

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ПОЛІТЕХНІКА»

**МЕТОДИЧНІ ВКАЗІВКИ
ДО ЛАБОРАТОРНИХ РОБІТ
з дисципліни
«Системи реального часу»
Робота із візуалізацією даних**

Одеса 2023

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ПОЛІТЕХНІКА»

МЕТОДИЧНІ ВКАЗІВКИ
ДО ЛАБОРАТОРНИХ РОБІТ
з дисципліни
«Системи реального часу»
Робота із візуалізацією даних

Затверджено
на засіданні кафедри комп'ютерних
інтелектуальних систем та мереж
Протокол № 7 від 23.03.2023

Одеса 2023

Методичні вказівки до лабораторних робіт з дисципліни «Системи реального часу» Робота із візуалізацією даних для здобувачів спеціальності 123 – "Комп'ютерна інженерія". /Укл.: О.Л. Шапоріна, В.О. Шапорін, І.П. Блінов, А.А. Нагорна – Одеса: Національний університет «Одеська політехніка», 2023. – 35с.

Укладачі:

Шапоріна О.Л., ст. викладач

Шапорін В.О., к.т.н., доцент

Блінов І.П., ст. викладач

Нагорна А.А., асистент

ЗМІСТ

Загальні положення	5
Мета лабораторних занять	5
Вимоги до виконання лабораторних робіт	5
Підготовка до лабораторних робіт	5
Оцінювання якості виконання лабораторних робіт	5
Звіт з виконання лабораторних робіт	5
Розширений план лабораторних занять	6
Лабораторний практикум	8
Робота 4. «Програмування ПЛК з використанням мов стандарту <i>IEC 61131-3</i> .»	8
Робота 5. «Робота з програмним забезпеченням для створення візуалізації інформації»	12
Робота 6. «Вивчення програмно-апаратної архітектури <i>CoDeSyS</i> .»	13
Методичні вказівки до виконання лабораторний робіт	15
Лабораторна робота №4.	15
Лабораторна робота №5	20
Лабораторна робота №6	27
Перелік джерел посилань	35

Загальні положення

Мета лабораторних занять – формування і розвиток спеціальних професійних дослідницько-інженерних компетентностей, які слугуватимуть студентам підґрунтям в їх подальшій професійній діяльності в галузі проектування автоматизованих систем реального часу, розподілених інформаційних систем, вбудованих систем тощо.

В ході лабораторних занять забезпечується розширення, поглиблення та деталізація знань, отриманих студентами на лекціях та в процесі самостійної роботи і спрямованих на забезпечення засвоєння навчального матеріалу на необхідному рівні, прищеплення умінь і навичок, розвиток дослідницько-інженерного мислення та професійного мовлення студентів.

Вимоги до виконання лабораторних робіт

Слухач може приступати до виконання лабораторної роботи у випадку успішно пройденого вхідного контролю. Вхідний контроль включає до себе наявність підготовленого звіту та виконання вимог з пункту підготовка до лабораторної роботи. У випадку відсутності звіту слухач вважається не допущеним до виконання лабораторної роботи та відсторонюється до моменту поки не пройде вхідний контроль.

Підготовка до лабораторних робіт

Ознайомитись з необхідною літературою по апаратному забезпеченню лабораторії. Ознайомитись з програмним забезпеченням. Виконати самостійні або індивідуальні завдання, вказані в завданні. Підготувати звіт до виконання лабораторної роботи.

Оцінювання якості виконання лабораторних робіт

Успішно виконаною вважається робота яку студент виконав вчасно, і де повністю розкрив розв'язання усіх поставлених завдань, та має можливість отримати додаткові бали до модульного контролю. В звіті повністю відображені кроки виконання роботи.

Зараховано вважається робота, яку студент виконав але допустив помилки або не вчасно виконав, або не повністю розв'язав поставлені задачі. Додаткові бали не передбачені.

Не зараховано вважається робота, де студент не виконав поставлені основні задачі.

За виконання додаткових завдань слухач має можливість отримати додаткові бали.

Звіт з виконання лабораторних робіт

Звіт з виконання лабораторної роботи включає в собі наступні пункти та повинен відображати повну інформацію про виконання лабораторної роботи:

- ФІО, курс, група, напрям підготовки.
- Завдання на лабораторну роботу: враховуючи вибір індивідуального варіанту.
- Підготовка до лабораторної роботи: визначення вхідних даних.
- Порядок виконання лабораторної роботи: визначення основних етапів виконання, вирішення поставлених задач, відображення процесу роботи.

- Результат успішного виконання роботи: відображення проведених тестів роботи.

-Висновки: пояснити одержаний результат та навички отримані в ході виконання лабораторної роботи.

Розширений план лабораторних занять

Назва та обсяг у годинах	Зміст лабораторної роботи	Мета заняття	Завдання
Робота 4. Програмування ПЛК з використанням мов стандарту IEC 61131-3. <i>Робота проводиться в лабораторії EduNet. Самостійна підготовка до лабораторної роботи вдома (6 год)</i> Формат лабораторної роботи: виконання індивідуальних завдань - робота в лабораторії (6 год) Поточний контроль (оцінка): виконання та захист індивідуального завдання.	1. Програмування ПЛК з використанням мов стандарту IEC 61131-3 2. Розробка додатків на мовах FBD, LD, IL, ST, SFC. 3. Виконання індивідуального завдання	Закріплення набутих навиків в програмуванні ПЛК на мовах стандарту IEC 61131-3. Закріплення отриманих навиків роботи з спеціалізованим ПЗ в програмуванні промислових ПЛК. Закріплення отриманих навиків в роботі з реальним та віртуальним обладнанням.	Виконання індивідуальних завдання. <i>При роботі в лабораторії зі стендами EduNet результат роботи відображається на індикаційній панелі.</i>
Робота 5. Робота з програмним забезпеченням для відображення інформації <i>Робота проводиться в лабораторії EduNet. Формат лабораторної роботи:</i>	1. Знайомство с ПЗ для відображення інформації з комплексу програм AUTOMATIONWORX Software Suite PhoenixContact™ WebVisit. 2. Принципи синхронізації програм комплексу AUTOMATIONWORX.	Набуття практичних навичок в візуалізації інформації за допомогою програмного забезпечення WebVisit. Отримання практичних навичок роботи з PCWorX та синхронізації програм з WebVisit.	1. Візуальне представлення даних, що є результатами роботи виконаних індивідуальних завдань. 2. Створення візуалізації для програм: – Відобразити графічний вид часової діаграми

майстер-клас. (1год) Виконання індивідуальних завдань (3год) Самостійна підготовка до лабораторної роботи (2 год) Поточний контроль (оцінка): виконання та захист індивідуального завдання.	3. Знайомство с відображенням програми <i>PCWorX</i> в <i>WebVisit</i>. Відображення результатів роботи програм. 4. Виконання індивідуальних завдань.	Навчитись відображати графічно інформацію від ПЛК	— Відобразити «сейфів» замок. — Програму для переміщення горизонтального крану. — Реалізувати автоматичне освітлення в приміщенні. — Реалізувати роботу світлофорів.
Робота 6. Вивчення програмно-апарат ної архітектури CoDeSyS. Робота проводиться в лабораторії <i>EduNet</i>. Формат лабораторної роботи: майстер-клас. (1год) Самостійна підготовка до лабораторної роботи (2 год) Виконання індивідуальних завдань (3год) Поточний контроль (оцінка): виконання та захист індивідуального завдання.	1. Знайомство з програмним комплексом <i>CoDeSyS</i>. 2. Конфігурування пристроїв та діагностування помилки. 3. Набуття практичних навичок в створенні та використанні компонент програмної організації, роботі з периферійними пристроями 3. Виконання індивідуальних завдань	Набуття навиків використання програмного комплексу <i>CoDeSyS</i> в вирішенні задач СРЧ. Отримання практичних навичок роботі в режимах симуляції та роботі з обладнанням: конфігурування, налаштування та завантаження програм. Набуття базових навиків в передачі цифрових та аналогових сигналів та відображенні їх на індикаційних панелях. Набуття навиків в програмуванні ПЛК на мовах стандарту <i>IEC 61131-3</i>. Набуття навиків при роботі з модулем візуалізації.	Застосувати набуті базові навички в роботі з ПЗ <i>CoDeSyS</i>: — Створити функціональний блок, який буде здійснювати індикацію чотирьох виходів. — Побудова часових діаграм. — Розробити «сейфів» замок. — Програму для переміщення горизонтального крану. — Реалізувати автоматичне освітлення в приміщенні. — Реалізувати роботу світлофорів.

Лабораторний практикум

Робота 4. «Програмування ПЛК з використанням мов стандарту *IEC 61131-3*.»

Робота проводиться в лабораторії EduNet.
Самостійна підготовка до лабораторної роботи вдома (3 год)
Формат лабораторної роботи: виконання індивідуальних завдань -
робота в лабораторії
Поточний контроль (оцінка): виконання та захист індивідуального
завдання.

Мета роботи: Набуття практичних навичок створення програм для ПЛК на основі мов стандарту IEC 61131-3: FBD (мова функціональних блоків), IL (мова інструкцій), мови LD (мова релейно-контактних схем), ST (структурований текст), SFC (послідовні функціональні діаграми). Отримання практичних навичок роботи з PC WORX. Закріплення практичних навичок в створенні та використанні компонент програмної організації, роботі з периферійними пристроями. Закріплення отриманих знань у формі виконанні індивідуальних завдань

Навички та вміння:

Набуття навиків в програмуванні ПЛК на мовах стандарту IEC 61131-3. Закріплення отриманих навиків роботи з спеціалізованим ПЗ в програмуванні промислових ПЛК. Закріплення отриманих навиків в роботі з реальним та віртуальним обладнанням.

Завдання роботи:

1. Запуск нового проекту. Налаштування та конфігурування пристроїв. Створення компонент програмної організації з вказанням властивостей блоків відповідаючи мові програмування.
2. Перевірка роботи програм на віртуальному ПЛК.
3. Виконання індивідуального завдання на реальному ПЛК з використанням індикаційної панелі.

Індивідуальне завдання:

1. Мови програмування FBD та ST.

А) Створити програму, що реалізує систему керування переміщенням горизонтального крану:

Три дискретних сигнали:

- Почати переміщення вліво;
- Почати переміщення вправо;
- Зупинити дію.

Два дискретних виходів для виконуючих механізмів:

- Переміщення крана вліво;
- Переміщення крана вправо.

Рухаючись в одному напрямі, кран не повинен реагувати на команду змінити напрям переміщення. Щоб змінити напрям переміщення, кран потрібно зупинити. Одночасна подача одиничних значень на виході не дозволяється. Час переміщення крана в кожному напрямі потрібно контролювати. Переміщення

вліво повинно зупинятись через 5с, а переміщення вправо – через 10 с. (Часові межі студент має обрати самостійно)

Б) Створити програму, що реалізує роботу «сейфового» замку. Використовувати потенціометр, як периферійний пристрій, для перевірки. Значення коду студент обирає самостійно, не менш 4 символів, та обов'язково значення мають змінюватись в обох напрямках. Обмежити введення спроб - 3 спроби. Після триразового неправильного введення даних - блокується можливість введення на 10 секунд.

В) Реалізувати потенціометр на мові *ST* і на *FBD*. Порівняти отримані програми.

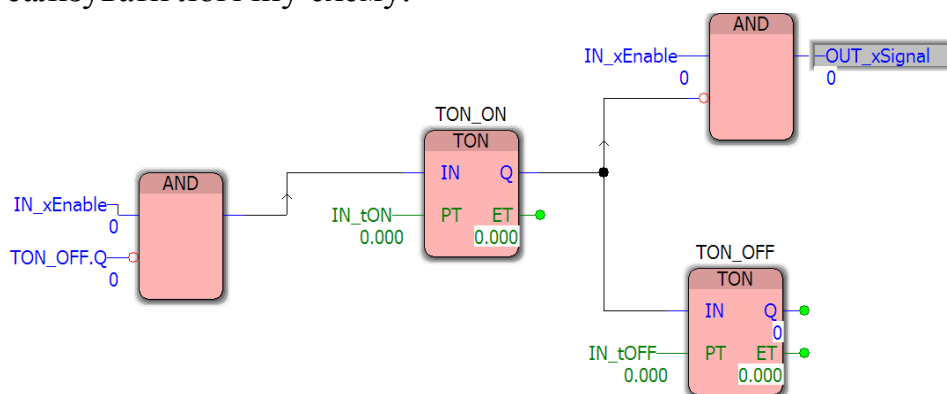
В реалізації даного завдання допоможуть такі ФБ: *SHR*, *AND*, *DIV*, *WORD_TO_REAL*, *WORD_TO_INT* в робочій області. Зміна властивостей блоків.

Г) Реалізувати на мові *ST* функцію «виключає АБО»

Д) Реалізувати програму для перетворення значень, що надходять з потенціометра в кольори. Виконувати перетворення значень 16# в кольори згідно стандарту.

2. Мова програмування IL.

Реалізувати логічну схему:



3. Мова програмування LD.

Реалізувати додаток для керування освітленням. Визначити часову затримку та автоматичне вимкнення, а також кілька режимів роботи. Керування освітленням виконати:

- В замкнутому просторі с розрахунком кількості присутніх в приміщенні. При відсутності людей світло повинно вимикатись через вибраний час.
- В довгому коридорі використовувати прохідний вимикач.

4. Мова програмування SFC

А) Реалізувати програму, що відобразить роботу світлофора. Для відображення використовувати індикаційну панель. Додати режим роботи «мигаючий жовтий – нічний режим»

Б) Відобразити роботу Т-триггеру.

Рекомендації до виконання роботи.

Для виконання лабораторної роботи необхідно ознайомитися з теоретичним матеріалом за темами "Програмний комплекс AUTOMATIONWORX Software Suite PhoenixContact™", "Апаратний комплекс ILC 130 StarterKit", "Стандарт IEC 61131", вивчити рекомендовану літературу []

Контрольні питання.

1. Дайте визначення мови *IL*.
2. Що таке мітка в мові *IL*?
3. Елементи командного рядка мови *IL*
4. Розмірність акумулятора в *IL*?
5. Чим визначається розмірність акумулятора в мові *IL*?
6. Обов'язкові елементи командного рядка мови *IL*.
7. Що таке модифікатори в мові *IL*? Що роблять модифікатори?
8. Для чого служать дужки в мові *IL*?
9. Перерахуйте стандартні оператори *IL*.
10. З операндами якого типу застосовуються оператори *S* і *R*?
11. Що являє собою програма на мові *IL*?
12. Як створити функцію на мові *IL*?
13. Як створити функціональний блок на мові *IL*
14. Дайте визначення мови *LD*.
15. Що таке котушка в мові *LD*?
16. Що таке контакт в мові *LD*?
17. Яким типом даних повинен бути фактичний параметр для контактів і котушок?
18. Для чого служать зв'язки? Які зв'язки розрізняють в мові *LD*?
19. Що є необхідною умовою виконання *FFB* в діаграмі *LD*?
20. Для чого служить елемент «Реле»?
21. Як додати *FBD* до *LD*-програми?
22. Що таке «*Network*» у *LD*-програмі?
23. Як створити функцію на основі *LD*-програми?
 24. Дайте визначення мови *ST*
 25. Що таке вирази в мові *ST*?
 26. Виклик функціональних блоків в *ST*.
 27. Яка операційна ієрархія в мові *ST*?
 28. Як працює оператор множинного вибору?
 29. Оператор вибору *IF*.
 30. Яка кількість повторень групи виразів забезпечує цикл *FOR*?
 31. Переривання ітерацій операторами *EXIT* і *RETURN*.
32. Дайте визначення мови *SFC*.
33. Що таке крок в мові *SFC*?
34. Що таке дія в мові *SFC*?
35. Види дій в мові *SFC*
36. Що таке перехід в мові *SFC*?
37. Використання функціональних блоків *SFC*.

38. Порівняйте мови *FBD*, *IL*, *LD*, *SFC*, *ST*.
39. Види розгалуження в мові *SFC*.

Робота 5. «Робота з програмним забезпеченням для відображення інформації»

Робота проводиться в лабораторії EduNet.

Самостійна підготовка до лабораторної роботи вдома (2 год)

Формат лабораторної роботи: майстер-клас (1 год), виконання індивідуальних завдань - робота в лабораторії

Поточний контроль (оцінка): виконання та захист індивідуального завдання.

Мета роботи: Ознайомитись з програмним забезпеченням для відображення інформації *WebVisit*. Отримати практичні навички роботи з *WebVisit*. Закріплення практичних навичок в створенні програм для промислових ПЛК та зв'язувати комплекс програм між собою. Навчитись візуалізувати графічно інформацію від ПЛК. Закріплення отриманих знань у формі виконанні індивідуальних завдань.

Навички та вміння:

Набуття практичних навичок в візуалізації інформації за допомогою програмного забезпечення *WebVisit*. Отримання практичних навичок роботи з *PCWorX* та синхронізації програм з *WebVisit*.

Завдання роботи:

1. Ознайомитись з програмним забезпеченням для відображення інформації *WebVisit*.
2. Створити простий додаток: відобразити приклад роботи логічної «І»
3. Виконати індивідуальні завдання.

Індивідуальне завдання:

Відобразити додатки із завданням з попередніх лабораторних робіт:

- Відобразити графічний вид часової діаграми
- Відобразити «сейфів» замок.
- Програму для переміщення горизонтального крану.
- Реалізувати автоматичне освітлення в приміщенні.
- Реалізувати роботу світлофорів.

Рекомендації до виконання роботи.

Для виконання лабораторної роботи необхідно ознайомитися з теоретичним матеріалом за темами "Програмний комплекс WebVisit ", "Стандарт IEC 61131", вивчити рекомендовану літературу []

Контрольні питання.

1. Інтерфейс середовища розробки WebVisit.
2. Можливості та особливості web-візуалізації.
3. Процедура розробки візуалізації.
4. Об'єкти та графічні примітиви web-візуалізації.
5. Параметри налаштування об'єктів та примітивів.
6. Відладка та виконання візуалізації.

Робота 6. «Вивчення програмно-апаратної архітектури CoDeSyS.»

Робота проводиться в лабораторії EduNet.

Самостійна підготовка до лабораторної роботи вдома (2 год)

Формат лабораторної роботи: майстер-клас (1 год), виконання індивідуальних завдань - робота в лабораторії

Поточний контроль (оцінка): виконання та захист індивідуального завдання.

Мета роботи: Ознайомитись з програмним комплексом *CoDeSyS*. Вивчити основні складові. Навчитись конфігурувати пристрої та діагностувати помилки. Отримати практичні навички роботи з *CoDeSyS*. Закріплення практичних навичок в створенні та використанні компонент програмної організації, роботі з периферійними пристроями. Закріплення отриманих знань у формі виконання індивідуальних завдань.

Навички та вміння:

Набуття базових навиків роботи зі спеціалізованим програмним забезпеченням програмування промислових логічних контролерів.

Набуття навиків використання програмного комплексу *CoDeSyS* в вирішенні задач СРЧ.

Отримання практичних навичок роботі в режимах симуляції та роботі з обладнанням: конфігурування, налаштування та завантаження програм.

Набуття базових навиків в передачі цифрових та аналогових сигналів та відображенні їх на індикаційних панелях.

Набуття навиків в програмуванні ПЛК на мовах стандарту *IEC 61131-3*.

Набуття навиків при роботі з модулем візуалізації.

Завдання роботи:

1. Запуск нового проекту. Налаштування та конфігурування пристроїв. Створення компонент програмної організації з вказанням властивостей блоків відповідаючи мові програмування.

2. Перевірка роботи програм на віртуальному ПЛК.

3. Виконання індивідуального завдання на реальному ПЛК з використанням індикаційної панелі.

Індивідуальне завдання:

Розробити додатки із завданням з попередніх лабораторних робіт в програмному комплексі *CoDeSyS*:

1. Створити функціональний блок, який буде здійснювати індикацію чотирьох виходів.

2. Побудова часових діаграм.

3. Розробити «сейфів» замок.

4. Програму для переміщення горизонтального крану.

5. Реалізувати автоматичне освітлення в приміщенні.

6. Реалізувати роботу світлофорів.

Додатки повинні супроводжуватись модулем візуалізації.

Рекомендації до виконання роботи.

Для виконання лабораторної роботи необхідно ознайомитися з теоретичним матеріалом за темами "Програмний комплекс CoDeSyS", "Стандарт IEC 61131", вивчити рекомендовану літературу []

Контрольні питання.

1. Які особливості середовища CoDeSys?
2. Що входить до базового складу комплексу CoDeSys?
3. Що містить проект CoDeSys?
4. Яка послідовність створення проекту в CoDeSys?
5. Характеристика мов програмування, які підтримує програмне забезпечення CoDeSys.

Методичні вказівки до виконання лабораторних робіт

Лабораторна робота №4

«Програмування ПЛК з використанням мов стандарту IEC 61131-3.»

Зміст лабораторної роботи

1. Програмування ПЛК з використанням мов стандарту IEC 61131-3:
розробка додатків на мовах *FBD, LD, IL, ST, SFC*.

Етапи розв'язання завдань

Розробка додатків на мовах FBD, LD, IL, ST, SFC

1. Потенціометр - FBD

Для реалізації даного завдання на мові FBD, необхідно використати такі функціональні блоки (рисунок 4.1):

- SHR - зсувний регістр (вправо);
- AND - «І»;
- WORD_TO_REAL - перетворення двійкового слова в дійсне число;
- DIV - ділення;
- REAL_TO_INT - перетворення з речового в ціле число.

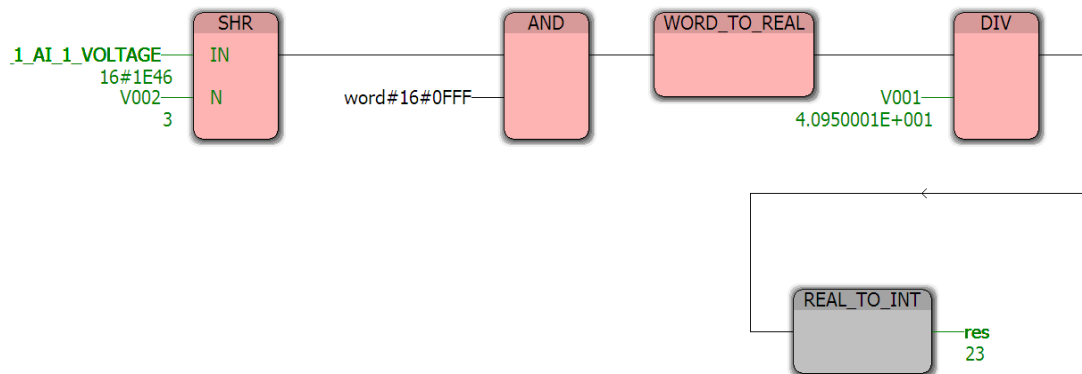


Рисунок 4.1 - Реалізація потенціометру на мові FBD

2. Потенціометр - ST

Для реалізації даного завдання на мові ST, необхідно використовувати такі оператори (рисунок 4.2): *REAL_TO_WORD, WORD_TO_REAL, SHR, &, /*.

volt := *REAL_TO_WORD(WORD_TO_REAL(SHR(IN, 3) & WORD#16#0FFF)/40.95);*

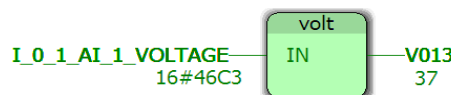


Рисунок 4.2 - Реалізація потенціометру на мові ST

3. XOR

Для початку необхідно оголосити змінні, принцип наведено на рисунку 4.3.

	Name	Type	Usage	Des
	Input Parameters			
	IN_x1	BOOL	VAR_INPUT	
	IN_x2	BOOL	VAR_INPUT	
	IN_x3	BOOL	VAR_INPUT	

Рисунок 4.3 - Оголошення змінних

Зовнішній вигляд програми наведено на рисунку 4.4.

```

1  FU_Xor3_ST := IN_x1 & Not IN_x2 & Not IN_x3
2              Or IN_x2 & Not IN_x3 & Not IN_x1
3              Or IN_x3 & Not IN_x1 & Not IN_x2;
```

Рисунок 4.4 - Вигляд програми

Далі необхідно відлагодити програму, приклад наведено на рисунку 4.5.

	Name	Online value	Type	Usage
	Default			
	IN_1	FALSE	BOOL	VAR_INPUT
	IN_2	FALSE	BOOL	VAR_INPUT
	IN_3	FALSE	BOOL	VAR_INPUT
	FU_XOR_ST	FALSE	BOOL	VAR_OUTPUT

Рисунок 4.5 - Відлагодження програми

Робочий блок FU_XOR_ST в розділі Main зображено на рисунку 4.6.

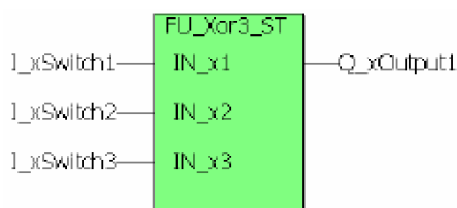


Рисунок 4.6 - Робочий блок FU_XOR_ST

3. Реалізувати логічну схему на мові IL, рисунок 4.7-4.8.

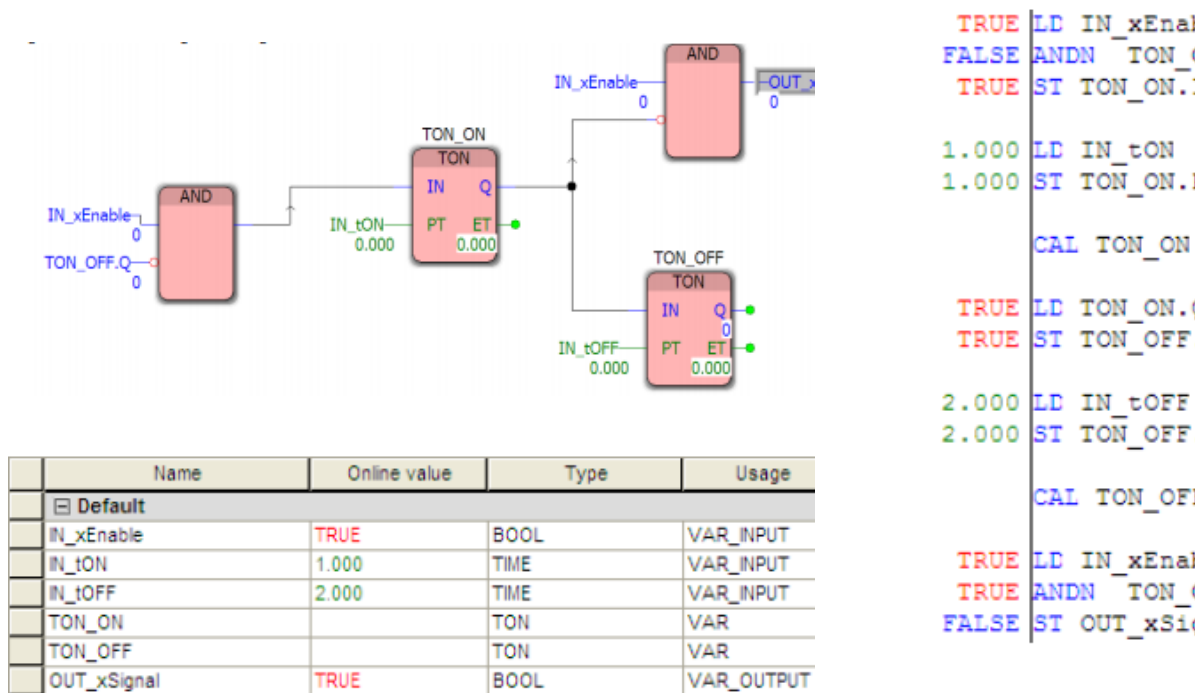


Рисунок 4.7 - Відлагодження та текст програми логічної схеми на мові IL

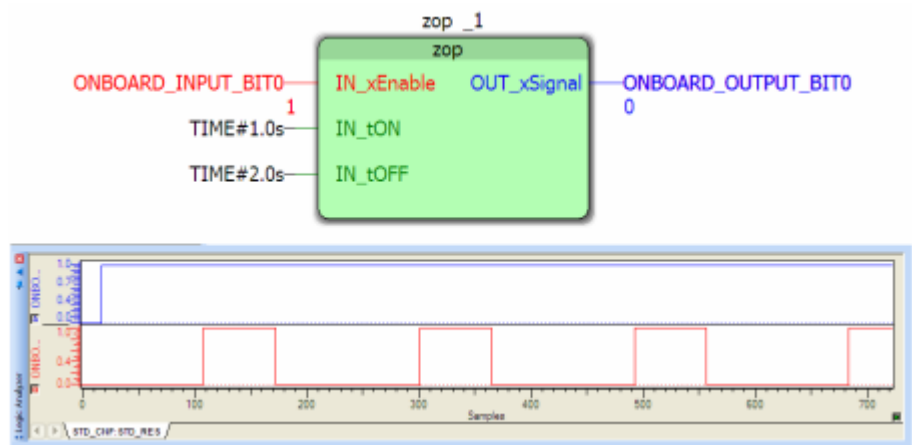


Рисунок 4.7 - Робочий блок в розділі Main та графік логічної схеми на мові IL

4. Створення програми, що реалізує простий принцип роботи світлофора. Початок роботи починається з заповнення таблиці змінних. Далі створюється функціональний блок користувача та в ньому створюється реалізація світлофора (рисунок 4.8).

Name	Type	Usage	Description
Input Parameters			
IN_tRed	TIME	VAR_INPUT	Red light time
IN_tRedYellow	TIME	VAR_INPUT	Red yellow light time
IN_tGreen	TIME	VAR_INPUT	Green light time
IN_tYellow	TIME	VAR_INPUT	Yellow light time
Output Parameters			
OUT_xRed	BOOL	VAR_OUTPUT	Red light
OUT_xYellow	BOOL	VAR_OUTPUT	Yellow light
OUT_xGreen	BOOL	VAR_OUTPUT	Green light
FB-Instances			
TON_Red	TON	VAR	FB for time control of red light
TON_RedYellow	TON	VAR	FB for time control of red yellow light
TON_Green	TON	VAR	FB for time control of green light
TON_Yellow	TON	VAR	FB for time control of yellow light

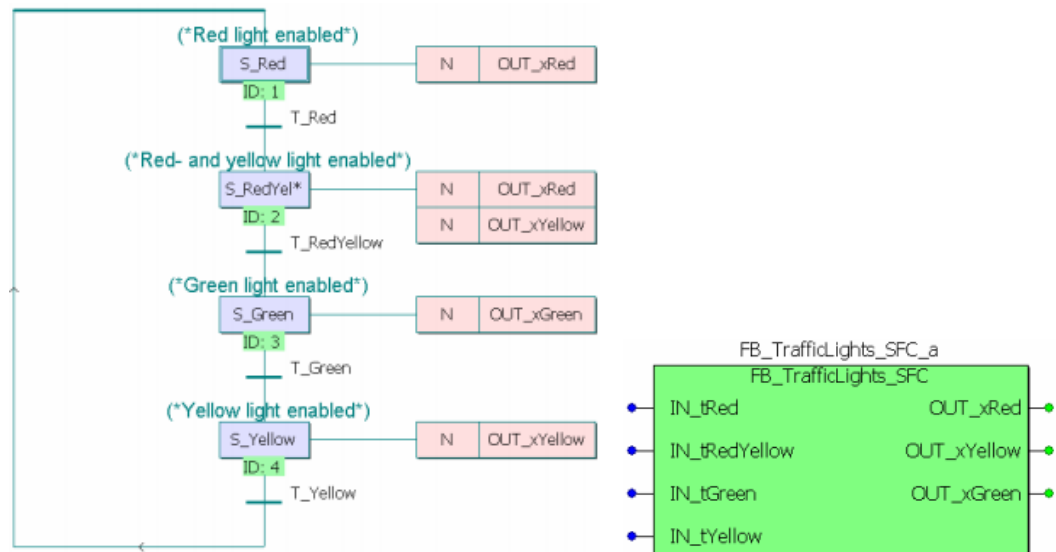


Рисунок 4.8 - Таблиця змінних та функціональний блок користувача

Переходи - це блок TON, де:

- Перший вхідний параметр – це ім'я кроку;
- Другий вхідний параметр – ім'я входу ФБ;
- Вихідний параметр – ім'я переходу.

Далі необхідно відлагодити програму світлофора, рисунок 4.9.

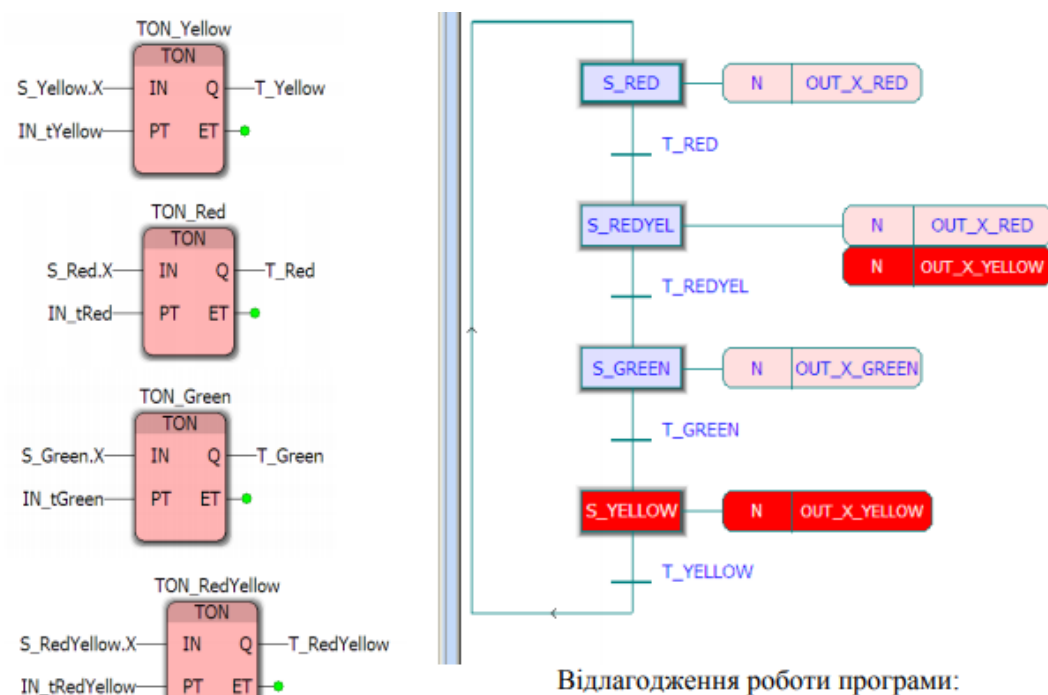


Рисунок 4.8 - Відлагодження роботи програми

5. Реалізувати роботу тригера. Необхідно заповнити таблицю змінних та створити функціональний блок тригера, рисунок 4.9.

	Name	Type	Usage	Descript
Input Parameters				
	IN_xTrigger	BOOL	VAR_INPUT	
Output Parameters				
	OUT_xSignal	BOOL	VAR_OUTPUT	

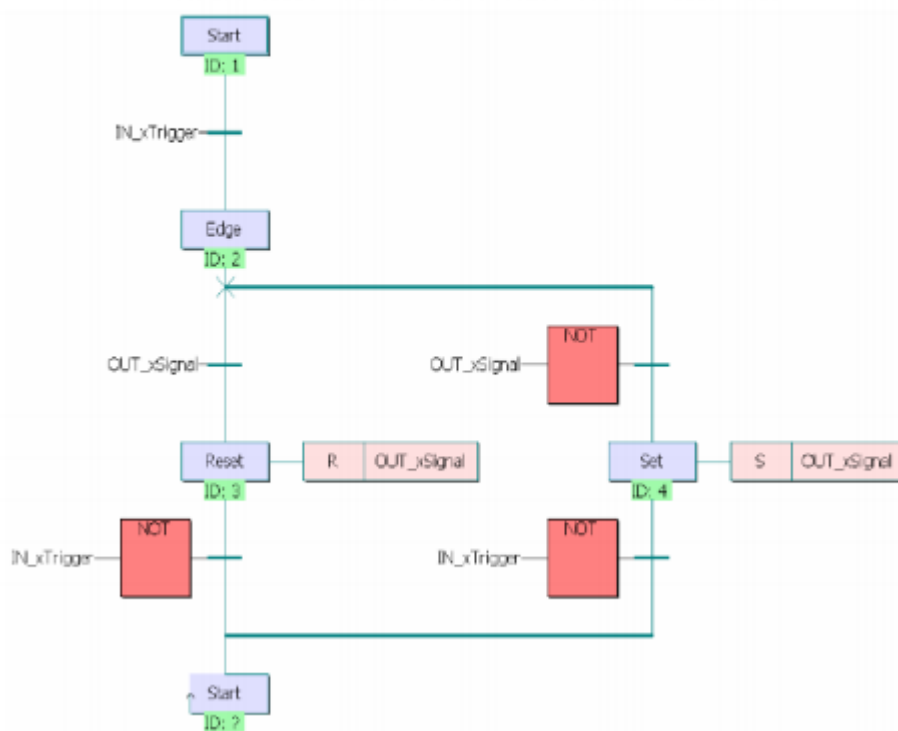


Рисунок 4.9 - Таблиця змінних та функціональний блок тригера

Лабораторна робота №5

«Робота з програмним забезпеченням для створення візуалізації інформації»

Зміст лабораторної роботи

1. Ознайомитись з програмним забезпеченням для відображення інформації *WebVisit*.
2. Створити простий додаток: відобразити приклад роботи логічної «І»
3. Виконати індивідуальні завдання.

Етапи розв'язання завдань

Створення проекту та огляд інтерфейсу користувача

Для того щоб запустити програму *WebVisit* необхідно перейти через Меню *Пуск* по вказаному на рисунку 1.1 шляху, або знайти програму через пошук *Windows*.

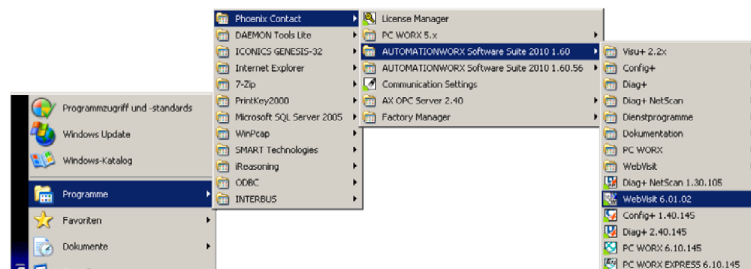


Рисунок 1.1 - Шлях запуску програми *WebVisit*

WebVisit відобразить користувацький інтерфейс, якщо програма була запущена в перший раз, рисунок 1.2. У разі, якщо який-небудь проект був запущений через *WebVisit*, програма автоматично завантажить останній проект.

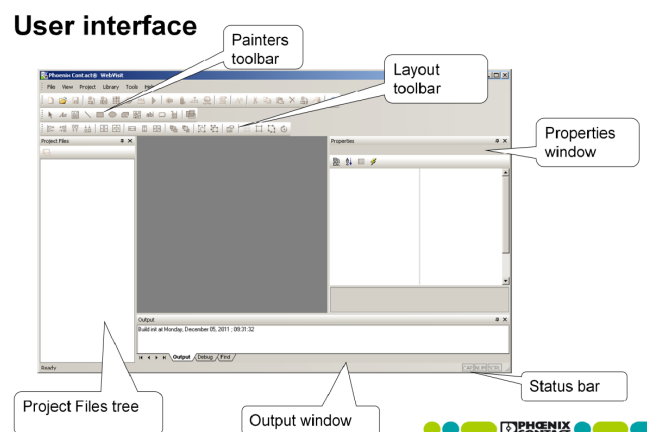


Рисунок 1.2 - Користувацький інтерфейс

Перш за все, перед тим як почати роботу з *WebVisit*, необхідно мати готовий проект у програмі *PC Worx*. Для прикладу, в програмі *PC Worx* була реалізована робота логічного елементу «I» з двома змінними, рисунок 1.3.

Усі змінні, які будуть прописані, повинні бути глобальними! Як наведено в графі «Usage» рисунок 1.4.

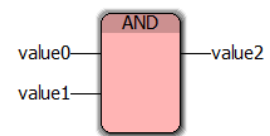


Рисунок 1.3 -
Логічний елемент
«I»

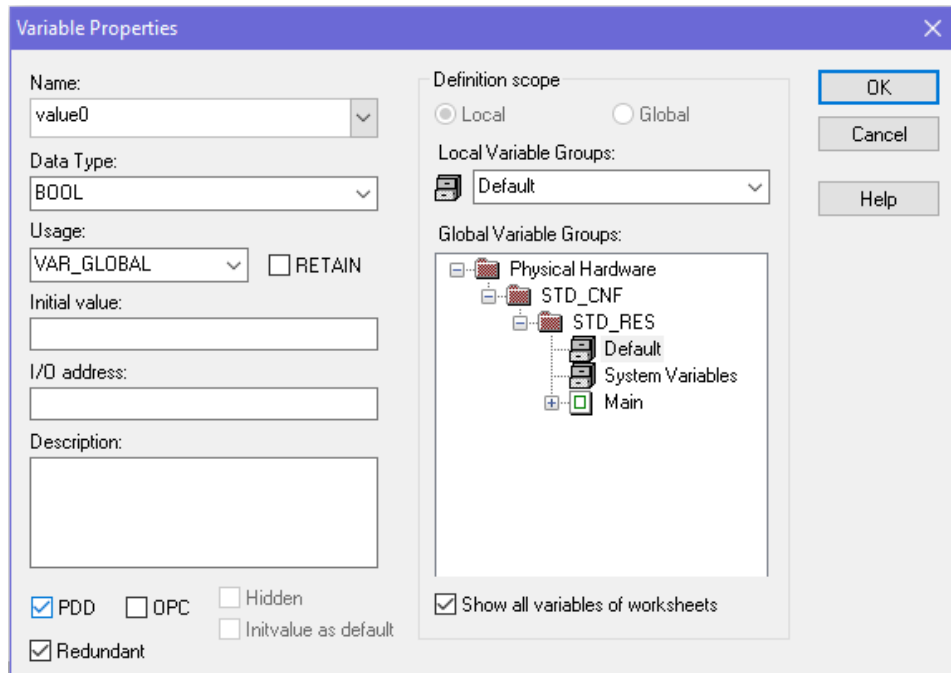


Рисунок 1.4 - Вікно змінних властивостей

УВАГА! Усі змінні повинні бути помічені як *PDD (Process Data Directory)* у таблиці змінних в *PC Worx* проекті, рисунок 1.5. Це необхідно для того, щоб проект після побудови створив *excel*-таблицю, яку в подальшому необхідно буде експортувати у конфігурації *WebVisit* проекту.

Name	Typ	Verwendung	Beschreibung	Adresse	Anfangswert	Remanent	PDD	OPC
WebVisitVariablen								
Mixer	ST_Mixer	VAR_GLOBAL	Als Strukturvariable. Wird in der POE Main über die Variable ON...					
ON_OFF_Button_Mixer	BOOL	VAR_GLOBAL	Schaltet den Schaltzustand der Strukturvariable aus der Visu					
Ventilatordre	INT	VAR_GLOBAL	Simuliert den drehenden Ventilator					
Gasengine	BOOL	VAR_GLOBAL	Wird über Debug simuliert					
Gasthroughput	INT	VAR_GLOBAL	Wird über Debug simuliert. Hier z.B. von 0 bis 90 m³					
Heating	BOOL	VAR_GLOBAL	Schaltet die Heizung aus der Visu					
Programminfo	AR_0_2	VAR_GLOBAL	Für das Mairo Dropdownlist					
Ventilator	BOOL	VAR_GLOBAL	Schaltet den Ventilator aus der Visu					
Sawtooth	DINT	VAR_GLOBAL	Ein Sägezahn von 0 bis 30000					
Fil_level	WORD	VAR_GLOBAL	Das Poti als externe Füllveisimulation	%MW2				
System Variables								
PLC_MODE_ON	BOOL	VAR_GLOBAL	PLC status ON	%MX1.0.0				
PLC_MODE_RUN	BOOL	VAR_GLOBAL	PLC status RUN	%MX1.0.1				
PLC_MODE_STOP	BOOL	VAR_GLOBAL	PLC status STOP	%MX1.0.2				
PLC_MODE_HALT	BOOL	VAR_GLOBAL	PLC status HALT	%MX1.0.3				
PLCDEBUG_BPSET	BOOL	VAR_GLOBAL	Breakpoint set	%MX1.1.4				
PLCDEBUG_FORCE	BOOL	VAR_GLOBAL	Variable(s) forced	%MX1.2.0				
PLCDEBUG_POWERFL...	BOOL	VAR_GLOBAL	Powerflow ON	%MX1.2.3				
PLC_TICKS_PER_SEC	INT	VAR_GLOBAL	System ticks per second	%MW1.44				
PLC_SYS_TICK_CNT	DINT	VAR_GLOBAL	Number of system ticks	%MD1.52				

Рисунок 1.5 - Таблиця змінних

Для створення нового проекту необхідно у вкладці *File* вибрати *New Project*, рисунок 1.6. Після цього з'явиться вікно налаштування нового проекту в якому необхідно вказати, що веб-сервер буде інтегрований у контролер PLC.

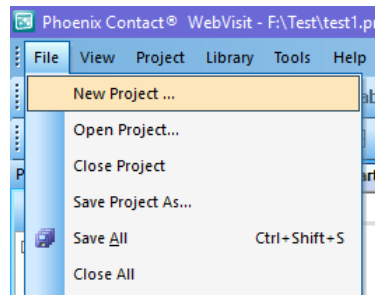


Рисунок 1.6 - Вкладка створення нового проекту

Далі вказуємо ім'я проекту та шлях, де буде збережено проект *WebVisit*, рисунок 1.7.

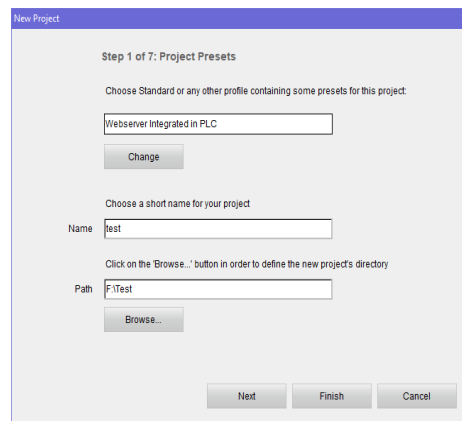


Рисунок 1.7 - Вікно створення нового проекту

В результаті, у проекті буде створено файл з розширенням «*.teq*» в який можна перейти через подвійний клік, після чого буде відображене робоче поле для розташування елементів проекту, рисунок 1.8.

У робоче поле можна помістити з лінійки елементів необхідний елемент, натиснувши на нього та повторно натиснувши на робоче поле. Це створить вибраний елемент на полі в одному екземплярі, рисунок 1.9.

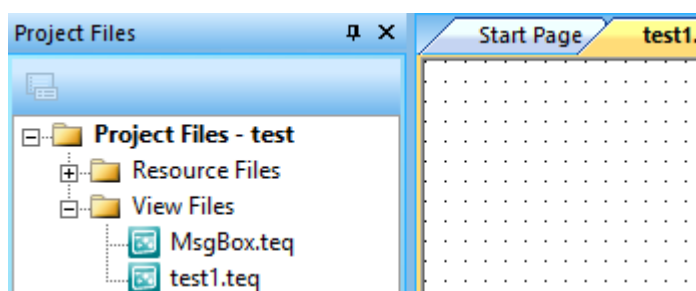


Рисунок 1.8 - Робоче поле

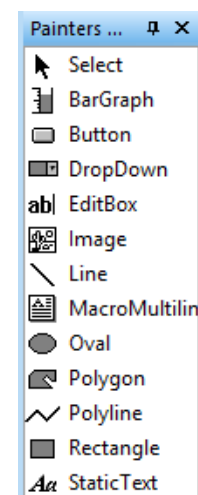


Рисунок 1.9 - Лінійка елементів

Організуємо візуалізацію роботи логічного елемента «І» з двома змінними, де вхідні змінні будуть кнопками, а для відображення змінної результату буде використано елемент овал (заміна елемента лампочки), рисунок 1.10.

Після того як будуть розташовані усі елементи для візуалізації, необхідно завантажити файл зі змінними (*PPO variable list*) із програми PC Worx. Даний файл має назву PDD(excel-файл) та знаходиться у папці проекту PC Worx —
“project_name”\C\STD_CNF\R\STD_RES\PDD.

Щоб завантажити файл зі змінними необхідно перейти у вкладці *Project* перейти у *Project Configurations*, там необхідно перейти у розширене налаштування (*Project Advanced*), рисунок 1.11.

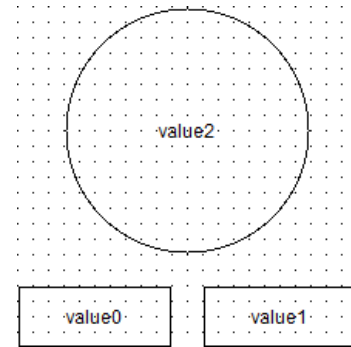


Рисунок 1.10 -
Логічний елемент «І»
з двома змінними

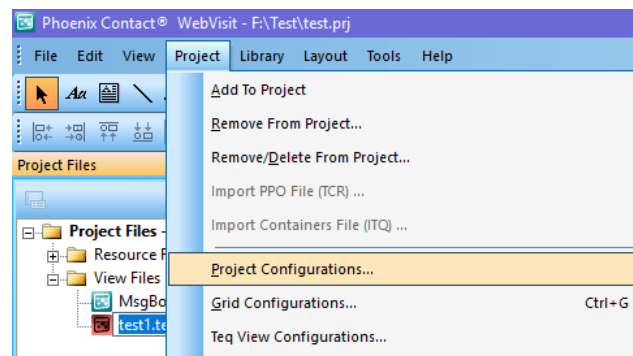
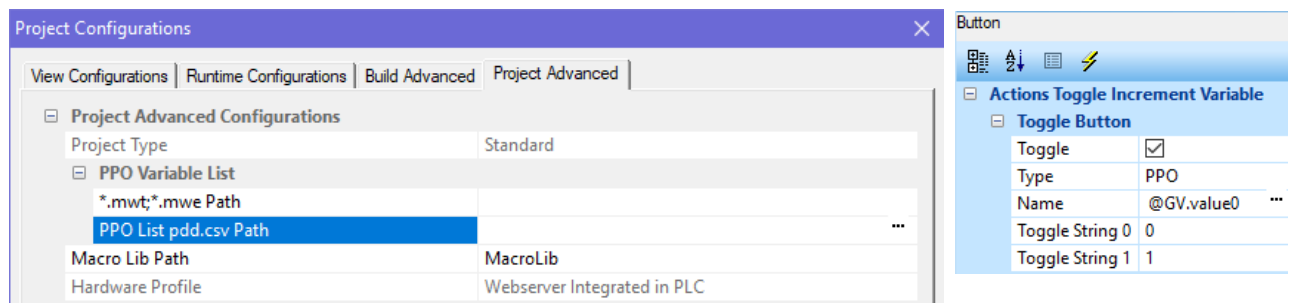


Рисунок 1.11 - Вкладка *Project*

У розширених конфігураціях необхідно вказати шлях до *PDD* файлу, рисунок 1.12.



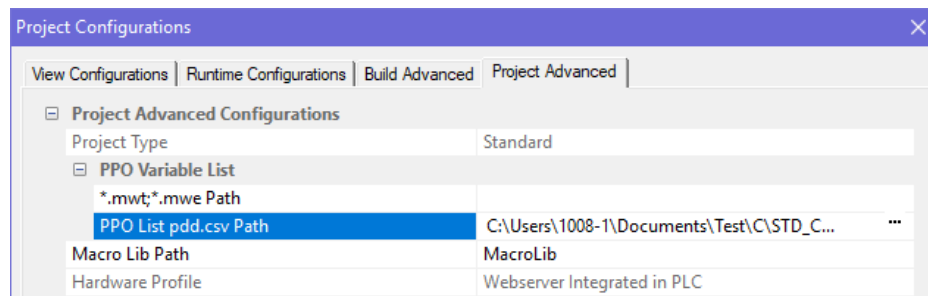
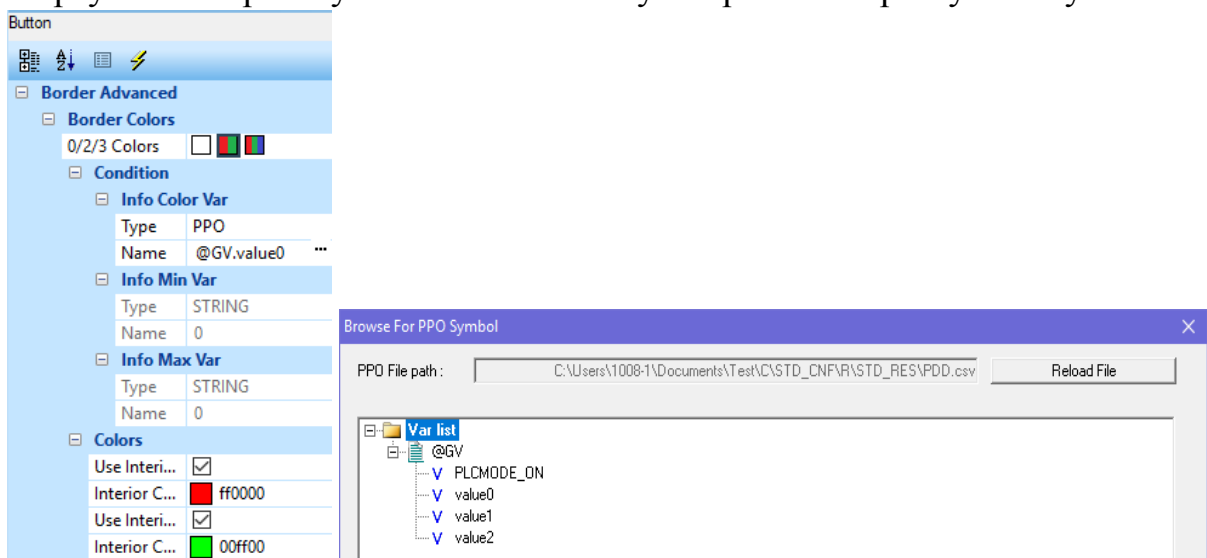


Рисунок 1.12 - Вікна розширених конфігурацій

Для того, щоб інтегрувати змінні у проект *WebVisit*, необхідно вибрати потрібний елемент інтерфейсу, у властивостях елемента у полі *Name* натиснути на "...", рисунок 1.13. Це відобразить список з усіма змінними які ви експортували із проекту *PC Worx* і з списку обираємо потрібну змінну.

Рисунок 1.13 - Інтегрування змінних у проект *WebVisit*

Далі необхідно налаштувати інтеракцію з кнопкою. Якщо ми натиснемо на кнопку, вона повинна змінити свій колір з червоного на зелений і навпаки. Для цього у вкладці *Border Colors* вказуємо, що кнопка буде мати два кольори (0/2/3). Після цього у нас з'являється можливість налаштувати відповідні кольори. Так само налаштовуємо усі інші елементи інтерфейсу.

Як тільки усі елементи інтерфейсу будуть налаштовані, необхідно побудувати проект (Project/Build All), рисунок 1.14. Як тільки проект буде побудований, необхідно завантажити його.

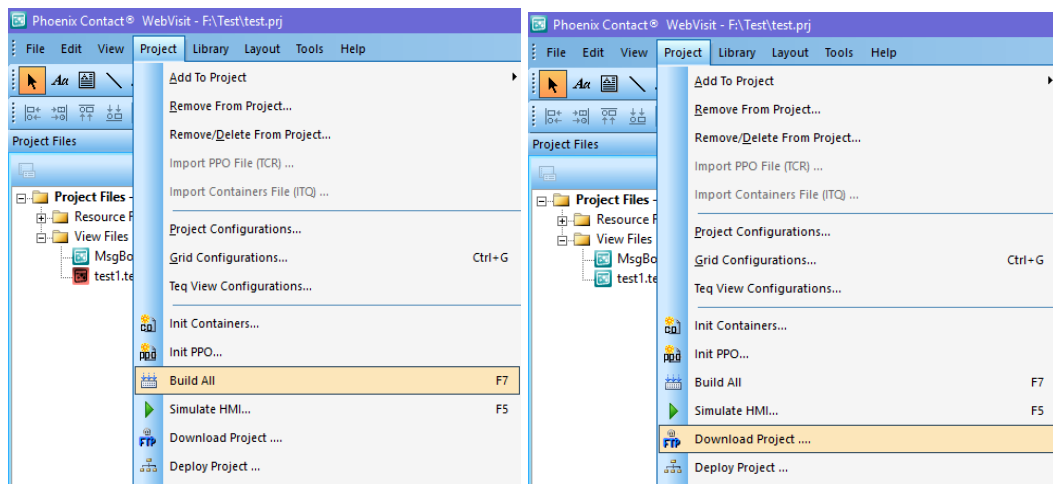


Рисунок 1.14 - Вкладка Project, побудова та завантаження проекту

Для завантаження проекту вказуємо адресу контролера та натискаємо *Connect*, рисунок 1.15.

Як тільки *WebVisit* проект буде завантажений необхідно дозволити запуск проекту у браузері *Internet Explorer*.

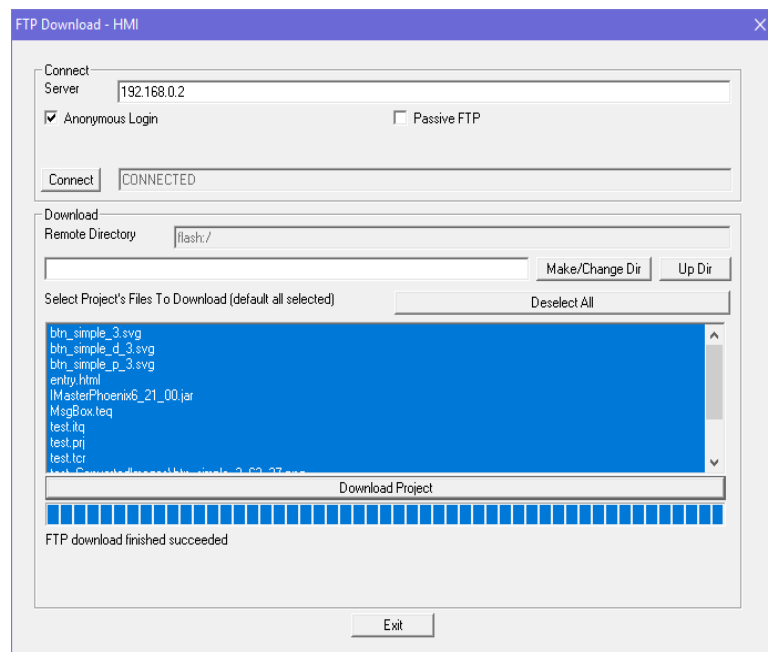
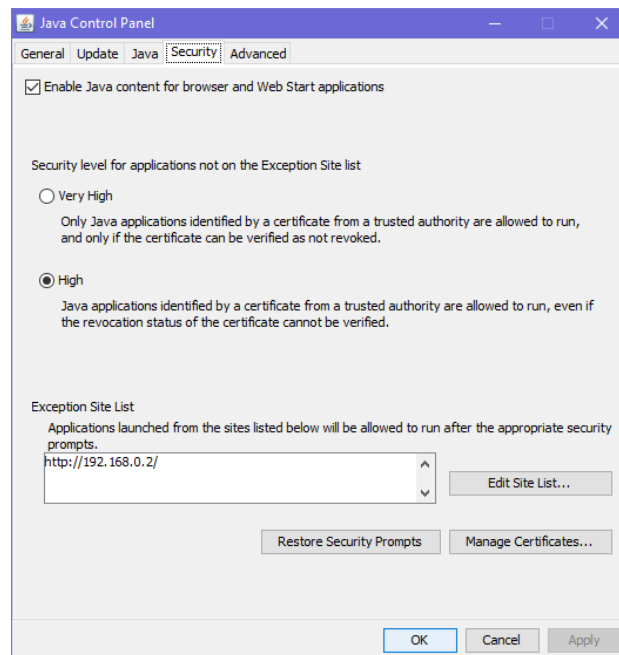


Рисунок 1.15 - Вікно завантаження проекту

УВАГА! Проект запускається тільки через браузер *Internet Explorer*.

Необхідно зайти в *Java Configurations* і у вкладці *Security* додаємо у лист виключень адресу нашого контролера, рисунок 1.16.

Рисунок 1.16 - Вікно *Java Configurations*

Як тільки адреса буде прописана у листі виключень можна запускати проект. Для цього у пошуковий рядок Internet Explorer прописуємо адресу контролера, рисунок 1.17.

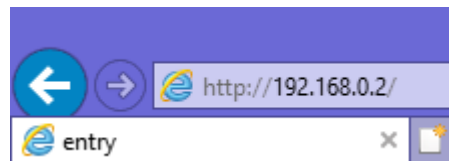


Рисунок 1.17 - Пошуковий рядок Internet Explorer

У результаті, ми отримуємо візуалізацію роботи логічного елементу «I» з двома змінними, рисунок 1.18.

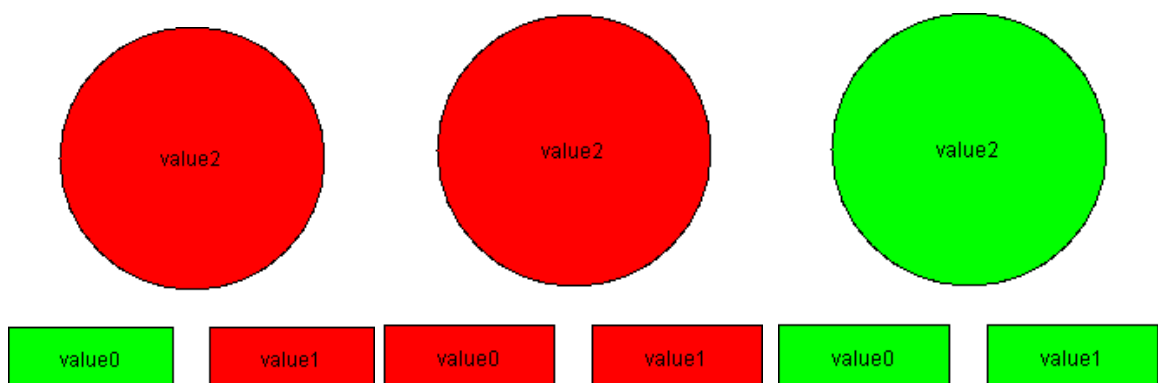


Рисунок 1.18 - Візуалізація роботи логічного елементу «I» з двома змінними

Лабораторної роботи №6 «Вивчення програмно-апаратної архітектури CoDeSyS.»

Зміст лабораторної роботи

1. Запуск нового проекту. Налаштування та конфігурування пристроїв. Створення компонент програмної організації з вказанням властивостей блоків відповідаючи мові програмування.
2. Перевірка роботи програм на віртуальному ПЛК.
3. Створення модуля візуалізації та підключення до змінних програми.
4. Виконання індивідуального завдання.

Етапи розв'язання завдань

Запуск нового проекту. *File -> New Project.* (Рисунок 2.1). Обрати *Standart project* (Рисунок 2.2).

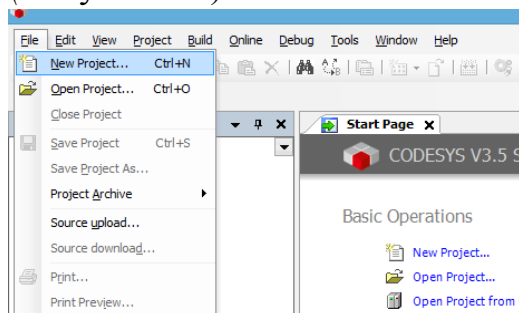


Рисунок 2.1 - Вкладка *File*

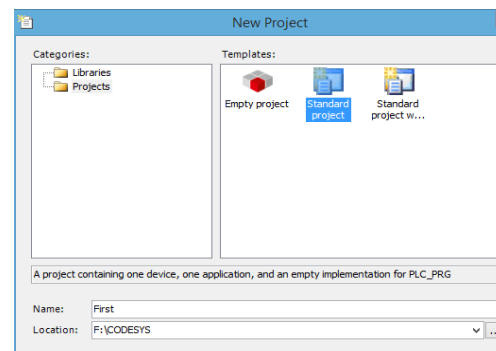


Рисунок 2.2 - Вікно запуску нового проекту

Обрати мову програмування та пристрій, де завантажується додаток. Наприклад, мова програмування *FBD*. Пристрій не має значення (рисунок 2. 3).

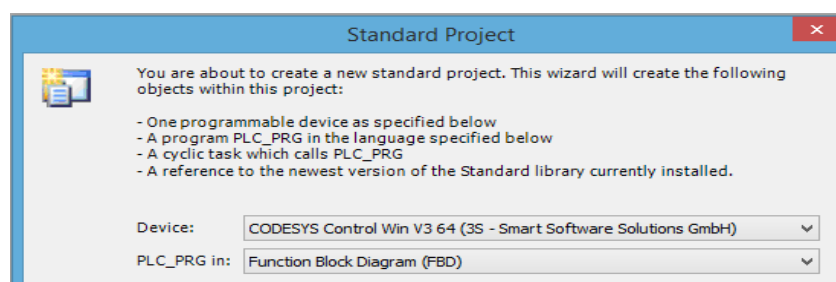


Рисунок 2.3 - Вікно налаштувань нового проекту

Зліва знаходиться *Вікно пристроїв*, де відображено дерево проекту. Обрати *PLC_PRG(PRG)* (рисунок 2. 4)

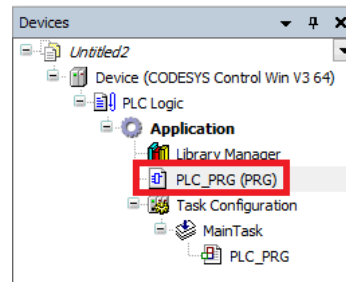


Рисунок 2.4 – Дерево проекту

У правій частині екрана розташовується панель інструментів, яка включає (рисунок 2.5): елементи для побудови функціональної схеми програми (привласнення змінних виходів логічних елементів, розгалуження, точки переходу), логічні елементи (І, АБО, виключає Або), математичні оператори (додавання, віднімання, множення, ділення; перевірки на рівність, більше, менше), інші оператори (селектор, мультиплексор і ін), функціональні блоки (тригера, затримка включення і виключення, лічильники). У разі виникнення питань про функціональне призначення оператора, необхідно зайти в розділ *Help-> Index* і в пошуковому рядку вбити назву оператора.

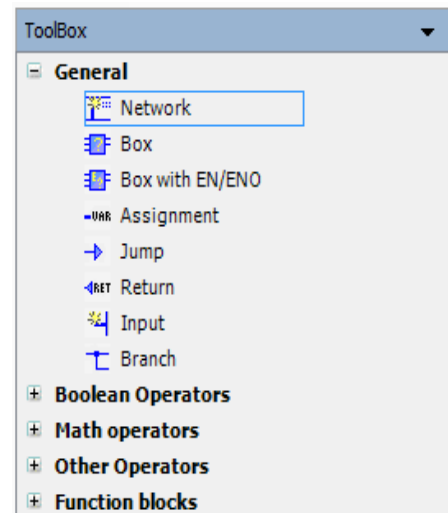


Рисунок 2.5 - Панель інструментів

Файл з назвою PLC_PRG (PRG) містить 2 поля: ВЕРХНЄ - таблиця змінних (де містяться всі змінні, які є в проекті) і НИЖНЄ - вікно програмування (де формується структура програми). Приклад зображений на рисунку 2.6.

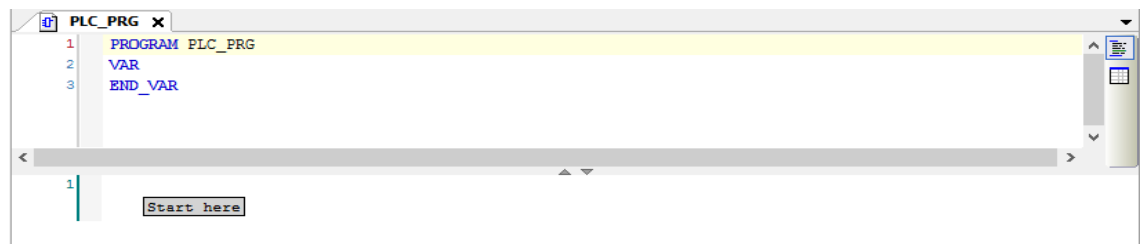


Рисунок 2.6 - Файл PLC_PRG (PRG)

Створимо логічний елемент І на робочому полі. За допомогою *drag&drop* перемістити елемент в точку *Start here*. Коли вона стане зеленою, то елемент можна відпускати, рисунок 2.7.

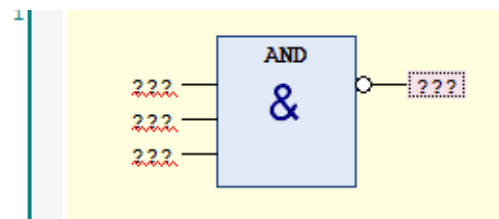


Рисунок 2.7 - Логічний елемент "І"

Для додавання входів до елементу: натиснути по елементу правою кнопкою миші -> *Append input*.

Для інверсії входу/виходу: натиснути правою кнопкою миші на підставах *ВХОДА* / *ВИХОДУ* і вибрати пункт *Negation*, рисунок 2.8.

Для присвоєння виходу логічного елементу з пункту *General* вибрати і перетягнути елемент *Assignment*, далі перетягнути його на вихід елемента.

Натиснувши по знаку «?» створити змінні для елемента і натиснути *Enter*. У вікні можна вибрати тип змінної (наприклад: *BOOL*), змінити ім'я і призначення (*VAR INPUT*, *VAR OUTPUT*, *VAR*), задати початкове значення (*Initialization*). Після присвоєння змінних у верхньому полі програми з'явилися їх оголошення: всі змінні *BOOL*, тип *VAR*; змінна *third* ініціалізована початковим значенням *TRUE*. (Рис.2.9)

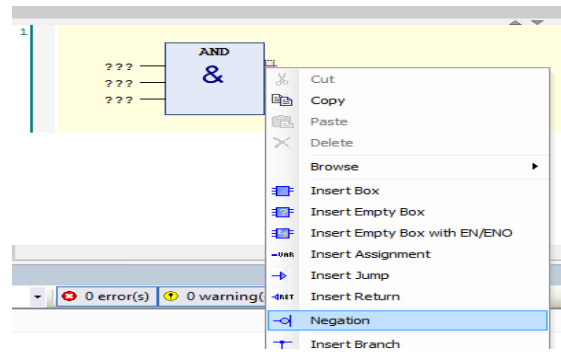


Рисунок 2.8 - Пункт *Negation*

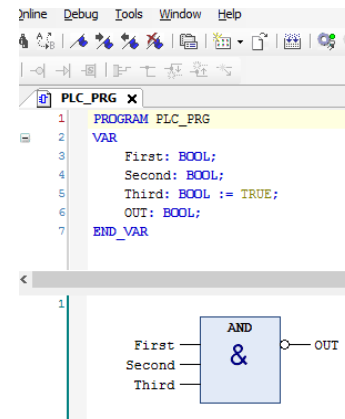
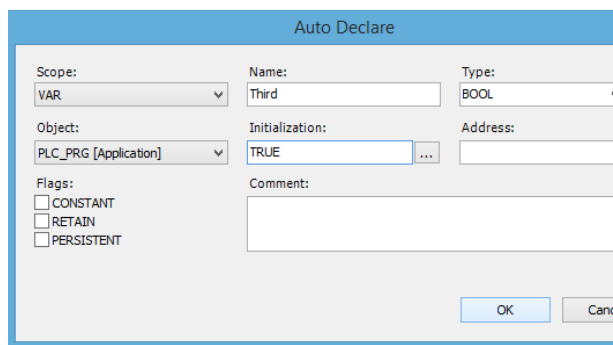


Рисунок 2.9 - Присвоєння змінних логічного елементу “І”

Для запуску додатку в режимі симуляції, натиснути *F11(Build)* та обрати *Online->Simulation*. Обрати *Online->Login -> Yes*, рисунок 2.10.

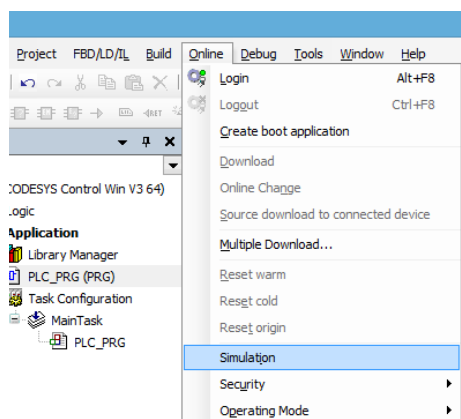


Рисунок 2.10 - Запуск додатку в режимі симуляції

Запуск додатку - *F5*. В полі *Watch1* можна обрати змінні та значення (рисунок 2.11).

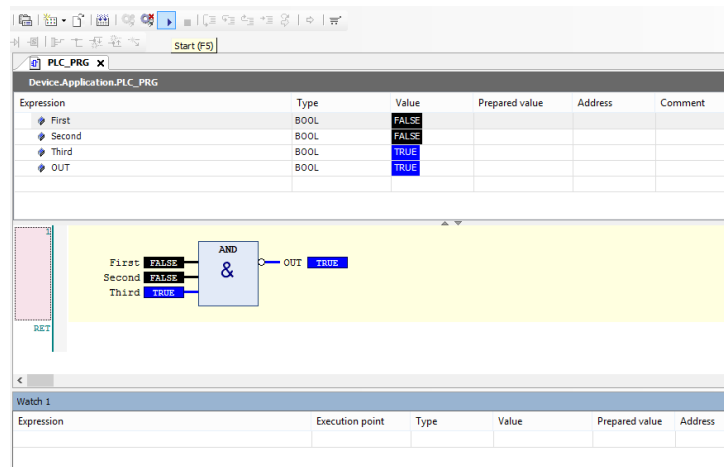


Рисунок 2.11 - Вікно змінних та значень

Для цього обрати змінні які знаходяться в *Device->Application->PLC_PRG*. Натиснути правою кнопкою миші по будь-якому полю в *Watch1* та обрати *Write values*. Приклад роботи програми в режимі симуляції зображений на рисунку 2.12

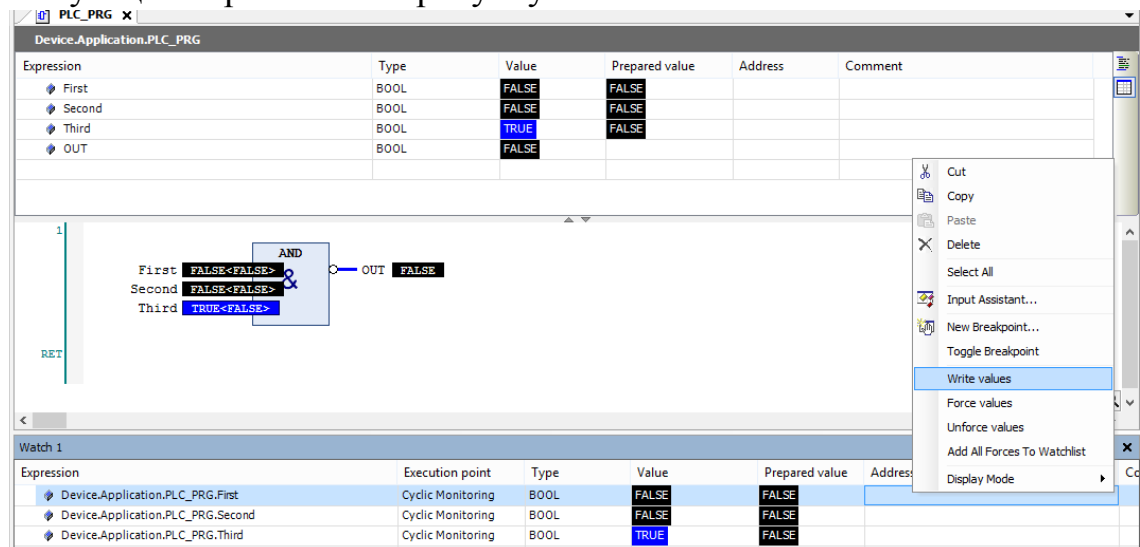


Рисунок 2.12 - Приклад роботи програми в режимі симуляції

Для графічного відображення інформації використовується візуалізація програми, за допомогою додавання модулю візуалізації до існуючого проекту: натиснути правою кнопкою миші по вкладці *Application* □ *Add object* □ *Visualization* □ *Add*. В правій частині з'явиться панель з елементами графічної візуалізації, рисунок 2.13.

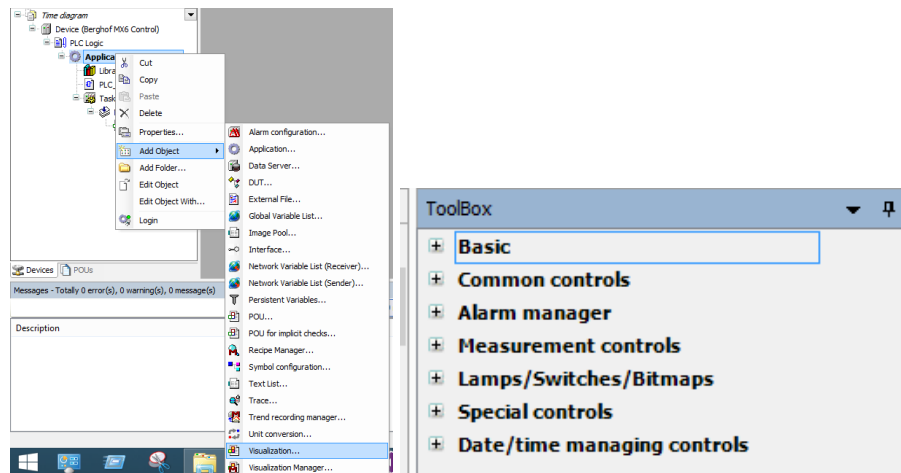


Рисунок 2.13 - Відкриття панелі з елементами графічної візуалізації

- *Basic*: базові фігури (прямокутник, еліпс, крива Безьє, лінії, полігони), зображення;
- *Common controls*: елементи управління (combobox, spin control, slider, text field, scroll bar, label and others)
- *Alarm manager*
- *Measurement controls*: елементи, що показують результати вимірювань (циферблати різних типів, гістограма, термометр). А також потенціометр - завдання певного діапазону.
- *Lamps / Switches / Bitmaps* - лампочки, вимикачі.
- *Special controls* - інструменти для створення «своїх» елементів управління.
- *Date / time managing controls* - віджети дата і час.

Приклад реалізації проекту. Завдання: Індикація світлодіодів.

Створити додаток, який буде відображати індикацію світлодіодів. Для того, щоб додати в проект функціональний блок, який буде подавати імпульси певної частоти, необхідно підключити бібліотеку *Util*. Зайти в менеджер бібліотек ☐ *Add library* ☐ *Application* ☐ *Common* ☐ *Util* ☐ *OK*, рисунок 2.14.

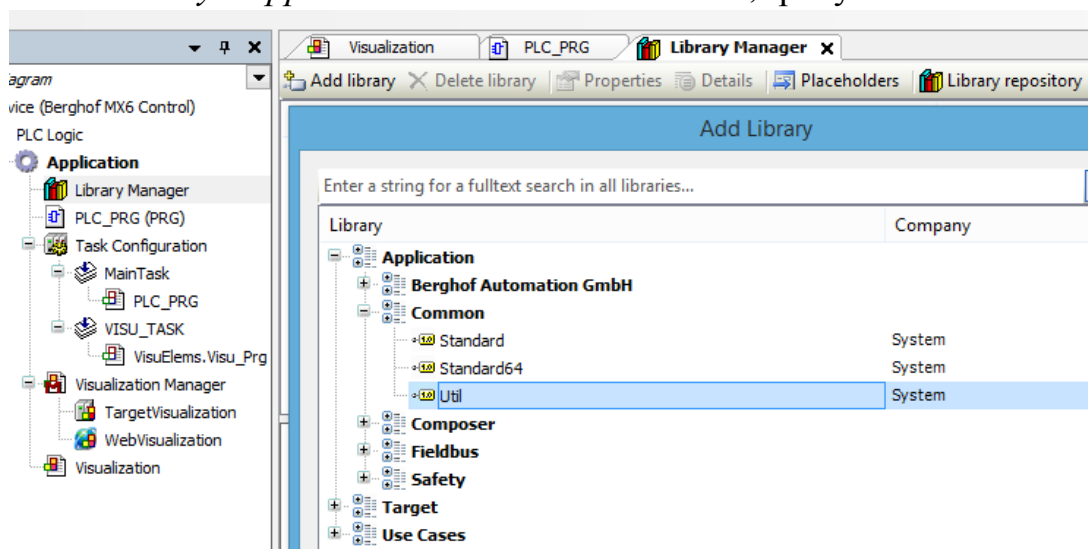


Рисунок 2.14 - Підключення бібліотеки *Util*

В файлі *PLC_PRG(PRG)* натиснути правою кнопкою миші по «рожевій» полосці *Insert Box* *Util* *Blink* *OK*, рисунок 2.15.

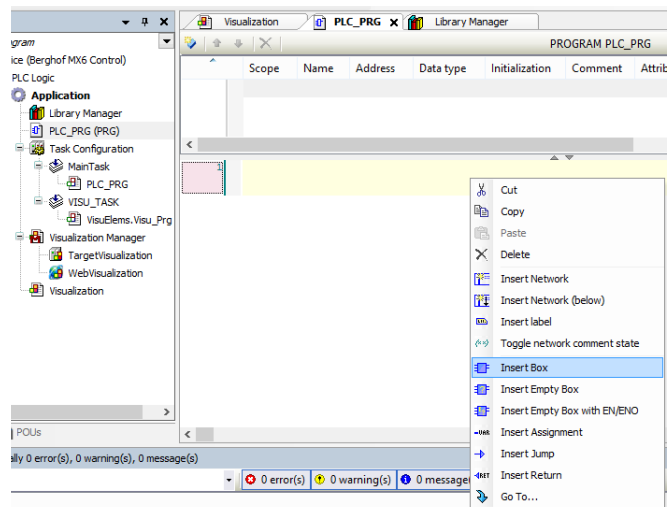


Рисунок 2.15 - Підключення бібліотеки *Util*

Функціональний блок *BLINK* має 3 входи, рисунок 2.16:

Enable - дозвіл на роботу (*BOOL*);
Timelow - скільки часу на виході буде 0;
Timehigh - скільки часу на виході буде 1;
Out - вихід логічного елементу (*BOOL*);

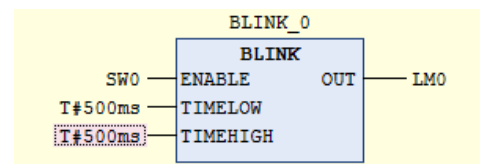


Рисунок 2.16 -

За допомогою *Assignment* необхідно привласнити змінну виходу елемента. Для вказівки часу роботи діода, необхідно написати наступну конструкцію: *T#500ms*.

Це означає, що на вхід логічного елемента подано інтервал часу в пів секунди (500 мілісекунд), рисунок 2.17.

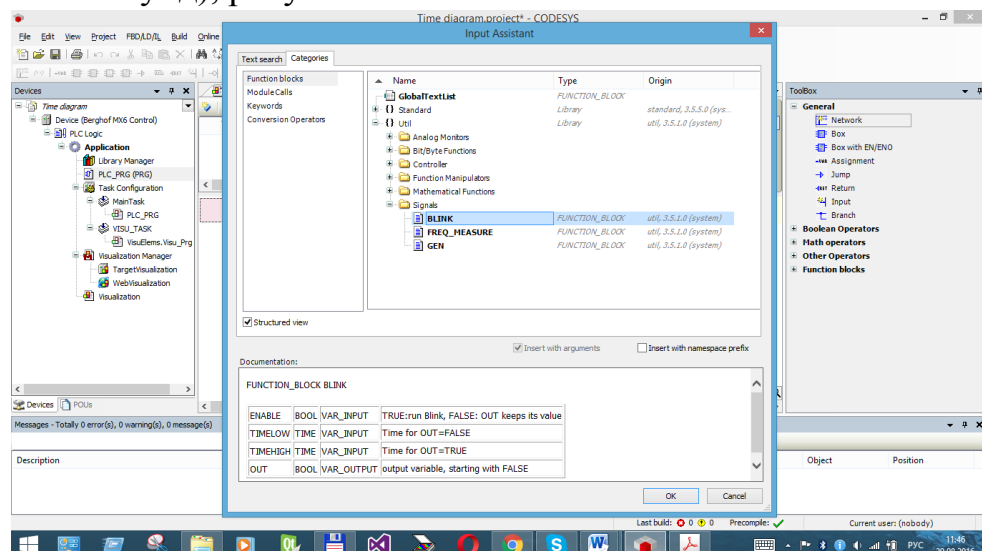


Рисунок 2.17 - Привласнення змінної виходу елемента.

Таким чином, діод підключений до цього функціонального блоку буде блимати з періодом в пів секунди. У загальному випадку значення *Timelow* і *Timehigh* можуть відрізнятися.

За допомогою інструменту *GeneralNetwork* на панелі інструментів створено ще 3 мережі і 3 блоки *BLINK* з періодами 1, 1.5, 2 секунди відповідно, рисунок 2.18.

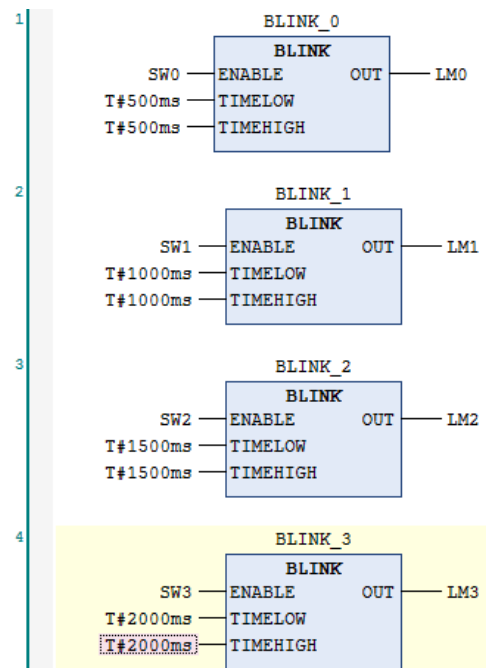


Рисунок 2.18 - Додаткові блоки *BLINK*

Далі створено в проекті файл візуалізації: перетягнути на вікно візуалізації 4 лампочки і 4 вимикача. Всі вимикачі працюють за одним принципом - вкл / вкл і відрізняються тільки зовнішнім виглядом.

При натисканні на елемент візуалізації, в правій частині програми з'являється панель інструментів для роботи з цим віджетом, рисунок 2.19. Для того, щоб з'єднати віджет зі змінною, необхідно в рядку *Variable* двічі клацнути і вибрати зі списку необхідну змінну. Наприклад, для вимикачів це будуть *SW0-SW3*, а для діодів *LM0-LM3*, рисунок 2.20.

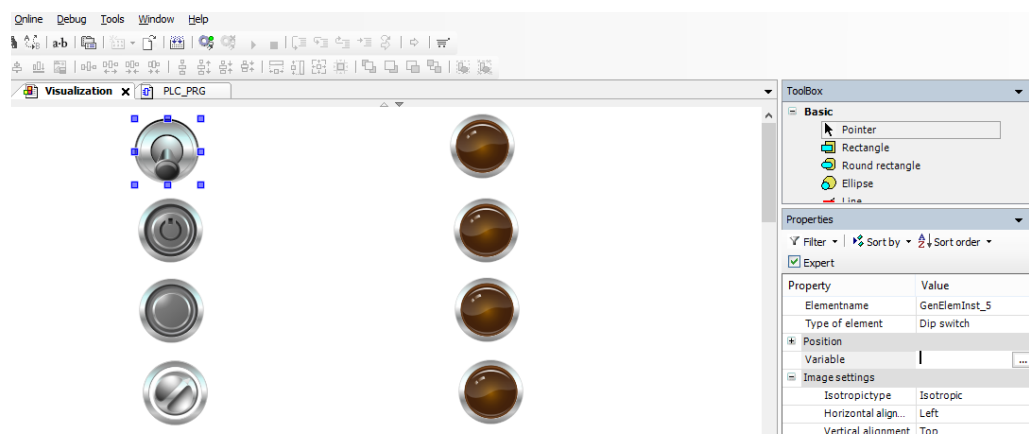


Рисунок 2.19 - Елементи візуалізації та панель інструментів

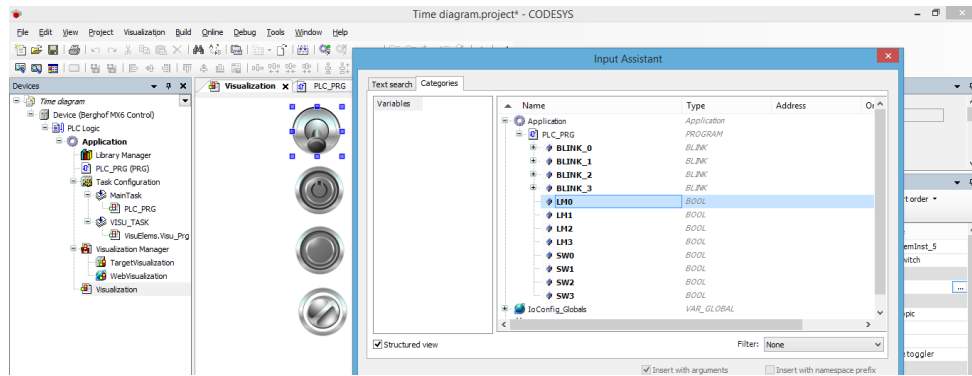


Рисунок 2.20 - Змінна LM0-LM3

Також в розділі *Background Image* можна змінити колір віджету. В вікні *Online* обрати *Simulation*. Запустити компіляцію, Логін та Запуск проекту, рисунок 2.21.

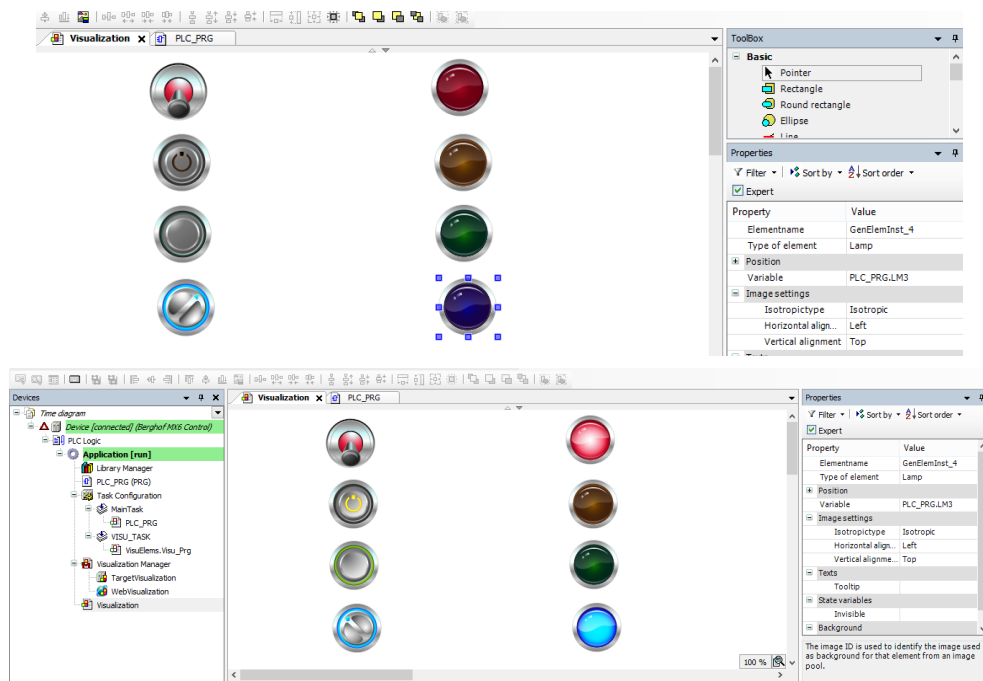


Рисунок 2.21 - Запуск проекту

Перелік джерел посилань

1. Programmable controllers: theory and implementation/L.A. Bryan, E.A. Bryan.—2nd ed. 1997
2. Посібник зі встановлення та використання Inline контролерів ILC 130 ETH, ILC 150 ETH ILC 155 ETH та ILC 170 ETH 2TX — 2010.
3. INTERNATIONAL STANDARD, CEI/IEC 61131-7, — 2006.
4. Getting Started With Qnx Neutrino 2: A Guide for Realtime Programmers —1999.
5. Розроблення людино-машинних інтерфейсів та систем збирання даних з використанням програмних засобів SCADA/HMI.: Навч.посіб. Київ : Видавництво Ліра-К, 2020. — 594 с.
6. Пупена О.М., Ельперін І.В., Програмування промислових контролерів у середовищі Unity Pro — 2021.
7. Промислові мережі та інтеграційні технології в автоматизованих системах. Пупена О.М., Ельперін І.В., Луцька Н.М., Ладанюк А.П. 2011