**Создание блоков на языке Си**

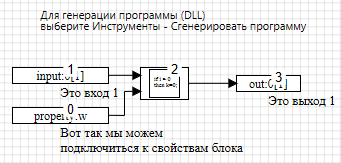
Среда динамического моделирования технических систем SimInTech является открытой средой, обеспечивающей в т.ч. расширение существующих моделирующих библиотек блоков или создание новых библиотек, путем создания новых блоков на любом языке программирования, в том числе и на языке Си. Один из наиболее быстрых способов создания собственных блоков на языке Си — это использование готовых шаблонов кода, входящих в дистрибутив.

Шаблоны SimInTech по умолчанию настроены таким образом, чтобы пользователь мог получить готовую DLL с минимальными затратами времени на настройку средств компиляции. В частности, в поставку включены шаблоны генерации кода, и предварительно настроенные компиляторы, и данная возможность значительно ускоряет создание как программ в виде DLL для Windows, так и исполняемого кода для аппаратуры управления различных производителей.

В случае, когда пользователь использует собственные средства создания программы, включая среду разработки и компилятор, необходимо настроить рабочие директории для совместной работы среды SimInTech и пользовательских средств разработки. При этом за первое приближение можно использовать исходные коды, созданные встроенными средствами генерации кода.

**Описание примера – заготовки блока**

В приложенном к тексту письма примере находится проект, в котором создан пример создания нового блока методом прямой генерации кода.



**Рисунок 1. Пример создания блока на языке Си.**

В качестве примера создается блок (номер 2 на схеме) типа «Язык Программирования» который на каждом шаге расчета выполняет вычисление по формуле:

**y = sin (x\*w)**, где

**у** – выход блока;

**x** – вход блока;

**w** – параметр (свойство), задаваемый пользователем.

Для определения параметров будущей DLL, мы используем специальные блоки:

Два блока типа *«Входной контакт S3»*

**Input:0** – будет создан порт входа в блок.

**Property:w -** будет использовано значение свойства “w” будущего блока.

Один блок типа *«Выходной контакт S3»*

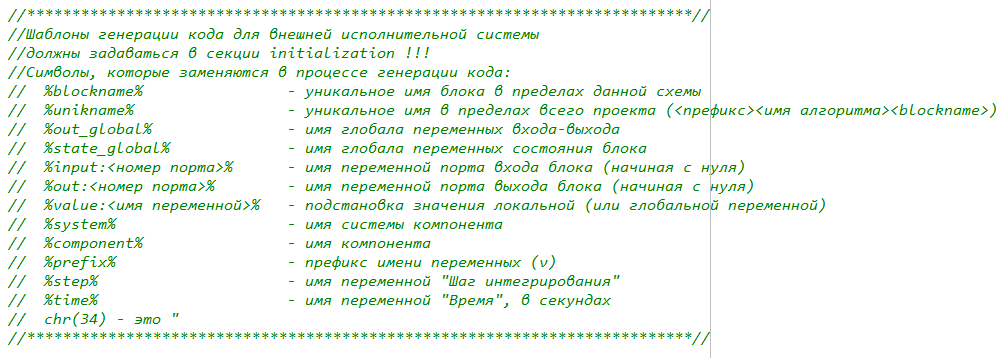
**Output:0** – будет создан один выход из блока.

Внутри блока 2 находится текст, подготовленный к генерации кода Си. В данном тексте пользователь может сформировать любой код, с использованием данных, задаваемых на схеме. Например, можно генерировать код, который использует имена блоков, задаваемых пользователем при создании схемы. Пользователь меняет данные на схеме и получает изменения в коде, для загрузки в контроллер.

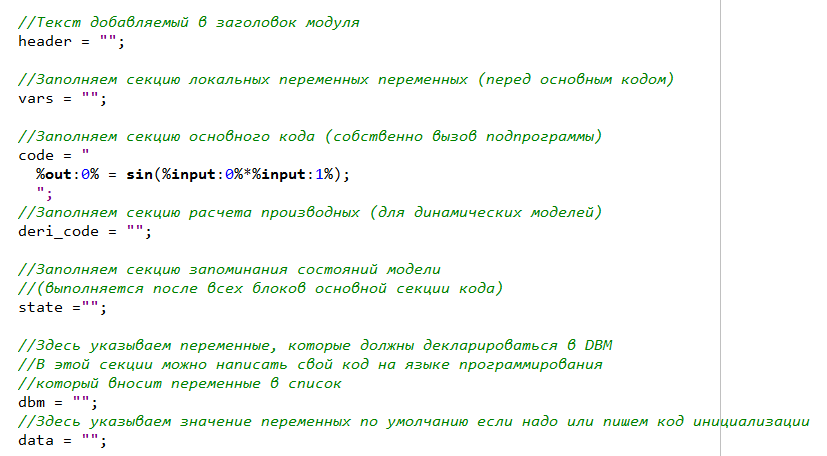
Рассмотрим текст блока в среде SimInTech.

Между ключевыми словами секции инициализации (**initialization** и **end;**) находится основной текст для **генерации кода Си.**

На рисунке 2 представлена часть, описывающая процесс замены специальных символов в тексте на значения из расчетной схемы.



**Рисунок 2. Текст комментариев для пояснения значений шаблона.**



**Рисунок 3. Текс основной части для генерации на коде Си.**

В нашем случае заполняется только секция расчета выхода по входу.

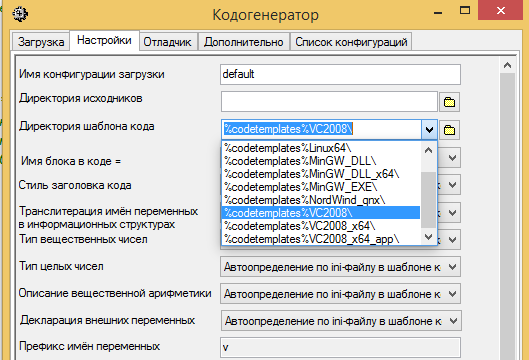
**out:0 –** выход;

**input:0 –** первый входной порт на схеме (см рисунок 1);

**input:1 –** второй входной порт в блоке на схеме (значение, задаваемое в свойствах будущего блока) (см. рисунок 1).

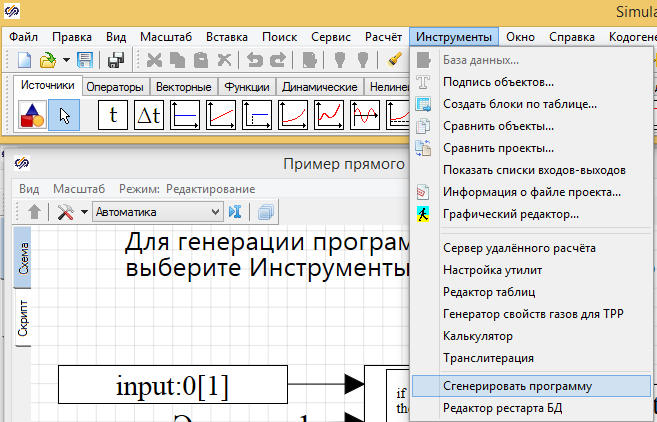
**Генерация кода Си**

Для настройки генерации кода необходимо настроить параметры кодогенерации, включая выбор требуемого шаблона. Пункт главного меню *«Кодогенартор»*. Диалоговое окно настройки кодогенератора представлено на рисунке 4. На данном рисунке выделен шаблон генерации кода, для компиляции с помощью среды Visual Studio 2008.



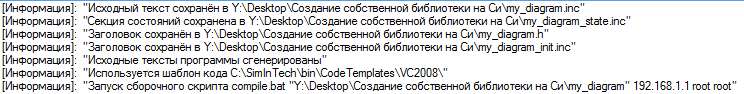
**Рисунок 4. Окно настройки параметров генерации кода.**

После настройки шаблона необходимо сгенерировать программу. Для этого выбираем в главном окне пункт меню *«Инструменты»* подпункт сгенерировать программу *«Сгенерировать программу»* (см. рисунок 5).



**Рисунок 5. Пункт меню сгенерировать программу.**

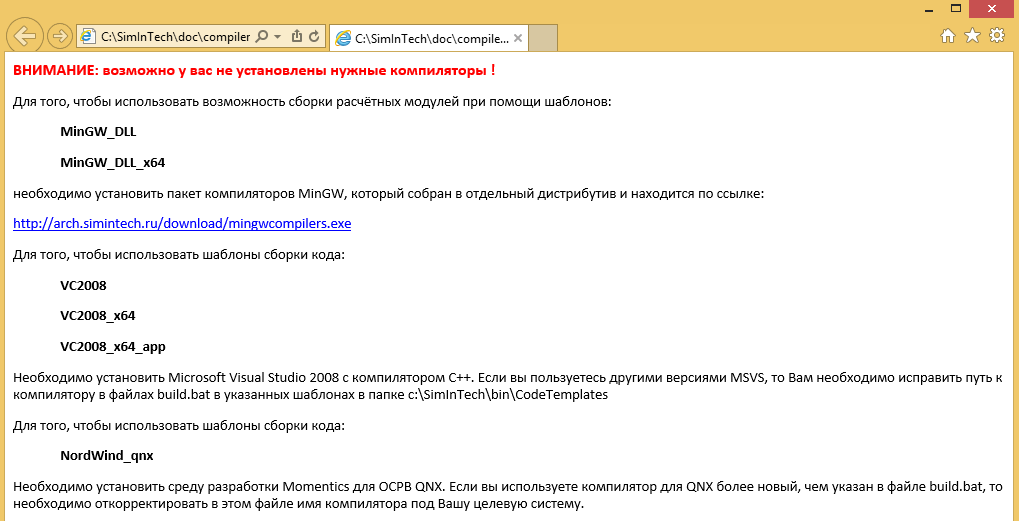
В результате работы генератора кода, произойдет создание всех необходимых файлов исходных кодов для компиляции dll c использованием Visual Studio 2008. В окне сообщений будет выведена информация как на рисунке 6.



**Рисунок 6. Сообщения о создании исходных кодов**

Если в системе установлено ПО Visual Studio 2008, то будет выполнена компиляция библиотеки и размещение ее в той же папке, где и исходный проект.

Если средства компиляции не установлены, то будет выведено следующее окно с предупреждением об отсутствии необходимых компонентов (см. рисунок 7).

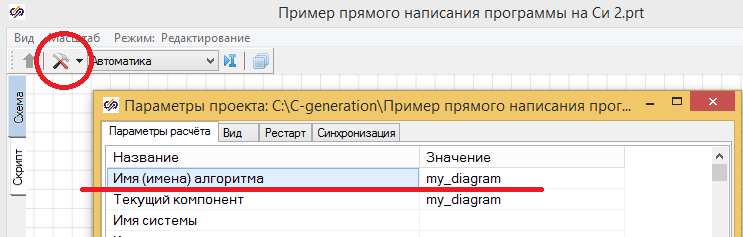


**Рисунок 7. Окно с предупреждением об отсутствии средств компиляции dll.**

**Работа с текстами Си.**

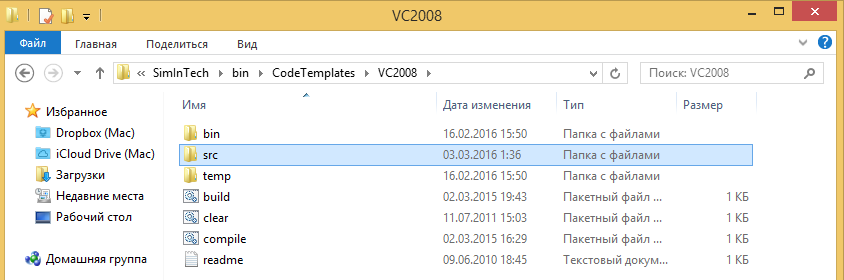
Вне зависимости от того, есть ли у вас соответствующий компилятор, исходный код уже сгенерирован, подготовлен для компиляции и помещен в соответствующую папку.

Имя dll задается в параметре расчета (кнопка с молотком, см. рисунок 8), по умолчанию имя задано как «my\_diagram».



**Рисунок 8. Задание имя dll в параметрах расчета.**

Для шаблона под Visual Studio 2008, исходные тексты будут помещены папку C:\SimInTech\bin\CodeTemplates\VC2008. Состав папки представлен на рисунке 9



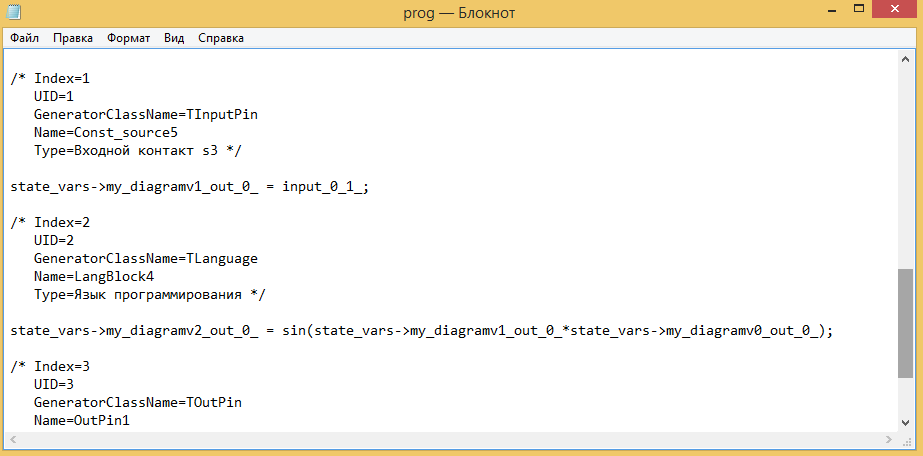
**Рисунок 9. Файлы в папке с щаблоном генерации кода Си для Visual Studio 2008.**

Данная папка настроена на автоматическое создание dll, и содержит командные файлы для запуска процедуры создания dll из исходных кодов.

Для работы с текстами Си и их ручной правки необходимо перенести всю папку вместе с ее содержимым в другую рабочую директорию, где и будет происходить непосредственно разработка блока.

Исходные коды, непосредственно отвечающие за выполнение математического расчет находятся в папке *src*.

В частности, непосредственно вычисление функции блока **y = sin (x\*w)**, находится в файле prog.inc (см. рисунок 10).



**Рисунок 10. Участок кода Си, осуществляющий расчет функции блока.**

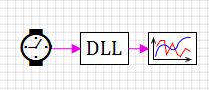
Для формирования уникальных имен переменных при автоматической генерации кода используется имена алгоритма, имена блоков и номера блоков.

Пользователь может самостоятельно менять код на Си для реализации собственной библиотеки блоков. И компилировать без использования командного файла, в том числе в режиме сторонней среды программирования.

**Использование созданной dll в проекте SimInTech.**

В приложенном к данному письму архиве находится проект *Загрузчик.prt.*

Структурная схема проекта приведена на рисунке 10.



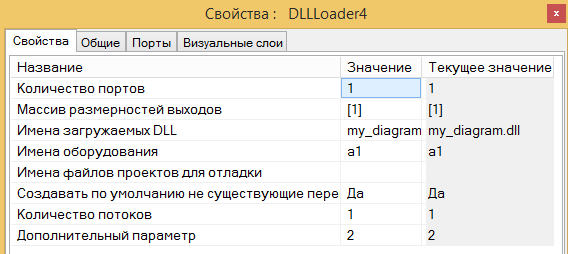
**Рисунок 10 структурная схема для подключения dll.**

Данная схема работает следующим образом:

1. Блок с часами формирует сигнал, соответствующий модельному времени.
2. Этот сигнал подается на блок, который мы тестируем.
3. Блок производит вычисление выхода.
4. Сигнал выхода блока отображается на графике.

Для подключения dll использован блок «*Внешняя dll*» распложённый в закладке *«Субструктуры»*.

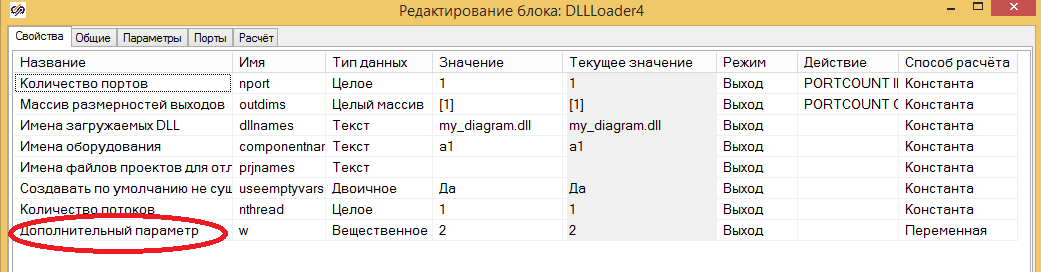
Свойства блока приведены на рисунке 11.

**Рисунок 11. Настройка блока «Внешняя dll».**

Пользователь задает количество портов, размерности портов (на вход в блок может быть подан вектор), а также указывает имя dll, которую нужно загрузить во время расчета.

Для нашего блока необходим дополнительный параметр w, который пользователь задает в свойствах блока. Что бы его добавить к стандартному блоку, необходимо:

1. выделить блок на схеме;
2. выбрать в главном окне пункт меню «*Правка*» подпункт «*Изменить блок*» (SimInTech должен находится в режиме разработчика);
3. в диалоговом окне добавить новое свойство (см. рисунок 12).



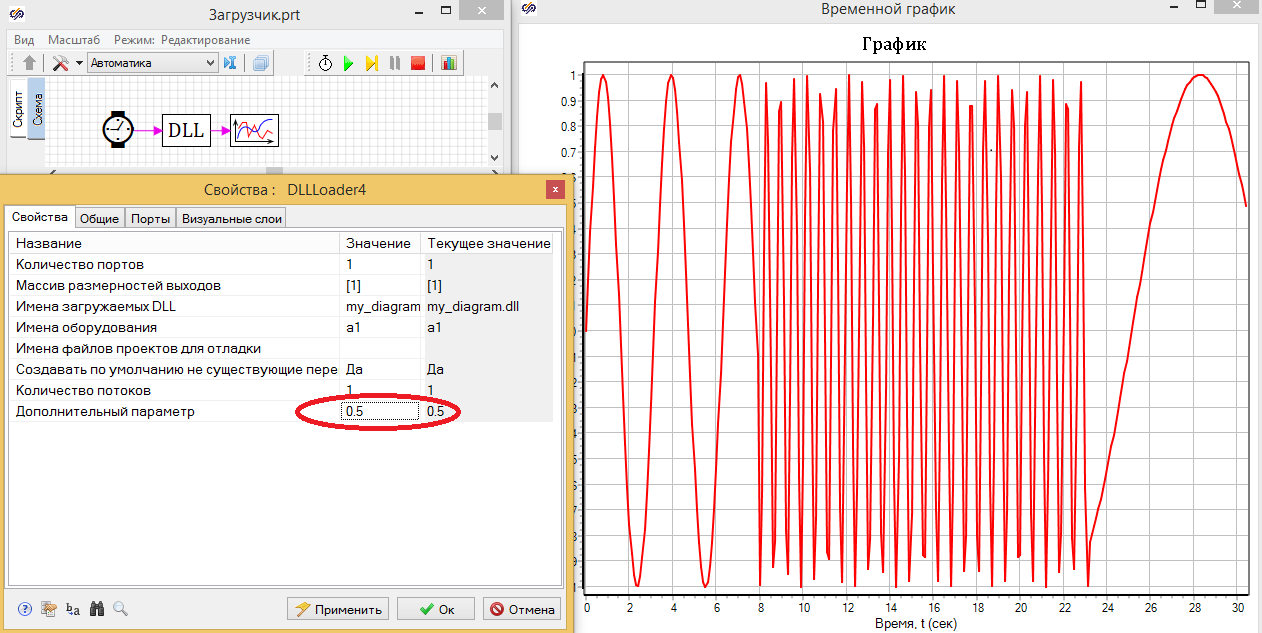
**Рисунок 12. Добавление параметра блока, задаваемого пользователем.**

Если запустить файл проекта «Загрузчик.prt» на расчет.

То будет подгружена my\_diagram.dll.

В данной dll происходит вычисление выхода по формуле у=sin(w\*x), где х – это модельное время, подаваемое на вход, w – параметр, задаваемый пользователем в настройках блока.

Пользователь во время расчета может войти в свойства блока dll и менять параметр w, на графике соответствующим образом будет меняться частота колебаний (см. рисунок 13).



**Рисунок 13. Тестирование блока, созданного в виде dll, на языке Си.**

**Заключение**

В данном тексте описаны действия по созданию простейшего блока на языке Си.

Пример может быть использован для разработки и подключения собственных блоков.