**Работа с инструментом генераци кода Си  
в среде динамического моделирования  
SimInTech**

ООО «ЗВ Сервис», 2016

# Введение

Среда динамического моделирования технических систем «SimInTech» является мощным инструментом для разработки алгоритмов управления различной степени сложности. Одним из важных преимуществ среды является поддержка концепции сквозного проектирования алгоритмов. Реализация данной концепции в SimInTech предусматривает, что сначала на базе идей технолога создается функциональная блок-схема алгоритма из типовых блоков, содержащихся в библиотеке (причем блоки могут быть как базовыми, т.е. изначально содержащимися в библиотеке, так и пользовательскими, т.е. созданными пользователем на основе базовых блоков и специального блока «Субмодель»), затем схема может быть проверена и, при необходимости, отлажена при совместном моделировании с моделью объекта управления, а после автоматически преобразован при помощи встроенных инструментов среды в исходный код для целевой системы, который после компиляции может быть использован в программируемом контроллере. Получаемый исходный код при этом функционально идентичен с исходной блок-схемой алгоритма управления или регулирования.

# Интерфейс работы с инструментом генерации кода

Рассмотрим инструмент среды SimInTech, предназначенный для генерации кода. Для того, чтобы его вызвать, необходимо в Главном Окне программы войти в меню «**Кодогенератор**».

|  |  |
| --- | --- |
| Рисунок 1. Выбор кодогенератора | В результате перед пользователем возникнет выпадающее меню из двух строк: «**Кодогенератор Си**» и «**Кодогенератор ST**» (см. Рисунок 1).  Для последующей работы мы будем использовать «**Кодогенератор Си**». Руководство пользователя по кодогенератору ST предоставляется по отдельному запросу. Основные принципы работы остаются такими же как и для кодогенератора на язык Си. |

При активации пункта меню «Кодогенератор Си» (осуществляется одиночным кликом левой кнопки мыши по соответствующей строке) кодогенератор становится активным (выбранным в ПО SimInTech как текущий генератор кода) и появляется диалоговое окно кодогенератора (Рисунок 2).

|  |  |
| --- | --- |
| Рисунок 2. Окно кодогенератора | Рисунок 3. Закладки окна кодогенератора |

В нем можно увидеть закладки, отвечающие за различные функции и настройки кодогенератора (Рисунок 3), а именно: **Загрузка**; **Настройки**; **Отладчик**; **Дополнительно**; **Список конфигураций**. Рассмотрим каждую из них.

## Закладка «Загрузка»

Закладка «**Загрузка**» отвечает за формирование списка задач (перечня проектов алгоритмов), предназначенных для генерации кода и загружаемых в исполнительную систему. Можно выделить три области, формируемые при активации закладки «**Загрузка**» (Рисунок 4):

|  |  |
| --- | --- |
| Рисунок 4. Области закладки «Загрузки» | * 1 область – **Панель кнопок** – содержит кнопки основных действий для формирования и обработки списка задач; * 2 область – **Таблица проектов** – представляет собой таблицу, которая содержит список проектов, которые будут загружаться в исполняемую среду реального времени; * 3 область – **Окно сообщений** – формирует информационные сообщения в процессе работы кодогенератора.   Рассмотрим подробнее каждую из этих областей. При дальнейшем знакомстве с инструментом «**Кодогенератор**» мы так же будем придерживаться принципа подробного рассмотрения его окон, полей и пр.  **Панель кнопок** содержит следующие кнопки: |

* «**Новая конфигурация**» - создает пустую таблицу проектов (пустую новую загрузку);
* «**Загрузить конфигурацию**» - загружает ранее сохраненную пользователем таблицу проектов;
* «**Сохранить конфигурацию**» - сохраняет текущую таблицу проектов. В случае, если текущая таблица является вновь созданной, то при сохранении перед пользователем возникнет диалоговое окно, в котором он сможет указать имя для файла таблицы сохраняемой конфигурации и место для его сохранения. В случае, если текущая таблица уже была сохранена, то нажатие на кнопку приведет к перезаписи файла, в который она была сохранена. Также пользователю доступна кнопка «**Сохранить конфигурцию как**», которая активируется при нажатии на символ стрелки, находящийся справа от иконки «**Сохранить конфигурацию**». Используя «**Сохранить конфигурацию как**», пользователь может сохранить текущую конфигурацию либо с новым именем, либо используя имена ранее сохраненных конфигураций с заменой содержимого данных конфигураций (при этом пользователю так же остается доступным сохранение конфигурации и с текущим именем). В результате сохранения конфигурации появляется файл конфигурации с расширением **.alt**, который по другому еще называется **alt-схемой**;
* «**Открыть выделенные**» - открывает (загружает) в ПО SimInTech файлы проектов, которые выделены пользователем в таблице;
* «**Добавить файлы**» - открывает диалоговое окно для выбора и загрузки файла(-ов) проекта (файла(-ов) с расширением **.prt**) в таблицу;
* «**Удалить файлы**» - удаляет выбранные пользователем проекты из таблицы проектов (при этом на диске сами проекты остаются);
* «**Сдвинуть вверх**» - перемещает выбранный пользователем проект вверх в таблице проектов;
* «**Сдвинуть вниз**» - перемещает выбранный пользователем проект вниз в таблице проектов;
* «**Пересобрать модули и конфигурацию**» - полностью собирает заново расчетный модуль, т.е. происходит генерация исходного кода для каждого проекта конфигурации, кросс-компиляция, формирование конфигурационного файла для целевой системы;
* «**Пересобрать только конфигурацию загрузки**» - собирает только конфирурацию загрузки (конфигурационные файлы для целевой системы), исключая пересборку и компиляцию расчетных модулей;
* «**Скопировать на целевую систему**» - производит автоматическое копирование на удаленный компьютер всех необходимых для работы программы на нём файлов;
* «**Очистить целевую систему**» - производит удаление файлов программы с удаленного компьютера.

Таблица проектов содержит следующие столбцы:

* **Файл схемы** – файл проекта, содержащий схему алгоритма, который предназначен для преобразования в исходный код программы для работы на целевой системе;
* **Имя алгоритма** – имя алгоритма, которое заполняется в случае если какой-либо файл расчетного модуля будет использоваться в приборе для обработки одинаковых по структуре алгоритмов, подключаясь при этом к разным внешним переменным для каждого из запущенных процессов;
* **Период, мс** – период вызова расчетного модуля в миллисекундах;
* **К-во вызовов** – количество вызовов одного расчетного модуля за 1 период вызова;
* **Образ (exe)** – имя исполняемого файла расчетного модуля.

## Закладка «Настройки»

Закладка «**Настройки**» отвечает за настройки кодогенератора (см. Рисунок 5).

Рассмотрим входящие в закладку «**Настройки**» поля и выпадающие списки. Для текущей работы мы не будем подробно останавливаться на всех полях и выпадающих списках данной закладки, а рассмотрим лишь основные, которые нам нужны для текущей работы.

|  |  |
| --- | --- |
| Рисунок 5. Закладка «Настройки» | В поле «**Имя конфигурации загрузки**» указывается имя файла конфигурации загрузки. Обратите внимание, что имя указывается без расширения **.alt**. По умолчанию имя конфигурации загрузки задано default.  В поле «**Директория исходиков**» прописывается путь сохранения сгенерированных программ. Рядом с полем расположена иконка, при нажатии на которую возникает окно, с помощью которого можнов выбирать путь сохранения можно с помощью стандартного инструментария Windows.  С помощью выпадающего списка «**Директория шаблона кода**» осуществляется выбор и настройка генерации программ в соответствии с типом целевой операционной системы, т.е. пользователь выбирает тот шаблон генерации кода, который соответствует целевой операционной системе и архитектуре. В соответствии с выбором пользователя инструментом «**Кодогенератор**» будет автоматически выбран шаблон, по которому будет генерироваться код, учитывающий особенности операционной системы, под которой он будет исполняться. Шаблоны генерации кода находятся в директории **C:\SimInTech\bin\CodeTemplates**.  Если в выпадающем списке нет необходимой пользователю системы, то он может самостоятельно, воспользовавшись иконкой выбора пути, указать путь к директории с необходимыми шаблонами. |

Выпадающий список «**Имя блока в коде**» определяет правило формирования имён переменных в программном коде для блоков, из которых собрана расчетная схема.

| Значение параметра | Примечание |
| --- | --- |
| Номер при сортировке | В качестве базы для имени внутренних переменных в генерируемом коде Си используются сквозные номера, присвоенные средой SimInTech при сортировке блоков расчетной схемы;  При дополнении или изменении расчетной схемы внутренним переменным будут присвоены новые номера в соответствии с новой сортировкой, происходящей при инициализации схемы и/или генерации кода. |
| Ограниченный md5-хэш (6 символов) | В качестве базы для имени внутренней переменной используются первые 5 символов от md5-хэш функции полного имени блока. |
| Транслитерированное имя | В качестве базы для имени внутренней переменной используется текстовое имя в формате <имя субмодели>\_<имя блока>. |
| Полный md5-хэш (32 символа) | В качестве базы для имени внутренней переменной используются первые 32 символа от md5-хэш функции полного имени блока. |
| Ограниченный md5-хэш (12 символов) | В качестве базы для имени внутренней переменной используются первые 12 символов от md5-хэш функции полного имени блока. |

Выпадающий список «**Стиль заголовка кода**» определяет способ генерации заголовочного файла на языке программирования Си, с помощью которого происходит автоматическая привязка текстовых идентификаторов к нужным адресам в памяти. Существует несколько способов генерации, которые перечислены в таблице ниже.

| Значение параметра | Примечание |
| --- | --- |
| С внутренними таблицами переменных (DLL) | В заголовочный файл, на основе которого генерируется динамически загружаемая библиотека или программа, встраиваются служебные таблицы с указанием имён переменных, соответствующим им адресом в памяти и типом данных (целый, вещественный и т.п.); |
| С внешними таблицами переменных (EXE) | В обрабатываемый компилятором заголовочный файл не пишутся служебные таблицы с именами переменных. Служебные таблицы генерируются в бинарные файлы, структура которых подробно описана в руководстве системного программиста |
| Автоопределение по ini-файлу в шаблоне кода | Опция позволяет автоматически выбирать тип генерации заголовочного файла: с внутренними или внешними таблицами переменных в соответствии с выбранным шаблоном кода.  При этом в директории шаблона кода должен присутствовать ini-файл, в котором в текстовом виде указывается идентификатор способа определения генерации заголовочного файла. |

Поле «**Команда сборки модуля**» содержит текст вызова скрипт-файла (командного файла, т.е. bat-файла), который производит автоматическую компиляцию расчетного модуля по сгенерированным кодогенератором исходным кодам.

Поле «**Команда загрузки в целевую систему**» содержит строку вызова скрипт-файла, который производит загрузку скомпилированных расчетных модулей (программ) и других необходимых файлов на целевую систему.

Поле «**Префикс имён переменных**» отвечает за символ, дописываемый в начало имени переменных Си кода.

## Закладка «Отладчик»

Закладка отвечает за настройки отладчика и управление расчетным процессом в приборе (см.Рисунок 6). Рассмотрим подробнее выпадающие списки и поля, находящиеся на данной закладке.

Список «**Режим отладки**» отвечает за выбор способа расчета схемы. Для пользователя доступны три способа:

|  |  |
| --- | --- |
| Рисунок 6. Закладка «Отладчик» | * «**Локальная**» - расчет схемы производится на компьютере пользователя без доступа к прибору; * «**Удаленная**» - расчет схемы производится на приборе. При этом на расчетной схеме, с которой работает пользователь на коспьютере, отображаются параметры блоков, которые насчитаны на приборе. В данном режиме пользователь на схеме может изменять свойства блоков, при этом будет осуществляться автоматическое изменение свойств соответствуюющих блоков в приборе; * «**Получать только входы**» - в данном режиме расчетная схема, открытая пользователем на компьютере, получает только значения переменных входа/выхода. Дальнейший расчет переменных осуществляется локально на компьютере, и синхронизации значений переменных, полученных в процессе расчета на компьютере и на приборе, не происходит. |

Выпадающий список «**Режим изменения параметров**» отвечает за выбор режима синхронизации изменяемых пользователем свойств и содержит два способа:

* «**Однократный**» - при изменении пользователем свойства какого-либо блока находящегося на расчетной схеме происходит его однократное изменение с однократной передачей на прибор;
* «**Списочный**» - средой SimInTech автоматически составляется список свойств блоков, передающихся на каждом расчетном шаге в прибор.

В поле «**Параметры соединения (хост:порт)**» указывается адрес целевой системы и через двоеточие порт подключения (в случае, если порт не указан, то подключение производится к порту по умолчанию с номером 22375). При этом если активировать пункт «**Автопересоединение**», то при потере связи среды SimInTech с удаленным компьютером периодически будут осуществляться попытки автоматического подсоединения до установления связи.

В поле «**Имя DLL удаленного доступа**» прописывается имя динамически загружаемой библиотеки, реализующей алгоритм подключения и сетевого взаимодействия (получения, отправки данных) с сервером отладки GdbServer.

В поле «**Имя объекта внутри DLL**» прописывается имя модуля, которое обеспечивает связь программы на удаленном компьютере, с сервером отладки. Используется при необходимости создания поддержки собственного протокола передачи данных, отличающегося от заложенного в динамическую библиотеку avrordbg.dll.

Удаленное управление программой, находящейся на приборе, осуществляется при помощи кнопок, находящихся в области «**Управление прибором**».

Назначение данных кнопок следущее:

* «**Удаленный запуск сервера обмена**» - осуществляет запуск сервера отладки GdbServer на целевой системе;
* «**Подключиться к прибору**» - осуществляет соединение среды SimInTech, установленной на компьютере пользователя, с сервером отладки целевой системы;
* «**Отключить**» - осуществляет отключение среды SimInTech, установленной на компбьютере пользователя, от сервера отладки целевой системы без завершения расчетного процесса;
* «**Запустить конфигурацию**» - осуществляет запуск на целевой системе конфигурации, имя которой указано в поле «**Имя конфигурации загрузки**» закладки «**Настройки**»;
* «**Пауза**» - приостанавливает процесс расчета программы, находящейся на целевой системе;
* «**Продолжить**» - переводит процесс расчета программы из режима паузы в режим расчета;
* «**Стоп**» - останавливает процесс расчета программы на целевой системе (в т.ч. включая завершение расчета дочерних процессов);
* «**Сохранить состояние**» - сохраняет снимок текущего состояния переменных в расчетных модулях программы в файл, имя которого указывается в поле «**Имя состояния**»;
* «**Загрузить состояние**» - загружает снимок ранее сохраненного состояния с именем, указанным в поле «**Имя состояния**».

## Закладка «Дополнительно»

Закладка «**Дополнительно**» (см. Рисунок 7) содержит окно ввода настроек сетевой подсистемы для формирования файла конфигурации стеевого обмена. Подробно формат конфигурации и ключевые секции и слова – см. руководство пользователя модуля сетевого обмена.

|  |  |
| --- | --- |
| Рисунок 7. Закладка «Дополнительно» | Рисунок 8. Закладка «Список конфигураций» |

## Закладка «Список конфигураций»

Закладка «**Список конфигураций**» отвечает за работу сразу с несколькими файлами конфигурации (**alt-схемами**). В случае, если пользователю необходимо сгенерировать и загрузить программы для нескольких целевых систем одновременно, то можно довольно просто это осуществить воспользовавшись данной закладкой.

Закладка «**Список конфигураций**» аналогично закладке «**Загрузка**» содержит три области: **панель кнопок**, **область конфигураций** и **окно сообщений**.

**Панель кнопок** содержит следующие кнопки:

* «**Очистить список конфигураций**» - создает пустой список конфигураций;
* «**Загрузить список конфигураций**» - загружает ранее сохраненный пользователем список конфигураций;
* «**Сохранить список конфигураций**» - сохраняет текущий список конфигураций. Сохранение списка конфигураций происходит аналогично сохранению конфигурации. В результате сохранения появляется файл с расширением **.confs**.
* «**Добавить файлы**» - открывает диалоговое окно для выбора и загрузки файла конфигурации (файла с расширением **.alt**) в список конфигураций;
* «**Удалить файлы**» - удаляет выбранные пользователем конфигурации из списка;
* «**Открыть выделенную конфигурацию**» - открывает проекты, входящие в выбранную пользователем конфигурацию;
* «**Собрать модули на всех**» - генерирует расчетные модули для всех конфигураций, внесенных в список;
* «**Скопировать на целевую систему для всех**» - загружает программы, полученные в результате генерации, в приборы, в которых они должны использоваться.

## Типы файлов среды SimInTech, с которыми работает инструмент кодогенератора

Работа с кодогенератором предполагает работу с файлами, имеющими следующие расширения:

* **.prt** – основной тип файла, создаваемый в процессе разработки пользователем расчетных схем в среде SimInTech;
* **.alt** – тип файла, создаваемый при сохранении таблицы (конфигурации) проектов (, из которых будет собран расчетный модуль;
* **.confs** – тип файла, создаваемый при сохранении списка таблиц (конфигураций) проектов.

# Основы работы с кодогенератором

Для получения навыков работы с кодогенератором проделаем следующее пошаговое упражнение, демонстрирующее основные этапы и действия, связанные с работой инструментом кодогенерации SimInTech.

* + 1. Приступим к работе с кодогенератором. В данной работе мы будем использовать ранее созданные в качестве примеров расчетные схемы, которые находятся в директории **C:\SimInTech\Demo\Automatic**.

Будем работать с расчетной схемой регулятора, которая находится в директории с именем **Регулятор**.

Для того, чтобы не «испортить» пример, находящийся в **Demo**, создадим в **C:\SimInTech\Projects** собственную директорию **Регулятор**, в которую скопируем содержимое директории из **C:\SimInTech\Demo\Automatic\Регулятор**:

* ШРП - штатный регулятор питания БС.mgr
* ШРП - штатный регулятор питания БС.prt
* Директория «блоки регулятора».
  + 1. В директории **C:\SimInTech\Projects\Регулятор** создадим еще одну директорию и назовем ее **src**: исходные файлы программы, которые будут получены в результате работы кодогенератора, будут сохраняться в данной директории.

Внешний вид директории **..\Регулятор** после выполнения действий 1 и 2, представлен на рисунке ниже.

Рисунок 9. Содержимое директории Регулятор

* + 1. В главном окне SimInTech заходим в меню **Кодогенератор** и активируем пункт «**Кодогенератор Си**». Открывается окно кодогенератора с чистой конфигурацией. Сохраняем данную конфигурацию с именем **shrp** (в каталог, куда только что скопировали проект регулятора). При этом в шапке окна кодогенератора прописывается полный путь к файлу концигурации, с которым сейчас работает кодогенератор (см. Рисунок 10).

Рисунок 10. Окно кодогенератора с загруженным файлом конфигурации

* + 1. Добавим файл проекта в созданную нами конфигурацию. Для этого воспользуемся кнопкой «**Добавить файлы**» и в появившемся диалоговом окне добавления проекта выберем **ШРП – штатный регулятор питания БС.prt**, а затем нажмем кнопку «**Открыть**». Файл проекта будет добавлен в файл конфигурации и появится в таблице проектов (см. Рисунок 11). По умолчанию каждому добавляемому файлу проекта проставляются следующие параметры расчета:
* Период, мс – 60;
* К-во вызовов – 1.

Таким образом, если для добавляемого файла оставить параметры, добавляемые по умолчанию, то в сгенерированной программе он будет вызываться 1 раз каждые 60 мс.

Рисунок 11. Файл конфигурации с добавленным проектом

* + 1. Для просмотра и правки параметров и схемы добавленного файла воспользуемся кнопкой быстрого доступа «**Открыть выделенные**». Выделим **проект ШРП – штатный регулятор питания БС.prt** в таблице проектов и нажмем на кнопку «**Открыть выделенные**» , после чего откроется окно проекта со схемой регулятора, представленное ниже (см. Рисунок 12).

Рисунок 12. Окно проекта с расчетной схемой

* + 1. В открывшемся окне зайдем в **Параметры расчета** и укажем в свойстве «**Имя (имена) алгоритма**» имя **a3pr1** (см. Рисунок 13). Сохраним внесенные изменения нажатием на кнопку «**Ok**».

Рисунок 13. Параметры расчета схемы регулятора с выделенным свойством «Имя (имена) алгоритма»

Сохраним проект регулятора, нажав на кнопку «**Сохранить проект**» на панели кнопок или через меню «**Файл**», выбрав пункт «**Сохранить проект**», в главном окне программы SimInTech.

* + 1. Вернемся в окно кодогенератора. Перейдем на вкладку «**Настройки**». Сделаем изменения следующих настроек (см. Рисунок 14):
* **Директория исходников** - ./**src/**
* **Директория шаблона кода - %codetemplates%NordWind\_qnx**, где выражение **%codetemplates%** определяет абсолютный путь к директории шаблонов для кодогенератора, а **NordWind\_qnx** определяет каталог и одновременно целевую платформу, под которую будет создаваться исходный код.

Рисунок 14. Изменения на вкладке «Настройки»

* + 1. Вернемся на вкладку «**Загрузка**». Нажмем на кнопку «**Пересобрать модули и конфигурацию**» . В окне сообщений кодогенератора появится запись «**Генерация исходников завершена успешно**» (см. Рисунок 15).

Рисунок 15. Сообщение «Генерация исходников завершена успешно»

* + 1. Зайдем в каталог src. После завершения генерации в ней появились файлы исходных кодов, а также бинарные файлы целевой программы (при наличии необходимого компилятора или кросс-компилятора, см. Рисунок 16).

Рисунок 16. Каталог с файлами, получившимися после работы генератора кода

* + 1. Продолжим работу на целевом компьютере. Зайдем в директорию с исполняемой средой и запустим сервер отладки:

**cd /home/NordWind/**

**./GdbSever**

Таким образом мы запустим GdbServer, используемый удалённо для запуска исполняемой среды, удаленной отладки и отображения результатов работы алгоритма на инструментальном компьютере.

* + 1. Копируем через протокол ftp в директорию**/home/NordWind/** следующие файлы:
* **a3pr1.so** – файл с алгоритмом регулятора, транслированный в динамическую библиотеку;
* **a3pr1.intvars.table** – файл с локальными переменными для обеспечения возможности удаленной отладки алгоритма;
* **a3pr1.extvars.table** – файл, содержащий внешние сигналы (переменные), с которыми работает алгоритм;
* **default.conf** – конфигурация загрузки, содержащая информацию об имени алгоритма, периоде и количестве вызовов (т.е. о тех свойствах, которые были заданы проекту в таблице проектов на вкладке «**Загрузка**»)

Отметим следующее: при формировании файлов **.so**, **.intvars**, **.extvars** участвует имя алгоритма, заданное в параметрах расчета проекта (см. пункт 6), а при формировании файла **.conf** используется имя конфигурации загрузки, задаваемое на вкладке «**Настройки**» в кодогенераторе.

* + 1. Вернемся к окну кодогенератора и перейдем на вкладку «**Отладчик**». Пропишем на данной вкладке в поле «**Параметры соединения (хост:порт)**» следующее выражение:

**<IP-адрес устройства>:22375;timeout=10000**, где IP-адрес устройства – это IP-адрес целевого компьютера, который можно узнать с помощью команды **ipconfig**, или аналогичной, набранной в командной строке.

Пример заполнения окна – смотрите рисунок:

Рисунок 17. Параметры расчета схемы регулятора с выделенным свойством «Имя (имена) алгоритма»

* + 1. После заполнения поля «**Параметры соединения (хост:порт)**» переходим к области «**Управление прибором**», где нажимаем на кнопку «**Подключиться к прибору**» . При нажатии на кнопку происходит соединение с целевым компьютером, и панель кнопок управления расчетом на приборе становится активной.

|  |  |
| --- | --- |
| Рисунок 18. Активация режима «Удаленный» в главном меню | * + 1. Перейдем в SimInTech. В меню «**Режим отладки**» устанавливаем режим «**Удаленный**» (Рисунок 18).     2. Возвратимся в окно кодогенератора. На вкладке «**Отладчик**» в области «**Управление устройством**» нажимаем кнопку «**Запустить конфигурацию**»: происходит запуск на целевой системе исполняемой среды (уже загруженной на него) с заданной конфигурацией. |

* + 1. Активируем окно с алгоритмом регулятора. Затем в главном окне среды SimInTech при нажимаем на кнопку «**Расчёт**», расположенную на панели кнопок. При этом, т.к. режим работы выставлен в **Удалённый**, происходит не локальное моделирование схемы, а подключение к серверу удалённой отладки на целевой системе. На схеме регулятора начинает идти модельное внутреннее время (время будет считываться с целевой системы и начинаться уже не с нуля!), прошедшее с момента запуска алгоритма. Смотрим расчет схемы.
    2. Инструмент «**Кодогенератор**» позволяет пользователю сохранять состояние расчетного процесса. Используя вкладку «**Отладчик**» можно сделать снимок состояния процесса в какой-либо момент времени от начала его расчета. Для этого необходимо в поле «**Имя состояния**» ввести какое-либо имя, а затем воспользоваться кнопками сохранения и загрузки состояния. Для примера, введем **State1**, а затем нажмем кнопку «**Сохранить состояние**».
    3. Продолжим расчет. Изменим ход процесса, изменив коэффициенты регуляторов, входящих в расчетную схему. Для того, чтобы вернуться к какому-либо исходному, ранее сохраненному, состоянию, необходимо так же в поле «**Имя состояния**» вести имя ранее сохраненного снимка состояния, а затем воспользоваться кнопкой загрузки. Так как ранее мы сохраняли состояние с именем **State1**, то загрузим данное состояние, введя его имя в поле, а затем нажав на кнопку «**Загрузить состояние**».

# Совместная работа алгоритма управления, работающего на удаленном компьютере, с локальной моделью, работающей на компьютере пользователя.

В среде динамического моделирования SimInTech возможно производить отладку алгоритма на целевом компьютере, используя расчетную модель какого-либо оборудования, с которым должен работать данный контроллер. Например, если есть задача подготовки алгоритма управления какой-либо теплогидравлической системой с помощью контроллера, то решить ее в среде SimInTech можно следующим образом (схема решения упрощенно выглядит так):

* Подготавливаем **алгоритмы** управления, отлаживаем их, производим генерацию исходных кодов, транслируем их на контроллер;
* Разрабатываем **модель** теплогидравлической системы, используя соответствующие теплогидравлические коды, отлаживаем ее;
* Обеспечиваем **совместную работу** теплогидравлической модели и программы на контроллере.

Такой подход дает следующие преимущества: еще до натурных испытаний на объекте уже можно с высокой долей вероятности получить представление обо всех особенностях работы алгоритма, исправить алгоритмические ошибки, понять, как исполнение алгоритма влияет на технологическую схему (т.е. проходит ли включение/отключение различного оборудования гладко или образует всплески параметров, остаются ли технологические параметры в различных режимах в заданных границах или переходят их и т.п.), оценить степень этого влияния, проанализировать устойчивость регуляторов и т.д.

Рассмотрим пример совместной работы теплогидравлической модели и алгоритмов, загруженных в контроллер, созданный на основе проекта системы KBA. Данный проект находится в директории **C:\SimInTech\Demo\TPP\KBA** и представляет собой набор файлов с общей базой данных сигналов, работающих в пакете проектов **kba.pak**. Каждый файл отвечает за определенную область моделирования:

* KBA.prt – содержит расчетную схему теплогидравлической модели системы KBA;
* Алгоритмы KBA.prt – содержит схемы алгоритмов, осуществляющих управление задвижками, клапанами, насосами и другими механизмами, установленными в схеме системы KBA;
* Блоки управления.prt – содержит схемы алгоритмов блоков управления задвижек, клапанов, насосов, а также модель датчика.

При загрузке и последующем расчете **kba.pak** проекты обмениваются данными через общую базу данных сигналов.

Для того, чтобы продемонстрировать совместную работу алгоритма управления на удаленном компьютере и локальной модели, нам необходимо выполнить ряд действий, которые изложены ниже.

* + 1. Скопируем директорию **C:\SimInTech\Demo\TPP\KBA** в директорию с аналогичными именем, находящуюся по пути **C:\SimInTech\Projects\KBA**.
    2. Для демонстрации совместной работы нам необходимо модернизировать файл **Алгоритмы KBA.prt**. Если мы откроем данный файл, то увидим, что алгоритмы управления задвижками насоса можно условно разделить на три группы: 1-я группа алгоритмов относится к насосу KBA31AP001, вторая группа алгоритмов относится ко второму насосу KBA32AP001, 3-я группа алгоритмов относится к третьему насосу KBA33AP001. При этом алгоритмы собраны по одинаковым схемам и имеют идентичный механизм работы. Для демонстрации совместной работы возьмём по 1 алгоритму из каждой группы, сгенерируем для них исходный код, который затем запустим на удаленном контроллере. Возьмем следующие алгоритмы:
* Алгоритм KBA31EY001;
* Алгоритм KBA32EY001;
* Алгоритм KBA33EY001.
  + 1. Создадим для каждого из алгоритмов отдельный prt-проект. Поместим **алгоритм KBA31EY001** в новый проект схемы автоматики, а затем сохраним его с именем **Управление задвижкой 1.prt**. Произведем аналогичные операции для алгоритмов KBA32EY001 и KBA33EY001, сохранив их в файлы с именами **Управление задвижкой 2.prt** и **Управление задвижкой 3.prt** соответственно.
    2. Настроим каждый из созданных проектов. Для этого зайдем в каждый проект по отдельности и укажем в параметрах расчета проекта в строке «**Имя (имена) алгоритма**» следующие имена:
* для проекта **Управление задвижкой 1.prt** - **kbaalgs1**
* для проекта **Управление задвижкой 2.prt** - **kbaalgs2**
* для проекта **Управление задвижкой 3.prt** - **kbaalgs3**

Для каждого из вновь созданных файлов проектов так же необходимо прописать пути доступа к базе данных на вкладке «**Настройка**» в параметрах расчета схемы.

* + 1. Произведем изменения в схемах алгоритмов. В блоках «**Чтение сигналов**» нужно изменить значение свойства «**Транслировать в исполнительную систему**» на «**Да**», а в блоке «**Выход алгоритма**» изменить значение свойства «**Транслировать из исполнительной системы**» на «**Да**». Данные изменения нужны для того, чтобы алгоритмы, которые будут обсчитываться на удаленном компьютере, могли участвовать в расчете общей модели, воспринимая изменения входящих в нее расчетных свойств.
    2. В проекте **Алгоритмы KBA.prt** удалим субмодели, отвечающие за эти алгоритмы, для того, чтобы не происходило одновременного расчета в разных файлах.
    3. Создадим с помощью инструмента «**Кодогенератор**» новую конфигурацию, в которую включим файлы **Управление задвижкой 1.prt, Управление задвижкой 2.prt, Управление задвижкой 3.prt** и сохраним ее с каким-либо именем, например, «**algs\_kba.alt**»
    4. Произведем генерацию исходных кодов для данной конфигурации, а затем загрузим ее на удаленный компьютер и произведем подключение к нему.
    5. Затем произведем запуск пакета проектов **kba.pak** и созданной конфигурации. В процессе расчета мы можем видеть, что модель, запущенная на локальном компьютере пользователя, обменивается данными с программой, запущенной на удаленном компьютере пользователя: те изменения, которые пользователь вносит в расчетную схему (например, при запуске насосов) передаются на расчет на удаленный компьютер, и возвращаются обратно с измененным после расчета состоянием.