

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

В. А. Волощук, О. В. Степанець, П. В. Гікало

ПРОГРАМУВАННЯ В АВТОМАТИЗОВАНИХ СИСТЕМАХ КЕРУВАННЯ ЛАБОРАТОРНИЙ ПРАКТИКУМ

Навчальний посібник

Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського як навчальний посібник для здобувачів ступеня магістра за спеціальністю G7 Автоматизація, комп'ютерно-інтегровані технології та робототехніка

Електронне мережеве навчальне видання

Київ
КПІ ім. ІГОРЯ СІКОРСЬКОГО
2025

УДК XXX.XX (XXX)

B19

Автори: Волощук Володимир Анатолійович, д-р техн. наук, проф.
Степанець Олександр Васильович, канд. техн. наук, доц.
Гікало Павло Валерійович

Рецензент Вербицький Є. В., д-р техн. наук, проф.,
кафедра електронних пристроїв та систем КПП ім. І. Сікорського

Відповідальний редактор Любицький С. В.

*Гриф надано Методичною радою КПП ім. Ігоря Сікорського
(протокол № 1 від 02.10.2025)
за поданням вченої ради Навчально-наукового інституту атомної та теплової енергетики
(протокол № 13 від 30.06.2025)*

Волощук В. А.

B19 Програмування в автоматизованих системах керування. Лабораторний практикум [Електронний ресурс] : навч. посіб. для здобувачів ступеня магістра за освіт. програмою «Автоматизація та комп'ютерно-інтегровані технології кібер-енергетичних систем» спец. G7 Автоматизація, комп'ютерно-інтегровані технології та робототехніка / В. А. Волощук, О. В. Степанець, П. В. Гікало ; КПП ім. Ігоря Сікорського. – Електрон. текст. дані (1 файл). – Київ : КПП ім. Ігоря Сікорського, 2025. – 183 с.

У представленому навчальному посібнику викладена послідовність розробки програмного забезпечення для автоматизованої системи керування у середовищі TIA Portal. Видання містить весь необхідний матеріал для створення реального проєкту автоматизації, наведено декілька типових схем керування технологічними процесами, приділено увагу ефективним методам розробки програмного забезпечення та запропоновані завдання для додаткового самостійного опрацювання матеріалу. Посібник призначений для здобувачів ступеня магістра за спеціальністю G7 Автоматизація, комп'ютерно-інтегровані технології та робототехніка, буде також корисним для студентів інших спеціальностей, освітні програми яких передбачають розробку програмного забезпечення для програмованих логічних контролерів Siemens.

УДК XXX.XX (XXX)

Реєстр. № НП 25/26-008. Обсяг 3,8 авт. арк.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
проспект Берестейський, 37, м. Київ, 03056
<https://kpi.ua>

Свідоцтво про внесення до Державного реєстру видавців, виготовлювачів
і розповсюджувачів видавничої продукції ДК № 5354 від 25.05.2017 р.

© В. А. Волощук, О. В. Степанець, П. В. Гікало, 2025
© КПП ім. Ігоря Сікорського, 2025

ЗМІСТ

1	Лабораторна робота №1.	5
1.1	Завдання	5
1.2	Опис установки для автоматизації	5
1.3	Порядок виконання роботи	7
1.3.1	Налаштування проєкту	7
1.3.2	Програмування алгоритму ручного режиму	13
1.3.3	Програмування захисту та індикації роботи насосу	24
1.3.4	Програмування алгоритму автоматичного режиму	26
1.3.5	Завантаження та відлагодження коду	31
1.4	Контрольні завдання	46
2	Лабораторна робота №2.	47
2.1	Завдання	47
2.2	Опис установки для автоматизації	47
2.3	Порядок виконання роботи	48
2.3.1	Налаштування проєкту	48
2.3.2	Вибір моделі контролера	49
2.3.3	Програмування алгоритмів увімкнення частотного перетворювача насоса	52
2.3.4	Робота з аналоговими значеннями	56
2.3.5	Регулювання технологічного параметра	61
2.4	Контрольні завдання	80
3	Лабораторна робота №3.	81
3.1	Завдання	81
3.2	Порядок виконання роботи	81
3.2.1	Створення архіву проєкта	81
3.2.2	Створення документації проєкта	84
3.2.3	Створення бібліотеки розроблених програмних компонентів	85
3.2.4	Заміна процесорного модуля у проєкті	92
3.2.5	Створення архіву бібліотеки	95
3.3	Контрольні завдання	99
4	Лабораторна робота №4.	100
4.1	Завдання	100
4.2	Порядок виконання роботи	100

4.2.1	Підготовка проєкту до імітаційного моделювання _____	100
4.2.2	Налаштування мережного підключення контролерів _____	103
4.2.3	Розробка алгоритму обміну даними між контролерами _____	105
4.2.4	Розробка алгоритму імітаційної моделі об'єкта керування _____	113
4.2.5	Проведення імітаційного моделювання системи регулювання _____	118
4.3	Контрольні завдання _____	126
5	<i>Лабораторна робота №5. _____</i>	<i>128</i>
5.1	Завдання _____	128
5.2	Порядок виконання роботи _____	128
5.2.1	Налаштування підключення панелі оператора до ПЛК _____	128
5.2.2	Відображення на екрані основних технологічних параметрів системи регулювання тиску _____	137
5.2.3	Реалізація можливості пуску та зупинки насосу в ручному режимі за допомогою панелі оператора _____	159
5.2.4	Відображення графіків перехідних процесів _____	164
5.2.5	Реалізація можливості налаштування параметрів регулятора за допомогою панелі оператора _____	170
5.2.6	Перевірка роботи системи з панеллю оператора _____	177
5.3	Контрольні завдання _____	182
	<i>Перелік рекомендованої літератури. _____</i>	<i>183</i>

1 Лабораторна робота №1.

Мета роботи: ознайомитись з інженерною мовою програмування LD; отримати навички програмування релейно-контактних схем на мові LD.

1.1 Завдання

1. Вивчити процедуру створення проєкту в інтегрованому середовищі розробки TIA Portal.
2. Розробити програму для ПЛК, яка керує насосом, яка повинна:
 - 2.1. Обирати ручний або автоматичний режим роботи насоса;
 - 2.2. У ручному режимі керувати насосом за допомогою кнопок «Пуск» і «Стоп»;
 - 2.3. В автоматичному режимі керувати насосом по показниках датчиків рівня рідини в баку;
 - 2.4. Реалізувати захист від перегріву двигуна насоса.
3. Модифікувати проєкт відповідно до контрольних завдань.
4. Дослідити роботу розробленого програмного проєкту.

1.2 Опис установки для автоматизації

Установка представляє собою бак, в якому накопичується рідина. Періодично, по наповненню баку, рідину треба відкачувати насосом до повного спорожнення баку. Бак обладнаний двома дискретними датчиками рівня – верхнім і нижнім. Насос вмикається контактором по сигналу від ПЛК. Насос обладнано датчиком перегріву двигуна – термореле.

Щит керування насосом обладнаний засобами людино-машинної взаємодії: трипозиційним перемикачем вибору режиму роботи «Авто/0/Ручний», кнопками «Пуск» і «Стоп», світловим індикатором стану команди увімкнення насоса.

У ручному режимі насос вмикається кнопками «Пуск» і «Стоп». В автоматичному режимі насос працює по показниках датчиків рівня: при спрацюванні датчика високого рівня насос вмикається, за відсутності сигналу від датчика нижнього рівня – вимикається.

Завжди, коли на дискретний вихід ПЛК, пов'язаний з контактором увімкнення насоса, надходить команда, вмикається світловий індикатор.

Перелік вхідних та вихідних сигналів ПЛК можна знайти нижче в таблиці 1.1.

Таблиця 1.1 Входи та виходи ПЛК

№ п/п	В/В	Тип В/В	Опис
1	SA1_1	Дискретний вхід	Перемикач «Авто/0/Ручний» у положенні «Авто»
2	SA1_2	Дискретний вхід	Перемикач «Авто/0/Ручний» у положенні «Ручний»
3	SB1	Дискретний вхід	Кнопка «Пуск» у ручному режимі
4	SB2	Дискретний вхід	Кнопка «Стоп» у ручному режимі
5	TK1	Дискретний вхід	Контакт термореле двигуна насоса (нормально замкнений)
6	LS1	Дискретний вхід	Нижній датчик-реле рівня рідини
7	LS2	Дискретний вхід	Верхній датчик-реле рівня рідини
8	KM1	Дискретний вихід	Котушка контактора увімкнення насоса
9	HL1	Дискретний вихід	Світловий індикатор команди увімкнення насоса

У результаті виконання роботи повинен бути створений код керування насосом мовою **LAD (Ladder Logic)**.

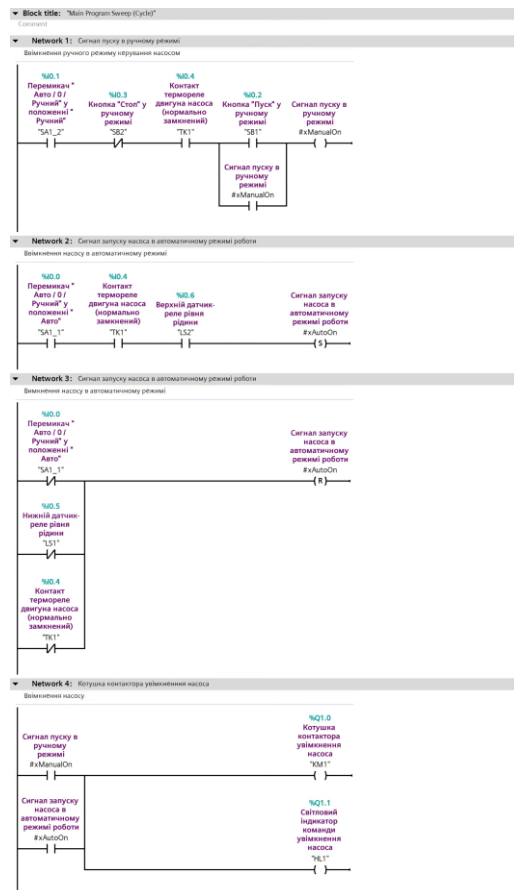


Рисунок 1.1 Алгоритм керування насосом.

1.3 Порядок виконання роботи

1.3.1 Налаштування проєкту

1. Запустити середовище розробки
2. Створити новий проєкт **Create new project**

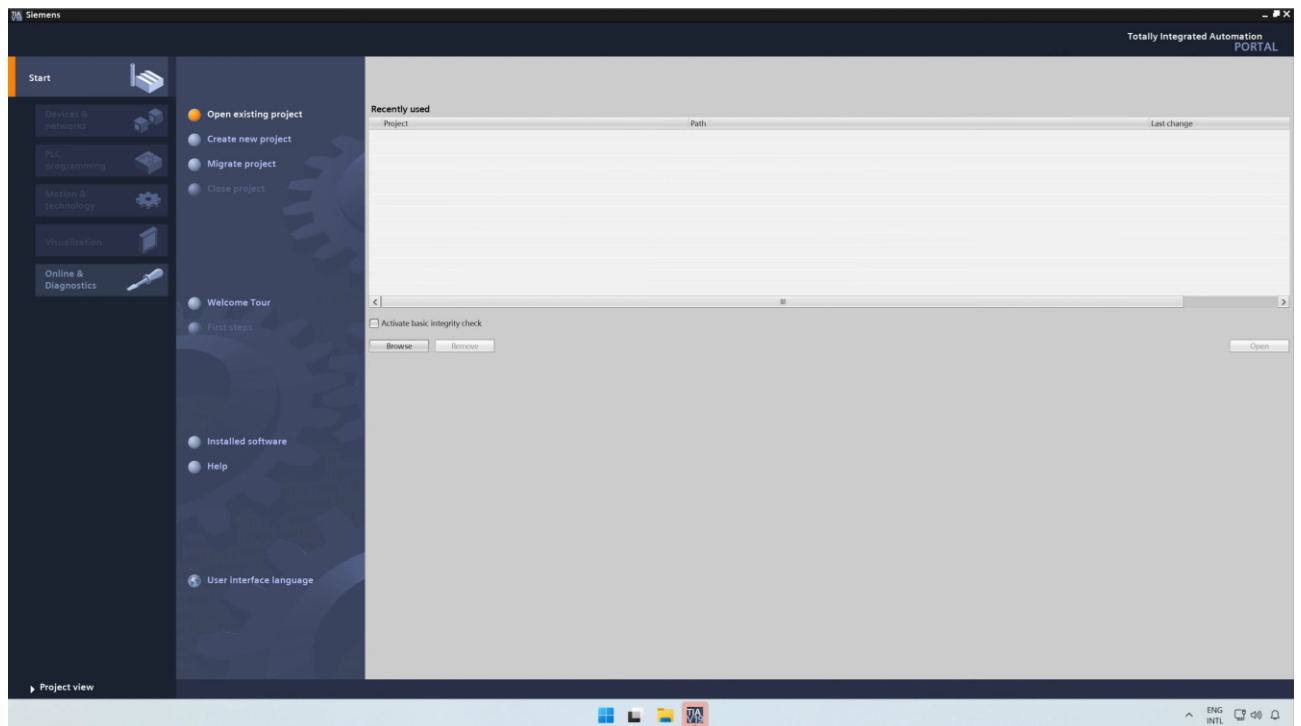


Рисунок 1.2 Створення нового проєкту TIA Portal.

- У вікні налаштувань проєкту, яке відкрилося, дати ім'я проєкту **Lab_1**, обрати місце розміщення файлів проєкту в полі **Path**, записати своє ім'я в полі **Author** та завдання у полі **Comment**.

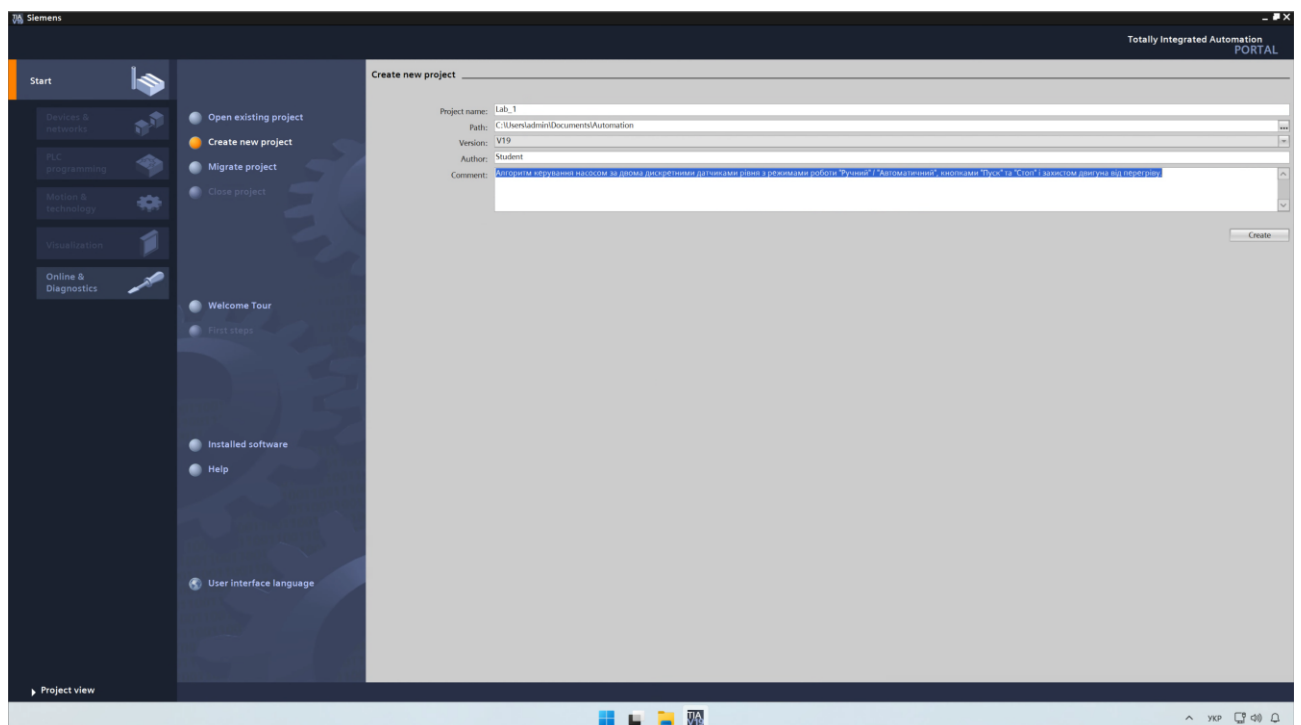


Рисунок 1.3 Внесення назви та даних проєкту.

- Перейти від режиму відображення порталу до режиму відображення проєкту TIA Portal.

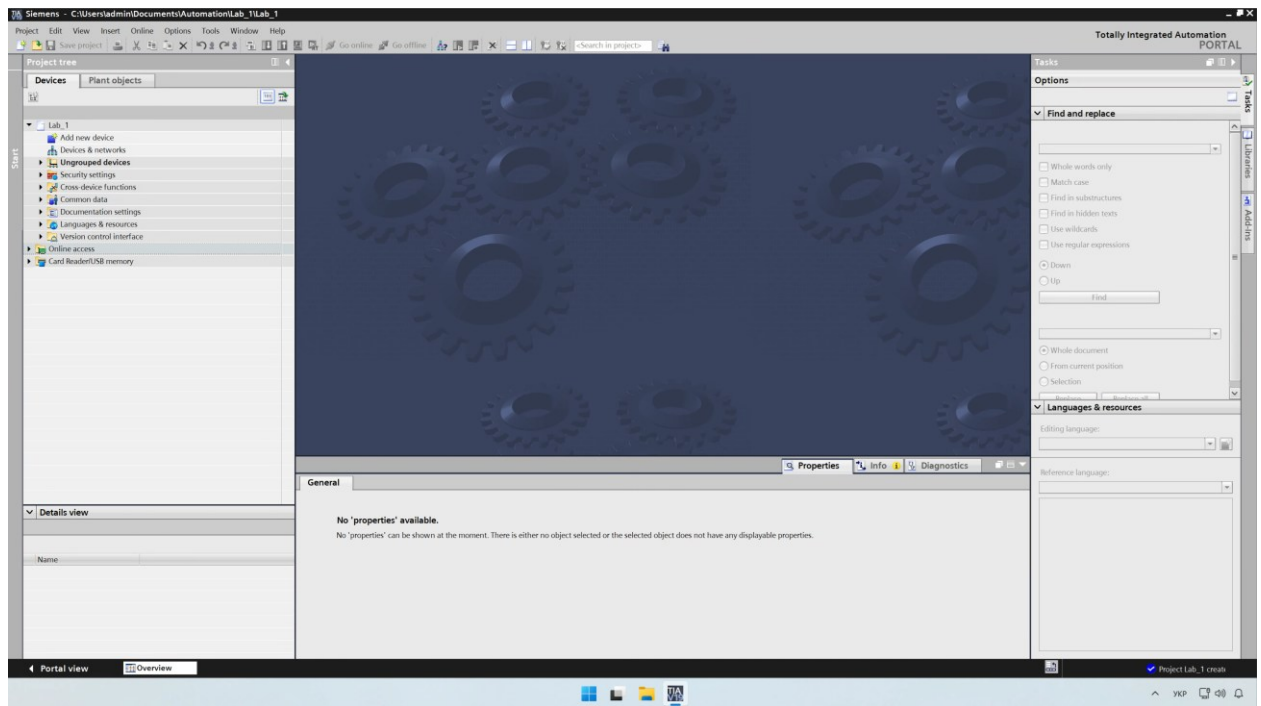


Рисунок 1.4 Режим відображення проєкту TIA Portal.

На цьому етапі розробки в проєкті є лише початкова структура з налаштуваннями за замовчуванням та без жодного пристрою. Для нашого проєкту необхідно додати програмований логічний контролер та виконати його конфігурування. З урахуванням рівня складності поставленої задачі, кількості та типів сигналів для цієї системи автоматизації можна обрати процесорний модуль **CPU 312C** з артикулом **6ES7 312-5BE03-0AB0**. Цей процесорний модуль має вбудовані дискретні канали вводу/виводу, тому для вирішення поставленої задачі не потрібно буде використовувати додаткових модулів розширення.

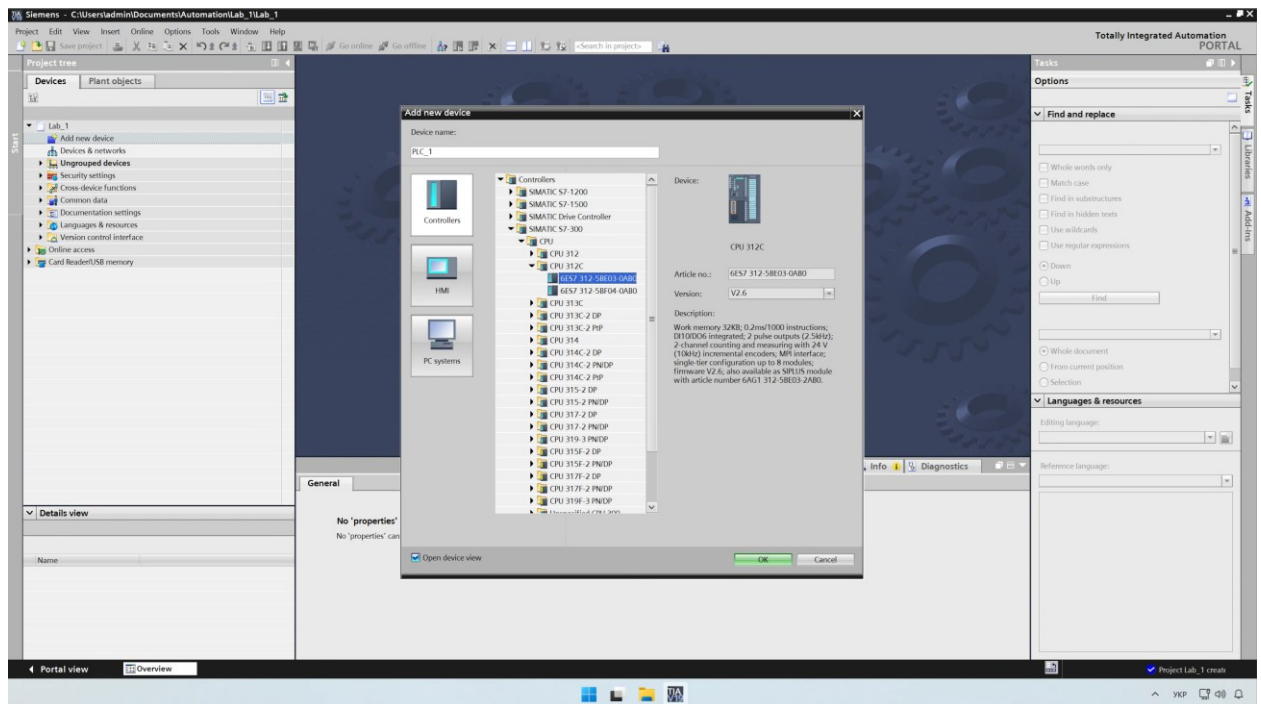


Рисунок 1.5 Вибір ПЛК для проєкту.

Після вибору ПЛК необхідно виконати його конфігурування. Конфігурування у загальному випадку передбачає налаштування рівня безпеки, встановлення мережних адрес для всіх пристроїв системи, типів сигналів для каналів вводу/виводу, які відповідають типам сигналів датчиків та виконавчих пристроїв системи, за необхідності – призначення назв, фільтрації та адрес для каналів вводу/виводу та інших параметрів.

Налаштування рівня безпеки контролера відбувається у розділі **Protection** вкладки **General** параметрів налаштувань контролера. Відкрити цей розділ можна за допомогою пункту **Properties** контекстного меню, яке з'являється за правим кліком на назві контролера у дереві проєкта TIA Portal.

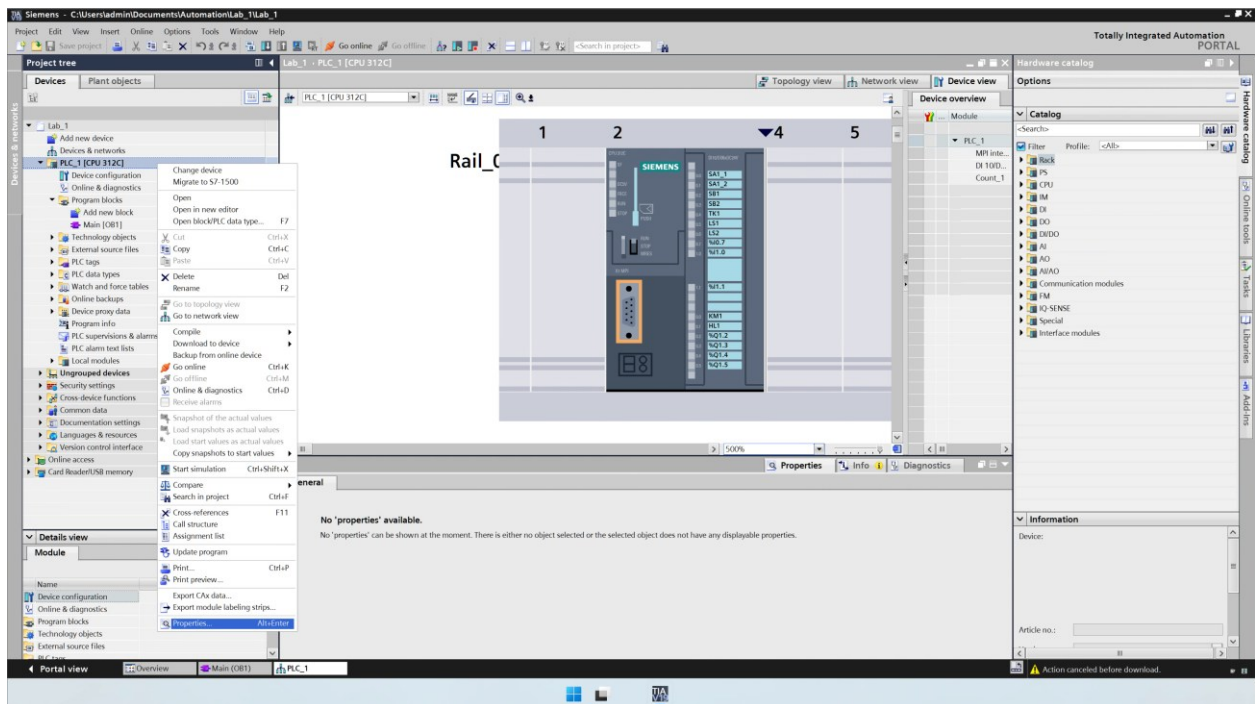


Рисунок 1.6 Відкриття вікна налаштувань контролера.

За замовчуванням, TIA Portal пропонує встановити максимальний рівень безпеки, який передбачає неможливість підключення до контролера без введення пароля. Такий рівень безпеки є рекомендованим для забезпечення захисту від зовнішнього несанкціонованого втручання у систему автоматизації. На етапі розробки алгоритму та його перевірки можна встановити мінімальний рівень безпеки, який дозволяє підключення та зміну програми контролера без необхідності введення паролю. Після завершення розробки алгоритму та для систем, які можуть зазнавати спроб зовнішнього втручання, рівень безпеки має бути повернений на максимальний зі встановленням надійного пароліного захисту.

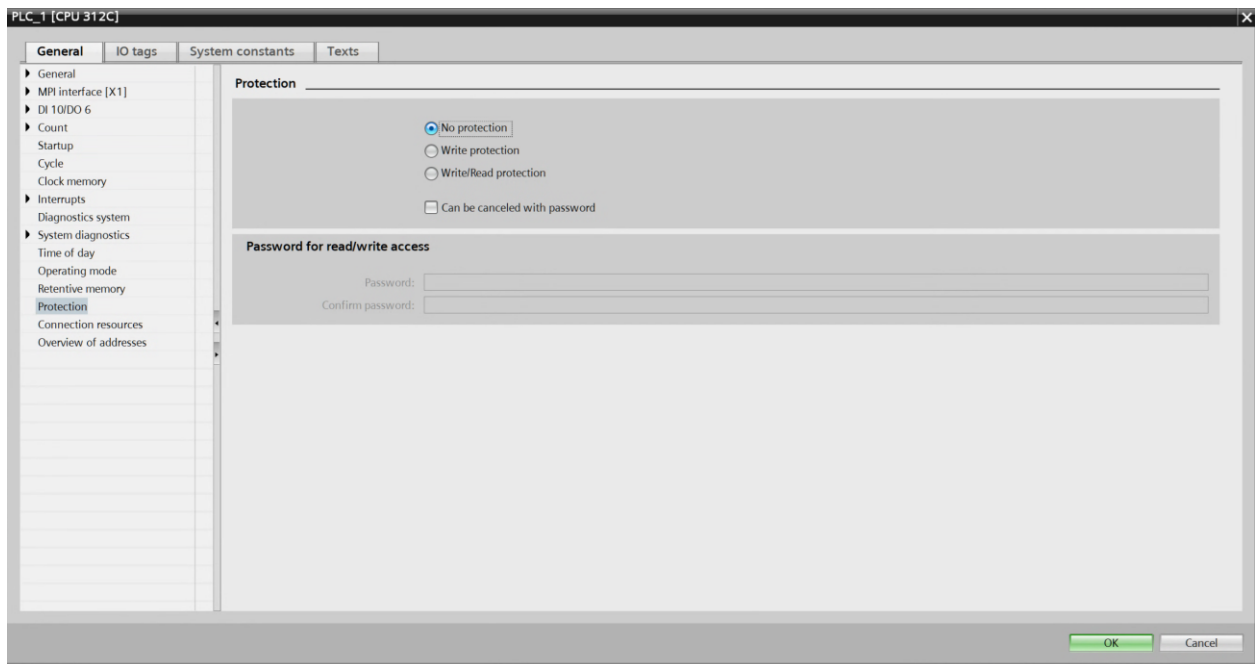


Рисунок 1.7 Зміна налаштувань рівня безпеки контролера.

Після встановлення налаштувань безпеки необхідно позначити канали вводу/виводу контролера, до яких буде здійснюватись підключення датчиків, органів ручного керування та виконавчих пристроїв. Ці налаштування можна змінити на вкладці IO tags вікна редагування параметрів контролера. Для більш зручної роботи з кодом призначимо назви каналів вводу та виводу та запишемо у коментарях призначення цих каналів відповідно до [таблиці 1.1](#) . При цьому потрібно не забувати про те, що до роботи з алгоритмом часто доводиться повертатись для внесення змін або пошуку помилок після тривалої перерви, тож для спрощення процесу розуміння алгоритму та прискорення роботи з ним гарною практикою є коментування всіх рішень, реалізованих в алгоритмі та позначення всіх змінних та сигналів, які в ньому використовуються, зрозумілим чином. TIA Portal дає можливість створювати коментарі до кожного елементу будь-якою мовою, тому назви каналів встановлюємо відповідно до завдання у полях **Name**, а пояснення записуємо у полях **Comment**.

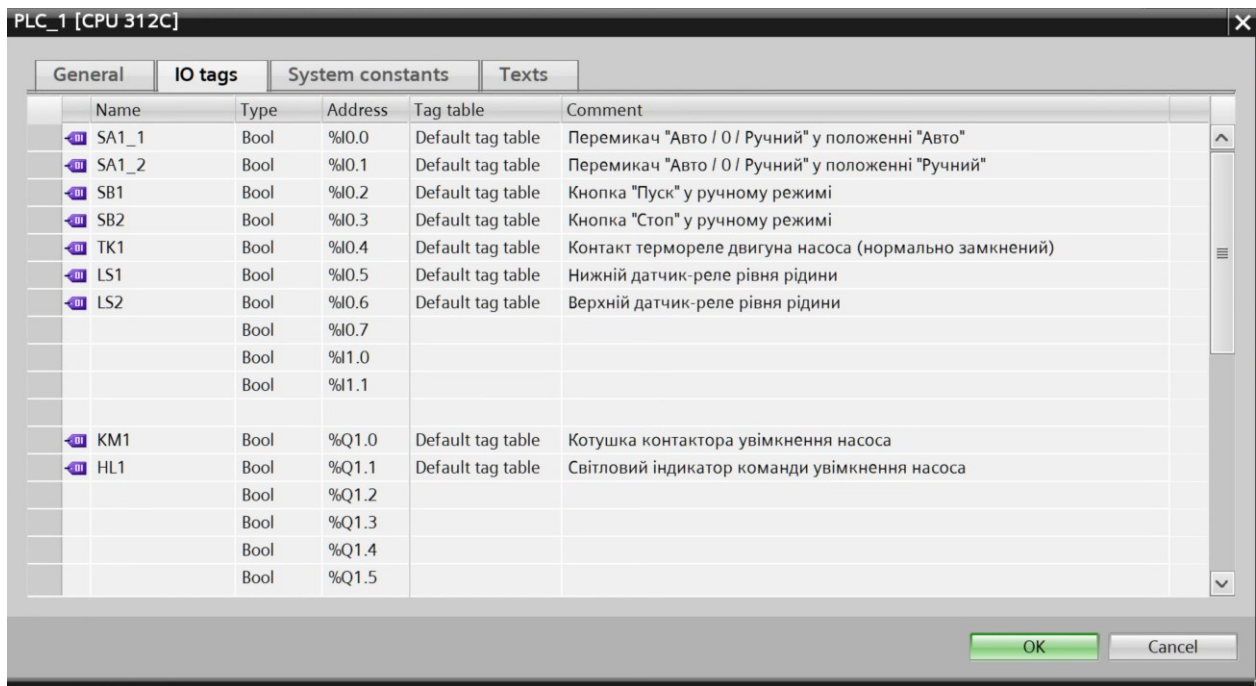


Рисунок 1.8 Позначення входів та виходів контролера.

Після виконання конфігурування введенні назви каналів можна буде побачити на візуалізації контролера.

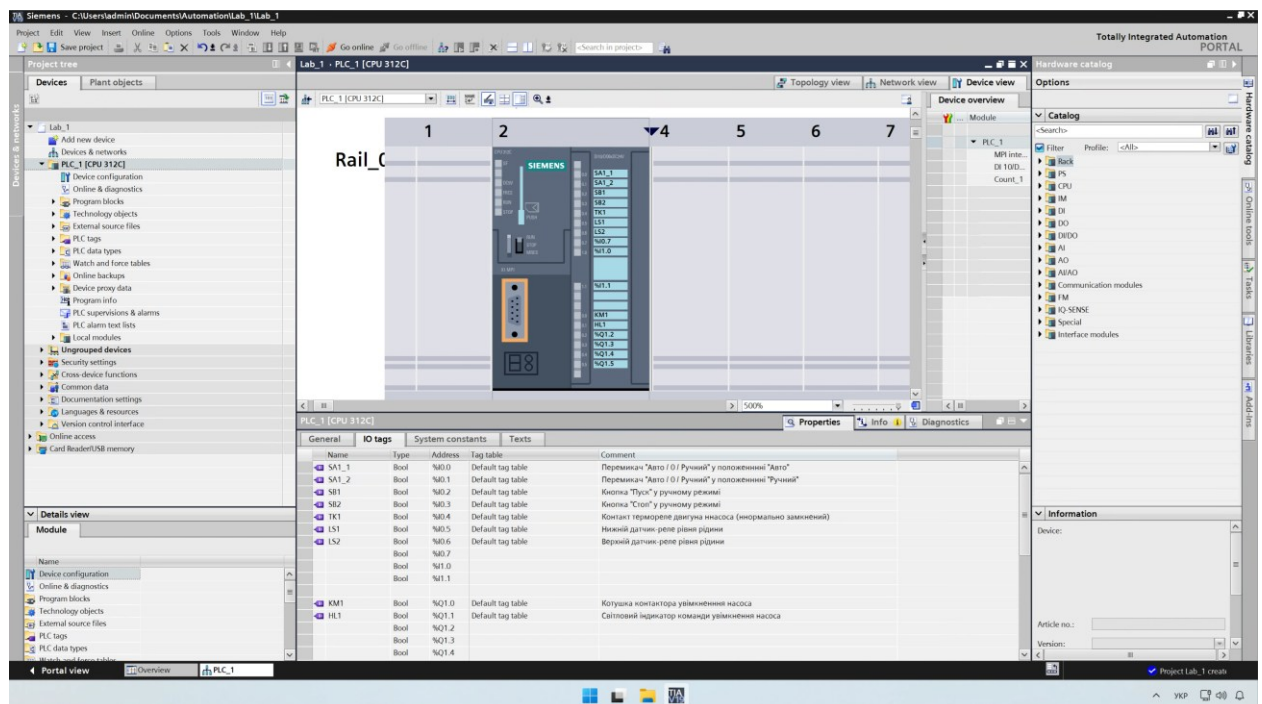


Рисунок 1.9 Вигляд передньої панелі контролера з позначеними каналами вводу/виводу.

1.3.2 Програмування алгоритму ручного режиму

За замовчуванням TIA Portal при створенні нового проєкту створює головний організаційний блок **Main [OB1]**, в якому користувач може мовою **LAD (Ladder**

Logic) створити алгоритм, що буде у нескінченному циклі виконуватись контролером.

Для редагування алгоритму необхідно у дереві проекту обрати контролер, перейти у розділ **Program blocks** та двічі клацнути на організаційному блоці **Main [OB1]**. При цьому має відкритись вікно редактору коду мовою LAD. За замовчуванням в редакторі коду вже буде показана перша секція Network 1 алгоритму у вигляді релейно-контактної схеми.

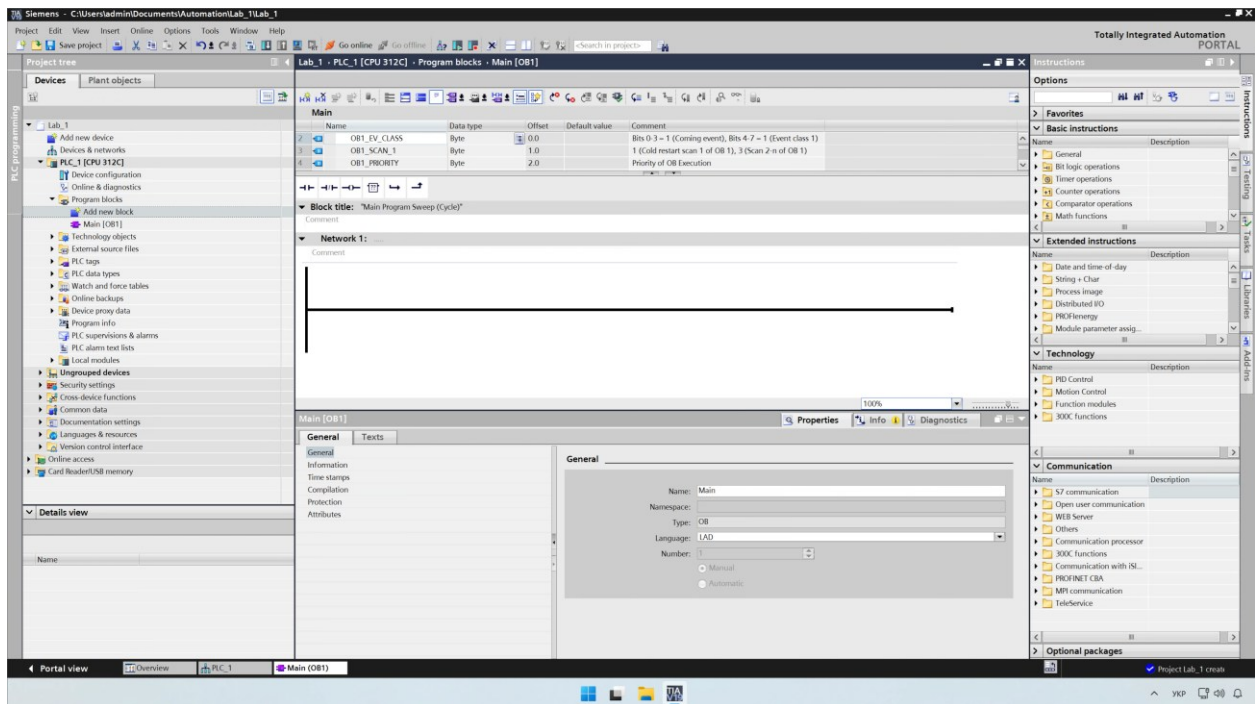


Рисунок 1.10 Редагування алгоритму мовою LAD (Ladder Logic).

1. Вставити нормально розімкнений контакт в електричне коло секції **Network 1**, вибравши елемент **Normally open contact** на панелі швидкого доступу.

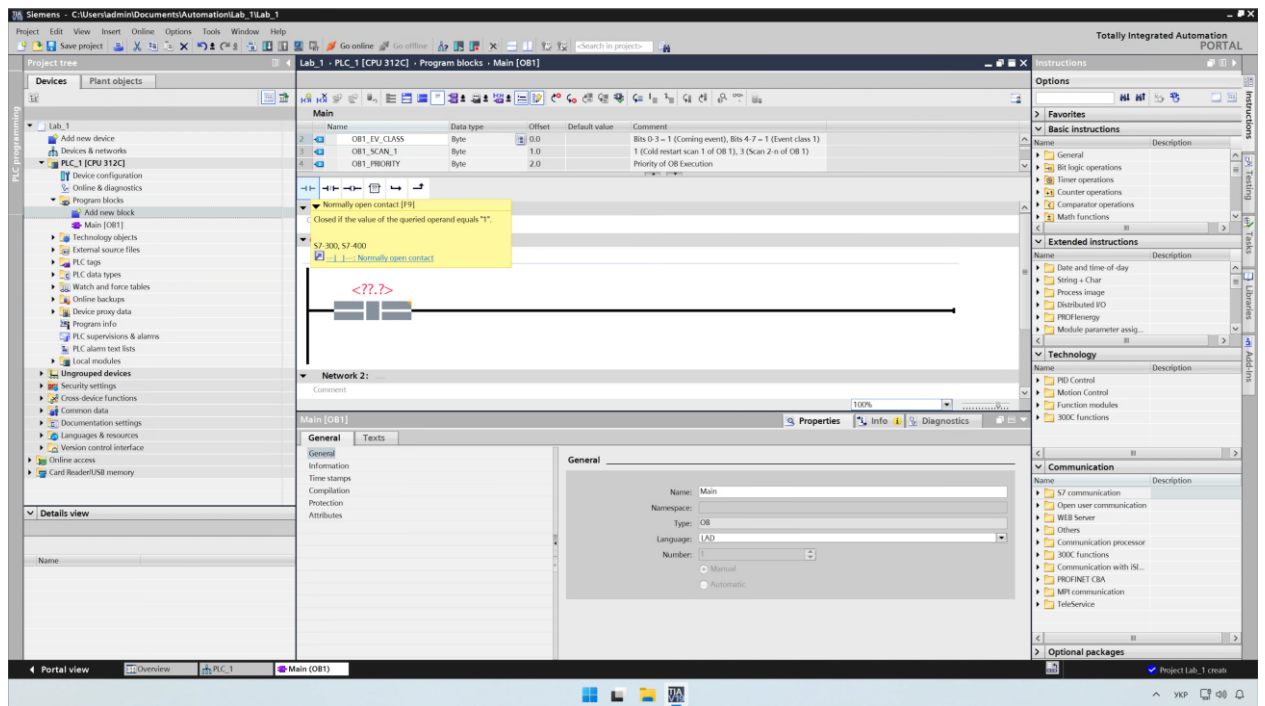


Рисунок 1.11 Додавання нормально розімкненого контакту.

Замість символів <???.?> потрібно записати назву, яку ми дали відповідному входу контролера, до якого має бути підключений перемикач «Авто/0/Ручний» у положенні «Ручний»:

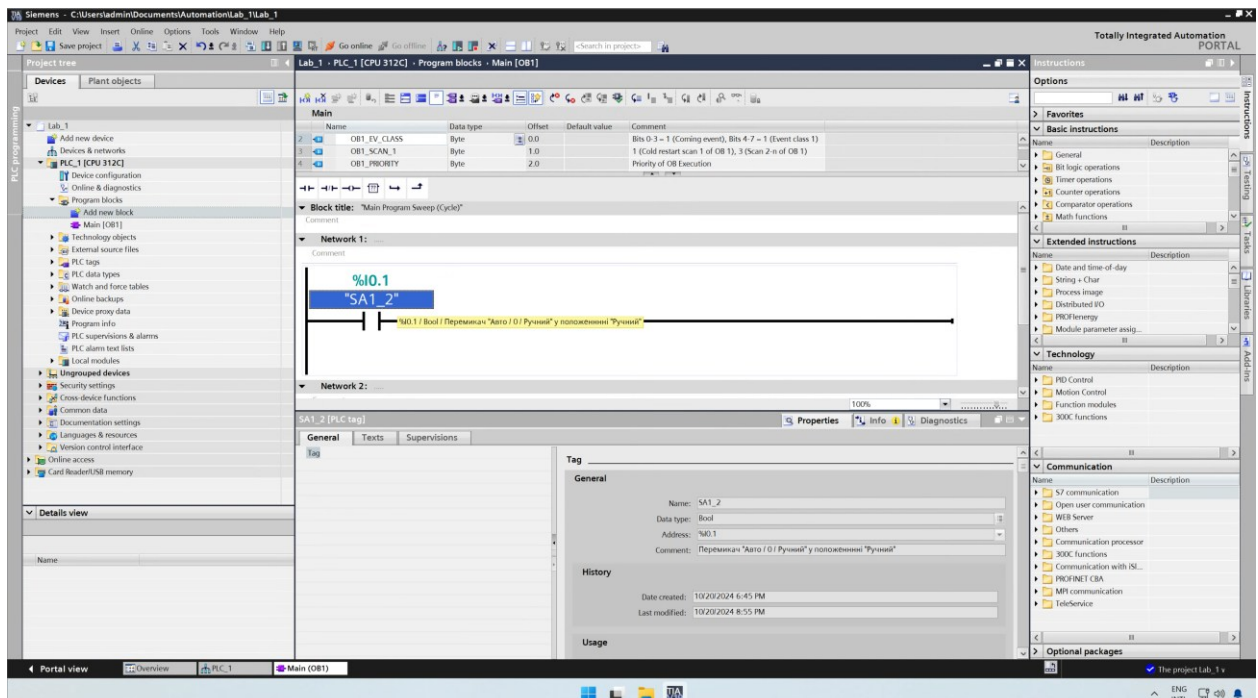


Рисунок 1.12 Додавання нормально розімкненого контакту SA1_2.

У результаті над контактом з'явилася введене найменування, а над ним – адреса фізичного дискретного входу контролера, до якого має бути підключена відповідна кнопка.

2. Приєднати до контакту праворуч котушку реле Assignment

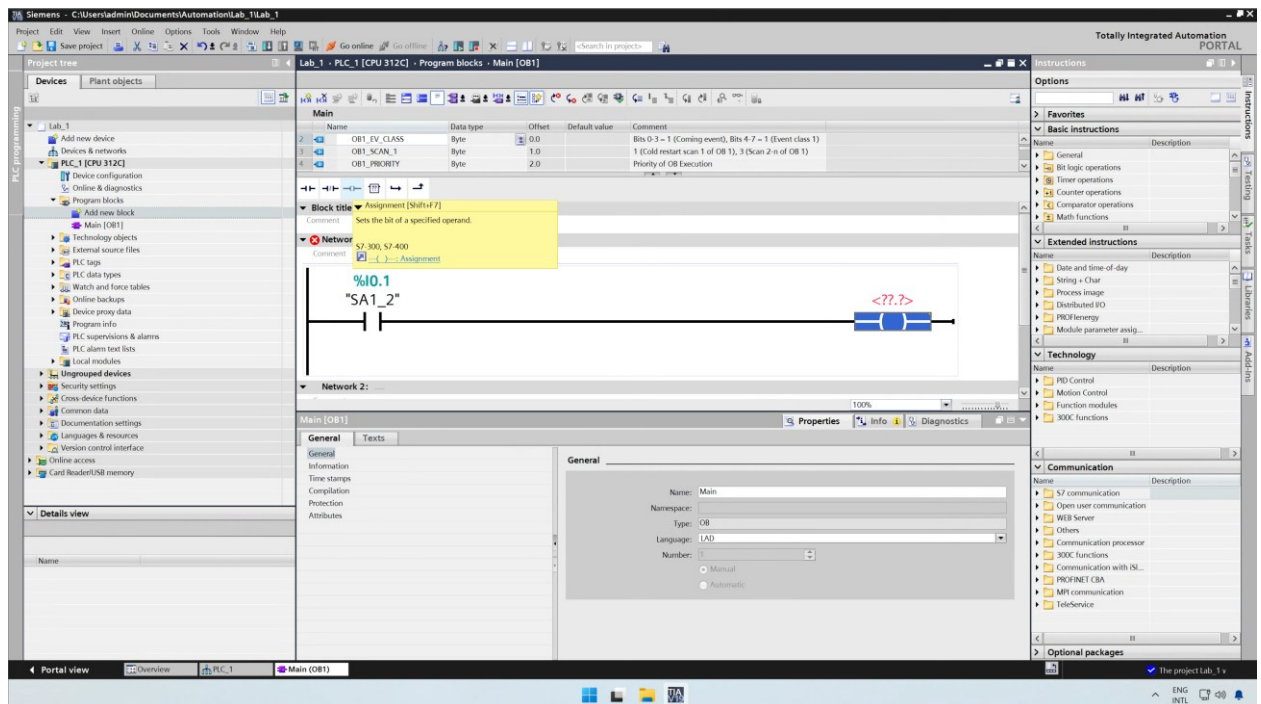


Рисунок 1.13 Додавання котушки реле.

3. Присвоїти котушці іменування xManualOn, написавши це замість символів <??.?> над котушкою.

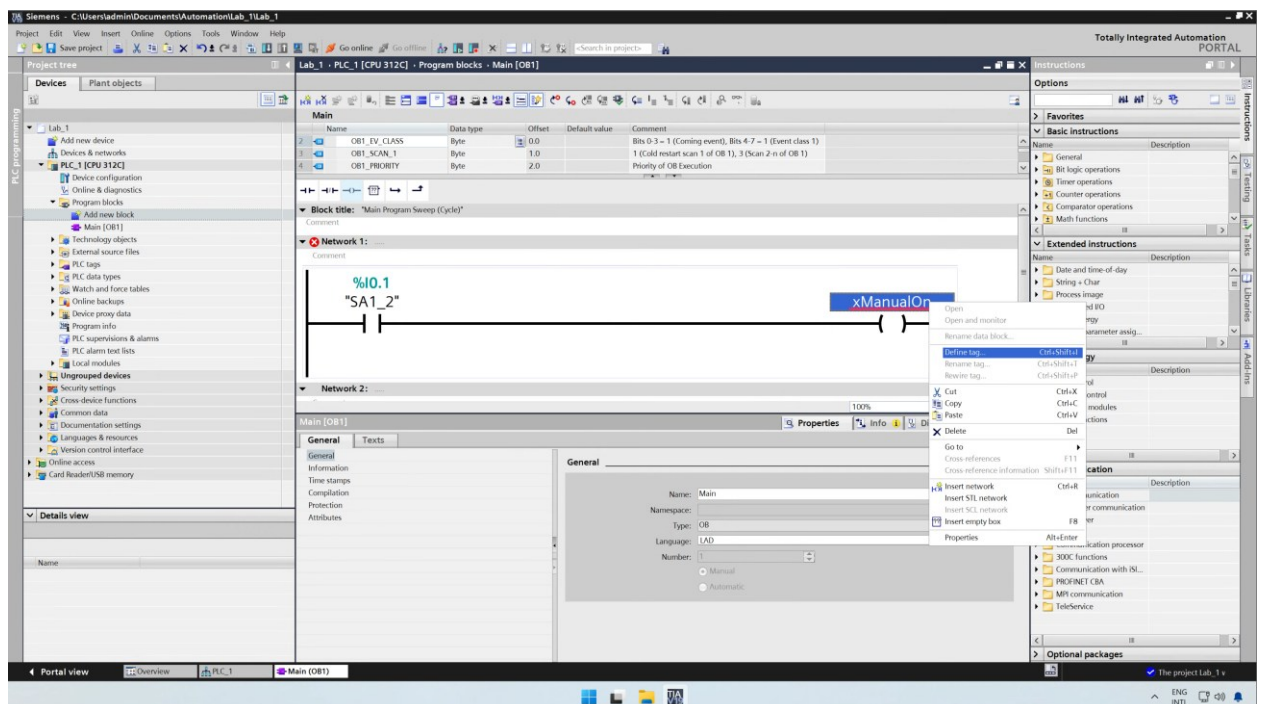


Рисунок 1.14 Створення змінної xManualOn.

Оскільки змінної з такою назвою не існує, TIA Portal позначатиме її назву червоним хвилястим підкресленням. Для оголошення локальної змінної з такою назвою необхідно клацнути правою клавішею миші на її назві та обрати у контекстному меню пункт **Define tag**. У вікні, яке з'явиться, переконатися, що у полі **Section** встановлене значення **Local Temp**, а у полі **Data type** - тип змінної **BOOL**. У полі **Comment** записати призначення створеної змінної «Команда пуску насосу в ручному режимі» та підтвердити внесені дані натисканням на кнопку «**Define**».

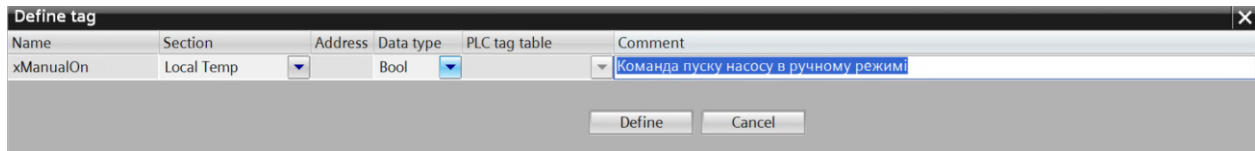


Рисунок 1.15 Оголошення змінної *xManualOn*.

У результаті над котушкою реле з'явиться введене найменування без підкреслення, а в реакторі інтерфейсу організаційного блоку буде додана змінна *xManualOn* типу **BOOL**.

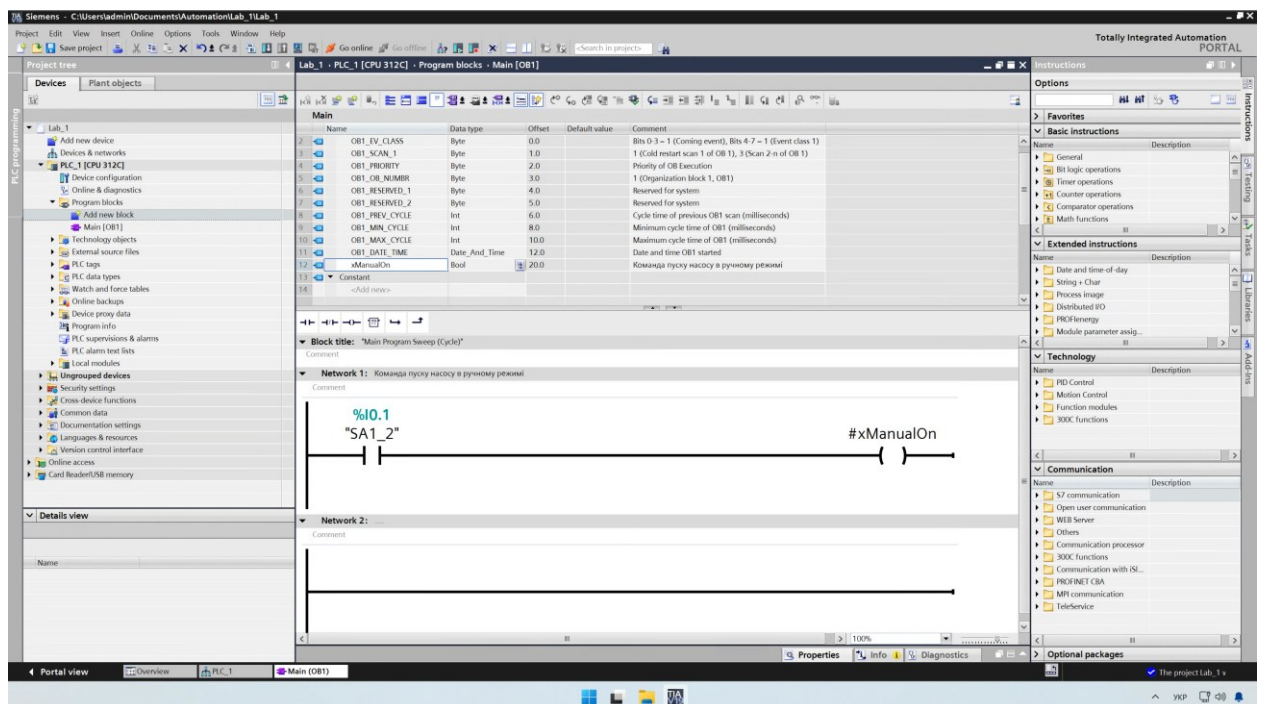


Рисунок 1.16 Результат оголошення локальної змінної *xManualOn*.

Тепер готовий мінімально функціональний ланцюг. Стан контакту SA1_2 впливатиме на стан котушки реле *xManualOn*: коли SA1_2 замкнений (система в режимі «Ручний»), то реле *xManualOn* активується.

Переходимо до редагування наступної секції коду. У ній міститиметься схема, яка керуватиме станом пов'язаного з насосом дискретного виходу. Насос запускатиметься, коли буде активована котушка реле КМ1.

4. Аналогічно до виконаних раніше дій, утворити електричне коло, у якому контакт, пов'язаний з реле **xManualOn**, впливатиме на реле увімкнення насосу **KM1**.

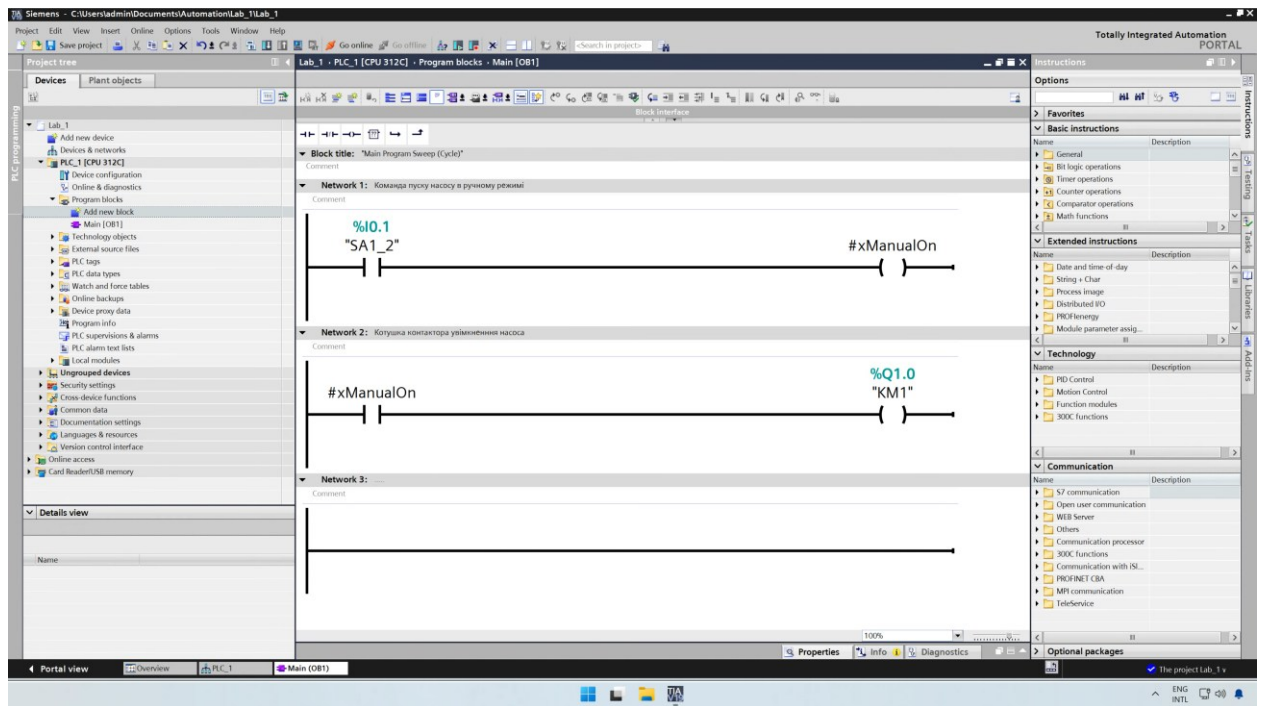


Рисунок 1.17 Редагування схеми ввімкнення насосу.

Коли програма стає більшою, аналізувати код стає складніше. Використання підказок для змінних, каналів вводу/виводу та частин коду може суттєво допомогти у розумінні коду та пошуку причин некоректної роботи. TIA Portal надає можливість відображати підказки безпосередньо на області редагування програми. Для цього можна скористатись головним меню TIA Portal **View → Display with → Tag information → Tag information with hierarchical comments**

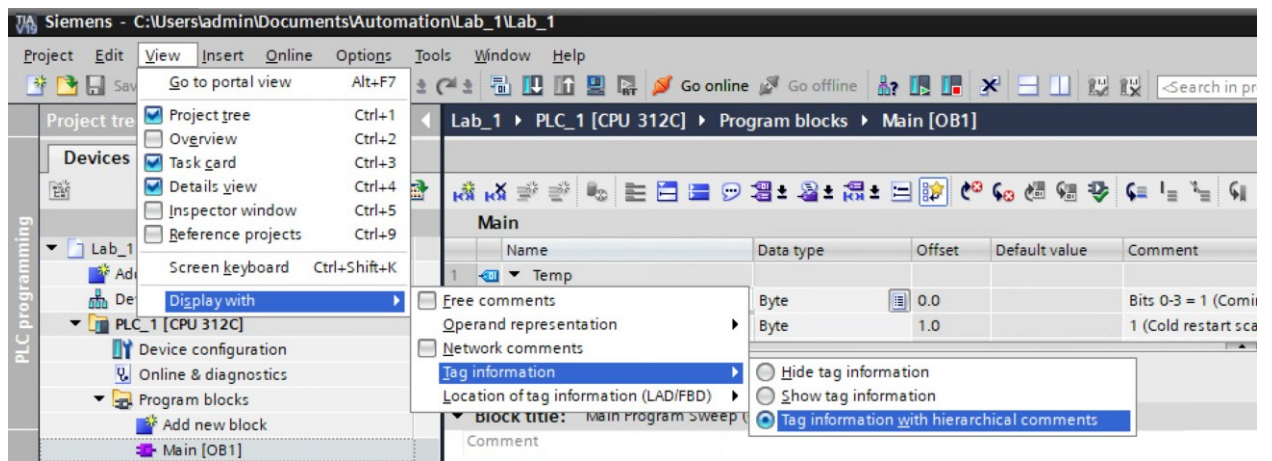


Рисунок 1.18 Ввімкнення відображення підказок для тегів.

Підказки можуть відображатись безпосередньо поряд з тегами або внизу кожної секції, ці режими відображення можна змінити за допомогою меню **View** → **Display with** → **Location of tag information (LAD/FBD)** → **At LAD/FBD address**

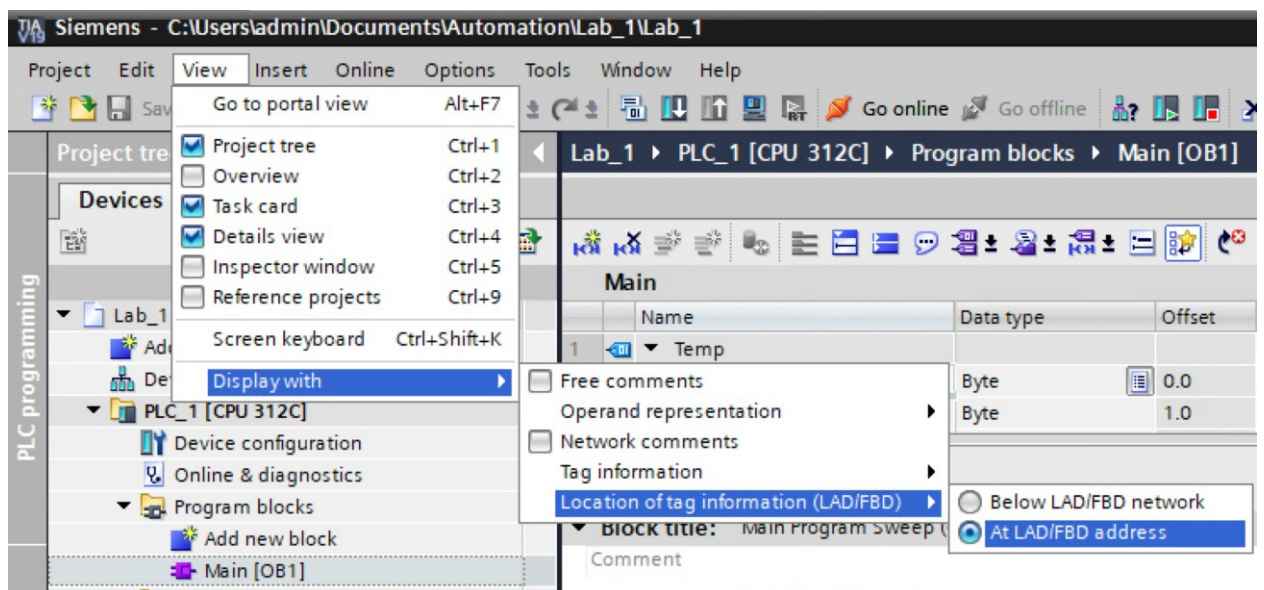


Рисунок 1.19 Зміна режиму відображення підказок для тегів.

За таких налаштувань програма в редакторі буде мати такий вигляд:

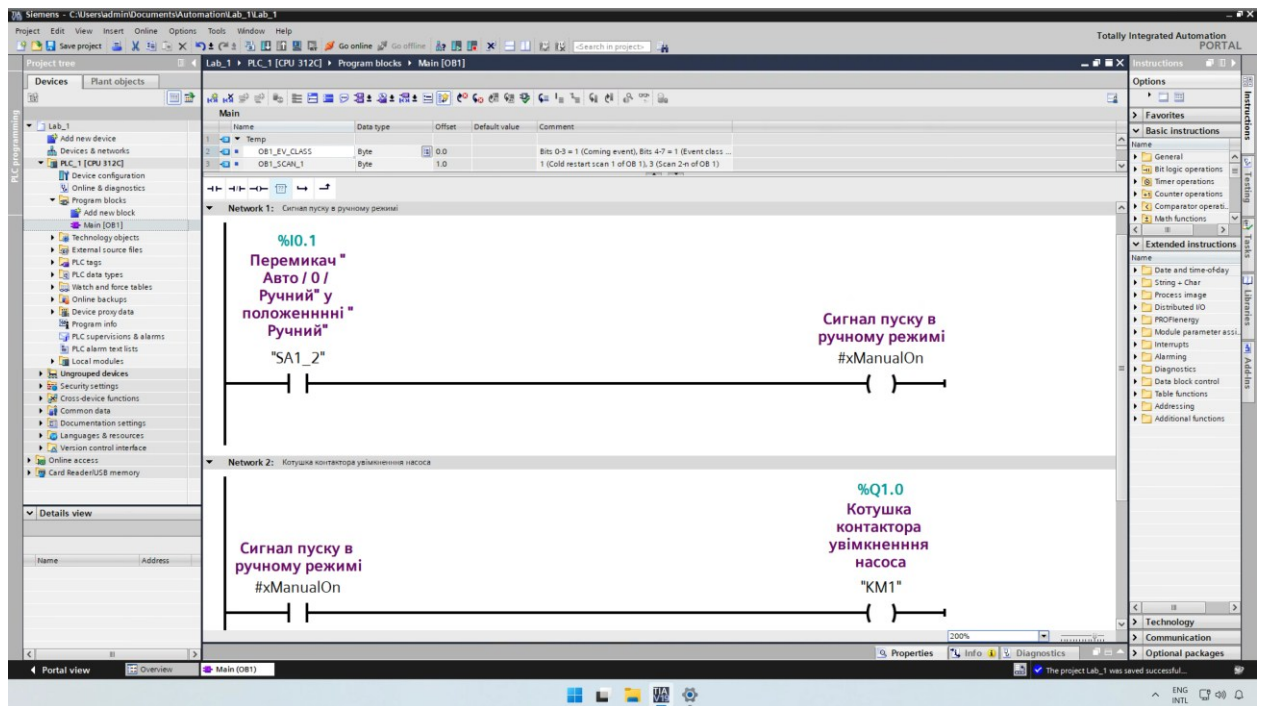


Рисунок 1.20 Програма з ввімкненими підказками біля тегів.

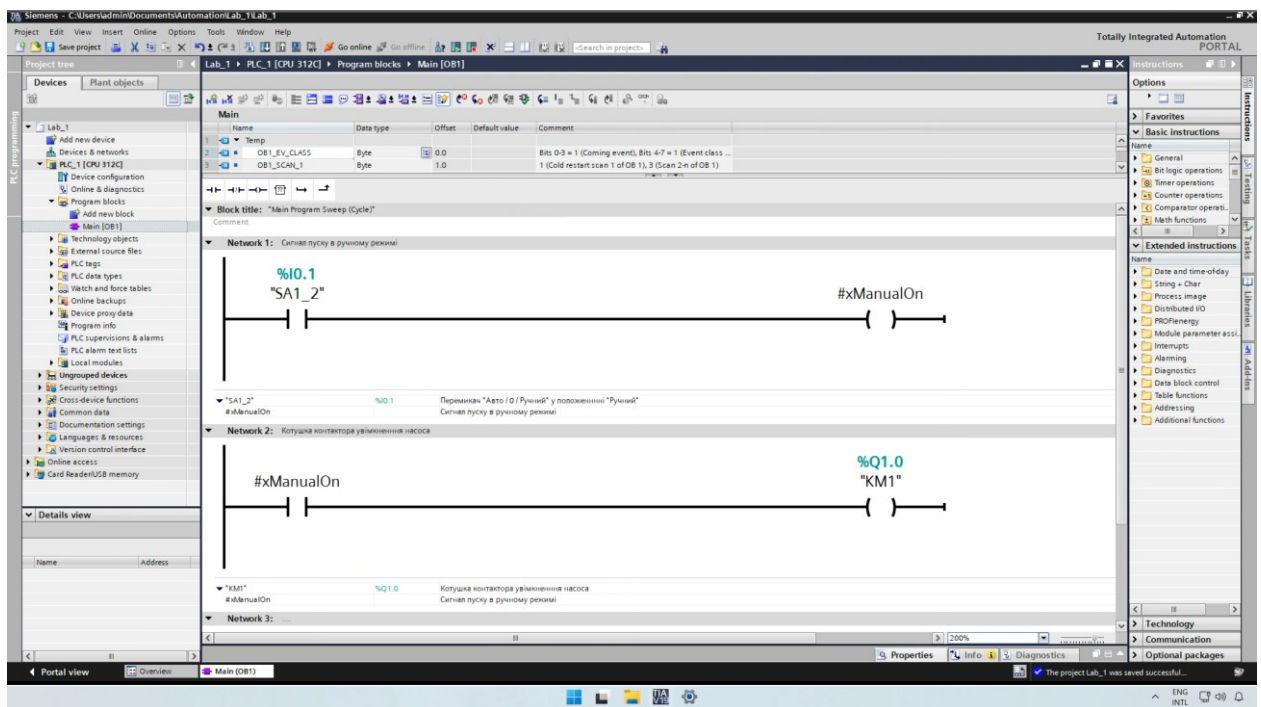


Рисунок 1.21 Програма з підказками у нижніх частинах секцій.

Коментарі можна додавати не тільки до тегів, але й до кожної частини програми. Для ввімкнення коментарів секцій можна скористатись головним меню TIA Portal View → Display with → Network comments:

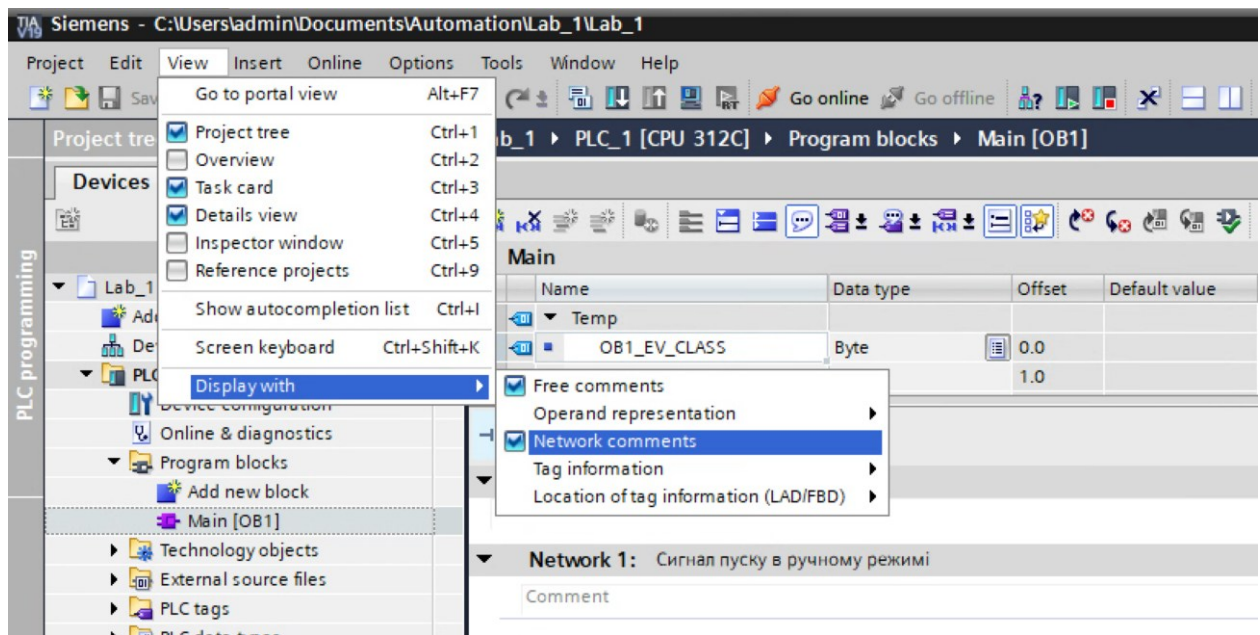


Рисунок 1.22 Ввімкнення відображення підказок до секцій.

- Описати призначення самих секцій за допомогою коментарів. Для цього потрібно написати потрібний текст у верхній частині секції.

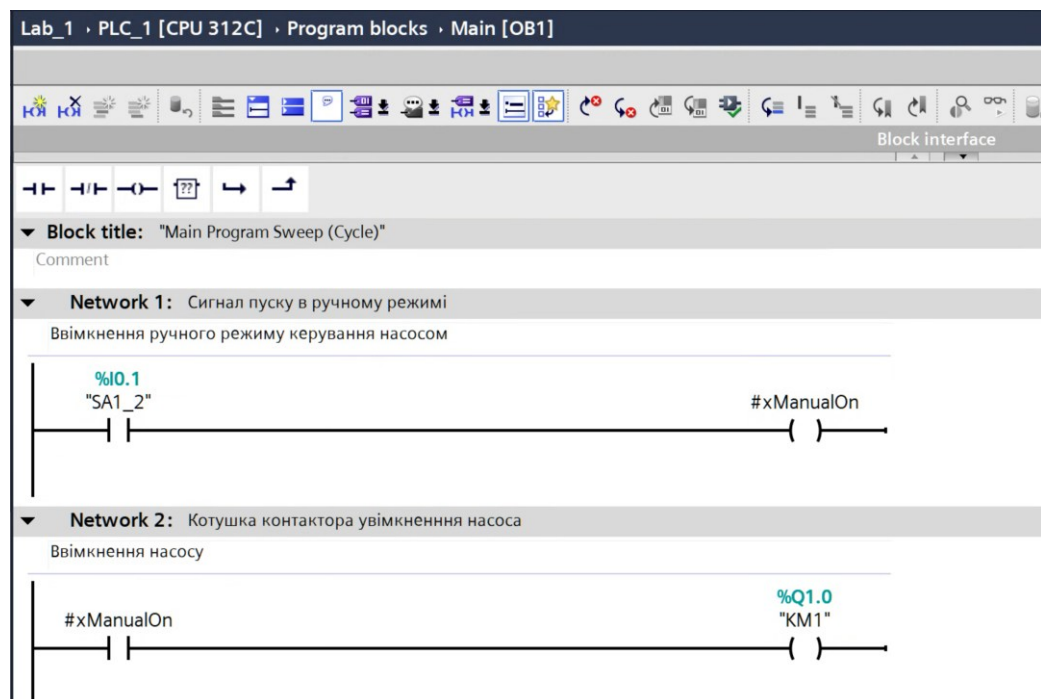


Рисунок 1.23 Пояснення до секцій алгоритму.

Наразі при увімкненні перемикача SA1_2 через проміжне реле xManualOn увімкнеться реле KM1 насоса. Тобто робота насоса залежить тільки від положення перемикача.

6. Удосконалити код, щоб положення перемикача **SA1_2** надавало дозвіл на роботу, а стан насоса визначався двома кнопками без фіксації **SB1** («Пуск») та **SB2** («Стоп»).

Це звична практика фізичного підключення поста керування двигуном. Так як обидві кнопки без фіксації, то для стабільної роботи двигуна потрібно реалізувати схему «самопідхвату». Фізична котушка контактора заживлюється через свій же додатковий контакт, коли отримала живлення через лінію пускової кнопки **SB1** та не заблокована стоповою кнопкою **SB2**.

Розмістити кнопки в секції **Network 1**.

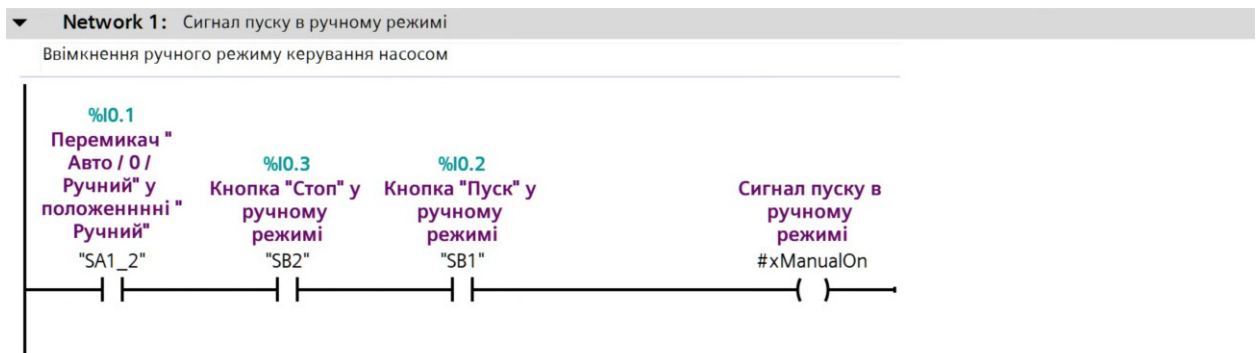


Рисунок 1.24 Додавання кнопок керування насосом в ручному режимі.

7. Кнопка **SB2** «Стоп» повинна бути нормально замкненою, щоб пропустити через себе сигнал, коли не натиснута (оператор не подає команду на зупин). Для цього потрібно інвертувати її контакт у коді. Це можна зробити натисненням за допомогою миші на нормально розімкнений контакт та вибором зі списку, що при цьому з'являється, нормально замкнутого контакту.

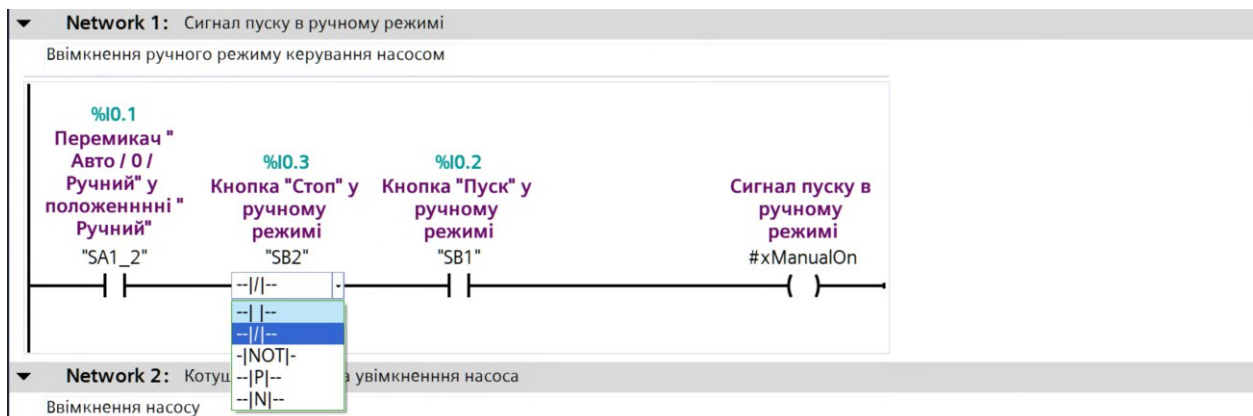


Рисунок 1.25 Зміна типу нормально розімкненого елемента.

8. Для забезпечення самопідхвату командного проміжного реле **xManualOn**, кнопку **SB1** «Пуск» потрібно зашунтувати контактом цього ж реле. Тобто розмістити паралельно контакту кнопки **SB1** контакт **xManualOn**.

Для цього на панелі швидкого доступу потрібно обрати елемент паралельного з'єднання **Open branch** і перетягнути його до контакту **SB1**. Після цього прив'язати контакт до реле **xManualOn**, присвоївши йому це найменування.

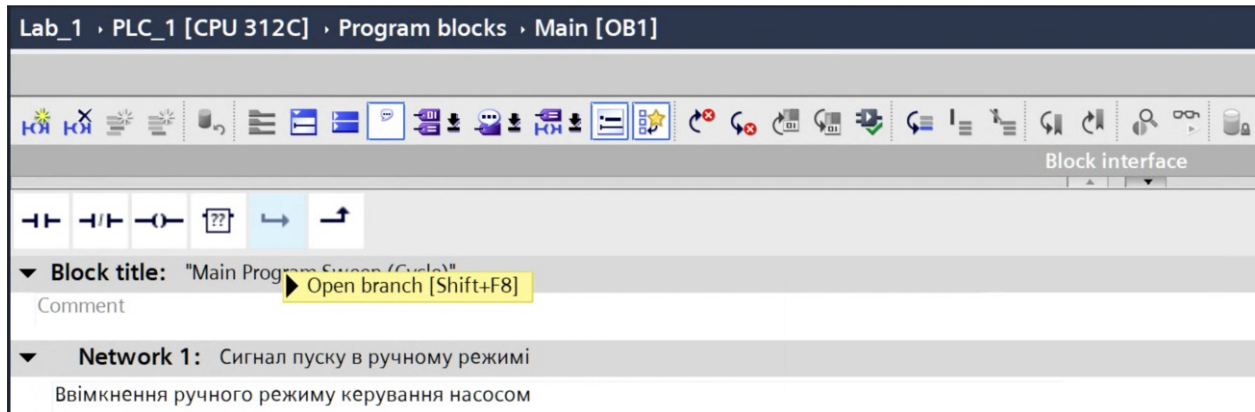


Рисунок 1.26 Додавання паралельної лінії

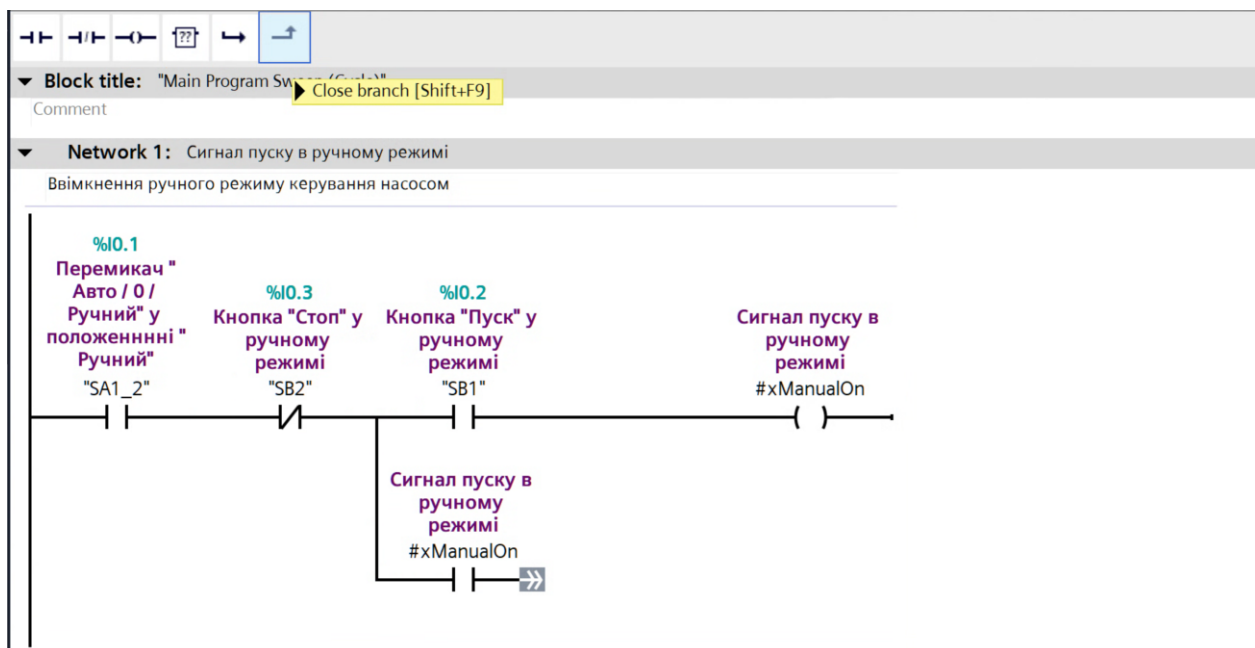


Рисунок 1.27 Закриття паралельної лінії.

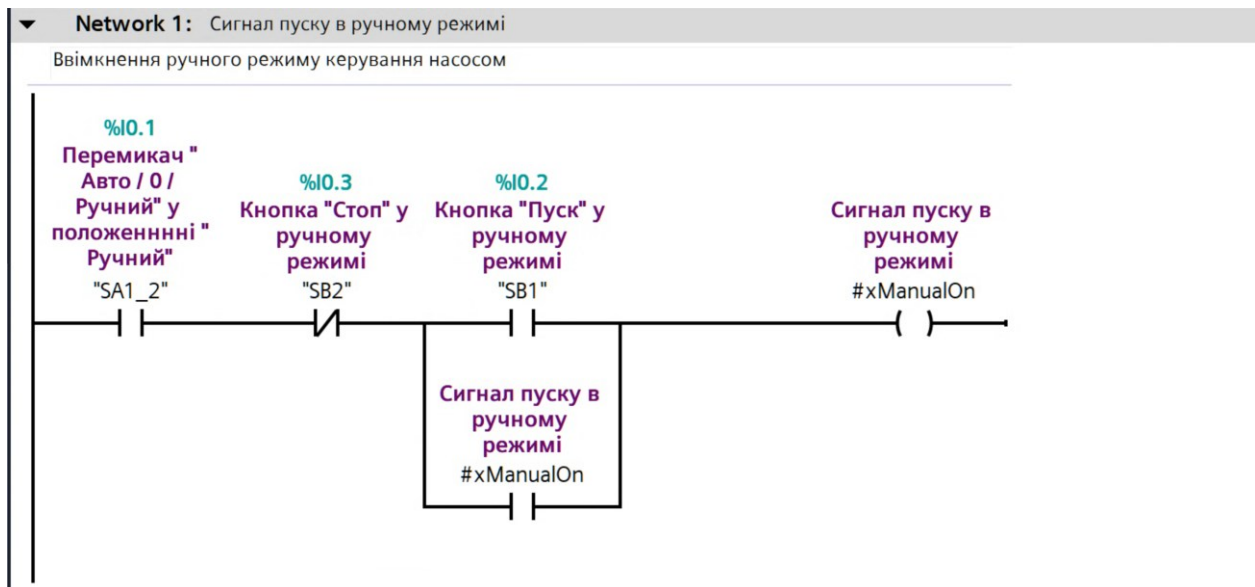


Рисунок 1.28 Схема самопідхвату сигналу пуску в ручному режимі.

Тепер, коли оператор обере ручний режим керування та натисне на кнопку **SB1**, умовний електричний струм потече від лівої шини через **SA1_2**, **SB2** та **SB1** до реле **xManualOn** й активує його. Реле **xManualOn** у свою чергу замкне паралельний до **SB1** контакт **xManualOn**, тим самим шунтуючи кнопку запуску. Реле **xManualOn** вимкнеться, коли оператор натисне на кнопку **SB2** та її нормально замкнений контакт розірве електричне коло.

1.3.3 Програмування захисту та індикації роботи насосу

1. З міркувань безпеки та захисту обладнання команду на увімкнення можна надсилати лише до справного механізму. За наявності критичної аварії, що унеможливує нормальну роботу устаткування, його потрібно вимкнути і заблокувати до усунення причин аварії. Для цього потрібно використовувати додаткові датчики, пов'язані з устаткуванням, та / або алгоритми обробки даних з них.

Насос обладнаний нормально замкненим термореле **TK1**, яке сигналізує про перегрів двигуна. Коли двигун перегрівається – термореле розмикає свій контакт. Таким чином, наявність сигналу з фізичного датчика говорить про нормальну роботу механізму. У разі, коли сигнал зник (термореле розімкнулось), алгоритм має вимкнути насос.

2. Розмістити контакт термореле **TK1** в секції **Network 1** керування насосом.

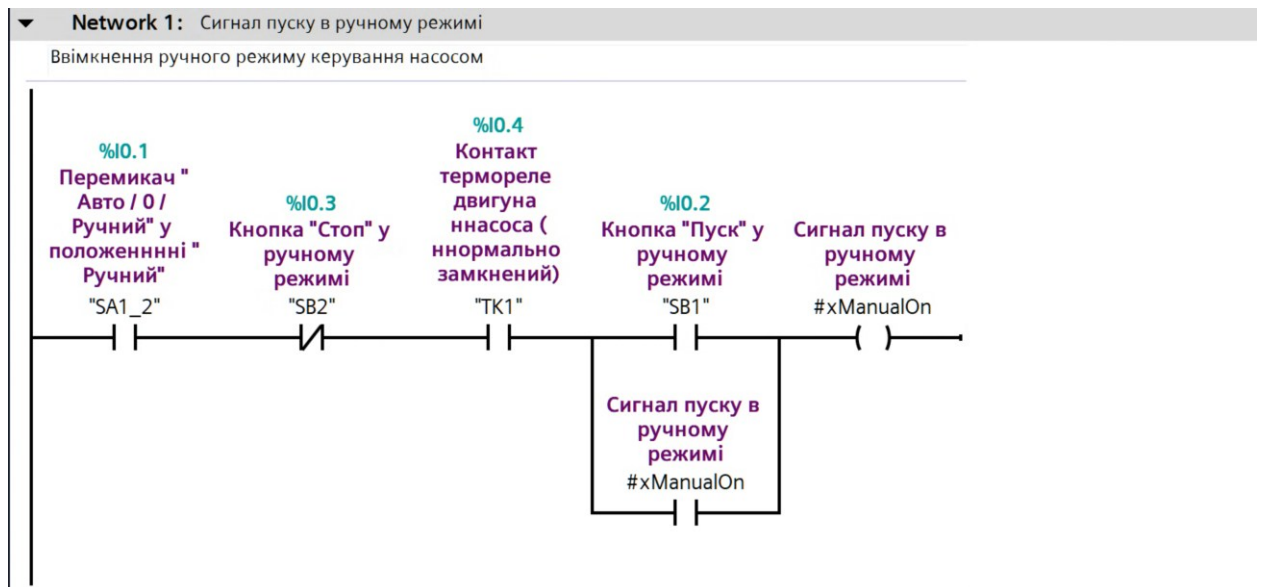


Рисунок 1.29 Додавання захисту двигуна від перегріву.

Таким чином, коли присутній сигнал від термореле **TK1**, його програмний контакт буде замкнено і струм (сигнал) може протікати далі до керуючого реле. Якщо сигнал від **TK1** зникне або оператор натисне на кнопку зупини **SB2** – насос зупиниться.

- Для кращого візуального сприйняття процесу варто намагатися показувати, в якому стані знаходяться механізми. Для цього зручно використовувати світлові індикатори. Паралельно до реле керування насосом в секції **Network 2** встановити котушку реле **HL1**. Тоді ПЛК одночасно з командою насосу буде активувати дискретний вихід, який керує станом фізичної лампочки **HL1**.

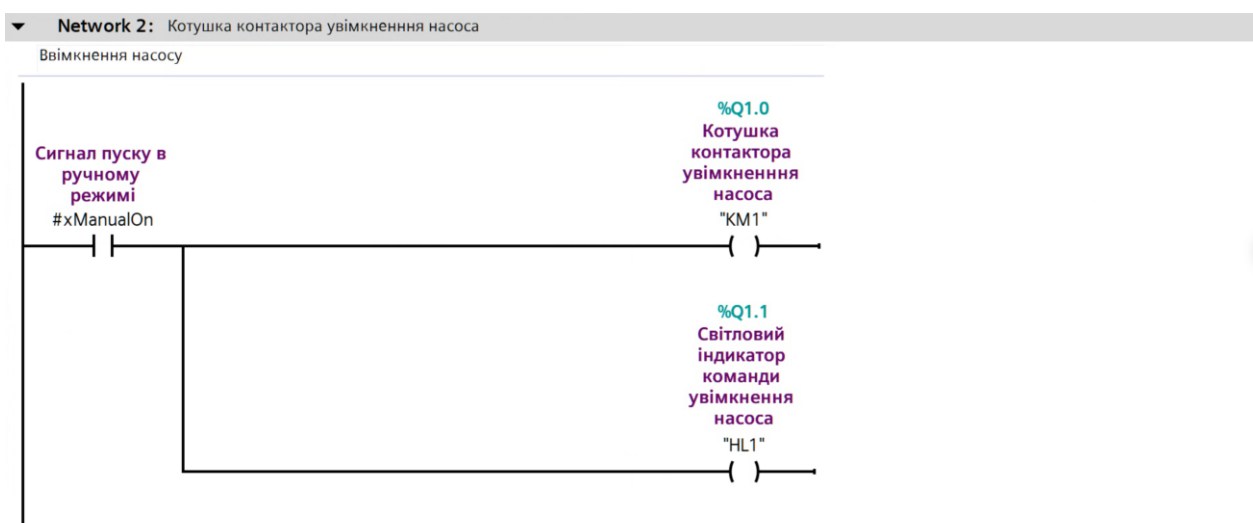


Рисунок 1.30 Додавання керування лампою індикації роботи насосу.

1.3.4 Програмування алгоритму автоматичного режиму

1. Основним режимом роботи обладнання, яке працює під управлінням програмованих логічних контролерів, має бути автоматичний. Закладені алгоритми повинні самі визначати, який стан має бути у механізмів, а оператор лише задає загальний тон роботи – надає дозвіл та визначає потрібний режим функціонування за допомогою параметрів налаштувань.

В автоматичному режимі насос має вмикатися, якщо перевищено верхній рівень рідини в баку (з'являється сигнал від дискретного датчика рівня **LS2** з нормально відкритим контактом) і вимикатися, коли рівень в баку впав нижче нижнього рівня (зникає сигнал від дискретного датчика рівня **LS1** з нормально відкритим контактом). Оператор вмикає автоматичний режим, перевівши перемикач «**Авто/0/Ручн**» в положення «**Авто**» (нормально відкритий контакт перемикача **SA1_1**).

Тобто система має зафіксувати, що перевищено верхній рівень, увімкнути насос, який почне відкачувати рідину, і спостерігати за тим, як рівень рідини спадає. При цьому спочатку вимкнеться датчик верхнього рівня **LS2**, а через певний час – і датчик нижнього рівня **LS1**. Тоді алгоритм має вимкнути насос і очікувати наступного циклу підняття рівня рідини.

Для реалізації алгоритму автоматичної роботи насоса використаємо бістабільні типи котушок реле – (**S**) та (**R**). Котушка (**S**) активується, коли на неї надходить сигнал з електричного кола. Вона залишається в активованому стані навіть тоді, коли вхідний сигнал вже зник. Тобто вона фіксує факт надходження сигналу. Скинути реле можна, подавши сигнал на котушку реле (**R**).

Створити дві секції між схемою ручного керування та управлінням контактором **KM1** насоса. Секція **Network 2** відповідатиме за увімкнення насосу в автоматичному режимі, а секція **Network 3** – за вимкнення. У кожній створеній секції встановити контакт перемикача режиму роботи «Авто» **SA1_1**. Він надаватиме дозвіл на роботу в автоматичному режимі.

Для додавання нової секції після секції **Network 1** можна клацнути правою кнопкою миші на заголовку **Network 1** та обрати в головному меню TIA Portal пункт **Insert** → **Network** або використати «гарячу» комбінацію клавіш **Ctrl+R**.

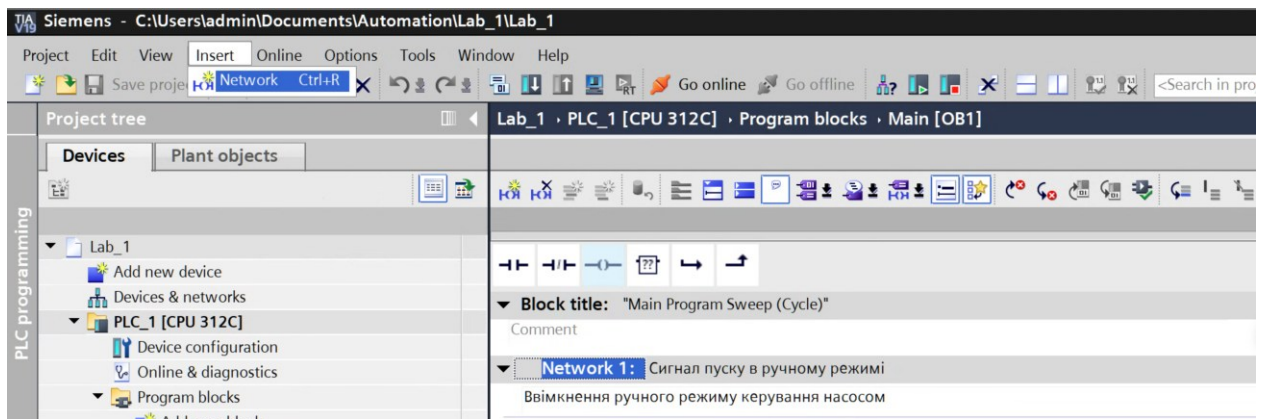


Рисунок 1.31 Додавання нової секції за допомогою головного меню.

Також нову секцію можна додати за допомогою пункту **Insert network** контекстного меню, яке з'являється при правому кліку миші на заголовку секції, після якої потрібно додати нову секцію:

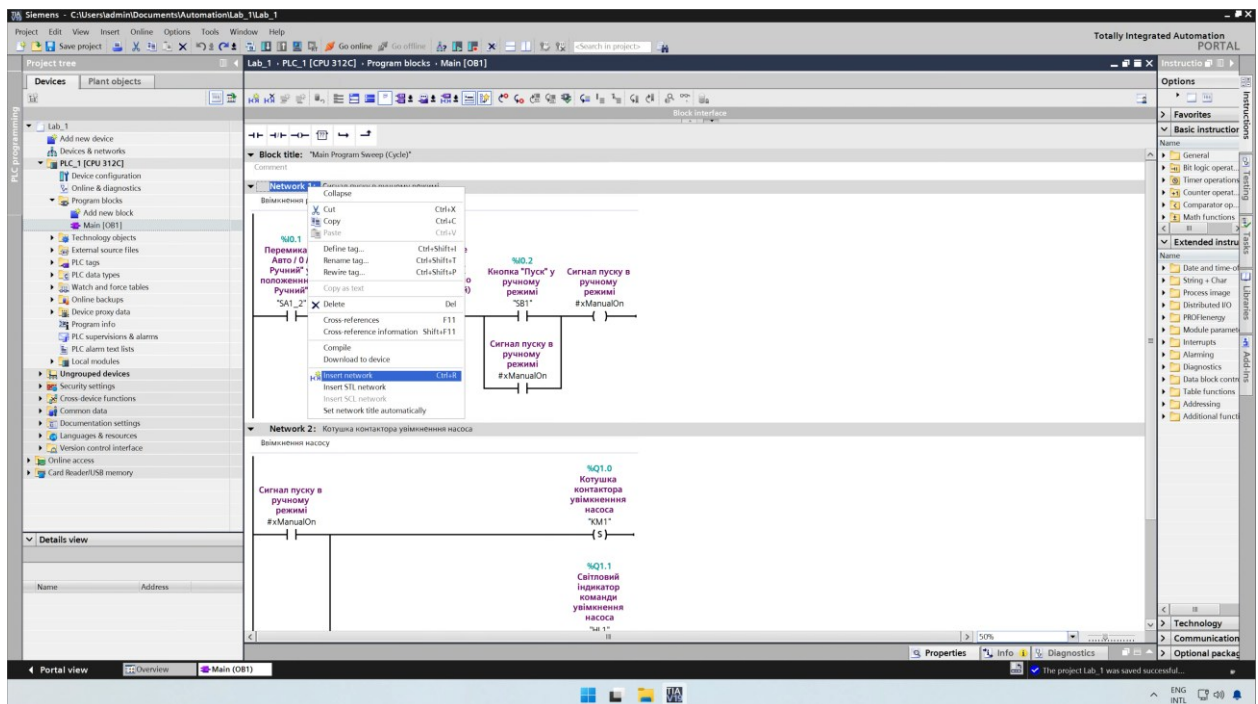


Рисунок 1.32 Додавання нової секції за допомогою контекстного меню.

Після додавання двох нових секцій записуємо у коментарях до них їх призначення: для секції **Network 2** – «Ввімкнення насосу в автоматичному режимі», для секції **Network 3** – «Вимкнення насосу в автоматичному режимі».

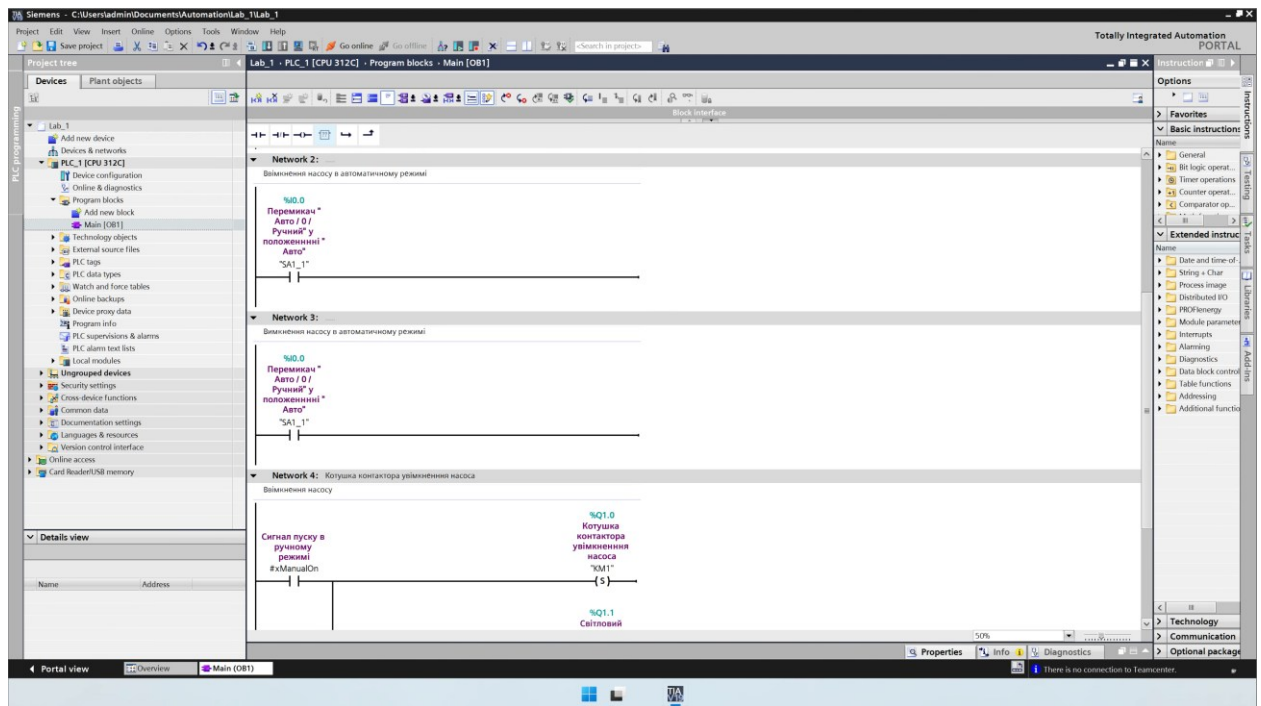


Рисунок 1.33 Додавання секцій керування насосом в автоматичному режимі.

2. У секції **Network 2** скомпонувати схему ввімкнення насоса. Насос вмикається в автоматичному режимі, коли виконуються декілька умов: оператор перевів перемикач «Авто/0/Ручн» в положення «Авто» (є сигнал від контакту SA1_1), датчик рівня LS2 сигналізує про високий рівень в баку (є сигнал), є сигнал від нормально замкнутого фізичного контакту термореле ТК1. При виконанні всіх цих умов має вмикатись котушка встановлення реле керування **xAutoOn**. Котушку встановлення Set output -(S) можна знайти в розділі **Bit logic operations** бібліотеки **Basic instructions**. За замовченням всі бібліотеки знаходяться у правій частині редактора TIA Portal. Змінну **xAutoOn** типу **BOOL** додаємо у розділ локальних змінних організаційного блоку **Main [OB1]** з коментарем «Сигнал запуску насоса в автоматичному режимі роботи».

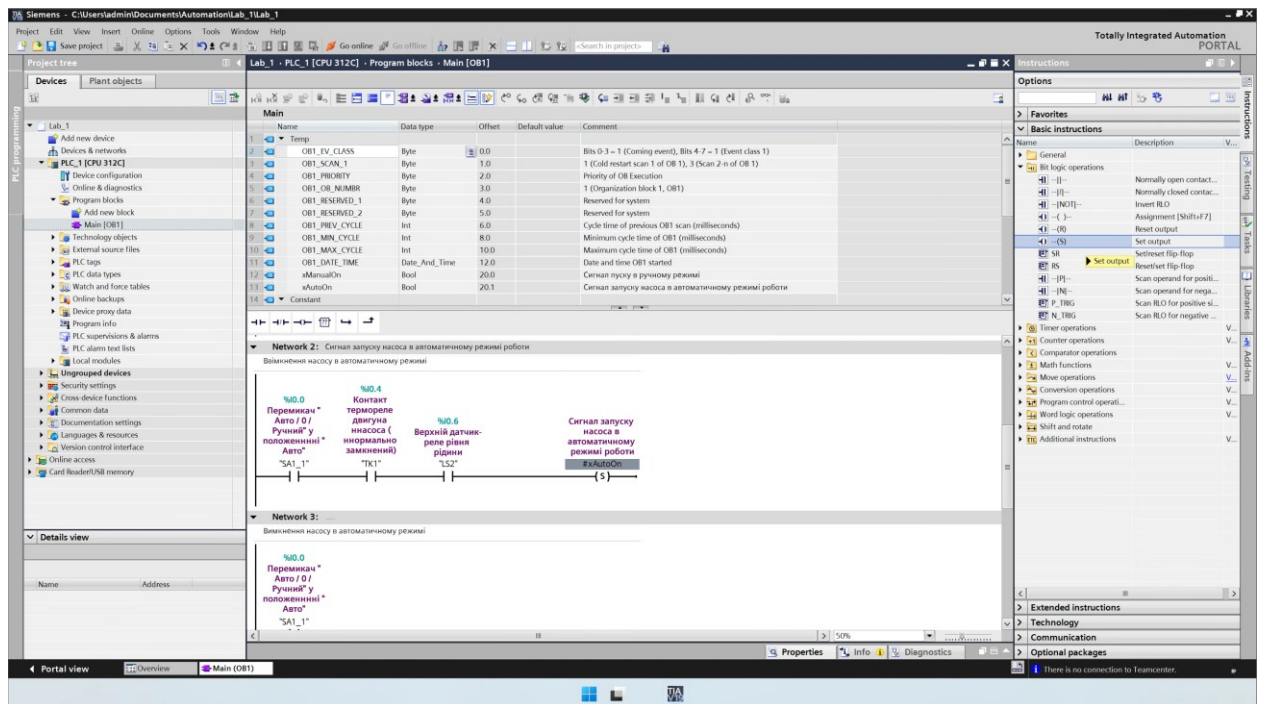


Рисунок 1.34 Алгоритм пуску насоса в автоматичному режимі.

- У секції **Network 3** скласти схему вимкнення насоса. Насос вимикається в автоматичному режимі, якщо виконується хоча б одна з умов: оператор перевів перемикач «Авто/0/Ручн» в положення, відмінне від «Авто» (немає сигналу від контакту SA1_1), датчик рівня LS1 сигналізує про низький рівень в баку (немає сигналу), зник сигнал від нормально замкнутого фізичного контакту термореле ТК1. При виконанні будь-якої з цих умов має скинутися реле керування xAutoOn. Це можна зробити, подавши сигнал на котушку скидання **Reset output -(R)** реле xAutoOn. Елемент **Reset output -(R)** також можна знайти у розділі **Bit logic operations** бібліотеки **Basic instructions**.

Алгоритм вимкнення насосу в автоматичному режимі зображений на рисунку нижче.

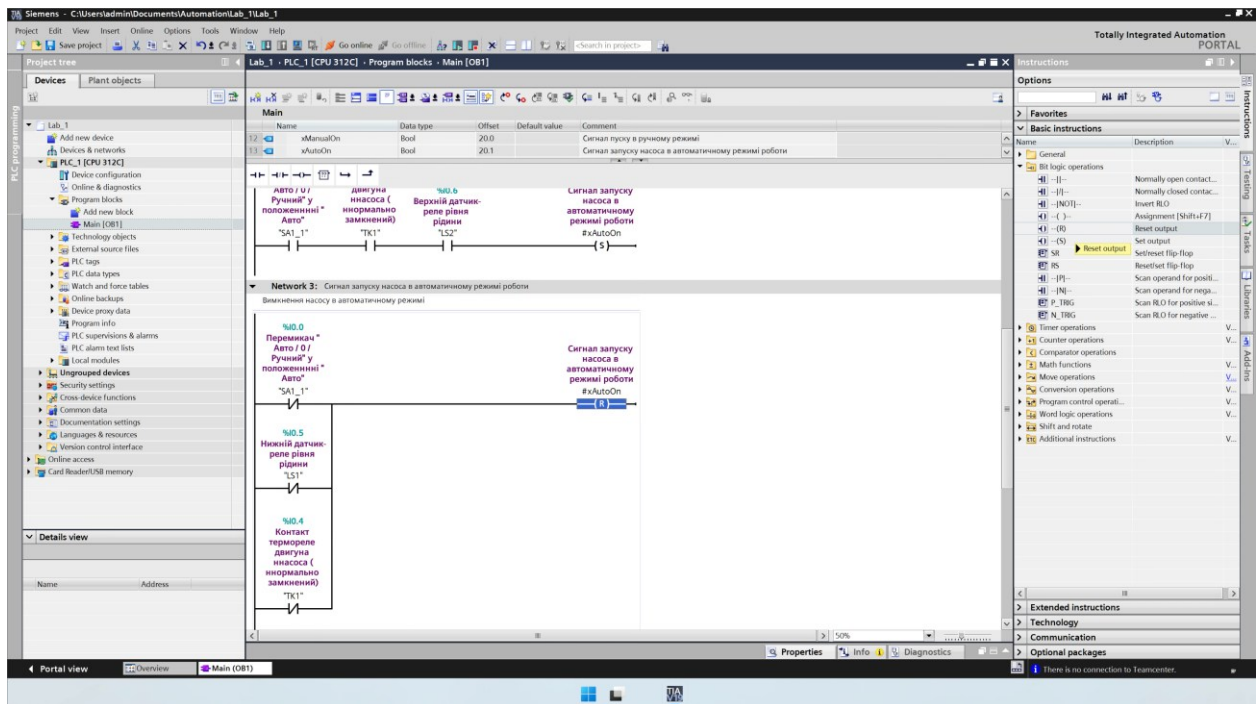


Рисунок 1.35 Алгоритм зупинки насоса в автоматичному режимі.

- Додати можливість надсилання на реле насосу **KM1** команди від автоматичного алгоритму. Оскільки умови ввімкнення у ручному та автоматичному режимах перевіряються у відповідних секціях, то достатньо в схемі секції **Network 4** додати нормально розімкнений контакт увімкнення **xAutoOn**, який подаватиме сигнал пуску насоса в автоматичному режимі.

Після завершення редагування алгоритму необхідно зберегти проєкт. Алгоритм пуску насосу в ручному та автоматичному режимах показаний на рисунку нижче.

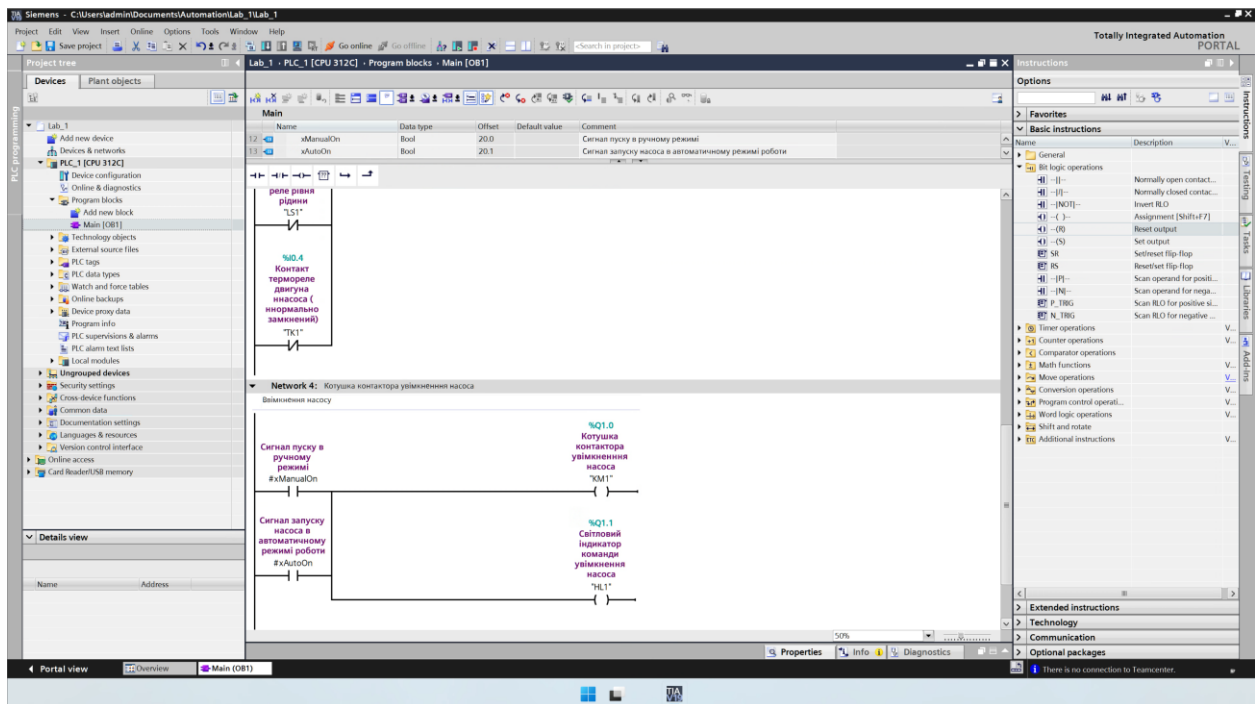


Рисунок 1.36 Алгоритм пуску насоса в ручному та автоматичному режимах.

Загальна рекомендація: вихідні змінні, які є результатом роботи алгоритмів керування, краще встановлювати чи скидати в одному місці наприкінці програмного компоненту. Це дозволяє уникнути ситуацій, коли одна частина алгоритму перезаписує значення змінної, встановлене в іншій частині та спрощує розуміння та процес пошуку помилок коду.

1.3.5 Завантаження та відлагодження коду

1. На перших етапах роботи над проєктом код зручно відлагоджувати в режимі симуляції. У цьому режимі можна працювати навіть за відсутності самого контролера – середовище розробки симулює його поведінку на комп'ютері. Для запуску симулятора контролера Siemens лінійки S7-300 потрібно спочатку обрати контролер, симулятор якого буде запускатись, в дереві проєкту (контролерів у проєкті може бути декілька, тому важливо обирати саме той, алгоритм якого перевіряється). Далі можна використати один зі способів: за допомогою кнопки **Start simulation** панелі інструментів TIA Portal; через головне меню TIA Portal **Online** → **Simulation** → **Start**; через пункт **Start simulation** контекстного меню за правим кліком миші на обраному контролері.

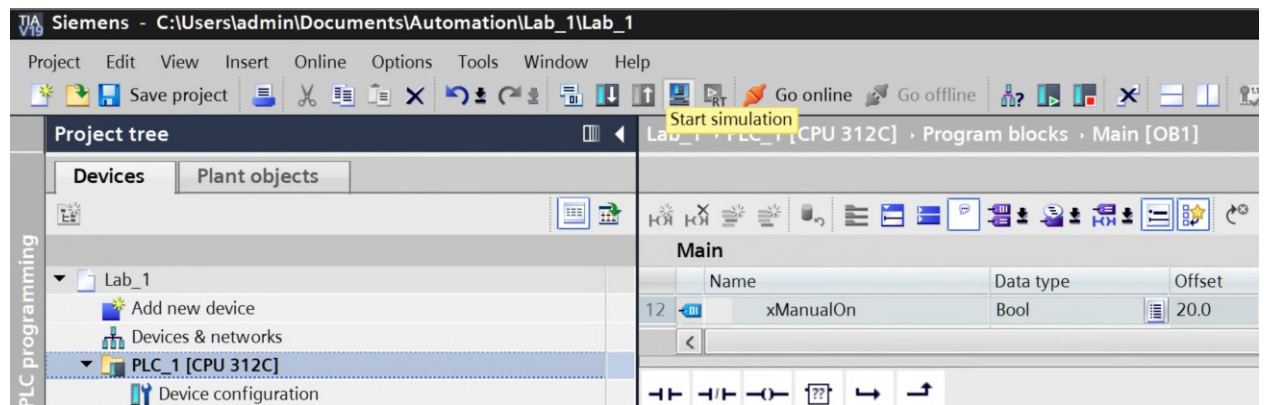


Рисунок 1.37 Запуск симулятора контролера кнопкою на панелі інструментів.

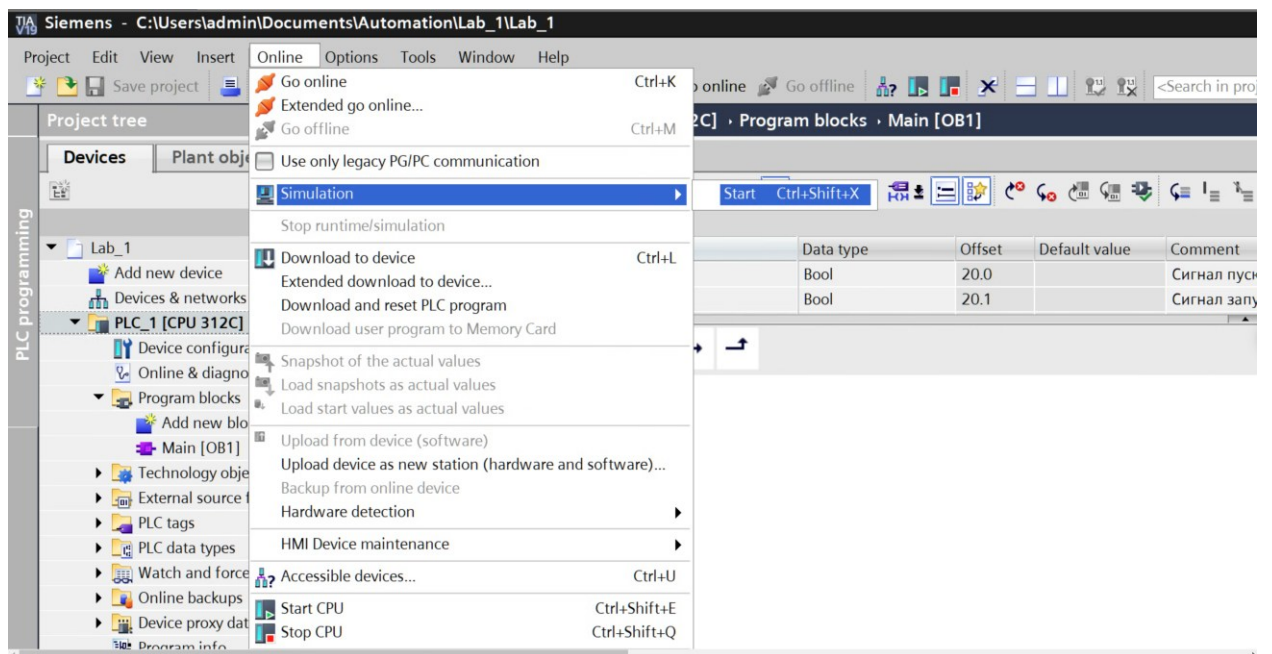


Рисунок 1.38 Запуск симулятора контролера з головного меню TIA Portal.

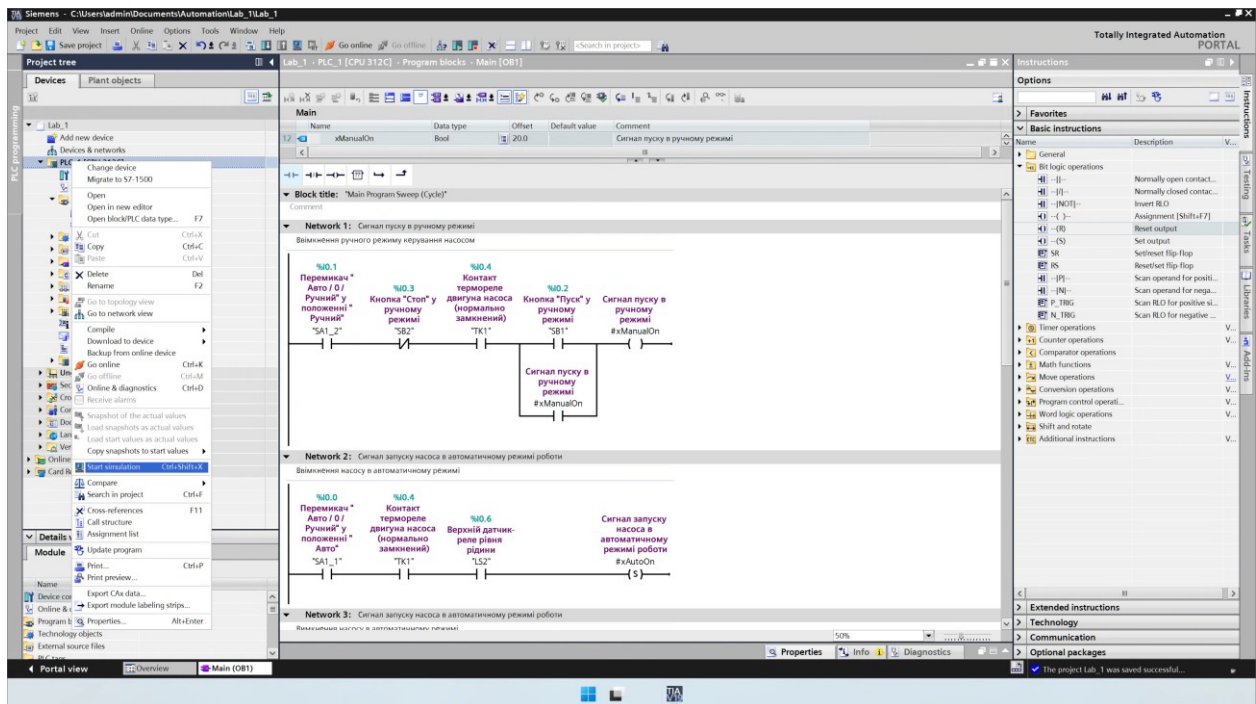


Рисунок 1.39 Запуск симулятора контролера за допомогою контекстного меню.

Після того, як користувач будь-яким з зазначених способів запусить режим симуляції, відкриється вікно з попередженням про те, що використання симулятора блокуватиме всі інші інтерфейси комп'ютера. Це не є помилкою і це вікно можна закрити натисненням на кнопку «OK». Якщо при цьому поставити відмітку напроти пункту «Do not show this message again», тоді це попередження більше не з'явиться.

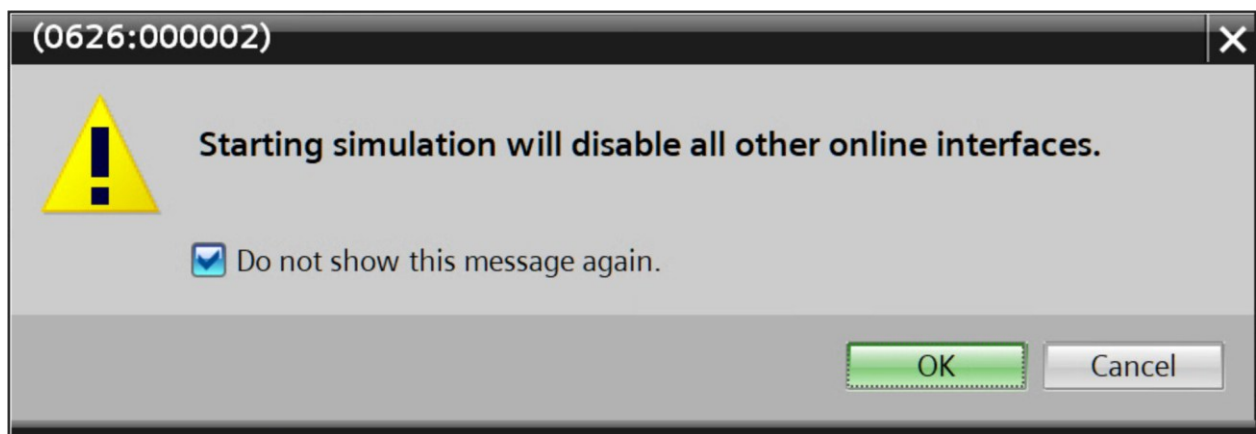


Рисунок 1.40 Попередження про блокування інтерфейсів TIA Portal.

Після закриття вікна попередження має з'явитись вікно програми S7 PLCSIM Simulating Modules (або з'явиться її значок на панелі задач Windows якщо вона запущена у фоновому режимі).

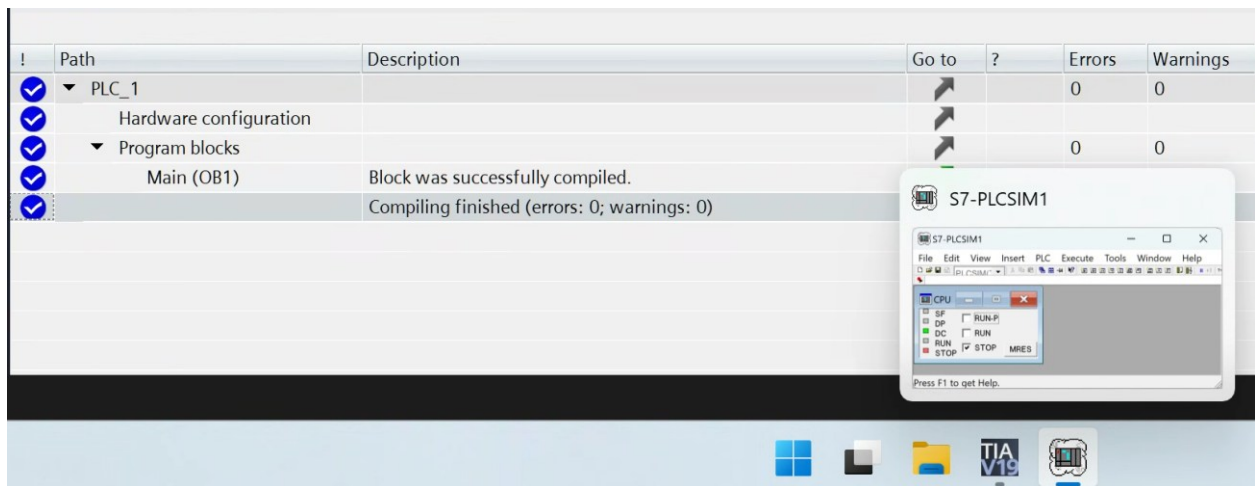


Рисунок 1.41 Значок програми S7-PLCSIM Simulating Modules на панелі задач Windows.

Якщо при створенні програми не було зроблено помилок, які заважають завантаженню програми на контролер, має з'явитись вікно попереднього перегляду завантаження, в якому зазначені всі частини проєкту, що будуть завантажені. Потрібно переконатись, що обрані всі елементи і напроти всіх стоїть синя позначка. У діагностичному вікні компіляції при цьому має з'явитись повідомлення, що компіляція завершена успішно без помилок.

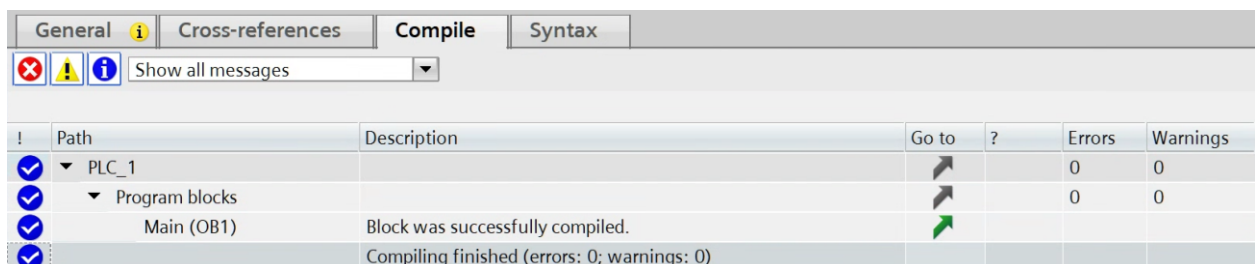


Рисунок 1.42 Діагностичне вікно за успішної компіляції проєкта.

Якщо ж при створенні програми користувач припустився помилок, тоді завантаження програми на контролер буде неможливим до усунення всіх помилок. У вікні завантаження при цьому будуть стояти червоні позначки напроти компонентів, в яких містяться помилки

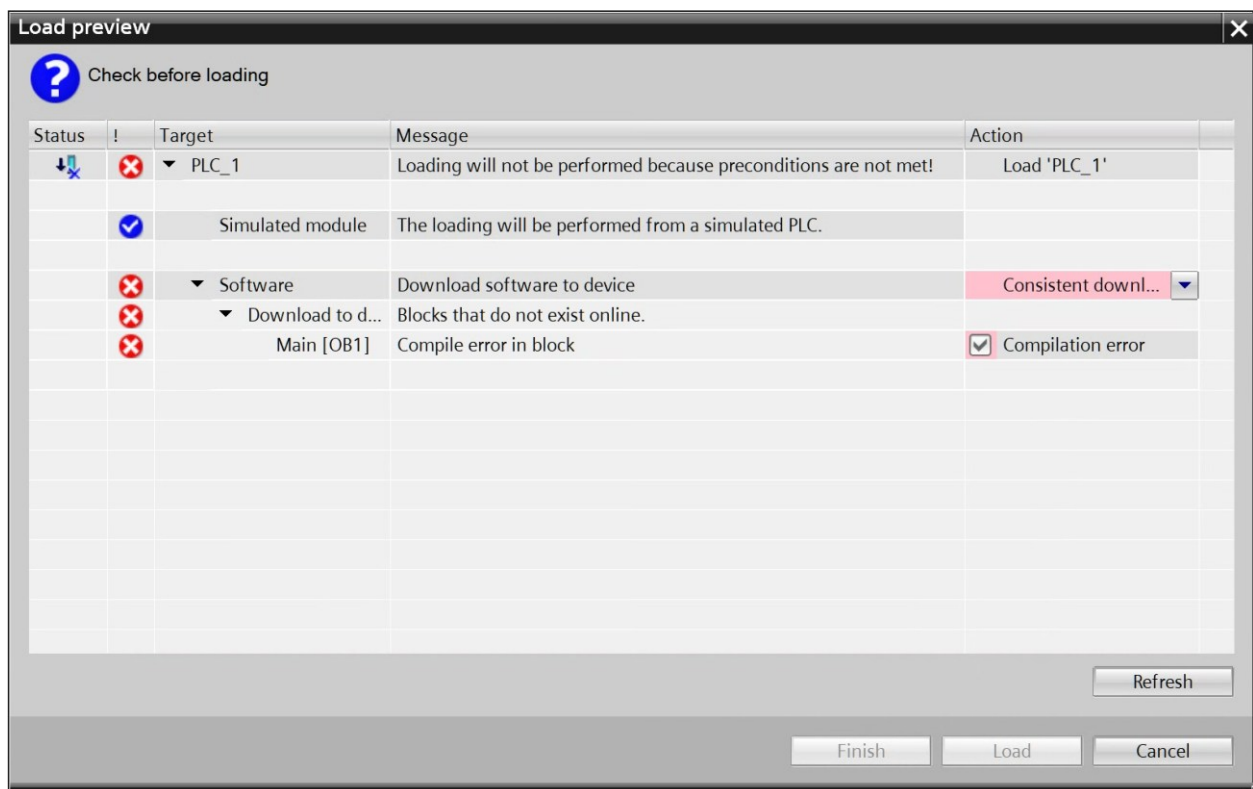


Рисунок 1.43 Вікно завантаження за наявності помилок компіляції.

а у діагностичному вікні компіляції можна переглянути опис всіх помилок та перейти до їх усунення.

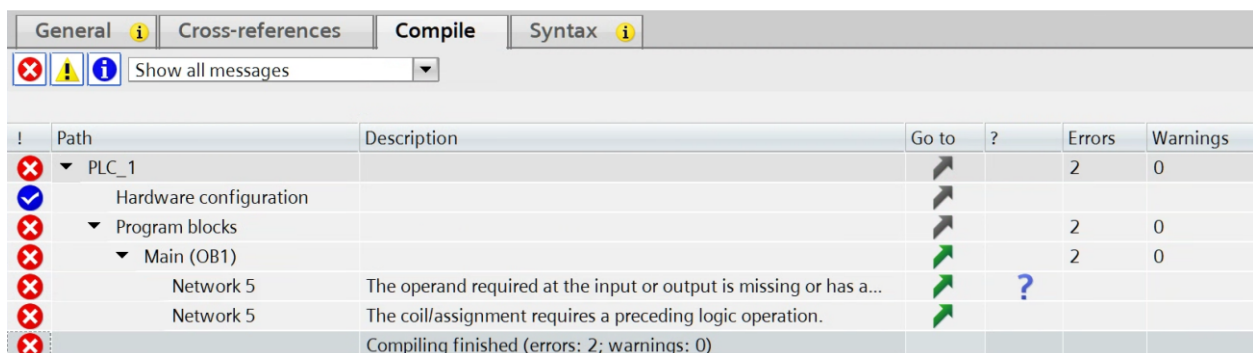


Рисунок 1.44 Діагностичне вікно за наявності помилок компіляції.

Якщо користувач завантажує на контролер всі файли проекту, у полі **Action** вікна завантаження буде позначка **Consistent download**.

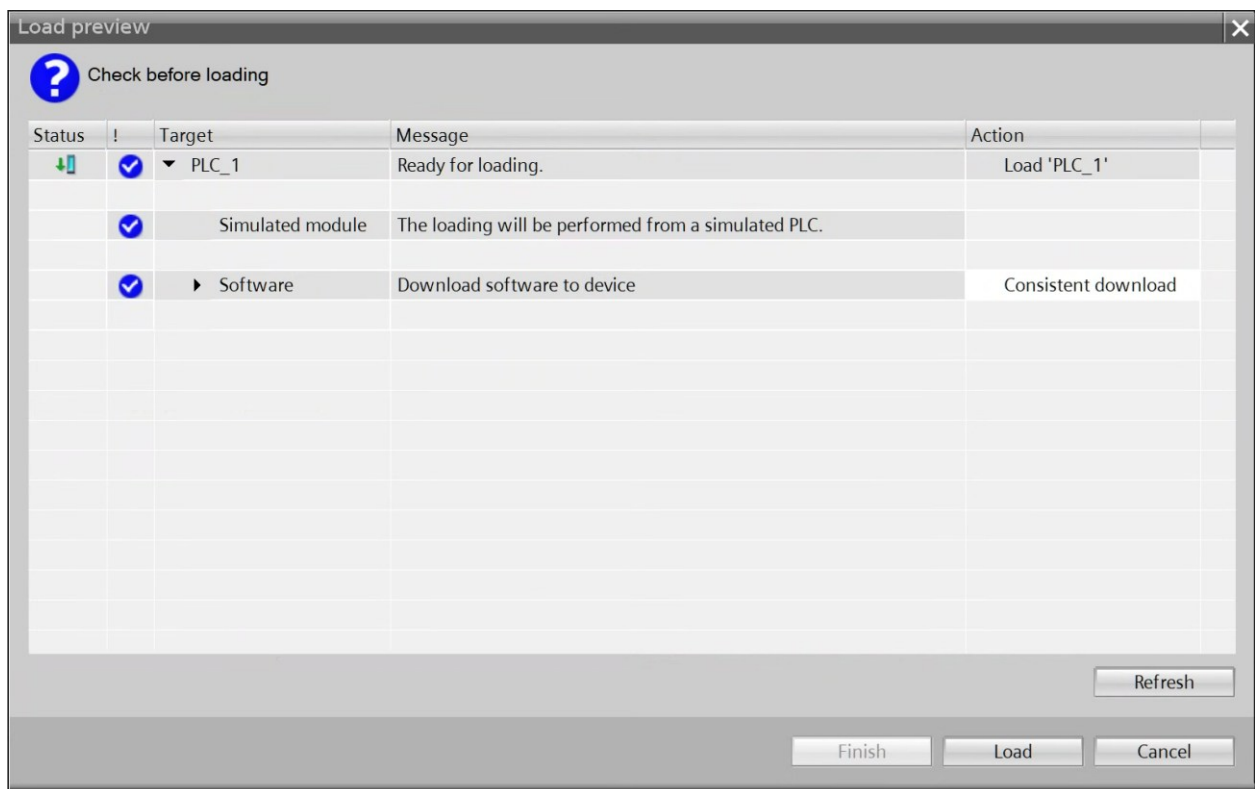


Рисунок 1.45 Вікно завантаження програми без помилок.

Для завантаження програми в симулятор за відсутності помилок компіляції потрібно натиснути кнопку **Load**. За успішного завершення завантаження у діагностичному вікні має з'явитись повідомлення **Loading completed (errors: 0; warnings: 0)**.

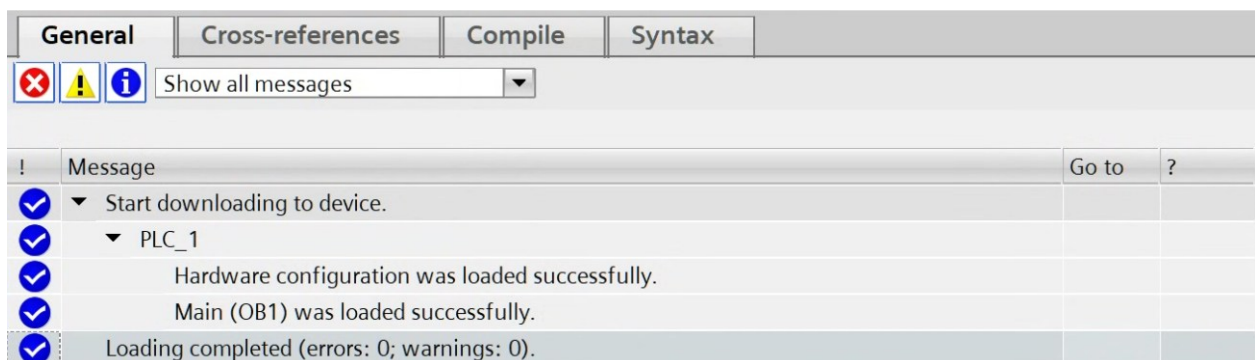


Рисунок 1.46 Діагностичне вікно за успішного завантаження програми.

У вікні симулятора **S7-PLCSIM Simulating Modules** за замовчуванням відкривається лише вікно керування процесорним модулем ПЛК **CPU**.

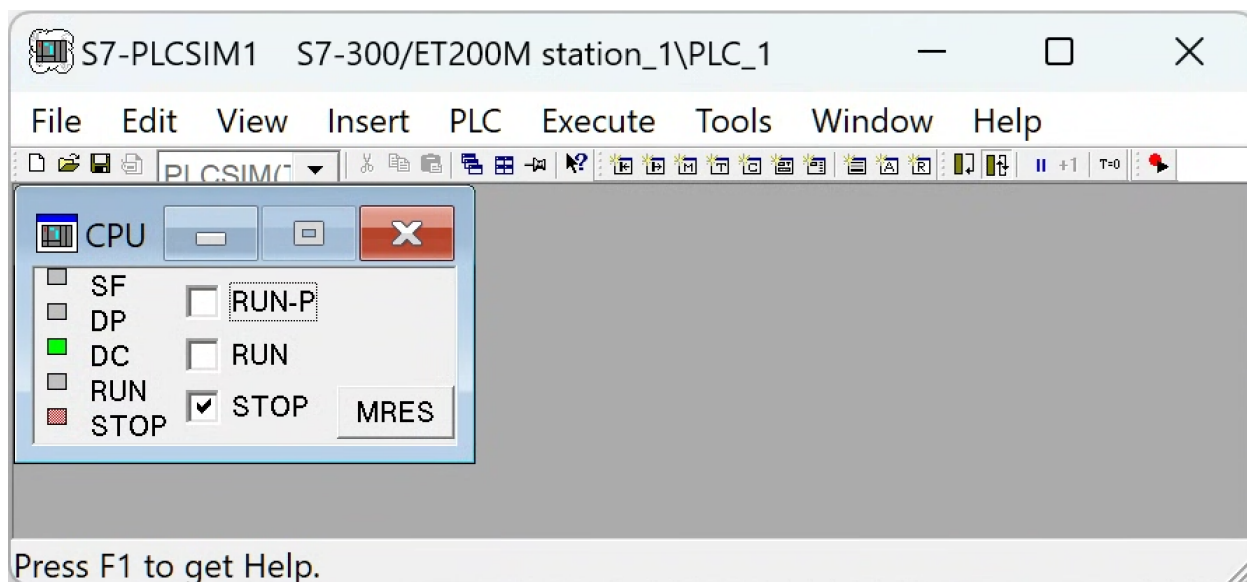


Рисунок 1.47 Вікно симулятора ПЛК S7-PLCSIM Simulating Modules.

При запуску симулятора процесорний модуль перебуває у стані **STOP**, що дозволяє записати на нього програму, але виконання програми у цьому стані буде неможливим. Для запуску програми на виконання процесорний модуль потрібно перевести у стан **RUN** або **RUN-P** за допомогою відповідних відміток у вікні **CPU**. Якщо процесорний модуль знаходиться у стані **RUN**, для запису нової версії програми його потрібно перевести у стан **STOP** або **RUN-P**. У стані **RUN-P** контролер також виконує завантажену програму, але, на відміну від режиму **RUN**, для внесення незначних змін у програму його не потрібно зупиняти, якщо ж зміни будуть суттєвими, TIA Portal запропонує зупинити контролер, а після внесення змін запропонує запустити знову.

Поточний стан контролера можна контролювати за допомогою відміток напроти режимів роботи контролера (**STOP**, **RUN** та **RUN-P**), а також за допомогою індикаторів у вікні симулятора, призначення яких відповідає індикаторам на процесорному модулі контролера:

SF (system fault) – індикатор, який вказує на наявність системної помилки у процесорному модулі контролера внаслідок некоректної роботи програми або помилки у конфігурації контролера.

DP (distributed peripherals) – індикатор, який вказує на обмін даними контролера з іншими пристроями.

DC – індикатор, який вказує на наявність живлення реального контролера або ввімкнений стан контролера при симуляції.

RUN – індикатор, який вказує на те, що контролер виконує завантажену програму.

STOP – індикатор, який вказує на те, що контролер зупинений.

Симулятор **S7-PLCSIM Simulating Modules** у режимі виконання завантаженої програми надає можливість імітувати вхідні сигнали контролера та відслідковувати зміну вихідних сигналів за допомогою вікон **Input Variable** та **Output Variable**. Ці вікна можна відкрити за допомогою відповідних кнопок панелі інструментів симулятора або через пункт меню меню симулятора **Insert**:

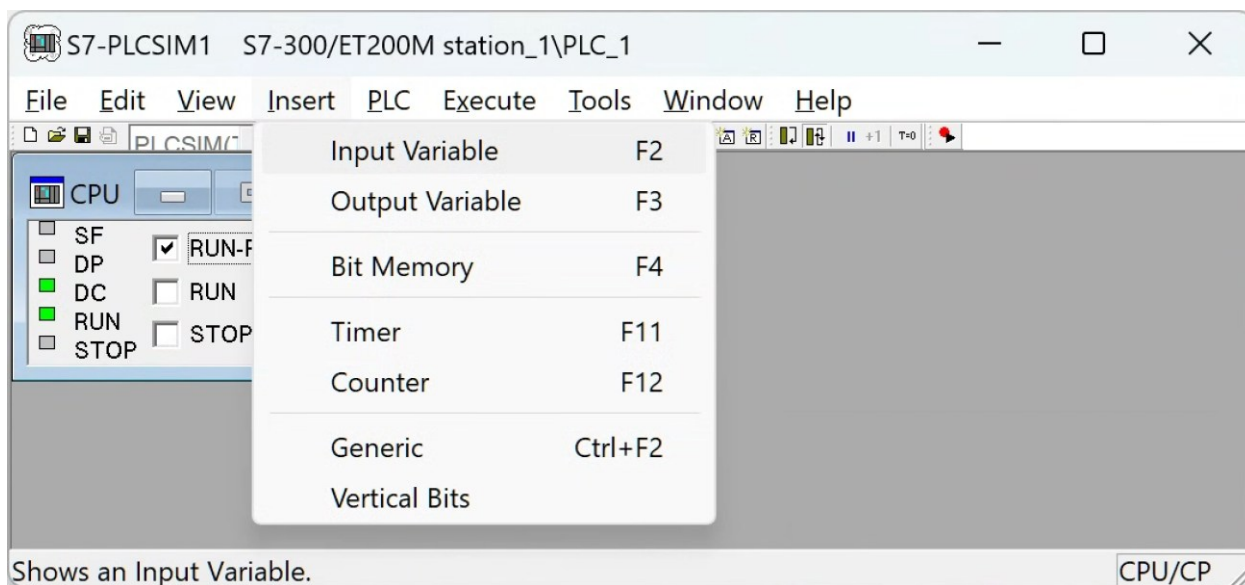


Рисунок 1.48 Додавання вікон зміни вхідних сигналів.

Для зміни сигналів, які відповідають дискретним входам контролера, необхідно ввести адресу вхідної змінної, у якій контролер зберігає їх стан. Відображення адрес вхідних та вихідних сигналів можна ввімкнути їх відображення за допомогою кнопки панелі інструментів вікна редактора програми **Absolute/symbolic operands**:

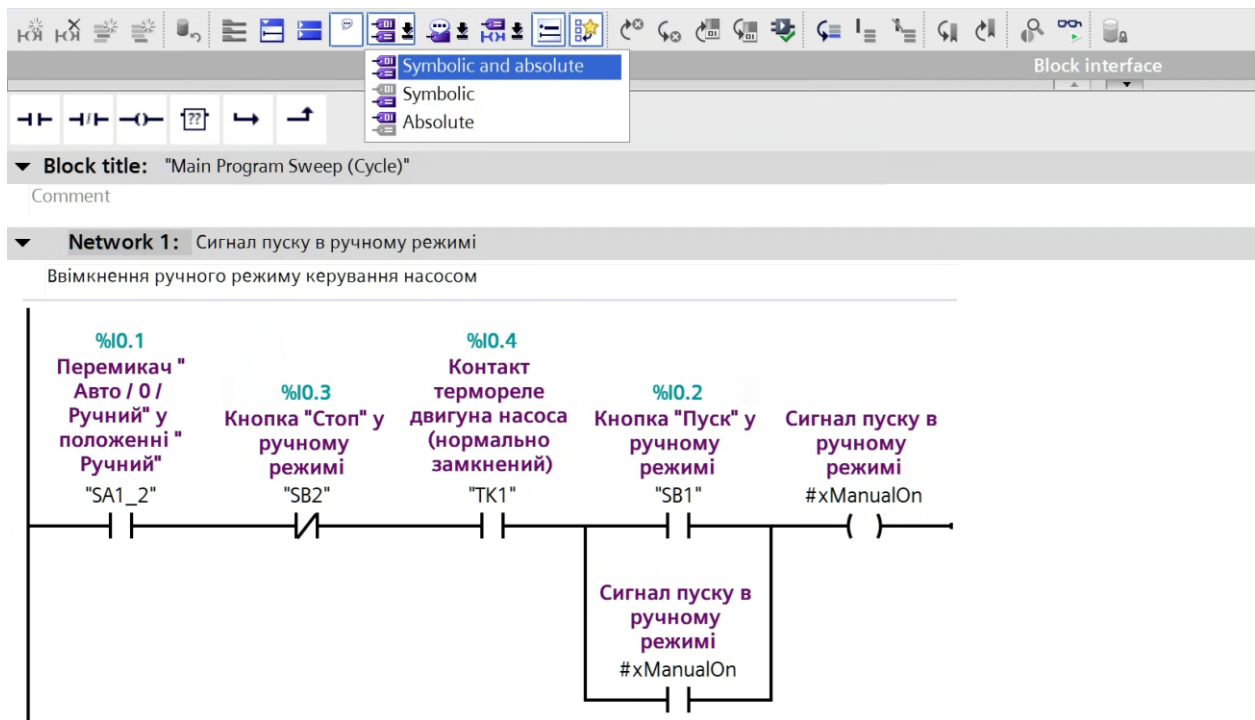


Рисунок 1.49 Відображення адрес вхідних сигналів.

Як видно з рисунку вище, адреси дискретних входів контролера, які відповідають змінним SA1_2, SB1, SB2 та TK1, мають вигляд %I0.1, %I0.2, %I0.3 та %I0.4 відповідно. Це означає, що всі вони знаходяться у нульовому вхідному байті (оскільки адреса починається з %I0), у першому, другому, третьому та четвертому бітах відповідно. Тому для зміни цих бітів у вікні симулятора потрібно відкрити вікно **Insert Input Variable** та ввести адресу нульового байту у форматі **IB0**, де літера **I** вказує, що відображається вхідна (Input) адреса, літера **B** вказує, що відображається стан одного байту (Byte), а цифра **0** відповідає адресі байту, стан якого виводиться у вікні.

Аналогічно потрібно відкрити вікно **Insert Output Variable** та вказати адресу байту, в якому зберігається стан дискретних виходів контролера, що відповідають вихідним сигналам. Для відображення стану першого байту дискретних виходів у вікні **Insert Output Variable** потрібно вказати адресу у форматі **QB1**, де літера **Q** вказує, що відображається вихідна (Quit) адреса, літера **B** вказує, що відображається стан одного байту (Byte), а цифра **1** відповідає адресі байту, стан якого виводиться у вікні:

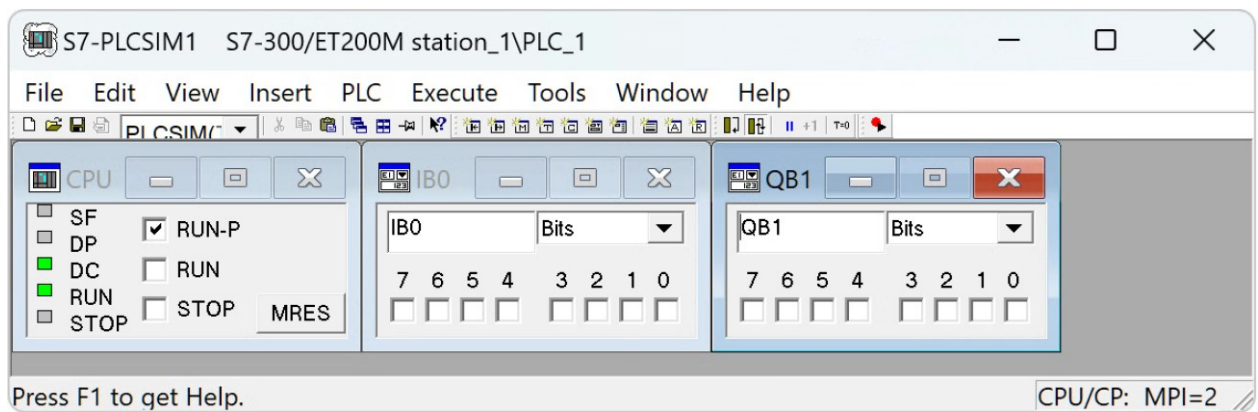


Рисунок 1.50 Відображення вікон зміни сигналів у симуляторі.

Для зміни вхідних сигналів у режимі виконання програми необхідно поставити прапорець, що відповідає потрібному дискретному входу. При ввімкненні алгоритмом дискретного виходу відповідний прапорець з'явиться у вікні **Insert Output Variable**:

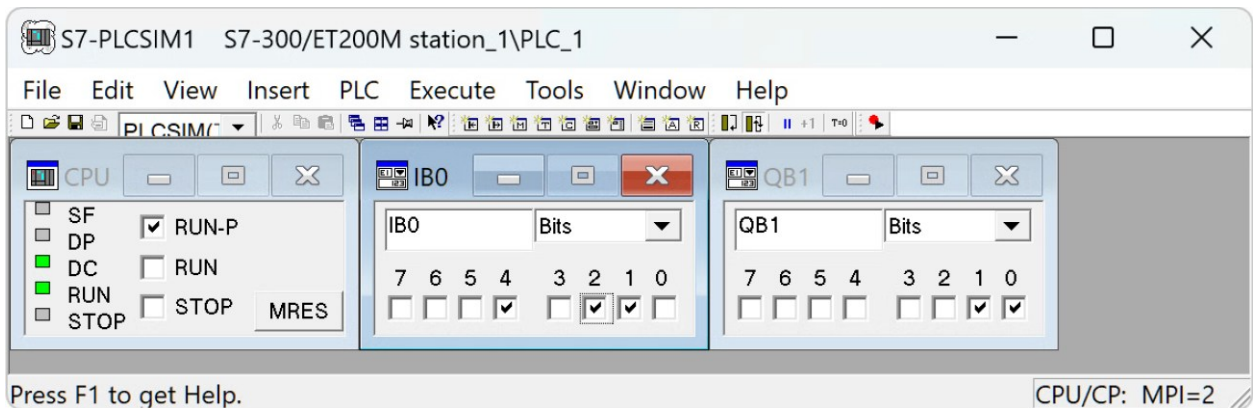


Рисунок 1.51 Перевірка роботи алгоритму у вікні симулятора.

Для нескладних алгоритмів з невеликою кількістю сигналів перевірки правильності програми за допомогою вікна симулятора може бути достатньо. Якщо алгоритм використовує більшу кількість сигналів, які розташовані в різних байтах пам'яті контролера або потрібно з'ясувати причину некоректної роботи коду, використання вікна симулятора може виявитись недостатньо. У такому випадку доцільно буде використати можливість **TIA Portal** перегляду роботи алгоритму в режимі моніторингу. Для ввімкнення цього режиму має бути запущений симулятор контролера (або комп'ютер повинен бути підключений до реального контролера).

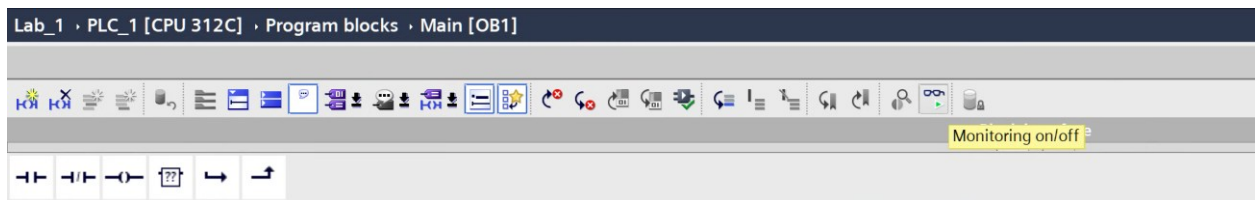


Рисунок 1.52 Ввімкнення режиму моніторингу.

Після успішного ввімкнення режиму моніторингу TIA Portal встановлює зв'язок з симулятором (або реальним контролером) та всі зміни сигналів можна переглядати безпосередньо у вікні редактору програми. Зміна дискретних сигналів відображається зміною кольору та типу сигнальних ліній.

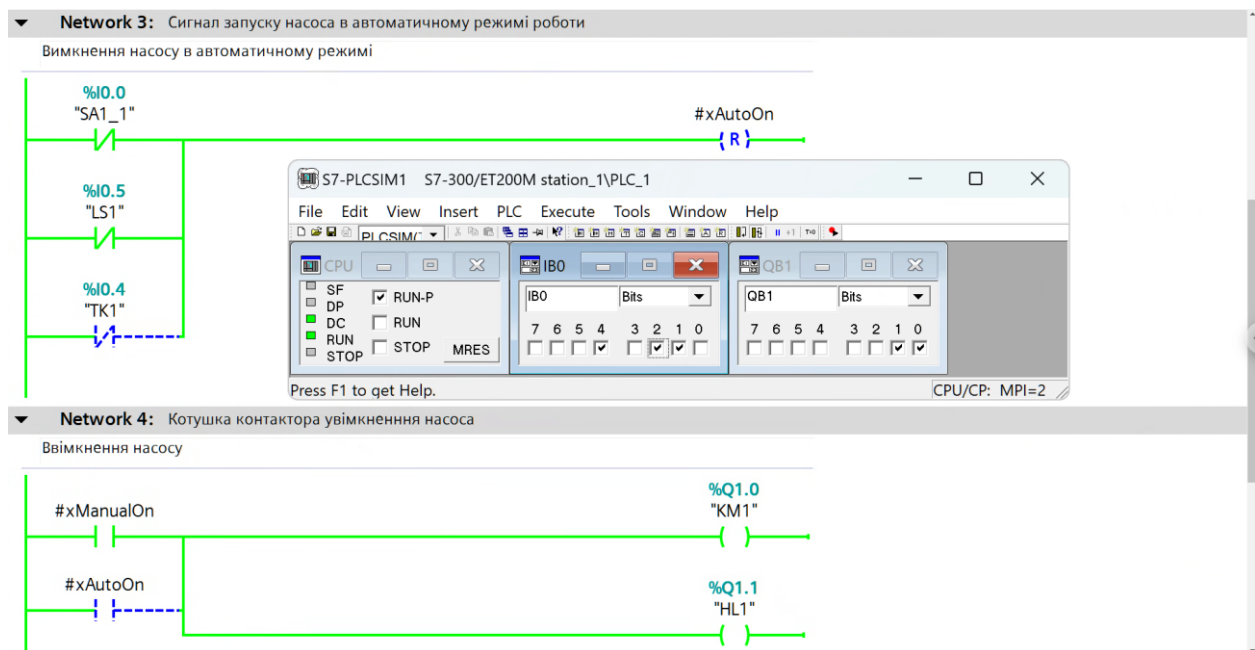


Рисунок 1.53 Перевірка роботи алгоритму в режимі моніторингу

Виконуємо перевірки для розробленого алгоритму за допомогою симулятора **S7-PLCSIM Simulating Modules**.

Спочатку перевіряємо, що алгоритм вірно запускає та зупиняє насос в ручному режимі. Для цього потрібно ввімкнути сигнали, які відповідають змінним **SA1_2**, що є сигналом від положення “Ручний” перемикача “Авто/0/Ручний”, **TK1**, який свідчить про відсутність перегріву насосу та **SB1**, який є сигналом від кнопки запуску насосу в ручному режимі.

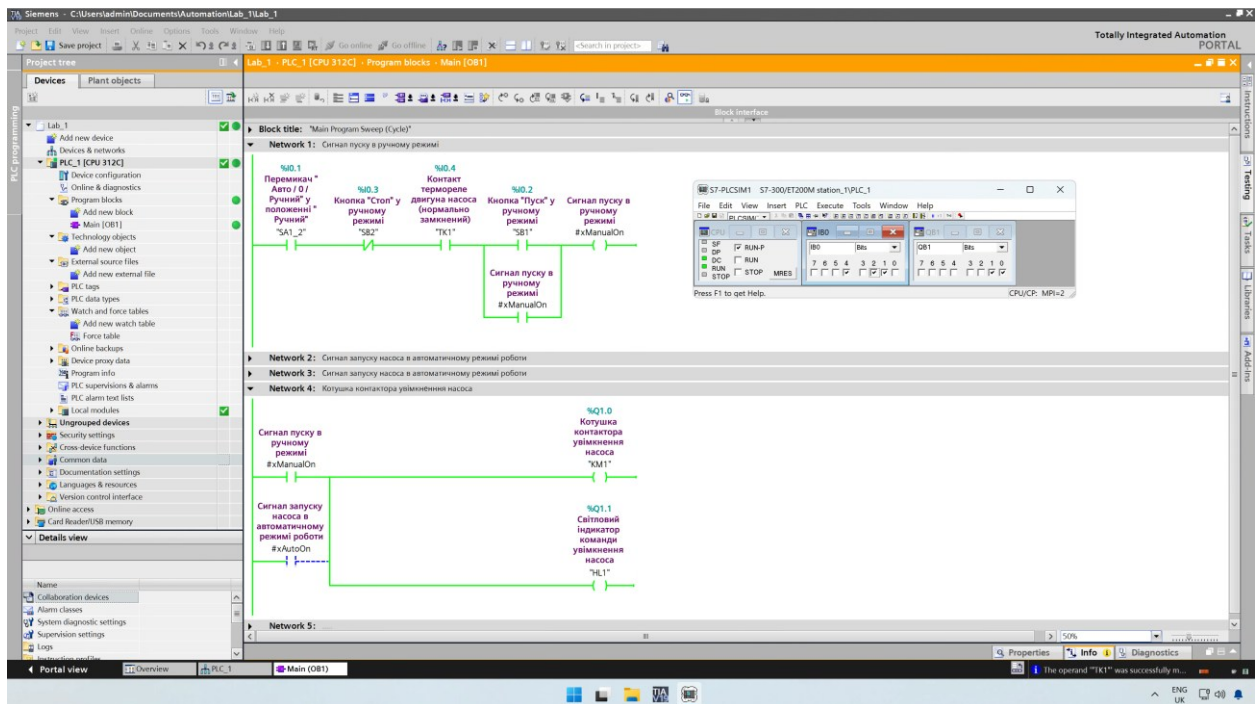


Рисунок 1.54 Перевірка роботи в ручному режимі.

Якщо при написанні алгоритму не було помилок, повинні ввімкнутись **КМ1** та **НЛ1**, які відповідають сигналам керування ввімкнення насосу та світловому індикатору ввімкнення насосу. Це ми можемо перевірити зміною кольору всього ланцюга керування у **Network 1**, який має ввімкнути змінну **xManualOn**, у **Network 4** ввімкнення змінної **xManualOn** має наслідком ввімкнення сигналів на котушку контактора увімкнення насоса **КМ1** та світлового індикатора команди увімкнення насоса **НЛ1**. У вікні симулятора при цьому повинні ввімкнутись біти (з'являться прапорці біля відповідних адрес), які відповідають цим сигналам.

У ручному режимі відсутність сигналу **ТК1** має блокувати подачу сигналів керування, тож вимкненням відповідного біта у вікні симулятора перевіряємо спрацювання теплового захисту насосу.

Коректна робота алгоритму захисту насосу від перегріву повинна вимикати сигнали, які йдуть на керування насосом та світловим індикатором увімкнення насосу. На рисунку нижче бачимо, що у вікні симулятора зникли прапорці виходів, а у вікні редактора сигнальні лінії змінились з суцільних зелених на переривчасті сині, які відповідають відсутності сигналу.

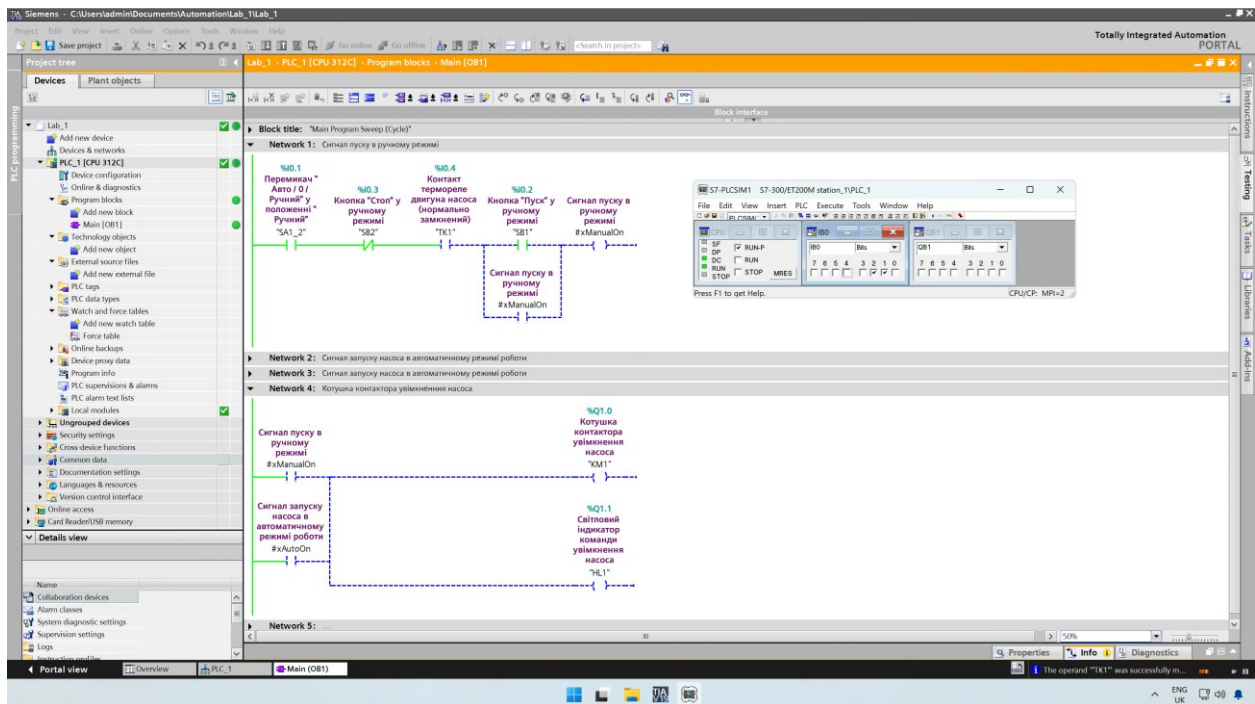


Рисунок 1.55 Перевірка роботи алгоритму захисту від перегріву.

Наступним кроком потрібно перевірити роботу алгоритму в автоматичному режимі. У цьому режимі сигнал керування насосом повинен вмикатись за умови ввімкнення сигналів SA1_1, що відповідає положенню “Авто” перемикача “Авто/0/Ручний”, ТК1, який свідчить про відсутність перегріву насоса та LS2, який є сигналом від верхнього датчика-реле рівня. Ввімкнення цих сигналів має подавати сигнал запуску насоса в автоматичному режимі роботи **xAutoOn** у **Network 2**. Оскільки ввімкнення **xAutoOn** відбувається з фіксацією, для його скидання використовується **Network 3**, в якому описані такі умови скидання цього сигналу: вимкнення автоматичного режиму SA1_1, зникання сигналу від нижнього датчику-реле рівня LS1, що має свідчити про повну викачку резервуару і запобігає сухому ходу насоса відкачки та зникнення сигналу від контакту термореле двигуна насоса ТК1, що запобігає перегріву насоса.

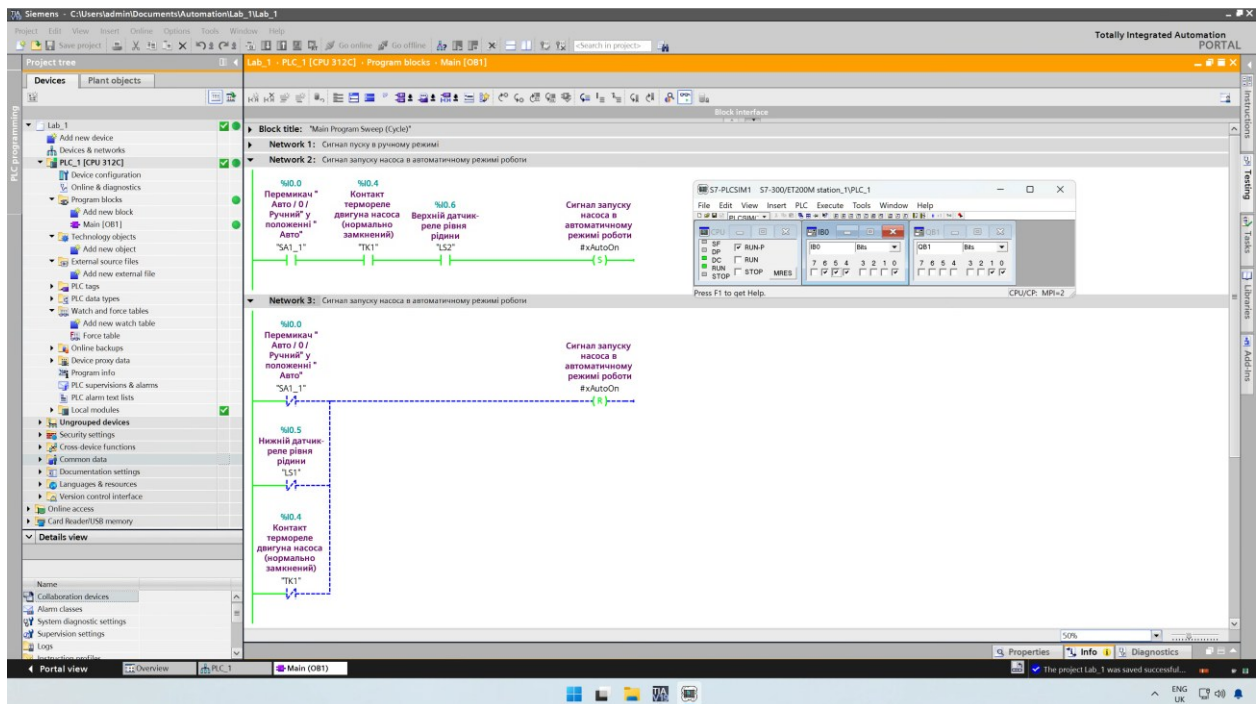


Рисунок 1.56 Перевірка роботи в автоматичному режимі.

Можемо переконатись, що зникнення в процесі роботи насоса в автоматичному режимі сигналу від верхнього датчика-реле рівня рідини не зупиняє насос:

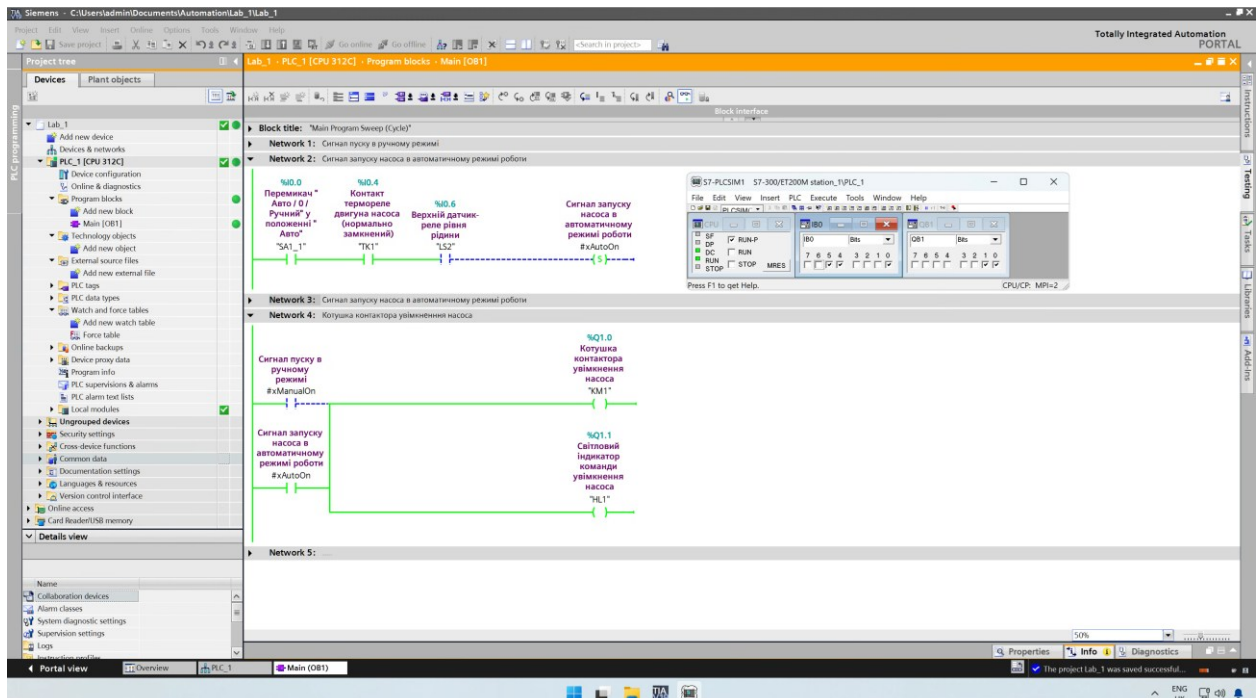


Рисунок 1.57 Перевірка утримання сигналу запуску насоса.

Перевіряємо алгоритм захисту від перегріву двигуна вимкненням сигналу від контакту термореле двигуна насоса:

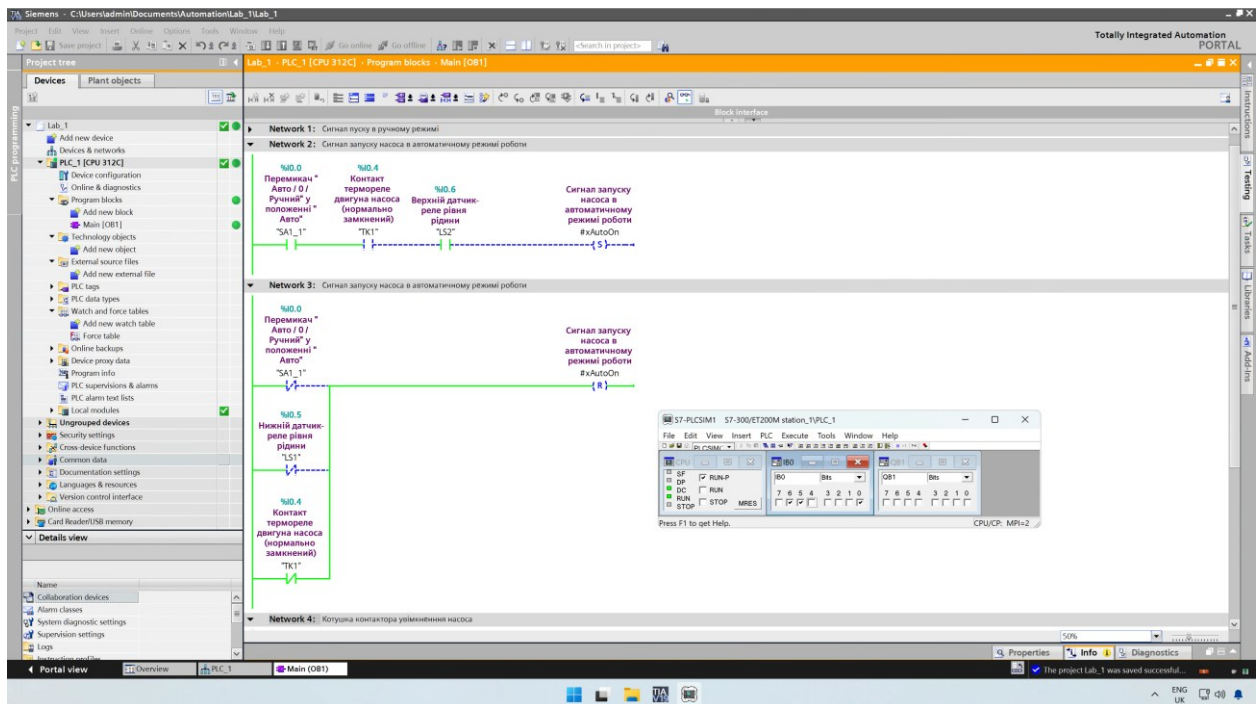


Рисунок 1.58 Перевірка алгоритму захисту від перегріву насоса.

Перевіряємо алгоритм захисту від сухого ходу насоса:

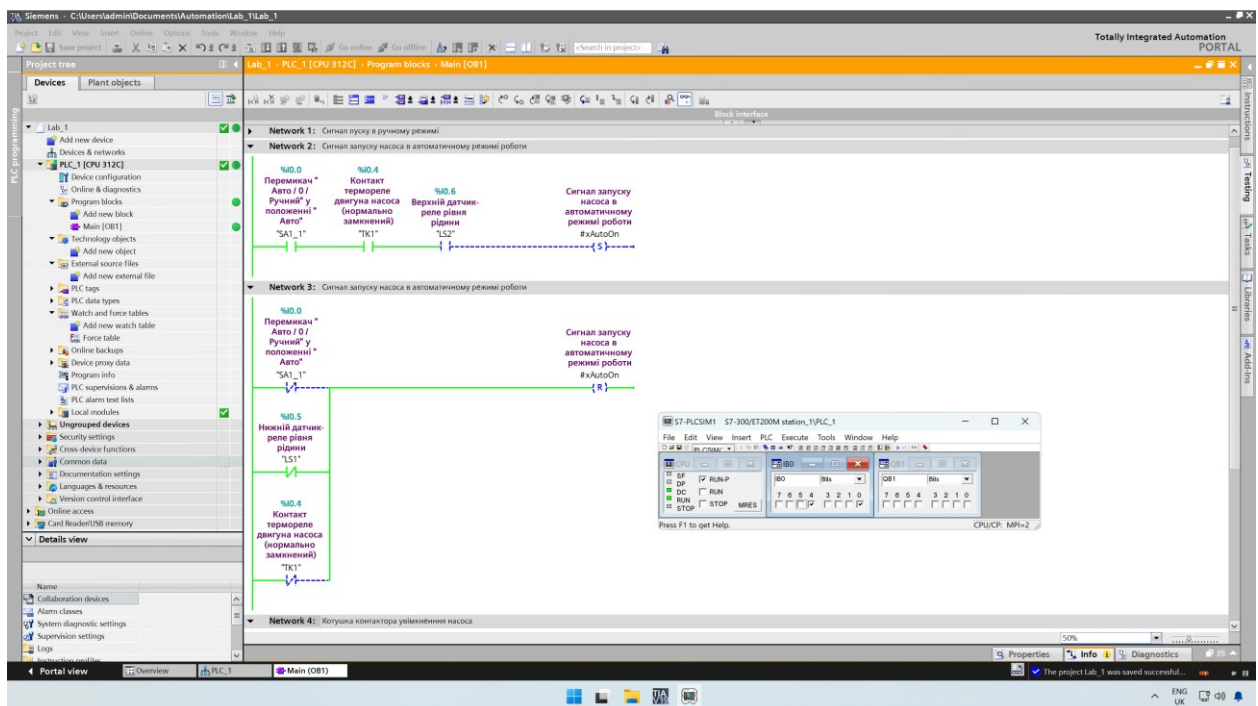


Рисунок 1.59 Перевірка алгоритму захисту від сухого ходу насоса.

Якщо зникнення сигналів від нижнього датчика-реле рівня рідини або від нормально замкненого контакту термореле двигуна насоса призводять до вимкнення насоса в автоматичному режимі (бачимо у вікні симулятора, що прапорці QV1.0 та QV1.1, які відповідають сигналам керування насосом та

світловим індикатором, вимикаються), можна зробити висновок, що алгоритми захисту функціонують вірно.

1.4 Контрольні завдання

1. Модифікувати програму так, щоб дозвіл на запуск в автоматичному режимі відбувався за умови потрібного положення перемикача SA1 та натисканні додаткової кнопки SB3.
2. Модифікувати програму так, щоб у разі спрацювання термозахисту увімкнення насоса блокувалося до моменту, коли оператор натисне кнопку скидання аварій SB4.
3. Переписати алгоритм роботи в автоматичному режимі по аналогії з алгоритмом самопідхвату в ручному режимі (секція №1), використавши звичайну котушку реле xAutoOn.
4. Змінити програму таким чином, щоб можна було замінити 3-позиційний перемикач двома кнопками без фіксації.

2 Лабораторна робота №2.

Мета роботи: отримати навички роботи з аналоговими сигналами, програмування релейно-контактних схем на мові LD з використанням функцій та функціональних блоків.

2.1 Завдання

1. Розробити програму керування насосом зі змінною продуктивністю, яка повинна:
 - 1.1. Обирати ручний або автоматичний режим роботи насоса;
 - 1.2. У ручному режимі керувати насосом за допомогою кнопок «Пуск» і «Стоп» з максимальною продуктивністю;
 - 1.3. В автоматичному режимі підтримувати тиск в трубопроводі, впливаючи на продуктивність насоса;
 - 1.4. Реалізувати захист від перегріву двигуна насоса;
 - 1.5. Реалізувати керування частотним перетворювачем насоса.
2. Модифікувати проєкт відповідно до контрольних завдань.
3. Дослідити роботу розробленого програмного проєкту.

Написати мовою LAD програму, яка:

- а) вмикає насос перемикачем “Авто/0/Ручн” і кнопками “Старт/Стоп”
- б) вимірює тиск в трубопроводі (функціональний блок масштабування фізичного сигналу в одиниці тиску, з можливістю фільтрації сигналу);
- в) підтримує заданий тиск у трубопроводі за допомогою ПД-регулятора

2.2 Опис установки для автоматизації

Установка представляє трубопровід, у якому за допомогою насоса підтримується надлишковий тиск рідини. Трубопровід обладнано аналоговим датчиком тиску. Керування насосом відбувається за допомогою частотного перетворювача, який змінює продуктивність насоса (оберти двигуна) залежно від команд ПЛК. Насос обладнано датчиком перегріву двигуна – термореле. Частотний перетворювач сприймає дискретну команду на запуск, аналоговий сигнал від ПЛК про потрібну частоту обертання в діапазоні 0-100% від максимальної частоти. Частотний перетворювач за допомогою дискретного сигналу повідомляє ПЛК про свою несправність.

Щит керування насосом обладнаний засобами людино-машинної взаємодії: трипозиційним перемикачем вибору режиму роботи «Авто/0/Ручний», кнопками «Пуск» і «Стоп».

У ручному режимі насос вмикається кнопками «Пуск» і «Стоп» і працює на максимальній продуктивності. В автоматичному режимі насос працює за значенням від датчика надлишкового тиску рідини в трубопроводі і повинен підтримувати заданий надлишковий тиск.

Перелік вхідних та вихідних сигналів ПЛК можна знайти в таблиці 2.1.

Таблиця 2.1. Входи та виходи ПЛК

№ п/п	В/В	Тип В/В	Опис
1	SA1_1	Дискретний вхід	Перемикач «Авто/0/Ручний» у положенні «Авто»
2	SA1_2	Дискретний вхід	Перемикач «Авто/0/Ручний» у положенні «Ручний»
3	SB1	Дискретний вхід	Кнопка «Пуск» у ручному режимі
4	SB2	Дискретний вхід	Кнопка «Стоп» у ручному режимі
5	TK1	Дискретний вхід	Контакт термореле двигуна насоса (нормально замкнений)
6	PT1	Аналоговий вхід	Датчик тиску рідини в трубопроводі
7	AL1	Дискретний вхід	Сигнал від частотного перетворювача про несправність
8	DO1	Дискретний вихід	Дозвіл на роботу частотного перетворювача
9	HL1	Дискретний вихід	Світловий індикатор команди увімкнення насоса
10	AO1	Аналоговий вихід	Команда заданої продуктивності частотного перетворювача

2.3 Порядок виконання роботи

2.3.1 Налаштування проєкту

1. Запустити середовище розробки
2. Створити новий проєкт з такими параметрами:
 - Project name : Lab_02

- Path: місце збереження директорії проєкта, за замовчуванням **C:\Users\admin\Documents\Automation**
- Author: Своє прізвище та ім'я
- Comment: Короткий зміст завдання на лабораторну роботу

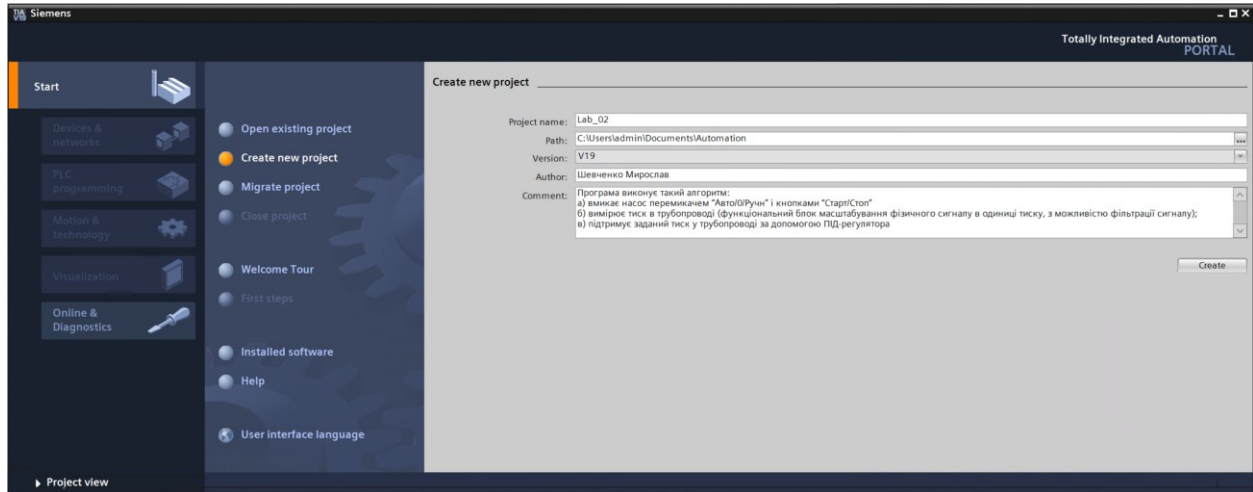


Рисунок 2.1 Створення нового проєкта.

2.3.2 Вибір моделі контролера

Для нашого проєкту необхідно додати програмований логічний контролер та виконати його конфігурування. Для додавання контролера можна скористатись, як у першій лабораторній роботі, пунктом **Add new device** у дереві нового проєкта. З урахуванням рівня складності поставленої задачі, кількості та типів сигналів для цієї системи автоматизації можна обрати процесорний модуль **CPU 313C** з артикулом **6ES7 313-5BG04-0AB0**. Цей процесорний модуль має вбудовані дискретні та аналогові канали вводу/виводу, тож для цей вибір дає можливість не використовувати додаткові модулі вводу /виводу. Відповідно до технічних характеристик, обраний контролер має вбудовані 24 дискретні входи, 16 дискретних виходів, 5 аналогових входів та 2 аналогові виходи. Цієї кількості входів та виходів вистачить з великим запасом для виконання поставленої задачі та перспективи її розширення, а за необхідності обрана модель дозволяє збільшити кількість сигналів за рахунок встановлення додаткових модулів розширення.

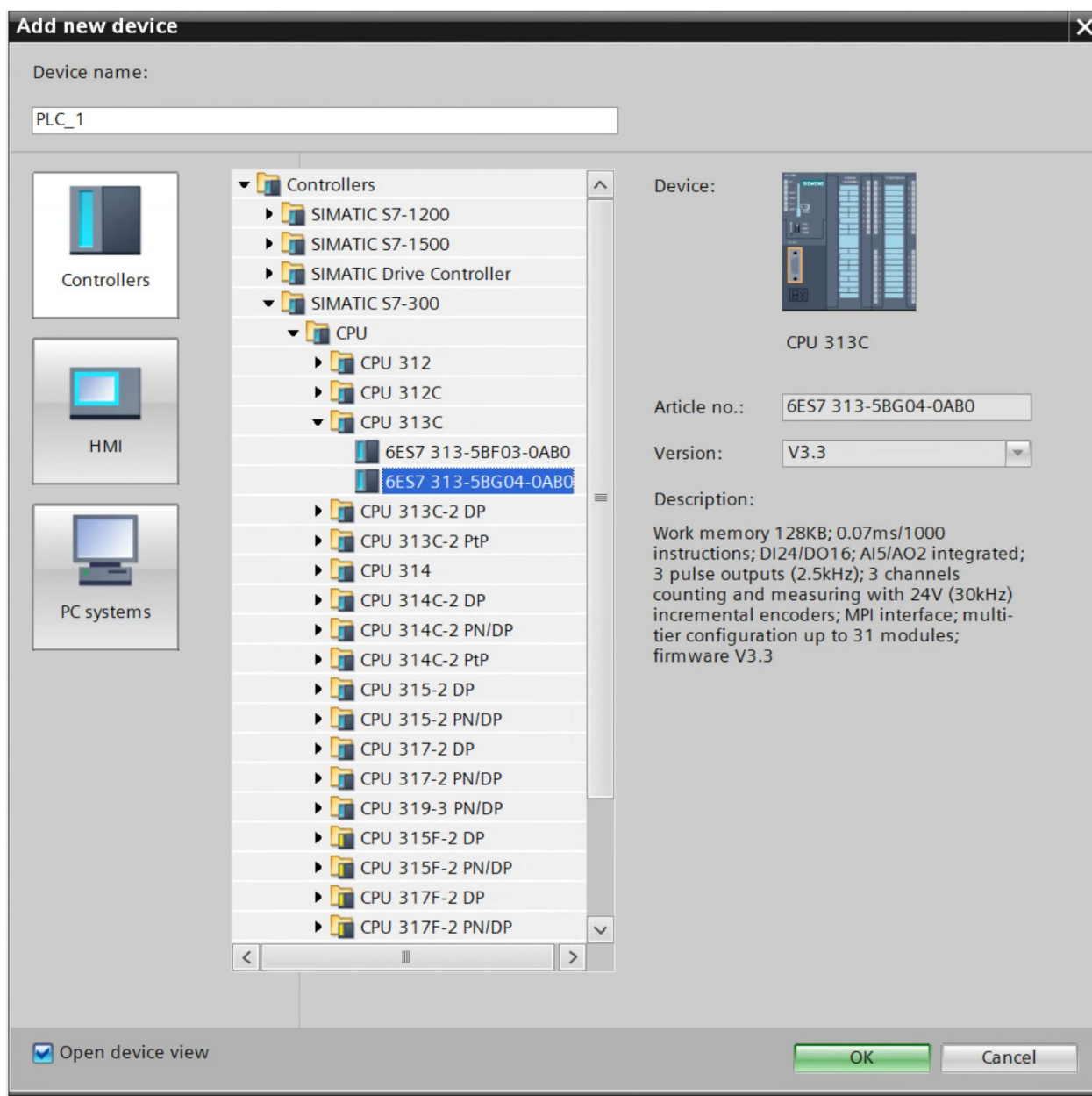


Рисунок 2.2 Вибір контролера.

Порядок конфігурування контролера аналогічний першій лабораторній роботі: необхідно виконати конфігурування рівня безпеки, адрес та типів сигналів для каналів вводу/виводу. Для встановлення адрес входів та виходів переходимо до редагування властивостей контролера з контекстного меню контролера.

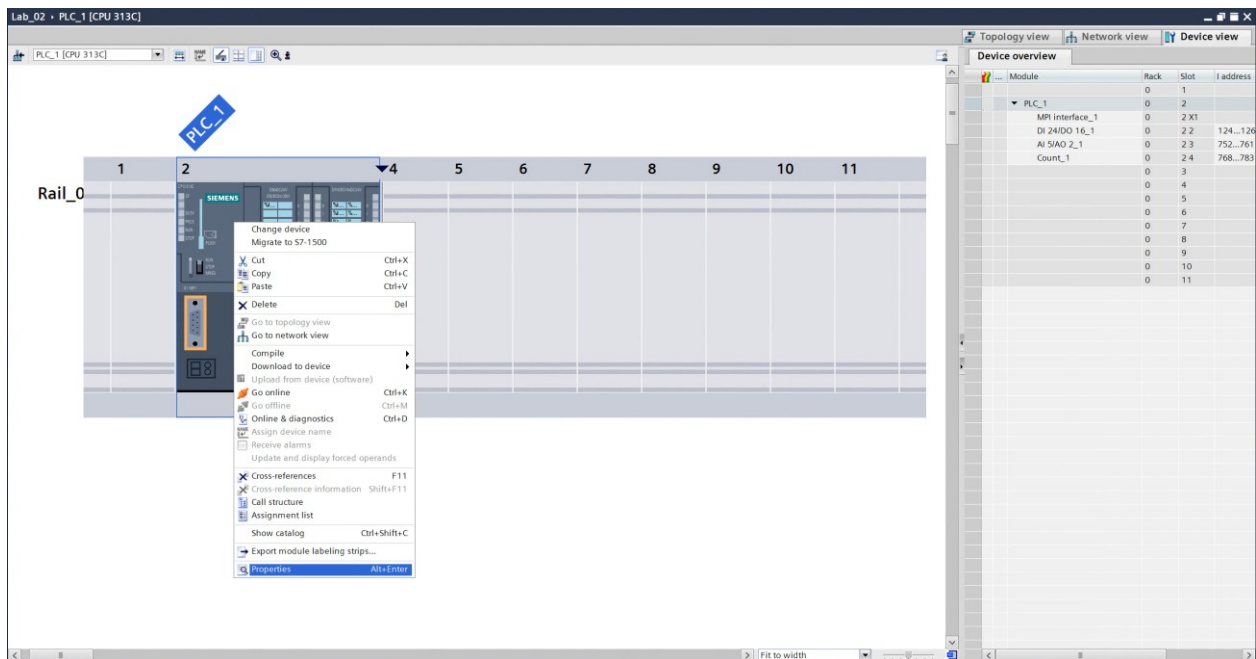


Рисунок 2.3 Перехід до редагування властивостей контролера.

Встановлюємо адреси дискретних входів та виходів відповідно з нульового вхідного байту та нульового вихідного байту.

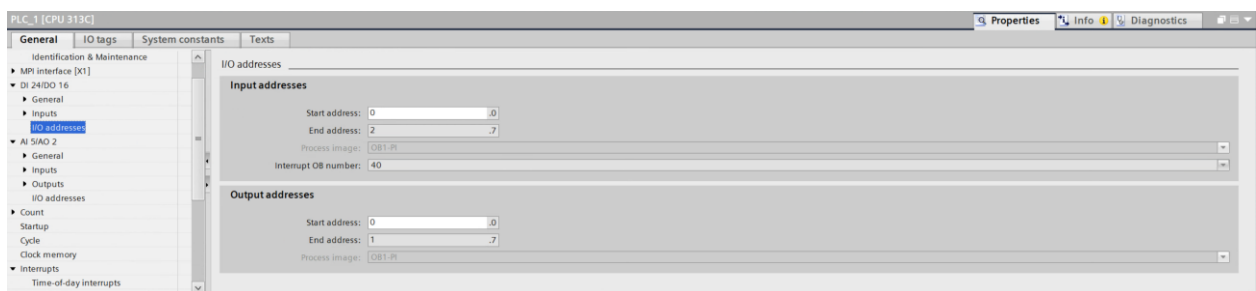


Рисунок 2.4 Встановлення діапазонів адрес дискретних сигналів.

Для аналогових входів будемо використовувати діапазон вхідних адрес з 3 по 12 байт, а для аналогових виходів – з 2 по 5 вихідний байти.

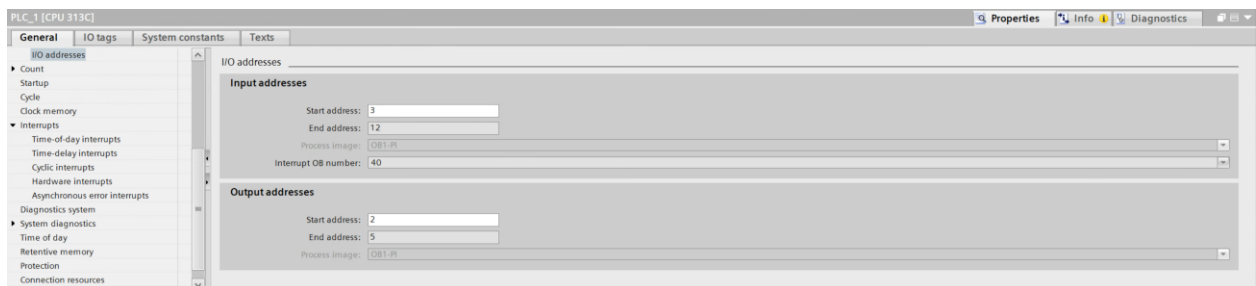


Рисунок 2.5 Встановлення діапазонів адрес аналогових сигналів.

Для зручності роботи з програмою будемо використовувати позначення для сигналів відповідно до таблиці 2.1 з поясненнями їх призначення у полі **Comment**:

PLC_1 [CPU 313C]					
Properties Info Diagnostics					
General IO tags System constants Texts					
Name	Type	Address	Tag table	Comment	
PT1	Int	%IW3	Default tag table	Датчик тиску рідини в трубопроводі	
	Int	%IW5			
	Int	%IW7			
	Int	%IW9			
	Int	%IW11			
AO1	Int	%QW2	Default tag table	Команда заданої продуктивності частотного перетворювача	
	Int	%QW4			
SA1_1	Bool	%I0.0	Default tag table	Перемикач "Авто/О/Ручний" у положенні "Авто"	
SA1_2	Bool	%I0.1	Default tag table	Перемикач "Авто/О/Ручний" у положенні "Ручний"	
SB1	Bool	%I0.2	Default tag table	Кнопка "Пуск" у ручному режимі	
SB2	Bool	%I0.3	Default tag table	Кнопка "Стоп" у ручному режимі	
TK1	Bool	%I0.4	Default tag table	Контакт термореле двигуна насоса (нормально замкнений)	
AL1	Bool	%I0.5	Default tag table	Сигнал від частотного перетворювача про несправність	
	Bool	%I0.6			
	Bool	%I0.7			
	Bool	%I1.0			
	Bool	%I1.1			
	Bool	%I1.2			
	Bool	%I1.3			
	Bool	%I1.4			
	Bool	%I1.5			
	Bool	%I1.6			
	Bool	%I1.7			
	Bool	%I2.0			
	Bool	%I2.1			
	Bool	%I2.2			
	Bool	%I2.3			
	Bool	%I2.4			
	Bool	%I2.5			
	Bool	%I2.6			
	Bool	%I2.7			
DO1	Bool	%Q0.0	Default tag table	Дозвіл на роботу частотного перетворювача	
HL1	Bool	%Q0.1	Default tag table	Світловий індикатор команди увімкнення насоса	
	Bool	%Q0.2			
	Bool	%Q0.3			
	Bool	%Q0.4			
	Bool	%Q0.5			
	Bool	%Q0.6			
	Bool	%Q0.7			
	Bool	%Q1.0			
	Bool	%Q1.1			
	Bool	%Q1.2			
	Bool	%Q1.3			
	Bool	%Q1.4			
	Bool	%Q1.5			
	Bool	%Q1.6			
	Bool	%Q1.7			

Рисунок 2.6 Позначення сигналів вводу/виводу.

2.3.3 Програмування алгоритмів увімкнення частотного перетворювача насоса

1. Важливою складовою ефективного супроводження коду впродовж тривалого часу є якість коментарів та іменування змінних. У такому випадку код буде просто розуміти з погляду його призначення та реалізації функціональності. Для спрощення повторного використання коду в інших проєктах, де іменування датчиків та виконавчих пристроїв може бути інакшим, змінним контактів і реле варто надавати апаратно-незалежні, але однозначно зрозумілі імена. Змінні оголошуються під час встановлення котушок і реле в схему або завчасно, в редакторі змінних. Можна використовувати локальні змінні за прикладом першої лабораторної роботи, але для зручності роботи над проєктом та полегшення його

масштабування краще виділити для змінних окрему ділянку пам'яті у вигляді блоку даних. Для цього у дереві проєкта потрібно обрати пункт **Add new block**, а у вікні створення блоку обрати **Data block** та тип блоку **Global DB**.

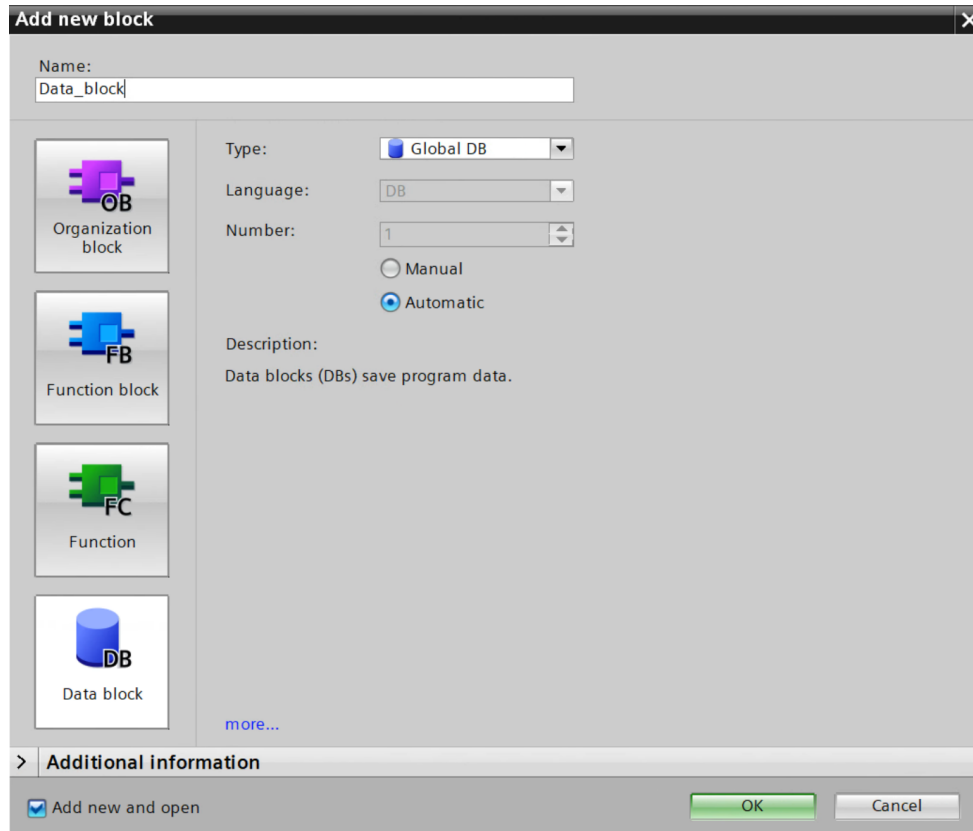


Рисунок 2.7 Додавання блоку даних.

Назва створеного блоку даних – **Data_block**.

Тепер можна оголосити потрібні змінні **xAutoOn** та **xManualOn** типу **Bool** для увімкнення насоса через частотний перетворювач в ручному та автоматичному режимах.

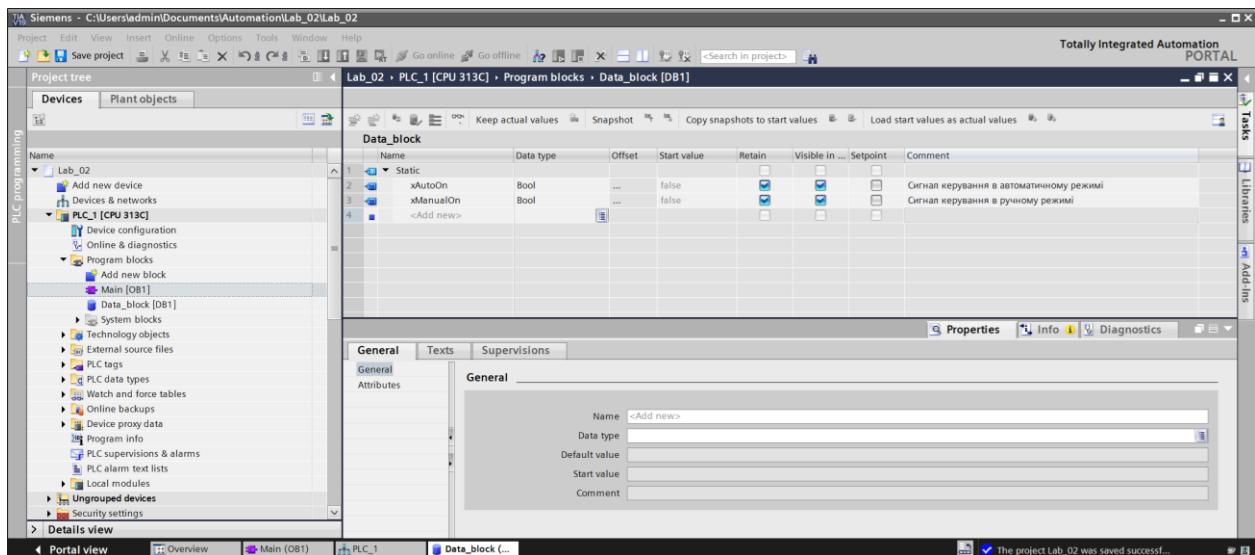


Рисунок 2.8 Оголошення змінних у блоці даних.

- У секції **Network 1** організаційного блоку **Main (OB1)** реалізувати алгоритм команди увімкнення частотного перетворювача насоса в ручному режимі, аналогічно до відповідного завдання першої лабораторної роботи. Для того, щоб використати оголошені у блоку даних змінні, при виборі змінної потрібно спочатку обрати блок даних (**Data_block**), після цього стануть доступні змінні, які у ньому описані.

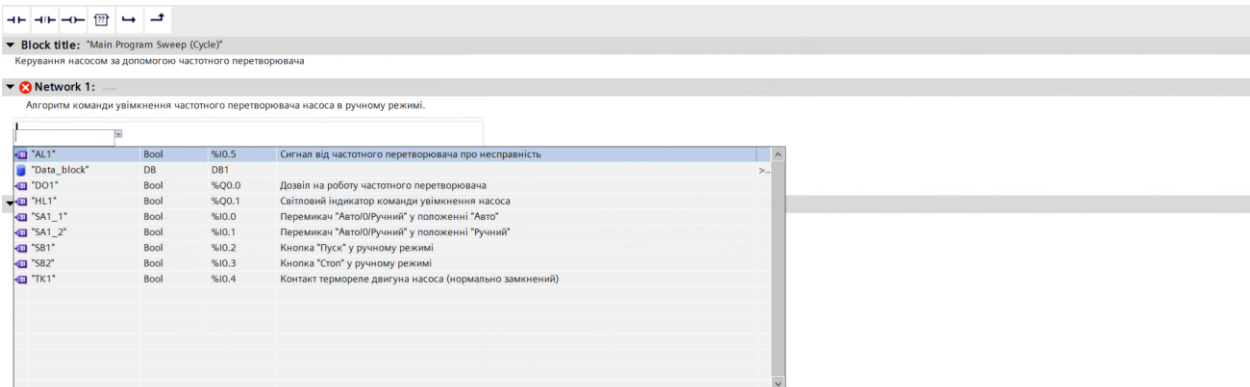


Рисунок 2.9 Використання у програмі визначених змінних.

Готова перша секція повинна мати такий вигляд:

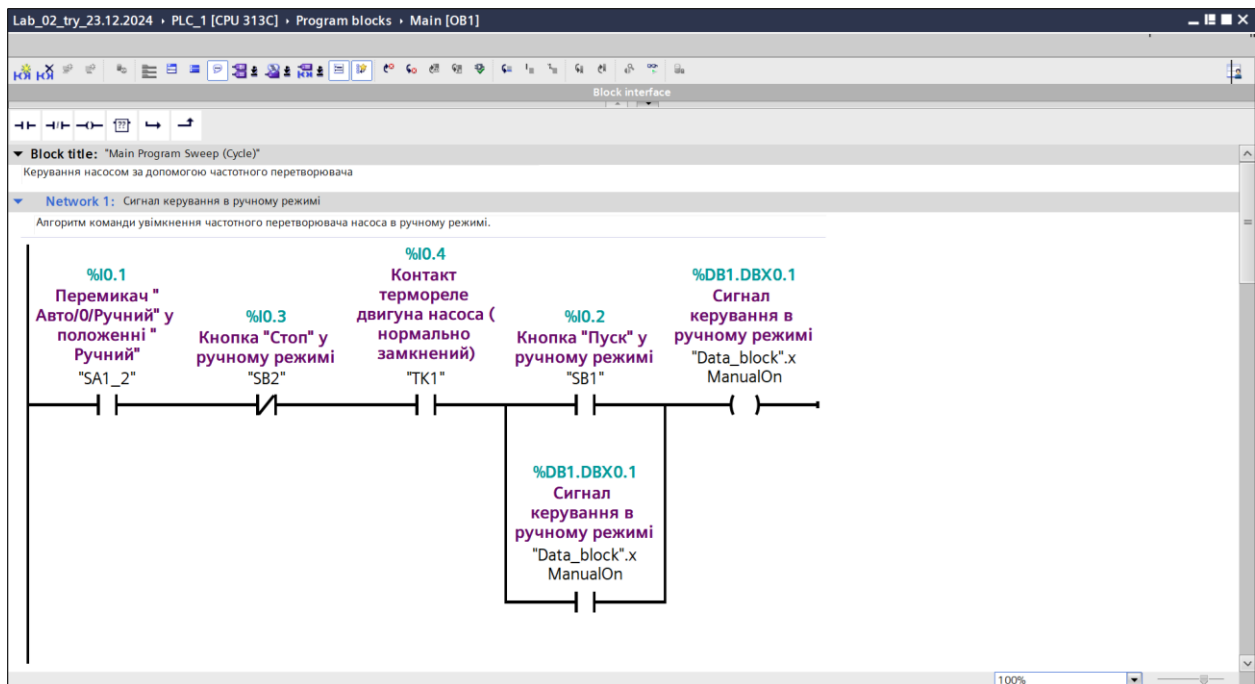


Рисунок 2.10 Алгоритм увімкнення насоса в ручному режимі.

Знаки питання у полі адреси змінної після назви блоку даних означають, що змінні ще не отримали адрес у блоці даних. Адреси будуть визначені після успішної компіляції проєкту.

- Створити секцію **Network 2** в організаційному блоці **Main (OB1)**, реалізувати алгоритм команди увімкнення частотного перетворювача насоса в автоматичному режимі.

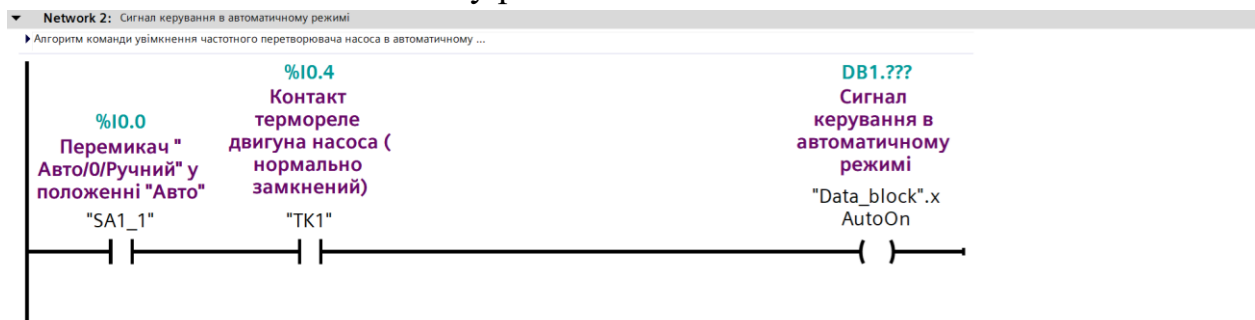


Рисунок 2.11 Алгоритм увімкнення частотного перетворювача в автоматичному режимі.

- Створити секцію **Network 3**, реалізувати алгоритм увімкнення насосу в ручному або автоматичному режимах з індикацією команди на світловому індикаторі.

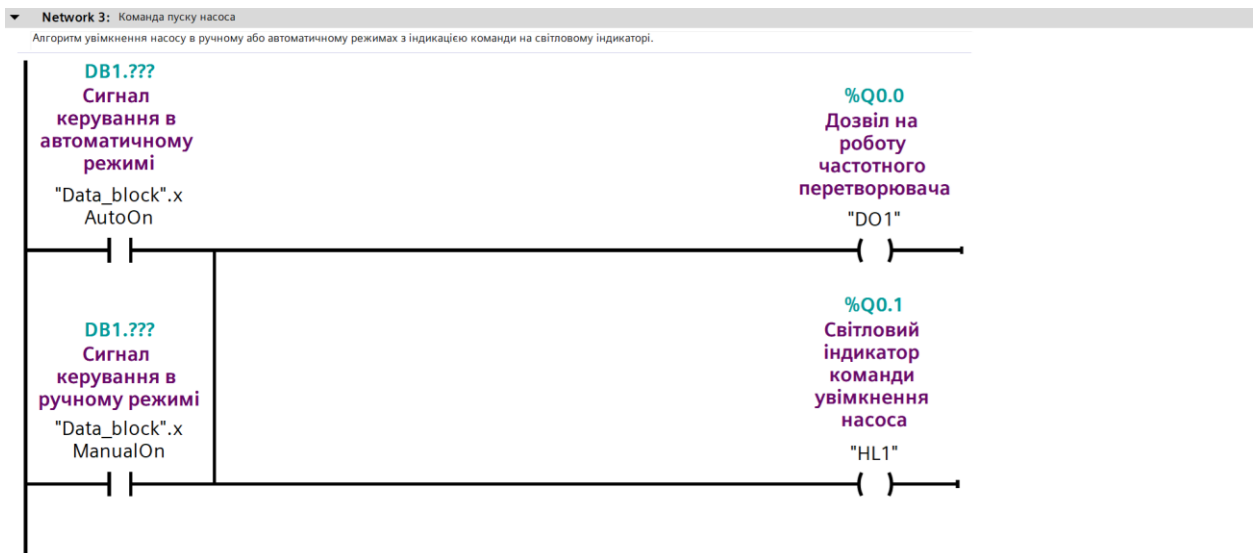


Рисунок 2.12 Алгоритм увімкнення насосу з індикацією команди.

2.3.4 Робота з аналоговими значеннями

1. Показники аналогових датчиків надходять на аналогові входи ПЛК у фізичних одиницях – електричному опорі (Pt100, Pt100), напрузі (0..10В, - 10..10В), силі струму (0..20мА, 4..20мА). Далі за допомогою аналого-цифрового перетворювача (АЦП) ця фізична величина перетворюється на число в конкретному діапазоні. Програміст може перетворити отримане число у зручний для використання формат за допомогою бібліотечних функцій або за допомогою власного алгоритму.

Створити функцію масштабування аналогового сигналу Map: у дереві проєкта в розділі **Program blocks** обрати пункт **Add new block**. У вікні додавання нового компонента, яке при цьому відкриється, потрібно обрати тип компонента **Function** (функція), мову редагування компонента (Language) **SCL** та ввести його назву **Map**.

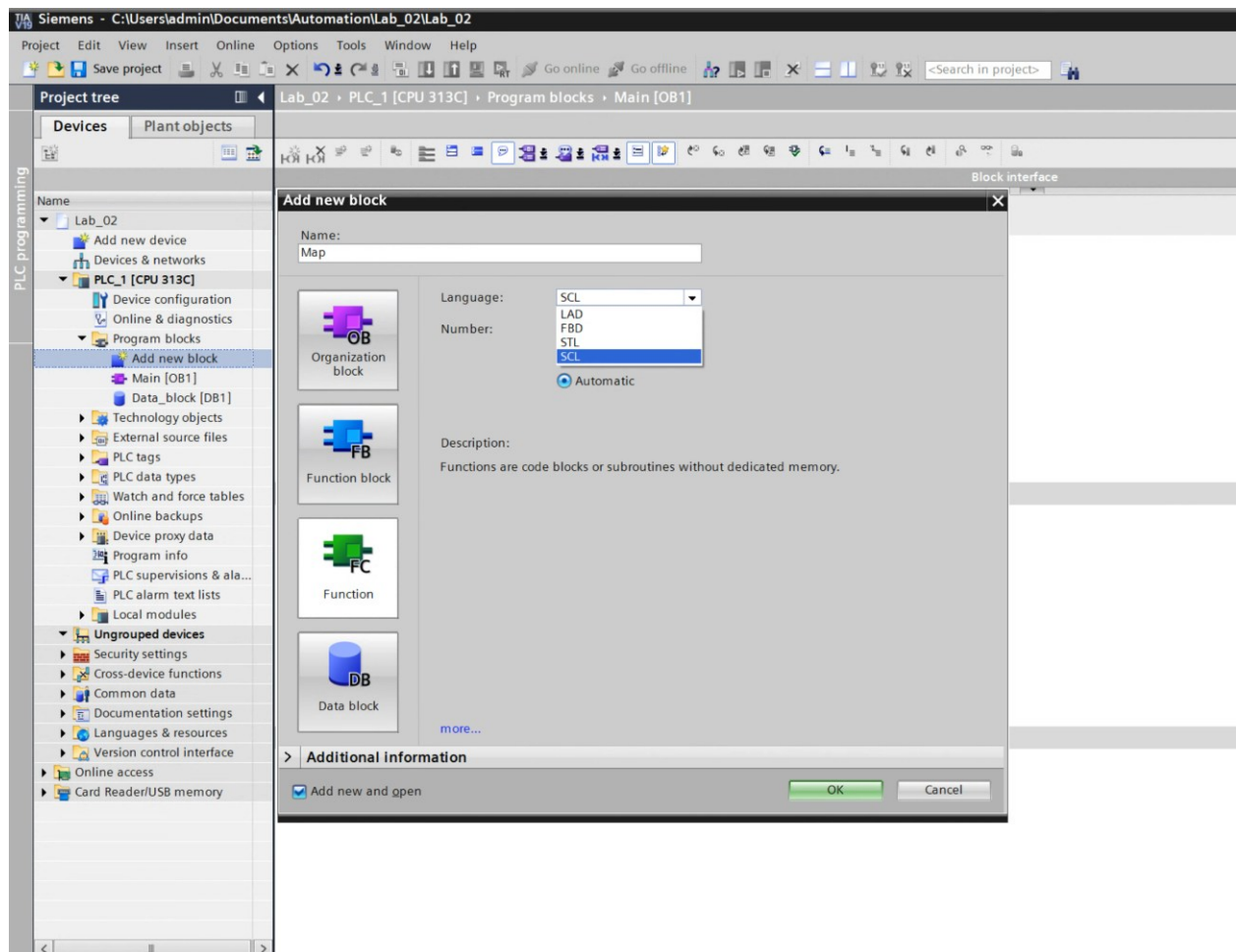


Рисунок 2.13 Створення нової функції.

2. Після вірного виконання попередніх дій має відкритись вікно редактора нової функції **Map** мовою **SCL (Structured Control Language)**. У верхній частині вікна редактора знаходиться перелік змінних, які використовує функція.

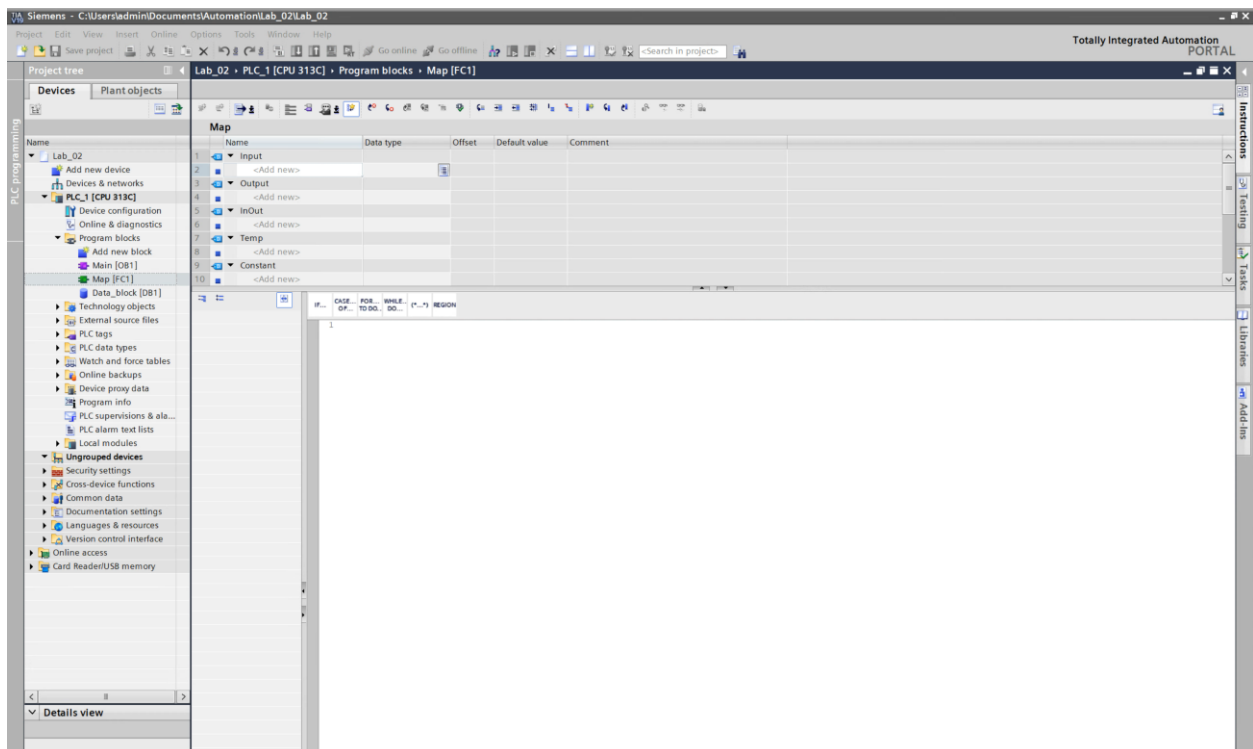


Рисунок 2.14 Вікно редактора SCL функції Map.

3. Необхідно створити інтерфейс функції масштабування сигналу, тобто визначити вхідні та вихідні параметри функції, до яких зможуть отримати доступ інші компоненти програми. Вхідними параметрами функції масштабування будуть:

iInput типу **Int** – значення аналогового сигналу від аналого-цифрового перетворювача;

rMinOutput типу **Real** - мінімальне значення масштабованої величини;

rMaxOutput типу **Real** - максимальне значення масштабованої величини;

Вихідний параметр буде один – значення масштабованої величини **rOutput** типу **Real**.

Для масштабування вхідного сигналу знадобляться також мінімальне та максимальне значення, які може видавати аналого-цифровий перетворювач. Для однополярних сигналів ці значення дорівнюють відповідно **0** та **27648**, їх можна записати у розділі оголошення констант у дійсному форматі (з типом **Real**) для запобігання втрати точності обчислень.

У вікні редактора коду функції Map необхідно записати формулу масштабування аналогового сигналу методом кусково-лінійної інтерполяції:

#rOutput := #rMinOutput + (#iInput - #iMinInput) / (#iMaxInput - #iMinInput) * (#rMaxOutput - #rMinOutput);

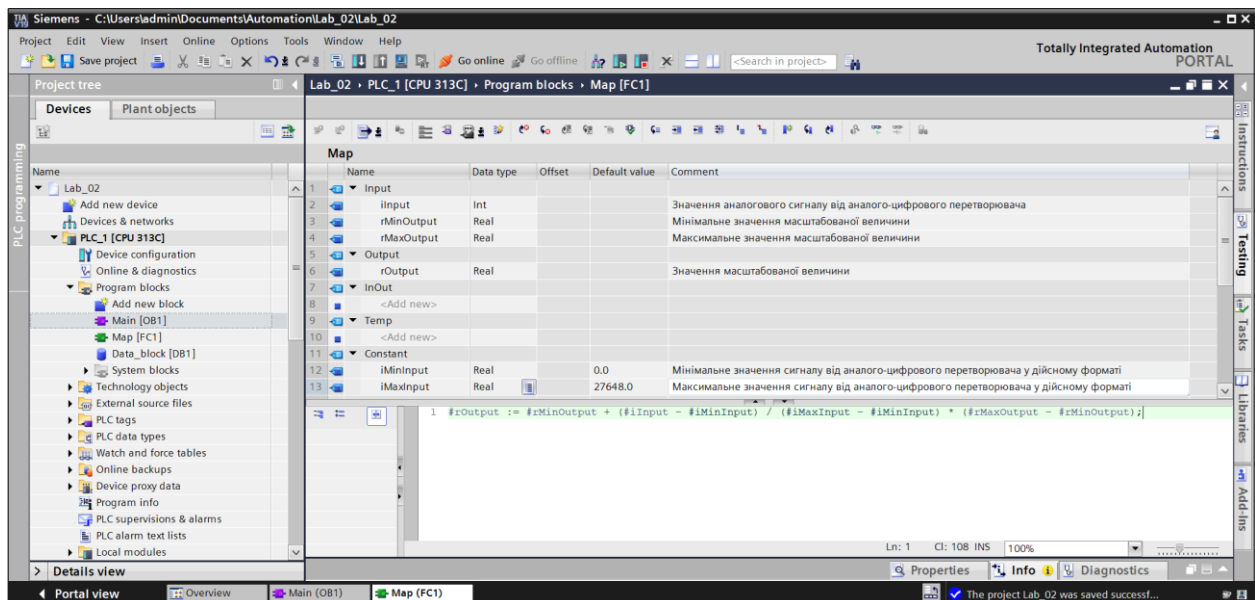


Рисунок 2.15 Створення функції масштабування сигналу.

4. Після створення функції її можна викликати у секції **Network 4** організаційного блоку **Main (OB1)** перетягуванням з дерева проєкта або додати пустий елемент через пункт **Insert empty box** контекстного меню секції та ввести назву створеної функції масштабування **Map**.

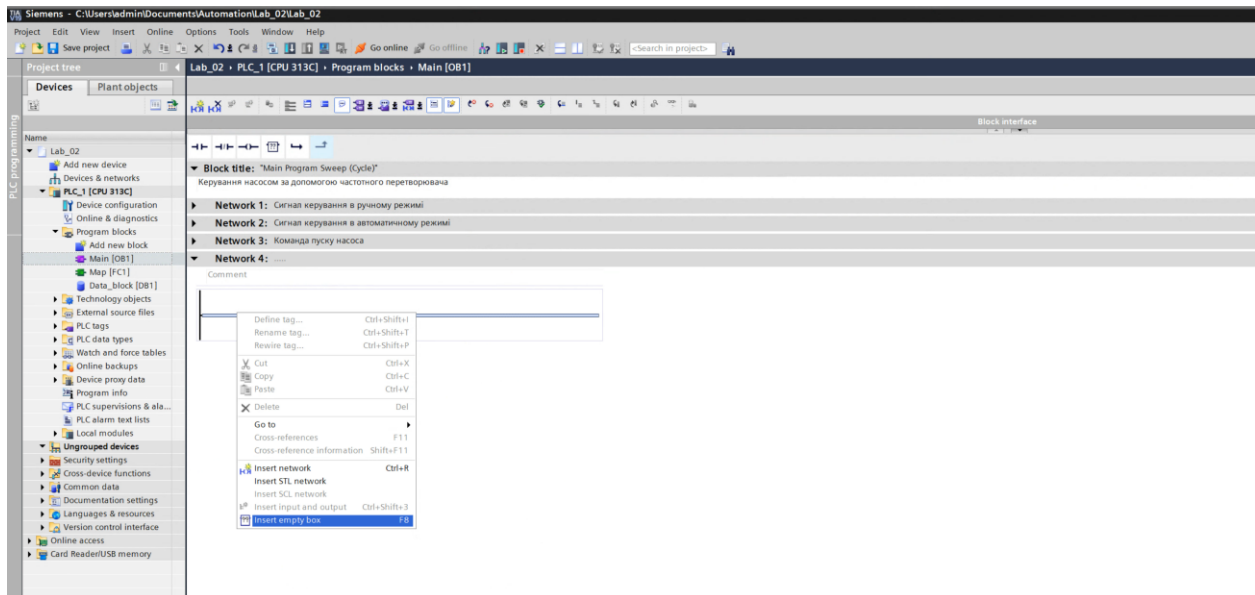


Рисунок 2.16 Додавання нового пустого елемента у секцію.

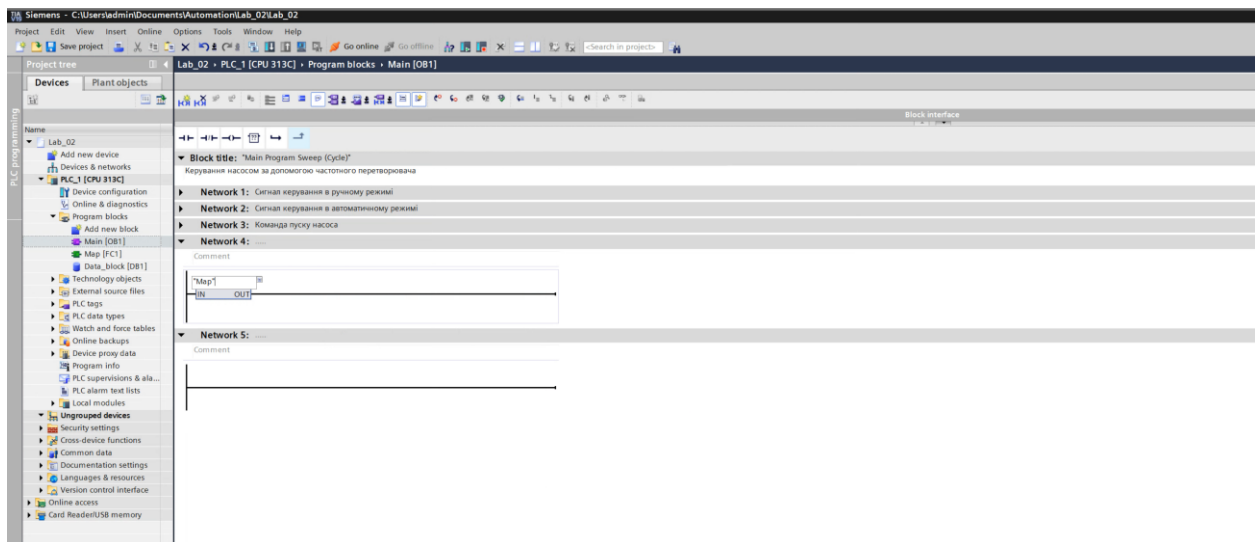


Рисунок 2.17 Виклик функції масштабування сигналу Map.

Для зручності, оголошуємо змінну **rPressureSensor** типу **Real** у блоці даних **Data_block (DB1)**. Ця змінна буде необхідна для збереження значення масштабованого сигналу від датчика тиску, що підключений до першого аналогового входу контролера.

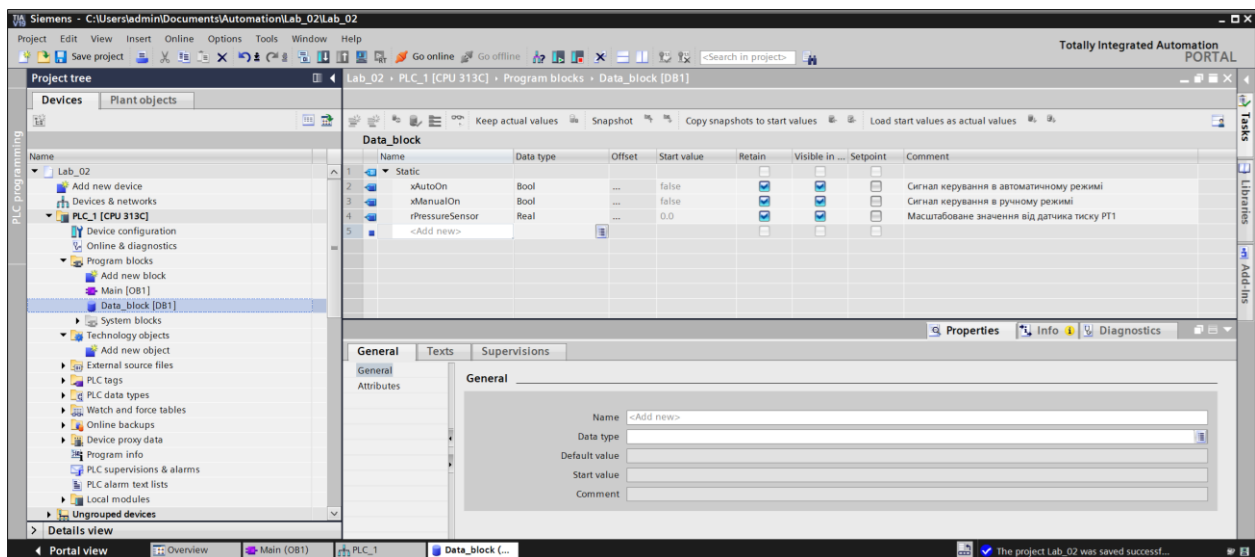


Рисунок 2.18 Оголошення змінної rPressureSensor.

Тепер нову змінну можна буде використовувати в алгоритмі. Тепер можемо вказати параметри функції **Map**: на вхід **iInput** записати змінну **PT1**, яка відповідає першому аналоговому входу контролера, до якого підключений датчик тиску, на вхід **rMinOutput** записуємо значення **0.0** (нижня межа діапазону вимірювання датчика), а на вхід функції **rMaxOutput** – значення **10.0** (верхня межа діапазону вимірювання датчика), на вихід **rOutput** записуємо щойно створену змінну **rPressureSensor**, яка зберігається у блоці даних **Data_block (DB1)**:

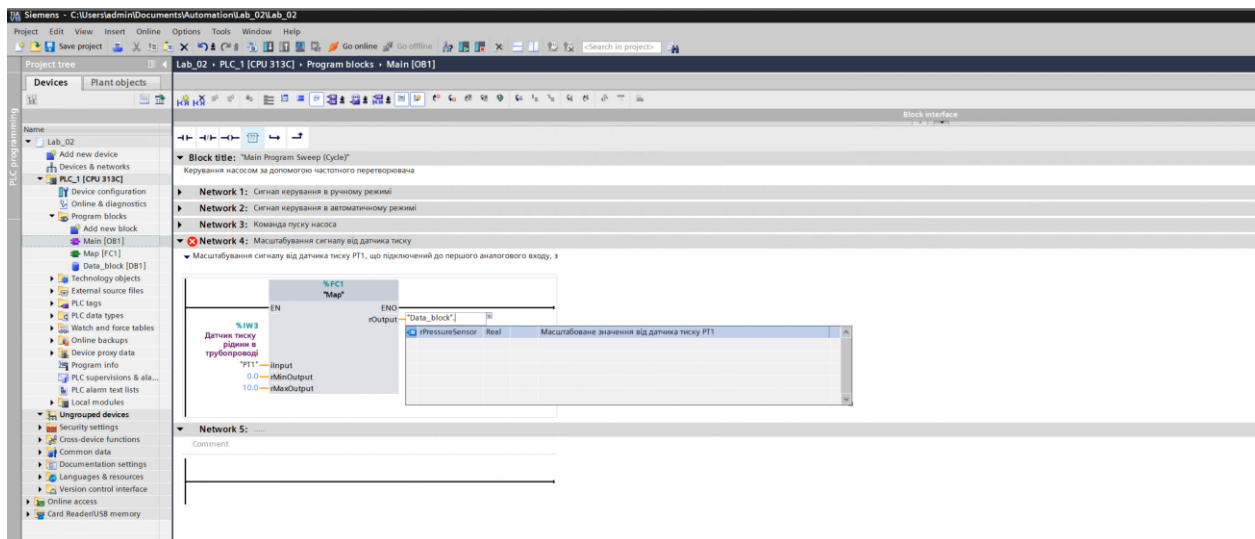


Рисунок 2.19 Заповнення параметрів виклику функції *Map*.

Після заповнення всіх параметрів секція Network 4 повинна мати такий вигляд:

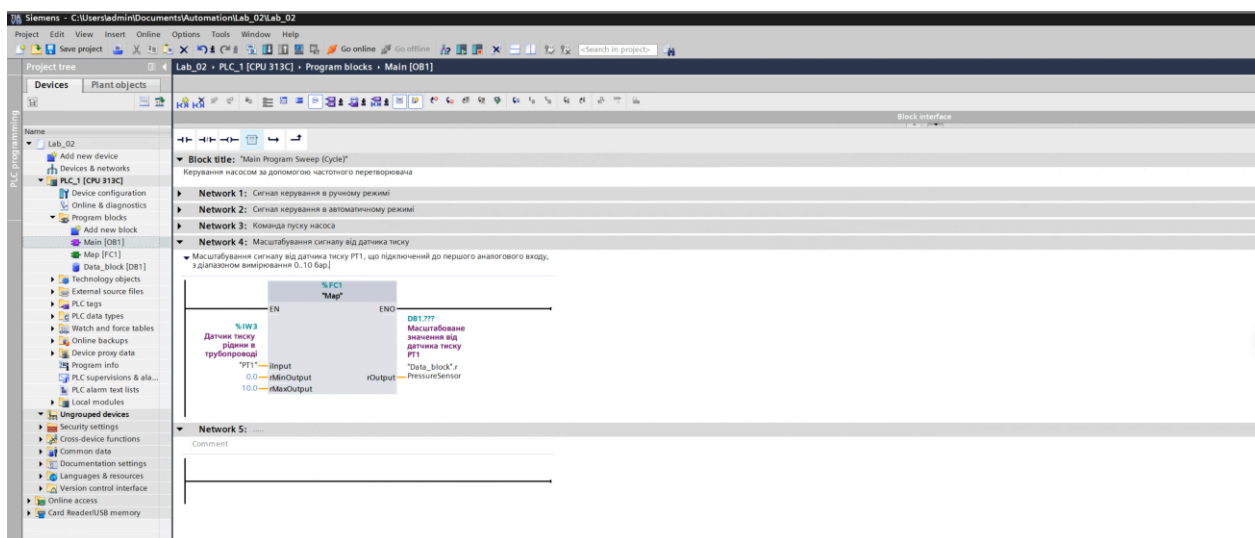


Рисунок 2.20 Секція масштабування сигналу датчика тиску.

2.3.5 Регулювання технологічного параметра

1. Існує багато способів та стратегій регулювання технологічних змінних. Вони залежать від особливостей установки, вимог до якості регулювання, динаміки контуру регулювання. Одним з найпоширеніших наразі підходів є використання пропорційно-інтегрально-диференціальних (ПІД) регуляторів. Як правило, бібліотеки середовищ розробки вже містять готові регулятори. Основні регулятори **TIA Portal** містяться у розділі інструкцій **Technology**. У цьому розділі можна обрати той регулятор, який найкраще підходить для даної задачі.

Оскільки за завданням регулювання тиску відбувається за допомогою частотного перетворювача, до якого підключений двигун насоса, найкраще для цієї задачі підходить ПІД-регулятор з неперервним виходом **CONT_C**.

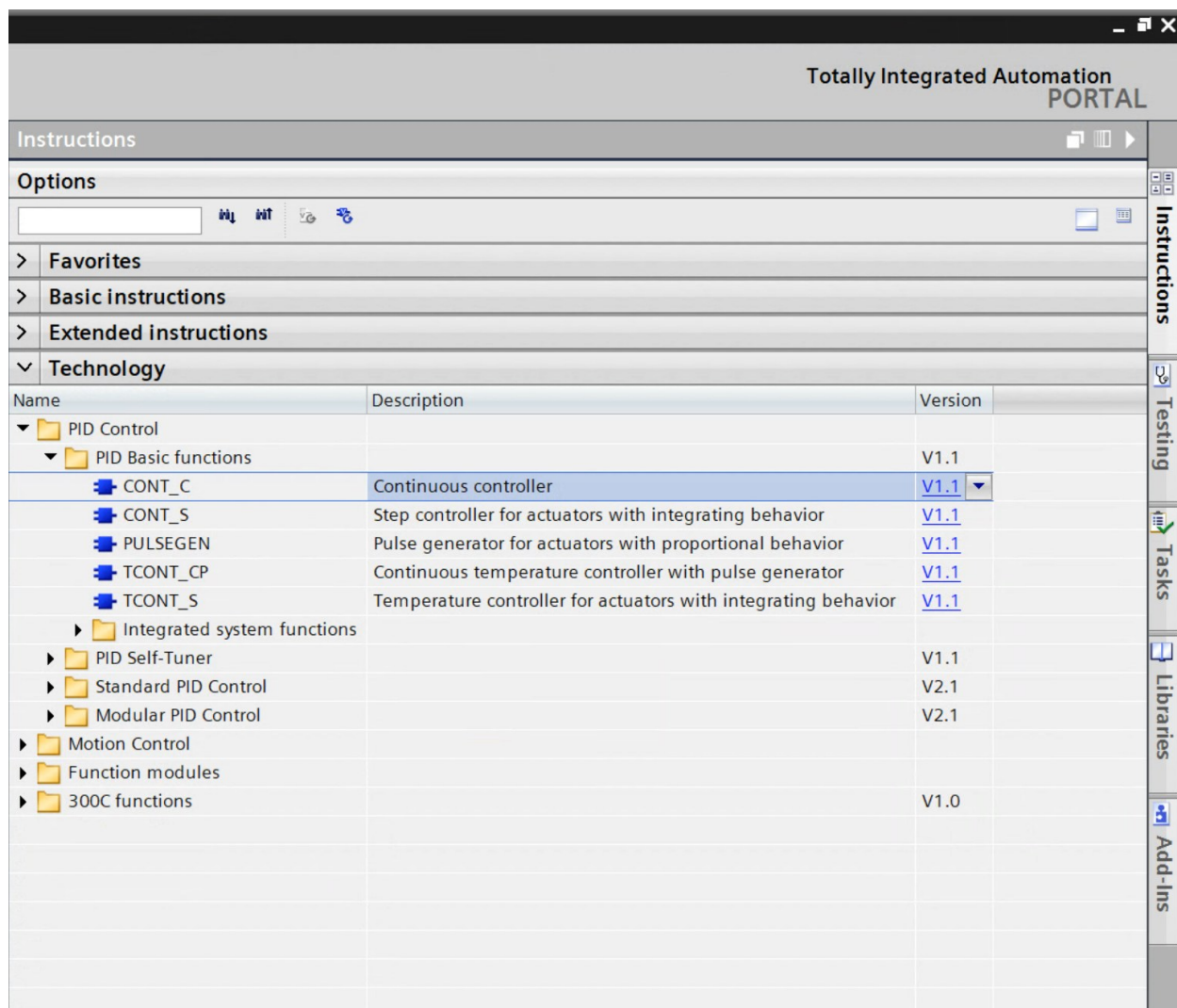


Рисунок 2.21 Основні PID-регулятори TIA Portal.

Для коректної роботи регулятора він повинен запускатись через рівні проміжки часу. Організаційний блок **Main (OB1)** не може забезпечити виконання цієї вимоги, оскільки період його запуску залежить від багатьох чинників та може змінюватись в процесі роботи контролера. Для забезпечення стабільного періоду виклику регулятора його необхідно викликати у циклічному організаційному блоці, який має пріоритет виконання вищий за пріоритет організаційного блоку **Main (OB1)**.

Для проєктів **TIA Portal** з використанням контролерів **S7-300** доступні циклічні організаційні блоки **OB 32..OB 35**. Для того, щоб додати в проєкт новий циклічний організаційний блок, необхідно викликати команду **Add new block** у дереві проєкта **TIA Portal** та у вікні додавання нового блоку обрати розділ **Organization block**, а у списку організаційних блоків обрати розділ

Time interrupts → **Cyclic**, при додаванні блоку варто перевірити, що обрана мова програмування **LAD**:

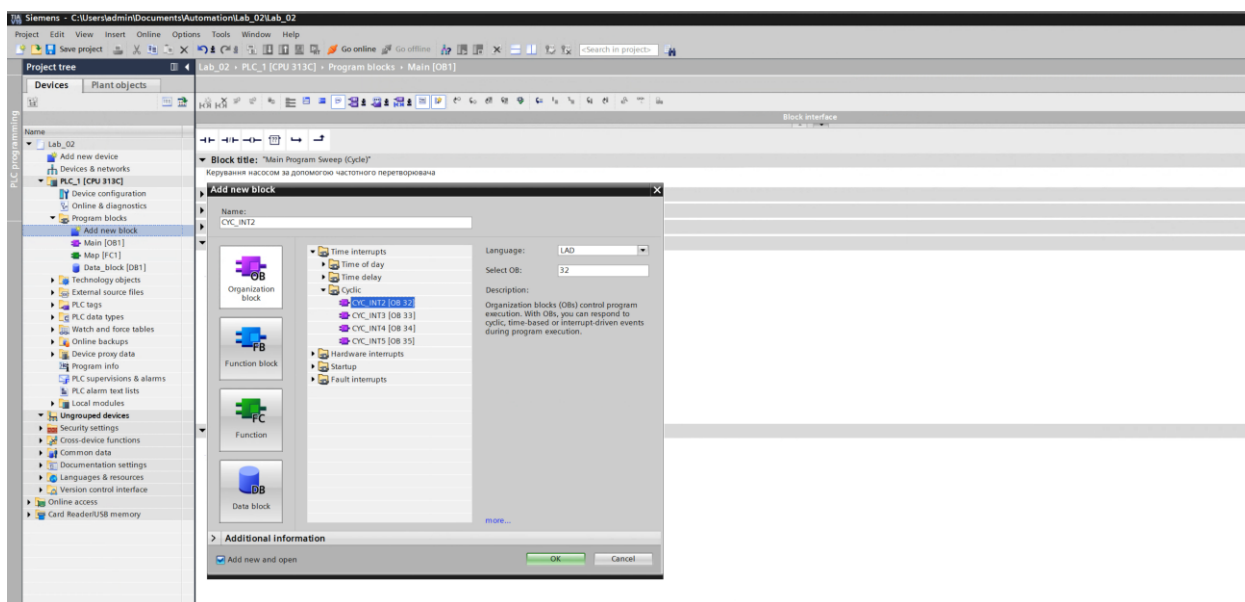
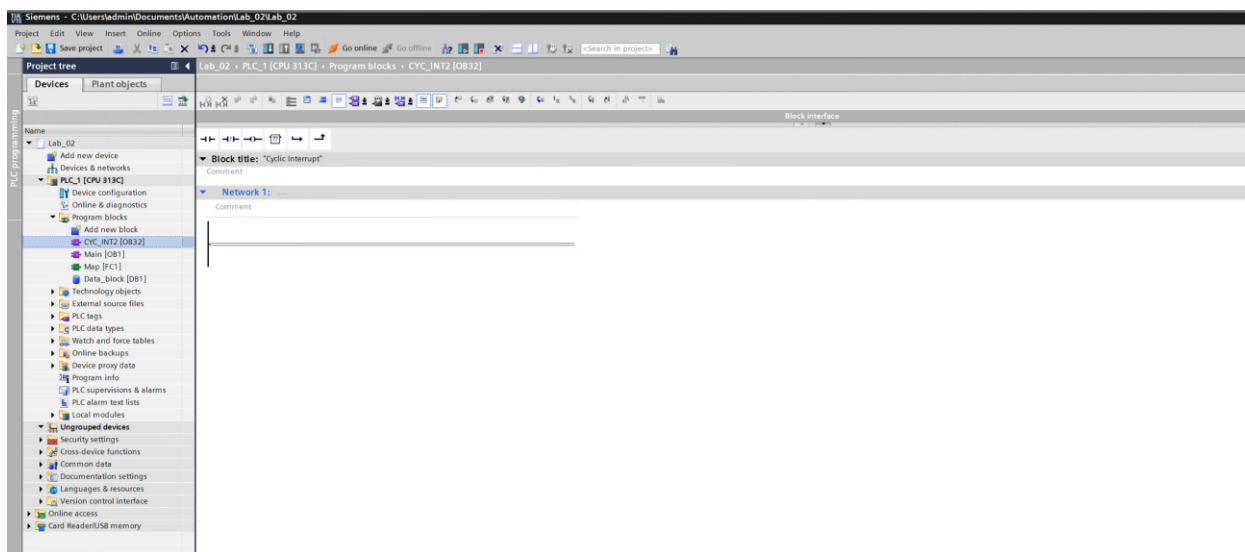


Рисунок 2.22 Додавання нового циклічного організаційного блоку.

Після успішного додавання циклічного організаційного блоку він має з'явитись у дереві проєкта:



*Рисунок 2.23 Результат додавання циклічного організаційного блоку **CYC_INT2 (OB32)**.*

Після додавання блоку **CYC_INT (OB32)** потрібно встановити період його виклику. Налаштування періодів виклику організаційних блоків знаходиться у розділі **Interrupts** → **Cyclic interrupts** параметрів контролера. За замовченням період виклику організаційного блоку **CYC_INT2 (OB32)** дорівнює **1000 мс**, тобто 1 секунда:

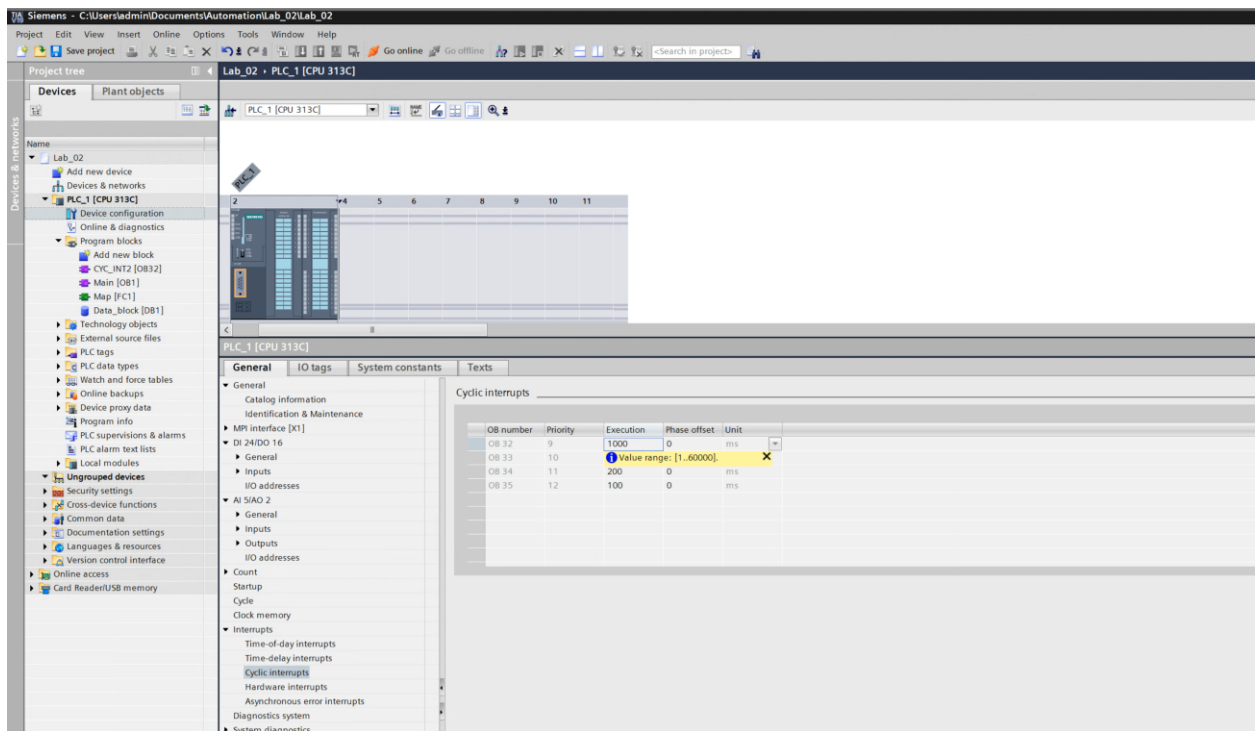


Рисунок 2.24 Розділ налаштувань періодів виклику циклічних організаційних блоків.

Встановлюємо значення періоду виклику **CYC_INT2 (OB32)** рівним **100 мс**:

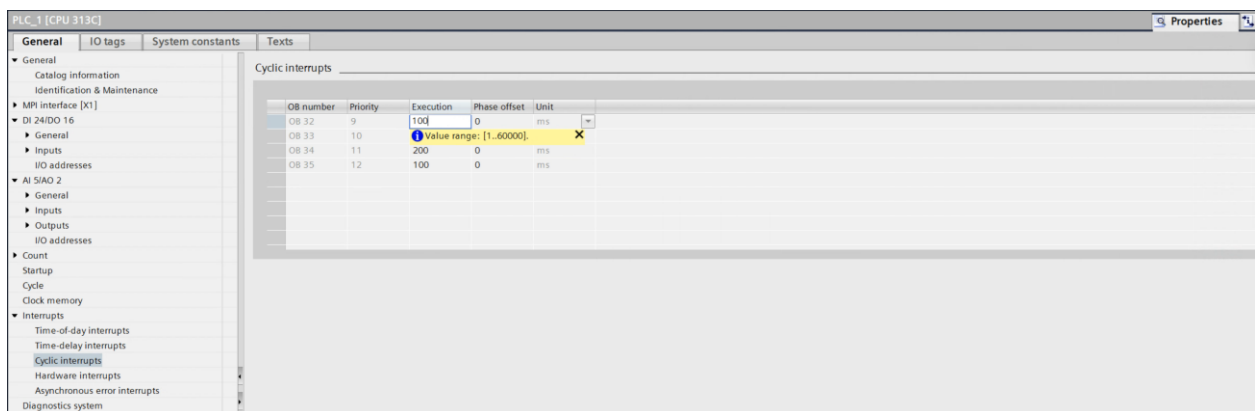


Рисунок 2.25 Зміна періоду виклику **CYC_INT2 (OB32)**.

Після виконання цих налаштувань можемо перейти до редагування алгоритму блоку **CYC_INT2 (OB32)**. Перетягуємо регулятор **CONT_C** зі списку технологічних інструкцій у редактор організаційного блоку **CYC_INT2 (OB32)** для створення екземпляра блоку регулятора з назвою екземпляра **CONT_C_DB**.

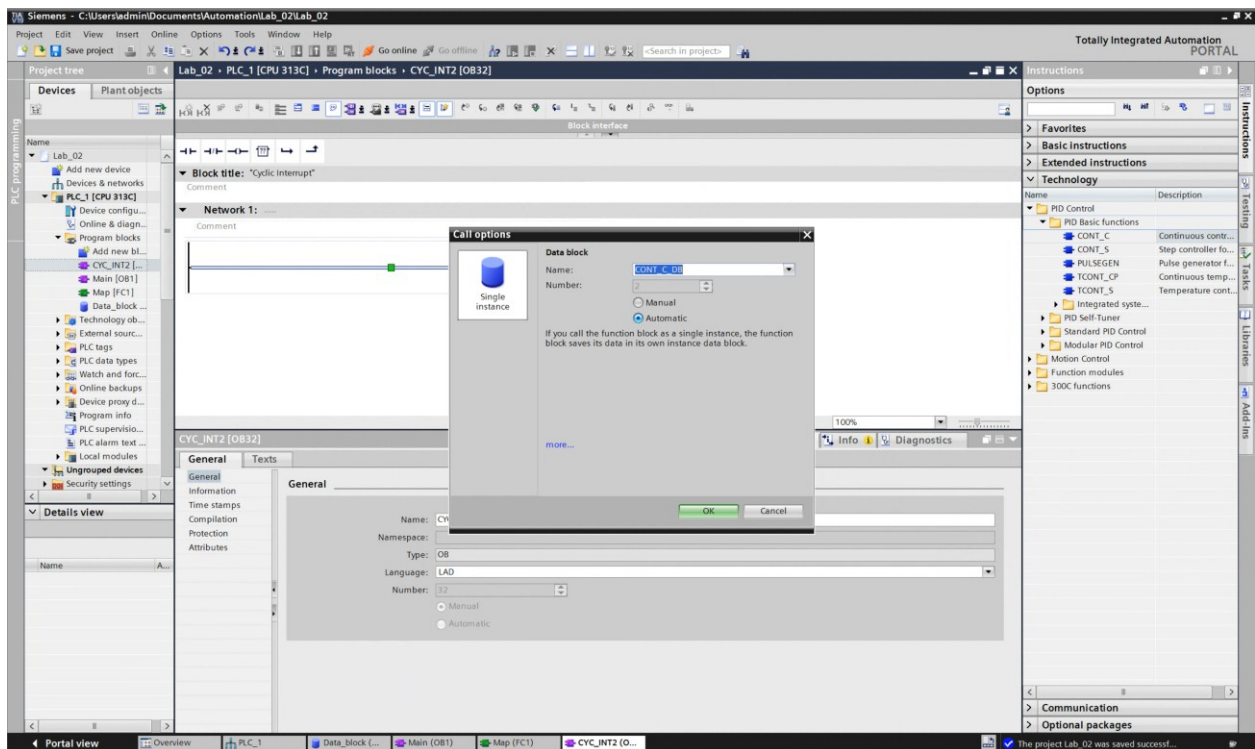


Рисунок 2.26 Створення екземпляра блока регулятора CONT_C.

Як вже зазначалось вище, регулятор CONT_C використовується у алгоритмах неперервного регулювання технологічних параметрів за пропорційно-інтегрально-диференціальним законом. Окрім, власне, параметрів налаштування закону регулювання, цей блок має додаткові параметри налаштування, які полегшують процес налагодження та експлуатації системи регулювання у різних режимах, дають можливість масштабувати регульований параметр та регулюючу дію, задавати діапазон вихідного параметра та змінювати структуру закону регулювання.

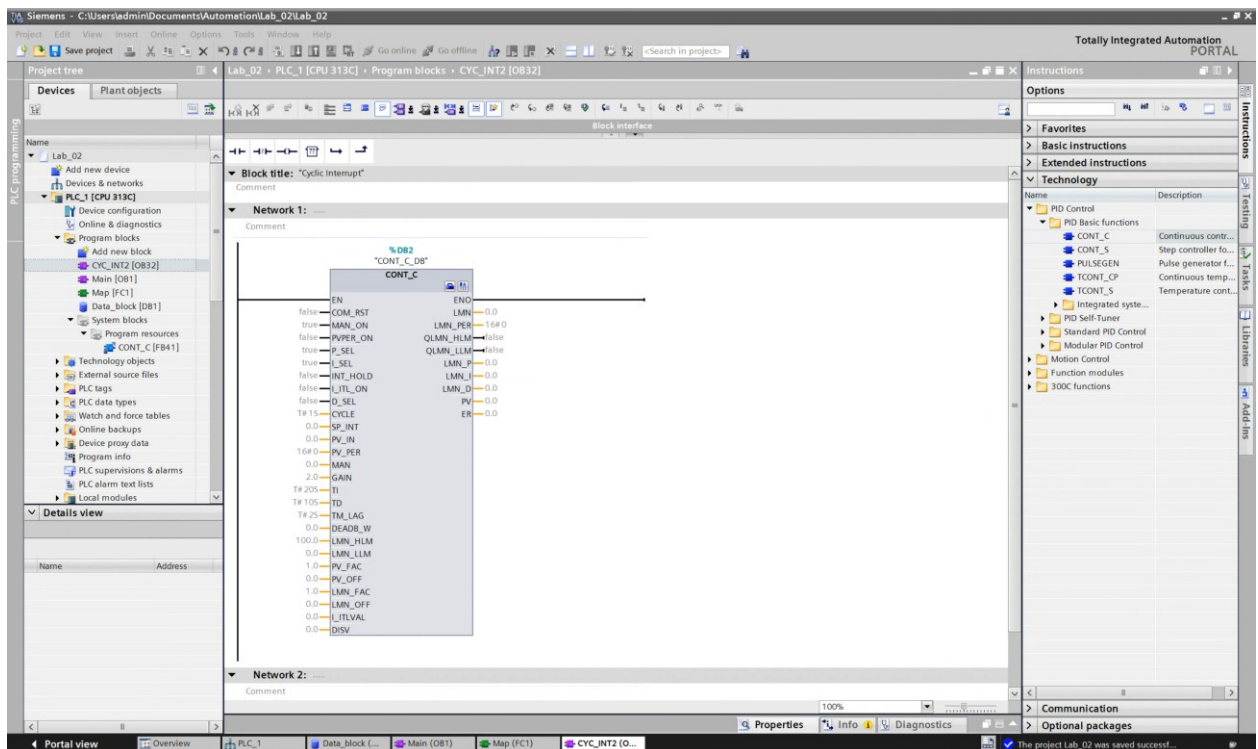


Рисунок 2.27 Екземпляр блоку CONT_C з параметрами за замовчуванням.

Для зручності подальшого використання створюємо окремий блок даних, в якому будуть міститись параметри ПІД-регулятора.

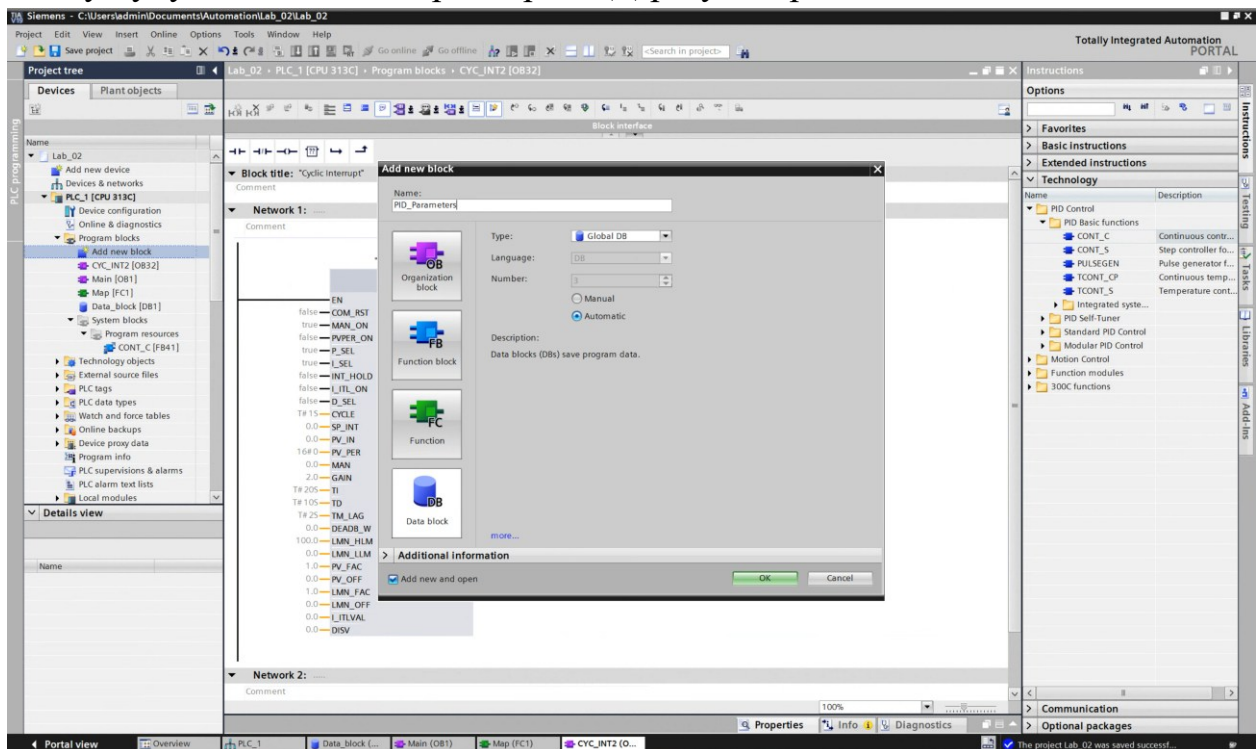


Рисунок 2.28 Створення нового блоку даних для параметрів ПІД-регулятора.

У створеному блоці даних **PID_Parameters** додаємо такі змінні:
xReset типу **Bool** для перезапуску блоку регулятора в разі необхідності;

xP_Sel типу **Bool** для можливості відключення пропорційної частини ПІД-регулятора;

xI_Sel типу **Bool** для можливості відключення інтегральної частини ПІД-регулятора;

xD_Sel типу **Bool** для можливості відключення диференційної частини ПІД-регулятора;

tCycle типу **Time** для передачі параметра періоду виклику блока регулятора;

rSetpoint типу **Real** для можливості зміни завдання регулятора;

rManualOutput типу **Real** для можливості зміни виходу регулятора в ручному режимі роботи;

rProportional типу **Real** для можливості зміни коефіцієнту передачі пропорційної частини регулятора;

tIntegral типу **Time** для можливості зміни параметру налаштування інтегральної частини регулятора;

tDerivative типу **Time** для можливості зміни параметру налаштування диференційної частини регулятора;

tDerivativeLag типу **Time** для можливості зміни інерційності диференційної частини регулятора;

rControlAction типу **Real** для подальшого використання сигналу виходу регулятора (регулююча дія)

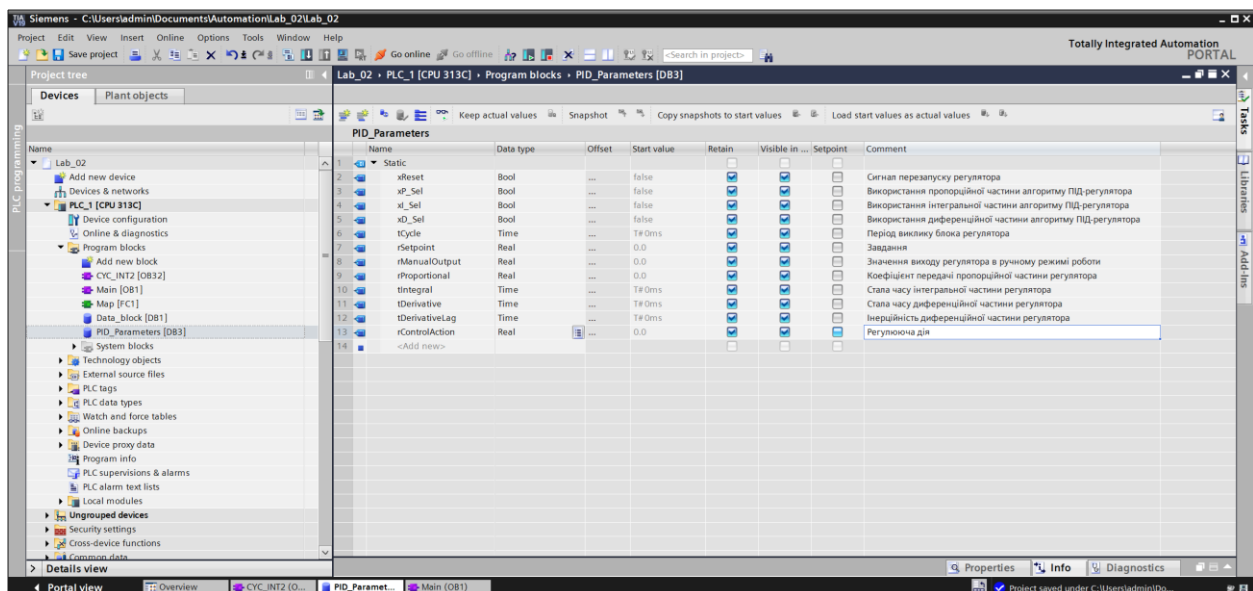


Рисунок 2.29 Оголошення змінних у блоці даних PID_Parameters (DB3).

Для того, щоб регулятор починав працювати з попередньо визначеними параметрами, їх потрібно записати у колонці **Start value**:

для параметру **xReset** встановимо значення за замовчуванням **False** (вимкнено);

для **xP_Sel** – **True** (ввімкнено);

для **xI_Sel** – **True**;

для **xD_Sel** - **False**;

tCycle встановлюємо рівним 100мс, для часових параметрів формат запису має вигляд **T#100ms**;

для параметру завдання **rSetpoint** встановлюємо значення за замовчуванням рівним **0.0**, варто звернути увагу на те, що дійсні значення записуються в TIA Portal у вигляді десяткового дробу;

rManualOutput встановлюємо за замовчуванням рівним **100.0** для того, щоб у ручному режимі насос працював за замовчуванням з максимальною продуктивністю;

rProportional встановлюємо рівним **2.0**;

tIntegral, який має тип **Time**, встановлюємо рівним 5 секунд, формат запису цього значення в **TIA Portal** має вигляд **T#5s**;

tDerivative встановимо рівним нулю, оскільки за замовчуванням диференціальна частина буде вимкнена, для цього записуємо значення **T#0ms**;

tDerivativeLag також встановлюємо рівним нулю **T#0ms**;

значення за замовчуванням **rControlAction** можна не встановлювати, оскільки воно буде обчислюватись за алгоритмом ПІД-регулятора.

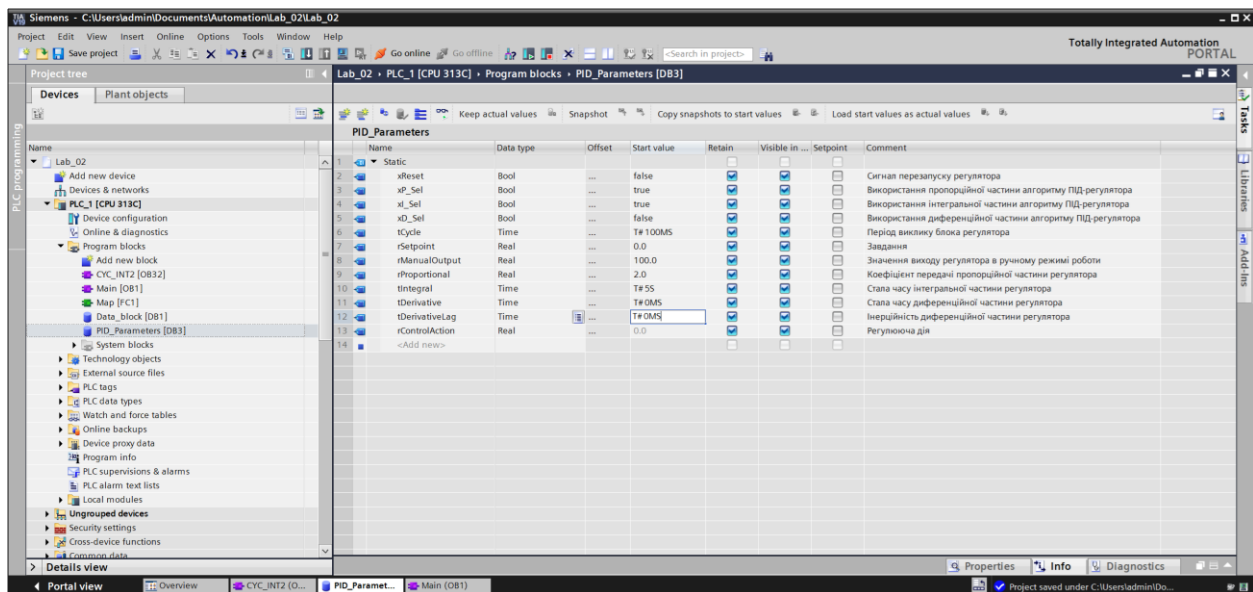


Рисунок 2.30 Визначення початкових значень параметрів.

Для того, щоб в процесі налаштування або роботи програми була можливість змінити значення параметрів без повторного запису програми на контролер, можна встановити для параметрів налаштування відповідної позначки у полі **Setpoint**. Встановимо такі позначки для параметрів **xP_Sel**, **xI_Sel**, **xD_Sel**,

rSetpoint, rManualOutput, rProportional, tIntegral, tDerivative та **tDerivativeLag**.

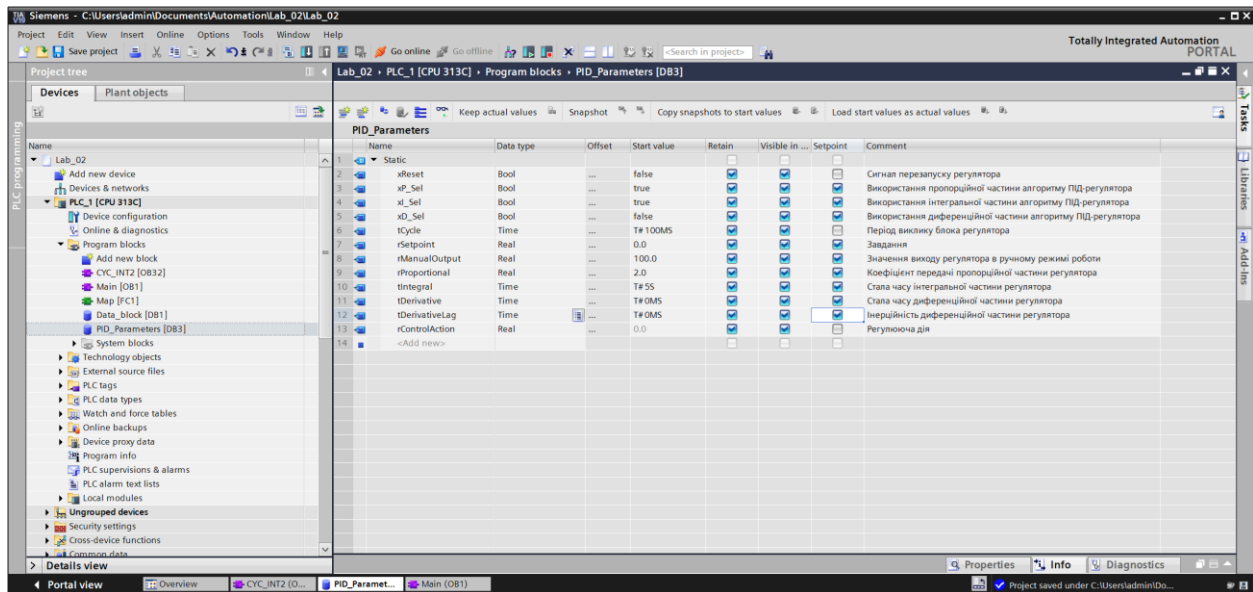


Рисунок 2.31 Встановлення можливості налаштування параметрів.

Після виконання всіх необхідних налаштувань встановлюємо відповідність параметрів входам та виходам регулятора **CONT_C**:

на вхід **EN** подаємо сигнал від нормально розімкнених контактів з параметрами **xAutoOn** та **xManualOn** блоку даних **Data_block**, які підключені паралельно за схемою «логічне АБО». Оскільки цей вхід дає дозвіл на запуск екземпляра блоку **PID_CP**, використання зазначеної схеми блокуватиме роботу блоку у випадку коли перемикач режиму роботи регулятора «Авто/0/Ручний» знаходиться у положенні «0»;

на вхід **COM_RST** подаємо сигнал від детектора переднього фронту параметра **xAutoON** блоку даних **Data_block**. Це дозволить ініціювати скидання значення виходу регулятора при переході в автоматичний режим роботи і таким чином перезапустить процес формування закону регулювання. Для коректної роботи детектору переднього фронту сигналу необхідна додаткова змінна, яка зберігатиме попередній стан сигналу **xAutoOn**, тож можемо використати для цього параметр **xReset** блоку даних **PID_Parameters**. Детектор переднього фронту сигналу можна знайти у розділі **Basic instructions** каталогу інструкцій;

на вхід **MAN_ON** подаємо сигнал від параметра **xManualOn** блоку даних **Data_block** для переведення регулятора в ручний режим роботи;

на вхід **P_SEL** подаємо сигнал від параметра **xP_Sel** блоку даних **PID_Parameters** для можливості відключення пропорційної частини ПІД-регулятора;

на вхід **I_SEL** подаємо сигнал від параметра **xI_Sel** блоку даних **PID_Parameters** для можливості відключення інтегральної частини ПІД-регулятора;

на вхід **D_SEL** подаємо сигнал від параметра **xD_Sel** блоку даних **PID_Parameters** для можливості підключення диференційної частини ПІД-регулятора;

на вхід **CYCLE** подаємо сигнал від параметра **tCycle** блоку даних **PID_Parameters**, в якому зазначається період виклику на виконання блоку регулятора;

на вхід **SP_INT** подаємо сигнал від параметра **rSetpoint** блоку даних **PID_Parameters** для можливості зміни заданого значення тиску в автоматичному режимі;

на вхід **PV_IN** подаємо сигнал від параметра **rPressureSensor** блоку даних **Data_block**, що містить масштабоване значення аналогового входу контролера, до якого підключений датчик тиску, що вимірює регульований параметр;

на вхід **MAN** подаємо сигнал від параметра **rManualOutput** блоку даних **PID_Parameters**, який містить значення виходу регулятора в ручному режимі роботи;

на вхід **GAIN** подаємо сигнал від параметра **rProportional** блоку даних **PID_Parameters**, який містить значення коефіцієнта підсилення пропорційної частини регулятора;

на вхід **TI** подаємо сигнал від параметра **tIntegral** блоку даних **PID_Parameters**, який містить значення сталої часу інтегральної частини регулятора;

на вхід **TD** подаємо сигнал від параметра **tDerivative** блоку даних **PID_Parameters**, який містить значення сталої часу диференційної частини регулятора;

на вхід **TM_LAG** подаємо сигнал від параметра **tDerivativeLag** блоку даних **PID_Parameters**, який містить значення сталої часу фільтра диференційної частини регулятора;

вихід **LMN** регулятора, який містить значення сигналу керування у дійсному форматі, передаємо параметру **rControlAction** блоку даних **PID_Parameters**. Після виконання цих налаштувань перший сегмент циклічного організаційного блоку **CYC_INT (OB32)** буде мати такий вигляд:

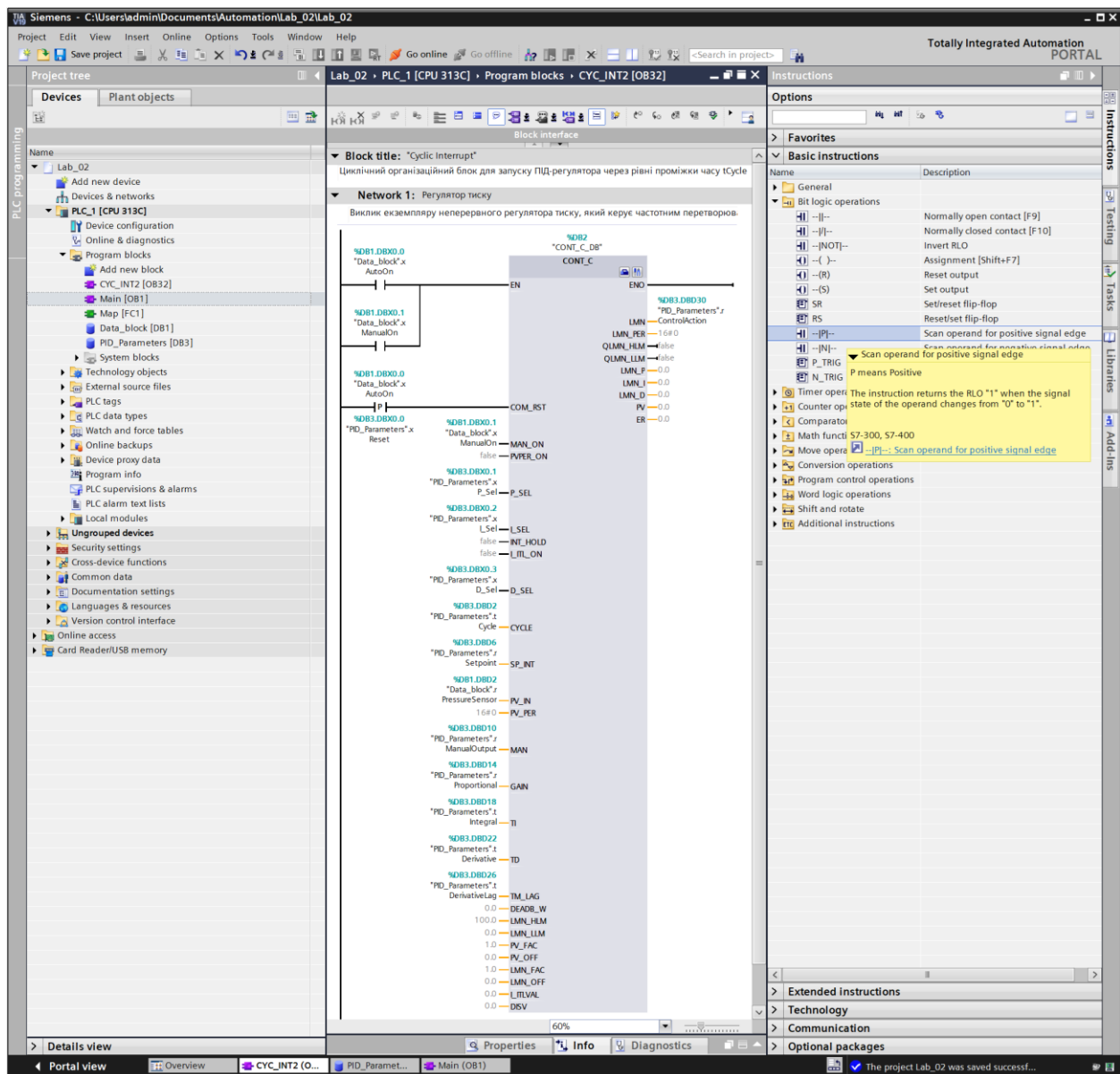


Рисунок 2.32 Налаштування регулятора тиску.

- Частотний перетворювач, який керує продуктивністю насоса, у даному алгоритмі отримує сигнал про потрібну продуктивність від ПЛК через аналоговий вихід. Як правило, цей сигнал уніфікований – **0..10В** або **0(4)..20мА**. Щоб ПЛК міг його сформувати у заданому діапазоні, алгоритм керування повинен передати розраховане регулятором в інженерних одиницях значення до цифро-аналогового перетворювача (ЦАП) у вигляді цілого числа, яке знаходиться в робочому діапазоні ЦАП. ЦАП сприймає передане йому число і перетворює його на фізичний електричний сигнал. Контролери лінійки **Siemens S7-300** використовують ЦАП, які можуть працювати в діапазоні перетворення **від 0 до 27648** для однополярних сигналів або в діапазоні **від -27648 до 27648** для двополярних сигналів незалежно від характеристик (розрядності) цифро-аналогового перетворювача. Регулятор **CONT_C** має два виходи, які формують

керуючу дію: на вихід **LMN** подається значення в інженерних одиницях (за замовчуванням – в діапазоні від **0.0** до **100.0**), на вихід **LMN_PER**, який має тип **WORD**, подається ціле значення, що дорівнює **LMN*27648/100**.

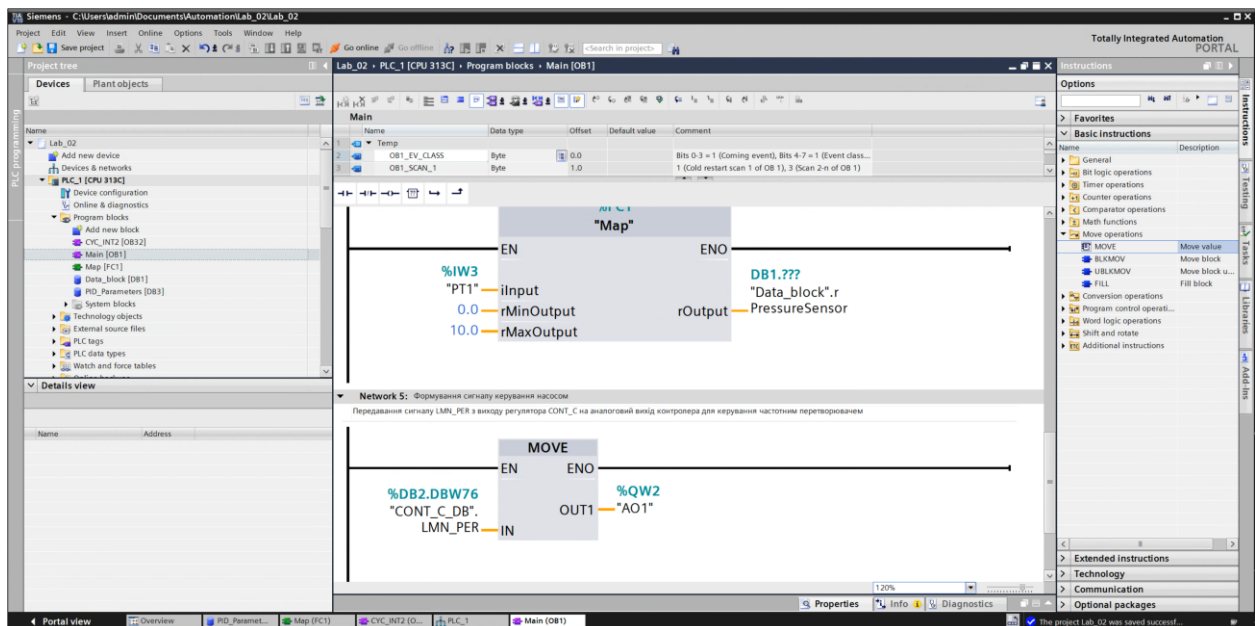


Рисунок 2.33 Передавання виходу LMN_PER регулятора CONT_C на аналоговий вихід ПЛК.

Для передачі виходу регулятора на аналоговий вихід контролера **AO1** можна використати команду **MOVE**, яка копіює значення з одної ділянки пам'яті контролера в іншу. В нашому випадку ми будемо копіювати значення виходу **LMN_PER** регулятора **CONT_C** у змінну, яка відповідає аналоговому виходу контролера. Для цього створюємо в організаційному блоці **Main (OB1)** нову секцію **Network 5**, в яку перетягуємо блок **MOVE**, який можна знайти у розділі **Move Operations** бібліотеки основних інструкцій у правій частині вікна **TIA Portal** чи ввести назву блока **MOVE** у пустий елемент, який можна додати через пункт **Insert empty box** контекстного меню секції.

Перевіримо розроблений алгоритм. Для цього потрібно обрати в дереві проекту контролер та запустити симулятор ПЛК за допомогою меню **TIA Portal Online** → **Simulation** → **Start**

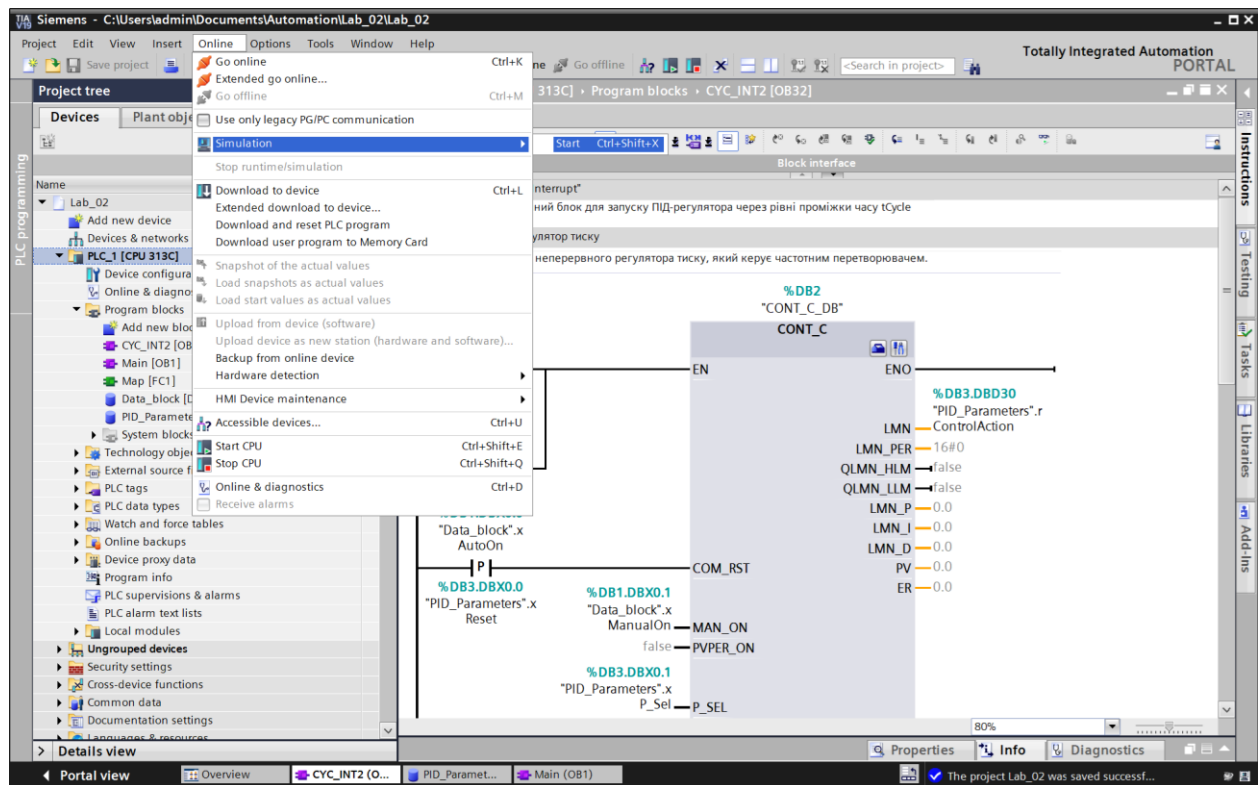


Рисунок 2.34 Запуск симулятора ПЛК.

Налаштовуємо параметри симулятора потрібно додати вікна контролю вхідних та вихідних сигналів контролера, які задіяні в алгоритмі. Перелік сигналів, які використовує розроблена програма, можна переглянути у таблиці тегів Default tag table, яку можна знайти у розділі PLC tags дерева проєкту. За таблицею можна визначити, що адреса аналогового входу, який використовується для підключення датчика тиску та відповідає змінній PT1, починається з третього вхідного байта, оскільки має адресу %IW3, а адреса аналогового виходу AO1, який керує частотним перетворювачем, починається з другого вихідного байта, оскільки значення змінної AO1 зберігається за адресою %QW2. Дискретні вхідні сигнали записуються до вхідного байта з адресою %I0, а вихідні – до вихідного байта %Q0.

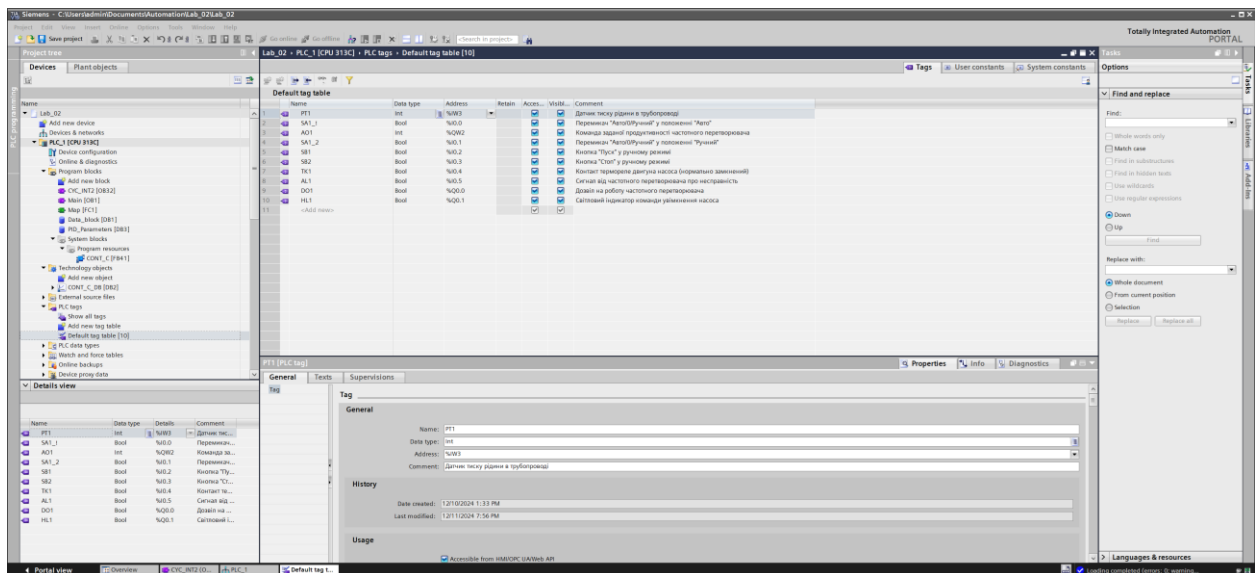


Рисунок 2.35 Таблиця тегів Default tag table.

Вводимо зазначені адреси у відповідних вікнах симулятора для можливості їх контролю та зміни у процесі перевірки правильності функціонування алгоритму.

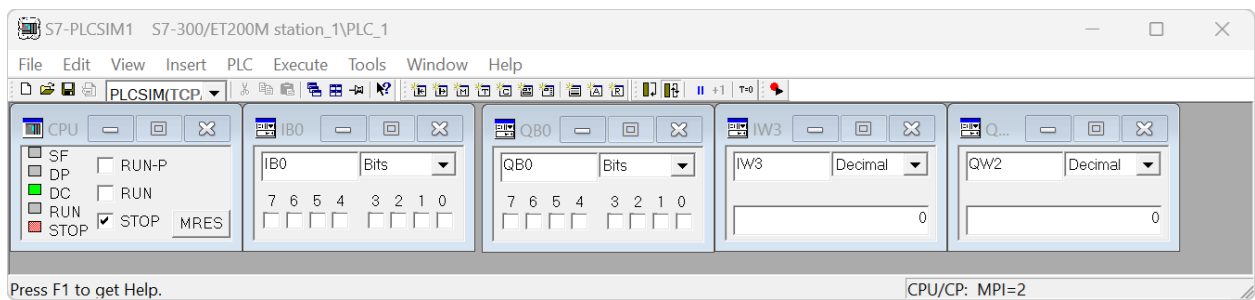


Рисунок 2.36 Відображення сигналів контролера в симуляторі.

Завантажуємо проєкт на симулятор контролера.

Якщо при створенні програми не було зроблено помилок, які заважають компіляції та завантаженню програми на контролер, напроти кожного елемента проєкту має бути синя позначка. Тоді можна завантажити проєкт на симулятор.

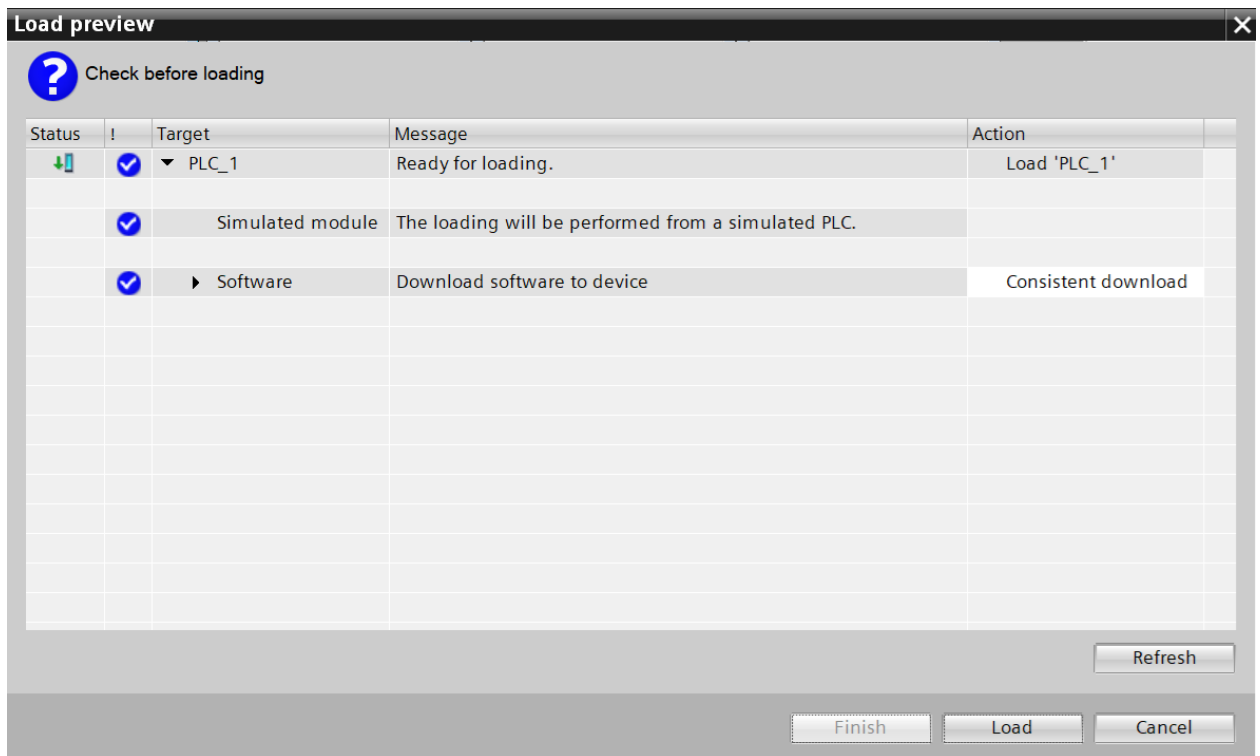


Рисунок 2.37 Вікно завантаження програми за умови відсутності помилок компіляції.

Переводимо симулятор у режим **RUN-P**, який дозволяє вносити зміни під час роботи без необхідності зупинки контролера.

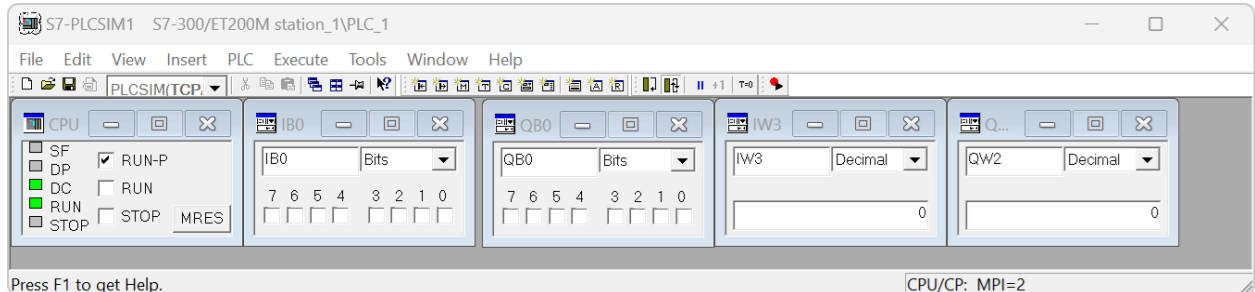


Рисунок 2.38 Переведення симулятора у режим роботи RUN-P.

Перевіряємо роботу алгоритму в ручному режимі.

Вмикаємо ручний режим сигналом **SA1_2 (%I0.1)**, імітуємо відсутність сигналу від перегріву двигуна сигналом **TK1 (%I0.4)** та подаємо команду пуску насоса сигналом **SB1 (%I0.2)**. При цьому має увімкнутись дискретний вихід контролера **DO1 (%Q0.0)** та світловий індикатор роботи насоса **HL1 (%Q0.1)**.

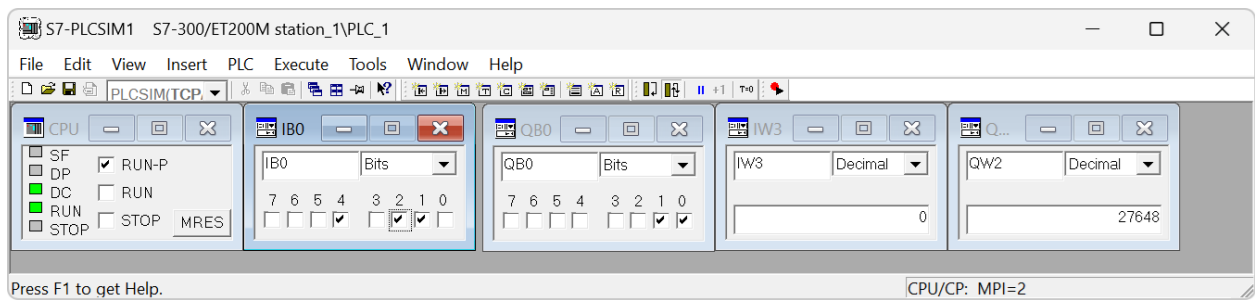


Рисунок 2.39 Перевірка алгоритму за допомогою вікна симулятора.

Оскільки значення продуктивності насосу в ручному режимі встановлено рівним максимальному значенню виходу регулятора (значення **rManualOutput** дорівнює **100.0**), на аналоговий вихід контролера подається значення, яке відповідає його максимуму, тобто **27648**.

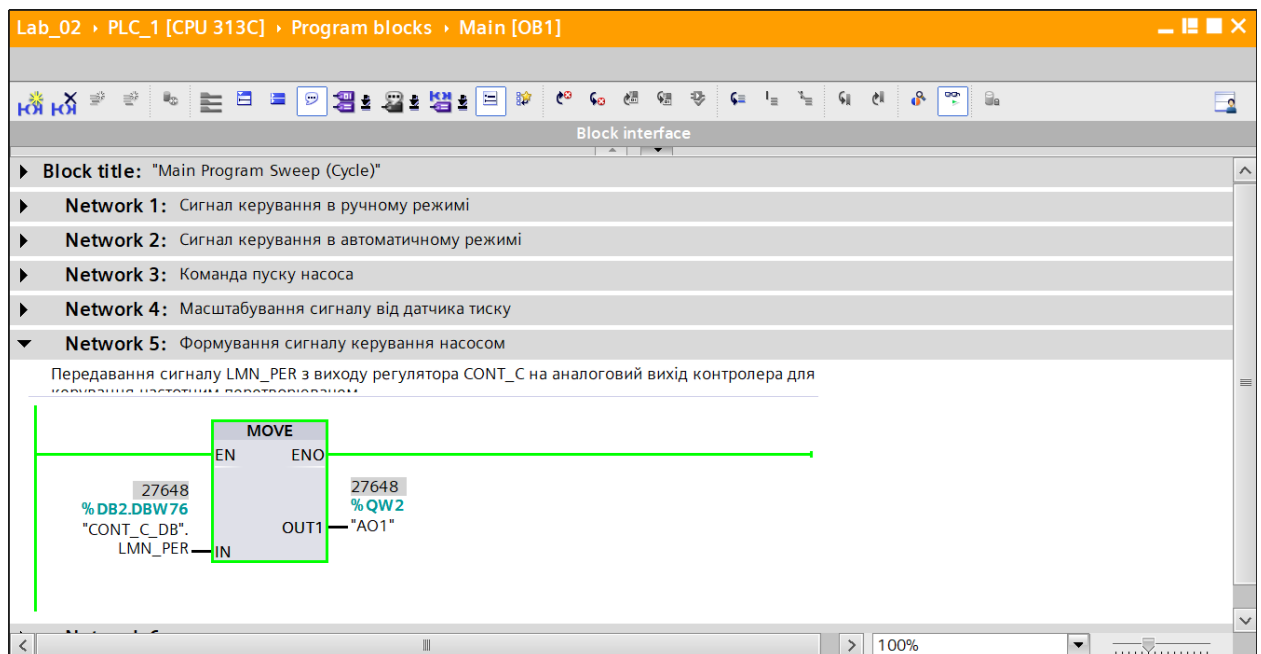


Рисунок 2.40 Перевірка формування сигналу керування частотним перетворювачем в режимі моніторингу.

Результат ввімкнення дискретних виходів контролера можна бачити як за допомогою вікна симулятора у вікні, що відображає стан вихідного байта, що відповідає даному виходу, так і за допомогою вікна редактора організаційного блоку Main (OB1) у режимі моніторингу.

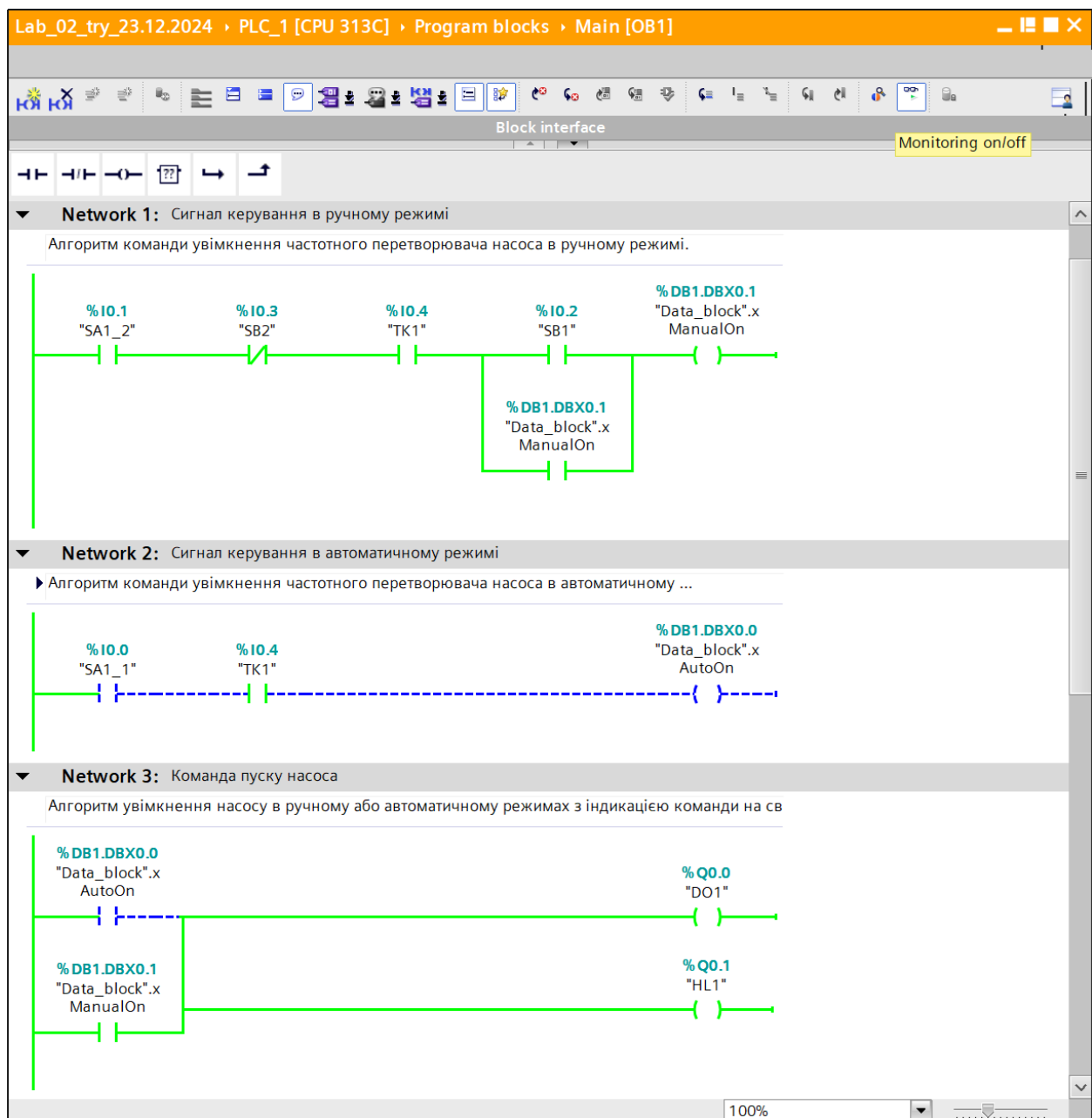


Рисунок 2.41 Перевірка правильності функціонування алгоритму ручного керування у режимі моніторингу.

При вимкненні **SB1** (**%I0.2**) насос має продовжувати працювати до вимкнення за допомогою кнопки **SB2** (**%I0.3**) або до спрацювання захисту двигуна насоса від перегріву, тобто вимкнення **TK1** (**%I0.4**).

Наступним кроком перевіримо роботу алгоритму в автоматичному режимі роботи.

Вмикаємо автоматичний режим сигналом **SA1_1** (**%I0.0**) та сигналом **TK1** (**%I0.4**). При цьому має ввімкнутись дискретний вихід контролера **DO1** (**%Q0.0**) та світловий індикатор роботи насоса **HL1**(**%Q0.0**).

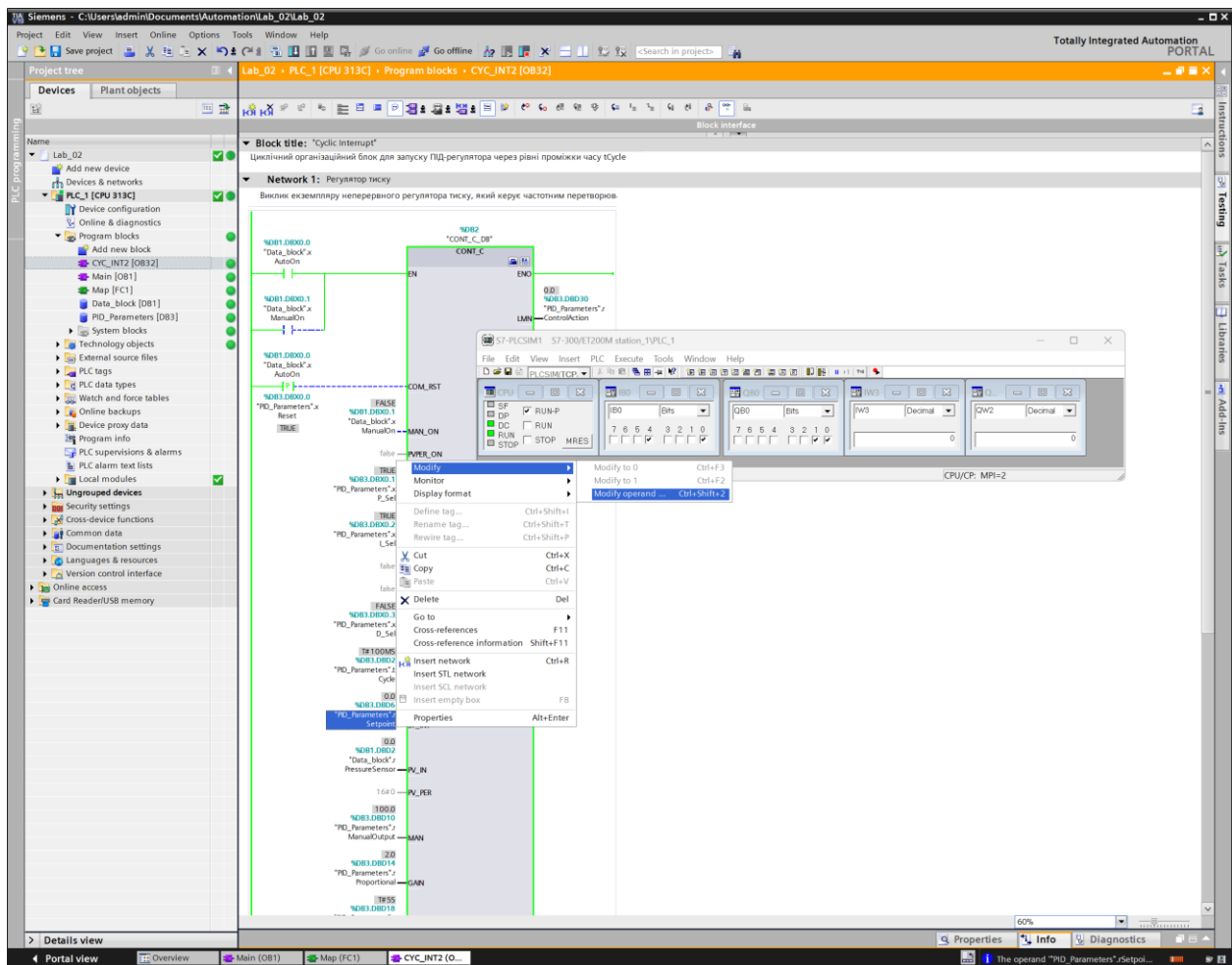


Рисунок 2.42 Зміна завдання регулятора тиску через контекстне меню в режимі моніторингу.

Швидість роботи насоса в автоматичному режимі визначатиметься різницею заданого значення тиску, що визначається параметром **rSetpoint** блока **PID_Parameters** та фактичним значенням тиску, який контролер отримує з першого аналогового входу і після масштабування записує у параметр **rPressureSensor** блока **Data_block**. Якщо значення **rPressureSensor** менше за значення **rSetpoint** - регулятор намагатиметься його компенсувати збільшенням сигналу керування, який записується у параметр **rControlAction** блока **PID_Parameters**, якщо ж задане значення тиску менше фактичного, керуюча дія буде зменшуватись. При цьому керуюча дія **rControlAction** має залишатись у межах, які визначені значеннями вхідних параметрів регулятора **LMN_LLM** та **LMN_HLM**.

Змінити завдання можна за допомогою контекстного меню параметра **rSetpoint** в режимі моніторингу. При цьому відкривається вікно зміни параметра, в якому потрібно ввести нове значення:

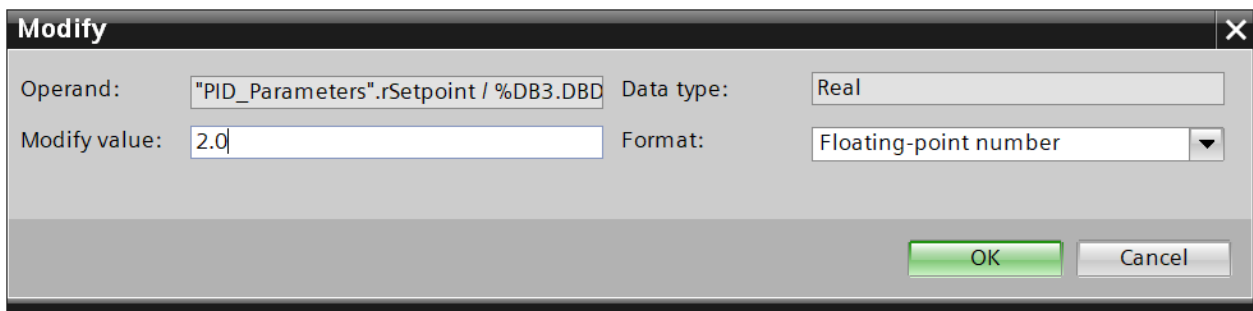


Рисунок 2.43 Вікно зміни параметра *rSetpoint* блоку даних *PID_Parameters*.

Перевіряємо, що збільшення завдання призводить до поступового зростання виходу регулятора, який подається на аналоговий вихід контролера, до максимального значення **27648**:

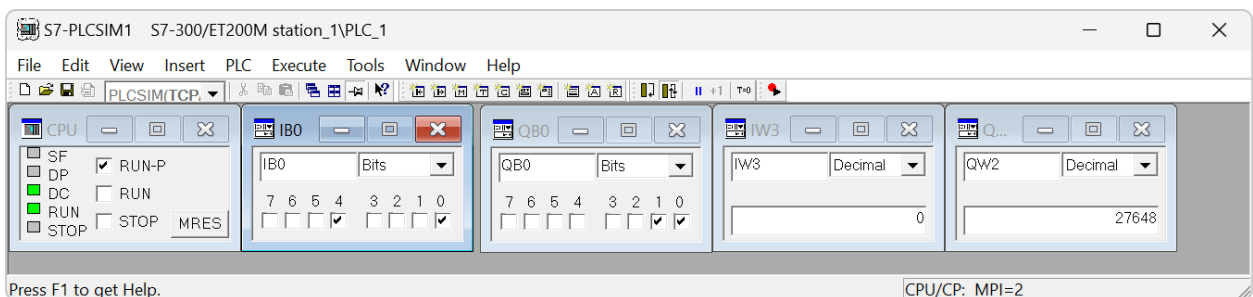


Рисунок 2.44 Контроль зростання виходу регулятора в автоматичному режимі.

Якщо встановити значення аналогового входу регулятора таким, що є більшим за задане, значення виходу регулятора має поступово зменшуватись до мінімального значення **0**:

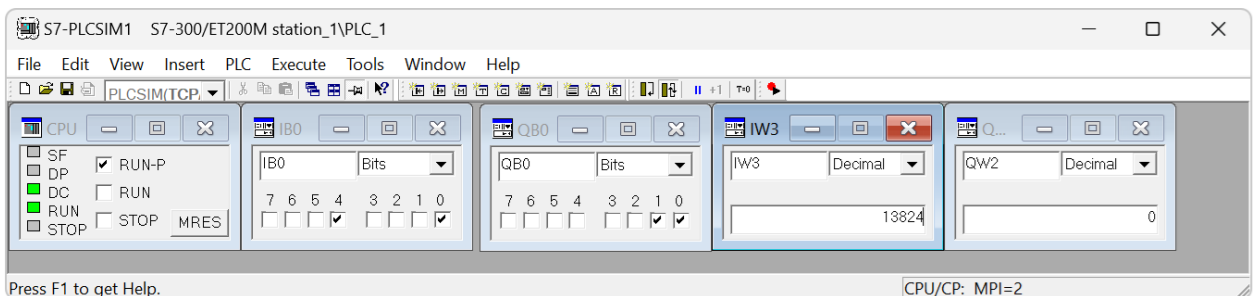


Рисунок 2.45 Контроль зменшення виходу регулятора в автоматичному режимі.

Спрацювання теплового захисту двигуна насоса має вимикати дискретні виходи контролера **DO1 (%Q0.0)** та **HL1 (%Q0.1)**. В такому випадку навіть за наявності на аналоговому вході частотного перетворювача ненульового сигналу частотний перетворювач не подаватиме напругу на двигун насоса.

2.4 Контрольні завдання

1. Додати в релейно контактну схему світловий індикатор HL2, який вмикається у разі перегріву двигуна насоса.
2. Модифікувати програму таким чином, щоб вихід значення тиску рідини в трубопроводі за допустимий робочий діапазон відображався увімкненням додаткового світлового індикатора HL3.
3. Проаналізувати вплив параметрів настройки регулятора Gain, TI, TD, TM_LAG на швидкість зміни виходу регулятора.

3 Лабораторна робота №3.

Мета роботи: отримати навички ефективної розробки програмного забезпечення, використання шаблонів та бібліотек в TIA Portal.

3.1 Завдання

1. Створення архіву проєкта
2. Створення документації проєкта
3. Створення бібліотеки розроблених програмних компонентів
4. Заміна процесорного модуля у проєкті
5. Створення архіву бібліотеки

3.2 Порядок виконання роботи

3.2.1 Створення архіву проєкта

Для третьої роботи збережемо проєкт другої лабораторної роботи з новою назвою. Для цього створюємо засобами операційної системи нову директорію з назвою Lab_03 та за допомогою пункту меню TIA Portal **Project** → **Save as...** відкриваємо вікно збереження проєкту за новим шляхом, де потрібно вказати створену нову директорію Lab_03 та назву для нового проєкту Lab_03.

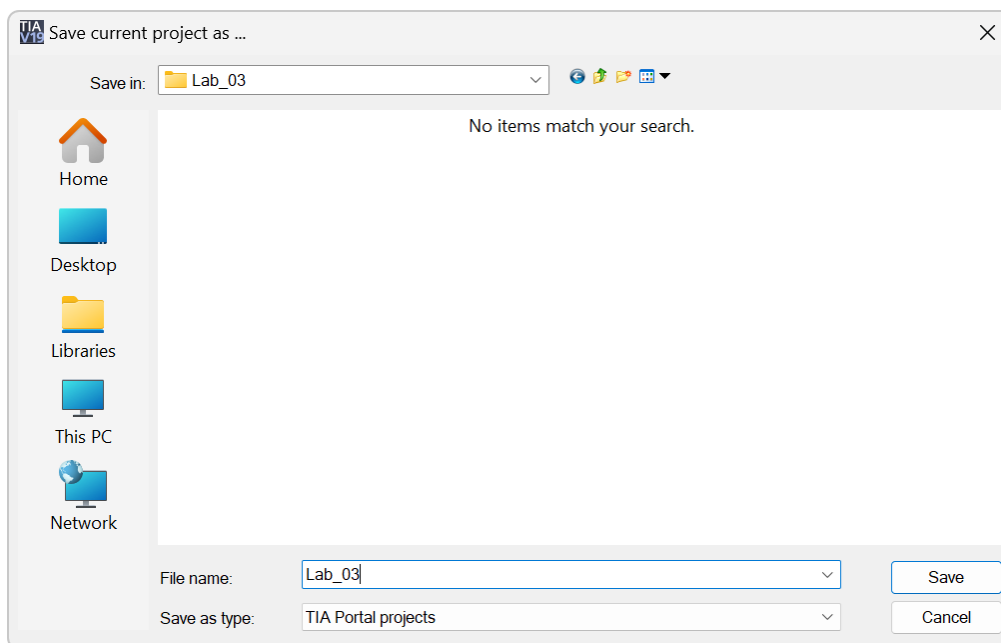


Рисунок 3.1 Збереження проєкту з новою назвою.

Перевіряємо вміст директорії, в якій збережений новий проєкт.

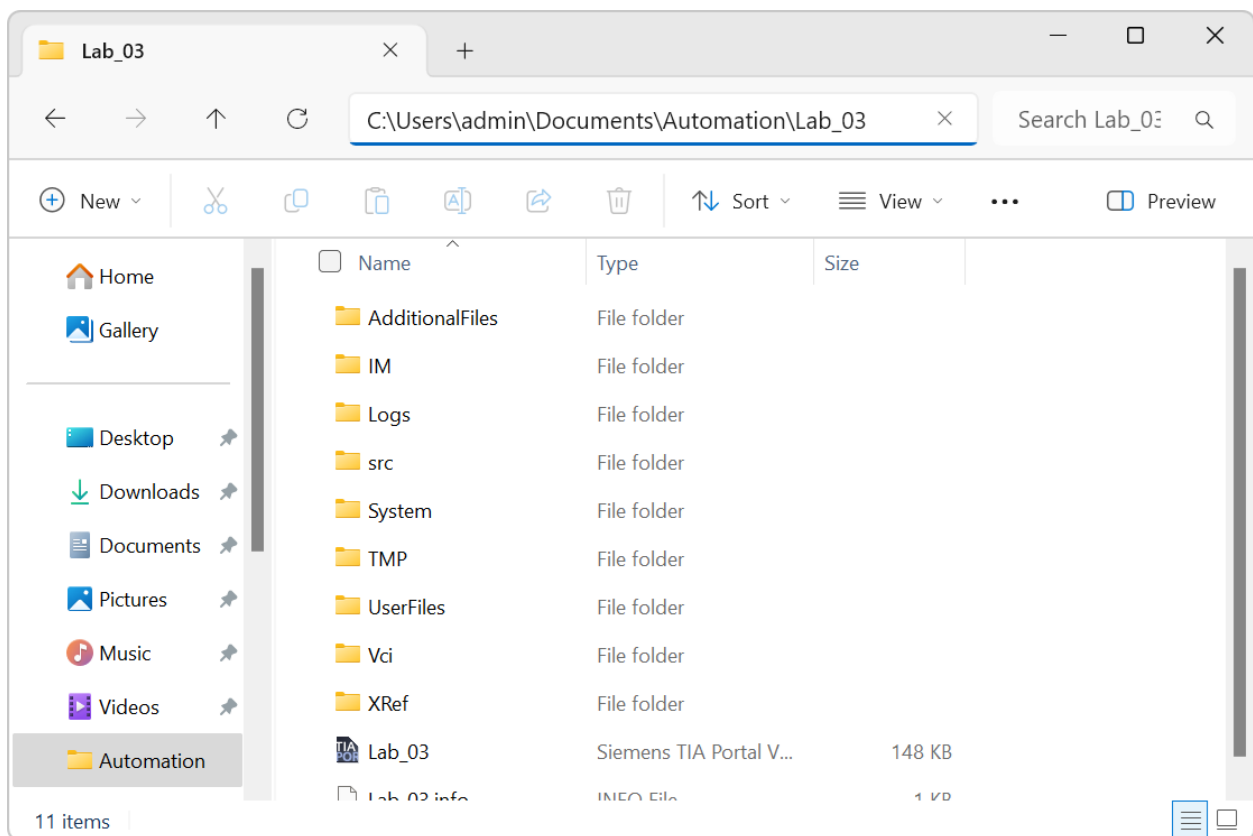


Рисунок 3.2 Вміст директорії проєкта TIA Portal.

Всередині директорії проєкта знаходяться декілька десятків файлів та більше десятка вкладених директорій. Для створення резервних копій проєктів або для переносу чи обміну проєктами такий формат збереження може бути незручним. Значно ефективніше використовувати для цього архіви, які дозволяють зберігати проєкти у окремих файлів, які окрім зменшення місця на диску також дають можливість краще організувати процес резервного копіювання робочих файлів. Створити архів проєкту можна за допомогою пункту меню TIA Portal **Project → Archive...**

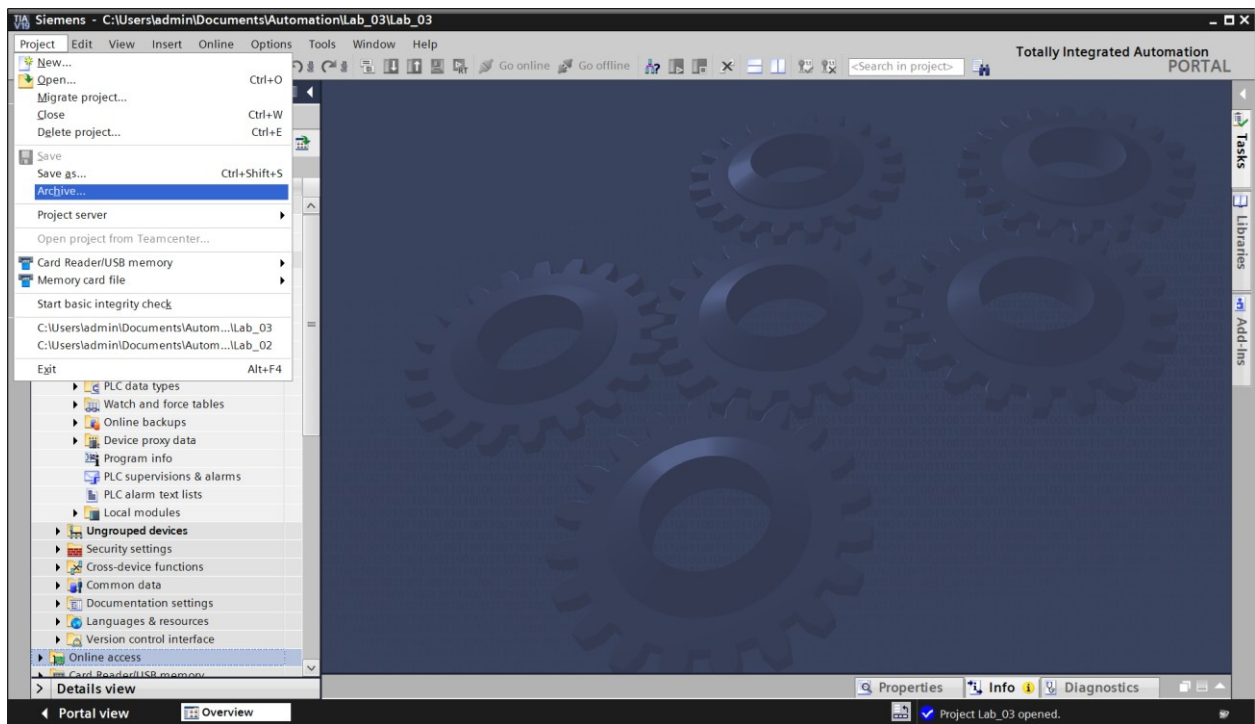


Рисунок 3.3 Створення архіву проєкту.

Результатом виконання цієї команди буде відкриття вікна створення архіву, в якому можна за необхідності змінити параметри створення архіву. Архів, який створює TIA Portal, має розширення **.zap19** для версії **TIA Portal v19**, попередні версії архівів мають розширення, що містять їх порядкові номери та, по суті, є архівом вмісту директорії проєкту у **ZIP** форматі. Відкрити створений раніше файл архіву можна через меню TIA Portal **Project** → **Open...**, у старих версій TIA Portal для відкриття архіву використовується пункт меню **Project** → **Retrieve**.

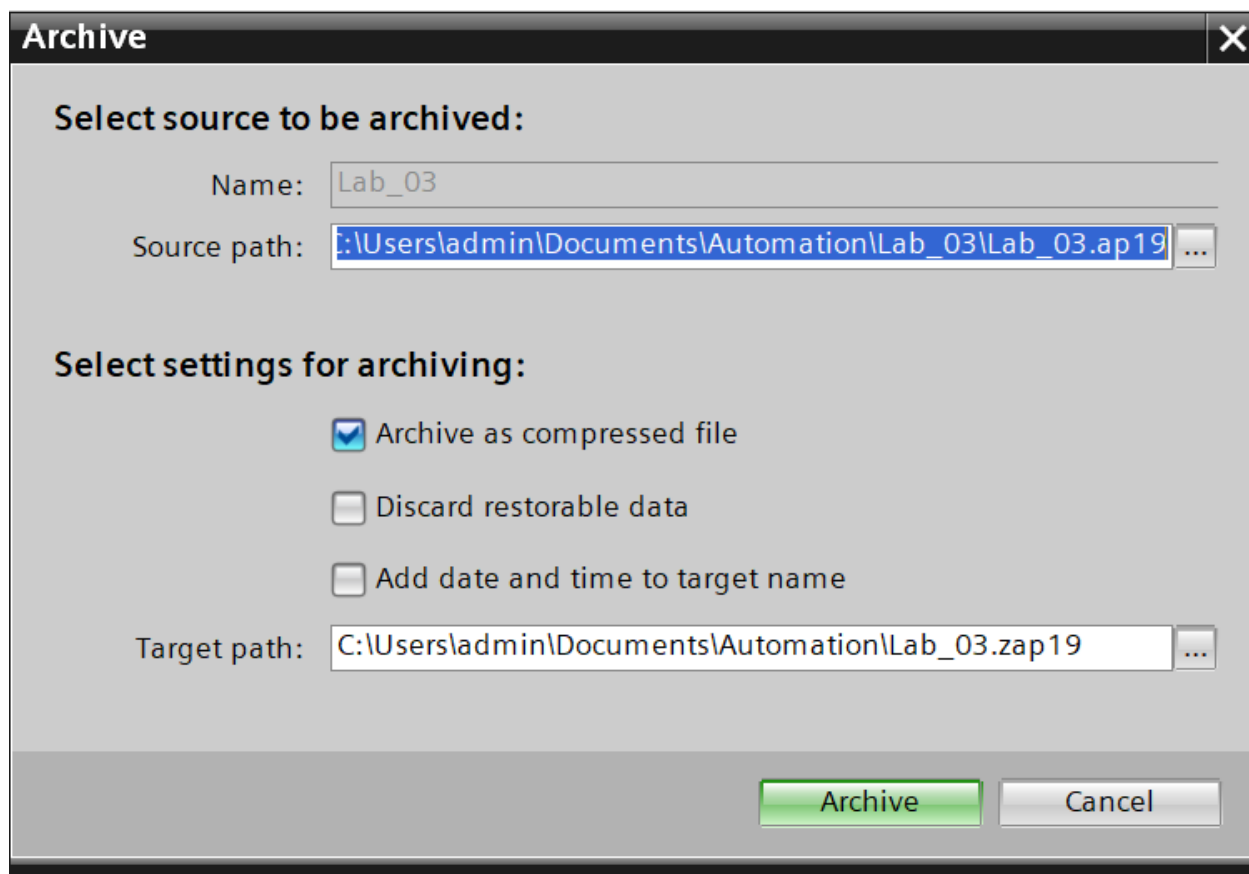


Рисунок 3.4 Вікно створення архіву проєкта.

3.2.2 Створення документації проєкта

Одним з найбільш відповідальних етапів роботи над проєктом, який вимагає значних ресурсів, є розробка комплекту документації проєкта. TIA Portal дає можливість значно спростити цю задачу за рахунок автоматичного генерування комплекту документації. Важливою вимогою для отримання якісної документації є відповідальне ставлення розробника проєкта до його опису у вигляді коментарів коду, пояснень призначення параметрів та змінних, зрозумілих назв змінних та позначень входів та виходів контролера. Якщо протягом роботи над проєктом розробник неухильно дотримувався вимог документування, весь комплект документації проєкту TIA Portal можна отримати протягом лічених хвилин за допомогою пункту меню **Project → Print preview...** Цей пункт стає доступним якщо попередньо обрати у дереві проєкту елемент, документацію якого потрібно створити: весь проєкт, окремий контролер (їх у проєкті може бути декілька) або інший елемент проєкту.

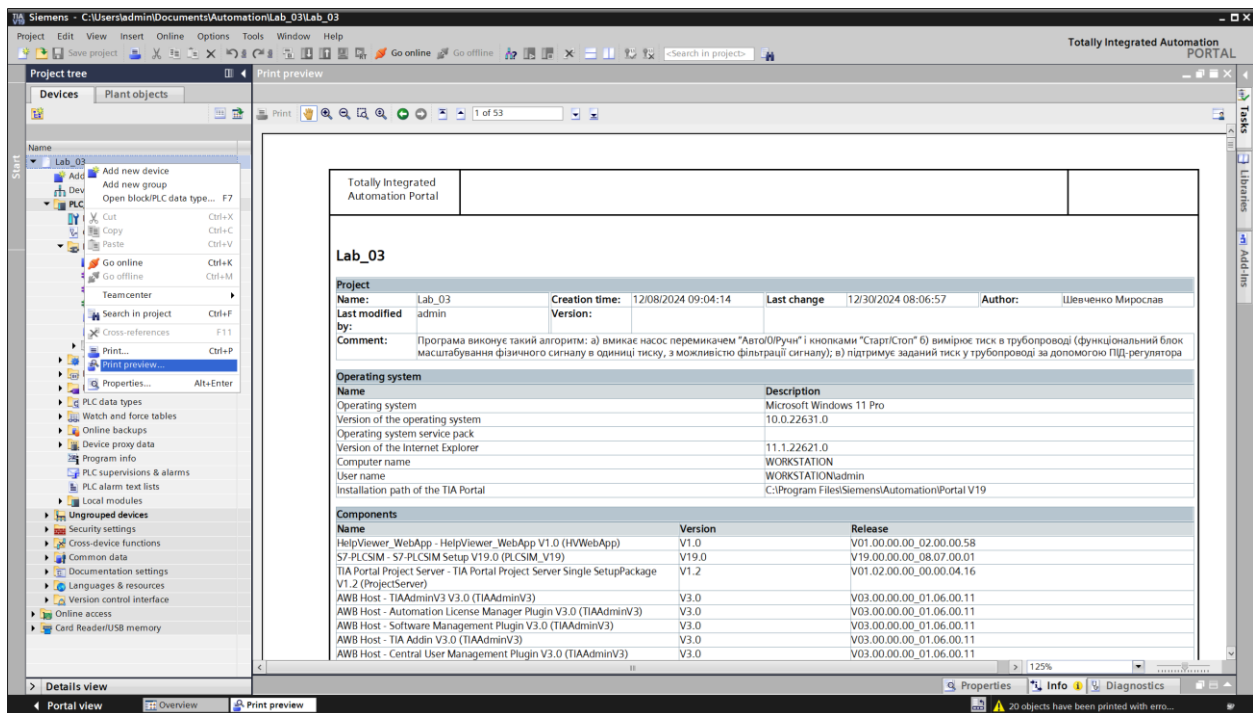


Рисунок 3.5 Створення документації проєкта.

Після створення документації її можна надрукувати або зберегти у PDF файлі, який можна буде використовувати для перегляду документації на інших пристроях.

3.2.3 Створення бібліотеки розроблених програмних компонентів

Досить часто алгоритми керування для типових систем мають незначні відмінності, що зазвичай пов'язані з розробкою додаткових функцій або внесення незначних змін до розроблених раніше функцій. Зрозуміло, що за таких умов розробка проєкту з самого початку є нераціональною витратою ресурсів. Для прискорення процесу розробки нових проєктів доцільно зберігати типові алгоритми або частини проєктів у доступному для повторного повного або часткового використання. TIA Portal пропонує для цього використовувати бібліотеки користувача, де можна зберігати елементи проєктів, які часто використовуються.

Для створення нової бібліотеки можна скористатись меню TIA Portal **Options** → **Global libraries** → **Create new library...** або натиснути кнопку **Create new global library** на панелі інструментів розділу **Global libraries** бібліотек TIA Portal.

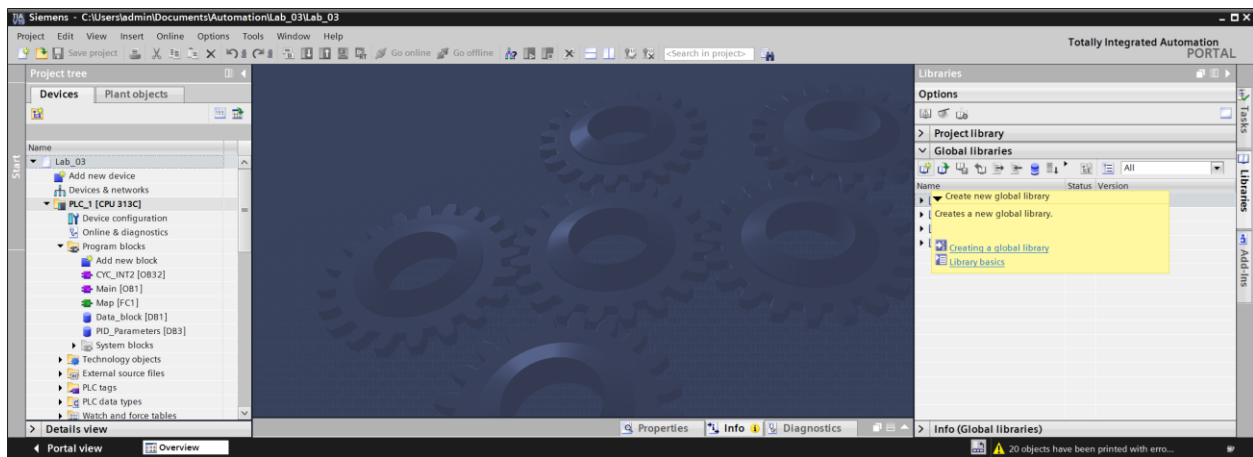


Рисунок 3.6 Створення нової бібліотеки.

У вікні створення нової бібліотеки потрібно вказати її назву та шлях, за яким вона буде зберігатись.

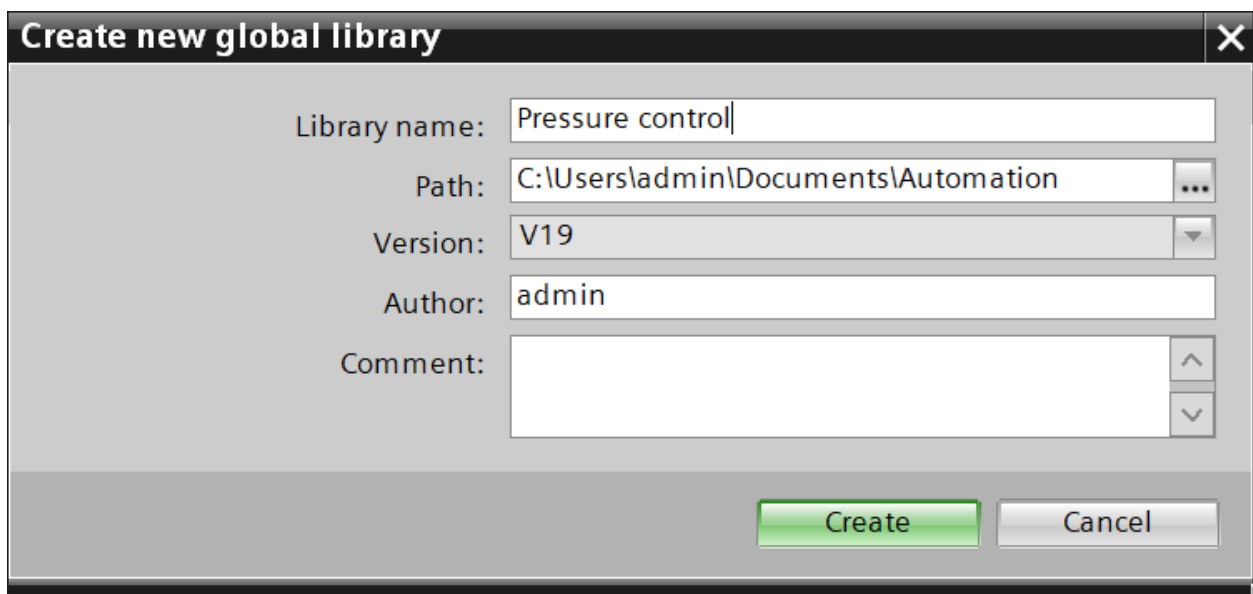


Рисунок 3.7 Вікно створення нової бібліотеки

Перенесемо до бібліотеки розроблену функцію масштабування Map. Для цього можемо виділити її у дереві проєкту та скопіювати до буферу обміну за допомогою комбінації гарячих клавіш **Ctrl+C** або пункту **Copy** контекстного меню, а потім вставити у розділ **Master copies** створеної бібліотеки за допомогою комбінації гарячих клавіш **Ctrl+V** чи пункту **Paste** контекстного меню. Також можна перетягнути функцію мишею з дерева проєкту до розділу **Master Copies** бібліотеки. Після внесення змін до бібліотеки її потрібно зберегти за допомогою пункту **Save library** контекстного меню бібліотеки або кнопки панелі інструментів бібліотеки **Save the changes to the global library**.

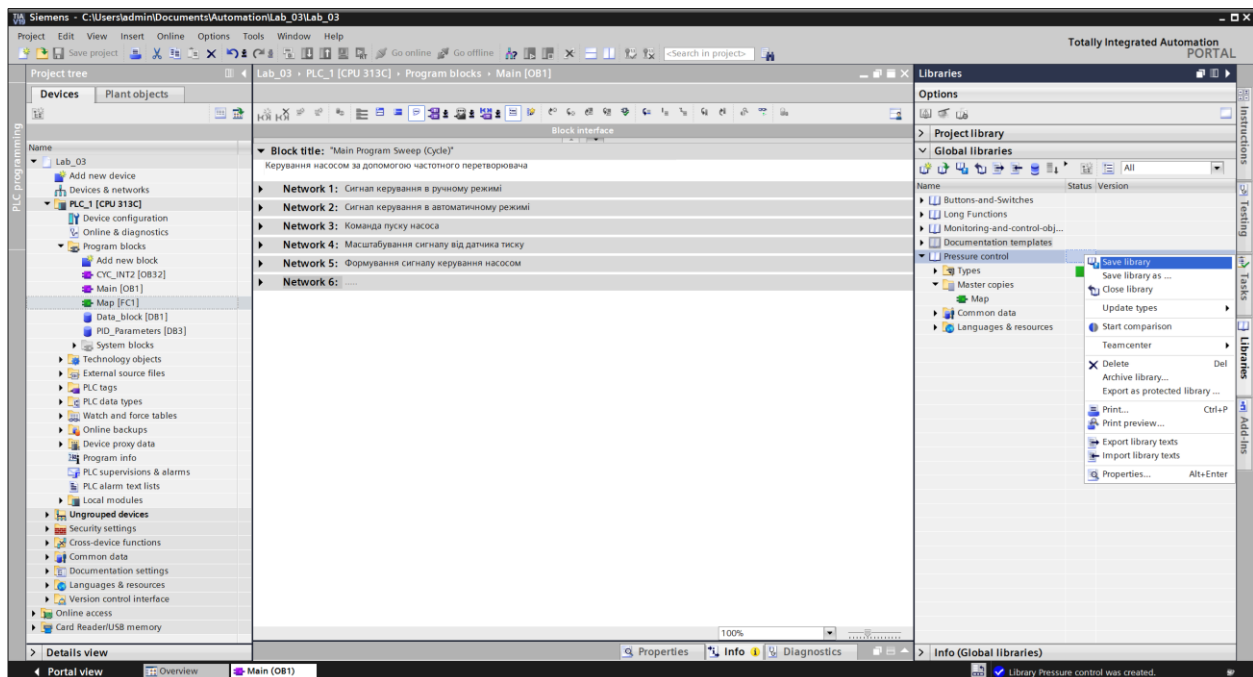


Рисунок 3.8 Збереження внесених до бібліотеки змін.

Бібліотека може містити не тільки окремі програмні компоненти, але й цілі конфігурації контролерів включно з розробленими для них алгоритмами керування. Зробимо копію контролера **PLC_1** у створеній бібліотеці **Pressure control** під назвою **Pressure PLC S7 313C** для використання у майбутніх проєктах та збережемо зміни у бібліотеці.

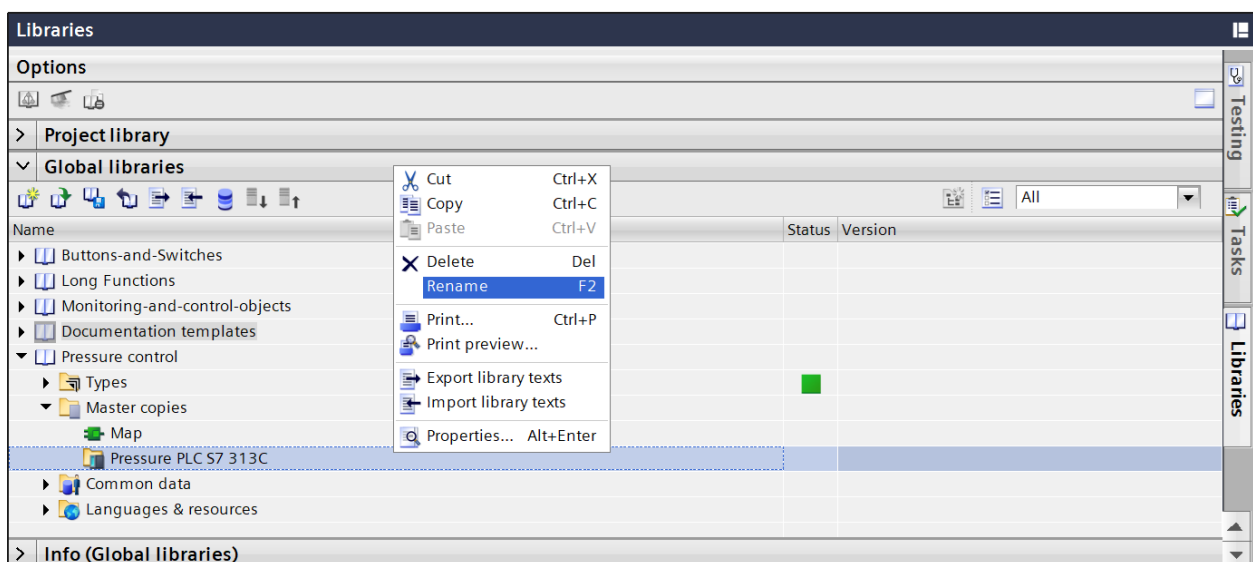


Рисунок 3.9 Перейменування контролера в бібліотеці.

Одним із способів скорочення часу, необхідного для розробки програмного забезпечення систем керування є використання готових шаблонів конфігурацій, які дають можливість обрати для розробки нового проєкту оптимальну за вартістю та технічними характеристиками модель контролера. Так, наприклад, у проєктах, де необхідно забезпечити мінімальну вартість системи, недоцільно

використовувати модулі з додатковими інтерфейсами чи з надмірною продуктивністю, яка впливає на його вартість. Натомість, у проєктах, які передбачають використання супервізорного керування, інтеграцію з іншими системами або подальше масштабування системи, необхідно використовувати моделі контролерів з відповідними технічними можливостями.

Виконаємо конфігурування контролера з можливістю мережної взаємодії для подальшого використання цієї конфігурації у проєктах. Для цього додаємо у поточний проєкт новий процесорний модуль серії **SIMATIC S7-300** з вбудованим інтерфейсом **PROFINET CPU 314C-2PN/DP**, який позначений номером **6ES7 314-6EH04-0AB0**.

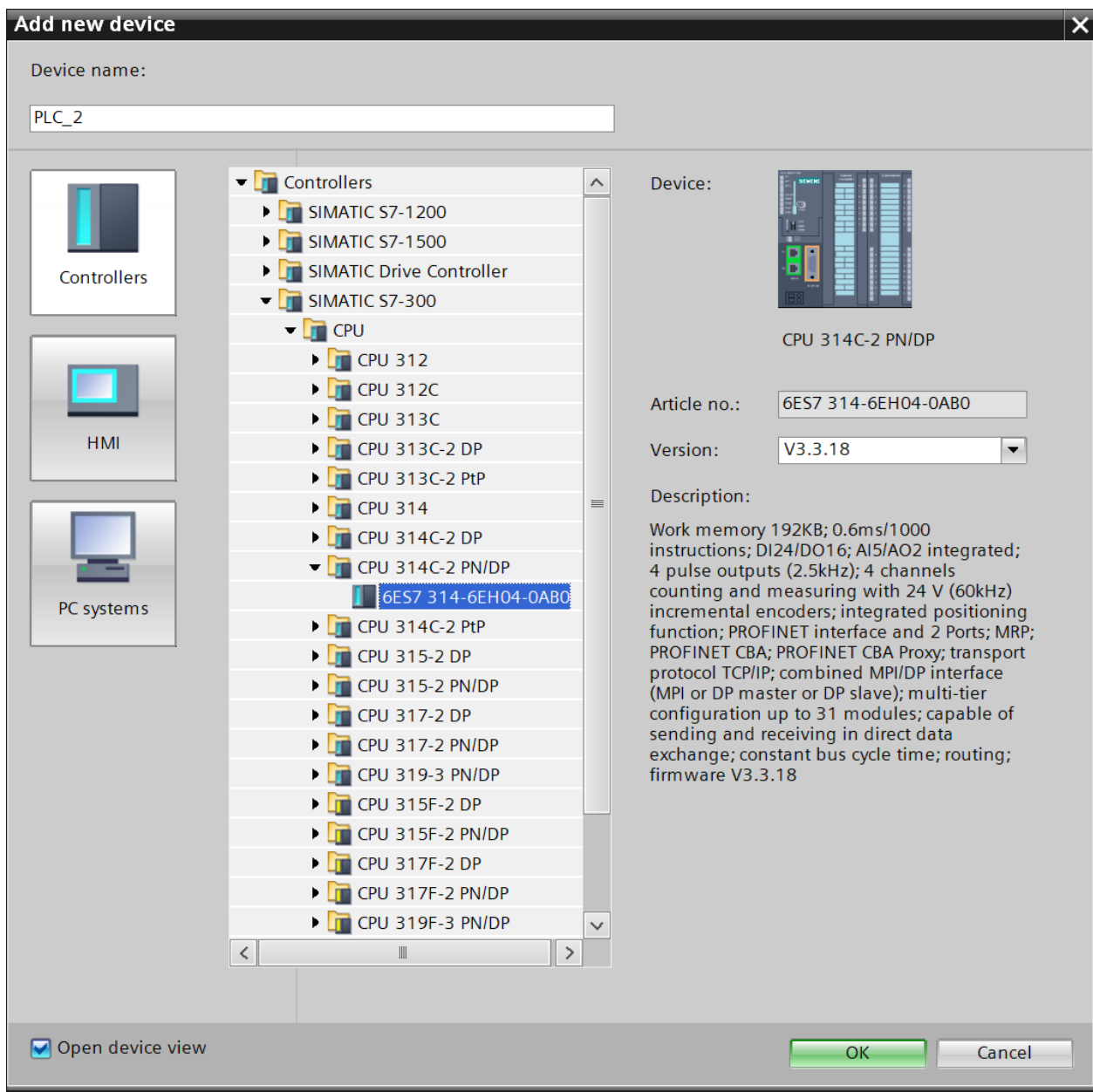


Рисунок 3.10 Додавання у проєкт контролера з інтерфейсом PROFINET.

Першочергово налаштуємо мережну IP (Internet Protocol) адресу контролера, яка окрім обміну даними з іншими пристроями системи керування може використовуватись також для конфігурування, дистанційного завантаження алгоритму на контролер або контролю правильності виконання алгоритму на контролері за допомогою TIA Portal. Якщо контролер буде підключений до локальної мережі підприємства, його IP-адреса та інші мережні налаштування (маска мережі, адреса маршрутизатора) повинні бути узгоджені з адміністратором локальної мережі. Для даного контролера вказуємо IP адресу **192.168.0.2** та маску підмережі **255.255.255.0**.

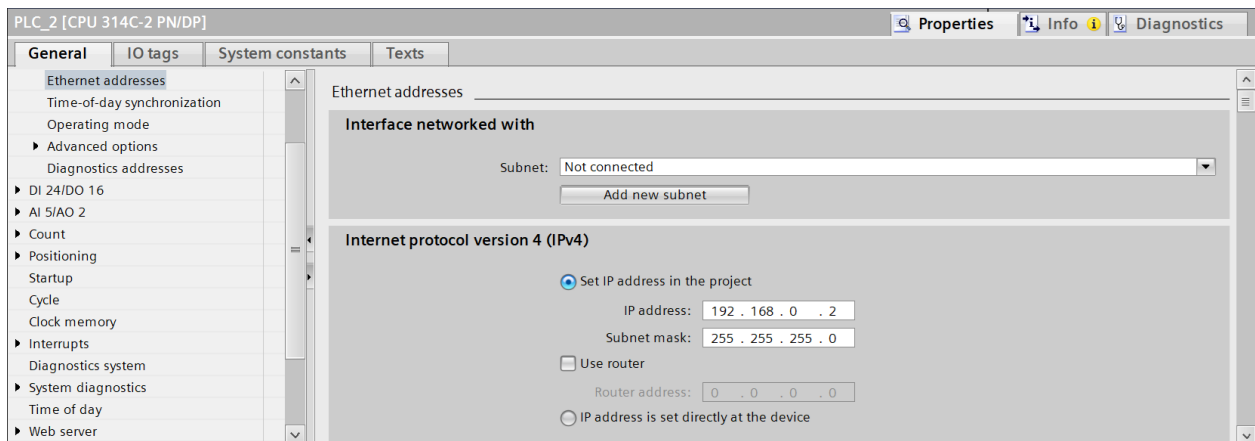


Рисунок 3.11 Конфігурування IP адреси контролера.

Для спрощення процесу розробки відключаємо у властивостях захист контролера на час написання алгоритму. Це дозволить завантажувати програму на контролер без необхідності вводити пароль.

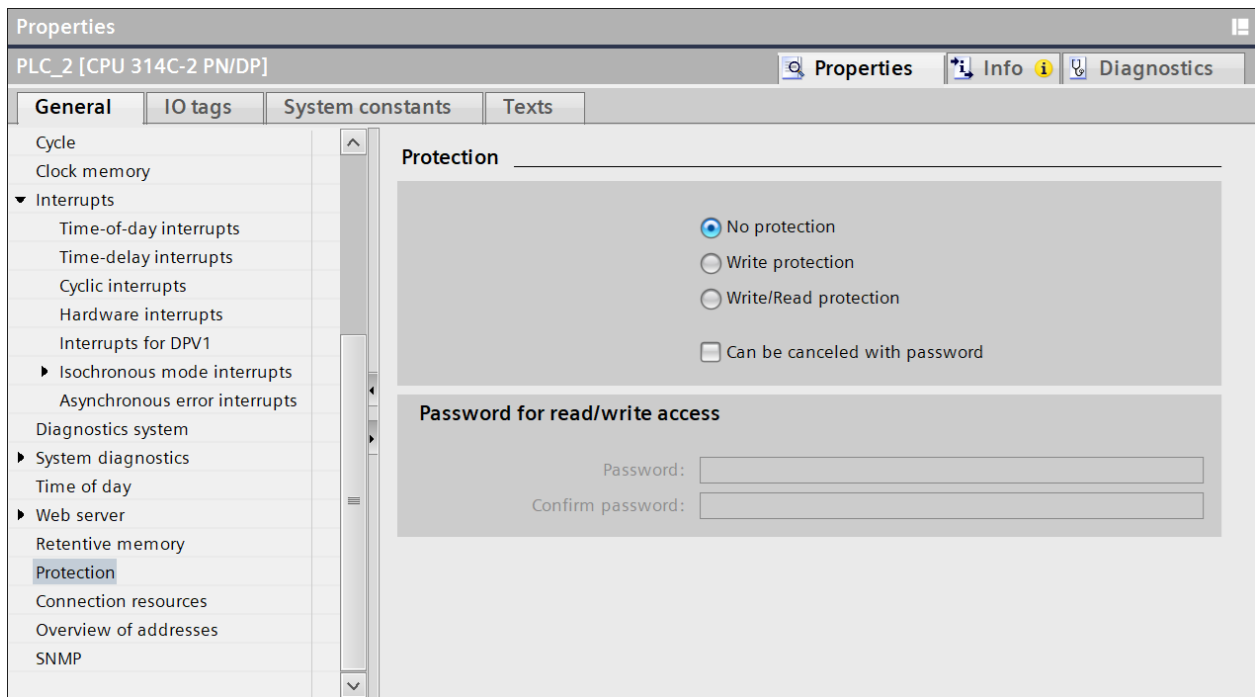


Рисунок 3.12 Відключення парольного захисту контролера.

Також вводимо нові адреси вбудованих дискретних входів та виходів контролера за прикладом, наведеним нижче.

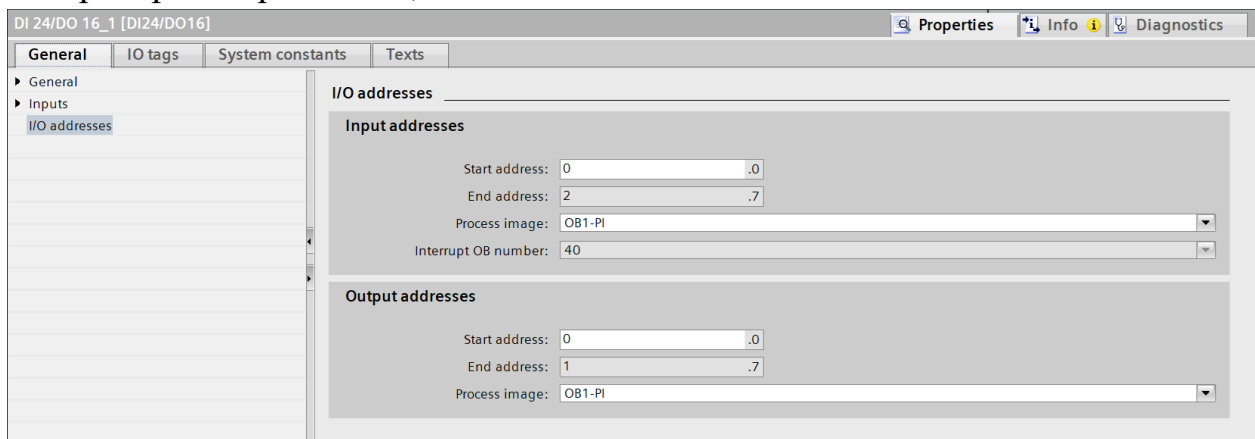


Рисунок 3.13 Конфігурування адрес дискретних входів та виходів контролера.

Конфігуруємо адреси аналогових входів та виходів процесорного модуля за зразком нижче. У випадку коли відома точна кількість та тип датчиків та виконавчих пристроїв, що будуть підключатись до контролера, можна також виконати конфігурування каналів вводу/виводу.

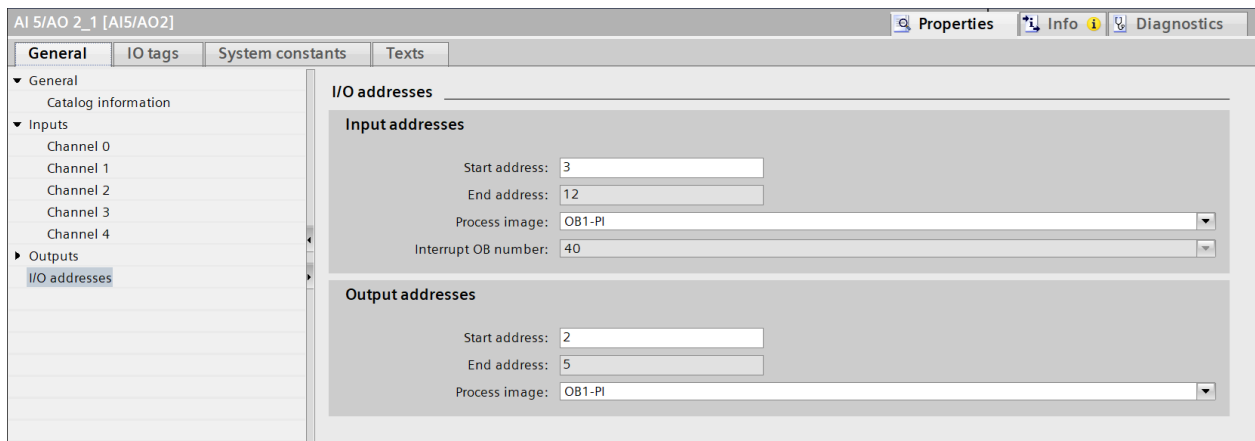


Рисунок 3.14 Конфігурування адрес аналогових входів та виходів контролера.

Після всіх налаштувань зберігаємо копію нового процесорного модуля у бібліотеку **Pressure control** з назвою **Blank CPU 314C-2 PN/DP**.

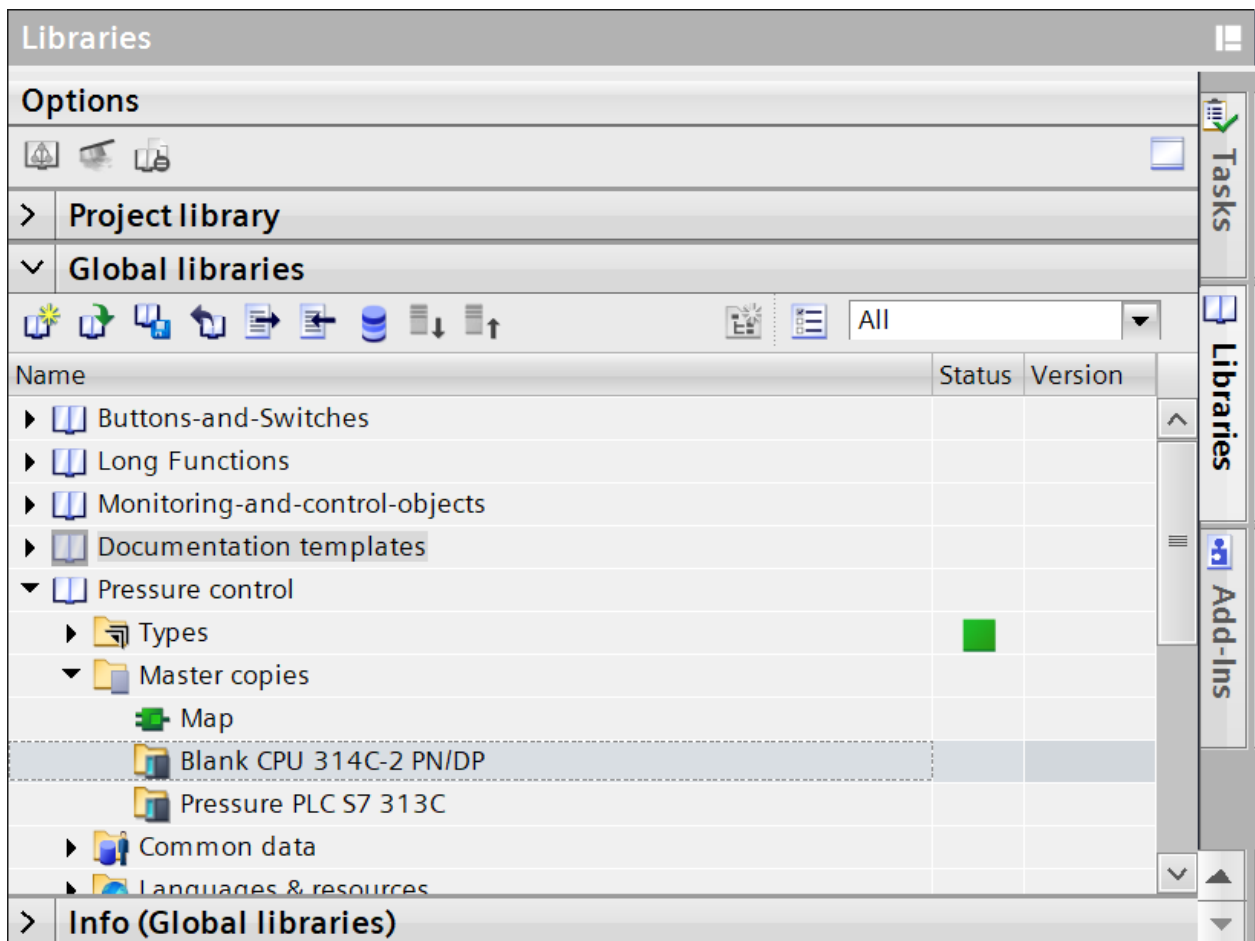


Рисунок 3.15 Додавання конфігурації Blank CPU 314C-2 PN/DP до бібліотеки Pressure control.

Після збереження змін у бібліотеці можна видалити процесорний модуль **PLC_2 [CPU 314C-2 PN/DP]** з проєкта за допомогою пункту **Delete** контекстного меню контролера у дереві проєкта TIA Portal.

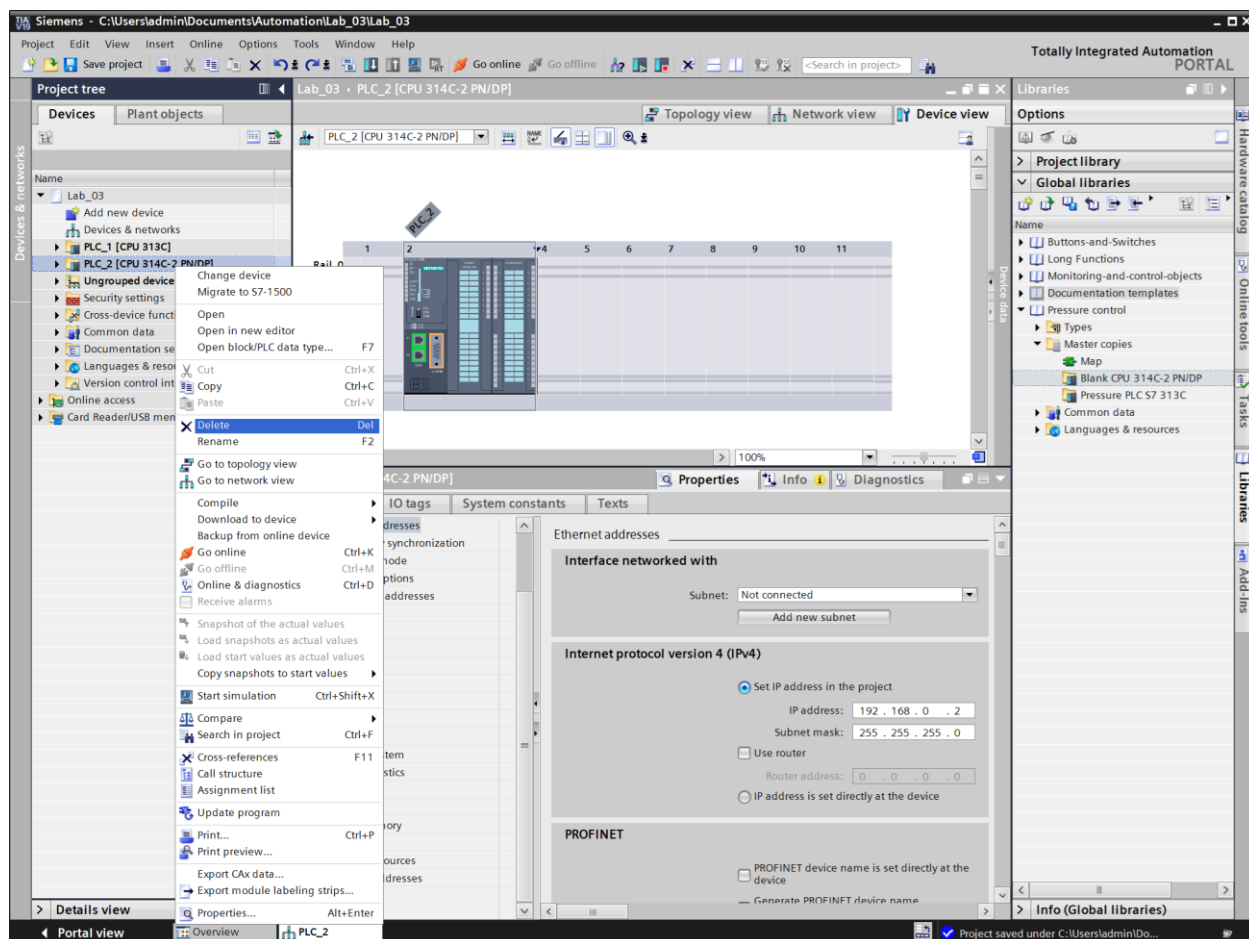


Рисунок 3.16 Видалення контролера з проєкта.

3.2.4 Заміна процесорного модуля у проєкті

У деяких випадках необхідність використання більш функціонального процесорного модуля може виявитись вже на етапі розробки програмного забезпечення після конфігурування та написання алгоритму. TIA Portal дозволяє спростити процес внесення таких змін. Для заміни процесорного модуля необхідно перейти до конфігурування контролера, правою кнопкою миші натиснути на зображенні процесорного модуля, який необхідно замінити та обрати пункт контекстного меню **Change device**. Має відкритись вікно вибору нового процесорного модуля з переліком всіх доступних для заміни варіантів.

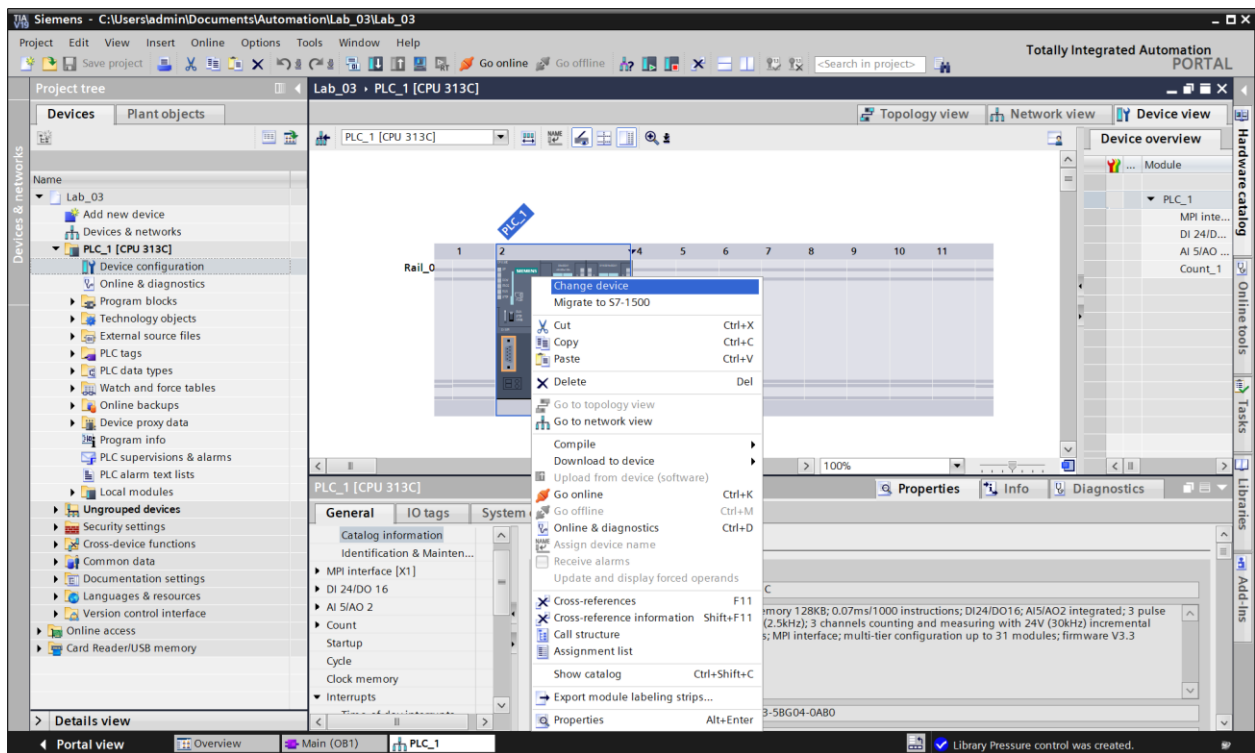


Рисунок 3.17 Виконання заміни процесорного модуля контролера.

У вікні заміни можна переглянути характеристики поточного процесорного модуля та модулів, які є доступними для заміни, а також всі зміни у проєкті, які будуть необхідні для використання нового процесорного модуля.

Розглянемо приклад заміни процесорного модуля контролера **CPU 313C** на процесорний модуль з мережним інтерфейсом **PROFINET**. Найближча за технічними характеристиками до поточного процесорного модуля модель, яка має мережний інтерфейс – це **CPU 314C-2 PN/DP 6ES7 314-6EH04-0AB0**, тож виконуємо заміну на цю модель. У нижній частині, де вказується інформація щодо сумісності модулів, відображені всі важливі зміни, які будуть виконані у проєкті при заміні поточного процесорного модуля на обраний.

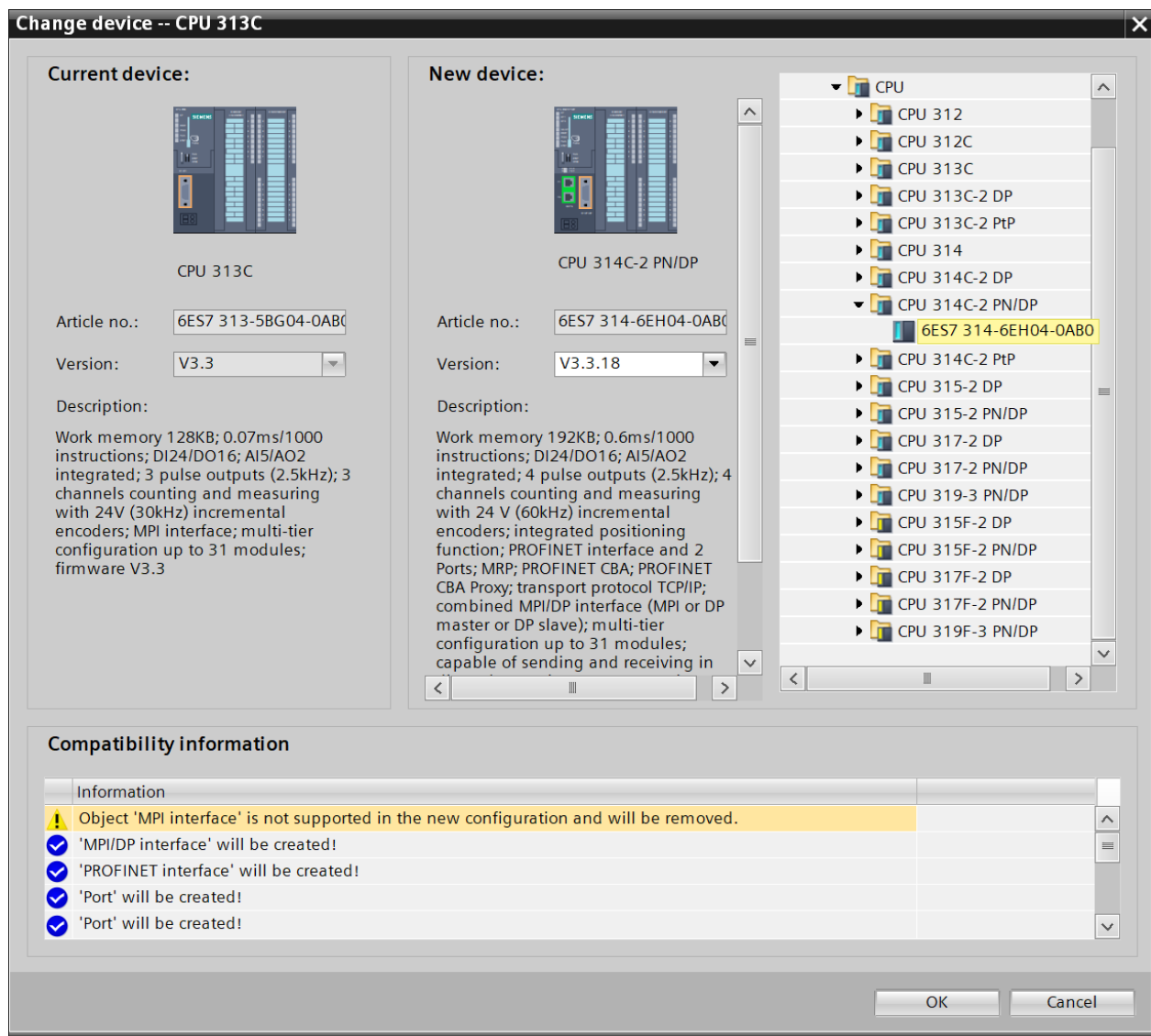


Рисунок 3.18 Порівняння характеристик процесорних модулів.

Після виконання заміни процесорного модуля зберігаємо проєкт. Для використання у наступних проєктах потрібно зберегти контролер з новим процесорним модулем у створеній бібліотеці, але перед цим перевірити та за необхідності змінити налаштування мережного інтерфейсу PROFINET. Для виконання наступних лабораторних робіт встановимо такі налаштування мережі: IP адреса **192.168.0.1**, маска підмережі **255.255.255.0**. Після виконання вказаних змін зберігаємо контролер з новим процесорним модулем у бібліотеці **Pressure control** з назвою **Pressure PLC CPU 314C-2 PN/DP**.

За результатом виконання лабораторної роботи має бути створена бібліотека **Pressure control** з функцією масштабування вхідного сигналу **Map**, контролером **Blank CPU 314C-2 PN/DP** з мінімальною конфігурацією та двома контролерами з алгоритмом керування - **Pressure PLC CPU 314C-2 PN/DP** з мережним інтерфейсом PROFINET та **Pressure PLC S7 313C** для проєктів без необхідності використання мережної взаємодії.

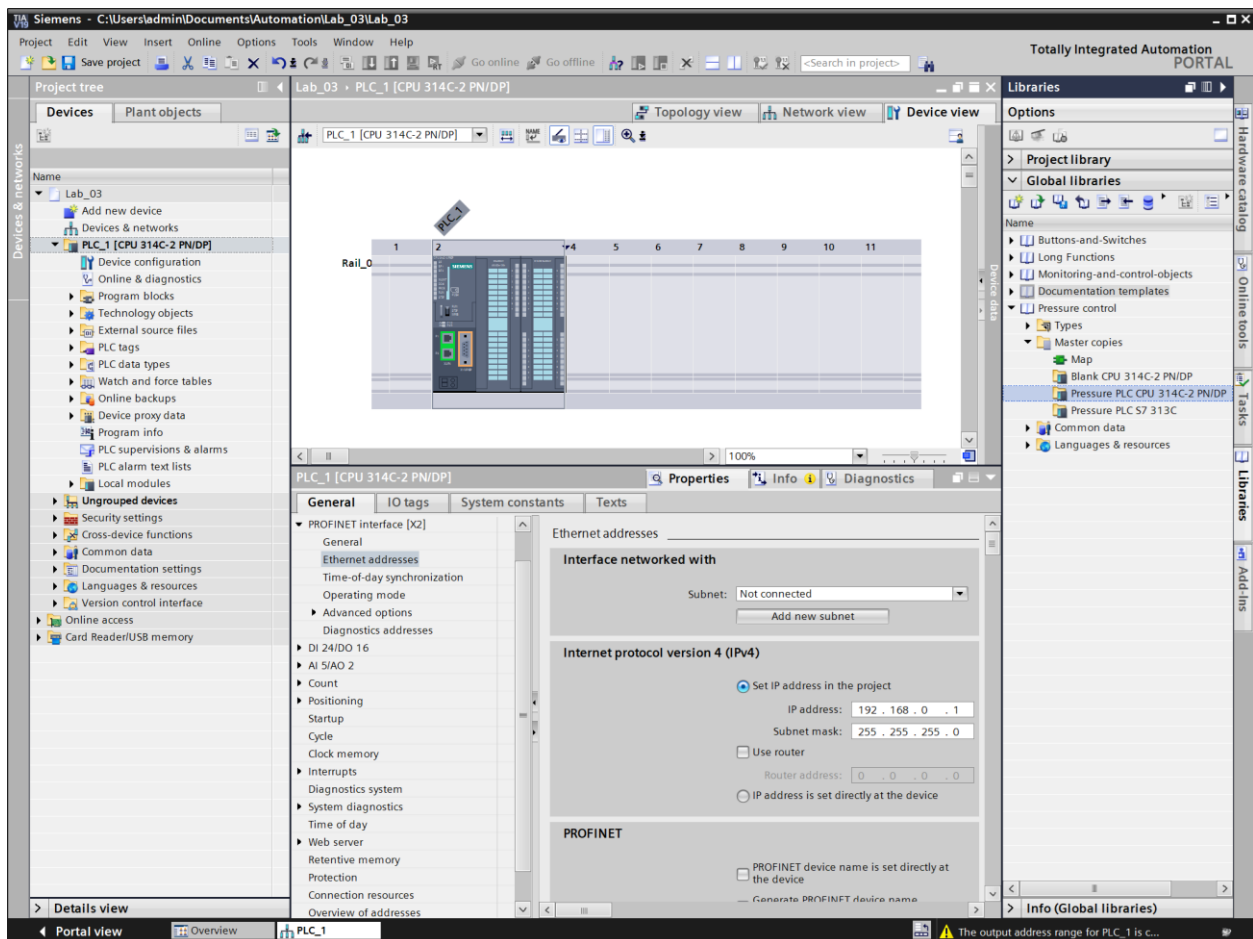


Рисунок 3.19 Зміни у бібліотеці Pressure control.

Зберігаємо зміни у бібліотеці Pressure control для подальшого використання.

3.2.5 Створення архіву бібліотеки

Як і проєкти, бібліотеки зберігаються на диску комп'ютера у вигляді набору файлів та директорій, що не завжди є зручним для резервного копіювання або пересилання проєктів електронною поштою чи іншими засобами передачі файлів. Так само, як і проєкти, бібліотеки також можуть зберігатись у вигляді архівів.

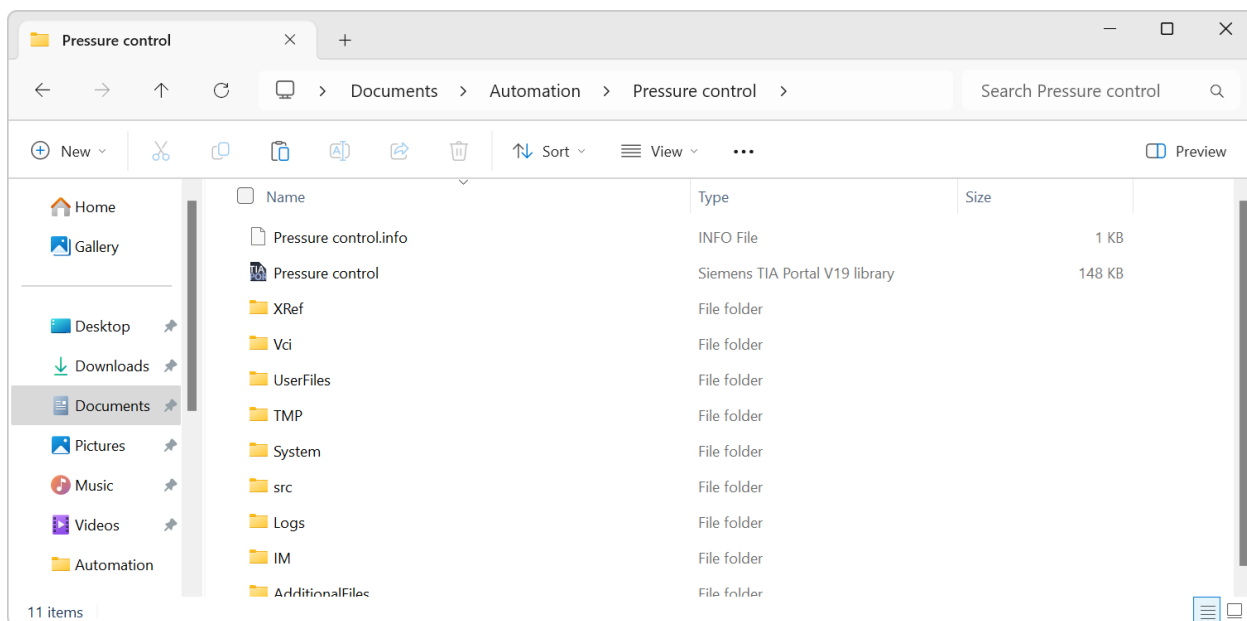


Рисунок 3.20 Вміст робочої директорії бібліотеки *Pressure control*.

Для створення архіву бібліотеки можна скористатись пунктом **Archive library...** контекстного меню бібліотеки.

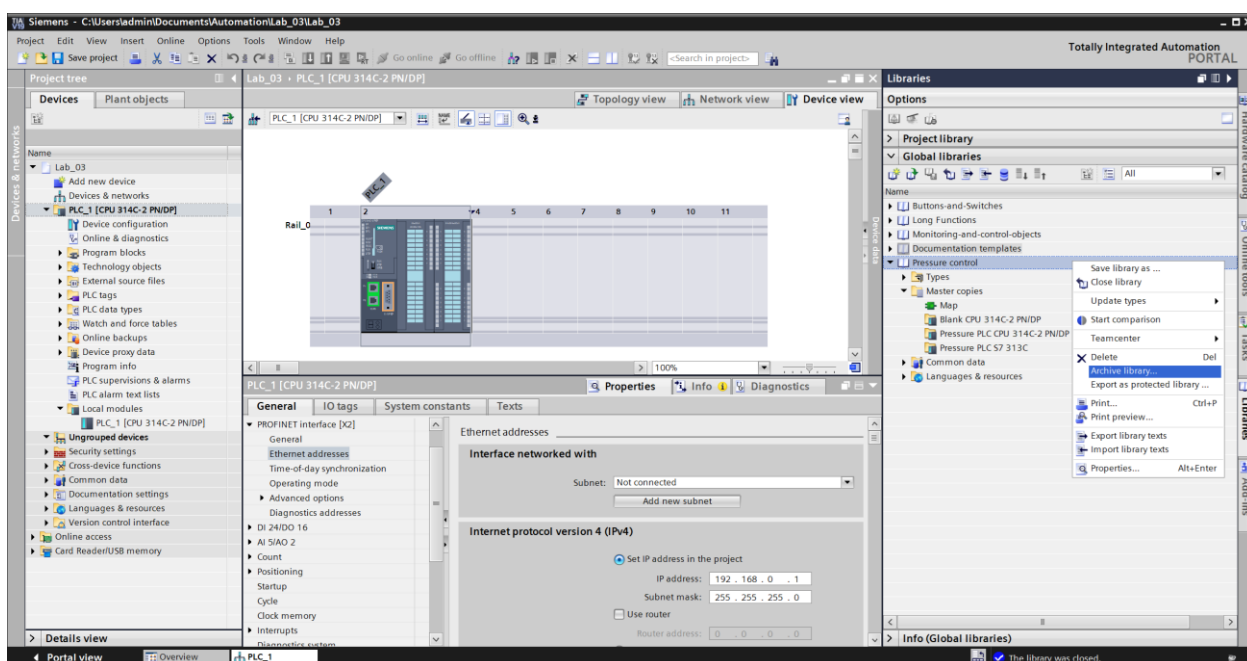


Рисунок 3.21 Створення архіву бібліотеки *Pressure control*.

Архів бібліотеки **TIA Portal v19** має розширення **.zal19** та, по суті, є архівом робочої директорії бібліотеки у форматі **ZIP**. Архіви бібліотек попередніх версій мають розширення, які містять порядкові номери цих версій TIA Portal. У вікні створення архіву бібліотеки потрібно вказати бібліотеку, яку потрібно архівувати та місце збереження архіву.

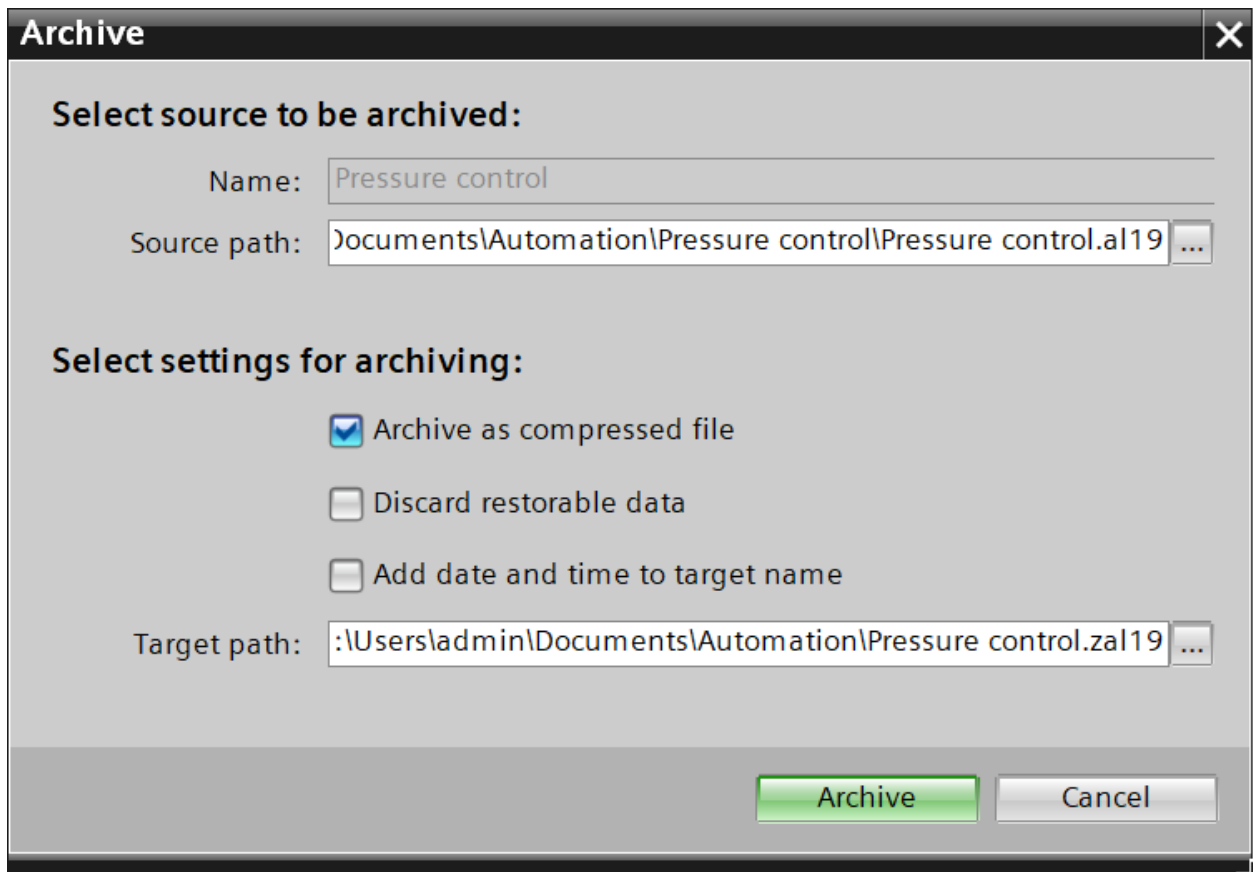


Рисунок 3.22 Вікно створення архіву бібліотеки.

Для відкриття архіву бібліотеки необхідно скористатись пунктом меню TIA Portal **Options → Global libraries → Open library...** За замовченням відкриється діалогове вікно відкриття бібліотеки TIA Portal, в якому архіви бібліотек не відображаються. Для того, щоб можна було побачити перелік доступних бібліотек, необхідно змінити значення у полі **Files of type** замість **Global library** на **Compressed libraries**. Після цього у відкритій директорії будуть відображатись назви всіх архівів бібліотек, які у ній збережені.

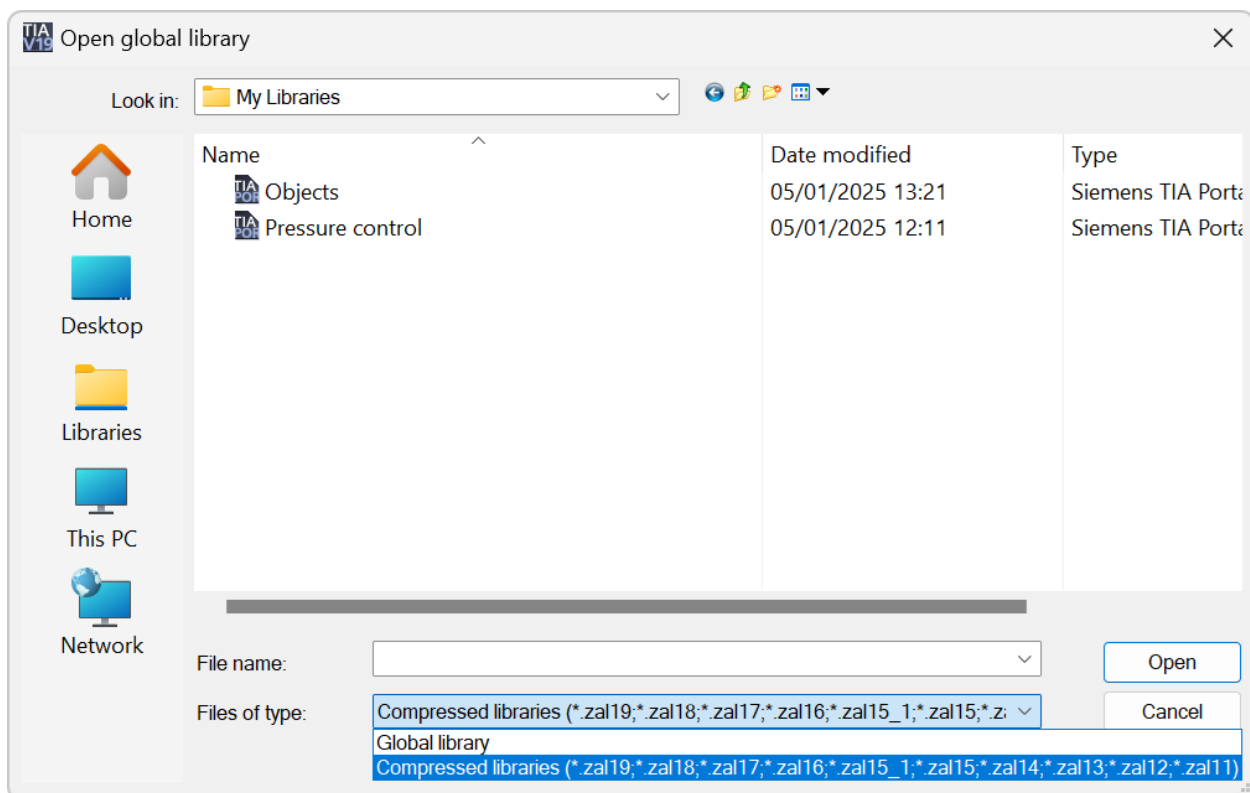


Рисунок 3.23 Відкриття архіву бібліотеки.

При відкритті бібліотеки (або архіву бібліотеки) варто звертати увагу на відмітку **Open as read-only**, яка знаходиться у нижній частині діалогового вікна відкриття бібліотеки. За замовченням ця відмітка є активною і всі бібліотеки відкриваються у режимі захисту від змін, тож за необхідності внесення змін у бібліотеку, цю відмітку при відкритті бібліотеки потрібно прибрати.

Якщо подальші зміни до бібліотеки є небажаними, її можна експортувати як захищену від запису. Робота з захищеною від запису бібліотекою не забороняє внесення змін у елементи бібліотеки, які були використані у проєкті, але вміст самої бібліотеки не може бути змінений, тому перед захистом бібліотеки варто зробити резервну копію бібліотеки у незахищеному вигляді. Для збереження бібліотеки у захищеному вигляді можна використовувати пункт **Export as protected library...** контекстного меню бібліотеки. Якщо бібліотека збережена у захищеному від запису вигляді, при спробі її відкрити зі знятою відміткою Open as read-only буде з'являться помилка.

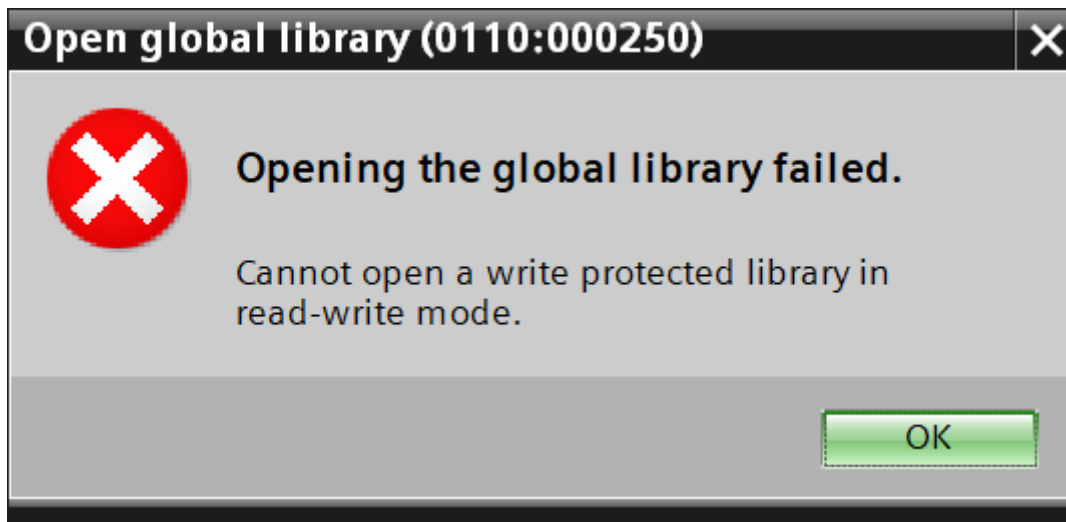


Рисунок 3.24 Спроба відкрити захищену від запису бібліотеку.

3.3 Контрольні завдання

1. Створити документацію для контролера Pressure PLC CPU 314C-2 PN/DP з бібліотеки Pressure control.
2. Додати до бібліотеки Pressure control контролер PLC_1 [CPU 312C] з конфігурацією та алгоритмом першої лабораторної роботи та зберегти архів бібліотеки.
3. Видалити з бібліотеки Pressure control функцію Map та зберегти архів бібліотеки з захистом від змін.

4 Лабораторна робота №4.

Мета роботи: ознайомитись з мережним обміном даними контролерів, отримати навички імітаційного моделювання та налагодження систем регулювання в ТІА Portal.

4.1 Завдання

1. Підготовка проєкту до імітаційного моделювання
2. Налаштування мережного підключення контролерів
3. Розробка алгоритму обміну даними між контролерами
4. Розробка алгоритму імітаційної моделі об'єкта керування
5. Проведення імітаційного моделювання системи регулювання

4.2 Порядок виконання роботи

4.2.1 Підготовка проєкту до імітаційного моделювання

Розробка та перевірка програмного забезпечення системи автоматизації є складним та відповідальним процесом, який може вимагати значних витрат часу при монтажі системи на реальному об'єкті. Скоротити цей час та виявити помилки алгоритмів можна за допомогою імітаційного моделювання, яке замість реального об'єкта використовує фізичні або програмні моделі об'єкта керування. Імітаційне моделювання також дає можливість скоротити час функціонування обладнання у неефективних або небезпечних режимах під час налагодження систем регулювання.

Для проведення імітаційного моделювання та визначення оптимальних параметрів налаштування регулятора в розробленому алгоритмі будемо використовувати програмну модель об'єкта керування. Для запобігання значним змінам у розробленому алгоритмі використаємо окремий контролер, який буде виконувати роль об'єкта. Одним з варіантів реалізації такої схеми може бути зв'язок двох контролерів за допомогою протоколу передачі даних PROFINET.

Оскільки модель процесорного модуля контролера CPU 313C, яка була обрана для реалізації алгоритму другої лабораторної роботи не має відповідного інтерфейсу, будемо використовувати варіант системи з процесорним модулем **Pressure PLC CPU 314C-2 PN/DP**, що знаходиться у бібліотеці **Pressure control**, створеної у третій лабораторній роботі. Для моделювання об'єкта керування будемо використовувати контролер **Blank CPU 314C-2 PN/DP** з бібліотеки **Pressure control**, який має інтерфейс PROFINET та не виконує алгоритм керування.

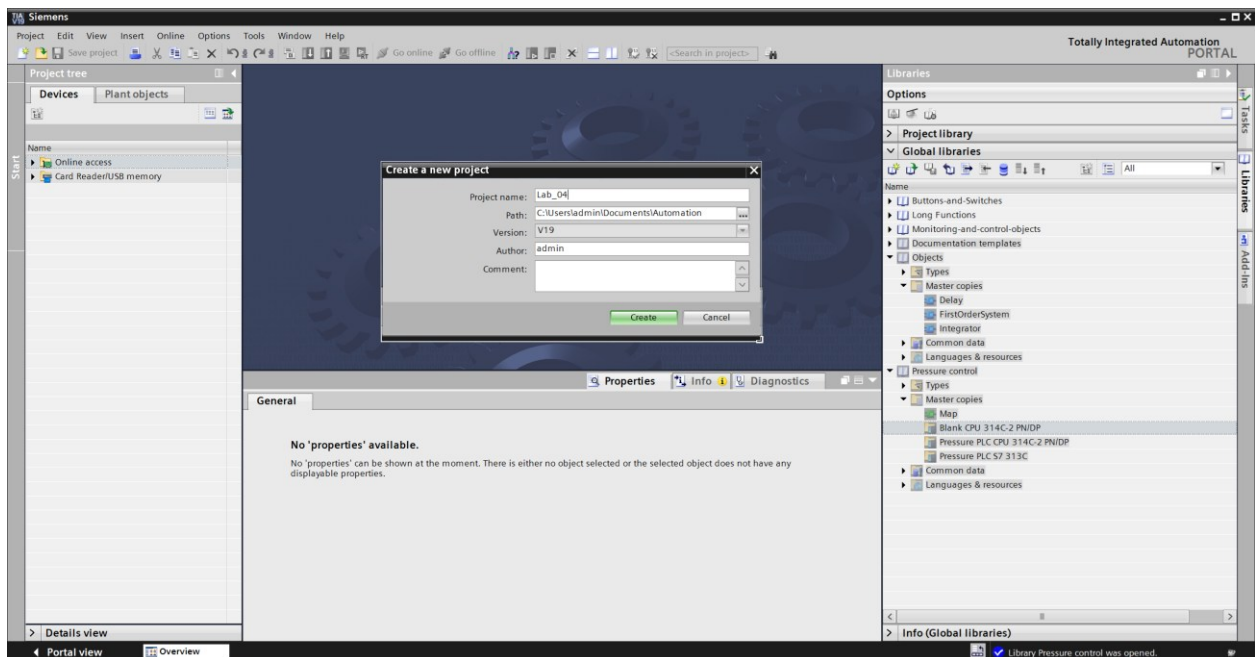


Рисунок 4.1 Створення проекту імітаційного моделювання.

Підготуємо проєкт для імітаційного моделювання. Для цього відкриваємо у режимі читання бібліотеку **Pressure control** та **Objects** і створюємо новий проєкт з назвою **Lab_04**. Перетягуємо контролер **Pressure PLC CPU 314C-2 PN/DP** з бібліотеки **Pressure control** у дерево нового проєкта **Lab_04**. Варто зазначити, що при перетягуванні контролера у дерево проєкта його назва у проєкті за замовченням стає **PLC_1 [CPU 314C-2 PN/DP]**, тож її потрібно буде змінити за допомогою пункту **Rename** контекстного меню в дереві проєкта на **Control system**.

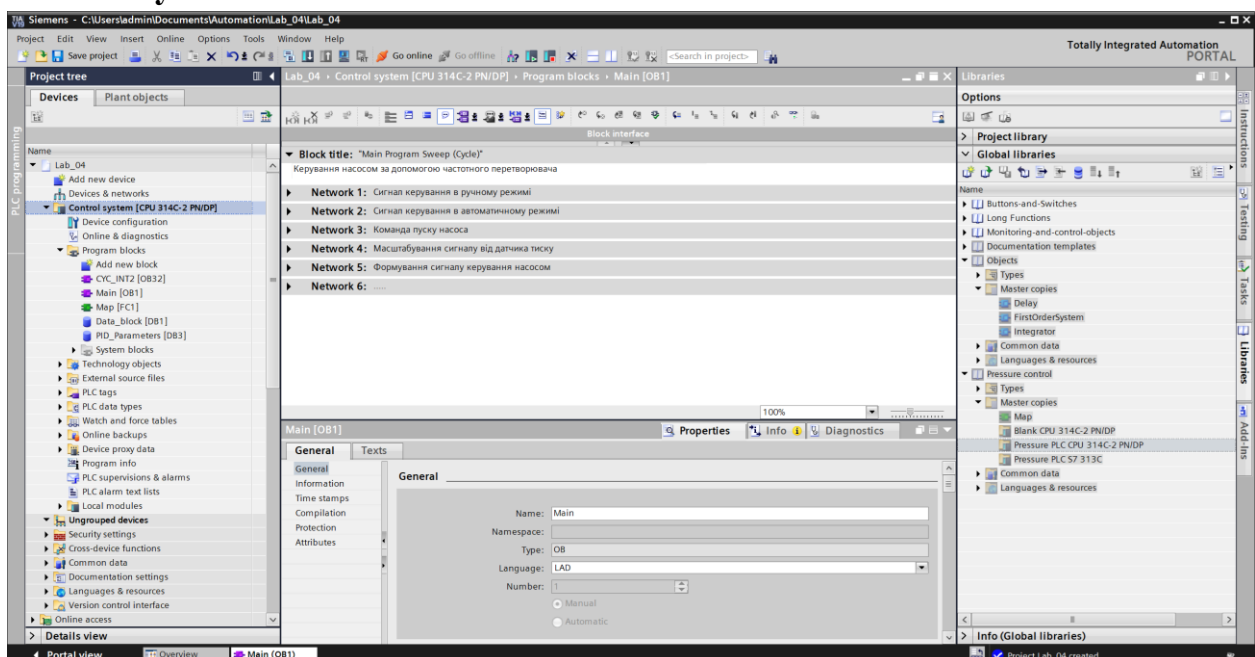


Рисунок 4.2 Перенесення контролера Pressure PLC CPU 314C-2 PN/DP з бібліотеки у проєкт.

Переносимо контролер **Blank CPU 314C-2 PN/DP** з бібліотеки **Pressure control**, у дерево проєкта та перейменовуємо його на **Imitation model**. Таким чином, у проєкті має бути два контролери: **Control system [CPU 314C-2 PN/DP]** з алгоритмом керування насосом зі змінною продуктивністю та **Imitation model [CPU 314C-2 PN/DP]** без алгоритму.

У алгоритмі керування контролера **Control system** необхідно зробити мінімальні зміни для того, щоб до контролера можна було підключити імітаційну модель. Оскільки замість реального значення від датчика тиску контролер має приймати значення тиску, отримане від контролера **Imitation model** шляхом математичного моделювання відповідно до зміни сигналу керування, у блоці даних **Data_block [DB1]** потрібно створити змінну **rSimulatedProcessValue** типу **Real**, яка після отримання значення від контролера **Imitation model** подаватиме значення на вхід **PV_IN** регулятора **CONT_C** замість змінної **rPressureSensor**.

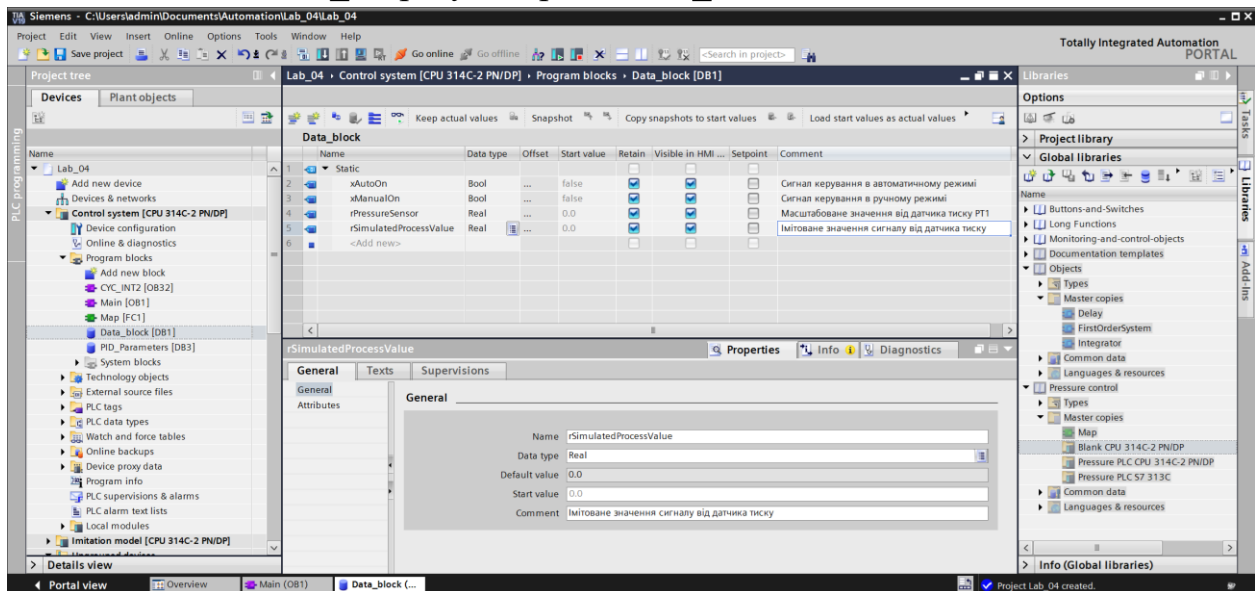


Рисунок 4.3 Створення змінної **rSimulatedProcessValue**.

Можна помітити, що після створення нової змінної у полі **Offset** напроти кожної змінної блоку даних зникає значення. Оскільки це значення визначає зміщення адреси кожної змінної відносно початку ділянки пам'яті, що виділена для збереження таблиці даних та використовується для налаштування передачі даних, його варто оновити. Це можна зробити якщо виділити у дереві проєкта блок даних **Data_block [DB1]** та виконати компілювання блоку за допомогою команди меню TIA Portal **Edit → Compile** або за допомогою гарячої комбінації клавіш **Ctrl+B**. Після компілювання всі змінні блоку даних повинні отримати оновлені адреси, які можна буде побачити у полі **Offset**.

Підключаємо змінну **rSimulatedProcessValue** до входу регульованого параметра **PV_IN** регулятора **CONT_C** замість змінної **rPressureSensor**. Відмічаємо, що адреса змінної **rSimulatedProcessValue** записана у форматі

%DB1.DB1. Перша частина адреси **DB1** означає, що змінна зберігається у блоці даних, що має номер **1**, друга частина **DB1** визначає, що ділянка пам'яті, де зберігається значення змінної, починається з 6 байту відносно початку ділянки пам'яті блоку даних. Запис **DB1** означає, що для збереження змінної виділено 4 байти пам'яті, оскільки змінна зберігає значення типу **Real**.

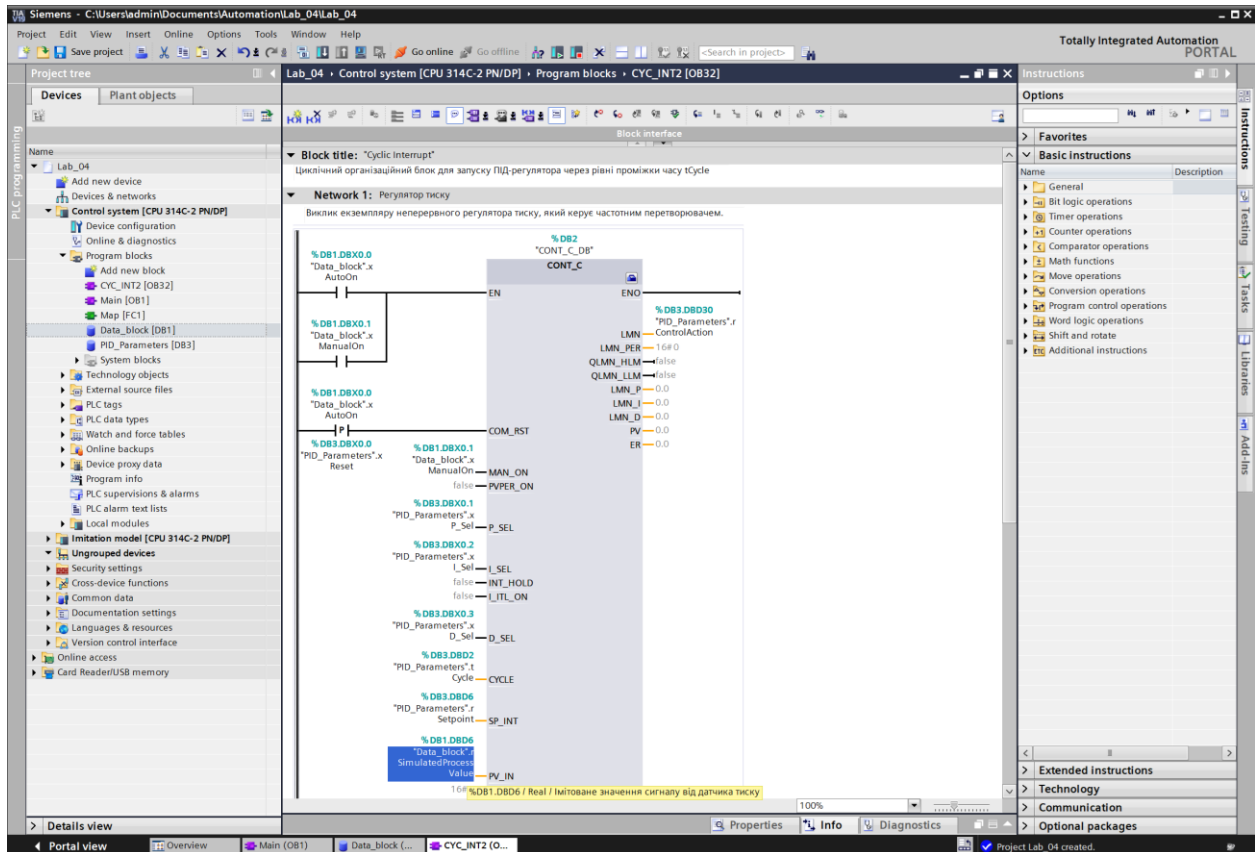


Рисунок 4.4 Заміна сигналу від датчика тиску на імітоване значення.

4.2.2 Налаштування мережного підключення контролерів

Оскільки для роботи імітаційної моделі потрібно передавати значення регулюючої **rControlAction** дії від контролера **Control system** до контролера **Imitation model** та передачу значення імітованого сигналу датчика тиску **rSimulatedProcessValue** у зворотному напрямку, потрібно встановити зв'язок між цими контролерами.

Встановлення зв'язку між двома контролерами передбачає декілька етапів. Перший етап передбачає створення підмережі, до якої будуть входити всі пристрої, які будуть обмінюватись даними, встановлення фізичного зв'язку між ними та призначення мережних адрес.

Переходимо до розділу **Devices & networks** дерева проєкта TIA Portal. У цьому розділі представлені зображення всіх пристроїв, які є у проєкті.

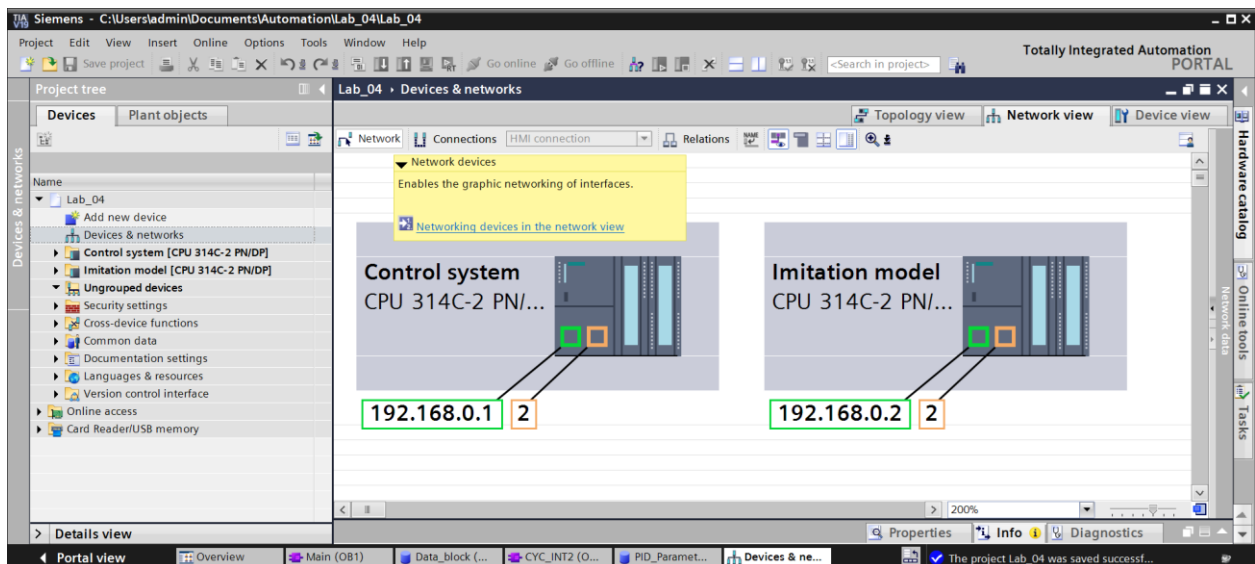


Рисунок 4.5 Розділ *Devices & networks* проєкта.

Потрібно активувати на панелі інструментів кнопку **Network** та за допомогою миші протягнути лінію зв'язку від умовного зображення мережного інтерфейсу **Ethernet** контролера **Control system** до контролера **Imitation model**.

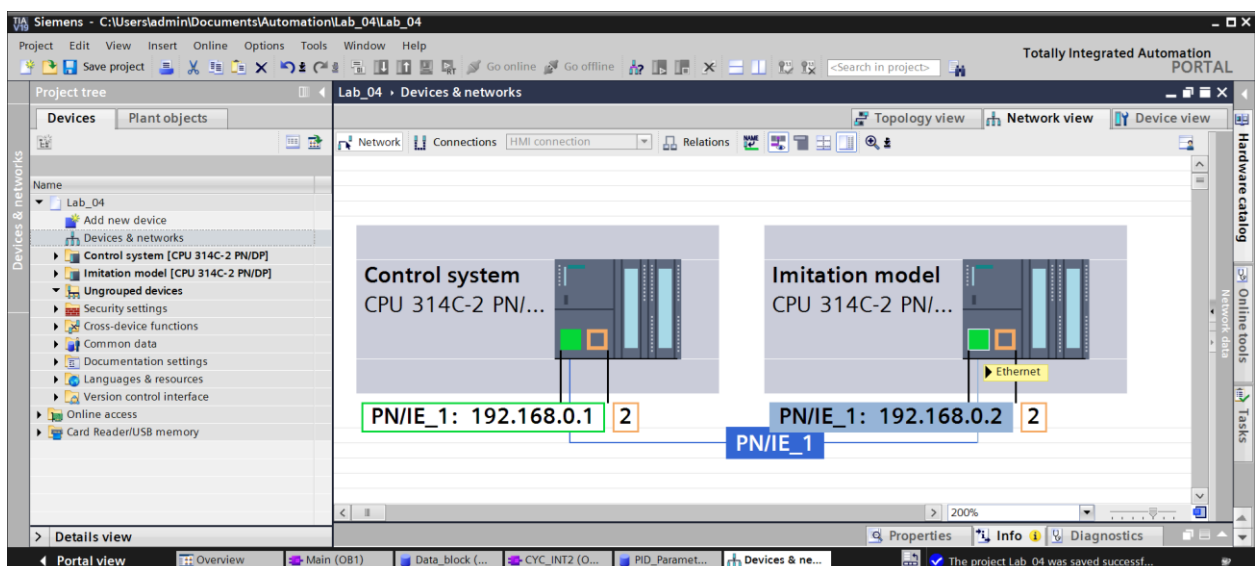


Рисунок 4.6 Встановлення фізичного зв'язку між контролерами.

Результатом має бути неперервна лінія зв'язку, яка з'єднає мережні інтерфейси обох контролерів, позначена назвою цього зв'язку **PN/IE_1**.

Наступним кроком потрібно на панелі інструментів активувати кнопку **Connections** та обрати у списку варіантів підключень пункт **S7 connection**.

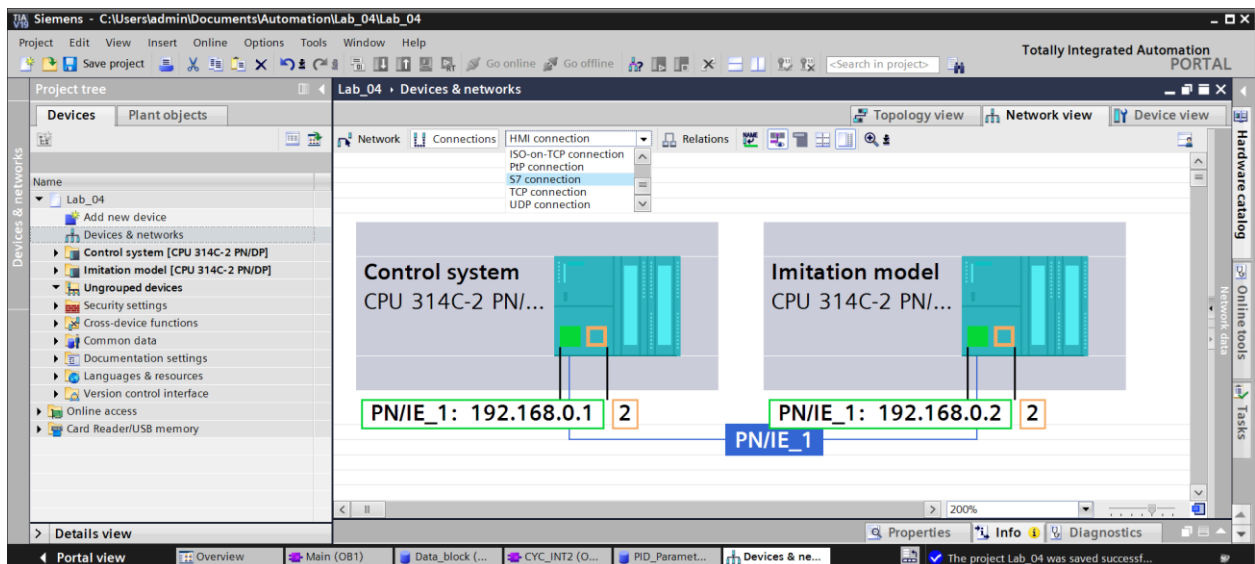


Рисунок 4.7 Вибір типу підключення.

За допомогою миші проводимо лінію зв'язку від інтерфейсу **Ethernet** контролера **Control system** до інтерфейсу контролера **Imitation model**.

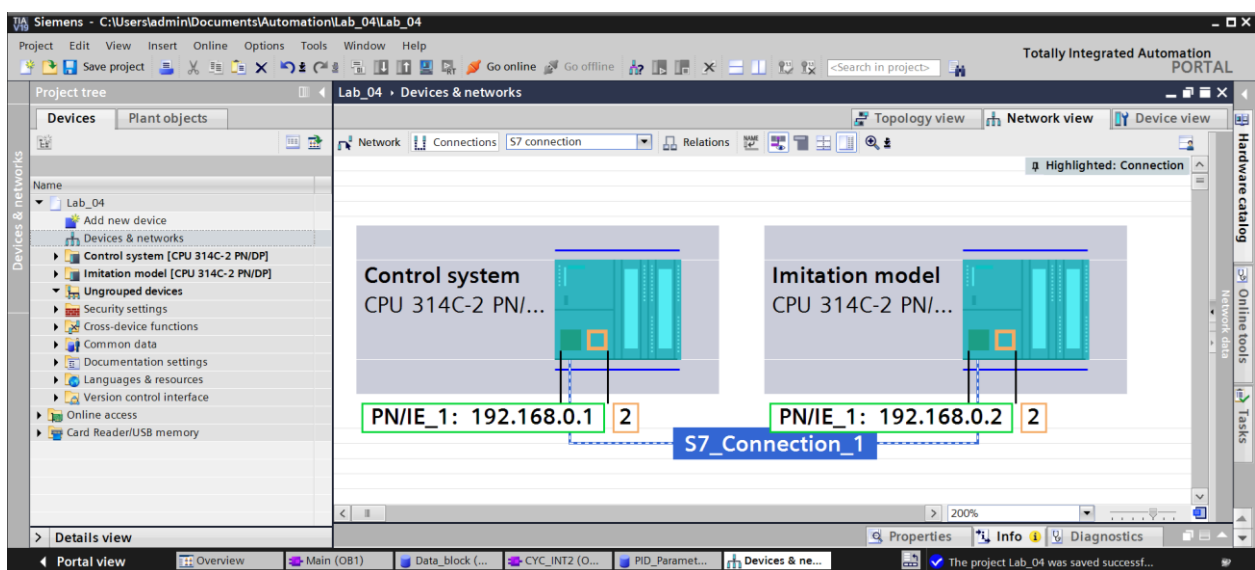


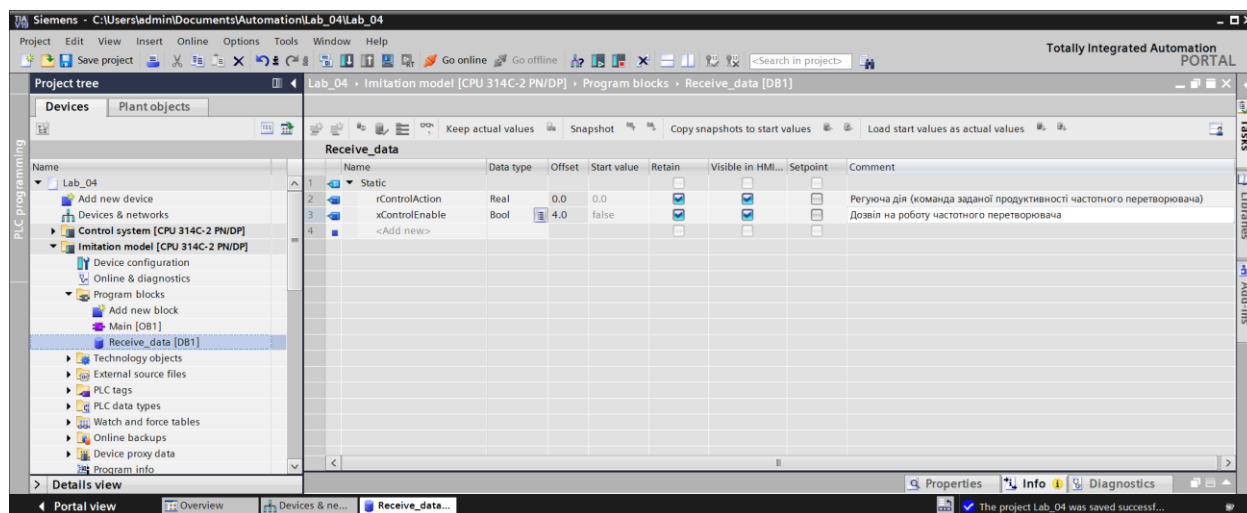
Рисунок 4.8 Налаштування типу підключення для обміну даними.

В результаті між контролерами має з'явитись пунктирна лінія зв'язку, яка підписана **S7_Connection_1**. Слід зазначити, що мережні адреси контролерів повинні бути різними, але знаходитись в одній підмережі. Як видно на рисунку вище, контролер **Control system** має IP адресу **192.168.0.1**, а контролер **Imitation model** - **192.168.0.2**.

4.2.3 Розробка алгоритму обміну даними між контролерами

Після налаштування мережного підключення контролерів переходимо до програмної реалізації обміну даними.

Для отримання даних з контролера **Control system** на контролері **Imitation model** створюємо блок даних **Receive_data**, у якому створюємо змінну **rControlAction** типу **Real** для отримання значення регулюючої дії та змінну **xControlEnable** для отримання сигналу дозволу на роботу частотного перетворювача, який подається на перший дискретний вихід **DO1** контролера **Control system**. Для отримання адрес змінних у блоці даних виконуємо його компіляцію командою меню TIA Portal **Edit → Compile**.



*Рисунок 4.9 Створення блоку даних **Receive_data**.*

Для надсилання даних з контролера **Imitation model** на контролер **Control system** створюємо блок даних **Send_data**, у якому створюємо змінну **rSimulatedProcessValue** типу **Real**, що міститиме значення імітованого сигналу від датчика тиску. Так само, як і для блоку даних **Receive_data**, виконуємо компілювання блоку даних **Send_data** для визначення адреси змінної **rSimulatedProcessValue**. Оскільки ця змінна є першою та єдиною у цьому блоці даних, вона буде мати нульову адресу відносно початку діапазону адрес блока даних **Send_data**.

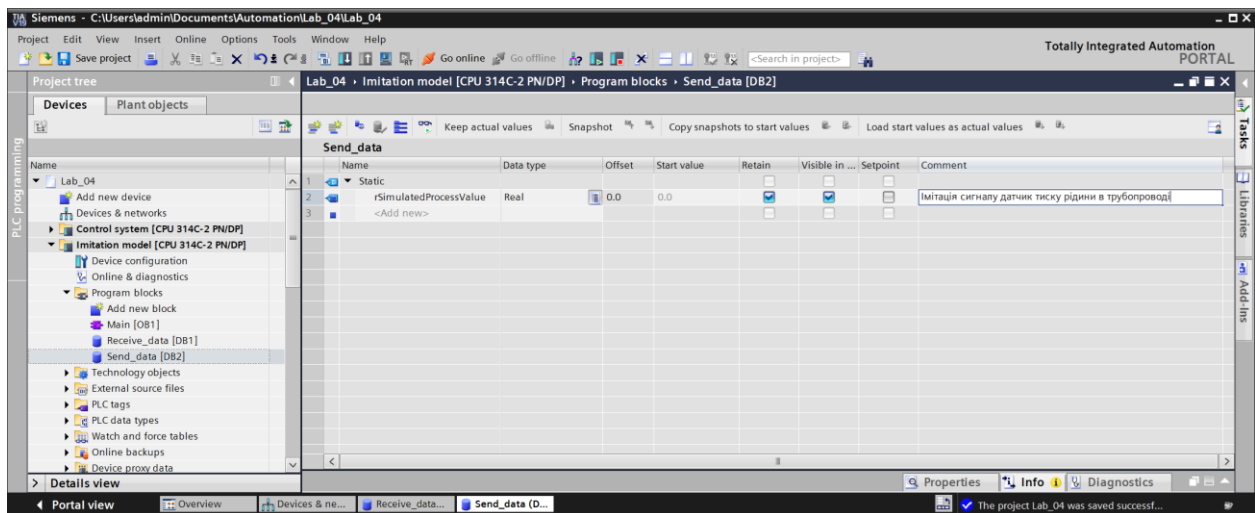


Рисунок 4.10 Створення блоку даних *Send_data*.

Алгоритм передачі та прийому даних будемо організовувати в організаційному блоці **Main [OB1]** за допомогою функціональних блоків **PUT** та **GET**. Блок **GET** призначений для зчитування даних з віддаленого контролера, а блок **PUT** – для запису даних на віддалений контролер. У секцію **Network 1** перетягуємо інструкцію **GET** з розділу **Communication** бібліотеки інструкцій TIA Portal.

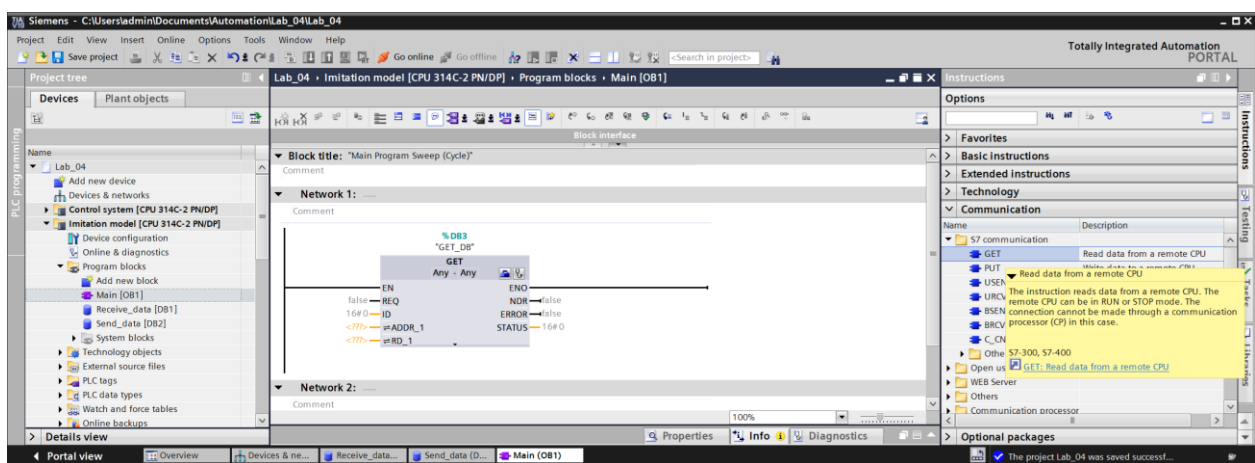


Рисунок 4.11 Додавання у проєкт блоку *GET*.

Для налаштування параметрів блоку можна скористатись вікном конфігурування, яке можна відкрити за допомогою кнопки **Start configuration** з зображенням синього ящика для інструментів, що знаходиться у правому верхньому куті блока **GET**. У вікні конфігурування переходимо до розділу налаштування параметрів підключення **Connection parameter**, напроти якого відображається червоний значок, що повідомляє про відсутність підключення, та обираємо у списку під заголовком **Partner** контролер, з якого потрібно отримати значення параметрів – **Control system [CPU 314C-2 PN/DP]**.

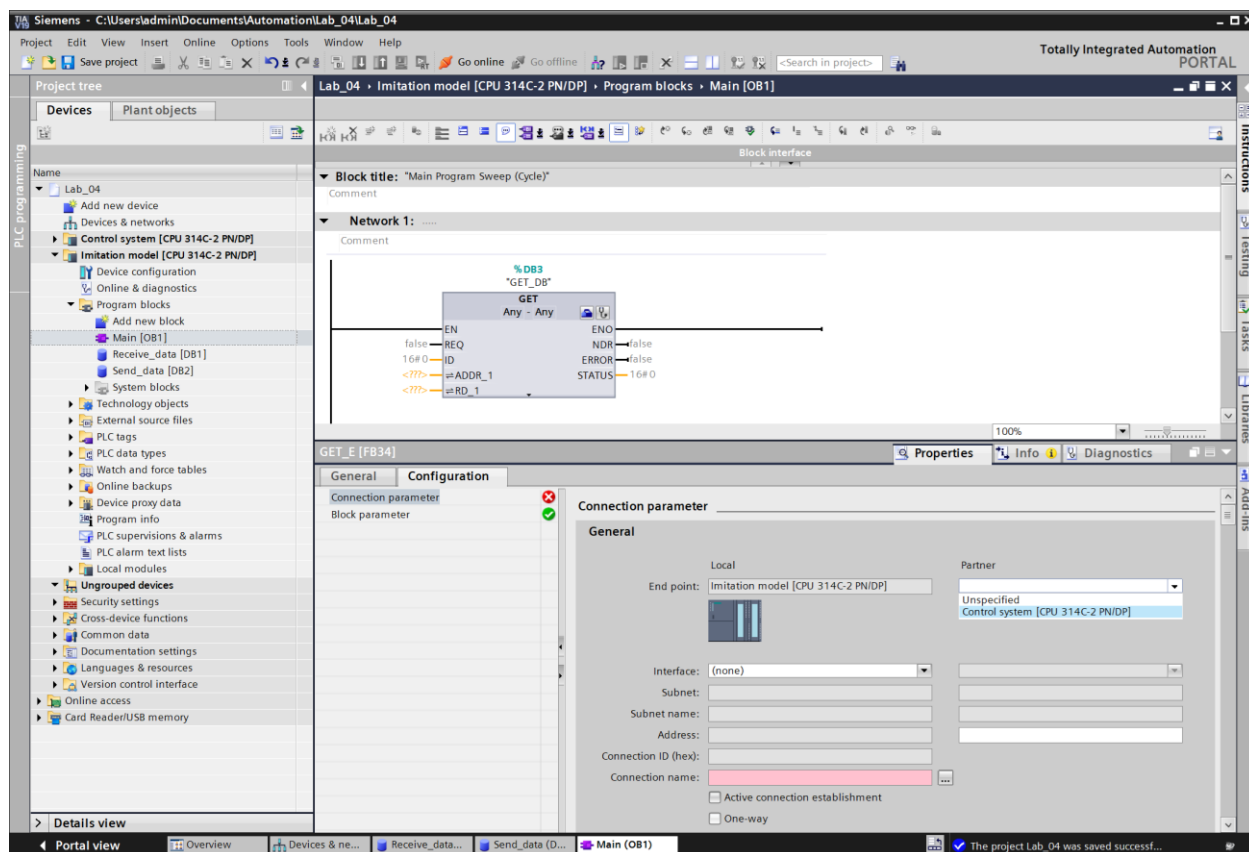


Рисунок 4.12 Налаштування параметрів підключення блоку GET.

Якщо все виконано вірно, після встановлення параметрів зв'язку червоний значок напроти пункту **Connection parameter** зміниться на зелений, а в полях налаштування автоматично з'являться інтерфейси контролерів, які будуть використовуватись для передачі даних, назва підмережі, мережні IP адреси контролерів та назва підключення **S7_Connection_1**.

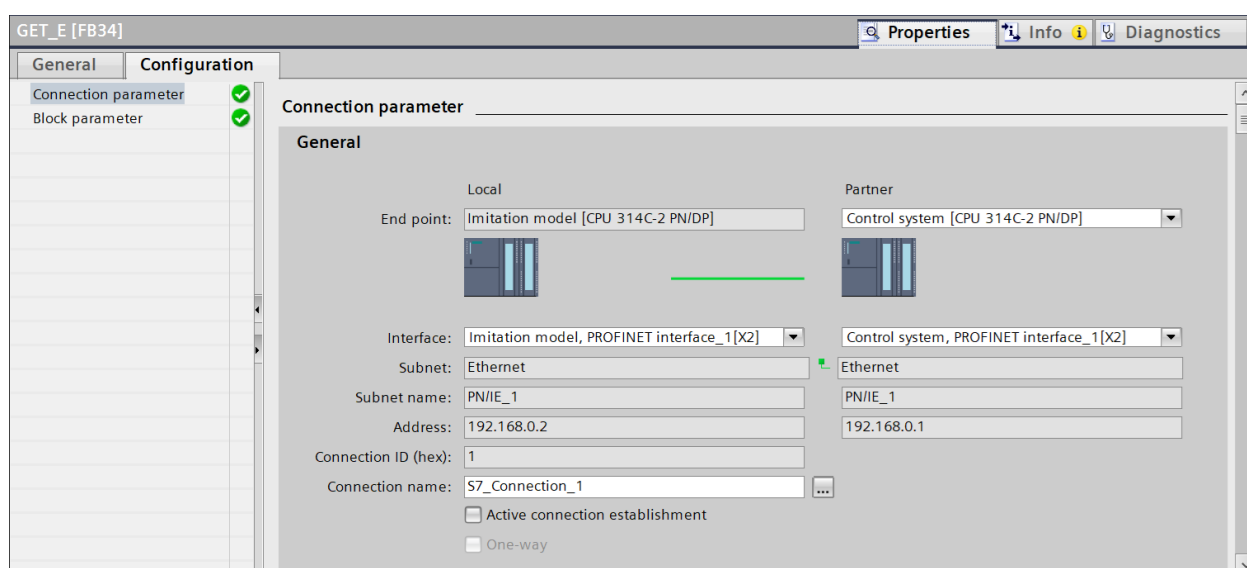


Рисунок 4.13 Параметри налаштування зв'язку блока GET.

Подальше конфігурування блоку GET виконуємо наступним чином: до входу **REQ**, який запускає процес отримання даних з віддаленого контролера, підключаємо нормально замкнений контакт, прив'язаний до вихідного параметра **NDR** цього ж блоку, що вказує на успішне отримання даних з віддаленого контролера.

Значення параметра **ADDR_1** є посиланням на адресу змінної **rControlAction** блоку **PID_Parameters** віддаленого контролера **Control system**. Запис **P#DB3.DBX30.0 REAL 1** означає, що команда **GET** зчитуватиме значення одної змінної типу **REAL** з блоку даних, який має порядковий номер **3**, починаючи з нульового біта **30** байту відносно початку блоку даних.

Значення **P#DB1.DBX0.0 REAL 1** параметра **RD_1** означає, що отримане з адреси **ARRD_1** значення команда **GET** записуватиме до локальної змінної типу **REAL**, яка знаходиться у блоці даних з порядковим номером **1**, починаючи з нульового біта нульового байту відносно початку блоку даних. Ця адреса відповідає змінній **rControlAction** локального блоку даних **Receive_data**.

Оскільки визначення змінних, що відповідають вказаним адресам, є задачею, що потребує витрат часу, для полегшення роботи з програмою варто залишати коментарі, що пояснюють значення параметрів.

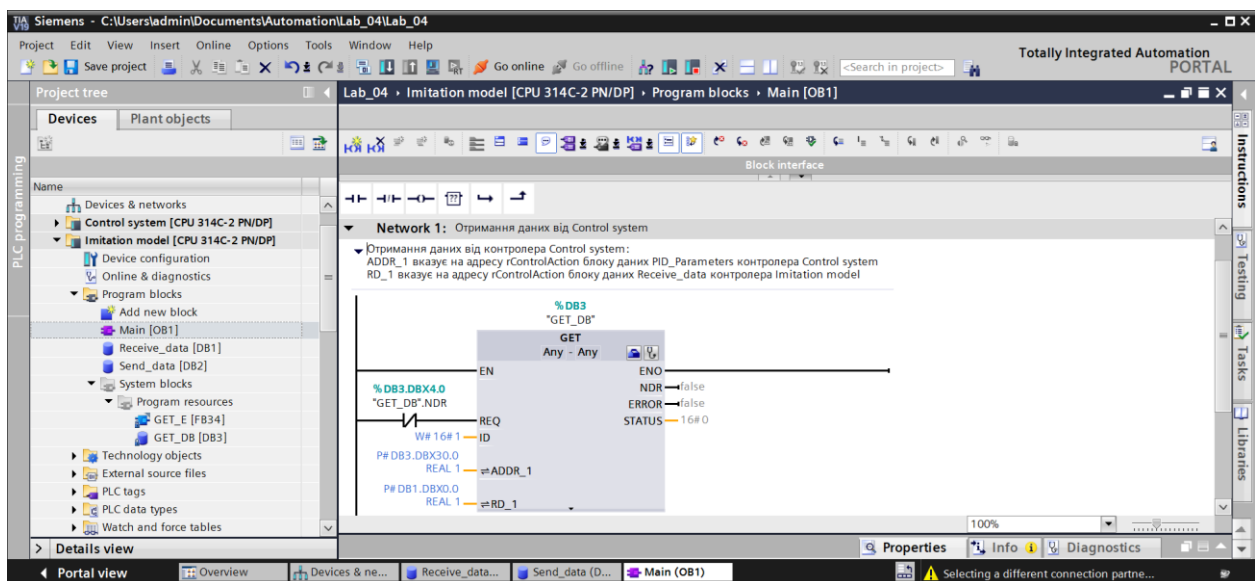


Рисунок 4.14 Параметри блока GET для зчитування значення регулюючої дії з контролера Control system.

Блок **GET** дозволяє зчитувати одразу декілька параметрів, що знаходяться у різних ділянках пам'яті віддаленого контролера, для відображення полів налаштування потрібно натиснути на зображення трикутника у нижній частині блока **GET**.

Заповнюємо параметри **ADDR_2** та **RD_2** блока **GET** посиланнями на адреси, які дозволять зчитувати стан першого дискретного виходу контролера

Control system, який відповідає за дозвіл на роботу частотного перетворювача та змінної **xControlEnable** локального блоку **Receive_data**, яка буде використовуватись для імітації роботи об'єкта.

Значення параметра **ADDR_2** потрібно встановити рівним **P#Q0.0 BOOL 1**, що відповідає нульовому біту нульового вихідного байту контролера **Control system**, в якому відповідно до конфігурації контролера **Control system** зберігається значення першого дискретного виходу.

Значення параметра **RD_2** встановлюємо рівним **P#DB1.DBX4.0 BOOL 1**, що є посиланням на адресу змінної **xControlEnable** локального блоку **Receive_data**.

Вказані зміни описуємо у коментарях секції **Network 1**.

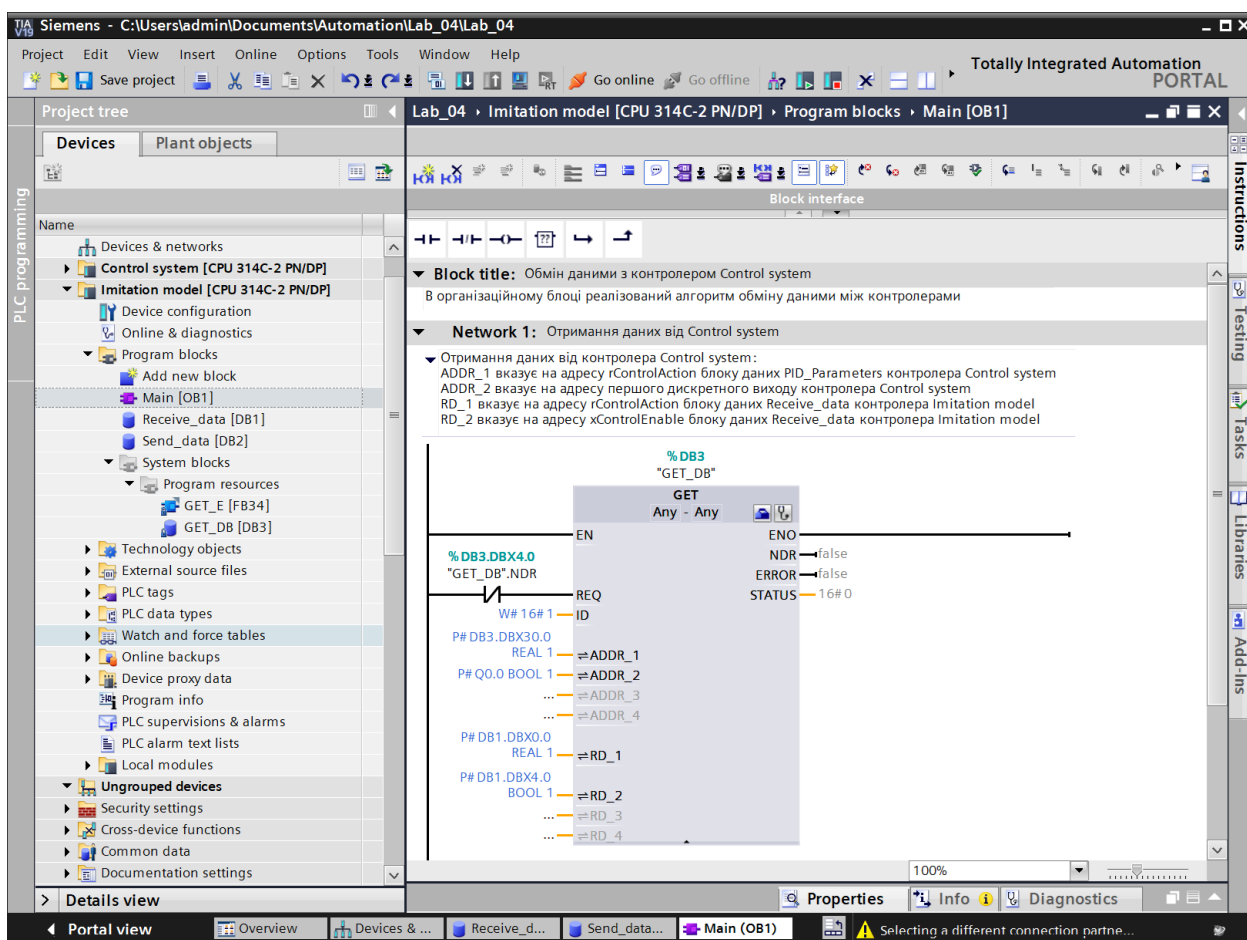


Рисунок 4.15 Конфігурація блоку GET.

Окрім зчитування значень регулюючої дії та дозволу на роботу частотного перетворювача, потрібно також здійснювати запис значення імітованого сигналу датчика тиску на контролер **Control system**. Для виконання передачі даних з локального контролера на віддалений призначений функціональний блок **PUT**, який також можна знайти у розділі **Communication** бібліотеки інструкцій TIA Portal.

Перетягуємо блок **PUT** до другої секції організаційного блоку **Main [OB1]**. Алгоритм його роботи є схожим на алгоритм функціонування блоку **GET**: при зміні стану входу **REQ** з вимкненого на ввімкнений відбувається запис локального значення, вказівник на яке задається входом блоку **SD_1**, у ділянку пам'яті віддаленого контролера, вказівник на цю ділянку задається входом блоку **ADDR_1**.

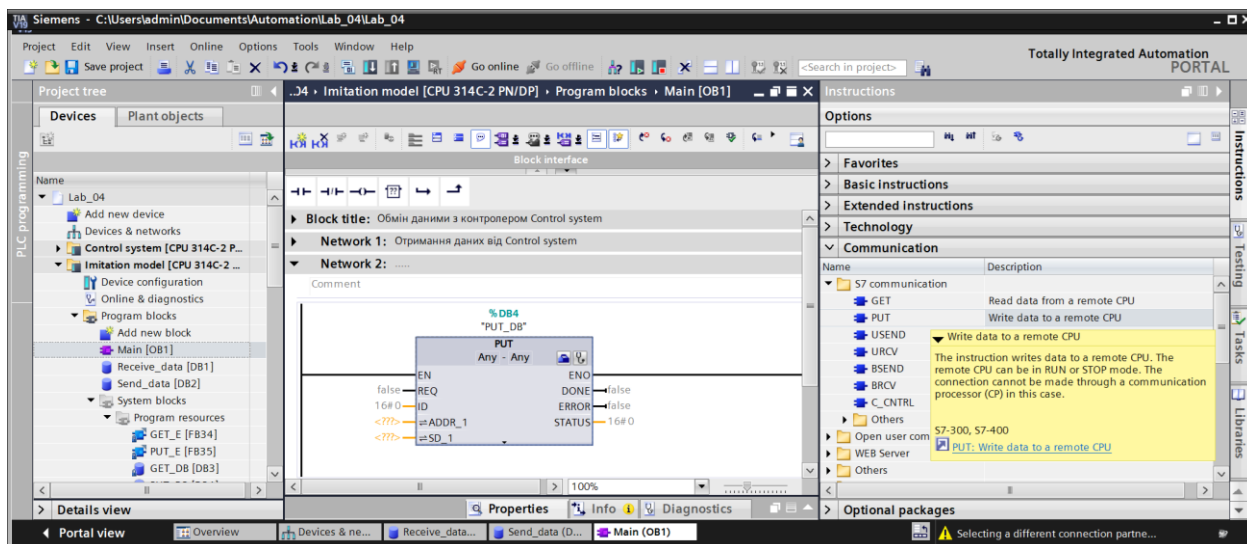


Рисунок 4.16 Додавання у проєкт блоку PUT.

Виконуємо налаштування підключення аналогічно блоку **GET**.

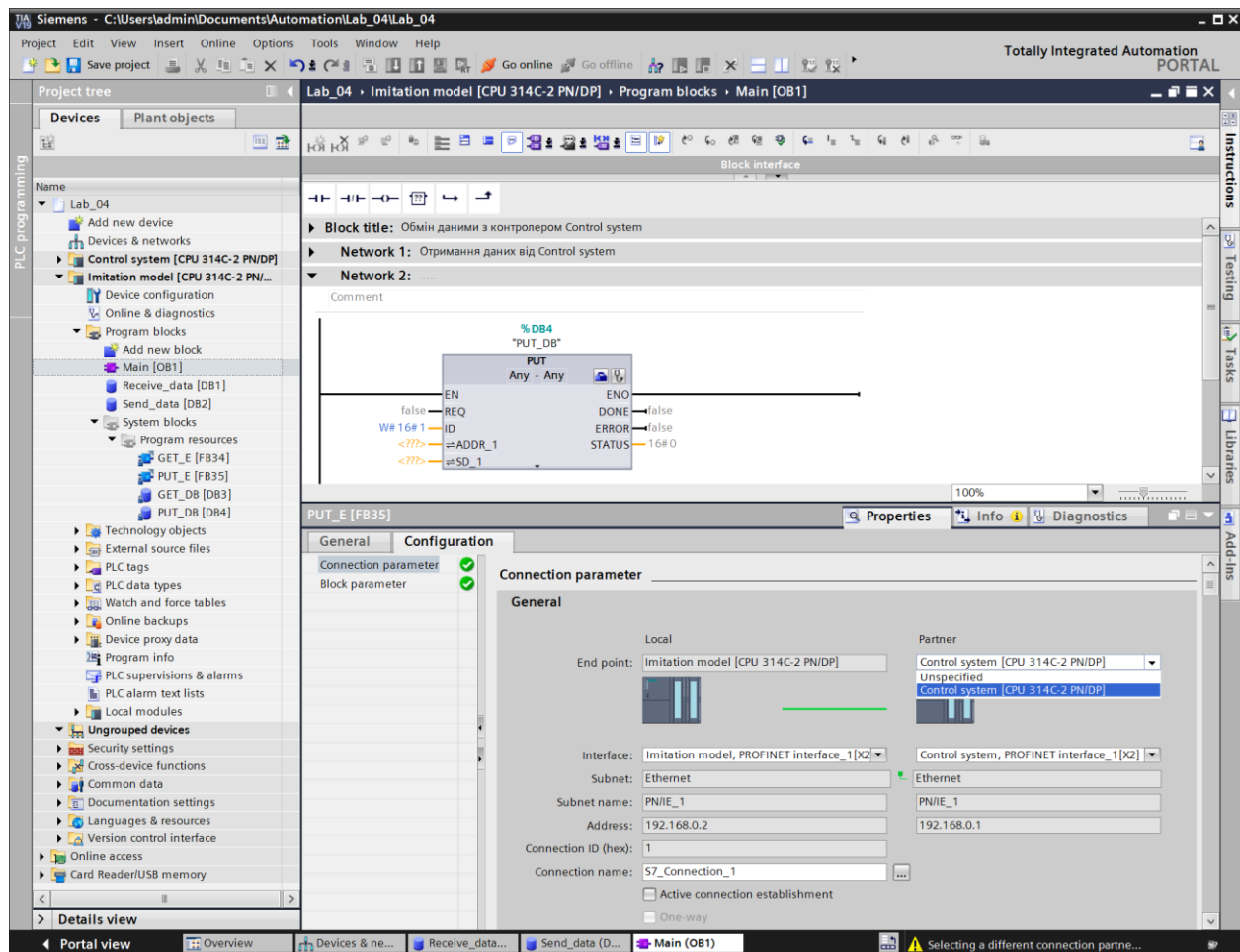


Рисунок 4.17 Параметри налаштування зв'язку блока PUT.

Подальше конфігурування блоку **PUT** виконуємо наступним чином: за аналогією з попередньою секцією **Network 1** до входу **REQ**, який запускає процес надсилання даних на віддалений контролер, підключаємо нормально замкнений контакт, прив'язаний до вихідного параметра **DONE** блоку даних **PUT_DB**, що вказує на успішну передачу даних на віддалений контролер.

Значення параметра **ADDR_1** встановлюємо рівним **P#DB1.DBX6.0 REAL 1**, що є посиланням на адресу змінної **rSimulatedProcessValue** блоку **Data_block** віддаленого контролера **Control system**.

Значенням параметра **SD_1** встановлюємо **P#DB2.DBX0.0 REAL 1**, що є посиланням на адресу змінної **rSimulatedProcessValue** блоку **Send_data** локального контролера **Imitation model**.

Для спрощення подальшої роботи з проєктом додаємо коментарі, що пояснюють призначення секції та встановлені значення параметрів блоку.

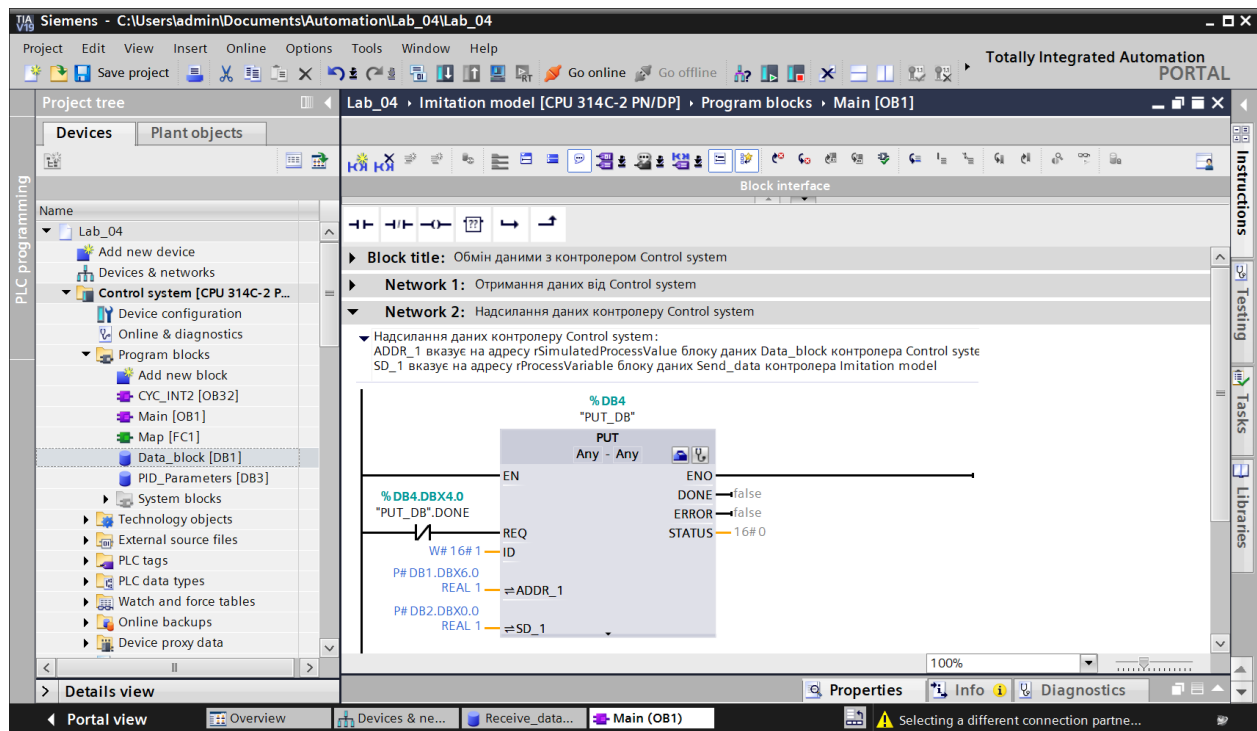


Рисунок 4.18 Конфігурація блоку PUT.

4.2.4 Розробка алгоритму імітаційної моделі об'єкта керування

Переходимо до створення алгоритму імітаційної моделі, який буде виконувати контролер **Imitation model**. Для створення імітаційної моделі об'єкта будемо використовувати бібліотеку **Objects**, що містить функціональні блоки, за допомогою яких можна змодельовати більшу частину об'єктів, які зустрічаються у промисловій автоматизації.

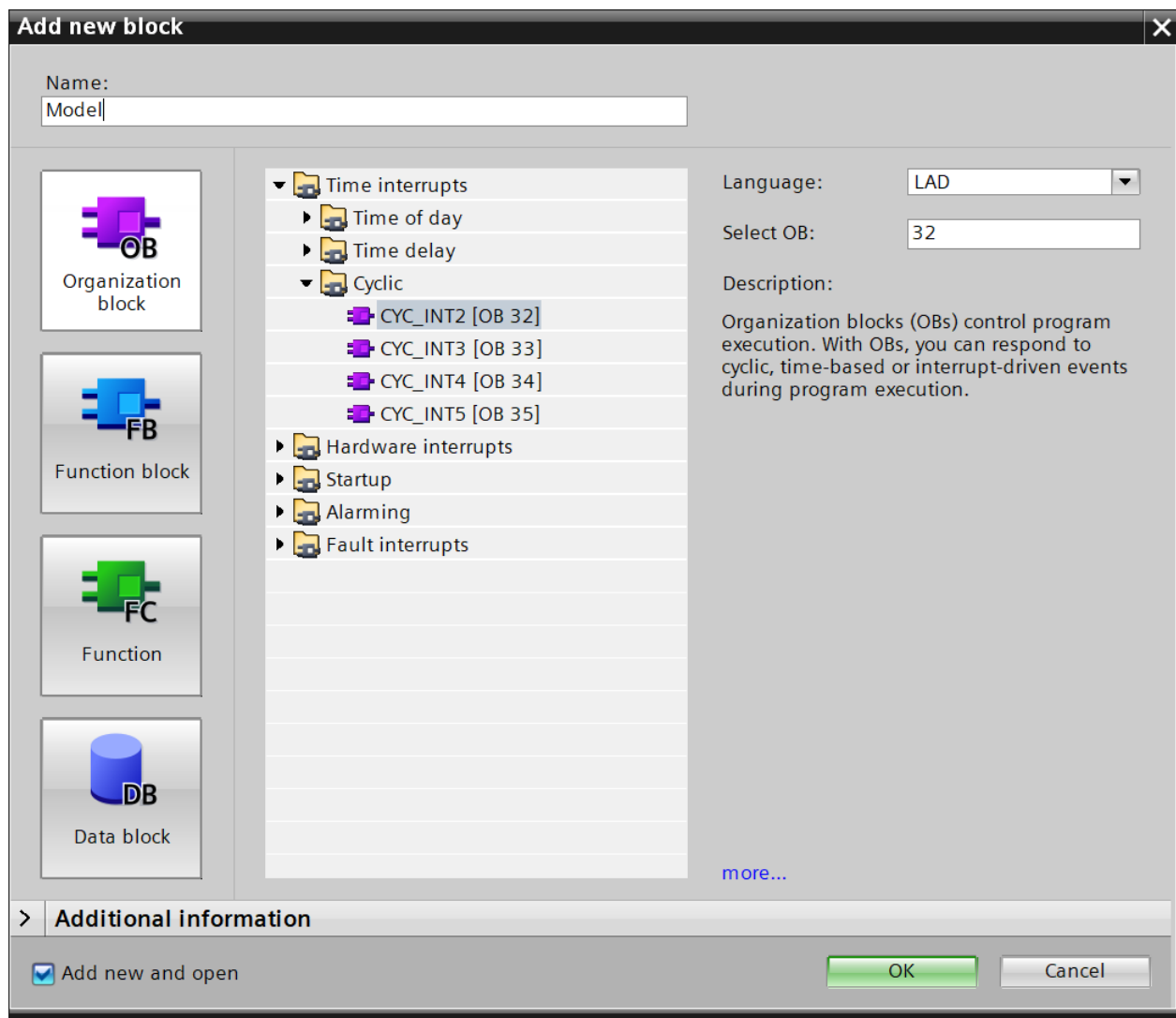


Рисунок 4.19 Додавання у проєкт циклічного організаційного блоку Model.

Всі функціональні блоки цієї бібліотеки потрібно викликати на виконання через рівні проміжки часу, для цього можна використовувати циклічний організаційний блок. Додаємо у проєкт циклічний організаційний блок OB32 та змінюємо його назву за замовченням з CYC_INT2 на Model.

Для забезпечення достатньої точності моделювання будемо використовувати крок моделювання **50 мс**. Для цього необхідно змінити у підрозділі **Cyclic interrupts** розділу переривань **Interrupts** загальних параметрів налаштування процесорного модуля контролера **Imitation model** період запуску організаційного блоку **OB32**.

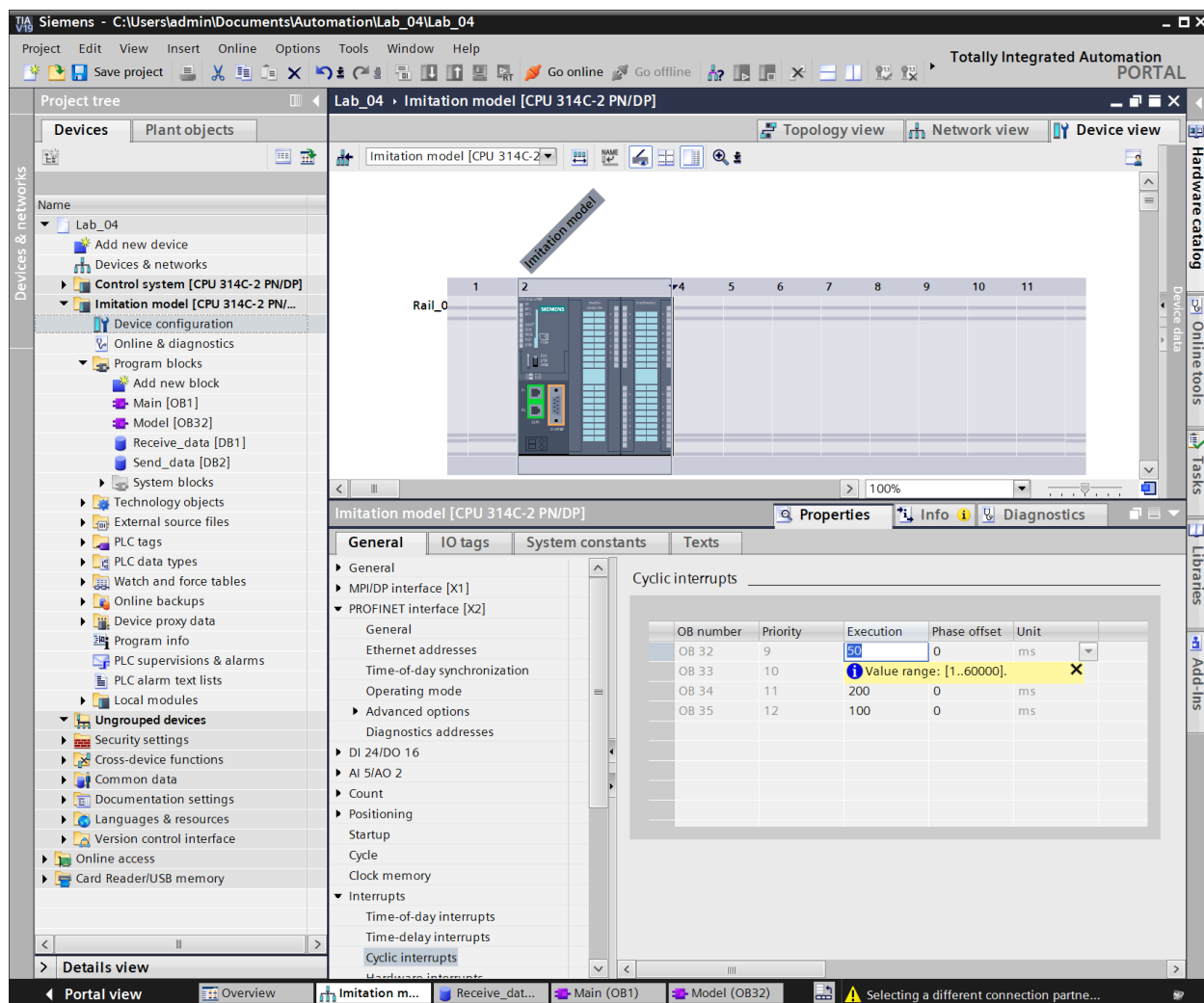


Рисунок 4.20 Зміна періоду виклику організаційного блоку OB32.

Виконане налаштування дає можливість викликати всі екземпляри функціональних блоків, які буде розміщено у даному організаційному блоці, через рівні проміжки часу, що дорівнюють встановленому значенню **50 мс**.

Для створення імітаційної моделі об'єкта керування будемо використовувати функціональні блоки **Delay** та **FirstOrderSystem**. Функціональний блок **Delay** дає можливість моделювати процеси, які мають транспортну затримку в діапазоні до **30 секунд** за умови кроку моделювання **50 мс**, а функціональний блок **FirstOrderSystem** реалізує алгоритм аперіодичної ланки першого порядку. Всі функціональні блоки бібліотеки **Objects** передбачають роботу з сигналами дійсного типу **Real**, тож їх можна підключати у будь-якій послідовності.

Приймаємо передавальну функцію моделі об'єкта керування у вигляді

$$W_{IM}(s) = \frac{K_{ob}}{T_{ob} \cdot s + 1} \cdot e^{-\tau_{Delay} \cdot s} \quad (4.1)$$

Для збереження параметрів імітаційної моделі об'єкта керування створюємо на контролері **Imitation model** блок даних **Model_parameters**.

Lab_04 ▸ Imitation model [CPU 314C-2 PN/DP] ▸ Program blocks ▸ Model_parameters [DB5]								
Model_parameters								
	Name	Data type	Offset	Start value	Retain	Visible in ...	Setpoint	Comment
1	Static							
2	rKob	Real	...	0.1	☒	☒	☐	Коефіцієнт передачі імітаційної моделі об'єкта керування
3	tTob	Time	...	T# 7S	☒	☒	☐	Стала часу імітаційної моделі об'єкта керування
4	tDelay	Time	...	T# 1S	☒	☒	☐	Транспортна затримка імітаційної моделі об'єкта керування
5	tPeriod	Time	...	T# 50MS	☒	☒	☐	Крок моделювання (період запуску організаційного блоку)
6	<Add new>							

Рисунок 4.21 Блок даних *Model_parameters* з параметрами імітаційної моделі.

У блоці **Model_parameters** створюємо такі змінні:

rKob типу **Real** зі значенням **0.1** для збереження коефіцієнту передачі імітаційної моделі об'єкта керування K_{ob}

tTob типу **Time** зі значенням **T#7S** для збереження сталої часу імітаційної моделі об'єкта керування T_{ob}

tDelay типу **Time** зі значенням **T#1S** для збереження транспортної затримки імітаційної моделі об'єкта керування t_{Delay}

tPeriod типу **Time** зі значенням **T#50MS** для збереження кроку моделювання, що дорівнює періоду запуску організаційного блоку **OB32**.

Перетягуємо функціональний блок **Delay** з бібліотеки **Objects** до секції **Network 1** організаційного блоку **Model [OB32]** для створення його екземпляра у проєкті. На вхід **rInput** цього блока будемо подавати значення **rControlAction** з блоку даних **Receive_data**, на вхід **tDelayTime** – значення **tDelay** з блоку даних **Model_parameters**, а на вхід **tPeriod** – значення періоду запуску організаційного блоку **OB32** **tPeriod** з блоку даних **Model_parameters**.

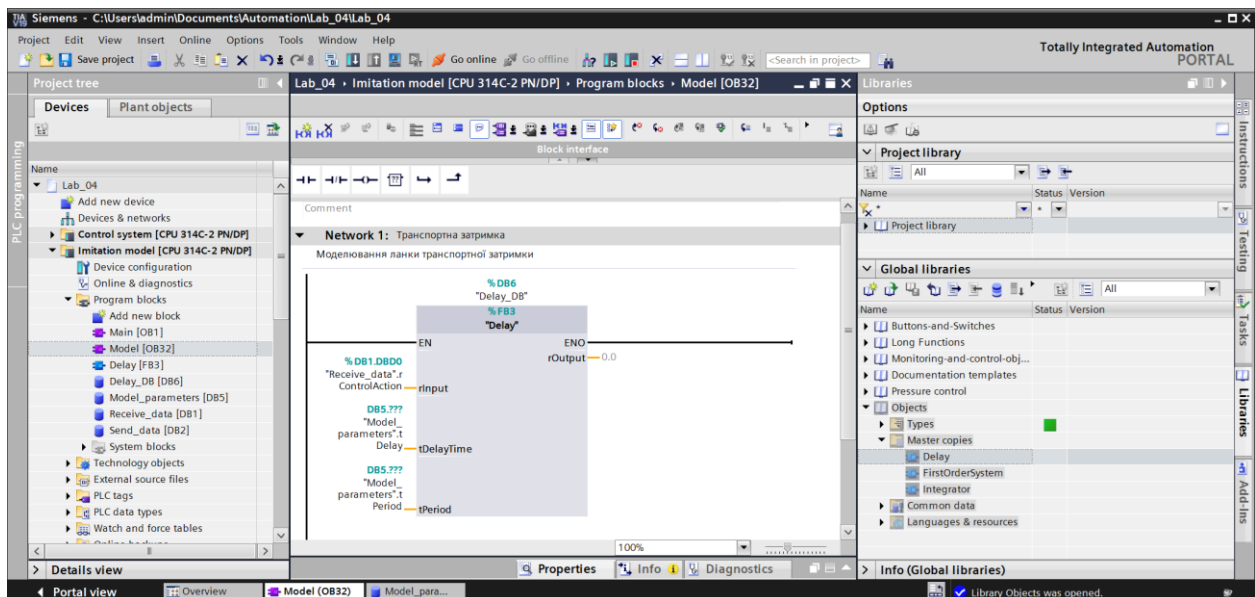


Рисунок 4.22 Моделювання транспортної затримки об'єкта.

Наступним кроком додаємо у другу секцію **Network 2** організаційного блоку **Model [OB32]** екземпляр блока **FirstOrderSystem** з бібліотеки **Objects**, за допомогою якого моделюватиметься аперіодична ланка першого порядку. На вхід **rInput** цього блока будемо подавати значення виходу **rOutput** екземпляра функціонального блоку транспортної затримки **Delay**, яке зберігається у блоці даних з назвою **Delay_DB [DB2]**, на вхід **tTime** – значення сталої часу об'єкта з параметра **tTob** з блоку даних **Model_parameters**, на вхід **rK** – значення коефіцієнту передачі об'єкта **rKob** з блоку даних **Model_parameters**, а на вхід **tPeriod** – значення періоду запуску організаційного блоку **OB32** **tPeriod** з блоку даних **Model_parameters**. Значення виходу **rOutput** функціонального блоку **FirstOrderSystem** будемо записувати у параметр **rSimulatedProcessVariable** блоку даних **Send_data** для подальшої передачі контролеру **Control system** на вхід регулятора замість сигналу від реального датчика тиску.

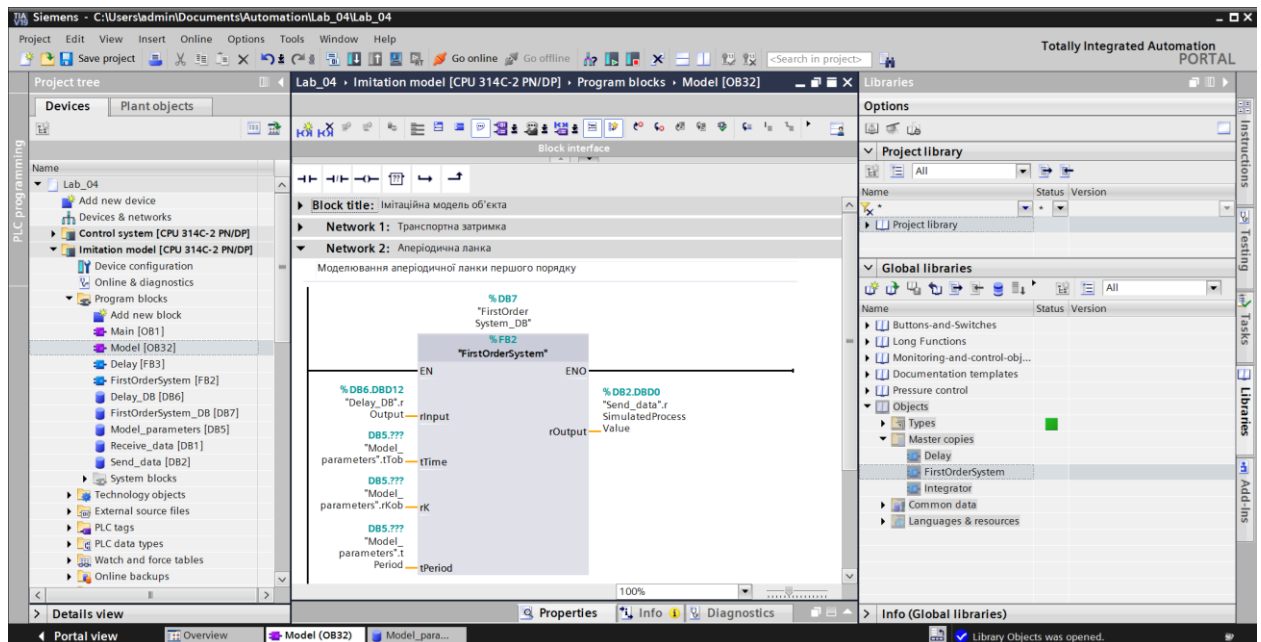


Рисунок 4.23 Моделювання аперіодичної ланки першого порядку.

Оскільки при зникненні сигналу дозволу частотного перетворювача на вхід об'єкта не подається регулююча дія, потрібно врахувати цю особливість у моделі. Для цього у третій секції **Network 3** організаційного блоку **Main [OB1]** додаємо інструкцію запису нульового значення до параметра **rControlAction** блоку даних **Receive_data** за умови зникнення сигналу дозволу.

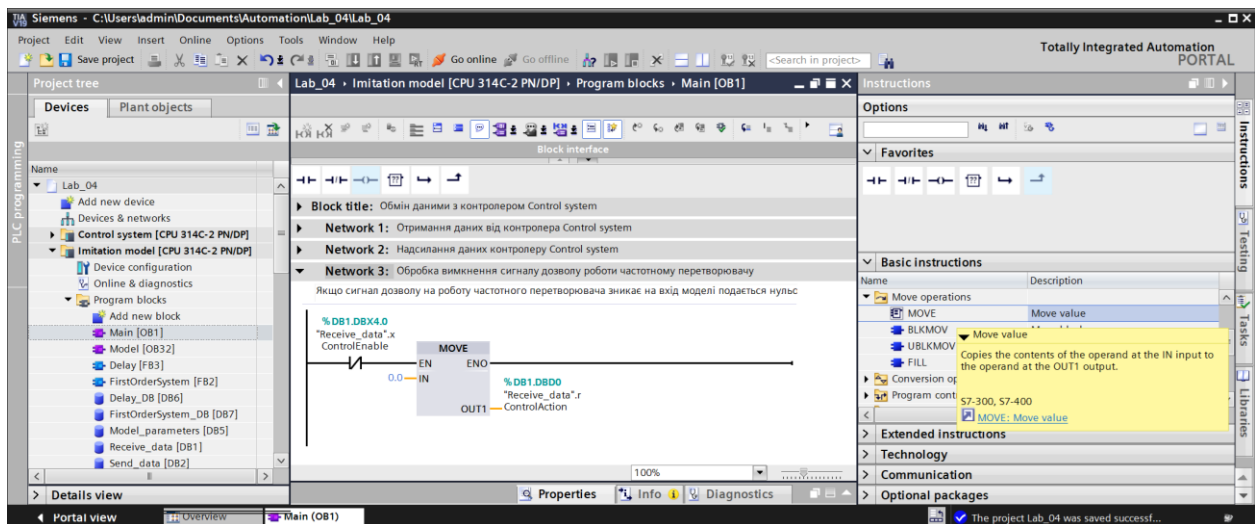


Рисунок 4.24 Обробка у імітаційній моделі команди вимкнення частотного перетворювача.

4.2.5 Проведення імітаційного моделювання системи регулювання

Після закінчення написання алгоритму імітаційної моделі об'єкта керування переходимо до перевірки правильності його роботи та налаштування параметрів ПІД-регулятора.

Для цього запускаємо симулятор контролера Control system, переводимо його в режим **RUN-P** та завантажуюємо програму. Після того, як програма завантажена на симулятор контролера **Control system**, аналогічно запускаємо симуляцію контролера **Imitation model**. Таким чином, одночасно має працювати два екземпляра симулятора **S7-PLCSIM**.

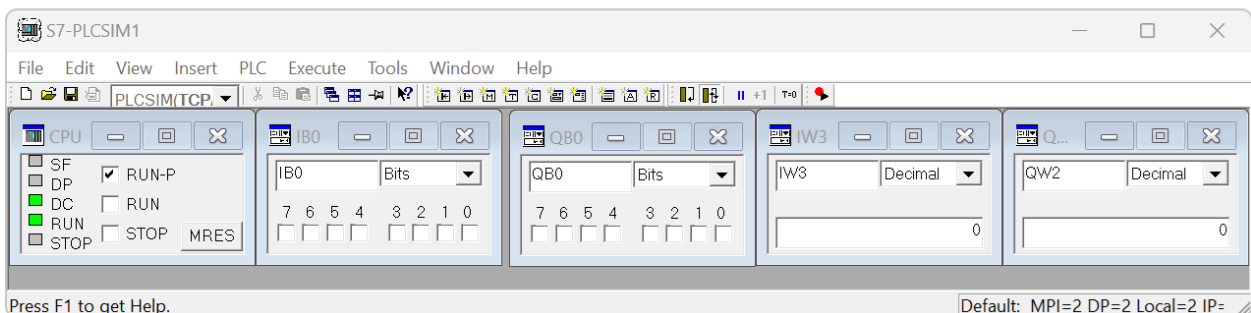
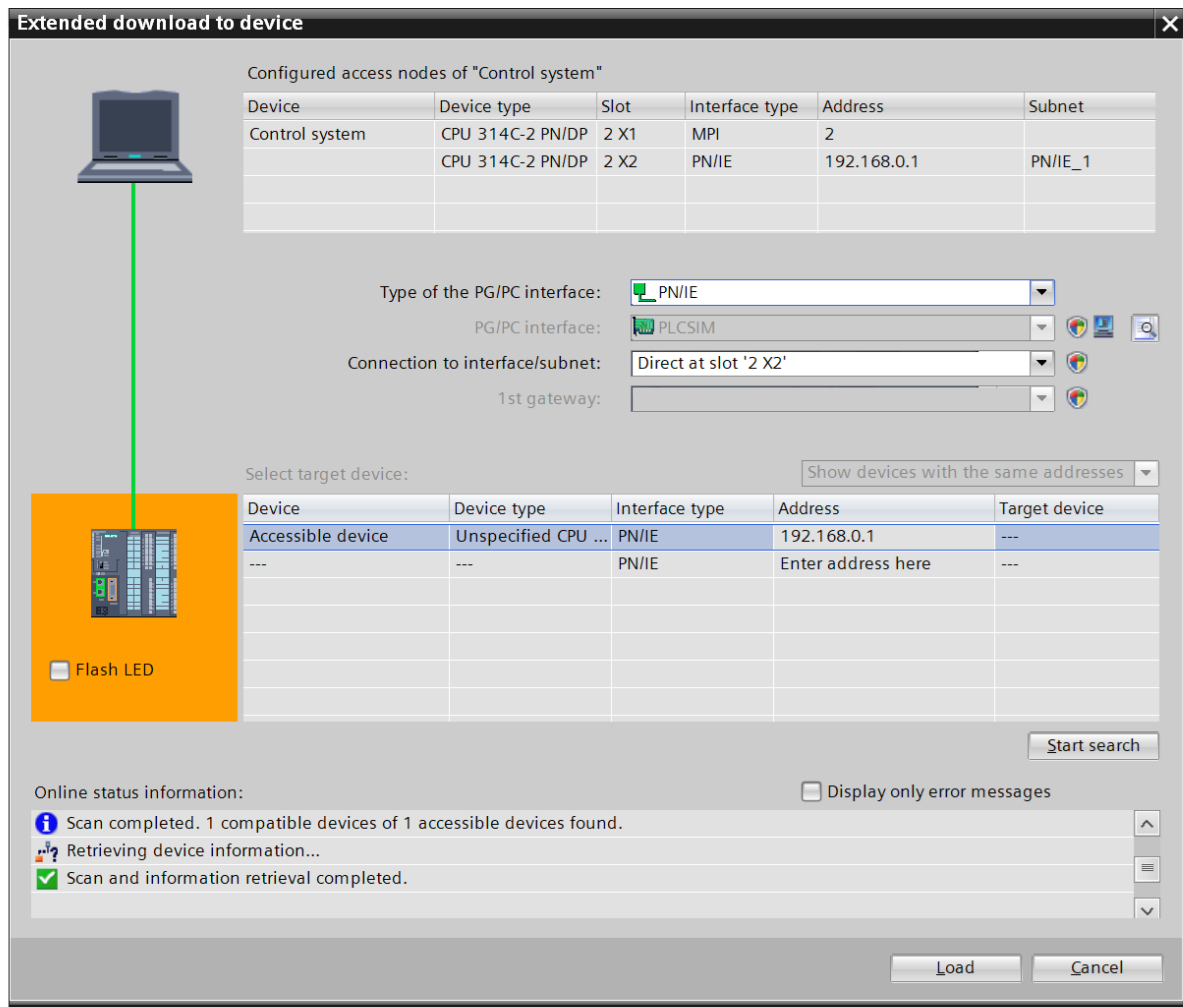


Рисунок 4.25 Запуск симулятора контролера Control system.

Перевіряємо роботу системи в ручному режимі. На симуляторі контролера **Control system** імітуємо сигнал від термореле захисту двигуна від перегріву дискретним входом контролера **Control system** за допомогою дискретного входу контролера **TK1 (%I0.4)**, що відповідає відсутності перегріву. Імітуємо дискретним входом контролера **SA1_2 (%I0.1)** переведення системи в ручний режим, а ввімкненням входу **SB1 (%I0.2)** команду пуску насоса в ручному режимі.

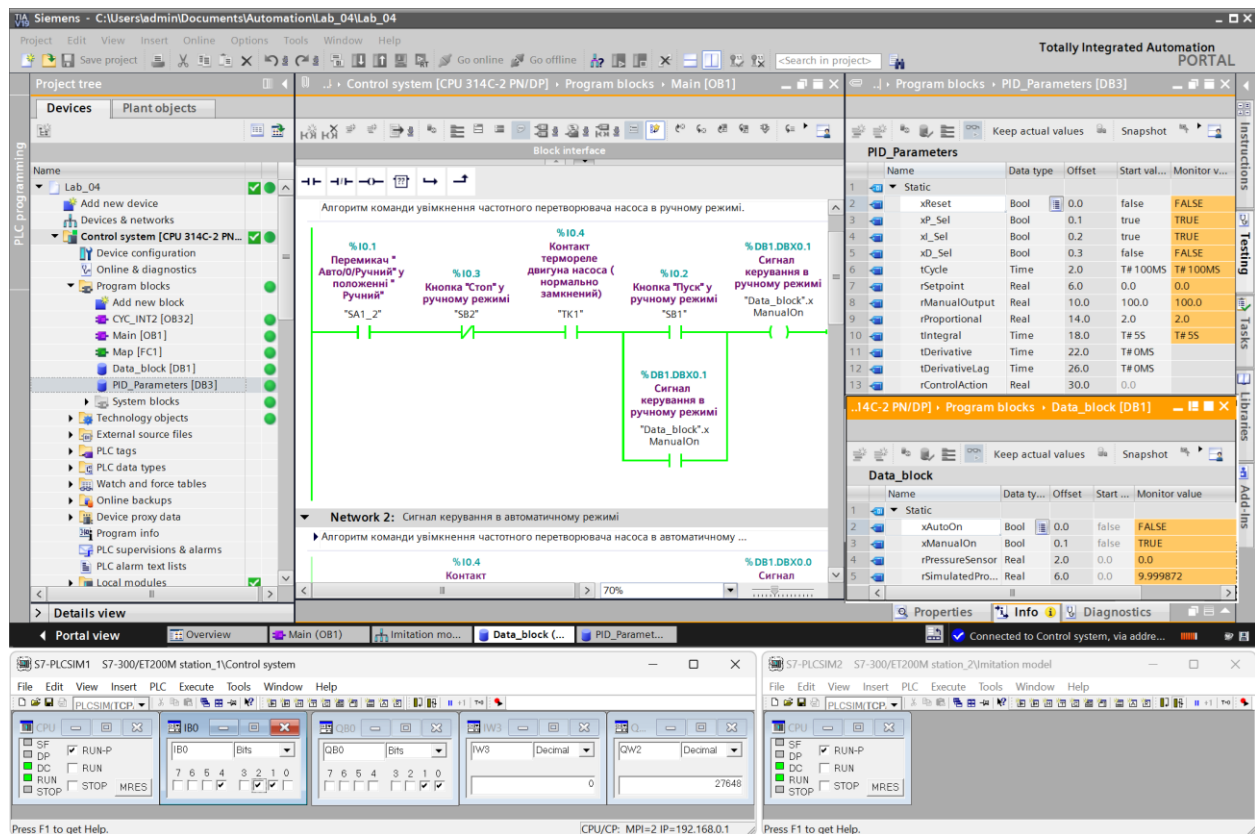


Рисунок 4.26 Перевірка роботи системи в ручному режимі.

Це має перевести регулятор в ручний режим та подати на його вихід значення **100.0**, а також ввімкнути дискретні виходи контролера **Control system DO1** (**%Q0.0**) та **HL1** (**%Q0.1**). Вимкнення сигналу від кнопки **SB1** (**%I0.2**) не повинно вимикати дискретні виходи контролера. Значення **100.0** має передатись на вхід імітаційної моделі через параметр **rControlAction** і значення виходу імітаційної моделі **rSimulatedProcessValue** поступово має зрости до максимального **10.0**. Імітація сигналу від кнопки подачі команди зупинки насоса в ручному режимі **SB2** (**%I0.3**) має відключати частотний перетворювач (дискретні виходи **DO1** та **HL1** вимикаються), значення **rControlAction** у блоці даних **Receive_data** при цьому стає рівним нулю і значення **rSimulatedProcessValue** поступово має зменшуватись до нуля.

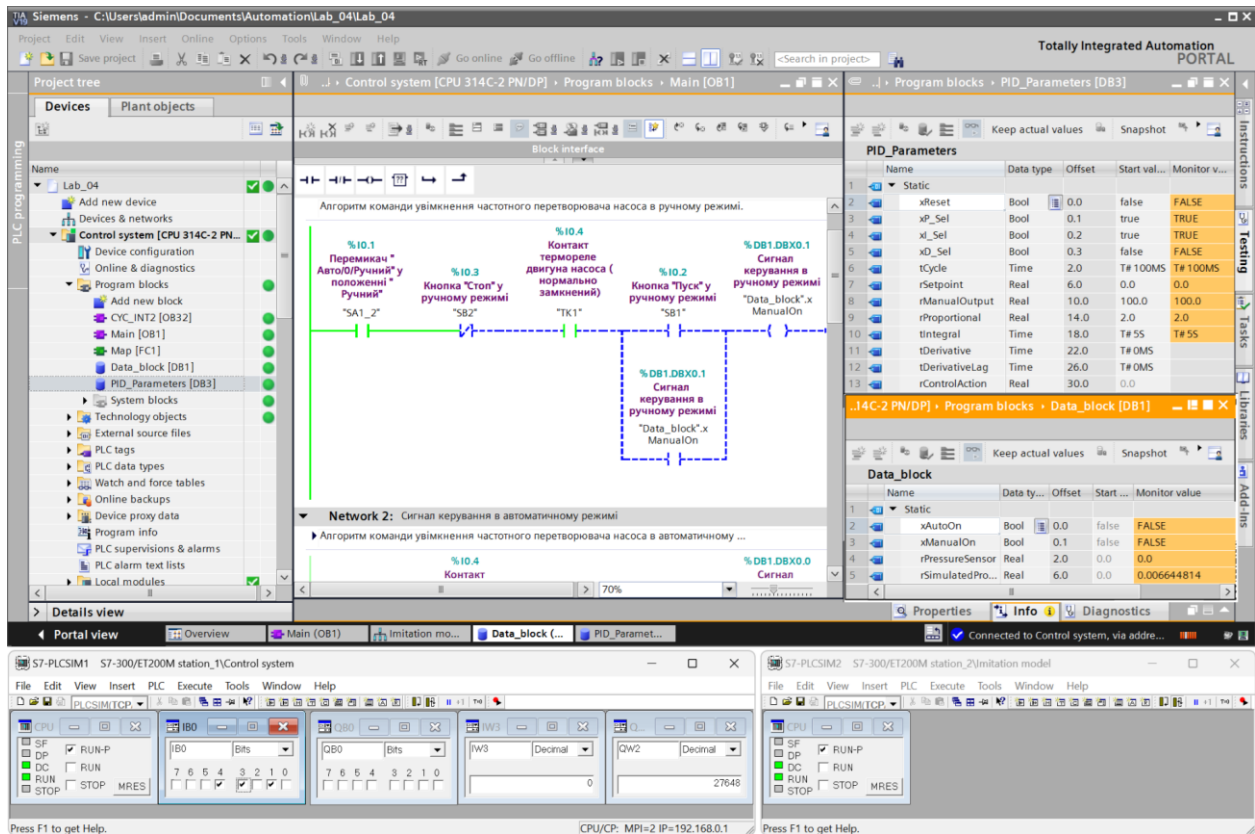


Рисунок 4.27 Імітація зупинки насоса в ручному режимі.

Для перевірки правильності роботи системи в автоматичному режимі необхідно ввімкненням дискретного входу контролера **Control system TK1** (%I0.4) імітувати відсутність перегріву двигуна насоса та перевести систему в автоматичний режим ввімкненням входу **SA1_1** (%I0.0). Оскільки за замовченням завдання регулятора, що задається параметром **rSetpoint** блоку даних **PID_Parameters**, дорівнює нулю, регулятор буде видавати нульову регулюючу дію. Змінити завдання можна безпосередньо у блоці даних **PID_Parameters** у режимі моніторингу – для цього потрібно обрати в контекстному меню параметра **rSetpoint** команду **Modify operand...**

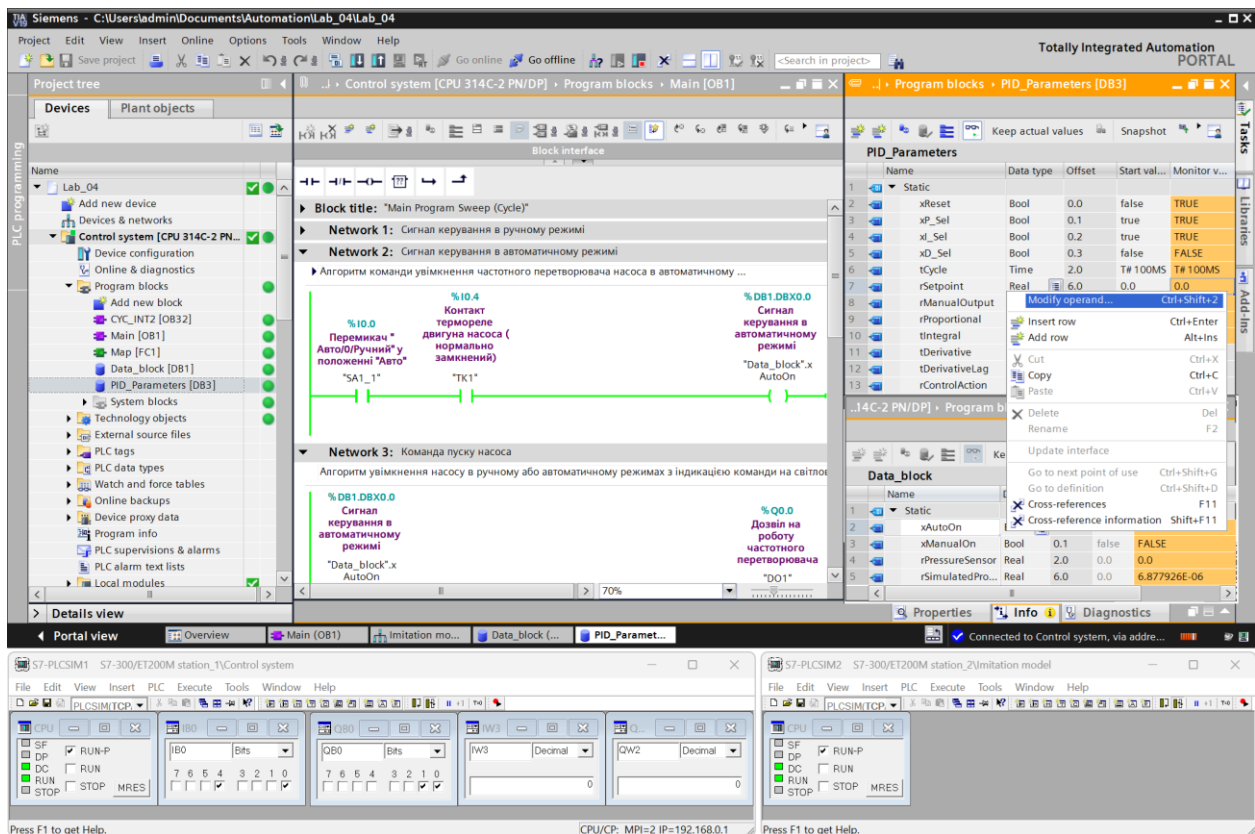


Рисунок 4.28 Перевірка роботи системи в автоматичному режимі.

У діалоговому вікні редагування параметра **rSetpoint** необхідно ввести нове значення завдання в робочому діапазоні об'єкта (від **0.0** до **10.0**). Спробуємо встановити завдання регулятора тиску рівним **8.0** та перевірити правильність функціонування регулятора та імітаційної моделі в автоматичному режимі.

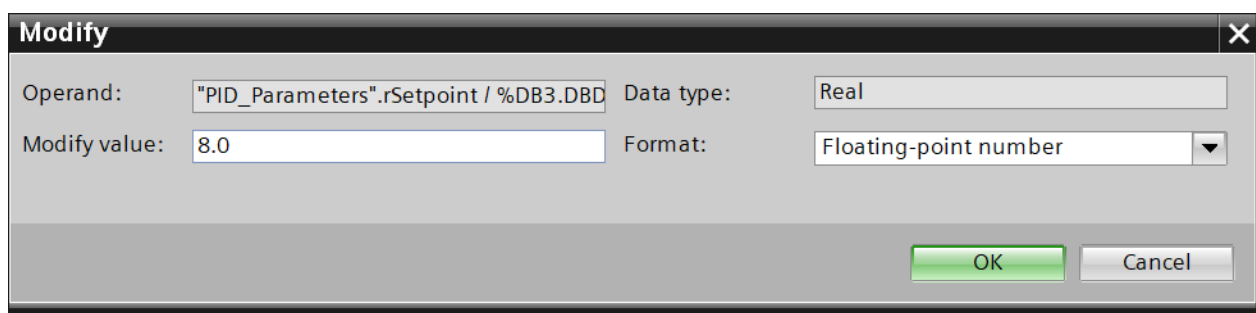


Рисунок 4.29 Зміна завдання регулятора в автоматичному режимі.

В автоматичному режимі вихід регулятора залежить від параметрів настройки ПІД-регулятора **CONT_C** та різниці між заданим значенням тиску **rSetpoint**, яке подається на вхід регулятора **SP_INT** і значенням **rSimulatedProcessVariable**, що приходить на вхід **PV_IN** від алгоритму імітаційної моделі, який реалізований на контролері **Imitation model**. За вірних налаштувань змінна **rSimulatedProcessVariable** має поступово вийти на значення, близьке до заданого.

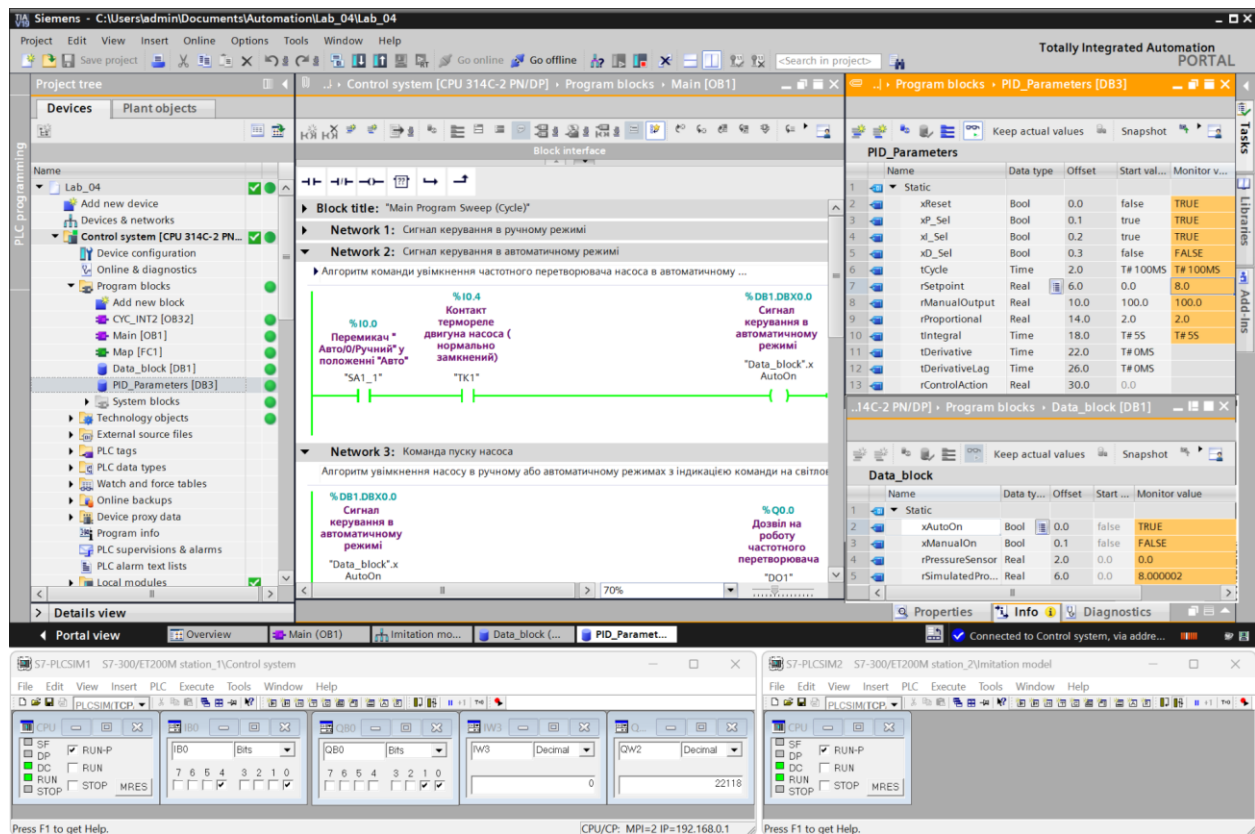


Рисунок 4.30 Перевірка відпрацювання регулятором заданого значення в автоматичному режимі.

За відсутності графіка перехідного процесу складно зробити висновок щодо доцільності зміни параметрів настройки регулятора та забезпечення необхідних показників якості регулювання. На етапі попереднього налаштування регулятора з використанням імітаційної моделі об'єкта можна оцінити якість процесу регулювання та необхідність корегування параметрів настройки регулятора за допомогою інструмента введення в експлуатацію регулятора **CONT_C**. Відкрити вікно введення в експлуатацію регулятора можна за допомогою кнопки з зображенням інструментів у правому верхньому куті блока регулятора **CONT_C**.

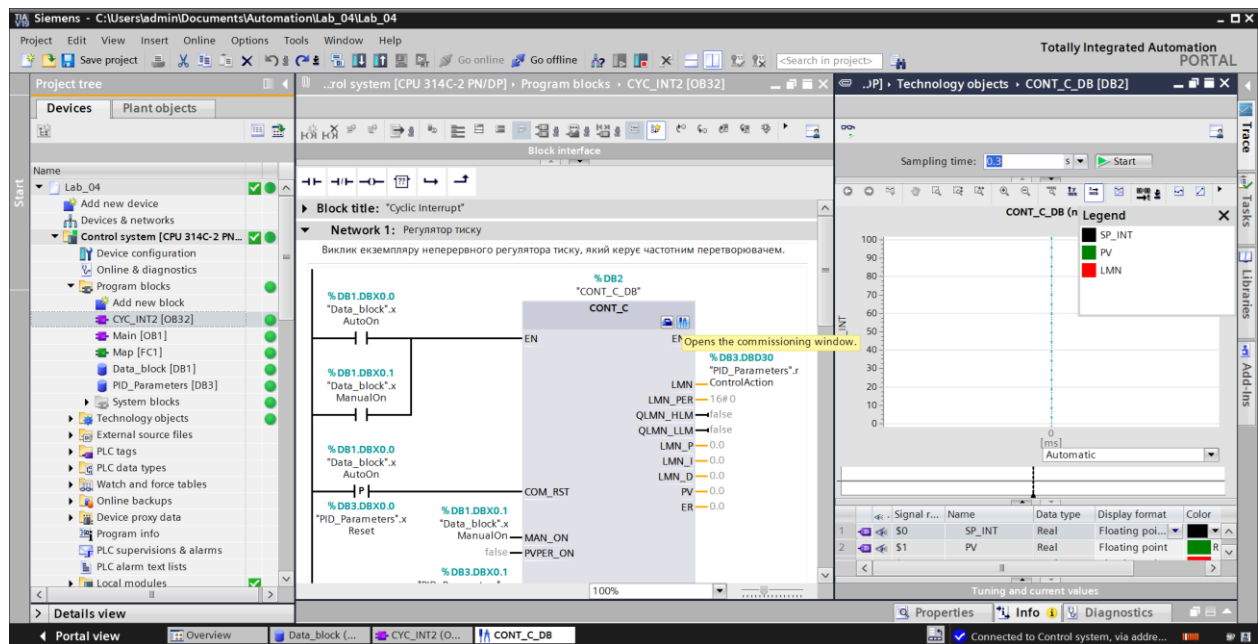


Рисунок 4.31 Відкриття вікна введення регулятора в експлуатацію.

Для побудови графіка перехідного процесу потрібно розпочати процес введення регулятора в експлуатацію кнопкою **Start**.

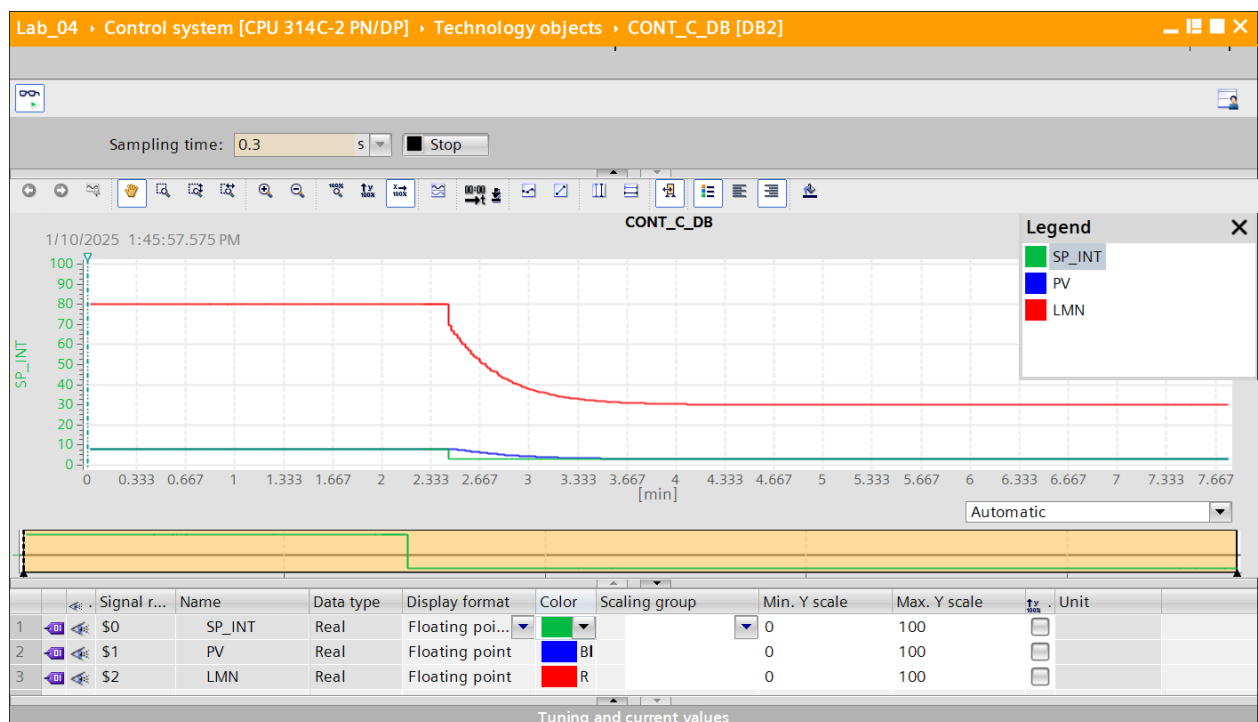


Рисунок 4.32 Запуск режиму введення регулятора в експлуатацію.

За замовчанням у вікні відображаються графіки завдання, регулюючої дії та регульованої величини. У таблиці під графіком розташовані налаштування, які дають можливість за необхідності приховати непотрібні графіки, змінити їх колір та масштаб побудови. Якщо спробувати отримати графік процесу виходу регульованого параметра на задане значення з початковими параметрами

настройки регулятора, можна помітити, що перехідний процес є доволі тривалий – регулятор дуже довго виводить об’єкт на задане значення. Підбором більш вдалих параметрів налаштування **rProportional** та **tIntegral**, а також, за необхідності, підключенням диференційної частини регулятора сигналом **xD_Sel** та параметрами її налаштування **tDerivative** і **tDerivativeLag** можна досягти значного зменшення тривалості перехідного процесу без значного погіршення інших показників якості регулювання.

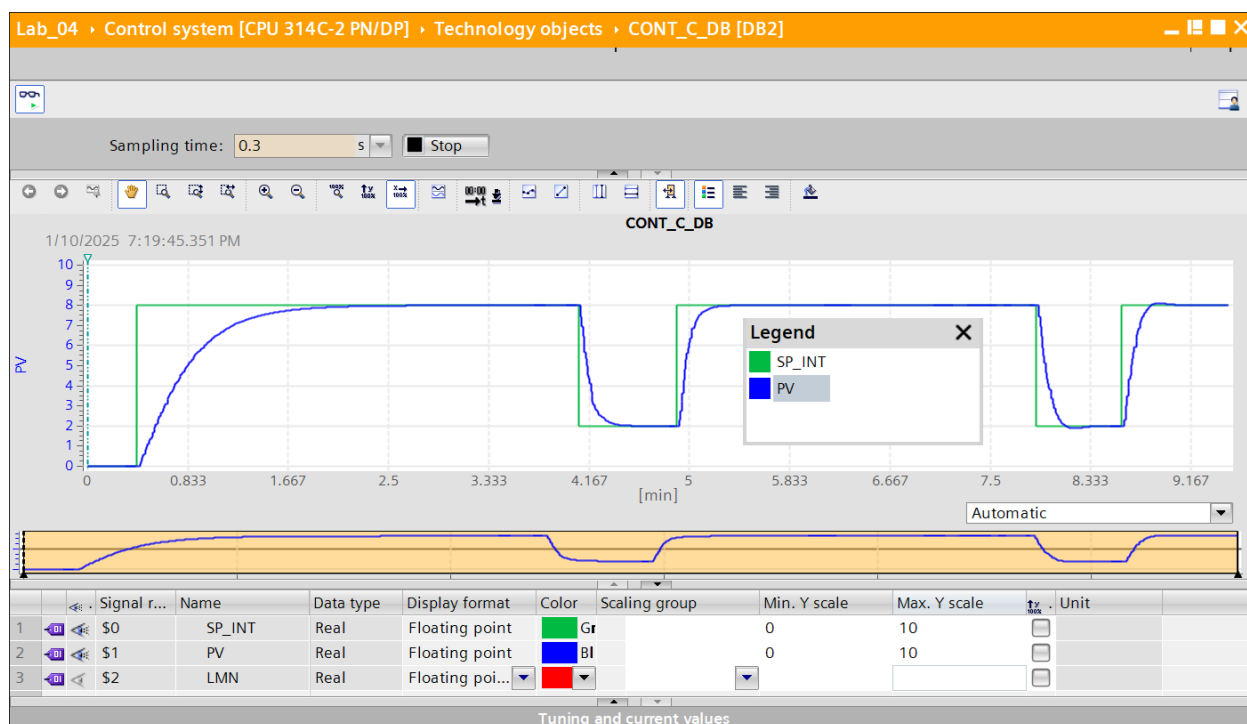


Рисунок 4.33 Результат налаштування параметрів регулятора CONT_C.

Після того, як визначені оптимальні параметри налаштування регулятора, їх можна зберегти за допомогою кнопки панелі інструментів блоку даних Snapshot of the actual values з зображенням фотокамери.

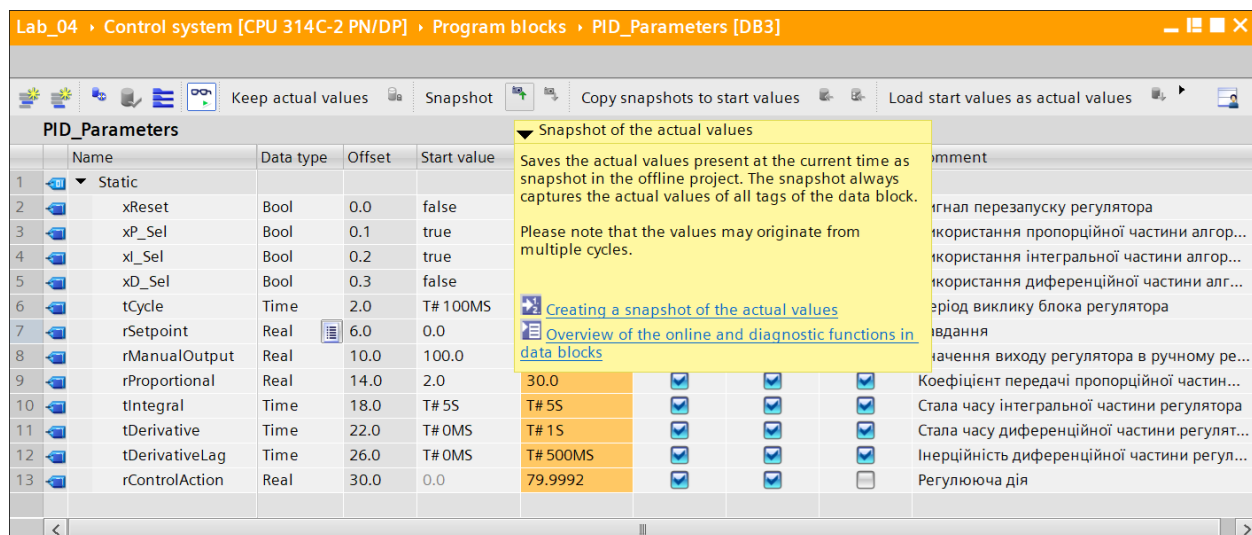


Рисунок 4.34 Збереження поточних значень у знімку.

Збережені значення можна встановити як значення за замовченням за допомогою однієї з кнопок панелі інструментів **Copy snapshots to start values** – або всі збережені значення (кнопка **All values**), або тільки ті значення, для яких встановлена позначка **Setpoint** (кнопка **Only setpoints**). Якщо зберегти вказані зміни у проєкті та записати їх на контролер, при ввімкненні контролера будуть встановлені оптимальні параметри настройки.

Варто пам'ятати, що після проведення налаштування регулятора методом імітаційного моделювання потрібно видалити частини алгоритму контролера Control system, що були зроблені для обміну даними з контролером Imitation model, тобто замість параметра rSimulatedProcessVariable блока даних Data_block контролера Control system, який подається на вхід функціонального блоку регулятора CONT_C, записати масштабоване значення від датчика тиску PT1, яке зберігається у змінній rPressureSensor блока даних Data_block контролера Control system.

Після внесення змін можна записати оновлену версію алгоритму з визначеними оптимальними параметрами настройки регулятора в бібліотеку Pressure control.

4.3 Контрольні завдання

1. Підібрати параметри налаштування регулятора CONT_C для покращення перехідного процесу.
2. Порівняти час виходу регульованого параметра на задане значення у ручному та автоматичному режимах.
3. Виконати перевірку алгоритму захисту двигуна насоса від перегріву.
4. Змінити алгоритм контролера Control system таким чином, щоб після перегріву двигуна його можна було запустити тільки після скидання

помилки перегріву додатковою кнопкою SB3, підключеною до шостого дискретного входу контролера.

5 Лабораторна робота №5.

Мета роботи: Навчитись використовувати в проєктах TIA Portal засоби людино-машинного інтерфейсу.

5.1 Завдання

1. Налаштувати підключення панелі до ПЛК
2. Відобразити на екрані основні технологічні параметри системи регулювання тиску
3. Реалізувати можливість пуску та зупинки насосу в ручному режимі за допомогою панелі оператора
4. Відобразити на екрані графіки перехідних процесів
5. Додати можливість налаштування параметрів регулятора за допомогою панелі оператора
6. Перевірити коректність роботи системи з панеллю оператора

5.2 Порядок виконання роботи

5.2.1 Налаштування підключення панелі оператора до ПЛК

Використання у проєктах засобів автоматизації без реалізації людино-машинного інтерфейсу зменшує як вартість обладнання, так і загальну вартість розробки та монтажу системи автоматизації. Натомість, забезпечення персоналу засобами контролю, аналізу функціонування та можливості оперативного впливу може підвищити ефективність роботи обладнання та зменшити час реагування оператора на погіршення показників якості роботи системи.

Залежно від складності системи автоматизації для розробки людино-машинного інтерфейсу можуть бути використані різні інструменти. Для технологічно простих установок часто буває достатньо забезпечити мінімальні засоби місцевого оперативного керування використанням недорогих панелей оператора, встановлених безпосередньо за місцем встановлення технологічного обладнання. Зі збільшенням кількості параметрів, які повинен контролювати оператор, доводиться обирати панелі оператора більшого розміру, які дозволяють відобразити оператору більше інформації. Для складних систем, які розподілені на великій площі та мають велику кількість технологічних параметрів та органів керування необхідно використовувати **SCADA (Supervisory Control And Data Aquisition)** системи, які дозволяють забезпечити оператору безпечні умови роботи, можливість контролю перебігу всього технологічного процесу, аналізу архівних даних та автоматичного генерування звітів з роботи обладнання.

TIA Portal надає розробнику можливість використовувати для побудови людино-машинного інтерфейсу як панелі оператора, так і SCADA-програми, які можуть запускатись на спеціалізованому обладнанні Siemens, так і на робочих станціях. Для даної роботи будемо використовувати проєкт системи регулювання тиску з імітаційною моделлю. Зберігаємо його з новою назвою **Lab_05**. Для реалізації людино-машинного інтерфейсу буде використана панель оператора **Siemens TP900Comfort**.

Для того, щоб додати у проєкт панель оператора використовуємо пункт Add new device дерева проєкта **TIA Portal**. У вікні додавання нового пристрою обираємо тип пристрою **HMI**, а у переліку HMI пристроїв – **Simatic Comfort Panel → 9” Display → TP900 Comfort → 6AV2 124-0JC01-0AX0**.

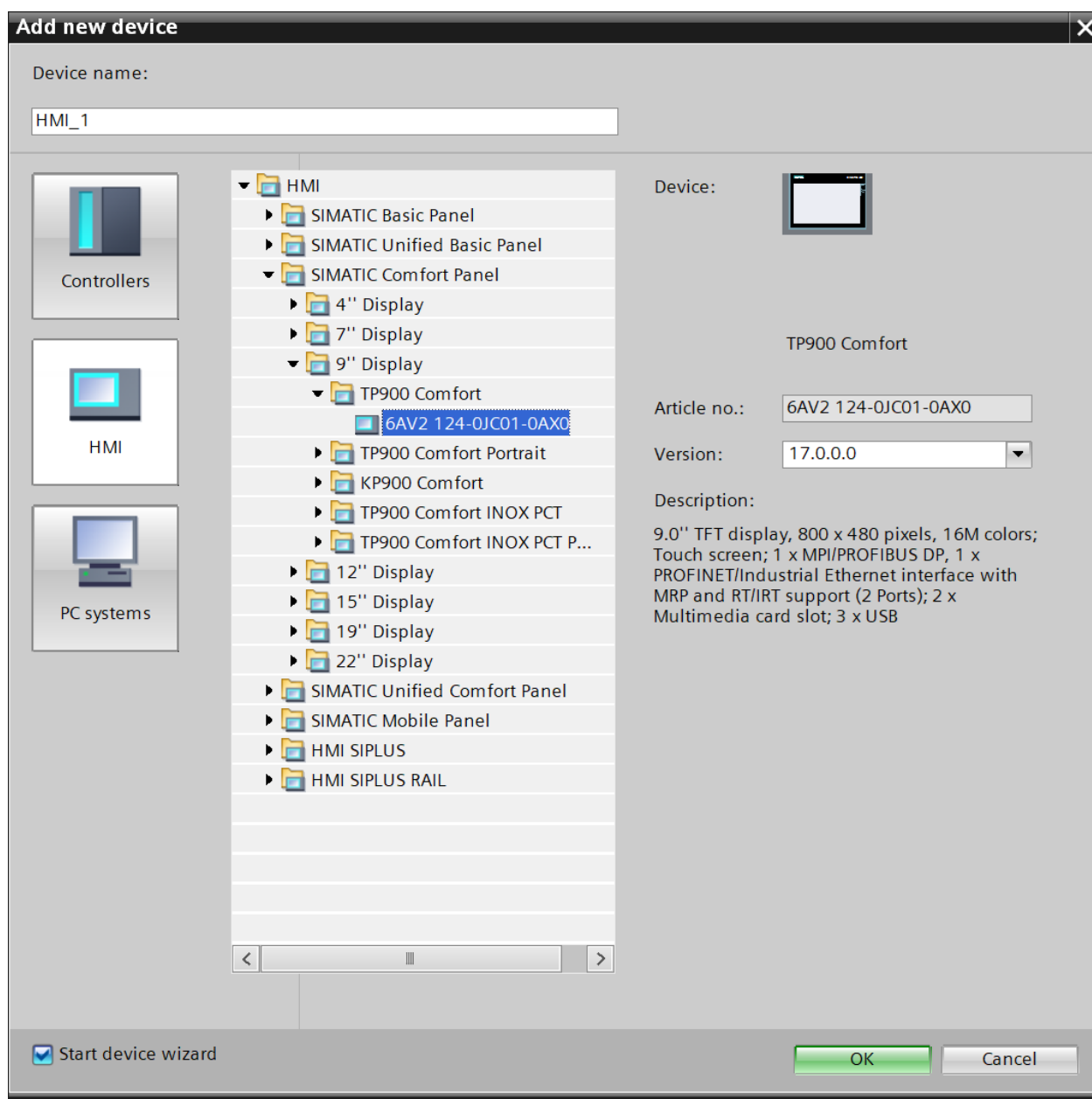


Рисунок 5.1 Додавання у проєкт панелі оператора.

З наведених у вікні технічних характеристик видно, що обрана панель оператора має розмір діагоналі екрана 9" (22,86 см) та розподільчу здатність екрана 800x480 точок, оператор може вводити команди за допомогою сенсорної панелі, а обмін даними з контролером може бути реалізований через наявний інтерфейс PROFINET.

Налаштування панелі оператора може бути виконане за допомогою майстра налаштування HMI пристрою або вручну. Після додавання панелі оператора відкривається вікно майстра, в якому потрібно обрати контролер, з яким потрібно встановити зв'язок.

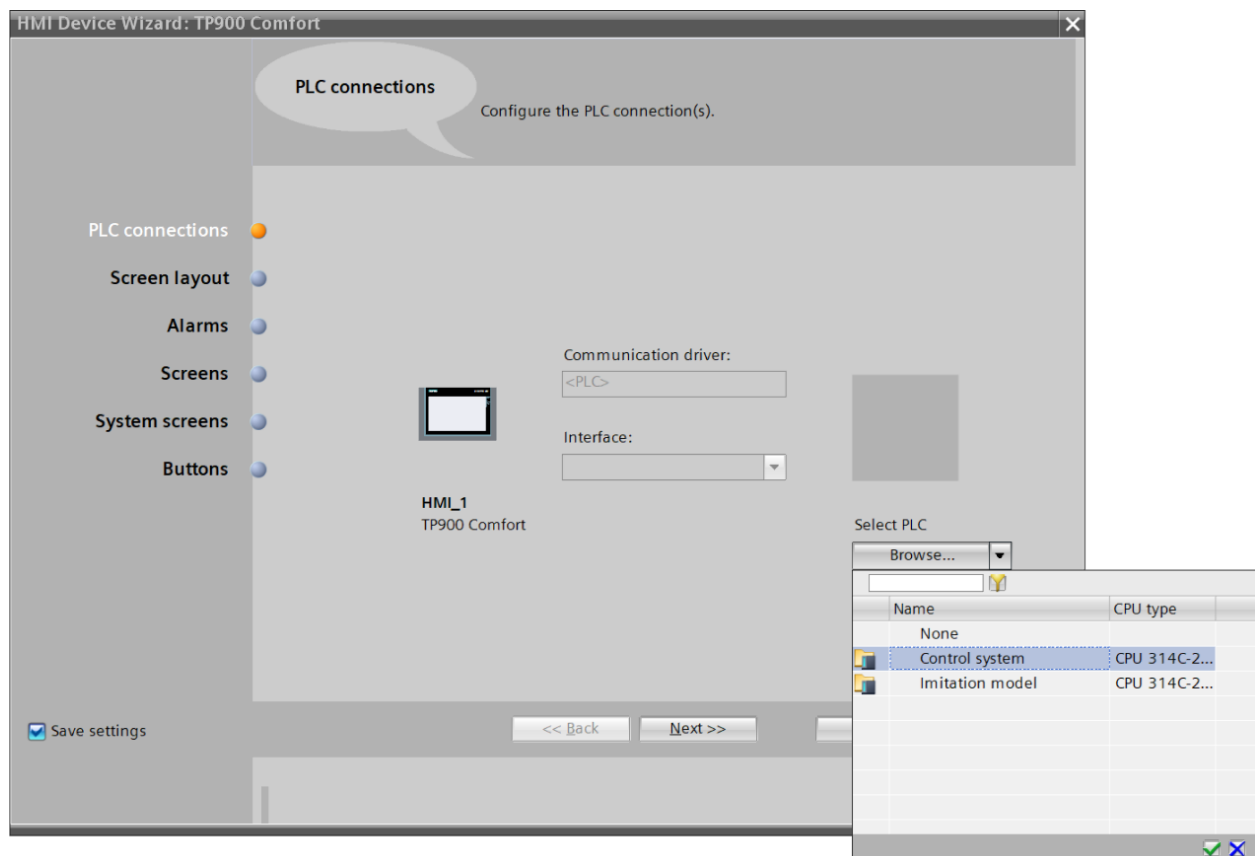


Рисунок 5.2 Встановлення підключення HMI панелі до ПЛК.

У списку вибору ПЛК обираємо контролер **Control system**, вибір потрібно підтвердити кнопкою з зображенням зеленої позначки.

Обмін даними між панеллю оператора та програмованим логічним контролером **Control system** буде відбуватись через інтерфейс **ETHERNET**, якщо підключення між HMI панеллю та контролером здійснюється за допомогою майстра, то після підтвердження вибору контролера у проєкті створюється підключення **HMI connection** з назвою **HMI_Connection_1**, а панель оператора отримає IP адресу, яка знаходиться в одній підмережі з контролером та не використовується у проєкті. Якщо передбачається підключення контролера та HMI панелі до локальної мережі підприємства, всі IP адреси та інші параметри налаштування мережі мають бути узгоджені з адміністратором мережі підприємства.

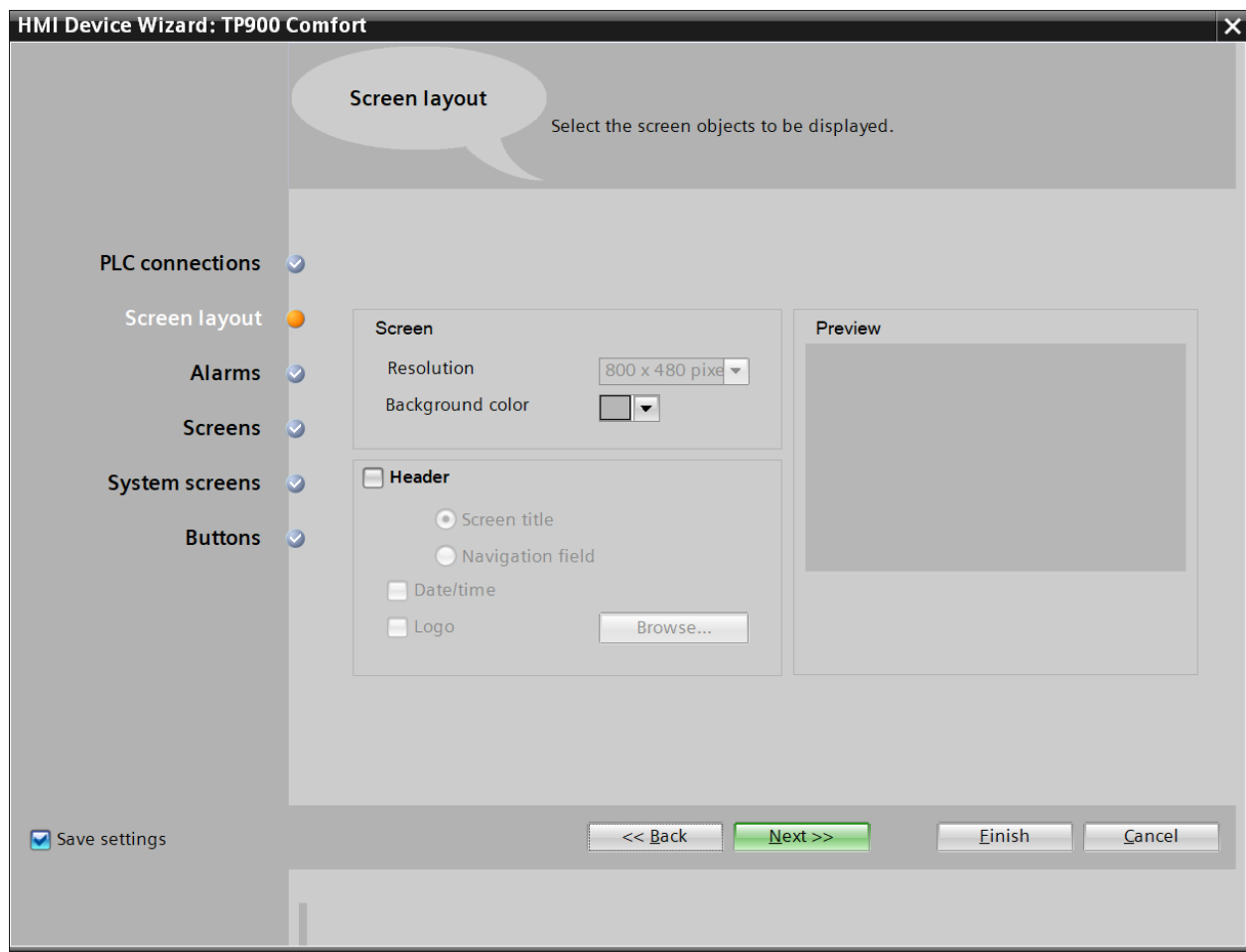


Рисунок 5.3 Налаштування макета екрана.

Наступне вікно майстра налаштування НМІ панелі пропонує визначити макет екрана інтерфейсу оператора, тобто елементи, що будуть спільними для всіх екранів людино-машинного інтерфейсу: змінити колір фону робочої області, відобразити заголовок вікна з його назвою або полем навігації, за необхідності у заголовку можна відобразити поточний час панелі оператора та логотип підприємства у одному з графічних форматів: **.bmp**, **.jpg**, **.gif** або **.png**. Якщо ці елементи не є необхідними, можна залишити робоче поле пустим, за необхідності всі зазначені елементи можна розмістити на екрані пізніше у режимі конструктора. Переходимо до наступного кроку налаштування.

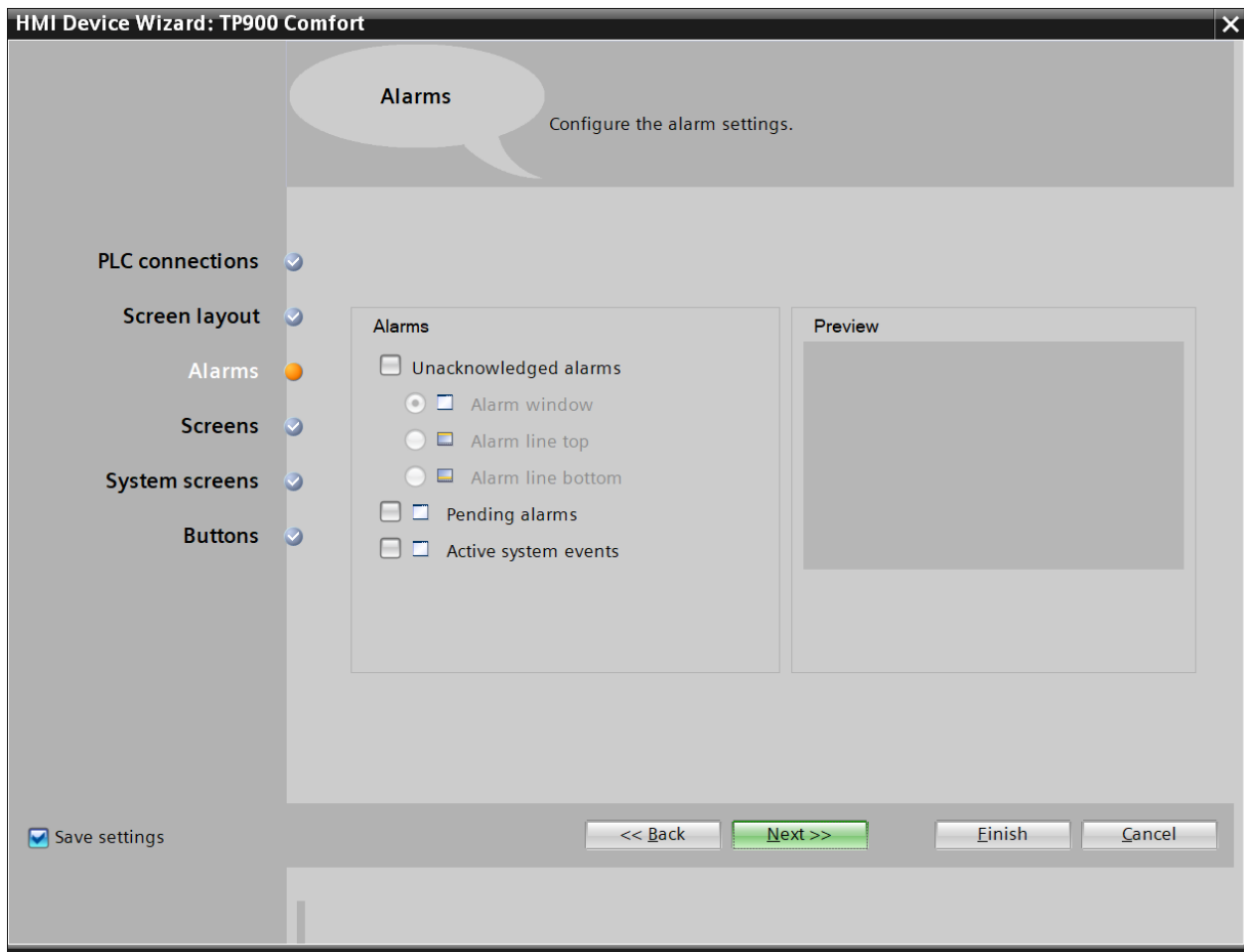


Рисунок 5.4 Вікно налаштування відображення алармів.

Наступне вікно майстра налаштування HMI панелі пропонує вказати типи повідомлень, які потрібно відображати та спосіб їх відображення. Так, пункт **Unacknowledged alarms** дозволяє вивести на екран непідтверджені тривоги одним із трьох способів: у вигляді окремого вікна (**Alarm window**), у вигляді строки у верхній частині екрана (**Alarm line top**) або у вигляді строки у нижній частині екрана (**Alarm line bottom**). Пункт **Pending alarms** дозволяє ввімкнути відображення поточних тривог, а **Active system events** – відображення активних системних повідомлень.

Залишаємо у вікні налаштування тривог всі відмітки знятими та переходимо до наступного кроку майстра налаштувань, який визначає ієрархічну структуру екранів людино-машинного інтерфейсу, тобто кількість екранів та порядок навігації між ними. Чим більше інформації та органів керування потрібно надати оператору, тим більше знадобиться екранів. При визначенні кількості та порядку навігації між екранів необхідно зважати на зручність користування та намагатись мінімізувати кількість необхідних дій для виконання найбільш частих операцій. За необхідності всі зазначені на цьому кроці елементи також можна налаштувати згодом у режимі конструктора.

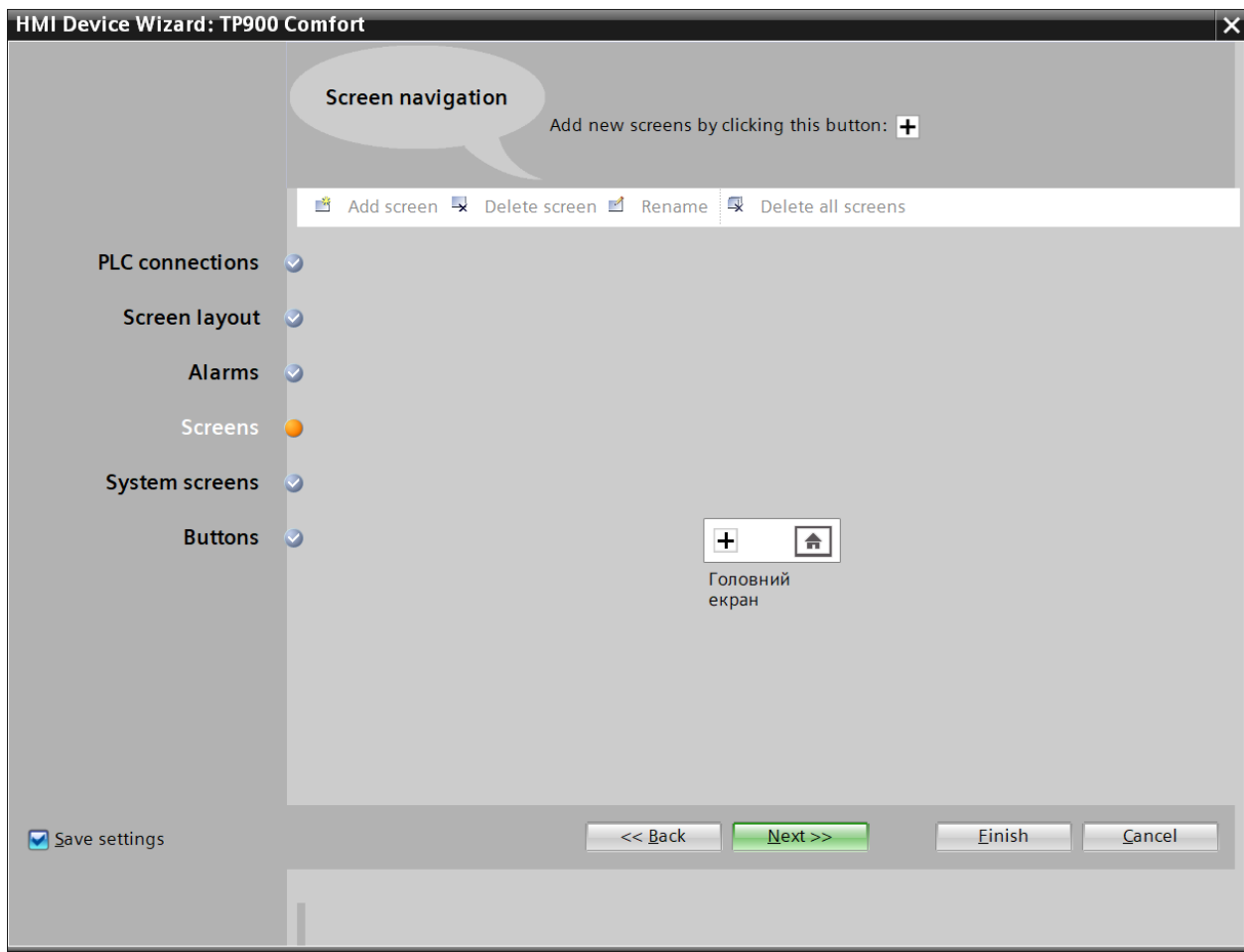


Рисунок 5.5 Налаштування екранів та навігації HMI.

Кожен екран інтерфейсу має свою назву та місце у ієрархічній структурі. Один з екранів обов'язково має бути визначений як стартовий – при запуску панелі оператора він відкривається перший. Для нашого проєкту достатньо буде одного головного екрану з назвою **Головний екран**, інші екрани можна буде додати пізніше.

Наступне вікно налаштувань дозволяє додати у проєкт системні екрани, на яких можна переглядати значення змінних ПЛК у вигляді таблиці **SIMATIC PLC Watch table**, стислу інформацію про проєкт, а саме його назву, час створення, автора та короткий опис **Project information**, екран обслуговування **System settings**, який дозволяє калібрувати екран та провести його механічну очистку без ризику втручання у перебіг технологічного процесу, екран адміністрування користувачів **User administration**, який призначений для встановлення прав та паролів доступу користувачам системи керування, екран з системною інформацією **System information**, на якому можна переглянути модель панелі оператора та ПЛК, а також екран з елементами керування панеллю (зміна режимів роботи **Operating modes**, зміна мови інтерфейсу **Language switching** та панель керування системними налаштуваннями **Control Panel**).

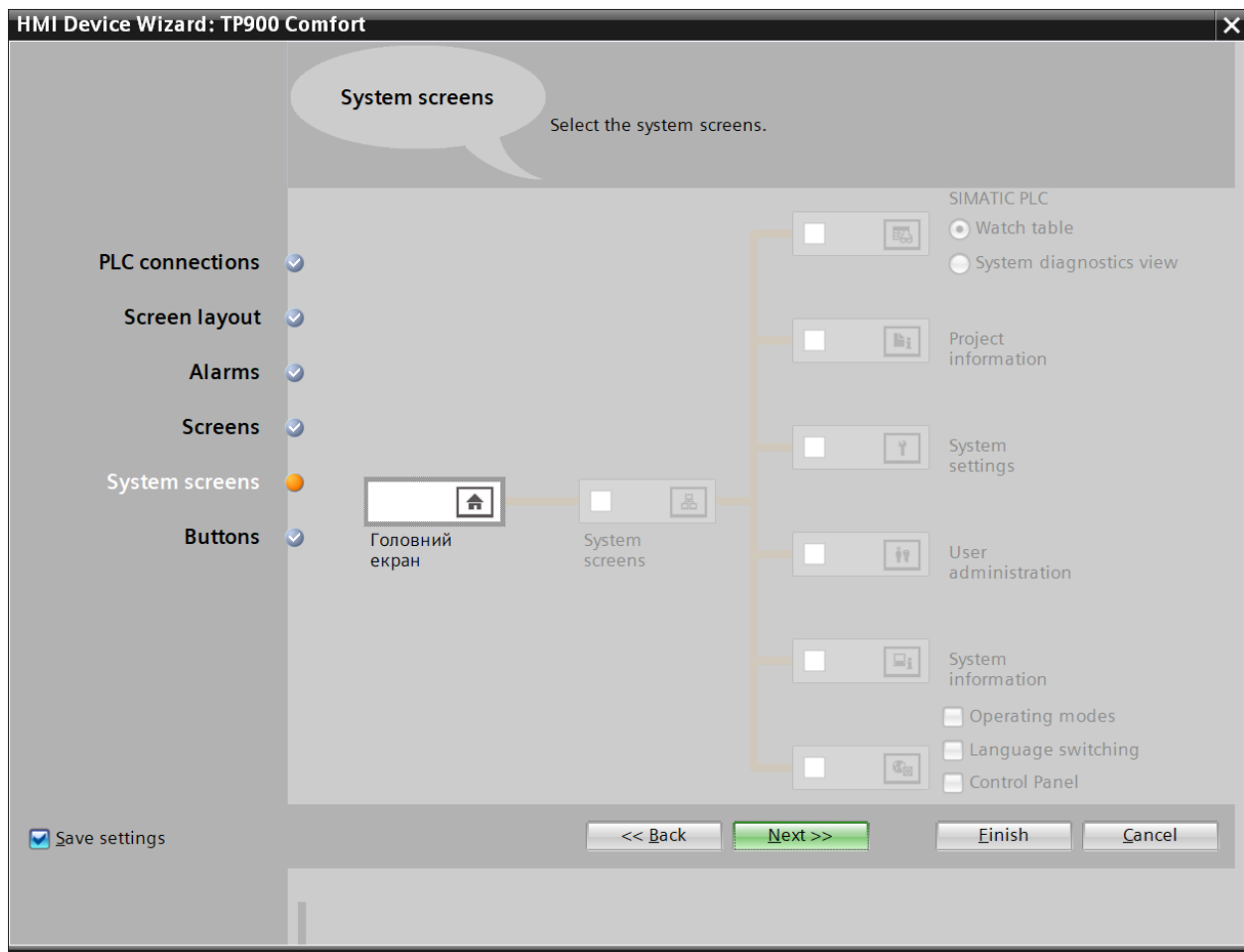


Рисунок 5.6 Налаштування системних екранів.

Ці вікна також можна не додавати, переходимо далі на останнє вікно майстра налаштувань Buttons, на якому можна визначити, які системні кнопки потрібно відображати та вказати місце, де вони будуть розташовані на кожному екрані людино-машинного інтерфейсу.

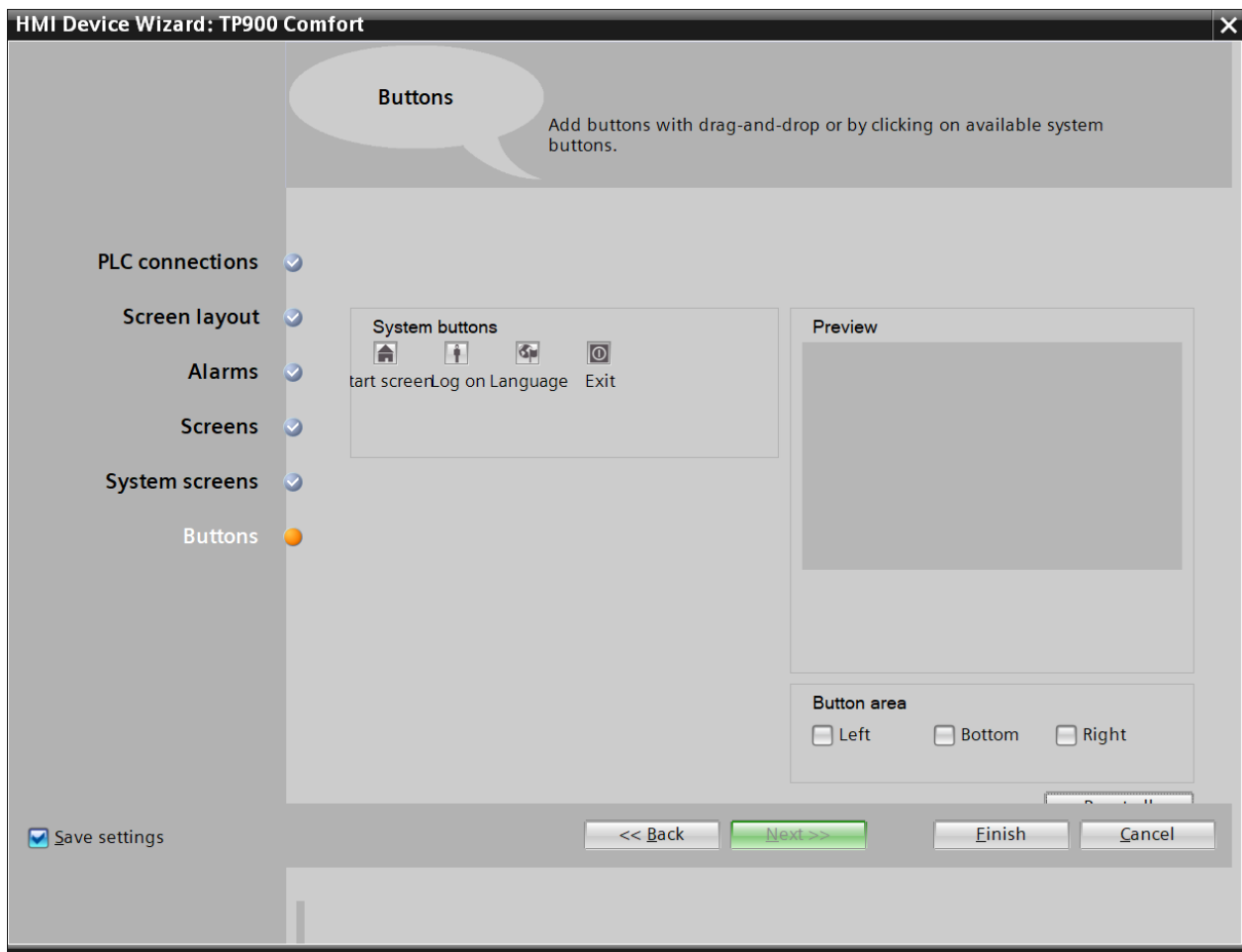


Рисунок 5.7 Налаштування постійних кнопок інтерфейсу.

За необхідності можна відображати на кожному екрані такі кнопки:

Start screen – кнопка для переходу на головний екран

Log on – кнопка для авторизації на панелі оператора

Language – кнопка для зміни мови відображення інтерфейсу

Exit – кнопка для виходу з облікового запису поточного користувача

Місце розташування кнопок **Button area** може бути вказана з лівого **Left**, нижнього **Bottom** та правого **Right** краю екрану.

Для даного проєкту постійні кнопки за допомогою майстра налаштувань додавати не потрібно, кнопкою **Finish** завершуємо роботу майстра налаштування панелі оператора. Після цього має відкритись режим редагування стартового екрану людино-машинного інтерфейсу доданої панелі оператора, а у дереві проєкта має з'явитись обрана панель оператора з заданими налаштуваннями у вигляді нового структурного елемента **HMI_1 [TP900 Comfort]**. Для зручності змінимо її назву на **HMI Control panel**.

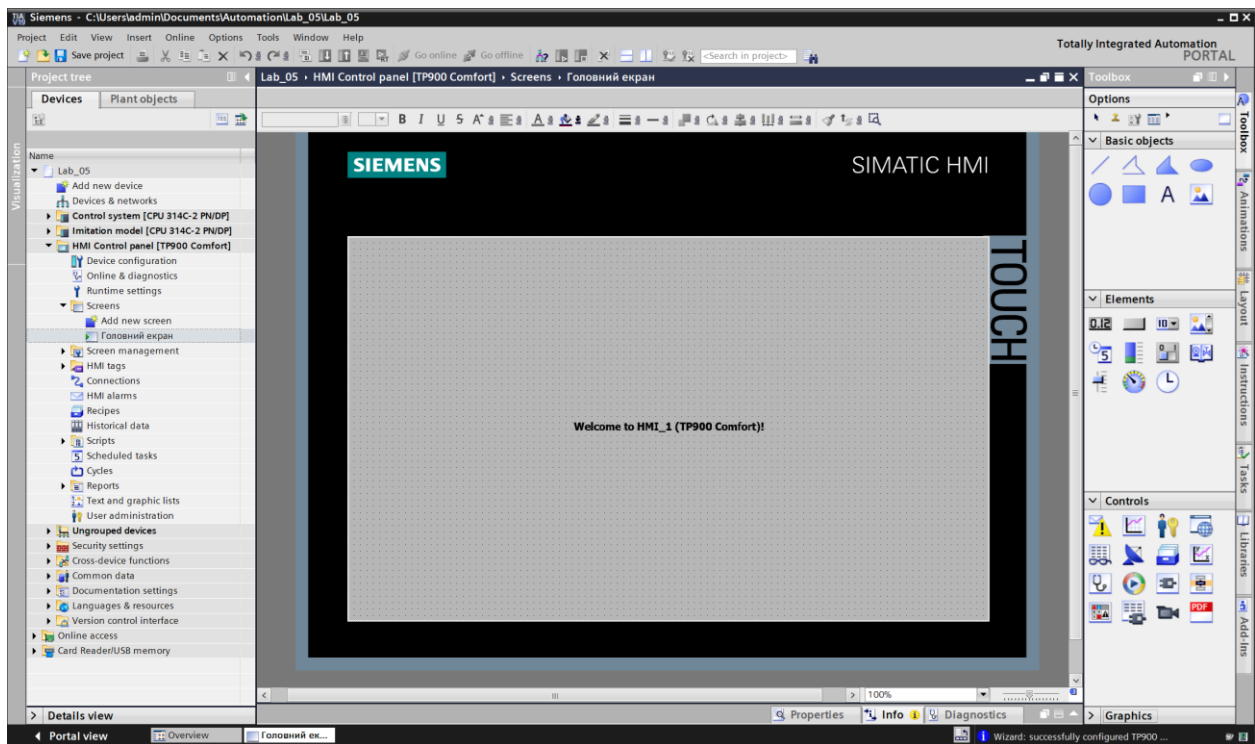


Рисунок 5.8 Редактор екранів HMI.

З правого боку робочої області редактора екранів HMI розташовані набори інструментів **Toolbox** для побудови інтерфейсів, до яких входять основні об'єкти інтерфейсу **Basic objects**, елементи вводу та виводу значень параметрів **Elements** та інструменти для роботи з журналами тривоги, обліковими записами користувачів, мультимедіа даними, інструменти діагностики системи тощо **Controls**.

Оскільки при налаштуванні панелі за допомогою майстра жодних елементів керування або додаткових екранів не було додано, стартовий екран містить лише один текстовий напис «**Welcome to HMI_1 (TP900 Comfort)!**» який можна виділити за допомогою миші та видалити.

5.2.2 Відображення на екрані основних технологічних параметрів системи регулювання тиску

Почнемо розробку інтерфейсу з додавання екранів для графіків та налаштування параметрів регулятора. Додати екрани можна за допомогою пункту **Add new screen** розділу **Screens** дерева проекту TIA Portal, який відповідає за налаштування панелі оператора **HMI Control Panel**. Додаємо таким чином два екрани та називаємо один з них **Графіки**, а другий – **Налаштування регулятора**.

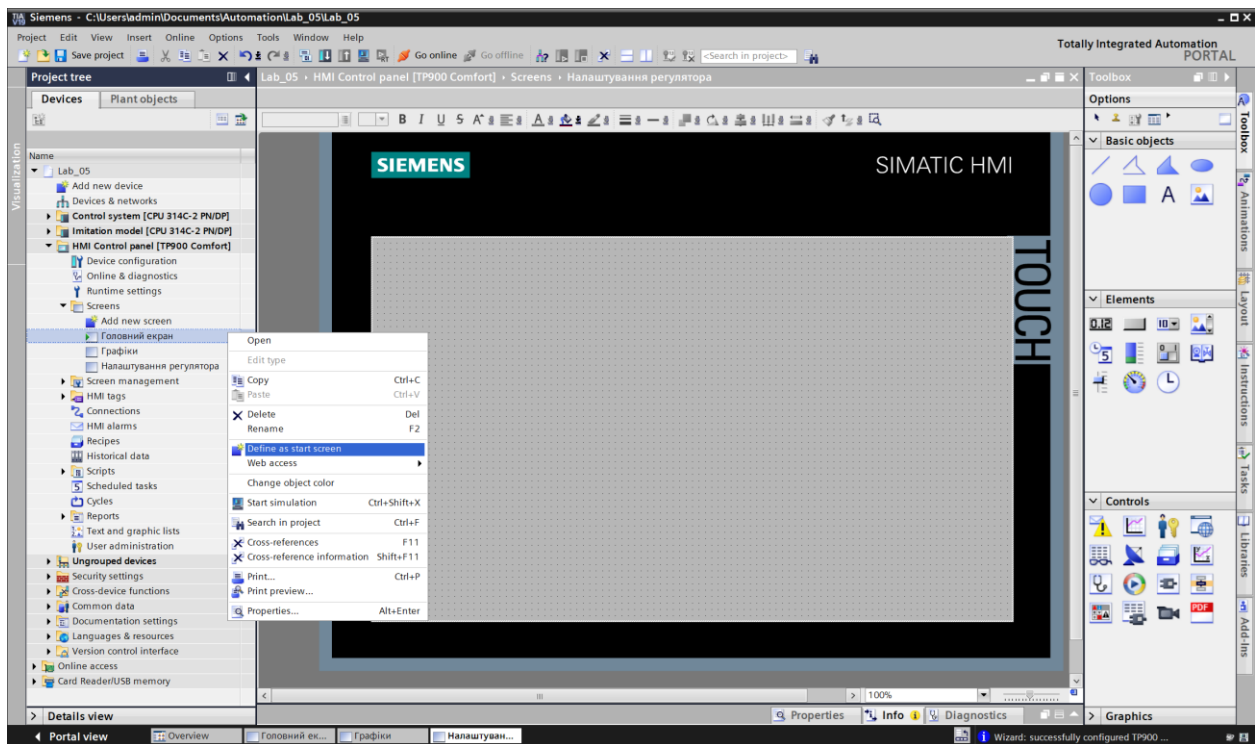


Рисунок 5.9 Призначення стартового екрану.

Слід зазначити, що значок екрану з назвою Головний екран має позначку з зображенням зеленого трикутника, який вказує на те, що він є стартовим, тобто відкривається першим при запуску панелі оператора. Змінити стартовий екран за необхідності можна за допомогою пункту **Define as start screen** контекстного меню.

Перейдемо до налаштування головного екрану. Для цього необхідно обрати у дереві проєкта елемент з назвою Головний екран та відкрити його редактор подвійним клацанням миші або за допомогою пункту **Open** контекстного меню. Для того, щоб мати можливість переходити між вікнами інтерфейсу оператора, нам знадобляться засоби навігації. Найпростішим способом реалізації навігації між вікнами є використання кнопок переходу, створити їх можна простим перетягуванням відповідних елементів з дерева проєкта на робочу область редактора головного вікна. При перетягуванні та відпусканні миші на робочій області екрани перетворюються на кнопки з текстом, що відповідає назві вікна.

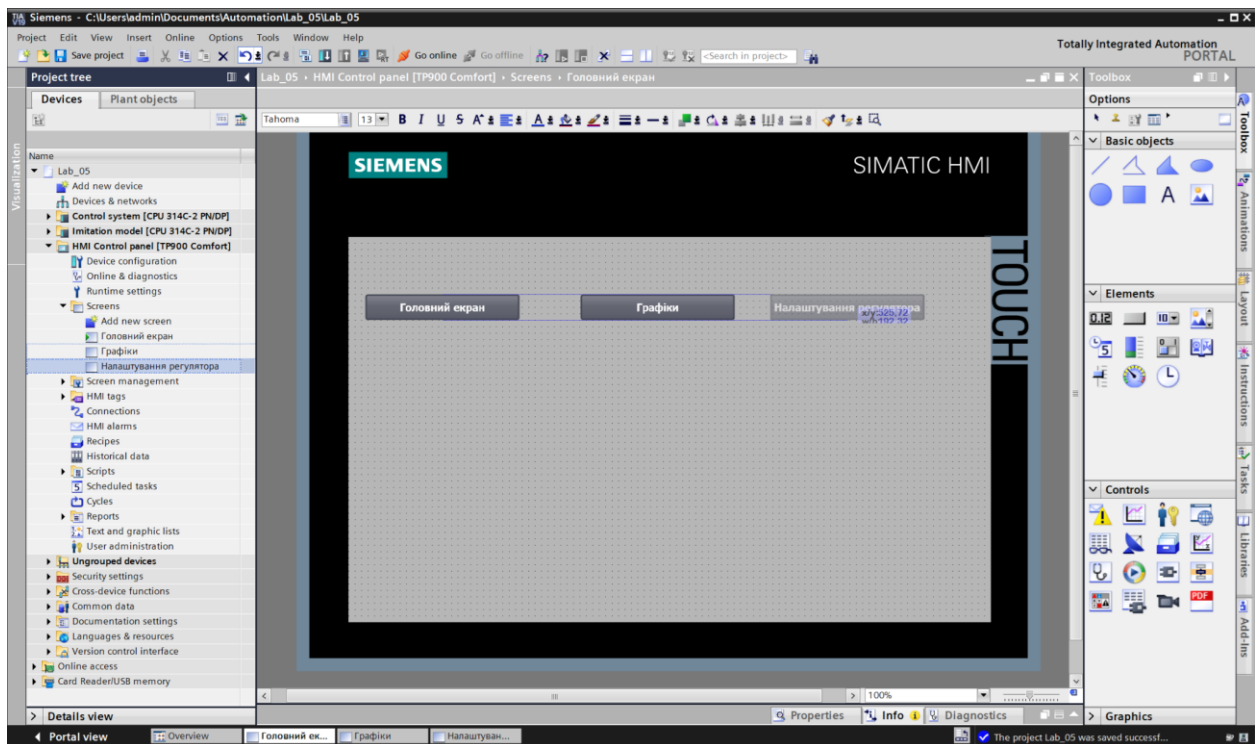


Рисунок 5.10 Встановлення кнопок переходу між екранами.

Для того, щоб кнопки переходу між екранами однаково були доступними з будь-якого екрана, можна створити виділену область у верхній частині екрана, яка дублюватиметься на кожному екрані та перемістити кнопки на цю область. Для того, щоб задати висоту цієї області, необхідно перейти у властивості виділеної області за допомогою пункту **Properties...** контекстного меню виділеної області, яка має назву **Permanent area** і знаходиться у підрозділі **Screen management** розділу налаштування панелі оператора **HMI Control Panel** дерева проєкта TIA Portal.

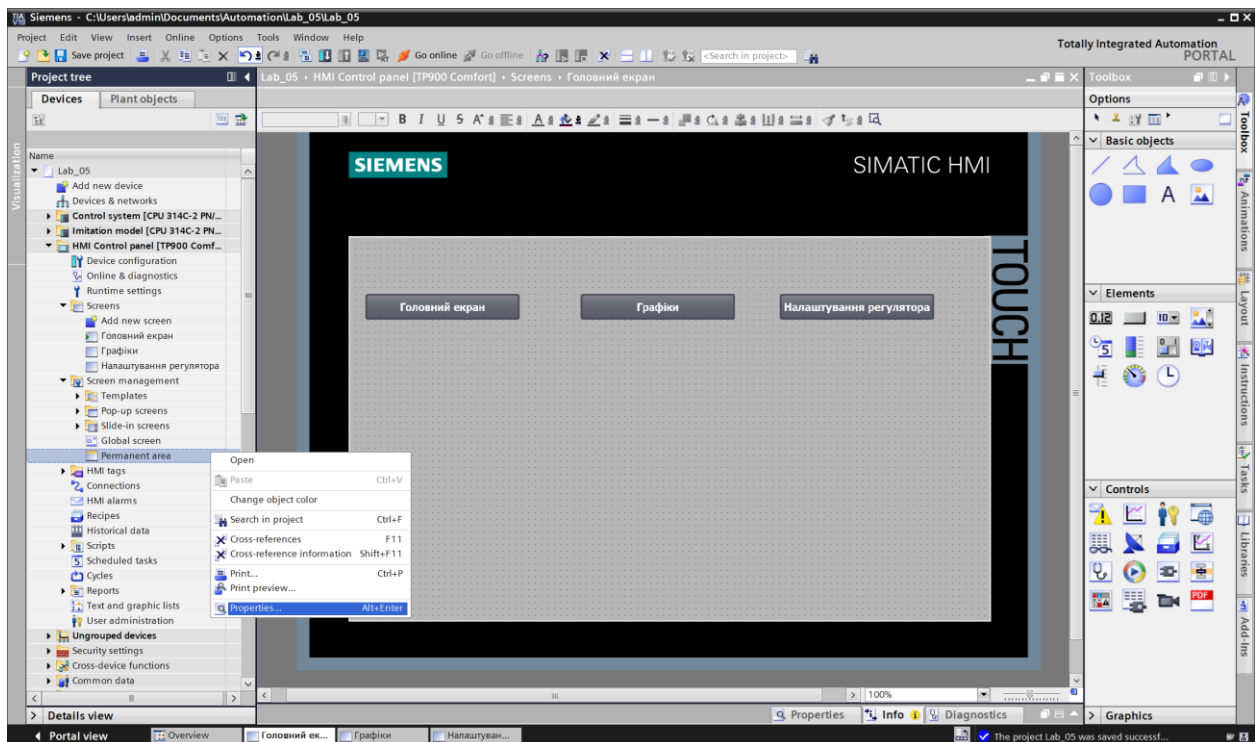


Рисунок 5.11 Перехід до визначення властивостей виділеної області.

Основним параметром виділеної області є її висота, яка задається у точках (пікселях), встановлюємо це значення рівним 60. За необхідності можемо змінити також колір виділеної області або лінії, які відділяють її від основної області екрана.

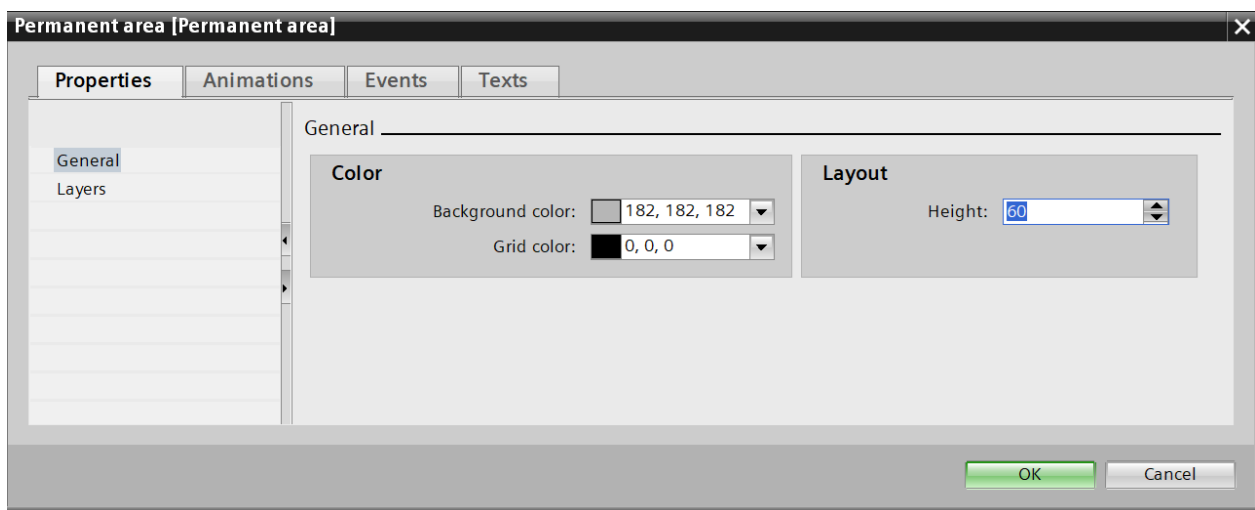


Рисунок 5.12 Вікно налаштування параметрів виділеної області.

Після створення виділеної області перетягуємо на неї кнопки переходу на екрани, підбираємо їх розмір та вирівнюємо відносно екрану. Значно легше ці дії виконувати за допомогою панелі інструментів, за допомогою якої можна як змінити шрифт надписів, так і відносне розташування елементів відносно інших елементів або відносно екрану. Для виділення кількох елементів потрібно

спочатку обрати головний елемент, відносно якого буде відбуватись вирівнювання, затиснути клавішу **Shift** та обрати інший або інші елементи. Головний елемент при цьому буде мати рамку виділення з можливістю зміни розмірів.

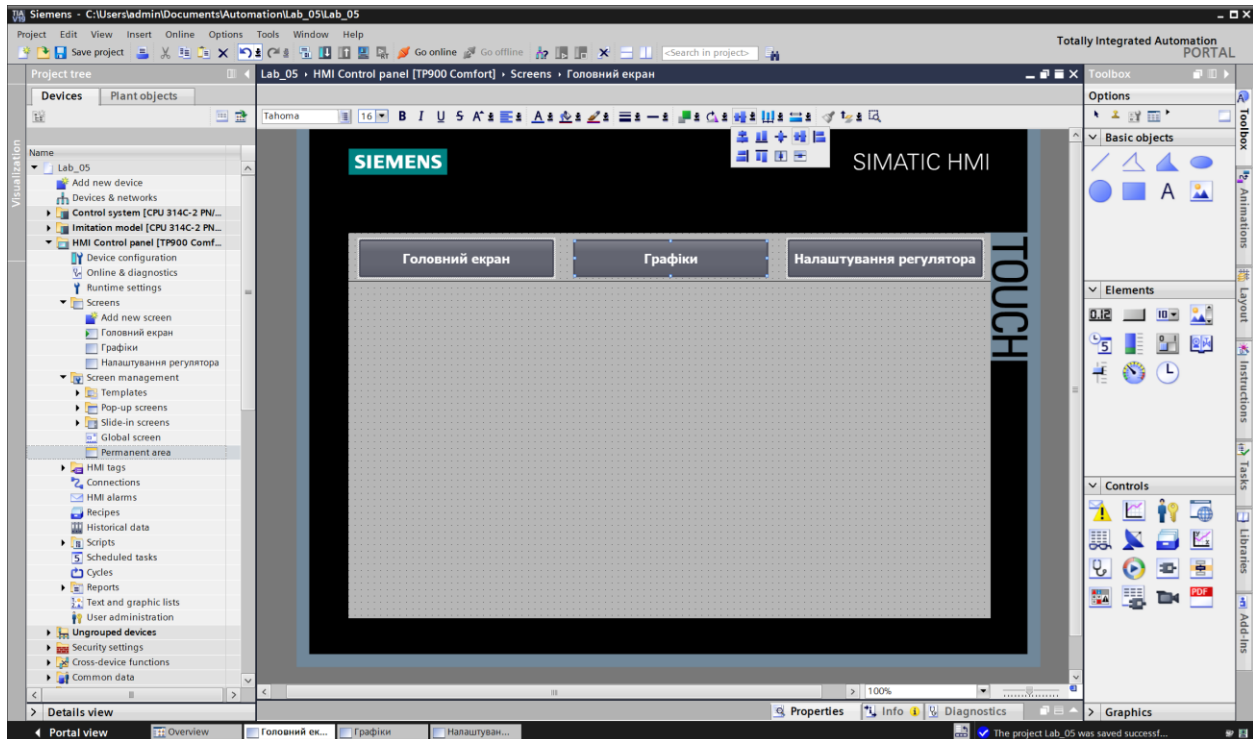


Рисунок 5.13 Встановлення кнопок навігації на виділеній області.

Таким чином, панель інструментів редактора екрана дозволяє змінювати параметри тексту, колір елементів та типи ліній, взаємне розташування елементів за різними шарами, коли один елемент може перекриватись іншими або бути на передньому плані та перекривати всі елементи, що знаходяться на нижчих шарах, здійснювати взаємне вирівнювання елементів або їх рівномірний розподіл, встановлювати однакові розміри елементів або копіювати їх властивості. Це дозволяє робити інтерфейс оператора зручним у користуванні, зробити процес розробки інтерфейсу швидшим, а його вигляд – більш професійним.

Для реалізації людино-машинного інтерфейсу необхідно відобразити такі дискретні (булеві) параметри, як робота системи регулювання в ручному режимі, робота системи регулювання в автоматичному режимі, перегрів двигуна насоса та сигнал дозволу на роботу частотного перетворювача. Зазвичай для таких параметрів використовують світлові індикатори з поясненням їх призначення. Для інформаційних сигналів рекомендовано використовувати індикатори нейтральних кольорів, для аварійних – яскраві, які привертають до себе увагу.

Як світлові індикатори режимів роботи системи та аварійного повідомлення про перегрів двигуна насоса будемо використовувати графічні елементи прямокутник Rectangle з текстовим полем Text field для пояснення призначення елемента. Розмір шрифта потрібно підбирати таким чином, щоб оператор міг його легко прочитати з комфортної відстані від панелі оператора. Розташуємо ці елементи на екрані Головний екран як показано нижче на рисунку.

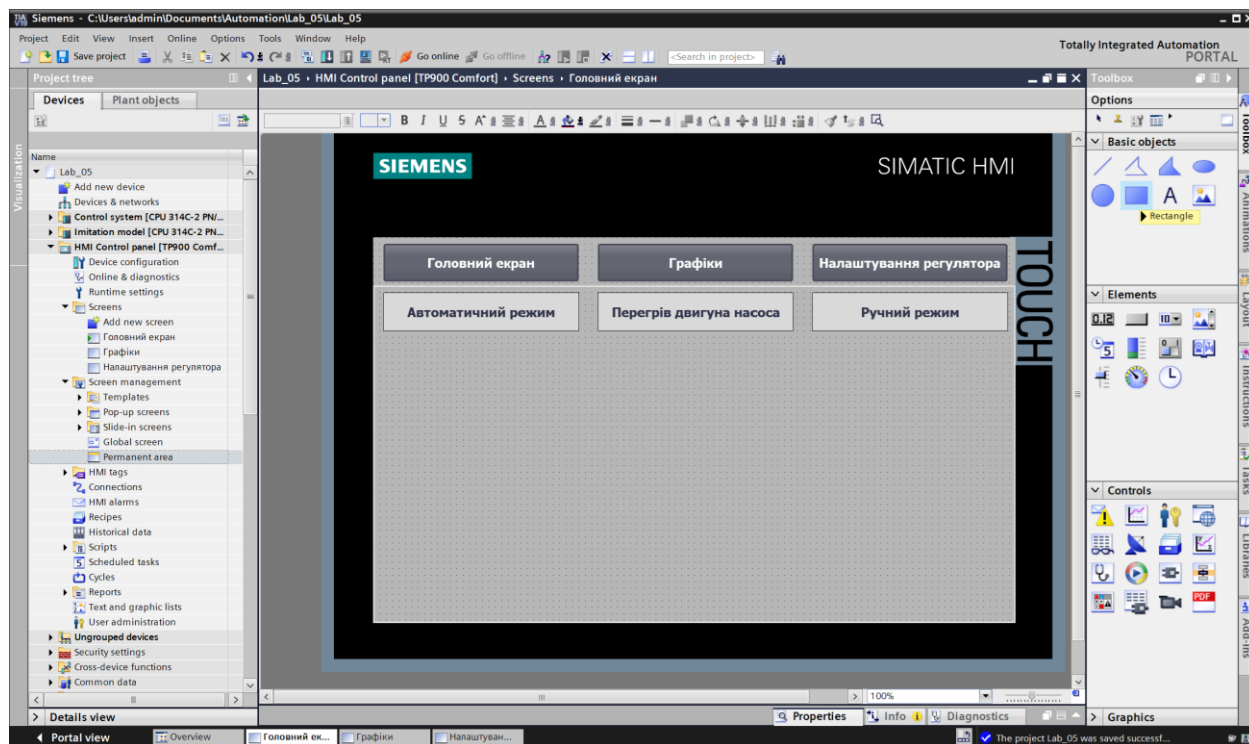


Рисунок 5.14 Встановлення елементів індикації станів системи.

При розташуванні елементів варто користуватись такими засобами панелі інструментів, як вирівнювання розмірів, рівномірний розподіл елементів та вирівнювання по горизонтальній лінії.

Для індикації сигналу дозволу на роботу частотного перетворювача, що запускає двигун насоса, будемо використовувати зображення насоса, яке можна знайти у бібліотеці символів **Symbol library**. Для того, щоб розмістити це зображення на екрані, необхідно спочатку перетягнути елемент **Symbol library**, що має вигляд книжки з символами пристроїв, з набору інструментів **Elements** на робочу область екрана.

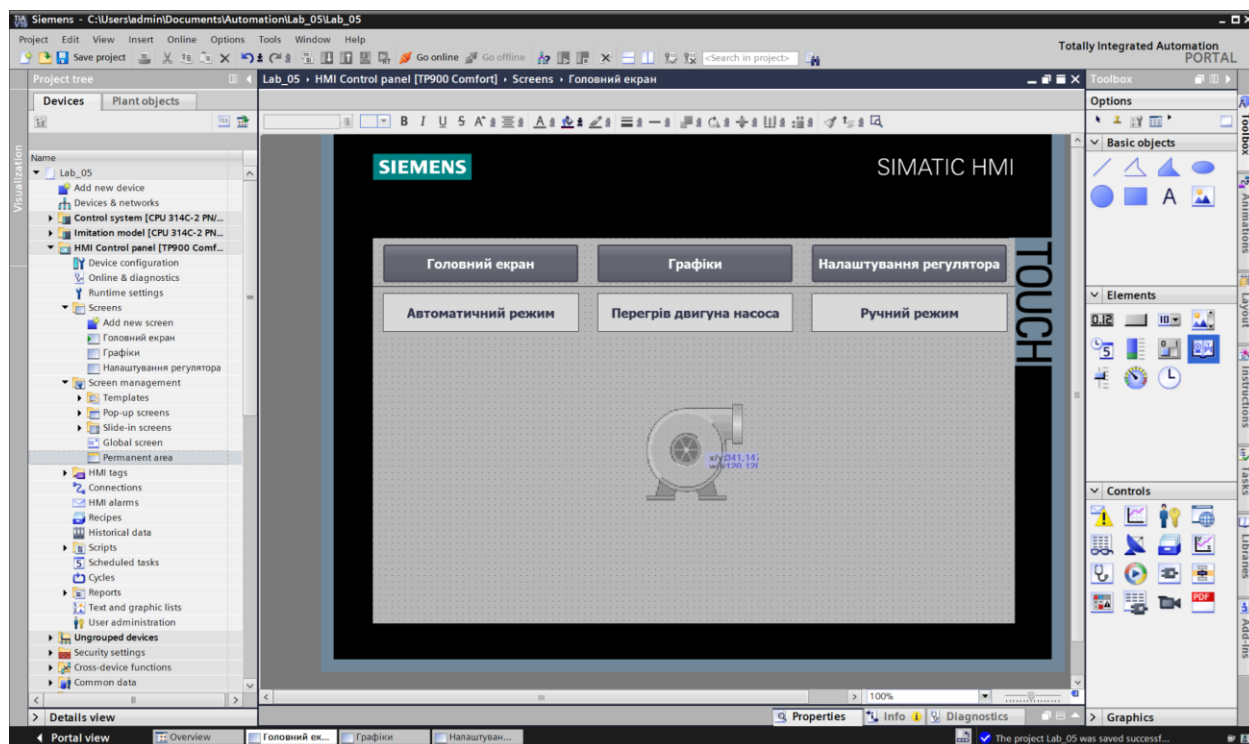


Рисунок 5.15 Розміщення на екрані елемента *Symbol library*.

При перетягуванні на робочу область екрана елемент **Symbol library** може мати вигляд, який не відповідає завданню. Для вибору необхідного зображення потрібно перейти до властивостей елемента **Symbol library** через пункт **Properties** контекстного меню елемента, у вікні налаштувань перейти до загальних налаштувань, які містяться у розділі **General** та обрати необхідне зображення. У бібліотеці представлена велика кількість графічних елементів, які груповані за призначенням. За допомогою цих елементів можна відносно швидко побудувати мнемосхеми та схеми різного призначення, зобразити будівлі, установки, та окремі елементи системи як у вигляді зображень, так і у вигляді стандартних умовних позначень. За необхідності можна змінити колір, розмір зображення, повернути зображення або дзеркально його відобразити.

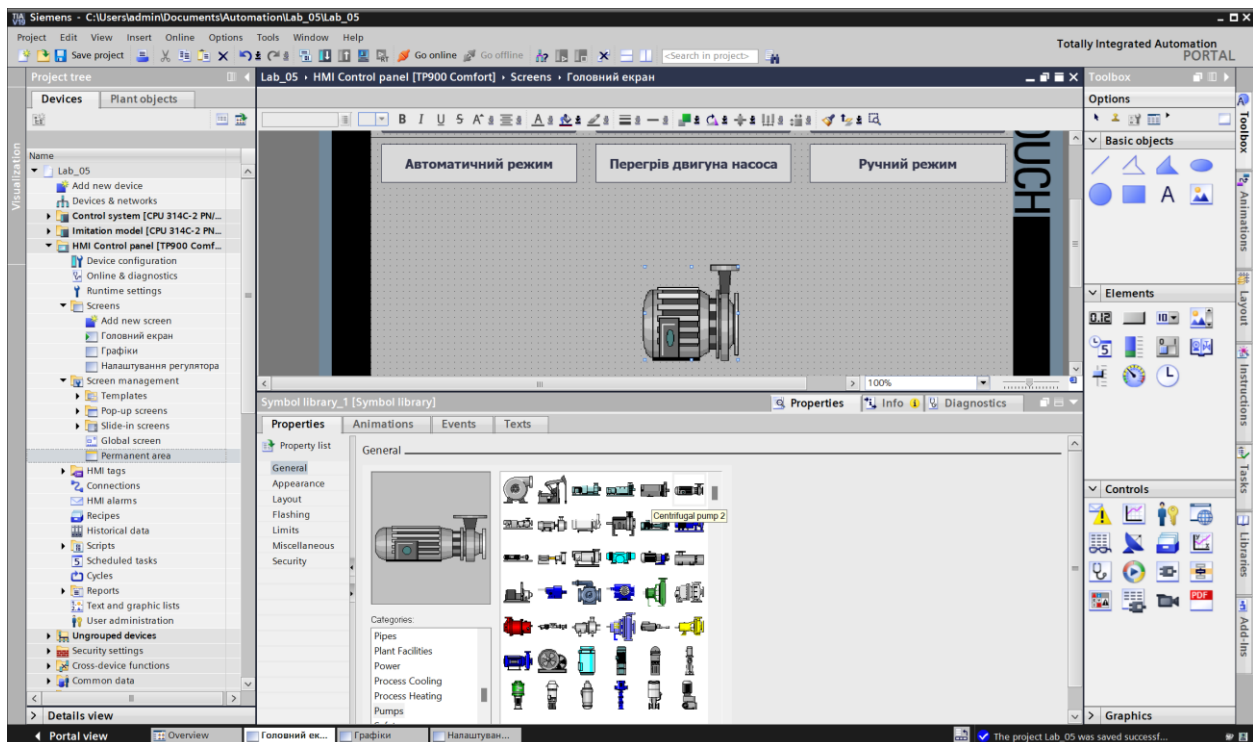


Рисунок 5.16 Зміна зображення елемента Symbol library.

Для відображення або вводу числових значень параметрів будемо використовувати поле вводу/виводу **I/O field** з набору інструментів **Elements**. На головному екрані потрібно буде відобразити інформацію про такі технологічні параметри: значення тиску в системі, заданий тиск у системі, поточна продуктивність насоса та продуктивність насоса у ручному режимі. Для пояснення призначення кожного параметра можна використовувати інструмент **Text field** з набору інструментів **Basic objects**.

Розміщуємо вказані елементи відповідно до наведеного нижче рисунку, встановлюємо їх розміри та розмір шрифту тексту таким чином, щоб значення та їх призначення можна було легко прочитати з комфортної відстані від панелі оператора.

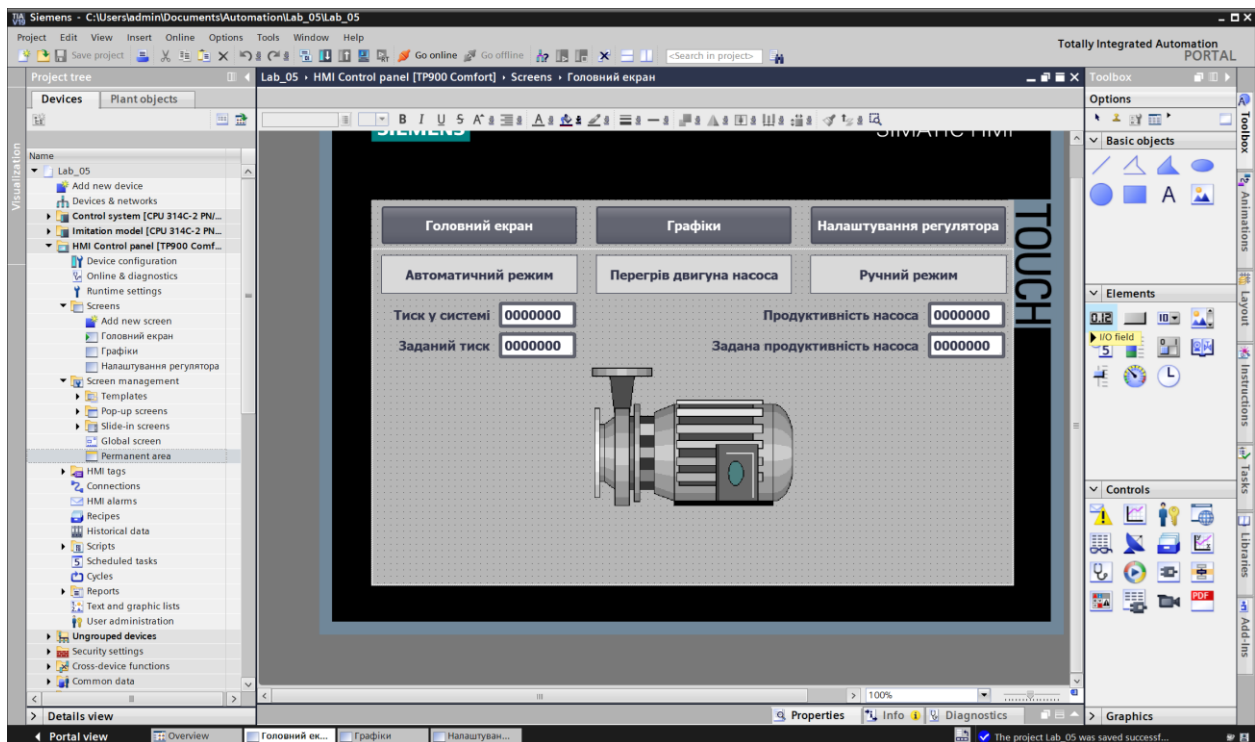


Рисунок 5.17 Розміщення елементів відображення технологічних параметрів.

Після розміщення елементів відображення технологічних параметрів на екрані необхідно налаштувати їх зв'язок з відповідними параметрами контролера **Control system**.

Для зміни кольору індикаторів автоматичного та ручного режимів будемо використовувати значення відповідних сигналів **SA1_1** та **SA1_2** від перемикача режимів роботи. Для налаштування відображення переходу системи у автоматичний режим роботи необхідно обрати елемент прямокутник **Rectangle_1**, що відповідає надпису **Автоматичний режим** та перейти на вкладку **Animations** вікна властивостей цього елемента. На цій вкладці будуть доступні варіанти анімації елемента зміною його положення **Movements** та зміною його відображення **Display**, що, у свою чергу, містить варіанти анімації зміною зовнішнього вигляду **Appearance** та видимості елемента **Visibility**. Для елемента **Rectangle_1** додаємо анімацію зміною зовнішнього вигляду **Appearance** натисканням на значок додавання варіанту анімації.

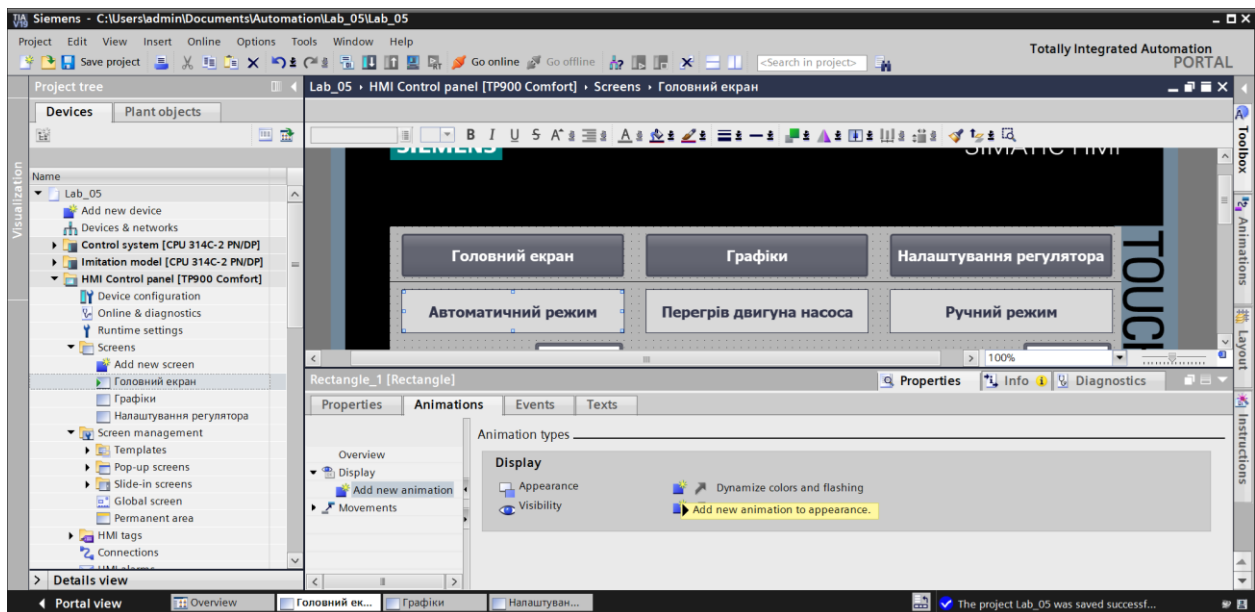


Рисунок 5.18 Додавання анімації індикатора автоматичного режиму роботи системи.

У вікні налаштувань анімації зовнішнього вигляду встановлюємо прив'язку анімації до сигналу **SA1_1**, який дає контролеру команду переводу системи у автоматичний режим роботи, зі списку тегів **Default tag table** контролера **Control system**.

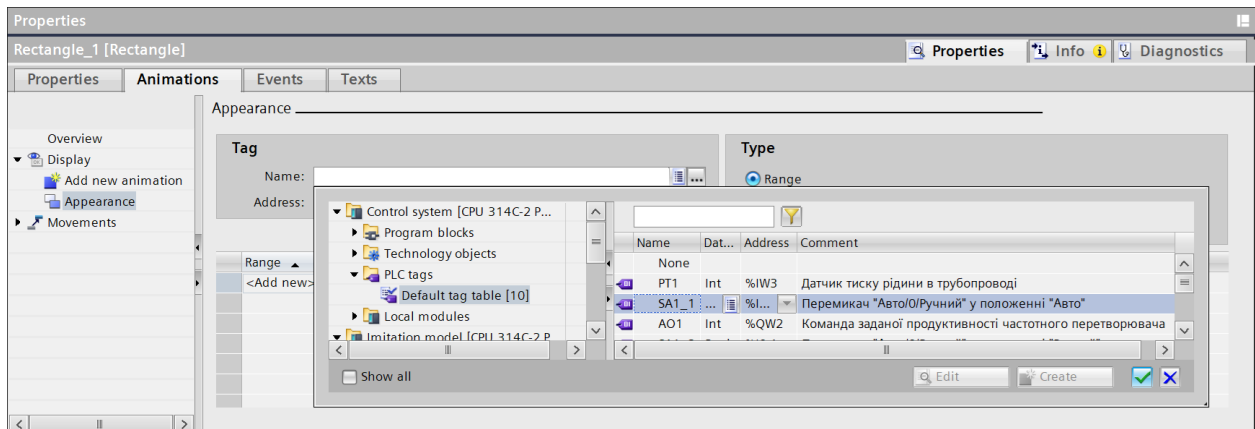


Рисунок 5.19 Прив'язка анімації елемента *Rectangle_1* до тегу контролера *SA1_1*.

Після встановлення прив'язки до тегу необхідно натисканням на текст **<Add new>** у таблиці відповідності зовнішнього вигляду елемента станам тегу додати можливі значення параметра **SA1_1** (для дискретних сигналів це значення **0** та **1**) і визначити у відповідних полях таблиці кольори заливки **Background color** елемента **Rectangle_1**, які будуть цим станам відповідати. Для неактивного стану автоматичного режиму, який відповідає значенню тегу **SA1_1**, що дорівнює 0, встановлюємо темно-сірий колір заливки, а для активного

стану, якому відповідає значення 1 – білий колір заливки. Інші параметри залишаємо незмінними.

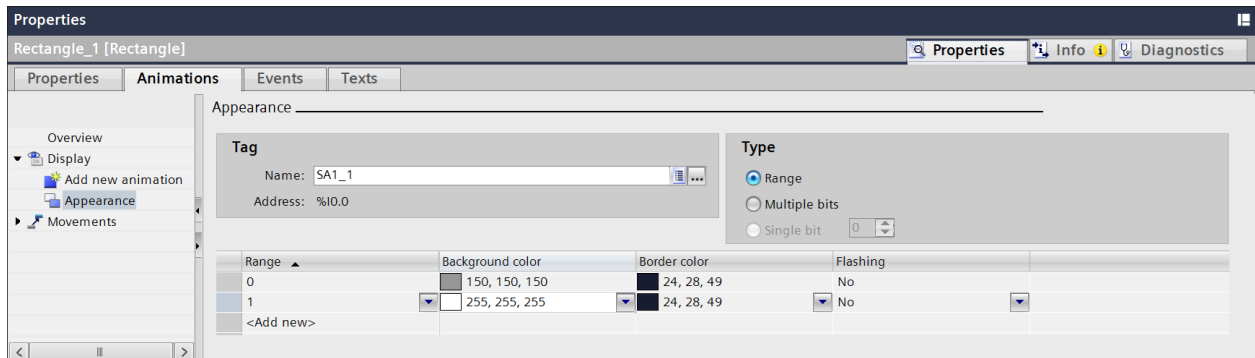


Рисунок 5.20 Параметри анімації індикатора автоматичного режиму роботи системи.

Для індикатора ручного режиму **Rectangle_3** аналогічно встановлюємо прив'язку до тегу **SA1_2** контролера **Control system**.

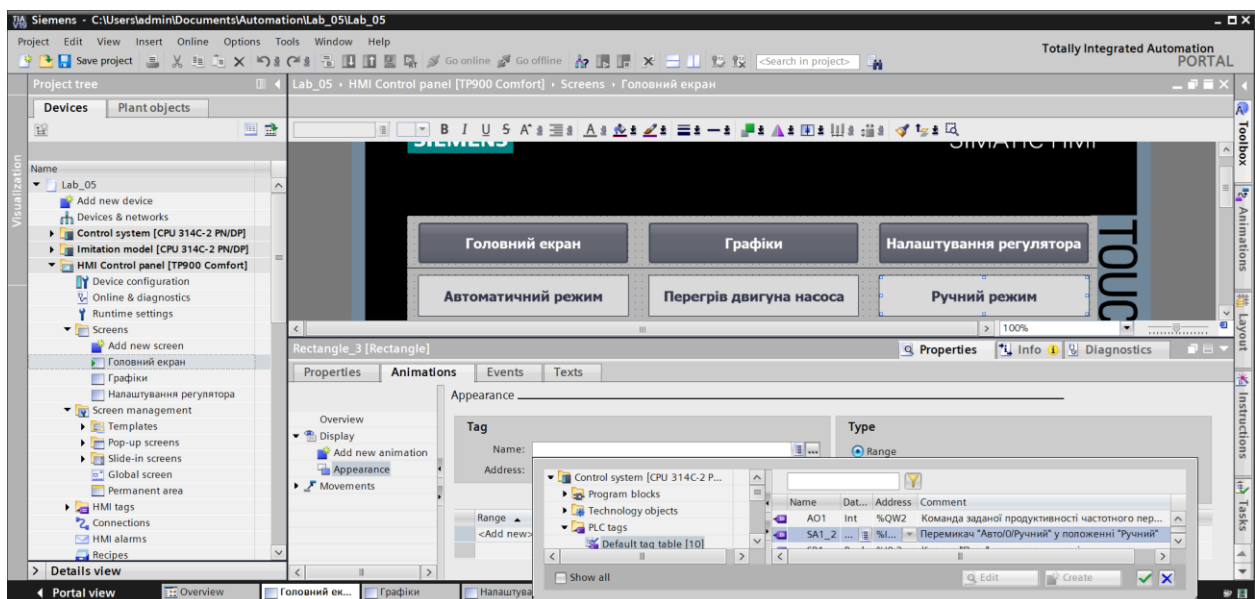


Рисунок 5.21 Прив'язка анімації елемента **Rectangle_1** до тегу контролера **SA1_2**.

Кольори анімації індикатора ручного режиму **Rectangle_3** встановлюємо аналогічно індикатору автоматичного режиму роботи.

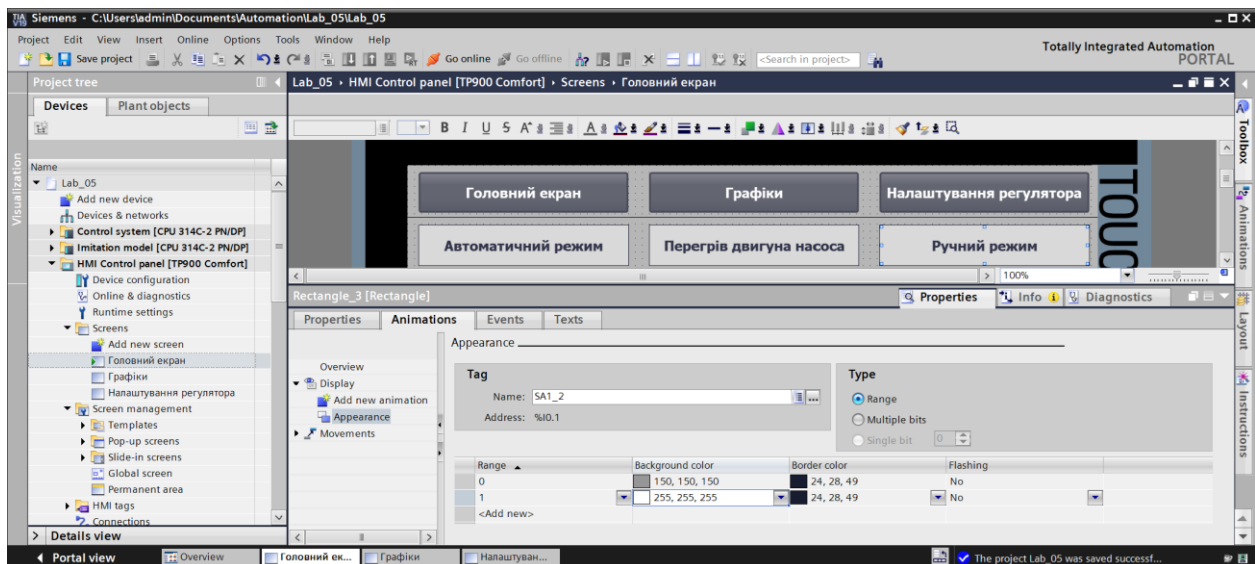


Рисунок 5.22 Параметри анімації індикатора ручного режиму роботи системи.

Оскільки перегрів двигуна насоса є аварійною подією, для неї потрібно створити більш яскраве відображення. Тому змінимо колір індикатора перегріву за замовченням на червоний, а колір тексту – на жовтий.

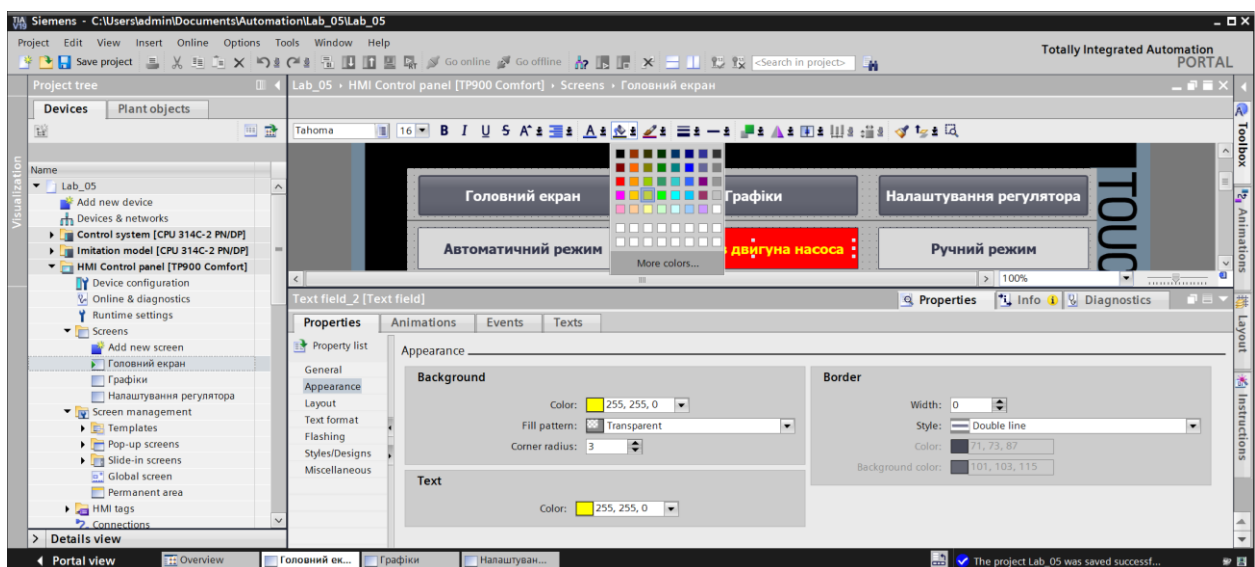


Рисунок 5.23 Зміна параметрів відображення індикатора перегріву.

Для цього необхідно виділити відповідний елемент та за допомогою панелі інструментів або розділу **Appearance** вікон властивостей індикатора **Rectangle_2** та текстового надпису **Text field_2** встановити відповідні кольори відображення.

Після налаштування зовнішнього вигляду елементів індикатора перегріву необхідно для кожного з них додати варіант анімації видимості **Visibility** та прив'язати її до значення тегу **TK1**, який відповідає нормально замкненому

контакту термореле двигуна насоса. Тег **TK1** можна знайти у таблиці **Default tag table** контролера **Control system**.

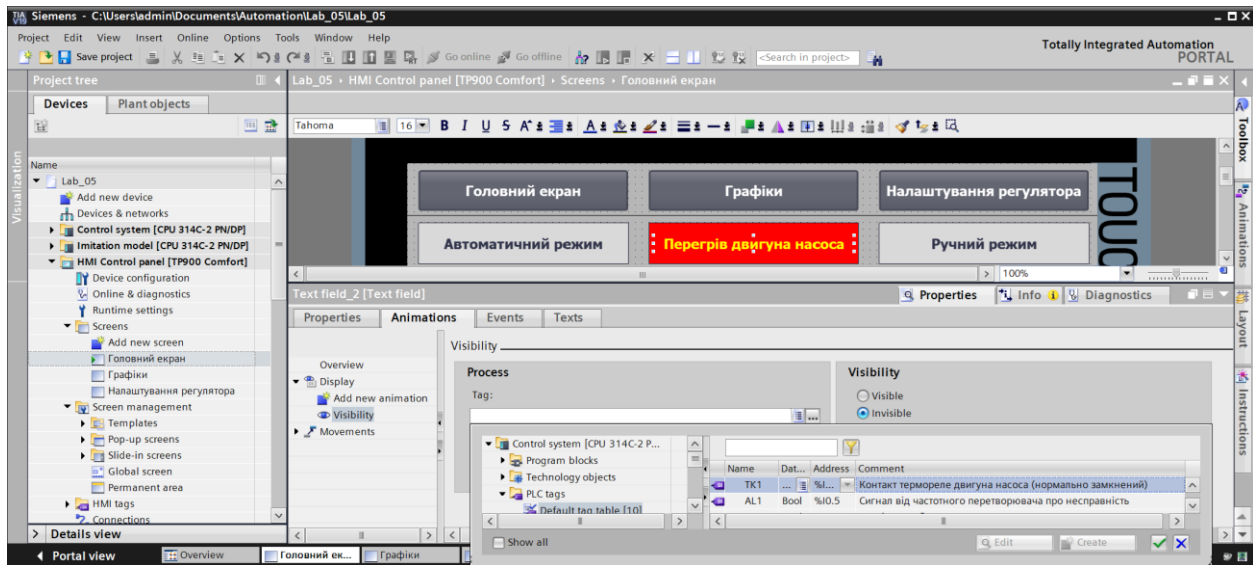


Рисунок 5.24 Прив'язка анімації тексту аварійного повідомлення до тегу **TK1**.

Слід зауважити, що для варіанту анімації **Visibility** необхідно встановити діапазон значень **Range** для тегу **TK1**, при якому активується анімація, а також вказати за допомогою перемикача **Visibility** режим анімації: **Visible** або **Invisible**. Цей параметр визначає, чи буде текст аварійного повідомлення **Text field_2** видимим (**Visible**) або невидимим (**Invisible**) при вказаному діапазоні значень тегу **TK1**. Оскільки сигнал **TK1** з'являється при нормальній роботі двигуна насоса та зникає при його перегріві, повідомлення про аварію має ставати невидимим при значенні **TK1**, що дорівнює одиниці, тому діапазон активації анімації буде складатись лише з одного значення **1**, яке потрібно записати у обох полях: **From** (нижнє значення діапазону) та **To** (верхнє значення діапазону), режим анімації встановлюємо **Invisible**.

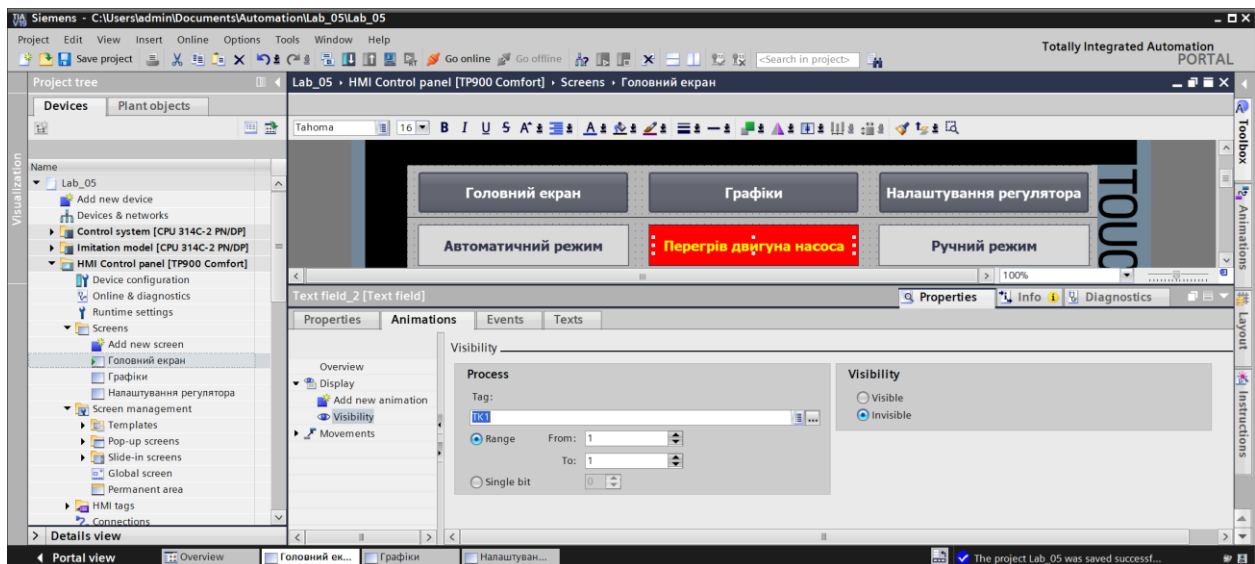


Рисунок 5.25 Налаштування анімації тексту аварійного повідомлення.

Налаштування анімації для прямокутного індикатора перегріву двигуна насосу **Rectangle_2** встановлюємо такі самі, як і для тексту повідомлення.

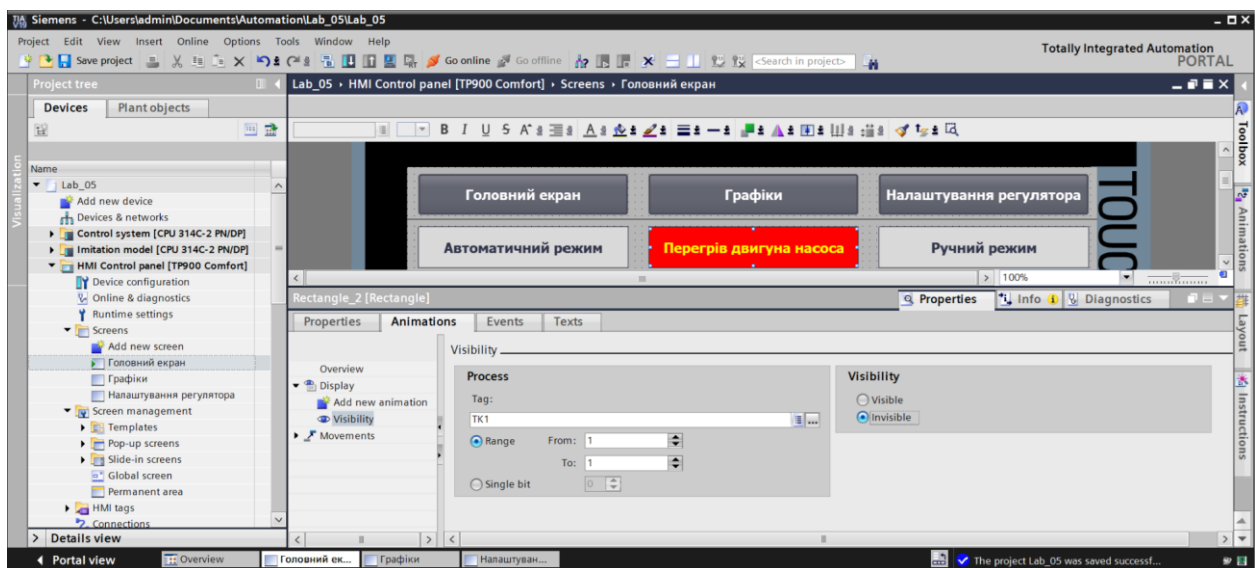


Рисунок 5.26 Параметри анімації індикатора перегріву двигуна насосу **Rectangle_2**.

Індикацію дозволу на роботу частотного перетворювача, який подається через дискретний вихід контролера **DO1** на частотний перетворювач, реалізуємо зміною кольору зображення насоса. Для цього необхідно спочатку змінити стиль заповнення елемента бібліотеки **Symbol library**, який використовується для відображення насоса на екрані, зі значення **Original** на значення **Shaded**, оскільки інші способи заповнення (**Hollow**, **Solid**, або **Original**) не передбачають можливості зміни кольору зображення зі збереженням всіх деталей зображення.

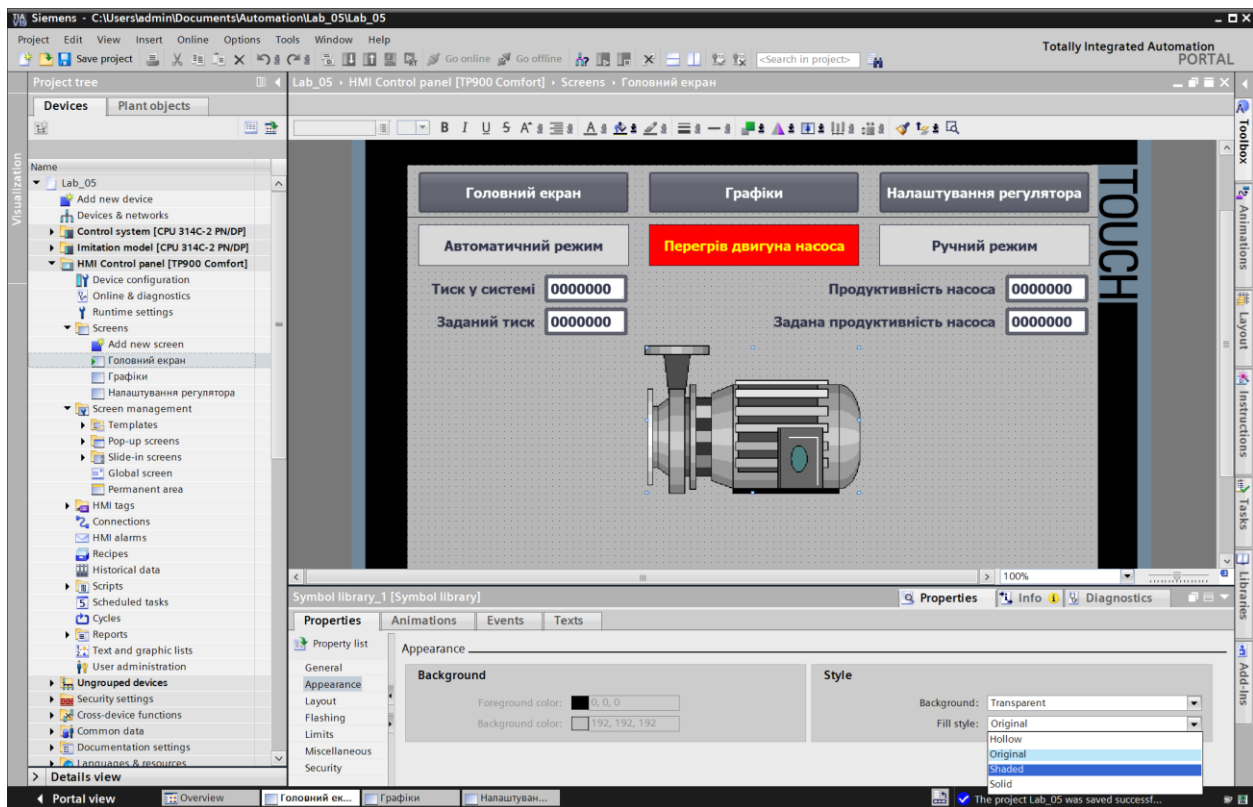


Рисунок 5.27 Зміна стилю заповнення елемента бібліотеки Symbol library на Shaded.

Після встановлення вказаного стилю заповнення відтінок зображення визначатиметься параметром **Foreground color**, тому для анімації зовнішнього вигляду насоса потрібно прив'язати анімацію цього параметра до значення тегу **DOI1**. Варіант анімації для зміни кольору – **Appearance**.

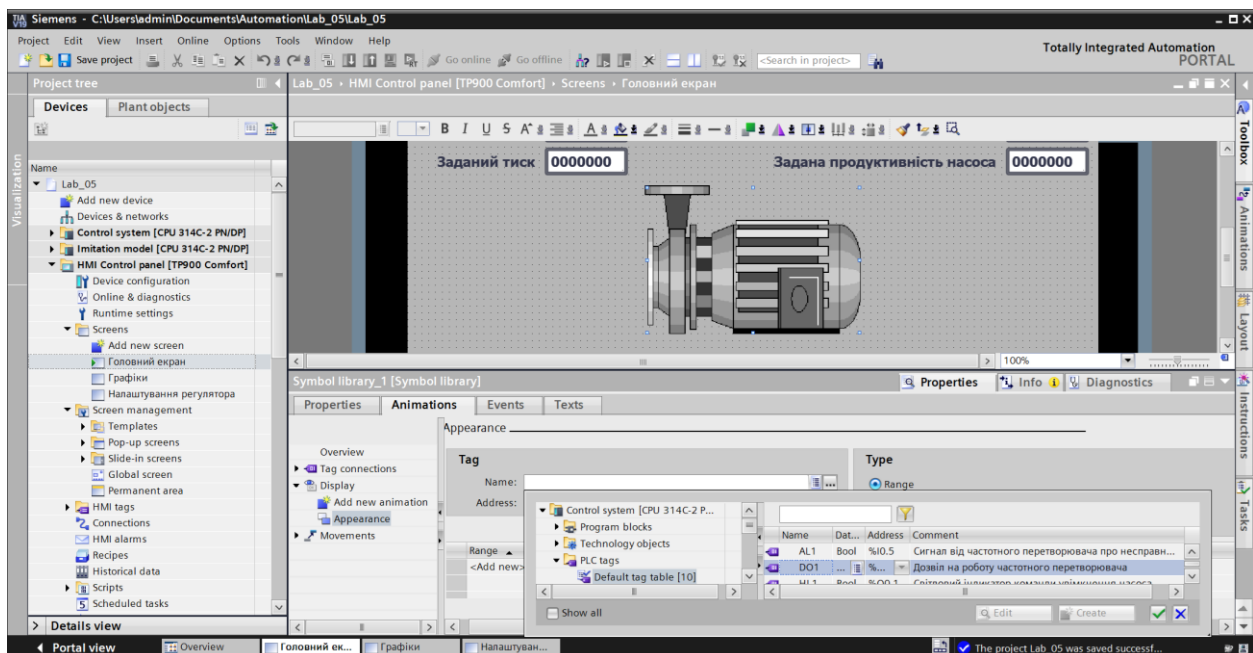


Рисунок 5.28 Прив'язка тегу DOI1 до анімації зображення насоса.

Встановлюємо такі параметри анімації елемента бібліотеки **Symbol library** з зображенням насосу:

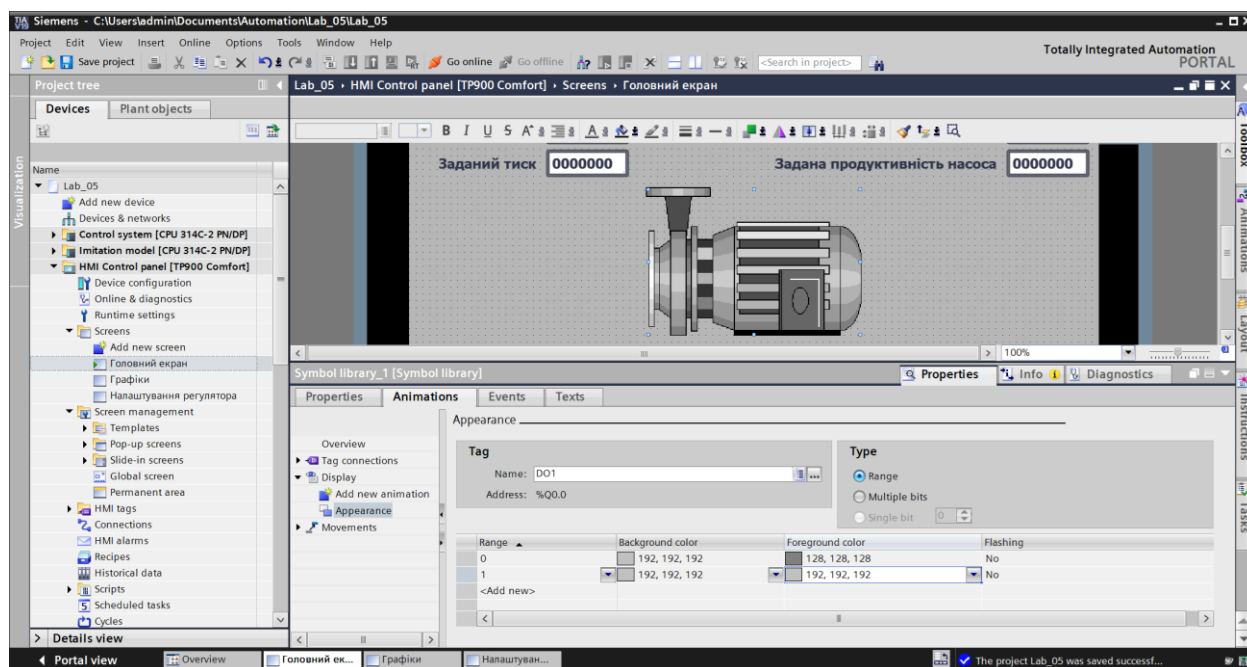


Рисунок 5.29 Параметри анімації зображення насосу.

При значенні сигналу **DO1**, що дорівнює **0**, колір контуру зображення **Foreground color** буде темно-сірим (128, 128, 128), а при значенні **1** – світло-сірим (192, 192, 192).

Переходимо до налаштування відображення числових значень технологічних параметрів. Відкриваємо налаштування вікна виводу значення тиску в системі **I/O field_1**. Необхідно встановити прив'язку цього елемента до імітованого значення регульованого параметра - змінної **rSimulatedProcessValue**, яку можна знайти у блоці даних **Data_block** контролера **Control system**.

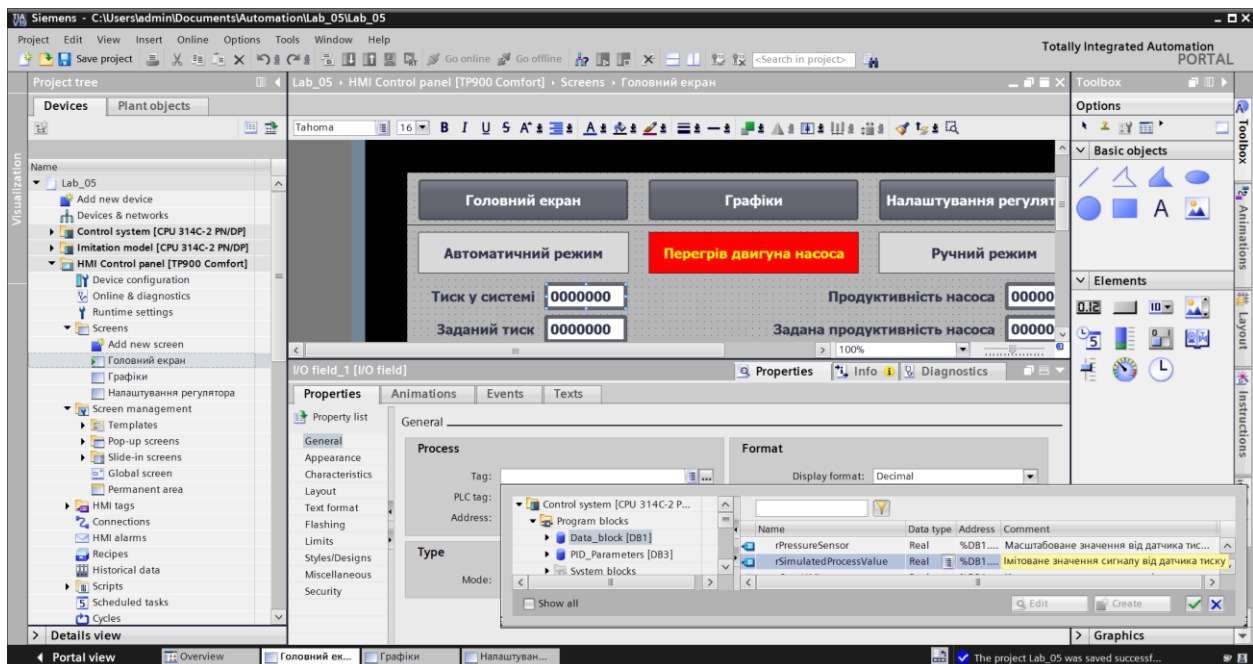


Рисунок 5.30 Прив'язка тегу *rSimulatedProcessValue* до поля виводу.

Крім цього, необхідно також зазначити тип поля вводу/виводу та формат відображення значення тегу. Тип поля вводу/виводу може бути **Input** – для тих параметрів, значення які потрібно тільки вводити за допомогою панелі оператора, **Output** – для тих параметрів, значення яких повинно виводитись на екран, але не може бути зміненим оператором та **Input/output** – для значень, які повинні виводитись на екран та які оператор за необхідності може змінювати. Оскільки значення тиску, яке в працюючій системі контролер отримує від датчика тиску, а в системі з імітаційним моделюванням це значення обчислюється контролером **Imitation model**, оператор не має можливості змінити за допомогою HMI панелі, тип елемента **I/O field_1** потрібно встановити рівним **Output**, тобто значення тиску в системі буде тільки виводитись на екран людино-машинного інтерфейсу. Формат відображення **Display format** числового значення потрібно встановити як десятковий **Decimal**. - він єдиний з доступних, який може відобразити дійсне числове значення у десятковій системі числення. Шаблон відображення **Format pattern** визначає кількість знаків цілої частини та кількість знаків після розділового знаку, необхідність зазначення знаків + та – для позначення додатних та від'ємних значень і, за необхідності – експоненціальний формат запису числа. Для позначення кількості значущих цифр при записі шаблону використовуються символи **9**, необхідність виділення місця для знаку числа позначається символом **s** перед першою цифрою, а для позначення експоненціального формату запису числа – символ **e** латиницею після останньої цифри шаблону.

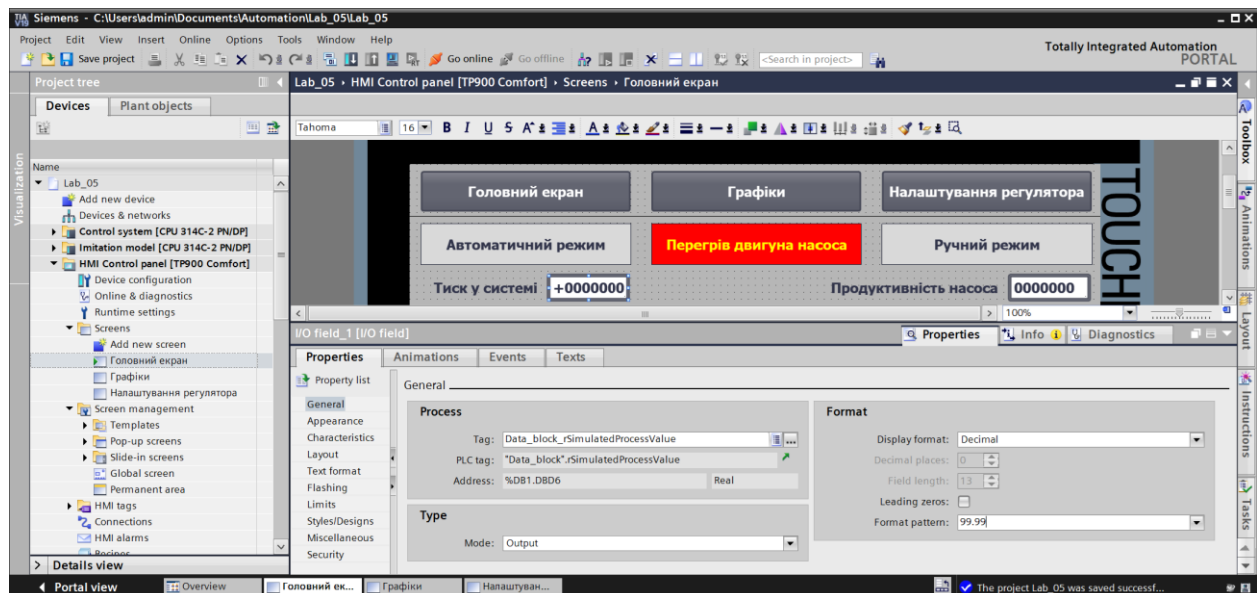


Рисунок 5.31 Налаштування загальних параметрів поля виводу значення тиску в системі.

Записуємо шаблон форматування значення тиску в системі у вигляді **99.99**, оскільки значення тиску за завданням може знаходитись у діапазоні від **0.0** до **10.0**, тож воно не може приймати від'ємні значення і символ **s**, який використовується в шаблоні для позначення знаку числа, використовувати не доцільно, а ціла частина значення тиску відповідно до завдання не може містити більше двох символів, тож використання більше двох символів після розділового знаку більше ускладнює сприйняття числового значення параметра, ніж надає корисної інформації. Експоненціальна форма запису значення цього параметру в даному випадку буде недоцільною.

Для однозначності сприйняття величини параметра число можна доповнити зазначенням його одиниці вимірювання. Налаштувати відображення одиниці вимірювання можна параметром **Unit** розділу **Appearance** властивостей елемента **I/O field**. Записуємо значення розмірності для тиску в системі як **bar**. Оскільки даний параметр оператор не може редагувати, варто його відображати у спосіб, що відрізняється від параметрів, які оператор може змінювати. Для позначення параметрів, які недоступні для редагування, будемо їх виводити без рамки (значення ширини **Width** рамки **Border** встановлюємо рівним **0**) та з кольором фону, що відповідає кольору заливки екрана – вказуємо значення кольору **Color** фону **Background**, яке відповідає сірому кольору з кодом **182, 182, 182**.

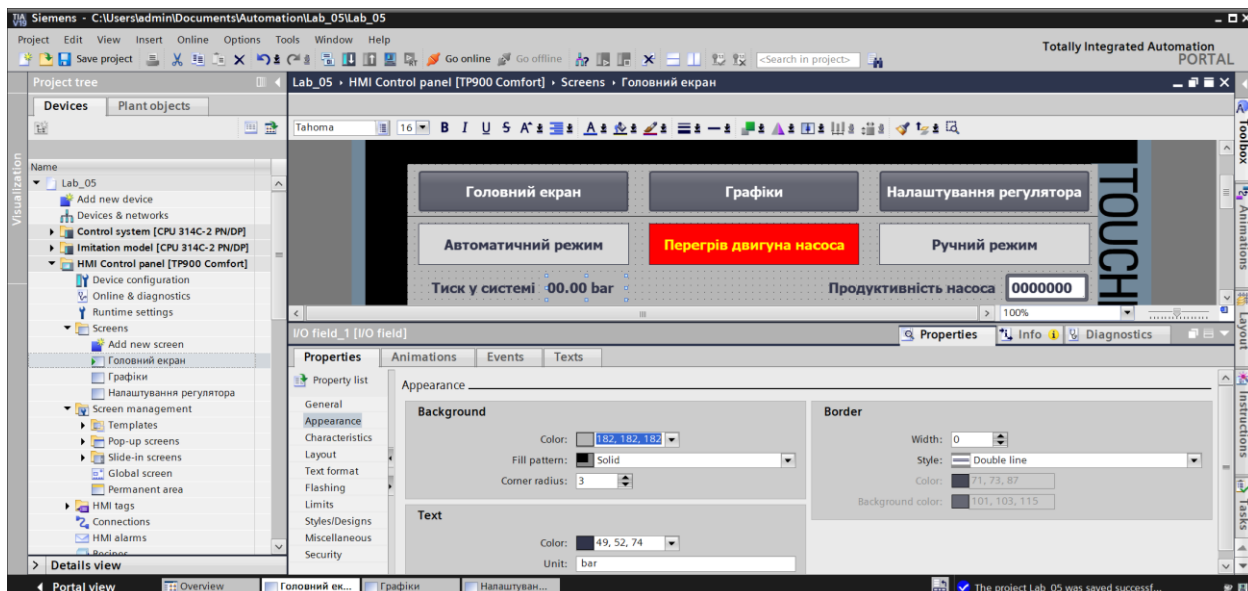


Рисунок 5.32 Налаштування вигляду поля виводу значення тиску в системі.

Після налаштування відображення значення тиску в системі налаштовуємо відображення поля виводу значення продуктивності насоса – параметра, величина якого визначається частотою обертання насоса та відповідає значенню керуючої дії, що дорівнює значенню змінної **rControlAction** блоку даних **PID_Parameters**. Загальні параметри відображення значення продуктивності насоса встановлюємо аналогічно параметрам відображення тиску в системі з зазначенням шаблону відображення **999.99**.

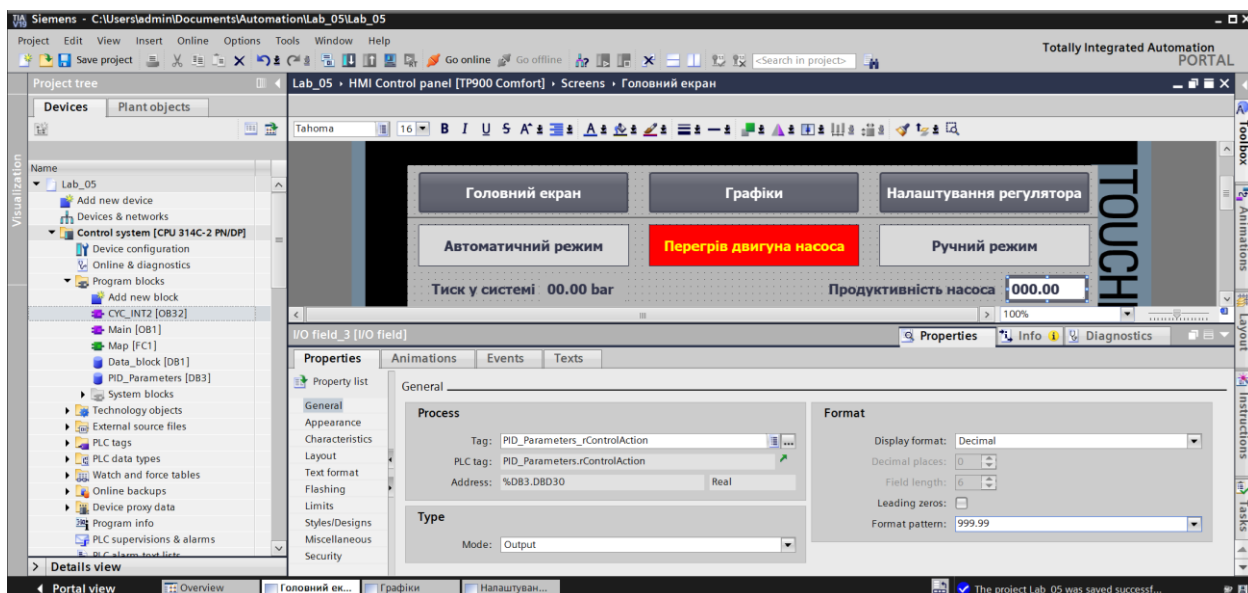


Рисунок 5.33 Налаштування загальних параметрів відображення значення продуктивності насоса.

Налаштовуємо вигляд поля виводу значення продуктивності насоса з одиницею вимірювання **Unit**, що дорівнює **%**.

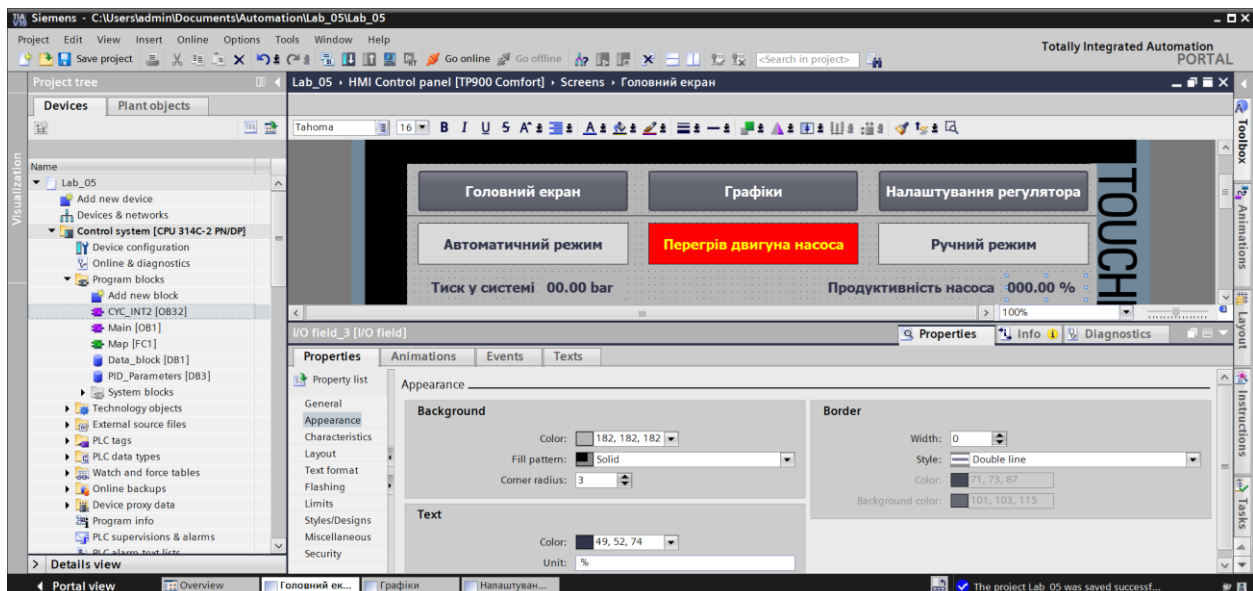


Рисунок 5.34 Налаштування вигляду поля виводу значення продуктивності насоса.

Шаблон відображення значення продуктивності насоса **999.99** з одиницею вимірювання **%** дозволяє відобразити значення в діапазоні **(0.0...100.0) %** з двома знаками після крапки.

Для того, щоб оператор міг задати значення тиску, який регулятор має підтримувати у системі, тип поля **I/O field_2**, що відповідає за відображення цього параметра, потрібно встановити рівним **Input/output**. При активації цього елемента на НМІ панелі з'являтиметься цифрова екранна клавіатура, за допомогою якої оператор має ввести нове значення завдання регулятору.



Рисунок 5.35 Цифрова екранна клавіатура панелі оператора.

Для параметра **Заданий тиск** потрібно встановити прив'язку тегу поля вводу/виводу **I/O field_2** до змінної **rSetpoint** блоку даних **PID_Parameters**, яка є вхідним параметром завдання **SP_INT** регулятора **CONT_C**. Інші основні властивості даного елемента збігаються з параметрами поля відображення тиску в системі **I/O field_1**: формат відображення **Decimal**, а шаблон відображення – **99.99**.

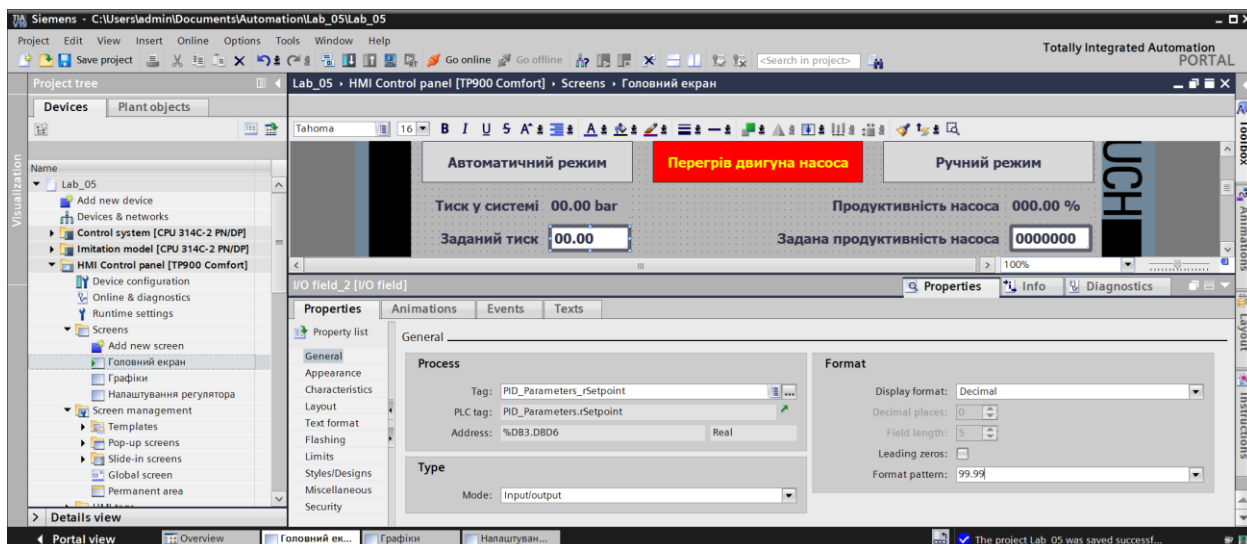


Рисунок 5.36 Налаштування загальних параметрів відображення значення заданого тиску.

Встановлюємо розмірність заданого тиску у параметрі **Unit** на вкладці **Appearance** рівним **bar**. Товщину рамки та колір фону залишаємо без змін, оскільки поле має можливість вводу інформації.

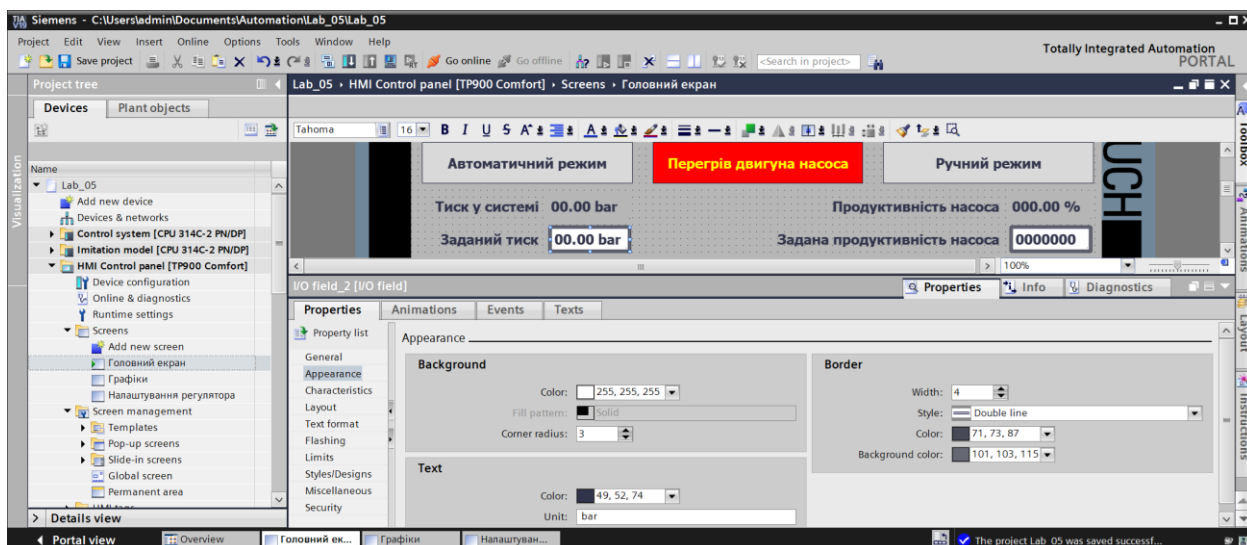


Рисунок 5.37 Налаштування вигляду поля вводу/виводу значення заданого тиску.

Використання у системі панелі оператора дає можливість оператору не просто вмикати та зупиняти насос в ручному режимі, а змінювати продуктивність насосу в цьому режимі без необхідності перепрограмування контролера **Control system**. Для реалізації цієї можливості потрібно надати оператору доступ до редагування параметра **rManualOutput** блока даних **PID_Parameters**.

Налаштовуємо параметри відображення значення продуктивності в ручному режимі з можливістю вводу оператором цього значення. Оскільки продуктивність можна задати в діапазоні **(0.0...100.0) %**, шаблон відображення встановлюємо **999.99**.

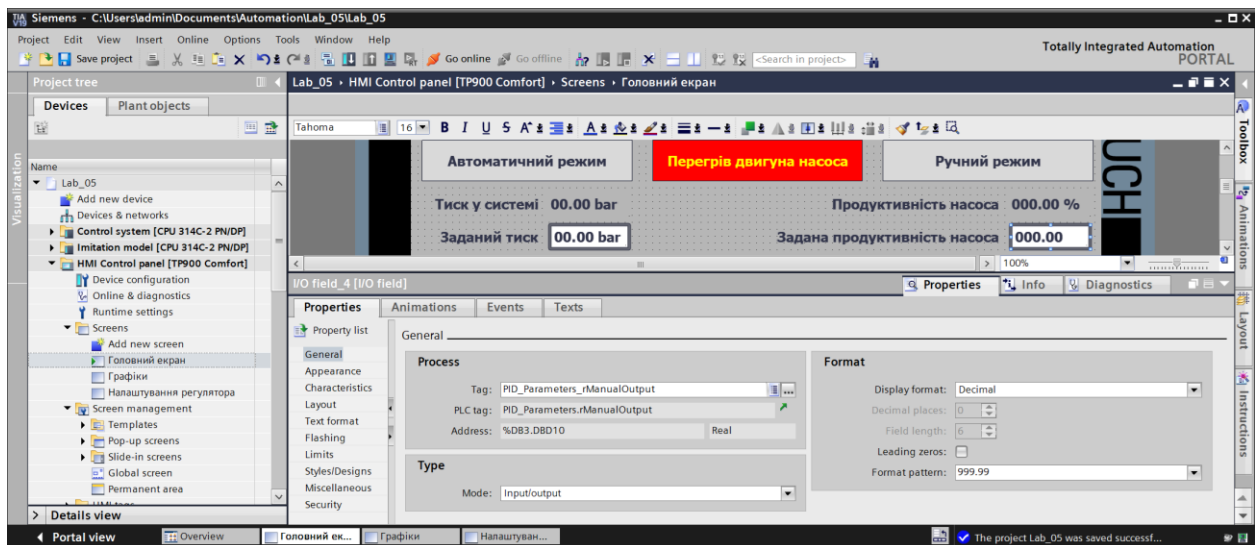


Рисунок 5.38 Налаштування загальних параметрів відображення значення заданої продуктивності насоса в ручному режимі.

За одиницю вимірювання **Unit** встановлюємо **%**, колір фону та рамку навколо поля **I/O field_4** залишаємо.

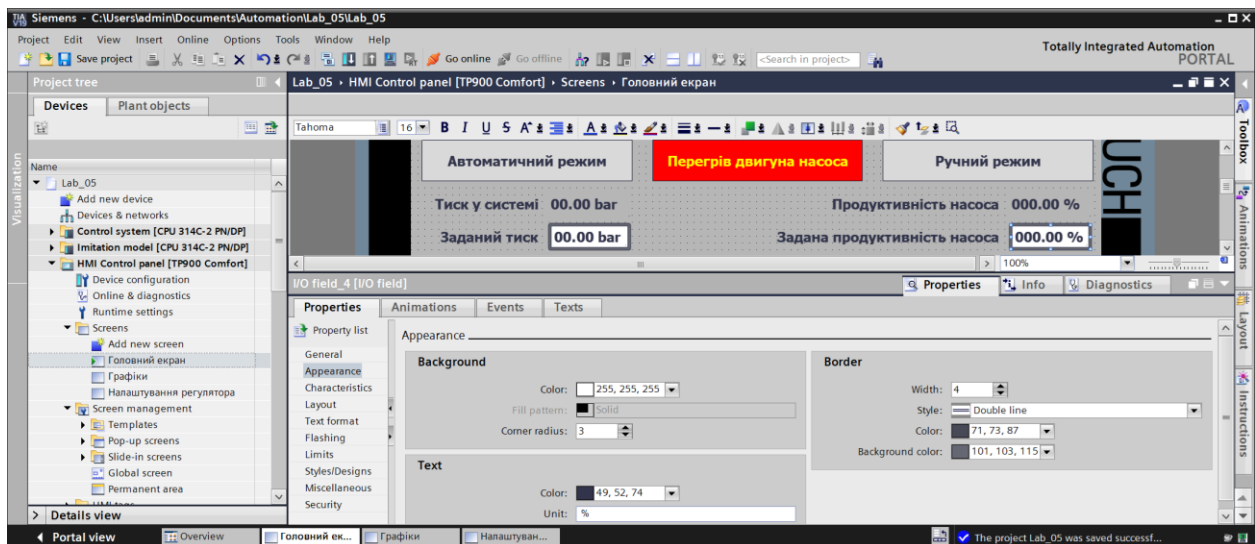


Рисунок 5.39 Налаштування вигляду поля вводу/виводу заданої продуктивності насоса в ручному режимі.

5.2.3 Реалізація можливості пуску та зупинки насосу в ручному режимі за допомогою панелі оператора

Для того, щоб оператор міг подавати команди не тільки кнопками **Пуск** та **Стоп**, які підключені до дискретних входів контролера, але й за допомогою кнопок на панелі оператора, необхідно доповнити алгоритм роботи контролера **Control system**.

Зміни полягають у необхідності додати до блоку даних **Data_block** змінні, які використовуватимуться для передачі сигналів запуску та зупинки насоса з панелі оператора **HMI Control panel**: для команди пуску насоса з панелі оператора додаємо змінну **xStart** типу **Bool**, а для команди зупинки насоса з панелі оператора – змінну **xStop** також типу **Bool**, записуємо у коментарі до змінних їх призначення для полегшення подальшої роботи з алгоритмом.

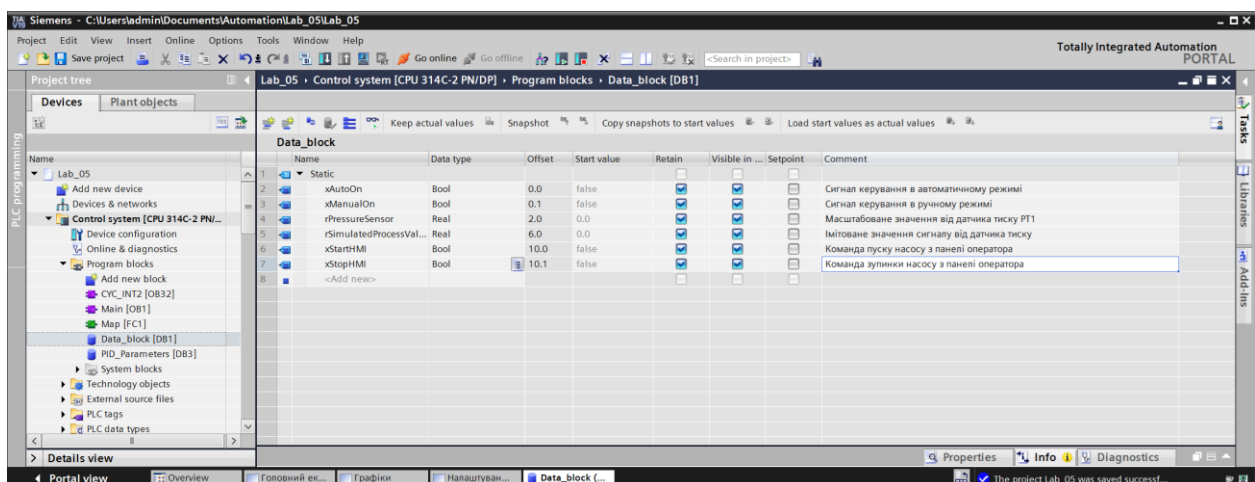


Рисунок 5.40 Додавання змінних для керування насосом з панелі оператора.

Вносимо зміни до алгоритму керування насосом у ручному режимі в організаційному блоці **Main [OB1]** контролера **Control system**:

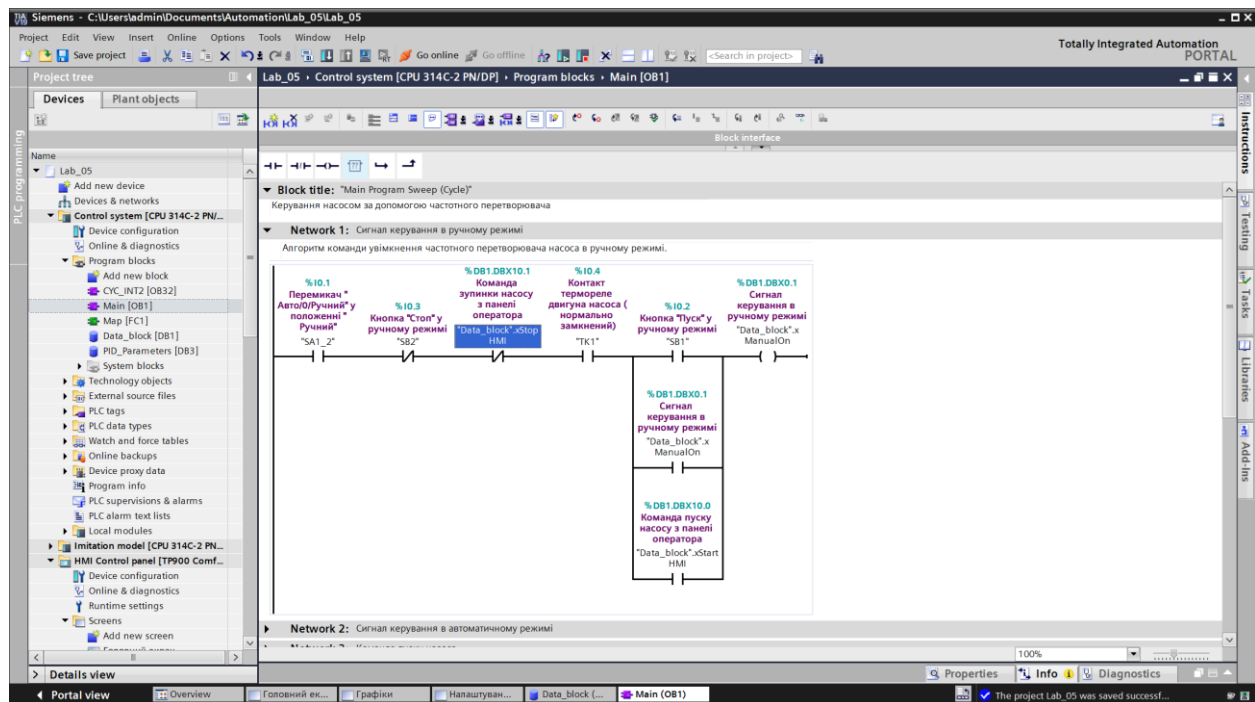


Рисунок 5.41 Зміна алгоритму керування насосом в ручному режимі.

Паралельно до нормально розімкненого контакту, що відповідає кнопці “Пуск” **SB1** встановлюємо нормально розімкнений контакт, що відповідає команді пуску насосу з панелі оператора **xStartHMI**. Таким чином, контролер зможе обробляти ці дві команди незалежно одна від одної і пуск насоса можна буде здійснювати як за допомогою кнопки **SB1**, так і за допомогою панелі оператора. Для обробки сигналу зупинки насоса з панелі оператора потрібно буде встановити нормально замкнений контакт, що відповідає команді пуску насосу з панелі оператора, після нормально замкнутого контакту кнопки **SB2**, яка подає сигнал зупинки насосу. Таким чином, кожен з цих нормально замкнених контактів при появі сигналу буде розривати ланцюг подачі сигналу на ввімкнення насосу в ручному режимі незалежно один від одного.

Для того, щоб оператор міг подавати команди за допомогою НМІ панелі необхідно вивести на екран елементи керування. Традиційним варіантом є використання кнопок. Для подачі команд пуску та зупинки насосу в ручному режимі розташуємо на екрані дві кнопки **Button** з набору інструментів **Elements** та назовемо їх **ПУСК** та **СТОП** відповідно. Розташуємо кнопки під зображенням насоса для того, щоб оператор міг бачити результат подачі команди на екрані. Розмір кнопок повинен бути таким, щоб оператор міг без помилки її натиснути навіть якщо має на руках рукавиці.

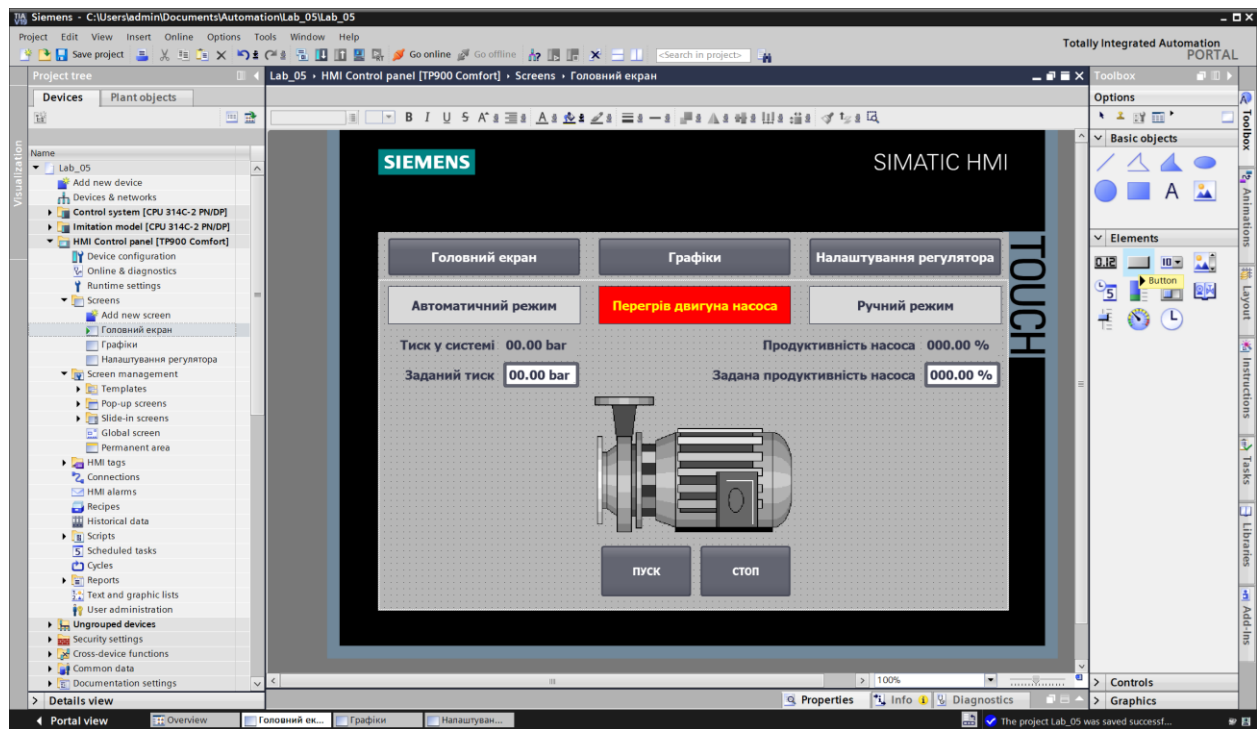


Рисунок 5.42 Додавання кнопок керування насосом в ручному режимі.

Натискання оператором кнопки на HMI панелі повинно вмикати відповідний тег, а при відпусканні кнопки тег повинен вимикатись для того, щоб не втрачалась можливість здійснення керування кнопками, що підключені до дискретних входів контролера у разі вимкнення HMI панелі.

Для обробки натискання кнопки **ПУСК Button_1** потрібно перейти на вкладку налаштування подій **Events** властивостей цієї кнопки, обрати тип події, на яку повинна реагувати панель оператора. Для кнопки такими подіями можуть бути **Click** (натискання кнопки з наступним відпусканням), **Press** (натискання), **Release** (відпускання), **Activate** (вибір кнопки на екрані без натискання, наприклад, за допомогою підключеної до HMI панелі клавіатури), **Deactivate** (переведення вибору на інший об'єкт) або **Change** (будь-яка зміна об'єкта). Для коректної подачі сигналів пуску та зупинки двигуна для кожної з кнопок необхідно буде визначити дії, які будуть виконуватись при натисканні на кнопку та при її відпусканні. Вибір події **Click** для цієї задачі не підходить, оскільки кожне натискання на кнопку інтерфейсу оператора має подавати сигнал, а відпускання кнопки – припиняти подачу сигналу, щоб залишалась можливість паралельного використання кнопок, що підключені до дискретних входів контролера. Для обробки натискання на кнопку **СТАРТ** обираємо подію **Press**, у правій частині вікна розкриваємо список функцій **Add function** та обраємо функцію **SetBit** з групи функцій **Edit bits**. Ця функція при натисканні кнопки **ПУСК** буде встановлювати значення логічної одиниці **TRUE** тегу, який до неї буде прив'язаний.

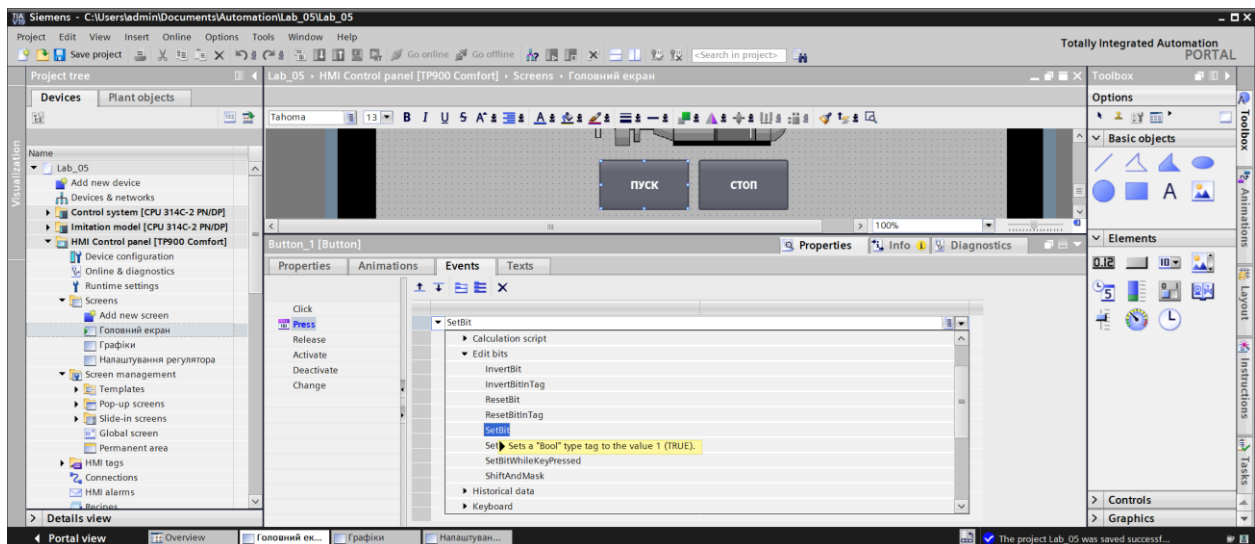


Рисунок 5.43 Налаштування обробки натискання кнопки ПУСК.

Після вибору функції необхідно вказати тег, який буде її аргументом. Для кнопки СТАРТ потрібно вказати змінну **xStartHMI** з блоку даних **Data_block** контролера **Control system**. Таким чином, щойно оператор натисне на екрані кнопку ПУСК, значення цієї змінної стане **TRUE** (логічна одиниця) і відповідний змінній **xStartHMI** нормально розімкнений контакт у секції **Network 1** організаційного блоку **Main [OB1]** контролера **Control system** запустить двигун у ручному режимі якщо такий запуск не блокуватиметься іншими сигналами. Вибір тегу **xStartHMI** необхідно підтвердити натисканням кнопки з зеленою позначкою у нижній частині списку.

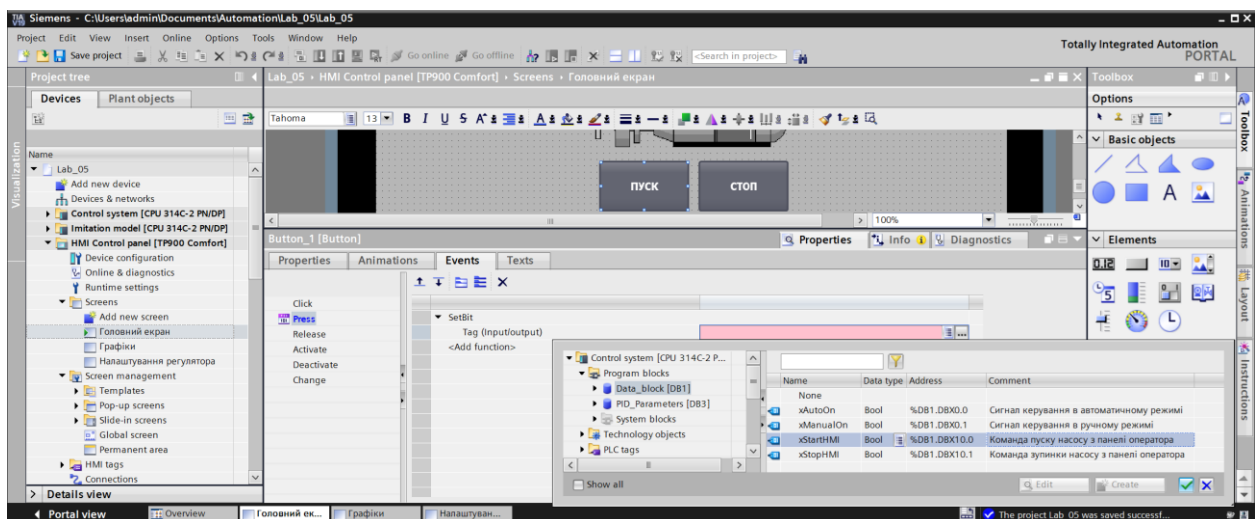


Рисунок 5.44 Визначення тегу для реакції на подію натискання кнопки СТАРТ.

Після підтвердження вибору змінної **xStartHMI** переходимо до налаштування обробки події відпускання кнопки СТАРТ. Вона відбувається аналогічно описаному вище порядку, за виключенням функції – замість функції **SetBit** необхідно обрати у списку функцію **ResetBit**.

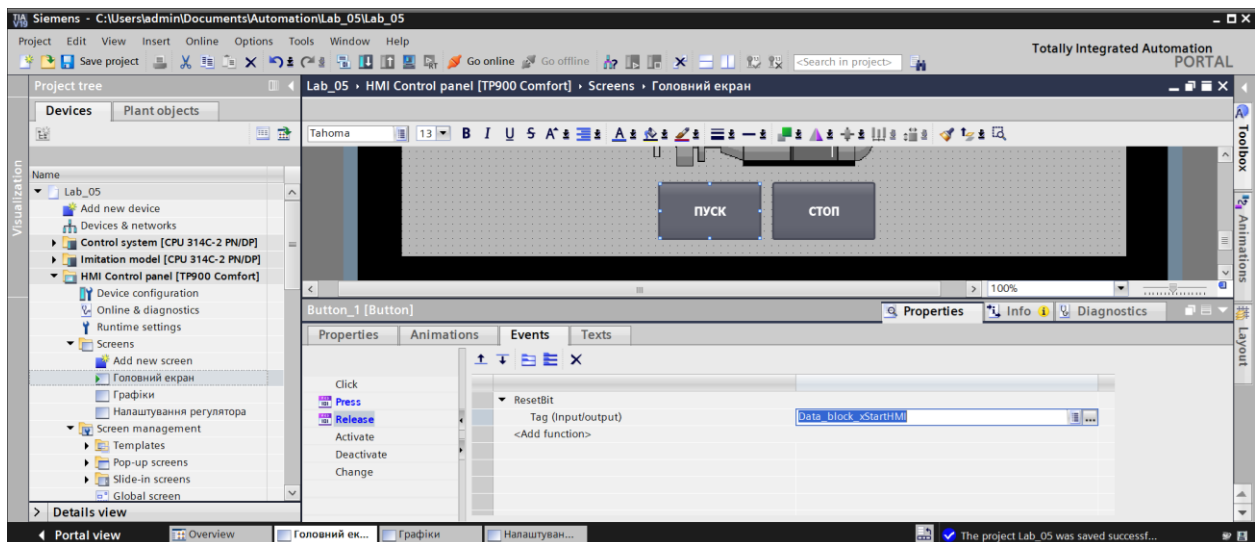


Рисунок 5.45 Налаштування події відпускання кнопки СТАРТ.

Після налаштування подій біля кожної з них з'явиться помітка, а сама подія виділяється жирним синім шрифтом – так можна контролювати для яких подій визначені функції.

Налаштування кнопки **СТОП** відбувають аналогічно до попередніх, за виключенням тегу, який буде змінюватись при натисканні та відпусканні кнопки. Для кнопки **СТОП** – це змінна **xStopHMI** блоку даних **Data_block**.

Обираємо на екрані кнопку **СТОП**, налаштовуємо обробку її натискання.

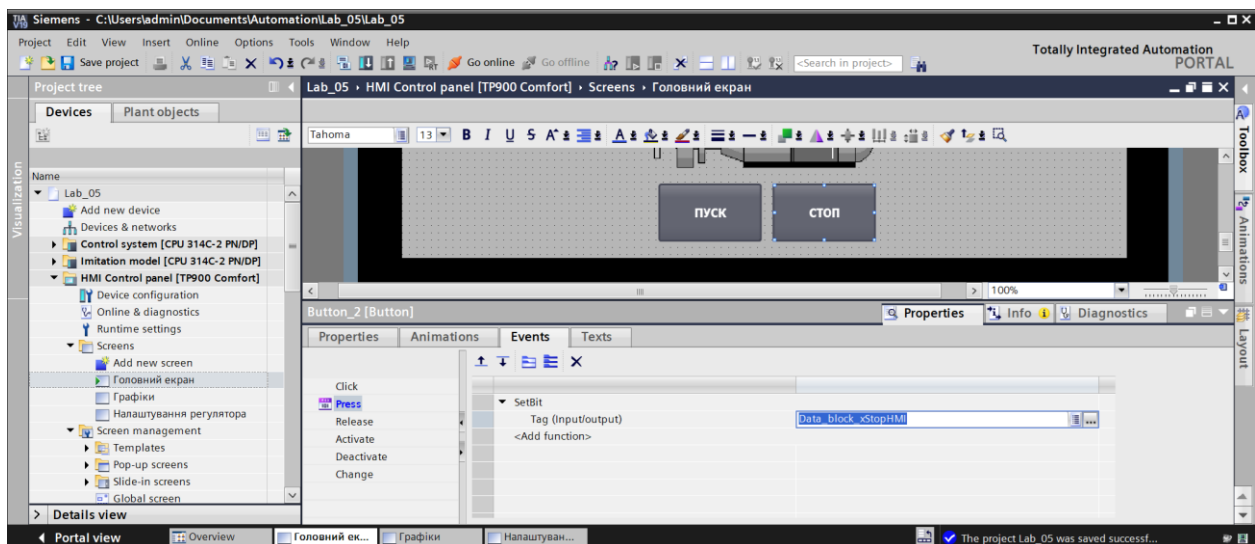


Рисунок 5.46 Налаштування реакції на натискання кнопки СТОП.

Також налаштовуємо обробку відпускання кнопки **СТОП**.

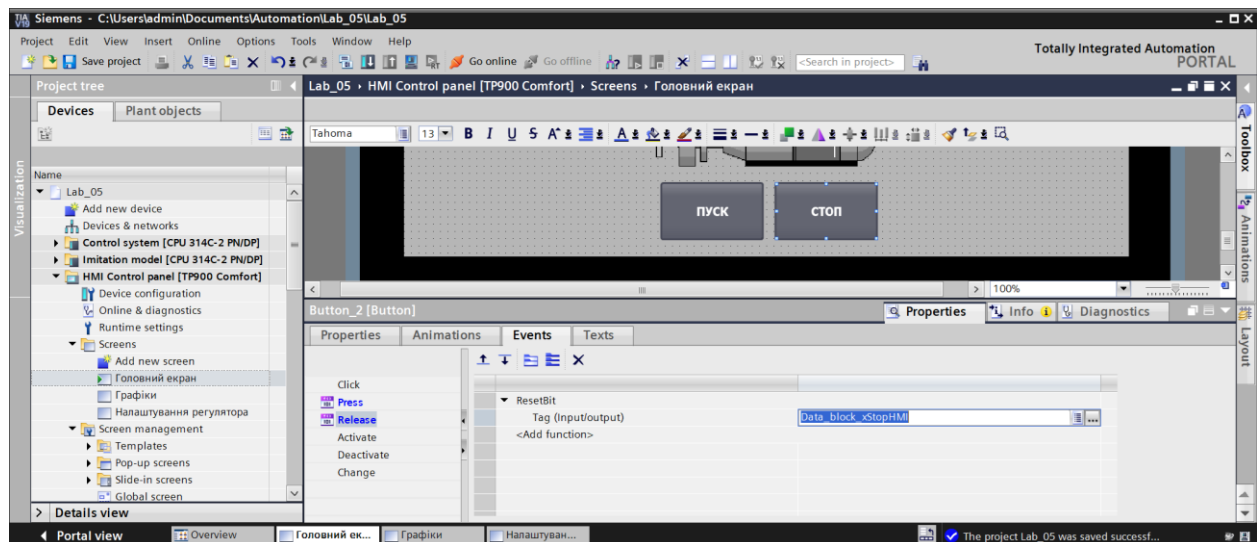


Рисунок 5.47 Налаштування реакції на відпускання кнопки СТОП.

На цьому етапі налаштування головного екрану інтерфейсу можна вважати завершеним.

5.2.4 Відображення графіків перехідних процесів

Безсумнівно, найбільш об'єктивну інформацію про перебіг технологічного процесу та якість роботи системи регулювання можна отримати з графіків перехідних процесів, адже миттєві значення параметрів не дають зрозуміти ані відповідності параметрів регулятора технічному завданню, ані величини відхилення від заданого значення тиску в системі протягом перехідного процесу.

З групи **Controls** набору інструментів **Toolbox** перетягуємо на екран **Графіки** інструмент **Trend view** та збільшуємо його розмір до розмірів всієї робочої області за виключенням постійної області, на якій розташовані кнопки навігації між вікнами.

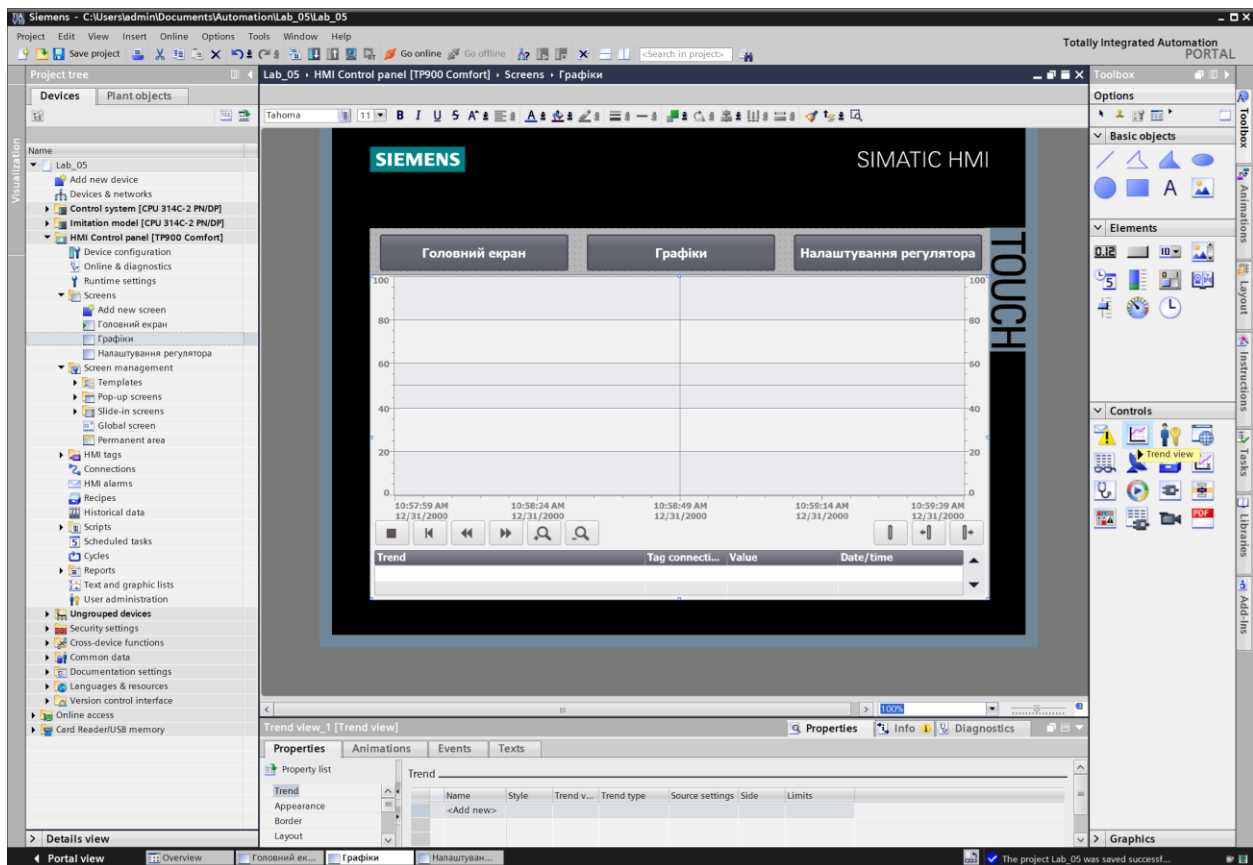


Рисунок 5.48 Додавання інструмента Trend view у проєкт.

На одному полі можна розмістити декілька ліній трендів, їх список та налаштування визначаються таблицею у розділі **Trend** властивостей елементів. Відкриваємо властивості елемента **Trend view_1**, який розмістили на екрані **Графіки** та переходимо на перший розділ **Trend**. Нову лінію тренду можна додати натисканням на поле **<Add new>** таблиці трендів. Додаємо новий тренд та називаємо його **Завдання**.

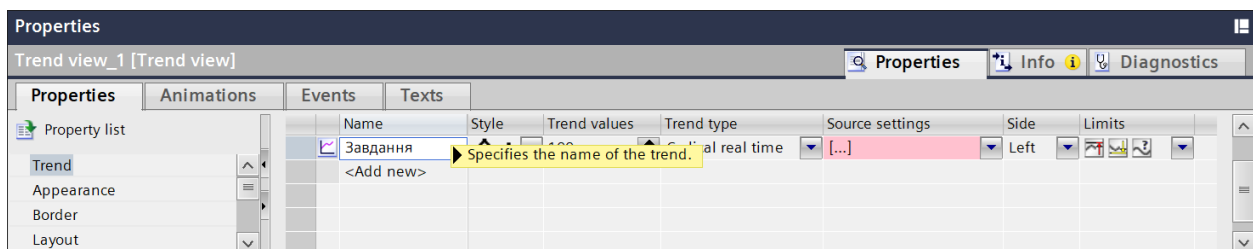


Рисунок 5.49 Додавання нового тренду.

Для кожного тренду необхідно крім його назви задати стиль **Style** лінії на екрані. Діалогове вікно налаштування стилю відкривається кнопкою вибору у колонці **Style**. Стиль складається з режиму відображення, товщини лінії у пікселях, кольору та типу лінії. Режим відображення **Mode** може бути у вигляді стовпчиків **Bar**, інтерпольованим **Interpolated**, при якому лінія тренду будується з'єднанням точок графіку за найкоротшим шляхом, точковим **Points**, при якому

показуються лише точки, та ступінчастим **Stepped**, при якому графік будується прямокутними переходами між точками. Для сигналу завдання в даному випадку буде доцільно використати ступінчастий режим відображення, оскільки значення завдання може змінюватись за допомогою панелі оператора тільки ступінчасто.

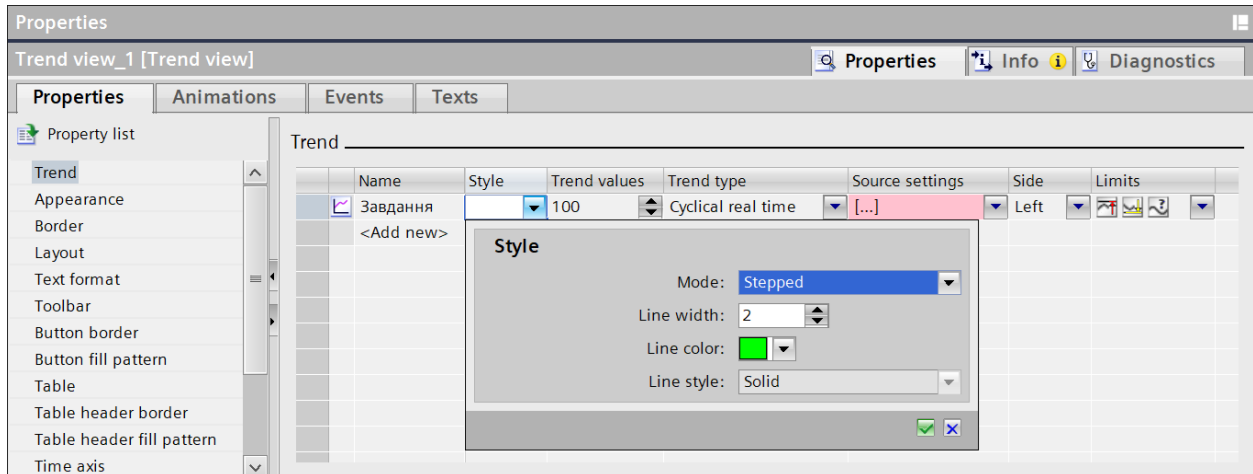


Рисунок 5.50 Діалогове вікно налаштування стилю графіку Завдання.

Наступний параметр налаштування **Trend values** визначає кількість значень тегу, яка виводиться на область побудови графіку. Якщо задана кількість точок буде недостатньою для побудови всього графіку, графік буде побудований лише частково та інформація буде неповною. Максимальне значення цього параметра для даної панелі оператора **999**.

Тип тренду **Trend type** визначає спосіб отримання даних. Для того, щоб панель оператора отримувала дані від контролера максимально швидко залишаємо циклічний спосіб отримання даних у реальному часі **Cyclical real time**.

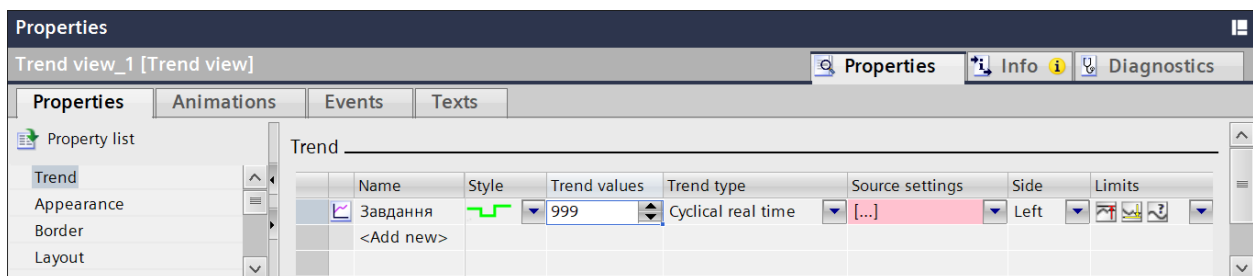


Рисунок 5.51 Встановлення кількості точок графіка.

Параметр **Source settings** визначає джерело отримання даних та період їх оновлення. Для сигналу завдання джерелом даних є змінна **rSetpoint** блока даних **PID_Parameters**, цикл оновлення **Cyclic** встановлюємо мінімально можливим **0.1 с**.

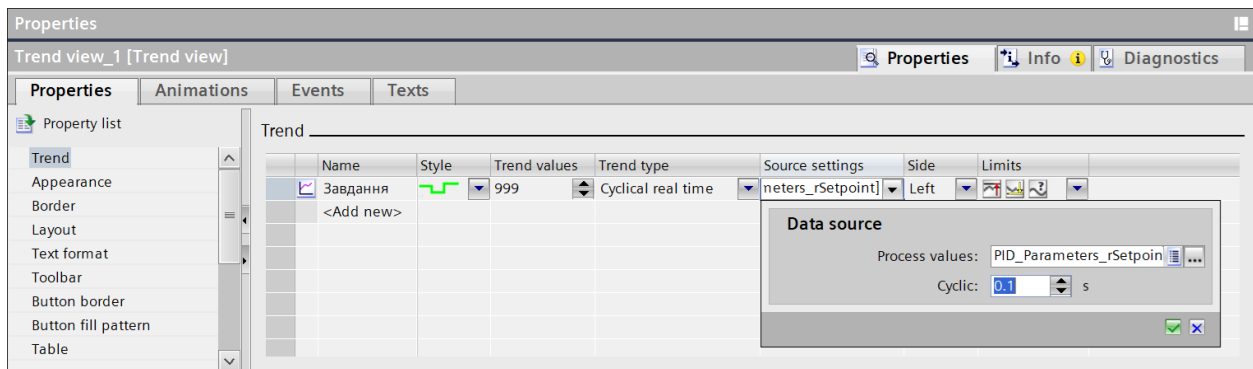


Рисунок 5.52 Налаштування джерела даних для графіка Завдання.

Параметр **Side** дозволяє встановити масштаб графіка відносно одної з двох шкал: лівої або правої. Оскільки ці шкали налаштовуються окремо, на екрані можна представити графіки різних величин для легшого аналізу перехідних процесів у системі. Залишаємо для тренду **Завдання** прив'язку до лівої осі.

Останній пункт налаштування – відображення обмежень тренду. При встановленні обмежень для тегу на панелі оператора можна налаштувати зміну відображення лінії тренду при виході за встановлені межі або при втраті зв'язку з контролером. Залишаємо ці налаштування без змін.

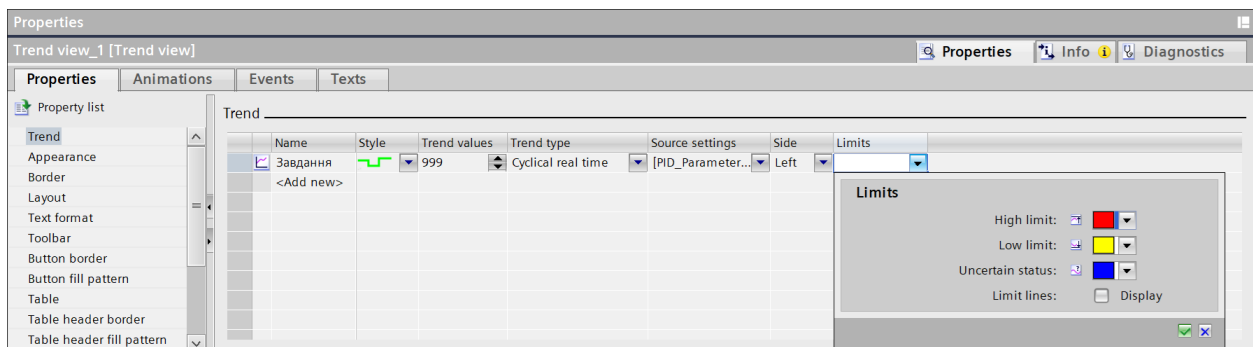


Рисунок 5.53 Налаштування відображення обмежень тренду Завдання.

Таким чином, налаштування відображення лінії тренду **Завдання** завершене. Окрім цього тренду необхідно додати ще два – значення регульованого параметра та значення сигналу керування. Регульований параметр **Тиск** будемо отримувати від імітаційної моделі, тому джерелом для нього буде змінна **rSimulatedProcessValue** з блоку даних **Data_block** контролера **Control system**, для сигналу керування **Насос** джерелом буде значення **rControlAction** блоку даних **PID_Parameters**. Оскільки ці параметри змінюються неперервно, режим відображення **Mode** стилю **Style** для них доцільно встановити інтерпольованим **Interpolated**, товщину ліній – 2 пікселі, тип ліній – суцільний **Solid**, колір регульованого параметра **Тиск** – **синій**, а сигналу керування **Насос** – **червоний**. Оскільки значення тиску та завдання мають одну одиницю вимірювання, встановлюємо прив'язку для тренду **Тиск** аналогічно тренду **Завдання** – до лівої

шкали, а для тренду **Насос**, який має діапазон зміни **(0.0...100.0) %** прив'язку встановлюємо до правої шкали. Інші параметри залишаємо аналогічними тренду **Завдання**.

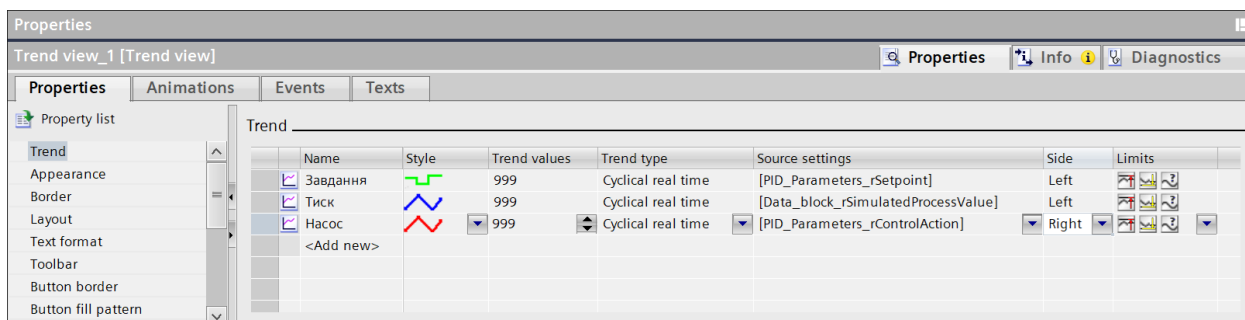


Рисунок 5.54 Налаштування трендів.

Для перегляду точних миттєвих значень параметрів можна вивести таблицю значень під графіками. Налаштування цієї таблиці відбувається у розділі **Table** параметрів налаштування. Для одночасного перегляду всіх трьох значень потрібно встановити значення параметра **Visible lines** рівним 3.

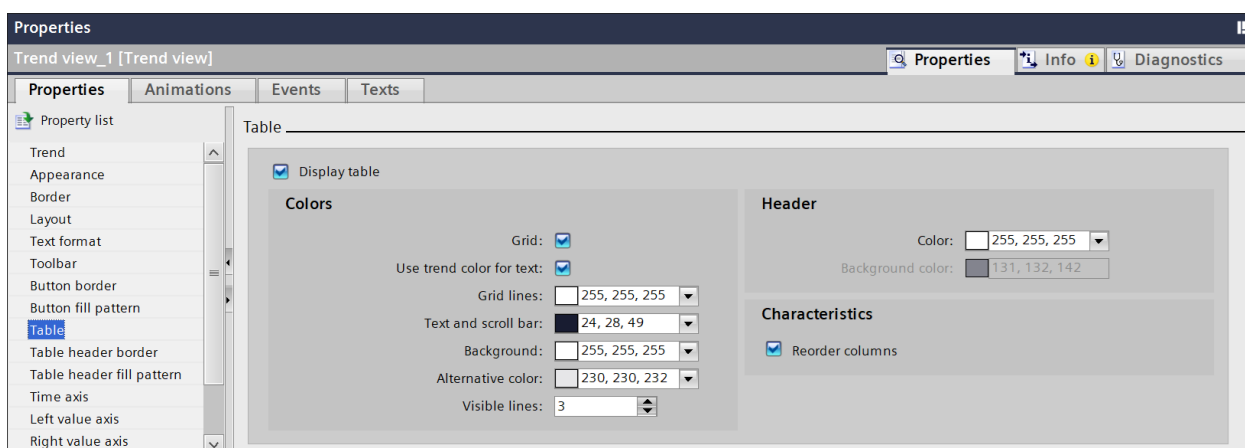


Рисунок 5.55 Налаштування таблиці значень.

Для налаштування лівої осі значень переходимо до розділу налаштувань **Left value axis**. Оскільки попередньо налаштовано, що до лівої осі прив'язані графіки **Тиск** та **Завдання**, які мають діапазон від **0.0** до **10.0**, встановлюємо початкове значення лівої осі **Axis start** рівним **0**, а кінцеве **Axis end** – рівним **10**. Кількість відведених для позначки символів **Label length** встановлюємо рівною **2**, оскільки максимальна позначка для цієї осі дорівнює **10**. Окрім цих параметрів встановлюємо також для лівої осі одиничний крок позначень для числових відміток **Increment** та кількість поділок між ними **Marks** рівною **1**, щоб за графіком можна було легко визначити поточне значення тиску та завдання.

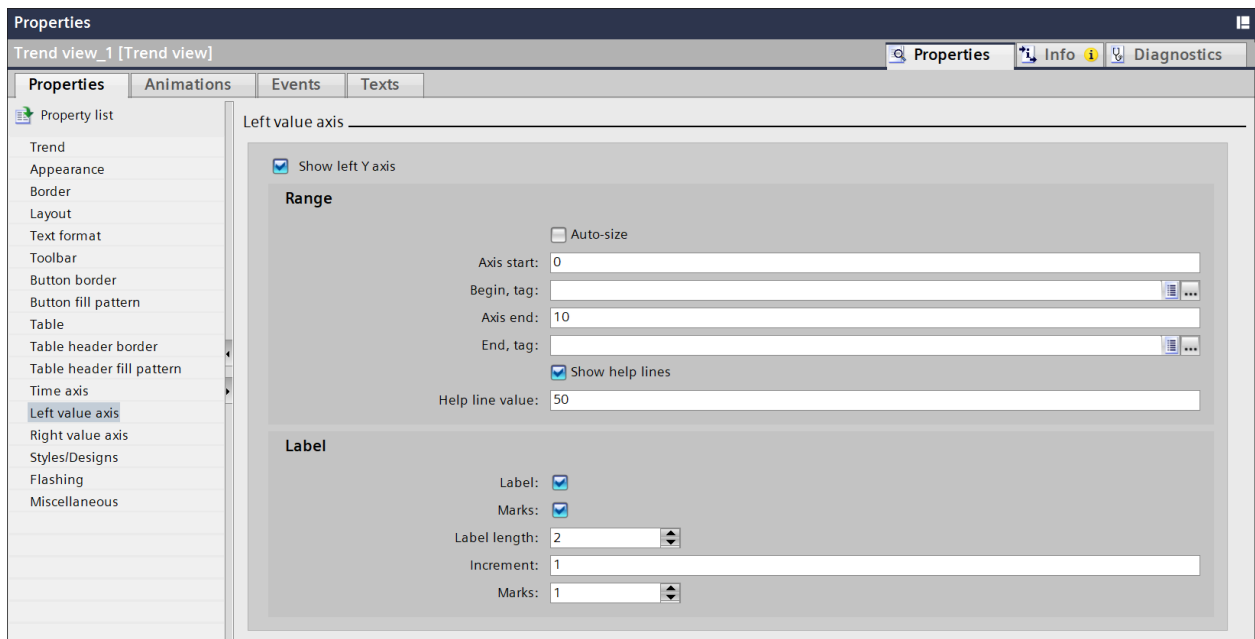


Рисунок 5.56 Налаштування лівої осі графіка.

Налаштовуємо параметри відображення правої осі.

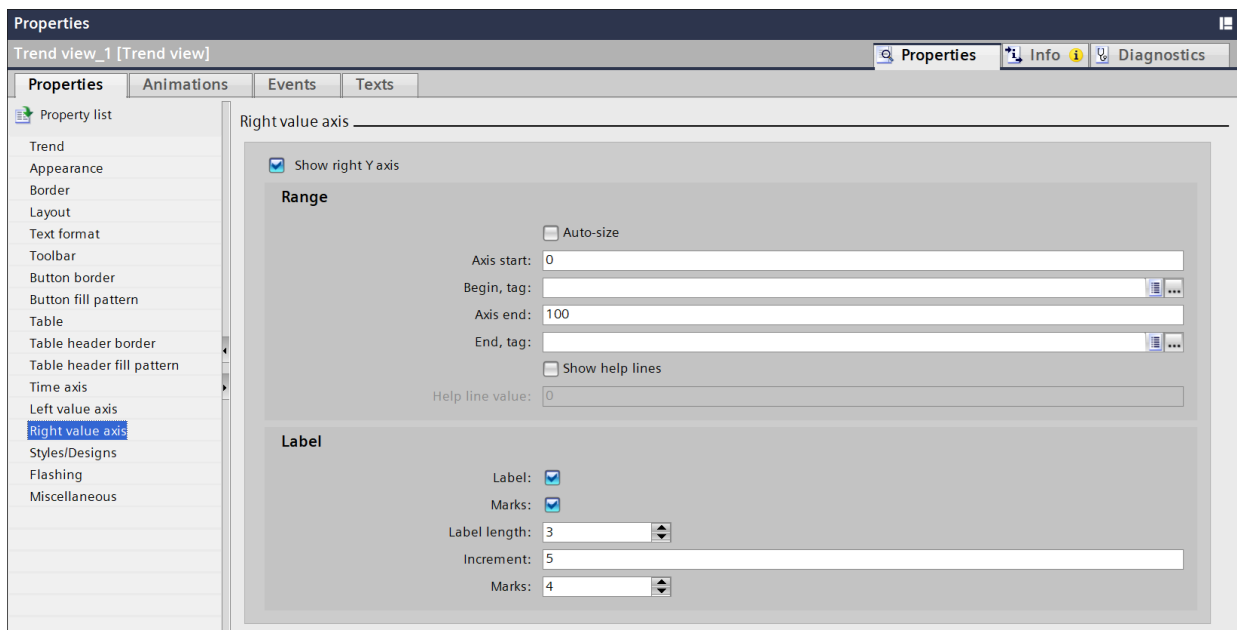


Рисунок 5.57 Налаштування правої осі графіка.

З урахуванням діапазону зміни значення продуктивності насоса від **0 %** до **100 %** встановлюємо початок шкали **Axis start** рівним **0**, а кінець шкали – рівним **100**. Кількість відведених для позначки символів **Label length** встановлюємо рівною **2**, крок позначень для числових відміток **Increment** рівним **5**, а кількість поділок між ними **Marks** – рівною **4**.

Результат зазначених налаштувань екрану **Графіки** можна побачити на рисунку нижче.

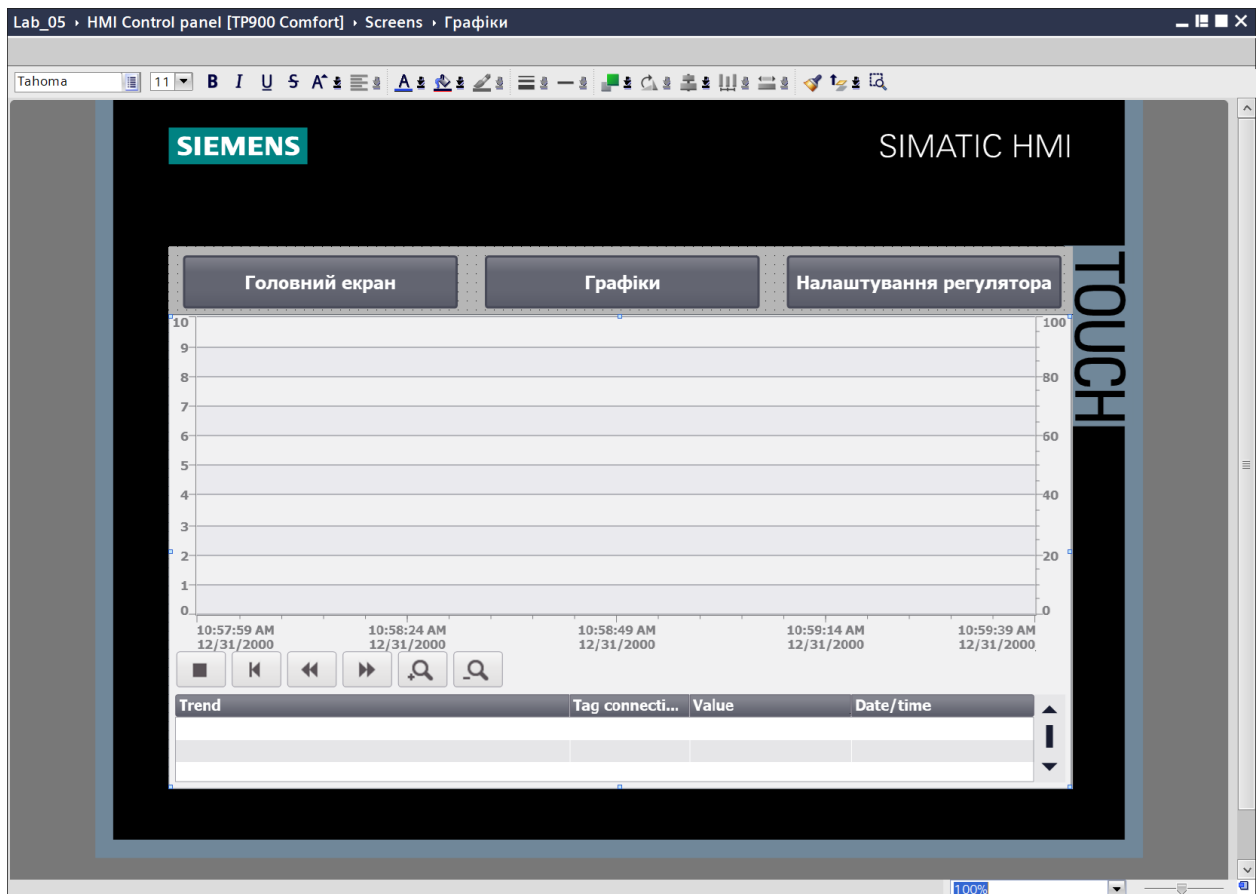


Рисунок 5.58 Вигляд екрану Графіки після налаштувань.

На цьому етапі налаштування екрану Графіки можна вважати завершеним та переходити до екрану налаштування регулятора.

5.2.5 Реалізація можливості налаштування параметрів регулятора за допомогою панелі оператора

Можливість налаштування параметрів регулятора за допомогою панелі оператора значно спрощує та прискорює процес налагодження системи. Також це дозволяє за необхідності провести повторне налаштування регулятора у випадку зміни характеристик об'єкта чи вимог до технологічного процесу без зупинки чи перепрограмування контролера.

Розташовуємо на екрані надписи Text field та поля вводу/виводу I/O field згідно з рисунком нижче.

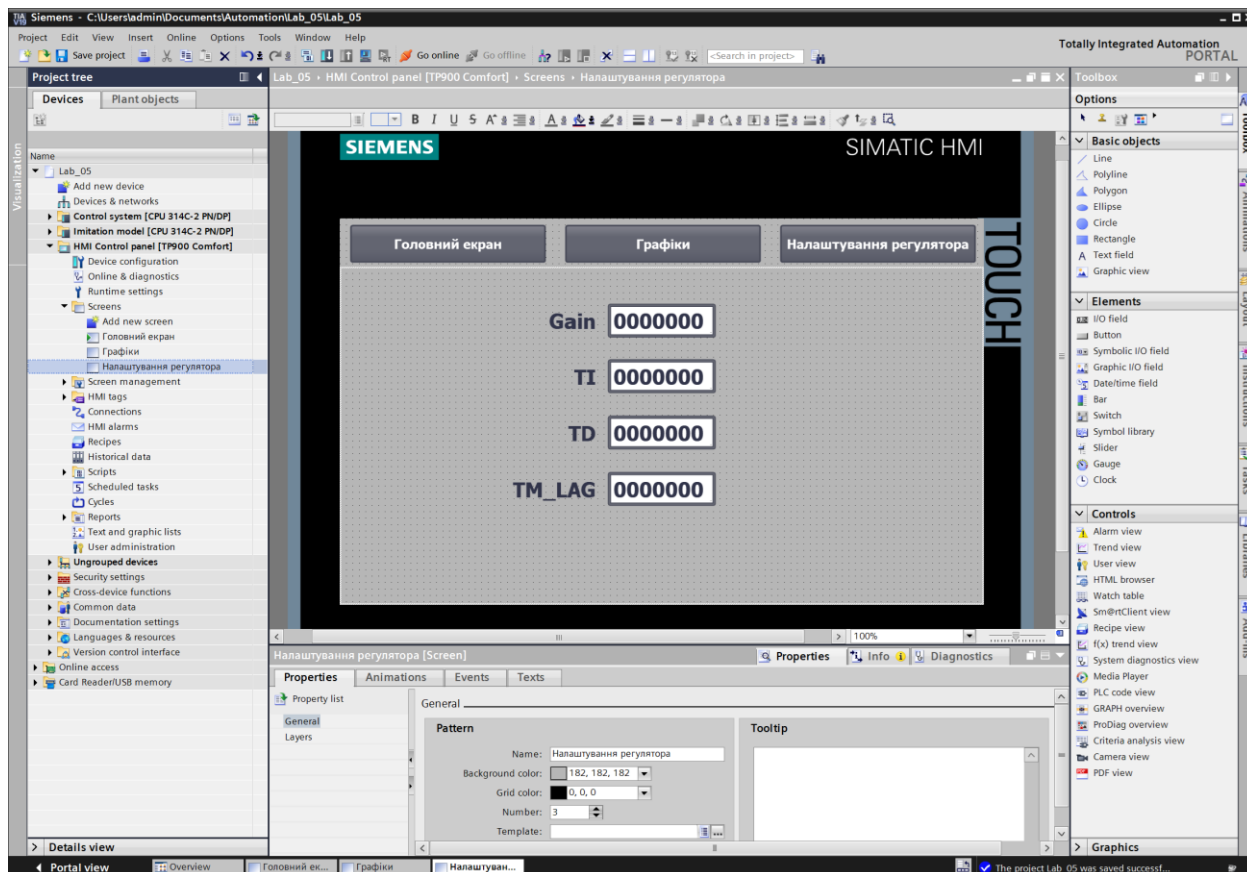


Рисунок 5.59 Встановлення полів для редагування параметрів регулятора.

Налаштування поля **I/O field_1**, яке буде використовуватись для редагування коефіцієнта передачі пропорційної частини регулятора **Gain**, встановлюємо таким чином: значенням поля **Tag** має бути змінна **rProportional** блоку даних **PID_Parameters**, режим **Mode** залишаємо **Input/output**, а формат відображення значення тегу **Format pattern** встановлюємо **999.99**.

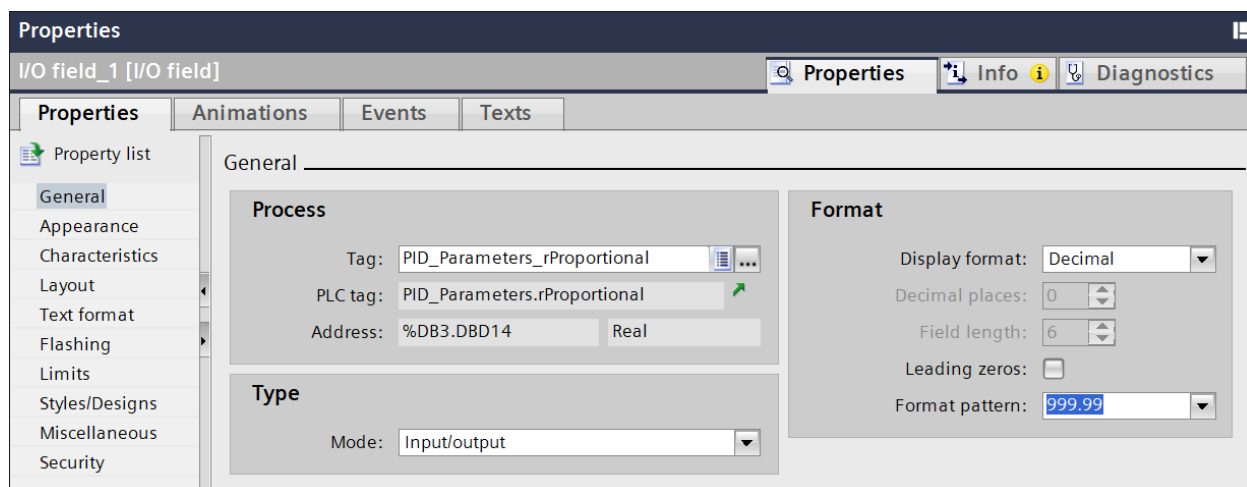


Рисунок 5.60 Налаштування поля вводу/виводу Gain.

Для налаштування поля **I/O field_2**, яке буде використовуватись для редагування сталої часу інтегральної частини регулятора **TI**, встановлюємо такі

значення: у полі **Tag** записуємо змінну **tIntegral** блоку даних **PID_Parameters**, режим **Mode** залишаємо **Input/output**, у форматі відображення значення тегу **Format pattern** встановлюємо значення **999999**.

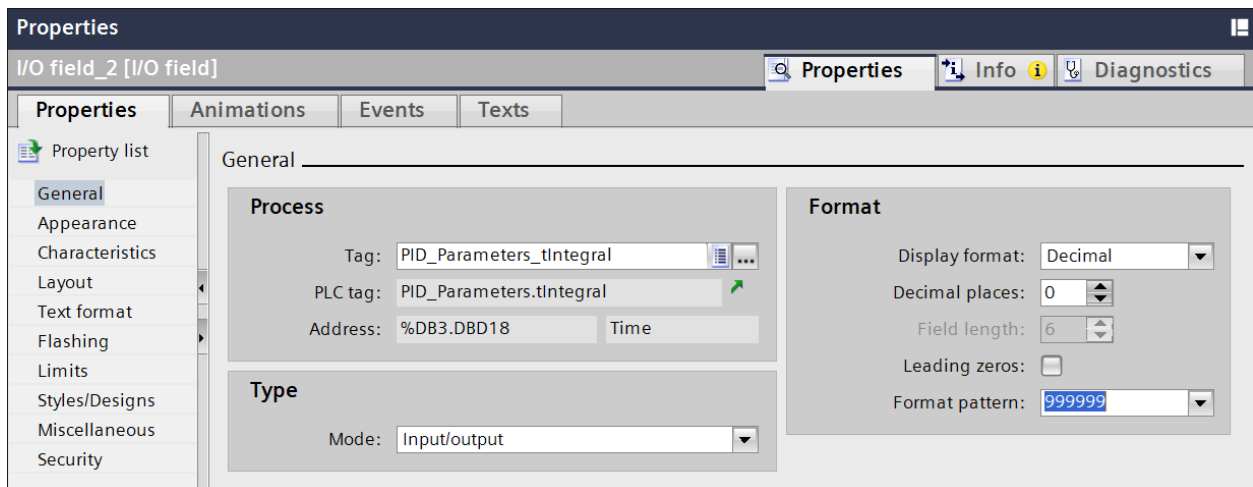


Рисунок 5.61 Налаштування поля вводу/виводу ТІ.

Оскільки змінна **tIntegral** має тип даних **Time**, її значення зберігається у пам'яті контролера як ціле число **мілісекунд**. Для того, щоб оператор враховував, що значення цього параметра потрібно задавати у мілісекундах, необхідно встановити йому підказку у вигляді розмірності – параметра **Unit** розділу **Appearance** налаштувань елемента **I/O field_2**.

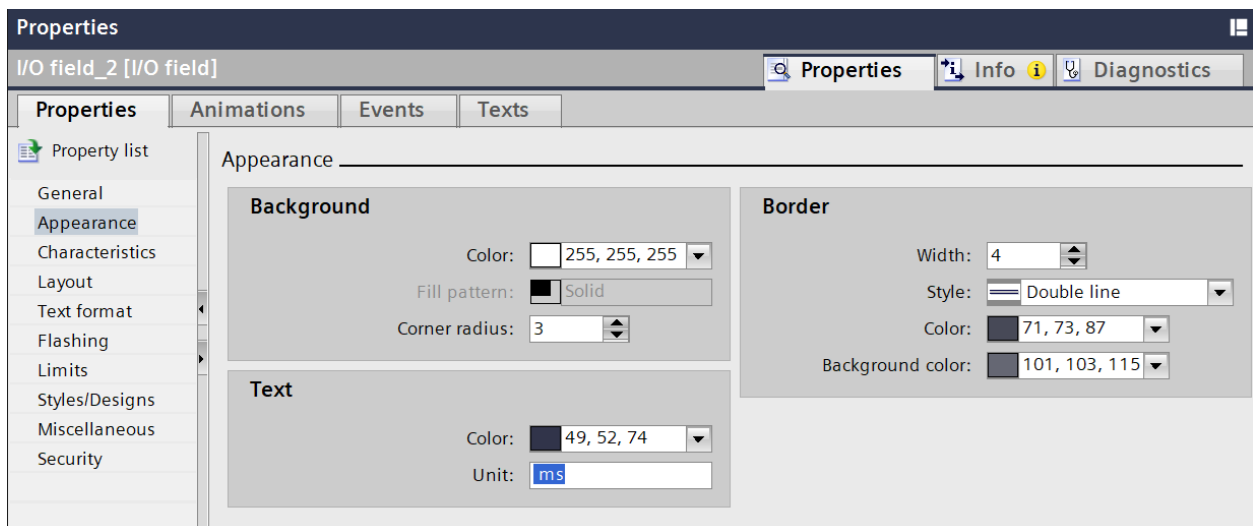


Рисунок 5.62 Налаштування одиниці вимірювання часу інтегрування.

Оскільки параметри **TD** та **TM_LAG** також мають тип даних **Time**, їх налаштування встановлюємо аналогічними за виключенням налаштування поля **Tag** – для елемента **I/O field_3**, що відповідає сталій часу диференційної частини регулятора **TD** це буде змінна **tDerivative** блоку даних **PID_Parameters**.

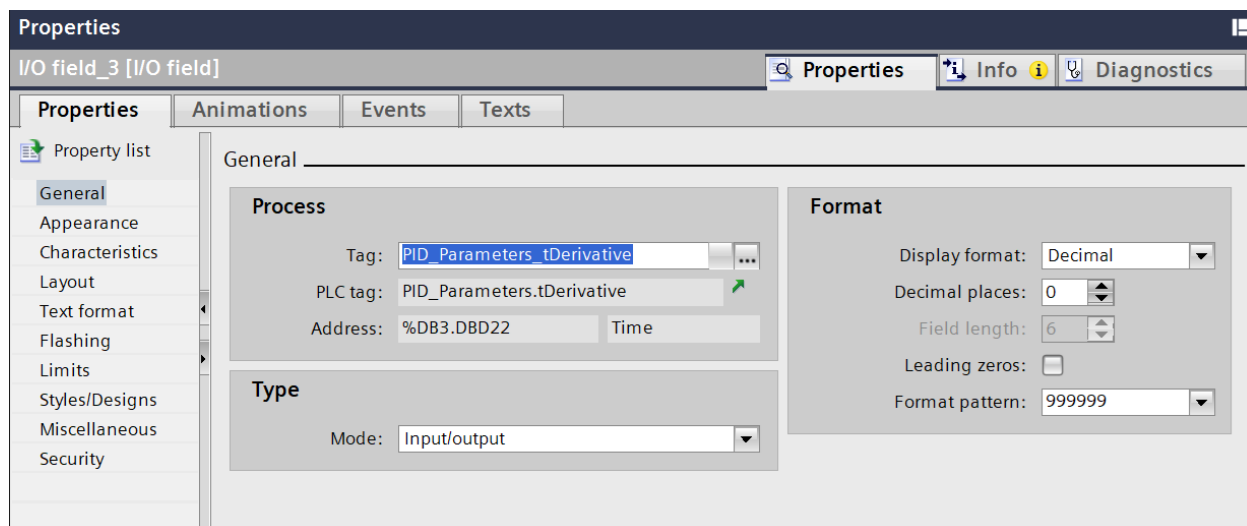


Рисунок 5.63 Налаштування поля вводу/виводу TD.

Для елемента **I/O field_4**, що відповідає налаштуванню інерційності диференційної частини регулятора **TM_LAG**, у полі **Tag** вказуємо змінну **tDerivativeLag** блоку даних **PID_Parameters**.

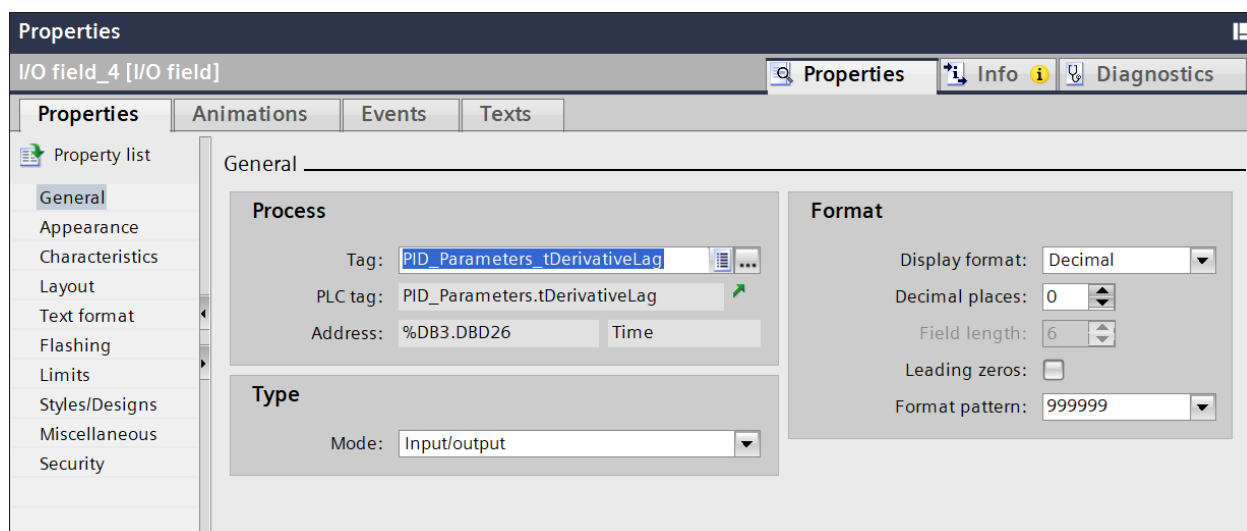


Рисунок 5.64 Налаштування поля вводу/виводу TM_LAG.

Додаємо на екран Налаштування регулятора 3 перемикачі Switch, які відповідатимуть за підключення пропорційної, інтегральної та диференційної частин і розташовуємо їх у нижній частині екрану.

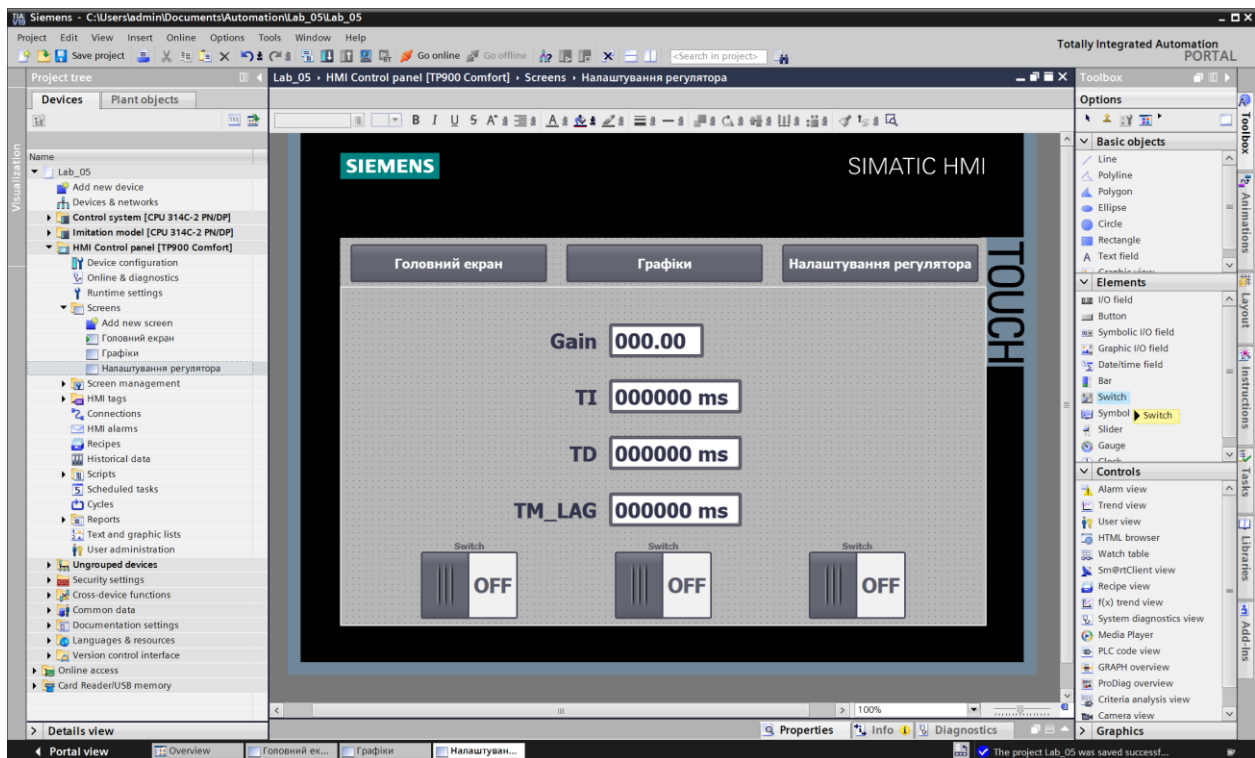


Рисунок 5.65 Розміщення перемикачів на екрані налаштувань регулятора.

Перший перемикач Switch_1 відповідатиме за параметр P_SEL регулятора, який підключає пропорційну частину регулятора. Налаштування цього перемикача встановлюємо таким чином: у полі Tag вказуємо змінну xP_Sel з блоку даних PID_Parameters, що відповідає входу P_SEL регулятора, у заголовку Title, який грає роль підказки оператору і знаходиться над перемикачем, записуємо назву параметра регулятора P_SEL.

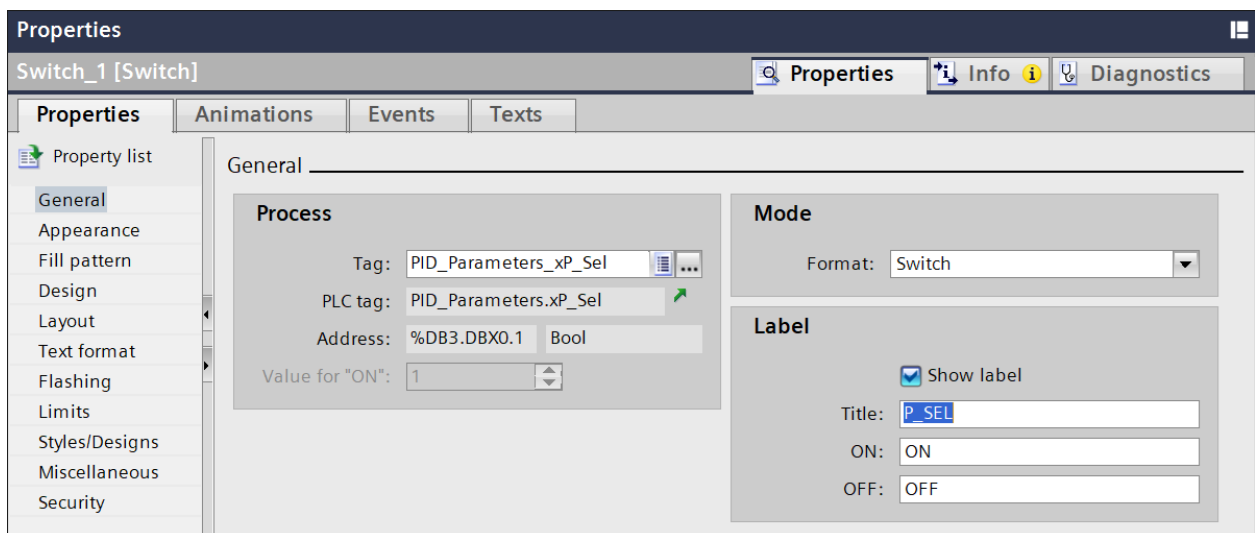


Рисунок 5.66 Основні налаштування перемикача P_SEL.

Розмір шрифту заголовка кнопки краще встановити рівним розміру шрифту підписів його положень. Зробити це можна у полі Header font розділу налаштувань перемикача Text format.

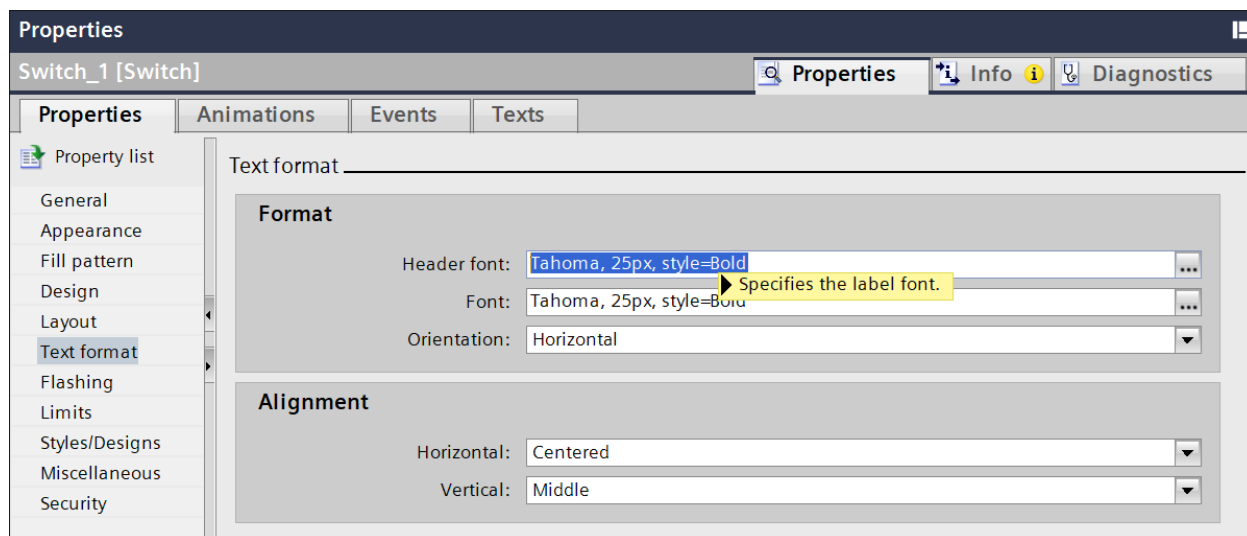


Рисунок 5.67 Налаштування шрифту перемикача.

Другий перемикач **Switch_2** буде відповідати за параметр **I_SEL** регулятора, який підключає інтегральну частину регулятора. Для цього перемикача встановлюємо у полі **Tag** змінну **xl_Sel** з блоку даних **PID_Parameters**, що відповідає входу **I_SEL** регулятора, у заголовку **Title** записуємо назву параметра регулятора **I_SEL**.

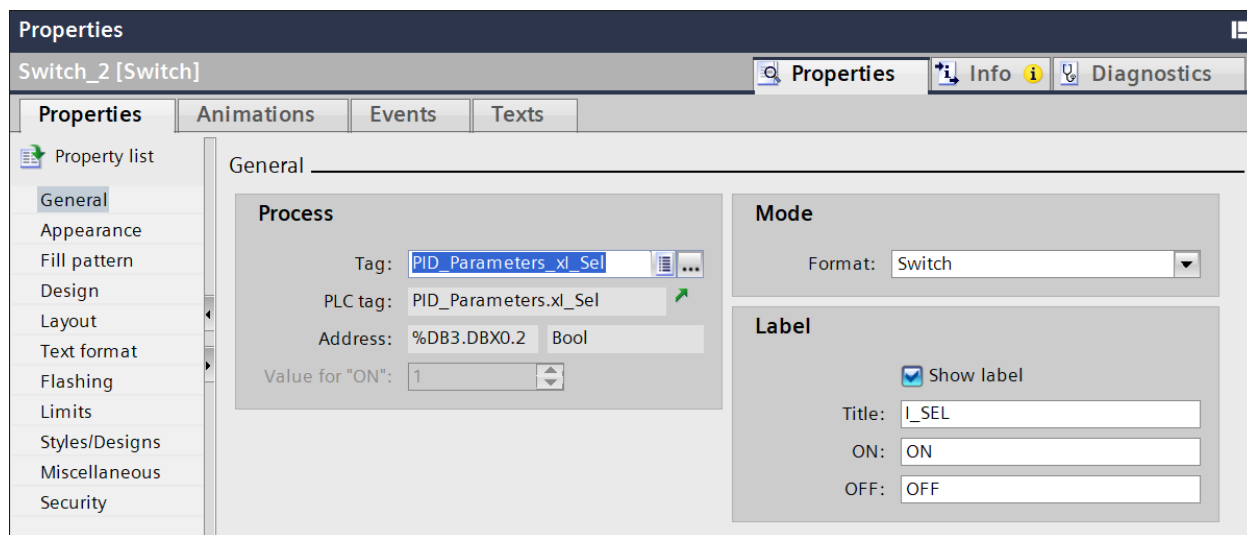


Рисунок 5.68 Основні налаштування перемикача **I_SEL**.

За параметр **D_SEL** регулятора, який підключає диференційну частину регулятора, буде відповідати третій перемикач **Switch_3**. Для нього встановлюємо у полі **Tag** змінну **xD_Sel** з блоку даних **PID_Parameters**, що

відповідає входу **D_SEL** регулятора, у заголовку **Title** записуємо назву параметра регулятора **D_SEL**.

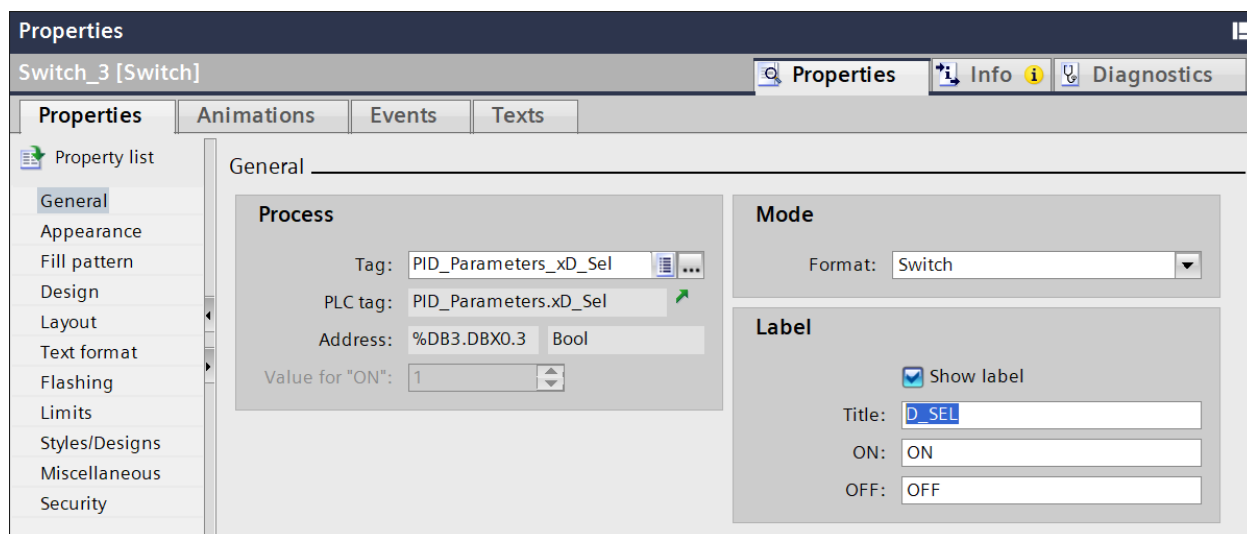


Рисунок 5.69 Основні налаштування перемикача D_SEL.

Кінцевий вигляд екрану Налаштування регулятора наведений нижче на рисунку.

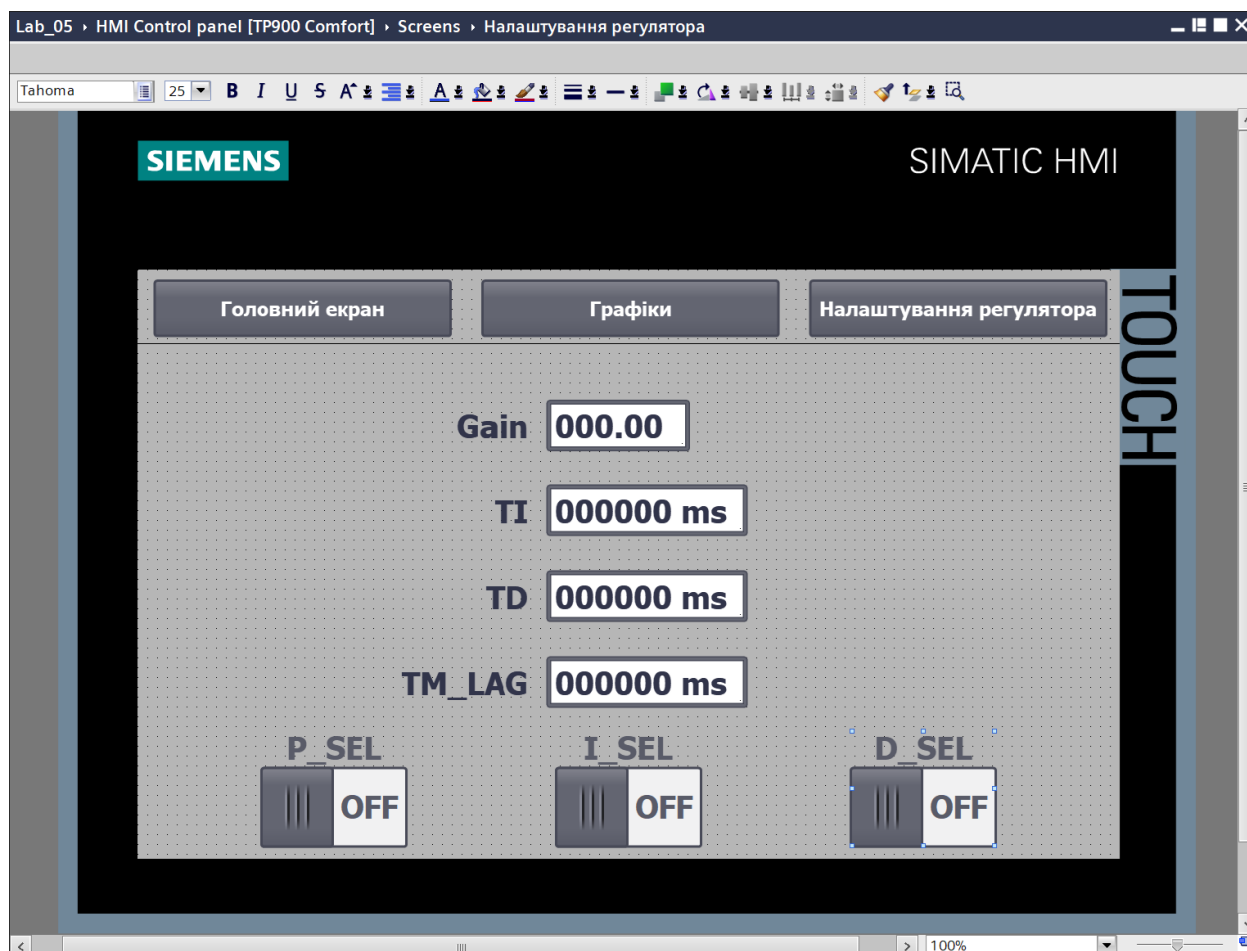


Рисунок 5.70 Вигляд екрану Налаштування регулятора.

На даному етапі проєкт завершений, тож після його збереження можемо переходити до перевірки.

5.2.6 Перевірка роботи системи з панеллю оператора

Для перевірки коректності роботи системи з панеллю оператора необхідно запустити симулятори контролерів **Control system** та **Imitation model**, завантажити в них програми та перевести у режим роботи **RUN** або **RUN-P**. Після запуску симулятора контролера **Control system** необхідно подати сигнали, які відповідають режиму роботи системи, що перевіряється. Для ручного режиму роботи системи це буде дискретний сигнал **SA1_2 (%I0.1)**, відсутність перегріву двигуна імітуємо сигналом **TK1 (%I0.4)**.

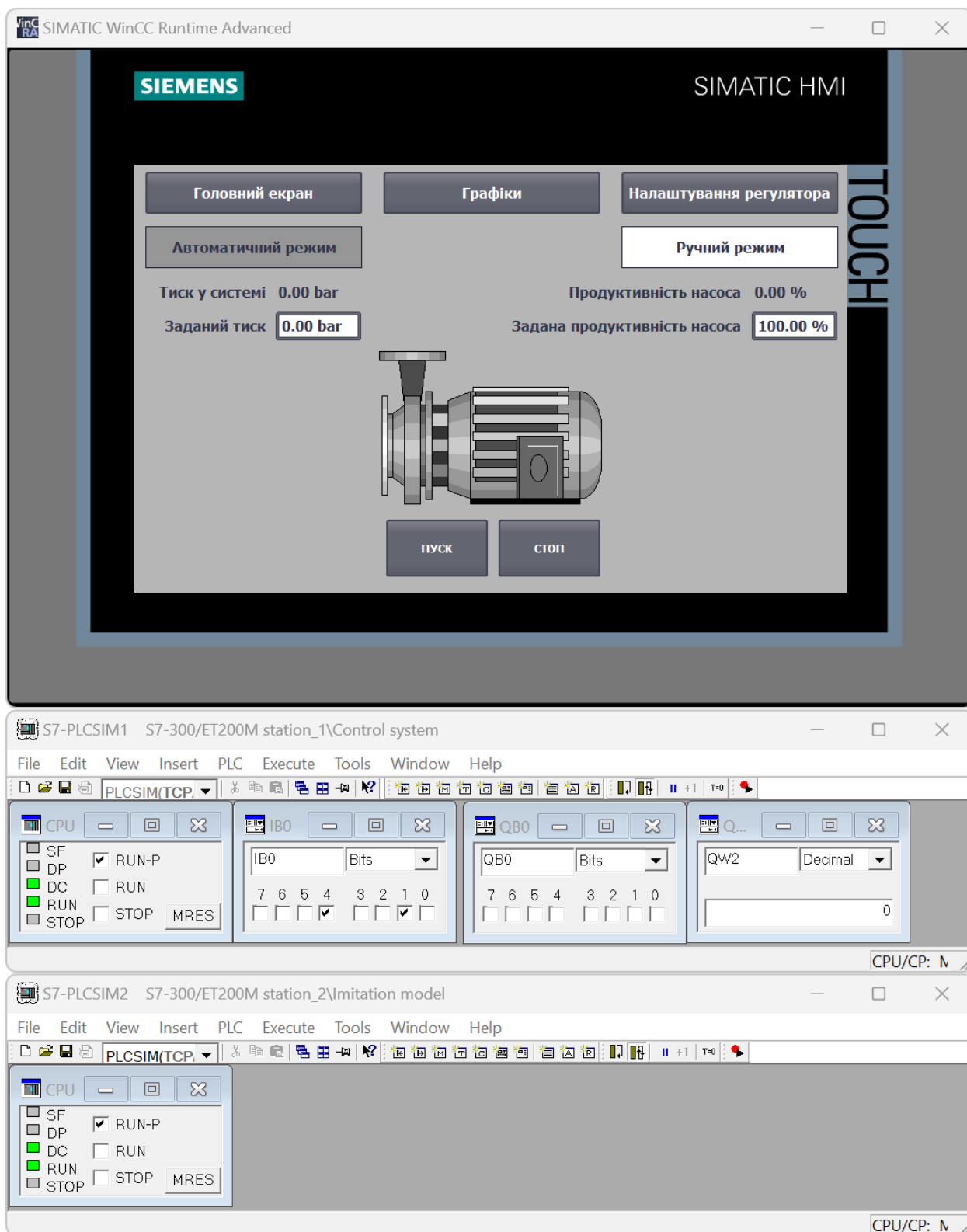


Рисунок 5.71 Запуск симулятора панелі оператора.

Запуск симулятора панелі оператора відбувається так само, як і запуск симулятора контролера – потрібно обрати у дереві проєкта TIA Portal панель оператора HMI Control panel та запустити симулятор одним із способів: через пункт меню TIA Portal **Online** → **Simulation** → **Start** або за допомогою кнопки **Start simulation** панелі інструментів TIA Portal або гарячою комбінацією клавіш

Ctrl + Shift + X. При запуску симулятора панелі оператора відкривається вікно зі стартовим екраном, після отримання симулятором поточних значень змінних вони з'являються на екрані. Для перевірки коректності реалізації алгоритму керування насосом в ручному режимі за допомогою панелі оператора натискаємо кнопку пуск.

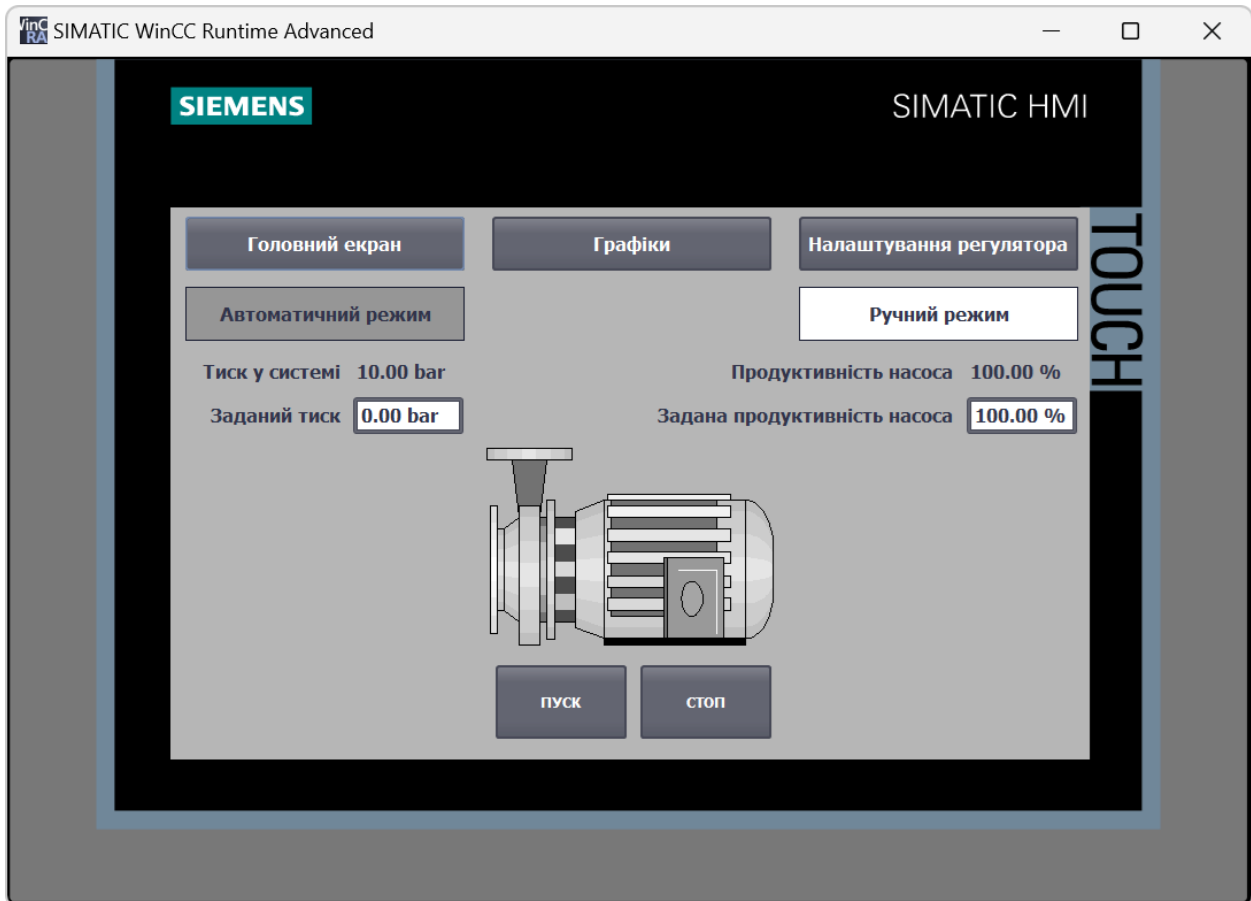


Рисунок 5.72 Реакція системи на команду пуску насоса в ручному режимі.

Значення тиску в системі має поступово вийти на максимальне значення, яке визначається заданою продуктивністю насоса в ручному режимі. Можна переглянути графік перехідного процесу на екрані **Графіки** – для переходу на нього потрібно натиснути кнопку **Графіки** у виділеній області екрану.

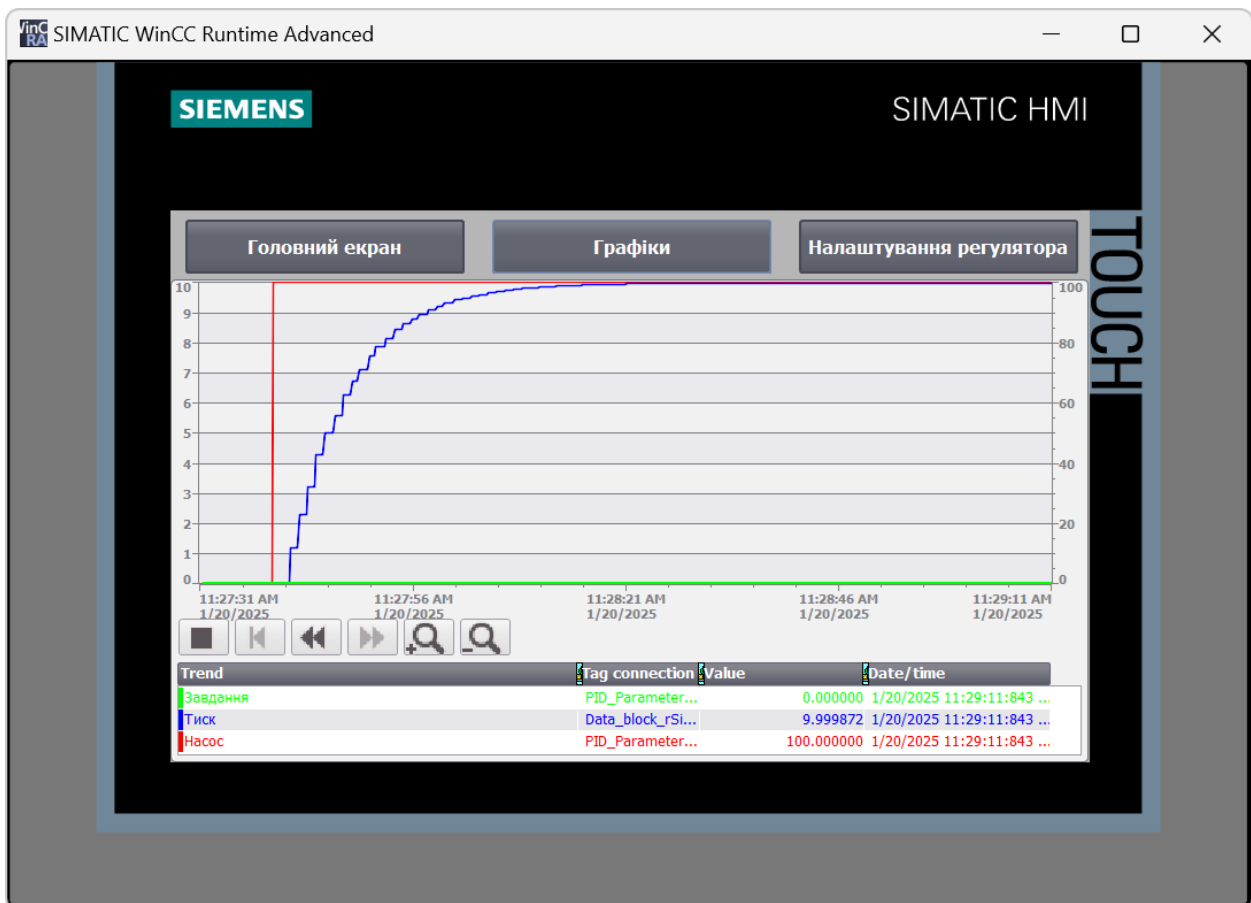


Рисунок 5.73 Графік перехідного процесу в ручному режимі.

Для перевірки роботи системи в автоматичному режимі необхідно вимкнути сигнал ручного режиму **SA1_2 (%I0.1)** та ввімкнути сигнал автоматичного режиму **SA1_1 (%I0.0)**, це має перевести систему в автоматичний режим роботи, в якому можна завдати значення тиску в системі та перевірити вплив параметрів настройки регулятора на перехідний процес в системі. Після встановлення постійного значення тиску наносимо збурення зміною завданням та порівнюємо перехідні процеси за різних налаштувань регулятора.

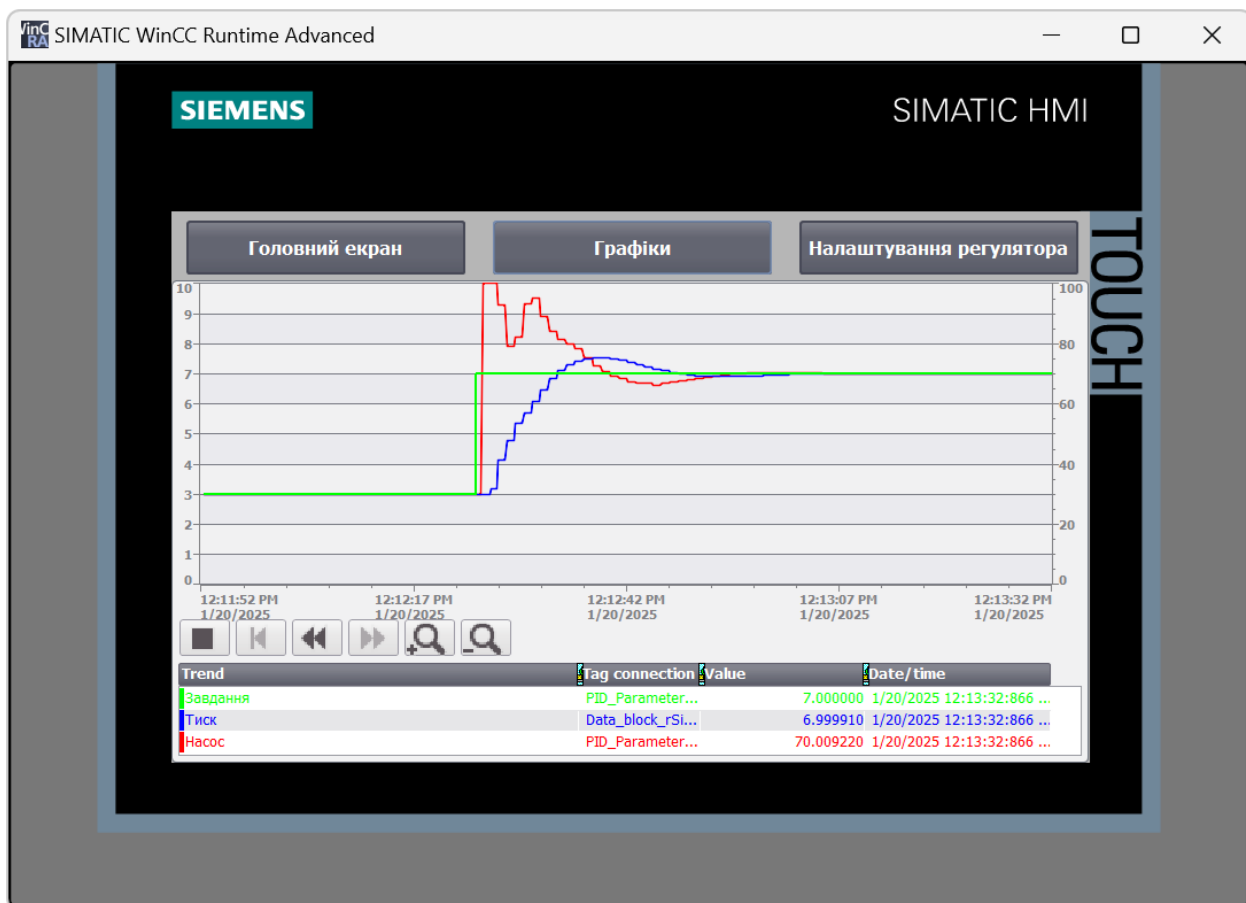


Рисунок 5.74 Графік перехідного процесу в автоматичному режимі.

На отриманих графіках можна помітити, що період оновлень даних на екрані суттєво більший за встановлене значення 0.1 с. Причиною цього є налаштування періоду оновлення HMI тегів, яке за замовченням складає 1 с.

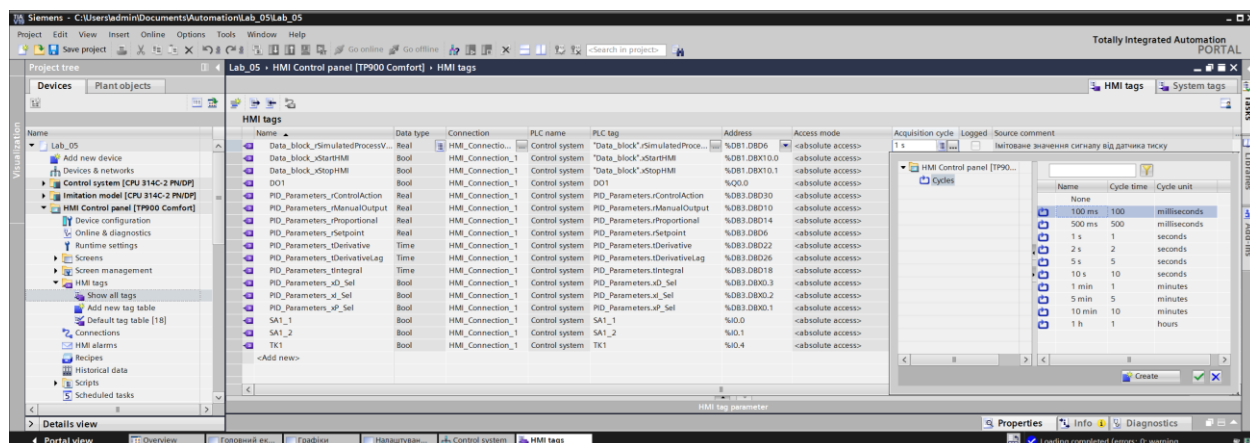


Рисунок 5.75 Налаштування періоду оновлення HMI тегів.

Налаштувати період оновлення HMI тегів можна у розділі **HMI tags** панелі оператора **HMI Control panel** дерева проєкту **TIA Portal**. Зміна цього значення для тегів, які виводяться на графіку, з **1 с** на **100 мс** суттєво покращує вигляд графіків.

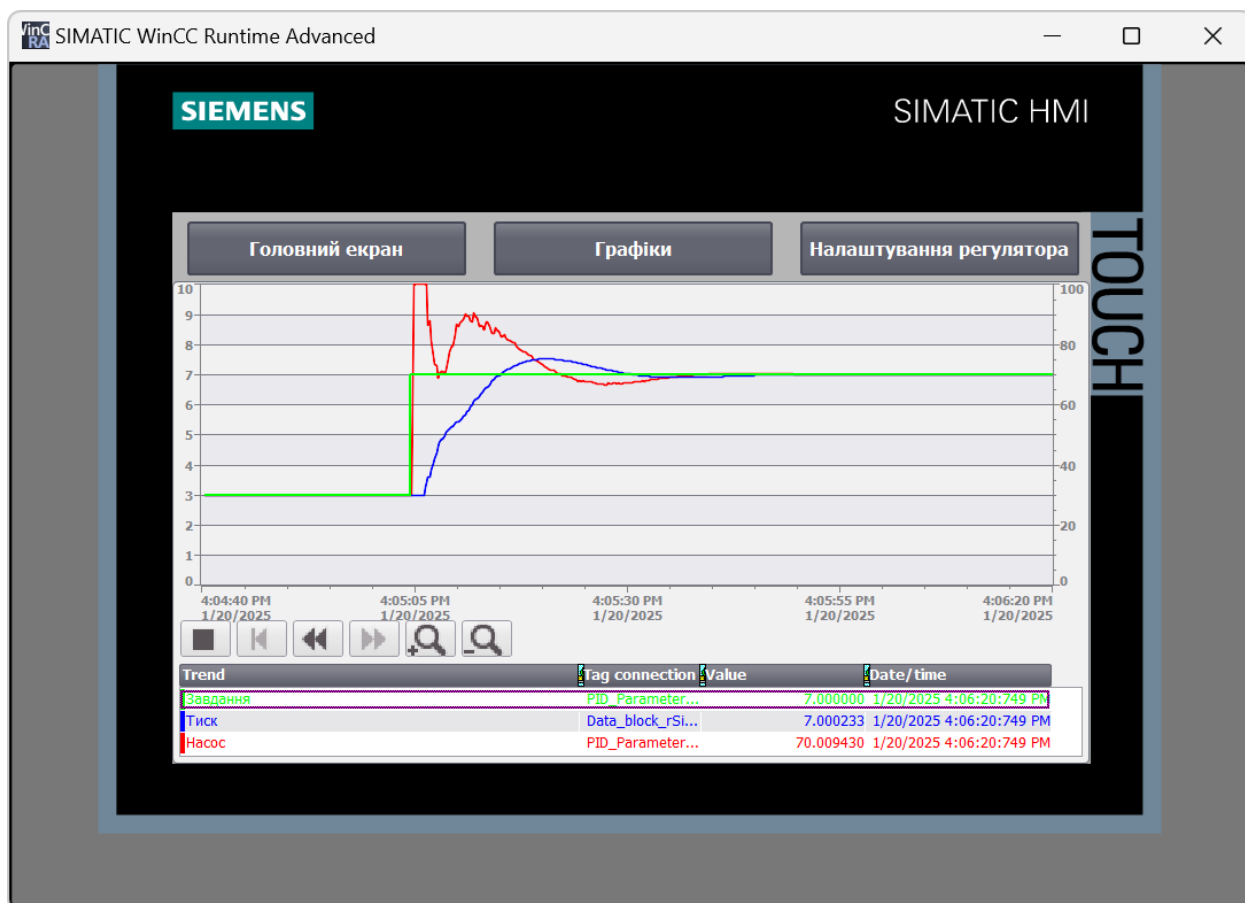


Рисунок 5.76 Графіки зі зменшеним циклом оновлення HMI тегів.

5.3 Контрольні завдання

1. Виконати перевірку роботи системи з імітаційною моделлю в різних режимах роботи.
2. Перевірити роботу системи при зміні заданої продуктивності насоса в ручному режимі.
3. Змінити параметри імітаційної моделі та визначити для неї параметри ПД-регулятора, за яких можна отримати мінімальний час регулювання тиску.
4. Створити на HMI Control Panel додатковий екран, за допомогою якого можна змінити налаштування параметрів імітаційної моделі, кнопку переходу на цей екран додати на головний екран.
5. Зробити у проєкті зміни, необхідні для його використання на реальній системі (без імітатора об'єкта).

Перелік рекомендованої літератури.

1. Bee, Liam. PLC and HMI Development with Siemens TIA Portal. Birmingham: Packt Publishing Limited, 2022. ISBN 978-1-80181-722-6.
2. Berger H. Automating with SIMATIC S7-300 inside TIA Portal: Configuring, Programming and Testing with STEP 7 Professional. 2nd edition. Erlangen: Publicis Publishing, 2014. 726 p. ISBN 978-3-89578-443-9.
3. Siemens AG. SIMATIC S7-300 Automation System, Hardware and Installation: CPU 312IFM – 318-2 DP. Installation Manual. Edition 06/2003. — Nuremberg: Siemens AG, 2003. — (Order No. 6ES7398-8FA10-8BA0, A5E00203919-01). — [Електронний ресурс]. — Режим доступу: https://cache.industry.siemens.com/dl/files/415/15390415/att_41918/v1/S7-300_IHB_e.pdf
4. SIMATIC TIA Portal STEP 7 Basic V10.5 Getting Started [Електронний ресурс] / Siemens AG. – Nuremberg: Siemens AG, 2009. – 197 с. – Режим доступу: https://cache.industry.siemens.com/dl/files/542/40263542/att_829827/v1/GS_STEP7Bas105enUS.pdf