МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

М. Я. Островерхов

КОМП'ЮТЕРНІ ЗАСОБИ АВТОМАТИЗАЦІЇ ЕЛЕКТРОТЕХНОЛОГІЧНИХ УСТАНОВОК

КОМП'ЮТЕРНИЙ ПРАКТИКУМ

Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського як навчальний посібник для здобувачів ступеня бакалавра за освітньою програмою «Електротехнічні пристрої та електротехнологічні комплекси» спеціальності 141 «Електроенергетика, електротехніка та електромеханіка»

Київ
КПІ ім. Ігоря Сікорського
2023

Комп’ютерні засоби автоматизації електротехнологічних установок: Комп’ютерний практикум [Електронний ресурс]: навч. посіб. для здобув. ступеня бакалавра спеціальності 141 «Електроенергетика, електротехніка та електромеханіка» / М. Я. Островерхов; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 3,9 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2023. – 145 с.

Гриф надано Методичною радою КПІ ім. Ігоря Сікорського (протокол № 3 від 07.12.2023 р.) за поданням Вченої ради факультету електроенерготехніки та автоматики (протокол № 3 від 30.10.2023 р.)

Електронне мережне навчальне видання

КОМП’ЮТЕРНІ ЗАСОБИ АВТОМАТИЗАЦІЇ ЕЛЕКТРОТЕХНОЛОГІЧНИХ УСТАНОВОК
КОМП’ЮТЕРНИЙ ПРАКТИКУМ

Укладач *Островерхов Микола Якович*, д-р техн. наук, професор

Відповідальний редактор *Перетятко Ю.В.,* канд. техн. наук, доцент

Рецензенти: *Будько В.І.*, д-р. техн. наук, завідувач кафедри відновлюваних джерел енергії факультету електроенерготехніки та автоматики

Чумак В.В., канд. техн. наук, завідувач кафедри електромеханіки факультету електроенерготехніки та автоматики

У навчальному посібнику представлено комп’ютерний практикум із створення, налагодження та запуску проектів автоматизованих систем керування електротехнологічними установками та комплексами в SCADA. Виконання робіт здійснюється в сучасному програмному комплексі SCADA, який дозволяє створювати системи автоматизації від простих автономно працюючих контролерів до територіально розподілених систем з використанням промислових та інтранет/інтернет мереж. Роботи містять мету, програму, методичні пояснення, завдання та контрольні запитання.

Призначено для здобувачів ступеня бакалавра факультету електроенерготехніки та автоматики, які навчаються за освітньою програмою «Електротехнічні пристрої та електротехнологічні комплекси».

©М. Я. Островерхов
© КПІ ім. Ігоря Сікорського, 2023

ЗМІСТ

Вступ 4
Комп'ютерний практикум № 1. Створення проекту системи автоматизації 5
Комп'ютерний практикум № 2. Запуск та налагодження проекту системи автоматизації з функцією керування 20
Комп'ютерний практикум № 3. Програмування логічних функцій 38
Комп'ютерний практикум № 4. Створення проекту одноконтурної системи автоматичного регулювання 57
Комп'ютерний практикум № 5. Запуск та налагодження проекту одноконтурної системи автоматичного регулювання 67
Комп'ютерний практикум № 6. Створення статичних та динамічних зображень 80
Комп'ютерний практикум № 7. Програмування на мовах ST та FBD 93
Комп'ютерний практикум № 8. Програмування на мовах IL та SFC 117
Комп'ютерний практикум № 9. Створення звіту тривог та СПАД-архіву 134
Список рекомендованої літератури 144
Додаток А. Приклад оформлення титульного листа 145

ВСТУП

Комп'ютерний практикум з дисципліни «Комп'ютерні засоби автоматизації електротехнологічних установок» призначені для закріплення теоретичних знань й набуття практичних навичок при розробці систем автоматизації в SCADA-системі. Методичні вказівки до кожного комп'ютерного практикуму включають назву, мету та програму роботи, завдання на роботу та контрольні запитання.

Протокол роботи та відповідний проект автоматизованої системи управління готується кожним студентом індивідуально до початку аудиторного заняття згідно з варіантом завдання. За допомогою довідкової системи SCADA самостійно вивчаються властивості графічних елементів та особливості проектування. На графічному екрані кожного проекту обов'язково розташовується інформація про розробника, зокрема ПІБ та номер групи студента.

Під час аудиторного заняття здійснюється допуск до роботи, остаточне налагодження проекту, отримання й зарахування результатів. На наступному занятті здійснюється захист звіту комп'ютерного практикуму.

Протокол комп'ютерного практикуму оформлюється на аркушах паперу формату А4, які скріплюються ліворуч. Текст та інші матеріали звіту розміщується з однієї сторони аркушів паперу, шрифт Times New Roman розміром 14. Звіт складається з титульного листа, мети роботи, номера й змісту завдання, результатів роботи (графічні екрани, графіки тощо), висновку. Приклад оформлення титульного листа наведено в додатку А.

Комп’ютерний практикум № 1

СТВОРЕННЯ ПРОЕКТУ СИСТЕМИ АВТОМАТИЗАЦІЇ

Мета роботи – засвоєння послідовності створення проекту системи автоматизації в SCADA-системі з використанням механізму автопобудови каналів.

1 ПРОГРАМА РОБОТИ

1.1 Створити вузол автоматизованого робочого місця (АРМ).

Для розробки проекту системи автоматизації запустити в роботу **Інтегроване середовище розробки** (ICP) шляхом виконання команди в групі установки в меню операційної системи Windows **(Пуск\Всі програми)** або подвійного натискання лівої клавіші комп’ютерної мишки (ЛК) на ярлику робочого столу Windows. В результаті відкривається вікно ICP, яке зображено на рис. 1.

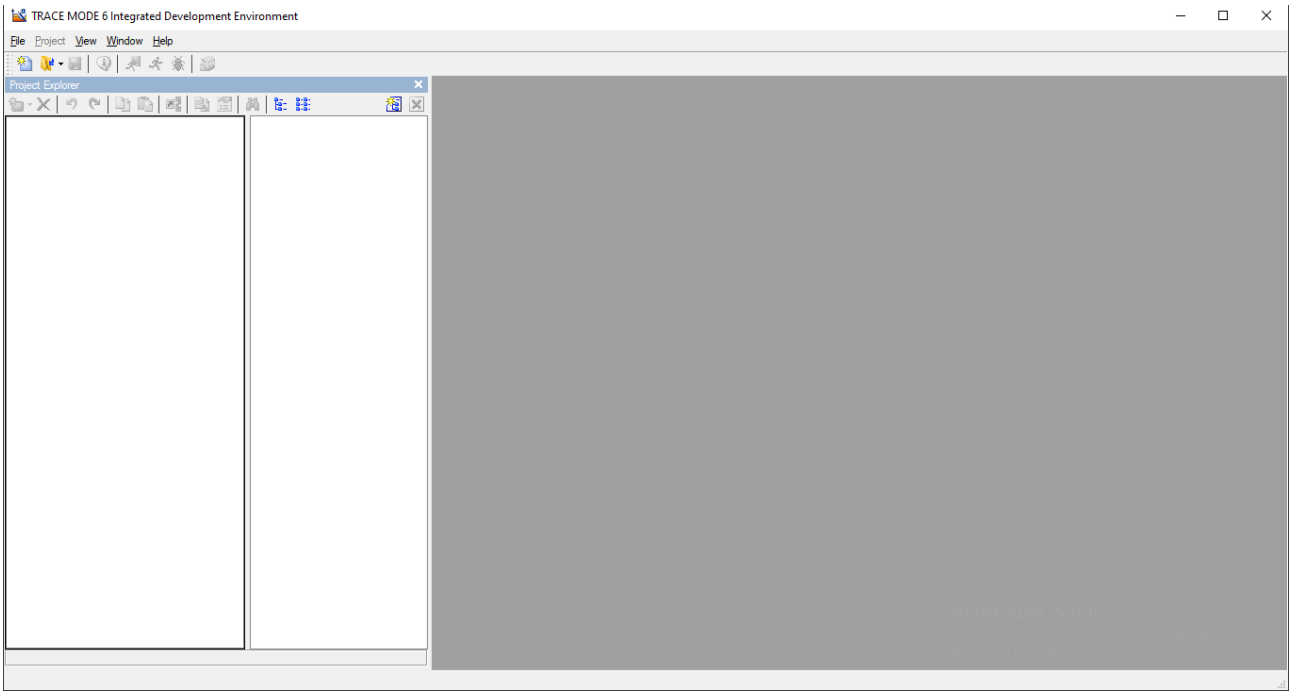


Рисунок 1 – Вікно інтегрованого середовища розробки

В меню **File** (Файл) вибрати пункт **Preferences...** (Налаштування...), а після відкриття вікна – команду **Complexity Level** (Рівень складності) та здійснити налаштування відповідно до рис. 2.

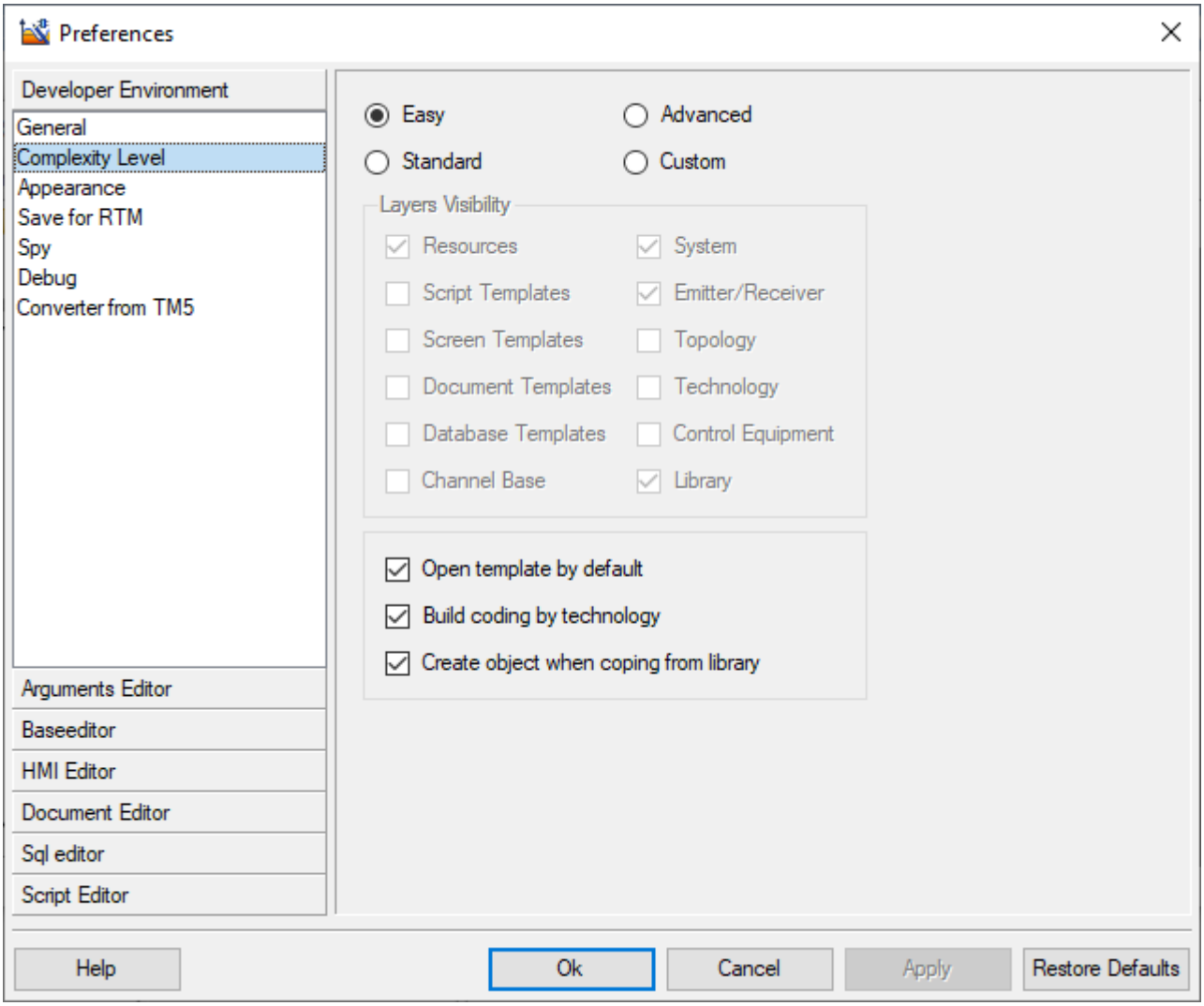


Рисунок 2 – Налаштування в ICP рівня складності

Потім у цьому ж вікні вибрати команду **Debug** (Налагодження) та провести налаштування згідно з рис. 3. Після настроювань натискається кнопка **Ok** (Готово).

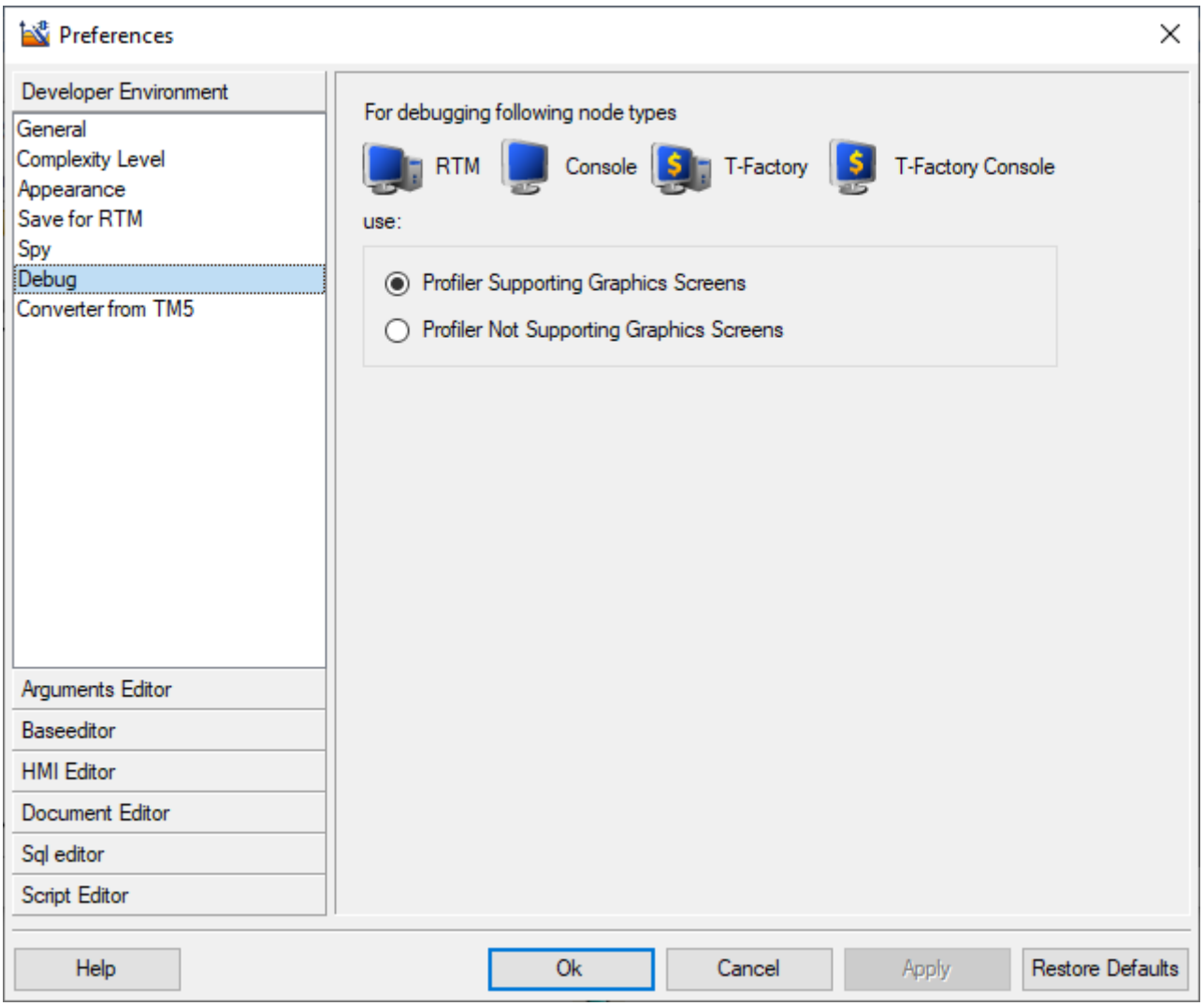


Рисунок 3 – Налаштування в ICP налагодження вузла

За допомогою кнопки на інструментальній панелі ICP (рис. 1) або через меню **File\New...** (Файл\Новий...) створити новий проект простого стилю шляхом вибору у вікні, показаному на рис. 4, стилю розробки **Easy** (Простий) та натискання ЛК кнопки **Create** (Створити).

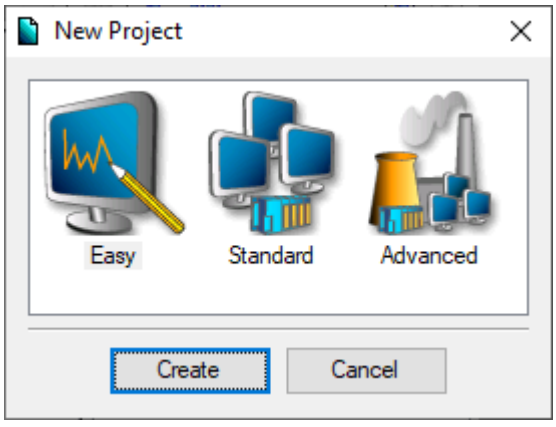


Рисунок 4 – Створення нового проекту

У лівому вікні навігатора проекту **Project Explorer** (рис. 1) з'явиться дерево проекту зі створеним вузлом АРМ **RTM_1**, як показано на рис. 5. Після подвійного натискання ЛК на вузлі **RTM_1** відобразиться у правому вікні навігатора проекту вміст вузла, а саме, порожня група **Channels** (Канали) та один канал класу **Screen_#1:1** (Екран_#1:1), призначений для відображення на вузлі АРМ графічного екрана.

Зберегти проект шляхом вибору в меню **File** (Файл) пункту **Save** (Зберегти) або **Save As...** (Зберегти як...) або натисканням кнопки **Save current project** (Зберегти поточний проект) на інструментальній панелі (рис. 1), вибравши потрібну папку для зберігання файлів проекту та задавши з клавіатури бажане ім’я, наприклад **Robota_1**. В результаті створюється основний файл проекту **Robota_1.prj** з розширенням **.prj**, а після компіляція та папка з таким же іменем **Robota_1** для зберігання інших файлів проекту.

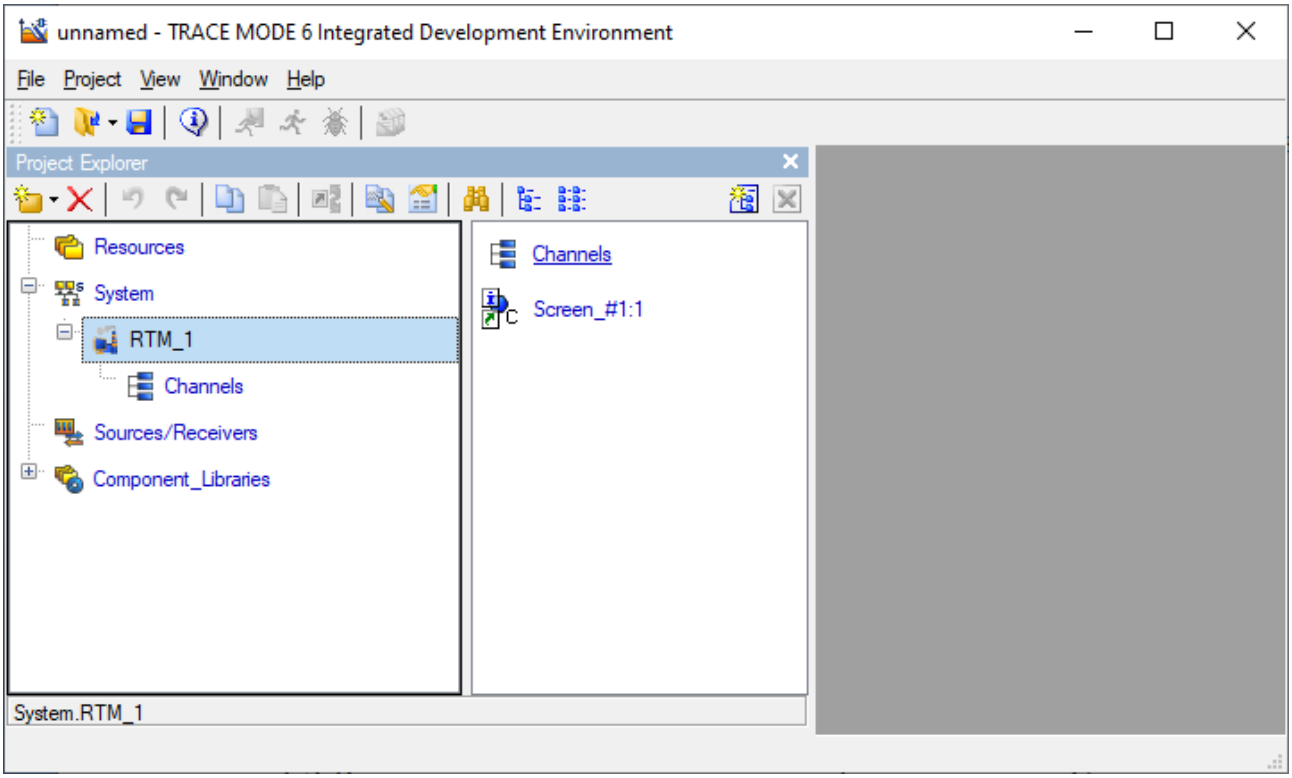


Рисунок 5 – Вікно навігатора проекту

1.2 Розмістити статичний текст на графічному екрані.

Створимо у лівому верхньому куті екрана напис (статичний текст) **Значення параметра**. Для цього подвійним натисканням ЛК на компоненті

Screen_#1:1 (Екран_#1:1) відкривається вікно графічного редактора. На панелі інструментів графічного редактора за допомогою ЛК натискається кнопка **Text** (Текст), а потім в лівий верхній кут екрана вставляється графічний елемент (ГЕ) у вигляді прямокутника шляхом фіксації ЛК точки прив'язки з наступним рухом курсору на потрібну відстань та фіксацією ЛК остаточного розміру ГЕ (рис. 6).

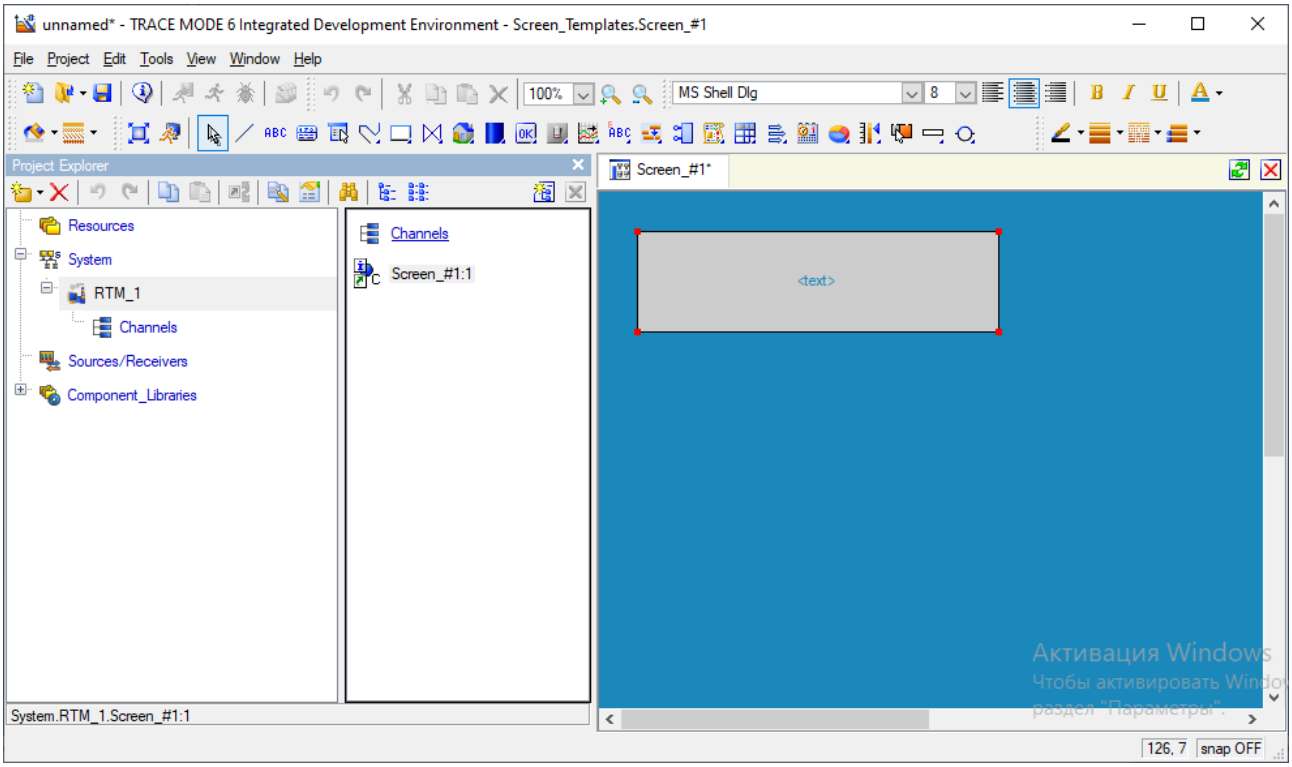


Рисунок 6 – Розміщення ГЕ статичного тексту на екрані

Для переходу в режим редагування отриманого ГЕ на панелі інструментів редактора натискається кнопка **Edit Mode** (Режим редагування), а потім подвійним натисканням ЛК на розміщеному ГЕ відкривається вікно його властивостей (рис. 7). Праворуч від слова **Text** (Текст) цього вікна набирається за допомогою клавіатури напис **Значення параметра**. Після натискання ЛК в межах вікна або натискання на клавіатурі клавіші **Enter** цей напис з'являється на ГЕ (рис. 8). У вікні властивостей ГЕ (рис. 7) обов'язково задати бажаний вид та колір контуру, колір залиття поверхні, колір тексту, тип вирівнювання та шрифт тексту, тощо. Детальна інформація про ГЕ отримується за допомогою

кнопки **Help** (Довідка) у вікні властивостей ГЕ. Відкриті вікна в ІСР закриваються кнопкою у верхньому правому куті.

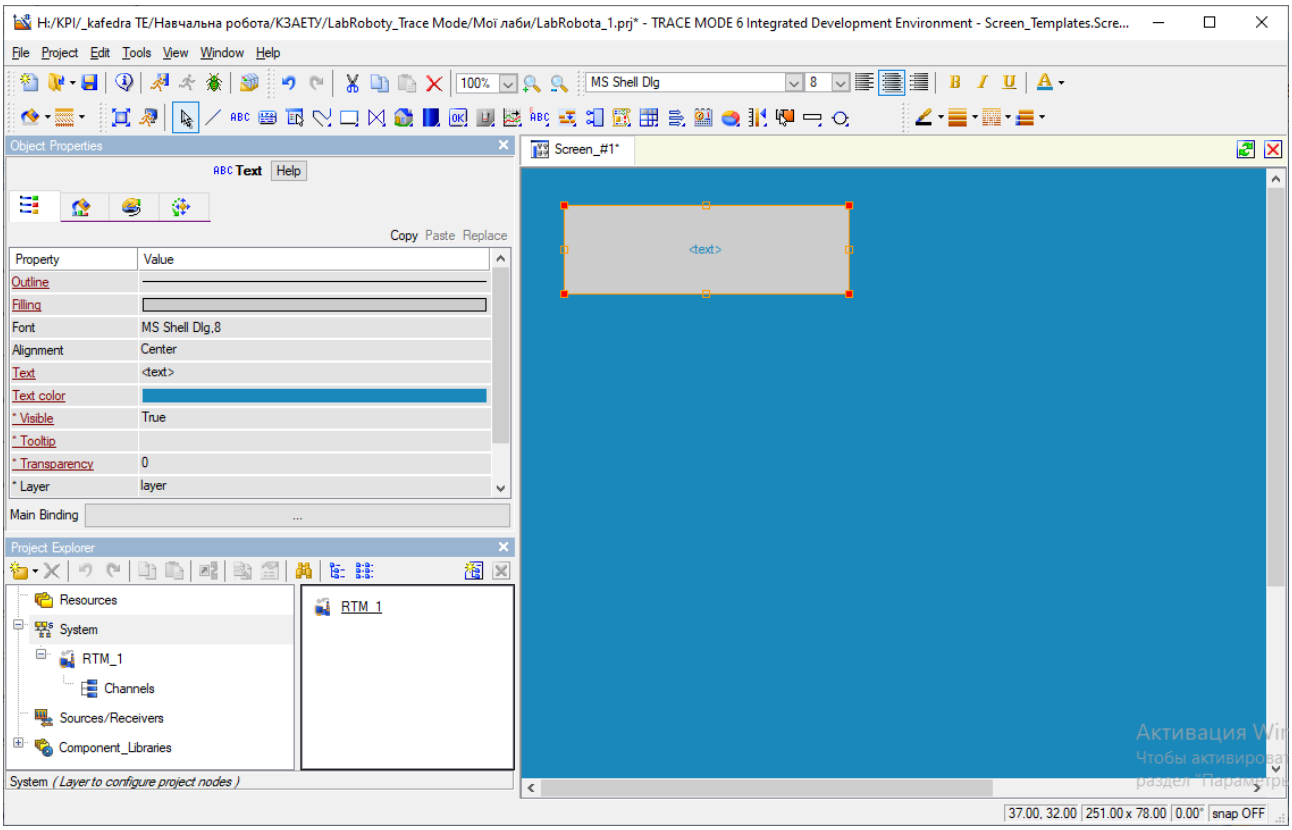


Рисунок 7 – Вікно властивостей ГЕ статичного тексту

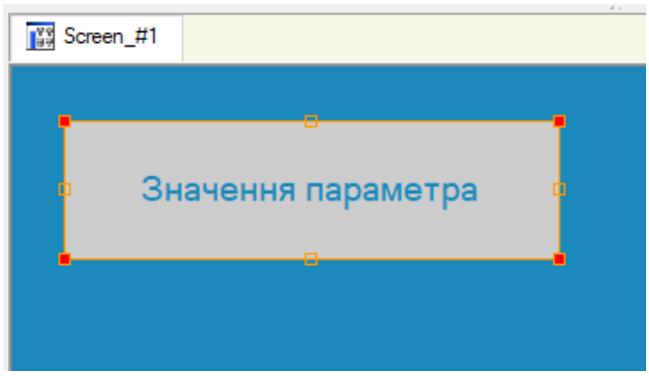


Рисунок 8 – Вигляд ГЕ статичного тексту

1.3 Створити динамічний текст та аргумент екрана для прив’язки до динамічного тексту.

Підготуємо ГЕ для відображення числового значення змінної (динамічного тексту) від джерела сигналу та створимо аргумент екрана для прив’язки змінної до цього джерела. Праворуч від існуючого напису «**Значення параметра**» створюється новий ГЕ статичного тексту

аналогічно до вищевикладеного у пункті 1.2. У вікні його властивостей подвійним натисканням ЛК на рядку **Text** (Текст) викликається меню **Indication type** (Вид індикації) (рис. 9). У правому полі відкритого рядка натискається ЛК та із списку доступних типів вибирається **Value** (Значення) (рис. 10).

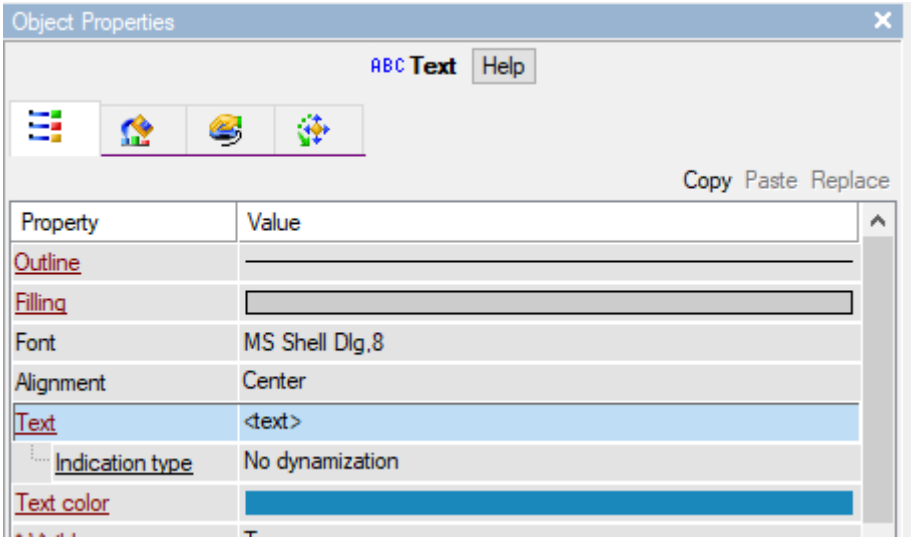


Рисунок 9 – Налаштування динамізації ГЕ

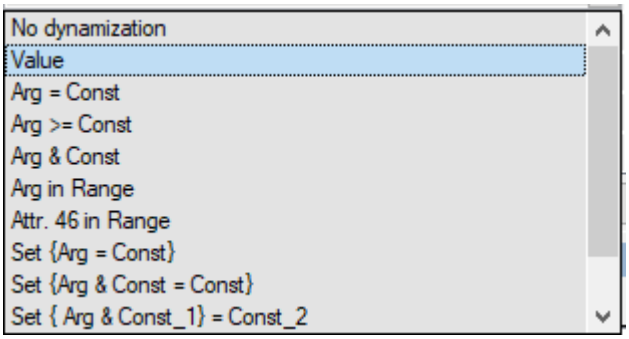


Рисунок 10 – Вибір типу динамізації

Після натискання ЛК у правому полі **Binding** (Прив'язка) (рис. 11) з'являється вікно **Binding setup** (Властивості прив'язки) (рис. 12).

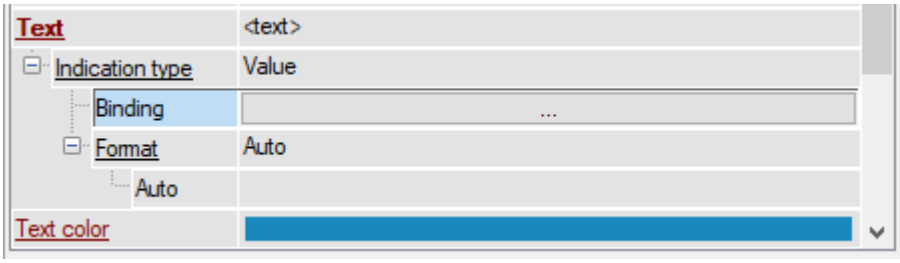


Рисунок 11 – Настроювання параметрів динамізації

Створюється аргумент екрана шляхом натискання кнопки **Create Argument** (Створити аргумент) на інструментальній панелі вікна. Аргументу у стовпці **Name** (Ім’я) автоматично надається ім’я **ARG_000.** Після подвійного натискання ЛК на імені **ARG_000** із клавіатури вводиться нове ім’я **Параметр** замість **ARG_000**, ввід якого завершується клавішею **Enter**. Зв'язок з цим аргументом підтверджується натисканням у вікні кнопки **Ok** (Готово). В результаті графічний екран буде мати вигляд, зображений на рис. 13.

У вікні властивостей ГЕ обов’язково задати бажаний вид та колір контуру, колір залиття поверхні, тощо.

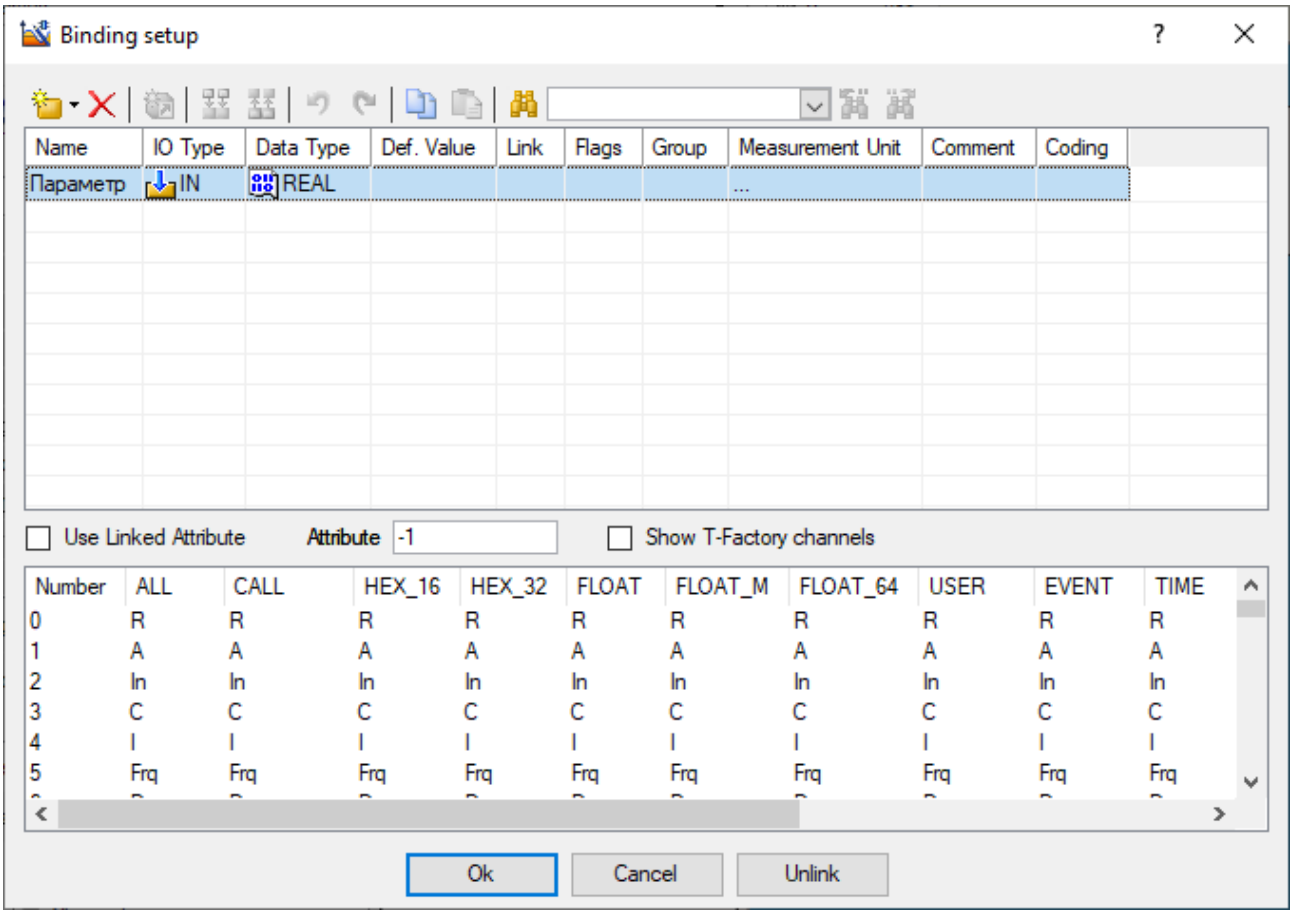


Рисунок 12 – Створення аргументу екрана

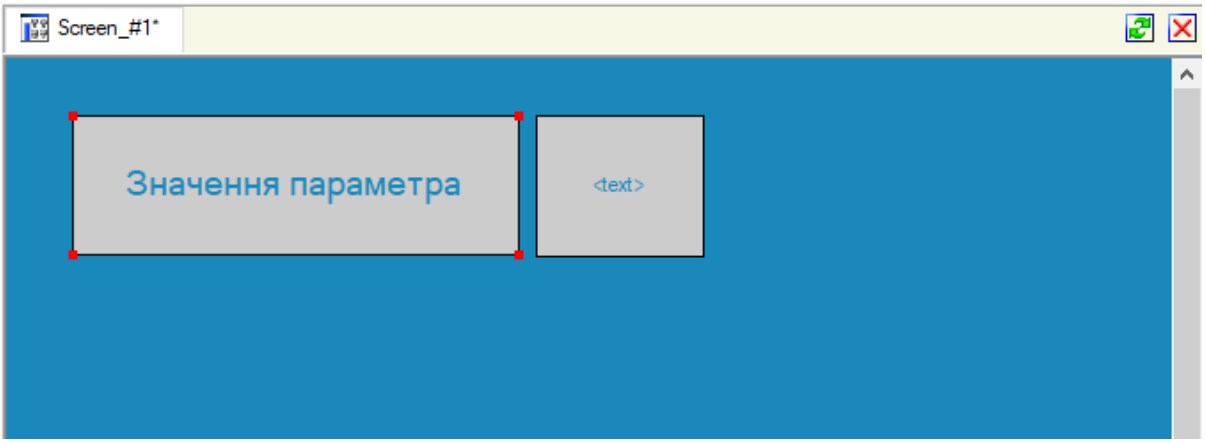


Рисунок 13 – Вигляд графічного екрана

1.4 Створити стрілочний прилад та прив'язати його показання до створеного в п. 1.3 аргументу.

Створимо новий тип ГЕ **Pointer Gage** (Стрілочний прилад) та зв’яжемо його показання з існуючим на екрані динамічним текстом. На інструментальній панелі редактора графічного вікна **Screen_#1:1** (Екран_#1:1) подвійним натисканням ЛК на кнопці **Gages** (Прилади) вибирається іконка **Pointer Gage** (Стрілочний прилад). За допомогою ЛК розміщується ГЕ приладу під існуючими текстовими ГЕ, як показано на рис. 14. У вікні його властивостей подвійним натисканням ЛК на рядку **Monitored Value** (Контрольоване значення) відкривається вікно редактора аргументів та праворуч у рядку **Link** (Прив’язка) вибирається створений аргумент екрана **Параметр**. Підтвердження здійснюється натисканням ЛК кнопки **Ok** (Готово).

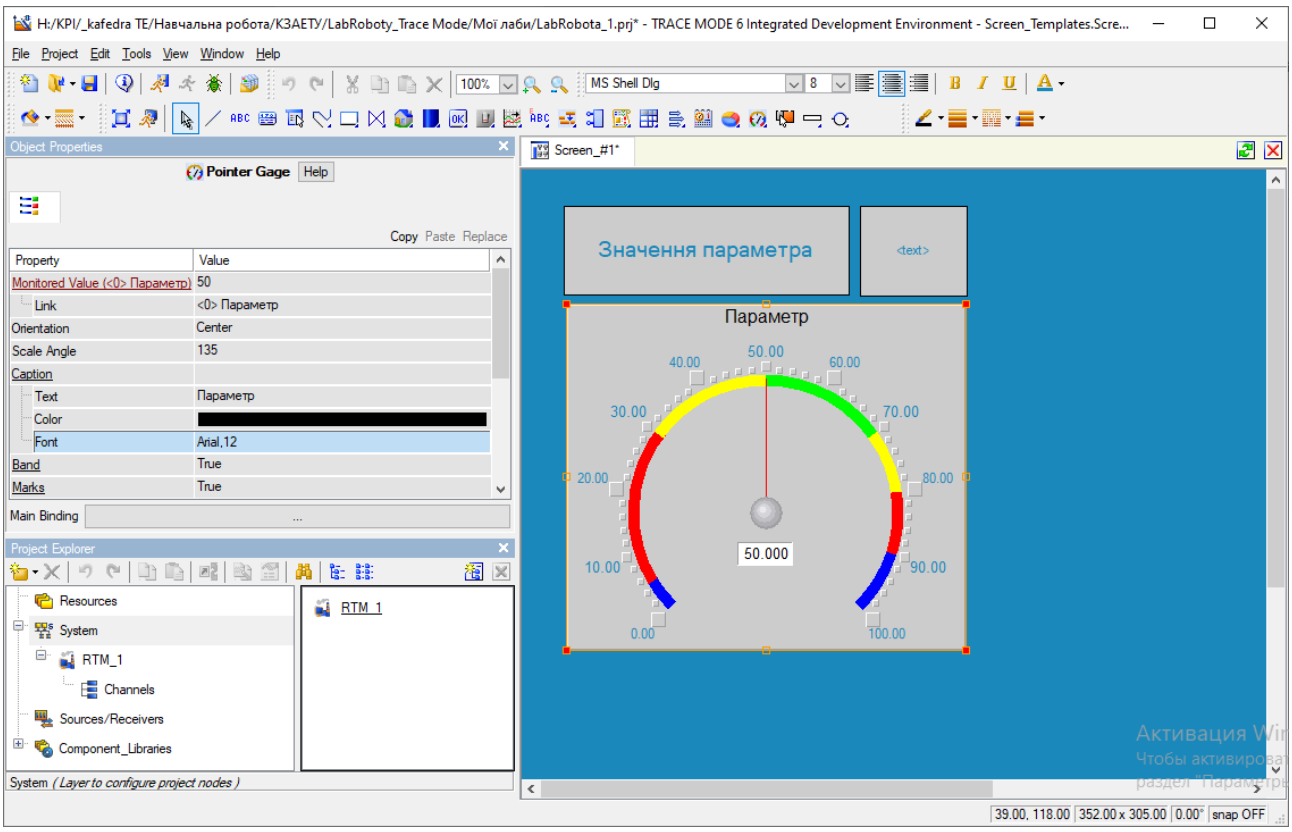


Рисунок 14 – Стрілочний прилад

Для зміни встановленого за згодою імені **Стрілочного приладу** здійснюється подвійне натискання ЛК у вікні властивостей ГЕ на команді **Caption** (Заголовок) та вводиться з клавіатури у рядок Text (**Текст)** нове ім’я приладу **Параметр**. В результаті графічний екран матиме вигляд, зображений на рис. 14. У вікні властивостей ГЕ приладу обов’язково задати бажані параметри шкали, колір залиття поверхні, тощо. Детальну інформацію про ГЕ можна отримати за допомогою кнопки **Help** (Довідка).

1.5 Автопобудувати канал.

Для створення каналу у вузлі проекту по аргументу екрана скористаємося механізмом автопобудови. Для вузла **RTM_1** у правому вікні навігатора проекту за допомогою правої клавіші комп’ютерної мишки (ПК) на **Screen_#1:1** (Екран_#1:1) викликати контекстне меню та ЛК вибрати пункт **Properties** (Властивості) (рис. 15).

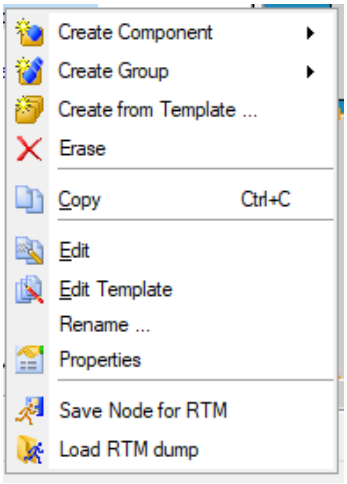


Рисунок 15 – Вибір властивостей екрана

У відкритому вікні вибрати вкладку **Arguments** (Аргументи), а в ній виділити ЛК наявний аргумент **Параметр**. На інструментальній панелі вікна ЛК натиснути кнопку **Create & link channels from arguments** (Створення та зв'язування каналів з аргументами), створивши таким чином канал з іменем **Параметр** (рис. 16).

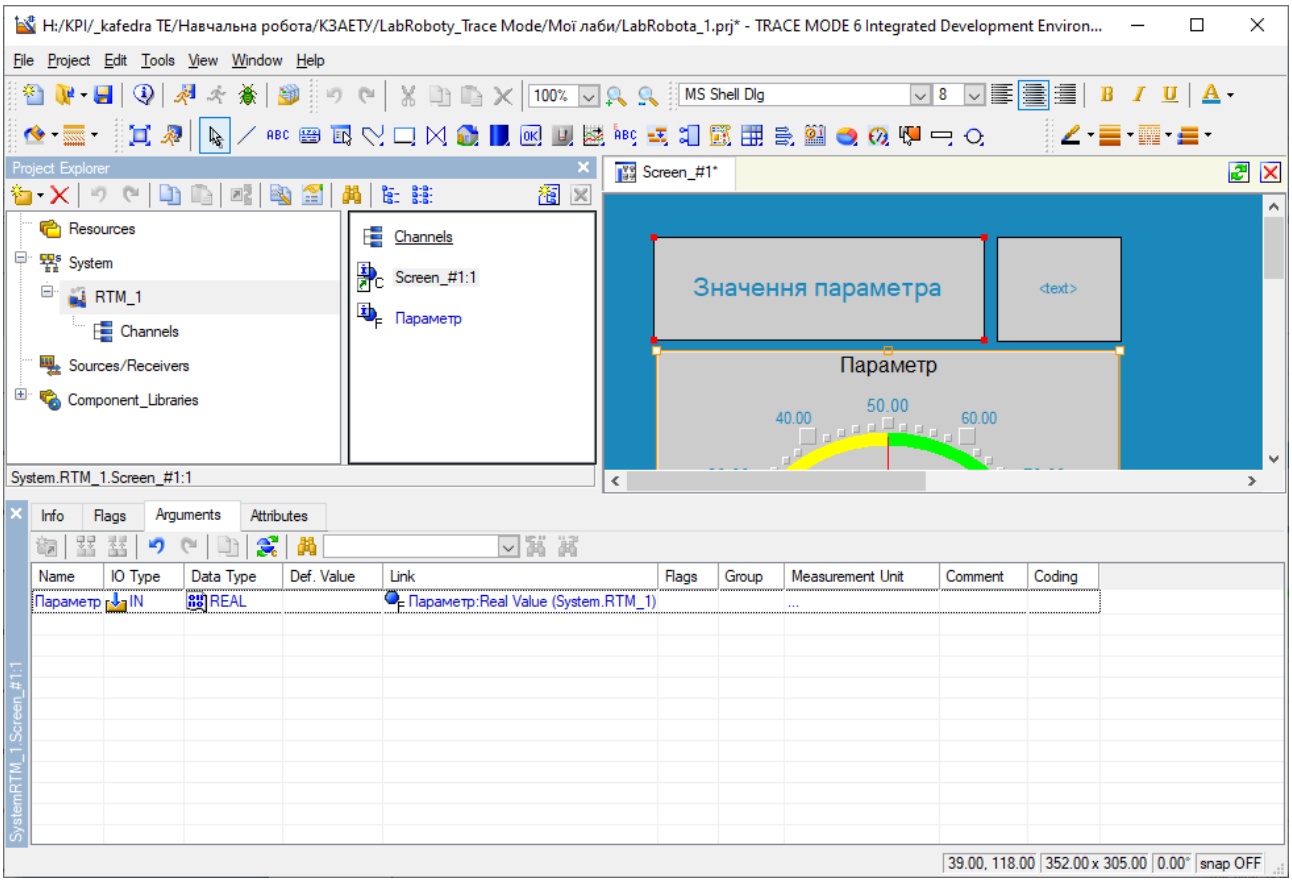


Рисунок 16 – Автопобудова каналу

1.6 Створити джерело синусоїдного сигналу та прив'язати його до каналу.

Уведемо до складу проекту джерело сигналу – вбудований (внутрішній) генератор синусоїдного сигналу та зв'яжемо його зі створеним каналом. У лівому вікні навігатора проекту для шару **Sources/Receivers** (Джерела/Приймачі) за допомогою ПК створити групу **Generators** (Генератори), як показано на рис. 17.

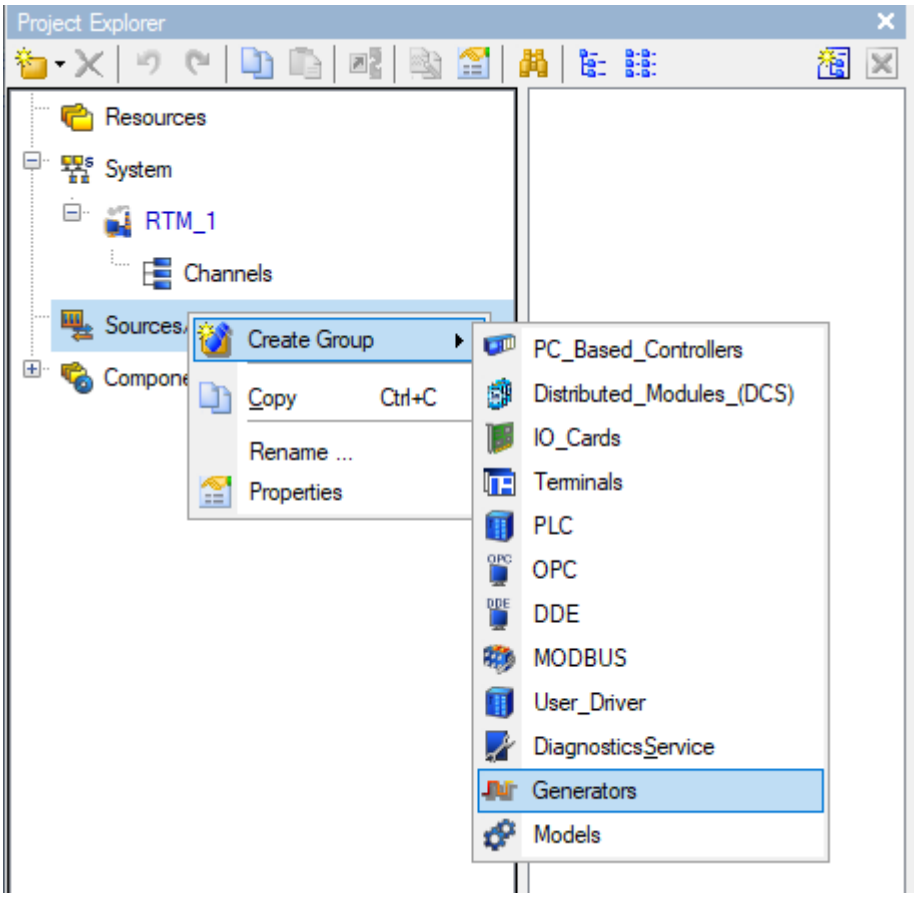


Рисунок 17 – Створення групи **Генератори**

За допомогою ПК на створеній групі **Generators_1** (Генератори_1) створити компоненту **Sinus** (Синусоїда) (рис. 18).

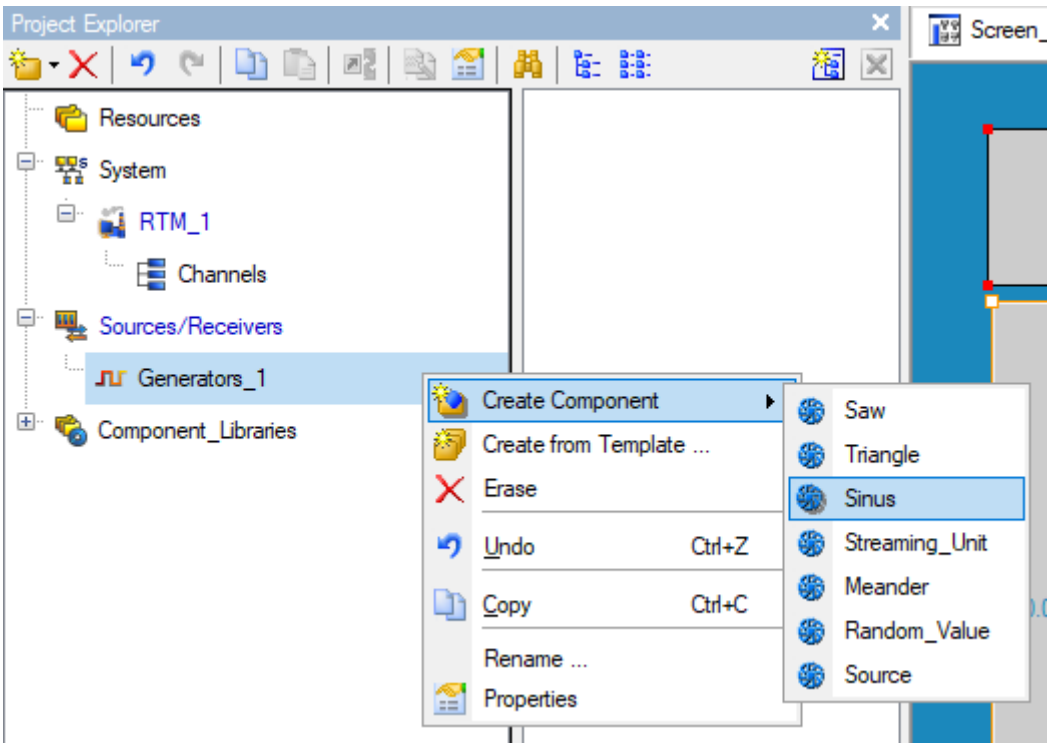


Рисунок 18 – Створення компоненти **Синусоїда**

Виділити ЛК у правому вікні створене джерело **Sinus#1** (Синусоїда#1), і не відпускаючи ЛК, перемістити курсор на вузол **RTM_1** у лівому вікні навігатора проекту, а після відкриття у правому вікні списку його компонентів, перемістити курсор на компоненту **Параметр** та відпустити ЛК (рис. 19).

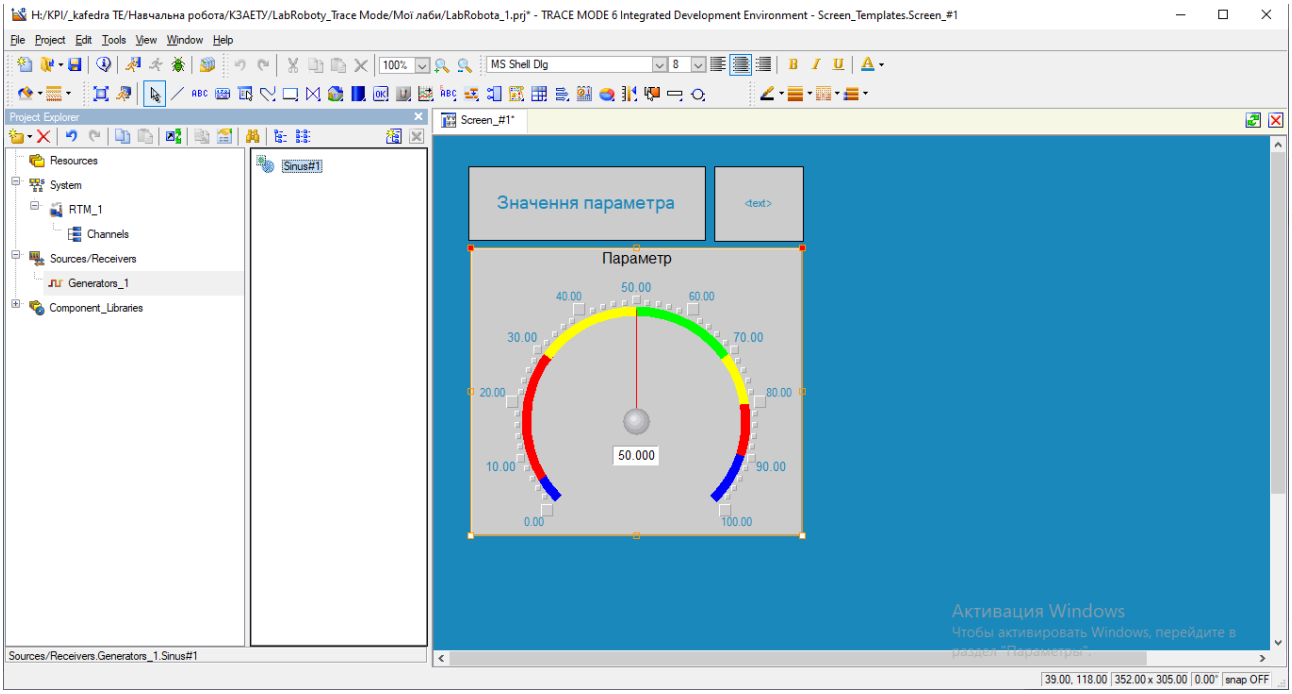


Рисунок 19 – Зв'язок компоненти **Синусоїда** з каналом **Параметр**

1.7 Вказати у проекті свої дані.

На графічному екрані розмістити у зручному місці шифр групи та своє ПІБ аналогічно до п. 1.2.

В меню **File\Information...**(Файл\Інформація...) в рядку **Owner** (Власник) написати за допомогою клавіатури своє ПІБ, а в рядку **Organization** (Організація) – шифр групи та факультету.

1.8 Запустити проект в роботу.

Зберегти проект натисканням кнопки . На головній інструментальній панелі ICP натиснути ЛК кнопку **Save for RTM** (Зберегти для МРЧ), скомпілювавши проект для запуску в реальному часі. Виділити ЛК в навігаторі проекту вузол **RTM_1**, а потім натиснути на головній панелі кнопку **Start Profiler** (Запустить профайлер). У вікні, що відкрилося, запустити проект на виконання шляхом натискання ЛК кнопки **Run** (Пуск/Зупинка). На екрані праворуч від напису **Значення параметра** повинно відображатися значення синусоїдного сигналу відповідно до показань стрілочного приладу, як показано на рис. 20.

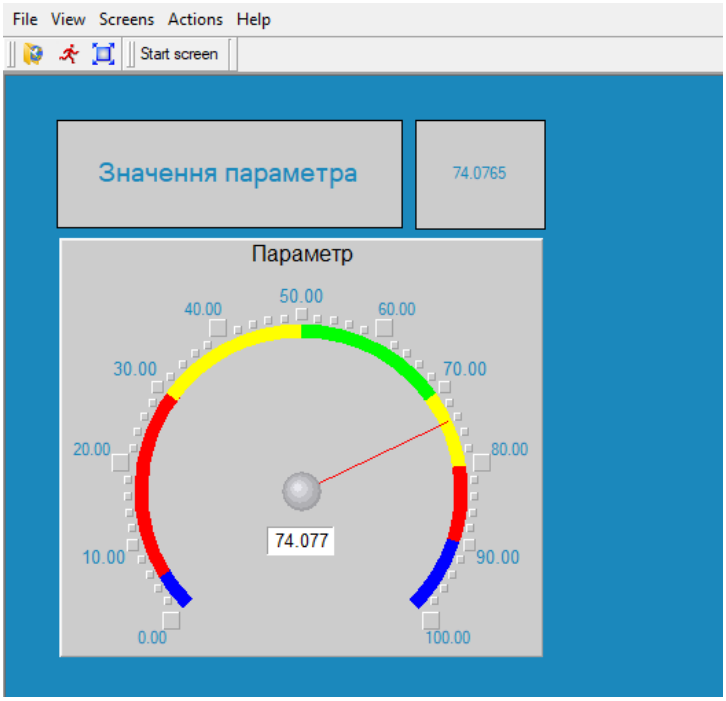


Рисунок 20 – Результат імітаційного запуску проекту

Зупинити виконання проекту шляхом натискання на панелі інструментів ЛК кнопки **Run** (Пуск/Зупинка). Закрити вікно графічного екрану через меню **File\Exit** (Файл\Вихід) або засобами Windows. Зберегти проект натисканням кнопки та завершити виконання роботи.

1.9 Оформити протокол комп'ютерного практикуму відповідно до вимог, наведених у вступі.

2 ПИТАННЯ ДЛЯ ЗАХИСТУ КОМП'ЮТЕРНОГО ПРАКТИКУМУ № 1

1. Етапи створення систем керування на основі SCADA.
2. Створення нового, відкриття існуючого проекту системи автоматизації.
3. Збереження проекту системи автоматизації.
4. Створення вузла автоматизованого робочого місця.
5. Розміщення графічних елементів на екрані.
6. Завдання властивостей графічних елементів.
7. Прив'язка показань стрілочного приладу до заданого аргументу.
8. Автопобудова каналу.
9. Вбудовані джерела сигналів.
10. Прив'язка джерел сигналів до заданого каналу.
11. Запуск проекту в роботу.

Комп’ютерний практикум № 2

ЗАПУСК ТА НАЛАГОДЖЕННЯ ПРОЕКТУ СИСТЕМИ АВТОМАТИЗАЦІЇ З ФУНКЦІЄЮ КЕРУВАННЯ

Мета роботи – засвоєння послідовності запуску та налагодження проекту системи автоматизації з функцією керування в SCADA-системі.

1 ПРОГРАМА РОБОТИ

1.1 Відкрити проект, створений у роботі № 1, за допомогою меню **File\Open...** (Файл\Відкрити...) або кнопки **Open existing project** (Відкрити існуючий проект) на інструментальній панелі.

1.2 Доповнити проект функцією керування.

Уведемо до складу графічного екрана засіб, що дозволяє вводити числові значення сигналу завдання з клавіатури, та створимо новий аргумент екрана для прийому цих значень. Викликати графічний екран на редагування. Вибрати ЛК на інструментальної панелі графічного редактора екрана ГЕ **Buttons** (Кнопки), а в ньому іконку одиночної кнопки **Button**. Розмістити ГЕ на екрані під стрілочним приладом (рис. 1).

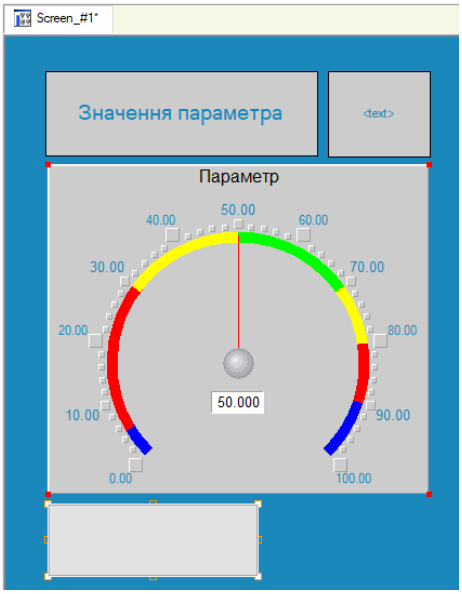


Рисунок 1 – Вид графічного екрана

Зберегти проект з новим ім’ям шляхом вибору в меню **File** (Файл) пункту **Save As...** (Зберегти як...), задавши з клавіатури бажане ім’я, наприклад **Robota_2**.

Подвійним натисканням ЛК на розміщеному ГЕ відкрити вікно його властивостей (рис. 2). У рядок праворуч від властивості **Text** (Текст) ввести з клавіатури напис **Керування**. У вікні властивостей ГЕ обов’язково задати бажані інші графічні параметри.

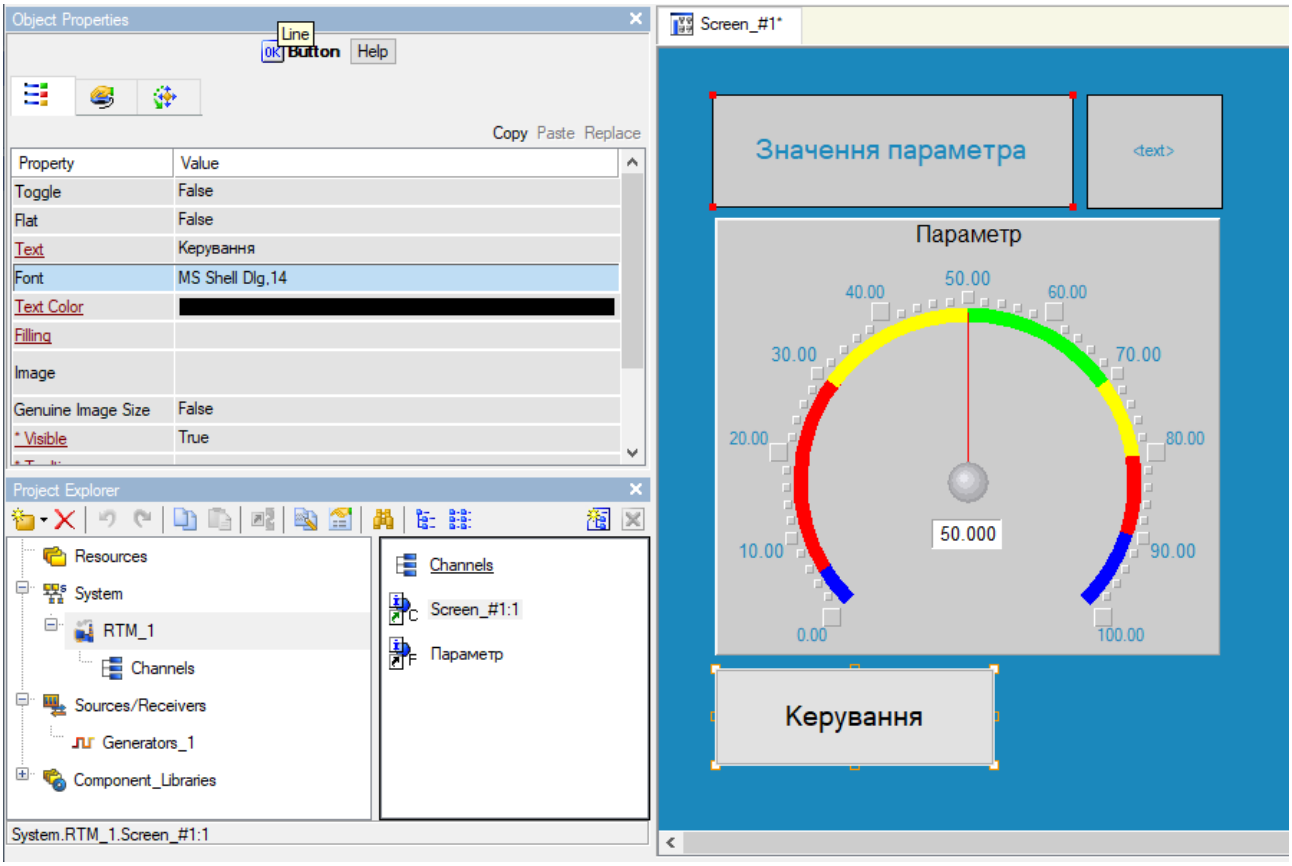


Рисунок 2 – Вікно властивостей графічного елемента **Кнопка**

Відкрити у вікні властивостей ГЕ **Button** (Кнопка) середню закладку **Actions** (Дії), а потім ПК розкрити меню **MousePress** та вибрати зі списку команду **Send Value** (Передати значення) (рис. 3).

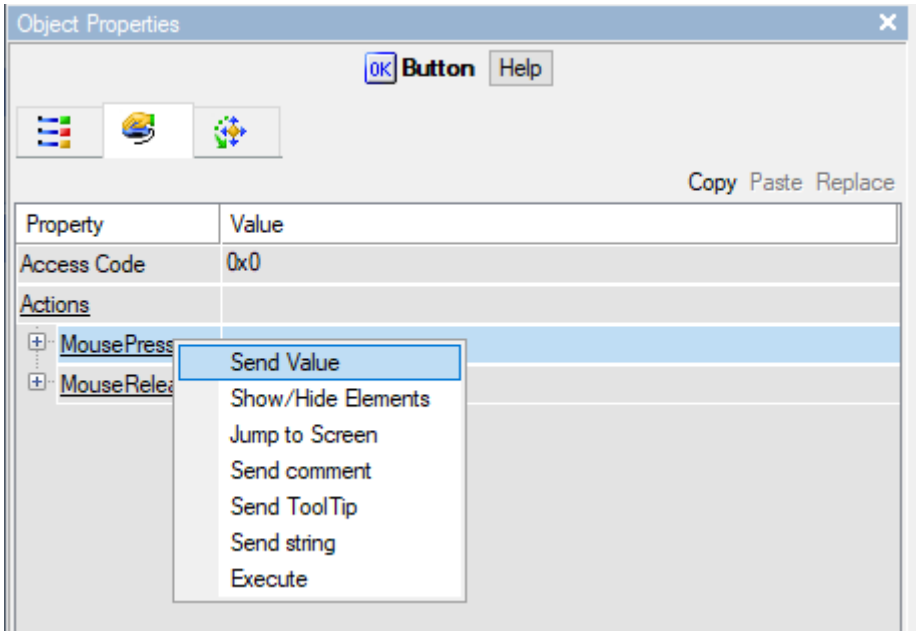


Рисунок 3 – Вікно закладки **Дії**

Праворуч у полі **Send Type** (Тип передачі) вибрати зі списку **Enter & Send** (Ввести і передати), а потім ЛК праворуч у полі **Destination** (Результат) викликати табличний редактор аргументів (рис. 4).

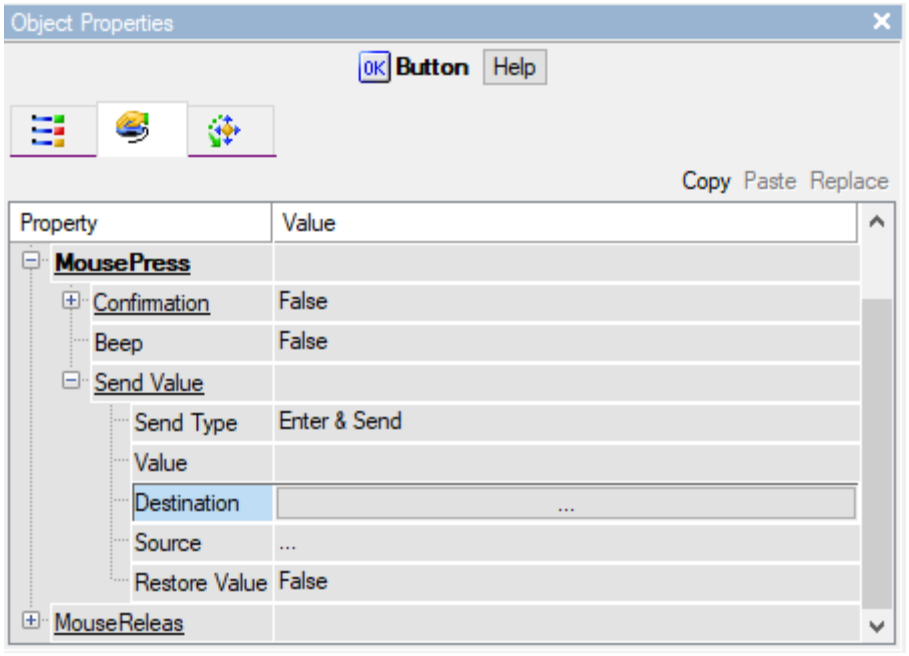


Рисунок 4 – Настроювання графічного елемента **Кнопка**

У відкритому вікні за допомогою кнопки **Create Argument** (Створити аргумент) створити новий аргумент і задати йому з клавіатури ім'я

Керування. У стовбці **IO Type** (тип вводу/виводу) за допомогою ЛК змінити тип аргументу з **IN** (ввід) на **IN/OUT** (ввід/вивід) та кнопкою **Ok** (Готово) підтвердити прив'язку атрибута ГЕ до цього аргументу (рис. 5).

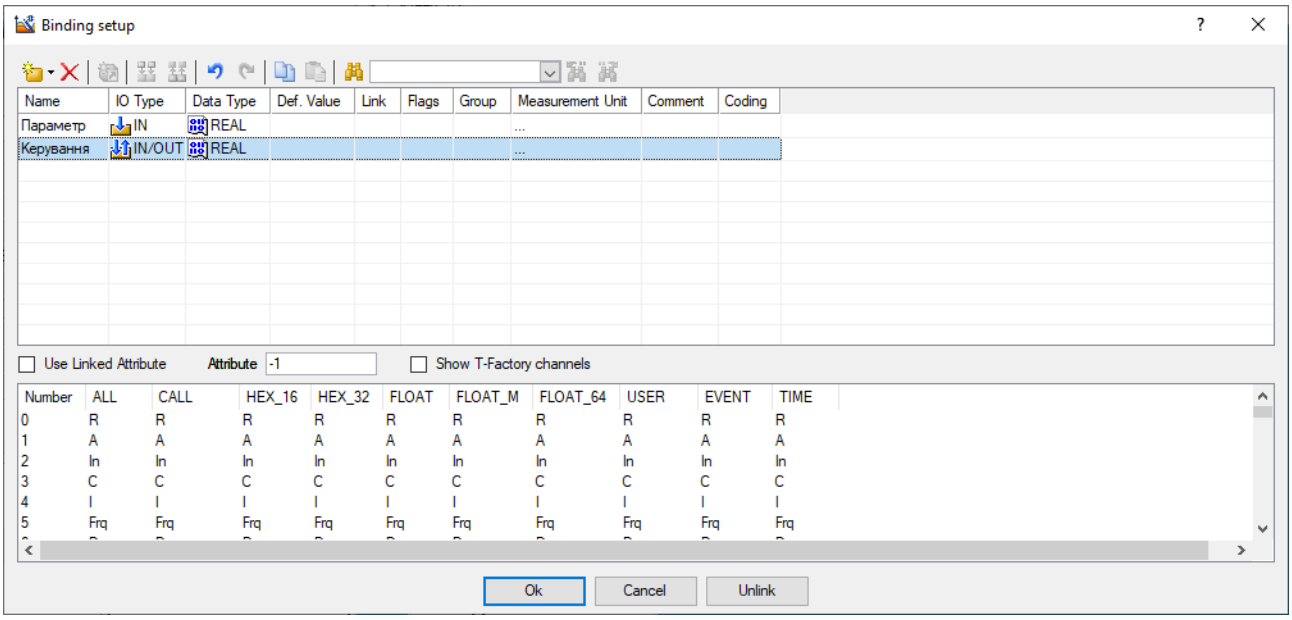


Рисунок 5 – Створення нового аргументу **Керування**

Створимо на екрані праворуч від ГЕ **Кнопка** графічний елемент **Text** (Текст) для відображення введених значень сигналу керування (рис. 6).

Рисунок 6 – Розміщення ГЕ **Текст** праворуч від ГЕ **Кнопка**

На вкладці його основних властивостей **Generals Properties** подвійним натисканням ЛК на рядку **Text** викликається меню **Indication type** (Вид індикації). У правому полі відкритого рядка натискається ЛК та із списку доступних типів вибирається **Value** (Значення). Після натискання ЛК у правому полі **Binding** (Прив'язка) з'являється вікно **Binding setup** (Властивості прив'язки) табличного редактора аргументів шаблону екрана. Виділити ЛК у списку аргумент **Керування** й натисканням ЛК на екранній кнопці **Ok** (Готово) підтвердити прив'язку атрибута ГЕ **Текст** до даного аргументу шаблону екрана. В результаті вікно властивостей ГЕ матиме вигляд, показаний на рис. 7.

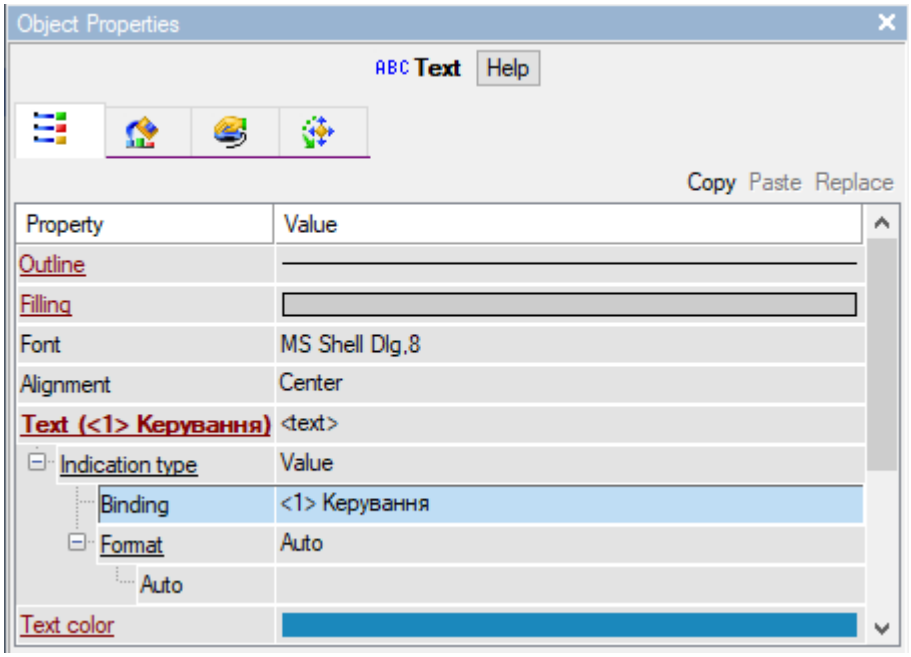


Рисунок 7 – Властивості графічного елемента **Текст**

1.3 Прив'язати аргумент екрана до нового каналу.

Створимо по аргументу **Керування** шаблону екрана новий канал та відредагуємо його прив'язку. У шарі **System** (Система) відкрити вузол **RTM_1**. Натисканням ПК на компоненті **Screen_#1:1** (Екран_#1:1) викликати контекстне меню та вибрати пункт **Properties** (Властивості). У вікні вибрати вкладку **Arguments** (Аргументи) та ЛК виділити аргумент **Керування** і за допомогою кнопки **Create & link channels from arguments** (Створення та зв'язування каналів з аргументами) створити новий канал (рис. 8).

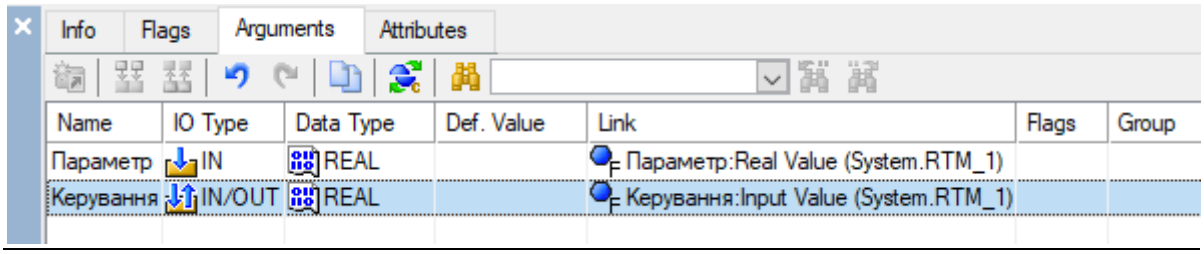


Рисунок 8 – Вікно властивостей екрана

В результаті у вузлі **RTM_1** буде автопобудовано канал з іменем **Керування** (рис. 9).

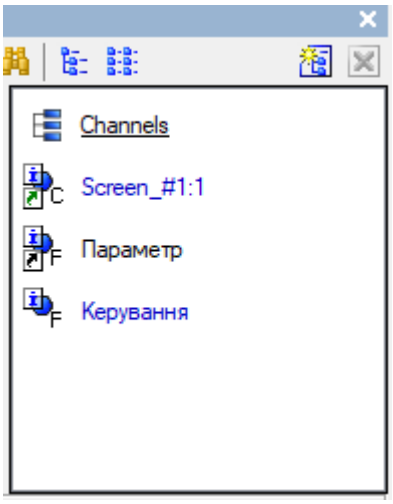


Рисунок 9 – Автопобудований канал **Керування**

У полі **Link** (Прив’язка) відкритого вікна властивостей екрана (рис. 8) для аргументу **Керування** викликати подвійним натисканням ЛК вікно настроювання зв'язку. Вибрати в ньому атрибут **Input Value** (Вхідне значення) та кнопкою **Link** підтвердити зв'язок аргументу екрана **Керування** з атрибутом **Вхідне значення** каналу **Керування** (рис. 10).

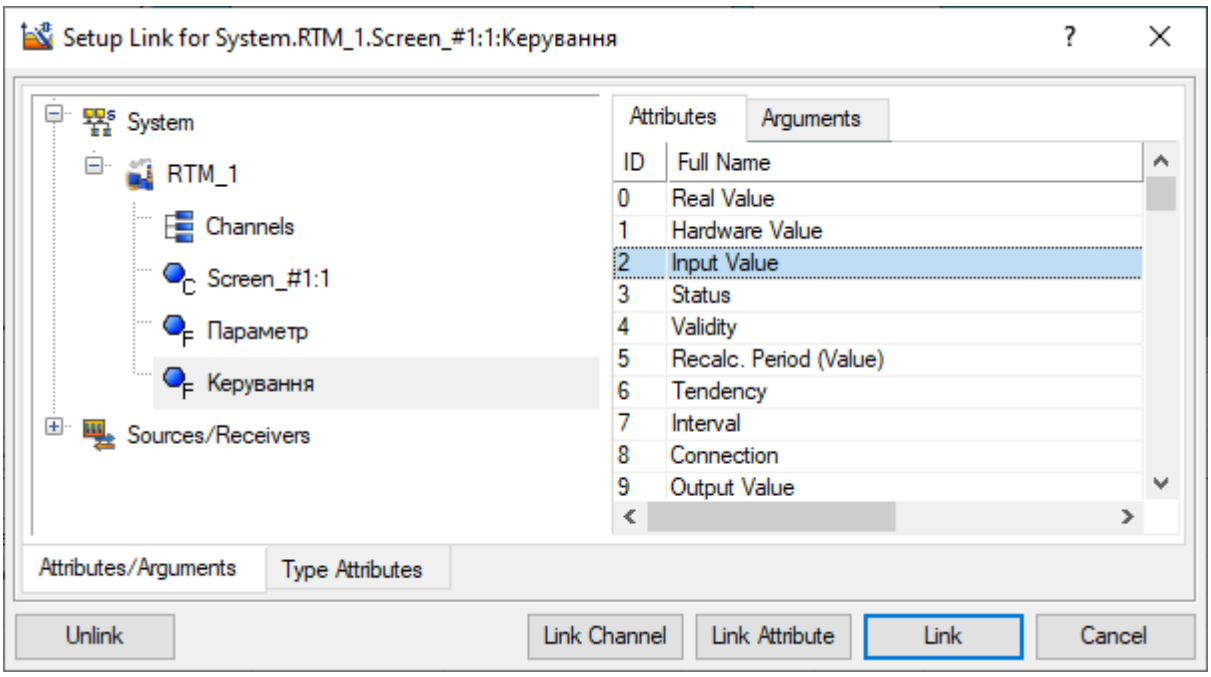


Рисунок 10 – Конфігурація системи

1.4 Розмістити на екрані ГЕ **Trend** (Тренд).

Доповнимо екран новим ГЕ **Trend** (Тренд) для перегляду змін значень каналів **Параметр** та **Керування** у часі та відстеження передісторії. Вибрати ЛК на інструментальної панелі графічного редактора екрана ГЕ **Trends** (Тренди), а в ньому іконку ГЕ **Trend** (Тренд) та розмістити його на екрані праворуч від наявних ГЕ. Основні властивості ГЕ **Тренд** залишити заданими за згодою (автоматично). Перейти у праву вкладку вікна властивостей **Curves** (Криві), виділивши ЛК рядок **Curves**, а потім за допомогою ПК послідовно створити дві нові криві **Curve**. Налаштувати їхні прив'язки до аргументів, а також товщину й колір ліній як показано на рис. 11. В результаті ГЕ **Тренд** прийме вигляд згідно рис. 12.

У вікні властивостей ГЕ самостійно освоїти завдання інших його параметрів. Інформацію про ГЕ можна отримати за допомогою кнопки **Help** (Довідка).

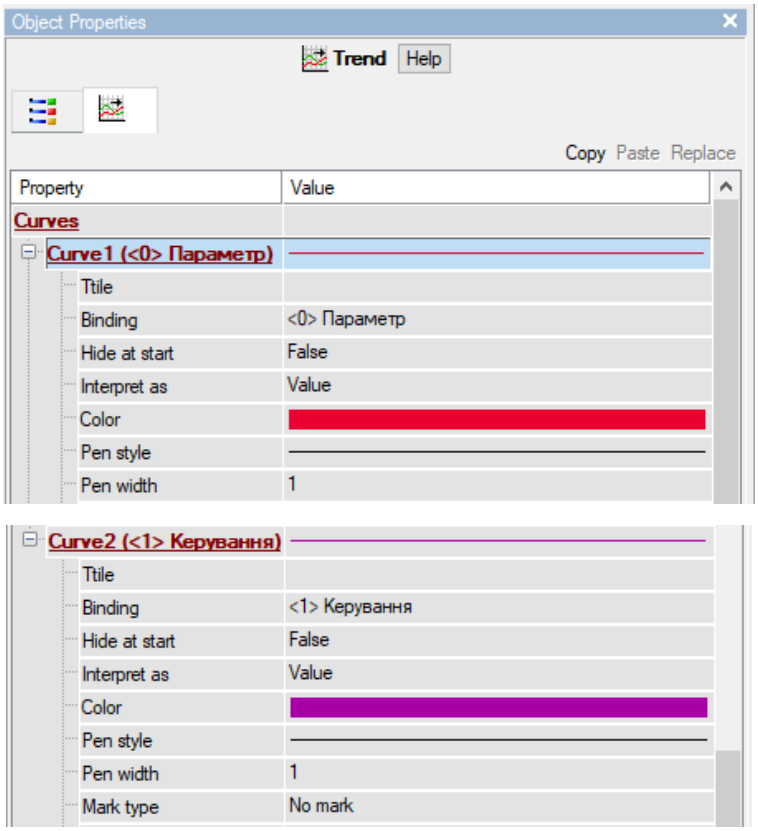


Рисунок 11 – Властивості **Кривих** графічного елемента **Тренд**

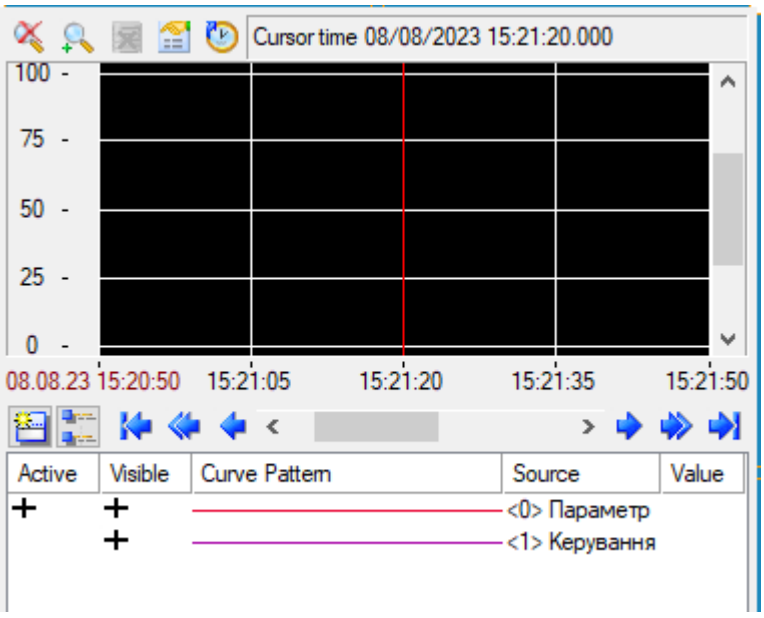


Рисунок 12 – Вигляд графічного елемента **Тренд**

1.5 Запустити проект в роботу.

Зберегти проект натисканням кнопки . На головній інструментальній панелі натиснути ЛК кнопку **Save for RTM** (Зберегти для МРЧ),

скомпілювавши проект для запуску в реальному часі. Виділити ЛК в навігаторі проекту вузол **RTM_1**, а потім натиснути на головній панелі кнопку **Start Profiler** (Запустить профайлер). У відкритому вікні запустити проект на виконання шляхом натискання ЛК кнопки **Run** (Пуск/Зупинка). Натиснути кнопку **Керування** та ввести бажане значення сигналу завдання. Спостерігати результат роботи системи на стрілочному приладі та на тренді. Освоїти меню та кнопки ГЕ **Тренд**, зокрема: зміна активності та видимості кривих, переміщення по екрану тренда різними способами, тощо.

Зупинити виконання проекту шляхом натискання на панелі інструментів ЛК кнопки **Run** (Пуск/Зупинка). Закрити вікно графічного екрану через меню **File\Exit** (Файл\Вихід) або засобами Windows.

1.6 Здійснити просту обробку даних.

Створимо у проекті шаблон програми. На мові **Structured Text** (ST програма) будемо знаходити суму реальних значень існуючих каналів **Параметр** та **Керування**, а результат помістимо в новостворений аргумент екрана **Сума** без застосування додаткового каналу у вузлі проекту. Значення **Сума** будемо відображати на існуючих ГЕ **Текст** та **Тренд**. Для цього виділити ЛК та скопіювати на екрані ГЕ **Значення параметра** та **Параметр** над стрілочним приладом і розмістити їх нижче ГЕ **Кнопка**, як показано на рис. 13.

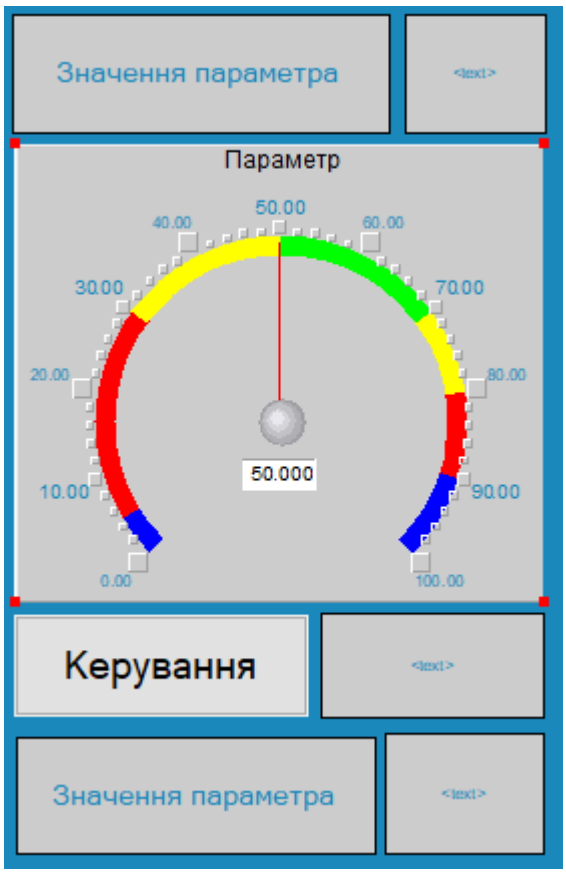


Рисунок 13 – Вигляд графічного екрана

Замінити статичний текст **Значення параметра** першого ГЕ на **Сума:**, як показано на рис. 14.

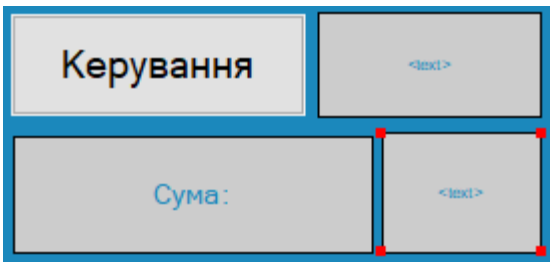


Рисунок 14 – Вид частини графічного екрана

Динаміку другого ГЕ прив'язати до третього аргументу екрана типу **IN** з іменем **Сума**, який створити в процесі прив'язки (рис. 15).

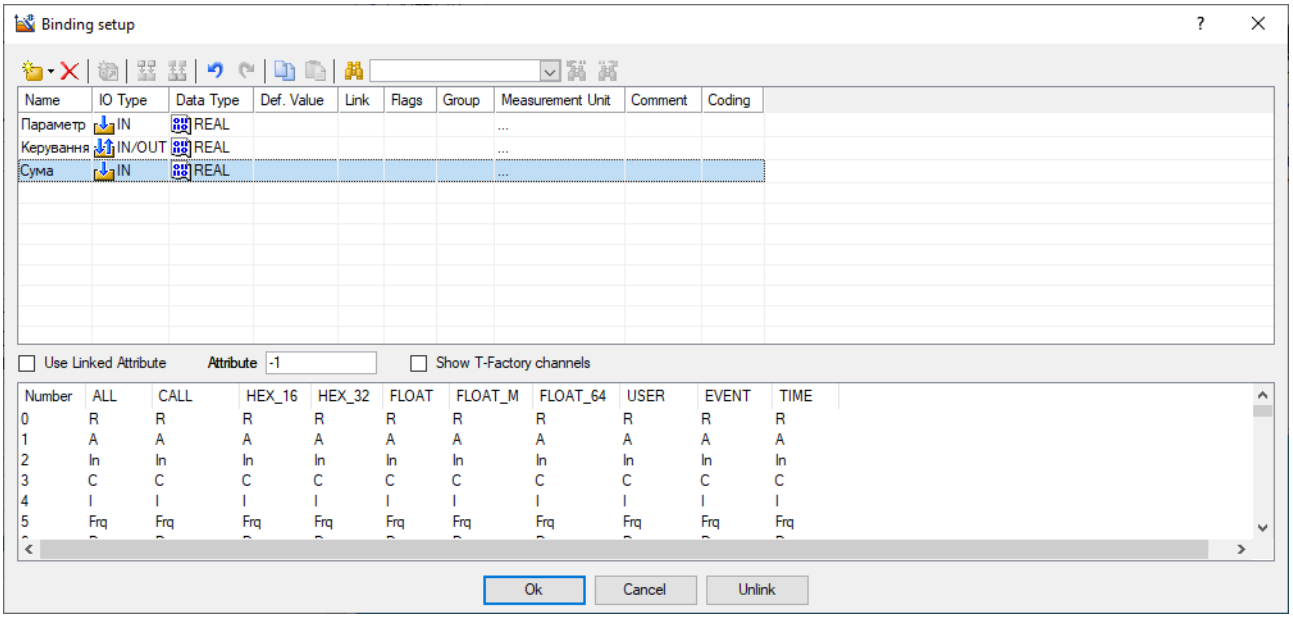


Рисунок 15 – Властивості прив'язки

Додати ще одну криву на існуючий **Тренд** із прив'язкою до аргументу **Сума** (рис. 16).

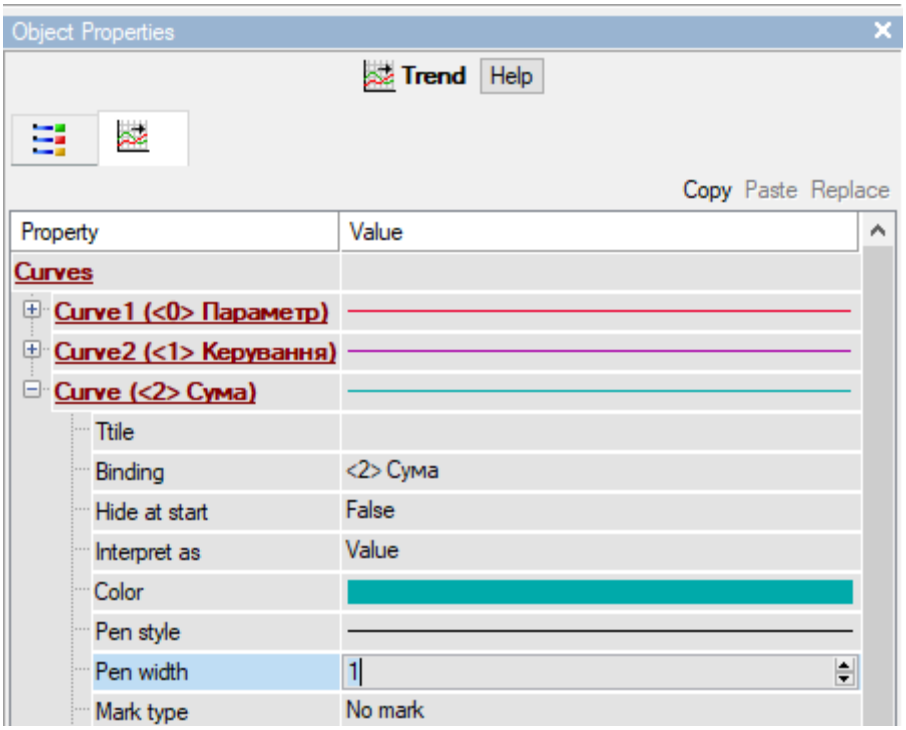


Рисунок 16 – Властивості кривої **Сума** графічного елемента **Тренд**

Подвійним натисканням ЛК відкрити вузол **RTM_1** та ПК створити в ньому компоненту **Program** (Програма) (рис. 17).

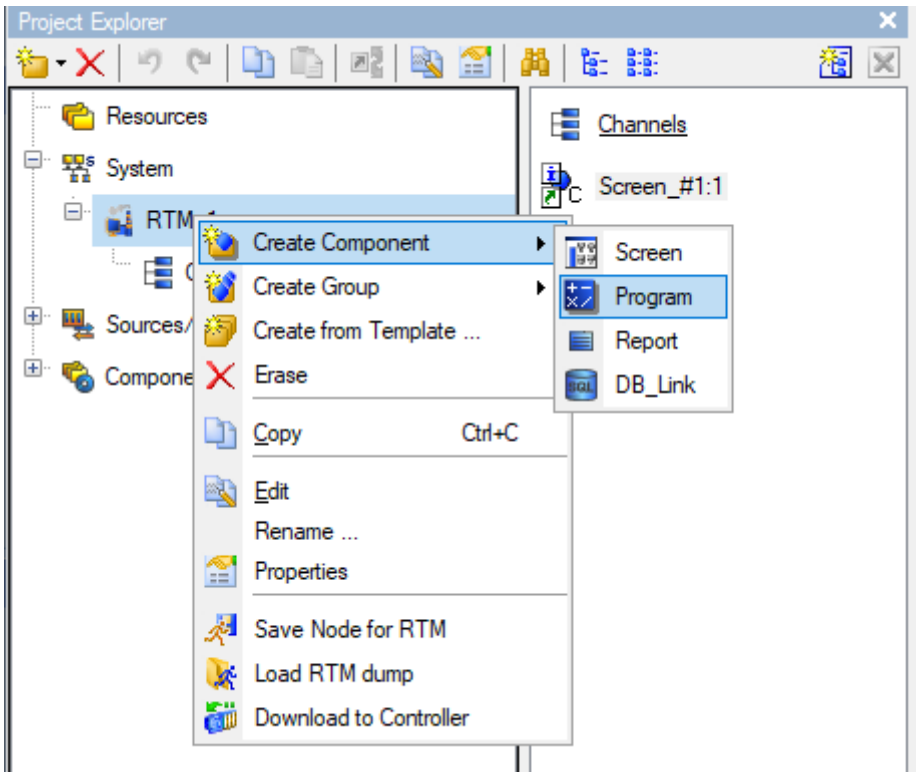


Рисунок 17 – Створення компоненти **Програма**

Виділити створену компоненту **Program#1:2** (Програма#1:2) та ПК викликати контекстне меню, вибравши в ньому ЛК пункт **Edit Template** (Редагувати шаблон), та перейти в режим редагування програми (рис. 18).

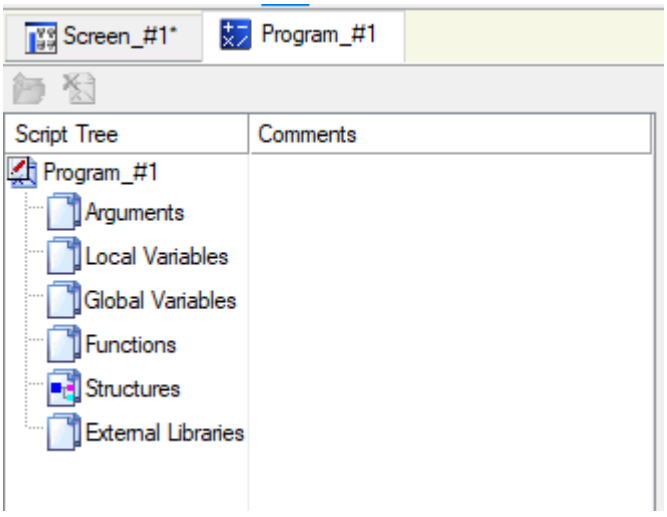


Рисунок 18 – Режим редагування програми

Виділити ЛК у дереві шаблону **Program#1** (Програма#1) рядок **Arguments** (Аргументи), викликавши табличний редактор аргументів. За

допомогою кнопки послідовно створити в редакторі аргументів три аргументи з іменами **Параметр**, **Керування** та **Сума**. Першим двом аргументам встановити тип **IN**, а третьому – **OUT** (рис. 19).

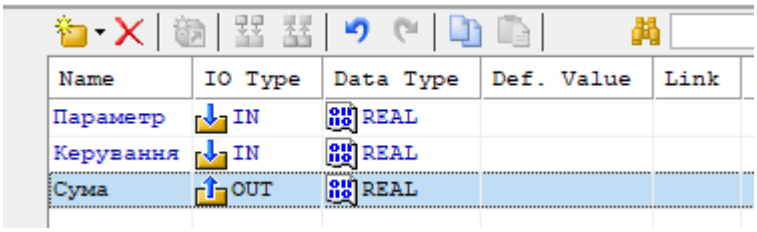


Рисунок 19 – Створення каналів програми

Виділити ЛК в дереві шаблону (рис. 18) рядок **Program#1** та у вікні вибрати мову програмування **Structured Text** (ST-програма) (рис. 20).

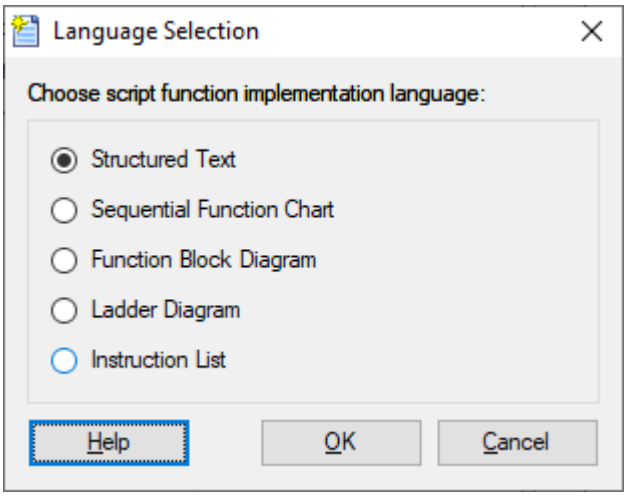


Рисунок 20 – Вибір мови програмування

Після натискання екранної кнопки **Ok** (Прийняти) у відкритому вікні редактора програм (рис. 21) після рядків з оголошеними змінними набрати з клавіатури у виділеному рядку наступний оператор:

Сума=Параметр+Керування;

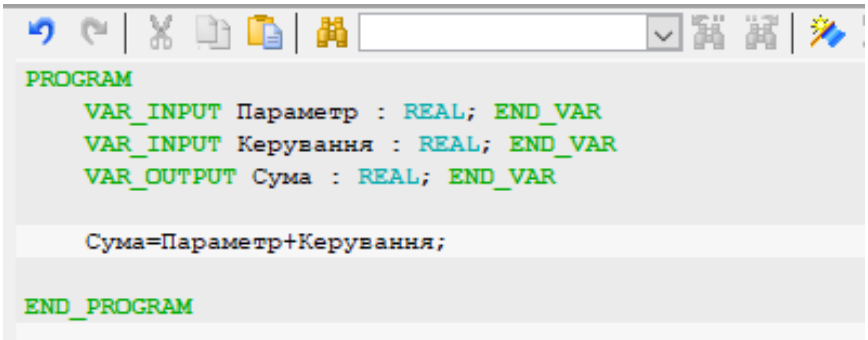


Рисунок 21 – Вид програми мовою **ST**

За допомогою іконки **Build (F7)** (Компіляція) на інструментальній панелі редактора або гарячої клавіші F7 скомпілювати програму та переконатися в успішній компіляції у вікні **Messages**, яке при необхідності викликається з інструментальної панелі за допомогою іконки **Messages** (Повідомлення) (рис. 22).

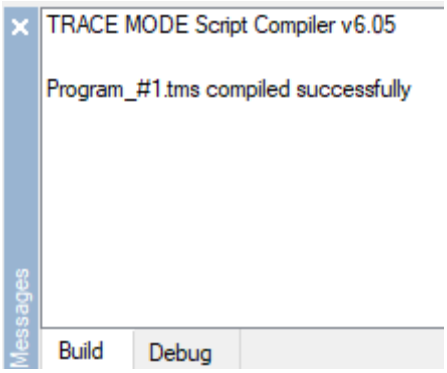


Рисунок 22 – Результат успішної компіляції програми

1.7 Прив'язати аргументи програми**.**

Виконаємо прив'язку аргументів програми до атрибутів каналів. У вікні навігатора проекту **Project Explorer** ПК на компоненті **Program#1:2** викликати контекстне меню та вибрати **Properties** (Властивості). У вікні, що відкрилося, вибрати вкладку **Arguments** (Аргументи)**.** Подвійним натисканням ЛК у стовпці **Link** (Прив'язка) для рядка **Параметр** відкрити вікно конфігурування зв'язку. Вибрати у лівій частині вікна елемент **Параметр**, потім у правій частині на вкладці **Attributes** (Атрибути) рядок **Real Value** (Реальне значення) та натиснути кнопку **Link** (Прив'язка) (рис. 23).

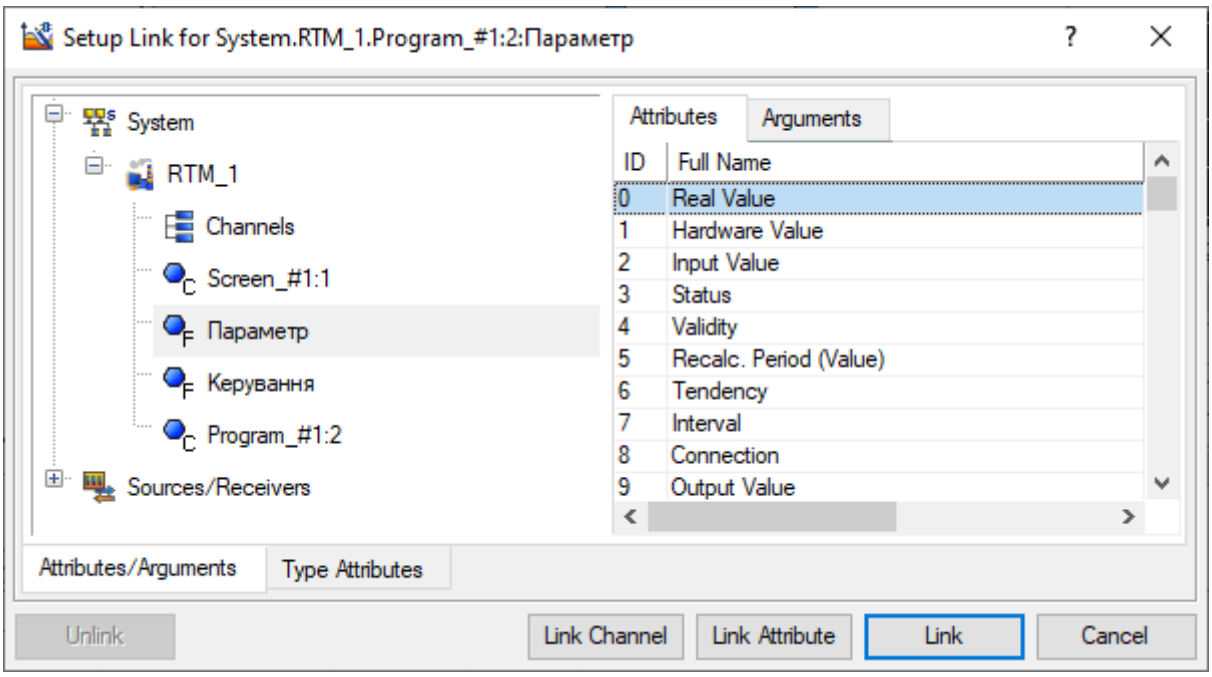


Рисунок 23 – Прив'язка аргументів програми до атрибутів каналів

Аналогічно подвійним натисканням ЛК у стовпці **Link** для рядка **Керування** відкрити вікно конфігурування зв’язку. Вибрати у лівій частині вікна елемент **Керування**, а потім у правій частині на вкладці **Attributes** рядок **Real Value** та натиснути кнопку **Link.**

Аналогічно подвійним натисканням ЛК у стовпці **Link** для аргументу програми **Сума** викликати вікно настроювання прив'язки. Вибрати у лівому вікні канал **Screen_#1:1**, а в правому вікні замість вкладки **Attributes** вибрати вкладку **Arguments** та вказати аргумент **Сума** й кнопкою **Link** підтвердити зв'язок (рис. 24). В результаті вікно властивостей компоненти **Program#1:2** буде мати вигляд, показаний на рис. 25.

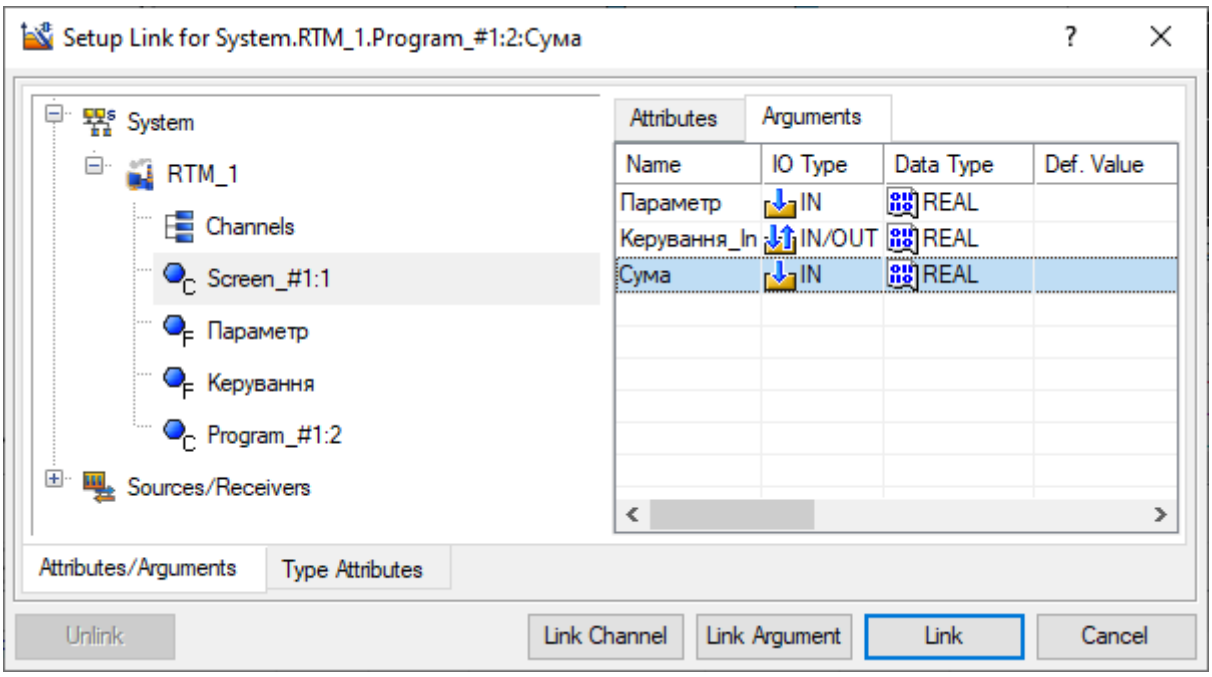


Рисунок 24 – Прив'язка аргументів програми до аргументів каналів

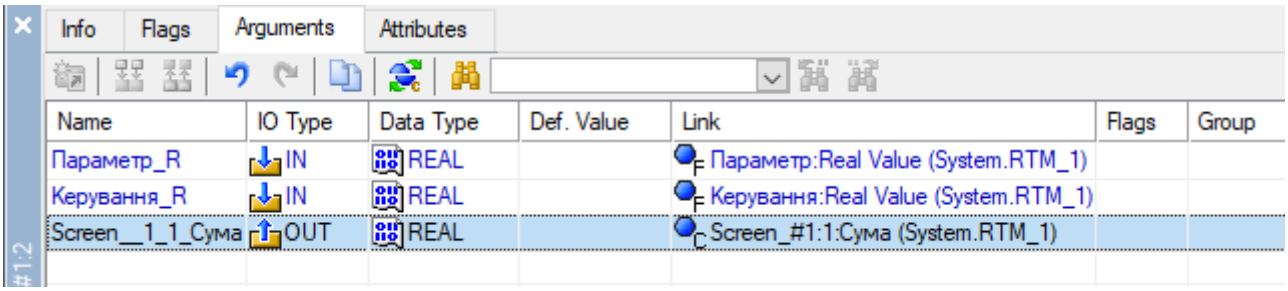


Рисунок 25 – Результати конфігурації

1.8 Вказати у проекті свої дані.

На графічному екрані розмістити у зручному місці шифр групи та своє ПІБ аналогічно до роботи № 1.

В меню **File\Information...**(Файл\Інформація...) в рядку **Owner** (Власник) написати за допомогою клавіатури своє ПІБ, а в рядку **Organization** (Організація) – шифр групи та факультету.

1.9 Запустити проект в роботу.

Зберегти проект натисканням кнопки . На головній інструментальній панелі ICP натиснути ЛК кнопку **Save for RTM** (Зберегти для МРЧ),

скомпілювавши проект для запуску в реальному часі. Виділити ЛК в навігаторі проекту вузол **RTM_1**, а потім натиснути на головній панелі кнопку **Start Profiler** (Запустить профайлер). У вікні, що відкрилося, запустити проект на виконання шляхом натискання ЛК кнопки **Run** (Пуск/Зупинка).

За допомогою кнопки **Керування** увести бажане значення сигналу керування і спостерігати відповідну зміна сигналу **Керування** і зсув сигналу **Сума** на ГЕ **Тренд** (рис. 26).

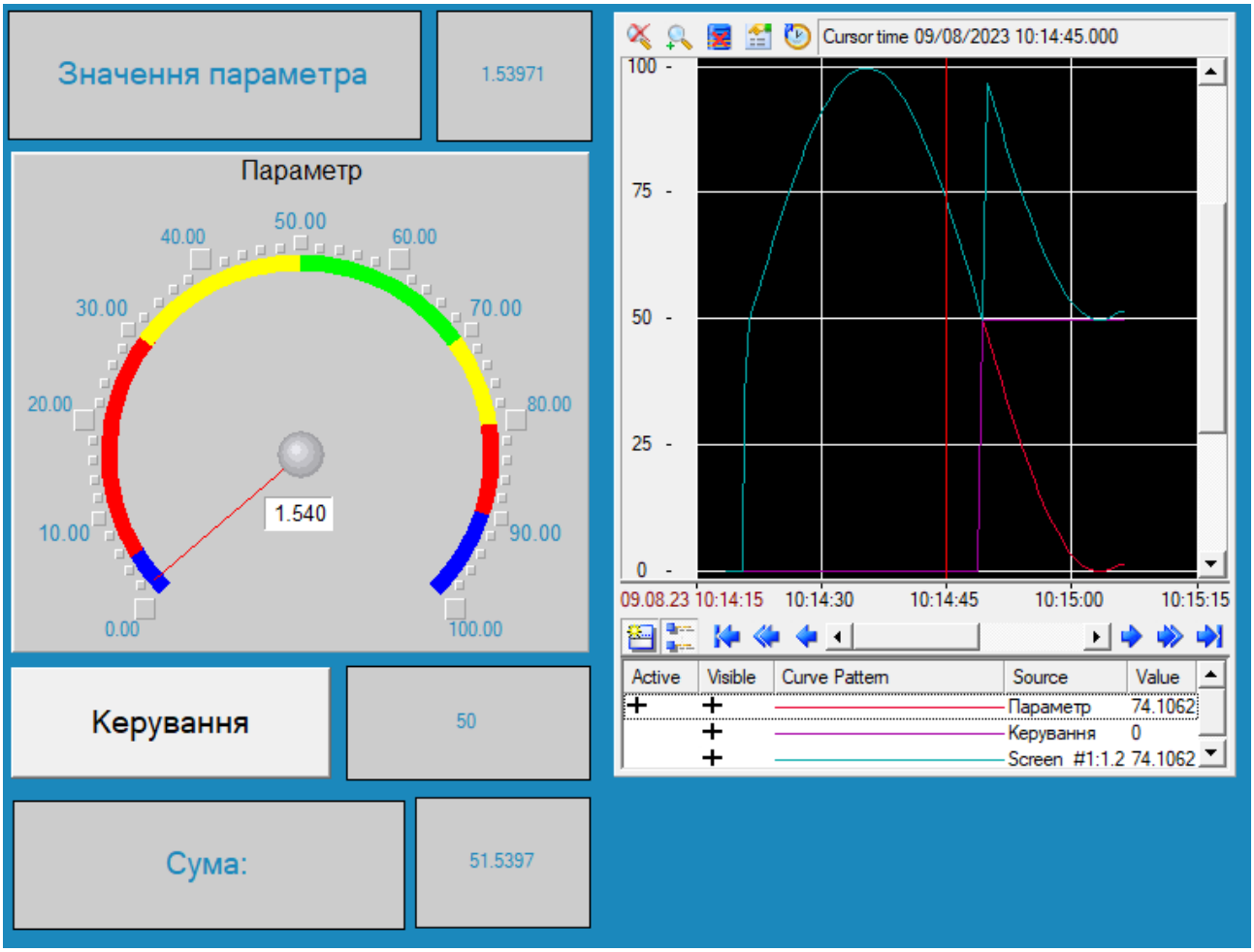


Рисунок 26 – Результат імітаційного запуску проекту

Зупинити виконання проекту шляхом натискання на панелі інструментів ЛК кнопки **Run** (Пуск/Зупинка). Закрити вікно графічного екрану через меню **File\Exit** (Файл\Вихід) або засобами Windows. Зберегти проект натисканням кнопки та завершити виконання роботи.

1.10 Оформити протокол комп’ютерного практикуму відповідно до вимог, наведених у вступі.

2 ПИТАННЯ ДЛЯ ЗАХИСТУ КОМП’ЮТЕРНОГО ПРАКТИКУМУ № 2

1. Етапи створення проекту систем автоматизації на основі SCADA.
2. Завдання властивостей ГЕ «Кнопка».
3. Завдання властивостей ГЕ «Тренд».
4. Створення нових аргументів та завдання їхніх параметрів.
5. Програмування мовою ST.
6. Автопобудова каналів.
7. Прив’язка аргументів.

Комп’ютерний практикум № 3

ПРОГРАМУВАННЯ ЛОГІЧНИХ ФУНКЦІЙ

Мета роботи – засвоєння методики програмування логічних функцій на мові FBD в SCADA-системі.

1 ПРОГРАМА РОБОТИ

1.1 Реалізувати логічну функцію в SCADA-системі мовою FBD відповідно до варіанту завдання, наведеного у розділі 2 даної роботи. Нижче викладається методика програмування на прикладі логічної функції $y = x_1 x_2 \overline{x}_4 + \overline{x}_1 \overline{x}_2 + \overline{x}_1 x_3 + \overline{x}_2 x_4$.

1.2 Створити новий проект стилю **Easy** (Простий), як у роботі № 1. Подвійним натисканням ЛК на компоненті **Screen_#1:1** (Екран_#1:1) відкрити вікно графічного редактора. Зберегти проект шляхом вибору в меню **File** (Файл) пункту **Save** (Зберегти) або натисканням кнопки на інструментальній панелі, задавши з клавіатури бажане ім’я, наприклад **Robota_3**.

1.3 Розмістити на екрані, як показано на рис. 1, п’ять ГЕ **Text** (Текст) для відображення чисельного значення чотирьох вхідних змінних **X1, X2, X3, X4** та одного вихідного значення **Y** логічної функції.

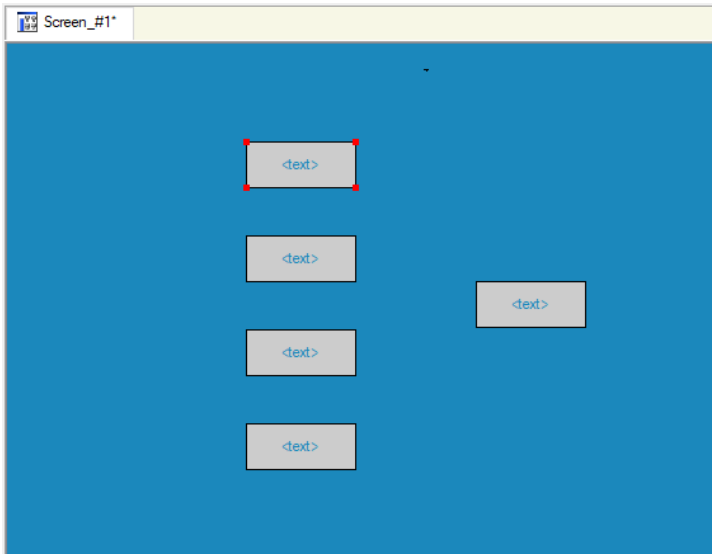


Рисунок 1 – Розташування п’яти ГЕ **Текст** на екрані

Виділити при натиснутій ЛК вертикальну групу ГЕ **Текст** та вирівняти їх по вертикалі, встановити однакові проміжки між ними, однакові розміри, тощо, за допомогою пунктів меню **Tools** (Інструменти).

Виділити на екрані ЛК верхній ГЕ **Текст** та відкрити вікно його властивостей. Подвійним натисканням ЛК на рядку **Text** відкрити меню **Indication type** (Тип індикації) (рис. 2). У правому полі відкритого рядка натискається ЛК та із списку доступних типів вибирається **Value** (Значення) (рис. 3).

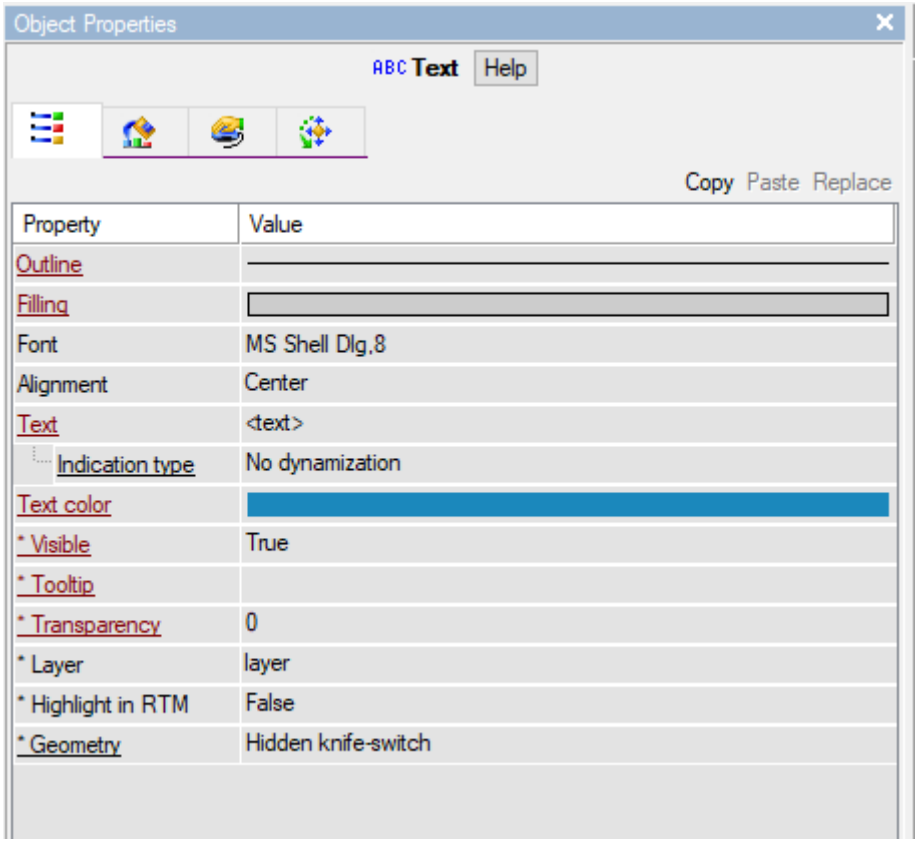


Рисунок 2 – Налаштування динамізації ГЕ **Текст**

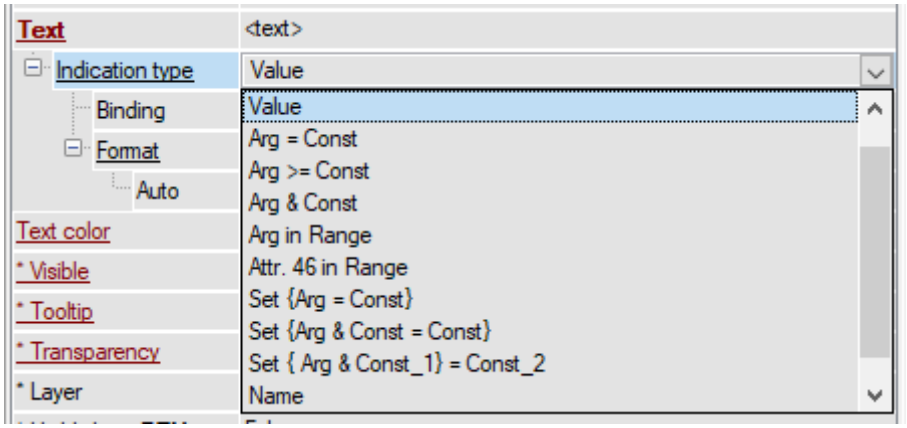


Рисунок 3 – Вибір типу динамізації ГЕ **Текст**

Після натискання ЛК у правому полі **Binding** (Прив'язка) з'являється вікно **Binding setup** (Властивості прив'язки). Створити аргумент екрана шляхом натискання кнопки на інструментальній панелі вікна. Аргументу автоматично надається ім'я **ARG_000.** Після подвійного натискання ЛК на імені **ARG_000** із клавіатури вводиться нове ім'я **X1**, ввід якого завершується клавішею **Enter** на клавіатурі. Встановити у третьому стовпці **Data Type** (Тип даних) логічний тип даних **BOOL** (рис. 4). Зв'язок з аргументом підтверджується натисканням у вікні кнопки **Ok** (Готово).

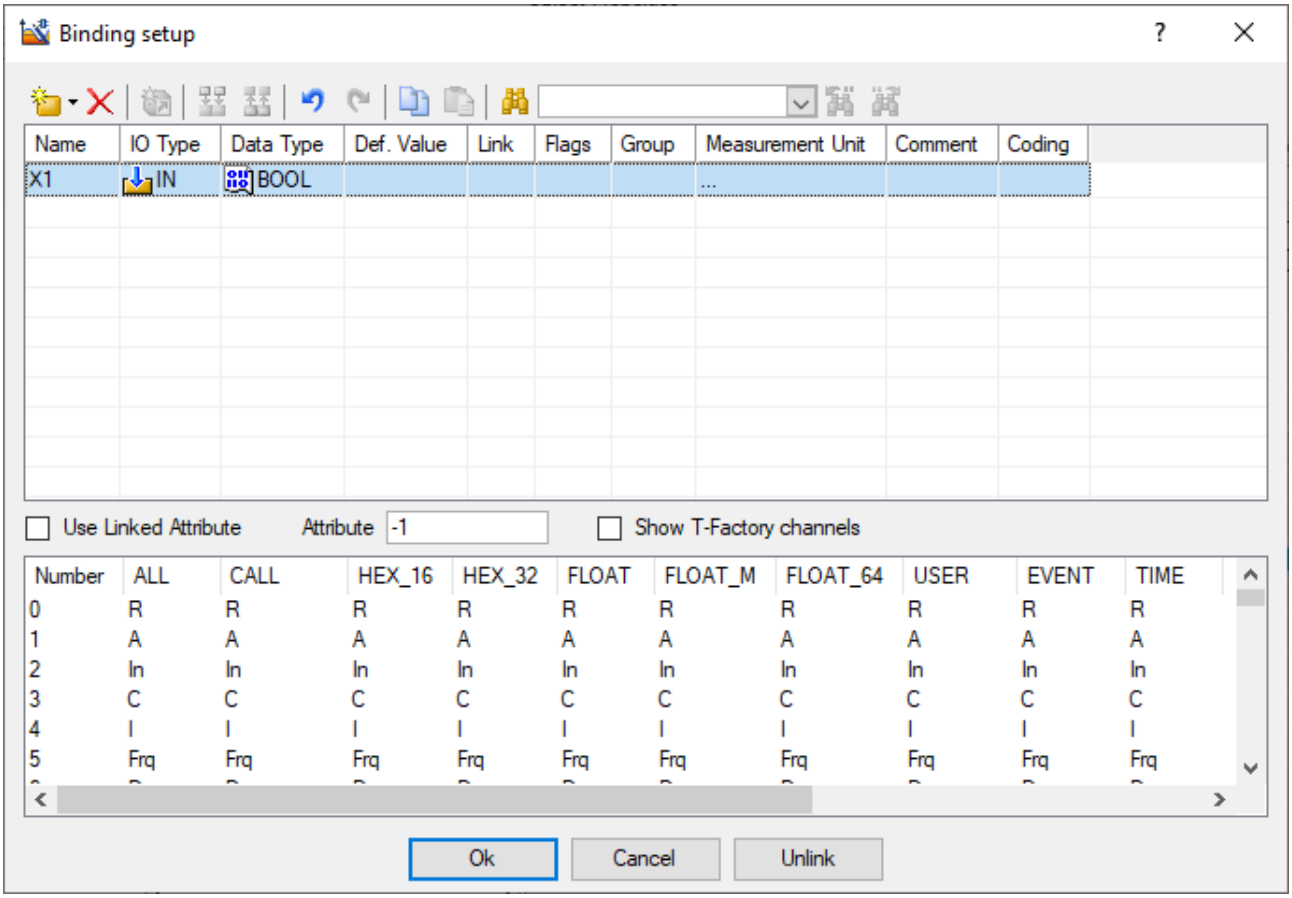


Рисунок 4 – Вікно прив'язки

Послідовно повторити аналогічні операції для інших чотирьох ГЕ **Text** (Текст), проте аргументу з іменем **Y** встановити у другому стовпці **IO Type** тип **OUT**. В результаті вікно **Binding setup** (Властивості прив'язки) повинно мати вигляд, показаний на рис. 5. У вікні властивостей всіх ГЕ **Текст** обов'язково задати бажані інші графічні параметри (колір, фон тощо).

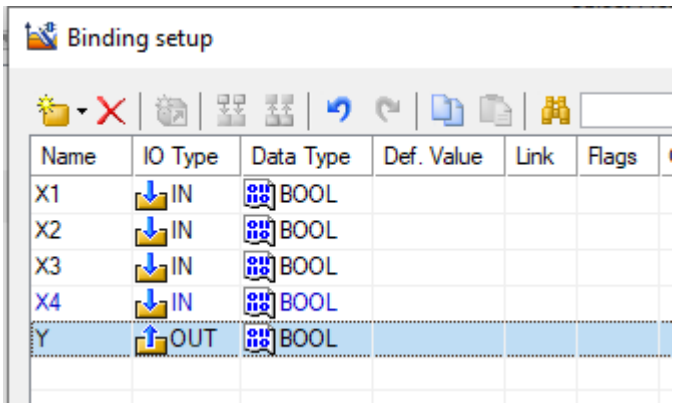


Рисунок 5 – Створення аргументів екрана

1.4 Ввести до складу графічного екрана засіб, що дозволяє реалізувати ввід з клавіатури числових значень вхідних змінних **X_1, X_2, X_3, X_4**. Вибрати послідовно на інструментальній панелі графічного редактора чотири ГЕ **Button** (Кнопка) та розмістити їх на екрані, як показано на рис. 6.

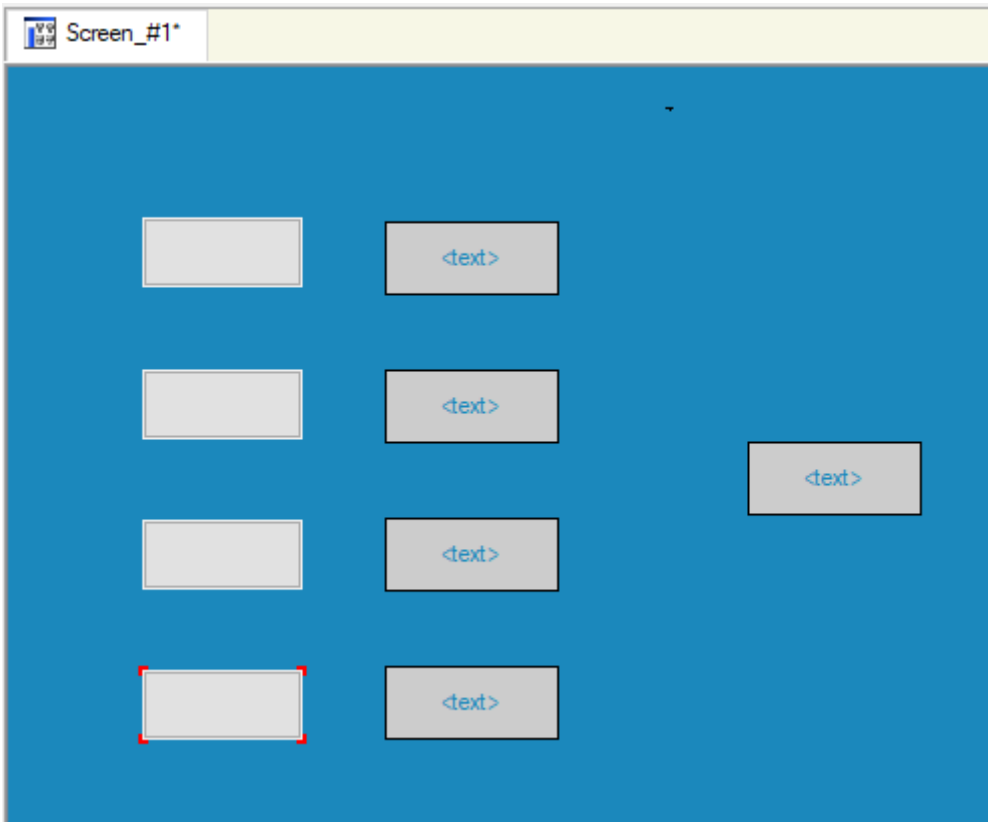


Рисунок 6 – Розташування на екрані ГЕ **Кнопка**

Перейти в режим редагування верхньої кнопки, викликавши вікно її властивостей, та ввести з клавіатури у рядок **Text** значення **X1** (рис. 7).

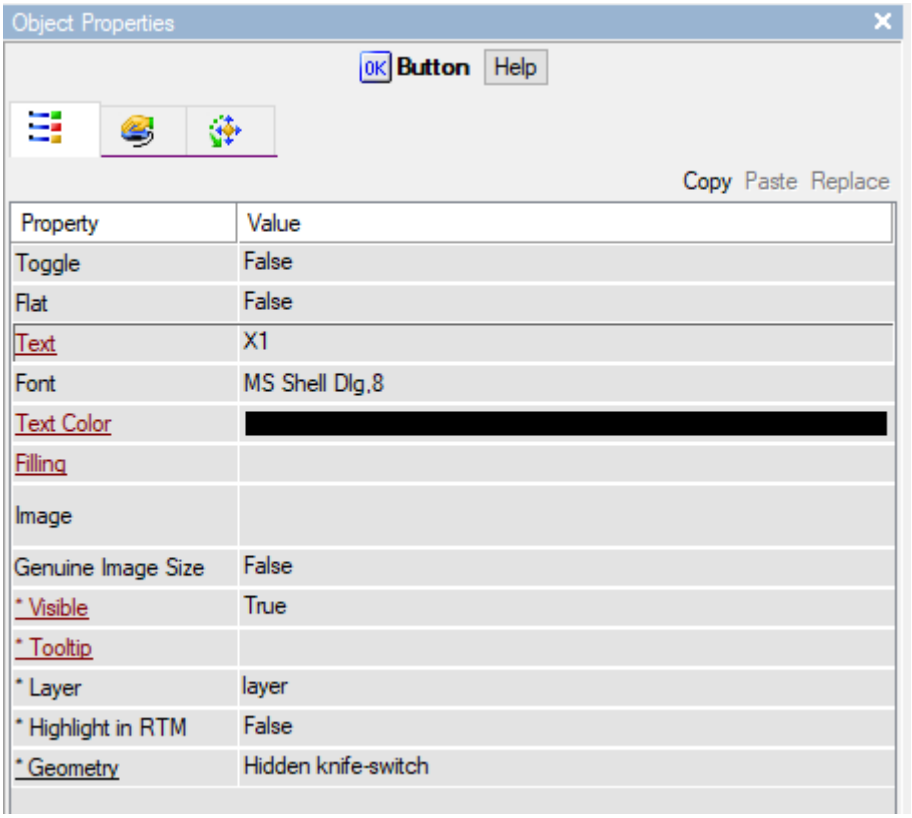


Рисунок 7 – Вікно властивостей ГЕ **Кнопка**

Відкрити у вікні властивостей ГЕ **Button** (Кнопка) середню закладку **Actions** (Дії), а потім ПК розкрити меню **MousePress** та вибрати зі списку команду **Send Value** (Передати значення) (рис. 8).

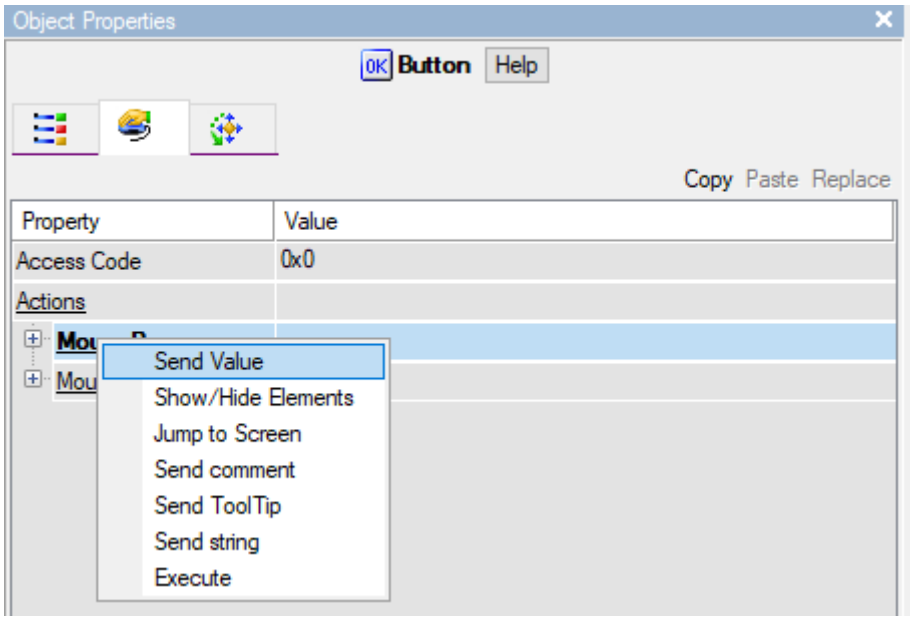


Рисунок 8 – Вікно закладки **Дії**

Праворуч у полі **Send Type** (Тип передачі) вибрати зі списку **Enter & Send** (Ввести і передати) (рис. 9), а потім ЛК праворуч у полі **Destination** (Результат) викликати табличний редактор аргументів і виконати прив'язку до аргументу **X1** (рис. 10).

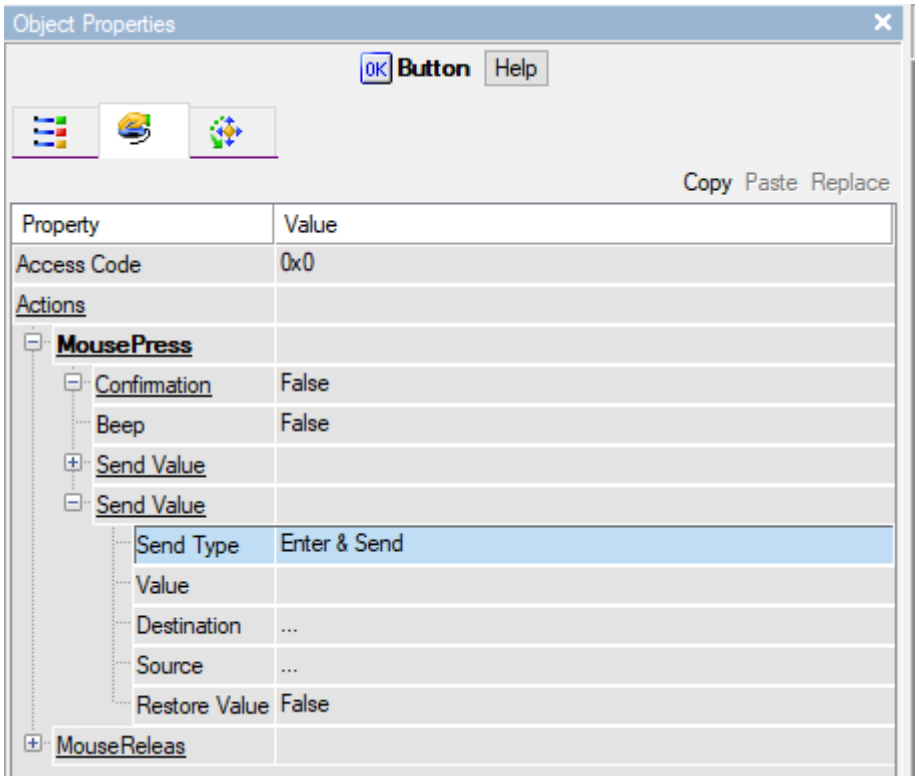


Рисунок 9 – Налаштування типу передачі

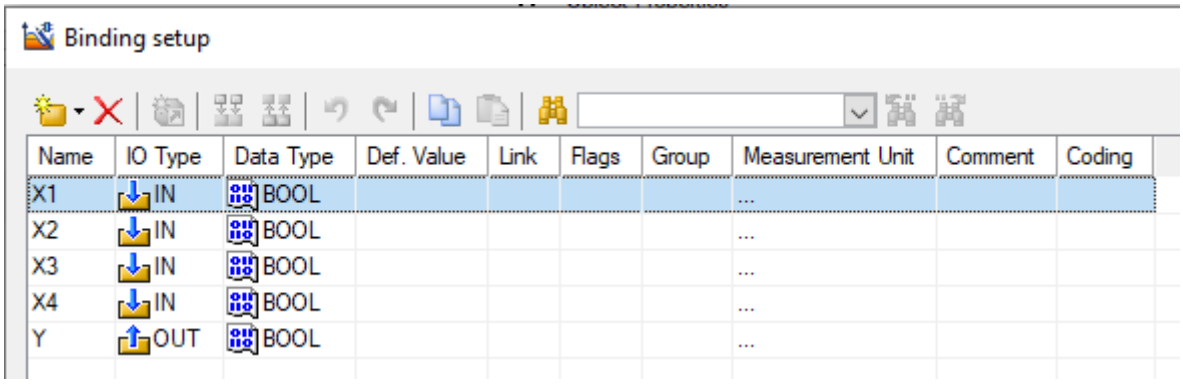


Рисунок 10 – Прив'язка до аргументу **X1**

Після прив'язки вікно властивостей ГЕ **Кнопка** виглядає, як показано на рис. 11.

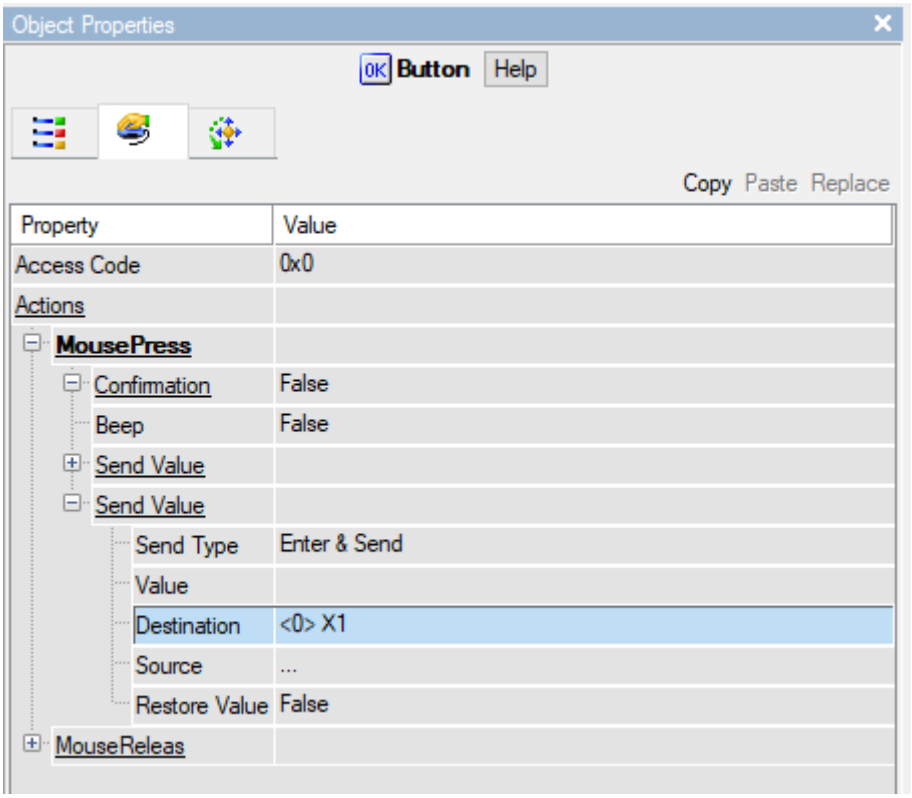


Рисунок 11 – Остаточний вигляд вікна властивостей ГЕ **Кнопка**

Аналогічно послідовно здійснити налаштування кнопок **X2, X3, X4.**

У вікні властивостей всіх ГЕ **Кнопка** обов’язково задати бажані інші графічні параметри (колір, фон тощо).

1.5 Прив'язати аргументи екрана до каналів. Створимо по аргументах входів **X1, X2, X3, X4** та виходу **Y** шаблону екрана нові канали й відредагуємо їхню прив'язку. У шарі **System** (Система) відкрити вузол **RTM_1** та за допомогою ПК на компоненті **Screen_#1:1** (Екран_#1:1) вибрати через контекстне меню пункт **Properties** (Властивості) (рис. 12).

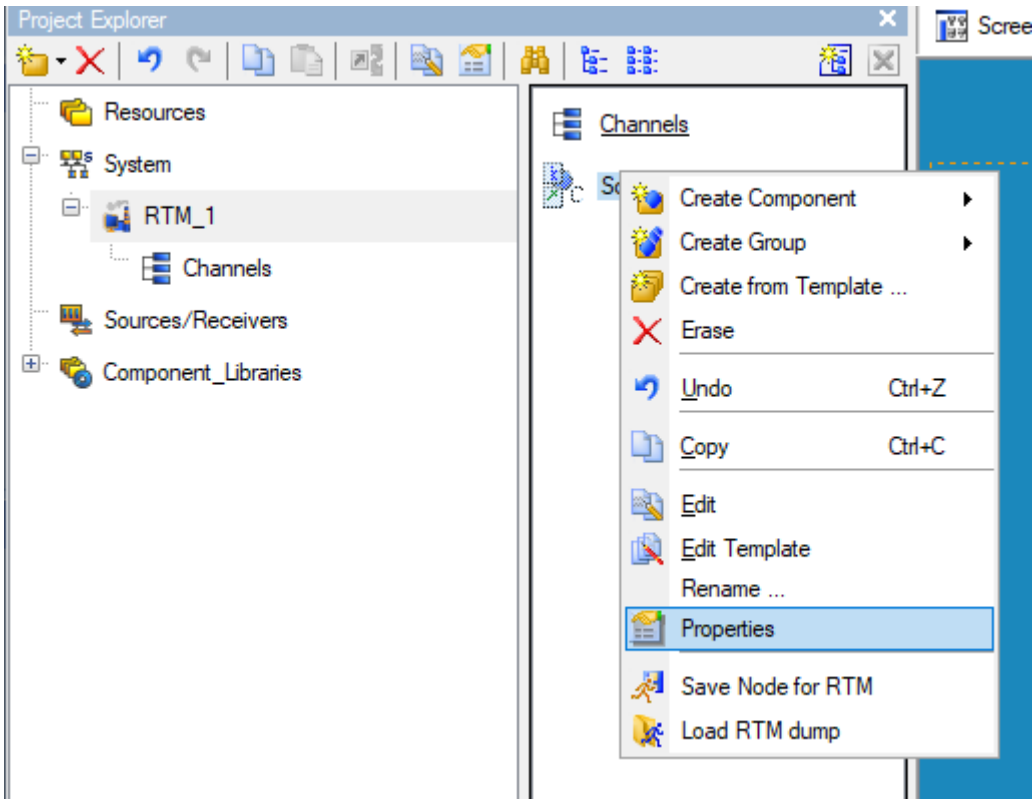


Рисунок 12 – Вибір меню **Властивості** екрана

У відкритому вікні вибрати вкладку **Arguments** (Аргументи) та ЛК при натиснутій на клавіатурі клавіші **Ctrl** виділити аргументи **X1, X2, X3, X4, Y** й за допомогою кнопки створити нові канали (рис. 13).

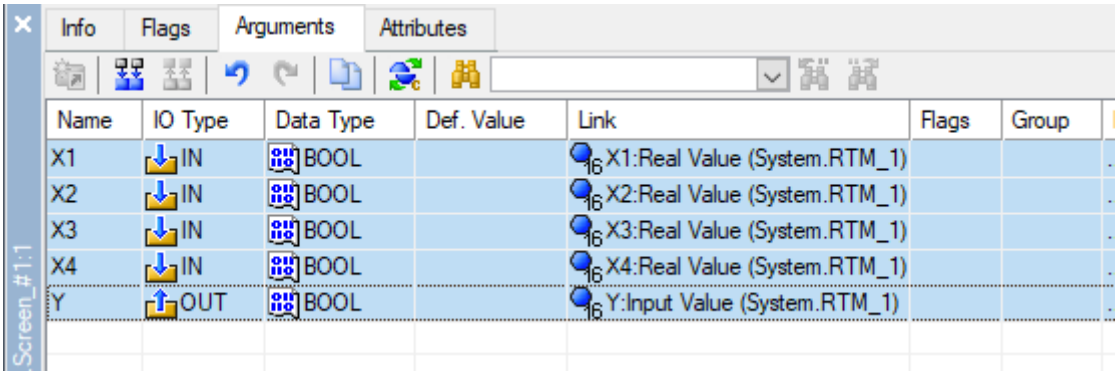


Рисунок 13 – Створення нових каналів

У результаті у вузлі **RTM_1** будуть автопобудовані канали **X1, X2, X3, X4, Y** (рис. 14).

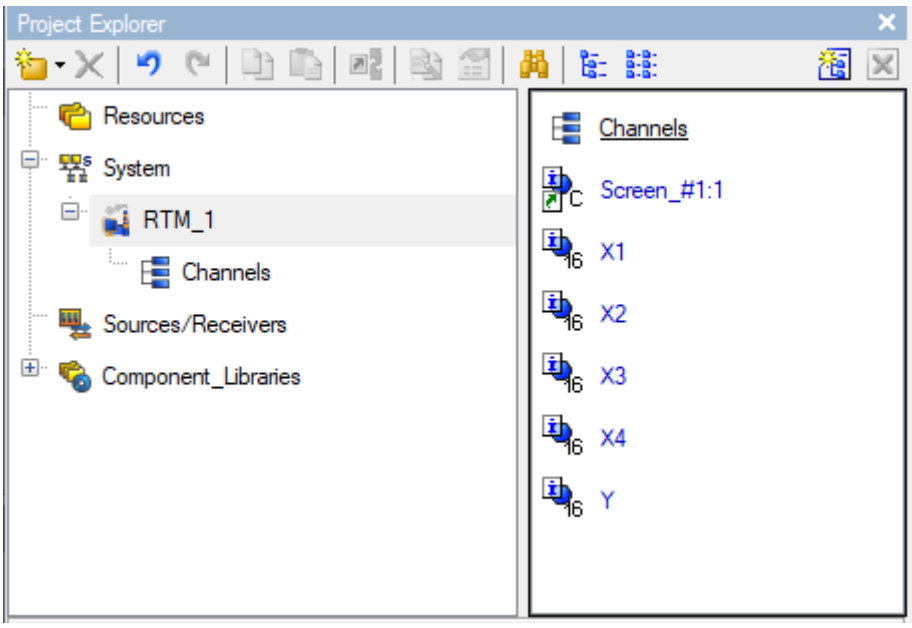


Рисунок 14 – Автопобудовані канали

1.6 Створити програму мовою **FBD.** Створимо програму, яка буде реалізовувати задану логічну функцію. Подвійним натисканням ЛК відкрити вузол **RTM_1** та створити в ньому компоненту **Program** (Програма) (рис. 15).

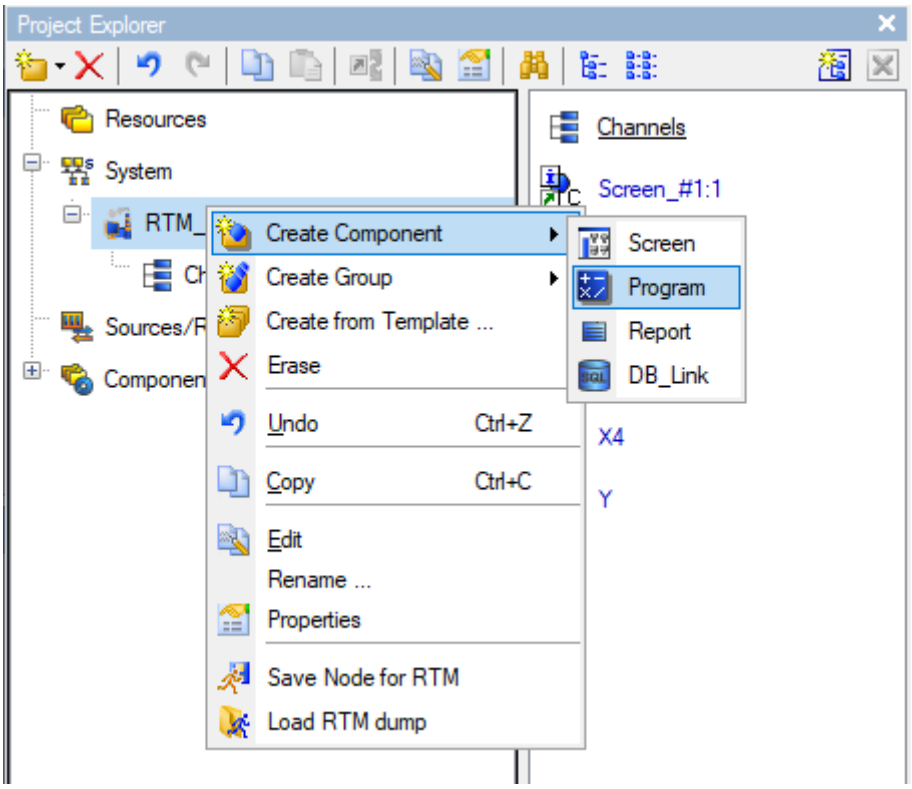


Рисунок 15 – Створення компоненти **Програма**

Виділити створену компоненту **Program#1:2** (Програма#1:2) та ПК викликати контекстне меню, вибравши в ньому ЛК пункт **Edit Template** (Редагувати шаблон) (рис. 16).

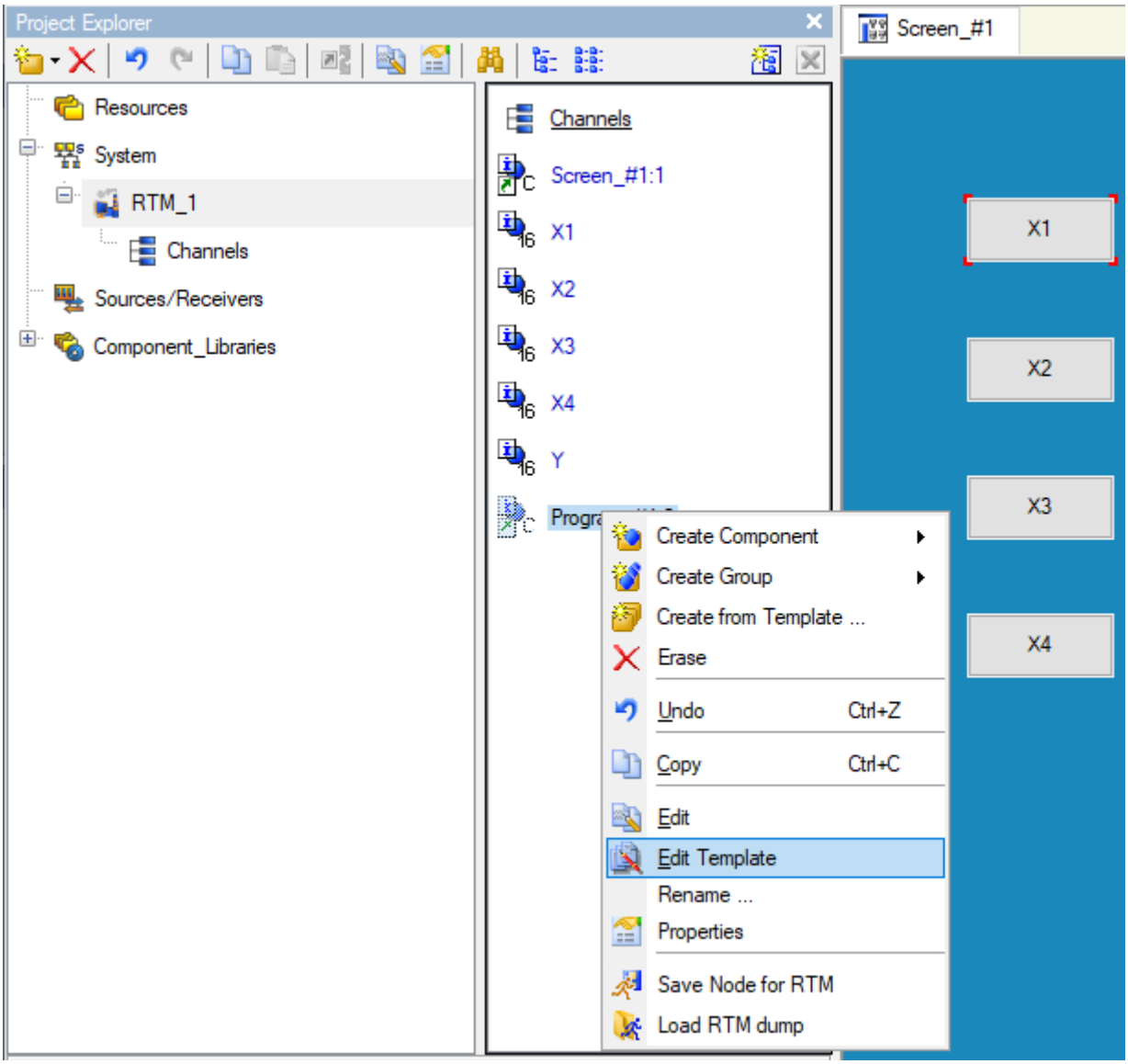


Рисунок 16 – Редагування аргументів **Програми**

Виділити ЛК у вікні **Script Tree** (Дерево шаблону) **Program#1** (Програма#1) рядок **Arguments (Аргументи),** викликавши табличний редактор аргументів. За допомогою кнопки послідовно створити чотири аргументи **X1, X2, X3, X4** та п’ятий **Y**. При цьому першим чотирьом аргументам встановити тип вводу/виводу **IN**, а останньому – **OUT**. Для всіх аргументів задати логічний тип даних **BOOL** (рис 17).

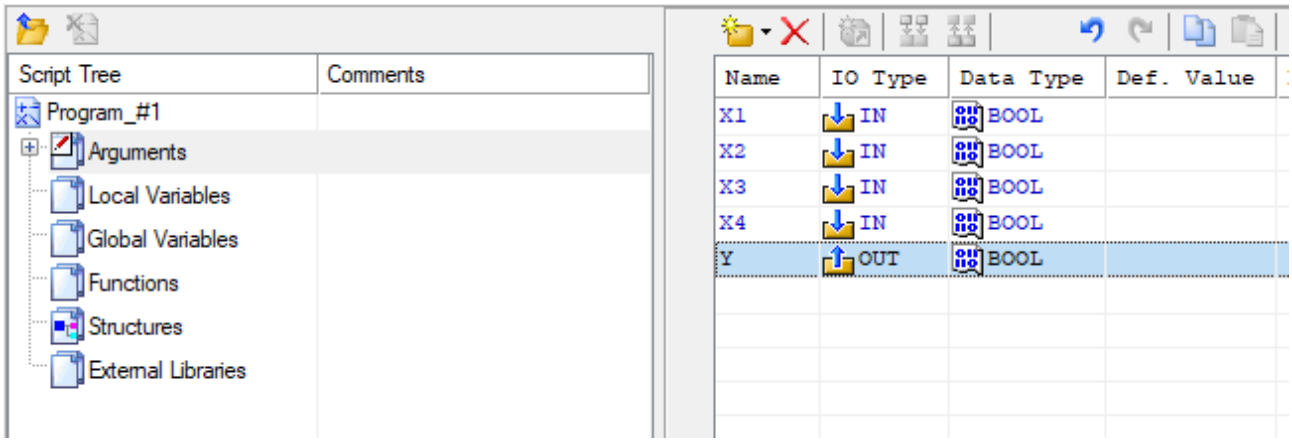


Рисунок 17 – Аргументи програми

Виділити ЛК в дереві шаблону **Script Tree** рядок **Program_#1** та у вікні, що відкрилося, вибрати мову **Function Block Diagram** (FBD-діаграма) (рис. 18).

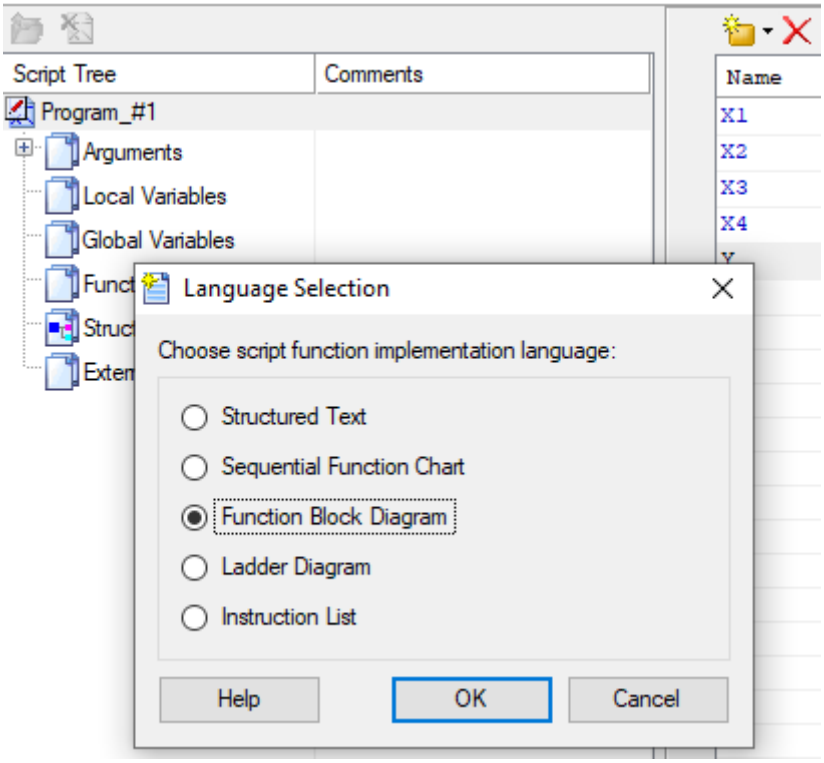


Рисунок 18 – Вибір мови програмування

Після натискання кнопки **Ok** (Прийняти) створимо програму праворуч у вікні редактора програм відповідно до варіанту завдання. Для цього викликати палітру FBD блоків шляхом натискання ЛК на палітрі вікна кнопки

Show/Hide FBD blocks palette (Показати/сховати палітру блоків FBD). Для зручності користування за допомогою ЛК вікно FBD блоків можна розмістити у бажаному місці на екрані та змінити його розміри. Вибираємо ЛК закладку **Logic** з логічними елементами (Рис. 19).

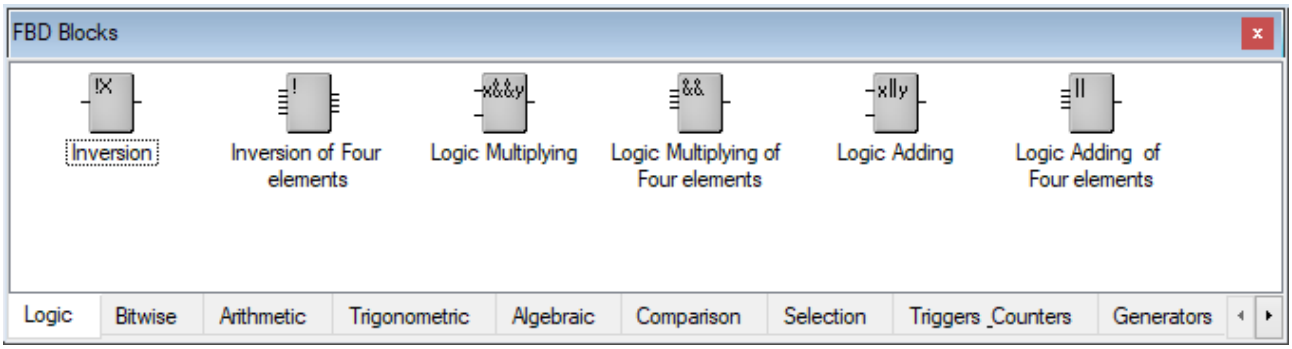


Рисунок 19 – Палітра FBD блоків

Аналізуємо задану логічну функцію $y = x_1 x_2 \overline{x}_4 + \overline{x}_1 \overline{x}_2 + \overline{x}_1 x_3 + \overline{x}_2 x_4$. Як видно, для її реалізації необхідно виконати операцію інверсії трьох змінних X_1, X_2, X_4, а також чотири операції логічного множення (одну для трьох змінних та три для двох змінних) та одну операцію логічного додавання чотирьох складових. Тому з палітри FBD блоків за допомогою натиснутої ЛК переміщуємо на поле програми три блоки **Inversion** (Інверсія), один блок **Logic Multiplying of Four elements** (Логічне множення чотирьох елементів), три блоки **Logic Multiplying** (Логічне множення**)** та один блок **Logic Adding of Four elements** (Логічне додавання чотирьох елементів) (рис. 20).

Розробка програми на основі FBD блоків полягає у з'єднанні їх входів/виходів, а також подача вхідних (X_1, X_2, X_3, X_4) та вихідного сигналу (Y) відповідно до операцій логічної функції. Треба мати на увазі, що верхній вхід FBD блоку не використовується, а інформаційними є входи, починаючи із другого. З'єднання входів здійснюється при натиснутій ЛК. Подача та вибір сигналу здійснюється після подвійного натискання ЛК на контакті блоку. Можна використовувати також кнопки панелі інструментів вгорі вікна (рис. 20) або контекстне меню після ПК на виділеному блоці.

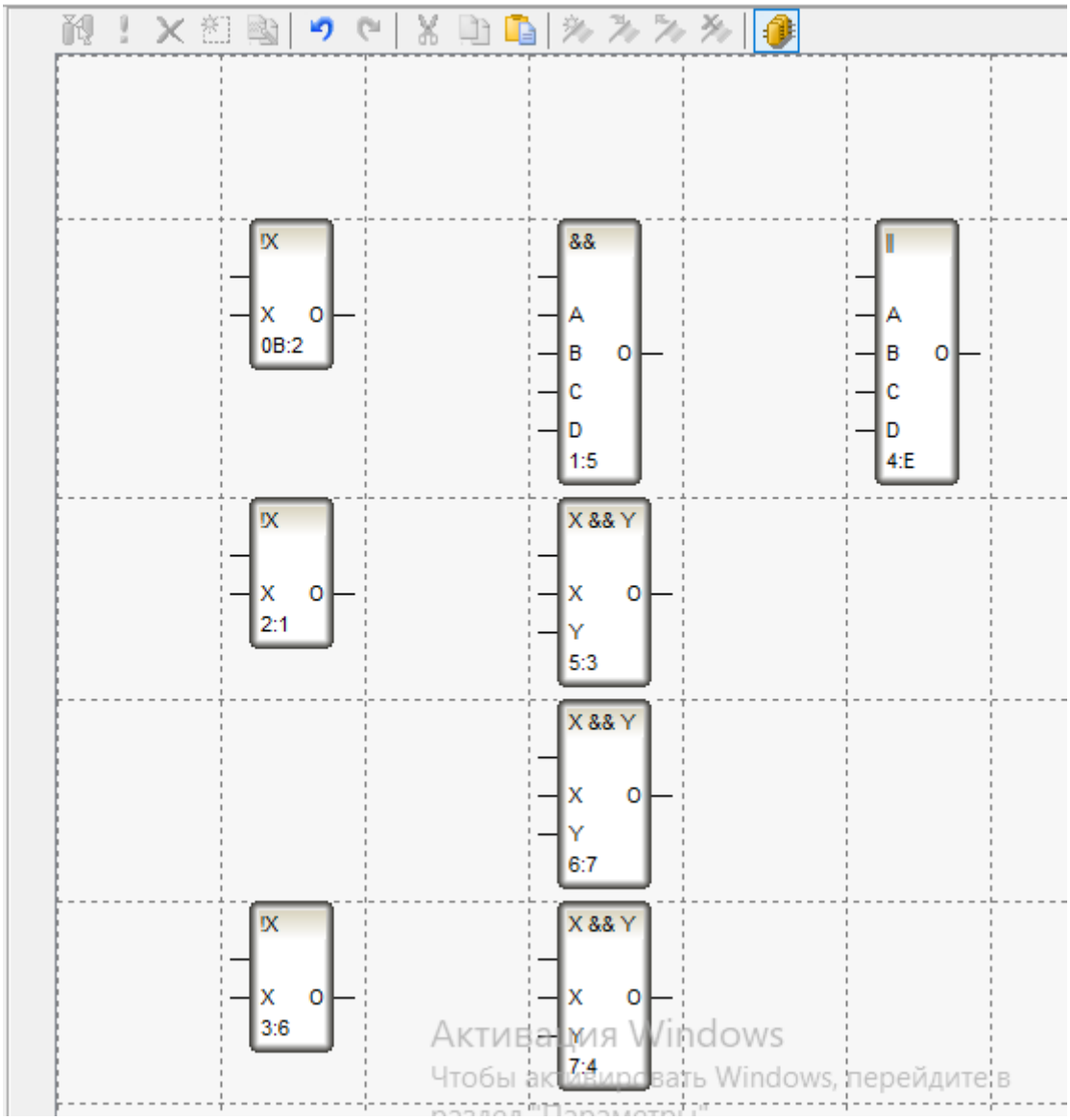


Рисунок 20 – Розташування логічних FBD блоків у вікні редактора програм

Після виконання всіх операцій для розглянутого прикладу програма буде мати вигляд, показаний на рис. 21. Важливо, всі входи FBD блоків повинні бути задіяні шляхом їх дублювання (наприклад, якщо кількість вхідних сигналів менше кількості входів), як для блоку **Логічне множення чотирьох елементів.**

За допомогою іконки **Build (F7)** (Компіляція) на інструментальній панелі редактора або гарячої клавіші F7 скомпілювати програму та переконатися в успішній компіляції у вікні **Messages**, яке при необхідності викликається з інструментальної панелі за допомогою іконки **Messages** (Повідомлення) (рис. 22).

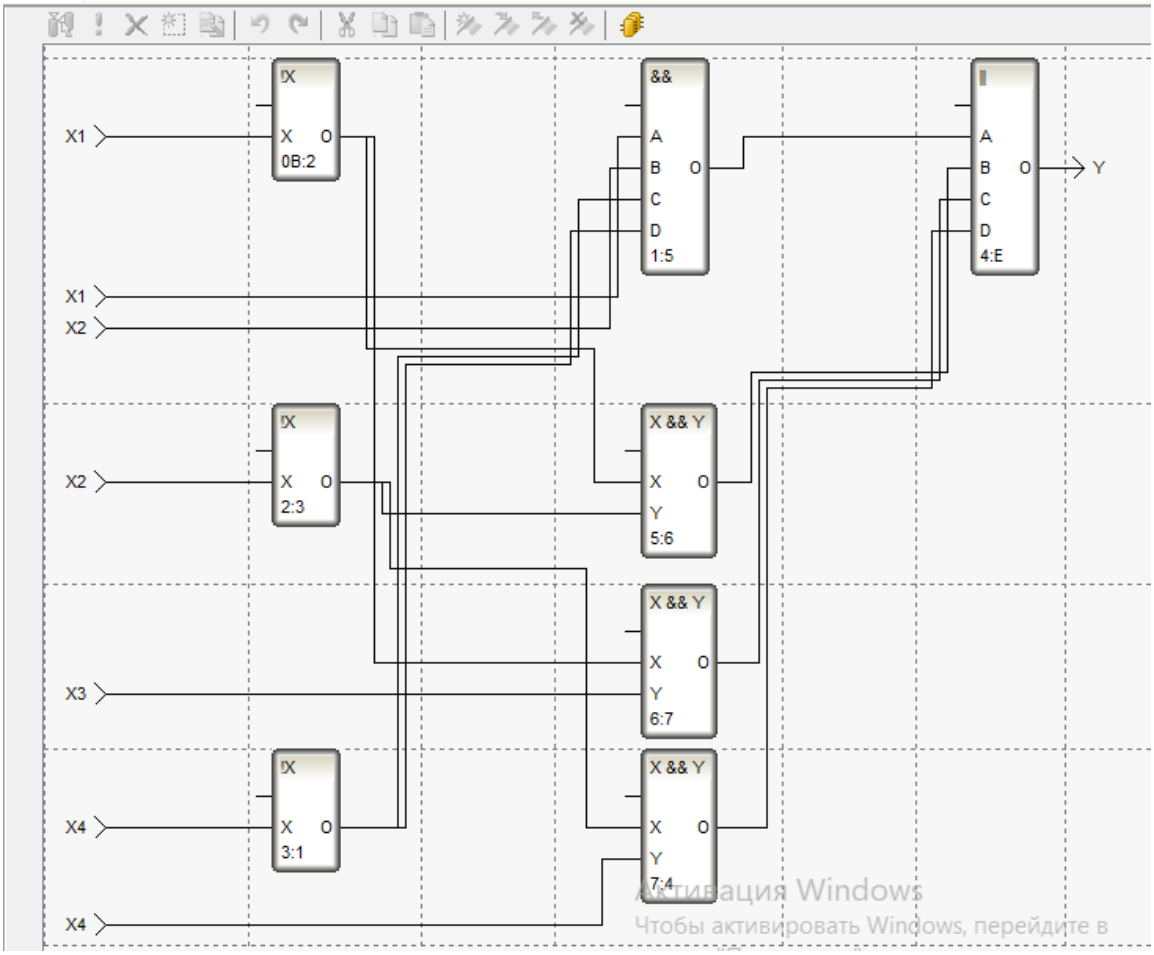


Рисунок 21 – Програма мовою **FBD**

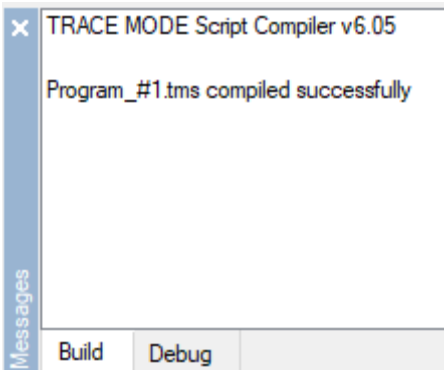


Рисунок 22 – Результат успішної компіляції програми

1.7 Прив'язка аргументів програми. Виконаємо прив'язку аргументів програми до атрибутів каналів. У вікні навігатора проекту **Project Explorer** ПК на компоненті **Program#1:2** викликати контекстне меню та вибрати **Properties** (Властивості). У вікні, що відкрилося, вибрати вкладку **Arguments** (Аргументи)**.** Подвійним натисканням ЛК у стовпці **Link** (Прив’язка) для рядка

X1 відкрити вікно конфігурування зв’язку. Вибрати у лівій частині вікна елемент **X1**, а потім у правій частині на вкладці **Attributes** (Атрибути) рядок **Real Value** (Реальне значення) та натиснути кнопку **Link** (Прив’язка) (рис. 23). Аналогічно прив'язати аргументи програми **X2, X3, X4** до реального значення каналів **X2, X3, X4**.

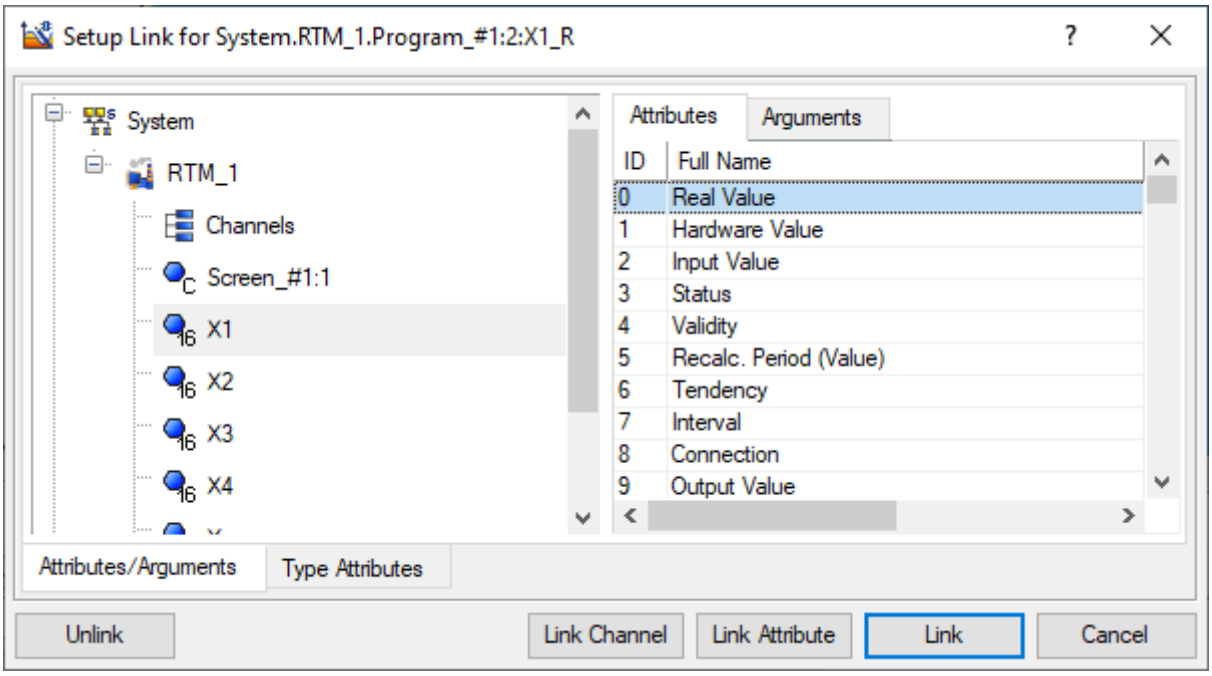


Рисунок 23 – Зв'язок аргументів програми з аргументами екрана

Подвійним натисканням ЛК у стовпці **Link** для аргументу програми **Y** викликати вікно настроювання прив'язки. Вибрати у лівому вікні канал **Screen_#1:1**, а в правому вікні замість вкладки **Attributes** вибрати вкладку **Arguments** та вказати аргумент **Y** й кнопкою **Link** підтвердити зв'язок (рис. 24). В результаті вікно властивостей компоненти **Program#1:2** буде мати вигляд, показаний на рис. 25.

1.8 Вказати у проекті свої дані.

На графічному екрані розмістити у зручному місці шифр групи та своє ПІБ аналогічно до роботи № 1.

На графічному екрані вказати свій варіант завдання та розмістити задану логічну функцію, використовуючи необхідні ГЕ з інструментальної панелі.

В меню **File\Information...**(Файл\Інформація...) в рядку **Owner** (Власник) написати за допомогою клавіатури своє ПІБ, а в рядку **Organization** (Організація) – шифр групи та факультету.

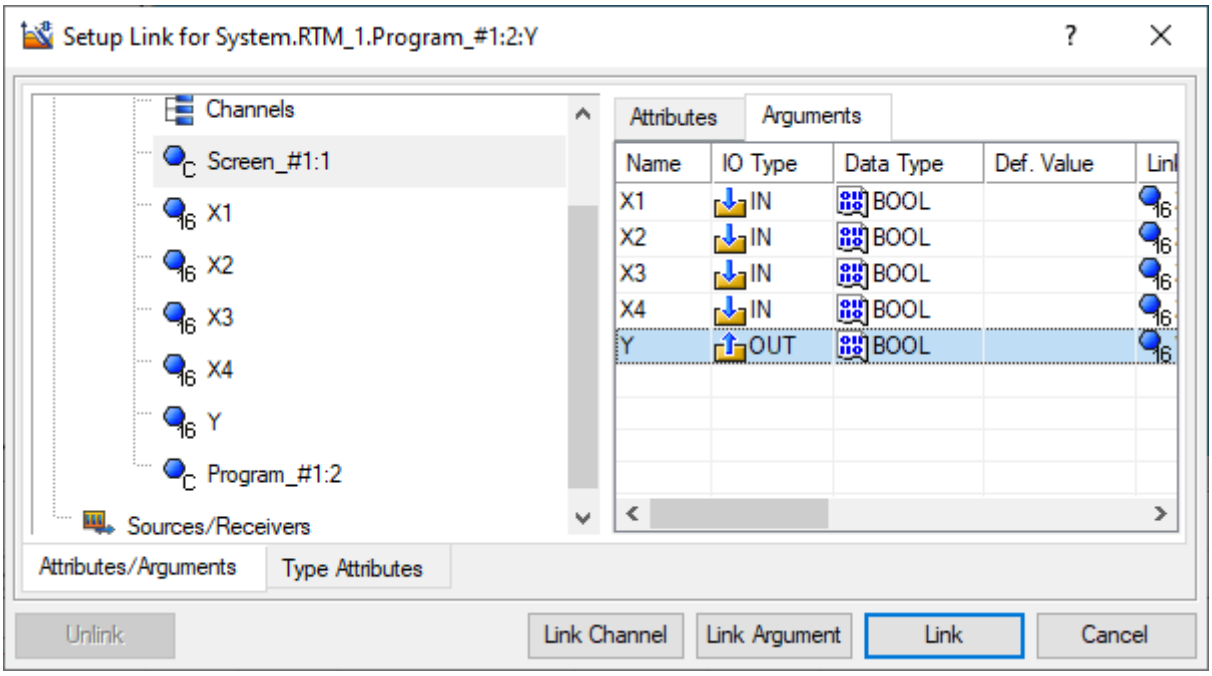


Рисунок 24 – Зв'язок вихідних аргументів

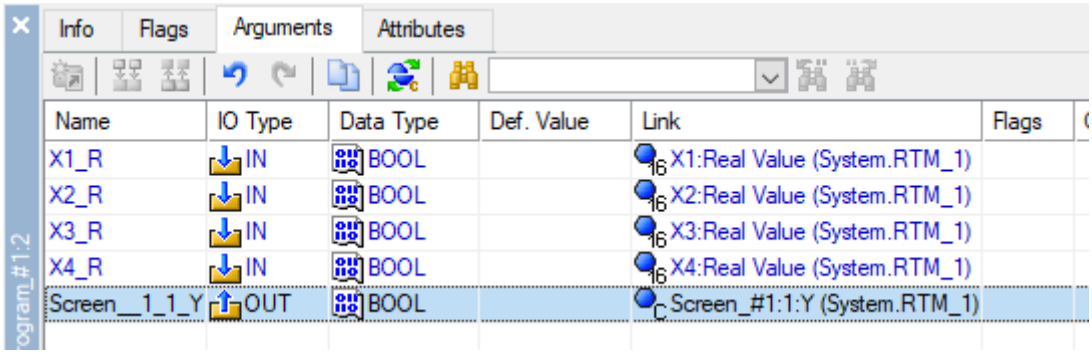


Рисунок 25 – Остаточний вигляд вікна настроювання зв'язку

1.9 Запустити проект в роботу.

Зберегти проект натисканням кнопки . На головній інструментальній панелі ICP натиснути ЛК кнопку **Save for RTM** (Зберегти для МРЧ),

скомпілювавши проект для запуску в реальному часі. Виділити ЛК в навігаторі проекту вузол **RTM_1**, а потім натиснути на головній панелі кнопку **Start Profiler** (Запустить профайлер). У вікні, що відкрилося, запустити проект на виконання шляхом натискання ЛК кнопки **Run** (Пуск/Зупинка).

За допомогою кнопок ввести необхідні комбінації значень **0** або **1** логічних змінних X_1, X_2, X_3, X_4 та за значенням логічної функції **Y** перевірити правильність її реалізації (рис. 26).

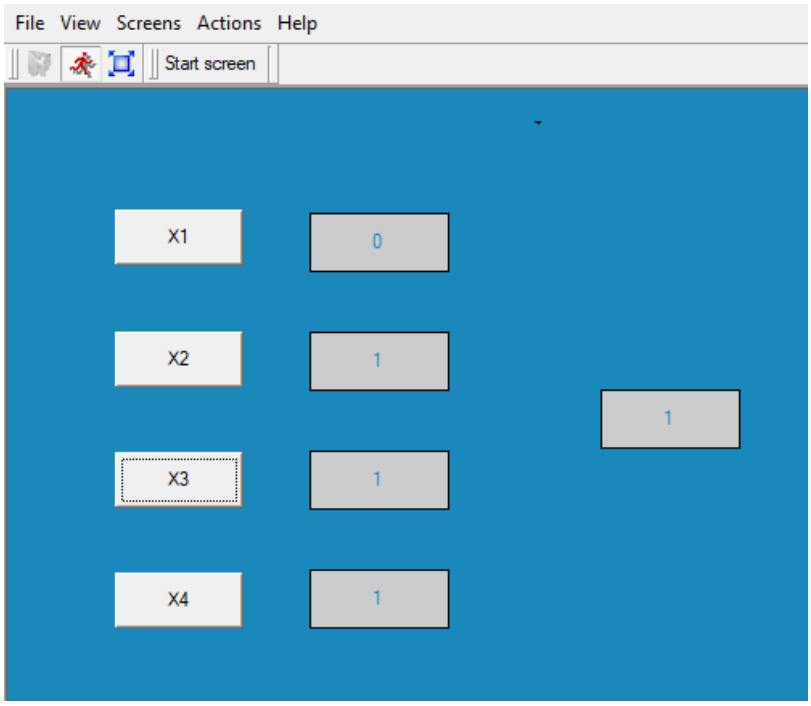


Рисунок 26 – Результат імітаційного запуску проекту

Зупинити виконання проекту шляхом натискання на панелі інструментів ЛК кнопки **Run** (Пуск/Зупинка). Закрити вікно графічного екрану через меню **File\Exit** (Файл\Вихід) або засобами Windows. Зберегти проект натисканням кнопки та завершити виконання роботи.

1.10 Оформити протокол комп'ютерного практикуму відповідно до вимог, наведених у вступі.

2 ЗАВДАННЯ ДО КОМП'ЮТЕРНОГО ПРАКТИКУМУ № 3

Номер варіанту завдання відповідає порядковому номеру студента у списку навчальної групи:

1. $y = x_1x_2 + x_1\overline{x}_3\overline{x}_4 + \overline{x}_1\overline{x}_2\overline{x}_3 + \overline{x}_1x_4$
2. $y = x_1\overline{x}_4 + x_1\overline{x}_2\overline{x}_3 + \overline{x}_1x_2 + \overline{x}_1x_4$
3. $y = x_1\overline{x}_2\overline{x}_3 + \overline{x}_1\overline{x}_2x_3 + x_4$
4. $y = \overline{x}_1\overline{x}_3 + \overline{x}_1x_4 + x_2\overline{x}_3 + \overline{x}_2x_4$
5. $y = x_1\overline{x}_3 + \overline{x}_1\overline{x}_2 + \overline{x}_1\overline{x}_4 + \overline{x}_2\overline{x}_4$
6. $y = x_1x_4 + x_1x_2x_3 + \overline{x}_1\overline{x}_2\overline{x}_4 + \overline{x}_3x_4 + \overline{x}_2\overline{x}_3$
7. $y = x_1x_2 + x_1\overline{x}_3 + \overline{x}_1\overline{x}_2x_3 + x_2\overline{x}_4 + x_3\overline{x}_4$
8. $y = \overline{x}_1x_3 + \overline{x}_1x_4 + x_2x_3 + \overline{x}_2\overline{x}_3x_4$
9. $y = x_1x_2x_3 + x_1x_3\overline{x}_4 + x_1x_2\overline{x}_4 + \overline{x}_2\overline{x}_3x_4 + \overline{x}_1x_4 + \overline{x}_1\overline{x}_2\overline{x}_3$
10. $y = x_1\overline{x}_2\overline{x}_3 + \overline{x}_1x_2 + \overline{x}_1x_3 + \overline{x}_1x_4 + x_2x_3x_4$
11. $y = x_1x_2x_4 + x_1\overline{x}_2\overline{x}_3 + \overline{x}_1x_3 + x_2x_3 + \overline{x}_1\overline{x}_2x_4 + x_3x_4$
12. $y = x_1x_3 + x_1x_4 + \overline{x}_1x_2\overline{x}_3 + \overline{x}_1\overline{x}_3\overline{x}_4 + \overline{x}_2x_3\overline{x}_4$
13. $y = x_1x_3 + x_1x_2x_4 + \overline{x}_1\overline{x}_2x_3 + \overline{x}_1\overline{x}_4$
14. $y = x_1\overline{x}_2x_3 + \overline{x}_1x_2x_3 + x_1x_2\overline{x}_3 + \overline{x}_1\overline{x}_2\overline{x}_3 + x_3x_4 + \overline{x}_3\overline{x}_4$
15. $y = \overline{x}_1\overline{x}_3 + x_1x_3x_4 + \overline{x}_1x_2 + \overline{x}_2\overline{x}_4 + \overline{x}_3\overline{x}_4$
16. $y = x_1x_2\overline{x}_3x_4 + x_1x_3\overline{x}_4 + x_1\overline{x}_2x_3 + \overline{x}_1\overline{x}_2\overline{x}_3 + \overline{x}_1\overline{x}_3\overline{x}_4 + \overline{x}_1x_3x_4 + \overline{x}_2\overline{x}_3\overline{x}_4$
17. $y = x_1x_3 + \overline{x}_1x_2\overline{x}_4 + \overline{x}_1\overline{x}_2x_4 + x_2x_3 + x_3x_4$
18. $y = \overline{x}_1x_3 + \overline{x}_1\overline{x}_2 + \overline{x}_1x_4 + \overline{x}_2\overline{x}_4 + x_2\overline{x}_3x_4$
19. $y = x_1x_3 + \overline{x}_1\overline{x}_2\overline{x}_3 + x_2\overline{x}_3\overline{x}_4$
20. $y = x_1x_2x_4 + x_1x_3x_4 + \overline{x}_1\overline{x}_4 + \overline{x}_1\overline{x}_2\overline{x}_3 + x_2x_3\overline{x}_4$
21. $y = x_1x_3 + \overline{x}_1\overline{x}_2\overline{x}_3 + \overline{x}_1\overline{x}_3\overline{x}_4 + \overline{x}_2\overline{x}_4$
22. $y = x_1x_2x_4 + x_1x_3x_4 + \overline{x}_1\overline{x}_2\overline{x}_3 + \overline{x}_1\overline{x}_2\overline{x}_4 + x_2x_3x_4 + \overline{x}_2\overline{x}_3\overline{x}_4$
23. $y = x_1x_2x_3 + \overline{x}_1\overline{x}_2 + \overline{x}_1\overline{x}_4 + \overline{x}_2\overline{x}_3x_4 + x_3\overline{x}_4$
24. $y = x_1\overline{x}_2x_3 + x_1x_3\overline{x}_4 + \overline{x}_1x_4 + x_2\overline{x}_3$
25. $y = x_1x_2\overline{x}_4 + \overline{x}_1\overline{x}_2 + \overline{x}_1x_3 + \overline{x}_2x_4$

3 ПИТАННЯ ДЛЯ ЗАХИСТУ КОМП'ЮТЕРНОГО ПРАКТИКУМУ № 3

1. Етапи створення проектів систем автоматизації в SCADA.
2. Програмування мовою FBD логічних функцій.
3. Палітра логічних FBD блоків.
4. Компіляція програми та перегляд інформації про результати компіляції.
5. Завдання властивостей ГЕ «Кнопка».
6. Завдання властивостей ГЕ «Текст».
7. Створення аргументів екрана та завдання їхніх параметрів.
8. Автопобудова каналів.
9. Прив'язка аргументів екрана до каналів.

Комп’ютерний практикум № 4

СТВОРЕННЯ ПРОЕКТУ ОДНОКОНТУРНОЇ СИСТЕМИ АВТОМАТИЧНОГО РЕГУЛЮВАННЯ

Мета роботи – освоїти методику створення проекту одноконтурної системи автоматичного регулювання за допомогою SCADA-системи.

1 ПРОГРАМА РОБОТИ

1.1 Реалізувати одноконтурну систему автоматичного регулювання (САР) в SCADA-системі. Функціональна схема САР показана на рис. 1. Вона містить об’єкт керування, ПІД-регулятор, датчик регульованої змінної, а також суматор, що обчислює похибку $e(t)= x(t)-y(t)$ між заданим $x(t)$ та дійсним $y(t)$ значенням регульованої змінної.

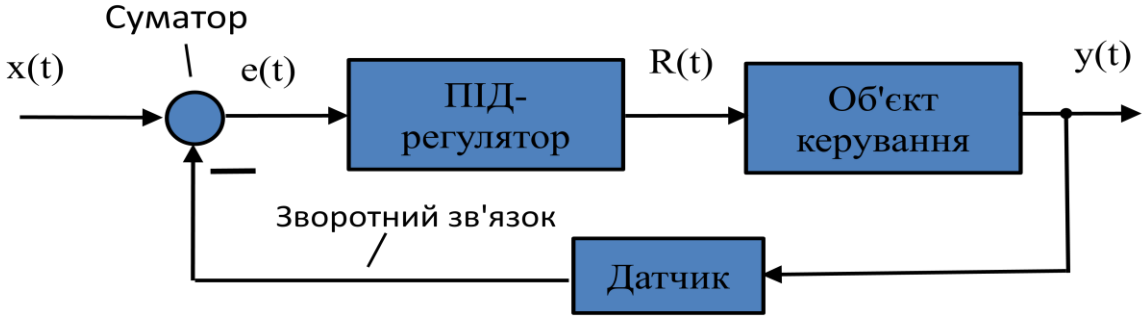


Рисунок 1 – Функціональна схема САР

Властивості об’єкту керування описуються аперіодичною ланкою першого порядку із запізненням, передавальна функція якої має вигляд

$$W_{ob}(s)=\frac{y(s)}{R(s)}=\frac{K_{ob}}{T_{ob}s+1}e^{-\tau s}$$

де: K_{ob} – коефіцієнт передавання об’єкту;

T_{ob} – стала часу об’єкту;

τ – час запізнювання;

$s=d../dt$ – оператор Лапласа.

Вихідний сигнал ПІД-регулятора складається з пропорційної, інтегральної та диференційної частини

$$R(t) = K_r \cdot e(t) + T_i \int e(t)dt + T_d \frac{d\,e(t)}{dt}$$

Параметри ПІД-регулятора, які забезпечують 20 % перерегулювання перехідної функції регульованої змінної, обчислюються за наступними залежностями:

- коефіцієнт підсилення

$$K_r = \frac{1,2 \cdot T_{ob}}{K_{ob} \cdot \tau}$$

- коефіцієнт інтегрування

$$T_i = \frac{K_r}{2 \cdot \tau}$$

- коефіцієнт диференціювання

$$T_d = 0,4 \cdot \tau \cdot K_r$$

Розрахувати параметри САР відповідно до варіанту завдання, наведеного у розділі 2 даної роботи.

1.2 Створити новий проект стилю **Easy** (Простий), як у роботі № 1. Подвійним натисканням ЛК на компоненті **Screen_#1:1** (Екран_#1:1) відкрити вікно графічного редактора. Зберегти проект шляхом вибору в меню **File** (Файл) пункту **Save** (Зберегти) або натисканням кнопки на інструментальній панелі, задавши з клавіатури бажане ім’я, наприклад **Robota_4**.

1.3 Розмістити на екрані, як показано на рис. 2, сім ГЕ **Button** (Кнопка) для вводу параметрів САР та значення сигналу завдання:

1) Аперіодична ланка першого порядку:

– коефіцієнт передавання об'єкту (Kob): **К об’єкту**;

– стала часу об'єкту (Tob): **Т об’єкту**;

– час запізнювання (Tau): **Час запізнювання**,

2) Величина сигналу завдання (Zav): **ЗАВДАННЯ**,

3) ПІД-регулятор:

– коефіцієнт підсилення регулятора (Kr): **К регулятора**;

– коефіцієнт інтегрування (Ti): **І регулятора**;

– коефіцієнт диференціювання (Td): **D регулятора**.

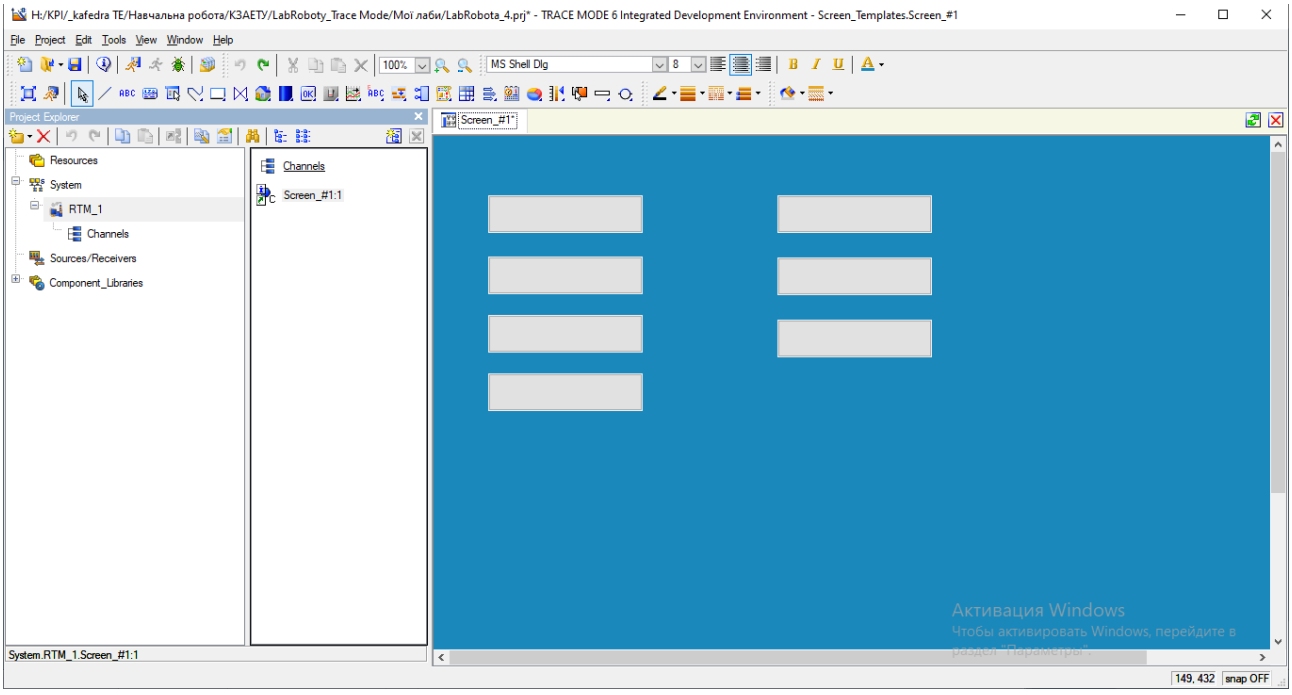


Рисунок 2 – Розміщення семи ГЕ **Кнопка** на графічному екрані

Подвійним натисканням ЛК на верхньому лівому ГЕ **Button** (Кнопка) відкрити вікно його властивостей. У рядок праворуч від властивості **Text** ввести з клавіатури напис **К об’єкту** (рис. 3).

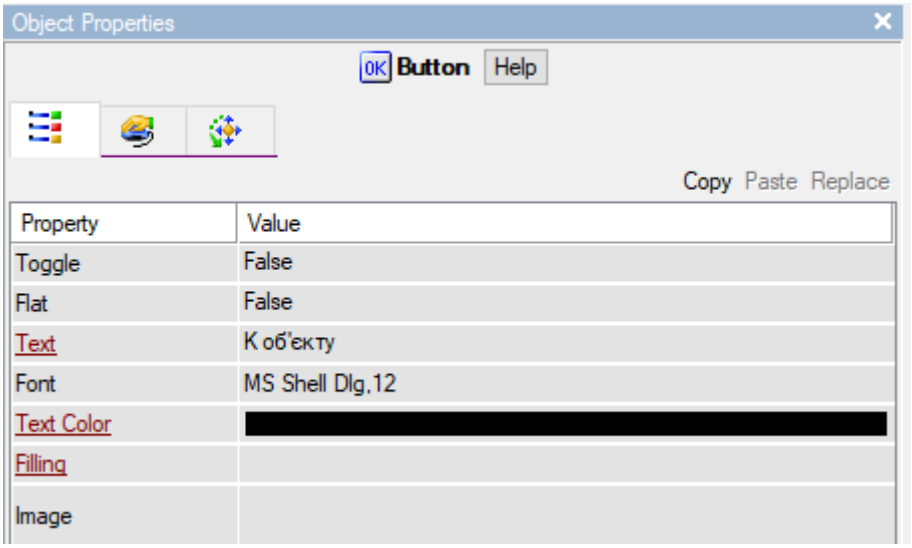


Рисунок 3 – Вікно властивостей графічного елемента **Кнопка**

Відкрити середню закладку **Actions** (Дії) у вікні властивостей **Кнопка**, а потім ПК розкрити меню **MousePress** та вибрати зі списку команду **Send Value** (Передати значення) (рис. 4).

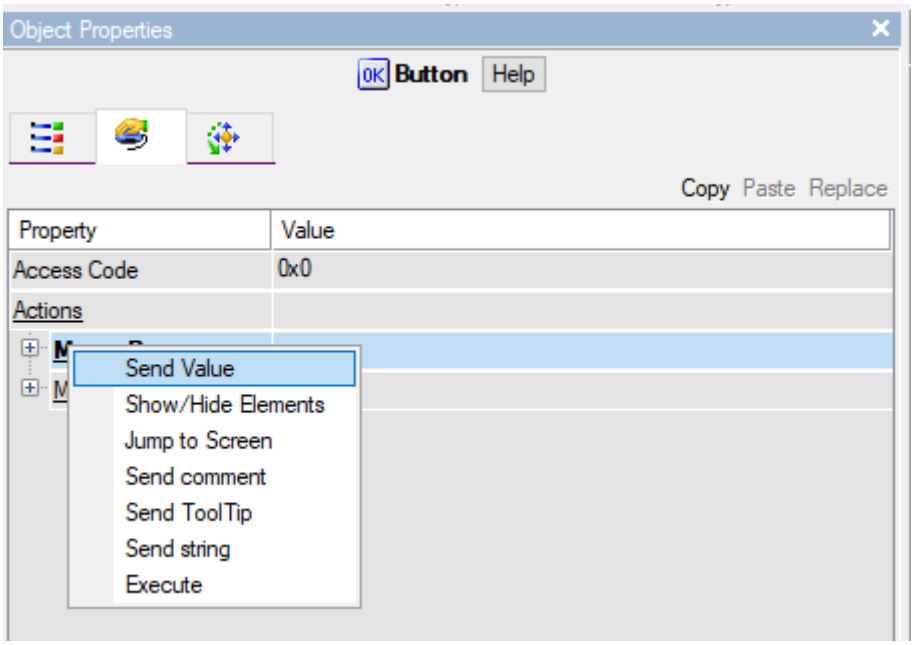


Рисунок 4 – Вікно закладки **Дії**

У полі **Send Type** (Тип передачі) вибрати зі списку **Enter & Send** (Ввести і передати) (рис. 5).

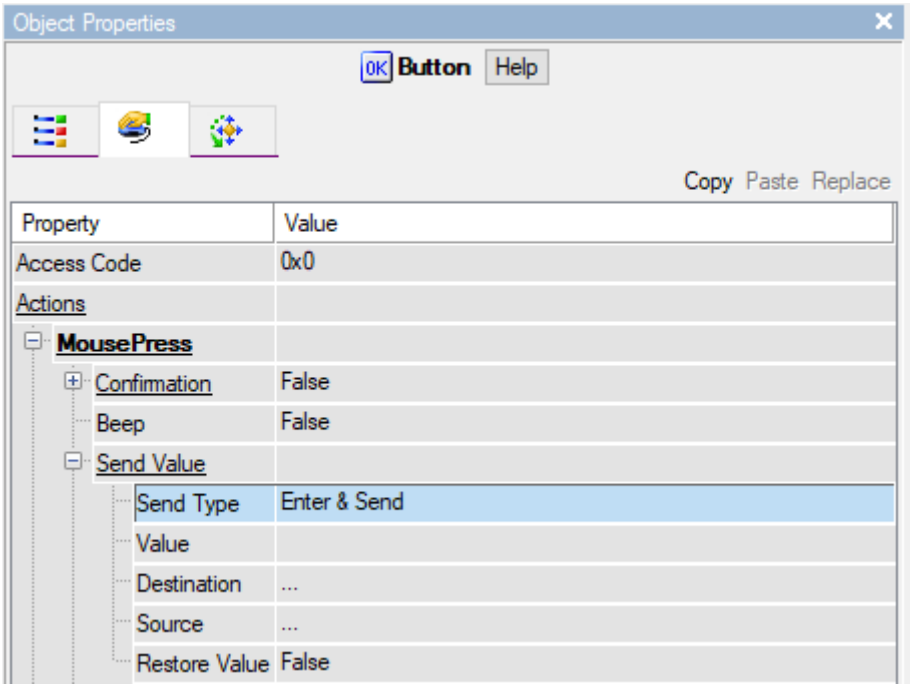


Рисунок 5 – Настроювання графічного елемента **Кнопка**

ЛК у полі **Destination** (Результат) викликати табличний редактор аргументів. Створити зразу вісім нових аргументів, а потім латиницею задати їм нові ім'я **Kob, Tob, Tau, Zav, Кr, Ti, Td** типу **IN** та **Vykhid** типу **OUT** (рис. 6).

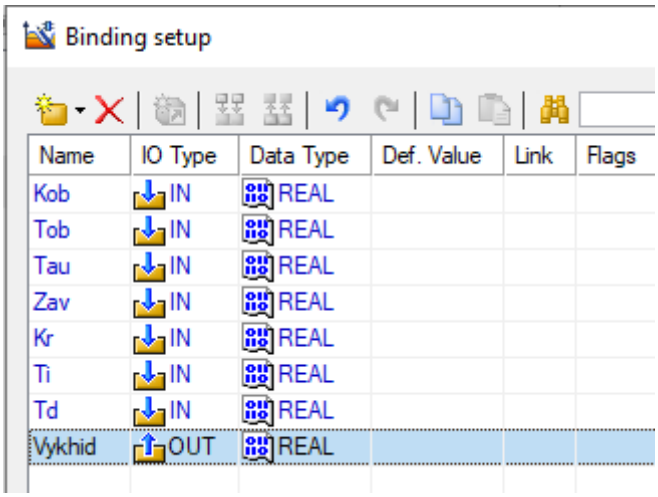


Рисунок 6 – Створення аргументів екрана

Вибрати у вікні ЛК аргумент **Kob** та кнопкою **Ok** (Готово) підтвердити прив'язку атрибута ГЕ до цього аргументу.

Аналогічно налаштувати кнопки для аргументів **Tob, Tau, Zav, Кг, Ti, Td**, задавши їм у полі **Text** (Текст) відповідні імена та здійснивши прив’язку. Після виконання прив'язок екран повинен мати вигляд, як показано на рис. 7.

У вікні властивостей всіх ГЕ **Кнопка** обов’язково задати бажані графічні параметри, а також вирівняти їх на екрані (по вертикалі, по горизонталі, однакові проміжки тощо), використовуючи меню **Tools** на інструментальній панелі.

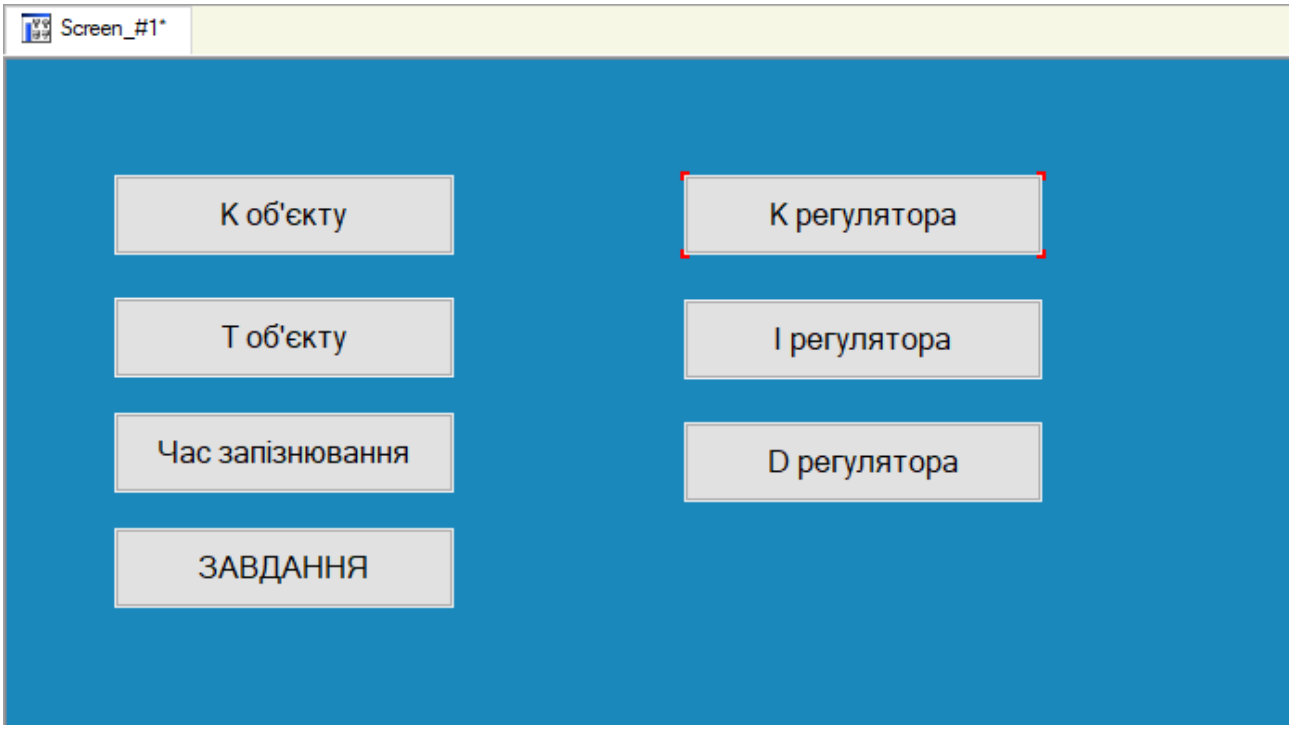


Рисунок 7 – Остаточний вигляд графічного екрану

1.4 Розмістити на екрані праворуч існуючих елементів **Кнопка** сім ГЕ **Text** (Текст) для відображення введених значень параметрів САР та сигналу завдання, як показано на рис. 8.

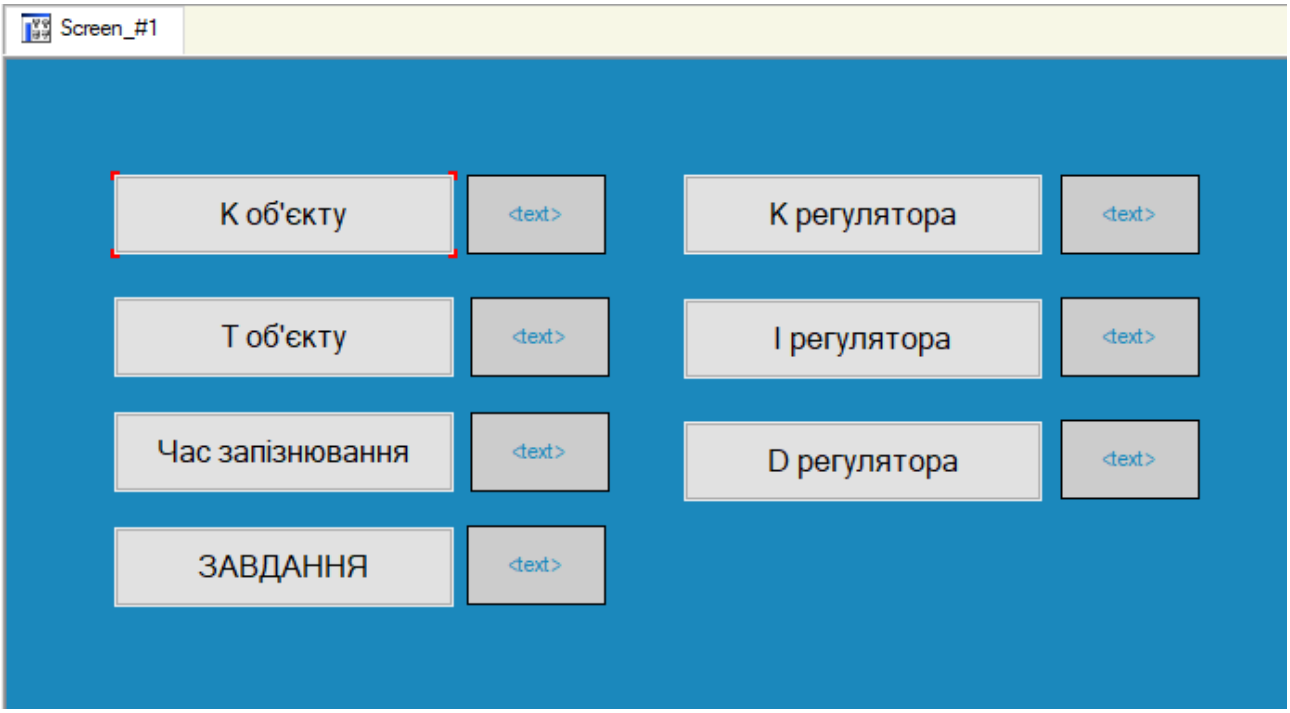


Рисунок 8 – Розташування ГЕ **Текст** на екрані

На вкладці основних властивостей **Generals Properties** елементу **Text** (Текст), який розташований праворуч кнопки **К об'єкту**, подвійним натисканням ЛК на рядку **Text** викликається меню **Indication type** (Вид індикації). У правому полі відкритого рядка натискається ЛК та із списку доступних типів вибирається **Value** (Значення). Після натискання ЛК у правому полі **Binding** (Прив'язка) з'являється вікно **Binding setup** (Властивості прив'язки) табличного редактора аргументів шаблону екрана. Виділити ЛК у списку аргумент **Kob** й натисканням ЛК на екранній кнопці **Ok** (Готово) підтвердити прив'язку атрибута ГЕ **Текст** до даного аргументу шаблону екрана. В результаті вікно властивостей ГЕ матиме вигляд, показаний на рис. 9.

Аналогічно прив'язати атрибути інших **Text** (Текст) до відповідних аргументів шаблону екрана.

У вікні властивостей всіх ГЕ **Текст** обов'язково задати бажані графічні параметри, а також вирівняти їх на екрані (по вертикалі, по горизонталі, однакові проміжки тощо), використовуючи меню **Tools** на інструментальній панелі.

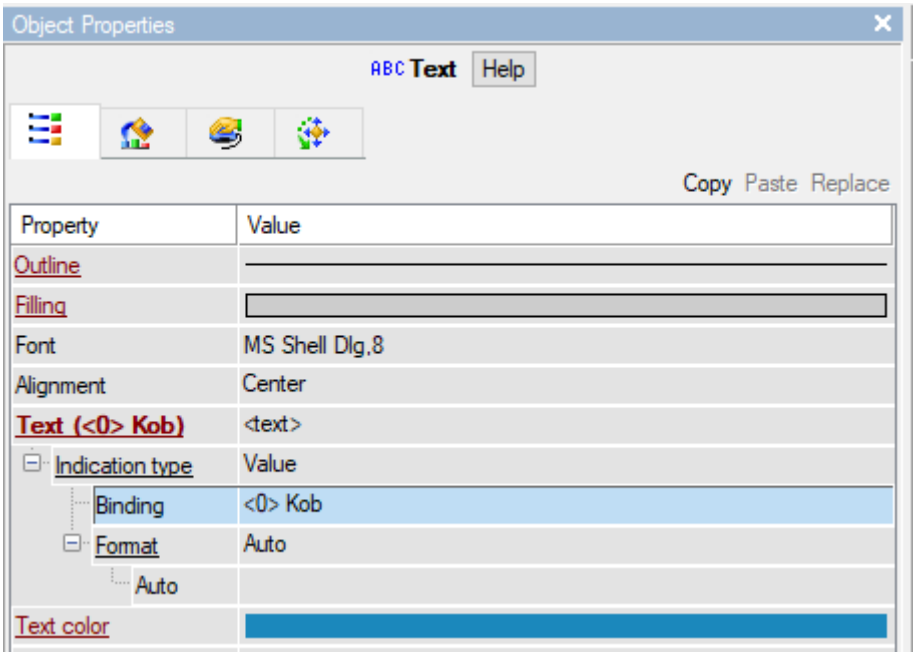


Рисунок 9 – Вікно властивостей ГЕ **Текст** після прив’язки до аргументу **Kob**

1.5 Вказати у проекті свої дані.

На графічному екрані розмістити у зручному місці шифр групи та своє ПІБ аналогічно до роботи № 1.

На графічному екрані розмістити вихідні дані об’єкту керування САР відповідно до свого варіанту та розраховані параметри ПІД-регулятора.

В меню **File\Information...**(Файл\Інформація...) в рядку **Owner** (Власник) написати за допомогою клавіатури своє ПІБ, а в рядку **Organization** (Організація) – шифр групи та факультету.

1.6 Запустити проект в роботу.

Зберегти проект натисканням кнопки . На головній інструментальній панелі ІСР натиснути ЛК кнопку **Save for RTM** (Зберегти для МРЧ), скомпілювавши проект для запуску в реальному часі. Виділити ЛК в навігаторі проекту вузол **RTM_1**, а потім натиснути на головній панелі кнопку **Start Profiler** (Запустить профайлер). У вікні, що відкрилося, запустити проект на виконання шляхом натискання ЛК кнопки **Run** (Пуск/Зупинка).

За допомогою кнопок ввести розраховані для свого варіанту значення параметрів САР та бажане значення сигналу завдання. Перевірити правильність їх відображення на графічному екрані (рис. 10).

Зупинити виконання проекту шляхом натискання на панелі інструментів ЛК кнопки **Run** (Пуск/Зупинка). Закрити вікно графічного екрану через меню **File\Exit** (Файл\Вихід) або засобами Windows. Зберегти проект натисканням кнопки та завершити виконання роботи.

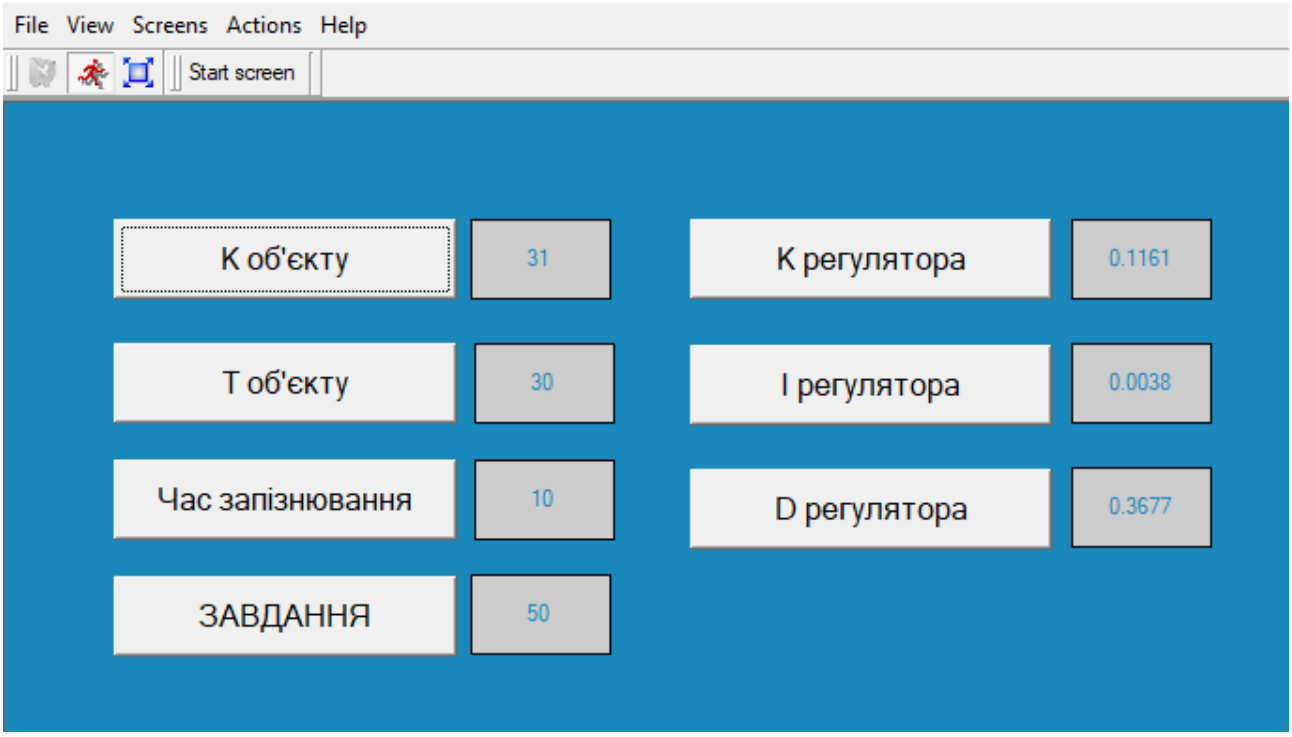


Рисунок 10 – Результат роботи проекту після запуску

1.7 Оформити протокол комп’ютерного практикуму відповідно до вимог, наведених у вступі.

2 ЗАВДАННЯ ДО КОМП’ЮТЕРНОГО ПРАКТИКУМУ № 4

Варіант завдання визначається номером першої букви **П**різвища та **І**мені студента в українському алфавіті відповідно до рис. 11.

Рисунок 11 – Номери букв українського алфавіту

Номер першої букви **Прізвища** – це величина коефіцієнта підсилення об'єкту керування **Kob**, а **Імені** – стала часу об'єкту **Tob**. Час запізнювання дорівнює **Tau=Tob/3**. Наприклад, прізвище та ім’я студента **Ц**иганков **Р**услан. В результаті отримується **Ц**=27; **Р**=21, отже **Kob=27**; **Tob=21;** **Tau=21/3=7**. Параметри ПІД-регулятора розраховуються на основі отриманих параметрів об’єкта керування відповідно до залежностей, наведених у підрозділі 1.1 даної роботи. Важливо: параметри ПІД-регулятора розрахувати з точністю до 4 цифр (крім нулів) після коми.

3 ПИТАННЯ ДЛЯ ЗАХИСТУ КОМП’ЮТЕРНОГО ПРАКТИКУМУ № 4

1. Етапи створення проектів систем автоматизації в SCADA.
2. Завдання властивостей ГЕ «Кнопка».
3. Завдання властивостей ГЕ «Текст».
4. Створення аргументів екрана та завдання їхніх параметрів.
5. Прив’язка атрибутів елементів до аргументів екрана.

Комп’ютерний практикум № 5

ЗАПУСК ТА НАЛАГОДЖЕННЯ ПРОЕКТУ ОДНОКОНТУРНОЇ СИСТЕМИ АВТОМАТИЧНОГО РЕГУЛЮВАННЯ

Мета роботи – освоїти методику програмування мовою FBD одноконтурної системи автоматичного регулювання за допомогою SCADA-системи.

1 ПРОГРАМА РОБОТИ

1.1 Відкрити проект, створений у роботі № 4, за допомогою меню **File\Open...** (Файл\Відкрити...) або кнопки **Open existing project** (Відкрити існуючий проект) на інструментальній панелі ICP.

1.2 Доповнити проект ГЕ **Trend** (Тренд) для перегляду у часі сигналу завдання та вихідної змінної САР. У нижній частині графічного екрана розмістити ГЕ **Тренд** для виводу значень змінних **Zav** та **Vykhid** (рис. 1).

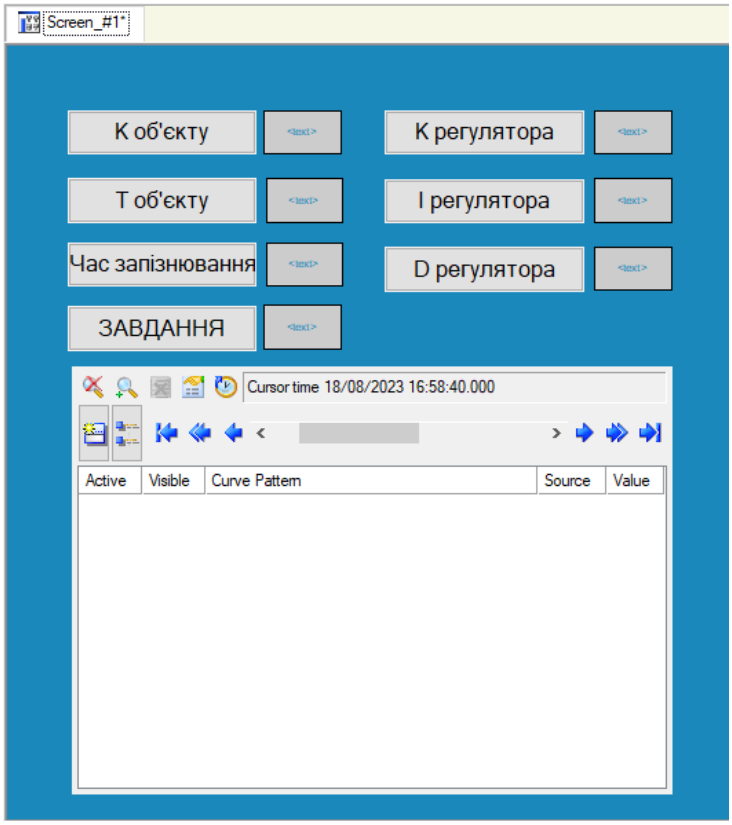


Рисунок 1 – Розміщення ГЕ **Тренд** на екрані

Основні властивості ГЕ **Тренд** залишити заданими за згодою (автоматично). Перейти у праву вкладку вікна властивостей **Curves** (Криві), виділивши ЛК рядок **Curves**, а потім за допомогою ПК послідовно створити дві нові криві **Curve**. Налаштувати їхні прив'язки до аргументів **Zav** та **Vykhid**, а товщину й колір ліній як показано на рис. 2.

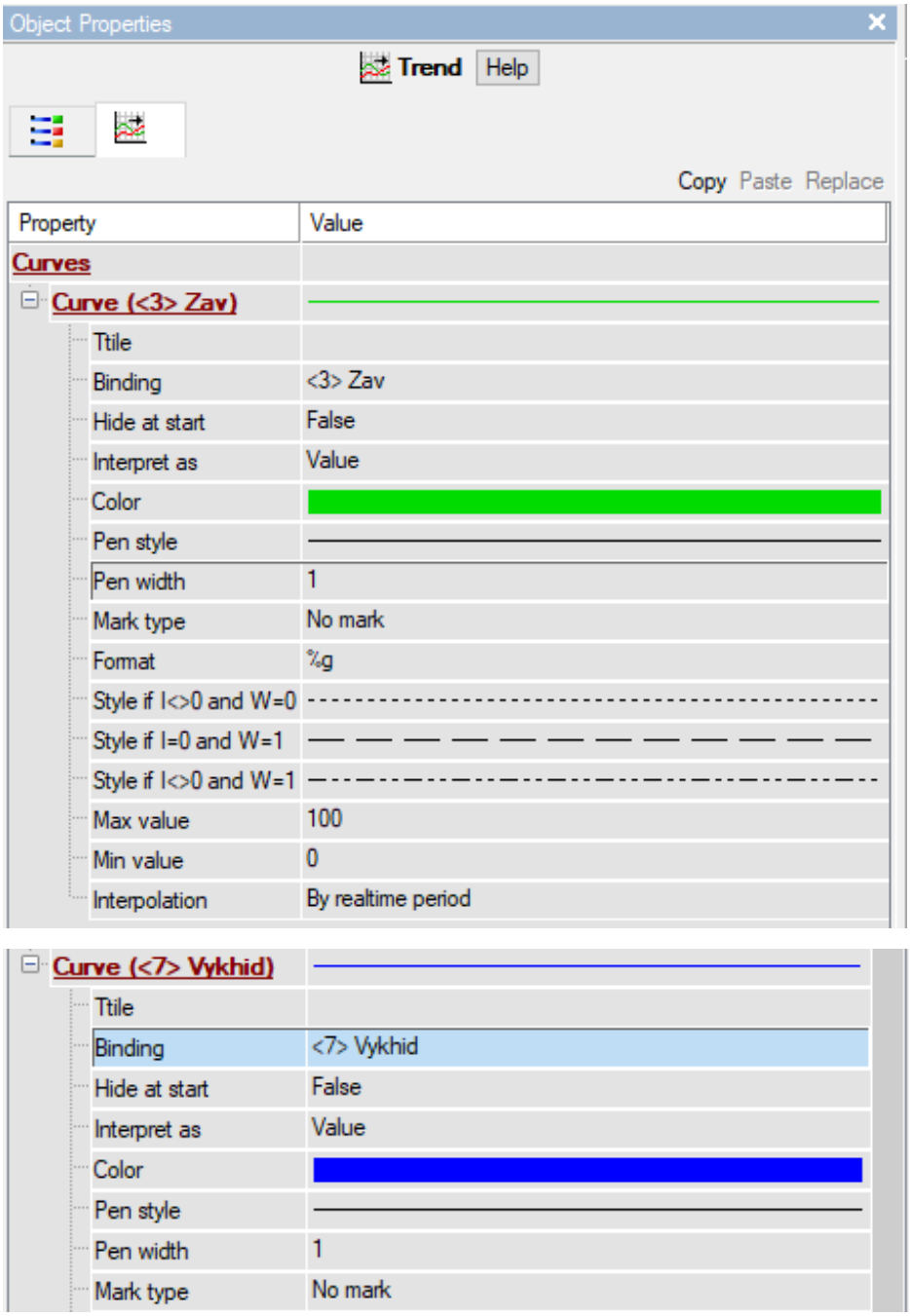


Рисунок 2 – Властивості графічного елемента **Тренд**

Зберегти проект з новим ім’ям шляхом вибору в меню **File** (Файл) пункту **Save As...** (Зберегти як...), задавши з клавіатури бажане ім’я, наприклад **Robota_5**.

1.3 Створити за аргументами шаблону екрана нові канали та відредагувати їхню прив'язку. У шарі **System** (Система) відкрити вузол **RTM_1** та за допомогою ПК на компоненті **Screen_#1:1** (Екран_#1:1) вибрати через контекстне меню пункт **Properties** (Властивості) (рис. 3).

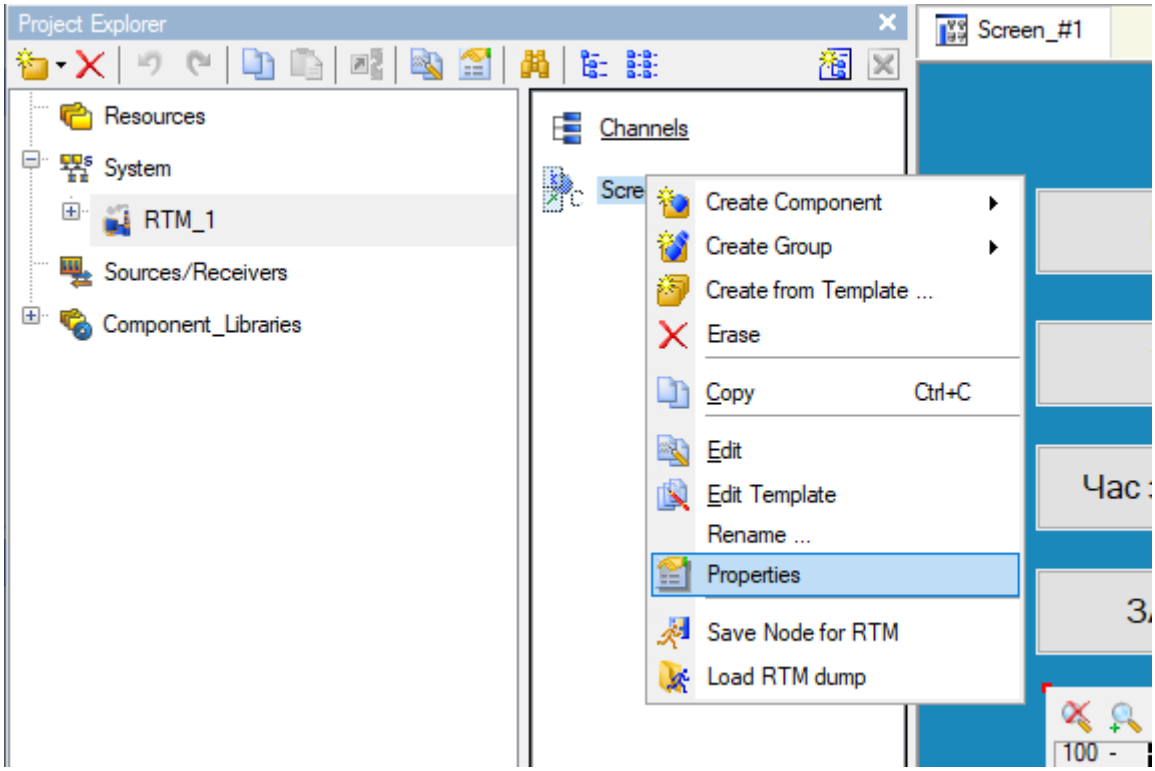


Рисунок 3 – Вибір меню **Властивості** екрана

У відкритому вікні вибрати вкладку **Arguments** (Аргументи) та ЛК при натиснутій на клавіатурі клавіші **Ctrl** виділити аргументи **Kob, Tob, Tau, Zav, Kr, Ti, Td, Vykhid** й за допомогою кнопки створити нові канали (рис. 4).

В результаті у вузлі **RTM_1** будуть автопобудовані канали, що показані на рис. 5.

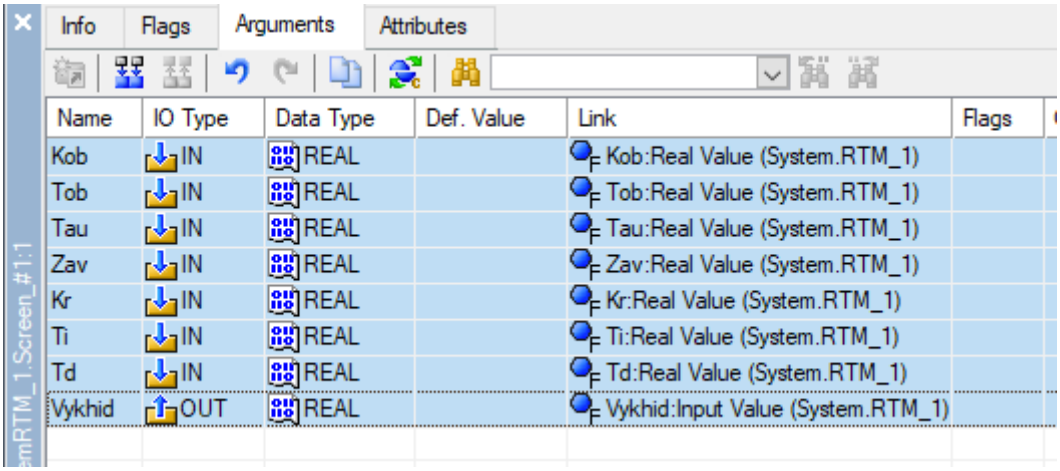


Рисунок 4 – Створення нових каналів

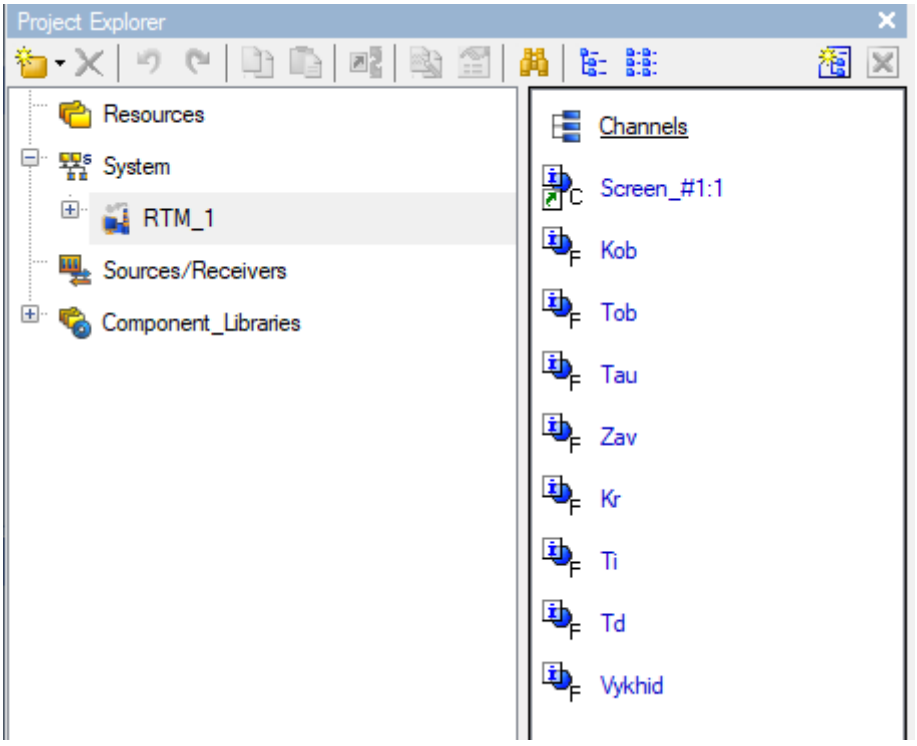


Рисунок 5 – Автопобудовані канали

1.4 Створити програму мовою FBD, яка буде реалізовувати одноконтурну систему автоматичного регулювання. Для цього ПК на вузлі **RTM_1** викликати контекстне меню та вибрати **Create Component** (Створити компоненту), а потім ЛК компонент **Program** (Програма) (рис 6).

Виділити компоненту **Program#1:2** та ПК викликати контекстне меню, вибравши ЛК пункт **Edit Template** (Редагувати шаблон) (рис. 7), й перейти в режим редагування програми (рис. 8).

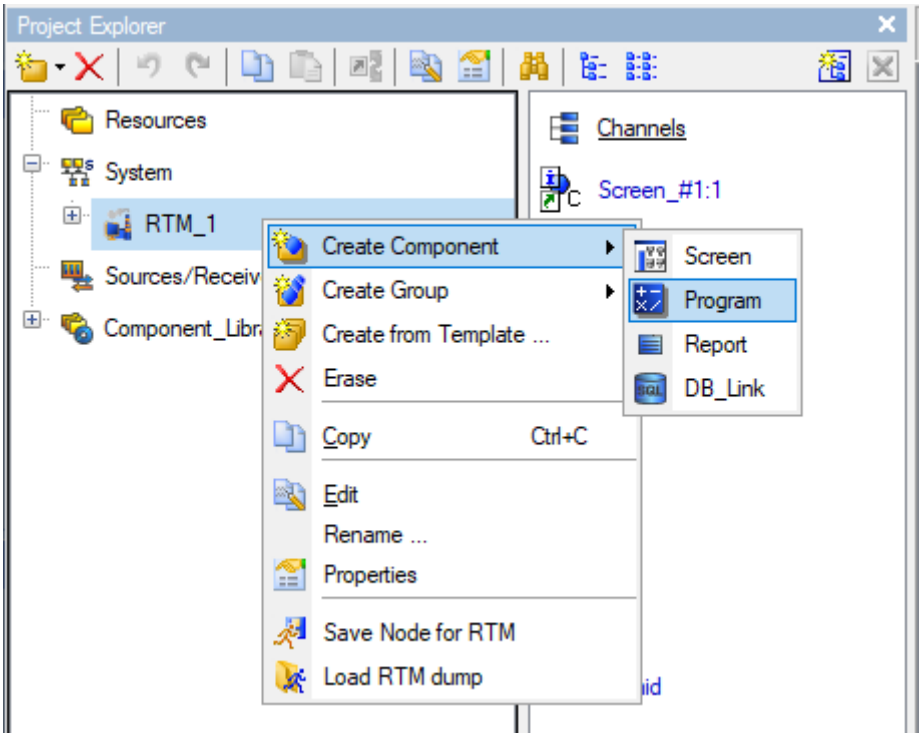


Рисунок 6 – Створення компоненти **Програма**

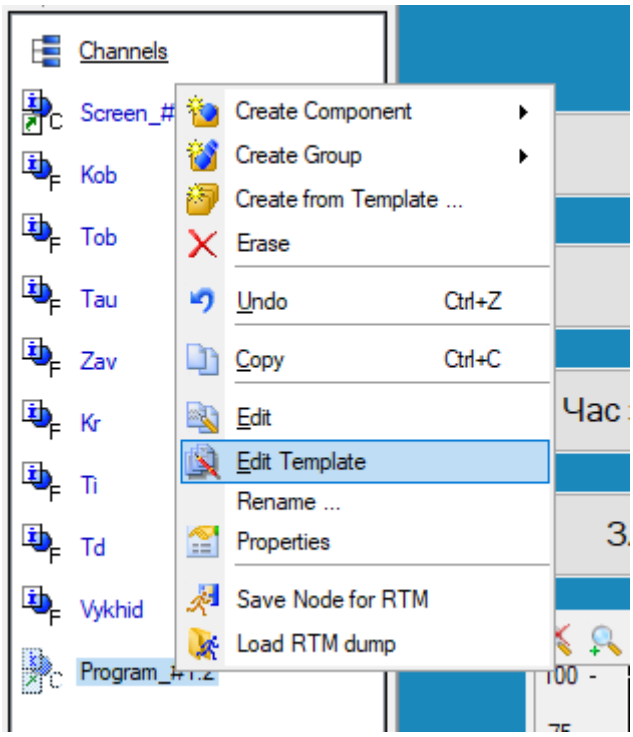


Рисунок 7 – Редагування аргументів **Програма**

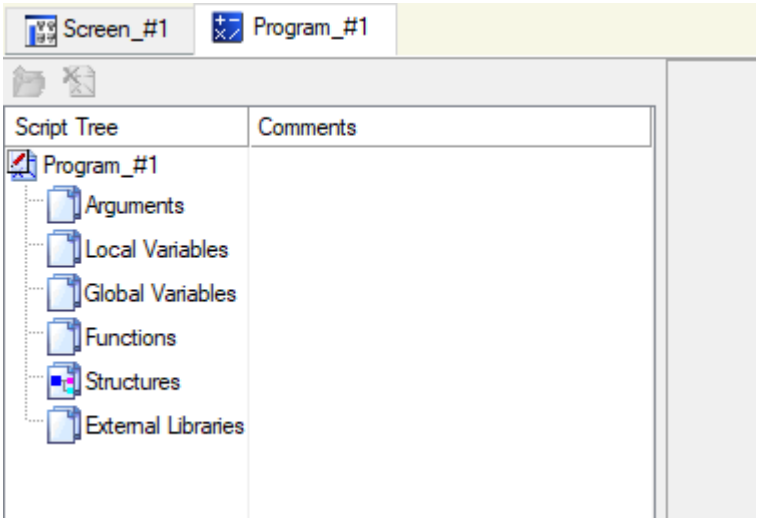


Рисунок 8 – Режим редагування програми

Виділити ЛК у дереві шаблону **Program#1** (Програма#1) рядок **Arguments** (Аргументи) та викликати табличний редактор. За допомогою кнопки створити послідовно аргументи **Kob, Tob, Tau, Zav, Kr, Ti, Td** типу **IN** та **Vykhid** типу **OUT** (рис 9).

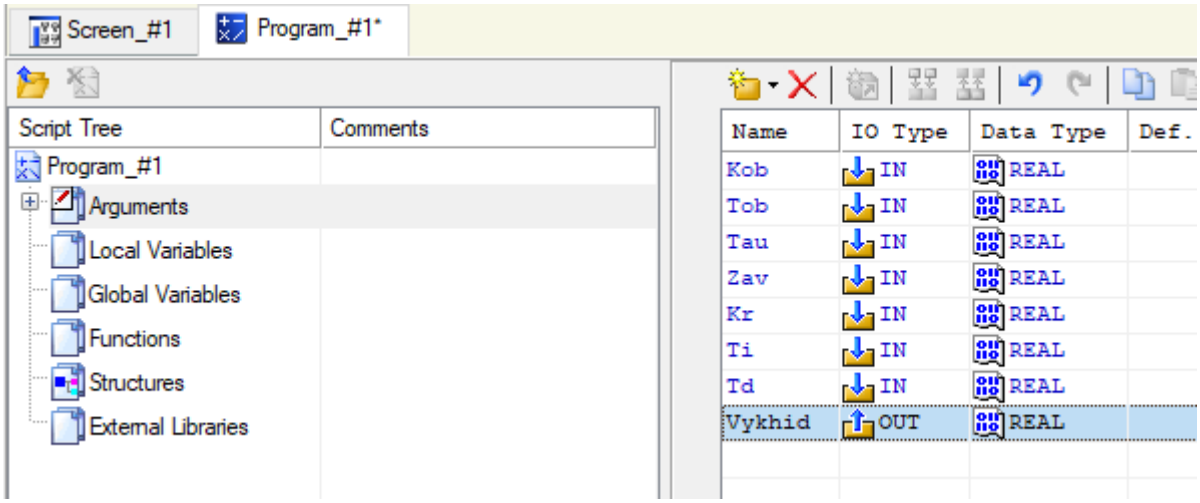


Рисунок 9 – Аргументи програми

Виділити ЛК в дереві шаблону рядок **Program#1** та у вікні, що відкрилося, вибрати мову **Function Block Diagram** (FBD-діаграма) (рис. 10). Після натискання кнопки **Ok** (Прийняти) у вікні редактора програм з оголошеними змінними створити програму відповідно до завдання. Для

виклику палітри FBD блоків необхідно ЛК натиснути на кнопку **Show/Hide FBD blocks palette** (Показати/сховати палітру блоків FBD), після чого з'явиться вікно вибору FBD блоків. Для зручності користування вікно FBD блоків за допомогою ЛК можна розмістити у бажаному місці на екрані та змінити його розміри.

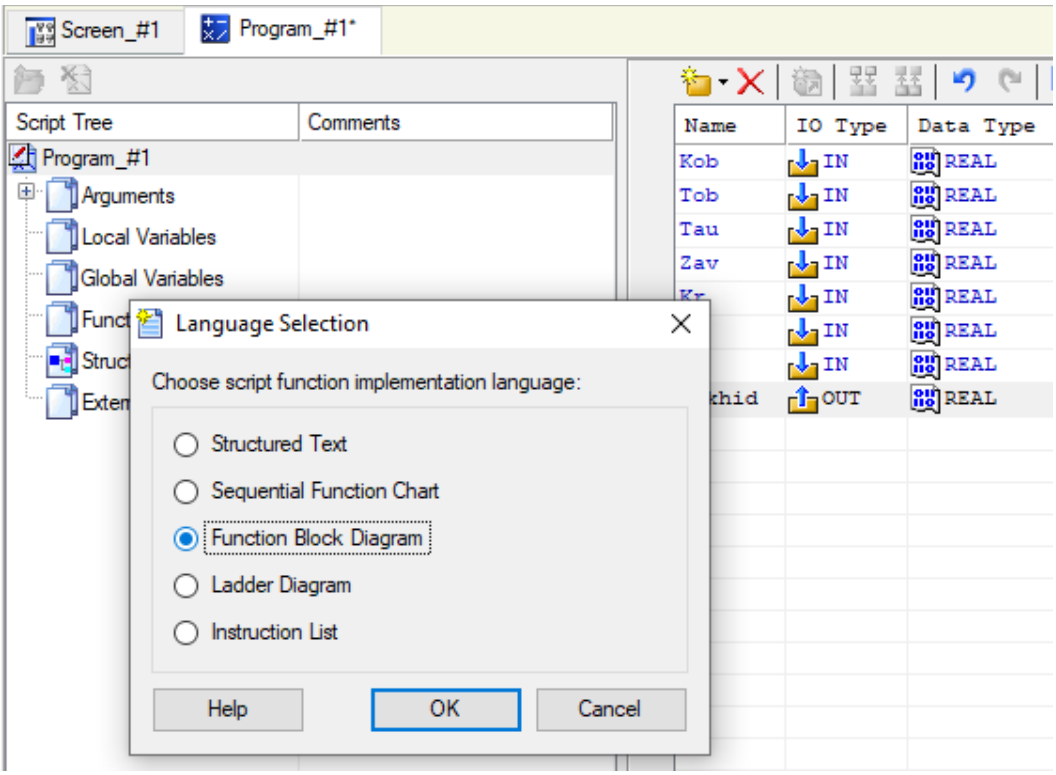


Рисунок 10 – Вибір мови програмування

Для створення системи регулювання з палітри FBD блоків за допомогою натиснутої ЛК перемістити на поле програми наступні блоки: з розділу **Arithmetic** (Арифметичні) блок **Subtraction** (Віднімання) (рис. 11), а з розділу **Control Loops** (Регулювання) блоки **PID Loop** (ПІД-регулятор) та **Object modeling** (Модель об’єкта) (рис. 12). Після розміщення всіх блоків поле програми буде виглядати, як показано на рис. 13.

Розробка програми на основі FBD блоків полягає у з’єднанні їх входів/виходів, подачі вхідного сигналу **Zav** та вихідного **Vykhid**. Треба мати на увазі, що верхні входи FBD блоків не використовуються, а інформаційними

є входи, починаючи із другого. З'єднання входів здійснюється при натиснутій ЛК. Подача та вибір сигналу здійснюється після подвійного натискання ЛК на контакті блоку. Можна використовувати також кнопки панелі інструментів вгорі вікна (рис. 13) або контекстне меню після ПК на виділеному блоці. Після виконання всіх операцій програма буде мати вигляд, показаний на рис. 14.

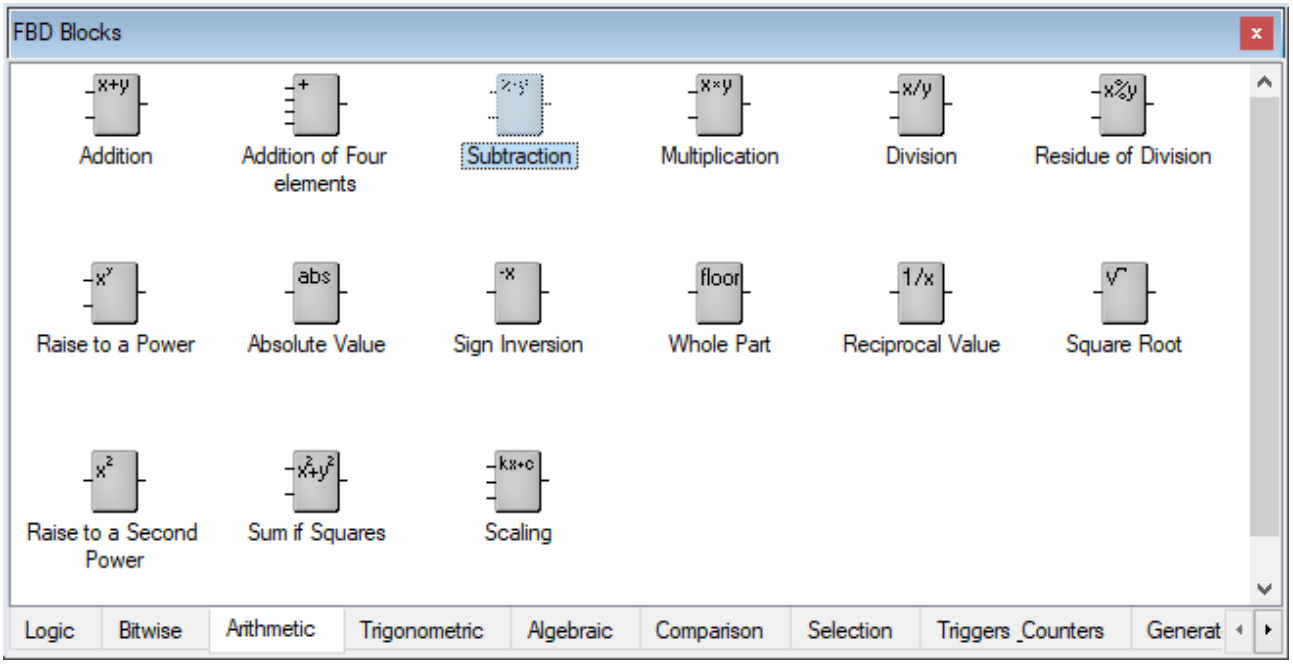


Рисунок 11 – Розділ **Арифметичні** палітри FBD блоків

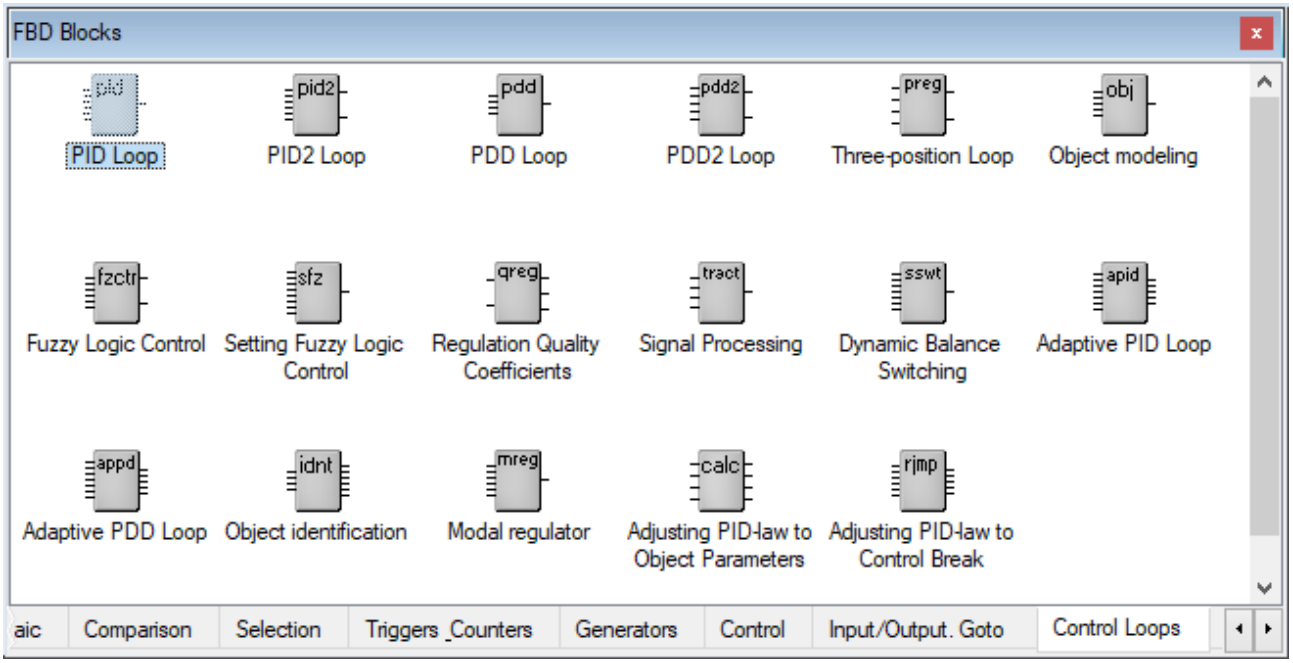


Рисунок 12 – Розділ **Регулювання** палітри FBD блоків

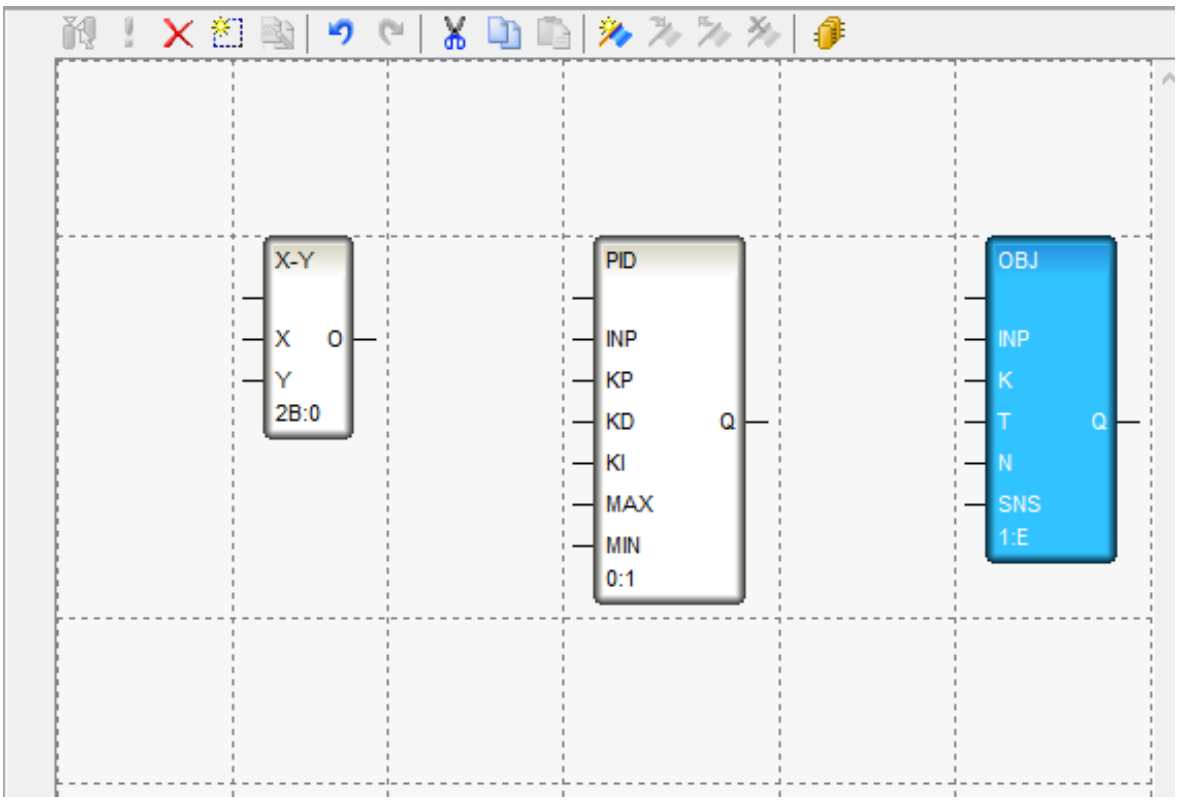


Рисунок 13 – Поле програми з розташованими FBD блоками

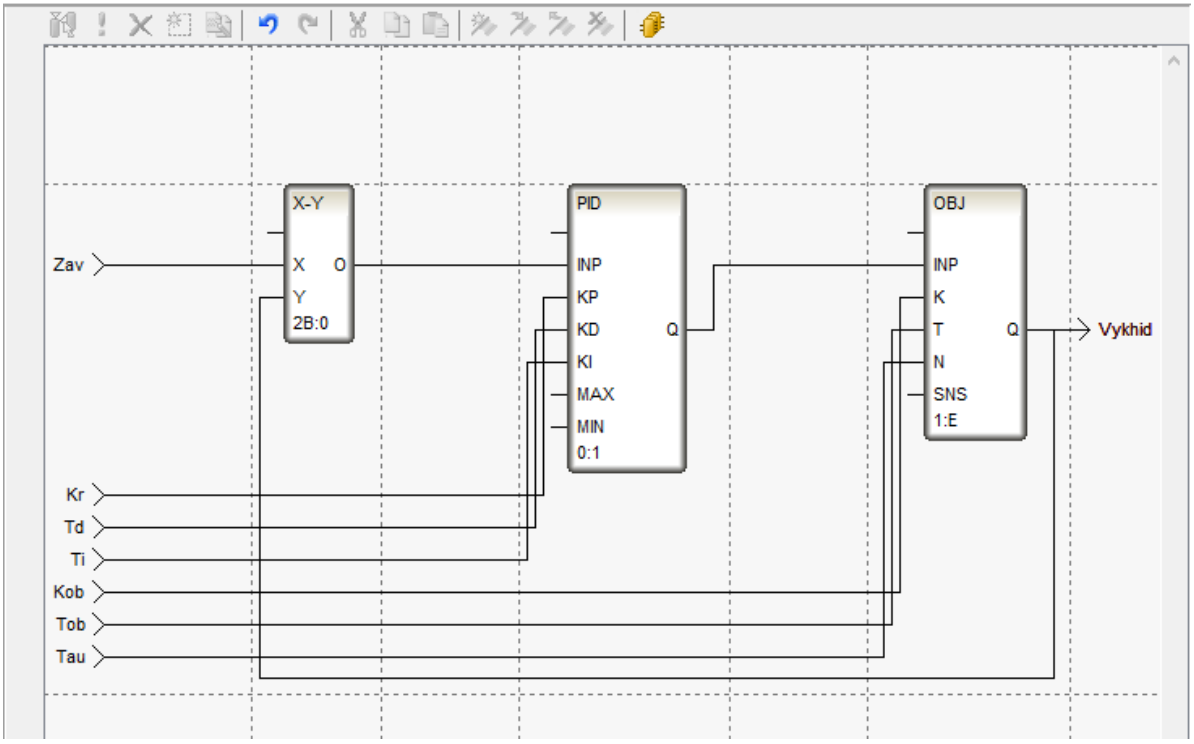


Рисунок 14 – Програма реалізації САР

За допомогою іконки **Build (F7)** (Компіляція) на інструментальній панелі редактора або гарячої клавіші F7 скомпілювати програму та переконатися в успішній компіляції у вікні **Messages**, яке при необхідності викликається з інструментальної панелі за допомогою іконки **Messages** (Повідомлення) (рис. 15).

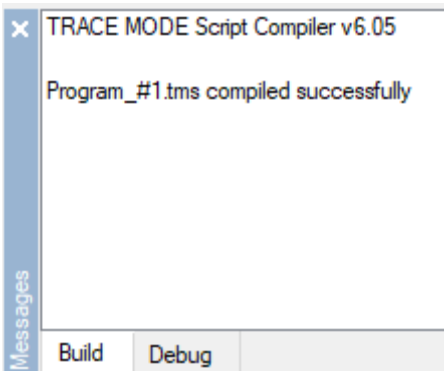


Рисунок 15 – Результат успішної компіляції програми

1.5 Виконати прив'язку аргументів програми до атрибутів каналів. У вікні навігатора проекту **Project Explorer** ПК на компоненті **Program#1:2** викликати контекстне меню та вибрати **Properties** (Властивості). У вікні, що відкрилося, вибрати вкладку **Arguments** (Аргументи)**.** Подвійним натисканням ЛК у полі **Link** (Прив’язка) для рядка **Kob** відкрити вікно конфігурування зв’язку. Вибрати у лівій частині вікна елемент **Kob**, потім у правій частині на вкладці **Attributes** (Атрибути) рядок **Real Value** (Реальне значення) та натиснути кнопку **Link** (Прив’язка) (рис. 16). Аналогічно прив'язати аргументи програми **Tob, Tau, Kr, Ti, Td, Zav** до реального значення каналів **Tob, Tau, Kr, Ti, Td, Zav**.

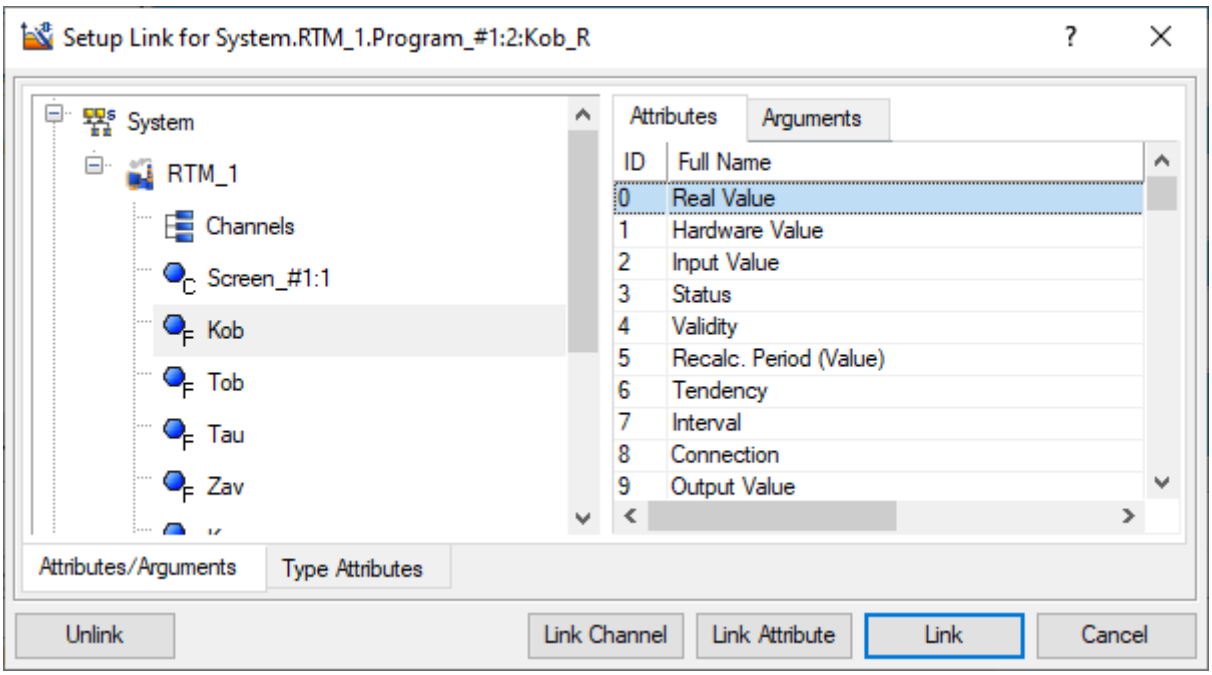


Рисунок 16 – Зв'язок аргументів програми з аргументами екрана

Подвійним натисканням ЛК у полі **Link** для аргументу програми **Vykhid** викликати вікно настроювання прив'язки. Вибрати в лівому вікні канал **Screen_#1:1**, а в правому вікні замість вкладки **Attributes** вибрати вкладку **Arguments** та вказати в ній аргумент **Vykhid** й кнопкою **Link** підтвердити зв'язок (рис. 17).

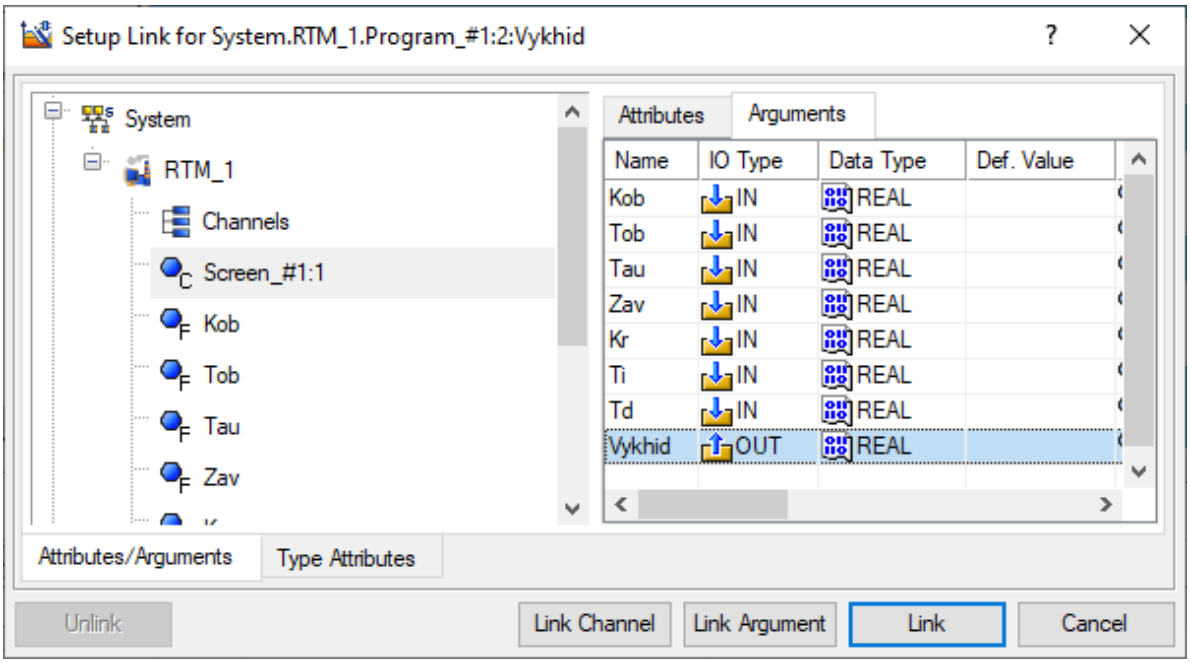


Рисунок 17 – Зв'язок вихідного аргументу

Остаточний результат операцій показано на рис. 18.

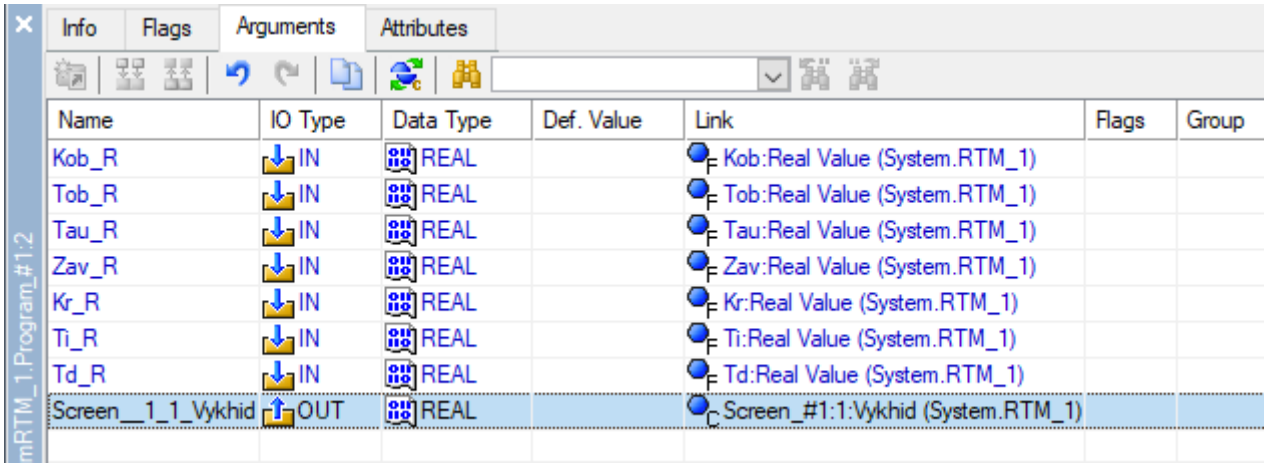


Рисунок 18 – Остаточне настроювання зв'язку

1.6 Запустити проект в роботу.

Зберегти проект натисканням кнопки . На головній інструментальній панелі ICP натиснути ЛК кнопку **Save for RTM** (Зберегти для МРЧ), скомпілювавши проект для запуску в реальному часі. Виділити ЛК в навігаторі проекту вузол **RTM_1**, а потім натиснути на головній панелі кнопку **Start Profiler** (Запустить профайлер). У вікні, що відкрилося, запустити проект на виконання шляхом натискання ЛК кнопки **Run** (Пуск/Зупинка).

За допомогою кнопок спочатку ввести параметри об’єкта, розраховані значення параметрів ПІД-регулятора, а потім сигнал завдання. Спостерігати на ГЕ **Тренд** перехідний процес регульованої змінної, яка після декількох коливань повинна прийняти усталене значення (рис. 19).

Зупинити виконання проекту шляхом натискання на панелі інструментів ЛК кнопки **Run** (Пуск/Зупинка). Закрити вікно графічного екрану через меню **File\Exit** (Файл\Вихід) або засобами Windows. Зберегти проект натисканням кнопки та завершити виконання роботи.

1.7 Оформити протокол комп’ютерного практикуму відповідно до вимог, наведених у вступі.

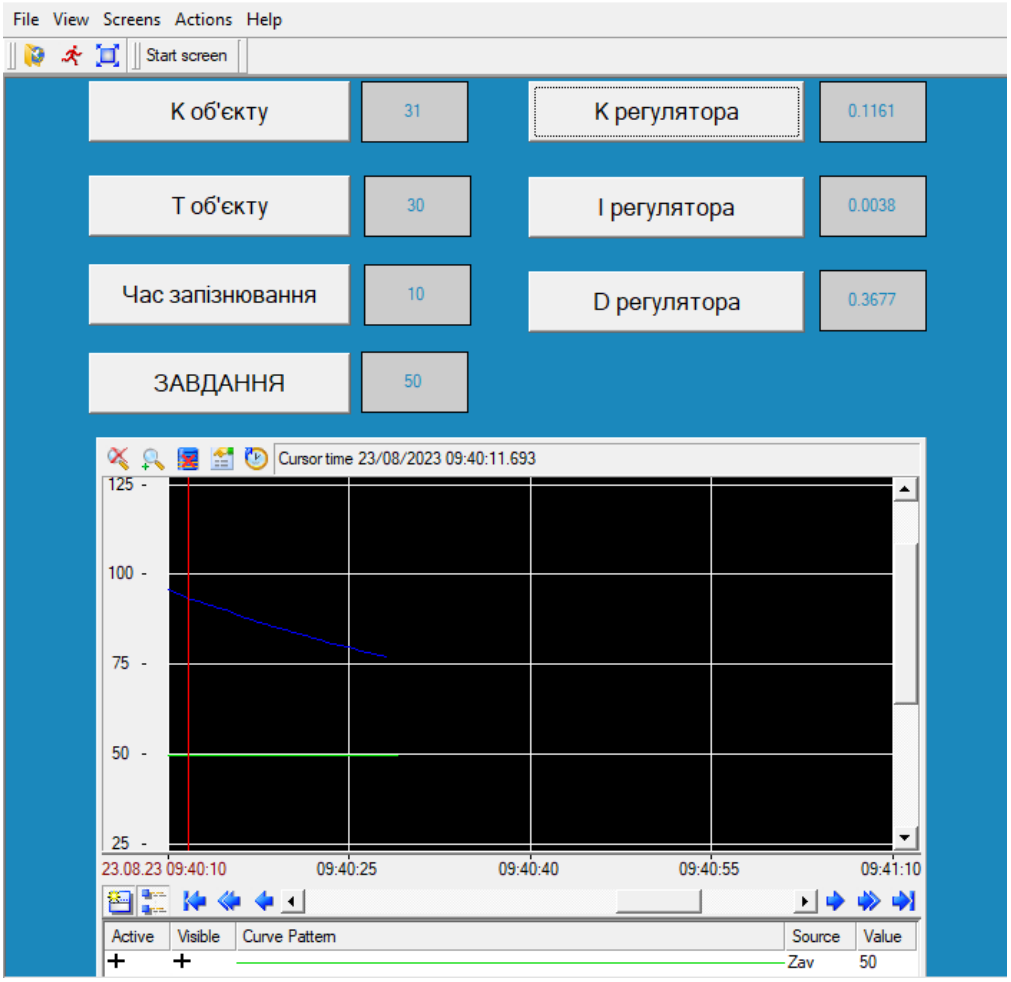


Рисунок 19 – Результат роботи проекту після запуску

2 ПИТАННЯ ДЛЯ ЗАХИСТУ КОМП'ЮТЕРНОГО ПРАКТИКУМУ № 5

1. Етапи створення проектів систем автоматизації в SCADA.
2. Завдання властивостей ГЕ «Кнопка».
3. Завдання властивостей ГЕ «Текст».
4. Завдання властивостей ГЕ «Тренд».
5. Створення аргументів екрана та завдання їхніх параметрів.
6. Прив'язка атрибутів елементів до аргументів екрана.
7. Автопобудова каналів.
8. Призначення входів/виходів FBD-блоків, використовуваних у роботі.
9. Послідовність створення FBD-програми.

Комп’ютерний практикум № 6

СТВОРЕННЯ СТАТИЧНИХ ТА ДИНАМІЧНИХ ЗОБРАЖЕНЬ

Мета роботи – засвоєння послідовності створення статичних та динамічних зображень в SCADA-системі.

1 ПРОГРАМА РОБОТИ

Зобразити на екрані ємність з вхідним на вихідним трубопроводом. На вихідному трубопроводі розташувати насос. За допомогою динамічного зображення зімітувати процес наповнення ємності рідиною. Рівень рідини прив’язати до сигналу вбудованого генератора. На екрані розташувати тренд для контролю рівня рідини. Приклад графічного екрану за результатами виконання роботи показано на рис. 1.

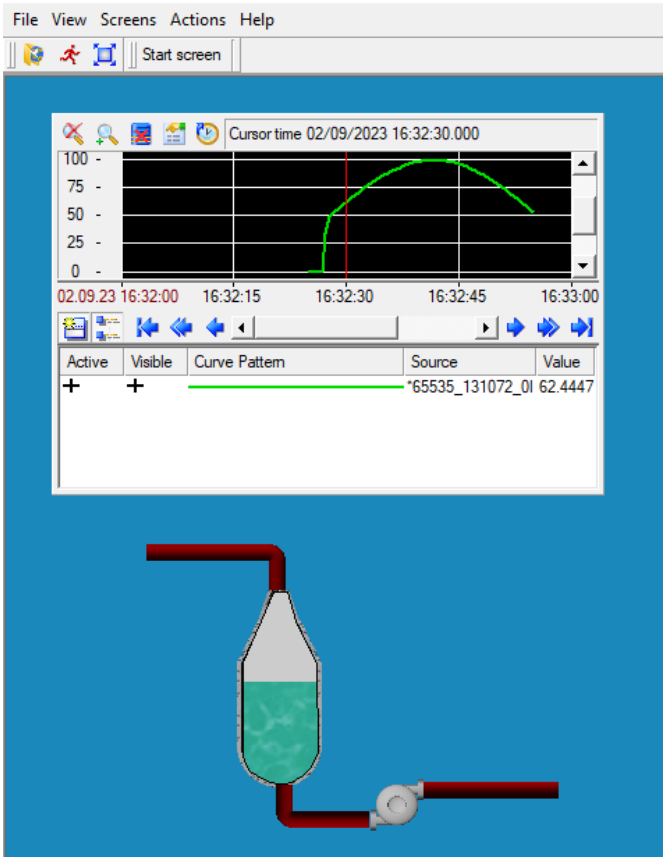


Рисунок 1 – Приклад екрану проекту системи автоматизації

1.1 Створити новий проект стилю **Easy** (Простий), як у роботі № 1. Подвійним натисканням ЛК на компоненті **Screen_#1:1** (Екран_#1:1) відкрити вікно графічного редактора. Зберегти проект шляхом вибору в меню **File** (Файл) пункту **Save** (Зберегти) або натисканням кнопки на інструментальній панелі, задавши з клавіатури бажане ім'я, наприклад **Robota_6**.

1.2 Створити генератор синусоїдного сигналу для імітування наповнення ємності рідиною.

У лівому вікні навігатора проекту для шару **Sources/Receivers** (Джерела/Приймачі) за допомогою ПК створити групу **Generators** (Генератори), як показано на рис. 2.

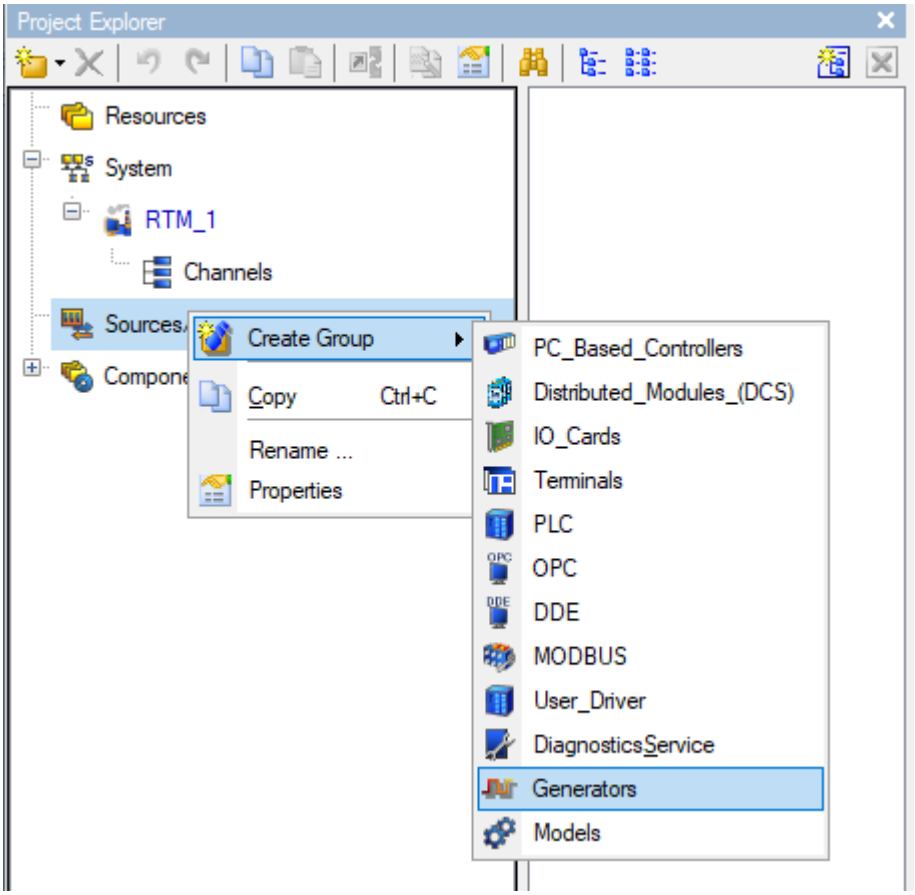


Рисунок 2 – Створення групи **Генератори**

За допомогою ПК на створеній групі **Generators_1** (Генератори_1) створити компоненту **Sinus** (Синусоїда) (рис. 3).

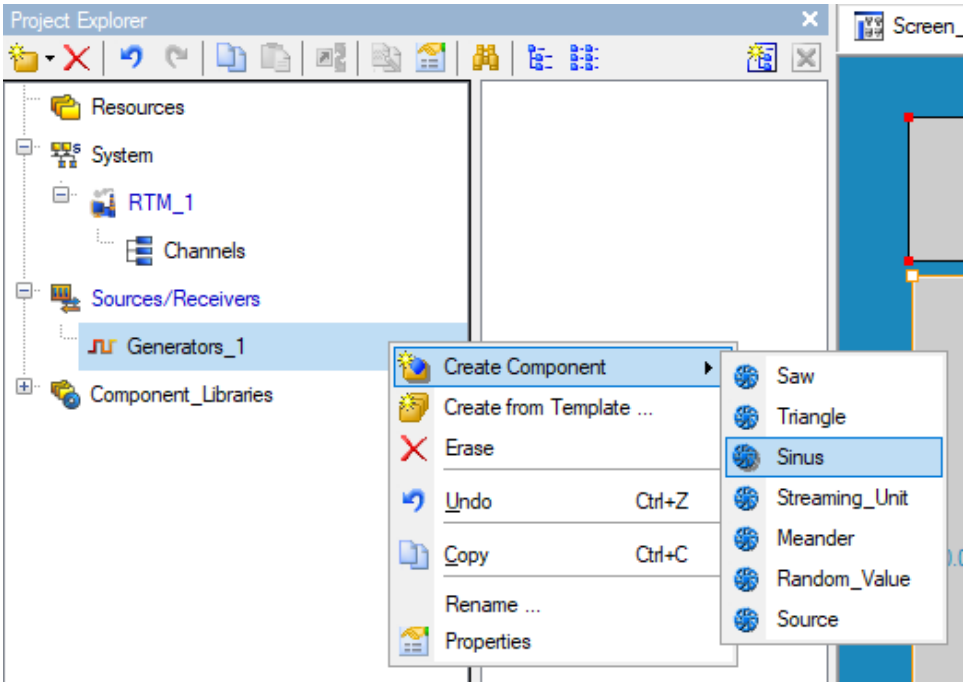


Рисунок 3 – Створення компоненти **Синусоїда**

1.3 Розмістити на екрані ГЕ **Trend** (Тренд).

Доповнимо екран ГЕ **Trend** (Тренд) для перегляду зміни рівня рідини в ємності у часі та відстеження передісторії. Вибрати ЛК на інструментальної панелі графічного редактора екрана ГЕ **Trends** (Тренди), а в ньому іконку ГЕ **Trend** (Тренд) та розмістити його на екрані у верхньому лівому куті. Перейти у праву вкладку вікна властивостей **Curves** (Криві), виділивши ЛК рядок **Curves**, а потім за допомогою ПК створити нову криву **Curve** та налаштувати бажані властивості, зокрема товщину кривої. У правому полі рядка **Binding** (Прив’язка) ЛК відкрити вікно **Binding setup** (Властивості прив’язки) та за допомогою кнопки **Create Argument** (Створити аргумент) створити новий аргумент. У стовпці **Link** (Прив’язка) нового аргументу викликати подвійним натисканням ЛК вікно настроювання зв'язку. Вибрати у лівій частині вікна компоненту **Sinus#1**, а в правій частині **Value** (Значення) (рис. 4). Кнопкою **Link** підтвердити зв'язок кривої тренда з сигналом

генератора (рис. 5), а потім кнопкою **Ok** закрити вікно властивостей. Властивості **Кривої** графічного елемента **Тренд** показано на рис. 6.

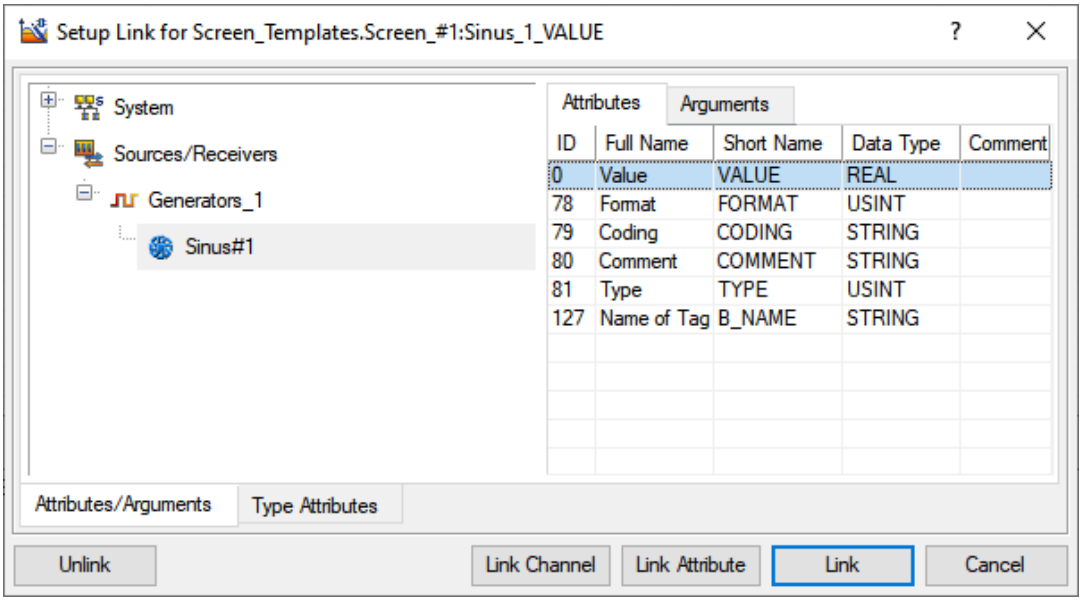


Рисунок 4 – Вікно настроювання прив’язки

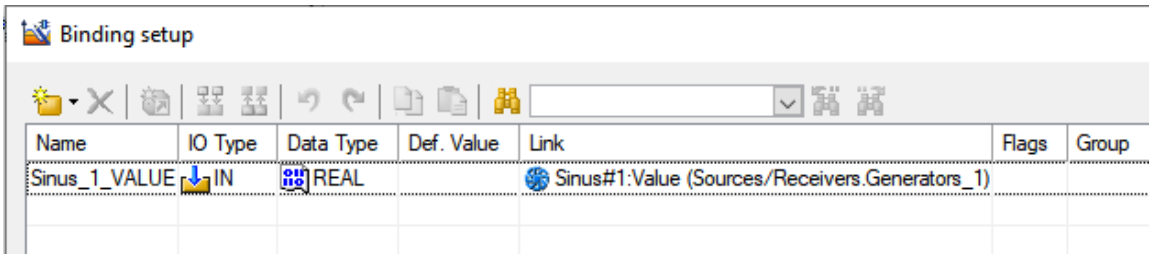


Рисунок 5 – Остаточний вигляд вікна **Властивості прив’язки**

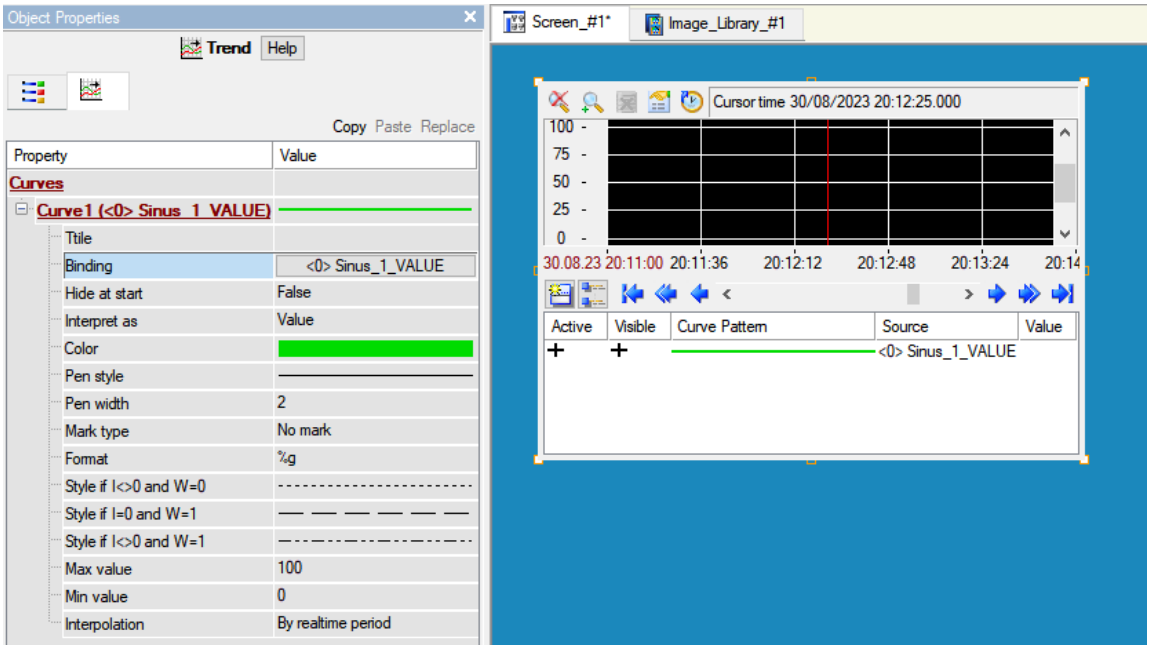


Рисунок 6 – Властивості **Кривої** графічного елемента **Тренд**

1.4 Створити статичне зображення ємності.

Натиснути ПК на інструментальній панелі іконку **Volumetric Elements** (Об’ємні фігури) і вибрати ГЕ **Tank** (Ємність) та розмістити на екрані нижче ГЕ **Тренд** (рис. 7).

Рисунок 7 – Розміщення на екрані ГЕ **Ємність**

Відкрити вікно властивостей елементу та у полі **Wall Thickness** (Товщина стінок) задати бажану товщину більше 0. Ємність стане зображена у розрізі (рис. 8).

Рисунок 8 – Вигляд ГЕ **Ємність** у розрізі

У полі **Top cap** (Верхній край) вибрати із списку бажаний вид краю, наприклад конусоподібний.

Задати матеріал ємності. Спочатку у полі **Material\Select from list** (Матеріал/Вибрати із списку) встановити значення **True**, а потім у полі **Materials** (Матеріали) вибрати зі списку бажаний матеріал, наприклад **Хром**. У полі **Standard Textures (**Стандартні текстури) вибрати бажану текстуру, приміром **Engraved** (Гравірування).

В результаті ємність буде мати вигляд як на рис. 9.

Рисунок 9 – Вигляд ГЕ **Ємність** з конусоподібним верхнім краєм

Виділити на панелі інструментів ПК іконку **Об'ємні фігури** та вибрати ГЕ **Конус** і розмістити його зверху на ємності. У вікні властивостей елементу у полі **Wall Thickness** (Товщина стінок) задати ту ж товщину, що й у ємності. Аналогічно до ГЕ **Ємність** вибрати бажаний **матеріал** та **гравірування**. Ємність прийме вигляд, зображений на рис. 10.

Рисунок 10 – Остаточний вигляд ємності

1.5 Створити статичне зображення насосу.

Виділити ПК на панелі інструментів іконку **Об'ємні фігури** та вибрати ГЕ **Pump** (Насос) і розмістити його внизу, праворуч від ємності (рис. 11). У вікні властивостей встановити бажаний **матеріал** та **форму** насосу.

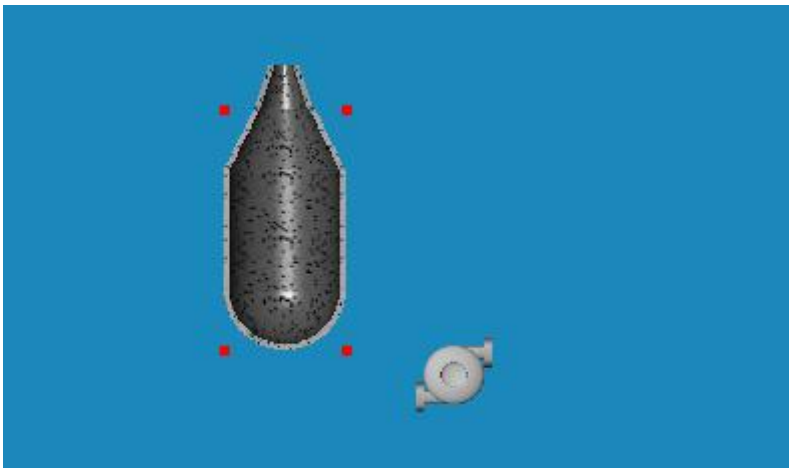


Рисунок 11 – Розміщення на екрані ГЕ **Насос**

1.6 Створити статичне зображення трубопроводів.

Виділити на панелі інструментів ПК іконку **Об'ємні фігури**, вибрати ГЕ **Pipe** (Труба) та розмістити його як вхідний трубопровід згори ємності, трубопровід між ємністю та насосом, вихідний трубопровід після насосу. Для вигину трубопроводу натискайте ЛК у потрібному місці на екрані. Завершення створення відрізку трубопроводу здійснюється ПК у потрібному місці екрану. У вікні властивостей елементу задайте бажану **Weight** (Товщину**)** та **Base color** (Колір) (рис. 12).

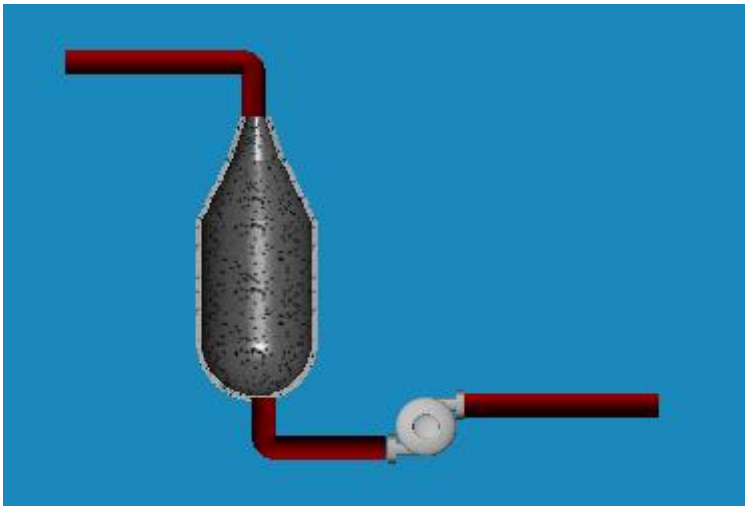


Рисунок 12 – Розміщення на екрані ГЕ **Труба**

1.7 Створити динамічне зображення рідини у ємності.

Виділити рядок **Resources** (Ресурси) навігатора проекту. Викликати ПК контекстне меню, вибрати рядок **Create Group** (Створити групу**)**, а потім **Images** (Картинки) (рис. 13).

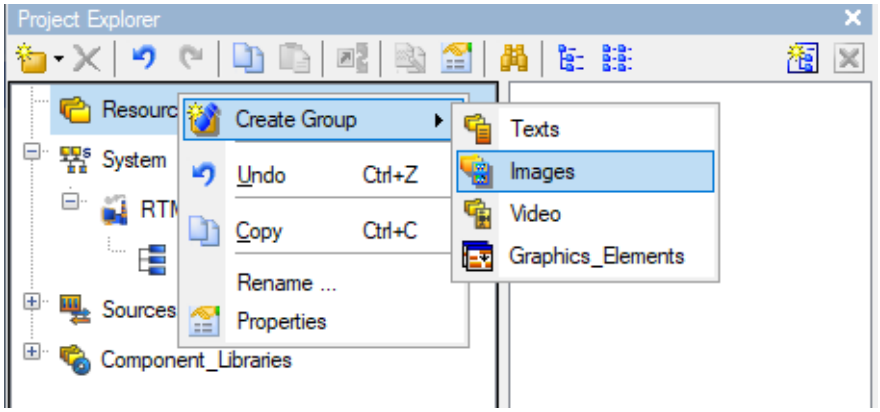


Рисунок 13 – Створення групи **Картинки**

Виділити створену групу **Images** (Картинки), викликати ПК контекстне меню, вибрати рядок **Create Component** (Створити компонент), а потім **Image_Liblary** (Бібліотека_Зображень) (рис. 14).

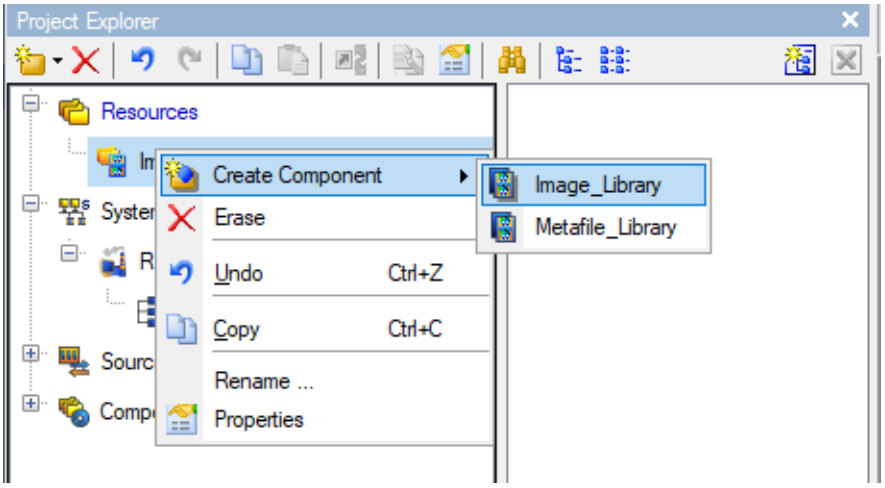


Рисунок 14 – Створення компоненти **Бібліотека_Зображень**

Подвійним ЛК на компоненті **Бібліотека_Зображень** відкрити відповідне вікно, на площині якого ПК викликати контекстне меню та вибрати рядок Import (Імпортувати) (рис. 15).

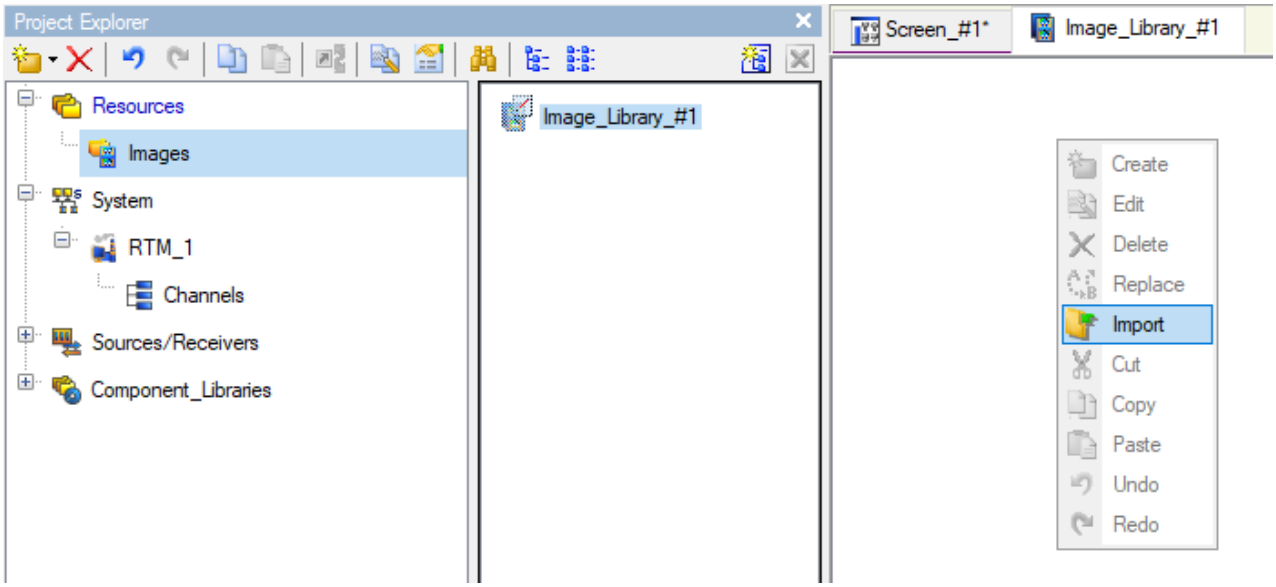


Рисунок 15 – Імпортування картинок у **Бібліотека_Зображень**

Відкриється діалогове вікно стандартного **Провідника** Windows для вибору на комп'ютері графічних файлів з рисунками текстур, що будуть імітувати рідину. У папці із встановленою програмою SCADA відкрити папку .../**Lib/Texture** та вибрати бажаний (бажані) графічний файл або інші файли з власної бібліотеки (рис. 16). У бібліотеку також потрібно завантажити своє фото для розміщення його на екрані проекту роботи. Файли бібліотеки стають також доступними через іконку **Resources** (Ресурси) на інструментальній панелі графічного екрану.

Рисунок 16 – Картинки у **Бібліотека_Зображень**

Виділити на панелі інструментів ПК іконку **Polylinens** (Ламані та криві), вибрати ГЕ **Polygon** (Багатокутник) та розмістити його на внутрішній поверхні ємності. Задати ЛК необхідну кількість точок зламу відрізків (кутів) багатокутника, а ПК – останній кут багатокутника (рис. 17). Для підвищення точності розташування багатокутника доцільно збільшити масштаб на панелі інструментів.

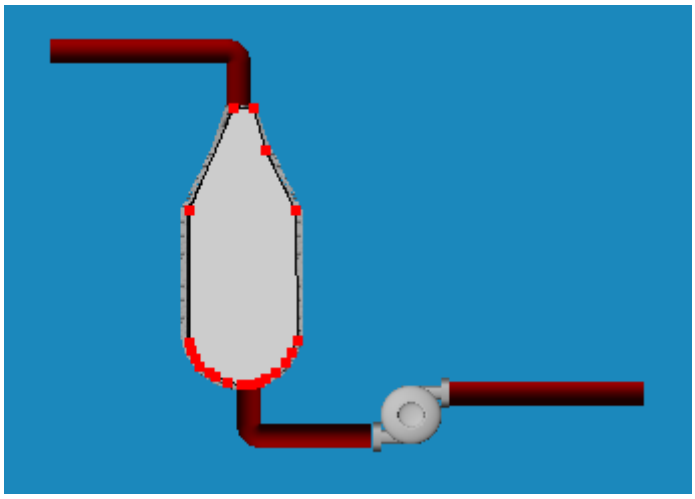


Рисунок 17 – Розташування ГЕ **Багатокутник** на внутрішній частині ємності

Відкрити вікно властивостей ГЕ **Багатокутник.** На закладці основних властивостей **Generals Properties** у рядку ***Visible** (Видимість) вибрати значення **True** (Дозволено) (рис. 18).

Перейти на вкладку динамічного заливання **Dynamic Fill**. Поставити прапорець **Enable** (Дозволено) та подвійним ЛК розкрити розділ **Layer** (Шари). У рядку **Binding** (Прив’язка) зробити прив'язку до аргументу **Sinus_1**, а у полі **Fill type** (Тип заливання) вибрати **Pixmap** (Зображення). Після ЛК у полі **Image** (Картинка) вибрати у вікні бібліотеки (рис. 16) бажане зображення. Вікно динамічного заливання властивостей ГЕ **Багатокутник** матиме вигляд, показаний на рис. 19.

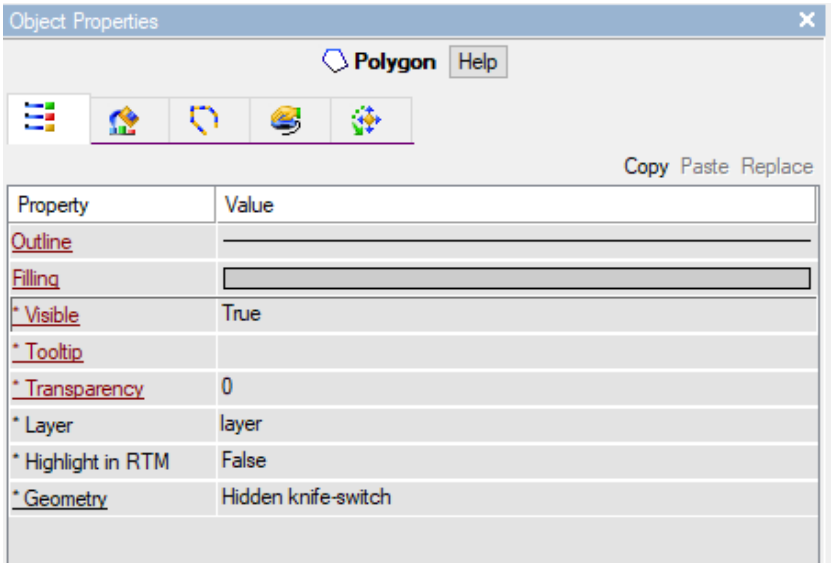


Рисунок 18 – Вікно основних властивостей ГЕ **Багатокутник**

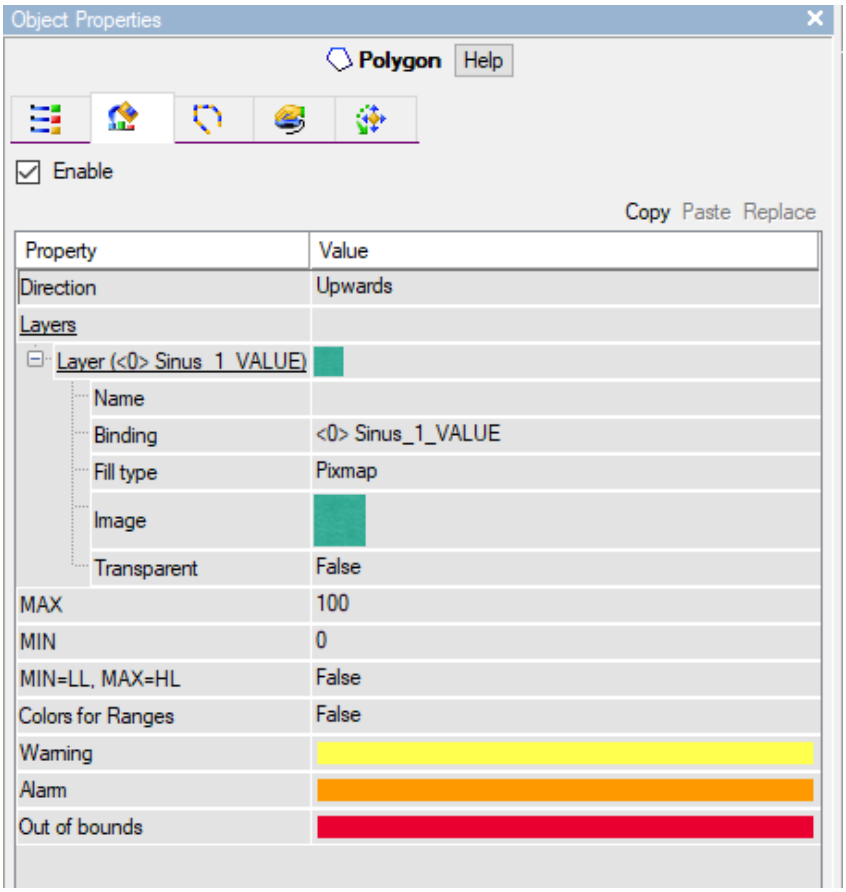


Рисунок 19 – Вікно динамічного заливання властивостей ГЕ **Багатокутник**

1.8 Вказати у проекті свої дані.

На графічному екрані розмістити у зручному місці шифр групи та своє ПІБ аналогічно до роботи № 1.

На графічному екрані розмістити у зручному місці своє фото.

В меню **File\Information...**(Файл\Інформація...) в рядку **Owner** (Власник) написати за допомогою клавіатури своє ПІБ, а в рядку **Organization** (Організація) – шифр групи та факультету.

1.9 Запустити проект в роботу.

Зберегти проект натисканням кнопки . На головній інструментальній панелі ICP натиснути ЛК кнопку **Save for RTM** (Зберегти для МРЧ), скомпілювавши проект для запуску в реальному часі. Виділити ЛК в навігаторі проекту вузол **RTM_1**, а потім натиснути на головній панелі кнопку **Start Profiler** (Запустить профайлер). У вікні, що відкрилося, запустити проект на виконання шляхом натискання ЛК кнопки **Run** (Пуск/Зупинка).

Спостерігати процес наповнення ємності рідиною та відповідне відображення процесу на тренді, як показано на рис. 20.

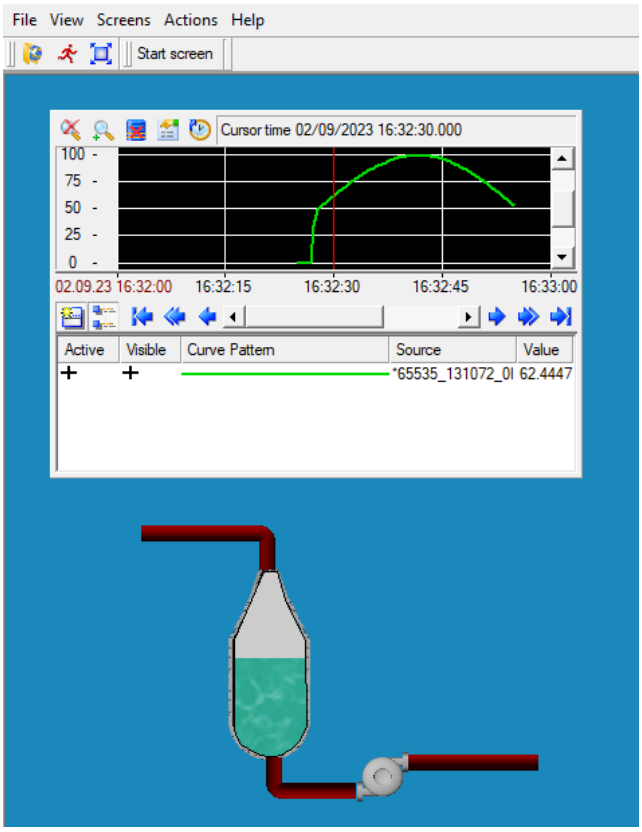


Рисунок 20 – Результат імітаційного запуску проекту

Зупинити виконання проекту шляхом натискання на панелі інструментів ЛК кнопки **Run** (Пуск/Зупинка). Закрити вікно графічного екрану через меню **File\Exit** (Файл\Вихід) або засобами Windows. Зберегти проект натисканням кнопки та завершити виконання роботи.

1.10 Оформити протокол комп'ютерного практикуму відповідно до вимог, наведених у вступі.

2 ПИТАННЯ ДЛЯ ЗАХИСТУ КОМП'ЮТЕРНОГО ПРАКТИКУМУ № 6

1. Етапи створення систем керування на основі SCADA.
2. Створення статичного зображення.
3. Створення динамічного зображення.
4. Відмінність статичного зображення від динамічного.
5. Імпорт зображень.
6. Завдання властивостей ГЕ «Ємність».
7. Завдання властивостей ГЕ «Конус».
8. Завдання властивостей ГЕ «Насос».
9. Завдання властивостей ГЕ «Труба».
10. Завдання властивостей ГЕ «Багатокутник».

Комп’ютерний практикум № 7

ПРОГРАМУВАННЯ НА МОВАХ ST ТА FBD

Мета роботи – засвоєння послідовності програмування на мовах ST та FBD в SCADA-системі.

1 ПРОГРАМА РОБОТИ

Доповнити проект попередньої роботи № 6 засобами для розрахунку та відображення значення вартості, поточних витрат, сумарних витрат, сумарної вартості рідини та періоду генерації сигналів, однакового для усіх каналів.

1.1 Відкрити проект, створений у роботі № 6, за допомогою меню **File\Open...** (Файл\Відкрити...) або кнопки **Open existing project** (Відкрити існуючий проект) на інструментальній панелі ICP. Зберегти проект з новим ім’ям шляхом вибору в меню **File** (Файл) пункту **Save As...** (Зберегти як...), задавши з клавіатури бажане ім’я, наприклад **Robota_7**.

1.2 Вилучити з проекту **Generators_1**. Виділити у навігаторі проекту елемент **Generators_1**, ПК викликати контекстне меню та за допомогою пункту **Erase** (Вилучити) видалити його (рис. 1).

1.3 Викликати ПК на елементі **Screen_#1:1** (Екран_#1:1) у правому вікні навігатора проекту контекстне меню та вибрати рядок **Edit** (Редагувати). У вікні редагування каналу встановити у полі **Recalc. Period** (Період перерахунку) значення **1**, а в полі **Mes. Unit** (Одиниця вимірювання) вибрати **s** (секунда) (рис. 2).

1.4 Розкрити дерево вузла **RTM_1** у лівому вікні навігатора проекту. Викликати ПК на елементі **Channels** (Канали) контекстне меню та вибрати рядок **Create Component** (Створити компоненту), а потім **FLOAT_Channel** (рис. 3).

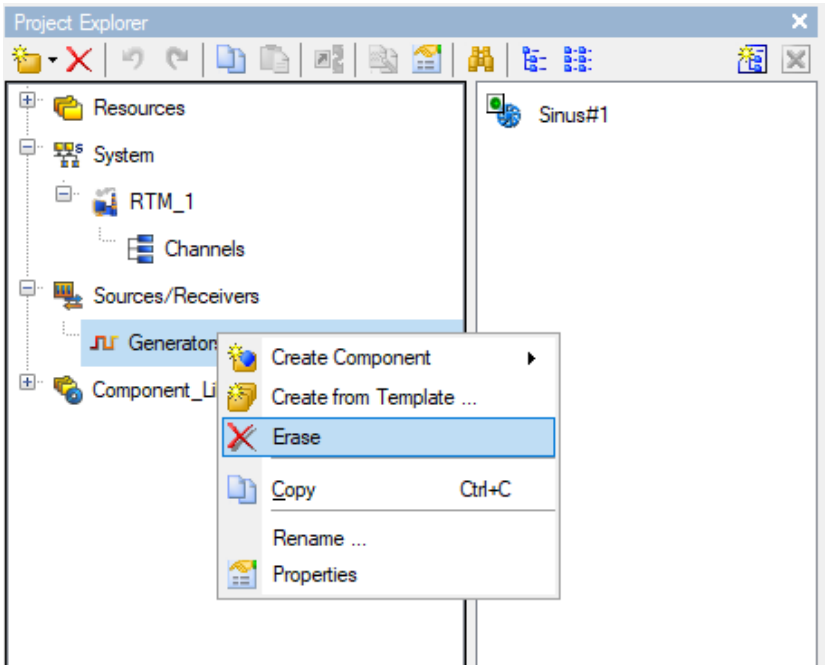


Рисунок 1 – Видалення з проекту елемента **Генератор**

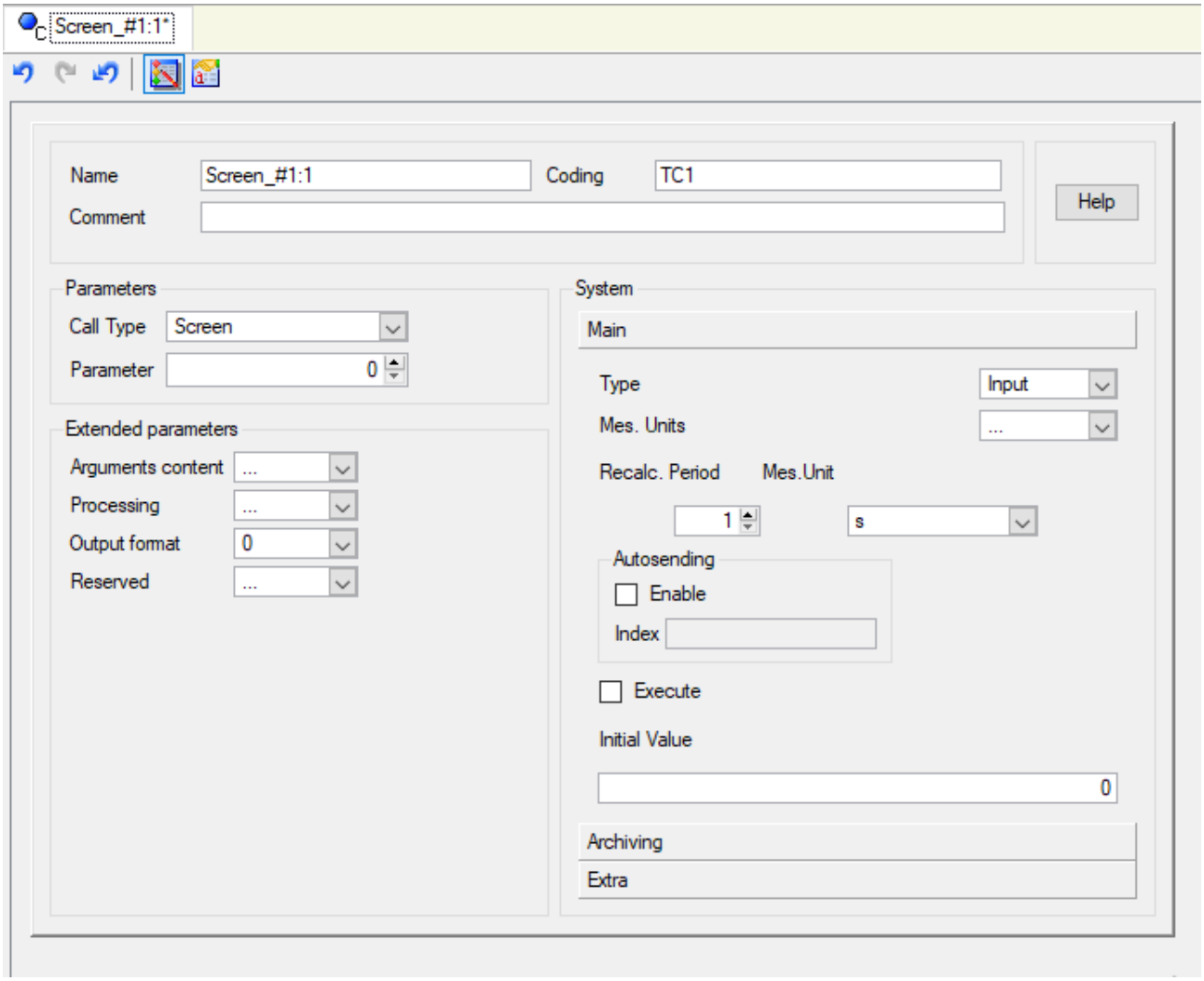


Рисунок 2 – Редагування атрибутів екрану

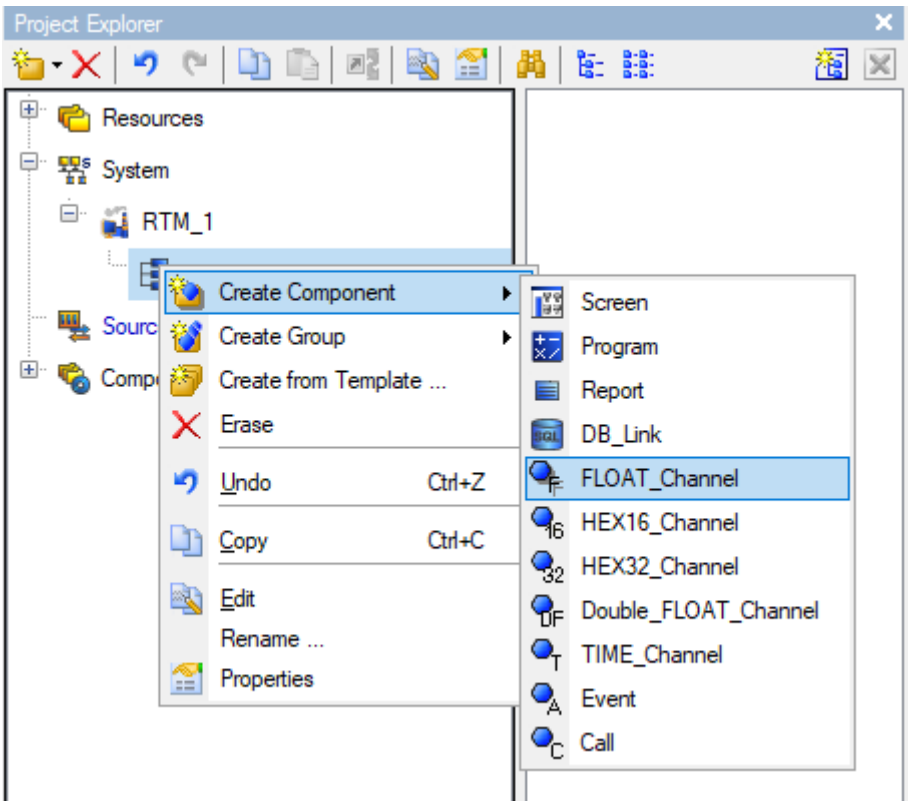


Рисунок 3 – Створення каналу типу **FLOAT**

У правій частині навігатора подвійним ЛК на створеному каналі відкрити вікно властивостей каналу та у рядку **Name** (Ім’я) задати з клавіатури нове ім’я **Вартість** для зберігання значень вартості рідини. Встановити тип каналу **Input** та період перерахування значення, рівний 1 секунді (рис. 4).

Послідовно аналогічно створити ще 5 каналів для зберігання значень витрат рідини, періоду генерації, сумарних витрат рідини, сумарної вартості рідини, рівня рідини у ємності, задавши з клавіатури відповідні імена: **Витрати, Період_генерації**, **Сумарна_вартість, Сумарні_витрати, Рівень** (ВАЖЛИВО: в іменах не використовувати символ «пропуск»). Встановити для всіх каналів тип **Input** та однаковий період перерахування значення, рівний 1 секунді. В результаті вікно навігатора проекту зі створеними каналами буде мати вигляд, як на рис. 5.

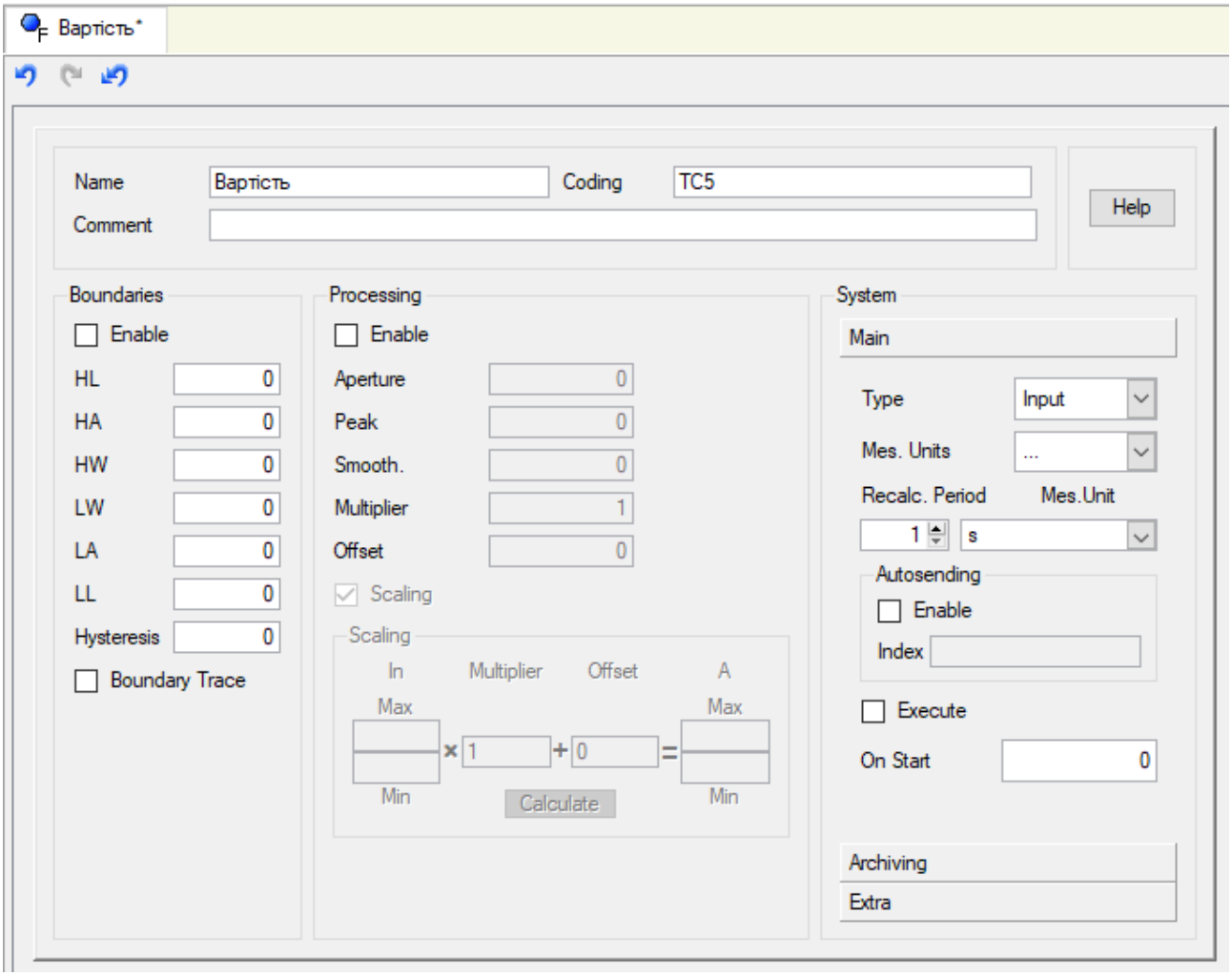


Рисунок 4 – Завдання параметрів каналу **Вартість**

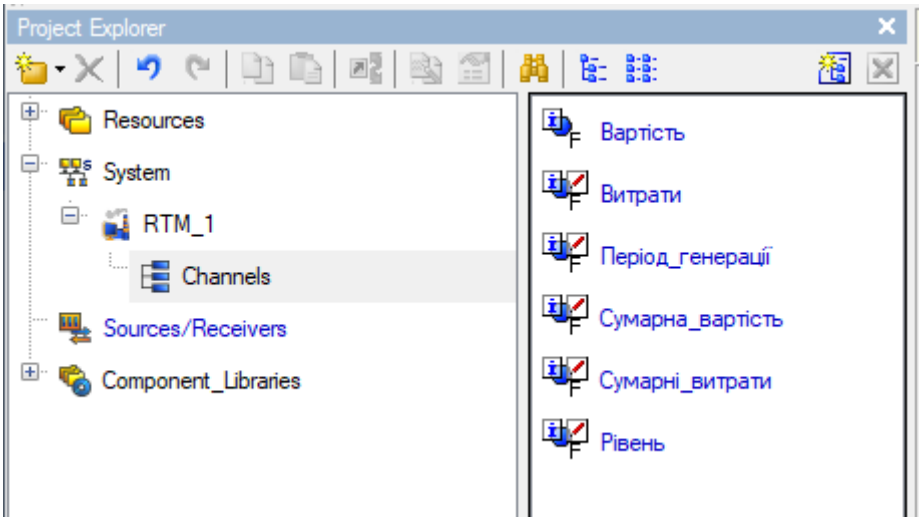


Рисунок 5 – Вікно навігатора проекту зі створеними каналами

1.5 Створити кнопки для завдання значень параметрів системи. Викликати графічний екран **Screen_#1:1** на редагування. Вибрати ЛК на інструментальної панелі графічного редактора екрана іконку одиночної кнопки **Button** та розмістити три ГЕ **Button** (Кнопка), як показано на рис. 6. Виділити при натиснутій ЛК ці кнопки та вирівняти їх по вертикалі, встановити однакові проміжки між ними тощо, за допомогою пунктів меню **Tools** (Інструменти).

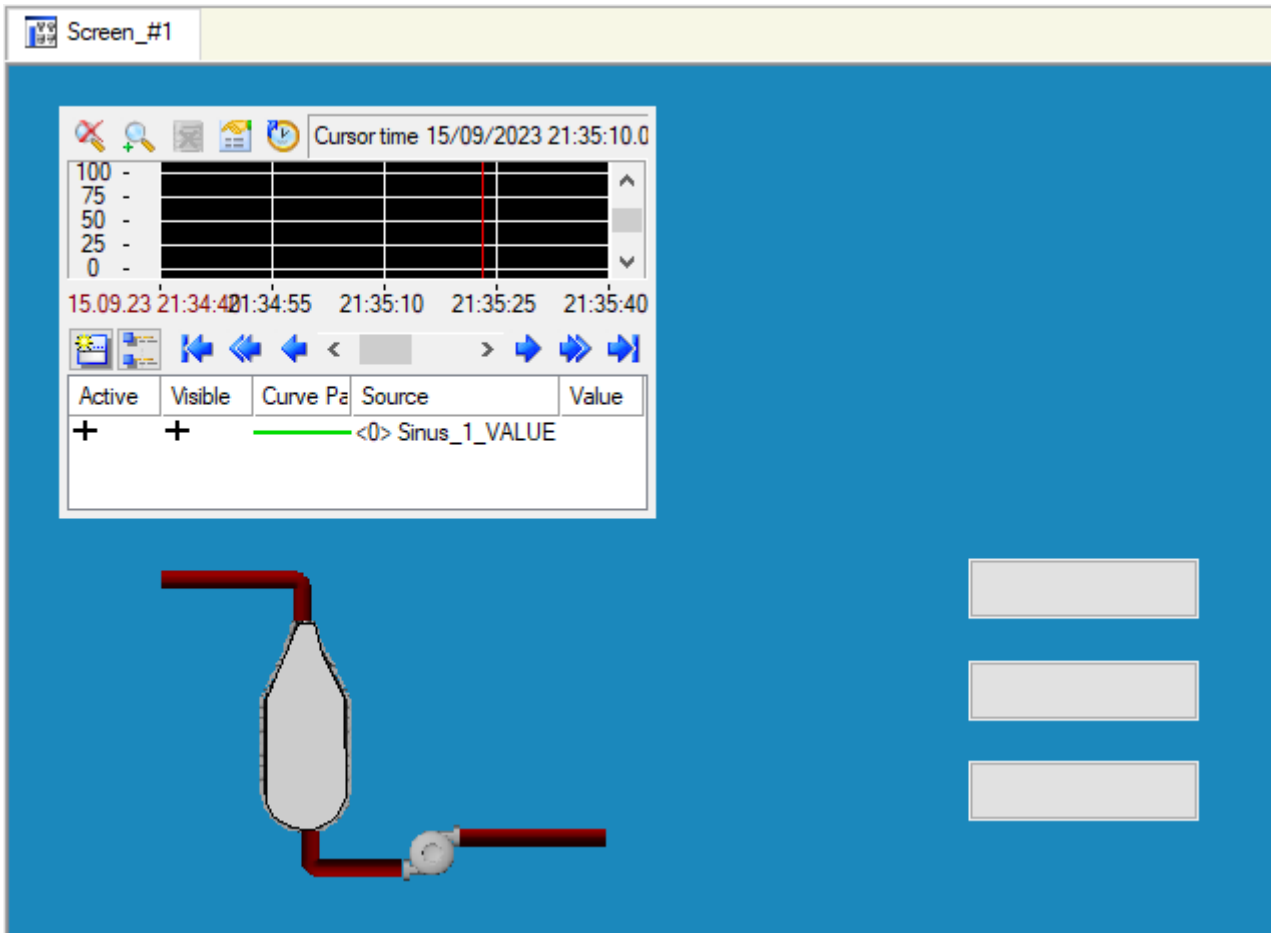


Рисунок 6 – Розташування кнопок на екрані

Подвійним натисканням ЛК на верхньому ГЕ **Кнопка** відкрити вікно його властивостей. У рядок праворуч від властивості **Text** ввести з клавіатури напис **Встановити період**. Відкрити середню закладку **Actions** (Дії), а потім ПК розкрити меню **MousePress** та вибрати зі списку команду **Send Value** (Передати значення) (рис. 7).

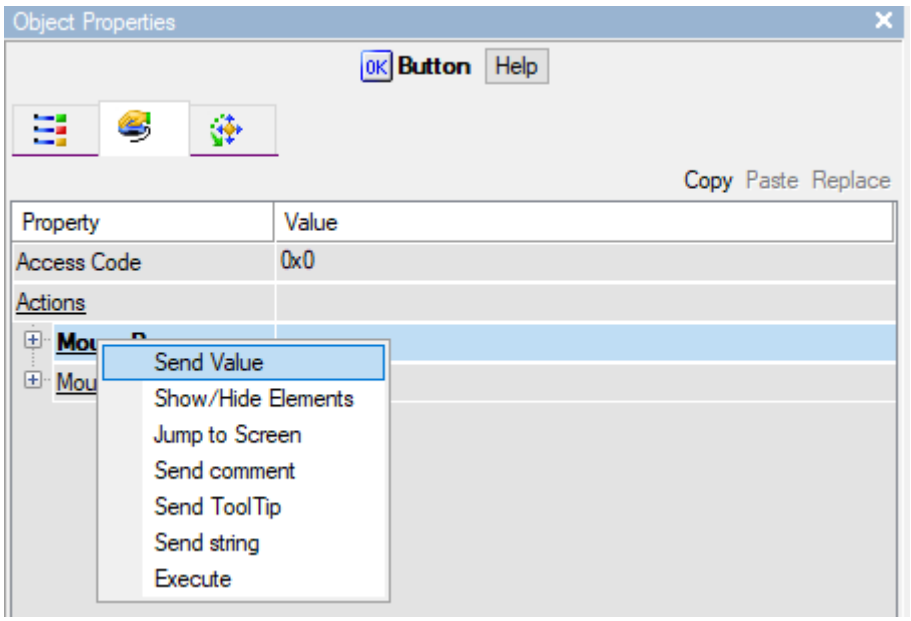


Рисунок 7 – Вікно закладки **Дії**

Праворуч у полі **Send Type** (Тип передачі) вибрати зі списку **Enter & Send** (Ввести і передати) (рис. 8).

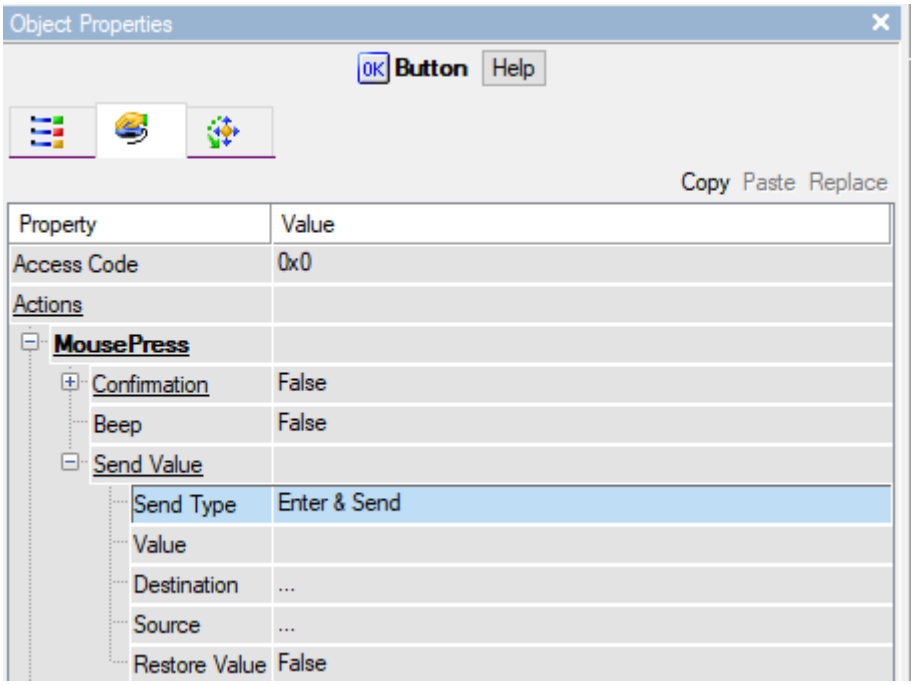


Рисунок 8 – Настроювання графічного елемента **Кнопка**

ЛК праворуч у полі **Destination** (Результат) викликати табличний редактор аргументів. Вилучити існуючий з попередньої роботи аргумент

Sinus_1 за допомогою кнопки **Delete Argument**. Створити новий аргумент і задати йому ім'я **Період_генерації**. Змінити тип аргументу на **OUT**. Подвійним ЛК у полі **Link** (Прив’язка) відкрити вікно, а потім розкрити ліворуч дерево **Channels** та вибрати канал **Період_генерації**. Праворуч у закладці **Attributes** вибрати ЛК **Input Value** (Вхідне значення) і послідовно кнопками **Link** та **Ok** підтвердити прив'язку атрибута ГЕ до цього аргументу (рис. 9).

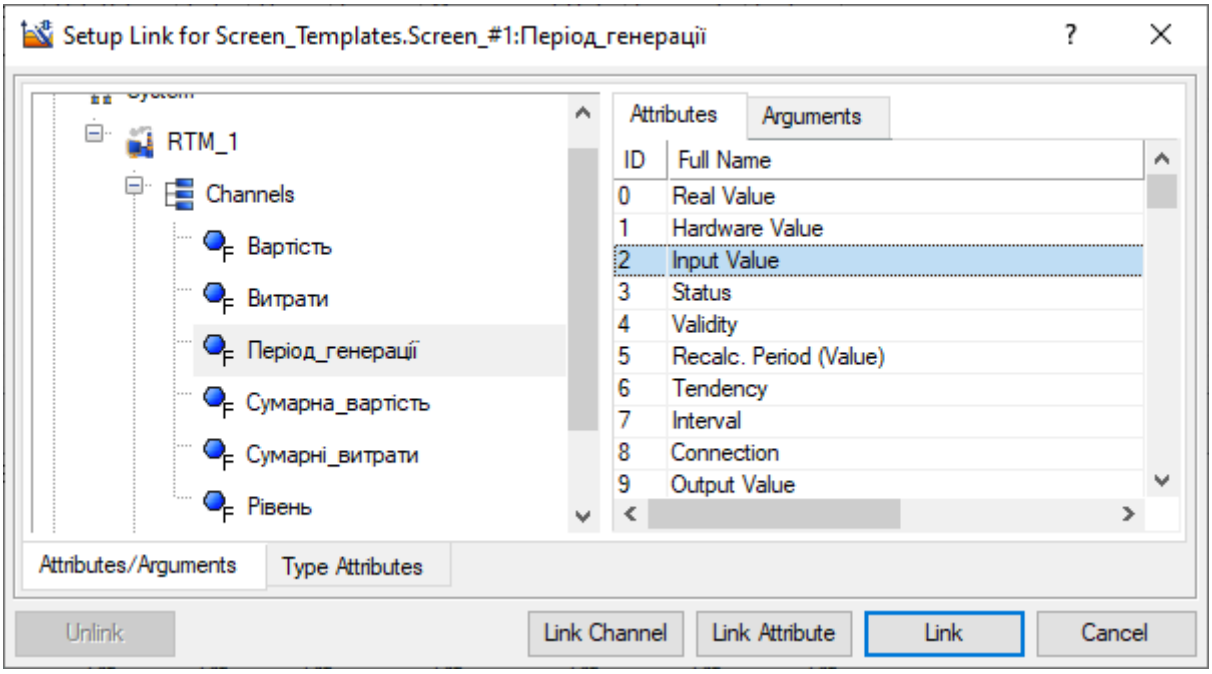


Рисунок 9 – Настроювання прив’язки до каналу елемента **Кнопка**

Аналогічним чином назвати **Встановити витрати**, **Встановити вартість** дві інші кнопки, створити для них аргументи з іменами **Витрати**, **Вартість** відповідно та виконати прив’язки. В результаті вікно прив’язок буде мати вигляд, як показано на рис. 10.

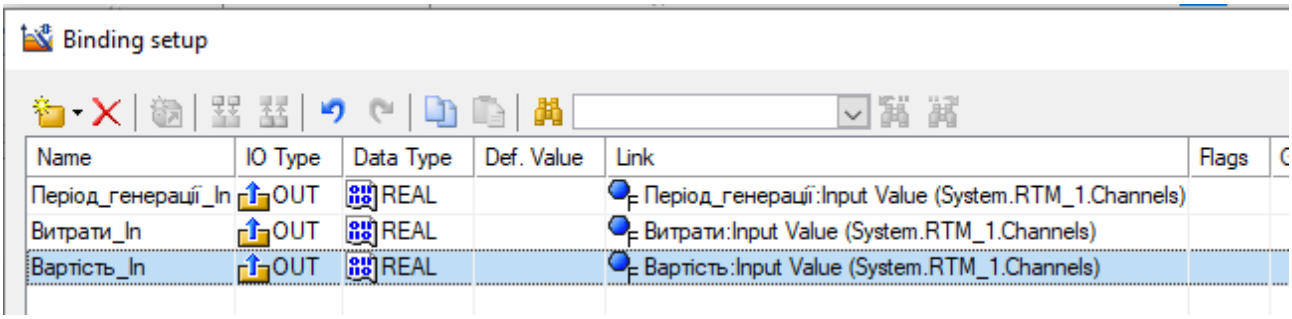


Рисунок 10 – Вікно прив’язок зі створеними аргументами

1.6 Розмістити на екрані десять ГЕ **Text** (Текст) для відображення чисельних значень п’яти змінних. Послідовно для кожного ГЕ лівого стовпця у вікні властивостей у полі **Text** ввести назву змінних:

Період генерації, с:

Витрати, л/с:

Вартість, грн/л:

Сумарні_витрати, л:

Сумарна_вартість, грн:

як показано на рис. 11.

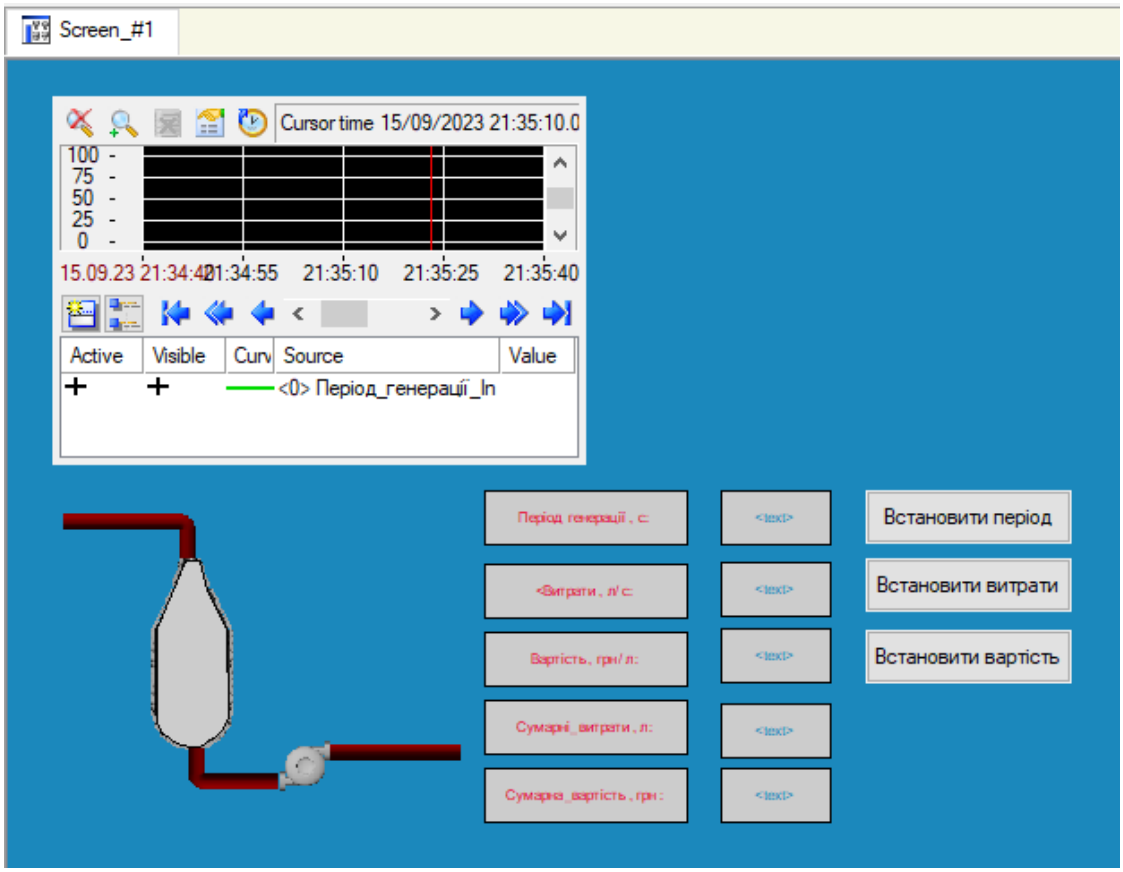


Рисунок 11 – Розташування на екрані ГЕ **Текст**

Послідовно для кожного ГЕ **Text** правого стовпця подвійним натисканням ЛК на слові **Text** відкрити меню **Indication type** (Вид індикації) (рис. 12). У правому полі відкритого рядка натисканням ЛК із списку доступних типів вибрати **Value** (Значення) (рис. 13). Після натискання ЛК у правому полі

Binding (Прив’язка) з’являється вікно **Binding setup** (Властивості прив’язки). Послідовно створити аргументи екрана шляхом натискання кнопки **Create Argument** на інструментальній панелі вікна. Аргументам дати імена **Період_генерації**, **Витрати, Вартість, Сумарні_витрати, Сумарна_вартість** та встановити тип **IN**. Подвійним ЛК у полі **Link** відкрити вікно, а потім розкрити ліворуч дерево **Channels** та вибрати відповідний канал. Праворуч у закладці **Attributes** вибрати ЛК **Real Value** (Реальне значення) та послідовно кнопками **Link** та **Ok** підтвердити прив'язку атрибута ГЕ до цього аргументу. В результаті вікно прив’язок буде мати вигляд, показаний на рис. 14.

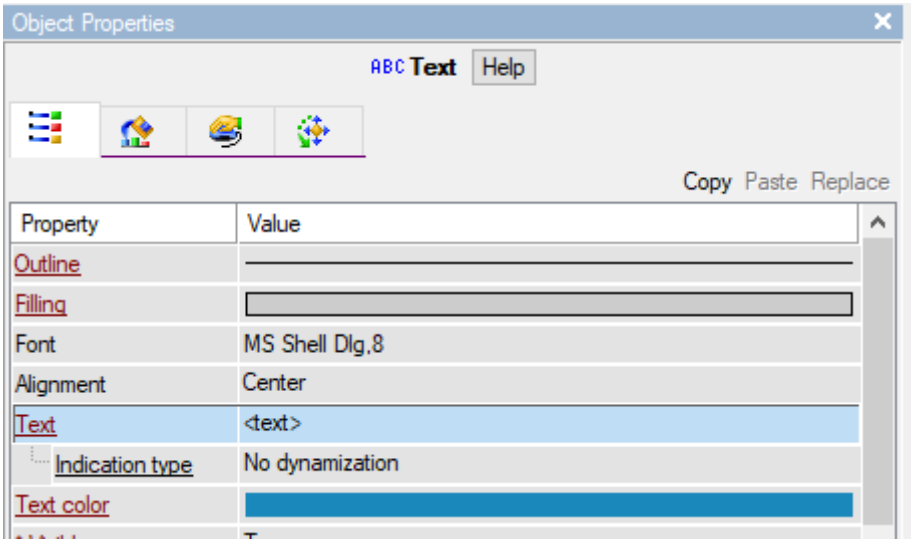


Рисунок 12 – Налаштування динамізації ГЕ **Текст**

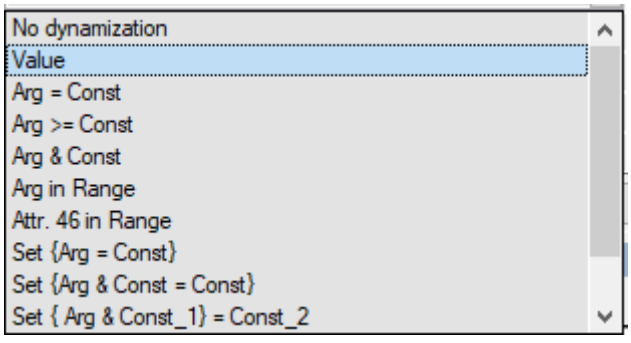


Рисунок 13 – Вибір типу динамізації

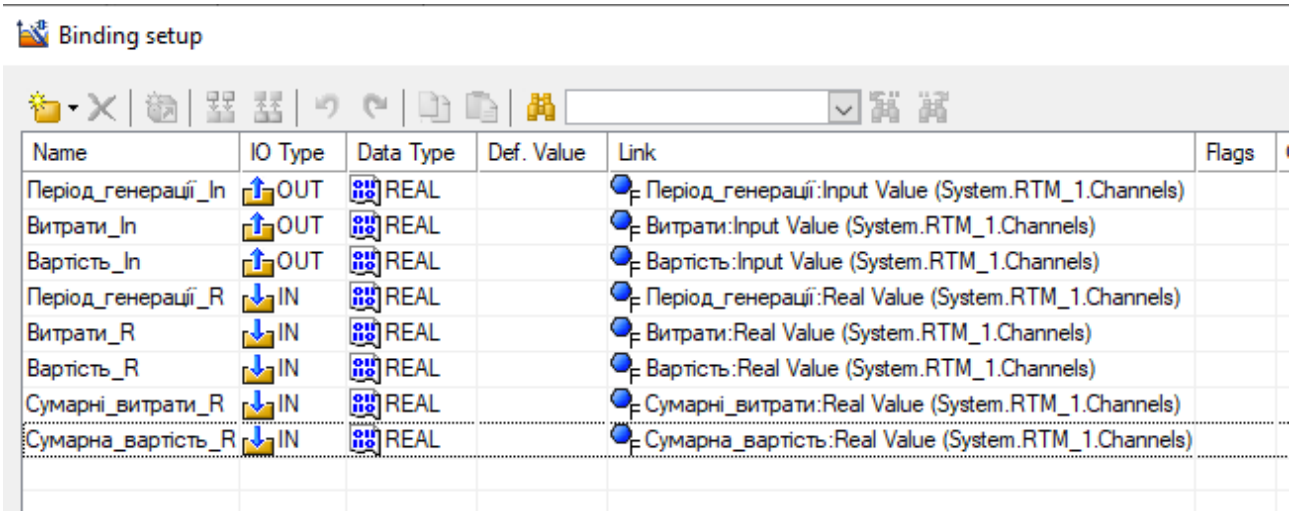


Рисунок 14 – Вигляд вікна прив’язок

1.7 Створити програму мовою **FBD** для імітації сигналу рівня рідини у ємності. Подвійним натисканням ЛК відкрити вузол **RTM_1** та ПК створити в ньому компоненту **Program** (Програма) (рис. 15).

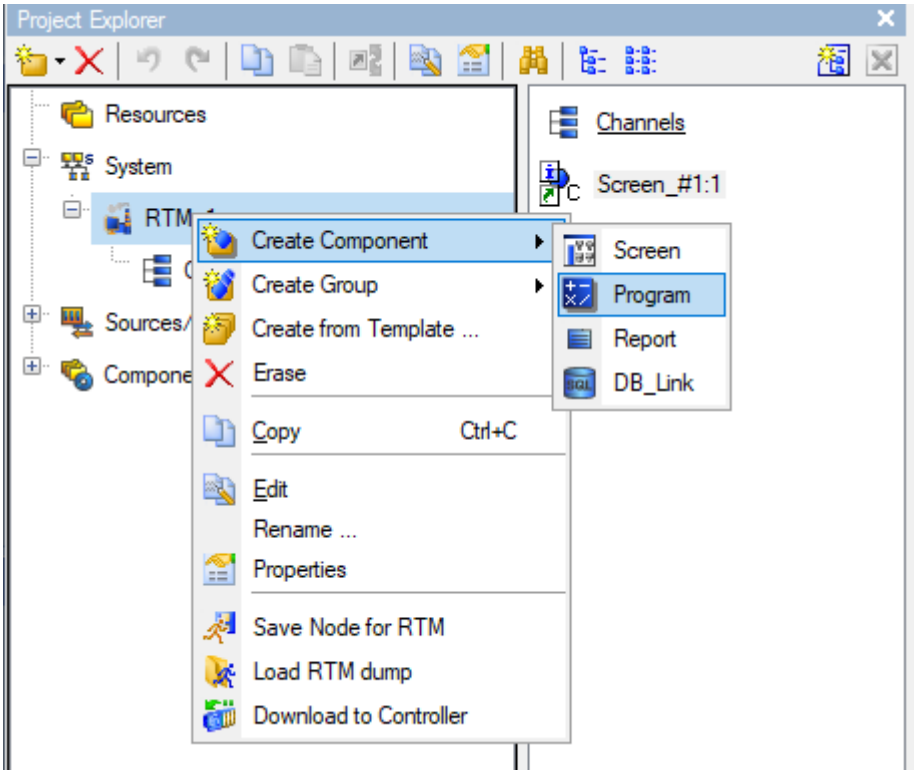


Рисунок 15 – Створення компоненти **Програма**

Виділити створену компоненту **Program_#1:8** (Програма_#1:8) та ПК викликати контекстне меню, вибравши в ньому ЛК пункт **Edit** (Редагувати). У відкритому вікні, аналогічно до рис. 4, змінити ім’я програми на **Синусоїда** та встановити період перерахування значення, рівний 1 секунді.

Виділити програму **Синусоїда** у вікні навігатора та ПК викликати контекстне меню, вибравши в ньому ЛК пункт **Edit Template** (Редагувати шаблон) (рис. 16). Виділити ЛК у дереві шаблону **Program_#1** рядок **Arguments** (Аргументи) та викликати табличний редактор аргументів. За допомогою кнопки створити послідовно два аргументи **Період_генерації** та **Рівень**. При цьому першому аргументу встановити тип **IN/OUT**, а другому – **OUT**. Подвійним ЛК у полі **Link** відкрити вікно, а потім розкрити ліворуч дерево **Channels** та вибрати відповідний канал. Праворуч у закладці **Attributes** вибрати ЛК **Real Value** (Реальне значення) та послідовно кнопками **Link** та **Ok** підтвердити прив'язку атрибута. В результаті вікно прив’язок буде мати вигляд, показаний на рис. 17.

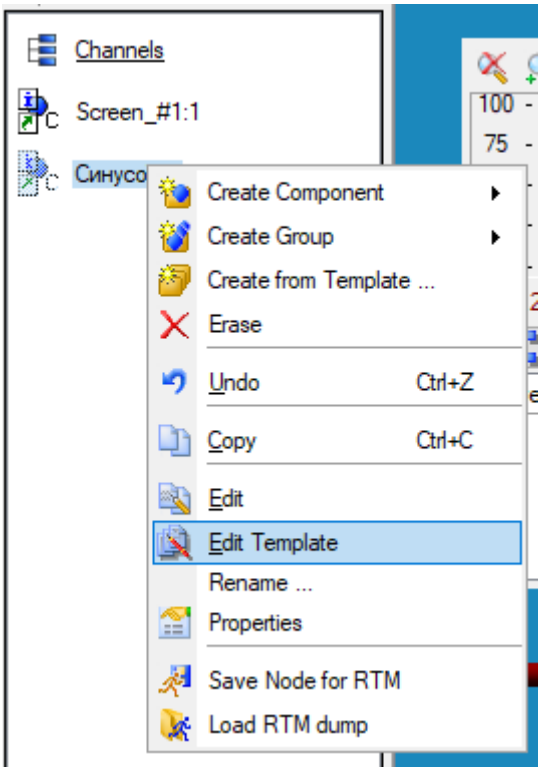


Рисунок 16 – Редагування аргументів програми

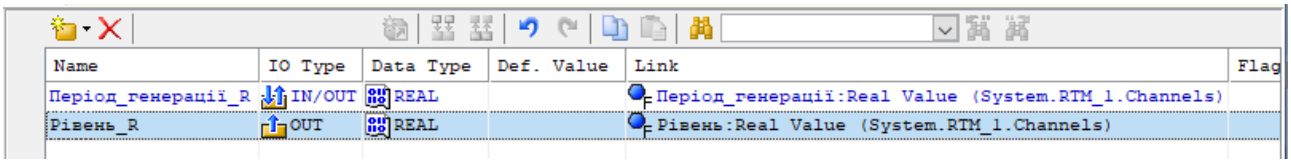


Рисунок 17 – Аргументи **FBD**-програми

Виділити ЛК в дереві шаблону рядок **Program_#1** та у вікні, що відкрилося, вибрати мову програмування **Function Block Diagram** (FBD-діаграма) (рис. 18).

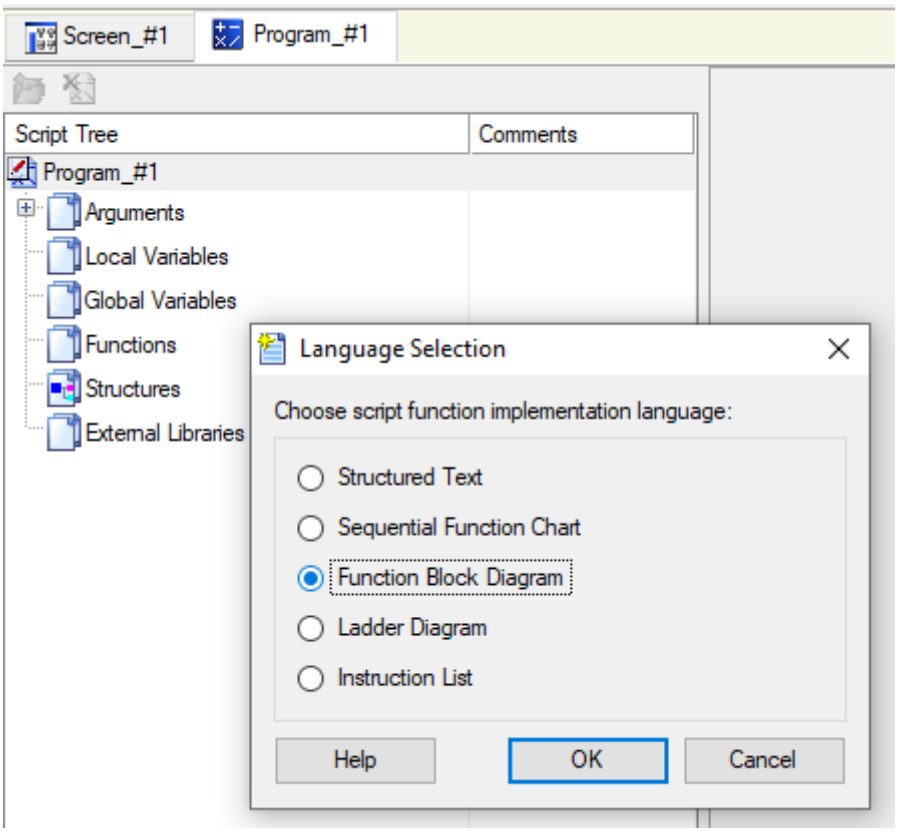


Рисунок 18 – Вибір мови програмування **FBD**

Після натискання кнопки **Ok** (Приняти) створити програму у вікні редактора програм з урахуванням наступних вимог:

1) Сигнал рівня рідини повинен бути синусоїдного типу в межах [0; +1]. Так як FBD-блоки генерують сигнали в межах [-1; +1], то потрібно провести масштабування сигналу за формулою

$$Y = kX + Z$$

Y – результат перетворення;

X – вхідне значення каналу;

$k=0,5$ – множник;

$Z=0,5$ – зсув.

2) Якщо період генерації буде менше **10 с**, то програма повинна встановити період рівний **10 с**.

Для виклику палітри FBD-блоків необхідно ЛК натиснути на кнопку , після чого з'явиться вікно вибору блоків. Для зручності користування вікно FBD-блоків за допомогою ЛК можна розмістити у бажаному місці на екрані та змінити його розміри. Для реалізації програми із закладки **Comparison** (Порівняння) за допомогою натиснутої ЛК переміщується на поле програми блок **Equal or More** (Більше або рівно) (рис. 19), із закладки **Selection** (Вибір) блок **Selection out of Two** (Вибір із двох) (рис. 20), із закладки **Generators** (Генератори) блок **Sinusoidal Signal** (Синусоїдний сигнал) (рис. 21), із закладки **Arithmetic** (Арифметичні) блок **Scaling** (Масштабування) (рис. 22).

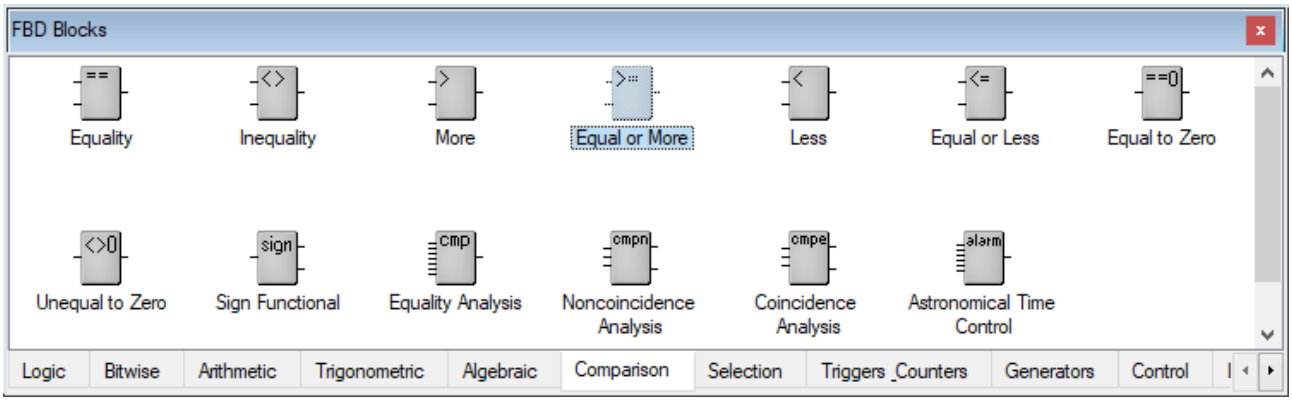


Рисунок 19 – Палітра FBD-блоків **Порівняння**

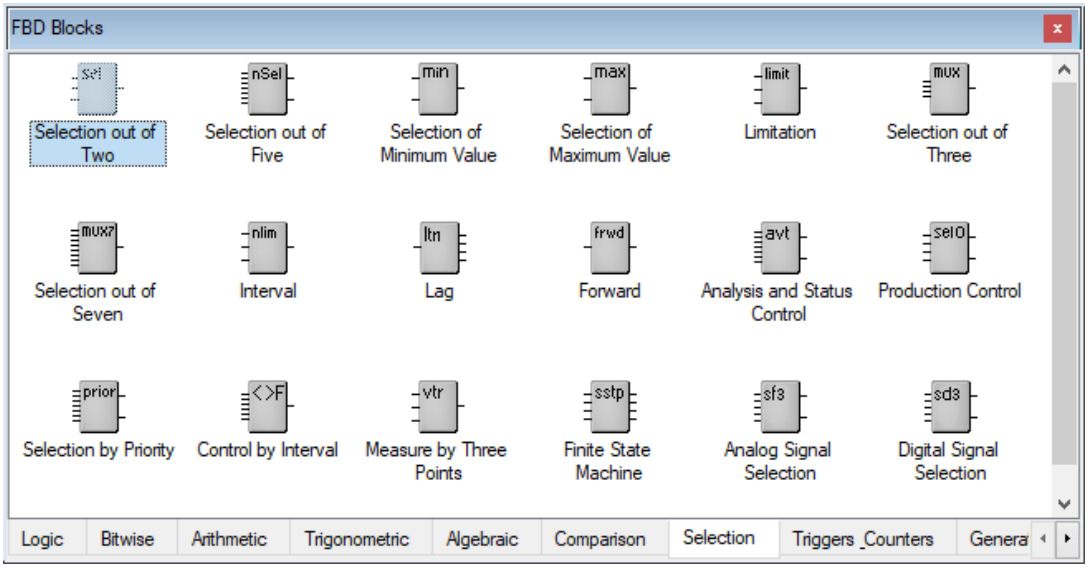


Рисунок 20 – Палітра FBD-блоків **Вибір**

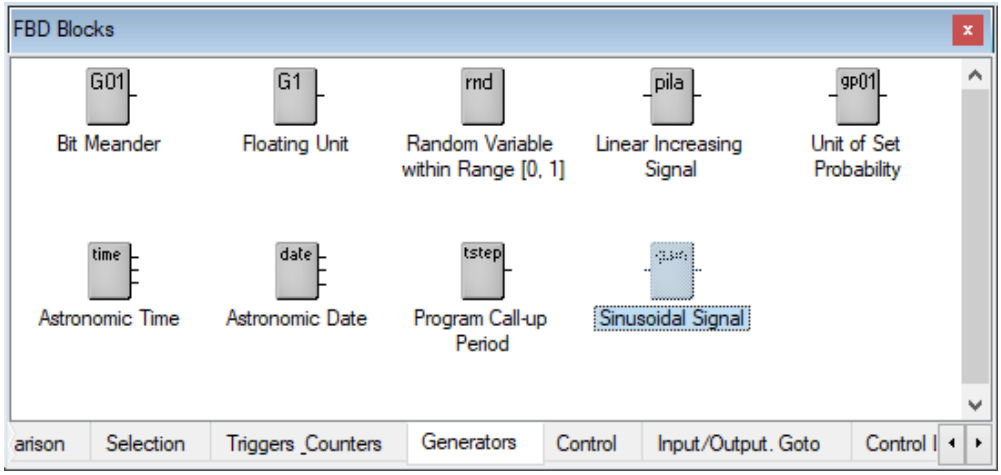


Рисунок 21 – Палітра FBD-блоків **Генератори**

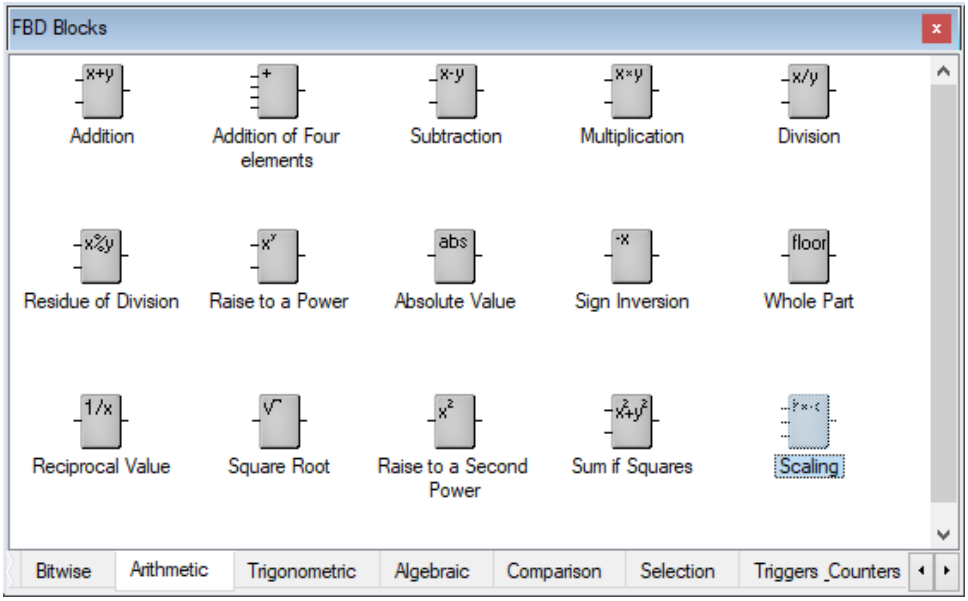


Рисунок 22 – Палітра FBD-блоків **Арифметичні**

Розробка програми на основі FBD-блоків полягає у з'єднанні їх входів/виходів, подача вхідних та вихідних сигналів. Треба мати на увазі, що верхні входи FBD-блоків не використовуються, а інформаційними є входи, починаючи із другого. З'єднання входів здійснюється при натиснутій ЛК. Подача та вибір сигналу здійснюється після подвійного натискання ЛК на контакті блоку. Можна використовувати також кнопки панелі інструментів вгорі вікна або контекстне меню після ПК на виділеному блоці. Після виконання всіх операцій програма буде мати вигляд, показаний на рис. 23.

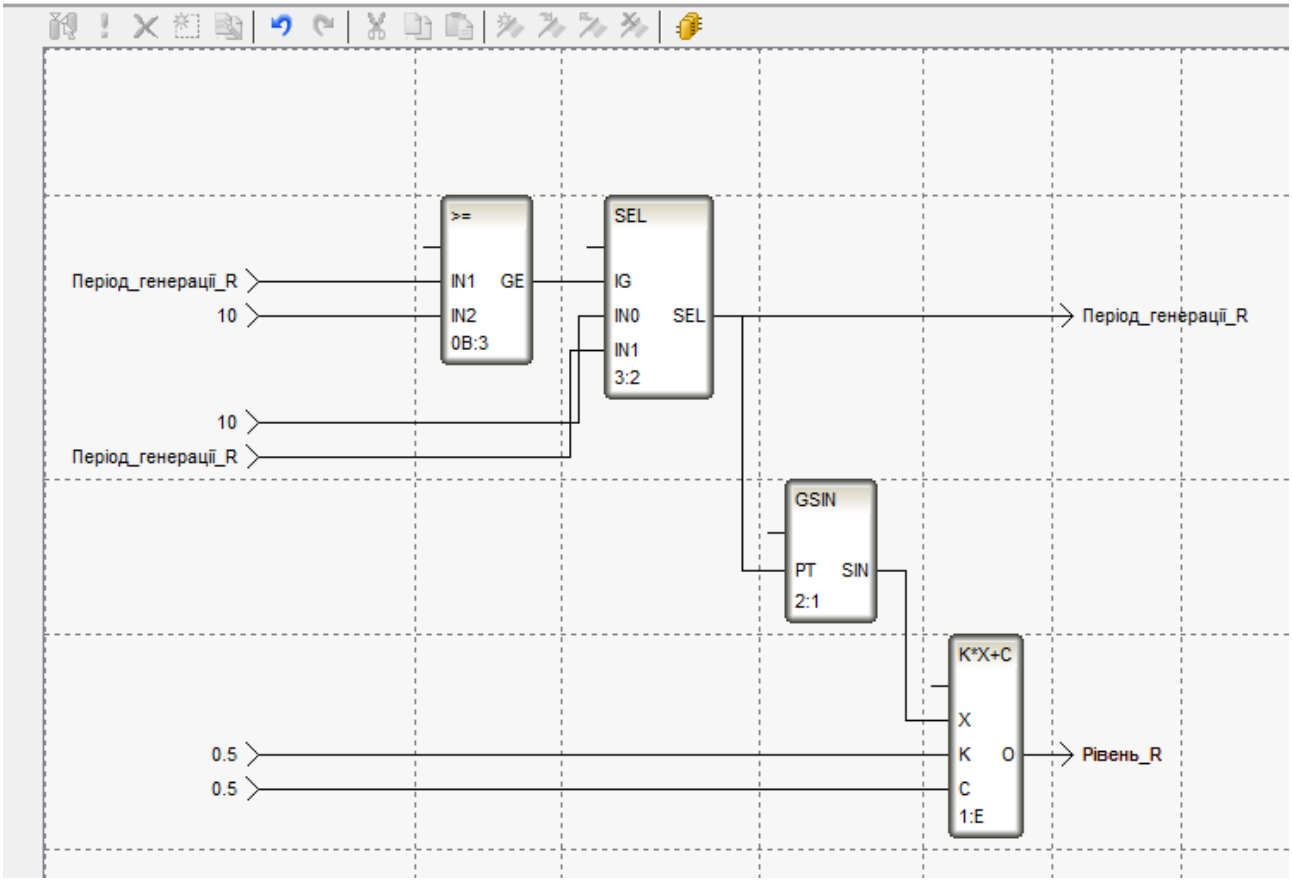


Рисунок 23 – Програма мовою **FBD**

За допомогою іконки **Build (F7)** (Компіляція) на інструментальній панелі редактора або гарячої клавіші F7 скомпілювати програму та переконатися в успішній компіляції у вікні **Messages**, яке при необхідності

викликається з інструментальної панелі за допомогою іконки **Messages** (Повідомлення) (рис. 24).

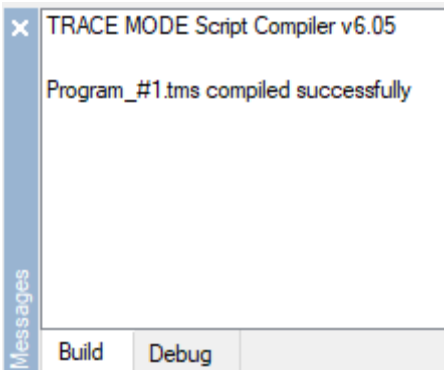


Рисунок 24 – Результат успішної компіляції програми

1.8 Створити програму мовою **ST** для підрахунку сумарних витрат та сумарної вартості рідини**.** Подвійним натисканням ЛК відкрити вузол **RTM_1** та створити в ньому компоненту **Program** (Програма)**,** як у попередньому пункті. Виділити компонент **Program_#2:9** (Програма_#2:9) та ПК викликати контекстне меню, вибравши в ньому ЛК пункт **Edit** (Редагувати). Змінити ім'я програми на **Розрахунок** та встановити період перерахування значення, рівний 1 секунді.

Виділити програму **Розрахунок** у вікні навігатора та ПК викликати контекстне меню, вибравши в ньому ЛК пункт **Edit Template** (Редагувати шаблон). Виділити ЛК у дереві шаблону **Program_#2** рядок **Arguments** та викликати табличний редактор аргументів. За допомогою кнопки створити послідовно п'ять аргументів **Витрати, Вартість, Період, Сумарні_витрати, Сумарна_вартість.** При цьому першим двом аргументам встановити тип **IN/OUT**, третьому – **IN,** а наступним двом – **OUT**. Подвійним ЛК у полі **Link** відкрити вікно, розкрити ліворуч дерево **Channels,** вибрати відповідний канал та здійснити прив'язку, задавши аргументи, як на рис 25.

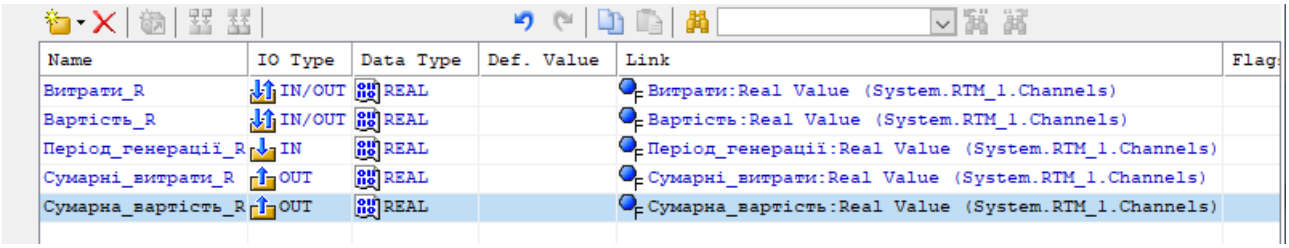


Рисунок 25 – Аргументи **ST**-програми

Виділити ЛК у дереві шаблону **Program_#2** рядок **Global Variables** (Глобальні змінні) та викликати табличний редактор аргументів. За допомогою кнопки створити глобальну змінну **Попередні_сумарні_витрати_R,** яка буде зберігати значення сумарних витрат, обчислених при попередньому виклику програми, задавши у полі **Initial Value** (Початкове значення) нульове значення (рис. 26).

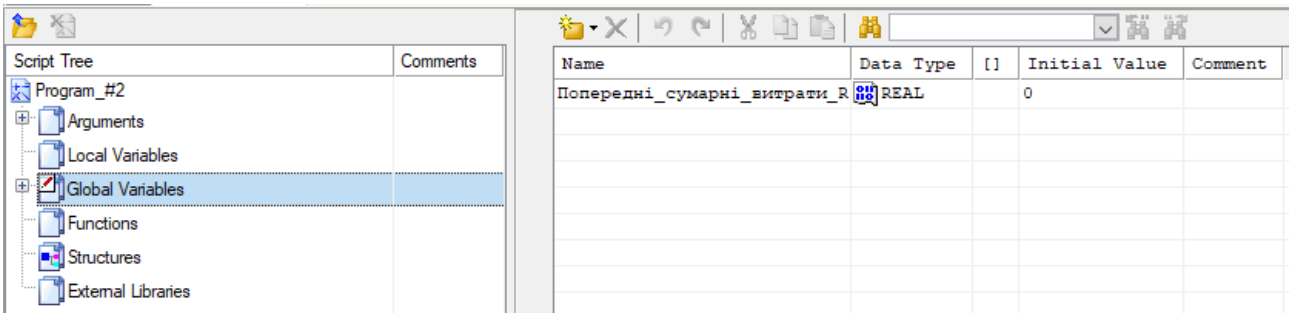


Рисунок 26 – Створення глобальної змінної

Виділити ЛК в дереві шаблону рядок **Program_#2** та у вікні, що відкрилося, вибрати мову **Structured Text** (ST-програма) (рис. 27).

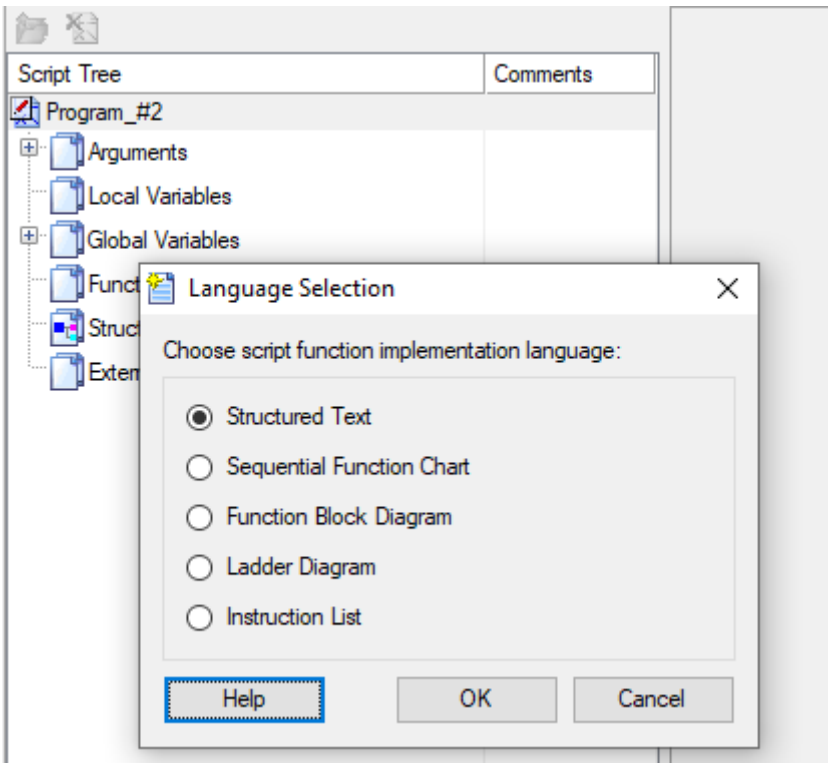


Рисунок 27 – Вибір мови програмування **ST**

Вікно шаблона програми із створеними вище іменами аргументів буде мати вигляд, показаний на рис. 28.

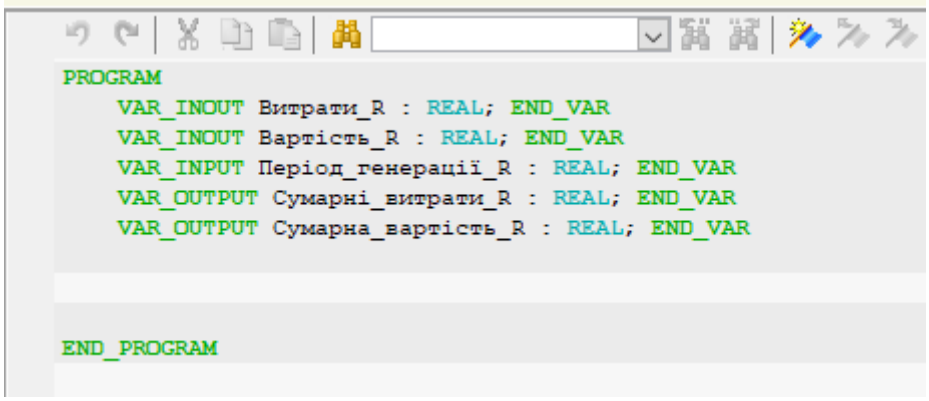


Рисунок 28 – Вікно програми на мові **ST**

У виконавчому блоці написати програму для розрахунку сумарних витрат та сумарної вартості.

Сумарні витрати обчислюються за формулою:

Сумарні_витрати = Попередні_сумарні_витрати + Витрати*Період

Сумарна вартість знаходиться за наступним виразом:

Сумарна_вартість = Сумарні_витрати*Вартість

Причому, якщо значення вартості або витрат буде менше одиниці, то вони приймаються рівними одиниці. Результуюча програма має вигляд, як показано на рис. 29.

```
PROGRAM
    VAR_INOUT Витрати_R : REAL; END_VAR
    VAR_INOUT Вартість_R : REAL; END_VAR
    VAR_INPUT Період_генерації_R : REAL; END_VAR
    VAR_OUTPUT Сумарні_витрати_R : REAL; END_VAR
    VAR_OUTPUT Сумарна_вартість_R : REAL; END_VAR

if Витрати_R < 1 then Витрати_R=1;
End_if;
if Вартість_R < 1 then Вартість_R=1;
End_if;
Сумарні_витрати_R = Попередні_сумарні_витрати_R + Витрати_R*Період_генерації_R;
Попередні_сумарні_витрати_R=Сумарні_витрати_R;
Сумарна_вартість_R = Сумарні_витрати_R*Вартість_R;

END_PROGRAM
```

Рисунок 29 – Програма на мові **ST**

За допомогою іконки **Build (F7)** (Компіляція) на інструментальній панелі редактора або гарячої клавіші F7 скомпілювати програму та переконатися в успішній компіляції у вікні **Messages**, яке при необхідності викликається з інструментальної панелі за допомогою іконки **Messages** (Повідомлення).

1.9 Прив’язати процес наповнення ємності рідиною до сигналу **Рівень** програми **Синусоїда**. На екрані подвійним ЛК викликати вікно властивостей ГЕ

Polygon (Багатокутник) на зображенні ємності. Перейти на вкладку динамічного заливання **Dynamic Fill** та подвійним ЛК розкрити розділ **Layer** (Шари). У правому полі рядка **Binding** (Прив'язка) ЛК відкрити вікно **Binding setup** (Властивості прив'язки) і за допомогою кнопки **Create Argument** (Створити аргумент) створити новий аргумент **Рівень**, встановивши йому тип **OUT**. У стовпці **Link** (Прив'язка) нового аргументу викликати подвійним натисканням ЛК вікно настроювання зв'язку. Вибрати у лівій частині вікна компоненту **Синусоїда**, а праворуч у закладці **Arguments** вибрати **Рівень** (рис. 30). Кнопкою **Link** підтвердити зв'язок, а потім кнопкою **Ok** закрити вікно властивостей прив'язки, показане на рис. 31.

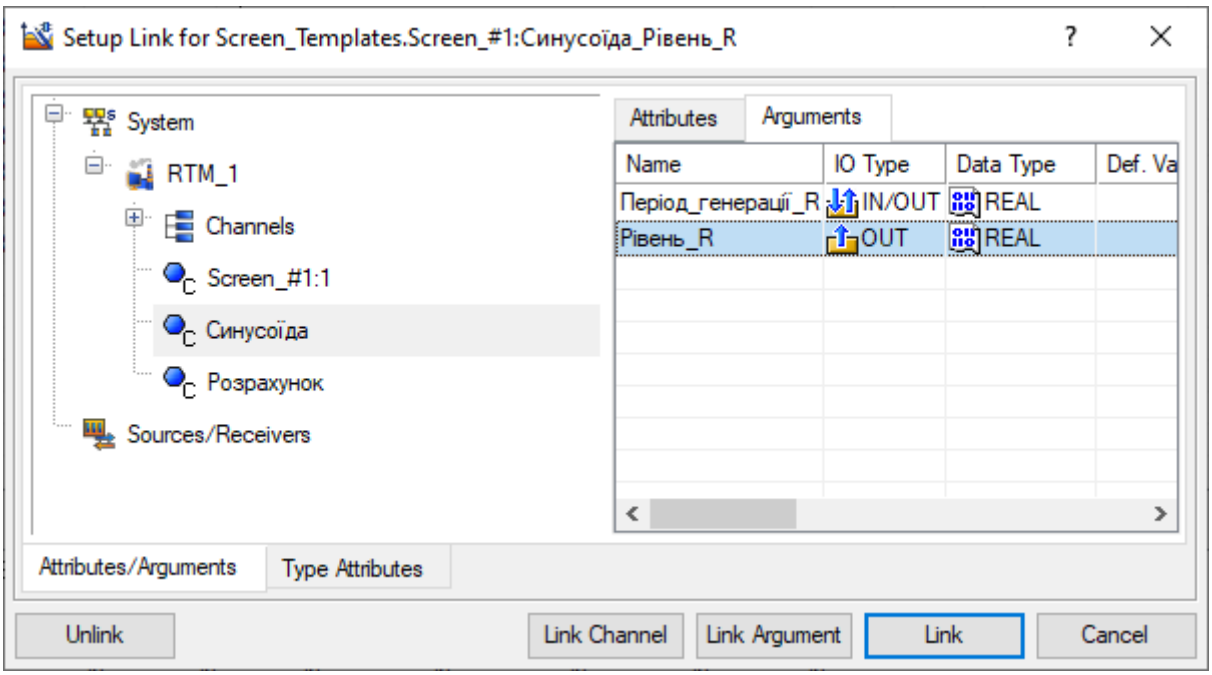


Рисунок 30 – Настроювання прив'язки

У рядку **MAX** на вкладці **Dynamic Fill** динамічного заливання багатокутника **Polygon** встановити значення рівне **1**, що відповідає максимальному значенню сигналу **Рівень** згідно п. 1.7 роботи.

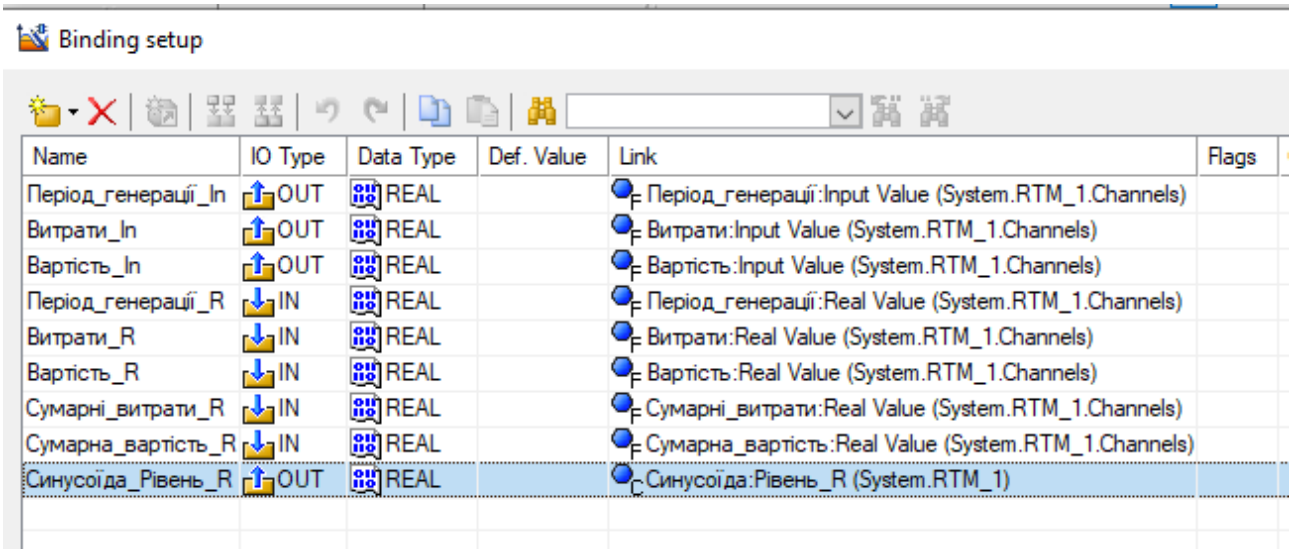


Рисунок 31 – Вигляд вікна прив’язок

1.10 Прив’язати криву ГЕ **Trend** (Тренд) до сигналу **Рівень** програми **Синусоїда** для перегляду зміни рівня рідини в ємності у часі та відстеження передісторії. На екрані подвійним ЛК на ГЕ **Trend** (Тренд) відкрити вікно його властивостей. Перейти на вкладку **Curves** та подвійним ЛК розкрити розділ **Curve1**. У правому полі рядка **Binding** (Прив’язка) ЛК відкрити вікно **Binding setup** та вибрати аргумент **Синусоїда_Рівень_R**. Кнопкою **Ok** закрити вікно властивостей.

У рядку **Max value** вікна розділу **Curve1** встановити значення рівне **1**, що відповідає максимальному значенню сигналу **Рівень** згідно п. 1.7 роботи.

В результаті вікно розділу **Curve1** буде мати вигляд, показаний на рис. 32.

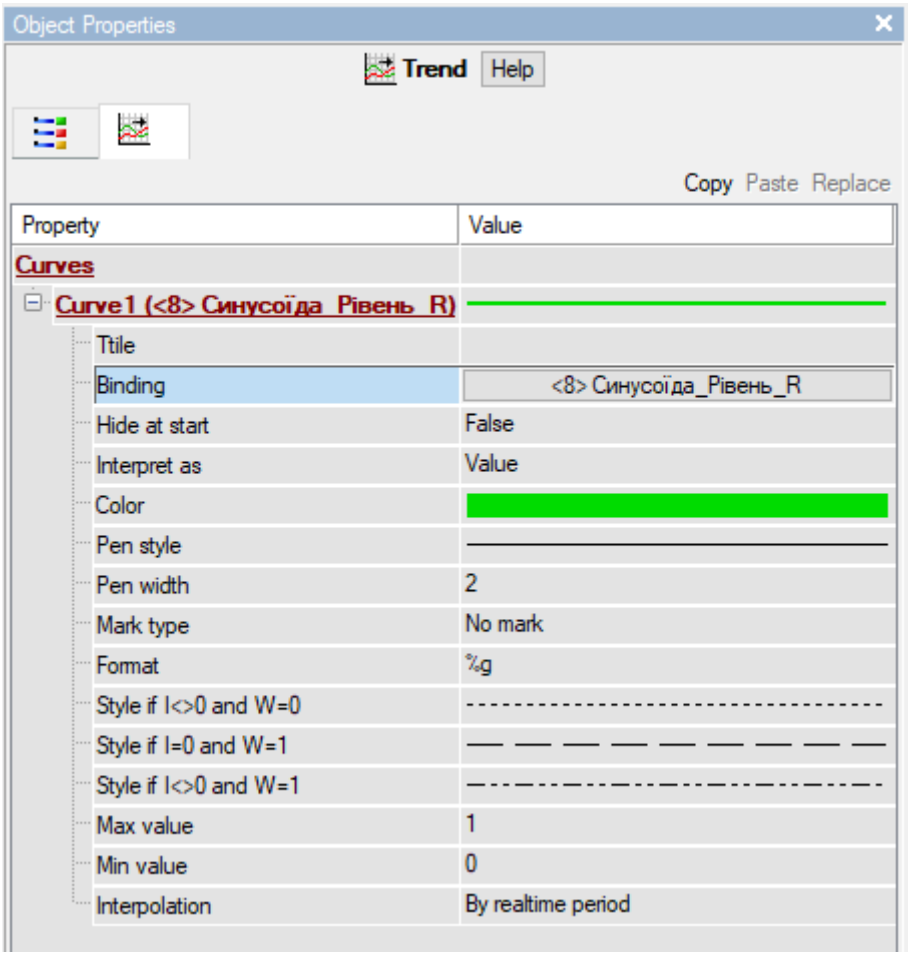


Рисунок 32 – Вигляд вікна властивостей кривої ГЕ **Тренд**

1.11 Вказати (перевірити) у проекті свої дані.

На графічному екрані розмістити у зручному місці шифр групи та своє ПІБ аналогічно до роботи № 1.

На графічному екрані розмістити у зручному місці своє фото.

В меню **File\Information...**(Файл\Інформація...) в рядку **Owner** (Власник) написати за допомогою клавіатури своє ПІБ, а в рядку **Organization** (Організація) – шифр групи та факультету.

1.12 Запустити проект в роботу. Зберегти проект натисканням кнопки . На головній інструментальній панелі ICP натиснути ЛК кнопку **Save for RTM** (Зберегти для МРЧ), скомпілювавши проект для запуску в реальному часі.

Виділити ЛК в навігаторі проекту вузол **RTM_1**, а потім натиснути на головній панелі кнопку **Start Profiler** (Запустить профайлер). У вікні, що відкрилося, запустити проект на виконання шляхом натискання ЛК кнопки **Run** (Пуск/Зупинка).

За допомогою кнопок ввести бажані значення параметрів системи. Спостерігати в ємності та на елементі **Тренд** перехідний процес рівня рідини, а на елементах **Текст** значення сумарних витрат та сумарної вартості, як показано на рис. 33.

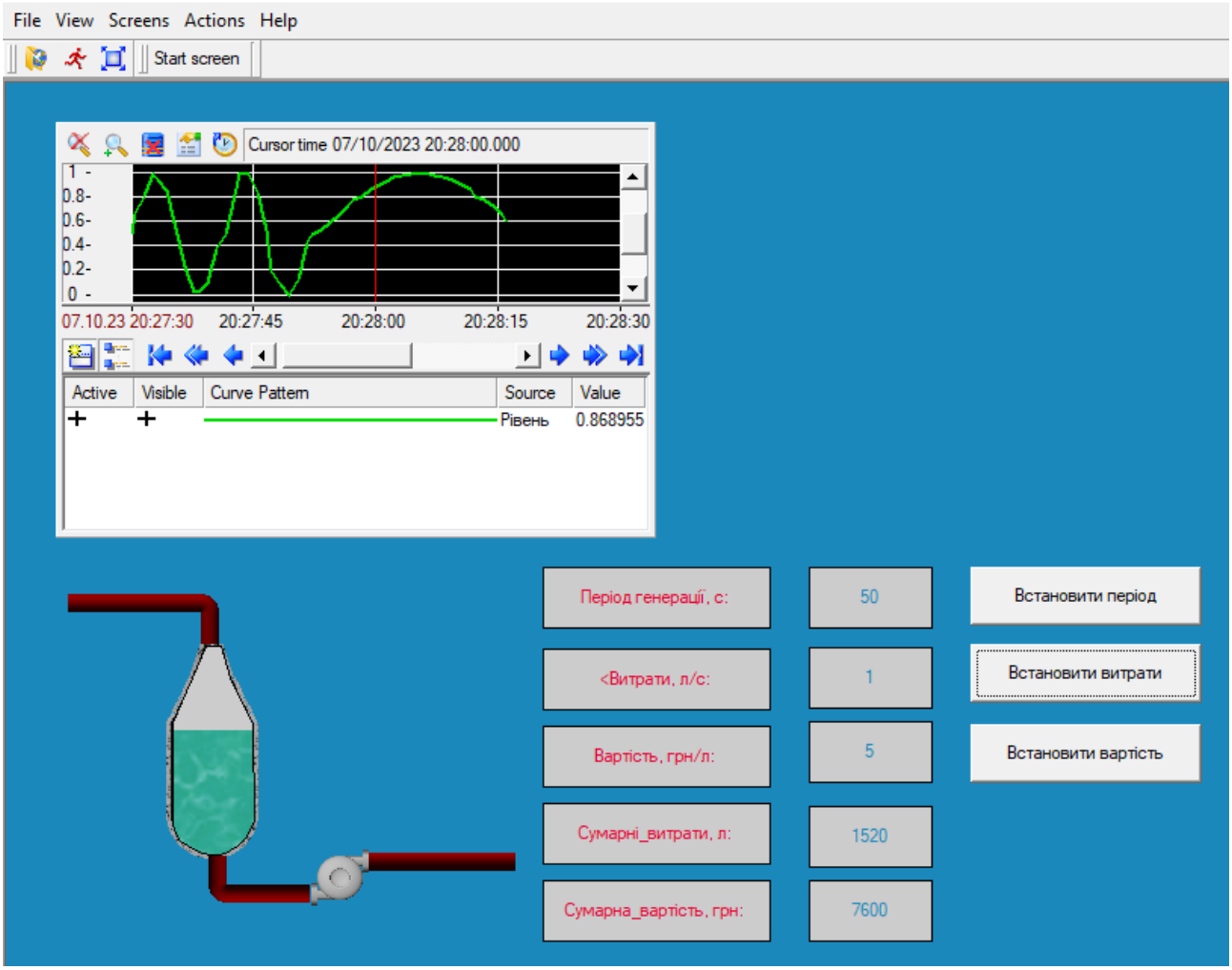


Рисунок 33 – Результат імітаційного запуску проекту

Зупинити виконання проекту шляхом натискання на панелі інструментів ЛК кнопки **Run** (Пуск/Зупинка). Закрити вікно графічного екрану через

меню **File\Exit** (Файл\Вихід) або засобами Windows. Зберегти проект натисканням кнопки та завершити виконання роботи.

1.13 Оформити протокол комп'ютерного практикуму відповідно до вимог, наведених у вступі.

2 ПИТАННЯ ДЛЯ ЗАХИСТУ КОМП'ЮТЕРНОГО ПРАКТИКУМУ № 7

1. Етапи створення проектів систем автоматизації в SCADA.
2. Завдання властивостей ГЕ «Кнопка».
3. Завдання властивостей ГЕ «Текст».
4. Завдання властивостей ГЕ «Тренд».
5. Послідовність створення FBD-програми.
6. Послідовність створення ST-програми.
7. Призначення та синтаксис оператора *if* мови ST.
8. Призначення та синтаксис оператора *case* мови ST.
9. Призначення та синтаксис оператора *while* мови ST.
10. Призначення та синтаксис оператора *repeat* мови ST.
11. Призначення та синтаксис оператора *for* мови ST.

Комп’ютерний практикум № 8

ПРОГРАМУВАННЯ НА МОВАХ IL ТА SFC

Мета роботи – засвоєння послідовності програмування на мовах IL та SFC в SCADA-системі.

1 ПРОГРАМА РОБОТИ

Доповнити проект попередньої роботи № 7 вимикачем, за сигналами якого проводиться або зупиняється розрахунок сумарних витрат та сумарної вартості рідини.

1.1 Відкрити проект, створений у роботі № 7, за допомогою меню **File\Open...** (Файл\Відкрити...) або кнопки **Open existing project** (Відкрити існуючий проект) на інструментальній панелі ICP. Зберегти проект з новим ім’ям шляхом вибору в меню **File** (Файл) пункту **Save As...** (Зберегти як...), задавши з клавіатури бажане ім’я, наприклад **Robota_8**.

1.2 Вилучити з проекту програми **Синусоїда** та **Розрахунок**. У правій частині навігатора ПК на елементі **Синусоїда** викликати контекстне меню та вибрати рядок **Erase** (Вилучити) (рис. 1). Аналогічно вилучити програму **Розрахунок**.

1.3 Розкрити дерево вузла **RTM_1** у лівому вікні навігатора проекту. Викликати ПК на елементі **Channels** (Канали) контекстне меню та вибрати рядок **Create Component** (Створити компоненту), а потім **HEXT16_Channel**. У правій частині навігатора подвійним ЛК на створеному каналі відкрити вікно властивостей каналу та у рядку **Name** (Ім’я) задати з клавіатури нове ім’я **Вимикач**. Встановити тип каналу **Input**, а у полі **Recalc. Period** (Період перерахунку) значення **1**, а в полі **Mes. Unit** (Одиниця вимірювання) вибрати **s** (секунда).

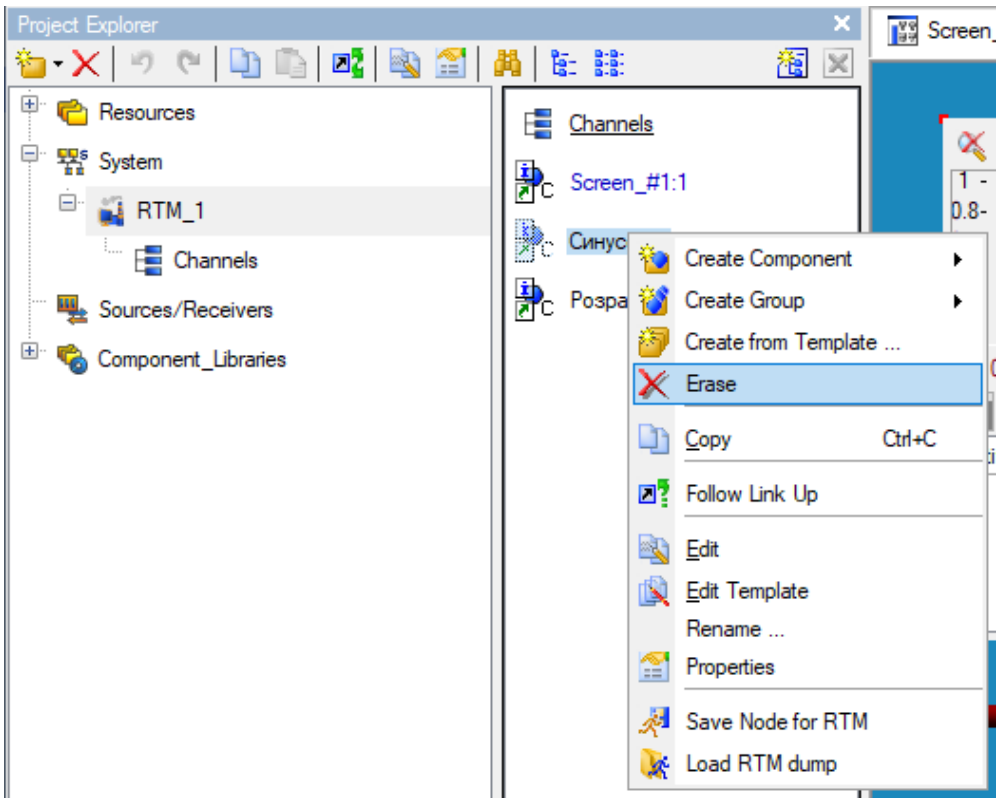


Рисунок 1 – Вилучення з проекту програми **Синусоїда**

1.4 Розмістити на екрані графічний елемент **Вимикач**. Вибрати на панелі інструментів до вподоби будь-який з 8 видів вимикачів групи **Switches** (Вимикачі) та розмістити його нижче кнопок, як показано на рис. 2.

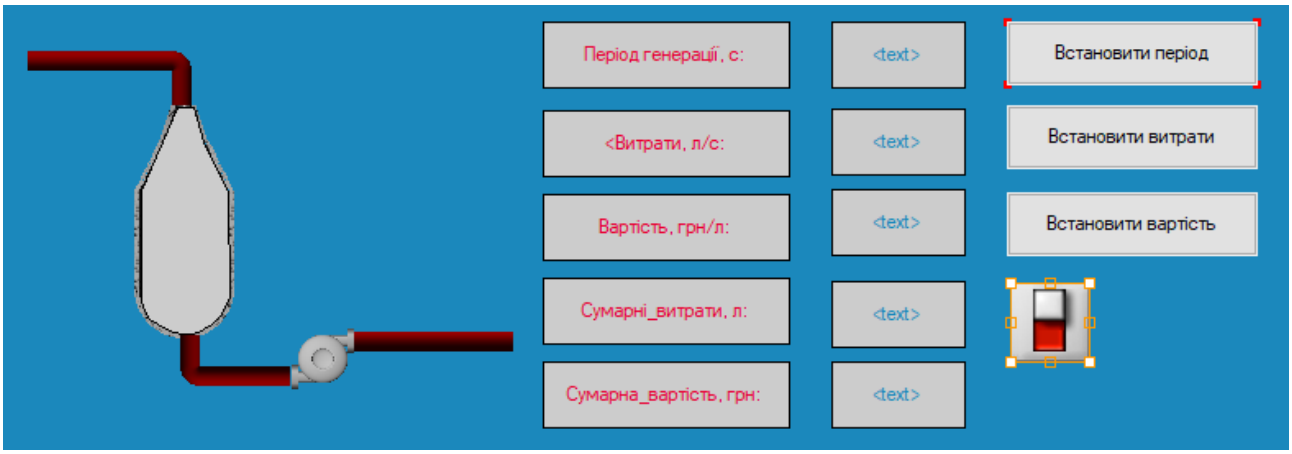


Рисунок 2 – Розміщення на екрані ГЕ **Вимикач**

Відкрити вікно властивостей вимикача та у полі **Constant** (Константа) і у полі **Value** (Значення) встановити значення **0x1** (рис. 3). Праворуч у полі

Binding (Привязка) ЛК відкрити вікно прив’язок та створити новий аргумент **Вимикач**, задавши тип **IN/OUT** та тип даних **BOOL** (рис. 4). У полі **Link** (Привязка) вікна подвійним ЛК відкрити вікно настроювань. Ліворуч у вікні вибрати канал **Вимикач**, а праворуч на вкладці **Attributes** (Атрибути) вибрати **Bit 1** (рис. 5). Послідовно кнопками **Link** та **Ok** завершити процес.

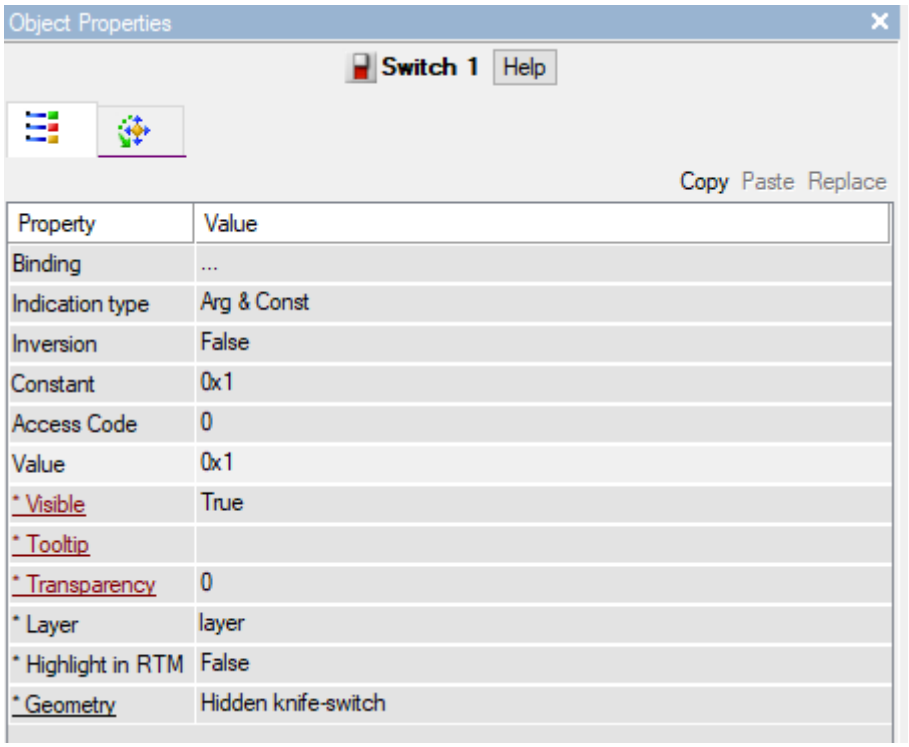


Рисунок 3 – Налаштування властивостей ГЕ **Вимикач**

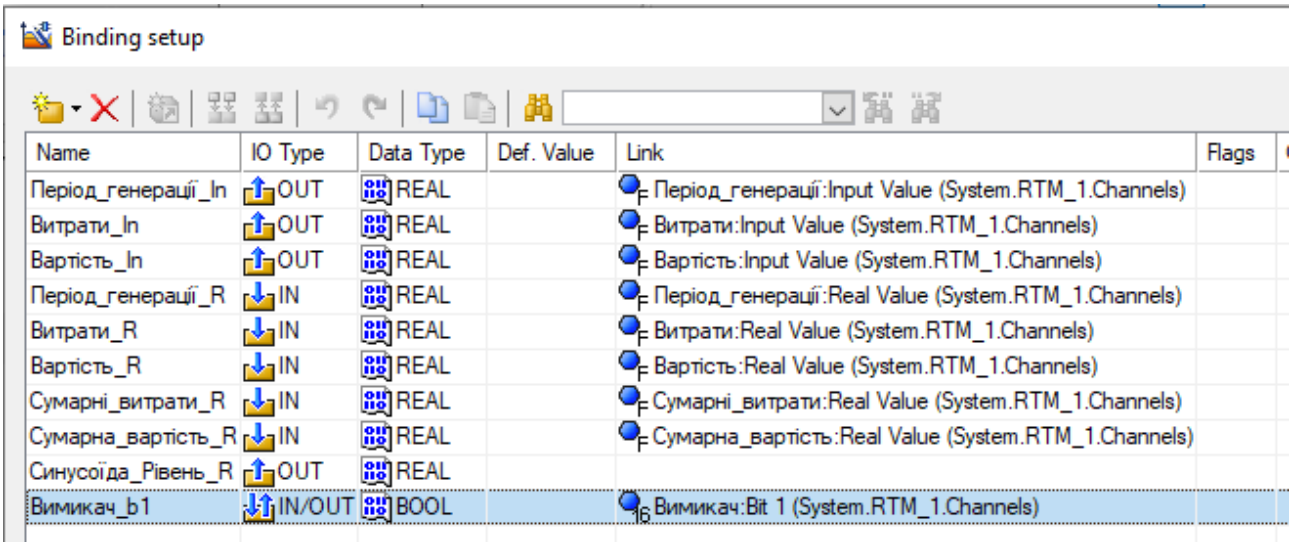


Рисунок 4 – Аргументи екрана та їх прив’язка

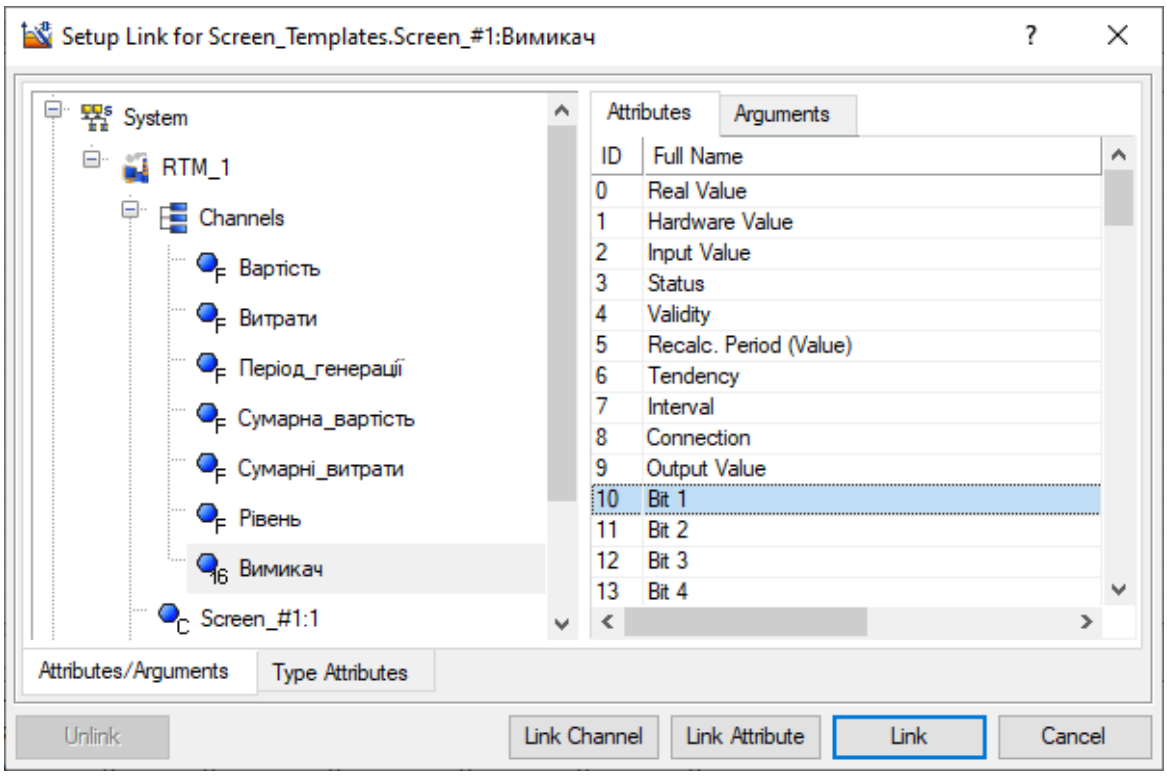


Рисунок 5 – Налаштування прив’язки вимикача

1.5 Створити SFC-діаграму. Виділити вузол **RTM_1** та ПК викликати контекстне меню, вибравши в ньому компоненту **Program** (Програма) (рис. 6).

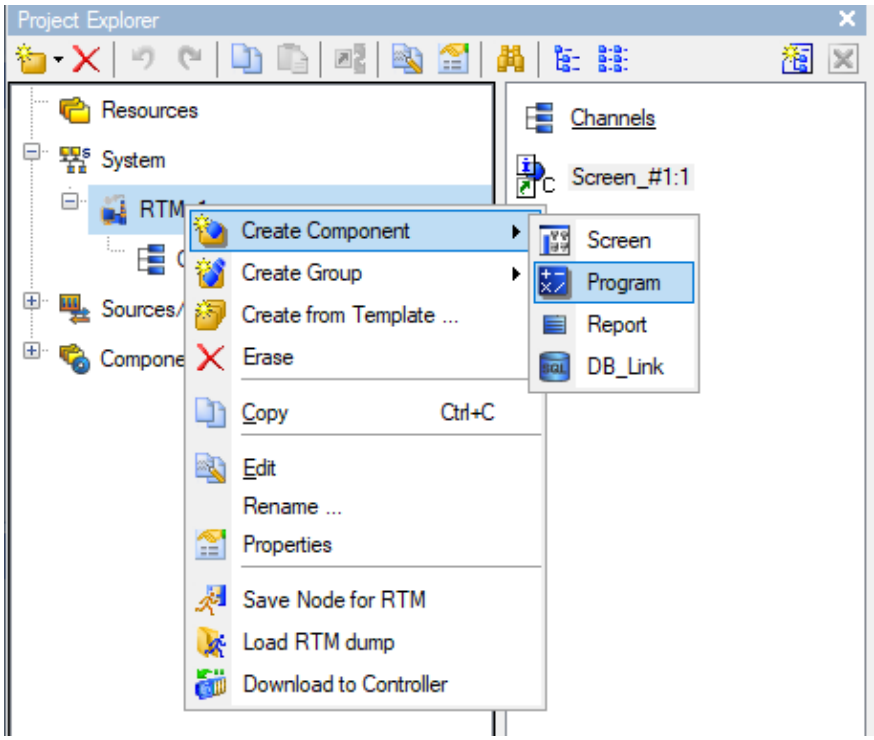


Рисунок 6 – Створення компоненти **Програма**

Виділити створену компоненту **Program_#1:9** та ПК викликати контекстне меню, вибравши в ньому ЛК пункт **Edit** (Редагувати). У вікні встановити період перерахування значення, рівний **1** секунді.

Виділити програму у вікні навігатора та ПК викликати контекстне меню, вибравши в ньому ЛК пункт **Edit Template** (Редагувати шаблон) (рис. 7).

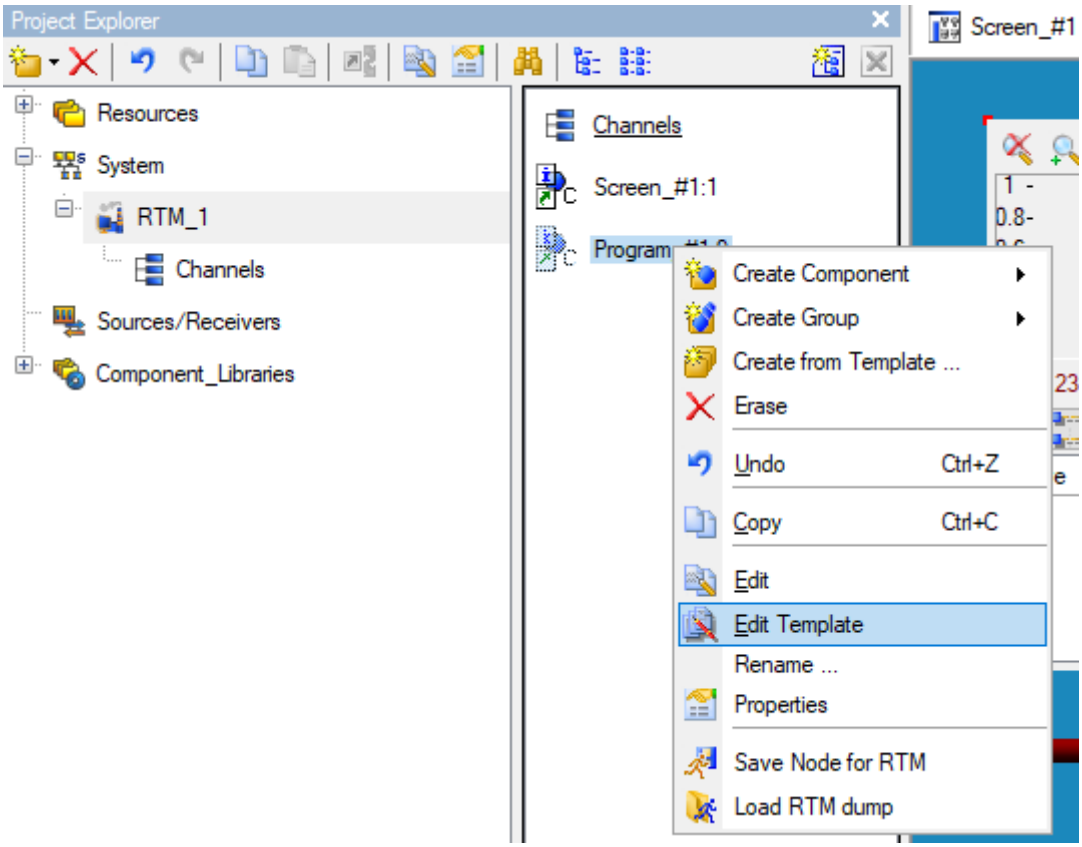
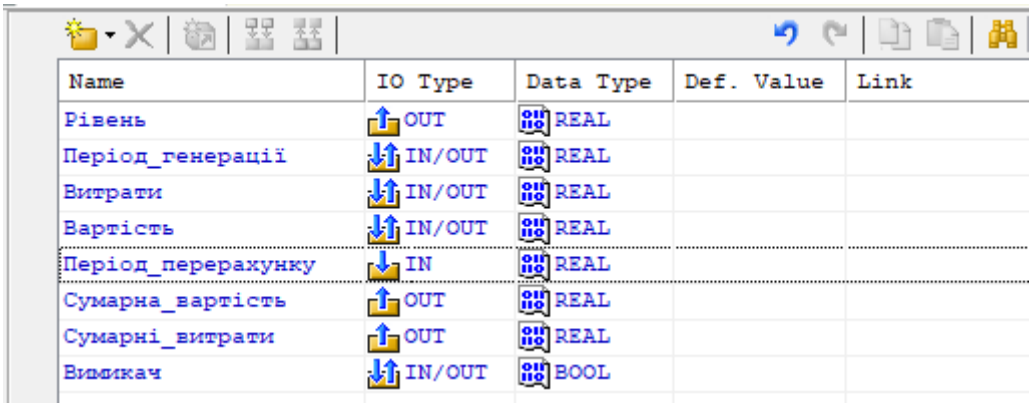


Рисунок 7 – Редагування аргументів програми

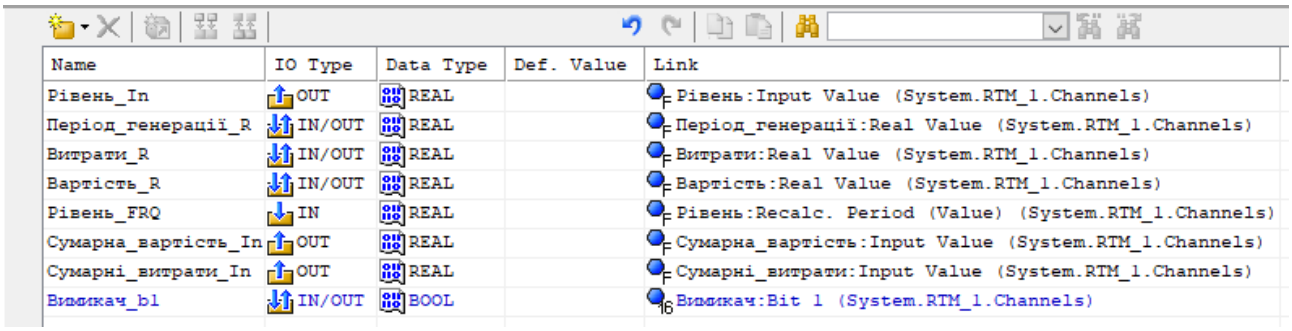
Виділити ЛК у дереві шаблону **Program_#1** рядок **Arguments** (Аргументи) та викликати табличний редактор аргументів. За допомогою кнопки створити послідовно 8 аргументів **Рівень, Період_генерації, Витрати, Вартість, Період_перерахунку, Сумарна_вартість, Сумарні_витрати, Вимикач** та задати їм тип, як показано на рис. 8. Подвійним ЛК у полі **Link** кожного аргументу відкрити вікно, розкрити ліворуч

дерево **Channels**, вибрати відповідний канал та здійснити прив'язку, задавши атрибути як на рис. 9.



Name	IO Type	Data Type	Def. Value	Link
Рівень	OUT	REAL		
Період_генерації	IN/OUT	REAL		
Витрати	IN/OUT	REAL		
Вартість	IN/OUT	REAL		
Період_перерахунку	IN	REAL		
Сумарна_вартість	OUT	REAL		
Сумарні_витрати	OUT	REAL		
Вимикач	IN/OUT	BOOL		

Рисунок 8 – Аргументи програми



Name	IO Type	Data Type	Def. Value	Link
Рівень_In	OUT	REAL		Рівень:Input Value (System.RTM_1.Channels)
Період_генерації_R	IN/OUT	REAL		Період_генерації:Real Value (System.RTM_1.Channels)
Витрати_R	IN/OUT	REAL		Витрати:Real Value (System.RTM_1.Channels)
Вартість_R	IN/OUT	REAL		Вартість:Real Value (System.RTM_1.Channels)
Рівень_FRQ	IN	REAL		Рівень:Recalc. Period (Value) (System.RTM_1.Channels)
Сумарна_вартість_In	OUT	REAL		Сумарна_вартість:Input Value (System.RTM_1.Channels)
Сумарні_витрати_In	OUT	REAL		Сумарні_витрати:Input Value (System.RTM_1.Channels)
Вимикач_bl	IN/OUT	BOOL		Вимикач:Bit 1 (System.RTM_1.Channels)

Рисунок 9 – Прив'язка аргументів програми

Виділити ЛК у дереві шаблону **Program_#1** рядок **Global Variables** (Глобальні змінні) та викликати табличний редактор аргументів. За допомогою кнопки створити глобальну змінну **Попередні_сумарні_витрати_R**, яка буде зберігати значення сумарних витрат, обчислених при попередньому виклику програми, задавши у полі **Initial Value** (Початкове значення) нульове значення (рис. 10).

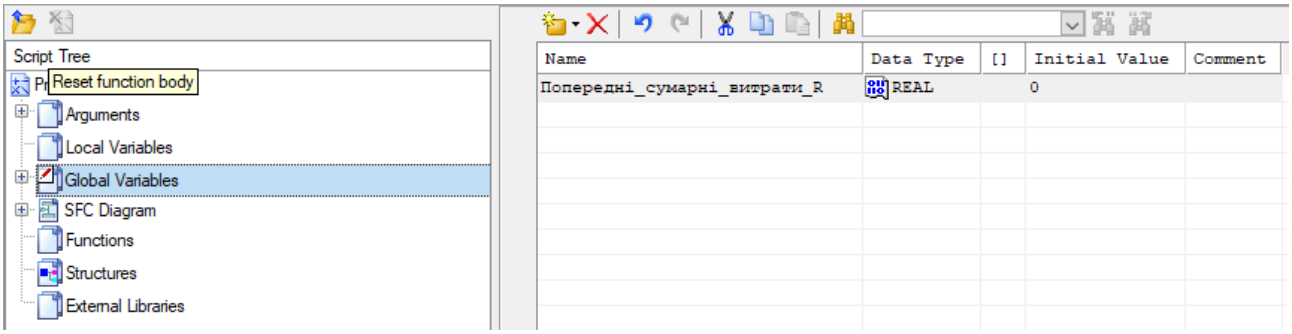


Рисунок 10 – Створення глобальної змінної

Виділити ЛК в дереві шаблону рядок **Program_#1** та у вікні, що відкрилося, вибрати мову **Sequential Function Chart (SFC)** (рис. 11). Після натискання кнопки **Ok** створюється програма у вікні редактора програм. На робочому полі відображається тільки один початковий крок **Initial Step** (рис. 12). Після пдвійного ЛК по даному кроці увести нове ім'я **Перевірка вихідних даних**. Для створення нового кроку виділити єдиний існуючий крок, а потім ЛК на іконці **Create Step/Transition** (Створити крок/Перехід) створити новий перехід **TRANSITION 0** та крок **STEP 1**.

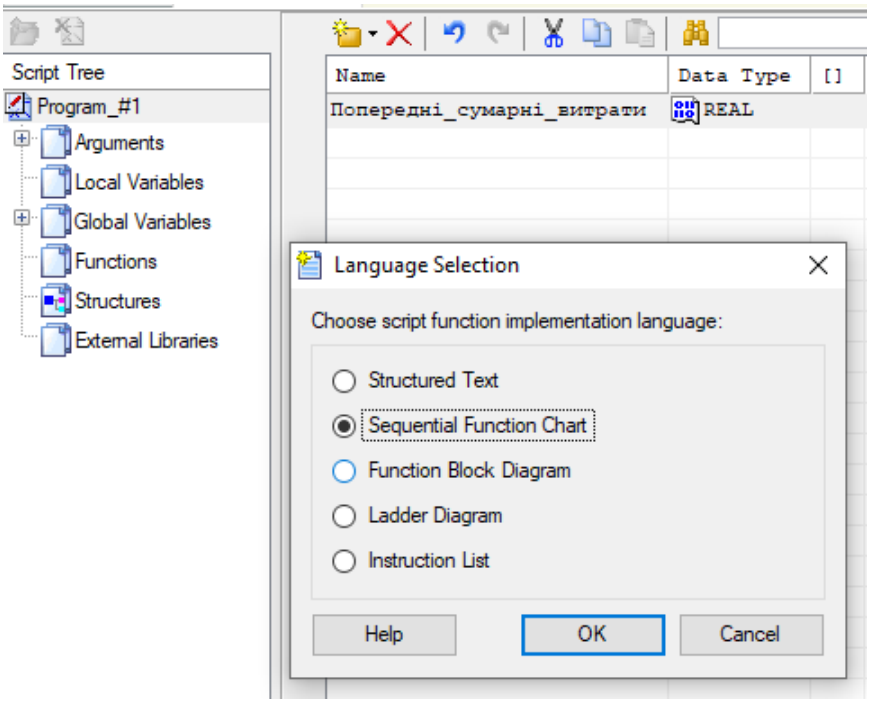


Рисунок 11 – Вибір мови програмування **SFC**

Рисунок 12 – Вікно нової програми **SFC**

Подвійним ЛК на створеному кроці змінити ім’я на **Генерування сигналу**, а переходу на **true**. Виділити крок **Генерування сигналу** та за допомогою іконки створити новий крок і перехід, назвавши їх **Повний розрахунок** та **Реальне значення вимикача** відповідно. Ще раз виділити крок **Генерування сигналу** та створити новий крок і перехід. Новий крок назвати **Розрахунок тільки попереднього значення сумарних витрат**, а перехід **Реальне значення вимикача із запереченням**. Вигляд створеної SFC-діаграми наведено на рис. 13.

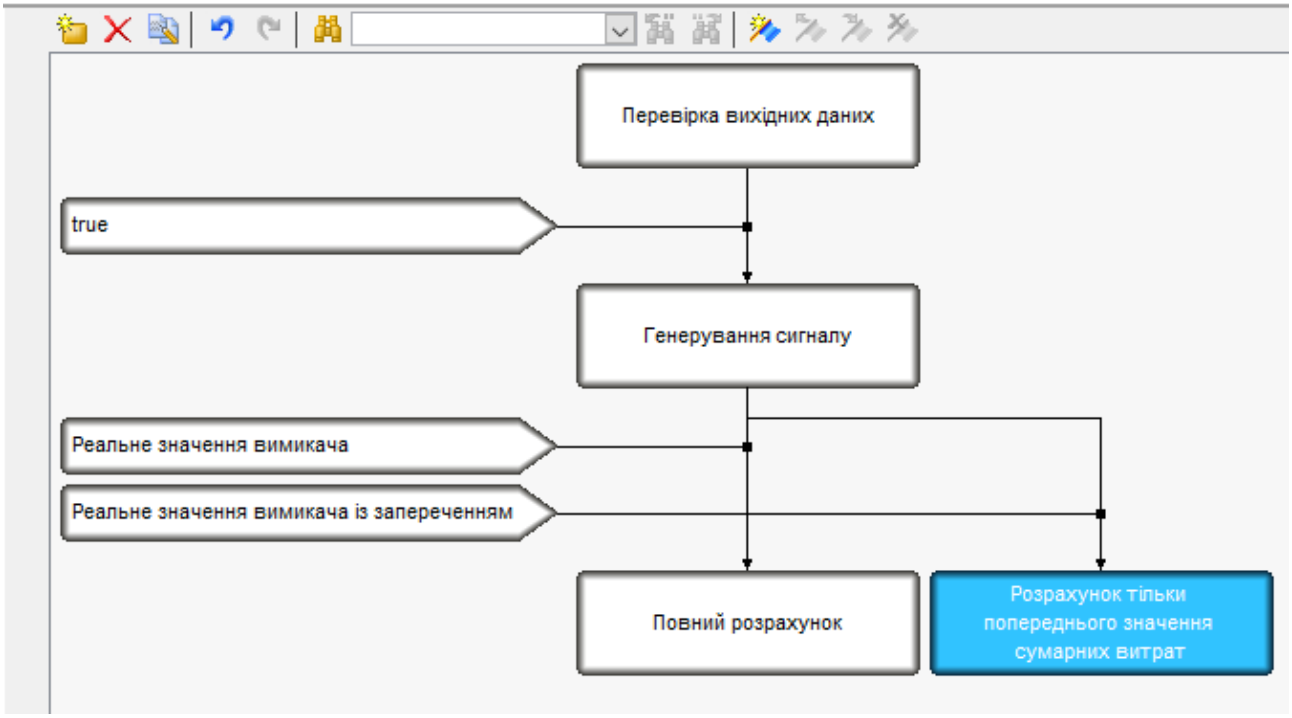


Рисунок 13 – Вікно SFC-діаграми

1.6 Реалізувати кроки програми SFC:

а) Реалізація кроку **Перевірка вихідних даних**. У структурі програми спочатку розкрити розділ **SFC Diagram**, а потім розділ **Steps** та виділити рядок **Перевірка вихідних даних.** Серед запропонованих мов вибрати мову програмування **Instruction List (IL)** (рис. 14)**.**

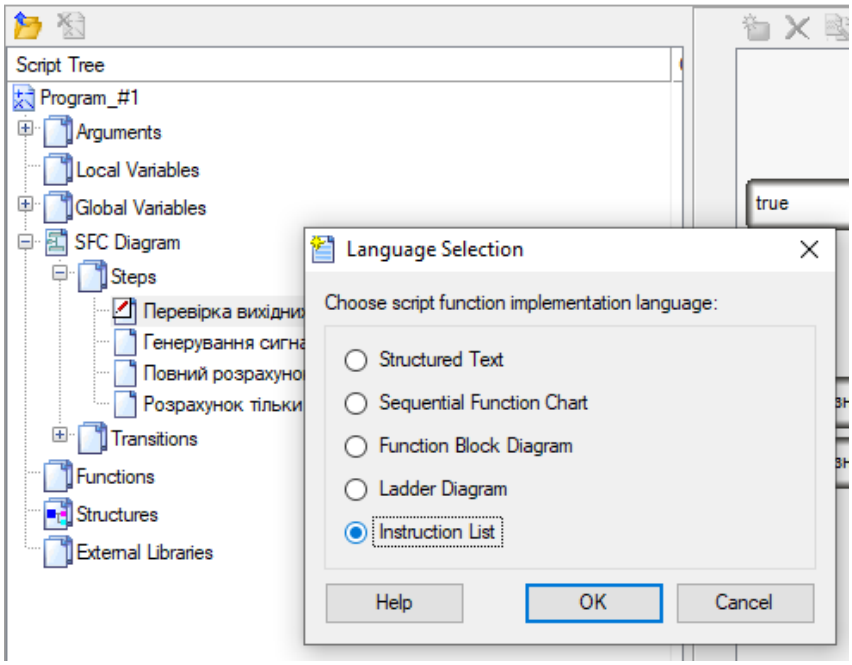


Рисунок 14 – Вибір мови програмування **IL**

При описаних вище іменах атрибутів і глобальної змінної текст програми має вид як на рис. 15.

```
SFC_STEP "Перевірка вихідних даних"
    VAR_OUTPUT Рівень_In : REAL; END_VAR
    VAR_INOUT Період_генерації_R : REAL; END_VAR
    VAR_INOUT Витрати_R : REAL; END_VAR
    VAR_INOUT Вартість_R : REAL; END_VAR
    VAR_INPUT Рівень_FRQ : REAL; END_VAR
    VAR_OUTPUT Сумарна_вартість_In : REAL; END_VAR
    VAR_OUTPUT Сумарні_витрати_In : REAL; END_VAR
    VAR_INOUT Вимикач_bl : BOOL; END_VAR


END_SFC_STEP
```

Рисунок 15 – Вікно вихідного поля програми **IL**

У виконавчому блоці програми написати програму з виконання наступних умов:

1) якщо значення вартості або витрат буде менше одиниці, то вони приймаються рівними одиниці;
2) якщо період генерації буде менше **10** секунд, то програма повинна встановити період рівний **10** секунд.

Результуюча програма має вигляд, як показано на рис. 16. За допомогою іконки **Build (F7)** (Компіляція) на інструментальній панелі редактора або гарячої клавіші F7 скомпілювати програму та переконатися в успішній компіляції у вікні **Messages**, яке при необхідності викликається з інструментальної панелі за допомогою іконки **Messages** (Повідомлення).

```
SFC_STEP "Перевірка вихідних даних"
    VAR_OUTPUT Рівень_In : REAL; END_VAR
    VAR_INOUT Період_генерації_R : REAL; END_VAR
    VAR_INOUT Витрати_R : REAL; END_VAR
    VAR_INOUT Вартість_R : REAL; END_VAR
    VAR_INPUT Рівень_FRQ : REAL; END_VAR
    VAR_OUTPUT Сумарна_вартість_In : REAL; END_VAR
    VAR_OUTPUT Сумарні_витрати_In : REAL; END_VAR
    VAR_INPUT Вимикач_bl : BOOL; END_VAR

    GE Витрати_R 1
    JMPC Перевірка_вартості
    LD 1
    ST Витрати_R
    Перевірка_вартості: GE Вартість_R 1
    JMPC Перевірка_періода
    LD 1
    ST Вартість_R
    Перевірка_періода: GE Період_генерації_R 10
    JMPC Кінець
    LD 10
    ST Період_генерації_R
    Кінець:

END_SFC_STEP
```

Рисунок 16 – Програма на мові **IL** кроку **Перевірка вихідних даних**

b) Реалізація кроку **Генерування сигналу**. Аналогічно до пункту a) виділити у розділі **Steps** рядок **Генерування сигналу**. Серед запропонованих

мов вибрати **FBD** та створити FBD-програму (рис. 17), як спрощений варіант аналогічної програми у роботі № 7. За допомогою іконки **Build (F7)** (Компіляція) на інструментальній панелі редактора або гарячої клавіші F7 скомпілювати програму та переконатися в успішній компіляції.

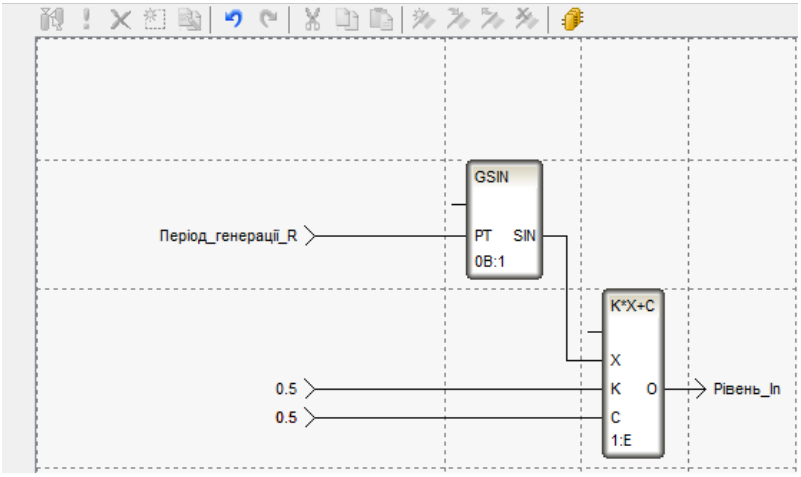


Рисунок 17 – Програма мовою **FBD**

Для реалізації переходу до створеного кроку у структурі програми (рис. 14) розкрити розділ **Transitions** та виділити рядок **true**. Серед запропонованих мов вибрати мову програмування **ST.** У виконавчому блоці програми ввести оператор **true**. Результуюча програма має вигляд, як показано на рис. 18.

```
SFC_TRANSITION "true" FROM( INITIAL_STEP ) TO( STEP_1 )
    VAR_OUTPUT Рівень_In : REAL; END_VAR
    VAR_INOUT Період_генерації_R : REAL; END_VAR
    VAR_INOUT Витрати_R : REAL; END_VAR
    VAR_INOUT Вартість_R : REAL; END_VAR
    VAR_INPUT Рівень_FRQ : REAL; END_VAR
    VAR_OUTPUT Сумарна_вартість_In : REAL; END_VAR
    VAR_OUTPUT Сумарні_витрати_In : REAL; END_VAR
    VAR_INPUT Вимикач_bl : BOOL; END_VAR

    true

END_SFC_TRANSITION
```

Рисунок 18 – Програма мовою **ST** для переходу до кроку 1

За допомогою іконки **Build (F7)** (Компіляція) на інструментальній панелі редактора або гарячої клавіші F7 скомпілювати програму та переконатися в успішній компіляції.

c) Реалізація кроку **Повний розрахунок**. Аналогічно до пункту а) виділити у розділі **Steps** рядок **Повний розрахунок.** Серед запропонованих мов вибрати **IL**. У виконавчому блоці програми ввести оператори для обчислення сумарних витрат та сумарної вартості продукту з використанням глобальної змінної аналогічно до роботи № 7 при поверненні перемикачем значення **1**, як показано на рис. 19. За допомогою іконки **Build (F7)** (Компіляція) на інструментальній панелі редактора або гарячої клавіші F7 скомпілювати програму та переконатися в успішній компіляції.

```
SFC_STEP "Повний розрахунок"
    VAR_OUTPUT Рівень_In : REAL; END_VAR
    VAR_INOUT Період_генерації_R : REAL; END_VAR
    VAR_INOUT Витрати_R : REAL; END_VAR
    VAR_INOUT Вартість_R : REAL; END_VAR
    VAR_INPUT Рівень_FRQ : REAL; END_VAR
    VAR_OUTPUT Сумарна_вартість_In : REAL; END_VAR
    VAR_OUTPUT Сумарні_витрати_In : REAL; END_VAR
    VAR_INPUT Вимикач_b1 : BOOL; END_VAR

    MUL Рівень_FRQ Витрати_R
    ADD Попередні_сумарні_витрати_R
    ST Попередні_сумарні_витрати_R
    ST Сумарні_витрати_In
    MUL Вартість_R
    ST Сумарна_вартість_In

END_SFC_STEP
```

Рисунок 19 – Програма мовою **IL** кроку **Повний розрахунок**

Для реалізації переходу до створеного кроку виділити у розділі **Transitions** рядок **Реальне значення вимикача**. Серед запропонованих мов вибрати **ST**. У виконавчому блоці програми ввести оператор **Вимикач_b1**. Результуюча програма має вигляд, як показано на рис. 20. За допомогою іконки

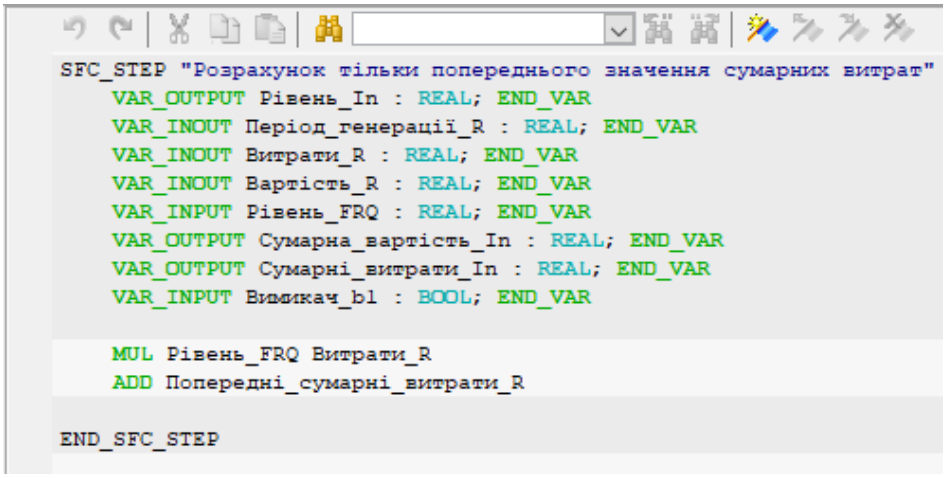
Build (F7) (Компіляція) на інструментальній панелі редактора або гарячої клавіші F7 скомпілювати програму та переконатися в успішній компіляції.

```
SFC_TRANSITION "Реальне значення вимикача" FROM( STEP_1 ) TO( STEP_2 )
    VAR_OUTPUT Рівень_In : REAL; END_VAR
    VAR_INOUT Період_генерації_R : REAL; END_VAR
    VAR_INOUT Витрати_R : REAL; END_VAR
    VAR_INOUT Вартість_R : REAL; END_VAR
    VAR_INPUT Рівень_FRQ : REAL; END_VAR
    VAR_OUTPUT Сумарна_вартість_In : REAL; END_VAR
    VAR_OUTPUT Сумарні_витрати_In : REAL; END_VAR
    VAR_INPUT Вимикач_bl : BOOL; END_VAR

    Вимикач_bl

END_SFC_TRANSITION
```

Рисунок 20 – Програма мовою **ST** для переходу до кроку 2

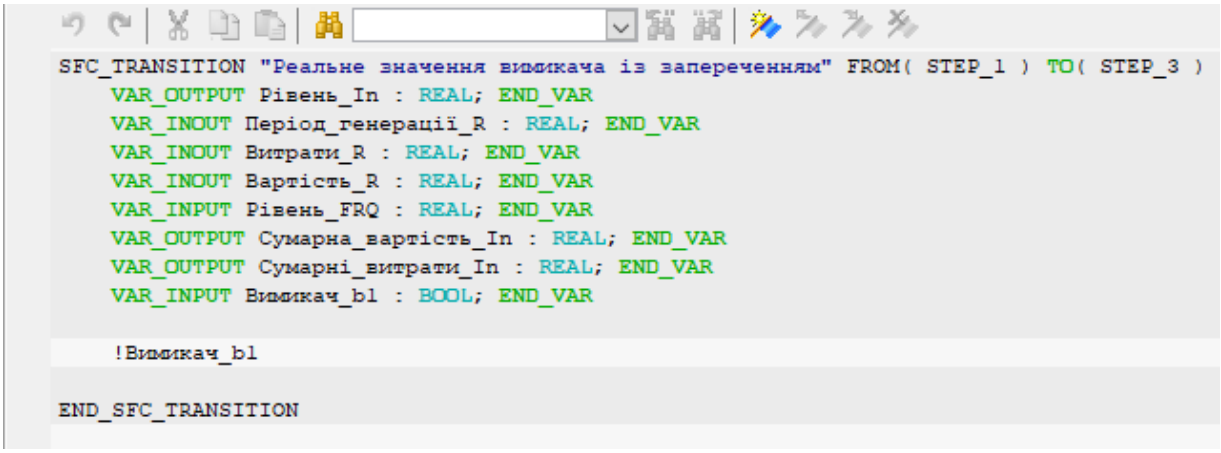
d) Реалізація кроку **Розрахунок тільки попереднього значення сумарних витрат**. Аналогічно до пункту а) виділити у розділі **Steps** рядок **Розрахунок тільки попереднього значення сумарних витрат.** Серед запропонованих мов вибрати **IL**. У виконавчому блоці програми ввести оператори для обчислення сумарних витрат при поверненні перемикачем значення **0**. При цьому сумарні витрати не повертаються до каналу для передачі інформації, а тільки присвоюються глобальній змінній (рис. 21). За допомогою іконки **Build (F7)** (Компіляція) на інструментальній панелі редактора або гарячої клавіші F7 скомпілювати програму та переконатися в успішній компіляції.

Для створення переходу до створеного кроку виділити у розділі **Transitions** рядок **Реальне значення вимикача із запереченням.** Серед запропонованих мов вибрати **ST.** У виконавчому блоці програми ввести оператор **!Вимикач_b1**. Результуюча програма має вигляд, як показано на рис. 22. За допомогою іконки **Build (F7)** (Компіляція) на інструментальній панелі редактора або гарячої клавіші F7 скомпілювати програму та переконатися в успішній компіляції.



```
SFC_STEP "Розрахунок тільки попереднього значення сумарних витрат"
    VAR_OUTPUT Рівень_In : REAL; END_VAR
    VAR_INOUT Період_генерації_R : REAL; END_VAR
    VAR_INOUT Витрати_R : REAL; END_VAR
    VAR_INOUT Вартість_R : REAL; END_VAR
    VAR_INPUT Рівень_FRQ : REAL; END_VAR
    VAR_OUTPUT Сумарна_вартість_In : REAL; END_VAR
    VAR_OUTPUT Сумарні_витрати_In : REAL; END_VAR
    VAR_INPUT Вимикач_bl : BOOL; END_VAR

    MUL Рівень_FRQ Витрати_R
    ADD Попередні_сумарні_витрати_R

END_SFC_STEP
```

Рисунок 21 – Програма на мові **IL** кроку **Розрахунок тільки попереднього значення сумарних витрат**



```
SFC_TRANSITION "Реальне значення вимикача із запереченням" FROM( STEP_1 ) TO( STEP_3 )
    VAR_OUTPUT Рівень_In : REAL; END_VAR
    VAR_INOUT Період_генерації_R : REAL; END_VAR
    VAR_INOUT Витрати_R : REAL; END_VAR
    VAR_INOUT Вартість_R : REAL; END_VAR
    VAR_INPUT Рівень_FRQ : REAL; END_VAR
    VAR_OUTPUT Сумарна_вартість_In : REAL; END_VAR
    VAR_OUTPUT Сумарні_витрати_In : REAL; END_VAR
    VAR_INPUT Вимикач_bl : BOOL; END_VAR

    !Вимикач_bl

END_SFC_TRANSITION
```

Рисунок 22 – Програма мовою **ST** для переходу до кроку 3

1.7 Прив’язати криву ГЕ **Trend** (Тренд) та процес наповнення ємності рідиною до сигналу **Рівень** програми. На екрані подвійним ЛК на ГЕ **Trend** (Тренд) відкрити вікно його властивостей. Перейти на вкладку **Curves** та подвійним ЛК розкрити розділ **Curve1**. У правому полі рядка **Binding** (Прив’язка) ЛК відкрити вікно **Binding setup** та вибрати рядок **Синусоїда_Рівень_R** (див. рис. 4). Подвійним ЛК у полі **Link** відкрити вікно, розкрити ліворуч дерево **Channels** та вибрати канал **Рівень** (рис. 23). Праворуч у закладці **Attributes** вибрати ЛК **Real Value** (Реальне значення) та послідовно кнопками **Link** та **Ok** підтвердити прив'язку атрибута. В результаті вікно прив’язок буде мати вигляд, показаний на рис. 24.

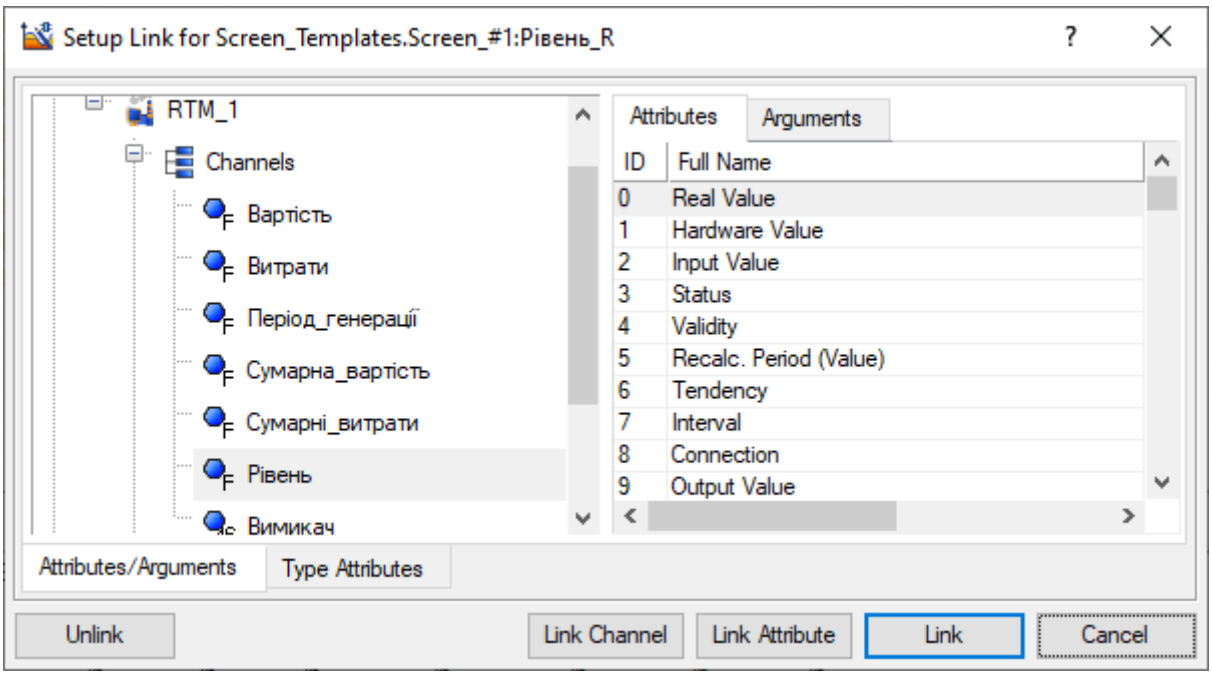


Рисунок 23 – Настроювання прив’язки до каналу **Рівень**

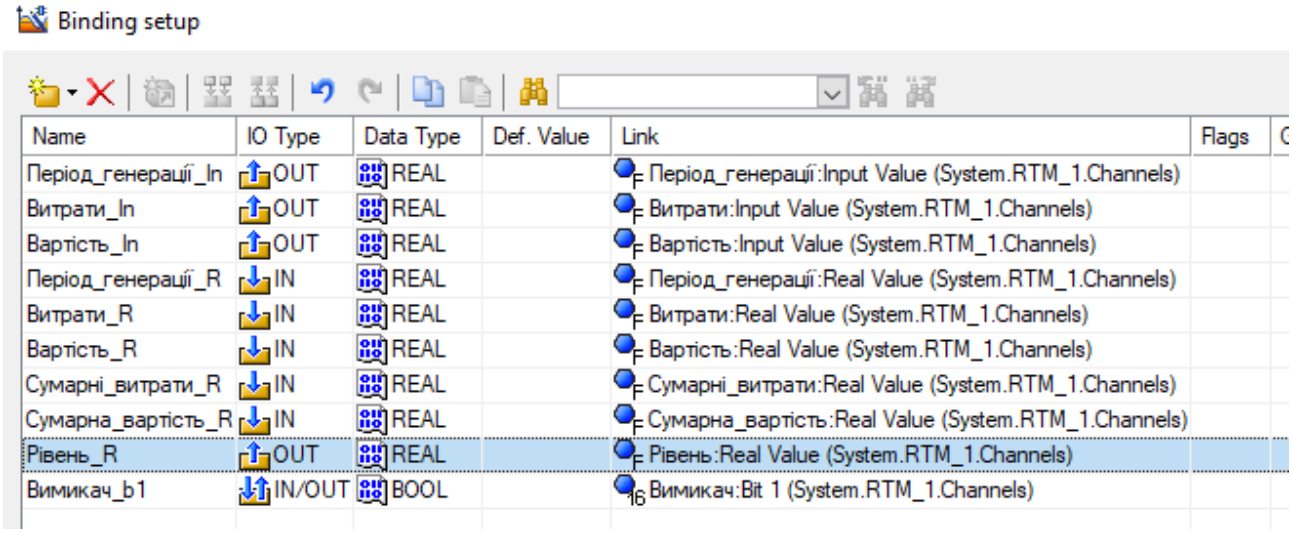


Рисунок 24 – Остаточний вигляд вікна прив’язок

1.8 Вказати (перевірити) у проекті свої дані.

На графічному екрані розмістити у зручному місці шифр групи та своє ПІБ аналогічно до роботи № 1.

На графічному екрані розмістити у зручному місці своє фото.

В меню **File\Information...**(Файл\Інформація...) в рядку **Owner** (Власник) написати за допомогою клавіатури своє ПІБ, а в рядку **Organization** (Організація) – шифр групи та факультету.

1.9 Запустити проект в роботу. Зберегти проект натисканням кнопки . На головній інструментальній панелі ICP натиснути ЛК кнопку **Save for RTM** (Зберегти для МРЧ), скомпілювавши проект для запуску в реальному часі. Виділити ЛК в навігаторі проекту вузол **RTM_1**, а потім натиснути на головній панелі кнопку **Start Profiler** (Запустить профайлер). У вікні, що відкрилося, запустити проект на виконання шляхом натискання ЛК кнопки **Run** (Пуск/Зупинка).

Спостерігати процес при різних значеннях вихідних даних та положеннях перемикача (рис. 25).

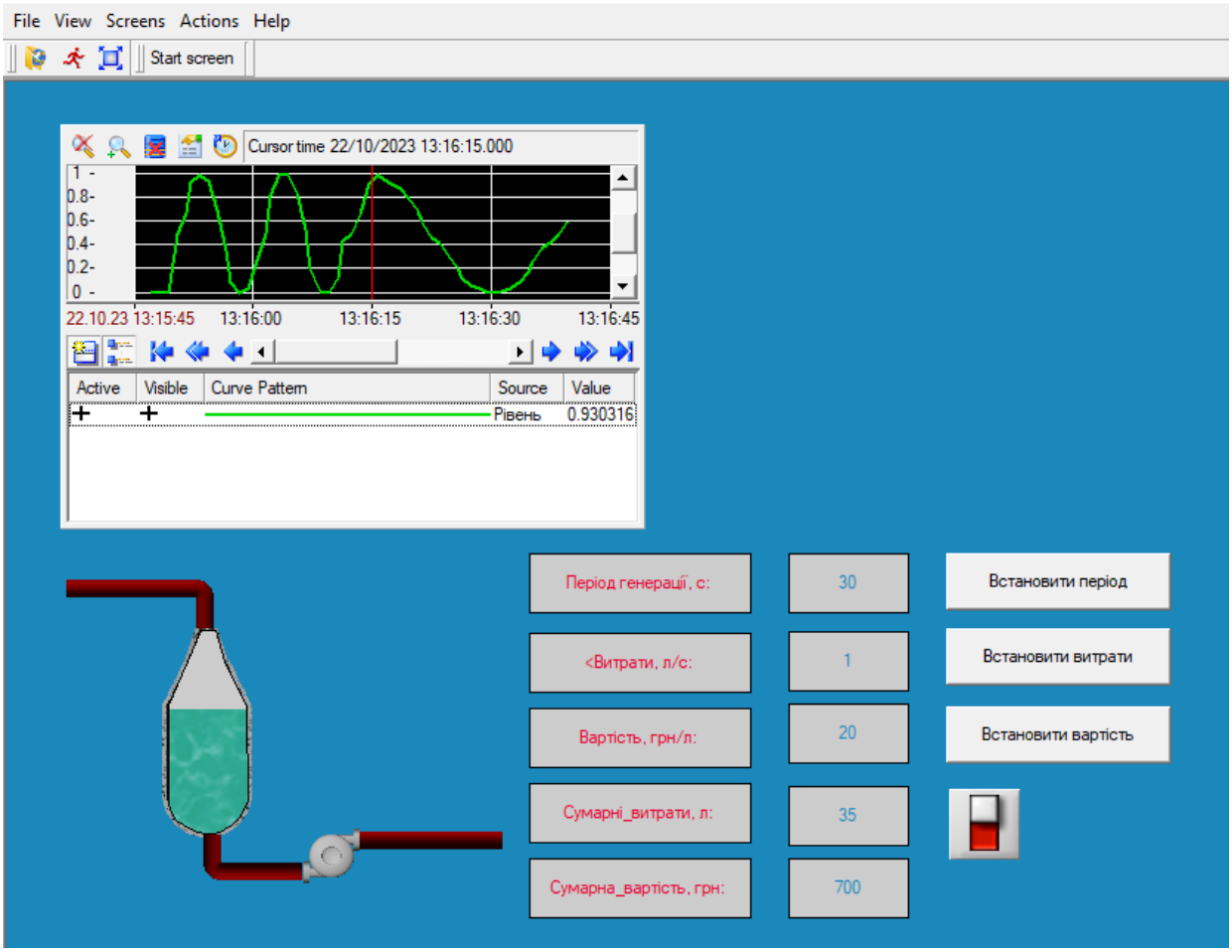


Рисунок 25 – Вигляд графічного екрану проекту

Зупинити виконання проекту шляхом натискання на панелі інструментів ЛК кнопки **Run** (Пуск/Зупинка). Закрити вікно графічного екрану через меню **File\Exit** (Файл\Вихід) або засобами Windows.

1.11 Оформити протокол комп’ютерного практикуму відповідно до вимог, наведених у вступі.

2 ПИТАННЯ ДЛЯ ЗАХИСТУ КОМП’ЮТЕРНОГО ПРАКТИКУМУ № 8

1. Етапи створення проектів систем автоматизації в SCADA.
2. Завдання властивостей ГЕ «Вимикачі».
3. Що представляє собою SFC-програма.
4. Послідовність створення нового кроку, циклу, паралельних кроків SFC-програми.
5. Операнди та модифікатори мови IL.
6. Акумулятор мови IL, основи роботи з ним.
7. Створення аргументів екрана та завдання їхніх параметрів.
8. Які мови програмування є в SCADA-системі.

Комп’ютерний практикум № 9

СТВОРЕННЯ ЗВІТУ ТРИВОГ ТА СПАД-АРХІВУ

Мета роботи – засвоєння послідовності створення звіту тривог та СПАД-архіву в SCADA-системі.

1 ПРОГРАМА РОБОТИ

1.1 Відкрити проект, створений у роботі № 8, за допомогою меню **File\Open...** (Файл\Відкрити...) або кнопки **Open existing project** (Відкрити існуючий проект) на інструментальній панелі ICP. Зберегти проект з новим ім’ям шляхом вибору в меню **File** (Файл) пункту **Save As...** (Зберегти як...), задавши з клавіатури бажане ім’я, наприклад **Robota_9**.

1.2 Вилучити з екрану існуючий тренд та розмістите на його місці новий ГЕ **Archive trend** (Архівний тренд) (рис. 1).

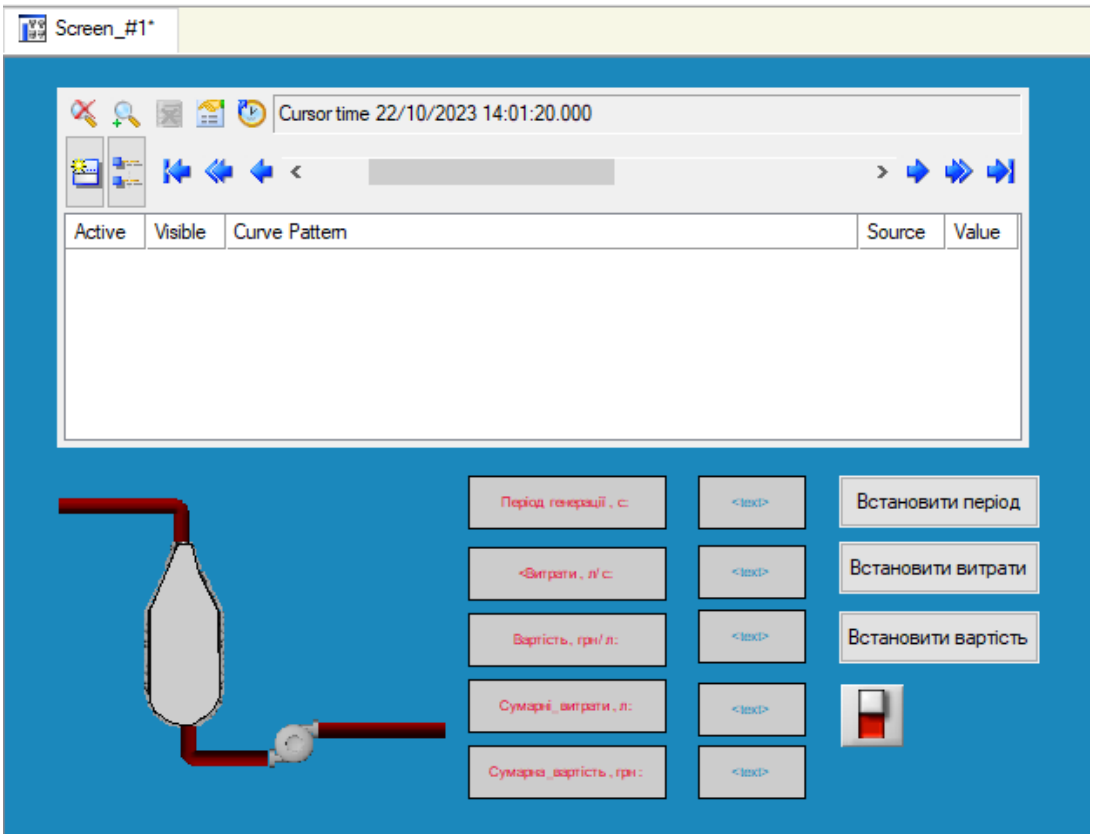


Рисунок 1 – Вікно проекту з ГЕ **Архівний тренд**

Відкрити вікно властивостей ГЕ **Архівний тренд,** створити криву та прив’язати її до аргументу **Рівень** (рис. 2). У рядку **Max value** вікна розділу **Curve1** встановити значення рівне **1**, а також задати бажані значення іншим параметрам тренду.

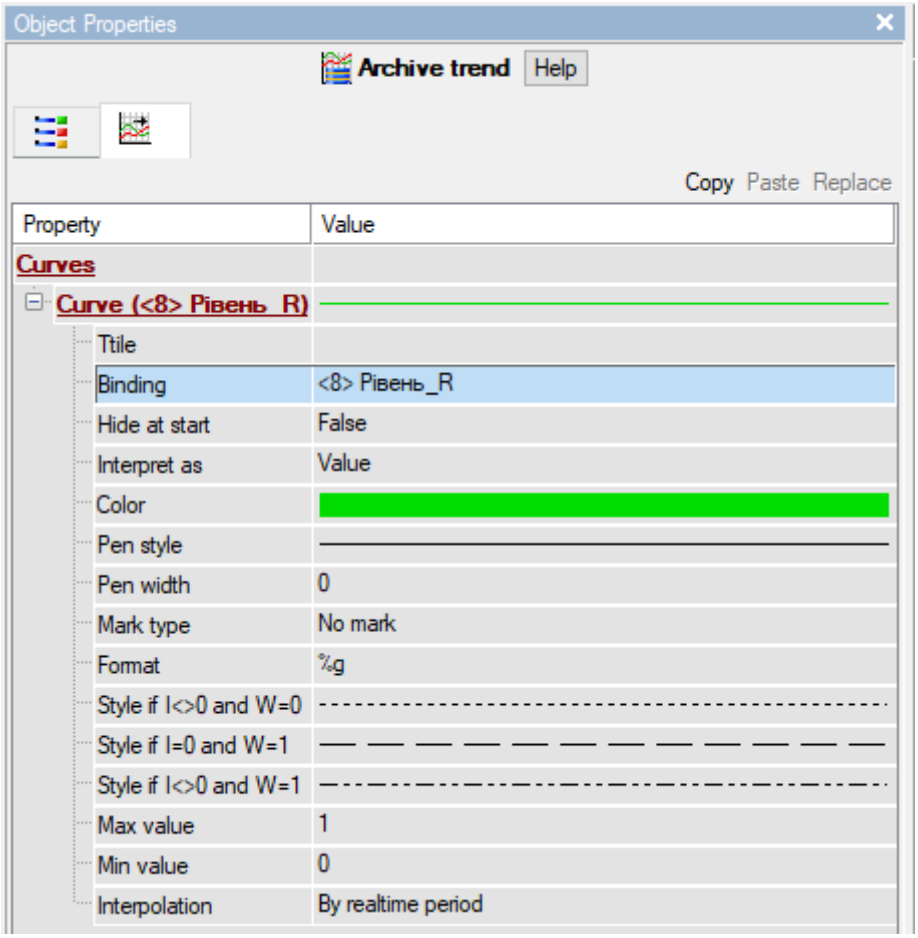
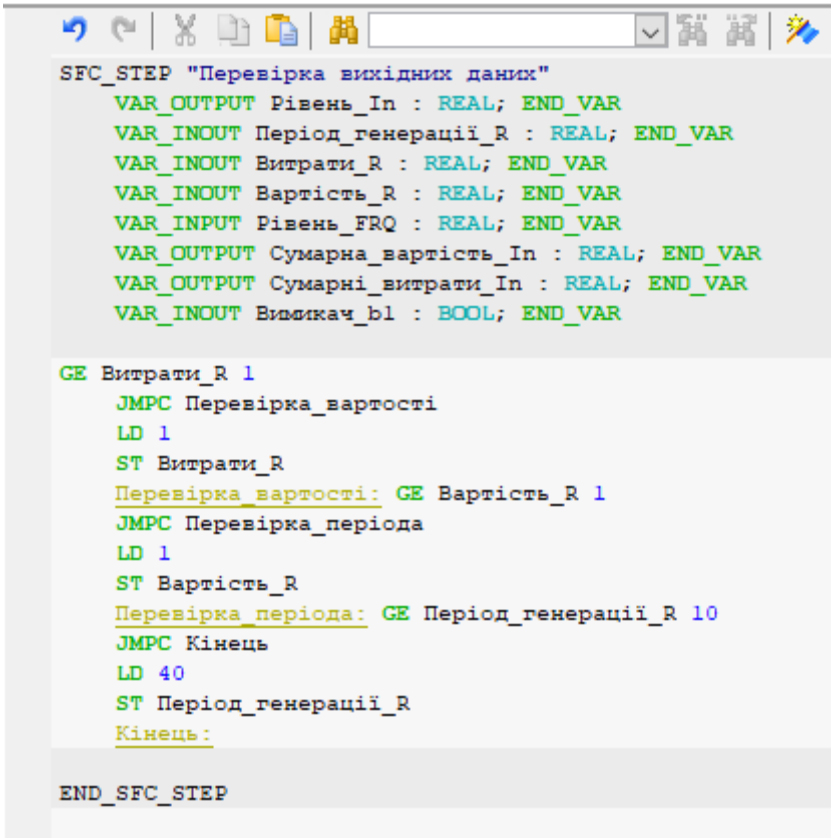


Рисунок 2 – Прив’язка та параметри кривої ГЕ **Архівний тренд**

1.3 Відредагувати крок програми **Перевірка вихідних даних**. Змінити значення **Період_генерації** з **10** на **40** (третій знизу оператор у виконавчому блоці програми, рис. 3). За допомогою іконки **Build (F7)** (Компіляція) на інструментальній панелі редактора або гарячої клавіші F7 скомпілювати програму та переконатися в успішній компіляції у вікні **Messages**, яке при необхідності викликається з інструментальної панелі за допомогою іконки **Messages** (Повідомлення).



```
SFC_STEP "Перевірка вихідних даних"
    VAR_OUTPUT Рівень_In : REAL; END_VAR
    VAR_INOUT Період_генерації_R : REAL; END_VAR
    VAR_INOUT Витрати_R : REAL; END_VAR
    VAR_INOUT Вартість_R : REAL; END_VAR
    VAR_INPUT Рівень_FRQ : REAL; END_VAR
    VAR_OUTPUT Сумарна_вартість_In : REAL; END_VAR
    VAR_OUTPUT Сумарні_витрати_In : REAL; END_VAR
    VAR_INOUT Вимикач_bl : BOOL; END_VAR

GE Витрати_R 1
    JMPC Перевірка_вартості
    LD 1
    ST Витрати_R
    Перевірка_вартості: GE Вартість_R 1
    JMPC Перевірка_періода
    LD 1
    ST Вартість_R
    Перевірка_періода: GE Період_генерації_R 10
    JMPC Кінець
    LD 40
    ST Період_генерації_R
    Кінець:

END_SFC_STEP
```

Рисунок 3 – Програма на мові **IL** кроку **Перевірка вихідних даних**

1.4 Налаштувати контроль меж каналу **Рівень.** Виділити канал **Рівень** у групі **Channels** вузла **RTM_1** та ПК викликати контекстне меню. Вибрати в меню рядок **Edit** (Редагувати). На панелі **Boundaries** (Межі) поставити прапорці **Enable** (Умівкнути) та **Boundary Trace** (Контроль меж). Встановити межі діапазонів, як показано на рис. 4.

1.5 Створити словник повідомлень. Виділити вузол **RTM_1** та ПК викликати контекстне меню. Вибрати рядок **Create Group** (Створити групу), а потім рядок **Message_Dictionaries** (Словники_повідомлень) (рис. 5). Виділити створену групу **Message_Dictionaries** та ПК викликати контекстне меню. Вибрати рядок **Create Component** (Створити компоненту), а потім серед запропонованих словників вибрати **Dictionary_for_FLOAT** (Словник_для_FLOAT) (рис. 6).

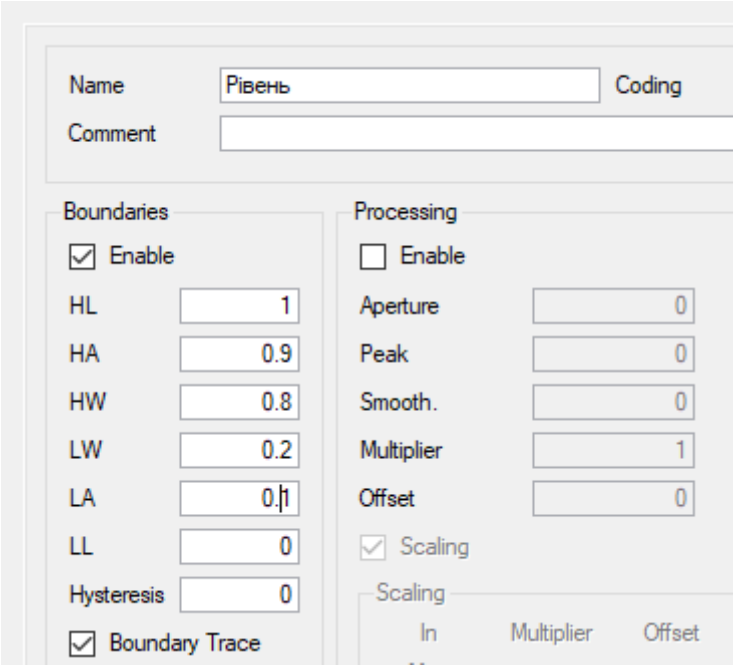


Рисунок 4 – Завдання меж каналу **Рівень**

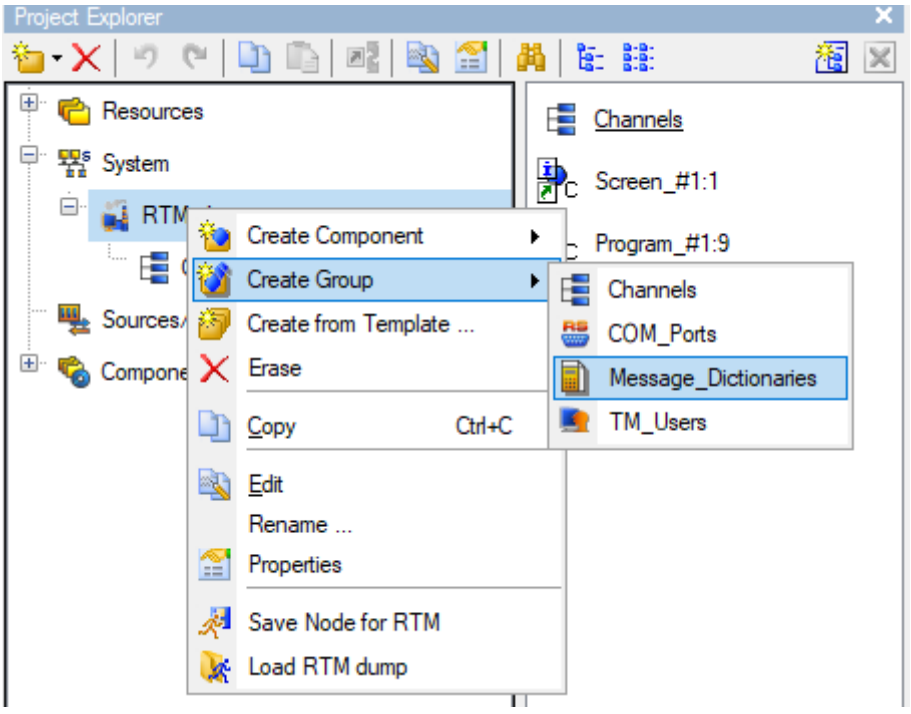


Рисунок 5 – Створення групи словників повідомлень

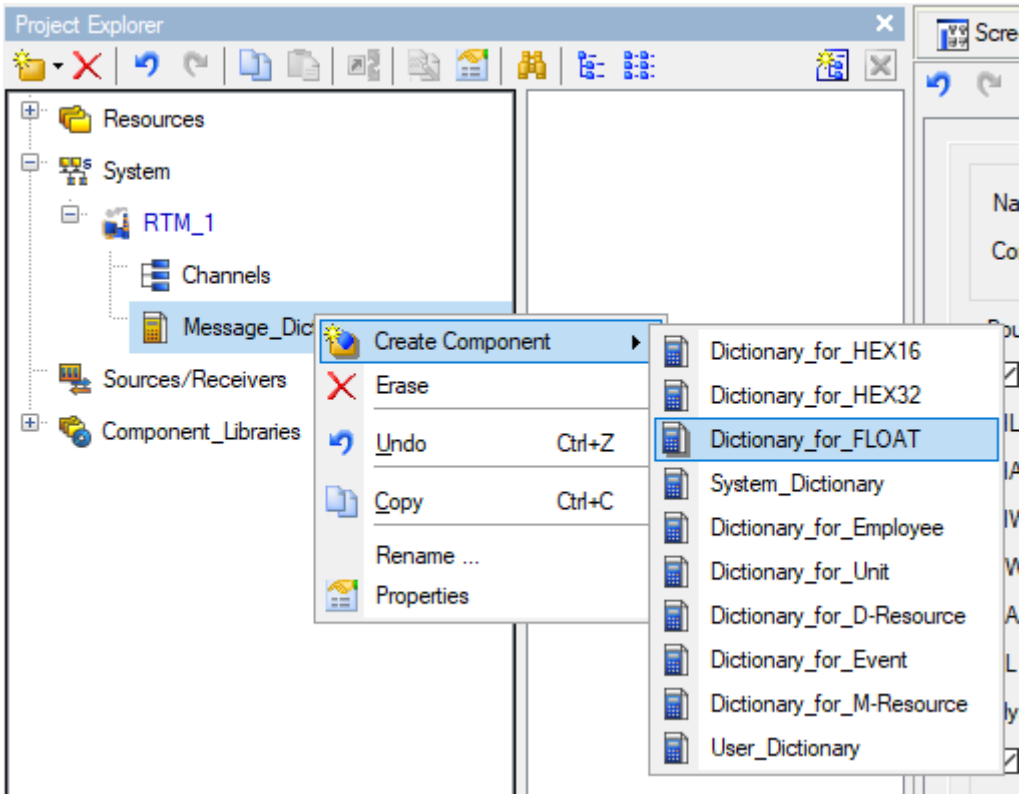


Рисунок 6 – Створення словника повідомлень

Подвійним ЛК на створеному словникові повідомлень вибрати **Edit** (Редагувати). У поле **Name** (Ім’я) ввести ім'я словника **Для_рівня**. Подвійним ЛК на кожному рядку відкрийте вікно для редагування повідомлення (рис. 7). Редагуючи послідовно всі повідомлення, налаштувати словник повідомлень, як показано на рис. 8.

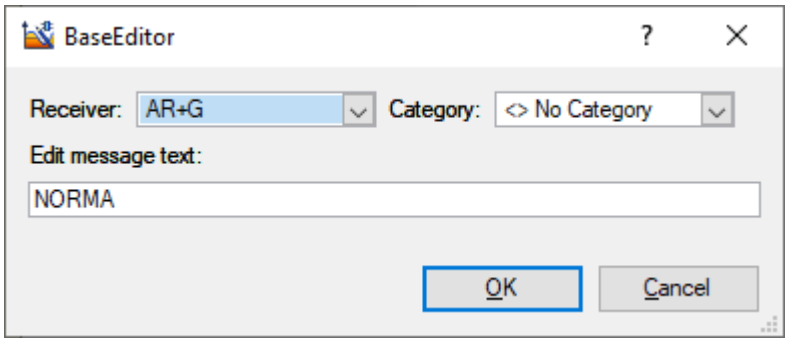


Рисунок 7 – Вікно завдання повідомлень

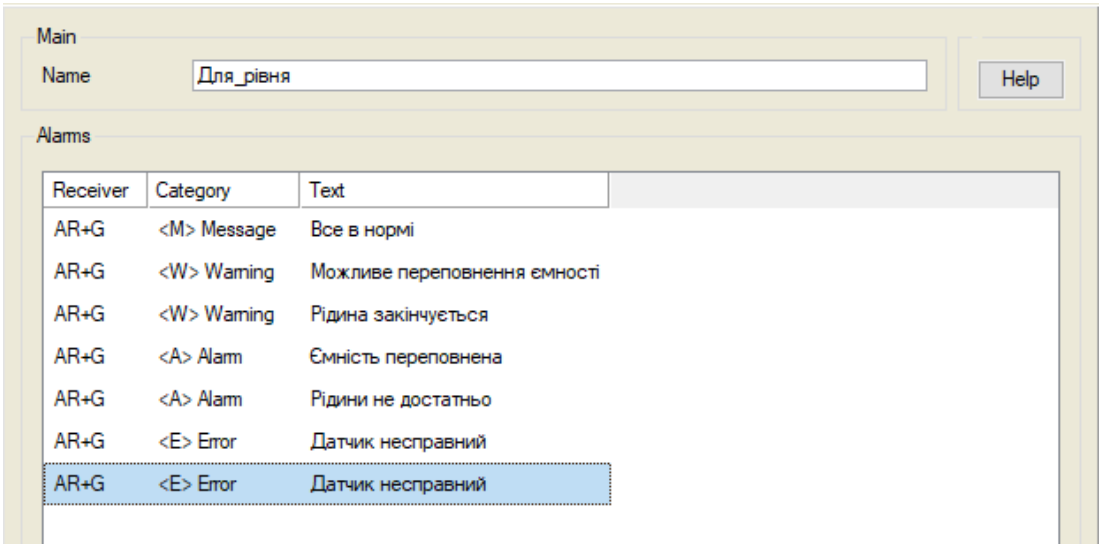


Рисунок 8 – Редактор словника повідомлень

1.6 Відредагувати вузол. Виділити вузол **RTM_1,** викликати ПК контекстне меню та вибрати **Edit** (Редагувати). У вікні перейти на вкладку **Alarm/Dump/Parameters** (Тривога/Дамп/Параметри). Заповнити поля, як показано на рис. 9. Перейти на вкладку **Archives** (Архіви) та заповнити поля, як показано на рис. 10.

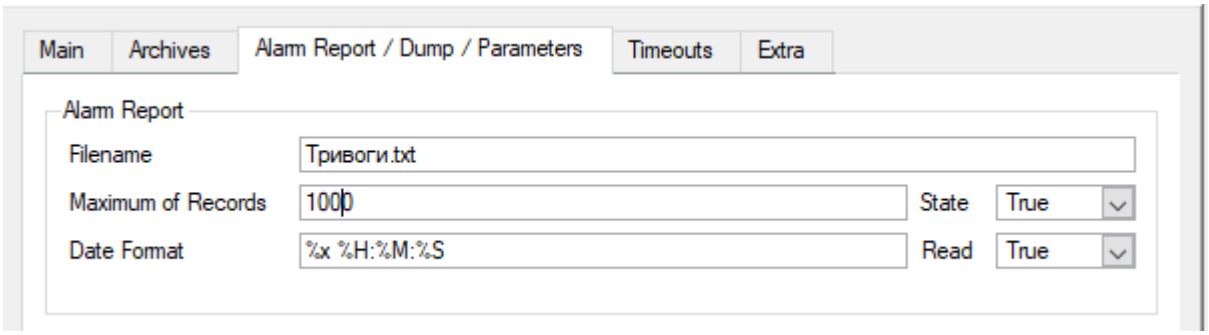


Рисунок 9 – Редагування звіту тривог вузла

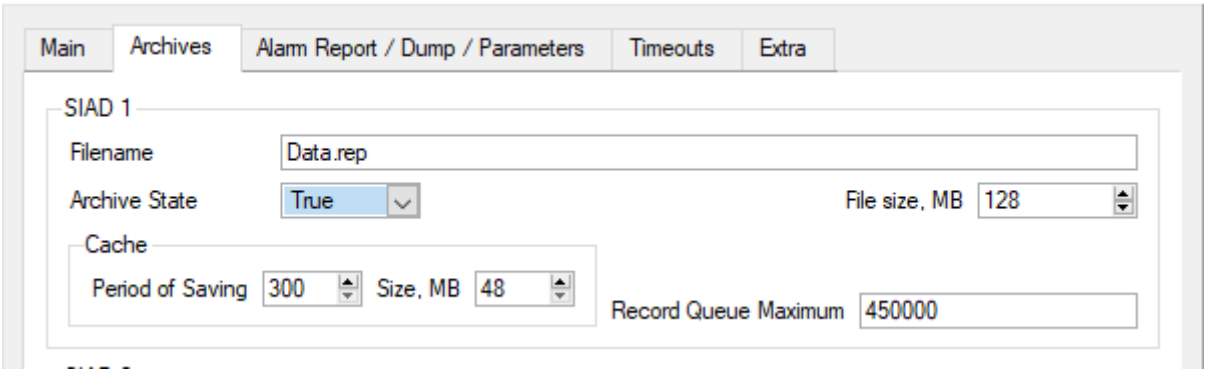


Рисунок 10 – Редагування архіву вузла

1.7 Редагування каналу**.** У вікні редагування каналу **Рівень** вибрати кнопку **Archiving** (Архівація) та налаштувати архівацію і створення звіту тривог, як показано на рис. 11.

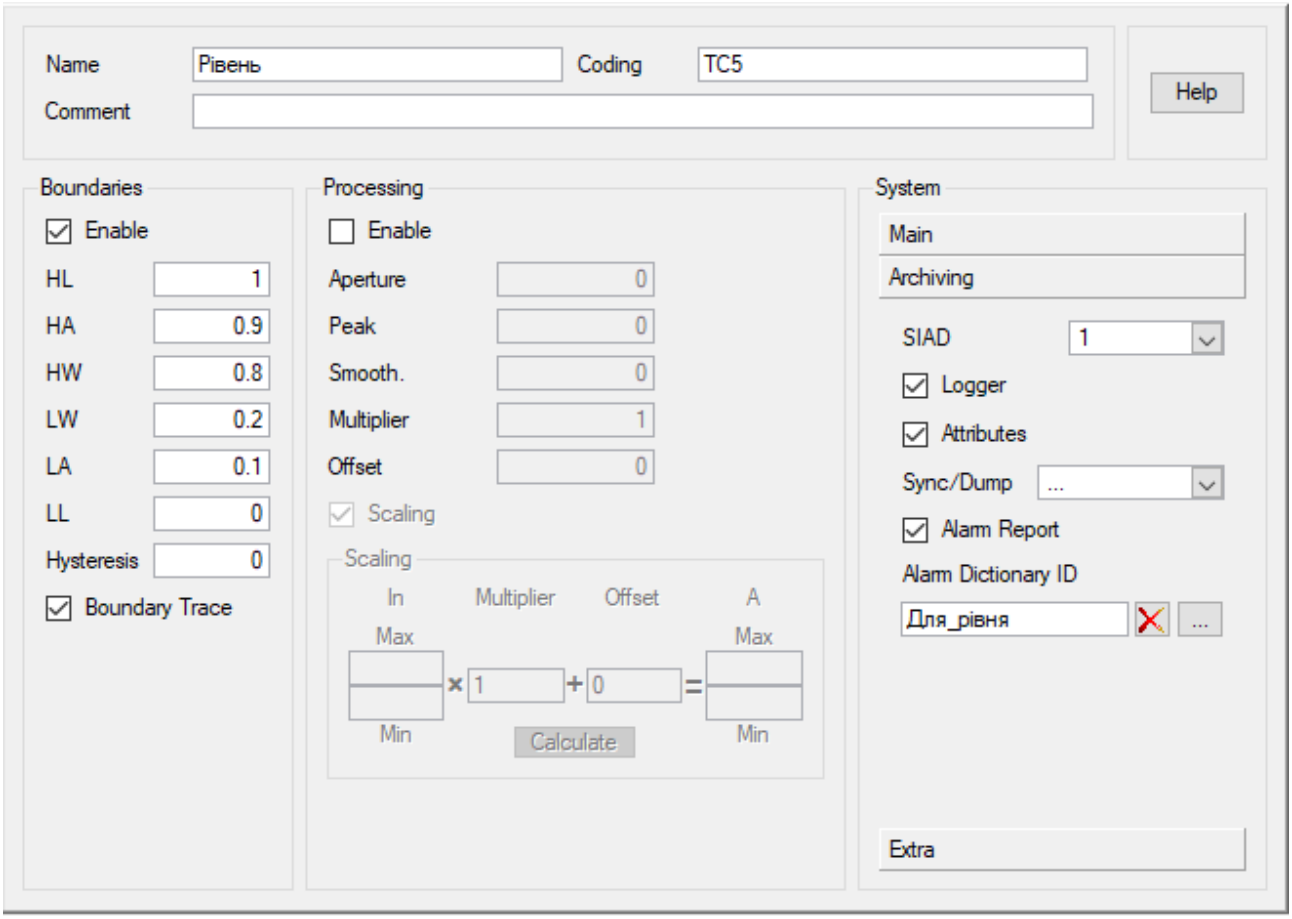


Рисунок 11 – Редагування каналу **Рівень**

1.8 Налаштувати кольори заливання ємності. Відкрити вікно властивостей ГЕ **Багатокутник** (елемент зображення в середині ємності, що забезпечує динамічне заливання). Для отримання залежності кольору заливання від стану контрольованого процесу зробити налаштування, як показано на рис. 12. Зокрема, здійснити прив’язку до аргументу **Рівень**, встановити для **Fill type** (Тип заливання) значення **Color** (Колір), а для **Colors for Ranges** (Кольори для діапазонів) значення **True**, задати бажані кольори заливання діапазонів.

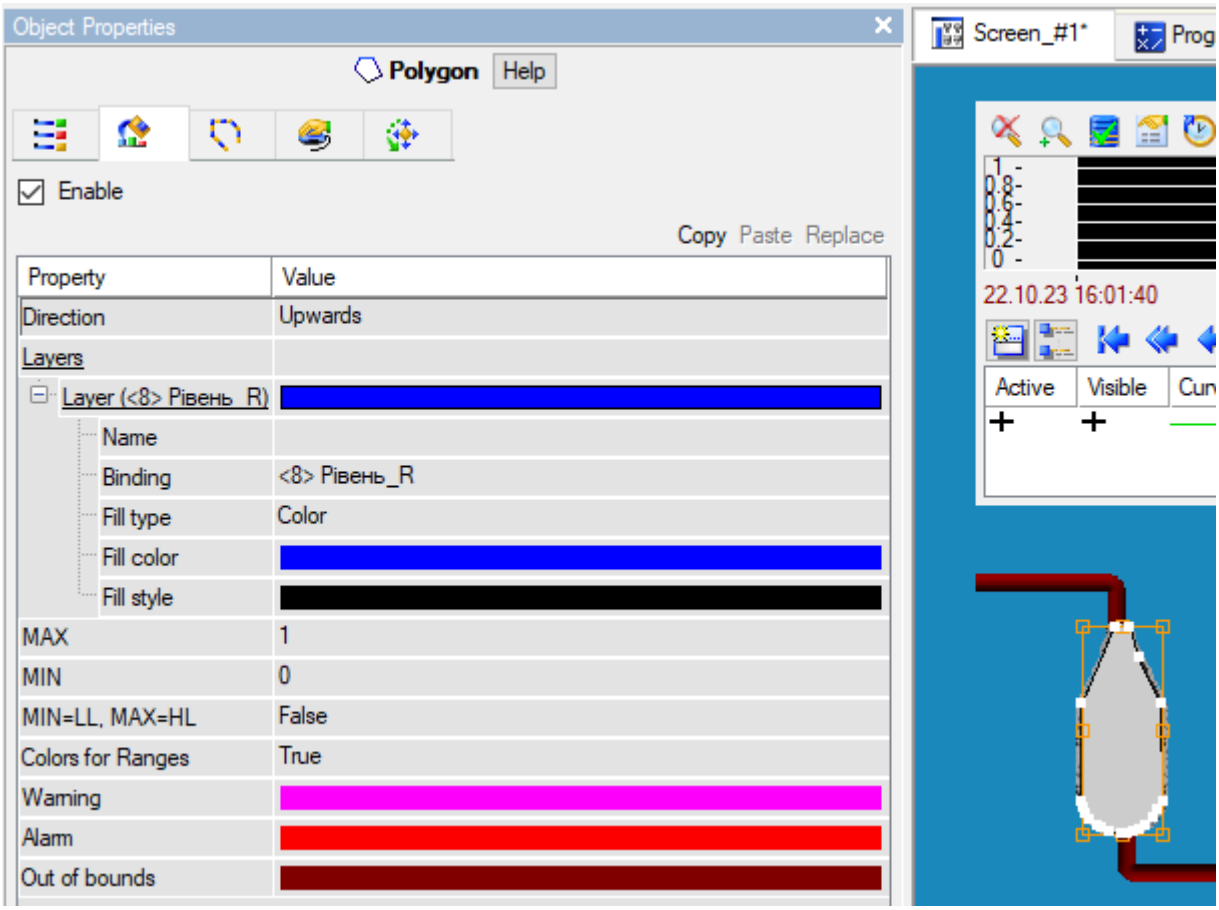


Рисунок 12 – Налаштування зміни кольору динамічного заливання

1.9 Вказати (перевірити) у проекті свої дані.

На графічному екрані розмістити у зручному місці шифр групи та своє ПІБ аналогічно до роботи № 1.

На графічному екрані розмістити у зручному місці своє фото.

В меню **File\Information...**(Файл\Інформація...) в рядку **Owner** (Власник) написати за допомогою клавіатури своє ПІБ, а в рядку **Organization** (Організація) – шифр групи та факультету.

1.10 Запустити проект в роботу. Зберегти проект натисканням кнопки . На головній інструментальній панелі ICP натиснути ЛК кнопку **Save for RTM** (Зберегти для МРЧ), скомпілювавши проект для запуску в реальному часі. Виділити ЛК в навігаторі проекту вузол **RTM_1**, а потім натиснути на головній панелі кнопку **Start Profiler** (Запустить профайлер). У вікні, що

відкрилося, запустити проект на виконання шляхом натискання ЛК кнопки **Run** (Пуск/Зупинка).

За допомогою кнопок ввести вихідні дані. Вигляд графічного екрану показано на рис. 13. Колір заливання ємності визначатиметься рівнем рідини у ємності. Архівний тренд будує криву на основі даних архіву з попередніх запусків проекту.

Переглянути створений звіт тривог у файлі **Тривоги**, який розміщується у папці з ім'ям вузла **RTM_1** в директорії проекту (роботи).

Зупинити виконання проекту шляхом натискання на панелі інструментів ЛК кнопки **Run** (Пуск/Зупинка). Закрити вікно графічного екрану через меню **File\Exit** (Файл\Вихід) або засобами Windows.

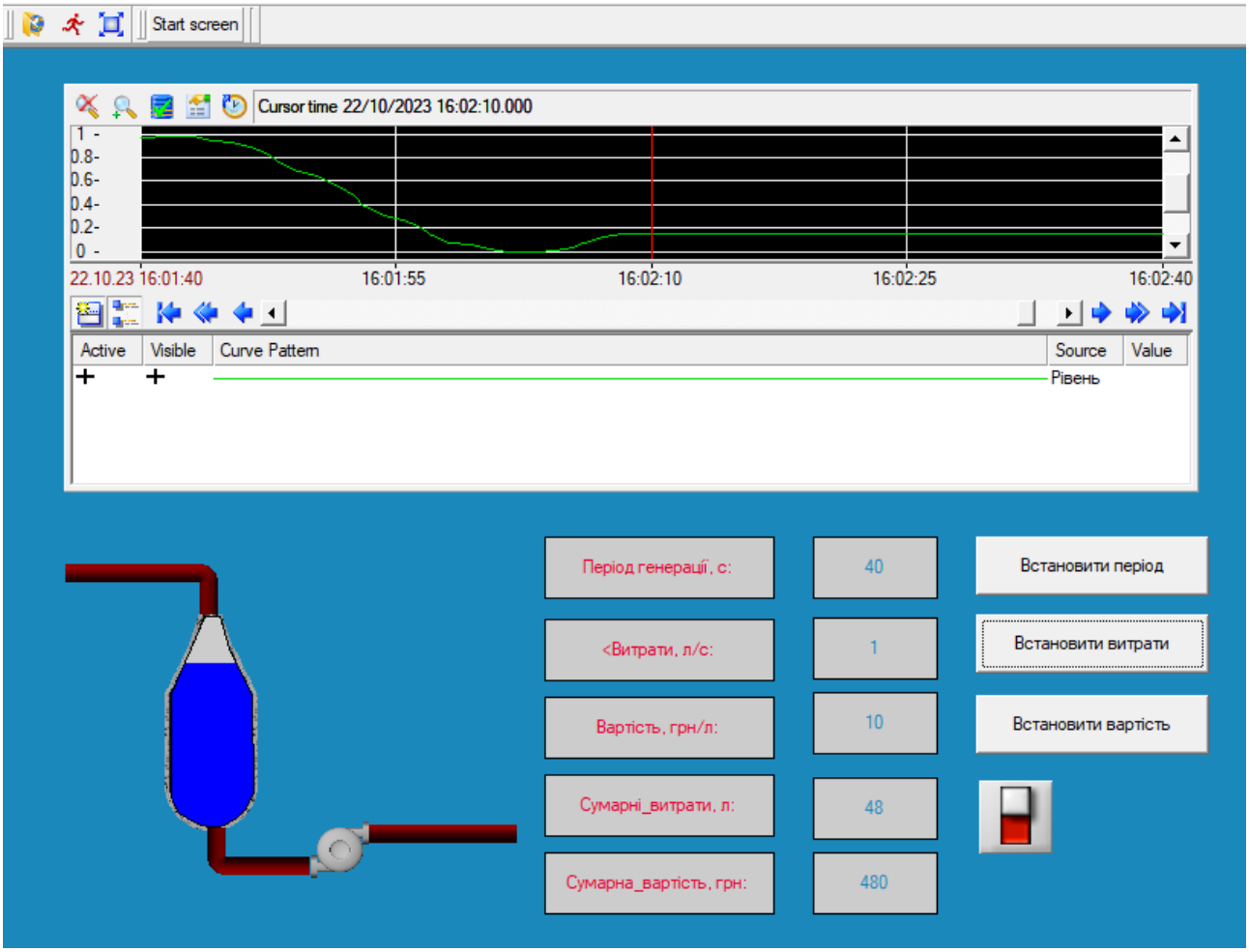


Рисунок 13 – Вигляд графічного екрану проекту

1.11 Оформити протокол комп’ютерного практикуму відповідно до вимог, наведених у вступі.

2 ПИТАННЯ ДЛЯ ЗАХИСТУ КОМП’ЮТЕРНОГО ПРАКТИКУМУ № 9

1. Етапи створення проектів систем автоматизації в SCADA.
2. Завдання властивостей ГЕ «Архівний тренд».
3. Аналогові та дискретні тривоги SCADA-системи.
4. Створення звіту тривог.
5. Словник повідомлень.
6. СПАД-архів і послідовність його створення.

СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Автоматизація виробничих процесів: Підручник / І.В. Ельперін, О.М. Пупена, В.М. Сідлецький, С.М. Швед. – Вид. 2-ге, виправлене.- К.: Ліра-К, 2015. – 378 с.
2. Автоматизовані системи керування технологічними процесами: Підручник / За редакцією І.О. Фурмана. – Харків: Факт, 2006. – 317 с.
3. Островерхов М. Я. Комп'ютерні засоби автоматизації електротехнологічних установок. Конспект лекцій: навч. посіб. для здобувачів ступеня бакалавра за освітньою програмою «Електротехнічні пристрої та електротехнологічні комплекси» спеціальності 141 «Електроенергетика, електротехніка та електромеханіка» / М. Я. Островерхов ; КПІ ім. Ігоря Сікорського. – Київ: КПІ ім. Ігоря Сікорського, 2022. – 222 с.
4. Лукінюк М.В. Автоматизація типових технологічних процесів: технологічні об'єкти керування та схеми автоматизації: Навчальний посібник. – К.: НТУУ "КПІ", 2008. – 236 с.

Додаток А

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Кафедра теоретичної електротехніки

Комп’ютерний практикум № 1
з дисципліни
«Комп’ютерні засоби автоматизації електротехнологічних установок»

СТВОРЕННЯ ПРОЕКТУ СИСТЕМИ АВТОМАТИЗАЦІЇ

Виконав:
студент групи ЕВ-81
Коломійчук Д.С.

Перевірив:
проф. Островерхов М.Я.

Київ 2023