

ФАКУЛЬТЕТ АВТОМАТИКИ, ТЕЛЕМЕХАНІКИ ТА ЗВ'ЯЗКУ

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

**ДОСЛІДЖЕННЯ ПРОГРАМНО-АПАРАТНИХ
ЗАСОБІВ ПРОГРАМУВАННЯ
МІКРОПРОЦЕСОРНИХ КОНТРОЛЕРІВ**

МЕТОДИЧНІ ВКАЗІВКИ

**до виконання лабораторних занять, самостійної роботи,
курсowego та дипломного проектування з дисципліни
*"СТАНЦІЙНІ СИСТЕМИ АВТОМАТИКИ"***

Харків - 2014

Методичні вказівки розглянуто і рекомендовано до друку
на засіданні кафедри автоматики та комп'ютерного

телекерування рухом поїздів 18 січня 2012 р., протокол № 5.

Методичні вказівки призначені для студентів спеціальності 7.05020203 «Автоматика та автоматизація на транспорті», що вивчають дисципліну «Станційні системи автоматики» усіх форм навчання.

Укладачі:

доц. В.П. Мороз,
асистенти С.О. Змій,
Р.В. Турчинов

Рецензент

проф. В.Ш. Хісматулін

ДОСЛІДЖЕННЯ ПРОГРАМНО-АПАРАТНИХ
ЗАСОБІВ ПРОГРАМУВАННЯ
МІКРОПРОЦЕСОРНИХ КОНТРОЛЕРІВ

МЕТОДИЧНІ ВКАЗІВКИ

до виконання лабораторних занять, самостійної роботи, курсового та
дипломного проектування з дисципліни
"СТАНЦІЙНІ СИСТЕМИ АВТОМАТИКИ"

Відповідальний за випуск Мороз В.П.

Редактор Буранова Н.В.

Підписано до друку 06.04.12 р.

Формат паперу 60x84 1/16. Папір писальний.

Умовн.-друк.арк. 1,5. Тираж 100. Замовлення №

Видавець та виготовлювач Українська державна академія залізничного транспорту,
61050, Харків-50, майдан Фейєрбаха, 7.
Свідоцтво суб'єкта видавничої справи ДК № 2874 від 12.06.2007 р.

УКРАЇНСЬКА ДЕРЖАВНА АКАДЕМІЯ
ЗАЛІЗНИЧНОГО ТРАНСПОРТУ

ФАКУЛЬТЕТ АВТОМАТИКА, ТЕЛЕМЕХАНІКА
ТА ЗВ'ЯЗОК

Кафедра «Автоматика та комп'ютерне телекерування
рухом поїздів»

МЕТОДИЧНІ ВКАЗІВКИ

до виконання лабораторних занять, самостійної роботи,
курсowego та дипломного проектування
з дисципліни: "СТАНЦІЙНІ СИСТЕМИ АВТОМАТИКИ"
«Дослідження програмно-апаратних засобів програмування
мікропроцесорних контролерів»

для студентів спеціальності 7.05020203 «Автоматика і
автоматизація на транспорті»

Харків 2014

Методичні вказівки розглянуто і рекомендовано до друку на засіданні кафедри «Автоматика та комп'ютерне телекерування рухом поїздів» 18 січня 2012 р., протокол № 5.

Методичні вказівки призначені для студентів спеціальності «Автоматика та автоматизація на транспорті», що вивчають дисципліну «Станційні системи автоматики» усіх форм навчання.

Укладачі:

доц. В.П. Мороз,
асистенти С.О. Змій,
Р.В. Турчинов

Рецензент

проф. В.Ш. Хісматулін

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

ЛАБОРАТОРНА РОБОТА

Дослідження програмно-апаратних засобів програмування мікропроцесорних контролерів

1 Мета роботи

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

2 Основні відомості

Сучасні вимоги до ринку електроніки складаються не лише з вимог до якості продукції, але і до новизни виробів, наприклад, нові пристрої, нова функціональність, кращі характеристики тощо. Разом з цим, на сьогодні більша частина електронних виробів базується на мікроконтролерах.

Тому для випуску конкурентної продукції потрібні як якісні, надійні і збалансовані за ціною компоненти, так і сучасні та зручні програмно-апаратні засоби програмування, що дозволяють у найкоротші терміни розробляти і модернізувати продукцію.

Компанія Schneider Electric на сьогодні є однією з передових у світі з виробництва контролерів для різних сфер промисловості [4].

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

Unity Pro надає до використання такі мови програмування для створення програм користувача:

- 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 проекті. Ці мови програмування відповідають вимогам ІЕС 61131-3 [5].

Для проведення моделювання існують *бібліотеки блоків*.

Блоки, які включені в розширені бібліотеки 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 об'єднані в бібліотеки і потім розподілені на сімейства.

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

Програма може бути створена:

- із завдання (MAST);
- швидкого завдання (FAST);
- одного до 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 усім вимогам людино-машинних систем. У таблиці 1 надано опис відповідних елементів інтерфейсу.

Мова східчастих діаграм (LD)

Структура секції LD відповідає ступеню для релейного перемикачання.

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

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

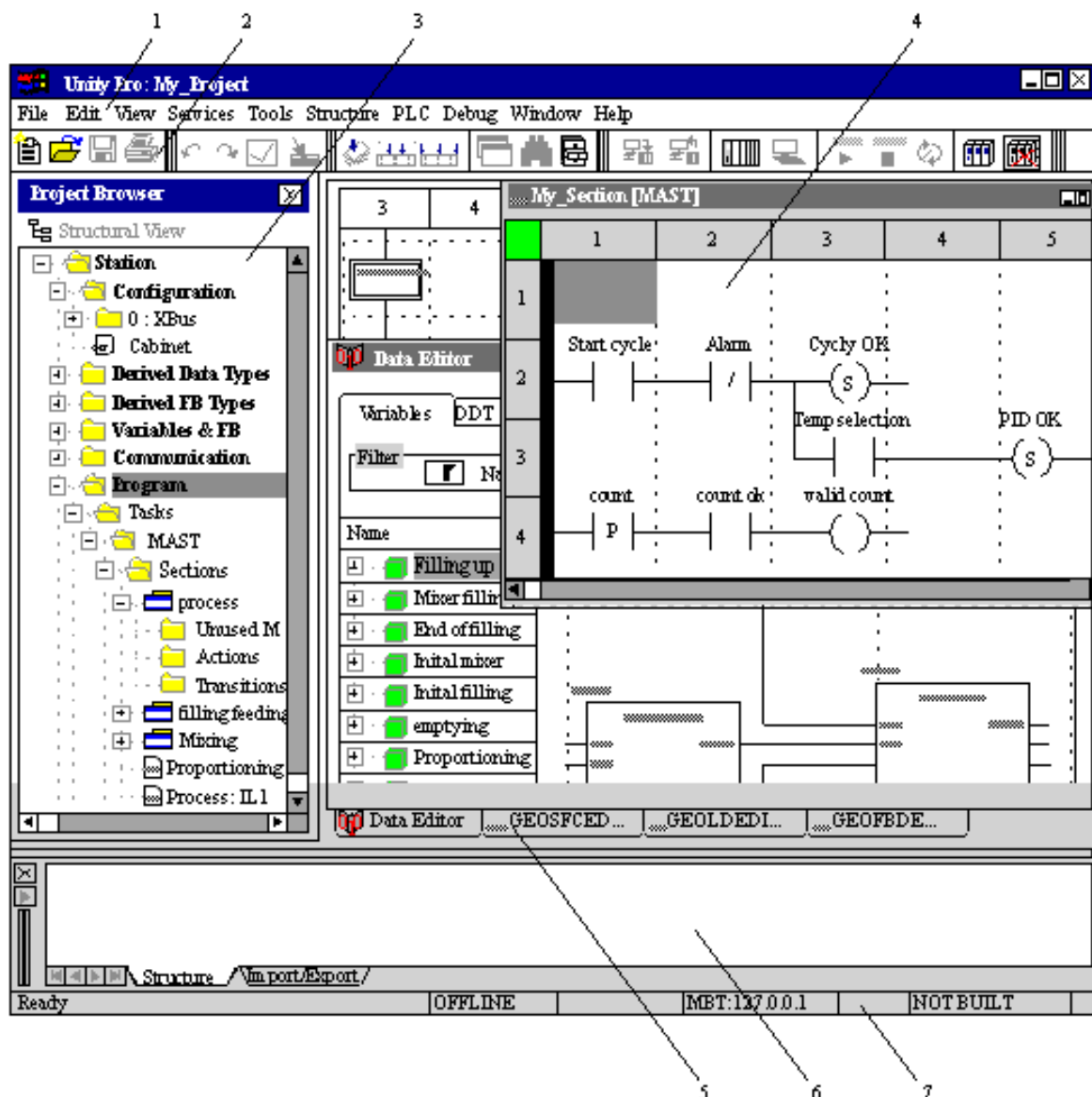


Рисунок 1 – Інтерфейс програми

Таблиця 1

Номер	Опис
1	Рядок меню
2	Панель інструментів
3	Браузер проекту
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.)

На рисунку 2 подано зображення секції LD.

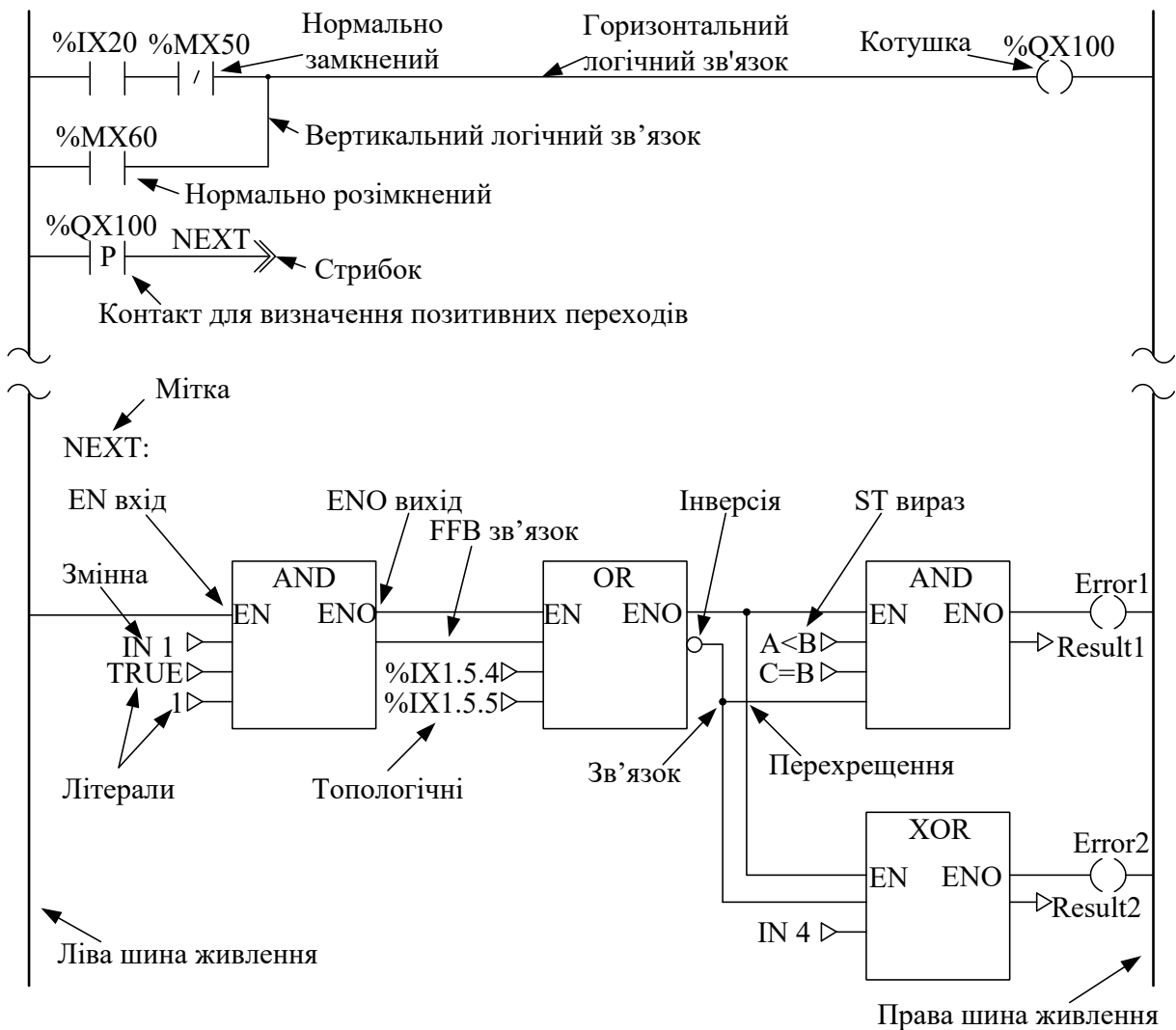


Рисунок 2 – Зображення секції LD

Об'єкти мови програмування LD дозволяють секцію ділити:

- 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 IEC 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 ділить секцію на стовпці і рядки.

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

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

Послідовність виконання

Послідовність виконання окремих об'єктів в секції LD визначається потоком даних у секції. Мережі, що 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 праву сторону.

Контакт не змінює значення пов'язаного фактичного параметра.

Контакти займають один осередок.

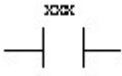
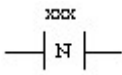
Нижче наведені авторизовані фактичні параметри:

- 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 явним чином.

У таблиці 2 надана характеристика доступних контактів.

Таблиця 2

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 в 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 в той же час. Інакше, статус правого зв'язку – OFF.

Елементи, що керують, використовуються для виконання переходів у межах секції LD і для 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 у 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. Текст повинен відповідати загальним угодам про іменування.</p> <p>Мітки переходів можуть розміщуватися тільки в першому осередку безпосередньо на шині живлення.</p> <p>Примітка – Мітки переходів не можуть "проходити крізь" мережі; передбачувана лінія від мітки переходу до правого краю секції не може перетинатися іншим об'єктом. Це також дійсно для логічних зв'язків і зв'язків FFB.</p>
Повернення	—<RETURN>—	<p>Кожна підпрограма і кожен DFB (похідний функціональний блок) залишається 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 у кінці набору контактів.</p>

Мова функціональних блоків FBD

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

На рисунку 3 подано зображення секції FBD.

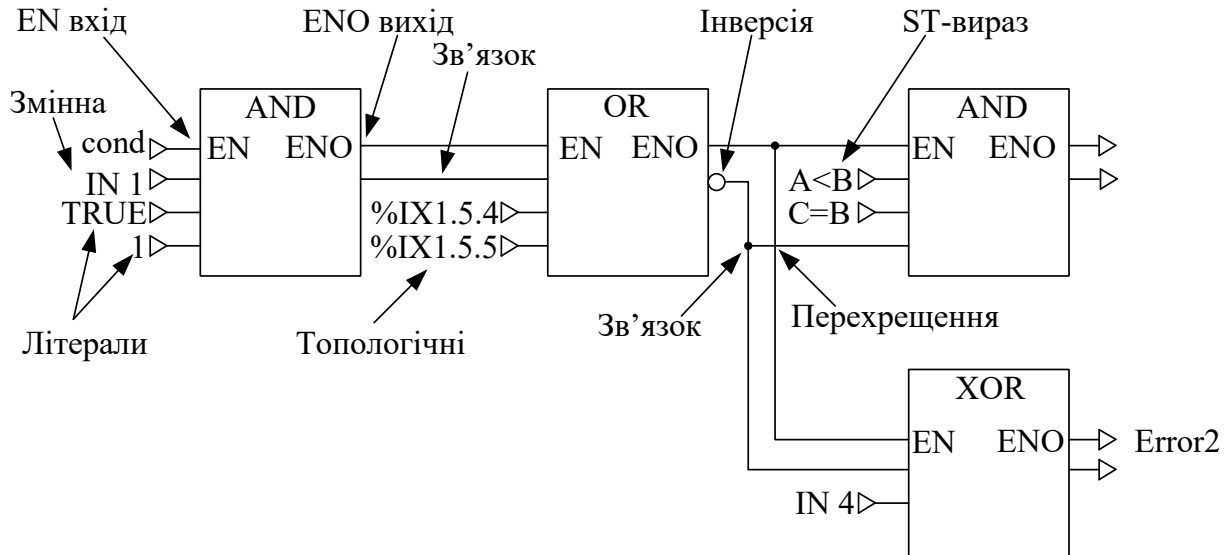


Рисунок 3 – Зображення секції FBD

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

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

Ці об'єкти, що 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 вхідні 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 із входами і одним виходом. Ці входи завжди розташовуються зліва від рамки, а вихід – справа.

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

Номер виконання функції показаний справа від типу функції.

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

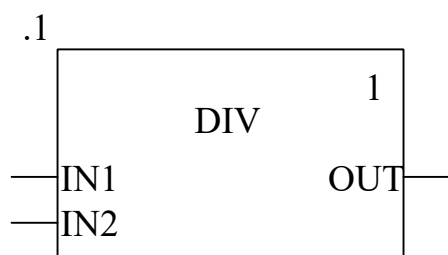


Рисунок 4 – Зображення елементарної функції

Число входів може бути збільшене 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 із входами і виходами. Входи завжди розташовуються зліва від рамки, а виходи – справа.

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 проекті.

Ім'я екземпляра створюється автоматично і має таку структуру:

FBI_n = Екземпляр Функціонального Блока,

де n – порядковий номер функціонального блока в проекті.

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

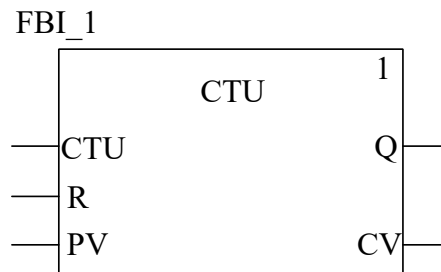


Рисунок 5 – Зображення елементарного функціонального блока

Похідні функціональні блоки (DFBs) мають такі самі характеристики, як 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.

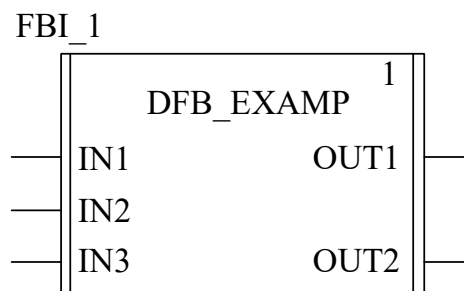


Рисунок 6 – Зображення похідного функціонального блока

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.

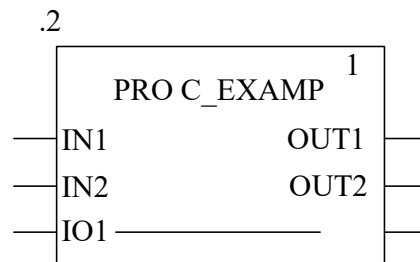


Рисунок 7 – Зображення процедури

Параметр

Входи (виходи) потрібні для передачі значень в (з) FFB, які є формальними параметрами.

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

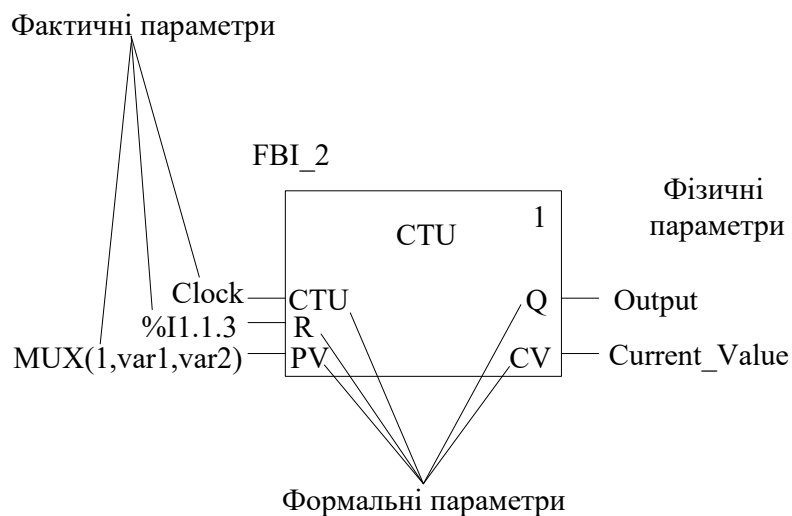


Рисунок 8 – Позначення параметрів

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 FFB, алгоритми не виконуються і ENO встановлюється в "0".

Якщо значення EN дорівнює "1", коли викликається FFB, 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 в "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 виключення – TOF

Функціональний блок використовується як затримка виключення.

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.

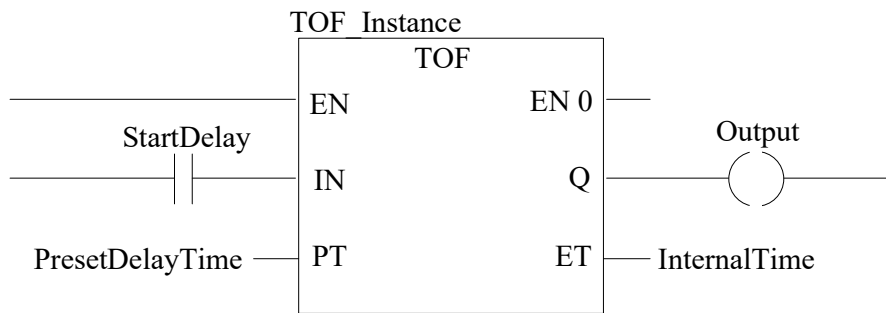


Рисунок 9 – Зображення в LD TOF

В таблиці 4 та 5 надано опис вхідних та 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 представлена часова діаграма затримки виключення TOF.

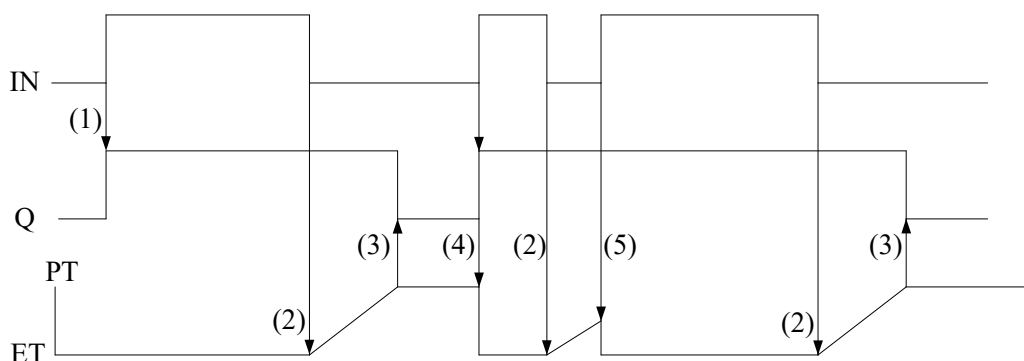


Рисунок 10 – Зображення часової діаграми TOF

Опис часової діаграми затримки виключення 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 відображає максимальну тривалість приблизно 49 днів.

Елементи часу, які авторизовані для представлення значення:

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

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

Затримка включення – TON

Функціональний блок може бути використаний як затримка включення.

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.

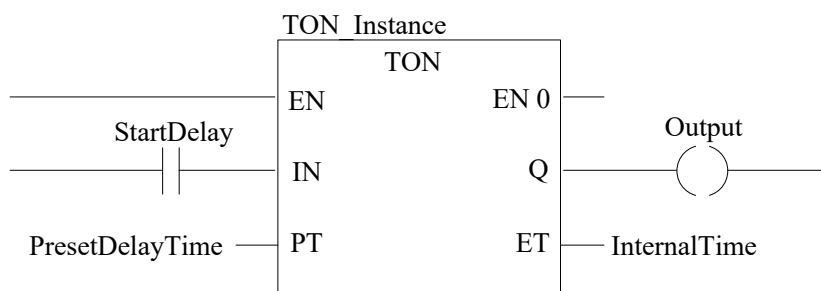


Рисунок 11 – Зображення в LD TON

В таблицях 6 та 7 надано опис вхідних та 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 виключення TON.

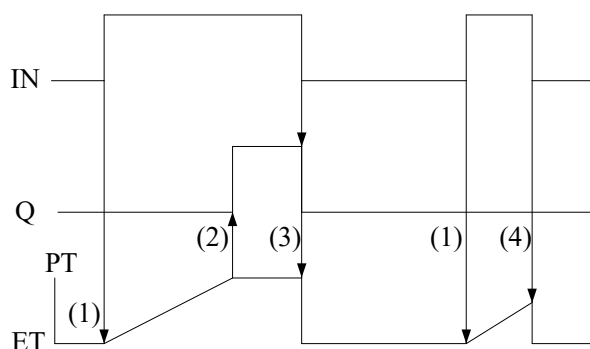


Рисунок 12 – Зображення часової діаграми затримки виключення TON

Опис часової діаграми затримки виключення 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 із 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

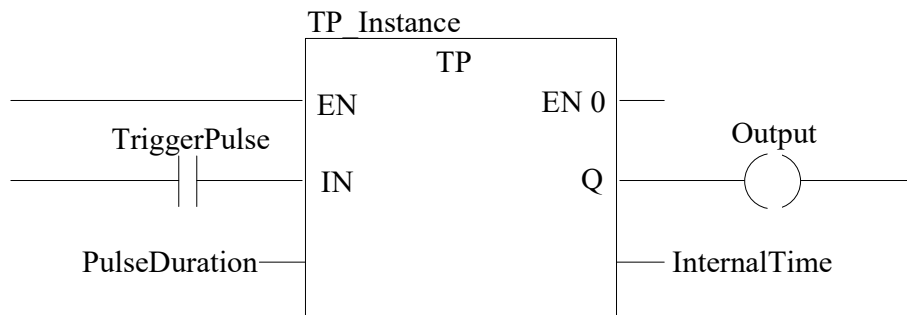


Рисунок 13 – Зображення в LD TP

В таблицях 8 та 9 надано опис вхідних та 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 подано зображення імпульсу TP.

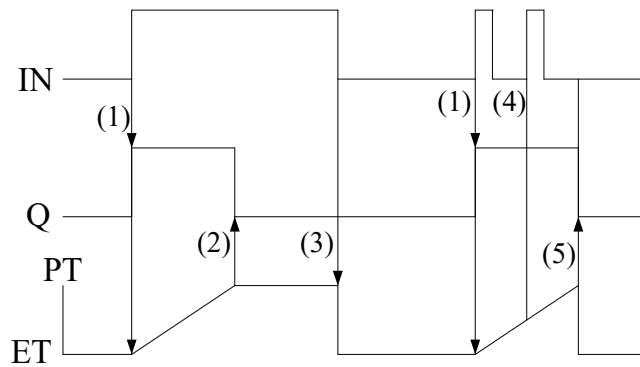


Рисунок 14 – Зображення імпульсу TP

Опис часової діаграми затримки виключення TP:

- (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) якщо 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 FFB є входом наступного FFB і вихід цього є входом першого).

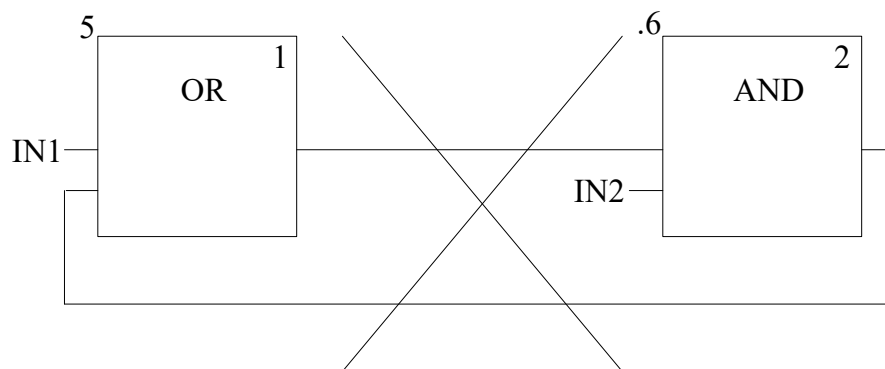


Рисунок 15 – Конфігурація петель не дозволена

3 Програма виконання лабораторної роботи

1 Ознайомитися з теоретичним матеріалом, що наведено у конспекті лекцій та навчальних посібниках і підручниках [1 – 5].

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

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

4 Конфігурування контролера.

4.1 В Project Browser натиснути лівою кнопкою «миші» 2 рази по рядку «Configuration».

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

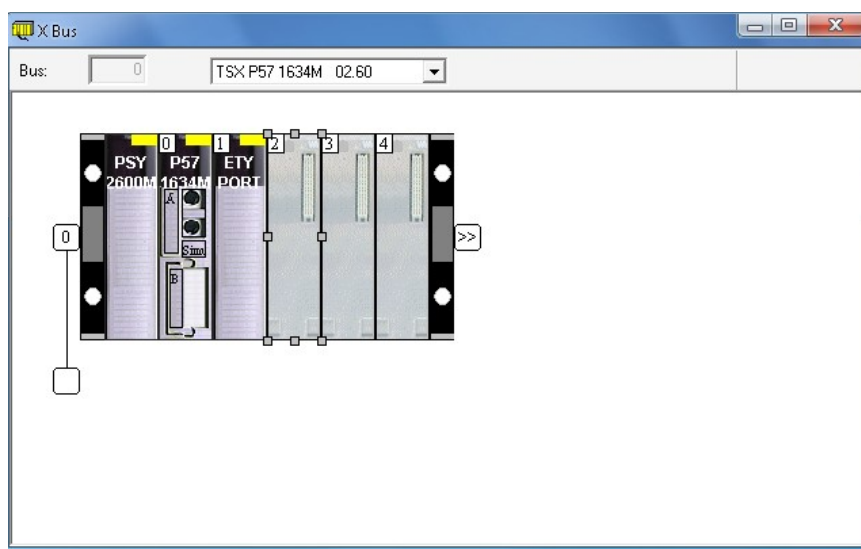


Рисунок 16 – Вибір модуля вводу «TSX DEY 16D2»

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

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

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

4.5 В 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.

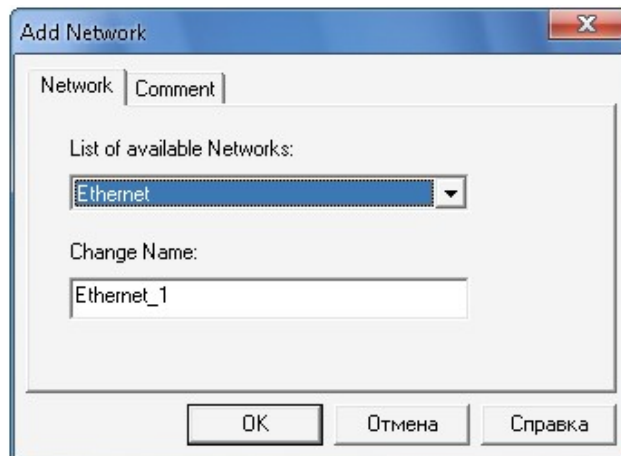


Рисунок 17 – Вікно підключення нової мережі

4.6 В Project Browser натиснути лівою кнопкою «миші» 2 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» та сконфігурувати його відповідно до рисунка 18.

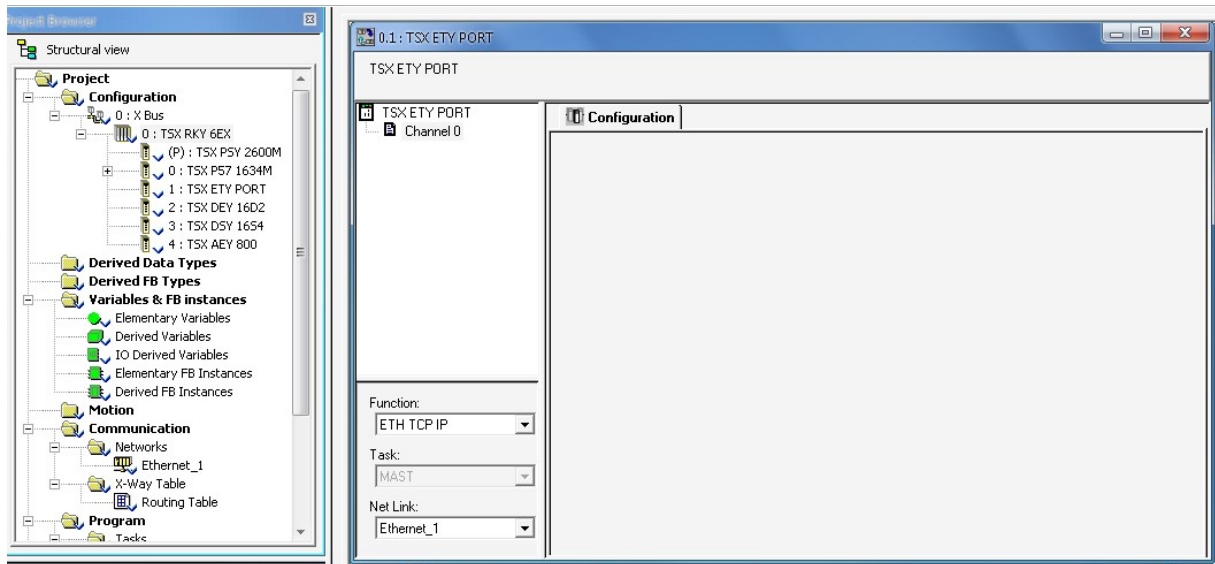


Рисунок 18 – Вікно конфігурації каналу зв'язку

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» та натиснути лівою кнопкою «миші» на пункті «LD». Встановити кількість стовпців у програмі – 25.

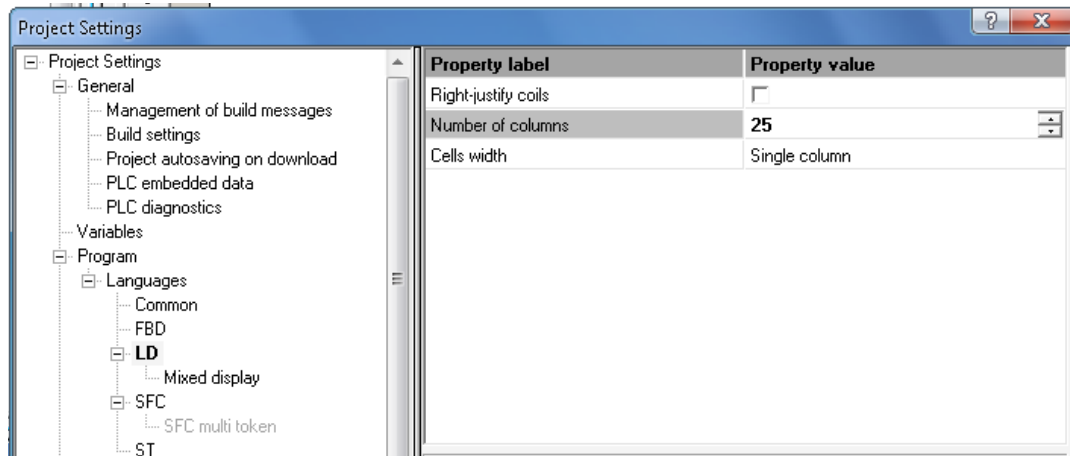


Рисунок 19 – Встановлення параметрів вікна програми

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

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

Натиснути кнопку «Ok».

5.3 В Project Browser в 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.

Таблиця 10

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

Відні дані подані в [1 – 3].

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

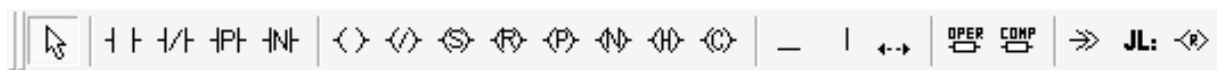


Рисунок 20 – Панель інструментів

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);

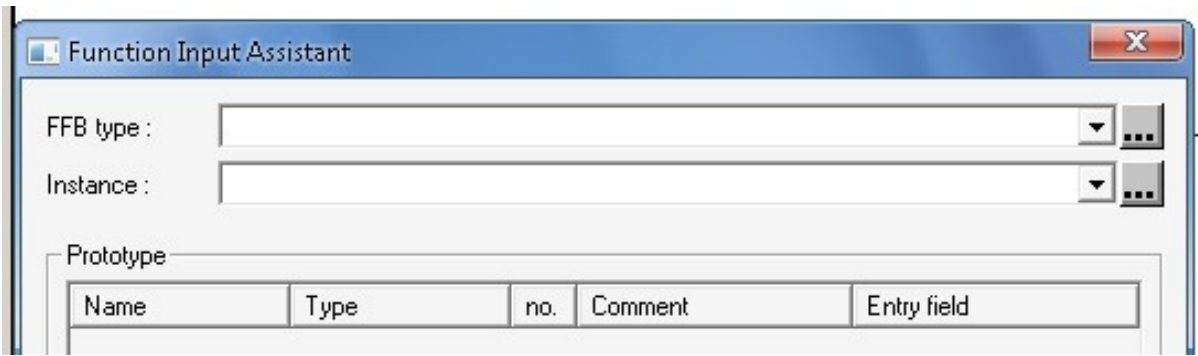


Рисунок 21 – Вікно додавання FFB

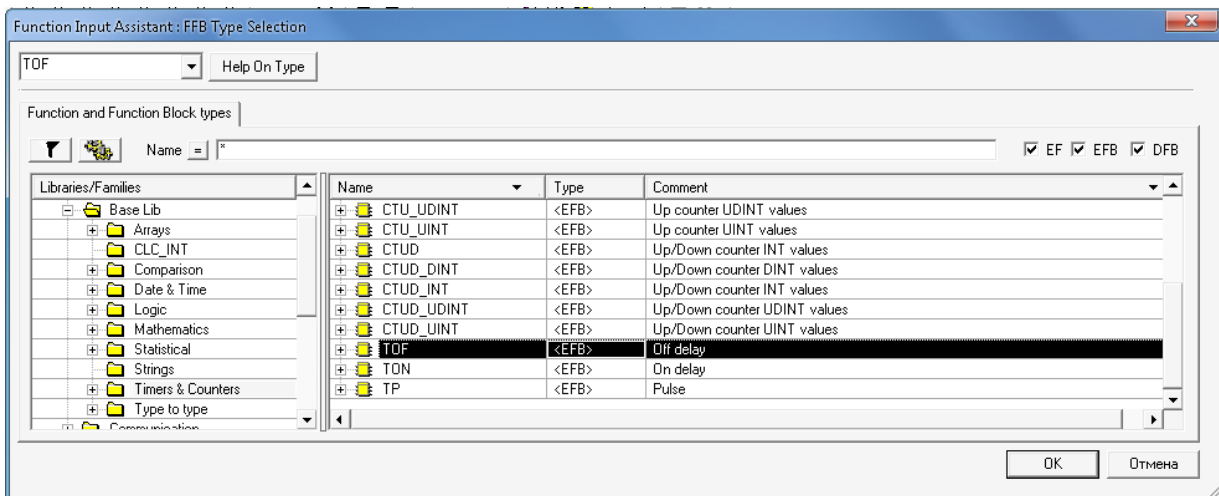


Рисунок 22 – Вікно вибору FFB

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

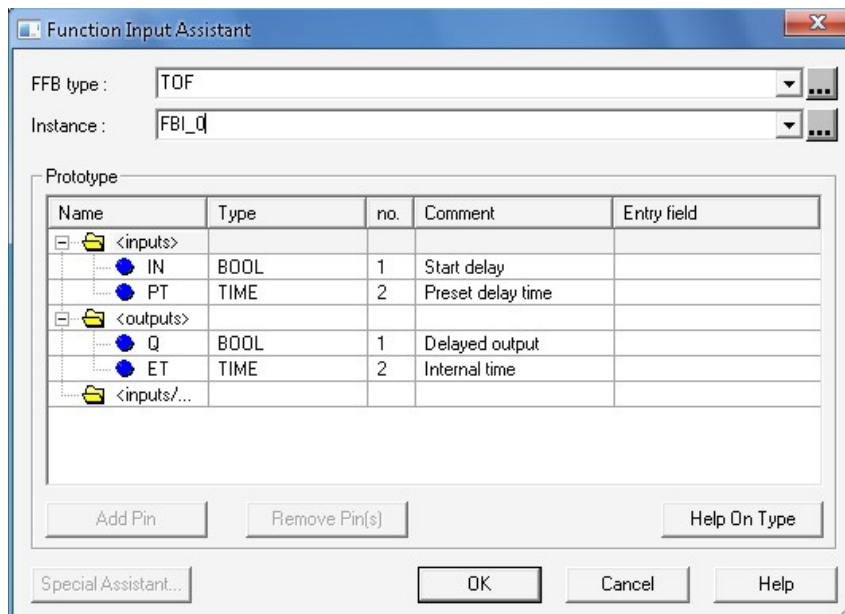


Рисунок 23 – Зображення вікна для введення параметрів

6 Запуск програми на віртуальному пристрої та моделювання.

6.1 Перед запуском програми необхідно:

– створити проект: 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).

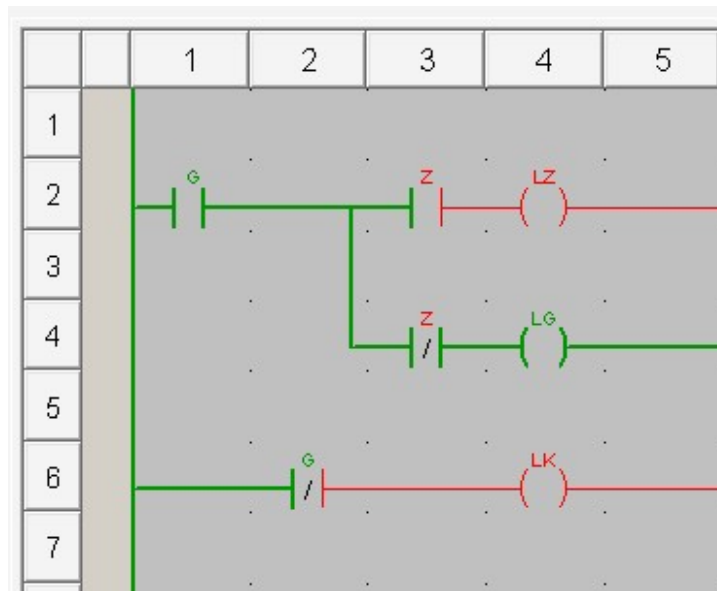


Рисунок 24 – Зображення частини програми

Для зміни стану контактів, що не мають обмотки, або кнопок необхідно натиснути правою кнопкою «миші» на контакт, що не має обмотки, або кнопці, обрати пункт «Set Value» та 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 в підрозділі «Operator Screens» натиснути праву кнопку «миші» та обрати пункт «New Screen». У вікні, що з'явилося, натиснути кнопку «Ok».

В 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 додавання функцій анімації стану необхідно:

- двічі натиснути лівою кнопкою «миші» на обраному елементі;
- у вікні, що з'явилося, перейти на вкладинку «Animation»;

– у рядку «Variable» натиснути кнопку «...» (рисунок 26);

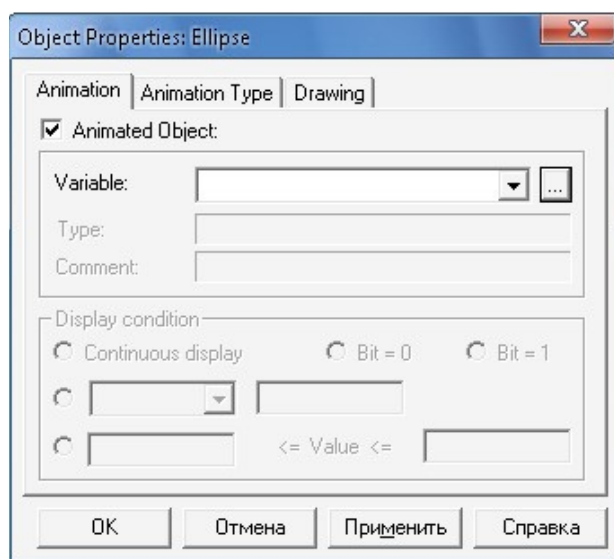


Рисунок 26 – Вікно додавання функцій анімації до об'єкта

– у вікні, що з'явилося, обрати елемент, стан якого необхідно візуалізувати;

– в пункті «Display condition» обрати стан елемента («Bit=0» або «Bit=1»), при якому він буде відображатися.

Створення елементів керування:

– після розміщення елементів управління двічі натиснути лівою кнопкою «миші» на одному з елементів;

– у вікні, що з'явилося, перейти на вкладинку «Control»;

– у рядку «Variable» натиснути кнопку «...»;

– у вікні, що з'явилося, обрати елемент, станом якого необхідно керувати.

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 пункту 4.

8 Письмово у звіті відповісти на питання, що наведені в пункті 5.

9 Отримати допуск до відпрацьовування лабораторної роботи.

10 Закінчити оформлення звіту відповідно до пункту 4.

4 Зміст звіту

- 1 Назва і мета роботи.
- 2 Письмові відповіді на контрольні питання.
- 3 Принципова схема згідно із завданням.
- 4 Програма, що написана мовою LD.
- 5 Графічне відображення моделювання роботи програми.
- 6 Короткі висновки з роботи.

5 Контрольні питання

- 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 Станционные системы автоматики и телемеханики / Под ред. Вл.В. Сапожникова. - М.: Транспорт, 1997.
- 2 Казаков А.А. Станционные устройства автоматики и телемеханики. -М.: Транспорт, 1990.
- 3 Альбоми типових схемних рішень ЕЦ.
- 4 Сайт підтримки Schneider Electric [Електронний ресурс]. – Режим доступу: <http://www.schneider-electric.com>
- 5 Засоби автоматизації. Електронна бібліотека документації. Schneider Electric. – 2005.