



**СВ АЛТЕРА**  
ЕЛЕКТРОТЕХНІКА & АВТОМАТИЗАЦІЯ

# Навчальні матеріали **SCADA zenon**

Базовий курс



**zenon**  
do it your way

# Продукція

СВ Альтера є дистриб'ютором таких відомих виробників обладнання, як Lenze, Lovato Electric, VIPA, WEG, Socomec, COPA-DATA, Relpol та багатьох інших.

Продукція компанії використовується на всіх етапах подачі і розподілу електроенергії, для автоматизації будь-яких технологічних процесів, що сприяє істотній економії енергії та інших ресурсів нашим клієнтам

## Програмовані засоби АСУ ТП

- » програмовані логічні контролери VIPA
- » ПЛК ОВЕН
- » модулі розширення ПЛК ОВЕН
- » контролери багатофункціональні Ascon AC
- » програмовані логічні реле / модулі
- » системи візуалізації (SCADA)
- » панелі оператора
- » промислові комунікації

## Обладнання КВП

- » контролери технологічного обладнання
- » контрольно-вимірювальні прилади і регулятори
- » лічильні пристрої мікропроцесорні
- » перетворювачі інтерфейсів
- » пристрої сигналізації та регулювання рівня
- » пристрої управління насосами
- » пристрої ізолюючі (бар'єри іскрозахисту)
- » датчики і перетворювачі
- » елементи пневмоавтоматики

## Електроприводна техніка

- » перетворювачі частоти для асинхронних двигунів
- » сервопривод
- » привід постійного струму
- » кроковий електропривод
- » пристрої плавного пуску
- » двигуни та мотор-редуктори

## Електротехніка

- » автоматичні вимикачі
- » рубильники та запобіжники
- » електричні щити та ящики
- » сухі трансформатори 10/0,4 kV
- » високовольтні секції
- » шинопроводи та пристрої для прокладки кабелю
- » контактори та реле
- » кнопки, перемикачі, сигнальні пристрої
- » клеми і роз'єми



**zenon**  
do it your way

## Зміст

Знайомство з системою .....	1
Структура	
Ліцензування	
Створення нового проекту .....	3
Поняття робочого простору	
Види проектів	
Конфігурування проекту	
Екрани та функції .....	7
Концепція фреймів	
Робота з екранами	
Елементи відображення і керування	
Типи екранів та їх шаблони	
Змінні .....	16
Драйвери	
Типи даних	
Прості та структуровані змінні	
Реалізація дисплейної мнемосхеми процесу .....	22
Відображення значень змінних	
Символи та екранні елементи	
Динамізація об'єктів	
Скрипти .....	49
Хронологічний список подій .....	52
Тривоги .....	59
Визначення лімітів	
Матриці реакції	
Архіви .....	64
Створення циклічного архіву	
Створення каскадного архіву	
Тренди .....	75
Віртуальна клавіатура .....	81

Адміністрування користувачів .....	83
Меню.....	93
Головне меню	
Контекстне меню	
Блокування .....	98
Перерозподілення .....	101
Інтернаціоналізація .....	102
Зміна мови	
Зміна одиниць вимірювання	
Математичний, системний та симуляційний драйвер .....	108
Математичний драйвер	
Системний драйвер	
Симуляційний драйвер	
Короткий опис модулів розширення функціональності системи.....	115

## Знайомство з системою

Ці матеріали познайомлять Вас з базовими можливостями системи керування та навчать створювати нескладні проекти візуалізації. Тут розглянуті найбільш важливі етапи проектування, такі як розробка екранів мнемосхем, елементів керування, ведення архівів, адміністрування користувачів та багато інших моментів з якими стикається інженер під час створення систем диспетчеризації.

*Виконання вказівок, представлених в даному посібнику, не заміняє проходження повноцінних навчальних курсів.*

### СТРУКТУРА

Концептуально SCADA zenon представляє собою програмно-технічний комплекс який складається з середовища розробки проектів – Editor та середовища виконання – Runtime. Це дві незалежні оболонки, які виконують кожна свою функцію і можуть встановлюватися на робочу станцію окремо одна від одної.

Середовище розробки містить набір модулів та інструментів необхідних для написання повнофункціональних програм і пакет драйверів для підключення до найбільш розповсюджених апаратних засобів PLC.

Відповідно до задач, які необхідно вирішувати, SCADA zenon має чотири редакції:

zenon Operator	HMI-система, призначена для створення нескладних проектів візуалізації для панелей оператора, вбудованих систем та локальних диспетчерських пунктів на ПК з певними функціональними обмеженнями.
zenon Supervisor	Незалежна SCADA-система, яка містить повний набір функцій для реалізації диспетчерського керування та збору даних. Ефективно вирішує задачі побудови систем автоматизації будь-якої складності.
zenon Energy Edition	Спеціальна редакція системи, яка містить додатковий набір функцій для енергетичної промисловості.
zenon Pharma Edition	Спеціальна редакція системи, яка містить додатковий набір функцій для фармацевтичної промисловості.

Вибір необхідної редакції здійснюється на етапі установки системи. Для цього тренінгу ми будемо використовувати редакцію **zenon Supervisor**.

## ЛІЦЕНЗУВАННЯ

Система ліцензується по кількості апаратних змінних та тих додаткових модулів, які використовуються в проекті і не входять в базову поставку.

Якщо Ви не маєте ліцензії, то програма (Editor\Runtime) після установки запуститься в DEMO-режимі.

Єдина відмінність ліцензованої версії від демонстраційної – це часові обмеження що накладаються на систему. Перші 40 запусків програма буде працювати 30 хвилин, після чого час безперебійної роботи урізається до 10 хвилин. По закінченню визначеного часу, з'явиться відповідне повідомлення і система буде закрита. Проект, який розробляв користувач, повністю зберігається. Для продовження роботи необхідно повторно запустити систему.

*Проходження даних навчальних курсів повністю можливе в демонстраційному режимі. Для комерційного використання необхідне придбання ліцензії. Для цього звертайтеся в центральний офіс компанії СВ Альтера: Київ, бул. Івана Лепсе 4, (044) 496-18-88, office@sv-altera.com*

## Створення нового проекту

В цьому розділі ми розглянемо поняття робочого простору та мультипроектності в zenon, видами проектів, і почнемо створювати наш перший проект.

### ПОНЯТТЯ РОБОЧОГО ПРОСТОРУ

Технологія горизонтальної та вертикальної відкритості передбачає побудову мультипроектних систем, головна ідея яких полягає в тому, що створюється ряд проектів, які ієрархічно пов'язані оди з одним. Тобто, є головний проект, який містить декілька підпроектів, які в свою чергу також можуть містити свої підпроекти. Система не обмежує розміри такої структури.

*Горизонтальна відкритість – можливість з одної робочої станції керувати різними проектами, які знаходяться на одному ієрархічному рівні на різних ПК.*

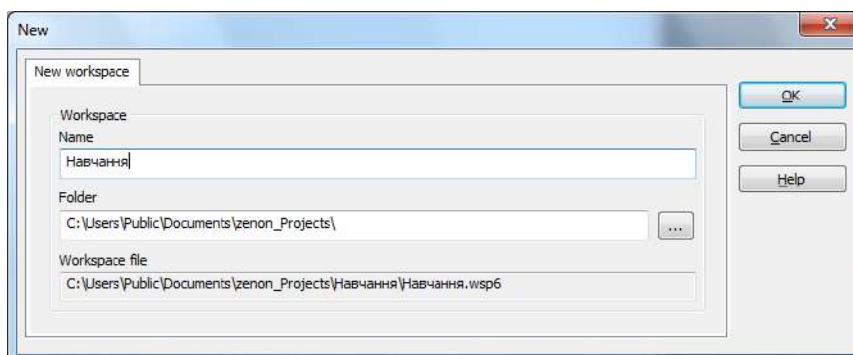
*Вертикальна відкритість – можливість з центральної робочої станції керувати проектами, які знаходяться нижче по ієрархічній структурі системи автоматизації.*

Для забезпечення реалізації мультипроектності, усі проекти створюються на базі робочого простору, де потім, при потребі, і виконується формування структури системи.

Мультипроектну структуру доцільно використовувати для великих, розгалужених проектів, в той час як для простих систем, в більшості випадків, в робочому просторі створюють один проект.

### Створення робочого простору

- ▶ В меню *File* виберіть *Workspace -> New...*
- ▶ Для робочого простору задайте ім'я 'Навчання', та при необхідності скоректуйте директорію розташування.



Після натиснення *OK*, система автоматично створить файл 'Навчання.wsp6'.



## ВИДИ ПРОЕКТІВ

Розрізняють *Стандартні проекти* та *Глобальні проекти*. Різниця між ними полягає у тому, що останній варіант доцільно використовувати, якщо ми будемо створювати декілька однотипних проектів. При цьому, в головній частині одноразово визначаються шрифти, кольорові палітри, моделі обладнання і т.д., які будуть актуальні для усіх проектів, що знаходяться в структурі глобального проекту.

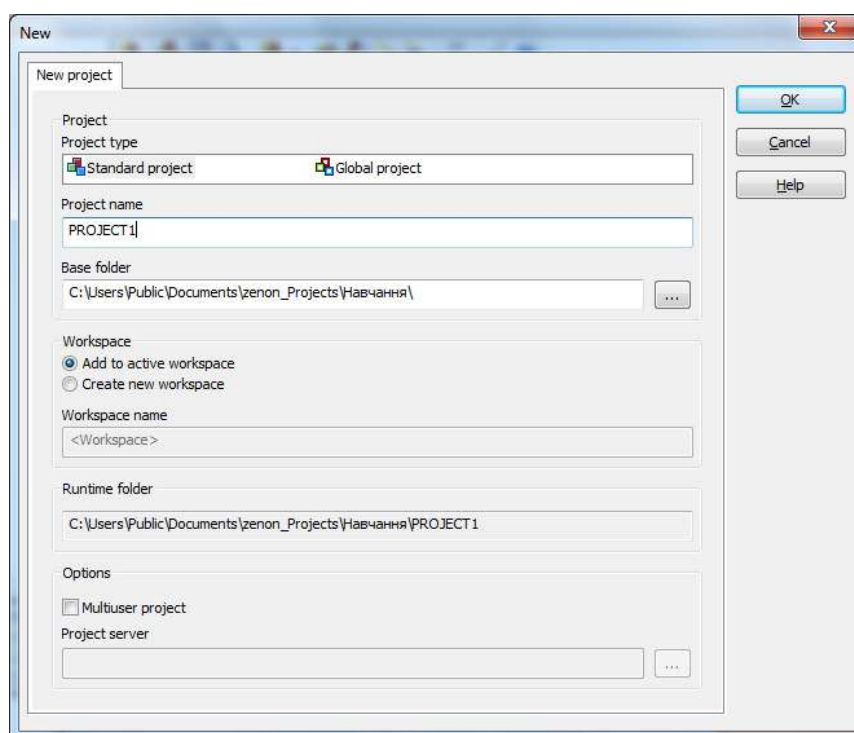
Також, на початковому етапі розробник має можливість активувати опцію *Багатокористувацької розробки*, яка є однією з визначальних характеристик zenon.

*Багатокористувацька розробка – можливість команді проєктувальників одночасно здійснювати розробку одного проєкту.*

В нашому навчанні ми будемо створювати простий стандартний проект.

## Створення проекту

- В меню *File* виберіть *Project new...*



- Задайте ім'я проекту 'PROJECT1'



Інші налаштування залиште без змін і, натиснувши кнопку *OK*, створіть проект. Система автоматично згенерує структуру та файли внутрішньої SQL-базу проекту.

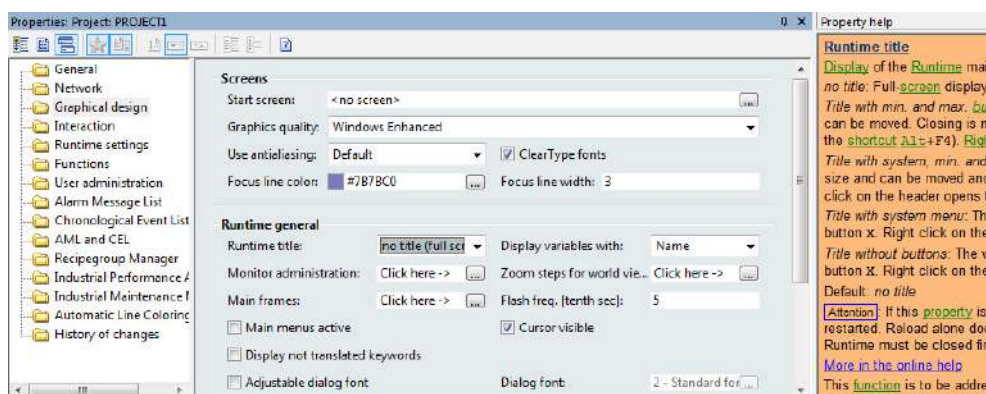
Після цього автоматично з'явиться діалогове вікно Майстра створення нового проекту, написаного на VSTA. Ми будемо розробляти проект уручну, тому просто закрийте його кнопкою *Cancel*.

## КОНФІГУРУВАННЯ ПРОЕКТУ

На даному етапі, після виконаних дій ми маємо робочий простір 'Навчання' з проектом 'PROJECT1'. Усі налаштування створеного проекту знаходяться у вікні властивостей, яке знаходиться в нижній частині середовища розробки. Залежно від вибраного пункту дерева проекту, вікно властивостей буде містити відповідний до нього контекст.

- В дереві проекту клікніть на назві проекту

У вікні властивостей з'являться усі найважливіші налаштування цього проекту. Більшість з них зараз займати не будемо. Поступово, ми з ними познайомимося. Зараз лише змінимо графічне представлення екранів нашого проекту.



- В розділі *Graphical design* для пункту *Runtime title* зі спадного меню виберіть *no title (full screen)*

При цьому, екрани дисплейних мнемосхем не будуть містити заголовку з системними кнопками. Інші варіанти вибору дозволяють встановлювати заголовки для екранів, системні кнопки закриття та згортання.

Справа від вікна властивостей проекту знаходиться вікно інтерактивної допомоги, яке автоматично виводить довідкову інформацію відносно активованих опцій. Таким чином, розробник відразу може орієнтуватись в призначенні тієї чи іншої властивості.

При створенні проекту система керування на жорсткому диску автоматично створить усі необхідні папки. По замовчуванню, RT-файли проекту містяться в директорії *zenon\_Projects*. Також, кожному проекту присвоюється унікальний

Project ID, який відповідає створеній папці у внутрішній SQL-базі zenon. Всю цю інформацію можна знайти у властивостях проекту, розділ *General*.

Конкретні шляхи розміщення файлової структури системи, залежно від операційної системи, описані в мануалі Installation and updates -> File structure.

Перенос і збереження проекту здійснюється за рахунок створення резервної копії самого проекту чи усього робочого простору.

### Резервна копія проекту

- ▶ В дереві проекту зайдіть в розділ *Project backups* і клікніть на іконці *Create backup*
- ▶ Після створення резервної копії експортуйте її у необхідну директорію: іконка *Export backup*

Проект представляє собою zip-архів з RT-файлами проекту та SQL-бази даних. Відкрити/відновити такий проект можна лише в існуючому робочому просторі.

- ▶ Головне меню *File -> Restore project backup*

### Резервна копія робочого простору

- ▶ Зайдіть в головне меню *File -> Workspace* і виберіть пункт *Create backup*
- ▶ Задайте ім'я та директорію збереження файлу робочого простору

Система створить wsb-файл, який тепер можна легко перенести на інший комп'ютер.

Для відновлення робочого простору з резервної копії, необхідно виконати наступні кроки:

- ▶ Зайдіть в головне меню *File -> Workspace* і виберіть пункт *Restore backup*
- ▶ Виберіть необхідний wsb-файл

Якщо при роботі Ви просто закрили робочий простір, створили новий, чи система керування автоматично не підгрузила останній робочий простір з яким Ви працювали, то для його відкриття необхідно виконати наступне:

- ▶ Зайдіть в головне меню *File -> Workspace* і виберіть пункт *Open*
- ▶ Виберіть в папці з назвою робочого простору файл \*.wsb6

## Екрани та функції

Екрани представляють собою інтерфейс взаємодії оператора з процесом. Це мнемосхеми відображення об'єкта автоматизації, які містять елементи керування та інформування оператора.

### КОНЦЕПЦІЯ ФРЕЙМІВ

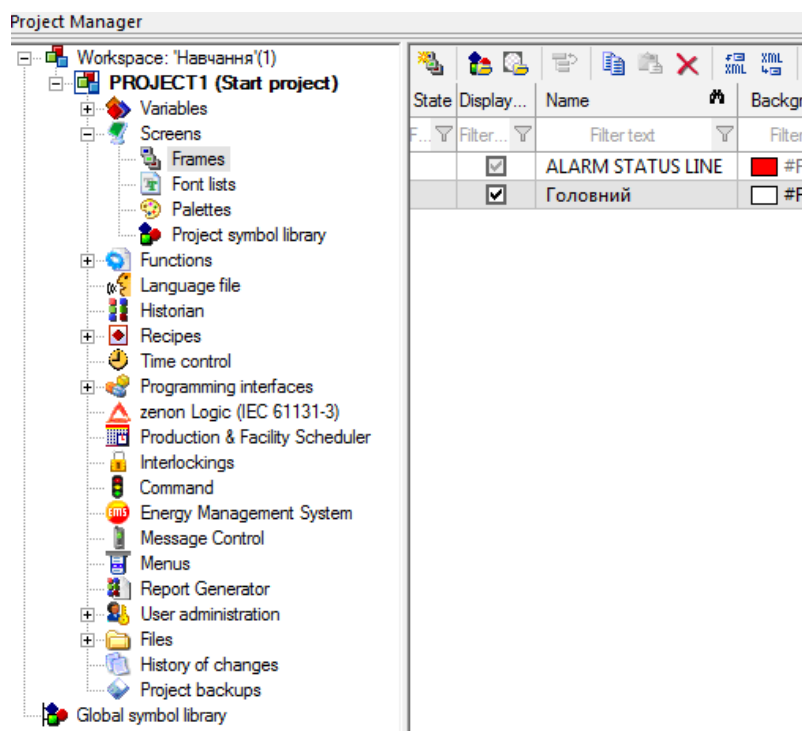
В zenon усі екрани базуються на фреймах які утворюють “кістяк” проекту. Відповідно до розміщення та розміру фрейму, екрани, що утворені на його основі, з'являються в проекті у чітко визначених зонах та приймають його розміри.

Зазвичай область представлення проекту розділяють на три зони: верхня зона інформаційного характеру, головна зона з представленням процесу, індикацією параметрів та елементами керування процесом, та нижня зона – панель навігації по проекту.

Для нашого демонстраційного проекту ми створимо два фрейми: 'Головний' та 'Навігація'.

### Створення фрейму

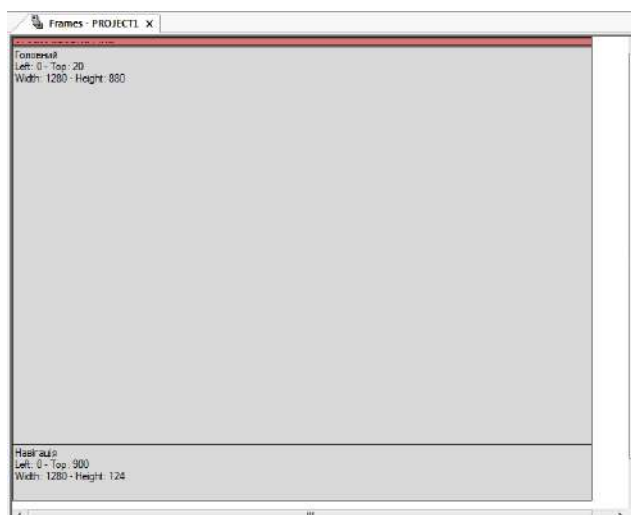
- ▶ В менеджері проекту відкрийте вузол *Screens* і виберіть пункт *Frames*
- ▶ У вікні детального перегляду клікніть на іконці *New frame*



В розділі *General* вікна властивостей перейменуйте створений фрейм 'Frame\_0' на 'Головний', а в розділі *Position* задайте йому розмір координатами 0\20 x 1280\900 (для монітору 1280 x 1024).

Для створення навігаційного фрейму виконайте аналогічні кроки. Задайте йому ім'я 'Навігація' та координати 0\900 x 1280\1024.

Тепер редактор фреймів повинен мати наступний вигляд:

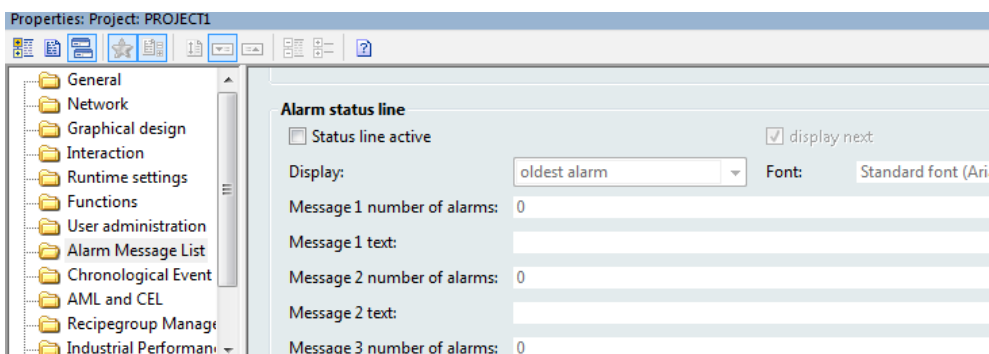


В верхній частині редактора фреймів ми залишили поле шириною в 20 пікселів для відображення головного меню проекту, яке створимо пізніше. Зараз це поле займає червоний фрейм Статусного вікна тривоги.

*Статусне вікно тривоги – спеціальний тип вікна, призначений для відображення в онлайн режимі тривоги, що з'являються в системі. Це вікно представляє одно-строкове поле, яке з'являється поверх усіх вікон і інформує оператора про появу тривоги.*

Давайте для нашого демонстраційного прикладу відключимо це вікно.

- У властивостях проекту зайдіть в розділ *Alarm Message List*
- В полі *Alarm status line* деактивуйте бокс *Status line active*

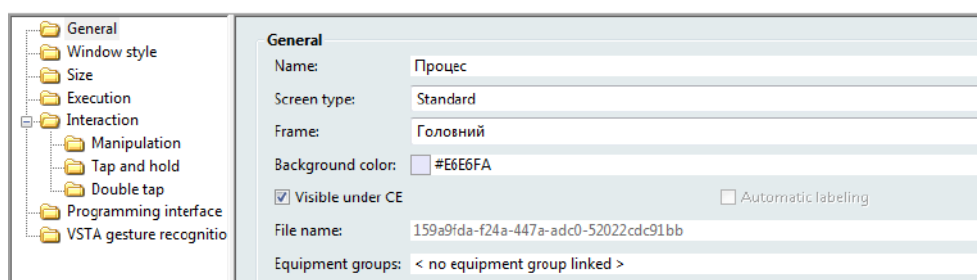


## РОБОТА З ЕКРАНАМИ

### Створення екрану

- ▶ В менеджері проекту виберіть вузол *Screens*
- ▶ У вікні детального перегляду клікніть на іконці *New screen*
- ▶ Через вікно властивостей перейменуйте створений екран на 'Процес' та задайте довільний колір фону

Проконтролюйте, щоб створений екран базувався на фреймі 'Головний' та мав тип *Standard*.

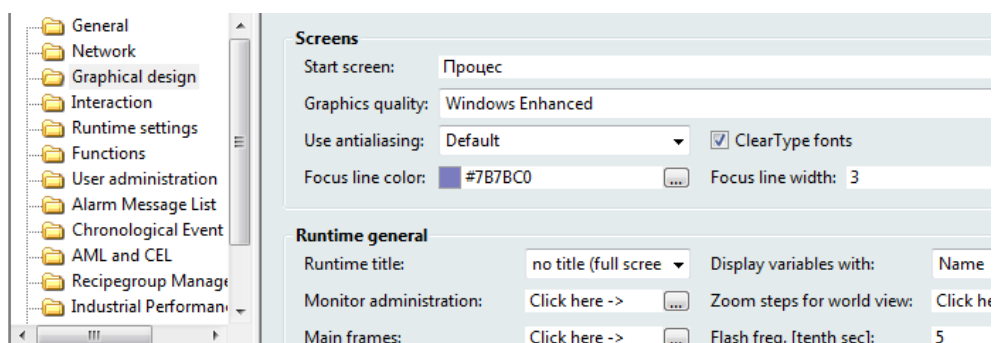


Після створення екрану, він автоматично відкриється в області головного вікна в Editor.

Аналогічним чином створіть екран для відображення нашої навігаційної панелі. Назвіть його 'Навігація', змініть колір фону та прив'яжіть до однойменного фрейму.

Таким чином, ми створили два екрани, при чому екран 'Процес' являється стартовим. Тобто саме він буде з'являтися першим при запуску проекту. За бажанням, цю прив'язку можна змінити.

- ▶ Виділіть мишкою в дереві проекту його назву і у вікні властивостей проекту зайдіть в розділ *Graphical design*.
- ▶ В полі *Start screen* повинно стояти 'Процес'



Залишіть все без змін.

Якщо ми зараз запустимо середовище виконання проекту – Runtime, то з'явиться тільки вікно 'Процес', а знизу буде пусте поле, оскільки в проекті не сконфігурований виклик екрану навігації.

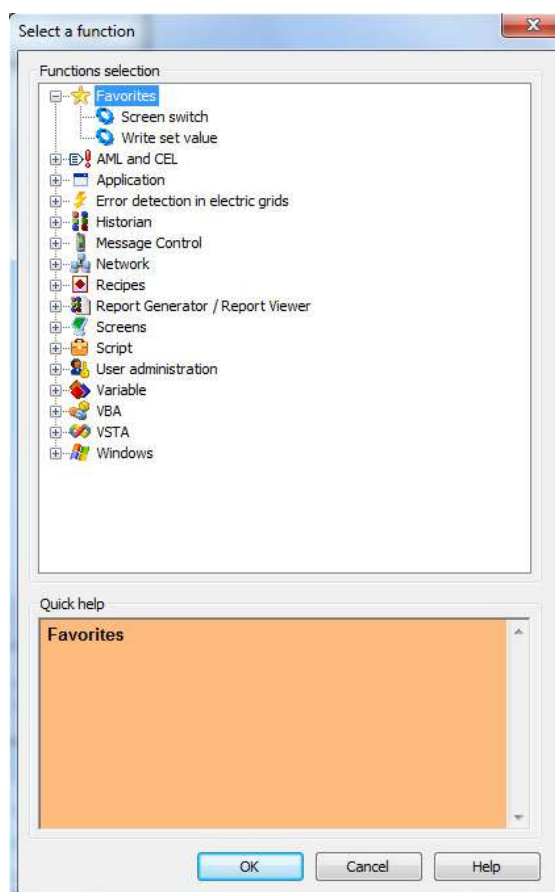
Керування екранами здійснюється через функції. Необхідно створити функцію виклику екрану навігації та сконфігурувати його виклик разом з головним вікном.

### Створення функції

- ▶ В менеджері проекту виберіть вузол *Functions*
- ▶ У вікні детального перегляду клікніть на іконці *New function*

Автоматично з'явиться вікно вибору функції. В zenon усі необхідні функції вже наперед створені. Проектувальнику необхідно лише вибрати відповідну функцію та сконфігурувати її.

- ▶ У вікні функцій виберіть *Screen switch*, яка знаходиться в розділі *Favorites*



- ▶ У наступному діалоговому вікні задайте перемикання на екран 'Навігація'

- ▶ У властивостях створеної функції зайдіть в розділ *General* і в полі *Start function* перейменуйте її на 'Перехід на екран Навігація'
- ▶ Зайдіть у властивостях екрану 'Процес' в розділ *Execution* і прив'яжіть створену функцію 'Перехід на екран Навігація'

Тепер ми запустимо режим Runtime і перевіримо зроблену роботу.

- ▶ На верхній інструментальній панелі Editor клікніть іконку *Start Runtime* або натисніть *F5*

Перевірте чи все вийшло як заплановано і натиснувши комбінацію клавіш *Alt+F4* поверніться назад в Editor.

### Графічні стилі екранів

Оскільки на початку створення проекту в налаштуваннях *Graphical design* пункт *Runtime title* ми обрали опцію *no title (full screen)*, то інтерфейс екранів проекту не має жодних заголовків чи системних кнопок, тобто дисплейна мнемосхема займає всю площу відображення. Проте, при потребі, існує можливість визначення цих елементів індивідуально для окремих екранів (наприклад, рор-уп діалоги). Для цього у властивостях фрейму, на якому базується відповідний екран, необхідно зайти в розділ *Border -> Border type* і вибрати тип границі.

## ЕЛЕМЕНТИ ВІДОБРАЖЕННЯ І КЕРУВАННЯ

Усі елементи, що можуть використовуватись на екранах, поділяються на статичні та динамічні.

До статичних елементів відносяться: лінії, прямокутники, кола, текстові поля...

До динамічних елементів відносяться: покажчики, слайдери, перемикачі, цифрові поля, ActiveX-елементи...

Ці всі елементи можна знайти на інструментальній панелі *Elements* (як правило знаходиться справа) або в головному меню редактора.



### Статичний текст

- ▶ Відкрийте екран 'Процес'
- ▶ На інструментальній панелі виберіть елемент *Static text*
- ▶ В верхній частині екрану клікніть мишкою і не відпускаючи потягніть до утворення прямокутної текстової зони



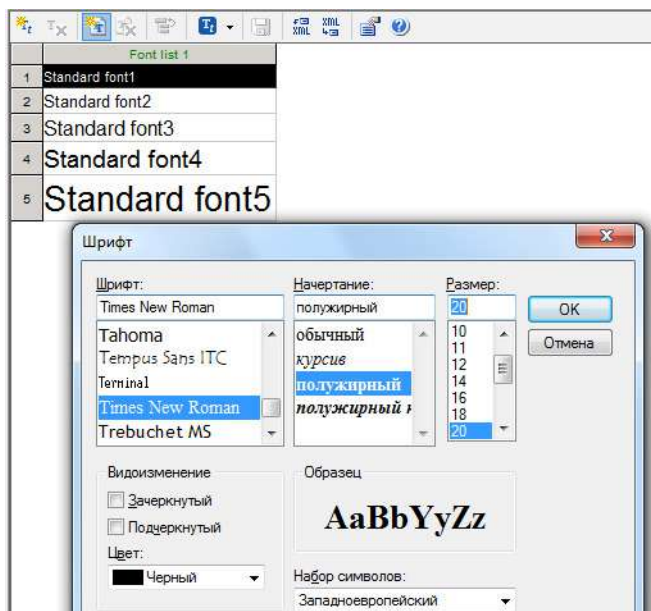
- Зайдіть в розділ *Representation* властивостей елемента і в полі *Text* задайте свій довільний текст, наприклад, 'Мій перший проект'

Після активації зроблених змін, в намальованому текстовому полі з'явиться введена фраза. Відредагуйте розмір таким чином, щоб вона повністю поміщалася в одну строку. Спробуйте для цього елемента змінити колір фону та тексту.

Давайте для цього тексту-заголовку створимо свій власний шрифт.

## Шрифт

- В дереві проекту відкрийте вузол *Screens* і виберіть пункт *Font lists*
- У вікні детального перегляду клікніть на іконці *New font*

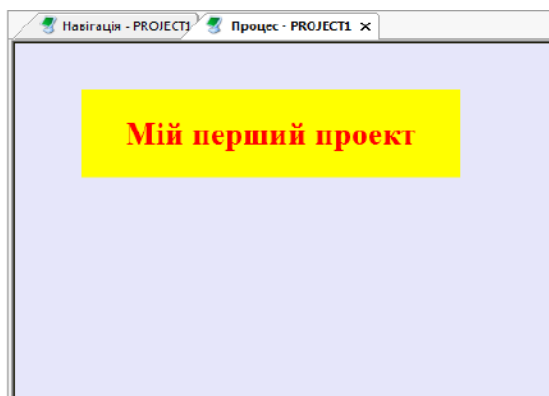


- В наступному діалоговому вікні виберіть шрифт Times New Roman, напівжирний, 20 і підтвердіть вибір кнопкою *OK*
- Через властивості створеного шрифту задайте йому ім'я 'Заголовок'

Цей шрифт ми можемо використовувати будь-де в проекті.

- Виберіть на головному екрані створений нами текстовий елемент і у властивостях *Representation* -> *Font* змініть шрифт на ім'я 'Заголовок'

Тепер наш екран повинен мати наступний вигляд:



Для керування проектом нам необхідно створити декілька кнопок, а саме: кнопку переходу на головний екран 'Процес', закриття Runtime і онлайн перезавантаження проекту.

*Онлайн перезавантаження – це функція, яка дозволяє без зупинки проекту вносити до нього зміни. Вона широко використовується під час розробки та в системах, критичних до переривання роботи середовища виконання.*

## Кнопки

Попередньо для кнопок необхідно створити відповідні функції.

- ▶ В менеджері проекту виберіть вузол *Functions*
- ▶ У вікні детального перегляду клікніть на іконці *New function*
- ▶ Виберіть функцію *Screen switch*
- ▶ У наступному діалоговому вікні задайте перемикання на екран 'Процес'
- ▶ Далі натисніть *OK* і на цьому закінчіть створення функції
- ▶ У властивостях цієї функції зайдіть в розділ *General* і в полі *Name* перейменуйте її на 'Перехід на екран Процес'

Аналогічним чином створіть функцію *Exit Runtime* та *Reload project online*, які знаходяться в вузлі *Application*. Перейменуйте їх на 'Закриття Runtime' та 'Онлайн перезавантаження' відповідно.

*Важливо давати функціям адекватні назви, щоб потім можна було легко в них орієнтуватися. Це ж стосується і назв екранів та змінних.*

- ▶ Відкрийте екран 'Навігація'

- ▶ На інструментальній панелі виберіть елемент *Button*
- ▶ В лівій частині екрану клікніть мишкою і не відпускаючи потягніть до утворення кнопки потрібного розміру
- ▶ Автоматично з'явиться діалогове вікно, в якому виберіть створену нами функцію 'Перехід на екран Процес'.
- ▶ Зайдіть в розділ *Representation* властивостей кнопки і в полі *Text line 1* підпишіть її як 'Процес'
- ▶ Змініть шрифт на *Standatd font 4* або створіть свій власний

В правій частині екрану навігації розмістіть системні кнопки закриття та перевантаження проекту 'Exit' і 'Reload'. Прив'яжіть до них відповідні функції. Ви можете за бажанням змінити колір кнопок та їх зовнішнє відображення.

На даному етапі екран 'Навігація' повинен мати наступний вигляд:

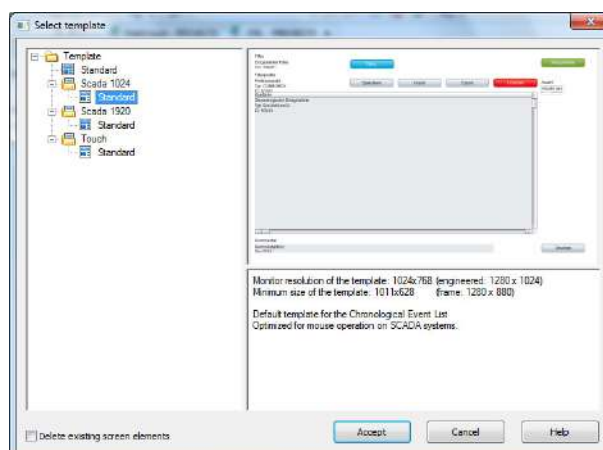


Запустіть Runtime і перевірте як працюють створені кнопки. Тепер ми можемо не закривати Runtime, а користуватись кнопкою Reload. Майте на увазі, що при натисканні кнопки 'Процес' в проекті нічого змінюватися не буде, оскільки цей екран в нас вже відкрито.

## ТИПИ ЕКРАНІВ ТА ЇХ ШАБЛОНИ

Для реалізації мнемосхем процесів та діалогових вікон використовуються звичайні екрани, які в zenon мають тип *Standard*. Для створення спеціальних екранів, таких як тривоги, архіви, тренди і т.д. необхідно використовувати відповідний тип, в якому системою наперед закладено ряд функцій та елементів, притаманних суто йому. Вибір типу здійснюється через властивості екрану *General* -> *Screen type*. Зайшовши в спадне меню можна вибрати будь-який спеціальний тип. Призначення кожного з них детально описується в мануалі [Screens](#).

Щоб зекономити розробнику час на створення елементів керування для спеціальних екранів, в систему керування інтегровано готовий набір шаблонів. Для кожного типу екрану створено свій шаблон з набором відповідних кнопок, текстових та списочних полів.



Докладніше процес створення спеціальних екранів та використання шаблонів буде розглянуто в подальших розділах даного навчального курсу.

## Змінні

В цьому розділі ми перейдемо до технічної сторони створення будь-якого проекту – організації зв'язку з апаратними засобами. Для цього ми виберемо та сконфігуруємо комунікаційний драйвер, а потім на його базі створимо декілька змінних.

### ДРАЙВЕРИ

Система керування отримує дані від обладнання через драйвер. Він вибирається відповідно до мережевого протоколу чи виробника PLC. В zenon інтегровано понад 300 різноманітних драйверів для підключення до найбільш розповсюдженого обладнання.

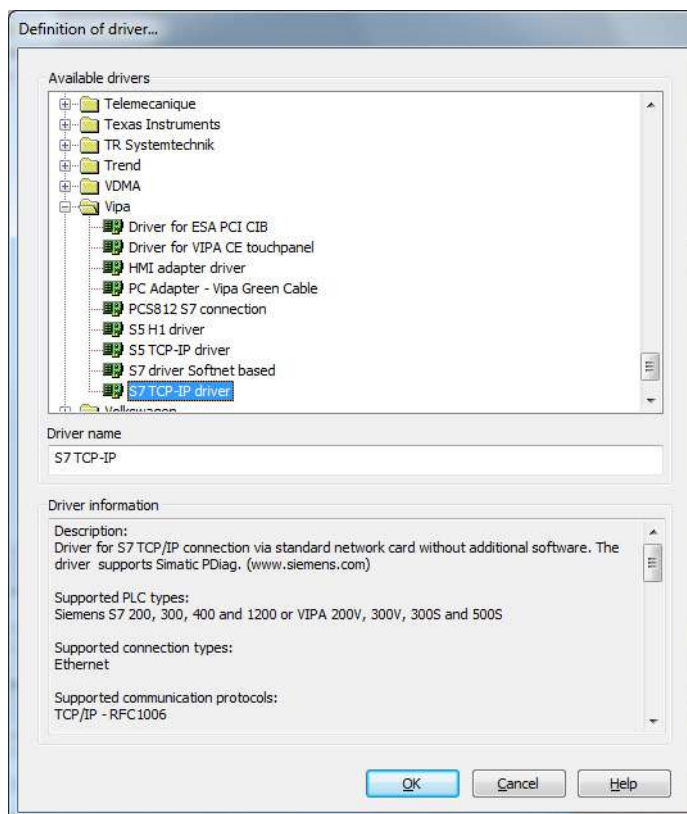
#### Конфігурування драйверу

- ▶ В менеджері проекту зайдіть в вузол *Variables* і виберіть пункт *Driver*

Як Ви бачите у вікні детального перегляду, тут вже є декілька драйверів для створення внутрішніх, математичних та системних змінних

- ▶ У вікні детального перегляду клікніть на іконці *New driver*

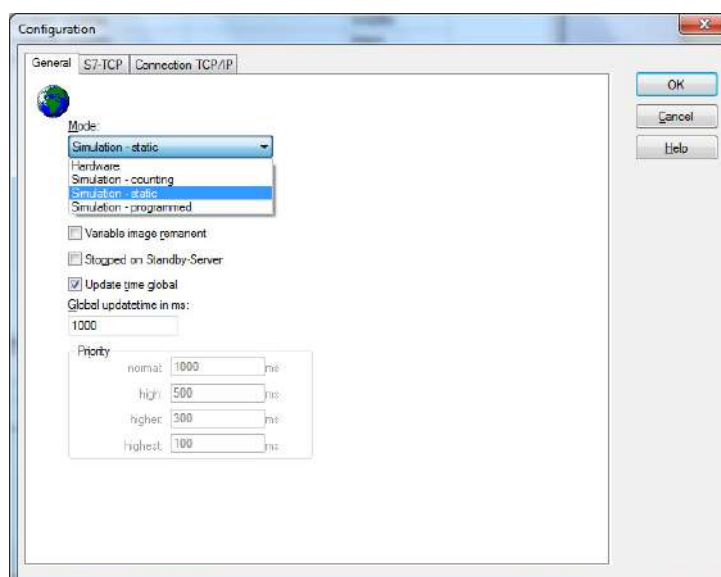
Автоматично відкриється вікно вибору драйверу:



- Знайдіть папку *Vipa* і виберіть драйвер *S7 TCP-IP*
- Перейменуйте його на '*VIPA 300*' і підтвердіть зроблений вибір кнопкою *OK*

*Перейменування драйверу може знадобитись тоді, коли Ви буде створювати мультитайверну систему з багатьма контролерами.*

Тепер відкриється діалогове вікно конфігурування драйверу:



- Переведіть *Mode* в *Simulation – static*

Якщо буде стояти режим *Hardware* після запуску Runtime система відразу спробує з'єднатися з PLC. Оскільки у нас фізично його немає, то в проєкті з'явиться помилка комунікації. Для нашого прикладу ми будемо працювати в стимуляційному режимі. В інших закладках налаштовуються комунікаційні параметри. Докладно про них можна знайти в мануалі на відповідний драйвер. При роботі в режимі симуляції жодних додаткових налаштувань робити не потрібно.

- Закінчіть конфігурування драйверу натиснувши кнопку *OK*

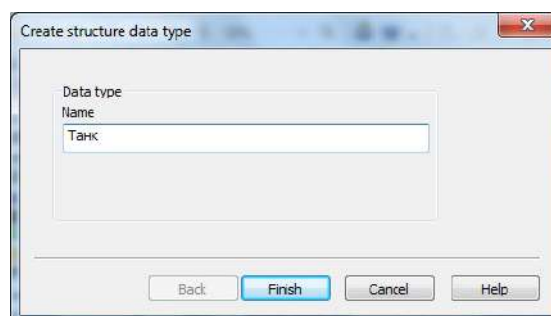
## ТИПИ ДАНИХ

Усі змінні повинні відповідати конкретним типам даних згідно до стандартів IEC. В zenon існує можливість використати існуючі типи або створити свій простий чи структурований тип даних.

В проєкті ми будемо використовувати існуючі типи даних, а також структурований, який ми зараз і створимо.

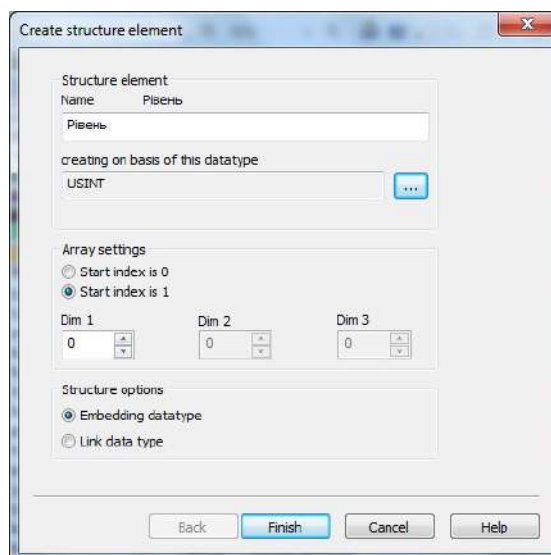
### Створення структурованого типу даних

- ▶ В менеджері проєкту зайдіть в вузол *Variables* і виберіть пункт *Datatypes*
- ▶ У вікні детального перегляду клікніть на іконці *New structure datatype*



- ▶ В якості імені нового структурованого типу задайте 'Танк'

Далі автоматично з'явиться вікно створення елементів структури



- ▶ Задайте ім'я 'Рівень', і як базовий тип даних виберіть *USINT*
- ▶ В полі опції повинно бути активно *Embedded*

*Якщо для елемента структури вибрано Embedded, то власності його типу даних можуть змінюватись незалежно від базового типу, наданого системою.*



- ▶ Кнопкою *Finish* закрийте це діалогове вікно

У вікні детального перегляду повинен з'явитись створений нами структурований тип даних. Даваймо додамо до нього ще декілька елементів.

- ▶ Правою кнопкою мишки активуйте контекстне меню структурованого типу 'Танк' і виберіть пункт *New structure element*

Далі аналогічним чином додайте до структури елемент 'Пуск\Стоп': BOOL та 'Клапан': BOOL, Dim1=4.

## ПРОСТІ ТА СТРУКТУРОВАНІ ЗМІННІ

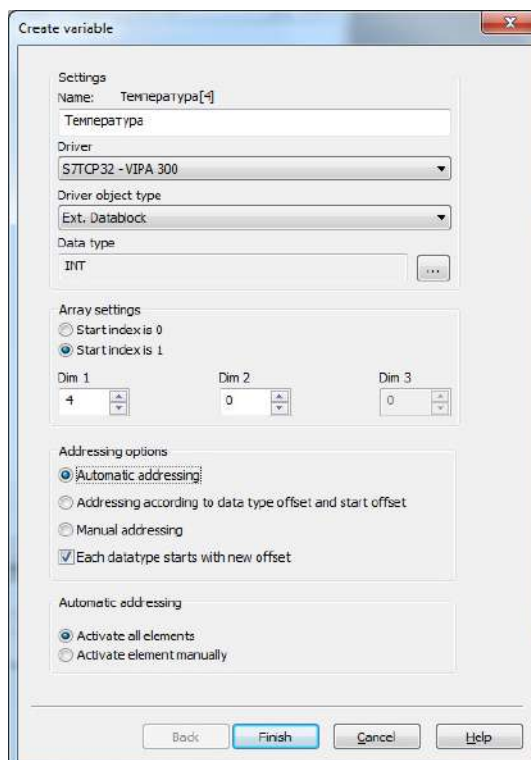
Процесні змінні являються з'єднуючою ланкою між джерелом даних (PLC, польова шина...) та системою керування. В даному розділі ми створимо декілька змінних, які базуються на простих та структурованих типах даних, скориствовавшись при цьому поняттям масивів.

*Багато драйверів системи керування дозволяють здійснювати прямий імпорт змінних з PLC-програми чи безпосередньо з самого контролера. Інформацію про можливість такого імпорту можна знайти в мануалі на відповідний драйвер.*

### Створення простої змінної

- ▶ В менеджері проекту виберіть *Variables*
- ▶ У вікні детального перегляду клікніть на іконці *Variable new*

Автоматично відкриється вікно створення змінної



- ▶ В якості імені задайте 'Температура'
- ▶ Драйвер *S7 TCP32 – VIPA 300*
- ▶ Об'єктний тип *Ext. Datablock*
- ▶ Тип даних виберіть *INT*

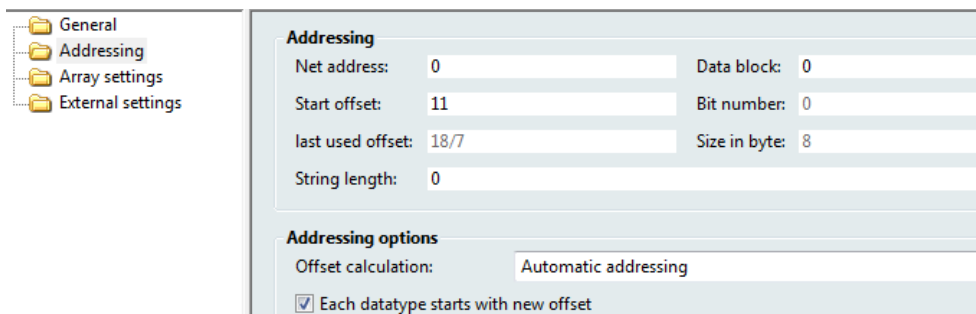
Оскільки ми будемо створювати масив змінних, нам необхідно зробити відповідні налаштування.

- ▶ В полі *Array settings* встановіть *Dim1* в 4
- ▶ Усі інші налаштування залиште як показано на малюнку вище

Після натиснення кнопки *Finish*, у вікні детального перегляду з'явиться створена нами перша проста масивна змінна, яка містить чотири змінні. Кожну з цих змінних можна налаштувати індивідуально.

Відповідно до встановлених налаштувань, система керування автоматично виконала розподіл адрес для усіх змінних масиву. Ми змінимо цю адресацію на свою власну.

- ▶ У властивостях масивної змінної 'Температура' зайдіть в розділ *Addressing* і змініть початкове зміщення адрес *Start offset* на 11



**Addressing**

Net address: 0 Data block: 0

Start offset: 11 Bit number: 0

last used offset: 18/7 Size in byte: 8

String length: 0

**Addressing options**

Offset calculation: Automatic addressing

☒ Each datatype starts with new offset

Таблиця адрес для елементів масиву тепер має наступний вигляд:

State	Name	Iden...	Measur...	Net a...	Offset	Driver	Data type	Dat
Filter text	Filter text	Filter...	Filter...	Filter...	Filter...	Filter text	Filter t...	Filter
-	Температура			0	0	S7TCP32 - VIPA 300	INT	
-	Температура[1]		°C	0	11	S7TCP32 - VIPA 300	INT	
-	Температура[2]		°C	0	13	S7TCP32 - VIPA 300	INT	
-	Температура[3]		°C	0	15	S7TCP32 - VIPA 300	INT	
-	Температура[4]		°C	0	17	S7TCP32 - VIPA 300	INT	

- ▶ В розділі *General* властивостей кожної змінної масиву в полі *Measuring unit* задайте одиницю виміру '°C'

Давайте виконаємо аналогічні дії, але вже з використанням структурованого типу даних.

### Створення структурованої змінної

- ▶ В менеджері проекту виберіть *Variables*
- ▶ У вікні детального перегляду клікніть на іконці *Variable new*
- ▶ Введіть ім'я 'Танк'
- ▶ В якості драйверу виберіть *Intern – Driver for internal variables*
- ▶ Виберіть створений нами структурований тип 'Танк'
- ▶ Об'єктний тип *Internal variable*
- ▶ В полі *Array settings* встановіть *Dim1* в 2

Після натиснення кнопки *Finish*, система запитає підтвердження на створення дванадцяти змінних. Таким чином, автоматично буде створено структурну змінну, яка містить готовий набір змінних для відображення параметрів двох танків.

Оскільки ми вибрали внутрішній драйвер, а не апаратний, то за розподілом адрес нам слідкувати не потрібно.

Там де необхідно, для змінних задайте одиниці вимірювання.

Ваш список змінних тепер повинен мати наступний вигляд:

State	Name	Iden...	Measur...	Net a...	Offset	Driver	Data type	Data b
	Filter text	Filter...	Filter...	Filter...	Filter...	Filter text	Filter text	Filter t
[-]	Температура			0	0	S7TCP32 - VIPA 300	INT	
[-]	Температура[1]		°C	0	11	S7TCP32 - VIPA 300	INT	
[-]	Температура[2]		°C	0	13	S7TCP32 - VIPA 300	INT	
[-]	Температура[3]		°C	0	15	S7TCP32 - VIPA 300	INT	
[-]	Температура[4]		°C	0	17	S7TCP32 - VIPA 300	INT	
[-]	Танк			0	0	Intern - Driver for int...	Танк	
[-]	Танк[1].Рівень		м	0	0	Intern - Driver for int...	USINT/<embedded> 1	
[-]	Танк[1].Пуск\Стоп			0	0	Intern - Driver for int...	BOOL/<embedded> 1	
[-]	Танк[1].Клапан[1]			0	0	Intern - Driver for int...	BOOL/<embedded> 2	
[-]	Танк[1].Клапан[2]			0	0	Intern - Driver for int...	BOOL/<embedded> 2	
[-]	Танк[1].Клапан[3]			0	0	Intern - Driver for int...	BOOL/<embedded> 2	
[-]	Танк[1].Клапан[4]			0	0	Intern - Driver for int...	BOOL/<embedded> 2	
[-]	Танк[2].Рівень		м	0	0	Intern - Driver for int...	USINT/<embedded> 1	
[-]	Танк[2].Пуск\Стоп			0	0	Intern - Driver for int...	BOOL/<embedded> 1	
[-]	Танк[2].Клапан[1]			0	0	Intern - Driver for int...	BOOL/<embedded> 2	
[-]	Танк[2].Клапан[2]			0	0	Intern - Driver for int...	BOOL/<embedded> 2	
[-]	Танк[2].Клапан[3]			0	0	Intern - Driver for int...	BOOL/<embedded> 2	
[-]	Танк[2].Клапан[4]			0	0	Intern - Driver for int...	BOOL/<embedded> 2	

18 total / 18 filtered / 0 selected 4 tags used / unlimited tags available

На цьому, наше знайомство зі створенням змінних можна вважати завершеним.

## Реалізація дисплейної мнемосхеми процесу

Ми дійшли до того моменту, коли усе готово для реалізації повноцінної мнемосхеми представлення техпроцесу: розміщення елементів відображення, символів обладнання та прив'язки до них змінних. В цьому навчальному посібнику ми розглянемо приклад реалізації фрагменту дифузійного відділення цукрового заводу. Основними елементами такого процесу є танки, клапани, двигуни, з'єднувальні труби.

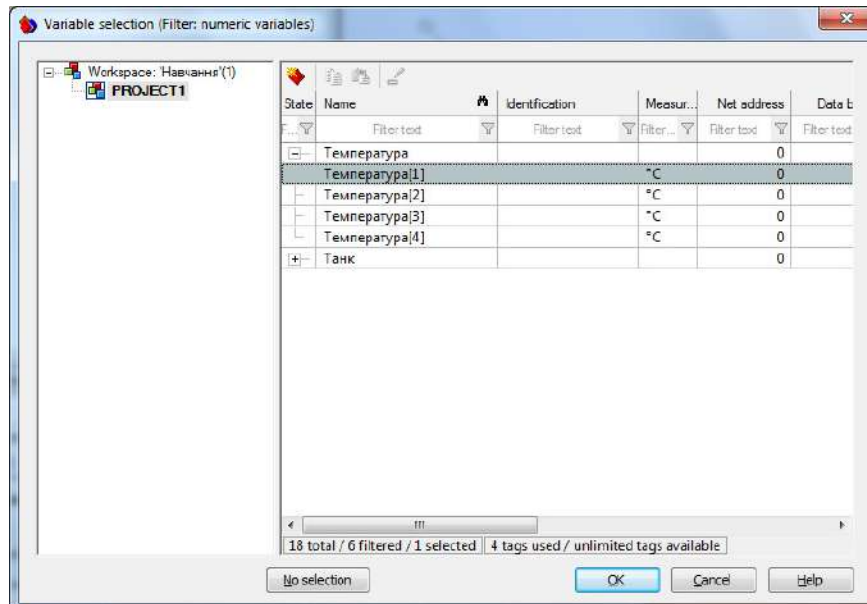
### ВІДОБРАЖЕННЯ ЗНАЧЕНЬ ЗМІННИХ

Відображення вимірюваних параметрів здійснюється через всілякі показуючі динамічні елементи. Зараз ми декілька з них розглянемо.

#### Цифровий показчик

- ▶ Відкрийте екран 'Процес'
- ▶ На інструментальній панелі елементів виберіть піктограму *Numerical value* і розмістіть його під вже існуючим текстовим елементом

Автоматично відкриється діалогове вікно вибору змінної

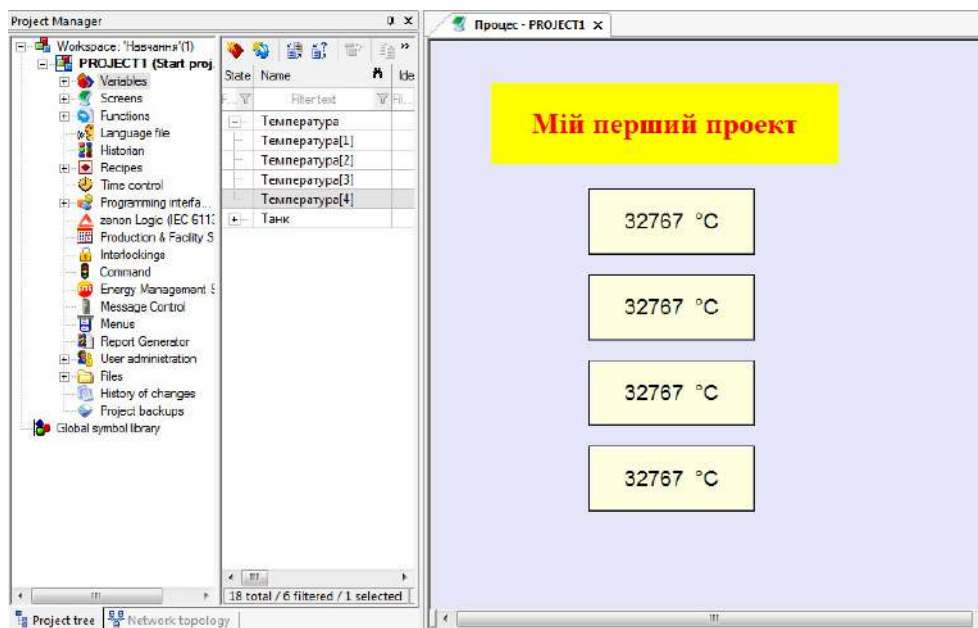


- ▶ Виберіть змінну 'Температура[1]' і натисніть кнопку **OK**
- ▶ Підлаштуйте розміри елементу та його розміщення на екрані
- ▶ Змініть колір фону елементу та шрифт
- ▶ У властивостях елементу Representation -> *Element border* виберіть *Edge deepened*

Аналогічним чином створіть елементи відображення для змінних 'Температура[2]', 'Температура[3]', 'Температура[4]'.

*Найпростіше це зробити шляхом копіювання вже існуючого елемента та перетягуванням на нього відповідної змінної (Drag&Drop).*

Наш головний екран процесу повинен мати наступний вигляд:



*Для полегшення чіткості позиціонування елементів на екрані та один відносно одного, можна ввімкнути координатну сітку та магнітні точки прив'язки: головне меню Options -> Settings.*

Давайте адаптуємо діапазон представлення температури до 0 – 150 °C

- ▶ У властивостях змінної 'Температура[1]' зайдіть в розділ *Value calculation -> Value adjustment linear*
- ▶ Виставте для мінімального значення діапазону '0', а для максимального '150'

Таким чином, від контролера ми отримуємо значення в діапазоні -32768 - 32767, а на дисплейній мнемосхемі будемо відображати його в діапазоні 0 - 150. Аналогічно зробіть і для інших змінних температур.

#### Стрілочний показник

- ▶ На інструментальній панелі елементів виберіть піктограму *Pointer instrument* і розмістіть його під існуючими елементами
- ▶ В наступному вікні виберіть змінну 'Температура[1]'

- ▶ Підлаштуйте розміри елементу, його колір, кількість рисок шкали, кут повороту шкали

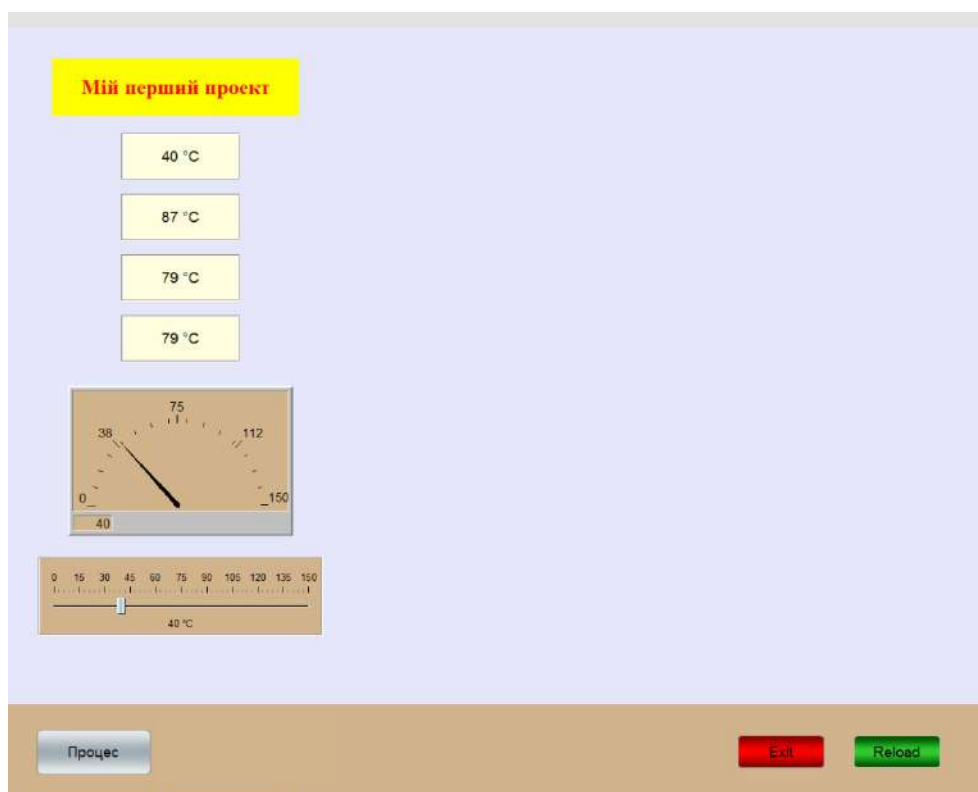
### Універсальний повзунок

- ▶ На інструментальній панелі елементів виберіть піктограму *Universal slider* і розмістіть його під існуючими елементами
- ▶ В наступному вікні виберіть змінну 'Температура[1]'

Автоматично відкриється помічник конфігурування цього елементу. Ми нічого змінювати не будемо, тому просто на кожному кроці натискайте *Next* і завершіть створення елементу кнопкою *Finish*.

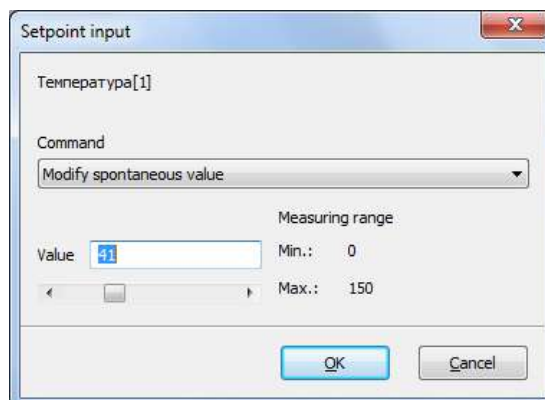
- ▶ Підлаштуйте розміри елементу, його колір, стиль кромки

Для перевірки зробленої роботи запустіть Runtime. Ваш проект повинен мати наступний вигляд:



Спробуйте для кожної змінної змінити значення.

При зміні значення через цифрові елементи чи стрілочний показчик буде з'являтися вікно вводу значення *Setpoint input*.



Давайте спробуємо інші варіанти вводу значень.

### Варіанти вводу значень

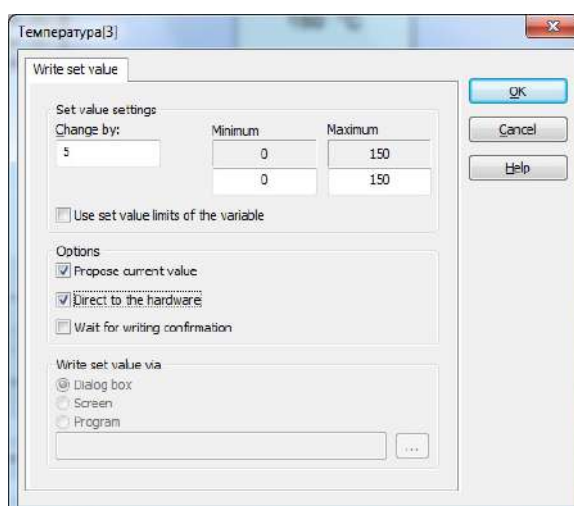
Поверніться в Editor не закриваючи при цьому Runtime (Alt+Tab).

- Виділіть другий цифровий показник і в розділі *Write set value* зі спадного боксу *Write set value via* змініть *Dialogbox* на *Element*

Встановлення значень для цифрового показника 3 ми будемо виконувати через кнопки додавання та віднімання значення. Для цього створимо відповідні функції та прив'яжемо їх до кнопок.

- Створіть нову функцію *Write set value* і в діалоговому вікні прив'яжіть її до змінної 'Температура[3]'

Автоматично з'явиться діалогове вікно налаштування функції



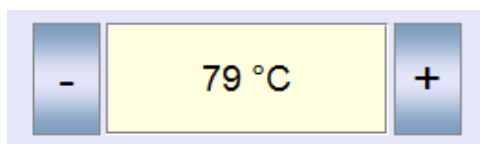
- Встановіть *Change by* в '5' та активуйте бокси *Propose current value* та *Direct to the hardware*
- Перейменуйте створену функцію на '+5 Температура3'



- ▶ Розмістіть на екрані справа від цифрового показчика нову кнопку
- ▶ Прив'яжіть до неї створену функцію '+5 Температура3' і підпишіть символом '+'

Аналогічним чином створіть функцію '-5 Температура3' і кнопку '-'. Підкорегуйте розміри кнопок, шрифт підпису, колір, приберіть закруглення кутів.

Створений фрагмент повинен мати наступний вигляд:



- ▶ Виділіть цифровий показчик 3, 4 та стрілочний показчик і у властивості *Write set value* деактивуйте галочку в боксі *Setting values active*

Таким чином, для першого цифрового показчика значення будуть вводиться як і раніше – через діалогове вікно, для другого – безпосередньо в елементі, для третього – через кнопки додавання\віднімання а для всіх інших показчиків – взагалі задаватись не будуть.

Тепер скопіюйте зроблені зміни. Для цього на верхній інструментальній панелі натисніть на піктограму *Create changed Runtime files*. Поверніться в Runtime і натисніть на зроблену нами на початку кнопку 'Reload'. Виконані в проекті зміни активуються без перезапуску середовище виконання проекту.

Попрактикуйтесь вводити значення різними способами.

## СИМВОЛИ ТА ЕКРАННІ ЕЛЕМЕНТИ

Давайте створимо частину технологічної структури на базі ємності – танку збору дифузійного соку. Ми не будемо самостійно малювати танк, а візьмемо готовий з глобальної бібліотеки символів.

### Танк

- ▶ Виберіть в самому низу дерева проекту *Global symbol library*

Тут усі символи згруповані по папкам згідно їх призначення

- ▶ Відкрийте папку *Tank* і виберіть символ *simple tank*

Для розміщення символу на екрані його необхідно просто перетягнути на необхідне місце.

- ▶ Збільшіть розмір символу

Тепер нам необхідно показувати рівень наповнення танку. Для цього ми використаємо стовпчиковий показуючий елемент.

- ▶ З інструментального меню виберіть елемент *Bar display*
- ▶ Розмістіть його уздовж бічної стінки танку
- ▶ У вікні прив'язки змінної вкажіть *Танк[1].Рівень*
- ▶ Виберіть шрифт відображення шкали, колір стовпчика, та активуйте опцію *Color -> Transparent*
- ▶ Над стовпчиковим елементом розмістіть ще один цифровий показчик, прив'яжіть змінну рівня та зробіть щоб цей елемент також був прозорим

Таким чином, ми будемо точно бачити актуальне значення рівня в цифровому виді.

На базі цих елементів створимо свій власний символ.

- ▶ Виділіть мишкою усі елементи і в контекстному меню виберіть *Symbol -> Create embedded symbol*

Тепер три елементи представляють єдиний символ. Додамо його в бібліотеку символів нашого проекту.

- ▶ В контекстному меню символу виберіть *Symbol -> Insert into symbol library*
- ▶ Виберіть *Project library (Project1)*

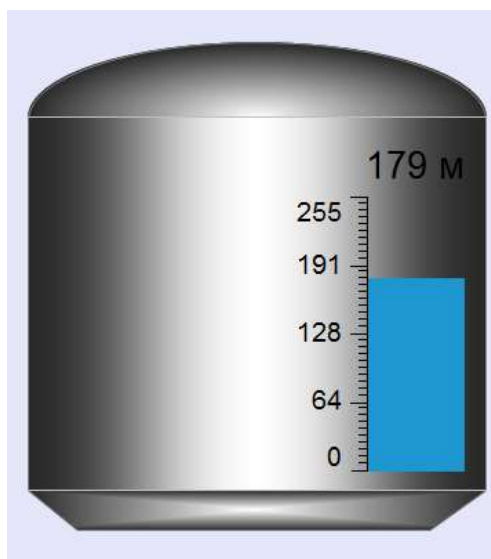
Створений символ з'явиться в дереві проекту *Screens -> Project symbol library*. В цьому розділі зберігаються символи, які стосуються тільки конкретно Вашого проекту. На даний момент тут повинний бути лише один символ: танк з показниками.

- ▶ Перейменуйте символ в 'Танк'.

Ми можемо усюди, де потрібно в проекті, використовувати цей символ. Перетягніть його на екран, а той символ що був – видаліть. При перетягуванні автоматично з'явиться вікно підстановки змінної. Нічого тут змінювати не потрібно. Таким чином, в проекті ми будемо використовувати символи з нашої бібліотеки символів проекту.

В правому верхньому куті символу повинна бути стрілочка, яка показує, що він являється зв'язаним з вихідним. Будь-які зміни в вихідному "батьківському" символі будуть автоматично відтворюватись в усіх "дочірніх" символах на екранах проекту.

Скомпілюйте зроблені зміни та активуйте їх в Runtime. Ваш танк повинен мати наступний вигляд:



Як бачите, програма показує нам значення в сотнях метрів. Нам необхідно адаптувати покази до реалій – перенести кому на дві позиції вліво.

- ▶ Зайдіть в розділ *Variables* дерева проекту і виберіть змінну 'Танк[1].Рівень'
- ▶ У властивостях *Value calculation* -> *Decimals* встановіть значення '2'
- ▶ Також тут для *Measuring range min\max* необхідно встановити '0.00' та '2.55' відповідно

Тепер значення рівня повинні відображатися коректно.

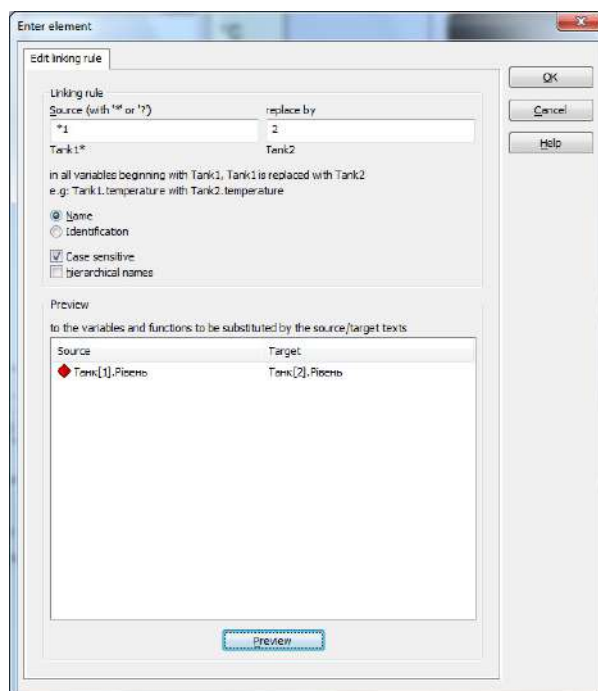
### Підстановка змінних

Давайте створимо ще один танк на базі нашого символу.

- ▶ Зайдіть в дереві проекту *Screens* -> *Project symbol library*
- ▶ Вберіть символ 'Танк' і перетягніть його на екран, поруч з уже існуючим танком

Автоматично з'явиться вікно підстановки змінних, за допомогою якого ми замінимо усі зв'язки символу. Для цього ми будемо використовувати маску підстановки.

- ▶ В полі *Linking rule* для опції *Source* введіть маску '\*1'
- ▶ В полі *Linking rule* для опції *replace by* введіть маску '2'



- Натисніть кнопку *Preview*

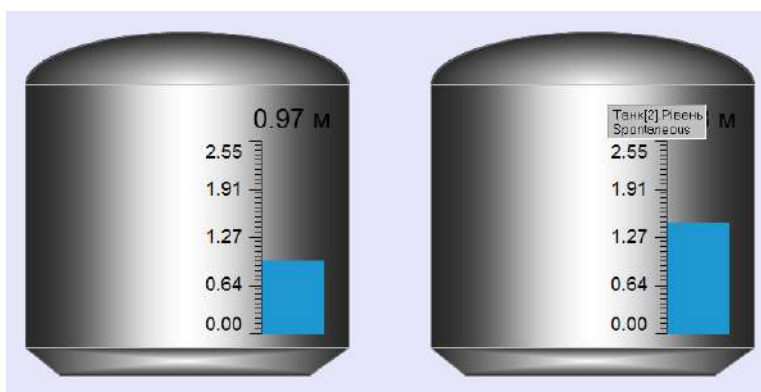
При цьому, у вікні *Preview* змінна 'Танк[1].Рівень' заміниться на 'Танк[2].Рівень'.

*Система підстановки дозволяє в якості маски використовувати символ '\*' – для пропуску фрази, і символ '?' – для пропуску літери. Таким чином можна формувати складні конструкції заміни.*

- Натиснувши кнопку *OK* завершіть конфігурування символу

Адаптуйте діапазон змінної 'Танк[2].Рівень' аналогічно до попередньої.

Перевірте в Runtime коректність виконаних підстановок. Для цього натисніть правою кнопкою мишки на динамічному елементі і тримайте. Повинна з'явитись інформація про прив'язану змінну та її статус.



Окрім виміряного значення та часової мітки система також надає інформацію відносно статусу змінної. Всього існує 64 статуси, серед них: спонтанне, недійсне, альтернативне, ревізія, користувацький... Докладно про призначення кожного статусу можна прочитати в мануалі *'Variables'*.

Також можна перевірити який екран зараз відкритий. Для цього необхідно правою кнопкою мишки натиснути на пустому полі екрану. Через декілька секунд з'явиться поле з ім'ям екрану.

## Клапан

Тепер нам необхідно до нашого танку додати зливний клапан.

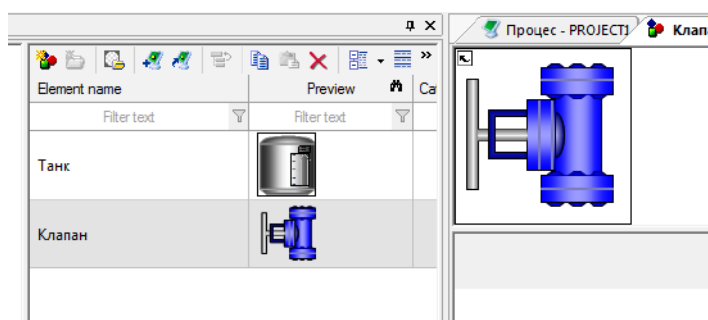
- ▶ Виберіть в глобальній бібліотеці символів з папки *Valves* символ *manual valve 1* і розмістіть під танком
- ▶ Через властивості *Position* -> *Rotation angle* поверніть його на 90°

Додамо символ клапану в бібліотеку символів нашого проекту.

- ▶ Виділіть символ і в контекстному меню виберіть *Symbol* -> *Create embedded symbol*
- ▶ Повторно зайдіть в контекстне меню символу і виберіть *Symbol* -> *Insert into symbol library*
- ▶ Виберіть *Project library (Project1)*

Створений символ з'явиться в локальній бібліотеці проект *Screens* -> *Project symbol library*.

- ▶ Переименуйте символ в 'Клапан'.



Перетягніть створений символ на екран, а той символ що був – видаліть. Таким чином, ми маємо клапан для першого танку. Скопіюйте символ клапану для другого танку.

Як ми згадували раніше, зміни в вихідному "батьківському" символі будуть автоматично відтворюватись в усіх "дочірніх" символах. Давайте перевіримо як це працює.

### Внесення змін в символ

Давайте додамо до символу елемент керування *Switch*, щоб оператор міг вручну вмикати і вимикати клапан.

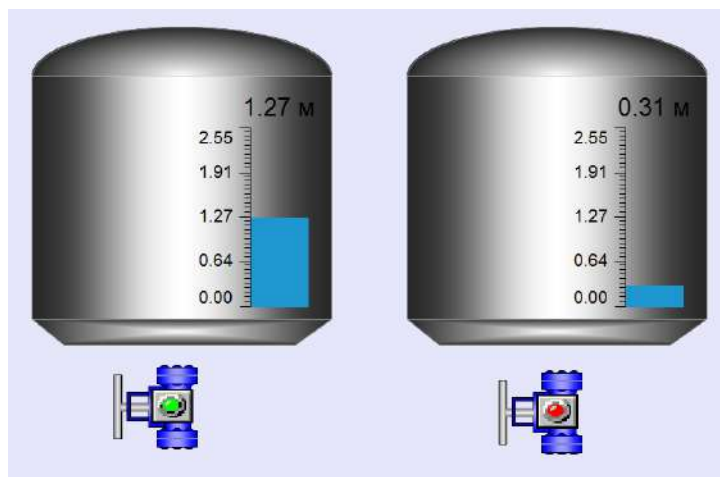
- ▶ В дереві проекту зайдіть в *Screens* -> *Project symbol library* і відкрийте символ 'Клапан' для редагування
- ▶ З інструментальної панелі виберіть елемент керування *Switch* і розмістіть його по центру клапану
- ▶ У вікні прив'язки змінної вкажіть 'Танк[1].Клапан[1]'
- ▶ Зайдіть у властивості цього елементу *Representation* -> *Predefined graphics* і виберіть представлення у вигляді кругової кнопки що змінює колір
- ▶ Підкорегуйте розміри та взаємне розміщення елементів і збережіть зроблені в символі зміни

Після цих дій на екрані обидва наші клапани змінять своє представлення, доповнившись кнопкою керування. Таким чином декількома кліками мишки можна централізовано керувати усіма символами проекту.

Нам залишилось лише переприсвоїти змінну до нашого другого клапану.

- ▶ На екрані 'Процес' виділіть клапан під другим танком
- ▶ У властивості *Linking rule* клікніть в полі *Preview*
- ▶ У вікні підстановки задайте маску '\*1' для вихідного значення і '2' для кінцевого
- ▶ Закрийте діалогове вікно підстановки кнопкою *OK*

На цьому робота по створенню клапанів завершена. Скопіюйте проект та перевірте його в Runtime. Керування клапанами повинне здійснюватись незалежно один від одного. Наша технологічна структура тепер повинна мати такий вигляд:



## Трубопроводи

Для відображення трубопроводів система керування має спеціальний елемент – *Pipe*.

- ▶ На інструментальній панелі виберіть елемент *Pipe* і намалуйте мережу трубопроводів

Зробіть так, щоб трубопроводи виходили з танків. Проходили через клапани, з'єднувались і виходили вправо.

- ▶ Виділіть трубопровід і у властивостях *Representation* -> *Line width* поміняйте товщину труби на 30
- ▶ Зайдіть в контекстне меню трубопроводу і виберіть *Element position* -> *Background* чи просто натисніть комбінацію 'Ctrl + -'

Аналогічно розмістіть вхідні трубопроводи в танки, але підключайте їх не по центру ємностей, а зліва, залишивши місце для мішалок, які ми створимо пізніше.

Підключення до танку вхідного трубопроводу виконаємо через фланці, які ми зробимо самостійно. Для цього ми використаємо елемент *Rectangle*.

- ▶ В бібліотеці символів проекту відкрийте символ 'Танк' для редагування
- ▶ На інструментальній панелі виберіть статичний елемент *Rectangle*
- ▶ Розмістіть декілька прямокутників утворивши фланець
- ▶ Для прямокутників задайте колір у вигляді градієнта чорного кольору і розмістіть їх на задньому фоні символу танку

Утворений фрагмент символу повинен мати наступний вигляд:



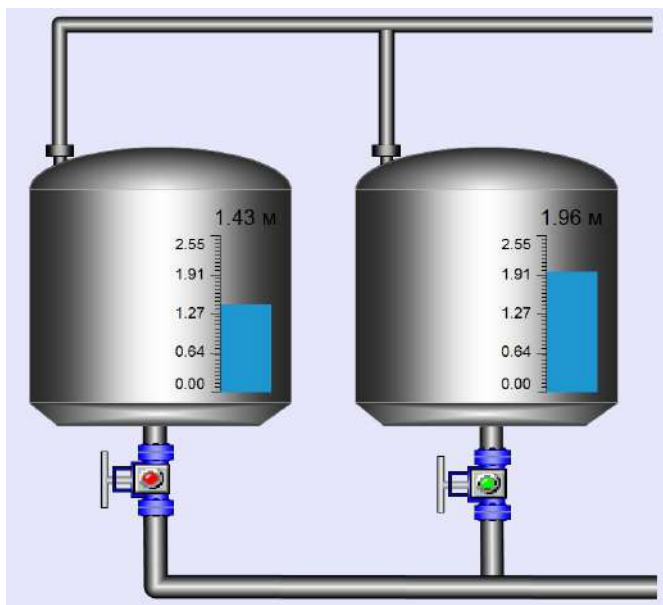
- ▶ Збережіть зміни зроблені в символі

Таким чином, Ви можете зі статичних елементів самостійно формувати частини символів чи цілі символи.

На екрані усі танки автоматично повинні змінити своє представлення, доповнивши його фланцем.



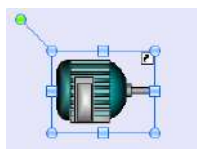
Тепер ми маємо наступну апаратну структуру:



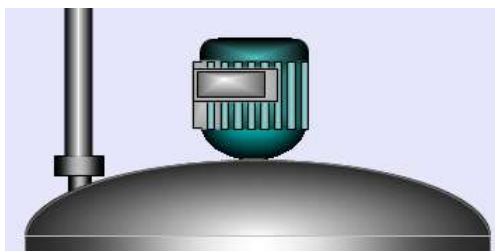
### Мішалка

На завершальному етапі побудови технологічної структури до кожного танку зверху ми додамо мішалку. Для цього використаємо символ двигуна.

- ▶ Виберіть в глобальній бібліотеці з папки *Engines* символ *engine 4* і розмістіть його над танком



- ▶ Через властивості *Position* -> *Rotation angle* поверніть його на -90°
- ▶ Розмістіть символ над танком, підкорегуйте його розміри та переведіть на задній план



- ▶ Скопіюйте зроблений символ і розмістіть його над другим танком

## Кнопки

Для керування мішалкою ми створимо спеціальну кнопку 'Пуск\Стоп' і прив'яжемо до неї відповідну змінну.

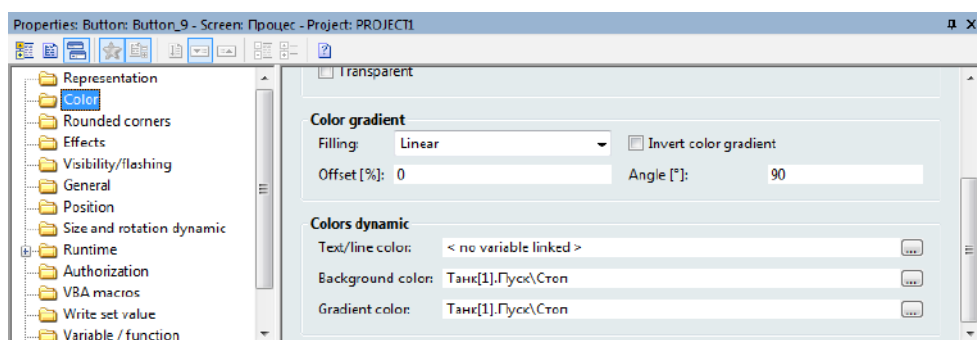
Давайте розглянемо декілька варіантів реалізації кнопок – на базі елементу *Button* та *Combined element*.

Для ввімкнення мішалки першого танку застосуємо звичайний елемент кнопки, який ми використовували для перемикання між екранами.

- ▶ На панелі інструментів виберіть елемент *Button* і розмістіть його на вільному місці над символом двигуна мішалки першого танку

Автоматично з'явиться вікно підстановки функції. Оскільки цю кнопку ми будемо використовувати для керування змінною, а не функцією, то просто закрийте діалогове вікно кнопкою *Cancel*.

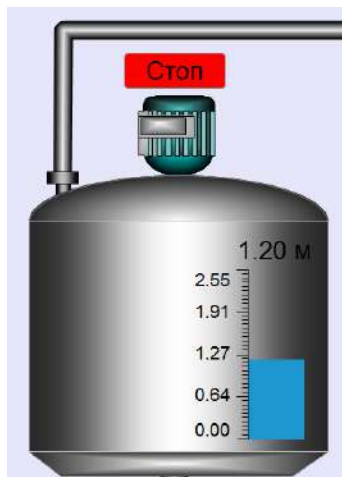
- ▶ У вікні властивостей кнопки зайдіть в розділ *Write set value* і активуйте бокс *Switch*
- ▶ Зайдіть в розділ *Variable/function* і в полі *Variable* прив'яжіть змінну 'Танк[1].Пуск\Стоп'
- ▶ Зайдіть в розділ *Representation* і в полі *Text line 1* підпишіть 'Стоп', в полі *Text when pressed* підпишіть 'Пуск'
- ▶ Змініть шрифт на 5 - *Standard font5 (Arial 20)*
- ▶ Зайдіть в розділ *Color -> Colors dynamic* і для полів *Background color* і *Gradient color* прив'яжіть змінну 'Танк[1].Пуск\Стоп'



Наша кнопка буде клікабельно встановлювати значення змінної в 0\1 та буде змінювати при цьому свій підпис і колір. Колір ми взяли з лімітів, які по замовчуванню визначаються системою для булівських змінних. При бажанні тут можна виставити будь-який колір. Докладніше з лімітами ми познайомимось пізніше.

Візуальне відображення кнопки також можна змінювати за рахунок прив'язки графічних файлів *Representation* -> *Graphics file*. При цьому, картинки необхідно попередньо завантажити в дереві проекту *Files* -> *Graphics*.

Скомпілюйте зроблені зміни і перевірте роботу кнопки в Runtime.

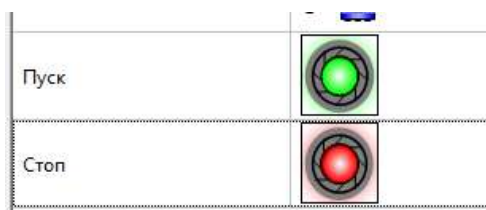


Давайте розглянемо варіант реалізації елемента керування на базі комбоелемента. Його ми використаємо для ввімкнення двигуна мішалки другого танку.

*Комбоелемент (Combined element) – це комплексний інструмент керування, який дозволяє присвоювати елементу різне візуальне представлення залежно від побітового значення чи статусу змінної. Налаштування здійснюється за допомогою спеціального майстра та конфігураційного вікна.*

Оскільки комбоелемент базується на символах, нам попередньо необхідно створити символи для відображення стану 'Пуск' і 'Стоп', після чого додати їх до проектною бібліотеки символів.

- ▶ Виберіть в глобальній бібліотеці з папки *interface\_buttons\_16\_9* символи *shutter\_red\_2* і *shutter\_green\_2*
- ▶ Додайте їх до проектною бібліотеки символів і назвіть 'Пуск' і 'Стоп' відповідно

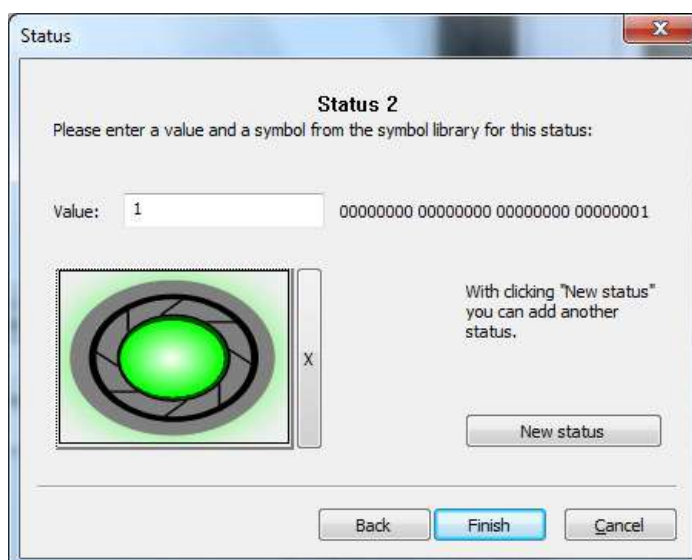


Тепер перейдемо безпосередньо до створення комбоелемента

- ▶ На панелі інструментів виберіть елемент *Combined element* і розмістіть його на вільному місці над символом двигуна мішалки другого танку
- ▶ Прив'яжіть змінну 'Танк[2].Пуск\Стоп'

Автоматично з'явиться майстер конфігурування цього елементу.

- ▶ Для вікна *Type of display* повинно стояти *Symbol from library*
- ▶ У вікні *Status* натисніть на великий квадрат і виберіть символ 'Стоп'
- ▶ Натисніть кнопку *New status* і в полі *Value* задайте '0'
- ▶ Знову натисніть кнопку *New status* і в полі *Value* задайте '1'
- ▶ Для цього статусу змініть візуальне представлення на символ 'Пуск'



- ▶ Завершіть конфігурування елементу кнопкою *Finish*

Отже, ми створили елемент який має три статуси: 0, 1 і *Default*. Кожен з цих статусів має свій символ відображення.

- ▶ Підкорегуйте розміри створеного елементу – кнопки

Більш складні налаштування цього елементу здійснюються у вікні конфігурування.

- ▶ У властивостях елементу *Representation* зайдіть в опцію *Configuration and test*

Тут можна не тільки визначати побітові символи, а й формульні значення і статуси.



## Зображення

Система керування дозволяє на екранах відображати як растрову графіку (bmp, jpg, gif, png...) так і векторну (wmf). Зображення можна ставити як фон для екрану або як окремий елемент.

Давайте в верхньому куті екрану розмістимо логотип 'zenon' Його легко знайти в пошуковій системі Google, задавши ключ пошуку 'logo scada zenon'.

Скачайте картинку та завантажте її в проект.

- ▶ В дереві проекту зайдіть в *Files -> Graphics*
- ▶ У вікні деталізованого перегляду виберіть піктограму *Import file*
- ▶ Завантажте відповідне зображення

*По можливості, зображення повинні займати не великий об'єм пам'яті, оскільки від цього залежить швидкість відкриття екранів проекту.*

Для показу картинки використовується елемент *Button*, але без прив'язки функції чи змінної.

- ▶ На інструментальній панелі виберіть елемент *Button* і розмістіть на вільному місці екрану
- ▶ Якщо з'явиться вікно прив'язки функції чи змінної, просто закрийте його кнопкою *Cancel*
- ▶ У властивостях елементу *Representation* зайдіть в опцію *Graphics file* і прив'яжіть відповідну картинку логотипу
- ▶ В полі *Graphic size* повинно стояти *Stretched*
- ▶ У властивості *Rounded corners* поставте закруглення кутів в '0'

Зроблений елемент повинен мати такий вигляд:



Аналогічним чином Ви можете розмістити на екрані будь-яке зображення чи анімацію.

## Годинник

Кожний повноцінний проект повинен відображувати оператору поточний час. Для цього система керування має окремий елемент – Clock.

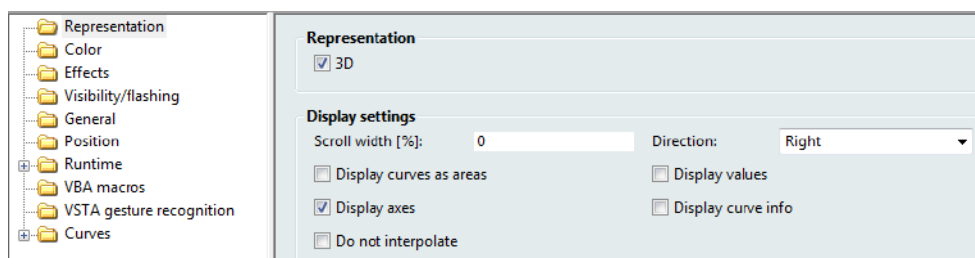
- ▶ На інструментальній панелі виберіть елемент *Clock* і розмістіть на вільному місці екрану поруч з логотипом
- ▶ У властивостях елементу *Representation* виберіть найбільший шрифт та формат виставте в показ лише часу
- ▶ У властивостях елементу *Color* активуйте бокс *Transparent*



## Трендовий елемент

Система керування має у своєму розпорядженні спеціальний елемент для відображення виміряної інформації у вигляді графіків. Цей трендовий елемент має досить низьку функціональність і використовується тільки для приблизного відображення графіків з невеликою точністю. Для повноцінного відображення графіків в zenon існує спеціальний тип екрану – *Extended Trend*. З ним ми познайомимось пізніше.

- ▶ На інструментальній панелі виберіть елемент *Trend element*
- ▶ Розмістіть його в нижній частині екрану
- ▶ У вікні що з'явилося прив'яжіть до нього змінну температури і рівнів двох баків
- ▶ У властивостях представлення зробіть налаштування, як показано на малюнку:



- ▶ У властивостях *Curves* для кожної кривої задайте свій колір, наприклад, чорний, червоний та зелений

Для того, щоб наші криві не зникали при перемиканні між екранами проекту, необхідно виміряні значення зберігати на жорсткий диск.

- ▶ Зайдіть у властивість змінної 'Температура[1]'

- В розділі *Harddisk data storage* активуйте бокс *Harddisk data storage active* і для *Number of values* встановіть 300

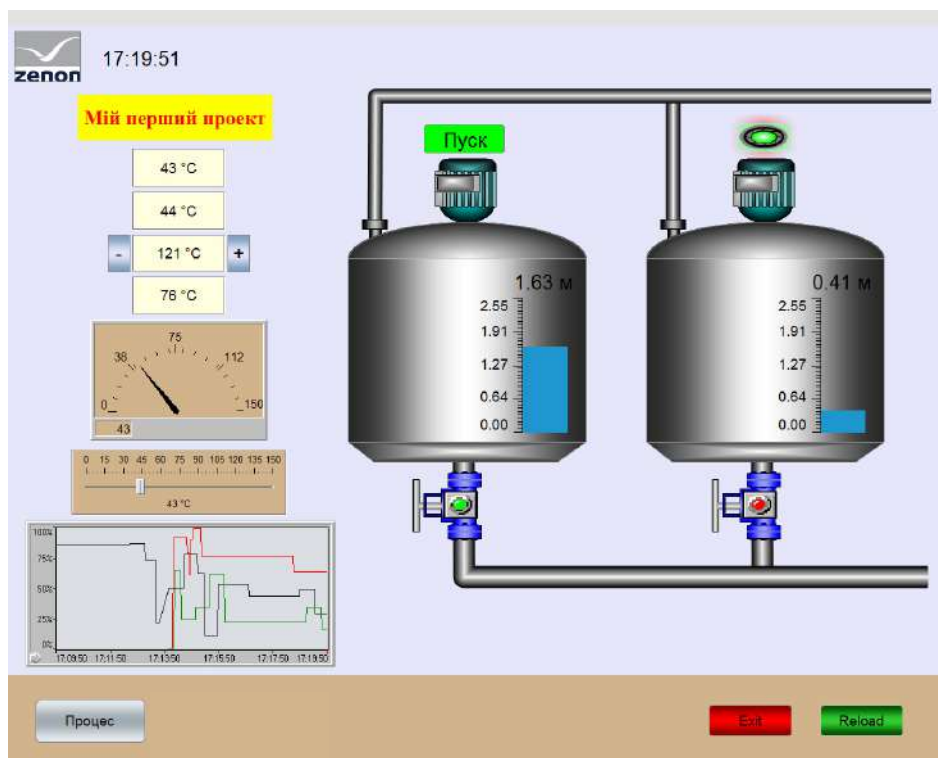
Таким чином, наш трендовий елемент буде відображати 300 циклів запису значень змінної. Цього нам цілком достатньо.

Виконайте аналогічні операції для змінних 'Танк[1].Рівень' і 'Танк[2].Рівень'.

Скомпілюйте зроблені зміни і перезапустіть Runtime. Щоб зміни по запису значень вступили в силу не достатньо виконати функцію *Reload*. Необхідно вимкнути Runtime і запустити його заново.

Виконайте модифікацію значень відповідних змінних і перевірте їхнє відображення в трендовому елементі.

Екран процесу тепер повинен мати такий вигляд:



## ДИНАМІЗАЦІЯ ОБ'ЄКТІВ

При розробці проектів візуалізації не обмежуються лише розміщенням на екрані елементів технологічних ліній та засобів керування. Розробнику проекту необхідно створити інтуїтивно зрозумілий інтерфейс взаємодії оператора з процесом. Це досягається за рахунок динамізації об'єктів. Під час здійснення тієї чи іншої дії оператор повинен однозначно розуміти що він зробив і яким був відгук системи на цю дію.



Одним з проявів динамізації, який ми вже використали, являється зміна кольору елементу керування. Давайте реалізуємо ще деякі можливості, які надає розробнику система керування.

### Динамічні кольори

Для наших мішалок зробимо так, щоб під час ввімкнення\вимкнення вони змінювали свій колір.

- ▶ Виберіть символ першої мішалки і у властивостях *Color* -> *Background color* призначте тут змінну 'Танк[1].Пуск\Стоп'
- ▶ Для символу другої мішалки виконайте аналогічні операції з прив'язкою тут змінної 'Танк[2].Пуск\Стоп'

В лімітах змінної за замовчуванням закладена зміна кольору, тому при керуванні мішалками їх кольори будуть змінюватись з червоного на зелений.

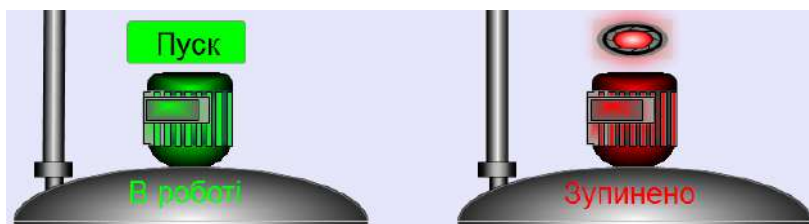
### Динамічний текст

Під час виконання тієї чи іншої операції система повинна інформувати оператора про зроблену дію. Для цього широко використовують динамічний текст.

- ▶ На інструментальній панелі виберіть елемент *Dynamic text* і розмістіть його під мішалкою на фоні першого танку
- ▶ Автоматично з'явиться вікно прив'язки змінної, в якому виберіть 'Танк[1].Пуск\Стоп'
- ▶ Зайдіть у властивості цієї змінної, розділ *Limits*
- ▶ Для першого ліміту, який означає значення '0', в полі *Limit text* введіть текст повідомлення 'Зупинено'
- ▶ Для другого ліміту, який означає значення '1', в полі *Limit text* введіть текст повідомлення 'В роботі'
- ▶ У властивостях елементу *Dynamic text* зайдіть в розділ *Representation* і виберіть 'Стандартний шрифт 5'
- ▶ Далі зайдіть у властивості *Color* і активуйте бокс *Transparent*

Для другої мішалки виконайте аналогічні операції, тільки до елементу динамічного тексту прив'яжіть змінну 'Танк[2].Пуск\Стоп'. Можна просто скопіювати елемент та перетягнути на нього цю змінну.

Наші елементи тепер мають наступний вигляд:



### Невидима кнопка

Керування таким об'єктом як мішалка, можна реалізувати не створюючи додаткових візуальних елементів.

- ▶ На інструментальній панелі виберіть елемент *Button* і розмістіть перед мішалкою першого танку
- ▶ Якщо з'явиться вікно прив'язки функції, просто закрийте його кнопкою *Cancel*
- ▶ В розділі властивостей елементу *Write set value* активуйте бокс *Switch*
- ▶ В розділі *Variable / function* призначте до елементу змінну 'Танк[1].Пуск\Стоп'
- ▶ В розділі *Color* активуйте бокс *Transparent*

Таким чином цей елемент керування стане прозорим і оператор матиме можливість керувати мішалкою безпосередньо через її символ.

Прозорі кнопки можна використовувати не тільки для зміни значень змінних, а й для виконання функцій.

### Діалогове вікно

Існують задачі, коли виконання керуючої дії повинно здійснюватись не відразу, а через діалогове вікно. Давайте створимо таке вікно та прив'яжемо його виклик до зливного клапану другого танку.

Для виконання цієї задачі ми використаємо усі знання та вміння які маємо на цей момент. Також тут розглянемо новий елемент – комбобокс, та використаємо функцію видимості екранних символів.

Для початку створимо екран діалогового вікна на базі нового фрейму.

- ▶ В менеджері проекту відкрийте вузол *Screens* і виберіть пункт *Frames*
- ▶ Створіть новий фрейм 'Клапан' розміром 280x170
- ▶ Проконтролюйте, щоб в розділі властивостей *Position* прив'язка позиції цього екрану відбувалась до елементу через який ми його викликаємо
- ▶ Усі інші налаштування фрейму залиште без змін
- ▶ В менеджері проекту виберіть вузол *Screens* і створіть новий екран на базі фрейму 'Клапан'
- ▶ У властивостях перейменуйте створений екран на 'Клапан' та задайте довільний колір фону
- ▶ Перейдіть до розділу властивостей *Window style* і активуйте бокс *Modal dialog*

*Модальне представлення екрану забезпечує неможливість виконання жодних операцій поза межами цього діалогового вікна до моменту його закриття.*

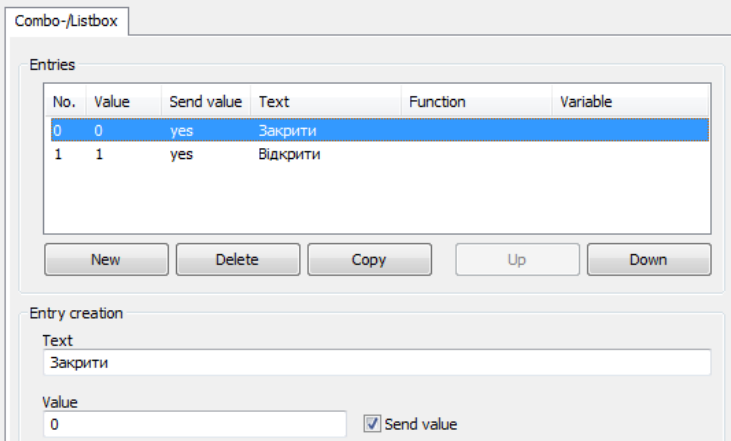
Таким чином, оператор не зможе проігнорувати цей екран незважаючи на те, що він має невеликі розміри.

Тепер давайте створимо функцію для відкриття та закриття цього екрану.

- ▶ В менеджері проекту виберіть вузол *Functions* і створіть нову функцію *Screen switch*
- ▶ У діалоговому вікні задайте перемикання на екран 'Клапан'
- ▶ Через властивості створеної функції перейменуйте її на 'Перехід на екран Клапан'
- ▶ Створіть нову функцію *Close frame*, яка знаходиться в розділі *Screens* вікна вибору функції
- ▶ Вкажіть фрейм 'Клапан' та потім перейменуйте створену функцію на 'Закриття екрану Клапан'

Залишилося створити усі необхідні елементи діалогового вікна екрану керування клапаном.

- ▶ За допомогою текстового поля у верхній частині екрану 'Клапан' розмістіть підпис 'Клапан 2'
- ▶ В куті екрану розмістіть червону кнопку закриття вікна, прив'язавши до неї функцію 'Закриття екрану Клапан'
- ▶ На інструментальній панелі виберіть елемент *Combo-/Listbox* і розмістіть його в центральній частині екрану
- ▶ До цього елемента прив'яжіть змінну `Танк[2].Клапан[1]`
- ▶ У властивостях комбобоксу в розділі *Representation* для поля *Type* повинно стояти *Combobox*
- ▶ Зайдіть в поле *Entries* і виконайте додавання позицій, відповідно до представленого малюнку нижче



No.	Value	Send value	Text	Function	Variable
0	0	yes	Закрити		
1	1	yes	Відкрити		

Buttons: New, Delete, Copy, Up, Down

Entry creation

Text: Закрити

Value: 0 ☒ Send value

Таким чином елемент комбобокс матиме дві позиції вибору – 'Закрити' і 'Відкрити', що відповідає '0' і '1'. Взагалі, комбобокс дозволяє створювати спадні списки на будь-яку кількість позицій.

Щоб нам відразу бачити реакцію системи на виконану дію давайте розмістимо тут елемент схематичного зображення клапану з використанням функції видимості.

- ▶ На інструментальній панелі виберіть елемент *Polygon* і намалюйте схематичне зображення клапану
- ▶ У властивостях полігону *Color* задайте зелений колір фону
- ▶ Зайдіть в розділ властивостей *Visibility/flashing* і в полі *Variable* вкажіть змінну 'Танк[2].Клапан[1]'
- ▶ Для полів *from* і *to* поставте значення '1'

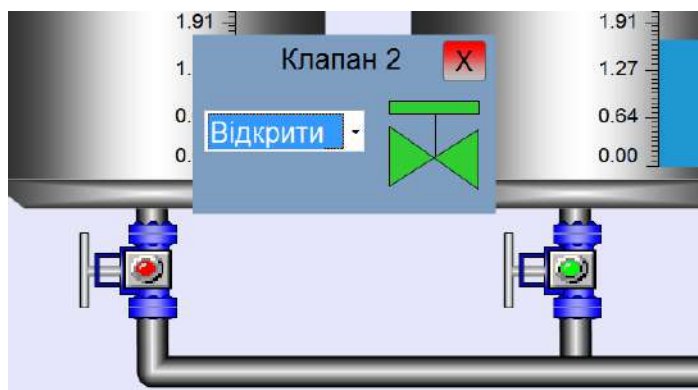
Остання дія забезпечить видимість цього елемента при значенні змінної '1'.

- ▶ Скопіюйте створений елемент полігону
- ▶ У властивостях задайте червоний колір фону і для полів видимості поставте значення '0'
- ▶ Розмістіть ці елементи на екрані один під одним

Нам залишилось створити кнопку для виклику цього діалогового вікна. Для цього ми створимо прозору кнопку.

- ▶ Відкрийте екран 'Процес'
- ▶ На інструментальній панелі виберіть елемент *Button* і розмістіть перед символом зливного клапану другого танку
- ▶ Автоматично з'явиться вікно прив'язки функції, в якому виберіть функцію 'Перехід на екран Клапан'
- ▶ В розділі властивостей елементу *Color* активуйте бокс *Transparent*

На цьому роботу по створенню діалогового вікна можна вважати завершеною. Скопіюйте проект і перейдіть в Runtime.



### Зміна розміру

Динамічне відображення набору рідини в танку можна реалізувати не тільки за допомогою спеціального елемента – стовпчикового показника, а й через звичайний прямокутник, задавши йому зміну висоти відповідно до значення змінної рівня. Даваймо реалізуємо це на нашій схемі.

- ▶ На інструментальній панелі виберіть елемент *Rectangle* і розмістіть його уздовж лівої стінки першого танку
- ▶ Зайдіть в розділ властивостей *Position* і в полі *Rotation angle* поставте значення '180'

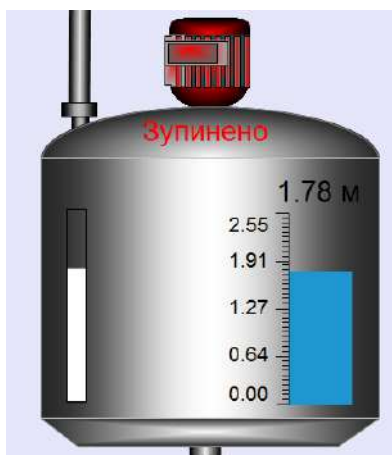
Цю дію необхідно зробити оскільки наповнення танку відбувається знизу вгору, а не навпаки.

- ▶ В розділі властивостей *Size and rotation dynamic* в полі *Change height* призначте змінну 'Танк[1].Рівень'
- ▶ Для поля *Min. height* поставте значення '0', для *Max. height* поставте значення '100'

Для візуально кращого сприймання зміни висоти прямокутника без наявності шкали, даваймо за нашим динамічним прямокутником розмістимо іще один з прозорим фоном.

- ▶ На інструментальній панелі виберіть елемент *Rectangle* і розмістіть його перед вже існуючим
- ▶ Зайдіть в розділ властивостей *Color* і в полі *Filling pattern* виберіть прозорий прямокутник
- ▶ Підгоніть розміри обох прямокутників, щоб вони співпадали і розміщувались чітко один за одним

Таким чином, в Runtime ми продублювали індикацію рівня рідини в танку. Аналогічним чином можна реалізувати зміну розмірів будь якого елемента по горизонталі.

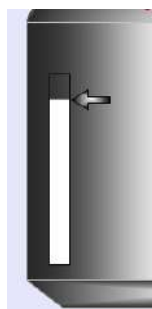


### Рухомий елемент

Для демонстрації руху елементів в системі, реалізуємо переміщення символу стрілки відповідно до зміни рівня в танку.

- ▶ В глобальній бібліотеці символів з папки *Arrows* виберіть символ 'left' і розмістіть його чітко біля верхнього кута прямокутника показу рівня, щоб кінчик стрілки вказував на нього
- ▶ Зайдіть в розділ властивостей *Position* і в полі *Y position dynamic* призначте змінну 'Танк[1].Рівень'
- ▶ Для поля *Min. movement* поставте значення '180', для *Max. movement* поставте значення '0'

Ці значення ми визначили дослідним шляхом, щоб рух символу стрілки найбільш чітко повторював рух рівня в танку.



### Зміна кольору

На завершення демонстрації динамічних можливостей системи керування, ми реалізуємо зміну кольору стовпчика рівня першого танку. Для цього ми визначимо чотири ліміти і задамо їм відповідні кольори:

0 – 0,55 м	'Низький рівень'	блакитний
0,55 – 2 м	'Нормативний рівень'	синій
2 – 2,35 м	'Високий рівень'	фіолетовий
2,35 – 2,55 м	'Критичний рівень'	червоний

- ▶ Зайдіть у властивості змінної 'Танк[1].Рівень' розділ *Limits*
- ▶ Через поле *Limit new* створіть новий ліміт

- ▶ В полі *Limit text* задайте підпис 'Низький рівень'
- ▶ В полі *Limit* задайте значення '0,55'
- ▶ Для *Additional attributes* задайте блакитний колір ліміту

Таким чином відображення рівня буде мати блакитний колір на проміжку значень 0 – 0,55 м.

Аналогічним чином створіть решту лімітів, згідно з наведеною вище таблицею.

Для того, щоб зроблене кольорове представлення лімітів не впливало на інші елементи, які використовують змінну 'Танк[1].Рівень', необхідно у властивостях цих елементів в розділі *Color* деактивувати бокс *Use limit color*.

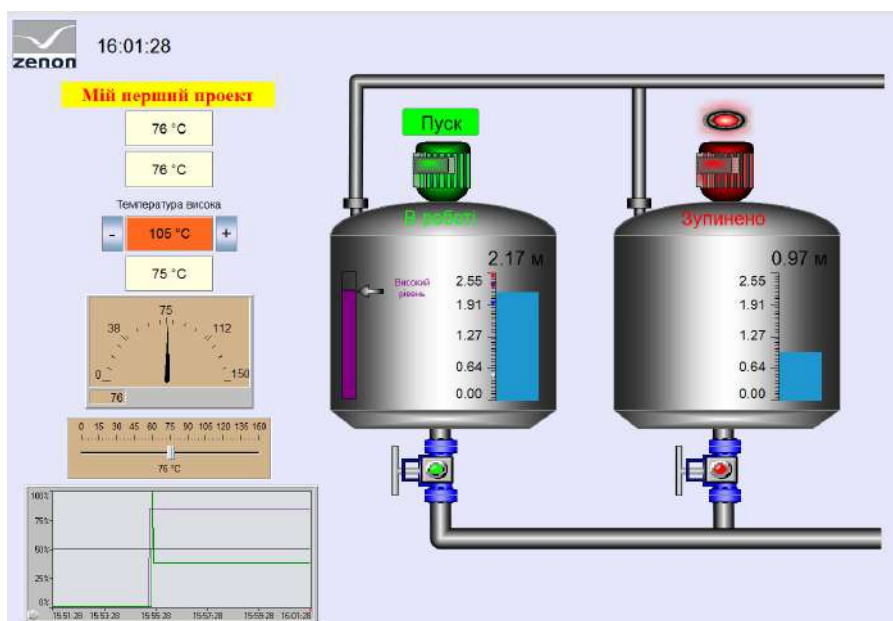
На завершення визначте змінну рівня, як база для відображення динамічного кольору фону елементу прямокутника – властивість *Color* -> *Color dynamic*.

Заданий текст для лімітів буде відображатись в якості підпису для тривоги, а також може бути показаний через елемент динамічного тексту. Ми це і зробимо.

- ▶ Розмістіть елемент *Dynamic text* по центру танку і прив'яжіть до нього змінну 'Танк[1].Рівень'
- ▶ Зробіть його прозорим, та в розділі *Representation* -> *Text format* активуйте бокс *Automatic word wrap*

Самостійно проробіть описану в даному розділі методику динамізації кольору для цифрового показника температури змінної 'Температура[3]' з визначенням чотирьох лімітних зон.

В результаті екран технології повинен мати такий вигляд:



## Блимання

Для того, щоб особливо привернути увагу оператора до критичного значення параметру, часто застосовують ефект блимання.

- ▶ У властивостях змінної 'Температура[3]' зайдіть в розділ *Limits* -> *Limit[4]*
- ▶ В полі *Additional attributes[4]* активуйте бокс *Flashing*

Таким чином, при досягненні температури четвертого граничного значення цифровий показчик змінної та відповідний динамічний текст почнуть блимати.

На цьому ми завершимо розгляд основних моментів створення технологічної мнемосхеми процесу та динамізації екранних елементів.

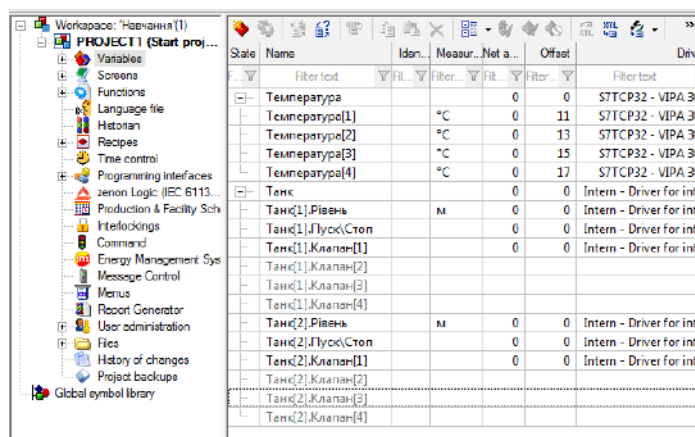
Познайомитись з іншими елементами, які ми не розглянули, та інструментами їх динамізації можна в манулі Screens -> Screen elements та Screens -> Edit screen element.

## Деактивація змінних

Як Ви бачите, в проекті ми задіяли не всі створені змінні. Для того, щоб система не опитувала незадіяні теги, в zenon існує можливість їх деактивувати. Ця функція доступна лише якщо працювати зі структурованими змінними. Часто, деактивацію змінних використовують для резервування змінних, якщо точно не відома їх кількість, але є потреба в закладанні певного масиву адрес.

- ▶ В менеджері проекту виберіть вузол *Variables*
- ▶ У вікні деталізованого перегляду виділіть через Ctrl змінні 'Танк[1].Клапан[2]', 'Танк[1].Клапан[3]' та 'Танк[1].Клапан[4]'
- ▶ У верхній частині цього вікна клікніть на піктограмі *Deactivate*

В результаті виконаної дії вибрані змінні стануть сірими. Це означає що вони деактивовані. Виконайте цю операцію і для незадіяних змінних другого танку.



State	Name	Idon.	Measur.	Not a...	Offset	Drive
	Температура				0	S7TCP32 - VIPA 3C
	Температура[1]		°C		11	S7TCP32 - VIPA 3C
	Температура[2]		°C		13	S7TCP32 - VIPA 3C
	Температура[3]		°C		15	S7TCP32 - VIPA 3C
	Температура[4]		°C		17	S7TCP32 - VIPA 3C
	Танк				0	Intern - Driver for int
	Танк[1].Рівень				0	Intern - Driver for int
	Танк[1].Пуск/Стоп				0	Intern - Driver for int
	Танк[1].Клапан[1]				0	Intern - Driver for int
	Танк[1].Клапан[2]					
	Танк[1].Клапан[3]					
	Танк[1].Клапан[4]					
	Танк[2].Рівень				0	Intern - Driver for int
	Танк[2].Пуск/Стоп				0	Intern - Driver for int
	Танк[2].Клапан[1]				0	Intern - Driver for int
	Танк[2].Клапан[2]					
	Танк[2].Клапан[3]					
	Танк[2].Клапан[4]					



## Скрипти

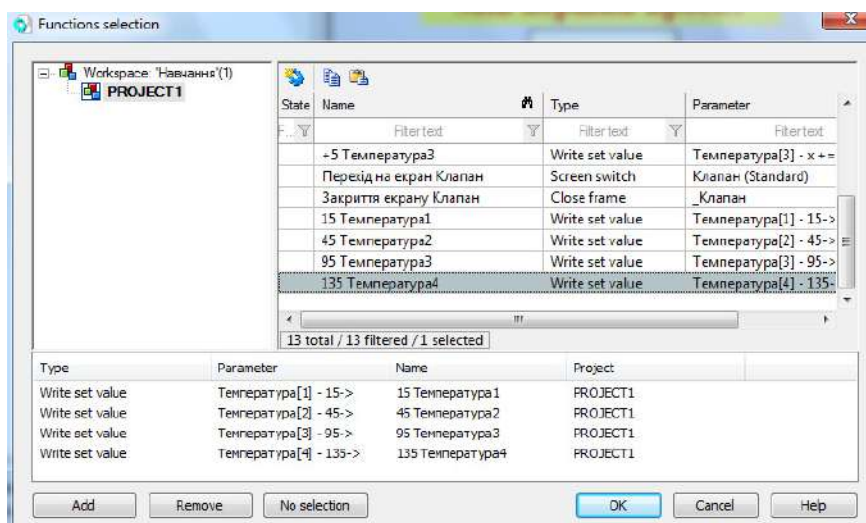
Існують задачі, коли при натисканні кнопки чи досягненні якогось критичного значення параметру необхідно виконати одночасно декілька функцій. Для цього в системі керування існує поняття скриптів.

*Скрипт – це одночасне виконання набору визначених функцій. На відміну від скриптів мов програмування, вони не вимагають написання жодної строки програмного коду.*

### Кнопочний скрипт

Давайте створимо скрипт встановлення уставок нашим температурним змінним. Попередньо нам необхідно створити набір функцій, які буде виконувати скрипт.

- ▶ В менеджері проекту виберіть вузол *Functions*
- ▶ Для змінної 'Температура[1]' створіть функцію *Write set value* - > 15
- ▶ Для змінної 'Температура[2]' створіть функцію *Write set value* - > 45
- ▶ Для змінної 'Температура[3]' створіть функцію *Write set value* - > 95
- ▶ Для змінної 'Температура[4]' створіть функцію *Write set value* - > 135
- ▶ В вузлі *Functions* виберіть підрозділ -> *Scripts*
- ▶ Піктограмою *New script* створіть новий скрипт та назвіть його 'Скрипт 1'
- ▶ Через контекстне меню створеного скрипту *Add functions* додайте створені функції



Тепер нам необхідно створити функцію виконання цього скрипту та прив'язати її до кнопки.

- ▶ Поверніться на рівень *Functions*
- ▶ В діалоговому вікні вибору нової функції зайдіть в розділ *Script* і виберіть функцію *Script: execute*
- ▶ Прив'яжіть сюди Ваш скрипт 'Скрипт 1'
- ▶ Поруч з цифровими показчиками температури створіть кнопку 'Виконання першого скрипту' і прив'яжіть до неї створену функцію

Перевірте роботу створеного скрипту.

### Граничний скрипт

Також можна створити скрипт, який буде активуватись по досягненню параметром встановленого граничного значення. Методика створення такого скрипту аналогічна. Відмінність полягає в прив'язці його виконання до граничного значення змінної.

Створіть скрипт 'Скрипт критичний рівень танку 1' з таким набором функцій: 'Відключення мішалки' та 'Відкриття зливного клапану'. Скрипт повинен виконуватись автоматично при досягненні в першому танку критичного рівня - значення  $> 2.35$  м.

- ▶ У властивостях змінної 'Танк[1].Рівень' зайдіть в налаштування четвертого ліміту
- ▶ Для поля *Function[4]* прив'яжіть функцію 'Скрипт критичний рівень танку 1'

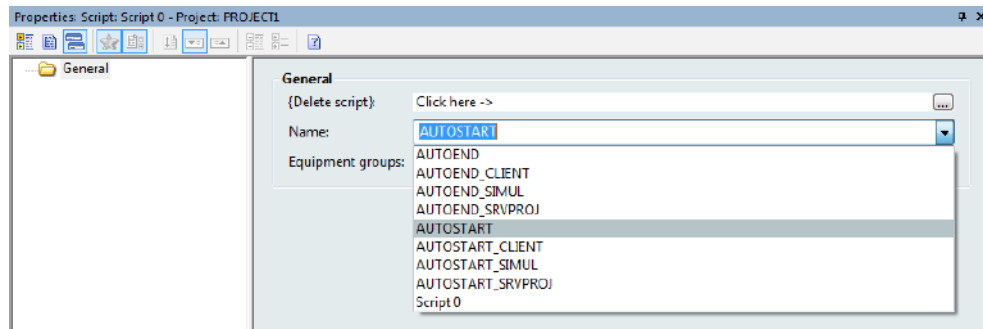
Перевірте в Runtime як виконується даний скрипт. Попередньо ввімкніть і відімкніть відповідні екранні елементи.

### Наперед визначені скрипти

Для виконання специфічних завдань в системі керування існують наперед визначені скрипти, які мають зарезервовані імена і чітко визначене призначення. Сюди відносяться скрипти AUTOSTART, AUTOEND, AUTO-START\_CLIENT і т.д. Повний перелік таких скриптів та їх опис наведено в мануалі *Scripts*.

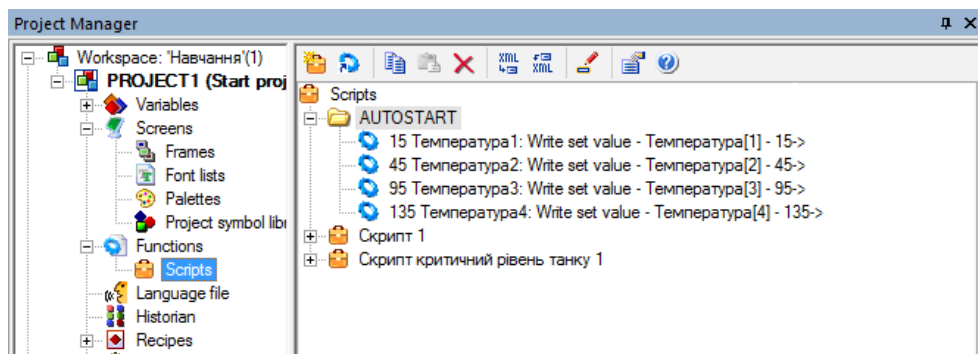
Давайте зробимо так, щоб при запуску проекту автоматично виконувалися функції, описані для нашого температурного скрипту.

- ▶ В вузлі *Functions* виберіть підрозділ *Scripts* і створіть новий скрипт
- ▶ У властивостях в полі *Name* зі спадного поля виберіть зарезервоване ім'я 'AUTOSTART'



- Додайте в скрипт функції встановлення значень температурним змінним

Таким чином, при старті проекту автоматично будуть виконуватись усі функції, визначені в скрипті автостарту. При чому, вони будуть виконуватись саме в тому порядку, в якому визначені в скрипті.



## Хронологічний список подій

Цим розділом ми почнемо розгляд спеціальних типів екранів візуалізації, які являються частиною будь-якого SCADA проекту, а саме подій, тривоги, архівів та трендів.

Давайте створимо екран для відображення Хронологічного списку подій.

*Chronologic Event List (CEL) – це спеціальний екран, в якому відображаються усі події, що відбуваються в системі, наприклад, запуск проекту, авторизація користувача, зміна значення параметру з чіткою фіксацією часу цієї дії. Функціонал такого екрану повністю відповідає вимогам FDA по безпеці керування процесами.*

### Створення екрану CEL

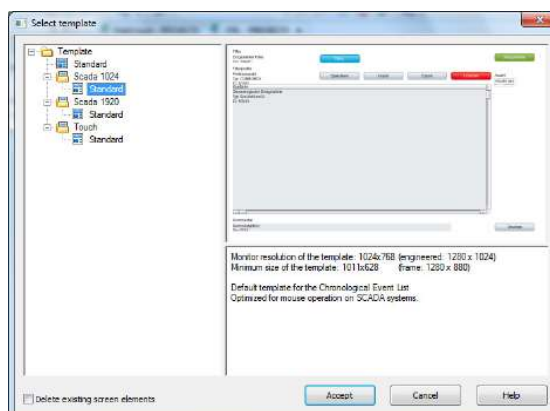
- ▶ В менеджері проекту виберіть вузол *Screens*
- ▶ У вікні детального перегляду клікніть на іконці *New screen*
- ▶ Через вікно властивостей перейменуйте створений екран на 'Хронологія'
- ▶ В полі *Screen type* розділу *General* виберіть спеціальний тип *Chronological Event List*
- ▶ В полі *Frame* повинно бути вибрано 'Головний'

*Майте на увазі, що після вибору спеціального типу для екрану, потім змінити його на інший вже не можна.*

Тепер на цей екран необхідно розмістити елементи керування і відображення. З цією метою в zenon розроблено ряд готових шаблонів. Для кожного типу екрану розроблено свій шаблон.

### Використання кнопочного шаблону

- ▶ При активному екрані 'Хронологія' зайдіть в головне меню *Control elements* і виберіть пункт *Insert template...*
- ▶ Далі з'явиться вікно готових наборів елементів керування, в якому виберіть *Standard* для *Scada 1024*



- Після підтвердження зробленого вибору кнопкою *Accept*, система автоматично розмістить на екрані усі необхідні елементи, притаманні суто цьому типу екрану

Підкорегуйте розміщення елементів та їх розміри відповідно до свого екрану, задайте свої підписи. Непотрібні кнопки чи текстові поля можете взагалі видалити.

Тепер для переходу на цей екран нам необхідно створити функцію *Screen switch* та кнопку 'Хронологія' на панелі навігації.

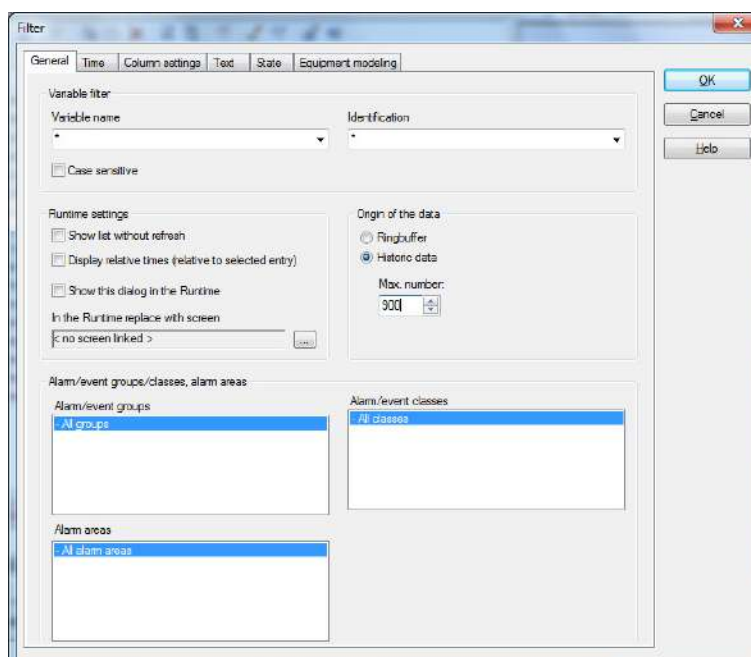
При створенні функції перемикавання з'явиться діалогове вікно конфігурування, яке раніше не зустрічалось. Це пов'язане з тим, що ми до цієї функції прив'язали не звичайний екран стандартного типу, а специфічний.

### Конфігурування функції перемикавання

Діалогове вікно конфігурування має шість вкладок, які ми поетапно розглянемо.

#### Вкладка *General*

Тут усі повідомлення можуть бути відфільтровані відповідно до заданих логічних критеріїв.

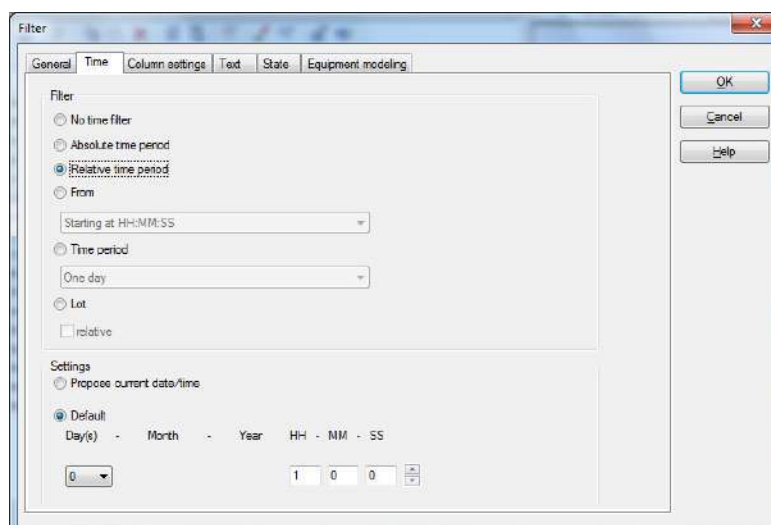


Для нашого прикладу нам необхідно зробити лише декілька змін.

- Активуйте опцію *Historic data*
- Встановіть максимально до 900 записів

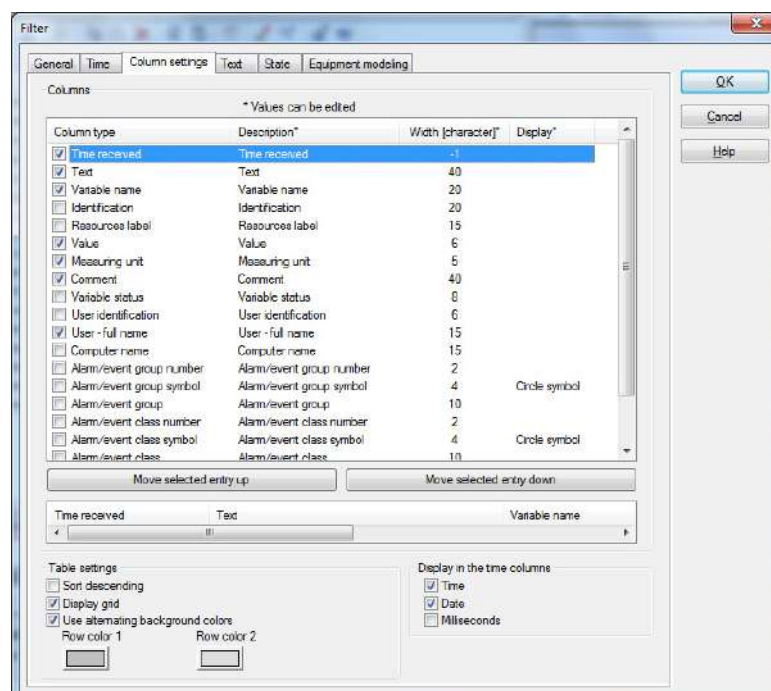
### Вкладка Time

Ця вкладка дозволяє нам налаштувати різноманітні варіанти часових фільтрів відображення інформації.



### Вкладка Column settings

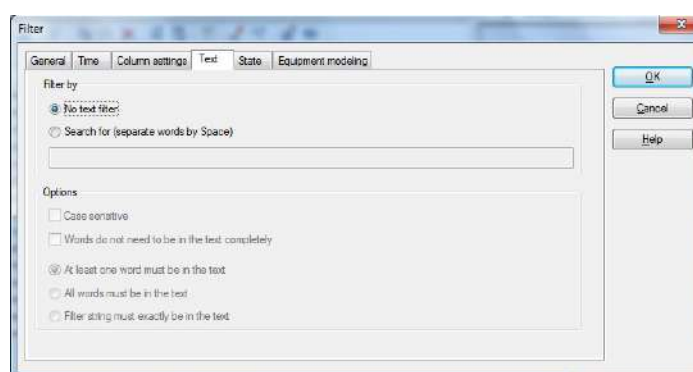
Дозволяє вибрати потрібні стовпчики, їх послідовність, задати їх ширину, підпис, стиль строк.



Зробіть так, щоб відображувались стовпці часу, тексту, змінної, значення, одиниць виміру, коментарю та користувача. Стовбець тексту та коментаря встановіть в 40 символів.

### Вкладка *Text*

Дозволяє здійснити фільтрування подій по заданому текстовому шаблону.



Нічого змінювати тут не будемо.

### Вкладка *State*

Дозволяє фільтрувати дані по стану змінної.

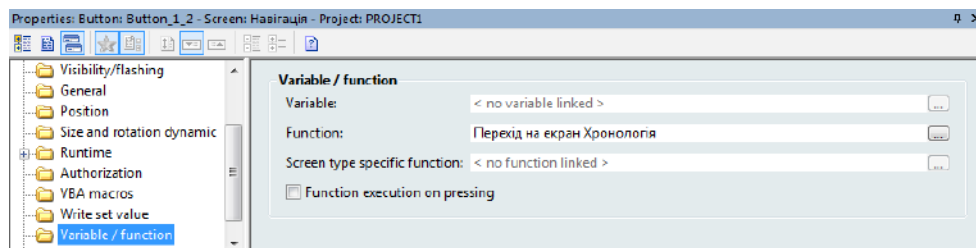
### Вкладка *Equipment modeling*

Якщо в проекті виконане моделювання обладнання, то тут можна визначити відображення списку подій, що стосуються певного обладнання.

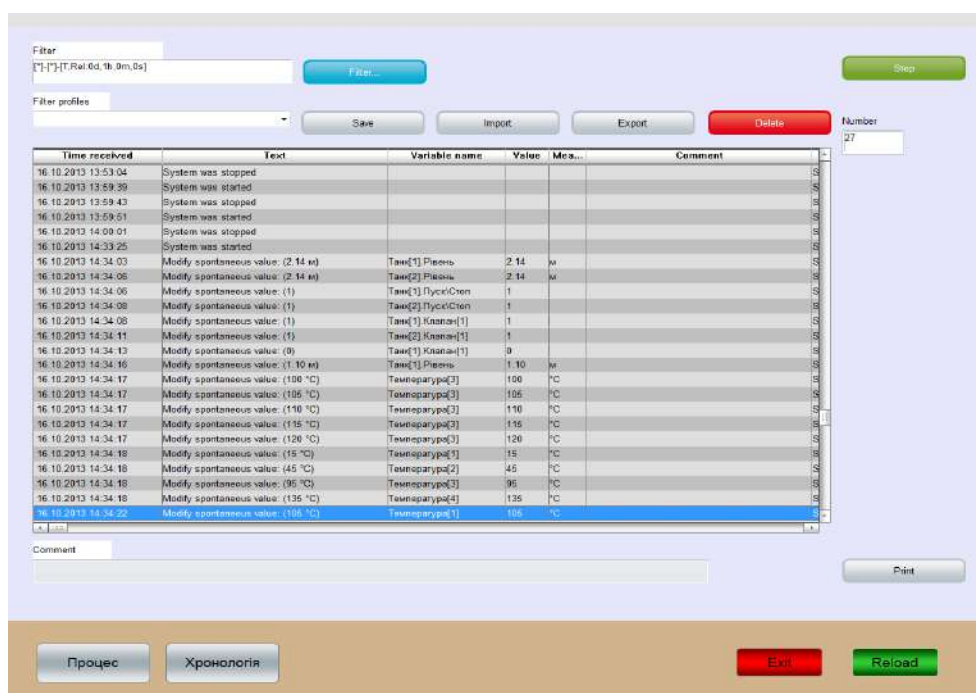
На даному етапі ми нічого змінювати не будемо, тому просто закінчіть створення та конфігурування функції натиснувши кнопку *OK*.

### **Робота з екраном**

Для перемикання на екран 'Хронологія' на панелі навігації необхідно створити відповідну кнопку. Її можна просто скопіювати з уже існуючої. Прив'язка функції здійснюється або перетягуванням її на відповідну кнопку (Drag&Drop), або через пункт властивостей *Variable / function -> Function*:



Запустіть Runtime і перевірте зроблену роботу. Ваш екран повинен мати наступний вигляд:



Представлення списку подій повністю налаштовується в функції перемикавання на цей екран.

Через кнопку 'Filter...' можна безпосередньо в онлайн режимі налаштовувати відображення списку подій під свої задачі.

Вибрані індивідуальні налаштування можуть бути збережені користувачем у вигляді профілю.

Через поле 'Comment' до кожної події, що відображається у списку, оператор має можливість задати свій коментар.

Для закріплення матеріалу, виконайте кілька операцій:

- Натисніть кнопку 'Filter...' і активуйте колонку *Computer name*



- Збережіть зроблені налаштування через профіль 'Filter profiles' з ім'ям 'Мій профіль'

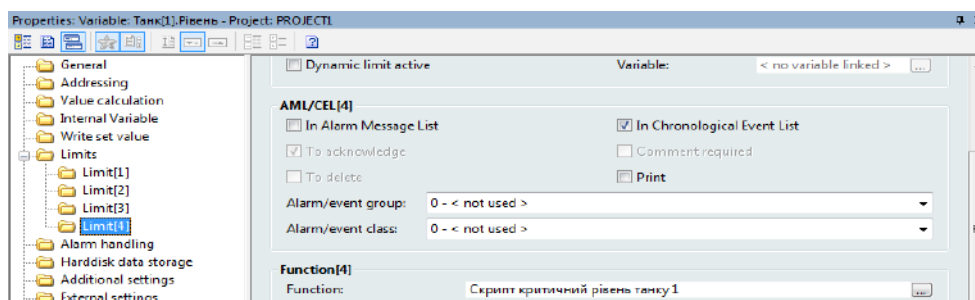
Виконайте перемикання між екранами 'Процес' та 'Хронологія' і перевірте як працює збережений профіль. За бажанням зробіть інші налаштування списку.

### Реєстрація лімітних значень

Екран хронології подій може реєструвати не тільки дії оператора, а й спрацювання граничного порогу (дублювання функцій екрану Тривог).

Давайте зробимо так, щоб при досягненні критичного рівня та температури ця подія заносилась в наш журнал.

- Виберіть змінну 'Танк[1].Рівень'
- У властивостях *Limit[4]* для поля *AML/CEL[4]* активуйте бокс *In Chronological Event List*

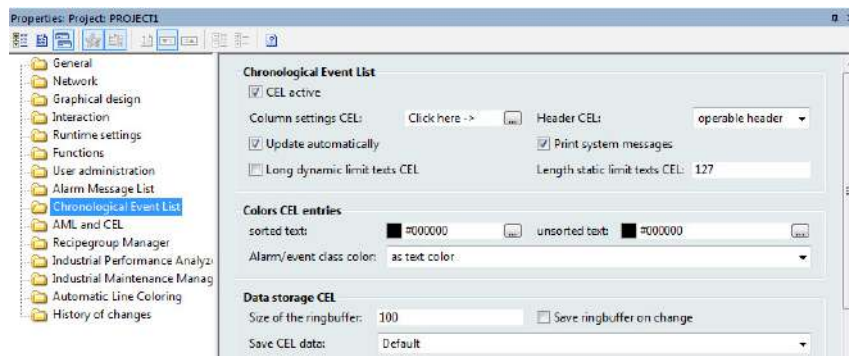


Для змінної 'Температура[3]' виконайте аналогічну операцію.

Поверніться в Runtime та задайте змінним критичні значення. Ця подія повинна з'явитись в Хронологічному списку подій. При чому, в колонці 'Text', буде зазначено текст для цього ліміту.

### Додаткові налаштування

Здійснити деякі додаткові налаштування відображення Хронологічного списку подій можна через властивості проекту, розділ *Chronologic Event List*.



Для поліпшення взаємодії оператора з екраном хронології існує спеціальний тип екрану - *Chronological Event List Filter*, який дозволяє в зручному вигляді здійснювати фільтрування списку подій по часу чи заданому ключовому слову. Цей екран фільтру повинен базуватись на своєму індивідуальному фреймі (як спливаюче pop-up вікно). Виклик такого екрану можна здійснювати через відповідну кнопку фільтру, чи разом з перемиканням на екран Хронологічного списку подій.

Дану функціональність перевірте самостійно.

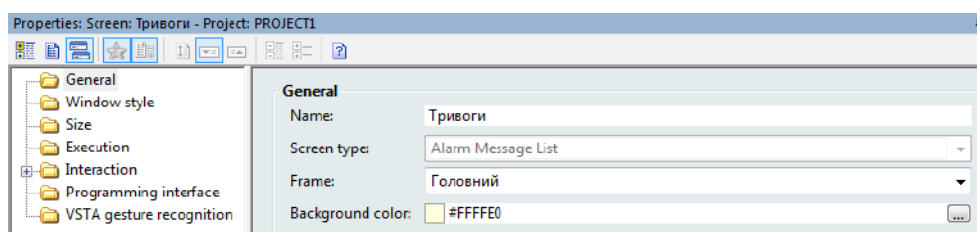
Докладніше про створення і роботу з екраном типу CEL можна познайомитись в мануалі [Chronological Event List](#).

## Тривоги

Досягнення критичних значень параметрів в системі реєструється в спеціальному журналі – Інформаційному списку тривог (Alarm Message List). До нього заноситься час появи тривоги, її статус (активна, не активна, підтверджена), опис тривоги, змінна, її значення, поточний оператор, робоча станція, коментар.

### Створення екрану AML

Аналогічно до попереднього розділу створіть новий екран, але як *Screen type* виберіть *Alarm Message List*. Назвіть екран 'Тривоги' та задайте довільний колір фону.



Через головне меню *Control elements* розмістіть на екрані усі необхідні елементи та відредагуйте їх розміщення.

Для переходу на цей екран як і в попередньому випадку нам необхідно створити функцію *Screen switch* та кнопку 'Тривоги' на панелі навігації.

При створенні функції перемикання з'явиться діалогове вікно конфігурування, яке має опції налаштування відображення тривог.

### Конфігурування функції перемикання

Діалогове вікно конфігурування має шість вкладок, суть яких подібна до тих, що були для функції перемикання на екран Хронології. Ми залишимо усі налаштування які стоять за замовчуванням. Закінчіть створення та конфігурування функції натиснувши кнопку *OK* і перейменуйте її в 'Перехід на екран Тривоги'.

### Робота з екраном

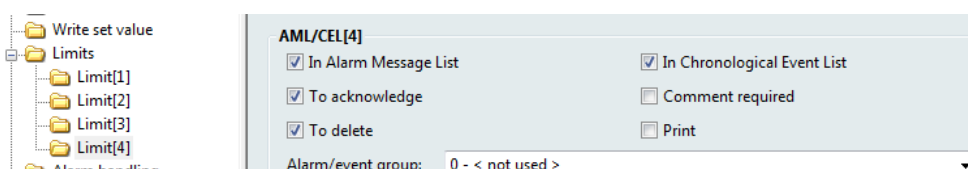
Для кнопки перемикання на екран тривоги прив'яжіть створену нами функцію 'Перехід на екран Тривоги'.

## ВИЗНАЧЕННЯ ЛІМІТІВ

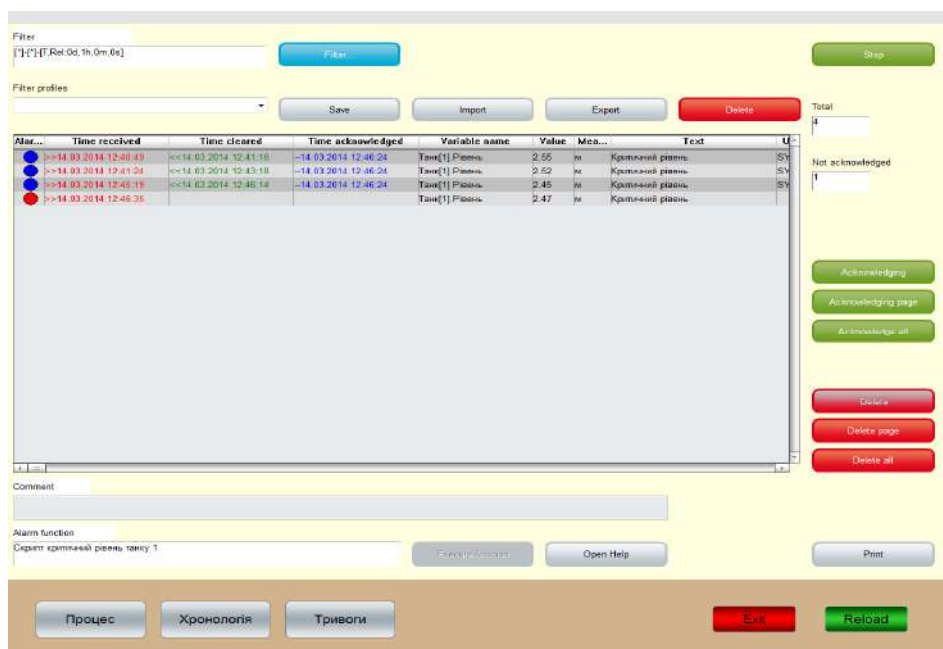
Якщо зараз запустити проект і перейти на екран тривоги, то жодних записів в списку не буде. Спрацювання тривоги і відображення в Інформаційному списку тривог визначається індивідуально для кожної змінної в лімітах. З ними ми вже познайомились при конфігуруванні кольорів для рівня танку.

Давайте зробимо так, щоб при досягненні останнього визначеного ліміту для змінної рівня - 'Критичний рівень', спрацювала тривога.

- Зайдіть у властивість змінної 'Танк[1].Рівень' розділ *Limits*
- В підрозділі *Limit[4]* для опції *AML/CEL[4]* активуйте бокс *In Alarm Message List*
- Також, активуйте бокси *To acknowledge* та *To delete*



Таким чином, при досягненні рівня верхнього лімітного значення (в нас визначено в діапазоні 2.35 - 2.55), спрацює тривога, яка буде відображена в журналі тривог. При цьому вона буде мати статус 'Активна' (червоне коло). Якщо рівень впаде нижче лімітного значення, то вона змінить свій статус на 'Не активна' (зелене коло). Також, тривога може змінити свій статус на 'Підтверджена' (синє коло), якщо оператор натисне кнопку 'Acknowledge'. Оскільки ми активували опцію *To delete*, то даний запис пропаде зі списку тільки якщо ми вручну видалимо його відповідною кнопкою.



Скомпілюйте зроблені в проекті зміни та запустіть Runtime. Змінюючи значення рівня промодельуйте спрацювання граничного значення в Танку1. Перейдіть на екран 'Тривоги'. Там повинний з'явитися відповідний запис з текстом 'Критичний рівень'. Спробуйте підтвердити цю тривогу та видалити.

Ваш екран повинен мати вигляд, як показано на малюнку вище.

Аналогічним чином сконфігуруйте спрацювання тривоги при переведенні змінної 'Танк[2].Пуск\Стоп' в стан стоп (значення '0').

- ▶ Зайдіть у властивість змінної 'Танк[1].Рівень' розділ *Limits*
- ▶ В підрозділі *Limit[1]* для опції *AML/CEL[4]* активуйте бокс *In Alarm Message List* та *To Acknowledge*

Перевірте в Runtime спрацювання цієї тривоги.

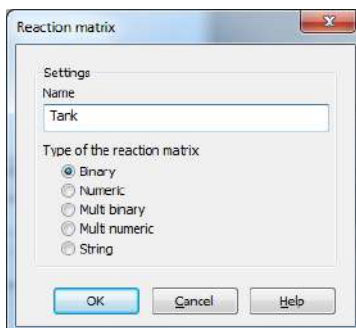
## МАТРИЦІ РЕАКЦІЇ

Окрім лімітів, в zenon існує можливість централізованого визначення граничних значень спрацювання тривог - матриці реакції.

*Матриці реакції (REMA) набагато більш функціональний інструмент в порівнянні з лімітами. Він дозволяє сконфігурувати спрацювання тривоги не тільки по значенню змінної, а й по 64 станам цієї змінної (недійсне значення, альтернативне...) та логічній умові (більше, менше, рівно).*

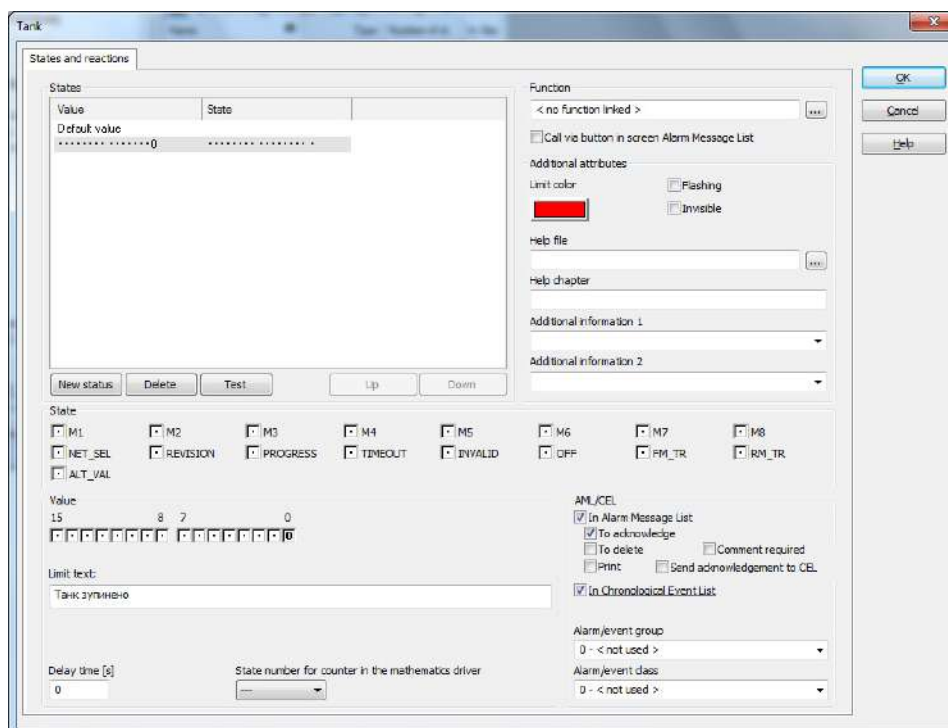
Матриця реакції буває бінарна, цифрова, мультибінарна, мультицифрова та строкова.

- ▶ В дереві проекту зайдіть в розділ *Variables* -> *Reaction matrix* і правою кнопкою миші створіть бінарну матрицю з назвою 'Tank'



Після натиснення кнопки *OK* з'явиться діалог налаштування матриці.

- ▶ Натисніть на кнопку *New status* і для першого біту в полі *Value* поставте значення '0'.
- ▶ В полі *AML/CEL* активуйте бокс *In Alarm Message List*, *To Acknowledge*, *In Chronological Event List*



Матриця реакції дозволяє сконфігурувати ініціацію перевірки необмеженої кількості статусів, які перебираються системою послідовно один за одним починаючи з верхнього. Як тільки статус підтверджується, дія сконфігурована для нього, автоматично виконується (поле *Function*). Тут також визначається колір відображення та текст тривоги.

- ▶ Конфігурування матриці реакції закінчить кнопкою *OK*

### Прив'язка матриці реакції до змінної

Тепер створену матрицю реакції необхідно прив'язати до відповідної змінної.

- ▶ Виберіть змінну 'Танк[2].Пуск\Стоп'
- ▶ В розділі *Limits* -> *Reaction matrix* виберіть створену матрицю 'Tank'



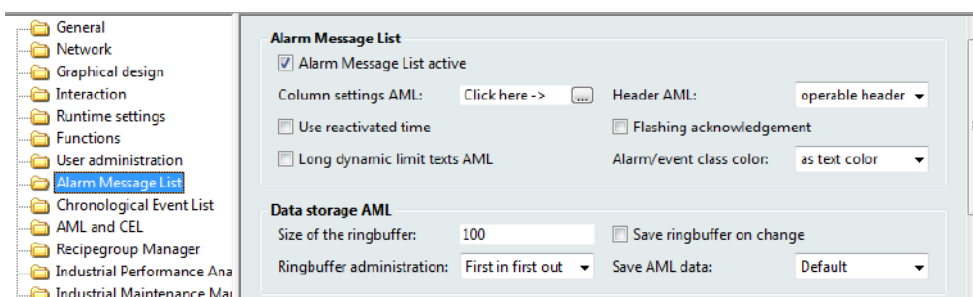
При цьому задані раніше лімітні значення деактивуються.

Якщо система має велику кількість однотипних змінних, то використання матриць реакції значно прискорює процес створення проекту.

Аналогічним чином можуть бути створені цифрова та строкова матриця реакції. В рамках цього навчання ми їх розглядати не будемо.

### Додаткові налаштування

Здійснити деякі додаткові налаштування відображення Інформаційного списку тривог можна через властивості проекту, розділ *Alarm Message List*.



Для поліпшення взаємодії оператора з екраном тривог існує спеціальний тип екрану – *Alarm Message List Filter*, який дозволяє в зручному вигляді здійснювати фільтрування списку подій по часу чи заданому ключовому слову. Цей екран фільтру повинен базуватись на своєму індивідуальному фреймі (як спливаюче рор-уп вікно). Виклик такого екрану можна здійснювати через відповідну кнопку фільтру, чи разом з перемиканням на екран Інформаційного списку подій.

Дану функціональність перевірте самостійно.

Докладніше про створення і роботу з екраном типу AML можна познайомитись в мануалі [Alarms administration](#).

## Архіви

Система керування дозволяє здійснювати архівування в трьох режимах: циклічному (раз в секунду, хвилину, годину...), по зміні параметру, по події (тригерне). Усі зібрані дані можуть зберігатися у внутрішньому форматі (arg), текстовому (csv), веб-форматі (xml) чи безпосередньо в SQL-базі даних. Також, окремо виділяється пакетне архівування, яке представляє собою спеціальний режим архівування, який використовується в харчовій та фармацевтичній промисловості при виробництві продукції по-партіям.

*Базова версія ліцензії zenon дозволяє здійснювати циклічне архівування. Усі інші режими архівування, у тому числі і створення каскадних архівів, доступні лише при замовленні повноцінного модуля архівів.*

### СТВОРЕННЯ ЦИКЛІЧНОГО АРХІВУ

Давайте створимо архів, в який кожні 5 секунд будуть заноситись значення змінних температури та рівня.

#### Конфігурування архіву

- В дереві проекту зайдіть в розділ *Historian* і правою кнопкою миші виберіть пункт *New Archive...*

Автоматично з'явиться майстер швидкого налаштування архіву:



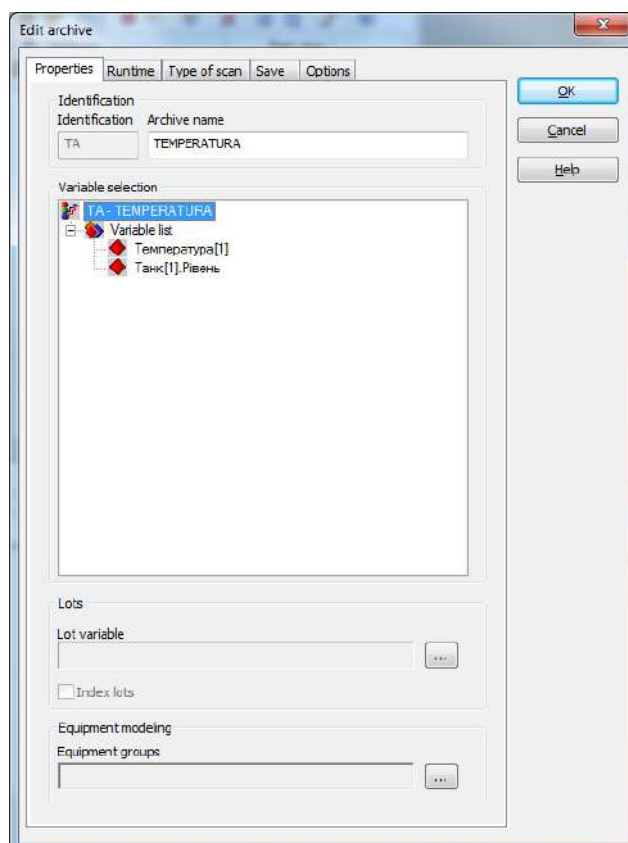
Ми виконаємо усі налаштування уручну, тому просто закрийте його кнопкою *Cancel*.

Після цього з'явиться діалогове вікно конфігурування архіву, яке містить п'ять вкладок. Далі ми розглянемо кожну з них.



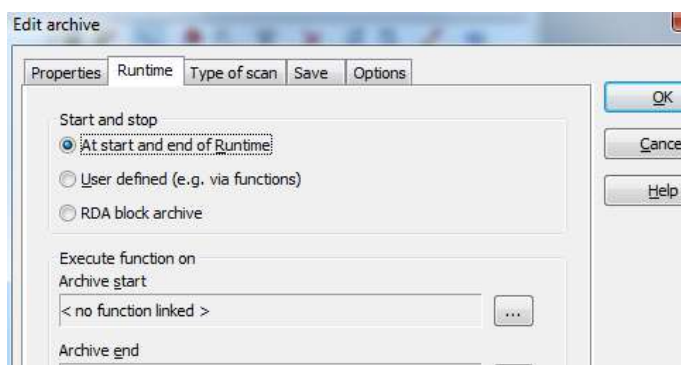
### Вкладка *Properties*

- ▶ Задайте довільний ідентифікатор архіву та його назву – 'TA', 'TEMPERATURA'
- ▶ В секції *Variable selection* правою кнопкою миші через *Add variable* додайте змінну 'Танк[1].Рівень' і 'Температура[1]'



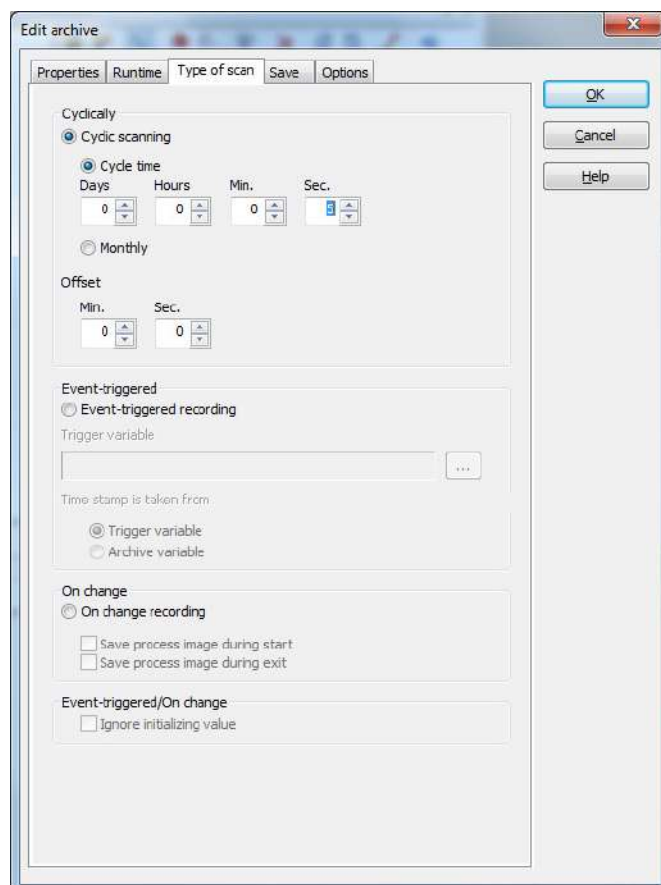
### Вкладка *Runtime*

На цій вкладці конфігурується запуск і зупинка ведення архіву. Залишимо запуск архівування при старті Runtime, як встановлено за замовчуванням.



### Вкладка *Type of scan*

На третій вкладці визначається частота зчитування значень змінних і заносу їх до архіву.



- ▶ Виберіть циклічний режим сканування *Cyclic scanning*
- ▶ Параметр *Cycle time* встановіть в 5 Sec.

Таким чином, значення наших змінних будуть заноситись до архіву кожні 5 секунд.

*Мінімальна дискретизація інтервалу запису в архів – 1 секунда. Для швидшого збереження даних, необхідно використовувати технологію RDA, при якій дані буферизуються в контролері, а потім пакетами передаються в SCADA систему.*

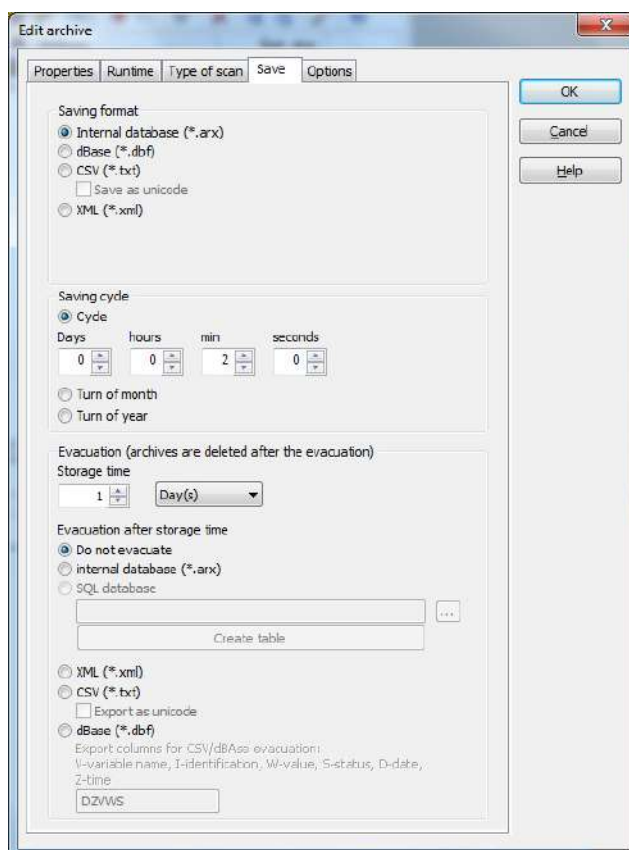
### Вкладка *Save*

Це найважливіша вкладка діалогового вікна налаштування архіву, в якій визначається формат збережених файлів, їх розмір та метод видалення.

- ▶ Встановіть збереження архіву у внутрішньому форматі *Internal database* з циклом *2 Min*

При таких налаштуваннях кожні дві хвилини система буде утворювати окремий архівний файл \*.arx, який містить по 24 записів (кожні 5 секунд протягом 2-х хвилин).

*Архівні файли внутрішнього формату – це зашифровані бінарні файли невеликої ємності, які ніякою програмою, за винятком самого zenon, не можуть бути відкриті. Окрім перегляду через екран архівів, ці дані також можна використовувати в інших модулях системи керування, таких як Тренди та Звіти.*



- ▶ Активуйте опцію *Do not evacuate* і встановіть час збереження файлів в 1 день

При таких налаштуваннях архівні файли не будуть вилучатись, а будуть зберігатись один день. Після чого вони у вигляді кільцевого буферу будуть перезаписуватись (нові дані записуються на місце старих).

### Вкладка Options

Містить додаткові налаштування архіву. Тут ми нічого чіпати не будемо.

- ▶ Закрийте діалогове вікно конфігурування архіву кнопкою *OK*

Зроблені налаштування архіву можна у будь-який момент змінити за допомогою піктограми *Edit archive* у вікні детального перегляду.

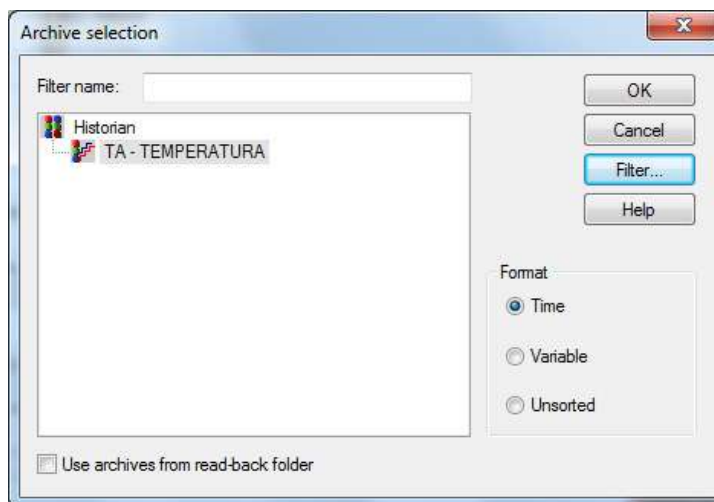
### Створення екрану перегляду архіву

- ▶ В дереві проекту зайдіть в розділ *Screens* і на базі фрейму 'Головний' створіть новий екран 'Архів'
- ▶ В якості спеціального типу виберіть *Archive revision*
- ▶ Через головне меню *Control elements* розмістіть на екрані усі необхідні елементи керування та відображення

### Конфігурування функції перемикавання

Для переходу на екран перегляду архіву ми створимо функцію *Screen switch* та прив'яжемо її до кнопки 'Архів', яку, як зазвичай, розмістимо на панелі навігації.

При створенні функції перемикавання з'явиться діалогове вікно конфігурування, яке має набір опцій налаштування.



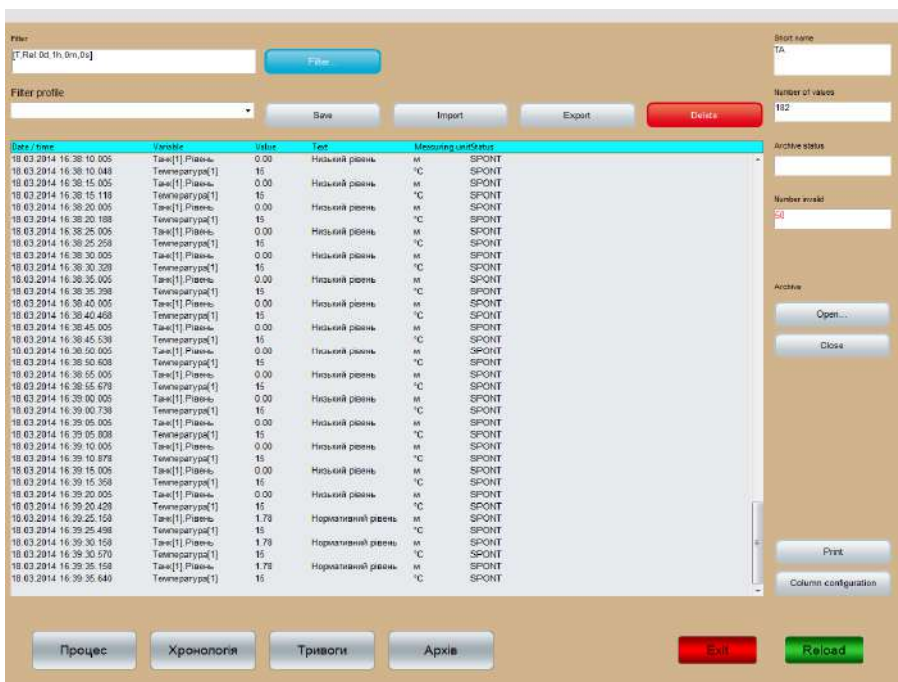
Тут можна визначити три режими сортування даних: в часовій послідовності потрапляння до архіву, згруповані по змінним, та взагалі без сортування. Особливої уваги заслуговує кнопка *Filter...*, яка містить ряд налаштувань представлення архівних даних. Тут можна виконати фільтрування по архівним змінним, які ми бажаємо бачити при переході на екран та діапазон часу.

Залишимо усі значення, як встановлено за замовчуванням. Завершіть конфігурування функції натиснувши кнопку **OK** і перейменуйте її в 'Перехід на екран Архів'.

### Робота з екраном

- Відкрийте екран 'Навігація' і створіть нову кнопку з ім'ям 'Архів'
- У властивостях *Variable \ function* -> *Function* виберіть створену нами функцію 'Перехід на екран Архів'

Запустіть проект і перейдіть на екран перегляду архіву. Система автоматично підвантажить усі дані, записані в архів у вигляді списку відсортованому по часу. Список містить сім колонок: час, ім'я змінної, ідентифікатор, значення, текст підпису граничного значення, одиниці виміру, статус точки даних. Керування шириною стовбців здійснюється через кнопку 'Column configuration'.



Date / time	Variable	Value	Text	Measuring unit/status
18.03.2014 16:38:10.005	Танк[1].Рівень	0.00	Низький рівень	м SPONT
18.03.2014 16:38:10.048	Температура[1]	15		°C SPONT
18.03.2014 16:38:15.005	Танк[1].Рівень	0.00	Низький рівень	м SPONT
18.03.2014 16:38:15.118	Температура[1]	15		°C SPONT
18.03.2014 16:38:20.005	Танк[1].Рівень	0.00	Низький рівень	м SPONT
18.03.2014 16:38:20.198	Температура[1]	15		°C SPONT
18.03.2014 16:38:25.005	Танк[1].Рівень	0.00	Низький рівень	м SPONT
18.03.2014 16:38:25.258	Температура[1]	15		°C SPONT
18.03.2014 16:38:30.005	Танк[1].Рівень	0.00	Низький рівень	м SPONT
18.03.2014 16:38:30.320	Температура[1]	15		°C SPONT
18.03.2014 16:38:35.005	Танк[1].Рівень	0.00	Низький рівень	м SPONT
18.03.2014 16:38:35.398	Температура[1]	15		°C SPONT
18.03.2014 16:38:40.005	Танк[1].Рівень	0.00	Низький рівень	м SPONT
18.03.2014 16:38:40.468	Температура[1]	15		°C SPONT
18.03.2014 16:38:45.005	Танк[1].Рівень	0.00	Низький рівень	м SPONT
18.03.2014 16:38:45.530	Температура[1]	15		°C SPONT
18.03.2014 16:38:50.005	Танк[1].Рівень	0.00	Низький рівень	м SPONT
18.03.2014 16:38:50.808	Температура[1]	15		°C SPONT
18.03.2014 16:38:55.005	Танк[1].Рівень	0.00	Низький рівень	м SPONT
18.03.2014 16:38:55.678	Температура[1]	15		°C SPONT
18.03.2014 16:39:00.005	Танк[1].Рівень	0.00	Низький рівень	м SPONT
18.03.2014 16:39:00.730	Температура[1]	15		°C SPONT
18.03.2014 16:39:05.005	Танк[1].Рівень	0.00	Низький рівень	м SPONT
18.03.2014 16:39:05.808	Температура[1]	15		°C SPONT
18.03.2014 16:39:10.005	Танк[1].Рівень	0.00	Низький рівень	м SPONT
18.03.2014 16:39:10.878	Температура[1]	15		°C SPONT
18.03.2014 16:39:15.005	Танк[1].Рівень	0.00	Низький рівень	м SPONT
18.03.2014 16:39:15.358	Температура[1]	15		°C SPONT
18.03.2014 16:39:20.005	Танк[1].Рівень	0.00	Низький рівень	м SPONT
18.03.2014 16:39:20.428	Температура[1]	15		°C SPONT
18.03.2014 16:39:25.150	Танк[1].Рівень	1.78	Нормальний рівень	м SPONT
18.03.2014 16:39:25.498	Температура[1]	15		°C SPONT
18.03.2014 16:39:30.158	Танк[1].Рівень	1.78	Нормальний рівень	м SPONT
18.03.2014 16:39:30.570	Температура[1]	15		°C SPONT
18.03.2014 16:39:35.158	Танк[1].Рівень	1.78	Нормальний рівень	м SPONT
18.03.2014 16:39:35.640	Температура[1]	15		°C SPONT

При кожному запуску проекту встановлені налаштування ширини стовбців будуть збиватись. Щоб цього не було параметри необхідно прописати в системному файлі програми *zenon.ini*

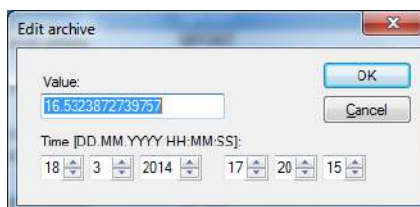
```
[ARCHEDIT]
ARCHEDITFONT=-13,0,0,0,400,0,0,0,3,2,1,49, Courier New
DATUMBREITE=25
EINHEITBREITE=10
STATUSBREITE=15
VARIABLEBREITE=10
WERTBREITE=10
ZUSTANDBREITE=10
```

*Цей файл знаходиться в системній директорії C:\ProgramData\COPA-DATA\System (для Windows 7). Будьте обережними зі змінами в ньому, оскільки невірні дії можуть вивести систему з ладу.*

Через кнопку 'Filter...' оператор має можливість в режимі Runtime виконати певні налаштування представлення списку даних.

Також, система керування дозволяє здійснювати ручну модифікацію значень змінних архіву.

- Двічі клікніть на значенні змінної яке Ви хочете відкорегувати

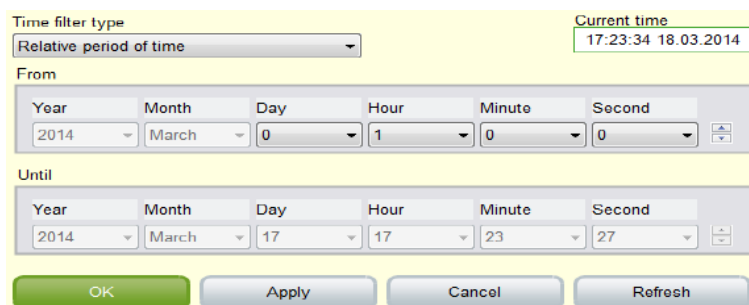


Помінявши значення, статус запису зміниться на MAN\_VAL і заголовок списку прийме червоний колір. Після збереження зроблених правок заголовок поверне свій початковий вигляд.

*Архівні дані представляються суцільним списком, який не дуже зручний для аналізу. Для представлення вимірної інформації у вигляді повноцінного табличного звіту з довільним розбиттям даних по стовбцям чи колонкам, необхідно використовувати модуль Reporting.*

### Часовий фільтр

Для поліпшення взаємодії оператора з екраном перегляду архіву існує спеціальний тип екрану – *Time filter*, за допомогою якого можна в зручному вигляді здійснювати фільтрування списку даних по часу. Цей екран фільтру повинен базуватись на своєму індивідуальному фреймі і викликатись разом з екраном архіву.

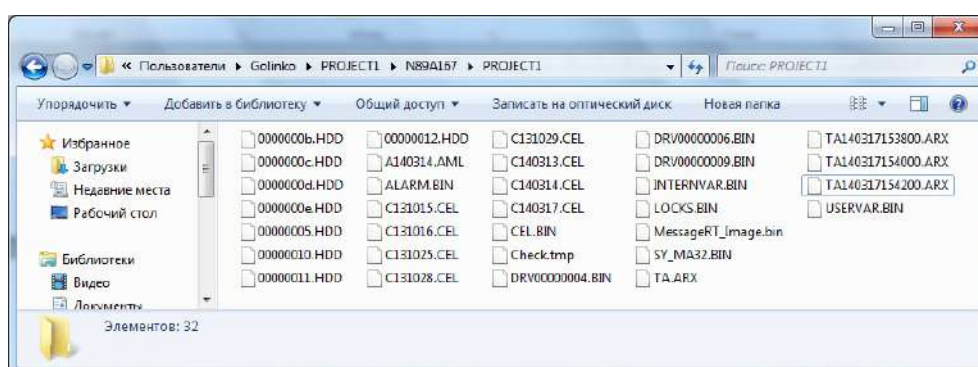


Дану функціональну можливість перевірте самостійно.

## Архівні файли

Система утворює два типи файлів. Один файл – заголовний TA.ARX, який містить налаштування архіву. Усі інші – файли даних, наприклад, TA140317153800.ARX. В назві зашифровано ідентифікатор архіву, дату та час створення файлу (по Гринвічу). Наступний створений системою файл буде мати ім'я TA140317154000.ARX.

Ці файли зберігаються папці проекту, яка містить його ім'я та назву робочої станції (наприклад, C:\Users\Golinko\PROJECT1\N89A167\PROJECT1). Точну директорію можна знайти через налаштування властивостей проекту *General* -> *Name/Folder* -> *Runtime folder*.



## Експорт архіву

Якщо при конфігуруванні архіву, на вкладці *Save* вибрати опцію збереження даних в форматі CSV, то інформація буде зберігатись в файли в папку *ExportArx*, місце розташування якої визначається у властивостях проекту *General* -> *File storage*. По замовчуванню це папка проекту (C:\Users\Golinko\PROJECT1\ExportArx).

Файли CSV-формату - це текстові файли з роздільниками, придатні до використання і аналізу в MS Excel.

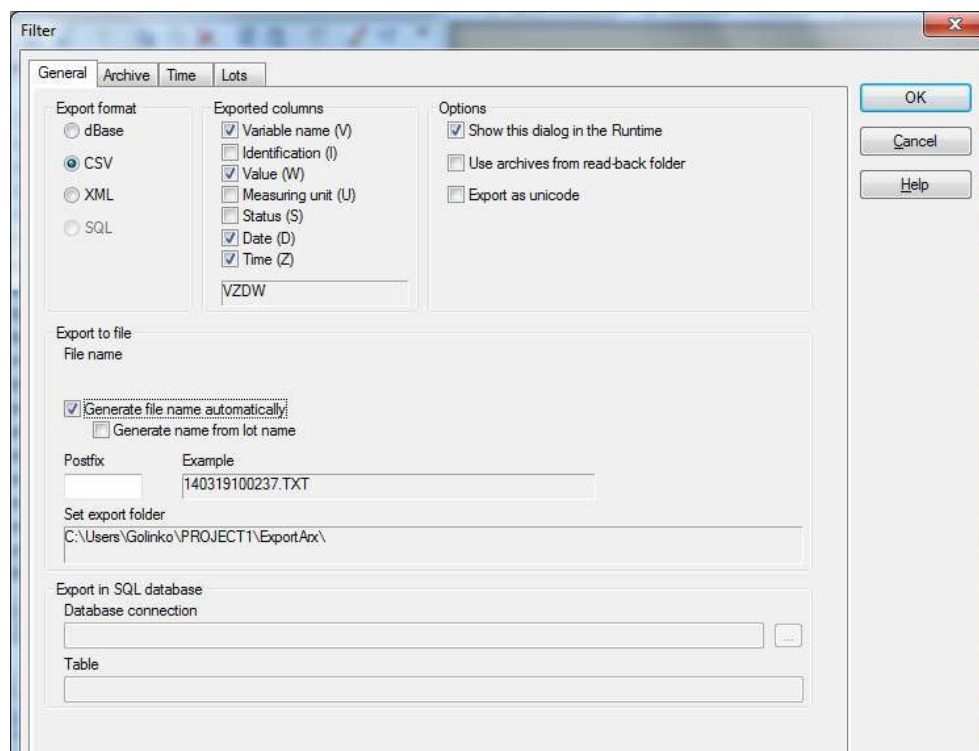
Проте, якщо у Вас базова ліцензія, то ця опція буде недоступна. В такому випадку експорт даних можна здійснювати через спеціально створену функцію.

- В менеджері проекту зійдіть в розділ *Function* і створіть нову функцію *Export archives*, яка знаходиться в підрозділі *Historian*

Далі з'явиться діалог налаштування функції експорту з набором вкладок.

- ▶ На вкладці *General* активуйте опції експорту в формат *CSV* з заносом даних змінної і часу, автоматичної генерації ім'я файлу та відображення цього діалогу в Runtime.
- ▶ На вкладці *Archive* виберіть архів 'TA-TEMPERATURA'

- На вкладці *Time* виберіть час, за який повинні експортуватись дані, наприклад, 1 годину



- Закінчіть конфігурування кнопкою *OK* і перейменуйте створену функцію в 'Експорт архіву'

Архівні файли будуть експортуватись в файли в папку *ExportArx*, місце розташування якої визначається у властивостях проекту *General -> File storage*. При бажанні тут можна визначити будь-яку директорію.

Тепер нам необхідно створити на екрані перегляду архіву відповідну кнопку експорту.

- Відкрийте екран 'Архів' і справа над кнопкою друку створіть нову кнопку з назвою 'Експорт'
- У властивостях створеної кнопки *Variable \ function -> Function* прив'яжіть функцію 'Експорт архіву'

Запустіть Runtime і виконайте експорт архіву. Оскільки при конфігуруванні функції експорту ми активували опцію відображення діалогу налаштування, то оператор має можливість в онлайн режимі вибрати формат збереження даних та параметри, які повинні заноситись до архіву.

Поетапно виконайте експорт архіву в усі доступні формати: dBase, CSV та XML.



## СТВОРЕННЯ КАСКАДНОГО АРХІВУ

Існують випадки, коли необхідно здійснювати архівацію не прямих значень змінних, а похідних даних, наприклад сумарного значення за певний період, середнього, мінімального чи максимального. Для цієї задачі в zenon існує поняття каскадного архіву.

Давайте на базі нашого циклічного архіву організуємо каскадний архів мінімальних значень температури та рівня.

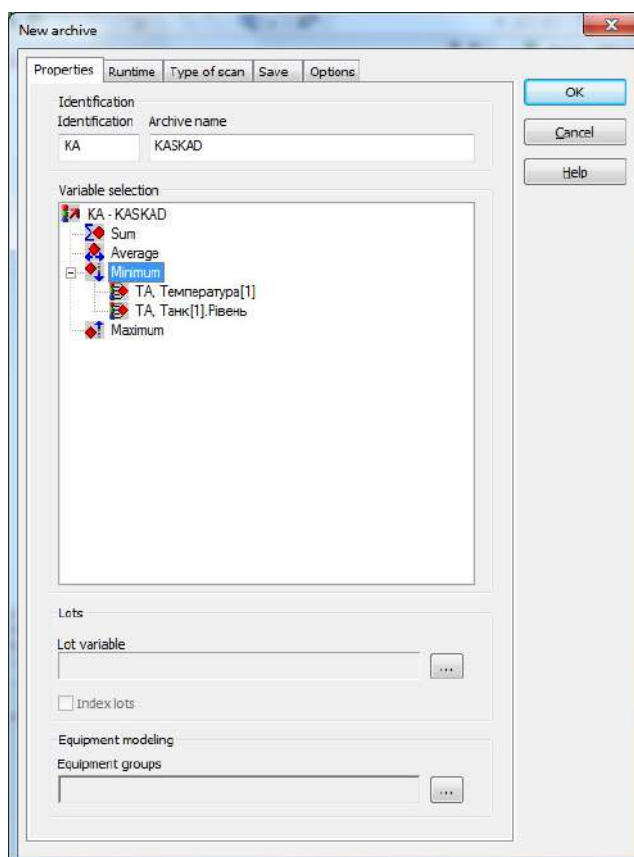
### Конфігурування архіву

- ▶ В дереві проекту зайдіть в розділ *Historian* і з контекстного меню архіву 'TA-TEMPERATURA' виберіть пункт *New Aggregated archive...*

Автоматично з'явиться майстер швидкого налаштування архіву. Ми виконаємо усі налаштування вручну, тому просто закрийте його кнопкою *Cancel*.

Після цього з'явиться вже знайоме нам діалогове вікно конфігурування архіву.

#### Вкладка *Properties*

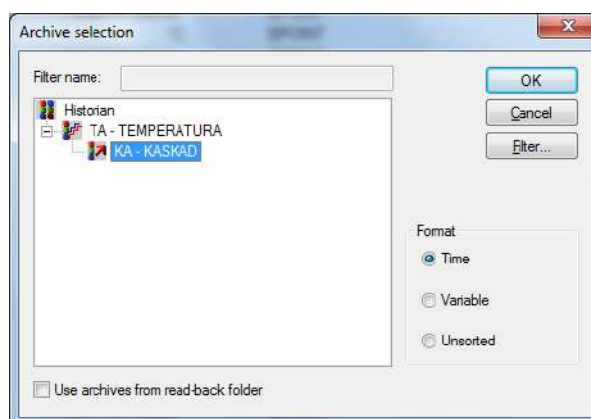


- ▶ Задайте довільний ідентифікатор архіву та його назву – 'KA', 'KASKAD'
- ▶ В секції *Variable selection* правою кнопкою миші через *Add variable* для розділу *Minimum* додайте змінну 'Танк[1].Рівень' і 'Температура[1]'

Налаштування на інших вкладках залиште без змін: запис по зміні, збереження у внутрішню пам'ять кожні 2 хвилини протягом 1 дня, без евакуації.

- ▶ Закрийте діалог налаштування кнопкою *OK*.

На цьому конфігурування каскадного архіву завершено. Перейдіть в Runtime і відкрийте вікно перегляду архіву. Кнопкою 'Open...' відкрийте архів 'KA - KASKAD'.



Система підгрузить лише мінімальні значення, які мали змінні протягом кожних 2-х хвилин. До імені змінної допишеться закінчення [MI] – мінімальне значення.

При інших типах каскадних архівів допишеться [MA] – максимум, [MW] – середнє значення, [SU] – сума. Спробуйте самостійно опрацювати ці варіанти.

Для знайомства з пакетним архівуванням та збереженням даних в SQL-базу даних зверніться до технічної документації [Historian](#).

## Тренди

З метою аналізу і зручного візуального сприйняття, виміряну інформацію представляють у вигляді графіків – трендів. На відміну від вже розглянутого нами трендового елементу, який має досить примітивну функціональність, zenon дозволяє створювати повноцінні трендові екрани з великою кількістю налаштувань. Трендові криві можуть будуватись на основі як онлайн так і архівних даних.

*Базова версія ліцензії zenon містить стартову редакцію модуля трендів. Для отримання повної функціональності, необхідно придбати відповідну ліцензію.*

### Створення екрану трендів

Давайте створимо екран відображення графіків наших архівних змінних – температури та рівня.

- ▶ В дереві проекту зайдіть в розділ *Screens* і на базі фрейму 'Головний' створіть новий екран 'Тренди'
- ▶ В якості спеціального типу виберіть *Extended Trend* і задайте довільний колір фону
- ▶ Через головне меню *Control elements*, використовуючи кнопочні шаблони, розмістіть на екрані усі необхідні елементи керування та відображення

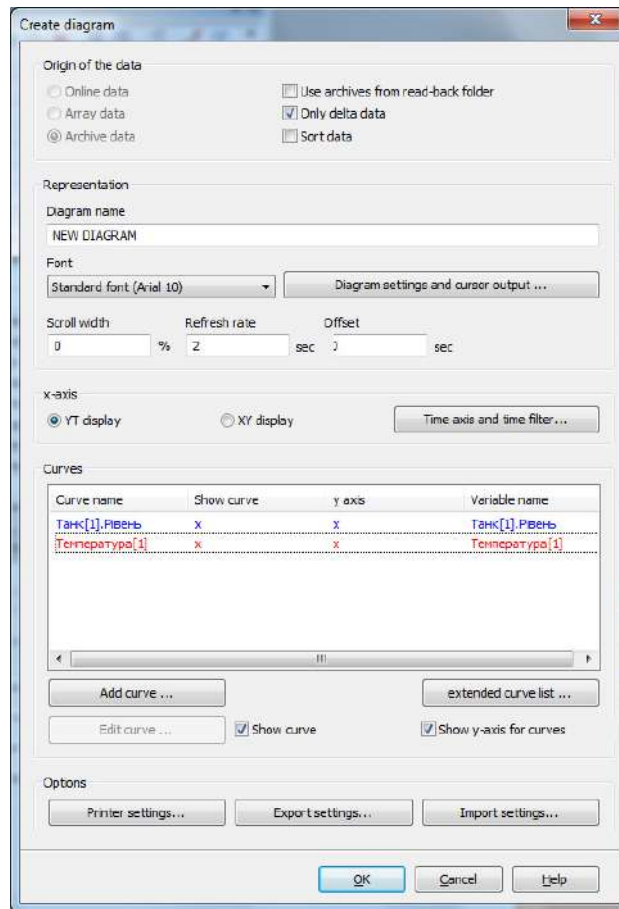
### Конфігурування функції

При створенні екрану трендів найважливіше значення має конфігурування функції перемикання.

- ▶ В дереві проекту зайдіть в розділ *Functions* і правою кнопкою миші створіть нову функцію *Screen switch* і вкажіть екран 'Тренди'

Автоматично з'явиться діалог конфігурування функції, через який здійснюється вибір кривих, що будуть відображатись та їх налаштування.

- ▶ Для поля *Origin of the data* виберіть *Archive data*
- ▶ *Refresh rate* встановіть в *2 sec*
- ▶ Активуйте опцію *YT-display*
- ▶ Натисніть кнопку *Add curve ...* і додайте з архіву змінну 'Танк[1].Рівень' та 'Температура[1]'
- ▶ Для кожної кривої проконтролюйте щоб були активовані бокси *Show curve* та *Show y-axis for curves*



Давайте налаштуємо представлення часової вісі діаграми.

- Натисніть кнопку *Time axis and time filter*

З'явиться додатковий діалог конфігурування з вкладками:



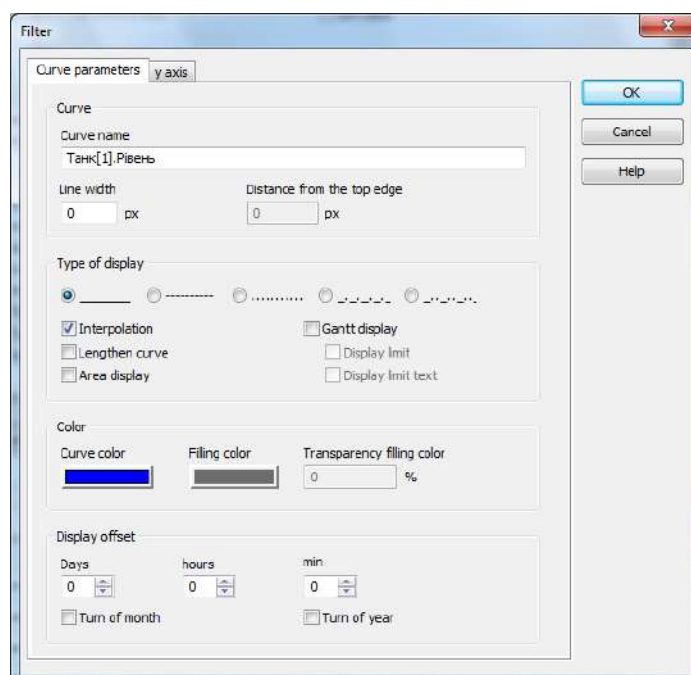
- ▶ На вкладці *x-axis* активуйте *Show vertical grid lines* і залиште підписи для годин хвилин та секунд
- ▶ На вкладці *Time* залиште *Relative time period* і встановіть період представлення шкали в 10 хвилин
- ▶ Закрийте вікно кнопкою *OK*

Тепер нам необхідно сконфігурувати окремо кожну криву.

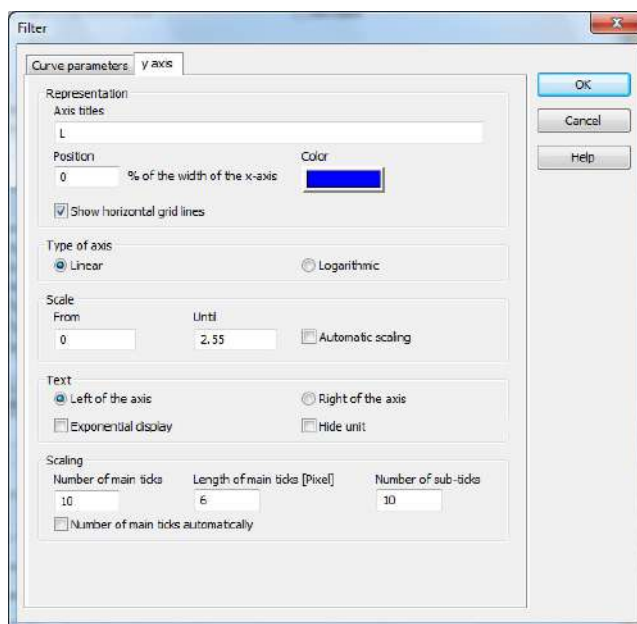
- ▶ В полі *Curves* двічі клікніть на кривій 'Танк[1].Рівень'

З'явиться додатковий діалог конфігурування з двома вкладками налаштування параметрів кривої та вісі ординат.

- ▶ На вкладці *Curve parameters* виберіть суцільну лінію представлення та активуйте інтерполяцію
- ▶ Колір кривої встановіть синій

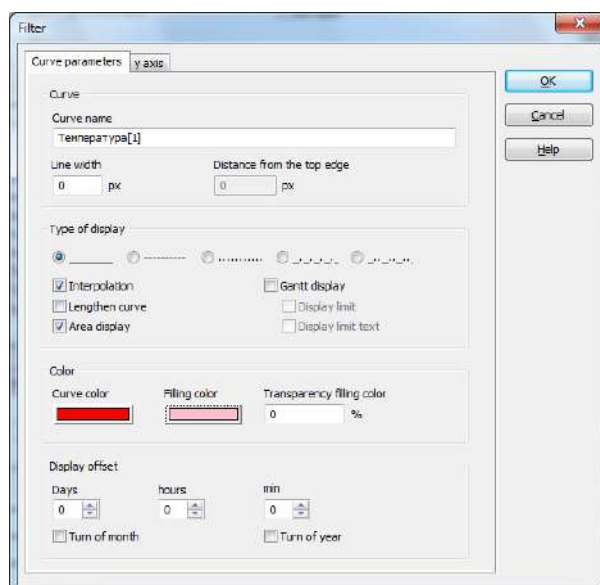


- ▶ На вкладці *y axis* задайте підпис вісі 'L',
- ▶ Колір вісі встановіть в синій
- ▶ Активуйте опцію *Show horizontal grid lines*
- ▶ Діапазон шкали залишіть як виставила система (0 – 2.55)
- ▶ Підписи шкали повинні відображатись з лівого боку
- ▶ Закрийте вікно кнопкою *OK*



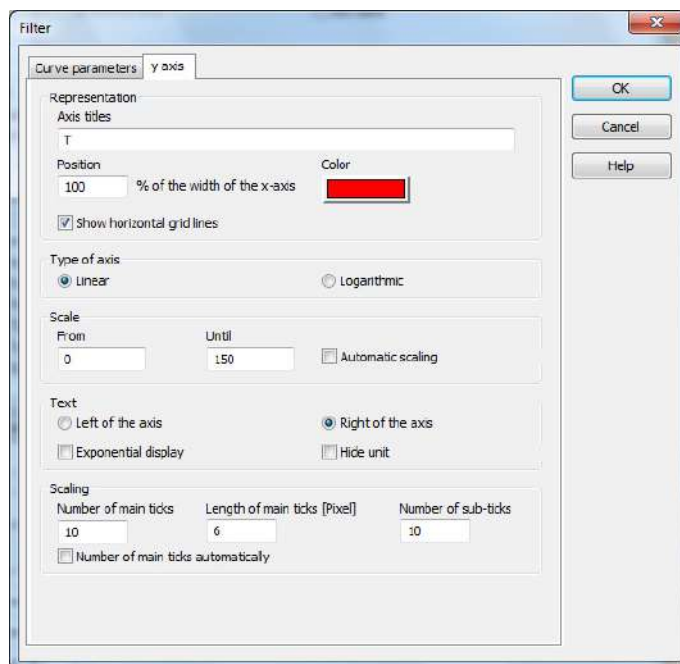
Повторимо зроблені операції для другої кривої.

- ▶ В полі *Curves* двічі клікніть на кривій 'Температура[1]'
- ▶ На вкладці *Curve parameters* виберіть суцільну лінію представлення, активуйте інтерполяцію та зафарбовування площини
- ▶ Встановіть колір кривої в червоний та рожеве зафарбовування площини під кривою



- ▶ На вкладці *y axis* задайте підпис вісі 'T',
- ▶ Позицію зміщення вісі виставте в 100 %

- ▶ Колір вісі встановіть в червоний
- ▶ Активуйте опцію *Show horizontal grid lines*
- ▶ Діапазон шкали залишіть як виставила система (0 – 150)
- ▶ Підписи шкали повинні відображатись з правого боку
- ▶ Закрийте вікно кнопкою *OK*



- ▶ Переіменуйте створену функцію в 'Перехід на екран Тренди'

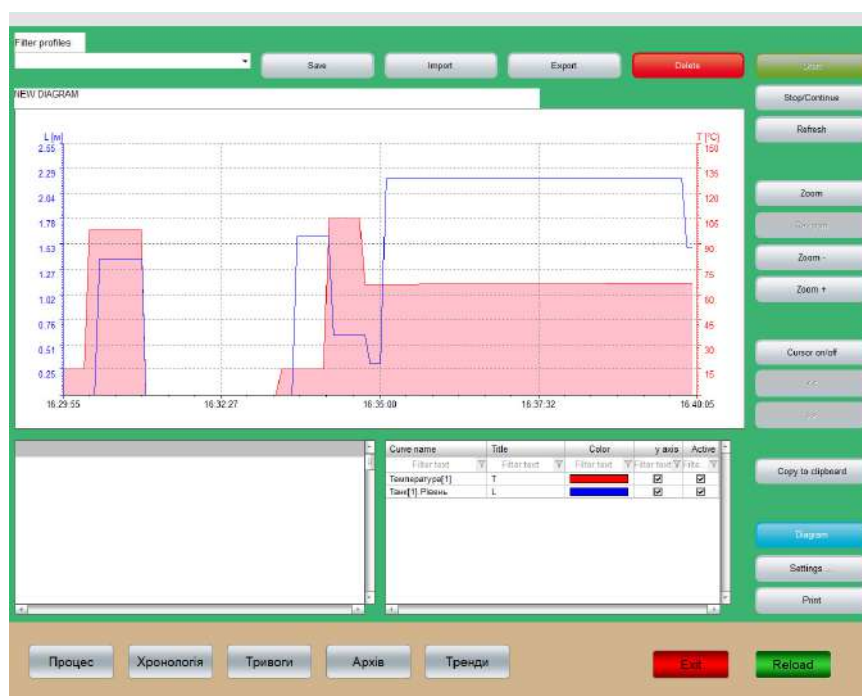
### Робота з екраном

- ▶ На екрані 'Навігація' створіть нову кнопку з ім'ям 'Тренди' і прив'яжіть відповідну функцію

Запустіть проект і перейдіть на екран трендів. У Вас повинен з'явитись екран з відображенням кривих рівня і температури з заданим часом поновлення діаграми – 2 секунди. Поверніться на основний екран процесу і просимулюйте хаотичну зміну значення трендових змінних.

Поверніться на екран трендів і перевірте функціонал наявних на ньому кнопок: сканування значень кривих, масштабування, активації (кривих та вісей), збереження індивідуальних конфігурацій в профілі.

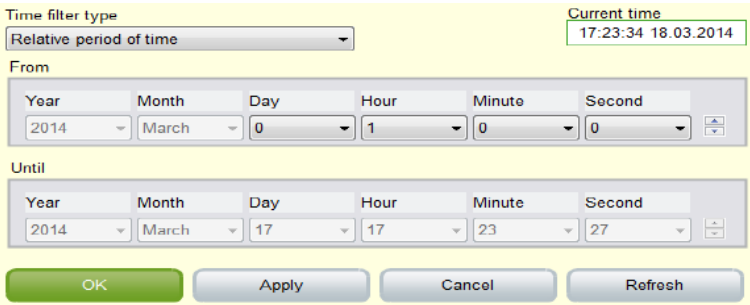
Також, оператор має можливість в режимі Runtime через кнопку 'Diagram' здійснювати додаткові налаштування відображення кривих.



Оскільки ми оперуємо архівними даними, то за допомогою кнопок переміщення по діаграмі можна виконувати прокрутку кривих в часі та їх аналіз.

### Часовий фільтр

По аналогії з архівами, за допомогою екрану зі спеціальним типом *Time filter*, можна в зручному вигляді здійснювати масштабування періоду відображення трендів. Цей екран фільтру повинен базуватись на своєму індивідуальному фреймі і викликатись разом з екраном трендів.



Якщо Ви вже створили цей екран та відповідну функцію при розгляді теми архівів, то Вам залишилось лише додати екран трендів у вже створену функцію виклику даного часового фільтру.

Докладніше про модуль трендів можна познайомитись в мануалі [Extended Trend](#).



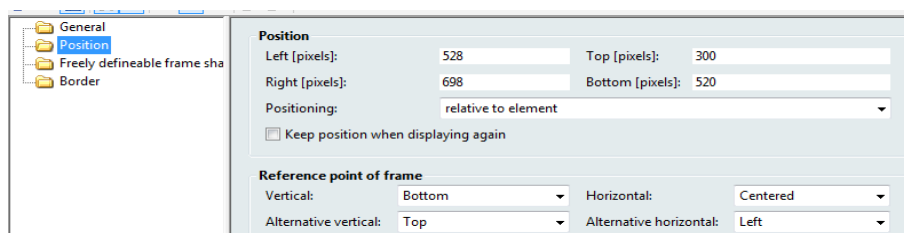
## Віртуальна клавіатура

Окрім тих варіантів установки значень змінним оператором, що ми вже розглянули, існує ще один дуже розповсюджений варіант – введення через віртуальну клавіатуру.

Давайте зробимо віртуальну цифрову клавіатуру, яку ми використаємо для вводу значення для цифрового елементу 4-ї температури.

- ▶ В менеджері проекту виберіть вузол *Screens* -> *Frames*
- ▶ Створіть новий фрейм 'Клавіатура' розміром 220 x 170
- ▶ У властивості *Position* -> *Positioning* зі спадного меню виберіть *relative to element*
- ▶ У властивості *Position* -> *Reference point of frame* для вертикальної точки прив'язки виберіть *Bottom*, а для горизонтальної - *Centered*

Таким чином, фрейм клавіатури буде з'являтися не в тому місці, де він розташований в конструкторі фреймів, а безпосередньо поруч з тим елементом в який здійснюється введення значення. Усі ніші налаштування залишіть без змін.



- ▶ В менеджері проекту виберіть вузол *Screens* і створіть новий екран 'Клавіатура'
- ▶ У властивості *General* -> *Frame* вкажіть створений нами фрейм та виставте сірий колір фону
- ▶ У властивості *General* -> *Screen type* виберіть спеціальний тип екрану – *Keyboard*

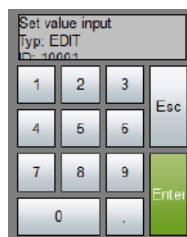
Тепер необхідно на цей екран розмістити усі необхідні елементи керування.

- ▶ В головному меню *Control elements* виберіть *Insert template*
- ▶ Серед запропонованих варіантів наборів кнопок виберіть *Scada - Numeric telephone*
- ▶ З головного меню *Control elements* у верхній частині екрану додатково розмістіть елемент вводу значень - *Set value element*

- У властивостях екрану *Window style* активуйте бокс *Modal dialog*

Таким чином, до тих пір поки відкрите діалогове вікно вводу значень через клавіатуру, жодні інші дії на екрані оператор не зможе зробити.

Ваша віртуальна клавіатура повинна мати наступний вигляд:

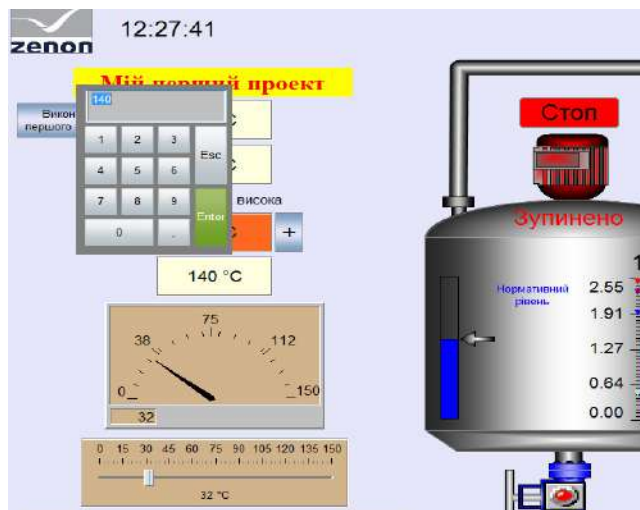


При необхідності відредагуйте зовнішній вигляд та розмір елементів.

Нам залишилось прив'язати цю клавіатуру до цифрового елемента.

- На екрані 'Процес' виберіть четвертий цифровий елемент зі змінною 'Температура[4]'
- У властивості *Write set value* активуйте бокс *Setting values active*
- Для опції *Write set value via* встановіть *Dialogbox*
- Активуйте опцію *Use screen Keyboard* та виберіть створену нами клавіатуру

Запустіть Runtime і перевірте зроблену роботу.



Докладніше про створення і використання екранних клавіатур можна познайомитись в мануалі [Keyboard](#).

## Адміністрування користувачів

Система керування дозволяє реалізувати ряд заходів для забезпечення найвищого рівня безпеки при керуванні процесом виробництва. Одним з таких заходів являється впровадження в проєкті багаторівневої системи адміністрування доступу користувачів.

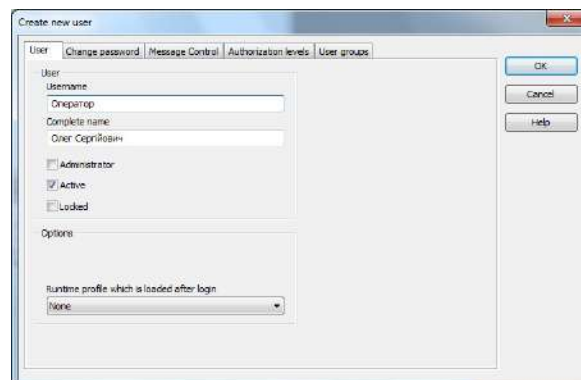
### Визначення користувачів

Для опрацювання системи адміністрування ми створимо двох користувачів: оператора та інженера, які будуть мати різні рівні доступу/авторизації.

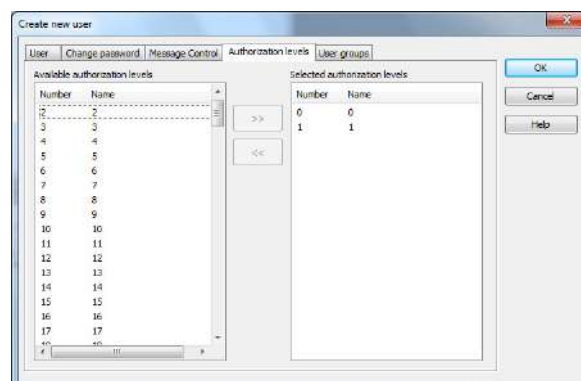
- ▶ В менеджері проєкту виберіть вузол *User administration* -> *User*
- ▶ Правою кнопкою миші створіть нового користувача

Автоматично з'явиться діалогове вікно налаштувань з набором вкладок.

- ▶ На вкладці *User* введіть ім'я 'Оператор' та довільне повне ім'я 'Олег Сергійович'



- ▶ На вкладці *Change password* введіть 6-ти значний пароль
- ▶ На вкладці *Authorization levels* для цього користувача визначіть рівень доступу 1



*В zenon рівні доступу не мають пріоритетності. Тобто, рівень 1 має такий же пріоритет як рівень 15. Один користувач може мати необмежену кількість рівнів (макс. 128).*

- ▶ Завершіть конфігурування кнопкою **OK**

У вікні детального перегляду тепер з'явився створений нами користувач. Усі налаштування можна у будь-який момент змінити через вікно властивостей.

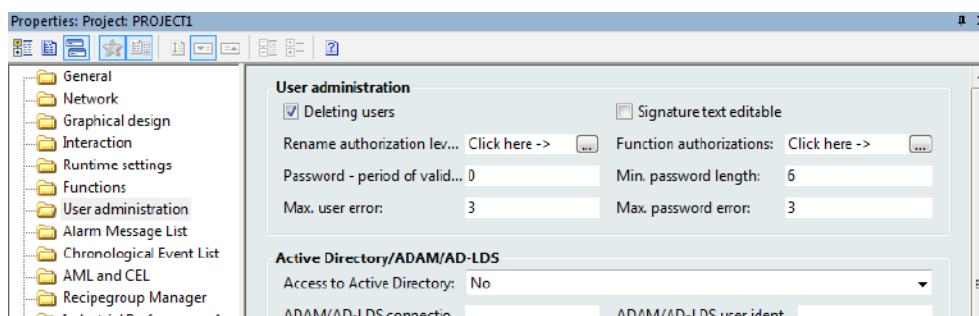
- ▶ Створіть нового користувача 'Інженер' з довільним повним ім'ям 'Олександр Анатолійович'
- ▶ Також активуйте опцію *Administrator*

*Права адміністратора дають можливість в Runtime блокувати чи розблоковувати користувачів (у випадку, якщо тричі було введено невірний пароль), створювати нових користувачів чи видаляти існуючих.*

- ▶ На вкладці *Change password* введіть 6-ти значний пароль
- ▶ На вкладці *Authorization levels* для цього користувача визначіть рівень доступу 1 та 2

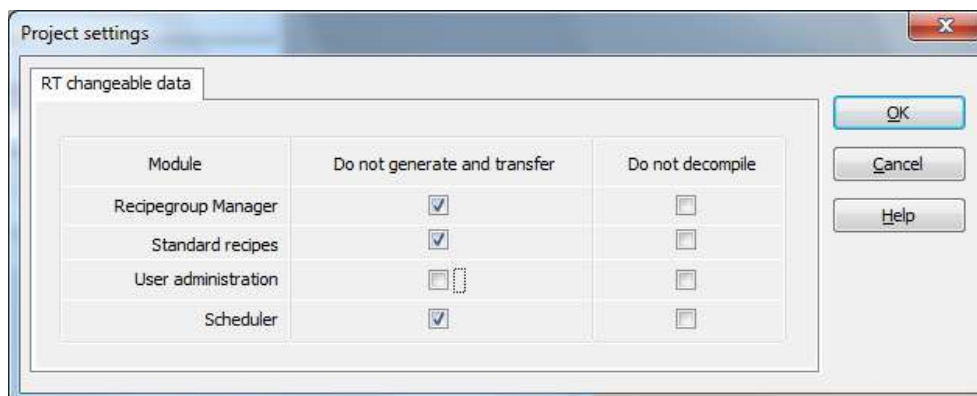
### Додаткові налаштування

У властивостях проекту існує спеціальний розділ - *User administration*, в якому можна визначити максимальну кількість разів невірного вводу паролю чи імені, мінімальну довжину паролю, переіменувати рівні авторизації та багато інших опцій. Пізніше ми сюди ще повернемося.



В zenon існує захисний механізм від втрати даних, які користувач-адміністратор додав в систему в режимі Runtime, а саме – нових користувачів. Ця опція називається *RT changeable data*. Якщо вона активна, то при компілюванні проекту файл 'password.cmp' не буде перезаписуватись і створені в середовищі розробки користувачі не «працюватимуть» в Runtime. Нас це не влаштовує. Тому цю функцію необхідно деактивувати.

- ▶ У властивостях проекту зайдіть в *General -> RT changeable data* і зніміть галочку з пункту *User administration*



Після компілювання проекту цю галочку можна повернути назад.

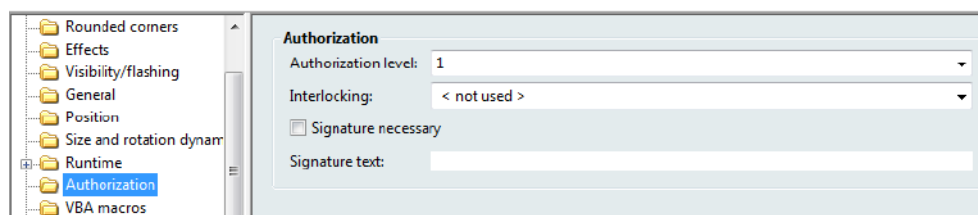
### Прив'язка рівнів доступу

Розділення прав доступу в zenon здійснюється за рахунок визначення різних рівнів авторизації до елементів керування – кнопок чи задатчиків значень. Таким чином, якщо, наприклад, користувач не уповноважений перейти на той чи інший екран, система не дозволить йому це зробити.

Для нашого проекту давайте розмежуємо права користувачів таким чином:

Користувач без авторизації	Доступ лише на екран 'Процес', без можливості встановлення значень усім температурним змінним.  Рівень авторизації - 0
Оператор	Доступ на всі екрани та встановлення значень всім змінним. Без можливості використання системних кнопок 'Exit' та 'Reload'.  Рівень авторизації – 0, 1
Інженер	Повний доступ.  Рівень авторизації – 0, 1, 2

- ▶ Відкрийте екран 'Навігація'
- ▶ Зайдіть у властивості кнопки 'Хронологія' *Authorization* -> *Authorization level* і встановіть рівень 1



- ▶ Аналогічну операцію виконайте для кнопок 'Тривоги', 'Архів' і 'Тренди'.

*Для прискорення виконання однотипних операцій, через Ctrl виділіть усі кнопки та централізовано змініть відповідну властивість.*

- ▶ Для кнопок 'Exit' та 'Reload' встановть рівень доступу 2
- ▶ Відкрийте екран 'Процес'
- ▶ Зайдіть у властивості цифрового елементу температурних змінних 1,2,4 *Authorization* -> *Authorization level* і встановть рівень 1
- ▶ Аналогічну операцію виконайте для кнопок встановлення значення для змінної Температура[3]

На даному етапі ми вже можемо перевірити зроблену роботу. Запустіть Runtime і натисніть кнопку 'Хронологія'. Система автоматично видасть діалог авторизації:



*Це так званий стандартний діалог тимчасової авторизації. Він діє одноразово. Якщо Ви захочете знову задіяти елемент, який має авторизацію, то це вікно з'явиться знову.*

*При старті проекту система автоматично запускає його як користувач 'SYSTEM' з рівнем авторизації 0.*

У відповідні поля введіть ідентифікатор створеного нами раніше користувача 'Оператор' і його пароль. Відкриється екран Хронологічного списку подій, до списку якого вже буде занесено зроблену нами процедуру авторизації.

Зазвичай така тимчасова процедура авторизації використовується в окремих конкретно визначених випадках і для керування цілим проектом є неприйнятною.

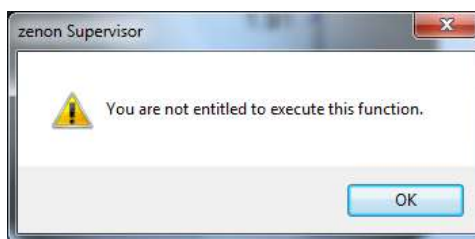
## Створення екрану авторизації

В проектах виконують постійну авторизацію через спеціально створене для цього діалогове вікно чи екран.

Щоб стандартний діалог тимчасової авторизації більше не з'являвся нам необхідно деактивувати цю процедуру

- ▶ Зайдіть у властивості проекту *User administration* -> *Temporary login* і деактивуйте опцію *Temp. login active*

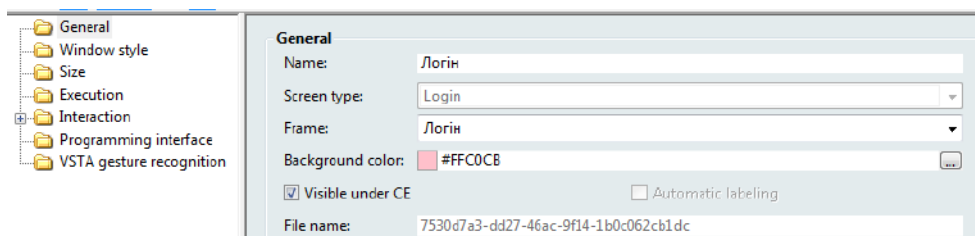
Тепер замість стандартного діалогу тимчасової авторизації, при спробі перейти на екран хронології, буде з'являтися системне повідомлення про те, що Ви не маєте права виконати цю операцію і необхідно виконати процедуру авторизації.



*Якщо для середовища розробки zenon у Вас вибрано російський інтерфейс, то це повідомлення буде представлене на російській мові. Через інструмент інтернаціоналізації, який ми розглянемо пізніше, Ви зможете подавати всі текстові підписи в проекті на будь-якій необхідній мові.*

Давайте створимо екран авторизації, через який користувач буде проводити процедуру логування в системі.

- ▶ В менеджері проекту виберіть вузол *Screens* -> *Frames* і створіть новий фрейм 'Логін' розміром 340 x 220 і розмістіть його в правій нижній частині екрану
- ▶ На базі цього фрейму створіть новий екран 'Логін' і в якості спеціального типу вкажіть для нього *Login*



- ▶ Зайдіть у вікно властивостей створеного екрану *Window style* і активуйте опцію *Modal dialog*

- ▶ Через головне меню *Control elements* розмістіть на екрані стандартний кнопочний шаблон
- ▶ Підредагуйте розміри елементів згідно розмірів екрану

### Робота з екраном авторизації

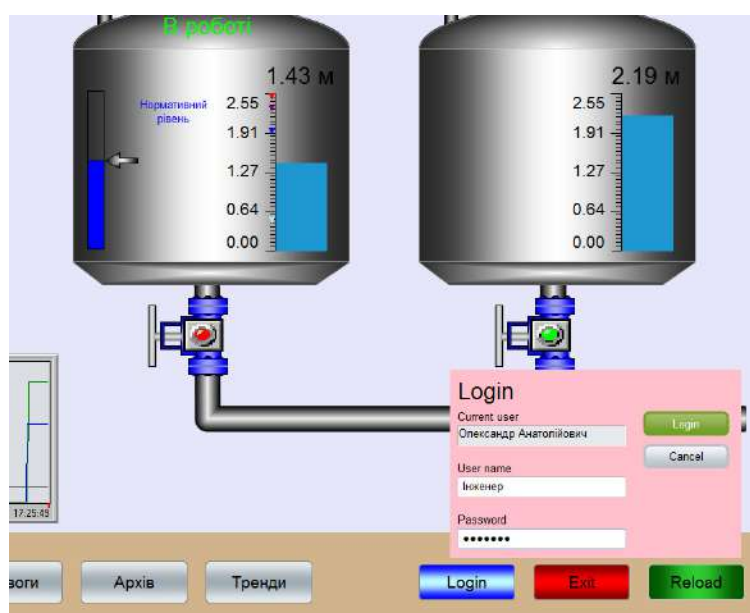
Тепер нам необхідно створити на панелі навігації кнопку виклику діалогового вікна авторизації і прив'язати до неї функцію перемикавання.

- ▶ На екрані 'Навігація' створіть нову кнопку з ім'ям 'Login', задайте їй синій колір і розмістіть поруч з іншими системними кнопками
- ▶ Проконтролюйте, щоб у цієї кнопки був рівень авторизації – 0



- ▶ В менеджері проекту створіть нову функцію *Screen switch* на екран 'Логін'
- ▶ Назвіть нову функцію 'Перехід на екран Логін' і прив'яжіть до відповідної кнопки на екрані навігації

На цьому створення екрану логонування можна вважати завершеним. Запустіть Runtime і натисніть кнопку з ім'ям 'Login'. З'явиться наступне вікно авторизації:



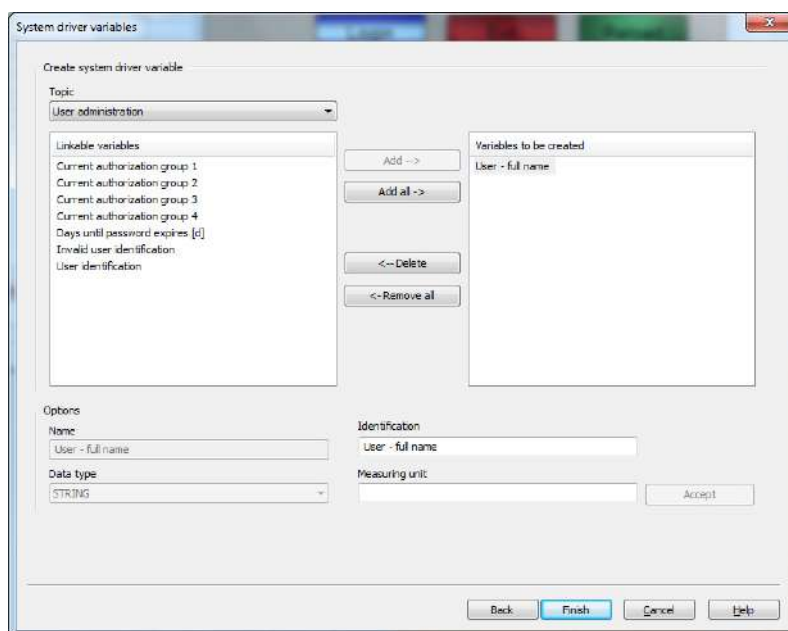


Після реалізації системи доступу, перевірте роботу усіх елементів керування, яким ми визначили різні рівні доступу: без авторизації користувача, з авторизацією як 'Оператор' і 'Інженер'.

### Візуалізація авторизації

Через системну змінну в проекті можна представити оператору інформацію про поточного авторизованого користувача. Давайте зробимо це.

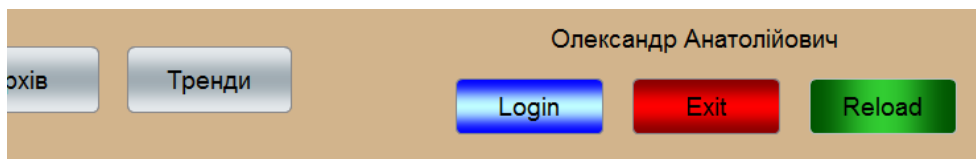
- ▶ Створіть нову змінну
- ▶ В якості драйверу виберіть системний – SYSDRV і натисніть кнопку *Next*
- ▶ З'явиться діалогове вікно налаштувань, в якому виберіть пункт *User administration -> User – full name*
- ▶ Кнопкою *Add ->* додайте цю змінну і закрийте діалог кнопкою *Finish*



Таким чином, в переліку змінних в нас з'явиться нова - 'User – full name', яка має наперед визначену функціональність та призначення.

- ▶ На екрані 'Навігація' над системними кнопками розмістіть елемент *Dynamic text*, який можна знайти на інструментальній панелі елементів
- ▶ Прив'яжіть цьому елементу створену нами змінну 'User – full name' і зробіть прозорим колір фону

Запустіть проект та авторизуйтеся як 'Інженер'. Зроблене поле відображає повне ім'я. Тепер в процесі роботи Ви будете бачити який користувач в даний момент є активним і відповідно знати які він має права доступу.



До послуг розробника існують іще деякі системні змінні, які Ви можете протестувати самостійно.

Для зручного сприйняття оператором елементів керування для яких потрібна авторизація, система дозволяє представити їх в декількох графічних варіантах: сірим кольором, з графічною позначкою в кутку (наприклад, замок чи інша іконка) або взагалі зробити їх невидимими.

- ▶ Зайдіть у властивості проекту *User administration* -> *Temporary login* -> *Locked buttons* і виберіть зі спадного меню *gray*
- ▶ Зайдіть у властивості проекту *Graphical design* -> *Locked/Interlocked elements* і активуйте бокс *Graphical identification active*

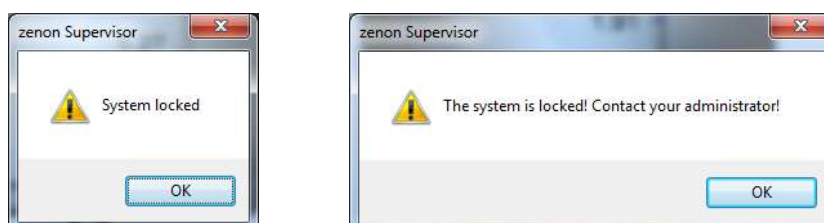


Для того, щоб графічне зображення замка в нас не плуталось з зображенням заблокованих елементів технологією блокувань, яку ми розглянемо пізніше, давайте встановимо усі елементи, для яких потрібна авторизація, в сірий колір, а виконаний нами другий пункт відмініть.

### Функції адміністратора

Як ми вже згадували, серед усіх користувачів, створених в проекті, обов'язково повинен бути один з правами адміністратора (в нашому проекті - 'Інженер').

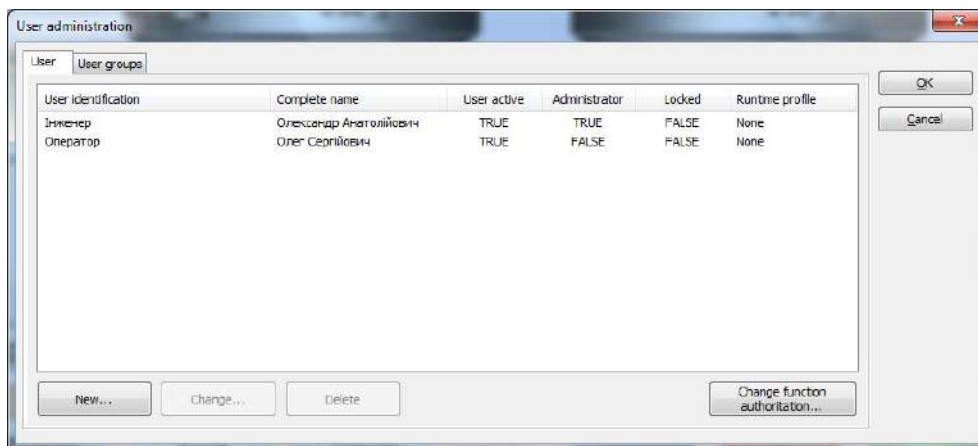
Спробуйте тричі ввести невірний пароль чи ім'я. Система заблокує Вас, а розблокувати зможе тільки адміністратор - 'Інженер'.



Також адміністратор може в онлайн режимі додавати нових користувачів. Давайте перевіримо як це працює.

- ▶ Створіть нову функцію *User administration* -> *Change user* і назвіть 'Редагування користувачів'
- ▶ На екрані 'Логін' розмістіть ще одну кнопку з ім'ям 'User edit' і прив'яжіть створену функцію

Запустіть проект і авторизуйтеся з правами адміністратора. Натисніть кнопку 'User edit'. З'явиться наступне вікно:



Якщо Ви маєте права адміністратора, в онлайн режимі Ви можете видаляти користувачів, створювати нових, редагувати (змінювати паролі та рівні доступу), а також тимчасово їх блокувати. В нашому випадку, якщо Ви авторизуєтесь як 'Оператор' система не дозволить виконати такі дії.

Давайте зараз створимо ще одного користувача.

- ▶ Натисніть кнопку *New...* і створіть нового користувача з такими даними:

Username: Директор

Complete name: Георгій Миколайович

Authorization levels: 0, 1

*Зверніть увагу на те, що адміністратор не може створити нового користувача з більшими рівнями доступу, ніж він має сам.*

*Також, під час першого логину система запропонує користувачу 'Директор' змінити пароль. Таким чином адміністратор не знає які паролі мають активні користувачі.*

Оскільки користувач 'Директор' має рівень доступу 0 і 1, то системні кнопки 'Exit' і 'Reload' будуть недоступні, аналогічно до прав користувача 'Оператор'.

Таким чином, в онлайн режимі можна додати необмежену кількість нових користувачів. Внесені дані будуть зберігатися в Runtime-файлі 'password.cmp'. В базі даних проекту цієї інформації немає, тому якщо в середовищі розробки

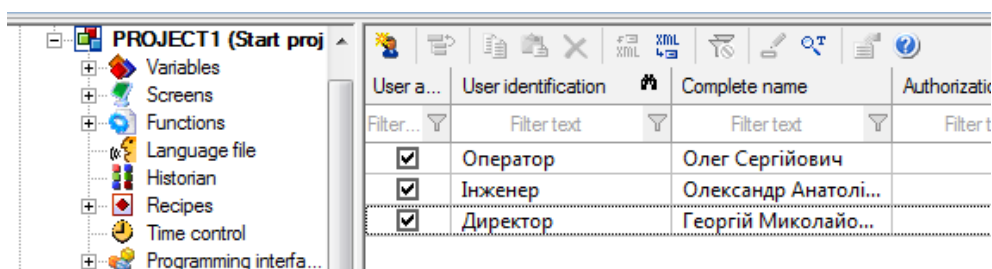
виконати компілювання проекту при знятій галочці *RT changeable data* -> *User administration*, то внесена в онлайн режим інформація буде втрачена.

В середовищі розробки проекту існує технологія, яка дозволяє виконати зворотнє зчитування Runtime-даних в базу проекту: інструментальна панель *Runtime files* -> *Import Runtime files*.



- ▶ На панелі інструментів натисніть іконку *Import Runtime files*

В результаті, створений нами користувач 'Директор' з'явиться в середовищі розробки проекту.



Самостійно промодельуйте виконання адміністратором блокування в онлайн режимі користувача 'Оператор'.

Докладніше про систему адміністрування можна познайомитись в мануалі [User administration](#).

## Меню

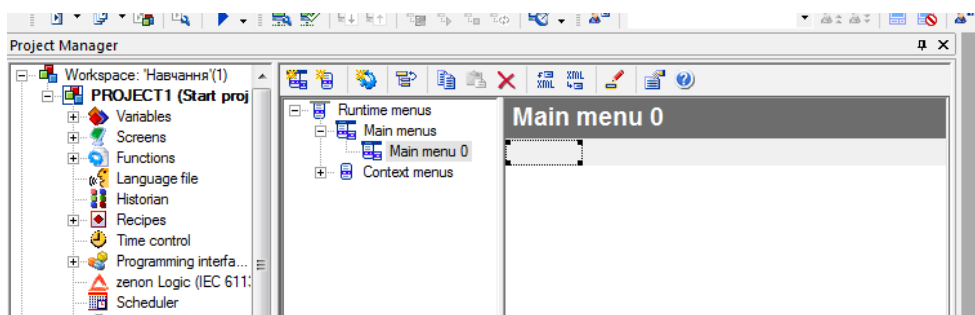
У випадку, якщо проект має велику кількість елементів керування – кнопок, то для оптимізації користувацького інтерфейсу логічно створити меню. zenon дозволяє створювати меню аналогічні до тих, які є в програмах Windows: головне меню, яке знаходиться в верхній частині екрану та контекстне меню, яке викликається при натисненні правою кнопкою миші на елементі інтерфейсу.

### ГОЛОВНЕ МЕНЮ

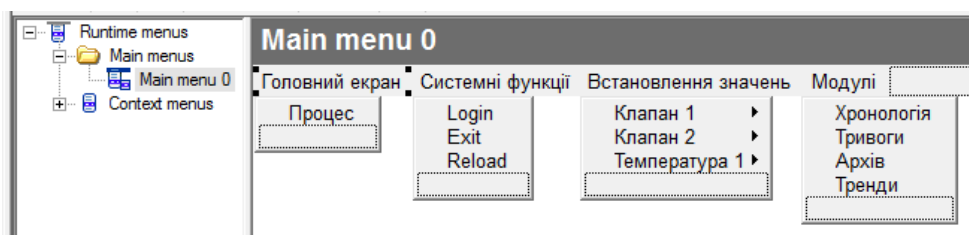
Давайте створимо головне меню, яке буде дублювати кнопки, які ми маємо на екрані навігації.

- В дереві проекту активуйте пункт *Menus* і через праву кнопку мишки виберіть пункт *New main menu*

У вікні детального перегляду з'явиться новий пункт – *Main menu 0*. Після його вибору у вікні конструктора відобразиться поле конфігурування пунктів та структури нашого головного меню.



Поступово сформуєте розділи та пункти головного меню, які мають такі елементи, як показано на малюнку нижче.

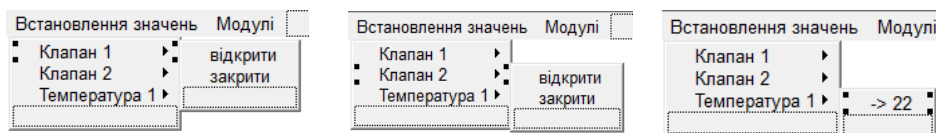


Також зробіть, щоб кожен пункт розділу 'Встановлення значень', мав свої власні підпункти.

- Виділіть пункт 'Клапан 1' і у властивостях *Representation / Type* активуйте бокс *Submenu*

В конструкторі меню автоматично з'явиться новий підпункт.

- ▶ Створіть нові підпункти, як показано на малюнку нижче
- ▶ Повторіть описані вище дії для пункту 'Клапан 2' та 'Температура 1'



## Визначення дій

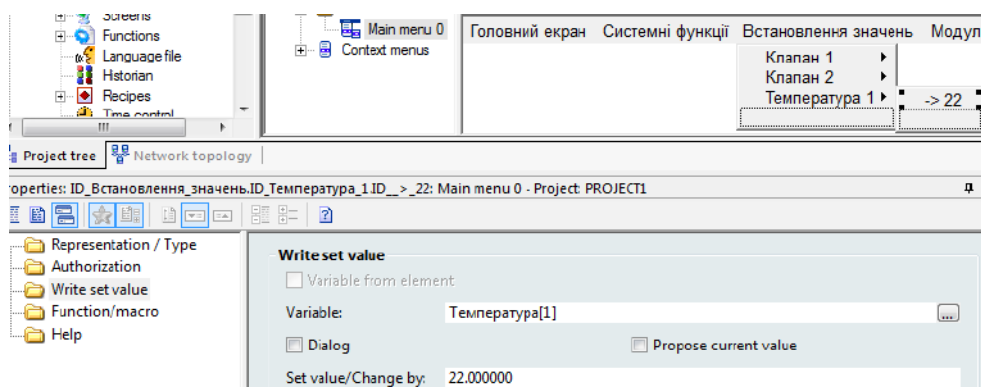
Для кожного пункту меню можна визначити три дії: виконання функції, встановлення певного значення змінній та виклик файлу допомоги.

- ▶ Виділіть пункт 'Процес' розділу меню 'Головний екран'
- ▶ У вікні властивостей зайдіть в *Representation / Type* -> *Action type*: і зі спадного меню виберіть *Function*
- ▶ В розділі властивостей *Function/ macro* прив'яжіть функцію 'Перехід на екран Процес'

Аналогічну операцію виконайте для пунктів розділу меню 'Системні функції' та 'Модулі'. Для пунктів розділу 'Встановлення значень' визначіть дію встановлення значень змінним.

- ▶ Виділіть пункт 'Клапан 1' -> 'відкрити' розділу меню 'Встановлення значень'
- ▶ У вікні властивостей зайдіть в *Representation / Type* -> *Action type*: і зі спадного меню виберіть *Write set value*
- ▶ В розділі властивостей *Write set value* -> *Variable*: виберіть змінну 'Танк[1].Клапан[1]', а також для опції *Set value/Change by*: встановіть значення '1'
- ▶ Для пункту 'Клапан 1' -> 'закрити' повторіть описані вище дії, але значення встановіть в '0'

Аналогічні операції виконайте для пункту 'Клапан 2' та 'Температура 1'. Для останнього встановіть значення в '22'.



Таким чином, через головне меню, окрім перемикання між екранами ми також зможемо відкривати/закривати клапани та встановлювати конкретне значення для першої температури.

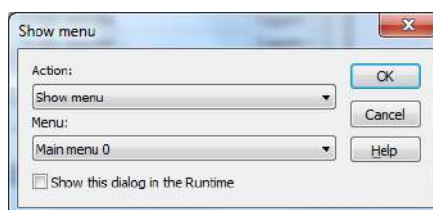
Також необхідно прив'язати відповідні рівні доступу до пунктів меню. Зробіть це на власний розсуд через властивість *Authorization* -> *Authorization level*.

На цьому конфігурування пунктів меню завершено.

### Активация головного меню

Лише створити меню не достатньо. Необхідно його активувати.

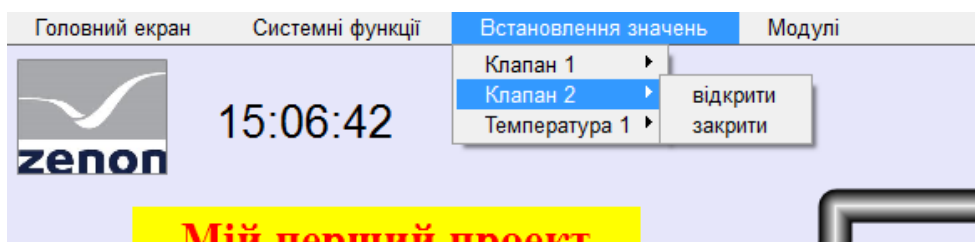
- ▶ У вікні властивостей проекту зайдіть в розділ *Graphical design* -> *Runtime general* і активуйте бокс *Main menus active*
- ▶ Створіть нову функцію *Screens* -> *Show menu*
- ▶ В діалоговому вікні налаштувань функції вкажіть створене нами меню.



- ▶ Задайте цій функції ім'я 'Відображення головного меню' і додайте її в скрипт 'AUTOSTART'

Тепер даний скрипт містить п'ять функцій, які будуть виконуватись при запуску проекту, у тому числі і запуск головного меню.

*Головне меню з'являється у фіксованому місці – верхнє поле шириною 20 пікс. Як пам'ятаєте, ми з самого початку при конфігуруванні розташування фреймів екранів залишили зверху вільне місце. Якраз це місце в Runtime буде займати головне меню. Якщо Ви так не зробите, то відкритий екран просто закrije собою створене меню.*



Запустіть Runtime і перевірте роботу меню.

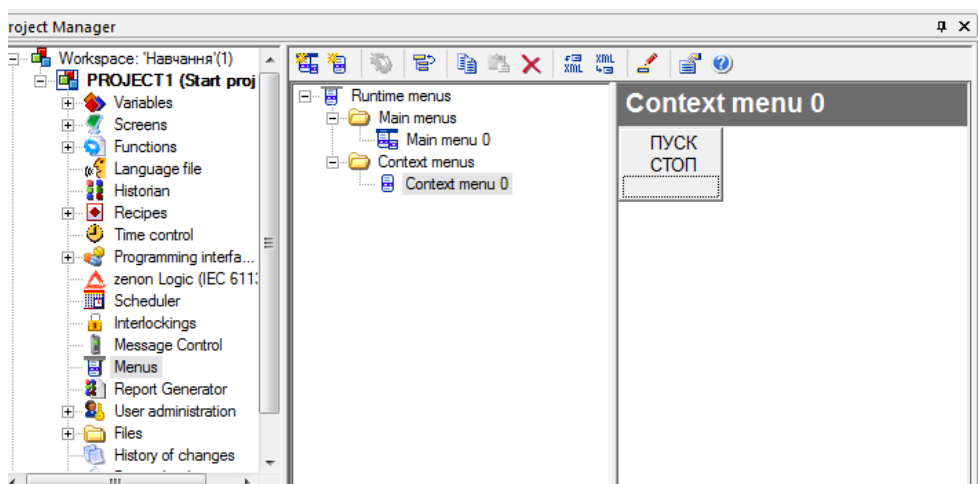
## КОНТЕКСТНЕ МЕНЮ

Давайте зробимо так, щоб другий танк запускався кнопкою лише через контекстне меню при натисненні на ній правою кнопкою мишки.

- ▶ В дереві проекту активуйте пункт *Menus* і через праву кнопку мишки виберіть пункт *New context menu*

У вікні детального перегляду з'явиться новий пункт – *Context menu 0*. Після його вибору у вікні конструктора відобразиться поле конфігурування пунктів та структури нашого контекстного меню.

- ▶ Створіть пункти 'ПУСК' та 'СТОП'



- ▶ Виділіть пункт 'ПУСК' і у вікні властивостей зайдіть в розділ *Representation / Type* -> *Action type*:
- ▶ Зі спадного меню виберіть тип дії *Write set value*
- ▶ В розділі властивостей *Write set value* -> *Variable*: виберіть змінну 'Танк[2].Пуск\Стоп', а також для опції *Set value/Change by*: встановіть значення '1'
- ▶ Для пункту 'СТОП' повторіть описані вище дії, але значення встановіть в '0'
- ▶ Відкрийте екран 'Процес' і виділіть комбоелемент пуску другого танку зі змінною 'Танк[2].Пуск\Стоп'
- ▶ В меню властивостей зайдіть в розділ *Runtime* -> *Context menu*: і зі спадного меню виберіть створене нами контекстне меню 'Context menu 0'

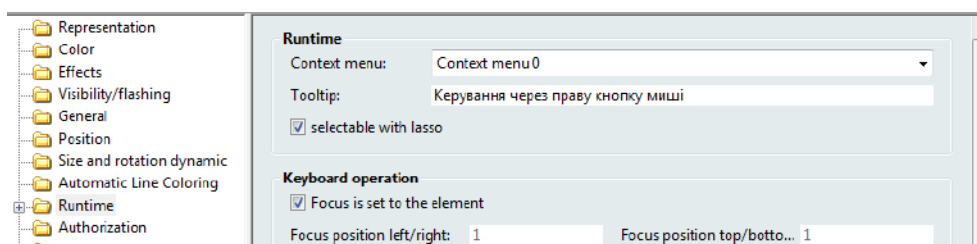
Незабудьте деактивувати в цьому елементі можливість керування звичайним методом – по натисненню лівої кнопки миші.



- ▶ В меню властивостей зайдіть в розділ *Write set value -> Binary value* і зніміть галочку з боксу *Switch*.

В zenon є функціонал так званої інтерактивної допомоги, коли при наведенні на елемент керування мишки, оператору з'являється підказка. До цього часу ми цю можливість не використовували. Давайте розглянемо її.

- ▶ Повторно виділіть комбоелемент пуску другого танку і в меню властивостей зайдіть в розділ *Runtime -> Tooltip*:
- ▶ В пустому полі введіть текст підказки: 'Керування через праву кнопку миші'



Запустіть Runtime і перевірте як працює контекстне меню та інтерактивна підказка.



Докладніше про створення меню можна познайомитись в мануалі [Menus](#).

## Блокування

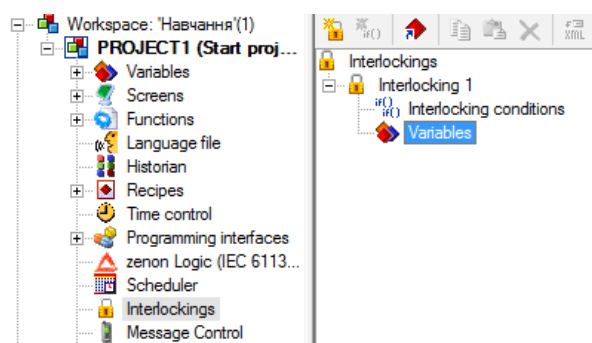
Існують задачі, коли з точки зору безпеки чи згідно з технологією необхідно заблокувати ті чи інші елементи екрану, щоб оператор не міг ними керувати. Інколи цю задачу реалізують на контролері, але з zenon таке блокування можна створити безпосередньо в проєкті візуалізації використовуючи стандартну функціональність.

Для нашого тестового проєкту ми реалізуємо наступне блокування: якщо змінна вологості повітря в приміщенні має значення більше за 75 %, то кнопка пуску першого танку буде недоступна для керування - заблокована.

### Визначення блокування

- ▶ В менеджері проєкту активуйте пункт *Interlockings* і через праву кнопку мишки виберіть пункт *New interlocking*

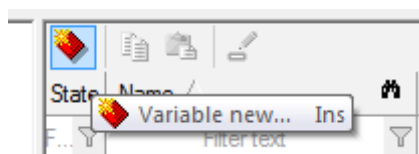
У вікні детального перегляду з'явиться новий пункт – *Interlocking 1* з елементами визначення умови блокування та змінної блокування.



- ▶ На елементі *Variables* натисніть правою кнопкою миші і виберіть *Add variable...*

В діалоговому вікні, що з'явилося, необхідно вибрати змінну. Оскільки змінної вологості в нас ще немає, ми можемо її створити одразу з цього вікна.

- ▶ В верхній частині виберіть піктограму *Variable new...* і створіть змінну 'Вологість' на базі драйверу VIPA 300 з типом даних INT



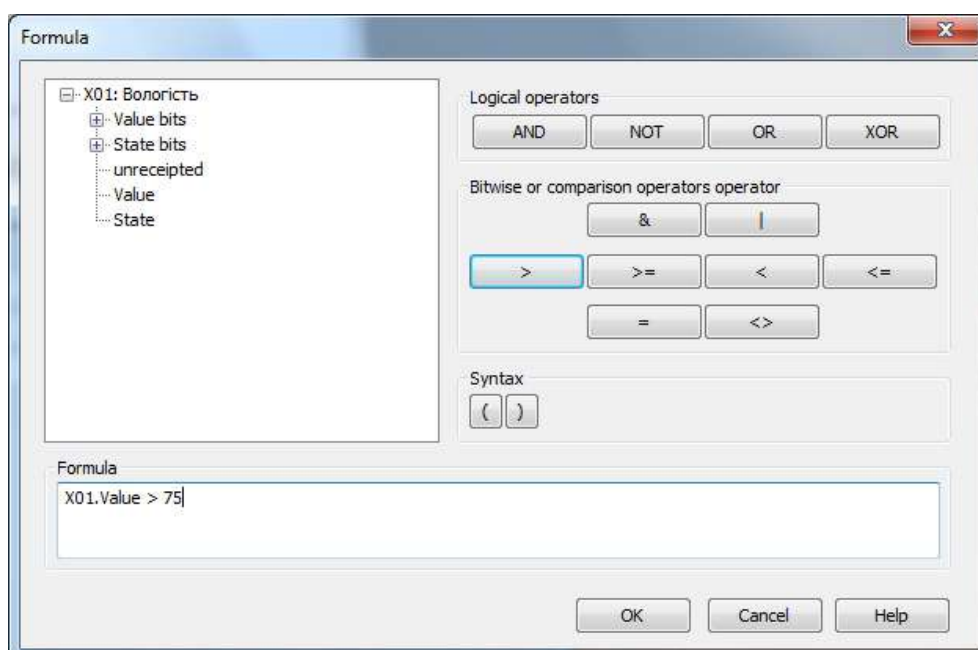
- ▶ На елементі *Interlocking conditions* натисніть правою кнопкою миші і виберіть *New interlocking condition*

*При необхідності в проекті можна визначити одночасно декілька умов блокувань по яким система буде проводити перевірку. Визначені тут умови можуть стосуватись декількох змінних.*

- У властивостях пункту *Condition 01* в полі *Logical link*: клікніть на полі визначення формули

З'явиться діалогове вікно формули прорахунку блокування.

- Виберіть пункт *Value* і сформуєте арифметичний вираз *X01.Value > 75*



- Після натиснення кнопки *OK* створення блокування буде завершеним

### Прив'язка до елементу

- На екрані 'Процес' виберіть кнопку 'Пуск' першого танку і у вікні властивостей для розділу *Authorization -> Interlocking* виберіть створене нами блокування 'Interlocking 1'

Оскільки система порівнює "сирі" значення змінної PLC зі встановленою нами величиною блокування в формулі, то необхідно, щоб діапазони значень змінної співпадали з реальним діапазоном вологості.

- Зайдіть у властивості змінної 'Вологість' *Value calculation -> Value range PLC* і встановіть його в діапазон 0 – 100
- Такий же діапазон встановіть для поля *Value adjustment linear*

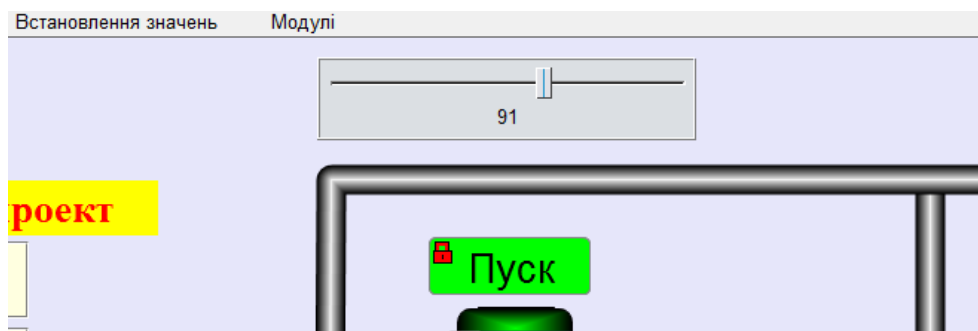
Давайте зробимо так, щоб оператор візуально бачив коли кнопка стає заблокованою.

- Зайдіть в розділ властивостей проекту *Graphical design* -> *Locked/Interlocked elements* і активуйте бокс *Graphical identification active*

Стан блокування відображається значком замка у верхньому лівому куті символу. Сюди можна назначить будь-яку картинку з бібліотеки файлів проекту. Також, заблокований елемент можна зробити невидимим чи сірим.

Для симулювання зміни значення створеної нами змінної 'Вологість' розмістіть в верхній частині екрану елемент універсальний повзунок, аналогічний до того, що ми на початку створили для керування змінної температури.

Запустіть проект і перевірте роботу блокування. Якщо Ви зробили все вірно, то при перетягуванні повзунка вище значення 75 кнопка 'Пуск' автоматично заблокується. При поверненні значення нижче 75 блокування повинне знятись.



Промодельюйте різні варіанти спрацювання цього блокування.

Докладніше про блокування можна прочитати в мануалі [Interlockings](#).

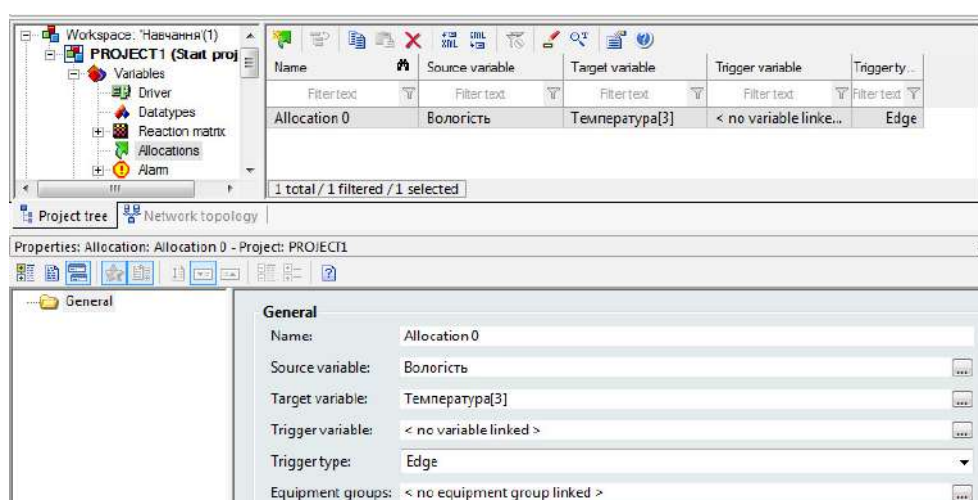
## Перерозподілення

Перерозподілення – це стандартний функціонал системи, який дозволяє передавати поточне значення однієї змінної до іншої, будь то драйверна змінна чи внутрішня.

При створенні перерозподілення визначають поняття вихідної змінної, яка являється джерелом значення, та кінцевої змінної, яка приймає відповідне значення.

Давайте створимо перерозподілення для змінної 'Вологість', яка буде передавати своє значення змінній 'Температура[3]'.

- ▶ В менеджері проекту активуйте пункт *Variables -> Allocations* і через праву кнопку мишки виберіть пункт *New allocation*
- ▶ У вікні властивостей створеного перерозподілення для поля *Source variable*: виберіть початкову змінну 'Вологість', і для поля *Target variable*: виберіть кінцеву змінну 'Температура[3]'



Більше жодних налаштувань робити не потрібно.

*При потребі, пере розподіленням можна керувати в онлайн режимі, використовуючи тригерну змінну. Коли ця змінна має значення '1' – перерозподілення активне, коли '0' – неактивне.*

Запустіть проект і перевірте роботу перерозподілення. Зверніть увагу на те, що змінна температури керується індивідуально, а при зміні значення вологості, воно автоматично передається до температурної змінної, тобто спрацює перерозподілення.

Докладніше про перерозподілення можна прочитати в мануалі [Allocations](#).

## Інтернаціоналізація

Під поняттям інтернаціоналізації ми розуміємо можливість системи керування в онлайн режимі представляти користувацький інтерфейс в зрозумілій для оператора формі, незалежно від країни в якій експлуатується проект. В нашому прикладі ми розглянемо питання зміни мови тексту проекту, а також конвертування одиниць вимірювання представлених змінних.

### ЗМІНА МОВИ

Технологія перемикавання мови дозволяє змінювати мову будь-якого тексту в Runtime (підписи елементів, граничні тексти, ідентифікатори змінних), записи списків тривоги, подій, стандартні діалоги, змінювати при перемиканні використовуваний шрифт, медіа-файли.

Мовне перемикавання здійснюється на базі сформованих в проекті мовних таблиць. Для кожної мови створюється своя таблиця, яка містить відповідники до оригінальних текстів проекту.

Текст, який підлягає перекладу позначається на початку ключовим символом '@'.

*При перекладі фрази дозволяється використання декількох символів '@' для розділення тих слів, які ми хочемо перекладати, а які ні.*

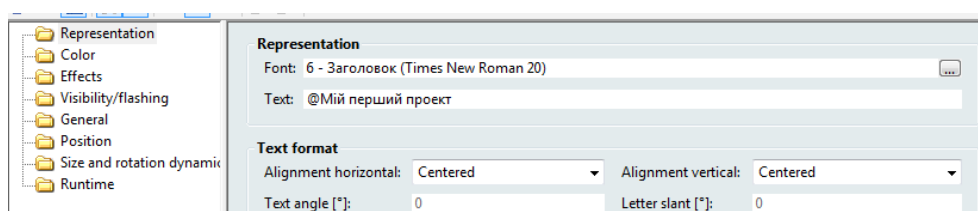
*Наприклад: @Користувач@ Олександр*

*Слово між першим і другим символом перекладеться, а останнє – ні. Сам ключовий символ на екрані не відображається.*

Давайте зробимо так, щоб оператор міг в онлайн режимі перемикає мову тексту 'Мій перший проект' на екрані процесу, а також підписи кнопок на панелі навігації. Переклад будемо здійснювати з української на англійську.

### Додавання ключового символу

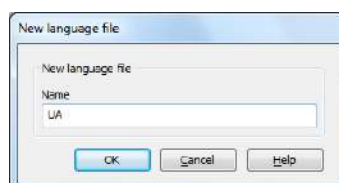
- ▶ Виділіть елемент статичного тексту на екрані 'Процес' і у вікні властивостей *Representation* -> *Text* перед текстом додайте символ '@'



- ▶ Аналогічним чином модифікуйте підписи на кнопках 'Процес', 'Хронологія', 'Тривоги', 'Архів', 'Тренди' навігаційної панелі

### Створення мовних таблиць

- ▶ В дереві проекту активуйте пункт *Language file* і через праву кнопку мишки виберіть пункт *New language file*
- ▶ У вікні що з'явилося задайте ім'я таблиці - 'UA'



- ▶ Повторіть цю операцію і створіть мовну таблицю 'EN'

На даному етапі у вікні детального перегляду повинно бути чотири колонки мовних таблиць: Keyword - слово чи вираз, який починається з символу '@', ZENONSTR.TXT – створюється та завантажується системою автоматично, мовна таблиця 'UA' та 'EN'.

Keyword	ZENONSTR.TXT	UA.TXT	EN.TXT
Filter text	Filter text	Filter text	Filter text

Поступово заповнимо рядки таблиць.

- ▶ В першій колонці запишіть ключове слово 'Мій перший проект'

Цей запис автоматично розповсюдиться на усі колонки

- ▶ В колонці 'EN' змініть напис на 'My first project'
- ▶ Аналогічним чином заповніть рядки для підписів навігаційних кнопок

Після заповнення мовна таблиця повинна мати наступний вигляд:

Workspace: "Навчання(1)"	Keyword	ZENONSTR.TXT	UA.TXT	EN.TXT
PROJECT1 (Start proj...)	Filter text	Filter text	Filter text	Filter text
Variables	Мій перший ...	Мій перший п...	Мій перший п...	My first project
Screens	Процес	Проце	Проце	Process
Functions	Хронологія	Хронологія	Хронологія	Chronology
Historian	Тривоги	Тривоги	Тривоги	Alarms
Recipes	Архів	Архів	Архів	Archive
Time control	Тренди	Тренди	Тренди	Trends
Programming interfaces				
zenon Logic (IEC 6113...				
Scheduler				
Interlockings				

*Для швидкого перекладу слів мовну таблицю можна експортувати в текстовому форматі та обробити зовнішньою програмою – мовним редактором.*

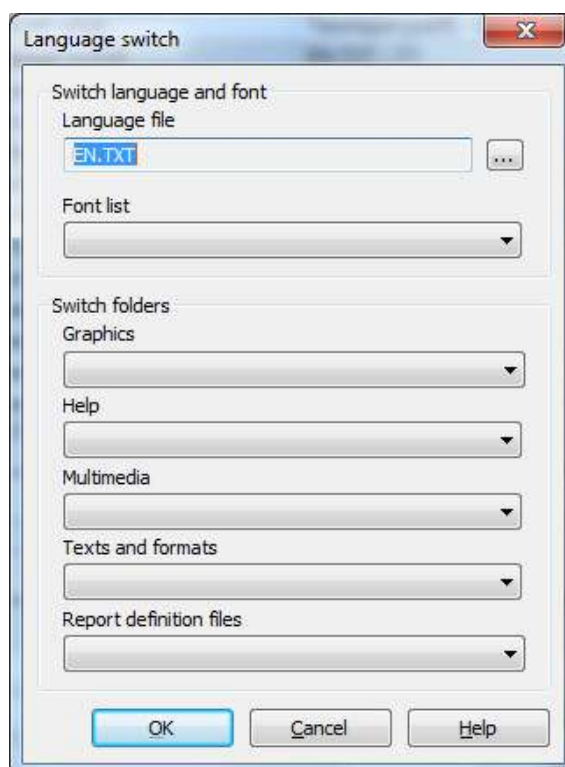
- ▶ Після закінчення заповнення таблиці натисніть зверху на піктограму збереження мовних файлів.

*Для кожної мовної таблиці в Runtime-файлах проекту система створює свій мовний файл (ZENONSTR.TXT, UA.TXT та EN.TXT). Також, в дереві проекту, розділ Files, система створила мовні папки для наповнення їх відповідним контентом.*

### Керування перемиканням

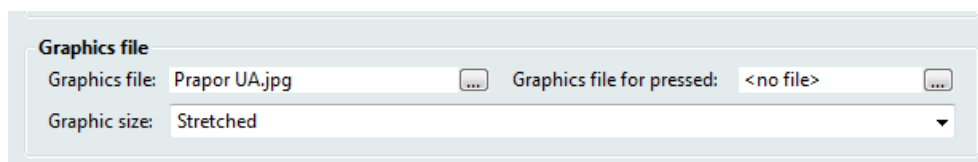
Для ініціації зміни мови нам необхідно створити відповідні функції та кнопки на панелі навігації.

- ▶ В менеджері проекту зайдіть в розділ *Functions* і створіть нову функцію *Application -> Language switch*
- ▶ В діалоговому вікні для поля *Language file* виберіть мовний файл 'UA'
- ▶ Закрийте діалог кнопкою *OK* і перейменуйте цю функцію в 'Зміна мови UA'
- ▶ Аналогічним чином створіть функцію 'Зміна мови EN'





- ▶ На екрані навігації справа створіть дві кнопки перемикавання на українську і англійську мову, та прив'яжіть до них створені функції зміни мовної таблиці
- ▶ В розділі *Representations* -> *Graphics file* вікна властивостей кнопки, прив'яжіть картинку відображення прапора держави. Попередньо завантажте відповідні файли в розділ *File* -> *Graphics* з інтернету



Запустіть проект і виконайте перемикавання. Проконтролюйте зміну мови.

Наша навігаційна панель тепер повинна мати наступний вигляд:

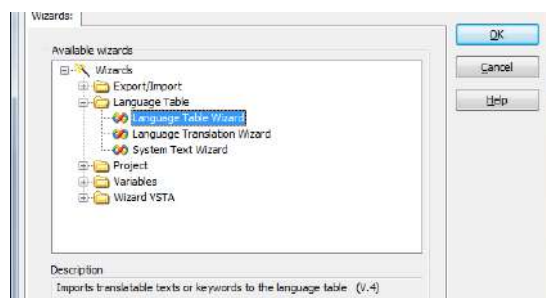


*Якщо потрібно, щоб відразу проект запускався на мові, відмінній від тієї на якій його зроблено, то необхідно функцію перемикавання вставити в скрипт автозапуску.*

У великих проектах дуже довго вручну складати мовні таблиці. Для полегшення цього процесу в систему керування інтегровано спеціального майстра-помічника, який в автоматичному режимі виконає задачу збору всіх текстів в проекті та додавання до них ключового символу.

- ▶ В середовищі розробки проекту зайдіть в головне меню *File* -> *Wizards...*

В папці *Language Table* ви знайдете помічника *Language Table Wizard*.



Спробуйте самостійно за його допомогою крок за кроком створити мовні таблиці проекту.

## ЗМІНА ОДИНИЦЬ ВИМІРЮВАННЯ

Існують випадки, коли для зручності представлення вимірюваних величин, їх необхідно відображати в різних одиницях. Наприклад, тиск можна представляти в барах чи паскалях, або температуру в цельсіях чи фаренгейтах.

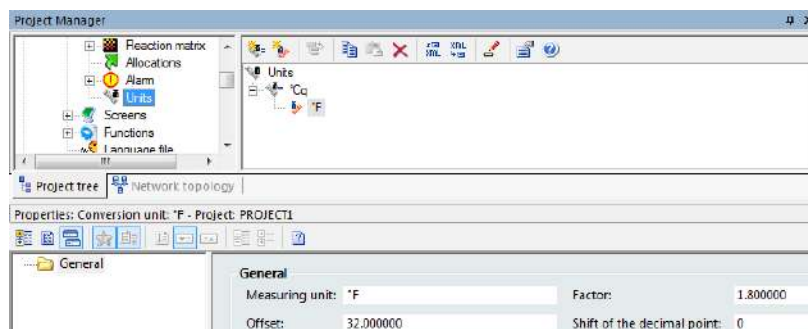
В нашому проекті є змінні температури, тому ми зробимо так, щоб оператор міг кнопками перемикаати одиниці вимірювання з °C на °F.

- ▶ В дереві проекту активуйте пункт *Variables -> Units* і через праву кнопку мишки виберіть *New base unit*
- ▶ Зайдіть у властивості створеної базової одиниці *General -> Measuring unit*: і перейменуйте на '°C'
- ▶ На панелі інструментів вікна детального перегляду виберіть піктограму *New conversion unit*
- ▶ Перейменуйте створену конвертаційну одиницю на '°F'

Тепер нам необхідно задати коефіцієнт перерахунку вимірюваної величини. Відоме наступне співвідношення конвертації одиниць:

$$1 \text{ градус Фаренгейта} = 1 \text{ градус Цельсія} \cdot 1.8 + 32$$

- ▶ Зайдіть у властивості конвертаційної одиниці *General -> Factor*: і задайте значення '1.8'
- ▶ Для *General -> Offset*: встановіть значення '32'

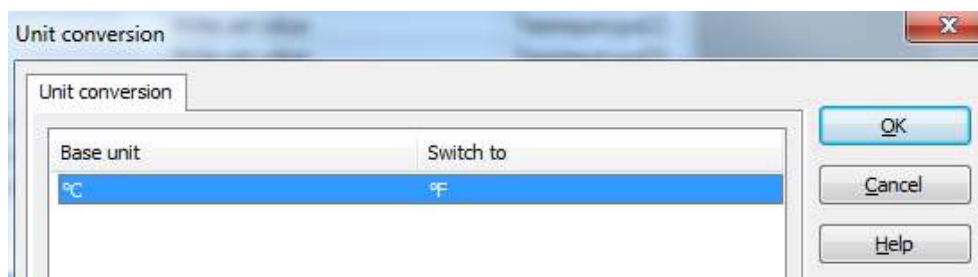


## Керування перемиканням

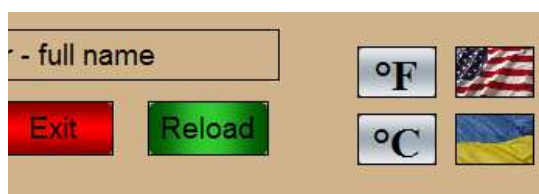
Для ініціації зміни одиниць вимірювання необхідно створити відповідні функції та кнопки на панелі навігації.

- ▶ В менеджері проекту зайдіть в розділ *Functions* і створіть нову функцію *Variable -> Unit conversion*
- ▶ В діалоговому вікні встановіть перемикаання °C на °F і закрийте його кнопкою *OK*

- Назвіть створену функцію 'Зміна одиниць вимірювання - F'



- Аналогічним чином створіть функцію повернення в базову одиницю і назвіть 'Зміна одиниць вимірювання - C'
- Створіть на екрані навігації дві системні кнопки '°C' та '°F' і прив'яжіть до них відповідні функції



Щоб конвертація запрацювала, необхідно вказати до яких змінних вона стосується.

- Зайдіть у властивості змінної 'Температура[1]' і в пункті *General -> Measuring unit*: зітріть ту одиницю вимірювання що там стоїть і зі спадного меню виберіть ту, яку ми тільки що створили - '°C'
- Повторіть цю операцію для інших температурних змінних

Запустіть проект і перевірте як працює система конвертації одиниць вимірювання.

Докладніше про зміну мови можна прочитати в мануалі [Language switch](#).

Докладніше про вбудованих в систему майстрів-помічників (не тільки про мовний) можна прочитати в мануалі [Wizards](#).

Докладніше про зміну одиниць вимірювання можна прочитати в мануалі [Measuring unit conversion](#).

## Математичний, системний та симуляційний драйвер

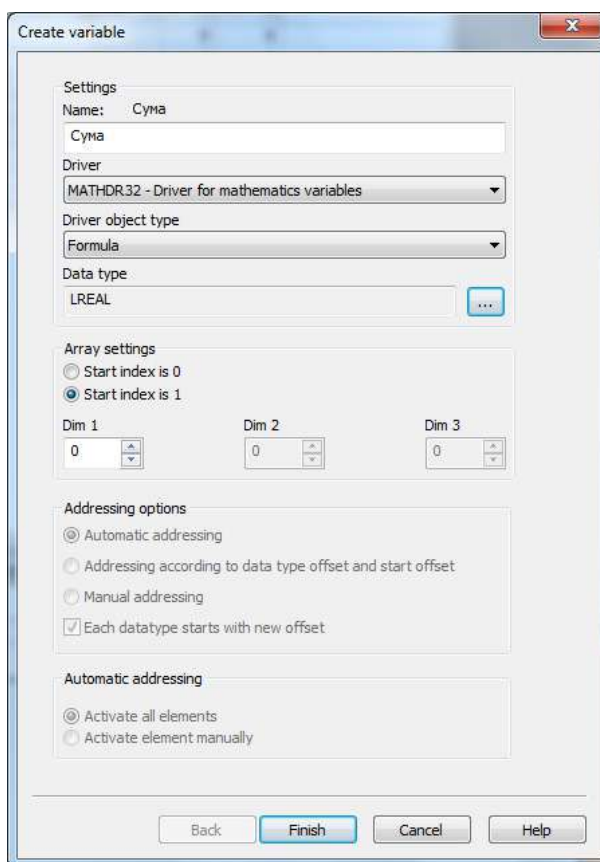
В цьому розділі ми розглянемо деякі корисні інструменти, які є в системі, що можуть знадобитись інженеру при розробці проекту.

### МАТЕМАТИЧНИЙ ДРАЙВЕР

zenon дозволяє виконати будь-яку математичну обробку виміряного значення. Наприклад, провести арифметичні дії з даними, обрахувати середнє чи мінімальне значення за певний проміжок часу, напрацювання машино-годин, виконати порівняння чи логічну операцію. Для цього існує математичний драйвер, який вбудовано в систему.

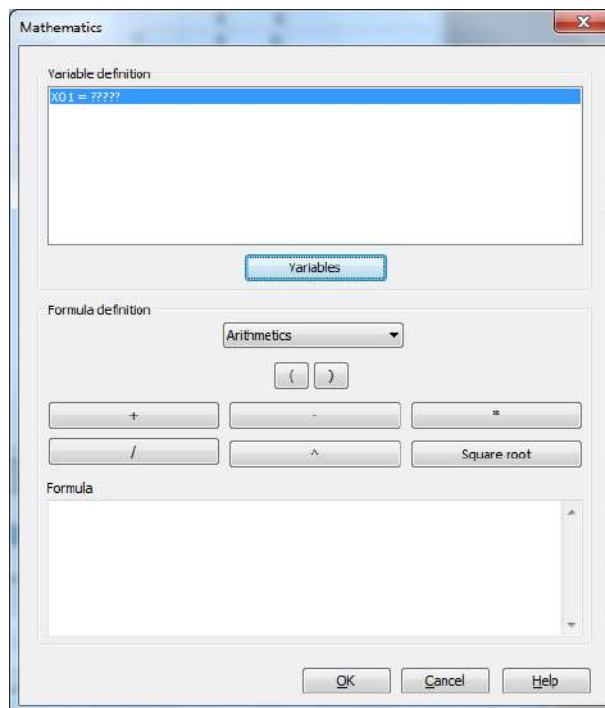
За замовчуванням цей драйвер вже створено в проекті системою. Нам залишається лише використати його. Для нашого проекту ми створимо змінну, яка буде представляти собою сумарне значення усіх температур.

- Створіть нову змінну 'Сума' і в якості драйвера зі спадного меню виберіть математичний драйвер - MATHDR32 та об'єктний тип - Formula

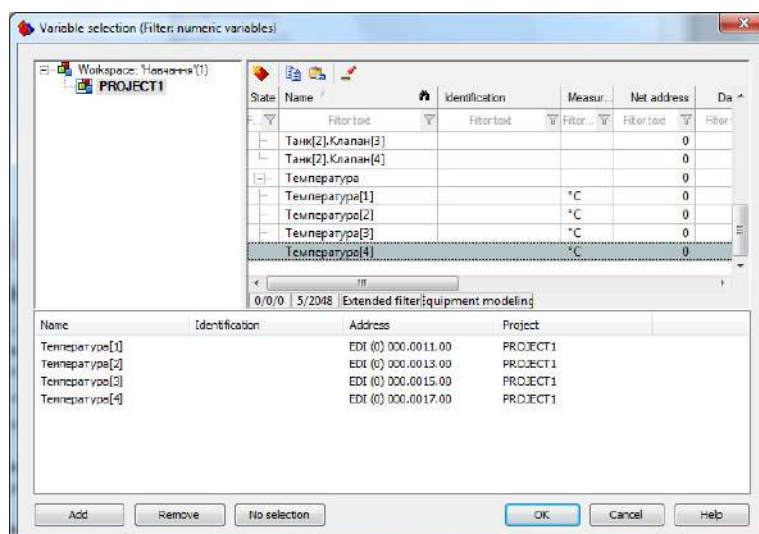


- Зайдіть у властивості створеної змінної *Value calculation* -> *Formula*

Автоматично з'явиться діалогове вікно запису формули:

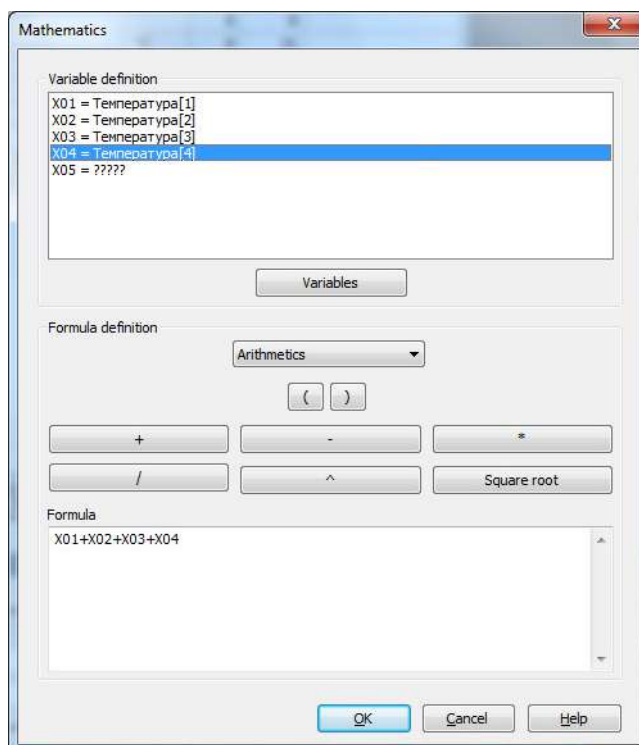


- Натисніть на кнопку *Variables* та додайте усі температурні змінні

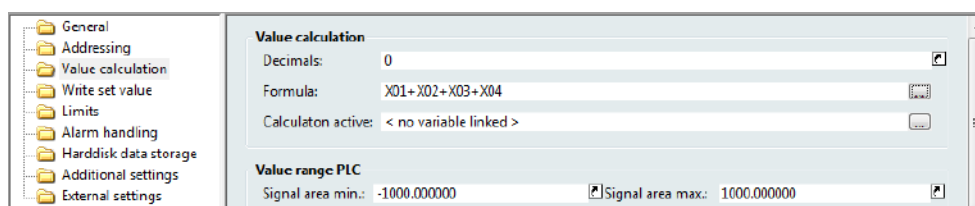


- Натисніть на кнопку *OK* і перейдіть безпосередньо до написання формули

Додаючи покроково змінні одна до одної, повинна сформуватись наступна формула:



Після натиснення кнопки **OK** у вікні властивостей змінної 'Сума' з'явиться запис даної формули:



- На екрані 'Процес' в верхній його частині створіть елемент відображення цифрового значення і прив'яжіть до нього створену нами математичну змінну

Запустіть проект і просимулюйте зміну значень температури. При цьому зверніть увагу на те, як змінюється значення математичної змінної. Простим додаванням значень переконайтесь, що Ви все зробили вірно.

## СИСТЕМНИЙ ДРАЙВЕР

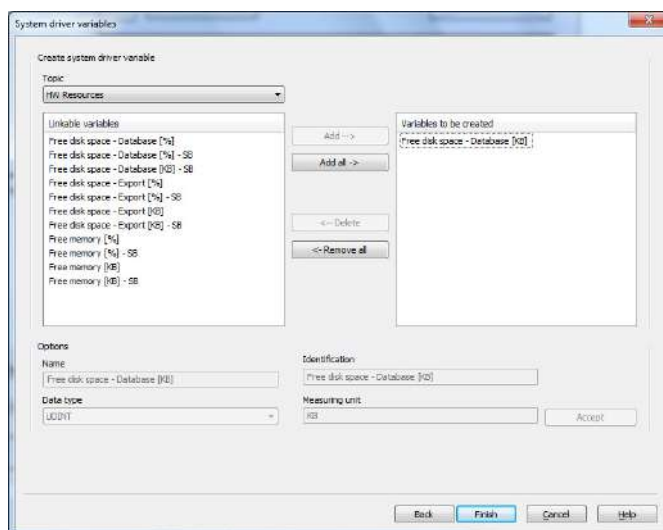
Для відображення в проєкті будь-якої системної інформації в zenon інтегровано спеціальний системний драйвер. За його допомогою можна наприклад, відобразити інформацію стосовно вільного місця на жорсткому диску ПК, шлях до проєктних директорій, мережевий стан клієнтських станцій, поточного авторизованого користувача та багато інших корисних даних.

При розгляді теми авторизації ми вже мали досвід по створенню системної змінної. Тепер давайте створимо змінну, яка буде відображати об'єм вільного місця на жорсткому диску.

- ▶ Створіть нову змінну і в якості драйвера зі спадного меню виберіть системний драйвер – SYSDRV

На наступному кроці з'явиться діалогове вікно з повним набором доступних користувачу системних змінних

- ▶ Зі спадного меню *Topic* виберіть розділ *HW Resources* і кнопкою *Add* додайте змінну *Free disk space – Database [KB]*



- ▶ Натиснувши кнопку *Finish* закінчіть створення змінної

Нам залишається лише розмістити елемент цифрового показника в верхньому правому куті екрану процесу і прив'язати до нього створену змінну.

Фрагмент екрану проєкту тепер має такий вигляд:



## СИМУЛЯЦІЙНИЙ ДРАЙВЕР

Реалізація симуляції зміни значень змінних в zenon може відбуватись трьома способами: шляхом переведення драйверу в стан Simulation-counting, шляхом написання симуляції на мовах IEC в zenon Logic та за допомогою стимуляційного драйверу.

Симуляційна змінна залежно від обраного режиму може збільшувати чи зменшувати своє значення з чітко заданим кроком. Крок, режим, та межі – це також симуляційні змінні.

Для реалізації симуляції створюють змінні з наступними параметрами:

Режим	Driver object type: PLC marker; Data type: UINT; Offset: 0 Signal resolution: 0..3; measuring range: 0..3  0 - симуляцію зупинено 1 - покрокове збільшення значення 2 - покрокове зменшення значення 3 - циклічне збільшення та зменшення значення  значення за замовчуванням: 1
Верхній ліміт	Driver object type: PLC marker; Data type: UINT; Offset: 1  значення за замовчуванням: 1000
Нижній ліміт	Driver object type: PLC marker; Data type: UINT; Offset: 2  значення за замовчуванням: 0
Крок симуляції	Driver object type: PLC marker; Data type: UINT; Offset: 3  значення за замовчуванням: 10
Змінна симуляції	Driver object type: PLC marker; Data type: UINT; Offset: 4 - 1023

Використовуючи зазначені вище дані, ми створимо змінну симуляції та змінну установки кроку симуляції.

- ▶ В менеджері проекту зайдіть в розділ *Variables* -> *Driver* і з папки *zenon system driver* виберіть драйвер *Driver for simulator variables (SIMUL32)*
- ▶ На базі цього драйверу створіть нову змінну 'Крок симуляції' і задайте усі параметри, як показано в таблиці вище
- ▶ У властивостях *Value calculation* встановіть діапазон значень для цієї змінної 0 – 100
- ▶ Аналогічним чином створіть змінну 'Змінна симуляції'



Таблиця змінних проекту тепер має наступний вигляд:

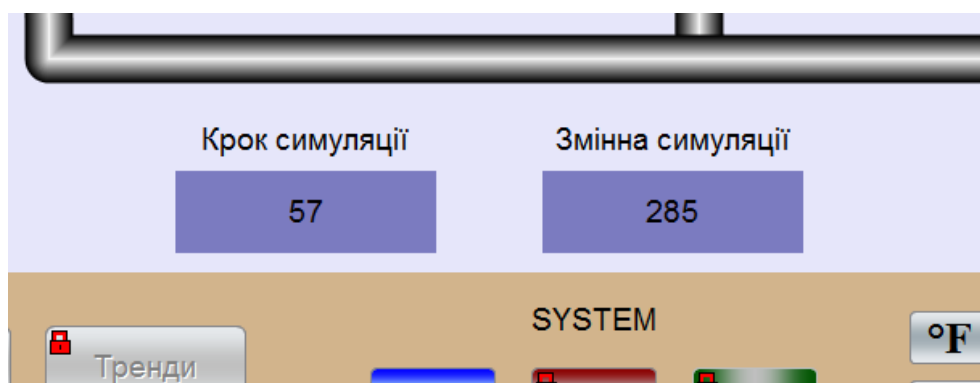
State	Name	Identification	Measur...	Net a...	Offset	Driver	Data type
	Filter text	Filter text	Filter...	Filter...	Filter...	Filter text	Filter text
	Температура			0	0	S7TCP32 - VIPA 300	INT
	Температура[1]		°C	0	11	S7TCP32 - VIPA 300	INT
	Температура[2]		°C	0	13	S7TCP32 - VIPA 300	INT
	Температура[3]		°C	0	15	S7TCP32 - VIPA 300	INT
	Температура[4]		°C	0	17	S7TCP32 - VIPA 300	INT
	Танк			0	0	Intern - Driver for internal variables	Танк
	Сума			0	0	MATHDR32 - Driver for mathematics vari...	LREAL
	Крок симуляції			0	3	SIMUL32 - Driver for simulator variables	UINT
	Змінна симуляції			0	4	SIMUL32 - Driver for simulator variables	UINT
	Вологість			0	0	S7TCP32 - VIPA 300	INT
	User - full name	User - fu...		0	0	SYSDRV - Driver for system variables	STRING
	Free disk space - Databas...	Free disk...	KB	0	0	SYSDRV - Driver for system variables	UDINT

- В нижній частині екрану процесу розмістіть два елементи цифрових показників і прив'яжіть до них створені нами симуляційні змінні
- Проконтролюйте, щоб для елемента зі змінною 'Крок симуляції' у властивостях *Write set value* було активовано бокс *Setting values active*

Таким чином, оператор зможе в онлайн режимі змінювати крок симуляції.

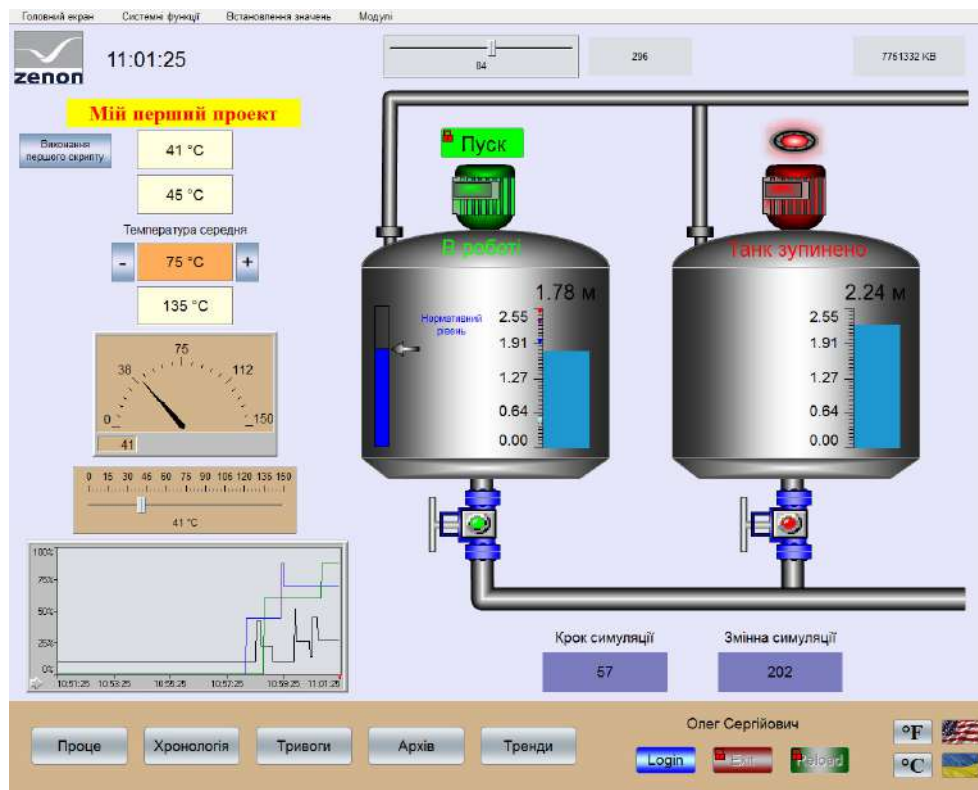
- Біля елементів цифрових показників розмістіть відповідні підписи для орієнтації призначення цих змінних.

Запустіть проект і перевірте зроблену роботу. При запуску змінна кроку симуляції буде одразу мати значення '10', оскільки це значення за замовчуванням. Змініть його в допустимому діапазоні і проконтролюйте як змінюється змінна симуляції.



Самостійно створіть та перевірте роботу симуляційних змінних встановлення режиму симуляції та діапазону.

Після виконання усіх операцій, описаних в даних навчальних матеріалах, Ваш екран процесу повинен мати такий вигляд:



Докладніше про використання математичного драйверу можна прочитати в технічній документації [MATHDR32](#).

Більше інформації про симуляційний драйвер можна знайти в технічній документації [SIMUL32](#).

## Короткий опис модулів розширення функціональності системи

Як ми вже зазначали, zenon має модульну структуру, що дозволяє при потребі значно розширювати функціонал системи, порівняно з базовими можливостями. В демонстраційному режимі доступні усі модулі системи. Деякі з них входять в базову конфігурацію поставки, а деякі замовляються окремо і потребують ліцензії.

В цьому розділі ми коротко надамо інформацію стосовно наявних модулів та їх доступності при замовленні ліцензії.

### БАЗОВИЙ КОМПЛЕКТ ПОСТАВКИ

Середовище розробки та виконання проекту – Editor і Runtime	Multiproject administration
Кількість змінних – Tags 64 – UNLIMITED	Programming interface VBA/C#/VB.NET
Alarms (AML)	Recipes
Chronology tracing (CEL)	Scheduler
Historian SE та Extended Trend SE	Worldview screen
Graphics libraries, ActiveX, WPF	Networking
User administration – FDA 21 CFR PART 11 compliant	Redundancy
Video, HTML	System drivers
Process Control Engine (PCE)	Multidriver interface
Support CE projects	Native multitouch support
Multimonitor administration	zenon Logic IEC 61131-3 programming

### Alarms - Список тривог

Інформування оператора про вихід параметрів за критичні значення чи появу аварійних ситуацій

- розбиття на групи, класи та області
- довідкова інформація по кожній тривозі
- експорт для подальшої обробки (XML, CSV, SQL)

### **Chronology tracing - Хронологічний список подій**

Ведення журналу дій оператора та моніторинг стану системи

- час, ім'я користувача, робоча станція
- змінна, старе та нове значення
- фільтрація по заданим параметрам
- експорт для подальшої обробки (XML, CSV, SQL)

### **Historian SE - Стартова редакція модуля архівів**

Реєстрація значень технологічних параметрів і формування історичної бази даних

- циклічна реєстрація даних
- вільний старт і стоп архівування (також по події)
- збереження у внутрішньому форматі
- експорт даних в XML, TXT, DBF

### **Extended Trend SE - Стартова редакція модуля трендів**

Представлення вимірюної інформації у вигляді графіків

- відображення історичних та онлайн даних
- до 8 трендових кривих
- різних Y-вісі

### **Process Control Engine - Програмний інструмент керування процесом**

- багатозадачність
- 7 рівнів пріоритету
- контроль часу циклу
- мови програмування: VB- та Java-скрипт
- циклічне чи подійове виконання задач

### **Recipes - Рецепти**

Реалізація різноманітних алгоритмів\сценаріїв керування виробничим обладнанням

### **Scheduler – Планувальник**

Формування календарного графіку виконання тих чи інших операцій

### **zenon Logic - Софт логіка**

Інтегрована підсистема написання програмної логіки на стандартних мовах IEC 61131-3 та реалізації Soft-PLC.

- IL, ST, LD, FBD, SFC
- створення програмної симуляції

## **МОДУЛІ РОЗШИРЕННЯ ФУНКЦІОНАЛЬНОСТІ СИСТЕМИ**

Наведені нижче модулі потребують придбання окремої ліцензії.

### **Historian - Модуль архівів**

- усі функціональні можливості стартової редакції модуля архівів
- запис даних по події та по зміні значення
- каскадне архівування
- запис по пакетам/партіям, змінам
- автоматичне вилучення даних
- збір даних в режимі реального часу (RDA)
- редагування даних архіву
- експорт в SQL базу даних (в комплекті з Historian SQL Server)

### **Extended Trend - Модуль трендів**

- усі функціональні можливості стартової редакції модуля трендів
- необмежена кількість трендових кривих
- аналіз даних
- логарифмічне представлення, діаграма Ганта
- масштабованість
- Т-вісь та Х-вісь

### **Reporting - Модуль звітів**

Формування табличних та графічних звітів на базі онлайн та історичних даних

- понад 150 функцій обробки інформації
- MS Report Builder 2.0
- експорт в TXT, XML, PDF

- вільноконфігурований друк

### **Recipegroup Manager – Розширений модуль рецептів**

- усі функціональні можливості стандартного модуля рецептів
- табличне представлення
- моніторинг статусу виконання
- непрямий виклик рецепту
- об'єднання рецептів в групи

### **Message Control - Модуль повідомлень**

Інформування ключових персон про появу нештатних ситуацій чи вихід параметрів за допустимі межі

- вихідні повідомлення на пейджер, e-mail, мобільний телефон
- голосові повідомлення
- реєстрація стану передачі
- користувацьке адміністрування

### **Industrial Maintenance Manager - Менеджер техобслуговування**

Організація календарного обслуговування, ремонту та заміни обладнання

### **Industrial Performance Analyzer - Аналізатор продуктивності**

Дозволяє здійснювати аналіз статистики аварій та виявляти «слабкі» місця системи. Забезпечує зменшення часу простою обладнання і дозволяє оптимізувати його використання.

### **Automatic Line Coloring - Менеджер топології**

Автоматична зміна кольору комунікаційних ліній відповідно до сформованої топологічної структури (трубопроводи, ланцюги електроживлення).

### **PLC Diagnosis – Simatic S7-GRAPH**

Діагностування PLC за допомогою Simatic S7-GRAPH. Відображає в графічному вигляді поточний крок виконання програми.

### **Production and Facility Scheduler - Планувальник виробництва та ресурсів**

Ведення планового виробництва, контроль наявних ресурсів та готової продукції на складах

- адміністрування часу спрацювання подій
- денні, тижневі та місячні розклади
- визначення пріоритетності
- зв'язок з VBA макросами
- попередній перегляд точок спрацювання подій

### **SAP Interface**

Реалізація двонаправленого зв'язку з SAP/RB ERP додатками.

### **Project Simulation - Проектна симуляція**

Дозволяє в безударному режимі відімкнутися від реальних апаратних засобів і перейти в режим симуляції проекту

### **Energy Management System - Система керування енергоспоживанням**

Здійснює оптимізацію споживання енергетичних ресурсів на виробництві завдяки короткостроковому прогнозуванню, усуненню пікових навантажень, автоматичному відключенню споживачів і підключенню додаткових джерел енергії, прив'язці до гранично допустимого рівня.

### **Historian SQL Server**

Дозволяє у відкритій формі здійснювати передачу архівної інформації в будь-яку SQL базу даних без втрати зв'язку з модулями zenon (ETM, Reporting, і т.д.).

Включає Microsoft SQL Server 2008 R2 Standard License 1 CAL

### **Process Gateway – Комунікаційний шлюз**

Служить для зв'язку з надбудовними системами за допомогою численних інтерфейсів і протоколів для одно і двобічної передачі даних.

- OPC DA Server
- OPC UA Server
- Modbus Slave
- SQL Online
- DNP3 Slave

- SNMP Server
- DEC Slave
- IEC 60870 Slave

### **Batch Control - Керування виробництвом по-партіям**

Модуль керування Batch-процесами у відповідності до стандарту ISA 88

- інтегрування рецептурної концепції
- матричне та процедурне (PFC) виконання операцій
- фазове представлення стану обладнання

### **WEB Server (Pro), Web Server Pro Light**

Віддалений контроль та керування проектом через інтернет

- повний доступ до проекту
- система адміністрування користувачів
- підтримка будь-яких браузерів



## HMI/SCADA системи zenon

■ бульвар Івана Лепсе, 4, м. Київ, 03680, Україна  
тел. (44) 496-18-88, факс (44) 496-18-18  
office@sv-altera.com  
www.svaltera.ua



■ **Вінниця**  
21027, вул. Келецька, 53, офіс 503  
Тел. (0-432) 52-30-13  
Факс (0-432) 52-30-98  
svaltera@utel.net.ua

■ **Дніпропетровськ**  
49064, пр-т Калиніна, 62  
Тел. (0-56) 376-92-86  
Факс (0-56) 376-92-78  
svaltera@a-teleport.com

■ **Донецьк**  
83048, пр-т Визволення Донбасу, 8-Б  
(1-й поверх)  
Тел./факс (0-62) 385-35-96, 385-35-97,  
348-12-39, моб. 095 480-00-26  
office@svaltera.dn.ua

■ **Житомир**  
10029, вул. Чапаєва, 7, офіс 212  
(2-й поверх)  
Тел. (0-412) 48-03-76, 48-03-77  
zhitomir@svaltera.ua

■ **Запоріжжя**  
69006, пр-т Металургів, 12А  
Тел. (0-61) 224-34-80,  
701-11-49, 222-48-55  
Факс (0-61) 222-48-56  
svaltera\_zp@svaltera.ua

■ **Івано-Франківськ**  
76006, вул. В. Симоненка, 23, офіс 308  
Тел./факс (0-342) 72-21-22, 72-32-33  
i-f@svaltera.ua

■ **Кіровоград**  
25001, вул. Можайського, 43, офіс 5  
(3-й поверх)  
Тел./факс (0-522) 33-93-44, 27-31-43  
Моб. 068 461-89-80, 066 331-12-51  
kirovograd@svaltera.ua

■ **Кременчук**  
39610, пр-т 50 років Жовтня, 17/11  
Тел. (0-5366) 4-86-67  
Факс (0-5366) 4-13-79  
Моб. 063 367-09-33  
kremenchug@svaltera.ua  
svaltera\_kr@ukr.net

■ **Кривий Ріг**  
50065, вул. XXII партз'їду, 37, офіс 1  
Тел./факс (0-56) 409-32-89,  
(0-56) 409-78-10, 405-21-99  
svaltera\_kr@optima.com.ua

■ **Львів**  
79000, вул. Симона Петлюри, 27  
Тел./факс (0-32) 297-66-90  
svaltera@svaltera.lviv.ua

■ **Луганськ**  
91055, вул. Жовтнева, 82  
Тел./факс (0-642) 93-72-50, 93-72-95  
Моб. 095 479-89-85  
svaltera\_lg@svaltera.ua

■ **Миколаїв**  
54030, вул. В. Морська, 23, офіс 29  
Тел. (0-512) 58-08-12, 58-06-41  
Факс (0-512) 58-06-33  
svaltera\_nik@mksat.net

■ **Одеса**  
65091, вул. Колонтаївська, 27  
Тел./факс (0-482) 33-28-60, 33-28-61,  
(0-48) 732-12-77  
office@sv-altera.od.ua

■ **Рівне**  
33003, вул. Гагаріна, 39  
Тел. (0-362) 46-05-35, 46-05-37  
svaltera@rivne.com

■ **Суми**  
40004, вул. Ремісничка, 35-2 (2-й поверх)  
Тел. (0-542) 65-35-01, 65-35-10  
Моб. 095 578-16-64, 096 282-19-74  
svaltera\_sm@svaltera.ua  
svaltera@meta.ua

■ **Харків**  
61052, вул. Полтавський шлях, 56,  
6-й поверх, к. 606  
Для пошти: 61052, а/я 10567  
Тел. (0-57) 758-72-91, 758-62-12  
svaltera\_kh@svaltera.ua

■ **Черкаси**  
вул. Луценка, 7/3, 2-й поверх  
Тел./факс (0-472) 63-96-45  
Тел. (0-472) 63-55-23, 56-94-37  
cherkassy@svaltera.ua

□ **Кишинів (республіка Молдова)**  
ICS "ElectroTehnoImport" SRL  
str. Gradiņa Botanica 2/1  
Chisinau MD 2002  
Тел./Fax: (+37322) 844-688  
Tel: (+37322) 92-11-71, 92-12-72  
www.electroimport.md  
elimport@mcc.md