

OpenSCADA 0.6.0

Оглавление

OpenSCADA 0.6.0	1
Введение	1
1 Концепция среды визуализации и управления (СВУ), а также визуализатор СВУ на основе библиотеки QT4	3
2 Значительное повышение стабильности, устойчивости и производительности системы	4
3 Унификация интерфейса управления и построение протокола взаимодействия на его основе	5
4 Унификация механизма построения параметров подсистемы «Сбор данных» по шаблонам	6
5 Расширение основного языка (JavaLikeCalc) и API уровня пользовательского программирования	6
6 Переход на библиотеку QT версии 4 в модулях, использующих эту библиотеку	6
7 Общие улучшения API ядра системы OpenSCADA	7
8 Общее улучшения сборочной системы, как следствие, упрощение и расширение поддерживаемых платформ	7
9 Новый сайт проекта OpenSCADA	8
10 Планы дальнейшего развития	8

Введение

Релиз открытой SCADA(Supervisory control and data acquisition) системы версии 0.6.0 позиционируется как концепт-релиз. Это связано с тем, что на данный момент, фактически, реализованы основные функции концепции SCADA системы, которые были сформулированы проектом OpenSCADA в 2003 году.

Основной целью данного релиза является предоставление сообществу пользователей и разработчиков свободного программного обеспечения (ПО) концепции и платформы для отработки решений построения комплексных SCADA-систем и других смежных решений, с целенаправленным совершенствованием и стабилизацией имеющихся компонентов, а также разработки нужных расширений и модулей.

Данный документ является обработкой(компиляцией) документа <ChangeLog> системы OpenSCADA версии 0.6.0, который призван вкратце и наглядно 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/Works/ChangeLog>.

Ключевыми особенностями данной версии являются:

- Концепция среды визуализации и управления (СВУ), а также визуализатор СВУ на основе библиотеки QT4.
- Значительное повышение стабильности, устойчивости и производительности системы.
- Унификация интерфейса управления и построение протокола взаимодействия на его основе.
- Унификация механизма построения параметров подсистемы «Сбор данных» по шаблонам.
- Расширение основного языка ([JavaLikeCalc](#)) и API уровня пользовательского

программирования.

- Переход на библиотеку QT версии 4 в модулях, использующих эту библиотеку.
- Общие улучшения API ядра системы OpenSCADA.
- Общее улучшения сборочной системы, как следствие, упрощение и расширение поддерживаемых платформ.
- Новый сайт проекта OpenSCADA.

Новые и обновленные модули:

- *Transport.Sockets (1.3.0)* — Общая стабилизация модуля.
- *Archive.FSArch (0.9.5)* — Оптимизирован архиватор значений. Добавлена архивация сообщений в формате плоского текста. Модуль переименован с «BaseArh» в «FSArch». Исправлено множество ошибок.
- *Archive.DBArch (0.5.0)* — Новый модуль архивирования сообщений и значений на БД.
- *DB.DBF (1.8.1)* — Общая стабилизация модуля.
- *DB.MySQL (1.3.0)* — Добавлена поддержка API запроса перечня доступных таблиц и структуры этих таблиц. Значительная стабилизация и оптимизация модуля.
- *DB.SQLite (1.3.0)* — Добавлена поддержка API запроса перечня доступных таблиц и структуры этих таблиц. Значительная стабилизация и оптимизация модуля.
- *DB.FireBird (0.6.0)* — Новый модуль поддержки СУБД «[FireBird](#)».
- *Protocol.HTTP (1.3.0)* — Значительная переработка и стабилизация модуля.
- *Protocol.SelfSystem (0.7.0)* — Новый модуль собственного протокола OpenSCADA, основанного на интерфейсе управления OpenSCADA.
- *Special.FLibComplex1 (1.0.0)* — Общая стабилизация модуля.
- *Special.FLibSYS (0.5.0)* — Результат объединения с модулем Special.FLibTime и работ по унификации пользовательского API работы с архивами.
- *Special.SystemTests (1.3.5)* — Обновлены некоторые тесты.
- *DAQ.DiamondBoards (1.0.0)* — Пересмотрен механизм сбора посредством прерываний, исправлено много ошибок, совместно с библиотекой API версии 5.91 исправлена проблема блокирования потоков и выпущен финальный релиз.
- *DAQ.System (1.5.0)* — Общая стабилизация модуля. Переименован с «OperationSystem».
- *DAQ.BlockCalc (1.0.0)* — Некоторые улучшения в механизме связывания, расширенная функция копирования контроллеров, блоков и параметров, исправления ошибок и финальный релиз.
- *DAQ.JavaLikeCalc (1.0.0)* — Расширена грамматика языка, добавлена поддержка механизма прекомпиляции пользовательских процедур и последующего их использования в разных частях OpenSCADA, исправления ошибок и финальный релиз.
- *DAQ.LogicLev (0.9.0)* — Вынесенный в модуль код подсистемы «Параметры» OpenSCADA 0.5.0 с расширением функционала и исправлением ошибок.
- *DAQ.CIF (0.9.0)* — Новый модуль источника данных контроллеров фирмы Siemens серии S7 (300,400) посредством сети ProfiBUS, протокола MPI и коммуникационного процессора фирмы Hilcher CIF50PB.
- *DAQ.SNMP (0.3.0)* — Новый модуль базовой поддержки протокола Simple Network Management Protocol в режиме чтения.
- *DAQ.ModBus (0.4.0)* — Новый модуль поддержки источников данных, доступных по протоколу ModBus.
- *DAQ.Transporter (0.3.0)* — Новый модуль поддержки механизма отражения данных подсистемы «Сбор данных» удалённых OpenSCADA станций на локальные.
- *UI.QTStarter (1.5.0)* — Переход на библиотеку QT версии 4, общая стабилизация и исключение внешних вызовов функций QT.
- *UI.QTCfg (1.5.0)* — Переход на библиотеку QT версии 4, поддержка изображений в интерфейсе управления, общая стабилизация и оптимизация.
- *UI.WebCfg (1.5.0)* — Общая стабилизация модуля.
- *UI.VCAEngine (0.5.0)* — Новый модуль движка среды визуализации и управления (СВУ).
- *UI.Vision (0.5.0)* — Новый модуль визуализатора среды визуализации и управления

(СВУ) на библиотеке QT.

- *UI.WebVision (0.2.0)* — Новый модуль визуализатора среды визуализации и управления (СВУ) на основе WEB-технологий (XHTML, JavaScript, CSS, AJAX).

1 Концепция среды визуализации и управления (СВУ), а также визуализатор СВУ на основе библиотеки QT4.

Основным нововведением релиза 0.6.0 системы OpenSCADA стала базовая реализация гибкой среды визуализации и управления (СВУ). СВУ OpenSCADA построена по концепции «модель данных» – «вид», где «модель данных» описывает структуру и вычисления визуального интерфейса (ВИ), а «вид» реализует непосредственное представление и контакт с пользователем. Такой подход позволяет реализовать визуализацию с использованием различных графических библиотек и окружений пользовательского интерфейса, в пределах единого конфигурационного описания структуры пользовательского интерфейса. Реализована поддержка режима разработки и исполнения пользовательского интерфейса на основе известной графической библиотеки QT4. На основе WEB-технологий создана базовая реализация визуализатора с поддержкой режима исполнения (RunTime).

Концепцией заложена возможность формирования широкого диапазона интерфейсов пользователя различной сложности. В основу всех этих интерфейсов положен узкий набор базовых примитивов. Комбинируя эти примитивы можно создавать как производные кадры пользовательских элементов визуализации, так и страницы конечных интерфейсов визуализации. Определён следующий набор базовых примитивов:

- *ElFigure* — Элементарные графические фигуры.
- *FormEl* — Элементы формы.
- *Text* — Текст.
- *Media* — Медиа.
- *Diagram* — Диаграмма.
- *Protocol* — Протокол.
- *Document* — Документ.
- *Function* — Функция API объектной модели OpenSCADA.
- *Box* — Контейнер.
- *Link* — Связующая линия.

Из числа перечисленных примитивов в данном релизе остались не реализованными: “Document”, “Function” и “Link”. Их реализация запланирована после выхода версии 0.6.0.

Реализуемый проект СВУ предусматривает поддержку следующих функций:

- три уровня сложности в формировании интерфейса визуализации, позволяющие органично осваивать и применять инструментарий по методике от простого к сложному:
 - 1) формирование из шаблонных кадров, путём назначения динамики (возможно и без графической конфигурации);
 - 2) графическое формирование новых кадров, путём использования готовых элементов визуализации из библиотеки(мнемосхемы);
 - 3) формирование новых кадров, шаблонных кадров и элементов отображение в библиотеки.
- построение интерфейсов визуализации разной сложности, начиная от простых плоских интерфейсов мониторинга и заканчивая полноценными иерархическими интерфейсами, используемыми в SCADA системах;
- предоставление различных способов формирования и конфигурации пользовательского интерфейса, основанных на различных интерфейсах графического представления (QT, Web, Java ...), или-же посредством стандартного интерфейса управления системой OpenSCADA;
- смену динамики в процессе исполнения;
- построение новых шаблонных кадров на уровне пользователя и формирование

специализированных, под область применения, библиотек кадров (например включение кадров параметров, графиков и других элементов с увязкой их друг с другом), в соответствии с теорией вторичного использования и накопления;

- построение новых пользовательских элементов визуализации и формирование специализированных под область применения библиотек кадров, в соответствии с теорией вторичного использования и накопления;
- описание логики новых шаблонных кадров и пользовательских элементов визуализации как простыми связями, так и лаконичным, полноценным языком пользовательского программирования;
- возможность включения в пользовательские элементы визуализации функций (или кадров вычисления функций) объектной модели OpenSCADA, практически связывая представление с алгоритмом вычисления (например, визуализируя библиотеку моделей аппаратов ТП, для последующего визуального построения моделей ТП);
- разделение данных пользовательских интерфейсов и интерфейсов представления этих данных, позволяющее строить интерфейс пользователя в одной среде, а исполнять во многих других (QT, Web, Java ...);
- возможность подключения к исполняющемуся интерфейсу, для наблюдения и коррекции действий (например, при обучении операторов и контроля в реальном времени за его действиями);
- визуальное построение различных схем с наложением логических связей и последующим централизованным исполнением в фоне (визуальное построение и исполнение математических моделей, логических схем, релейных схем и иных процедур);
- предоставление функций объектного API в систему OpenSCADA, может использоваться для управления свойствами интерфейса визуализации из пользовательских процедур;
- построение серверов кадров, элементов визуализации и проектов интерфейсов визуализации с возможностью обслуживания множественных клиентских соединений;
- простая организация клиентских станций на различной основе (QT, Web, Java ...) с подключением к центральному серверу;
- полноценный механизм разделения полномочий между пользователями, позволяющий создавать и исполнять проекты с различными правами доступа к его компонентам;
- гибкое формирование правил сигнализаций и уведомления с учётом и поддержкой различных способов уведомления;
- поддержка пользовательского формирования палитры и шрифтовых предпочтений для интерфейса визуализации (темы);
- поддержка пользовательского формирования карт событий под различное оборудование управления и пользовательские предпочтения;
- поддержка профилей пользователей, позволяющая определять различные свойства интерфейса визуализации (цветовая гамма, шрифтовые особенности, предпочтительные карты событий);
- гибкое хранение и распространение библиотек виджетов, кадров и проектов интерфейсов визуализации в БД, поддерживаемых системой OpenSCADA; практически пользователю нужно только зарегистрировать полученную БД с данными.

Более детально ознакомиться с проектом концепции СВУ можно по ссылке: <http://diyaorg.dp.ua/oscadawiki/Doc/KoncepcijaSredyVizualizacii>

2 Значительное повышение стабильности, устойчивости и производительности системы.

В процессе работ над версией 0.6.0 было выполнено ряд мероприятий по общей стабилизации системы OpenSCADA, а также по повышению производительности системы. Перечислим наиболее важные из них:

- Исправлена глобальная проблема недостаточного покрытия ключевых переменных

ресурсами. Проблема отчётливо выявила себя на многоядерных архитектурах.

- Исключена возможность прямого вызова QT-методов из не QT-потокот в всех QT-модулях. Приводило к падению OpenSCADA при остановке.
- Механизм захвата ресурсов пересмотрен, исправлен для работы в многоядерных системах и оптимизирован. В результате повысилась общая стабильность и производительность.
- Исправлены функции TSYS::addr2str() и TSYS::str2addr(). Приводило к непонятным падениям модуля QTCfg.
- Внедрение ассоциативных контейнеров (map и hash_map) в наиболее ответственные узлы OpenSCADA. Значительно повысило скорость доступа к элементам больших массивов, например:
 - при доступе к динамическим объектам, динамического дерева, во время разбора полного пути динамического объекта;
 - при работе со свойствами элементов конфигурации и таблиц БД.
- Систематизирован и унифицирован интерфейс управления системой и его язык. Повысило функциональные качества и реактивность интерфейсов и протоколов построенных на нём.
- Добавлена возможность адресации отдельных элементов записи таблиц БД. Повысило производительность доступа к БД путём запроса только нужных элементов записи.
- Добавлен механизм запоминания смещения в строке, для основных функций, при разборе пути или адреса с разделителем. Исключило геометрический рост затрат времени на разбор строки от количества элементов в ней.
- Оптимизация реализации запросов в модулях БД: [SQLite](#) и [MySQL](#). Значительно повысило эффективность доступа к БД этих типов.
- Оптимизирован механизм проверки прав доступа. Позволило снизить время проверки прав доступа.

3 Унификация интерфейса управления и построение протокола взаимодействия на его основе.

В данной версии произведена унификация интерфейса управления OpenSCADA в направлении формализации языка запросов, упрощения формирования информационной части и обработки командных запросов, а также с целью упрощения формирования транспортных протоколов на основе данного интерфейса. Результатом данной унификации стало повышение общей производительности запросов по данному интерфейсу, а так-же создание транспортного протокола [SelfSystem](#). За основу этого протокола взяты запросы к интерфейсу управления OpenSCADA, обёрнутые в тонкий слой управляющего протокола.

Посредством данного протокола реализовано взаимодействие между различными станциями OpenSCADA в сети. Например, модуль конфигурации на библиотеке QT, [QTCfg](#), получил возможность конфигурации удалённых станций, позволяя строить распределённые системы с единым конфигурационным центром. Визуализатор пользовательских интерфейсов на основе библиотеки QT, [Vision](#), также, получил возможность формирования пользовательского интерфейса и исполнение сеансов пользовательских интерфейсов, расположенных на удалённых OpenSCADA станциях, формируя тем самым конфигурацию с сервером визуализации. Ко всему этому был реализован первый модуль отражения данных удалённых станций OpenSCADA на локальных, им стал модуль отражения данных подсистемы «Сбор данных». В последствии планируется создание таких, транспортных, модулей для остальных модульных подсистем.

А следом за данными решениями был добавлен механизм сервисных функций интерфейса управления. Данный механизм позволяет получить доступ к данным со сложным форматированием и данным больших объёмов, в приоритетном режиме запроса и оптимальной для передачи форме.

4 Унификация механизма построения параметров подсистемы «Сбор данных» по шаблонам.

В версию 0.5.0 был внедрён механизм формирования параметров подсистемы «Сбор данных» с нужной пользователю структурой, т.е. по шаблону. Данный механизм был реализован в виде отдельной подсистемы «Параметры». В данном релизе идея формирования параметров подсистемы «Сбор данных» по шаблонам была расширена на все остальные модули подсистемы «Сбор данных». При этом ранее существующая подсистема «Параметры» была преобразована в равноценный модуль логического уровня параметров «[LogicLev](#)», а шаблоны стало возможным формировать на уровне подсистемы «Сбор данных», группируя их по библиотекам шаблонов. Фактически любой модуль, предоставляющий доступ к данным высокоинтеллектуальных источников, может включать поддержку параметров, формируемых по шаблону, внося свой смысл в ссылку внутреннего уровня. Примером такого модуля стал модуль доступа к данным контроллеров фирмы Siemens серии S7 по протоколу MPI сети ProfiBUS [CIF](#).

Шаблоны параметров стали формироваться независимо от существующих функций, путём формирования собственной функции обработки параметров и последующей её компиляции в модуле, предоставляющем механизм компиляции языка пользовательского программирования. Шаблон может быть сформирован вообще без пользовательской процедуры, если нужно только пассивное, выборочное отражение параметров.

5 Расширение основного языка ([JavaLikeCalc](#)) и API уровня пользовательского программирования.

Компилирующий интерпретатор Java-подобного языка вместе с библиотеками функций пользовательского программирования фиксированного типа (API) формируют среду пользовательского программирования системы OpenSCADA.

В данном релизе Java-подобный язык получил:

- полноценную поддержку циклов `while` и `for`;
- механизм прерывания зациквившихся процедур;
- операции префиксного и постфиксного инкремента и декремента;
- механизм прекомпиляции пользовательских программ различных подсистем;
- прямой механизм адресации параметров подсистемы «Сбор данных».

API функций пользовательского программирования было унифицировано и определяется библиотеками функций:

- *FLibSYS* — Библиотека системного API, на момент данного релиза содержит функции для работы с временем и датой, а также архивами сообщений и значений.
- *FLibMath* — Библиотека математических функций.
- *FLibComplex1* — Библиотека функций совместимости со SCADA «Complex1».

6 Переход на библиотеку QT версии 4 в модулях, использующих эту библиотеку.

В виду выхода новой версии библиотеки QT (версия 4) и повсеместного её распространения, а также начала работ над средой визуализации и управления (СВУ), решено было перевести существующие модули, основанные на QT, на новую версию.

Такая работа была выполнена и модули подсистемы «Пользовательские интерфейсы»: [QTCfg](#), [QTStarter](#) и [Vision](#) функционируют на основе библиотеки QT версии 4. В виду наличия в ранних версиях определённых проблем и отсутствия некоторых функций рекомендуется использовать версию библиотеки не менее 4.3.

В результате перехода на библиотеку QT4 удалось добиться большей стабильности и расширить функциональные характеристики модулей её использующих.

7 Общие улучшения API ядра системы OpenSCADA.

В процессе работ над данным релизом были внесены определённые изменения в API системы OpenSCADA. В большей мере это связано с процессами стабилизации, оптимизации, фиксации ошибок и общим улучшением API и в меньшей степени с добавлением новых функций. Перечислим наиболее значительные изменения:

- Унификация API интерфейса управления OpenSCADA. Подробнее об этом в разделе 3.
- В подсистеме безопасности, у пользователей и групп пользователей, устранена индексация пользователей и групп пользователей. Также, устранена группа пользователя по умолчанию. Выполнено с целью устранения двойной идентификации, поскольку присутствует быстрая идентификация по символьному идентификатору/имени.
- Упразднена подсистема «Параметры», а её функции вынесены в модуль логического уровня «[LogicLev](#)» подсистемы «Сбор данных». Шаблоны параметров сгруппированы по библиотекам и размещены в подсистеме «Сбор данных». Подробнее об этом в разделе 4.
- Добавлен механизм компиляции процедур в модулях подсистемы «Сбор данных», для использования пользовательского программирования в различных узлах OpenSCADA.
- Добавлен шаблон функции локализации сообщений `_()` с целью унификации, повышения читабельности исходного текста и уменьшения насыщенности.
- Добавлены шаблоны функций для формирования сообщений в OpenSCADA и повышения читабельности исходного текста, а так-же уменьшения насыщенности. Добавлены шаблоны функций: `message()`, `mess_err()`, `mess_info()` и подобные.
- Унифицированы точки состояний контроллеров подсистемы. Теперь состояние «Включен» подразумевает установку всех связей, а «Запущен» только фактический сбор данных.
- Практически все функции установки параметров переименованы к виду `setMethod()`.
- Добавлена поддержка атрибута видимости полей записи БД. Это позволяет повысить производительность работы с БД за счёт получения только нужных данных.
- Добавлена поддержка глобального кода ошибки в объекте `TErr`. Используется для избирательной обработки ошибок.
- Механизм захвата ресурсов пересмотрен, исправлен для работы в многоядерных системах и оптимизирован.
- Добавлен механизм предварительной инициализации создаваемых архивов значений в модулях сбора данных. Удалось добиться исключения дополнительной настройки архива при конфигурации атрибутов параметров.
- Добавлена поддержка механизмов запроса перечня доступных в БД таблиц и их структуры, а также редактирования их содержимого посредством интерфейса управления и конфигураторов OpenSCADA.
- Реализована поддержка исходящих протоколов. Внутренний интерфейс протокола реализован потоковым, со специализированной для протокола структурой.

Детальнее ознакомиться с API данной версии можно в соответствующем документе «[API системы OpenSCADA](#)».

8 Общее улучшения сборочной системы, как следствие, упрощение и расширение поддерживаемых платформ.

С целью реализации функции полноценного, раннего, уведомления об отсутствии нужных для сборки зависимостей сценарий проверки окружения был дополнен кодом проверки основных зависимостей сборки. В результате этого, уведомления о всех неудовлетворённых зависимостях будут выданы на стадии конфигурации сборочной системы, что ускоряет разрешение зависимостей

или принятие решения об отключения сборки того или иного модуля.

Формат документации, размещённой в дереве исходного текста и тарболе проекта, был изменён на pdf, в виду простой манипуляции с ним на стадии установки.

9 Новый сайт проекта OpenSCADA.

В процессе работы над данным релизом, для проекта OpenSCADA был создан новый сайт ( <http://oscada.diyaorg.dp.ua>), основанный на CMS-движке  [Туро3](#). Сайт является многоязычным и содержит основную информацию о проекте OpenSCADA, сохраняя большой потенциал по расширению. Для этого сайта были выполнены переводы основных информационных материалов на Украинский и Английский языки.

Запланировано добавление следующих функций на главном сайте: форума, галереи скриншотов и других медиа-материалов, почтовой рассылки, трекера изменений и работ ведущихся по проекту.

10 Планы дальнейшего развития

Планируются последующие, частые релизы в пределах версии 0.6.0, с целью оперативного отслеживания процесса прикладной адаптации и стабилизации. В финале планируется выпуск стабильной-промышленной версии 0.7.0 для платформы Linux x86, x86-64.

Для реализации к версии 0.7.0, по промежуточным версиям 0.6.0, можно определить следующие важные задачи:

- Web-визуализатор СВУ.
- Модульный механизм уведомлений и сигнализации.
- Разработка и реализация примитивов СВУ: Документ, Функция и связь.
- Реализация механизма: «Темы оформления» и «Карты событий» для СВУ.
- Расширение перечня поддерживаемых источников данных.