**Понятие «Субмодель»**

*Возможности работы с блоком «Субмодель», неограниченный уровень вложенности субмоделей*

Блок «Субмодель» является стандартным блоком библиотеки SimInTech и присутствует во всех библиотеках. Существует два основных варианта применения данного блока:

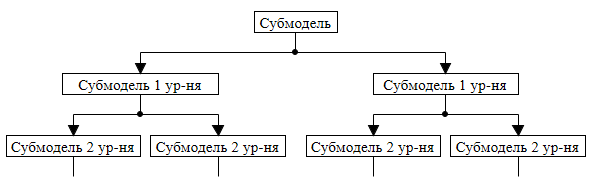
1. Логическое разделение схем и структурирование схем, набранных в SimInTech;
2. Создание на основе блока «Субмодель» нового структурного блока.

Рассмотрим подробнее особенности и преимущества первого варианта применения блока «Субмодель».

Допустим, что нам нужно создать несколько гидравлических схем, либо схем автоматики на основе известных нам. Пускай структурно они разделены, но при этом каждая схема теми или иными параметрами влияет друг на друга. Ясно, что с точки зрения моделирования и настройки, удобнее и правильнее совместить эти схемы в одном проекте SimInTech: с точки зрения моделирования, потому что не нужно создавать множество проектов, а потом их совместную работу, а с точки зрения настройки потому, что довольно легко перемещаясь по единому проекту, контролировать настройку параметров. При этом можно поместить все схемы в одно поле видимости, но зачастую получившаяся итоговая схема очень неудобна с точки зрения работы с ней, т.к. для контроля параметров схемы приходится постоянно перемещаться по областям, удаленным друг от друга. При грамотной работе с SimInTech правильно разнести такие схемы по разным субмоделям. Сделав это, мы обеспечиваем с одной стороны структурную обособленность каждой схемы (т.е. мы точно будем знать, что в каждой субмодели будет лежать тот «кусок» схемы, который мы сами туда определили), а с другой стороны при этом мы облегчаем доступ к этим «кускам» в случае, когда нам это необходимо. Например, при отладке гидравлической схемы, состоящей из группы теплообменников, баков и насосной группы, у пользователя сложилось впечатление, что неправильно работает насосная группа. Легко перемещаясь между субмоделями пользователь сможет быстро проверить параметры данной группы, при необходимости подправить их, или понять, что проблема не в этом «куске», и отправиться на поиск в другой «кусок». Так же легко отслеживать параметры схемы: в этом случае мы лишь перемешаемся по субмоделям и отслеживаем соответствуют ли параметры схем заданным. В случае же, если бы схема была одна и большая, то пользователь был бы вынужден перемещаться по схеме, неизбежно теряя из поля зрения и восприятия различные ее области.

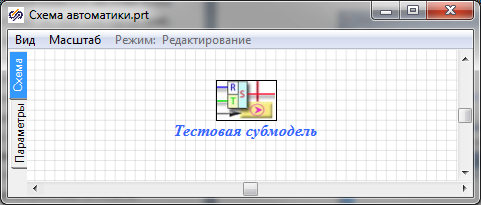
Важным преимуществом использования субмоделей в качестве разделителя схем является возможность использования закладок. Закладки представляют собой ссылки на листы, которые заданы пользователем. Т.к. при установке субмодели в проект, внутри субмодели создается новый лист (который при этом принадлежит проекту и не выходит за его пределы), то пользователь может закладкой создать ссылку на этот лист.

Еще одно важное преимущество работы с блоком «Субмодель» - возможность создания «древовидной» структуры проекта, реализуемой путем установки блока в уже существующем. Схема, представляющая принцип такой структуры, приведена на **Рисунке 1**.

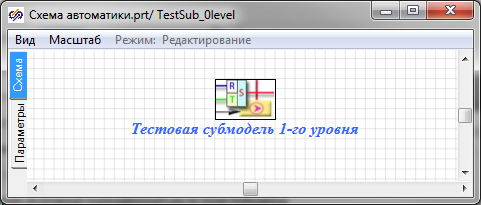


**Рисунок 1. Древовидная структура проекта**

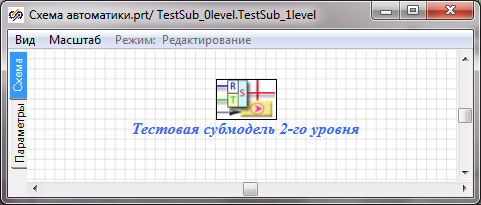
При этом количество вложенных друг в друга субмоделей неограниченно, т.е. можно создать «дерево» любого размера. Отображение уровня, на котором находится пользователь, можно отслеживать по изменению имени в заголовке окна (см. **Рисунки 2, 3, 4**). К заголовку проекта при переходе на субмодель более низкого уровня прибавляется имя субмодели уровня выше.



**Рисунок 2. Основной лист**

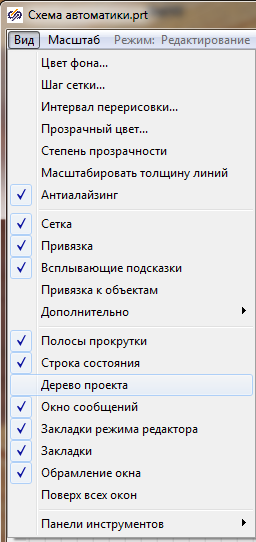


**Рисунок 3. Изменение имени заголовка при входе в «Тестовую субмодель»**

****

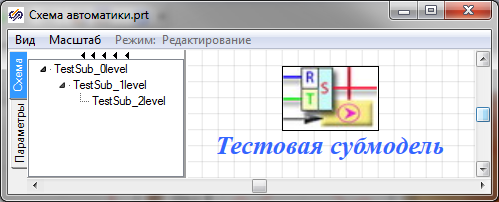
**Рисунок 4. Изменение имени заголовка при входе в «Тестовую субмодель 1-го уровня»**

При этом отображение уровней так же дублируется в дереве проекта. Дерево проекта позволяет увидеть все блоки, используемые в проекте. Для того чтобы его увидеть нужно зайти в меню «Вид» и активировать «Дерево проекта» (см. **Рисунок 5**).



**Рисунок 5. Активация режима «Дерево проекта»**

После этого в поле проекта слева появится дополнительное окно (см. **Рисунок 6**).

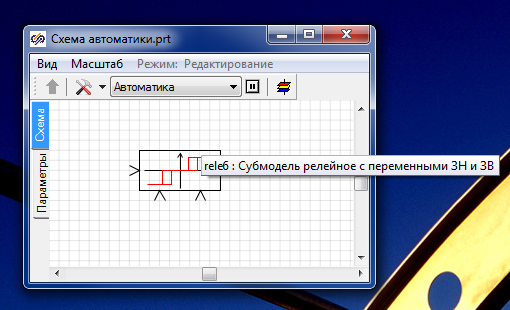


**Рисунок 6. Дерево проекта с раскрытой структурой субмоделей**

Для связи субмоделей, находящихся на разных уровнях, используются блоки «В память»/ «Из памяти». Они позволяют связывать между собой разные части проектов, не прибегая при этом к дополнительным сигналам проекта. Фактически эти блоки присваивают имена (метки) линиям связи.

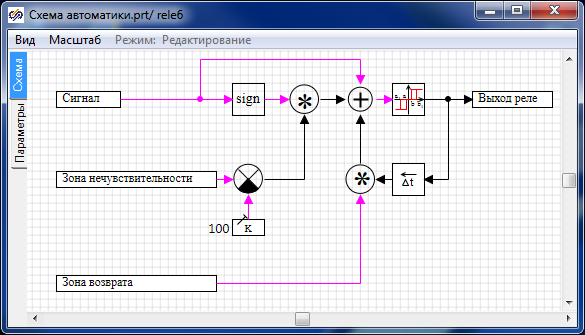
Второй вариант применения блока «Субмодель» - разработка на его основе блока, которого нет в стандартной библиотеке.

Иногда решение задачи, стоящей перед пользователем, приводит к тому, что необходимо создать новый блок, т.к. имеющиеся не решают поставленных целей (например, необходим блок с какой-то нестандартной анимацией) либо возникает ситуация, когда имеется алгоритм, который используется в различных местах схемы (или даже может использоваться в различных проектах). В этих случаях целесообразно создать блок, который может быть включен в библиотеку элементов под своим уникальным именем класса, для последующего использования пользователем либо группой пользователей. В качестве примера можно рассмотреть блок «Субмодель релейное с переменными ЗН и ЗВ» (подсистема автоматики, вкладка «Нелинейные»). Внешний вид данного блока приведен на **Рисунке 7**.



**Рисунок 7. Внешний вид блока «Субмодель релейное с переменными ЗН и ЗВ»**

Данный блок был создан на основе блока «Субмодель», в котором была реализована структурная схема работы, в этом можно убедиться, нажав двойным кликом левой кнопки мыши по блоку – см. **Рисунок 8**).



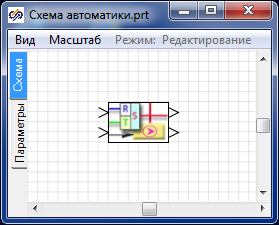
**Рисунок 8. Структурная схема работы блока «Субмодель релейное с переменными ЗН и ЗВ»**

Блок «Субмодель релейное с переменными ЗН и ЗВ» включен в состав стандартной библиотеки блоков автоматики, и, таким образом, может быть использован любым пользователем.

Данный подход к проектированию схем сильно упрощает процесс моделирования: достаточно всего лишь один раз отладить алгоритм или скрипт, управляющий, например, внешней анимацией субмодели, понять, какие параметры пользователь может изменять во внутреннем алгоритме (например, временные константы), вывести эти параметры в свойства субмодели, и дальше можно применять новый созданный блок там, где это необходимо.

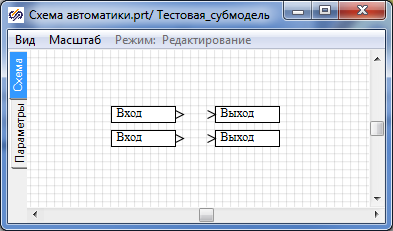
Использование блока «Субмодель» предоставляет широкие возможности пользователю по созданию нового блока. На основе субмодели могут быть созданы новые блоки с алгоритмами, которые могут обсчитывать как единичный сигнал, так и векторный (в зависимости от поставленной задачи), что дает возможность работать с массивом данных, причем размерность массива может быть изменяемой. Данная функция очень полезна при работе с категориями базы данных, так как дает возможность векторно обрабатывать сигналы какой-либо категории по одному или нескольким алгоритмам. Новый блок на основе субмодели можно привязать к стандартной подпрограмме – это дает возможность работать с одним и тем же блоком и обновлять его содержимое, в случае использования пользователями разных библиотек, обновление которых невозможно или нежелательно.

Связь блоков «Субмодель» между собой обеспечивается блоками «Порт входа»/«Порт выхода» и блоками «В память»/«Из памяти». Блоки «Порт входа»/«Порт выхода» обеспечивают появление входов и выходов в субмодели. Внешний вид субмодели с добавленными портами представлен на **Рисунке 9**.



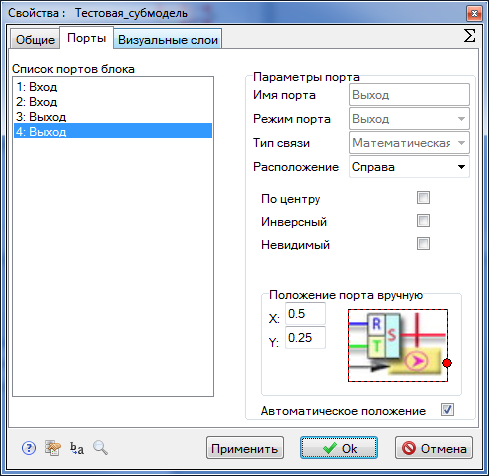
**Рисунок 9. Внешний вид субмодели с добавленными портами входа и выхода**

Для того, чтобы добавить порты в субмодель, нужно поставить на лист субмодели блоки «Порт входа»/«Порт выхода» (см. **Рисунок 10**).



**Рисунок 10. Внешний вид субмодели с добавленными портами входа и выхода**

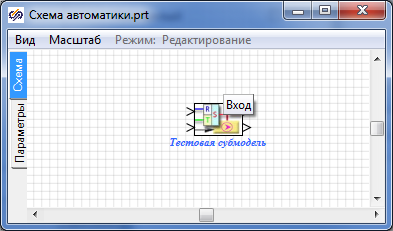
При этом установленные порты появляются на закладке «Порты» в свойствах субмодели (см. **Рисунок 11**).



**Рисунок 11. Закладка «Порты» с внесенными внутрь субмодели блоками «Порт входа» и «Порт выхода»**

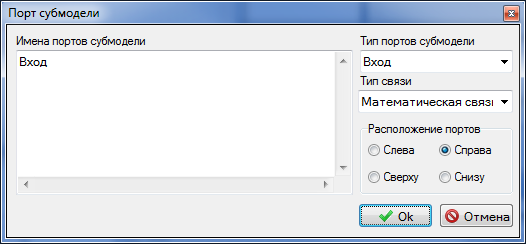
На данной закладке пользователь может по своему усмотрению изменять такие свойства портов, как расположение, видимость и пр. Данные свойства меняются у каждого конкретного порта.

При установке блоков «Порт входа»/«Порт выхода» имя порта субмодели не изменяется автоматически, поэтому нужно изменять это имя вручную, чтобы при подключении портов не возникло путаницы. Имя порта субмодели транслируется наружу субмодели, поэтому пользователь всегда может видеть с каким портом он связывается (см. **Рисунок 12**).



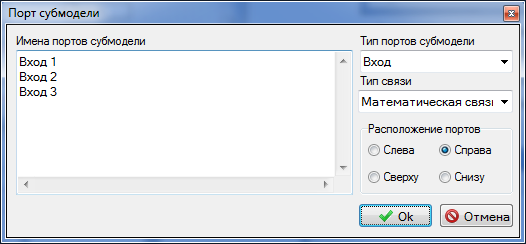
**Рисунок 12. Отображение имени порта снаружи субмодели**

## Изменить имя порта можно в свойствах блоков «Порт входа»/«Порт выхода». Для этого изменения нужно двойным кликом левой кнопки мыши вызвать окно изменения свойств (см. Рисунок 13) и в нем изменить на свое усмотрение имя и тип, а также расположение портов относительно блока.

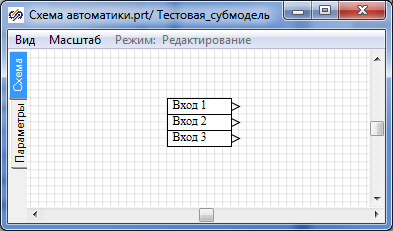


**Рисунок 13. Изменение имени порта**

В свойствах блоков «Порт входа»/«Порт выхода» можно так же дописывать имена входов, т.е. пользователю необязательно для заведения нового входа или нового выхода ставить блок «Порт входа»/«Порт выхода» (см. **Рисунок 14**). Внешний вид блока при этом изменится в соответствии с количеством добавленных портов (см. **Рисунок 15**).



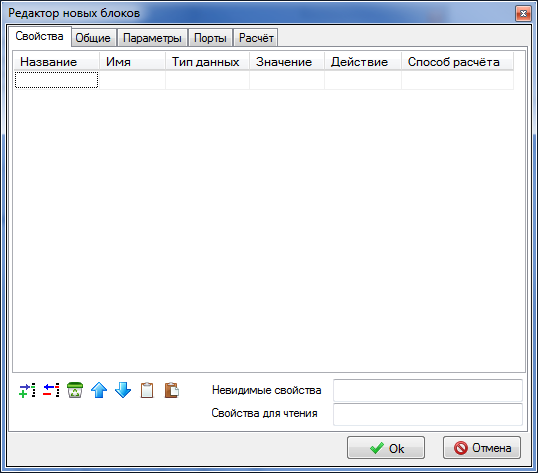
**Рисунок 14. Добавление портов**



**Рисунок 15. Внешний вид блока «Порт входа» после добавления портов**

При создании нового блока на основе субмодели у пользователя так же может возникнуть необходимость каким-либо образом задавать свойства блоков, находящихся в субмодели. Блок «Субмодель» позволяет реализовать это посредством создания новых свойств субмодели и последующей привязкой этих свойств к свойствам блоков, находящихся внутри субмодели, с помощью внутреннего языка программирования.

Для того, чтобы добавить новые свойства, необходимо выделить блок «Субмодель», а затем в главном меню пройти по пути «**Правка**» → «**Изменить блок**». После этого перед пользователем появится окно редактора блока (см. **Рисунок 16**).



**Рисунок 16. Окно редактора блоков**