

OpenSCADA 0.6.0

Оглавление

OpenSCADA 0.6.0	1
Вступ	1
1 Концепція середовища візуалізації та керування (СВК), а також візуалізатор СВК на базі бібліотеки QT4	3
2 Значне підвищення стабільності, стійкості та продуктивності системи	4
3 Уніфікація інтерфейсу керування та побудова протоколу взаємодії на його основі	5
4 Уніфікація механізму побудови параметрів підсистеми «Збір даних» за шаблоном	5
5 Розширення основної мови (JavaLikeCalc (RU)) та API рівня користувальницького програмування	6
6 Перехід на бібліотеку QT версії 4 у модулях, які використовують цю бібліотеку ..	6
7 Загальне покращення API ядра системи OpenSCADA	6
8 Загальне вдосконалення збіркової системи, як наслідок, спрощення та розширення підтримуємих платформ	7
9 Новий сайт проекту OpenSCADA	8
10 Плани подальшого розвитку	8

Вступ

Реліз відкритої SCADA(Supervisory control and data acquisition) системи версії 0.6.0 позиціонується як концепт-реліз. Це пов'язано з тим, що на цей час фактично реалізовано основні функції концепції SCADA системи, які було сформульовано проектом OpenSCADA у 2003 році.

Головною метою даного релізу є надання спільноті користувачів та розробників вільного програмного забезпечення (ПЗ) концепції та платформи для відпрацювання рішень побудови комплексних SCADA-систем та інших суміжних рішень з цілеспрямованим удосконаленням та стабілізацією наявних компонентів, а також розробки потрібних розширень та модулів.

Цей документ є обробкою(компіляцією) документа <ChangeLog> системи OpenSCADA версії 0.6.0, який призначено коротко та наочно освітлити нові можливості системи OpenSCADA. Детально ознайомитися зі змінами у системі OpenSCADA можна у файлі ChangeLog з дистрибутиву системи або тут: <http://diyaorg.dp.ua/oscadawiki/Works/ChangeLog>.

Ключовими особливостями даної версії є:

- Концепція середовища візуалізації та керування (СВК), а також візуалізатор СВК на основі бібліотеки QT4.
- Значне підвищення стабільності, стійкості та продуктивності системи.
- Уніфікація інтерфейсу керування та побудова транспортного протоколу взаємодії на його основі.
- Уніфікація механізму побудови параметрів підсистеми «Збір даних» за шаблоном.
- Розширення основної мови ([JavaLikeCalc \(RU\)](#)) та API рівня користувальницького програмування.
- Перехід на бібліотеку QT версії 4 у модулях, які використовують цю бібліотеку.
- Загальне покращення API ядра системи OpenSCADA.
- Загальне покращення збіркової системи, як наслідок покращення та розширення підтримуваних платформ.

- Новий сайт проекту OpenSCADA.

Нові та поновлені модулі:

- *Transport.Sockets (1.3.0)* — Загальна стабілізація модуля.
- *Archive.FSArch (0.9.5)* — Оптимізовано архіватор значень. Додано архівацію повідомлень у форматі плоского тексту. Модуль перейменовано з «BaseArh» у «FSArch». виправлено багато помилок.
- *Archive.DBArch (0.5.0)* — Новий модуль архівації повідомлень та значень до БД.
- *DB.DBF (1.8.1)* — Загальна стабілізація модуля.
- *DB.MySQL (1.3.0)* — Додано підтримку API запиту переліку наявних таблиць та структури цих таблиць. Значна стабілізація та оптимізація модуля.
- *DB.SQLite (1.3.0)* — Додано підтримку API запиту переліку наявних таблиць та структури цих таблиць. Значна стабілізація та оптимізація модуля.
- *DB.FireBird (0.6.0)* — Новий модуль підтримки СУБД «[FireBird \(RU\)](#)».
- *Protocol.HTTP (1.3.0)* — Значна переробка та стабілізація модуля.
- *Protocol.SelfSystem (0.7.0)* — Новий модуль власного протоколу OpenSCADA, базованого на інтерфейсі керування OpenSCADA.
- *Special.FLibComplex1 (1.0.0)* — Загальна стабілізація модуля.
- *Special.FLibSYS (0.5.0)* — Результат поєднання з модулем Special.FLibTime та робіт по уніфікації користувальницького API роботи з архівами.
- *Special.SystemTests (1.3.5)* — Поновлено деякі тести.
- *DAQ.DiamondBoards (1.0.0)* — Переглянуто механізм збору за посередництвом переривань, виправлено багато помилок, разом з бібліотекою API версії 5.91 виправлено проблеми блокування потоків та випущено фінальний реліз.
- *DAQ.System (1.5.0)* — Загальна стабілізація модуля. Перейменовано з «OperationSystem».
- *DAQ.BlockCalc (1.0.0)* — Деякі покращення у механізмі зв'язування, розширена функція копіювання контролерів, блоків та параметрів, виправлення помилок та фінальний реліз.
- *DAQ.JavaLikeCalc (1.0.0)* — Розширено граматику мови, додано підтримку механізму прекомпіляції процедур користувача та наступного їх використання у різних частинах OpenSCADA, виправлення помилок та фінальний реліз.
- *DAQ.LogicLev (0.9.0)* — Винесений до модуля код підсистеми «Параметри» OpenSCADA 0.5.0 з розширенням функціоналу та виправленням помилок.
- *DAQ.CIF (0.9.0)* — Новий модуль джерела даних контролерів фірми Siemens серії S7 (300,400) за допомогою мережі ProfiBUS, протоколу MPI та комунікаційного процесору фірми Hilcher CIF50PB.
- *DAQ.SNMP (0.3.0)* — Новий модуль базової підтримки протоколу Simple Network Management Protocol у режимі читання.
- *DAQ.ModBus (0.4.0)* — Новий модуль підтримки джерел даних, які доступно за протоколом ModBus.
- *DAQ.Transporter (0.3.0)* — Новий модуль підтримки механізму відображення даних підсистеми «Збір даних» віддалених OpenSCADA станцій на локальні.
- *UI.QTStarter (1.5.0)* — Перехід на бібліотеку QT версії 4, загальна стабілізація та виключення зовнішніх викликів функцій QT.
- *UI.QTCfg (1.5.0)* — Перехід на бібліотеку QT версії 4, підтримка зображень у інтерфейсі керування, загальна стабілізація та оптимізація.
- *UI.WebCfg (1.5.0)* — Загальна стабілізація модуля.
- *UI.VCAEngine (0.5.0)* — Новий модуль рушія середовища візуалізації та керування (СВК).
- *UI.Vision (0.5.0)* — Новий модуль візуалізатора середовища візуалізації та керування (СВК) на бібліотеці QT.
- *UI.WebVision (0.2.0)* — Новий модуль візуалізатора середовища візуалізації та керування (СВК) на основі WEB-технологій (XHTML, JavaScript, CSS, AJAX).

1 Концепція середовища візуалізації та керування (СВК), а також візуалізатор СВК на базі бібліотеки QT4.

Основним нововведенням релізу 0.6.0 системи OpenSCADA стала базова реалізація гнучкого середовища візуалізації та керування (СВК). СВК OpenSCADA побудована за концепцією «модель даних» – «вигляд», де «модель даних» описує структуру та обчислення візуального інтерфейсу (VI), а «вигляд» реалізує безпосереднє представлення та контакт з користувачем. Такий підхід дозволяє реалізовувати візуалізацію з використанням різних графічних бібліотек та оточень інтерфейсу користувача. Реалізована підтримка режиму розробки та виконання інтерфейсу користувача на основі відомої графічної бібліотеки QT4. На основі WEB-технологій створена базова реалізація візуалізатора з підтримкою режиму виконання (RunTime).

Концепцією передбачена можливість формування широкого діапазону інтерфейсів користувача різної складності. В основу всіх цих інтерфейсів покладено обмежений набір базових примітивів. Комбінуючи ці примітиви можна створювати як похідні кадри елементів візуалізації користувача, так і сторінки кінцевих інтерфейсів візуалізації. Визначено наступний набір базових примітивів:

- *ElFigure* — Елементарні графічні фігури.
- *FormEl* — Елементи форми.
- *Text* — Текст.
- *Media* — Медіа.
- *Diagram* — Діаграма.
- *Protocol* — Протокол.
- *Document* — Документ.
- *Function* — Функція API об'єктної моделі OpenSCADA.
- *Box* — Контейнер.
- *Link* — Лінії зв'язку.

З числа перелічених примітивів у даному релізі залишилися не реалізованими: “Document”, “Function” та “Link”. Їх реалізацію заплановано після виходу версії 0.6.0.

Реалізуємий проект СВК передбачає підтримку наступних функцій:

- три рівні складності у формування інтерфейсу візуалізації, дозволяє органічно освоювати та застосовувати інструментарій за методикою від простого до складного:
 - 1) формування з шаблонних кадрів, шляхом призначення динаміки (можливо і без графічної конфігурації);
 - 2) графічне формування нових кадрів, шляхом використання готових елементів візуалізації з бібліотеки (мнемосхеми);
 - 3) формування нових кадрів, шаблонних кадрів та елементів відображення в бібліотеці.
- побудова інтерфейсів візуалізації різної складності починаючи від простих плоских інтерфейсів моніторингу і закінчуючи повноцінними ієрархічними інтерфейсами, які використовуються в SCADA системах;
- надання різних способів формування та конфігурації користувальницького інтерфейсу, які основою на різних інтерфейсах графічного представлення (QT, Web, Java ...) або-ж за посередництвом стандартного інтерфейсу керування системою OpenSCADA;
- зміну динаміки у процесі виконання;
- побудова нових шаблонних кадрів на рівні користувача та формування спеціалізованих, під галузь застосування, бібліотек кадрів (наприклад включення кадрів параметрів, графіків та інших елементів з ув'язкою їх один з одним), у відповідності з концепцією вторинного використання та накопичення;
- побудова нових користувальницьких елементів візуалізації та формування спеціалізованих під галузь застосування бібліотек кадрів, у відповідності з теорією вторинного використання та накопичення;
- формування логіки нових шаблонних кадрів та користувальницьких елементів візуалізації як простими зв'язками, так і лаконічною, повноцінною мовою користувальницького

програмування;

- можливість включення у користувальницькі елементи візуалізації функцій (або кадрів обчислення функцій) об'єктної моделі OpenSCADA, практично пов'язуючи представлення з алгоритмом обчислення (наприклад, візуалізуючи бібліотеку моделей апаратів ТП для наступної візуальної побудови моделей ТП);
- розподіл даних користувальницьких інтерфейсів та інтерфейсів представлення цих даних, який дозволяє будувати інтерфейси користувача у одному середовищі, а виконувати у багатьох інших (QT, WEB, Java ...);
- можливість підключення до інтерфейсу який виконується для спостереження та корекції дії (наприклад, при навчанні операторів та контролю у реальному часі за його діями);
- візуальна побудова різноманітних схем з накладенням логічних зв'язків та наступним централізованим виконанням у фоні (візуальна побудова та виконання математичних моделей, логічних схем, релейних схем та інших процедур);
- надання функцій об'єктного API у систему OpenSCADA, може використовуватися для керування властивостями інтерфейсу візуалізації з процедур користувача;
- побудова серверів кадрів, елементів візуалізації та проектів інтерфейсів візуалізації з можливістю обслуговування множини клієнтських з'єднань;
- проста організація клієнтських станцій на різних засадах (QT, WEB, Java ...) з підключенням до центрального серверу;
- повноцінний механізм розподілу повноважень поміж користувачами, який дозволяє створювати та виконувати проекти з різними правами доступу до його компонентів;
- гнучке формування правил сигналізації та повідомлення з урахуванням та підтримкою різних способів повідомлення;
- підтримка користувальницького формування палітри та шрифтових переваг для інтерфейсу візуалізації (теми);
- підтримка користувальницького формування карт повідомлень під різноманітне обладнання керування та переваг користувача;
- підтримка профелів користувачів, яка дозволить визначати різні властивості інтерфейсу візуалізації (гама кольорів, шрифтові особливості, карти повідомлень користувача);
- гнучке зберігання та розповсюдження бібліотек віджетів, кадрів та проектів інтерфейсів візуалізації у БД, які підтримуються системою OpenSCADA; користувачу практично потрібно тільки зареєструвати отриману БД з даними.

Більш детально ознайомитися з проектом концепції СВК можна за посиланням: <http://diyaorg.dp.ua/oscadawiki/Doc/KoncepcijaSredyVizualizacii> (RU).

2 Значне підвищення стабільності, стійкості та продуктивності системи.

У процесі робіт над версією 0.6.0 була виконана низка заходів по загальній стабілізації системи OpenSCADA, а також по підвищенню продуктивності системи. Перелічимо найважливіші з них:

- виправлена глобальна проблема недостатнього покриття ключових параметрів ресурсами. Проблема чітко проявила себе на багатоядерних архітектурах.
- виключена можливість прямого виклику QT-методів з не QT-потоків, у всіх QT-модулях. Приводило до падіння OpenSCADA під час зупинки.
- механізм захоплення ресурсів переглянуто, виправлено для робіт у багатоядерних системах та оптимізовано. У результаті підвищилась загальна стабільність та продуктивність.
- виправлені функції TSYS::addr2str() та TSYS::str2addr(). Приводило до незрозумілих падінь модуля QTCfg.
- втілення асоціативних контейнерів (map та hash_map) у найбільш відповідальні вузли OpenSCADA. Значно підвищило швидкість доступу до елементів великих масивів, наприклад:

- при доступі до динамічних об'єктів, динамічного дерева, під час розбору повного шляху динамічного об'єкту;
- при роботі з властивостями елементів конфігурації та таблиць БД.
- Систематизовано та уніфіковано інтерфейс керування системою та його мова. Підвищило функціональні якості та реактивність інтерфейсів та протоколів, побудованих на ньому.
- Додано можливість адресації окремих елементів запису таблиць БД. Підвищило продуктивність доступу до БД, шляхом запиту тільки потрібних елементів запису.
- Додано механізм запам'ятовування зміщень у стрічці для основних функцій при розборі шляху або адресу з розділювачем. Виключило геометричний ріст витрат часу на розбір стрічки від кількості елементів у ній.
- Оптимізація реалізації запитів у модулях БД: [SQLite \(RU\)](#) та [MySQL \(RU\)](#). Значно підвищило ефективність доступу до БД цих типів.
- Оптимізовано механізм перевірки прав доступу. Дозволило знизити час перевірки прав доступу.

3 Уніфікація інтерфейсу керування та побудова протоколу взаємодії на його основі.

В цій версії виконано уніфікацію інтерфейсу керування OpenSCADA у напрямку формування мови запитів, спрощення формування інформаційної частини та обробки командних запитів, а також з метою спрощення формування транспортних протоколів на основі даного інтерфейсу. Результатом даної уніфікації стало підвищення загальної продуктивності запитів за цим інтерфейсом, а також створення транспортного протоколу [SelfSystem \(RU\)](#). За основу цього протоколу взято запити до інтерфейсу керування OpenSCADA, обернуті у тонкий шар керуючого протоколу.

За посередництвом даного протоколу реалізовано взаємодію між різними станціями OpenSCADA в мережі. Наприклад, модуль конфігурації на бібліотеці QT, [QTCfg \(RU\)](#), отримав можливість конфігурації віддалених станцій, дозволяючи будувати розподілені системи з єдиним центром конфігурації. Візуалізатор інтерфейсів користувача на основі бібліотеки QT, [Vision \(RU\)](#), також, отримав можливість формування інтерфейсів та виконання сеансів інтерфейсів користувача, які розташовано на віддалених OpenSCADA станціях, формуючи тим самим конфігурацію з сервером візуалізації. До всього цього було реалізовано перший модуль відображення даних віддалених станцій OpenSCADA на локальні; їм став модуль відображення підсистеми «Збір даних». В подальших планах є створення таких, транспортних, модулів для інших модульних підсистем.

Слідом за даними рішеннями було додано механізм сервісних функцій інтерфейсу керування. Цей механізм дозволяє отримати доступ до даних зі складним форматуванням та великих об'ємів, у пріоритетному режимі запити та оптимальній для передачі формі.

4 Уніфікація механізму побудови параметрів підсистеми «Збір даних» за шаблоном.

У версію 0.5.0 було впроваджено механізм формування параметрів підсистеми «Збір даних» з потрібною користувачу структурою, тобто за шаблоном. Цей механізм було реалізовано у вигляді окремої підсистеми «Параметри». У цьому релізі ідея формування параметрів підсистеми «Збір даних» за шаблоном була поширена на всі модулі підсистеми «Збір даних». При цьому раніше існуюча підсистема «Параметри» була перетворена у рівноцінний модуль логічного рівня параметрів «[LogicLev \(RU\)](#)», а шаблони стало можливим формувати на рівні підсистеми «Збір даних», групуючи їх за бібліотеками шаблонів. Фактично будь який модуль, який надає доступ до даних високоінтелектуальних джерел, може включати підтримку параметрів, які формуються за шаблоном, вносячи власний сенс у посилання внутрішнього рівня. Прикладом такого модуля став

модуль доступу до даних контролерів фірми Siemens серії S7 за протоколом MPI мережі ProfiBUS [CIF \(RU\)](#).

Шаблони параметрів стали формуватися незалежно від існуючих функцій, шляхом формування власної функції обробки параметрів та наступної її компіляції у модулі, який надає механізм компіляції мови користувальницького програмування. Шаблон може бути сформовано взагалі без процедури користувача, якщо потрібне тільки пасивне, вибіркове відображення параметрів.

5 Розширення основної мови ([JavaLikeCalc \(RU\)](#)) та API рівня користувальницького програмування.

Компілюючий інтерпретатор схожої на Java мови, разом з бібліотеками функцій користувальницького програмування фіксованого типу (API), формують середовище користувальницького програмування OpenSCADA.

У даному релізі подібна до Java мова отримала:

- повноцінну підтримку циклів while та for;
- механізм переривання зацикленних процедур;
- операції префіксного та постфіксного інкременту та декременту;
- механізм прекомпіляції користувальницьких програм різних підсистем;
- прямий механізм адресації параметрів підсистеми «Збір даних».

API функцій користувальницького програмування було уніфіковано і визначається бібліотеками функцій:

- *FLibSYS* — Бібліотека системного API, на час цього релізу містить функції для роботи з часом та датою, а також з архівами повідомлень та значень.
- *FLibMath* — Бібліотека математичних функцій.
- *FLibComplex1* — Бібліотека функцій сумісності зі SCADA «Complex1».

6 Перехід на бібліотеку QT версії 4 у модулях, які використовують цю бібліотеку.

У зв'язку з виходом нової версії бібліотеки QT (версія 4) та повсюдного її розповсюдження, а також початку робіт над середовищем візуалізації та керування (СВК), було прийнято рішення про переведення існуючих модулів, основаних на QT, на нову версію.

Цю роботу було виконано цілком та модулі підсистеми "Інтерфейси користувача" [QTCfg \(RU\)](#), [QTStarter \(RU\)](#) та [Vision \(RU\)](#) функціонують на основі бібліотеки QT версії 4. У зв'язку з наявністю деяких проблем у ранніх версіях та відсутності декількох функцій рекомендується використовувати версію бібліотеки не менш 4.3.

У результаті переходу на бібліотеку QT4 вдалося добитися більшої стабільності та розширити функціональні характеристики модулів які її використовують.

7 Загальне покращення API ядра системи OpenSCADA.

В процесі робіт над даним релізом були внесені значені зміни до API системи OpenSCADA. Значною мірою це пов'язано з процесами стабілізації, оптимізації, виправленням помилок та загальним покращенням API і у меншій мірі з доданням нових функцій. Перелічимо найбільш значні зміни:

- Уніфікація API інтерфейсу керування OpenSCADA. Детально про це у розділі 3.
- У підсистемі безпеки, у користувачів та груп користувачів, усунена індексація користувачів та груп користувачів. Також, усунена група користувача по замовченню.

Виконано з метою усунення подвійної ідентифікації, оскільки вже присутня швидка ідентифікація за символьним ідентифікатором/ім'ям.

- Скасовано підсистему «Параметри», а її функції винесено до модуля логічного рівня «[LogicLev \(RU\)](#)» підсистеми «Збір даних». Шаблони параметрів згруповано за бібліотеками та розміщено у підсистемі «Збір даних». Детальніше про це у розділі 4.
- Додано механізм компіляції процедур у модулях підсистеми «Збір даних» для використання користувальницького програмування у різних вузлах OpenSCADA.
- Додано шаблон функції локалізації повідомлень `_()` з метою уніфікації, підвищення читабельності вихідного тексту та зменшення насиченості.
- Додані шаблони функцій для формування повідомлень в OpenSCADA та підвищення читабельності вихідного тексту, а також зменшення насиченості. Додані шаблони функцій: `message()`, `mess_err()`, `mess_info()` та подібні.
- Уніфіковані точки станів контролерів підсистеми «Збір даних». Тепер стан «Включено» відповідає встановленню всіх зв'язків, а «Запущено» тільки фактичному збору даних.
- Практично всі функції встановлення параметрів перейменовано к вигляду `setMethod()`.
- Додана підтримка атрибуту видимості полів запису БД. Це дозволяє підвищити продуктивність роботи з БД за рахунок отримання тільки потрібних даних.
- Додана підтримка глобального коду помилки у об'єкті `TError`. Використовується для вибіркової обробки помилок.
- Механізм захоплення ресурсів переглянуто, виправлено для роботи з багатоядерними системами та оптимізовано.
- Додано механізм попередньої ініціалізації створюваних архівів значень у модулях збору даних. Вдалося виключити додаткове настроювання архіву при конфігурації атрибутів параметрів.
- Додана підтримка механізмів запити переліку доступних у БД таблиць та їх структури, а також редагування їх вмісту за посередництвом інтерфейсу керування та конфігурації OpenSCADA.
- Реалізована підтримка вихідних протоколів. Внутрішній інтерфейс протоколу реалізовано потоковим зі спеціалізованою для протоколу структурою.

Детальніше ознайомитися з API даної версії можна у відповідному документі «[API системы OpenSCADA \(RU\)](#)».

8 Загальне вдосконалення збіркової системи, як наслідок, спрощення та розширення підтримуємих платформ.

З метою реалізації функції повноцінного, раннього, повідомлення про відсутність потрібних для збірки залежностей сценарій перевірки оточення було доповнено кодом перевірки основних залежностей збірки. В результаті цього повідомлення про всі незадоволені залежності будуть видані на стадії конфігурації збіркової системи, що прискорює вирішення залежностей або прийняття рішення про відключення збірки того або іншого модуля.

Формат документації, яку розміщено в дереві вихідних текстів та тарболі проекту, було змінено на pdf, у зв'язку з більш спрощеними маніпуляціями на стадії встановлення.

9 Новий сайт проекту OpenSCADA.

В процесі роботи над даним релізом, для проекту OpenSCADA було створено новий сайт ( <http://oscada.diy.org.dp.ua>), базований на CMS-рушії  [Туро3](#). Сайт є багатомовним та містить основну інформацію про проект OpenSCADA, зберігаючи великий потенціал по розширенню. Для цього сайту було виконано переклади основних інформаційних матеріалів на Українську та Англійську мови.

Заплановано додання наступних функцій на головний сайт: форум, галерею скриншотів та інших медіа-матеріалів, поштових розсилок, трекеру змін та робіт які ведуться за проектом.

10 Плани подальшого розвитку

Плануються наступні, часті релізи у межах версії 0.6.0, з метою оперативного відслідковування процесу прикладної адаптації та стабілізації. В фіналі планується випуск стабільної-промислової версії 0.7.0 для платформи Linux x86, x86_64.

Для реалізації до версії 0.7.0, по проміжним версіям 0.6.0, можна визначити наступні важливі завдання:

- Web-візуалізатор СВК.
- Модульний механізм повідомлення та сигналізації.
- Розробка та реалізація примітивів СВК: Документ, Функція та зв'язок.
- Реалізація механізму: «Тема оформлення» та «Карта повідомлень» для СВК.
- Розширення переліку підтримуваних джерел даних.