



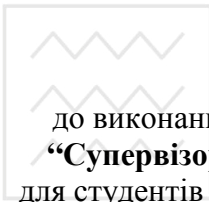
Національний університет
водного господарства
та природокористування

Міністерство освіти і науки України

**Національний університет водного господарства та
природокористування**

**Кафедра автоматизації, електротехнічних та комп'ютерно-
інтегрованих технологій**

04-03-56



Методичні вказівки

до виконання лабораторної роботи №5 з дисципліни
“Супервізорні системи керування та збору даних”
для студентів спеціальності 8.05020201 „Автоматизоване
управління технологічними процесами” денної форми
навчання

Рекомендовано методичною комісією за
спеціальністю „Автоматизоване
управління технологічними процесами”
Протокол № 4 від 10.12.2013 р.

Рівне 2013



Національний університет

Методичні вказівки до виконання лабораторної роботи №5 з дисципліни “Супервізорні системи керування та збору даних” для студентів спеціальності 8.05020201 „Автоматизоване управління технологічними процесами” денної форми навчання / Пастушенко В.Й., Стеценко А.М. – Рівне: НУВГП, 2013. - 51 с.

Упорядники: Пастушенко В.Й., к.т.н., професор, Стеценко А.М., старший викладач.

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



Національний університет
водного господарства
та природокористування

© Пастушенко В.Й.,
Стеценко А.М., 2013
© НУВГП, 2013



Робота №5. Розробка і випробування системи контролю та керування процесом приготування суміші з трьох компонентів на базі ПЛК Schneider Micro TSX 37-22, програмного забезпечення PL7 PRO і SCADA системи Citect.

1. Мета роботи

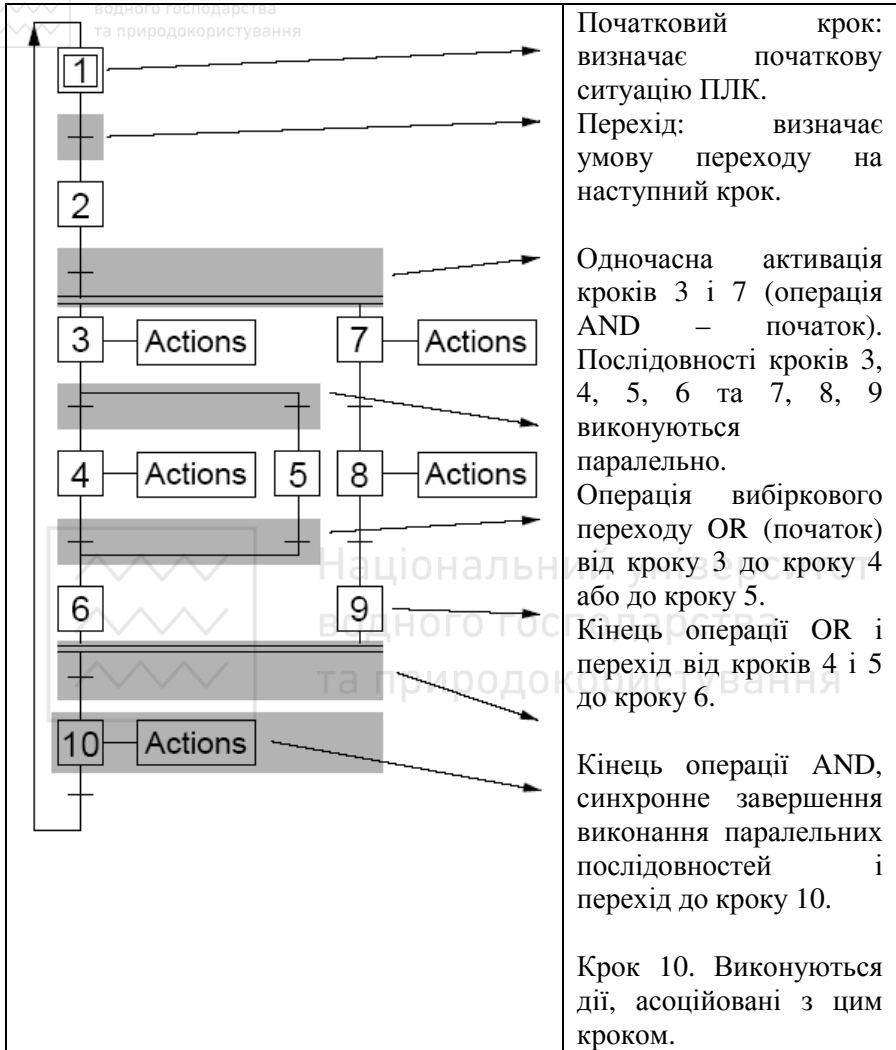
Навчитися програмувати промисловий логічний контролер (ПЛК) Schneider Micro TSX 37-22 мовою Grafset у середовищі програмування PL7 PRO. Навчитися розробляти дворівневі системи контролю та керування на базі Schneider Micro TSX 37-22.

2. Теоретичні відомості

2.1 Створення програмного забезпечення мовою Grafset для керування технологічним процесом на базі ПЛК Schneider Micro

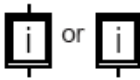

2.1.1 Основні принципи написання програм мовою Grafset

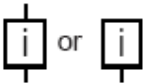

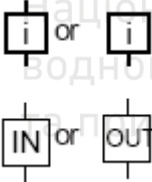

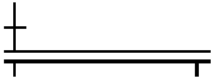
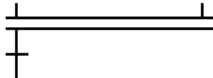
Мова програмування Grafset ще називається Sequential Function Chart language (SFC) (мова послідовного функціонального рисунку) і належить до стандарту ІЕС 1131-3. Мова Grafset використовується для представлення програми у структурованому графічному вигляді. При написанні програми використовуються спеціальні графічні елементи:

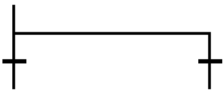
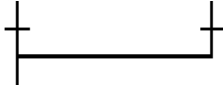





2.1.2 Графічні елементи мови Grafset

Таблиця 5.1

Назва	Зображення	Функція
Початкові кроки (Initial steps)	 or 	Початкові кроки ініціалізуються при запуску програми або після

<p>Одиничний крок (Single steps)</p>		<p>холодного старту.</p> <p>Максимальна кількість кроків:</p> <ul style="list-style-type: none"> - 1-96 для TSX 37-10, - 1-128 для TSX 37-20, - 1-250 для TSX 57. <p>Максимальна кількість одночасно активних кроків конфігурується.</p>
<p>Макрокроки (Macro-step)</p>		<p>Макрокроки – цілісна структура (підпрограма), яка складається з кроків і переходів. Максимальна кількість макрокроків – від 0 до 63 для ПЛК TSX 57.</p>
<p>Кроки макрокроків (Macro-step steps)</p>		<p>Максимальна кількість кроків для кожного макрокрока – від 0 до 250 для TSX 57.</p> <p>Кроки IN та OUT – початок та кінець макрокрока. З кроком OUT не можуть бути асоційовані дії.</p>
<p>Переходи (Transitions)</p>		<p>Перехід від одного кроку до іншого. Перехід відбувається, коли умова переходу, асоційована з ним, істинна. Максимальна кількість переходів – 1024. Максимальна кількість одночасно активних одночасно, конфігурується.</p>
<p>Операція AND початок (AND divergences)</p>		<p>Перехід від одного кроку до декількох кроків (максимум 11) одночасно.</p>
<p>Операція AND кінець (AND convergences)</p>		<p>Перехід від декількох кроків до одного. Відбувається одночасна деактивація</p>

		декількох (максимум 11) кроків.
Операція OR початок (OR divergences)		Перехід від одного кроку до декількох (максимум 11) згідно умов переходу.
Операція OR кінець (OR convergences)		Перехід від декількох кроків до одного. Завершення операції вибору OR.
Конектор-приймач (Source connector)		n – номер кроку, від якого здійснений перехід (крок-джерело).
Направляючий конектор (Destination connector)		n – номер кроку, до якого здійснюється перехід.
З'єднувальні лінії (Directed links)		Використовуються для переходу до певного кроку (пропуск декількох кроків, повторне виконання кроків).

Максимальна кількість кроків (основні кроки + кроки макрокроків) у секції Grafset не повинна перевищувати 1024 для ПЛК TSX 57.

2.1.3 Об'єкти мови Grafset

Таблиця 5.2

Назва	Адреса	Опис
Біти кроків (1=активний крок) (Step bits)	%Xi	Стан кроку i (i від 0 до n, залежно від процесора).
	%XMj	Стан макрокroku j (j від 0 до 63 для TSX/PMX/PCX 57).
	%Xj.i	Стан кроку i макрокroku j.
	%Xj.IN	Стан вхідного кроку макрокroku j.
	%Xj.OUT	Стан вихідного кроку макрокroku j.
Системні біти (Grafset system bits)	%S21	Ініціалізує Grafset секцію.
	%S22	Скидає всі секції Grafset в нуль.
	%S23	Заморожує Grafset секцію.

	%S24	Скидає макрокроки в нуль згідно з системними словами %SW22-%SW25.
	%S26	Встановлюється в 1 при: <ul style="list-style-type: none"> - переповненні таблиці (кроки/переходи); - виконання некоректної секції (направляючий конектор вказує на крок, який не належить до даної секції).
Слова кроків (Step words)	%Xi.T	Активний час кроку і Grafset секції.
	%Xj.i.T	Активний час кроку і макрокроку j.
	%Xj.IN.T	Активний час вхідного кроку макрокроку j.
	%Xj.OUT.T	Активний час вихідного кроку макрокроку j.
Системні слова (Grafset system words)	%SW20	Слово, яке вказує кількість активних кроків, кроків для активації та кроків для деактивації для даного циклу.
	%SW21	Слово, яке вказує число доступних переходів, недоступних переходів та переходів, які будуть доступні для даного циклу.
	%SW22 – %SW25	Визначають макрокроки для скидання бітом %S24.

Біти кроків %Xi, макрокроків %XMj, кроків макрокроків %Xj.i, %Xj.IN, %Xj.OUT:

- Дорівнюють 1, коли кроки активні.
- Доступні для перевірки у всіх задачах, але записані можуть бути на етапі попереднього програмування (preprocessing) основної задачі мовами LD, IL, ST.
- Дані біти не можуть бути індексовані.

Слова %Xi.T, %Xj.i.T, %Xj.IN.T, %Xj.OUT.T:



- Змінюються від 0 до 9999, нарощуються кожні 100 ms.
- Зміна значень відбувається, коли відповідний крок активний.
- При деактивації відповідного кроку значення часу заморожується.
- При повторній активації кроку значення часу обнулюється, а потім починає наростати.
- Кожне слово для зберігання часу, на протязі якого активний крок, закріплюється за цим кроком.
- Дані слова не можуть бути індексовані.

2.1.4 Можливості мови Grafset

Можливості мови Grafset залежать від процесора ПЛК.

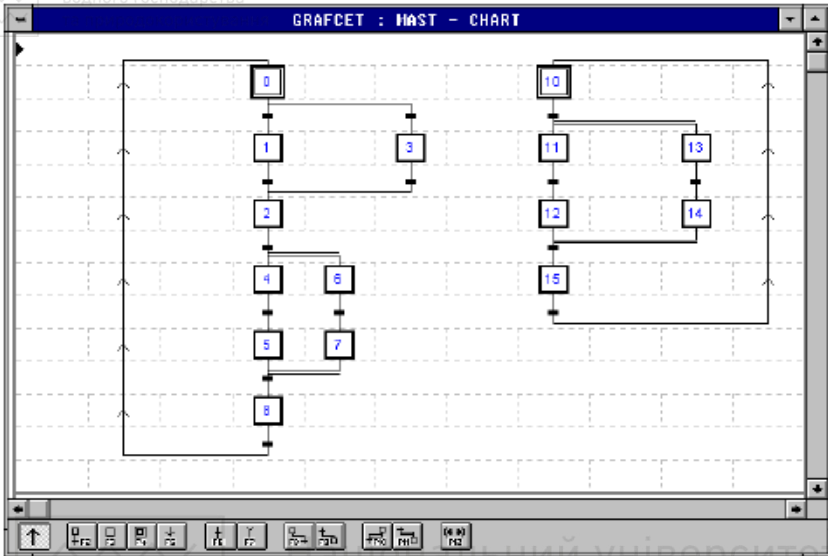
Таблиця 5.3

Number	TSX 37-10		TSX 37-20		TSX 57	
	Default	Max	Default	Max	Default	Max
Main chart steps	96	96	128	128	128	250
Macro-steps	0	0	0	0	8	64
Macro-step steps	0	0	0	0	64	250
Total steps	96	96	128	128	640	1024
Steps active simultaneously	16	96	20	128	40	250
Transitions enabled simultaneously	20	192	24	256	48	400

Кількість одночасно активних переходів не повинна перевищувати 64. Загальна кількість переходів не повинна перевищувати 1024.

2.1.5 Представлення секції Grafset

Секція Grafset програмується на 8 сторінках (від 0 до 7). Кожна сторінка має 14 рядків та 11 колонок, які розділяють її на 154 чарунок. Один графічний елемент займає одну чарунок.



Правила написання програми:

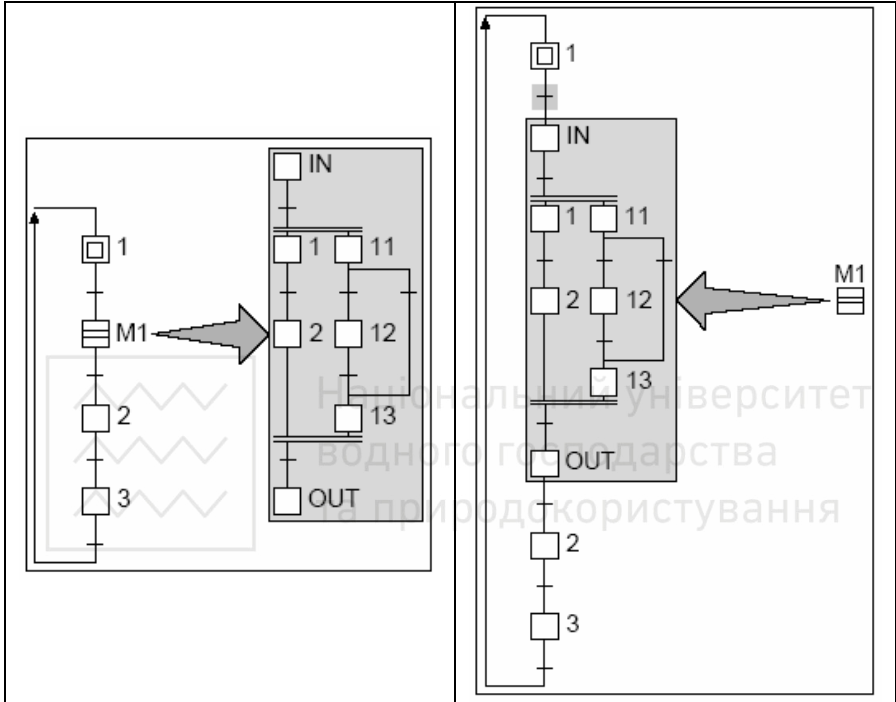
- Перша лінія використовується для розміщення конектора-приймача.
- Остання лінія використовується для розміщення направляючого конектора.
- Парні лінії (від 2 до 12) служать для розміщення кроків і направляючих конекторів.
- Непарні лінії (від 3 до 13) служать для розміщення переходів і конекторів-приймачів.
- Кожен крок нумерується (від 0 до 127) у довільному порядку.
- Декілька програм може бути розміщено на одній сторінці.

2.1.6 Макрокроки (macro-steps)

Макрокроки – цілісна структура, що складається з кроків та переходів між ними. Макрокроки завжди починається з кроку IN і завершується кроком OUT. Крок IN підкоряється тим же правилам, що і одиничні кроки, а крок OUT не може мати асоційованих з ним дій.



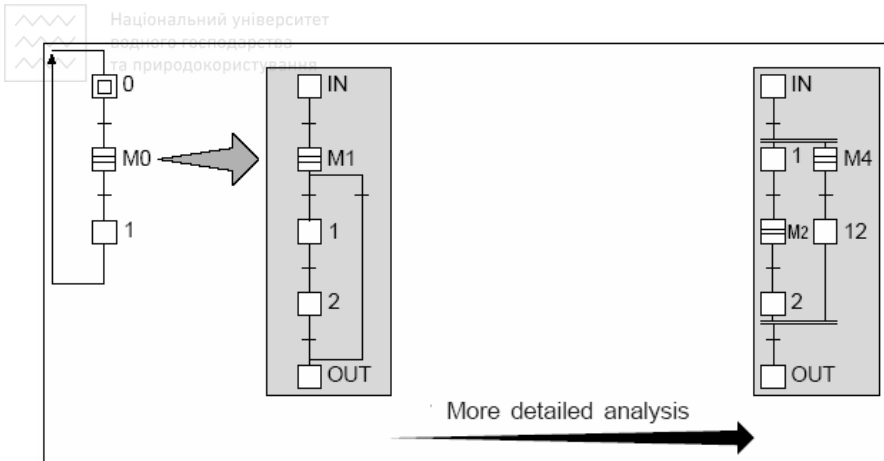
У прикладі, наведеному нижче, макрокрок M1 активізується тоді, коли крок 1 активний і істинним є перехід від кроку 1. Макрокроки M1 деактивується, коли його крок OUT активний і істинним є перехід M1 -> 2. Тоді активним стає крок 2.



У середовищі PL7 можна використати до 64 макрокроків (M0 – M63). Один макрокрок може бути запрограмований на 8 сторінках і містити до 250 кроків плюс крок IN та крок OUT. Один макрокрок може містити в собі інші макрокроки. Максимальне число ієрархічних рівнів – 64.

Макрокроки може містити 1 або кілька початкових кроків (initial steps). Вони активізуються під час запуску програми на виконання або ініціалізуються програмно. Тоді макрокрок стає активним.

Таким чином, використання макрокроків дозволяє спростити структуру основної секції Grafset, зробити її більш придатною для читання і розуміння.



2.1.7 Дії, асоційовані з кроками

Кожен крок має асоційовані з ним дії, запрограмовані мовами LD, IL або ST. Ці дії виконуються тоді, коли крок, до якого вони належать, активний. Середовище PL7 надає можливість створювати наступні типи дій:

- дії по активації (actions on activation): дії, які виконуються 1 раз під час активації кроку, до якого вони належать;
- дії по деактивації (actions on deactivation): дії, які виконуються 1 раз під час деактивації кроку, до якого вони належать;
- тривалі дії (continuous actions): дії, які виконуються постійно на протязі всього активності кроку, до якого вони належать.

Дані 3 типи дій можуть бути присвоєні для кожного кроку.

Кожна дія являє собою програму, назва якої має наступний вигляд:

MAST - <Grafcet section name> - CHART (or MACROK)- PAGE n %Xi x

x = P1 – дія по активації;

N1 – тривала дія;

P0 – дія по деактивації;

n = номер сторінки;

i = номер кроку, до якого належить дія.

Приклад: MAST - Paint - CHART - PAGE 0 %X1 P1



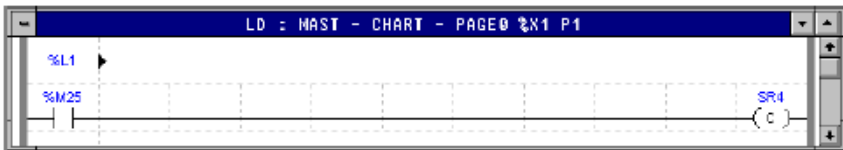
Дія по активації кроку 1, який розміщений на сторінці 0 секції Paint.

Дії по активації та деактивації кроку

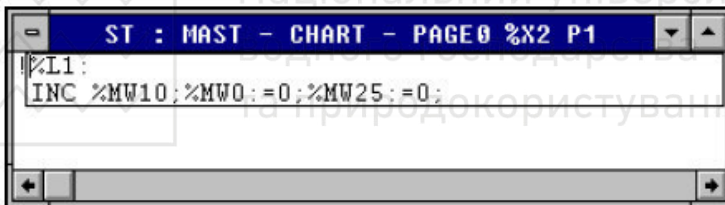
Виконуються тільки 1 раз за кожний цикл програми. Використовуються для виклику підпрограм, нарощування значення лічильника тощо.

Приклади:

- Виклик підпрограми:



- Інкремент слова %MW10, скидання в нуль слів %MW0 та %MW25.

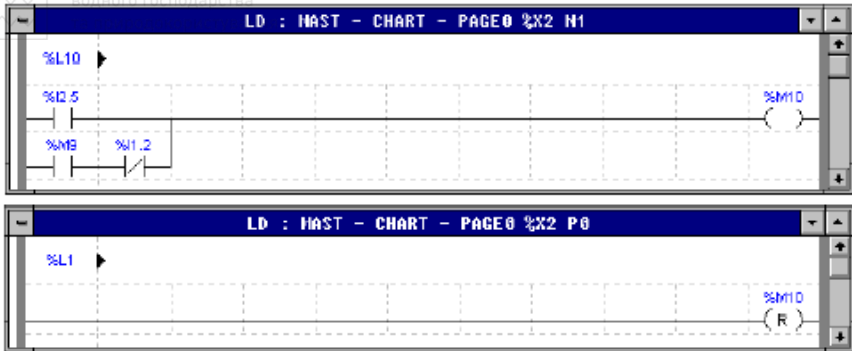


Тривалі дії

- Дії з умовою (Conditional action)

Приклад:

Біт %M10 керується дискретним входом %I2.5, внутрішнім бітом %M9 та дискретним входом %I1.2. Поки активний крок 2 буде виконуватися дана дія. Після деактивації кроку 2 значення біта %M10 запам'ятовується і лишається без змін. Тому необхідно скинути біт %M10 в нуль у дії по деактивації кроку 2.



- **Дії з умовою за часом (Timed conditional action)**

Приклад:

Біт %M12 буде рівним 1, коли час виконання кроку 3 буде меншим за 10 секунд (базовий час становить 100 мс).



- **Безумовні дії (Unconditional actions)**

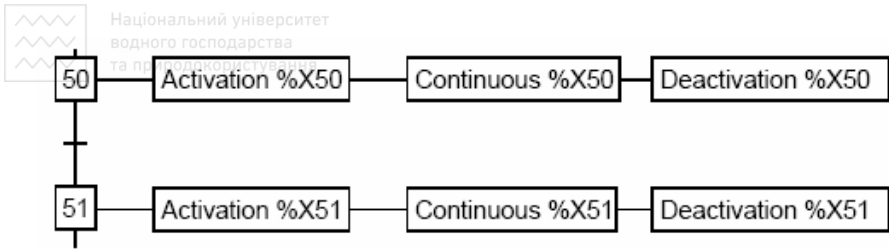
Порядок виконання дій

Приклад:

Коли крок 51 стає активним, дії виконуються у такій послідовності:

- дії по деактивації кроку 50;
- дії по активації кроку 51;
- тривалі дії кроку 51.

Після деактивації кроку 51 тривалі дії, асоційовані з ним, перестають виконуватися.



2.1.8 Дії, асоційовані з переходами

- Кожен перехід має асоційовану з ним умову, яка програмується мовами LD, IL або ST.
- Умова переходу перевіряється ПЛК тоді, коли перехід стає активним.
- Перехід, який не запрограмований, завжди хибний, тобто його результат дорівнює false.

Кожен перехід має назву, яка має наступний вигляд:

MAST - <Grafcet section name> - CHART (or MACROk) - PAGE n %X(i) --> % X(j)

n = номер сторінки;

i = номер кроку, від якого здійснюється перехід;

j = номер кроку, до якого здійснюється перехід.

Приклад: MAST - Paint - CHART - PAGE 0 %X(0) --> %X(1)

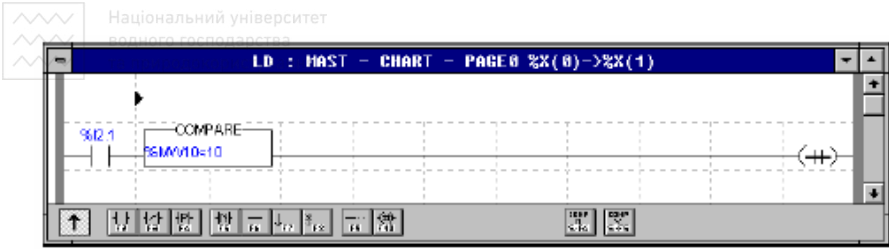
Умова переходу від кроку 0 до кроку 1 на сторінці 0 секції Paint.

Якщо активізуються одночасно декілька кроків, друга адреса є адресою першого кроку зліва.

Правила програмування переходів мовою драбинкових діаграм LD

Програма має вигляд стандартного рангу, як і у звичайній LD програмі. Але можуть використовуватися тільки такі елементи:

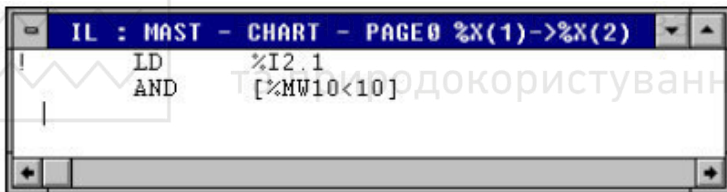
- контакти (%Mi, %I, %Q, %TMi.D), блоки порівняння;
- котушки переходу (інші види котушок не важливі у даному випадку).



Правила програмування переходів мовою послідовних інструкцій IL

Програма у даному випадку являє собою список інструкцій тільки тестового типу. Програма для переходу відрізняється від стандартної IL програми наступним:

- відсутня мітка %L;
- відсутні інструкції-дії (дискретні виходи, слова функціональних блоків);
- відсутні переходи на мітки, виклики підпрограм.



Правила програмування переходів мовою структурованого тексту ST

Умова переходу являє собою логічний, арифметичний вираз або їх комбінацію. Така програма відрізняється від стандартної ST програми наступним:

- відсутня мітка %L;
- відсутні арифметичні, умовні операції, циклічні оператори;
- відсутні дії над бітами;
- відсутні переходи на мітки, виклик підпрограм;
- відсутні операції з функціональними блоками.



Умова переходу за часом активності кроку

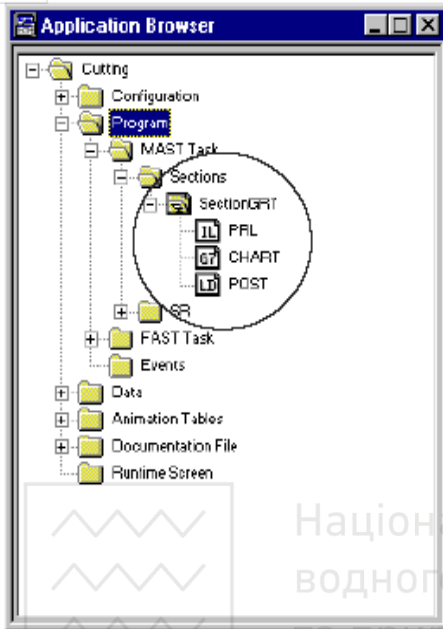
Бувають випадки, коли необхідно, щоб окремий крок виконувався протягом певного інтервалу часу (робота мішалки, випікання хліба, вмикання світла, випалювання цегли тощо).

Приклад:

Необхідно, щоб крок 3 був активним на протязі 15 секунд. Тоді умова переходу від кроку 3 до кроку 4 мовою структурованого тексту матиме вигляд:



2.1.9 Структура Grafcet секції



Програма мовою Grafcet складається з таких послідовно виконуваних секцій:

- початкова частина програми (preprocessing – Prl) – мовою LD, IL або ST;
- основна послідовність (sequential processing – Chart) – мовою Grafcet;
- заключна частина програми (post-processing – Post) – мовою LD, IL або ST.

Grafcet секція завжди належить до MAST задачі.

2.2 Створення другого рівня системи контролю та керування засобами SCADA системи Citect

Розробником SCADA (Supervisory Control and Data Acquisition – Супервізорний контроль та збір даних) системи Citect є австралійська компанія Ci Technologies Pty. Ltd. Особливість компанії Ci Technologies у тому, що вона не тільки розробник Citect, але і системний інтегратор, тому в Citect вкладений накопичений досвід розробки систем автоматизації технологічних процесів.

2.2.1 Основні характеристики Citect

1. Головні особливості:

- Безкоштовне середовище розробки. Гнучка система ліцензування, основана на врахуванні кількості одночасно працюючих комп'ютерів та числа точок введення-виведення.
- Висока продуктивність роботи визначається тим, що Citect побудована на базі мультизадачного ядра реального часу.



На протязі 1 секунди Citect може опитувати 5000 точок у мережевому режимі з декількома станціями.

- Модульна, здатна до масштабування, архітектура клієнт-сервер.

Citect складається з п'яти функціональних модулів:

I/O – сервер введення-виведення. Забезпечує передачу даних між фізичними пристроями введення-виведення та іншими модулями Citect.

Display – клієнт візуалізації. Забезпечує операторський інтерфейс: відображає дані, які надходять від інших модулів Citect і керує виконанням команд оператора.

Alarms – сервер тривоги. Відслідковує дані, порівнює їх з допустимими межами, перевіряє виконання заданих умов і відображає тривоги на відповідному вузлі візуалізації.

Trends – сервер трендів. Збирає і реєструє трендову інформацію, дозволяючи відображати розвиток процесу у реальному часі або у ретроспективі.

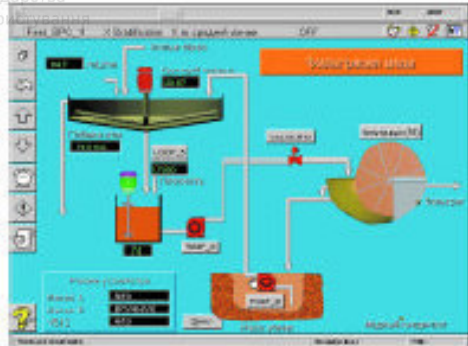
Reports – сервер звітів. Генерує звіти через певні інтервали часу, при виникненні певної події або по запиті оператора.

Кожен функціональний модуль Citect виконується як окрема задача незалежно від того, виконуються модулі на одному комп'ютері або на різних. Тому Citect дозволяє проектувати як прості системи, коли всі модулі працюють на одному комп'ютері, так і складні, в яких функціональні модулі частково чи повністю розподілені по окремих вузлах локальної мережі.

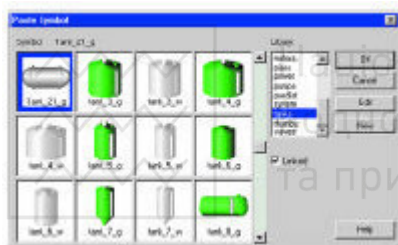
- Можливість резервування функціональних модулів, мережевих з'єднань між вузлами, зв'язку вузлів з ПЛК і самі ПЛК.

2. Можливості розробки графічного інтерфейсу оператора

Головне завдання SCADA- системи – забезпечення інтерфейсу оператора технологічного процесу:



Для розробки опереторського інтерфейсу Citect має бібліотеки простих графічних об'єктів (лінії, фігури, текст, труби тощо) і технологічних апаратів (механізми, резервуари, насоси).

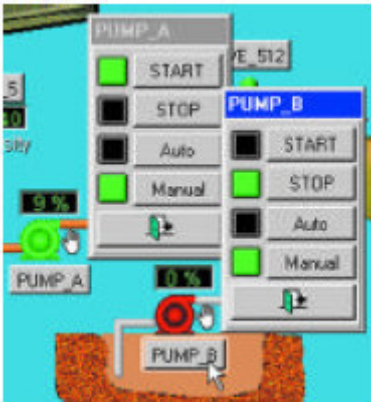


Бібліотеки джинів та суперджинів

Джин – декілька об'єктів, об'єднаних в групу. Джин може використовуватися як єдиний об'єкт: копіюватися, перемішуватися, масштабуватися тощо. З джином зв'язується одна або декілька змінних Citect-програми, причому присвоєння змінних відбувається на етапі розробки програми.

Суперджин – це динамічні сторінки або вікна, але не об'єкти. Змінні, зв'язані з даними сторінками визначаються на стадії виконання програми. Таким чином, один і той самий суперджин може багаторазово активізуватися під час виконання програми, але з різними значеннями змінних, які визначаються на етапі їх активації.

Джин як графічний об'єкт дозволяє вводити різні значення змінних, з іншого боку він може викликати суперджин. Тому бібліотеки джинів можуть містити зв'язані з джинами суперджини.



Створені одного разу з орієнтацією на певні типи проектів бібліотеки джинів-суперджинів спрощують процес розробки мнемосхем технологічного процесу багаторазовим використанням подібних компонентів як у рамках одного проекту, так і в інших майбутніх проектах.

Тривоги (Alarms)

Тривоги – попереджувальні повідомлення. Вони передаються оператору у спеціалізованих вікнах, які називаються сторінками, або через анімовані графічні об'єкти, наприклад, колір об'єкту зміниться з зеленого на червоний при виникненні аварійної ситуації. З кожною тривоگوю можна зв'язати певну дію, яка буде виконуватися при виникненні доної події, наприклад, запустити звуковий файл. Для роботи оператори з тривогами можна створити спеціальні довідкові вікна, які міститимуть опис дій, які слід виконати оператору для виправлення аварійної ситуації. Інформація про тривоги і реакція оператора на них може виводитися як автоматично, так і за запитом оператора.

В Citect розрізняють 5 типів тривог:

- **дискретні тривоги**, які виникають при зміні стану дискретних змінних (з 0 на 1 або навпаки);
- **аналогові тривоги**, які виникають при виході аналогових змінних за вказані верхні та нижні межі, при відхиленні від заданого значення, при відхиленні від заданого значення швидкості зміни змінної;
- **тривоги з мітками часу** дозволяють реєструвати повідомлення з точністю до мілісекунд; мітка часу використовується для вивчення ситуацій, коли одночасно

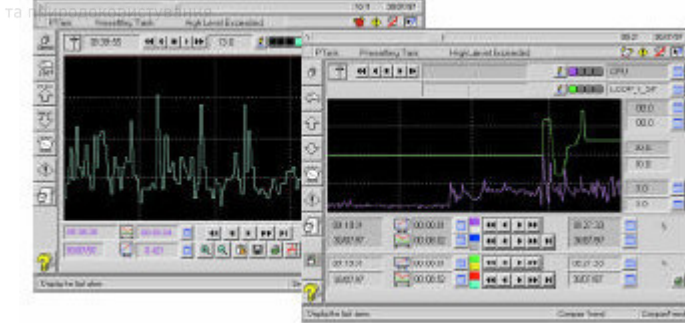


настають декілька тривог і потрібно визначити порядок їх виникнення;

- **складені тривоги**, які виникають в результаті комбінації подій;
- **апаратні (діагностичні) тривоги** – на відміну від вище наведених видів тривог, не можуть бути сконфігуровані користувачем. Citect регулярно запускає діагностичні процедури для перевірки як власного стану, так і стану пристроїв введення-виведення. Відомості про виявлені несправності повідомляються оператору автоматично. Апаратні тривоги завжди реєструються окремо і відображаються на окремому дисплеї тривог.

Тренди (Trends)

Розподілена система побудови трендів Citect може обробляти сотні змінних, не впливаючи на продуктивність або цілісність даних. Реєструватися і виводитися на екран може будь-який виробничий параметр. Тренд в Citect – це зображення зміни значення певної змінної (об'єм кінцевого продукту, рівня, температури тощо) з плином часу, а також графічна оцінка роботи пристрою або ходу процесу. У кожен тренд може відображатися декілька змінних, які виводяться на екран, даючи візуальне уявлення про поведінку процесу в часі. Необхідні вибірки можуть здійснюватися як періодично, так і у момент виникнення у системі певних подій. Частота вибірки може змінюватися від 10 мілісекунд до 24 годин. Citect постачається у комплекті з набором вже готових шаблонів, які забезпечують швидке створення трендів, оснащених необхідними засобами навігації і читання технологічних параметрів. Тренди можуть виводитися у вигляді одинарних, подвійних або спливаючих вікон, але при необхідності користувач може сконфігурувати свій власний тренд і включити в нього необхідні функції.



Звіти (Reports)

Звіт Citect – це документ, який відображає деякі виробничі показники і видається періодично, за запитом або при виникненні певної події (наприклад, при зміні стану змінної, у момент запуску Citect або у вказаний час доби).

Звіти можуть генеруватися у будь-якому зручному для користувача форматі. У нього може входити форматований текст, оперативна і накопичувальна інформація, а також результати математичних обрахунків. Крім того, звіти можуть містити і деякі команди: заміни виробничих параметрів, завантаження інструкцій, виконання діагностики, зміни складу сумішей тощо). Звіти можуть виводитися на екран, роздруковуватися, а також зберігатися на диску для подальшого друку або перегляду. Звіти можна створювати як у текстовому форматі (наприклад, *.rtf), так і у форматі бази даних (*.dbf). Звіт можна обробляти засобами будь-якого текстового редактора і за допомогою SQL- запитів.

Shift Report

Friday, 27
June 1997

Total Milk In:	338150	L
Total Starter In:	3090	L
Total Mid and Starter In:	338250	L
Production Time Forward:	668	mins
Production Time In Divert:	10	mins
Total Production Time:	668	mins
Number of Diverts:	8	diverts
Number of Vats:	22	vats
Total Cheese Weight:	23441.02	Kg
Total Number of Blocks:	1272	blocks
Yield:	0.669	Kg/L

Grower Status

Date: Friday, Jan 9, 1998.

Run No.	Batch No.	Mechart	Processed Modules	AV Turnout
66	13	ADF	8	36.52%
66	13	ADF	2	39.77%
67	14	ADF	6	37.28%
67	14	ADF	2	38.41%
68	15	BBF	10	39.80%
68	15	BBF	4	36.85%



3. Мови програмування

- Мова програмування Cicode

Включає понад 800 функцій, дозволяє створювати прикладну систему будь-якої складності. Мова Cicode підтримує понад 40 операторів для керування тривогами, 25 операторів для роботи з файлами, 20 SQL- функцій, 60 операторів для організації роботи з трендами, оператори для управління комунікаційними портами та багато інших. В Cicode є також функції для інженерної перевірки продуктивності роботи системи. Під час виконання системи може відкриватися спеціальне ядро, яке підтримує команди моніторингу контролерів і мережевих взаємодій, перевірки завантаженості процесора, перевірки помилок тощо.

- Мова програмування Citect VBA (Visual Basic for Application)

Починаючи з версії v. 5.40 Citect має вбудовану підтримку мови програмування Citect VBA. Це багатопоточна мова програмування, займає невеликий (400 К) об'єм пам'яті. Функції Citect VBA можуть бути викликані безпосередньо із Cicode і навпаки, забезпечують повну підтримку ActiveX об'єктів, змінних Citect, змінних тривоги. Citect VBA повністю сумісний з Microsoft VBA 6.0.

4. Платформи

Citect працює як 32-розрядна прикладна програма Windows 98, 2000, NT4, XP. Збір даних, формування тривоги і побудова трендів відбувається одночасно з редагуванням і компіляцією.

2.2.2 Компоненти середовища розробки Citect

- **Citect Explorer** – представлення списку проектів та їх стандартних папок в ієрархічному вигляді з доступом до будь-якого компоненту проекту;
- **Project Editor** (редактор проектів) – середовище розробки, конфігурування і редагування задач, не пов'язаними з графічними сторінками проекту;
- **Graphics Builder** (будівник інтерфейсів) – середовище створення і редагування графічного інтерфейсу;



- **Cicode Editor** (редактор Cicode) – повнофункціональне інтегроване середовище для створення і налагодження програм мовою Cicode.

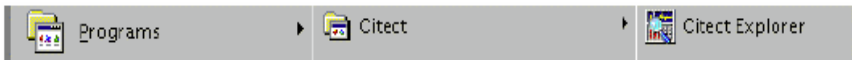
Проект Citect зазвичай складається із ряду сторінок (Pages), які виводяться на екран комп'ютера. Ці графічні сторінки забезпечують “вікно в процес”. За допомогою графічних сторінок відбувається процес взаємодії оператора з системою керування, у тому числі сприйняття даних і введення керуючих впливів. Важливо створити графічні сторінки таким чином, щоб вони охоплювали весь технологічний процес і надавали оператору всю необхідну для керування інформацію. Причому процес створення графічних сторінок проекту повинен бути максимально спрощений, і розробника необхідно забезпечити повним та зручним інструментарієм.

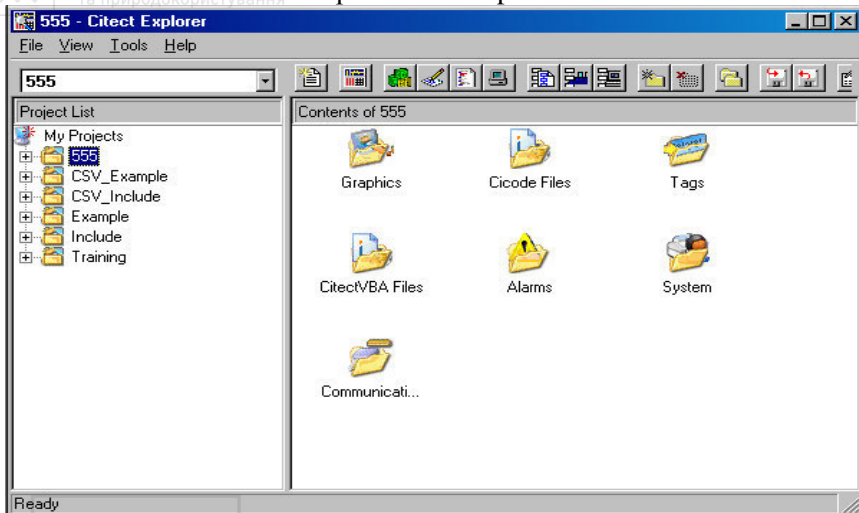
Citect надає розробнику наступні можливості:

- шаблони більшості типів найчастіше використовуваних сторінок (вікон);
- інструментарій для створення і динамізації графічних об'єктів;
- спеціальний редактор – Bitmap Editor – для створення точкових зображень;
- бібліотеку статичних об'єктів – Library Objects;
- бібліотеку джинів та суперджинів.

Citect Explorer

- Для запуску Citect Explorer необхідно вибрати у меню ПУСК





- Для виклику довідки необхідно вибрати пункт меню **Help\Explorer help**.

Наперед встановлені проекти (Include Projects)

Дані проекти встановлюються разом з системою Citect і містять елементи, які можна використовувати для власних проектів. Ці елементи включають: опис клавіш клавіатури (keyboard definitions), опис шрифтів (font definitions), джини (genies), суперджини (super genies) і бібліотеки символів (symbol libraries). Ви можете як показувати, так і ховати меню Include project у Списку Проектів (Project List) за допомогою меню **View\Show Include Project**. Не слід вносити зміни в наперед встановлені проекти, оскільки при встановленні новішої версії Citect ваші зміни будуть втрачені. Краще створювати свої власні елементи, які можуть зберігатися при встановленні іншої версії Citect.

Редактор проекту (Project Editor)

Редактор проекту автоматично запускається і згортається при запуску Citect Explorer.




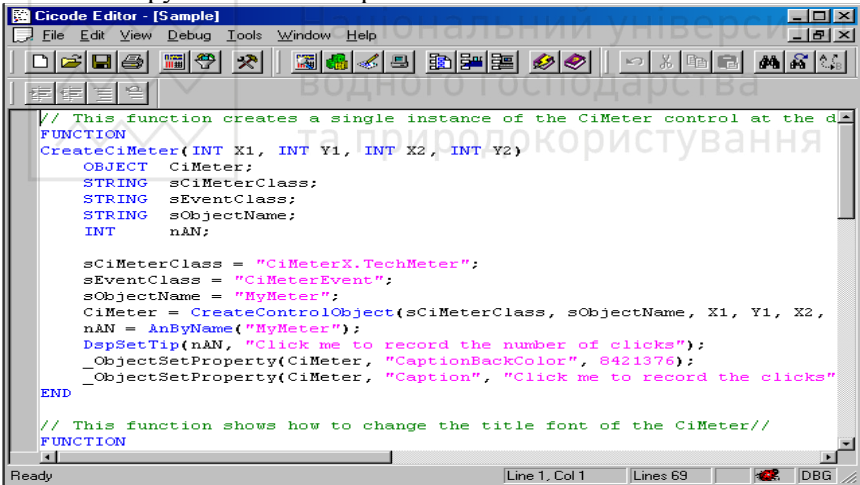
Будівник графіків (Graphics Builder)

Будівник графіків автоматично запускається і згортається при запуску Citect Explorer.



Редактор Cicode (Cicode Editor)

- Для запуску Cicode Editor потрібно натиснути  на панелі інструментів або вибрати меню Tools\Cicode Editir.



2.2.3 Керування проектами Citect

Citect Explorer – це прикладна програма Citect, яка дозволяє організувати проекти, налагоджувати і запускати їх на виконання. Citect Explorer вирішує такі основні завдання, як створення (creating), видалення (deleting), створення резервних копій (backing up) і відновлення (restoring) проектів.



Створення нового проекту

Перше, що необхідно зробити при використанні Citect, це створити новий проект, в якому будемо зберігати всю інформацію про технологічний процес. Кожен проект займає свій підкаталог у каталозі \User. Підкаталог для кожного проекту створюється у момент створення проекту і має теж ім'я, що і проект.

Увага! Citect не підтримує довгі імена каталогів. Якщо назва вашого проекту довше за 8 символів, тільки перші 8 символів буде запам'ятовано.

Майстер налаштування Комп'ютера (Computer Setup Wizard)

Майстер налаштування комп'ютера дозволяє швидко встановлювати і налаштовувати комп'ютер для роботи із Citect. Завжди, коли створюється новий проект у Citect Explorer, повинен використовуватися і майстер комп'ютера.

Комп'ютер може встановлюватися як виконуючий Citect- проект незалежно, так і як комп'ютер у мережі, виконуючий частину Citect- проекту.

Stand-alone

(той, що знаходиться окремо)

Комп'ютер працює окремо:

Сервер (Server) або Спостерігаючий Клієнт (Display Client).

Network

(мережевий)

Комп'ютер працює як:

Сервер і Клієнт (Server and Display Client),
Спостерігаючий Клієнт (Display Client),
Керуючий Клієнт (Manager Client).

Видалення Проектів (Deleting a Project)

Коли проект видаляється із Citect Explorer, усі зв'язані з ним файли і директорія проекту також видаляються.

Для видалення проекту необхідно, знаходячись в Citect Explorer, натиснути на проект, який ви хочете видалити у Списку Проектів (Project List), і вибрати меню **File \ Delete Project**. Або натиснути правою клавішею миші на піктограмі проекту і вибрати **Delete Project** у контекстному меню.



Створення резервних копій і відновлення Проектів (Backing Up & Restoring a Project)

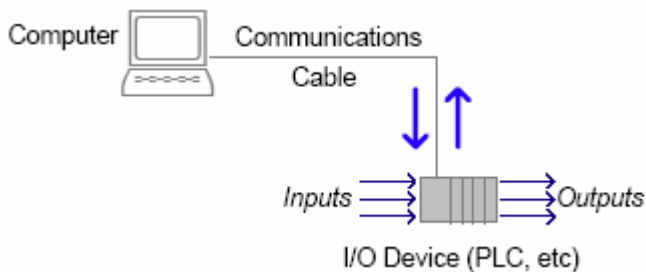
Для Citect-проектів можливим є створення резервних копій у вигляді архівних файлів, які займають набагато менший простір на диску, ніж вихідні проекти. Резервне копіювання повинне виконуватися на випадок, якщо файли будуть помилково видалені або пошкоджені. Так само дуже важливо вести історію резервних копій, щоб завжди була можливість повернутися до попередньої версії проекту, особливо якщо зміни були внесені у вже працюючу систему.

Увага! Починаючи з версії Citect v.5.10.0, введено новий формат файлів резервних копій. Файли більше не називаються StBackup.001 – тепер їм можна присвоїти будь-яке ім'я. За замовчуванням файлам задається розширення **.ctz**.

2.2.4 Встановлення зв'язків між Citect і пристроями введення-виведення

Citect може зв'язуватися з різноманітними пристроями введення-виведення (I/O Devices). Наприклад, з програмованими логічними контролерами (ПЛК) (Programmable Logic Controllers – PLC), пристроями зчитування штрих-коду, лабораторними аналізаторами, віддаленими терміналами (RTU), розподіленими керуючими системами (DCS).

Citect безпосередньо зв'язується з пристроями введення-виведення і дозволяє обмінюватися між ними і керуючою системою.





Майстер Швидкого Встановлення Зв'язку (Express Communications Wizard)

Даний майстер допомагає налаштувати зв'язок з існуючими і новими пристроями введення-виведення.

У будь-якому Citect- проекті є як мінімум один сервер введення-виведення (I/O Server), призначений для зв'язку з пристроями введення-виведення. Кожен пристрій введення-виведення повинен бути сконфігурований у системі Citect. При цьому необхідно задати протокол зв'язку, лату введення-виведення (interface board) і порт зв'язку.

Майстер Express Communications Wizard послідовно запитує:

- ім'я сервера введення-виведення (**I/O Server**);
- інтерфейсну плату в комп'ютері (**Boards**);
- порт зв'язку на платі (**Ports**);
- пристрій введення-виведення, під'єднаний до порту (**I/O Devices**).

Примітка. Кожен комп'ютер у системі Citect може працювати тільки як один Citect- сервер введення-виведення. якщо проект буде працювати у мережі Citect- комп'ютерів, і пристрої введення-виведення будуть підключені до більш, ніж одного комп'ютера, то і серверів введення-виведення повинно бути стільки ж, скільки комп'ютерів з пристроями введення-виведення.

Змінні (Variable Tags)

Змінні визначають дані, які передаються між пристроями введення-виведення і сервером введення-виведення Citect. Кожна змінна має унікальне ім'я, визначає тип даних, адресу і пристрій введення-виведення.

Запис адреси змінної залежить від пристрою введення-виведення.

Так для **Disk** і **Memory PLC** маємо:

I/O Devices Form

Name	Будь-яке ім'я
Number	Повний мережевий номер пристрою введення-виведення
Address	<File List> для DiskDrv або пусте поле для Memory

Protocol	Generic
Port name	DiskDrv або Memory

Generic Variables

Name	Short Name	Type	Size (in bits)	#
Digital#	D#	Digital	1	0-64000
Int#	I#	Int	16	0-16375
Long#	L#	Long	32	0-16375
Real#	R#	Real	32	0-16375
String#	S#	String	1024 (128 bytes)	0-510

Files

- **Generic.dbf** – вбудована база даних змінних для протоколу Generic (вбудована база даних Citect).

Для визначення змінної, знаходячись Citect Explorer, виберіть

проект, відкрийте папку **Tags** і двічі натисніть **Variable Tags**.
Або, знаходячись у Project Editor, виберіть меню **Tags \ Variable Tags**.

Або, знаходячись у Project Editor, натисніть на піктограму **Variable Tags** на панелі інструментів.

Форми представлення інформації в Citect (Citect Forms)

Усі форми представлення інформації в Citect мають один і той самий стандартний інтерфейс. Його кнопки наступні:

Add (додати)	Додає інформацію, яка відображається на даний час, у базу даних у вигляді нового запису.
Replace (замінити)	Замінює поточний запис на відображувану інформацію.
Delete (видалити)	Видаляє поточний запис.
Help (довідка)	Відкриває довідкову систему (Citect Help Topics) за параметрами даної форми.



Записи розташовуються у базі даних у порядку їх введення.

Поради:

Для пошуку запису виберіть пункт меню **Edit \ Find...** . При цьому буде проведений пошук у поточній формі. Форма буде показувати тепер тільки записи, які задовольняють заданій умові. Використовуйте лінійку прокрутки, якщо записів багато.

Перевірка зв'язку (Test Communications)

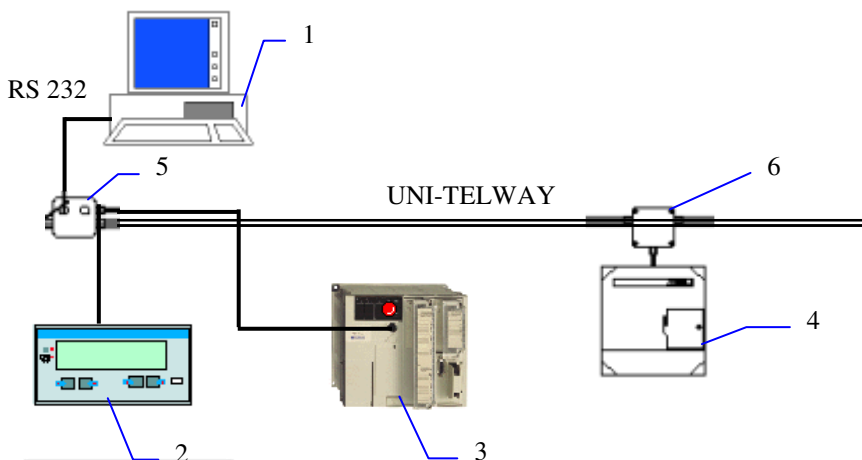
Дуже важливо протестувати зв'язок перед розробкою нового проекту. Без надійного зв'язку Citect- проект працюватиме неефективно. Перевірка зв'язку також допомагає переконатися, що використовується правильний формат адрес для змінних проекту.

4. Програма роботи

1. Навчитися програмувати ПЛК Schneider Micro TSX 37-22 мовою Grafset у середовищі програмування PL7 PRO.
2. Написати програму для контролера Schneider Micro TSX 37-22 для автоматизації технологічного процесу приготування суміші з трьох компонентів.
3. Створити другий рівень системи контролю і керування технологічним процесом засобами програмного забезпечення PL7 PRO.
4. Створити другий рівень системи контролю і керування технологічним процесом засобами SCADA- системи Citect.
5. Перевірити роботу програми.

5. Опис лабораторного обладнання

1. Персональний комп'ютер.
2. Операційна система Windows.
3. Лабораторний стенд з ПЛК Schneider Micro TSX 37-22.
4. Середовище програмування ПЛК Schneider Micro TSX 37-22 PL7 PRO, середовище програмування операторської панелі Magelis XBT-L-1000, SCADA система Citect.



1- АРМ оператора; 2 - операторська панель Magelis XBT P 012010;
3 - ПЛК Schneider Micro TSX 37-22; 4 - частотний перетворювач
Altivar 58; 5 - адаптер відгалуження TSX P ACC01; 6 -
розгалужувач TSX SCA 62.

6. Порядок виконання роботи

1. Під'єднати контролер Schneider Micro TSX 37-22 до COM-порту комп'ютера за допомогою комунікаційного кабелю TSX PCX 1031. Ввімкнути живлення стенду.
2. Запустити середовище PL7 PRO.
3. Написати програму для контролера Schneider Micro TSX 37-22 для автоматизації технологічного процесу приготування суміші з трьох компонентів мовою програмування Grafset.
4. Створити анімаційну таблицю для відслідковування значень використаних у програмі змінних. Наряду з анімаційною таблицею використати операторську панель Magelis.
5. Створити вікно супервізора (Run-Time screen) засобами програмного забезпечення PL7 PRO, в якому намалювати ФСА технологічного процесу.
6. Записати програму у контролер. Для цього виконати наступні дії:



- виконати команду **Conect**;
 - вибрати напрям запису програми PC -> PLC;
 - перевести контролер у режим **RUN**.
7. Провести моніторинг роботи програми за допомогою анімаційної таблиці, вікна супервізора.
8. Створити вікно супервізора засобами програмного забезпечення Citect, в якому намалювати ФСА технологічного процесу, та провести моніторинг роботи програми. Для цього виконати наступні дії:



Крок 1

Розглянемо приклад проекту, який постачається з Citect.

1.1 Запустіть Citect Explorer.

- В меню ПУСК Windows виберіть **Программы \ Citect \ Citect Explorer**.



1.2 Виберіть приклад проекту (**Example project**) і розгляньте його компоненти.

- В екрані Citect Explorer натисніть на піктограмі **Example** в **Project List** для того, щоб зробити його активним проектом.
- Натисніть символ “+” для розгортання папки проекту і переміщуйтесь по компонентах даного проекту.

1.3 Натискайте на піктограми для перемикання між редакторами, які входять до складу Citect.

1.4 В екрані Citect Explorer знайдіть меню інструментів **Tools menu** і переключіться в інший додаток Citect.

1.5 Запустіть проект Example.

- В Citect Explorer виберіть проект **Example**.
- Натисніть на кнопці Запуск Проекту (**Run Project**)  на панелі інструментів.
- Переміщуйтесь у проєкті, натискаючи відповідні кнопки для зміни сторінок.
- Для закриття проєкту натисніть кнопку Закрити (**Close Box**) .



Крок 2

Створення нового проекту.

2.1 Створіть новий проєкт з назвою **Training**.



Знаходячись в Citect Explorer, натисніть кнопку **New**, або

- виберіть меню **File \ New Project....**, або натисніть правою клавішею миші на **Project List** і виберіть **New Project...**

З'явиться наступне вікно:

Заповніть усі опції вікна на натисніть OK.

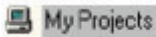
Після цього Citect створить базу даних конфігурації і побудує дерево каталогів для вашого проекту. Знайдіть новий запис з іменем **Training** в **Project List** – цей запис містить всю інформацію про конфігурацію вашого проекту.



Крок 3 Налаштування комп'ютера.

Запустіть **Computer Setup Wizard**. Для цього, знаходячись в Citect Explorer, натисніть кнопку **My Projects**

3.1



а далі кнопку **Computer Setup** або виберіть меню **Tools \ Computer Setup**.



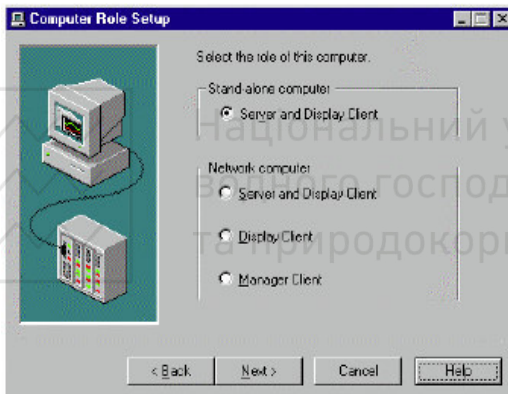


З'явиться наступне вікно:



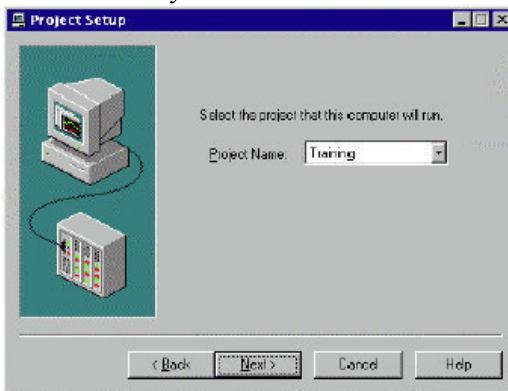
Виберіть **Express Setup** і натисніть Next.

З'явиться вікно:



Виберіть **Server and Display Client** для **Stand-alone** computer і натисніть Next.

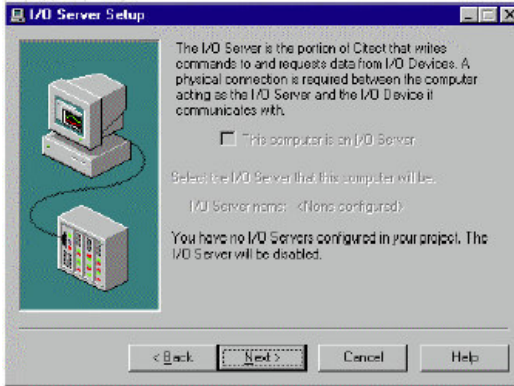
З'явиться наступне вікно:



Виберіть проект із випадального меню **Project Name**. Після цього натисніть Next.



З'явиться наступне вікно:



Даний діалог повідомляє, що конфігурація введення-виведення відсутня. Це пов'язано з тим, що ми ще не встановили зв'язки (communications). Натисніть Next для продовження роботи майстра.

З'явиться наступне вікно:




Натисніть Finish для збереження усіх змін і виходу із Майстра Citect Computer Setup.



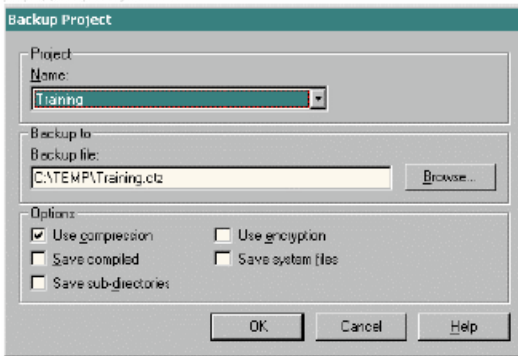
Крок 4

Створення резервної копії проекту, видалення та відновлення проекту.

- 4.1 Створення резервної копії проекту **“Training”**. Знаходячись в Citect Explorer, виділіть піктограму **Training**.
- Натисніть кнопку **Backup**  або виберіть пункт меню **Tools \ Backup...**



З'явиться наступне вікно:

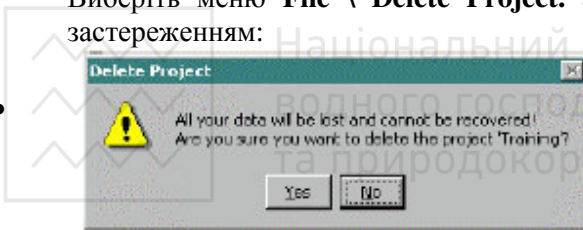


Заповніть його поля і натисніть ОК.

4.2 Видалення проекту “Training”.

Знаходячись в Citect Explorer, виділіть піктограму **Training** в Project List.

Виберіть меню **File \ Delete Project**. З'явиться діалог із застереженням:



Натисніть **Yes** для підтвердження видалення проекту.

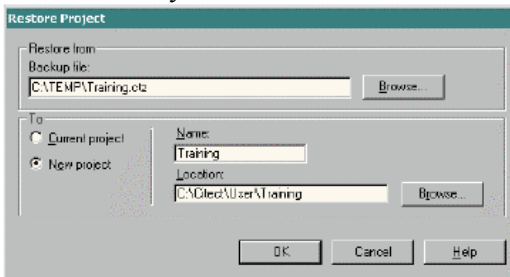
4.3 Відновлення проекту “Training”.

Знаходячись в Citect Explorer, натисніть піктограму Restore



або виберіть меню **Tools \ Restore...**

З'явиться наступний діалог:



При необхідності використовуйте кнопку **Browse...** для вибору файлу, який містить резервну копію. Далі виберіть опцію **New project**.



Обережно! Якщо при відновленні ви обрали опцію **Current project**, то проект, вибраний в Citect Explorer, буде перезаписаний. Ви можете відновлювати будь-які проекти, використовуючи опцію **New project**, і присвоювати їм або нові імена, або ті самі, що були у них раніше.

- Натисніть ОК для відновлення проекту “Training”.



Крок 5

Встановлення зв'язку Citect- проекту з обладнанням.

5.1 Знаходячись в Citect Explorer, виділіть проект **Training**. Двічі натисніть на папці Зв'язок (Communications). Двічі

- натисніть на **Express I/O Device Setup** виберіть меню **Tools \ Computer Setup**.



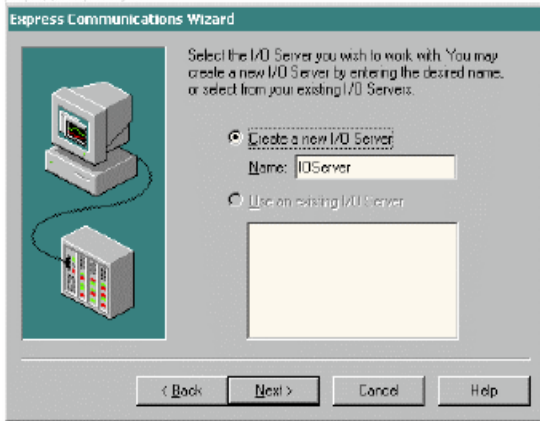
З'явиться наступний діалог:



Натисніть **Next** для продовження.

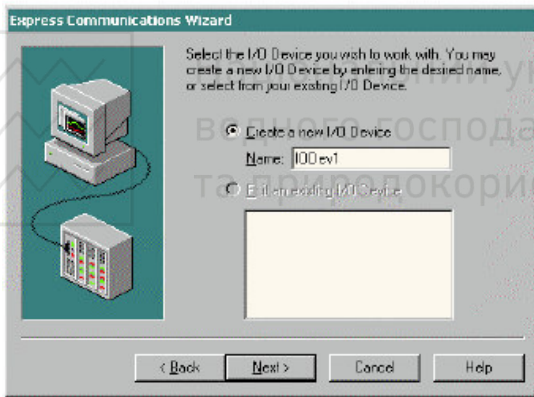


З'явиться наступний діалог:



Залиште ім'я сервера **“I/OServer”** і натисніть **Next** для продовження.

З'явиться наступне вікно:



Виберіть **Create a new I/O Device** і введіть ім'я **“I/ODev1”**. Натисніть **Next** для продовження.



Якщо ваш комп'ютер у класі під'єднаний до пристрою введення-виведення, виберіть **External I/O Device**, в іншому випадку – **Disk I/O Device**.

Поради:

Disk I/O Device зберігає дані у вигляді файлу у певному місці вінчестера. При перезавантаженні Citect дані залишаються доступними. **Disk I/O Device** може використовуватися і іншими комп'ютерами через локальну мережу. Дані, записані у **Memory I/O Device**, втрачаються при перезапуску комп'ютера.

Натискаючи Next, ви продовжите процес встановлення зв'язків під керівництвом Майстра. При цьому усі пункти наступних діалогів залежатимуть від пристрою, з яким встановлюється зв'язок.

Після завершення роботи майстра Express Communications Wizard виконана частина роботи над проектом. Ця частина може бути відкрита із Citect Explorer, у папці **Communications** або із Citect Project Editor через меню **Communications**.

Для того, щоб продивитися усі зв'язки, необхідно двічі натиснути на **I/O Servers, Boards, Ports, I/O Devices**.

Для того, щоб дізнатися призначення кожного поля діалогу, натисніть кнопку **Help**.



Крок 6

Створення змінних та мнемосхеми проекту.
Компіляція і запуск проекту. Перевірка



- 6.1 Знаходячись в Citect Explorer, виберіть проект **Training** і відкрийте папку **Tags**. Потім двічі натисніть на **Variable Tags**.

З'явиться наступне вікно:

Variable Tag Name: Data Type:

I/O Device Name: Address:

Raw Zero Scale: Raw Full Scale:

Eng Zero Scale: Eng Full Scale:

Eng Units: Format:

Comment:

Record: 1

-

Заповніть поля форми наступним чином (при використанні пристрою введення-виведення **Disk**):

Variable Tag Name	Test
Data Type	Digital
I/O Device Name	I/ODev1
Address	D0

Натисніть кнопку **Add**.

- 6.2 Створіть нову графічну сторінку для відображення значення змінної **Test**.

- Знаходячись в Citect Explorer, виберіть проект **Training**, відкрийте папку **Graphics**, а потім **Pages**.

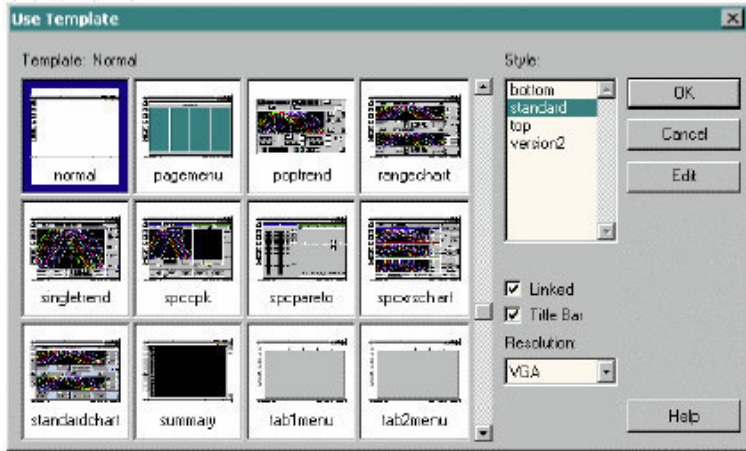
-

Двічі натисніть на кнопці **Create a new page**





З'явиться наступне вікно:



Для створення нової сторінки залиште опцію, яка показана на рисунку. Натисніть ОК.

Для того, щоб намалювати кнопку на вашій сторінці,

виберіть інструмент **Button**  на панелі інструментів.

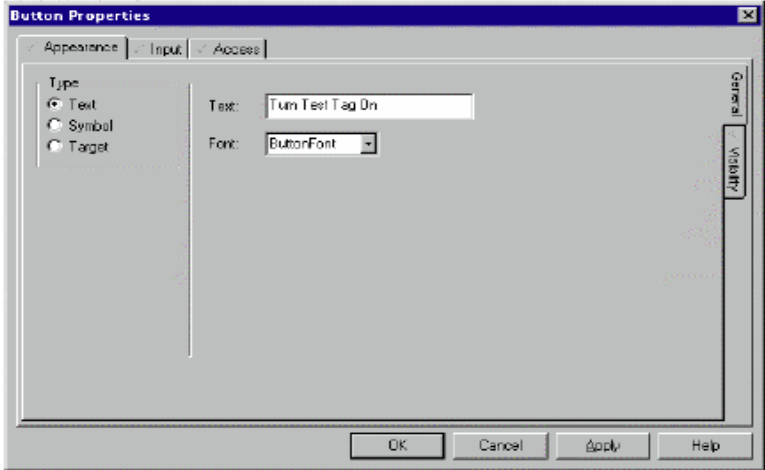
Натисніть і притримуйте ліву клавішу миші. Відведіть мишу у бік для створення кнопки потрібного розміру і відпустіть

клавішу:



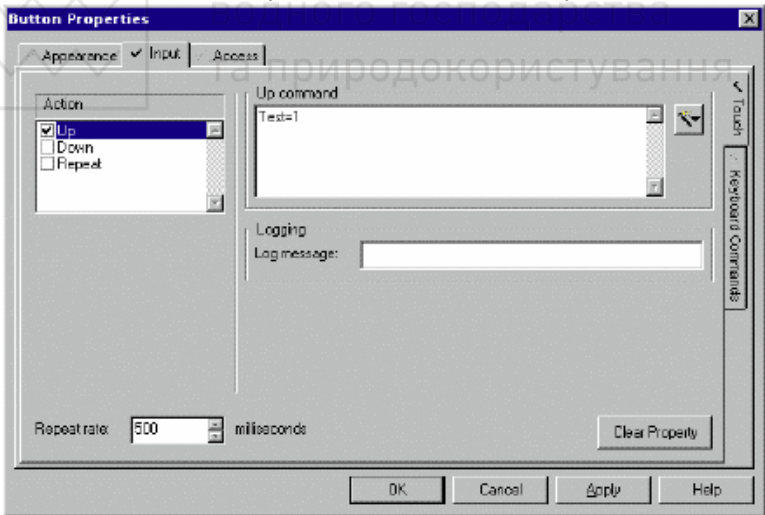


З'явиться наступне вікно:



Заповніть діалогове вікно, як показано, але не натискайте ОК.


Натисніть на закладку **Input**. З'явиться наступний діалог:

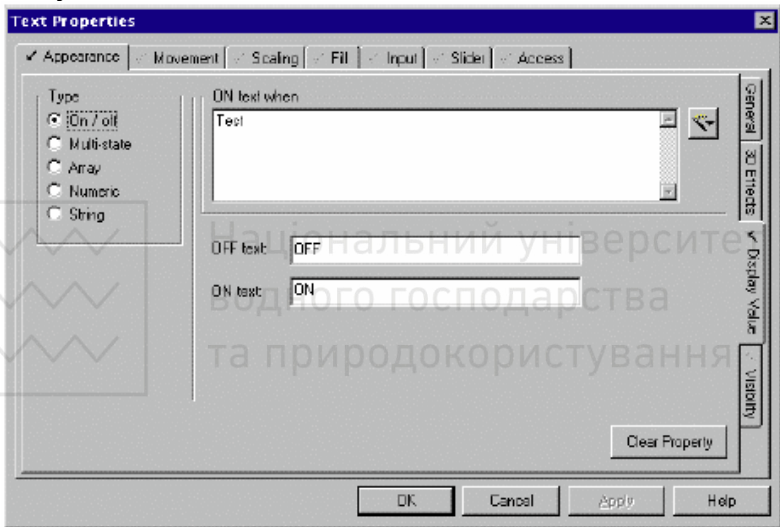


У списку **Action** натисніть на прямокутник **Up** і у полі **Up command** наберіть **Test=1**. Натисніть ОК.

- Таким самим способом намалюйте другу кнопку і опишіть її, набрав у полі **Text** наступний рядок: **Turning Test Tag off**, а у полі **Up** – рядок **Test=0**.



- Виберіть інструмент **Text**  на панелі інструментів і пересуньте курсор у точку поряд із щойно намальованою кнопкою. Тут ви зможете ввести текст, який відображатиме стан змінної **Test**.
- Натисніть на клавішу “Пробіл”, а потім натисніть ліву клавішу миші. З’явиться відповідне вікно.
Натисніть на закладку **Appearance** у верхній частині вікна і на закладці **Display Value** у правій частині вікна. З’явиться наступне вікно:

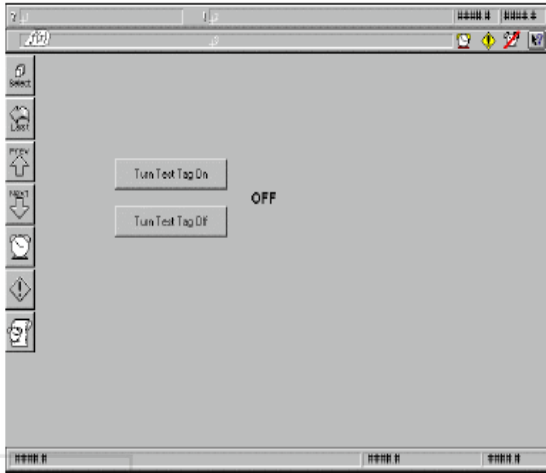



У полі **ON Text when** наберіть **Test** (за замовчуванням значення поля **Type** вже встановлено **On/Off**), у полі **OFF text** введіть **OFF**, у полі **ON Text** введіть **ON**.



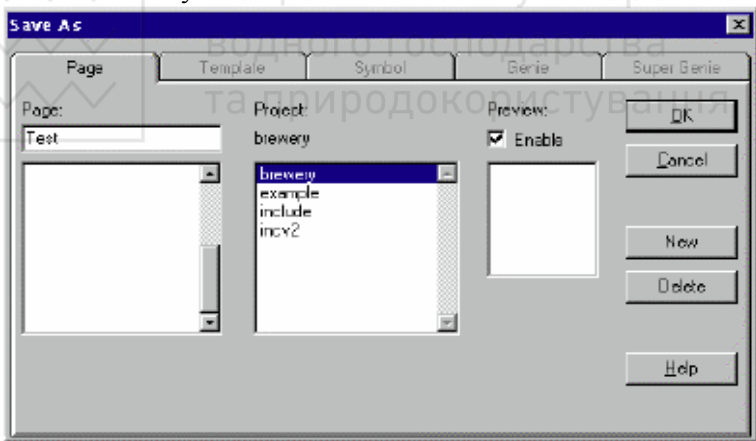
всередині господарства
та природодобувної галузі

Натисніть **ОК**. Тепер сторінка повинна виглядати приблизно таким чином:



Натисніть кнопку **Save**  на панелі інструментів для збереження сторінки.

З'явиться наступне вікно:






У списку проектів (**Project**) виберіть **Training**. Далі у полі введіть **TestP**. Натисніть **ОК**.

6.3

Запустіть Майстер Налаштування комп'ютера (**Computer Setup Wizard**).

- Знаходячись в Citect Explorer, натисніть на **My Projects**, а потім двічі натисніть на **Computer Setup**.
- Виберіть режим **Express Setup**.
- Виберіть **Stand-alone Computer** (Server and Display Client).



- Виберіть із випадуючого меню проект **Training**.
Якщо у проекті Training заданий тільки один сервер введення-виведення, то він буде вибраний автоматично. Натисніть ОК.
- Натисніть **Finish** для завершення роботи майстра і збереження налаштувань.
- 6.4 Компіляція і запуск проекту. Перевірка функціонування зв'язку.
Знаходячись у Project Editor, натисніть кнопку **Run Project**
. При цьому проект буде скомпільовано і запущено на виконання.
Примітка:
Коли ви компілюєте проект, Citect перевіряє наявність помилок. Якщо звіт компілятора показує наявність помилок, натисніть кнопку **GoTo** для переходу до місця виявлення помилки. Після виправлення усіх помилок збережіть сторінку і натисніть кнопку **Run Project**  ще раз.
Зараз проект запуститься на виконання і буде працювати до закриття, реагуючи на дані, що вводяться з клавіатури і миші.
- Перевірте роботу створеної графічної сторінки, змінюючи значення змінної Test за допомогою кнопок. Для завершення роботи проекту натисніть кнопку  у правому верхньому кутку вікна проекту.
- 6.5 Створення повної бази даних змінних для автоматизації технологічного процесу приготування суміші.
Тепер, коли ми встановили і протестували роботу проекта, можна додати інші, необхідні для функціонування технологічного процесу, змінні. Більшість налаштувань однакові для всіх змінних. Якщо необхідно додати змінну, аналогічну до попередньої, змініть потрібні поля і натисніть **Add** (Додати). Для виправлення помилок внесіть зміни і натисніть **Replace** (Замінити).



У багатьох випадках, особливо якщо число змінних досить велике, додавання нових змінних може бути занадто трудомною роботою. Оскільки всі вікна використовують дані формату DBF, то можливе безпосереднє редагування баз даних з використанням таких програм як Microsoft Excel. Файл з базою даних змінних **Variable.dbf** знаходиться в директорії **\Citect\User\.**

Примітка:

Редагування файлів поза Citect може призвести до того, що файли не будуть читатися Citect- програмами. Завжди робіть резервну копію файла Variable.dbf (або іншого редагованого файлу) перед його редагуванням. Тоді у випадку виникнення проблем файл можна буде відновити з резервної копії.

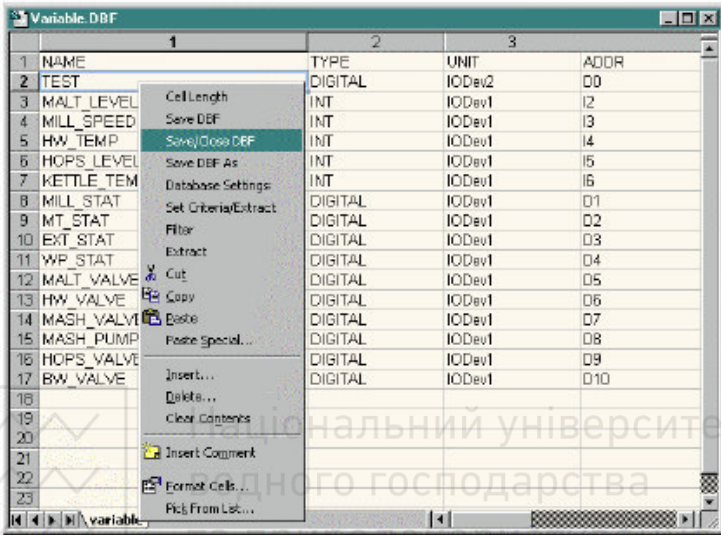
- Зробіть резервну копію файла **Variable.dbf**. Запустіть Excel і відкрийте файл **Variable.dbf** у директорії вашого проекту.
- Відкрийте файл **Save_dbf.xls** із каталога ...Citect\Bin. Він завантажує макрос Excel для збереження файлів у потрібному форматі.
- Додайте або змініть змінні у файлі. При цьому є можливість користуватися інструментами **Копіювати**, **Вирізати**, **Вставити**.

Поради:


- В Excel, захопивши правий нижній кут чарунки мишею, можна “розтягувати” дані на діапазон чарунок. При цьому номери будуть автоматично зростати. Утримуючи натиснутою клавішу **Ctrl**, можна розтягувати дані без нарощування номерів.



Після завершення редагування розташуйте курсор миші поверх будь-якої чарунки і натисніть правою клавішею миші. Це активізує контекстне меню. Виберіть опцію **Save / Close DBF** для збереження і закриття файла.



Закрийте Excel.

- Поверніться в Citect Project Editor і виберіть меню **File \ Pack**. Це призведе до переіндексації бази даних.
 - Відкрийте форму **Variable Tags** і переконайтеся, що змінні було успішно відредаговано.
- 6.6** Розробка мнемосхеми технологічного процесу приготування суміші
- Аналогічно вище наведеному пункту 6.2 створіть нову графічну сторінку для розробки мнемосхеми технологічного процесу приготування суміші.
 - Використовуючи стандартні бібліотеки намалюйте мнемосхему процесу.
 - Скористайтеся формами відображення змінних (зокрема трендами) для проведення моніторингу за ходом процесу.
 - Запустіть проект на виконання. Для цього натисніть кнопку **Run Project** .

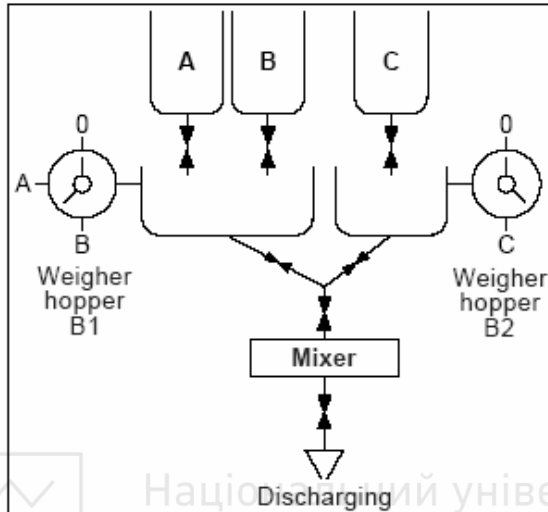


Рис. 5.1 Технологічна схема змішування трьох компонентів

Продукти А і В надходять у бункер В1 і зважуються. Продукт С зважується у бункері В2. Із бункерів В1 і В2 речовини надходять у змішувач, де змішуються на протязі часу, встановленого користувачем. Як тільки час змішування спливає, змішувач зупиняється. Утворена суміш вивантажується, якщо надходить дозволяючий сигнал від оператора.

Функціональний аналіз процесу

Даний процес можна розбити на такі частини:

- зважування трьох компонентів;
- наповнення мішалки;
- змішування трьох компонентів у мішалці;
- відвантажування утвореної суміші.

На рис. 5.2 представлені основні етапи процесу.

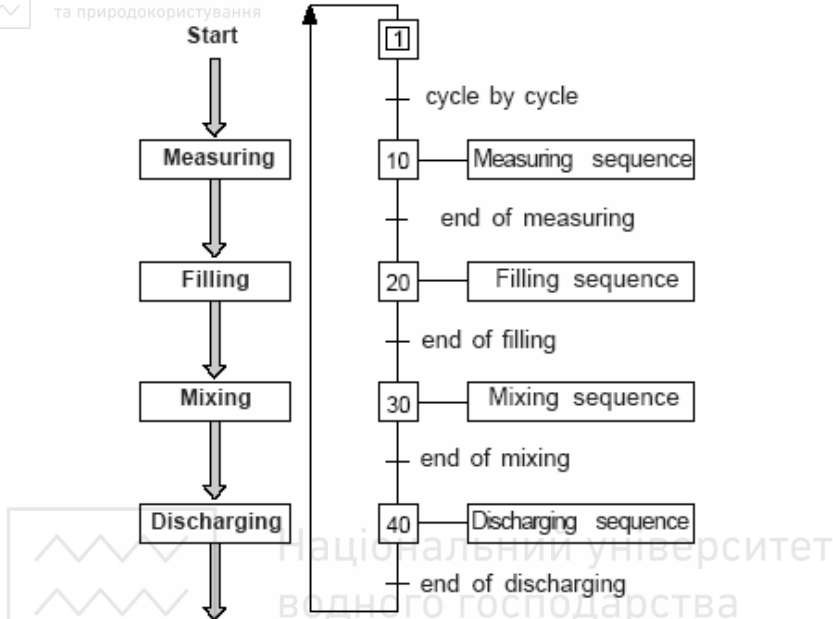


Рис. 5.2 Основні етапи процесу приготування суміші із трьох компонентів

При проведенні більш детального аналізу кожен основний етап можна теж розбити на стадії. Після цього можна приступати до розробки програмного забезпечення.

7. Контрольні запитання

1. Назвіть основні принципи написання програм мовою Grafset.
2. Які є графічні елементи для написання програм мовою Grafset?
3. Які є види і якими мовами можна запрограмувати дії для кроків мови Grafset? Поясніть їх призначення.
4. Які правила програмування переходів між кроками у мові Grafset?
5. З яких частин складається програмна секція мовою Grafset? Поясніть призначення кожної з них.
6. Для автоматизації яких процесів найкраще використовувати мову Grafset при розробці програмного забезпечення для ПЛК?



7. Назвіть основні характеристики SCADA- системи Citect.
8. Перерахуйте та дайте характеристику основним стадіям розробки нового проекту в SCADA- системі Citect.
9. Яким чином налаштувати зв'язок між пристроями введення-виведення і SCADA- системою Citect?
10. Якими способами можна відредагувати базу даних змінних у SCADA- системі Citect? Дайте характеристику кожному способу.
11. Які засоби візуалізації ходу технологічного процесу є у SCADA- системі Citect?
12. Яким чином можна створити мнемосхему технологічного процесу у SCADA- системі Citect?
13. Яким чином налаштувати об'єкт джин у SCADA- системі Citect?
14. Яким чином налаштувати об'єкт суперджин у SCADA- системі Citect?
15. Яким чином відобразити значення технологічного параметру у SCADA- системі Citect?
16. Яким чином створити кнопку та присвоїти їй виконання певної дії у SCADA- системі Citect?
17. Яким чином створити тренд та вивести на нього значення технологічного параметру у SCADA- системі Citect?