

# Проект модуля “Виртуальный контроллер” подсистемы “Контроллеры” системы OpenSCADA.

Автор: [/Roman Savochenko](#)

## 1. Введение

Виртуальный контроллер представляет собой вычислительный блок с функциями вычисления по заданному алгоритму. Алгоритм вычисления может формироваться как по структурному(блочному) принципу так и на уровне языка программирования.

Виртуальный контроллер предназначен для выполнения следующих задач в системе OpenSCADA:

- Программы управления в контроллерах технологических процессов.
- Модели технологических, химических, физических и других процессов. А также реализация адаптивных механизмов и управления по моделям.
- Пользовательские программы управления внутренними функциями системы OpenSCADA её подсистемами и модулями.
- Гибкое формирование структуры параметров на уровне пользователя. Для создания параметров с нестандартной структурой и заполнение её по алгоритму пользователя.
- Вспомогательные вычисления.

## 2. Требования

Виртуальный контроллер должен удовлетворять следующим требованиям:

- Высокая надёжность. Поскольку система OpenSCADA позиционируется как среда исполнения контроллеров то блок вычисления должен иметь высокую надёжность и исключать фатальные сбои.
- Гибкая структура параметров виртуального контроллера. Подразумевает создание механизма формирования типов и структур параметров. Параметры должны группироваться в библиотеки параметров.
- Блочная структура (алгоблоки). Подразумевает создание блоков с функциями из библиотеки функций.
- Функциональная модульность блока вычислений. Подразумевает возможность расширения библиотеки функций внешними библиотеками функций.
- Допускать корректировку схемы блоков и её взаимосвязей в процессе выполнения.

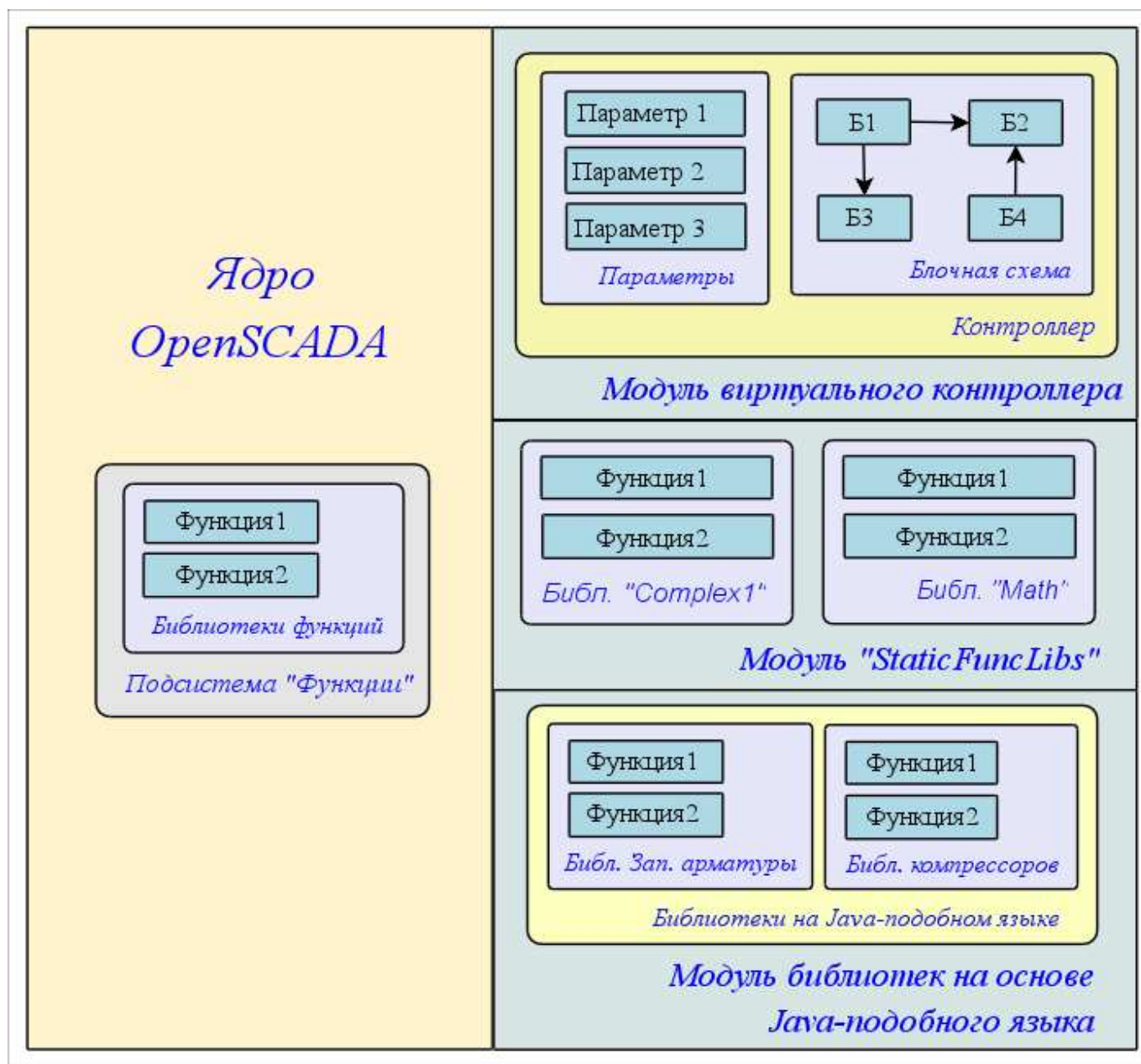
## 3. Архитектура

Модуль виртуального контроллера имеет структуру рисунок 1. Из структуры видно, что общими для модуля являются “Библиотека структур параметров” и “Библиотека функций”.

Библиотека структур параметров содержит группу доступных структур параметров. Под структурой параметра понимается конфигурация атрибутов которые будет содержать параметр. Механизм свободной манипуляции структурой параметра позволяет создавать параметры под любые требования.

Библиотека функций содержит функции доступные в виртуальном контроллере. Функции в библиотеке разбиты на группы функций. Группы функций группируют

функции по их назначению, например: математические, логические, пользовательского интерфейса, технологические, управление и т.д.



Структурно, каждый модуль типа “Контроллеры” содержит контроллеры. При этом контроллеры содержат множество параметров различных типов. Модуль виртуальных контроллеров, кроме параметров содержит основную блочную схему. На схеме располагается множество взаимосвязанных блоков. Блоки содержат функции вычисляемые с периодичностью контроллера и с учетом взаимосвязей между ними.

## 4. Блоки и Алгоблоки

Алгоритм вычисляемой программы строится по блочному принципу посредством размещения блоков на блочной схеме и формирования связи между блоками. Связи между блоками формируются через входы/выходы (I/O) блоков. Сам по себе блок не имеет I/O, однако приобретает их после назначения блоку функции из библиотеки функций. Сборка Блок-Функция называется алгоблоком.

Блоки поддерживают следующие типы связей:

- Межблочные. Подключение входа блока к выходу другого блока.

- Дальние межблочные. Соединение блоков из центральных блочных схем различных контроллеров.
- Коэффициент. Превращение входа в константу. Все входы/выходы по умолчанию инициализируются как константы.
- Внешний атрибут параметра.
- Векторные. Векторные связи являются виртуальными! На самом деле происходит связывание одноимённых I/O одного блока и другого. Используется для создания смысловых связей между блоками. Соединяться могут только одноимённые вектора.

## 5. Библиотеки функций и функции

Функции содержат предопределённый алгоритм действий над своими входами/выходами (I/O). Описание I/O содержит: строковый идентификатор, описание, тип и идентификатор вектора.

Поддерживаются типы I/O:

- целое (int);
- вещественное (double);
- строковое (string);
- логическое (bool);
- вектор (vector).

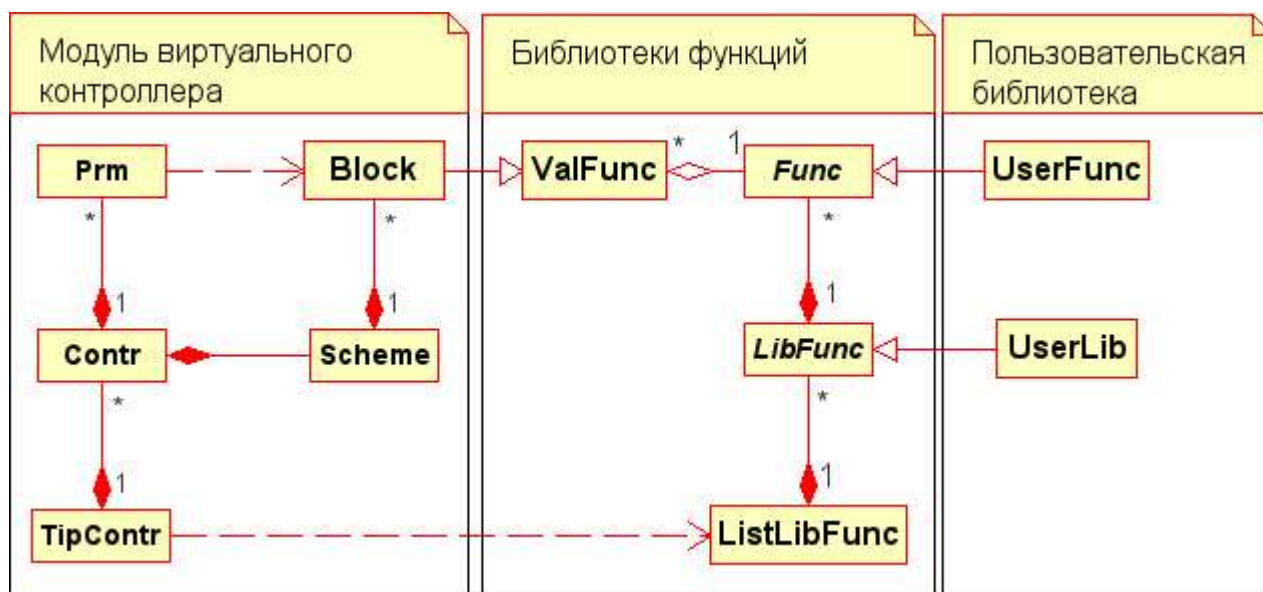
I/O, и даже сам вектор, могут принадлежать вектору. Принадлежность вектору означает скрывание I/O за смысловым содержанием вектора.

Функции группируются и распространяются в библиотеках функций. Библиотеки функций могут иметь различную природу и происхождение. Например планируется создание библиотек:

- `complex1` – библиотека функций импортированных из `Complex1`;
- `free` – библиотека свободных формул (механизм и язык описания из `Complex2`). Библиотека позволяет пользователю формировать новые функций. Описание функций выполняется на языке математической записи в одну строку.
- `block` – библиотека блочных схем. Библиотека позволяет пользователю формировать новые функций. Алгоритм формируется по блочному принципу, идентичному центральной блочной схемы.
- `math` – библиотека математических функций.

## 6. Реализация

Перед реализацией архитектуры модуля “Виртуальный контроллер” была написана модель будущего модуля. Статическая диаграмма классов приведена на рисунке



Описание объектов на диаграмме приведено в таблице

Имя	Описание
<i>TipContr</i>	Унаследован от класса системы TTipController. Является корневым классом модуля.
<i>Contr</i>	Унаследован от класса системы TController. Инкапсулирует функции отдельного экземпляра контроллера. Содержит класс Scheme с центральной блочной схемой контроллера.
<i>Prm</i>	Унаследован от класса системы TParamContr. Инкапсулирует функции отдельного параметра. Связывается с блоком или отдельным i/o блока на центральной блочной схеме контроллера.
<i>Scheme</i>	Класс центральной блочной схемы контроллера. Содержит центральную блочную схему, блоки и связи между блоками.
<i>Block</i>	Класс блока блочной схемы. Обязательно связан с библиотечной функцией и содержит текущие значения и связи i/o.
<i>ListLibFunc</i>	Класс контейнер, содержит список доступных библиотек функций. Физически будет располагаться в ядре системы.
<i>LibFunc</i>	Класс библиотеки функций. Является абстрактным и должен быть унаследован. Содержит группу функций. Может предоставлять механизм для создания новых функций пользователем (в потомке).
<i>Func</i>	Класс функции. Является абстрактным и должен быть унаследован. Содержит описание i/o. В потомке должен содержать реализацию алгоритма функции.
<i>ValFunc</i>	Класс значений функции. Содержит значения функции в соответствии со структурой IO класса Func. ValFunc передается в Func при выполнении алгоритма функции над указанными значениями.
<i>UserLib</i>	Класс реализации функций пользовательской библиотеки.
<i>UserFunc</i>	Класс реализации отдельной функций пользовательской библиотеки.