Міністерство освіти і науки України

ОДЕСЬКА НАЦІОНАЛЬНА АКАДЕМІЯ ЗВ'ЯЗКУ ім. О.С. ПОПОВА

Кафедра комп'ютерно-інтегрованих технологічних процесів і виробництв

Методичні вказівки до виконання лабораторної роботи "Програмування ПЛК. Вивчення мови функціональних блокових діаграм (FBD)" з дисципліни "Мікропроцесорні та програмні засоби автоматизації"

Модуль 1 Керування процесом за допомогою цифрових пристроїв у засобах автоматизації. Побудова систем управління на контролерах та мікроконтролерах різних типів

Для студентів галузі знань 0502 за напрямом вищої освіти 6.050202 "Автоматизація та комп'ютерно-інтегровані технології"

Одеса 2016

Рецензент – Стопакевич А.О.

Тігарєв А.М. Методичні вказівки до виконання лабораторної роботи "Програмування ПЛК. Вивчення мови функціональних блокових діаграм (FBD)" / Тігарєв А.М. – Одеса: ОНАЗ ім. О.С. Попова, 2016. – 32 с.

У методичних вказівках розглянуто реалізацію автоматизації ділянок технологічного процесу за допомогою мови функціональних блокових діаграм (FBD). Указано вимоги до виконання програм мовою програмування FBD за стандартом MEK 61131.

Ухвалено на кафедрі комп'ютерно-інтегрованих технологічних процесів і виробництв і рекомендовано до друку Протокол № 6 від 22.02.2016 р. Затверджено методичною радою академії зв'язку Протокол № 6

© Тігарєв А.М., 2016 © ОНАЗ ім. О.С. Попова, 2016

3MICT

Лабораторна робота № 3. ПРОГРАМУВАННЯ ПЛК. ВИВЧЕННЯ МОВИ ФУНКЦІОНАЛЬНИХ БЛОКОВИХ ДІАГРАМ (FBD)

4	
1 Мета роботи	4
2 Основні положення	4
2.1 Відображення РОU	5
2.2 З'єднувальні лінії	6
2.3 Порядок виконання FBD	6
2.4 Інверсія логічних сигналів	6
2.5 З'єднувачі й зворотні зв'язки	6
2.6 Мітки, переходи й повернення	7
2.7 Вирази ST у FBD	7
2.8 Редактор FBD	9
3 Контрольні питання	15
4 Домашнє завдання	16
5 Лабораторне завдання	16
6 Зміст протоколу	17
7 Перелік рекомендованої літератури	17
Лабораторная работа № 3. ПРОГРАММИРОВАНИЕ ПЛК.	
ИЗУЧЕНИЕ ЯЗЫКА ФУНКЦИОНАЛЬНЫХ БЛОКОВЫХ	
ДИАГРАММ (FBD)	
18	
1 Цель работы	
2 Основные положения	
2.1 Отображение компонентов (POU)	
2.2 Соединительные линии	20
2.3 Порядок выполнения FBD	20
2.4 Инверсия логических сигналов	20
2.5 Соединители и обратные связи	21
2.6 Метки, переходы и возврат	21
2.7 Выражения ST в FBD	
2.8 Редактор FBD	23
3 Контрольные вопросы	
4 Домашнее задание	
5 Лабораторное задание	
6 Содержание протокола	
7 Перечень рекомендованной литературы	

Лабораторна робота № 3

ПРОГРАМУВАННЯ ПЛК. ВИВЧЕННЯ МОВИ ФУНКЦІОНАЛЬНИХ БЛОКОВИХ ДІАГРАМ (FBD)

1 Мета роботи

Метою роботи є ознайомлення з принципами побудови систем автоматичного керування з застосуванням програмованих логічних контролерів, а також набуття навичок розробки програм для програмованих логічних контролерів мовою FBD за стандартом MEK 61131 у середовищі програмування CoDeSys.

2 Основні положення

FBD (Function Block Diagram) – це графічна мова програмування. Діаграма FBD дуже нагадує принципову схему електронного пристрою на мікросхемах (див. рис. 2.1). На відміну від LD «провідники» у FBD можуть проводити сигнали (передавати змінні) будь-якого типу (логічний, аналоговий, час і т.д.). Шини живлення на FBD діаграмі не показуються. Виходи блоків можуть бути подані на входи інших блоків або безпосередньо на виходи ПЛК. Самі блоки, показані на схемі як «чорні ящики», можуть виконувати будь-які функції. FBD-схеми дуже чітко відбивають взаємозв'язок входів і виходів діаграми. Якщо алгоритм одвічно добре описується з позиції сигналів, то його FBD-подання завжди виходить наочніше, ніж у текстових мовах [1].

FBD працює з послідовністю ланцюгів, кожен з яких містить логічний або арифметичний вираз, виклик функціонального блока, перехід або інструкцію повернення. Типовий приклад програми на FBD показаний на рис. 2.1.

Як і у мові LD робоча область складається з розділу оголошень змінних у верхній частині й розділу програми, яка складається з окремих ланцюгів [2, 3]. Ланцюг може складатися з одного або декількох компонентів у вигляді графічних об'єктів, з'єднаних між собою. Кожен компонент може бути операндом, функцією функціональним блоком, програмою й називається POU (Program Organization Unit – Програмний організатор об'єктів). Вибір об'єктів виконується за допомогою мишки шляхом її установки на відповідну кнопку на панелі інструментів і натискання на праву кнопку. Ланцюги відокремлюються лініями й мають номер, який відображається у її лівій частині.



Рисунок 2.1 – Вікно програми на FBD у середовищі програмування СоDeSys. Діаграма FBD із двох ланцюгів

2.1 Відображення РОU

Діаграма FBD будується з компонентів, відображуваних на схемі прямокутниками. За замовчуванням графічний блок (прямокутник), що вставляється в ланцюг POU завжди логічний елемент AND, який можна перейменовувати й змінювати на блоки, наявні в бібліотеках. Входи РОU зображуються ліворуч від прямокутника, виходи – праворуч. Усередині прямокутника вказується тип РОИ і найменування входів і виходів. Для примірника функціонального блока його найменування вказується зверху, над графічних системах прямокутником. У програмування прямокутник компонента може містити картинку, що відображає його тип. Розмір прямокутника залежить від числа входів і виходів і встановлюється графічним редактором автоматично. Приклад графічного надання примірника Blinker функціонального блока BLINK показаний на рис. 2.2.



Рисунок 2.2 – Графічне надання примірника функціонального блока

Програма у FBD не обов'язково повинна представляти більшу єдину схему. Як і в LD, діаграма утвориться із множини ланцюгів, які виконуються один за одним.

В CoDeSys всі ланцюги одного POU відображаються в єдиному графічному вікні, пронумеровані й розділені горизонтальними лініями (рис. 2.1). Значення змінних, обчислені в одному ланцюзі, доступні в наступних ланцюгах відразу у тому самому робочому циклі.

2.2 З'єднувальні лінії

Прямокутники POU у FBD з'єднані лініями зв'язку. З'єднання мають направленість зліва направо. Вхід блока може бути з'єднаний з виходом блока, розташованого ліворуч від нього. Крім цього, вхід може бути з'єднаний зі змінною або константою. З'єднання повинне зв'язувати змінні або входи й виходи одного типу. На відміну від компонента змінна зображується на діаграмі без прямокутної рамки. Ширина з'єднувальної лінії у FBD не має значення. Стандарт допускає використання з'єднувальних ліній різної ширини й стилю для з'єднань різного типу.

2.3 Порядок виконання FBD

Виконання FBD-ланцюгів йде зліва направо, зверху вниз. Блоки, розташовані ліворуч, виконуються раніше. Блок починає обчислюватися тільки після обчислення значень усіх його входів. Подальші обчислення не будуть продовжені до обчислення значень на всіх виходах. Інакше кажучи, значення на усіх виходах графічного блока з'являються одночасно. Обчислення ланцюга вважається закінченим тільки після обчислення значень на виходах усіх вхідних у неї елементів.

У деяких системах програмування користувач має можливість вільно пересувати блоки зі збереженням зв'язків. У цьому випадку орієнтуватися потрібно виходячи з порядку з'єднань. Редактор FBD CoDeSys автоматично розставляє блоки в порядку виконання.

2.4 Інверсія логічних сигналів

Інверсія логічного сигналу у FBD зображується у вигляді кола на з'єднанні, перед входом або змінною. Інверсія не є властивістю самого блока й може бути легко додана або скасована безпосередньо в діаграмі. У CoDeSys це робиться командою «Negate» (Інверсія).

2.5 З'єднувачі й зворотні зв'язки

З'єднувачі (connectors) являють собою пойменоване з'єднання, яке можна розірвати й перенести в наступний ланцюг. Такий прийом може знадобитися при обмеженій ширині вікна редактора FBD. У CoDeSys ширина вікна не обмежена, тому з'єднувачі тут не потрібні.

Стандарт не забороняє з'єднання, що йдуть із виходу блока на свій вхід або вхід блоків, що виконувались раніше. Зворотний зв'язок не утворить цикл, подібний FOR, просто деяке обчислене значення надійде на вхід при наступному виклику діаграми. Фактично це означає неявне створення змінної, яка зберігає своє значення між викликами діаграми. Для усунення неоднозначності необхідно присвоїти безпечне початкове значення змінного зворотного зв'язку. Але як це зробити для змінної, яка не оголошена в явній формі, незрозуміло?

У редакторі FBD CoDeSys зворотні з'єднання заборонені. Для створення зворотного зв'язку використайте явно оголошену внутрішню змінну. За необхідності перенесення або розгалуження з'єднання в інші ланцюги також необхідно використати проміжні локальні змінні.

2.6 Мітки, переходи й повернення

Порядок виконання FBD-ланцюгів діаграми можна примусово змінювати, використовуючи мітки й переходи, точно так само, як і в релейних схемах.

Мітка ставиться на початку будь-якого ланцюга, будучи, по суті, назвою даного ланцюга. Ланцюг може містити тільки одну мітку. Імена міток підпорядковані загальним правилам найменування ідентифікаторів МЕК. Графічний редактор автоматично нумерує ланцюги діаграми. Ця нумерація застосовується винятково для документування й не може замінювати мітки.

Перехід обов'язково пов'язаний з логічної змінної й виконується, якщо змінна має значення ІСТИНА (TRUE). Для створення безумовного переходу використовується константа ІСТИНА, яка пов'язана з переходом. Мітки й переходи у FBD показані на прикладі, рис. 2.3. Зверніть увагу на останній ланцюг на рис. 2.3 – він є порожнім. Порожній ланцюг позначається єдиною константою TRUE.

Оператор повернення RETURN можна використовувати у FBD так само, як і перехід на мітку, тобто у зв'язуванні з логічною змінною. Повернення приводить до негайного закінчення роботи програмного компонента й повернення на верхній рівень вкладень. Для основної програми це початок робочого циклу ПЛК.

2.7 Вирази ST у FBD

CoDeSys дозволяє записувати вирази ST на вході графічних блоків. Такий прийом розширює стандартний FBD і часто виявляється досить зручним (рис. 2.4). Компактна форма надання виразів полегшує запис і читання функціональних діаграм.



Рисунок 2.3 – Мітки та переходи у FBD



Рисунок 2.4 – Фрагмент FBD-діаграми з виразами

Редактор FBD – графічний редактор [2, 3]. Він працює зі списком ланцюгів, кожен з яких складається з логічних або арифметичних виразів, викликів функцій, програм або функціональних блоків, інструкцій повернення й переходу.

Найбільш важливі функції можна знайти в контекстному меню, що викликається правою кнопкою миші або поєднанням клавіш «Ctrl>+<F10>.

Позиція курсора у FBD

Текстовий курсор може встановлюватися в будь-яку частину FBD ланцюга, що містить текст. Обраний текст виділяється, синім, і може бути змінений. Поточну позицію графічного курсора можна побачити по прямокутнику з пунктирною межею. Далі на прикладі надається список усіх можливих позицій курсора:

1) Будь-яке поле з текстом (обведені чорним):



2) Будь-який вхід блока



3) Будь-який оператор, функція або функціональний блок.



4) Вихід блока, якщо до нього приєднана змінна або інструкція переходу.



5) Перетинання ліній над присвоюванням, переходом або поверненням:



6) Місце за самим правим об'єктом схеми ("остання позиція курсора"; використовується для вибору всієї схеми)



7) Перетинання ліній перед змінною:



Установка позиції курсора

Позицію курсора можна встановити за допомогою лівої кнопки миші або за допомогою клавіатури.

При використанні клавіш переміщення ви будете змінювати поточну позицію курсора на сусідню в заданому напрямку. При використанні цього способу можна вибрати будь-яку позицію курсора, у тому числі й текстове поле. Клавіші нагору й вниз дозволяють вибрати попередню й наступну позиції курсора.

Порожня схема містить тільки три знаки питання "???". Таку схему можна вибрати, клацнувши на них мишею.

"Insert" "Assign"

Позначення: Швидке введення: <Ctrl> + <A>.

Ця команда вставляє інструкцію присвоювання у схему.

Залежно від позиції курсора (див. вище "Позиції курсора в FBD") присвоювання буде вставлено прямо перед обраним входом (позиція курсора

№ 2), перед виходом (позиція курсора № 4) або наприкінці схеми (позиція курсора № 6). Після вставки присвоювання з'являться три знаки питання, виділивши які, можна вводити ім'я змінної. Ім'я змінної зручно вводити за допомогою Асистента введення (Input Assistant) клавіша <F2>.

Зверніть також увагу на можливість введення адрес замість імен змінних (Коментарі до схеми, команда "Extras" 'Options').

Щоб ввести додаткове присвоювання до існуючого, використайте команду "Insert" "Output".

"Insert" "Jump" (Вставити) (Перехід)

Позначення: Швидке введення: <Ctrl>+<L>. Ця команда вставляє інструкцію переходу.

Залежно від позиції курсора (див. вище "Позиції курсора у FBD") інструкція переходу буде вставлена прямо перед обраним входом (позиція курсора № 2), перед виходом (позиція курсора № 4) або наприкінці схеми (позиція курсора № 6).

Після вставки інструкції переходу з'являться три знаки питання, виділивши які, можна вводити ім'я мітки.

"Insert" "Return" (Вставити) (Повернення)

Позначення: Швидке введення: <Ctrl>+<R>.

Ця команда вставляє інструкцію повернення Return.

Залежно від позиції курсора (див. вище "Позиції курсора у FBD") інструкція повернення буде вставлена прямо перед обраним входом (позиція курсора № 2), перед виходом (позиція курсора № 4), перед пересічними лініями із приладу № 5 або наприкінці схеми (позиція курсора № 6).

"Insert" "Вох" (Вставити) (Об'єкт)

Позначення: Швидке введення: <Ctrl>+.

За допомогою цієї команди у схему можна вставляти оператори, функції, функціональні блоки й програми. Відразу після виконання цієї команди у схемі з'являється оператор "AND". Вибравши текстове поле, де написано "AND", цей оператор можна перетворити в будь-який інший об'єкт (функцію, функціональний блок, програму, оператор), написавши ім'я потрібного об'єкта. Це ім'я зручно вибирати, використовуючи Input Assistant (<F2>). Якщо новий блок має інше число входів, ніж оператор AND, то будуть додані нові входи або вилучені непотрібні.

У функціях і функціональних блоках зображуються формальні вхідні й вихідні параметри. Над функціональними блоками перебуває поле, у якому потрібно ввести ім'я примірника функціонального блока. Якщо тип функціонального блока введений некоректно (функціональний блок не описаний), то з'являється блок, що має два входи. Якщо обрано поле введення імені примірника функціонального блока, то за допомогою клавіші <F2> можна викликати Input Assistant.

Новий РОU вставляється в обрану позицію.

Обрано вхід блока (позиція курсора № 2). У цьому випадку РОU вставляється в позицію перед входом. Перший вхід цього РОU з'єднується з гілкою, раніше з'єднаною з обраним входом. Вихід РОU з'єднується з обраним входом.

Обрано вихід (позиція курсора № 4), тоді РОU вставляється після цього виходу. Перший вхід цього РОU з'єднується з обраним виходом. Вихід вставленого РОU з'єднується з гілкою, раніше з'єднаною з обраним виходом.

Обраний РОU (позиція курсора № 3), тоді старий блок буде замінений на новий. Наскільки це можливо, новий блок буде приєднаний до схеми так само, як і старий. Якщо новий елемент має менше входів, ніж старий, то непотрібні гілки будуть вилучені. Так само вірно й для виходів.

Обрано інструкцію переходу або повернення, тоді РОU буде вставлений перед нею. Перший вхід цього РОU з'єднується з гілкою, раніше з'єднаною ліворуч з обраним елементом. Перший вихід цього РОU з'єднується з гілкою, раніше з'єднаною праворуч з обраним елементом.

Обрано останню позицію схеми (позицію курсора № 6). Новий РОU з'єднується з останнім блоком схеми.

Всі входи POU, які не вдалося з'єднати автоматично, з'єднуються з трьома знаками питання. Цей текст можна замінити на ім'я змінної або константу.

Якщо праворуч від вставленого РОU перебуває гілка, то вона буде з'єднана з першим виходом цього РОU.

"Insert" "Input" (Вставити) (Вхід)

Позначення: Якщо вибрати перетинання ліній над присвоюванням (позиція курсора № 5, див. вище "Позиція курсора у FBD") або вихід прямо перед ним (позиція курсора № 4), то після вже існуючого присвоювання буде вставлене нове.

У випадку, якщо лінії перетинаються прямо перед обраним присвоюванням (позиція курсора № 4), то нове присвоювання буде вставлено перед обраним.

Ліворуч від вставленого присвоювання з'явиться рядок "???". Замість нього потрібно ввести ім'я змінної або константу, для чого можна скористатися Input Assistant.

Зверніть також увагу на можливість введення адрес замість імен змінних (Коментарі до схеми, команда "Extras" 'Options'). Швидке введення: <Ctrl>+<U>. Додає вхід оператора. Деякі оператори можуть мати змінне число входів (наприклад, ADD може мати два й більше входи). Для того щоб додати вхід, виберіть уже існуючий вхід (позиція курсора № 1), перед яким необхідно вставити новий і виконуйте команду "Insert" "Input". Є інший спосіб: виберіть оператор (позиція курсора № 3) і виконуйте команду "Insert" "Input", тоді новий вхід буде самим нижнім.

Ліворуч від вставленого входу з'явиться рядок "???". Замість нього потрібно ввести ім'я змінної або константу, для чого можна скористатися Input Assistant.

Зверніть також увагу на можливість введення адрес замість імен змінних (Коментарі до схеми, команда "Extras" 'Options').

"Insert" "Output" (Вставити) (Вихід)

Позначення: Додає нове присвоювання до уже існуючому. Це дозволяє передати одне значення відразу декільком змінним.

Якщо вибрати перетинання ліній над присвоюванням (позиція курсора № 5, див. вище "Позиція курсора у FBD") або вихід прямо перед ним (позиція курсора № 4), то після вже існуючого присвоювання буде вставлене нове.

У випадку, якщо лінії перетинаються прямо перед обраним присвоюванням (позиція курсора № 4), то нове присвоювання буде вставлене перед обраним.

Ліворуч від вставленого присвоювання з'явиться рядок "???". Замість нього потрібно ввести ім'я змінної або константу, для чого можна скористатися Input Assistant.

Зверніть також увагу на можливість введення адрес замість імен змінних (Коментарі до схеми, команда "Extras" 'Options').

"Extras" "Negate" (Доповнити) (Інверсія)

Позначення: Швидке введення: <Ctrl>+<N>.

За допомогою цієї команди можна інвертувати входи, виходи, інструкції переходу або повернення.

Символ заперечення – невелике коло на місці з'єднання.

Якщо обрано вхід (позиція курсора № 2), то цей вхід буде інвертований. Так само вірно й для виходу.

При інвертуванні інструкцій переходу або повернення вони виконуються, якщо гілка, до якої вони приєднані, передає FALSE.

Зняти заперечення можна через повторне заперечення.

"Extras" "Set/Reset"(Доповнити) (Вхід/Вихід)



Set-вихід набуває значення TRUE (Істина), а Reset-вихід – значення FALSE (Неправда), якщо гілка, до якої вони приєднані, передає TRUE. Якщо ця гілка передає FALSE, то змінні зберігають свої значення.

При багаторазовому виконанні цієї команди можна отримати Set-вихід, Reset-вихід і звичайний вихід.

"Extras" "View" (Доповнити) (POU у FBD редакторі)

Використовуючи цю команду, можна використовувати редактор LD або FBD для програмних компонентів (POU) створених у FBD редакторі. Це можливо як в Offline, так і в Online режимах.

Open instance (Відкрити) (Примірник)

Команда аналогічна команді "Project" (Проект) "Open instance" (Відкрити примірник). Вона присутня в контекстному меню (<F2>) і в меню "Extras" (Доповнення), якщо курсор установлений на імені функціонального блока в графічному або текстовому редакторі.

Команди вирізати (Cut), копіювати (Copy), виділити (Paste) і видалити (Delete) у FBD.

Ці команди можна знайти в меню Edit (Правка).

Якщо обрано перетинання ліній (позиція курсора № 5), то присвоювання, інструкції переходу або повернення, розташовані під пересічними лініями, будуть вилучені, вирізані або скопійовані.

Коли обрано POU (позиція курсора № 5), то ці дії будуть виконані над обраним об'єктом і всіма гілками, які з'єднують цей об'єкт зі схемою.

Крім того, гілки, що повністю розташовані перед позицією курсора, будуть вирізані, вилучені або скопійовані.

Скопійовані або вирізані частини схеми перебувають у буфері й можуть бути вставлені в потрібне місце, яке перед цим потрібно вибрати. Можна вибирати входи й виходи.

Якщо POU вставляється із буфера (не забудьте, що в цьому випадку всі з'єднуючі гілки, крім першої, зберігаються у буфері як єдине ціле), перший вхід з'єднується з гілкою перед обраною точкою.

В іншому випадку (з буфера вставляється не POU), гілка, що перебуває перед обраною точкою, повністю замінюється на вміст буфера.

В обох випадках останнім вставляється елемент що, з'єднується з гілкою, розташованою праворуч від обраної точки.

Зауваження. За допомогою вирізання й вставки вирішується наступна проблема: новий оператор вставляється у середину схеми; гілка, що розташована праворуч від оператора, тепер з'єднується з першим входом, але може бути з'єднана з другим. Необхідно вибрати перший вхід і виконати команду "Edit" (Правка) "Cut" (Вирізати). Потім, виділіть другий вхід і виконайте команду "Edit" (Правка) "Paste" (Вставити). Тепер гілка з'єднається з другим входом.

FBD діаграма в режимі Online

У режимі Online у редакторі FBD можна встановлювати точки зупину. Якщо в ланцюзі була встановлена точка зупину, то номер відповідного ланцюга стане синім. Виконання програми зупиняється перед ланцюгом, у який установлена точка зупину. У цьому випадку номер ланцюга стає червоним. Використовуючи команду "Step in" (Крок вперед) або "Step over" (Крок назад), можна послідовно виконувати ланцюги, зупиняючись після кожного.

На екран виводиться поточне значення кожної змінної. Виключення становить той випадок, коли вхід функціонального блока — це вирази. Тоді виводиться тільки значення першої змінної у виразі.

Подвійне клацання мишею по змінній виводить діалогове вікно для введення нового значення змінної. Якщо змінна є логічною, то діалогове вікно не виводиться, а значення змінної просто перемикається. Для запису значення змінних у контролер використовується команда "Online" (Онлайн) "Write values" (Записати значення). Після цього змінні знову стають чорними.

Контроль потоку виконання програми запускається за допомогою команди "Online" (Онлайн) "Display Flow Control" (Відображати потік виконання). Використовуючи цю команду, ви можете переглянути значення, передані по лініях з'єднання.

Якщо лінії з'єднання передають нелогічні значення, то ці значення зображуються в окремих полях. Поля для змінних, які не використовуються, зображуються сірим кольором. Якщо лінія передає значення TRUE (ІСТИНА), то вона зображується синім. Ця команда дозволяє спостерігати за потоком інформації під час виконання програми.

У режимі Online, якщо ви перемістите вказівник миші на змінну, то в підказці з'явиться тип, коментарі й адреса цієї змінної.

3 Контрольні питання

- 1 Який порядок виконання FBD ланцюгів?
- 2 Як оператор AND можна перетворити в будь-який інший об'єкт?
- 3 Де ставиться мітка у FBD ланцюга?

4 Домашнє завдання

4.1На підставі розробленого алгоритму керування ділянки виробничого процесу й обраної викладачем конкретної частини розробити його програмну реалізацію мовою FBD.

4.2Підготувати протокол згідно з розд. 6.

5 Лабораторне завдання

5.1 Запустити середовище розробки CoDeSys. Якщо в організаторі об'єктів автоматично відкривається проект, потрібно закрити його за допомогою меню File → Close (Файл → Закрити) [2, 3].

5.2 Відкрити новий проект за допомогою кнопки New (Створити) на панелі інструментів або меню File → Open (Файл → Відкрити). У вікні, що відкрилося, Target Setting (Настроювання цільової платформи) вибрати None. У новому вікні, що відкрилося, New POU (Новий програмний компонент) вибрати тип POU «Програма» і мова програмування FBD. Натисніть кнопку OK.

5.3 У робочій області, що з'явилася, реалізувати схему, що виконує автоматизацію ділянки технологічного процесу [4, 5]. Для цього за допомогою кнопок на панелі інструментів вставити в перший ланцюг елемент AND (TA). За необхідності змінити ім'я елемента. Входам присвоїти значеннєві назви змінних і вибрати їхній тип. Підключаючи додаткові елементи, зібрати потрібну схему. Для створення нового ланцюга скористайтеся правою кнопкою мишки, або кнопкою Insert («Вставити») у рядку меню. Зберегти проект під своїм ім'ям. Занести до протоколу перелік програм (PRG), які входять у даний проект. Цей перелік відображається в організаторі об'єктів.

5.4 Включити режим емуляції за допомогою меню Online → Simulation Mode («Онлайн – Режим емуляції»).

5.5 Приєднати середовище до емуляції ПЛК за допомогою меню Online > Login («Онлайн» – «Підключення»). Зверніть увагу на те, що під час підключення виконується компіляція проекту.

5.6 Запустити проект у виконання за допомогою меню Online \rightarrow Run («Онлайн» – «Старт»).

5.7 У режимі Online виконати програму за кроками. Для цього підключити контроль потоку виконання програми за допомогою команди "Online" (Онлайн) "Display Flow Control" (Відображати потік виконання) і послідовним натисканням правою кнопкою мишки по кнопці "Старт" пройти по ланцюгах діаграми FBD. При зміні змінних для імітації вхідних впливів (спрацьовування датчика й та ін.) необхідно виконати подвійне клацання правою кнопкою мишки і потім підтверджувати ці значення за допомогою меню Online → Write values (Ctrl+F7, «Онлайн» – «Записати значення»).

5.8 Під час функціонування проекту досліджуйте процес відображення змінних в усіх блоках.

5.9 Виконати п. 4.1 домашнього завдання, зовнішній вигляд FBD схеми занести до протоколу.

Протокол лабораторної роботи "Програмування ПЛК. Вивчення мови функціональних блокових діаграм (FBD)" оформляється в окремому зошиті для лабораторних робіт. Протокол повинен містити назву лабораторної роботи, її мету, відповіді на контрольні питання, хід виконання домашнього й лабораторного завдання, алгоритм програми домашнього завдання у словесному виді й у вигляді блок-схеми, коментарі до програм, які розроблені і налагоджені під час виконання лабораторної роботи, висновки.

7 Перелік рекомендованої літератури

1. Петров И.В. Программируемые контроллеры. Стандартные языки и приемы прикладного проектирования ; под ред. проф. В.П. Дьяконова.-М.:СОЛОН-Пресс, 2004. – 256 с.: ил. – (Серия «Библиотека инженера»).

2. Веб-сторінка фірми Smart Software Solutions Gmb виробника середовища CoDeSys <u>http://www.3s-software.com/</u>.

3. Веб-сторінка ПК "Пролог", підтримка середовища CoDeSys російською мовою <u>http://www.codesys.ru/</u>.

4. Методичні посібники для лабораторних робіт даного курсу.

5. Конспект лекцій з курсу МПіПЗА.

Лабораторная работа № 3

ПРОГРАММИРОВАНИЕ ПЛК. ИЗУЧЕНИЕ ЯЗЫКА ФУНКЦИОНАЛЬНЫХ БЛОКОВЫХ ДИАГРАММ (FBD)

1 Цель работы

Целью работы являются ознакомление с принципами построения систем автоматического управления с применением программированных логических контроллеров, а также приобретение навыков разработки программ для программированных логических контроллеров на языке FBD по стандарту MEK 61131 в среде программирования CoDeSys.

2 Основные положения

FBD (Function Block Diagram) – это графический язык программирования. Диаграмма FBD очень напоминает принципиальную схему электронного устройства на микросхемах (см. рис. 2.1). В отличие от LD «проводники» в FBD могут проводить сигналы (передавать переменные) любого типа (логический, аналоговый, время и т. д.). Шины питания на FBD диаграмме не показываются. Выходы блоков могут быть поданы на входы других блоков либо непосредственно на выходы ПЛК. Сами блоки, представленные на схеме как «черные ящики», могут выполнять любые функции. FBD-схемы очень четко отражают взаимосвязь входов и выходов диаграммы. Если алгоритм изначально хорошо описывается с позиции сигналов, то его FBD-представление всегда получается нагляднее, чем в текстовых языках [1].

FBD работает с последовательностью цепей, каждая из которых содержит логическое или арифметическое выражение, вызов функционального блока, переход или инструкцию возврата. Типичный пример программы на FBD показан на рис. 2.1.

Как и в языке LD рабочая область состоит из раздела объявлений переменных в верхней части и раздела программы, состоящей из отдельных цепей [2, 3]. Цепь может состоять из одного или нескольких компонентов в виде графических объектов, соединенных между собой. Каждый компонент может быть операндом, функцией, функциональным блоком, программой и называется POU (Program Organization Unit – Программный организатор объектов). Выбор объектов выполняется с помощью мышки путем ее установки на соответствующую кнопку на панели инструментов и нажатия на правую кнопку. Цепи отделяются линиями и имеют номер, отображаемый в ее левой части.



Рисунок 2.1 – Окно программы на FBD в среде программирования СоDeSys. Диаграмма FBD из двух цепей

2.1 Отображение компонентов (POU)

Диаграмма FBD строится из компонентов, отображаемых на схеме прямоугольниками. По умолчанию графический блок (прямоугольник), вставляемый в цепь POU всегда логический элемент AND, который можно переименовывать и изменять на блоки, имеющиеся в библиотеках. Входы РОU прямоугольника, изображаются слева ОТ выходы справа. Внутри прямоугольника указывается тип РОU и наименования входов и выходов. Для экземпляра функционального блока его наименование указывается сверху, над прямоугольником. В графических системах программирования прямоугольник компонента может содержать картинку, отражающую его тип. Размер прямоугольника зависит от числа входов и выходов и устанавливается графическим редактором автоматически. Пример графического представления экземпляра Blinker функционального блока BLINK дан на рис. 2.2.

Программа в FBD не обязательно должна представлять большую единую схему. Как и в LD, диаграмма образуется из множества цепей, которые выполняются одна за другой.

В CoDeSys все цепи одного POU отображаются в едином графическом окне, пронумерованные и разделенные горизонтальными линиями (рис. 2.1). Значения переменных, вычисленные в одной цепи, доступны в последующих цепях сразу в том же рабочем цикле.



Рисунок 2.2 – Графическое представление экземпляра функционального блока

2.2 Соединительные линии

Прямоугольники POU в FBD соединены линиями связи. Соединения имеют направленность слева направо. Вход блока может быть соединен с выходом блока, расположенного слева от него. Помимо этого, вход может быть соединен с переменной или константой. Соединение должно связывать переменные или входы и выходы одного типа. В отличие от компонента переменная изображается на диаграмме без прямоугольной рамки. Ширина соединительной линии в FBD роли не играет. Стандарт допускает использование соединительных линий разной ширины и стиля для соединений разного типа.

2.3 Порядок выполнения FBD

Выполнение FBD-цепей идет слева направо, сверху вниз. Блоки, расположенные левее, выполняются раньше. Блок начинает вычисляться только после вычисления значений всех его входов. Дальнейшие вычисления не будут продолжены до вычисления значений на всех выходах. Другими словами, значения на всех выходах графического блока появляются одновременно. Вычисление цепи считается законченным только после вычисления значений на выходах всех входящих в неё элементов.

В некоторых системах программирования пользователь имеет возможность свободно передвигать блоки с сохранением связей. В этом случае ориентироваться нужно исходя из порядка соединений. Редактор FBD CoDeSys автоматически расставляет блоки в порядке выполнения.

2.4 Инверсия логических сигналов

Инверсия логического сигнала в FBD изображается в виде окружности на соединении, перед входом или переменной. Инверсия не является свойством самого блока и может быть легко добавлена или отменена непосредственно в диаграмме. В CoDeSys это делается командой «Negate» (Инверсия).

Соединители (connectors) представляют собой поименованное соединение, которое можно разорвать и перенести в следующую цепь. Такой прием может понадобиться при ограниченной ширине окна редактора FBD. В CoDeSys ширина окна не ограничена, поэтому соединители здесь не нужны.

Стандарт не запрещает соединения, идущие с выхода блока на свой вход или вход ранее исполняемых блоков. Обратная связь не образует цикл, подобный FOR, просто некоторое вычисленное значение поступит на вход при следующем вызове диаграммы. Фактически это означает неявное создание переменной, которая сохраняет свое значение между вызовами диаграммы. Для устранения неоднозначности необходимо присвоить безопасное начальное значение переменной обратной связи. Но как это сделать для переменной, которая не объявлена в явной форме, непонятно?

В редакторе FBD CoDeSys обратные соединения запрещены. Для создания обратной связи используйте явно объявленную внутреннюю переменную. При необходимости переноса или разветвления соединения в другие цепи также необходимо использовать промежуточные локальные переменные.

2.6 Метки, переходы и возврат

Порядок выполнения FBD-цепей диаграммы можно принудительно изменять, используя метки и переходы, точно так же, как и в релейных схемах.

Метка ставится в начале любой цепи, являясь, по сути, названием данной цепи. Цепь может содержать только одну метку. Имена меток подчинены общим правилам наименования идентификаторов МЭК. Графический редактор автоматически нумерует цепи диаграммы. Эта нумерация применяется исключительно для документирования и не может заменять метки.

Переход обязательно связан с логической переменной и выполняется, если переменная имеет значение ИСТИНА (TRUE). Для создания безусловного перехода используется константа ИСТИНА, связанная с переходом. Метки и переходы в FBD представлены в примере, показанном на рис. 2.3. Обратите внимание на последнюю цепь на рис. 2.3 – она является пустой. Пустая цепь обозначается единственной константой TRUE.

Оператор возврата RETURN можно использовать в FBD так же, как и переход на метку, т. е. в связке с логической переменной. Возврат приводит к немедленному окончанию работы программного компонента и возврату на верхний уровень вложений. Для основной программы это начало рабочего цикла ПЛК.



Рисунок 2.3 – Метки и переходы в FBD

2.7 Выражения ST в FBD

CoDeSys позволяет записывать выражения ST на входе графических блоков. Такой прием расширяет стандартный FBD и часто оказывается достаточно удобным (рис. 2.4). Компактная форма представления выражений облегчает запись и чтение функциональных диаграмм.



Рисунок 2.4 – Фрагмент FBD-диаграммы с выражениями

2.8 Редактор FBD

Редактор FBD – графический редактор [2, 3]. Он работает со списком цепей, каждая из которых состоит из логических или арифметических выражений, вызовов функций, программ или функциональных блоков, инструкций возврата и перехода.

Наиболее важные функции можно найти в контекстном меню, которое вызывается правой кнопкой мыши или сочетанием клавиш <Ctrl>+<F10>.

Позиция курсора в FBD

Текстовый курсор может устанавливаться в любую часть FBD цепи, содержащую текст. Выбранный текст выделяется, синим, и может быть изменен. Текущую позицию графического курсора можно увидеть по прямоугольнику с пунктирной границей. Далее на примере приводится список всех возможных позиций курсора:

1) Любое поле с текстом (обведены черным)



2) Любой вход блока



3) Любой оператор, функция или функциональный блок



4) Выход блока, если к нему присоединена переменная или инструкция перехода



5) Пересечение линий над присваиванием, переходом или возвратом



6) Место за самым правым объектом схемы ("последняя позиция курсора"; используется для выбора всей схемы)



7) Пересечение линий перед переменной



Установка позиции курсора

Позицию курсора можно установить с помощью левой кнопки мыши или с помощью клавиатуры.

При использовании клавиш перемещения вы будете менять текущую позицию курсора на соседнюю в заданном направлении. При использовании этого способа можно выбрать любую позицию курсора, в том числе и текстовое поле. Клавиши вверх и вниз позволяют выбрать предыдущую и следующую позицию курсора.

Пустая схема содержит только три знака вопроса "???". Такую схему можно выбрать, щелкнув на них мышью.

"Insert" "Assign" (Вставить) (Установить)

Обозначение: Быстрый ввод: <Ctrl> + <A>.

Эта команда вставляет инструкцию присваивания в схему.

В зависимости от позиции курсора (см. выше "Позиции курсора в FBD") присваивание будет вставлено прямо перед выбранным входом (позиция курсора № 2), перед выходом (позиция курсора № 4) или в конце схемы (позиция курсора № 6). После вставки присваивания появятся три знака вопроса, выделив которые, можно вводить имя переменной. Имя переменной удобно вводить с помощью Ассистента ввода (Input Assistant) клавиша <F2>.

Обратите также внимание на возможность ввода адресов вместо имен переменных (Комментарии к схеме, команда "Extras" 'Options').

Чтобы ввести дополнительное присваивание существующему, К используйте команду "Insert" "Output" (Вставить) (Выход).

"Insert" "Јитр" (Вставить) (Переход)

Обозначение: Быстрый ввод:<Ctrl>+<L>.

Эта команда вставляет инструкцию перехода.

В зависимости от позиции курсора (см. выше "Позиции курсора в FBD") инструкция перехода будет вставлена прямо перед выбранным входом (позиция курсора № 2), перед выходом (позиция курсора № 4) или в конце схемы (позиция курсора № 6).

После вставки инструкции перехода появятся три знака вопроса, выделив которые, можно вводить имя метки.

"Insert" "Return" (Вставить) (Возврат)



Обозначение: Быстрый ввод:<Ctrl>+<R>.

Эта команда вставляет инструкцию возврата Return.

В зависимости от позиции курсора (см. выше "Позиции курсора в FBD") инструкция возврата будет вставлена прямо перед выбранным входом (позиция курсора № 2), перед выходом (позиция курсора № 4), перед

пересекающимися линиями из прибора №5 или в конце схемы (позиция курсора № 6).

"Insert" "Вох" (Вставить) (Объект)

Обозначение: Быстрый ввод:<Ctrl>+.

С помощью этой команды в схему можно вставлять операторы, функции, функциональные блоки и программы. Сразу после выполнения этой команды в схеме появляется оператор "AND". Выбрав текстовое поле, где написано "AND", этот оператор можно превратить в любой другой объект (функцию, функциональный блок, программу, оператор), написав имя нужного объекта. Это имя удобно выбирать, используя Input Assistant (<F2>). Если новый блок имеет другое число входов, чем оператор AND, то будут добавлены новые входы или удалены ненужные.

В функциях и функциональных блоках изображаются формальные входные и выходные параметры. Над функциональными блоками находится поле, в котором нужно ввести имя экземпляра функционального блока. Если тип функционального блока введен некорректно (функциональный блок не описан), то появляется блок, имеющий два входа. Если выбрано поле ввода имени экземпляра функционального блока, то с помощью клавиши <F2> можно вызвать Input Assistant.

Новый РОИ вставляется в выбранную позицию.

Выбран вход блока (позиция курсора № 2). В этом случае РОU вставляется в позицию перед входом. Первый вход этого РОU соединяется с ветвью, ранее соединенной с выбранным входом. Выход РОU соединяется с выбранным входом.

Выбран выход (позиция курсора № 4), тогда РОU вставляется после этого выхода. Первый вход этого РОU соединяется с выбранным выходом. Выход вставленного РОU соединяется с ветвью, ранее соединенной с выбранным выходом.

Выбран РОU (позиция курсора № 3), тогда старый блок будет заменен на новый. Насколько это возможно, новый блок будет присоединен к схеме так же, как и старый. Если новый элемент имеет меньше входов, чем старый, то ненужные ветви будут удалены. То же верно и для выходов.

Выбрана инструкция перехода или возврата, тогда POU будет вставлен перед ней. Первый вход этого POU соединяется с ветвью, ранее соединенной слева с выбранным элементом. Первый выход этого POU соединяется с ветвью, ранее соединенной справа с выбранным элементом.

Выбрана последняя позиция схемы (позиция курсора № 6). Новый РОU соединяется с последним блоком схемы.

Все входы POU, которые не удалось соединить автоматически, соединяются с тремя знаками вопроса. Этот текст можно заменить на имя переменной или константу.

Если справа от вставленного POU находится ветвь, то она будет соединена с первым выходом этого POU.

"Insert" "Input" (Вставить) (Вход)

В случае, если линии пересекаются прямо перед выбранным присваиванием (позиция курсора № 4), то новое присваивание будет вставлено перед выбранным.

Слева от вставленного присваивания появится строка "???". Вместо нее нужно ввести имя переменной или константу, для чего можно воспользоваться Input Assistant.

Обратите также внимание на возможность ввода адресов вместо имен переменных (Комментарии к схеме, команда "Extras" 'Options'). Быстрый ввод: <Ctrl>+<U>.

Добавляет вход оператора. Некоторые операторы могут иметь переменное число входов (например, ADD может иметь два и более входа). Для того чтобы добавить вход, выберите уже существующий вход (позиция курсора № 1), перед которым необходимо вставить новый и выполните команду "Insert" "Input". Есть другой способ: выберите оператор (позиция курсора № 3) и выполните команду "Insert" "Input" (Вставить) (Вход), тогда новый вход будет самым нижним.

Слева от вставленного входа появится строка "???". Вместо нее нужно ввести имя переменной или константу, для чего можно воспользоваться Input Assistant.

Обратите также внимание на возможность ввода адресов вместо имен переменных (Комментарии к схеме, команда "Extras" 'Options').

"Insert" "Output" (Вставить) (Выход)

Обозначение: Добавляет новое присваивание к уже существующему. Это позволяет передать одно значение сразу нескольким переменным.

Если выбрать пересечение линий над присваиванием (позиция курсора № 5, см. выше "Позиция курсора в FBD") или выход прямо перед ним (позиция курсора № 4), то после уже существующего присваивания будет вставлено новое.

В случае, если линии пересекаются прямо перед выбранным присваиванием (позиция курсора № 4), то новое присваивание будет вставлено перед выбранным.

Слева от вставленного присваивания появится строка "???". Вместо нее нужно ввести имя переменной или константу, для чего можно воспользоваться Input Assistant.

Обратите также внимание на возможность ввода адресов вместо имен переменных (Комментарии к схеме, команда "Extras" 'Options').

"Extras" "Negate" (Дополнить) (Инверсия))

Обозначение: Быстрый ввод:<Ctrl>+<N>.

С помощью этой команды можно инвертировать входы, выходы, инструкции перехода или возврата.

Символ отрицания – небольшая окружность на месте соединения.

Если выбран вход (позиция курсора № 2), то этот вход будет инвертирован.

Тоже верно и для выхода.

При инвертировании инструкций перехода или возврата они выполняются, если ветвь, к которой они присоединены, передает FALSE.

Снять отрицание можно через повторное отрицание.

"Extras" "Set/Reset" (Дополнить) (Входы/Выходы)

Обозначение: При помощи этой команды вы можете определить Set- и Reset-выходы. Set-выход обозначается буквой S, а Reset-выход – буквой R.

Set/Reset выходы в FBD



Set-выход принимает значение TRUE (Истина), а Reset-выход – значение FALSE (Ложь), если ветвь, к которой они присоединены, передает TRUE. Если эта ветвь передает FALSE, то переменные сохраняют свои значения.

При многократном выполнении этой команды можно получить Setвыход, Reset-выход и обычный выход.

"Extras" "View" (Дополнить) (РОИ в FBD редакторе)

Используя эту команду, можно использовать редактор LD или FBD для программных компонентов (POU) созданных в FBD редакторе. Это возможно как в Offline, так и в Online режимах.

Open instance (Открыть экземпляр)

Команда аналогична команде "Project" (Проект) "Open instance" (Открыть экземпляр). Она присутствует в контекстном меню (<F2>) и в меню "Extras" (Дополнения), если курсор установлен на имени функционального блока в графическом или текстовом редакторе.

Команды вырезать (Cut), копировать (Copy), выделить (Paste) и удалить (Delete) в FBD.

Эти команды можно найти в меню Edit (Правка).

Если выбрано пересечение линий (позиция курсора № 5), то присваивания, инструкции перехода или возврата, расположенные под пересекающимися линиями, будут удалены, вырезаны или скопированы.

Когда выбрано POU (позиция курсора № 5), то эти действия будут выполнены над выбранным объектом и всеми ветвями, которые соединяют этот объект со схемой.

Кроме того, ветви, полностью расположенные перед позицией курсора, будут вырезаны, удалены или скопированы.

Скопированные или вырезанные части схемы находятся в буфере и могут быть вставлены в нужное место, которое перед этим нужно выбрать. Можно выбирать входы и выходы.

Если POU вставляется из буфера (не забудьте, что в этом случае все соединяющие ветви, кроме первой, хранятся в буфере как единое целое), первый вход соединяется с ветвью перед выбранной точкой.

В другом случае (из буфера вставляется не POU), ветвь, находящаяся перед выбранной точкой, полностью заменяется на содержимое буфера.

В обоих случаях последний вставляемый элемент соединяется с ветвью, расположенной справа от выбранной точки.

Замечание. С помощью вырезания и вставки решается следующая проблема: новый оператор вставляется в середину схемы; ветвь, расположенная справа от оператора, теперь соединяется с первым входом, но может быть соединена со вторым. Для этого необходимо выбрать первый вход и выполнить команду "Edit" (Правка) "Cut" (Вырезать). Затем, выделить второй вход и выполнить команду "Edit" (Правка) "Paste" (Вставить). Теперь ветвь соединится со вторым входом.

FBD диаграмма в режиме Online

В режиме Online в редакторе FBD можно устанавливать точки останова. Если в цепи была установлена точка останова, то номер соответствующей цепи станет синим. Выполнение программы останавливается перед цепью, в которой установлена точка останова. В этом случае номер цепи становится красным. Используя команду "Step in" (Шаг вверх) или "Step over" (Шаг вниз), можно последовательно выполнять цепи, останавливаясь после каждой.

На экран выводится текущее значение каждой переменной. Исключение составляет тот случай, когда вход функционального блока – это выражение. Тогда выводится только значение первой переменной в выражении.

Двойной щелчок мышью по переменной выводит диалоговое окно для ввода нового значения переменной. Если переменная является логической, то диалоговое окно не выводится, а значение переменной просто переключается. Для записи значения переменных в контроллер (точнее в программу) используется команда "Online" (Онлайн) "Write values" (Записать значения). После этого переменные снова становятся черными.

Контроль потока выполнения программы запускается с помощью команды "Online" (Онлайн) "Display Flow Control" (Отображать поток выполнения). Используя эту команду, можно просмотреть значения, передаваемые по линиям соединения.

Если линии соединения передают нелогические значения, то эти значения изображаются в отдельных полях. Поля для переменных, которые не используются, изображаются серым цветом. Если линия передает значение TRUE (Истина), то она изображается синим. Эта команда позволяет наблюдать за потоком информации во время выполнения программы.

В режиме Online, если вы переместите указатель мыши на переменную, то в подсказке появится тип, комментарии и адрес этой переменной.

3 Контрольные вопросы

- 1 Какой порядок выполнения FBD цепей?
- 2 Как оператор AND можно превратить в любой другой объект?
- 3 Где ставится метка в FBD цепи?

4 Домашнее задание

4.1 На основании разработанного алгоритма управления участка производственного процесса и выбранной преподавателем конкретной части разработать его программную реализацию на языке FBD.

4.2 Подготовить протокол согласно разд. 6.

5 Лабораторное задание

5.1Запустить среду разработки CoDeSys. Если в организаторе объектов автоматически приоткрывается проект, нужно закрыть его с помощью меню File \rightarrow Close (Файл \rightarrow Закрыть) [2, 3].

5.2Открыть новый проект с помощью кнопки New на панели инструментов или меню File → Open (Файл – Открыть). В открывшемся окне Target Setting (Настройки целевой платформы) выберите None. В новом

открывшемся окне New POU (Новый программный компонент) выбрать тип POU «Программа» и язык программирования FBD. Нажмите кнопку OK.

5.3В появившейся рабочей области реализовать схему, выполняющую автоматизацию участка технологического процесса [4, 5]. Для этого с помощью кнопок на панели инструментов вставить в первую цепь элемент AND (И). При необходимости изменить имя элемента. Входам присвоить смысловые названия переменных и выбрать их тип. Подключая дополнительные элементы, собрать требуемую схему. Для создания новой цепи воспользуйтесь правой кнопкой мышки, либо кнопкой Insert («Вставить») в строке меню. Сохранить проект под своим именем. Занести в протокол перечень программ (PRG), которые входят в данный проект. Этот перечень отображается в организаторе объектов.

5.4Включить режим эмуляции с помощью меню Online → Simulation Mode («Онлайн – Режим эмуляции»).

5.5Подсоединить среду к эмуляции ПЛК с помощью меню Online → Login («Онлайн» – «Подключение»). Обратите внимание на то, что во время подключения выполняется компиляция проекта.

5.63апустить проект во исполнение с помощью меню Online \rightarrow Run («Онлайн» – «Старт»).

5.7В режиме Online выполнить программу по шагам. Для этого подключить контроль потока выполнения программы с помощью команды "Online" («Онлайн») "Display Flow Control" («Отображать поток выполнения») и последовательным нажатием правой кнопкой мышки по значку «Run» ("Старт") пройти по цепям диаграммы FBD. При изменении переменных для имитации входных воздействий (срабатывание датчика и т.п.) необходимо выполнить двойной щелчок правой кнопкой мышки и затем подтверждать эти значения с помощью меню Online → Write values (Ctrl+F7, «Онлайн» – «Записать значения»).

5.8Во время функционирования проекта исследуйте процесс отображения переменных во всех блоках.

5.9Выполнить п. 4.1 домашнего задания, внешний вид FBD схемы занести в протокол.

6 Содержание протокола

Протокол лабораторной работы "Программирование ПЛК. Изучение языка функциональных блочных диаграмм (FBD)" оформляется в отдельной тетради для лабораторных работ. Протокол должен содержать название лабораторной работы и ее цель, ответы на контрольные вопросы, ход выполнения домашней и лабораторной задачи, алгоритм программы домашнего задания в словесном виде и в виде блок-схемы, комментарии к программам, разработанным и налаженным во время выполнения лабораторной работы, выводы.

7 Перечень рекомендованной литературы

1. Петров И.В. "Программируемые контроллеры. Стандартные языки и приемы прикладного проектирования"; под ред. проф. В.П. Дьяконова.-М.:СОЛОН-Пресс, 2004. – 256 с.: ил. – (Серия «Библиотека инженера»).

2. Веб-страница фирмы Smart Software Solutions Gmb производителя среды CoDeSys <u>http://www.3s-software.com/</u>.

3. Веб-страница ПК "Пролог", поддержка среды CoDeSys на русском языке <u>http://www.codesys.ru/</u> Руководство пользователя по программированию ПЛК в CoDeSys 2.3.

4. Методические пособия для лабораторных работ данного курса.

5. Конспект лекций из курса МПиПСА.