

OpenSCADA 0.8.0 LTS

(Зауваження до релізу)

Показник

Вступ	1
1 Реалізація планових задач	4
2 Оптимізація, підвищення стабільності, стійкості та продуктивності системи	5
3 Удосконалення та стабілізація графічної підсистеми	8
4 Формування, розширення та стабілізація API користувацького програмування	10
5 Загальносистемні розширення	11
6 Публікація рішень OpenSCADA	14
Заключення	15
Посилання	15

Вступ

Реліз відкритої SCADA(Supervisory control and data acquisition) системи версії 0.8.0 є стабільним промисловим релізом тривалої підтримки (LTS).

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

Цей реліз є наступним стабільним релізом, для якого надається технічна підтримка від розробників та для якого планується випуск виправлень на протязі тривалого часу. Життєвий цикл попереднього стабільного релізу 0.8.0 тривалої підтримки (LTS) буде припинено протягом місяця після випуску 0.8.0, останнім поновленням.

Цей документ є обробкою(компіляцією) документа "ChangeLog" системи OpenSCADA версії 0.8.0, включаючи проміжні робочі релізи 0.7.1 та 0.7.2, які покликані коротко та наочно освітлити нові можливості системи OpenSCADA. Детально ознайомитися із змінами можна у файлі "ChangeLog" з дистрибутиву системи або тут: <http://wiki.oscada.org/Works/ChangeLog>.

Ключовими особливостями цієї версії є:

- Реалізація планових задач.
- Оптимізація, підвищення стабільності, стійкості та продуктивності системи.
- Удосконалення та стабілізація графічної підсистеми.
- Розширення API користувацького програмування.

- Формування, розширення та стабілізація API користувачького програмування.
- Загальносистемні розширення.
- Публікація рішень OpenSCADA.

Нові та оновлені модулі:

- *Archive.DBArch (0.9.5)* — Додано ліміти запиту даних за часом запиту та ще деякі розширення.
- *Archive.FSArch (1.5.1)* — Значна стабілізація. Багато розширень, у том числі:
 - Підвищення продуктивності читання індексів файлів архівів значень.
 - Адаптація збереження та читання реального до ARM FPA.
 - Додані ліміти запиту даних за часом запиту.
 - Додано ліміт на загальний розмір файлів архіватору значень.
- *DB.MySQL (1.7.1)* — Стабілізація. Додано можливість встановлення таймаутів підключення.
- *DB.SQLite (1.6.4)* — Стабілізація.
- *DB.FireBird (0.9.7)* — Стабілізація.
- *DB.PostgreSQL (0.9.2)* — Стабілізація.
- *DAQ.DiamondBoards (1.2.5)* — Додано можливість зміни типу параметра. Стабілізація.
- *DAQ.System (1.7.5)* — Стабілізація. Залежність від бібліотеки "Libsensors" зроблено опціональною. Додано планування викликів за CRON.
- *DAQ.BlockCalc (1.6.0)* — Стабілізація. Багато розширень, в том числі:
 - Додано підсвічення синтаксису.
 - Додано планування викликів за CRON.
 - Додано запускаючий та зупиняючий виклики блоків.
 - Додано підтримку об'єктного типу параметрів.
- *DAQ.JavaLikeCalc (2.0.0)* — Значна стабілізація. Багато розширень, в тому числі:
 - Додано підсвічення синтаксису.
 - Виконано адаптацію до апаратної архітектури ARM.
 - Додано реалізацію регулярних виразів та низку розширень API користувача.
 - Додано можливість запису символів рядка за допомогою вісімкового "\041" та шістнадцяткового "\x21" числа.
 - Додано прямий, динамічний виклик бібліотечних функцій.
 - Додано пряме об'єднання рядкових констант.
 - Додано умовний виклик аргументів виразу (другого аргументу) з логічними операціями ||(OR) та &&(AND).
 - Додано підтримку об'єктного типу параметрів джерела даних.
- *DAQ.LogicLev (1.3.0)* — Стабілізація. Багато розширень, в том числі:
 - Додано можливість зміни типу параметра.
 - Додано планування викликів за CRON.
 - Додано підтримку об'єктного типу параметрів.
- *DAQ.SNMP (0.7.0)* — Значна стабілізація. Багато розширень, у тому числі:
 - Реалізація повної підтримки клієнтської частини протоколу.
 - Додано підтримку всіх типів значень.
 - Додано підтримку запису значень.
 - Покращено обробку помилок.
 - Додано додаткові параметри конфігурації підключення.
 - Додано можливість прямого запиту скалярів.
- *DAQ.Siemens (1.4.0)* — Значна стабілізація. Багато розширень, у том числі:
 - Версію бібліотеки LibnoDave оновлено до 0.8.4.6.
 - Додано планування викликів за CRON.
 - Додано функцію перепідключення для "Industrial Ethernet" з'єднань.
 - Додано варіант протоколу ISO_TCP для S7-200.
 - Додано підтримку об'єктного типу параметрів.
 - Додано реалізацію протоколу "ADS".
 - Деяко уніфіковано помилки підключення.
 - Додано можливість встановлення номеру БД та зсуву у шістнадцятковому вигляді.
- *DAQ.ModBus (1.3.0)* — Значна стабілізація. Багато розширень, в тому числі:

- Додано підсвічування синтаксису переліку атрибутів.
- Додано підтримку типу параметра "Логічний" для роботи за шаблоном параметру, а також функції користувачького API для надсилання довільних-нестандартних ModBus-запитів із шаблонів.
- Додано та виправлено функцію формування повідомлень про порушення у контролері.
- Додано підтримку функцій групового запису (0x0F, 0x10).
- Додано параметр встановлення обмеження розміру блоку групових запитів.
- Додано підтримку об'єктного типу параметрів.
- Розмір максимального блоку запиту зроблено опціональним.
- Додано можливість коментування елементів переліку атрибутів, стандартного типу параметра, за посередництвом символу '#' у початку.
- *DAQ.DCON (0.5.1)* — Модуль повністю переписана на предмет підтримки змішаних модулів, за посередництвом обрання команд запитів. Стабілізація.
- *DAQ.ICP_DAS (0.8.0)* — Додано бібліотеку API "ICP DAS" libi8k.a для архітектури ARM та включено можливість збірки модуля для ARM. Збірка по замовченню виключено для попередження збірки на непридатному ARM оточені. Стабілізація.
- *DAQ.DAQGate (0.9.5)* — Додано кеш параметрів для їх ініціалізації під час автономної роботи. Додано планування викликів за CRON. Стабілізація.
- *DAQ.SoundCard (0.6.2)* — Стабілізація.
- *DAQ.OPC-UA (0.6.2)* — Стабілізація. Додано підсвічування синтаксису переліку атрибутів. Адаптація збереження та читання реального до ARM FPA.
- *DAQ.BFN (0.5.1)* — Новий модуль.
- *Protocol.ModBus (0.6.4)* — Стабілізація. Деякі покращення.
- *Protocol.OPC-UA (0.6.2)* — Стабілізація.
- *Transport.Sockets (1.5.1)* — Значна стабілізація. Додано таймаут повтору запита вихідного транспорту.
- *Transport.SSL (1.0.1)* — Стабілізація. Деякі удосконалення.
- *Transport.Serial (0.8.0)* — Стабілізація. Під час ініціалізації порту використано попередні налаштування та реалізовано можливість опустити визначення деяких налаштувань. Додано керування потоком за допомогою сигналу RTS для простих перетворювачів сигналів RS232->RS485.
- *Protocol.HTTP (1.6.0)* — Стабілізація. Багато розширень, у тому числі:
 - Додано підтримку змінної HTTP "Transfer-Encoding=chunked" у вихідному транспорті.
 - Додано підтримку користувачьких шаблонів для внутрішнього вмісту модуля.
 - Додано генерацію повідомлення автентифікації користувачів.
 - Додано підтримку всіх основних варіантів завершення рядка під час розбору HTTP-запиту.
- *Protocol.SelfSystem (0.9.5)* — Стабілізація.
- *Protocol.UserProtocol (0.6.2)* — Додано підсвічування синтаксису. Стабілізація.
- *Special.FLibComplex1 (1.1.0)* — Додано прямий, динамічний виклик бібліотечних функцій.
- *Special.FLibMath (0.6.0)* — Додано прямий, динамічний виклик бібліотечних функцій.
- *Special.FLibSYS (1.0.0)* — Додано прямий, динамічний виклик бібліотечних функцій.
- *Special.SystemTests (1.5.1)* — Невеликі удосконалення.
- *UI.QTStarter (1.7.0)* — Стабілізація. Значні розширення та удосконалення.
- *UI.QTCfg (2.1.1)* — Стабілізація. Значні розширення та удосконалення.
- *UI.WebCfg (1.5.6)* — Стабілізація.
- *UI.WebCfgD (0.8.1)* — Стабілізація. Значні розширення та удосконалення.
- *UI.VCAEngine (1.3.0)* — Стабілізація. Значні розширення та удосконалення.
- *UI.Vision (1.3.0)* — Стабілізація. Значні розширення та удосконалення.
- *UI.WebVision (1.0.1)* — Значна стабілізація. Значні розширення та удосконалення.
- *UI.WebUser (0.6.2)* — Стабілізація. Додано підсвічування синтаксису процедур.

1 Реалізація планових задач

У відповідність з планом релізу було виконано наступні завдання:

- *Формування надання комерційних послуг на основі OpenSCADA.* - Побудовано концепцію та створено механізми надання комерційних послуг розробниками на основі OpenSCADA (<http://oscada.org/ua/poslugi>).
- *Адаптація системи OpenSCADA до роботи на апаратній платформі ARM.* - Здійснено збірку, адаптацію та повномасштабне тестування OpenSCADA до архітектури ARM; на інтернет планшеті фірми [Nokia](#) - [N800](#) (<http://wiki.oscada.org/Works/Tests/ARM> (RU)). Виконано збірку та адаптацію OpenSCADA для складного (дуже старого) програмного оточення контролера [LP-5451](#), а також збірка для контролеру [SMH2Gi](#) та смартфонів фірми [Nokia](#): [N900](#), [N950](#), [N9](#). У межах збірки та адаптації на різні мобільні пристрої було виконано наступні задачі:
 - *SYS:*
 - Прототипи функцій `TMess::put()`, `TMess::get()`, `TCntrNode::grpSize()`, `TCntrNode::grpId()`, `TCntrNode::grpAt()` змінено для виправлення використання беззнакового типу "char" на ARM як знакового.
 - Виконано адаптацію для збірки з Glibc версії менш 2.5 (2.3.2).
 - *TVariant, DAQ.{AMRDevs, OPC-UA}*: Виправлено невірний доступ до даних.
 - *DAQ.OPC-UA*: Додано системні функції `floatLE()`, `floatLErev()`, `doubleLE()`, `doubleLErev()` для перетворення формату збереження реального числа на різних архітектурах.
 - Додано адаптацію до uClibc. Збірка починається з версії 0.9.32 яка має підтримку функцій реального часу, на зразок `clock_nanosleep()`.
 - Додано перевірку та відключення використання "Iconv" та "Intl" (I18N) API.
 - Додано автоматичне виявлення та включення параметру конфігурації `--enable-CrossCompile` для оточення крос-компіляції.
 - Додано параметр конфігурації `--enable-CoreLibStatic` для відключення збірки поділюваної бібліотеки ядра OpenSCADA та статичної лінковки програми виклику-запуску.
 - Функцію `usleep()` скрізь замінено на власну функцію-обгортка `TSYS::sysSleep()` по причині відсутності функції `usleep()` у "uClibc" та доступності її тільки при спеціальній конфігурації збірки uClibc. В `TSYS::sysSleep()` використано `nanosleep()`.
 - Функцію `pthread_yield()` замінено на `sched_yield()`.
 - *DAQ:*
 - *JavaLikeCalc*: Віртуальну машину переписано для прямого використання структур команд, з метою виключення проблеми вирівнювання, а також для деякого підвищення продуктивності.
 - *Siemens*: Виправлено проблему бібліотеки NoDave на предмет глобального виклику команди упаковки структур "#pragma pack(1)".
 - *ICP_DAS*: Додано бібліотеку API "ICP DAS" `libi8k.a` для архітектури ARM та включена можливість збірки модуля для ARM.
 - *Archive:*
 - Прототипи функцій `TArchiveS::messPut()` и `TArchiveS::messGet()` змінено для виправлення змінної рівня повідомлення як знакового числа.
 - *FSArch:*
 - Виправлено збереження реального числа формату LE у архіві, на архітектурі ARM.
 - В алгоритмі швидкого підрахунку кількості бітів виправлено функцію невірний читання `TSYS::getUnalign32()`.
 - Розмір поля кодування у плоскому архіві повідомлення розширено з 9 до 99 символів.
 - *UI.QTcfg*: Додано короточасне засинання після закриття всіх вікон у функції виключення, з метою надати час остаточно зруйнуватися об'єктам всіх вікон та

попередити падіння.

- *Реалізація механізму відкату змін редагування у Vision.* — У межах вікна візуального редагування віджетів реалізовано багаторівневий механізм відкату змін для всіх основних операцій: візуальна зміна геометрії, зміна значення атрибуту віджета, додавання/видалення віджета, копіювання віджета та редагування віджетів на основі примітиву "ElFigure".

2 Оптимізація, підвищення стабільності, стійкості та продуктивності системи

З моменту останнього стабільного релізу, у процесі робіт над даною версією, а також її практичної адаптації, було виявлено та виправлено в загальній складності більш 300 помилок. В цілому була виконана роботи по чистці коду від попереджувальних повідомлень компілятора, з прапорцем "-Wall":

- видалення невикористаних змінних;
- додавання повернення до всіх функцій;
- ініціалізація змінних з можливістю їх використання невизначеними;
- виключення неповної обробки перелічених типів у операції обрання;
- порівняння знакового з беззнаковим цілим.

Перелічимо більш суттєві помилки, виправлення яких значно відобразилося на підвищені стабільності:

- *Загальносистемні:*
 - Команда складальної системи "\$ make dist" виправлена та адаптована для коректного створення дистрибутивів OpenSCADA.
 - Функція *TSYS::cron()*, для виклику за розкладом, неодноразово виправлена.
 - Тип змінної для збереження результату функції *find()*, рядки, скрізь змінено на коректний "size_t".
 - Виправлено позапланові виклики за розкладом, пов'язані з розсинхронізацією значень функцій *time()* та *clock_gettime(CLOCK_REALTIME,&sp_tm)*.
 - Виправлено перевірку та очікування потоків на доступність при перекритті з зупинкою попереднього, однойменного потоку.
 - Виключено очікування ініціалізації для відокремлюємих задач з метою попередження зависання на очікуванні швидко закриваних завдань.
 - Виправлено роботу функцій невіривнюваного читання *getUnalign*()* на ARM. Проблема виявлено на PXA270.
 - *Function:* Виключена можливість витоку пам'яті у випадку спроби встановити об'єкт у необ'єктний параметр функції.
 - *DAQ.ModBus, Transport.Serial, UI.{WebCfg, WebCfgD, WebUser, WebVision}:* Виправлено помилкове використання символу завершення рядка '\n' для множини програмних платформ UNIX, MAC, DOS/Windows, шляхом заміни на "\x0A".
 - *ResString:* Для читання-запису рядка використано оборот "string(vl.data(), vl.size())", з метою попередження COW алгоритму.
 - *Function:* Виправлено перевірку на модифікацію при встановленні значень різних типів IO (Реальне в Ціле).
- *Бази даних:*
 - *DB.PostgreSQL:* Виправлено обробку адреси БД на предмет визначення порожніх полів.
- *Транспорти та протоколи:*
 - *Transport:*
 - Виправлено завантаження всіх транспортів із БД, у випадку наявності записів про транспорт, модуль якого відсутній.
 - Виправлено використання беззнакового типу у поверненні функцій *read()* та *write()*. Додано ресурси до лічильникам вхідних/вихідних запитів.
 - Включена можливість переривання вихідного транспорту сигналом на очікування таймаута. Корисно для переривання транспорту при виході/зупинці.

- *Sockets*:
 - Виправлено пропуск ініціалізації розміру повернутого значення для функції *getsockopt()*.
 - Виправлено закриття клієнтського підключення по таймауту часу життя, для підключень при відсутності даних.
 - Unix-сокет встановлено у неблокуючий режим за посередництвом прапорця `O_NONBLOCK`. Виправлено формування повідомлень про помилки для вихідних транспортів.
 - *Serial*: Встановлено прапорець `O_NONBLOCK` для функції *open()* послідовного пристрою з метою попередження блокування відкриття порту на деяких USB->RS485 перетворювачах.
 - *Sockets, SSL*: Додано очищення вхідного буферу для попередження повторного використання на високолатентних каналах, після таймауту.
 - *Transport, DAQ.{AMRDevs, DCON, ICP_DAS, ModBus, OPC-UA}, Protocol.HTTP*: виправлено використання функції *TTransportOut::messIO()* під час запиту хвоста та отриманні нульової відповіді.
 - *Protocol.HTTP, UI.{WebCfg, WebCfgD, WebVision, WebUser}*: Виправлено розбір елементів POST-запитів на предмет розбору всіх атрибутів кожного елемента.
- *Збір даних та архіву*:
 - *Archive*:
 - Виправлено використання буферу м'якої ґратки з низькою та високою роздільною здатністю часу у архівах значень.
 - Виправлено обчислення початку архіву в функції *begin()*, при порожньому буфері.
 - *FSArch*:
 - Виправлено створення файлів архіву повідомлень при розташуванні безлічі старих повідомлень. Додано параметр попередження дублікатів.
 - Результат функції *fgetc()* записано в цілочисельну змінну, замість символної, що виправило виявлення "EOF".
 - Виправлено читання даних з буферу архіву у відповідності з розміром буфера в алгоритмі швидкого обчислення кількості бітів.
 - Виправлено розташування зміщення до кешу із алгоритма швидкого обчислення кількості бітів.
 - Виправлено некоректну обробку кінця невіривняних блоків швидкого алгоритма підрахунку кількості бітів.
 - Додано монопольний ресурс до виклику функції *calcVIOff()* з метою попередження некоректної роботи з кешем.
 - *DAQ*:
 - *System*: Виправлено падіння при виході для джерела "Hddtemp".
 - *ModBus*: Режим встановлення значень атрибутів параметрів змінено на активний та покращено обробку помилок.
 - *JavaLikeCalc*:
 - Додано перевірку індекса при доступі до символу рядка за індексом, з метою попередження виходу за діапазон.
 - Виправлено падіння функції *replace()*, об'єкту "RegExp".
 - *Siemens*:
 - Виправлено отримання та запис значень реальних типів.
 - Виправлено перепідключення у випадку помилки.
 - Виправлено використання властивості розміру цілого, реального та строкового типів.
 - Додано очищення зв'язків параметру при його виключенні.
 - *Інтерфейси візуалізації*:
 - *VCAEngine*:
 - Виправлено блокування сеансу для динамічно активних проектів при динамічному доступі до власних атрибутів віджету.
 - Попереджено можливість втрати змін в сервері візуалізації при відображенні

у візуалізаторах шляхом попереднього збереження значень тактового лічильника у сервісному запиті "openList".

- Ресурси атрибутів віджетів зроблено більш прозорими (за посередництвом використання рекурсивного мютексу) та використано один загальний ресурс, для сеансів та іншого, з метою попередження перехресного блокування однієї задачі іншою.
- *Vision, WebVision:*
 - Підвищено роздільну здатність координат всіх віджетів до трьох знаків після коми, з метою забезпечення нормального позиціонування на великих масштабах.
 - виправлено доступ за границю вектору під час відображення графіків примітиву "FormEls".
 - виправлено оновлення зображення примітиву "Елементарна фігура" під час зміни масштабу.
 - виправлено оновлення примітиву "Протокол" за часом та розміром.
- *Vision:*
 - виправлено падіння сеансу для проектів з періодом виконання більш 500 мс.
 - Реалізовано доведення координат фігур примітиву "Елементарна фігура" до залишку положення віджету.
 - Масштабування тексту примітиву "Текст" змінено на зміну розміру шрифту замість масштабування кінцевого зображення, що підвищило загальну якість відображення шрифту.
- *WebVision:*
 - Видалено перевірку попадання точок фігур у область відображення примітиву "Елементарна фігура". Цю функцію перекладено на бібліотеку GD.
 - Виключено можливість падіння для примітивів "Diagram" або "ElFigure", у випадку передачі деяких значень атрибутів зображень "xSc" або "ySc" із Web-браузера.
 - виправлено відкриття та заміна кореневої сторінки.
 - виправлено обробку масштабу вкладених сторінок, з розрахунком масштабу кореневої.
 - виправлено очищення хвосту примітиву "Протокол".

Оптимізація та підвищення продуктивності:

- *SYS:*
 - Механізм ресурса об'єкта "ResString" замінено з RW-блокувань на мютекс, з метою підвищення продуктивності та скоротити споживання пам'яті.
 - *TConfig:* Об'єкт елементу конфігурації (TCfg) оснований на TVariant. Доступ до об'єкту "TCfg" зроблено прозорим, що дозволяє використовувати зв'язки без базових типів.
 - Механізм зберігання дочірніх вузлів у карті за ключем "string" змінено на тип ключа "const char*". Всі вузли адаптовано до константності ідентифікатору.
- *Archive.FSArch:* Обробку індексної таблиці архіву значень значно прискорено за посередництвом використання швидкого алгоритма обчислення кількості бітів у 32-разрядному цілому.
- *UI.WebCfgD:*
 - Додано зміну розміру іконок на боці сервера з метою зниження трафіку.
 - Використано груповий запит вмісту сторінок для значного підвищення продуктивності на повільних та високолатентних каналах.
- *UI.QTCfg:*
 - Використано груповий запит до інтерфейсу керування для підвищення продуктивності на повільних та високолатентних підключеннях.
 - Виключено багаторазовий запит елементів обраних рядків таблиць.
- *UI.Vision:* Підвищено продуктивність відображення заповнень примітиву "ElFigure".
- *UI.VCAEngine:*
 - Виконано заходи по підвищенню швидкості запуску сеансу проекта в числі:

- Пропущено ініціалізація базових атрибутів у зв'язку їх наступного наслідування.
- Видалено переклад деяких повідомлень часу виконання сеансу.
- Реалізовано включення тільки потрібних сторінок при запуску сеансу. Решта сторінок виконуваного сеансу включаються по зверненню до них. Це дозволило значно підвищити швидкість запуску сеансу, а також зменшити залежність швидкості запуску від складності проекту візуалізації.
- Реалізація архівного режиму примітиву "Документ" змінено на пряму роботу з БД, що дозволило розширити глибину архіву до 1000000 документів та без шкоди для оперативної пам'яті.

3 Удосконалення та стабілізація графічної підсистеми

Значну роботу було виконано у межах графічної підсистеми, а саме у модулях рушія СВК *UI.VCAEngine*, візуалізаторів *UI.Vision* та *UI.WebVision*, а також конфігураторах. Внесені зміни були направлені на стабілізацію, оптимізацію споживання пам'яті та покращення користувацьких властивостей.

Удосконалення графічної підсистеми:

- *Додано API підсвічення синтаксису*. На основі даного API підсвічення синтаксису додано до:
 - *DAQ*:
 - *BlockCalc*: Підсвічування синтаксису у опису атрибутів параметрів.
 - *JavaLikeCalc*: Підсвічування синтаксису до компілятора мови *JavaLikeCalc.JavaScript*, а також до всіх текстових полів, які використовують цю мову всередині модуля.
 - *ModBus, OPC-UA*: Підсвічування синтаксису переліку атрибутів.
 - *Protocol.UserProtocol*: Підсвічування синтаксису до текстів процедур протоколу.
 - *UI*:
 - *QTCfg*: Відображення багаторівневого підсвічування синтаксису, яке передається з моделі даних у вигляді правил регулярних виразів.
 - *VCAEngine*: Підсвічування синтаксису до текстових атрибутів примітивів віджету, для яких це потрібно.
 - *Vision*: Підсвічування синтаксису до текстових атрибутів примітивів віджету, а також текстам програм віджетів.
 - *WebUser*: Підсвічування синтаксису процедур формування Web-інтерфейсу.
- *UI.QTStarter*:
 - Адаптація для ARM.
 - Шрифт повідомлення у "сплеші" зафіксовано у розмірі 10 пікселів, для забезпечення однакового відображення.
 - Реалізовано зупинку OpenSCADA у випадку закриття останнього вікна діалогу обрання QT UI-модулів.
 - Реалізовано проброс параметрів командного рядка QT4 із OpenSCADA (наприклад, *--qws* та *--style* у *-qws* та *-style*).
- *UI.QTCfg*:
 - Додано можливість ручної зміни висоти текстових та табличних полів. Зміна відбувається захопленням та пересуванням нижнього краю поля.
 - Адаптація для ARM.
 - Груповий запит "CntrReqs" використано для запиту контексту сторінки.
 - Додано обмеження на висоту рядків таблиць у половину висоти таблиці.
 - Введено обмеження на розмір діалогу вводу та включено адаптивне визначення його розміру за вмістом.
 - У елементів доступних для зміни висоти користувачем, для забезпечення роботи всюди, встановлено фіксований стиль "StyledPanel".
 - Додано функцію повнотекстового редагування тексту комірок таблиць.
 - Рушії відтворення правил підсвічування синтаксису повністю переписано для

- спрощення та попередження прихованих помилок.
- *UI.{VCAEngine, Vision, WebVision}*:
 - виправлено звернення за межі масиву (вектору) під час побудови трендів, примітиву "Діаграма".
 - Додано та реалізовано властивість ширини графіку, примітиву "Діаграма".
 - Додано періодичну перевірку дерева віджетів сторінки з метою виявлення факту видалення віджетів, шляхом запиту повного переліку віджетів.
 - Реалізовано відображення графіків, групи графіків примітиву "Діаграма" у шкалі значення у випадку різниці шкал всіх графіків не більш ніж на 20%.
 - Додано та реалізовано можливість обрання режиму підтвердження для елементів редагування рядка та тексту примітиву "Елементи форми".
 - виправлено можливість втрати деяких змін у моделі, під час відображення їх у візуалізаторах.
 - Додано та реалізовано властивість кількості значень на піксел у графіках примітиву "Діаграма", з метою керування деталізацією експорту у CSV та інше.
 - Реалізовано логарифмічну шкалу значень для графіків примітиву "Діаграма".
 - Додано та реалізовано властивість збереження співвідношення сторін головної сторінки під час розгортання головного вікна виконання сеансу проекту.
 - Додано новий тип "Повне Відео" до примітиву "Медіа". Для реалізації в *UI.Vision* використано медіа-рущій "Phonon".
 - Додано поле властивостей архіву "prg{X}prop" у примітиві "Діаграма", яка заповнюється значенням доступної глибини та періодичності архіву.
- *UI.VCAEngine*:
 - Додано специфічні властивості до примітивів, а саме допомога та підсвічування синтаксису.
 - Реалізовано включення сторінок сеансу проекту за потребою, з метою підвищення швидкості запуску та оптимізації використання оперативної пам'яті.
 - Примітив "Документ" перероблено для ведення архівів, архівного режиму, повністю у БД, а також додано функцію користувацького API для доступу до елементів архіву.
 - Додано періодичну (30 хвилин) перевірку та закриття вже невикористаних (втрачених) сеансів проектів.
 - Видалено попередній (суперечливий) механізм відміни видалення вкладених віджетів та замінено повноцінним механізмом відкатів у візуалізаторі *UI.Vision*.
 - Виконано оптимізацію використання пам'яті елементами сеансу до 20%.
 - Видалено підтримку першої, застарілої, версії структури БД середовища візуалізації.
 - Додано обмеження на час генерації документів у примітиві "Документ" у 5 секунд.
 - Додано підтримку типу атрибутів "Об'єкт" та "Текст".
 - Вимкнено примусову ініціалізацію нових користувацьких атрибутів у EVAL.
 - Додано контекстну підказку до атрибутів примітивів віджетів.
 - Примітив "Документ" переключено у режим повного розбору XML з метою повноцінного збереження розмітки XHTML.
 - Додано індикацію стану коректності посилань у вигляді "(+)" в кінці.
 - Додано перенаправлення необроблених повідомлень до сторінки вище. Це дозволяє глобально обробляти повідомлення на самій верхній сторінці.
 - Додано можливість встановлення зв'язку між віджетами типу "wdg:" як у вигляді абсолютної, так і відносної форми.
 - Додано підтримку типу зв'язку "arh:".
 - Додано підтримку об'єктного типу параметрів джерела даних.
- *UI.Vision*:
 - Додано відображення допомоги за атрибутами віджету.
 - Адаптація для ARM.
 - Додано індикацію повідомлення про втрату з'єднання з сервером візуалізації, а також відновлення, при віддаленій візуалізації.
 - Додано окремий повнотекстовий редактор для текстових атрибутів.

- Додано реалізацію повноцінних відмін та повторів змін під час візуального редагування віджетів.
- Покращення у чутливості та оновлені змін сеансу виконання проекту: оптимізовано продуктивність відтворення примітиву "ElFigure", обробка зміни рівня віджету "geomZ".
- Додано експорт у CSV формат із примітивів "Діаграма" та "Документ".
- Введено обмеження на розмір діалогів вводу та включено адаптивне визначення їх розмірів за вмістом.
- Повністю переписано механізм виконання правил підсвічування синтаксису.
- Додано створення нового сеансу після відновлення віддаленого підключення та відсутності попереднього сеансу.
- Фон вікна редагування віджету встановлено у шаблон QT::Dense7Pattern з метою усунення можливості перекриття за кольором.
- Додано обробку повідомлень вибору віджету під час натиску у області скролінгу, але за межами віджету.
- Примітив "Елементарна фігура":
 - Додано діалог встановлення властивостей окремо обраних фігур.
 - Виконано реорганізацію операцій із діалогом властивостей та контекстне меню з динамічними та статичними властивостями.
 - У процесі додання фігури реалізовано її відтворення під час пересування курсору миші.
- *UI.WebVision*:
 - Додано підтримку gif та jpg зображень у елементах примітиву "Елементарна фігура".
 - виправлено обробку масштабу вкладених кадрів, а також переглянуто механізм формування-включення скролу.
 - Додано функцію перетворення зображень на боці серверу. Функцію перетворення використано для зміни розміру та обезбарвлення зображень неактивних кнопок.
 - Реалізовано відкриття малих вікон вкладених сторінок у вигляді DIV-блоків, що вирішує проблему блокування та тривалого відкриття зовнішніх вікон багатьма браузерами.
 - Головну сторінку інтерфейсу відцентровано у вікні браузера.
- *UI.WebCfgD*:
 - Зменшено яскравість обезбарвлених-пасивних кнопок.
 - Всі вікна діалогів реалізовано у вигляді DIV-блоків з метою виключити проблеми та затримки при відкритті зовнішніх вікон різноманітними браузерами.
 - Використано груповий запит вмісту сторінок для значного підвищення продуктивності на повільних та високолатентних каналах.
 - Додано інформацію у строку статусу про поточного користувача та можливість його зміни.
 - Виконано адаптацію для роботи у повну ширину екрану.

4 Формування, розширення та стабілізація API користувачького програмування

Було продовжено формування об'єктного API користувачького програмування, яке передбачає інтеграцію користувачьких функцій до дерева об'єктів системи OpenSCADA. Крім того було внесено низку змін у існуючі бібліотеки функцій користувачького API.

Зокрема було здійснено наступні зміни:

- *SYS*:
 - API користувача для об'єкта TArrayObj розширено на предмет можливості використання об'єктних властивостей, якщо властивості не числа.
 - До дерева об'єктів OpenSCADA API користувача додано функцію *SYS.Security.access()* з метою контролю доступу користувача OpenSCADA до ресурсів з потрібними правами.

- Додано функції роботи з файлами: *SYS.fileRead()* та *SYS.fileWrite()*.
- До API об'єкту *XMLNodeObj* додано функцію *getElementBy()* для пошуку вкладених вузлів за значенням атрибуту.
- Додано функцію здійснення кодування тексту між різними символічними кодуваннями.
- Додано функцію *NodeObj.nodePath()*.
- *SYS, DAQ.JavaLikeCalc*: Додано реалізацію підтримки користувацького API регулярних виразів, заснованих на "PCRE", за посередництвом об'єкту "TRegExp". Реалізовано інтерфейс функцій *RegExp.exec()* та *RegExp.test()* у стандарті для мови "JavaScript". Для *DAQ.JavaLikeCalc* реалізовано "**new RegExp()**" ініціалізацію для об'єкта "TRegExp". Для *DAQ.JavaLikeCalc* реалізовані додаткові функції з використанням "RegExp": *int search(); Array match(); Array split(RegExp pat, int limit); string replace(string substr, string str); string replace(RegExp pat, string str)*.
- *TConfig, TBD, TTransportIn, TTransportOut, TUser, TGroup, TPrmTplLib, TPrmTempl, TParamContr, TController, TVArchive, TVArchivator, TMArchivator*: Додано функції користувацького API *cfg()* та *cfgSet()* для доступу до конфігурації об'єктів (які зберігаються у БД).
- *TFunction, DAQ.JavaLikeCalc*: Додано об'єкт API користувача "TFuncArgObj" для отримання аргументів функції.
- *DB*: Додано функції користувацького API *fieldStruct(), fieldSeek(), fieldGet(), fieldSet()* та *fieldDel()* для об'єкта таблиці, з метою надати користувацький доступ до БД за посередництвом інтерфейсу БД OpenSCADA.
- *DAQ*:
 - Додано функцію *TController::alarmSet()* для генерації типових порушень у об'єкті контролеру модулів підсистеми "Збір даних".
 - Додано функцію *SYS.DAQ["Modul"]["Controller"]["Parameter"]["Attribute"].arch()* для прямого звернення до об'єкта архіву, пов'язаного з атрибутом параметра.
 - Додано функції користувацького API *enable()* та *start()* для прямого контролю за станом об'єкту контролера.
 - *JavaLikeCalc*:
 - Додано вбудовану функцію *typeof()* для перевірки типу значення.
 - Для типу даних "null" додано функцію *isEval()*, яка завжди повертає "true".
 - До об'єктів бібліотечних функцій додано функцію користувацького API *call()*.
 - *ModBus*: Додано функцію користувацького API *messIO()* до об'єкта контролеру "ModBus" з метою надання можливості надсилання нестандартних ModBus-запитів безпосередньо із шаблону параметра.
- *DAQ.JavaLikeCalc, Special.{FLibComplex1, FLibMath, FLibSYS}*: Додано функцію динамічного виклику бібліотечних функцій *SYS.DAQ.JavaLikeCalc["lib_{Lib}"].funcId(prms, ...)*.
- *Archive*: Додано функції користувацького API *status(), end()* та *begin()* для доступу до стану та властивостей об'єкту архіватора повідомлень.
- *UI.VCAEngine*: Додано функцію користувацького API *wdgAt()* для простого підключення до віджету як за абсолютним, так і відносним шляхом.

5 Загальносистемні розширення

До загальносистемного API системи OpenSCADA були внесені значні зміни та розширення з метою загальної стабілізації та у зв'язку із загальними заходами по очищенню коду, а також реорганізації дерева вихідних текстів:

- *SYS*:
 - Реалізовано нову структуру дерева вихідних текстів.
 - Всі ресурси проекту переміщено до окремих архівів.
 - Для створення архівів вихідних текстів та ресурсів можна використовувати команди "\$ make dist-lzma" та "\$ make dist-reslzma".
 - Документація виконана статичною, для модулів, та поділена за мовами. Для

кожної мови документації формується окремий пакет. В документації кожної мови створено індексний файл по всьому переліку локальної та мережевої документації. Загальний перелік локальної документації помітно розширено новими документами, а також допоміжною мережевою документацією.

- Додано пакети бібліотечних БД `"*-LibDB.Main"` та `"*-LibDB.VCA"`, розташовано у `"data/LibsDB"`.
- Замість `"DemoDB"` пакета додано пакети моделей `"AGLKS"` та `"Boiler"`, розташовано у `"data/ModelsDB"`.
- Конфігураційні файли для побудови RPM та DEB пакетів оновлено.
- Залежність на бібліотеку `PCRECPP` глобально замінено на `PCRE`.
- Додано нову функцію `TSYS::time2str()` для перетворення інтервалів часу в строку типу `"1год 23хвил 10сек"`. Цю функцію всюди використано.
- Додано реалізацію власного XML-парсеру, замість `Expat`. Власний XML-парсер вирішує проблему підтримки різних кодувань, відмінних від UTF, а також безпечної обробки ситуації наявності недозволених, для поточного кодування, символів.
- `TMess`: Загальні повідомлення інтерфейсу керування переміщено до однієї функції `TMess::lab*`.
- Об'єкт зберігання рядка з ресурсом `"ResString"` значно розширено на предмет прозорого перетворення з/у тип `"std::string"`.
- Додано можливість збереження до конфігураційного файлу:
 - Додано префікс БД `"<cfg>"` для представлення конфігураційного файлу у ролі джерела завантаження/збереження конфігурації.
 - Функції `TDBS::dataSeek()` та `TDBS::dataDel()` оновлено для суворої обробки конфігураційного файлу.
 - Функцію `chkSelDB()` адаптовано для підтримки запису до конфігураційного файлу.
 - Додано збереження великих та багаторядкових значень полів конфігурації у текстовому полі окремого тегу конфігураційного файлу.
- Тип `"long long"`, за звичай 64-рядкове ціле замінено всюди на більш визначений `"int64_t"` на всіх архітектурах.
- Об'єкти потоків `OpenSCADA` жорстко пов'язано з власним потоком. Додано статистику завантаження та викликів періодичних потоків.
- Реалізовано можливість перевірки завантажених об'єктів на їх відсутність у БД та видалення. Функція перевірки активується тільки у випадку прямого завантаження із БД.
- Додано системну періодичну функцію `perSYSCall()` виклику об'єктів підсистем та їх модулів. Використовується для сервісних цілей ненавантажених та рідких задач модулів та підсистем.
- Функцію тестування виразу за шаблоном переміщено до об'єкту `"TRegExp"`.
- `XMLNode`:
 - Повністю видалено код використання XML-парсеру `"Expat"`.
 - Додано можливість завантаження текстових частин тегу у окремі-спеціальні теги, у повному режимі. Призначено для повного збереження конфігурації тегу.
 - Додано можливість завантаження та збереження блоків коментарів у окремі-спеціальні теги, у повному режимі.
- Додано вбудовану функцію включення генерації дампу пам'яті (зняття обмеження на розмір файлу) за посередництвом параметру запуску `OpenSCADA --CoreDumpAllow`.
- До всіх скриптів запуску `OpenSCADA` додано процедуру автоматичної обробки файлу дампа пам'яті (формування файлу розгортання стеку — `backtrace`). Файл дампу пам'яті очікується у робочій директорії, у вигляді `"core"`.
- Додано можливість включення (вбудовування) модулів у бібліотеку ядра `OpenSCADA`.
- Версію модульного API всіх підсистем збільшено до 6.
- `TConfig`, `TValFunc`, `TVal`: Збереження строкових конфігураційних полів переведено

до об'єкту ресурсного рядка "ResString", а потім повернено з використанням внутрішнього ресурсу.

- *TVariant*: Механізм збереження для рядка змінено до збереження покажчика на масив у "union" з адаптивним механізмом виділення пам'яті (проста, безпосередньо до 7 символів зберігається статично, а під більше пам'ять виділяється динамічно).
- Складальну систему була значно оновлено на предмет:
 - Додано перевірку та виключення використання "Iconv" та "Intl" (I18N) API.
 - Додано автоматичне виявлення та включення параметру конфігурації `--enable-CrossCompile` для оточення крос-компіляції.
 - Додано параметр конфігурації `--enable-CoreLibStatic` для виключення збірки поділяємої бібліотеки ядра OpenSCADA та статичної лінковки програми виклику-запуску.
 - Додано зведену таблицю конфігурації, в кінці.
 - Переміщення коду обробки I18N до файлу I18N.mk та включення його безпосередньо до Make-файлів, за потребою, на останній стадії конфігурації.
 - Додано загальний макрос `AX_MOD_EN()` для уніфікованого контролю модулів.
 - Додано встановлення файлів що включаються (заголовки).
 - Додано файл `openscada.pc` для перевірки конфігурації OpenSCADA під час побудови зовнішніх модулів.
 - Глобальна версія вихідних файлів OpenSCADA змінена у 0.8.0 та версія бібліотеки ядра OpenSCADA змінена у 2.0.0.
- Реалізація маніпуляції об'єктами через атрибути параметрів:
 - Реалізовано загальну поточну безпечність (`threadsafe`) для об'єктів користувацького API: "TVarObj", "TArrayObj" та "XMLNodeObj". Додано автоматичне видалення об'єкту у "AutoHD" при поверненні функцією `AHDDisConnect()` — "true".
 - Додано зворотну серіалізацію (розбір XML представлення об'єкту) для об'єктів "TVarObj", "TArrayObj" та "XMLNodeObj". Включено для "TVariant", "TVal", "TValFunc", "UI.VCAEngine.Widget", що дозволяє зберігати та завантажувати значення типу Об'єкт у БД.
 - *TVal*, *DAQ*.{*BlockCalc,JavaLikeCalc,LogicLev,ModBus,Siemens*}, *UI.VCAEngine*: Додано підтримку типу значення "Об'єкт". Включено для повсякчасного використання.
- *DOC*:
 - Виконано перевірку та виправлення Англійських текстів у ядрі OpenSCADA. Оновлено переклади ядра OpenSCADA на Німецьку, Російську та Українську мови.
 - Виконано перевірку та виправлення Англійських текстів модулів OpenSCADA. Оновлено переклади модулів OpenSCADA на Німецьку, Російську та Українську мови.
 - Оновлені всі основні документи на доступних мовах: `openscada.pdf`, `build.pdf`, `properties.pdf`, "OpenSCADA API", "Про OpenSCADA", "Бібліотека TechApp", "Quick Start", "WLib Основні", "WLib Елементи мнемосхем".
- *Бази даних бібліотек*:
 - До бібліотеки основних візуальних елементів додано:
 - Кадри реалізації редагування та виконання "Рецептів" — "Користувацьких програм".
 - Діалог обрання та конфігурації параметрів для кадру "Група Графіків".
 - Додано деякі покращення до кадру "Група Графіків" для обрання параметрів користувачем та іншого.
 - У кадр "Група Графіків" додано горизонтальний скрол-бар огляду архівів.
 - До комплекту пакету бібліотек додано [бібліотеку елементів Електричних схем \(RU\)](#).
 - Кадр головної сторінки доповнено механізмом "програвання" моделей технологічних процесів (ТП).
 - Модель [промислового котлоагрегату\(RU\)](#) перекладено на Англійську та

Українську мови.

- *DB*: Додано обмеження за часом у 5 секунд та поле встановлення початкового зсуву для запиту вмісту великих таблиць.
- *Transport*:
 - Додано функцію *TTransportS::traf2str()* для перетворення значень лічильника трафіку у рядок вигляду "12.5KiB".
 - Типи форматів вихідних запитів розширені варіантами тексту з різним завершенням рядку: LF, CR, CR/LF.
- *DAQ*:
 - Додано можливість обрання типу атрибута шаблону "Об'єкт".
 - *DiamondBoards, ModBus, LogicLev*: Додано можливість зміни типу параметра для виключених параметрів багатотипових модулів підсистеми "Збір даних".
 - *LogicLev, ModBus, Siemens*:
 - У випадку виконання за розкладом до атрибута "f_frq" записується час після останнього обчислення у негативному значенні.
 - Додано ініціюючі та зупиняючі виклики при включені/виключені параметру.
 - Додано оновлення атрибутів параметру логічного типу у випадку зміни типу, прапорця "ТількиЧитання" або назви у шаблоні.
- *TArchives*:
 - Додано можливість використання регулярних виразів при пошуку за категорією та за посередництвом "/match/"
- * *DBArch, FSArch*: Додано межу часу під час запиту повідомлень із архіву.
 - Адресу посилання на атрибут параметру спрощено до вигляду "DAQMod.Cntr.Prm.attr". Підтримка старих адрес збережено.

6 Публікація рішень OpenSCADA

В процесі здійснення робіт над цим релізом на різних конференціях та виставках були представлені рішення на основі проекту OpenSCADA:

- [FOSS Sea 2011](#) — на конференції було зачитано доповідь про рішення на основі OpenSCADA, а в холі було організовано міні-стенд з рішеннями на основі OpenSCADA вживу.
- [Міжнародний інноваційний форум 2011](#) — у складі стенду ООО НІП "ДІЯ" та Дніпродзержинського Державного Університету (ДДТУ) було представлено обладнання та матеріали з рішеннями на основі OpenSCADA.
- [10 Всеукраїнська Конференція розробників та користувачів вільного ПЗ](#) — на конференції було зачитано доповідь про рішення на основі OpenSCADA, а в холі було організовано міні-стенд з рішеннями на основі OpenSCADA вживу.
- Дні розробників QT [Qt Developer Days 2011](#) у Мюнхені та Сан-Франциско: у межах цих заходів в Qt Demo Pavilion відбулася демонстрація OpenSCADA за посередництвом представлення рішень [Динамічна модель парового котла №9 ДМК](#) на ПК та [Динамічна модель реального часу Анастасіївської ГЛКС](#) на Nokia N9.

Заключення

В процесі створення нового стабільного релізу 0.8.0, промислового призначення з тривалим строком підтримки, виконано велику роботу по стабілізації, розширенню функціональних можливостей, а також адаптації для роботи на альтернативних апаратних платформах (на архітектурі ARM). Все це в цілому дозволило розширити межі повноцінного застосування OpenSCADA на всі рівні систем автоматизації, від ПЛК, серверів SCADA-систем та до розподілених систем візуалізації.

У появі нової промислової версії системи OpenSCADA 0.8.0 активну участь прийняли:

- [Савоченко Роман](#): Основний об'єм робіт по розробці, документуванню та тестуванню.
- [Лысенко Максим](#): Супровід та стабілізація примітиву "Елементарна фігура" середовища візуалізації та керування (СВК), також переклад і оновлення більшої частини документації на Англійську мову.
- *Попкова Ирина*: Переклад і оновлення інтерфейсу системи OpenSCADA та її модулів на Німецьку мову.
- *Алмаз Каримов*: Розширення модуля підтримки протоколу DCON та активне тестування.
- [Попков Олексій](#): Збірка пакетів для Fedora-based дистрибутивів.
- Багато інших користувачів системи OpenSCADA, за посередництвом всебічного тестування.

Наступні зусилля розробки будуть направлені на реалізацію програмної багатоплатформності, а також розширення функціональності шляхом створення модулів підтримки нових інтерфейсів, в том числі і за схемою авансованої розробки, з розподілом витрат реалізації на продані екземпляри (для модулів які взаємодіють з комерційними системами та інтерфейсами). Для реалізації централізованого розповсюдження рішень, послуг та розширень на основі OpenSCADA планується створення магазину, у межах головного інформаційного ресурсу OpenSCADA (<http://oscada.org>).

Посилання

Протокол тестування промислового релізу: <http://wiki.oscada.org/Works/Tests/release080>