

OpenSCADA 0.6.2

Оглавление

OpenSCADA 0.6.2	1
Вступ	1
1. Підвищення стабільності, стійкості та продуктивності системи	3
2. Удосконалення та стабілізація СВКУ	3
3. Загальносистемні розширення	4
4. Вдосконалення частини модулів підсистеми «Збір даних»	5
5. Плани подальшого розвитку	5

Вступ

Реліз відкритої SCADA(Supervisory control and data acquisition) системи версії 0.6.2 є наступним стабілізуючим релізом гілки 0.6.0. У межах цієї версії пройдено великий шлях по стабілізації та адаптації до практичних завдань.

Цей документ є обробкою(компіляцією) документа <ChangeLog> системи OpenSCADA версії 0.6.2, який призначено коротко та наочно освітити 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> (RU).

Ключовими особливостями даної версії є:

- Підвищення стабільності, стійкості та продуктивності системи.
- Вдосконалення та стабілізація СВК.
- Загальносистемні розширення.
- Вдосконалення частини модулів підсистеми «Збір даних».

Нові та поновлені модулі:

- *Archive.FSArch (1.0.0)* — Виправлення деградації продуктивності при великій кількості файлів архіву. Інтеграція механізму глобального контролю за модифікацією. Чистка вихідного коду.
- *Archive.DBArch (0.7.0)* — Інтеграція механізму глобального контролю за модифікацією. Чистка вихідного коду.
- *DB.MySQL (1.4.0)* — Адаптація до останньої версії MySQL. Додання спливаючої підказки по формату адреси БД. Інтеграція механізму глобального контролю за модифікацією. Чистка вихідного коду.
- *DB.SQLite (1.4.0)* — Додання спливаючої підказки по формату адреси БД. Інтеграція механізму глобального контролю за модифікацією. Чистка вихідного коду.
- *DB.FireBird (0.8.0)* — Додання спливаючої підказки по формату адреси БД. Інтеграція механізму глобального контролю за модифікацією. Чистка вихідного коду.
- *DB.DBF (1.9.0)* — Додання спливаючої підказки по формату адреси БД. Інтеграція механізму глобального контролю за модифікацією. Чистка вихідного коду.
- *DAQ.BlockCalc (1.1.0)* — Уніфікація функції копіювання елементів. Заміна механізму синхронізації БД на стандартний. Додання спливаючої підказки по формату атрибутів параметрів DAQ. Стабілізація модуля. Інтеграція механізму глобального контролю за модифікацією. Чистка вихідного коду.
- *DAQ.DiamondBoards (1.1.0)* — Стабілізація модуля. Інтеграція механізму глобального контролю за модифікацією. Чистка вихідного коду.
- *DAQ.JavaLikeCalc (1.2.0)* — Переписано механізм роботи з цілими для приведення основних чисельних операцій у реальну форму. Інтеграція з глобальним механізмом

копіювання елементів. Стабілізація модуля. Інтеграція механізму глобального контролю за модифікацією. Чистка вихідного коду.

- *DAQ.SNMP (0.3.2)* — Стабілізація модуля. Інтеграція механізму глобального контролю за модифікацією. Чистка вихідного коду.
- *DAQ.LogicLev (0.9.1)* — Стабілізація модуля. Інтеграція механізму глобального контролю за модифікацією. Чистка вихідного коду.
- *DAQ.ModBus (0.9.0)* — Значне розширення функціональності. Стабілізація модуля. Інтеграція механізму глобального контролю за модифікацією. Чистка вихідного коду.
- *DAQ.Siemens (1.1.0)* — Додання підтримки протоколу ISO_TSAP (ProfiNet) за посередництвом бібліотеки Libnodave. Стабілізація модуля. Інтеграція механізму глобального контролю за модифікацією. Чистка вихідного коду.
- *DAQ.System (1.6.1)* — Стабілізація модуля. Інтеграція механізму глобального контролю за модифікацією. Чистка вихідного коду.
- *DAQ.Transporter (0.3.2)* — Стабілізація модуля. Інтеграція механізму глобального контролю за модифікацією. Чистка вихідного коду.
- *Transport.Sockets (1.3.5)* — Додання спливаючої підказки по формату адреси вхідних та вихідних сокетів. Стабілізація модуля. Інтеграція механізму глобального контролю за модифікацією. Чистка вихідного коду.
- *Protocol.HTTP (1.3.2)* — Чистка вихідного коду.
- *Protocol.SelfSystem (0.8.0)* — Додання компресії трафіку, а також можливість формування прямих запитів без підключення. Стабілізація модуля. Чистка вихідного коду.
- *Special.FLibComplex1 (1.0.2)* — Інтеграція механізму глобального контролю за модифікацією. Чистка вихідного коду.
- *Special.FLibMath (0.5.1)* — Інтеграція механізму глобального контролю за модифікацією. Чистка вихідного коду.
- *Special.FLibSYS (0.6.1)* — Інтеграція механізму глобального контролю за модифікацією. Чистка вихідного коду.
- *Special.SystemTests (1.3.6)* — Інтеграція механізму глобального контролю за модифікацією. Чистка вихідного коду.
- *UI.QTStarter (1.5.1)* — Стабілізація модуля. Інтеграція механізму глобального контролю за модифікацією. Чистка вихідного коду.
- *UI.QTCfg (1.6.0)* — Додано функцію уніфікованого завантаження та збереження. Додано функцію уніфікованого копіювання. Додано дії над об'єктом безпосередньо у дереві навігації. Додано контроль за модифікацією та попередження при виході. Стабілізація модуля. Чистка вихідного коду.
- *UI.VCAEngine (0.7.0)* — Реалізовано механізм сигналізації та повідомлення. Переглянуто механізм розподілення прав користувачів. Стабілізація та оптимізація модуля. Інтеграція до глобального механізму копіювання. Реалізовано механізм відкладеного вилучення. Інтеграція механізму глобального контролю за модифікацією. Чистка вихідного коду.
- *UI.Vision (0.7.0)* — Значне удосконалення примітиву "ElFigure". Удосконалення механізму слідування та попередження про модифікацію. Додано кеш ресурсів. Уніфіковано механізм розподілення користувацького доступу. Додано підтримку механізму сигналізації та повідомлення. Стабілізація та оптимізація модуля. Чистка вихідного коду.
- *UI.WebVision (0.4.0)* — Реалізація активного режиму роботи примітиву "ElForm". Додання обробки повідомлень клавіатури та миші. Додання підтримки фокусу. Чистка вихідного коду.
- *UI.WebCfg (1.5.1)* — Стабілізація та незначне удосконалення модуля. Чистка вихідного коду.

1. Підвищення стабільності, стійкості та продуктивності системи.

Під час робіт над цією версією, а також практичної адаптації, було виявлено та виправлено біля півтори сотні помилок. Біля чверті сотні з них було виявлено завдяки упередженому тестуванню Попковим Олексієм та біля десятка Калитою Андрієм. Значна кількість помилок була виявлена у компонентах середовища візуалізації та керування.

Крім виправлення безлічі помилок було проведено роботу по оптимізації різних компонентів OpenSCADA. Особливо помітної оптимізації піддалося середовище візуалізації та керування (СВК) та його візуалізатори. Основним вектором у оптимізації була оптимізація на продуктивність, питання якої почали спливати під час росту навантаження на систему.

2. Удосконалення та стабілізація СВКУ.

В процесі роботи з системою OpenSCADA було внесено багато удосконалень та оптимізацій до СВК та його візуалізаторів.

Розглянемо важливі удосконалення:

- Реалізація механізму сигналізації та повідомлення. Реалізація підтримки трьох методів сповіщення: візуальне, сигнальна та мовна. В цілому механізм інтегрований у концепцію СВК не є фіксованим та дозволяє реалізовувати сигналізацію за різними критеріями, а не тільки за традиційними для автоматизованих систем керування технологічними процесами (АСК ТП) критеріями.
- Для спрощення роботи з інспектором атрибутів було додано переперегляд шрифтів, зображень та кольорів, та діалоги прямої їх модифікації.
- До примітиву "ElFigure" додано:
 - Підтримку заповнення замкнених контурів кольором або зображенням. Підтримується також заповнення напівпрозорими кольорами.
 - Додано функцію копіювання всередині примітиву. Ця функція дозволить спростити завдання створення великих статичних зображень всередині одного віджету.
 - Додано підтримку масштабування товщини лінії та бордюру за мінімальним масштабом по x та y. Зробило процес масштабування більш адекватним та дозволило перемалювати багато елементів з використанням простих ліній.
- Уніфіковано та повністю реалізовано у Vision механізм розподілу прав користувачів. В результаті чого додано можливість зміни користувача у режимі виконання з відстеженням зміни прав доступу на окремі компоненти. Ця робота зробила реальністю можливість формування інтерфейсів користувача з розподілом прав доступу до окремих елементів між ними.
- Додана підтримка автоматичного скролінгу у контейнерах. Зроблено для можливості розміщення та адекватного представлення великих кадрів на інтерфейсах користувача.
- Реалізовано кеш ресурсів. Покликано підвищити продуктивність на високо-динамічних кадрах у яких часто змінюються ресурси у вигляді зображень та інших сутностей.

Важливі оптимізаційні удосконалення СВК – Vision:

- Приховування сторінок під час завантаження з повідомленням про завантаження у режимі виконання. Робить інтерфейс більш зрозумілим та прибирає питання під час достатньо тривалого відкриття великих кадрів.
- Оптимізація операцій переміщення та зміни розмірів віджетів. Підвищило реактивність інтерфейсу у режимі розробки.
- Оптимізовано механізм збереження індивідуальних даних примітивів. Дозволило досягти одразу двох цілей, перша ціль – оптимізація використання пам'яті та друга ціль – підвищення продуктивності формування примітивів інтерфейсу.

- Виконана оптимізація вилучення та копіювання-вставки великих груп віджетів у режимі розробки. Дозволило значно прискорити групові операції над віджетами у процесі створення високонавантажених кадрів.
- Операцію виділення мишею та прямокутником оптимізовано за часом. Тепер на не швидких обчислювальних системах не потрібно очікувати оновлення диспетчеру атрибутів після натиснення кнопки миші з метою виділення групи віджетів прямокутником, повідомлення вибору елементу генерується по відпусканню миші.

Виконано значне розширення UI.WebVison. А саме:

- Всі вигляди примітиву елементів форми (FormEl) стали активними, що дозволяє формувати форми взаємодії з користувачем через Web-інтерфейс.
- Реалізовано підтримку фокусу мишею та часткова підтримка клавіатурного фокусу. Додано можливість побудови повноцінних інтерфейсів візуалізації з навігацією та керуванням.

3. Загальносистемні розширення.

У ядро системи OpenSCADA було внесено деякі зміни розширюючого характеру. Перелічимо їх:

- Втілення механізму глобального контролю за модифікацією об'єктів динамічного дерева та, як наслідок, додання можливості вибіркового завантаження та збереження окремих об'єктів у гілках динамічного дерева. До мови контролю та керування OpenSCADA додано команди для відстеження стану модифікації гілок та окремих об'єктів, а також для їх збереження та завантаження. Цей механізм було втілено у QTCfg та Vison, що надало можливість користувачеві контролювати зміни та уникати втраті даних.
- Створено уніфікований механізм копіювання об'єктів динамічного дерева. Команди цього механізму було додано до мови контролю та керування OpenSCADA. На основі цих команд у конфігураторі QTCfg було реалізовано функцію прозорого копіювання об'єктів в дереві навігації.
- Реалізовано механізм відкладеного вилучення об'єктів динамічного дерева. Механізм призначено для забезпечення можливості повернення операції вилучення та використано для об'єктів рушія СВК.
- Додано функцію компресії з використанням бібліотеки ZLib. На основі цієї функції було реалізовано пакування трафіку протоколу "SelfSystem".
- Розширення мови контролю та керування:
 - Уніфікація інтерфейсу створення та вилучення об'єктів у вузлах динамічного дерева. На основі цього механізму в QTCfg додана функція додання та вилучення елементів дерева об'єктів безпосередньо у дереві навігації.
 - Додано атрибут для розміщення супровідної допомоги (атрибут "help" поля "fld"). Використано у конфігураторі QTCfg для створення спливаючих підказок.
 - Додано підтримку елементів статичного переліку, який призначено спростити роботу зі статичними списками у інтерфейсі.
 - Додано загальний інтерфейс для надання можливості вибору БД. Призначено для простої та уніфікованої, разом з спливаючою допомогою, інтеграцією комбобоксу вибору БД у об'єкти які цього потребують.
- Виконана адаптація для збірки на компіляторі gcc версії 4.3 та у оточенні дистрибутиву Mandriva 2008.1.
- Виконана адаптація до особливостей нової версії бібліотеки QT 4.4.
- Для більш гнучкого контролю за особливостями встановлення бібліотеки QT у скрипт конфігурації було додано параметр «--with-qt4-dir=DIR», який дозволяє вказувати місці розташування інсталяції QT.
- Виконана загальна чистка вихідного коду проекту від надлишкових пробілів та відхилень у стилі оформлення.

4. Вдосконалення частини модулів підсистеми «Збір даних».

Значному вдосконаленню підлягли модулі джерел даних “DAQ.Siemens”, “DAQ.JavaLikeCalc” та “DAQ.ModBus”.

У модуль “DAQ.Siemens” додана підтримка протоколу ISO_TSAP за посередництвом бібліотеки Libnodave. Це дозволило отримати доступ до широкого кола контролерів фірми Siemens за посередництвом мережі ProfiNET.

Мову високого рівня модуля “DAQ.JavaLikeCalc” було розширено операторами побітового зсуву, а потім було переписано механізм обробки чисельних значень. Як результат базові операції переведено на роботу тільки з реальними, що усунуло невизначеності у синтаксисі, які часто приводили до втрати знаку реальних чисел.

До модуля “DAQ.ModBus” було внесено багато змін. Значна частина їх пов'язана з роботами по тестуванню Попковим Олексієм. В цілому були внесені зміни розширення функціональних можливостей, стабілізації та оптимізації модуля. Розглянемо їх детальніше:

- Додано можливість встановлення часу відновлення зв'язку з недоступними контролерами на послідовних інтерфейсах. Використовується для зменшення навантаження на мережу RS485 за рахунок розрядження спроб відновити зв'язок.
- Додано можливість встановлення таймаутів RS485 мережі окремо для кожного контролера. Дозволяє підключати різноманітні контролери на послідовну шину та враховувати власні особливості часових таймаутів контролерів, оптимізуючи продуктивність мережі в цілому.
- Додана підтримка опитування та запису бітових параметрів за допомогою стандартних функцій 01 та 05, а також вхідних регістрів (04) та вхідних бітів (02). Таким чином реалізована підтримка всіх стандартних множин параметрів протоколу ModBus.
- Додано власний лог протоколу обміну. Дозволяє вирішувати багато проблем комунікації шляхом відстеження реальних запитів та відповідей.

5. Плани подальшого розвитку

До релізу наступної версії заплановане вирішення таких завдань:

- Реалізація підтримки групових запитів у мові контролю та керування OpenSCADA та оптимізація на його основі мережових комунікацій різних підсистем OpenSCADA.
- Розробка та реалізація концепції примітиву CBK “Document” призначеного для формування звітної документації.
- Міграція модуля конфігурації системи OpenSCADA UI.WebVision на динамічний інтерфейс DHTML.
- Реалізація безпечного транспорту на основі OpenSSL.
- Реалізація примітивів CBK “Link” та “Function”.