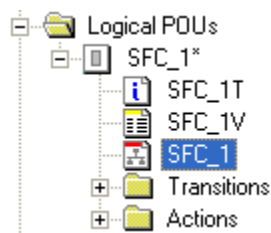


Лабораторна робота №6 Створення програмного коду SFC - мовою

Знайомство з редактором SFC – мови

Одною з головних особливостей редагування SFC – мовою є те, що система програмування завжди прагне створити закриту SFC-мережу. Коли вставляються нові кроки і переходи, частина мережі під точкою вставлення автоматично зміщується вертикально вниз. Проте структуру SFC – мережі необхідно організовувати так, щоб був і достатній простір для її розширення убік.

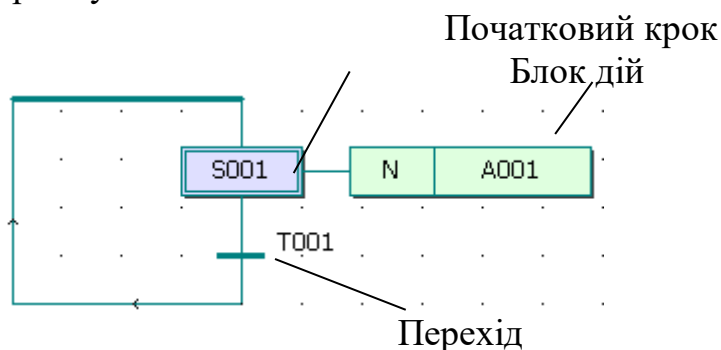
Для редагування робочого листка SFC- коду, необхідно відкрити його подвійним клацанням лівої клавіші миші по відповідній іконі теки Logical POU's у дереві проекту:



Протягом редагування робочого листка створюється SFC- мережа, яка складається з кроків і переходів, редагованих LD-,FBD-,IL- або ST-мовою. Робочі листки тіла коду кроків і переходів розташовуються у відповідній піддиректорії проектного дерева і називаються деталлю.

Для створення SFC- мережі у робочому листку тіла коду курсором позначте місце вставки;

– в панелі інструментів клацніть по іконі Create step transition sequence (Вставити послідовність крок-перехід) і на екрані з'явиться SFC- мережа з одним кроком і одним переходом. Перший вставлений крок автоматично є початковим кроком, він має подвійну рамку.



Щоб розширити існуючу мережу за рахунок нових кроків і переходів:

– клацніть лівою клавішею миші по бажаному кроці, щоб позначити місце розширення мережі;

– в панелі інструментів клацніть по іконі Create step transition sequence (Вставити послідовність крок - перехід) і інша пара (крок-перехід) з'явиться між позначеним кроком і наступним переходом.

Якщо позначити перехід замість кроку і вставити нову послідовність (крок – перехід), то вона вставиться нижче позначеного переходу.

Ширина кроків і дій SFC-мережі завжди однакова, якщо у підменю Layout (Розташування) альтернатива Fixed SFC objects Width (Постійна ширина SFC об'єкта)- активізована. Якщо ні, ширина їх коригується автоматично відповідно до довжини імені кроку або дії.

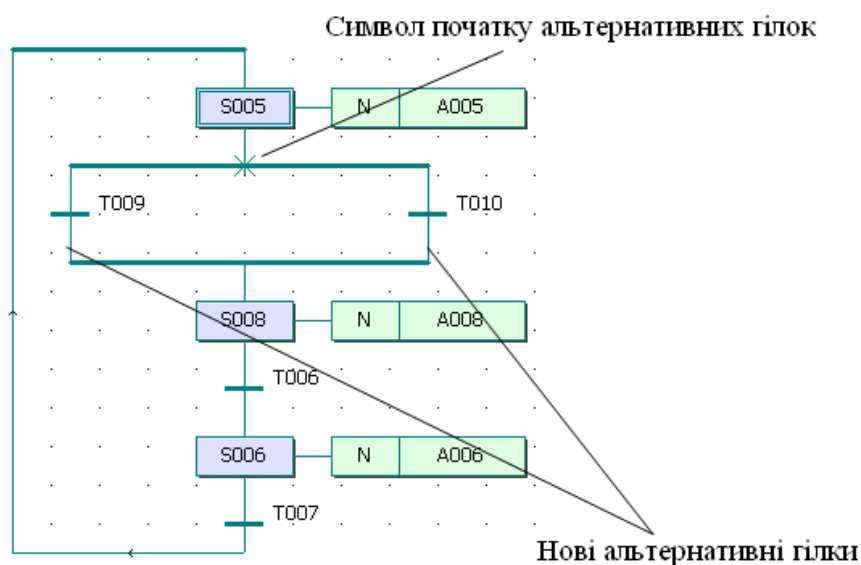
При необхідності можна змінити початковий крок на звичайний, а звичайний – на початковий. Для цього:

- двічі клацнути лівою клавішею миші по кроці, який бажано змінити. З'явиться діалог Step (Крок);
- активізувати перемикач Initial step (Початковий крок), щоб змінити звичайний крок на початковий;
- вимкнути перемикач Initial step (Початковий крок), щоб змінити початковий крок на звичайний. Звичайний крок зображується в робочому листку прямокутником з одинарною рамкою.

Протягом редагування можна вставити у SFC–мережу альтернативні або паралельні гілки.

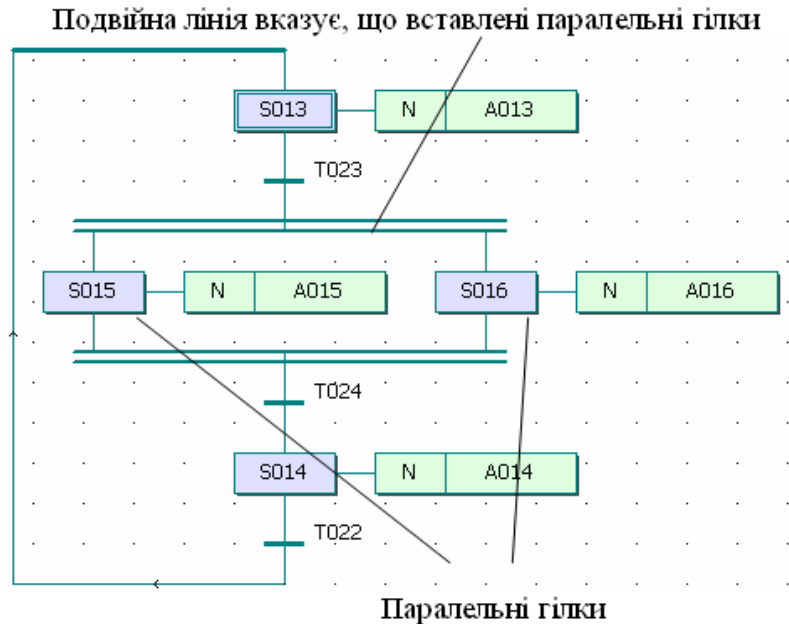
Щоб вставити альтернативні гілки:

- позначте крок, де бажано вставити гілки;
- в панелі інструментів клацніть по іконі Insert Simultaneous/Alternativ Branches (Вставка Одночасної/Альтернативної гілки). З'явиться діалог Divergence (Розходження);
- введіть бажану кількість гілок у вікні діалогу Branches Count (Рахунок гілок) і натисніть ОК:



Щоб вставити паралельні гілки:

- позначте перехід, де бажано вставити гілки;
- клацніть по іконі Insert Simultaneous/Alternativ Branches (Вставка Одночасної/Альтернативної гілки);
- введіть необхідну кількість гілок у вікні діалогу Branches Count (Рахунок гілок) і натисніть ОК:



Вставка гілок можна здійснювати, використовуючи режим редагування SFC – гілок.

Щоб вставити паралельні гілки:

- в панелі інструментів клацніть лівою клав'яшею миші по іконі Insert SFC branch (Вставка SFC – гілки). З'явиться курсор у вигляді SFC – гілки;
- спрямуйте курсор на перехід, після якого має бути розходження гілок і клацніть лівою клав'яшею миші. З'явиться подвійна лінія. Перемістіть курсор вбік на вільне місце, де бажаєте встановити новий крок з блоком дії;
- клацніть лівою клав'яшею миші, щоб зафіксувати верхню подвійну лінію;
- перемістіть курсор на перехід, де бажано встановити нижню подвійну лінію, і клацніть лівою клав'яшею миші.

Режим редагування SFC – гілок завершується автоматично після того, як нова гілка вставлена. Для вставлення іншої гілки повторно виберіть режим редагування SFC - гілок, знову клацаючи по іконі Insert SFC branch (Вставка SFC – гілки).

Щоб вставити альтернативну гілку, треба повторити такі самі дії, як і у попередньому варіанті, тільки спочатку спрямуйте курсор у вигляді SFC – гілки не на перехід, а на крок.

В існуючих SFC – мережах можна вставити стрибки. Стрибки вказують шлях виконання програми у розімкненій SFC-мережі. Стрибок має ім'я мети, куди він прямує. Щоб вставити стрибок:

- двічі клацніть лівою клав'яшею миші по кроці, з якого має бути стрибок. З'явиться діалог Step (Крок);
- у рядку Name(Ім'я) введіть ім'я кроку, на який здійснюється стрибок;
- активізуйте перемикач Jump (Стрибок) і натисніть ОК;
- з'явиться діалог Jump/End step insert control (Стрибок/Кінець управління кроком вставки);
- активізуйте перемикач Show marked objects for delete (Показ позначених об'єктів для видалення) і натисніть ОК, щоб побачити усі об'єкти SFC-мережі, що видаляються при вставленні стрибка;

- знову викличте діалог Jump/End step insert control (Стрибок/Кінець управління кроком вставки);
- натисніть кнопку Change (Заміна) для видалення усіх непотрібних об'єктів і вставлення стрибка з іменем мети.

Можна вставити кінцеві кроки, якщо бажаєте закінчити SFC - мережу.

Для цього:

- клацніть по кроці, який бажаєте зробити кінцевим. З'явиться діалог Step (Крок);
- активізуйте перемикач End step (Кінцевий крок) і натисніть ОК;
- з'явиться діалог Jump/End step insert control (Стрибок/Кінець управління кроком вставки);
- активізуйте перемикач Show marked objects for delete (Показ позначених об'єктів для видалення) і натисніть ОК, щоб побачити усі об'єкти SFC-мережі, що видаляються при вставленні стрибка;
- знову викличте діалог Jump/End step insert control (Стрибок/Кінець управління кроком вставки);
- натисніть кнопку Change (Заміна) для видалення усіх непотрібних об'єктів і вставлення кінцевого кроку.

Можна вставити блок дій до кроку, якщо він був видалений. Можна додати нові блоки дій до одного існуючого і створити, таким чином, у робочому листку тіла коду крок із складеним блоком дій. Щоб видалити існуючий блок дій, клацніть по ньому лівою клавішею миші і натисніть «Delete» на клавіатурі комп'ютера.

Щоб вставити новий блок дії:

- позначте крок, до якого бажаєте вставити новий блок дії або позначте існуючий блок дії, якщо бажаєте мати складений блок дій;
- в панелі інструментів клацніть по іконі Create action (Створити дію). З'явиться діалогове вікно Action Properties (Властивості дії);
- введіть нове ім'я дії, якщо бажаєте, і натисніть ОК. Новий блок дії з'явиться у робочому листку тіла коду.

Робочий листок тіла коду дії, так звану деталь, можна присвоїти будь-якому кроку існуючої SFC – мережі.

Для цього:

- позначте крок, якому бажаєте присвоїти нову деталь;
- в панелі інструментів клацніть по іконі Create action (Створити дію);
- з'явиться діалог Action Properties (Властивості дії);
- у першому рядку поля Name (Ім'я) введіть ім'я нового робочого листка тіла коду (деталі);
- активізуйте кнопку Detail (Деталь) і натисніть ОК. Позначений крок отримує нову дію з власним іменем.

Дії можуть мати логічну змінну або тіло коду.

Протягом редагування проекту можна змінити властивості кроку і блоку дії, а також переходу.

Для зміни властивостей кроку:

– двічі клацніть по кроку лівою клавішею миші. З'явиться діалогове вікно Step(Крок);

– змініть ім'я або тип кроку і натисніть ОК.

Щоб змінити властивості блока дії:

– правою клавішею миші клацніть по блоку дії і виберіть пункт контекстного меню Objects Properties (Властивості об'єкту). З'явиться діалогове вікно Action Properties (Властивості дії);

– зробіть зміну імені або класифікатора і натисніть ОК.

Для зміни властивостей переходу:

– клацніть правою клавішею миші по переходу, щоб викликати контекстне меню, і виберіть пункт Objects Properties (Властивості об'єкта). З'явиться діалог Transition (Перехід);

– змініть ім'я або тип переходу і натисніть ОК.

Переходи можуть мати деталі або прямі зв'язки. Коли перехід має власне тіло коду, тобто деталь, у діалозі Transition (Перехід) треба активізувати тип Detail (Деталь).

Прямий зв'язок означає, що у переходу немає тіла коду, і він безпосередньо з'єднується з LD – або FBD – мережею, або має змінну. У цьому випадку в діалозі Transition (Перехід) потрібно активізувати тип Direct connection (Прямий зв'язок). У разі прямого зв'язку перехід не має імені.

Для створення деталі дії або переходу в робочому листку SFC-коду подвійно клацніть по дії або переходу, з'явиться діалогове вікно Insert (Вставити);

– виберіть мову програмування деталі;

– натисніть ОК і робочий листок дії або переходу відкриється для редагування і одночасно з'явиться у проектному дереві.

Дії в робочих листках тіла коду можуть бути зображені логічними змінними.

Щоб присвоїти дії ім'я змінної або замінити на ім'я змінної, яка вже декларована:

– правою клавішею миші клацніть по дії, щоб відкрити контекстне меню;

– виберіть пункт Object Properties (Властивості об'єкту). З'явиться діалог Action Properties (Властивості дії);

– у діалоговій сторінці Action (Дія) активізуйте кнопку Variable (Змінна);

– зі списку поля Name (Ім'я), що містить усі оголошені змінні, виберіть бажану і клацніть по ній лівою клавішею миші;

– у рядку Local Variable Groups (Група локальній змінній) активізуйте Default (За замовчуванням);

– уатисніть ОК і змінна буде вставлена у блок дії.

Щоб декларувати нову змінну дії:

– клацніть по дії правою клавішею миші, щоб відкрити контекстне меню;

– виберіть Object Properties (Властивості об'єкту) і клацніть лівою клавішею миші. З'явиться діалог Action Properties (Властивості дії);

– у діалоговій сторінці Action (Дія) активізуйте перемикач Variable (Змінна);

- у першому рядку поля Name (Ім'я) надрукуйте ім'я дії, виберіть відповідні установки для змінної.

Після вибору компетенції клацніть ОК, змінна вставиться у робочий листок тіла коду. Автоматично її декларація потрапляє у робочий листок відповідної сітки змінних. У разі глобальної змінної декларація вставляється у сітку локальних змінних, використовуючи ключове слово VAR_EXTERNAL, а у робочий листок сітки глобальних змінних, використовуючи VAR_GLOBAL.

У разі декларування змінних переходів можна вставити змінну вже приєднаною до переходу або з'єднати з переходом змінну, яка встановлена у будь-якому місці робочого листка тіла коду.

Щоб присвоїти переходу ім'я вже декларованої змінної або надати йому нове ім'я, або замінити існуюче ім'я переходу:

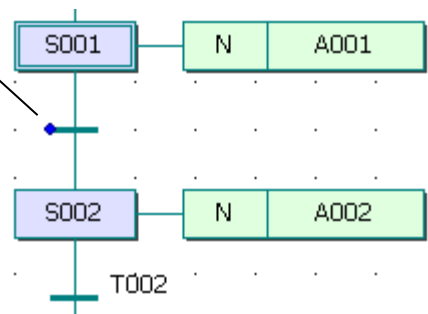
- клацніть по переходу правою клавішею миші, щоб відкрити контекстне меню;

- виберіть пункт Object Properties (Властивості об'єкта). З'явиться діалогове вікно Transition (Перехід);

- активізуйте перемикач Direct connection (Прямий зв'язок) і натисніть ОК.

Перехід показується синьою точкою з'єднання.

Синя точка



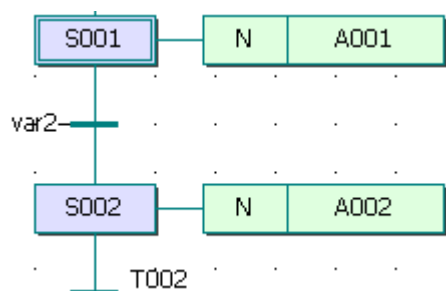
- Позначте перехід, лівою клавішею миші;

- в панелі інструментів клацніть по іконі Variable (Зміна), з'явиться діалогове вікно Variable Properties (Властивості змінної);

- у вікні вже декларованих змінних поля Name (Ім'я) клацніть по бажаному імені, і воно увійде у перший рядок. Якщо немає - надрукуйте нове ім'я.

Коли переходу надаєте нове ім'я у діалоговому вікні Variable Properties (Властивості змінної) виберіть відповідні установки і визначтесь з компетенцією змінної.

Закрийте діалогове вікно і змінна потрапить у робочий листок тіла коду, а її декларація автоматично потрапить у відповідний робочий листок сітки змінних:



Замість змінних до переходу можуть бути приєднані функції або LD- мережі за допомогою прямих зв'язків. У цих випадках спочатку необхідно в робоче поле тіла коду вставити функцію чи LD- мережу, а потім з'єднати її з точкою зв'язку переходу, активізуючи в панелі інструментів ікону Connect Objects (З'єднати об'єкти).

При видаленні фрагментів SFC-коду, в які входять більш ніж один крок і один перехід, або цілі гілки, на робочому полі тіла коду іноді залишаються дуже довгі лінії сполучення. У цих випадках довжину ліній можна оптимізувати автоматично. Для цього:

- позначте за допомогою лівої клавіші миші частину лінії, яку потрібно оптимізувати;
- у підменю Edit>Stretch/Compress (Редагування>Розтягання/ Стиснення) виберіть пункт Optimize SFC Lines (Оптимізація SFC –лінії) і довжина ліній оптимізується автоматично.

Створення проекту користувача

Створити SFC-мовою програму керування роботою двигуна згідно з алгоритмом наведеним у лабораторній роботі №2.

Для цього відкрийте діалог New Project (Новий проект), використовуючи пункт меню File (Файл). Двічі клацніть по Project Wizard (Майстер проекту) і пройдіть увесь шлях створення проекту аналогічно варіанту програмування LD-мовою, тобто призначте ім'я проекту та його POU, виберіть у даному випадку SFC-мову програмування, бажані імена і типи конфігурації, ресурсу і задачі.

Будь-який проектний код, що створено SFC – мовою, складається з кроків і переходів, об'єднаних SFC – схемою. Тому перед тим, як їх програмувати необхідно побудувати структуру цієї схеми.

Створення SFC – схеми керування роботою двигуна:

Спочатку побудуємо послідовну мережу кроків та переходів, які забезпечують ініціалізацію змінних, програмування умов вмикання в роботу двигуна, програмування перемикача для створення можливості непередбаченого вимкнення двигуна та лічильника кількості активізацій двигуна.

На робочому полі проектного коду позначте курсором місце початку програмування;

- в панелі інструментів клацніть лівою клавішею миші по іконі Creat step transition sequence(Створити послідовні крок і перехід). На робочому полі з'явиться початкова SFC-схема з одного кроку S001 і дії A001, одного переходу T001, а також повернення назад.

Якщо з'явився не початковий крок, а звичайний, тобто прямокутник кроку створений не подвійною лінією, а одинарною, то перетворить його у початковий:

Для призначення імені компонентам SFC-схеми :

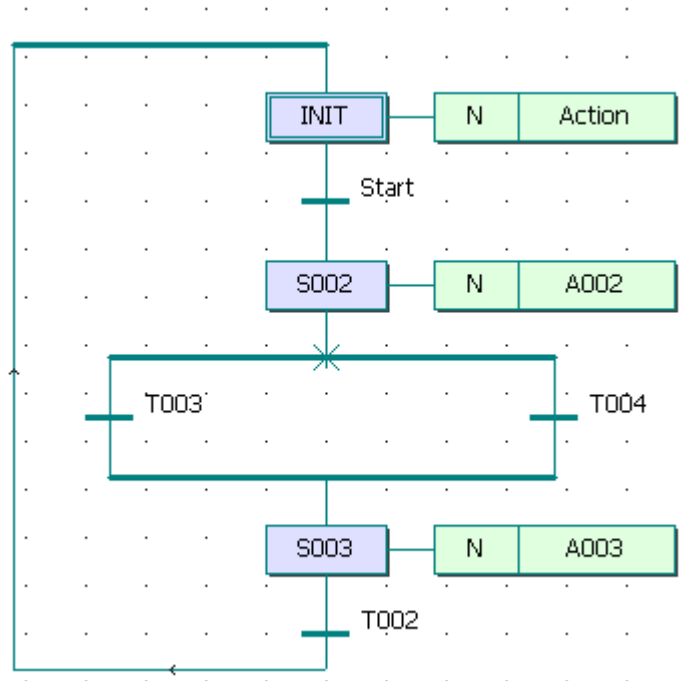
- клацніть правою клавішею миші по кроку S001 і в меню, що з'явиться, лівою клавішею миші активізуйте Object Properties... У вікні Step у рядку Name надрукуйте ім'я INIT і клацніть ОК;

- таке саме зробіть послідовно з дією A001 та переходом T001, привласнюючи їм імена відповідно Action з класифікатором (Qualifier) N замість A001 у діалоговому вікні Action Properties (Властивості дії) та Start замість T001 у діалоговому вікні Transition (Перехід), при цьому в обох вікнах альтернатива Detail (Деталь) має бути активізована.

Для створення другої пари (крок і перехід) клацніть лівою клав'яшею миші спочатку по переходу Start, а потім в панелі інструментів по іконі Creat step transition sequence(Створити послідовні крок і перехід). В наслідок цього існуюча SFC-схема збільшиться ще на один крок S002 і перехід T002. При цьому присвоювати імена їм поки не будемо, оскільки далі на базі другої пари кроку і переходу створимо у SFC-схемі розгалуження.

Для створення альтернативних гілок розгалуження позначте курсором другий крок S002 SFC-схеми і в панелі інструментів клацніть по іконі Insert Simultaneous/Alternative Divergence (Вставити Паралельне/Альтернативне розгалуження);

- у діалоговому вікні Divergence (Розгалуження), що з'явиться, у полі Branches Count (Кількість гілок) надрукуйте 2 і клацніть ОК. Проектний код у робочому полі набуває вигляд:



У лівій та правій гілках альтернативного розгалуження створимо стрибки на початковий крок INIT. Для цього:

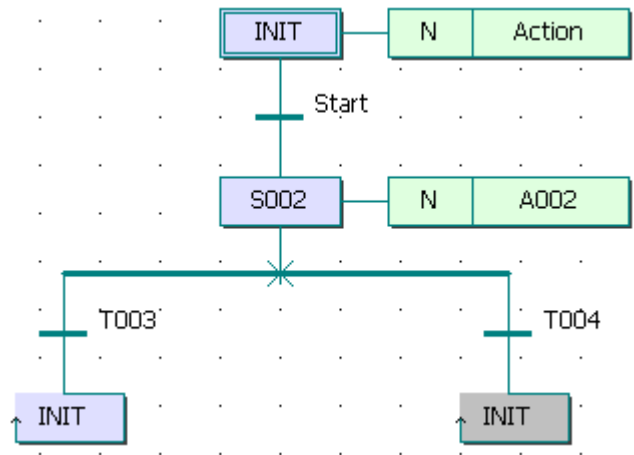
- позначте курсором перехід T003 і в панелі інструментів клацніть по іконі Creat step transition sequence (Створити послідовні крок і перехід), у лівій гілці з'являться крок S004 з дією A004 і перехід T005;

- правою клав'яшею миші клацніть по кроку S004 і в меню, що з'явиться, лівою клав'яшею миші виберіть Object Properties... У полі Type активізуйте альтернативу Jump(Стрибок) і клацніть ОК;

- у діалоговому вікні Jump/End step insert control (Стрибок/Кінець управління кроком вставки), що з'явиться, натисніть Change(Змінити) і в лівій гілці залишаться тільки перехід T003 і безадресний стрибок S004 у вигляді прямокутника зі стрілкою;

- щоб призначити адресу стрибка S004 лівою клавішею миші подвійно клацніть по ньому і у діалоговому вікні Step(Крок), що з'явиться, вкажіть у полі Name ім'я адреси – початковий крок INIT. Натисніть ОК і у прямокутнику стрибка замість S004 з'явиться INIT;

- такі самі перетворення, тільки на базі переходу T004, зробіть і з правою гілкою SFC-схеми. Після цього SFC-схема набуває вигляд:



- призначте новим компонентам SFC-схеми, що з'явилися на робочому полі (крок S002, дія A002 і переходи T003 і T004), імена відповідно Step1, Action1, Trans і Trans1, користуючись вже описаними раніше правилами.

Програмування кроків і переходів:

Після створення SFC-схеми необхідно запрограмувати дії кроків і умови переходів відповідно до алгоритму керування роботою двигуна.

Згідно з принципами SFC – мови, перший крок є ініціалізаційним. Тому змінній керування роботою двигуна Motor необхідно присвоїти значення FALSE, а лічильник скинути. Для цього:

- лівою клавішею миші двічі клацніть по дії початкового кроку Action, у діалоговому вікні Insert (Вставлення), що з'явиться, виберіть ST-мову програмування і натисніть ОК. Відкриється робоче поле дії Action;

- користуючись синтаксисом ST-мови присвойте змінній Motor, що керує роботою двигуна, значення FALSE, тобто надрукуйте: Motor :=FALSE;

- перейдіть на новий рядок і в панелі інструментів активізуйте ікону Edit Wizard (Майстер проекту), у таблиці Group, що з'явиться на екрані, виберіть меню Keywords;

- подвійно клацніть по оператору IF і на робочому листку проекту з'явиться його синтаксис з відповідним коментарем;

- установіть курсор після оператора IF і надрукуйте вираз Cycle_Count>10;

- установіть курсор після оператора THEN і надрукуйте вираз Cycle_Count:=Cycle_Count+1, який рахує кількість активізацій двигуна.

Створений фрагмент програми дії Action забезпечує обнуління лічильника після 10 активізацій двигуна.

Після цього ST-програма дії Action набуває завершений вигляд:

```

motor:=FALSE;
IF Cycle_Count>10    (*EXPRESSION (must return a boolean value)*)
    THEN Cycle_Count:=0(*If returned value of EXPRESSION = TRUE*
    (*STATEMENT*);
END_IF;

```

Щоб оголосити ці змінні, установіть курсор спочатку на ім'я Motor, а потім на Cycle_Count і в панелі інструментів лівою клав'яшею миші клацніть по іконі Variable (Змінна);

- у діалогових вікнах Variables (Змінні), що з'являться, заповнити необхідні рядки, для кожної локальної дискретної змінної, з урахуванням фізичної адреси змінної Motor – %QX0.0;

Програмування переходу Start:

Умовою початку роботи двигуна є триразове натискання стартової кнопки. Запрограмуємо її LD-мовою, використовуючи прямий зв'язок, коли перехід не має тіла коду, а безпосередньо сполучається з LD-мережею. У даному випадку перехід не повинен мати ім'я.

Щоб реалізувати цей намір:

- клацніть правою клав'яшею миші по переходу Start, у діалоговому вікні Transition (Перехід), що з'явиться, активізуйте альтернативу Direct Connection (Безпосереднє сполучення), рядок Name (Ім'я) зробіть порожнім і натисніть ОК;

- діалогове вікно закриється, а у робочому полі SFC - програми після початкового кроку INIT залишиться безіменна смужка переходу з блакитною точкою;

- для створення умови переходу LD-мовою клацніть лівою клав'яшею миші у зручному місці робочого поля зліва від безіменного переходу;

- в панелі інструментів клацніть по іконі Contact network (Мережа контактів) і в установленому місці робочого поля з'явиться LD- мережа 001 певної ширини з контактом і котушкою;

- вилучите з LD- мережі праву шину живлення і котушку C001 послідовно помічаючи їх лівою клав'яшею миші і натискаючи клавішу Delet на клавіатурі ЕОМ;

- в панелі інструментів активізуйте ікону Connect objects (З'єднати об'єкти) і за допомогою курсору з'єднайте прямий контакт C000 LD- мережі з блакитною точкою смужки безіменного переходу;

- лівою клав'яшею миші позначте лінію, що сполучає прямий контакт і перехід;

- в панелі інструментів клацніть по іконі Edit Wizard (Майстер редагування) і з таблиці Group (Група), що з'явиться на екрані, виберіть функціональний блок лічильника STU і присвойте йому ім'я Motor-Count;

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

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

- в панелі інструментів активізуйте 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 C000;

- в панелі інструментів клацніть по іконі Mark (Позначка), щоб деактивізувати кнопку Connect objects (З'єднати об'єкт).

Оголошення властивостей прямих контактів LD- мережі:

- двічі клацніть лівою клав'яшею миші по контакту C000 і оголошіть змінну Motor_Start з фізичною адресою %IX0.0;

- двічі клацніть по контакту C002 і у діалоговому вікні Contact/Coil Properties (Властивості Контакт/котушка), що з'явиться, з існуючого списку вже оголошених змінних виберіть Motor і клацніть ОК. На робочому листку з'явиться змінна Motor замість C002;

- для PV-входу введіть константу INT#3, щоб двигун (Motor) спрацював після триразового натиснення стартової кнопки;

- для CV-виходу лічильника, задекларуйте змінну Pressed, яка запам'ятовує поточне значення лічильника.

Програмування дії Actin1 кроку Step1:

Після триразового натискання кнопки Motor_Start має працювати двигун. Тому у наступному кроці Step1 запрограмуємо роботу двигуна, передбачивши можливість його екстреного зупинення. Для цього:

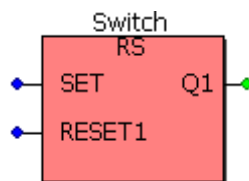
- лівою клав'яшею миші двічі клацніть по дії першого кроку Action1, у діалоговому вікні Insert (Вставлення), що з'явиться, виберіть FBD-мову програмування і натисніть ОК;

- у робочому полі дії Action1, що з'явиться, лівою клав'яшею миші позначте місце створення програмного коду;

- в панелі інструментів активізуйте ікону Edit Wizard (Майстер проекту) і у таблиці Group, що з'явиться на екрані, виберіть функціональний блок перемикача з домінантою вимкнення RS;

- двічі клацніть по ньому і у діалоговому вікні Variable properties (Властивості змінної) змініть типове ім'я RS -1 на Switch;

- клацніть ОК, діалогове вікно закриється, а у робочому листку дії Action1 залишиться функціональний блок перемикача Switch створений FBD-мовою:



Оголошення змінних блока Switch:

- двічі клацніть по SET- входу перемикача і у діалоговому вікні Variable properties (Властивості змінної), що з'явиться, у полі Name надрукуйте ім'я виходу лічильника кількості натисків стартової кнопки - Motor_Count.Q;

- натисніть ОК, діалогове вікно закриється, а SET- входу перемикача присвоюється змінна Motor_Count.Q ;

- двічі клацніть по RESET1- входу перемикача і у діалоговому вікні Variable properties (Властивості змінної), що з'явиться, у полі Name надрукуйте ім'я нової змінної Emergency_Stop(Непередбачене зупинення), у рядку Usage (Вживання) виберіть VAR, а у рядку Data Type (Тип даних) - BOOL;

- у рядку I/O address (Адреса вводу/виводу) надрукуйте фізичну адресу вхідного каналу модуля вводу %IX0.1;
- у рядку Local Variable Groups (Група локальній змінній) активізуйте Default (За замовчуванням);
- натисніть ОК, діалогове вікно закриється, а RESET1- входу перемикача Switch присвоїться змінна Emergency_Stop (Непередбачене зупинення);
- двічі клацніть по Q1- виходу перемикача і у діалоговому вікні Variable properties (Властивості змінної), що з'явиться, у полі Name надрукуйте ім'я вже оголошеної змінної Motor;
- натисніть ОК, діалогове вікно закриється, а Q1- виходу перемикача присвоюється змінна Motor;
- закрийте робоче поле дії Action1 кроку Step1, а його вміст запам'ятайте, натиснувши відповідну кнопку у діалоговому вікні MULTIPROG.

Разом з початком роботи двигуна має працювати лічильник його активізацій. Тому для першого кроку створимо ще одну дію, яка має виконуватися разом з дією Action1. Для цього:

- на SFC-схемі лівою клавішею миші клацніть по прямокутнику дії Action1, а в панелі інструментів по іконі Create action (Створення дії). Поряд з прямокутником дії Action1 з'явиться прямокутник іншої дії, якому присвойте ім'я Counter, керуючись вже відомою схемою.

Програмування дії Counter кроку Step1:

Створимо лічильник, який рахує кожне вмикання двигуна. При цьому після 10 циклів роботи показання лічильника скидаються у нуль.

- Лівою клавішею миші двічі клацніть по дії Counter;
- у діалоговому вікні, що з'явиться, виберіть ST-мову, закрийте вікно і на екрані з'явиться робоче поле для програмування лічильника;
- в панелі інструментів активізуйте ікону Edit Wizard (Майстер проекту) і у таблиці Group, що з'явиться на екрані, виберіть альтернативу Keywords (Ключові слова);
- двічі клацніть по оператору IF і у робочому полі з'явиться його синтаксис з відповідним коментарем:

```
IF  (*EXPRESSION (must return a boolean value)*)
    THEN      (*If returned value of EXPRESSION = TRUE*)
              (*STATEMENT*);
END_IF;
```

Оголошення змінних дії Counter:

- Установіть курсор після оператора IF і в панелі інструментів лівою клавішею миші клацніть по іконі Variable (Змінна);
- у діалоговому вікні Variable Properties (Властивості змінної), що з'явиться, у рядку Name надрукуйте ім'я виходу лічильника кількості натисків стартової кнопки - Motor_Count.Q і натисніть ОК;
- діалогове вікно закриється, а біля оператора IF з'явиться логічна змінна Motor_Count.Q;

- установіть курсор після оператора THEN і надрукуйте вираз Cycle_Count:=Cycle_Count+1.

Оскільки змінна Cycle_Count вже оголошена, лічильник активізацій двигуна є створеним. Закрийте робоче поле переходу, а його код запам'ятайте.

Програмування переходу Trans:

Щоб показання лічильника скидалися після 10 активізацій двигуна створимо IL – мовою відповідну програму переходу Trans . Для цього:

- лівою клавішею миші двічі клацніть по переходу Trans;
- у діалоговому вікні, що з'явиться, виберіть IL-мову, закрийте його і на екрані з'явиться робоче поле для програмування;
- в панелі інструментів активізуйте ікону Edit Wizard (Майстер проекту) і у таблиці Group, що з'явиться на екрані, в альтернативі Operators (Оператори) виберіть оператор GT (Більш ніж);
- двічі клацніть по ньому і на робочому полі переходу Trans з'явиться синтаксис цього оператора:

```
LD var_1  
GT var_2  
ST var_out
```

Замість var1 надрукуйте вже оголошену змінну Cycle_Count, замість var2 – константу 10, а замість var_out – ім'я переходу Trans;

- установіть курсор на ім'я переходу Trans і клацніть лівою клавішею миші, з'явиться діалогове вікно Variables, у рядку Name якого надруковано ім'я переходу Trans;
- лівою клавішею миші клацніть по ОК;
- закрийте робоче поле переходу Trans і запам'ятайте створену програму.

Програмування часу роботи двигуна:

Оскільки двигун після вмикання працює 20 секунд, умовою переходу Trans1 має бути саме 20с, які відлічує таймер.

Для програмування таймера ST- мовою:

- лівою клавішею миші двічі клацніть по переходу Trans1;
- у діалоговому вікні, що з'явиться, виберіть ST-мову, закрийте вікно і на екрані з'явиться робоче поле для програмування таймера;
- в панелі інструментів активізуйте ікону Edit Wizard (Майстер проекту) і у таблиці Group, що з'явиться на екрані виберіть функціональний блок таймера TON;
- двічі клацніть по ньому і у діалоговому вікні Variable properties (Властивості змінної) змініть типове ім'я TON_1 на Motor_Time;
- натисніть ОК, діалогове вікно закриється, а в робочому полі залишиться ST- програма функціонального блока таймера Motor_Time:

```
Motor_Time(IN:>(* BOOL *),PT:>(* TIME *));  
(* BOOL *) :=Motor_Time.Q;  
(* TIME *) :=Motor_Time.ET;
```

- установіть курсор перед коментарем IN- входу функціонального блока Motor_Time і в панелі інструментів лівою клавішею миші клацніть по іконі Variable (Змінна);
- у діалоговому вікні Variable Properties (Властивості змінної), що з'явиться, у рядку Name надрукуйте ім'я Step1.X, яке відповідає назві кроку Step1 і натисніть ОК;
- діалогове вікно закриється, а IN- входу функціонального блока таймера Motor_Time присвоюється змінна Step1.X;
- установіть курсор перед коментарем PT- входу і в панелі інструментів клацніть по іконі Variable (Змінна);
- у діалоговому вікні Variable Properties (Властивості змінної), що з'явиться, у рядку Name надрукуйте константу T#20s, яка встановлює час роботи двигуна і натисніть ОК;
- діалогове вікно закриється, а PT- входу таймера присвоюється константа затримки роботи двигуна;
- установіть курсор перед коментарем змінної, яка присвоюється Q- виходу таймера і в панелі інструментів лівою клавішею миші клацніть по іконі Variable (Змінна);
- у діалоговому вікні Variable Properties (Властивості змінної), що з'явиться, у рядку Name надрукуйте ім'я переходу Trans1 і натисніть ОК;
- діалогове вікно закриється, і змінна Trans1 присвоюється Q- виходу функціонального блока таймера;
- установіть курсор перед коментарем ET- виходу таймера і в панелі інструментів клацніть по іконі Variable (Змінна);
- у діалоговому вікні Variable Properties (Властивості змінної), що з'явиться, у рядку Name надрукуйте ім'я внутрішньої змінної Act_Time, яка має відображати фактичний час роботи двигуна, у рядку Usage (Вживання) виберіть VAR, а у рядку Data Type (Тип даних) – Time;
- у рядку Local Variable Groups (Група локальних змінних) активізуйте Default (За замовчуванням);
- натисніть ОК, діалогове вікно закриється, а ET-виходу функціонального блока Motor_Time присвоюється змінна Act_Time.

Таким чином, відповідно до створеної програми після триразового натискання кнопки Motor_Start вмикається двигун і лічильник його активізацій. Оскільки у лівій гілці альтернативного розгалуження умови для переходу Trans відбуваються тільки після 10 вмикань двигуна програма виконується відповідно до умов переходу Trans1 правої гілки розгалуження, тобто після 20 секунд роботи двигуна у виконанні програми здійснюється стрибок на початковий крок і робота двигуна зупиняється. Для повторного запуску двигуна необхідно знову тричі натиснути стартову кнопку. Після десятої активізації роботи двигуна виконуються умови переходу Trans лівої гілки розгалуження і стрибок на початковий крок вже здійснюється через неї. Це призводить не тільки до зупинки двигуна, а і до обнуління лічильника його активізацій.

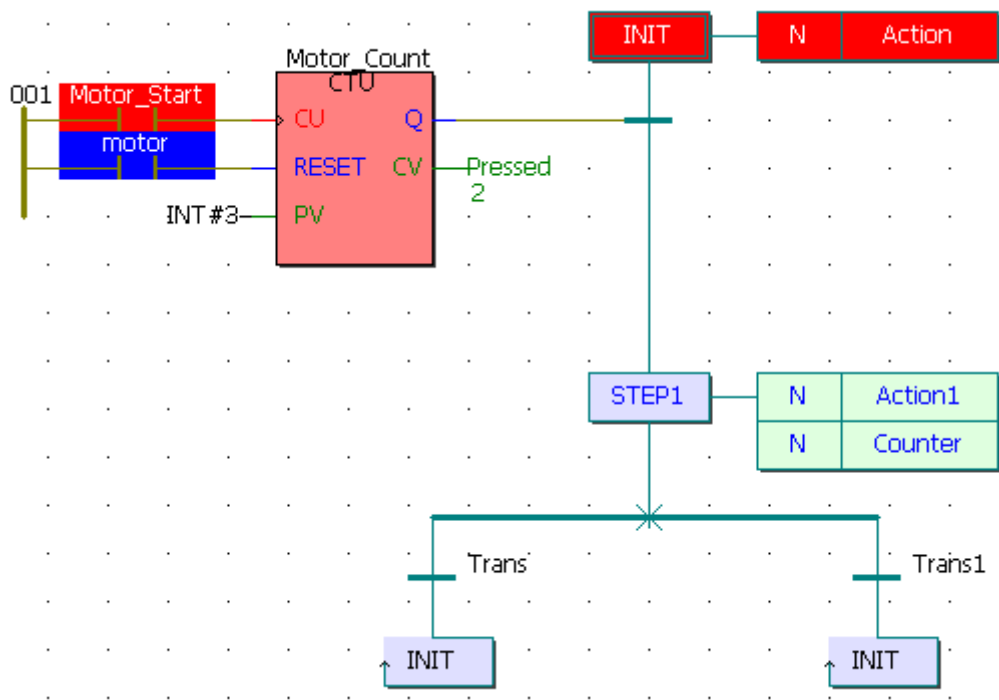
Після створення проектного коду в панелі інструментів натисніть ікону Make (Створювати), для компіляції проекту.

Якщо помилки є, відкоригуйте проект, якщо немає - завантажте його, використовуючи відомий шлях: Project Control Dialog → Download→Download.

Перевірка роботи програми:

- Для холодного запуску симулятора ПЛК натисніть кнопку Cold (Холодний) у діалоговому вікні Resource (Ресурс), що з'явиться після активізації ікони Project Control Dialog (Діалог управління проектом) в панелі інструментів;
- в панелі інструментів клацніть по іконі Debug on/off (Налагодження В/В), щоб перейти в оперативний режим роботи. При цьому усі змінні позначаються різними кольорами;
- лівою клавішею миші клацніть по Demoio_Driver унизу екрана, для відкриття симулятора I/O;
- тричі подвійно клацніть лівою клавішею миші по нульовому світлодіоду нульового модуля вводу In.

Програма почне виконуватися, про що свідчить перехід червоного кольору від початкового кроку INIT до кроку STEP1 після триразового натискання на нульовий світлодіод нульового модуля вводу симулятора.



Після затримки на 20 секунд червоний колір від кроку STEP1 через перехід Trans1 перестрибує на початковий крок INIT. При цьому з кожним новим запуском програми у дії Counter змінна лічильника Cycle_Count збільшується на одиницю. Для екстреної зупинки двигуна необхідно натиснути на перший світлодіод нульового модуля вводу. Коли кількість активізацій двигуна перевищує цифру 10, виконання програми продовжується через перехід Trans лівої гілки розгалуження на початковий крок INIT. При цьому робота двигуна зупиняється, а лічильник скидається у нуль.

Завдання для самостійної роботи

1. Розробити програму керування роботою двох двигунів, які після запуску одного з двигунів безперервно по черзі вмикаються і вимикаються. Тривалість роботи двигунів відповідно 5 і 10 секунд. Початковий запуск

першого двигуна здійснюється одноразовим натисненням пускової кнопки, а початковий запуск другого двигуна – дворазовим. При цьому двигуни можна у будь-яку мить зупинити, а кожне вмикання двигунів підраховується лічильником. Коли кількість вмикань двигунів досягає десяти, їх робота автоматично зупиняється.

2. Розробити програму керування роботою світлофора, що працює на перехресті доріг. Перехід із зеленого кольору на жовтий у світлофорі відбувається після триразового миготіння зеленого.

Контрольні запитання

1. Як створити першу SFC-мережу?
2. Як розширити існуючу мережу?
3. Як змінити початковий крок на звичайний, а звичайний – на початковий?
4. Як вставити в SFC-мережу альтернативні гілки?
5. Як вставити в SFC-мережу паралельні гілки?
6. Як користуватися режимом редагування SFC – гілок?
7. Як вставити в SFC-мережу стрибки?
8. Як додати нові блоки дій до одного існуючого?
10. Як змінити властивості кроку, блока дії, переходу?
11. Як створити деталь дії або переходу?
12. Як до переходу можуть бути приєднані функції або LD- мережі?
14. Як видалити фрагменти SFC-коду?
15. Як здійснюється програмування кроків і переходів SFC-схеми?
16. Коли у режимі налагодження елементи SFC-схеми позначаються червоним кольором?