

**ГОСУДАРСТВЕННАЯ КОРПОРАЦИЯ ПО АТОМНОЙ ЭНЕРГИИ «РОСАТОМ»  
АКЦИОНЕРНОЕ ОБЩЕСТВО «ИНЖЕНЕРНО-ТЕХНИЧЕСКИЙ ЦЕНТР «ДЖЭТ»  
(АО ИТЦ «ДЖЭТ»)**



# **ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ**

## **Кодогенератор AUTO**

### **Руководство пользователя**

**б/н**

**Номер редакции 1.0**

**На 38 листах**

Собственность . Запрещается без предварительного письменного разрешения собственника воспроизводить, переводить, изменять в любой форме или частично, передавать во временное или постоянное пользование другим организациям или лицам, разглашать или использовать сведения в коммерческих интересах лиц или организаций, не связанных договорными

**Москва, 2023**

## ПРАВА НА СОДЕРЖАНИЕ

Настоящий документ является собственностью АО ИТЦ «ДЖЭТ» и защищен законодательством Российской Федерации и международными соглашениями об авторских правах и интеллектуальной собственности

Копирование документа либо его фрагментов в любой форме, распространение, в том числе в переводе, воспроизводство, изменение в любой форме или частично, а также передача во временное или постоянное пользование третьим лицам, разглашение или использование сведений в коммерческих интересах третьих лиц возможны только с письменного разрешения АО ИТЦ «ДЖЭТ».

Документ и связанные с ним графические изображения могут быть использованы только в информационных, некоммерческих или личных целях.

АО ИТЦ «ДЖЭТ» оставляет за собой право на изменение или обновление настоящего документа без предварительного уведомления.

Все названия компаний и продуктов, которые являются товарными знаками или зарегистрированными товарными знаками, являются собственностью соответствующих владельцев.

За содержание, качество, актуальность и достоверность используемых в документе материалов, права на которые принадлежат другим правообладателям, а также за возможный ущерб, связанный с использованием этих материалов, ответственности не несет.

### **АО «ИТЦ «ДЖЭТ»**

117335, г. Москва, Нахимовский проспект, дом 58

Сайт компании: <https://get-sim.ru/>

Тел.: +7 495 788 04 06

Электронный адрес службы поддержки: [itcget@rosatom.ru](mailto:itcget@rosatom.ru)

**СОДЕРЖАНИЕ**

ПЕРЕЧЕНЬ ПРИНЯТЫХ СОКРАЩЕНИЙ .....	5
1 ОБЩИЕ СВЕДЕНИЯ И НАЗНАЧЕНИЕ .....	6
1.1 Назначение программы .....	6
1.2 Основные характеристики программы .....	6
1.3 Ограничения, накладываемые на область применения программы .....	6
2 Создание схемы автоматики .....	7
2.1 Общие понятия и алгоритмы .....	7
2.1.1 Создание проекта системы автоматики .....	7
2.1.2 Настройка расчетного слоя (решателя) .....	7
2.2 Генерация расчетной схемы .....	7
2.3 Основы моделирования систем автоматики и логических устройств .....	8
2.4 Типы объектов расчетной схемы .....	8
2.4.1 Субмодель AUTO .....	8
2.4.2 Порт входа AUTO .....	9
2.4.3 Порт выхода AUTO .....	9
2.4.4 Уровень неподсоединенных портов .....	10
2.4.5 Блок записи свойств .....	10
2.4.6 Условие выполнения субмодели .....	10
2.4.7 Шаблон AUTO .....	11
2.4.8 Константа .....	11
2.4.9 Меандр .....	12
2.4.10 Входной контакт s3 .....	13
2.4.11 Датчик s3 .....	14
2.4.12 Выходной контакт s3 .....	15
2.4.13 Шаг интегрирования .....	16
2.4.14 Равномерный шум .....	17
2.4.15 Входной контакт s3 матричный .....	17
2.4.16 Сумматор .....	19
2.4.17 Сравнивающее устройство .....	19
2.4.18 Перемножитель .....	21
2.4.19 Перемножение элементов вектора .....	22
2.4.20 Делитель .....	23
2.4.21 Деление скаляра на вектор .....	23
2.4.22 Усилитель .....	24
2.4.23 Векторный усилитель .....	24
2.4.24 Размножитель .....	25
2.4.25 Знак .....	26
2.4.26 Целая часть .....	26
2.4.27 Дробная часть .....	27
2.4.28 Линейный преобразователь .....	28
2.4.29 Динамическая выборка .....	28
2.4.30 Выборка по активному элементу .....	29
3 МОДЕЛИРОВАНИЕ СХЕМ АВТОМАТИКИ .....	30

3.1.1 Создание проекта системы автоматика.....	30
3.1.2 Настройка расчетного слоя (решателя) .....	30
3.1.3 Блоки для моделирования систем автоматика.....	31
3.1.4 Режимы отображения рабочего поля .....	31
3.1.5 Основы моделирования систем автоматика .....	31
3.2 Особенности генерации компонентов автоматика .....	32
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	37
Лист регистрации изменений .....	38

**ПЕРЕЧЕНЬ ПРИНЯТЫХ СОКРАЩЕНИЙ**

<b>ГОСТ Р</b>	Государственный стандарт Российской Федерации
<b>ПО</b>	Программное обеспечение
<b>ПК</b>	Персональный компьютер
<b>РП</b>	Руководство пользователя
<b>РЭ</b>	Руководство по эксплуатации
<b>ОС</b>	Операционная система
<b>САПФИР</b>	Система Автоматического Проектирования Физических Инженерных Расчетов – интегрированная программная система для создания и проведения различных физических расчетов и их отладки.
<b>СУБД</b>	Система управления базами данных
<b>ESUSDS</b>	англ. executive system of Universal Software Development System (досл. исполнительная система Универсальной Системы Разработки Программного Обеспечения) – интегрированная программная система, которая поддерживает документирование, разработку, выполнение в режиме реального времени и тестирование всего комплекса программного обеспечения тренажера.

# 1 ОБЩИЕ СВЕДЕНИЯ И НАЗНАЧЕНИЕ

Этот документ предназначен для прикладных программистов, разрабатывающих модели теплогидравлических сетей в рамках US3 технологии. Он содержит наиболее полную информацию по работе с графическим интерфейсом кодогенератора, правила построения схем, имена используемых массивов и переменных, а также физические соотношения и законы, использованные для построения математических моделей в режиме реального времени.

## 1.1 Назначение программы

Кодогенератор АУТО является основной составляющей частью математической модели тренажера атомного энергоблока в части моделирования схем автоматики и АСУ ТП.

Инженер может графически создавать логические связи (алгоритмы) защит, блокировок, управления и контроля технологическими процессами, используя библиотеку стандартных блоков. Непосредственно на экране компьютера, вводить значения уставок срабатывания защит, временных задержек и импульсов, а так же корректировать настройки регуляторов. Основываясь на топологии введенной пользователем схемы и значениях уставок, временных задержек и импульсов, код генератора автоматически создает исходный код модели на языке С, файлов описания переменных и файлов инициализации начальных условий для последующей компоновки в исполняемый файл под управлением исполнительной системы ESUSDS. В кодогенератор также входят функции проверки корректности и достоверности исходных данных, отсутствие алгебраических петель, деление на ноль или отсутствие неподключенных выходных контактов.

Кодогенератор спроектирован для уменьшения усилий разработчиков, увеличения производительности и улучшения качества работ по созданию симуляционных задач в режиме реального времени. Он использует мощный графический редактор САПФИР, который позволяет инженерам моделировать сложные электрические схемы более легко.

## 1.2 Основные характеристики программы

Программа является исполняемым файлом для ОС Windows и Linux.

## 1.3 Ограничения, накладываемые на область применения программы

Разработанные модели функционируют только в рамках исполнительной системы ESUSDS.

## 2 Создание схемы автоматике

### 2.1 Общие понятия и алгоритмы

#### 2.1.1 Создание проекта системы автоматике

Для создания схемы автоматике в среде САПФИР необходимо:

- Выбрать тип проекта: «Файл» → «Создать» → «АУТО»;
- Включить видимость панели библиотеки классов: «Вид» → «Показать панель библиотеки классов»;
- Загрузить библиотеку классов для моделирования схем автоматике: «Библиотека» → «Загрузить библиотеку классов»;
- Сохранить текущий проект.

#### 2.1.2 Настройка расчетного слоя (решателя)

Чтобы вызвать окно свойств решателя, требуется на панели инструментов «Слой» левой кнопкой мыши нажать на кнопку  (редактировать свойства текущего решателя) (Рисунок 1 – ):

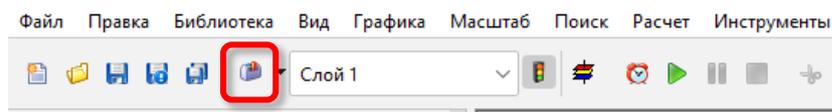


Рисунок 1 – Панель инструментов «Слой», кнопка редактирования свойств текущего слоя

Редактировать параметры расчета можно только для выбранного слоя (Рисунок 2 –):



Рисунок 2 – Выбор слоя (решателя) для дальнейшей работы

Перечни всех параметров расчета задаются шаблоном расчетного условия. Параметры слоя (решателя) могут быть сохранены в файл и затем использованы в другом проекте. Можно скопировать параметры слоя (решателя) проекта и вставить в другой проект.

### 2.2 Генерация расчетной схемы

Генерация расчетной схемы происходит с помощью генератора кода АУТО. Генератор кода АУТО предназначен для автоматического создания в графической оболочке расчётной схемы любой необходимой конфигурации и соответствующего набора файлов (расчётных модулей, составляющих код АУТО), содержащих математическую модель проектируемой системы автоматике.

Расчетная схема создается по определенным правилам в графическом редакторе из библиотечного набора элементов, характерных для разрабатываемого проекта. Для каждого из элементов расчетной схемы выводится список параметров, которые должны быть введены при подготовке расчетной схемы.

На этапе генерации генератор кода проводит автоматическую проверку правильности подключения элементов на схеме и заданных параметров.

Генератор кода AUTO позволяет быстро и качественно создавать расчетные схемы, объединять их между собой и проводить отладку, как отдельных технологических систем, так и целых комплексов.

Схемы автоматики набираются в графической оболочке из элементарных блоков. Для блоков задаются материально-геометрические и расчетные параметры. Указывается связь с соответствующими полями в БД. Определяются точки контроля и отказы. С помощью соответствующих блоков задаются граничные условия.

## 2.3 Основы моделирования систем автоматики и логических устройств

Соединение объектов систем автоматики и логических устройств происходит с помощью линий от источника до получателя или от получателя к источнику (2.3Рисунок 3 –:

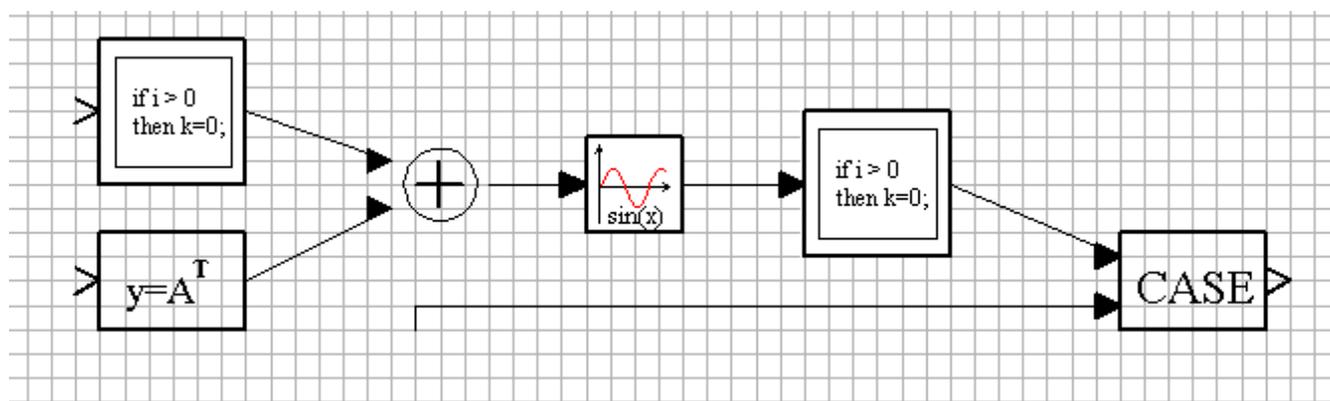


Рисунок 3 – Соединение объектов систем автоматики и логических устройств

Блоки систем автоматики и логических устройств устанавливается на схему и соединяются между собой.

## 2.4 Типы объектов расчетной схемы

Реальные объекты моделируются в расчетной схеме набором стандартных элементов и связей между ними.

### 2.4.1 Субмодель AUTO

Блок формирует «оболочку» для создания вложенной (-ых) схемы (схем), набранной как из типовых блоков, так и из Субмоделей более глубокого уровня вложенности (Рисунок 4 –.



Рисунок 4 – Субмодель AUTO

**Входы:** Количество входов данного блока определяется автоматически при наполнении внутренней структуры блока типовыми блоками «Порт входа».

**Выходы:** Количество выходов данного блока определяется автоматически при наполнении внутренней структуры блока типовыми блоками «Порт выхода».

Количество свойств блока редактируется с помощью с помощью вставки свойств и может быть различным.

### 2.4.2 Порт входа AUTO

Блок используется внутри Субмодели и определяет наличие у неё входного порта. Служит для соединения внешней части схемы со схемой, находящейся внутри Субмодели (Рисунок 5 –).



Рисунок 5 – Порт входа AUTO

Блок организует прием данных от типовых блоков и Субмоделей, внешних по отношению к данной, а также передачу данных для преобразования типовыми блоками (или Субмоделями) внутренней структуры данной субмодели. Количество блоков «Порт входа» автоматически определяет количество входов в данную Субмодель. Порт входа может обрабатывать как скалярные, так и векторные входные сигналы.

### 2.4.3 Порт выхода AUTO

Блок используется внутри Субмодели и определяет наличие у нее выходного порта. Служит для соединения внешней части схемы со схемой, находящейся внутри субмодели (Рисунок 6 –).

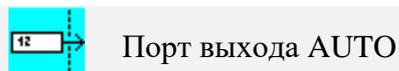


Рисунок 6 – Порт выхода AUTO

Блок организует прием данных от типовых блоков и Субмоделей внутренней структуры данной Субмодели, а также передачу данных для преобразования типовыми блоками (или Субмоделями), внешними по отношению к данной Субмодели. Количество блоков «Порт выхода» автоматически определяет количество выходов из данной Субмодели. Порт выхода может обрабатывать как скалярные, так и векторные сигналы.

#### 2.4.3.2 В память AUTO

Блок формирует «оболочку» для создания вложенной (-ых) схемы, набранной как из типовых блоков, так и из Субмоделей более глубокого уровня вложенности (2.4.3.2Рисунок 7 – ).

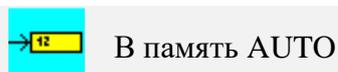


Рисунок 7 – В память AUTO

#### 2.4.3.3 Из памяти AUTO

Блок предназначен для соединения одной части схемы с другой, находящейся в произвольном месте проекта (2.4.3.3Рисунок 8 –).



Рисунок 8 – Из памяти AUTO

Для выбора источника, заданного при помощи блока «В память», необходимо произвести двойной щелчок по блоку и выбрать в списке нужную точку соединения.

#### 2.4.4 Уровень неподсоединенных портов

Данный блок позволяет подключить все неподключенные порты всех блоков данного проекта к единому источнику (Рисунок 9 –).



Уровень неподсоединенных портов

Рисунок 9 – Уровень неподсоединенных портов

Для этого необходимо поместить блок на схему и соединить его с выходом источника. Наиболее часто блок используется в больших схемах для задания нулевого (равного 0) сигнала как входящего по умолчанию на все неиспользуемые порты всех блоков.

#### 2.4.5 Блок записи свойств

Блок позволяет задать свойство другого блока (точнее, значение свойства другого блока) из заданного источника (Рисунок 10 –).



Блок записи свойств

Рисунок 10 – Блок записи свойств

Для этого надо вставить данный блок в другой (разместить на другом блоке) и указать в нём имя присваиваемого свойства блока.

Таблица 1 – Свойства блока «Блок записи свойств»

Название	Имя свойства	Значение	Имя переменной
Имя свойства	propname	Имя свойства блока, в который вставлен данный блок, значение которого должно быть присвоено из источника.	

#### 2.4.6 Условие выполнения субмодели

Блок представляет собой механизм для заморозки и разморозки расчета субмодели в зависимости от внешнего условия (Рисунок 11 –).



Условие выполнения субмодели

Рисунок 11 – Условие выполнения субмодели

Если внутри Субмодели разместить блок «Условие выполнения субмодели», то у субмодели появится дополнительный вход, а сама субмодель будет выполняться (расчитываться) только в том случае, если на этот вход подается логическая 1. Если на новый вход субмодели подается логический 0, то вся субмодель и все блоки, находящиеся внутри неё, «замораживаются» до тех пор, пока на вход субмодели не станет вновь поступать логическая 1.

Таблица 2 – Свойства блока «Условие выполнения субмодели»

Название	Имя свойства	Значение	Имя переменной
Игнорировать вышестоящее условие	ignoreup		

### 2.4.7 Шаблон АУТО

Блок реализует шаблон описания алгоритма логики управления АУТО (Рисунок 12 –).

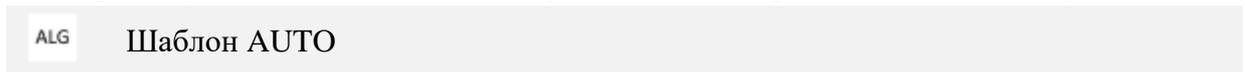


Рисунок 12 – Шаблон АУТО

Блок представляет собой субмодель (сборку) (Рисунок 13 –):

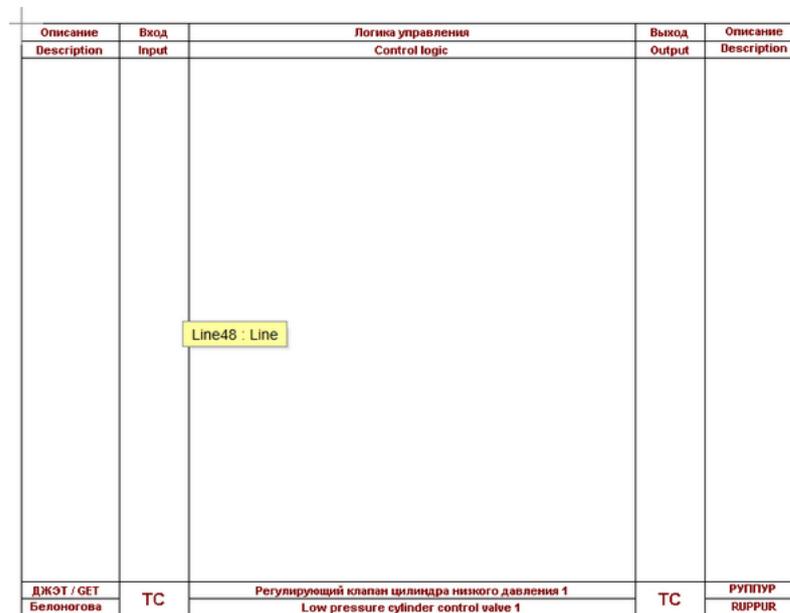


Рисунок 13 – Структура субмодели блока «Шаблон АУТО»

Таблица 3 – Свойства блока «Шаблон АУТО»

Название	Имя свойства	Значение	Имя переменной
Название субмодели	creator	Название субмодели	
Тренажер (ru)	trainer_ru	Принадлежность тренажера (на русском языке)	
Тренажер (en)	trainer_en	Принадлежность тренажера (на английском языке)	
Разработчик	creator	Разработчик (ФИО)	
Описание алгоритма (ru)	logic_ru	Описание алгоритма (на русском языке)	
Описание алгоритма (en)	logic_en	описание алгоритма (на английском языке)	
Имя системы	system	Имя системы	

### 2.4.8 Константа

Блок (Рисунок 14 –) возвращает на выходе заданную пользователем постоянную величину:

$$Y = a$$

Для работы блока необходимо задать величину константы  $a$  (по умолчанию  $a=1$ ).

 Константа

Рисунок 14 – Константа

Таблица 4 – Свойства блока «Константа»

Название	Имя свойства	Значение	Имя переменной
Значение	a	Значение величины a на выходе блока	
Тип данных (для генерации кода)	src_type	Тип данных выходной переменной, создаваемой при генерации кода для внешней целевой системы. Постоянная величина (константа) при генерации кода может быть действительным числом (тип double) или целым числом (тип integer) или логическим сигналом (тип boolean)	
Название	txt	Название формируемой постоянной величины, для справки	

Свойство  $a$  может быть вектором:  $a = [a_1, a_2, \dots, a_n]$ , в этом случае на выходе будет также вектор:  $Y=a$ .

По умолчанию блок формирует скалярный выходной сигнал.

Именованные свойства могут быть заданы как локальные переменные модели (или субмодели) в скрипте на языке программирования.

### 2.4.9 Меандр

Реализует источник прямоугольных импульсов (Рисунок 15 –) с заданными временными и амплитудными параметрами:

$$y(t) = \begin{cases} y_1, & \text{при } \left(\frac{t}{T_1 + T_2}\right) \leq \text{frac}\left(\frac{T_1}{T_1 + T_2}\right) \\ y_2, & \text{при } \left(\frac{t}{T_1 + T_2}\right) > \text{frac}\left(\frac{T_1}{T_1 + T_2}\right) \end{cases}$$

где  $y_1, y_2, T_1, T_2$  – свойства блока,  $y(t)$  – выходной сигнал,  $t$  – текущее модельное время в секундах,  $\text{frac}(t/(T_1+T_2))$  – дробная часть.



Меандр

Рисунок 15 – Меандр

Таблица 5 – Свойства блока «Меандр»

Название	Имя свойства	Значение	Имя переменной
Длительность 1-го полупериода	t1	Время T1, в течение которого блок на выходе формирует значение первого полупериода	
Значение 1-го полупериода	y1	Сигнал y1, подаваемый блоком на выход в течение первого полупериода	
Длительность 2-го полупериода	t2	Время T2, в течение которого блок на выходе формирует значение второго полупериода	
Значение 2-го полупериода	y2	Сигнал y2, подаваемый блоком на выход в течение второго полупериода.	

Свойства могут быть векторами:  $T_1 = [T_{11}, T_{12}, \dots, T_{1n}]$ ,  $y_1 = [y_{11}, y_{12}, \dots, y_{1n}]$  и  $y_2 = [y_{21}, y_{22}, \dots, y_{2n}]$ , размерности которых должны быть одинаковы, в этом случае на выходе будет также формироваться векторный сигнал:

$$y_i(t) = \begin{cases} y_{1i}, & \text{при } \left(\frac{t}{T_{1i} + T_{2i}}\right) \leq \text{frac}\left(\frac{T_1}{T_{1i} + T_{2i}}\right) \\ y_{2i}, & \text{при } \left(\frac{t}{T_{1i} + T_{2i}}\right) > \text{frac}\left(\frac{T_1}{T_{1i} + T_{2i}}\right) \end{cases}$$

По умолчанию блок формирует скалярный выходной сигнал.

#### 2.4.10 Входной контакт s3

Реализует считывание переменной из базы данных и передачу её на выход блока (Рисунок 16 –).



Входной контакт s3

Рисунок 16 – Входной контакт s3

Таблица 6 – Свойства блока «Входной контакт s3»

Название	Имя свойства	Значение	Имя переменной
Значение по	a	Значение на выходе блока, может быть	

АО ИТЦ «ДЖЭТ»	Программное обеспечение АУТО Руководство пользователя	Номер редакции 1.0
---------------	--	--------------------

умолчанию		векторным	
Имя контакта	cont_name	Имя переменной s3, считываемой блоком	
Тип контакта	cont_type	Тип данных выходной переменной при генерации кода для полномасштабной модели (Float / Boolean / Integer)	
Формат имени переменной	decl_format	<префикс><компонент><контакт/запрос> / <префикс><контакт/запрос> / <префикс><запрос><контакт> / <контакт/запрос>	
Добавить в базу сигналов	dbm_add	Да / Нет	
Формат имени переменной	query		
Автоматически сбрасывать	auto_reset	Да / Нет	
Операция для нескольких контактов	vec_mode	Операция ИЛИ / Запаковка в вектор	
Функциональный план	plan		
Отлаживаемый компонент	debug_comp		
Дополнительный шаблон для файла DBM	add_decl		
Порт для сортировки	sort_port	Да / Нет	
Переход на другой файл	cross_name		
Уникальный идентификатор	old_video	Уникальный идентификатор в базе сигналов системы s3	

### 2.4.11 Датчик s3

Блок предназначен для вывода параметров в режиме отладки (Рисунок 17 –).



Датчик s3

Рисунок 17 – Датчик s3

Таблица 7 – Свойства блока «Датчик s3»

Название	Имя свойства	Значение	Имя переменной
Значение по умолчанию	a	Значение на выходе блока, может быть векторным	
Имя контакта	cont_name	Имя переменной s3, считываемой блоком	
Тип контакта	cont_type	Тип данных выходной переменной при	

б/н	14
-----	----

Название	Имя свойства	Значение	Имя переменной
		генерации кода для полномасштабной модели (Float / Boolean / Integer)	
Формат имени переменной	decl_format	<префикс><компонент><контакт/запрос> / <префикс><контакт/запрос> / <префикс><запрос><контакт> / <контакт/запрос>	
Добавить в базу сигналов	dbm_add	Да / Нет	
Формат имени переменной	query		
Автоматически сбрасывать	auto_reset	Да / Нет	
Операция для нескольких контактов	vec_mode	Операция ИЛИ / Запаковка в вектор	
Функциональный план	plan		
Отлаживаемый компонент	debug_comp		
Дополнительный шаблон для файла DBM	add_decl		
Порт для сортировки	sort_port	Да / Нет	
Тип датчика	meter_type		
Общее кол-во знаков	precision		
Кол-во знаков после запятой	digits		
Формат числа	num_format	Основной / Найчный / Фиксированный / Целый	
Описание датчика	desc		

#### 2.4.12 Выходной контакт s3

Реализует запись переменной исполнительной системы со входа блока в базу данных (Рисунок 18 –).

 Выходной контакт s3

Рисунок 18 – Выходной контакт s3

Таблица 8 – Свойства блока «Выходной контакт s3»

Название	Имя свойства	Значение	Имя переменной
Значение по умолчанию	a	Значение на выходе блока, может быть векторным	
Имя контакта	cont_name	Имя переменной s3, считываемой блоком	

Название	Имя свойства	Значение	Имя переменной
Тип контакта	cont_type	Тип данных выходной переменной при генерации кода для полномасштабной модели (Float / Boolean / Integer)	
Формат имени переменной	decl_format	<префикс><компонент><контакт/запрос> / <префикс><контакт/запрос> / <префикс><запрос><контакт> / <контакт/запрос>	
Добавить в базу сигналов	dbm_add	Да / Нет	
Формат имени переменной	query		
Транслировать в исполнительную систему	translate_out		
Применять операцию ИЛИ	enable_or		
Операция для нескольких контактов	vec_mode	Операция ИЛИ / Запаковка в вектор	
Функциональный план	plan		
Отлаживаемый компонент	debug_comp		
Дополнительный шаблон для файла DBM	add_decl		
Порт для сортировки	sort_port	Да / Нет	
Переход на другой файл	cross_name		
Уникальный идентификатор	old_video	Уникальный идентификатор в базе сигналов системы s3	

### 2.4.13 Шаг интегрирования

Блок (Рисунок 19 –) формирует на выходе значение (величину) текущего шага интегрирования (расчёта, моделирования) задачи:

$$Y = f(t) = \Delta t$$

где Y – выходной сигнал блока; f(t),  $\Delta t$  – величина текущего шага интегрирования.

$\Delta t$  Шаг интегрирования

Рисунок 19 – Шаг интегрирования

Во встроенном языке программирования существует специальное служебное слово (переменная `stepsize`), также равная текущему значению шага интегрирования.

#### 2.4.14 Равномерный шум

Блок генерирует псевдослучайную последовательность вещественных чисел, равномерно распределенных в диапазоне от `x_min` до `x_max` (по Гауссу) (Рисунок 20 –).

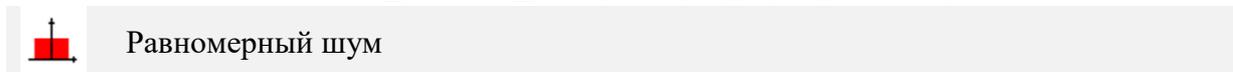


Рисунок 20 – Равномерный шум

Таблица 9 – Свойства блока «Равномерный шум»

Название	Имя свойства	Значение	Имя переменной
Минимальное значение	<code>xmin</code>		
Максимальное значение	<code>xmax</code>		
Период квантования в секундах	<code>qt</code>	Время в секундах, через которое будет обновляться случайный сигнал на выходе блока. Если <code>qt = 0</code> , то случайный сигнал на выходе блока будет обновляться после выполнения каждого очередного шага интегрирования.	

Если период квантования задан равным 0 (нулю), то случайный сигнал на выходе блока будет обновляться после выполнения каждого очередного шага интегрирования.

Параметры могут быть векторами, размерности которых должны быть одинаковы, в этом случае на выходе будет также вектор.

Именованные свойства могут быть заданы как локальные переменные модели (субмодели) в скрипте на языке программирования

#### 2.4.15 Входной контакт s3 матричный

Расширенная версия блока «Входной контакт s3». Реализует считывание двумерного массива  $M \times N$  из базы данных (Рисунок 21 –).

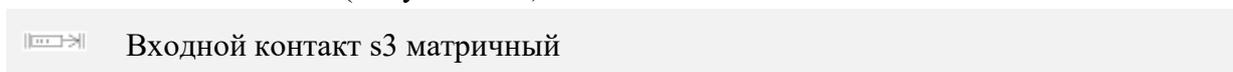


Рисунок 21 – Входной контакт s3 матричный

На выход блока передается векторный сигнал, представляющий собой исходную матрицу, упакованную по строкам: сначала идут элементы первой строки, затем элементы второй строки и т.д.

Таблица 10 – Свойства блока «Входной контакт s3 матричный»

Название	Имя свойства	Значение	Имя переменной
Значение по	<code>a</code>	- значение можно задать в виде:	

Название	Имя свойства	Значение	Имя переменной
умолчанию		<ul style="list-style-type: none"> <li>- матрицы коэффициентов: "[[a11, a12, a13,...];[a21, a22, a23,...];...]" – в этом случае размерность матрицы можно поределить по этому значению;</li> <li>- скаляра: "a0" – если все элементы можно инициализировать одинаковым значением – в этом случае размерность матрицы нужно задать через параметры блока.</li> </ul>	
Имя контакта	cont_name	Имя переменной s3, считываемой блоком	
Тип контакта	cont_type	Тип данных выходной переменной при генерации кода для полномасштабной модели (Float / Boolean / Integer)	
Размерность сигнала	set_dim	<ul style="list-style-type: none"> <li>- определяет способ задания размерности сигнала: <ul style="list-style-type: none"> <li>- Определить по значению – размерность определяется по числу элементов в параметра «Значение по умолчанию», в этом случае «Значение по умолчанию» должно содержать матрицу коэффициентов, например, "[[a11, a12, a13,...];[a21, a22, a23,...];...]";</li> <li>- Задать через параметры – число строк и столбцов задается в параметрах блока (параметры «Число строк» и «Число столбцов»);</li> </ul> </li> </ul>	
Число строк	nrow	Число строк в исходной матрице размерности сигнала	
Число столбцов	ncol	Число столбцов в исходной матрице размерности сигнала	
Формат имени переменной	decl_format		
Добавить в базу сигналов	dbm_add	Да / Нет	
Автоматически сбрасывать	auto_reset		
Операция для нескольких контактов	vec_mode	Копирование / Распаковка вектора	
Функциональный план	plan		
Отлаживаемый компонент	debug_comp		
Дополнительный	add_decl		

Название	Имя свойства	Значение	Имя переменной
шаблон для файла DBM			
Порт для сортировки	sort_port	Да / Нет	

### 2.4.16 Сумматор

Блок векторизован (Рисунок 22 –), реализует операцию алгебраического поэлементного суммирования входных сигналов с учетом весовых коэффициентов:

$$Y = a_1 \cdot u_1(t) + a_2 \cdot u_2(t) + a_3 \cdot u_3(t) + \dots + a_n \cdot u_n(t),$$

где  $Y$  – выходной сигнал блока;  $u_1(t), u_2(t), \dots, u_n(t)$  – входные сигналы;  $a_1, a_2, a_3, \dots, a_n$  – весовые коэффициенты для каждого входного сигнала блока.

Входные сигналы могут быть векторами  $u_1 = [u_1 1, u_1 2, \dots, u_1 m], u_2 = [u_2 1, u_2 2, \dots, u_2 m] \dots u_n = [u_n 1, u_n 2, \dots, u_n m]$ , размерности которых должны быть одинаковы, в этом случае на выходе будет также вектор:

$$y_i(t) = a_1 \cdot u_{1i}(t) + a_2 \cdot u_{2i}(t) + a_3 \cdot u_{3i}(t) + \dots + a_n \cdot u_{ni}(t),$$

где  $i = 1 \dots m$ ;  $y_i(t)$  –  $i$ -ый элемент выходного векторного сигнала блока;  $u_1 i(t), u_2 i(t)$  –  $i$ -ый элемент первого, второго и т.д. входного сигнала.



Сумматор

Рисунок 22 – Сумматор

Блок устанавливается на схему в нужном месте, присоединение блока к элементам схемы Свойства блока «Сумматор»

Название	Имя свойства	Значение	Имя переменной (где применимо)
Весовые множители для каждого из входов	a	Вектор действительных чисел, на которые умножаются значения входных сигналов блока.	

По умолчанию блок формирует выходной сигнал, равный сумме двух входных сигналов (весовых коэффициентов два, и оба равны 1). Именованные свойства могут быть заданы как локальные переменные модели (субмодели) в скрипте на языке программирования/

### 2.4.17 Сравнивающее устройство

Блок векторизован (Рисунок 23 –), реализует (по умолчанию) поэлементное вычитание второго входного векторного сигнала из первого:

$$\tilde{y}(t) = \vec{u}_1(t) - \vec{u}_2(t)$$

где  $y(t)$  – выходной сигнал блока;  $u_1(t), u_2(t)$  – входные сигналы.



Сравнивающее устройство

### Рисунок 23 – Сравнивающее устройство

В остальном (кроме поведения по умолчанию и изображения блока) блок аналогичен блоку «Сумматор» – можно задавать любые весовые коэффициенты для каждого из входов.

Входные сигналы могут быть векторами  $u1=[u1\ 1, u1\ 2, \dots, u1\ m]$ ,  $u2=[u2\ 1, u2\ 2, \dots, u2\ m]$  ...  $un=[un\ 1, un\ 2, \dots, un\ m]$ , размерности которых должны быть одинаковы, в этом случае на выходе будет также вектор:

$$Y = a_1 \cdot u_{1i}(t) + a_2 \cdot u_{2i}(t) + a_3 \cdot u_{3i}(t) + \dots + a_n \cdot u_{ni}(t),$$

где  $i = 1 \dots m$ ;  $y_i(t)$  –  $i$ -ый элемент выходного векторного сигнала блока;  $u_1\ i(t)$ ,  $u_2\ i(t)$  –  $i$ -ый элемент первого, второго и т.д. входного сигнала.

Таблица 11 – Свойства блока «Сравнивающее устройство»

Название	Имя свойства	Значение	Имя переменной (где применимо)
Весовые множители для каждого из входов	a	Вектор действительных чисел, на которые умножаются значения входных сигналов блока. По умолчанию весовые коэффициенты равны [1,-1] для реализации вычитания.	

По умолчанию блок формирует выходной сигнал, равный разности двух входных сигналов (весовых коэффициентов два, и они равны +1 и -1).

Именованные свойства могут быть заданы как локальные переменные модели (или субмодели) в скрипте на языке программирования, как глобальные сигналы.

#### 2.4.17.2 Сложение вектора с числом

В случае скалярного сигнала на втором входе, блок (Рисунок 24 –) реализует операцию суммирования каждого из элементов векторного сигнала на первом входе со скалярным сигналом на втором входе:

$$y_i = a_i \cdot (t) + b(t),$$

где  $a_i(t)$  –  $i$ -ый элемент входного векторного сигнала на первом входе;  $b(t)$  – сигнал на втором входе, скаляр;  $y_i(t)$  – элемент выходного векторного сигнала.

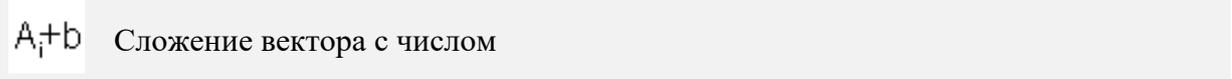


Рисунок 24 – Сложение вектора с числом

Размерность выхода равна размерности первого входа.

В случае векторного сигнала на втором входе, блок реализует операцию суммирования каждого из элементов векторного сигнала на первом входе поочередно с каждым из элементов векторного сигнала на втором входе. Размерность выходного сигнала при этом равна произведению размерностей сигналов на первом и втором портах:

$$y_k = a_i \cdot (t) + b_j(t),$$

где  $a_i(t)$  –  $i$ -ый элемент входного векторного сигнала на первом входе;  $b_j(t)$  –  $j$ -ый элемент входного векторного сигнала на втором входе;  $y_k(t)$  – элемент выходного векторного сигнала.

### 2.4.17.3 Суммирование элементов вектора

Блок (Рисунок 25 –) реализует суммирование всех элементов векторных входных сигналов. Выход – скалярная величина (по умолчанию):

$$y(t) = a_1 \cdot \sum u_i(t) + a_2 \cdot \sum a_2(t) + \dots$$

где  $u_1(t), u_2(t), \dots$  – элементы векторов входных сигналов блоков (элементы векторов  $u_1(t), u_2(t), \dots$ ),  $y(t)$  – выходной сигнал блока,  $a_1, a_2, \dots$  – весовые коэффициенты входов.

 Суммирование элементов вектора

Рисунок 25 – Суммирование элементов вектора

Если задано свойство суммировать через  $k > 1$  элементов, то размерность выходного сигнала равна  $k$ . Например, алгоритм блока для  $k=2$ :

$$y_1 = a_1 (u_{11}(t) + u_{13}(t) + u_{15}(t) + \dots) + a_2 (u_{21}(t) + u_{23}(t) + u_{25}(t) + \dots) + \dots;$$

$$y_2 = a_1 (u_{12}(t) + u_{14}(t) + u_{16}(t) + \dots) + a_2 (u_{22}(t) + u_{24}(t) + u_{26}(t) + \dots) + \dots;$$

Таблица 12 – Свойства блока «Суммирование элементов вектора»

Название	Имя свойства	Значение	Имя переменной (где применимо)
Весовые коэффициенты входов	a	Числа $a_1, a_2, \dots$ , на которые умножаются значения входных портов блока.	

Блок формирует скалярный выходной сигнал при  $k=1$  и векторный выходной сигнал при  $k > 1$ .

Именованные свойства могут быть заданы как локальные переменные модели (или субмодели) в скрипте на языке программирования, как глобальные сигналы.

### 2.4.18 Перемножитель

Блок (Рисунок 26 –) векторизован, реализует операцию алгебраического поэлементного суммирования входных сигналов с учетом весовых коэффициентов:

$$y(t) = u_1(t) \cdot u_2(t) \cdot \dots \cdot u_n(t)$$

где  $y(t)$  – выходной скалярный сигнал блока;  $u_1(t), u_2(t)$  – первый, второй и т.д. входной сигнал;

или поэлементное умножение входных величин блока (при векторных сигналах на входе):

$$y_i(t) = u_{1i}(t) \cdot u_{2i}(t) \cdot \dots \cdot u_{ni}(t)$$

где  $y_i(t)$  –  $i$ -ый элемент выходного векторного сигнала блока;  $u_{1i}(t), u_{2i}(t)$  –  $i$ -ый элемент первого, второго и т.д. входного сигнала.

 Перемножитель

Рисунок 26 – Перемножитель

Таблица 13 – Свойства блока «Перемножитель»

Название	Имя свойства	Значение	Имя переменной (где применимо)
Количество портов	port	Количество входных портов блока n (целое натуральное число)	

Размерность выходного сигнала всегда равна размерности входных сигналов. Блок формирует скалярный выходной сигнал при скалярных входных сигналах. Если входные сигналы векторные, то их размерности должны совпадать.

Именованное свойство может быть задано как локальная переменная модели (субмодели) в скрипте на языке программирования, как глобальный сигнал.

### 2.4.19 Перемножение элементов вектора

Блок реализует перемножение всех элементов всех векторов входов (Рисунок 27 –).

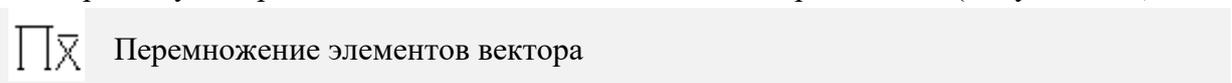


Рисунок 27 – Перемножение элементов вектора

Выходной сигнал – скалярная величина:

$$y(t) = \prod_{i=1}^{n_1} u_{1i}(t) \times \prod_{i=1}^{n_2} u_{2i}(t) \times \prod_{i=1}^{n_n} u_{ni}(t)$$

где  $u_{1i}(t)$ ,  $u_{2i}(t)$ , ... – элементы векторов входных сигналов блоков (элементы векторов  $u_1(t)$ ,  $u_2(t)$ , ..., размерность которых равна  $n_1$ ,  $n_2$ ,  $n_n$ );  $y(t)$  – выходной сигнал блока.

Таблица 14 – Свойства блока «Перемножение элементов вектора»

Название	Имя свойства	Значение	Имя переменной (где применимо)
Количество входов	port	Число (целое натуральное число), указывающее сколько входных векторных сигналов поступает в блок. Число входных портов блока равно этому количеству	

Блок всегда формирует скалярный выходной сигнал. Входные сигналы могут быть как векторными, так и скалярными.

Именованное свойство может быть задано как локальная переменная модели (субмодели) в скрипте на языке программирования, как глобальный сигнал.

#### 2.4.19.2 Умножение на число

Блок (Рисунок 28 –) реализует умножение каждого из элементов векторной величины на первом входе на скалярную величину на втором входе.:

$$y_i = a_1 \cdot (t) \cdot b(t),$$

где  $a_i(t)$  –  $i$ -ый элемент входного векторного сигнала  $a(t)$  на первом входе,  $b(t)$  – сигнал на втором входе, скаляр;  $y_i(t)$  – элемент выходного векторного сигнала  $y(t)$ . Размерность выхода равна размерности первого входа.

 Умножение на число

Рисунок 28 – Умножение на число

#### 2.4.20 Делитель

Блок (Рисунок 29 –) реализует поэлементное деление первой входной величины на вторую (при скалярных сигналах на входе):

$$y(t) = \frac{u_1(t)}{u_2(t)}$$

где  $y(t)$  – выходной скалярный сигнал блока;  $u_1(t)$ ,  $u_2(t)$  – входные сигналы;

или поэлементное деление входных величин блока (при векторных сигналах на входе):

$$y_i(t) = \frac{u_{1i}(t)}{u_{2i}(t)}$$

где  $y_i(t)$  –  $i$ -ый элемент выходного векторного сигнала блока  $y(t)$ ;  $u_{1i}(t)$ ,  $u_{2i}(t)$  –  $i$ -ые элементы первого и второго входных сигналов,  $u_1(t)$ ,  $u_2(t)$  – размерность входов и выхода.

 Делитель

Рисунок 29 – Делитель

Обязательным условием является неравенство делителя нулю  $U_{2i}(t) \neq 0!$

Размерность выходного сигнала всегда равна размерности входных сигналов. Блок формирует скалярный выходной сигнал при скалярных входных сигналах. Если входные сигналы векторные, то их размерности должны совпадать.

#### 2.4.21 Деление скаляра на вектор

Блок (Рисунок 30 –) реализует деление первой скалярной входной величины на каждый из элементов второй величины. Размерность выхода равна размерности второго входа:

$$y_i(t) = \frac{u_1(t)}{u_{2i}(t)}$$

где  $u_1(t)$  – сигнал на первом входе, скаляр;  $u_{2i}(t)$  –  $i$ -ый элемент входного векторного сигнала на втором входе  $u_2(t)$ ;  $y_i(t)$  – элемент выходного векторного сигнала,  $y(t)$ ,  $n$  – размерность второго входного сигнала и выходного сигнала блока.

 Деление скаляра на вектор

Рисунок 30 – Деление скаляра на вектор

Размерность выхода равна размерности первого входа.

Обязательным условием является неравенство делителя нулю ( $u_{2i}(t) \neq 0!$ )

Размерность выходного сигнала равна размерности второго входного сигнала. Если второй входной сигнал – скалярный, то блок реализует деление первого входного сигнала на второй входной сигнал, в этом случае выход – скалярная величина.

### 2.4.22 Усилитель

Блок (Рисунок 31 –) производит умножение (усиление) входной величины на заданный коэффициент усиления  $k$  (в случае скалярного сигнала на входе):

$$y(t) = a \cdot u(t)$$

или поэлементное умножение входного векторного сигнала на заданный коэффициент усиления  $a$ :

$$\vec{y}(t) = a \cdot \vec{u}(t) \Leftrightarrow y_i(t) = a \cdot u_i(t)$$

где  $u_i(t)$  –  $i$ -ый элемент входного векторного сигнала  $u(t)$ ,  $y_i(t)$  – элемент выходного векторного сигнала  $y(t)$ ,  $n$  – размерность входа и выхода.



Рисунок 31 – Усилитель

Таблица 15 – Свойства блока «Усилитель»

Название	Имя свойства	Значение	Имя переменной (где применимо)
Коэффициент усиления	$a$	Действительное число $a$ , скалярный множитель для входной величины.	

Размерность выходного сигнала равна размерности входного сигнала.

Именованное свойство может быть задано как локальная переменная модели (субмодели) в скрипте на языке программирования, как глобальный сигнал.

### 2.4.23 Векторный усилитель

Блок (Рисунок 32 –) реализует функцию умножения (усиления) мгновенных значений элементов входного векторного сигнала на соответствующие элементы заданного векторного коэффициента усиления:

$$y_i(t) = a_i \cdot u_i(t)$$

где  $y_i(t)$  –  $i$ -ый элемент выходного векторного сигнала блока  $y(t)$ ;  $u_i(t)$  –  $i$ -ый элемент входного сигнала  $u_i(t)$ ,  $n \geq 1$  – размерность входа и выхода,  $a_i$  –  $i$ -ый элемент вектора коэффициентов усиления  $a$ . Размерность выхода равна размерности входа и вектора коэффициентов усиления.

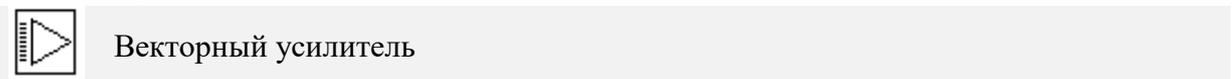


Рисунок 32 – Векторный усилитель

Таблица 16 – Свойства блока «Векторный усилитель»

Название	Имя свойства	Значение	Имя переменной (где применимо)
Коэффициент усиления	$a$	Вектор коэффициентов усиления $a = [a_1, a_2, \dots a_n]$ .	

Размерность выходного сигнала равна размерности входного сигнала и размерности вектора коэффициентов усиления.

Именованные свойства могут быть заданы как локальные переменные модели (субмодели) в скрипте на языке программирования.

#### 2.4.23.2 Абсолютное значение

Блок (Рисунок 33 –) реализует вычисление модуля каждого из элементов входной векторной величины:

$$y(t) = |u(t)|$$

где  $u(t)$ ,  $y(t)$  – входной и выходной сигналы. Для векторного входного сигнала:

$$y_i(t) = |u_i(t)|$$

где  $u_i(t)$ ,  $y_i(t)$  –  $i$ -ые элементы входного и выходного векторных сигналов.

 Абсолютное значение

Рисунок 33 – Абсолютное значение

#### 2.4.24 Размножитель

Блок реализует умножение входного скалярного сигнала (или элементов векторного входного сигнала) на вектор-столбец (или векторы-столбцы) коэффициентов усиления (Рисунок 34 –).

 Размножитель

Рисунок 34 – Размножитель

Если на входе скалярный сигнал  $u(t)$ , блок реализует функцию многоканального усилительного звена:

$$\vec{y}(t) = \vec{k} \times u(t) \ll == \gg y_i(t) = k_i(t)$$

где  $u(t)$  – входной сигнал (скаляр),  $k_i$  – элемент вектора-столбца коэффициентов усиления  $k = [k_1, k_2, \dots, k_m]$ ,  $y_i(t)$  – элемент выходного векторного сигнала  $y(t) = [y_1(t), y_2(t), \dots, y_m(t)]$ . В этом случае размерность выходного сигнала определяется размерностью вектора коэффициентов усиления.

Если на вход блока подается векторный сигнал (размерностью  $n \geq 2$ ), то в качестве коэффициентов размножения следует задавать соответствующее количество  $n$  вектор-столбцов одинаковой размерности  $m$ . Количество векторов-столбцов коэффициентов размножения обязательно должно быть равно размерности входного векторного сигнала. При этом блок реализует преобразование входного векторного сигнала по алгоритму:

$$u(t) = [u_1(t), u_2(t), \dots, u_n(t)] \rightarrow \begin{bmatrix} u_{1,1}(t) = k_{1,1} \cdot u_1(t) \dots u_{1,n}(t) = k_{n,1} \cdot u_n(t) \\ u_{2,1}(t) == k_{1,2} \cdot u_1(t) \dots u_{2,n}(t) = k_{n,2} \cdot u_n(t) \\ \dots \\ u_{n,1}(t) == k_{1,m} \cdot u_1(t) \dots u_{m,n}(t) = k_{n,m} \cdot u_n(t) \end{bmatrix}$$

$$y(t) = [u_{1,1}(t), \dots, u_{1,n}(t), \dots, u_{m,1}(t), \dots, u_{m,n}(t), ]$$

Входной вектор  $u(t)$ , интерпретируемый как вектор-строка, преобразуется в матрицу размера  $n \times m$ , каждый  $j$ -ый столбец которой получен из  $j$ -ого элемента входного вектора умножением на  $j$ -ый вектор-столбец коэффициентов усиления. Затем матрица считывается по строкам и полученный вектор  $y(t)$  подается на выход блока.

Таблица 17 – Свойства блока «Размножитель»

Название	Имя свойства	Значение	Имя переменной (где применимо)
Коэффициенты размножения	M	Вектор-столбец (или векторы-столбцы) коэффициентов усиления	

Размерность выходного сигнала равна произведению размерности входного сигнала и количества векторов коэффициентов размножения –  $n \times m$ .

Именованные свойства могут быть заданы как локальные переменные модели (субмодели) в скрипте на языке программирования, как глобальные сигналы проекта.

#### 2.4.25 Знак

Блок (Рисунок 35 –) реализует вычисление знака входного сигнала по следующему алгоритму:

$$y(t) = \begin{cases} -1, & \text{если } u(t) < 0 \\ 0, & \text{если } u(t) = 0 \\ +1, & \text{если } u(t) > 0 \end{cases}$$

где  $u(t)$ ,  $y(t)$  – входной и выходной векторные сигналы. В случае векторного входного сигнала блок вычисляет знак каждого из элементов входного вектора:

$$y_i(t) = \begin{cases} -1, & \text{если } u_i(t) < 0 \\ 0, & \text{если } u_i(t) = 0 \\ +1, & \text{если } u_i(t) > 0 \end{cases}$$

где  $u_i(t)$ ,  $y_i(t)$  –  $i$ -ые элементы входного и выходного векторных сигналов  $u(t)$ ,  $y(t)$ ,  $n \geq 2$  – размерность входного сигнала.

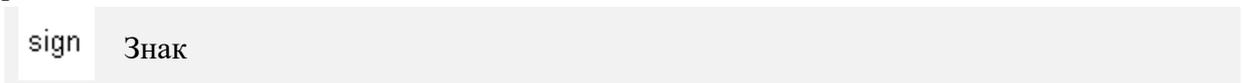


Рисунок 35 – Знак

#### 2.4.26 Целая часть

Блок реализует операцию выделения целой части входного сигнала (Рисунок 36 –).

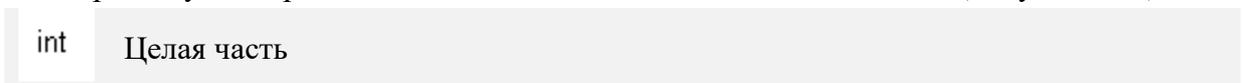


Рисунок 36 – Целая часть

Под целой частью понимается ближайшее целое число в меньшую сторону (по абсолютной величине). То есть целая часть положительного числа – меньше или равна числу, а целая часть отрицательного числа – больше или равна числу (Рисунок 37 –):

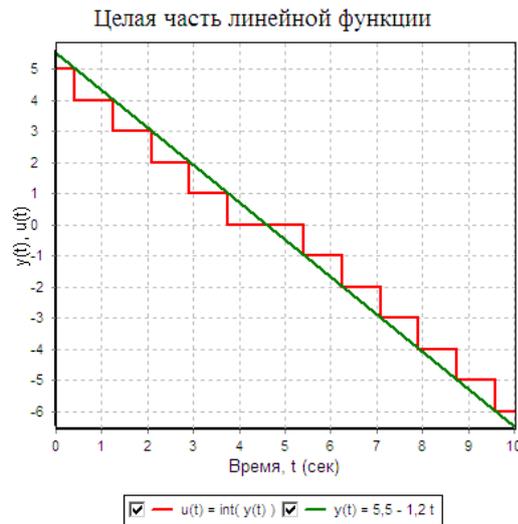


Рисунок 37 – Демонстрация работы блока

Алгоритм вычисления целой части числа:

$$y(t) = \begin{cases} \text{int}[u(t)] = \min(n \in Z \mid n \leq u(t)), & \text{при } u(t) \geq 0 \\ \text{int}[u(t)] = \min(n \in Z \mid n \leq u(t)), & \text{при } u(t) < 0 \end{cases}$$

где  $y(t)$  – выходной сигнал,  $u(t)$  – входной сигнал блока,  $Z$  – множество целых чисел. В случае векторного входного сигнала, блок реализует операцию выделения целой части каждого из элементов входной векторной величины.

#### 2.4.27 Дробная часть

Блок реализует операцию выделения дробной части входного сигнала (Рисунок 38 –).

`frac`    Дробная часть

Рисунок 38 – Дробная часть

Под дробной частью понимается разность между сигналом и его целой частью. Дробная часть положительного числа – неотрицательна, а дробная часть отрицательного числа – неположительна (Рисунок 39 –):

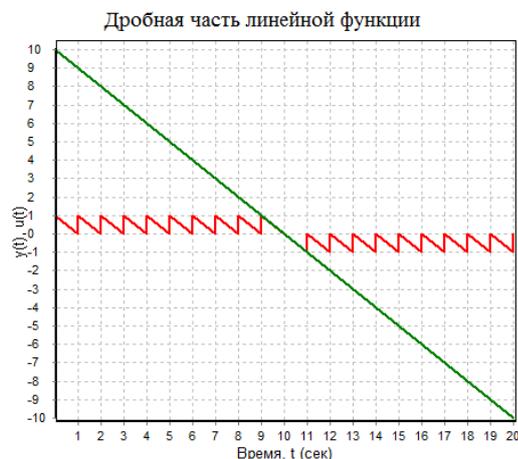


Рисунок 39 – Демонстрация работы блока

Алгоритм вычисления дробной части числа:

$$y(t) = \text{frac}[u(t)] = u(t) - \text{int}[u(t)]$$

где  $y(t)$  – выходной сигнал,  $u(t)$  – входной сигнал блока. В случае векторного входного сигнала, блок реализует операцию выделения дробной части каждого из элементов входной векторной величины.

#### 2.4.28 Линейный преобразователь

Блок реализует линейную интерполяцию входной величины (Рисунок 40 –).



Линейный преобразователь

Рисунок 40 – Линейный преобразователь

При этом значение входа, равное минимальной входной величине соответствует нулю на выходе, а значение входа равное максимальной входной величине соответствует единице на выходе:

$$y(t) = \frac{u(t) - x_{min}}{x_{max} - x_{min}}$$

где  $u(t)$  – входной скалярный сигнал,  $x_{min}$ ,  $x_{max}$  – параметры блока,  $y(t)$  – выходной скалярный сигнал. Если на вход поступает векторный сигнал, то параметры блока должны быть заданы векторами соответствующей входу размерности, тогда и на выходе будет векторный сигнал:

$$y_i(t) = \frac{u_i(t) - x_{min,i}}{x_{max,i} - x_{min,i}}$$

где  $u(t) = [u_1, u_2, \dots, u_n]$  – входной векторный сигнал,  $x_{min,i}$ ,  $x_{max,i}$  –  $i$ -ые элементы векторов свойств блока,  $u(t) = [u_1, u_2, \dots, u_n]$  – выходной векторный сигнал,  $n \geq 1$  – размерность входного и выходного сигналов, а также свойств блока.

Таблица 18 – Свойства блока «Линейный преобразователь»

Название	Имя свойства	Значение	Имя переменной (где применимо)
Минимальное значение входа	a	Минимальная входная величина	
Максимальное значение входа	b	Максимальная входная величина	

По умолчанию блок формирует скалярный выходной сигнал.

Именованные свойства могут быть заданы как локальные переменные модели (субмодели) в скрипте на языке программирования, как глобальные сигналы проекта.

#### 2.4.29 Динамическая выборка

Блок выполняет процедуру выбора элемента из вектора по указанному номеру (Рисунок 41 –).



Динамическая выборка

Рисунок 41 – Динамическая выборка

Алгоритм работы блока:

Цикл  $i = 1 \dots n$

Если  $u_{1i}(t) = i$  то  $y(t) = u_{2i}(t)$ ; Выход из цикла

Конец цикла

где  $u_1(t)$  – первый входной скалярный сигнал, определяющий номер выбираемого элемента, начиная с 1;  $u_{2i}(t)$  –  $i$ -ый элемент второго входного векторного сигнала  $u_2(t)$ , из которого осуществляется выборка;  $n \geq 1$  – размерность второго входа, натуральное число.

#### 2.4.30 Выборка по активному элементу

Блок (Рисунок 42 –) выполняет процедуру выбора элемента из второго вектора по активному элементу управляющего (первого) вектора по следующему алгоритму:

$Y(t) = 0$ ,

Цикл  $i = 1 \dots n$

Если  $u_{1,i}(t) = i$  то  $y(t) = u_{2,i}(t)$ ; Выход из цикла

Конец цикла

где  $u_1 = [u_{1,1}, u_{1,2}, \dots, u_{1,n}]$  – управляющий входной векторный сигнал;  $u_2 = [u_{2,1}, u_{2,2}, \dots, u_{2,n}]$  – входной векторный сигнал выбираемых значений;  $n \geq 1$  – размерность входных сигналов.



Выборка по активному элементу

Рисунок 42 – Выборка по активному элементу

## 3 МОДЕЛИРОВАНИЕ СХЕМ АВТОМАТИКИ

### 3.1.1 Создание проекта системы автоматки

Для создания схемы автоматки необходимо открыть САПФИР:

- Выбрать тип проекта: «Файл» → «Создать» → «AUTO»;
- Включить видимость панели библиотеки классов: «Вид» → «Показать панель библиотеки классов»;
- Загрузить библиотеку классов для моделирования схем автоматки: «Библиотека» → «Загрузить библиотеку классов»;
- Сохранить текущий проект.

### 3.1.2 Настройка расчетного слоя (решателя)

Чтобы вызвать окно свойств решателя, требуется на панели инструментов «Слои» левой кнопкой мыши нажать на кнопку  (редактировать свойства текущего решателя) (Рисунок 43 –):

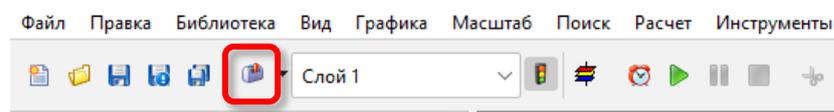


Рисунок 43 – Панель инструментов «Слои», кнопка редактирования свойств текущего слоя

Редактировать параметры расчета можно только для выбранного слоя (Рисунок 44 –):



Рисунок 44 – Выбор слоя (решателя) для дальнейшей работы

Описание параметров расчета слоя (решателя) для выбранного слоя дано в документе «Программное обеспечение САПФИР. Руководство пользователя, часть 2», раздел 1.4. Внешний вид окна свойств решателя приведен на Рисунок 45 –:

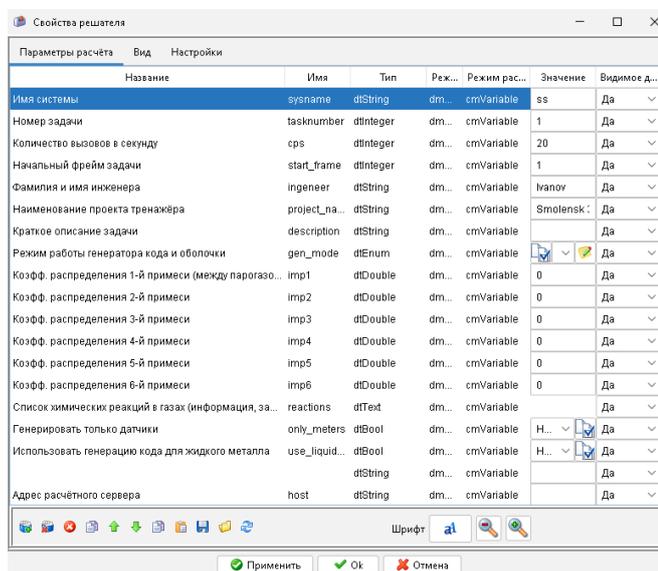


Рисунок 45 – Внешний вид окна свойств расчета слоя (решателя)

Перечни всех параметров расчета задаются шаблоном расчетного условия. Параметры слоя (решателя) могут быть сохранены в файл и затем использованы в другом проекте. Можно скопировать параметры слоя (решателя) проекта и вставить в другой проект.

### 3.1.3 Блоки для моделирования систем автоматики

Для моделирования систем автоматики и логических устройств основные элементы находятся во вкладках «Субструктуры», «Отказы», «Источники», «Операторы», «Векторные», «Функции», «Динамические», «Нелинейные», «Данные», «Ключи», «Логические», «Триггеры», «Задержки и импульсы», «Релейные блоки» (Рисунок 46 –).

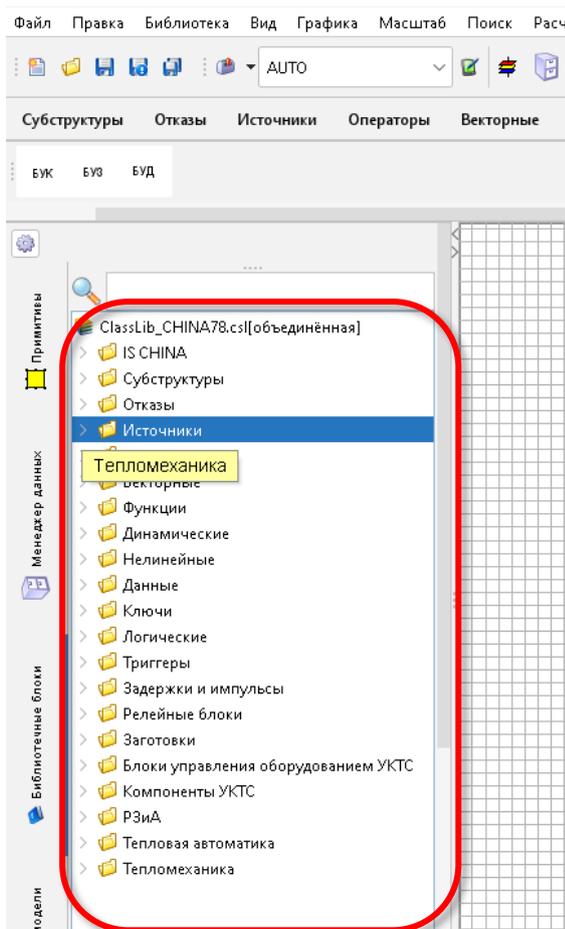


Рисунок 46 – Блоки для моделирования систем автоматики и логических устройств в панели примитивов

### 3.1.4 Режимы отображения рабочего поля

Режимы отображения рабочего поля описаны в разделе **Error! Reference source not found.**

### 3.1.5 Основы моделирования систем автоматики

Основы моделирования систем автоматики даны в документе «Программное обеспечение САПФИР. Руководство пользователя, часть 3/1», раздел 3.1.

### 3.2 Особенности генерации компонентов автоматике

После запуска САПФИР`а создать новый проект на основе шаблона AUTO, для чего выбрать пункт меню «**Файл**» → «**Создать**» (Рисунок 47 –):

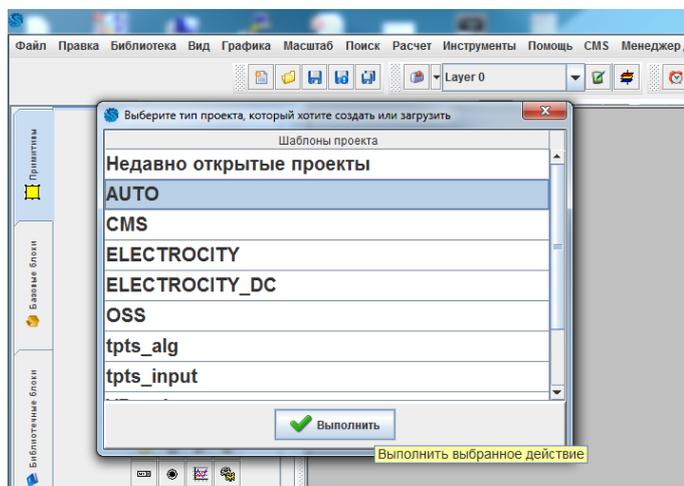


Рисунок 47 – Создание нового проекта на основе шаблона AUTO

Загрузить библиотеку классов, для чего выбрать пункт меню «**Библиотека**» → «**Загрузить библиотеку классов**» (Рисунок 48 –):

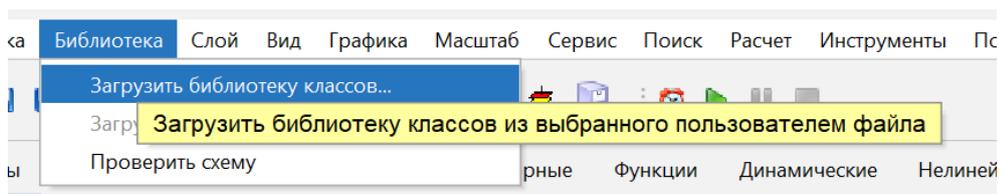


Рисунок 48 – Загрузка библиотеки классов

Выбрать и загрузить файл ClassLib\_AUTO.csl (библиотеки находятся в каталоге: \Sapfir\_64\SETTINGS).

Отредактировать свойства решателя (Рисунок 49 – и Рисунок 50 –), для чего нажать кнопку  в панели инструментов «Правка»:

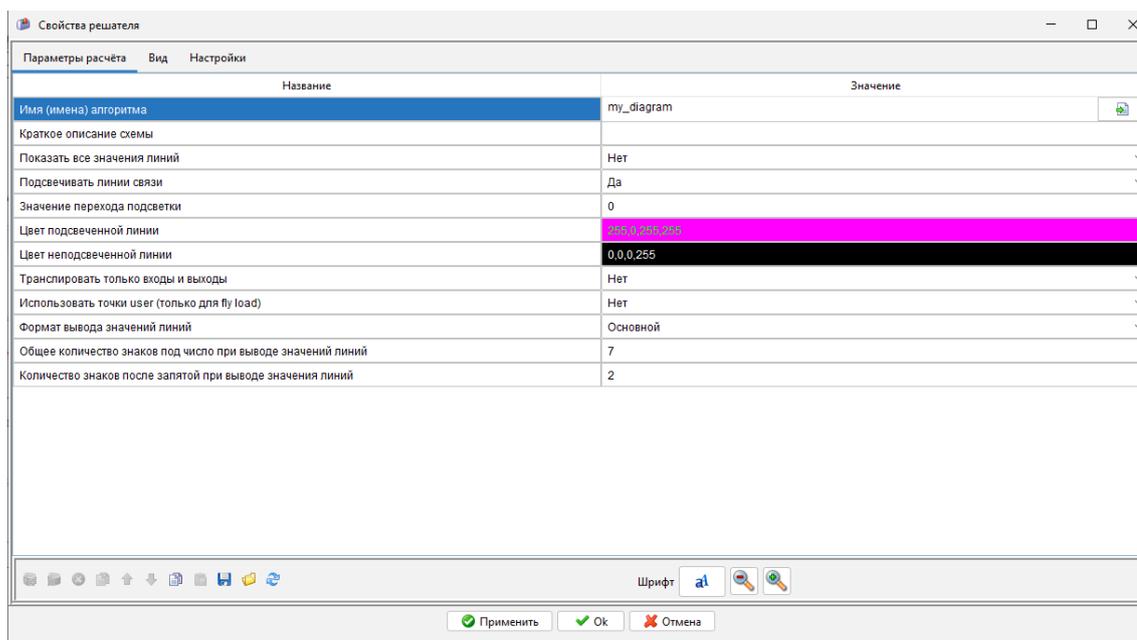


Рисунок 49 – Свойства решателя – Параметры расчета

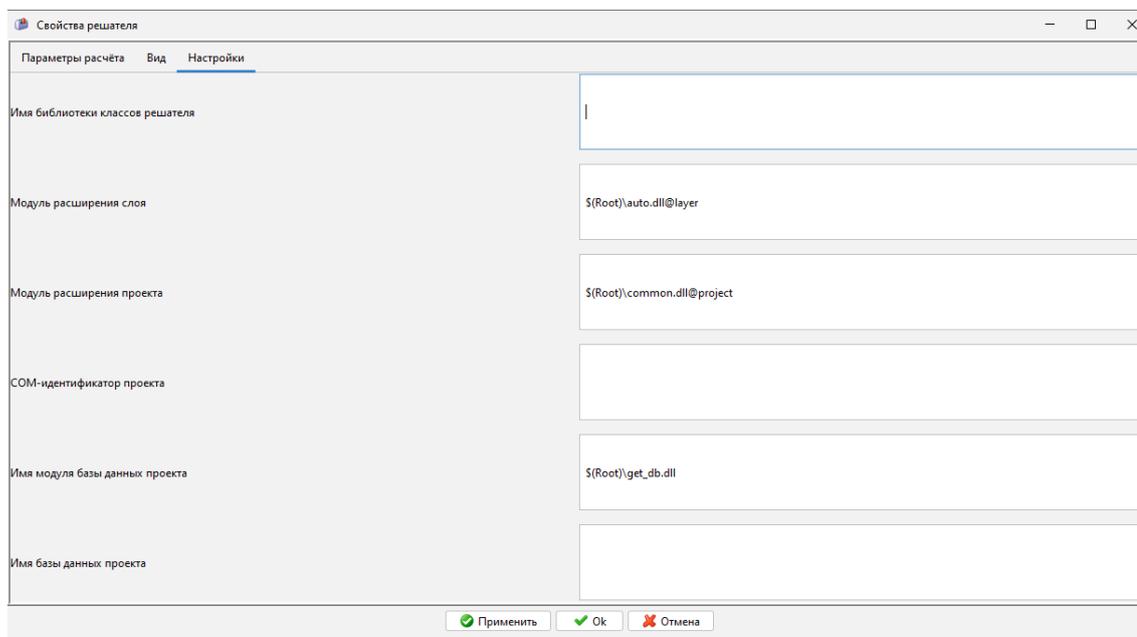


Рисунок 50 – Свойства решателя – Настройка

В строке «Имя (имена) алгоритма» написать необходимое. Имя должно состоять из латинских букв, цифр

Имя должно ясно отражать суть содержимого. Желательно использовать понятные и понятийные сокращения или обозначения, чтобы имя алгоритма было легко распознаваемо и отличимо от других.

Выбрать пункт меню «Инструменты» → «База данных», выбрать проект проверить и, при необходимости, изменить настройки (Рисунок 51 –):

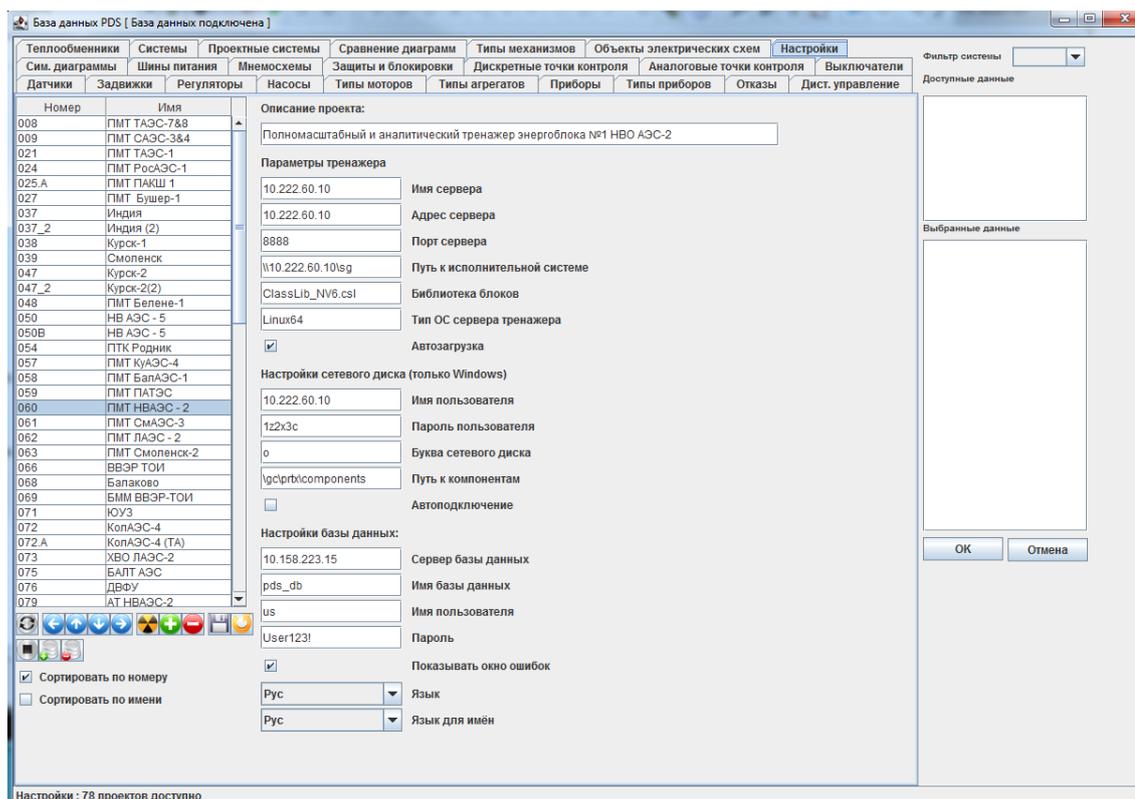


Рисунок 51 – Пример настроек проекта

Выбрать пункт меню «**Менеджер диаграмм**» → «**Менеджер диаграмм автоматки**» (Рисунок 52 –):

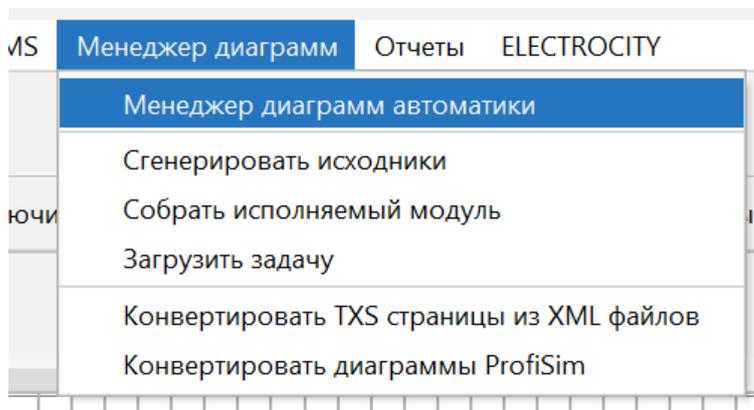


Рисунок 52 – Вызов менеджера диаграмм

В открывшемся окне создать конфигурацию для каждой используемой в проекте системы (например,  $\{sys\}_{vlv}$ ,  $\{sys\}_{pmp}$ ), заполнить настройки и сохранить.

Можно использовать заготовки, подготовленные ранее (для других систем и/или других проектов) (Рисунок 53 –):

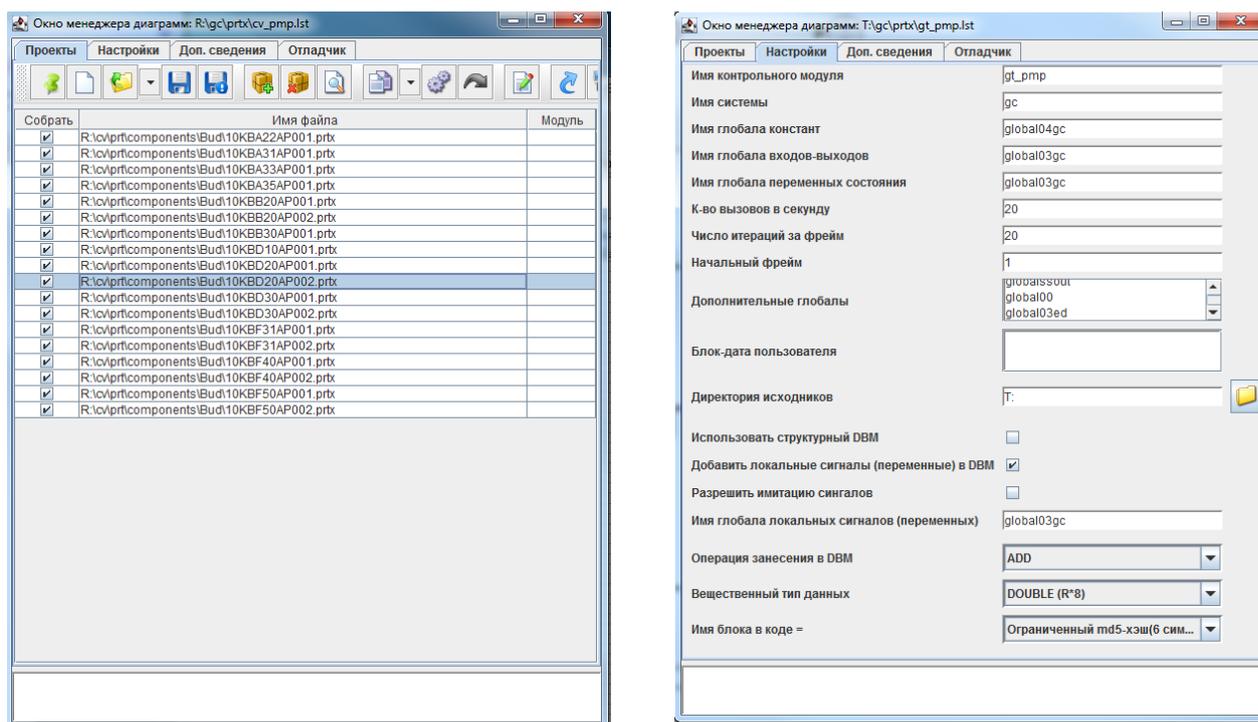


Рисунок 53 – Пример заполнения настроек для используемых в проекте систем

Рекомендуются следующие настройки (вкладка «Настройки»):

- Для насосов
  - К-во вызовов в секунду: 20.
  - Число итерация за фрейм: 20;
  - Начальный фрейм: 1
- Для задвижек
  - К-во вызовов в секунду: 20.
  - Число итерация за фрейм: 1;
  - Начальный фрейм: 1
- Дополнительные глобалы:
  - globalyp
  - globallo
  - globalao
  - global03gc
  - global04gc
  - globalvuout
  - globalvuin
  - globalss
  - globalssin
  - globalssout
  - global00
  - global03ed
- Операция занесения в dbm:
  - ADD – следует выбирать, когда добавлены новые переменные.
  - MDD – следует выбирать, когда в существующие переменные внесены изменения.

Подготовить схему и алгоритмы поведения автоматики.

Должны быть заготовлены шаблоны компонентов в виде файлов .prtx. Они должны лежать в папке SG\gc\prtx\components.

В папке SG\{sys}\prt\ должна быть папка \components. Если ее нет, то нужно создать.

В PDS для задвижек, насосов должно быть заполнено поле «Шаблон» в виде {имя шаблона.prtx}. Имена шаблонов должны совпадать с именами шаблоны компонентов в виде файлов .prtx.

Для первой генерации необходимо выбрать дополнительное подменю для выборочной генерации кода «Сгенерировать исходники» (Рисунок 54 –):

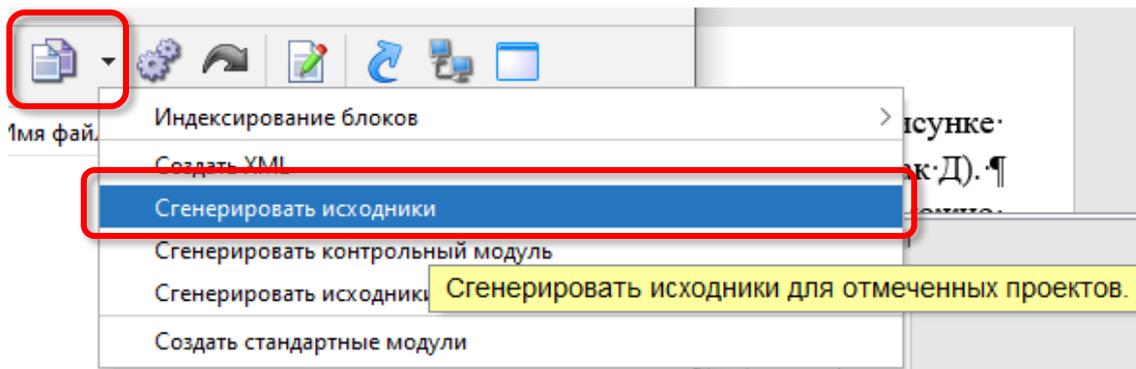


Рисунок 54 – Дополнительное подменю для выборочной генерации кода «Сгенерировать исходники»

При окончании генерации возможно предупреждение «Генерация исходников завершена с предупреждениями», что не является ошибкой.

В процессе генерации создаются файлы:

- \*\_dbm, \*\_sel, \*\_dsd (сохраняются в папке: {диск проекта}\{sys}\gc\selects).
- \*.c (сохраняются в папке: {диск проекта}\{sys}\gc\sources).

Перед каждой генерацией рекомендуется папку {диск проекта}\gc\src очищать от старой генерации.

Следующим шагом проводится сборка исполняемого модуля, для чего нажать кнопку «Собрать исполняемый модуль»:

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

- 1) ГОСТ 19.105–78 ЕСПД. Общие требования к программным документам (Раздел «Аннотация»).

