

OpenSCADA 0.7.0

Показник

OpenSCADA 0.7.0	1
Вступ	1
1 Реалізація планових завдань	3
2 Оптимізація, підвищення стабільності, стійкості та продуктивності системи	3
3 Удосконалення та стабілізація СВК	5
4 Формування та стабілізація API програмування користувача	6
5 Загальносистемні розширення	7
6 Значні удосконалення та виправлення окремих модулів	8
7 Формування документації та переклади документації і інтерфейсу	9
8 Уніфікація демонстраційної БД	9
9 Рішення, які побудовано на основі OpenSCADA	9
Висновок	10
Посилання	10

Вступ

Реліз відкритої SCADA(Supervisory control and data acquisition) системи версіях 0.7.0 є першим стабільним-промисловим релізом, що пов'язано з завершенням стадії бета-тестування, розпочатої з версії 0.6.4.

Основною метою цього релізу є надання співтовариству користувачів та розробників вільного програмного забезпечення (ПЗ) платформи для побудови рішень комплексних систем автоматизації та інших суміжних рішень, а також надання комерційних послуг на основі проекту OpenSCADA.

Цей реліз є першим стабільним релізом, для якого надається технічна підтримка від розробників та для якого планується випуск виправлень протягом довгого проміжку часу.

Цей документ становить обробку(компіляцію) документа "ChangeLog" системи OpenSCADA версії 0.7.0, який прикликано 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://wiki.oscada.org/Works/ChangeLog>.

Ключовими особливостями даної версії є:

- Реалізація планових завдань.
- Оптимізація, підвищення стабільності, стійкості та продуктивності системи.
- Удосконалення та стабілізація СВК.
- Формування та стабілізація API програмування користувача.
- Загальносистемні розширення.
- Значні покращення та виправлення окремих модулів.
- Формування документації та перекладі документації і інтерфейсу.
- Уніфікація демонстраційної БД.

Нові та оновлені модулі:

- Archive.FSArch - 1.3.1 -> 1.4.1, виправлення та покращення для підвищення продуктивності
- Archive.DBArch - 0.9.0 -> 0.9.2, виправлення та деякі покращення
- DB.DBF - 2.0.1 -> 2.0.2, виправлення
- DB.MySQL - 1.6.1 -> 1.6.2, деякі покращення
- DB.SQLite - 1.6.1 -> 1.6.2, покращення

- DB.FireBird - 0.9.1 -> 0.9.5, значні покращення
- DB.PostgreSQL - 0.9.0, новий модуль
- DAQ.DiamondBoards - 1.2.0 -> 1.2.1, адаптація до змін API
- DAQ.System - 1.7.0 -> 1.7.2, очищення коду та адаптація до змін API
- DAQ.BlockCalc - 1.2.1 -> 1.4.0, покращення та виправлення
- DAQ.JavaLikeCalc - 1.5.0 -> 1.8.0, значні покращення, виправлення та оптимізація
- DAQ.LogicLev - 1.0.1 -> 1.1.2, покращення та виправлення
- DAQ.SNMP - 0.4.0 -> 0.4.1, адаптація до змін API
- DAQ.Siemens - 1.2.1 -> 1.2.3, виправлення та адаптація до змін API
- DAQ.ModBus - 1.0.1 -> 1.1.1, виправлення та покращення
- DAQ.DCON - 0.3.0 -> 0.3.3, виправлення та покращення
- DAQ.ICP_DAS - 0.5.0 -> 0.7.2, значні покращення та виправлення
- DAQ.DAQGate - 0.8.0 -> 0.9.1, значні покращення та виправлення
- DAQ.SoundCard - 0.6.0 -> 0.6.1, адаптація до змін API
- DAQ.OPC-UA - 0.6.0, новий модуль
- Transport.Sockets - 1.4.0 -> 1.4.5, покращення та виправлення
- Transport.SSL - 0.8.2 -> 0.9.5, покращення та виправлення
- Transport.Serial - 0.5.0 -> 0.7.1, значні покращення та виправлення
- Protocol.HTTP - 1.4.1 -> 1.5.0, значні покращення
- Protocol.SelfSystem - 0.9.1 -> 0.9.3, виправлення та адаптація до змін API
- Protocol.UserProtocol - 0.6.0, новий модуль
- Protocol.ModBus - 0.5.0 -> 0.6.1, виправлення та покращення
- Protocol.OPC-UA - 0.6.0, новий модуль
- Special.FLibComplex1 - 1.0.4 -> 1.0.6, виправлення та покращення
- Special.FLibMath - 0.5.1 -> 0.5.2, адаптація до змін API
- Special.FLibSYS - 0.9.0 -> 0.9.2, виправлення та покращення
- Special.SystemTests - 1.3.7 -> 1.5.0, переробка та уніфікація тестів
- UI.QTStarter - 1.5.3 -> 1.6.0, покращення
- UI.QTCfg - 1.8.1 -> 1.9.0, значні покращення та виправлення
- UI.WebCfg - 1.5.3 -> 1.5.4, виправлення та адаптація до змін API
- UI.WebCfgD - 0.6.2 -> 0.6.5, виправлення
- UI.VCAEngine - 0.9.0 -> 1.0.0, значні покращення, виправлення та оптимізація
- UI.Vision - 0.9.0 -> 1.0.0, значні покращення, виправлення та оптимізація
- UI.WebVision - 0.7.0 -> 0.9.0, значні покращення та виправлення
- UI.WebUser - 0.6.0, новий модуль

1 Реалізація планових завдань

У відповідності з планом релізу було виконано наступні завдання:

- *Реалізація примітивів "Зв'язок" та "Функція" СВК.* - У зв'язку з не принциповістю для промислового релізу та відсутності потреби, на цей час, це завдання було винесено за межі першого промислового релізу.
- *Стабілізація та випуск промислової гілки OpenSCADA.* - Зроблено велику роботу по стабілізації та оптимізації промислового релізу, про що детальніше о у наступному розділі.
- *Реалізація модуля БД ОДБС, PostgreSQL та транспортного модуля БД.* - Максимом Лисенком було виконано реалізацію модуля БД "PostgreSQL". Реалізація модуля БД "ODBC" було визнано не доцільним оскільки технологія "ODBC" фактично є інтерфейсом SQL-запитів, який не враховує особливостей SQL-діалектів реальних БД, а це означає що він не може безпосередньо, без огляду на тип БД, використовуватися у OpenSCADA при вирішенні її основних завдань. Перспектива використання "ODBC" залишається для реалізації доступу до БД у яких немає "рідного" API прямого доступу до БД, але є драйвер ODBC, або для надання простого інтерфейсу SQL-запитів у OpenSCADA. Реалізацію транспортного модуля БД було винесено за межі першого промислового релізу.
- *Реалізація нового механізму/модуля тестів системи OpenSCADA.* - Для виконання внутрішніх системних тестів OpenSCADA механізм модуля тестування було виконано у вигляді функції API користувача, що з одного боку зберегло можливість автономного виклику статичних тестів та з іншого боку надало можливість виклику тестів по одному, в ручну, а також з скриптів користувача, у будь якій послідовності та у потрібній кількості окремих завдань.

2 Оптимізація, підвищення стабільності, стійкості та продуктивності системи.

У процесі работ над цією версією, а також її практичної адаптації, було виявлено та виправлено в загальній складності біля 200 помилок.

Крім виправлення багатьох помилок була зроблена робота по оптимізації різних компонентів OpenSCADA. Особливо помітної оптимізації все ще було піддано середовище візуалізації та керування (СВК) та його візуалізатори.

Перелічимо більш істотні помилки виправлення яких значно відобразилося на підвищені стабільності:

- *Загальносистемні:*
 - Виправлено не коректне встановлення політики реального часу для потоків. Не було встановлено прапор потоку PTHREAD_EXPLICIT_SCHED.
 - Виправлено контроль за подвійним захопленням ресурсу у одному потоці та захоплення ресурсу за таймаутом.
 - Механізм запобігання багаторазового запуску виправлено на предмет перевірки PID заблокованого процесу, по причині видалення системою lock-файлів з директорії /tmp.
 - Виключено пакування даних структури класів за допомогою #pragma pack(push,1) по причині появи неявної помилки із зависанням одного з потоків та блокування залишених у прошивці OpenSCADA для ПЛК LP8x81.
 - Код ініціалізації та оновлення модулів оновлено для виключення подвійної ініціалізації поділяємих бібліотек.
 - Функцію створення завдання виправлено для запобігання падіння під час створення автоматично-зачиняємих завдань, під час швидкого закриття.
 - Виправлено зависання функції cron() у випадку встановлення порожнього розкладу.

- *Транспорту:*
 - *Serial, DAQ.DCON, DAQ.ModBus:* Транспорт послідовного інтерфейсу оптимізовано, зроблено типовим та виправлено на предмет декількох помилок. Клієнтів транспорту адаптовано для використання нового механізму.
 - *Sockets, SSL:* Виправлено можливість не повного запису функцією write(). Під час запису до сокету ця функція пише тільки частину розміром у свій буфер (50кБ). Для повного запису великих повідомлень додано повтор запису.
 - *Transport.SSL:* Остаточо виправлено очікування відповіді серверу шляхом доочікування обробки даних у функції BIO_read().
 - *Transport.Sockets:* Виправлено обробку нульового результату очікування хвосту повідомлень, для виключення зависання на запитах у DAQ.System.HDDTemp.
 - *Transport.Serial:* Виправлено використання прапора "CSTOPB" для встановлення кількості стоп-бітів.
- *Збір даних:*
 - *ModBus:* Виправлено втрату ресурсу під час збору даних. З цієї причини виключено можливість привласнення строки безпосередньо до об'єкту ресурсної строки.
 - *JavaLikeCalc:* Виправлено використання функцій для параметрів контролерів DAQ з IO функцій для першого використання.
 - *DAQ, BlockCalc, JavaLikeCalc, LogicLev, Siemens:* Виправлено отримання помилкових даних за схемою резервування у не резервованих завдань та при відсутності резервних станцій.
- *Protocol.HTTP, UI.WebCfg, UI.WebCfgD, UI.WebUser, UI.Vision:* Значення HTTP-параметру "Content-Type" виправлено для елемента "Charset".
- *UI.VCAEngine:* Виправлено падіння рушія візуалізації під час зміни мутованих атрибутів, які багаторазово успадковано.

Оптимізація та підвищення продуктивності:

- *SYS.XML:* Функцію перетворення дерева XML у строку оптимізовано за часом.
- *SYS.Archive:* Додано механізм адаптивної генерації запитів до архівів різної якості, у випадку загального запиту. Запобігає тривалій обробці у випадку запиту великих проміжків часу та наявності менш якісних архівів.
- *DAQ, DAQ.DAQGate:* Виконано оптимізацію запитів між резервними станціями за розміром запитів та за часом. Запитуються тільки архівні атрибути параметрів DAQ та атрибути, заплановані до запиту.
- *Archive.FSArch:* Додано створення інформаційних файлів упакованих архівів, без збереження цих даних у БД. Ця функція дозволяє забезпечити швидке підключення великих архівів до інших станцій.
- *UI.Vision:* У примітиві "ElFigure", для відображення, замість QImage використано QPixmap, що значно підвищило продуктивність.
- *UI.Vision:* Для побудови зображення примітиву "Diagram" використано QImage замість QPixmap, що значно підвищило продуктивність.
- *TFunction, DAQ.JavaLikeCalc:* Додано вкладений контекст функцій з метою підвищення продуктивності зовнішніх викликів.

Оптимізація використання пам'яті:

- *SYS:* Систему переключено на використання традиційного "map" замість "hash_map" у всіх об'єктах з метою зменшення споживання пам'яті.
- *UI.VCAEngine:* Об'єкт атрибуту "Attr" відірвано від об'єкту "TCntrNode" з метою зменшення споживання пам'яті до 25%.

3 Удосконалення та стабілізація СВК.

Помітні зміни було внесено у межах середовища візуалізації та керування (СВК), а саме у модулях рушія СВК *UI.VCAEngine*, візуалізаторів *UI.Vision* та *UI.WebVision*. Внесені зміни було спрямовано на стабілізацію, оптимізацію споживання пам'яті та покращення споживчих властивостей СВК.

Покращення СВК:

- Додано повну підтримку вертикального масштабування діаграм.
- Додано підтримку альфа-каналу у атрибути кольору всіх примітивів.
- *UI.VCAEngine*:
 - Включено та зроблено безпечним успадкування опису мутуючих атрибутів.
 - Додано нові функції API користувача для сеансів *wdgAdd()*, *wdgDel()* та *link()*, *linkSet()*. Додання цих функцій дозволило реалізувати концепцію повністю динамічного формування інтерфейсу користувача.
 - Властивості прав доступу до віджету реалізовано атрибутами "owner" та "perm" для підвищення гнучкості керування.
 - Додано механізм автоматичного створення та запуску сеансу під час завантаження.
- *UI.Vision*:
 - До інспектору атрибутів додано можливість групового встановлення однойменних атрибутів різних віджетів. Встановлення відбувається у груповому контейнері властивостей інспектору атрибутів, який з'являється під час виділення одночасно декількох віджетів.
 - У примітиві "ElFigure", для відображення замість QImage використано QPixmap, що значно підвищило продуктивність.
 - Для побудови зображення примітиву "Diagram" використано QImage замість QPixmap, що значно підвищило продуктивність.
 - Додано підтримку властивостей інтерфейсу користувача. Додано керування та збереження розміру іконок панелей інструментів.
 - Додано можливість копіювання значень атрибутів та зв'язків за посередництвом контекстного меню інспекторів атрибутів та зв'язків.
 - Додано функцію пошуку по елементу "TextEdit".
- *UI.WebVision*:
 - Додано підтримку обертання примітиву "Text".
 - Додано генерацію основних повідомлень миші. Для обробки деяких повідомлень додано стек функцій обробників.
 - Для користувача-відправника адаптовано доступ та керування до його сеансу.

4 Формування та стабілізація API програмування користувача.

Значним чином було виконано формування нового об'єктного API програмування користувача, яке передбачає інтеграцію функцій користувача у дерево об'єктів системи OpenSCADA. Крім цього було внесено низку змін до існуючих бібліотек функцій API користувача.

Зокрема було внесено наступні зміни:

- *SYS*:
 - *SYS, Special.FLibSYS*: Об'єкт API користувача XMLNodeObj переміщено до ядра OpenSCADA (файли tvariant.h, tvariant.cpp).
 - Системне API користувача OpenSCADA, яке засновано на об'єкті TCntrNodeObj, розширено для контролю прав доступу користувача.
 - Об'єкт TAreaObj перейменовано у TArrayObj.
 - Властивості об'єктів, виклики функцій та обробка виключень API користувача уніфіковано.
 - *SYS.TVarObj*: Додано функцію propList() для можливості сканування атрибутів.
- *DAQ.JavaLikeCalc*:
 - Додано функції до API користувача: insert(), replace(), toReal(), toInt(), parse(), parsePath() та path2sep() до об'єкту значень строкового типу.
 - Додано функцію isEval() до базових типів, з метою виявлення EVAL-значень.
- *Special.FLibComplex1*: Функція ПІД розширено новими параметрами: Kd, Tzd та followSp. Видалено параметр Tf.
- *Об'єктне API програмування користувача*:
 - *SYS*: Функції API користувача: system(), message(), XMLNode(), cntrReq(), time(), localtime(), strftime(), strptime(), cron(), messDebug(), messInfo(), messNote(), messWarning(), messErr(), messCrit(), messAlert(), messEmerg() та strFromCharCode() додано до кореневого об'єкту ядра OpenSCADA.
 - *DB*: Функцію API користувача SQLReq() додано до об'єкту БД.
 - *Transport*: Функцію API користувача messIO() додано для об'єкту вихідного транспорту.
 - *UI.VCAEngine*:
 - Додано функції API користувача: user(), alrmSndPlay() та alrmQuittance() до об'єкту сеансу проекту.
 - До об'єкту "Widget" додано функції API користувача: ownerSess(), ownerPage(), ownerWdg(), attrPresent(), attr(), attrSet(), wdgAdd(), wdgDel(), link() та linkSet().
 - Додано спеціальний атрибут "this" до всіх скриптів СВК з метою доступу до об'єкту віджету та наступного виклику його функцій API користувача.
 - *Archives*: Функцію API користувача messGet() додано для об'єкту підсистеми "Архіви".
 - *DAQ.Value*: Додано функції API користувача get() та set() для атрибутів.

5 Загальносистемні розширення.

До загальносистемного API системи OpenSCADA було внесено значні зміни та розширення з метою підвищення функціональності та загальної стабілізації:

- *SYS*:
 - Додано механізм запобігання повторному запуску у скрипти: `demo-start` та `user-start`.
 - Додано глобальні функції для створення(реєстрації) `taskCreate()` та закриття `taskDestroy()` потоків. Всі компоненти OpenSCADA використовують нове API.
 - Додано файли `debian/*` для підтримки побудови Debian пакетів.
 - Додано функцію `strParse()` для розширення функції `strSepParse()` на предмет підтримки багатосимвольних роздільників та поєднання односимвольних.
 - Додано інфраструктуру лічильників налагодження.
 - Додано можливість призначення процесорів для потоків OpenSCADA на багато-процесорних системах та багато-ядерних процесорах.
 - Додано менеджер завдань. Додано поле ідентифікатору процесу потоку до менеджера завдань.
 - Для реалізації специфічних функцій додано обгортку над завданнями OpenSCADA. Додано обробку виключень "TError" для завдань всередині обгортки.
 - Код ядра системи OpenSCADA поміщено у область імен "OSCADA". Версія бібліотеки ядра OpenSCADA підвищено до 1:0:0.
 - Уніфіковано права доступу до всіх елементів інтерфейсу керування.
 - Додано перевірку на версію більш за 2.0 утиліти збору `libtool`. Включено збір тільки поділяємих бібліотек для модулів.
 - Ідентифікатор модуля розташовано до класу конструктору з метою спрощення ініціалізації перекладу текстових повідомлень модулів.
 - Для всіх перекладів використано утиліту `xgettext "-C --no-location --no-wrap -k_"`. Використано для очищення сміття файлів перекладів.
 - Версії API всіх модульних підсистем встановлено у 5.
- *SYS.Function*, *DAQ.BlockCalc*, *DAQ.LogicLev*, *UI.VCAEngine*: Додано механізм контролю за змінами параметрів контролерів підсистеми "Збір даних".
- *DB*:
 - Додано автоматичне відкриття наявних таблиць за запитом до них.
 - Додано керування транзакціями до функції `sqlReq()`, а також додано функцію `transCloseCheck()`.
- *TFunction*, *DAQ.JavaLikeCalc*: Додано вкладений контекст функцій з метою підвищення продуктивності зовнішніх викликів.
- *DAQ*: Код помилки додано до стану контролера, що дозволяє ідентифікувати стан та відображати його.
- *Protocol*, *Transport*, *DAQ.ModBus*, *Protocol.UserProtocol*: Додано функцію `itemListIn()` для вибору елементів протоколу. Додано вибір елементів протоколу за посередництвом функції `TProtocol::itemListIn()`.

6 Значні удосконалення та виправлення окремих модулів.

В процесі роботи над цим промисловим релізом під значні зміни, удосконалення та стабілізації підпали окремі модулі OpenSCADA:

- *DB.MySQL*:
 - Додано підтримку необмежених значень типів даних "Реальне" та "Ціле".
 - Додано перетворення значення EVAL для реального типу.
- *DB.SQLite*: Покращено механізм транзакцій для підтримки всередині SQL-запитів користувача та закриття старих транзакцій.
- *DB.FireBird*: Додано підтримку багатомовності текстових змінних. Уніфіковано підтримку транзакцій.
- *DAQ*:
 - *DAQGate*: Додано підтримку можливості запиту окремих атрибутів. Використано для запиту архівованих та часто запитуваних атрибутів. Всі інші атрибути запитуються з періодом синхронізації шляхом повного запиту. Використано для зменшення навантаження та часу протягом обміну DAQGate з віддаленої OpenSCADA станції.
 - *ICP_DAS*: Додано стан DIP-перемикача для ПЛК LP-8781. Функції відправлення повідомлень за послідовним інтерфейсом переміщено у загальну функцію serReq() та всі запити використовують її.
 - *ModBus*: Додано суфікси "i2,i4,f,b0" до реєстрів (R та RI) для гнучкого доступу за різними типами даних.
 - *JavaLikeCalc*:
 - Додано підтримку оглядового циклу для сканування властивостей об'єкту "for(<i> in <obj>) <code>;".
 - Властивості об'єктів, виклики функцій та обробка виключень API користувача уніфіковано.
- *Transport*:
 - *Serial*:
 - Додано можливість вибору типу керування потоком. Підтримується апаратне (CRTSCTS) та програмне (IXON IXOFF) керування потоком.
 - Додано підтримку режиму модема для всіх вхідних та вихідних транспортів.
 - *SSL*:
 - Додано підтримку KeepAlive лімітів. Додано підтримку перепідключення після втрати з'єднання та для запобігання втрати повідомлень.
 - Додано отримання адреси відправника.
 - *SSL, Sockets*: До вихідного транспорту додано таймінги з метою окремого керування часом з'єднання та часом очікування наступних відповідей.
 - *Protocol.HTTP*:
 - Додано функцію вихідного протоколу.
 - Додано підтримку автоматичного входу.
- *Archive.FSArch*: Додано створення інформаційних файлів упакованих архівів, без збереження цих даних у БД. Ця функція дозволяє забезпечити швидке підключення великих архівів до інших станцій.
- *UI.QTcfg*:
 - Додано підлаштування рядків для таблиць, після додання нових рядків. Додано підтримку копіювання вмісту таблиці до буферу обміну.
 - Додано функцію пошуку у елементу "TextEdit".
 - Додано можливість копіювання переліку вибраних вузлів.

7 Формування документації та переклади документації і інтерфейсу.

Оскільки цей реліз позиціонується як промисловий реліз з тривалим періодом підтримки то до його виходу було закінчено написання, на Російській мові, а також переклад на Англійську мову, всієї основної документації. Крім того завершено переклад інтерфейсу програми на Англійську, Російську, Українську та Німецьку мови:

- Додано переклад README на Українську.
- Додано переклад документу DAQ та узагальнюючого документу на Англійську.
- Додано документ "Швидкий старт".
- Виконано переклад документу "Швидкий старт" на Англійську мову.
- Загальний документ "OpenSCADA" на Англійській оновлено для включення частин "Швидкий старт" та DAQ.OPC-UA.
- Оновлено переклад інтерфейсу ядра системи на Російську, Українську та Німецьку мови.
- Для всіх перекладів використано параметри утиліти xgettext "-C --no-location --no-wrap -k_". Використано для очищення файлів перекладу від сміття.

8 Уніфікація демонстраційної БД.

У процесі підготовки промислового релізу значну увагу було приділено покращенню демонстраційної БД, як яскравої демонстрації функцій системи OpenSCADA:

- Значне оновлення для підтримки нових можливостей та включення нових удосконалень з системи автоматизації кульових млинів.
- Додано підтримку користувальницького виклику демонстраційної БД у повноцінному режимі на запис.
- Додано масштабування існуючих елементів у випадку відсутності рядків та стовпчиків для оглядового кадру та зведеної таблиці.
- Оновлено для включення нових моделей апаратів технологічних процесів та адаптації всіх інших моделей.
- Оновлено для:
 - Додання об'єктів сигналізації компресорів: KM201, KM301, KM202 та KM302.
 - Підтримка ручного вводу змінних з панелі керування.
 - Оновлення перекладу.
 - виправлення декількох помилок.
- Скрипт головної сторінки проекту переписано для використання нових функцій API користувача.
- Використано новий механізм доступу до елементів інтерфейсу користувача. Надано реальний доступ на читання для пересічного користувача ("user").

9 Рішення, які побудовано на основі OpenSCADA

На основі проекту OpenSCADA побудовано фактично весь спектр рішень автоматизації, які яскраво демонструють можливості, а також становлять сигнал готовності системи для широкого промислового застосування.

В цілому на основі OpenSCADA авторами проекту вирішувалися завдання: повного динамічного моделювання технологічних процесів (ТП), управління ТП на рівні програмованого логічного контролеру (ПЛК) та формування людино-машинного інтерфейсу ТП.

Моделювання ТП представлено рішеннями: «Бібліотека моделей апаратів технологічних процесів», «Динамічна модель реального часу Анастасіївської ГЛКС» та «Динамічна модель парового котлу №9 ДМК». Причому рішення «Динамічна модель реального часу Анастасіївської ГЛКС» було покладено у основу демонстраційної БД проекту та яка розповсюджується зі всіма

дистрибутивами OpenSCADA.

Сфера управління ТП на рівні програмованого логічного контролера (ПЛК) представлена рішеннями: «OpenSCADA у програмованому логічному контролері (ПЛК)» та «Середовище виконання ПЛК LP-8x81 фірми ICP DAS на основі OpenSCADA».

Сферу формування людино-машинного інтерфейсу ТП представлено рішеннями: «Бібліотеки графічних елементів інтерфейсів користувача» та «АСУ ТП кульовими млинами ШБМ 287\410 котлоагрегату БКЗ 160–100 ПТ».

Потрібно також відзначити наявність низки рішень, які не увійшли до вищенаведеного переліку, та виконаних розробниками системи для третіх осіб або безпосередньо її користувачами.

Висновок

Система OpenSCADA готова для практичного використання у складних промислових умовах та виконання широкого спектру завдань автоматичного управління технологічними процесами.

З метою надання якісного сервісу на основі рішень OpenSCADA, а також розвитку у напрямку підтримки спеціалізованого обладнання, протоколів та інтерфейсів; розробниками планується, та активно ведеться, робота по формуванню політики надання послуг на комерційній основі. У число комерційних послуг буде входити: технічна підтримка, консультації та цільова розробка розширень, інтеграція та впровадження рішень, а також розробка спеціалізованих рішень та продуктів на основі OpenSCADA.

Версія 0.7.0 надає рішення OpenSCADA тільки для платформ Linux x86 та x86_64. Для забезпечення підтримки інших апаратних та програмних платформ, запланованих проектом, роботу буде продовжено. Так до версії 0.8.0 планується розширити підтримку апаратних платформ мінімум на платформу АРМ, як представника RISC архітектур.

У появі першої промислової версії системи OpenSCADA активну участь брали:

- [Савоченко Роман](#): Основний об'єм робіт по проектуванню, розробці, документуванню та тестуванню.
- [Лисенко Максим](#): Розробка примітиву "Елементарна фігура" середовища візуалізації та керування (СВК), створення модуля підтримки БД PostgreSQL та переклад більшої частини документації на Англійську мову.
- [Яшина Ксенія](#): Розробка примітиву "Елементарна фігура" СВК.
- [Попкова Ірина](#): Переклад інтерфейсу системи OpenSCADA та її модулів на Німецьку мову.
- [Алмаз Каримов](#): Розробка модуля підтримки протоколу DCON та активне тестування.
- [Попков Олексій](#): Активне тестування та участь.
- Багато інших користувачів системи OpenSCADA, за посередництвом всебічного тестування.

Посилання

Протокол тестування промислового релізу: <http://wiki.oscada.org/Works/Tests/release070>