

3.5. Методика програмування ST-мовою

Створимо ST-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). Подвійно 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 ST-мову програмування, бажані імена і типи конфігурації, ресурсу і задачі.

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

Створюючи проектний код відповідно до алгоритму керування роботою двигуна, будемо з бібліотеки системи програмування MULTIPROG послідовно вибирати функціональні блоки лічильника STU - для підрахунку кількості натисків на кнопку старту двигуна, RS перемикача з домінантою вимкнення - для створення можливості раптової зупинки двигуна, таймера TON – для визначення часу роботи двигуна, а також запрограмуємо лічильник активізацій двигуна.

Запрограмуємо ST-мовою функціональний блок лічильника для підрахунку кількості натисків на кнопку старту двигуна:

- На робочому полі проектного коду позначте курсором місце початку програмування;
- в панелі інструментів активізуйте ікону Edit Wizard (Майстер проекту) і у таблиці Group, що з'явилася на екрані виберіть функціональний блок STU;
- двічі клацніть по ньому і у діалоговому вікні Variable properties (Властивості змінної) змініть типове ім'я STU-1 на Motor-Count;
- клацніть ОК, діалогове вікно закриється, а на робочому листку проекту залишиться програма функціонального блока лічильника Motor_Count створеного ST- мовою:

```
Motor_Count(CU:=(* BOOL *),RESET:=(* BOOL *),PV:=(* INT *));  
(* BOOL *) :=Motor_Count.Q;  
(* INT *) :=Motor_Count.CV;
```

Оголосимо змінні блока Motor_Count:

У наведеній ST - програмі місця змінних займають коментарі – підказки їх типів.

- Установіть курсор перед коментарем CU- входу лічильника 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 змінної Motor_Start, за допомогою якої буде активізуватися двигун, і натисніть кнопку Apply (Застосувати);
- відкрийте сторінку діалогу Common (Загальний). 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 %IX0.0;

- відкрийте сторінку діалогу Local score (Локальна компетенція) і активізуйте Default;
- натисніть ОК, діалогове вікно закриється, а CU- входу лічильника Motor_Count присвоїться змінна Motor_Start;
- установіть курсор перед коментарем RESET- входу лічильника, щоб оголосити його змінну;
- в панелі інструментів клацніть по іконі Variable (Змінна), у діалоговому вікні Variable Properties (Властивості змінної), що з'явилося, у рядку Name надрукуйте ім'я змінної Motor, натисніть кнопку Apply (Застосувати);
- відкрийте сторінку діалогу Common(Загальний). У рядку Usage (Вживання) виберіть VAR, а у рядку Data Type (Тип даних) - BOOL;
- відкрийте сторінку діалогу Local score (Локальна компетенція) і активізуйте Default;
- натисніть ОК, діалогове вікно закриється, а RESET- входу лічильника Motor_Count присвоїться змінна Motor;
- установіть курсор перед коментарем PV- входу для визначення параметрів лічильника;
- в панелі інструментів клацніть по іконі Variable (Змінна), у діалоговому вікні Variable Properties (Властивості змінної), що з'явилося, у рядку Name надрукуйте константу INT#3, натисніть послідовно кнопки Apply (Застосувати) та ОК і PV- входу лічильника присвоються параметри, що визначають кількість натисків кнопки;
- установіть курсор перед коментарем змінної, що присвоюється Q- виходу лічильника Motor_Count;
- в панелі інструментів клацніть по іконі Variable (Змінна), у діалоговому вікні Variable Properties (Властивості змінної), що з'явилося, зі списку імен виберіть вже оголошену змінну Motor і натисніть кнопку Apply (Застосувати);
- відкрийте сторінку діалогу Common (Загальний). 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 score (Локальна компетенція) і активізуйте Default;
- натисніть ОК, діалогове вікно закриється, а Q- виходу лічильника Motor_Count присвоїться змінна Motor;
- установіть курсор перед коментарем змінної, яка присвоюється CV- виходу лічильника Motor_Count;
- 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 –Pressed, яка буде відображати кількість натисків на кнопку пуску двигуна, і клацніть по кнопці Apply (Застосувати);
- відкрийте сторінку діалогу Common(Загальний). У рядку Usage (Вживання) виберіть VAR, а у рядку Data Type (Тип даних) - INT;
- натисніть ОК, діалогове вікно закриється, а CV- виходу лічильника Motor_Count присвоїться змінна Pressed.

Запрограмуємо непередбачену зупинку двигуна:

- Позначте курсором новий рядок для продовження програми;
- в панелі інструментів активізуйте ікону Edit Wizard (Майстер проекту) і у таблиці Group, що з'явилася на екрані виберіть функціональний блок RS, який має властивості перемикача із домінантою вимикача;
- двічі клацніть по ньому і у діалоговому вікні Variable properties (Властивості змінної) змініть типове ім'я блока на ім'я його екземпляру - RS-1;
- натисніть кнопку Apply (Застосувати) і клацніть ОК, діалогове вікно закриється, а у робочому листку проекту залишиться ST- програма функціонального блока RS:

```
RS_1 (SET:=( * BOOL * ), RESET1:=( * BOOL * ));  
(* BOOL *) :=RS_1.Q1;
```

Оголосимо змінні блока RS:

У наведеній ST - програмі місця змінних займають коментарі – підказки їх типів.

- Установіть курсор перед коментарем SET- входу функціонального блока RS 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 внутрішньої змінної Out і натисніть кнопку Apply (Застосувати);
- відкрийте сторінку діалогу Common (Загальний). У рядку Usage (Вживання) виберіть VAR, а у рядку Data Type (Тип даних) - BOOL;
- відкрийте сторінку діалогу Local scope (Локальна компетенція) і активізуйте Default;
- натисніть ОК, діалогове вікно закриється, а SET- входу функціонального блока RS-1 присвоїться змінна Out;
- установіть курсор перед коментарем RESET1-входу. Оскільки двигун має зупинитися після закінчення установленого часу роботи, а також примусово у будь-який момент, для RESET1-входу оголосимо дві альтернативні змінні;
- в панелі інструментів клацніть по іконі Variable (Змінна), у діалоговому вікні Variable Properties (Властивості змінної), що з'явилося, у рядку Name надрукуйте ім'я першої альтернативної змінної Emergency_Stop (Непередбачений_Останов) і натисніть кнопку Apply (Застосувати);
- відкрийте сторінку діалогу Common(Загальний). 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 ОК, діалогове вікно закриється, а RESET1-входу присвоїться перша альтернативна змінна Emergency_Stop;
- надрукуйте поряд з нею логічний оператор OR 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 другої альтернативної змінної Mot_Time, яка зупинить двигун за командою таймера;

- послідовно натисніть кнопки Apply (Застосувати) та ОК і друга альтернативна змінна Mot_Time присвоїться RESET1-входу;
- установіть курсор перед коментарем змінної, яка присвоюється виходу Q1 функціонального блока RS;
- в панелі інструментів клацніть по іконі Variable (Змінна), у діалоговому вікні Variable Properties (Властивості змінної), що з'явилося, зі списку імен виберіть вже оголошену змінну Motor і натисніть кнопку Apply (Застосувати);
- відкрийте сторінку діалогу Common (Загальний). 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 Q1-виходу функціонального блока RS-1 присвоїться змінна Motor.

Запрограмуємо час роботи двигуна:

- Позначте курсором новий рядок для продовження програми;
- 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 TON1 на M_Time;
- натисніть кнопку Apply (Застосувати) і клацніть ОК, діалогове вікно закриється, а на робочому листку проекту залишаться ST- програма функціонального блока таймера M_Time:

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

- установіть курсор перед коментарем IN- входу функціонального блока M_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 вже оголошеної змінної Motor і послідовно натисніть кнопки Apply (Застосувати) і ОК;
- діалогове вікно закриється, а IN- входу функціонального блока таймера M_Time присвоїться змінна Motor;
- 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 присвоюється константа затримки роботи двигуна;

- установіть курсор перед коментарем змінної, якої присвоюється Q-вихід 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 вже оголошеної змінної Mot_Time і послідовно натисніть кнопки Apply (Застосувати) і ОК;
- діалогове вікно закриється, і змінна Mot_Time 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 Actual_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 ET-виходу функціонального блока M_Time присвоїться змінна Actual_Time.

Запрограмуємо лічильник активізації двигуна:

- Позначте курсором новий рядок для продовження програми;
- активізуйте в панелі інструментів ікону Edit Wizard (Майстер проекту) і 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 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 вже оголошеної змінної Out і послідовно натисніть кнопки Apply (Застосувати) і ОК;
- діалогове вікно закриється, а біля оператора IF з'явиться логічна змінна Out;
- установіть курсор після оператора THEN і надрукуйте вираз Cycle_Count:=Cycle_Count+1;

Для компіляції проекту в панелі інструментів натисніть ікону Make (Створювати).

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

Після цієї дії ST-програма набуває завершений вигляд:

```

(*Програмування лічильника кількості натисків стартової кнопки*)
Motor_Count(CU:=Motor_Start,RESET:=Motor,PV:=INT#3);
out:=Motor_Count.Q;
Pressed(* INT *):=Motor_Count.CV;
(*Програмування непередбаченої зупинки двигуна*)
RS_1(SET:=out,RESET1:=Emergency_Stop OR Mot_Time);
Motor:=RS_1.Q1;
(*Програмування таймера часу роботи двигуна*)
M_Time(IN:=Motor,PT:=T#20s);
Mot_Time:=M_Time.Q;
Actual_Time:=M_Time.ET;
(*Програмування лічильника активацій двигуна*)
IF out
    THEN Cycle_Count:=Cycle_Count+1;
END_IF;

```

Перевіримо роботу програми:

- Натисніть кнопку Cold (Холодний) у діалоговому вікні Resource (Ресурс) для холодного запуску симулятора ПЛК;
- 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;
- клацніть лівою клавішею миші по Demoio_Driver, що унизу екрана, для відкриття симулятора I/O;
- тричі подвійно клацніть лівою клавішею миші по нульовому світлодіоду нульового модуля вводу In.

Програма почне виконуватися, а біля змінної Actual_Time таймера M_Time будуть змінюватися секунди затримки. При цьому з кожним новим запуском програми вміст лічильника Cycle_Count збільшується на одиницю.

```

(*Програмування лічильника кількості натисків стартової
кнопки*)
FALSE Motor_Count(CU:=Motor_Start,RESET:=Motor,PV:=INT#3);
FALSE out:=Motor_Count.Q;
0 Pressed(* INT *):=Motor_Count.CV;
(*Програмування непередбаченої зупинки двигуна*)
FALSE RS_1(SET:=out,RESET1:=Emergency_Stop OR Mot_Time);
TRUE Motor:=RS_1.Q1;
(*Програмування таймера часу роботи двигуна*)
TRUE M_Time(IN:=Motor,PT:=T#20s);
FALSE Mot_Time:=M_Time.Q;
5.700 Actual_Time:=M_Time.ET;
(*Програмування лічильника активацій двигуна*)
FALSE IF out
    2 THEN Cycle_Count:=Cycle_Count+1;
END_IF;

```