

OpenSCADA 0.7.0

Оглавление

OpenSCADA 0.7.0	1
Введение	1
1 Реализация плановых задач	3
2 Оптимизация, повышение стабильности, устойчивости и производительности системы	3
3 Усовершенствование и стабилизация СВУ	5
4 Формирование и стабилизация API пользовательского программирования	6
5 Общесистемные расширения	7
6 Значительные улучшения и исправления отдельных модулей	8
7 Формирование документации и переводы документации и интерфейса	9
8 Унификация демонстрационной БД	9
9 Решения, построенные на основе OpenSCADA	9
Заключение	10
Ссылки	10

Введение

Релиз открытой SCADA(Supervisory control and data acquisition) системы версии 0.7.0 является первым стабильным промышленным релизом, что связано с завершением стадии бета-тестирования, начатой с версии 0.6.4.

Основной целью данного релиза является предоставление сообществу пользователей и разработчиков свободного программного обеспечения (ПО) платформы для построения решений комплексных систем автоматизации и других смежных решений, а также предоставления коммерческих услуг на основе проекта OpenSCADA.

Данный релиз является первым стабильным релизом, для которого предоставляется техническая поддержка от разработчиков и для которого планируется выпуск исправлений в течении продолжительного времени.

Данный документ является обработкой(компиляцией) документа "ChangeLog" системы OpenSCADA версии 0.7.0, 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://wiki.oscada.org/Works/ChangeLog>.

Ключевыми особенностями данной версии являются:

- Реализация плановых задач.
- Оптимизация, повышение стабильности, устойчивости и производительности системы.
- Усовершенствование и стабилизация СВУ.
- Формирование и стабилизация API пользовательского программирования.
- Общесистемные расширения.
- Значительные улучшения и исправления отдельных модулей.
- Формирование документации и переводы документации и интерфейса.
- Унификация демонстрационной БД.

Новые и обновленные модули:

- Archive.FSArch - 1.3.1 -> 1.4.1, исправления и улучшения для повышения производительности
- Archive.DBArch - 0.9.0 -> 0.9.2, исправления и некоторые улучшения
- DB.DBF - 2.0.1 -> 2.0.2, исправления
- DB.MySQL - 1.6.1 -> 1.6.2, некоторые улучшения
- DB.SQLite - 1.6.1 -> 1.6.2, улучшения
- DB.FireBird - 0.9.1 -> 0.9.5, значительные улучшения
- DB.PostgreSQL - 0.9.0, новый модуль
- DAQ.DiamondBoards - 1.2.0 -> 1.2.1, адаптация к изменению API
- DAQ.System - 1.7.0 -> 1.7.2, очистка кода и адаптация к изменению API
- DAQ.BlockCalc - 1.2.1 -> 1.4.0, улучшения и исправления
- DAQ.JavaLikeCalc - 1.5.0 -> 1.8.0, значительные улучшения, исправления и оптимизация
- DAQ.LogicLev - 1.0.1 -> 1.1.2, улучшения и исправления
- DAQ.SNMP - 0.4.0 -> 0.4.1, адаптация к изменению API
- DAQ.Siemens - 1.2.1 -> 1.2.3, исправления и адаптация к изменению API
- DAQ.ModBus - 1.0.1 -> 1.1.1, исправления и улучшения
- DAQ.DCON - 0.3.0 -> 0.3.3, исправления и улучшения
- DAQ.ICP_DAS - 0.5.0 -> 0.7.2, значительные улучшения и исправления
- DAQ.DAQGate - 0.8.0 -> 0.9.1, значительные улучшения и исправления
- DAQ.SoundCard - 0.6.0 -> 0.6.1, адаптация к изменению API
- DAQ.OPC-UA - 0.6.0, новый модуль
- Transport.Sockets - 1.4.0 -> 1.4.5, улучшения и исправления
- Transport.SSL - 0.8.2 -> 0.9.5, улучшения и исправления
- Transport.Serial - 0.5.0 -> 0.7.1, значительные улучшения и исправления
- Protocol.HTTP - 1.4.1 -> 1.5.0, значительные улучшения
- Protocol.SelfSystem - 0.9.1 -> 0.9.3, исправления и адаптация к изменению API
- Protocol.UserProtocol - 0.6.0, новый модуль
- Protocol.ModBus - 0.5.0 -> 0.6.1, исправления и улучшения
- Protocol.OPC-UA - 0.6.0, новый модуль
- Special.FLibComplex1 - 1.0.4 -> 1.0.6, исправления и улучшения
- Special.FLibMath - 0.5.1 -> 0.5.2, адаптация к изменению API
- Special.FLibSYS - 0.9.0 -> 0.9.2, исправления и улучшения
- Special.SystemTests - 1.3.7 -> 1.5.0, переработка и унификация тестов
- UI.QTStarter - 1.5.3 -> 1.6.0, улучшения
- UI.QTCfg - 1.8.1 -> 1.9.0, значительные улучшения и исправления
- UI.WebCfg - 1.5.3 -> 1.5.4, исправления и адаптация к изменению API
- UI.WebCfgD - 0.6.2 -> 0.6.5, исправления
- UI.VCAEngine - 0.9.0 -> 1.0.0, значительные улучшения, исправления и оптимизация
- UI.Vision - 0.9.0 -> 1.0.0, значительные улучшения, исправления и оптимизация
- UI.WebVision - 0.7.0 -> 0.9.0, значительные улучшения и исправления
- UI.WebUser - 0.6.0, новый модуль

1 Реализация плановых задач

В соответствии с планом релиза были выполнены следующие задачи:

- *Реализация примитивов "Связь" и "Функция" СВУ.* - Ввиду непринципиальности для промышленного релиза и неостребованности на данный момент эта задача была вынесена за пределы первого промышленного релиза.
- *Стабилизация и выпуск промышленной ветки OpenSCADA.* - Прделана большая работа по стабилизации и оптимизации промышленного релиза, о чём подробнее в следующем разделе.
- *Реализация модуля БД ОДБС, PostgreSQL и транспортного модуля БД.* - Максимом Лысенко была выполнена реализация модуля БД "PostgreSQL". Реализация модуля БД "ODBC" была признана нецелесообразной, поскольку технология "ODBC" фактически является интерфейсом SQL-запросов, который не учитывает особенностей SQL-диалектов реальных БД, а значит не может прямо, без оглядки на тип БД, использоваться в OpenSCADA при решении её основных задач. Перспектива использования "ODBC" остаётся для реализации доступа к БД, у которых нет "родного" API прямого доступа к БД, но есть драйвер ODBC, или для предоставления простого интерфейса SQL-запросов в OpenSCADA. Реализация транспортного модуля БД была вынесена за пределы первого промышленного релиза.
- *Реализация нового механизма/модуля тестов системы OpenSCADA.* - Для выполнения внутрисистемных тестов OpenSCADA механизм модуля тестирования был реализован в виде функций пользовательского API, что с одной стороны сохранило возможность автономного вызова статичных тестов и с другой стороны предоставило возможность вызова тестов по одному, вручную, а также из пользовательских скриптов в любом порядке и в нужном количестве отдельных задач.

2 Оптимизация, повышение стабильности, устойчивости и производительности системы.

В процессе работ над данной версией, а также её практической адаптации, было обнаружено и исправлено в общей сложности около 200 ошибок.

Кроме исправления множества ошибок была проделана работа по оптимизации различных компонентов OpenSCADA. Особенно заметной оптимизации по-прежнему подверглась среда визуализации и управления (СВУ) и её визуализаторы.

Перечислим наиболее существенные ошибки, исправление которых значительно отразилось на повышении стабильности:

- *Общесистемные:*
 - Исправлена некорректная установка политики реального времени для потоков. Не был установлен флаг потока PTHREAD_EXPLICIT_SCHED.
 - Исправлен контроль за двойным захватом ресурса в одном потоке и захват ресурса по таймауту.
 - Механизм предотвращения многократного запуска исправлен на предмет проверки PID заблокированного процесса по причине удаления системой lock-файлов из директории /tmp.
 - Отключена упаковка данных структуры классов с помощью #pragma pack(push,1) по причине появления неявной ошибки с зависанием одного из потоков и блокирования остальных в прошивке OpenSCADA для ПЛК LP8x81.
 - Код инициализации и обновления модулей обновлен для исключения двойной инициализации разделяемых библиотек.
 - Функция создания задачи исправлена для предотвращения падения при создании автоматически закрываемых задач при быстром закрытии.
 - Исправлено зависание функции stop() в случае указания пустого расписания.
- *Транспорты:*

- *Serial, DAQ.DCON, DAQ.ModBus*: Транспорт последовательного интерфейса оптимизирован, сделан типовым и исправлен на предмет нескольких ошибок. Клиенты транспорта адаптированы для использования нового механизма.
- *Sockets, SSL*: Исправлена возможность неполной записи функцией `write()`. При записи в сокет эта функция пишет только часть размером в свой буфер (50кБ). Для полной записи больших сообщений добавлен повтор записи.
- *Transport.SSL*: Окончательно исправлено ожидание данных ответа сервера путём доживания обработки данных в функции `BIO_read()`.
- *Transport.Sockets*: Исправлена обработка нулевого результата ожидания хвоста сообщений для исключения зависания на запросах в `DAQ.System.HDDTemp`.
- *Transport.Serial*: Исправлено использование флага "CSTOPB" для указания количества стоп-битов.
- *Сбор данных*:
 - *ModBus*: Исправлена потеря ресурса во время сбора данных. По этой причине исключена возможность присвоения строки прямо к объекту ресурсной строки.
 - *JavaLikeCalc*: Исправлено использование функций для параметров контроллеров DAQ из IO функций для первого использования.
 - *DAQ, BlockCalc, JavaLikeCalc, LogicLev, Siemens*: Исправлено получение ошибочных данных по схеме резервирования в нерезервированных задачах и при отсутствии резервных станций.
- *Protocol.HTTP, UI.WebCfg, UI.WebCfgD, UI.WebUser, UI.Vision*: Значение HTTP-параметра "Content-Type" исправлено для элемента "Charset".
- *UI.VCAEngine*: Исправлено падение движка визуализации при изменении мутированных атрибутов, которые многократно унаследованы.

Оптимизация и повышение производительности:

- *SYS.XML*: Функция преобразования дерева XML в строку оптимизирована по времени.
- *SYS.Archive*: Добавлен механизм адаптивной генерации запросов к архивам различного качества в случае общего запроса. Предотвращает продолжительную обработку в случае запроса больших интервалов времени и наличия менее качественных архивов.
- *DAQ, DAQ.DAQGate*: Выполнена оптимизация запросов между резервными станциями по размеру запросов и по времени. Запрашиваются только архивные атрибуты параметров DAQ и атрибуты, запланированные к запросу.
- *Archive.FSArch*: Добавлено создание информационных файлов упакованных архивов без хранения этих данных в БД. Эта функция позволяет обеспечить быстрое подключение больших архивов к другим станциям.
- *UI.Vision*: В примитиве "ElFigure" для отрисовки вместо QImage использован QPixmap, что значительно повысило производительность.
- *UI.Vision*: Для построения изображения примитива "Diagram" использован QImage вместо QPixmap, что значительно повысило производительность.
- *TFunction, DAQ.JavaLikeCalc*: Добавлен вложенный контекст функций с целью повышения производительности внешних вызовов.

Оптимизации использования памяти:

- *SYS*: Система переключена на использование традиционного "map" вместо "hash_map" во всех объектах с целью уменьшения потребления памяти.
- *UI.VCAEngine*: Объект атрибута "Attr" оторван от объекта "TCntrNode" с целью уменьшения потребления памяти до 25%.

3 Усовершенствование и стабилизация СВУ.

Заметные изменения были выполнены в рамках среды визуализации и управления (СВУ), а именно в модулях движка СВУ *UI.VCAEngine*, визуализаторов *UI.Vision* и *UI.WebVision*. Внесённые изменения были направлены на стабилизацию, оптимизацию потребления памяти и улучшение пользовательских свойств СВУ.

Улучшения СВУ:

- Добавлена полная поддержка вертикального масштабирования диаграмм.
- Добавлена поддержка альфа-канала в атрибуты цвета всех примитивов.
- *UI.VCAEngine*:
 - Включено и сделано безопасным наследование описания мутирующих атрибутов.
 - Добавлены новые функции пользовательского API для сеансов *wdgAdd()*, *wdgDel()*, *link()*, *linkSet()*. Добавление этих функций позволило реализовать концепцию полностью динамического формирования интерфейса пользователя.
 - Свойства прав доступа к виджету реализованы атрибутами "owner" и "perm" для повышения гибкости управления.
 - Добавлен механизм автоматического создания и запуска сеанса при загрузке.
- *UI.Vision*:
 - В инспектор атрибутов добавлена возможность групповой установки одноимённых атрибутов разных виджетов. Установка производится в групповом контейнере свойств инспектора атрибутов, который появляется при выделении одновременно нескольких виджетов.
 - В примитиве "EImage" для отрисовки вместо QImage использован QPixmap, что значительно повысило производительность.
 - Для построения изображения примитива "Diagram" использован QImage вместо QPixmap, что значительно повысило производительность.
 - Добавлена поддержка свойств пользовательского интерфейса. Добавлено управление и сохранение размера иконок панелей инструментов.
 - Добавлена возможность копирования значений атрибутов и связей посредством контекстного меню инспекторов атрибутов и связей.
 - Добавлена функция поиска по элементу "TextEdit".
- *UI.WebVision*:
 - Добавлена поддержка поворота примитива "Text".
 - Добавлена генерация основных событий мыши. Для обработки некоторых событий добавлен стек функций обработчиков.
 - Для пользователя-отправителя адаптирован доступ к его сеансу и управление.

4 Формирование и стабилизация API пользовательского программирования.

В значительной степени было произведено формирование нового объектного API пользовательского программирования, которое предусматривает интеграцию пользовательских функций в дерево объектов системы OpenSCADA. Кроме этого, был внесен ряд изменений в существующие библиотеки функций пользовательского API.

В частности были осуществлены следующие изменения:

- *SYS*:
 - *SYS, Special.FLibSYS*: Объект пользовательского API XMLNodeObj перемещён в ядро OpenSCADA (файлы tvariant.h, tvariant.cpp).
 - Пользовательское системное API OpenSCADA основанное на объекте TCntrNodeObj расширено для контроля прав доступа пользователя.
 - Объект TAreaObj переименован в TAreaObj.
 - Свойства объектов, вызовы функций и обработка исключений пользовательского API унифицировано.
 - *SYS.TVarObj*: Добавлена функция propList() для возможности сканирования атрибутов.
- *DAQ.JavaLikeCalc*:
 - Функции пользовательского API добавлены: insert(), replace(), toReal(), toInt(), parse(), parsePath() и path2sep() в объект значения строкового типа.
 - Добавлена функция isEval() к базовым типам, для обнаружения EVAL-значений.
- *Special.FLibComplex1*: Функция ПИД расширена новыми параметрами: Kd, Tzd и followSp. Удалён параметр Tf.
- *Объектное API пользовательского программирования*:
 - *SYS*: Функции пользовательского API: system(), message(), XMLNode(), cntrReq(), time(), localtime(), strftime(), strptime(), cron(), messDebug(), messInfo(), messNote(), messWarning(), messErr(), messCrit(), messAlert(), messEmerg() и strFromCharCode() добавлены в корневой объект ядра OpenSCADA.
 - *DB*: Функция пользовательского API SQLReq() добавлена в объект БД.
 - *Transport*: Функция пользовательского API messIO() добавлена для объекта исходящего транспорта.
 - *UI.VCAEngine*:
 - Функции пользовательского API: user(), almSndPlay() и almQuittance() добавлены в объект сеанса проекта.
 - В объект "Widget" добавлены функции пользовательского API: ownerSess(), ownerPage(), ownerWdg(), attrPresent(), attr(), attrSet(), wdgAdd(), wdgDel(), link() и linkSet().
 - Добавлен специальный атрибут "this" во все скрипты СВУ для доступа к объекту виджета и последующего вызова его функций пользовательского API.
 - *Archives*: Функция пользовательского API messGet() добавлена для объекта подсистемы "Архивы".
 - *DAQ.Value*: Добавлены функции пользовательского API get() и set() для атрибутов.

5 Общесистемные расширения.

В общесистемное API системы OpenSCADA были внесены значительные изменения и расширения с целью повышения функциональности и общей стабилизации:

- *SYS*:
 - Добавлен механизм предотвращения повторного запуска в скрипты: `demo-start` и `user-start`.
 - Добавлены глобальные функции для создания (регистрации) `taskCreate()` и закрытия `taskDestroy()` потоков. Все компоненты OpenSCADA используют новое API.
 - Добавлены файлы `debian/*` для поддержки построения Debian пакетов.
 - Добавлена функция `strParse()` для расширения функции `strSepParse()` на предмет поддержки многосимвольных разделителей и объединения односимвольных.
 - Добавлена инфраструктура счётчиков отладки.
 - Добавлена возможность назначения процессоров для потоков OpenSCADA на много-процессорных системах и много-ядерных процессорах.
 - Добавлен менеджер задач. Добавлено поле идентификатора процесса потока в менеджер задач.
 - Для реализации специфических функций добавлена обёртка над задачами OpenSCADA. Добавлена обработка исключений "TErroг" для задач внутри обёртки.
 - Код ядра системы OpenSCADA помещён в область имён "OSCADA". Версия библиотеки ядра OpenSCADA повышена до 1:0:0.
 - Унифицированы права доступа ко всем элементам интерфейса управления.
 - Добавлена проверка на версию более 2.0 утилиты сборки `libtool`. Включена сборка только разделяемых библиотек для модулей.
 - Идентификатор модуля помещён в класс конструктора для упрощения инициализации перевода текстовых сообщений модулей.
 - Для всех переводов использовано параметры утилиты `xgettext "-C --no-location --no-wrap -k_"`. Использовано для очистки от мусора файлов переводов.
 - Версии API всех модульных подсистем установлено в 5.
- *SYS.Function, DAQ.BlockCalc, DAQ.LogicLev, UI.VCAEngine*: Добавлен механизм контроля за изменениями параметров контроллеров подсистемы "Сбор данных".
- *DB*:
 - Добавлено автоматическое открытие доступных таблиц по запросу к ним.
 - Добавлено управление транзакциями в функцию `sqlReq()`, а также добавлена функция `transCloseCheck()`.
- *TFunction, DAQ.JavaLikeCalc*: Добавлен вложенный контекст функций с целью повышения производительности внешних вызовов.
- *DAQ*: Код ошибки добавлен в статус контроллера, что позволяет идентифицировать состояние и отображать его.
- *Protocol, Transport, DAQ.ModBus, Protocol.UserProtocol*: Добавлена функция `itemListIn()` для выбора элементов протокола. Добавлен выбор элементов протокола посредством функции `TProtocol::itemListIn()`.

6 Значительные улучшения и исправления отдельных модулей.

В процессе работы над данным промышленным релизом значительных изменений, улучшений и стабилизации претерпели отдельные модули OpenSCADA:

- *DB.MySQL*:
 - Добавлена поддержка неограниченных значений типов данных "Вещественное" и "Целое".
 - Добавлено преобразование значения EVAL для вещественного типа.
- *DB.SQLite*: Улучшен механизм транзакций для поддержки внутривпользовательских SQL-запросов и закрытия старых транзакций.
- *DB.FireBird*: Добавлена поддержка многоязыковости текстовых переменных. Унифицирована поддержка транзакций.
- *DAQ*:
 - *DAQGate*: Добавлена поддержка возможности запроса отдельных атрибутов. Использовано для запроса архивируемых и часто запрашиваемых атрибутов. Все остальные атрибуты запрашиваются с периодом синхронизации путём полного запроса. Используется для уменьшения нагрузки и времени при обмене DAQGate с удалённой OpenSCADA станцией.
 - *ICP_DAS*: Добавлен статус DIP-переключателя для ПЛК LP-8781. Функции отправки сообщений по последовательному интерфейсу перемещены в общую функцию setReq() и все запросы используют её.
 - *ModBus*: Добавлены суффиксы "i2,i4,f,b0" к регистрам (R и RI) для гибкого доступа посредством различных типов данных.
 - *JavaLikeCalc*:
 - Добавлена поддержка обзорного цикла для сканирования свойств объекта "for(<i> in <obj>) <code>;".
 - Свойства объектов, вызовы функций и обработка исключений пользовательского API унифицировано.
- *Transport*:
 - *Serial*:
 - Добавлена возможность выбора типа управления потоком. Поддерживается аппаратное (CRTSCTS) и программное (IXON IXOFF) управление потоком.
 - Добавлена поддержка режима модема для всех входных и выходных транспортов.
 - *SSL*:
 - Добавлена поддержка KeepAlive лимитов. Добавлена поддержка переподключения после потери соединения и для предотвращения потери сообщений.
 - Добавлено получение адреса отправителя.
 - *SSL, Sockets*: В исходящий транспорт добавлены тайминги для отдельного управления временем соединения и временем ожидания последующих ответов.
 - *Protocol.HTTP*:
 - Добавлена функция исходящего протокола.
 - Добавлена поддержка автоматического входа.
- *Archive.FSArch*: Добавлено создание информационных файлов упакованных архивов без хранения этих данных в БД. Эта функция позволяет обеспечить быстрое подключение больших архивов к другим станциям.
- *UI.QTcfg*:
 - Добавлена подстройка строк для таблиц, после добавления новых строк. Добавлена поддержка копирования содержимого таблицы в буфер обмена.
 - Добавлена функция поиска по элементу "TextEdit".
 - Добавлена возможность копирования списка выбранных узлов.

7 Формирование документации и переводы документации и интерфейса.

Поскольку данный релиз позиционируется как промышленный релиз с продолжительным периодом поддержки, то к его выходу было закончено написание на Русском языке, а также перевод на Английский язык всей основной документации. Кроме того, завершён перевод интерфейса программы на Английский, Русский, Украинский и Немецкий языки:

- Добавлен перевод README на Украинский.
- Добавлен перевод документа DAQ и обобщающего документа на Английский.
- Добавлен документ "Быстрый старт".
- Выполнен перевод документа "Быстрый старт" на Английский язык.
- Общий документ "OpenSCADA" на Английском обновлен для включения частей "Быстрый старт" и DAQ.OPC-UA.
- Обновлен перевод интерфейса ядра системы на Русский, Украинский и Немецкий языки.
- Для всех переводов использовано параметры утилиты xgettext "-C --no-location --no-wrap -k_". Использовано для очистки от мусора файлов переводов.

8 Унификация демонстрационной БД.

В процессе подготовки промышленного релиза значительное внимание было уделено улучшению демонстрационной БД как яркой демонстрации функций системы OpenSCADA:

- Значительное обновление для поддержки новых возможностей и включения новых улучшений из системы автоматизации шаровых мельниц.
- Добавлена поддержка пользовательского вызова демонстрационной БД в полноценном режиме на запись.
- Добавлено масштабирование существующих элементов в случае отсутствия строк и колонок для обзорного кадра и сводной таблицы.
- Обновлено для включения новых моделей аппаратов технологических процессов и адаптации всех остальных моделей.
- Обновлено для:
 - Добавления объектов сигнализации компрессоров: KM201, KM301, KM202 и KM302.
 - Поддержки ручного ввода переменных с панели управления.
 - Обновления перевода.
 - Исправления нескольких ошибок.
- Скрипт главной страницы проекта переписан для использования новых функций пользовательского API.
- Использован новый механизм доступа к элементам пользовательского интерфейса. Предоставлен реальный доступ на чтение для простого пользователя ("user").

9 Решения, построенные на основе OpenSCADA

На основе проекта OpenSCADA построен фактически весь спектр решений автоматизации, которые ярко демонстрируют возможности, а также являются сигналом готовности системы для широкого промышленного применения.

В целом, на основе OpenSCADA авторами проекта решались задачи: полного динамического моделирования технологических процессов (ТП), управление ТП на уровне программируемого логического контроллера (ПЛК) и формирование человеко-машинного интерфейса ТП.

Моделирование ТП представлено решениями: «Библиотека моделей аппаратов технологических процессов», «Динамическая модель реального времени Анастасиевской ГЛКС» и «Динамическая модель парового котла №9 ДМК». Причём решение «Динамическая модель реального времени Анастасиевской ГЛКС» было положено в основу демонстрационной БД проекта и распространяется со всеми дистрибутивами OpenSCADA.

Сфера управления ТП на уровне программируемого логического контроллера (ПЛК) представлена решениями: «OpenSCADA в программируемом логическом контроллере (ПЛК)» и «Среда исполнения ПЛК LP-8x81 фирмы ICP DAS на основе OpenSCADA».

Сфера формирования человеко-машинного интерфейса ТП представлена решениями: «Библиотеки графических элементов пользовательских интерфейсов» и «АСУ ТП шаровых мельниц ШБМ 287\410 котлоагрегата БКЗ 160–100 ПТ».

Нужно также отметить наличие ряда решений, не вошедших в вышеприведенный перечень, и выполненных разработчиками системы для третьих лиц или непосредственно её пользователями.

Заключение

Система OpenSCADA готова для практического использования в сложных промышленных условиях и выполнения широкого спектра задач автоматического управления технологическими процессами.

С целью предоставления качественного сервиса на основе решений OpenSCADA, а также развития в направлении поддержки специализированного оборудования, протоколов и интерфейсов; разработчиками планируется и активно ведётся работа по формированию политики предоставления услуг на коммерческой основе. В число коммерческих услуг будет входить: техническая поддержка, консультации и целевая разработка расширений, интеграция и внедрение решений, а также разработка специализированных решений и продуктов на основе OpenSCADA.

Версия 0.7.0 предоставляет решения OpenSCADA только для платформы Linux x86 и x86_64. Для осуществления поддержки других аппаратных и программных платформ, запланированных проектом, работа будет продолжена. Так, к версии 0.8.0 планируется расширить поддержку аппаратных платформ минимум на платформу APM, как представителя RISC архитектур.

В появлении первой промышленной версии системы OpenSCADA активное участие приняли:

- [Савоченко Роман](#): Основной объём работ по проектированию, разработке, документированию и тестированию.
- [Лысенко Максим](#): Разработка примитива "Элементарная фигура" среды визуализации и управления (СВУ), создание модуля поддержки БД PostgreSQL и перевод большей части документации на Английский язык.
- [Яшина Ксения](#): Разработка примитива "Элементарная фигура" СВУ.
- [Попкова Ирина](#): Перевод интерфейса системы OpenSCADA и её модулей на Немецкий язык.
- [Алмаз Каримов](#): Разработка модуля поддержки протокола DCON и активное тестирование.
- [Попков Алексей](#): Активное тестирование и участие.
- Многие другие пользователи системы OpenSCADA, посредством всестороннего тестирования.

Ссылки

Протокол тестирования промышленного релиза: <http://wiki.oscada.org/Works/Tests/release070>