

Лабораторна робота №4 Створення програмного коду ІЛ - мовою

Знайомство з редактором ІЛ –мови

Тіло коду ІЛ-мовою редагується текстовим редактором, який друкує інструкції або вставляє їх у робочий листок за допомогою Edit Wizard (Майстер редагування). При цьому елементи синтаксису ІЛ-мови зображуються різними кольорами: оператори – блакитні; змінні й імена зразків – чорні; коментарі – зелені.

Для того, щоб редагувати тіло коду ІЛ-мовою, необхідно у проектному дереві відкрити робочий листок, подвійно клацаючи по відповідній іконі теки Logical POUs(Логічні POUs).

За допомогою Майстра редагування інструкції вставляються у робочий листок тіла коду із вже завершеною структурою і користувач її просто заповнює змінними.

Якщо Edit Wizard (Майстер редагування) не активізований, треба натиснути на клавіатурі <SHIFT>+<F2> або <Alt>+<3>, або в панелі інструментів клацнути по іконі Edit Wizard.

Щоб вставити інструкцію:

- відкрийте робочий листок тіла коду, курсором визначте положення нової інструкції на екрані і клацніть лівою клавішею миші;
- у відкритому вікні Майстра редагування зі списку Group (Група), виберіть Operators (Оператори) і клацніть лівою клавішею. Майстер редагування покаже доступні оператори;
- виберіть необхідний оператор і двічі клацніть по ньому, щоб вставити у позначене місце. Деякі оператори вставляються окремо, а деякі - разом з іншими, відповідно до синтаксису ІЛ-мови. Наприклад, при вставленні оператора ADD, в робочому полі з'являється така мовна структура:

```
LD (* IN1 as ANY_NUM *)  
ADD (* IN2 as ANY_NUM *)  
ST (* Result as ANY_NUM *)
```

Щоб декларувати фактичну змінну біля оператора, позначте курсором змінну, що з'явиться з мовною структурою за замовчуванням і натисніть<F5> або в панелі інструментів клацніть по іконі Variable (Змінна). Відкриється діалогове вікно Variables (Змінні).

Для того, щоб вставити в мовну структуру змінну, яка вже декларована:

- на сторінці Variables (Змінні) діалогового вікна Variables (Змінні) у списку Name (Ім'я) вже декларованих змінних позначте бажане ім'я і клацніть лівою клавішею миші;
- позначене ім'я увійде у перший рядок списку Name (Ім'я);
- у рядку Local Variable Groups (Група локальній змінній) активізуйте Default і закрийте діалог. Змінна опиниться у позначеному місці робочого листка тіла коду.

Щоб декларувати нову змінну:

- надрукуйте нову змінну у бажаній позиції коду; наприклад, LD T_value;

- позначте курсором нове ім'я змінної і в панелі інструментів клацніть по іконі Variable (Змінна);
- з'явиться діалогове вікно Variables (Змінні) з надрукованим іменем T_value у першому рядку поля Name (Ім'я);
- виберіть відповідні установки для змінної у рядках Usage (Вжити), Data Type (Тип Даної), а також в інших, якщо потрібно,
- а у рядку Local Variable Groups (Група локальній змінній) активізуйте Default (За замовчуванням) і закрийте діалогове вікно.

Змінна вставиться у робочий листок тіла коду, а її декларація автоматично увійде у робочий листок сітки відповідних змінних.

У разі глобальної змінної, декларація вставляється у сітку локальних змінних з ключовим словом VAR_EXTERNAL і у робочий листок сітки глобальних змінних з ключовим словом VAR_GLOBAL.

Можна замінити змінну на таку, яка вже оголошена в робочому листку сітки змінних. Можна також замінити ім'я змінної на нове і таким чином декларувати при заміні нову змінну.

Щоб замінити змінну:

- позначте змінну, яка змінюється, курсором і натисніть <F5>. З'явиться діалогове вікно Variables (Змінні);
- для заміни поточної змінної на вже декларовану змінну виберіть у діалоговій сторінці Variables (Змінні) зі списку змінних бажане ім'я і далі продовжуйте, як при вставленні вже оголошеної змінної.

Для заміни поточної змінної новою, у діалоговій сторінці Variables (Змінні) введіть нове ім'я, а потім продовжуйте, як при декларуванні нової змінної.

Для зміни властивостей поточної змінної відкрийте діалогову сторінку Variables (Змінні), зробіть необхідні заміни і закрийте діалогове вікно.

Щоб в текстовому редакторі викликати функцію, використовується її ім'я у якості оператора та відповідні параметри, як це показано в наступному прикладі:

```
LD Inpar1
  Funktion name   par2,par3
ST var1
```

Перший декларований вхідний параметр «inpar1» завантажується у попередній рядок викликаної функції. Усі інші вхідні параметри «par2, par3» записуються через кому в другому рядку, як операнди. Результат зберігається змінною «var1», як це показано в останньому рядку прикладу.

Більш зручніше і надійніше редагування викликаної функції здійснювати за допомогою Майстра редагування (Edit Wizard). Якщо Майстер редагування невидимий на екрані, натисніть <Shift>+<F2>або <Alt>+<3> або в панелі інструментів клацніть по іконі Edit Wizard (Майстер редагування).

- Виберіть в робочому листку тіла коду позицію, де нова функція має бути вставлена, і клацніть лівою клавішею миші;
- відкрийте список Group (Група) у вікні Майстра редагування і виберіть курсором Functions (Функції). З'явиться список доступних функцій;
- виберіть бажану функцію і подвійно клацніть по ній лівою клавішею миші. Функція автоматично з'явиться у зазначеному місці робочого поля;

– замініть зелені коментарі (оточені круглими дужками і зірочками) необхідними елементами.

Наступний приклад показує вже редаговану функцію GE, яка вставлена в робочий листок, використовуючи Майстра редагування.

```
LD (* IN1 as ELEMENTARY *)
GE (* IN2 as ELEMENTARY *)
ST (* Result as BOOL *)
```

Функціональні блоки викликаються в ІЛ-редакторі за допомогою оператора CAL й імені зразка функціонального блока, як операнда.

Більш зручніше, і це перешкоджає появі помилок, викликати функціональний блок за допомогою Майстра редагування (Edit Wizard).

Якщо Майстер редагування невидимий на екрані, натисніть <Shift>+<F2> або <Alt>+<3>, або в панелі інструментів клацніть по іконі Edit Wizard (Майстер редагування).

– Виберіть позицію в тілі коду, де буде вставлено новий функціональний блок. Клацніть лівою клавішею миші;

– відкрийте список Group (Група) у вікні Майстра редагування (Edit Wizard) і виберіть групу Function blocks (Функціональні блоки). Майстер редагування покаже доступні функціональні блоки;

– подвійно клацніть по бажаному функціональному блоку. З’явиться діалогове вікно Variables (Змінні), де у полі Name (Ім’я) за замовчуванням пропонується ім’я екземпляра блока (наприклад, для функціонального блока STU пропонується ім’я „STU_n” де n – порядковий номер цього імені екземпляра).

Щоб призначити екземпляру функціонального блока бажане ім’я, необхідно:

– ввести у перший рядок діалогової сторінки Variables (Змінні) нове ім’я зразка або вибрати ім’я з текстового поля Name (Ім’я);

– натиснути ОК, і функціональний блок автоматично вставиться у позицію, вказану текстовим курсором, а його декларація автоматично увійде у групу локальних змінних;

– змініть підказки, що оточені круглими дужками, на імена вхідних і вихідних змінних, необхідними.

Наступний приклад показує вже редагований STU- функціональний блок, викликаний за допомогою Майстра редагування.

```
LD (* BOOL *)
ST STU_1.CU
LD (* BOOL *)
ST STU_1.RESET
LD (* INT *)
ST STU_1.PV
CAL STU_1
LD STU_1.Q
ST (* BOOL *)
LD STU_1.CV
ST (* INT *)
```

Вже редагований функціональний блок STU з іменем STU_1

Зелені коментарі займають місця, які мають бути переписані фактичними значеннями й іменами

В ІЛ робочих листках можуть використовуватися стрибки, які редагуються за допомогою оператора JMP, а також модифікаторів „C” або „CN” і мітки.

Щоб редагувати стрибки:

- встановіть текстовий курсор у бажане місце робочого листка тіла коду;
- введіть оператор JMP, додаткові модифікатори і мітку;
- виберіть в тілі коду місце, куди доведеться ставити мітку;
- введіть мітку, двокрапки й інструкцію.

Приклад використання стрибка має вигляд:

```
LD    var1
EQ    INT#100
JMPC  label
LD    var2
ADD   var3
ST    var4
Label:LD %IX2.2
```

Система програмування може здійснити перекомпіляцію існуючого POU, створеного ІЛ-, FBD- або LD-мовою, кожною з інших двох мов цього переліку.

Для реалізації мовної конверсії:

- переконайтесь, що проект вже побудований і компільований;
- у проектному дереві відкрийте піддерево Logical POUs (Логічні POUs) і позначте ікону існуючого POU;



- клацніть правою клавішею миші по іконі POU, щоб відкрити його контекстне меню;
 - виберіть пункт Source conversion (Початок конверсії) і клацніть лівою клавішею. З'явиться діалогове вікно Source conversion (Початок конверсії).
 - У діалоговому вікні Source conversion (Початок конверсії), що з'явилося, активізуйте перемикач Overwrite source POU (Переписати початковий POU), якщо бажаєте, визначте нове ім'я POU і виберіть мову перекомпіляції.
 - Закрийте діалогове вікно і у дереві проекту на місці позначеного POU з'явиться новий POU - конвертований вибраною мовою.
- В панелі інструментів клацніть лівою клавішею миші по іконі Compile Worksheet (Компілювати Робочий листок), щоб компілювати POU після конверсії.

Створення проекту користувача

Створити ІЛ-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 IL-мову програмування, бажані імена і типи конфігурації, ресурсу і задачі.

Програмування IL-мовою функціонального блока 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(Група), що з'явиться, 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 послідовність IL-інструкцій з коментарями, яка програмує функціональний блок Motor_Count.

```
LD (* BOOL *)
ST Motor_Count.CU
LD (* BOOL *)
ST Motor_Count.RESET
LD (* INT *)
ST Motor_Count.PV
CAL Motor_Count
LD Motor_Count.Q
ST (* BOOL *)
LD Motor_Count.CV
ST (* INT *)
```

У наведеній послідовності IL-інструкцій місця змінних займають коментарі – підказки їх типів.

- Установіть курсор перед коментарем першого рядка і в панелі інструментів лівою клавішею миші клацніть по іконі Variable (Змінна);

- у діалоговому вікні Variable Properties (Властивості змінної), що з'явиться, оголошіть змінну Motor_Start з фізичною адресою вхідного каналу модуля вводу %IX0.0;

- у першому рядку між інструкцією LD і коментарем з'явиться змінна Motor_Start, яка наступною інструкцією ST запам'ятовується у CU-вході лічильника Motor_Count.

- установіть курсор перед коментарем третього рядка, щоб оголосити змінну, яка призначена для RESET(Скид)-входу лічильника Motor_Count;

- в панелі інструментів клацніть по іконі Variable (Змінна) і у діалоговому вікні Variable Properties (Властивості змінної), що з'явиться, оголошіть змінну Motor з фізичною адресою вихідного каналу модуля виводу %QX0.0;

- у третьому рядку між інструкцією LD і коментарем з'явиться змінна Motor, яка наступною інструкцією ST запам'ятовується у RESET- вході лічильника Motor_Count;

- установіть курсор перед коментарем п'ятого рядка для визначення параметрів PV - входу;

- в панелі інструментів клацніть по іконі Variable (Змінна), у діалоговому вікні Variable Properties (Властивості змінної), що з'явиться, у рядку Name надрукуйте константу INT#3, натисніть ОК і у п'ятому рядку IL-інструкцій з'являться параметри лічильника;

- установіть курсор перед коментарем дев'ятого рядка, і оголосіть змінну Out, якою керує Q-вихід лічильника Motor_Count;

- Після натискання ОК у дев'ятому рядку між інструкцією ST і коментарем з'явиться змінна Out;

- установіть курсор перед коментарем останнього рядка, щоб оголосити змінну, яка запам'ятовує імпульси лічильника Motor_Count;

- в панелі інструментів клацніть по іконі Variable (Змінна), у діалоговому вікні Variable Properties (Властивості змінної), що з'явиться, у рядку Name надрукуйте Pressed, у рядку Usage (Вживання) виберіть VAR, а у рядку Data Type (Тип даних) - INT;

- у рядку Local Variable Groups (Група локальній змінній) активізуйте Default (За замовчуванням) і натисніть ОК, діалогове вікно закриється, а у останньому рядку з'явиться змінна Pressed.

Програмвання таймера роботи двигуна:

- Позначте курсором новий рядок для продовження програми;

- активізуйте в панелі інструментів ікону Edit Wizard (Майстер проекту) і у таблиці Group(Група), що з'явиться, виберіть функціональний блок TON;

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

- натисніть ОК, діалогове вікно закриється, а на робочому листку проекту залишаться IL – інструкції, що програмують таймер M_Time.

```
LD (* BOOL *)
ST M_Time.IN
LD (* TIME *)
ST M_Time.PT
CAL M_Time
LD M_Time.Q
ST (* BOOL *)
LD M_Time.ET
ST (* TIME *)
```

За аналогією попереднього функціонального блока оголосіть змінні таймера M_Time, біля операндів LD, враховуючи, що на IN-вході таймера змінна Motor, на PT-вході константа часу T#20s, на Q-виході – змінна Mot_time, а ET-виході – змінна Actual_Time.

Програмування непередбаченої зупинки двигуна.

Позначте курсором новий рядок і оголосіть функціональний блок перемикача RS:

```

LD   (* BOOL *)
ST   RS_1.SET
LD   (* BOOL *)
ST   RS_1.RESET1
CAL  RS_1
LD   RS_1.Q1
ST   (* BOOL *)

```

На SET- вхіді оголошіть змінну з ім'ям Out і натисніть ОК. Діалогове вікно закриється, а у першому рядку між інструкцією LD і коментарем з'явиться змінна Out, яка наступною інструкцією ST запам'ятовується у SET - вхіді перемикача RS-1;

- на RESET1-вхіді оголошіть змінну Emergency_Stop(Непередбачена зупинка з фізичною адресою вхідного каналу модуля вводу %IX0.1;

- натисніть ОК і у третьому рядку між інструкцією LD і коментарем з'явиться змінна Emergency_Stop (Непередбачена зупинка), яка наступною інструкцією ST запам'ятовується у RESET1- вхіді перемикача RS-1.

Оскільки двигун має зупинитися не тільки під впливом вхідної змінної Emergency_Stop(Непередбачена зупинка), а ще і після 20 секунд роботи, додамо з клавіатури після змінної Emergency_Stop оператора LD альтернативну вихідну змінну Mot_Time таймера M_Time з оператором OR.

Установіть курсор перед коментарем останнього рядка IL – інструкцій перемикача RS-1 і в панелі інструментів лівою клавішею миші клацніть по іконі Variable (Змінна);

- у діалоговому вікні Variable Properties (Властивості змінної), що з'явиться, виберіть ім'я Motor і натисніть ОК, оскільки змінна Motor вже оголошена;

- діалогове вікно закриється, а у останньому рядку між інструкцією LD і коментарем з'явиться змінна Motor, якою керує Q- вихід перемикача RS-1.

В цілому IL- програма керування роботою двигуна має такий вигляд:

```

(*Програмування лічильника кількості натисків стартової кнопки*)
LD Motor_Start (* BOOL *)
ST Motor_Count.CU
LD Motor (* BOOL *)
ST Motor_Count.RESET
LD int#3 (* INT *)
ST Motor_Count.PV
CAL Motor_Count
LD Motor_Count.Q
S out (* BOOL *)
LD Motor_Count.CV
ST Pressed (* INT *)
(*Програмування таймера*)
LD Motor (* BOOL *)
ST M_Time.IN
LD t#20s (* TIME *)
ST M_Time.PT
CAL M_Time
LD M_Time.Q
R Mot_Time (* BOOL *)
LD M_Time.ET
ST Actual Time (* TIME *)
(*Програмування непередбаченої зупинки двигуна*)
LD out(* BOOL *)
ST RS_1.SET
LD Emergency_Stop(* BOOL *)
OR Mot_Time
ST RS_1.RESET1
CAL RS_1
LD RS_1.Q1
ST Motor(* BOOL *)

```

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.

Перевіряння роботи програми здійснить в оперативному режимі за допомогою ікони Debug on/off (Налагодження В/В) і симулятора I/O.

При виконанні програми біля змінних різнокольоровим текстом буде висвічуватися їх логічний стан, а при змінній Actual_Time таймера M_Time будуть змінюватися секунди затримки.


```

      (*Програмування лічильника кількості натисків
TRUE   |   стартової кнопки*)
TRUE LD  Motor_Start (* BOOL *)
TRUE ST  Motor_Count.CU
TRUE LD  Motor      (* BOOL *)
      ST  Motor_Count.RESET
      3 LD  int#3      (* INT *)
      ST  Motor_Count.PV
FALSE CAL Motor_Count
TRUE LD  Motor_Count.Q
      0 S   out        (* BOOL *)
      0 LD  Motor_Count.CV
      ST  Pressed     (* INT *)
TRUE   |   (*Програмування таймера*)
TRUE LD  Motor      (* BOOL *)
      ST  M_Time.IN
20.000 LD t#20s      (* TIME *)
      ST  M_Time.PT
FALSE CAL M_Time
FALSE LD  M_Time.Q
4.200 R   Mot_Time   (* BOOL *)
4.200 LD  M_Time.ET
      ST  Actual_Time (* TIME *)
TRUE   |   (*Програмування непередбаченої зупинки двигуна*)
TRUE LD  out (* BOOL *)
FALSE ST  RS_1.SET
FALSE LD  Emergency_Stop (* BOOL *)
FALSE OR  Mot_Time
      ST  RS_1.RESET1
TRUE CAL RS_1
TRUE LD  RS_1.Q1

```

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 натисканням 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 миготіння світлодіода з одночасним підрахунком кількості вмикань.

Контрольні запитання

1. Як вставити IL-інструкцію у робочий листок тіла коду?
2. Як замінити змінну на таку, що вже оголошена?
3. Як у текстовому редакторі викликати функцію?
4. Як у текстовому редакторі викликати функціональний блок?

5. Як редагуються стрибки в робочих листках тіла коду?
6. До яких мов можна застосувати перекомпіляцію?
7. Як запрограмувати ПЛ-мовою функціональний блок?
8. Як оголосити змінні функціонального блока?
9. Як запрограмувати лічильник кількості натисків стартової кнопки?
10. Як запрограмувати екстрену зупинку двигуна?