

**Утверждаю**

генеральный директор

ООО «3В Сервис»



Петухов В.Н.



## Среда динамического моделирования технических систем SimInTech™

### Техническое задание

Модуль генерации кода для систем реального времени

ШИФР ГК16ТЗ

Москва, 2016



## Аннотация

В данном программном документе приведено техническое задание на разработку модуля генерации кода для управляющих систем реального времени.

Указаны основные требования к модулю.



## СОДЕРАЖНИЕ

<b>Аннотация</b>	2
<b>1. Введение</b>	4
<b>2. Основание для разработки</b>	6
<b>3. Термины и определения</b>	7
<b>4. Назначение разработки</b>	14
<b>5. Требования к программе</b>	15
<b>5.1 Требование к функциональным характеристикам</b>	15
<b>5.1.1 Требование к составу выполняемых функций</b>	15
5.1.1.1 Подготовка исходных данных в SimInTech	15
5.1.1.1.1 Создание шаблона для генерации	16
5.1.1.1.2 Создание базы данных сигналов	16
5.1.1.1.3 Создание функционально-блочной диаграммы SimInTech	16
5.1.1.2 Верификация прикладного программного обеспечения	17
5.1.1.3 Генерация исходного кода программы	17
<b>5.1.2 Требование к организации входных данных</b>	19
5.1.2.1 Требования к шаблону для генерации кода	19
5.1.2.2 Требования к базе данных сигналов	19
5.1.2.3 Требования к проекту прикладного ПО в SimInTech	20
5.1.2.3.1 Требования к атрибутам расчетной схемы SimInTech	20
5.1.2.3.2 Требования к модульности расчетной схемы прикладного ПО	21
5.1.2.3.3 Требования к оформлению расчетной схемы	21
5.1.2.3.4 Требования к математической модели SimInTech	22
<b>5.1.3 Требование к организации выходных данных</b>	23
5.1.3.1 Состав файлов исходного кода прикладного ПО	24
5.1.3.2 Содержание файлов исходного кода прикладного ПО	25
5.1.3.2.1 Заголовочный файл	25
5.1.3.2.2 Основной текст программы	27
5.1.3.2.3 Инициализация начальных значений переменных	27
5.1.3.2.4 Запоминание состояния	28
5.1.1 Требование к временным характеристикам	28
<b>5.2 Требование к надежности</b>	28
5.2.1 Требование к обеспечению надежного функционирования программы	28
5.2.2 Время восстановления после отказа	29
<b>5.3 Условия эксплуатации</b>	29
5.3.1 Климатические условия эксплуатации	29
5.3.2 Требования к видам обслуживания	29
5.3.3 Требования к численности и квалификации персонала	29
<b>6. Технико-экономические показатели</b>	33
<b>7. Стадии и этапы разработки</b>	34
<b>8. Порядок контроля и приемки</b>	35
<b>Приложение 1. Список блоков для генерации кода</b>	36
<b>Приложение 2. Список функций и операторов языка программирования для генерации кода</b>	43



## 1. Введение

Программа для ЭВМ “Среда динамического моделирования SimInTech”, (сокращенное название «SimInTech»), свидетельство о регистрации №2010617758 - современная среда интеллектуальной системы автоматизированного **проектирования** (САПР), предназначенная для детального исследования и анализа нестационарных процессов в системах автоматического управления, в следящих приводах и роботах, в любых технических системах, описание динамики которых может быть реализовано методами структурного моделирования.

SimInTech обеспечивает создание алгоритмов управления в виде функционально-блочных диаграмм. ПО содержит в себе математическое ядро для проведения динамического расчета созданного алгоритма управления путем задания входных воздействий и анализа изменений внутренних параметров и выходных значений во время моделирования.

SimInTech является базовым программным обеспечением для разрабатываемого модуля. Комплексная система моделирования систем управления и программирования приборов включает в себя:

- модуль генерации кода для автоматической генерации исходных кодов и исполняемых модулей;
- среду разработки для проектирования алгоритмов управления в виде наглядной функционально-блочной диаграммы;
- систему исполнения программ для приборов, для выполнения сгенерированных при помощи генератора кода исполняемых модулей на приборах.

Под прибором следует понимать программируемые логические контроллеры (ПЛК) в составе промышленных компьютеров, работающие под управление POSIX-совместимых операционных систем реального времени.

Предметом работы и данного ТЗ является разработка новой версии модуля для SimInTech, обеспечивающего автоматическое создание исходных кодов программ на универсальных языках программирования для исполнения в системах управления реального времени. В рамках работ по созданию новой версии будет выполнено расширения списка



блоков для которых осуществляется генерация кода, а так же добавлена возможность создания кода Си из скриптов написанных на встроенном языке программирования.

Предыдущая версия модуля генерации кода SimInTech имеет сертификат соответствия № РОСС RU.0001.01АЭ00.77.10.2446.



## 2. Основание для разработки

Основанием для разработки является:

Договор N 437 - 01 от 26.07.2013 по теме: «Разработка программного обеспечения верхнего уровня программно-технического комплекта средств автоматического управления».

Заказчик ООО «Московский завод «ФИЗПРИБОР».

План особо важных работ по доработке программного обеспечения на 2015 год.  
Утвержден 15.02.2013. ООО «ЗВ Сервис».



### 3. Термины и определения

**3.1 Анимация (animation):** процесс, посредством которого указанное в спецификации поведение демонстрируется с реальными значениями, полученными из задающих поведение выражений и некоторых входных величин.

**3.2 Алгоритм (algorithm):** набор последовательности работы ПО для преобразования входных данных программы или подпрограммы в выходные данные.

**3.3 Схема алгоритма (algorithm diagram):** алгоритм, составленный в виде графической схемы (функционально-блочной диаграммы) на проблемно ориентированном языке SimInTech.

**3.4 Лист алгоритма (algorithm page):** схема алгоритма в SimInTech, оформленная в соответствии с требованиями к оформлению спецификации ПО - графическое изображение алгоритма управления или его части в виде функционально-блочной диаграммы. Является частным случаем расчетной схемы SimInTech.

**3.5 Группа алгоритмов (group of algorithm):** совокупность листов алгоритмов, объединённые в субмодель в среде SimInTech.

**3.6 Расчетная схема SimInTech (SimInTech simulation diagram):** структурная схема, созданная в окне графического редактора SimInTech, описывающая на предметно-ориентированном языке математическую модель алгоритма, процесса или объекта, динамику поведения которого во времени, можно представить в виде системы алгебраических и дифференциальных уравнений в форме Коши. На основании расчетной схемы ядро SimInTech обеспечивает математическое моделирование динамического поведения объекта во времени с заданной точностью.

**3.7 Блок (block):** базовый элемент расчетной схемы (функционально-блочной диаграммы в SimInTech). Основными атрибутами блока являются его графический образ, свойства (задаваемые пользователем), параметры (вычисляемые блоком), математическая модель, входные и выходные порты. Графическое изображение может быть статическим или анимированным.

В математическом аспекте блоки представляют собой операторы преобразования входных сигналов блока в его выходные сигналы. Совокупность блоков и соединяющих их



порты линий связи образует расчетную схему объекта или алгоритм прикладного программного обеспечения в SimInTech.

**3.8 Библиотека блоков (block library):** набор блоков, объединенных по определенному (в основном, функциональному) признаку. Набор блоков содержится в файле с расширением «.csl» и предназначен для набора проекта SimInTech.

**3.9 Видеокادر (mnemo):** проект SimInTech в виде интерактивной и анимированной структурной схемы, позволяющий при моделировании оказывать воздействие на алгоритм или модель и наблюдать результаты работы.

**3.10 Проект SimInTech (SimInTech project):** файл, содержащий расчетную схему, созданную в графическом редакторе SimInTech, сохраненный на диске в виде бинарного и/или текстового файла с уникальным именем и расширением «.prt» (для бинарного) и «.xprrt» (для текстового) файла. Проект SimInTech содержит расчетную схему – математическую модель, предназначенную для расчета тем или иным математическим решателем или расчетным кодом.

**3.11 Пакет SimInTech (SimInTech pack):** - файл, содержащий перечень проектов SimInTech и порядок их совместного запуска на расчет (моделирование), имеющий расширение «.pak» и являющийся основным файлом для организации комплексной модели. Проекты, запускаемые на расчет в пакетном режиме, имеют одну базу сигналов в памяти компьютера и единый синхронизатор расчетного (модельного) времени, за счет чего они могут обмениваться значениями граничных (входных и выходных) сигналов между собой на каждом шаге расчета и осуществлять моделирование в едином синхронном модельном времени.

**3.12. Прикладная функция (application function):** функция системы контроля и управления, выполняющая задачу, связанную с контролируемым процессом, а не с функционированием самой системы.

**3.13. Проблемно-ориентированный язык (application oriented language):** компьютерный язык, специально разработанный для определенного типа применений и используемый лицами, являющимися специалистами в данном типе применений. В рамках данного документа графическое представление алгоритмов в виде функционально-блочной диаграммы SimInTech рассматривается как вид проблемно-ориентированного языка.





**3.14. Прикладное программное обеспечение (application software):** часть программного обеспечения системы контроля и управления, которая обеспечивает выполнение прикладных функций.

**3.15. Автоматизированная генерация кода (automated code generation):** функция автоматизированных инструментов, позволяющая преобразовывать проблемно-ориентированный язык в форму, пригодную для компиляции и выполнения на целевой системе.

**3.16. Идентификатор объекта (identification):** целое число (тип данных integer), являющееся уникальным внутренним именем каждого объекта на схеме и используемое для прямого обращения к объекту на низком уровне во многих функциях встроенного языка программирования.

**3.17. Комплексная модель (complex model):** совокупность проектной базы сигналов и файлов проектов, содержащих расчетные схемы алгоритмов прикладной программы и математическую модель объекта управления, созданных в SimInTech. Комплексная модель является виртуальным объектом, динамическое поведение которого совпадает с поведением реального технического объекта с заданной степенью точности.

**3.18. Линия связи (connection line):** служебный блок в виде полилинии, второй базовый элемент расчетной схемы, соединяющий выходной порт одного блока и входной порт другого блока. В общем случае линия связи может соединять множество входных портов с одним выходным портом. В математическом аспекте линии связи являются шинами данных (сигналов) и осуществляют направленную передачу данных от выходов блока к входам других блоков.

**3.19. Параметр блока (blocks parameter):** формируемая (вычисляемая) блоком переменная, характеризующая работу блока.

**3.20. Свойство блока (blocks property):** задаваемая пользователем характеристика (константа или переменная величина, какого-либо определённого типа данных) для работы блока.

**3.21. Стандартная подпрограмма (standard program):** см. типовое решение.

**3.22. Субмодель (макроблок) (sub model):** блок, математическая модель которого задана в виде структурной схемы, расположенной «внутри» субмодели. Макроблоки позволяют реализовать принцип вложенности структурных схем, являются одним из



механизмов создания и сохранения в библиотеке новых типов блоков. Математическая модель макроблока может быть сохранена в отдельном файле (в отдельном проекте) для многократного использования в других проектах.

**3.23. Типовое решение (standard program):** один или несколько связанных специальных алгоритмов, определяющих принцип управления, и (или) измерения данного типа объекта. Применяется для генерации типового алгоритма для объектов данного типа. Например, типовым решением может быть способ контроля достоверности датчика (на обрыв, зашкал, превышение скорости изменения измеряемой величины, перевод измеренного сигнала в физические единицы измерения для дальнейшего использования в алгоритмах). Типовое решение также называют стандартной подпрограммой.

**3.24. Типовой алгоритм (standard algorithm):** алгоритм конкретного объекта (устройство управления, канал измерения), созданный автоматически на основе типового решения для данного объекта.

**3.25. Канал (channel):** совокупность взаимосвязанных компонентов внутри системы, имеющая один выход. Канал теряет свою идентичность тогда, когда сигналы на единственном выходе сочетаются с сигналами от других каналов, например, от канала контроля или от канала активизации защиты. [Глоссарий МАГАТЭ NS-G-1.3]

**3.26. Вычислительный прибор (вычислительный узел, computing unit)** (см. Компьютер).

**3.27. Компьютер (computer):** программируемое функциональное устройство, которое состоит из одного или нескольких процессоров и периферийного оборудования, управляется хранящимися внутри программами и способно выполнять основные вычисления, включая многочисленные арифметические или логические операции без вмешательства в этот процесс человека.

**П р и м е ч а н и е** – Компьютер может быть автономным устройством или может состоять из нескольких взаимосвязанных устройств.

**3.28. Компьютерная программа (computer program):** набор упорядоченных команд и данных, которые описывают операции в форме, приемлемой для их выполнения компьютером.

**3.29. Компьютерная система (computer-based system):** система контроля и управления, функции которой, в большей своей части, зависят от использования



микропроцессоров, программируемого электронного оборудования или компьютеров, либо полностью определяются таким использованием.

**П р и м е ч а н и е** – Эквивалентно следующему: цифровые системы, системы с программным обеспечением, программируемые системы.

**3.30. Данные (data):** представление информации или команд в виде, пригодном для передачи, интерпретации или обработки с помощью компьютера.

**П р и м е ч а н и е** – Данные, необходимые для определения параметров и для реализации прикладных и служебных функций в системе называются «прикладными данными».

**3.31. Динамический анализ (dynamic analysis):** процесс оценки системы или компоненты, основанный на их поведении в процессе работы. В противоположность статическому анализу. [IEEE 610]

**3.32. Отказ (failure):** отклонение реальной работы от запланированной.

**3.33. Универсальный язык (general-purpose language):** компьютерный язык, предназначенный для всех видов применения.

**П р и м е ч а н и е 1** – Программное обеспечение операционной системы групп оборудования обычно реализуется с использованием универсальных языков. **П р и м е ч а н и е 2** – Примеры: Ада, Си, Паскаль.

**3.34. Инициализировать (initialize):** установить счетчики, переключатели, адреса или содержимое устройств памяти на нулевое значение или на другие начальные величины в начале или в заданной точке выполнения компьютерной программы.

Для математической модели установить все переменные состояния в начальное положение, задать значения всех переменных, необходимых для проведения расчета.

**3.35. Комплексные тестирования (integration tests):** тестирования, проводимые во время процесса интеграции технического и программного обеспечения до валидации компьютерной системы с целью проверки совместимости программного обеспечения и технического обеспечения компьютера.

**3.36. Библиотека (library):** набор связанных элементов ПО, сгруппированных вместе, но индивидуально отбираемых для включения в окончательный продукт ПО.

**3.37. Операционное системное программное обеспечение (operation system software):** программное обеспечение, выполняемое на целевом процессоре во время работы,



такое как драйверы и сервисы ввода/вывода, управление прерываниями, планировщик, драйверы связи, библиотеки прикладных программ, диагностирование во время работы, управление резервированием и смягченной деградацией.

**3.38. Ролевое управление доступом (role-based access control):** управление доступом на основе правил, определяющих разрешение доступа пользователей к объекту (функции, данные) не на индивидуальном основании, а на основании принадлежности к группам с идентичными задачами.

**3.39. Сигнальная траектория (signal trajectory):** динамика изменения всех состояний оборудования, внутренних состояний, входных сигналов и действий оператора, определяющих выходы системы.

**3.40. Программное обеспечение (ПО) (software):** программы (т.е. набор упорядоченных команд), данные, правила и любая соответствующая документация, относящаяся к работе компьютерной системы контроля и управления. [МЭК 62138, 3.27]

**3.41. Разработка ПО (software development):** стадия жизненного цикла ПО, которая приводит к созданию ПО системы контроля и управления или программного продукта. Она охватывает деятельность, начиная от спецификации требований и до валидации и установки на объекте.

**3.42. Модификация ПО (software modification):** изменение в уже согласованном документе (или документах), ведущее к изменению рабочей программы.

**П р и м е ч а н и е** – Модификации ПО могут происходить либо в процессе первоначальной разработки ПО (например, устранение ошибок, обнаруженных на поздних этапах разработки), либо когда ПО уже находится в эксплуатации.

**3.43. Версия ПО (software version):** экземпляр программного продукта, полученный путем модификации или корректировки предыдущего программного продукта.

**3.44. Спецификация (specification):** документ, в котором полным, точным и проверяемым образом изложены требования, проектные свойства и другие характеристики системы или компоненты и, часто, процедуры подтверждения удовлетворения этим требованиям.

**П р и м е ч а н и е** – Существуют различные типы спецификаций, например, спецификация требований к ПО или спецификация проекта.



**3.45. Статический анализ (static analysis):** процесс оценки системы или ее компоненты, основанный на ее форме, структуре, содержании или документации. В дополнение к динамическому анализу.

**3.46. Системное ПО (system software):** часть ПО системы контроля и управления, созданная для конкретного компьютера или семейства оборудования с целью облегчения разработки, эксплуатации и модификации этих объектов и связанных с ними программ.

**3.47. Валидация системы (system validation):** подтверждение путем проверки и предоставления других свидетельств того, что система в целом соответствует спецификации требований (функциональность, время отклика, устойчивость к дефектам и ошибкам, запас прочности).

**3.48. Верификация (verification):** подтверждение путем проверки и предоставления объективных свидетельств того, что результаты деятельности соответствуют целям и требованиям, определенным для этой деятельности.



## 4. Назначение разработки

Создаваемое программное обеспечение предназначено для автоматической генерации кода программного обеспечения, для применения при разработке автоматизированных систем управления технологическими процессами (АСУ ТП) на базе операционных систем реального времени (QNX, КПДА и др). Применение SimInTech должно обеспечить сокращение времени выполнения проекта и общей стоимости создания систем управления за счет:

- Уменьшение количества ошибок в алгоритмах управления за счет предварительной проверки на стадии разработки проекта, путем математического моделирования в SimInTech.
- Исключение ошибочных отклонений реализации программы в промышленном контроллере от проекта, созданного в SimInTech.
- Сокращение количество ошибок, выявляемых на стадии отладки АСУ ТП на реальном объекте.

Для достижения вышеуказанных целей выполняется разработка модуля к SimInTech. После выполнения работ согласно данному ТЗ программный комплекс должен обеспечить:

- создание проекта системы контроля и управления, включая создание спецификации требований к проекту;
- документирование проекта системы управления;
- создание прикладного программного обеспечения на проблемно-ориентированном языке (функционально-блочная диаграмма SimInTech);
- создание математической модели прикладного программного обеспечения;
- динамический анализ прикладного программного обеспечения системы управления с целью верификации его работы;
- автоматическую генерацию кода прикладного ПО на языке Си;
- генерацию кода прикладного ПО на языке FIL.



## **5. Требования к программе**

### **5.1 Требование к функциональным характеристикам**

#### **5.1.1 Требование к составу выполняемых функций**

Разрабатываемый модуль должен обеспечить автоматическое создание исходного кода для управляющих программ систем управления реального времени в составе ПО «Среда динамического моделирования технических систем «SimInTech».

Разрабатываемый модуль предназначен для создания прикладного ПО.

Для систем контроля и управления важных для безопасности, подготовка исходных данных должна происходить с учетом рекомендаций, изложенных в документе «Типовой процесс создания прикладного программного для систем управления, важных для безопасности АЭС».

Разрабатываемый модуль работает в составе «SimInTech», где осуществляется разработка прикладной программы на предметно-ориентированном языке для генерации кода.

##### **5.1.1.1 Подготовка исходных данных в SimInTech**

Исходными данными для модуля генерации кода являются:

1. База данных сигналов в SimInTech, описывающая переменные программы, включая входные, выходные сигналы, а так же необходимые промежуточные переменные прикладного ПО.
2. Расчетная схема SimInTech, описывающая прикладное программное обеспечение в графическом виде в SimInTech. Расчетная схема SimInTech является программой, созданной на предметно-ориентированном языке в графическом виде и, одновременно, однозначной математической моделью прикладного программного обеспечения. С помощью схемы SimInTech можно создавать спецификации требований к ПО.
3. Пользовательские настройки процесса генерации кода, такие как: один или несколько исходных файлов, целевая платформа для последующего исполнения создаваемого кода, тип операционной системы, тип процессора, такт исполнения программы, язык программирования Си или FIL и т.п.



#### 4. Файлы шаблонов для генерации кода.

Исходные данные для генерации кода подготавливаются в несколько этапов.

##### **5.1.1.1.1 Создание шаблона для генерации**

Данный этап выполняется один раз для выбранной программно-аппаратной платформы (операционная система, процессор, компиляторы и линковщики). Требования к шаблону приведены в п. 7.1.1.1 настоящего ТЗ. Результатом выполнения данного этапа является набор всех необходимых файлов для компиляции прикладного ПО для выбранной программно-аппаратной платформы.

Создание шаблона кода не является предметом данного технического задания.

##### **5.1.1.1.2 Создание базы данных сигналов**

Данный этап служит для удобного определения переменных программного обеспечения, включая входные и выходные сигналы, переменные состояния, вспомогательные переменные, используемые при создании расчетной схемы и прикладного ПО.

База данных сигналов – структурированный список переменных, содержащий все переменные проекта, сгруппированные в структуры согласно проекту. Группировка переменных должна происходить с учетом требований проекта. Для наименования переменных может быть использована система кодирования, принятая в проекте системы управления или в проекте технической системы, для которой разрабатывается система управления. Требования к базе данных сигналов приведены в п. 7.1.1.2 настоящего ТЗ.

##### **5.1.1.1.3 Создание функционально-блочной диаграммы SimInTech**

Функционально-блочная диаграмма SimInTech содержит полную спецификацию требований к прикладному программному обеспечению.

Требования к функционально-блочной диаграмме приведены в п. 7.1.1.3 настоящего ТЗ.

Создаваемая диаграмма должна быть связана с базой данных сигналов.

Должна быть предусмотрена возможность задания пароля на чтение, просмотр и изменение любой части диаграммы.





### **5.1.1.2 Верификация прикладного программного обеспечения**

Верификация прикладного программного обеспечения осуществляется путем математического моделирования в среде SimInTech с заданием вектора входных воздействий и анализом результата.

SimInTech должен обеспечить динамический анализ прикладного ПО с возможностью анимации любых параметров моделируемой системы.

Для целей верификации возможно задавать в SimInTech произвольный вектор входных воздействий.

Должна быть обеспечена возможность записи вектора входных воздействий для дальнейшего использования в качестве исходных данных для верификации программного обеспечения на целевой программно-аппаратной платформе.

SimInTech обеспечивает возможность проводить математическое моделирование работы прикладного программного обеспечения совместно с математической моделью объекта. В последнем случае вектор входных воздействий формируется с использованием динамической модели объекта.

### **5.1.1.3 Генерация исходного кода программы**

5.1.1.3.1 Модуль генерации кода должен обеспечивать анализ топологически сложных расчетных схем больших размерностей (сотни и тысячи блоков, сигналов, параметров). Модуль генерации кода должен обеспечивать сортировку блоков схемы таким образом, чтобы выход любого блока в заданный момент времени мог быть рассчитан на основании данных, имеющихся к началу вычисления блока.

5.1.1.3.2 Анализ схемы должен обеспечивать создание кода из схем произвольной глубины вложенности.

5.1.1.3.3 Процедура анализа топологии структурной схемы должна находить алгебраические петли и развязывать их.

5.1.1.3.4 В случае нахождения ошибки или невозможности создания кода, модуль должен сообщать пользователю об ошибках.

5.1.1.3.5 Результатом работы модуля генерации кода должен быть код на заданном пользователем языке программирования (Си или FIL).



5.1.1.3.6 Программное обеспечение, созданное на основе сгенерированного исходного кода, должно функционировать в полном соответствии с математической моделью в среде SimInTech. Соответствие между исходным кодом ПО и математической моделью означает, что при одинаковых начальных значениях переменных в модели и ПО, для заданного вектора входных воздействий, изменяющегося во времени по заданному закону, изменения вектора выходных значений в математической модели SimInTech, должны соответствовать изменениям вектора выходных значений в программном обеспечении, созданном на основе исходного кода, с заданной точностью (см. Рисунок 1).



Рисунок 1. Требования к эквивалентности модели и сгенерированного кода

5.1.1.3.7 Поскольку разработка средств компиляции и исполнительных систем для исходного кода не входит в состав данного задания, то верификация соответствия может быть выполнена, с использованием компилятора на базе ПО GCC версии 4.8.3, с исполнением ПО на операционной системе Windows.

При использовании других средств компиляции объектного кода должно быть проведено сравнение работы модели и объектного кода, в том числе в режиме работы на компьютере.

5.1.1.3.8 Модуль генерации кода не должен вносить изменения в исходные файлы проекта.



## 5.1.2 Требование к организации входных данных

### 5.1.2.1 Требования к шаблону для генерации кода

Шаблон для генерации кода должен быть разработан до создания прикладного программного обеспечения в среде SimInTech. Шаблон для генерации кода должен размещаться в папке SimInTech/CodeTemplates/[Имя шаблона]

В состав шаблона генерации кода должны входить все файлы, необходимые для настройки компилятора и линковщика для конкретной программной аппаратной платформы.

В состав шаблона генерации кода должны входить автоматически запускаемые командные файлы для запуска компиляции и создания прикладного программного обеспечения.

В состав набора командных файлов, при необходимости, должны входить команды для загрузки созданного программного обеспечения на аппаратуру.

Описание файлов шаблона генерации кода должно находиться в руководстве системного программиста.

### 5.1.2.2 Требования к базе данных сигналов

База сигналов должна содержать в себе структурированный список переменных для функционального ПО.

Имена переменных должны быть уникальны.

Имя переменной должно содержать латинские буквы и цифры.

Имя переменной в базе данных сигналов должно состоять из двух частей:

- 1) Имя группы сигналов;
- 2) Имя сигнала в группе.

Переменные в прикладном ПО должны формироваться путем комбинации имени группы сигналов и имени сигнала в группе, соединенных символом подчеркивания «\_».

Должна быть предусмотрена возможность экспорта базы данных сигналов в текстовый или табличный файл.

База данных сигналов должна проверяться на корректность автоматическими средствами SimInTech.



### 5.1.2.3 Требования к проекту прикладного ПО в SimInTech

Проект программного обеспечения в среде SimInTech должен создаваться в виде набора функционально-блочных диаграмм – совокупностью одной или нескольких расчетных схем и базы данных сигналов.

Для идентификации пользователя проект алгоритмов может быть защищен паролем.

Проект должен содержать в себе программу на прикладном языке программирования SimInTech, оформленную согласно требованиям п. 5.1.2.3.1.

Проект в среде SimInTech должен представлять собой расчетную схему – математическую модель, рассчитываемую стандартным математическим ядром SimInTech, отвечающую требованиям п. 5.1.2.3.2.

Проект в среде SimInTech должен быть оформлен таким образом, чтобы была возможность осуществить печать проекта в виде документа.

Проект в среде SimInTech должен проверяться средствами автоматической проверки.

#### 5.1.2.3.1 Требования к атрибутам расчетной схемы SimInTech

Расчетная схема должна содержать следующие атрибуты:

- Организация – название организации разработчика.
- Проект – название проекта.
- Система – название системы, для которой создается прикладное ПО.
- Разработал – Фамилия и инициалы разработчика.
- Проверил – Фамилия и инициалы проверяющего.
- Утвердил – Фамилия и инициалы утверждающего.
- Ревизия – номер ревизии схемы.
- Дата ревизии – дата выпуска ревизии проекта ПО.

Диаграмма в SimInTech должна содержать уникальный идентификатор, осуществляющий однозначную связь между расчетной схемой и сгенерированными исходными кодами.



#### **5.1.2.3.2 Требования к модульности расчетной схемы прикладного ПО**

Расчетная схема SimInTech должна быть разделена на листы, объединенные в группы алгоритмов.

Каждый лист и каждая группа алгоритмов, должны проектироваться таким образом, чтобы можно было провести динамический анализ путем математического моделирования с помощью расчетного ядра SimInTech, задав входные воздействия через базу данных сигналов. Таким образом, каждая группа алгоритмов может рассматриваться, разрабатываться и тестироваться как отдельный модуль программы.

#### **5.1.2.3.3 Требования к оформлению расчетной схемы.**

Расчетная схема SimInTech должна быть наглядной и однозначно читаемой. Изображения блоков и их функциональное содержание должно быть описано в руководстве пользователя.

Лист алгоритма должен представлять собой субмодель, внутренняя структурная схема которой разделена на четыре области: область штампа; таблица входов; область логики алгоритма; таблица выходов.

Область штампа содержит атрибуты организации, исполнителя, проекта, технологической системы и т.д. В таблице входов указываются имена входных по отношению к данному листу сигналов, выбираемых из базы сигналов проекта.

В области логики реализуется структурная схема алгоритма, обеспечивающая необходимое преобразование входных сигналов листа алгоритма в его выходные сигналы.

В таблице выходов указываются имена выходных сигналов листа алгоритма. Содержание таблицы выходов автоматически переносится в соответствующую запись базы сигналов проекта (в таблицу «Алгоритмы», запись с полным именем листа алгоритма).

Основным атрибутом листа алгоритма является имя листа - уникальный в пределах алгоритма цифро-буквенный код. Субмодель «Лист алгоритма» вставляется в структурную схему субмодели «группа алгоритмов». Полное имя листа алгоритма складывается из имени алгоритма, которому он принадлежит, и имени листа.

Совокупность листов алгоритмов, имеющих одно общее имя, но разные имена листов, объединяются в понятие «Группа Алгоритмов». На структурной схеме представляет собой субмодель, внутренняя структурная схема которой состоит из блоков типа «Лист алгоритма», либо «группа алгоритмов». Основным атрибутом алгоритма является его имя -



уникальный в пределах всего комплексного проекта (проекта системы управления) цифро-буквенный код.

#### **5.1.2.3.4 Требования к математической модели SimInTech**

Модель должна отвечать следующим требованиям:

Должно быть обеспечено однозначное начальное состояние – значение всех переменных в модели SimInTech в начальный момент времени моделирования.

Модель должна быть вычисляемой. Запуск модели на расчет в среде SimInTech с заданными вектором входных воздействий, не должен приводить к ошибкам моделирования.

Модель должна обеспечивать верификацию программного обеспечения за счет динамического моделирования.

Модель может содержать вложенные структуры (субмодели) произвольной глубины вложенности.

Модель должна содержать только блоки, перечисленные в Приложении 1 к данному техническому заданию.

Функциональные требования к допустимым блокам описаны в приложении А к настоящему техническому заданию

В состав модели могут входить скрипты на встроенном языке программирования.

Скрипты, входящие в состав модели, должны содержать только функции, перечисленные в Приложении 2 к данному техническому заданию.

Функциональные требования к функциям допустимым для генерации кода приведены в приложении Б к данному техническому заданию.



### 5.1.3 Требование к организации выходных данных

Выходными данными программы является код на языке Си или FIL.

Создаваемый код должен сохраняться в виде набора файлов, задаваемых шаблоном в указанном пользователем месте на жестком диске.

Создаваемый код на языке Си должен соответствовать требованиям стандарта ANSI C ISO/IEC 9899:1990 и требованиям компилятора GCC версии 3.4.5 и старше.

Возможна настройка исходного кода для заданной целевой программно-аппаратной платформы, которая не должна изменять математическое содержание ПО.

Создаваемый код на языке FIL должен соответствовать требованиям исполнительной системы.

Файлы с исходным кодом должны содержать всю информацию, необходимую для создания объектного файла, готового к исполнению на выбранной целевой платформе (аппаратной и программной).



### 5.1.3.1 Состав файлов исходного кода прикладного ПО

Файлы исходных кодов должны содержать:

1) Общую часть - зависимую от программно-аппаратной платформы часть кода. Эта часть кода одинакова для любого прикладного программного обеспечения, предназначенного для конкретной аппаратной платформы.

Общая часть кода должна входить в шаблон и обеспечивает компиляцию исходных кодов в объектные файлы выбранными средствами компиляции.

2) Прикладная часть кода – файлы, содержащие непосредственно исходный код прикладной программы, сгенерированные из схемы в SimInTech.

Имена файлов исходных кодов прикладного ПО должны совпадать с именем алгоритма, указанным в настройках схемы SimInTech, свойство **имя алгоритма**.

При генерации исходного кода для прикладного ПО должны быть созданы файлы следующие файлы:

1) **<имя алгоритма>.h** - заголовочный файл с описанием привязки переменных к рабочим массивам.

2) **<имя алгоритма>.inc** – текст основной программы на языке Си (без заголовка функции), которая выполняется на каждом шаге расчёта.

3) **<имя алгоритма>\_init.inc** – текст программы на языке Си, которая выполняется для инициализации начальных значений переменных.

4) **<имя алгоритма>\_state.inc** – текст программы на языке Си, которая выполняется после основной программы для запоминания состояний задержек на шаг интегрирования и некоторых других блоков, имеющих дискретные состояния с приоритетной сортировкой.





### 5.1.3.2 Содержание файлов исходного кода прикладного ПО

Содержание файлов исходных кодов, для функциональных диаграмм, должно быть эквивалентно содержанию файлов исходных кодов созданных сертифицированной версией модуля генерации кода.

#### 5.1.3.2.1 Заголовочный файл

Файл с именем **<имя алгоритма>.h** должен описывать привязку переменных к рабочим массивам прикладного ПО и содержать следующую информацию:

- Имя программы.
- Описание программы.
- Имя файла схемы SimInTech, из которой выполнена генерация кода.
- Определение использованных типов данных.
- Идентификатор базы данных, обеспечивающий проверку соответствия кода и актуальной базы данных сигналов.
- Идентификаторы расчетной схемы, обеспечивающий проверку соответствия кода и актуальной схемы, использованной для генерации кода.
- Константы с начальными значениями переменных модели.
- Общие размерностные параметры расчетной схемы SimInTech, использованной для генерации кода, включая:
  - количество внешних переменных;
  - количество динамических переменных;
  - количество переменных состояния;
  - количество констант.
- Настройки исходной расчетной схемы, для использования в сгенерированной программе, включая:
  - минимальный шаг интегрирования;
  - максимальный шаг интегрирования;
  - шаг синхронизации;
  - метод интегрирования;
  - относительная ошибка интегрирования;



- абсолютная ошибка интегрирования;
  - конечное время интегрирования;
  - количество итераций математических петель.
- Таблица имен и размерностей внешних переменных модели.
- Определение привязки внешних переменных к массиву указателей.
- Таблица имен и размерностей динамических переменных модели.
- Определения привязки динамических переменных к общему массиву.
- Таблица имён и размерностей переменных состояния модели.
- Определение структуры с переменными состояния модели.
- Таблица имён и размерностей констант модели.
- Определение структуры с константами модели.



#### 5.1.3.2.2 Основной текст программы

Файл с именем **<имя алгоритма>.inc** должен содержать текст основной программы на языке Си, без заголовочного файла, который выполняется на каждом шаге.

- Имя программы.
- Описание программы.
- Имя файла схемы SimInTech из которой выполнена генерация кода.
- Определение зарезервированных локальных (стековых) переменных.

Для прикладной программы должен быть создан селектор выполняемой секции кода. В зависимости от стадии выполнения программы должны быть созданы следующие секции:

- Код, выполняемый при остановке расчёта.
- Код, выполняемый на стадии вычисления значений производных динамических переменных.
- Код, выполняемый на стадии вычисления выходов всех блоков основного или промежуточного шага.
- Код, выполняемый на стадии вычисления значений функций алгебраических переменных.
- Код, выполняемый на стадии вычисления выходов всех блоков основного или промежуточного шага.

Для каждого блока схемы должен генерироваться код, соответствующий приложению 2.

Для каждого блока схемы в тексте исходных кодов, должен указываться уникальный идентификатор, соответствующий идентификатору блока на схеме SimInTech, из которого происходит генерация кода.

Для каждого блока в комментариях должно быть указано его имя на расчетной схеме SimInTech.

#### 5.1.3.2.3 Инициализация начальных значений переменных

Файл **<имя алгоритма>\_init.inc** – текст программы на языке Си, которая выполняется для инициализации начальных значений переменных, должен содержать:

- Имя программы.
- Описание программы.



- Имя файла схемы SimInTech из которой выполнена генерация кода.
- Определение зарезервированных локальных (стековых) переменных.
- Копирование данных из констант в рабочие массивы.

#### 5.1.3.2.4 Запоминание состояния

Файл <имя алгоритма>\_state.inc – текст программы на языке Си, которая выполняется после основной программы для запоминания состояний, задержек на шаг интегрирования и блоков, имеющих дискретные состояния:

- Имя программы.
- Описание программы.
- Имя файла схемы SimInTech из которой выполнена генерация кода.
- Определение зарезервированных локальных (стековых) переменных.
- Текст программ на языке Си, который вызывается на стадии обновления выходов (промежуточной или конечной) шага интегрирования после основной программы – запоминание состояний блоков задержки на шаг.

### 5.1.1 Требование к временным характеристикам

Требования не предъявляются.

## 5.2 Требование к надежности

### 5.2.1 Требование к обеспечению надежного функционирования программы

Для обеспечения надежной работы программного обеспечения, необходимо обеспечить:

- надежное электрическое питание технических средств;
- использование актуальных совместимых и лицензионных версий операционной среды Windows;
- использование актуальной и совместимой версии SimInTech;
- отсутствие вирусов в системе.



Модуль генерации кода должен гарантированно и без ошибок запускаться в соответствующей версии программного обеспечения «Среда динамического моделирования технических систем SimInTech».

Модуль генерации кода при любых сбоях и неполадках не должен вносить какие-либо изменения в исходные данные, используемые для генерации кода.

При любых некорректных действиях пользователя во время работы модуль генерации кода не должен вносить изменения в исходные данные для генерации кода.

Модуль генерации кода должен работать по возможности изолированно и не влиять на работу базового ПО SimInTech.

Модуль генерации кода не должен влиять на результаты математического моделирования в SimInTech.

### 5.2.2 Время восстановления после отказа.

Время восстановления после отказов, вызванного внешними причинами, в работающей операционной системе не должно превышать времени, необходимого на перезагрузку программного обеспечения SimInTech.

## 5.3 Условия эксплуатации

### 5.3.1 Климатические условия эксплуатации.

Требования по климатическим условиям эксплуатации должны соответствовать требованиям, предъявляемым к техническим средствам, на которых осуществляется работа модуля генерации кода.

### 5.3.2 Требования к видам обслуживания.

Для начала эксплуатации программа требует установки в составе ПО SimInTech.

Программа не требует специального обслуживания.

При обновлении базового ПО SimInTech, модуль генерации кода должен обновляться при необходимости для обеспечения совместимости версий.

### 5.3.3 Требования к численности и квалификации персонала.



Для работы с программой необходимы системный администратор и пользователь, возможно выполнение обеих функций одним работником с соответствующей квалификацией.

Для установки программы необходима квалификация системного администратора используемой операционной системы, а также соответствующие права администратора.

Для работы с программой необходима квалификация сертифицированного пользователя среды SimInTech.

## 5.4 Требования к составу и параметрам технических средств

Состав и параметры технических средств должны соответствовать рекомендациям со стороны разработчика используемой операционной системы.

## 5.5 Требования к информационной и программной совместимости

5.5.1 Модуль генерации кода должен работать с актуальной версией программы «Среда динамического моделирования технических систем SimInTech».

5.5.2 Модуль генерации кода должен корректно работать с файлами проектов SimInTech созданными в предыдущих версиях SimInTech начиная с версии 1.6.

5.5.3 Требования к операционной системе должны соответствовать требованиям предъявляемым базовым ПО SimInTech версии 1.6.

5.5.4 Модуль генерации кода должен работать с любыми исходными данными, совместимыми с используемой средой SimInTech.

5.5.5 Исходный код Си, созданный модулем должен быть пригоден для компиляции компилятором GCC версии 4.8.3.

5.5.6 Модуль должен быть написан на языке Delphi (Object Pascal) и компилироваться в среде Delphi XE 7 и старше.

## 5.6 Требования к маркировке и упаковке

Требования к маркировке и упаковке не предъявляются, модуль поставляется в электронном виде, в виде набора файлов.





## 5.7 Требования к транспортировке и хранению

Требования к транспортировке и хранению не предъявляться, модуль поставляется в электронном формате, в виде набора файлов.

## 5.8 Требования к программной документации

Состав программной документации должен включать в себя следующие документы:

- Техническое задание.
- Программа и методика испытаний.
- Отчет об испытаниях модуля генерации кода.
- Руководство программиста системы программирования для вычислительных приборов.
- Руководство пользователя системы программирования для вычислительных приборов.
- План обеспечения качества программного обеспечения.
- План управления конфигурацией системы.
- План верификации программного обеспечения.
- Анализ угроз защищенности.





## 6. Технико-экономические показатели

Разрабатываемый модуль предназначен для использования в системе проектирования систем управления и программирования вычислительных приборов реального времени.

Модуль генерации кода, обеспечивает сокращение ручного труда по созданию программ для систем управления. Создание модели системы управления в среде SimInTech позволяет сократить общее время разработки программ управления за счет:

- сокращения времени разработки за счет наглядного представления схем алгоритмов в графическом виде;
- сокращения ошибок за счет проверки математических моделированием процесса управления в различных режимах;
- сокращения времени для подготовки документации;
- сокращения натурных испытаний.

В сравнении с существующими системами автоматической генерации кода, разрабатываемый модуль является частью комплексной системы «Среды динамического моделирования технических систем SimInTech», что обеспечивает использование проектных моделей АЭС, созданных на этапе проектирования для проектирования и тестирования систем контроля и управления важных для безопасности АЭС.

Схема алгоритмов проверяется путем совместной работы с моделью технических систем АЭС, что повышает общее качество разрабатываемых алгоритмов управления, и позволяет произвести проверку работы алгоритмов в переходных и аварийных режимах.

Таким образом, создаваемый модуль обеспечивает сокращение времени разработки и повышение качества разрабатываемых систем управления.



## 7. Стадии и этапы разработки

Перед началом работ должны быть разработаны следующие документы:

- План обеспечения качества программного обеспечения.
- План управления конфигурацией системы.

Работа выполняется в один этап.

Работа должна выполняться с учетом требований плана обеспечения качества.

Все исходные коды разрабатываемого модуля должны разрабатываться с учетом плана управления конфигурацией системы.

При необходимости в документы, входящие в состав программной документации вносятся изменения, связанные с разрабатываемым модулем.

После завершения разработки программного обеспечения, разрабатывается программа и методика испытаний.



## 8. Порядок контроля и приемки

Для выполнения контроля создается комиссия, в состав которой должны входить представители отдела разработки и администрации.

Приемка осуществляется согласно «Программы и методики испытаний».

Результат испытаний должен быть зафиксирован протоколом.

По результатам приёмки создается акт испытаний.

При необходимости в документы, входящие в состав программной документации вносятся изменения, связанные с разрабатываемым модулем.



## Приложение 1. Список блоков для генерации кода

- Блок «Шаг интегрирования»
- Блок «Часы»
- Блок «Константа»
- Блок «Ступенька»
- Блок «Синусоида»
- Блок «Экспонента»
- Блок «Равномерный шум»
- Блок «Кусочно-линейная»
- Блок «Меандр»
- Блок «Циклограмма»
- Блок «Управляемый синусоидальный генератор»
- Блок «Сумматор»
- Блок «Сравнивающее устройство»
- Блок «Сложение вектора с числом»
- Блок «Суммирование элементов вектора»
- Блок «Перемножитель»
- Блок «Перемножение элементов вектора»
- Блок «Умножение на число»
- Блок «Делитель»
- Блок «Деление скаляра на вектор»
- Блок «Усилитель»
- Блок «Векторный усилитель»
- Блок «Абсолютное значение»
- Блок «Размножитель»
- Блок «Знак»
- Блок «Номер активного элемента»
- Блок «Динамическая выборка»
- Блок «Стоп-расчет»



- Блок «Линейный преобразователь»
- Блок «Выборка по активному элементу»
- Блок «Целая часть»
- Блок «Дробная часть»
- Блок «Округление»
- Блок «Мультиплексор»
- Блок «Демультиплексор»
- Блок «Распаковка матрицы»
- Блок «Выборка из вектора»
- Блок «Линейная функция»
- Блок «Параболическая функция»
- Блок «Синусоидальная функция»
- Блок «Экспоненциальная функция»
- Блок «Гиперболическая функция»
- Блок «Арсинус»
- Блок «Арккосинус»
- Блок «Арктангенс»
- Блок «Гиперболический синус»
- Блок «Гиперболический косинус»
- Блок «Гиперболический тангенс»
- Блок «Степенная функция»
- Блок «Логарифм натуральный»
- Блок «Логарифм десятичный»
- Блок «Логарифм натуральный с защитой 0»
- Блок «Логарифм десятичный с защитой 0»
- Блок «Корень квадратный»
- Блок «Синус с косинусом»
- Блок «Язык программирования»
- Блок «Интегратор»



- Блок «Инерционное звено 1-го порядка»
- Блок «Идеальное транспортное запаздывание»
- Блок «Инерционно-дифференцирующее звено»
- Блок «Производная»
- Блок «Инерционно-интегрирующее звено»
- Блок «Интегратор с ограничением»
- Блок «Интегратор с изменяемыми начальными условиями»
- Блок «Интегратор с ограничением и сбросом начальных условий»
- Блок «Круговой интегратор»
- Блок «Аналитическая апериодика 1-го порядка»
- Блок «Дискретная апериодика 1-го порядка»
- Блок «Интегратор на усилителях»
- Блок «ДИФ-производная»
- Блок «Фильтрация сигнала»
- Блок «Линейное с насыщением»
- Блок «Линейное с зоной нечувствительности»
- Блок «Релейное неоднозначное (гистерезис)»
- Блок «Релейное с зоной нечувствительности»
- Блок «Минимум по входам»
- Блок «Минимум по всем элементам вектора»
- Блок «Излом»
- Блок «Ломаная статическая характеристика»
- Блок «Многомерная интерполяция»
- Блок «Зазор»
- Блок «Люфт»
- Блок «Квантователь по уровню»
- Блок «Запоминание значений сигналов»
- Блок «Запоминание длительности истинного сигнала»
- Блок «Задержка на шаг интегрирования»



- Блок «Субмодель релейное с переменными ЗН и ЗВ»
- Блок «Субмодель релейное плюс с переменными ЗН и ЗВ»
- Блок «Субмодель релейное минус с переменными ЗН и ЗВ»
- Блок «Сигнал больше уставки»
- Блок «Сигнал меньше уставки»
- Блок «ПОР ИКР (порог срабатывания)
- Блок «НОР ИКП»
- Блок «Гистерезис входного сигнала»
- Блок «Мертвая зона (изменяемая)»
- Блок «ФЗН (формирование зоны нечувствительности)»
- Блок «Импульсное открытие/закрытие»
- Блок «Запоминание сигнала»
- Блок «Линейное с переменным ограничением»
- Блок «Субмодель»
- Блок «Порт входа»
- Блок «Порт выхода»
- Блок «В память»
- Блок «Из памяти»
- Блок «Двунаправленная шина (вход)»
- Блок «Двунаправленная шина (выход)»
- Блок «Уровень неподсоединённых портов»
- Блок «Блок записи свойств»
- Блок «Указатель сортировки»
- Блок «Условие выполнения субмодели»
- Блок «Внешняя DLL»
- Блок «Самогенерирующий блок»
- Блок «Чтение из списка сигналов»
- Блок «Запись в список сигналов»
- Блок «Чтение сигналов»



- Блок «Входной контакт»
- Блок «Выходной контакт»
- Блок «Датчик»
- Блок «Выход алгоритма (векторный)»
- Блок «Чтение сигналов (векторный)»
- Блок «Клиент OPC»
- Блок «Ключ ручное перекидной»
- Блок «Ключ-0»
- Блок «Ключ-1»
- Блок «Ключ-2»
- Блок «Ключ-3»
- Блок «Ключ интегратора»
- Блок «Ключ управляемый нормально разомкнутый тип 1»
- Блок «Ключ управляемый нормально разомкнутый тип 2»
- Блок «Ключ управляемый нормально замкнутый тип 1»
- Блок «Ключ управляемый нормально замкнутый тип 2»
- Блок «Ключ управляемый нормально разомкнутый тип 3»
- Блок «Ключ управляемый нормально разомкнутый тип 4»
- Блок «Ключ управляемый нормально замкнутый тип 3»
- Блок «Ключ управляемый нормально замкнутый тип 4»
- Блок «Ключ управляемый перекидной по входам тип 1»
- Блок «Коммутатор входных сигналов»
- Блок «Выделение минимума с формированием признака»
- Блок «Выделение максимума с формированием признака»
- Блок «Задатчик с варьируемым значением»
- Блок «Оператор И»
- Блок «Оператор ИЛИ»
- Блок «Оператор НЕ»
- Блок «Логические операции»





- Блок «Операция БОЛЬШЕ»
- Блок «Операция МЕНЬШЕ»
- Блок «Операция РАВНО»
- Блок «Операция НЕ РАВНО»
- Блок «БОЛЬШЕ ИЛИ РАВНО»
- Блок «МЕНЬШЕ ИЛИ РАВНО»
- Блок «XOR»
- Блок «NOT XOR»
- Блок «М из N»
- Блок «М из N поэлементное»
- Блок «Один из многих»
- Блок «Один из многих по выбору»
- Блок «Счётчик»
- Блок «Векторное И»
- Блок «Векторное ИЛИ»
- Блок «Логическое И с пассивным входом»
- Блок «RS-триггер с приоритетом по сбросу»
- Блок «RS-триггер с приоритетом по установке»
- Блок «Триггер T»
- Блок «Триггер TR»
- Блок «Триггер TS»
- Блок «Определение первого события»
- Блок «Задержка по включению»
- Блок «Задержка по включению и выключению»
- Блок «Задержка по выключению»
- Блок «Импульс по фронту»
- Блок «Импульс по срезу»
- Блок «Импульс по фронту или срезу»
- Блок «Импульс»



- Блок «Импульс длительностью не более заданной»
- Блок «Импульс с пролонгированием»
- Блок «Временное подтверждение»
- Блок «Одновибратор»
- Блок «Обмотка реле»
- Блок «Двухпозиционное реле (SET)»
- Блок «Двухпозиционное реле (RESET)»
- Блок «Замыкающий контакт реле»
- Блок «Размыкающий контакт реле»
- Блок «Переключающий контакт реле»
- Блок «Замыкающий контакт реле с задержкой по замыканию»
- Блок «Замыкающий контакт реле с задержкой по размыканию»
- Блок «Размыкающий контакт реле с задержкой по замыканию»
- Блок «Замыкающий контакт реле с задержкой по размыканию»
- Блок «Внешний контакт реле»
- Блок «Двухпозиционное реле (RS)»
- Блок «Внешний размыкающий контакт реле»
- Блок «Экстраполятор»
- Блок «Среднее арифметическое»
- Блок «Среднеквадратичное отклонение»
- Блок «Буфер последовательного ввода данных»
- Блок «Буфер последовательного вывода данных»
- Блок «Быстрое преобразование Фурье»
- Блок «Обратное быстрое преобразование Фурье»



## Приложение 2. Список функций и операторов языка программирования для генерации кода

- Оператор **+** «Сложение двух переменных»
- Оператор **-** «Вычитание одной переменной из другой»
- Оператор **\*** «Умножение одной переменной на другую»
- Оператор **/** «Деление первого выражения на второе»
- Оператор **^** «Возведение первого выражения в степень второго»
- Оператор **.\*** «Поэлементное перемножение действительных или комплексных матриц»
- Оператор **./** «Поэлементное деление действительных или комплексных матриц»
- Оператор **.^** «Поэлементное возведение в степень двух матриц или матрицы в действительную степень»
- Оператор **>** «Один элемент больше другого»
- Оператор **<** «Один элемент меньше другого»
- Оператор **>=** «Один элемент больше или равен другому»
- Оператор **<=** «Один элемент меньше или равен другому»
- Оператор **<>** «Элементы не равны друг другу»
- Оператор **not** «Оператор целочисленного или логического отрицания»
- Оператор **and** «Оператор побитового логического И»
- Оператор **or** «Оператор побитового логического ИЛИ»
- Оператор **xor** «Оператор побитового логического исключающего ИЛИ»
- Оператор **shl** «Оператор побитового сдвига влево»
- Оператор **shr** «Оператор побитового сдвига вправо»
- Оператор **mod** «Оператор получения остатка от целочисленного деления»
- Оператор **div** «Оператор целочисленного деления»
- Оператор **#** «Оператор размножения»
- Оператор **..** «Оператор вычисления интервала целых чисел»
- Оператор **[]** «Операторы доступа к элементу массива и упаковки элементов в вектор»
- Функция **abs** «Функция получения модуля вещественного или комплексного числа или вектора»



- Функция **sin** «Функция вычисления синуса вещественного или комплексного числа»
- Функция **cos** «Функция вычисления косинуса вещественного или комплексного числа»
- Функция **ch** «Функция вычисления гиперболического косинуса вещественного или комплексного числа»
- Функция **sh** «Функция вычисления гиперболического синуса вещественного или комплексного числа»
- Функция **th** «Функция вычисления гиперболического тангенса вещественного или комплексного числа»
- Функция **exp** «Функция вычисления экспоненты вещественного или комплексного числа»
- Функция **sqrt** «Функция вычисления корня квадратного вещественного или комплексного числа»
- Функция **tg** «Функция вычисления тангенса вещественного или комплексного числа»
- Функция **arctg** «Функция вычисления арктангенса вещественного или комплексного числа»
- Функция **arcsin** «Функция вычисления арксинуса вещественного или комплексного числа»
- Функция **arccos** «Функция вычисления арккосинуса вещественного или комплексного числа»
- Функция **ln** «Функция вычисления натурального логарифма вещественного или комплексного числа»
- Функция **lg** «Функция вычисления десятичного логарифма вещественного или комплексного числа»
- Функция **complex** «Функция получения комплексного числа»
- Функция **arg** «Функция получения аргумента комплексного числа»
- Функция **real** «Функция получения вещественной части комплексного числа»
- Функция **case** «Выбор аргумента по номеру»
- Функция **imag** «Функция получения мнимой части комплексного числа»
- Функция **atan2** «Функция вычисления арктангенса точки с учетом квадранта»



- Функция **min** «Функция вычисления минимального значения из двух значений или из значений вектора»
- Функция **max** «Функция вычисления максимального значения из двух значений или из значений вектора»
- Функция **sign** «Функция определяет знак числа»
- Функция **extract** «Функция извлечения элементов из матрицы или вектора по номерам»
- Функция **cols** «Функция возвращает количество столбцов матрицы или количество элементов массива»
- Функция **transp** «Функция транспонирования матрицы или вектора»
- Функция **lsolve** «Функция решения системы линейных алгебраических уравнений (СЛАУ)»
- Функция **det** «Функция вычисления детерминанта матрицы»
- Функция **polyroot** «Функция вычисления корней характеристического полинома вещественной матрицы»
- Функция **Interpol** «Функция двумерной линейной, трехмерной линейной, кубической сплайн - интерполяции аргумента x»
- Функция **vector** «Функция создания нулевого вектора»