

3.6. Методика програмування SFC-мовою

Створимо SFC-мовою програму керування роботою двигуна згідно з алгоритмом наведеним у 3.2.

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.

Коли новий проект з'явиться у вікні проектного дерева, можна починати програмування.

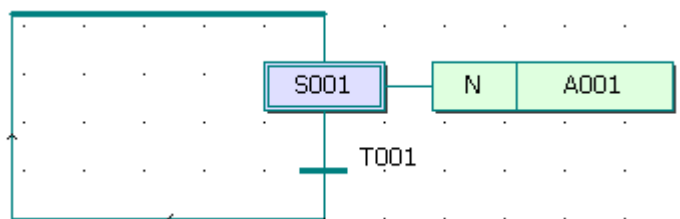
Будь- який проектний код, що створений SFC – мовою, складається з кроків і переходів, об'єднаних SFC – схемою. Тому перед тим, як їх програмувати необхідно побудувати структуру цієї схеми.

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 клав'яшею миші по іконі Create 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:

- двічі клацніть лівою клав'яшею миші по прямокутнику і у діалоговому вікні Step (Крок), що з'явилося, активізуйте Initial step (Початковий крок);
- закрийте вікно Step (Крок), натиснувши ОК, і на робочому полі звичайний крок перетвориться у початковий.

Для призначення імені компонентам SFC-схеми :

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

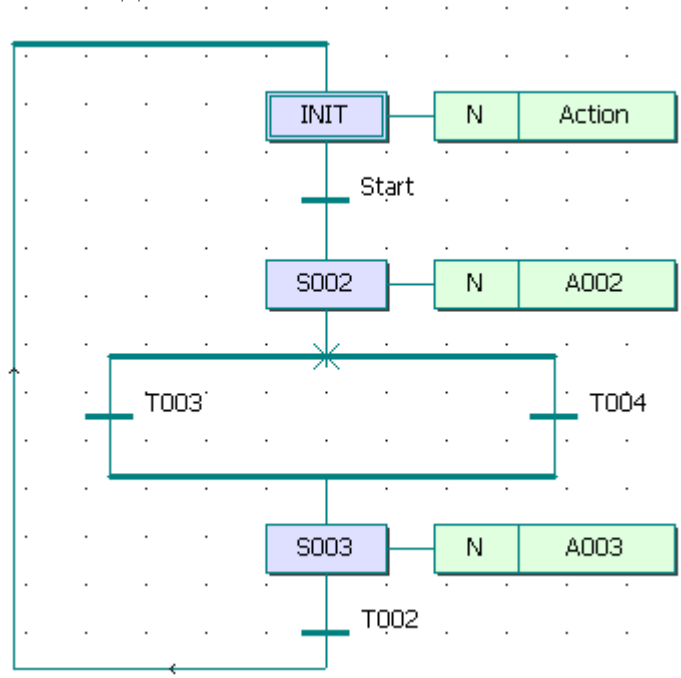
- таке саме зробіть послідовно з дією A001 та переходом T001, привласнюючи їм імена відповідно Action з класифікатором (Qualifier) N замість A001у діалоговому вікні Action Properties (Властивості дії) та Start

замість T001у діалоговому вікні Transition (Перехід), при цьому в обох вікнах альтернатива Detail (Деталь) має бути активізована;

Для створення другої пари (крок і перехід) клацніть лівою клавiшею 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'явилося, 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:

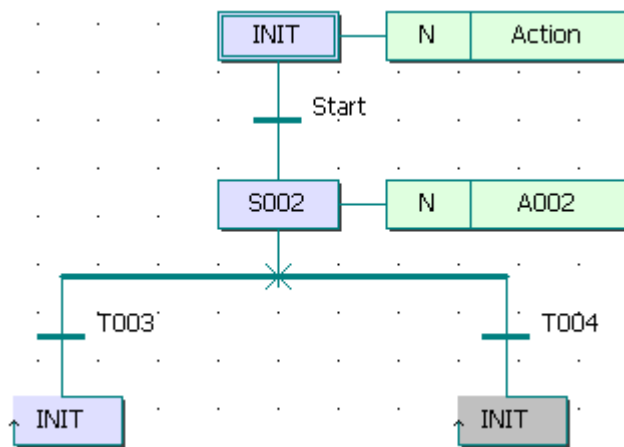
- позначте курсором перехід T003 і в панелі інструментів клацніть по іконі Creat step transition sequence(Створити послідовні крок і перехід), у лівій гілці з'являться крок S004 з дією A004 і перехід T005;

- правою клавiшею миші клацніть по кроку S004 і в меню, що з'явилося, лівою клавiшею 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'явилося, 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 клавiшею миші подвійно клацніть по ньому і у діалоговому вікні Step(Крок), що з'явилося, 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'явилося, виберіть ST-мову програмування і натисніть ОК. Відкриється робоче поле дії Action;

- користуючись синтаксисом ST-мови присвойте змінній motor 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'явилася на екрані, виберіть меню Keywords;

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

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

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

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

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;

```

Тут: Motor - змінна, що керує роботою двигуна;

Cycle_Count – змінна, що рахує кількість активізацій двигуна.

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

- у діалоговому вікні Variables (Змінні), що з'явилося, у рядку Name побачите ім'я змінної Motor, натисніть кнопку Apply (Застосувати);
- відкрийте сторінку діалогу Common (Загальний) і 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 вихідного каналу модуля виводу %QX0.0;
- відкрийте сторінку діалогу Local scope (Локальна компетенція) і активізуйте Default;
- натисніть ОК, діалогове вікно закриється, а змінна Motor автоматично з'явиться у списку змінних проекту.

Після цього установіть курсор на ім'я внутрішньої змінної Cycle_Count і в панелі інструментів лівою клавішею миші клацніть по іконі Variable (Змінна);

- у діалоговому вікні Variables (Змінні), що з'явилося, у рядку Name побачите ім'я змінної Cycle_Count, натисніть кнопку Apply (Застосувати);
- відкрийте сторінку діалогу Common (Загальний) і у рядку Usage (Вживання) виберіть VAR, а у рядку Data Type (Тип даних) - INT;
- відкрийте сторінку діалогу Local scope (Локальна компетенція) і активізуйте Default;
- закрийте робоче поле дії Action і запам'ятайте зроблену ST-програму, натиснувши у діалоговому вікні MULTIPROG, що з'явилося, відповідну кнопку.

Програмування переходу Start:

Умовою початку роботи двигуна є триразове натиснення 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 клавішею миші по переходу Start, у діалоговому вікні Transition (Перехід), що з'явилося, 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 клавішею 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;

- вилучте з LD- мережі праву шину живлення і котушку C001 послідовно помічаючи їх лівою клавішею миші і натискаючи клавішу Delet на клавіатурі EOM;

- 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 клавішею 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'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;

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

Таким чином на позначеному місці LD- мережі з'явиться лічильник.

- 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 клавішею миші по контакту C000, щоб оголосити змінну, яка має запуснути двигун. У діалоговому вікні Contact/Coil Properties (Властивості Контактів/Котушок), 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 (Вживання) виберіть VAR, а у рядку Data Type (Тип даних) - BOOL.

Призначімо змінній Motor_Start фізичну адресу симулянта модуля вводу/виводу I/O ПЛК:

- введіть у поле I/O address (Адреса вводу/виводу) фізичну адресу симулянта модуля вводу ПЛК - %IX0.0;

- відкрийте сторінку діалогу Local scope (Локальна компетенція) і активізуйте в неї Default;

- клацніть ОК, підтверджуючи діалог Contact/Coil Properties (Властивості Контактів/Котушок). Після цього змінна є оголошеною і вставленою у

відповідний список, а на робочому листку проекту замість C000 з'явиться локальна змінна Motor_Start;

- двічі клацніть по контакту C002 і у діалоговому вікні Contact/Coil Properties (Властивості Контакт/котушка), що з'явилося, з існуючого списку вже оголошених змінних виберіть Motor і клацніть лівою клавішею миші.

Оскільки змінна Motor визначена раніше, немає потреби конкретизувати Local scope (Локальна 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 константу INT#3, щоб двигун (Motor) спрацював після триразового натиснення стартової кнопки;

- 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);

- у рядку Name (Ім'я) надрукуйте Pressed - це внутрішня змінна, яка запам'ятовує поточне значення лічильника;

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

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

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

Програмування дії Action1 кроку Step1:

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

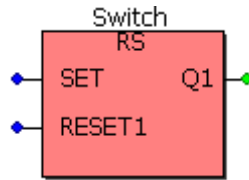
- лівою клавішею миші двічі клацніть по дії першого кроку Action1, у діалоговому вікні Insert (Вставлення), що з'явилося, виберіть FBD-мову програмування і натисніть ОК;

- у робочому полі дії Action1, що з'явилося, лівою клавішею миші позначте місце створення програмного коду;

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

- двічі клацніть по ньому і у діалоговому вікні Variable properties (Властивості змінної) змініть типово 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 блоку Switch:

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

- послідовно натисніть кнопки Apply (Застосувати) і OK, діалогове вікно закриється, а SET- входу перемикача присвоїться змінна Motor_Count.Q ;

- двічі клацніть по RESET1- входу перемикача і у діалоговому вікні Variable properties (Властивості змінної), що з'явилося, у полі Name надрукуйте ім'я нової змінної Emergency_Stop(Непередбачене зупинення);

- натисніть кнопки Apply (Застосувати) і відкрийте сторінку діалогу Common (Загальний блок). 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;

- відкрийте сторінку діалогу Local scope (Локальна компетенція) і активізуйте Default;

- натисніть OK, 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'явилося, у полі Name надрукуйте ім'я вже оголошеної змінної Motor;

- послідовно натисніть кнопки Apply (Застосувати) і OK, діалогове вікно закриється, а Q1- виходу перемикача присвоїться 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'явилося, 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'явилася 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 клав'яшею миші клацніть по іконі Variable (Змінна);

- у діалоговому вікні Variable Properties (Властивості змінної), що з'явилося, у рядку Name надрукуйте ім'я виходу лічильника кількості натисків стартової кнопки - Motor_Count.Q і послідовно натисніть кнопки Apply (Застосувати) і 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 клав'яшею миші двічі клацніть по переходу Trans;

- у діалоговому вікні, що з'явилося, 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'явилася 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 клав'яшею миші, з'явиться діалогове вікно Variables, у рядку Name якого надруковано ім'я переходу Trans;

- лівою клав'яшею миші послідовно клацніть по Apply (Застосувати) і 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'явилося, 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'явилася 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 кнопку Apply (Застосувати) і клацніть 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'явилося, у рядку Name надрукуйте ім'я Step1.X, яке відповідає назві кроку Step1 і послідовно натисніть кнопки Apply (Застосувати) і ОК.
- діалогове вікно закриється, а IN- входу функціонального блока таймера Motor_Time присвоїться 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'явилося, у рядку Name надрукуйте константу T#20s, яка встановлює час роботи двигуна і натисніть послідовно кнопки Apply (Застосувати) і ОК;
- діалогове вікно закриється, а PT- входу таймера присвоїться 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'явилося, у рядку Name надрукуйте ім'я переходу Trans1 і послідовно натисніть кнопки Apply (Застосувати) і ОК;
- діалогове вікно закриється, і змінна Trans1 присвоїться 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'явилося, у рядку Name надрукуйте ім'я внутрішньої змінної Act_Time, яка має відображати фактичний час роботи двигуна і натисніть кнопку Apply (Застосувати);

- відкрийте сторінку діалогу Common (Загальний). У рядку Usage (Вживання) виберіть VAR, а у рядку Data Type (Тип даних) – Time;
- відкрийте сторінку діалогу Local scope (Локальна компетенція) і активізуйте Default;
- натисніть ОК, діалогове вікно закриється, а ЕТ-виходу функціонального блока Motor_Time присвоїться змінна Act_Time.

Таким чином, відповідно до створеної програми після триразового натиснення 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:

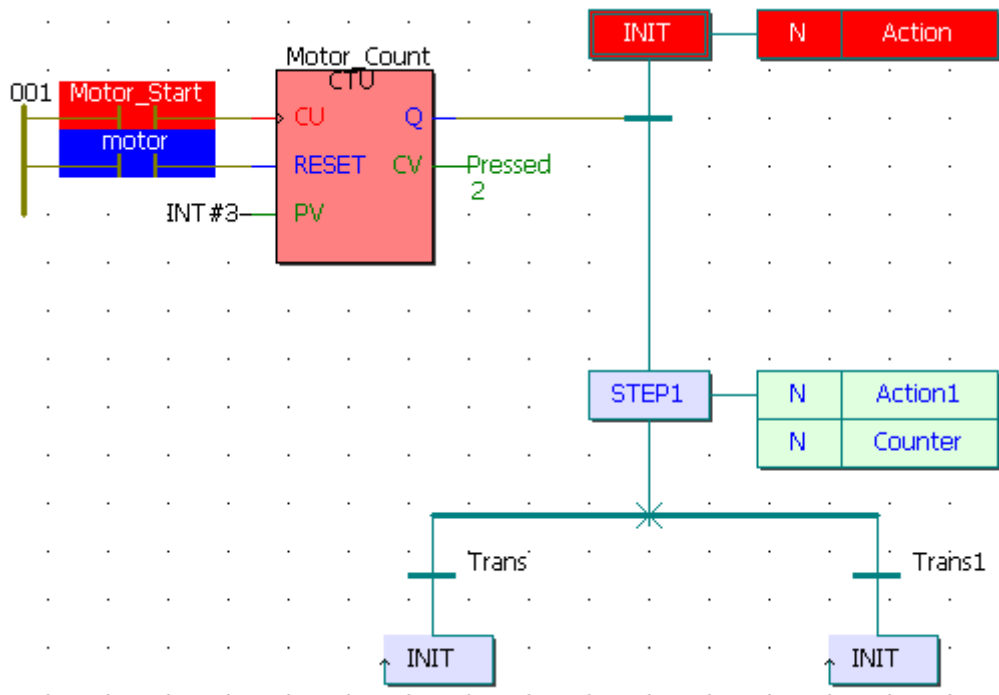
- Для холодного запуску симулятора ПЛК натисніть кнопку Cold (Холодний) у діалоговому вікні Resource (Ресурс), що з'являється після активізації ікони Project Control Dialog (Діалог управління проектом) в панелі інструментів;

- 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 натиснення 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.