выхода он получает имя **Выход**. Его можно поменять немедленно или позже, выделив нужный выход и повторно щелкнув по нему мышью. После того, как выход будет добавлен, его можно связать с одним из выходов алгоблока, входящего в состав составного алгоритма. При удалении выхода привязка к выходу алгоблока удаляется автоматически;
- **Добавить канал.** Добавляет канал. Канал визуально объединяет несколько входов и выходов составного алгоритма и не несет логической нагрузки. По умолчанию он носит название **Канал**, которое можно изменить сразу после добавления, либо позже, выделив нужный канал и повторно щелкнув по нему мышью. Пункт доступен, если вызвать контекстное меню щелчком на заголовке составного алгоритма (в примере, изображенном на рисунке – это корневая ветвь **Уровень 1**). При удалении канала, все входы и выходы, расположенные в нем, также будут удалены;
- **Удалить.** Удаляет ту ветвь, для которой было вызвано контекстное меню, а также все её дочерние ветви. Если вызвать пункт **Удалить** для корневой ветви, то будет удален весь интерфейс составного алгоблока целиком, при этом корневая ветвь останется;
- **Связывание интерфейсных выводов.** Для того, чтобы настроить интерфейс составного алгоритма, следует связать его входы и выходы со входами и выходами алгоблоков, входящих в его состав. Для этого следует выделить выход или вход, который необходимо связать, нажать кнопку  и затем щелкнуть на входе или выходе нужного алгоблока в составном алгоритме. Связыванию поддаются только те выводы алгоблоков, которые не имеют других связей. Один вывод алгоблока в составе составного алгоритма может иметь не более одной связи с его интерфейсным выводом. Для обхода этих ограничений, следует использовать алгоритмы **Размнож** (алгоритм размножения). После того, как вывод алгоблока будет привязан к выводу интерфейса, он помечается красным маркером: . Если установить курсор мыши над этим выводом, в дереве интерфейса составного алгоритма, произойдет выделение того вывода интерфейса, к которому данный вывод привязан. Сразу после привязывания интерфейсного вывода его параметры будут автоматически настроены редактором. Для того, чтобы поменять эти настройки, служит окно, расположенное ниже дерева интерфейса (рисунок 15). В этом окне отображается название вывода. Для его смены, следует выделить нужный вывод и затем еще раз щелкнуть мышью, чтобы перейти в режим редактирования названия. Тип вывода жестко определяется типом вывода алгоблока, с которым он связан (если вывод еще не связан, тип вывода может быть изменен с помощью выпадающего списка).

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

Название: X1
 Тип: плавающая запятая
 Значение: 0
 Min: -3e308
 Max: 3e308
 Подсказка: Число для возведения в
 Привязка: Alg 1 : 1

Рисунок 15 – Окно редактора связей

Если вывод является входом, то в полях **Значение**, **Min** и **Max** отображаются значения сигнала по умолчанию (подставляется автоматически, сразу после установки составного алгоритма в задачу), минимальное и максимальное значения сигналов для данного типа (контролируются на стадии редактирования значений сигнала с помощью встроенных мастеров редактирования Мезон-редактора). Сразу после связывания входа эти значения берутся у того входа алгоблока, с которым произошло связывания. Если есть необходимость – эти значения могут быть изменены вручную. Изменять значения следует после связывания вывода, т.к. если это сделать до связывания, они будут затерты значениями, взятыми у алгоблока.

Если вывод является выходом – поля **Значение**, **Min** и **Max** становятся недоступными. Поле **Подсказка** по умолчанию после связывания заполняется соответствующим значением вывода алгоритма. После связывания, подсказка может быть изменена так, чтобы нести смысловую нагрузку, описывая функциональность этого вывода в контексте составного алгоритма.

Поле **Привязка** служит для отображения информации о том с каким выводом какого алгоблока связан данный вывод интерфейса. Запись **Alg 1 : 1** означает, что данный вывод связан с выводом с идентификатором 1 алгоблока с порядковым номером 1. Значение этого поля изменяется в момент привязки или перепривязки вывода интерфейса.

Для того, чтобы перепривязать связанный ранее вывод интерфейса, достаточно выделить нужный вывод, нажать на кнопку  и выбрать другой вывод алгоблока.

Для того, чтобы изменения, внесенные в интерфейс составного алгоритма вступили в силу и могли быть отображены редактором, следует сохранить модуль составного алгоритма.

3.2.3.3 Рабочая область проекта

Вид рабочей области проекта зависит от того в каком режиме открывается модуль проекта (см. 3.4.4). Если модуль открывается с установленным флагом **Показать схему проекта**, задачи проекта будут отображаться с дополнительными входами, демонстрирующими какие входы каких алгоблоков в задаче участвуют в межмодульном обмене (рисунок 16).

В левой части окна появится дерево, отображающее структуру проекта с иерархической информацией о составных алгоритмах. В правой части будут отображаться задачи и мнемосхемы с детализацией связей. При этом все глобальные связи между алгоблоками разных модулей на этом уровне будут показаны в виде локальных связей между задачами и мнемосхемами.

Из дерева проекта можно открыть любой модуль, расположенный на любом уровне вложенности (модуль открывается не контекстно). Для этого нужно два раза щелкнуть на нужной ветви дерева или выбрать пункт контекстного меню .

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

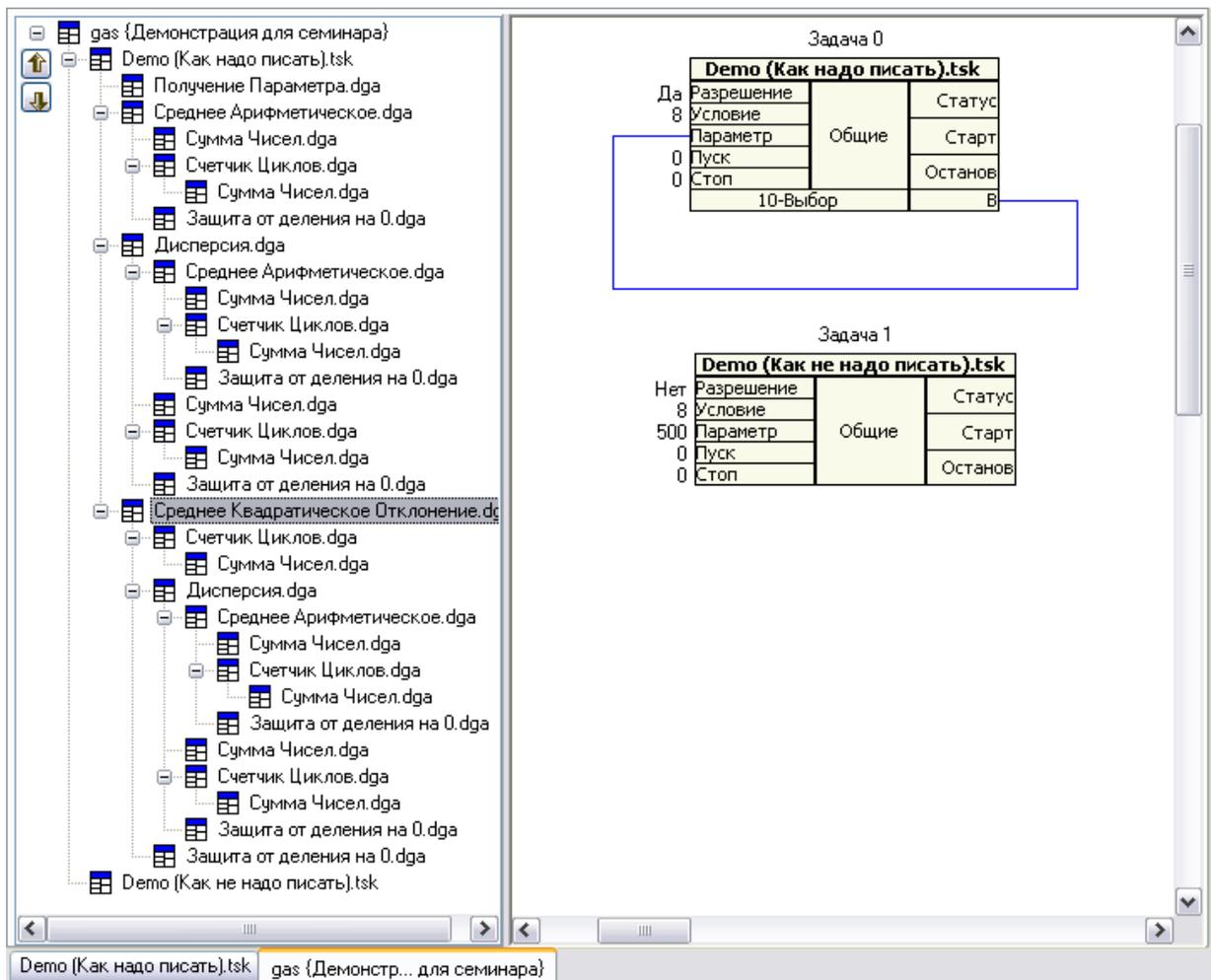


Рисунок 16 – Рабочая область проекта

3.2.3.4 Контекстное меню модуля

Вид контекстного меню (рисунок17 и рисунок 18) зависит от типа модуля и режима, в котором работает Мезон-редактор (редактирование, отладка).

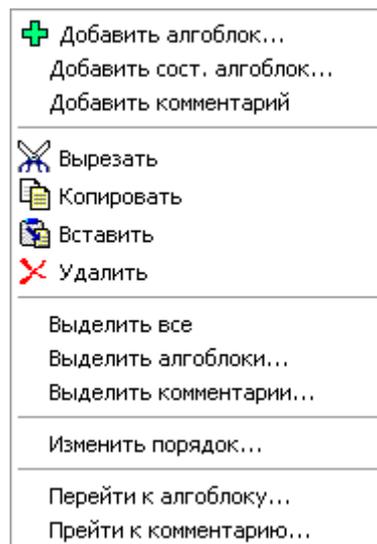


Рисунок 17 - Контекстное меню задачи, мнемосхемы, составного алгоритма

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

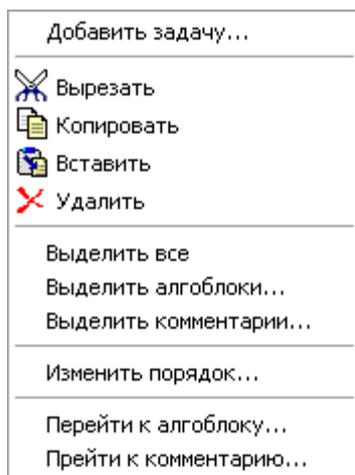


Рисунок 18 Контекстное меню проекта

Пункты контекстного меню модуля дублируют некоторые наиболее часто используемые пункты главного меню Мезон-редактора:

- **Добавить алгоблок** дублирует пункт главного меню **Вставка – Простой алгоблок**;
- **Добавить сост. алгоблок** дублирует пункт главного меню **Вставка – Составной алгоблок**;
- **Добавить комментарий** дублирует пункт главного меню **Вставка – Комментарий**;
- **Добавить задачу** дублирует пункт главного меню **Вставка – Задачу** и **Вставка – Мнемосхему**;
- **Вырезать** дублирует пункт главного меню **Правка – Вырезать**;
- **Копировать** дублирует пункт главного меню **Правка – Копировать**;
- **Вставить** дублирует пункт главного меню **Правка – Вставить**;
- **Удалить** дублирует пункт главного меню **Правка – Удалить**;
- **Выделить все** дублирует пункт главного меню **Правка – Выделить все**;
- **Выделить алгоблоки** дублирует пункт главного меню **Правка – Выделить алгоблоки**;
- **Выделить комментарии** дублирует пункт главного меню **Правка – Выделить комментарии**;
- **Изменить порядок** дублирует пункт главного меню **Правка – Изменить порядок**;
- **Перейти к алгоблоку** дублирует пункт главного меню **Правка – Перейти к алгоблоку**;
- **Перейти к комментарию** дублирует пункт главного меню **Правка – Перейти к комментарию**.

Примечание - Для того чтобы при вызове контекстного меню не сбрасывалось выделение объектов, и можно было пользоваться пунктами контекстного меню **Вырезать**, **Копировать**, **Вставить**, **Удалить**, следует выделить нужные объекты любым способом, а затем, удерживая кнопку **Ctrl** на клавиатуре, вызвать контекстное меню модуля.

3.2.3.5 Контекстное меню алгоблока

Набор и доступность пунктов контекстного меню алгоблока (рисунки 19, 20 и 21), зависит от типа алгоблока (простой, составной), его функциональных возможностей (наличие мастера настройки, справки), а так же от режима, в котором находится Мезон-редактора (редактирование, отладка).



Рисунок 19- Контекстное меню алгоблока в режиме редактирования

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

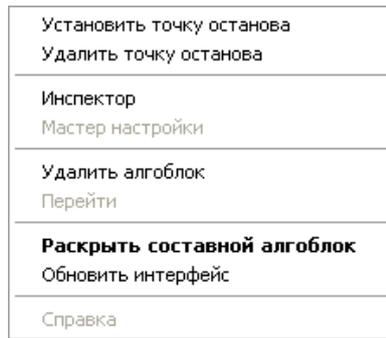


Рисунок 20 - Контекстное меню алглобка в режиме отладки (техпрограмма запущена)



Рисунок 21 - Контекстное меню алглобка в режиме отладки (техпрограмма в паузе)

- **Дойти до алглобка.** Позволяет автоматически выполнить технологическую программу до данного алглобка и затем перевести ее на паузу. При необходимости будет автоматически выполнена компиляция и загрузка техпрограммы на указанный сервер. В результате (если и стадии компиляции и загрузки ТП прошли успешно) как только очередь выполнения алглобков дойдет до данного, техпрограмма будет переведена на паузу и рядом с алглобком появится маркер, указывающий, что выполнение техпрограммы прервано на нем (рисунок 22);

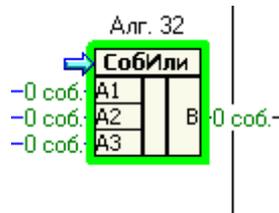


Рисунок 22 – Выделение алглобка в паузе

Рабочая область модуля будет прокручена так, чтобы алглобок, на котором прервано выполнение, оказался в видимой части окна.

- **Сделать шаг.** Позволяет автоматически выполнить техпрограмму так, что очередь выполнения алглобков сместится на один алглобок, согласно порядку их выполнения. При этом техпрограмма будет переведена на паузу на следующем по порядку алглобке. При необходимости будет автоматически выполнена компиляция и загрузка техпрограммы на указанный сервер. За один шаг происходит переход от одного алглобка к следующего за ним по порядку. В результате, если до следующего алглобка выполняются алглобки неявного преобразования типов сигналов (подставляются компилятором), техпрограмма будет прервана на одном из них. Т.к. эти алглобки не отображаются в окне редактора, то ни один алглобок при этом может оказаться не выделенным. Для того, чтобы перейти к следующему неавтоматически сгенерированному алглобку, следует несколько раз повторить команду **Сделать Шаг**.
Рабочая область модуля будет прокручена так, чтобы алглобок, на котором прервано выполнение, оказался в видимой части окна;

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

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

- **Установить точку останова.** Команда позволяет установить точку прерывания выполнения технологической программы на выбранном алгоблоке. При этом рядом с изображением алгоблока появится маркер точки останова (рисунок 23).



Рисунок 23 – Алгоблок с точкой останова

Как только очередь выполнения дойдет до алгоблока, на котором установлена точка останова, выполнение техпрограммы будет прервано и она будет переведена на паузу. При этом маркер изменит свой вид (рисунок 24).

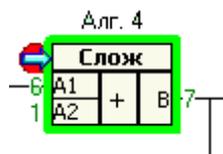


Рисунок 24 – Прерывание выполнения техпрограммы в точке останова

- Пункт доступен, только, если Мезон-редактор находится в режиме отладки техпрограммы;
- **Удалить точку останова.** Удаляет ранее установленную точку останова. Изображение маркера точки останова при этом исчезает. Пункт доступен, только, если Мезон-редактор находится в режиме отладки техпрограммы;
- **Инспектор.** Вызов диалога **Инспектор алгоблоков**. Если вызвать контекстное меню алгоблока на его входе или выходе, в дереве инспектора алгоблоков будет выделена ветвь, соответствующая данному выводу;
- **Мастер настройки.** Вызов **Мастера настройки алгоблока**. Вид и функциональность **Мастера настройки алгоблока** может отличаться от того, было ли вызвано меню на шапке алгоблока или на одном из его выводов. К примеру, для алгоритма АЦП из группы **Моделирование**, вызов Мастера при щелчке на шапке алгоритма откроет диалог настройки канальности алгоритма, а вызов Мастера при щелчке на одном из входов – откроет диалог настройки шкалы для данного входа.
Мастера настройки специфичны для каждого алгоблока или группы алгоблоков и описываются отдельно в справке по соответствующим алгоритмам. Пункт доступен, если Мезон-редактор находится в режиме редактирования и алгоблок имеет Мастера настройки;
- **Удалить алгоблок.** Удаляет тот алгоблок, на котором было вызвано контекстное меню;
- **Перейти.** Осуществляет переход по связи от вывода текущего алгоритма до связанного вывода другого алгоритма. Если вывод имеет более одной связи, будет открыт **Инспектор алгоблоков**, позволяющий уточнить переход по какой связи необходим, а также перейти по выбранной связи. Пункт доступен только тогда, когда контекстное меню было вызвано на выводе алгоблока и этот вывод имеет не менее 1 связи;
- **Редактировать составной алгоблок.** Позволяет открыть составной алгоблок в виде модуля и начать его редактирование. Пункт доступен только при вызове контекстного меню для составного алгоблока и только в режиме редактирования;
- **Раскрыть составной алгоблок.** Пункт позволяет контекстно открыть составной алгоблок в виде модуля в режиме отладки. Контекстно открытый составной алгоблок идентифицируется длинным именем, поясняющим контекст его открытия. В этом случае, открытый таким образом составной алгоблок, может быть отлажен (видна отладочная информация, есть возможность установить точки останова и пр.). Пункт доступен в режиме отладки;
- **Обновить интерфейс.** Позволяет перерисовать внешний вид составного алгоблока, в случае, если в него были внесены изменения (например, с помощью окна редактирования интерфейса составного алгоблока). Для перерисовки можно также воспользоваться пунктом главного меню **Вид - Обновить**;
- **Справка.** Отображает справку по соответствующему алгоритму. Пункт доступен, если справка экспортируется алгоритмом.

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

3.2.3.6 Контекстное меню комментария

Пункт меню предназначен для работы с **Комментариями** техпрограммы. Содержит два подпункта:

- **Мастер настройки.** Позволяет вызвать Мастера настройки комментария (рисунок 25). С его помощью можно добавить или изменить текст комментария, а также выровнять текст относительно краёв комментария с помощью кнопок   . Для отмены изменений можно щёлкнуть в любом месте рабочей области модуля или нажать кнопку **Cancel**;

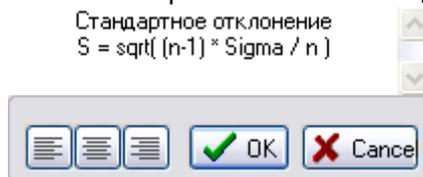


Рисунок 25 – Окно вызова Мастера настройки комментария

- **Удалить комментарий.** Позволяет удалить комментарий, для которого было вызвано контекстное меню.

3.2.3.7 Контекстное меню закладки

Пункт меню предназначен для работы с **Закладками**. Содержит три подпункта:

- **Выбрать.** Дублирует пункт меню **Файл – Модули**;
- **Закреть.** Дублирует кнопку  на **Панели инструментов**;
- **Сохранить.** Дублирует пункт меню **Файл – Сохранить**.

3.2.4 Строка состояния

Вид строки состояния зависит от режима работы редактора и выполняемых им в текущий момент действий.

В режиме редактирования строка состояния имеет вид, показанный на рисунке 26.



Рисунок 26 – Вид строки состояния в режиме редактирования

В первой колонке отображается тип текущего открытого модуля (в примере **Сост. Алг.** – составной алгоритм). Во второй колонке отображается порядковый номер алгоблока и его символическое название (в примере – порядковый номер 8, алгоблок **Слож** – сложение). В третьей колонке – отображается дополнительная информация, зависящая от того, на что указывает курсор мыши. Если курсор указывает на вывод алгоблока, то в колонке будет отображаться название вывода, число связей с ним, а также подсказка.

В режиме выполнения «долгих» операций строка состояния отображает краткое описание совершаемой операции, индикатор прогресса выполнения и, если для операции допускается отмена, кнопку отмены (рисунок 27).



Рисунок 27 – Вид строки состояния в режиме выполнения «долгих» операций

Для отмены длительно выполняющейся операции в любом месте её выполнения можно нажать на кнопку **Отмена**.

В режиме отладки в первой колонке отражается состояние задачи или мнемосхемы (рисунок 28).



Рисунок 28 – Вид строки состояния в режиме отладки

Всего может быть 5 состояний:

- **Выполняется.** Задача или мнемосхема выполняется максимально быстро в каждом расчетном цикле;
- **Не выполняется.** Задача или мнемосхема не выполняется по любой причине (нет разрешения, остановлена по входу **Стоп**, выполнился алгоритм **Ошибка** и пр.);

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

- **Прервана.** Задача или мнемосхема завершила свой цикл не до конца (была прервана алгоритмом **Сначала**);
- **Ожидает свой цикл.** На входе **Условие** задано значение 4 (выполняться один раз в N циклов);
- **Ожидает своё время.** На входе **Условие** задано значение 5 – 8 (выполняться один раз в N мсек).

Состояния могут «мигать», т.к. статус задачи может быстро меняться по изменению внутренних или внешних условий выполнения.

3.3 Окно трассировки

Окно трассировки Мезон-редактора предназначено для вывода служебной информации, а также ошибок и предупреждений от диагностических средств редактора. Вариант вида окна показан на рисунке 29.



Рисунок 29 – Вариант вида окна трассировки

Для выделения типа сообщения многие из них снабжаются значками.

- . Это информационное сообщение;
- . Это предупреждение;
- . Это сообщение об ошибке.

Некоторые сообщения являются интерактивными, т.е. позволяют отыскать источник их возникновения. Для этого достаточно подвести курсор мыши к строке с таким сообщением. Если курсор изменит свой вид на изображение руки – значит, сообщение интерактивное и двойной щелчок на нем позволит найти источник. Как правило, интерактивными являются сообщения, так или иначе связанные с алгоблоками. Например, при компиляции некоторых алгоблоков могут появиться ошибки или предупреждения. Тогда двойной щелчок на таких сообщениях откроет соответствующий модуль, и проблемный алгоблок будет выделен (на рисунке примером интерактивного предупреждения является первая отображаемая строка «Связь может привести к побочному эффекту, вход: 1»).

Для очистки сообщений служит кнопка

Для того, чтобы открепить окно трассировки (или прикрепить к нижней границе основного окна редактора) служит кнопка

Для сокрытия окна трассировки служит кнопка

Многие существенные сообщения об ошибках и предупреждения дублируются также в консоли окна **Квинтегратор** и попадают в лог-файл (часто содержание сообщения в лог-файле может несколько отличаться от того, что отображается в окне трассировки, т.к. дополняется информацией, необходимой разработчикам программы).

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

Если число сообщений, выводимых в окне трассировки, превышает 1000 – новые сообщения перестают появляться, вместо них в последней строке выводится предупреждение **Слишком много ошибок... [2154]**, где число в квадратных скобках означает общее количество сообщений (те, что попали в лог-файлы, плюс те, которые были отброшены).

3.4 Диалоги

3.4.1 Диалог Проект

Этот диалог появляется, если выбрать пункт меню **Файл / Проект...** или нажать на кнопку . Вариант вида окна диалога показан на рисунке 30.

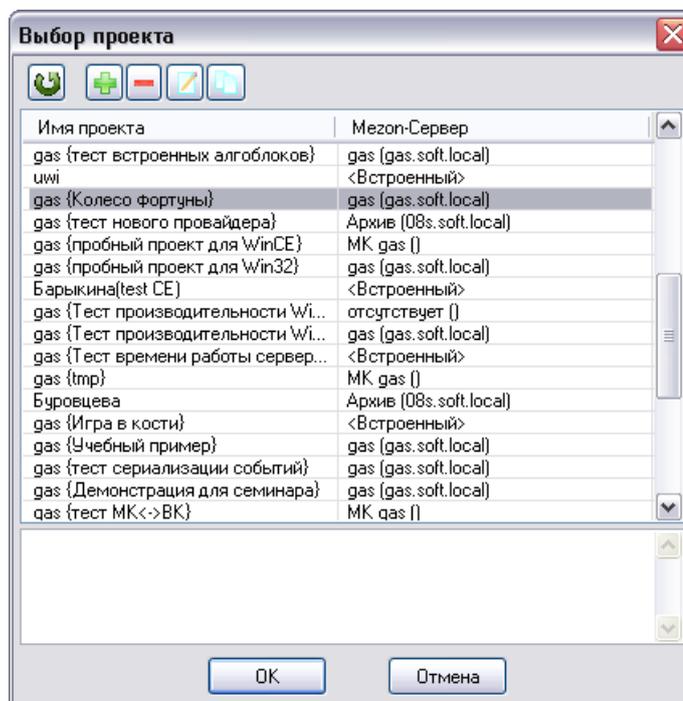


Рисунок 30 - Вариант вида диалогового окна Проект

С помощью диалогового окна можно выбрать проект, созданный ранее, поменять настройки существующего проекта, удалить проект, скопировать проект под другим именем или создать новый проект. Для выбора проекта следует выделить его название и нажать на кнопку **Ок**.

Для изменения настроек существующего проекта, следует выбрать его в списке проектов и нажать на кнопку . В появившемся диалоговом окне **Редактирование проекта**, следует ввести имя проекта (допускается не более 40 символов), выбрать сервер выполнения (по умолчанию – всегда выбирается встроенный сервер выполнения) и задать комментарий (не более 200 символов). В список серверов выполнения попадают все имена компьютеров, для которых в приложении **Администратор БД** разрешён запуск приложений типа **Мезон-сервер** (меню **Администратора БД / Файл / Абоненты сети / Компьютеры**), а также все **Мезон-контроллеры**. Для применения новых настроек проекта, следует нажать на кнопку **Ок**. Если проект, настройки которого необходимо поменять, открыт в **Мезон-редакторе**, его следует предварительно закрыть при помощи меню **Файл / Закрывать проект** и лишь потом приступить к редактированию.

Для добавления нового проекта следует воспользоваться кнопкой  диалога **Выбор проекта**. Появится диалоговое окно **Добавление проекта**, функционально аналогичное описанному выше. При создании нового проекта, **Мезон-редактор** предложит сохранить новую задачу с именем, совпадающим (насколько это возможно) с именем проекта. Это имя можно поменять до сохранения задачи. От сохранения задачи по умолчанию можно отказаться, а вместо нее выбрать уже имеющуюся задачу или мнемосхему.

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

Для удаления существующего проекта, следует выбрать проект в списке проектов и нажать на кнопку . Если удаляемый проект открыт в Мезон-редакторе, его следует предварительно закрыть при помощи меню **Файл / Закрыть проект**.

Для копирования проекта следует выделить тот проект, копию которого необходимо получить, и нажать на кнопку . Появится диалоговое окно **Копирование проекта**, которое функционально аналогично диалоговому окну **Редактирование проекта**. С его помощью можно задать имя копии проекта, изменить комментарий и выбрать другой сервер выполнения. Для завершения создания копии проекта, следует щёлкнуть кнопку **Ок**.

3.4.2 Диалог Инспектор алглоблоков

Вариант вида окна диалога показан на рисунке 31.

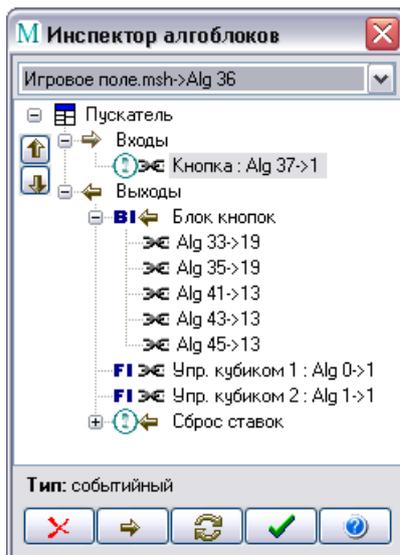


Рисунок 31 - Вариант вида диалогового окна Инспектор алглоблоков

В Инспекторе все входы и выходы алглоблока представлены в виде дерева. Корнем дерева является сам алглоблок, а ветвями – его входы и выходы. Входы и выходы алглоблока условно группируются в подузлах **Входы** и **Выходы** соответственно.

Ветвь дерева, олицетворяющая конкретный выход алглоблока начинается с пиктограммы, условно показывающей тип выхода:

- **Bl** - логический;
- **It** - целый;
- **Vk** - векторный;
- - дата и время;
- **Fl** - действительный;
- **St** - строковый;
- **Dt** - контейнер данных.

Если вход не связан, далее следует его название и значение константы. Если же вход связан, то за пиктограммой типа вывода появляется пиктограмма типа связи (- глобальные связи, - локальные связи). Далее следует описание связи, которое строится по следующим правилам:

[имя задачи ->]Alg # ->вывод ## [.[!]###], где:

- **#** - локальный номер алглоблока в задаче;
- **##** - номер вывода алглоблока;
- **###** - номер бита;
- **!** - знак инверсии дискретного значения.

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

Инспектор алглоблоков обеспечивает:

- редактирование значения на входе алглоблока;
- переход по связи;
- удаление связи;
- изменение связи;
- удаление алглоблоков;
- переход между алглоблоками.

3.4.2.1 Редактирование значения на входе

Для того, чтобы отредактировать значение константы на входе алглоблока следует щёлкнуть левой кнопкой мыши по ветви нужного входа. В результате появится поле редактирования значения. В режиме редакции введённое в это поле значение будет запомнено. В режиме отладки в поле будет указано текущее значение константы на входе, взятое от Мезон-сервера. Внесённые изменения будут переданы Мезон-серверу.

3.4.2.2 Переход по связи

Инспектор алглоблоков позволяет упростить поиск противоположного конца связи.

Для того, чтобы перейти по связи, проведённой между алглоблоками достаточно щёлкнуть правой кнопкой мыши на связанном входе или выходе алглоблока.

В случае если щелчок был произведён по выходу алглоблока, в появившемся окне инспектора следует уточнить какая именно связь выбирается для перехода (в случае, если выход связан более чем с одним входом). По умолчанию выбирается последняя проведённая связь от данного выхода. После чего следует нажать кнопку . Если связь проведена в пределах того же модуля, редактор прокрутит рабочую область модуля до того места, где расположен связанный алглобл и установит указатель мыши на его связанном выводе. Если же связь межмодульная, то предварительно будет открыт тот модуль, в котором находится связанный алглобл.

3.4.2.3 Удаление связи

В случае, если щелчок был произведён по выходу алглоблока в появившемся окне инспектора следует уточнить какая именно связь выбирается для удаления (в случае, если выход связан более чем с одним входом). По умолчанию выбирается последняя проведённая связь от данного выхода. После чего следует нажать кнопку .

В случае, если следует удалить все связи, проведённые от данного выхода, следует выбрать выход (а не конкретную связь) и нажать кнопку удаления.

Если требуется удалить все связи, закрепленные за всеми входами или выходами, следует выбрать подузел **Входы** или **Выходы** и нажать кнопку удаления.

3.4.2.4 Изменение связи

Довольно часто в процессе редактирования техпрограммы необходимо изменять привязку связи, перебрасывая ее с одного входа алглоблока на другой. Для упрощения данной операции служит кнопка . Она становится активна только когда выбрана связь, привязанная ко входу алглоблока. При её нажатии связь отделяется от входа алглоблока, источник связи запоминается, и редактор переходит в режим проведения связи, позволяя тем самым изменять тип и число приёмников связи.

3.4.2.5 Удаление блока

Прямо из Инспектора алглоблоков можно удалить алглобл для которого он был вызван. Для этого следует выбрать пиктограмму алглоблока и нажать кнопку удаления.

3.4.2.6 Переход между алглоблоками модуля

Инспектор алглоблоков позволяет осуществлять переход между алглоблоками в пределах одного модуля. Для этих целей служит выпадающий список с названиями алглоблоков в верхней части окна инспектора . В списке перечислены все алглобки, входящие в состав текущего модуля. При выборе нового алглоблока, рабочая область модуля прокручивается так, что алглобл оказывается в её центре.

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

3.4.3 Диалог Копирование алглоблоков

Диалоговое окно (рисунок 32) появляется, когда происходит вставка алглоблоков из буфера обмена Windows.

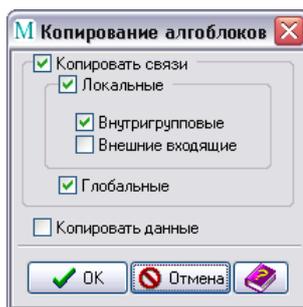


Рисунок 32 – Диалоговое окно Копирование алглоблоков

С помощью окна можно установить параметры копирования алглоблоков:

- **Копировать связи.** Опция позволяет открыть группу копирования всех видов связей между копируемыми алглоблоками. Выбор необходимых видов связей осуществляется при помощи флагов в данной группе;
- **Локальные.** Опция позволяет открыть группу копирования локальных связей между копируемыми алглоблоками. Выбор необходимых типов локальных связей осуществляется при помощи флагов в данной группе;
- **Внутригрупповые.** Опция позволяет копировать локальные связи между копируемыми алглоблоками;
- **Внешние входящие.** Опция позволяет копировать все связи, которые являются внешними связями со входами алглоблоков в копируемой группе. Следует отметить, что в случае выбора этого флага и копирования алглоблоков из одной задачи в другую внешние входящие связи для копируемой группы станут глобальными (то есть межзадачными);
- **Глобальные.** Опция позволяет копировать межзадачные связи со входами алглоблоков копируемой группы. Аналогично предыдущему случаю внешние глобальные связи могут стать локальными, если связанные в разных задачах алглобокки при копировании попадают в одну задачу;
- **Копировать данные.** Опция позволяет копировать значения констант на всех (в том числе и на скрытых) входах алглоблоков копируемой группы.

3.4.4 Диалог Управление модулями

В диалоговом окне (рисунок 33) отображаются все модули проекта, а так же сам проект.

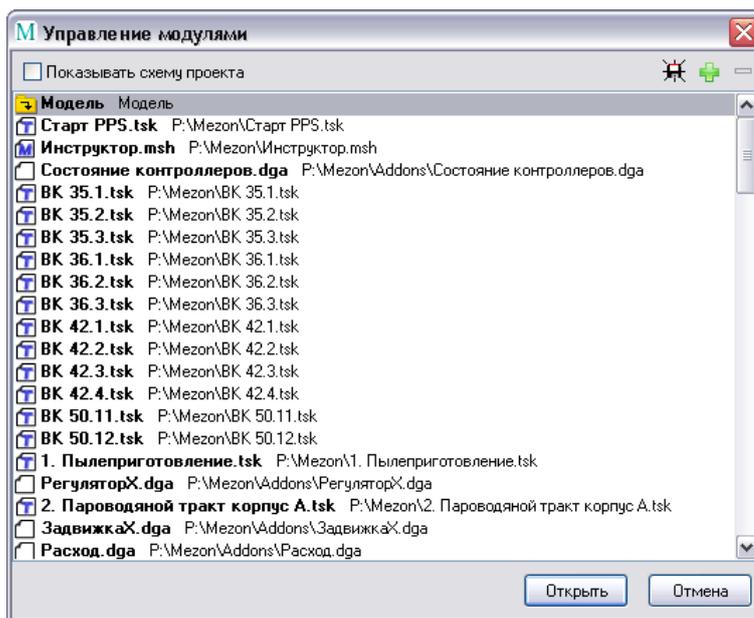


Рисунок 33 – Диалоговое окно Управление модулями

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

Каждому типу модуля соответствует своя пиктограмма:

-  - модуль проекта;
-  - модуль задачи;
-  - модуль мнемосхемы;
-  - модуль составного алгоритма.

Полужирным шрифтом выделяется имя файла модуля, обычным шрифтом – полный путь к этому файлу. Для открытия нескольких модулей, следует выделить их в списке и нажать **Открыть**.

С помощью данного диалога можно выполнить следующие действия:

- открыть проект в виде модуля;
- открыть модули, входящие в состав проекта;
- изменить имя модуля;
- добавить новый модуль;
- удалить имеющийся модуль.

3.4.4.1 Открыть проект в виде модуля

Модуль проекта всегда располагается первым в списке модулей и может быть открыт в упрощённом виде, либо с полной схемой проекта.

Для открытия в упрощённом виде, следует убрать флаг **Показывать схему проекта**. Для открытия с полной схемой – флаг следует установить.

Если проект открывается с полной схемой, то все задачи и мнемосхемы, входящие в состав проекта, будут изображаться с дополнительными входами и выходами. Состав таких входов формируется редактором автоматически и зависит от межмодульных связей. С левой стороны будет отображаться иерархическое дерево модулей (см. рисунок 16).

3.4.4.2 Открыть модули, входящие в состав проекта

Для открытия одного или нескольких модулей из списка, следует выделить их в списке модулей и нажать на кнопку **Открыть**.

3.4.4.3 Изменить имя модуля

Для изменения имени модуля, следует выбрать любой модуль из списка и нажать кнопку . Действие этой кнопки полностью повторяет действие пункта главного меню **Файл / Сохранить как**.

3.4.4.4 Добавить новый модуль

Для добавления модуля в проект, следует нажать кнопку . Откроется диалог выбора модуля. Выбранный с его помощью модуль попадёт в состав текущего проекта. При этом список модулей будет обновлён, так как в нём появится добавленный модуль, а также новые модули, входящие в его состав.

3.4.4.5 Удалить имеющийся модуль

Позволяет удалять модули задач и мнемосхем из проекта. Для этого, следует выделить нужную задачу или мнемосхему и нажать кнопку . Можно удалить только одну задачу или мнемосхему за один раз. Модуль проекта или составного алгоритма удалить данного списка нельзя. Для удаления проекта, следует воспользоваться диалогом **Проект**. Составные алгоритмы следует удалять из соответствующих модулей проекта.

3.4.1 Диалог Создание модуля проекта

Диалоговое окно (рисунок 34) состоит из двух частей. В правой части можно выбрать способ создания модулей:

- **Стандартные**. Позволяет создать стандартные модули (модуль задачи, мнемосхемы, составного алгоритма);

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

- **Шаблоны.** Позволяет создавать модули на основе имеющегося шаблона. В качестве шаблона может быть взят один из составных алгоритмов, входящих в библиотеку алгоритмов или один из модулей текущего проекта.

Выбрав нужный способ создания и тип модуля, и нажав кнопку **Ok**, можно сохранить новый модуль в одном из доступных хранилищ. При этом, если модуль является задачей или мнемосхемой – он будет добавлен в проект и автоматически открыт для редактирования.

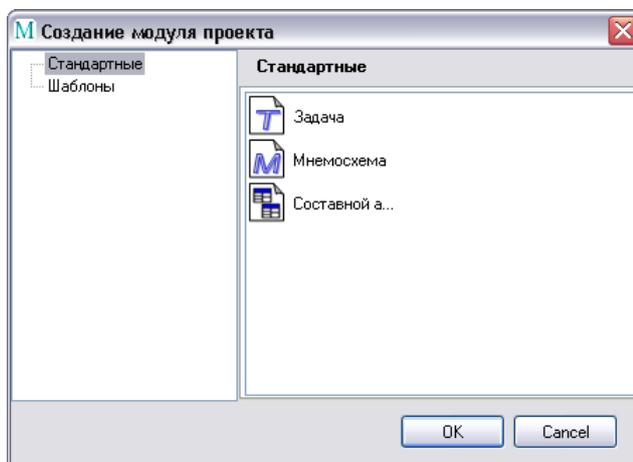


Рисунок 34- Диалоговое окно Создани модуля проекта

3.4.2 Диалог Настройка хранилища

С помощью этого диалогового окна можно настроить параметры **Хранилища исходных кодов** проекта Мезона (рисунок 35).

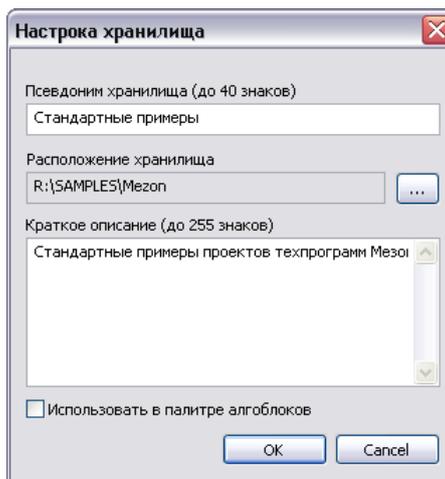


Рисунок 35 – Диалоговое окно Настройка хранилища

В строке **Псевдоним хранилища** указывается символическое название хранилища, длиной не более 40 знаков.

Для указания физического расположения хранилища, следует щёлкнуть на кнопке  и в появившемся диалоге выбора папки, указать место расположения хранилища. Таким местом может быть как локальная папка на компьютере, так и общедоступный сетевой ресурс (сетевая папка, расположенная в общем доступе).

В поле **Краткое описание** можно добавить пояснение, относящееся к типу файлов, расположенных в данном хранилище.

Флаг **Использовать в палитре компонентов** позволяет Мезон-редактору, добавлять составные алгоритмы, расположенные на первом уровне указанного хранилища (в корне хранилища), в библиотеке алгоритмов. При этом такие составные алгоблоки попадут в палитру алгоритмов и их можно использовать наравне со стандартными алгоритмами из библиотеки Мезона.

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

3.4.3 Диалог Выбор алгоритма

С помощью этого диалогового окна (рисунок 36) можно добавлять алгоритмы в текущий открытый модуль.

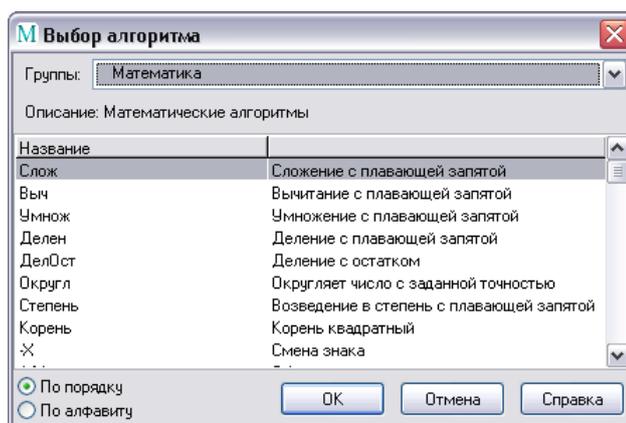


Рисунок 36- Диалоговое окно Выбор алгоритма

Выпадающий список **Группы** позволяет выбрать нужную группу алгоритмов (например, **Математика**) или все алгоритмы, входящие в состав библиотеки алгоритмов. В последнем случае, следует выбрать виртуальную группу **Все** (расположена первой в списке групп).

В строке **Описание** - приводится краткое описание выбранной группы.

Далее следует таблица алгоритмов, относящихся к выбранной группе. В левой колонке отображаются названия алгоритмов, в правой колонке приводится их краткое описание. Можно отсортировать алгоритмы в таблице по их порядку, в котором они предоставляются плагином, или в алфавитном порядке.

Для добавления алгоритма в модуль, можно воспользоваться одним из следующих способов:

- нажать на выбранном алгоритме основной кнопкой мыши и, удерживая кнопку нажатой, перетащить в нужное место рабочей области модуля. Для установки алгоритма следует отпустить кнопку. Для отмены добавления алгоритма, следует, не отпуская основную кнопку мыши, щёлкнуть дополнительной на свободном месте рабочей области модуля. Данный способ позволяет добавить несколько алгоритмов, не закрывая диалоговое окно;
- выделить выбранный алгоритм и нажать кнопку **ОК**. Изображение выбранного алгоритма «прилипнет» к курсору мыши и будет перемещаться вслед за ним. Для того, чтобы расположить его в нужном месте, щёлкните основной кнопкой мыши на свободном участке рабочей области модуля. Для отмены добавления, щёлкните дополнительной кнопкой мыши.

Если, прежде чем будет добавлен в модуль, алгоритм требует предварительной настройки, то отобразится мастер предварительной настройки алгоритма. После настройки алгоритм перейдёт в режим добавления и его можно добавить описанными способами.

3.4.4 Диалог Выделение алгоблоков, Выделение комментариев

Диалоговое окно **Выделение алгоблоков** показано на рисунке 37, а **Выделение комментариев** - на рисунке 38.

Оба диалога похожи по принципу работы. Если до открытия данных диалогов некоторые объекты соответствующих типов были выделены, они будут помечены флагами. Для того, чтобы выделить нужные объекты, следует установить флаг рядом с объектом в списке, иначе – сбросить флаг. Для выделения всех объектов, следует нажать кнопку для инвертирования выделения - для сброса - . Имеется возможность выделения нескольких однотипных объектов. Для этого следует щёлкнуть по выпадающему списку. В нем будут перечислены все имеющиеся типы объектов, выбор типа приводит к установке флагов у всех объектов данного типа.

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

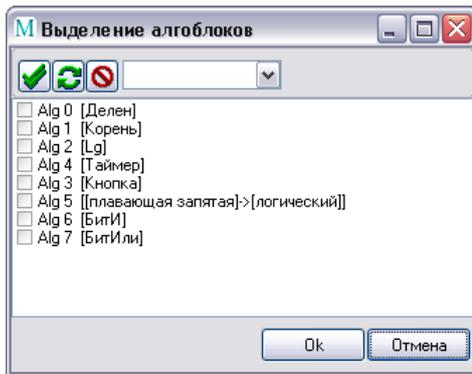


Рисунок 37 - Окно Выделение алглоблоков

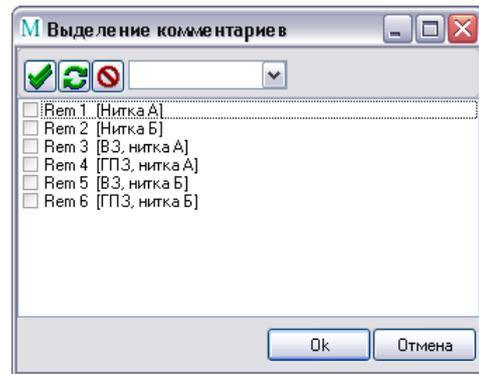


Рисунок 38 - Окно Выделение комментариев

3.4.5 Диалог Перейти к алглоблоку, Перейти к комментарию

Диалоговое окно **Перейти к алглоблоку** показано на рисунке 39, а **Перейти к комментарию** - на рисунке 40.

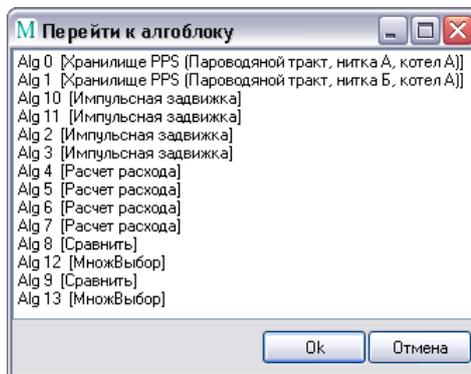


Рисунок 39 – Окно Перейти к алглоблоку

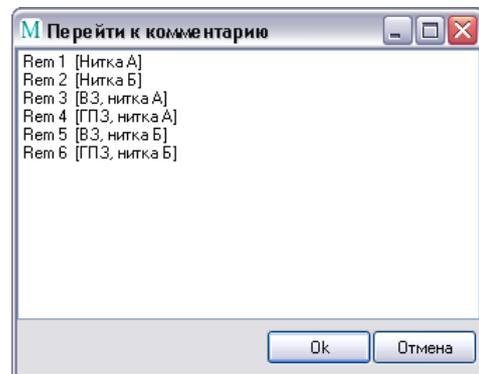


Рисунок 40 Окно Перейти к комментарию-

Диалоги похожи по принципу работы. Каждый позволяет перейти к соответствующему объекту. Для перехода следует выделить объект в списке и нажать **Ок**. Рабочая область модуля будет прокручена так, чтобы выбранный объект по возможности расположился в центре окна.

3.4.6 Диалог «Точки останова»

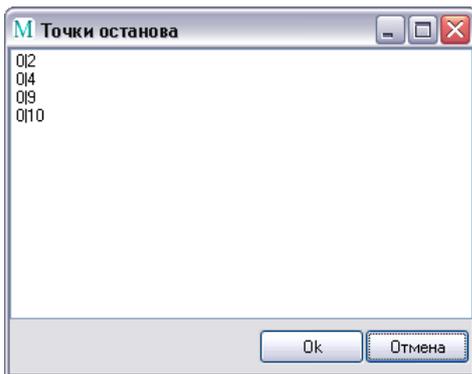


Рисунок 41 – Окно выбора одной точки останова

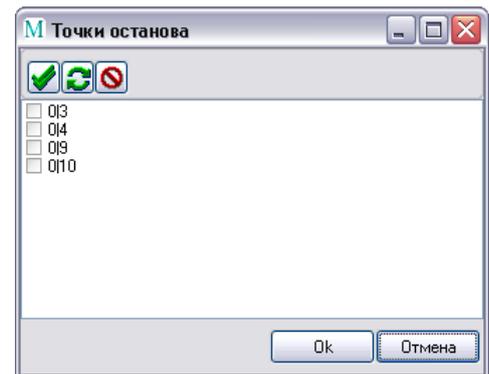


Рисунок 42 – Окно выбора одной или нескольких точек останова

Первый тип диалога (рисунок 41) позволяет выбрать одну из точек останова, установленных во время отладки техпрограммы на сервере и перейти к тому алглоблоку, на котором она установлена.

Второй (рисунок 42) позволяет отметить одну или несколько точек останова и удалить их. Кнопка служит для выбора всех точек останова, - для инвертирования выделения, - для сброса. Для подтверждения действия следует нажать кнопку **Ок**, для отмены – **Отмена**.

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

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

3.4.7 Диалог Изменить порядок алглоблоков

В диалоговом окне (рисунок 43) можно выделить один или несколько алглоблоков и изменить их порядок по отношению к невыделенным алглоблокам.

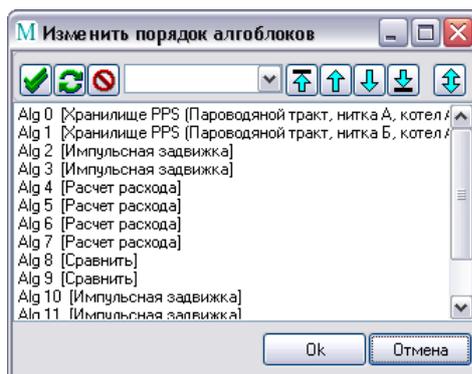


Рисунок 43 – Окно Изменить порядок алглоблоков

Выделять алглоблоки можно произвольным образом, при этом при перемещении такой группы «заякор» между ними будет оставаться неизменным. Для того, чтобы выделить нужные алглоблоки, следует щелкать по нужным строкам мышью, удерживая кнопку **Ctrl**, для сброса выделения – следует щёлкнуть на строке повторно. Для выделения всех алглоблоков, следует нажать кнопку

для инвертирования выделения - , для сброса - . Имеется возможность выделения нескольких однотипных алглоблоков. Для этого следует щёлкнуть по выпадающему списку. В нем будут перечислены все имеющиеся типы алглоблоков, выбор типа приводит к выделению строк всех алглоблоков данного типа.

Для изменения порядка алглоблоков следует воспользоваться кнопками с изображением стрелок:

- Сдвигает выделенные алглоблоки вверх до тех пор, пока первый выделенный алглоблок не достигнет верхнего края;
- Сдвигает выделенные алглоблоки на одну позицию вверх;
- Сдвигает выделенные алглоблоки на одну позицию вниз;
- Сдвигает выделенные алглоблоки вниз до тех пор, пока последний выделенный алглоблок не достигнет нижнего края;
- Взаимно меняет позицию двух выделенных алглоблоков. Кнопка доступна, если выделено не более 2-х алглоблоков.

Для подтверждения смены позиции следует нажать кнопку **Ok**, иначе – **Отмена**.

Процесс смены может занять большое время (особенно, если происходит смена порядка выполнения задач в проекте) и зависит от числа объектов и их взаимного перемещения.

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

3.4.8 Диалог Добавить алгоблок, диалог Добавить задачу в техпрограмму

Диалоги, окна которых показаны на рисунках 44 и 45, позволяют добавлять объекты в исходные модули техпрограммы Мезона.

Рисунок 44 – Окно Добавить алгоблок

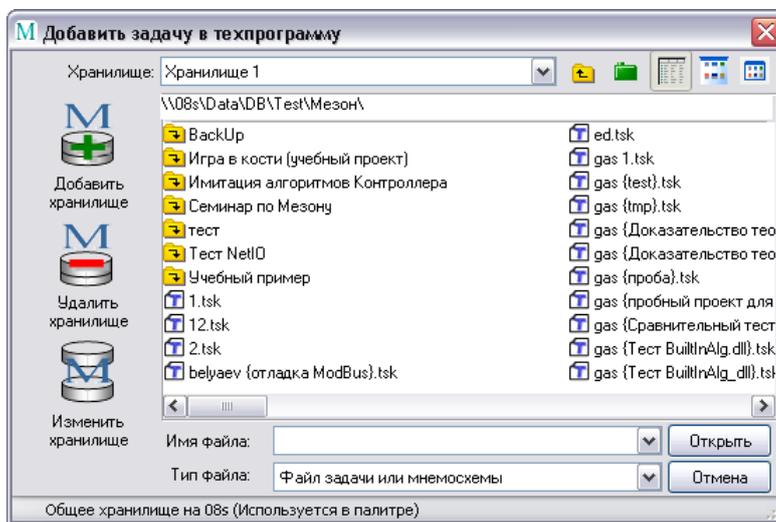


Рисунок 45 – Окно Добавить задачу в техпрограмму

Для того, чтобы выбрать объект для добавления, следует выбрать хранилище в котором он расположен. Для этого служит выпадающий список **Хранилище**. Внутри хранилища располагаются файлы и папки. Каждому типу файла соответствует своя пиктограмма:

-  - вложенная папка хранилища;
-  - файл задачи;
-  - файл мнемосхемы;
-  - файл составного алгоритма.

Если нужный файл располагается во вложенной папке, следует дважды щёлкнуть на изображении нужной папки, в результате чего она будет раскрыта. Полный физический путь отображается в строке, расположенной под панелью навигации (на рисунках это пути \\08s\Data\DB\Test\Мезон\). Для выбора нужного файла нужно его выделить и нажать кнопку **Открыть** или дважды щёлкнуть на нем. Для отмены открытия файла – следует щёлкнуть на кнопке **Отмена**. Для сортировки файлов по их типам следует воспользоваться выпадающим списком **Тип файла**. При этом в окне будут отображаться только файлы выбранного типа. Для того. Чтобы подняться из текущего места на один уровень вверх, следует воспользоваться кнопкой  на панели навигации.

Для изменения режима отображения списка файлов, существует три кнопки:

-  Отображает файлы в виде многоколоночного списка;
-  Отображает файлы в виде больших иконок;
-  Отображает файлы в виде списка, представленного в одной колонке;

Кнопка  служит для создания новой вложенной папки.

Кнопки , , , расположенные в левой части диалога, позволяют управлять хранилищами и соответствуют пунктам меню **Файл / Хранилища / Добавить**, **Файл / Хранилища / Удалить** и **Файл / Хранилища / Изменить**, соответственно. В строке состояния диалога выводится примечание к текущему хранилищу.

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

3.4.9 Диалог Управление окнами

Окно диалога (рисунок 46) позволяет управлять открытыми модулями (окнами).

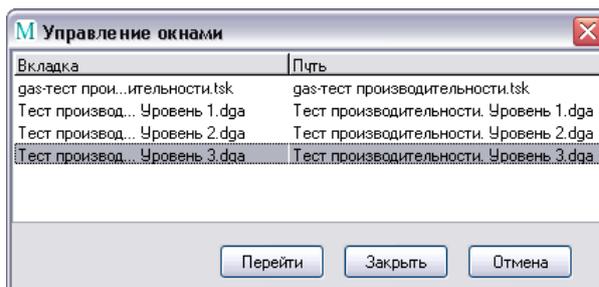


Рисунок 46 – Окно Управление окнами

В первой колонке таблицы приводится название вкладки (для экономии места оно может быть сокращённым названием открытого модуля). Во второй колонке – полное название модуля с контекстным окном его открытия, если модуль был открыт контекстно.

Для того, чтобы сделать модуль активным, следует выделить его в таблице и нажать кнопку **Перейти**. Модуль будет открыт в Мезон-редактор. Для закрытия модуля можно воспользоваться кнопкой **Закреть**.

3.4.10 Диалог Обзор плагинов

Окно диалога позволяет просмотреть библиотеку алгоритмов на уровне составляющих её модулей – плагинов.

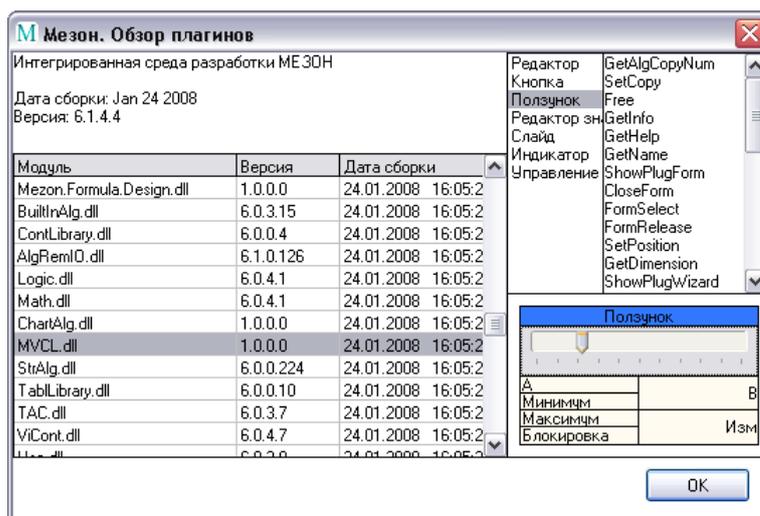


Рисунок 47 – Окно Обзор плагинов

В таблице отображается список модулей (рисунок 47), их версия и время сборки. Если выделить строку в таблице – в правой части появится список алгоритмов, реализуемый модулем, а также список функций, которые он экспортирует. Если щёлкнуть на названии любого алгоритма в нижнем правом углу диалога появится его изображение.

3.4.11 Диалог Параметры

Окно предназначено для настройки основных параметров редактора. Следует учитывать, что некоторые параметры могут настраиваться как для редактора в целом, так и для текущего открытого проекта, если вызвать данный диалог в момент, когда открыт какой-либо проект Мезона.

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

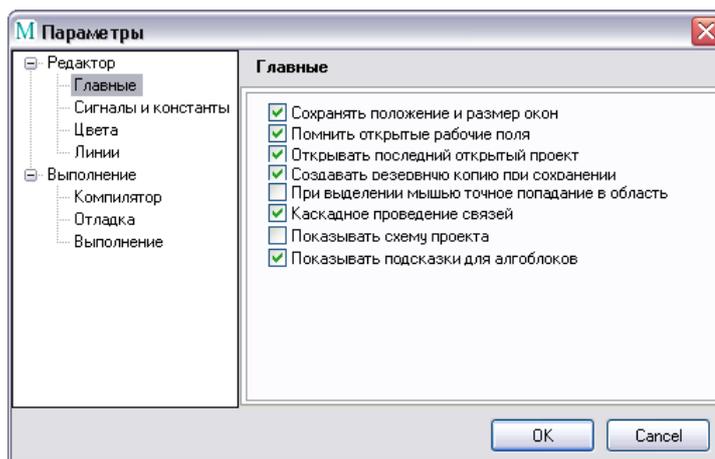


Рисунок 48 – Окно Параметры

В левой части диалога располагается дерево настроек (рисунок 48), в правой – отображается страница с соответствующими настройками.

3.4.11.1 Редактор

Дерево **Редактор** содержит настройки параметров редактора. Настройки всегда применяются к редактору в целом и не зависят от текущего проекта Квинта.

3.4.11.1.1 Главные

Ветвь редактора **Главные** содержит следующие параметры:

- **Сохранять положение и размер окон** – если опция установлена, то размер и положение всех окон редактора будут запоминаться, и в следующий раз окно будет появляться в том же виде, в котором было закрыто в последний раз;
- **Помнить открытые рабочие поля** – если опция установлена – при открытии проекта будут восстановлены все модули, открытые в момент последней работы с проектом;
- **Открывать последний открытый проект** – если опция установлена, то при открытии редактора будет так же открыт проект, с которым работали в последний раз;
- **Создавать резервную копию при сохранении** – если опция установлена, то до начала сохранения изменений в модуле будет сохраняться его резервная копия. Резервная копия сохраняется в той же папке, в которой находится исходный файл, но в поддиректории **Backup**. Резервной копии модуля присваивается имя, формируемое по следующему шаблону: `<имя_пользователя><дата>_<время><имя_исходного_модуля>`;
- **При выделении мышью точное попадание в область** – если опция установлена – то при выделении объектов мышью они будут выделены, только если целиком попадут в прямоугольник выделения;
- **Каскадное проведение связей** – если опция установлена, то при проведении связи от выхода алгоблока к входу другого алгоблока, режим выделения связи не будет сброшен и связь от данного выхода можно провести с еще одним входом. Для выхода из режима проведения связи, следует щелкнуть мышью на любом свободном месте рабочей области модуля;
- **Показывать схему проекта** – если опция установлена, проект будет открываться в виде полной схемы с деревом задач (подробнее см. пункт «Открыть проект в виде модуля»);
- **Показывать подсказки для алгоблоков** – если опция установлена, то при проведении курсора мыши над заголовком или выводами алгоблока будет появляться окно подсказки, выводящее полную информацию о соответствующем объекте.

3.4.11.1.2 Сигналы и константы

Ветвь редактора **Сигналы и константы** содержит следующие параметры:

- **Отображать векторы в двоичном виде** – после установки данной опции значения сигналов с типом **векторный** отображается в виде набора бит с возможностью изменения каждого бита в отдельности;

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

- **Отображать целые значения в шестнадцатеричной форме** – после установки данной опции значения сигналов с типом **целый** и **векторный** (если не установлена опция **отображать векторы в двоичном виде**);
- **Отображать логические сигналы как Да и Нет** – после установки данной опции значения сигналов *0* и *1*, типа **логический** будут отображаться как **Да** и **Нет**, соответственно;
- **Отображать модельное время в миллисекундах** – после установки данной опции значения сигналов, типа **модельное время** будут отображаться в виде **xx** мс или **~xx: (yx)** мс для абсолютного и относительного времени соответственно, где **xx** – заданное время в мс, **yx** – базовое время цикла в мс.;
- **Сокращать строки до ... знаков** – при установке данной опции становится доступным цифровое поле, в котором можно задать число знаков при отображении значений сигналов типа **строковые**;
- **Отображать дроби с точностью ... знака** – при установке данной опции становится доступным цифровое поле, в котором можно задать число знаков после запятой (с округлением) при отображении значений сигналов типа **действительный**.

3.4.11.1.3 Цвета

Ветвь редактора **Цвета** содержит следующие параметры:

- **Цвет надписи** – при нажатии на данном пункте, появляется диалог выбора цвета, который позволяет задать цвет надписей в окне Мезон-редактора;
- **Цвет надписи при выделении** - при нажатии на данном пункте, появляется диалог выбора цвета, который позволяет задать цвет интерактивных надписей при выделении их с помощью мыши. Интерактивными являются все надписи констант на входах алгоритмов;
- **Цвет отладочной информации** – позволяет при помощи диалога выбора цвета, задать цвет информации, выводимой во время отладки проекта на входах и выходах алгоблоков;
- **Цвет линий** – при помощи диалога выбора цвета можно задать цвет линий связи между алгоблоками;
- **Цвет линий обратной связи** – при помощи диалога выбора цвета можно задать цвет линий обратных связей между алгоблоками. Обратными связями считаются такие связи между алгоблоками, когда выход алгоблока с номером N, в модуле M связан с входом с входом того же алгоблока, либо входом алгоблока с номером N-i в модуле M, либо с входом любого алгоблока в модуле M-j;
- **Цвет линий при наведении** – при помощи диалога выбора цвета можно задать цвет линий при наведении указателя мыши на связанные этой линией вход и выход алгоблоков.

3.4.11.1.4 Линии

Ветвь редактора **Линии** содержит следующие параметры:

- **Толщина линий связи** – позволяет задать толщину линий связи в пикселях;
- **Отступ от объекта** – позволяет задать величину отступа линий связи от алгоблоков, источников или приемников связи;
- **Размежевание** – позволяет задать величину размежевания линий связи, исходящих с выходов или приходящих на входы алгоблоков. При размежевании = 1 – линии связи с входов и выходов одного алгоблока собираются в шины. При размежевании = 0 – линии связи с входов и выходов одного алгоблока сливаются в одну;
- **Упрощенная отрисовка** – установка данной опции позволяет не рисовать связи между алгоблоками, расположенными вне видимой рабочей области модуля;
- **Одиночная линия** – установка опции позволяет отображать связи в виде сплошной одиночной линии;
- **Пунктирная линия** – установка опции позволяет отображать связи в виде пунктирной одиночной линии (только если **толщина линий связи** = 1). Годится для отрисовки линий связи в модулях мнемосхем, т.к. не дает сливаться линии с темным фоном мнемосхемы;
- **Двойная линия** – установка опции позволяет отображать связи в виде двойной линии заданной толщины. Предназначена для тех же целей, что и пунктирная линия, с той разницей, что в модулях задач, проектов и составных алгоблоков линия кажется одинарной, а в модулях мнемосхем – вторая светлая линия не дает слиться с фоном;

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

3.4.11.2 Выпoнeниe

Настройка параметров выполнения техпрограммы на Мезон-сервере. Некоторые настройки всегда к редактору в целом, другие к текущему проекту Квинта (будет сказано при описании соответствующей группы настроек).

3.4.11.2.1 Компилятор

Ветвь выполнения **Компилятор** содержит следующие параметры:

- **Использовать встроенный компилятор** – установка опции позволяет выбрать внешний компилятор техпрограмм Мезона;
- **Примечание** - Данная опция не реализована.

3.4.11.2.2 Отладка

Ветвь выполнения **Отладка** содержит следующие параметры:

- **Отлаживать после компиляции** – позволяет переводить Мезон-редактор в режим отладки сразу после успешного окончания компиляции;
- **Закрывать контекстно открытые составные алгоблоки** – опция позволяет автоматически закрывать все экземпляры составного алгоритма, открытые во время отладки контекстно (например, вследствие перехода внутрь отлаживаемого составного алгоблока).

3.4.11.2.3 Выпoнeниe

Данные настройки зависят от текущего проекта Квинта.

- **Тайм-аут команды** – позволяет задать время ожидания выполнения команды, посланной Мезон-серверу, в секундах. Во время ожидания в строке состояния появляется индикатор обратного отсчета времени выполнения, с помощью которого ожидание можно прекратить досрочно;
- **Всегда создавать новый сервер** – установка опции позволяет в начале каждого сеанса отладки автоматически создавать новый экземпляр техпрограммы на выбранном Мезон-сервере. Если данная опция не установлена и на Мезон-сервере уже был запущен экземпляр техпрограммы появится диалог позволяющий подключиться к нему или создать новый экземпляр. Опция удобна в случае отладки отдельных частей техпрограммы различными пользователями. В таком случае каждый пользователь сможет отлаживать свою «независимую» техпрограмму, не мешая при этом остальным. Если опция установлена, рекомендуется устанавливать опцию **Всегда разрушать созданный сервер**, чтобы на сервере выполнения не накапливались созданные и не разрушенные техпрограммы;
- **Всегда разрушать созданный сервер** – установка опции позволяет автоматически разрушать созданный сервер. Эта опция используется совместно с опцией **Всегда создавать новый сервер**;
- **Режим блокировки** – позволяет установить режим блокировки создаваемого экземпляра техпрограммы, для ограничения возможностей управления им с других клиентов (Мезон-редакторов). Всего существует 4 режима:
 - а) **Нет.** Режим блокировки техпрограммы на сервере отсутствует. Созданной техпрограммой можно управлять (ставить на паузу, останавливать, отлаживать и т.п.) с любого клиента (Мезон-редактора);
 - б) **Управление.** Режим блокировки техпрограммы, при котором остановка и перезапуск экземпляра техпрограммы с любого другого клиента (кроме того, с которого он был создан) становятся невозможными. При этом остаётся возможность отладки;
 - в) **Управление и отладка.** Режим блокировки, при котором невозможно управлять и отлаживать экземпляр техпрограммы с любого клиента, кроме того, с которого она была создана;
 - г) **Монопольная.** Режим блокировки, при котором к техпрограмме невозможно подключиться всем клиентам, кроме того, который её создал;

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

3.6.1 Компиляция, загрузка и запуск техпрограммы

См. 2.7. Учебный пример. Шаг 7. Компилирование и запуск проекта.

3.6.2 Отладка по шагам

См. 2.10. Учебный пример. Шаг 9. Трассировка и отладка.

3.6.3 Отладка по точкам останова

См. 2.10. Учебный пример Шаг 9. Трассировка и отладка

Инв. № подл.	Подп. и дата	Взаим. инв. №	Инв. №	Подп. и дата	ПФДИ.421457.009 И3.6					Лист
					Изм	Лист	№ докум	Подп.	Дата	53

4 Часто возникающие вопросы

4.1 Вопрос 1. Есть готовая dll, в которой реализована некая функция. Как максимально быстро начать использовать её в Мезоне?

Для того, чтобы в Мезоне использовать функцию, реализованную в некоторой произвольной dll можно воспользоваться алгоритмом **Формула**. Допустим у вас имеется файл динамической библиотеки (для примера возьмем библиотеку, входящую в состав Квинта – MezonUtils.dll), скомпилированной в виде native кода (например, в Visual C++, Borland C++Builder или Delphi) в которой реализована функция получения хэш-кода (например, CRC32) массива произвольной длины, возвращающая результат в виде целого числа без знака. Для примера возьмем функцию, объявленную как:

```
(C++) unsigned int __stdcall mCalcHashCode(char *massive, int mass_length);
```

Добавляем в задачу, в которой мы хотели бы использовать данную функцию, алгоритм «Формула» (вкладка «Общие»), вызываем мастер настройки. На вкладке «Дополнительно» подключаем dll и объявляем функцию в синтаксисе языка C#:

```
[System.Runtime.InteropServices.DllImport("MezonUtils.dll",  
CallingConvention=System.Runtime.InteropServices.CallingConvention.StdCall,  
Charset = System.Runtime.InteropServices.CharSet.Ansi)]  
public static extern uint mCalcHashCode(char[] massive, int mass_length);
```

Будем использовать данную функцию для подсчета хэш-кода строки, передаваемой на вход алгоритма формула. Для этого добавим вход **str**, строкового типа и выход **hc**, целочисленного типа – это искомый хэш-код. На вкладке **Вычисление** напишем:

```
char[] tmp = str.ToCharArray();  
hc = mCalcHashCode(tmp, tmp.Length);
```

Компилируем проект и убеждаемся, что все работает так, как требовалось. В более сложных случаях, когда в функцию необходимо передавать структуры или интерфейсные объекты, их следует предварительно объявить на вкладке **Дополнительно**.

4.2 Вопрос 2. Как происходит преобразование сигналов различных типов?

Мезон-редактор поддерживает связывание выводов алгоблоков с различными типами сигналов. При проведении некоторых связей накладываются ограничения (например, однонаправленное связывание), могут появляться диалоги, позволяющие пользователю выбрать дополнительные настройки преобразования одного типа сигнала к другому, либо могут возникать предупреждения о побочных эффектах. Ниже в таблице показаны все возможные связи различных типов сигналов, перечислены возможные диалоги связывания, указаны ограничения.

В таблице 1 условно показаны возможности конфигурирования различных типов входов и выходов алгоблоков.

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

Таблица 1 – Конфигурирования различных типов входов и выходов алгоблоков

		Входы										
		логич.	целый	вектор.	модел. время	действ.	строковый	контейнер данных	дата и время	длит.	событийный	вирт.
Выходы	логич.	<->	[<->]	[1]	--	[<->]	[->]	<->	--	--	[->]	<->
	целый	[<->]	<->	<->	<->	[<->]	[->]	<->	--	--	--	<->
	вектор.	[1]	<->	<->	--	[<->]	[->]	<->	--	--	--	<->
	модел. время	--	--	--	<->	--	[->]	<->	--	--	--	<->
	действ.	[<->]	[<->]	--	--	<->	[->]	<->	--	<->	--	<->
	строковый	--	--	--	--	--	<->	<->	--	--	--	<->
	контейнер данных	--	--	--	--	--	<->	<->	--	--	--	<->
	дата и время	--	--	--	--	->	[2]	<->	<->	--	--	<->
	длит.	--	--	--	--	<->	[3]	<->	--	<->	--	<->
	событийный	[<->]*	--	--	--	--	--	--	--	--	<->	<->
	неопред.	<->	<->	<->	<->	<->	<->	<->	<->	<->	<->	<->

В таблице использованы следующие обозначения:

<-> - непосредственное соединение (преобразование не происходит);

-> непосредственное однонаправленное связывание (преобразование не происходит, обратное связывание невозможно или осуществляется иначе);

[<->] - скрытое преобразование без настройки (в этом случае, на этапе компиляции будет подставлен алгоблок, преобразующий один тип сигнала в другой);

[->] - скрытое однонаправленное преобразование без настройки (обратное связывание невозможно или осуществляется иначе, на этапе компиляции будет подставлен алгоблок, преобразующий один тип сигнала в другой);

[X] - настраиваемое скрытое преобразование (при связывании сигналов появляется диалог, позволяющий дополнительно настроить параметры преобразования);

-- - связь типов невозможна.

4.2.1 Настраиваемое скрытое преобразование [1]

Предлагается выбор одного из вариантов:

- логический->векторный как вектор, сложив все биты по ИЛИ;
- логический->векторный один из битов вектора;
- логический->векторный один из битов вектора с инверсией;
- векторный->логический как вектор, распространив на все биты;
- векторный->логический как один из битов вектора;
- векторный->логический как один из битов вектора с инверсией.

4.2.2 Настраиваемое скрытое преобразование [2]

Предлагается настройка отображения даты и времени в следующих сочетаниях:

- выбор частей отображаемой даты (день, месяц, год);
- выбор последовательности отображения частей даты;
- выбор отображения месяца (слово/число);
- выбор разделителя для отдельных частей даты;
- выбор отображения времени;
- выбор отображения секунд и миллисекунд;
- выбор отображения подписей к отдельным частям времени (ч., м., с., мс.).

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

4.2.3 Настраиваемое скрытое преобразование [3]

Предлагается настройка отображения интервала времени в следующих сочетаниях:

- выбор отображения или скрытия незначащих цифр;
- выбор отображения миллисекунд;
- выбор отображения подписей к отдельным частям длительности (сут., ч., м., с., мс.).

Примечание – При преобразовании событийного выхода → логический входной сигнал выдаётся предупреждение **Связь типов <Логический> и <Событийный> может вносить побочные эффекты!**. Это связано с тем, что если событие передаётся как логическое значение в пределах одного модуля, то все работает нормально. Но, если оно будет передаваться в другой модуль (задачу, составной алгоблок и т.д.) тогда могут возникнуть побочные эффекты в виде пропуска значений. Таким образом это потенциальная ошибка, о которой предупреждает как редактор, так и компилятор (он тоже выводит это предупреждение и показывает какой именно алгоблок тому причиной).

4.3 Вопрос 3. Каков порядок выполнения алгоблоков и как на него влиять?

В техпрограмме над каждым алгоблоком показан его порядковый номер в пределах модуля. Этот номер автоматически присваивается любому алгоблоку при добавлении его в модуль. Порядковые номера алгоблоков указывают порядок их выполнения в техпрограмме. Для изменения порядкового номера следует использовать меню **Правка – Изменить порядок**.

Каждой задаче, включённой в проект Мезона так же присваивается порядковый номер, однако его влияние на выполнение задач менее очевидно в силу того, что в общем случае задачи могут работать асинхронно, с разными временами циклов, а то и вовсе исключаться из цикла выполнения (например, по входу **Разрешение**). Тем не менее, если все задачи в проекте выполняются со значением **Условие**, равным 1, то вначале будут выполнены все алгоблоки той задачи, у которой порядковый номер меньше, а затем все алгоблоки той, у которой он больше. Аналогично и для составных алгоблоков, работающих в составе задачи или других составных алгоблоков.

Важное замечание – Следует уделять большое внимание порядку алгоблоков в техпрограмме, так как он может существенно повлиять на работу программы.

Рассмотрим, к примеру, два варианта части техпрограммы, показанные на рисунках 49 и 50. В первом варианте алгоблок смены знака выполняется раньше алгоблока сложения, поэтому на входе последнего всегда будут два числа с разными знаками, а на выходе всегда будет ноль.

Во втором варианте алгоблок смены знака выполняется после алгоблока сложения, поэтому на входе последнего будут разность текущего и предыдущего чисел. При подаче на вход синусоидального сигнала получаем на выходе отличный от нуля сигнал.

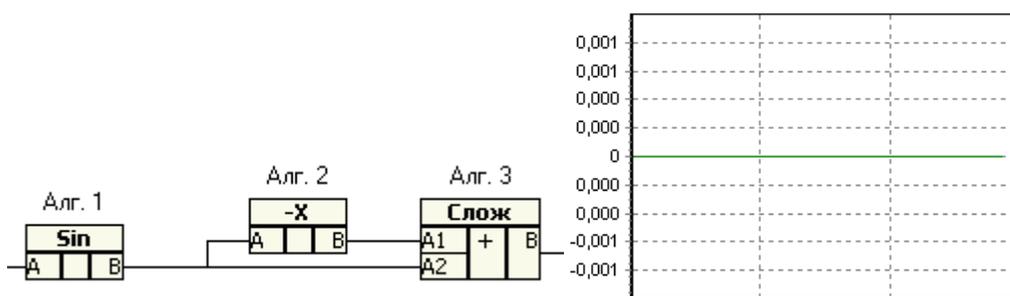


Рисунок 49 – Первый вариант части техпрограммы

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

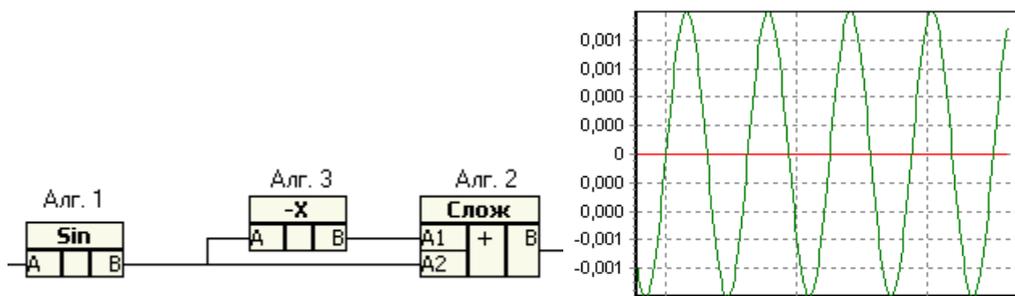


Рисунок 50 – Второй вариант части техпрограммы

Если алгоблок удаляется из модуля, в порядковых номерах образуются «дырки». Они ни на что не влияют, т.к. на порядок выполнения одного алгоблока оказывает влияние не конкретный порядковый номер, а знак разности с другим порядковым номером. Если эта разность положительна, то алгоблок с вычитаемым порядковым номером будет выполняться позже алгоблока с уменьшаемым порядковым номером, если отрицательна – будет выполняться раньше. Тем не менее, среда редактора старается заполнять «дырки» в первую очередь. Поэтому при добавлении нового алгоблока в модуль, имеющий «дырки» в порядковой нумерации, ему будет присвоен один из пропущенных порядковых номеров. Так, как это не всегда то, что хотелось бы разработчику, алгоблок следует переместить вверх или вниз по массиву порядковых номеров, так, чтобы он занял нужное место.

Ликвидировать «дырки» в модуле можно двумя способами:

- выделить все алгоблоки модуля (Ctrl+A), вырезать, вставить. Способ быстрый, однако, неудобен, если существуют межзадачные связи, т.к. при вставке из буфера алгоблоков межзадачные связи автоматически не восстанавливаются;
- добавить в модуль любые «временные» алгоблоки в количестве, равном числу «дырок». После этого изменить порядок алгоблоков, сместив «временные» алгоблоки в конец. После этого их удалить.

4.4 Вопрос 4. Что означают предупреждения компилятора о побочных эффектах связей?

При компиляции некоторые виды связей могут не одобряться компилятором. В таком случае выдаются предупреждения. Всего может быть 2 вида предупреждений:

- обратная связь может привести к побочному эффекту;
- связь может привести к побочному эффекту.

Не следует пренебрегать сообщениями о возможности побочных эффектов, т.к. могут возникнуть настолько сложно предсказуемые ошибки, что их отладка отнимет много времени и сил.

4.4.1 Побочные эффекты обратных связей

Предупреждение появляется, когда с выхода на вход одного и того же алгоблока проведена прямая связь. Суть предупреждения сводится к тому, что в подобном случае поведение алгоблока может отличаться от ожидаемого.

Для примера возьмем алгоблок конкатенации нескольких строк (алгоритм **Сцепить** в группе **Строки**). Допустим, мы хотим, чтобы сцепление происходило по формуле $S(t) = S(t-1) + Cs$, где $S(t)$ – значение строки в цикле N, $S(t-1)$ – значение строки в цикле N-1, Cs – константная строка.



Рисунок 51 – Пример прямой обратной связи

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

При компиляции получим предупреждение о том, что связь может привести к побочным эффектам. Действительно в одном и том же цикле значения на входе А3 по сути равно значению выхода В, которое в начале работы алгоритма всегда приравнивается значению первого входа, т.е. константе Cs. Как результат – реализуется формула $S(t) = Cs + Cs$ и вопреки ожиданиям, значение выхода всегда будет равно 'Cs Cs'. Попробуйте теперь разорвать связь между А3 и В и провести ее между А2 и В. В результате значение на выходе В изменится и будет всегда равно 'Cs'. Видно, что результат не только отличен от ожидаемого, но еще зависит от порядка связывания идентичных входов с выходом. Такой результат является непредсказуемым и его принято характеризовать, как побочный эффект.

Для устранения этого нежелательного эффекта, можно ввести дополнительный алгоблок - повторитель между выходом алгоблока конкатенации и его входом, как показано на рисунке 52

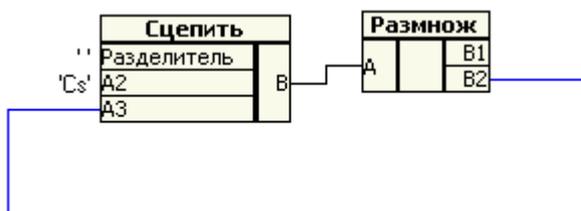


Рисунок 52 – Пример устранения обратной связи.

В таком случае значения на входе А3 будет значением выхода В от предыдущего цикла, что и требовалось для реализации формулы $S(t) = S(t-1) + Cs$.

4.4.2 Связь может привести к побочному эффекту

Некоторые виды связей могут приводить к непредсказуемым побочным эффектам, вследствие особенностей конвертирования сигналов различных типов. Рассмотрим случай, когда логический выход алгоблока А1, находящегося в задаче 31, связывается с событийным входом алгоблока А2, находящегося в задаче 32. Если задачи работают асинхронно (например, с разными временами циклов), то существует большая вероятность того, что логический сигнал, сформированный на выходе А1, не будет конвертирован в событийный сигнал на входе А2. Это может произойти в силу того, что алгоблок неявного конвертирования логического сигнала в событийный размещается компилятором в той же задаче, в которой находится алгоблок – приемник сигнала (в нашем случае алгоблок А2). Как следствие – между задачами передается логический сигнал, что в определенных ситуациях может привести к его потере на стороне приемника. Действительно, предположим, что в момент, когда на выходе А1 был сформирован логический сигнал, задача 32 не выполнялась (например, ожидала свое время цикла). За это время на выходе А1 логический сигнал успел перейти в состояние Нет. После чего задача 32 начала выполняться. Понятно, что на входе неявного алгоритма конвертера уже нет логической 1 (Да) и событие сформировано не будет. Таким образом, работа конвертера сигналов становится непредсказуемой и заранее не очевидной – это может приводить к побочным эффектам. Для того, чтобы исправить положение, нужно явно поместить конвертер сигналов логического типа в событийный тип в одну из задач (31 или 32). Если требуется гарантированное формирование событийного сигнала по фронту логического, то конвертер [логический]->[событийный] (группа Общие) следует поместить в задаче 31. Если же требуется формировать сигнал только по факту его появления в момент выполнения задачи 32, то алгоблок явного конвертирования, следует поместить в задаче 32.

4.4.3 Побочные эффекты связей, не отслеживаемые компилятором

Существуют некоторые виды побочных эффектов, которые не могут быть отслежены компилятором, но, тем не менее, могут приводить к серьезным ошибкам, вследствие побочных эффектов. Для примера рассмотрим случай, когда алгоблок А1 выполняется в задаче 31 и алгоблок А2 выполняется в задаче 32. Логический выход алгоблока А1 связан с логическим входом алгоблока А2. Если задачи работают асинхронно, то возможна ситуация, когда передний фронт логического сигнала будет потерян, а сам сигнал пропущен. Действительно, если значение Да на выходе А1 успело появиться и исчезнуть до того, как А2 успел выполняться, такой сигнал будет пропущен и поведение программы может стать непредсказуемым. Подобная ситуация не отслеживается компилятором, но тем не менее чревата ошибками. Если стоит задача обязательного определения фронта сигнала на входе алгоблока А2 (т.е. сигнал обязан быть воспринят, даже если он перестал формироваться на момент выполнения А2), его следует превратить в событие. Для этого в задачу 31 добавляется алгоблок явного преобразования [логический]->[событийный], вход которого связывается с выходом А1. С выхода

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

преобразователя, сигнал можно передать на вход **A2**. При этом в задачу **32** будет подставлен неявный преобразователь **[событийный]->[логический]**. Таким образом между задачами будет передаваться событийный сигнал, а значит фронт логического сигнала на входе **A2** не будет потерян при любых условиях.

4.5 Вопрос 5. Что нужно сделать, чтобы запустить Мезон-редактор?

Мезон является подсистемой Квинта, поэтому он запускается на любой Рабочей станции. Персональный компьютер, на котором необходимо запустить Мезон, должен быть описана в Базе данных проекта АСУ ТП.

4.6 Вопрос 6. Почему отсутствует нужный Мезон-сервер в диалоге создания и выбора техпрограмм Мезона?

В Базе данных проекта АСУ ТП должно быть установлено разрешение на запуск Мезон-сервера на нужной Рабочей станции. Для этого в закладке **Компьютер** пункта **Абоненты сети** программного приложения **Администратор БД** для компьютера этой Рабочей станции необходимо установить флаг **Мезон-сервер**. После этого в диалоге создания и выбора техпрограмм Мезона (меню **Файл / Техпрограмма**) появится возможность выбрать Мезон-сервер для той или иной техпрограммы.

Правила работы с приложением **Администратор БД** см. в документе «Комплексы программно-технические Квint-6. Администрирование проектов АСУ ТП. Руководство пользователя ПФДИ.421457.009 Из. 3».

4.7 Вопрос 7. Как управлять техпрограммами, запущенными на Мезон-станции?

Остановить или перезапустить техпрограмму, запущенную на Мезон-станции, можно тремя способами:

- при помощи приложения **RemoteMezonServerConsole**, входящего в состав пакета **Квинтегратор**. Из данного приложения можно управлять всеми серверами Мезона, Для выбора нужного сервера необходимо указать адрес компьютера, на которой он запущен, и подключиться к нему. После этого будут доступны команды управления техпрограммами, а также возможности изменения настроек ядра Мезона;
- при помощи кнопок на **Мезон-станции**. Для того, чтобы они были доступны, Мезон-станцию необходимо запускать с параметром **/localdir**;
- при помощи **Мезон-редактора**. Для этого необходимо открыть в Мезон-редакторе нужный проект, запустить его в режиме отладки и в диалоге выбрать запущенную техпрограмму. Мезон-редактор подключится к ней, и далее её можно остановить с утвердительным ответом на вопрос о ликвидации сервера. Для загрузки и остановки техпрограмм с помощью Мезон-редактора, пользователь должен иметь права на управление техпрограммами (**Администратор БД / Файл / Пользователи / <нужный пользователь> / Права / Пользователя / Функции / Выбор Функции / Загрузка тех. программы**).

4.8 Вопрос 8. Как перенести проект из одной базы Квинта в другую?

Физически **модуль проекта** находится в одной из таблиц в составе Базы данных проекта, а все **модули более низкого уровня вложенности** (задачи, мнемосхемы, составные алгоритмы) находятся в **хранилищах** в виде отдельных файлов. Файлы модулей имеют расширения:

- *.tsk – задача;
- *.msh – мнемосхема;
- *.dga – составной алгоблок;
- *.mpe – файл импорта/экспорта проекта.

Для переноса проекта Мезона в другую Базу данных другого проекта на базе Квинта следует экспортировать проект Мезона в файл. Для этого нужно выполнить пункт меню **Файл / Экспортировать ТП**. Затем выбрать Базу данных другого проекта, открыть Мезон-редактор и с помощью меню **Файл / Импортировать ТП** импортировать проект Мезона. После этого подключить хранилища Мезона с помощью меню **Файл / Хранилища**. Если одновременно требуется создать новую копию хранилищ, то достаточно скопировать хранилища, содержащие

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

нужные файлы модулей проекта (с сохранением структуры хранилищ) в новое место и подключить его в качестве хранилища.

4.9 Вопрос 9. Как перенести модуль (задачу, мнемосхему, составной алгоритм) из одного проекта в другой?

Модули физически представляют собой файлы (*.tsk - задача, *.msh - мнемосхема, *.dga – составной алгоблок). В пределах Базы данных одного проекта один и тот же модуль может быть подключен в нескольких проектах Мезона.

Если модуль требуется использовать в других проектах АСУ ТП, то можно применить два способа:

- подключить к данному проекту хранилище, содержащее требуемый модуль;
- скопировать файл модуля в одно из хранилищ другого проекта. При этом синхронизация изменений в модуле и его копии становится заботой разработчика.

4.10 Вопрос 10. Как на локальной машине открыть архив?

Часто при отладке техпрограмм на машине, не имеющей доступа к сети проекта Квинта, возникает необходимость работы с архивом проекта.

В таком случае необходимо скопировать нужные тома архива на локальную машину и подключить архив. Это можно осуществить двумя способами:

- прописать в Базе данных проекта локальный компьютер в качестве архивной станции и на этом компьютере запустить приложение **Архивная станция**, указав местоположение томов архива. Способ позволяет считывать и записывать данные в архив;
- воспользоваться функцией перенаправления Архивных станций. Для этого необходимо открыть архив (**Квинт-6 / Анализ / Открытие архива**) и затем настроить переопределение архива на текущий компьютер. Способ позволяет только считывать данные из архива.

4.11 Вопрос 11. Как из техпрограммы Мезона записать значения в архив?

Из техпрограммы Мезона можно записывать значения в архив при помощи алгоритма **Писать параметр**. При этом возможна работа только с маркой типа **Параметр расчётный**.

Примечание - В Базе данных проекта объекты с марками типа **Параметр расчётный** нужно создавать «с нуля», а не копировать (с пом. кнопки ++) строки, существующие в приложении **Аркада**.

4.12 Вопрос 12. Почему отображается надпись «ошибка» в рабочей области проекта Мезона?

Вид рабочей области проекта зависит от того, в каком режиме открывается модуль проекта (см. 3.4.4).

Если модуль открывается с установленным флагом **Показать схему проекта**, задачи проекта будут отображаться с дополнительными входами, демонстрирующими какие входы каких алгоблоков в задаче участвуют в межмодульном обмене (глобальные связи).

Если флаг **Показать схему проекта** снят, то дополнительные входы и выходы задач отображаться не будут. В таком случае глобальные связи отобразить невозможно, на что и обращает внимание сообщение об ошибке.

На работоспособность технологической программы это не влияет.

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

Изм	Лист	№ докум	Подп.	Дата	ПФДИ.421457.009 И3.6	Лист
						60

4.13 Вопрос 13. Как настроить автоматический запуск технологической программы?

Автоматический запуск можно реализовать при помощи программы **Монитор приложений**.

В таблице 2 приведены настройки **Монитора приложений** для запуска Мезона.

Таблица 2 – Настройки Монитора приложений

№ п/п	Пункт настроек	Подпункт настроек	Параметр	Настройка параметра
1	Общие настройки	Безопасность	Рабочий стол	default
2	Приложения	Мезон-клиент	Главное окно	Максимизировано
3			Загружать профиль	Да
4			Код выключения активности	0
5			Командная строка	Текст вида «/workdir {08DCDA35-737A-42EB-93B9-ADC6D6C5C8BA} /run 100000001». Для получения командной строки: 1. открыть в Мезон-редакторе необходимую техпрограмму 2. выбрать пункт меню: Сервис -> Командная строка 3. нажать Ок 4. вставить из буфера обмена Windows текст полученной командной строки в пункт Монитора приложений
6			Мониторинг активен	Да
7			Название	Мезон-клиент (может быть произвольным)
8			Один экземпляр	Да
9			Переменные среды	
10			Период опроса	5
11			Пользователь	Должен быть указан пользователь с правами администратора
12			Приложение	C:\Program Files\Kvint6\BIN\Mezon.exe
13			Рабочий каталог	C:\Program Files\Kvint6\BIN\
14			Рабочий стол	default
15			Счетчик ошибок	10
16			Таймаут зависания	60
17			Таймаут зависания при старте	60
18			Приложения	Мезон-станция
19	Загружать профиль	Да		
20	Код выключения активности	0		
21	Командная строка			
22	Мониторинг активен	Да		
23	Название	Мезон-станция (может быть произвольным)		
24	Один экземпляр	Да		
25	Переменные среды	Workdir = {текст из фигурных скобок командной строки Мезон-клиента}		
26	Период опроса	5		
27	Пользователь	Должен быть указан пользователь с правами администратора		
28	Приложение	C:\Program Files\Kvint6\BIN\MezonServerConsole.exe		
29	Рабочий каталог	C:\Program Files\Kvint6\BIN\		
30	Рабочий стол	default		
31	Счетчик ошибок	10		
32	Таймаут зависания	60		
33	Таймаут зависания при старте	60		

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

При указанных в таблице настройках обеспечивается автоматический запуск Мезон-станции и Мезон-клиента при их закрытии оператором или перезапуске ЭВМ.

Если в параметре **Рабочий стол** указано значение **default**, то запуск соответствующего приложения будет произведён на текущем рабочем столе.

Если необходимо ограничить доступ оператора только к окнам задач, запущенных **Монитором приложений**, то в параметре **Рабочий стол** необходимо указать любое значение (к примеру, **новый стол**). В этом случае, при запущенном **Мониторе приложений**, будет открыт пустой рабочий стол с открытыми окнами приложений, указанными в **Мониторе приложений**. Значение **winlogon** указывать не рекомендуется, т.к. в этом режиме существует внутреннее ограничение Windows, препятствующее работе Мезон-клиентов.

Настройки тайм-аутов относятся к запуску самого приложения **MezonServerConsole**, а не к техпрограммам. После запуска приложения действуют настройки таймаутов ядра Мезона.

4.14 Вопрос 14. Как контекстно открыть модуль техпрограммы?

Если в задаче используется составной алгоблок, то для его отладки следует открыть данный алгоблок контекстно. Что это означает? Допустим имеется составной алгоритм **A**, который дважды вставлен в задачу **3** (т.е. задача содержит 2 составных алгоблока, типа **A**, **A1** и **A2** соответственно). Тогда, чтобы отладить конкретный экземпляр алгоблока, следует открыть его контекстно, т.е. пройдя все вложения до данного составного алгоритма. В нашем случае в режиме отладки следует открыть задачу **3**, а затем открыть составной алгоблок **A1** и(ли) **A2**. В таком случае отладчик «поймёт» экземпляр какого алгоблока следует отлаживать и сможет отобразить правильные значения, специфичные именно для данного экземпляра. Аналогичным образом следует поступать для отладки составных алгоблоков любого уровня вложенности. Полный путь к модулю со всеми вложениями, будет показан в правом верхнем углу главного окна Мезона. После окончания сеанса отладки все модули, открытые контекстно, будут автоматически закрыты. Управлять этим можно с помощью настройки опции **Сервис / Параметры / Выполнение / Отладка / Закрывать макроблоки при отладке**.

4.15 Вопрос 15. Почему в режиме отладки некоторые модули становятся серыми, а внизу появляется надпись «модуль открыт не контекстно...»?

Иногда такое сообщение появляется, если модуль, открытый в Мезоне во время отладки не принадлежит текущему проекту Мезона (например, был открыт с помощью перетаскивания файлы курсором в рабочую область Мезона). Тогда следует закрыть данный модуль и попытаться открыть его, начиная с уровня проекта. Для этого следует выбрать пункт **Файл / Модули** и в появившемся диалоге выбрать корневой модуль проекта (первый пункт). Затем следует выбрать нужный модуль и открыть его контекстно.

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

5 Библиотека алгоритмов Мезона

Таблица 3 – Перечень алгоритмов

Группа	Алгоритм	Описание	Win32	WinCE
Общие	Алгоритмы общего назначения			
	Память	Запоминает входное значение при старте задачи или по событию	+	-
	Задержка	Передает значение со входа на выход с задержкой в 1 цикл	+	-
	Выбор	Передает на выход значение одного из двух входов, в зависимости от значения логического входа	+	-
	МножВыбор	Передает на выход значение одного из N входов, в зависимости от значения целочисленного входа (обобщение АнМульт на другие типы входов)	+	-
	Инициализация	Копирует на выход один из входов в зависимости от контекста выполнения	+	-
	Размнож (Разветв)	Размножитель значения	+	-
	Формула	Производит вычисления по введенной формуле	+	-
	Сравнить	Универсальный алгоритм сравнения для любых типов сигналов	+	-
	ТВЫ	Типизированный вывод	+	+
	ТВВ	Типизированный ввод	+	+
	Переменная	Типизированная переменная	+	+
	[логический]->[целый]	Явные преобразователи типов	+	+
	[логический]->[плавающая запятая]		+	+
	[логический]->[строковый]		+	+
	[логический]->[событийный]		+	+
	[целый]->[логический]		+	+
	[целый]->[плавающая запятая]		+	+
	[целый]->[строковый]		+	+
	[Плавающая запятая]->[логический]		+	+
	[Плавающая запятая]->[целый]		+	+
	[Плавающая запятая]->[строковый]		+	+
	[строковый]->[контейнер данных]		+	+
	[контейнер данных]->[строковый]		+	+
	[дата и время]->[строковый]		+	+
	[длительность]->[строковый]		+	+
	[событийный]->[логический]		+	+
Логика	Алгоритмы булевой логики			
	Не	Инверсия логического сигнала	+	+
	Или	Все алгоблоки работают с логическими сигналами. Число входов настраивается	+	+
	И		+	+
	ИсклИли		+	+

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

Продолжение таблицы 3

Группа	Алгоритм	Описание	Win32	WinCE
Математика		Математические алгоритмы		
1	Слож	Число входов настраивается	+	+
2	Выч		+	+
3	Умнож	Число входов настраивается	+	+
4	Делен		+	+
5	ДелОст	Деление с остатком	+	+
6	Округл	Округляет число с заданной точностью	+	+
7	Степень	Возведение числа в степень	+	+
8	Корень	Алгебраические алгоритмы с одним входом и одним выходом	+	+
9	-X		+	+
10	1/X		+	+
11	10^X		+	+
12	E^X		+	+
13	Lg		+	+
14	Ln		+	+
15	Sin		+	+
16	Cos		+	+
17	Tg		+	+
18	ArcSin		+	+
19	ArcCos	+	+	
20	ArcTg	+	+	
21	X!	Факториал	+	+
Биты		Алгоритмы битовой логики и битовых операций		
22	БитНе	Все алгоблоки работают с векторными сигналами. Число входов не настраивается	+	+
23	БитИ		+	+
24	БитИли		+	+
25	БитИскл		+	+
26	СдвигВлево		+	+
27	СдвигВправо		+	+
28	БитШифр		+	+
29	БитДешифр		+	+
Строки		Алгоритмы работы со строками		
30	РегСтр	Изменение регистра строки	+	-
31	ДлСтр	Длина строки	+	-
32	Сцепить	Конкатенация N строк	+	-
33	ПодСтр	Выделение подстроки указанного размера от указанной позиции	+	-
34	ПовторСтр	Повторяет строку N раз	+	-
35	ПоискСтр	Поиск указанной подстроки от указанной позиции	+	-
36	ПреобрСтр	Преобразует строку в целый, плавающий или дата-время формат	+	-
Время		Алгоритмы операций со временем		
37	Сейчас	Упаковывает значение текущей даты и времени	+	-
38	УпакВр	Упаковывает заданное значение даты и времени	+	-
39	РаспакВр	Распаковывает значение даты и времени	+	-
40	УпакДлит	Упаковывает значение длительности	+	-
41	РаспакДлит	Распаковывает значение длительности	+	-
42	СмещВр	Вычисляет смещение во времени относительно заданного + или - длительность	+	-
43	ДлитВр	Вычисляет длительность между двумя значениями даты и времени	+	-

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

Продолжение таблицы 3

Группа	Алгоритм	Описание	Win32	WinCE
События	Алгоритмы для работы с событиями			
44	СобИли	Выдает событие на выходе, если произошло хотя бы одно из N входных событий	+	+
45	Соби	Выдает событие на выходе, если произошли все N входных событий	+	+
46	СчетСоб	Подсчитывает кол-во событий поступивших на вход	+	+
47	ВилСоб (СобВил)	Направляет событие на 1 из двух выходов в зависимости от логического входа	+	+
48	ЕстьСоб	Преобразование события в логическое значение	+	+
Управление	Алгоритмы управления			
49	Сначала	Блокирует выполнение всех последующих алгоблоков этой задачи	+	-
50	Останов	Останавливает выполнение задачи	+	-
51	Ошибка	Останавливает выполнение задачи и формирует код ошибки	+	-
52	RS-триггер	Хранение двух устойчивых состояний (reset/set)	+	+
53	T-триггер	Переключение между двумя устойчивыми состояниями	+	+
54	Таймер	Предназначен для выполнения действий при истечении заданной длительности	+	+
55	Цикл	Предназначен для выполнения повторяющихся действий	+	+
56	Шаг	Предназначен для выполнения действий по таймауту или по условию	+	+
Отладка	Алгоритмы, предназначенные для отладки проекта			
57	ОтлПечать	Отладочная печать в консоль	+	-
58	ВремяЦикла	Тест времени выполнения задачи	+	+
59	MIDI-Вывод	Вывод в midi-синтезатор	+	-
60	MIDI-Ввод	Ввод нот с устройства ввода midi	+	-
61	MIDI-Нота	Назначение нот для вывода в midi-синтезатор	+	-
62	Такт	Тактирование midi-синтезатора	+	-
Мнемосимволы	Элементы пользовательского интерфейса			
63	Кнопка	Реализация кнопки с полностью настраиваемыми параметрами.	+	+
64	Редактор	Редактор с функцией ввода-вывода многострочного текста	+	+
65	Индикатор	Настраиваемый индикатор с возможностью динамического задания цвета фона и текста	+	+
66	Ползунок	Многофункциональный аналоговый задатчик с	+	+
67	Редактор значений	Ввод чисел с плавающей точкой в строчном виде	+	+
68	Слайд	Вывод изображений по номеру	+	+
69	Управление задачей	Запуск и останов выполнения выбранной задачи в ручную или по расписанию	+	+
70	График	Построение графиков	+	+

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

Продолжение таблицы 3

Группа	Алгоритм	Описание	Win32	WinCE
Моделирование	Виртуальные контроллеры и модели УСО			
71	Виконт	Виртуальный контроллер	+	-
72	Задвижка	Универсальная модель задвижки	+	-
73	БОК	Универсальная модель быстроотсечного клапана	+	-
74	РегКлапан	Универсальная модель регулирующего клапана	+	-
75	АД-1	Модель асинхронного электродвигателя	+	-
76	Фрейм	Сохранение и восстановление фреймов модели	+	-
77	АЦП	Модель Аналогово-Цифрового Преобразователя	+	-
78	ДЦП	Модель Дискретно-Цифрового Преобразователя	+	-
79	ДВВ	Модель Дискретного Ввода-Вывода	+	-
80	АВВ	Модель Аналогового Ввода-Вывода	+	-
81	ЦАП	Модель Цифро-Аналогового Преобразователя	+	-
82	ЦДП	Модель Цифро-Дискретного Преобразователя	+	-
83	ЦИП	Модель Цифро-Импульсного Преобразователя	+	-
84	МЗТ	Модель Модуля Защиты Турбины	+	-
85	МЧТ	Модель модуля Мониторинга Частоты Турбины	+	-
ТАУ	Алгоритмы моделей элементарных звеньев			
86	Задер	Звено задержки	+	-
87	Интегр	Интегрирующее звено	+	-
88	Диффер	Дифференцирующее звено	+	-
89	ИнЗвено1	Инерциальное звено 1-го порядка	+	-
90	ИнЗвено2	Инерциальное звено 2-го порядка	+	-
Квинт	Алгоритмы связи с Ремиконтами и другими абонентами Квинта			
91	ЧитатьПараметр	Читает текущее значение параметра из указанного источника	+	-
92	ПисатьПараметр	Записывает значение параметра в указанный приемник	+	-
93	История	Читает значение параметра из архива	+	-
94	ЧислоВРем	Переводит значение в формат контроллера	+	-/+
95	РемВЧисло	Переводит значения контроллера в обычный формат	+	-/+
96	ИВЫ	Осуществляет информационный вывод в ремиконты	+	+
97	ИВВ	Осуществляет информационный ввод из ремиконтов	+	+
Excel	Алгоритмы для работы с Microsoft Excel			
98	КнигаXL	Открывает книгу Excel	+	-
99	ЗаписьXL	Записывает данные в книгу, открытую алгоритмом КнигаXL	+	-
100	ЧтениеXL	Читает данные из книги открытой алгоритмом КнигаXL	+	-
101	ЛистXL	Возвращает использованный диапазон листа книги, открытой алгоритмом КнигаXL	+	-
102	ДиапазонXL	Позволяет задать настройки диапазона ячеек книги, открытой алгоритмом КнигаXL	+	-
103	ЯчейкаXL	Позволяет задать настройки ячейки книги, открытой алгоритмом КнигаXL	+	-

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

Продолжение таблицы 3

Группа	Алгоритм	Описание	Win32	WinCE
Файлы	Алгоритмы управления файлами			
104	Процесс	Запускает процесс	+	-
105	Выбор файла	Открывает файл для операций чтения/записи	+	-
106	Запись	Записывает данные в файл	+	-
107	Чтение	Считывает данные из файла	+	-
Таблицы	Алгоритмы операций с таблицами			
108	Писать в таблицу	Записывает данные в таблицу MS Access или SQL	+	-
109	Читать таблицу	Читает данные из таблицы MS Access или SQL	+	-
Случайные	Генерация случайных чисел и событий			
110	СлучЧисло	Генерирует случайное действительное число в заданном диапазоне	+	+
111	СлучЦелое	Генерирует случайное целое число в заданном диапазоне	+	+
112	СлучЛог	Генерирует случайное логическое значение	+	+
113	СлучСоб	Генерирует случайное событие с заданной периодичностью	+	+
Modbus	Прием и передача данных по протоколу ModBus			
114	ModBus-ввод	Получение данных по протоколу ModBus/TCP/IP	+	+
115	ModBus-вывод	Передача данных по протоколу ModBus/TCP/IP	+	+

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

6.2 Группа 'Биты'

6.2.1 БитНе

6.2.1.1 Назначение

Побитовая инверсия 32-х разрядного битового поля (вектора).

6.2.1.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	A	вектор	0	Преобразуемое 32-х битовое значение
Выход	B	вектор	-	Результат побитовой инверсии

6.2.1.3 Назначение

Побитовая инверсия 32-х разрядного битового поля (вектора)

6.2.1.4 Описание алгоритма

Алгоритм производит инверсию каждого бита сигнала на входе A, результат которой представлен на выходе B.

6.2.2 БитИ

6.2.2.1 Назначение

Логическое побитовое И двух 32-х разрядных битовых полей (векторов).

6.2.2.2 Параметры

Наименование	Обознач.	Тип	Номер	Нач. знач.	Назначение
Входы	A1	вектор	1	0	Первый 32-х битовый операнд
	A2	вектор	2	0	Второй 32-х битовый операнд
Выход	B	вектор	1	-	Результат побитового логического И между парными битами каждого операнда

6.2.2.3 Описание алгоритма

Алгоритм производит логическое умножение битов с одинаковыми номерами на входах A1 A2 по следующей таблице истинности:

A1	A2	B
0	0	0
0	1	0
1	0	0
1	1	1

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

6.2.3 БитИли

6.2.3.1 Назначение

Логическое побитовое ИЛИ двух 32-х разрядных битовых полей (векторов).

6.2.3.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	A1	вектор	0	Первый 32-х битовый операнд
	A2	вектор	0	Второй 32-х битовый операнд
Выход	B	вектор	-	Результат побитового логического ИЛИ между парными битами каждого операнда

6.2.4 Описание алгоритма

Алгоритм производит логическое сложение битов с одинаковыми номерами на входах A1 и A2 по следующей таблице истинности:

A1	A2	B
0	0	0
0	1	1
1	0	1
1	1	1

6.2.5 БитИскл

6.2.5.1 Назначение

Логическое побитовое ИСКЛЮЧАЮЩЕЕ ИЛИ двух 32-х разрядных битовых полей (векторов).

6.2.5.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	A1	вектор	0	Первый 32-х битовый операнд
	A2	вектор	0	Второй 32-х битовый операнд
Выход	B	вектор	-	Результат побитового логического ИСКЛЮЧАЮЩЕГО ИЛИ между парными битами каждого операнда

6.2.5.3 Описание алгоритма

Алгоритм производит логическое исключаящее сложение битов с одинаковыми номерами на входах A1 и A2 по следующей таблице истинности:

A1	A2	B
0	0	0
0	1	1
1	0	1
1	1	0

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

6.2.6 СдвигВлево

6.2.6.1 Назначение

Побитовый СДВИГ ВЛЕВО 32-х разрядного битового поля (вектора).

6.2.6.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	A	вектор	0	32-х битовый операнд
	Сдвиг	целый	0	Число, на которое следует сдвинуть биты
Выход	B	вектор	-	Результат побитового сдвига влево

6.2.6.3 Описание алгоритма

Алгоритм, сдвигающий битовый вектор A влево на число бит, указанных на входе Сдвиг. Сдвиг влево на 1 бит соответствует умножению числа (вектора) на 2, если результат находится в пределах разрядности двойного слова (32 бита)

6.2.7 СдвигВправо

6.2.7.1 Назначение

Побитовый СДВИГ ВПРАВО 32-х разрядного битового поля (вектора).

6.2.7.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	A	векторный	0	32-х битовый операнд
	Сдвиг	целый	0	Число, на которое следует сдвинуть биты
Выход	B	векторный	-	Результат побитового сдвига вправо

6.2.7.3 Описание алгоритма

Алгоритм, сдвигающий битовый вектор A вправо на число бит, указанных на входе Сдвиг. Сдвиг вправо на 1 бит соответствует делению числа (вектора) на 2, если результат находится в пределах разрядности двойного слова (32 бита).

6.2.8 БитШифр Назначение

Битовый шифратор.

6.2.8.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Мин. знач.	Макс. знач.	Назначение
Входы	N	целый	1	1	32	Номер бита в векторе, который будет зашифрован первым
	A.dN	логический	0	-	-	первый бит dN
	A.dN+1	логический	0	-	-	второй бит dN+1
	A.dN+2	логический	0	-	-	третий бит dN+2
	A.dN+3	логический	0	-	-	четвертый бит dN+3
Выход	B	векторный	-	-	-	Вектор, часть бит которого (начиная с N) зашифрованы

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

6.2.8.3 Описание алгоритма

Алгоритм шифрации вектора. Предназначен для установки значения четырех бит вектора, начиная с N-ого бита.

6.2.9 БитДешифр

6.2.9.1 Назначение

Битовый дешифратор.

6.2.9.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Мин. знач.	Макс. знач.	Назначение
Входы	N	целый	1	1	32	Номер бита, с которого начинается дешифрация
	A	логический	0	-	-	Вектор, часть бит которого, необходимо дешифровать (выделить)
Выход	V.dN	логический	-	-	-	значение бита под номером N вектора X
	V.dN+1	логический	-	-	-	значение бита под номером N+1 вектора X
	V.dN+2	логический	-	-	-	значение бита под номером N+2 вектора X
	V.dN+3	логический	-	-	-	значение бита под номером N+3 вектора X

6.2.9.3 Описание алгоритма

Алгоритм дешифрации вектора. Предназначен для выделения четырех бит вектора, начиная с N-ого бита.

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

Изм	Лист	№ докум	Подп.	Дата	ПФДИ.421457.009 И3.6	Лист
						72

6.3 Группа 'Общие'

6.3.1 Память

6.3.1.1 Назначение

Запоминает входное значение при старте задачи или по событию.

6.3.1.2 Параметры

Наименование	Обозначение	Тип	Назначение
Входы	<i>A</i>	Любой	Сохраняемое значение
	<i>Сохр</i>	Событийный	Событие, при котором происходит сохранение
Выходы	<i>B</i>	Любой	Сохраненное значение

6.3.1.3 Описание алгоритма

Выход *B* имеет тот же тип, что и вход *A*. Алгоритм копирует значение входа *A* на выход *B* при следующих условиях:

- если вход *Сохр* связан, то при появлении события на входе *Сохр*;
- если вход *Сохр* не связан, то при запуске задачи, содержащей алгоблок **Память**.

6.3.1.4 Мастер настройки

Позволяет выбрать тип сохраняемого значения. После того как вход *A* или выход *B* связаны, их тип изменить нельзя.

6.3.2 Задержка

6.3.2.1 Назначение

Передаёт значение с входа на выход с задержкой в 1 цикл.

6.3.2.2 Параметры

Наименование	Обозначение	Тип	Назначение
Входы	<i>A</i>	Любой	Входное значение
Выходы	<i>B</i>	Любой	Выходное значение

6.3.2.3 Описание алгоритма

Выход *B* имеет тот же тип, что и вход *A*. Алгоритм имеет одну внутреннюю переменную того же типа, что *A* и *B*. Алгоритм копирует значение внутренней переменной на выход *B*, а затем копирует значение входа *A* во внутреннюю переменную.

Если значение событийного типа, то алгоритм выдаёт на выходе столько событий, сколько поступило на вход в предыдущем цикле.

6.3.2.4 Мастер настройки

Позволяет выбрать тип значения. После того как вход *A* или выход *B* связаны, их тип изменить нельзя.

6.3.3 Сравнить

6.3.3.1 Назначение

Сравнивает значения двух входов.

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

6.3.3.2 Параметры

Наименование	Обозначение	Тип	Назначение
Входы	<i>A</i>	Любой	Первое сравниваемое значение
	<i>B</i>	Любой	Второе сравниваемое значение
Выходы	<i>==</i>	Логический	<i>A</i> равно <i>B</i>
	<i>!=</i>	Логический	<i>A</i> не равно <i>B</i>
	<i>></i>	Логический	<i>A</i> больше <i>B</i>
	<i><</i>	Логический	<i>A</i> меньше <i>B</i>
	<i>Знак</i>	Целый	0 при $A=B$, 1 при $A>B$, -1 при $A<B$

6.3.3.3 Описание алгоритма

Входы *A* и *B* – одного типа. Алгоритм сравнивает значения на этих входах и выдает результат сравнения на выходы. Например, на выходе *>* будет значение ДА при $A > B$, и НЕТ при $A \leq B$.

6.3.3.4 Мастер настройки

Позволяет выбрать тип входов и состав необходимых выходов. Если входы имеют строковый тип, в мастере дополнительно указывается, следует ли учитывать регистр символов строк при их сравнении. После того как входы связаны, их тип изменить нельзя.

6.3.4 Выбор

6.3.4.1 Назначение

Копирует на выход одно из двух значений в зависимости от условия.

6.3.4.2 Параметры

Наименование	Обозначение	Тип	Назначение
Входы	<i>Усл</i>	Логический	Условие
	<i>Да</i>	Любой	Копируется на выход при $Усл=Да$
	<i>Нет</i>	Любой	Копируется на выход при $Усл=Нет$
Выходы	<i>B</i>	Любой	Результат выбора

6.3.4.3 Описание алгоритма

Входы *Да*, *Нет* и выход *B* – одного типа. Алгоритм проверяет значение входа *Усл* и копирует на выход одно из значений на входах *Да*, *Нет* в зависимости от значения *Усл*.

6.3.4.4 Мастер настройки

Позволяет выбрать тип значения.

6.3.5 МножВыборНазначение

Копирует на выход одно из нескольких входных значений.

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

6.3.5.2 Параметры

Наименование	Обозначение	Тип	Назначение
Входы	<i>Индекс</i>	Целый	Индекс
	A_0	Любой	Копируется на выход при <i>Индекс</i> =0
	A_1	Любой	Копируется на выход при <i>Индекс</i> =1

	A_{N-1}	Любой	Копируется на выход при <i>Индекс</i> =N-1
	<i>Иначе</i>	Любой	Копируется на выход при прочих значениях <i>Индекс</i>
Выходы	B	Любой	Результат выбора

6.3.5.3 Описание алгоритма

Количество N вариантов выбора настраивается в Мастере. Алгоритм вычисляет значение *i* входа *Индекс*. Если это значение лежит в пределах $0 \leq i < N$, то алгоритм копирует на выход значение входа A_i . В противном случае копирует на выход значение входа *Иначе*.

6.3.5.4 Мастер настройки

Позволяет выбрать тип значения и количество вариантов выбора.

6.3.6 Размнож

6.3.6.1 Назначение

Размножает входное значение на несколько выходов.

6.3.6.2 Параметры

Наименование	Обозначение	Тип	Назначение
Входы	A	Любой	Входное значение
Выходы	B_1	Любой	Выход 1
	B_2	Любой	Выход 2

	B_N	Любой	Выход N

6.3.6.3 Описание алгоритма

Все выходы и вход – одинакового типа. Алгоритм копирует значение входа на все выходы. При редактировании обычных задач в этом алгоритме нет необходимости, поскольку один выход и так можно связать с несколькими входами. Потребность в алгоритме Размнож возникает при редактировании групп, когда нужно связать вывод алгоблока с выводом группы.

6.3.6.4 Мастер настройки

Позволяет задать количество выходов.

6.3.7 Инициализация

6.3.7.1 Назначение

Копирует на выход один из входов в зависимости от контекста выполнения.

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

6.3.7.2 Параметры

Все входы и выходы имеют один и тот же тип.

Наименование	Обозначение	Тип	Назначение
	Начало	Любой	Копируется на выход при запуске задачи
	Продолжение	Любой	Копируется на выход в каждом цикле кроме первого
Выходы	V	Любой	Выходное значение
	Запусков	Целый	Количество запусков задачи. Равно 1 при первом запуске сервера, увеличивается на 1 при каждом запуске задачи, содержащей этот алгоблок.

6.3.7.3 Описание алгоритма

Алгоритм удобно использовать для инициализации сигналов, значения которых изменяются в ходе работы задачи. Например, пусть требуется реализовать арифметическую прогрессию: чтобы при запуске задачи значение некоторого выхода было равно A , и в каждом цикле оно увеличивалось на B .

В приведённом на рисунке 53 примере $A = 100$, $B = -1$:



Рисунок 53 – Пример применения алгоблока Инициализация

6.3.7.4 Мастер настройки

Позволяет выбрать тип сохраняемого значения. Тип можно изменить, только если входы и выходы алгоблока не связаны.

6.3.8 Формула

6.3.8.1 Назначение

Расчёт по формуле, заданной в виде текста.

6.3.8.2 Параметры

Состав входов и выходов полностью задаётся в Мастере настройки

6.3.8.3 Описание алгоритма

Формула описывает зависимость выходных значений от входных. Алгоритм целесообразно применять в тех случаях, если эта зависимость достаточно сложна и уже записана в виде математического выражения. В силу своей универсальности этот алгоритм можно применять и в более широких целях, однако это требует серьезной квалификации программиста.

Формула использует язык программирования C#. Пользователь вводит текст программы в Мастере настройки. При закрытии мастера введенная программа компилируется встроенным компилятором C# и хранится затем в откомпилированном виде. При запуске сервера выполнения откомпилированный код выполняется.

Синтаксис языка C# в части составления выражений аналогичен языку C. Подробное описание языка C# можно найти в соответствующей литературе. В таблице 4 приведён список основных операторов.

Имп. № подл.	Подп. и дата
Взаим. имп. №	Подп. и дата
Имп. №	Подп. и дата
Подп. и дата	Подп. и дата

Таблица 4 – Список операторов

Тип	Приоритет	Оператор	Назначение	Пример	
				Выражение	Результат
Унарные	13	++	Увеличивает операнд на 1		
		--	Уменьшает операнд на 1		
		-	Противоположное число	-(3+5)	-8
		!	Логическое отрицание	!true	false
		~	Побитовое дополнение	~0x00000001	0xFFFFFFFF
Бинарные	12	*	Умножение	2 * 2	4
		/	Деление	5 / 2 5.0 / 2	2 2.5
		%	Остаток от деления	5 % 2 5.0 % 2.2	1 0.6
	11	+	Сложение	3 + 5 "abc" + "def"	8 "abcdef"
		-	Вычитание	3 - 5	-2
	10	<<	Побитовый сдвиг влево	3 << 5	96
		>>	Побитовый сдвиг вправо	8 >> 1	4
	9	<	Меньше	3 < 3	false
		>	Больше	5 > 3	true
		<=	Меньше или равно	3 <= 3	true
		>=	Больше или равно	3 >= -2	true
		==	Равно (сравнение)	2*2 == 4	true
	8	!=	Не равно	"abc" != "abc"	false
	7	&	Побитовое И		
	6	^	Побитовое исключающее ИЛИ		
5		Побитовое ИЛИ			
4	&&	Логическое И			
3		Логическое ИЛИ			
Тернарный	2	?:	Выбор альтернативы	x > 0? Math.Sqrt(x) : 0	Корень из x, если x положителен, иначе 0
Бинарные	1	=	Выполнение соответствующей операции с последующим присваиванием результата левой части	x = 2; x += 3;	5
		*=			
		/=			
		%=			
		+=			
		-=			
		<<=			
		>>=			
		&=			
		^=			
=					

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

Изм	Лист	№ докум	Подп.	Дата

6.3.8.4 Мастер настройки

6.3.8.4.1 Задание состава входов и выходов

Для добавления входа щелкните правой кнопкой мыши по узлу Входы в дереве. В выпавшем контекстном меню выберите пункт Добавить. В открывшемся окне диалога укажите свойства входа (см. ниже). Выходы добавляются аналогично.

Для изменения свойств входа или выхода в его контекстном меню укажите Изменить. Откроется то же самое окно диалога. Для удаления входа или выхода выберите в его контекстном меню пункт Удалить.

6.3.8.4.2 Свойства входов и выходов

Тип определяет тип значения входа или выхода. Поддерживаются следующие типы (таблица 5):

Таблица 5 – Типы входов и выходов

bool	Логическая величина, принимает значения true или false .	Встроенные типы C#
Int	Целое число	
double	Вещественное число	
string	Строка	
DateTime	Дата + время	
TimeSpan	Интервал времени	
EventPush или EventPop	Событие	Специальные классы, используемые для передачи событий. EventPush используется на выходе, EventPop - на входе. Методы EventPush: void Push() - генерирует событие на выходе Методы EventPop: int Pop() - возвращает кол-во событий, поступивших на данный вход с момента предыдущего вызова bool HasOrigin() - возвращает true, если данный вход связан, false - в противном случае.

Имя входа-выхода будет использоваться в тексте формул, а также при отображении алгоблока в окне задачи. Имена всех входов и выходов данного алгоблока должны быть **различными**. Маленькие и большие буквы в имени различаются, т.е. один вход можно назвать x, а другой - X. В имени можно использовать буквы (как латинские, так и русские), цифры и символ подчеркивания. Имя не может начинаться с цифры.

Описание входа или выхода будет появляться в виде подсказки при наведении мыши на алгоблок в окне задачи. Описание не является обязательным, но рекомендуется.

6.3.8.4.3 Редактирование исходного текста

В окне Мастера настройки имеется 4 закладки для ввода текста формулы. Вводимый в них код имеет различное назначение. В дальнейшем под родительской задачей будем понимать задачу, содержащую данный алгоблок **Формула**. Имеются следующие закладки:

- **Вычисление.** Наиболее употребительная закладка. Введенный здесь код выполняется в каждом цикле работы родительской задачи;
- **Старт.** Этот код выполняется однократно при запуске родительской задачи. В данном коде можно использовать дополнительную переменную *startCount* типа int, значение которой равняется 1 при первом запуске задачи и увеличивается на 1 при каждом последующем запуске;
- **Стоп.** Этот код выполняется однократно при останове родительской задачи. В данном коде можно использовать дополнительную переменную *stopReason* типа int, значение которой позволяет, значение которой позволяет узнать причину останова задачи: Возможные значения этой переменной:
 - а) 0 - задача остановлена алгоблоком **Стоп** или по инициативе другого алгоблока;

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

- б) 1 - задача остановлена снаружи по событию **Стоп**;
 - в) 2 - Останавливается сервер выполнения;
 - г) $\geq 0 \times 1000$ - код ошибки при останове по алгоритму **Авария** или по инициативе другого алглобка
- **Дополнительно.** В этой закладке можно размещать описание и инициализацию внутренних переменных алглобка, а также вспомогательных функций. Внутренние переменные создаются и инициализируются при старте сервера выполнения и уничтожаются при останове сервера выполнения;
 - **Просмотр.** Эта закладка является не редактируемой. В ней выводится сгенерированный текст программы, который передается компилятору C#.

6.3.8.4.4 Вставка элементов дерева в текст

Если выполнить двойной щелчок мыши по узлу дерева, обозначающего вход или выход, то имя последнего будет вставлено в редактируемый текст. Кроме того, в дереве представлены наиболее употребительные функции (точнее, методы), которые тоже можно вставить в текст двойным щелчком.

6.3.8.4.5 Задание названия формулы

Самый верхний узел дерева определяет название данного алглобка. Название можно отредактировать непосредственно в поле дерева. Название используется при отображении алглобка в окне задачи, но никак не влияет на работу алглобка.

6.3.8.4.6 Компиляция кода

При нажатии кнопки Проверка выполняется компиляция всех введенных фрагментов кода. Если компилятор обнаруживает ошибку в одном из фрагментов, то соответствующая закладка активизируется, курсор указывает на позицию ошибки в тексте, а сообщение об ошибке выводится в панель состояния.

При нажатии кнопки ОК происходит то же самое, и, в случае отсутствия ошибок, окно Мастера закрывается. При нажатии кнопки **Отмена** окно Мастера закрывается без сохранения введенных изменений.

6.3.8.4.7 Подключение дополнительных сборок

Опытные программисты могут вызывать из программы код других сборок .NET. Для этого нужно добавить имена этих сборок в дерево (узел **Дополнительные сборки**). По умолчанию используется сборка **System**, содержащая базовые математические, строковые и др. функции.

6.3.9 Преобразователи

6.3.9.1 Назначение

Алгоритмы преобразования типов сигнала позволяют привести сигнал одного типа с выхода алглобка к сигналу другого типа на входе алглобка. Те же самые алгоритмы преобразования подставляются компилятором во время компиляции проекта, в случае, когда связываются разнотипные выходы алглобков. Такое преобразование носит неявный характер. Однако в некоторых случаях необходимо иметь возможность явного преобразования сигналов из одного типа в другой. Например неявное преобразование сигнала типа логический в сигнал типа событийный приведет к тому, что на этапе компиляции будет сформировано предупреждение о побочных эффектах такого преобразования. Действительно, выполняя неявное преобразование сигнала, компилятор подставляет алглобок преобразования так, что он выполняется в той же задаче, где и алглобок – приемник, но раньше него по порядку. Таким образом, между задачами будет передаваться логический сигнал, что в случае его небольшой длительности (по сравнению с периодом выполнения задачи с алглобком – приемником) может приводить к потере фронтов и неожиданным результатам работы проекта. В этом случае правильнее произвести явное преобразование сигнала типа логический в событийный тип сигнала, разместив алглобок преобразования сразу за алглобком – источником сигнала. Таким образом между задачами будет передаваться событийный сигнал, гарантирующий соблюдение фронтов (точнее их количества). Кроме приведенного примера, когда явное преобразование сигналов предпочтительнее неявного, существуют и другие. Использование

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

явного преобразования в некоторых случаях позволяет более наглядно отобразить логику выполнения задачи при этом никакие дополнительные накладные расходы (по сравнению с неявным преобразованием) внесены не будут.

Всего существует 15 алгоритмов-преобразователей. Состав их входов и выходов одинаков, поэтому таблица состава входов и выходов приводится одна на все алгоритмы.

6.3.9.2 Параметры

Наименование	Обозначение	Тип	Назначение
Входы	A	Зависит от преобразования	Преобразуемый сигнал
Выходы	B	Зависит от преобразования	Преобразованный сигнал

6.3.9.3 [Логический] -> [Целый]

6.3.9.3.1 Описание алгоритма

Алгоритм позволяет конвертировать логическое значение сигнала в целый или векторный тип.

6.3.9.3.2 Мастер настройки

Мастер настройки появляется до установки алгоритма в задачу и не может быть вызван у алгоблока, размещенного в задаче. С его помощью можно настроить способ преобразования логического значения в целое. Можно задать тип соединения и признак необходимости конвертирования. Соединить сигнал можно как единый вектор, при этом все биты выходного сигнала будут принимать значение входа; или как единичный бит, тогда в выходном сигнале тот бит, номер которого выбран в мастере настройки будет принимать значение входа. Если установить опцию инвертирования, то для преобразования будет взято инверсное значение входного сигнала.

6.3.9.4 [Логический] -> [Действительный]

6.3.9.4.1 Описание алгоритма

Алгоритм позволяет конвертировать логическое значение сигнала в значение в формате плавающей запятой. Если значение входного сигнала будет равно 0 (Нет) выходной сигнал примет значение 0,0. Иначе, выходной сигнал станет равным 1,0.

6.3.9.5 [Логический] -> [Строковый]

6.3.9.5.1 Описание алгоритма

Алгоритм позволяет конвертировать логическое значение сигнала в строковое. Если значение входного сигнала будет равно 0 (Нет) выходной сигнал примет значение '0'. Иначе, выходной сигнал станет равным '1'.

6.3.9.6 [Логический] -> [Событийный]

6.3.9.6.1 Описание алгоритма

Алгоритм позволяет конвертировать логическое значение сигнала в событие. При этом событием считается появление переднего фронта сигнала [изменение входного значения с 0 (Нет) на 1 (Да)].

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

6.3.9.7 [Целый] -> [Логический]

6.3.9.7.1 Описание алгоритма

Алгоритм позволяет конвертировать целое или векторное значение сигнала в логическое.

6.3.9.7.2 Мастер настройки

Мастер настройки появляется до установки алгоритма в задачу и не может быть вызван у алгоблока, размещенного в задаче. С его помощью можно настроить способ преобразования целого значения в логическое. Можно задать тип соединения и признак необходимости конвертирования. Соединить сигнал можно как единый вектор, при этом значение выходного сигнала будет равно 0 (Нет), только если все биты входного сигнала равны 0 (монтажное ИЛИ); или как единичный бит, тогда выходной сигнал будет принимать значение соответствующего бита входного сигнала, номер которого выбран в мастере настройки. Если установить опцию инвертирования, то после преобразования выходное значение будет инвертировано.

6.3.9.8 [Целый] -> [Действительный]

6.3.9.8.1 Описание алгоритма

Алгоритм позволяет конвертировать целое или векторное значение сигнала в значение с плавающей запятой. При этом целая часть числа с плавающей запятой будет равна входному значению, а дробная часть всегда будет равна нулю.

6.3.9.9 [Целый] -> [Строковый]

6.3.9.9.1 Описание алгоритма

Алгоритм позволяет конвертировать целое или векторное значение сигнала в строковое значение. Выходной сигнал будет содержать символическое представление входного значения.

6.3.9.10 [Действительный] -> [Логический]

6.3.9.10.1 Описание алгоритма

Алгоритм позволяет конвертировать значение с плавающей запятой в логическое. При этом выходной сигнал будет равен 0 (Нет) только если входной сигнал будет строго равен 0,0.

6.3.9.11 [Действительный] -> [Целый]

6.3.9.11.1 Описание алгоритма

Алгоритм позволяет конвертировать значение с плавающей запятой в целое или векторное. При этом выходной сигнал будет равен целой части входного сигнала (округление целой части не производится). Если модуль значения входного сигнала больше чем 2 в степени 32, значение выходного сигнала будет случайным

6.3.9.12 [Действительный] -> [Строковый]

6.3.9.12.1 Описание алгоритма

Алгоритм позволяет конвертировать значение с плавающей запятой в строковое. Выходное значение будет равно символическому представлению входного сигнала в инженерном формате (мантисса и порядок). Точность мантиссы до 16 знаков.

6.3.9.13 [Строковый] -> [Контейнер данных]

6.3.9.13.1 Описание алгоритма

Алгоритм позволяет конвертировать строковое значение тип контейнера данных. В режиме выполнения из строкового сигнала будет получен физический адрес строки, который будет передан на выход.

Инд. №				
Подп. и дата				
Инд. №				

6.3.10 ТВЫ

6.3.10.1 Назначение

Вывод типизированных данных из текущей техпрограммы Мезона в другую техпрограмму. Две техпрограммы могут находиться на различных Мезон-серверах различных типов.

6.3.10.2 Параметры

Наименование	Обозначение.	Тип	Назначение
Входы	<i>A</i>	Любой	Передаваемое значение
	<i>Период посылки</i>	Модельное время	Абсолютное или виртуальное время, задающее период отправки сообщений
Выходы	<i>Ошибка</i>	Целый	Код ошибки. 0 – нет ошибки.
	<i>Описание ошибки</i>	Строковый	Описание ошибки, возникшей при передаче
	<i>Осталось</i>	Модельное время	Время, оставшееся до срабатывания тайм-аута отправки сообщения

6.3.10.3 Описание алгоритма

Нет никаких ограничений на количество алгоритмов ТВЫ в одной техпрограмме, на число Мезон-серверов, в которые осуществляется вывод, на типы Мезон-серверов, а также на любые сочетания.

Алгоритм передает одно значение сигнала определенного типа, с указанной периодичностью. Для уникальной идентификации передаваемого значения используется произвольный идентификатор (id). При передаче сигнала контролируется тайм-аут. При этом успешный прием контролируется по принципу точка-точка. Т.е. успешной передачей считается прием сигнала алглобком на принимающей стороне. Во всех остальных случаях значение считается не отправленным и отсчитывается тайм-аут.

6.3.10.4 Мастер настройки

Позволяет выбрать тип передаваемого значения, задать идентификатор сигнала, тайм-аут передачи и строку подключения. Идентификатор сигнала может быть любым строковым значением (например, описанием передаваемого сигнала). Также можно получить гарантированно уникальный идентификатор, если нажать на кнопку «Задать уникальный». При добавлении нового алглобка ТВЫ автоматически создается уникальный идентификатор. Тайм-аут передачи задается в миллисекундах и по умолчанию равен 4000. Строка подключения идентифицирует приемник и строится по следующему правилу:

`alias=WINS_Name|DNS_Name|localhost|127.0.0.1|IP1[,IP2]`. Где WINS_Name – WINS имя принимающего Мезон-сервера, DNS_Name – «длинное» имя, раздаваемое DNS-сервером (предпочтительнее по сравнению с WINS), localhost – указывается, если связь осуществляется с техпрограммой, расположенной на том же Мезон-сервере, 127.0.0.1 – аналогично localhost, IP1[,IP2] – один или два IP адреса (в случае дублированной сети) в формате IPv4 принимающего Мезон-сервера (IP адреса следует использовать только если в качестве принимающего Мезон-сервера используется Мезон-контроллер).

Примеры строк подключения:

- `alias=COMP1` – связь с Мезон-станцией на машине COMP1 по ее WINS имени;
- `alias=Comp1.domain.local` – связь с Мезон-станцией на машине COMP1 по ее DNS имени;
- `alias=localhost` – связь с Мезон-сервером на локальной машине (связь между двумя техпрограммами одного Мезон-сервера);
- `alias=127.0.0.1` – аналогично;
- `alis=192.168.26.111,192.168.27.111` – связь с Мезон-сервером на машине с двумя сетевыми адаптерами с IP адресами 192.168.26.111 и 192.168.27.111.

Идентификатор сигнала и строка подключения должны образовывать уникальную пару в пределах одной системы.

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

Изм	Лист	№ докум	Подп.	Дата	ПФДИ.421457.009 И3.6	Лист
						83

6.3.11 ТВВ Назначение

Ввод типизированных данных из другой техпрограммы. Две техпрограммы могут находиться на различных Мезон-серверах различных типов.

6.3.11.2 Параметры

Наименование	Обозначение.	Тип	Назначение
Выходы	<i>В</i>	Любой	Значение принятого сигнала
	<i>Ошибка</i>	Целый	Код ошибки
	<i>Описание ошибки</i>	Строковый	Описание ошибки, возникшей при передаче
	<i>Осталось</i>	Модельное время	Время, оставшееся до срабатывания тайм-аута приема сообщения

6.3.11.3 Описание алгоритма

Нет никаких ограничений на количество алгоритмов **ТВВ** в одной техпрограмме, на число Мезон-серверов, из которых осуществляется прием, на типы Мезон-серверов, а также на любые сочетания. Алгоритм принимает одно значение сигнала определенного типа. Для уникальной идентификации передаваемого значения используется произвольный идентификатор (id). При приеме сигнала контролируется тайм-аут. В случае, если очередное значение сигнала не получено в течение установленного тайм-аута – формируется ошибка.

6.3.11.4 Мастер настройки

Позволяет выбрать тип принимаемого значения, задать идентификатор сигнала и тайм-аут приема. Идентификатор сигнала может быть любым строковым значением (например, описанием принимаемого сигнала). Также можно получить гарантированно уникальный идентификатор, если нажать на кнопку «Задать уникальный». При добавлении нового алглобка ТВВ автоматически создается уникальный идентификатор. В любом случае, для того, чтобы прием значения сигнала осуществился, идентификатор и тип сигнала должны совпадать с идентификатором и типом сигнала, передаваемого в данную техпрограмму из другой техпрограммы. Тайм-аут передачи задается в миллисекундах и по умолчанию равен 4000.

Примечание - Если в данную техпрограмму будут передаваться сигналы с одинаковых идентификатором, но от различных источников, последовательность приема таких сигналов будет непредсказуемой. Идентификатор принимаемого сигнала должен быть уникален в пределах одного Мезон-сервера.

6.3.12 Переменная

6.3.12.1 Назначение

Алгоритм предназначен для имитации функциональности переменной в языках программирования. Позволяет задать начальное значение, изменить его на значение выхода другого алгоритма или получить его на входе другого алгоритма.

6.3.12.2 Параметры

Наименование	Обозначение.	Тип	Назначение
Вход	<имя переменной>	Любой	Установка значения по умолчанию или присвоение значение с выхода другого алглобка
Выходы	<i>В</i>	Любой	Значение переменной

6.3.12.3 Описание алгоритма

Нет никаких ограничений на число используемых переменных в одной техпрограмме. Объявление переменной должно содержать уникальное в пределах техпрограммы имя. Имя может содержать не более 20 любых символов. Переменная может **использоваться** в различных задачах и мнемосхемах одной техпрограммы. Под использованием понимается однократное присваивание значения с выхода другого алглобка и многократное получение значения на входе других алглобков. Таким образом, значение переменной может меняться лишь в одной задаче. Это позволяет избежать неопределенности значения,

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

которая может возникнуть при одновременной модификации переменной из разных задач, работающих асинхронно. Во время компиляции техпрограммы алгоритм проверяет выполнение следующих условий:

- уникальность имени переменной в пределах техпрограммы;
- однократное присваивание значения с выхода другого алглока;
- совпадение типов переменных при их использовании с типом объявленной переменной;
- наличие объявленной переменной до ее первого использования. Другими словами, проверяется, чтобы переменная была объявлена раньше, чем она будет впервые использована;
- наличие связей входа и выхода алглока Переменная. Такая ситуация не допускается и в общем не имеет смысла.

Невыполнение любого из перечисленных условий приводит к возникновению соответствующих ошибок во время компиляции техпрограммы.

6.3.12.3.1 Объявление переменной

Для того, чтобы объявить переменную, следует разместить алглок **Переменная** в любой задаче, мнемосхеме или составном алглоке. При этом важно, чтобы модуль, в котором размещается объявление переменной, выполнялся в техпрограмме раньше всех модулей в которых она используется. При размещении алглока с помощью Мастера следует задать имя и тип объявляемой переменной. На входе алглока можно установить значение соответствующего типа. Таким образом можно проинициализировать переменную значением по умолчанию.

6.3.12.3.2 Присваивание значения переменной

Для того, чтобы присвоить значение переменной с выхода другого алглока, следует добавить алглок переменная и в мастере настройки указать те же имя и тип, что при объявлении. Затем следует связать выход другого алглока со входом алгоритма Переменная. Допускается однократное присваивание значения в пределах одной техпрограммы.

6.3.12.3.3 Получение значения переменной

Для того, чтобы получить значение переменной на входе другого алглока, следует добавить алглок переменная и в мастере настройки указать те же имя и тип, что при объявлении. Затем следует связать выход алглока Переменная со входом другого алглока.

6.3.12.3.4 Рекомендации

Рекомендуется для объявления переменных сделать отдельную задачу, в техпрограмме и разместить ее раньше остальных задач. Внутри такой задачи можно объявить все необходимые переменные. При этом задача может не выполняться (например, может быть установлено значение **Нет** (0) на входе **Разрешение задачи**), т.к. для объявления переменных выполнение задачи не требуется. Таким образом эта задача не будет отнимать процессорного времени для своего выполнения.

Если значение переменной используется в одном модуле на входе более чем одного алглока, рекомендуется добавлять один алглок переменная и от него вести связи на несколько входов других алглоков, если это не противоречит наглядности. Т.е. алглоки, использующие значение переменной на своих входах расположены близко. Это позволит не снижая наглядности использования переменной уменьшить вычислительные расходы.

Рекомендуется давать переменным осмысленные имена, желательно понятные без дополнительных комментариев. Это позволит «самодокументировать» технологическую программу.

6.3.12.4 Мастер настройки

Позволяет выбрать тип переменной и задать ее имя. Имя переменной может содержать не более 20 любых символов. Если переменная используется (связан вход или выход), изменить ее тип с помощью мастера невозможно. Для этого вначале следует разорвать связи, а после изменения типа вновь соединить.

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

6.3.13 Переменная 2

6.3.13.1 Назначение

Алгоритм предназначен для имитации функциональности переменной в языках программирования. Позволяет изменить значение переменной на значение выхода другого алгоритма или получить его на входе другого алгоритма. В отличие от алгоритма **Переменная** данный алгоритм не предназначен для объявления переменной.

6.3.13.2 Параметры

Наименование	Обозначение	Тип	Назначение
Вход	Имя	Строковый	Символьное имя переменной
	A	Любой	Присвоение значение с выхода другого алгоблока
Выходы	B	Любой	Значение переменной

6.3.13.3 Описание алгоритма

Нет никаких ограничений на число используемых переменных в одной техпрограмме. Имя может содержать не более 20 любых символов. Переменная может **использоваться** в различных задачах и мнемосхемах одной техпрограммы. Под использованием понимается однократное присваивание значения с выхода другого алгоблока и многократное получение значения на входе других алгоблоков. Таким образом, значение переменной может меняться лишь в одной задаче. Это позволяет избежать неопределенности значения, которая может возникнуть при одновременной модификации переменной из разных задач, работающих асинхронно. Во время компиляции техпрограммы алгоритм проверяет выполнение следующих условий:

- отсутствие связи на входе **Имя**. Имя переменной не может быть изменено во время выполнения. Поэтому нельзя связывать вход, определяющий имя переменной;
- уникальность имени переменной в пределах техпрограммы;
- однократное присваивание значения с выхода другого алгоблока;
- совпадение типов переменных при их использовании с типом объявленной переменной;
- наличие объявленной переменной до ее первого использования. Другими словами, проверяется, чтобы переменная была объявлена раньше, чем она будет впервые использована;
- наличие связей входа и выхода алгоблока Переменная. Такая ситуация не допускается и в общем не имеет смысла.

Невыполнение любого из перечисленных условий приводит к возникновению соответствующих ошибок во время компиляции техпрограммы.

6.3.13.3.1 Присваивание значения переменной

Для того, чтобы присвоить значение переменной с выхода другого алгоблока, следует добавить алгоблок **Переменная** и в Мастере настройки указать те же имя и тип, что при объявлении. Затем следует связать выход другого алгоблока со входом алгоблока **Переменная**. Допускается однократное присваивание значения в пределах одной техпрограммы.

6.3.13.3.2 Получение значения переменной

Для того, чтобы получить значение переменной на входе другого алгоблока, следует добавить алгоблок **Переменная** и в мастере настройки указать те же имя и тип, что при объявлении. Затем следует связать выход алгоблока **Переменная** со входом другого алгоблока.

6.3.13.3.3 Рекомендации

Рекомендуется для объявления переменных сделать отдельную задачу, в техпрограмме и разместить ее раньше остальных задач. Внутри такой задачи можно объявить все необходимые переменные. При этом задача может не выполняться (например, может быть установлено значение **Нет** (0) на входе **Разрешение задачи**), т.к. для объявления переменных выполнение задачи не требуется. Таким образом эта задача не будет отнимать процессорного времени для своего выполнения.

Имп. № подл.	Подп. и дата
Взаим. имп. №	Имп. №
Подп. и дата	Подп. и дата

Изм	Лист	№ докум	Подп.	Дата	ПФДИ.421457.009 И3.6	Лист
						86

6.4 Группа 'Логика'

6.4.1 Не

6.4.1.1 Назначение

Логическое отрицание.

6.4.1.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	A	логический	0	Вход A
Выходы	B	логический	-	Результат

6.4.1.3 Описание алгоритма

Если на входе **Да**, то на выходе **Нет** и наоборот.

6.4.2 И

6.4.2.1 Назначение

Логическое "И".

6.4.2.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	A1	логический	0	Вход A1
	A2	логический	0	Вход A2

	AN	логический	0	Вход AN
Выходы	B	логический	-	Результат

6.4.2.3 Описание алгоритма

Конъюнкция или логическое умножение.

Результат равен **Да** только если все **A1...AN** равны **Да**.

6.4.2.4 Мастер настройки

Число входов **N** задается с помощью Мастера,
при этом: $2 \leq N \leq 100$.

6.4.3 Или

6.4.3.1 Назначение

Логическое "ИЛИ".

6.4.3.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	A1	логический	0	Вход A1
	A2	логический	0	Вход A2

	AN	логический	0	Вход AN
Выходы	B	логический	-	Результат

6.4.3.3 Описание алгоритма

Дизъюнкция или логическое сложение.

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

Результат равен **Нет** только если все **A1...AN** равны **Нет**.

6.4.3.4 Мастер настройки

Число входов **N** задается с помощью Мастера,
при этом: $2 \leq N \leq 100$.

6.4.4 ИСКЛИЛИ

6.4.4.1 Назначение

Исключающее ИЛИ.

6.4.4.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	A1	логический	0	Вход A1
	A2	логический	0	Вход A2

	AN	логический	0	Вход AN
Выходы	B	логический	-	Результат

6.4.4.3 Описание алгоритма

Результат равен **Да**, если число входов со значением **Да** нечетно.

6.4.4.4 Мастер настройки

Число входов **N** задается с помощью Мастера,
при этом: $2 \leq N \leq 100$.

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

6.5 Группа 'Математика'

6.5.1 Слож

6.5.1.1 Назначение

Сложение.

6.5.1.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	A1	Действительный	0	Слагаемое A1
	A2	Действительный	0	Слагаемое A2

	AN	Действительный	0	Слагаемое AN
Выходы	B	Действительный	-	Сумма

6.5.1.3 Описание алгоритма

Складывает числа.

6.5.1.4 Мастер настройки

Количество слагаемых N задается с помощью Мастера, при этом: $2 \leq N \leq 100$.

6.5.2 Выч

6.5.2.1 Назначение

Вычитание.

6.5.2.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	A1	Действительный	0	Уменьшаемое
	A2	Действительный	0	Вычитаемое
Выходы	B	Действительный	-	Разность

6.5.2.3 Описание алгоритма

Вычитает одно число из другого.

6.5.3 Умнож

6.5.3.1 Назначение

Умножение.

6.5.3.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	A1	Действительный	0	Множитель A1
	A2	Действительный	0	Множитель A2

	AN	Действительный	0	Множитель AN
Выходы	B	Действительный	-	Произведение

6.5.3.3 Описание алгоритма

Умножает числа.

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

6.5.3.4 Мастер настройки

Количество множителей N задается с помощью Мастера,
при этом: $2 \leq N \leq 100$.

6.5.4 Делен

6.5.4.1 Назначение

Деление.

6.5.4.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	A1	Действительный	0	Делимое
	A2	Действительный	1	Делитель
Выходы	B	Действительный	-	Частное

6.5.4.3 Описание алгоритма

Делит одно число на другое. Делитель не должен равняться 0.

6.5.5 ДелОст

6.5.5.1 Назначение

Деление с остатком.

6.5.5.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	A1	Действительный	0	Делимое
	A2	Действительный	1	Делитель
Выходы	B	целый	-	Частное
	Ост	Действительный	-	Остаток

6.5.5.3 Описание алгоритма

Делит одно число на другое с остатком.
Делитель не должен равняться 0.

6.5.6 Округл

6.5.6.1 Назначение

Округление.

6.5.6.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Мин. знач.	Макс. знач.	Назначение
Входы	A	Действительный	0	-	-	Округляемое число
	N	целый	0	-100	+100	Точность (кол-во значащих цифр после запятой)
Выходы	B	Действительный	-	-	-	Округленное число

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

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

6.5.6.3 Описание алгоритма

Округляет число с заданной точностью. Если $A2=0$, то округляемое число будет округлено до целых. Если $A2<0$, то округляемое число будет округлено до 10^{A2} .

6.5.7 Степень

6.5.7.1 Назначение

Возведение в степень.

6.5.7.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	A	Действительный	0	Число для возведения в степень
	N	Действительный	0	Степень
Выходы	B	Действительный	-	Число, возведенное в степень

6.5.7.3 Описание алгоритма

Возводит число в степень.

6.5.8 Корень Назначение

Извлечение квадратного корня.

6.5.8.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	A	Действительный	0	Число для извлечения квадратного корня
Выходы	B	Действительный	-	Корень квадратный

6.5.8.3 Описание алгоритма

Извлекает арифметический квадратный корень из числа, где A – неотрицательное число.

6.5.9 $X = -X$ Назначение

Смена знака.

6.5.9.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	A	Действительный	1	Число для смены знака
Выходы	B	Действительный	-	Число с обратным знаком

6.5.9.3 Описание алгоритма

Меняет знак на противоположный у заданного числа.

6.5.10 $1/X$

6.5.10.1 Назначение

Обратная величина.

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

6.5.10.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	A	Действительный	1	Число для обращения
Выходы	B	Действительный	-	Обратная величина

6.5.10.3 Описание алгоритма

Вычисляет обратную величину к заданной, где **A** не равно 0.

6.5.11 10^X

6.5.11.1 Назначение

Обратный десятичный логарифм.

6.5.11.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	A	Действительный	0	Степень числа 10
Выходы	B	Действительный	-	Результат

6.5.11.3 Описание алгоритма

Вычисляет выражение вида “десять в степени **A**”.

6.5.12 E^X

6.5.12.1 Назначение

Обратный натуральный логарифм.

6.5.12.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	A	Действительный	0	Сепень числа e
Выходы	B	Действительный	-	Результат

6.5.12.3 Описание алгоритма

Вычисляет выражение вида “e в степени **A**”.

6.5.13 Lg

6.5.13.1 Назначение

Десятичный логарифм.

6.5.13.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Мин. знач.	Назначение
Входы	A	Действительный	1	0	Вход A
Выходы	B	Действительный	-	-	Результат

6.5.13.3 Описание алгоритма

Вычисляет десятичный логарифм от заданного числа, где **A** – положительное число.

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

6.5.14 Ln

6.5.14.1 Назначение

Натуральный логарифм.

6.5.14.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Мин. знач.	Назначение
Входы	A	Действительный	1	0	Вход A
Выходы	B	Действительный	-	-	Результат

6.5.14.3 Описание алгоритма

Вычисляет натуральный логарифм от заданного числа, где A – положительное число.

6.5.15 Sin Назначение

Синус.

6.5.15.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	A	Действительный	0	Вход A
Выходы	B	Действительный	-	Результат

6.5.15.3 Описание алгоритма

Вычисляет синус от заданного числа.

6.5.16 Cos

6.5.16.1 Назначение

Косинус.

6.5.16.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	A	Действительный	0	A
Выходы	B	Действительный	-	Результат

6.5.16.3 Описание алгоритма

Вычисляет косинус от заданного числа.

6.5.17 Tg Назначение

Тангенс.

6.5.17.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	A	Действительный	0	A
Выходы	B	Действительный	-	Результат

6.5.17.3 Описание алгоритма

Вычисляет тангенс от заданного числа.

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

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

6.5.18 ArcSin Назначение

Арксинус.

6.5.18.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	A	Действительный	0	A
Выходы	B	Действительный	-	Результат

6.5.18.3 Описание алгоритма

Вычисляет арксинус от заданного числа.

6.5.19 ArcCos Назначение

Аркосинус.

6.5.19.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	A	Действительный	0	A
Выходы	B	Действительный	-	Результат

6.5.19.3 Описание алгоритма

Вычисляет аркосинус от заданного числа.

6.5.20 ArcTg

6.5.20.1 Назначение

Арктангенс.

6.5.20.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	A	Действительный	0	A
Выходы	B	Действительный	-	Результат

6.5.20.3 Описание алгоритма

Вычисляет арктангенс от заданного числа.

6.5.21 Факториал

6.5.21.1 Назначение

Вычисление факториала числа.

6.5.21.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Мин. знач.	Макс. знач.	Назначение
Входы	A	целый	0	0	+170	A
Выходы	B	Действительный	-	-	-	Результат

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

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

6.5.21.3 Описание алгоритма

Вычисляет произведение всех натуральных чисел до n включительно, где $n = A$.

$$n! = 1 \cdot 2 \cdot \dots \cdot n = \prod_{i=1}^n i$$

Формула вычисления:

По определению полагают $0! = 1$. Факториал определён только для целых неотрицательных чисел, иначе - ошибка.

6.5.22 Модуль

6.5.22.1 Назначение

Вычисление абсолютного значения (модуля) вещественного числа.

6.5.22.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Мин. знач.	Макс. знач.	Назначение
Входы	A	Действительный	0	-	-	Число
Выходы	B	Действительный	-	-	-	Модуль числа

6.5.22.3 Описание алгоритма

Абсолютная величина вещественного числа A есть неотрицательное число, обозначаемое $|A|$ и определяемое следующим образом:

если $A > 0$, то $|A| = A$;

если $A \leq 0$, то $|A| = -A$.

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

6.6 Группа 'Время'

6.6.1 Сейчас

6.6.1.1 Назначение

Алгоритм предназначен для получения упакованного значения текущей даты и времени.

6.6.1.2 Параметры

Наименование	Обознач.	Тип	Номер	Нач. знач.	Назначение
Выход	<i>Вр.</i>	Дата и время	1		Упакованное значение текущей даты и времени

6.6.1.3 Описание алгоритма

Алгоритм определяет значение текущей даты и времени и заносит его в формате **Дата-время** на выход **Вр.**

6.6.2 УпакВр

6.6.2.1 Назначение

Алгоритм предназначен для получения упакованного значения даты и времени.

6.6.2.2 Параметры

Наименование	Обознач.	Тип	Номер	Нач. знач.	Назначение
Входы	<i>Год</i>	Целый	1	2005	Год
	<i>Мес.</i>	Целый	2	1	Месяц
	<i>День</i>	Целый	3	1	День
	<i>Час</i>	Целый	4	0	Час
	<i>Мин.</i>	Целый	5	0	Минута
	<i>Сек.</i>	Целый	6	0	Секунда
	<i>Мсек.</i>	Целый	7	0	Миллисекунда
	<i>З(Л)</i>	Логический	8	Нет	Признак отнесения заданного времени к зимнему или летнему периоду
Выходы	<i>Вр.</i>	Дата и время	1		Упакованное значение даты и времени
	<i>ДД</i>	Логический	2		Достоверность упакованного значения

6.6.2.3 Описание алгоритма

Алгоритм вычисляет значение даты и времени согласно значениям входов **Год, Мес., День, Час, Мин., Сек., Мсек. и З(Л)** и заносит его в формате **Дата-время** на выход **Вр.** Если значение даты и времени корректно, на выход **ДД** заносится значение **Да** (true), в противном случае – **Нет** (false).

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

6.6.3 РаспакВр

6.6.3.1 Назначение

Алгоритм предназначен для получения распакованного значения даты и времени.

6.6.3.2 Параметры

Наименование	Обознач.	Тип	Номер	Нач. знач.	Назначение
Вход	<i>Вр.</i>	Дата и время	1	0	Упакованное значение даты и времени
Выходы	<i>Год</i>	Целый	1		Год
	<i>Мес.</i>	Целый	2		Месяц
	<i>День</i>	Целый	3		День
	<i>Час</i>	Целый	4		Час
	<i>Мин.</i>	Целый	5		Минута
	<i>Сек.</i>	Целый	6		Секунда
	<i>Мсек.</i>	Целый	7		Миллисекунда
	<i>З(Л)</i>	Логический	8		Зимнее или летнее время

6.6.3.3 Описание алгоритма

Алгоритм распаковывает значение входа **Вр.** на выходы **Год, Мес., День, Час, Мин., Сек., Мсек. и З(Л).**

6.6.4 УпақДлит

6.6.4.1 Назначение

Алгоритм предназначен для получения упакованного значения интервала времени.

6.6.4.2 Параметры

Наименование	Обознач.	Тип	Номер	Нач. знач.	Назначение
Входы	<i>Знак</i>	Целый	1	1	Знак числа, выражающего значение интервала
	<i>Дни</i>	Целый	2	0	Количество дней
	<i>Часы</i>	Целый	3	0	Количество часов
	<i>Мин.</i>	Целый	4	0	Количество минут
	<i>Сек.</i>	Целый	5	0	Количество секунд
	<i>МСек.</i>	Целый	6	0	Количество миллисекунд
Выход	<i>Длит.</i>	Длительность	1		Упакованное значение интервала времени

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

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

6.6.4.3 Описание алгоритма

Алгоритм вычисляет значение интервала времени согласно значениям входов **Дни, Часы, Мин., Сек. и МСек.** и заносит его на выход **Длит.** Если значение на входе **Знак** = 1, интервал времени будет положительным, в противном случае – отрицательным.

6.6.5 РаспакДлит

6.6.5.1 Назначение

Алгоритм предназначен для получения распакованного значения интервала времени.

6.6.5.2 Параметры

Наименование	Обознач.	Тип	Номер	Нач. знач.	Назначение
Вход	<i>Длит.</i>	Длительность	1	0	Упакованное значение интервала времени
Выходы	<i>Знак</i>	Строковый	1		Знак
	<i>Дни</i>	Целый	2		Количество дней
	<i>Часы</i>	Целый	3		Количество часов
	<i>Мин.</i>	Целый	4		Количество минут
	<i>Сек.</i>	Целый	5		Количество секунд
	<i>МСек.</i>	Целый	6		Количество миллисекунд

6.6.5.3 Описание алгоритма

Алгоритм распаковывает значение входа **Длит.** на выходы **Дни, Часы, Мин., Сек. и МСек.** Если интервал времени положительный, на выход **Знак** заносится число 1, в противном случае – число -1.

6.6.6 СмещВрНазначение

Алгоритм предназначен для получения смещения во времени.

6.6.6.2 Параметры

Наименование	Обознач.	Тип	Номер	Нач. знач.	Назначение
Входы	<i>НачВр.</i>	Дата и время	1	0	Упакованное значение даты и времени
	<i>Длит.</i>	Длительность	2	0	Упакованное значение интервала времени
Выход	<i>КонВр.</i>	Дата и время	1		Упакованное значение даты и времени со смещением на величину интервала

6.6.6.3 Описание алгоритма

Алгоритм вычисляет смещение во времени согласно значениям на входах **НачВр.** и **Длит.** и заносит результат на выход **КонВр.** Смещение по времени вперед или назад зависит от знака значения на входе **Длит.**

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

6.6.7 ДлитВрНазначение

Алгоритм предназначен для получения разницы между двумя значениями даты и времени.

6.6.7.2 Параметры

Наименование	Обознач.	Тип	Номер	Нач. знач.	Назначение
Входы	<i>Вр1.</i>	Дата и время	1	0	Упакованное значение даты и времени
	<i>Вр2.</i>	Дата и время	2	0	Упакованное значение даты и времени
Выход	<i>Длит.</i>	Длительность	1		Упакованное значение интервала времени

6.6.7.3 Описание алгоритма

Алгоритм вычисляет разницу между значениями на входах **Вр1.** и **Вр2.** и заносит результат на выход **Длит.**

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

6.7 Группа 'ТАУ'

6.7.1 Задер

6.7.1.1 Назначение

Алгоритм предназначен для реализации задержки сигнала с плавающей запятой на произвольный момент времени, вычисляемого в абсолютных или относительных единицах.

6.7.1.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	X	действительный	0	Входной сигнал
	N	целый	5	Длина очереди задержки
	T	модельное время	2	Время задержки
Выходы	Y	событийный	-	Выходной сигнал

6.7.1.3 Описание алгоритма

Алгоритм осуществляет задержку появления значения сигнала X на выходе Y . Время задержки сигнала T может быть абсолютным (астрономическим интервалом времени) или относительным (относительно времени выполнения задачи). Размер очереди задержки N – определяет точность повторения выходного сигнала по отношению к входному. Чем длиннее очередь, тем больше точность повторения, но тем больше памяти будет потрачено на ее хранение и наоборот. Вообще говоря для того, чтобы входной сигнал был задержан без потери точности, длина очереди должна вычисляться по формуле:

$$N = T / T_{ц}, \text{ где } T_{ц} \text{ – время цикла задачи.}$$

Для пояснения, на рисунке 54 приведен график задержки сигнала с потерей точности, на рисунке 55 – без потери точности. График 1 – исходный сигнал, график 2 – задержанный на 2000 мс. Время цикла задачи ($T_{ц}$) – 100 мс.



Длина очереди – 5

Рисунок 54 – График задержки сигнала с потерей точности



Длина очереди – 20

Рисунок 55 - График задержки сигнала без потери точности

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

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

6.7.2 Интегр

6.7.2.1 Назначение

Алгоритм предназначен для интегрирования сигнала с возможностью установки начального значения.

6.7.2.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	X_n	действительный	0	Начальное значение выходного сигнала
	C_n	Логический	Нет	Установка начального значения
	X_v	действительный	0	Входной сигнал
	T_i	модельное время	2	Время интегрирования
Выходы	Y	событийный	-	Выходной сигнал

6.7.2.3 Описание алгоритма

Алгоритм осуществляет численное интегрирование входного сигнала X_v на заданном интервале времени T_i . Для установки начального значения выходного сигнала Y предназначены входы X_n и C_n . Когда C_n меняет значение с Лог 0 (Нет) на Лог 1 (Да) (передний фронт), значение на выходе приравнивается значению X_n и интегрирование продолжается с учетом этого значения. Алгоритм реализует передаточную функцию, вида:

$$W(P) = 1 / T_i * P.$$

6.7.3 Диффер

6.7.3.1 Назначение

Алгоритм предназначен для дифференцирования сигнала с возможностью установки начального значения.

6.7.3.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	X_n	действительный	0	Начальное значение выходного сигнала
	C_n	Логический	Нет	Установка начального значения
	X_v	действительный	0	Входной сигнал
	T_d	модельное время	2	Время дифференцирования
Выходы	Y	событийный	-	Выходной сигнал

6.7.3.3 Описание алгоритма

Алгоритм осуществляет численное дифференцирование входного сигнала X_v на заданном интервале времени T_d . Для установки начального значения выходного сигнала Y предназначены входы X_n и C_n . Когда C_n меняет значение с Лог 0 (Нет) на Лог 1 (Да) (передний фронт), значение на выходе приравнивается значению X_n и дифференцирование продолжается с учетом этого значения. Алгоритм реализует передаточную функцию, вида:

$$W(P) = T_d * P / (T_d * P + 1).$$

6.7.4 ИнЗвено1

6.7.4.1 Назначение

Алгоритм предназначен для реализации модели инерционного звена первого порядка с возможностью установки начального значения.

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

Изм	Лист	№ докум	Подп.	Дата	ПФДИ.421457.009 И3.6	Лист
						102

6.7.4.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	X_n	действительный	0	Начальное значение выходного сигнала
	C_n	Логический	Нет	Установка начального значения
	X_v	действительный	0	Входной сигнал
	T_i	модельное время	2	Время инерционности
Выходы	Y	событийный	-	Выходной сигнал

6.7.4.3 Описание алгоритма

Алгоритм моделирует инерционное звено первого порядка, рассчитывая инерционность входного сигнала X_v со временем инерционности T_i . Для установки начального значения выходного сигнала Y предназначены входы X_n и C_n . Когда C_n меняет значение с Лог 0 (Нет) на Лог 1 (Да) (передний фронт), значение на выходе приравнивается значению X_n и расчет инерционности продолжается с учетом этого значения. Алгоритм реализует передаточную функцию, вида:

$$W(P) = 1 / (T_i * P + 1) .$$

6.7.5 ИнЗвено2

6.7.5.1 Назначение

Алгоритм предназначен для реализации модели инерционного звена второго порядка (колебательное звено) с возможностью установки начального значения.

6.7.5.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	X_n	действительный	0	Начальное значение выходного сигнала
	C_n	Логический	Нет	Установка начального значения
	X_v	действительный	0	Входной сигнал
	T_{i1}	модельное время	2	Время инерционности
	T_{i2}	модельное время	2	Время инерционности
Выходы	Y	событийный	-	Выходной сигнал

6.7.5.3 Описание алгоритма

Алгоритм моделирует инерционное звено второго порядка, рассчитывая инерционность входного сигнала X_v со временами инерционности T_{i1} и T_{i2} . Для установки начального значения выходного сигнала Y предназначены входы X_n и C_n . Когда C_n меняет значение с Лог 0 (Нет) на Лог 1 (Да) (передний фронт), значение на выходе приравнивается значению X_n и расчет инерционности продолжается с учетом этого значения. Алгоритм реализует передаточную функцию, вида:

$$W(P) = 1 / ((T_{i1} * P + 1) * (T_{i2} * P + 1)) .$$

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

6.8 Группа 'Моделирование'

6.8.1 Виконт Назначение

Алгоритм предназначен для управления и связи с одним виртуальным контроллером (ВК) Квинта. С его помощью можно запустить ВК с заданным номером, управлять ходом выполнения его техпрограммы, сохранять и восстанавливать различные срезы техпрограммы и данных, задавать температуру «каркаса».

6.8.1.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	<i>Пауза</i>	событийный	-	Требование остановки выполнения техпрограммы ВК
	<i>Пуск</i>	событийный	-	Требование запуска выполнения техпрограммы ВК
	<i>Шаг</i>	событийный	-	Требование пройти 1 цикл техпрограммы ВК и остановиться
	<i>Установить</i>	событийный	-	Требование установки значения температуры «каркаса»
	<i>T, °C</i>	целый	25	Температура «каркаса», в °C
	<i>Имя среза</i>	строковый	-	Имя сохраняемого или восстанавливаемого среза техпрограммы и данных
	<i>Сохранить</i>	событийный	-	Требование сохранить срез контроллера с именем « <i>имя среза</i> »
	<i>Восстановить</i>	событийный	-	Требование восстановить срез контроллера с именем « <i>имя среза</i> »
	<i>A1...A16</i>	контейнер данных	Нет	Связь с алгоблоками имитации УСО
Выходы	<i>Перезапуск</i>	событийный	-	Событие перезапуска контроллера
	<i>Запущен</i>	логический	-	Признак того, что техпрограмма контроллера выполняется и с ним есть связь по сети
	<i>Ав. вых.</i>	логический	-	Признак того, что контроллер будет завершать свою работу при отсутствии связи со шлюзом более, чем 1 минуту

6.8.1.3 Описание алгоритма

После добавления алгоблока в задачу, его надлежит настроить при помощи Мастера. В результате в его заголовке будут отображаться основные сведения о конфигурации.

Алгоблок может работать в двух основных режимах: с управлением ВК и без управления. В первом случае при первом выполнении алгоблока будет произведен запуск ядра ВК, а при останове сервера выполнения – останов ВК. Во втором случае – запуск ядра не произойдет. И в том и в другом случаях алгоблок будет пытаться поддерживать связь с ВК, номер которого задан мастером настройки (и отображается в заголовке алгоблока). Интенсивность связи с ВК определяется временем выполнения задачи, но не превышает 1 обмена за 50 мс.

Посредством алгоблока можно управлять ВК в отладочном режиме. Для остановки выполнения технологической программы (например, для детального исследования какого-то промежуточного состояния) можно сформировать событие на входе **Пауза**, после чего техпрограмма ВК перестанет выполняться, а значения всех сигналов будут заморожены на неопределенно долгое время. При этом связь со всеми Рабочими станциями, а также некоторые возможности управления (например, подача объектных команд) сохранятся. В режиме паузы ВК, можно выполнять технологическую программу по шагам, при этом одним шагом является однократный проход технологической программы от первого до последнего алгоблока. Для выполнения одного шага, следует сформировать событие на входе **Шаг**. Если ВК выполнял техпрограмму (т.е. не находился в режиме паузы), формирование *первого* события на входе **Шаг** приведет к переводу контроллера в режим паузы. Все последующие события на входе **Шаг** будут выполнять техпрограмму ВК по шагам. Для

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

Изм	Лист	№ докум	Подп.	Дата	ПФДИ.421457.009 ИЗ.6	Лист
						104

запуска техпрограммы из паузы, следует сформировать событие на входе **Пуск**. Для отладки техпрограмм, в которых важно изменять температуру каркаса контроллера, можно устанавливать температуру в «каркасе» от 0 до 99 °С. Для этого следует сформировать событие на входе **Установить** и значение температуры в ВК поменяется на то, которое было задано на входе **Т,°С**.

В ходе проведения отладочных испытаний техпрограмм, а также для целей создания модели технологического объекта, могут быть случаи, когда необходимо запомнить какое-то промежуточное состояние контроллера, для того, чтобы можно было к нему вернуться в любой момент времени. Такое промежуточное состояние принято называть срезом (или фреймом, от англ. слова frame – кадр). Оно включает в себя данные о техпрограмме и текущие значения всех сигналов ВК, а также некоторых служебных параметров (например, температуру «каркаса»). ВК позволяет сохранять неограниченное число именованных срезов. Для этого следует на входе **Имя среза** задать какое-нибудь название среза (желательно, чтобы оно было осмысленным) и сформировать событие на входе **Сохранить**. При этом ВК сохранит срез на жестком диске в виде файла. Если срез с таким названием существовал, он будет перезаписан без предупреждения. Для восстановления среза следует также задать его название на входе **Имя среза** и сформировать событие на входе **Восстановить**.

Алгоблок позволяет подключить до 16 любых алгоблоков имитации УСО. Для этого используются входы **А1...А16**. Обмен сигналами с УСО ВК происходит один раз в цикл задачи, но не чаще, чем раз в 50 мс.

Если ядро ВК по тем или иным причинам перестает выполняться – оно будет перезапущено алгоблоком. При этом число перезапусков ВК не может быть больше того, которое было задано в Мастере настройки. При перезапуске ВК на выходе **Перезапуск** будет сформировано событие. На выходе **Запущен** формируется признак того, что техпрограмма ВК выполняется (т.е. ВК не в режиме паузы) и с ним есть связь. Если ВК переведен в режим паузы или с ним не было связи более 2 секунд, на выходе **Запущен** появится логический ноль (Нет). Если в Мастере настройки указать, что ВК не должен выключаться при потере связи со шлюзом, то на выходе **Ав. вых.** появится логический ноль, иначе – логическая 1.

6.8.1.4 Мастер настройки

6.8.1.4.1 Группа «Управление»

В данной группе можно настроить управление ВК.

- флаг **Управлять ВК**. Если установлен – алгоблок будет запускать и перезапускать (в случае его останова) ядро ВК. Иначе – алгоблок будет пытаться установить связь с уже запущенным ВК;
- флаг **Делать консоль видимой**. Если установлен – консоль ядра ВК будет видна и доступна;
- флаг **Не останавливать при отсутствии связи**. Если установлен – ядро ВК не будет самостоятельно выгружаться при отсутствии связи со шлюзом более 1 минуты;
- флаг **Запускать под отладкой**. Предназначен для сервисного инжиниринга;
- цифровое поле **Перезапускать не более раз**. Позволяет задать число перезапусков ВК, в случае, если его ядро по тем или иным причинам перестает выполняться.

6.8.1.4.2 Группа «Адрес»

В данной группе можно настроить адрес ВК, с которым алгоблок будет поддерживать связь. Если алгоблок одновременно управляет ВК (установлен флаг **Управлять ВК**), то он запустит ядро ВК и сконфигурирует его для работы с указанным номером системного модуля, и собственным К-номером.

- цифровое поле **Системный модуль**. Задаёт К-номер системного модуля (шлюза) с которым должен быть связан ВК
- цифровое поле **Контроллер**. Задаёт собственный К-номер ВК.

Примечание - Номера системного модуля и контроллера сверяются с Базой данных проекта при нажатии кнопки Ок в Мастере настройки. Если таких номеров в базе нет – появится соответствующее предупреждение и Мастер вновь предложит задать номера.

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

Изм	Лист	№ докум	Подп.	Дата	ПФДИ.421457.009 И3.6	Лист
						105

6.8.1.4.3 Группа «Сети»

В данной группе можно настроить номера сетевых адаптеров, с которыми будет работать ВК. Для того, чтобы ВК смог общаться по сети – на том компьютере, где он будет запускаться (там где находится сервер выполнения, исполняющий техпрограмму Мезона, которая содержит данный алгоблок), должен быть установлен сетевой протокол **NetBEUI**.

- цифровое поле **1**. Задаёт номер первого сетевого адаптера. Если эта сеть не должна использоваться – выберите пункт **Нет сети**
- цифровое поле **2**. Задаёт номер второго сетевого адаптера. Если эта сеть не должна использоваться – выберите пункт **Нет сети**.

ВК поддерживает работу с двумя параллельными сетями (как и его реальный прототип – Р380), но может быть настроен и для работы по одной сети. При этом один из адаптеров может быть не выбран (пункт **Нет сети**). Если алгоблок управляет запуском ВК (флаг **Управлять ВК** установлен), то должна быть выбрана хотя бы одна сеть. Номера сетей двух адаптеров должны различаться. В противном случае появится соответствующее предупреждение и мастер вновь предложит задать номера.

При выборе номера сетевого адаптера – появится список в котором указаны пункты **Нет сети**, **Рабочая 1 ... Рабочая N**, **Возможная 1 ... Возможная N**. Если список не содержит пункты **Рабочая X** - значит на том компьютере, где запущен данный мастер настройки, не установлен протокол NetBEUI. Наличие пунктов **Возможная X** позволяет настроить алгоблок для работы на удаленном сервере выполнения, сетевые номера адаптеров, которого могут отличаться от тех, что есть на компьютере, на которой подготавливается техпрограмма Мезона. В любом случае, если номера адаптеров того компьютера известны, то следует на нём запустить Мезон, выбрать нужную техпрограмму, вызвать Мастера настройки алгоблока и выбрать сетевые адаптеры. В дальнейшем, когда номера адаптеров этого компьютера станут известны, задать их номера можно будет на любом компьютере.

6.8.2 АЦП

6.8.2.1 Назначение

Алгоритм предназначен для имитации одного модуля АЦП-60-1 или АЦП-60-2 (8 и 16 каналов, соответственно).

6.8.2.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	<i>A_i</i>	действительный	0	Значение аналогового сигнала в физических единицах
Выходы	<i>ВК</i>	контейнер данных	-	Связь с алгоблоком Виконт
	<i>Статус</i>	логический	-	Признак того, что данное УСО обслуживается ВК

6.8.2.3 Описание алгоритма

При добавлении алгоблока в задачу появляется мастер, позволяющий выбрать тип АЦП (8 или 16 канальный). Изменить канальность алгоблока можно также и после добавления его в задачу. При этом, если число выводов алгоблока, после изменения его канальности уменьшается – все связи с удаляемыми выводами будут автоматически оборваны (для этого нужно утвердительно ответить на вопрос об автоматическом обрыве связей). Для того, чтобы входные аналоговые сигналы из физического формата преобразовать в процентный (принятый в контроллере), следует настроить шкалы преобразования сигналов по отдельности для каждого канала. Шкалу того или иного сигнала можно узнать в приложении **Аркада**.

6.8.2.4 Мастер настройки

Для изменения числа каналов алгоблока, находящегося в задаче, следует вызвать Мастера настройки, щелкнув на заголовке алгоблока. Для задания значения верхней и нижней шкал в отдельном канале алгоблока, следует вызвать Мастера, щелкнув на требуемом канале.

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

Изм	Лист	№ докум	Подп.	Дата	ПФДИ.421457.009 ИЗ.6	Лист
						106

6.8.3 ЦАП

6.8.3.1 Назначение

Алгоритм предназначен для имитации одного модуля ЦАП-60-1 или ЦАП-60-2 (8 и 16 каналов, соответственно).

6.8.3.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Выходы	<i>ВК</i>	контейнер данных	-	Связь с алгоблоком Виконт
	<i>Статус</i>	логический	-	Признак того, что данное УСО обслуживается ВК
	<i>Вi</i>	действительный	-	Значение аналогового сигнала в процентах ($\pm 199,998\%$)

6.8.3.3 Описание алгоритма

При добавлении алгоблока в задачу появляется Мастер, позволяющий выбрать тип ЦАП (8 или 16 канальный). Изменить канальность алгоблока можно также и после добавления его в задачу. При этом, если число выводов алгоблока, после изменения его канальности уменьшается – все связи с удаляемыми выводами будут автоматически оборваны (для этого нужно утвердительно ответить на вопрос об автоматическом обрыве связей).

При потере связи с ВК на время более 2 секунд на выходе **Статус** появляется значение Лог 0 (Нет), а все выходные значения обнуляются.

6.8.3.4 Мастер настройки

Для изменения числа каналов алгоблока, находящегося в задаче, следует вызвать мастер настройки, щелкнув на заголовке алгоблока.

6.8.4 ДЦП

6.8.4.1 Назначение

Алгоритм предназначен для имитации одного модуля ДЦП-60-1, ДЦП-60-2, ДЦП-60-3 или ДЦП-60-4 (16, 32, 48, 64 канала, соответственно).

6.8.4.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	<i>Ai</i>	логический	Нет	Значение дискретного сигнала
Выходы	<i>ВК</i>	контейнер данных	-	Связь с алгоблоком Виконт
	<i>Статус</i>	логический	-	Признак того, что данное УСО обслуживается ВК

6.8.4.3 Описание алгоритма

При добавлении алгоблока в задачу появляется мастер, позволяющий выбрать тип ДЦП (16, 32, 48, 64 канальный). Изменить канальность алгоблока можно также и после добавления его в задачу. При этом, если число выводов алгоблока, после изменения его канальности уменьшается – все связи с удаляемыми выводами будут автоматически оборваны (для этого нужно утвердительно ответить на вопрос об автоматическом обрыве связей).

6.8.4.4 Мастер настройки

Для изменения числа каналов алгоблока, находящегося в задаче, следует вызвать мастер настройки, щелкнув на заголовке алгоблока.

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

6.8.5 ЦДП

6.8.5.1 Назначение

Алгоритм предназначен для имитации одного модуля ЦДП-60-1, ЦДП -60-2, ЦДП -60-3 или ЦДП -60-4 (16, 32, 48, 64 канала, соответственно).

6.8.5.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Выходы	<i>ВК</i>	контейнер данных	-	Связь с алгоблоком Виконт
	<i>Статус</i>	логический	-	Признак того, что данное УСО обслуживается ВК
	<i>Vi</i>	логический	-	Значение дискретного сигнала

6.8.5.3 Описание алгоритма

При добавлении алгоблока в задачу появляется мастер, позволяющий выбрать тип ЦДП (16, 32, 48, 64 канальный). Изменить канальность алгоблока можно также и после добавления его в задачу. При этом, если число выводов алгоблока, после изменения его канальности уменьшается – все связи с удаляемыми выводами будут автоматически оборваны (для этого нужно утвердительно ответить на вопрос об автоматическом обрыве связей).

При потере связи с ВК на время более 2 секунд на выходе **Статус** появляется значение Лог 0 (Нет), а все выходные значения обнуляются.

6.8.5.4 Мастер настройки

Для изменения числа каналов алгоблока, находящегося в задаче, следует вызвать мастер настройки, щелкнув на заголовке алгоблока.

6.8.6 ДВВ

6.8.6.1 Назначение

Алгоритм предназначен для имитации одного модуля ДВВ-60-1 или ДВВ-60-2 (16, 32 канала, соответственно).

6.8.6.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	<i>Ai</i>	логический	Нет	Значение входного дискретного сигнала
Выходы	<i>ВК</i>	контейнер данных	-	Связь с алгоблоком Виконт
	<i>Статус</i>	логический	-	Признак того, что данное УСО обслуживается ВК
	<i>Vi</i>	логический	-	Значение выходного дискретного сигнала

6.8.6.3 Описание алгоритма

При добавлении алгоблока в задачу появляется мастер, позволяющий выбрать тип ДВВ (16, 32 канальный). Изменить канальность алгоблока можно также и после добавления его в задачу. При этом, если число выводов алгоблока, после изменения его канальности уменьшается – все связи с удаляемыми выводами будут автоматически оборваны (для этого нужно утвердительно ответить на вопрос об автоматическом обрыве связей).

При потере связи с ВК на время более 2 секунд на выходе *Статус* появляется значение Лог 0 (Нет), а все выходные значения обнуляются.

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

6.8.6.4 Мастер настройки

Для изменения числа каналов алгоблока, находящегося в задаче, следует вызвать мастер настройки, щелкнув на заголовке алгоблока.

6.8.7 ЦИП

6.8.7.1 Назначение

Алгоритм предназначен для имитации одного модуля ЦИП-60-1 или ЦИП-60-2 (10, 20 каналов, соответственно).

6.8.7.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Выходы	<i>ВК</i>	контейнер данных	-	Связь с алгоблоком Виконт
	<i>Статус</i>	логический	-	Признак того, что данное УСО обслуживается ВК
	<i>Vi</i>	целый	-	Значение сигнала скорости (-128, +127)

6.8.7.3 Описание алгоритма

При добавлении алгоблока в задачу появляется мастер, позволяющий выбрать тип ЦИП (10, 20 канальный). Изменить канальность алгоблока можно также и после добавления его в задачу. При этом, если число выводов алгоблока, после изменения его канальности уменьшается – все связи с удаляемыми выводами будут автоматически оборваны (для этого нужно утвердительно ответить на вопрос об автоматическом обрыве связей).

На входах алгоритма исполнительного механизма должно быть задано время Тим перемещения рабочего органа механизма из одного крайнего положения в другое. Приняв значение скорости V от контроллера, алгоритм исполнительного механизма в текущем цикле работы рассчитывает прибавку:

$$\Delta Y = 100\% * (V/128) * (T_0 / T_{им}), \text{ где } T_0 \text{ – время предыдущего цикла задачи.}$$

Затем полученное значение ΔY добавляется к выходу Y положения исполнительного механизма.

При потере связи с ВК на время более 2 секунд на выходе *Статус* появляется значение Лог 0 (Нет), а все выходные значения обнуляются.

6.8.7.4 Мастер настройки

Для изменения числа каналов алгоблока, находящегося в задаче, следует вызвать мастер настройки, щелкнув на заголовке алгоблока.

6.8.8 АВВ

6.8.8.1 Назначение

Алгоритм предназначен для имитации одного модуля АВВ-61 (4 канала).

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

6.8.8.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	A_i	действительный	0	Значение входного аналогового сигнала в физических единицах
Выходы	BK	контейнер данных	-	Связь с алгоблоком Виконт
	$Статус$	логический	-	Признак того, что данное УСО обслуживается ВК
	B_i	действительный	-	Значение аналогового сигнала в процентах ($\pm 199,998\%$)

6.8.8.3 Описание алгоритма

Для того, чтобы входные аналоговые сигналы из физического формата преобразовать в процентный (принятый в контроллере), следует настроить шкалы преобразования сигналов по отдельности для каждого канала. Шкалу того или иного сигнала можно узнать в приложении **Аркада**.

При потере связи с ВК на время более 2 секунд на выходе **Статус** появляется значение Лог 0 (Нет), а все выходные значения обнуляются.

6.8.9 МЗТ

6.8.9.1 Назначение

Алгоритм предназначен для имитации одного модуля МЗТ-61-1 и одного модуля МИН-61 (3 следящих датчика частоты).

6.8.9.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	F	действительный	0	Значение частоты вращения турбины (0...4000 об/мин)
	F_n	действительный	0	Предупредительная уставка частоты (0...4000 об/мин)
	F_a	действительный	0	Аварийная уставка частоты (0...4000 об/мин)
	<i>Датчик 1</i>	целый	0	Код неисправности датчика
	<i>Датчик 2</i>	целый	0	Код неисправности датчика
	<i>Датчик 3</i>	целый	0	Код неисправности датчика
	<i>Модуль</i>	целый	0	Код неисправности модуля
Выходы	BK	контейнер данных	-	Связь с алгоблоком Виконт
	$Статус$	логический	-	Признак того, что данное УСО обслуживается ВК
	$Защита$	логический	-	Сигнал «включить защиту»

6.8.9.3 Описание алгоритма

На вход **F** поступает значение частоты вращения турбины, в пределах 0...4000 об/мин. Оно сравнивается с предупредительным и аварийным значениями уставок. При значении частоты на входе **F** больше значения на входе **F_a** на выходе **Защита** будет сформирована Лог 1 (Да), которая может использоваться в модели объекта. Сброс Лог1 на этом выходе может быть осуществлен только при значении **F** равном 0.

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

Алгоблок позволяет имитировать основные аварийные ситуации по каждому датчику отдельно, а также по модулю УСО в целом. Аварийные ситуации для датчиков задаются кодами ситуации на входах **Датчик i**, где *i* – номер датчика. Поддерживаются следующие коды:

- 0 – датчик исправен;
- 1 – нет питания датчика;
- 2 – верхняя недостоверность показания датчика.

Аварийные ситуации для модуля в целом, задаются кодами на входе **Модуль**. Поддерживаются следующие коды:

- 0 – модуль исправен;
- 1 – отказ эталонной частоты;
- 2 – нет основного питания;
- 3 – нет резервного питания.

По технологическим причинам в реальном контроллере, модуль МЗТ занимает 4 позиции в каркасе: одну позицию, на которой он собственно установлен, а также еще 3 позиции, следующие за ним. По этой причине не имеет смысла (хотя и возможно) задействовать следующие 3 позиции в алгоблоке Виконт.

6.8.10 МЧТ

6.8.10.1 Назначение

Алгоритм предназначен для имитации одного модуля МЧТ-61-1 (3 следящих датчика частоты).

6.8.10.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	<i>F</i>	действительный	0	Значение частоты вращения турбины (0...4000 об/мин)
	<i>Датчик 1</i>	целый	0	Код неисправности датчика
	<i>Датчик 2</i>	целый	0	Код неисправности датчика
	<i>Датчик 3</i>	целый	0	Код неисправности датчика
	<i>Модуль</i>	логический	Нет	Признак отказа эталонной частоты
Выходы	<i>ВК</i>	контейнер данных	-	Связь с алгоблоком Виконт
	<i>Статус</i>	логический	-	Признак того, что данное УСО обслуживается ВК

6.8.10.3 Описание алгоритма

На вход **F** поступает контролируемое значение частоты вращения турбины, в пределах 0...4000 об/мин.

Алгоблок позволяет имитировать основные аварийные ситуации по каждому датчику отдельно, а также по модулю УСО в целом. Аварийные ситуации для датчиков задаются кодами ситуации на входах **Датчик i**, где *i* – номер датчика. Поддерживаются следующие коды:

- 0 – датчик исправен;
- 1 – нет питания датчика;
- 2 – верхняя недостоверность показания датчика.

Аварийные ситуации для модуля в целом, задаются кодами на входе **Модуль**. Поддерживаются следующие коды:

- 0 – модуль исправен;
- 1 – отказ эталонной частоты.

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

6.8.11 Задвижка

6.8.11.1 Назначение

Алгоритм предназначен для имитации запорной арматуры с потенциальным или импульсным управлением. С его помощью можно имитировать поведение задвижек с промежуточным положением запорного механизма.

6.8.11.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	<i>Открыть</i>	логический или событийный	0	Потенциальная или импульсная команда открытия задвижки
	<i>Закрыть</i>	логический или событийный	0	Потенциальная или импульсная команда закрытия задвижки
	<i>Стоп</i>	событийный	0	Импульсная команда останова задвижке в среднем положении
	<i>Tхода</i>	модельное время	2	Относительное или абсолютное время хода запорного механизма между концевыми датчиками, мс
	<i>Tхко</i>	модельное время	2	Относительное или абсолютное время схода или наезда на концевой датчик открытия, мс
	<i>Tхкз</i>	модельное время	2	Относительное или абсолютное время схода или наезда на концевой датчик закрытия, мс
	<i>Имитатор</i>	контейнер данных	-	Вход подключения алгоблока имитации аварийных состояний
Выходы	<i>SQоткр</i>	логический	-	Концевой выключатель на открытие
	<i>SQзакр</i>	логический	-	Концевой выключатель на закрытие
	<i>ХодОткр</i>	логический	-	Подтверждение хода задвижки на открытие
	<i>ХодЗакр</i>	логический	-	Подтверждение хода задвижки на закрытие
	<i>Uвыс</i>	логический	-	Признак наличия высокого напряжения обмоток электропривода
	<i>Uпус</i>	логический	-	Признак наличия питающего напряжения обмоток пускателя
	<i>% откр</i>	действительный	-	Положение рабочего органа, в % (0 – закрыт, 100 – открыт)

6.8.11.3 Описание алгоритма

Алгоритм может работать в одном из двух режимов управления: потенциальном и импульсном.

В потенциальном режиме ход рабочего механизма происходит по наличию соответствующей команды на входах **Открыть** и **Закрыть**. Запорный механизм останавливается при пропадании любого из потенциальных сигналов управления, либо при равенстве этих сигналов (наличие одновременно двух сигналов Лог 1 (Да) на входах **Открыть** и **Закрыть** расценивается как отсутствие сигналов).

В импульсном режиме – рабочий механизм управляется по наличию событий (импульсов) на входах **Открыть**, **Закрыть**, **Стоп**. При появлении события на входе **Открыть** или **Закрыть** – процент открытия задвижки увеличивается или уменьшается, соответственно, до тех пор, пока не достигнет граничного значения (100 или 0 %, соответственно) или пока не появится событие на входе **Стоп**.

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

Задать время хода между концевыми датчиками можно, установив константу **Тхода**. За это время рабочий механизм задвижки должен проделать путь между 0,2% ... 99,8 % (т.е. между концевыми). Времена движения рабочего механизма на концевых контактах при открытии и закрытии, можно задавать константами **Тхко**, **Тхкз**. Эти времена определяют продолжительность перемещения задвижки в течение которого соответствующий концевой выключатель будет выдавать значение Лог 1 (Да) до того, как запорный механизм сойдет с концевого или до того, как он упрется в крайнее положение.

Если вход **Имитатор** доступен (см. 6.8.11.3), его можно соединить с выходом соответствующего алгоблока имитации аварийных состояний.

Выходы **SQоткр** и **SQзакр** определяют значение концевых выключателей на открытие и закрытие, соответственно. Таблица 5 определяет значения концевых выключателей в зависимости от положения запорного механизма задвижки.

Таблица 5 - Значения концевых выключателей в зависимости от положения запорного механизма задвижки

Положение запорного механизма	SQоткр	SQзакр	Время хода
0...0,2%	Лог 1 (Да)	Лог 0 (Нет)	Тхко
0,2...99,8%	Лог 0 (Нет)	Лог 0 (Нет)	Тхода
99,8...100%	Лог 0 (Нет)	Лог 1 (Да)	Тхкз

Если запорный механизм открывается, значение на выходе **ХодОткр** равно Лог 1 (Да), когда закрывается на выходе **ХодЗакр** Лог 1 (Да). Эти сигналы можно использовать по отдельности или в совокупности для формирования сигналов подтверждения открытия, закрытия и движения задвижки.

Если на выходе **Увыс** Лог 0 (Нет), значит «обмотки электропривода» задвижки обесточены и управление задвижкой невозможно (рабочий механизм задвижки остановится вне зависимости от управляющих сигналов).

Если на выходе **Упус** Лог 0 (Нет), значит обмотки пускателя обесточены и управление задвижкой невозможно. В этом случае, если задвижка имеет импульсное управление и ее рабочий механизм находился в движении, то он дойдет до крайнего положения и не будет реагировать на импульсные сигналы управления. В случае потенциального управления – задвижка будет остановлена и управление ею окажется невозможным.

На выходе **% откр** отображается процент открытия рабочего механизма задвижки. Задвижка считается полностью закрытой, когда на выходе 0%, и полностью открытой, когда – 100%.

6.8.11.4 Мастер настройки

Для настройки типа управления задвижкой, а также отображения некоторых входов, следует щелкнуть на заголовке алгоблока и вызвать Мастера настройки. С его помощью можно выбрать тип управления (потенциальное или импульсное). Времена хода рабочего органа на разных отрезках пути могут быть заданы жестко, а могут быть выведены в качестве входов, которые можно соединить с другими алгоблоками. В последнем случае следует установить флаг **Управлять временами схода**. Установка флага **Имитация аварийных ситуаций** приведет к появлению дополнительного входа **Имитатор**, который можно соединить с выходом алгоблока имитации аварийных ситуаций.

6.8.12 Фрейм

6.8.12.1 Назначение

Алгоритм предназначен для сохранения и восстановления именованных **Срезов данных** на сервере Мезона. Алгоритм может работать в паре с алгоритмом **Виконт**, что позволяет организовать сохранение полномасштабного среза данных на сервере выполнения Мезон и на виртуальных контроллерах. Это дает возможность создавать модель объекта и системы управления с возможностью сохранения промежуточных состояний и быстрого перехода к ним по необходимости.

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

6.8.12.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	Имя фрейма	строковый	""	Имя сохраняемого или восстанавливаемого среза данных.
	Сохранить	событийный	-	Команда сохранения среза данных
	Восстановить	событийный	-	Команда восстановления среза данных
Выходы	Сохранить	событийный	-	Генерация команды сохранения для других алглобков
	Восстановить	событийный	-	Генерация команды восстановления для других алглобков
	Текст ошибки	строковый	""	Текст ошибки, возникшей при выполнении последней команды. Если ошибки нет, то – пустая строка
	Ошибка	событийный	-	Событие, генерирующееся при возникновении ошибки сохранения или восстановления среза.

6.8.12.3 Описание алгоритма

На входе **Имя фрейма** должно быть сформировано символическое имя сохраняемого фрейма данных. В качестве символов могут использоваться любые символы, допустимые в именах файлов. При возникновении на входе **Сохранить** события, алгоритм получит срез данных с сервера выполнения на котором он выполняется и в случае успеха запишет результат в файл, с именем, равным имени фрейма и расширением **frm**. Если сохранение в файл прошло успешно – на выходе **Сохранить** будет сформировано событие, которое можно передать другим алглобкам. Аналогичным образом можно восстановить срез данных, сохраненный ранее. Если файл среза с именем, равным имени фрейма будет найден и успешно применен на сервере выполнения, на выходе **Восстановить** будет сформировано событие. В случае возникновения ошибки в момент сохранения или восстановления среза данных на выходе **Текст ошибки** появится подробное описание ее причины, а на выходе **Ошибка** будет сформировано событие: **Внимание! Сохранение среза данных под существующим именем затрет ранее сохраненный срез данных.**

Данный алгоритм приспособлен для работы в паре с алгоритмами **Виконт**. Для того, чтобы сохранить срез данных не только на сервере выполнения Мезона, но и в виртуальных контроллерах под одним и тем же именем, следует обеспечить одинаковое имя среза на входах **Имя фрейма** алглобка **Фрейм** и **Имя Среза** алглобков **Виконт**. Событийные выходы алглобка **Фрейм** следует связать с соответствующими событийными входами алглобков **Виконт**. Примерная схема конфигурации алглобков для сохранения общего фрейма данных сервера и виртуальных контроллеров показана на рисунке 56.

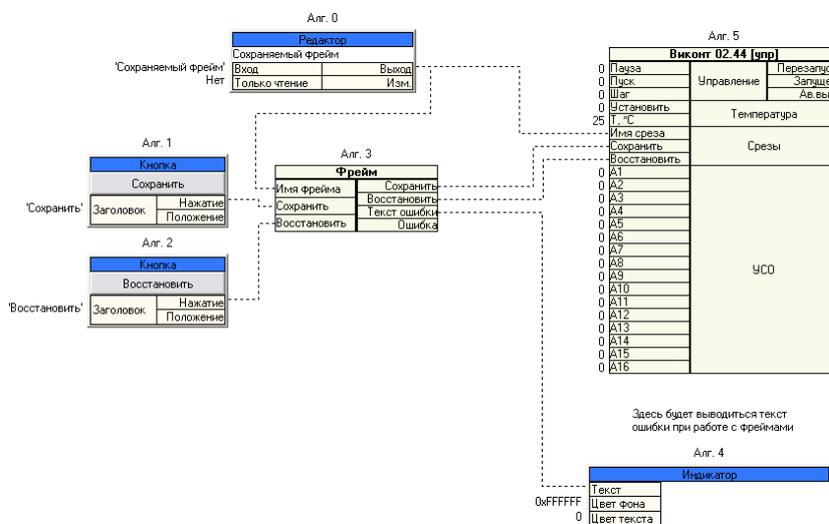


Рисунок 56 – Пример конфигурации алглобков для сохранения общего фрейма данных

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

6.8.13 БОК

6.8.13.1 Назначение

Алгоритм предназначен для имитации запорной арматуры с потенциальным или импульсным управлением. С его помощью можно имитировать поведение быстроотсечных клапанов (БОК).

6.8.13.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	<i>Открыть</i>	событийный	0	Импульсная команда открытия БОКа
	<i>Закрыть</i>	событийный	0	Импульсная команда закрытия БОКа
	<i>Управление</i>	событийный	0	Потенциальная команда управления БОКом
	<i>Тоткр</i>	модельное время	2	Относительное или абсолютное время открытия запорного механизма, мс
	<i>Тзкр</i>	модельное время	2	Относительное или абсолютное время закрытия запорного механизма, мс
	<i>Имитатор</i>	контейнер данных	-	Вход подключения алглока имитации аварийных состояний
Выходы	<i>SQоткр</i>	логический	-	Концевой выключатель на открытие
	<i>SQзакр</i>	логический	-	Концевой выключатель на закрытие
	<i>ХодОткр</i>	логический	-	Подтверждение хода БОКа на открытие
	<i>ХодЗакр</i>	логический	-	Подтверждение хода БОКа на закрытие
	<i>Uвыс</i>	логический	-	Признак наличия высокого напряжения обмоток электропривода
	<i>Uпус</i>	логический	-	Признак наличия питающего напряжения обмоток пускателя
	<i>% откр</i>	действительный	-	Положение рабочего органа, в % (0 – закрыт, 100 – открыт)

6.8.13.3 Описание алгоритма

Алгоритм может работать в одном из двух режимов управления: потенциальном и импульсном.

В потенциальном режиме рабочий орган открывается и удерживается в открытом состоянии до тех пор, пока действует потенциальная команда на входе **Управление** (Да) и закрывается при пропадании этой команды (Нет). В импульсном режиме ход рабочего механизма происходит по наличию соответствующей команды на входах **Открыть** и **Закрыть**. (Наличие одновременно двух сигналов Лог 1 (Да) на входах **Открыть** и **Закрыть** расценивается как отсутствие сигналов).

Задать время хода на открытие БОКа можно установив константу **Тоткр**. За это время рабочий механизм задвижки должен проделать путь между 0,2% ... 99,8 % (т.е. между концевыми). Время хода на закрытие БОКа как правило на один или несколько порядков меньше времени хода на открытие и задается на входе **Тзкр**.

Если вход **Имитатор** доступен (см. 6.8.13.3), его можно соединить с выходом соответствующего алглока имитации аварийных состояний.

Выходы **SQоткр** и **SQзакр** определяют значение концевых выключателей на открытие и закрытие, соответственно. Таблица 6 определяет значения концевых выключателей в зависимости от положения запорного механизма клапана.

Таблица 6 - Значения концевых выключателей в зависимости от положения запорного механизма клапана

Положение запорного механизма	SQоткр	SQзакр	Время хода
0...0,2%	Лог 1 (Да)	Лог 0 (Нет)	<i>Тхко</i>
0,2...99,8%	Лог 0 (Нет)	Лог 0 (Нет)	<i>Тхода</i>
99,8...100%	Лог 0 (Нет)	Лог 1 (Да)	<i>Тхкз</i>

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

Изм	Лист	№ докум	Подп.	Дата	ПФДИ.421457.009 И3.6	Лист
						115

Если запорный механизм открывается, значение на выходе **ХодОткр** равно Лог 1 (Да), когда закрывается на выходе **ХодЗакр** Лог 1 (Да). Эти сигналы можно использовать по отдельности или в совокупности для формирования сигналов подтверждения открытия, закрытия и движения задвижки.

Если на выходе **Увыс** Лог 0 (Нет), значит «обмотки электропривода» задвижки обесточены и управление задвижкой невозможно (рабочий механизм задвижки остановится вне зависимости от управляющих сигналов).

Если на выходе **Упус** Лог 0 (Нет), значит обмотки пускателя обесточены и управление задвижкой невозможно. В этом случае, если задвижка имеет импульсное управление и ее рабочий механизм находился в движении, то он дойдет до крайнего положения и не будет реагировать на импульсные сигналы управления. В случае потенциального управления – задвижка будет остановлена и управление ею окажется невозможным.

На выходе **% откр** отображается процент открытия рабочего механизма задвижки. БОК считается полностью закрытым, когда на выходе 0%, и полностью открытым, когда – 100%.

6.8.13.4 Мастер настройки

Для настройки типа управления БОК, а также отображения некоторых входов, следует щелкнуть на заголовке алгоблока и вызвать Мастера настройки. С его помощью можно выбрать тип управления (потенциальное или импульсное). Времена хода рабочего органа на открытие и закрытие могут быть заданы жестко, а могут быть выведены в качестве входов, которые можно соединить с другими алгоблоками. В последнем случае следует установить флаг **Управлять временами схода**. Установка флага **Имитация аварийных ситуаций** приведет к появлению дополнительного входа **Имитатор**, который можно соединить с выходом алгоблока имитации аварийных ситуаций.

6.8.14 РегКлапан

6.8.14.1 Назначение

Алгоритм предназначен для имитации запорной арматуры с шаговым управлением. С его помощью можно имитировать поведение регуляторов с промежуточным положением запорного механизма, управляемого шаговым электродвигателем.

6.8.14.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	<i>V</i>	цельный	0	Значение скорости перемещения рабочего органа в процентах от номинальной (-127 соответствует -100%, 127 – соответствует 100%)
	<i>Tхода</i>	модельное время	2	Относительное или абсолютное время хода запорного механизма между концевыми датчиками, мс
	<i>Tлюфт</i>	модельное время	2	Максимальное относительное или абсолютное время выбора люфта при изменении направления движения рабочего органа, мс
	<i>Имитатор</i>	контейнер данных	-	Вход подключения алгоблока имитации аварийных состояний
Выходы	<i>SQоткр</i>	логический	-	Концевой выключатель на открытие
	<i>SQзакр</i>	логический	-	Концевой выключатель на закрытие
	<i>ХодОткр</i>	логический	-	Подтверждение хода задвижки на открытие
	<i>ХодЗакр</i>	логический	-	Подтверждение хода задвижки на закрытие
	<i>Увыс</i>	логический	-	Признак наличия высокого напряжения обмоток электропривода
	<i>Упус</i>	логический	-	Признак наличия питающего напряжения обмоток пускателя
	<i>% откр</i>	действительный	-	Положение рабочего органа, в % (0 – закрыт, 100 – открыт)
	<i>% люфт</i>	действительный	-	Выбор люфта рабочим органа, в %

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

				(0 – люфт не выбран, 100 – люфт полностью выбран)
--	--	--	--	---

Рабочий орган регулирующего клапана изменяет свое положение от 0 до 100% (от полностью закрыт до полностью открыт) и обратно за **Tхода** при движении с максимальной скоростью. Если скорость меньше максимальной – время хода между концевыми датчиками увеличивается. Его можно рассчитать по формуле:

$$t_{\text{хода}} = T_{\text{хода}} / v / 127.$$

Если скорость **V** – отрицательная – рабочий орган клапана начинает закрываться. При нулевой скорости – сохраняется промежуточное положение клапана. При изменении знака скорости (а значит и направления хода рабочего органа) редуктор привода должен вначале выбрать люфт, поэтому, прежде, чем рабочий орган начнет двигаться в противоположную сторону пройдет время, равное времени выбора люфта:

$$T_{\text{люфт}} / v / 127.$$

Алгоритм может управляться алгоритмом **БК** посредством алгоритма ЦИП, на выходе которого формируется правильное значение скорости, зависящее от задания.

Если вход **Имитатор** доступен (см. 6.8.14.3), его можно соединить с выходом соответствующего алгоблока имитации аварийных состояний.

Если запорный механизм открывается, значение на выходе **ХодОткр** равно Лог 1 (Да), когда закрывается на выходе **ХодЗакр** Лог 1 (Да). Эти сигналы можно использовать по отдельности или в совокупности для формирования сигналов подтверждения открытия, закрытия и движения задвижки.

Если на выходе **Uвыс** Лог 0 (Нет), значит «обмотки электропривода» задвижки обесточены и управление задвижкой невозможно (рабочий механизм задвижки остановится вне зависимости от управляющих сигналов).

Если на выходе **Uпус** Лог 0 (Нет), значит обмотки пускателя обесточены и управление задвижкой невозможно. В этом случае, если задвижка имеет импульсное управление и ее рабочий механизм находился в движении, то он дойдет до крайнего положения и не будет реагировать на импульсные сигналы управления. В случае потенциального управления – задвижка будет остановлена и управление ею окажется невозможным.

На выходе **% откр** отображается процент открытия рабочего механизма задвижки. Задвижка считается полностью закрытой, когда на выходе 0%, и полностью открытой, когда – 100%.

6.8.14.3 Мастер настройки

Для настройки отображения некоторых входов, следует щелкнуть на заголовке алгоблока и вызвать Мастера настройки. Времена хода рабочего органа на разных отрезках пути могут быть заданы жестко, а могут быть выведены в качестве входов, которые можно соединить с другими алгоблоками. В последнем случае следует установить флаг **Управлять временами схода**. Установка флага **Имитация аварийных ситуаций** приведет к появлению дополнительного входа **Имитатор**, который можно соединить с выходом алгоблока имитации аварийных ситуаций.

6.8.15 АД-1

6.8.15.1 Назначение

Алгоритм предназначен для имитации работы односкоростных асинхронных электродвигателей. С его помощью можно имитировать поведение любых асинхронных электродвигателей.

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

6.8.15.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	<i>Пуск</i>	событийный	-	Команда запуска электродвигателя
	<i>Стоп</i>	событийный	-	Команда останова электродвигателя
	<i>M</i>	модельное время	2	Сигнал нагрузки на валу, кГс/м
	<i>Имитатор</i>	контейнер данных	-	Вход подключения алгоблока имитации аварийных состояний
Выходы	<i>Uвыс</i>	логический	-	Признак наличия высокого напряжения обмоток электропривода
	<i>Uпус</i>	логический	-	Признак наличия питающего напряжения обмоток пускателя
	<i>Скор</i>	логический	-	Признак достижения двигателем номинальной скорости
	<i>I</i>	действительный	-	Ток, потребляемая статорными обмотками двигателя, А
	<i>N</i>	целый	-	Текущее число оборотов вала, об/мин

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

Изм	Лист	№ докум	Подп.	Дата

ПФДИ.421457.009 И3.6

6.9 Группа 'Отладка'

6.9.1 ОтлПечать

6.9.1.1 Назначение

Выводит текст в окно консоли Квинтегратора.

6.9.1.2 Параметры

Наименование	Обозначение	Тип	Назначение
Входы	Разреш	Логический	Разрешает вывод текста
	Ошибка	Логический	Является ли сообщение ошибкой
	A1	Строковый	1-й параметр
	A2	Строковый	2-й параметр
	A3	Строковый	3-й параметр
	A4	Строковый	4-й параметр
A5	Строковый	5-й параметр	

6.9.1.3 Описание алгоритма

Алгоритм позволяет выводить любой текст в окно консоли **Квинтегратора** (если Мезон запущен из Квинтегратора) или **Монитора приложений** (если Мезон запущен Монитором приложений). Вывод осуществляется только если на значение на входе **Разреш** равно Да. В каждом цикле выводится одна строка. Строка представляет собой соединение строк-параметров **A1 – A5**, разделенных пробелом.

Каждая строка консольного вывода в Квинте принадлежит к некоторой **группе**. Каждая группа может быть включена или выключена. Когда группа выключена, строки, принадлежащие этой группе не формируются. Включать и отключать группы можно в приложении **Настройка – Отладочная печать**.

Если значение на входе **Ошибка** равно Да, выводимая строка приписывается к группе **MezonAlg_Error**, в противном случае – к группе **MezonAlg_Info**. По умолчанию, группа **MezonAlg_Error** включена, а группа **MezonAlg_Info** – выключена.

6.9.1.4 Рекомендации

Если вывод строки происходит в каждом цикле, то это существенно тормозит выполнение задач в Мезоне и засоряет log-файлы. Поэтому рекомендуется:

- 1 Если алгоритм **ОтлПечать** используется для анализа значений на выходах других алгоблоков, устанавливайте значение Нет на входе **Ошибка**. Тогда печать будет работать только при включенной группе **MezonAlg_Info**.
- 2 Если не имеет смысла выполнять печать в каждом цикле, свяжите вход **Разреш** и управляйте им с помощью других алгоблоков.
- 3 Если на входе **Ошибка** установлено значение Да, всегда управляйте печатью с помощью входа **Разреш**.
- 4 Если в проекте используется несколько алгоблоков **ОтлПечать**, задайте на их первых входах константу - строку описания алгоблока. Это поможет различить выводимые строки.

6.9.2 ВремяЦикла

6.9.2.1 Назначение

Алгоритм предназначен для определения тайминга задачи в которой он выполняется.

Позволяет подсчитать время выполнения задачи, с набором статистики по циклам ее выполнения. Определение времени цикла ведется с точностью не хуже 1 мкс.

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

Изм	Лист	№ докум	Подп.	Дата	ПФДИ.421457.009 И3.6	Лист
						119

6.9.2.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	N	целый	10	Число циклов на которых усредняется результат
	CLS	логический	Нет	Сброс мгновенных значений
Выход	$T...$	Действительный	-	Среднее время цикла за N циклов, мкс
	$Tmin...$	Действительный	-	Минимальное время цикла за N циклов, мкс
	$Tmax...$	Действительный	-	Максимальное время цикла за N циклов, мкс
	T	целый	-	Время предыдущего цикла, мкс
	$Tmin$	целый	-	Минимальное время цикла, начиная от первого цикла или от первого сброса, мкс
	$Tmax$	целый	-	Максимальное время цикла, начиная от первого цикла или от первого сброса, мкс

6.9.2.3 Описание алгоритма

Алгоритм позволяет ориентировочно оценить время цикла задачи, определить укладывается ли ее выполнение в заданный интервал времени. Для этих целей могут служить значения $T...$, $Tmin...$, $Tmax...$. Среднее время цикла задачи показывает примерную производительность задачи за N -ное число циклов. Минимальное и максимальное значения за N циклов, позволяют судить о правильной настройке задачи. Так, если задача должна выполняться раз в 50 мс, а среднее время ее выполнения за 100 циклов, равно 120000 мкс, значит ни один цикл выполнения задачи не укладывается в заданный интервал и его целесообразно увеличить. Если же среднее время цикла примерно равно, к примеру, 49000 мкс, а минимальные и максимальные времена равны 10000 и 88000 мкс, соответственно, то это означает, что некоторые циклы выполнения задачи не уложились в заданный интервал, и в следующих циклах задача уменьшила время цикла с целью усреднения времени выполнения до заданного значения (последнее справедливо при настройке задачи с компенсацией времени цикла). Большая разница между минимальным и максимальным временами циклов на интервале усреднения свидетельствует о сильной неравномерности выполнения задачи по тем или иным причинам.

Значения предыдущего цикла, а также минимальное и максимальное время выполнения от начала выполнения задачи или от момента последнего сброса мгновенных значений (T , $Tmin$, $Tmax$), могут быть использованы для самостоятельной обработки, а также для записи в файл (или в Базу данных), с помощью соответствующих алгоблоков.

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

6.10 Группа 'Excel'

6.10.1 КнигаXL

6.10.1.1 Параметры

Наименование	Обознач.	Тип	Номер	Нач. знач.	Назначение
Входы	<i>Читать из файла</i>	строковый	1		Имя открываемого в Excel файла
	<i>Сохранить в файл</i>	строковый	2		Имя сохраняемого в Excel файла
	<i>Шаблон</i>	строковый	3		Файл шаблона Excel
	<i>Сохранить</i>	событийный	4		Событие сохранения файла
	<i>Печать</i>	событийный	5		Событие печать файла на принтер
Выходы	<i>Книга</i>	контейнер данных	1		Указатель на открытую в Excel книгу

6.10.1.2 Назначение

Создание и открытие в Excel рабочей книги. Автоматическое сохранение и печать результатов.

6.10.1.3 Описание алгоритма

Алгоритм создает рабочую книгу Excel и открывает в ней файл, указанный на входе **Читать из файла**. Если значение не указано, то открывается пустая книга. На выходе **Книга** алгоритм предоставляет остальным алгоритмам группы указатель на открытую книгу. Алгоритм сохраняет результаты в файл, указанный на входе **Сохранить в файл**. Если имя сохраняемого файла не указано, результаты не сохраняются. Сохранение происходит при получении алгоритмом на входе **Сохранить** события, или, если вход **Сохранить** не законфигурирован, при штатном завершении задачи. Для представления результатов в требуемом пользователем виде, можно использовать файлы шаблонов Excel. Алгоритм также позволяет вывести результаты на принтер. Для этого на вход **Печать** необходимо подать событие.

6.10.1.4 Мастер настроек

Значения на входах 1, 2, 3 задаются с помощью Мастера. Имя открываемого файла и имя шаблона задаются с помощью стандартных диалогов открытия файлов. Имя сохраняемого файла задается с помощью стандартного диалога сохранения файлов. Имя сохраняемого файла можно также задать отредактировав или законфигурировав этот вход. Диалоги вызываются нажатием на кнопки соответствующие редактируемым полям.

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

Изм	Лист	№ докум	Подп.	Дата	ПФДИ.421457.009 И3.6	Лист 121
-----	------	---------	-------	------	----------------------	-------------

6.10.2 ЗаписьXL

6.10.2.1 Параметры

Наименование	Обознач.	Тип	Номер	Нач. знач.	Назначение
Входы	<i>Книга</i>	контейнер данных	1	0	Указатель на книгу Excel
	<i>Лист</i>	целый	2	1	Номер листа книги Excel
	<i>Разрешить</i>	логический	3	True	Признак разрешения записи значений в ячейки
	<i>Столбец</i>	строковый	4	1	Номер столбца ячейки
	<i>Строка</i>	целый	5	1	Номер строки ячейки
	<i>Значение</i>	строковый	6		Значение помещаемое в ячейку (Столбец; Строка)
Выходы	<i>Статус</i>	целый	1		Статус успешности записи значения в ячейку

6.10.2.2 Назначение

Запись значений в книгу Excel.

6.10.2.3 Описание алгоритма

Алгоритм осуществляет запись значений в книгу Excel, указатель на которую должен быть подан на вход **Книга** с выхода **Книга** алгоритма **КнигаXL**. Если вход **Книга** останется незаконфигурированным, значения записываться не будут. Значения записываются в лист рабочей книги, заданный на входе **Лист**. Значения записываются в ячейку, заданную на входах **Столбец** и **Строка**. Значения поступают на вход **Значение**. Запись осуществляется если на входе **Разрешить** установлено True. На выходе **Статус** устанавливается «0» если запись была успешной и «1» в противном случае.

6.10.3 ЧтениеXL

6.10.3.1 Параметры

Наименование	Обознач.	Тип	Номер	Нач. знач.	Назначение
Входы	<i>Книга</i>	контейнер данных	1	0	Указатель на книгу Excel
	<i>Лист</i>	целый	2	1	Номер листа книги Excel
	<i>Столбец</i>	строковый	3	1	Номер столбца ячейки
	<i>Строка</i>	целый	4	1	Номер строки ячейки
Выходы	<i>Значение</i>	строковый	1	-	Значение из ячейки (Столбец; Строка)
	<i>Статус</i>	целый	2	-	Статус успешности чтения значения из ячейки

6.10.3.2 Назначение

Чтение значений из книги Excel.

6.10.3.3 Описание алгоритма

Алгоритм осуществляет чтение значений из книги Excel, указатель на которую должен быть подан на вход **Книга** с выхода **Книга** алгоритма **КнигаXL**. Если вход **Книга** останется

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

Изм	Лист	№ докум	Подп.	Дата	ПФДИ.421457.009 ИЗ.6	Лист
						122

незаконфигурированным, значения читаться не будут. Значения читаются из листа рабочей книги, заданный на входе **Лист**. Значения читаются из ячейки, заданной на входах **Столбец** и **Строка**. Значения поступают на выход **Значение**. На выходе **Статус** устанавливается «0» если чтение было успешным и «1» в противном случае.

6.10.4 ЛистXL

6.10.4.1 Параметры

Наименование	Обознач.	Тип	Номер	Нач. знач.	Назначение
Входы	<i>Книга</i>	контейнер данных	1	0	Указатель на книгу Excel
	<i>Лист</i>	целый	2	1	Номер листа книги Excel
	<i>Имя</i>	строковый	3		Имя листа книги Excel
Выходы	<i>Столбцов</i>	строковый	1	-	Количество столбцов в использованном диапазоне
	<i>Строк</i>	целый	2	-	Количество строк в использованном диапазоне

6.10.4.2 Назначение

Изменение свойств листа Excel. Получение использованного диапазона Excel.

6.10.4.3 Описание алгоритма

Алгоритм позволяет изменять некоторые свойства рабочего листа Excel, принадлежащего книге, указатель на которую должен быть подан на вход **Книга** с выхода **Книга** алгоритма **КнигаXL**. Номер листа задается на входе **Лист**. Если вход **Книга** останется незаконфигурированным, алгоритм работать не будет. Количество строк в использованном диапазоне выводится на выход **Строк**. Количество столбцов в диапазоне выводится на выход **Столбцов**.

6.10.5 ДиапазонXL

6.10.5.1 Параметры

Наименование	Обознач.	Тип	Номер	Нач. знач.	Назначение
Входы	<i>Книга</i>	контейнер данных	1	0	Указатель на книгу Excel
	<i>Лист</i>	целый	2	1	Номер листа книги Excel
	<i>Разрешить</i>	логический	3	False	Признак разрешения изменения формата диапазона ячеек
	<i>Столбец</i>	строковый	4	1	Номер столбца первой ячейки диапазона
	<i>Строка</i>	целый	5	1	Номер строки первой ячейки диапазона
	<i>Столбец</i>	строковый	6	1	Номер столбца последней ячейки диапазона
	<i>Строка</i>	целый	7	1	Номер строки последней ячейки диапазона

6.10.5.2 Назначение

Изменение формата диапазона ячеек Excel.

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

Изм	Лист	№ докум	Подп.	Дата	ПФДИ.421457.009 ИЗ.6	Лист
						123

6.10.5.3 Описание алгоритма

Алгоритм осуществляет изменение формата диапазона ячеек книги Excel, указатель на которую должен быть подан на вход **Книга** с выхода **Книга** алгоритма **КнигаXL**. Если вход **Книга** останется незаконфигурированным, изменения формата не произойдет. Форматируются ячейки листа рабочей книги, заданного на входе **Лист**. Диапазон определяется парами значений **Столбец** и **Строка** соответствующих первой и последней ячейкам диапазона. Изменение формата осуществляется если на входе **Разрешить** установлено True.

6.10.5.4 Мастер настроек

Мастер настроек формата ячеек позволяет изменить:

- шрифт текста ячейки. Кнопкой **Шрифт** вызывается стандартный диалог выбора шрифта;
- цвет заливки ячейки. Кнопкой **Цвет заливки** вызывается стандартный диалог выбора цвета;
- границы и отдельные линии ячеек. Кнопками с изображением соответствующих линий. Повторное нажатие отменяет взведенный признак;
- цвет линий. Кнопкой **Цвет линии**.

6.10.6 ЯчейкаXL

6.10.6.1 Параметры

Наименование	Обознач.	Тип	Номер	Нач. знач.	Назначение
Входы	<i>Книга</i>	контейнер данных	1	0	Указатель на книгу Excel
	<i>Лист</i>	целый	2	1	Номер листа книги Excel
	<i>Разрешить</i>	логический	3	False	Признак разрешения изменения формата ячейки
	<i>Столбец</i>	строковый	4	1	Номер столбца ячейки
	<i>Строка</i>	целый	5	1	Номер строки ячейки

6.10.6.2 Назначение

Изменение формата ячейки Excel.

6.10.6.3 Описание алгоритма

Алгоритм осуществляет изменение формата ячейки книги Excel, указатель на которую должен быть подан на вход **Книга** с выхода **Книга** алгоритма **КнигаXL**. Если вход **Книга** останется незаконфигурированным, изменения формата не произойдет. Форматируются ячейки листа рабочей книги, заданного на входе **Лист**. Ячейка определяется парой значений **Столбец** и **Строка**. Изменение формата осуществляется если на входе **Разрешить** установлено True.

6.10.6.4 Мастер настроек

Мастер настроек формата ячейки аналогичен мастеру настроек для диапазона ячеек (см. 6.10.5.4), за исключением некоторых линий, которые не задаются для отдельной ячейки.

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

										Лист
										124
Изм	Лист	№ докум	Подп.	Дата	ПФДИ.421457.009 И3.6					

максимальное и минимальное параметра соответственно, а на выход **ПроцЗн** - значение сигнала в процентах, на выход **НедостЗн** - значение "Нет".

При выполнении алгоритма могут возникнуть следующие ситуации, которые в задаче можно отследить и обработать:

- Значение параметра пока не получено. На выходы **Готово**, **ЕстьЗн** и **НедостЗн** – заносится значение "Нет", а на выход **Статус** - число 2 (бит 02 = 1);
- Считано недостоверное значение. На выходы **Готово**, **ЕстьЗн** и **НедостЗн** – заносится значение "Да", на выход **Статус** заносится число 1 (бит 01 = 1), а на выход **Текст** - текстовое значение параметра с предшествующим ему символом "@";
- Ошибка от источника. На выход **Готово** заносится значение "Да", на выходы **ЕстьЗн** и **НедостЗн** – значение "Нет", на выход **Статус** - число 4 (бит 03 = 1), на выход **Ошибка** - код ошибки, а на выход **Текст** – описание ошибки.

6.11.1.4 Мастер настроек

Мастер настроек позволяет ввести значения на входы алгоритма, выбрать источник данных, задать нужное количество выходов алгоритма и задать автоматическое прерывание задачи в случае ошибки чтения параметра.

Установка значений на входах **Марка** и **Параметр** осуществляется нажатием кнопки **Выбрать параметр**.

Источник данных устанавливается включением одной из радио кнопок группы **Источник**, где тип **По умолчанию** означает, что источник определяется при запуске приложения Мезон, а тип **Первоисточник** - означает, что источник определяется в зависимости от типа привязок параметра в базе данных. Так, для расчетного параметра первоисточником является архив, а для параметра, привязанного в Пилоне – контроллер.

Необязательные выходы назначаются установкой соответствующих флагов в группе **Дополнительные выходы**.

Управление прерыванием выполнения задачи в случае ошибки осуществляется установкой соответствующих флагов в группе **Автоматическое прерывание задачи**. Если эти флаги не установлены, пользователю рекомендуется анализировать значение на выходе **Статус** и самому решать, что делать в случае той или иной ошибки.

6.11.2 Писать Параметр

6.11.2.1 Назначение

Алгоритм предназначен для записи значения параметра объекта ПТК КВИНТ в указанный приемник.

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

6.11.2.2 Параметры

Наименование	Обознач.	Тип	Номер	Нач. знач.	Назначение
Входы	<i>Писать</i>	Событийный	1		Событие записи
	<i>Марка</i>	Строковый	2		Марка объекта
	<i>Параметр</i>	Строковый	3		Имя параметра объекта
	<i>Значение</i>	Действительный	4		Значение параметра
	<i>НедостЗн</i>	Логический	5	Нет	Признак записи недостоверного значения
	<i>Ждать</i>	Логический	6	Да	Признак ожидания подтверждения записи
	<i>Обновить</i>	Логический	7	Нет	Признак перезаписи предыдущего значения
Выходы	<i>Ошибка</i>	Целый	1	0	Код ошибки
	<i>Текст</i>	Строковый	2		Описание ошибки
	<i>Размерн.</i>	Строковый	3		Техническая размерность параметра
	<i>МинЗн</i>	Действительный	4		Минимальное значение параметра
	<i>МаксЗн</i>	Действительный	5		Максимальное значение параметра

6.11.2.3 Описание алгоритма

Алгоритм имеет 4 входа, 1 обязательный и 4 дополнительных выхода. Обязательным выходом является **Ошибка**. Необязательные выходы назначаются в Мастере настроек.

Алгоритм пытается записать параметр, имя, марка и значение которого заданы на входах **Марка**, **Параметр** и **Значение**, в указанный приемник при условии, что на вход **Писать** поступило событие. Если вход **Писать** не законфигурирован, алгоритм не будет делать попыток записать параметр.

Приемник, марка и параметр задаются в Мастере настроек, однако марку и(или) параметр можно также задать редактированием значений на входах или конфигурированием этих выходов. Если марка не задана, алгоритм выполняться не будет. Если не задан параметр, алгоритм выберет из системной Базы данных Главный параметр объекта. Алгоритм не будет выполняться и в случае, когда марка и (или) параметр не найдены в Базе данных.

Если записываемое значение параметра недостоверно, то на входе **НедостЗн** следует задать значение "Да". Если требуется ждать подтверждения записи, то вход **Ждать** должен иметь значение "Да". Если предыдущее значение параметра нужно заменить новым (только для архива), то на входе **Обновить** должно быть значение "Да".

При успешном выполнении на выход **Ошибка** - заносится число 0. Если для алгоритма заданы дополнительные выходы, то на выход **Размерн.** - заносится техническая размерность параметра, а на выходы **МинЗн** и **МаксЗн** – минимальное и максимальное значение параметра соответственно.

Если алгоритму не удалось записать параметр, на выход **Ошибка** заносится код ошибки, а на выход **Текст** – ее описание.

6.11.2.4 Мастер настроек

Мастер настроек позволяет ввести значения на входы алгоритма, выбрать приемник данных, задать нужное количество выходов алгоритма, заказать запись в журнал действий пользователя и задать автоматическое прерывание задачи в случае ошибки.

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

Установка значений на входах **Марка** и **Параметр** осуществляется нажатием кнопки **Выбрать параметр**.

Приемник данных устанавливается включением одной из радиокнопок группы **Приемник**, где тип **По умолчанию** означает, что приемник определяется при запуске приложения Мезон, а тип **Первоисточник** означает, что источник определяется в зависимости от типа привязок параметра в Базе данных. Так, для расчетного параметра первоисточником является архив, а для параметра, привязанного в Пилоне – контроллер.

Необязательные выходы назначаются установкой соответствующих флагов в группе **Дополнительные выходы**.

Управление записью в журнал действий пользователя осуществляется установкой флага **Писать в журнал действий пользователя**.

Управление прерыванием выполнения задачи в случае ошибки осуществляется установкой флага **Если ошибка - вызывать алгоритм Ошибка**. Если этот флаг не установлен, пользователю рекомендуется самому решать, что делать в случае той или иной ошибки.

6.11.3 История

6.11.3.1 Назначение

Алгоритм предназначен для выбора из архива одного значения параметра какого либо объекта Квинта по заданным признакам на заданном интервале времени. Алгоритм устарел, вместо него рекомендуется использовать алгоритм **История 2**.

Инв. № подл.	Подп. и дата	Взаим. инв. №	Инв. №	Подп. и дата	ПФДИ.421457.009 И3.6					Лист
					Изм	Лист	№ докум	Подп.	Дата	128

6.11.3.2 Параметры

Наименование	Обознач.	Тип	Номер	Нач. знач.	Назначение
Входы	<i>Марка</i>	Строковый	1		Марка объекта
	<i>Параметр</i>	Строковый	2		Имя параметра объекта
	<i>НачВр</i>	Дата и время	3	0	Дата и время начала интервала времени
	<i>КонВр</i>	Дата и время	4	0	Дата и время конца интервала времени
	<i>Шаг</i>	Длительность	5	0	Интервал просеивания по времени
	<i>Повторить</i>	Событийный	6	0	Событие повторения выборки значений параметра
Выходы	<i>Готово</i>	Логический	1	Нет	Признак завершения запроса параметра (Да - получено значение или ошибка)
	<i>ЕстьЗн</i>	Логический	2	Нет	Признак получения значения параметра
	<i>НедостЗн</i>	Логический	3	Нет	Признак недоверности значения параметра
	<i>Конец</i>	Логический	4	Нет	Признак конца значений параметра
	<i>ФизЗн</i>	Действительный	5		Физическое значение параметра
	<i>ПроцЗн</i>	Действительный	6		Значение параметра в процентах
	<i>Текст</i>	Строковый	7		Текстовое значение параметра или описание ошибки
	<i>Время</i>	Дата и время	8	0	Дата и время получения очередного значения
	<i>Статус</i>	Векторный	9	2	Результат выполнения операции поиска параметра
	<i>Ошибка</i>	Целый	10	0	Код ошибки
	<i>КолЗн</i>	Целый	11	0	Количество найденных значений
	<i>Размерн.</i>	Строковый	12		Техническая размерность параметра
	<i>МинЗн</i>	Действительный	13		Минимальное значение параметра
	<i>МаксЗн</i>	Действительный	14		Максимальное значение параметра

6.11.3.3 Описание алгоритма

Алгоритм имеет 6 входов, 5 обязательных и 9 дополнительных выходов. Обязательными выходами являются **Готово**, **ЕстьЗн**, **ФизЗн**, **Время** и **КолЗн**. Необязательные выходы назначаются в Мастере настроек.

Алгоритм пытается получить из архива одно значение параметра, имя и марка которого заданы на входах **Марка** и **Параметр**, внутри интервала времени, заданном входами **НачВр** и **КонВр**. Если **НачВр** меньше или равно **КонВр**, то поиск данных в архиве производится в прямом направлении, если же **НачВр** больше **КонВр**, то в обратном.

Марка и параметр задаются в Мастере настроек, а также - редактированием значений на входах или конфигурированием этих входов. Если марка не задана алгоритм выполняться не будет. Если не задан параметр, алгоритм выберет по системной базе данных главный

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

параметр объекта. Алгоритм не будет выполняться и в случае, когда марка и (или) параметр не найдены в базе данных.

Если значение параметра успешно получено, то на выходы **Готово** и **ЕстьЗн** – заносится значение **Да**, на выход **ФизЗн** - физическое значение параметра, на выход **Время** - дата и время записи его в архив, а значение на выходе **КолЗн** увеличивается на единицу.

Если для алгоритма заданы дополнительные выходы **Ошибка** и **Статус**, то они принимают значение **0**. На дополнительный выход **НедостЗн** заносится значение **Нет** для достоверного значения параметра или значение **Да** для недостоверного значения параметра.

Если задан дополнительный выход **Текст** – он заполняется округленным текстовым значением достоверного параметра или округленным текстовым недостоверным значением параметра с предшествующим символом **@**. На дополнительный выход **Размерн.** заносится техническая размерность параметра, на дополнительные выходы **МинЗн** и **МаксЗн** – минимальное и максимальное значение параметра соответственно, а на дополнительный выход **ПроцЗн** - значение сигнала в процентах.

При следующем выполнении алгоритма будет сделана попытка получить следующее значение на указанном интервале и т.д. до тех пор, пока не будут выбраны все значения.

Если при очередном выполнении алгоритма на указанном интервале времени значений параметра больше нет, то значение на выходе **КолЗн** не меняется, на выход **Статус** заносится число 8, а на выход **Конец** – **Да**. После чего за этим параметром алгоритм к архиву обращаться не будет.

Возобновить работу алгоритма можно одним из двух способов :

- послать событие на вход **Повторить**;
- заменить значения на входах **Марка** и (или) **Параметр**.

При возобновлении работы алгоритма заново считываются значения со входов **НачВр**, **КонВр** и **Шаг**.

Алгоритм позволяет просеивать выбираемые значения по определенным признакам (например, выбрать только первое значение). Величина интервала просеивания по времени задается на входе **Шаг** и выражается в секундах. Если **Шаг** равен 0, то выбираются все значения, если **Шаг** не равен 0, то на каждом временном отрезке равном значению входа **Шаг** из архива выбирается не более одного значения параметра по каждому признаку. Признаки задаются в Мастере настроек. Просеивание производится на Архивной станции и позволяет уменьшить сетевой трафик.

При выполнении алгоритма могут возникнуть следующие ситуации, которые в задаче можно отследить и обработать :

- значение параметра пока не получено. На выход **Статус** заносится число 2 (бит 02 = 1)
- считано недостоверное значение. На выход **Статус** заносится число 1 (бит 01 = 1)
- ошибка обращения к архиву. На выход **Статус** заносится число 4 (бит 03 = 1), на выход **Ошибка** - код ошибки, а на выход **Текст** – описание ошибки.

6.11.3.4 Мастер настроек

Мастер настроек позволяет установить значения на входах **Марка** и **Параметр**, задать нужное количество выходов алгоритма, определить совокупность признаков для отбора значений и задать автоматическое прерывание задачи в случае ошибки выбора параметра.

Установка значений на входах **Марка** и **Параметр** осуществляется нажатием кнопки **Выбрать параметр**.

Признаки, по которым будут выбираться значения, устанавливаются совокупностью флагов группы **Признаки отбора значений**.

Необязательные выходы назначаются совокупностью флагов группы **Дополнительные выходы**.

Управление прерыванием выполнения задачи в случае ошибки осуществляется установкой флага **Если ошибка - вызывать алгоритм Ошибка**. Если этот флаг не

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

установлен, пользователю рекомендуется самому решать, что делать в случае той или иной ошибки.

6.11.4 ЧислоВРем

6.11.4.1 Назначение

Алгоритм предназначен для перевода входного значения в формат контроллера.

6.11.4.2 Параметры

Наименование	Обознач.	Тип	Номер	Нач. знач.	Назначение
Входы	X	Настраиваемый	1		Входное значение
	Бит	Целый	2		Имя параметра объекта
	Недост	Целый	2		Недостоверность аналогового сигнала
Выходы	Y	Целый	1		Значение в формате контроллера

Примечания:

1 Вход **Бит** появляется, если в Мастере выбран формат преобразования **Бит**.

2 Вход **Недост** появится если в Мастере выбран формат преобразования **Аналоговый**.

6.11.4.3 Описание алгоритма

Алгоритм поддерживает следующие форматы: **целый, аналоговый, коэффициент пропорциональности, коэффициент масштабирования, время, скорость, дискретный и бит.**

Алгоритм имеет один обязательный вход – '**X**' и один дополнительный. Тип входа '**X**' определяется форматом преобразования и может быть – целым (ф-т преобразования – целый), числом с плавающей точкой (ф-ты преобразования аналоговый, коэффициент пропорциональности, коэффициент масштабирования, время, скорость) и логическим (ф-ты преобразования дискретный и бит). Формат преобразования выбирается в Мастере настроек. Наличие дополнительного входа определяется форматом преобразования. Если в Мастере настроек выбран аналоговый формат, то у алгоритма появится вход достоверность **Недост**. Если выбрать формат преобразования бит, то появится вход **Бит**. Преобразованное значение выводится на выход **Y**.

6.11.4.4 Мастер настроек

Мастер настроек позволяет выбрать один из форматов преобразования:

- целый;
- аналоговый;
- дискретный;
- коэффициент пропорциональности;
- коэффициент масштабирования;
- время;
- скорость;
- бит.

6.11.5 РемВЧисло

6.11.5.1 Назначение

Алгоритм предназначен для перевода входного значения в формате контроллера в соответствующий обычный формат.

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

6.11.5.2 Параметры

Наименование	Обознач.	Тип	Номер	Нач. знач.	Назначение
Входы	X	Целый	1		Входное значение в формате контроллера
	Бит	Целый	2		Имя параметра объекта
Выходы	Y	Настраиваемый	1		Значение в обычном формате
	Недост	Целый	2		Недостоверность аналогового сигнала

Примечания:
 1 Вход **Бит** появляется, если в Мастере выбран формат преобразования **Бит**.
 2 Выход **Недост** появится если в Мастере выбран формат преобразования **Аналоговый**.

6.11.5.3 Описание алгоритма

Алгоритм поддерживает следующие форматы – **целый, аналоговый, коэффициент пропорциональности, коэффициент масштабирования, время, скорость, дискретный и бит.**

Алгоритм имеет 1 обязательный вход – '**X**' и один дополнительный. Формат преобразования выбирается в мастере настроек. Наличие дополнительного входа определяется форматом преобразования. Если выбрать формат преобразования бит, то появится вход **Бит**. Преобразованное значение выводится на выход '**Y**'. Тип выхода '**Y**' определяется форматом преобразования и может быть: целым (ф-т преобразования – целый), числом с плавающей точкой (ф-ты преобразования аналоговый, коэффициент пропорциональности, коэффициент масштабирования, время, скорость) и логическим (ф-ты преобразования дискретный и бит). Если в мастере настроек выбран аналоговый формат, то у алгоритма появится выход **Недост**.

6.11.5.4 Мастер настроек

Мастер настроек позволяет выбрать один из форматов преобразования:

- целый;
- аналоговый;
- дискретный;
- коэффициент пропорциональности;
- коэффициент масштабирования;
- время;
- скорость;
- бит.

6.11.6 ИВЫ

6.11.6.1 Назначение

Алгоритм предназначен для передачи сигналов, сформированных контроллером, другим контроллерам или архиву по сетевому интерфейсу.

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

6.11.6.2 Параметры

Наименование	Обознач.	Тип	Номер	Нач. знач.	Назначение
Входы	<i>№ с. мод</i>	Целый	1	1	Номер системного модуля-приемника
	<i>№ конт</i>	Целый	2	1	Номер контроллера-приемника
	<i>№ гр</i>	Целый	3	1	Номер группы интерфейсного вывода
	<i>Период</i>	Действительный	4	0	Период передачи
	<i>Разрешить</i>	Логический	5	0	Разрешить передачу
	<i>X1</i>	Целый	6	0	Передаваемые сигналы
	<i>..</i>	Целый	5+n	0	Передаваемые сигналы
	<i>Xn</i>	Целый	5	Да	Передаваемые сигналы

6.11.6.3 Описание алгоритма

На входах **№ с. мод** и **№ конт** задается сетевой адрес приемника. Параметр **№ с. мод** должен находиться в диапазоне 1..254, а параметр **№ конт** – в диапазоне 0..255. Эти параметры устанавливаются согласно следующим правилам:

- если сообщение направляется другому контроллеру, то в качестве **№ с. мод** указывается номер системного модуля-приемника и в качестве **№ конт** – номер контроллера-приемника;
- если сообщение направляется в архив, то в качестве **№ с. мод** указывается сетевой номер Архивной станции и устанавливается **№ конт** = 0.

На входе **№ гр** задается номер группы сигналов, передаваемой через интерфейсный канал. Если сообщение направляется другому контроллеру, в нем алгоблок информационного ввода, который должен принимать данное сообщение, должен иметь в настройке адреса тот же номер группы **№ гр**. Значение **№ гр** может находиться в диапазоне 1-32, причем одно конкретное значение может задаваться в разных алгоритмах **ИБВЫ** только, если в них заданы разные сетевые адреса приемка.

На входе **Период** устанавливается период передачи информации. На входе **Разрешить** устанавливается **Да** если передача разрешена и **Нет** в противном случае.

Алгоритм имеет переменное число входов **X**, на которые подаются передаваемые значения. Количество входов задается с помощью Мастера настройки в пределах от 1 до 121. Чтобы корректно принять передаваемые сигналы, в контроллере-приемнике должен быть предусмотрен алгоблок с алгоритмом интерфейсного ввода **ИБВВ**, имеющим такое же число каналов (для Пилона) или выходов (для Мезона), что и соответствующий передающий алгоблок, и корректно заданный адрес передатчика.

При наличии ошибки передача информации не ведется.

6.11.6.4 Мастер настроек

Мастер настроек позволяет настроить количество входов **X**.

6.11.7 ИВВ Назначение

Алгоритм предназначен для приема сигналов от другого контроллера по сетевому интерфейсу.

Ивв. № подл.	Подп. и дата	Взаим. ивв. №	Ивв. №	Подп. и дата

6.11.7.2 Параметры

Наименование	Обознач.	Тип	Номер	Нач. знач.	Назначение
Входы	<i>№ с. мод</i>	Целый	1	1	Номер системного модуля-источника
	<i>№ конт</i>	Целый	2	1	Номер контроллера-источника
	<i>№ гр</i>	Целый	3	1	Номер группы источника
	<i>Тк</i>	Действительный	4	0	Контрольное время приема
Выходы	<i>Статус</i>	Целый	1		Статус приема
	<i>Y1</i>	Целый	2		Принимаемые сигналы
	<i>...</i>	Целый	i		Принимаемые сигналы
	<i>Yn</i>	Целый	n		Принимаемые сигналы

6.11.7.3 Описание алгоритма

На входах **№ с. мод** и **№ конт** задается сетевой адрес источника, которым является алгоритм интерфейсного вывода **ИВЫ** другого контроллера. Параметр **№ с. мод** должен находиться в диапазоне 1..254, а параметр **№ конт** – в диапазоне 1..255. На входе **№ гр** задается номер группы сигналов, тот же номер группы, что и у источника. Значение **№ гр** может находиться в диапазоне 1-32.

На входе **Тк** задается контрольное время приема. Обычно оно задается с превышением (например, в 3-5 раз) по отношению к периоду передачи, установленному в контроллере-источнике. Если за контрольное время значения не были получены на выходе **Статус** устанавливается «1».

Алгоритм имеет переменное число выходов **Y**, на которые выводятся принятые сигналы. Количество выходов задается с помощью мастера настройки в пределах от 1 до 121.

6.11.7.4 Мастер настроек

Мастер настроек позволяет настроить количество выходов **Y**.

6.11.8 Читать Внешний Параметр

6.11.8.1 Назначение

Алгоритм предназначен для чтения текущего значения параметра какого либо объекта Квинта из архива другого проекта.

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

Изм	Лист	№ докум	Подп.	Дата	ПФДИ.421457.009 И3.6	Лист
						134

6.11.8.2 Параметры

Наименование	Обознач.	Тип	Номер	Нач. знач.	Назначение
Входы	<i>Объект</i>	Целый	1	0	Идентификатор объекта
	<i>Параметр</i>	Целый	2	0	Номер параметра объекта
	<i>Архивная станция</i>	Строковый	3		Сетевое имя компьютера, на котором запущена Архивная станция другого проекта
	<i>Алгоритм "Ошибка"</i>	Логический	4	Нет	Признак вызова алгоритма "Ошибка"
Выходы	<i>Готово</i>	Логический	1	Нет	Признак завершения запроса параметра (Да - получено значение или ошибка)
	<i>ЕстьЗн</i>	Логический	2	Нет	Признак получения значения параметра
	<i>НедостЗн</i>	Логический	3	Нет	Признак недостоверности значения параметра
	<i>ФизЗн</i>	Действительный	4	0	Физическое значение параметра
	<i>Время</i>	Дата и время	5	0	Дата и время получения значения параметра
	<i>Статус</i>	Векторный	6	2	Результат выполнения операции чтения параметра
	<i>Ошибка</i>	Целый	7	0	Код ошибки

6.11.8.3 Описание алгоритма

Алгоритм имеет 4 входа и 7 выходов.

Алгоритм пытается получить из указанного архива текущее значение параметра, номер которого и идентификатор объекта заданы на входах **Объект** и **Параметр**.

Сетевое имя компьютера, где запускается Архивная станция другого проекта, задается на входе **Архивная станция**. Значения на входах задаются вручную или конфигурированием этих входов с выходами других алгоритмов.

В качестве сетевого имени компьютера допускается короткое **NetBIOS-имя**, полное **DNS-имя** и **IP-адрес**. В дублированной сети задается полное имя. В случае дублированного архива следует указать имена обеих Архивных станций, разделенных точкой с запятой. Пример: `srv1.domain.local; srv2.domain.local`.

Если идентификатор объекта равен 0 и (или) номер параметра равен 0 и (или) Архивная станция не задана, алгоритм выполняться не будет.

Требуемые значения идентификаторов объектов и номеров параметров можно найти одним из двух способов:

- при помощи программного приложения **Администратор серверов Квинта**, выбрав для требуемой Архивной станции узел **Данные/Параметры технологических объектов**. В этом узле раскрывается список технологических параметров, каждый элемент которого содержит марку объекта, а в квадратных скобках – категорию архивных данных (она равна 1), идентификатор объекта и номер параметра. Найдя нужную марку можно определить идентификатор объекта и номер параметра;
- при помощи программного приложения **Аркада**, выбрав закладку **Обзор**. В этой закладке перечисляются параметры выбранного объекта с указанием их номеров. Таким образом на форме **Объекты** можно определить идентификатор объекта (поле Идентификатор), а в указанной закладке – номер параметра.

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

При успешном выполнении на выход **ФизЗн** заносится физическое значение параметра, на выход **Время** - дата и время получения значения, а на выходы **Готово** и – **«Да»**, на выходы **Статус** и **Ошибка** - число 0 а на выход **НедостЗн** - значение «**Нет**».

При выполнении алгоритма могут возникнуть следующие ситуации, которые в задаче можно отследить и обработать :

- значение параметра пока не получено. На выходы **Готово**, **ЕстьЗн** и **НедостЗн** – заносится значение **«Нет»**, а на выход **Статус** - число 2 (бит 02 = 1);
- считано недостоверное значение. На выходы **Готово**, **ЕстьЗн** и **НедостЗн** – заносится значение **«Да»**, на выход **Статус** заносится число 1 (бит 01 = 1);
- ошибка от архива. На выход **Готово** заносится значение **«Да»**, на выходы **ЕстьЗн** и **НедостЗн** – значение **«Нет»**, на выход **Статус** - число 4 (бит 03 = 1), на выход **Ошибка** - код ошибки.

Пользователь может анализировать значения на указанных выходах и решать, что делать в случае той или иной ситуации.

Управление прерыванием выполнения задачи в случае ошибки осуществляется установкой признака вызова Алгоритма **Ошибка** в значение **«Да»** или **«Нет»**.

6.11.9 История 2

6.11.9.1 Назначение

Алгоритм предназначен для выбора из архива одного значения параметра какого либо объекта Квинта по заданным признакам отбора значений на заданном интервале времени.

Инв. № подл.	Подп. и дата				Инв. №	Подп. и дата				Взаим. инв. №	Подп. и дата				Инв. № подл.
					ПФДИ.421457.009 И3.6					Лист					
										136					
Изм		Лист		№ докум		Подп.		Дата							

6.11.9.2 Параметры

Наименование	Обознач.	Тип	Номер	Нач. знач.	Назначение
Входы	<i>Марка</i>	Строковый	1		Марка объекта
	<i>Параметр</i>	Строковый	2		Имя параметра объекта
	<i>НачВр</i>	Дата и время	3	0	Дата и время начала интервала времени
	<i>КонВр</i>	Дата и время	4	0	Дата и время конца интервала времени
	<i>Шаг</i>	Длительность	5	0	Интервал просеивания по времени
	<i>Повторить</i>	Событийный	6	0	Событие повторения выборки значений параметра
Выходы	<i>ЕстьЗн</i>	Логический	1	Нет	Признак получения значения параметра
	<i>НедостЗн</i>	Логический	2	Нет	Признак недостоверности значения параметра
	<i>Конец</i>	Логический	3	Нет	Признак конца значений параметра
	<i>ФизЗн</i>	Действительный	4		Физическое значение параметра
	<i>ПроцЗн</i>	Действительный	5		Значение параметра в процентах
	<i>Текст</i>	Строковый	6		Текстовое значение параметра или описание ошибки
	<i>Время</i>	Дата и время	7	0	Дата и время текущего шага или дата и время записи значения параметра в архив
	<i>ВремяЗн</i>	Дата и время	8	0	Дата и время записи значения параметра в архив
	<i>Статус</i>	Векторный	9	2	Результат выполнения операции поиска параметра
	<i>Ошибка</i>	Целый	10	0	Код ошибки
	<i>КолЗн</i>	Целый	11	0	Количество выведенных значений параметра, в том числе найденных в архиве и добавленных
	<i>КолЗнАр</i>	Целый	12	0	Количество значений параметра, найденных в архиве
	<i>Размерн.</i>	Строковый	13		Техническая размерность параметра
	<i>МинЗн</i>	Действительный	14		Минимальное значение параметра
	<i>МаксЗн</i>	Действительный	15		Максимальное значение параметра

6.11.9.3 Описание алгоритма

Алгоритм имеет 6 входов, 4 обязательных и 11 дополнительных выходов. Обязательными выходами являются **ЕстьЗн**, **ФизЗн**, **Время** и **КолЗн**. Дополнительные выходы назначаются в Мастере настроек.

Марка и параметр задаются в Мастере настроек, или редактированием значений на входах или конфигурированием этих входов. Если марка не задана алгоритм выполняться не будет. Если не задан параметр, алгоритм выберет по системной Базе данных Главный параметр объекта. Алгоритм не будет выполняться и в случае, когда марка и (или) параметр не найдены в Базе данных.

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

Изм	Лист	№ докум	Подп.	Дата

Алгоритм позволяет ограничить большой массив архивированных данных (просеивать выбираемые значения), если задать интервал времени между запрашиваемыми данными и указать, какие данные должны выбираться на каждом интервале. Величина интервала времени просеивания задается на входе **Шаг** и выражается в секундах, а выбираемые данные при помощи специальных признаков задаются в **Мастере настроек**. Просеивание данных производится на Архивной станции и позволяет уменьшить сетевой трафик.

Алгоритм может выполняться по одному из двух вариантов:

- 1 Этот вариант задается в Мастере настроек сбросом флага **Добавлять недостающие значения параметра** и позволяет просто выбирать из архива накопленные данные, т.е. получать по одному значению параметра в интервале времени, заданном входами **НачВр** и **КонВр**. Если **НачВр** меньше или равно **КонВр**, то поиск данных в архиве производится в прямом направлении, если же **НачВр** больше **КонВр**, то в обратном. Если **Шаг** равен 0, то выбираются все значения параметра от **НачВр** до **КонВр**, если **Шаг** не равен 0, то на каждом временном отрезке, равном значению входа **Шаг**, из архива выбирается по одному значению параметра по каждому признаку. Если в Мастере настроек установлен флаг Недостовренные значения, то будут выбираться и недостоверные значения.
- 2 Этот вариант задается в Мастере настроек установкой флага **Добавлять недостающие значения параметра** и не нулевым значением на входе **Шаг**. По этому варианту, который выполняется только в прямом направлении и только для первых значений на каждом шаге, накопленные в архиве данные дополняются недостающими значениями параметра. То есть, если на текущем шаге значение параметра не найдено, считается, что оно равно значению параметра предыдущего шага. В этом случае значение параметра предыдущего шага добавляется (выводится) со временем текущего шага. Следует иметь в виду, что перед началом выборки значений из архива алгоритм ищет значение параметра за время предшествующее началу интервала **НачВр - КонВр**. Оно может быть использовано в качестве предыдущего значения до тех пор, пока не будет найдено значение параметра внутри интервала. Если очередное значение параметра не найдено или найдено за время, превышающее конец интервала, значение параметра предыдущего шага будет добавлено (выведено) нужное количество раз до достижения конца интервала. Т.о. количество выведенных алгоритмом значений параметра может превышать количество накопленных в архиве значений.

Если значение параметра успешно получено или добавлено, то на выход **ЕстьЗн** заносится значение **Да**, на выход **ФизЗн** - физическое значение параметра, на выход **Время** по первому варианту заносится дата и время записи значения в архив, а по второму варианту – дата и время текущего шага. Значение на выходе **КолЗн** увеличивается на единицу.

Если заданы дополнительные выходы, то они заполняются следующим образом:

- на выход **ВремяЗн** заносится дата и время полученного очередного значения из архива;
- значение на выходе **КолЗнАр** увеличивается на единицу только при получении очередного значения параметра из архива;
- выходы **Ошибка** и **Статус** принимают значение «0»;
- на выход **НедостЗн** заносится значение **«Нет»** для достоверного значения параметра или значение **«Да»** для недостоверного значения;
- выход **Текст** заполняется округленным текстовым значением достоверного параметра или округленным текстовым недостоверным значением параметра с предшествующим символом **«@»**;
- на выход **Размерн.** заносится техническая размерность параметра;
- на выходы **МинЗн** и **МаксЗн** заносятся минимальное и максимальное значение параметра соответственно;
- на выход **ПроцЗн** заносится значение сигнала в процентах;
- выход **Конец** принимает значение **«Нет»**.

Если при очередном выполнении алгоритма на указанном интервале времени значений параметра больше нет, то значение на выходе **КолЗн** не меняется, на выход **Статус** заносится число 8, а на выход **Конец** – значение **«Да»**. После чего за этим параметром алгоритм к архиву обращаться не будет.

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

Возобновить работу алгоритма можно одним из двух способов :

- послать событие на вход **Повторить**;
- заменить значения на входах **Марка** и (или) **Параметр**.

При возобновлении работы алгоритма заново считываются значения со входов **НачВр**, **КонВр** и **Шаг**.

При выполнении алгоритма могут возникнуть следующие ситуации, которые в задаче можно отследить и обработать :

- считано недостоверное значение. На выход **Статус** заносится число 1 (бит 01 = 1)
- ошибка обращения к архиву. На выход **Статус** заносится число 4 (бит 03 = 1), на выход **Ошибка** - код ошибки, а на выход **Текст** – описание ошибки.

6.11.9.4 Мастер настроек

Мастер настроек позволяет установить значения на входах **Марка** и **Параметр**, задать вариант выполнения алгоритма, установить нужное количество дополнительных выходов алгоритма, определить совокупность признаков отбора значений и задать автоматическое прерывание задачи в случае ошибки при выборе значения параметра.

Установка значений на входах **Марка** и **Параметр** осуществляется нажатием кнопки **Выбрать параметр**.

Управление вариантами работы алгоритма осуществляется установкой флага **Добавлять недостающие значения параметра**.

Признаки, по которым будут выбираться данные на каждом шаге заданного интервала времени, устанавливаются при помощи флага группы **Признаки отбора значений**. В том числе: первое значение, последнее, минимальное или максимальное.

Если алгоритму назначено выполнение по первому варианту, установка от одного до нескольких флагов приведет к тому, что на каждом шаге по каждому признаку будет выбираться одно соответствующее ему значение. Причем, повторяющиеся значения выводиться не будут. То есть, если в текущем шаге при установке всех флагов будут найдены всего два значения: первое и последнее, и они совпадут с минимальным и максимальным, то алгоритм выведет только два значения, где первое значение является минимальным, а последнее – максимальным. Если в текущем шаге при установке всех флагов будут найдено всего одно значение, то оно будет выведено один раз и оно будет первым, последним, минимальным и максимальным.

Если алгоритму назначено выполнение по второму варианту, то на каждом шаге интервала времени принудительно будут выбираться только первые значения. Если будут установлены другие флаги, то они будут проигнорированы.

Дополнительные выходы назначаются совокупностью флагов группы **Дополнительные выходы**.

Управление прерыванием выполнения задачи в случае ошибки осуществляется установкой флага **Если ошибка - вызывать алгоритм Ошибка**. Если этот флаг не установлен, пользователю рекомендуется самому решать, что делать в случае той или иной ошибки.

6.11.10 История Внешнего Параметра

6.11.10.1 Назначение

Алгоритм предназначен для выбора из архива другого проекта одного значения параметра какого либо объекта Квинта по заданным признакам отбора значений на заданном интервале времени.

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

Изм	Лист	№ докум	Подп.	Дата	ПФДИ.421457.009 И3.6	Лист
						139

6.11.10.2 Параметры

Наименование	Обознач.	Тип	Номер	Нач. знач.	Назначение
Входы	<i>Объект</i>	Целый	1	0	Идентификатор объекта
	<i>Параметр</i>	Целый	2	0	Номер параметра объекта
	<i>Архивная станция</i>	Строковый	3		Сетевое имя компьютера, на котором запущена архивная станция другого проекта
	<i>НачВр</i>	Дата и время	4	0	Дата и время начала интервала времени
	<i>КонВр</i>	Дата и время	5	0	Дата и время конца интервала времени
	<i>Шаг</i>	Длительность	6	0	Интервал просеивания по времени
	<i>Повторить</i>	Событийный	7	0	Событие повторения выборки значений параметра
Выходы	<i>ЕстьЗн</i>	Логический	1	Нет	Признак получения значения параметра
	<i>НедостЗн</i>	Логический	2	Нет	Признак недостоверности значения параметра
	<i>ФизЗн</i>	Действительный	3	0	Физическое значение параметра
	<i>Текст</i>	Строковый	4	0	Текстовое значение параметра
	<i>Время</i>	Дата и время	5	0	Дата и время текущего шага или дата и время записи значения параметра в архив
	<i>ВремяЗн</i>	Дата и время	6	0	Дата и время записи значения параметра в архив
	<i>КолЗн</i>	Целый	7	0	Количество выведенных значений параметра, в том числе найденных в архиве и добавленных
	<i>КолЗнАр</i>	Целый	8	0	Количество значений параметра, найденных в архиве
	<i>Статус</i>	Векторный	9	2	Результат выполнения операции поиска параметра
	<i>Ошибка</i>	Целый	10	0	Код ошибки
	<i>Конец</i>	Логический	11	Нет	Признак конца значений параметра

6.11.10.3 Описание алгоритма

Алгоритм имеет 7 входов, 4 обязательных и 7 дополнительных выходов. Обязательными выходами являются **ЕстьЗн**, **ФизЗн**, **Время** и **КолЗн**. Дополнительные выходы назначаются в Мастере настроек.

Алгоритм пытается получить из архива значение параметра с номером, заданном на входе **Параметр**, и идентификатором объекта, заданном на входе **Объект**. Сетевое имя компьютера, где запускается Архивная станция другого проекта, задается на входе **Архивная станция**. В качестве сетевого имени компьютера допускается короткое **NetBIOS-имя**, полное **DNS-имя** и **IP-адрес**. В дублированной сети задается полное имя. В случае дублированного архива следует указать имена обеих архивных станций, разделенных точкой с запятой. Пример: `srv1.domain.local;srv2.domain.local`.

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

Требуемые значения идентификаторов объектов и номеров параметров можно найти одним из двух способов:

- при помощи программного приложения **Администратор серверов Квинта**, выбрав для требуемой Архивной станции узел **Данные/Параметры технологических объектов**. В этом узле раскрывается список технологических параметров, каждый элемент которого содержит марку объекта, а в квадратных скобках – категорию архивных данных (она равна 1), идентификатор объекта и номер параметра. Найдя нужную марку можно определить идентификатор объекта и номер параметра;
- при помощи программного приложения **Аркада**, выбрав закладку **Обзор**. В этой закладке перечисляются параметры выбранного объекта с указанием их номеров. Таким образом на форме **Объекты** можно определить идентификатор объекта (поле Идентификатор), а в указанной закладке – номер параметра.

Значения на входах задаются вручную или конфигурированием этих входов с выходами других алгоритмов.

Если **Идентификатор** объекта равен 0 и (или) **номер параметра** равен 0 и (или) **Архивная станция** не задана, алгоритм выполняться не будет.

Алгоритм позволяет ограничить большой массив архивированных данных (просеивать выбираемые значения), если задать интервал времени между запрашиваемыми данными и указать, какие данные должны выбираться на каждом интервале. Величина интервала времени просеивания задается на входе **Шаг** и выражается в секундах, а выбираемые данные при помощи специальных признаков задаются в Мастере настроек. Просеивание данных производится на Архивной станции и позволяет уменьшить сетевой трафик.

Алгоритм может выполняться по одному из двух вариантов:

- 1 Этот вариант задается в Мастере настроек сбросом флага **Добавлять недостающие значения параметра** и позволяет просто выбирать из архива накопленные данные, т.е. получать по одному значению параметра в интервале времени, заданном входами **НачВр** и **КонВр**. Если **НачВр** меньше или равно **КонВр**, то поиск данных в архиве производится в прямом направлении, если же **НачВр** больше **КонВр**, то в обратном. Если **Шаг** равен 0, то выбираются все значения параметра от **НачВр** до **КонВр**, если **Шаг** не равен 0, то на каждом временном отрезке, равном значению входа **Шаг**, из архива выбирается по одному значению параметра по каждому признаку. Если в Мастере настроек установлен флаг Недостоверные значения, то будут выбираться и недостоверные значения.
- 2 Этот вариант задается в Мастере настроек установкой флага **Добавлять недостающие значения параметра** и не нулевым значением на входе **Шаг**. По этому варианту, который выполняется только в прямом направлении и только для первых значений на каждом шаге, накопленные в архиве данные дополняются недостающими значениями параметра. То есть, если на текущем шаге значение параметра не найдено, считается, что оно равно значению параметра предыдущего шага. В этом случае значение параметра предыдущего шага добавляется (выводится) со временем текущего шага. Следует иметь в виду, что перед началом выборки значений из архива алгоритм ищет значение параметра за время предшествующее началу интервала **НачВр - КонВр**. Оно может быть использовано в качестве предыдущего значения до тех пор, пока не будет найдено значение параметра внутри интервала. Если очередное значение параметра не найдено или найдено за время, превышающее конец интервала, значение параметра предыдущего шага будет добавлено (выведено) нужное количество раз до достижения конца интервала. Т.о. количество выведенных алгоритмом значений параметра может превышать количество накопленных в архиве значений.

Если значение параметра успешно получено или добавлено, то на выход **ЕстьЗн** заносится значение **Да**, на выход **ФизЗн** - физическое значение параметра, на выход **Время** по первому варианту заносится дата и время записи значения в архив, а по второму варианту – дата и время текущего шага. Значение на выходе **КолЗн** увеличивается на единицу.

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

Если заданы дополнительные выходы, то они заполняются следующим образом:

- на выход **ВремяЗн** заносится дата и время полученного очередного значения из архива;
- значение на выходе **КолЗнАр** увеличивается на единицу только при получении очередного значения параметра из архива;
- выходы **Ошибка** и **Статус** принимают значение «0»;
- на выход **НедостЗн** заносится значение “Нет” для достоверного значения параметра или значение “Да” для недостоверного значения;
- выход **Текст** заполняется округленным текстовым значением достоверного параметра или округленным текстовым недостоверным значением параметра с предшествующим символом “@”;
- на выход **Размерн.** заносится техническая размерность параметра;
- на выходы **МинЗн** и **МаксЗн** заносятся минимальное и максимальное значение параметра соответственно;
- на выход **ПроцЗн** заносится значение сигнала в процентах;
- выход **Конец** принимает значение “Нет”.

Если при очередном выполнении алгоритма на указанном интервале времени значений параметра больше нет, то значение на выходе **КолЗн** не меняется, на выход **Статус** заносится число 8, а на выход **Конец** – значение “Да”. После чего за этим параметром алгоритм к архиву обращаться не будет.

Возобновить работу алгоритма можно одним из двух способов :

- послать событие на вход **Повторить**;
- заменить значения на входах **Марка** и (или) **Параметр**.

При возобновлении работы алгоритма заново считываются значения со входов **НачВр**, **КонВр** и **Шаг**.

При выполнении алгоритма могут возникнуть следующие ситуации, которые в задаче можно отследить и обработать :

- считано недостоверное значение. На выход **Статус** заносится число 1 (бит 01 = 1)
- ошибка обращения к архиву. На выход **Статус** заносится число 4 (бит 03 = 1), на выход **Ошибка** - код ошибки, а на выход **Текст** – описание ошибки.

6.11.10.4 Мастер настроек

Мастер настроек позволяет задать вариант выполнения алгоритма, установить нужное количество дополнительных выходов алгоритма, определить совокупность признаков отбора значений и задать автоматическое прерывание задачи в случае ошибки выбора параметра.

Управление вариантами работы алгоритма осуществляется установкой флага **Добавлять недостающие значения параметра**.

Признаки, по которым будут выбираться данные на каждом шаге заданного интервала времени, устанавливаются при помощи флагов группы **Признаки отбора значений**. В том числе: первое значение, последнее, минимальное или максимальное.

Если алгоритму назначено выполнение по первому варианту, установка от одного до нескольких флагов приведет к тому, что на каждом шаге по каждому признаку будет выбираться одно соответствующее ему значение. Причем, повторяющиеся значения выводиться не будут. То есть, если в текущем шаге при установке всех флагов будут найдены всего два значения: первое и последнее, и они совпадут с минимальным и максимальным, то алгоритм выведет только два значения, где первое значение является минимальным, а последнее – максимальным. Если в текущем шаге при установке всех флагов будут найдено всего одно значение, то оно будет выведено один раз и оно будет первым, последним, минимальным и максимальным.

Если алгоритму назначено выполнение по второму варианту, то на каждом шаге интервала времени принудительно будут выбираться только первые значения. Если будут установлены другие флаги, то они будут проигнорированы.

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

Дополнительные выходы назначаются совокупностью флагов группы **Дополнительные выходы**.

Управление прерыванием выполнения задачи в случае ошибки осуществляется установкой флага **Если ошибка - вызывать алгоритм Ошибка**. Если этот флаг не установлен, пользователю рекомендуется самому решать, что делать в случае той или иной ошибки.

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

Изм	Лист	№ докум	Подп.	Дата

ПФДИ.421457.009 И3.6

6.12 Группа 'Строки'

6.12.1 РегСтр

6.12.1.1 Параметры

Наименование	Обознач.	Тип	Номер	Нач. знач.	Мин. знач.	Макс. знач.	Назначение
Входы	<i>Строка</i>	строковый	1	-	-	-	Исходная строка
	<i>Регистр</i>	целый	2	0	0	2	Признак регистра
Выходы	<i>Результат</i>	строковый	-	-	-	-	Измененная строка

6.12.1.2 Назначение

Измерение регистра символов в строке.

6.12.1.3 Описание алгоритма

Алгоритм преобразует регистр символов строки на входе **Строка** согласно значению на входе **Регистр**: **0** - Верхний регистр; **1** - Нижний регистр; **2** - Инверсное преобразование. Результат помещается на выход **Результат**.

6.12.2 ДлСтр

6.12.2.1 Параметры

Наименование	Обознач.	Тип	Номер	Нач. знач.	Мин. знач.	Макс. знач.	Назначение
Входы	<i>Строка</i>	строковый	1	-	-	-	Исходная строка
Выходы	<i>Длина</i>	целый	-	-	-	-	Количество символов в строке

6.12.2.2 Назначение

Определение длины строки

6.12.2.3 Описание алгоритма

Алгоритм определяет длину строки на входе **Строка**. Результат помещается на выход **Длина**.

6.12.3 Сцепить

6.12.3.1 Параметры

Наименование	Обознач.	Тип	Номер	Нач. знач.	Мин. знач.	Макс. знач.	Назначение
Входы	<i>Разделитель</i>	строковый	1	-	-	-	Разделитель между строками
	<i>Строка</i>	строковый	2	-	-	-	1-я строка
	<i>Строка</i>	строковый	3	-	-	-	...
	<i>...</i>	строковый	n	-	-	-	n-я строка
Выходы	<i>Результат</i>	строковый	-	-	-	-	Результат конкатенации n строк и разделителя

6.12.3.2 Назначение

Конкатенация N строк.

6.12.3.3 Описание алгоритма

Алгоритм предназначен для конкатенации N строк с использованием разделителя.

Инд. №	Подп. и дата

6.12.3.4 Мастер настройки

Количество строк задается с помощью мастера. Минимальное количество строк - 2, максимальное 50;

6.12.4 ПодСтр

6.12.4.1 Параметры

Наименование	Обознач.	Тип	Номер	Нач. знач.	Мин. знач.	Макс. знач.	Назначение
Входы	<i>Строка</i>	строковый	1	-	-	-	Исходная строка
	<i>Позиция</i>	целый	2	1	1	-	Позиция в строке, начиная с 1
	<i>ДлПодстр</i>	целый	3	-	-	-	Длина подстроки
Выходы	<i>Подстр</i>	строковый	-	-	-	-	Найденная подстрока

6.12.4.2 Назначение

Выделение подстроки указанного размера от указанной позиции

6.12.4.3 Описание алгоритма

Алгоритм предназначен для выделения из строки на входе **Строка** от позиции на входе **Позиция** подстроки длиной **ДлПодстр**.

6.12.5 ПовторСтрПараметры

Наименование	Обознач.	Тип	Номер	Нач. знач.	Мин. знач.	Макс. знач.	Назначение
Входы	<i>Строка</i>	строковый	1	-	-	-	Исходная строка
	<i>Количество</i>	целый	2	1	1	-	Количество повторений
Выходы	<i>Результат</i>	строковый	-	-	-	-	Результат

6.12.5.2 Назначение

Повторение строки N раз.

6.12.5.3 Описание алгоритма

Алгоритм повторяет строку на входе **Строка** столько раз, сколько указано на входе **Количество**.

6.12.6 ПоискСтр

6.12.6.1 Параметры

Наименование	Обознач.	Тип	Номер	Нач. знач.	Мин. знач.	Макс. знач.	Назначение
Входы	<i>Строка</i>	строковый	1	-	-	-	Исходная строка
	<i>Подстр</i>	строковый	2	-	-	-	Подстрока
	<i>Позиция</i>	целый	3	1	1	-	Позиция в строке, начиная с 1
Выходы	<i>Результат</i>	целый	-	-	-	-	Позиция начала подстроки в исходной строке

6.12.6.2 Назначение

Поиск указанной подстроки от указанной позиции.

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

6.13 Группа 'События'

6.13.1 СчетСоб

6.13.1.1 Параметры

Наименование	Обознач.	Тип	Номер	Нач. знач.	Мин. знач.	Макс. знач.	Назначение
Входы	<i>Событие</i>	событийный	1	-	-	-	Событийный вход
	<i>Сброс</i>	событийный	2	-	-	-	Событие сброса счетчика
Выходы	<i>Количество</i>	целый	-	-	-	-	Количество событий

6.13.1.2 Назначение

Подсчет количества событий.

6.13.1.3 Описание алгоритма

Алгоритм предназначен для подсчета количества событий, пришедших на вход **Событие**. Счетчик обнуляется, если на вход **Сброс** приходит событие. Подсчет вновь пришедших событий возобновиться в следующем цикле выполнения.

6.13.2 СобИли

6.13.2.1 Параметры

Наименование	Обознач.	Тип	Номер	Нач. знач.	Мин. знач.	Макс. знач.	Назначение
Входы	<i>A1</i>	событийный	1	-	-	-	Событие
	<i>A2</i>	событийный	2	-	-	-	Событие
	...	событийный	-	-	-	-	Событие
	<i>Ап</i>	событийный	n	-	-	-	Событие
Выходы	<i>В</i>	событийный	-	-	-	-	Результирующее событие

6.13.2.2 Назначение

Событийное ИЛИ.

6.13.2.3 Описание алгоритма

Алгоритм генерирует на выходе *В* событие в том случае, если на его входах произошло хотя бы одно событие.

6.13.2.4 Мастер настройки

Алгоритм имеет настраиваемое с помощью мастера количество входов.

6.13.3 СобИ

6.13.3.1 Параметры

Наименование	Обознач.	Тип	Номер	Нач. знач.	Мин. знач.	Макс. знач.	Назначение
Входы	<i>A1</i>	событийный	1	-	-	-	Событие
	<i>A2</i>	событийный	2	-	-	-	Событие
	...	событийный	-	-	-	-	Событие
	<i>A3</i>	событийный	n	-	-	-	Событие
Выходы	<i>В</i>	событийный	-	-	-	-	Результирующее событие

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

6.13.3.2 Назначение

Событийное И.

6.13.3.3 Описание алгоритма

Алгоритм запоминает все события, которые произошли на его входах. Как только произойдут события на всех входах генерируется результирующее событие на выходе **В**. После этого алгоритм сбрасывает запомненные события.

6.13.3.4 Мастер настройки

Алгоритм имеет настраиваемое с помощью мастера количество входов.

6.13.4 СобВил

6.13.4.1 Параметры

Наименование	Обознач.	Тип	Номер	Нач. знач.	Мин. знач.	Макс. знач.	Назначение
Входы	<i>Условие</i>	логический	1	-	-	-	Условие, определяющее тип события
	<i>Событие</i>	событийный	2	-	-	-	Событие
Выходы	<i>Да</i>	событийный	-	-	-	-	Событие если условие - Да
	<i>Нет</i>	событийный	-	-	-	-	Событие если условие - Нет

6.13.4.2 Назначение

Событийная Вилка.

6.13.4.3 Описание алгоритма

Алгоритм генерирует на выходе **Да** событие в том случае, если на входе **Условие** присутствует «Да». Алгоритм генерирует на выходе **Нет** событие в том случае, если на входе **Условие** присутствует «Нет». События генерируются если на вход **СОБЫТИЕ** пришло событие.

6.13.5 ЕстьСоб

6.13.5.1 Параметры

Наименование	Обознач.	Тип	Номер	Нач. знач.	Мин. знач.	Макс. знач.	Назначение
Входы	А	событийный	1	-	-	-	Событие
Выходы	В	логический	-	-	-	-	При возникновении события - Да, иначе - Нет

6.13.5.2 Назначение

Преобразование события в логическое значение.

6.13.5.3 Описание алгоритма

Алгоритм выставляет на выходе **В** состояние «Да», если на вход **А** пришло событие или состояние «Нет» в противном случае.

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

6.14 Группа 'Управление'

6.14.1 RS-триггер

6.14.1.1 Назначение

Хранение двух устойчивых состояний (reset/set).

6.14.1.2 Параметры

Наименование	Обознач.	Тип	Назначение
Входы	Установ	событийный	Событие установка триггера
	Сброс	событийный	Событие сброса триггера. Событие приоритетное
Выходы	В	логический	Текущее состояние триггера

6.14.1.3 Описание алгоритма

Событие на входе **Установ** устанавливает на выходе **В** состояние «Да».

Событие на входе **Сброс** устанавливает на выходе **В** состояние «Нет».

6.14.2 Т-триггер

6.14.2.1 Назначение

Хранение двух устойчивых состояний.

6.14.2.2 Параметры

Наименование	Обознач.	Тип	Назначение
Входы	Установ	событийный	Событие установка триггера
	Сброс	событийный	Событие сброса триггера
	А	событийный	Событие изменения состояния триггера
Выходы	В	логический	Текущее состояние триггера

6.14.2.3 Описание алгоритма

Событие на входе **Установ** устанавливает на выходе **В** состояние «Да».

Событие на входе **Сброс** устанавливает на выходе **В** состояние «Нет». Событие сброса приоритетное.

Событие на входе **А** устанавливает на выходе **В** состояние «Да», если на нем перед приходом события было состояние «Нет», и устанавливает на выходе **В** состояние «Нет», если на нем перед приходом события было состояние «Да»

6.14.3 Таймер

6.14.3.1 Назначение

Предназначен для выполнения действий при истечении заданного интервала времени.

6.14.3.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Мин. знач.	Назначение
Входы	Интервал	Длительность	1	0.001	Время между срабатываниями таймера
	Пуск	событийный	-	-	Активирует таймер
	Сброс	событийный	-	-	Отменяет текущий отсчет времени
Выходы	Осталось	Длительность	-	-	Обратный отсчет времени до срабатывания
	Ошибка	целый	-	-	0 : ОК; 1 : Интервал < 0.001
	Сработал	событийный	-	-	Событие, возникающее при достижении нулевого значения на входе Осталось

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

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

6.14.3.3 Описание алгоритма

Входы:

- **Интервал.** Это время между срабатываниями таймера. В режиме выполнения при смене значения на этом входе, новое значение сразу же вступает в силу;
- **Пуск.** Если вход связан, то при возникновении на этом входе события начинается отсчет. Событие, пришедшее во время начавшегося отсчета игнорируется. Если вход не связан, то отсчет начинается после старта задачи;
- **Сброс.** Отменяет текущий отсчет времени, дальнейшее поведение определяется входом **Пуск**. Событие сброса приоритетное.

Выходы:

- **Осталось.** Это обратный отсчет времени до срабатывания;
- **Ошибка:**
 - а) равно 0, если всё в порядке;
 - б) равно 1, если ошибка;
 - в) меньше 0.001, если интервал;
- **Сработал.** Это событие, возникающее при достижении нулевого значения на входе.

6.14.4 Цикл

6.14.4.1 Назначение

Предназначен для выполнения повторяющихся действий.

6.14.4.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Мин. знач.	Назначение
Входы	Кол-во шагов	целый	1	1	Количество итераций в цикле
	Пуск	событийный	-	-	Активирует цикл
	Выход	событийный	-	-	Событие досрочного выхода из цикла
	Выполнился	событийный	-	-	Событие выполнения тела цикла
Выходы	Счетчик	целый	-	-	Текущее значение итератора цикла
	Активен	логический	-	-	Признак активности цикла
	Ошибка	целый	-	-	0 : ОК; 1 : Кол-во шагов < 1; 2 : Выполнился не связан; 3 : Кол-во шагов < Счетчик
	Закончен	событийный	-	-	Событие формируется, когда Кол-во шагов = Счетчик
	Новый цикл	событийный	-	-	Событие формируется при новом цикле

6.14.4.3 Описание алгоритма

Входы:

- **Кол-во шагов.** Это количество шагов (итераций) в цикле. Если в режиме выполнения количество шагов равно значению на выходе **Счетчик**, то возникает событие на выходе **Закончен**;
- **Пуск.** Это событие запуска цикла. Если вход связан, то цикл запускается по событию, иначе цикл запускается автоматически при старте задачи. Событие пуска приоритетно, если на выходе **Активен** признак «Нет»;
- **Выход.** Это событие досрочного выхода из цикла (переводит алгоритм в начальный режим и формирует событие на выходе **Закончен**). Событие выхода приоритетно, если на выходе **Активен** признак «Да»;
- **Выполнился.** Это событие выполнения тела цикла. Появление события приводит к следующей итерации цикла (изменение значения на выходе **Счетчик** и формирование события на выходе **Новый цикл**).

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

Выходы:

- **Счетчик.** Это текущее значение итератора цикла. Начальное значение равно 0. Увеличивается на 1 (инкрементируется) во время очередной итерации (т.е. при возникновении события на входе **Выполнился**);
- **Новый цикл.** Это событие формируется при первом цикле (по событию на входе **Пуск** или при старте задачи, если вход **Пуск** не связан) или при поступлении события на входе **Выполнился**;
- **Закончен.** Это событие формируется по условию, когда количество шагов равно значению на выходе **Счетчик**, при этом на выходе **Активен** устанавливается значение «Нет».

6.14.5 Шаг

6.14.5.1 Назначение

Предназначен для выполнения действий по таймауту или по условию.

6.14.5.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Мин. знач.	Назначение
Входы	Пуск	событийный	-	-	Активирует шаг
	Условие	логический	-	-	Если Условие = Да, то шаг деактивируется и выдает событие на выходе Да
	Таймаут	Длительность	0	0	0 : таймаут не используется; иначе : время, после которого шаг деактивируется
Выходы	Активен	логический	-	-	Признак активности шага
	Да	событийный	-	-	Событие возникает при завершении шага по условию
	Нет	событийный	-	-	Событие возникает при завершении шага по таймауту
	Осталось	Длительность	-	-	Сколько осталось ждать до таймаута

6.14.5.3 Описание алгоритма

Входы:

- **Пуск.** Активирует шаг, Если вход не связан, шаг активируется при запуске задачи;
- **Условие.** Если устанавливается «Да», то шаг завершается (деактивируется) и выдает событие на выходе **Да**;
- **Таймаут.** Если с момента активации прошло время \geq значению на входе **Таймаут**, шаг завершается (деактивируется) и выдает событие на выходе **Нет**. Если значению на входе **Таймаут** указано 0, шаг завершается только когда на входе **Условие** устанавливается значение «Да». Если на входе Таймаут значение < 0 , то тона выходе **Активен** устанавливается значение «Нет».

Выходы:

- **Активен.** Это признак активности шага;
- **Да.** Это событие выдается при завершении шага по условию;
- **Нет.** Это событие выдается при завершении шага по таймауту;
- **Осталось.** Сколько осталось ждать до таймаута.

6.14.6 Ошибка

6.14.6.1 Назначение

Алгоритм предназначен для аварийного останова выполнения задачи в которой он находится. Позволяет остановить задачу с определенным кодом ошибки, который будет передан при вызове функции **Finalize** всех алгоблоков в этой задаче. При этом статус задачи примет значение **jsAborted**.

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

Изм	Лист	№ докум	Подп.	Дата	ПФДИ.421457.009 И3.6	Лист
						151

6.14.6.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	<i>Код</i>	целый	4096	Код останова задачи
	<i>Треб.</i>	логический	Нет	Требование останова задачи (передний фронт сигнала)

6.14.6.3 Описание алгоритма

Алгоблок позволяет остановить задачу и «предупредить» все другие алгоблоки о том, что она аварийно остановлена с указанием кода ошибки. Код ошибки не может быть меньше 4096. Для повторного запуска задачи следует сформировать событие на ее входе **Пуск**, либо сформировать передний фронт логического сигнала на входе **Разрешение**.

6.14.7 Останов

6.14.7.1 Назначение

Алгоритм предназначен для штатного останова выполнения задачи в которой он находится.

Позволяет штатно остановить задачу с кодом «0», который будет передан при вызове функции **Finalize** всех алгоблоков в этой задаче. При этом статус задачи примет значение **jsAborted**.

6.14.7.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	<i>Треб.</i>	логический	Нет	Требование штатного останова задачи (передний фронт сигнала)

6.14.7.3 Описание алгоритма

Алгоблок позволяет штатно остановить задачу и «предупредить» все другие алгоблоки о том, что она остановлена с кодом ошибки «0» (штатный останов). Алгоблок полезен в случае, если дальнейшее продолжение работы задачи бессмысленно (например, основной ресурс, используемый каким-то алгоблоком задачи не найден).

6.14.8 Сначала

6.14.8.1 Назначение

Алгоритм предназначен для перезапуска выполнения задачи в которой он находится.

Позволяет штатно перезапустить задачу с произвольным кодом, который можно узнать, используя глобальное окружение сервера выполнения. При этом статус задачи, примет значение **jsBreak**.

6.14.8.2 Параметры

Наименование	Обознач.	Тип	Нач. знач.	Назначение
Входы	<i>Код</i>	целый	0	Код перезапуска задачи
	<i>Треб.</i>	логический	Нет	Требование перезапуска задачи (наличие лог.1)

6.14.8.3 Описание алгоритма

Алгоблок позволяет перезапустить задачу и заставить ее выполняться сначала. Если на входе **Треб.** сигнал «Да», то все алгоблоки, расположенные до него выполняются, а все алгоблоки, расположенные после – нет. Так будет продолжаться в каждом цикле выполнения задачи до тех пор, пока сигнал на входе **Треб.** не примет значение «Нет». Алгоблок полезен в случае, если выполнение алгоблоков, следующих за ним, бессмысленно (например, ресурс, используемый этими алгоблоками не доступен или данные для них пока не готовы).

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

6.15 Группа 'Файлы'

6.15.1 Выбор файла Назначение

Открывает файл для чтения или записи.

6.15.1.2 Параметры

Наименование	Обозначение	Тип	Назначение
Выходы	Файл	Контейнер данных	Указатель на файл

6.15.1.3 Описание алгоритма

Алгоритм открывает существующий файл для чтения или записи данных (или создает новый файл для записи данных) и возвращает указатель на открытый файл.

Данный указатель используется алгоритмами **Запись** и **Чтение**.

6.15.1.4 Мастер настройки

Позволяет указать имя файла и режим работы. В режиме записи данных можно указать, будут ли новые данные записываться поверх старых (если они существуют) или в конец файла. Имя открываемого файла вводится в текстовое поле или задается с помощью стандартного диалогового окна открытия файла. Если в режиме чтения указан несуществующий файл, новые настройки не будут приняты и появится окно с предупреждением. Если несуществующий файл указан в режиме записи, алгоритм создаст новый файл по указанному адресу. Кнопка **Просмотр файла** позволяет открыть выбранный файл в сопоставленном приложении.

6.15.2 Запись

6.15.2.1 Назначение

Записывает данные в файл

6.15.2.2 Параметры

Наименование	Обозначение	Тип	Назначение
Входы	Файл	Контейнер данных	Указатель на файл
	Данные	Настраиваемый	Тип записываемых данных
	Разреш	Логический	Флаг, разрешающий запись в файл
Выходы	Ошибка	Логический	Ошибка записи данных

6.15.2.3 Описание алгоритма

Алгоритм записывает значение выбранного в Мастере настройки типа со входа Данные в файл, на который указывает значение на входе **Файл** (Это значение можно получить на выходе алгоритма **Выбор файла**). Значение записывается на текущую позицию в файле. После записи текущая позиция смещается на число байт, равное размеру записанных данных.

Если несколько алгоблоков **Запись** работают с одним файлом, они используют общую текущую позицию.

Если значение на входе **Разреш** установлено «Нет» алгоритм не выполняется.

Если попытка записать данные в файл по каким-то причинам не удалась (со входа **Файл** получено неправильное значение указателя, попытка записи в файл с установленным атрибутом **Только чтение** и т.д.) значение на выходе Ошибка установится в «Да».

6.15.2.4 Мастер настройки

Позволяет выбрать тип записываемого в файл значения и формат файла.

Алгоритм позволяет записывать в бинарный файл следующие типы: **char**, **unsigned char**, **short**, **unsigned short**, **int**, **unsigned int**, **float**, **double**, строку с нулевым символом в конце (**string1**) и строку заданной длины (**string2**, размер строки хранится в первых четырех байтах).

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

В текстовый файл алгоритм позволяет записывать целые (**int**) и вещественные(**double**) числа, а также заканчивающиеся нулевым символом строки (**string**).

Если выбран тип данных **string**, **string1** или **string2** на вход **Данные** должны приходить значения строкового типа.

Если выбран тип **unsigned int**, **float** или **double** на вход **Данные** должны приходить значения вещественного типа. В остальных случаях должно приходить значение целого типа.

Если ко входу **Данные** проведена связь, Мастер настройки позволяет выбирать лишь те типы записываемых данных, которые не требуют изменения типа значения, приходящего на вход.

Остальные типы выделяются светло-серым цветом и не реагируют на попытку выбора.

6.15.3 ЧтениеНазначение

Считывает данные из файла

6.15.3.2 Параметры

Наименование	Обозначение	Тип	Назначение
Входы	<i>Файл</i>	Контейнер данных	Указатель на файл
	<i>Разреш</i>	Логический	Флаг, разрешающий считывание из файла
Выходы	<i>Данные</i>	Настраиваемый	Тип считываемых данных
	<i>Конец файла</i>	Логический	Достигнут конец файла
	<i>Ошибка</i>	Логический	Ошибка чтения данных

6.15.3.3 Описание алгоритма

Алгоритм считывает значение выбранного в Мастере настройки типа из файла, определяемого значением на входе **Файл**. (Это значение можно получить на выходе алгоритма **Выбор файла**), и отправляет его на выход **Данные**. Значения считываются с текущей позиции в файле. После считывания текущая позиция смещается на число байт, равное размеру считанных данных.

Если несколько алгоблоков **Чтение** работают с одним файлом, они используют общую текущую позицию.

Если значение на входе **Разреш** установлено «Нет» алгоритм не выполняется.

Если в процессе чтения достигнут конец файла значение на выходе **Конец файла** установится в «Да».

Если попытка считать данные из файла по каким-то причинам не удалась значение на выходе «Ошибка» установится в единицу.

6.15.3.4 Мастер настройки

Позволяет выбрать тип считываемых из файла данных и формат файла. Алгоритм позволяет считывать из бинарного файла следующие типы: **char**, **unsigned char**, **short**, **unsigned short**, **int**, **unsigned int**, **float**, **double**, строку с нулевым символом в конце (**string1**) и строку заданной длины(**string2**, размер строки хранится в первых четырех байтах).

Из текстового файла алгоритм позволяет считывать целые (**int**) и вещественные(**double**) числа, а также заканчивающиеся нулевым символом строки (**string**).

Если выбран тип данных **string**, **string1** или **string2** выход **Данные** содержит значения строкового типа.

Если выбран тип **unsigned int**, **float** или **double** выход **Данные** содержит значения вещественного типа.

В остальных случаях выход **Данные** содержит значения целого типа.

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

Если от выхода **Данные** проведена связь, Мастер настройки позволяет выбирать лишь те типы считываемых данных, которые не требуют изменения типа значения, находящегося на выходе.

Остальные типы выделяются светло-серым цветом и не реагируют на попытку выбора.

6.15.4 Процесс

6.15.4.1 Назначение

Запуск процессов.

6.15.4.2 Параметры

Наименование	Обознач.	Тип	Номер	Нач. знач.	Мин. знач.	Макс. знач.	Назначение
Входы	<i>Файл</i>	Строковый	1	-	-	-	Имя файла
	<i>Парам</i>	Строковый	2	-	-	-	Параметр запуска
	<i>Старт</i>	Событийный	3	-	-	-	Событие запуска процесса
	<i>Стоп</i>	Событийный	4	-	-	-	Событие остановки процесса
Выходы	<i>Активн</i>	Логический	-	-	-	-	Состояние процесса
	<i>Код взр</i>	Целый	-	-	-	-	Код возврата процесса

6.15.4.3 Описание алгоритма

Алгоритм осуществляет запуск процессов (приложений), имя которых задается на входе **Файл**. Файл может быть как выполняемый (с расширением .exe), так и произвольный (если файлу с данным расширением соответствует приложение его запускающее).

На входе **Парам** задается командная строка запуска, например, имя открываемого в приложении файла.

Процесс запускается если на вход **Старт** пришло событие.

Процесс останавливается и закрывается если на вход **Стоп** пришло событие.

Если процесс не был остановлен самим алгоритмом, то он выполняется до тех пор пока его не остановит пользователь или пока он не завершится сам.

Если процесс активен, то выход **Активн** принимает значение «Да», в противном случае – «Нет».

Если процесс запущен успешно, то выход **Код взр** принимает значение отличное от равного значению 0, в противном случае равно значению 0.

6.15.4.4 Мастер настройки

Имя файла или приложения задается с помощью Мастера настройки. Выбор осуществляется стандартным диалогом открытия файлов.

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

6.16 Группа 'Таблицы'

6.16.1 Писать в таблицу

6.16.1.1 Назначение

Алгоритм предназначен для добавления одной записи в таблицу и заполнения данными тех полей таблицы, имена которых представлены набором входов.

6.16.1.2 Параметры

Наименование	Обознач.	Тип	Номер	Нач. знач.	Назначение
Входы	<i>Писать</i>	Событийный	1		Событие записи
	<i>Сервер</i>	Строковый	2		Имя сервера
	<i>База данных</i>	Строковый	3		Имя базы данных
	<i>Таблица</i>	Строковый	4		Имя таблицы
	<i>?...?</i>	Настраиваемый	4+1		Имя поля таблицы
	<i>?...?</i>	Настраиваемый	...		Имя поля таблицы
	<i>?...?</i>	Настраиваемый	4+N		Имя поля таблицы
Выходы	<i>Статус</i>	Векторный	1		Результат выполнения операции записи данных в таблицу

6.16.1.3 Описание алгоритма

Алгоритм имеет 3 или 4 обязательных, N дополнительных входов и 1 выход.

Обязательными входами являются **Писать**, **Сервер**, **База данных** и **Таблица**.

Дополнительные входы **?...?** (наименования полей таблицы) имеют переменные обозначения и назначаются в Мастере настроек. Количество дополнительных входов меньше или равно количеству полей таблицы.

Алгоритм пытается добавить запись в таблицу, имя и путь к которой заданы на входах **Сервер**, **База данных** и **Таблица**. Запись добавляется при соблюдении следующих условий :

- имеется хотя-бы один дополнительный вход;
- вход **Писать** конфигурирован;
- на вход **Писать** поступило событие.

В противном случае алгоритм вообще не будет делать попыток добавить запись.

Значения полей **Сервер**, **База данных** и **Таблица** устанавливаются только в Мастере настроек, т.е. их нельзя задать редактированием значений на входах или конфигурированием.

Если запись в таблицу успешно добавлена и значение с каждого дополнительного входа помещено в соответствующее ему поле таблицы, то на выход **Статус** записывается число 0.

При выполнении алгоритма могут возникнуть следующие ситуации, которые в задаче можно отследить и обработать:

- изменилась структура данных таблицы. На выход **Статус** записывается число 1 (бит 01 равен 1);
- База данных или таблица недоступны. На выход **Статус** записывается число 2 (бит 02 равен 1);
- не задано ни одного поля. На выход **Статус** записывается число 4 (бит 03 равен 1).

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

6.16.1.4 Мастер настроек

Мастер настроек позволяет установить тип базы данных, значения на входах **Сервер**, **База данных** и **Таблица**, определить нужное количество дополнительных входов, т.е. полей таблицы, управлять предварительной очисткой таблицы перед началом выполнения задачи и задать автоматическое прерывание задачи в случае ошибки чтения таблицы.

Установка типа Базы данных производится выбором закладки **SQL** или **MS Access**.

Если выбрана закладка **SQL**, в алгоритме будет три входа: **Сервер**, **База данных** и **Таблица**, а если **MS Access** – два входа: **База данных** и **Таблица**.

Если выбрана закладка **MS Access**, то установка значения на входе **База данных** осуществляется нажатием кнопки **Выбор базы данных**, а значения на входе **Таблица** - выбором из списка таблиц.

Если выбрана закладка **SQL**, то установка значения на входе **Сервер** осуществляется нажатием кнопки **Выбор сервера**, значения на входе **База данных** – выбором из списка Баз данных, а значения на входе **Таблица** - выбором из списка таблиц.

Дополнительные входы назначаются в группе **Поля таблицы - дополнительные входы**. Следует иметь в виду, что в списке полей таблицы будут представлены только те поля, типы данных которых поддерживаются для входов алгоритмов, а именно:

- целое число (I);
- действительное число (F);
- строка (S);
- логические данные (B);
- дата и время (W).

Управление прерыванием выполнения задачи в случае ошибки осуществляется установкой флага **Если ошибка - вызывать алгоритм Авария**. Если этот флаг не установлен, пользователю рекомендуется самому решать, что делать в случае той или иной ошибки.

Управление предварительной очисткой таблицы осуществляется установкой флага **Чистить таблицу перед выполнением задачи**.

Если какие-либо дополнительные входы были законфигурированы с другими алгоритмами, то при смене базы данных и (или) таблицы или изменении списка отмеченных полей все соединения будут удалены.

6.16.2 Читать таблицу

6.16.2.1 Назначение

Алгоритм предназначен для чтения одной записи таблицы и выбора данных из тех полей таблицы, имена которых представлены набором выходов.

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

6.16.2.2 Параметры

Наименование	Обознач.	Тип	Номер	Нач. знач.	Назначение
Входы	<i>Сервер</i>	Строковый	1		Имя сервера
	<i>База данных</i>	Строковый	2		Имя базы данных
	<i>Таблица</i>	Строковый	3		Имя таблицы
Выходы	<i>Статус</i>	Векторный	1		Результат выполнения операции чтения записи
	<i>КолЗап</i>	Целый	2		Количество записей таблицы
	<i>?...?</i>	Настраиваемый	2+1		Имя поля таблицы
	<i>?...?</i>	Настраиваемый	...		Имя поля таблицы
	<i>?...?</i>	Настраиваемый	2+N		Имя поля таблицы

6.16.2.3 Описание алгоритма

Алгоритм имеет 2 или 3 входа, 2 обязательных и N дополнительных выходов. Обязательными выходами являются **Статус** и **КолЗап**. Дополнительные выходы *?...?* (наименования полей таблицы) имеют переменные обозначения и назначаются в Мастере настроек. Количество дополнительных выходов меньше или равно количеству полей таблицы.

Алгоритм пытается прочитать запись из таблицы, имя и путь к которой заданы на входах **Сервер**, **База данных** и **Таблица**.

Значения полей **Сервер**, **База данных** и **Таблица** устанавливаются только в Мастере настроек, т.е. их нельзя задать редактированием значений на входах или конфигурированием.

Если запись таблицы успешно прочитана, то на выход **Статус** записывается число 0, а значение на выходе **КолЗап** увеличивается на единицу. В этом случае при следующем выполнении алгоритма будет сделана попытка получить следующую запись и т.д. до тех пор, пока не будут прочитаны все записи.

Если для алгоритма заданы дополнительные выходы, т.е. поля таблицы, то на каждый выход заносится значение соответствующего ему поля.

Если при очередном выполнении алгоритма все записи окажутся прочитанными, то значение на выходе **КолЗап** не меняется, а на выход **Статус** записывается число 8. После этого алгоритм таблицу читать не будет, даже если задача продолжает выполняться.

При выполнении алгоритма могут возникнуть следующие ситуации, которые в задаче можно отследить и обработать:

- изменилась структура данных таблицы. На выход **Статус** записывается число 1 (бит 01 равен 1);
- База данных или таблица недоступны. На выход **Статус** записывается число 2 (бит 02 равен 1);
- не задано ни одного поля. На выход **Статус** записывается число 4 (бит 03 равен 1).
- достигнут конец таблицы. На выход **Статус** записывается число 8 (бит 04 = 1)

6.16.2.4 Мастер настроек

Мастер настроек позволяет установить тип базы данных, значения на входах **Сервер**, **База данных** и **Таблица**, определить нужное количество дополнительных входов, т.е. полей таблицы, управлять предварительной очисткой таблицы перед началом выполнения задачи и задать автоматическое прерывание задачи в случае ошибки чтения таблицы.

Установка типа Базы данных производится выбором закладки **SQL** или **MS Access**.

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

Если выбрана закладка **SQL**, в алгоритме будет три входа: **Сервер**, **База данных** и **Таблица**, а если **MS Access** – два входа: **База данных** и **Таблица**.

Если выбрана закладка **MS Access**, то установка значения на входе **База данных** осуществляется нажатием кнопки **Выбор базы данных**, а значения на входе **Таблица** - выбором из списка таблиц.

Если выбрана закладка **SQL**, то установка значения на входе **Сервер** осуществляется нажатием кнопки **Выбор сервера**, значения на входе **База данных** – выбором из списка Баз данных, а значения на входе **Таблица** - выбором из списка таблиц.

Дополнительные выходы назначаются в группе **Поля таблицы - дополнительные выходы**. Следует иметь в виду, что в списке полей таблицы будут представлены только те поля, типы данных которых поддерживаются для входов алгоритмов, а именно:

- целое число (I);
- действительное число (F);
- строка (S);
- логические данные (B);
- дата и время (W).

Управление прерыванием выполнения задачи в случае ошибки осуществляется установкой флага **Если ошибка - вызывать алгоритм Авария**. Если этот флаг не установлен, пользователю рекомендуется самому решать, что делать в случае той или иной ошибки.

Управление предварительной очисткой таблицы осуществляется установкой флага **Чистить таблицу перед выполнением задачи**.

Если какие-либо дополнительные входы были законфигурированы с другими алгоритмами, то при смене базы данных и (или) таблицы или изменении списка отмеченных полей все соединения будут удалены.

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

6.17 Группа 'Мнемосимволы'

6.17.1 Кнопка

6.17.1.1 Назначение

Реакция на событие нажатия и подсчёт кол-ва нажатий.

6.17.1.2 Параметры

Наименование	Обозначение	Тип	Назначение
Входы	<i>Заголовок</i>	Строка	Заголовок кнопки
Выходы	<i>Нажатие</i>	Событийный	Генерирование события при нажатии
	<i>Положение</i>	Логический	Нажата кнопка или нет

6.17.1.3 Описание алгоритма

При поступлении строчного значения на вход **Заголовок** на кнопке появляется соответствующий текст для надписи кнопки. При нажатии на выходе **Нажатие** генерируется событие нажатия. Если установлен флаг **Залипание** в Мастере настроек, то на выходе **Положение** будет установлено соответствующее значение при зажатии/отпуске кнопки. Если залипание не установлено – всегда состояние «Нет».

6.17.1.4 Мастер настройки

Позволяет выбрать текст, его положение относительно заголовка и стиль кнопки - залипающий или нет.

6.17.2 Ползунок

6.17.2.1 Назначение

Ручное регулирование значения с плавающей точкой.

6.17.2.2 Параметры

Наименование	Обозначение	Тип	Назначение
Входы	<i>A</i>	Действительный	Принудительное задание значения
	<i>Минимум</i>	Действительный	Минимально возможное значение
	<i>Максимум</i>	Действительный	Максимально возможное значение
	<i>Блокировка</i>	Логический	Запрет ручного ввода
Выходы	<i>B</i>	Действительный	Выходное значение
	<i>Изм.</i>	Событийный	Событие изменения значения

6.17.2.3 Описание алгоритма

При сдвиге ползунка мышью или клавиатурой на выход подаётся значение. При поступлении сигнала на вход **Значение** положение ползунка принудительно устанавливается в соответствии с поступившим значением. При поступлении на вход **Минимум** сигнала, по величине большего текущего значения, текущее значение приводится к поступившему. Аналогично для входа **Максимум**.

6.17.2.4 Мастер настройки

Позволяет выбрать стиль и положение рисок, размер ползунка, цену деления для рисок и шаг дискретизации интервала (минимальное абсолютное значение приращения при минимальном сдвиге ползунка).

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

6.17.3 Редактор

6.17.3.1 Назначение

Ввод/вывод строчных данных.

6.17.3.2 Параметры

Наименование	Обозначение	Тип	Назначение
Входы	<i>A</i>	Строковый	Принудительная установка содержимого редактора
	<i>Только чтение</i>	Логический	Блокировка ввода с клавиатуры
Выходы	<i>B</i>	Строковый	Содержимое редактора
	<i>Изм.</i>	Событийный	Событие изменения содержимого редактора

6.17.3.3 Описание алгоритма

Алгоритм предназначен для отображения и / или ввода текстовых данных.

6.17.3.4 Мастер настройки

Позволяет выбрать шрифт текста редактора и цвет фона.

6.17.4 Редактор значений

6.17.4.1 Назначение

Ввод и отображение значения с плавающей точкой в текстовом виде.

6.17.4.2 Параметры

Наименование	Обозначение	Тип	Назначение
Входы	<i>A</i>	Действительный	Принудительная установка значения
	<i>Мин.</i>	Действительный	Минимально возможное значение
	<i>Макс.</i>	Действительный	Максимально возможное значение
Выходы	<i>B</i>	Действительный	Результат выбора
	<i>Изм.</i>	Событийный	Событие изменения значения

6.17.4.3 Описание алгоритма

Работа алгоритма аналогична алгоритму **Ползунок** с той разницей, что значение представлено в текстовом виде и не дискретно.

6.17.4.4 Мастер настройки

Отсутствует.

6.17.5 Слайд

6.17.5.1 Назначение

Позволяет отображать последовательно некоторое множество изображений.

6.17.5.2 Параметры

Наименование	Обозначение	Тип	Назначение
Входы	<i>Индекс изображения</i>	Целый	Индекс изображения

6.17.5.3 Описание алгоритма

Алгоритм показывает одно из выбранных изображений в зависимости от значения на входе **Индекс изображения**. Нумерация индексов начинается с 0. Если значение превышает N-1 (где N – кол-во изображений, содержащихся в алгоблоке), то отображается изображение с наивысшим индексом (т.е. последнее в списке).

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

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

6.17.5.4 Мастер настройки

Позволяет выбрать множество изображений, содержащихся в алгоблоке, экспортировать изображения из списка в файл и задать опции отображения:

- **Растягивать.** Это масштабирование изображения под размер алгоблока;
- **Центрировать.** Это выравнивание изображения по центру алгоблока;
- **Сохранять пропорции.** Опция действительна при включенном масштабировании.

6.17.6 Индикатор

6.17.6.1 Назначение

Отображает текст на заданном цветовом фоне.

6.17.6.2 Параметры

Наименование	Обозначение	Тип	Назначение
Входы	<i>Текст</i>	Строковый	Отображаемая строка
	<i>R</i>	Целый	Красная составляющая цвета фона
	<i>G</i>	Целый	Зеленая составляющая цвета фона
	<i>B</i>	Целый	Синяя составляющая цвета фона

6.17.6.3 Описание алгоритма

Алгоблок отображает текст, взятый со входа **Текст** заданным шрифтом на цветовом фоне, покомпонентно заданном значениями на входах **R**, **G** и **B**.

6.17.6.4 Мастер настройки

Позволяет задать шрифт и выравнивание текста относительно границ алгоблока.

6.17.7 Управление задачей

6.17.7.1 Назначение

Запуск и останов выполнения задачи вручную или по расписанию.

6.17.7.2 Параметры

Наименование	Обозначение	Тип	Назначение
Входы	<i>Статус</i>	Вектор	Отображение статуса
Выходы	<i>Пуск</i>	Событийный	Пуск прикрепленной задачи
	<i>Стоп</i>	Событийный	Останов прикрепленной задачи

6.17.7.3 Описание алгоритма

Алгоблок должен быть связан с одной задачей в окне редактирования проекта. С выхода **Статус** задачи должна быть проведена связь на соответствующий вход **Статус** алгоблока. С выходов **Пуск** и **Стоп** должны быть проведены связи к соответствующим входам задачи. Алогоблок позволяет вручную запустить или остановить связанную задачу, или задать расписание по которому будут выполнены данные действия.

6.17.7.4 Мастер настройки

Позволяет задать текст отображаемый на алгоблоке в поле **Имя задачи**. При нажатии на кнопку **Задать расписание** будет выведено диалоговое окно, в котором можно задать параметры периодического запуска/останова задачи. Если установить флаг **Точка старта**, то запуск/останов будет производиться, начиная с указанной точки времени. Если нет, то с момента запуска задачи. В поле **Период запуска** задается период выдачи сигнала на выход **Пуск**, в поле **Тайм аут** – период времени от момента выдачи очередного сигнала **Пуск**, по окончании которого будет подан сигнал на выход **Стоп**.

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

6.17.8 График

6.17.8.1 Назначение

Отображение кривых на основе значений на входах в данный момент времени.

6.17.8.2 Параметры

Наименование	Обозначение	Тип	Назначение
Входы	A1	Действительный	Значение откладываемое по левой шкале
	A2	Действительный	Значение откладываемое по левой шкале
	...	Действительный	Значение откладываемое по левой шкале
	A30	Действительный	Значение откладываемое по левой шкале

6.17.8.3 Описание алгоритма

Алгоблок строит кривые по значениям со входов (левая шкала) в данный момент времени (нижняя шкала).

Кнопки управления на графике позволяют изменять масштаб (отдельно, по нижней и левой осям), прокручивать построенные кривые (вперед/назад, вверх/вниз), управлять видимостью кривых (в легенде и мастере настроек).

При необходимости можно очистить график, нажав на соответствующей кнопке (с изображением чистого листа).

6.17.8.4 Мастер настройки

С помощью Мастера можно задавать следующие настройки:

- добавлять/удалять количество входов графика:
 - а) минимально – 1;
 - б) максимально - 30;
- переименовывать входы графика;
- управлять видимостью кривых с тем условием, что хотя бы одна кривая должна оставаться видимой;
- изменять цвет кривых;
- показывать кривые в ступенчатом виде;
- выделять точки на кривой, соответствующих реальным отсчетам; изменять их стиль и размер;
- задавать видимый период графика от 5 секунд до одного года;
- задавать видимую часть графика, то есть указывать положение крайней правой (последней) точки кривой в режиме автопрокрутки. Эта опция имеет смысл, когда график будет работать в режиме автоматической прокрутки и когда используется кнопка **Подогнуть**;
- задавать период обновления в миллисекундах (минимальное значение: 50 мс);
- показывать график в объемном виде и указывать атрибуты объемного представления: перспективу и ортогональность;
- изменять цвет фона графика, задавать градиент;
- показ легенды;
- задание заголовка над графиком, атрибуты заголовка: выравнивание заголовка, цвет шрифта, цвет фона.

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

6.18 Группа 'Случайные'

6.18.1 СлучЧисло

6.18.1.1 Назначение

Генерация случайного числа в заданном диапазоне.

6.18.1.2 Параметры

Наименование	Обознач.	Тип	Номер	Нач. знач.	Мин. знач.	Макс. знач.	Назначение
Входы	<i>Разрешить</i>	Логический	1	Да	-	-	Разрешить генерацию
	<i>Минимум</i>	Действительный	2	0			Минимальное значение
	<i>Максимум</i>	Действительный	3	1			Максимальное значение
Выходы	<i>Результат</i>	Действительный	-	-	-	-	Случайное число

6.18.1.3 Описание алгоритма

Если на вход **Разрешить** подано значение «Да», то 'алгоритм генерирует случайное действительное число на выходе **Результат** в диапазоне, указанном на входах **Минимум** и **Максимум**.

6.18.2 СлучЦелое

6.18.2.1 Назначение

Генерация случайного целого числа в заданном диапазоне.

6.18.2.2 Параметры

Наименование	Обознач.	Тип	Номер	Нач. знач.	Мин. знач.	Макс. знач.	Назначение
Входы	<i>Разрешить</i>	Логический	1	Да	-	-	Разрешить генерацию
	<i>Минимум</i>	Целый	2	0			Минимальное значение
	<i>Максимум</i>	Целый	3	9			Максимальное значение
Выходы	<i>Результат</i>	Целый	-	-	-	-	Случайное целое

6.18.2.3 Описание алгоритма

Если на вход **Разрешить** подано значение «Да», то 'алгоритм генерирует случайное целое число на выходе **Результат** в диапазоне, указанном на входах **Минимум** и **Максимум**.

6.18.3 СлучЛог

6.18.3.1 Назначение

Генерация случайного логического значения.

6.18.3.2 Параметры

Наименование	Обознач.	Тип	Номер	Нач. знач.	Мин. знач.	Макс. знач.	Назначение
Входы	<i>Разрешить</i>	Логический	1	Да	-	-	Разрешить генерацию
Выходы	<i>Результат</i>	Логический	-	-	-	-	Случайное логическое значение

6.18.3.3 Описание алгоритма

Если на вход **Разрешить** подано значение «Да», то 'алгоритм генерирует случайное логическое значение («Да» или «Нет») на выходе **Результат** в диапазоне, указанном на входах **Минимум** и **Максимум**.

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

