



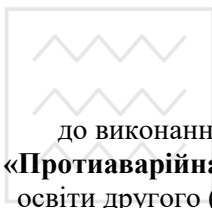
Національний університет
водного господарства та
природокористування

Міністерство освіти і науки України

Національний університет водного господарства та
природокористування

Кафедра автоматизації, електротехнічних та комп'ютерно-
інтегрованих технологій

04-03-241



Національний університет
водного господарства

Методичні вказівки

до виконання практичних робіт з навчальної дисципліни
«Противарійна автоматика. Частина 1» для здобувачів вищої
освіти другого (магістерського) рівня за освітньо-професійною
програмою «Електроенергетика, електротехніка та електромеханіка»
спеціальності 141 «Електроенергетика, електротехніка та
електромеханіка» денної та заочної форм навчання

Рекомендовано науково-методичною
радою з якості ННІ АКОТ
Протокол № 6 від 21.02.2020 р.

Рівне – 2020



Методичні вказівки до виконання практичних робіт з навчальної дисципліни «Протиаварійна автоматика. Частина 1» для здобувачів вищої освіти другого (магістерського) рівня за освітньо-професійною програмою «Електроенергетика, електротехніка та електромеханіка» спеціальності 141 «Електроенергетика, електротехніка та електромеханіка» денної та заочної форм навчання [Електронне видання] / Стеценко А. М. – Рівне : НУВГП, 2020. – 50 с.

Укладач: Стеценко А. М., к.т.н., доцент кафедри автоматизації, електротехнічних та комп'ютерно-інтегрованих технологій.

Відповідальний за випуск: Древецький В. В., д.т.н., професор, завідувач кафедри автоматизації, електротехнічних та комп'ютерно-інтегрованих технологій.



Керівник групи забезпечення спеціальності 141 «Електроенергетика, електротехніка та електромеханіка»: Василець С. В., д.т.н., професор, професор кафедри автоматизації, електротехнічних та комп'ютерно-інтегрованих технологій.



Зміст

Робота №1. Програмований логічний контролер (ПЛК) Zelio Logic як засіб керування та протиаварійної автоматики. Розробка і випробування системи контролю та керування типу «Розумний дім».

Робота №2. Розробка і випробування системи контролю та керування на базі ПЛК Zelio Logic мовами LD та FBD.





Робота №1. Програмований логічний контролер (ПЛК) Zelio Logic як засіб керування та протиаварійної автоматики. Розробка і випробування системи контролю та керування типу «Розумний дім».

1. Мета роботи

Навчитися розробляти системи контролю та керування типу «Розумний дім» на базі програмованого логічного контролера.

2. Теоретичні відомості

2.1 Системи контролю та керування типу «Розумний дім»

Першим інтелектуальним будинком у світі став «Будинок трону» японського професора Кена Сакамури у Токіо, побудований наприкінці 1980-х років. Давачі погоди відкривали вікна, коли дув свіжий бриз, і вмикали кондиціонер, коли ставало жарко; якщо радіо грало надміру голосно, вікна автоматично закривалися, щоб не турбувати сусідів; якщо дзвонив телефон, комп'ютер знижував звук аудіо системи, і так далі.

Основою «розумного» керування інженерними системами і системами життєзабезпечення являється технологія, що дозволяє здійснювати інтелектуальне керування, регулювання та гнучку оптимізацію систем:

- керування опаленням: керування системами опалення різних типів, економія витрат на опалення;
- водопостачання: контроль протікання води у всіх приміщеннях, керування системами водо підготовки;
- газопостачання: контроль витікання газу;
- охорона: контроль проникнення у приміщення, периметральний контроль, імітація присутності людей, імітація присутності тварин, обмеження доступу, відеонагляд;
- пожежна безпека: контроль займання;
- керування електроенергією: контроль електромереж, економія затрат;
- керування мікрокліматом у приміщенні: керування температурою, відносною вологістю повітря, контроль вмісту небезпечних газів;
- оповіщення: голосове оповіщення, дзвінок на мобільні та міські номери, sms, Internet;



освітлення: зв'язок із присутністю людей, керування за часом, сценарне керування, сценарії зовнішнього освітлення;

- будильник: голос, телефон, мелодії;
- голосовий пейджер: запис та відтворення повідомлень;
- басейн: керування нагрівом, керування фільтрацією;
- полив: автоматичний полив;
- жалюзі: автоматичне відкриття-закриття жалюзі та воріт, сценарне керування жалюзі;
- керування побутовими пристроями: керування усіма пристроями з одного пульта дистанційного керування, зв'язок між приміщеннями, вхідний дзвінок з широким вибором мелодій.

2.2 Технічні засоби для реалізації системи «Розумний будинок»

Для реалізації системи «Розумний дім» використовуються різноманітні технічні засоби: датчики, контролери, пульти та панелі керування, домофони, GSM модеми, виконавчі механізми і регулюючі органи. Для реалізації функцій керування використовуються прості за набором функцій програмовані контролери із контролерного ряду тієї чи іншої фірми-виробника: Zelio Logic (Schneider Electric), Logo (Siemens), Mitsubishi (Mitsubishi Electric), Forth Logic (Електросвіт) та інші. Розглянемо детальніше програмований контролер Zelio Logic, на базі якого і будемо реалізовувати систему керування адміністративною будівлею.

2.2.1 Будова контролера Zelio Logic2

Загальний вигляд контролера Zelio Logic2 представлено на рис. 1.1.



Рис. 1.1 Загальний вигляд контролера Zelio Logic2

Передня панель

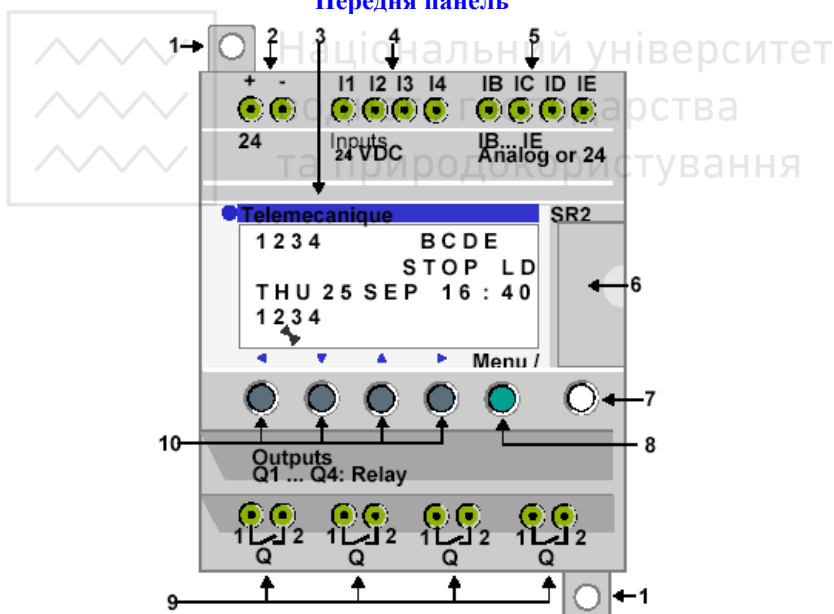


Рис. 1.2 Вигляд передньої панелі контролера Zelio Logic2



Таблиця 1.1

Позиція	Опис
1	Висувна монтажна ніжка
2	Гвинтові клеми для підключення живлення
3	Рідкокристалічний дисплей (4 рядки, 18 символів у кожному)
4	Гвинтові клеми для дискретних входів
5	Гвинтові клеми для аналогових входів (0...10 В), які можуть використовуватися як дискретні у деяких моделях
6	Роз'єм для резервної пам'яті або кабеля під'єднання ПК
7	Клавіша Shift
8	Клавіша вибору і підтвердження
9	Гвинтові клеми дискретних виходів
10	Клавіші-стрілки (або сконфігуровані Z-клавіші)

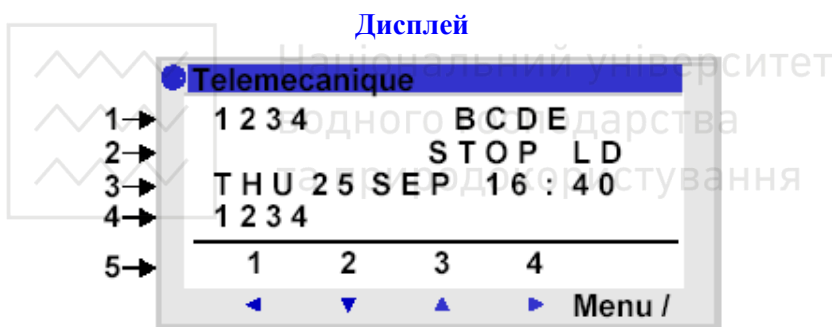


Рис. 1.3 Дисплей контролера Zelio Logic2

Таблиця 1.2

Позиція	Опис
1	Відображення стану входів (В...Е відображають аналогові входи)
2	Відображення режиму роботи (RUN / STOP) та режиму програмування (LD / FBD)
3	Відображення дати (число і час для пристроїв, що підтримують таку можливість)
4	Відображення стану виходів
5	Контекстне меню / кнопки швидкого доступу / іконки, які відображають режим роботи



Підключення контролера постійного струму

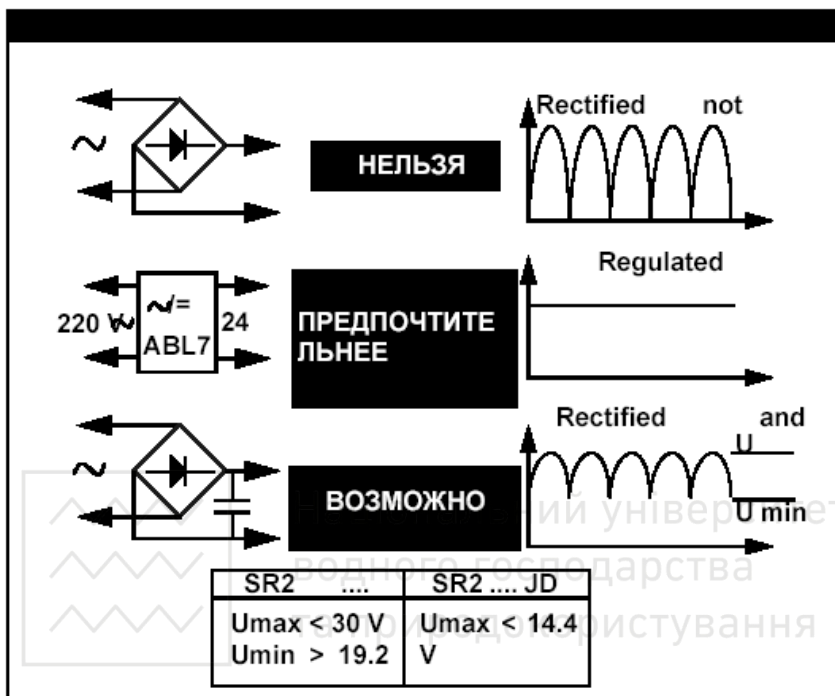


Рис. 1.4 Правила підключення контролера постійного струму

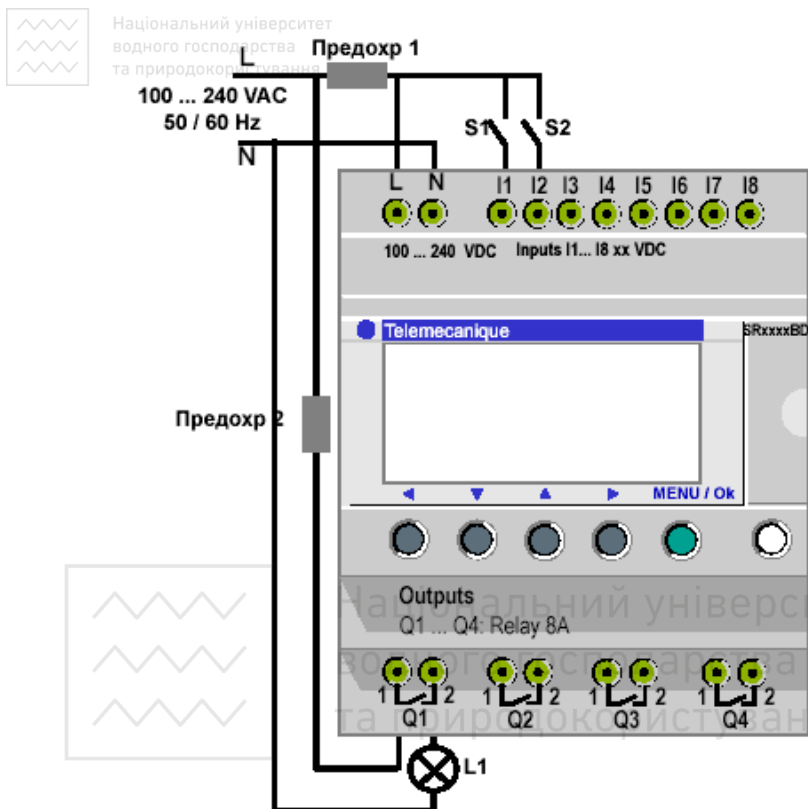


Рис. 1.5 Приклад підключення 2-х вимикачів до входів I1 та I2 та лампочки до виходу Q1 контролера

2.2.3 Меню

Головне меню

Таблиця 1.3

№	Назва	Призначення
1	Programming	Введення драбинкових діаграм.
2	Parameters	Введення та зміна параметрів елементів програми.
3	Monitoring	Доступний тільки в режимі LD / RUN. Даний режим використовується для динамічного відображення стану входів-виходів контролера. Користувач може динамічно змінювати значення параметрів функцій, якщо вони розблоковані (клавіші Shift + Param).

4	RUN / STOP	Запуск чи зупинка програми. Коли здійснюється перемикання від STOP до RUN, програма ініціалізується.
5	Configuration	Меню конфігурації.
6	Clear Program	Видалення програми.
7	Transfer	Передача програми у модуль резервної пам'яті або навпаки.
8	Language Menu	Меню вибору мови.
9	Version	Визначення версії системних компонентів: вбудованого програмного забезпечення, FBD-функції, LD-функції.
10	Fault	Меню помилок, яке дозволяє відображати номери помилок або попереджень, виявлених програмним забезпеченням контролера.

Меню конфігурації – Configuration

Таблиця 1.4

№	Назва	Призначення
1	Password	Меню введення пароля.
2	Filter	Вибір швидкості визначення зміни стану всіх дискретних входів.
3	Zx-Keys	Дозволяє користувачу дозволити або заборонити використання клавіш курсору у якості кнопок. Доступно тільки в режимі LD.
4	Change D/T	Задання дати та часу для модулів з годинником.
5	Change Summ/Wint	Автоматична зміна часового діапазону літо-зима для модуля з годинником.
6	Watchdog Cycle	Задання тривалості циклу виконання програми.

2.2.4 Програмування контролера Zelio Logic2 мовами драбинкових діаграм (LD) та FBD-блоків

Контролер Zelio Logic2 можна програмувати за допомогою командних клавіш (мова LD) та за допомогою спеціалізованого програмного забезпечення Zelio Soft2 (мови LD та FBD).

Розглянемо правила програмування контролера Zelio Logic2 за допомогою спеціалізованого програмного забезпечення Zelio Soft2.



2.2.4.1 Створення нової програми

Для того, щоб створити нову програму необхідно виконати наступні дії.

1. Вибрати меню **File/New** або натиснути піктограму **Create a new program** під час запуску програми Zelio Soft2. В результаті з'явиться вікно вибору типу модуля.

2. Виберіть потрібний модуль за допомогою ЛК миші. Модулі згруповані за такими ознаками:

- кількість входів-виходів;
- наявність або відсутність дисплею;
- можливість під'єднання шасі розширення.

В результаті здійснення вибору типу модуля з'являється весь список модулів даного типу у вигляді таблиці.

3. Виберіть модуль подвійним натисканням миші або натисніть клавішу **Далее (Next)**. На цьому етапі може виникнути такі ситуації:

- модуль не підтримує можливості розширення та програмується лише мовою LD – перейдіть до кроку 7;
- модуль не підтримує можливості розширення і програмується мовами LD та FBD – перейдіть до кроку 6;
- модуль підтримує розширення, тоді на екрані з'явиться інформація про тип модуля, вибраний у попередньому пункті та типи можливих розширень. Натиснувши кнопку **Add**, можна вибрати необхідне розширення. Для переходу до наступного кроку натисніть кнопку **Далее**. Модуль розширення при необхідності можна видалити кнопкою **Delete**.

4. Виберіть необхідний модуль розширення.

5. Підтвердіть вибір (кнопка **Далее**). В результаті з'явиться вікно вибору мови програмування.

6. Виберіть потрібну мову (LD або FBD). Натисніть кнопку **Далее (Next)**.

7. З'явиться вікно редагування програми.

2.2.4.2 Мова драбинкових діаграм (Ladder Diagram Language)

Загальні положення

Програма, написана мовою LD, складається з мереж з'єднаних між собою елементів. Виконання програми відбувається згідно географічного розміщення елементів зверху донизу і зліва направо.

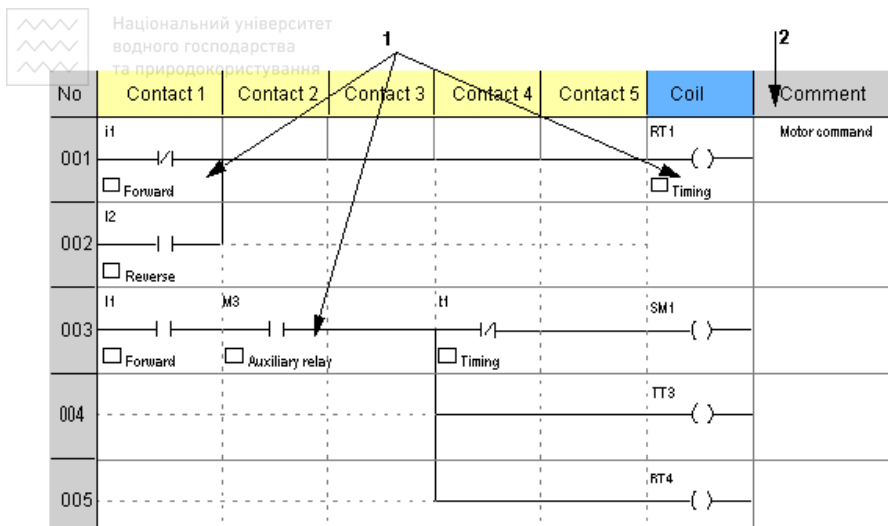


Рис. 1.6 Приклад програми, написаної мовою LD

Таблиця 1.5

№	Елемент	Функція
1	Графічні елементи	<ul style="list-style-type: none"> - Входи-виходи контролера (кнопки, давачі, реле, світлові індикатори тощо). - Контакти функціональних блоків (таймерів, лічильників тощо). - Логічні операції. - Внутрішні змінні (допоміжні реле) контролера.
2	Коментар	Для кожної лінії драбинкової діаграми або конкретного графічного елемента (є необов'язковим).

Вікно створення програми поділяється на декілька частин (рис. 1.7).

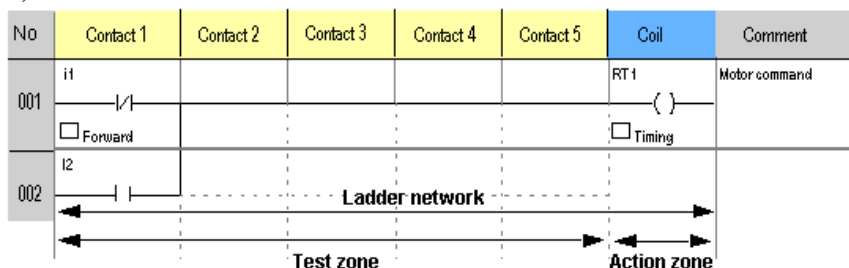


Рис. 1.7 Структура вікна створення програми мовою LD



Кожен рядок LD-програми складається з набору графічних елементів, які розміщені у прямокутниках сітки:

- максимальна кількість рядків – 120;
- кожен рядок може містити до 5 контактів і 1 котушку.

LD-діаграма поділена на 2 зони:

- **test zone**, в якій розміщуються контакти;
- **action zone**, в якій розміщуються котушки (відображає результат операцій).

Графічні елементи у мові LD

1. Контакти

Дані елементи розміщуються у test zone і займають 1 чарунку.

Таблиця 1.6

Назва	Графічне представ- лення (LD)	Графічне представ- лення (Electrical)	Функції
Нормальний відкритий контакт			Замикається, коли асоційований з ним елемент (вимикач, давач), активний.
Нормальний замкнутий контакт			Розмикається, коли асоційований з ним елемент (вимикач, давач), активний.

2. Елементи зв'язку

Використовуються для зв'язку між елементами, що розміщені у test-зоні або action-зоні.

Таблиця 1.7

Назва	Графічне представ- лення	Функції
Горизонтальний зв'язок		Використовується для зв'язку між послідовно розміщеними елементами.

Вертикальний зв'язок	Використовується для зв'язку між паралельно розміщеними елементами.
----------------------	---

3. Котушки

Таблиця 1.8

Назва	Графічне представлення (LD)	Графічне представлення (Electrical)	Функції
Пряма котушка			На котушку подається живлення, якщо контакти, до яких вона під'єднана, активні.
Імпульсна котушка			На котушку подається живлення, якщо контакти, до яких вона під'єднана, змінюють свій стан.
Замкнена котушка (Set)			На котушку подається живлення, якщо контакти, до яких вона під'єднана, активні. Її стан не зміниться навіть якщо її контакти розімнуться.
Розімкнена котушка (Reset)			Котушку стає неактивною, якщо контакти, до яких вона під'єднана, активні. Вона зберігає свій стан незалежно від подальшої поведінки її контактів.

Увага! Котушки Set і Reset повинні використовуватися разом. Функція Reset має пріоритет перед функцією Set. Кожен з виходів Q повинен використовуватися у програмі тільки 1 раз, крім котушок Set і Reset.

Етапи створення LD-програми

1. Складіть список входів-виходів і присвойте їм коментарі.



2. Складіть список необхідних керуючих функцій (підрахунок кількості машин, дискретне керування).
3. Реалізуйте кожну функцію за такою схемою.

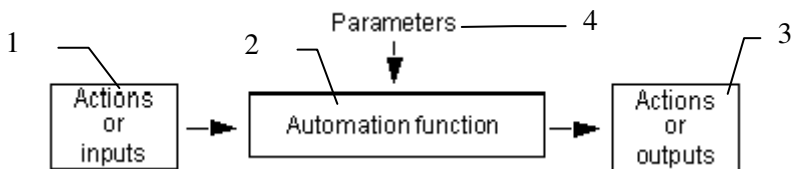


Рис. 1.8 Схема реалізації керуючої функції

- 1 – керуючі дії або входи;
- 2 – керуюча функція;
- 3 – керуючі дії або виходи;
- 4 – параметри.
4. Присвойте коментар кожній функції.
5. Протестуйте кожну функцію, використовуючи режим емуляції.

Режими програмування

1. Режим Zelio (Zelio Entry Mode)

Даний режим дозволяє створити програму мовою LD за допомогою емуляції використання командних кнопок на передній панелі контролера.

Приклад:

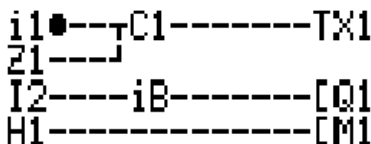


Рис. 1.9 Приклад LD-програми у режимі Zelio

Даний режим зручний тим, що надає програмісту можливість навчитися програмувати контролер через командні кнопки у середовищі Zelio Soft, не маючи самого контролера.

2. Вільний режим (Free Entry Mode)

Даний режим дозволяє засвоїти прийоми створення програм за допомогою самого середовища програмування без імітації передньої панелі контролера, а саме:

- використання панелей інструментів;



створення програм за допомогою перетягування елементів у потрібне місце драбинкової діаграми;

- використання вікон параметрів елементів;
- можливість переглядати усю програму у цілому.

Приклад:

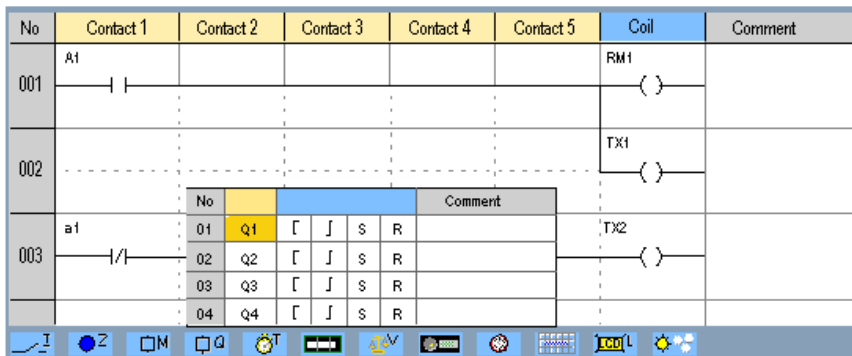


Рис. 1.10 Приклад LD-програми у вільному режимі

Даний режим надає ще ряд додаткових переваг:

- вибір типу представлення символів (LD, електричні аналоги);
- можливість додати коментар до рядка.

3. Параметричний режим (Parameter Mode)

Даний режим дозволяє представити усі функції керування, які використовуються у програмі, з їх параметрами у вигляді списку, який має такі позиції:

- номер за порядком;
- функція: таймер, лічильник тощо;
- мітка, що визначає функціональний блок;
- тип;
- уставка;
- вказівник замикання;
- коментар.

Приклад:

Національний університет
водного господарства
та природокористування

Zelio entry o r o c i e n a r Free entry

Parametering Text entry

No	Function	Label	Type	Preset	Lock	Comment
001	Counter	C1		C1 = 00001	No	Number of vehicles
002	Clock	⌚ 1			No	Opening time
003	Analog	A1	5: 7.0 <= 1B	R = 7.0V	No	Primary circuit, voltage
004	Text Block	X1			Yes	Current counter value

Рис. 1.11 Список функціональних блоків, використаних у LD-програмі, при використанні параметричного режиму

Параметри конкретного функціонального блоку можна змінити, якщо двічі натиснути мишкою на рядку, де він описаний.

4. Текстовий режим (Text Entry Mode)

Даний режим дозволяє представити усі входи-виходи, які використані у програмі, у вигляді списку. Важливо присвоїти кожному входу та виходу зрозумілий коментар для того, щоб зробити програму зрозумілішою.

Приклад:

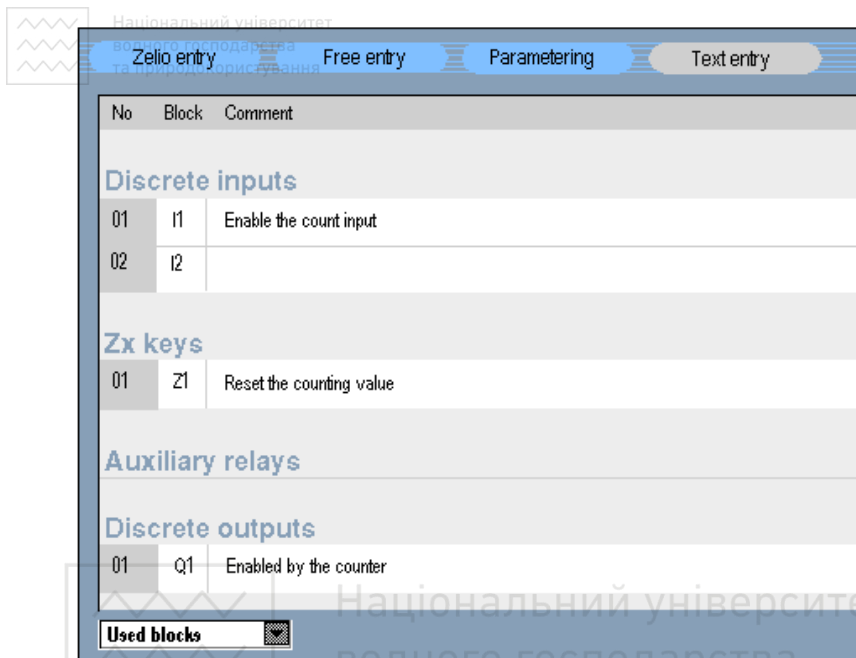


Рис. 1.12 Представлення входів та виходів у текстовому режимі

Введені коментарі будуть відображатися під графічними зображеннями елементів у драбинковій діаграмі (рис. 1.7).


Елементи мови драбинкових діаграм (LD)

Таблиця 1.9

№	Піктограма у Zelio Soft	Назва
1		Дискретні входи (Discrete Inputs)
2		Дискретні виходи (Discrete Outputs)
3		Входи-виходи Modbus (Modbus Inputs-Outputs)
4		Допоміжні реле Auxiliary Relays)
5		Z-клавіши (Zx Keys)
6		Лічильники (Counters)

7		Функціональний блок порівняння лічильника (Counter Comparator)
8		Швидкий лічильник (Fast Counter)
9		Годинники (Clocks)
10		Функціональний блок переходу на зимовий та літній час (Change to Summer/Winter Time)
11		Таймери (Timers)
12		Аналогові компаратори (Analog Comparators)
13		Функціональний блок Текст (Texts)
14		Функціональний блок підсвітки екрану (LCD Screen Backlighting)

Автоматична перевірка LD-програми

Середовище програмування Zelio Soft2 дозволяє здійснювати емуляцію програм, їх покрокову перевірку та налагодження. Якщо знайдена якась помилка, піктограма , що розташована у верхній частині вікна редагування програми, змінює свій колір з блакитного на червоний.

Приклад:

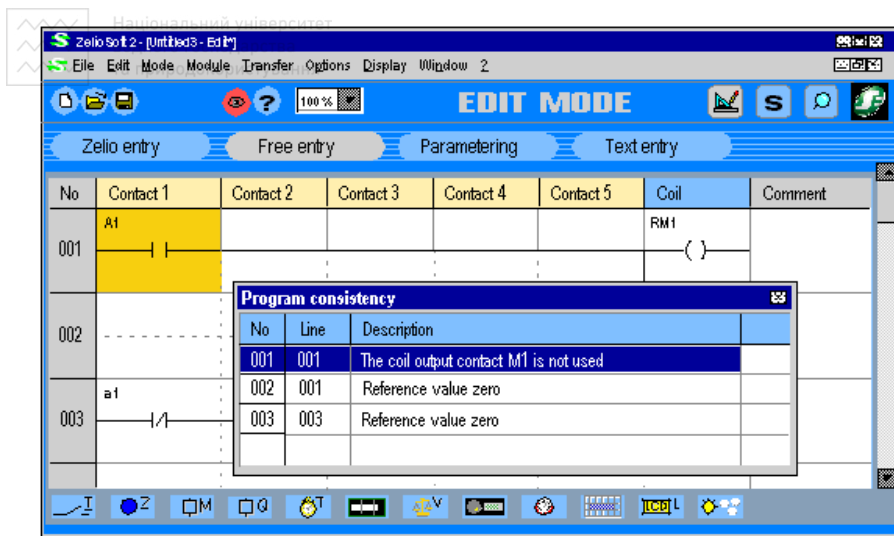


Рис. 1.13 LD-програма у режимі налагодження

Натиснувши на піктограму “Око”, можна відкрити діалогове вікно, яке містить наступну інформацію:

- номер помилки;
- місцезнаходження помилки: рядок або стовпчик;
- опис помилки.

Подвійне натискання мишки на назві помилки призводить до підсвітки конкретного елемента програми, де міститься помилка.

Увага! Навіть якщо символ “Око” став червоним, програма може бути запущена на емуляцію. Елементи, розміщені у програмі з допущенням помилок (відсутнє з’єднання), будуть неактивні.

Емуляція та моніторинг LD-програми

Функція емуляції використовується для виконання програми безпосередньо в Zelio Soft2 (у реальному часі).

Приклад:

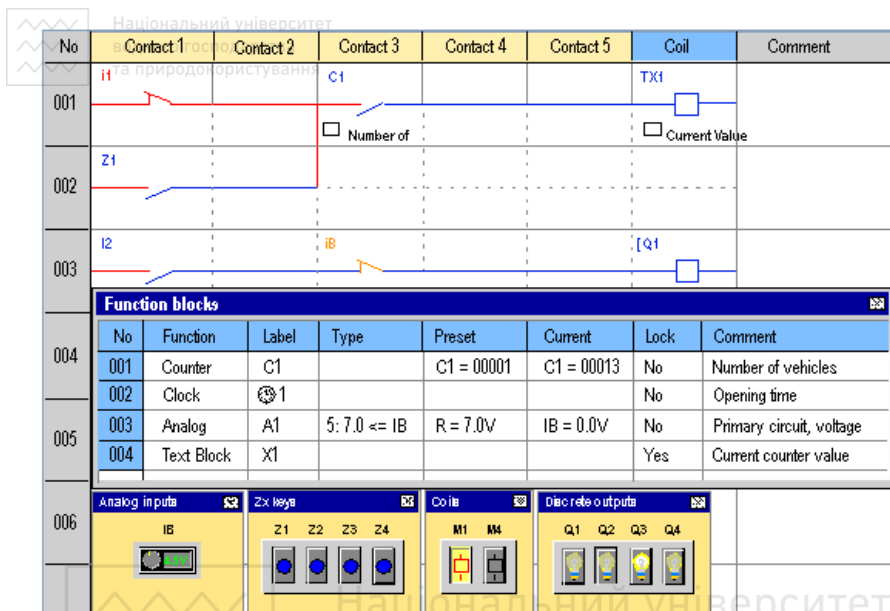


Рис. 1.14 LD-програма у режимі емуляції

Таблиця 1.10

Піктограма	Функція
	Перехід у режим емуляції
	Перехід у режим редагування
	Перехід у режим моніторингу
	Запуск програми на виконання
	Зупинка виконання програми
	Пуза/Запуск: зупиняє або перезапускає програму (активна тільки у режимі Run)
	Емуляція збою енергопостачання (активна тільки у режимі Run)

Функція моніторингу використовується для виконання програми в контролері (у режимі реального часу) і одночасної візуалізації в Zelio Soft 2. Потрібно, щоб на комп'ютері і в контролері була запущена та сама програма. У даному режимі користувач може:



- відображати значення різних параметрів у вікні функціональних блоків;
- натиснути на блок, щоб змінити його параметри.

3. Програма роботи

1. Вивчити будову та характеристики контролера Zelio Logic 2.
2. Дослідити правила підключення контролера Zelio Logic 2.
3. Навчитися користуватися меню контролера Zelio Logic 2
4. Навчитися програмувати контролер Zelio Logic 2 мовою драбинкових діаграм (LD) за допомогою командних клавiш передньої панелі.
5. Навчитися програмувати контролер Zelio Logic 2 мовою драбинкових діаграм (LD) у середовищі Zelio Soft 2.
6. Навчитися записувати програми у контролер.
7. Виконати завдання, поставлені викладачем.

4. Опис лабораторного обладнання

1. Персональний комп'ютер.
2. Операційна система Windows.
3. Лабораторний стенд з логічним контролером Zelio Logic.
4. Середовища програмування Zelio Soft, Zelio Soft 2.

5. Порядок виконання роботи

1. Напишіть програму для програмованого логічного контролера (ПЛК) Zelio Logic мовою драбинкових діаграм (LD).
2. Проведіть емуляцію та налагодження програми у середовищі програмування Zelio Soft. Для цього скористайтеся меню Simulation та натисніть кнопку Run. Після проведення перевірки роботи програми натисніть кнопку Stop.
3. Запишіть створену програму у ПЛК та дослідіть її роботу. Для цього необхідно виконати наступні дії:
 - подати живлення на стенд;
 - перевести контролер у режим STOP;
 - увійти у головне меню, натиснувши клавішу Sel./OK;
 - вибрати пункт меню Transfer.\ PC -> Module;
 - у середовищі програмування Zelio Soft вибрати пункт меню Transfer\ Transfer Program\ PC -> Module.
4. Навчитися керувати запуском та зупинкою програми з клавіатури контролера та з комп'ютера. Для керування



виконанням програми з клавіатури контролера скористайтеся пунктом меню Run\ Stop, з комп'ютера – пунктами меню середовища Zelio Soft Transfer\ Run Module, Transfer\ Stop Module.

5. Для відслідковування роботи програми на комп'ютері виберіть пункт меню Mode\ On Line Monitoring Mode.
6. Складіть пояснюючу таблицю використаних входів і виходів ПЛК, в якій зазначте назву входу-виходу, тип, призначення.

Завдання 1.1

Освітлення вітрини

За допомогою ПЛК повинні автоматично освітлюватися товари, що виставлені у вітрині. При цьому розрізняють 4 групи джерел світла. Одна для освітлення вдень, одна для додаткового освітлення ввечері, одна для мінімального освітлення вночі і одна для створення світових плям, які мають окремо освітлювати розміщені предмети.

Вітрина повинна освітлюватися з понеділка по п'ятницю з 8:00 до 22:00, в суботу з 8:00 до 24:00 і у неділю з 12:00 до 20:00. Протягом цих інтервалів через реле часу включається група джерел освітлення №1 на Q1. Крім того, по вечорах підключається група джерел освітлення 2, коли спрацьовує сутінкове реле на вході I1. Поза вище названих інтервалів часу група джерел освітлення 3 на Q3 бере на себе мінімальне освітлення після деблокування сутінкового реле. Через сигналізатор переміщень на I4 протягом усього часу включаються або виключаються світлові плями (група джерел освітлення 4 на Q4). За допомогою тестової кнопки на I3 можна на 1 хвилину включити усі групи джерел освітлення, щоб, наприклад, перевірити їх функціонування або їх виправити.

Компоненти, які використовуються:

Модуль - SR2-B121BD

- I1 – сутінкове реле (NO – normally opened);
- I2 – вимикач (NO – normally opened);
- I3 – тестовий вимикач (NO – normally opened);
- I4 – сигналізатор переміщень (NO – normally opened);
- Q1 – група джерел освітлення 1;
- Q2 – група джерел освітлення 2;
- Q3 – група джерел освітлення 3 (мінімальне освітлення);

**Переваги і особливості:**

Встановлені інтервали часу можуть бути у будь-який час довільно змінені. Простота вибору інших комбінацій джерел освітлення. Необхідно менше компонентів, ніж при традиційному рішенні.

Завдання 1.2**Керування жалюзіями**

За допомогою ПЛК необхідно керувати жалюзіями житлового будинку. За допомогою селекторного перемикача може бути вибраний ручний режим або автоматичне керування. В залежності від часу, темноти і дня тижня жалюзі автоматично закриваються або відкриваються.

Ручний режим:

За допомогою вимикачів I2 (відкрити) і I3 (закрити) жалюзі можна відкривати і закривати вручну. Передпосилкою для цього являється те, що селективний перемикач на I6 не знаходиться в положенні «Автоматика».

Автоматичний режим:

Для переходу в автоматичний режим селекторний перемикач I6 повинен знаходитися у положенні «Автоматика». Коли спрацьовує сутінкове реле на I1, жалюзі закриваються на період з 18:00 до 7:00. Відкриваються вони протягом дня між 7:00 і 18:00. Через кінцеві вимикачі на I4 та I5 здійснюється опитування, відкриті жалюзі чи закриті.

Компоненти, які використовуються:

Модуль - SR2-B121BD

- I1 – сутінкове реле (NO – normally opened);
- I2 – вимикач для відкриття у ручному режимі (NO – normally opened);
- I3 – вимикач для закриття в ручному режимі (NO – normally opened);
- I4 – кінцевий вимикач «Жалюзі відкриті» (NC – normally closed);
- I5 – кінцевий вимикач «Жалюзі закриті» (NC – normally closed);
- I6 – селекторний перемикач в положенні «Автоматика»;
- Q1 – відкрити жалюзі;



Переваги і особливості:

Часові інтервали можуть бути просто пристосовані до індивідуальних умов: наприклад, різний час у робочі дні, в кінці тижня або під час відпустки. Різне керування для двох діапазонів (ситуацій) через ще вільні виходи. Економія енергії завдяки поєднанню реле часу і сутінкового реле.

6. Контрольні запитання

1. Яким чином можна охарактеризувати системи контролю та керування типу «Розумний дім»?
2. Яка будова контролера Zelio Logic2?
3. Які основні характеристики контролера Zelio Logic2?
4. Які правила підключення Zelio Logic2?
5. Назвіть пункти та функції головного меню контролера Zelio Logic2.
6. Назвіть пункти та функції меню конфігурації контролера Zelio Logic2.
7. Яким чином запрограмувати контролер мовою драбинкових діаграм (Ladder Diagram Language)?
8. Назвіть основні графічні елементи мови драбинкових діаграм.
9. Яким чином записати програму у ПЛК з комп'ютера? Яким чином зчитати записану програму з ПЛК на комп'ютер?
10. Як можна відслідкувати роботу програми з комп'ютера у реальному часі?



Робота №2: Розробка і випробування системи контролю та керування на базі ПЛК Zelio Logic мовами LD та FBD.

1. Мета роботи

Навчитися розробляти системи контролю та керування на базі програмованого логічного контролера. Навчитися програмувати контролер Zelio Logic2 мовами драбинкових діаграм (LD - Ladder Diagram) та функціональних блоків (FBD – Function Block Diagram). Навчитися розробляти дворівневі системи контролю та керування на базі Zelio Logic2.

2. Теоретичні відомості

Програмований логічний контролер Zelio Logic для керування процесом

Розробимо систему керування процесом на базі програмованого контролера Zelio Logic фірми Schneider Electric. Будову, характеристики, правила підключення, меню програмованого контролера Zelio Logic та правила написання програм мовою драбинкових діаграм розглянуто у роботі 1. Тому розглянемо правила програмування контролера мовою функціональних блоків.

Програмування контролера Zelio Logic2 мовою FBD-блоків

Програми для контролера Zelio Logic2 мовою функціональних блоків (FBD – Function Block Diagram) можна писати, використовуючи спеціалізоване програмне забезпечення Zelio Soft2.

Створення нової програми

Для того, щоб створити нову програму необхідно виконати наступні дії.

1. Вибрати меню **File/New** або натиснути піктограму **Create a new program** під час запуску програми Zelio Soft2. В результаті з'явиться вікно вибору типу модуля.

2. Виберіть потрібний модуль за допомогою ЛК миші. Модулі згруповані за такими ознаками:

- кількість входів-виходів;
- наявність або відсутність дисплею;
- можливість під'єднання шасі розширення.

В результаті здійснення вибору типу модуля з'являється весь список модулів даного типу у вигляді таблиці.

3. Виберіть модуль подвійним натисканням миші або натисніть клавішу **Далее (Next)**. На цьому етапі може виникнути такі ситуації:

- модуль не підтримує можливості розширення та програмується лише мовою LD – перейдіть до кроку 7;
- модуль не підтримує можливості розширення і програмується мовами LD та FBD – перейдіть до кроку 6;
- модуль підтримує розширення, тоді на екрані з'явиться інформація про тип модуля, вибраний у попередньому пункті та типи можливих розширень. Натиснувши кнопку **Add**, можна вибрати необхідне розширення. Для переходу до наступного кроку натисніть кнопку **Далее**. Модуль розширення при необхідності можна видалити кнопкою **Delete**.

4. Виберіть необхідний модуль розширення.

5. Підтвердіть вибір (кнопка **Далее**). В результаті з'явиться вікно вибору мови програмування.

6. Виберіть потрібну мову (LD або FBD). Натисніть кнопку **Далее (Next)**.

7. З'явиться вікно редагування програми.


Мова FBD-блоків (Function Blocks Diagram Language)

При програмуванні мовою FBD-блоків можливе використання 2-х типів вікон:

- вікно редагування;
- вікно супервізора.

Вікно редагування

FBD-програми створюються у вікні редагування. Вікно редагування можна запустити за допомогою меню **Mode/Edit** або за допомогою

піктограми .

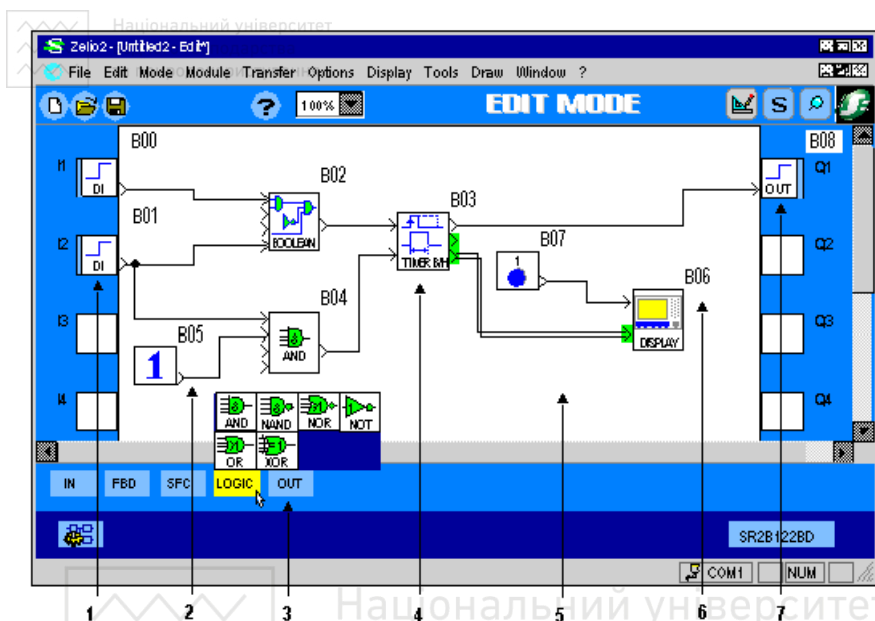




Рис. 2.1 Вікно редагування FBD-програми

Таблиця 2.1

Номер	Опис
1	Входи Zelio
2	З'єднання між 2-ма функціональними блоками
3	Панель функціональних блоків
4	Функціональний блок
5	Вікно програми
6	Номер функціонального блоку
7	Виходи Zelio

Вікно супервізора

Вікно супервізора можна запустити наступним чином:


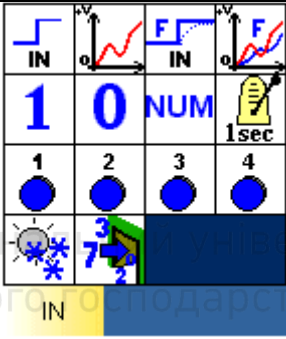
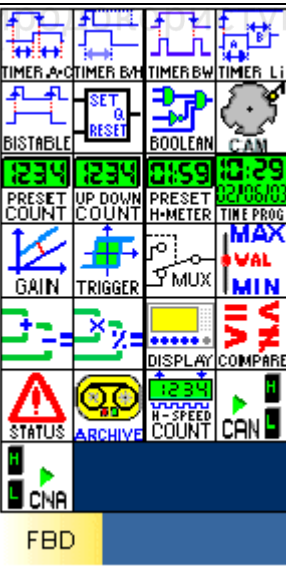
- **симуляція (simulation):** пункт меню [mode/simulation](#) або піктограма ;
- **моніторинг (monitoring):** пункт меню [mode/monitoring](#) або піктограма .


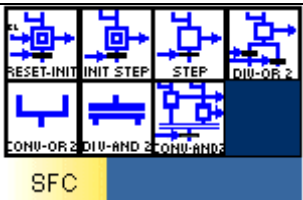



У вікно супервізора можна вставити частини FBD-програми, скопійовані у вікні редагування. У вікні супервізора можна малювати, вставляти текст і малюнки (*.bmp). Доступний також режим емуляції.

Панель елементів

Усі елементи, які необхідні для створення FBD-програми, розміщені на панелі елементів у нижній частині вікна редагування. Елементи розділені на функціональні групи.

Таблиця 2.2

<p>1. Входи</p> 	 <p>IN</p>
<p>2. Стандартні функції</p>	 <p>FBD</p>

	Національний університет водного господарства та природокористування	
3. SFC-функції		
4. Логічні функції		
5. Виходи		
6. Входи-виходи Modbus		

1. Входи

Елементи мови FBD-блоків











Дискретні входи (Discrete-type inputs) –



Тип дискретного входу можна вибрати із вікна [Параметри \(Parameters\)](#).

Таблиця 2.3

Тип	Неактивний стан	Активний стан
Дискретний вхід (discrete input)		
Контакт (contact)		
Вимикач (limit switch)		
Давач наближення (proximity sensor)		


Давач присутності (presence sensor)		
Кнопка з під світкою (illuminated pushbutton)		
Перемикач (selector switch)		
Кнопка (pushbutton)		
Нормально відкрите реле (normally open relay)		

Дискретний вхід з фільтром (Filtered discrete-type input) –

З метою зменшення або навіть ліквідації завад після дискретного входу встановлюють фільтр. Якщо сигнал стабільний на протязі усього визначеного періоду часу, вихід дискретного входу з фільтром приймає виміряне значення, в іншому випадку сигнал лишається незмінним.

Значення параметру (від 1 до 255), введене у вікно **Параметри (Parameters)**, використовується для визначення мінімального часу, на протязі якого сигнал давача повинен бути стабільним. Даний параметр множиться на циклічний час модуля (Zelio).

У режимі симуляції або моніторингу при активації дискретного входу

з фільтром він набуває такого вигляду – .

Аналоговий вхід (Analog-type input) –

Аналоговий вхід доступний для всіх модулів постійного струму. Вхідна аналогова напруга конвертується у числове значення від 0 до 255 8-бітним АЦП. Аналоговими входами можуть бути входи від ІВ до ІГ модуля.



За замовчуванням значення аналогового входу лежить в межах 0 – 10 В постійного струму. Тип електричного з'єднання входу можна конфігурувати у вікні **Параметри (Parameters)**. У цьому ж вікні можна вибрати тип аналогового входу.

Таблиця 2.4

Тип	Зображення
Вхід за замовчуванням (Input by default)	
Вхід (Input)	
Температура (Temperature)	
Потенціометр (Potentiometer)	

Аналоговий вхід з фільтром (Filtered analog input) –

Робота такого входу аналогічна аналоговому з тією відмінністю, що після аналогового входу встановлений низькочастотний фільтр.

- Фільтр повністю відновлює вхідний сигнал (частоту, амплітуду та зсув фаз), якщо частота сигналу значно нижча за частоту зрізу фільтра.
- Коли частота вхідного сигналу наближається до частоти зрізу фільтра, вихідний сигнал тієї ж частоти стає меншим за амплітудою і зсунутим за фазою.
- Коли частота вхідного сигналу рівна частоті зрізу фільтра, вихідний сигнал знижується на 30% і зсувається за фазою на 45° .
- Якщо частота вхідного сигналу перевищує частоту зрізу фільтра, вихідний сигнал значно знижується, а зсув за фазою досягає 90° .

У вікні **Параметри (Parameters)** зазначаються такі параметри аналогового входу:



Національний університет

вхідна напруга; за замовчуванням – від 0 до 10 В постійного струму;

- частота зрізу фільтра (від 0,06 до 88,25 Гц).



Ціле числове значення (Integer type input) –

Дана функція використовується для введення цілого 16-бітного цілого числа в діапазоні від –32768 до +32767. Цілий числовий вхід може бути присвоєний входам від J9 до JВ модулів розширення.

Спеціальні входи у мові FBD (Special inputs in FBD language)

У мові FBD доступні наступні спеціальні входи:

- кнопка;
- дискретні константи;
- числові константи;
- літній час;
- миготіння на протязі 1 секунди.

Дані входи можуть розміщуватися на сторінці FBD-програми.

Кнопки (Button type inputs) –



Входи типу “кнопка” відповідають клавішам на передній панелі Zelio. Дані входи можуть бути вставлені у FBD-діаграму і у режимі симуляції або моніторингу можуть симулювати контакти.

Дискретні константи (Discrete constant-type inputs)

Існує 2 типи дискретних констант:

- 1 -
- 0 -

У режимі емуляції або моніторингу можна використовувати дані входи в інверсному стані. Тоді символи набувають червоного кольору.



Числові константи (Numerical constant-type inputs) –

Числова константа **NUM** представляє собою ціле число у діапазоні від -32768 до +32767. Дана константа може використовуватися для встановлення значення функціональних не з'єднаних входів:



- GAIN;
- COMP IN ZONE;
- TRIGGER.

Значення константи можна встановити у вікні **Параметри (Parameters)**. У режимі емуляції або моніторингу є можливість змінювати константу.

Вхід відображення літнього та зимового часу (Summer Time Input)

Вхід активний на протязі літнього часу –  і неактивний на протязі зимового часу – .






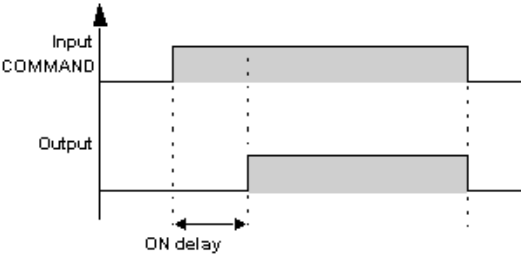
Імпульсний вхід (Blinking input)

Даний вхід активний кожену секунду. Його активний символ – , його неактивний символ – .

2. Стандартні функції

Таблиця 2.5








Функція	Сим-вол	Пояснення												
BOOLEAN Equation		Вихід приймає значення 1 або 0 в залежності від значень входів згідно таблиці істинності.												
SET-RESET		<p>Вихід Q набуває значення 1 або 0 згідно таблиці істинності:</p> <table border="1"> <thead> <tr> <th>SET</th><th>RESET</th><th>Q</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>0</td></tr> <tr> <td>імпульс 0 → 1</td><td>0</td><td>1</td></tr> <tr> <td>0</td><td>імпульс 0 → 1</td><td>0</td></tr> </tbody> </table>	SET	RESET	Q	0	0	0	імпульс 0 → 1	0	1	0	імпульс 0 → 1	0
SET	RESET	Q												
0	0	0												
імпульс 0 → 1	0	1												
0	імпульс 0 → 1	0												





 Національний університет водного господарства та природокористування		1	1	стан виходу залежить від встановленого пріоритету входів
PRESET COUNT Up-Down Counter		Лічильник, який використовується для підрахунку імпульсів від нуля до значення уставки (preset) та від значення уставки до нуля.		
FAST COUNTER		Лічильник, який дозволяє вести підрахунок імпульсів з частотою до 1 КГц. Входи даного лічильника неявно зв'язані з входами П1 (сумуючий вхід) та І2 (декрементуючий вхід) контролера. Тому входи П1 та І2 не рекомендується використовувати для інших цілей при використанні швидкого лічильника. Даний функціональний блок активний лише тоді, коли активний вхід ENABLE. Роботу лічильника не можна симулювати у середовищі програмування.		
UP-DOWN COUNTER		Лічильник, який використовується для підрахунку імпульсів, починаючи від значення уставки, яка вираховується поза межами функціонального блоку. Щоб занести у лічильник значення уставки, необхідно підключити до входу PRESET константу NUM, аналоговий вхід тощо та подати 1 на вхід PRESET FORCING. Вихід лічильника активний, якщо число підрахованих імпульсів дорівнює або більше значення уставки PRESET.		
TIMER A/C		<p>Таймер, який може працювати у трьох режимах:</p> <p>A - затримка включення (on-delay):</p>  <p>C - затримка виключення (off-delay):</p>		


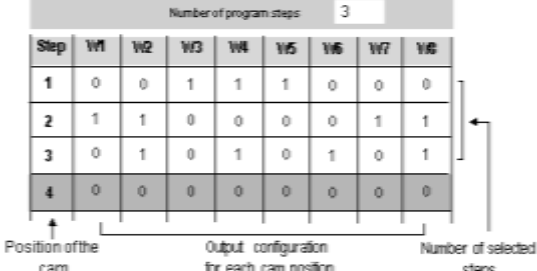


<div data-bbox="106 103 176 172" data-label="Image"> </div>	<div data-bbox="199 103 415 172" data-label="Text"> <p>Національний університет водного господарства та приладокористування</p> </div>	<div data-bbox="448 119 960 375" data-label="Figure"> </div> <div data-bbox="431 379 767 414" data-label="Text"> <p>A/C - комбінований режим:</p> </div> <div data-bbox="448 414 960 670" data-label="Figure"> </div>
<div data-bbox="112 805 252 837" data-label="Text"> <p>TIMER BW</p> </div>	<div data-bbox="300 778 378 861" data-label="Image"> </div>	<div data-bbox="425 683 974 805" data-label="Text"> <p>Таймер використовується для генерації імпульсу певної тривалості на виході при подачі імпульсу на вхід таймера. Можливі типи вхідних імпульсів:</p> </div> <div data-bbox="476 805 890 901" data-label="List-Group"> <ul style="list-style-type: none"> • імпульс по передньому фронту; • імпульс по задньому фронту; • обидва типи імпульсів. </div> <div data-bbox="425 901 974 962" data-label="Text"> <p>При налаштуванні таймера потрібно вибрати один із зазначених типів вхідних імпульсів.</p> </div>
<div data-bbox="117 1181 240 1212" data-label="Text"> <p>TIMER Li</p> </div>	<div data-bbox="300 1157 378 1240" data-label="Image"> </div>	<div data-bbox="425 967 974 1061" data-label="Text"> <p>Таймер працює у режимі Li, який призначений для генерації імпульсів. Є три модифікації даного режиму:</p> </div> <div data-bbox="476 1061 974 1125" data-label="List-Group"> <ol style="list-style-type: none"> 1. Тривала генерація імпульсів (Continuous Flashing) </div> <div data-bbox="448 1125 960 1396" data-label="Figure"> </div> <div data-bbox="476 1396 974 1436" data-label="List-Group"> <ol style="list-style-type: none"> 2. Генерація певного числа </div>


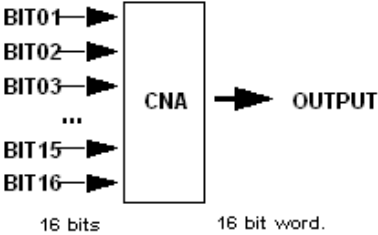

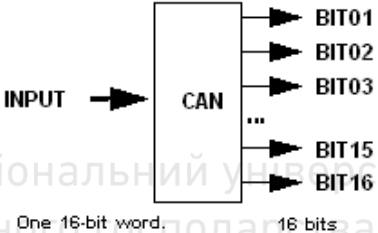
<div data-bbox="105 97 179 172" data-label="Image"> </div>	<div data-bbox="199 97 415 172" data-label="Text"> <p>Національний університет водного господарства та природокористування</p> </div>	<div data-bbox="520 113 857 145" data-label="Section-Header"> <h3>імпульсів(Number of Flashes)</h3> </div> <div data-bbox="434 153 966 411" data-label="Figure"> </div> <div data-bbox="472 419 971 483" data-label="Section-Header"> <h3>3. Генерація імпульсів на протязі певного часу (Duration of flashes)</h3> </div> <div data-bbox="442 483 983 826" data-label="Figure"> </div>
<div data-bbox="109 1046 250 1075" data-label="Text"> <p>TIMER B/H</p> </div>	<div data-bbox="301 1023 379 1102" data-label="Image"> </div>	<div data-bbox="424 834 971 957" data-label="Text"> <p>Даний таймер генерує імпульс виході під час приходу переднього фронту імпульсу на вхід. Даний таймер може працювати у двох режимах:</p> </div> <div data-bbox="474 959 552 987" data-label="List-Group"> <ul style="list-style-type: none"> • <i>B</i> </div> <div data-bbox="498 987 901 1251" data-label="Figure"> </div> <div data-bbox="474 1260 556 1289" data-label="List-Group"> <ul style="list-style-type: none"> • <i>H</i> </div>

	Національний університет водного господарства та природокористування	
COMP IN ZONE Comparison		Функція порівняння використовується для порівняння значення змінної з мінімальним та максимальним значеннями.
PRESET H- METER hour counter		Даний блок призначений для вимірювання проміжку часу, на протязі якого активний командний вхід блоку (Command). Коли період часу активності входу досягає значення уставки, стає активним вихід блоку. Уставка за часом може задаватися в годинах (32757 максимум) та у хвилинах (від 0 до 59). Скидання виходу блоку в нуль відбувається шляхом активації входу Reset.
SCHMITT TRIGGER		Блок тригер Шмітта дозволяє контролювати значення аналогової змінної по відношенню до двох сталих значень – нижньої межі та верхньої межі. Дискретний вихід блоку змінює свій стан на протилежний, якщо: <ul style="list-style-type: none"> • вхідне значення менше значення нижньої межі (min); • вхідне значення більше значення верхньої межі (max). Вихід не змінює свого значення, якщо вхідне значення знаходиться між min і max. Блок працює, тільки якщо активний дискретний вхід <u>Enable</u> .
COMPARE		Блок порівняння призначений для порівняння значень двох аналогових змінних. Вихід блоку активний, якщо результат порівняння змінних <u>Value 1</u> та <u>Value 2</u> позитивний, а дозволяючий вхід <u>Enable</u> активний або ні з чим не з'єднаний. Конкретну операцію порівняння потрібно вибрати у вікні параметрів даного блоку.

 <p>GAIN</p>		<p>Функція для масштабування аналогових сигналів за формулою: Calculation Output = A/B * Calculation Input +C Для роботи даного функціонального блоку його вхід <u>Enable Function Input</u> повинен бути активний або ні з чим не з'єднаний.</p>
<p>DISPLAY</p> 		<p>Дана функція дозволяє виводити на екран контролера такі дані, як текст (до 72 символів), дату, час, числове значення. У програмі одночасно можуть бути активними до 32 блоків Display. Вхід Enable Function Input має бути активним для запуску блоку. Вхід Value Input визначає дані, які будуть виводитися на екран:</p> <ul style="list-style-type: none"> • вхід ні з чим не з'єднаний – на екран виводитимуться дані, задані користувачем у User options zone при налаштуванні блоку; • вхід з'єднаний з джерелом даних (виходом попереднього функціонального блоку) – на екран виводитимуться дані, які надходять на вхід Value Input.
<p>TIME PROG</p>		<p>Функціональний блок Годинник, який дозволяє задавати інтервали часу у днях тижня, годинах, хвилинах, коли блок активний чи неактивний.</p>
<p>BISTABLE Impulse Relay</p>		<p>Бістабільне імпульсне реле, яке змінює стан свого виходу на протилежний під час приходу запускаючого імпульсу по передньому фронту на його вхід Command. Вхід Reset блоку призначений для скидання його виходу в нуль.</p>
<p>Multiplexin g</p>		<p>Функціональний блок мультиплексора, призначений для комутування двох інформаційних входів (Channel A і Channel B) на один вихід. Адресу входу для опитування задає адресний вхід Command. Якщо вхід Command=0 або ні з чим не з'єднаний, то опитується вхід A, якщо Command=1, то опитується вхід B.</p>

 <p>ADD-SUB</p>		<p>Функціональний блок для здійснення арифметичних операцій додавання та віднімання трьох цілочисельних даних. Значення виходу блоку розраховується за формулою:</p> <p>Calculation Output = Input 1 + Input 2 – Input 3</p> <p>Якщо вхід блоку Error Propagation=1, розрахунок не проводиться.</p> <p>Вихід Error/Overrun=1, якщо результат операції вийшов за межі відрізка [-32768; +32767] або вхід Error Propagation=1.</p>
<p>MUL-DIV</p>		<p>Функціональний блок для здійснення арифметичних операцій множення та ділення трьох цілочисельних даних. Значення виходу блоку розраховується за формулою:</p> <p>Calculation Output = Input 1 * Input 2 / Input 3</p> <p>Якщо вхід блоку Error Propagation=1, розрахунок не проводиться.</p> <p>Вихід Error/Overrun=1, якщо результат операції вийшов за межі відрізка [-32768; +32767], вхід Error Propagation=1 або вхід Input 3=0.</p>
<p>CAM BLOCK</p>		<p>Функціональний блок кулачкового механізму контролює роботу 8-ми дискретних виходів. На кожному кроці виходи приймають наперед задані значення. Обертання кулачкового механізму контролюють 2 дискретні входи:</p> <ul style="list-style-type: none"> • MOVE FORWARD – перехід на крок вперед (має більший пріоритет); • MOVE BACKWARD – перехід на крок назад. <p>Вхід RESET використовується для ініціалізації функціонального блоку (переведення його на перший крок). Вихід POSITION вказує на номер поточного кроку.</p> <p>При конфігуруванні кулачкового механізму необхідно задати наступні параметри:</p> <ul style="list-style-type: none"> • кількість кроків (number of steps) від 1 до 50; • значення виходів на кожному кроці.



	<p>Національний університет водного господарства та природокористування</p>	
<p>ARCHIVE</p> 		<p>Функціональний блок для запам'ятовування значень двох змінних.</p> <p>Входи блоку ARCHIVE:</p> <ul style="list-style-type: none"> • LATCHING – дискретний вхід, при подачі імпульсу на який відбувається запам'ятовування значень змінних; • RESET – дискретний вхід скидання блоку, при цьому запам'ятовані раніше дані втрачаються; • ARCHIVE VALUE 1 – перший вхід для запам'ятовування; • ARCHIVE VALUE 2 – другий вхід для запам'ятовування. <p>Значення змінних запам'ятовуються разом із відомості про час і дату їх надходження.</p> <p>Виходи блоку ARCHIVE:</p> <ul style="list-style-type: none"> • VALID ARCHIVE – дискретний вихід, який вказує на наявність запам'ятованих даних; • MINUTE – хвилина (від 0 до 59); • HOUR – година (від 0 до 23); • DAY – день (від 1 до 31); • MONTH – місяць (від 1 до 12); • YEAR – рік (від 0 до 99); • ARCHIVE 1 – значення входу VALUE 1; • ARCHIVE 2 - значення входу VALUE 2.
<p>Module STATUS</p>		<p>Дана функція дозволяє користувачу керувати статусом Zelio 2 і змінювати тим самим поведінку його FBD або SFC програми.</p>


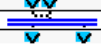
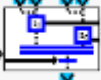

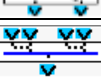
 <p>CAN bits to word conversion</p>	<p>Функціональний блок, який перетворює 16-бітний двійковий код у 16-бітне ціле слово.</p> 
 <p>CAN word to bits conversion</p>	<p>Функціональний блок, який перетворює 16-бітне ціле слово у 16-бітний двійковий код.</p> 

3. SFC-функції

SFC- функції (Sequential Function Chart) ідентичні функціям мови Grafcet стандарту IEC 1131-3. Мова програмування Grafcet призначена для представлення програми у графічній структурованій формі. Діаграма, складена із SFC- функцій, виконується з верху до низу і складається з кроків (Steps) та переходів (Transitions) між ними.

Таблиця 2.6

Функція	Символ	Опис
Initial Step		Початковий крок
Resettable Initial Step		Початковий крок з ініціалізацією за командою.


Step		Крок
AND Divergence		Перехід до двох кроків одночасно
AND Convergence		Перехід від двох паралельно виконуваних кроків до одного
OR Divergence		Перехід від одного кроку до одного або двох кроків
OR Convergence		Перехід від одного або чотирьох кроків до одиничного кроку

4. Логічні функції

У мові програмування FBD є можливість використовувати логічні функції.

Таблиця 2.7

Функція	Символ	Опис	К-сть входів	Тип входів
NO		Вихід активний, якщо вхід неактивний або не підключений і навпаки.	1	Digital
AND		Вихід активний, якщо усі входи активні або не підключені.	4	Digital
OR		Вихід активний, якщо активний хоча б 1 з входів.	4	Digital
NO AND		Вихід активний, якщо хоча б 1 вхід неактивний. Якщо усі входи активні або не підключені, то вихід неактивний.	4	Digital
NO OR		Вихід активний, якщо усі входи неактивні або не підключені. Якщо хоча б 1 вхід активний, то вихід	4	Digital

національний університет водного господарства та природокористування	неактивний.		
EXCLUSIVE OR		Вихід активний, якщо один вхід неактивний, а інший вхід активний або не підключений. Якщо обидва входи активні, неактивні або не підключені, то вихід неактивний.	2 Digital



















5. Виходи















Дискретні виходи (Discrete-type outputs) –

Тип дискретного виходу можна вибрати із вікна [Параметри \(Parameters\)](#).

Таблиця 2.8

Тип	Неактивний стан	Активний стан
Дискретний вихід (discrete output)		
Нормально відкрите реле (normally open relay)		
Лампа (lamp)		
Твердотільне реле (solid state relay)		
Вентиль (valve)		
Поршень (actuator)		
Двигун (motor)		
Резистор (resistance)		
Звуковий сигнал (audible signal)		

Зелена індикаторна лампа (green indicator lamp)		
Червона індикаторна лампа (red indicator lamp)		
Оранжева індикаторна лампа (orange indicator lamp)		
Світловий індикатор (indicator light)		
Нагрівач (heating)		
Вентилятор (fan)		

Цілочисельний вихід (Integer-type output) –

Дана функція використовується для генерування 16-бітного цілого числа у діапазоні від -32768 до +32767. Цілий числовий вихід може бути присвоєний виходам від O9 до JВ модулів розширення.

Вихід підсвітки екрану (LCD Screen Backlighting output)

Дана функція дозволяє програмі контролювати режим підсвічування екрану, вона не може бути присвоєна релейним виходам.



- неактивний стан,



- активний стан.

6. Входи-виходи Modbus

Модуль розширення Modbus **SR3 MBU01BD** може бути підключений до базового модуля Zelio **SR3 BxxxBD**. У FBD режимі 4 (16 бітів) вхідних слова (з J1XT1 по J4XT1) і 4 вихідних слова (з O1XT1 по O4XT1) доступні програмно. Модуль Zelio працює лише у режимі Modbus slave.

Параметри



Параметри встановлюються у робочому середовищі за допомогою пункту меню **Edition\Configuration** у режимі **program\Extension MODBUS Tab** або при подвійному натисканні на піктограму **Program**



Configuration.

При запуску програми Zelio ініціалізує модуль розширення Modbus.

Модуль має 4 параметри:

- число UART з'єднань і frame формат у мережі Modbus;
- швидкість передачі даних у біт/с;
- UART- парність;
- мережева адреса модуля розширення Modbus.

Входи Modbus

Модуль розширення Modbus SR3 MBU01BD має 4 (16 бітів) входи:

- J1XT1 – шістнадцяткова адреса (0010);
- J2XT1 – 0x0011;
- J3XT1 – %0x0012;
- J4XT1 – %0x0013.

Дані завантажуються з пристрою, що працює у режимі master.

Виходи Modbus

Модуль розширення Modbus SR3 MBU01BD має 4 (16 бітів) виходи:

- O1XT1 – шістнадцяткова адреса (0000);
- O2XT1 – 0x0001;
- O3XT1 – 0x0002;
- O4XT1 – %0x0003.

Дані завантажуються у пристрій, що працює у режимі master.

3. Програма роботи

1. Вивчити будову та характеристики контролера Zelio Logic 2.
2. Дослідити правила підключення контролера Zelio Logic 2.
3. Навчитися користуватися меню контролера Zelio Logic 2
4. Навчитися програмувати контролер Zelio Logic 2 мовою драбинкових діаграм (LD) за допомогою командних клавіш передньої панелі.
5. Навчитися програмувати контролер Zelio Logic 2 мовою драбинкових діаграм (LD) у середовищі Zelio Soft 2.



6. Навчитися програмувати контролер Zelio Logic 2 мовою FBD-блоків у середовищі Zelio Soft 2.
7. Навчитися записувати програми у контролер.
8. Навчитися створювати вікно супервізора у середовищі Zelio Soft 2.
9. Виконати завдання, поставлені викладачем.

4. Опис лабораторного обладнання

1. Персональний комп'ютер.
2. Операційна система Windows.
3. Лабораторний стенд з логічним контролером Zelio Logic.
4. Середовища програмування Zelio Soft, Zelio Soft 2.

5. Порядок виконання роботи

1. Напишіть програму для програмованого логічного контролера (ПЛК) Zelio Logic мовою драбинкових діаграм (LD) для системи контролю та керування зазначеним процесом.
2. Проведіть емуляцію та налагодження програми у середовищі програмування Zelio Soft. Для цього скористайтеся меню Simulation та натисніть кнопку Run. Після проведення перевірки роботи програми натисніть кнопку Stop.
3. Запишіть створену програму у ПЛК та дослідіть її роботу. Для цього необхідно виконати наступні дії:
 - подати живлення на стенд;
 - перевести контролер у режим STOP;
 - увійти у головне меню, натиснувши клавішу Sel./OK;
 - вибрати пункт меню Transfer.\ PC -> Module;
 - у середовищі програмування Zelio Soft вибрати пункт меню Transfer\ Transfer Program\ PC -> Module.
4. Навчитися керувати запуском та зупинкою програми з клавіатури контролера та з комп'ютера. Для керування виконанням програми з клавіатури контролера скористайтеся пунктом меню Run\ Stop, з комп'ютера – пунктами меню середовища Zelio Soft Transfer\ Run Module, Transfer\ Stop Module.
5. Для відслідковування роботи програми на комп'ютері виберіть пункт меню Mode\ On Line Monitoring Mode.
6. Складіть пояснюючу таблицю використаних входів і виходів ПЛК, в якій зазначте назву входу-виходу, тип, призначення.



7. Напишіть програму для програмованого логічного контролера (ПЛК) Zelio Logic2 мовою FBD-блоків для системи контролю та керування процесом. При цьому використайте середовище програмування Zelio Soft2.
8. Створіть вікно супервізора, в якому намалюйте ФСА технологічного процесу. Для того, щоб переключитися у вікно супервізора скористайтеся меню Window\Supervision.
9. Проведіть емуляцію та налагодження програми у середовищі програмування Zelio Soft2. Дослідіть роботу програми у середовищі програмування Zelio Soft2. При цьому скористайтеся керуючими кнопками, наведеними у таблиці 9.
10. Складіть пояснюючу таблицю використаних входів і виходів ПЛК, в якій зазначте назву входу-виходу, тип, призначення.
11. Зробіть висновки щодо можливостей двох мов програмування – LD та FBD – для розробки програмного забезпечення для керування технологічним процесом.

Завдання 2.1

Керування мішалкою для молока та вершків

За допомогою Zelio необхідно керувати мішалкою для молока та вершків на молочній фермі. За допомогою перемикача режимів роботи може бути вибраний автоматичний режим або режим безпосереднього керування. Несправності сигналізуються за допомогою лампи і аварійного звукового сигналу.

Якщо перемикач режимів роботи знаходиться у положенні «Автоматика» (I1), то мішалка (на Q1) запускається негайно. Автоматичний режим означає, що мішалка вмикається і вимикається через задані інтервали часу (15 секунд включена, 10 секунд – пауза). Мішалка працює із цими інтервалами, поки перемикач режимів роботи не буде переведений у положення 0. В режимі прямого керування (I2 – положення «Пряме керування») мішалка працює без врахування інтервалів часу. При спрацюванні автомата захисту двигуна (I3) активується лампа сигналізації про несправність (Q2) і аварійний звуковий сигнал (Q3). Інтервали, з якими подається звуковий сигнал, встановлюються за допомогою давача тактових імпульсів на 3 секунди. Звуковий сигнал може бути перерваний за допомогою кнопки скидання на I4. Якщо несправність усунена, то сигнальна лампа і звуковий сигнал знову скидаються. За допомогою



кнопки «Контроль аварійної сигналізації» на I5 можна пеервірити як сигнальну лампу, так і звуковий сигнал.

Компоненти, які використовуються:

Модуль - SR2-A101BD

- I1 – перемикач режимів роботи – положення «Автоматика» (NO – normally opened);
- I2 – пеермикач режимів роботи – положення «Пряме керування» (NO – normally opened);
- I3 – аварійний контакт автомата захисту двигуна (NO – normally opened);
- I4 – кнопка скидання звукового сигналу (NO – normally opened);
- I5 – кнопка пеервірки функціонування аварійної сигналізації (NO – normally opened);
- Q1 – мішалка;
- Q2 – сигнальна лампа;
- Q3 – аварійний звуковий сигнал.

Переваги і особливості:

Довільна зміна інтервалів змішування. Необхідно менше компонентів, ніж при використанні рішення без контролера.

6. Контрольні запитання

1. Які функціональні блоки були вами використані у програмі для керування процесом? Яке їх призначення?
2. Для чого призначений функціональний блок ГОДИННИК? Яким чином він програмується?
3. Як визначити параметри ПЛК Zelio Logic за його маркуванням?
4. Яким чином запрограмувати логічний контролер Zelio Logic через командні кнопки?
5. Яким чином запрограмувати логічний контролер Zelio Logic через середовище Zelio Soft?
6. Яким чином запрограмувати контролер мовою драбинкових діаграм (Ladder Diagram Language)?
7. Яким чином запрограмувати контролер мовою FBD-блоків (Function Blocks Diagram Language)?
8. Яким чином записати програму у ПЛК з комп'ютера? Яким чином зчитати записану програму з ПЛК на комп'ютер?



9. Як можна відслідкувати роботу програми з комп'ютера у реальному часі?
10. Для автоматизації яких процесів слід використовувати ПЛК Zelio Logic? Які можливості має ПЛК Zelio Logic для роботи у дворівневих системах контролю та керування?
11. Порівняйте можливості двох мов програмування – LD та FBD – при написанні програм для керування процесом.

