МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

МЕТОДИЧНІ ВКАЗІВКИ
до лабораторних робіт
з дисципліни
"Мікропроцесорні системи керування технологічними процесами"

для студентів спеціальності 172 "Електронні комунікації та радіотехніка" (освітні програми "Радіоелектронні апарати та засоби", "Інтелектуальні технології мікросистемної радіоелектронної техніки") усіх форм навчання

2024

Методичні вказівки до лабораторних робіт з дисципліни "Мікропроцесорні системи керування технологічними процесами" для студентів спеціальності 172 "Електронні комунікації та радіотехніка" (освітні програми "Радіоелектронні апарати та засоби", "Інтелектуальні технології мікросистемної радіоелектронної техніки") усіх форм навчання / Уклад. : Олександр МАЛИЙ, Олександр ПІРОЖЕНКО, Сергій ГАРАЧУК, Запоріжжя : НУЗП, 2024. 36 с.

Укладачі: Олександр МАЛИЙ, к.т.н., доцент, зав. каф. ІТЕЗ;
Олександр ПІРОЖЕНКО, ст. викладач каф. ІТЕЗ;
Сергій ГАРАЧУК, ст. викладач каф. ІТЕЗ.

Рецензент: Наталія ФУРМАНОВА, к.т.н., доцент, декан ФІБЕК

Відповідальний за випуск:
Олександр МАЛИЙ, к.т.н., доцент, зав. каф. ІТЕЗ

Затверджено
на засіданні кафедри ІТЕЗ
протокол № 1 від 10.09.24 р.

Рекомендовано до видання
НМК ФІБЕК
протокол № 2 від 19.09.24 р.

ЗМІСТ

1 ЛАБОРАТОРНА РОБОТА 1. СТВОРЕННЯ ПРОСТОГО ПРОЄКТУ КЕРУВАННЯ ЛАМПАМИ

Мета роботи: ознайомитись з інтегрованим середовищем розробки ISPSoft та програмою для створення людино-машинного інтерфейсу DOPSoft, отримати базові навички програмування промислових контролерів на мові Ladder Diagram.

1.1 Теоретична частина

1.1.1 Створення віртуального контролеру

Для симуляції роботи промислового логічного контролера (ПЛК) використовується програма COMMGR (рис. 1.1). Щоб створити новий драйвер відкриваємо програму COMMGR через панель задач та натискаємо «**Add**» (додати). У новому вікні обираємо назву віртуального приладу та обираємо серію AS200. Створений віртуальний драйвер буде використовуватись для всіх наступних лабораторних робіт.

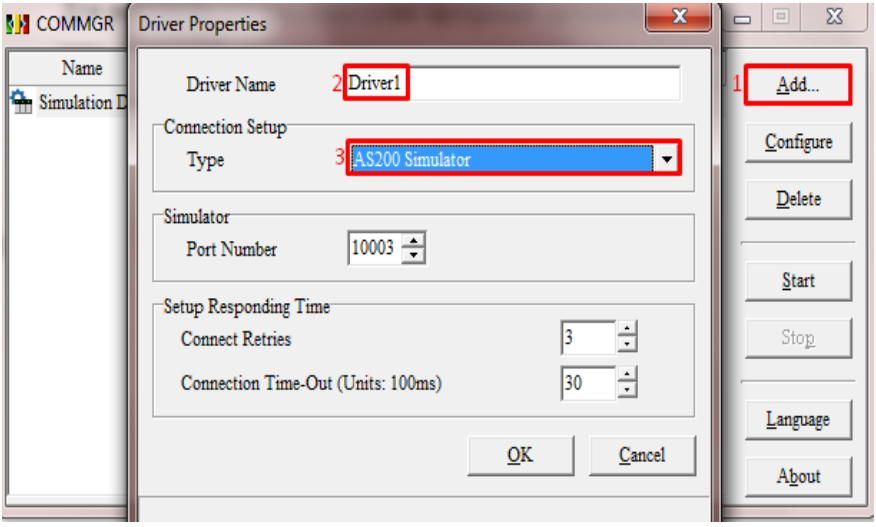

Рисунок 1.1 – Створення віртуального контролера

1.1.2 Створення проєкту у ISPSoft

ISPSoft – це інтегроване середовище розробки, яке дозволяє проводити розробку та налагодження програм для ПЛК Delta. ISPSoft підтримує стандартизовані мови програмування відповідно до європейських та міжнародних стандартів: **Ladder Diagram (LD)**, **Sequential Function Chart (SFC)** та **Structured Text (ST)**.

Для створення нового проєкту у ISPSoft натискаємо «**New**» або «**Ctrl+N**». У новому вікні (рис. 1.2) встановлюємо наступні параметри:

- назва проєкту. Бажано обирати латинськими літерами за схемою: прізвище, група, номер лабораторної роботи, номер варіанту;
- серія контролера – AS;
- модель - AS228P;
- шлях до папки проєкту.

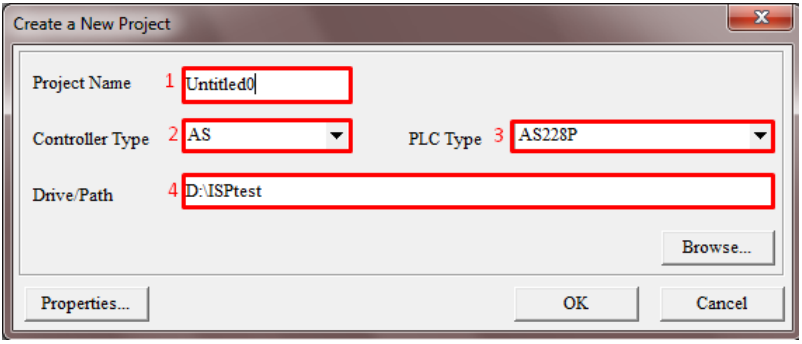

Рисунок 1.2 – Вікно створення проєкту у ISPSoft

В лівій частині робочої області розташовано вікно проєкту. В ньому є інформація щодо моделі ПЛК, додаткових модулів, глобальних змінних, також там розташовані файли програм та команди («**APIs**»). Команди або директиви згруповані по категоріям.

Для створення нової програми натискаємо правою кнопкою миші по вкладці «**Programs**» у вікні проєкту та обираємо «**New**». У новому вікні вводимо назву файлу та обираємо мову програмування Ladder Diagram (LD).

Створений файл має ранги («**Network**»), за допомогою яких створюється логіка виконання програми. У верхній частині вікна

(«**Local Symbols**») можна створювати або додавати локальні змінні для конкретної програми.

Ранги та логіка виконання програми мають бути побудовані наступним чином: «**умова**» - «**дія**». Найпростішими елементами системи виступають контакт та котушка.

Контакт («**Contact**») – бітовий елемент, який найчастіше виступає умовою для виконання рангу (рис. 4). Зазвичай це регістри М («**BOOL**») або Х (фізичний вхід ПЛК). Існує декілька видів контактів (рис. 1.3):

- нормально відкритий – за замовчуванням «0», при встановленні на «1» почне виконання рангу;
- нормально закритий - за замовчуванням «1», при встановленні на «0» припинить виконання рангу;
- тригер фронту імпульсу – дає один імпульс наступній операції у рангу при встановленні;
- тригер спаду імпульсу – дає один імпульс наступній операції у рангу при спаді імпульсу.

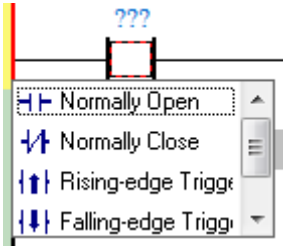

Рисунок 1.3 – Види контактів

Котушка («**Coil**») – бітовий елемент, який дозволяє змінювати стан бітів та продовжувати логіку програми. Існує декілька видів котушок (рис 1.4):

- вихід – встановлює «1», поки виконується умова;
- установка («**Set**») – встановлюється в «1» поки не буде скидання;
- скид («**Reset**»)– встановлює біт в стан «0».

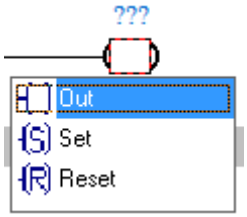

Рисунок 1.4 – Види котушок

Приклад програми з використанням контактів та котушок та написаної на мові Ladder Diagram (LD) приведений на рисунку 1.5.

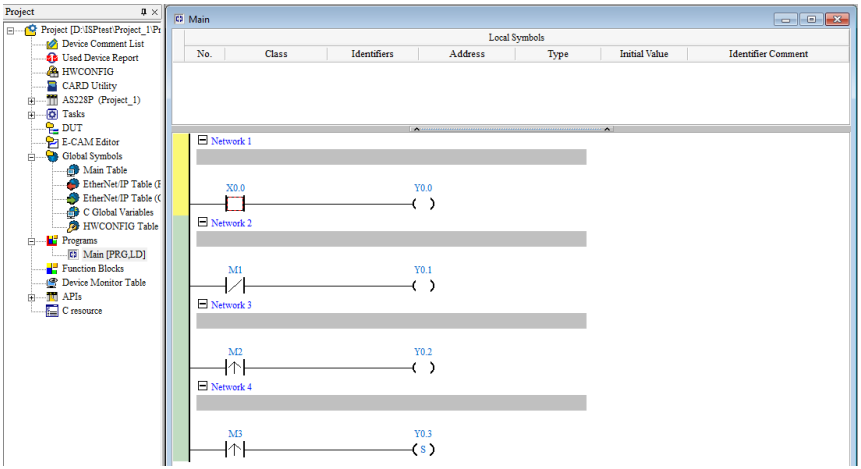

Рисунок 1.5 – Приклад програми

1.1.3 Створення проекту в DOPSoft

DOPSoft – це програмне забезпечення для створення людино-машинного інтерфейсу (**HMI – Human Machine Interface**), який дозволяє відслідковувати та керувати технологічними процесами на виробництві.

Середовище розробки дозволяє створювати кнопки, різного роду індикатори, здійснювати ввід та вивід даних, малювати графіки та анімації для відслідковування роботи системи.

Для створення нового проєкту необхідно натиснути кнопку «**New project**», обрати панель «**107BV**» та ввести назва проєкту. Назву бажано обирати латинськими літерами за схемою: прізвище, група, номер лабораторної роботи, номер варіанту.

В налаштуваннях проекту (рис. 1.6) виставляємо необхідну серію контролера «**Delta AS series PLC**», обираємо тип інтерфейсу RS485, встановлюємо біти даних («**Data Bits**») на «8 біт», швидкість («**Baud Rate**») – 115200, вимикаємо біти парності та обираємо створений віртуальний драйвер.

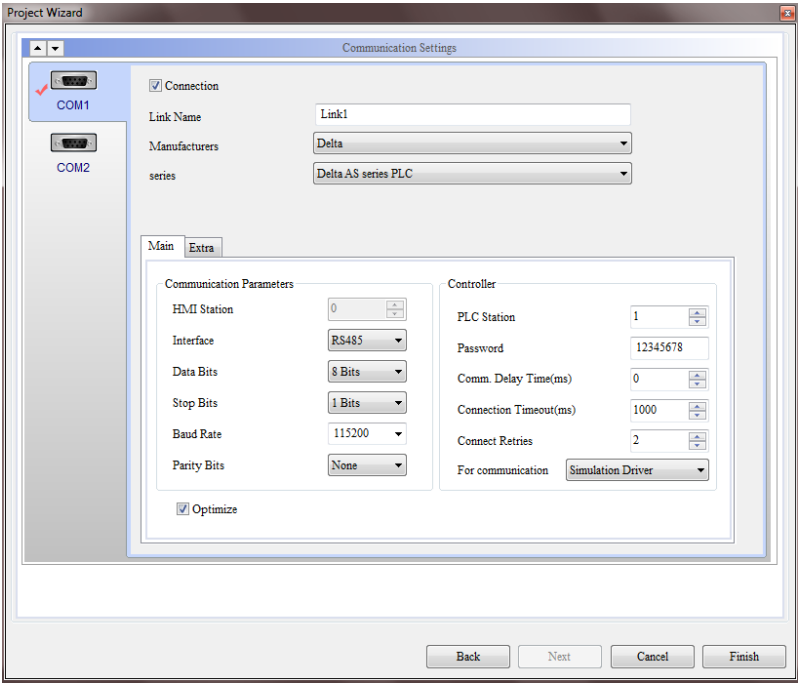

Рисунок 1.6 – Налаштування проекту в DOPSoft

В робочій області програми можна створювати різноманітні елементи інтерфейсу. Найпростішими є кнопка та індикатор.

Для створення кнопки перемикання стану переходимо у вкладку «**Elements**», обираємо «**Button**» та «**Maintained**» (рис. 1.7).

Після цього натискаємо на робоче поле панелі та «тягнемо» для зміни розміру кнопки.

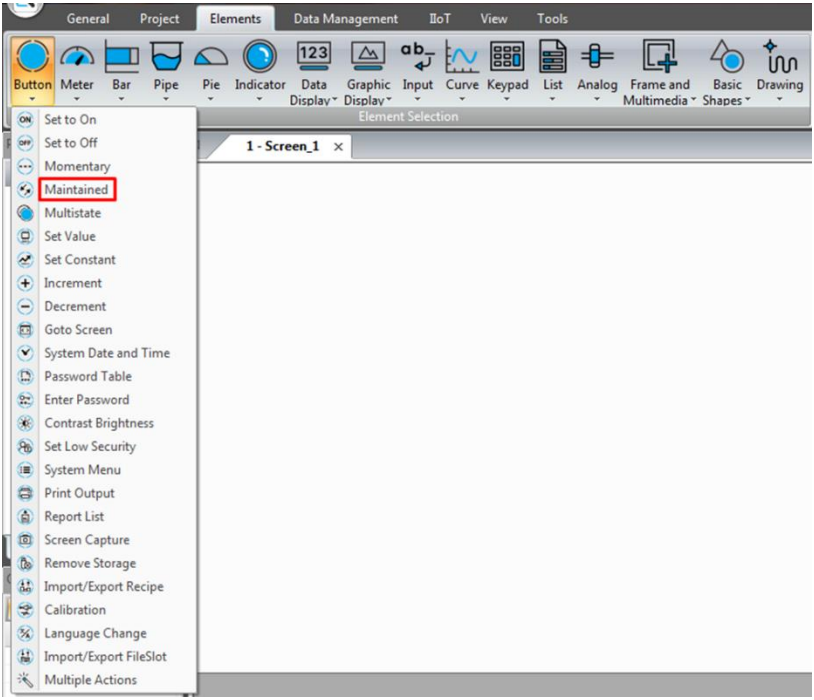

Рисунок 1.7 – Створення кнопки перемикання

При подвійному натисканні на створений елемент відкривається вікно налаштування. У полі кнопки «**Write address**» обираємо необхідну комірку пам'яті. Тепер при натисканні кнопки стан біту буде змінюватись на протилежний. У вкладках «**Style**» та «**Text**» можна змінювати параметри відображення (колір, форму, прозорість або додати текст для різних станів кнопки). Такі налаштування відображення є типовими майже для всі елементів.

Для створення лампи індикації обираємо елемент «**Multistate Indicator**» (рис. 1.8). У полі «**Read address**» виставляємо необхідну адресу для її зчитування. Для більшої зручності відображення можна налаштувати колір та напис. Приклад з двома кнопками та двома лампами індикації приведений на рисунку 1.9.

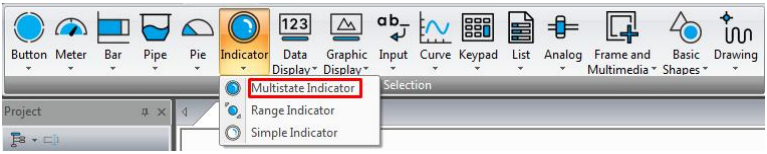

Рисунок 1.8 – Створення лампи індикації

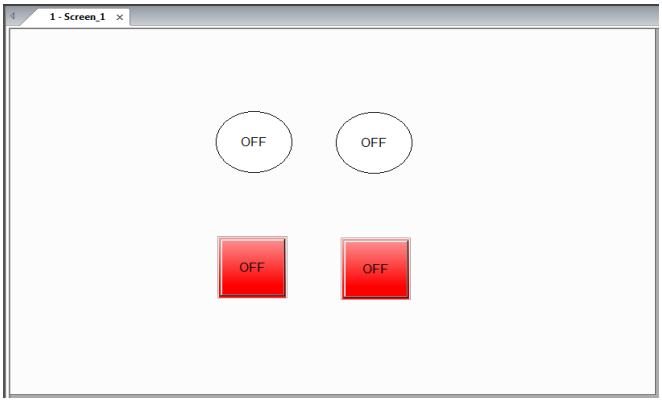

Рисунок 1.9 – Приклад панелі оператора

1.1.4 Симуляція роботи проєкту

Створену програму у ISPSoft необхідно скомпілювати натиснувши у верхній панелі програми «**Compile the Project**» (Ctrl+F7), завантажити у віртуальний ПЛК «**Download to PLC**» (Ctrl+F8), зайти в он-лайн режим «**Online Mode**» (Ctrl+F4) та запустити систему «**RUN**» (Ctrl+F11). зупинити симуляцію можна натиснувши кнопку зупинки «**STOP**» (Ctrl+F12). Порядок запуску симуляції приведений на рисунку 1.10.

Рисунок 1.10 – Порядок запуску симуляції

Для запуску симуляції панелі у DOPSoft переходимо у вкладку «Project» та обираємо «**Off-line Simulation**» (рис. 1.11).

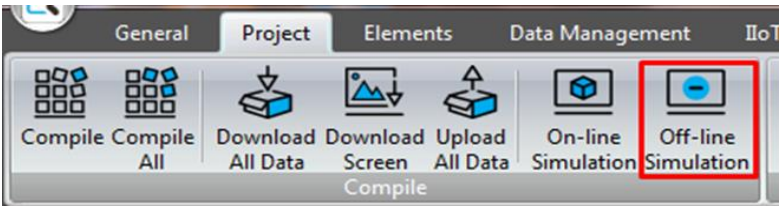

Рисунок 1.11 – Запуск симуляції панелі оператора

Приклад симуляції програмного коду у DOPSoft наведений на рисунку 1.12.

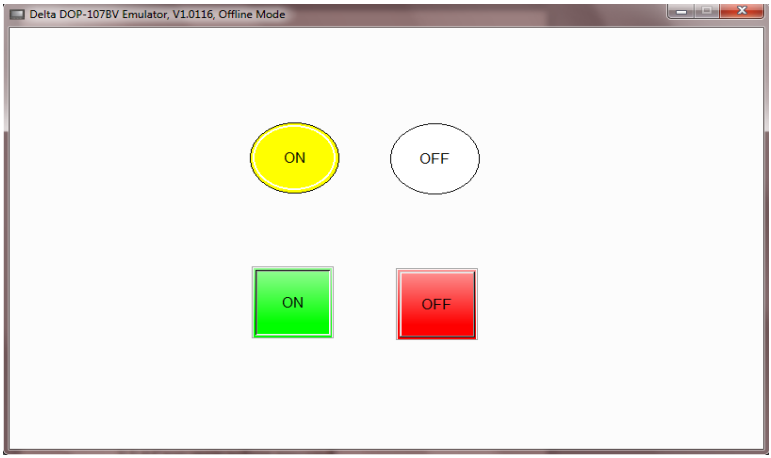

Рисунок 1.12 – Приклад симуляції у DOPSoft

1.1.5 Завдання на лабораторну роботу

Необхідно розробити схему підключення, що складається з контролера Delta AS228P, кнопок вмикання та 10-ти ламп.

Після розробки схеми необхідно написати програму керування контролером в ISPSoft та створити панель оператора згідно свого варіанта (табл. 1.1).

Таблиця 1.1 – Варіанти завдань

Варіант	Завдання
1	При натисканні кнопки 1 ввімкнути лампи 1, 4, 5. При натисканні кнопки 2 ввімкнути лампи 3, 7. При натисканні кнопки 3 ввімкнути лампи 6, 8, 10.
2	При натисканні кнопки 1 ввімкнути лампи 1, 2. При натисканні кнопки 2 ввімкнути лампи 4, 6. При натисканні кнопки 3 ввімкнути лампу 3, 10. При натисканні кнопки 4 ввімкнути лампи 5, 8, 9.
3	При натисканні кнопки 1 ввімкнути лампи 4, 7. При натисканні кнопки 2 ввімкнути лампи 1, 3, 9. При натисканні кнопки 3 ввімкнути лампи 2, 5, 6, 8.
4	При натисканні кнопки 1 ввімкнути лампи 2, 5, 7. При натисканні кнопки 2 ввімкнути лампи 4, 6. При натисканні кнопки 3 ввімкнути лампи 1, 9, 10.
5	При натисканні кнопки 1 ввімкнути лампи 4, 8. При натисканні кнопки 2 ввімкнути лампи 1, 5, 10. При натисканні кнопки 3 ввімкнути лампу 9. При натисканні кнопки 4 ввімкнути лампи 2, 6.
6	При натисканні кнопки 1 ввімкнути лампи 4, 5, 6. При натисканні кнопки 2 ввімкнути лампи 7, 8, 9. При натисканні кнопки 3 ввімкнути лампи 1, 2, 3.
7	При натисканні кнопки 1 ввімкнути лампи 7, 8, 10. При натисканні кнопки 2 ввімкнути лампи 4, 5. При натисканні кнопки 3 ввімкнути лампи 1, 3. При натисканні кнопки 4 ввімкнути лампу 6.
8	При натисканні кнопки 1 ввімкнути лампу 5. При натисканні кнопки 2 ввімкнути лампи 1, 7. При натисканні кнопки 3 ввімкнути лампи 2, 8. При натисканні кнопки 4 ввімкнути лампи 3, 4, 10. При натисканні кнопки 5 ввімкнути лампу 9.
9	При натисканні кнопки 1 ввімкнути лампи 5, 6, 7. При натисканні кнопки 2 ввімкнути лампи 2, 4, 8. При натисканні кнопки 3 ввімкнути лампи 1, 9.
10	При натисканні кнопки 1 ввімкнути лампи 1, 3, 6. При натисканні кнопки 2 ввімкнути лампи 2, 4, 7. При натисканні кнопки 3 ввімкнути лампи 5, 8, 9.

1.2 Зміст звіту

1.2.1 Тема та мета роботи.

1.2.2 Вхідні дані відповідно до варіанту.

1.2.3 Комутаційна схема системи.

1.2.4 Розроблений код у середовищі ISPSoft.

1.2.5 Розроблений людино-машинний інтерфейс у середовищі DOPSoft.

1.2.6 Висновки.

1.3 Контрольні питання

1.3.1 Надайте визначення програмованого логічного контролера (ПЛК).

1.3.2 Опишіть структуру мікропроцесорної системи управління.

1.3.3 Опишіть узагальнену структуру ПЛК.

1.3.4 Назвіть основні етапи робочого циклу ПЛК.

1.3.5 Які типи модулів ПЛК ви знаєте? Яке їх призначення?

1.3.6 Де може використовуватись ПЛК? Які задачі він виконує?

1.3.7 Які бувають типи контактів? Наведіть приклади їх використання.

1.3.8 Які бувають типи котушок? Наведіть приклади їх використання.

1.3.9 Які фізичні порти є у ПЛК Delta AS228P?

1.3.10 Які мови програмування ПЛК ви знаєте?

2 ЛАБОРАТОРНА РОБОТА 2. РОЗРОБКА ПРОСТОЇ СИСТЕМИ ПОЖЕЖОГАСІННЯ

Мета роботи: розробити просту систему пожежогасіння використовуючи ПЛК, отримати навички роботи з таймерами, лічильниками та математичними операціями.

2.1 Теоретична частина

2.1.1 Види змінних

Промисловий логічний контролер DELTA підтримує декілька типів змінних або об'єктів як загального, так і спеціального призначення.

У таблиці 2.1 приведені типи значень, які можуть використовуватись у ПЛК.

Таблиця 2.1 – Таблиця значень

Значення	Опис
Біт	Базовий блок у двійковій системі обчислення. Може приймати значення 0 або 1
Полубайт	Містить чотири послідовні біти
Байт	Містить 8 бітів. Може приймати значення від 0 до 255
Слово	Два послідовних байти або 16 біт
Подвійне слово	Два послідовних слова або 32 біти

При створенні змінної у середовищі ISPSoft програма пропонує обрати її розмірність (рис. 2.1) або автоматично визначає її при вибори типу змінної.

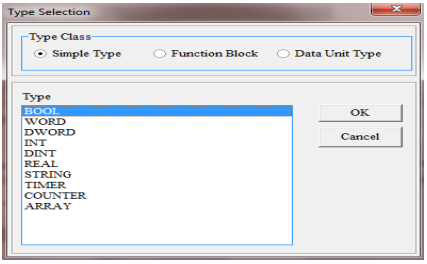

Рисунок 2.1 – Вибір типу змінної

В таблиці 2.2 приведені об’єкти змінних, які використовуються у ПЛК Delta AS228P. Типи змінних та їх кількість може відрізнятись в залежності від марки, серії або моделі ПЛК.

Таблиця 2.2 – Таблиця об’єктів

Тип	Найменування об’єкту		Кількість	Діапазон
Бітові об’єкти	Вхідне реле	X	1024	X0.0-X63.15
	Вихідне реле	Y	1024	Y0.0-Y63.15
	Регістр даних	D	48,000	D0.0-D29999.15
	Допоміжне реле	M	8192	M0.0-M8191
	Спеціальне допоміжне реле	SM	2048	SM0.0-SM4095
	Шагове реле	S	2048	S0.0-S2047
	Таймер	T	512	C0-C511
	Лічильник	C	512	C0-C511
	32-бітний лічильник	HC	256	HC0-HC256
Об’єкти слова	Вхідне реле	X	64	X0-X63
	Вихідне реле	Y	64	Y0-Y63
	Регістр даних	D	30000	D0-D29999
	Спеціальне допоміжне реле	SR	2048	SR0-SR2047
	Файловий регістр	FR	65536	FR0-FR65536
	Таймер	T	512	T0-T511
	Лічильник	C	512	C0-C511
	32-бітний лічильник	HC	256	HC0-HC256
	Індексний регістр	E	10	E0-E9
Константи	Десятковий формат	K	16 біт: -32768-32767 32 біт: -2147483648-2147483647	
Константи	Шістнадцятковий формат	16#	16 біт: 16#0-16#FFFF 32 біт: 16#0-16#FFFFFFFF	
	З плаваючою комою	F	32 біт: ±1.17549435-38-±3.40282347+38	
Символьна змінна	Символьна змінна	“$”	1-31 символів	

2.1.2 Таймери

Таймер – інструкція для підрахунку часу. Задається значення, до якого буде рахувати таймер. При виконанні умови починається відлік. Після отримання заданого значення таймер вмикає реле контакту. Кінцевим значенням може бути константа або комірка пам'яті. Після скиду реле контакту починається новий відлік.

Існує декілька видів таймерів:

- TMR – одиниця синхронізації 100 мс;
- TMRM -– одиниця синхронізації 10 мс;
- TMRH – одиниця синхронізації 1 мс.

Таймери мають особливий тип даних. Вони містять: значення до якого треба рахувати, поточне значення та біт виконання (закінчення рахування).

2.1.3 Лічильники

Лічильник – інструкція для підрахунку кількості спрацювань певної умови. Задається значення, до якого буде рахувати лічильник. При виконанні умови відбувається інкрементація поточного значення. Після отримання заданого значення лічильник вмикає реле контакту закінчення роботи.

Лічильник рахує вгору. Кінцевим значенням може бути константа або комірка пам'яті. Після скиду реле контакту починається новий відлік.

2.1.4 Математичні операції

Для виконання математичних операцій використовуються інструкції: додавання «+», віднімання «-», множення «*», ділення «/».

Існують модифікації даних інструкцій, які використовуються з певним розміром або типом змінної. Ці операції включають в себе:

- **D** – операції з 32 бітними операндами;
- **P** – імпульсна версія команди (розрахунок буде проводитись однократно та буде перераховуватись після нового спрацювання умови);
- **F** – операції з числами з плаваючою комою.

Наприклад, операція D+ - це складання двох 32 розрядних комірок (подвійне слово), а операція F* - це множення значень з плаваючою комою.

При виконанні операцій потрібно коректно визначати тип даних. Наприклад, при множенні слова (16 біт) на слово (16 біт) вихідний результат буде складати подвійне слово (32 біти).

Для використання та зберігання значень з плаваючою комою потрібно використовувати тип даних **REAL** (32 біти).

Базовий розмір комірки пам'яті ПЛК 16 біт. Якщо використовується змінна типа **DWORD** або **REAL**, вона буде займати 2 комірки пам'яті – вказану та наступну. Наприклад, змінна DWORD, яка зберігається в комірці D10, буде займати її та D11. Треба враховувати це при проведенні математичних операцій.

2.1.5 Відображення даних у DOPSoft

Для відображення значень на панелі оператора потрібно обрати «**Numeric Display**» (рис. 2.2).

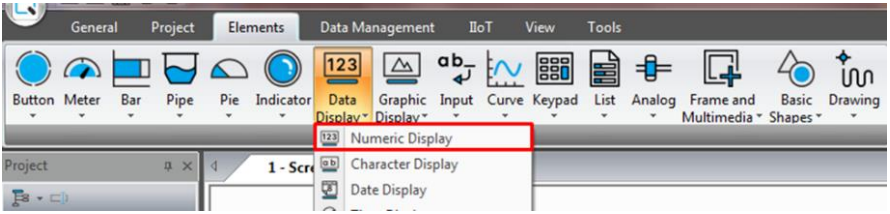

Рисунок 2.2 - Інструкція для виводу значень на панель

Для налаштування відображення (рис 2.3) потрібно обрати відповідну комірку пам'яті, розмір змінної «**Data Type**», формат «**Data Format**», кількість цілих знаків «**Integer Digits**», кількість знаків після коми «**Fractional**».

Сума «**Integer Digits**» та «**Fractional**» не може перевищувати 7 значущих цифр.

Вкладка «**Unit conversion**» дозволяє роботи перетворення таких величин без їх зміни в комірці пам'яті: швидкість, тиск, температура, відстань, вага та об'єм. Такою програма дозволяє створити власну формулу для розрахунку величин.

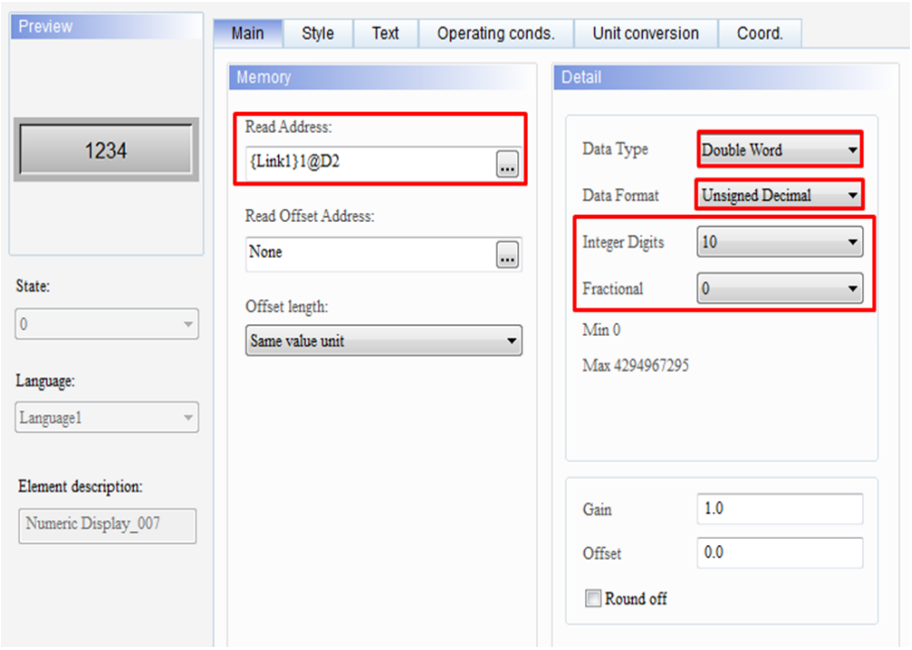

Рисунок 2.3 – Панель налаштування виводу

2.1.6 Завдання на лабораторну роботу

Розробити систему пожежогасіння. Реалізувати програму для моніторингу пожежі, вмикання електроклапанів води, створити панель оператора та графічне відображення стану приміщення.

Апаратне забезпечення: дискретні датчики вогню, електроклапани для подачі води (дискретні), лампа для оповіщення про пожежу.

На панелі розташувати рівномірно датчики вогню та вентилі подачі води. Для можливості симуляції спрацювання датчиків на панелі їх реалізувати як кнопки. При спрацюванні датчиків розрахувати та вивести відсоток ураження приміщення вогнем, ввімкнути вентилі подачі води, що перекривають площу враження вогнем. Забезпечити лічильник кількості спрацювань датчиків вогню. Забезпечити блимання лампи аварійного сповіщення та індикатору на панелі.

У таблиці 2.3 приведені варіанти завдань.

Таблиця 2.3 – Варіанти завдань

Варіант	Форма приміщення	Кількість датчиків вогню	Кількість електроклапанів	Частота мерехтіння лампи
1	Прямокутна	8	6	1 секунда
2	Г-образна	12	8	0,5 секунди
3	П-образна	11	9	1,5 секунди
4	Квадратна	12	9	1 секунда
5	Прямокутна	13	12	2 секунди
6	Г-образна	14	11	1,5 секунди
7	П-образна	14	10	1 секунда
8	Квадратна	9	12	2 секунди
9	П-образна	12	10	0,5 секунди
10	Прямокутна	10	14	2 секунди

2.2 Зміст звіту

2.2.1 Тема та мета роботи.

2.2.2 Вхідні дані відповідно до варіанту.

2.2.3 Комутаційна схема системи.

2.2.4 Розроблений код у середовищі ISPSoft, розроблена панель керування та моніторингу стану пожежі DOPSoft.

2.2.6 Результати симуляції.

2.2.7 Висновки.

2.3 Контрольні питання

2.3.1 Які типи значень використовуються у ПЛК Delta AS228P?

2.3.2 Які типи змінних ПЛК Delta AS228P ви знаєте?

2.3.3 Назвіть особливості роботи зі змінними типу REAL.

2.3.4 Яким чином зберігаються змінні з плаваючою комою?

2.3.5 У чому різниця між змінними DWORD та INT?

2.3.6 Які спеціальні регістри даних ви знаєте?

2.3.7 Приведіть приклад використання спеціальних регістрів.

2.3.8 Що таке таймер? Наведіть приклад його використання.

2.3.9 Що таке лічильник? Наведіть приклад його використання.

2.3.10 Які існують модифікації інструкцій. Наведіть приклад.

3 ЛАБОРАТОРНА РОБОТА 3. РОЗРОБКА СИСТЕМИ КОНТРОЛЮ ТЕМПЕРАТУРИ В ПЕЧІ

Мета роботи: отримати навички роботи з операціями порівняння, нецілими числами, вводом та виводом інформації на панель оператора.

3.1 Теоретична частина

3.1.1 Операції порівняння

Операції порівняння використовуються для порівняння двох або більше величин, перевірки входження в інтервал та виступають як умова для виконання дії.

Для операцій порівняння використовуються інструкції типу LD (табл. 3.1).

Таблиця 3.1 – Операції порівняння

Номер API	16-бітна інструкція	32- бітна інструкція	Умова виконання	Умова розриву
0000	LD =	**D**LD =	$S_1 = S_2$	$S_1 \neq S_2$
0001	LD < >	**D**LD < >	$S_1 \neq S_2$	$S_1 = S_2$
0002	LD >	**D**LD >	$S_1 > S_2$	$S_1 \leq S_2$
0003	LD > =	**D**LD > =	$S_1 \geq S_2$	$S_1 < S_2$
0004	LD <	**D**LD <	$S_1 < S_2$	$S_1 \geq S_2$
0005	LD < =	**D**LD < =	$S_1 \leq S_2$	$S_1 > S_2$

Існують модифікації даних інструкцій вони включають в себе:
- D – операції з 32 бітними операндами;
- $ – порівняння строк;
- Z – порівняння абсолютних значень.

3.1.2 Ввід даних у DOPSoft

Для вводу значень на панелі оператора потрібно обрати «**Numeric Entry**» (рис. 3.1).

Для налаштування (рис. 3.2) елементу треба обрати адресу пам’яті, розмір та тип даних, максимальне та мінімальне значення.

Після виклику вікна вводу відкриється клавіатура для введення числового значення.

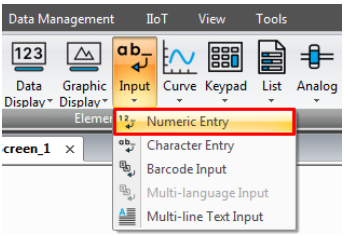

Рисунок 3.1 – Інструкція для вводу значень з панелі

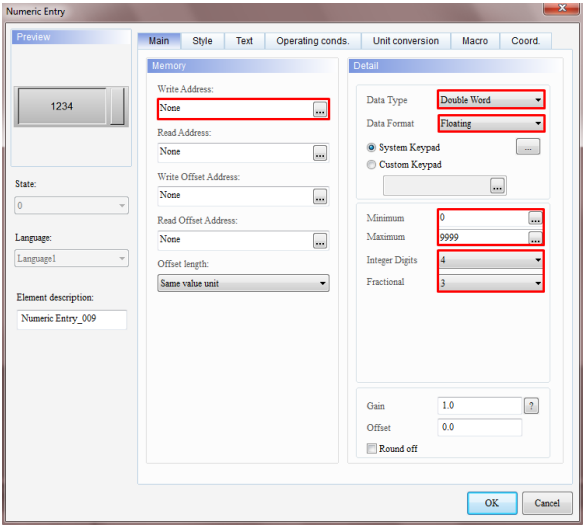

Рисунок 3.2 – Вікно налаштування панелі вводу

Для роботи з нецілими числами потрібно створити зміну типу REAL у ISPSoft. У вікні налаштування елементу вводу або виводу вказати відповідну комірку, обрати розмір «**Data Type**» - «**Double Word**», формат змінної «**Data Format**» - «**Float**», та обрати кількість символів до та після коми.

3.1.3 Завдання на лабораторну роботу

Використовуючи характеристику термопари в полі вводу вводити значення напруги, яку видає термопара на виході. Програма на ПЛК має розрахувати значення температури у ºC та ºF, які потрібно

вивести на панель оператора. Забезпечити виконання додаткових умов згідно варіанту. На рисунку 3.4 наведено графік залежності напруги від температури. Для спрощення розрахунків будемо вважати що всі залежності лінійні.

Апаратне забезпечення:

- термопара – чутливий елемент термоелектричного перетворювача у вигляді двох ізольованих провідників із різнорідних матеріалів, з'єднаних на одному кінці, принцип дії якого ґрунтується на використанні термоелектричного ефекту для вимірювання температури, графік залежності напруги від температури приведений на рисунку 3.3;

- вентилятор – пристрій для охолодження та/або зменшення температури;

- сервопривід – пристрій дистанційного керування для здійснення механічного переміщення регулюючого органу (дверці для охолодження);

- тен – електричний нагрівник опору, що складається з нагрівального елементу з контактними стрижнями на кінцях, використовується для нагріву та/або збільшення температури.

У таблиці 3.2 приведені варіанти завдань для лабораторної роботи.

Таблиця 3.2 - Варіанти завдань

Варіант	Апаратне забезпечення	Додаткові умови
1	2	3
1	Термопара (тип Pt-Pd 0°C…1500°C), вентилятор з трьома режимами роботи (відк., 2 швидкості)	Якщо температура менша за 100 °C вентилятор вимкнено, від 100 °C до 500 °C вмикати вентилятор на 1-й швидкості, більше 500 °C вмикати вентилятор на 2-й швидкості. Режими роботи вентилятора додатково відображати у вигляді апаратних ламп та індикаторів на панелі. Забезпечити лічильник кількості перемикання режимів вентилятора.

Продовження таблиці 3.2 - Варіанти завдань

1	2	3
2	Термопара (тип E 0°C…800°C), сервопривід лінійного руху з інвертором зворотного ходу (2 контакти керування – прямого та зворотного ходу)	Сервопривід лінійного ходу керує рівнем відкривання дверці для охолодження (зауважте, що рівень відкривання / закривання регулюється часом подачі відповідного сигналу). Якщо температура менша за 100 °C дверці закриті, при температурі від 100 °C до 800 °C відсоток відкривання дверці пропорційний температури. Забезпечити текстове відображення проценту відкриття дверці (при бажанні графічне відображення на панелі).
3	Термопара (тип N 0°C…1100°C), вентилятор з трьома режимами роботи (відк., 3 швидкості)	Якщо температура менша за 100 °C вентилятор вимкнено, від 100 °C до 400 °C вмикати вентилятор на 1-й швидкості, від 400 °C до 800 °C вмикати вентилятор на 2-й швидкості, більше 800 °C вмикати вентилятор на 3-й швидкості. Режими роботи вентилятора додатково відображати у вигляді апаратних ламп та індикаторів на панелі. Забезпечити лічильник кількості перемикання режимів вентилятора.
4	Термопара (тип T - 185°C…300°C), сервопривід лінійного руху з інвертором зворотного ходу (2 контакти	Сервопривід лінійного ходу керує рівнем відкривання дверці для охолодження (зауважте, що рівень відкривання / закривання регулюється часом подачі відповідного сигналу). Якщо температура менша за 30 °C включити тен для додаткового нагрівання, якщо температура менша за 80 °C дверці закриті, при температурі від 80 °C до 300 °C відсоток відкривання дверці

Продовження таблиці 3.2 - Варіанти завдань

1	2	3
	керування – прямого та зворотного ходу) , тен	пропорційний температури. Забезпечити текстове відображення проценту відкриття дверці (при бажанні графічне відображення на панелі).
5	Термопара (тип J 0ºС…700ºС), вентилятор з трьома режимами роботи (відк, 3 швидкості), тен	Якщо температура менша за 100 ºС вмикаємо тен, якщо температура менша за 200 ºС вентилятор вимкнено, від 200 ºС до 400 ºС вмикати вентилятор на 1й швидкості, від 400 ºС до 600 ºС вмикати вентилятор на 2й швидкості, більше 600 ºС вмикати вентилятор на 3й швидкості. Режими роботи вентилятора додатково відображати у вигляді апаратних ламп та індикаторів на панелі. Забезпечити лічильник кількості перемикання режимів вентилятора. Для тренування роботи з таймерами забезпечити блимання додаткової лампи та індикатору на панелі раз на 3 секунди.
6	Термопара (тип K 0ºС…1100ºС), сервопривід лінійного руху з інвертором зворотного ходу (2 контакти керування – прямого та зворотного ходу), 2 тени	Сервопривід лінійного ходу керує рівнем відкривання дверці для охолодження (зауважте, що рівень відкривання / закривання регулюється часом подачі відповідного сигналу). Якщо температура менша за 30 ºС включити обидва тени для додаткового нагрівання. Якщо температура менша за 80 ºС працює лише 1 тен, якщо температура менша за 200 ºС дверці закриті, при температурі від 200 ºС до 1100 ºС відсоток відкривання дверці пропорційний температури. Забезпечити текстове відображення проценту відкриття дверці (при бажанні графічне відображення на панелі).

Продовження таблиці 3.2 - Варіанти завдань

1	2	3
7	Термопара (тип P 0ºC…1100ºC), сервопривід лінійного руху з інвертором зворотного ходу (2 контакти керування – прямого та зворотного ходу), 2 тени	Сервопривід лінійного ходу керує рівнем відкривання дверці для охолодження (зауважте, що рівень відкривання / закривання регулюється часом подачі відповідного сигналу). Якщо температура менша за 50 ºC включити обидва тени для додаткового нагрівання. Якщо температура менша за 150 ºC працює лише 1 тен, якщо температура менша за 250 ºC дверці закриті, при температурі від 350 ºC до 1100 ºC відсоток відкривання дверці пропорційний температури. Забезпечити текстове відображення проценту відкриття дверці (при бажанні графічне відображення на панелі).
8	Термопара (тип K 0ºC…1100ºC), вентилятор з трьома режимами роботи (відк, 3 швидкості), тен	Якщо температура менша за 100 ºC вмикаємо тен, якщо температура менша за 200 ºC вентилятор вимкнено, від 200 ºC до 400 ºC вмикати вентилятор на 1й швидкості, від 400 ºC до 600 ºC вмикати вентилятор на 2й швидкості, більше 600 ºC вмикати вентилятор на 3й швидкості. Режими роботи вентилятора додатково відображати у вигляді апаратних ламп та індикаторів на панелі. Забезпечити лічильник кількості перемикання режимів вентилятора. Для тренування роботи з таймерами забезпечити блимання додаткової лампи та індикатору на панелі раз на 3 секунди.
9	Термопара (тип E 0ºC…800ºC), вентилятор з трьома	Якщо температура менша за 100 ºC вмикаємо 2 тени, якщо температура менша 200 ºC, працює один тен, якщо температура менша за 350 ºC вентилятор вимкнено, від 500 ºC до 600 ºC вмикати

Продовження таблиці 3.2 - Варіанти завдань

1	2	3
	режимами роботи (відк, 3 швидкості), 2 тени	вентилятор на 1й швидкості, від 600 ºС до 700 ºС вмикати вентилятор на 2й швидкості, більше 700 ºС вмикати вентилятор на 3й швидкості. Режими роботи вентилятора додатково відображати у вигляді апаратних ламп та індикаторів на панелі. Забезпечити лічильник кількості перемикання режимів вентилятора. Для тренування роботи з таймерами забезпечити блимання додаткової лампи та індикатору на панелі раз на 1,5 секунди.
10	Термопара (тип N 0ºС…1100ºС), сервопривід лінійного руху з інвертором зворотного ходу (2 контакти керування – прямого та зворотного ходу), 1 тен	Сервопривід лінійного ходу керує рівнем відкривання дверці для охолодження (зауважте, що рівень відкривання / закривання регулюється часом подачі відповідного сигналу). Якщо температура менша за 500 ºС включити тен для додаткового нагрівання. Якщо температура менша за 650 ºС дверці закриті, при температурі від 650 ºС до 1100 ºС відсоток відкривання дверці пропорційний температури. Забезпечити текстове відображення проценту відкриття дверці (при бажанні графічне відображення на панелі).

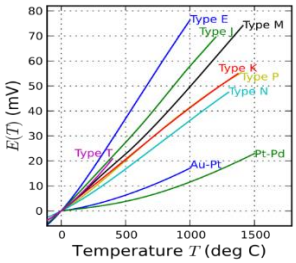

Рисунок 3.3 - Залежність напруги від температури

3.2 Зміст звіту

3.2.1 Тема та мета роботи.

3.2.2 Вхідні дані відповідно до варіанту.

3.2.3 Комутаційна схема системи.

3.2.4 Розроблений код у середовищі ISPSoft, розроблена панель керування у DOPSoft.

3.2.6 Результати симуляції.

3.2.7 Висновки.

3.3 Контрольні питання

3.3.1 Що таке операції порівняння? З якою метою вони використовуються?

3.3.2 Яким чином відбувається порівняння строк?

3.3.3 Яким чином вводяться значення з панелі оператора?

3.3.4 Які особливості мають числа з плаваючою комою при роботі з ними в середовищі DOPSoft?

3.3.5 Що таке термопара? Які ви знаєте типи термопар?

3.3.6 Назвіть основні характеристики та критерії вибору термопари.

3.3.7 Чи можна підключити термопару до базової комплектації ПЛК Delta AS228P? Обґрунтуйте свою відповідь.

3.3.8 Як виводяться значення з плаваючою комою на панель оператора? Які є особливості?

3.3.9 Які додаткові модулі існують для ПЛК Delta AS228P? Назвіть декілька типів та їх призначення.

3.3.10 Назвіть особливості роботи з аналоговими модулями.

4 ЛАБОРАТОРНА РОБОТА 4. СИСТЕМА ЗМІШУВАННЯ ФАРБИ З ВИКОРИСТАННЯМ ФУНКЦІОНАЛЬНИХ БЛОКІВ

Мета роботи: отримати практичні навички застосування Function block (FB) для автоматизації на виробництві, ознайомитись з особливостями їх використання.

4.1 Завдання до роботи

4.1.1 Створення змінних

Змінна – це математична величина, яка може змінюватись у ході виконання програми. Вона має значення, тип, адресу та ім’я (ідентифікатор). Ім’я змінної має відображати її призначення та не містити зарезервованих слів або операцій.

Для створення їх у головній програмі натискаємо правою кнопкою миші під заголовком «Локальні змінні» («**Local Symbols**») та обираємо «Додати символ» («**Add a Symbol**») (рис. 4.1).

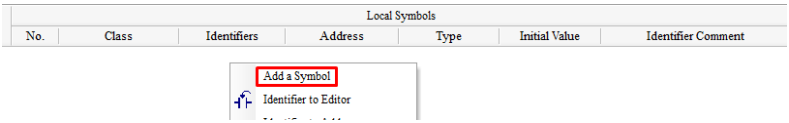

Рисунок 4.1 – Додавання змінних

У вікні, що відкрилось (рис 4.2), необхідно встановити ім’я змінної («**Identifier**»), визначити адресу («**Address**») та тип змінної («**Type**»). Якщо тип не обраний, він встановлюється автоматично відповідно до адреси змінної.

Якщо необхідно створити глобальну змінну, ставимо галку напроти («**Define global**»). Глобальні змінні будуть доступні не тільки в конкретному файлі, а у всьому проекті.

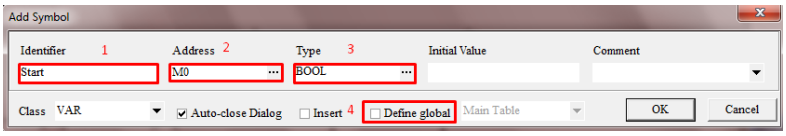

Рисунок 4.2 – Створення змінної

4.1.2 Функціональні блоки

Функціональний блок (ФБ) — це блок, який описує функцію між вхідними та вихідними змінними.

Для створення функціонального блоку натискаємо правою кнопкою миші по вкладці у вікно проєкту (рис. 4.3).

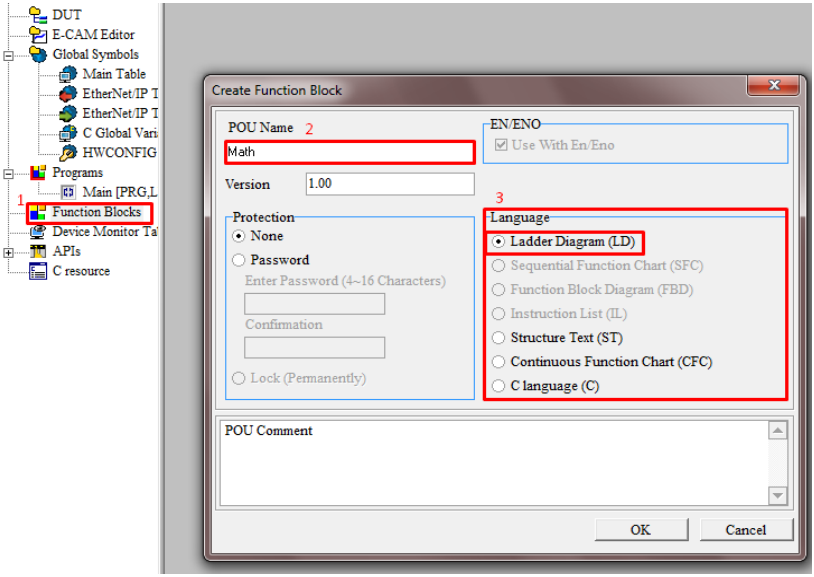

Рисунок 4.3 – Створення функціонального блоку

Задаємо назву та обираємо мову програмування. Функціональні блоки на мові **Ladder Diagram (LD)** підкорюються загальним правилам програмування .

Створення змінних у ФБ відбувається за тим самим принципом, як і у звичайному файлі програми (рис. 4.4).

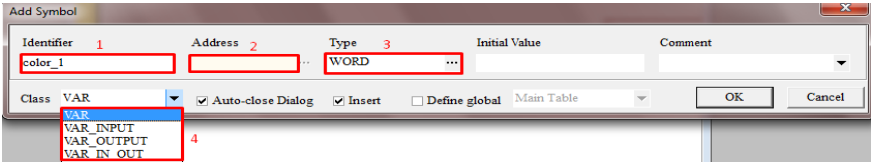

Рисунок 4.4 – Створення змінної у ФБ

Встановлюємо ім’я змінної («**Identifier**») та тип змінної («**Type**»). Створена змінна буде внутрішньою, тому адресу встановити неможливо. Також треба обрати клас («**Class**») змінної. Це може бути внутрішня змінна («**VAR**»), вхідна («**VAR_INPUT**») для отримання значення з програми та її обробки, вихідна («**VAR_OUTPUT**») для передачі результату у головну програму або вхідна-вихідна («**VAR_IN_OUT**»).

Функціональний блок може використовуватись декілька разів. Це зручно для керування процесами, які мають схоже управління та параметри. Для кожного нового функціонального блоку створюється окрема змінна в програмі, яка його викликає. Також кожен окремий ФБ має свої окремі змінні (рис. 4.5), які можна дивитись при роботі ПЛК або симуляції. Приклад функціонального блоку приведений на рисунку 4.6.

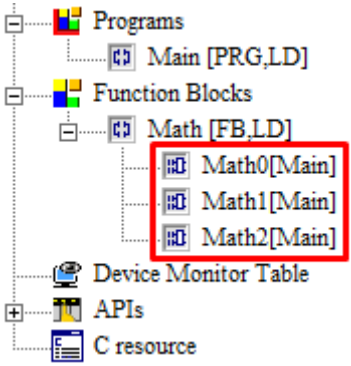

Рисунок 4.5 – Окремі ФБ при їх повторенні в програмі

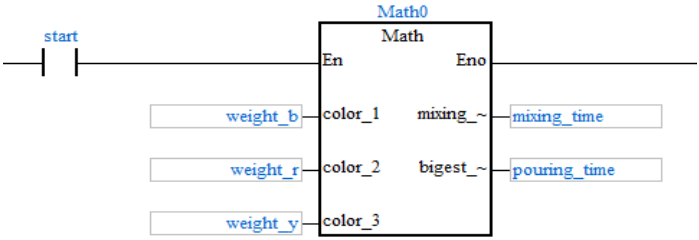

Рисунок 4.6 - Приклад функціонального блоку

4.1.3 Використання таймерів та лічильників у функціональному блоку

Використання інструкцій таймерів та лічильників у функціональному блоку мають деякі особливості. Для роботи потрібно створити глобальну змінну (наприклад, **FB_Timer1** та **FB_Counter1**) та передати її у функціональний блок (рис 4.7).

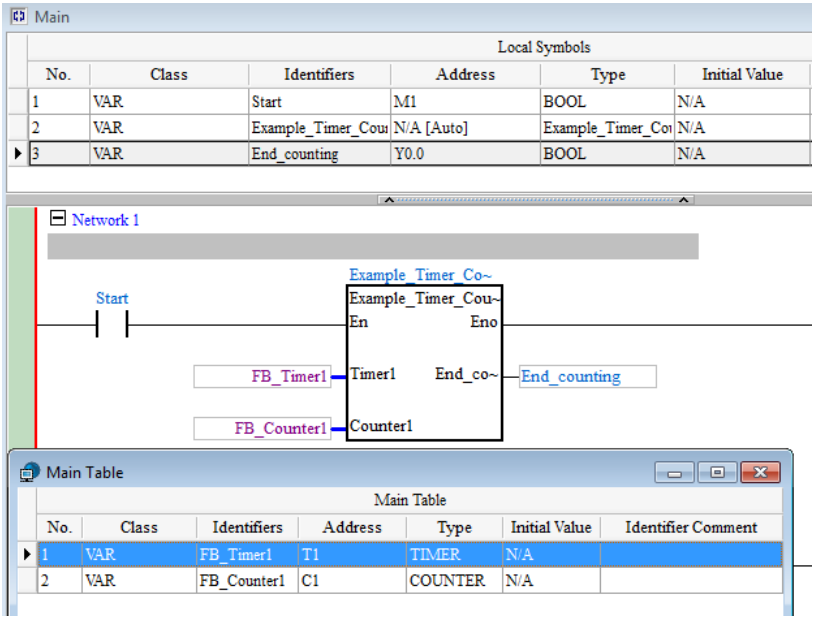

Рисунок 4.7 – Використання таймерів та лічильників у ФБ

У самому функціональному блоку необхідно обрати клас «**VAR_IN_OUT**» та тип змінної T_POINTER або C_POINTER (рис. 4.8).

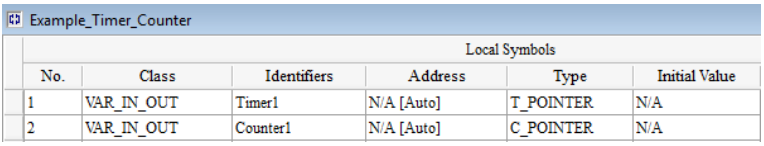

Рисунок 4.8 – Створення змінних у функціональному блоці

На рисунку 4.9 приведений приклад функціонального блоку з використанням таймеру та лічильника.

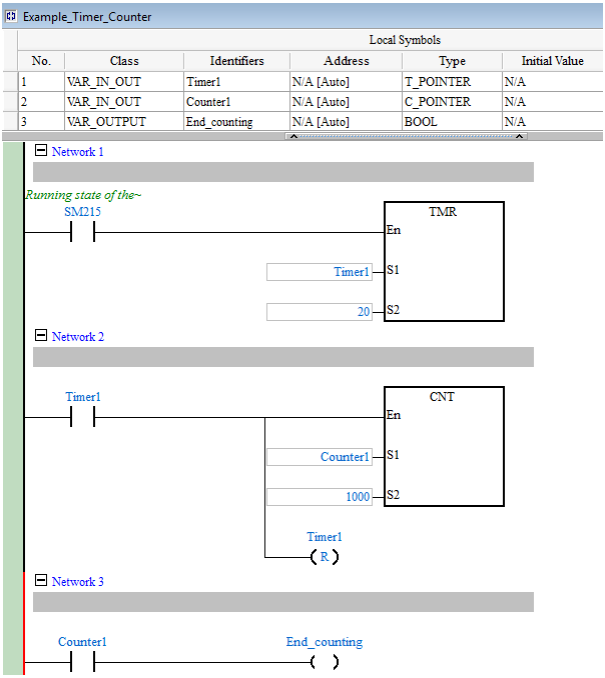

Рисунок 4.9 – Приклад програми

Додатково відслідковувати поточний стан змінних можна за допомогою вкладки «**Device Monitor Table**» у вікні проекту. Для цього потрібно натиснути правою кнопкою миші на вкладку та натиснути «**New**», обрати назву таблиці. Щоб додати змінні, необхідно натиснути правою кнопкою миші в полі вікна та обрати «**Select Symbols**», у новому вікні обрати програми, а в них змінні, стан яких ви бажаєте відслідковувати (рис. 4.10).

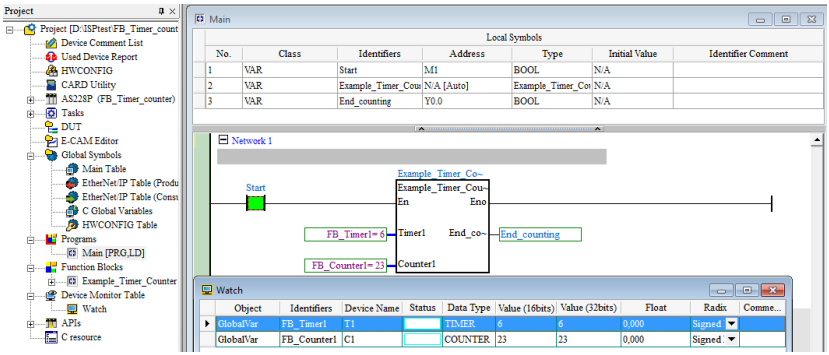

Рисунок 4.10 – Приклад виконання програми з використанням ФБ

4.1.4 Завдання на лабораторну роботу

В загальну чашу для змішування подається три фарби різного кольору. Об’єм кожної фарби визначається окремо та задається з панелі оператора. Будемо вважати, що 1 одиниця дорівнює одному літру, а швидкість виливання дорівнює 5 секунд/літр. В чаші є два гвинти, які змішують фарбу.

В ході роботи необхідно створити змінні для: старту роботи, фарби кожного кольору (будемо вважати, що в нас є синя, жовта та червона фарби), часу наливання фарби, часу змішування та для керування гвинтами для змішування.

Для функціонального блоку необхідно створити змінні для фарби кожного кольору, загальну вагу, загальний час змішування, найбільшу вагу та найбільший час виливання фарби.

Після цього створюємо алгоритм для розрахунку загальної ваги шляхом додавання. А також час необхідний для змішування фарби. Будемо вважати, що на кожний кілограм фарби необхідно витратити 5 секунд розмішування.

Знаходимо фарбу найбільшого об’єму та розраховуємо час виливання (швидкість виливання дорівнює 5 секунд/літр).

Необхідно створити програму, яка при запуску буде розраховувати необхідні параметри фарб. Реалізувати таймери для затримки наливання та змішування фарби.

На панелі оператора забезпечити можливість введення значень маси фарб, вивід загальної ваги та індикацію роботи гвинтів для змішування.

4.2 Зміст звіту

4.2.1 Тема та мета роботи.

4.2.2 Вхідні дані.

4.2.3 Комутаційна схема системи.

4.2.4 Створенні змінні у головній програмі та змінні функціонального блоку.

4.2.5 Розроблений код у середовищі ISPSoft, розроблена панель керування у DOPSoft.

4.2.6 Результати симуляції.

4.2.7 Висновки.

4.3 Контрольні питання

4.3.1 Які змінні ви використовували в лабораторній роботі? Дайте їм характеристику.

4.3.2 Що таке функціональний блок? Для чого він може використовуватись?

4.3.3 Які існують особливості або обмеження для використання ФБ?

4.3.4 Які особливості існують для роботи зі змінними у ФБ?

4.3.5 Яким чином використовується таймер у ФБ?

4.3.6 Яким чином використовується лічильник у ФБ?

4.3.7 В чому різниця між локальними та глобальними змінними?

4.3.8 В яких випадках доречно створювати локальні змінні, а в яких глобальні?

4.3.9 Що таке високошвидкісний лічильник? В яких випадках він використовується?

4.3.10 Яким чином можна дивитись поточні значення змінних при роботі програми?

ЛІТЕРАТУРА

1. AS Series Hardware and Operation Manual. URL : https://filecenter.deltaww.com/Products/download/06/060301/Manual/DELTA_IA-PLC_AS_HOM_EN_20220530.pdf (дата звернення : 01.09.2024).

2. AS Series Programming Manual. URL : https://filecenter.deltaww.com/Products/download/06/060301/Manual/DELTA_IA-PLC_AS_PM_EN_20210624.pdf (дата звернення : 01.09.2024).

3. ISPSoft User Manual. URL : https://filecenter.deltaww.com/Products/download/06/060301/Manual/DELTA_IA-PLC_ISPSoft_UM_EN_20210329.pdf (дата звернення : 01.09.2024).

4. DOPSoft User Manual. URL : https://filecenter.deltaww.com/Products/download/06/060302/Manual/DELTA_IA-HMI_DOPSoft_UM_EN_20211230.pdf (дата звернення : 01.09.2024).