

# Лабораторна робота №1

## Розробка проекту користувача в системі програмування MULTIPROG і робота в редакторі проектного дерева

### Створення проекту користувача

Для створення проекту користувача необхідно запуснути систему програмування, двічі клацнувши по піктограм MULTIPROG на робочому столі ЕОМ. Використовуючи пункт меню File, відкрити діалогове вікно New Project, рис. 1.1.

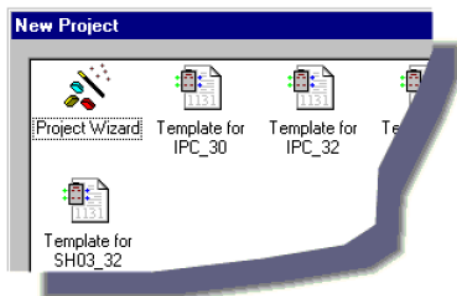


Рис. 1.1. Фрагмент діалогового вікна

Двічі клацнути по Project Wizard (Майстер проекту). У діалоговому вікні, що з'явиться, ввести 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 замовчуванням шлях проекту вводиться автоматично. Якщо бажаєте створити інший шлях, натисніть кнопку, на якій є три крапки і у діалозі Select Directory (Вибрати директорію) виберіть теку для нового проекту, клацніть ОК.

Натисніть кнопку Next (Далі) і у рядок Name of POU (Ім'я 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 зі списком виберіть тип конфігурації проекту IPC\_40, тому що компілятор генерує intel- код для ProConOS 4.8.

Натисніть кнопку Next (Далі) і введіть ім'я ресурсу – Resource (Ресурс). У вікні зі списком виберіть тип ресурсу – PCOS\_NT.

Натисніть кнопку Next (Далі) і введіть ім'я задачі –Task (Задача). У вікні зі списком 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.1.2. Проект складається з бібліотеки (Libraries), типу даних (Data Type), програмних

модулів (Logical POUs) і набору елементів конфігурації, які зображені у піддереві Physical Hardware (Фізична апаратура).

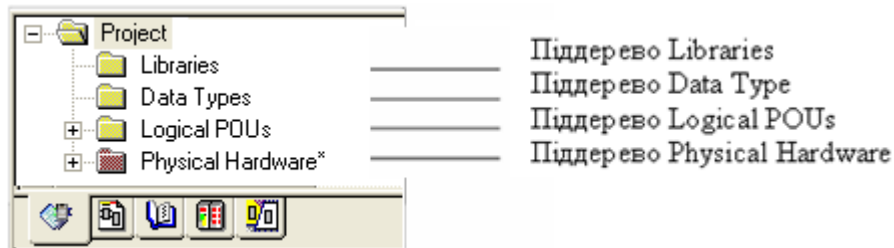


Рис.1.2 Дерево проекту

## ***Теки і робота в редакторі проектного дерева***

### **Тека бібліотеки**

**Бібліотеки** (Libraries) – це POUs, що оголошені як бібліотеки. Існують бібліотеки мікропрограм і бібліотеки користувача.

Бібліотеки мікропрограм – це бібліотеки, які створені виробниками PLC. Файл такої бібліотеки має розширення \*.fwl.

Бібліотеки, що призначені користувачем, – це проекти, функціональні блоки та функції, які створені раніше. Маючи вже закінчений проект, створений функціональний блок або функцію, можна використовувати їх POUs і робочі листки у нових проектах багато разів. Тобто користувачеві немає потреби створювати тіло коду, яке вже існує.

Бібліотека користувача знаходиться у файлі, з розширенням \*.mwl.

При створенні бібліотеки потрібно дотримуватися наступного:

- для оголошення бібліотек можуть використовуватися тільки POUs;
- в оглядовому режимі робочі листки POUs оголошених бібліотек користувача можуть бути тільки показані і не можуть редагуватися;
- у той же час робочі листки бібліотеки користувача, що відкриті в оглядовому режимі, можуть перемикатися в оперативний режим, в якому можливе їх налагодження.

**Щоб оголосити бібліотеку необхідно** у проектному дереві позначити теку піддерева Libraries (Бібліотеки). В панелі інструментів клацнути по іконі Add Object (Додати об'єкт) або вибрати пункт меню Add Library (Додати бібліотеку) в підменю Project (Проект).

У діалоговому вікні Include library (Вставити бібліотеку), що з'явиться, у рядку «Папка» вибрати і відкрити теку MULTIPROG, а в теці Project (Проект) позначити файл з розширенням \*.mwl, який бажано оголосити як бібліотеку, і у діалоговому вікні Include library (Включити бібліотеку) натиснути кнопку Include. Ікона створеної бібліотеки вставиться у проектне дерево.

Бібліотеки мають власне піддереву у проектному дереві. Можна показати повне проектне дерево або тільки піддереву “Бібліотеки”, клацаючи по етикетці Libraries (Бібліотеки), що внизу вікна проектного дерева.

Піддереву Libraries (Бібліотеки), рис.1.3, складається з двох або більше ікон. Перша ікона – це вузол директорії. Елемент цього вузла директорії являє оголошену бібліотеку.



Рис. 1.3 Піддерево бібліотеки користувача

*Примітка.* Якщо при створенні бібліотеки ікона Add Object (Додати об'єкт) інертна, то доступ до бібліотеки захищений. Для отримання доступу необхідно ввести пароль, використовуючи пункт меню Enter password (Ввести пароль) в підменю File (Файл).

*Для видалення бібліотеки* треба позначити її ікону в проектному дереві і натиснути клавішу "DEL", з'явиться попередження, в якому потрібно підтвердити видалення.

### Тека типу даних

*Типи даних*, які призначені користувачем, оголошуються у робочому листку типу даних, обов'язково зі словами TYPE і END\_TYPE. До типу даних користувача відносяться перелічення, структури, масиви і масиви структур.

*Щоб вставити робочі листки типу даних у проектне дерево*, треба позначити теку Data Types (Типи даних). В панелі інструментів клацнути по іконі Add Object (Додати об'єкт) або в підменю Project (Проект) вибрати пункт Add Data Type (Додати тип даних).

У діалоговому вікні Insert (Вставка), що з'явиться ввести ім'я нового робочого листка типу даних і підтвердити діалог. Новий робочий листок типу даних буде вставлено у відповідне піддерево, рис. 1.4.

Щоб відкрити робочий листок типу даних з текстовим редактором, необхідно у дереві проекту подвійно клацнути по іконі Data Types Worksheet (Робочий листок типу даних).

*Примітка.* Якщо ікона Add Object (Додати об'єкт) інертна, захист з використанням пароля активізований. Тому треба ввести пароль використовуючи пункт меню Enter password (Ввести пароль) в підменю File (Файл).

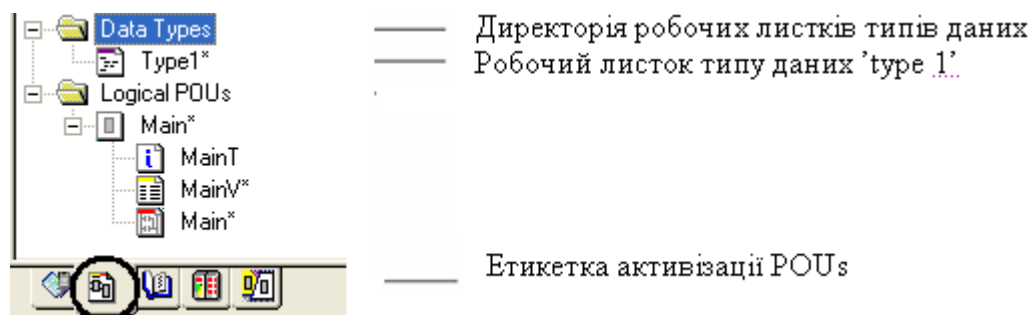


Рис. 1.4 Піддерево типів даних

Піддерево Data Type (Тип даних) можна оглядати, показуючи повне проектне дерево або тільки піддерева Data Type (Тип даних) і Logical POU's (Логічні POU's). Для цього в меню проектного дерева, що розташоване внизу вікна, треба клацнути по етикетці POU's.

## Тека POUs

Програми, функціональні блоки і функції можуть редагуватися у піддереві Logical POUs (Логічні POUs) проектного дерева. Кожне POU складається з декількох робочих листків, рис. 1.5.

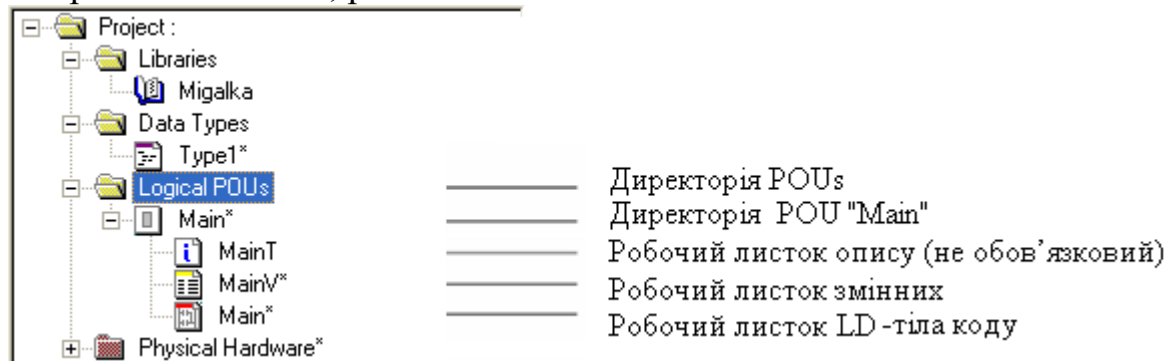


Рис. 1.5. Структура директорії POUs

Робочий листок опису створюється для документації і містить текст, введений користувачем, проте він не є обов'язковим. Робочий листок змінних містить декларовані локальні змінні. Робочий листок тіла коду містить фактичний код, що створювався однією з мов програмування. Ці робочі листки зображуються графічно іконами і мають назву, яка співпадає з назвою власного POU.

При програмуванні SFC – мовою робочі листки POUs доповнюються робочими листками дій та переходів. Якщо робочий листок коду відкритий, можна легко відкрити зв'язаний з ним робочий листок змінних, двічі клацнувши по відповідній іконі у піддереві Logical POUs (Логічні POUs) або вибравши пункт меню Open Variables Worksheet (Відкрити робочий листок змінних) у підменю View (Огляд), або клацаючи по іконі Variables Worksheet (Змінні робочого листка) в панелі інструментів.

Піддерево Logical POUs (Логічні POUs) містить усі POUs, які використовуються в проекті.

Можна оглядати піддерево Logical POUs (Логічні POUs), показуючи повне проектне дерево або тільки піддерева Data types (Тип даних) і Logical POUs (Логічні POUs). Для цього необхідно клацнути по етикетці POUs в меню проектного дерева, що розташоване внизу вікна.

**Щоб вставити POUs у проектне дерево,** треба позначити теку Logical POUs (Логічні POUs) і в панелі інструментів клацнути по одній з наступних ікон:

Add program ( Додати програму), щоб вставити програму.

Add FB ( Додати FB ), щоб вставити функціональний блок.

Add Function (Додати функцію), щоб вставити функцію.

Альтернативно можна вибрати відповідний пункт меню у підменю Project > Add POU.

З'явиться діалог Insert (Вставка), в якому ввести ім'я нового POU, вибрати бажану мову, при необхідності ввести PLC і/або тип процесора і, якщо потрібно, позначити перемикач Use Reserve (Використання резерву). Підтвердити діалог.

Use Reserve (Використання резерву) означає, який резерв пам'яті POU використовується для Patch POU (Вставка POU). Значення Use Reserve (Використання резерву) записано у діалозі Data Area (Область даних), який викликається натисканням кнопки Data Area (Область даних) у діалозі Resource setting (Установки ресурсу) вашого ПЛК.

Після цих дій новий POU є вставлений у проектне дерево. Він містить робочий листок тіла коду, робочий листок змінних і робочий листок опису. Нові робочі листки в проектному дереві позначені зірочкою, це означає, що вони ще не компільовані.

*Примітка.* Якщо право на вставлення POU у проектне дерево обмежене паролем користувача, з'явиться повідомлення про помилку. В даному випадку необхідно зареєструватися в проекті, використовуючи дійсний пароль проекту. Для цього потрібно відкрити підменю File (Файл) і вибрати пункт меню Enter password (Ввести пароль).

### Теки елементів конфігурації

Елементи конфігурації у проектному дереві зображені графічно. Система програмування відображає структуру елементів конфігурації у піддереві Physical Hardware (Фізична апаратура).

Протягом створення робочого коду програм и функціональних блоків у піддереві Physical Hardware (Фізична апаратура), рис. 1.6, додаються їх екземпляри, а також редагуються окремі робочі листки глобальних змінних.

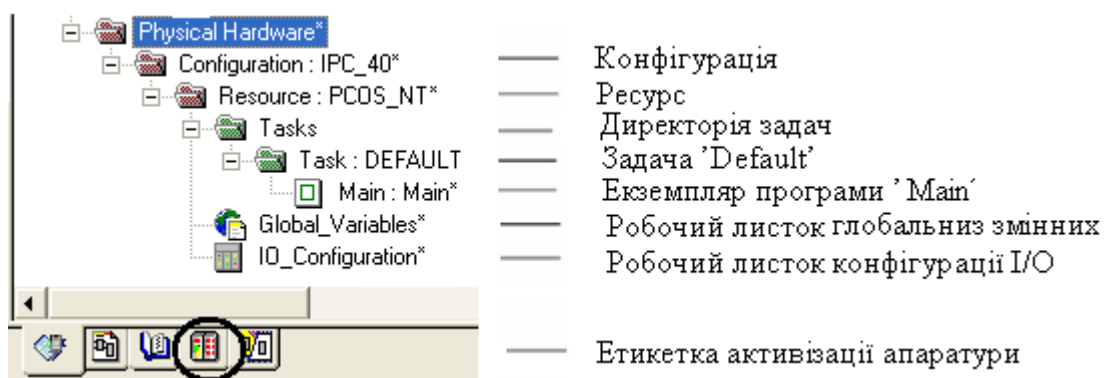


Рис. 1.6 Структура піддерева Physical Hardware

Як результат кожний екземпляр програми або функціонального блока має власний набір даних з безпосереднім доступом.

Екземпляри програми і функціонального блока корисні, якщо маємо, наприклад, дві ідентичні машини свердління, які керуються одним ПЛК через різні модулі вводу-виводу. Кожний екземпляр працює з даними його власної декларації глобальних змінних.

Щоб зробити видимим дерево екземпляра, необхідно клацнути по етикетці Instances (Екземпляри) в меню, що розташоване внизу вікна проектного дерева.

### Завдання до лабораторної роботи

1. Зрозуміти методику створення нового проекту і засвоїти можливості редактора проектного дерева.

2. Створити новий проект, в якому оголосити власну бібліотеку, у проектному дереві додати робочі листки типу даних і ROUs, захистити доступ до бібліотеки. Відкрити дерево екземпляра.

### **Контрольні запитання**

1. З яких етапів складається створення нового проекту?
2. Як оголосити власну бібліотеку?
3. Які типи даних розташовуються у теці Data Types проектного дерева?
4. Що треба зробити, щоб вставити робочі листки типу даних у проектне дерево?
5. Що таке ROU і, з яких робочих листків він складається?
6. Як можна вставити ROU у проектне дерево?
7. Як у проектному дереві додати робочі листки типу даних і ROU?
8. Які елементи конфігурації містить проектне дерево?