

УТВЕРЖДАЮ
Генеральный директор

_____ В.Н. Петухов
« » июля 2020 года

**Работа с инструментом генерации кода Си в SimInTech
для встраиваемых систем по управлению ОС QNX и Linux**

Технический директор,
руководитель работ

подпись, дата

Ф.И. Баум

Оглавление

1	Введение.....	3
2	Структура ПО NordWind.....	3
3	Технология генерации кода и сборки расчетных модулей	5
3.1	Закладка «Загрузка»	6
3.2	Закладка «Настройки»	7
3.3	Закладка «Отладчик»	10
3.4	Закладка «Дополнительно»	12
3.5	Закладка «Список конфигураций»	12
4	Типы файлов SimInTech, с которыми работает инструмент кодогенератора	13
5	Основы работы с кодогенератором	13
6	Генерация СИ-кода для нескольких алгоритмов и запуск их на управляющем контроллере.....	19
7	Совместная работа алгоритма управления, работающего на удаленном компьютере, с локальной моделью, работающей на компьютере пользователя.	25

1 Введение

Программное обеспечение (ПО) SimInTech является мощным инструментом для разработки алгоритмов управления различной степени сложности. Одним из важных преимуществ SimInTech 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 затем автоматически преобразована при помощи встроенных инструментов SimInTech 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.

Для выполнения программного кода на контроллере с ОС QNX, Linux 32, 64 bit для архитектур x86, arm используется ПО NordWind (в случае выбора шаблона генерации СИ-кода NordWind) кроме СИ-кода происходит дополнительная сборка СИ-кода в динамически разделяемую библиотеку (.so расчетный модуль) и создание таблицы с описанием сигналов используемых алгоритмом управления. После трансляции библиотеки (.so) на контроллер и запуска ПО NordWind происходит инициализация библиотеки расчетного модуля и размещения в области разделяемой памяти массива внешних сигналов алгоритма, после чего происходит вызов функции алгоритма с заданным временным интервалом.

Стоит отметить, что ПО NordWind имеет возможность запускать на выполнение последовательно и параллельно (в виде потоков расчетных модулей) несколько расчетных модулей с разными временными тактами работы. Средства отладки TCP/IP сервер (GdbServer) позволяет отображать на исходной схеме работу расчетного модуля на контроллере в реальном масштабе времени, а также изменять значения констант.

2 Структура ПО NordWind

Программное обеспечение состоит из нескольких модулей (динамических библиотек), объединенных в единую программу NordWind, сервера удаленной отладки и архивного сервера, а также библиотеки для работы с базой данных сигналов реального времени.

В зависимости от требований к целевой системе состав динамических библиотек может быть различным.

- **NordWind** – основной модуль программы, который обеспечивает автоматическое создание общей областей памяти для внешних переменных расчетных модулей (общая область, описывающая массив структур внешних переменных и общая область памяти, хранящая значения внешних переменных, их загрузку, загрузку начального состояния прибора, загрузка библиотек расчетных модулей, функциональных модулей, реализованных в виде динамически подгружаемых библиотек;

- **libnet.so** – подсистема межсетевого обмена, реализует прием и посылку пакетов со значениями сигналов по сети используя сетевой стек TCP/IP;

- **libalg(1..n).so** – расчетные модули, реализующие алгоритмы управления, разработанный в SimInTech, имена библиотек могут быть произвольными, так их работа и порядок вызова определяется в файле загрузки;
- **libio.so** – подсистема связи с устройствами ввода/вывода, содержит прототипы функций, разрабатываемые для чтения и записи в устройства ввода/вывода, библиотека содержит набор сервисных функций, позволяющих обращаться к базе данных сигналов по именам и производить операции чтения и записи значений сигналов в базу данных;
- **libtrends.so** – подсистема архивирования;
- **libNetInit.so** – подсистема восстановления канала управления: копирует области памяти расчетных модулей у резервных каналов;
- **GdbServer** – сервер удаленной отладки алгоритмов, реализован в виде исполняемого файла;
- **DataBaseServer** – сервер получения архивных данных, по запросу программы просмотра архивных данных, реализован в виде исполняемого файла;
- **LibSdk** – представляет из себя библиотеку доступа к базе данных контроллера, содержит функции, позволяющие подключиться к базе данных реального времени контроллера из отдельной программы и функции, позволяющие изменять значения, статусы и времена возникновения сигналов.

Подробно описание работы и настройки модулей ПО NordWind можно найти в "Руководстве системного программиста" (документ предоставляется по запросу).

Циклограмма работы ПО NordWind на контроллере приведена ниже (Рисунок 1).



Рисунок 1 – Циклограмма работы ПО NordWind

3 Технология генерации кода и сборки расчетных модулей

Рассмотрим инструмент SimInTech, предназначенный для генерации кода. Для необходимо в главном окне программы войти в меню «**Кодогенератор**»: в результате перед пользователем возникнет выпадающее меню из двух строк: «**Кодогенератор Си**» и «**Кодогенератор ST**» (Рисунок 2).

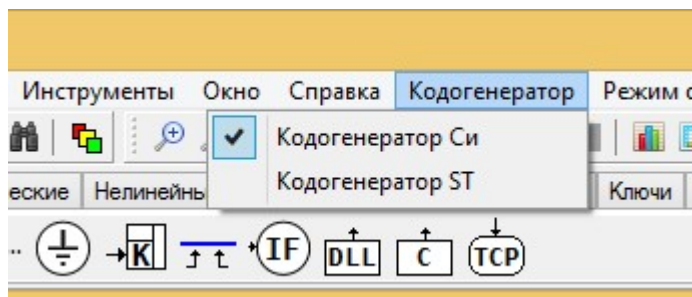


Рисунок 2 – Выбор кодогенератора

Для последующей работы мы будем использовать «**Кодогенератор Си**». Работа с кодогенератором ST (Руководство пользователя по кодогенератору ST предоставляется по отдельному запросу) будет аналогична работе с кодогенератором 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 окно кодогенератора (Рисунок 3).

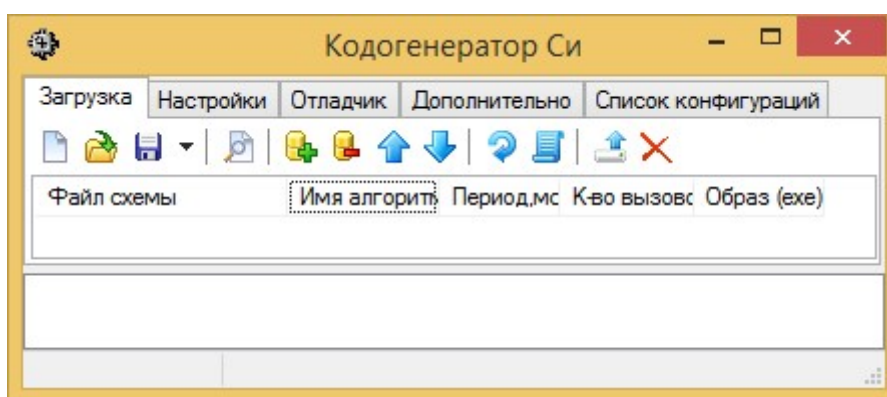


Рисунок 3 – Окно кодогенератора Си

В нем можно увидеть закладки, отвечающие за различные функции и настройки кодогенератора (Рисунок 4), а именно:

- Загрузка;
- Настройки;
- Отладчик;
- Дополнительно;
- Список конфигураций.

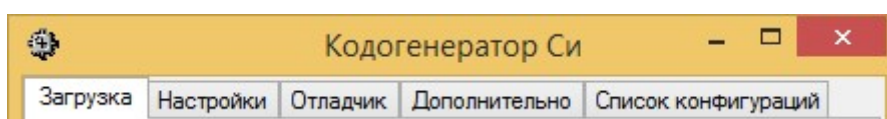


Рисунок 4 – Закладки окна кодогенератора Си

Рассмотрим каждую из них.

3.1 Закладка «Загрузка»

Закладка «**Загрузка**» отвечает за формирование списка задач (перечня проектов алгоритмов), предназначенных для генерации кода и загружаемых в исполнительную систему (Рисунок 5).

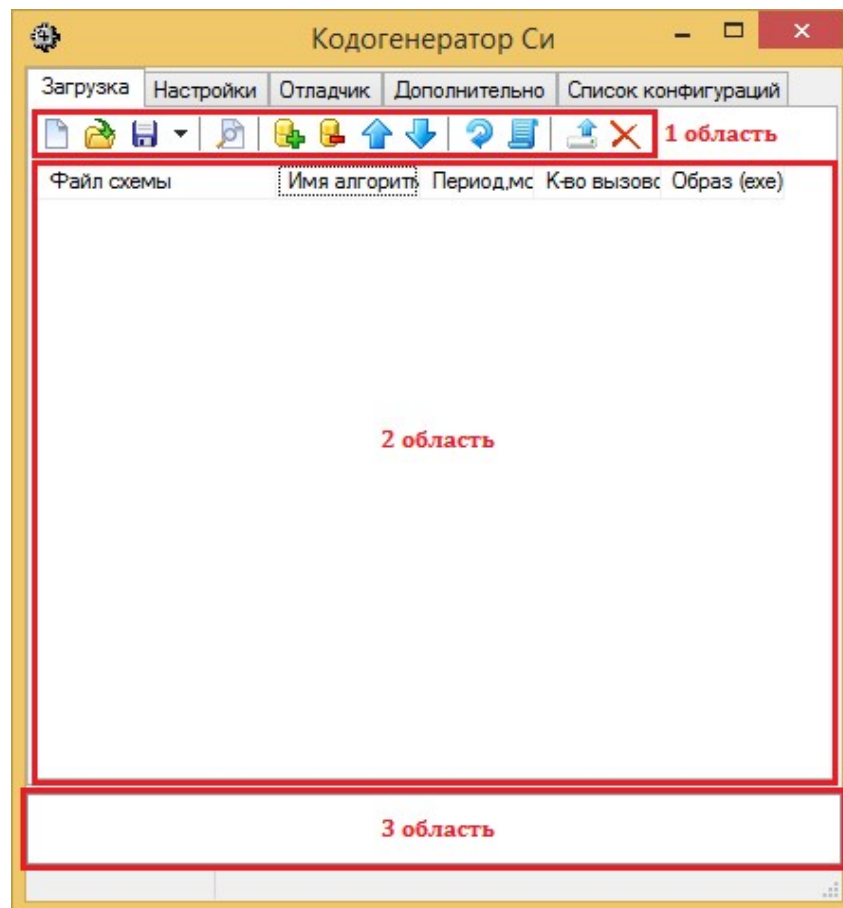



Рисунок 5 – Области закладки «Загрузки»

Можно выделить три области, формируемые при активации закладки «**Загрузка**»:

- 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.

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 те-

 3VS SOFTWARE DEVELOPMENT	Работа с инструментом генерации кода Си в SimInTech для встраиваемых систем	B01
--	---	-----

кущая таблица уже была сохранена, то нажатие на кнопку приведет к перезаписи файла, в который она была сохранена. Также пользователю доступна кнопка **«Сохранить конфигурацию 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 конфигурацию 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.

3.2 Закладка «Настройки»

Закладка **«Настройки»** отвечает за настройки кодогенератора (Рисунок 6).

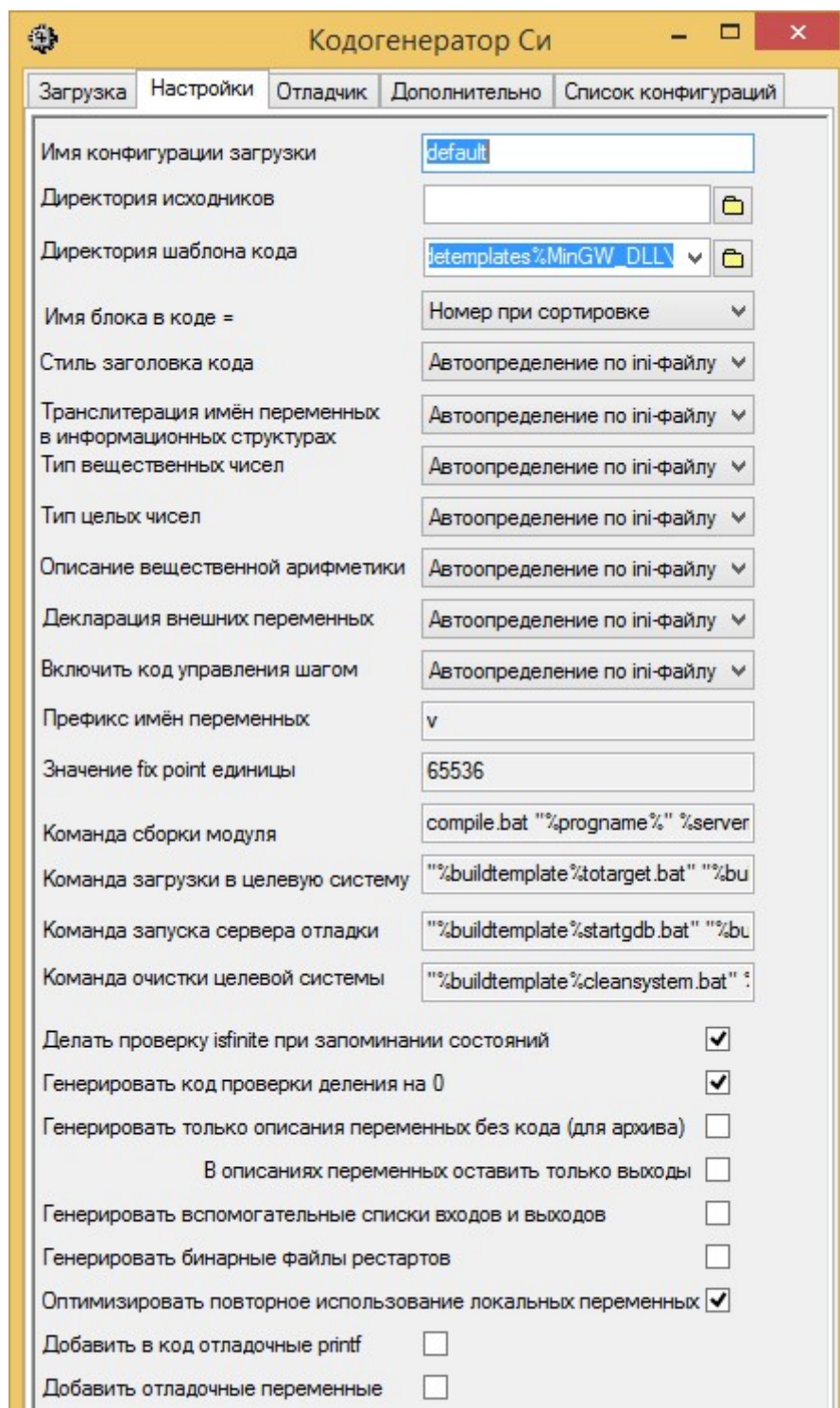


Рисунок 6 – Закладка «Настройки»

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.

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 исходников**» прописывается путь сохранения сгенерированных программ. Рядом с полем расположена иконка, при нажатии на которую возникает окно, с помощью которого можно 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.

- **Номер при сортировке.** В качестве базы для имени внутренних переменных в генерируемом коде Си используются сквозные номера, присвоенные 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.

- **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. Стоит отметить, для того, чтобы собрать из Си-кода расчетные модули, необходимо наличие соответствующих инструментальных средств на инструментальной машине SimInTech, например, для сборки расчетных модулей для ОС QNX Neutrino должна быть установлена IDE QNX Momentics, варианта Linux ARM хост компилятор.

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.

3.3 Закладка «Отладчик»

Закладка отвечает за настройки отладчика и управление расчетным процессом в приборе (Рисунок 7).

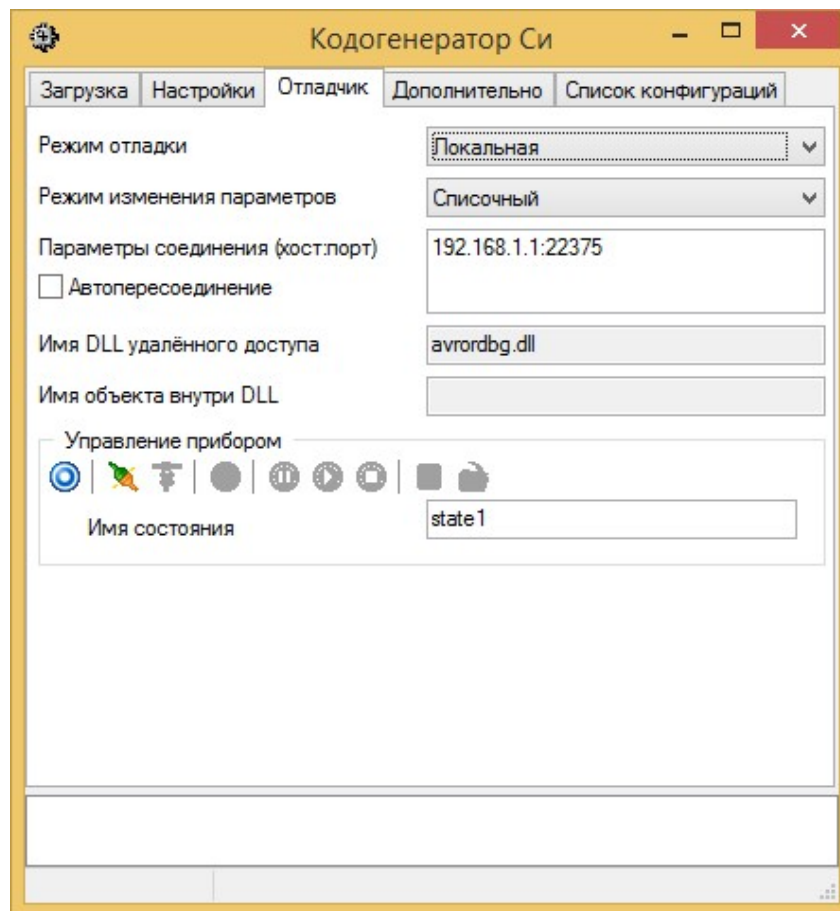


Рисунок 7 – Закладка «Отладчик»

Рассмотрим подробнее выпадающие списки и поля, находящиеся на данной закладке.

Список **«Режим отладки»** отвечает за выбор способа расчета схемы.

Для пользователя доступны три способа:

- **«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 компьютере, отображаются параметры блоков, которые насчитаны на приборе. В данном режиме пользователь на схеме может изменять свойства блоков, при этом будет осуществляться автоматическое изменение свойств соответствующих 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»** – 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 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»**.

У данных кнопок следующее назначение:

- **«Удаленный запуск сервера обмена»** – осуществляет запуск сервера отладки GdbServer на целевой системе;

- **«Подключиться к прибору»** – осуществляет соединение SimInTech, установленной на компьютере пользователя, с сервером отладки целевой системы;

- **«Отключить»** – осуществляет отключение SimInTech, установленной на компьютере 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»**.

3.4 Закладка «Дополнительно»

Закладка **«Дополнительно»** (Рисунок 8) содержит окно ввода настроек сетевой подсистемы для формирования файла конфигурации сетевого обмена. Подробно формат конфигурации и ключевые секции и слова – см. руководство пользователя модуля сетевого обмена.

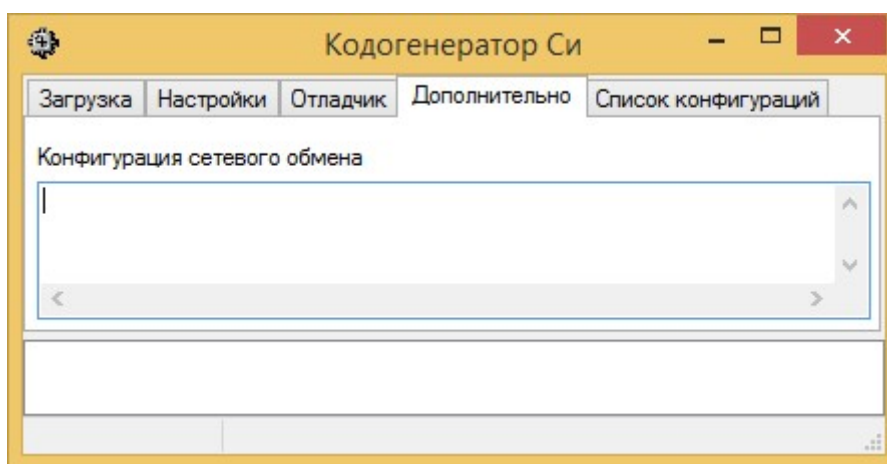


Рисунок 8 – Закладка «Дополнительно»

3.5 Закладка «Список конфигураций»

Закладка **«Список конфигураций»** (Рисунок 9) 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.

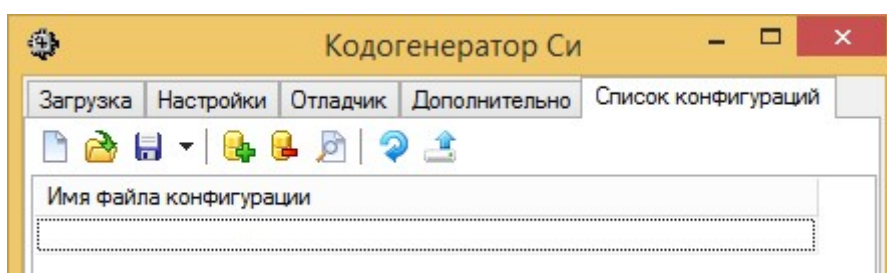


Рисунок 9 – Закладка «Список конфигураций»

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.

4 Типы файлов 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 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.

5 Основы работы с кодогенератором

Для получения навыков работы с кодогенератором сделаем следующее пошаговое упражнение, в рамках которого можно пройти основные этапы и действия, связанные с работой инструментом кодогенерации SimInTech.

В данной работе мы будем использовать ранее созданные в качестве примеров расчетные схемы, которые находятся в директории **C:\SimInTech\Demo**.

Для работы воспользуемся расчетной схемой регулятора, которая находится в директории **C:\SimInTech\Demo\Автоматика и математика\Типовые блоки алгоритмов АЭС\Штатный регулятор питания барабан-сепаратора (БС)\Штатный регулятор питания (ШРП) БС**.

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\Автоматика и математика\Типовые блоки алгоритмов АЭС\Штатный регулятор питания барабан-сепаратора (БС)\Штатный регулятор питания (ШРП) БС**:

- ШРП – штатный регулятор питания БС.mgr

- ШРП – штатный регулятор питания БС.prt
- Директория «блоки регулятора».

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 **C:\SimInTech\Projects\Регулятор** после выполнения действий 1 и 2, представлен на рисунке ниже (Рисунок 10).

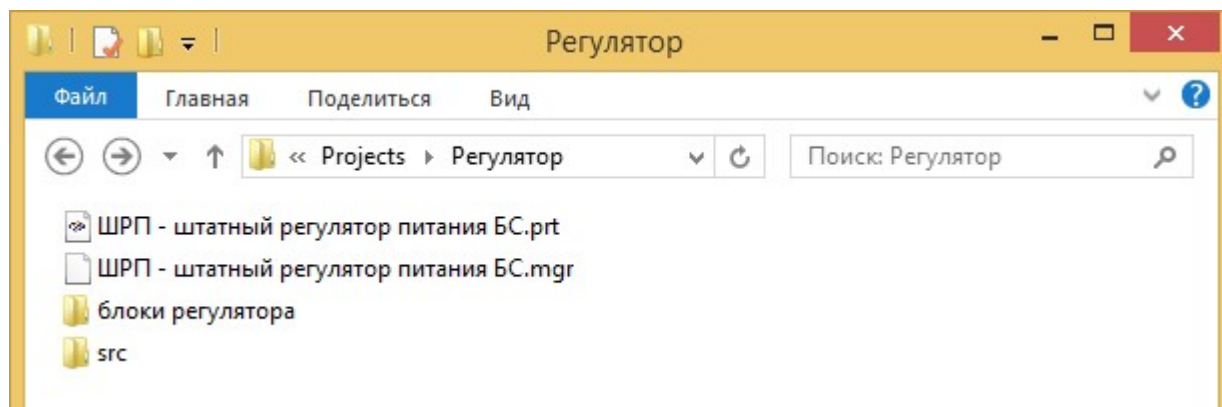


Рисунок 10 – Содержимое директории Регулятор

Приступим к выполнению упражнения:

1. В главном окне SimInTech заходим в меню **Кодогенератор** и активируем пункт **«Кодогенератор Си»**. Открывается окно кодогенератора с чистой конфигурацией.

2. 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 конфигурации, с которым сейчас работает кодогенератор (Рисунок 11).

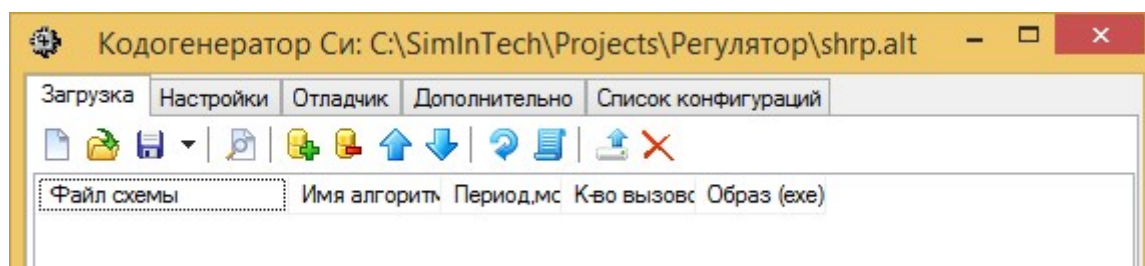


Рисунок 11 – Окно кодогенератора с загруженным файлом конфигурации

3. 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 (Рисунок 12).

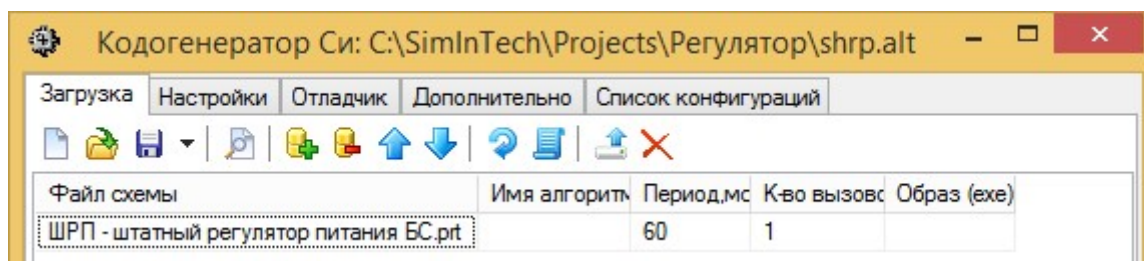


Рисунок 12 – Файл конфигурации с добавленным проектом

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.

4. 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 (Рисунок 13).

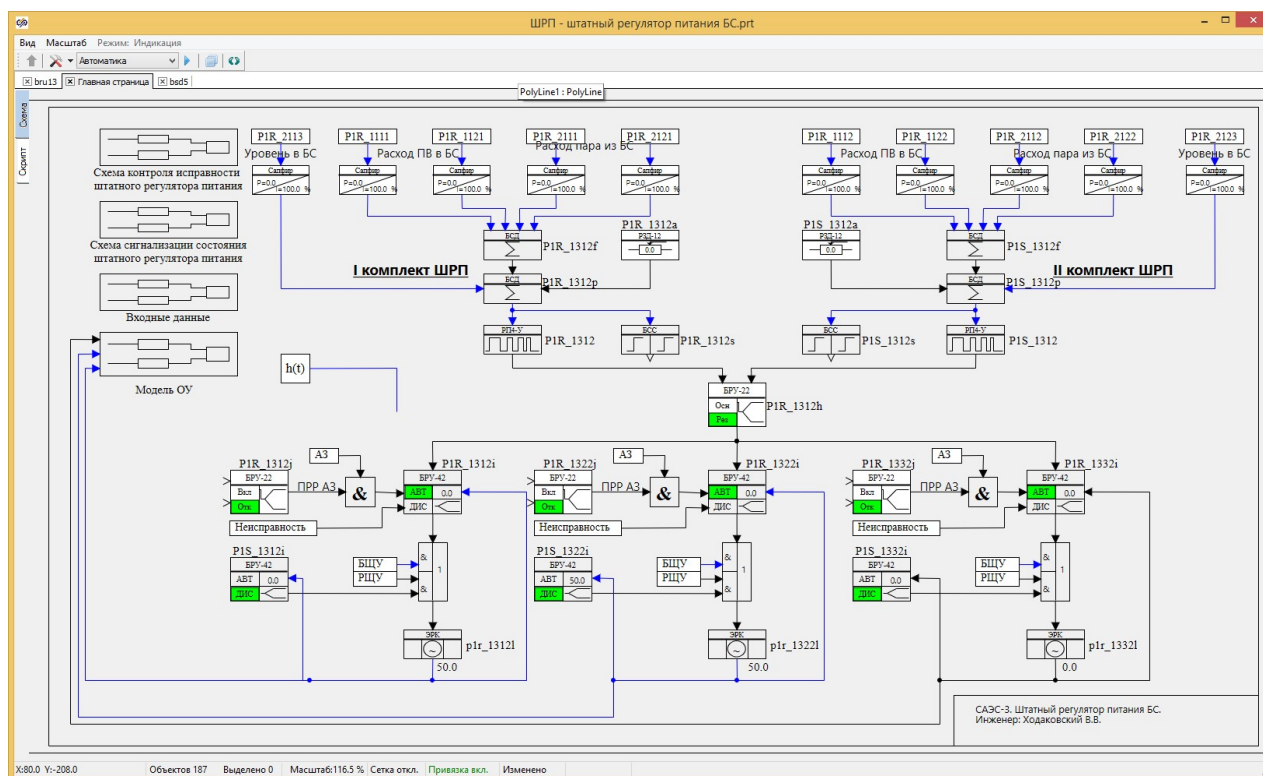


Рисунок 13 – Окно проекта с расчетной схемой

5. В открывшемся окне зайдем в **Параметры расчета** и укажем в свойстве «Имя (имена) алгоритма» имя **азрг1** (Рисунок 13). Сохраним внесенные изменения нажатием на кнопку «Ok».

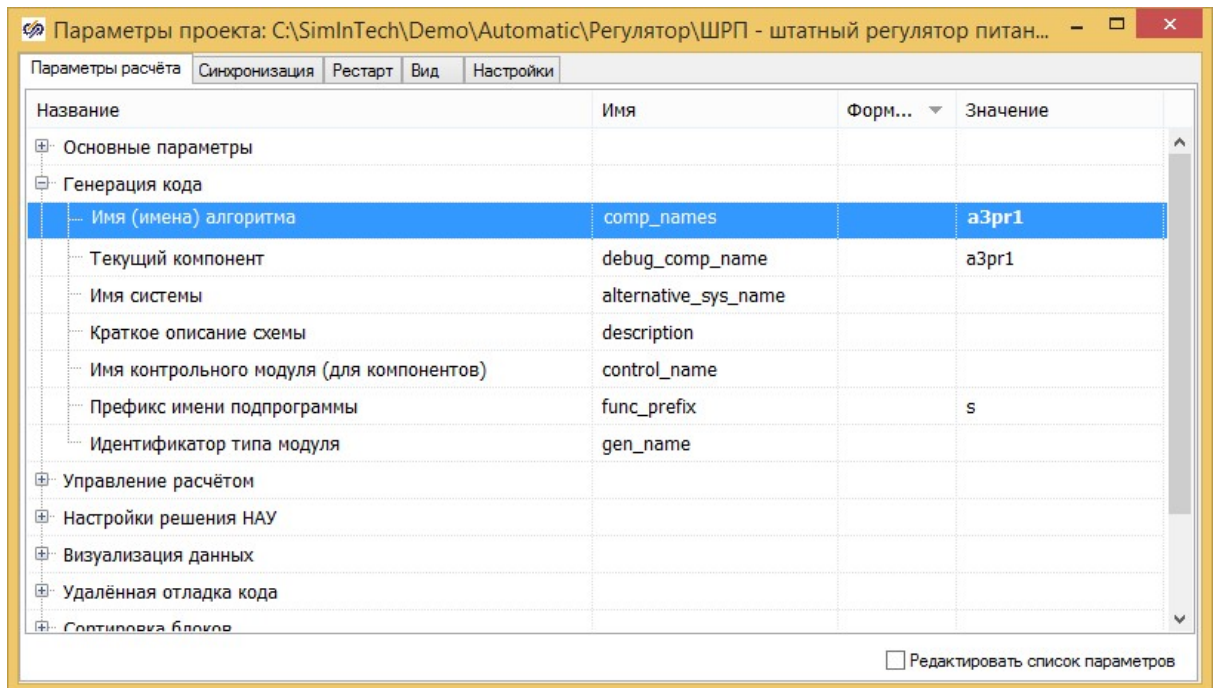


Рисунок 14 – Параметры расчёта схемы регулятора с выделенным свойством «Имя (имена) алгоритма»

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

6. Вернемся в окно кодогенератора. Перейдем на вкладку «**Настройки**». Сделаем изменения следующих настроек (Рисунок 15):

- **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.

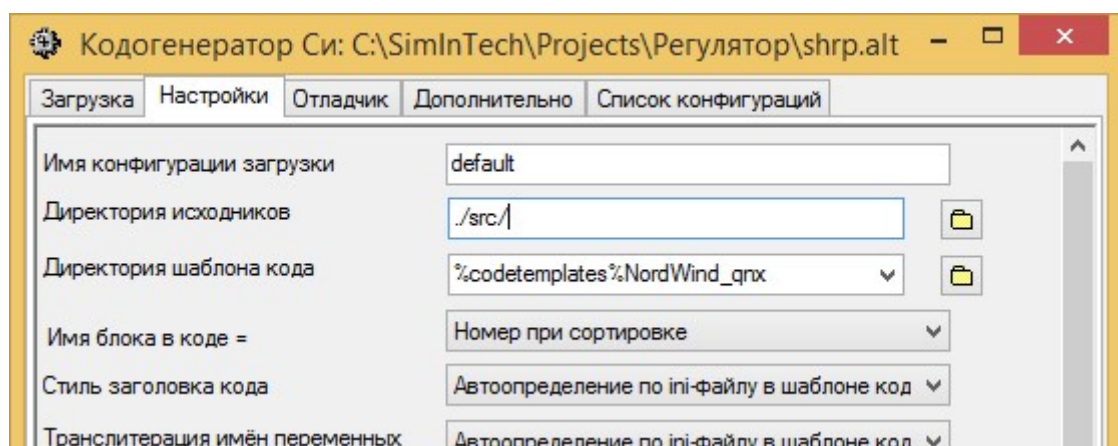


Рисунок 15 – Изменения на вкладке «Настройки»

7. Вернемся на вкладку «**Загрузка**». Нажмем на кнопку «**Пересобрать модули и конфигурацию**». В окне сообщений кодогенератора появится запись «**Генерация исходников завершена успешно**» (Рисунок 16).

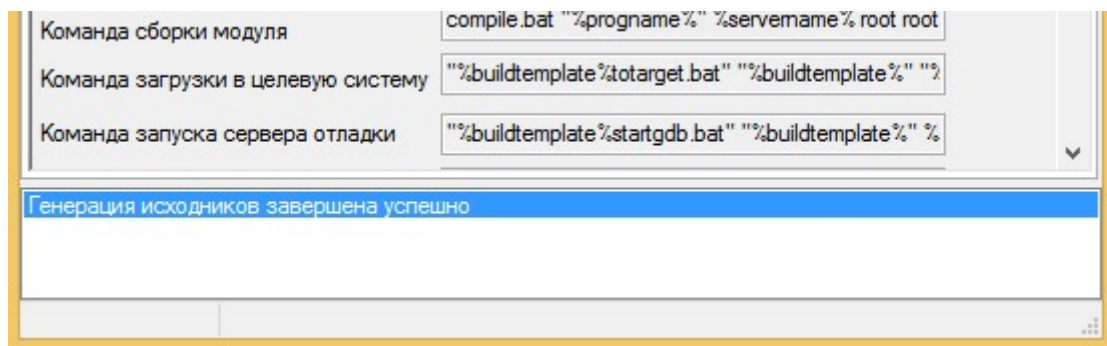


Рисунок 16 – Сообщение «Генерация исходников завершена успешно»

8. Зайдем в директорию **src**. После завершения генерации в ней появились файлы исходных кодов, а также расчетные модули **.so** (Рисунок 17) (при наличии необходимого компилятора или кросс-компилятора).

C:\SimInTech\Projects\Регулятор\src				
И	Имя	Размер	Дата	Время
..		Вверх	10.04.16	15:30
a3pr1.h		47229	10.04.16	15:24
a3pr1.inc		108154	10.04.16	15:24
a3pr1.inputs		941	10.04.16	15:24
a3pr1.log		21748	10.04.16	15:24
a3pr1.outputs		2327	10.04.16	15:24
a3pr1.so		85931	10.04.16	15:24
a3pr1.extvars.table		27520	10.04.16	15:24
a3pr1.intvars.table		96750	10.04.16	15:24
a3pr1_init.inc		49797	10.04.16	15:24
a3pr1_state.inc		3360	10.04.16	15:24
default.conf		18	10.04.16	15:24
default.list		7	10.04.16	15:24

Рисунок 17 – Каталог с файлами, получившимися после работы генератора кода

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

```
cd /home/NordWind/
./GdbSever -t 0 -m 0
```

Таким образом мы запустим GdbServer, используемый удалённо для запуска исполняемой среды NordWind, удаленной отладки и отображения результатов работы алгоритма на инструментальном компьютере. Существует возможность запуска NordWind из консоли: в этом случае нет необходимости запускать NordWind, используя сервер отладки, сервер отладки будет использован для удаленного контроля работы проекта.

10. Копируем через протокол ftp в директорию **/home/NordWind/** следующие файлы:

- **a3pr1.so** – файл с алгоритмом регулятора (расчетный модуль), 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, а при формировании файла **.conf** используется имя конфигурации загрузки, задаваемое на вкладке «Настройки» в кодогенераторе.

11. 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 приведен на рисунке ниже (Рисунок 18).

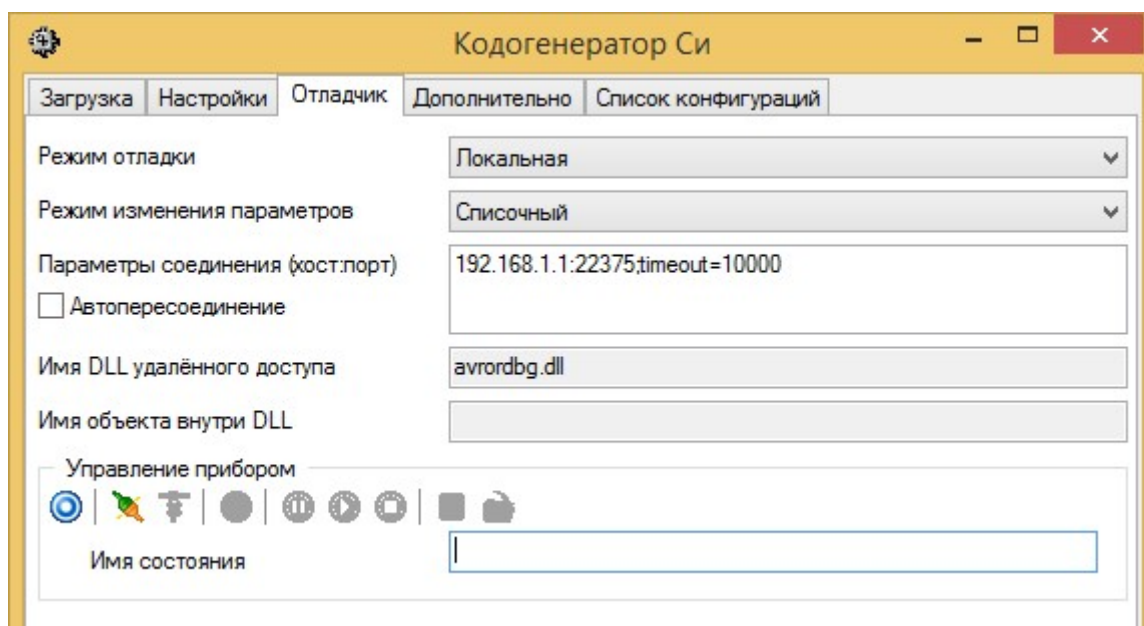


Рисунок 18 – Параметры расчета схемы регулятора с выделенным свойством «Имя (имена) алгоритма»

12. 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.

13. Перейдем в SimInTech. В меню «**Режим отладки**» устанавливаем режим «**Удаленный**» (Рисунок 19).

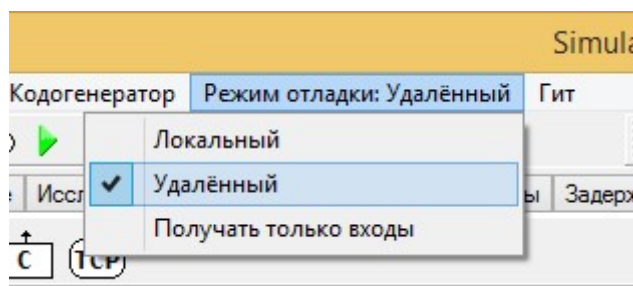


Рисунок 19 – Активация режима «Удаленный» в главном меню

14. 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.

15. Активируем окно с алгоритмом регулятора. Затем в главном окне SimInTech нажимаем на кнопку «Расчёт», расположенную на панели кнопок. При этом, т.к. режим работы выставлен в **Удаленный**, происходит не локальное моделирование схемы, а подключение к серверу удаленной 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.

16. 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».

17. 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» ввести 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». При этом состояние алгоритма будет изменено на то, которое мы загрузили.

6 Генерация СИ-кода для нескольких алгоритмов и запуск их на управляющем контроллере

В SimInTech можно производить сборку расчетных модулей для нескольких алгоритмов запускать и отлаживать их на удаленном контроллере.

Для демонстрации используем проект расположенный в директории **C: SimInTech\Demo\TRP\КВА_CTL**.

В состав пакета входят три проекта: **Управление задвижкой 1.prt**, **Управление задвижкой 2.prt**, **Управление задвижкой 3.prt**. В процессе моделирования мы добились желаемой работы алгоритмов совместно с математической моделью объекта управления (контура охлаждения ядерного реактора), предполагается, что при создании АСУ ТП данные алгоритмы будут выполняться на контроллере с ОС (QNX, Linux64) под управлением ПО NordWind.

Для этого нам необходимо получить из алгоритмов SimInTech расчетные модули (динамически разделяемые библиотеки **so**).

Выполним ряд действий:

1. Перейдем во вкладку "**Загрузка**" кодогенератора и добавим три файла проектов: **Управление задвижкой 1.prt**, **Управление задвижкой 2.prt**, **Управление задвижкой 3.prt** (Рисунок 20).

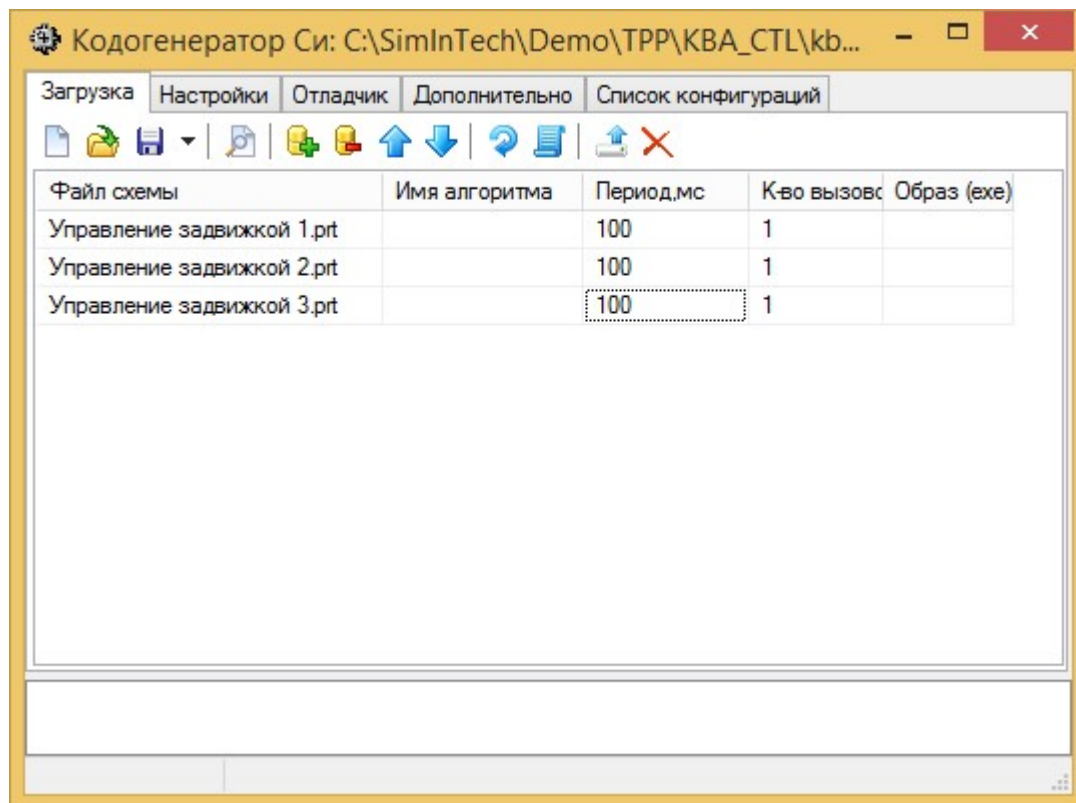


Рисунок 20 – Добавленные проекты в таблицу проектов кодогенератора

2. Перейдем во вкладку "**Настройки**" и выберем шаблон для генерации СИ-кода и сборки расчетных модулей в зависимости от ОС инсталлированной на контроллере (Рисунок 21).

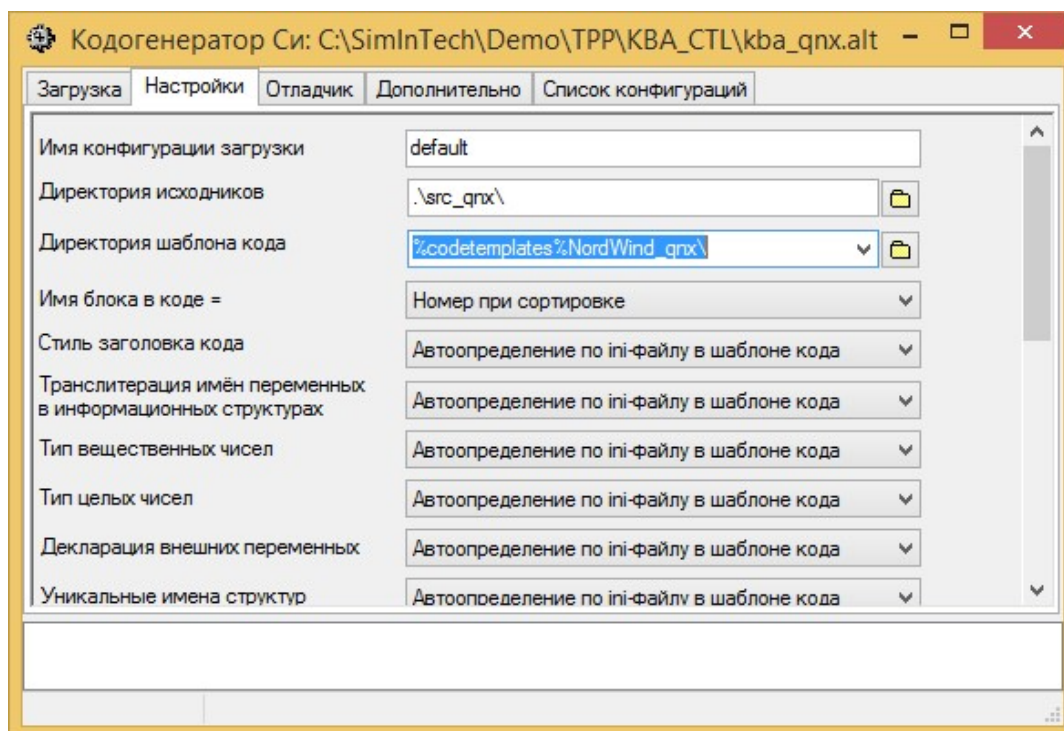


Рисунок 21 – Выбор шаблона для генерации СИ-кода и сборки расчетных модулей для ОС QNX

Мы выбрали шаблон генерации Си-кода "**NordWind_qnx**" это означает, что Си-код будет собран в расчетный модуль для ОС QNX Neutrino 6.50. В том случае, если на контроллере установлена ОС Linux, необходимо выбрать шаблон генерации Си-кода "**NordWind_x64**" (шаблон будет работать и в случае 32bit архитектуры).

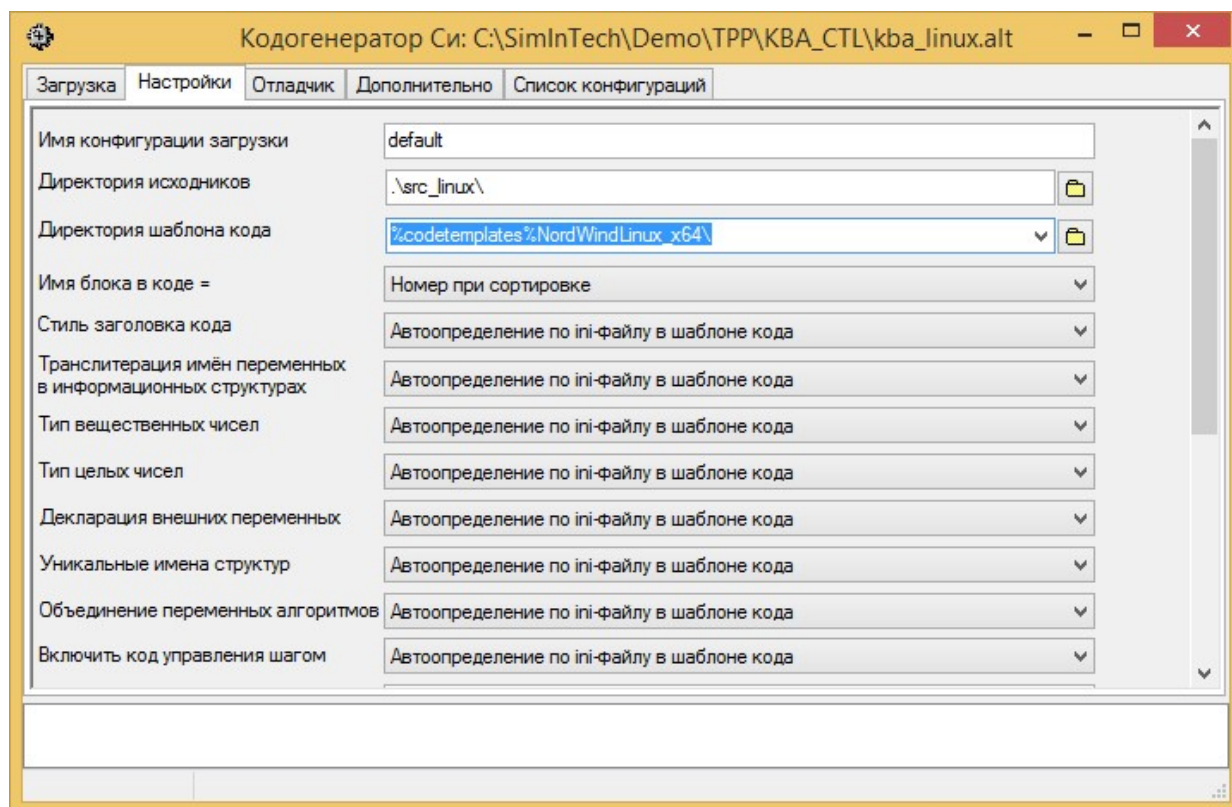


Рисунок 22 – Выбор шаблона для генерации СИ-кода и сборки расчетных модулей для ОС Linux64

В том случае, если выбраны шаблоны сборки расчетных модулей **"NordWind_qnx"** и **"NordWind_Linux_Arm"**, то средства компиляции установлены на инструментальной машине, т.е. весь процесс генерации Си-кода и сборки расчетных модулей будет происходить на машине разработчика SimInTech, то есть в результате работы кодогенератора мы получаем Си-код алгоритмов SimInTech и расчетные модули алгоритмов (динамически разделяемые библиотеки **.so**).

В случае использования шаблона сборки расчетных модулей **"NordWindLinux_x64"** генерация Си-кода и дальнейшая сборка расчетных модулей будет происходить на Linux машине. В качестве Linux машины можно использовать виртуальную машину с установленным на ней ОС Linux или контроллер с ОС Linux, для которого мы собираем расчетные модули, поэтому для сборки расчетных модулей на Linux машине необходимо организовать доступ по протоколу ssh (регистрация пользователя), а в закладке **"Отладчик"** определить имя пользователя (логин) и пароль доступа на Linux машину.

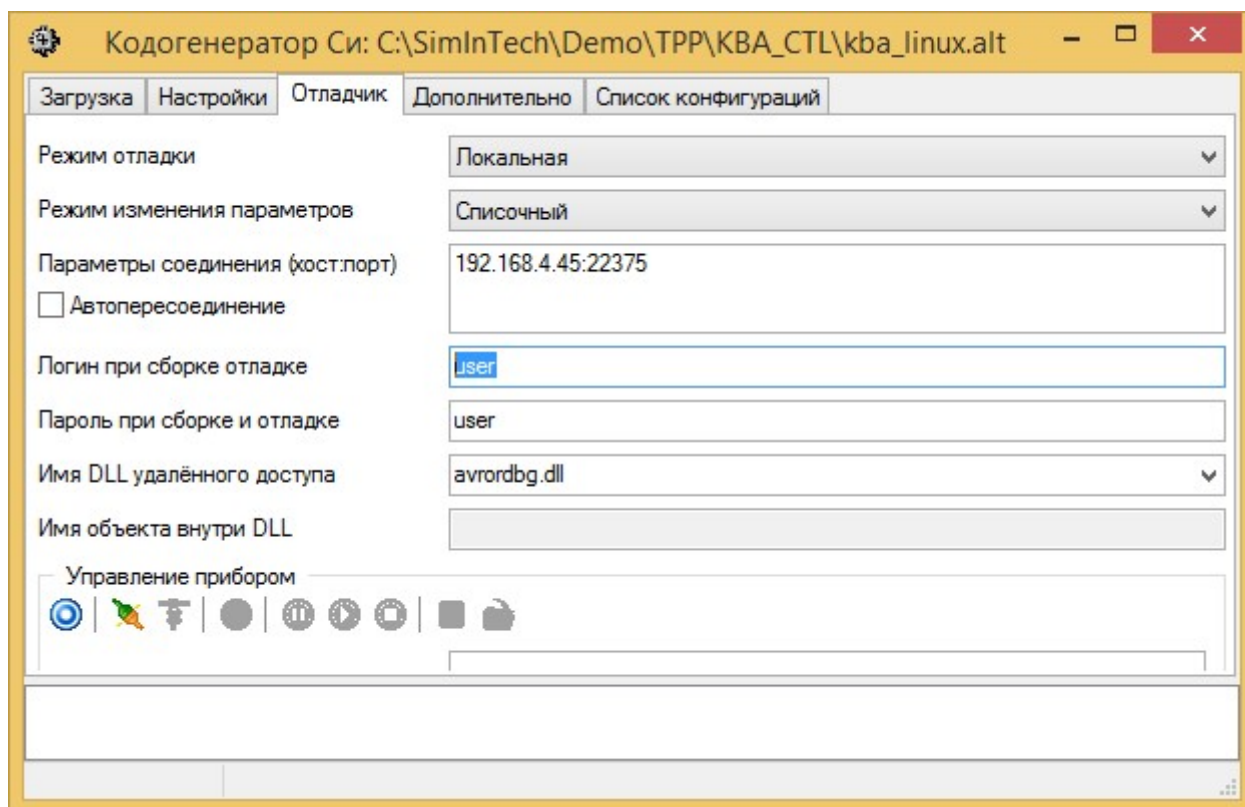


Рисунок 23 – Закладка "Отладчик". Определение логина пользователя и пароля доступа на машину сборки расчетных модулей

3. После нажатия на кнопку «**Пересобрать модули и конфигурации**» на вкладке «**Загрузка**», сгенерированный Си-код алгоритмов будет транслирован на Linux машину, где будет произведена сборка расчетных модулей, после завершения которой, они будут транслированы в указанную пользователем директорию исходников инструментальной машины SimInTech.

На рисунке 24 выделены цветом расчетные модули (**kbaalgs1.so, kbaalgs2.so, kbaalgs3.so**) и таблицы с сигналами (**kbaalgs1.extvars.table, kbaalgs2..extvars.table, kbaalgs3..extvars.table, kbaalgs1.intvars.table, kbaalgs2.intvars.table, kbaalgs3.intvars.table**). Рисунок 24 будет характерен для всех типов шаблонов Nord-Wind.

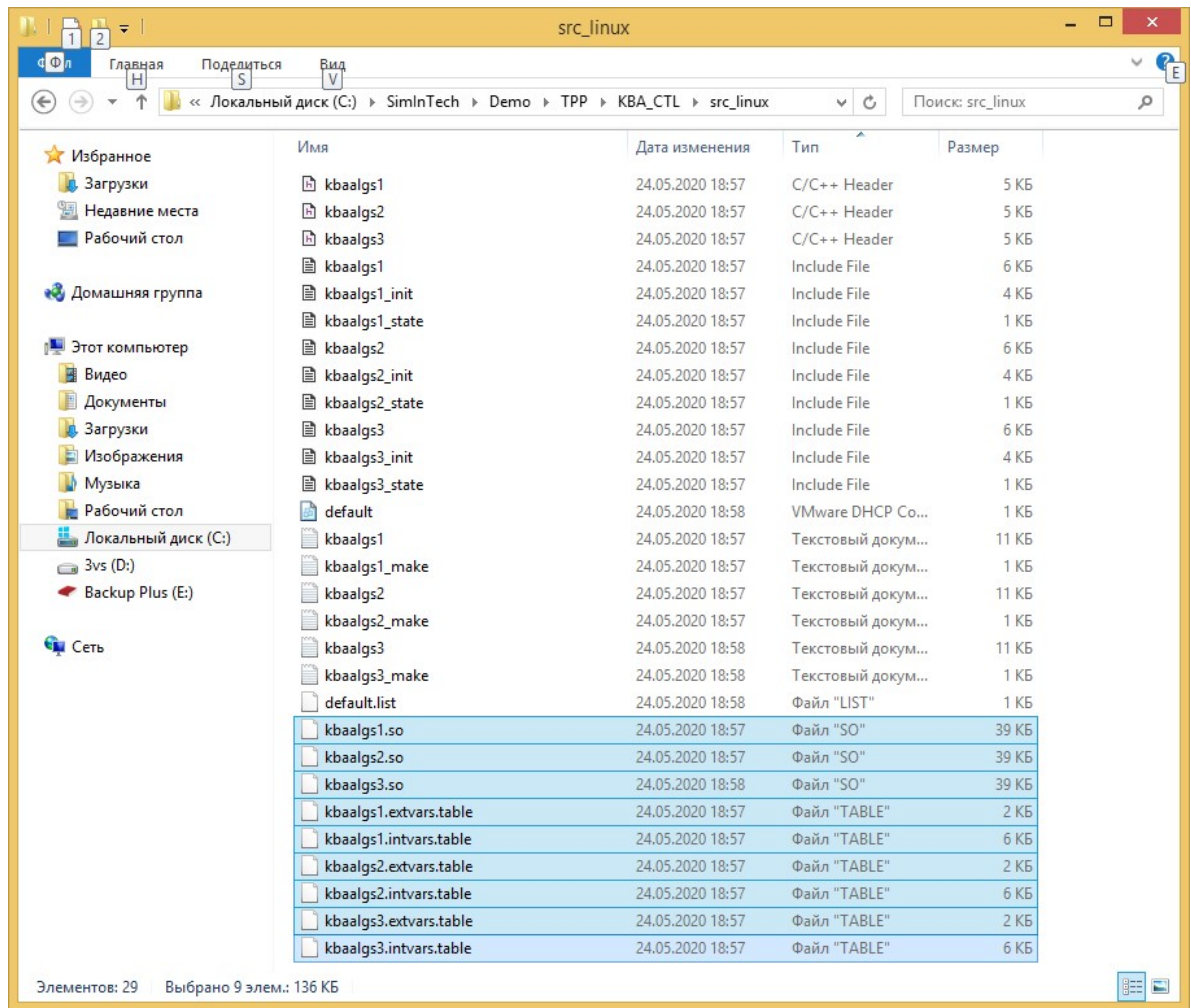


Рисунок 24 – Директория исходников содержит Си-код алгоритмов и расчетные модули

Файл с конфигурацией загрузки **default.conf** будет содержать информацию для загрузки и работы трех расчетных модулей (Рисунок 25).

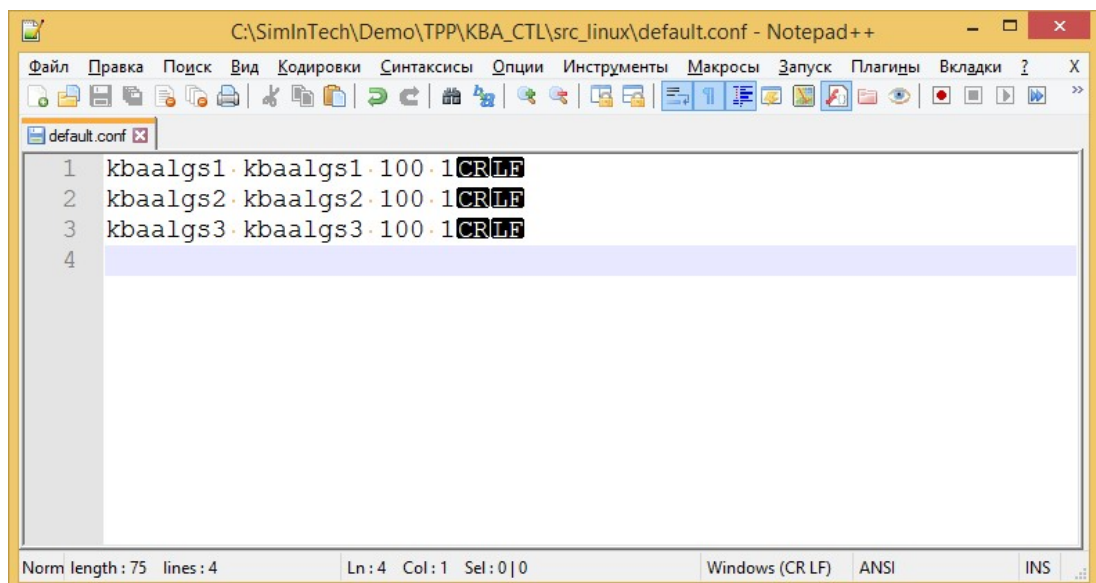
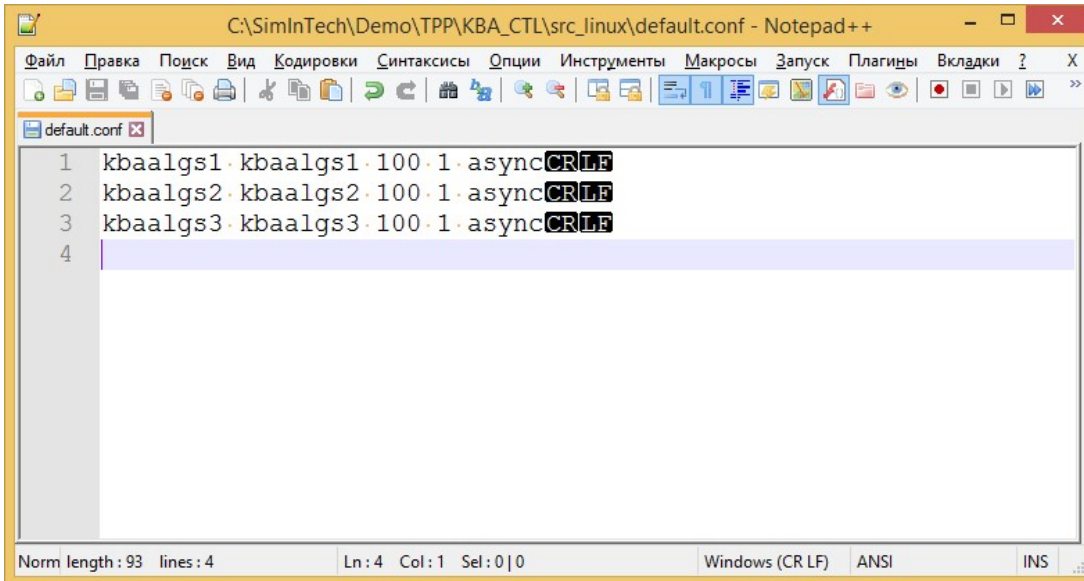


Рисунок 25 – Файл загрузки расчетных модулей для трех алгоритмов

Далее в соответствии с п. 10 главы «**Технология генерации кода и расчетных модулей**» транслируем расчетные модули, таблицы с сигналами и файл загрузки расчетных модулей на контроллер и производим удаленную отладку трех алгоритмов в SimInTech. Существует возможность настроить параллельную работу расчетных модулей на целевой системе, в этом случае расчетные модули будут выполняться в виде потоков. Подобное решение следует принимать в том случае, если ваш контроллер располагает достаточным количеством процессорных ядер и логика работы алгоритмов не требует их последовательного выполнения, в этом случае файл конфигурации загрузки будет выглядеть следующим образом (Рисунок 26).



```
C:\SimInTech\Demo\TPP\KBA_CTL\src_linux\default.conf - Notepad++
Файл Правка Поиск Вид Кодировки Синтаксисы Опции Инструменты Макросы Запуск Плагины Вкладки ? X
default.conf x
1 kbaalgs1 · kbaalgs1 · 100 · 1 · async CR LF
2 kbaalgs2 · kbaalgs2 · 100 · 1 · async CR LF
3 kbaalgs3 · kbaalgs3 · 100 · 1 · async CR LF
4
Norm length : 93 lines : 4 Ln : 4 Col : 1 Sel : 0 | 0 Windows (CR LF) ANSI INS
```

Рисунок 26 – Файл загрузки расчетных модулей для трех алгоритмов для реализации многопоточной работы

В директории SimInTech\Demo\TPP\KBA_CTL расположены сохраненные конфигурации для трех шаблонов сборки расчетных модулей: kba_qnx.alt, kba_linux.at, kba_linux_arm.alt, для контроллеров с ОС QNX, Linux64 и Linux arm соответственно.

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

В 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 SimInTech можно следующим образом (схема решения упрощенно выглядит так):

- Подготавливаем **алгоритмы** управления, отлаживаем их;
- Разрабатываем **модель** теплогидравлической системы, используя соответствующие теплогидравлические коды, отлаживаем ее;
- В составе комплексной модели (объект управления + алгоритмы управления) исследуем и оптимизируем алгоритмы управления, подбираем значения оптимальных коэффициентов регуляторов и добиваемся заданного качества переходных процессов;
- Используя инструмент генерации Си-кода на основе алгоритмов управления собираем расчетные модули и транслируем их на контроллер управления;

– Обеспечиваем **совместную работу** теплогидравлической модели на SimInTech и расчетных алгоритмов на контроллере управления.

Такой подход дает следующие преимущества: еще до испытаний на объекте уже можно с высокой долей вероятности получить представление обо всех особенностях работы алгоритма, подключить к контроллеру управления дополнительное оборудование, 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.

Существует два варианта решения данной задачи:

Вариант 1

Контроллер управления будет обмениваться данными с моделью объекта в SimInTech через интернет. В качестве примера будем использовать проект КВА из директории C:\SimInTech\Demo\TRP\КВА_CTL.

КВА - промконтур охлаждения ядерного реактора и представляет собой набор проектов с общей базой данных сигналов, работающих в пакете проектов **kba.pak**.

Каждый проект отвечает за определенную область моделирования:

- КВА.prt – содержит расчетную схему теплогидравлической модели системы КВА;
- Алгоритмы КВА.prt – содержит схемы алгоритмов, осуществляющих управление задвижками, клапанами, насосами и другими механизмами, установленными в схеме системы КВА;
- 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. Для демонстрации совместной работы нам необходимо собрать расчетные модули для трех алгоритмов управления задвижками насосов КВА31AP001, КВА32EY001, КВА33EY001, соответственно **Управление задвижкой 1.prt**, **Управление задвижкой 2.prt**, **Управление задвижкой 3.prt**.

2. Настроим каждый из созданных проектов. Для этого зайдём в каждый проект по отдельности и укажем в параметрах расчета проекта в строке «**Имя (имена) алгоритма**» следующие имена (Рисунок 27):

- для проекта **Управление задвижкой 1.prt** - **kbaalgs1**;
- для проекта **Управление задвижкой 2.prt** - **kbaalgs2**;
- для проекта **Управление задвижкой 3.prt** - **kbaalgs3**;

3. Также реализуем это для проектов **Управление задвижкой 2.prt**, **Управление задвижкой 3.prt**.

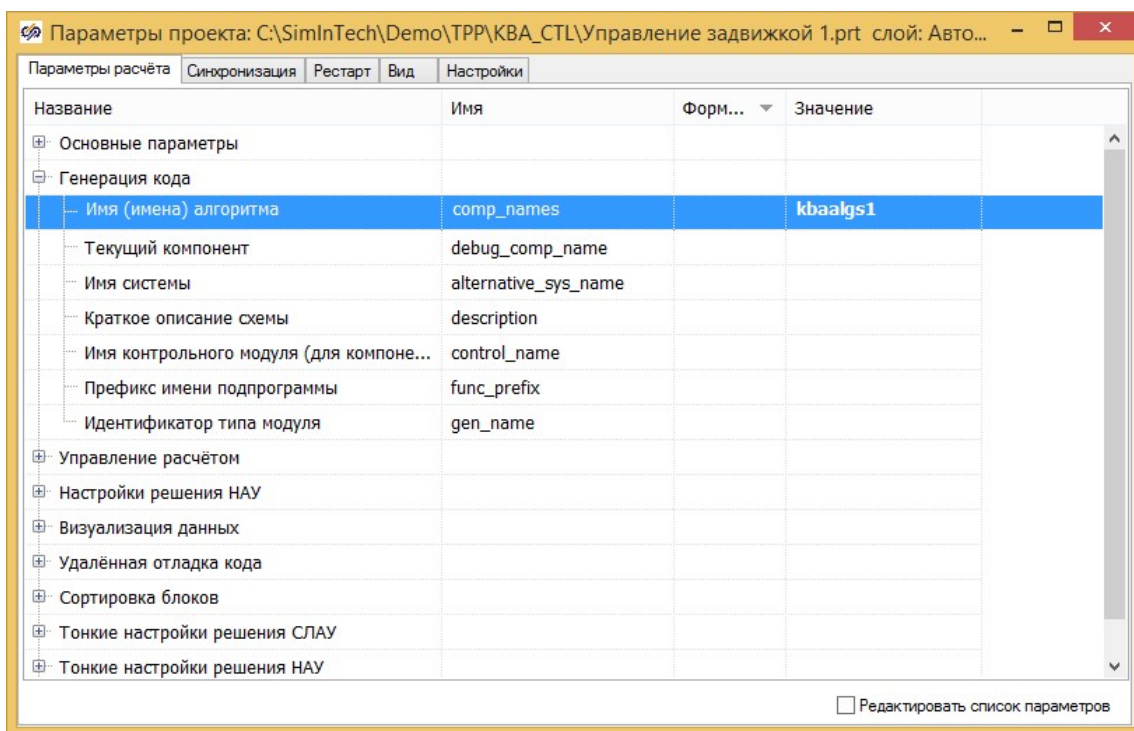


Рисунок 27 – Задание имени алгоритма в параметрах проекта

4. Необходимо изменить в параметрах расчета в секции «Удаленная отладка кода->Транслировать все входы алгоритмов в исполнительную систему и все выходы» установить «да». Данные изменения нужны для того, чтобы алгоритмы, которые будут работать на удаленном компьютере, могли получать данные от модели и возвращать в модель результаты расчета.

В секции «Удаленная отладка кода ->Режим отладки» установить «Удаленный» (Рисунок 28) для всех трех алгоритмов, иначе расчет алгоритмов будет происходить на локальной машине.

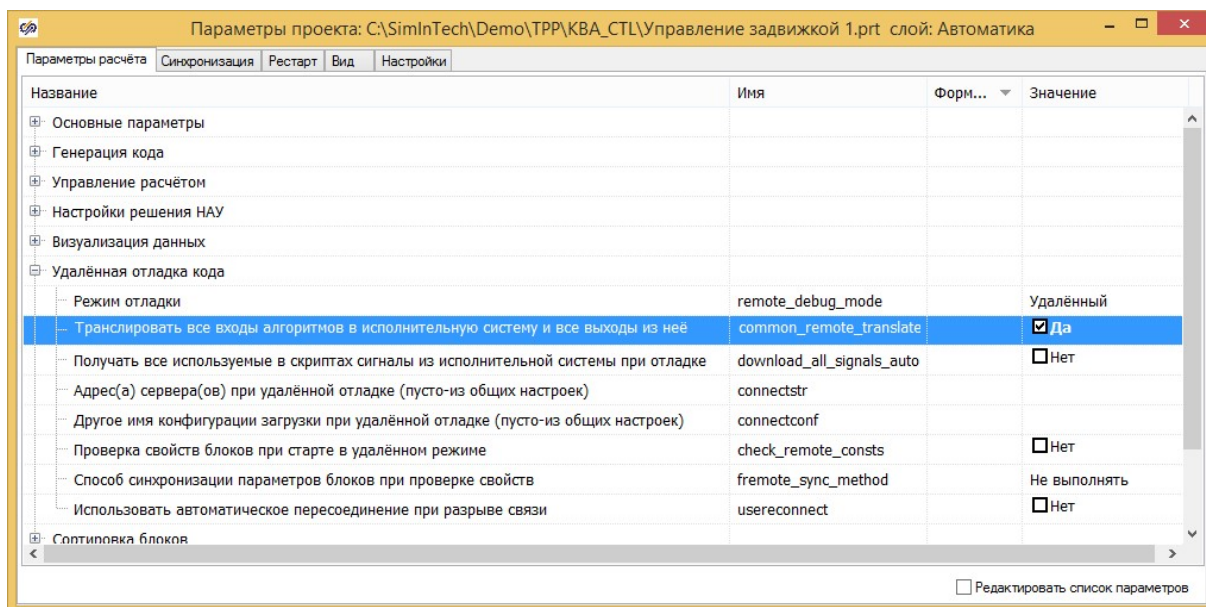
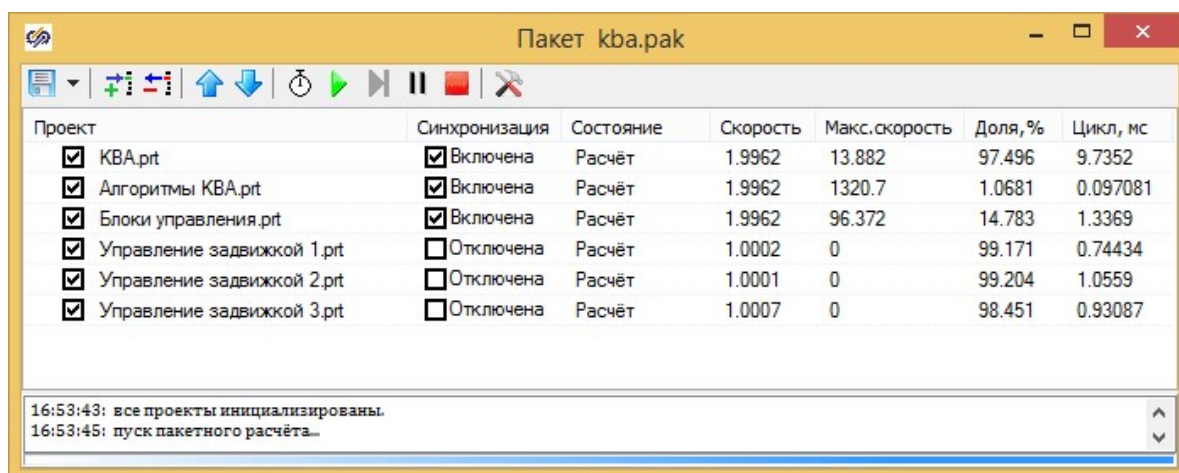


Рисунок 28 – Установка режима транслировать все входы алгоритмов в исполнительную систему и все выходы из нее, режим отладки удаленный

5. 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**», выше мы рассматривали генерацию расчетных модулей для трех алгоритмов.

6. Произведем генерацию расчетных модулей для данной конфигурации, транслируем ее на контроллер и используя инструменты удаленной отладки запустим три удаленных расчетных модуля на счет. В том случае если предполагается использование трех контроллеров при проектировании АСУ ТП можно каждый расчетный модуль удаленно запустить на отдельном контроллере.

7. Затем произведем запуск пакета проектов **kba.pak** (режим отладки: локальный) входящие в состав пакета **Управление задвижкой 1.prt**, **Управление задвижкой 2.prt**, **Управление задвижкой 3.prt** запущены удаленно. В процессе работы пакета мы можем видеть, что модель, запущенная на локальном компьютере пользователя, обменивается данными с расчетными модулями, запущенными на контроллере: те изменения, которые пользователь вносит в расчетную схему (например, при запуске насосов) передаются по сети интернет расчетным модулям на контроллере, и возвращаются обратно с измененным после расчета состоянием.



Проект	Синхронизация	Состояние	Скорость	Макс. скорость	Доля, %	Цикл, мс
<input checked="" type="checkbox"/> KBA.prt	<input checked="" type="checkbox"/> Включена	Расчёт	1.9962	13.882	97.496	9.7352
<input checked="" type="checkbox"/> Алгоритмы KBA.prt	<input checked="" type="checkbox"/> Включена	Расчёт	1.9962	1320.7	1.0681	0.097081
<input checked="" type="checkbox"/> Блоки управления.prt	<input checked="" type="checkbox"/> Включена	Расчёт	1.9962	96.372	14.783	1.3369
<input checked="" type="checkbox"/> Управление задвижкой 1.prt	<input type="checkbox"/> Отключена	Расчёт	1.0002	0	99.171	0.74434
<input checked="" type="checkbox"/> Управление задвижкой 2.prt	<input type="checkbox"/> Отключена	Расчёт	1.0001	0	99.204	1.0559
<input checked="" type="checkbox"/> Управление задвижкой 3.prt	<input type="checkbox"/> Отключена	Расчёт	1.0007	0	98.451	0.93087

16:53:43: все проекты инициализированы.
16:53:45: пуск пакетного расчёта...

Рисунок 29 – Запуск пакета на расчет

Вариант 2

Вариант совместной работы математической модели объекта и алгоритмов управления на контроллере реализуется в том случае если контроллер получает значения от датчиков и выдает управляющие воздействия с помощью устройств ввода/вывода. В данном случае контроллер готов к работе в составе АСУ ТП и мы имеем возможность испытать все аспекты его работы:

- качества работы алгоритмов регулирования;
- работа устройств ввода/вывода.

Для этого необходимо разработать устройство сопряжения с объектом (УСО). УСО связано через интернет с моделью SimInTech, а с контроллером с помощью устройств ввода/вывода. Технология совместной работы модели объекта управления и алгоритмов на контроллере заключается в следующем: математическая модель в SimInTech транслирует значения физических параметров модели через интернет на УСО. В задачу УСО входит преобразование значений физических величин в электрические величины и трансляцию их через устройства ввода/вывода контроллеру, который после получения электрических величин и преобразования их в физические зна-

чения измеряемых параметров, реализовывает цикл регулирования, возвращая управляющие воздействия на УСО.

УСО, получив управляющие воздействия с устройств вывода контроллера, преобразует их в значения физических величин и по сети интернет передает модели SimInTech. В том случае, если время моделирования на стороне SimInTech равно или быстрее реального, совместная работа модели и контроллера позволит с высокой степенью достоверности испытать программное обеспечение, разработанное для управляющего контроллера, алгоритм регулирования и корректность работы устройств ввода вывода. Стоит отметить, что ПО контроллера не обязательно должно быть разработано с помощью технологии SimInTech.

Возможен вариант взаимодействия модели и контроллера без использования УСО, например, с помощью обмена пакетами данных по протоколу udp.