

**УКРАЇНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ЗАЛІЗНИЧНОГО ТРАНСПОРТУ**

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КЕРУЮЧИХ СИСТЕМ
ТА ТЕХНОЛОГІЙ**

**Кафедра автоматики та комп'ютерного телекерування
рухом поїздів**

МЕТОДИЧНІ ВКАЗІВКИ

**до виконання лабораторної роботи, самостійної роботи,
курсowego та дипломного проектування**

з дисципліни

«ІНТЕГРОВАНІ ІНФОРМАЦІЙНО-КЕРУЮЧІ СИСТЕМИ»

Частина 2

**Програмно-апаратні засоби розроблення
програмного забезпечення для мікропроцесорних контролерів**

Харків – 2023

Методичні вказівки розглянуто і рекомендовано до друку на засіданні кафедри автоматики та комп'ютерного телекерування рухом поїздів 4 вересня 2023 р., протокол № 1.

Наведено завдання та методичні рекомендації до виконання лабораторної, самостійної робіт, курсового та дипломного проектування, а також рекомендації до самостійного вивчення відповідних розділів курсу.

Методичні вказівки призначено для здобувачів вищої освіти другого (магістерського) рівня спеціальності 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка», які вивчають дисципліну «Інтегровані інформаційно-керуючі системи» усіх форм навчання, є другим виданням, переробленим і доповненим.

Укладачі:

доценти С. О. Змій,
О. В. Щебликіна

Рецензент

проф. В. І. Мойсеєнко

Перелік умовних скорочень

AUX – допоміжне завдання,
DFBs – похідні функціональні блоки,
EFs та EFBs – елементарні функції та елементарні функціональні
блоки,
EVTx – секції для обробки апаратно-керованих подій,
FAST – швидке завдання,
FBD – функціональна блок-схема,
FFBs – функціональні блоки, що виконують окремі функції,
IL – список інструкцій,
LD – мова східчастих діаграм,
MAST – завдання,
SFC – послідовне керування,
SR – секції підпрограм,
ST – структурований текст,
Timerx – секції для обробки подій, керованих часом.

ПРОГРАМНО-АПАРАТНІ ЗАСОБИ РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ МІКРОПРОЦЕСОРНИХ КОНТРОЛЕРІВ

Мета роботи:

- 1) дослідження принципів побудови програмного забезпечення;
- 2) моделювання та налагодження роботи мікропроцесорних контролерів.

1 Основні відомості

Сучасні вимоги до ринку електроніки включають вимоги не лише до якості продукції, але й інновацій, таких як нові пристрої, функції та кращі експлуатаційні характеристики. У той же час більшість сучасних електронних продуктів базується на мікроконтролерах.

Тому конкурентоспроможні продукти потребують як високоякісних, надійних і дорогих компонентів, так і сучасних, зручних для користувача, програмних і апаратних засобів програмування, які дають змогу якнайшвидше розробляти і модернізувати продукти.

Сьогодні компанія Schneider Electric є одним з провідних світових виробників контролерів для різних галузей промисловості [4].

Для розроблення, налагодження та моделювання використовується інтегроване середовище розроблення програмного забезпечення Unity Pro.

Unity Pro надає такі мови програмування для створення користувацьких програм:

- функціональних блок-схем (FBD);
- східчастих діаграм (LD);
- списку інструкцій (IL);
- структурованого тексту (ST);
- послідовного керування (SFC).

Усі ці мови програмування можуть використовуватися разом в одному проєкті. Ці мови програмування відповідають вимогам ІЕС 61131-3 [5].

Для проведення моделювання існують *бібліотеки блоків*.

Блоки, включені до розширених бібліотек для Unity Pro, варіюються від блоків для простих булевих операцій до блоків для операцій з рядками і масивами, а також блоків для керування складними контурами.

Для наочності різні блоки згруповані в бібліотеки і поділені на сімейства.

Блоки можуть використовуватися в мовах програмування FBD, LD, IL і ST.

Програма може бути створена:

- із завдання (MAST);
- швидкого завдання (FAST);
- одного з чотирьох допоміжних завдань (AUX);
- секцій для обробки подій, керованих часом (Timerx);
- секцій для обробки апаратно-керованих подій (EVTx);
- секцій підпрограм (SR).

Інтерфейс користувача складається з декількох вікон, що конфігуруються, і панелей інструментів.

Інтерфейс користувача (рисунок 1) відповідає всім вимогам людино-машинної системи. У таблиці 1 описані відповідні елементи інтерфейсу.

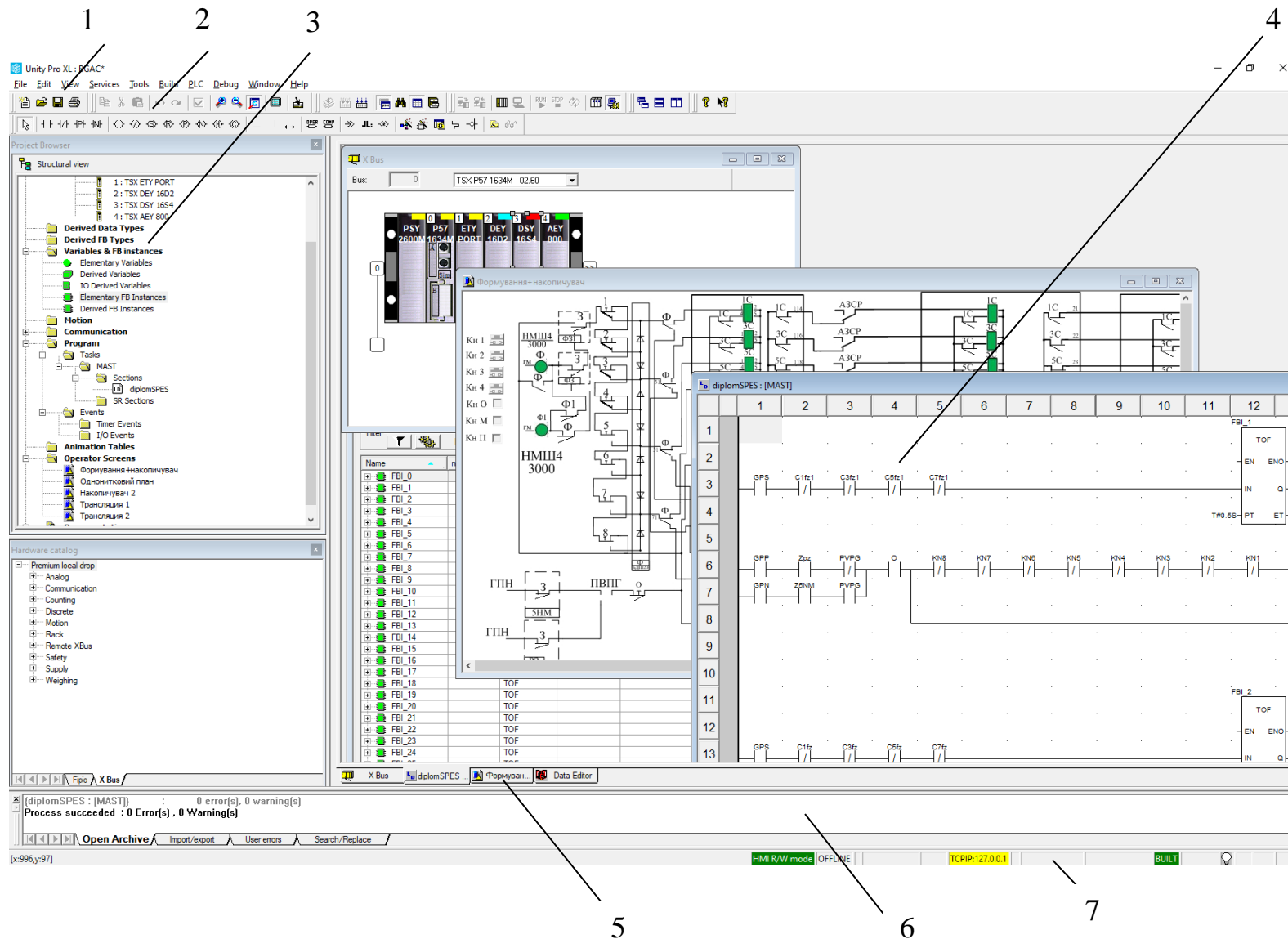


Рисунок 1 – Интерфейс програми

Таблиця 1 – Опис інтерфейсу програми

Номер	Опис
1	Рядок меню
2	Панель інструментів
3	Браузер проєкту
4	Вікно редактора (редактори мов програмування, редактор даних і т. ін.)
5	Вкладки покажчиків для прямого доступу до вікна редактора
6	Інформаційне вікно (видає інформацію про помилки, що сталися, стеження за сигналами, імпортовані функції і т. ін.)
7	Рядок стану

Мова східчастих діаграм (LD)

Структура секції LD відповідає ступеню для релейного перемикання.

Ліва шина живлення розташована в лівій частині редактора LD, відповідає фазі (L ladder – сходи) ступеня. У програмуванні LD, як і в ступені, тільки об'єкти LD, пов'язані з джерелом живлення, тобто сполучені з лівою шиною живлення, «виконуються». Права шина живлення відповідає нейтралі. Проте всі котушки і виходи FFB пов'язані з нею прямо або побічно, що і створює потік живлення.

Група об'єктів, які сполучені разом один під одним і не мають зв'язків з іншими об'єктами (виключаючи шину живлення), називається мережею або ступенем

На рисунку 2 зображено секцію LD.

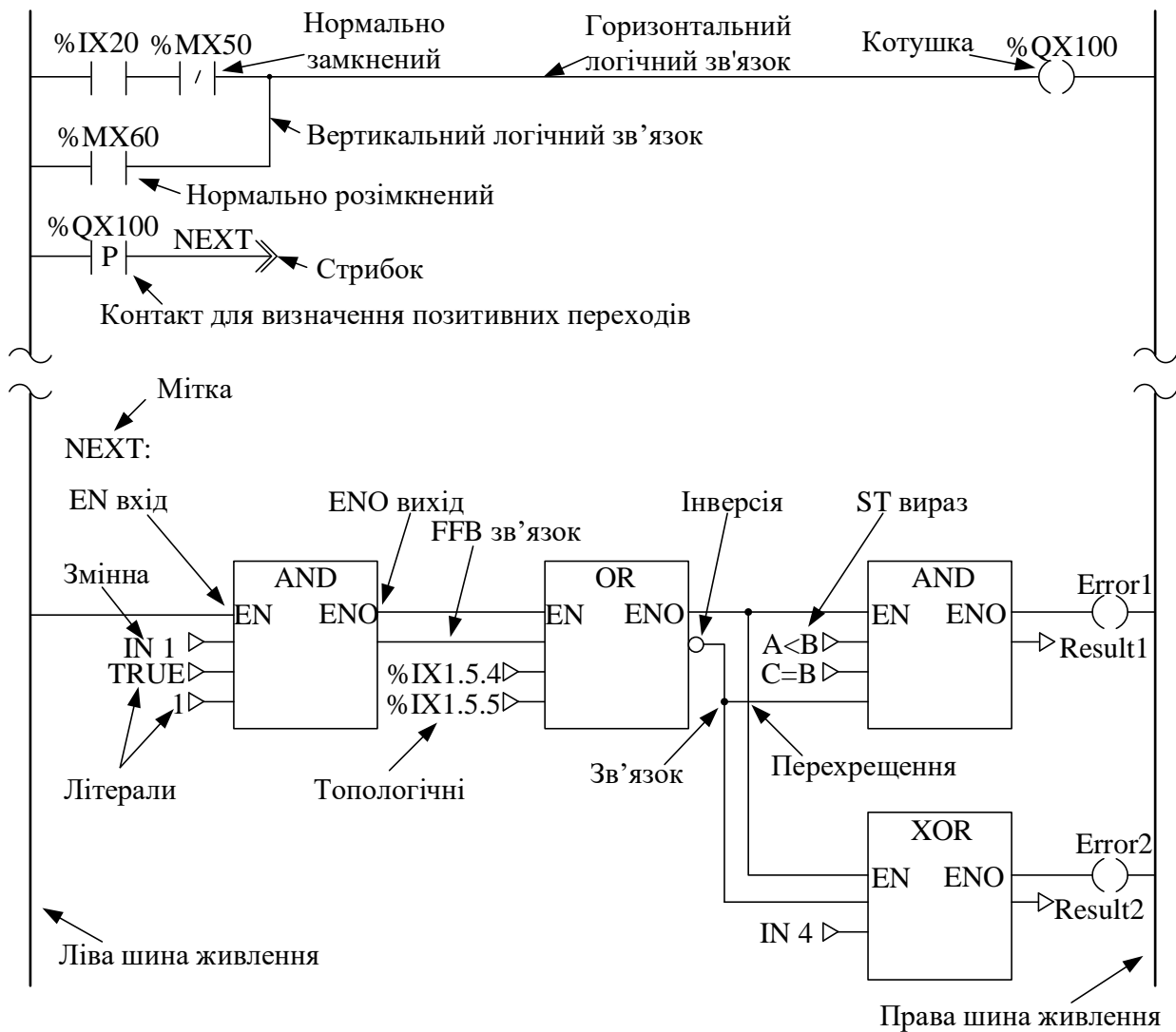


Рисунок 2 – Зображення секції LD

Об'єкти мови програмування LD дають змогу поділити секцію:

- на контакти;
- котушки;
- EFs і EFBs (елементарні функції та елементарні функціональні блоки);
- DFBs (похідні функціональні блоки);
- процедури;
- елементи, що керують;
- блоки виконання і порівняння, які являють собою доповнення до ІЕС 61131-3.

Ці об'єкти можуть бути сполучені один з одним за допомогою:

- зв'язків;
- фактичних параметрів (FFBs).

Коментарі логіки секції можуть бути забезпечені текстовими об'єктами.

Розмір секції

Одна секція LD складається з вікна з однією сторінкою.

Ця сторінка має сітку, яка поділяє секцію на стовпці і рядки.

Ширина 11-64 стовпців і 17-2000 рядків можуть бути визначена для секцій LD.

Мова програмування LD орієнтована на осередки, наприклад тільки один об'єкт може бути розміщений в одному осередку.

Послідовність виконання

Послідовність виконання окремих об'єктів у секції LD визначається потоком даних у секції. Мережі, сполучені з лівою шиною живлення, виконуються зверху вниз (зв'язок з лівою шиною живлення). Мережі, які не залежать одна від одної, виконуються в порядку положення (зверху вниз).

Контакт – це елемент LD, який передає статус горизонтального зв'язку на правий бік.

Контакт не змінює значення пов'язаного фактичного параметра.

Контакти займають один осередок.

Нижче наведені авторизовані фактичні параметри:

- логічні змінні;
- логічні константи;
- логічні адреси (топологічні адреси або символічні адреси);
- ST – вирази, які повертають логічний результат (наприклад VarA OR VarB).

ST-вирази, як фактичні параметри для контактів, є доповненням до IEC 61131-3 і мають дозволятися в такий спосіб.

У таблиці 2 надана характеристика доступних контактів.

Таблиця 2 – Характеристика доступних контактів

Призначення	Зображення контактів	Опис
Нормально розімкнений		Для нормально розімкнених контактів стан лівого зв'язку передається в правий зв'язок, якщо статус пов'язаного логічного фактичного параметра (вказується xxx) = ON. Інакше статус правого зв'язку – OFF
Нормально замкнений		Для нормально замкнених контактів стан лівого зв'язку передається в правий зв'язок, якщо статус пов'язаного логічного фактичного параметра (вказується xxx) = OFF. Інакше статус правого зв'язку – ON
Контакт для визначення позитивних переходів		Для контактів: для визначення позитивних переходів правий зв'язок циклу програми = ON, якщо перехід пов'язаного фактичного параметра (поміченого xxx) відбувається з OFF в ON, і статус лівого зв'язку = ON у той же час. Інакше статус правого зв'язку – OFF
Контакт для визначення негативних переходів		Для контактів: для визначення позитивних переходів правий зв'язок циклу програми = ON, якщо перехід пов'язаного фактичного параметра (поміченого xxx) відбувається з ON в OFF, і статус лівого зв'язку = ON у той же час. Інакше статус правого зв'язку – OFF

Елементи, що керують, використовуються для виконання переходів у межах секції LD і повернення з підпрограми (SRx) або похідного функціонального блока (DFB) в основну програму.

Елементи, що керують, займають один осередок.

У таблиці 3 надана характеристика доступних елементів, що керують.

Таблиця 3

Призначення	Зображення елементів	Опис
Перехід	NEXT —>>	<p>Коли стан лівого зв'язку 1, відбувається перехід до мітки (у поточній секції).</p> <p>Для генерування безумовного переходу об'єкт переходу має бути розміщений безпосередньо на лівій шині живлення.</p> <p>Для генерування умовного переходу об'єкт переходу розміщується в кінці набору контактів</p>
Мітка	LABEL:	<p>Мітки (цілі переходів) вказуються як текст із двокрапкою на кінці.</p> <p>Цей текст обмежений 32 символами і має бути унікальним у межах усієї секції, відповідати загальним угодам про іменування.</p> <p>Мітки переходів можуть розміщуватися тільки в першому осередку безпосередньо на шині живлення.</p> <p><i>Увага!</i> Мітки переходів не можуть «проходити крізь» мережі, передбачувана лінія від мітки переходу до правого краю секції не може перетинатися іншим об'єктом. Те саме і для логічних зв'язків і зв'язків FFB</p>
Повернення	—<RETURN>—	<p>Кожна підпрограма і кожен DFB (похідний функціональний блок) залишаються після виконання, відбувається повернення в основну програму.</p> <p>Якщо підпрограма/DFB залишається передчасно, повернення в основну програму може форсуватися об'єктом повернення.</p> <p>Якщо стан лівого зв'язку 1, виконується повернення з підпрограми або DFB (похідного функціонального блока) в основну програму.</p> <p>Об'єкти повернення можуть використовуватися тільки в DFBs- або SR-підпрограмах. Вони не можуть використовуватися в основній програмі.</p> <p>Для генерування умовного повернення об'єкт повернення розміщується в кінці набору контактів</p>

Мова функціональних блоків FBD

Редактор FBD використовується для графічного програмування функціонального плану відповідно до IEC 61131-3.

На рисунку 3 зображено секцію FBD.

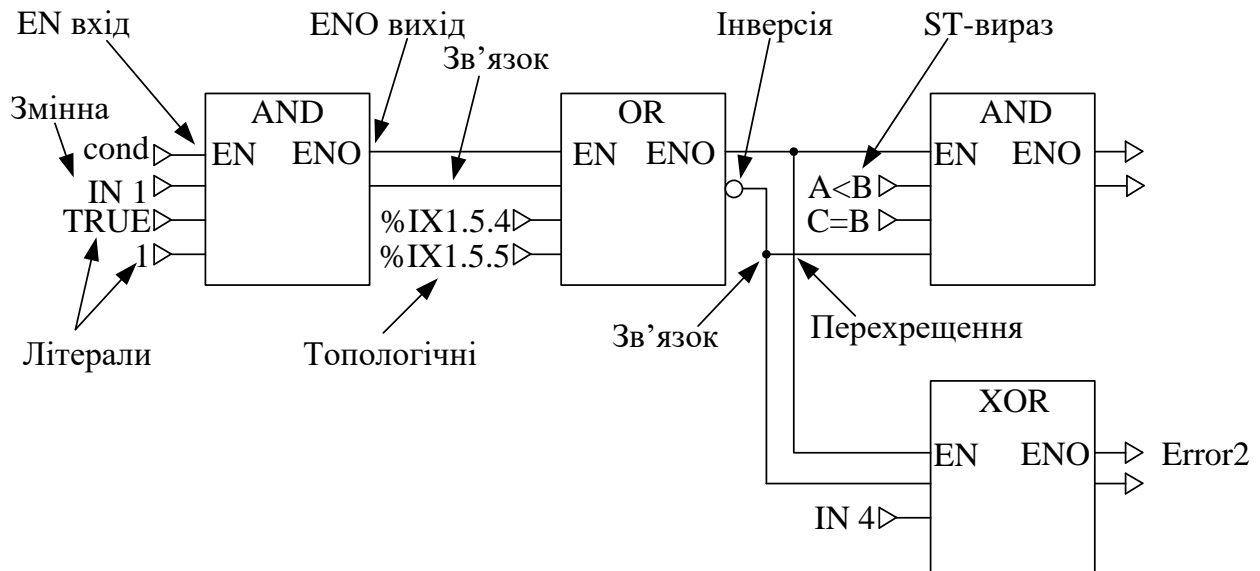


Рисунок 3 – Зображення секції FBD

Об'єкти мови програмування FBD (Function Block Diagram) поділяють секцію на декілька:

- EFs та EFBs (елементарні функції та елементарні функціональні блоки);
- DFBs (похідні функціональні блоки);
- процедур;
- елементів, що керують.

Ці об'єкти, об'єднані під ім'ям FFBs, можуть бути пов'язані один з одним за допомогою:

- зв'язків;
- фактичних параметрів.

Коментарі до логіки секції можуть бути забезпечені за допомогою текстових об'єктів.

Елементарна функція EF

Елементарні функції не мають внутрішніх станів. Якщо входні значення однакові, значення виходу буде однаковим для всіх викликів функції (наприклад складання двох значень дає однаковий результат при кожному виклику).

Елементарна функція (рисунок 4) зображується графічно як рамка з входами і одним виходом. Ці входи завжди розташовуються ліворуч від рамки, а вихід – праворуч.

Назва функції, наприклад тип функції, зображується в центрі рамки.

Номер виконання функції показаний праворуч від типу функції.

Лічильник функції показаний над рамкою. Лічильник функції – це порядковий номер функції в поточній секції. Лічильники функції не можуть змінюватися.

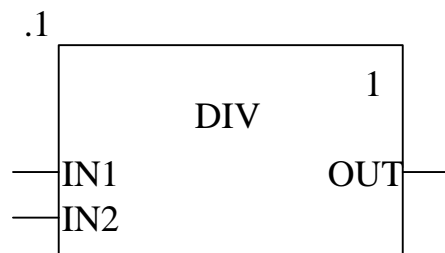


Рисунок 4 – Зображення елементарної функції

Кількість входів може бути збільшена деякими елементарними функціями.

Елементарний функціональний блок

Елементарні функціональні блоки (EFBs) мають внутрішні стани. Якщо входи мають однакові значення, значення виходу може бути різним для кожного виконання (наприклад у лічильнику значення виходу збільшується на 1).

Елементарний функціональний блок (рисунок 5) зображується графічно як рамка з входами і виходами. Входи завжди розташовуються ліворуч від рамки, а виходи – праворуч.

Функціональні блоки можуть мати більше за один вихід.

Назва функціонального блока, наприклад тип функціонального блока, зображується в центрі рамки.

Ім'я екземпляра є унікальним ідентифікатором для функціонального блока в проєкті.

Ім'я екземпляра створюється автоматично і має таку структуру:

$$FBI_n = \text{Екземпляр Функціонального Блока},$$

де n – порядковий номер функціонального блока в проєкті.

Це автоматично генероване ім'я може бути змінене для наочності. Ім'я екземпляра (максимум 32 символи) має бути унікальним у межах усього проєкту і не є чутливим до регістру.

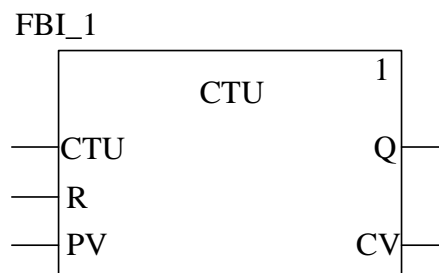


Рисунок 5 – Зображення елементарного функціонального блока

Похідні функціональні блоки (DFBs) мають такі самі характеристики, як і елементарні функціональні блоки. Користувач може створювати їх мовами FBD, LD, IL і/або ST.

Єдина відмінність від елементарних функціональних блоків полягає в тому, що похідний функціональний блок (рисунок 6) зображується як рамка з подвійними вертикальними лініями.

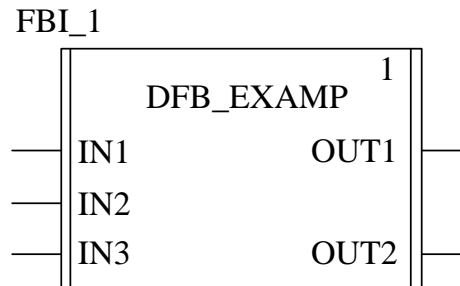


Рисунок 6 – Зображення похідного функціонального блока

Процедура

Формально процедури – це функції.

Єдина відмінність від елементарних функцій полягає в тому, що процедури можуть зайняти більше одного виходу і підтримують тип даних VAR_IN_OUT.

На вигляд процедури (рисунок 7) не відрізняються від елементарних функцій.

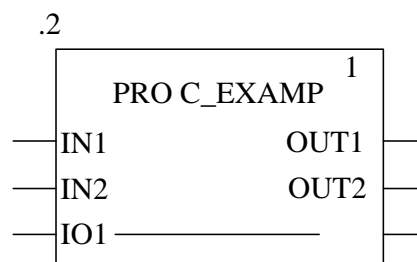


Рисунок 7 – Зображення процедури

Параметр

Входи (виходи) потрібні для передавання значень у(з) FFB, які є формальними параметрами.

Об'єкти приєднуються до формальних параметрів, які включають поточний стан процесу і є фактичними параметрами (рисунок 8).

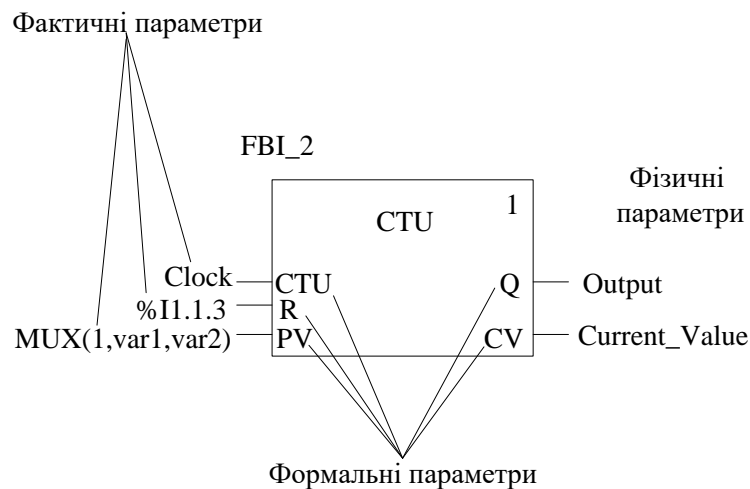


Рисунок 8 – Позначення параметрів

Під час виконання програми значення від процесу передаються в FFB через фактичні параметри і потім видаються після обробки.

Тільки один об'єкт (фактичний параметр) нижченаведених типів може бути пов'язаний із входами FFB:

- змінна;
- адреса;
- літера;
- ST-вирази;
- зв'язок.

Наступні комбінації об'єктів (фактичні параметри) можуть бути пов'язані з виходами FFB:

- одна змінна;
- одна змінна і одне або більше з'єднань;
- адреса;
- адреса і одне або більше з'єднань;
- одне або більше з'єднань.

Вхід EN і вихід ENO

Один вхід EN і один вихід ENO можуть використовуватися в усіх FFBS.

Якщо значення EN дорівнює «0», коли викликається FFBS, алгоритми не виконуються, і ENO встановлюється в «0».

Якщо значення EN дорівнює «1», коли викликається FFBS, алгоритми виконуватимуться. Після успішного виконання цих алгоритмів значення ENO встановлюється в «1». Якщо відбувається помилка під час виконання цих алгоритмів, ENO буде встановлений в «0».

Якщо ENO встановлений у «0» (викликається EN=0 або помилкою під час виконання):

- виходи функціонального блока зберігають стан з останнього циклу;
- виходи функції і процедури встановлюються в «0».

Поведінка виходу FFBS не залежить від того, чи викликався FFBS без EN/ENO або з EN=1.

Затримка вмикання – TOF

Функціональний блок використовується як затримка вмикання.

Коли функціональний блок викликається перший раз, початковий стан ET дорівнює «0».

ENO можуть бути конфігуровані як додаткові параметри.

На рисунку 9 подано зображення в LD.

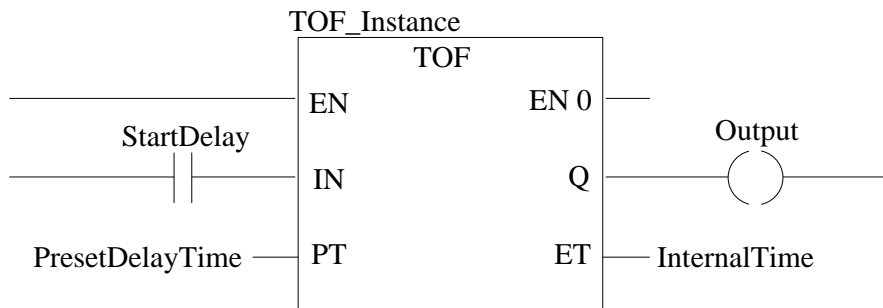


Рисунок 9 – Зображення в LD TOF

У таблицях 4 і 5 надано опис вхідних і вихідних параметрів TOF.

Таблиця 4 – Опис вхідних параметрів TOF

Параметр	Тип даних	Значення
IN	BOOL	Запуск затримки
PT	TIME	Передвстановлений час затримки

Таблиця 5– Опис вихідних параметрів TOF

Параметр	Тип даних	Значення
Q	BOOL	Вихід
ET	TIME	Внутрішній час

На рисунку 10 подана часова діаграма затримки вимикання TOF.

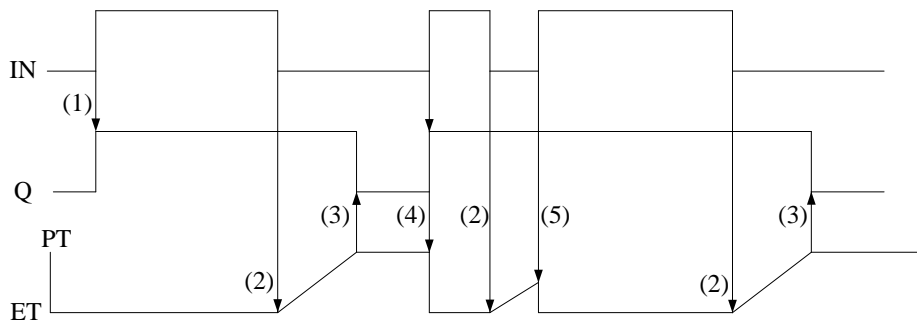


Рисунок 10 – Зображення часової діаграми TOF

Опис часової діаграми затримки вимикання TOF:

- якщо IN стає «1», Q стає «1»;
- якщо IN стає «0», запускається відлік внутрішнього часу (ET);
- якщо внутрішній час досягає значення PT, Q стає «0»;
- якщо IN стає «1», Q стає «1», а підрахунок внутрішнього часу зупиняється/скидається;
- якщо IN стає «1» до того, як внутрішній час досяг значення PT, підрахунок внутрішнього часу зупиняється/скидається, а вихід Q не встановлюється в «0».

Тип «TIME» (T# або TIME#) є беззнаковим подвійним цілим (UDINT) типом. Виражає тривалість у мілісекундах, яка відображує максимальну тривалість приблизно 49 днів.

Елементи часу, авторизовані для подання значення:

- дні (D);
- години (H);
- хвилини (M),
- секунди (S);
- мілісекунди (MS).

Наприклад, T#125 означає час – 125 мілісекунд; T#23M11S250MS – 23 хвилини 11 секунд і 250 мілісекунд.

Затримка вмикання – TON

Функціональний блок може бути використаний як затримка вмикання.

Коли функціональний блок викликається перший раз, початковий стан ET дорівнює «0».

ENO можуть бути конфігуровані як додаткові параметри.

На рисунку 11 подано зображення в LD.

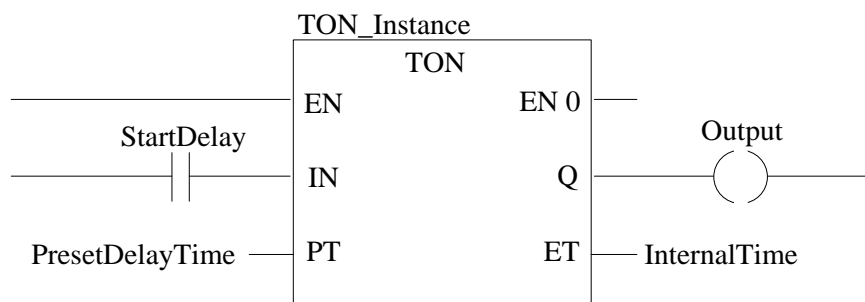


Рисунок 11 – Зображення в LD TON

У таблицях 6 і 7 надано опис вхідних і вихідних параметрів TON.

Таблиця 6 – Опис вхідних параметрів TON

Параметр	Тип даних	Значення
IN	BOOL	Запуск затримки
PT	TIME	Передвстановлений час затримки

Таблиця 7 – Опис вихідних параметрів TON

Параметр	Тип даних	Значення
Q	BOOL	Вихід
ET	TIME	Внутрішній час

На рисунку 12 подана часова діаграма зображення затримки вимикання TON.

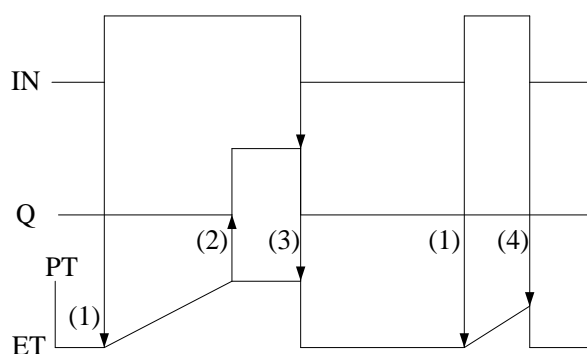


Рисунок 12 – Зображення часової діаграми затримки вимикання TON

Опис часової діаграми затримки вимикання TON:

- (1) якщо IN стає «1», запускається відлік внутрішнього часу (ET);
- (2) якщо внутрішній час досягає значення PT, Q стає «1»;
- (3) якщо IN стає «0», Q стає «0», а підрахунок внутрішнього часу зупиняється/скидається;
- (4) якщо IN стає «0» до того, як внутрішній час досяг значення PT, підрахунок внутрішнього часу зупиняється/скидається, а вихід Q не встановлюється в «0».

Імпульс – TP

Функціональний блок використовується для генерування імпульсу з заданою тривалістю.

Коли функціональний блок викликається перший раз, початковий стан ET дорівнює «0».

ENO можуть бути конфігуровані як додаткові параметри.

На рисунку 13 подано зображення в LD.

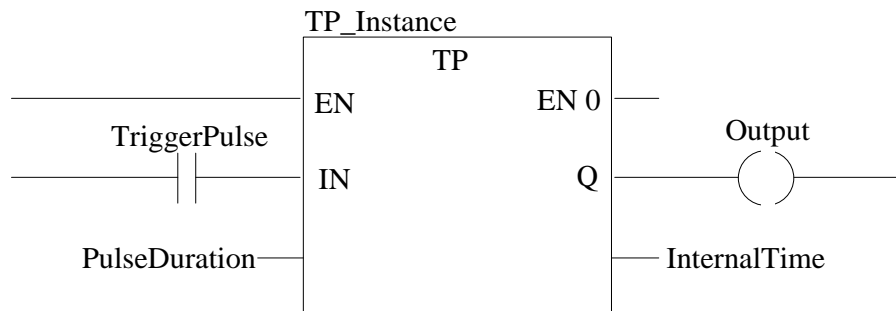


Рисунок 13 – Зображення в LD TP

У таблицях 8 і 9 надано опис вхідних і вихідних параметрів.

Таблиця 8 – Опис вхідних параметрів TP

Параметр	Тип даних	Значення
IN	BOOL	Запуск імпульсу
PT	TIME	Передвстановлений час імпульсу

Таблиця 9 – Опис вихідних параметрів TP

Параметр	Тип даних	Значення
Q	BOOL	Вихід
ET	TIME	Внутрішній час

На рисунку 14 зображено імпульс TP.

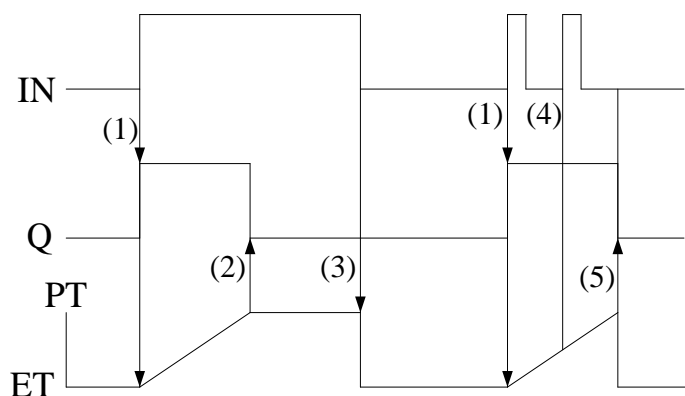


Рисунок 14 – Зображення імпульсу TR

Опис часової діаграми затримки вимикання TR:

- (1) якщо IN стає «1», Q стає «1», і починається відлік внутрішнього часу (ET);
- (2) якщо внутрішній час досягає значення PT, Q стає «0» (незалежно від IN);
- (3) відлік внутрішнього часу зупиняється/скидається, якщо IN стає «0»;
- (4) якщо внутрішній час не досяг значення PT, імпульс IN не впливає на внутрішній час;
- (5) якщо внутрішній час досяг значення PT, і IN дорівнює «0», відлік внутрішнього часу зупиняється/скидається, і Q стає «0».

Недозволені петлі

Конфігурація петель прямо через зв'язки не дозволена (рисунок 15), оскільки неможливо чітко визначити сигнальний потік (вихід одного FFБ є входом наступного FFБ і вихід цього є входом першого).

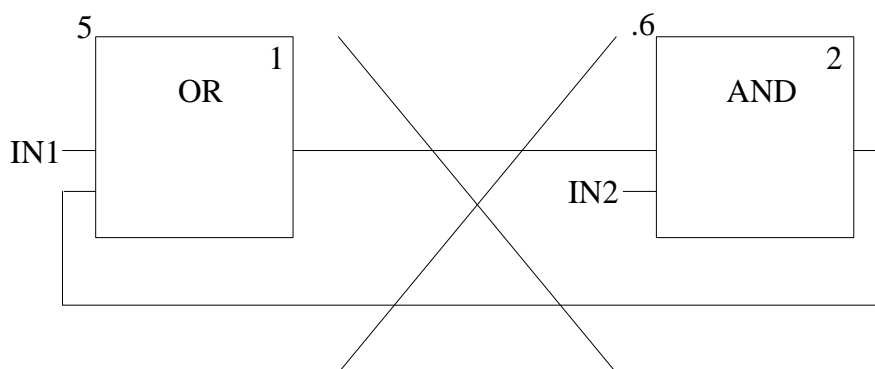


Рисунок 15 – Конфігурація петель не дозволена

2 Програма виконання лабораторної роботи

1 Ознайомитися з теоретичним матеріалом, наведеним у конспекті лекцій і навчальних посібниках і підручниках [1-5].

2 Запустити програму «Unity Pro XL», що знаходиться: Пуск -> Програми -> Schneider Electric -> SoCollaborative -> Unity Pro.

3 Створити новий проєкт, для чого необхідно в рядку меню вибрати «New». У відкритому вікні вибрати в розділі Premium контролер типу TSX P57 1634M.

4 Конфігурування контролера.

4.1 У Project Browser натиснути лівою кнопкою «миші» два рази на рядку «Configuration».

4.2 У вікні, що з'явилося, натиснути правою кнопкою «миші» на елементі з цифрою «2» і вибрати в меню «New device» (рисунок 16).

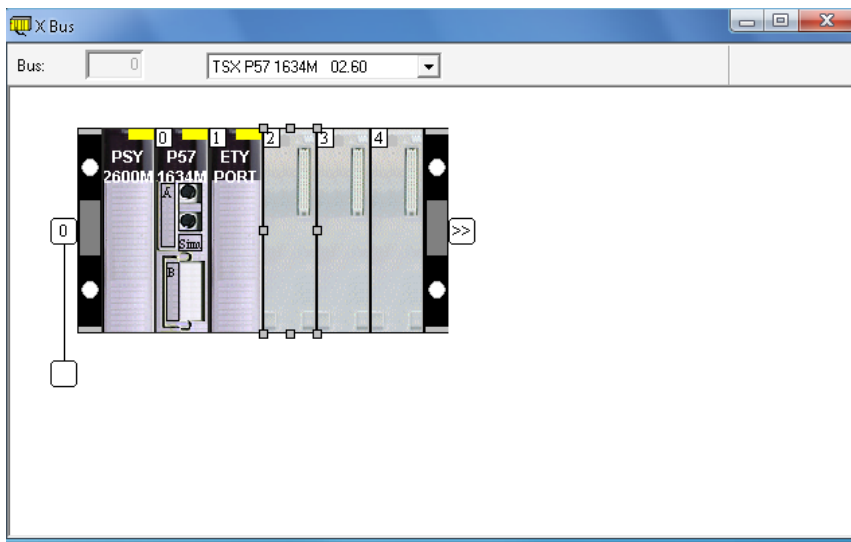


Рисунок 16 – Вибір модуля введення «TSX DEY 16D2»

Вибрати в розділі «Discrete» модуль введення «TSX DEY 16D2».

4.3 Повторити пункт 4.2 для елемента «3», але вибрати модуль виведення «TSX DSY 16S4».

4.4 Повторити пункт 4.2 для елемента «4», але вибрати в розділі «Analog» модуль аналогового введення «TSX AEY 800».

4.5 У Project Browser вибрати «Communication», натиснути правою кнопкою «миші» на «Networks» та вибрати рядок «New Network».

Заповнити поля вікна, що з'явилося, відповідно до рисунка 17.

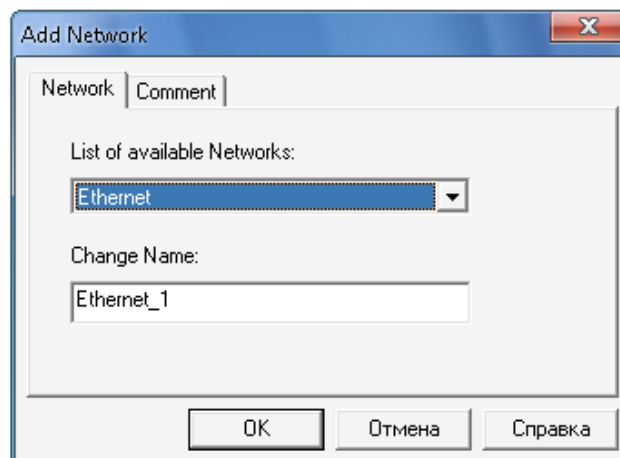


Рисунок 17 – Вікно підключення нової мережі

4.6 У Project Browser натиснути лівою кнопкою «миші» два рази по рядку «Configuration». Вибрати елемент TSX ETY PORT під цифрою «1». У вікні, що з'явилося, натиснути лівою кнопкою «миші» по рядку «Chanel 0» і сконфігурувати його відповідно до рисунка 18.

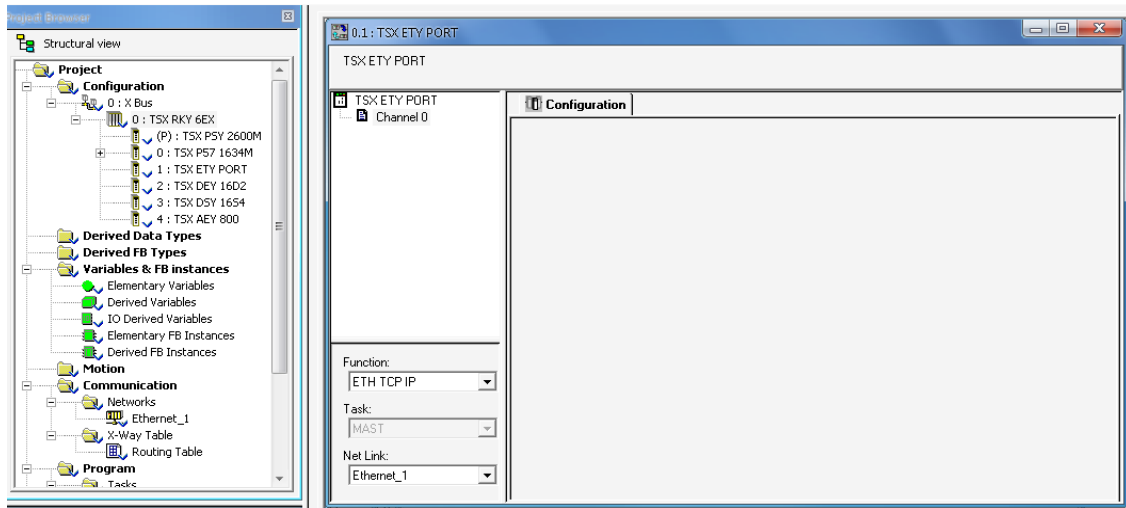


Рисунок 18 – Вікно конфігурації каналу зв'язку

4.7 У меню «Edit» вибрати «Validate» для закінчення конфігурації.

5 Створення програми.

5.1 У меню «Tools» (рисунок 19) вибрати «Project Settings». Вибрати «Program» -> «Languages» і натиснути лівою кнопкою «миші» на пункті «LD». Встановити кількість стовпців у програмі – 25.

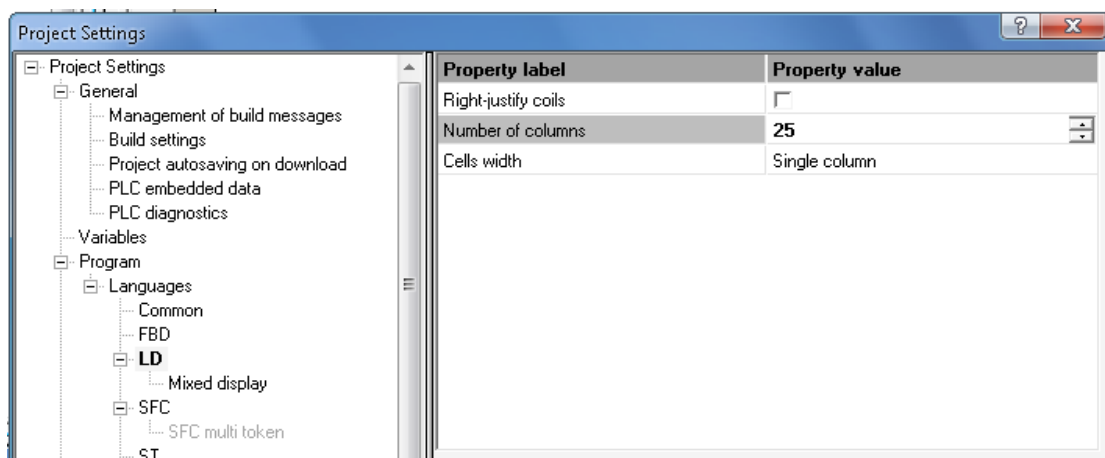


Рисунок 19 – Установлення параметрів вікна програми

5.2 У Project Browser вибрати «Program» -> «Tasks» -> «Mast», натиснути правою кнопкою «миші» на «Sections» і вибрати рядок «New Section».

У вікні, що з'явилося, вибрати основну мову програмування (пункт «Language») – LD. Ім'я проєкту може бути довільним, але унікальним і латинськими літерами.

Натиснути кнопку «Ok».

5.3 У Project Browser у підрозділі «Program» -> «Tasks» -> «Mast», -> «Sections» з'явиться файл з ім'ям, що було вписано в попередньому пункті. Вибрати файл, натиснувши на ньому двічі лівою кнопкою «миші».

5.4 Відповідно до завдання, що обирається з таблиці 10 за вказівкою викладача, розробити програму роботи відповідної схеми.

Вихідні дані подані в роботах [1 – 3].

Таблиця 10

Варіант завдання	Номер завдання
Вмикання вогнів вхідного світлофора	1
Блок СП ВГ БМРЦ	2
Блок МІ ВГ БМРЦ	3
Блок ВІ ВГ БМРЦ	4
Блок ВД ВГ БМРЦ	5
Блок УП ВГ БМРЦ	6
Блок МП ВГ БМРЦ	7
Блок ВП ВГ БМРЦ	8
Блок МП ВГ БМРЦ	9
Блок ПС БМРЦ	10
Блок НПМ НГ БМРЦ	11
Блок НМІ НГ БМРЦ	12
Блок НМП НГ БМРЦ	13
Блок НН НГ БМРЦ	14
Блок НМП НГ БМРЦ	15
Формування та реєстрація завдань	16
Накопичувач завдань БГАЦ	17
Блоки трансляції завдань БГАЦ	18
Керування гірковою стрілкою	19

Основні елементи обраної мови програмування розміщено на панелі інструментів (рисунок 20).

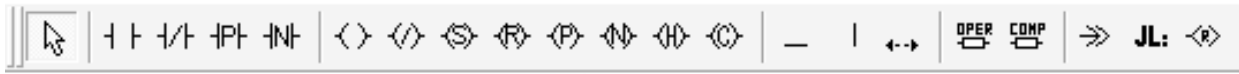


Рисунок 20 – Панель інструментів

Для додавання функцій необхідно:

- натиснути правою кнопкою «миші» на вільній частині поля та вибрати пункт «FFB Input Assistant»;
- у вікні, що з'явилося, натиснути «...» у рядку «FFB type» (рисунок 21);
- у вікні, що з'явилося, необхідно знайти необхідну функцію, наприклад «TOF» (рисунок 22);

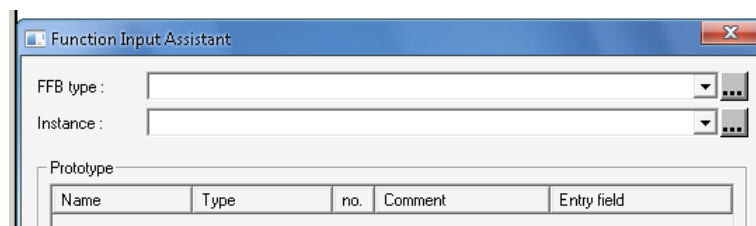


Рисунок 21 – Вікно додавання FFB

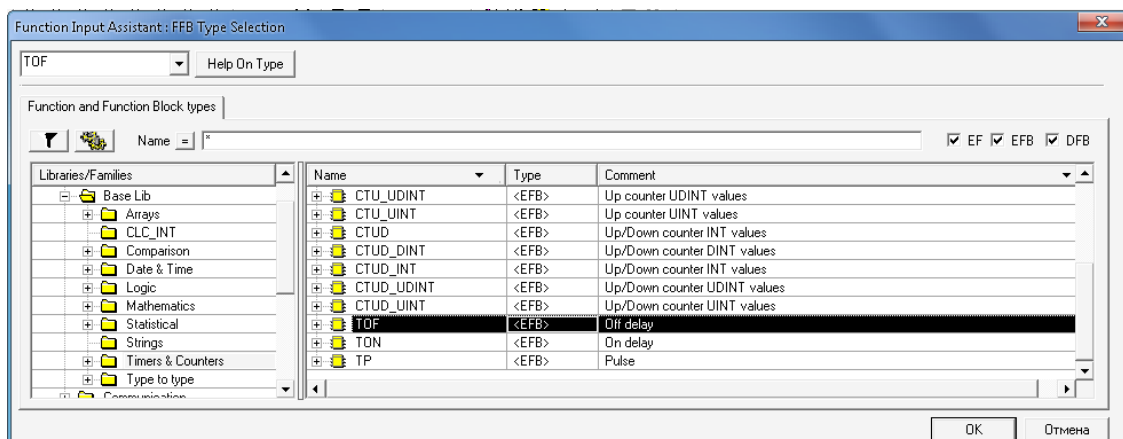


Рисунок 22 – Вікно вибору FFB

– після вибору необхідної функції з'являється вікно, у якому необхідно ввести необхідні параметри у стовпчик «Entry field» (рисунок 23).

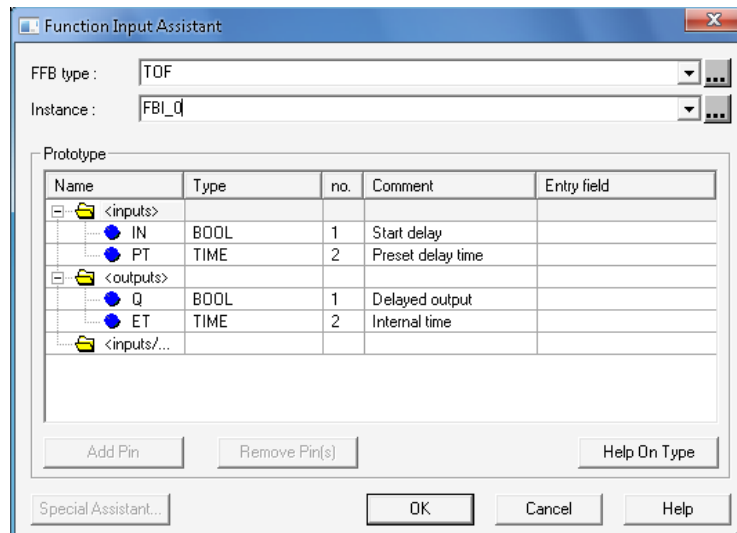


Рисунок 23 – Зображення вікна для введення параметрів

6 Запуск програми на віртуальному пристрої та моделювання.

6.1 Перед запуском програми необхідно:

– створити проєкт: вибрати в меню пункт «Build» -> «ReBuild All Project»;

– увімкнути віртуальний пристрій: вибрати в меню пункт «PLC» -> «Connect»;

– записати до віртуального пристрою програму: вибрати в меню пункт «PLC» -> «Transfer Project to PLC»;

– у вікні, що з'явилося, підтвердити запис натисканням кнопки «Transfer».

6.2 Для запуску роботи програми необхідно вибрати в меню пункт «PLC» -> «Run».

Умовно струм, що протікає, зображено стовщеною лінією зеленим кольором (рисунок 24).

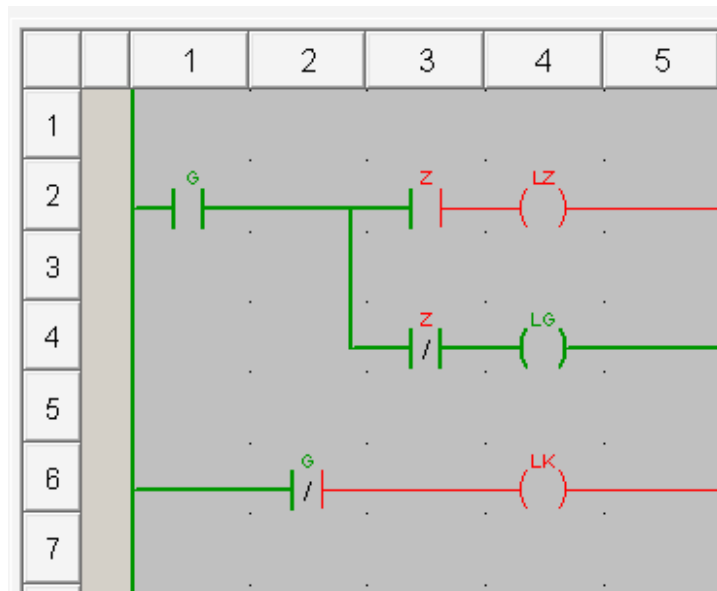


Рисунок 24 – Зображення частини програми

Для зміни стану контактів, що не мають обмотки або кнопок, необхідно натиснути правою кнопкою «миші» на контакт, що не має обмотки, або кнопки, обрати пункт «Set Value» і встановити значення «1» або «0».

6.3 Для наочності моделювання середовище Unity Pro має можливості графічного відображення процесів роботи програми.

Для цього необхідно в Project Browser у підрозділі «Operator Screens» натиснути праву кнопку «миші» та обрати пункт «New Screen». У вікні, що з'явилося, натиснути кнопку «Ok».

У Project Browser у підрозділі «Operator Screens» з'явиться файл з «Screen». Вибрати файл, натиснувши на ньому двічі лівою кнопкою «миші».

Основні елементи для створення рисунків розміщено на панелі інструментів (рисунок 25).



Рисунок 25 – Основні елементи створення рисунків

Для переходу з режиму редагування до режиму моделювання необхідно вибрати в меню пункт «Services»->«Enable Variable Modification» або натиснути кнопку F7.

7 Підготувати звіт відповідно до п. 3.

8 Письмово у звіті відповісти на контрольні запитання.

9 Отримати допуск до відпрацьовування лабораторної роботи.

10 Закінчити оформлення звіту відповідно до п. 3.

3 Зміст звіту

1 Назва і мета роботи.

2 Письмові відповіді на контрольні запитання.

3 Принципова схема згідно з завданням.

4 Програма, написана мовою LD.

5 Графічне відображення моделювання роботи програми.

6 Висновки з роботи.

Контрольні запитання

1 Назвіть мови програмування, що використовуються Unity Pro для створення програм.

2 Що являє собою мова східчастих діаграм?

3 Вкажіть недоліки та переваги мови східчастих діаграм.

4 Що являє собою мова функціональних блоків?

5 Вкажіть недоліки та переваги мови функціональних блоків.

6 Чому не дозволена конфігурація петель прямо через зв'язки?

СПИСОК ЛІТЕРАТУРИ

1 Автоматизовані станційні системи керування рухом поїздів / В. І. Мойсеєнко та ін.; за заг. ред. В. І. Мойсеєнка. Харків, 2013. 393 с.

2 Báez-Lópe D., Guerrero-Castro E., Circuit F. Analysis with Multisim. Synthesis lectures on digital circuits and systems. 2011. 200 p. URL: <https://ece.charlotte.edu/wp-content/uploads/sites/301/2023/05/circuit-analysis-with-multisim.pdf>.

3 Варбанець М. Г. Системи залізничної автоматики і телемеханіки: навч. посіб. для студ. вищ. навч. закл. Харків : УкрДАЗТ, 2008. 190 с.

4 Сайт підтримки Schneider Electric. URL: <http://www.schneider-electric.com>.

5 Засоби автоматизації. Електронна бібліотека документації. Schneider Electric. 2005. URL: <http://www.schneider-electric.com>.

6 Unity Pro: Program Languages and Structure Reference Manual / Schneider Electric, 2012. 722 p. URL: http://users.isr.ist.utl.pt/~jag/courses/mapi22d/docs/PLC_2_Unity_Reference.pdf.

7 Пупена О. М., Ельперін І. В. Програмування промислових контролерів у середовищі Unity Pro : навч. посіб. Київ : Вид-во Ліра, 2013. 376 с.

8 Типові матеріали для проектування: 410417-ТМП, 410207-ТМП, 410108-ТМП, 410905-ТМП, 410104-ТМП, I-320-08, 410811-ТМП. URL : <http://scbist.com/wiki/7670-tmp.html>.

МЕТОДИЧНІ ВКАЗІВКИ

до виконання лабораторної роботи, самостійної роботи,
курсowego та дипломного проєктування

з дисципліни

«ІНТЕГРОВАНІ ІНФОРМАЦІЙНО-КЕРУЮЧІ СИСТЕМИ»

Частина 2

Програмно-апаратні засоби розроблення
програмного забезпечення для мікропроцесорних контролерів

Відповідальний за випуск Щебликіна О. В.

Редактор Ібрагімова Н. В.

Підписано до друку 30.10.2023 р.
Умовн. друк. арк. 2,0. Тираж . Замовлення № .
Видавець та виготовлювач Український державний університет залізничного
транспорту,
61050, Харків-50, майдан Фейєрбаха,7.
Свідоцтво суб'єкта видавничої справи ДК № 6100 від 21.03.2018 р.