

Лабораторна робота № 0

Знайомство з системою програмування KW MULTIPROG

MULTIPROG – це SoftLogic-система програмування промислових логічних контролерів, яка відповідає стандарту IEC 61131-3. Система базується на віконній технології, яка використовує графічні можливості MS - Windows. Тому принципи стандарту здебільшого реалізуються графічно з символами і іконами або діалогами, в яких можуть бути визначені властивості елементів.

За допомогою MULTIPROG можна програмувати на двох текстових ST та IL і трьох графічних – LD, FBD та SFC мовах стандарту IEC 61131-3. Здійснюється це в діалоговому режимі, використовуючи, так званого, Project Wizard (Майстр Проекту).

Графічний редактор системи дозволяє вільне розміщення об'єктів на робочому полі проекту, вибирати мову програмування та використовувати бібліотеку операторів, функцій і функціональних блоків. Він підтримує змішування графічних мов в одному робочому листку, вставлення нових елементів в існуючі мережі, переміщення об'єктів або мереж.

Текстовий редактор дозволяє легко і швидко створювати програми, завдяки кольоровому зображенню ключових слів, використанню бібліотечних блоків і елементів.

Система програмування має могутній засіб налагодження із зручним інтерфейсом користувача. За допомогою MULTIPROG перевірка програми контролера здійснюється на симуляторі без участі самого контролера.

MULTIPROG працює разом з програмованою операційною системою реального часу ProConOs (Programmable Controller Operating System), яка вбудована в контролер[6].

2.1. Проекти в системі програмування

Проект у MULTIPROG містить усі необхідні елементи системи автоматизації. Він складається з бібліотеки(Libraries), типу даних(Data Type), програмних модулів (Logical POUs) і набору елементів конфігурації, які зображені у піддереві Physical Hardware (Фізична апаратура). Дерево проекту приведено на рис.2.1.

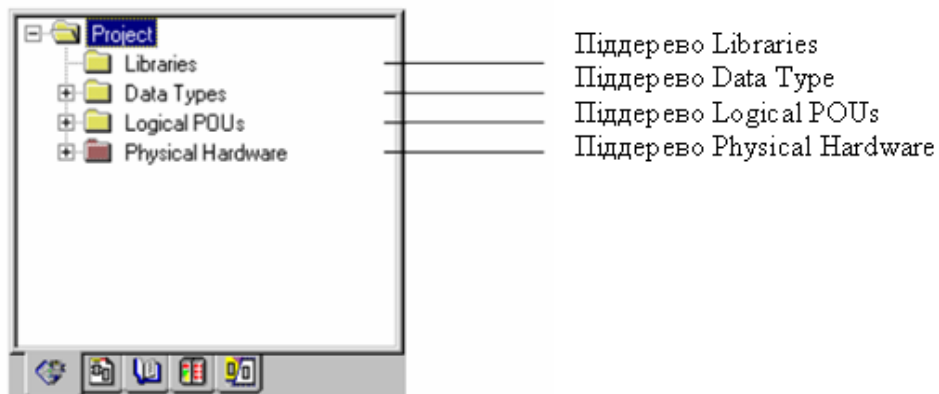


Рис.2.1. Дерево проекту

Редактор проектного дерева дозволяє створювати, вставляти, видаляти, копіювати та переміщувати об'єкти в проектному дереві.

Щоб створити функцію або функціональний блок користувача, необхідно:

- вставити новий ROU в проектне дерево, викликаючи правою клав'яшею миші у піддереві Logical ROUs (Логічні ROUs) діалог Insert (Вставка);
- відкрити робочий листок тіла коду новоствореного ROU;
- відредагувати тіло коду, вставляючи інструкції, оператори, контакти, катушки і т.і. (залежно від мови програмування), а також функції і функціональні блоки та змінні;
- створити робочий листок сітки змінних ROU;
- зберегти відредаговані робочі листки;
- компілювати редагований ROU.

Після компіляції нова функція або функціональний блок поміщаються в область вибору Edit Wizard (Майстер редагування) - група Project name (Ім'я проекту) і можуть вибиратися і вставлятися, як певна функція або функціональний блок, в інші ROUs поточного проекту.

Об'єкти можуть бути вставлені в проектне дерево за допомогою контекстного меню ікон проектного дерева. Контекстне меню кожного об'єкта проектного дерева містить підменю Insert (Вставка), яке показує об'єкти, що можуть бути вставлені в проектне дерево.

Наприклад, підменю Insert (Вставка) піддерева Logical ROUs (Логічні ROUs) містить пункти Program(Програма), Function Blok(Функціональний блок), Function(Функція), а контекстне підменю ікони робочого листка тіла коду дозволяє вставити тільки інший робочий листок тіла коду.

Щоб вставити об'єкт:

- позначте курсором у проектному дереві бажану ікону і клацніть по неї правою клав'яшею миші;
- відкриється підменю Insert (Вставка), виберіть об'єкт, який має бути вставлений. З'явиться відповідне діалогове вікно Insert (Вставка);
- після призначення імені та вибору мови програмування натисніть ОК і новий об'єкт з'явиться у проектному дереві.

З проектного дерева можна видалити програми, функціональні блоки, функції і всі робочі листки. Можна також видалити переходи і дії, що створені при програмуванні SFC-мовою. Перед видаленням ROU або окремого робочого листа необхідно закрити всі відкриті робочі листки ROU. Після цього позначте об'єкт або декілька об'єктів, які потрібно видалити. Виберіть пункт Delet (Видалення) в підменю Edit (Редагування) або натисніть клавішу "DEL".

Іноді швидше і легше копіювати існуючий об'єкт, ніж створювати новий.

Щоб це зробити:

- позначте лівою клав'яшею миші об'єкт або декілька об'єктів, що копіюються;
- в підменю Edit (Редагування) виберіть Copy (Копія) і клацніть лівою клав'яшею миші;
- позначений об'єкт скопіюється у буфер обміну і може бути встановлений в інше місце, використовуючи команду Paste (Вставити) у підменю Edit (Редагування).

Не можна вставити об'єкт у несумісні структури, програми. Наприклад, робочі листи FBD - мовою, в ROU ST – програми. Тому команда Paste (Вставити) активізується тільки після того, як буде визначене правильне місце для вставлення скопійованого об'єкта.

Для вставлення об'єкта:

- у проектному дереві позначте місце, куди бажано додати скопійований об'єкт, наприклад, для ROU це може бути піддерево Logical ROUs(Логічні ROU);
- в підменю Edit (Редагування) виберіть Paste (Вставити);
- копійований об'єкт буде вставлений з буфера обміну в позначене положення.

Щоб перемістити об'єкт в проектному дереві, необхідно виконати два кроки – спочатку вирізати об'єкт, а потім вставити його в точку призначення.

Для переміщення об'єкта:

- позначте об'єкт, який переміщується, якщо треба позначте декілька об'єктів;
- в підменю Edit (Редагування) виберіть Cut (Вирізати);
- позначений об'єкт копіюється в буфер обміну і може бути вставлений в інше положення, використовуючи команду Paste (Вставити).

Після вставлення об'єкта в нове положення, зі старого положення він буде видалений.

Система програмування може забезпечити захист проекту проти несанкціонованого вставлення або вилучення об'єктів (таких як ROUs і робочі листки) у проектному дереві. Залежно від установок безпеки, зроблених за допомогою Project owner (Власник проекту), доступ до проекту можна обмежити завдяки пароллю. Після цього у дерево об'єкта, в якому було визначено обмеження, додається ікона з символом ключа.

2.1.1. Бібліотеки

Бібліотеки (Libraries) – це ROUs , 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 ROUs 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.

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

Піддерево Libraries (Бібліотеки), рис.2.2, складається з двох або більше ікон. Перша ікона – це вузол директорії. Елемент цього вузла директорії являє оголошену бібліотеку.

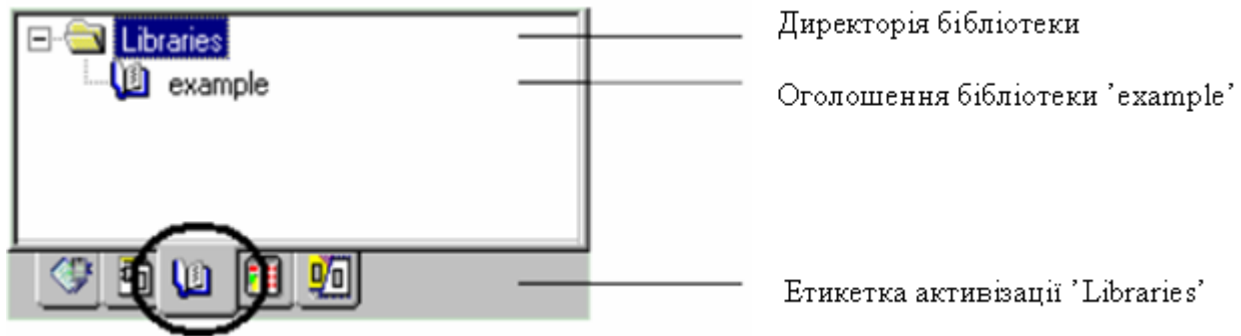


Рис. 2.2. Піддереву бібліотеки користувача

При створенні бібліотеки потрібно дотримуватися наступного:

- для оголошення бібліотек можуть використовуватися тільки ROUs;
- в оглядовому режимі робочі листки ROUs оголошених бібліотек користувача можуть бути тільки відкриті, тобто тільки показані й не можуть бути редагованими;
- одночасно робочі листки бібліотеки користувача, що відкриті в оглядовому режимі, можуть перемикатися в оперативний режим, в якому можливе їх налагодження;
- робочі листки бібліотеки мікропрограм не можуть бути відкритими ні в оглядовому режимі роботи, ні в оперативному;
- робочі листки бібліотеки не можуть бути надрукованими;
- право для оголошення бібліотек і огляду їх робочих листків може обмежуватися паролем користувача.

Щоб оголосити бібліотеку, позначте у проектному дереві теку піддерева Libraries (Бібліотеки).

- В панелі інструментів клацніть по іконі Add Object (Додати об'єкт). Альтернативно можна вибрати пункт меню Add Library (Додати бібліотеку) в підменю Project (Проект).

Примітка. Якщо 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).

- У діалоговому вікні Include library (Вставити бібліотеку), що з'явилося, виберіть бажану бібліотеку, тобто mwt-файл бібліотеки користувача, або fwl-файл бібліотеки мікропрограм.

- Виберіть у теці KWSOFT системи програмування MULTIPROG теку Project (Проект), а в неї - файл, який бажано оголосити як бібліотеку, і у діалоговому вікні Include library (Включити бібліотеку) натисніть кнопку Include. Ікона створеної бібліотеки вставиться у проектне дерево. Щоб переглянути бібліотеку, необхідно відкрити проектне дерево "Libraries", а в ньому - необхідні робочі листки.

Для видалення бібліотеки позначте її ікону в проектному дереві і натисніть клавішу "DEL", з'явиться попередження, в якому потрібно підтвердити видалення.

2.1.2. Типи даних

Типи даних у системі програмування – це, в основному, ті самі типи даних, які описані в стандарті IEC 61131-3. Певні типи даних, які призначені користувачем, редагуються в робочих листках типу даних, що знаходяться у піддереві Data Types (Типи даних), рис. 2.3, дерева проекту.



Рис. 2.3. Піддерево типів даних

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

Примітка. Якщо ікона інертна, захист з використанням пароля активізований. Тому треба ввести пароль використовуючи пункт меню Enter password (Ввести пароль) в підменю File (Файл).

З'явиться діалогове вікно Insert (Вставка). Введіть ім'я нового робочого листка типу даних і підтвердіть діалог. Новий робочий листок типу даних буде вставлено у відповідне піддерево.

Щоб відкрити робочий листок типу даних з текстовим редактором, в дереві проекту подвійно клацніть по іконі Data Types Worksheet (Робочий листок типу даних). Може статися, що виробник ПЛК визначив типи даних, які можуть використовуватися при редагуванні програми. Тому треба заздалегідь подивитися в документацію технічного засобу.

Піддерево Data Type (Тип даних) можна оглядати, показуючи повне проектне дерево або тільки піддерево Data Type (Тип даних) і Logical POUs (Логічні POUs). Для цього клацніть по етикетці POUs в меню проектного дерева, що розташоване внизу вікна.

2.1.3. POUs

Програми, функціональні блоки і функції можуть редагуватися у піддереві Logical POUs (Логічні POUs), рис. 2.4, проектного дерева.

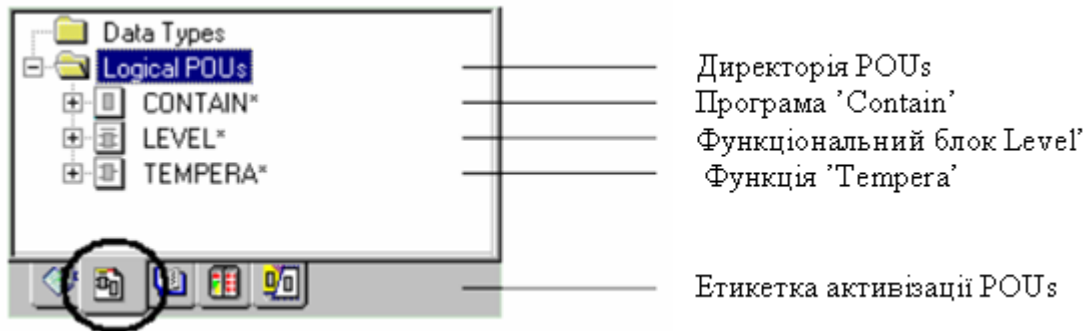


Рис. 2.4. Піддерево одиниць організації програми (POUs)

Кожне POU складається з декількох робочих листків, рис. 2.5:

- Робочий листок опису створюється для документації і містить текст, введений користувачем. В ньому описується POU або елементи конфігурації, проте він не є обов'язковим;
- Робочий листок сітки змінних містить декларацію локальних змінних;
- Робочий листок тіла коду містить фактичний код, що редагувався однією з найзручніших мов програмування.

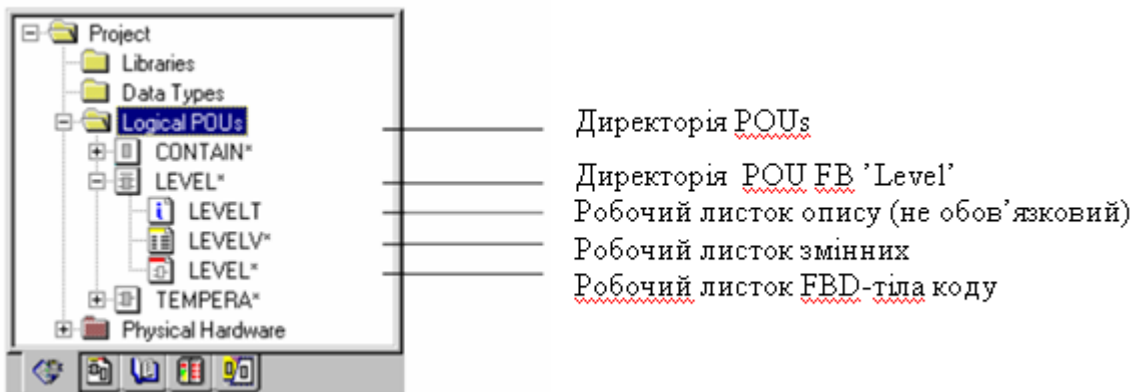


Рис. 2.5. Структура директорії POUs

Ці робочі листки у піддереві Logical POUs (Логічні POUs) зображаються 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 сітки 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, позначте теку Logical POUs (Логічні POUs) і в панелі інструментів клацніть 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(Проект > Додати POU) або клацнути правою клавiшею миші;

- з'явиться діалог Insert (Вставка);

- введіть ім'я нового POU;

- виберіть бажану мову, активізуючи відповідний перемикач в зоні Language (Мова);

- якщо потрібно введіть PLC і/або тип процесора;

- позначте перемикач Use Reserve (Використання резерву), якщо потрібно.

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 натисненням кнопки Data Area (Область даних) у діалозі Resource setting (Установки ресурсу) вашого ПЛК.

- Підтвердіть діалог.

Новий POU є вставлений у проектне дерево. Він містить робочий листок тіла коду, робочий листок змінних і робочий листок опису.

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).

Нові робочі листки в проектному дереві позначені зірочкою.

Зірки означають, що ці робочі листки можуть бути вставлені або замінені, але ще не компільовані.

Система забезпечує захист структури піддерева Logical POUs (Логічний POUs) проти несанкціонованої вставки або видалення POUs і робочих листків, використовуючи пароль.

В POU можна вставити робочі листки коду програми, проте тільки того самого типу. Наприклад, можна вставити в IL- POU тільки IL-робочі листки.

Щоб вставити новий робочий листок:

- відкрийте бажаний POU;

- позначте один з робочих листків POU;

- в панелі інструментів клацніть по іконі Add Object (Додати об'єкт), з'явиться діалогове вікно Insert (Вставка).

Примітка. Якщо ікона інертна, в проекті використовується захист паролем. Треба ввести пароль. Для цього виберіть пункт меню Enter password (Ввести пароль) у підменю File (Файл).

- У діалоговому вікні введіть ім'я нового робочого листка;

- визначтесь, новий робочий листок буде вставлений після (перемикач Append (Приєднати)) чи до (перемикач Insert (Вставка)) позначеного робочого листка;

- підтвердіть діалог Insert (Вставка).

Новий робочий листок є вставленим у проектне дерево.

Нові робочі листки позначені у проектному дереві зірочками, які означають, що робочі листки встановлені або змінені, але не компільовані.

Можна переглянути у проектному дереві властивості усіх існуючих ROUs і робочих листків. Деякі властивості можуть бути змінені.

Мова програмування існуючого ROUs або робочого листка не може бути змінена оскільки вона визначається при створенні ROU.

Для перегляду або зміни властивостей:

- клацніть правою клавішею миші по бажаній іконі теки ROU або іконі робочого листка у проектному дереві. З'явиться контекстне меню ікони;
- виберіть пункт меню Properties (Властивості) для відкриття діалогу Properties (Властивості);
- змініть властивості, якщо бажаєте.

Перейменування ROU або робочого листка не можливо, коли робочий лист відкритий.

2.1.4. Елементи конфігурації у системі програмування

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

Взагалі можуть використовуватися одна або декілька конфігурацій. У кожній конфігурації можуть бути оголошені один або декілька ресурсів. У межах одного ресурсу можуть використовуватися декілька задач, зв'язаних з їх програмами. Задачі визначають планування часу програм.

У ресурсі можуть бути оголошені глобальні змінні, які дійсні тільки в межах цього ресурсу.

Протягом створення робочого коду програм и функціональних блоків у піддереві Physical Hardware (Фізична апаратура), рис. 2.6, додаються їх екземпляри, а також редагуються окремі робочі листки глобальних змінних. Як результат кожний екземпляр програми або функціонального блока має власний набір даних з безпосереднім доступом.

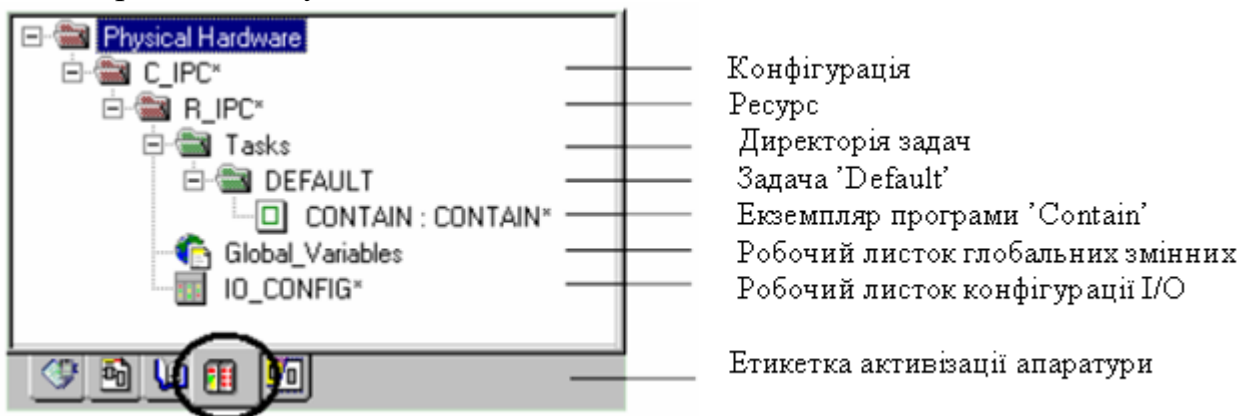


Рис. 2.6. Структура піддерева Physical Hardware

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

У піддереві Physical Hardware (Фізична апаратура) можна створити тільки екземпляр функціонального блока в програмі і не можна створити екземпляр функціонального блока в межах функціонального блоку.

Ім'я екземпляра, яке використовується у піддереві Physical Hardware (Фізична апаратура) має відповідати імені програми, яке використовується у піддереві Logical ROUs (Логічні ROUs).

Приклад екземпляра програми та функціонального блока з відповідним робочим листком глобальних змінних приведено на рис. 2.7:

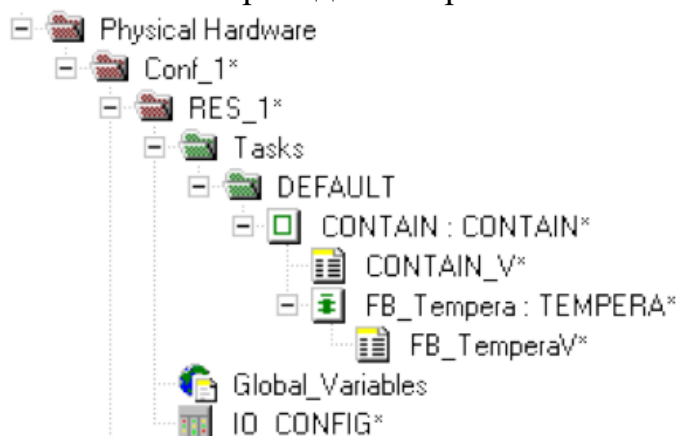


Рис. 2.7. Екземпляр програми і функціонального блока

Дерево екземпляра показує доступні ресурси, задачі і зв'язані з ними програми з усіма функціями і функціональними блоками ресурсу. Структура дерева створюється протягом компіляції ресурсу. Тому дерево екземпляра не може редагуватися.

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

Для оголошення екземплярів програми або функціонального блока, доведеться використовувати додаткову змінну – ключові слова.

Ці додаткові ключові слова можуть бути вибрані зі стовпчика Usage (Вжити) робочого листка сітки змінних, коли оголошуються екземпляри.

Завдання до лабораторної роботи

1. Зрозуміти можливості системи програмування KW MULTIPROG.
2. Навчитися працювати з системою програмування KW MULTIPROG.

Контрольні запитання

1. Що собою являє система програмування MULTIPROG?
2. Який вигляд має проект, створений у MULTIPROG?
3. Що таке бібліотека у системі програмування MULTIPROG?
4. Як створити бібліотеку?
5. Як зображаються типи даних у системі програмування MULTIPROG?
6. Який вигляд має директорія ROUs і що вона містить?
7. Як система програмування відображає структуру елементів конфігурації?
8. Як вставити проект у проектне дерево?
9. Як створити функцію або функціональний блок?
10. Як у проектному дереві об'єкти видаляються, копіюються та переміщуються?
11. Як вставити ROU у проектне дерево?
12. Для чого створюються екземпляри програм і функціональних блоків?