



## **Нові можливості CODESYS V3.5 SP17**

Посібник користувача

## Зміст

1. Модуляризація середовища програмування та значущі зміни .....	3
1.1. Вступ .....	3
1.2. Дистрибутив .....	3
1.3. CODESYS Installer .....	3
1.4. Сумісність з попередніми версіями .....	6
1.5. Рекомендації для користувачів .....	8
1.6. Менеджер пакетів і пов'язані з ним можливі проблеми.....	10
1.7. Імпорт пакетів минулих версій під час першого запуску та пов'язані з цим можливі проблеми.....	11
1.8. Помилка при відкритті документації бібліотек та інструкція з її усунення .....	12
1.9. Типові запитання під час встановлення V3.5 SP17 .....	13
1.10. Автоматична активація керування користувачами для віртуального контролера .....	14
1.11. Помилка під час підключення до CODESYS OPC Server V3.....	16
1.12. Зміна системи нумерації компонентів/плагінів на 4.x.x.x.....	16
2. Редактор мови CFC .....	18
2.1. Автоматичний контроль порядку виконання блоків у CFC.....	18
2.2. Оптимізація використання місця на полотні.....	20
2.3. Підтримка темної теми .....	21
2.4. Зміна порядку входів/виходів блоку за допомогою "перетягування" мишею .....	21
2.5. Автоматичне додавання входів у блоках із розширюваним числом входів за допомогою перетягування зв'язку .....	22
2.6. Підтримка гарячих клавіш у редакторі CFC.....	22
2.7. Налаштування кількості відображуваних символів змінних в онлайн-режимі .....	22
3. OPC UA .....	24
3.1. Підтримка інформаційних моделей OPC UA і виклику методів ФБ з боку клієнта .....	24
3.2. Підтримка профілю OPC UA Alarms and Conditions .....	25
3.3. Підтримка нових профілів безпеки зі специфікації OPC UA 1.04 .....	25
4. Середовище розробки.....	26
4.1. Імпорт проєктів із CoDeSys V2.3.....	26
4.2. Підвищення швидкодії середовища .....	26
4.3. Підтримка типів даних LDT/LDATE/LTOD .....	26
4.4. Зменшення числа команд у меню Компіляція.....	27
4.5. Опціональні аргументи для функцій і методів.....	27
4.6. Автоматичне створення стандартних методів для функціональних блоків. ....	27
4.7. Оператори __POUNAME і __POSITION.....	28
4.8. Автооголошення змінних.....	28
4.9. Вибір активної програми .....	29
4.10. Відображення "реального" значення фіксованих змінних .....	29
4.11. Перехід до фрагментів коду з панелі точок зупинки .....	30
4.12. Відображення заданого часу циклу завдання в конфігурації завдань .....	30
4.13. Покращення в інструментах аналізу пам'яті.....	30
4.14. Відображення дерева залежностей у менеджері бібліотек .....	31
4.15. Покращення в компоненті Трасування.....	31
4.16. Сортування файлів на вкладці Device - Файли .....	32
4.17. Завантаження відсутніх бібліотек через панель повідомлень .....	32
5. Бібліотеки .....	33
5.1. Бібліотека Standard64.....	33
5.2. Бібліотека CAA File .....	33
5.3. 2.Бібліотека Util .....	33
5.4. Бібліотека Net Base Services .....	34
5.5. Бібліотека CmpCrypto .....	34
5.6. Бібліотека CAA File .....	35
6. Modbus .....	36

6.1. Запис значень у holding-реєстри з програми ПЛК для Modbus Slave Device (Serial/TCP) .....	36
6.2. Незалежні області пам'яті для біт у Modbus Slave Device (Serial/TCP) .....	36
6.3. Додаткові методи та змінні для Modbus Slave Device (Serial/TCP) .....	37
6.4. Можливість прив'язки до конкретного адаптера для Modbus TCP Slave Device.....	38
6.5. Функціонал шлюзу RTU/TCP (Serial Gateway) для Modbus TCP Slave Device .....	38
6.6. Збільшення допустимої кількості одночасно підключених клієнтів для Modbus TCP Slave Device ...	38
6.7. Можливість додавання слейвів зі співпадаючими Slave ID у Modbus Slave COM Port .....	38
6.8. Оптимізації в драйвері Modbus Master (Serial/TCP).....	39
6.9. Поліпшення в механізмі редагування каналів для Modbus Master (Serial/TCP).....	39
6.10. Надсилання повторного запиту в разі відсутності відповіді у Modbus Serial Master .....	40
6.11. Змінено опрацювання каналів із налаштуванням "Встановити в нуль" у разі розриву TCP- з'єднання для Modbus TCP Master .....	40
6.12. Поліпшення можливостей діагностики .....	40
6.13. Можливість групового оновлення версій компонентів .....	42
7. Візуалізація .....	43
7.1. Підтримка технології overlay .....	43
7.2. Порядок перемикання елементів за допомогою клавіші Tab.....	44
7.3. Покращення в роботі з трендами .....	45
7.4. Вибір списку текстів у випадаючому списку за допомогою рядкової змінної .....	48
7.5. Сортування повідомлень у таблиці та банері тривоги .....	49
7.6. Покращення в роботі з тривогами.....	49
7.7. Підтримка змінних типу LDATE/LTOD/L DT .....	51
7.8. ФБ для закриття діалогів з коду програми.....	51
7.9. Фонові зображення .....	51
7.10. Покращення у відображенні системного часу .....	52
7.11. Покращення в роботі з рецептами .....	53
7.12. Спрощене введення змінних типу TIME/LTIME .....	53
7.13. Покращення в елементі XY-графік.....	53
7.14. Переміщення діалогів введення за допомогою миші .....	55
7.15. Оновлення команди Close Dialog .....	55
7.16. Відключення невипростаних в проєкті екрани візуалізації .....	56
7.17. Відключення в проєкті POU для неявних перевірок у бібліотеках візуалізації .....	56
7.18. Керування видимістю кнопок вкладок в елементі Tabs (Набір вкладок).....	57
7.19. Оновлення події OnValueChanged .....	57
7.20. Обробка натискань на елементи візуалізації в коді програми .....	58
7.21. Додаткові опції для обробки мультикасання .....	58
7.22. Зміна зовнішнього вигляду діалогу передавання файлів.....	59

# 1. Модуляризація середовища програмування та значущі зміни

## 1.1. Вступ

У **CODESYS V3.5 SP17** відбулася велика архітектурна реструктуризація. Раніше більша частина функціональних можливостей середовища програмування була об'єднана в єдиний додаток. Тільки додаткові платні плагіни **CODESYS SoftMotion** і **CODESYS Professional Developer Edition** поширювалися окремо. Під час реструктуризації ступінь модульності середовища було розширено до рівня основних функцій. У результаті редактори мов програмування, конфігуратори протоколів і генератори коду тепер виділені в окремі плагіни. Те ж саме стосується складного і масштабного функціоналу - наприклад, візуалізації та символічної конфігурації. У ядрі залишилися тільки важливі компоненти інфраструктури середовища програмування - наприклад, оболонка призначеного для користувача інтерфейсу (панель меню, елементи навігації, елементи відображення повідомлень тощо), фронтенд компілятора та компоненти для обробки проєктів і встановлення з'єднання з контролерами. Варто зазначити, що в майбутньому ще більше компонентів можуть бути винесені з ядра в окремі плагіни.

Для користувачів CODESYS модуляризація несе в собі низку переваг:

реліз окремих поліпшень і виправлень відбувається одразу після їхнього завершення (не треба чекати на реліз наступного патча або сервіс-пака);

зацікавлені користувачі зможуть отримувати бета-версії оновлень для можливості надання своєчасного зворотного зв'язку. При цьому бета-версія плагіна запускатиметься в стабільній версії середовища; непотрібні користувачеві плагіни можуть бути видалені, щоб заощадити місце на диску ПК і трохи знизити час запуску середовища.

Безумовно, ця гнучкість пов'язана з низкою технологічних складнощів. У наступних пунктах описуються обмеження, пов'язані з модуляризацією; дії, що дають змогу керувати складністю використання середовища; рекомендації для типових сценаріїв використання.

## 1.2. Дистрибутив

Дистрибутив CODESYS, як і раніше, містить у собі всі необхідні компоненти (зокрема, всі необхідні користувачеві компоненти з SP16 і більш ранніх версій). Це означає, що в результаті встановлення користувачі отримають звичне їм середовище без втрати будь-якого функціоналу<sup>1</sup>.

Розробники CODESYS вважають, що спочатку користувачі працюватимуть у "стандартному" середовищі і тільки потім за бажанням створюватимуть своє індивідуальне **оточення**<sup>2</sup>, видаляючи непотрібні плагіни і встановлюючи додаткові. Середовище не примушуватиме користувачів розбиратися зі встановленням або видаленням плагінів під час першого запуску. Крім того, цілком зрозуміло, що багатьом користувачам взагалі не цікаво цим займатися і їх влаштовує стандартне встановлення з повним набором плагінів.

## 1.3. CODESYS Installer

Під час встановлення CODESYS V3.5 SP17 і вище буде автоматично встановлено утиліту **CODESYS Installer**. Її можна використовувати для керування всіма оточеннями CODESYS, встановленими на ПК, а також їхніми плагінами. Тобто, якщо у Вас вже були встановлені на ПК версії CODESYS молодші за SP17

---

<sup>1</sup> У разі повного ("complete") варіанта встановлення

<sup>2</sup> Оточення - це сукупність конкретної версії CODESYS і встановленого в ній набору плагінів з урахуванням версій цих плагінів

## Модуляризація середовища програмування та значущі зміни

(SP5, SP11, SP14 тощо) вони автоматично стануть доступними для налаштування з **CODESYS Installer**. Для кожного оточення можна задати свою назву. CODESYS Installer - ключовий інструмент для користувачів, які хочуть отримати максимум переваг від модуляризації середовища.

CODESYS Installer дає змогу керувати будь-якою кількістю оточень (зі встановлених на ПК). Для кожного з них можна визначити свій набір плагінів. CODESYS Installer дає змогу:

- створити нове оточення для потрібної версії середовища (кнопка **Add Installation**);
- встановити для обраного оточення новий патч (за допомогою відповідної піктограми, наприклад: );  
**↑ Patch 7**
- встановити для обраного оточення оновлення плагінів (кнопка **Change**);
- видалити вибране оточення (кнопка **Uninstall**);
- запустити CODESYS в обраному оточенні (кнопка **Start**);
- провести повторну інсталяцію обраного оточення (кнопка **Repair**).

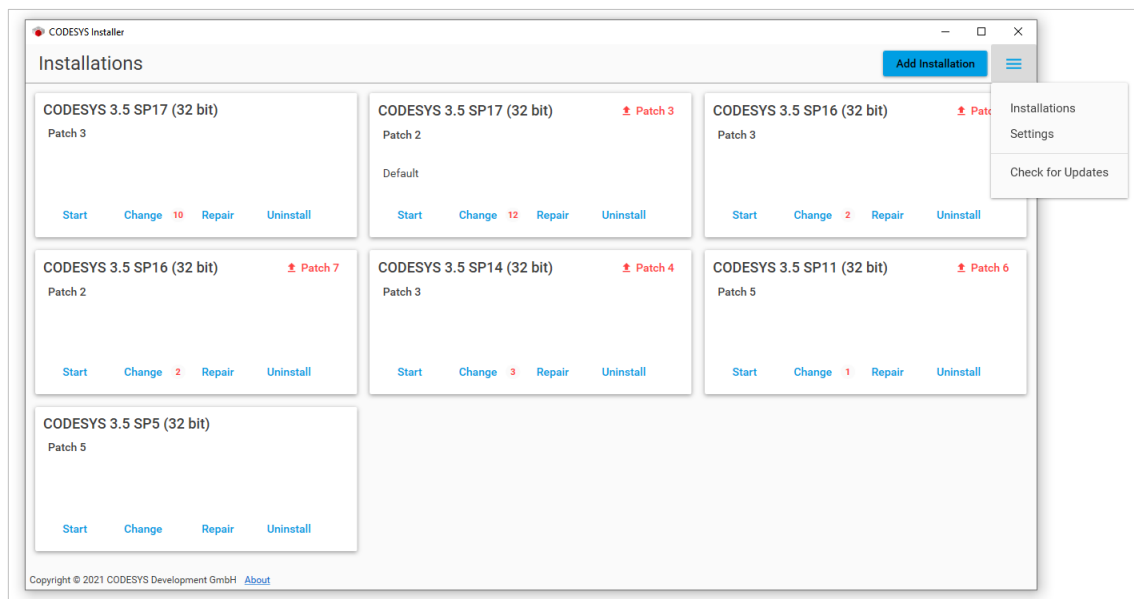


Рисунок 1.1 – Загальний вигляд інтерфейсу Codesys Installer

На вкладці **Settings** можна вибрати:

- канал постачання оновлень для всіх оточень за замовчуванням (релізні версії, бета-версії або відключити оновлення);
- канал постачання оновлень плагінів для всіх оточень за замовчуванням (релізні версії, бета-версії або відключити оновлення);
- версію сервісу **CODESYS Gateway**, яка буде за замовчуванням запускатися під час старту сервісу;
- версію віртуального контролера **CODESYS Control Win V3**, яка буде за замовчуванням запускатися під час старту сервісу.

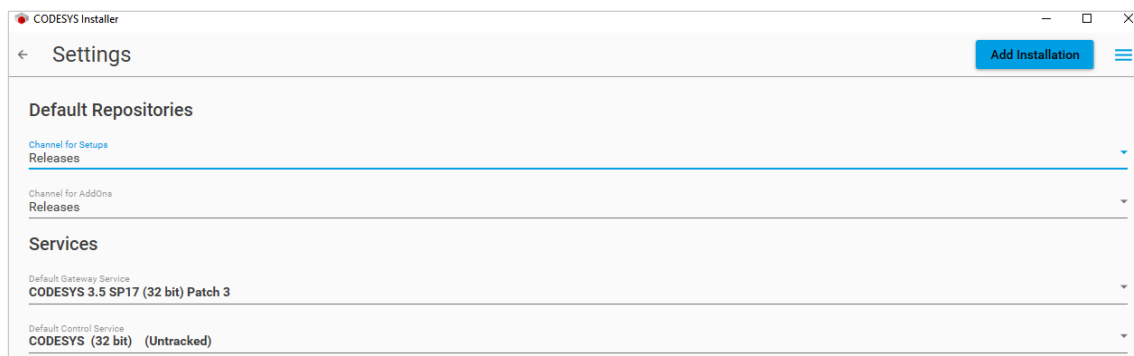


Рисунок 1.2 – Вигляд вкладки Settings застосунка Codesys Installer

## Модуляризація середовища програмування та значущі зміни

На вкладці конкретного оточення, що відкривається за кнопкою **Change**, можна:

- перейти в директорію встановлення цього оточення;
- вибрати канал постачання оновлень версій середовища (релізні версії, бета-версії або відключити оновлення);
- вибрати канал постачання оновлень версій плагінів (релізні версії, бета-версії або відключити оновлення);
- подивитися список встановлених плагінів і пакетів або видалити їх (вкладка **Installed**);
- встановити додаткові плагіни та приклади (вкладка **Browse**);
- оновити версії плагінів (вкладка **Updates**);
- встановити додаткові пакети (.package) - наприклад, пакети таргет-файлів (кнопка **Install File**);
- експортувати або імпортувати конфігурацію оточення<sup>3</sup> цієї версії середовища (кнопки **Export Config** та **Import Config**).

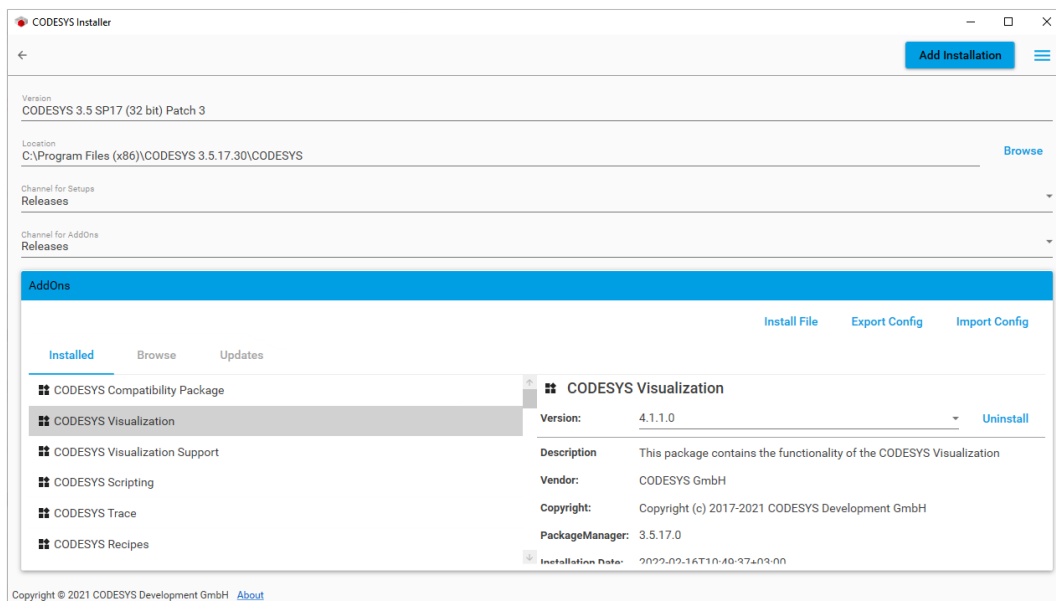


Рисунок 1.3 – Вигляд вкладки Change-Installed для обраного оточення

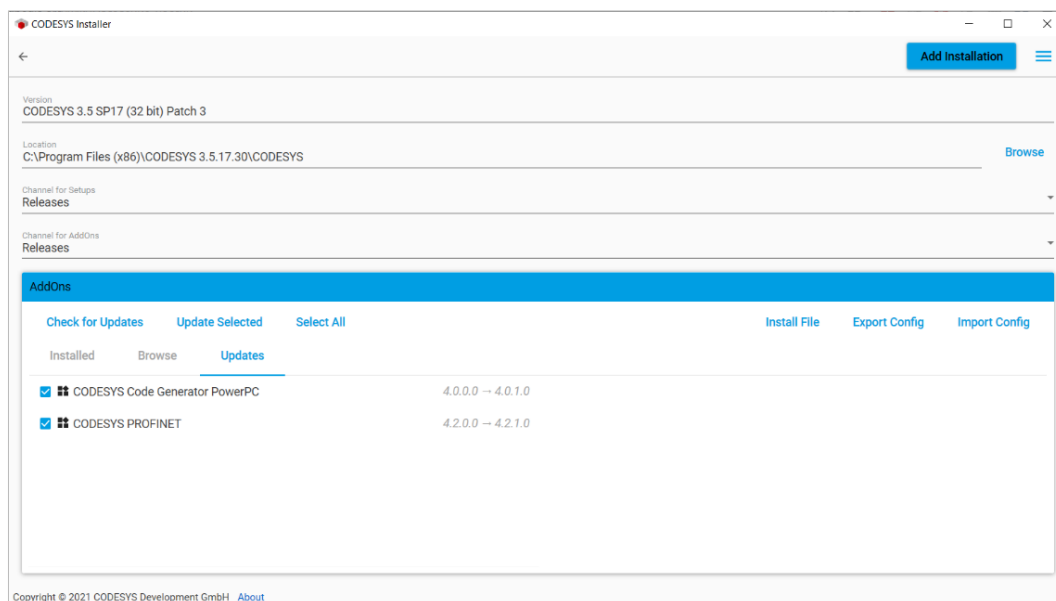


Рисунок 1.4 – Вигляд вкладки Change-Updates для обраного оточення

<sup>3</sup> Конфігурація являє собою файл формату **.installation-config** (фактично JSON), який містить перелік встановлених плагінів цієї версії середовища із зазначенням їхніх версій

## Модуляризація середовища програмування та значущі зміни

Список доступних оновлень плагінів можна також подивитися прямо в середовищі CODESYS у **Центрі сповіщень**:

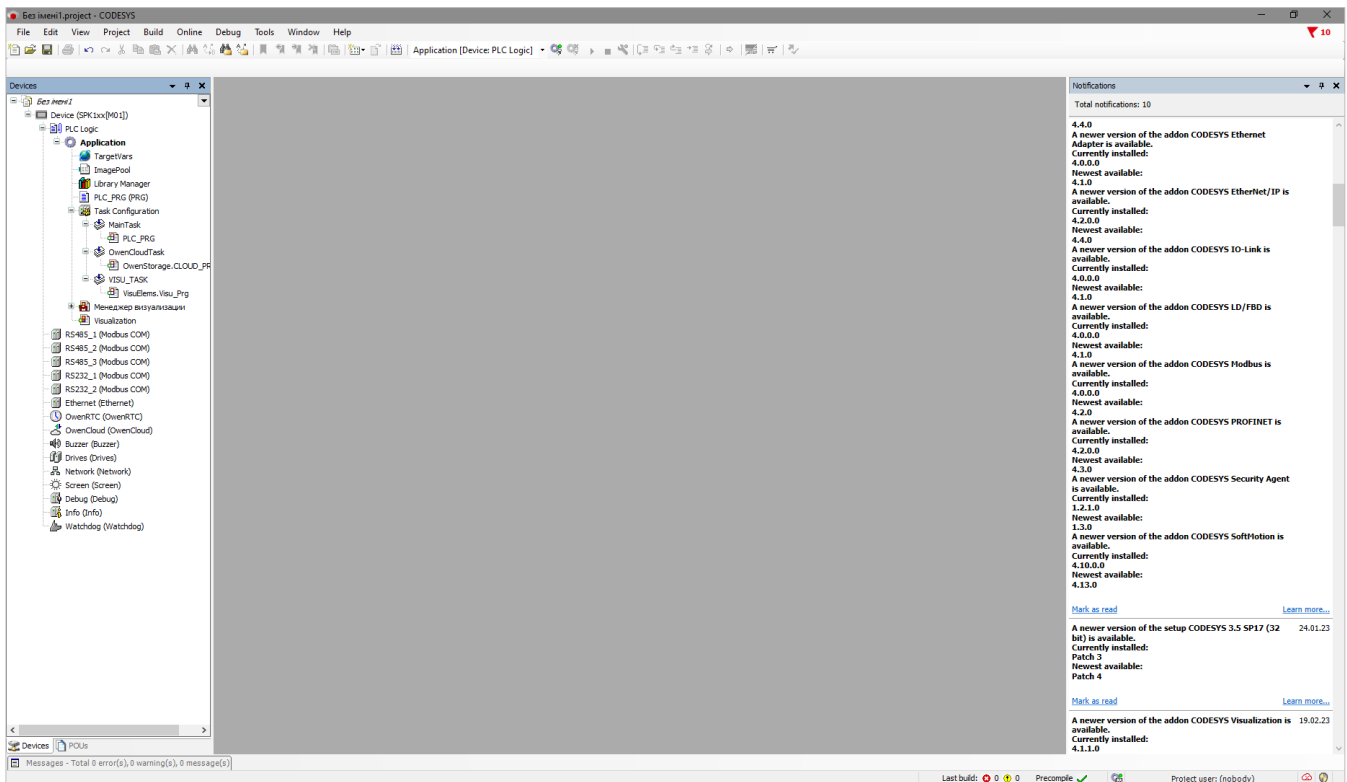


Рисунок 1.5 – Зовнішній вигляд вбудованого центру сповіщень

### 1.4. Сумісність з попередніми версіями

Питання сумісності - це найскладніший момент з точки зору користувача, який принесла модуляризація CODESYS. Якщо раніше CODESYS підтримував сумісність проєктів "від низу до верху" (у новій версії CODESYS можна було відкрити проєкт, створений у старій версії), то після модуляризації середовища справи йдуть дещо складніше.

Технічні рішення, описані нижче, були свідомо прийняті не через відсутність можливості іншої реалізації, а з позиції осмисленого обмеження внутрішньої складності CODESYS.

#### 1.4.1. Сумісність проєктів

Ключове питання цього пункту - чи зможе конкретне оточення CODESYS відкрити проєкт або бібліотеку, створені в іншому оточенні?

Раніше існуючий механізм не змінився. Поведінка ідентична версіям CODESYS до модуляризації (**версіям до SP17**). Якщо в проєкті є дані, які не можуть бути прочитані або інтерпретовані в поточному оточенні, оскільки вони були створені в іншому специфічному оточенні, то відповідні об'єкти позначаються піктограмою "червоний хрест" у дереві проєкту і мають постфікс "incomplete" (якщо редактор об'єкта можна відкрити) або "unknown" (якщо редактор об'єкта відкрити не можна). В обох випадках проєкт не можна завантажити в контролер (оскільки це може призвести до невизначеної поведінки користувацького додатку) і доступна тільки команда "Зберегти як" (для запобігання випадковому перезапису початкового проєкту, а отже, втрати даних).

Така реалізація довела свою ефективність за багато років існування CODESYS V3.

### 1.4.2. Сумісність коду

Ключове питання цього пункту - чи можна в одному оточенні CODESYS завжди згенерувати код проєкту, бінарно еквівалентний коду того самого проєкту, відкритого в іншому оточенні? Простіше кажучи: чи можна відкрити проєкт в оточенні, яке відрізняється від того, в якому цей проєкт був створений, та підключитися до контролера без онлайн-зміни або завантаження?

Для підтримки такої можливості раніше і аж до CODESYS V3.5 SP17 (включно з SP17) використовувався механізм версій компілятора. У версії CODESYS V3.5 SP18 цей механізм усунуто. Якщо вам потрібно мати можливість під'єднатися до контролера, не виконуючи онлайн-зміну або завантаження додатку, то слід відкрити проєкт саме в тому середовищі CODESYS, у якому його було створено.

Є низка вагомих причин, які спонукали розробників CODESYS відмовитися від механізму версій компілятора:

- за згенерований код відповідає не тільки компілятор, а й редактори мов програмування та конфігуратори протоколів обміну. Оскільки тепер вони виділені в плагіни - в принципі, більше не може бути однієї уніфікованої версії компілятора для конкретної версії середовища. Розробники вважають, що спроба об'єднати безліч різних версій компілятора для конкретних плагінів у свого роду профіль версії компілятора була б надмірно складною з точки зору розуміння такого механізму користувачем. Для CODESYS UML (частина CODESYS Professional Developer Edition) протягом багатьох років існувала окрема версія генерації мовних моделей, але навіть із цим одним доповненням така концепція не зарекомендувала себе на практиці (не кажучи вже про те, що починаючи з SP17 з'явилася безліч окремих плагінів);
- навіть у найпростішому випадку, використання потрібної версії компілятора вже вимагає від користувача конкретних знань про проєкт. У складніших випадках виправити випадково неправильно обрану версію компілятора часто було вкрай проблематично. Користувачі, які мали невдалий досвід із цим або, як правило, не знали, яка саме версія компілятора їм потрібна, і раніше встановлювали на ПК кілька версій CODESYS, щоб мати можливість гарантованого під'єднання до контролерів, що експлуатуються, без онлайн-зміни або перезапису проєкту, якщо це потрібно для їхнього обслуговування. Як буде описано нижче - тепер у такій ситуації користувач в інтерфейсі середовища побачить відповідне діалогове вікно;
- кожна нова версія компілятора збільшує кодову базу CODESYS. З одного боку, це негативно позначається на продуктивності середовища і розмірі дистрибутива. З іншого боку, розробники просто фізично не можуть протестувати всі комбінації цієї кодової бази. Крім того, оскільки підтримка і доопрацювання версій компілятора також певною мірою пов'язана з потенційним додаванням нових помилок, то це прямо суперечить постійно зростаючим вимогам до якості. Інакше кажучи, розробники CODESYS не можуть гарантувати, що механізм ручного вибору версій компілятора буде надійно працювати у всіх випадках, і вони потрапили б у дуже скрутне становище, якби їм би довелося зробити цей механізм ще складнішим через модуляризацію;
- резюмуючи все вищесказане: з точки зору користувача буде зручно взагалі не мати справи з версією компілятора. Слід ще раз підкреслити, що підключення до контролера без онлайн-зміни або завантаження проєкту буде доступне тільки під час відкриття проєкту в тому оточенні CODESYS, у якому його було створено (і планується додавання спливаючих підказок у тих випадках, коли підключення неможливе, із зазначенням необхідної версії середовища).

### 1.4.3. Сумісність IDE і системи виконання

Ключове питання цього пункту - наскільки встановлена користувачем версія середовища розробки CODESYS сумісна із системою виконання контролера?



## Модуляризація середовища програмування та значущі зміни

У цьому сенсі в SP17 не відбулося жодних змін. По суті, версії середовища розробки та системи виконання контролера не прив'язані одна до одної<sup>4</sup> за винятком таких обмежень:

- у новій версії середовища розробки можуть бути додані функції, які не підтримуватимуться старою системою виконання;
- стара версія середовища розробки може не підійти для програмування ПЛК з більш новою версією системи виконання, якщо це пов'язано зі змінами налаштувань безпеки (наприклад, змінився механізм управління користувачами або були підтримані нові алгоритми шифрування).

### 1.5. Рекомендації для користувачів

Сценарії використання, описані в цьому розділі, не є взаємовиключними - користувач може комбінувати їх. Розробники CODESYS вважають, що ці сценарії є типовими, і орієнтуються на них під час розроблення середовища. У майбутньому планується оптимізувати використання CODESYS саме для цих сценаріїв.

Таблиця 1.1 – Сценарії та рекомендації по роботі с Codesys SP17

Сценарій	Рекомендація	Підтримка з боку середовища
Щоденна робота над поточними проєктами.	Оновлювати версії CODESYS і плагінів у міру їхнього випуску. Розробники CODESYS безперервно працюють над додаванням нового функціоналу, виправленням помилок і оновленнями безпеки, тож новіша версія завжди є найкращим варіантом. Немає причин триматися за старі версії <sup>4</sup> .	CODESYS Installer показує всі доступні оновлення середовища і плагінів. Їх можна завантажити та встановити кількома кліками миші.  Крім того, Центр повідомлень, інтегрований у CODESYS, також містить список доступних оновлень (для конкретної запущеної версії середовища). Натиснувши на сповіщення, можна відразу перейти до CODESYS Installer - тож немає сенсу тримати його запущеним постійно.  Усі дистрибутиви та оновлення, що завантажуються, з метою безпеки забезпечені цифровим підписом. Перевірка підпису відбувається під час спроби встановлення пакета.
Щоденна командна робота над поточними проєктами в ситуації, коли всі розробники повинні використовувати одну й ту саму версію CODESYS.	Див. вище.  Один із розробників самостійно встановлює доступні оновлення та тестує їх. Після успішного проходження перевірки ці оновлення встановлюються всіма учасниками команди.	У CODESYS Installer можна експортувати та імпортувати конфігурацію оточення конкретної версії середовища для її перенесення на інший ПК. Цей механізм також доступний з командного рядка, тому він дає змогу автоматизувати процес розгортання оновлень.

<sup>4</sup> І тим не менш, найбільш правильний (що гарантує найменше число потенційних проблем) підхід для користувача - використовувати версію середовища розробки, яка в точності збігається з версією системи виконання контролера.

**Модуляризація середовища програмування та значущі зміни**

Продовження таблиці 1.1

Сценарій	Рекомендація	Підтримка з боку середовища
Обслуговування експлуатованого контролера без зміни наявного проекту. Має бути гарантія того, що можна під'єднатися до контролера без онлайн-зміни або завантаження проекту	Необхідна версія CODESYS, у якій було створено цей проект. Гарантується генерація бінарно еквівалентного коду без необхідності розбиратися у версіях компілятора	<p>Якщо під час спроби під'єднання до контролера виявляються відмінності між поточним оточенням і тим, у якому було створено проект, - з'являється діалогове вікно з можливістю встановлення необхідного оточення. Після цього проект буде автоматично відкрито в новому оточенні.</p> <p>Інший варіант - у цьому ж вікні вибрати потрібні проекту версії плагінів (якщо вони встановлені на цьому ПК).</p> <p>У CODESYS Installer кожному оточенню можна задати свою назву - це дасть змогу користувачам не заплутатися в них.</p> <p>Щоб таке оточення не видавало постійно повідомлення про доступність нових версій - вони за замовчуванням будуть для нього відключені</p>
Внесення змін до раніше створених проектів	Внесення змін до проекту здійснюється в поточному оточенні. Оскільки в код у будь-якому разі вносяться зміни, то про можливість підключення до контролера без онлайн-зміни або завантаження проекту можна не замислюватися	Як і раніше, CODESYS може відкривати проекти, створені в попередніх версіях. Якщо необхідний плагін не встановлено на ПК, то його можна завантажити та встановити безпосередньо під час відкриття проекту. В іншому разі цей проект не вийде завантажити в контролер

Ось такий вигляд має вікно, що з'являється під час відкриття проекту в оточенні, яке відрізняється від того, в якому було створено проект:

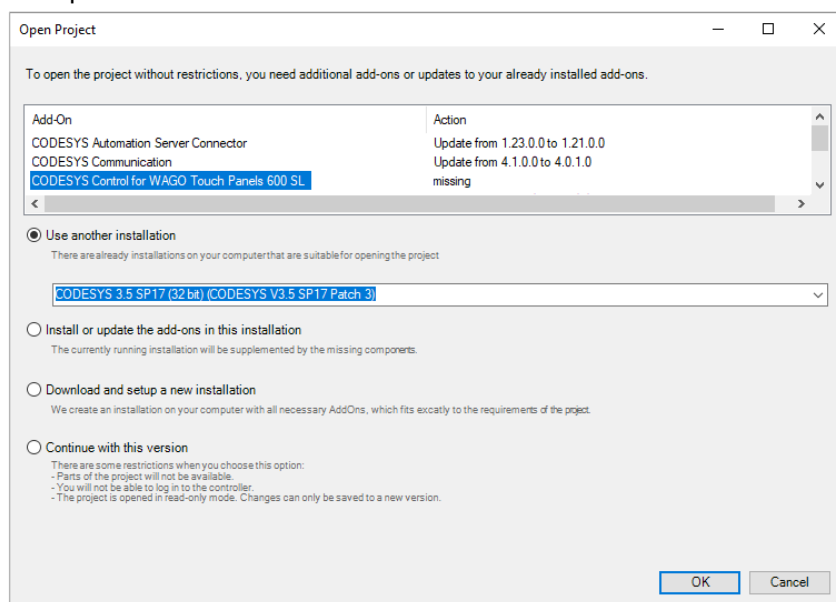


Рисунок 1.6 – Зовнішній вигляд вікна відкриття проекту, створеного в іншій версії оточення

У користувача є 4 варіанти дій у цій ситуації:

- **Use another installation** - вибрати інше оточення для відкриття проекту;
- **Install or update the add-ons in this installation** - встановити відсутні версії плагінів для можливості роботи з проектом у цьому оточенні;
- **Download and setup a new installation** - завантажити та встановити нове оточення, яке точно відповідає цьому проекту;
- **Continue with this version** - продовжити роботу з проектом у цьому оточенні.

### 1.6. Менеджер пакетів і пов'язані з ним можливі проблеми

Починаючи з версії **CODESYS V3.5 SP17 Patch 3** менеджер пакетів відсутній - замість нього в меню **Інструменти** додано команду запуску **CODESYS Installer**:

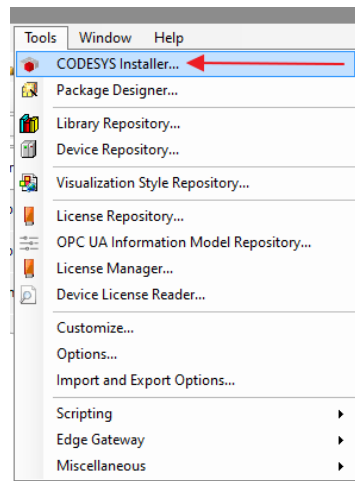


Рисунок 1.7 – Меню для виклику Codesys Installer із середовища розробки

Встановлення пакетів (.package) тепер виконується через CODESYS Installer:

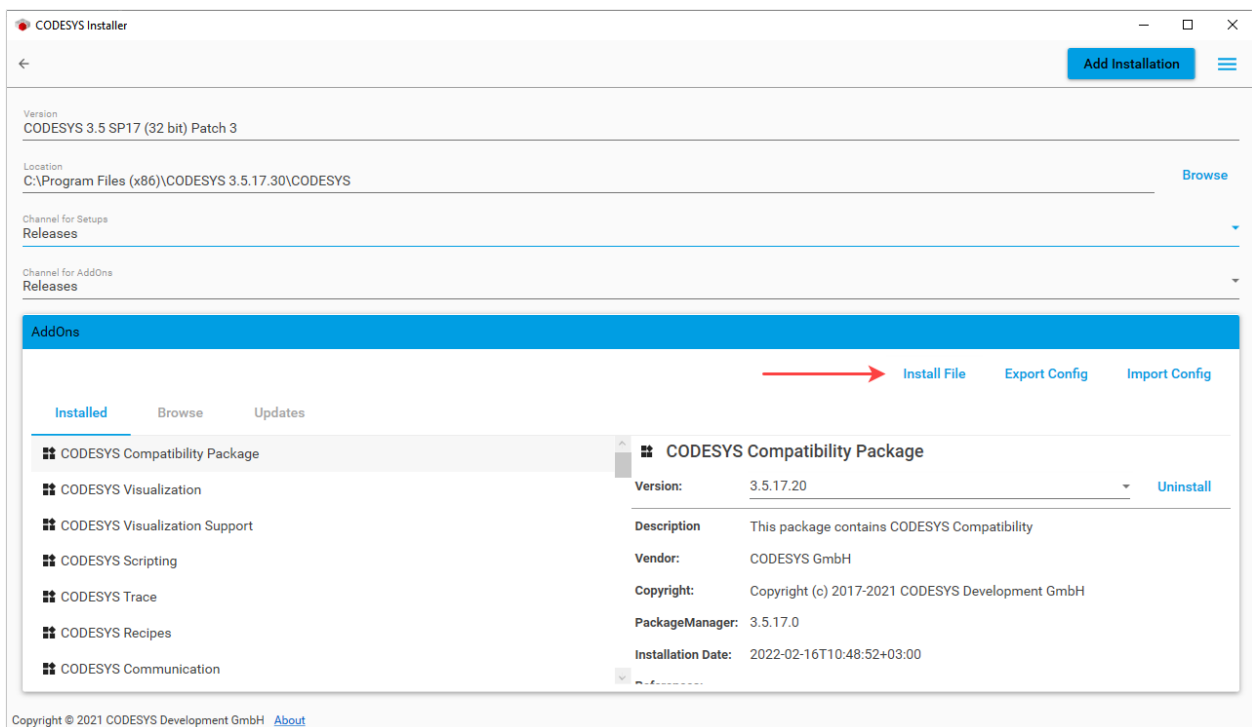


Рисунок 1.8 – Зовнішній вигляд вікна Codesys Installer під час пошуку пакетів (\*.package)

## Модуляризація середовища програмування та значущі зміни

За допомогою вкладки **Browse** можна знайти та встановити додаткові пакети - зокрема, пакет **CODESYS Package Manager**. Тоді в меню **Інструменти** команда запуску CODESYS Installer буде замінена на команду запуску менеджера пакетів.

### УВАГА



*Фактично менеджер пакетів у цій ситуації перестане працювати належним чином - з його допомогою не вийде встановлювати або видаляти пакети. Більш того, є ймовірність, що і встановити/видалити їх з CODESYS Installer теж більше не вдасться - потрібно видалити CODESYS і встановити його заново. Тому настійно не рекомендується встановлювати в CODESYS V3.5 SP17 Patch 3 і плагін менеджера пакетів до нього.*

Цю помилку виправлено у версії **CODESYS Installer 1.3.0** - у ній усунуто можливість встановлення плагіна менеджера пакетів.

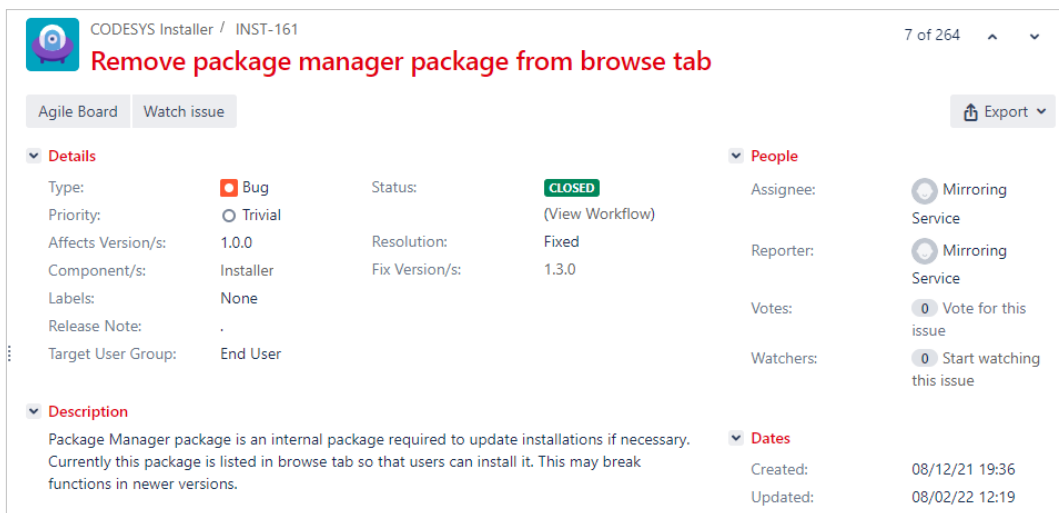


Рисунок 1.9 – Повідомлення у баг-трекері CODESYS

## 1.7. Імпорт пакетів минулих версій під час першого запуску та пов'язані з цим можливі проблеми

Після встановлення **CODESYS V3.5 SP17 Patch 3** під час першого запуску середовища може з'явитися таке вікно:

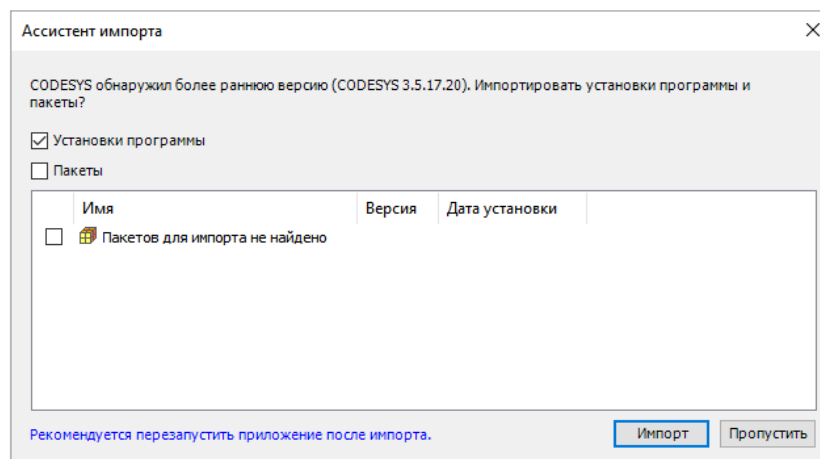


Рисунок 1.10 – Вікно запиту на імпорт пакетів при першому запуску Codesys SP17

**УВАГА**


**Обов'язково натисніть "Пропустити". Під час спроби імпорту пакетів попередніх версій можливе пошкодження конфігураційних файлів, унаслідок чого CODESYS перестане запускатися (після натискання на ярлик не буде жодної реакції). Для вирішення проблеми потрібно видалити CODESYS і встановити його наново.**

**1.8. Помилка при відкритті документації бібліотек та інструкція з її усунення**

Починаючи з версії **V3.5 SP17** модуль формування документації бібліотек виділено в окремий плагін **CODESYS Library Documentation Support** (документація відображається у менеджері бібліотек). За замовчуванням цей пакет може бути не встановлений - у цьому разі під час відкриття документації бібліотеки з'явиться помилка:



Рисунок 1.11 – Повідомлення про відсутність плагіну відображення документації бібліотек

Для її усунення потрібно встановити потрібний плагін через CODESYS Installer:

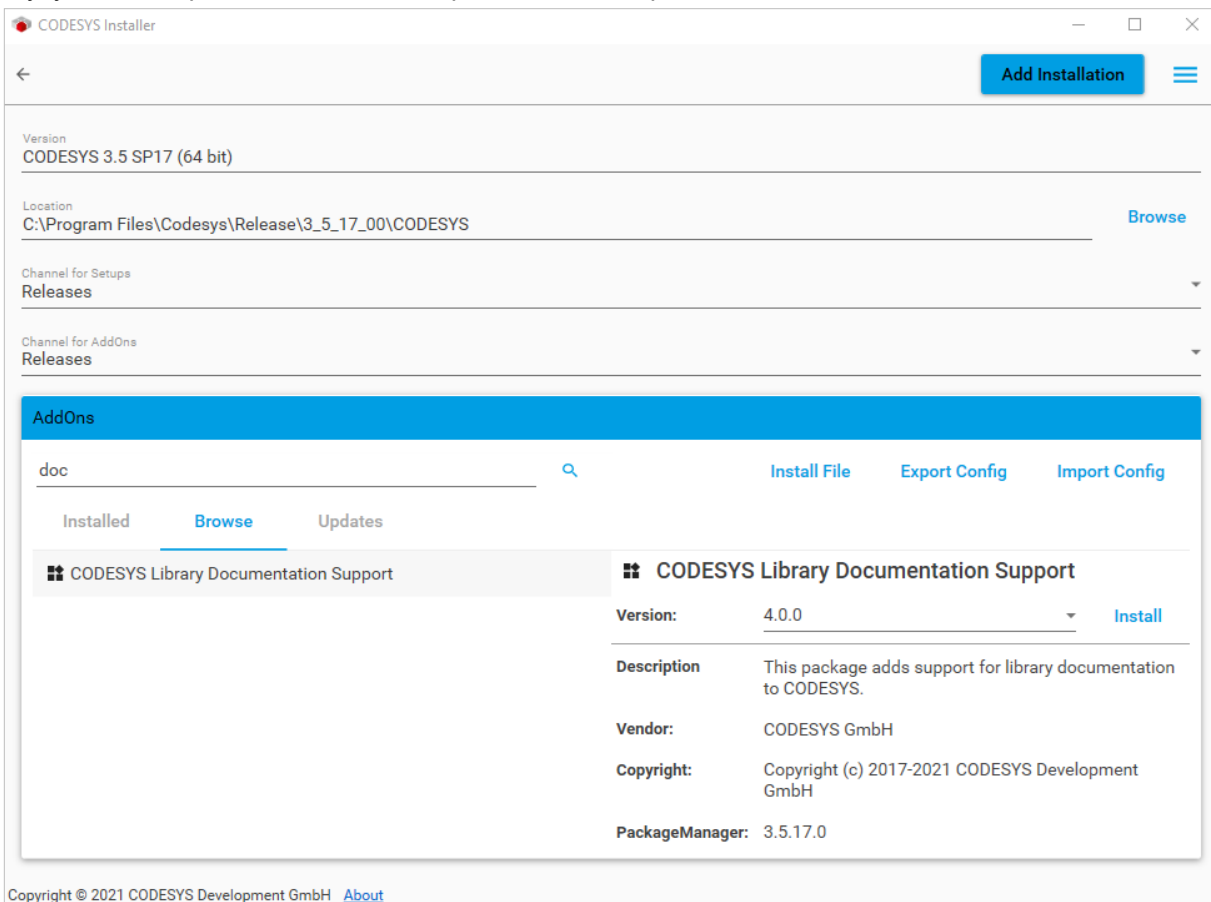


Рисунок 1.12 – Вікно Codesys Installer з вибраним для встановлення плагіном Library Documentation Support

**УВАГА**


Наразі можлива ситуація, коли в документації до бібліотеки не відображається чи відображається некоректно опис перерахувань та перерахувань з підтримкою списків текстів (ENUM та ENUM With TextList Support). Така поведінка трапляється не на кожному ПК. Відповідна інформація все зафіксована у баг-трекеру CODESYS. Номери заявок: CDS-84906 та CDS-84908.

**1.9. Типові запитання під час встановлення V3.5 SP17**
**1.9.1. "Я можу створити POU тільки на ST, де інші мови"?**

Редактори мов LD/FBD/IL, CFC і SFC виділені в окремі плагіни, що встановлюються через CODESYS Installer:

- CODESYS LD/FBD - редактори LD, FBD і IL;
- CODESYS CFC - редактор CFC;
- CODESYS SFC - редактор SFC.

**1.9.2. Під час встановлення виникла помилка встановлення пакета "Compatibility Package"**

- видаліть CODESYS. Перевірте, що в директорії інсталяції не залишилося жодних файлів (за необхідності - видаліть їх вручну);
- видаліть вміст директорії C:\ProgramData\AP\PlugInCaches;
- перевстановіть CODESYS;
- якщо проблема збережеться - видаліть CODESYS, видаліть усі пакети Microsoft Visual C++ Redistributable і встановіть CODEYS наново.

**1.9.3. "Я намагаюся встановити пакет 'A', в якому є залежність від пакета 'B', але пакет 'B' у мене не встановлений - тому з'являється помилка"**

Встановіть пакет через CODESYS Installer - він автоматично встановить усі необхідні пакети. Або встановіть спочатку пакет 'B' вручну.

**1.9.4. "Я не можу знайти CODESYS Installer"**

Пуск - Усі програми - CODESYS<sup>5</sup> - CODESYS Installer



Рисунок 1.13 – Дека доступу до Codesys Installer через меню Пуск Windows

Повний шлях до програми: C:\Program Files (x86)\CODESYS\APInstaller\APInstaller.GUI.exe

<sup>5</sup> У більш ранніх дистрибутивах папка називалася 3S CODESYS, а тепер - CODESYS

### **1.9.5. "Я не можу використовувати свіжу версію якогось плагіна (наприклад, плагіна візуалізації) для роботи з проектом, створеним у старій версії CODESYS"**

Якщо ви хочете використовувати у своєму проекті профіль візуалізації версії 4.1.0.0, то вам необхідно встановити плагін саме цієї версії (а не, наприклад, 4.2.0.0 або 4.0.0.0).

Якщо ви хочете використовувати у своєму проекті версію 4.0.0.0 або більш ранню (наприклад, 3.5.16.10), то вам необхідно встановити плагін саме цієї версії (а не, наприклад, 4.1.0.0).

Це пов'язано з тим, що завдяки модуляризації версія CODESYS більше не жорстко прив'язана до конкретної версії плагіна. Таким чином, для забезпечення максимальної сумісності користувач тепер може бути впевненим, що під час роботи з проектом у конкретному профілі візуалізації він використовує відповідну йому версію плагіна.

Варіанти поведінки користувача:

- оновити версії плагінів під час відкриття проекту. Якщо ви відкриваєте старий проект, оновить профіль візуалізації до останньої з доступних версій. Проблема лише в тому, що ви не зможете під'єднатися до контролера, у який вже завантажено проект, без онлайн-зміни або перезапису цього проекту;
- використовувати CODESYS Installer, щоб встановити оточення саме з тими версіями плагінів, які були використані при створенні проекту. Тоді буде можливість під'єднатися до контролера, у який уже завантажено проект, без онлайн-зміни та перезавантаження проекту.

У майбутньому, коли ви відкриєте проект, створений за допомогою плагіна візуалізації 4.1.0.0 в оточенні з плагіном візуалізації 4.2.0.0, вам буде запропоновано або переключитися на потрібне оточення, або створити нове оточення, яке буде підходящим для того проекту.

Такі ж правила сумісності тепер діють майже для всіх плагінів - якщо ви хочете відкрити якийсь старий проект, то вам буде потрібно або встановити версії плагінів, які використовуються при його створенні, або оновити в проекті версії плагінів (якщо це можливо у вашому випадку). Після оновлення плагінів онлайн-підключення до контролера, до якого вже завантажено цей проект, може бути неможливим - потрібно завантажити проект наново.

### **1.9.6. "У деяких комунікаційних драйверах відсутні окремі вкладки»**

Наприклад, немає вкладки 'Загальні' у компонентів Modbus".

Встановіть плагін відповідного протоколу (наприклад, CODESYS Modbus) через CODESYS Installer.

Джерело інформації: <https://faq.codesys.com/pages/viewpage.action?pageId=105152517>

## **1.10. Автоматична активація керування користувачами для віртуального контролера**

Починаючи з версії **CODESYS V3.5 SP17** для віртуального CODESYS Control Win V3 за замовчуванням активовано управління користувачами. Під час першого під'єднання до віртуального контролера потрібно створити користувача і під час кожного наступного під'єднання вводити його логін і пароль.

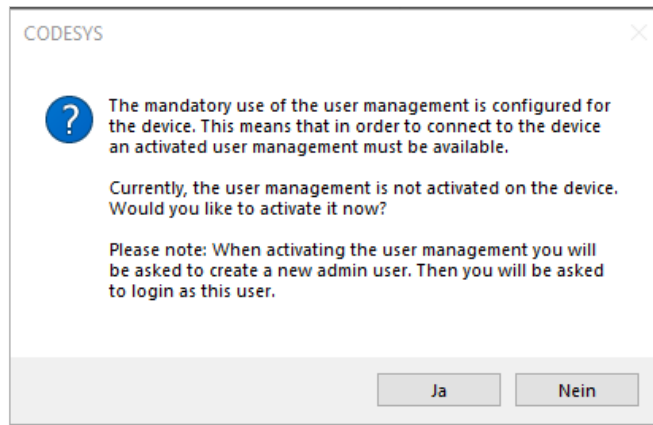


Рисунок 1.14 – Вікно запиту активування керування користувачами

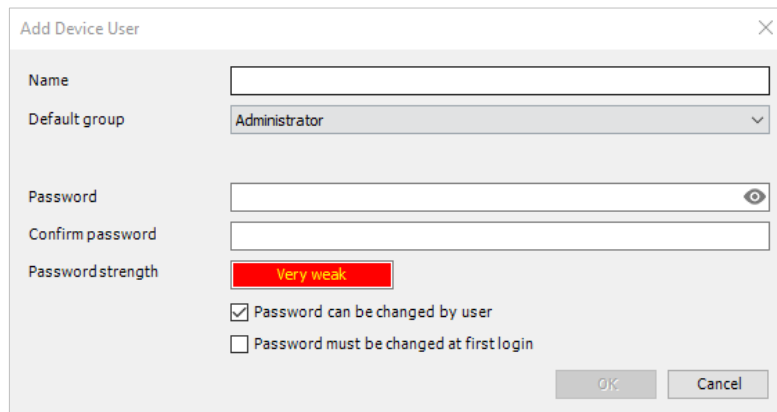


Рисунок 1.15 – Вікно створення користувача контролеру

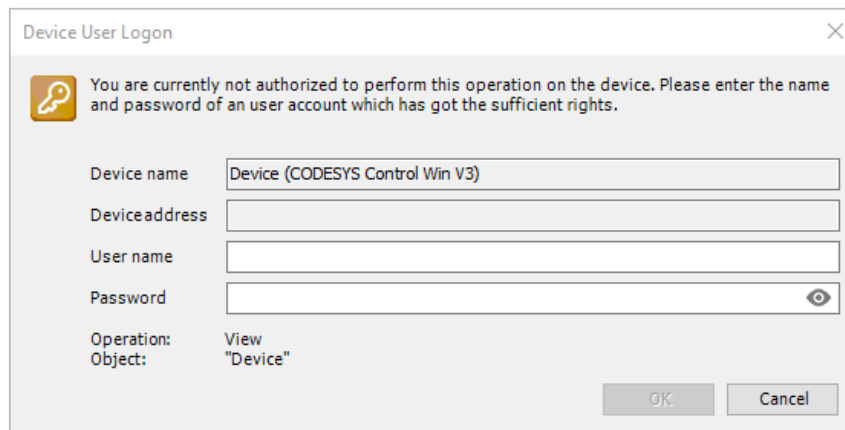


Рисунок 1.16 – Вікно авторизації при підключенні до контролера

Якщо це небажано, то можна вимкнути цю поведінку, відредагувавши файл C:\ProgramData\CODESYS\CODESYSControlWinV3\<ідентифікатор><sup>6</sup> \CODESYSControl.cfg: у секції [CmpUserMgr] розкоментуйте рядок

**;*SECURITY.UserMgmtEnforce=NO***

видаливши символ ';' на її початку. Має вийти так:

<sup>6</sup> Якщо у вас встановлено кілька версій CODESYS - то орієнтуйтеся на дату останньої зміни директорії



```

CODESYSControl.cfg – Блокнот
Файл  Правка  Формат  Вид  Справка
[СmpCAASStorage]
Memsys5SizeInMB=16

[СmpBACnet]
IniFile=bacstac.ini

[СmpUserMgr]
;allow usage of CODESYSControl runtime without activated user management (not recommended):
SECURITY.UserMgmtEnforce=NO ←
AsymmetricAuthKey=b52c068354161c2d2c51feb394cda70e8b4d9082

[СmpSecureChannel]
CertificateHash=863df12f29487c8e62dec9cbfadf255459cf3105
    
```

Рисунок 1.17 – Приклад редагування конфігураційного файлу віртуального ПЛК

Збережіть файл і перезапустіть віртуальний контролер, щоб зміни набули чинності.

Джерело інформації:

<https://faq.codesys.com/display/CDSFAQ/Disable+%28forced%29+device+user+management+as+of+SP17>

### 1.11. Помилка під час підключення до CODESYS OPC Server V3

Починаючи з версії **CODESYS V3.5 SP17** - OPC-сервер **CODESYS OPC Server V3** виключено з дистрибутиву CODESYS, його потрібно встановлювати окремо: <https://store.codesys.com/de/codesys-opc-da-server-sl.html>



Рисунок 1.18 – Вікно повідомлення у баг-трекері CODESYS

### 1.12. Зміна системи нумерації компонентів/плагінів на 4.x.x.x.x

Оскільки в результаті модуляризації розробка бібліотек/компонентів/плагінів тепер проводиться незалежно від розробки середовища програмування - то для них введена нова система версійності (**4.x.x.x.x**)

System_VisuElem3DPath = VisuElem3DPath, 4.1.0.0 (System)	VisuElem3DPath	4.1.0.0
System_VisuElemCamDisplayer = VisuElemCamDisplayer, 4.1.0.0 (System)	VisuElemCamDisplayer	4.1.0.0
System_VisuElemMeter = VisuElemMeter, 4.1.0.0 (System)	VisuElemMeter	4.1.0.0
System_VisuElems = VisuElems, 4.1.0.0 (System)	VisuElems	4.1.0.0
System_VisuElemsAlarm = VisuElemsAlarm, 4.1.0.0 (System)	VisuElemsAlarm	4.1.0.0
System_VisuElemsDateTime = VisuElemsDateTime, 4.1.0.0 (System)	VisuElemsDateTime	4.1.0.0
System_VisuElemsSpecialControls = VisuElemsSpecialControls, 4.1.0.0 (System)	VisuElemsSpecialControls	4.1.0.0
System_VisuElemsWinControls = VisuElemsWinControls, 4.1.0.0 (System)	VisuElemsWinControls	4.1.0.0
System_VisuElemTextEditor = VisuElemTextEditor, 4.1.0.0 (System)	VisuElemTextEditor	4.1.0.0
System_VisuElemTrace = VisuElemTrace, 4.1.0.0 (System)	VisuElemTrace	4.1.0.0
System_VisuElemXYChart = VisuElemXYChart, 4.1.0.0 (System)	VisuElemXYChart	4.1.0.0
system_visuinputs = visuinputs, 4.1.0.0 (system)	visuinputs	4.1.0.0
System_VisuNativeControl = VisuNativeControl, 4.1.0.0 (System)	VisuNativeControl	4.1.0.0

Рисунок 1.19 – Вікно доданих до проекту бібліотек за Visualization Manager

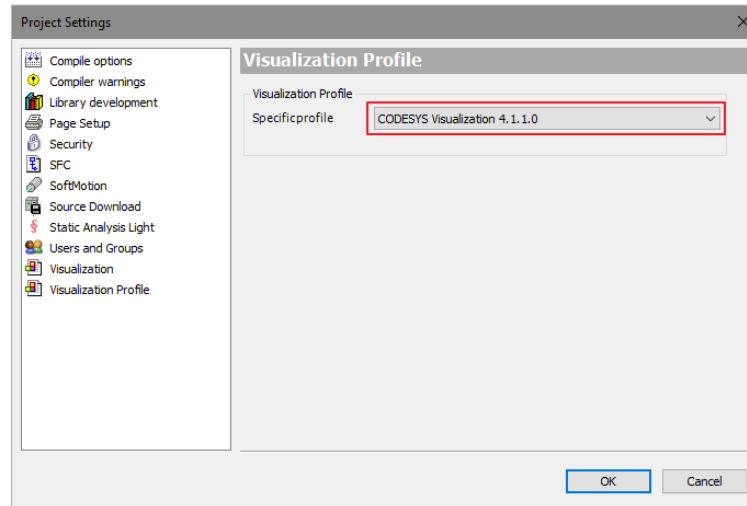


Рисунок 1.20 – Вікно зміни профілю візуалізації

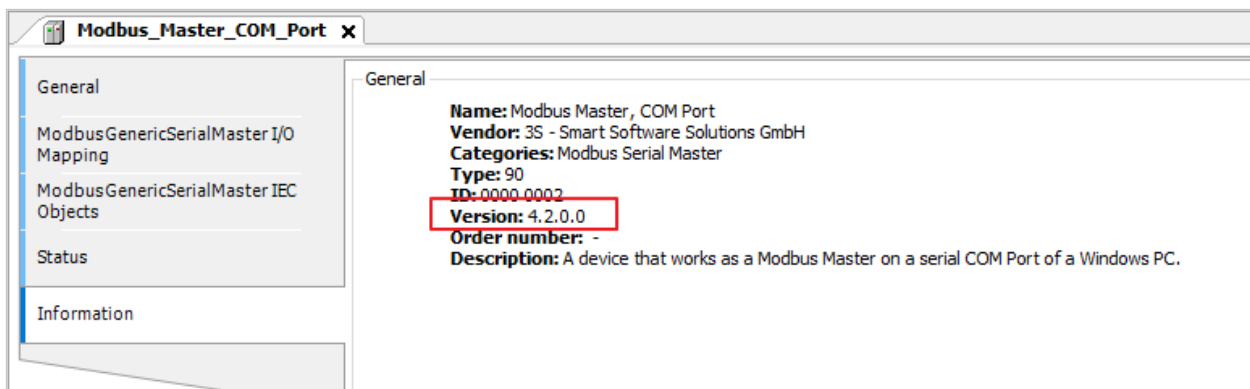


Рисунок 1.21 – Приклад версії компонента Modbus COM Port

## 2. Редактор мови CFC

### 2.1. Автоматичний контроль порядку виконання блоків у CFC

У мові CFC порядок виконання блоків визначається номерами, що відображаються в їхньому верхньому правому куті. За замовчуванням номер кожного блоку, що додається на полотно, залежить від його позиції по вертикалі відносно інших блоків (наприклад, якщо блок, що додається по вертикалі, буде розташовано між блоками з номерами 1 і 2, то він отримає номер 2, а вихідний блок 2 отримає номер 3). Під час переміщення блоків полотном їхні номери не змінюються, що може призвести до таких ситуацій ("значення з виходу одного блоку не передається на вхід іншого"):

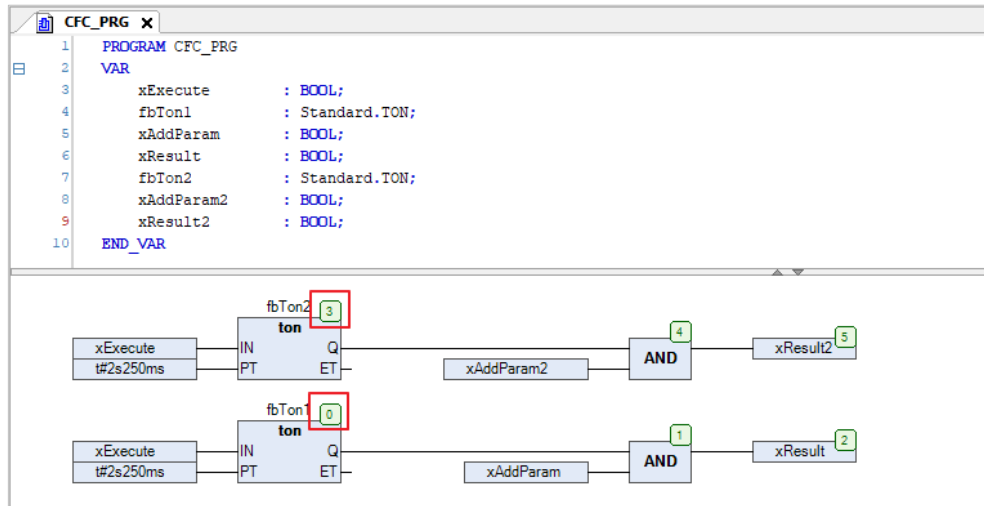


Рисунок 2.1 – Приклад некоректної нумерації порядку виконання блоків

Раніше користувачам доводилося ретельно контролювати номери блоків і за потреби змінювати їх через контекстне меню - зазвичай за допомогою команди **Execution Order by Data Flow** (Впорядкувати відповідно до потоку даних), яка перерозподіляла номери блоків за принципом "зліва направо, зверху вниз".

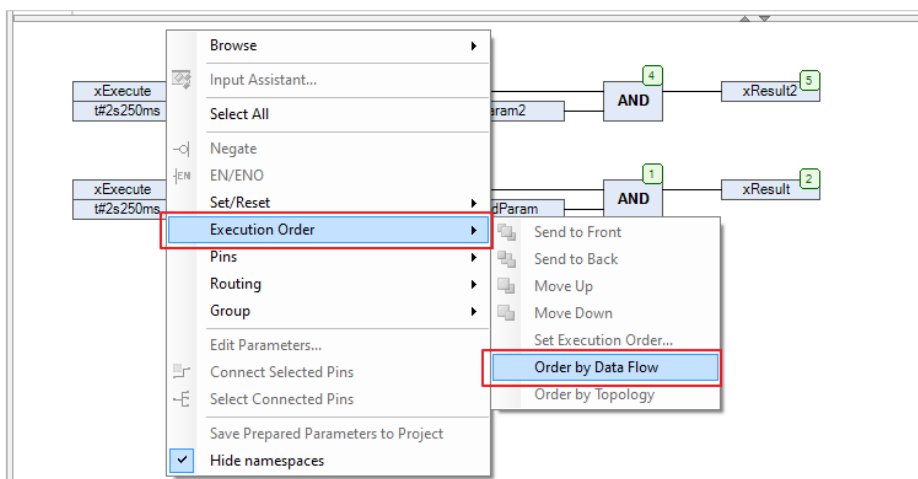


Рисунок 2.2 – Ручний спосіб змінити порядок виконання блоків

У новій версії CODESYS можна ввімкнути *автоматичний режим упорядкування номерів*. Для цього треба натиснути ПКМ на будь-який POU з мовою CFC, вибрати пункт **Properties** (Властивості), у вікні, що відкрилося, вибрати вкладку **CFC Execution Order** (Порядок виконання CFC) і задати порядок виконання **Auto Data Flow Mode** (Режим потоку даних). Режим **Explicit Execution Order Mode** (Заданого порядку виконання) відповідає стандартній поведінці з попередніх версій, описаних вище. За допомогою кнопки

**Apply to All CFCs** (Застосувати до всіх CFC) можна автоматично змінити це налаштування у всіх POU проекту.

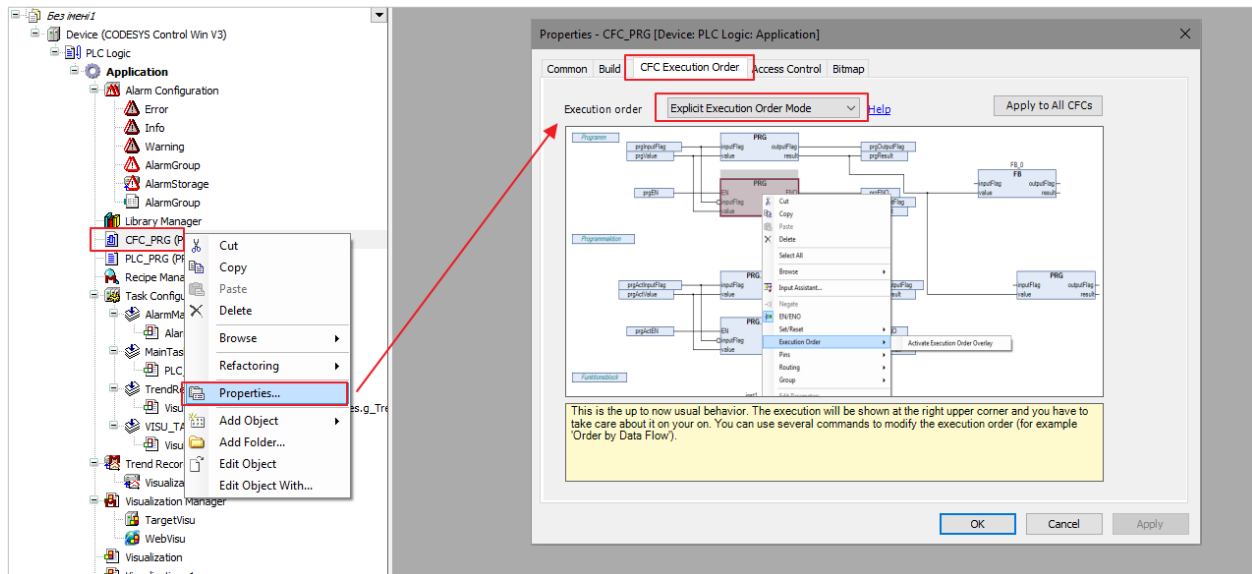


Рисунок 2.3 – Вікно вибору режиму встановлення порядку виконання блоків на полотні POU

У результаті номери блоків на полотні перестануть відображатися - вони розподілятимуться автоматично відповідно до потоку даних і змінюватимуться "на льоту" під час перетягування блоків по полотну. Якщо користувач хоче все-таки упевнитися в порядкових номерах блоків - то це можна зробити за допомогою спеціальної кнопки на панелі інструментів. Номери будуть підсвічуватися блакитним кольором і зникнуть після першого натискання на полотно.

#### ПРИМІТКА



**Кнопка перевірки порядку виконання блоків активна лише при встановленому Auto Data Flow Mode (Режим потоку даних).**

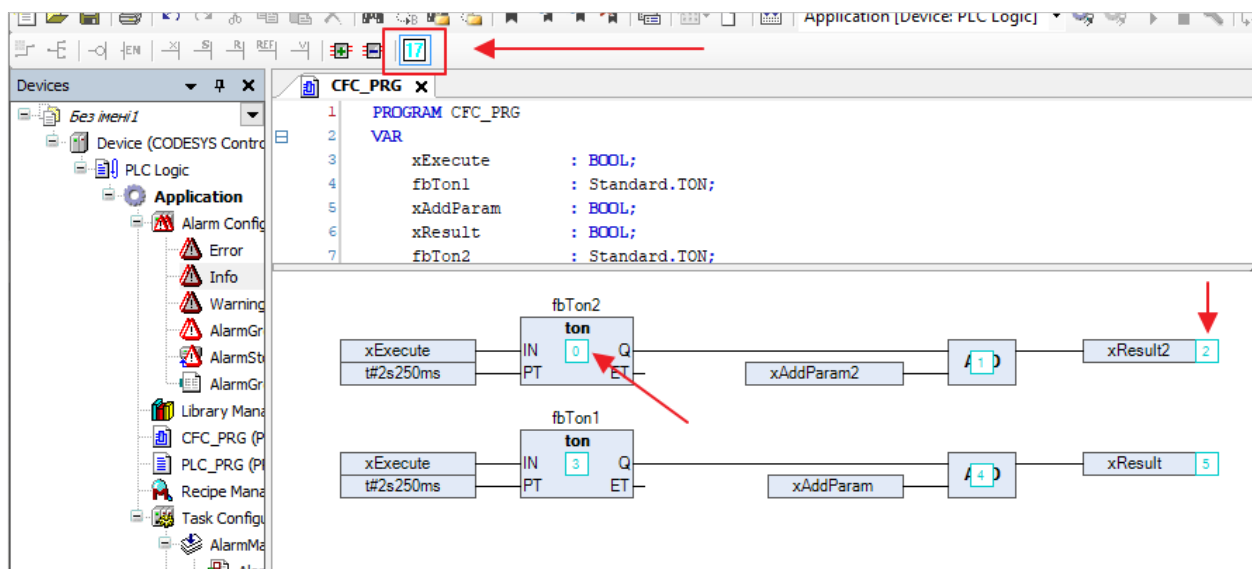


Рисунок 2.4 – Кнопка меню для перевірки автоматично встановленого порядку виконання блоків  
У деяких випадках хотілося б використовувати автоматичний розподіл номерів, але водночас мати можливість задавати їх вручну - зокрема, в тих блоках, які охоплені зворотним зв'язком. Для цього використовується спеціальна команда контекстного меню:

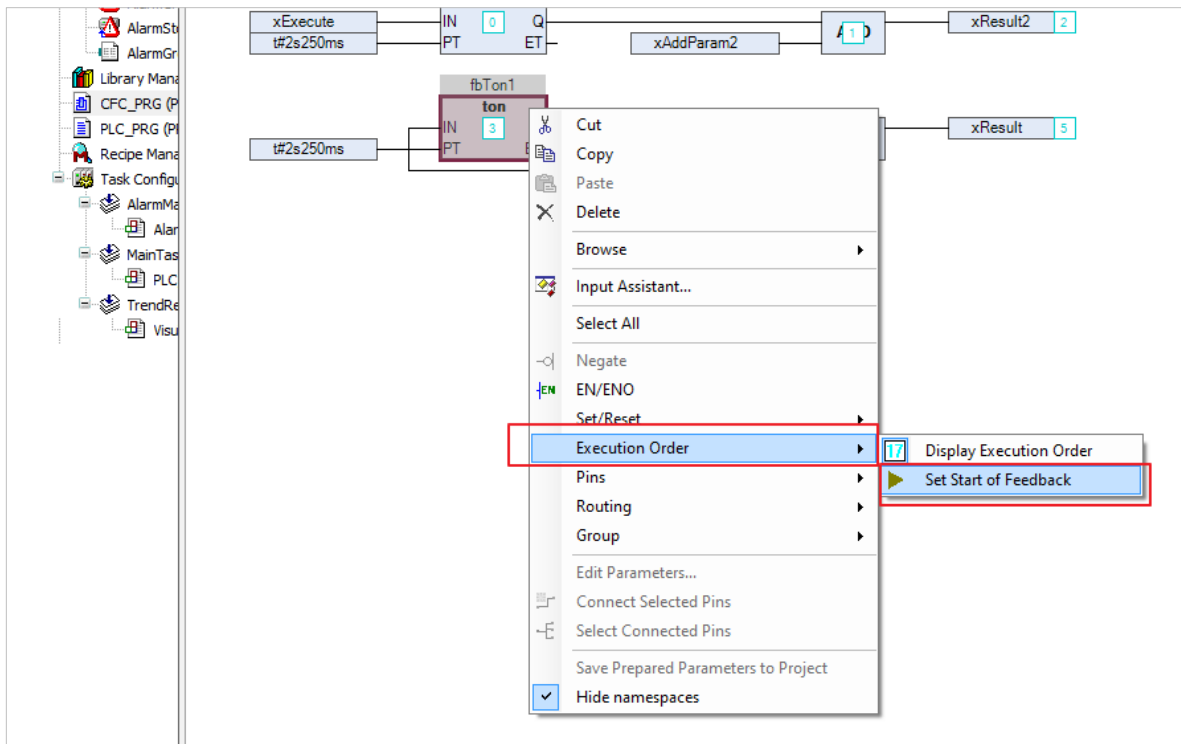


Рисунок 2.5 – Команда для ручного встановлення порядку виконання блоків

І ще одне дрібне поліпшення в CFC - у режимі налагодження значення тепер відображається не тільки на початку сполучної лінії, а й на її кінці. Це зручно на великих схемах із довгими лініями зв'язку.

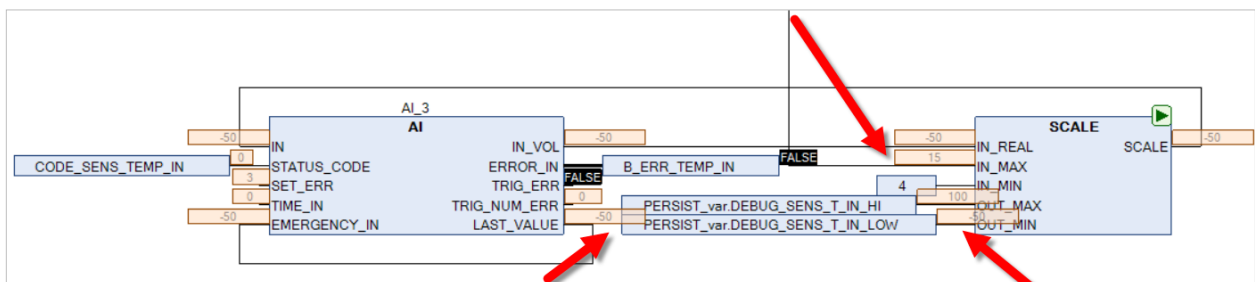



Рисунок 2.6 – Відображення значення змінних у обох кінцях ліній зв'язку

## 2.2. Оптимізація використання місця на полотні

У версії CODESYS SP17 добавили такі зміни:

- розмір блоків тепер підганяється під довжину їхніх імен;
- у назвах об'єктів тепер не відображаються простори імен (назви бібліотек, імена структур тощо) - замість цього відображається піктограма . Для відображення просторів імен потрібно або навести курсор на об'єкт для появи спливаючої підказки, або натиснути 2 рази ЛКМ на об'єкті для переходу в режим редагування його імені.

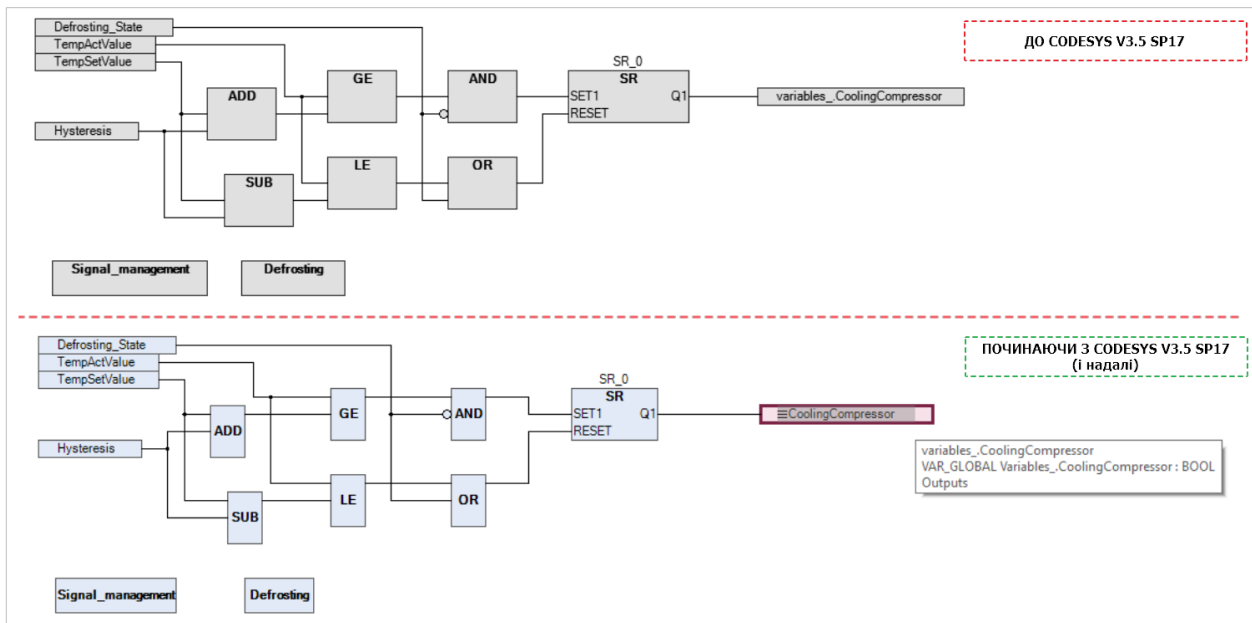


Рисунок 2.7 – Відображення відмінності у показі просторів імен

### 2.3. Підтримка темної теми

Для редактора мови CFC тепер підтримується темна тема.

Для зміни теми перейдіть до меню **Tools – Options - Text editor – Theme**.

(Інструменти - Опції - Текстовий редактор - Тема)

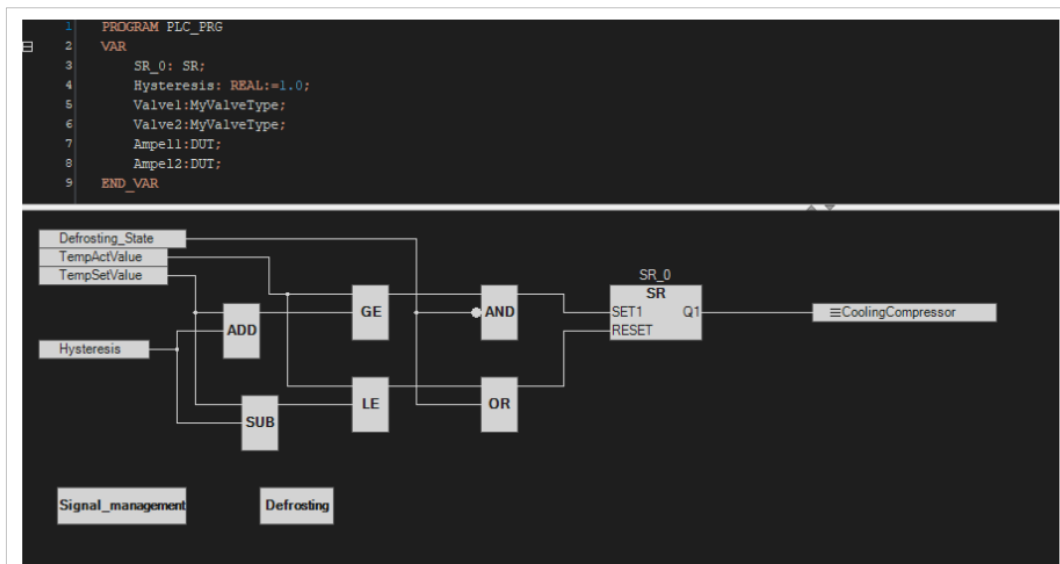


Рисунок 2.8 – Стандартне відображення темної теми редактору CFC

Див. демонстраційне відео: [https://www.codesys.com/fileadmin/data/News\\_Images/good-to-know/CODESYS-Clip-SP17-CFC-DarkTheme.mp4](https://www.codesys.com/fileadmin/data/News_Images/good-to-know/CODESYS-Clip-SP17-CFC-DarkTheme.mp4)

### 2.4. Зміна порядку входів/виходів блоку за допомогою "перетягування" мишею

Див. демонстраційне відео: [https://www.codesys.com/fileadmin/data/News\\_Images/good-to-know/CODESYS-Clip-SP17-CFC-RearrangePins.mp4](https://www.codesys.com/fileadmin/data/News_Images/good-to-know/CODESYS-Clip-SP17-CFC-RearrangePins.mp4)

## 2.5. Автоматичне додавання входів у блоках із розширюваним числом входів за допомогою перетягування зв'язку

Для блоків із розширюваним числом входів (ADD, AND тощо) тепер достатньо перетягнути зв'язки від змінної в тіло блоку - відповідний вхід буде додано автоматично.

Див. демонстраційне відео: [https://www.codesys.com/fileadmin/data/News\\_Images/good-to-know/CODESYS-Clip-SP17-CFC-LargerDropArea.mp4](https://www.codesys.com/fileadmin/data/News_Images/good-to-know/CODESYS-Clip-SP17-CFC-LargerDropArea.mp4)

## 2.6. Підтримка гарячих клавіш у редакторі CFC

Гарячі клавіші налаштовуються в меню **Tools – Customize – Keyboard – CFC**.  
(Інструменти - Налаштування - Клавіатура – CFC)

Див. демонстраційне відео: [https://www.codesys.com/fileadmin/data/News\\_Images/good-to-know/CODESYS-Clip-SP17-CFC-Shortcuts.mp4](https://www.codesys.com/fileadmin/data/News_Images/good-to-know/CODESYS-Clip-SP17-CFC-Shortcuts.mp4)

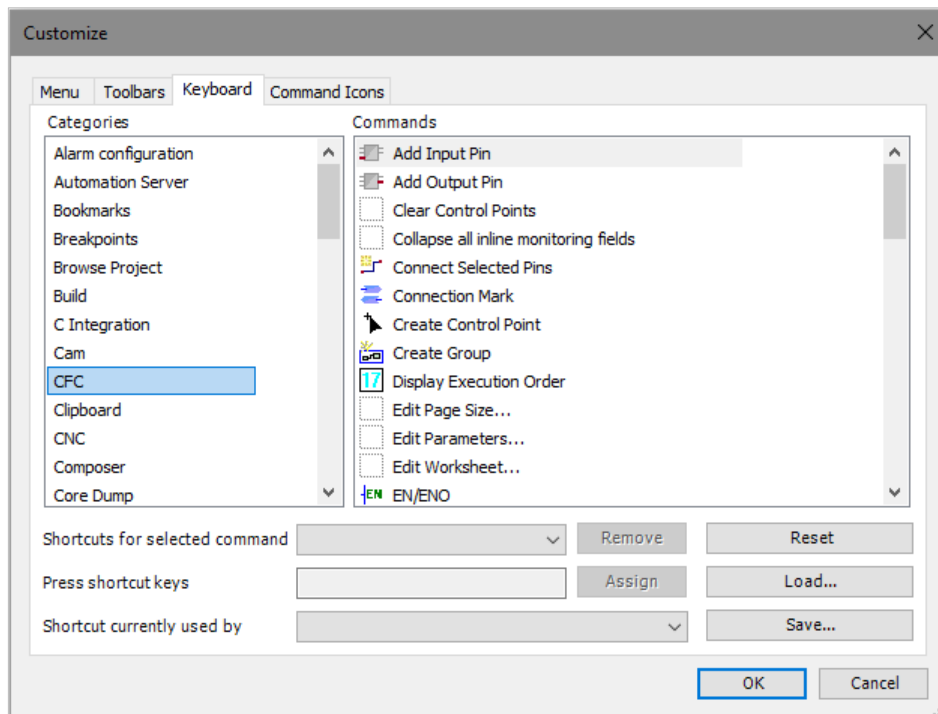


Рисунок 2.9 – Меню кастомізації гарячих клавіш для редактору CFC

## 2.7. Налаштування кількості відображуваних символів змінних в онлайн-режимі

Налаштування задаються в меню **Tolls – Options – CFC Editor – Monitoring**.  
(Інструменти - Опції - CFC-редактор – Monitoring)

Вони визначають, скільки символів для змінних із плаваючою крапкою і рядків буде відображатися в значеннях змінних на полотні в онлайн-режимі. Для перегляду повного значення потрібно навести на нього курсор для появи спливаючої підказки. Крім того, можна приховати значення, натиснувши на хрестик.

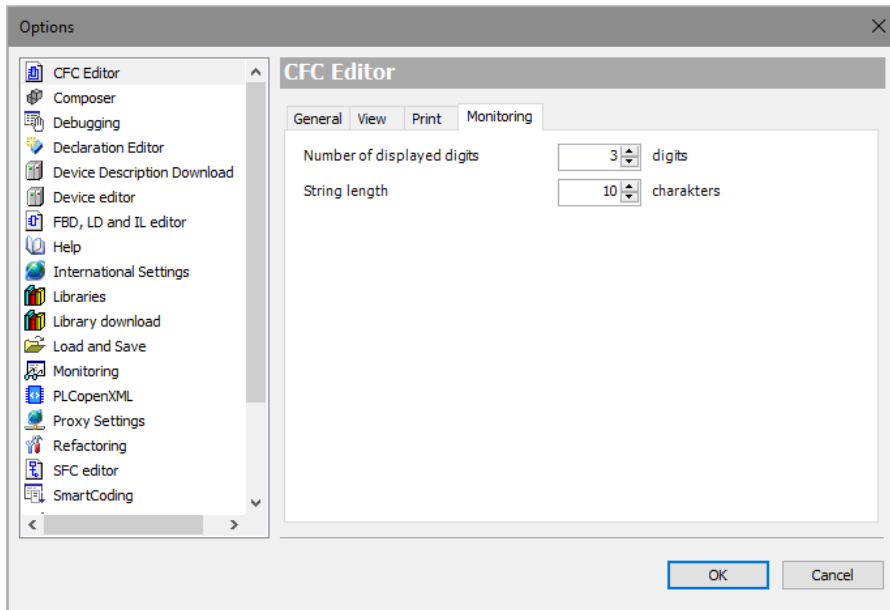


Рисунок 2.10 – Меню налаштувань відображення кількості символів при онлайн підключенні

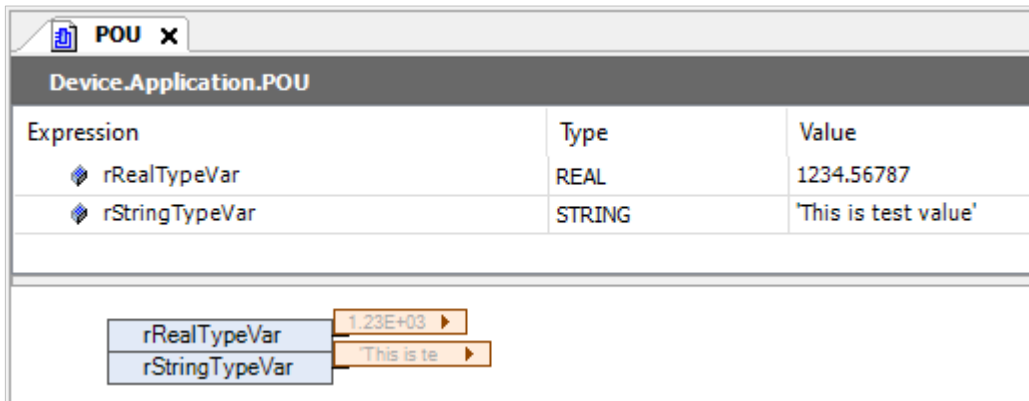


Рисунок 2.11 – Приклад відображення значення змінних після збереження налаштувань з Рис. 2.10



### 3. OPC UA

#### 3.1. Підтримка інформаційних моделей OPC UA і виклику методів ФБ з боку клієнта

На стороні CODESYS налаштування полягає в такому:

- **Tools - OPC UA Information Model Repository** (Інструменти – OPC UA Information Model Repository) - встановлення потрібної інформаційної моделі;
- ПКМ на вузол **Application – Add Object - Communication Manager** (Application – Додавання об'єкту – Communication Manager);
- ПКМ на **Communication Manager - Add Object - OPC UA Information Model** (ПКМ на Communication Manager – Додавання об'єкту – OPC UA Information Model);
- ПКМ на **OPC UA Information Model - Add Object - Information Model** (ПКМ на OPC UA Information Model - Додавання об'єкту – Information Model);
- створення об'єктів і налаштування моделі в доданих раніше компонентах.

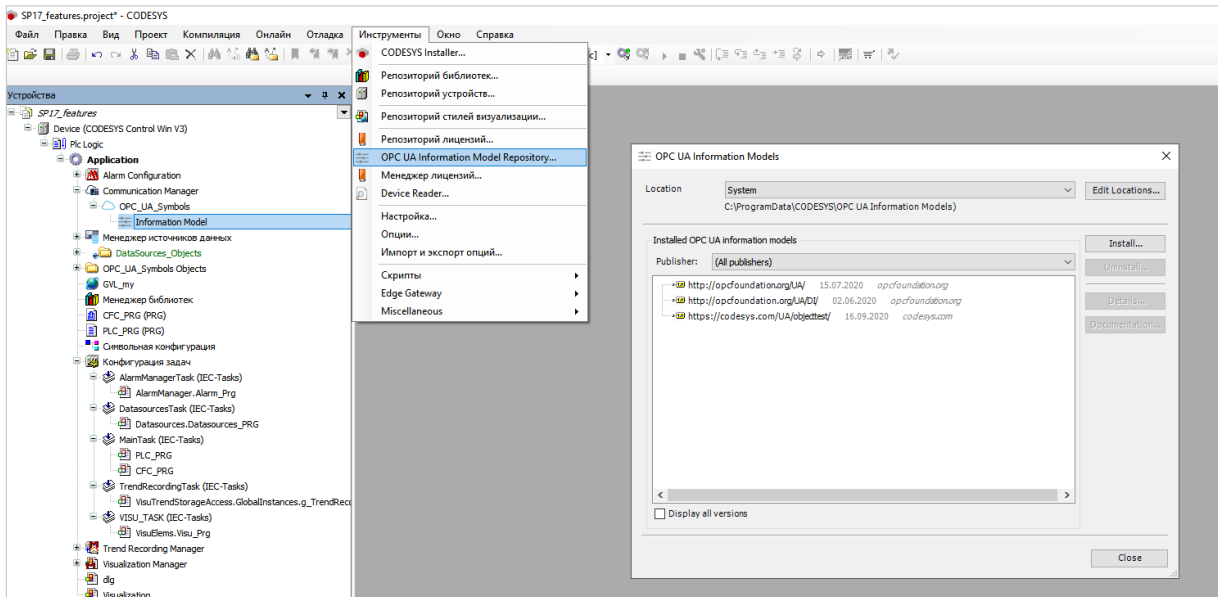


Рисунок 3.1 – Вікно встановлювання інформаційної моделі

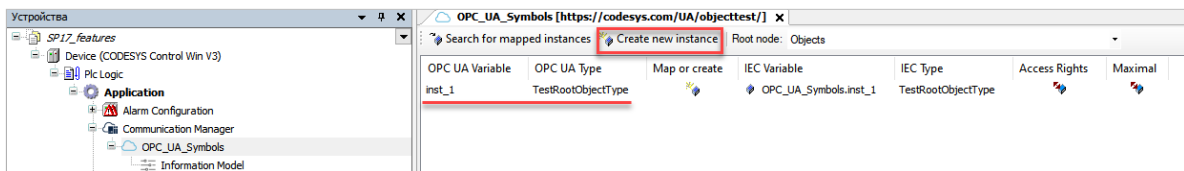


Рисунок 3.2 – Створення нового екземпляру

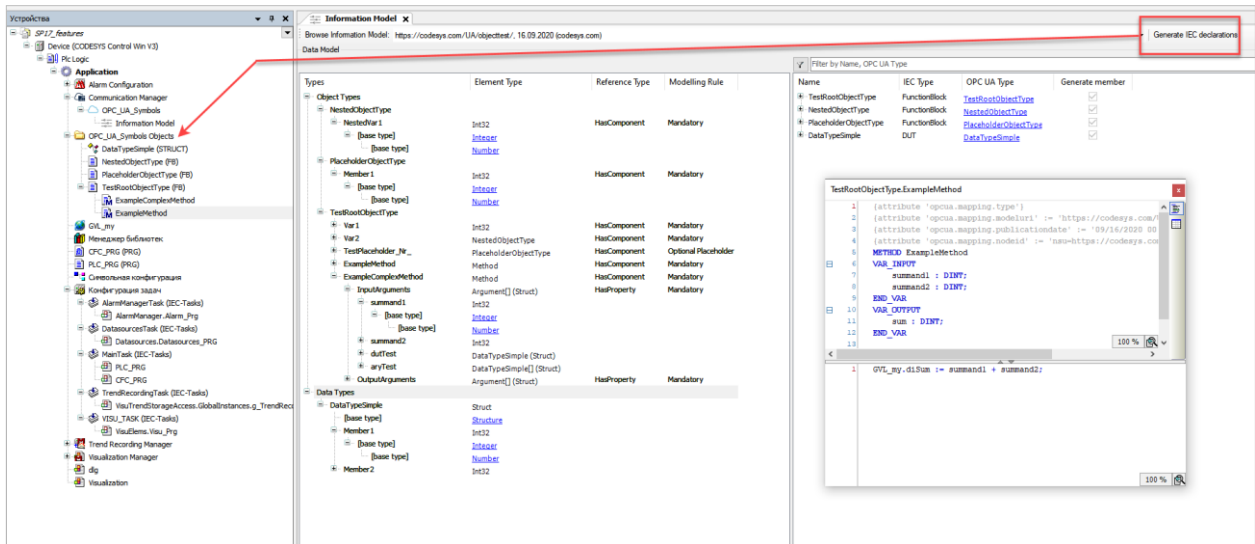


Рисунок 3.3 – Налаштування створених компонентів

### 3.2. Підтримка профілю OPC UA Alarms and Conditions

У CODESYS SP17 були додані такі можливості:

- передавання тривог із конфігурації тривог в OPC UA Client;
- додаткове налаштування не потрібне - достатньо додати в проєкт бібліотеку **StrmOPCUAProviderAlarmConfiguration** (див. опис, вбудований у бібліотеку).

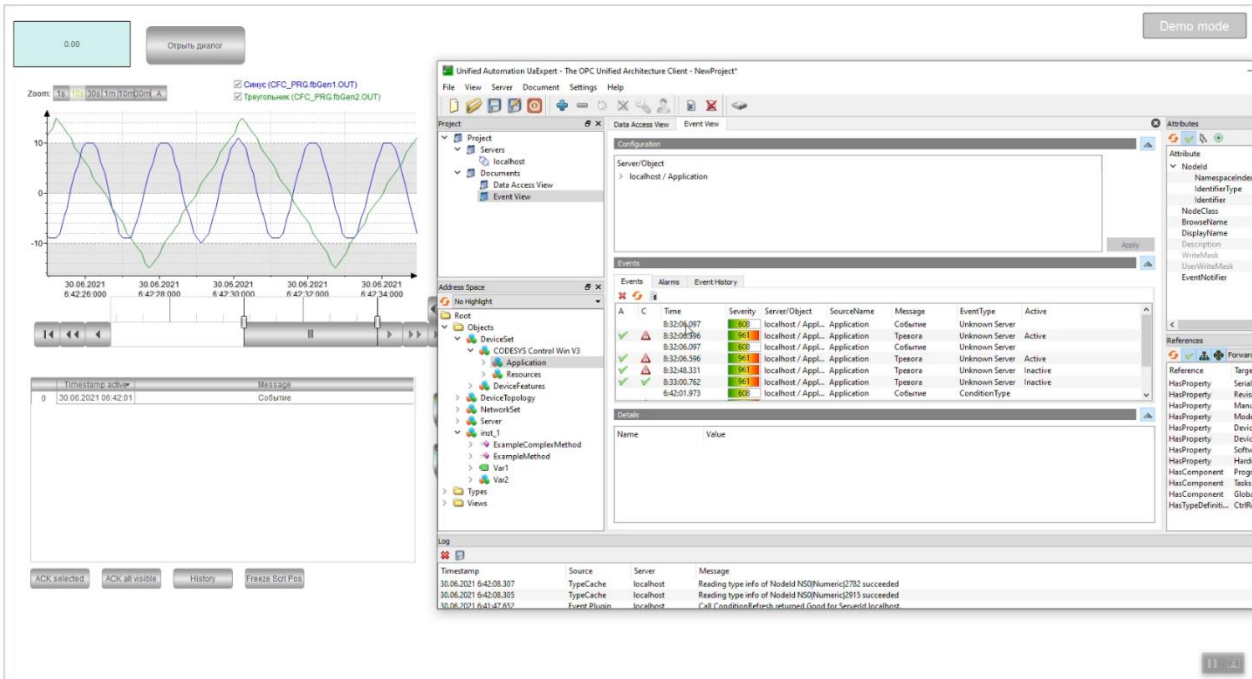


Рисунок 3.4 – Передавання тривог до OPC UA

### 3.3. Підтримка нових профілів безпеки зі специфікації OPC UA 1.04

Aes128Sha256RsaOaep;  
Aes256Sha256RsaPSS.

## 4. Середовище розробки

### 4.1. Імпорт проєктів із CoDeSys V2.3

Плагін для імпорту проєктів з CoDeSys V2.3 тепер не входить до дистрибутиву CODESYS, але може бути [завантажений з CODEYS Store](#). Зверніть увагу, що плагін доступний тільки для 32-бітної версії середовища розробки.

### 4.2. Підвищення швидкодії середовища

Проєкти тепер відкриваються на 15%-35% відсотків швидше. Завантаження проєкту в ПЛК займає менше часу.

### 4.3. Підтримка типів даних LDT/LDATE/LTOD

Типи даних **LDT/LDATE/LTOD** є 64-бітними версіями типів **DT/DATE/TOD** з розширеним діапазоном і підтримкою точності до наносекунд.

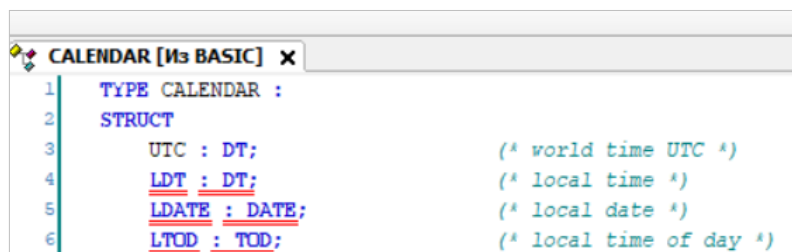
```
ldtDateAndTime:    LDT      := LDT#2021-05-14-09:15:10.123456789;
ltodTimeOfDay:    LTOD     := LTOD#12:34:56.123456789;
ldateDate:        LDATE    := LDATE#3000-05-12;
```

Рисунок 4.1 – Підтримка типів даних LDT/LDATE/LTOD

#### ПРИМІТКА



У бібліотеці *OSCAT Basic* позначення "LDT", "LDATE" і "LTOD" використовуються як імена змінних структури *CALENDAR*. Із введенням у *CODESYS* однойменних типів даних ці назви стали зарезервованими, тому спроба оголосити цю структуру (наприклад, вона використовується у ФБ *CALENDAR\_CALC*) призведе до помилок компіляції. Але оскільки бібліотека доступна у вихідних кодах - ви можете відкрити її, змінити імена цих змінних і перевстановити в репозиторій бібліотек для вирішення цієї проблеми. Крім того, [виправлену версію](#) вже виклали в *CODESYS Store*.



```
CALENDAR [Из BASIC] x
1  TYPE CALENDAR :
2  STRUCT
3      UTC : DT;           (* world time UTC *)
4      LDT : DT;           (* local time *)
5      LDATE : DATE;      (* local date *)
6      LTOD : TOD;        (* local time of day *)
```

Рисунок 4.2 – Помилка у бібліотеці OSCAT Basic

#### 4.4. Зменшення числа команд у меню Компіляція

Команди **Build**, **Rebuild** та **Generate runtime system files** (Компіляція, Перекомпіляція та Генерувати файли системи виконання) тепер відсутні (компіляція автоматично виконується під час генерації коду). У разі потреби їх можна повернути: **Tools – Customize – Menu - Build** (Інструменти - Налаштування - Меню – Компіляція).

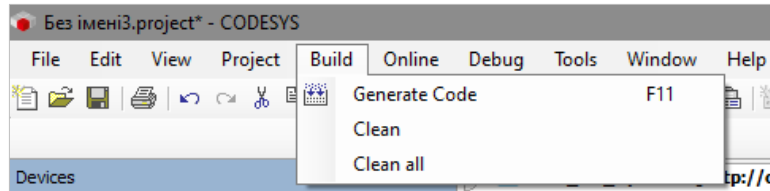


Рисунок 4.3 – Нове наповнення меню компіляції проєкту

#### 4.5. Опціональні аргументи для функцій і методів

Тепер якщо входи функцій і методів мають початкові значення, то їх можна не вказувати під час виклику (раніше було потрібне явне привласнення всіх змінних входам функції/методу під час виклику).

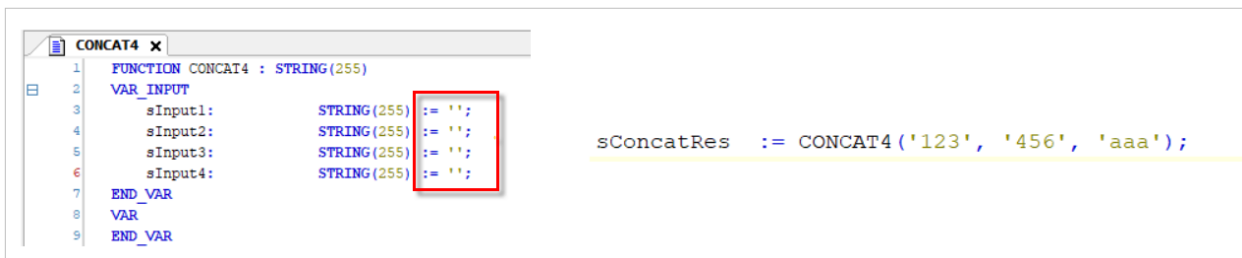


Рисунок 4.4 – Демонстрація можливості не вказувати усі аргументи функції при виклику

#### 4.6. Автоматичне створення стандартних методів для функціональних блоків.

Під час додавання методів тепер можна вибрати шаблон одного зі стандартних методів (**FB\_Init** або **FB\_Exit**).

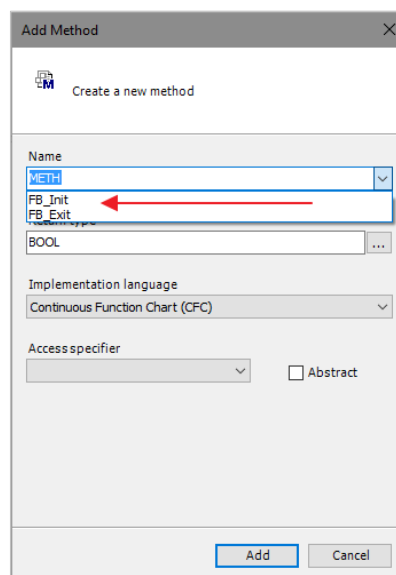


Рисунок 4.5 – Вікно додавання методів до ФБ з можливістю одразу вибрати один з двох стандартних

## 4.7. Оператори \_\_POUNAME і \_\_POSITION

Нові системні оператори **\_\_POUNAME** і **\_\_POSITION** дають змогу отримати ім'я POU (включно з усіма просторами імен) і номер його рядка. Це може стати в пригоді, наприклад, під час логування помилок програми.

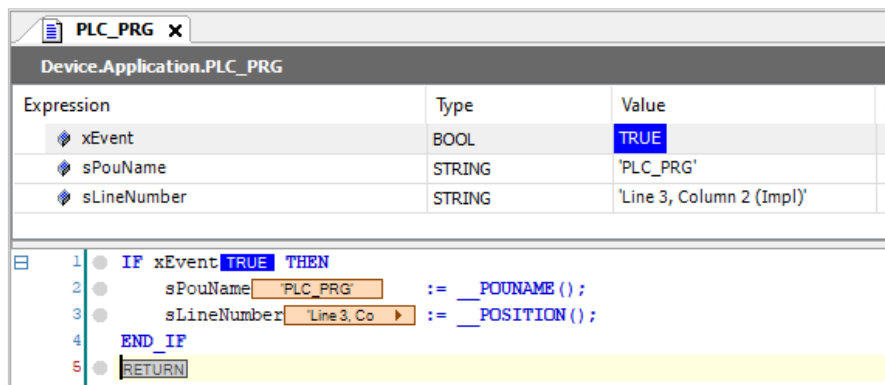


Рисунок 4.6 – Приклад використання нових операторів **\_\_POUNAME** і **\_\_POSITION**

## 4.8. Автооголошення змінних

Повернення "класичного" автооголошення змінних при використанні редактору мови ST

У минулих версіях CODESYS після введення в редакторі ST імені неіснуючої змінної і натискання на Enter з'являлося вікно автооголошення.

Починаючи з версії **CODESYS V3.5 SP14** поведінка змінилася: після введення неіснуючої змінної тепер потрібно було розмістити поруч із нею курсор, дочекатися появи "лампочки", натиснути на неї (або натиснути Ctrl+.), вибрати потрібну команду - і тільки після цього побачити вікно автооголошення.

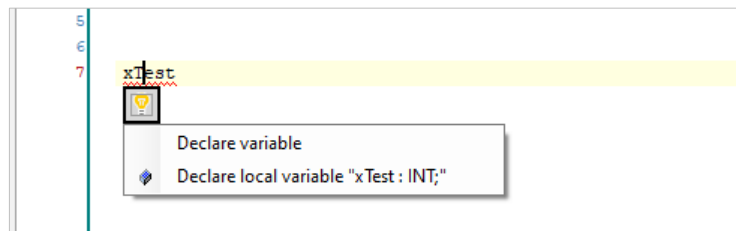


Рисунок 4.7 – Приклад поведінки автооголошення до SP17

На щастя багатьох користувачів, у наступних версіях з'явилася можливість увімкнути стару поведінку за допомогою галочки в меню **Tools – Options – SmartCoding** (Інструменти - Опції - Інтелектуальне введення).

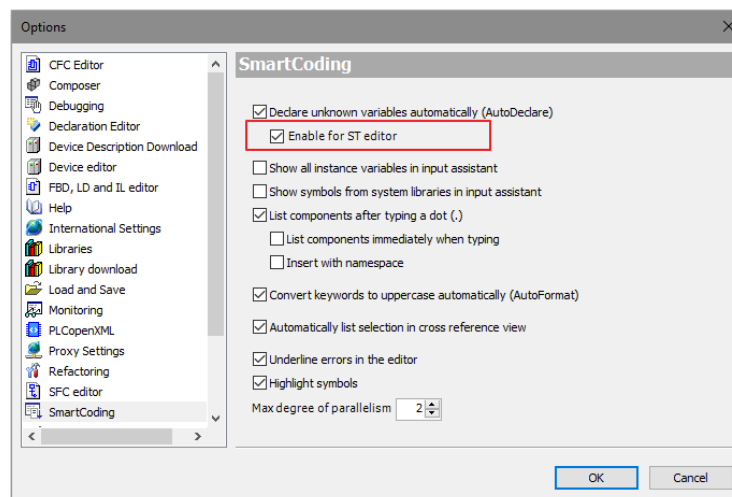


Рисунок 4.8 – Налаштування для увімкнення режиму автооголошення змінних для редактору ST

### 4.9. Вибір активної програми

Якщо в проєкті додано кілька контролерів - то потрібно перемикатися між їхніми застосунками для підключення до конкретного ПЛК. Раніше для цього потрібно було натиснути ПКМ на застосунок і використовувати команду **Set Active Application** (Встановити активний застосунок).

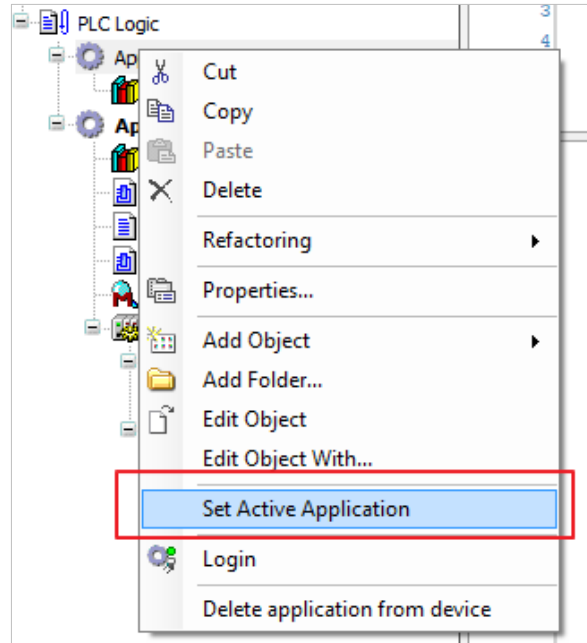


Рисунок 4.9 –Вибір активної програми для запису у ПЛК через контекстне меню

Тепер же це можна зробити за допомогою випадаючого списку на панелі інструментів.

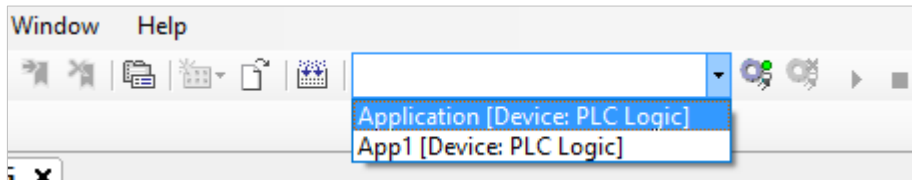


Рисунок 4.10 – Вибір активної програми для запису у ПЛК через панель інструментів

### 4.10. Відображення "реального" значення фіксованих змінних

Тепер у полі відображення всіх змінних ROU є стовбець в якому знаходиться фактичне значення перезаписаних користувачем змінних.

Expression	Type	Value	Prepar...	Overwritten value at start of cycle	Overwritten value at end of cycle
Device.Application.PLC_PRGj	DINT	55		55	1923
Device.Application.PLC_PRG.inst_dj	DINT	78		78	1923

Рисунок 4.11 – Відображення фактичного значення змінних

#### 4.11. Перехід до фрагментів коду з панелі точок зупинки

При подвійному натисканні на точку зупинки в панелі точок зупинки відбувається перехід до відповідного фрагмента коду.

Breakpoints							
Application: Application [Device: PLC Logic]							
POU	Location	Instance Path	Tasks	Condition	Hit Count Condition	Current Hit Count	Watched Values Last Updated
PLC_PRG	Line 1, Column 1 (Impl)	(any)	(any)	Break always	Break always	1	

Рисунок 4.12 – Панель керування контрольними точками (точками зупинки)

#### 4.12. Відображення заданого часу циклу завдання в конфігурації завдань

Тепер у Task Configuration відображається задане користувачем значення часу циклу завдання. Це зручно для оцінки завантаженості ресурсів контролера.

Task Configuration												
Task Groups	Monitor	Variable Usage	System Events	Properties	CPU Load							
Task	Status	IEC-Cycle Count	Cycle Count	Configured Cycle Period	Last Cycl...	Average ...	Max. Cycl...	Min. Cy...	Jitter (µs)	Min. Ji...	Max. Ji...	Core
MainTask	Valid	1470	1470	20 ms	3	3	38	2	-200	-1663	1653	1

Рисунок 4.13 – Відображення заданого значення часу для кожного завдання

#### 4.13. Покращення в інструментах аналізу пам'яті

У версії CODESYS SP17 додали такі нові функції роботи з пам'яттю:

- підтримка пошуку конкретного значення в пам'яті: **View – Open Memory Scan** (Вид - Open Memory Scan);
- підтримка копіювання адреси пам'яті в буфер обміну за допомогою контекстного меню або поєднання Ctrl+C;
- перевірка пам'яті: **Debug - Check Memory for Active Application** (Налагодження - Check Memory for Active Application) тепер виконується швидше.

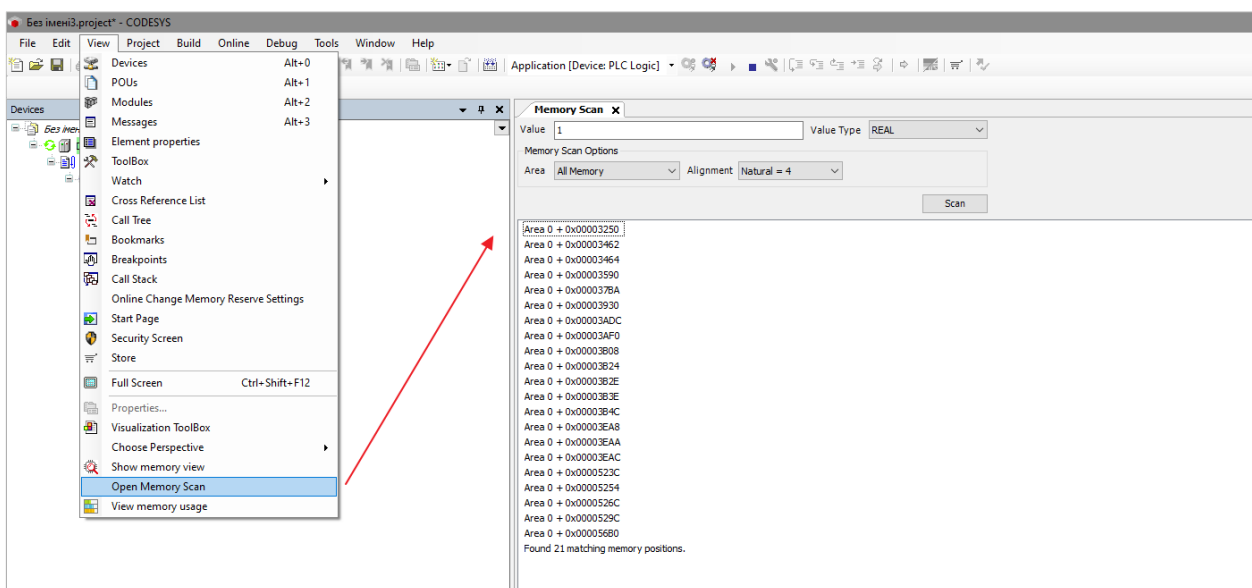


Рисунок 4.14 – Вікно Memory Scan

#### 4.14. Відображення дерева залежностей у менеджері бібліотек

Це вкрай корисний інструмент для розуміння які бібліотеки вже підключені до проекту. Дерево залежностей наочно демонструє, як одні бібліотеки залежать від інших.

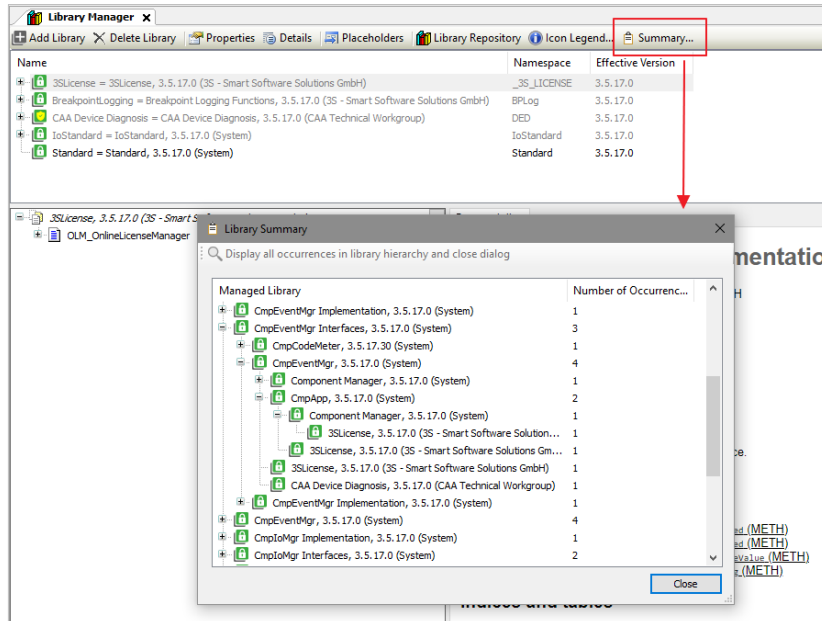


Рисунок 4.15 – Приклад відображення дерева залежностей бібліотек

#### 4.15. Покращення в компоненті Трасування

До компонента **Trace** (Трасуванні) додали:

- можливість зміни видимості діаграм/пір'їв в онлайн-режимі за допомогою відповідних прапорців (чекбоксів);
- можливість перетягування пір'я між діаграмами в онлайн-режимі, а також зміни порядку діаграм шляхом перетягування. У разі зміни порядку діаграм в онлайн-режимі він автоматично зміниться і в меню налаштування;
- у вікні додавання трасування користувач одразу вказує задачу (task), до якого воно буде прив'язане;
- у разі додавання кількох курсорів - відображається різниця між їхніми значеннями.

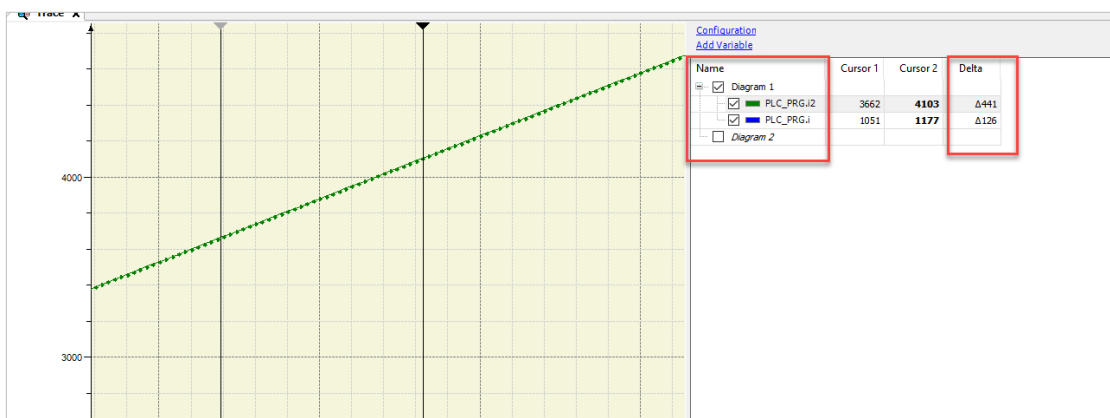


Рисунок 4.16 – Вікно компоненту Trace (Трасування)



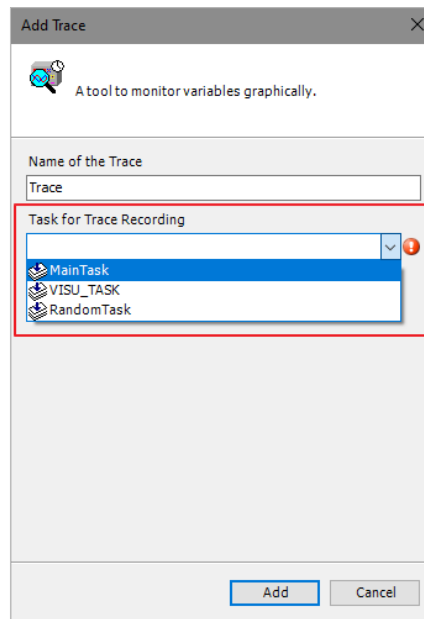


Рисунок 4.17 – Вікно створення нового об'єкту Trace з можливістю обрати завдання для нього

#### 4.16. Сортування файлів на вкладці Device - Файли

Тепер на вкладці **Device – Files** (Device – Файли) можна сортувати файли за ім'ям, розміром або датою зміни. Для цього потрібно натиснути на заголовок відповідного стовпчика.

#### 4.17. Завантаження відсутніх бібліотек через панель повідомлень

Тепер відсутні бібліотеки можна завантажити не тільки через Менеджер бібліотек, а й просто після натискання на відповідну помилку на панелі повідомлень.

## 5. Бібліотеки

### 5.1. Бібліотека Standard64.

У бібліотеці **Standard** є ФБ STU/CTD/CTUD, які використовуються для підрахунку імпульсів. Тип змінної, в якій зберігається результат підрахунку, - **WORD**, тобто лічильники здатні підрахувати тільки **65535** імпульсів, після чого відбудеться переповнення.

У свіжих версіях CODESYS у бібліотеку **Standard64** додано ФБ LCTU/LCTD/LCTUD - це версії стандартних ФБ лічильників з типом даних **LWORD**. Вони здатні підрахувати до  $(2^{64} - 1)$  імпульсів – вважаємо, для більшості завдань цього вистачить.

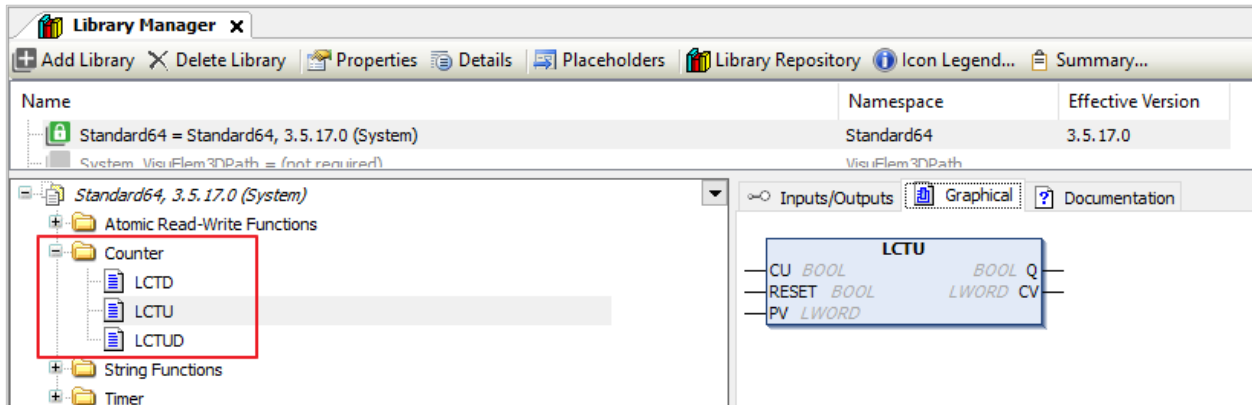


Рисунок 5.1 – Нові лічильники з більшим вихідним значення з бібліотеки Standard64

### 5.2. Бібліотека CAA File

До бібліотеки CAA File було додано два нових режима відкриття файлів: **MWRITEPLUS** і **MAPPENDPLUS**

- MWRITEPLUS - створює новий файл і відкриває його в режимі читання/запису. Якщо файл вже існує - його вміст видаляється;
- MAPPENDPLUS - файл відкривається в режимі дозапису (запису нових даних у кінець файлу). Якщо файл не існує - то його буде створено.

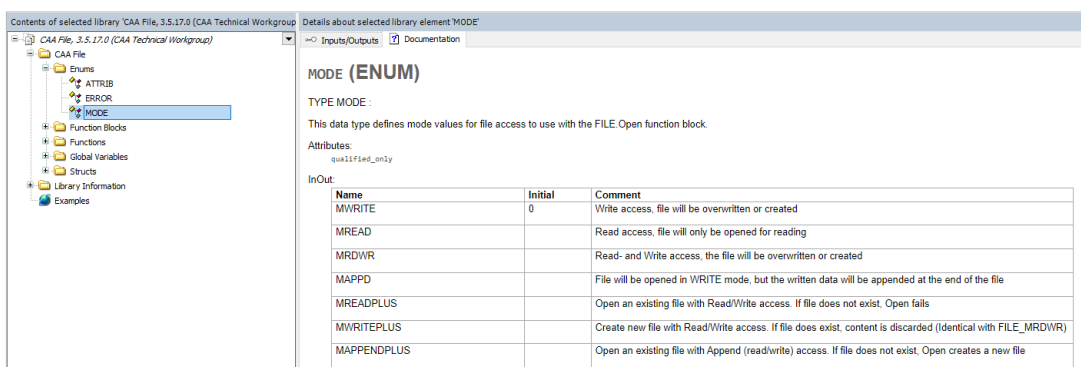


Рисунок 5.2 – Оновлене перерахування MODE бібліотеки CAA File

### 5.3. 2.Бібліотека Util

До бібліотеки Util було додано кілька нових функцій:

- GETBIT/SETBIT - аналоги EXTRACT/PACK з підтримкою типу LWORD;
- GetTimeStampsDifference - визначення різниці між двома мітками часами (у вигляді числа днів, годин, хвилин тощо).

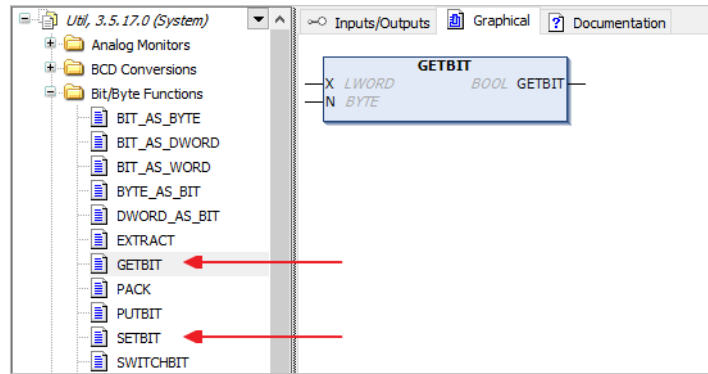


Рисунок 5.3 – Функції GETBIT та SETBIT бібліотеки Util

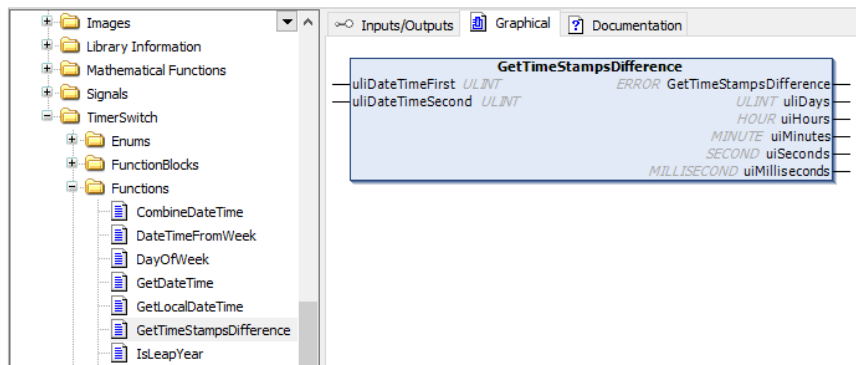


Рисунок 5.4 – Функція GetTimeStampDifference бібліотеки Util

## 5.4. Бібліотека Net Base Services

Нова бібліотека **Net Base Services** є більш просунутою версією бібліотеки **CAA Net Base Services** (використовуваної для обміну за протоколами TCP і UDP) з підтримкою криптографічного протоколу [TLS](#).

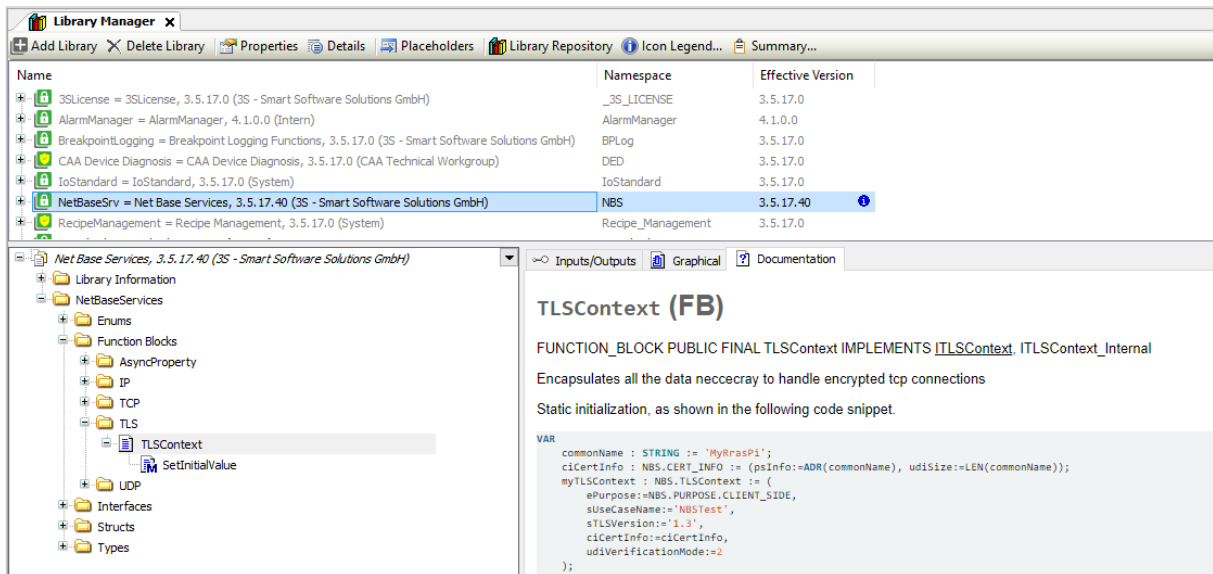


Рисунок 5.5 – Функціональний блок TLSContext бібліотеки Net Base Services

## 5.5. Бібліотека CmpCrypto

У бібліотеку **CmpCrypto** додано функції для асинхронного шифрування.

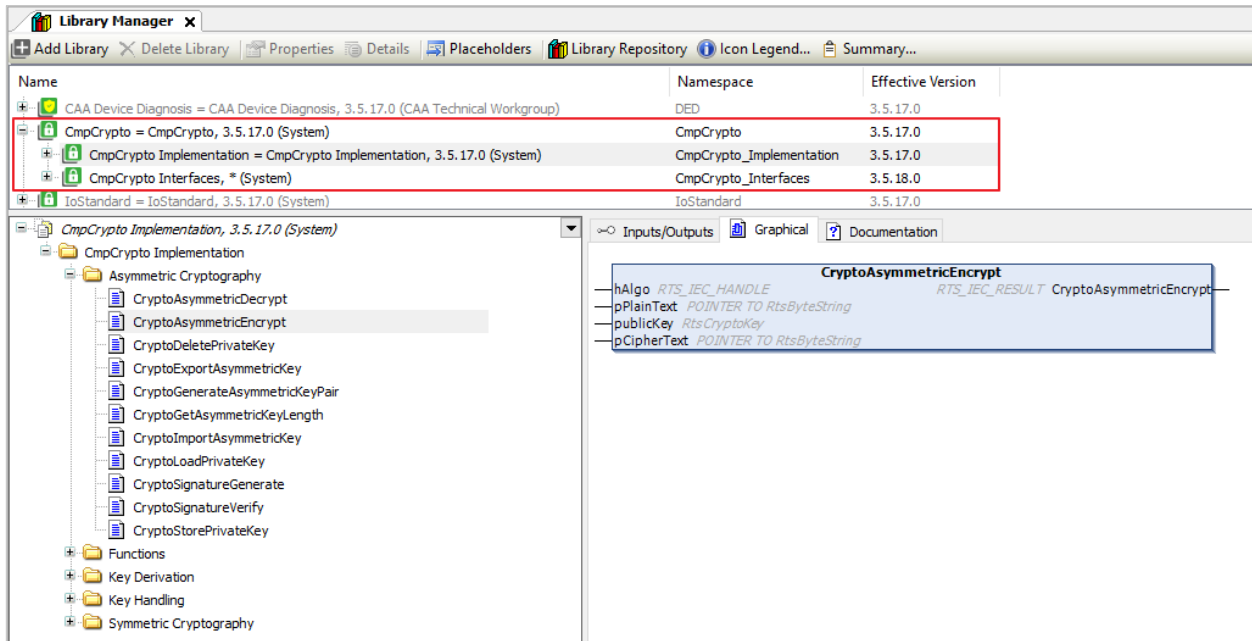


Рисунок 5.6 – Функції асинхронного шифрування бібліотеки CmpCrypto

## 5.6. Бібліотека CAA File

У бібліотеці CAA File зазнав виправлення ФБ DirList.

Тепер ФБ **DirList** повертає дату останньої зміни не тільки файлів, а й директорій (раніше для директорій завжди поверталось DT#1970-01-01-00:00:00).

## 6. Modbus

### 6.1. Запис значень у holding-реєстри з програми ПЛК для Modbus Slave Device (Serial/TCP)

Історично в CODESYS V3.5 у режимі Modbus Slave не можна було змінювати значення holding-реєстрів (і coils) з програми ПЛК - тільки з боку майстер-пристрою. У свіжих версіях ця можливість, нарешті, з'явилася - достатньо в налаштуваннях компонента поставити галочку **Writeable** (Запис).

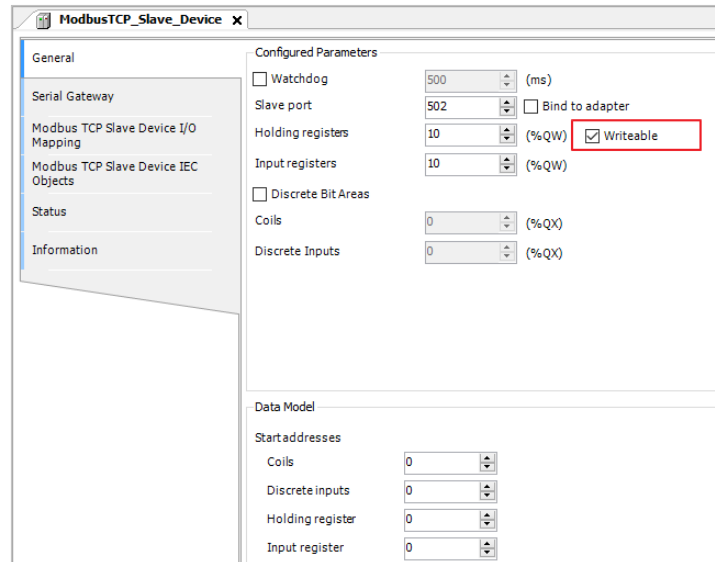


Рисунок 6.1 – Налаштування запису у holding-реєстри

#### ПРИМІТКА



*Під час активації цієї галочки значення holding-реєстрів не можна приєднувати в каналах компонента за допомогою команди **Debug – Write Values** (Налагодження - Запис значення) - тобто в режимі онлайн-налагодження вам потрібно буде змінити значення прив'язаної до каналу **змінної** у POU, де оголошено цю змінну, або через вкладку **View – Watch** (Вигляд - Перегляд).*

### 6.2. Незалежні області пам'яті для біт у Modbus Slave Device (Serial/TCP)

Історично в CODESYS V3.5 у режимі Modbus Slave область пам'яті Coils накладалася на область Holding Registers, а область Discrete Inputs - на область Input Registers. Тепер під час активації галочки **Discrete Bit Areas** (Дискретні бітові області) ці області стають незалежними і користувач може вказати їхній розмір у бітах. Тобто тепер coil з адресою 0 не буде відповідати нульовому біту holding-реєстру з адресою 0 - він буде розміщений у своїй області пам'яті.

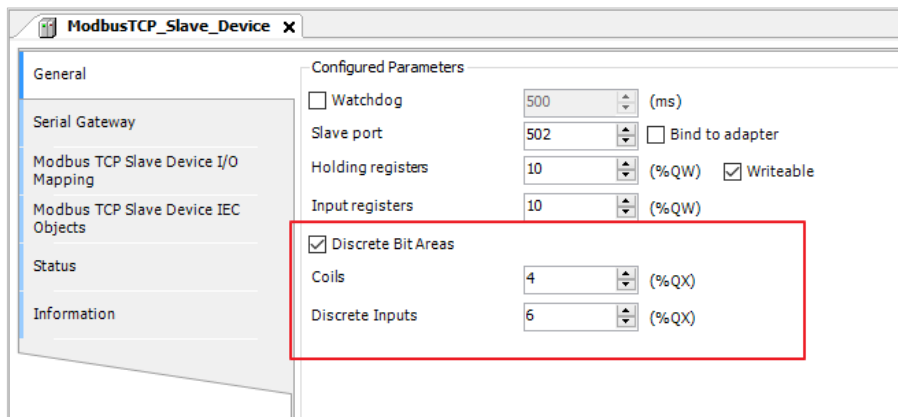


Рисунок 6.2 – Налаштування кількості Coils і Discrete Inputs у конфігурації Modbus Slave Device

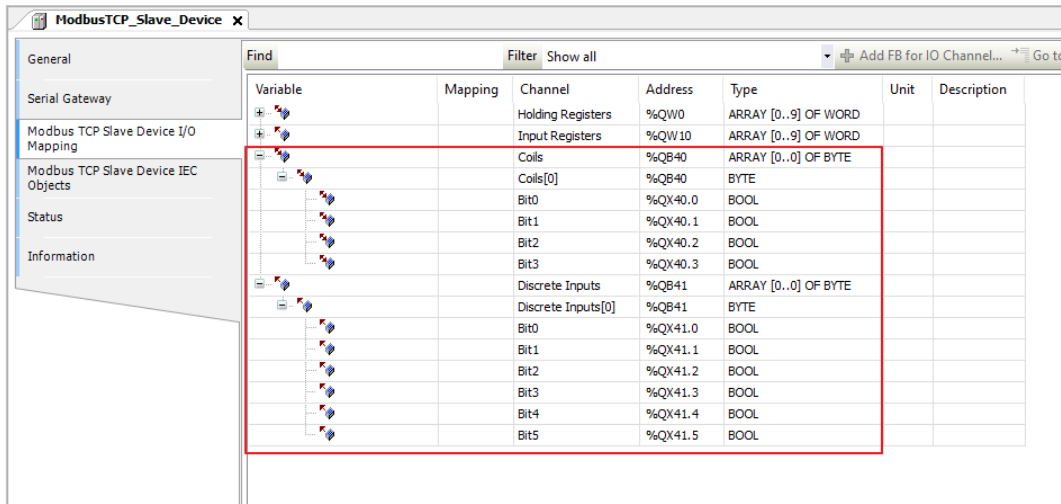


Рисунок 6.3 – Відображення доданих Coils та Discrete Inputs на вкладці I/O Mapping

### 6.3. Додаткові методи та змінні для Modbus Slave Device (Serial/TCP)

У компонента Modbus Slave Device додано методи **UpdateComPortSettings** і **UpdateUnitID**, які дають змогу змінити налаштування COM-порту та адресу слейва в процесі роботи ПЛК. Перед викликом методів необхідно призупинити роботу компонента за допомогою присвоєння його вхідній змінній **xEnable** значення **FALSE**. Після виклику методів компонент можна знову запустити в роботу, присвоївши змінній значення **TRUE**. Зверніть увагу, що під час перезавантаження контролера налаштування компонента знову будуть взяті з дерева проекту - це потрібно врахувати в логіці програми, організувавши в коді переналаштування компонента після завантаження контролера.



Рисунок 6.4 – Метод UpdateComPortSettings компонента Modbus Slave Device

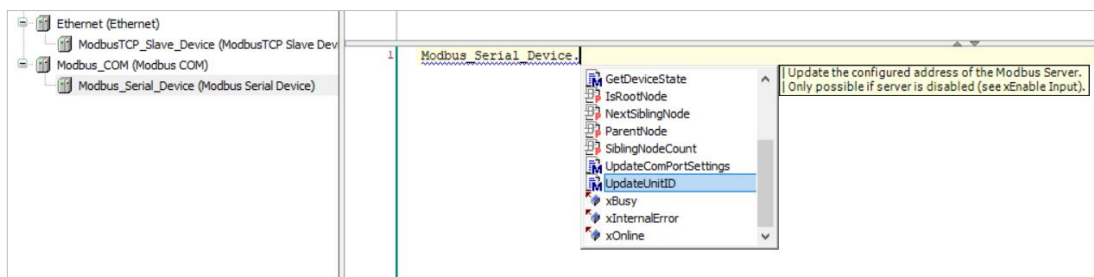


Рисунок 6.5 – Метод UpdateUnitID компонента Modbus Slave Device

В обох компонентів додано нові вихідні змінні:

- xOnline (BOOL) - прапор активності компонента (отримання запитів від майстра);
- ErrorCode (UDINT) - код помилки компонента;
- Port (UINT; тільки для компонента Modbus TCP Slave Device) - номер TCP-порту компонента;
- xTimeout (BOOL; тільки для компонента Modbus TCP Slave Device) - прапорець спрацьовування таймауту відсутності запитів від клієнта (у разі встановлення в налаштуваннях компонента відповідної галочки), у цей момент відбувається обнулення holding-регістрів і coils.

## 6.4. Можливість прив'язки до конкретного адаптера для Modbus TCP Slave Device

Раніше компонент Modbus TCP Slave Device завжди автоматично прив'язувався (bind) до інтерфейсу 0.0.0.0 (INADDR\_ANY) - тобто слейв був доступний за будь-яким з інтерфейсів ПЛК. Тепер у налаштуваннях компонента з'явилася галочка **Bind to adapter** (Прив'язати до адаптера). У разі її активації слейв прив'язується до інтерфейсу, який обрано в батьківському для слейва компоненті Ethernet.

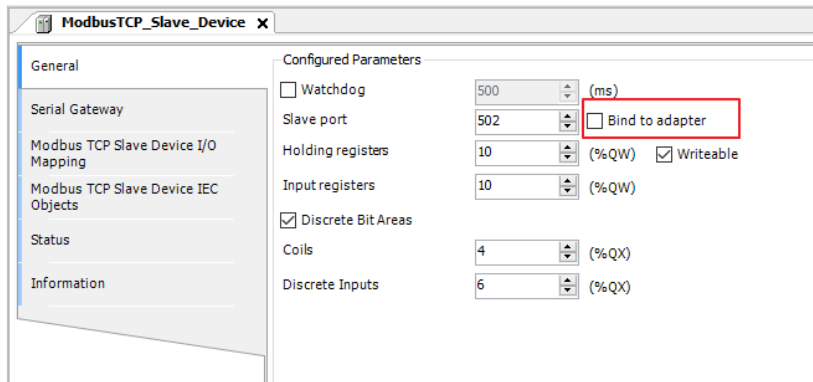


Рисунок 6.6 – Налаштування для прив'язки Modbus TCP Slave Device до адаптеру

## 6.5. Функціонал шлюзу RTU/TCP (Serial Gateway) для Modbus TCP Slave Device

Тепер компонент Modbus TCP Slave Device містить вкладку **Serial Gateway**, що дає змогу налаштувати ПЛК як шлюз Modbus RTU/Modbus TCP - запити, надіслані Modbus TCP Master'ом, будуть перетворені на запити Modbus RTU та надіслані в обраний на вкладці COM-порт (відповідно, цей COM-порт не повинен використовуватися в інших компонентах проекту). Відповіді від Modbus RTU Slave'ів будуть перетворені в Modbus TCP і відправлені назад майстер-пристрою.

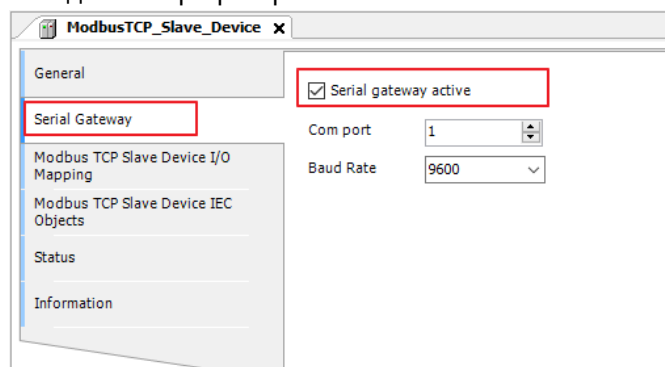


Рисунок 6.7 – Налаштування шлюзу Modbus TCP Slave Device

## 6.6. Збільшення допустимої кількості одночасно підключених клієнтів для Modbus TCP Slave Device

Раніше у слейва було обмеження на 32 одночасно під'єднаних клієнта, у свіжих версіях їхню кількість збільшено до 64.

## 6.7. Можливість додавання слейвів зі співпадаючими Slave ID у Modbus Slave COM Port

Раніше додавання кількох компонентів Modbus Slave COM Port зі співпадаючими Slave ID призводило до помилки компіляції, тепер - тільки до попередження. Це може виявитися корисним у тих випадках, коли потрібно створити для слейву понад 100 каналів опитування (таке обмеження встановлено на один компонент) або, наприклад, у разі використання шаблонів опитування, у яких бракує кількох потрібних параметрів - тепер можна додати їх у межах окремого компонента.

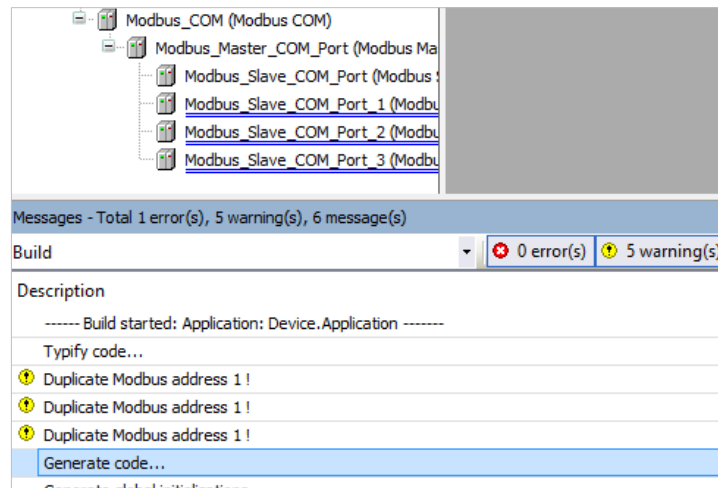


Рисунок 6.8 – Повідомлення про співпадаючі адреса Modbus Slave Device

## 6.8. Оптимізації в драйвері Modbus Master (Serial/TCP)

Проведено оптимізацію драйверів Modbus Master, яка дала змогу прискорити копіювання даних між каналами опитування і прив'язаними до них змінними.

## 6.9. Поліпшення в механізмі редагування каналів для Modbus Master (Serial/TCP)

Тепер після створення каналу опитування можна відредагувати його ім'я і змінювати порядок каналів у компоненті за допомогою кнопок **Move Up / Move Down** (Вище/Нижче).

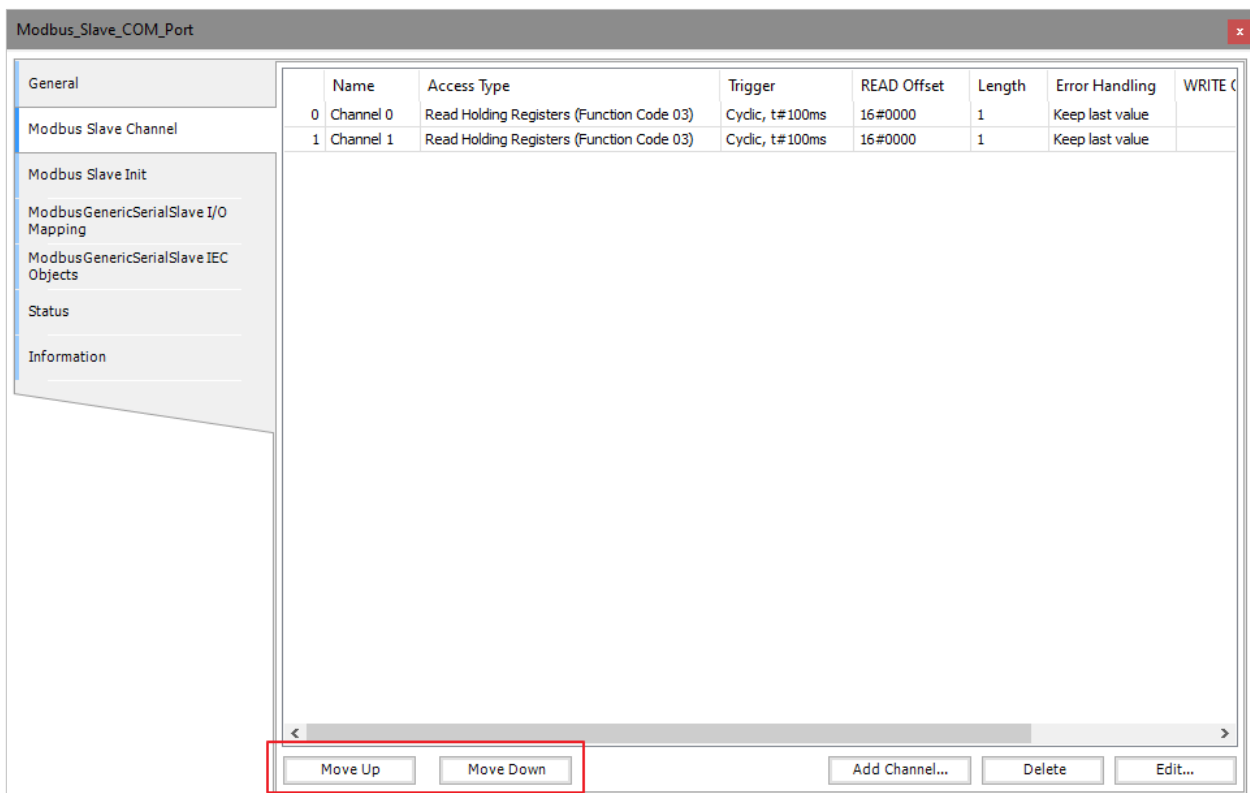


Рисунок 6.9 – Редагування порядку опитування каналів



## 6.10. Надсилання повторного запиту в разі відсутності відповіді у Modbus Serial Master

Для ігнорування одиничних помилок обміну тепер **Modbus Serial Master** у разі відсутності відповіді одноразово надсилає один повторний запит.

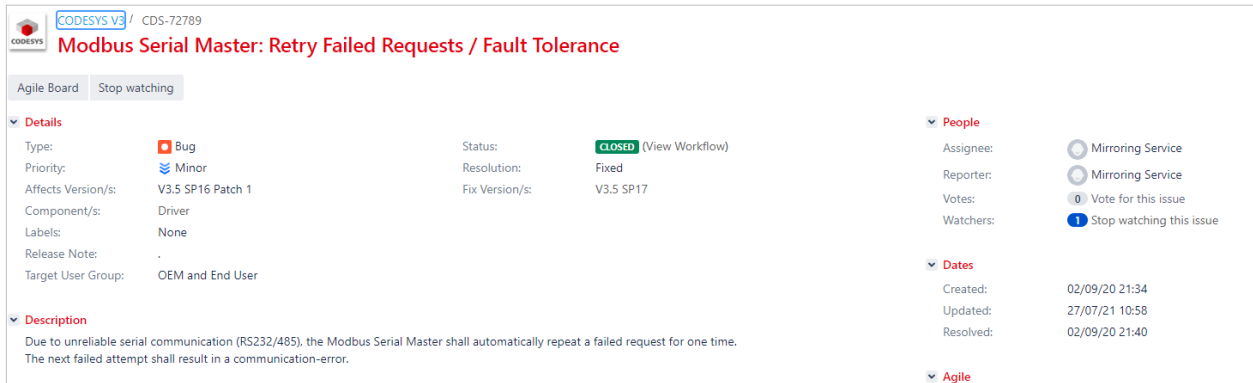


Рисунок 6.10 – Повідомлення у баг-трекері CODESYS

## 6.11. Змінено опрацювання каналів із налаштуванням "Встановити в нуль" у разі розриву TCP-з'єднання для Modbus TCP Master

У минулих версіях під час розриву TCP-з'єднання в 0 скидалося значення тільки того каналу, який оброблявся в момент розриву зв'язку. Тепер у цьому разі в 0 скидаються значення всіх каналів, для яких встановлено налаштування "Встановити в нуль".

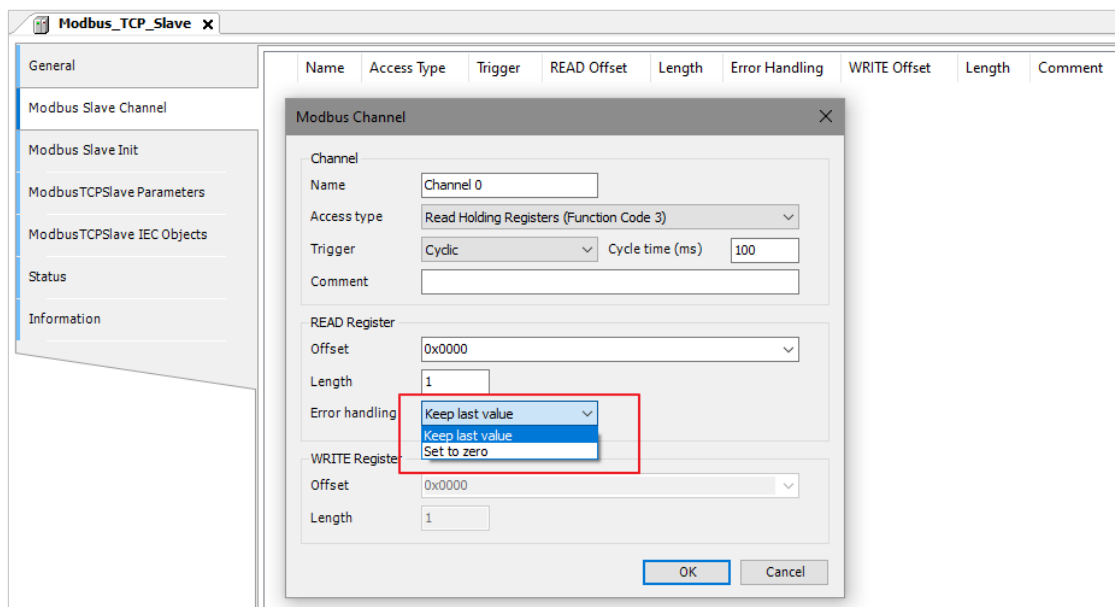


Рисунок 6.11 – Налаштування поведінки при розривах зв'язку

## 6.12. Поліпшення можливостей діагностики

Серйозно розширено функціонал діагностики всіх modbus-компонентів.

Тепер якщо в якомусь із згорнутих дочірніх компонентів сталася помилка - то в іконки батьківського елемента в правому нижньому кутку буде відображатися відповідна піктограма.



Рисунок 6.12 – Нова інфографіка помилок обміну

У компонента Modbus TCP Master з'явилася вкладка **Log** (Журнал), на якій відображається журнал підключень/втрат зв'язку зі slave-пристроями.

У компонентів Modbus TCP Slave і Modbus Slave COM Port на вкладці **Status** (Стан) тепер відображається інформація діагностики:

- стан з'єднання (для Modbus TCP Slave);
- лічильник запитів, надісланих слайву;
- лічильник відповідей із кодом помилки Modbus, отриманих від слейва;
- інформація про останню помилку - мітка часу, індекс каналу запиту (якщо помилка не пов'язана з каналом, то відображається -1) і код помилки (наприклад, ILLEGAL FUNCTION).

Крім того, тепер у разі виникнення і подальшого зникнення помилки на іконці елемента продовжує відображатися блідий знак оклику. Для його зникнення треба натиснути кнопку **Acknowledge** (Підтвердити).

У компоненті Modbus TCP Slave Device на вкладці **Status** (Стан) теж відображається інформація діагностики:

- кількість підключених клієнтів;
- статус TCP-порту;
- лічильник запитів, отриманих компонентом;
- лічильник відповідей із кодом помилки Modbus, які компонент надіслав майстру (на жаль, без конкретизації коду помилки);
- статус, лічильник запитів і лічильник помилок для Serial Gateway.

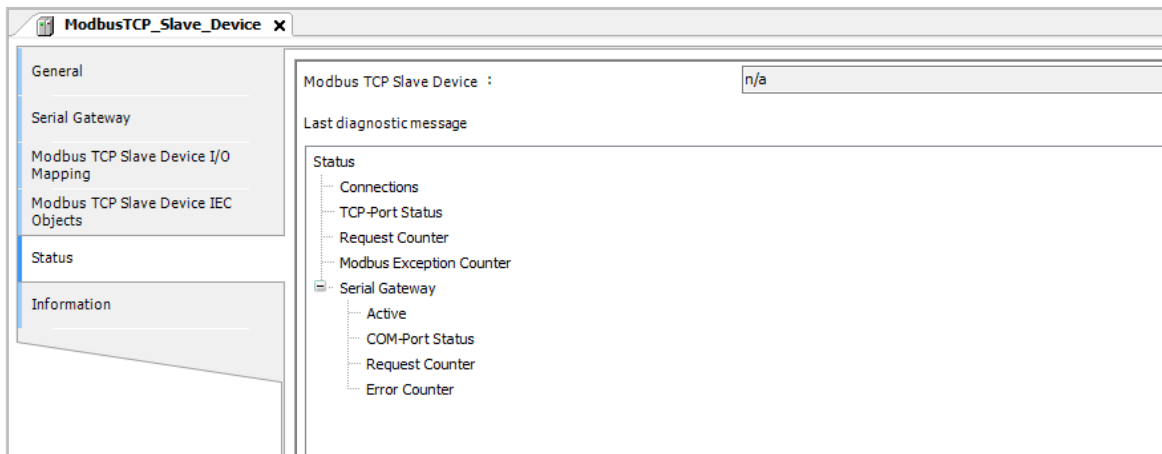


Рисунок 6.13 – Інформація на вкладці Status Modbus Slave Device

#### ПРИМІТКА



**У свіжих версіях CODESYS якщо до Modbus TCP Slave Device не під'єднано жодного клієнта (Connections = 0), то це відображається іконкою елемента (помаранчевим трикутником).**

### 6.13. Можливість групового оновлення версій компонентів

Тепер під час оновлення версії компонента (команда контекстного меню **Update Device** (Оновити пристрій)) можна оновити одразу всі компоненти проєкту до обраної версії за допомогою галочки **Update same device in project** (Оновити однакові пристрої в проєкті).

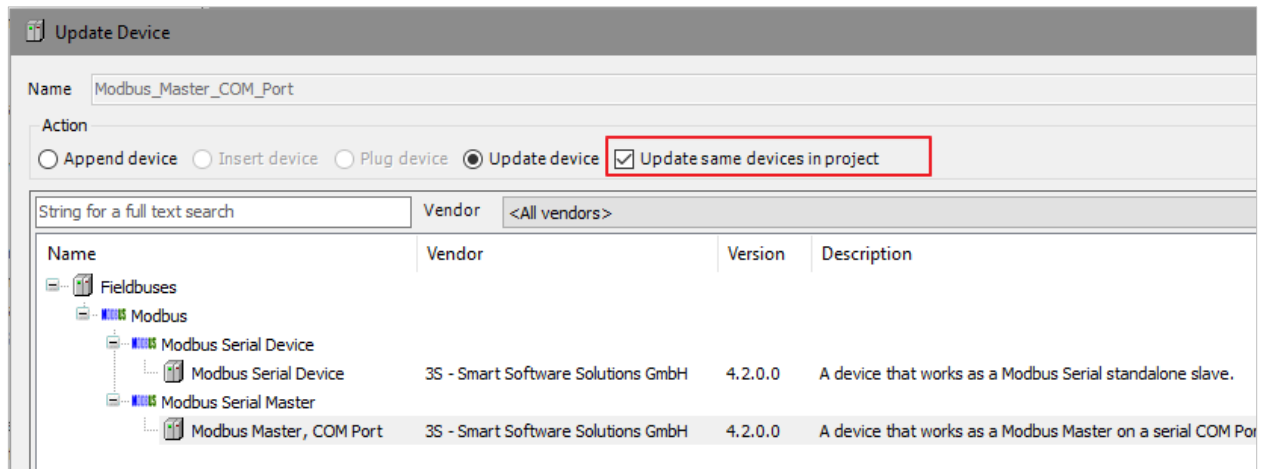


Рисунок 6.14 – Налаштування для оновлення усіх однакових пристроїв

## 7. Візуалізація

### 7.1. Підтримка технології overlay

Починаючи з V3.5 SP14 у CODESYS було додано preview-версію цієї технології, а повноцінна її підтримка з'явилася у V3.5 SP16. Для її активації слід встановити галочку в **Visualization Manager** (Менеджері візуалізації):

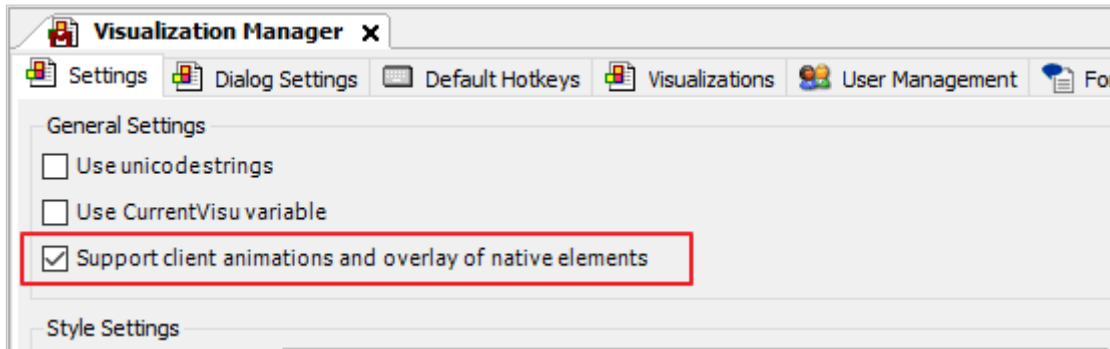


Рисунок 7.1 – Налаштування для активації режиму overlay

Поки що технологію підтримано тільки для web-візуалізації, і якщо в проєкті є таргет-візуалізація - то галочка не відобразиться. Підтримка технології overlay для таргет-візуалізації очікується в наступних версіях CODESYS.

Технологія змінює логіку відтворення візуалізації, представляючи кожен елемент як окремий об'єкт. Це, наприклад, дає змогу організувати накладення елементів і перемикання їх між шарами.

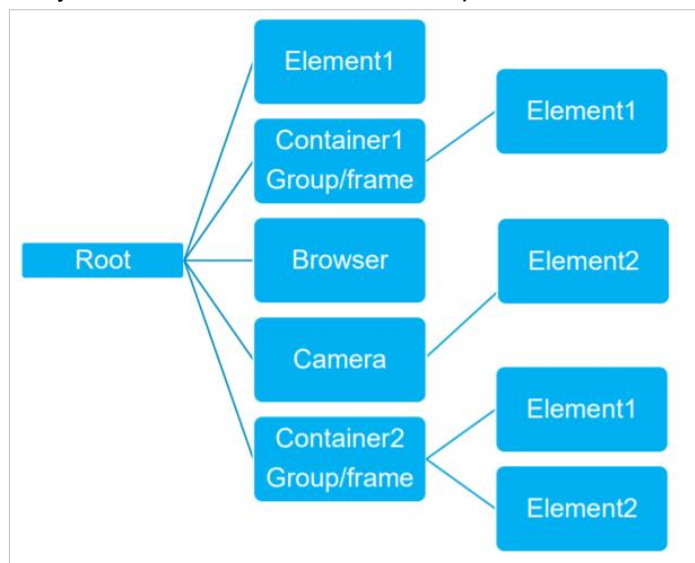


Рисунок 7.2 – Схема роботи технології overlay

Після активації галочки у кожного елемента візуалізації з'являються параметри **Animation duration** (Тривалість анімації) та **Bring to foreground** (Перемістити на передній план).

State variables	
Invisible	
Deactivate inputs	
Animation duration	0
Bring to foreground	

Рисунок 7.3 – Нові параметри при увімкненому режимі overlay

## Візуалізація

Параметр **Animation duration** (Тривалість анімації) має тип **INT** і визначає тривалість анімації в мілісекундах.

Анімація вмикається під час:

- переміщення елемента (за допомогою параметрів вкладки **Absolute movement** (Абсолютне переміщення));
- відкриття/закриття діалогу за допомогою цього елемента;
- зміни видимості елемента;
- перемикавання екрана у фреймі (якщо параметр задано для фрейму).

Тобто якщо раніше під час зміни параметрів вкладки **Absolute movement/Movement** (Абсолютне переміщення/Переміщення) елемент просто перемальовувався за новими координатами, то тепер він буде плавно переміщатися від старих координат до нових за заданий користувачем час. Цей самий ефект "плавності" з'являтиметься під час відкриття/закриття діалогів тощо.

Треба зазначити, що обробка анімації відбувається на стороні клієнта і ніяк не пов'язана з часом циклу завдання **VISU\_TASK**.

Параметр **Bring to foreground** (Перемістити на передній план) має тип **BOOL**. Якщо він набуває значення **TRUE**, то елемент перемальовується у верхньому шарі екрана (це буде помітно для елементів, накладених один на одного).

Також під час активації галочки вкладки **Absolute movement** (Абсолютне переміщення) з'являється в усіх елементів візуалізації - зокрема в графіків, таблиць тощо.

І останній функціонал, який надає overlay-технології - можливість відображення у візуалізації анімованих зображень у форматі '.gif' та '.svg'.

## 7.2. Порядок перемикавання елементів за допомогою клавіші Tab

Якщо в **Visualization Manager** (Менеджері візуалізації) встановити галочку **Standard keyboard handling** (Увімкнути стандартне використання клавіатури), то можна буде перемикати обраний для взаємодії елемент візуалізації за допомогою клавіші Tab.

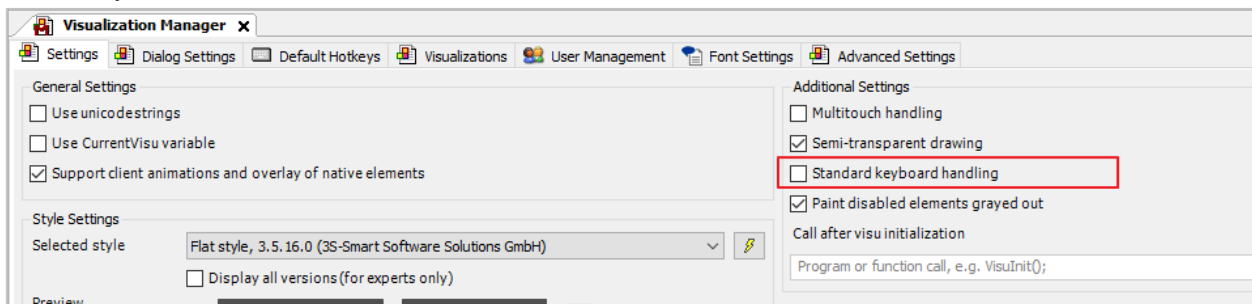
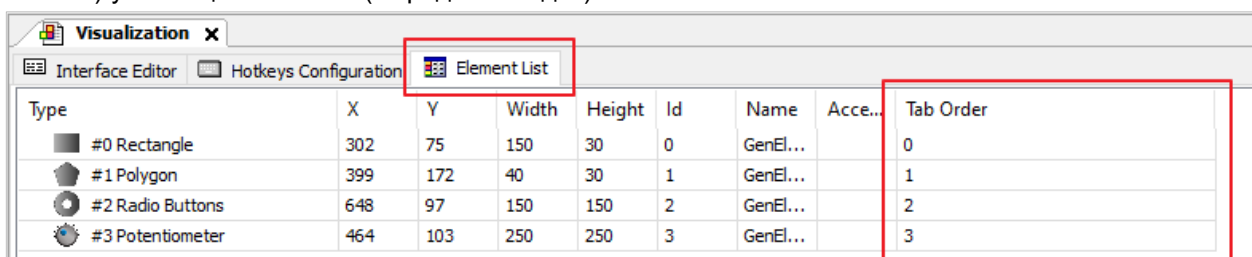


Рисунок 7.4 – Налаштування для стандартного використання клавіатури

Ця можливість з'явилася вже давно, але раніше порядок перемикавання елементів залежав від їхніх внутрішніх ідентифікаторів і, по суті, був довільним. У свіжих версіях CODESYS з'явилася можливість задати порядок перемикавання елементів у редакторі екрана візуалізації на вкладці **Element List** (Список елементів) у стовпці **Tab Order** (Порядок вкладок).



Type	X	Y	Width	Height	Id	Name	Acce...	Tab Order
#0 Rectangle	302	75	150	30	0	GenEl...		0
#1 Polygon	399	172	40	30	1	GenEl...		1
#2 Radio Buttons	648	97	150	150	2	GenEl...		2
#3 Potentiometer	464	103	250	250	3	GenEl...		3

Рисунок 7.5 – Відображення порядку перемикавання елементів візуалізації кнопкою Tab

### 7.3. Покращення в роботі з трендами

Версія CODESYS SP17 привнесла такі зміни у роботу з трендами:

- нове меню налаштувань, у якому всі пункти об'єднані на одній вкладці. Замість прив'язки до задачі (task) для тренда можна в налаштуваннях задати інтервал виклику;
- відображення переривання роботи тренда (наприклад, якщо ПЛК перебував у режимі Стоп або був вимкнений) як видимого розриву на графіку (раніше - остання точка попереднього запису і перша точка нового запису з'єднувалися лініями). За допомогою змінної **VisuTrendStorageAccess.GlobalInstances.g\_TimeFactorToDetectRecordingInterruption** користувач може вказати число часових зрізів тренду, після якого детектується переривання. Значення за замовчуванням - 3, значення 0 відповідає поведінці в минулих версіях;
- можливість вмикати/вимикати запис конкретних змінних тренда;
- можливість вмикати/вимикати видимість конкретних пір'їв тренду за допомогою чекбоксів, що відображаються в легенді тренду поруч із назвами пір'їв;
- можливість змінювати мінімум/максимум осі Y для тренда за допомогою змінних з коду програми;
- можливість зумування/панорамування тренда;
- можливість прокрутки/зумування за допомогою жестів.
- **Вимоги:** у **Visualization Manager** (Менеджері візуалізації) мають бути встановлені чекбокси **Support client animations and overlay of native elements** (Підтримка клієнтських анімацій і накладення вихідних елементів) і **Multitouch handling**;
- можливість очищення історії тренда з коду програми;
- покращено обробку пошкодження файлів тренда;
- зменшено споживання пам'яті;
- додано можливість зміни кольорів фону та осей;
- виправлено помилку - тепер у легенді тренда можна відображати назви пір'я кирилицею.

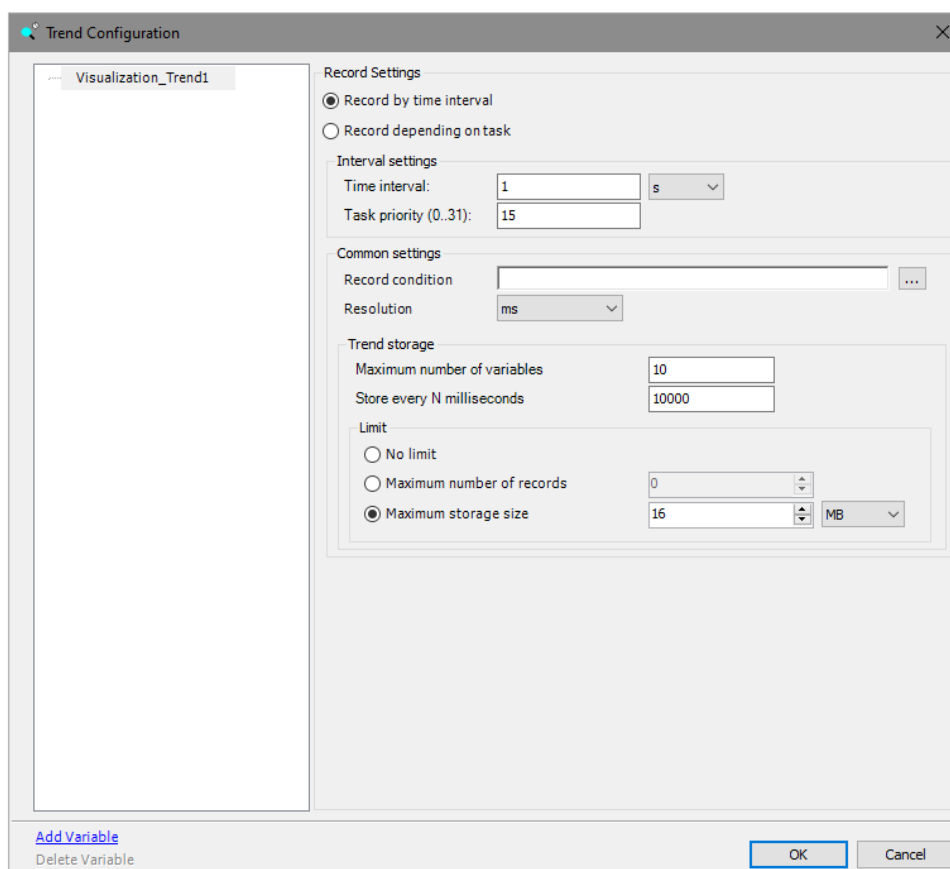


Рисунок 7.6 – Оновлене меню налаштування тренда



Рисунок 7.7 - Відображення переривання роботи тренда і чекбокси управління видимістю пір'я в легенді

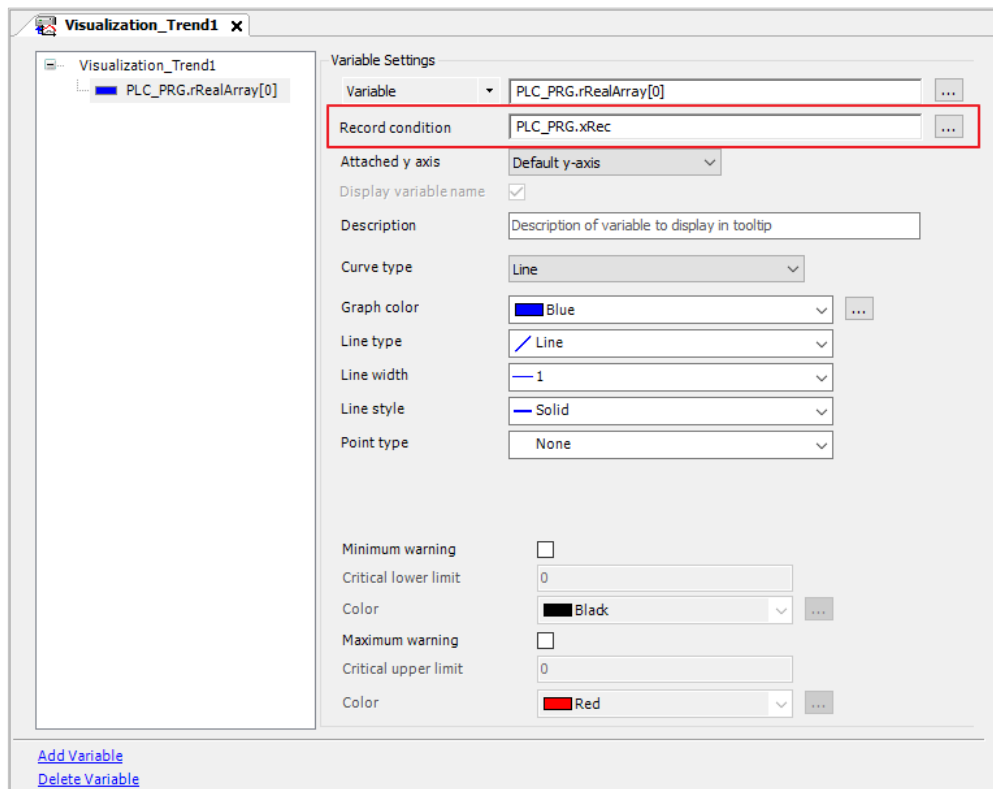


Рисунок 7.8 - Налаштування для керування записом конкретного піра

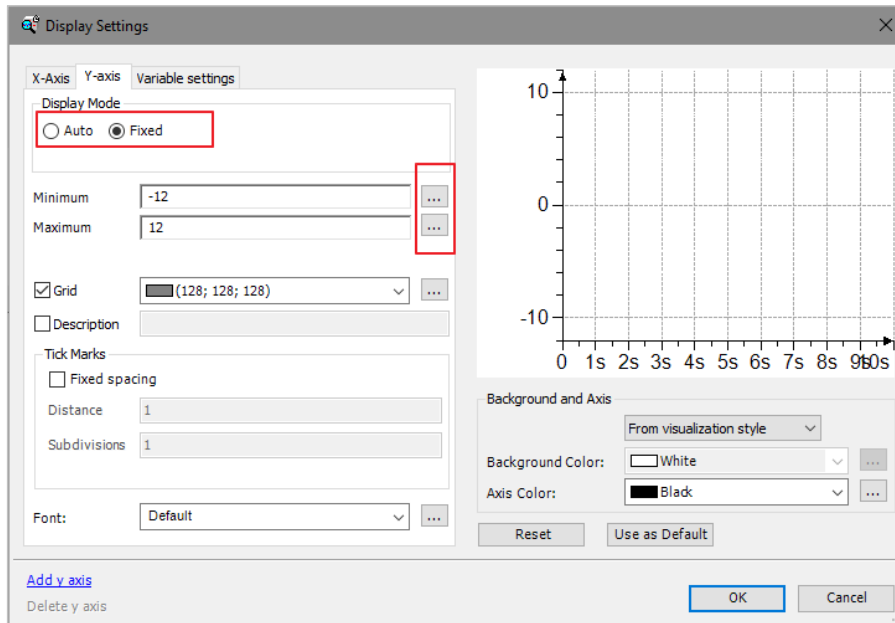


Рисунок 7.9 - Налаштування для керування діапазоном осі Y з коду програми

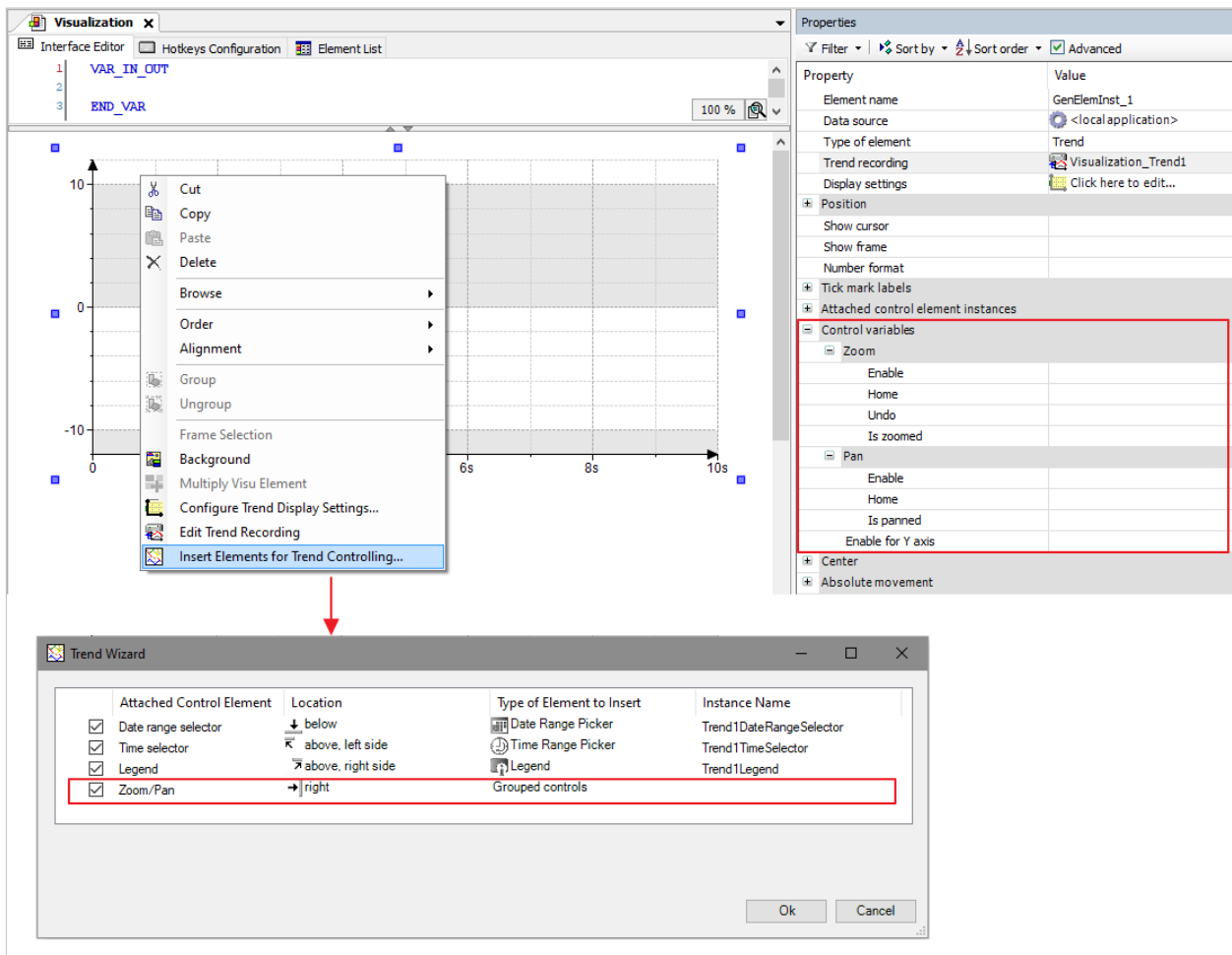


Рисунок 7.10- Додавання елементів для зумування/панорамування



Приклад очищення історії тренда з коду проекту:

```
// Очищення історії тренда з коду програми
VAR
  itfTrendRecording      : ITrendRecording;
  itfTrendStorageWriter : ITrendStorageWriter;
  itfTrendStorageWriter3 : ITrendStorageWriter3;
  // Назва запису тренда в дереві проекту
  sTrendRecordingName   : STRING := 'Visualization_Trend1';

  xClearTrendHistory   : BOOL;
ENV_VAR

IF xClearTrendHistory THEN

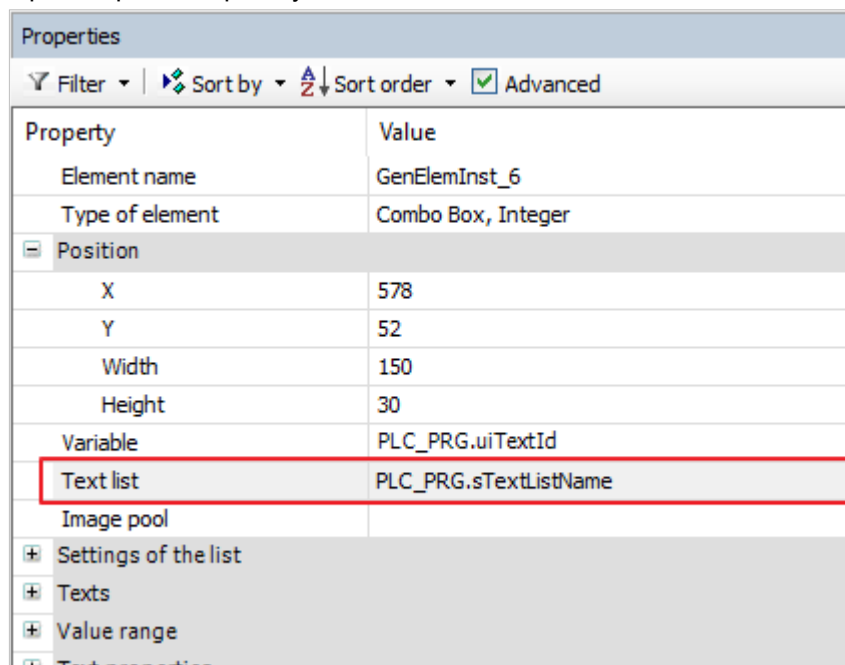
  itfTrendRecording := GlobalInstances.g_TrendRecordingManager.FindTrendRecording(ADR(sTrendRecordingName));
  IF itfTrendRecording <> 0 THEN
    itfTrendStorageWriter := itfTrendRecording.GetTrendStorageWriter();
    IF __QUERYINTERFACE(itfTrendStorageWriter, itfTrendStorageWriter3) THEN
      itfTrendStorageWriter3.ClearHistory();
    END_IF
  END_IF

  xClearTrendHistory := FALSE;

END_IF
```

#### 7.4. Вибір списку текстів у випадяючому списку за допомогою рядкової змінної

Тепер можна прив'язати до елемента Список, що випадає (Комбіноване вікно, ComboBox), не конкретний список текстів, а змінну типу **STRING**, що міститиме його назву - таким чином, можна перемикати списки текстів елемента в процесі роботи проекту.



Property	Value
Element name	GenElemInst_6
Type of element	Combo Box, Integer
Position	
X	578
Y	52
Width	150
Height	30
Variable	PLC_PRG.uiTextId
<b>Text list</b>	<b>PLC_PRG.sTextListName</b>
Image pool	
Settings of the list	
Texts	
Value range	
Text properties	

Рисунок 7.11 – Прив'язка рядкової змінної як імені списку текстів

## 7.5. Сортування повідомлень у таблиці та банері тривог

За допомогою нових параметрів таблиці та банера тривог можна сортувати повідомлення за часом появи або значенням першої latch-змінної. Детальнішу інформацію наведено в [довідці](#) та [прикладі](#).

Property	Value
Element name	GenElemInst_5
Type of element	Alarm Table
Data source	<local application>
Alarm configuration	
Alarm groups	All
Priority from	0
Priority to	255
Alarm classes	All
Filter by latch 1	
Filter variable	
Filter type	
Filter by time range	
Filter variable, from	
Filter variable, to	
Filter type	
General table configuration	

Рисунок 7.12 – Параметри для фільтрування повідомлень у таблиці тривог

## 7.6. Покращення в роботі з тривогами

Також у роботі з тривогами були додані наступні можливості:

- можливість очищення історії тривог із коду програми;
- можливість зчитування інформації за виділеною у візуалізації тривоною. Використовуйте ФБ **VisuElemsAlarm.AlarmSelectionInfoDefault** з бібліотеки **VisuElemsAlarm** - він дає змогу отримати інформацію за 10-ма останніми виділеними тривогами. Вихід **AlarmSelectionInfo** екземпляра цього ФБ треба прив'язати до параметра **Вибір/Variable for selected alarm information** елемента візуалізації **Таблиця тривог**.

Приклад очищення історії тривог із коду проекту:

```
// Очищення історії тривог із коду програми
VAR
  itfAlarmConfiguration : AlarmManager.IAlarmConfiguration;
  itfAlarmConfiguration7 : AlarmManager.IAlarmConfiguration7;
END_VAR

FOR i := 0 TO AlarmGlobals.g_AlarmHandler.AlarmConfigurationCount - 1 DO
  itfAlarmConfiguration := AlarmGlobals.g_AlarmHandler.GetAlarmConfiguration(i);
  IF __QUERYINTERFACE(itfAlarmConfiguration, itfAlarmConfiguration7) THEN
    // Зберігаємо до журналу поточні тривоги
    itfAlarmConfiguration7.ClearHistory(xKeepActiveAlarms := TRUE);
  END_IF
END_FOR
```

```

PLC_PRG x
1  PROGRAM PLC_PRG
2  VAR
3  fbAlarmInfo : VisuelemsAlarm.AlarmSelectionInfoDefault;
4  END_VAR
5
    
```

Рисунок 7.13 – Приклад оголошення ФБ для отримання інформації про виділені тривоги

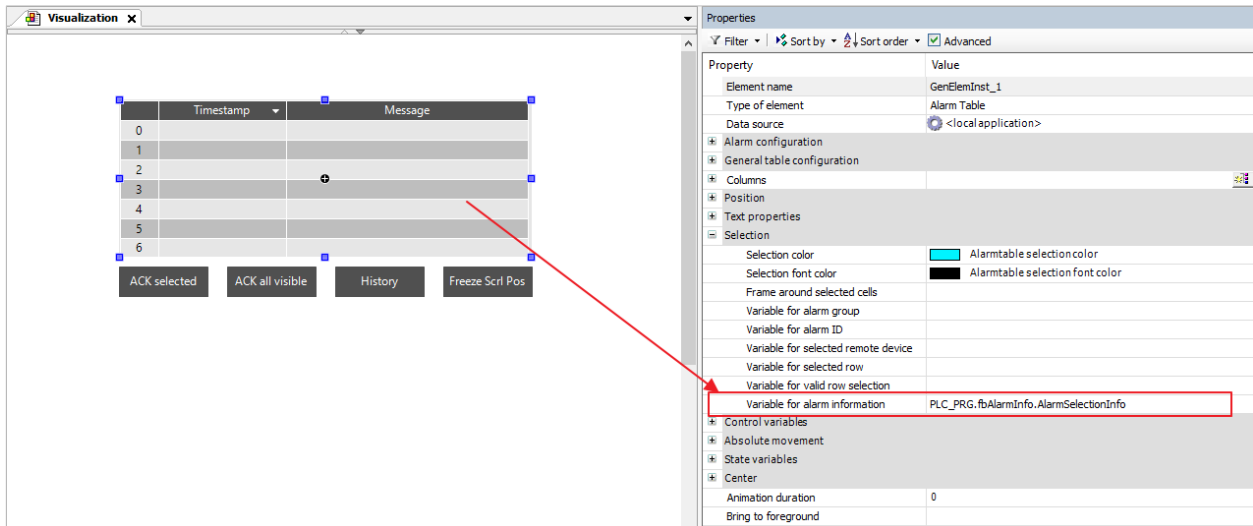


Рисунок 7.14 – Поле прив'язки ФБ отримання інформації про виділені тривоги

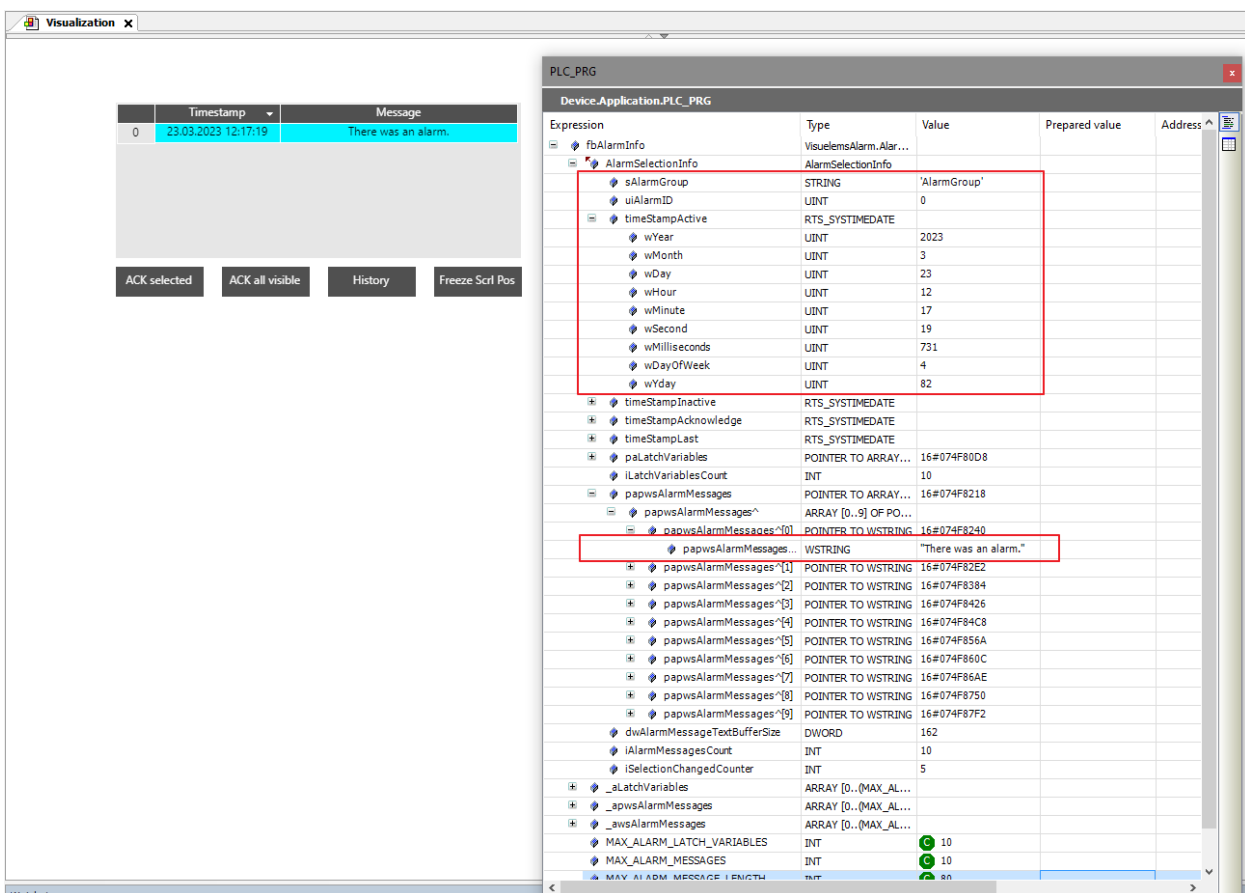


Рисунок 7.15 – Приклад отриманих даних про виділену тривогу з таблиці тривог

## 7.7. Підтримка змінних типу LDATE/LTOD/L DT

В елементах візуалізації, тривогах і рецептах тепер можна використовувати змінні типів **LDATE/LTOD/LTD**.

## 7.8. ФБ для закриття діалогів з коду програми

У бібліотеку **Visu Utils** додано ФБ **FbCloseDialog**, який дає змогу закривати діалоги з коду програми.

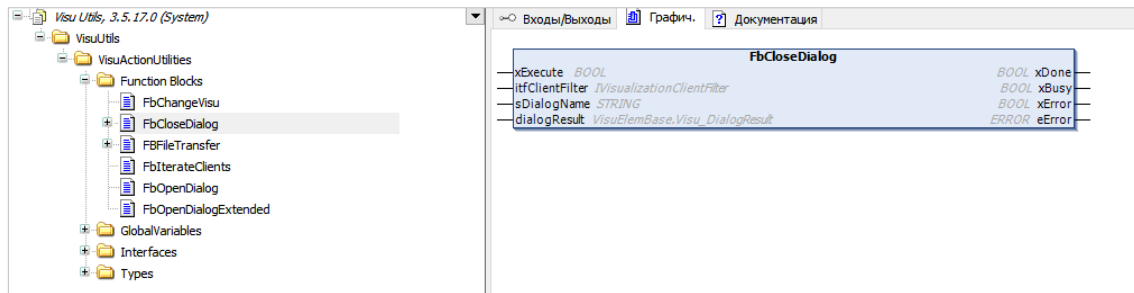


Рисунок 7.16 – Функціональний блок FbCloseDialog бібліотеки Visu Utils

## 7.9. Фонові зображення

У **Менеджері візуалізації** з'явилася нова вкладка - **Advanced Settings**. На неї перенесено налаштування, які раніше розташовувалися на вкладці **Установки** в групі параметрів **Advanced**, а також нову групу налаштувань **Background Configuration**. Вона дає змогу вибрати зображення з пулу зображень, яке відобразиться як фон на всіх екранах проекту. За допомогою координат можна обмежити робочу область (content area), у якій можна розташовувати елементи візуалізації, щодо цього зображення. Див. скріншот - зображення має розмір 800x480, налаштування content area - 100/100/500/500/500. У результаті:

- зображення зміщене щодо робочого поля. Верхня ліва точка зображення в редакторі фактично має координати (-100, -100);
- робоча область, у якій можна розміщувати елементи, представлена на скріншоті білим прямокутником (тільки для наочності) і має розміри 400x400 (500 - 100 = 400).

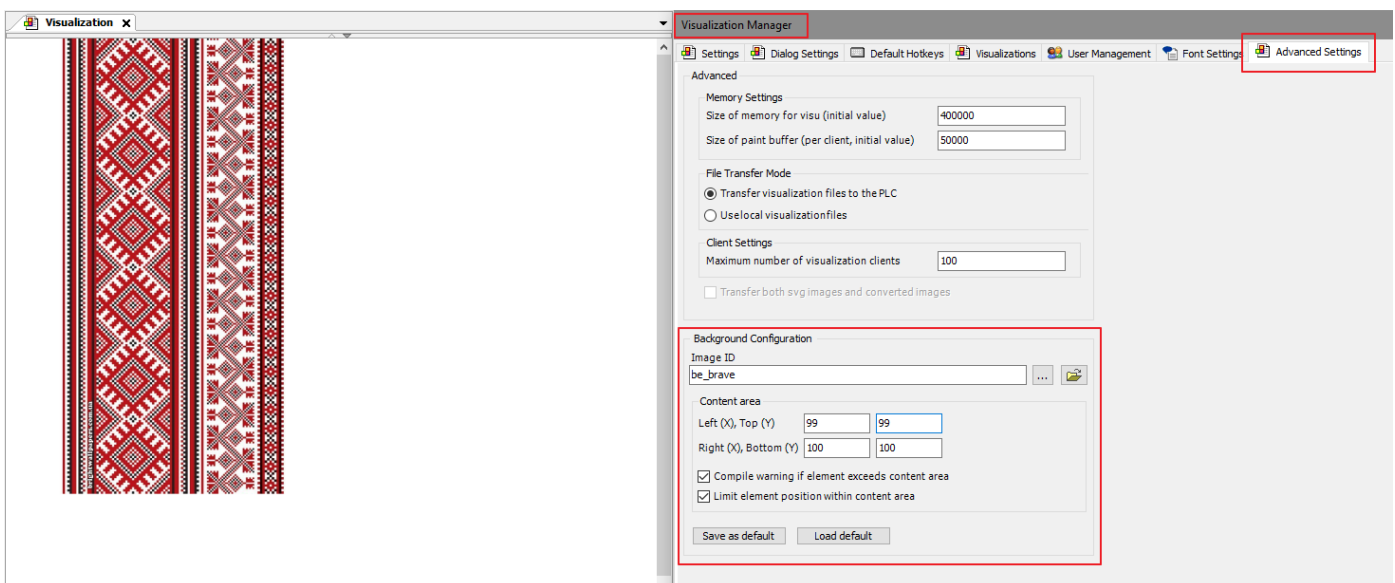


Рисунок 7.17 – Приклад налаштувань фонового зображення

## Візуалізація

Налаштування **Compile warning if element exceeds content area** вмикає генерацію попереджень компілятора для тих випадків, коли елементи візуалізації виходять за межі робочої області.

Налаштування **Limit element position within content area** забороняє розміщення елементів за межами робочої області (редактор не дасть змоги користувачеві це зробити за допомогою невидимих меж під час розміщення елементів).

Кнопки **Save as default** і **Load Default** дають змогу зберегти/завантажити налаштування за замовчуванням для цієї групи налаштувань.

Нова команда контекстного меню редактора візуалізації **Move into content area** автоматично переносить усі елементи екрана, розташовані за межами робочої області, у робочу область.

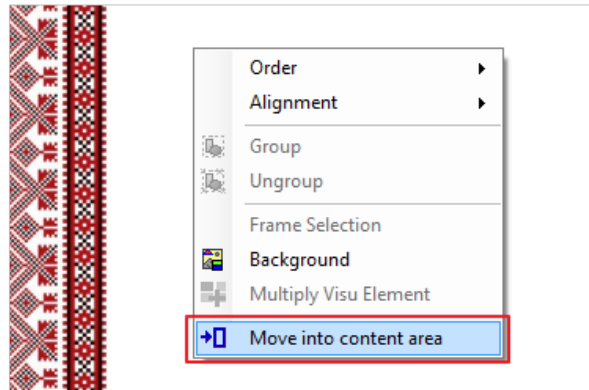


Рисунок 7.18 – Команда Move into content area контекстного меню

## 7.10. Покращення у відображенні системного часу

З появою плагіна візуалізації 4.1.0.0. з'явилися наступні можливості роботи з системним часом:

- можливість використовувати в специфікаторах формату часу часовий пояс (Z). Приклад: **%t[dd:MM:yyyy HH:mm:ss:ms:Z]**. Специфікатор можна використовувати в:
  - елементах візуалізації, що відображають системний час (тобто у яких у параметрі **Texts** (Тексти) вказані специфікатори формату часу, але до яких не прив'язано текстової змінної);
  - тренді (параметр **Internationalization** (Інтернаціоналізація));
  - налаштуваннях таблиці тривоги і банера тривоги (у системній змінній: **AlarmManager.AlarmGlobals.g\_sTimeFormat := 'HH:mm:ss:Z'**).
- перемикання часу елементів візуалізації між локальним і всесвітнім (UTC) за допомогою системної змінної **VisuElements.Visu\_DateTime.DisplayUTC** (тип **BOOL**). Змінна впливає на елементи:
  - ті, що відображають системний час (тобто у яких у параметрі **Texts** (Тексти) вказано специфікатори формату часу, але до яких не прив'язано текстової змінної);
  - тренд;
  - таблицю тривоги і банер тривоги;
  - аналоговий годинник;
  - елемент вибору дати і часу (якщо до нього не прив'язана змінна).

23:03:2023 12:14:55:+03:00

Рисунок 7.19 – Приклад відображення системного часу з часовим поясом

## 7.11. Покращення в роботі з рецептами

Нова галочка в налаштуваннях рецептів дає змогу визначити, чи потрібно перезаписувати файли рецептів під час завантаження проєкту в контролер.

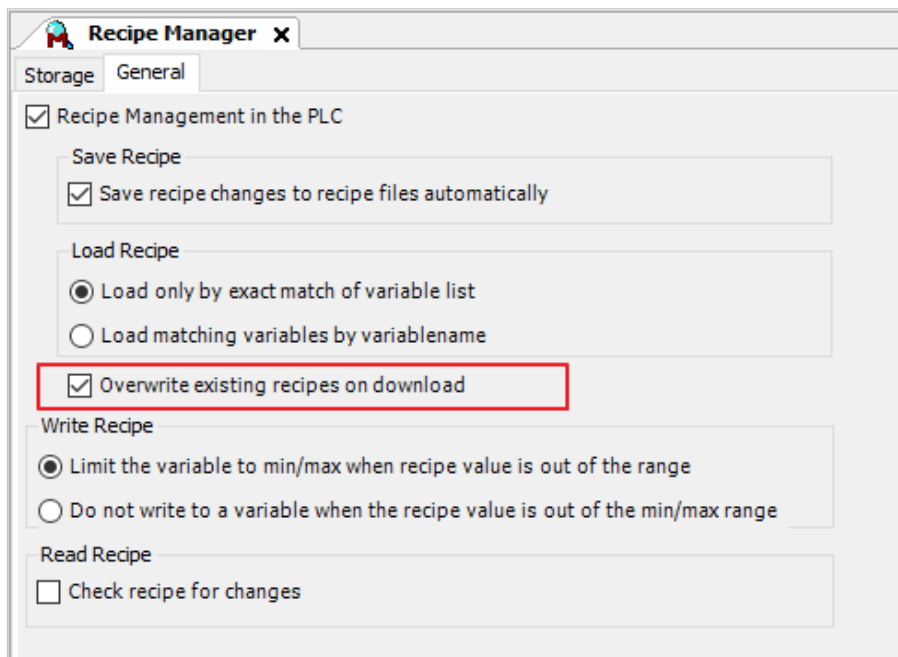


Рисунок 7.20 – Нове налаштування у меню рецептів

## 7.12. Спрощене введення змінних типу TIME/LTIME

У минулих версіях CODESYS для введення у візуалізації змінної типу TIME/LTIME потрібно було використовувати МЕК-нотацію (наприклад, **T#22h**). У **CODESYS V3.5 SP17** додали можливість вводити такі значення в простішій формі. Для цього додано можливість для специфікаторів формату часу вказувати "число знаків після коми". Наприклад:

- **%t[HH2]** - введення години, кількість знаків після коми - 2. Якщо ввести **6.25**, то у змінну буде записано **T#6h15m** ("6 з чвертю годин");
- **%t[HH4]** - введення години, кількість знаків після коми - 4. Якщо ввести **6.1250**, то у змінну буде записано **T#6h7m30s** ("6 з однієї восьмої годин").

Функціонал не дає змоги в межах одного елемента ввести значення для кількох розрядів часу (тобто не можна налаштувати елемент так, щоб введення значення **6.25** означало "6 годин 25 хвилин").

## 7.13. Покращення в елементі XY-графік

Наразі доступні такі нові можливості у роботі з компонентом XY-графік:

- можливість вставити команди зумування/панорамування за допомогою контекстного меню елемента;
- можливість змінити орієнтацію підписів біля поділок осі Y і відображати замість цифр тексти з обраного списку текстів у **Configure XYChart Settings – Axes - Extended display parameters** (Конфігурація графіка XY - Налаштування осі Y - Extended display parameters).

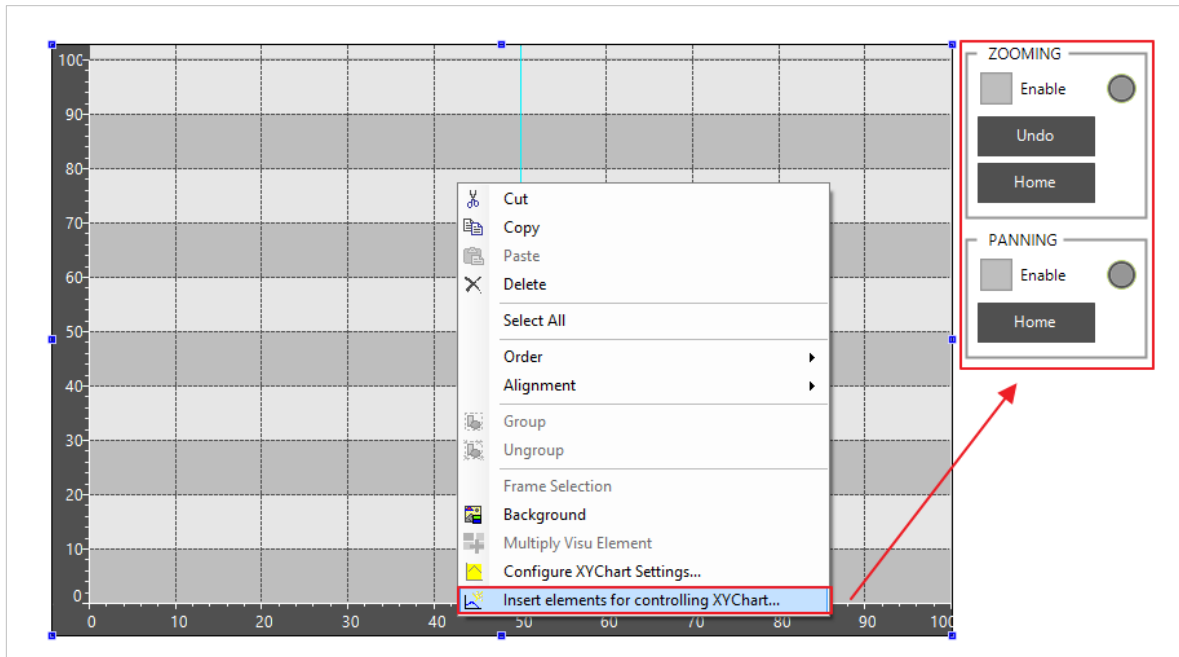


Рисунок 7.21 – Контекстна команда для додавання на екран елементів керування відображенням графіку

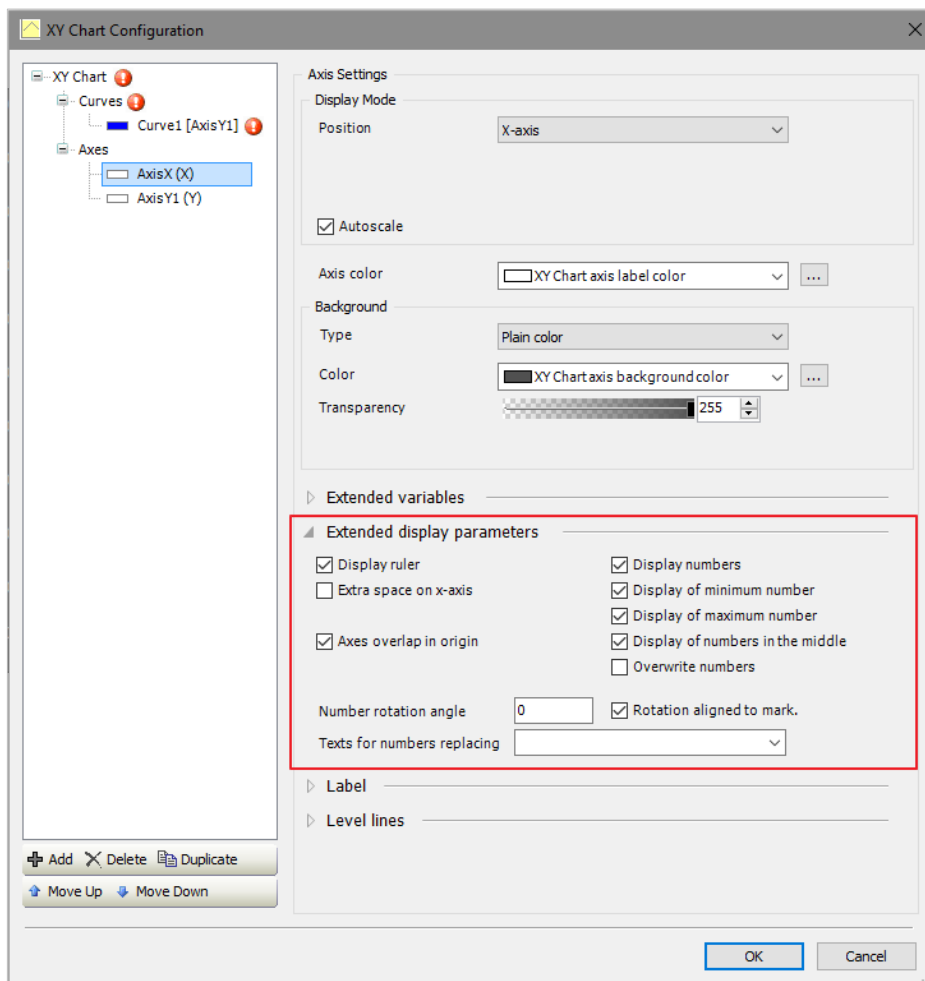


Рисунок 7.22 – Нові налаштування для осей графіку

## 7.14. Переміщення діалогів введення за допомогою миші

Наразі можливо переміщати діалоги вводу по екрану.

Інформацію про цю нову функцію:

- для переміщення потрібно навести курсор на вільне місце екрана діалогу або елемент **Invisible Input** (Невидиме введення) зі встановленою галочкою **Used as pointing area**, затиснути ЛКМ і перемістити курсор у потрібне місце екрана;
- у стандартних діалогах введення з бібліотеки **VisuDialogs** потрібно затиснути ЛКМ на заголовку діалогу;



### ПРИМІТКА

*Наразі підтримується тільки в web-візуалізації і тільки в разі встановлення в менеджері візуалізації галочки **Support client animations and overlay of native elements** (Підтримка клієнтських анімацій і накладення вихідних елементів).*

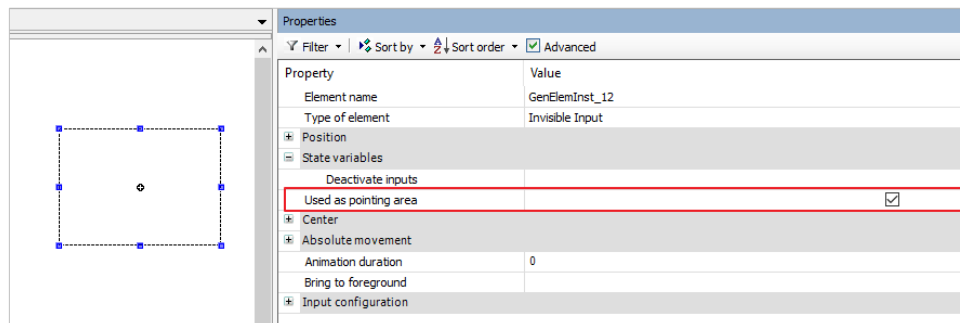


Рисунок 7.23 – Налаштування для активації функції переміщення

## 7.15. Оновлення команди Close Dialog

Команда **Close Dialog** (Закрити діалог) на вкладці параметрів **Input Configuration** (Конфігурація введення) здобула автоматичну підстановку поточного діалогу

Якщо команда **Close Dialog** (Закрити діалог) використовується в елементі, доданому в будь-який діалог - то за замовчуванням пропонується закрити саме цей діалог (використовується заміник 'Current Dialog').



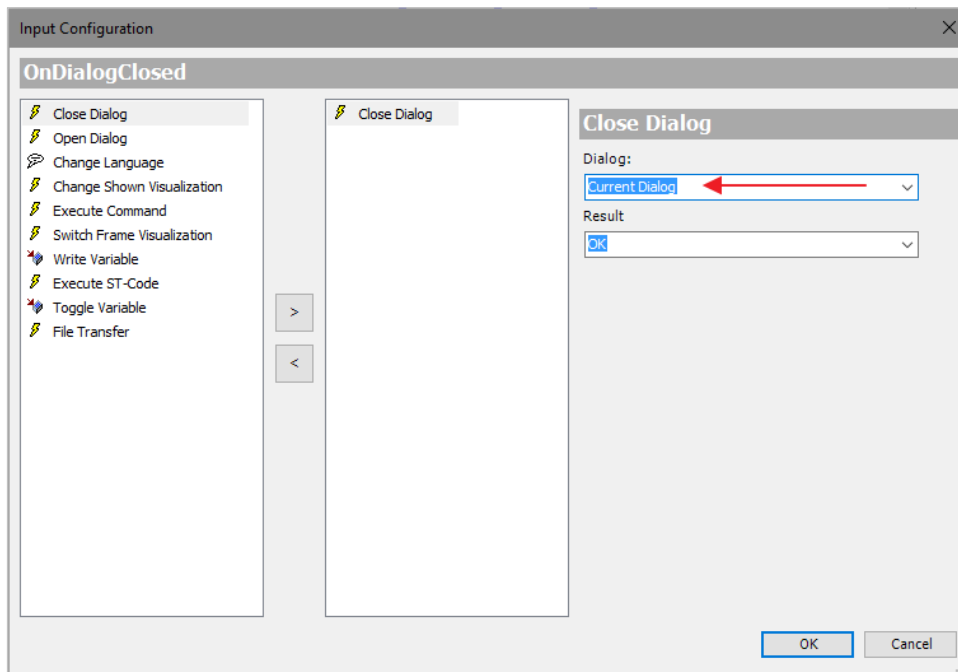


Рисунок 7.24 – Автоматичний заміник імені діалогу

### 7.16. Відключення невикористаних в проєкті екрани візуалізації

Наразі у CODESYS SP17 наявна можливість відключити в проєкті невикористовувані екрани візуалізації. Для цього треба викликати **Deactivate Unused Visualizations** команду:

**Visualization Manager – Visualizations - ПКМ на імені екрану - Deactivate Unused Visualizations**  
(Менеджер візуалізації - Візуалізації - ПКМ на екран - Deactivate Unused Visualizations)

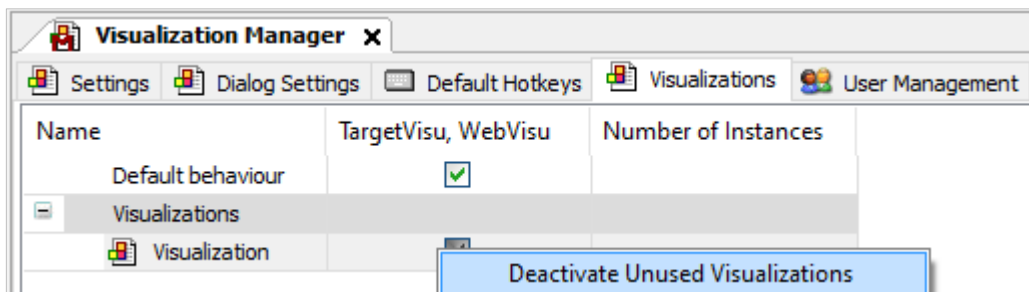


Рисунок 7.25 – Команда на відключення невикористаних екранів

### 7.17. Відключення в проєкті POU для неявних перевірок у бібліотеках візуалізації

Під час додавання в проєкт POU для неявних перевірок (ПКМ на пристрій **Application – Add Object – POU for Implicit Checks** (ПКМ на пристрій Application - Додавання об'єкта - POU для неявних перевірок)) їхнє виконання за замовчуванням відбуватиметься і в бібліотеках візуалізації, що певною мірою знижує швидкість відмальовування й опрацювання візуалізації.

Тепер ці перевірки для візуалізації можна відключити:

**Project – Project Settings – Visualization – General – встановити галочку Advanced - зняти галочку Activate implicit checks for visualization POU**

(Проєкт - Установки проєкту - Візуалізації - Загальне - Додатково - зняти галочку Activate implicit checks for visualization POU).

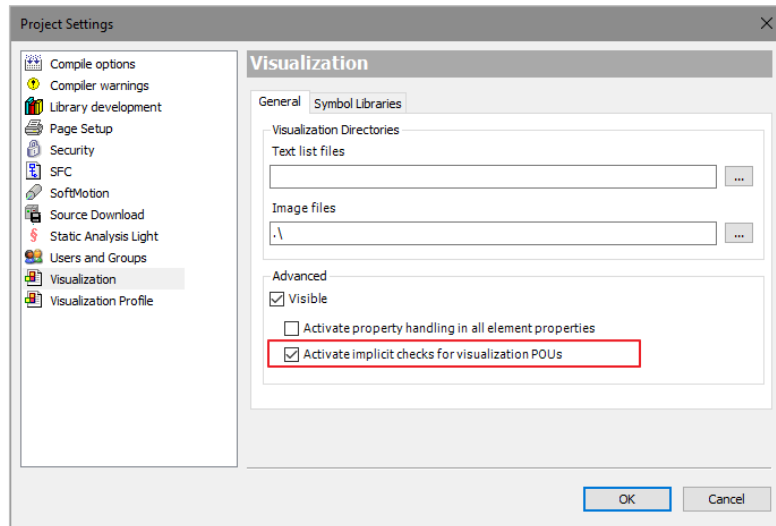


Рисунок 7.26 – Налаштування для відключення POU для неявних перевірок

## 7.18. Керування видимістю кнопок вкладок в елементі Tabs (Набір вкладок)

Оновлень також зазнав компонент візуалізації Tabs (Набір вкладок).

У групі параметрів **References** (Посилання) для кожного фрейма тепер присутній параметр **Invisible**, який дає змогу керувати видимістю кнопки цієї вкладки.

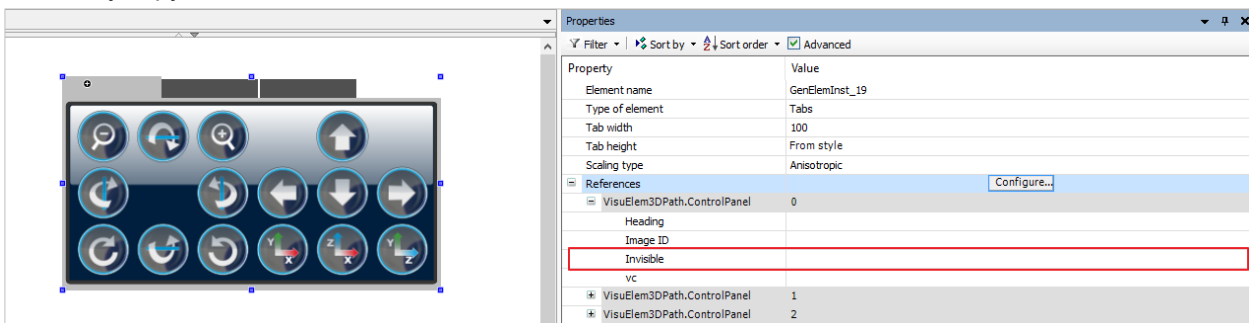


Рисунок 7.27 – Налаштування видимістю кожного фрейма компоненту Tabs

## 7.19. Оновлення події OnValueChanged

У вкладці **Input configuration** (Конфігурація введення) додано подію **OnValueChanged**. Подія генерується при зміні значення змінної, прив'язаної до елемента, за допомогою самого елемента (наприклад, за допомогою екранної клавіатури). Це, зокрема, дає змогу провести валідацію нового значення після його введення.



### ПРИМІТКА

*Подія не генерується в тому випадку, якщо значення змінної змінюється з коду програми.*

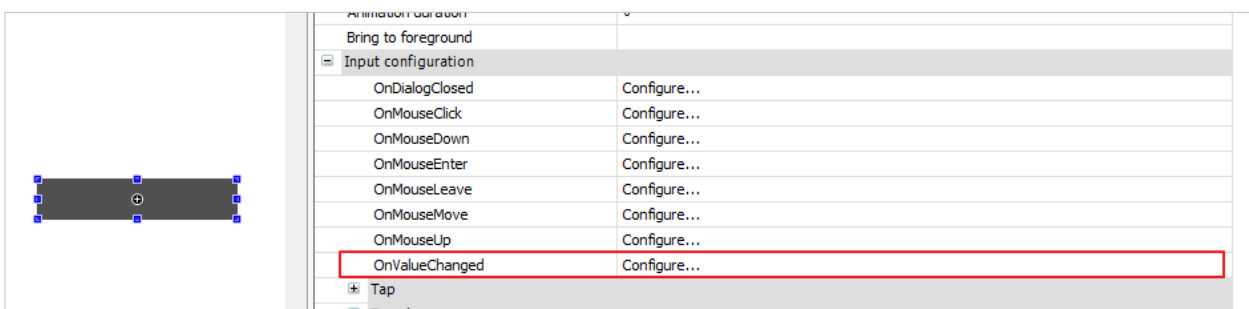


Рисунок 7.28 – Подія OnValueChanged у конфігурації введення

## 7.20. Обробка натискань на елементи візуалізації в коді програми

Для обробки натискань на елементи візуалізації в коді програми слід:

- створити ФБ з реалізацією інтерфейсу (**IMPLEMENTS**) **VisuElems.VisuElemBase.IInputOnElementEventHandler**;
- додати реалізацію для його методу обробки натискань **HandleInputOnElementEvent** (див. вбудовану документацію) на підставі значення входу **event** (структура типу **VisuElems.VisuElemBase.VisuStructInputOnElementEvent**; містить інформацію про клієнт візуалізації, елемент візуалізації та тип події - натискання, відпускання тощо);
- оголосити в програмі екземпляр створеного ФБ;
- ініціалізувати його в програмі: **VisuElems.VisuElemBase.g\_VisueEventManager.SetInputOnElementEventHandler(<ім'я екземпляра>)**. Бажано створити окрему програму і прив'язати її в **Visualization Manager** (Менеджері візуалізації) до поля **Call after visu initialization**. Прив'язувати цю програму до завдання не потрібно;
- екземпляр ФБ буде автоматично викликатися при кожному натисканні на будь-який з елементів візуалізації.

## 7.21. Додаткові опції для обробки мультикасання

У версії 4.1.0.0 плагіну візуалізації додано:

- нове налаштування в **Visualization Manager** (Менеджері візуалізації): якщо встановлено галочку **Multitouch handling** (обробка мультикасання), то доступна галочка **Scroll elements with scrollbar**. У разі встановлення цієї галочки для елементів, що підтримують прокрутку (наприклад, таблиця і Combobox), буде відображатися смуга прокрутки. У разі відсутності галочки смуга прокрутки у таких елементів відображатися не буде - прокрутка повинна буде здійснюватися жестами;
- нове налаштування у вузлі **WebVisu** (Веб-візуалізація): якщо встановлено галочку **Handle touch as mouse events**, то на пристроях із сенсорним екраном жести обробляються як натискання миші. Ця опція потрібна, наприклад, для керування повзунком або смугою прокрутки на сенсорному пристрої.

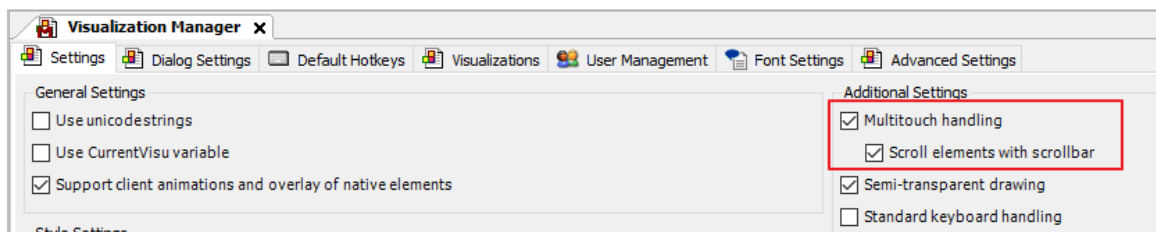


Рисунок 7.29 – Налаштування мультикасання та активації смог прокрутки

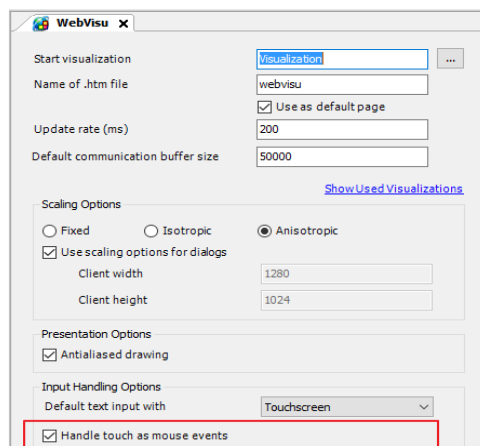


Рисунок 7.30 – Налаштування для обробки жестів як натискання миші

## 7.22. Зміна зовнішнього вигляду діалогу передавання файлів

Зовнішній вигляд діалогу завантаження/вивантаження файлів може бути змінений шляхом редагування [CSS](#) у файлі в пам'яті ПЛК: /<робоча директорія CODESYS>/PlcLogic/Visu/webvisu.htm

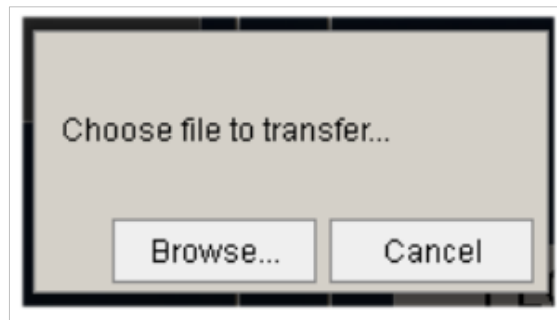


Рисунок 7.31 – Зовнішній вигляд стандартного діалогу передавання файлів

```

C:\ProgramData\CODESYS\CODESYSControlWinV3\FA001425\PlcLogic\visu\webvisu.htm - Notepad++
Файл Правка Поиск Вид Кодировки Синтаксисы Опции Инструменты Макросы Запуск Плагины Вкладки ?
webvisu.htm
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta name="viewport" content="initial-scale=1.0">
5 <meta http-equiv="content-type" content="text/html; charset=UTF-8" />
6
7 <style type="text/css">
8 body {font-family: Arial, Helvetica, sans-serif;}
9 .fileTransferDialog {
10 display: block;
11 position: fixed;
12 top: 0px;
13 left: 0px;
14 z-index: 300;
15 width: 100%;
16 height: 100%;
17 overflow: hidden;
18 background-color: rgba(0,0,0,0.4);
19 }
20
21 .fileTransferDialogContent {
22 display: block;
23 margin: 0;
24 position: absolute;
25 top: 50%;
26 left: 50%;
27 margin-right: -50%;
28 transform: translate(-50%, -50%);
29 background-color: #d4d0c8;
30 border: 1px solid #888;
31 width: 200px;
32 height: 100px;
33 box-shadow: 2px 2px 6px 6px rgba(0,0,0,0.5);
34 }
35
36 .fileTransferButton {
37 position: relative;
38 margin-top: 2px;
39 margin-right: 5px;
40 float: right;
41 font-size: 12px;
42 width: 80px;
43 height: 25px;
44 border: 1px solid #888;
45 }
46

```

Рисунок 7.32 – Файл редагування стилю діалогу передавання файлів.