

3. ПРОГРАМУВАННЯ У СИСТЕМІ MULTIPROG

3.1. Розробка проекту – загальні кроки

Щоб реалізувати проект, необхідно зробити декілька загальних кроків.

Крок 1. Визначення структури проекту у проектному дереві

Визначення структури проекту у проектному дереві означає вставлення у проект необхідних одиниць організації програми, так званих POUs, якими можуть бути програми, функціональні блоки і функції.

Крок 2. Редагування POUs

Редагування POUs означає декларування параметрів і змінних та редагування тіла коду за допомогою графічного та текстового редакторів і Edit Wizard (Майстр Редагування).

Крок 3. Зв'язок програм і задач

Для того, щоб генерувати код до контролера, кожен програму необхідно зв'язати з задачею, яка транслює програму до ПЛК. Для цього у проектному дереві треба позначити теку задачі і в панелі інструментів клацнути по іконі Add Object (Додати об'єкт), щоб викликати діалогове вікно Insert (Вставка). У діалоговому вікні необхідно активізувати перемикач Program (Програма), ввести ім'я зразка програми і вибрати бажану програму у вікні списку Program Type (Тип Програми).

Після підтвердження діалогу зразок програми автоматично вставиться у теку задачі.

Крок 4. Компіляція проекту

Коли редагування проекту закінчено, він компілюється, використовуючи в панелі інструментів ікону Compile Worksheet (Компілювати Робочий лист). При цьому тіло коду перевіряється на правильний синтаксис.

Крок 5. Завантаження

Компільований проект має бути завантаженим до ПЛК або симулятора. Для цього в панелі інструментів необхідно клацнути по іконі Project Control Dialog (Діалог Управління Проектом). Відкриється діалогове вікно Resource (Ресурс), в якому натисніть Download (Завантаження).

Крок 6. Налаштування програми ПЛК

Наявність завантаженого проекту дає змогу перевірити програму, використовуючи оперативний режим налагодження. Перехід у цей режим роботи здійснюється за допомогою ікони Debug on/off (Налаштування В/В) в панелі інструментів після того, як відповідні робочі листки будуть відкриті (робочий листок тіла коду програми або робочий листок сітки змінних). За допомогою команд налагодження Step (Крок) і Trace (Відслідковувати), а також встановлюючи контрольні точки і, активізуючи змінні, можна відстежити коректність виконання програми.

Крок 7. Друкування проекту

Для створення документації корисно надрукувати проект. Тому система забезпечує могутній інтерфейс друкування, який дозволяє використовувати вбудований макет сторінок і вибирати для друку необхідні листки замість всього проекту.

У деяких випадках, для більш зручнішого створення проекту, можна використовувати в панелі інструментів ікону Patch POU (Вставка POU) замість Make (Створювати). Після створення основи проекту і завантаження його до ПЛК або симулятора можна поповнювати програму шляхом редагування різних робочих листків коду POU за рахунок, так би мовити, латання їх в процесі виконання контролером програми [6].

3.2. Методика програмування LD-мовою

Знайомство з програмуванням у системі MULTIPROG почнемо на прикладі розробки програми LD-мовою (Крокових діаграм), яка здійснює циклічне управління роботою двигуна, що вмикається триразовим натисненням стартової кнопки, а вимикається автоматично за 20с роботи [6].

Розробка проекту складається з кількох етапів, по яких веде Project Wizard (Майстер проекту) системи програмування MULTIPROG..

Для створення проекту запустіть систему програмування, двічі натиснувши піктограму MULTIPROG.

Щоб отримати кращий результат в процесі створення проекту, будемо використовувати запропоновані ідентифікатори й імена.

3.2.1. Створення нового проекту

Визначимо ім'я та шлях створення проекту, мову програмування і тип ПЛК, що використовується.

Використовуючи пункт меню File (файл), відкрийте діалогове вікно New Project (Новий проект), рис. 3.1;

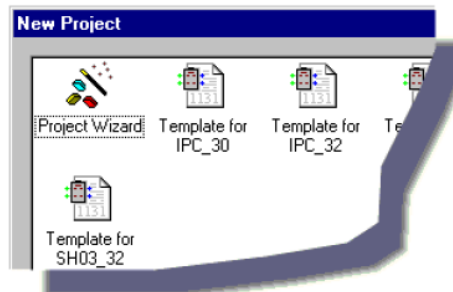


Рис. 3.1. Фрагмент діалогового вікна

- двічі клацніть по Project Wizard (Майстер проекту) і з'явиться відповідний діалог;

- введіть ім'я проекту My_first_Project (Мій перший проект). Майстер проекту збереже проект у відповідному файлі My_first_Project .mwt і створить підтеку того ж імені, де будуть запам'ятовуватися тіло коду і файли змінних.

Максимальна довжина імені проекту може складати 24 символи і не повинна містити пропуски або спецсимволи.

За умовчанням шлях проекту вводиться автоматично. Якщо бажаєте створити інший шлях, натисніть кнопку, на якій є три крапки;

- у діалозі Select Directory(Вибрати директорію) виберіть теку для нового проекту і клацніть ОК;

- натисніть кнопку Next (Далі) і у рядок Name of POU (Імя POU) введіть ім'я першого організатора програми (POU)- Main (Головний);

- оскільки зразковий проект будемо програмувати графічною LD-мовою активізуйте відповідний перемикач;
- натисніть кнопку Next (Далі) і введіть бажане ім'я конфігурації. У нашому прикладі це - Configuration (Конфігурація);
- у вікні списку виберіть тип конфігурації проекту. У нашому прикладі вибираємо IPC_32 , тому що компілятор генерує intel- код для ProConOS 3.2.
- натисніть кнопку Next (Далі) і введіть ім'я ресурсу – Resource (Ресурс).
- у вікні списку виберіть тип ресурсу - PCOS_NT. Вікно списку пропонує тільки типи CPU , які належать конфігурації, що визначена в діалозі Майстра проекту;
- натисніть кнопку Next (Далі) і введіть ім'я задачі -Task (Задача);
- у вікні списку виберіть тип задачі - Cyclic (Циклічний);
- натисніть кнопку Next (Далі). На екрані з'явиться зміст зроблених установок. Якщо помилок немає, натисніть Finish (Готов), в іншому разі скоригуйте кроки проекту, де є помилка, користуючись кнопкою „Back” (Назад).

Створений новий проект Main з'явиться у вікні проектного дерева, рис.3.2, у піддеревах Logical POU's і Physical Hardware.

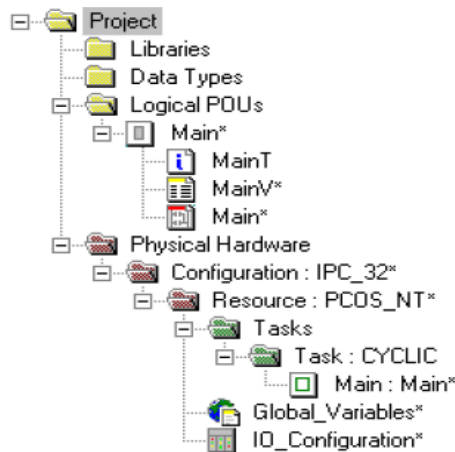


Рис. 3.2. Дерево проекту

3.2.2. Розробка коду програми

Тепер, коли новий проект створено і на екрані комп'ютера з'явилося його дерево, розробимо проектний код LD-мовою. У наступних кроках пояснимо, як створити першу LD-мережу, як оголосити властивості об'єктів, що автоматично з'явилися з першою LD- мережею, як вставити і з'єднати функціональний блок у робочому листку тіла LD- коду за допомогою Edit Wizard (Майстер редагування), як вставити і з'єднати контакт у робочому листку коду програми, як оголосити властивості контактів і котушок, як вставити другу LD-мережу і відредагувати коментар мережі.

Вставка першої LD- мережі:

- клацніть лівою клавшею миші у зручному місці робочого листка для установки мітки першої вставки;
- в панелі інструментів клацніть по іконі Contact network (Мережа контактів) і в установленому місці з'явиться LD- мережа 001 певної ширини з контактом і котушкою.



Оголошення властивостей змінних, які автоматично з'явилися з першою LD- мережею:

- двічі клацніть лівою клавiшею миші по контакту C000, щоб оголосити змінну, яка має запустити двигун. У діалоговому вікні Contact/Coil Properties (Властивості Контакт/Котушка), що з'явилося, замініть типове ім'я змінної C000 на Motor_Start;

- натисніть кнопку Apply (Застосувати), автоматично відкриється сторінка діалогу Common (Загальний). У вікні списку Usage (Вжити) виберіть VAR, а у рядку Data Type (Тип даних) - BOOL.

Таким чином, змінна оголошується як локальна, а тому вона може використовуватися тільки в ROU проекту.

Призначимо тепер змінній Motor_Start фізичну адресу симулянта модуля вводу/виводу ПЛК:

- введіть у поле I/O address (Адреса вводу/виводу) %IX0.0, тут 0.0 – позначає перший модуль і його перший канал; I – фізичний вхід; X – один-бітний розмір;

- відкрийте по черзі сторінки діалогів Local scope (Локальна компетенція) і Global scope (Глобальна компетенція) і активізуйте в них Default;

- клацніть ОК, підтверджуючи діалог Contact/Coil Properties (Властивості Контакт/Котушка). Після цього змінна є оголошеною і вставленою у відповідний список, а на робочому листку проекту замість C000 з'явиться глобальна змінна Motor_Start .

Вставка в LD- мережу лічильника натисків стартової кнопки:

Оскільки двигун має увімкнутися після триразового натиснення стартової кнопки вставимо в LD- мережу лічильник.

- Клацніть по лінії між Motor_Start і C001, щоб позначити місце вставки лічильника;

- в панелі інструментів клацніть по іконі Edit Wizard (Майстер редагування) і з таблиці Group (Група), що з'явилася на екрані, виберіть функціональний блок лічильника STU;

- двічі клацніть по блоку STU, у діалоговому вікні Variable properties (Властивості змінної), що з'явилося, змініть типове ім'я STU-1 на Motor-Count;

- натисніть послідовно кнопки Apply (Застосувати) і ОК, підтверджуючи діалог, на позначеному місці LD- мережі з'явиться лічильник;

- в панелі інструментів клацніть по іконі Edit Wizard (Майстер редагування), щоб сховати Майстра редагування.

Вставка контакту скидання лічильника:

- клацніть на функціональному блоці STU по синій точці Reset-входу;

- в панелі інструментів клацніть по іконі Add contact left (Додати контакт зліва) і на вході лічильника з'явиться контакт C002;

- активізуйте в панелі інструментів кнопку Connect objects (З'єднати об'єкти);

- клацніть по контакту C002 і перемістіть його курсором ліворуч до шини живлення 001 так, щоб він опинився під контактом Motor_Start;

- в панелі інструментів клацніть по іконі Mark (Позначка), щоб деактивувати кнопку Connect objects (З'єднати об'єкти);

Оголошення властивостей контакту скидання лічильника:

- двічі клацніть по контакту C002 і у діалоговому вікні Contact/Coil Properties (Властивості Контакт/котушка), що з'явилося, змініть ім'я C002 на Motor;

- клацніть по Apply (Застосувати), з'явиться сторінка діалогу Common(Загальний);

- у вікні списку Usage (Вжити) виберіть VAR, тобто змінна оголошується, як локальна і може використовуватися тільки в ROU проекту, тип змінної – BOOL виберіть у рядку Data Type (Тип даних);

- для призначення змінній Motor фізичної адреси симулятора модуля вводу/виводу I/O PLC у рядку I/O Address введіть %QX0.0,

тут Q – позначає фізичний вихід; 0.0 – перший вихідний модуль і його перший канал; X – позначає одинбітний розмір.

Оскільки група змінних визначена раніше, немає потреби конкретизувати Local scope (Локальна компетенція) та Global scope (Глобальна компетенція);

- клацніть ОК, і на робочому листку з'явиться змінна Motor замість C002;

- для визначення параметрів лічильника двічі клацніть по блакитній точці PV-входу, з'явиться діалогове вікно Variable properties (Властивості змінної);

- у рядку Name (Ім'я) введіть INT#3, щоб двигун (Motor) спрацював після триразового натискання стартової кнопки, тут INT – означає ціле число; # – означає константу; 3 – фактичне значення константи;

- клацніть ОК і на PV-вході лічильника з'явиться INT#3, тобто ціле число є вставленим в тіло коду;

- двічі клацніть по зеленій точці CV-виходу лічильника, з'явиться діалогове вікно Variable properties (Властивості змінної);

- в рядку Name (Ім'я) введіть Pressed - це змінна, яка запам'ятовує поточне значення лічильника;

- натисніть кнопку Apply (Застосувати), відкриється сторінка діалогу Common(Загальний);

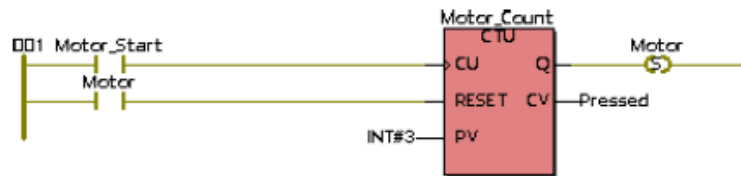
- у рядку Usage (Вжити) виберіть VAR, а у рядку Data type (Тип даних) – INT, оскільки змінною є число;

- клацніть ОК і у робочому листку поряд з CV-виходом з'явиться Pressed;

- двічі клацніть по C001, з'явиться діалогове вікно Contact/Coil Properties (Властивості Контакт/Котушка);

- зі списку вже оголошених змінних, виберіть Motor. Оскільки двигун має працювати безперервно після його увімкнення, у рядку Type (Тип) виберіть –(S)–(увімкнути вихід і зафіксувати);

- клацніть ОК, у першій LD- мережі з'явиться інструкція –(S)–.



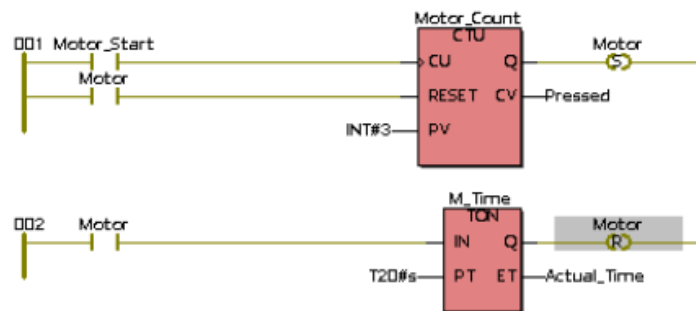
Вставка другої LD- мережі:

- клацніть лівою клавішею миші нижче першої LD- мережі, на робочому листку з'явиться мітка установки нової мережі;
- в панелі інструментів клацніть по іконі Contact Network (Мережа контактів) і на екрані з'явиться друга LD- мережа;
- двічі клацніть по контакту C003 для оголошення його властивостей;
- у діалозі Contact/Coil Properties (Властивості Контакт/Котушка) із списку вже оголошених змінних виберіть Motor і клацніть лівою клавішею миші;
- клацніть ОК для зміни у LD- мережі імені контакту C003 на Motor .

Вставка таймера, що керує тривалістю роботи двигуна:

- в панелі інструментів клацніть по іконі Edit Wizard (Майстер редагування) , на екрані з'явиться вікно Group (Група);
- клацніть по лінії другої LD- мережі між контактом Motor і C004, щоб позначити місце вставки таймера;
- за допомогою курсору у вікні Group (Група) виберіть Function blocks (Функціональні боки);
- зі списку блоків курсором виберіть таймер TON і двічі клацніть лівою клавішею миші, з'явиться діалогове вікно Variable properties (Властивості змінної);
- у рядку Name (Ім'я) введіть M_Time і послідовно натисніть кнопки Apply (Застосувати) і ОК, підтверджуючи завершення діалогу;
- клацніть по іконі Edit Wizard (Майстер редагування) для приховування вікна Group (Група);
- для визначення часу роботи таймера лівою клавішею миші двічі клацніть по блакитній точці його RT-входу;
- у діалоговому вікні Variable properties (Властивості змінної), що з'явилося, у рядку Name (Ім'я) введіть T#20s;
- Тут T – позначає префікс часу; # - означає константу; 20s – час роботи двигуна (20с);
- клацніть ОК і на RT-вході лічильника з'явиться константа часу.
- Щоб визначити змінну для ідентифікації фактичного часу роботи таймера двічі клацніть по зеленій точці ET-виходу таймера;
- у діалоговому вікні Variable properties (Властивості змінної), що з'явилося, у рядку Name (Ім'я) введіть Actual_Time (Фактичний час), як ім'я локальної змінної;
- натисніть кнопки Apply (Застосувати), відкриється сторінка діалогу Common (Загальний);
- зі списку Data_Type (Тип даних) виберіть Time (Час);
- клацніть ОК, на ET-виході таймера з'явиться Actual_Time;

- двічі клацніть по котушці C004, відкриється діалогове вікно Contact/Coil Properties (Властивості Контакт/Котушка);
- для зупинки двигуна виберіть курсором зі списку змінних ім'я Motor, а у вікні Type (Тип) - котушку –(R)– (вимкнути вихід і зафіксувати);
- клацніть ОК.



Присвоєння імені розробленої мережі:

- двічі клацніть по лівій шині живлення першої LD- мережі, з'явиться діалогове вікно Comment (Коментар);
- у діалоговому вікні Comment (Коментар) надрукуйте Circuit (Схема) і клацніть Fout>> (Шрифт);
- для зміни властивостей шрифту у вікні, що з'явилося, виберіть блакитний колір і розмір 20;
- клацніть ОК у вікні Fout (Шрифт) і у вікні Comment (Коментар). Над розробленою мережею з'явиться ім'я Circuit.

3.2.3. Компіляція проекту

Тепер, коли редагування закінчено, зробимо компіляцію проекту. Протягом компіляції вміст робочих листків відтрансльовується і перетворюється у спеціальний код, який може бути виконаний ПЛК.

Оскільки у цьому прикладі замість ПЛК використовується симулятор, активізуємо його:

- у дереві проекту клацніть по теці Resource (Ресурс);
- клацніть правою клавішею миші і виберіть з контекстного меню Settings (Установки);
- у вікні Resource settings for IRC_32 (Установки ресурсу для IRC-32), що з'явилося, активізуйте Simulation 1 (Симуляція 1), якщо це необхідно і закрийте діалог, клацнувши ОК;
- в панелі інструментів клацніть по іконі Make (Утворювати) ;
- у вікні Build (Конструкція) внизу екрана відобразяться помилки (Error) і попередження (Warning), якщо будуть знайдені протягом компіляції.

Помилки синтаксичного або структурного характеру перешкоджають завершенню процесу компіляції.

Попередження вказують на потенційні проблеми, подібно змінній, якої немає, а вона використовується. Попередження не перешкоджають завершенню процесу компіляції. Попередженнями можна нехтувати, а помилки потрібно виправити.

- Для виводу на екран списку знайдених помилок у вікні Build (Конструкція) клацніть по Error (Помилки);

- для виводу на екран списку попереджень у вікні Build (Конструкція) клацніть по Warning (Попередження);
 - двічі клацніть по вказаній помилці або попередженню, відкриється відповідний робочий листок;
 - виправіть усі помилки і попередження і повторно компілюйте проект, клацнувши в панелі інструментів по іконі Make (Утворювати);
- Тільки після цього можна завантажувати програму до ПЛК або симулятора.

3.2.4. Завантаження проекту до ПЛК або симулятора

Зв'язок з ПЛК або симулятором здійснюється завдяки діалогу Resource (Ресурс), який керує контролером.

- Клацніть по іконі Project Control Dialog (Діалогове управління проектом), з'явиться діалогове вікно Resource (Ресурс) для управління ПЛК або симулятором;
 - натисніть кнопку Download (Завантаження), з'явиться відповідне діалогове вікно ;
 - продублюйте у цьому вікні натиснення на кнопку Download і проект завантажиться у пам'ять ПЛК або симулятор;
 - успішний процес завантаження проекту супроводжується бігом синьої стрічки внизу екрана;
 - у діалозі управління Resource (Ресурс) натисніть кнопку Cold (Холодний) для холодного пуску симулятора;
- Стан діалогового вікна Resource (Ресурс) змінюється кнопками Stop (Зупинка) і Cold (Холодний).

3.2.5. Налаштування проекту

Робочі листки проекту можуть перемикатися з режиму редагування на режим налагодження і навпаки, за допомогою ікони Debug on/off (Налаштування В/В) в панелі інструментів. Коли режим налагодження активізовано, стан і поточні значення змінних на робочому листку проекту показані різними кольорами: блакитний – хибний, червоний – істинний.

Режим налагодження використовується для знаходження помилок програмування і переконання, що програма ПЛК працює правильно. Для запуску програми у діалозі управління Resource (Ресурс) натисніть кнопку Cold (Холодний) .

- Для виводу на екран симулятора модулів I/O (Вводу/виводу) клацніть по Demoio - Driver, що видно внизу монітора;
- розмістіть симулятор таким чином, щоб проект на робочому листку не був прихований;
- тричі клацніть по зеленому віртуальному світлодіоду з адресою IX0.0 і спостерігайте реакцію схеми в робочому листку проекту.

Motor (Двигун) почне працювати після того, як з'явиться цифра три біля Pressed, тому що поточне значення CV- виходу на блоці Motor_Count досягає встановленого значення на PV- вході того ж блока.

Коли котушка Motor у мережі 001 вмикає двигун, вмикається контакт Motor у мережі 002 і таймер M_Time починає відлік часу протягом 20с. Тривалість руху двигуна продовжується доки час, що фіксує Actual_Time (Фактичний час) біля ET-виходу таймера, не співпаде з уставкою біля RT-виходу (T#20s). По завершенню роботи таймера котушка –(R)– у мережі 002 вимикає двигун (Motor), що у мережі 001.

3.2.6. Оперативне редагування

Оперативне редагування можливе без зупинки програми, яка реалізується на ПЛК або симуляторі. Цей режим викликається, клацаючи в панелі інструментів по іконі Patch POU (Вставка POU) .

Коли використовується режим Patch POU (Вставка POU), зміни, що зроблені в кодї програми автоматично завантажуються до ПЛК або симулятора. Протягом всього часу роботи в режимі Patch POU (Вставка POU) виконання коду програми не припиняється.

Як приклад роботи в режимі Patch POU (Вставка POU), вставимо у програму змінну Emergency_Stop(Непередбачена зупинка). Активізація Emergency_Stop (Непередбачена зупинка) негайно зупинить двигун.

- Клацніть по іконі Debug on/off (Налагодження В/В), щоб перевести робочий листок коду програми в режим редагування. Проте, ресурс так само, як і реальний контролер, працює;

- установіть курсор на робочому листку коду програми нижче шини живлення OO2 і клацніть лівою клавішею;

- в панелі інструментів клацніть по іконі Contact Network (Мережа контактів), і на робочому листку з'явиться нова LD- мережа 003;

- двічі клацніть по контакту C005, відкриється діалог Contact/Coil Properties (Властивості Контакт/Котушка). У рядку Name (Ім'я) замініть типове ім'я C005 на Emergency_Stop (Непередбачена зупинка);

- клацніть ОК, автоматично відкриється сторінка діалогу Common(Загальний);

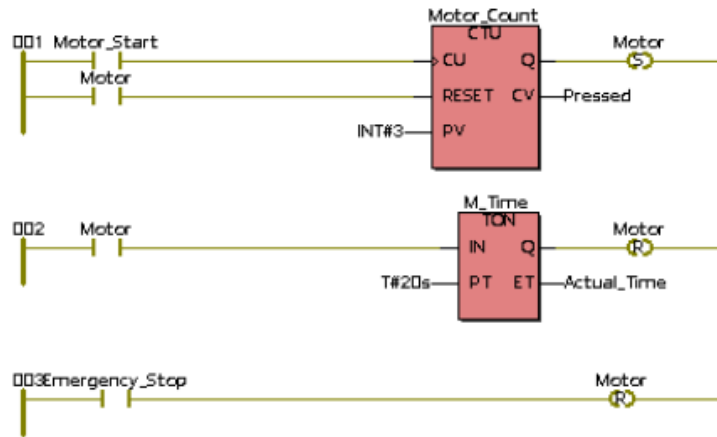
- з вікна списку Usage (Вжити) виберіть VAR_EXTERNAL, декларуючи таким чином, Emergency_Stop (Непередбачена зупинка), як глобальну змінну.

Для непередбаченої зупинки двигуна використаємо другий канал симулятора I/O:

- у рядку I/O Address (Адреса В/В) введіть % IXO.1, клацніть ОК, у мережі 003 з'явиться змінна Emergency_Stop замість C005;

- двічі клацніть по контакту C006, з'явиться діалог Contact/Coil Properties (Властивості Контакт/Котушка);

- з вікна списку Type (Тип) виберіть –(R)–, а з вікна списку Variable (Змінна) – Motor, клацніть ОК, і на робочому листку з'явиться третя відкоригована LD- мережа.



Тепер, коли код програми змінений, використаємо режим Patch POU (Вставка POU), щоб прокомпілювати змінні і завантажити їх до симулятора без його зупинки.

- В панелі інструментів клацніть по іконі Patch POU (Вставка POU), змінений код проекту прокомпілюється і завантажиться у симулятор I/O.

Після того, як режим Patch POU (Вставка POU) успішно завершиться, робочий листок буде автоматично переведений до оперативного режиму.

- Клацніть по Demoio-Driver, відкриється симулятор I/O;
- тричі подвійним клацанням по нульовому віртуальному світлодіоду нульового модуля I/O змініть стан біту;
- використовуючи новий контакт Emergency_Stop (Непередбачена зупинка), негайно зупиніть двигун Motor, клацаючи по світлодіоду 1-го нульового модуля вводу.

3.2.7. Створення певної функції користувача

Система програмування дозволяє користувачеві створювати власну функцію. У цьому проекті створимо функцію користувача, яка має перелічити кількість активізацій двигуна. Зробимо це, використовуючи текстову ST-мову. Перед створенням функції виберіть в меню Extras (Доповнення) і увійдіть в Options → Graphical Editor (Опції → Графічний редактор);

- активізуйте Function with EN/ENO і клацніть OK; тут EN/ENO – означає додатковий логічний вхід EN і додатковий логічний вихід ENO для функцій стандарту IEC61131 в мовах програмування LD і FBD;

- в панелі інструментів клацніть по іконі Debug on/off (Налагодження В/В) для переходу в режим редагування;

- в панелі інструментів клацніть по іконі Add Function (Додати функцію) для створення функції користувача, з'явиться вікно Insert (Вставка);

- в полі Name (Ім'я) введіть Cycle_Count (Рахувати цикли), як ім'я функції, крапками позначте Function (Функція) у групі Type (Вид) і ST у групі Language (Мова). У рядку Data Type of Return Value (Тип даних, значення, що повертається) конкретизуйте, який тип даних застосовується до виходу функцій. Змінна, що зв'язана з виходом функції, має відповідати типу даних, що повертаються. У нашому прикладі використовуємо INT зі списку, що надається;

- клацніть OK, функція додається до дерева проекту. Зірка наприкінці імені створеної функції вказує, що новий POU ще не скомпільований;

- клацніть по Cycle_Count (Рахувати цикли), відкриється робочий листок Cycle_Count (Рахувати цикли) для редагування ST-коду;
- у відкритому листку надрукуйте наступний код:
 1. Cycle_Count := Count+1;
 Цей рядок коду генеруватиме значення, яке має безперервно збільшуватися після кожного запуску двигуна;
- розмістіть курсор на змінній Count (Рахувати) і в панелі інструментів клацніть по іконі Variables(Змінні), з'явиться діалогове вікно Variables(Змінні);
- активізуйте Common (Загальний) у вікні Variables (Змінні) і декларуйте локальну змінну:
 - у рядку Usage (Вжити) виберіть VAR_INPUT;
 - у рядку Data Type (Тип даних) виберіть INT;
 - клацніть ОК, підтверджуючи діалог. Декларована змінна автоматично вставиться у робочий листок змінних нового ST-POU;
 - закрийте ST- робочий листок і збережіть змінні.
 Після закриття нового ST-робочого листка користувача функція залишається доступною Майстру редагування (Editor Wizard) і може бути встановленою в інші робочі листки проекту.
- Щоб викликати створену функцію користувача Cycle_Count (Рахувати цикли) із головного POU програми:
 - у дереві проекту подвійним клацанням по Main (Головний) відкрийте робочий листок;
 - клацніть нижче LD- мережі 003, щоб позначити місце вставки нової мережі;
 - в панелі інструментів клацніть по іконі Contact Network (Мережа контактів) , з'явиться мережа 004;
 - позначте курсором лінію сполучення між C007 та C008 і клацніть лівою клавішею миші;
 - в панелі інструментів клацніть по іконі Edit Wizard (Майстер редагування) , щоб активізувати Майстра редагування;
 - у вікні Group (Група) виберіть My_First_Project, який містить функцію користувача проекту Cycle_Count (Рахувати цикли);
 - двічі клацніть по Cycle_Count (Рахувати цикли) і функція з'явиться на позначеному місці мережі 004;
 - в панелі інструментів клацніть по іконі Edit Wizard (Майстер редагування), вікно Group (Група) закриється;
 З'єднаємо нову змінну з входом Count (Підрахунок), щоб бачити її значення:
 - двічі клацніть по блакитній точці на вході Count (Підрахунок) функції Cycle_Count (Рахувати цикли), з'явиться діалог Variable Properties (Властивості змінної);
 - у полі Name (Ім'я) декларуйте локальну змінну Motor_Cycle;
 - клацніть ОК, підтверджуючи діалог, автоматично активізується вікно Common (Загальний);
 - у діалозі Common (Загальний) у рядку списку Name (Ім'я) виберіть VAR, а у вікні списку Data Type виберіть INT;

- клацніть ОК, закриється вікно Variable properties (Властивості змінної) і змінна Motor_Cycle потрапляє у середину тіла коду проекту, а її декларація – у список локальних змінних робочого листка.

З'єднаємо також змінну Motor_Cycle з виходом функції Cycle_Count (Рахувати цикли):

- двічі клацніть по зеленій точці на виході функції Cycle_Count (Рахувати цикли), з'явиться діалог Variable Properties (Властивості змінної);

- зі списку змінної виберіть курсором Motor_Cycle і клацніть лівою клавішею миші;

- клацніть ОК, діалог Variable Properties закриється і змінна Motor_Cycle увійде у відповідний список.

Змінимо ім'я контакту 007 у мережі 004 на Motor (Двигун):

- двічі клацніть по контакту C007 у робочому листку проекту, відкриється діалог Contact/Coil Properties (Властивості Контакт/Котушка);

- зі списку змінних виберіть курсором Motor (Двигун) і клацніть лівою клавішею миші;

- клацніть ОК, і вибране ім'я замінить C007;

- в панелі інструментів клацніть по іконі Edit Wizard (Майстер редагування), з'явиться таблиця Group (Група);

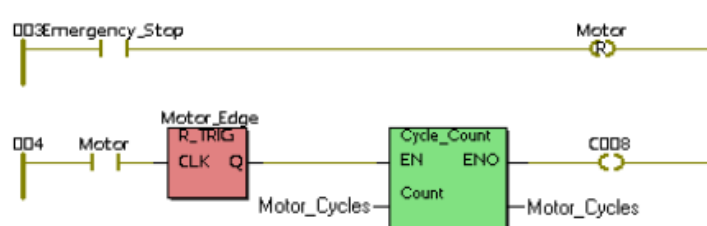
- з наведеного списку виберіть курсором Function Blocks (Функціональні блоки) і клацніть лівою клавішею, з'явиться перелік блоків;

- подвійно клацніть по функціональному блоку детектора імпульсів R_TRIG, з'явиться діалог Variable Properties (Властивості змінної);

- у полі Name (Ім'я) введіть Motor_Edge (Двигун_Фронт), клацніть ОК;

- закрийте діалог Variable Properties (Властивості змінної), клацніть ОК і функціональний блок R_TRIG потрапить у LD- мережу C004, а його декларація – у код програми;

- клацніть по іконі Edit Wizard (Майстер редагування), таблиця Group (Група) зникне з екрана;



- двічі клацніть по котушці C008, з'явиться діалогове вікно Contact/Coil Properties (Властивості Контакт/Котушка);

- у рядку Name (Ім'я) введіть Motor_Counted (Мотор_Рахівник) і клацніть ОК, активізується діалогове вікно Common (Загальний);

- у вікні списку Usage (Вжити) виберіть VAR, а у вікні списку Data Type (Тип даних) виберіть BOOL і клацніть ОК, діалогове вікно зникне;

- в панелі інструментів клацніть по іконі Make (Створювати) і проект скомпілюється.

Якщо помилки є, відкоригуйте проект, якщо їх немає, то створення зразкового проекту завершено і його треба завантажити, використовуючи відомий шлях Project Control Dialog → Download → Download;

- клацніть Cold (Холодний) у діалозі Resource (Ресурс) для холодного пуску симулятора ПЛК.

Перевіримо поведінку програми:

- в панелі інструментів клацніть по іконі Debug on/off (Налагодження В/В), щоб перейти в оперативний режим (всі змінні позначаються різними кольорами);

- клацніть по Demoio_Driver, що внизу екрана і симулятор I/O - відкриється;

- тричі подвійно клацніть по нульовому світлодіоду нульового модуля вводу In.

Програма почне виконуватися, а біля виходу Actual_Time (Фактичний час) таймера M_Time будуть змінюватися секунди затримки. При цьому з кожним новим запуском програми значення змінної Motor_Cycles у функції Cycles_Count (Рахувати_Цикли) збільшується на одиницю, рис.3.3.

З поточного робочого листка можна перейти на функцію користувача, якщо позначити Cycles_Count (Рахувати_Цикли):

- двічі клацніть по функції Cycles_Count (Рахувати_Цикли) і в LD-робочому листку з'явиться діалог;

- підтвердіть його, натиснувши Yes(Так), робочий листок тіла коду функції Cycles_Count (Рахувати – Цикли) відкриється і в ньому буде показано поточне значення лічильника;

- закрийте робочий листок тіла коду, щоб повернутися до LD- мережі.

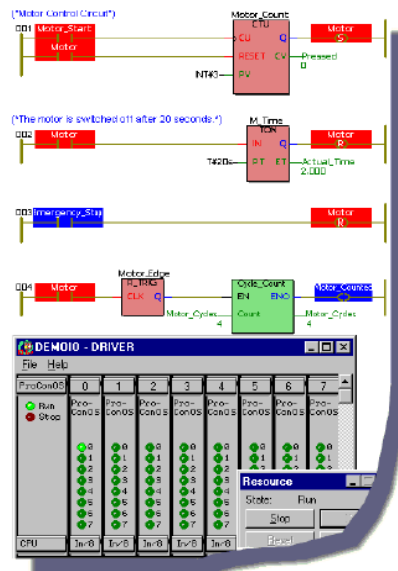


Рис. 3.3. Фрагмент робочого листка проекту в режимі налагодження.

3.2.8. Додаткові можливості оперативного редагування

3.2.8.1. Вікно довідкове і спостереження змінних

Вікно спостереження змінних – це могутній інструмент, що дозволяє легко вставити різні змінні у список і спостерігати динаміку їх поведінки. Як тільки змінна додана у вікно спостереження, не потрібно відкривати

відповідний робочий листок для контролю за нею. Таким чином, можна зосередитися лише на змінних, які потрібні для аналізу роботи програми.

- Клацніть по іконі Debug on/off (Налагодження В/В), щоб перевести робочий листок в оперативний режим;

- в панелі інструментів клацніть по іконі Watch Window (Вікно спостереження), унизу екрана з'явиться вікно спостереження;

- установіть курсор на змінну Motor_Start і клацніть правою клавішею миші;

- у контекстному меню, що з'явилося, виберіть Add to Watch Window (Додати до вікна спостереження) і клацніть лівою клавішею миші, змінна Motor_Start буде внесена до списку;

- продублюйте такі самі дії зі змінними Pressed (Натиснути) і Actual_Time (Фактичний час). У вікні спостереження з'явиться список з трьох змінних;

- натисніть Demoio-Driver для виклику симулятора I/O;

- тричі подвійно клацніть по нульовому світлодіоду нульового модуля I/O, щоб запустити програму, і спостерігайте стан змінних одночасно на робочому листку і у вікні спостереження.

Окрім вікна спостереження змінних при оперативному редагуванні дуже зручним є перехресне довідкове вікно. Перехресний довідковий список містить всі змінні, функціональні блоки, переходи, мітки та зв'язки, які використовувалися у проекті. Цей інструмент особливо корисний при налагодженні і визначенні помилок.

- В панелі інструментів клацніть по іконі Cross Reference (Перехресна довідка), унизу екрана відкриється перехресне довідкове вікно, якщо вже не відкрито;

- установіть курсор на полі довідкового вікна і клацніть правою клавішею миші, відкриється контекстне меню;

- виберіть пункт меню Build Cross Reference (Створення перехресної довідки) і клацніть лівою клавішею миші, відкриється довідковий список;

- двічі клацніть по будь-якій змінній, що у вікні перехресної довідки. Вікно відкриє робочий листок, в якому ця змінна підсвічується. Підсвічується також ця змінна у вікні перехресної довідки;

- закрийте вікно перехресної довідки і вікно повідомлення, клацаючи послідовно в панелі інструментів по іконах Cross Reference (Перехресна довідка) і Message (Повідомлення) .

3.2.8.2. Контрольні точки

Контрольні точки можуть бути встановлені в робочі листки, використовуючи праву частину діалогу Debug: Resource (Налагодження: Ресурс). Коли контрольна точка встановлена, виконання програми зупиниться у місці її установки і поновлюється, коли користувач змушує це поновлення. Система програмування забезпечує виконання програми поки не буде досягнута наступна контрольна точка (Єдиний крок) або поки та ж сама контрольна точка не буде досягнута знову (Єдиний цикл).

Якщо контрольна точка досягається, стан ПЛК змінюється на HALT (Зупинник). З'являється вікно з кнопками GO (Пуск), Step (Крок) і Trace (Відслідковування).

Пуск. Натискання кнопки GO (Пуск) відновлює виконання програми до наступної контрольної точки.

Крок. Натискання кнопки Step (Крок) примушує програму виконати наступний крок.

Відслідковування. Якщо функція або функціональний блок, що визначив користувач, досягнуті у процесі виконання програми, тіло коду функції або функціонального блока відкривається для покрокового налагодження.

Треба бути дуже обережними при використанні контрольних точок під час роботи ПЛК, оскільки контрольна точка фактично зупиняє виконання програми.

Щоб спостерігати дію контрольної точки, робочий листок повинен знаходитися в оперативному режимі, тобто ікона Debug on/off (Налагодження В/В) має бути активізована:

- клацніть по іконі Project Control Dialog (Діалогове управління проектом), щоб відкрити діалог Resource (Ресурс) управління ресурсом;

- в робочому листку проекту двічі клацніть по Motor_Start, з'явиться діалог Debug: Resource (Налагодження: Ресурс);

- виберіть Set (Вставити) для встановлення точки на вибраній змінній. При цьому в робочому листку змінна Motor_Start підсвічується помаранчевим кольором;

- у діалозі Resource (Ресурс) натисніть кнопку GO (Пуск) для поновлення виконання програми до наступної контрольної точки.

Оскільки встановлена тільки одна точка, програма знову зупиняється на Motor_Start, завершуючи перший цикл.

- Клацніть Step (Крок) декілька разів і помітьте, що помаранчеве підсвічування кожного разу переміщується на наступну інструкцію, вказуючи точку, де виконання програми зупиниться. При цьому змінна Motor_Start має червоний колір, показуючи, де була встановлена контрольна точка;

- двічі клацніть по Motor_Start, з'явиться діалог Debug: Resource (Налагодження: Ресурс);

- натисніть Reset (Скидання), червона відмітка на Motor_Start зникне, залишиться тільки помаранчева, вказуючи на нове місце контрольної точки;

- для продовження програми у діалозі Resource (Ресурс) клацніть GO (Пуск);

- для переходу в режим редагування проекту клацніть по іконі Debug on/off (Налагодження В/В) ;

- для зупинення симулятора клацніть Stop (Стоп) у діалозі Resource (Ресурс);

- клацніть Close (Закрити) у діалозі Resource (Ресурс), щоб закрити діалогове вікно керування.

3.2.8.3. Зміна часу циклу задачі

Система програмування дозволяє змінювати цикл задачі, тобто інтервал часу, в якому задача виконується. Зменшення часу циклу задачі збільшує

швидкість процесу її виконання. У даному прикладі змінимо час циклу задачі від 100ms до 90ms.

Найменший цикл задачі залежить від типу ПЛК, що використовується:

- в панелі інструментів клацніть по іконі Debug on/off (Налагодження В/В) для переходу в режим редагування, якщо це потрібно. Щоб змінити конфігурацію задачі у піддереві Physical Hardware (Фізична апаратура) дерева проекту, активізуйте Task: Cyclic (Задача: Цикл) і клацніть правою клав'яшею миші, з'явиться контекстне меню;
- з контекстного меню виберіть Setting (Установки), відкриється діалог Task setting for IPC_32 (Задача установки для IPC_32);
- у рядку Interval (Інтервал) введіть 90 замість 100, клацніть ОК для підтвердження діалогу;
- в панелі інструментів клацніть по іконі Make (Створювати) і проект скомпілюється;
- для завантаження проекту, в панелі інструментів клацніть по іконі Project Control Dialog (Діалог управління проектом), і послідовно натисніть Download (Завантаження) → Download (Завантаження) у діалогах, що з'являються на екрані;
- підтвердіть завантаження, натискаючи Yes (Так).

3.2.8.4. Зміна конфігурації симулятора

Configuration I/O (Конфігурація В/В) у проектному дереві використовується для редагування I/O у робочому листку конфігурації. Пояснення з використання Configuration I/O (Конфігурація В/В) зробимо на прикладі збільшення кількості вхідних модулів в існуючому симуляторі з 8 до 10:

- у піддереві Physical Hardware (Фізична апаратура) двічі клацніть по Configuration I/O (Конфігурація В/В). З'явиться діалог Configuration I/O (Конфігурація В/В);
- у діалозі Configuration I/O (Конфігурація) натисніть кнопку Properties (Властивості);
- у полі Length (Довжина) введіть 10 і на клавіатурі ЕОМ натисніть клавішу <<Тав>>. У рядку End address (Кінцева адреса) з'явиться адреса %ІВ9;
- клацніть ОК для підтвердження нових властивостей і клацніть ОК у діалозі Configuration I/O (Конфігурація В/В), щоб повернутися до програмування;
- в панелі інструментів клацніть по іконі Make (Створювати), щоб скомпілювати проект.
- завантажте проект, як було наведено раніше;
- клацніть по Demoio-Driver внизу екрана і на симуляторі I/O, що з'явився, замість 8 модулів буде 10.

3.2.9. Друкування проектної документації

Після створення проекту корисно надрукувати документацію. Система програмування проходить декілька етапів у створенні документації: попереднього перегляду поточної сторінки, визначення установки принтера і друку повного проекту або одного робочого листка. Система програмування

за бажанням користувача дозволяє друкувати різні сторінки проекту. При друкуванні створеного проекту автоматично використовується типова сторінка фірми виробника системи програмування.

Вибір принтера:

- виберіть пункт меню File (Файл) і клацніть лівою клавішею миші;
- увійдіть в підменю Print (Друк), відкриється стандартний Windows (Вікна);
- введіть назву принтера і клацніть ОК;
- з меню File (Файл) виберіть Print Project (Друк проекту) і клацніть лівою клавішею миші;
- у діалоговому вікні Print Project (Друк проекту), що з'явилося, зніміть позначки частин проекту, які не бажано друкувати;
- увімкніть принтер і клацніть Print (Друк).

Попередній перегляд і друк єдиного робочого листка

Попередній перегляд друку дозволяє побачити, який вигляд матиме робочий листок після друкування і якщо потрібно скоригувати його.

Щоб викликати попередній перегляд:

- переконайтесь, що робочий листок, який треба переглянути дійсно у вікні;
- виберіть в меню шлях File → Print Preview(Файл → Перегляд друку) і зробіть попередній огляд друку активного робочого листка;
- щоб надрукувати один робочий листок, що показаний, в головному меню натисніть кнопку Print (Друк) .