

# OpenSCADA 0.5.0

## Оглавление

<u>OpenSCADA 0.5.0</u> .....	1
<u>Введение</u> .....	1
<u>1 Добавлена полная поддержка архивирования значений</u> .....	2
<u>2 Переименована подсистема «Контроллеры» в “DAQ”. Реализована поддержка жесткого реального времени</u> .....	3
<u>3 Добавлена поддержка параметров логического уровня</u> .....	3
<u>4 Унифицирован механизм уведомления об ошибках через параметры контроллера</u> .....	4
<u>5 Добавлена поддержка параметров в модуль вычисления на основе Java-подобного языка "Java Like Calc" и модуль вычислений на основе блоков "Block Calc"</u> .....	4
<u>6 Полноценная поддержка сбора данных плат фирмы Diamond Systems в модуле DAQ/DiamondBoards</u> .....	5
<u>7 Значительно переработана подсистема «БД» и концепция доступа к БД. Предоставлен интерфейс доступа к БД прямо через SQL-запросы</u> .....	5
<u>8 Реконструирован интерфейс управления системой OpenSCADA</u> .....	5
<u>9 Создана объектная модель с пользовательскими функциями. Создано несколько библиотек функций для объектной модели системы</u> .....	6
<u>10 Добавлена среда визуализации Vision</u> .....	6
<u>12 Обновлено скрипты управления сборкой. Добавлена поддержки платформы x86_64</u> .....	6
<u>13 Планы дальнейшего развития</u> .....	6
<u>14 Замечания к релизу</u> .....	7

## Введение

Версию 0.5.0 системы OpenSCADA можно с уверенностью назвать переломной. Дело в том, что данная версия содержит ядро SCADA системы в котором реализованы все запроктированные базовые функции. Кроме этого, система практически вплотную приблизилась к реализации всех запроктированных базовых функций SCADA системы. Так, в дистрибутив системы была включена среда представления Vision Зайчука Евгения. Разработка которой, однако, ещё не завершена. Также, система дважды была использована для решения практических задач. Первой задачей стала – построение динамической модели технологического процесса газо-лифтной компрессорной станции (ГЛКС). Вторая задача представляла собой высокочастотный сбор и архивирование 6 каналов датчиков давлений и 2 канала перепада той-же компрессорной станции на основе одноплатного компьютера (PC/104) с УСО “Athena” фирмы Diamond Systems.

Данный документ является обработкой(компиляцией) документа <ChangeLog> системы OpenSCADA версии 0.5.0 призванный вкратце и доходчиво осветить новые возможности системы OpenSCADA. Детально ознакомиться с изменениями в системе OpenSCADA можно в файле `ChangeLog` из дистрибутива системы или здесь: <http://diyaorg.dp.ua/oscadawiki/wakka.php?wakka=Works/ChangeLog>

Ключевыми особенностями данной версии являются:

- Добавлена полная поддержка архивирования значений.
- Переименована подсистема «Контроллеры» в “DAQ”. Реализована поддержка жесткого реального времени.
- Добавлена поддержка параметров логического уровня.
- Унифицирован механизм уведомления об ошибках через параметры контроллера.
- Добавлена поддержка параметров в модуль вычисления на основе Java-подобного языка "[Java Like Calc](#)" и модуль вычислений на основе блоков «BlockCalc”.
- Полноценная поддержка сбора данных плат фирмы Diamond Systems в модуле DAQ/DiamondBoards.
- Значительно переработана подсистема «БД» и концепция доступа к БД. Предоставлен интерфейс доступа к БД прямо через SQL-запросы.
- Реконструирован интерфейс управления системой OpenSCADA.
- Создана объектная модель с пользовательскими функциями. Создано несколько библиотек функций для объектной модели системы.
- Добавлена среда визуализации Vision.
- Обновлено скрипты управления сборкой. Добавлена поддержки платформы x86\_64.

Новые и значительно обновленные модули:

- *DAQ/OperationSystem (1.2.0)* — Общая стабилизация модуля.
- *DAQ/JavaLikeCalc (0.9.5)* — Добавлена поддержка параметров в контроллере. Добавлена поддержка UTF8 в компиляторе.
- *DAQ/BlockCalc (0.9.0)* — Добавлена поддержка параметров в контроллер.
- *DAQ/DiamondBoards (0.9.0)* — Полноценная поддержка платы Athena от Diamond System.
- *DB/DBF (1.8.0)* — Исправлены досадные ошибки не проверки размера имени столбца (10) и размера значения ячейки (255). Добавлено кодирование/декодирование сообщений в кодировку БД.
- *DB/SQLite (1.2.0)* — Добавлено кодирование/декодирование сообщений в кодировку БД.
- *DB/MySQL (1.2.0)* — Добавлено кодирование/декодирование сообщений в кодировку БД.

- *UI/QTCfg* (1.2.0) — Множество дополнений и исправлений. Поддержка обновленной спецификации языка управления.
- *UI/WebCfg* (1.4.0) — Множество дополнений и исправлений. Поддержка обновленной спецификации языка управления.
- *Archive/BaseArh* (0.9.0) — Добавлена полная поддержка архивирования сообщений и значений на файловую систему с поддержкой механизма упаковки.
- *Special/SystemTests* (1.3.0) — Обновлено, исправлены и дополнены многие тесты. Также добавлены новые тесты: ValBuf, Archive и Base64Code.
- *Special/FLibComplex1* (0.9.0) — Добавлена. Библиотека функций совместимости со SCADA Complex1 фирмы НИП “DIYA”.
- *Special/FLibMath* (0.5.0) — Добавлена. Библиотека стандартных математических функций.
- *Special/FLibTime* (0.5.0) — Добавлена. Библиотека функций времени.

## 1 Добавлена полная поддержка архивирования значений.

С целью обеспечения архивирования значений в системе OpenSCADA, была реконструирована подсистема «Архивы». В результате была обеспечена поддержка архивирования как сообщений так и значений.

Архивирование значений строится из архивов и архиваторов. Архив – независимый элемент абстрактного хранилища состоящего из буфера значений и физических элементов архива распределённых между ответственными архиваторами. Архиватор – модульный механизм архивирования в физическое хранилище.

Архив может работать в трёх режимах:

- Пассивное архивирование из любого источника. Поток данных обеспечивает внешний источник.
- Пассивное архивирование атрибута параметра. Поток данных обеспечивает атрибут параметра.
- Активное архивирование атрибута параметра. Поток данных обеспечивает сам архиватор периодически навещая источник.

Отдельно взятый архиватор может иметь различную периодичность глубины и физическую основу хранилища, обеспечивая тем самым гибкость при выборе механизма архивирования.

В результате, архив может архивироваться многими архиваторами, предоставляя при этом простой механизм доступа как к данным в целом, так и данным отдельно взятого архиватора.

В модуль архивирования на файловую систему "[Base Arh](#)" была добавлена полная поддержка механизма архивирования значений в файлы. Кроме этого, обеспечен разумный подход в архивировании, позволивший значительно сократить размеры архивов. Подход заключается в двойной упаковке архивов. Первая упаковка реализует быстрый последовательный механизм заключающийся в сворачивании смежных точек с одинаковыми значениями в одно значение. Вторая упаковка подразумевает дожатие файла архива стандартным архиватором gzip.

Кроме этого, в модуле [Base Arh](#) реализован механизм упаковки данных четырёх стандартных типов системы OpenSCADA: логическое, целое, вещественное и строковое.

Также, в архиватор сообщений модуля [Base Arh](#) добавлена поддержка формата архива плоского текста и упаковки файлов архивов архиватором gzip.

## **2 Переименована подсистема «Контроллеры» в “DAQ”. Реализована поддержка жесткого реального времени.**

Подсистема «Контроллеры» получила более адекватное статусу название: “DAQ”, т.е. Data acquisition – сбор данных.

Во все задачи периодического опроса добавлена поддержка режима жесткого реального времени на основе POSIX функции `clock_nanosleep()`. Режим жесткого реального времени поддерживается только при запуске системы с привилегиями “root”. В процессе испытаний удалось достичь уверенной реакции не превышающей 2мс на ОС ALTLinux 3.0.

## **3 Добавлена поддержка параметров логического уровня.**

Для предоставления возможности пользователю гибко формировать параметры необходимой ему структуры, добавляя при этом послеобработку параметров физического уровня, была разработана и реализована подсистема параметров логического уровня.

Подсистема содержит параметры и шаблоны параметров логического уровня. На основе шаблона можно легко создавать параметры предопределённой шаблоном структуры и с предопределённой обработкой. В целом, параметр логического уровня может полностью отражать параметр физического уровня или строиться на основе шаблона, собирая в один параметр данные с нескольких физических. Шаблон, в свою очередь, формируется на основе функции из объектной модели системы OpenSCADA. А значит, такой функцией может быть и созданная пользователем функция реализации нужных алгоритмов.

## **4 Унифицирован механизм уведомления об ошибках через параметры контроллера.**

Поскольку параметр контроллера, практически, выступает в роли шлюза между физической средой и SCADA системой, то и уведомлять про состояние физической среды вынужден именно он. Унифицированный механизм уведомления состоит из двух частей.

Первая это уведомление доступности атрибута параметра. Возможна ситуация, когда часть атрибутов параметра доступна, а часть недоступна. При этом, в первую очередь нас интересует именно само состояние доступности. Для индикации о недоступности атрибута было введено понятие «Ошибка значения (EVAL)” которое присваивается атрибуту. Для каждого типа значения было зарезервировано собственное ошибочное значение: `EVAL_BOOL(2)`, `EVAL_INT(-2147483647)`, `EVAL_REAL(-3.3E308)`, `EVAL_STR(<EVAL>)`. Также, в механизм приведения была добавлена поддержка ошибочных значений. Хотя данный механизм и изымает из обращения крайние значения, он всё-же предпочтительнее поскольку идентификация доступности становится предельно простой, что несомненно важно для среды программирования и других подсистем.

Вторая часть это предоставление информации об ошибке в доступном виде. Для этой цели каждый параметр наделён атрибутом “err” строкового типа. Принят следующий механизм формирования этого сообщения:

- В случае отсутствия ошибки устанавливать атрибут в виде “0”.
- В случае ошибки устанавливать атрибут в виде "1:Сообщение ошибки". Где цифра это код ошибки.

Такой механизм уведомления позволяет совместить сразу три способа уведомления:

- Признак ошибки. Если опросить атрибут ошибки как логический признак, то мы получим признак возникновения ошибки.

- Код ошибки. Если опросить атрибут как целое, то мы получим код ошибки.
- Сообщение ошибки. Если опросить атрибут как строку, то получим сообщение об ошибке с кодом ошибки.

## **5 Добавлена поддержка параметров в модуль вычисления на основе Java-подобного языка "[Java Like Calc](#)" и модуль вычислений на основе блоков "[Block Calc](#)"**

С целью обмена вычисленными данными между системой OpenSCADA и вычислительными модулями была добавлена поддержка параметров в модули вычислителей [Java Like Calc](#) и [Block Calc](#).

Параметры выполняют отражение своих атрибутов на данные вычислителя. В случае с модулем "[Java Like Calc](#)", выполняется отражение параметров вычислительной функции контроллера на атрибут параметра контроллера. В модуле "[Block Calc](#)" выполняется отражение параметров блоков, блочной схемы контроллера, на атрибуты параметра этого-же контроллера. Суть отражения заключается в перенаправлении запроса к значению на внутреннюю структуру.

Этот механизм позволяет создавать параметры с закономерностью изменения значений в атрибутах заложенных пользователем.

## **6 Полноценная поддержка сбора данных плат фирмы Diamond Systems в модуле DAQ/DiamondBoards.**

Реализован полноценный однопоточный доступ к драйверу фирмы Diamond System с поддержкой сбора данных по прерыванию. Практически это первый пример покадрового сбора и архивирования данных реализованный в системе OpenSCADA. Метод позволяет достичь частоты опроса в 100кГц для платы Athena фирмы Diamond systems.

## **7 Значительно переработана подсистема «БД» и концепция доступа к БД. Предоставлен интерфейс доступа к БД прямо через SQL-запросы.**

В первую очередь был добавлен механизм доступа к нескольким записям для функции сканирования таблицы и удаления нескольких записей одним запросом. Данный механизм основан на конкретизации части ключевых полей с обобщением другой части полей. Так, для обобщения по отдельному ключевому полю нужно значение этого поля, при запросе, инициировать пустой строкой.

С целью расширения API подсистемы «БД» была добавлена функция доступа к БД посредством SQL-запросов. Естественно, эта функция работает только для БД поддерживающих SQL-запросы.

Однако, наиболее важным изменением стало создание механизма предварительной регистрации(открытия) нужных БД и последующей работы с ними. Этот механизм позволил скрыть особенности настройки БД (могут содержать и пароли) за её идентификатором. Также, механизм регистрации БД позволил раскрыть идею одновременного использования различных БД для суммарной загрузки системы. Например, можно выделять конфигурации отдельных контроллеров, сохранять их в собственной БД и распространять независимо. Этот пример вы можете увидеть в демонстрационном пакете системы OpenSCADA данной версии.

## **8 Реконструирован интерфейс управления системой OpenSCADA.**

С целью унификации адресации динамических узлов для низкоуровневого доступа и доступа через интерфейс управления был реконструирован интерфейс управления системой. В результате реконструкции появилась возможность прозрачного доступа по единому полному пути к любому узлу системы, как на низком уровне так и через интерфейс управления. Кодирование элементов пути было унифицировано, что позволило использовать в пути специальные символы.

Также, был пересмотрен механизм формирования информационной страницы, с целью повысить производительность и сделать этот процесс более надёжным.

Для более надёжного и безопасного приведения типов, при адресации, было внедрено динамическое приведение типов.

Важным приобретением, также, стала поддержка изображений языком интерфейса управления и включение библиотеки формирования изображения GD в систему. На основе этих возможностей были добавлены:

- иконки в интерфейс управления;
- построение графика значений архива в конфигуураторах;
- экспериментальное построение графиков в модуле WEB-интерфейса WebDbg.

## **9 Создана объектная модель с пользовательскими функциями. Создано несколько библиотек функций для объектной модели системы.**

Ранее существующая подсистема функции преобразована в объектную модель с функциями пользователей в роли методов объектов. Теперь, любой объект может предоставлять пользовательские функции для программирования на уровне системы OpenSCADA.

Данный релиз поставляется с четырьмя библиотеками функций для объектной модели системы OpenSCADA. Три из них выполнены в динамических библиотеках, а одна построена на основе Java-подобного языка модуля JavaLikeCalc. Этими библиотеками являются:

- Библиотека совместимости со SCADA Complex1 фирмы НИП “DIYA” (модуль OpenSCADA Special/FLibComplex1).
- Библиотека стандартных математических функций (модуль OpenSCADA Special/FLibMath).
- Библиотека функций времени (модуль OpenSCADA Special/FLibTime).
- Библиотека моделей аппаратов технологических процессов (файл БД:SQLite OscadaLibs.db с функциями на Java-подобном языке).

Используя эти библиотеки и вычислители модулей JavaLikeCalc и BlockCalc пользователь может строить свои вычислительные алгоритмы или создавать новые библиотеки и функции.

## **10 Добавлена среда визуализации Vision.**

В данный релиз включена первая реализация среды визуализации и управления на основе библиотеки QT – “Vision”. Среда всё ещё находится в разработке и не готова к практическому использованию. В тоже время, уже сейчас можно создавать кадры визуализации, наполняя их базовыми компонентами, с последующим их исполнением.

## 12 Обновлено скрипты управления сборкой. Добавлена поддержки платформы x86\_64.

С целью повышения управляемости процессом сборки системы OpenSCADA, система сборки была дополнена условиями позволяющими выборочно отключать сборку модулей.

На основе этого было добавлено условие для исключения из компиляции модуля DiamonBoards и условие выбора системной директории 64-разрядных библиотек на платформе x86\_64.

## 13 Планы дальнейшего развития

Для реализации в следующей версии можно сформулировать следующие важные задачи:

- Полный пересмотр интерфейса управления с выносом проверки прав доступа внутрь него. Построение на основе нового интерфейса управления транспортного протокола для распределённой архитектуры системы OpenSCADA.
- Внедрение JavaScript в модуль UI/WebCfg, добавление навигатора.
- Разработка общей концепции среды визуализации.
- Внедрение идеи построения параметров по шаблону в физический уровень.
- Добавление механизма полного резервирования параметров, включая и архивы, на логическом уровне.

## 14 Замечания к релизу

В процессе тестирования была подтверждена ошибка от [19.12.2005](#). Ошибка проявляется в связке NPTL+UI/QTcfg в момент останова системы OpenSCADA. Других проблем эта ошибка не вызывает. На момент данного релиза локализовать проблему не удалось.