

OpenSCADA — цільова стабілізація

Савоченко Роман Олексійович

OpenSCADA Team

<http://oscada.org>

Анотація

Розробка проекту OpenSCADA перейшла у фазу глибокої стабілізації з елементами розширення існуючого функціоналу націлену на надання стабільного та надійного оточення автоматизації промислових та суміжних завдань. У доповіді освітлено питання поточної політики розробки рішень тривалої підтримки OpenSCADA у поєднанні із системним оточенням платформи, а також загальні досягнення проекту за минулий час.

Вступ

У поточному році проект OpenSCADA тихо відзначив десятиріччя заснування переходом у фазу глибокої стабілізації. Це рішення значним чином пов'язане із активізацією робіт з адаптації та, як наслідок, всебічним тестуванням з виявленням та усуненням специфічних проблем.

Часто під стабілізацією багато з користувачів та розробників вільного ПЗ мають на увазі фактичну зупинку розробки, та повну стабілізацію — смерть. Ця думка значним чином пов'язана із сутністю ВПЗ — розробка заради задоволення «Just For Fun». Оскільки процес стабілізації, або підтримки, протягом тривалого часу прив'язує власне створіння до розробника, а у випадку із непоступальним розвитком і розпилує сили, задоволення та часу залишається мало. Такий стан речей, у багатьох випадках, породив стан безперервного процесу розробки, гонитви за новими сумнівними функціями, із фактичною смертю старих версій на етапі досягнення максимального рівня їх стабільності та початком робіт над новою, нестабільною, версією, а інколи і розробкою нового рішення що прямо перекреслює старе. Це явище дуже часто характеризується наявністю безлічі звітів про проблеми, що вказує на нагальність, але повну відсутність реакції розробників.

На етапі збірки ВПЗ у дистрибутив, дистриб'ютори фактично вимушені займатися поверхневою стабілізацією останніх, нестабільних версій, а інколи і підтримкою попередніх стабільних версій, що часто можна порівняти із доопрацюванням незакінченого виробу напилком. Як результат рішення-дистрибутиви виходять недостатнього рівня стабільності, а користувацькою аудиторією їх залишається клас просунутих користувачів, спроможних налаштувати, обійти, виправити або змиритись з проблемами. Якщо на робочій стільниці користувача такі рішення все-ж припустимі, то на комп'ютерах відповідального призначення, наприклад, у пультах операторів та диспетчерів систем промислового контролю це неприпустиме.

Дуже яскраво проблема відсутності стабільності та тривалої підтримки видно на оточеннях робочої стільниці (DE), де багато виробників рішень для кінцевих користувачів дуже рідко беруть за основу чинні DE, часто беруть за основу чинні графічні ToolKit (GTK: Maemo, Qt: MeGoo) з графічним стеком, а інколи навіть створюють цілком нове оточення з власною підтримкою (Android). І це коли доступні вихідні тексти рішень та у межах них вже зроблену величезну роботу!

Хорошим правилом та ефективною практикою є природній шлях стабілізації ПЗ, поєднаний з поступальним розвитком. А саме участь цільової аудиторії у стабілізації, шляхом всебічного-природного використання, зворотнім зв'язком та оперативною реакцією розробників. Однак мотивації для останнього у ВПЗ відсутня та необхідно поєднання ВПЗ із зовнішнім фінансуванням, зазвичай через пожертви та/або з наданням комерційного сервісу. По шляху комерційного сервісу, а також організації збору пожертв та цільового фінансування розробки пішов і проект OpenSCADA.

Стабілізація та розвиток OpenSCADA

Командою розробників приділяється значна увага стабілізації OpenSCADA, як через оперативність реакції на повідомлення, так і у межах цільового сервісу. В цілому хотілося б з гордістю відзначити відсутність, на цей час, відкритих повідомлень та оперативність усунення помилок безпосередньо у коді проекту.

У межах стабільного релізу тривалої підтримки 0.8.0 LTS було вже виправлено біля 150 помилок та випущено 8 оновлень з циклом два місяці. Крім безпосередньо виправлень у стабільну гілку внесено безліч змін по спрощенню та розширенню, які не призводять до порушення зворотної сумісності API та ABI модулів.

Значна увага у стабілізації та укріпленні було приділено через аналіз продуктивності, використання оперативної пам'яті, ревізії та спрощення надлишкового коду та загальній ефективності роботи. Для фіксації результатів цих заходів у базі знань проекту створено спеціальний розділ розширеного тестування, аналізу та оптимізації.

Відзначимо більш важливі виправлення та зміни у LTS гілці:

- *Документація*: Завершено документ "Створення модулів OpenSCADA". Значно оновлено та доповнено документ "Швидкий старт OpenSCADA" та створено відео-доповнення до нього. Створено посібник оператора-користувача, на основі моделі «АГЛКС».
- *БД*: Значні розширення, виправлення та оптимізація генерації документів за часом, а також реалізація нової концепції рушія рецептів.
- *Системні*:
 - Реальне нестворення задачі із високопріоритетного потоку на деяких Linux ядрах, для прикладу 2.6.29-rt, виправлено заміною очікування виклику sched_yield() на TSYS::sysSleep(1e-3).
 - Заміна небагатопотокових функцій: Функцію обчислення хешу паролю crypt() замінено на crypt_r(). Функцію "getopt" повністю замінено на власну функцію TSYS::getCmdOpt(). Приховано-проблемну функцію readdir() замінено на багатопотоковий варіант readdir_r().
 - Системні атрибути MessLev, LogTarget, Lang, Lang2CodeBase, WorkDB, Workdir, IcoDir, ModDir дозволено для читання та запису тільки у конфігураційний файл.
 - До функції створення задач додано контроль та звільнення завершених, але з наявним об'єктом, невід'єднаних задач.
 - Адаптація до втраченої підтримки односимвольних кодувань, перевірено на KOI8-R.
 - Додано контроль на можливість редагування для конфігураційних полів різних об'єктів, залежно від стану "Включено" та "Запущено".
 - Користувацькі об'єкти: Виправлена безкінечна рекурсія під час встановлення значення рядка у тип Об'єкт. Збереження посилання у XmlNodeObj на предка змінено на прямий покажчик для попередження самоблокування та витоку пам'яті. Додано об'єкт TEValObj для подібної до скалярів обробки EVAL.
 - Додано функцію користувацького API SYS.mktime() для отримання часу з епохи 1.1.1970 з конвертацією із поділеного часу.
 - Функцію користувацького API XmlNodeObj::childIns() виправлено для виявлення аргументу 1 типу XmlNodeObj.
 - DAQ.Templates,Protocol.ModBus: Додано використання попереднього рядка, як шаблону, при створення нового. Додано нумерацію строкових міток за посередництвом strLabEnum().
 - SYS.{WorkDir,IcoDir,ModDir},Archive.FSArch,DB.SQLite,Protocol.HTTP,Transport.Serial: Функцію ctrlListFS() додано та використано для навігації за теками та файлам у конфігураційних полях.
- Підсистема «БД»:
 - DB.SQLite: Додано закриття відкритої транзакції під час зупинки модуля, з метою попередження втрати даних при падінні деяких модулів на відключені.
- Підсистема «Транспорти»:
 - Transport.{Serial,Sockets}: Додано обробку номера помилки EAGAIN у функції write() для вихідних транспортів з метою попередження помилок на передачі великих блоків даних.
 - Transport.Serial: Додано конфігураційне поле "Пріоритет" для задачі вхідного транспорту.
- Підсистема «Збір даних»: Виправлено витікання пам'яті у об'єктах деструктору

TVal::~~TVal(). Обмеження вихідного тексту процедур розширено до 1000000 символів. Для модулів за шаблоном попереджено перевантаження виконавчого контексту із БД на запущених об'єктах контролерів.

- DAQ.System: значно оновлено та поліпшено для створення нових об'єктів джерел при їх появі.
- DAQ.SoundCard: Переписано збір для забезпечення реального часу, а також контролю частоти вибірки та втрати кадрів. Додано можливість встановлення пріоритету задачі збору, із PortAudio. Додано деякі повідомлення про помилки. Дещо збільшено продуктивність.
- DAQ.DCON: Значні зміни від Алмаза Каримова: додано читання аналогових виходів одним методом; виправлено роботу методу 1AO (#AA) (блоки NL-1AO, I-7021, ADAM-4021); додано метод 4AO (#AACN) для роботи з блоками ADAM-4024; метод читання дискретних входів @AA замінено на \$AA6 для сумісності із блоками ADAM; методи запису дискретних виходів 4DO (#AA000F), 8DO (#AA00FF), 12DO (#AA000FFF) додано для роботи їх блоками ADAM.
- DAQ.JavaLikeCalc: У мову додано поділ призначень символом ',' для відокремлення операцій всередині оператору «for». Видалення помилкового правила граматики "expr OR CONST{". Умовна логічна команда '||' та '&&' виправлена для постійних.
- DAQ.ModBus: Виправлено помилку встановлення EVAL для атрибуту за ідентифікатором посилання, у випадку помилки запису за цим посиланням. Додано підтримку комплексного типу "Рядок", блок реєстрів, у DAQ за посередництвом суфіксу "_s" до реєстру "R[I]". Додано підтримку комплексних типів "Ціле" (32 біт, знакове), "Реальне" та "Рядок" (блок реєстрів) до об'єкту вузла вхідного протоколу.
- DAQ.{BlockCalc,LogicLev,ModBus},Archive.TArchVal,UI.VCAEngine: Навігацію за атрибутами параметрів та самими параметрами переміщено до загальної функції TDAQ::ctrListPrmAttr() та використано скрізь.
- DAQ.OPC-UA,{Lib,Protocol}.OPC-UA: Частина протоколу OPC UA для клієнту та серверу переміщено у LGPL v3 бібліотеку з метою спрощення використання іншими проектами. Об'єкт контролеру спрощено на предмет обрання Кінцевих Вузлів (Endpoints) та транспортів; Ви можете просто встановити Кінцевий Вузол, а транспорт буде створено із нього. Для включеного об'єкта контролеру Ви отримаєте перелік наявних Кінцевих Вузлів. TokenId та SessionId адаптовано до комплексного типу NodeId та перевірено на Ignition OPC UA сервері. OPC UA сервер виправлено на предмет обробки об'єднаних вхідних повідомлень. Обробку основних атрибутів вузлів значним чином поліпшено та розширено.
- TVal: Додано зупинку архіву під час відключення об'єкту атрибута, для розблокування та завершення дії.
- Підсистема «Архіви»: Виправлено включення архіву із DAQ параметрів та падіння архіву значень, значно поліпшено обробку архівів. Додано запит за потрібною періодичністю для забезпечення швидкого доступу до архівів значень високої роздільної здатності, за посередництвом визначеного розміру кадру та періодичності. Попереджено у Archive.FSArch руйнування архівних даних через перекриття пам'яті буферу індексів, виправлено розташування великих повідомлень у архів плоского тексту, при досягненні ліміту на розмір буферу.
- Підсистема «Користувацькі інтерфейси»:
 - UI.{VCAEngine,Vision,WebVision}: Розширено верхню межу кількості параметрів примітиві "Діаграма" до 20, реалізовано поділ шкал окремих параметрів, додано формування трендів із прямо вказаних даних через префікси "data:" та "line:". У примітивах "FormEl", "ElFigure", "Media", "Box" використано визначення індексів із "types.h".
 - UI.{Vision,WebVision}: Виправлено завал буфера даних примітиву "Діаграма" поточними значеннями та втрата старих, видимих значень при досягненні обмеження буфера. Виправлено завантаження конфігурації елементів примітиву "ElFigure" з урахуванням

вимоги присутності всіх точок. Функція `attr{s}Set()` примітиву "ElFigure", та "FormEl" у `UI.Vision`, повністю переглянуто та оптимізовано.

- `UI.{Vision,VCAEngine}`: Додано варіант огляду елементів "Дерево" до примітиву "FormEl". Додано одно-рядкова навігація ресурсів та файлів для `image/media` атрибутів. Таблиця "Mime" контейнерів дозволяє пряме завантаження, а не тільки для обраних елементів, при цьому "mime" береться із розширення; "mime" також використано при вивантаженні файлів. Встановлено ліміт на розмір файлів для завантаження.
- `UI.VCAEngine`: Додано завантаження/включення віджетів бібліотек та проектів за вимогою. Ліміт на архів примітиву "Документ" виправлено на предмет розширення до 1000000. Експортні символи переміщено до файлу `types.h` та призначено для включення візуалізаторами. Додано зміну частини адреси батька вкладених віджетів при зміні батька бібліотечного віджета або сторінки, при включенні.
- `UI.WebVision`: Обчислення часу періоду оновлення переписано для забезпечення більшої чутливості та адаптації до реальних властивостей мережевого каналу. Додано спробу завантаження зовнішнього JS файлу "WebVisionVCA.js" із поточної теки. Зміни у JS код: значне очищення та виправлення; додано перевірка на нульову прозорість для відключення фону; спроба використання асинхронних запитів у основному циклі; примітиви "ElFigure", "Media", "FormEl" значним чином адаптовано для попередження повної заміни контексту на динамічних змінах одиночних властивостей; відключено схоплення для зображень у Firefox. ID об'єкту сеансу перейменовано у `PATH` та `nodeName()` встановлено у рядок із роздільником за шляхом. Додано опцію рівня стиснення PNG для деякого контролю продуктивності формування зображень, та значення по замовченню встановлено у "1".
- `UI.Vision`: Додано перевірку на реальну видимість частини віджета (за непрозорістю пікселю). `UI.Vision`: Додано функцію `WdgView::attrsSet()` та використано для встановлення групи атрибутів у одному запиті для попередження втрати деяких пов'язаних повідомлень при обробці, а також підвищено продуктивність у мережі. Зміни та виправлення примітиву "ElFigure": додано та використано визначення діапазону значень; додано перелічення `Shapeltem::Type` для фігур "Лінія", "Дуга" та "Безьє"; додано копіювання індексів "Ширин", "Стилів" та "Кольорів" при копіюванні фігур; виправлено поділ точок 5 декількох дуг при переміщенні клавіатурою та мишею; додано переміщення курсору миші з переміщенням клавіатурою; додано розрив контуру кола переміщенням з клавіатури; виправлено падіння примітиву "ElFigure" при натиску на загальні точки 1,2 дуги та точку 1 лінії; значне очищення коду. Виправлено переформування вектору `index_agray` для точок 1 та 2 для розташування тільки загальних фігур одноразово (включаючи Дугу, у випадку поєднання точок 1 та 2). Виправлено падіння на Лініях та Дугах (попереднє помилкове виправлення). Примітив "ElFigure" виправлено на предмет масштабування та розмасштабування товщин шляхом перенесення масштабування прямо на час зображення фігур.
- `UI.WebCfг`: Виправлено відображення поля "Дата+Час" у таблиці та інших місцях.
- Підсистема «Спеціальні»: `Spec.ModSched`: Поля "ModAllow" та "ModDeny" відкрито для редагування та збереження у конфігураційному файлі.

Хоча з моменту випуску релізу тривалої підтримки 0.8.0 LTS пройшло більш року та з того моменту ще не було жодного релізу робочої гілки, робота над версією 0.8.1 активно ведеться та до неї включено низку знакових змін, які з причини порушення сумісності не можуть бути перенесені до LTS версії. В цілому треба відзначити, що робоча гілка проекту також піддалася духу стабілізації та розробка здійснюється у руслі покращення та розширення старих модулів та функцій, створення нових модулів, та збірки робочої гілки використовуються у створенні нових відповідальних рішень.

Відзначимо найбільш знакові покращення у робочій гілці, які несумісні із LTS версією та не включені до неї:

- *DB*: Значне розширення та доповнення бібліотек процедур та візуальних елементів OpenSCADA.

- *SYS*: Переписано скрипт запуску проектів робочих станцій OpenSCADA — «openscada_start» для зберігання всіх даних проекту у окремій теці; можливість запуску проекту вказаного у змінній оточення «OSCADA_ProjName»; можливість запуску проекту вказаного у назві посилання «openscada_{ProjName}»; можливість вибору та створення нового проекту у діалозі від «dialog», «kdialog», «zenity» та «Xdialog».
- *SYS*: Складальну систему розширено для встановлення всіх потрібних даних програми (прямо через «make install»). Шлях встановлення даних змінено на «/usr/share/openscada». Із конфігураційного файлу прибрано всі шляхи, а вказані при конфігурації та збірці встановлені як по замовченню прямо у бібліотеці ядра OpenSCADA. Залишені конфігураційні зміни встановлюються у файли конфігурації через підміну «*.in». Уніфіковано цілі створення архівів дистрибутиву. Спрощено та уніфіковано механізм включення модулів у ядро програми.
- *SYS*: Змінено механізм формування повідомлень налагодження. Повідомлення налагодження включаються встановленням рівня повідомлень у системі — «Налаштування (0)» та вибір у вкладці головної сторінки «Налаштування» генерації повідомлень потрібних підсистем. Із загального шляху локального об'єкту OpenSCADA видалено ідентифікатор станції.
- *SYS*: Реалізовано новий механізм закінчення виконання системних задач, заснований на сигналі SIGUSR1 та перевірці функцією `TSYS::taskEndRun()`.
- *DAQ*: Додано функцію «Діагностика», що дозволяє у відповідній вкладці джерела даних обрати рівень діагностики, а рівнем «Налаштування (0)» включати режим налаштування запитів та реакції на них.
- *DAQ.JavaLikeCalc*: Підвищення продуктивності внутрішніх обчислень за рахунок збереження контексту обчислення та переводу алгоритму перевірки таймаута безпеки на порівняння із зовнішньою змінною часу.
- *DAQ.ICP_DAS*: Фактично повністю переписано на предмет підтримки більшості модулів серії I8k та I87k, а також плат на шині ISA.
- *DAQ.Comedy*: Новий модуль підтримки різноманітних DAQ-плат за посередництвом бібліотеки <http://comedy.org>, на шинах ISA, PCI, USB.
- *DAQ.DiamondBoards*: Повністю переписано для підтримки більшості плат DAQ фірми «Diamond Systems», а також реалізовано всі три механізми збору даних: синхронний, асинхронний та пакетний.
- *DAQ.OneWire*: Новий модуль первинної підтримки сенсорів на шині 1-Wire.
- *DAQ.SMH2Gi*: Новий, спеціалізований модуль реалізації прямого опитування УСО контролеру SMH2Gi від Segnetics.
- *DAQ.DAQGate*: Додано можливість шлюзування повідомлень, які генеруються джерелами даних що обслуговуються.
- *UI*: Функції отримання іконок та ресурсів графічних інтерфейсів `icoPresent()`, `icoGet()` та `icoPath()` об'єднано у одну функцію `icoGet()` та адаптовано до обробки декількох тек. Теки іконок по замовченню встановлено у "icons/;@oscd_datadir_full@/icons/".
- *UI.{VCAEngine,Vision}*: Кількість можливих параметрів примітиву «Діаграма» збільшено до 100. Вигляд «Кнопка» примітиву «Елементи Форми» розширено режимами: «Стандартний», «Вибір», «Меню», «Завантаження», «Збереження».
- *UI.VCAEngine*: Генерацію документу у примітиві «Документ» переміщено до окремого завдання з метою попередити блокування основного на момент обробки. При обробці документу результат оновлюється кожні 5 секунд. Атрибут "process" додано для контролю закінчення.
- *UI.Vision*: Можливість використання WebKit для реєндерингу примітиву «Документ».

Загальна стабілізація оточення виконання — дистрибутив Linux із OpenSCADA

OpenSCADA — середовище роботи із динамічними даними у реальному часі, сама працює у оточенні ОС Linux, а також графічного стеку, робочого столу та із використанням різноманітних вільних бібліотек. Відтак загальна стабільність кінцевого комплексного рішення залежить від злагодженої роботи та стабільності програмного комплексу в цілому. З цієї причини командою розробників OpenSCADA приділяється значна увага стабілізації оточення виконання OpenSCADA шляхом створення стабільного репозиторію пакетів, на основі гілки дистрибутиву ALTLinux T6, а також безпосередньої участі у стабілізації окремих частин цього оточення.

У вступі казалося про гостроту проблеми стабілізації та тривалої підтримки вільного оточення ОС Linux, особливо для відповідальних задач. Однак спостерігаються позитивні тенденції щодо переходу на схему випуску LTS-версій як окремих програм, так і дистрибутивів в цілому:

- «RedHat Linux» (Centos, Scientific Linux) — випускає стабільні LTS дистрибутиви, спрямовані на серверне використання.
- «Ядро Linux» — чітко окреслене виокремлення LTS релізів та тривала їх підтримка.
- «Ubuntu» — у низці продуктів явно фігурують, випускаються та підтримуються стабільні дистрибутиви.
- «Trinity DE» — підтримка та розвиток кодової бази KDE3, містить явно окреслену LTS гілку.
- «KDE4» — заплановано перехід на LTS версії 4.11, хоча вже на горизонті проглядається 4.12.
- Web-браузери та низка іншого прикладного ПЗ.

Однак у оточенні ОС Linux все ще залишилася низка компонентів, політика розробки яких не передбачає LTS, що часто призводить до проблемам при погодженні із обладнанням. Особливо гостро ця проблема стоїть із XOrg.

Команда OpenSCADA безпосередньо приймає участь у наступних проектах вільного оточення SCADA-системи:

- «Linux kernel» — для низки задач програмованих логічних контролерів (ПЛК) здійснюється тонка адаптація та збірка ядра реального часу, на основі ядра 2.6.29 та PREEMPT_RT патчів.
- «TDE» — для надання налагодженого, невимогливого та розвиненого оточення робочого столу TDE у репозиторіях ALTLinux було реанімовано, виправлено критичні помилки та здійснюється подальша підтримка та збірка LTS випусків.

Заключення

Розробка проекту OpenSCADA наразі, значним сином, спрямована на цільову стабілізацію та розширення, в наслідок чого довгострокові плани значним чином переглянуто. Дату випуску наступного стабільного релізу тривалої підтримки 0.9.0 LTS перенесено на Травень 2014 року, а час життя поточного стабільного релізу тривалої підтримки 0.8.0 LTS продовжено до цього строку.

Не дивлячись на пріоритет у стабілізації, розробка також продовжується. Про що свідчать значні розширення та спрощення у робочій гілці проекту, які не можуть бути перенесені до стабільної гілки, але помітно покращують програму в цілому. Розробка стала більш спрямованою, та із затребуваних задач було сформовано пул для бажаючих прийняти участь у проекті та в цілому для формування перспективи розробки. Підведення ризику під розробкою заплановано випуском першого релізу робочої гілки 0.8.1 до кінця цього року.

Після випуску релізу робочої гілки 0.8.1 заплановано значне оновлення представницьких матеріалів (логотип, дизайн, ...), відкриття програм цільового та загального фінансування для залучення до проекту більше розробників, яких на цей час небагато, однак зростаюча зацікавленість у проекті дозволяє казати про перспективність та значний потенціал цих заходів із прискорення стабілізації, розвитку та адаптації на різні платформи. Також, після випуску 0.8.1, планується приступити до запланованої адаптації та збірки OpenSCADA для альтернативних програмних платформ.

Команда розробників OpenSCADA вдячна Вам за увагу до проекту та запрошує до участі та співпраці у вирішенні різноманітних задач повсюдної автоматизації!