

**Міністерство освіти і науки України
Донбаська державна машинобудівна академія**

**Методичні вказівки
до самостійної роботи
з дисципліни “Мікропроцесорні системи
керування”**

(для спеціальності 7.092203 “Електромеханічні системи автоматизації
та електропривід”)

для студентів всіх форм навчання

Перезатверджено
на засіданні кафедри ЕСА
Протокол № 1 від 21.08.2012 р.

Краматорськ 2012

ЗМІСТ

ВСТУП.....	4
1 КОНТРОЛЕР LOGO!.....	5
1.1 Знайомство з LOGO!.....	5
1.2 Програмування LOGO!.....	8
1.2.1 Елементи з'єднання.....	8
1.2.2 Блоки й номери блоків.....	9
1.2.3 Від комутаційної схеми до LOGO!.....	10
1.2.4 Робота з LOGO!.....	11
1.2.5 Уведення й запуск комутаційної програми.....	12
1.3 Програмування LOGO! за допомогою LOGO!Soft Comfort.....	14
1.3.1 Створення програм.....	15
1.3.2 Симуляція комутаційної програми.....	17
1.4 Завдання для самостійної роботи.....	19
2 ЛОГІЧНІ КОНТРОЛЕРИ SIMATIC S7-200.....	37
2.1 Знайомство з SIMATIC S7-200 CPU 221.....	37
2.2 Програмування S7-200.....	41
2.2.1 STEP 7-Micro/WIN	41
2.2.2 Приклад програмування.....	49
2.3 Завдання для самостійної роботи.....	53
Додаток А.....	56
Додаток Б.....	61

ВСТУП

Для роботи машин, механізмів і процесів майже на всіх виробничих ділянках поряд з подачею енергії потрібні й елементи регулювання. У будь-якій установці або машині потрібно мати можливість починати, впливати, спостерігати й закінчувати виробничий процес.

У минулому для звичайного програмувального регулювання програмна логіка визначалася специфічним з'єднанням ключів і реле.

Сьогодні системи програмного керування застосовуються для рішення завдань автоматизації. Логіка, вкладена в програмну пам'ять системи автоматизації, не залежить від структури пристрою й з'єднань і може бути в будь-який час змінена за допомогою програматора. Виробничі процеси розглядаються не як окремі частини, а як інтегральні компоненти одного загального виробничого процесу.

Повна інтеграція автоматизації досягається сьогодні за допомогою:

- комплексного проектування й програмування окремих частин системи;
- комплексного збереження даних;
- комплексної комунікації між всіма компонентами, що беруть участь у автоматизації.

SIMATIC – це базисна система автоматизації фірми **SIEMENS** для рішення завдань автоматизації у всіх галузях, що складається зі стандартних компонентів **Hard-** і **Software**, і залишає за собою безліч можливостей для розширення залежно від вимог поставленні задачі

Основам програмування малих модульних систем керування **LO GO!** та **S7-200** для нижчого рівня систем автоматизації присвячені наступні методичні вказівки.

1 КОНТРОЛЕР LOGO!

З 1996 року універсальний логічний модуль **LOGO!** фірми **Siemens** перебуває на міжнародному ринку й із цього часу вже завоював всесвітню

популярність і дуже полюбився широкому колу користувачів. Завдяки великій кількості убудованих функцій і невисокої вартості, у порівнянні зі стандартною комутаційною технікою, LOGO! знаходить застосування при рішенні завдань автоматизації в побутовій техніці, апарато- і машинобудуванні.

1.1 Знайомство з LOGO!

LOGO! – це універсальний логічний модуль фірми Siemens.

LOGO! містить у собі:

- пристрій керування;
- панель керування й індикації з фоновим підсвічуванням;
- блок живлення;
- інтерфейс для модулів розширення;
- інтерфейс для програмного модуля (плата) і кабель РС;
- готові стандартні функції, часто використовувані на практиці, наприклад, функції затримки включення й вимикання, імпульсне реле й програмувальна клавіша;
- годинний вимикач;
- цифрові й аналогові прапори;
- входи й виходи відповідно до типу пристрою.

LOGO! пропонує рішення різних технічних завдань, у тому числі в електроустаткуванні житлових приміщень (наприклад, висвітлення сходових кліток, зовнішнє висвітлення, штори, жалюзі, висвітлення вітрин магазинів і т.д.), у комутаційних шафах, у керуванні машинами й апаратами (наприклад, системи керування воротами, вентиляційні системи або насоси для господарської води й багато чого іншого).

LOGO! можна використати також для спеціальних систем керування в оранжереях і теплицях, для попередньої обробки сигналів керування й, при підключенні комунікаційного модуля (напр., ASi), для децентралізованого керування машинами й процесами на місці. Є спеціальні варіанти без панелі керування й індикації для серійних систем керування у мікромашинобудуванні, апаратобудуванні й шафах керування.

LOGO! Basic є для двох класів напруги:

- клас 1 < 24 V, тобто 12 V постійного струму, 24 V постійного струму, 24 V перемінного струму;
- клас 2 > 24, V тобто 115...240 V постійного і перемінного струму у варіантах:
 - з дисплеєм: 8 входів і 4 виходи;
 - без дисплея: 8 входів і 4 виходи.

Кожний варіант може складатися з 4 субблоків, оснащений інтерфейсом для підключення розширення й надає у ваше

розпорядження 33 готових до використання основних і спеціальних функції для розробки програми комутації.

Існують наступні модулі розширення:

- цифрові модулі LOGO! є для напруг 12 V постійного струму, 24 V постійного струму, 24 V перемінного струму, 115...240 V постійного і перемінного струму з 4 входами й 4 виходами;
- аналогові модулі LOGO! є для напруг 12 і 24 V постійного струму із двома аналоговими входами;
- комунікаційні модулі LOGO! (CM).

LOGO! забезпечує рішення в діапазоні від невеликих домашніх установок і малих завдань автоматизації до складних інженерних завдань, що включають вбудовування системи шин.

Кожний LOGO! Basic, незалежно від кількості підключених модулів, надає у ваше розпорядження наступні входи, виходи й прапори для створення комутаційної програми:

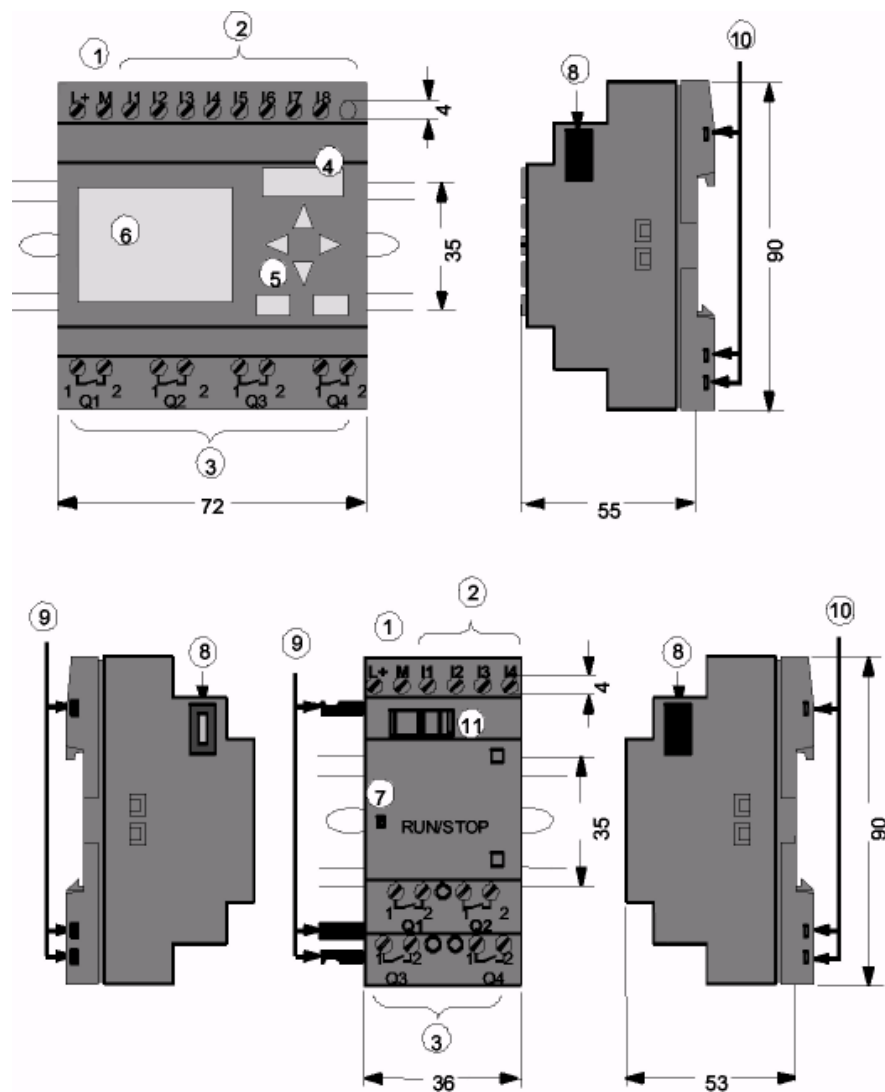
- цифрові входи I1 – I24;
- аналогові входи AI1 – AI8;
- цифрові виходи Q1 – Q16;
- аналогові входи AQ1 і AQ2;
- цифрові прапори M1 – M24, M8 - прапор запуску;
- аналогові прапори AM1 - AM6;
- біти регістрів зрушення S1-S8;
- 4 клавіші керування курсором;
- 16 непідключених виходів X1 - X16.

Конструкція LOGO! Показана на рисунку 1.1.

Кожний базовий пристрій LOGO! Basic може бути розширений тільки за допомогою модулів розширення того ж класу напруги. Підключенню друг до друга пристроїв, що відносяться до різних класів напруги, перешкоджає механічне кодування (штифти в корпусі). **Виключення:** лівий інтерфейс аналогового або комунікаційного модуля має потенційну розв'язку. Тому ці модулі розширення можуть з'єднуватися із пристроями іншого класу напруги.

Позначення LOGO! містить інформацію про різні його характеристики:

- 12: варіант на 12 V постійного струму;
- 24: варіант на 24 V постійного струму;
- 230: варіант на 115/240 V змінного струму;
- R: релейні виходи (без R: транзисторні виходи);
- C: убудований часовий вимикач на 7 днів;
- O: варіант без дисплея;
- DM: цифровий модуль;
- AM: аналоговий модуль;
- CM: комунікаційний модуль.



- 1- підключення джерела живлення;
- 2- входи;
- 3- виходи;
- 4- гніздо для модуля із кришкою;
- 5- панель керування;
- 6- ЖКІ;
- 7- індикатор стану;
- 8- інтерфейс розширення;
- 9- механічне кодування – штифти;
- 10- механічне кодування – гнізда;
- 11- защіпка

Рисунок 1.1 – Основні елементи LOGO!

1.2 Програмування LOGO!

Під програмуванням ми маємо на увазі створення комутаційної програми. Комутаційна програма LOGO! є насправді нічим іншим, як комутаційною схемою, поданою в трохи зміненій формі!

1.2.1 Елементи з'єднання

У LOGO! є входи й виходи. Приклад конфігурації з декількома модулями показано на рисунку 1.2.

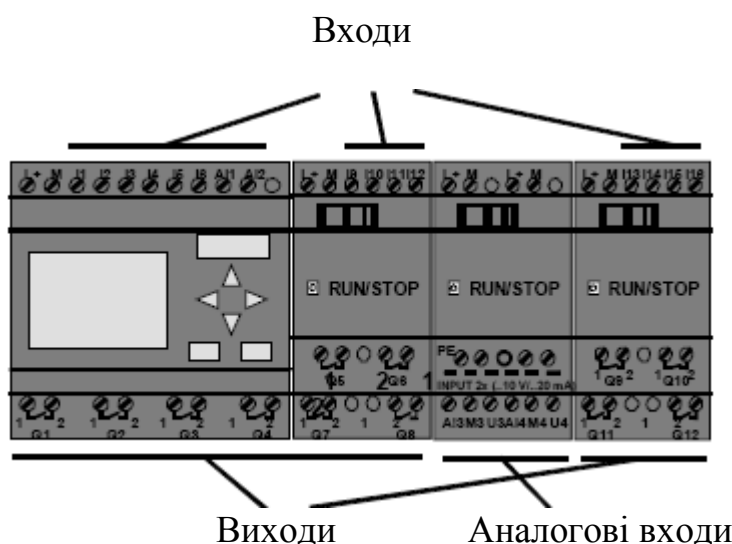


Рисунок 1.2 – Розташування входів та виходів на модулях LOGO!

Кожний вхід позначається буквою I і номером. Коли ви дивитесь на LOGO! спереду, та зверху ви бачите клеми для входів. Тільки в аналогових модулях LOGO! AM 2 і AM 2 PT100 входи перебувають знизу. Кожний вихід позначається буквою Q і номером. Клеми для виходів видні на рисунку 1.2 знизу. LOGO! може розпізнавати, зчитувати й включати входи й виходи всіх модулів розширення незалежно від їхнього типу. Входи й виходи подані в тій же послідовності, у якій розташовані модулі.

Термін «елемент з'єднання» відноситься до всіх з'єднань й станів в LOGO!. Входи й виходи можуть мати стан '0' або '1'. '0' означає, що на вході немає напруги; '1' означає, що вона є. Щоб полегшити створення комутаційної програми, ми введемо і елементи з'єднання hi, lo і x:

- 'hi' (high = високий) має фіксований стан '1',
- а 'lo' (low = низький) має фіксований стан '0'.

Ви не зобов'язані використати всі елементи з'єднання блоку. Комутаційна програма автоматично привласнює не використовуваним елементам з'єднання стан, що забезпечує належне функціонування відповідного блоку. Якщо хочете, ви можете позначити невикористані сполучні елементи символом 'x'.

1.2.2 Блоки й номери блоків

Блок в LOGO! – це функція, що перетворить вхідну інформацію на вихідну інформацію. Раніше необхідно підключати окремі елементи в шафі керування або в клемній коробці. При створенні комутаційної програми ви зв'яжете елементи з'єднання із блоками. Для цього ви просто вибираєте бажане з'єднання з меню С. Для позначення меню використано ім'я С, що є скороченням англійського терміна «Connector», тобто з'єднуючий елемент.

Найпростішими блоками є логічні операції:

- AND [І]
- OR [АБО] (показаний на рисунку 1.3)

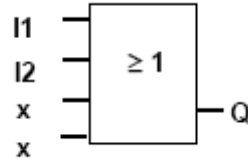


Рисунок 1.3 –Блок OR [АБО]

Входи I1 та I2 підключені тут до блоку OR [АБО]. Останні два входи блоку не використовуються й позначені розроблювачем комутаційної програми символом x.

Значно могутнішими є спеціальні функції:

- імпульсне реле;
- реверсивний лічильник;
- затримка включення;
- програмний вимикач.

Повний список функцій LOGO! поданий у додатку А.

На рисунку 1.4 показаний типовий вид дисплея LOGO!. Як видно з рисунка 1.4, у кожний момент часу може бути поданий тільки один блок. Тому введені номери блоків, щоб допомогти контролювати структуру схеми.

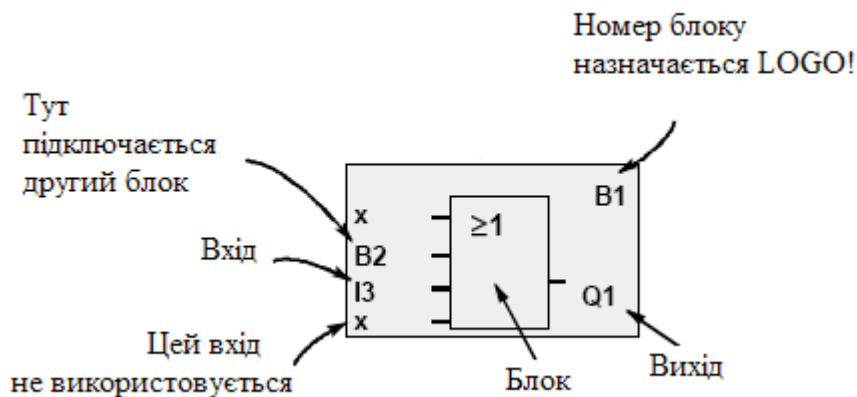


Рисунок 1.4 – Вид дисплея LOGO!

Щораз, коли вставляється в програму блок, LOGO! призначає цьому блоку номер. LOGO! використовує номери блоків для відображення зв'язків між блоками. Тому номери блоків служать, насамперед, для того, щоб полегшити ваше орієнтування в програмі. На рисунку 1.5 показані три покази дисплея LOGO!, які разом утворюють комутаційну програму.

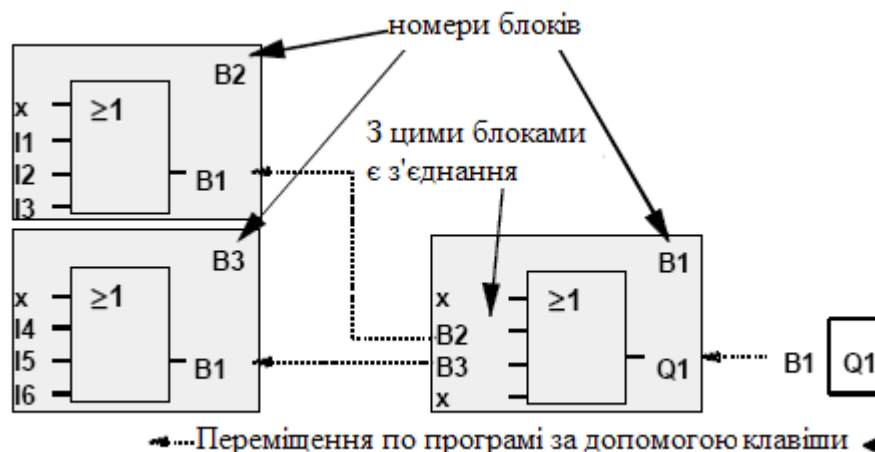


Рисунок 1.5 – Фрагмент програми

Як видно з рисунка 1.5, LOGO! зв'язує блоки один з одним за допомогою їхніх номерів. Можливо з'єднання майже будь-якого блоку із входом поточного блоку, використовуючи номер. Таким чином, ви можете багаторазово використати проміжні результати логічних або інших операцій. Це заощаджує зусилля, необхідні для повторного введення елементів, а також місце в пам'яті й забезпечує збереження ясності схеми. У цьому випадку необхідно знати, які імена LOGO! привласнив блокам.

Для підвищення ефективності роботи, рекомендується скласти оглядову блок-схему програми. Цим ви полегшите собі створення комутаційної програми. Потім ви зможете ввести до цієї блок-схеми номери блоків, призначені LOGO!.

1.2.3 Від комутаційної схеми до LOGO!

Схема прикладу показана на рисунку 1.6.

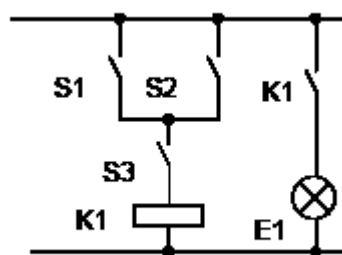


Рисунок 1.6 – Вихідна схема прикладу

Споживач E1 включається й вимикається за допомогою вимикачів (S1 OR S2) AND S3. Реле K1 спрацьовує, коли замкнуті S1 або S2, а також S3.

У LOGO! схема створюється з'єднанням один з одним блоків і елементів з'єднання. На рисунку 1.7 показано підключення модуля LOGO! та програма прикладу.

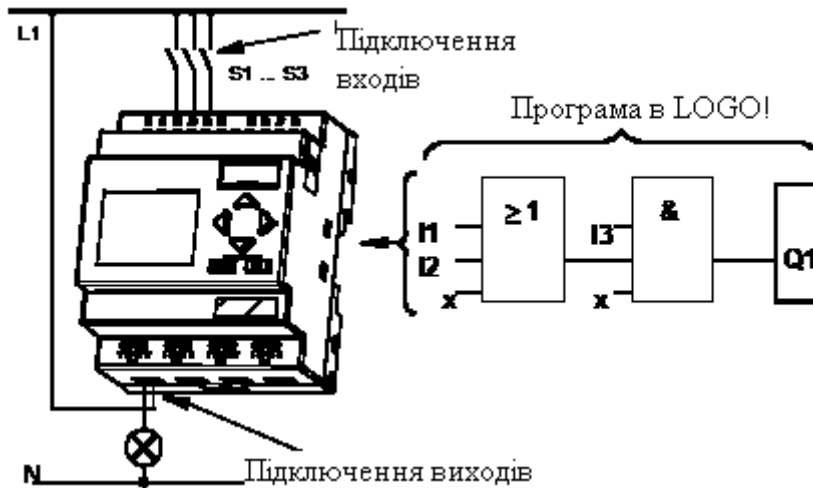


Рисунок 1.7 – Реалізація прикладу

Для реалізації схеми в LOGO! починають із виходу схеми. Виходом є навантаження або реле, яким необхідно управляти. Схема перетвориться на блоки. Для цього пройдіть за схемою від виходу до входу:

Крок 1: На виході Q1 є послідовне з'єднання нормально відкритого контакту S3 з іншим компонентом схеми. Послідовне з'єднання відповідає блоку AND [I].

Крок 2: S1 і S2 з'єднані паралельно. Паралельному з'єднанню відповідає блок OR [АБО].

1.2.4 Робота з LOGO!

Зміна режиму роботи

- Комутаційна програма створюється в режимі **програмування**. Після включення живлення, якщо відображається «No Program/ Press ESC [Немає програми/ Натисніть ESC]», ви попадаєте в режим програмування натисканням клавіші **ESC**.

- Значення часу й параметрів існуючої програми можна змінювати в режимах

параметризації й програмування. Під час параметризації LOGO! перебуває в режимі **RUN**, тобто він продовжує виконання комутаційної програми. Для роботи в режимі **програмування** необхідно завершити виконання комутаційної програми, викликавши команду **“Stop”**.

- Перехід у режим **RUN** відбувається виконанням команди “**Start**” у головному меню.
- Якщо система перебуває в **RUN**, то ви можете повернутися в режим **параметризації**, нажавши клавішу **ESC**.
- Якщо LOGO! перебуває в режимі **параметризації** й необхідно повернутися до режиму **програмування**, то необхідно виконати команду “**Stop**” у меню параметризації. З появою на екрані підказки “**Stop Prg**” перемістіть курсор на “**Yes**” [Так] і підтвердіть команду, нажавши **OK**.
Додаткову інформацію про режими роботи ви знайдете в додатку В.

Увага

Наступні твердження дійсні для пристроїв версії 0BA2 і раніше:

• Ви потрапляєте до режиму програмування, натиснувши одночасно три клавіші **◀, ▶** і **OK**.

• Ви потрапляєте до режиму параметризації, нажавши одночасно дві клавіші: **ESC** і **OK**.

Виходи й входи

- Комутаційна програма завжди вводиться від виходу до входу. Можна з'єднати вихід з декількома входами, але не кілька виходів з одним входом.
- Не можна з'єднувати вихід з попереднім входом у межах одного програмного шляху. Для утворення таких внутрішніх зворотних зв'язків (рекурсій) включайте проміжні прапори або виходи.

Курсор і його переміщення

При уведенні комутаційної програми:

- Коли курсор з'являється у формі знака підкреслення, його можна **переміщати**: за допомогою клавіш **◀, ▶, ▲, ▼** усередині комутаційної програми, за допомогою **OK** ви переходите до вибору елемента з'єднання або блоку, за допомогою **ESC** ви залишаєте режим програмування.
- Коли курсор з'являється у формі суцільного прямокутника, ви повинні вибрати елемент з'єднання або блок.
 - Для вибору елемента з'єднання або блоку використайте клавіші **▲, ▼**.
 - Натисніть **OK**, щоб прийняти вибір.
 - Натисніть **ESC**, щоб повернутися назад на один крок.

Планування

- Перед уведенням комутаційної програми намалюйте повний її план на папері або програмуйте LOGO! безпосередньо, використовуючи LOGO! Soft Comfort.
- LOGO! може зберігати тільки повні програми, що не мають помилок.

1.2.5 Уведення й запуск комутаційної програми

При підключенні LOGO! до напруги живлення на дисплеї з'являється наступне повідомлення:

No Program Press ESC

LOGO! Переключається у режим програмування, натисканням клавіші “ESC”. Після цього ви потрапите в головне меню LOGO!:

>Program..
Card..
Clock..
Start

Першим символом у першому рядку є знак ”>”. Для переміщення “>” нагору й униз використайте клавіші ▲, ▼. Перемістіть ”>” на “Program..” і натисніть ОК. LOGO! переходить до меню програмування:

>Edit..
Clear Prg
Password

Перемістіть ”>” на “Edit..” (для редагування, тобто уведення) і натисніть ОК, LOGO! переходить до меню редагування:

>Edit Prg
Edit Name
Memory?

Перемістіть символ ”>” на “Edit Prg” (для редагування комутаційної програми) і натисніть клавішу ОК. Тепер LOGO! покаже вам перший вихід Q1. Послідовність подальших дій показана на рисунку 1.8.

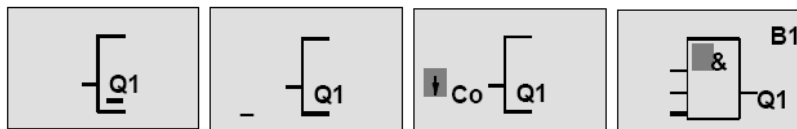


Рисунок 1.8 – Порядок введення програми

Як видно, буква Q в Q1 підкреслена. Це підкреслення являє собою курсор. Курсор показує поточну позицію в програмі. Курсор можна переміщати натисканням клавіш ◀, ▶, ▲, ▼. Тепер натисніть клавішу ◀ Курсор переміщається вліво. Для переходу до режиму уведення натисніть ОК. Курсор відображається у вигляді суцільного прямокутника: тепер ви можете вибрати елемент з'єднання або блок. У цей час LOGO! пропонує вам різні можливості вибору. Виберіть GF (основні функції), натискаючи клавішу ▼, поки не з'явиться GF, і натисніть ОК. Тепер LOGO! відобразить перший блок у списку основних функцій. Натисніть ОК, щоб підтвердити свій вибір і покинути діалог. Тепер ви ввели перший блок. Кожний блок, що ви вводите, одержує номер. Тепер необхідно підключити входи блоку.

Натисніть клавішу ОК.

Виберіть список Co.

Натисніть ОК.

За допомогою клавіші ▼ ви переміщаєтеся від початку списку Co: I1, I2...Io до 'x'. Виберіть I3, натисніть ОК. Далі виберіть попередній блок і підключіть до

НЬОГО ВХОДИ.

Якщо ви хочете ще раз переглянути свою програму, то ви можете використати клавішу ◀ або ▶ для переміщення курсору програмою.

Поверніться в меню програмування: натисніть **ESC**.

LOGO! тепер зберіг вашу комутаційну програму в енергонезалежній пам'яті. Програма залишиться в пам'яті LOGO! доти, поки ви явно не видалите її введенням відповідної команди.

Переведення LOGO! у режим RUN робиться в головному меню.

Поверніться в головне меню: Натисніть **ESC**.

Перемістите '>' на **'Start'**.

Прийміть **'Start'**: Натисніть **OK**. LOGO! запускає комутаційну програму.

1.3 Програмування LOGO! за допомогою LOGO!Soft Comfort

За допомогою LOGO!Soft Comfort можуть бути ефективно й комфортабельно створені комутаційні програми для LOGO!. Створення програм відбувається шляхом простого розташування й з'єднання програмних елементів на поверхні програмування.

Особливо зручними, крім іншого, є симуляція програм у режимі оффлайн (Offline), у якій можна одночасно подивитися статус багатьох спеціальних функцій, а також професійне документування створених програм.

Після старту програми ви бачите панель програмування LOGO!Soft Comfort показану на рисунку 1.9. При цьому, більша частина екрана відведена під створення комутаційної програми. На цій панелі програмування будуть розташовані символи й з'єднання комутаційного плану.

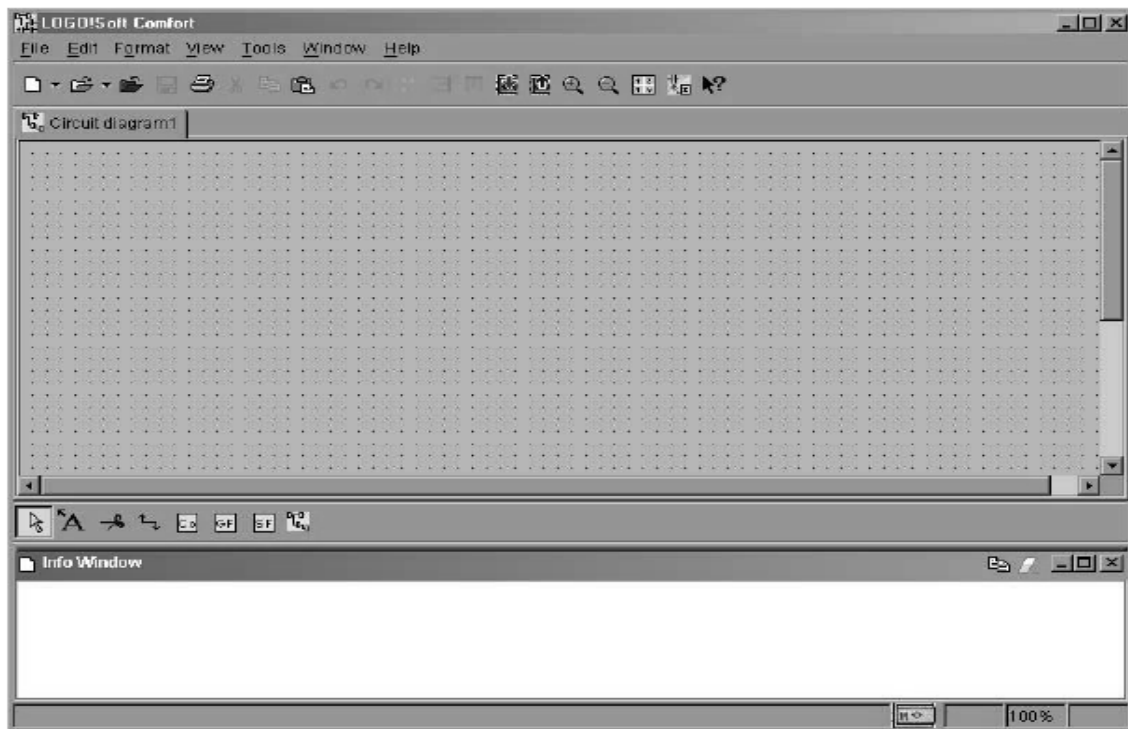


Рисунок 1.9 – Панель програмування LOGO!Soft Comfort

На самому верху у вікні LOGO!Soft Comfort розташована панель меню. На панелі меню ви знайдете різні команди для керування й роботи з комутаційними програмами, а також налаштування й функції завантаження програм.

Над панеллю програмування перебуває лінійка стандартних символів. При кожному старті LOGO!Soft Comfort з'являється лінійка стандартних символів, скорочена до основних функцій. У вас є можливість викликати панель програмування для нової комутаційної програми, відкрити вже існуючий проект або завантажити програму з LOGO! в LOGO!Soft Comfort для подальшого редагування й доповнення. Повна лінійка стандартних символів з'являється при відкритті проекту. Тут можна, використовуючи кнопки на лінійці символів, викликати панель для створення нового проекту або завантажити вже існуючий проект, зберегти його або роздрукувати. Також можна вирізати об'єкти із програми, копіювати їх, вставляти або активувати передачу даних з LOGO! – і в LOGO!. Кнопкою миші ви можете виділити й пересунути лінійку стандартних символів. Якщо ви закриєте лінійку стандартних символів, вона буде автоматично розташована зверху під панеллю меню.

Унизу перебуває панель інструментів. Використовуючи запропоновані тут символи можна переходити в різні робочі режими для швидкої й легкої роботи із програмою й складання комутаційного плану. Кнопкою миші ви можете виділити й пересунути панель інструментів. Якщо ви закриєте панель інструментів, вона буде автоматично розташована зверху під лінійкою стандартних символів.

Під панеллю програмування перебуває інформаційне вікно. В інформаційному вікні відображаються інформація й повідомлення, що виникають при симуляції програми. Також, модулі LOGO!, які можуть бути запропоновані вам викликом функції “Інструменти” як можливі модулі для роботи програми. Звичайно, інформаційне вікно відкривається разом із проектом. Кнопкою миші ви можете переміщати інформаційне вікно або зовсім вивести його за межі оболонки LOGO!Soft Comfort, так що буде утворене самостійне вікно. Інформаційне вікно можна відкрити й закрити за допомогою меню “Вид” -> “Інформаційне вікно” або “Швидкою клавішею” [F4].

Рядок стану перебуває на нижній межі оболонки LOGO!Soft Comfort. Тут ви одержите дані про застосований інструмент, обраному модулі LOGO!, статусі програми, обраному масштабі й сторінці комутаційного плану.

1.3.1 Створення програм

Після того, як ви запустили LOGO!Soft Comfort, ви можете відразу почати створення нової програми.

Для цього, на лінійці стандартних символів клацніть по значку "Новий".
-> Новий файл

Після цього, LOGO!Soft Comfort відкриє нове вікно й ви зможете проектувати нову комутаційну схему на вільній панелі програмування.

Щоб почати програмування в LOGO!Soft Comfort знадобляться тільки клавіші символної лінійки й панелі інструментів.

Для того, щоб вибрати потрібний інструмент або символ, підведіть до нього показник миші й клацніть її лівою клавішею. Обраний символ буде відображений задіяним.

Перший крок створення комутаційної програми складається у виборі необхідних функціональних блоків для реалізованої схеми. Ви можете вибирати, що розташувати на панелі програмування спочатку: входи або виходи, базові або спеціальні функції, порядок не має значення.

На панелі інструментів під символом **Co** ви знайдете елементи з'єднання й константи, тобто входи, виходи й постійні рівні сигналу. Під символом **GF** ви знайдете базові функції булевої алгебри або прості логічні елементи. Блоки спеціальних функцій перебувають під символом **SF**. Ви також можете викликати відповідні функціональні групи "Швидкими клавішами".

Клацніть по символі або натисніть на "Швидку клавішу" функціональної групи, що містить необхідний вам блок. Ви також можете вибрати його за допомогою функціональних кнопок. Під панеллю програмування з'являться всі функціональні блоки, які належать до обраної функціональної групи. Приклад показано на рисунку 1.10.



Рисунок 1.10 – Приклад функціональної групи для ФБ редактора

За допомогою миші ви можете розташувати обрану функцію на панелі програмування. Перша функція групи попередньо задіяна, інші функції можуть бути обрані мишею перед їхнім транспортуванням на панель програмування.

Немає необхідності зараз точно розташовувати об'єкт. При з'єднанні блоків один з одним, а також при коментуванні схеми може знадобитися знову пересувати їх для досягнення оптимального виду програми.


Номера блоків не можуть бути змінені. Нумерація базових і спеціальних функцій відбувається автоматично в тім порядку, у якому ви їх

розташовуєте на панелі програмування. Константи й елементи з'єднання мають на місці номера блоку відповідне маркування як на самому LOGO! або просте позначення блоку. При цьому, кожному **Входу**, **Виходу** й **Маркеру** може бути привласнені символічне ім'я й коментарі. Блоки сигналів High і Low не мають нумерації.

Якщо клацнути правою клавішею миші, відкриється контекстне меню яке, крім інших режимів роботи з об'єктом, запропонує вам відповідну допомогу. Зміст контекстного меню залежить від того, на якому об'єкті перебуває стрілка миші. Об'єктами можуть бути не тільки блоки й сполучні лінії, але й панель програмування й символічні лінійки.

Подвійним клацанням миші по функціональному блоку відкривається вікно для завдання властивостей блоку. У “Спеціальних функцій”, деяких “Базових функцій” і “Констант/ елементів з'єднання” крім карти коментарів є одна або кілька карт для реєстрації параметрів. У них ви можете заносити значення або установки, якими повинен володіти блок у вашій програмі. Натисканням кнопки «Допомога» можна одержати інформацію про параметри відповідного блоку.

Блоки спеціальних функцій ви впізнаєте також за їхніми поточними параметрами, які розташовані з лівої сторони блоку й виділені зеленим шрифтом.

Для того, щоб закінчити програму, окремі функціональні блоки повинні бути з'єднані між собою. Натисніть для цього символ «З'єднати»  на панелі інструментів. Перемістіть стрілку миші до піну блоку й натисніть ліву клавішу. Утримуючи ліву клавішу миші натиснутою, перемістіть стрілку до іншого піну блоку, з яким ви хочете встановити з'єднання й відпустіть клавішу. LOGO!Soft Comfort з'єднає блоки між собою.

Наступні правила діють при з'єднанні функціональних блоків:

- З'єднання може бути встановлено тільки між виходом одного блоку й входом іншого.
- Один вихід може бути з'єднаний з багатьма входами, однак один вхід - не з багатьма виходами.
- Вхід і вихід не можуть бути з'єднані в одному програмному шляху (рекурсія). При необхідності в такому з'єднанні використайте проміжний Вихід або Маркер.
- Спеціальні функції мають "входи" зелених кольорів. Вони не служать для з'єднання, а тільки лише забезпечують прив'язку встановлених параметрів.
- Аналогові входи й виходи не можуть бути з'єднані з дискретними входами й виходами.

1.3.2 Симуляція комутаційної програми

За допомогою симуляції програма може бути протестирована й змінена щодо її параметрів. Таким чином, ви можете бути впевнені в тім, що

завантажите в LOGO! працюючу й оптимізовану програму.

Для того, щоб почати симуляцію, клацніть мишею по символі “Симуляція” на панелі інструментів. Програма перейде в режим симуляції. Відкриється інформаційне вікно. В інформаційному вікні відображаються можливі повідомлення про помилки під час початку симуляції, а також можливі модулі LOGO!, які були встановлені через меню “Інструменти” або “Швидкою” клавішею [F2]. Крім того, біля дати й часу вказується назва програми, для якої відображається повідомлення. Таким чином, у випадку з багатьма відкритими програмами ви можете встановити, до якої програми відноситься повідомлення.

На початку симуляції програма аналізується щодо задіяних ресурсів і застосовуваного LOGO!. Програмні ресурси й можливі повідомлення про помилки відображаються в інформаційному вікні.

Інформаційне вікно може бути відкрито й закрито за допомогою меню “Вид” -> “Інформаційне вікно”, або “Швидка” клавіша [F4]. При виклику інформаційне вікно звичайно з'являється внизу на панелі програмування. Також, як і символічні лінійки, його можна переміщати мишею й установлювати ліворуч або праворуч, або зверху над панеллю програмування. Якщо за допомогою миші висунути інформаційне вікно з LOGO!Soft Comfort, то воно може бути відкрито як самостійне вікно за межами LOGO!Soft Comfort. Усю інформацію, яка з'являється в LOGO!Soft Comfort, можна переглянути тут за допомогою смуг прокручування. Інформація не зберігається довго в інформаційному вікні. Після повторного запуску LOGO!Soft Comfort інформаційне вікно буде порожнім.

Тексти, що з'являються в інформаційному вікні, можна виділяти для стирання або копіювання в інші додатки. Ви також можете вносити ваші власні замітки в інформаційне вікно.

У режимі симуляції під панеллю програмування й над панеллю інструментів з'являється індикатор або вікно для відбиття статусу входів, виходів і мережі. За допомогою миші ви можете вибрати й перемістити індикатор статусу або зовсім вивести його за межі LOGO!Soft Comfort так, що він утворить власне окреме вікно. Це, насамперед, корисно тоді, коли ви застосовуєте велику кількість входів і виходів у програмі. Тоді можна розташовувати входи й виходи на панелі на ваше бажання.

Те, який сигнал у цей момент несе лінія з'єднання (1 або 0), ви можете визначити за кольорами лінії. Лінії, що мають сигнал “1”, зображуються звичайно червоним кольором, лінії із сигналом “0” – звичайно, синім.

Приклад для ФБ редактора показано на рисунку 1.11.

Ви можете також змінити роботу входу в режимі симуляції, клацнувши по ньому правою клавішею миші. У меню “Властивості блоку” ви знайдете реєстраційну картку “Симуляція”, у якій ви можете вибрати режим роботи входу (Перемикач/Кнопка/Частотний). Клавішею “Застосувати” ви активуєте обраний режим.

Кнопка активна той час, що ви втримуєте неї в натиснутому стані. Як тільки ви відпустите кнопку, контакт буде знову відкритий.

Контакти перемикача замикаються після короткого натискання й розмикаються після повторного натискання.

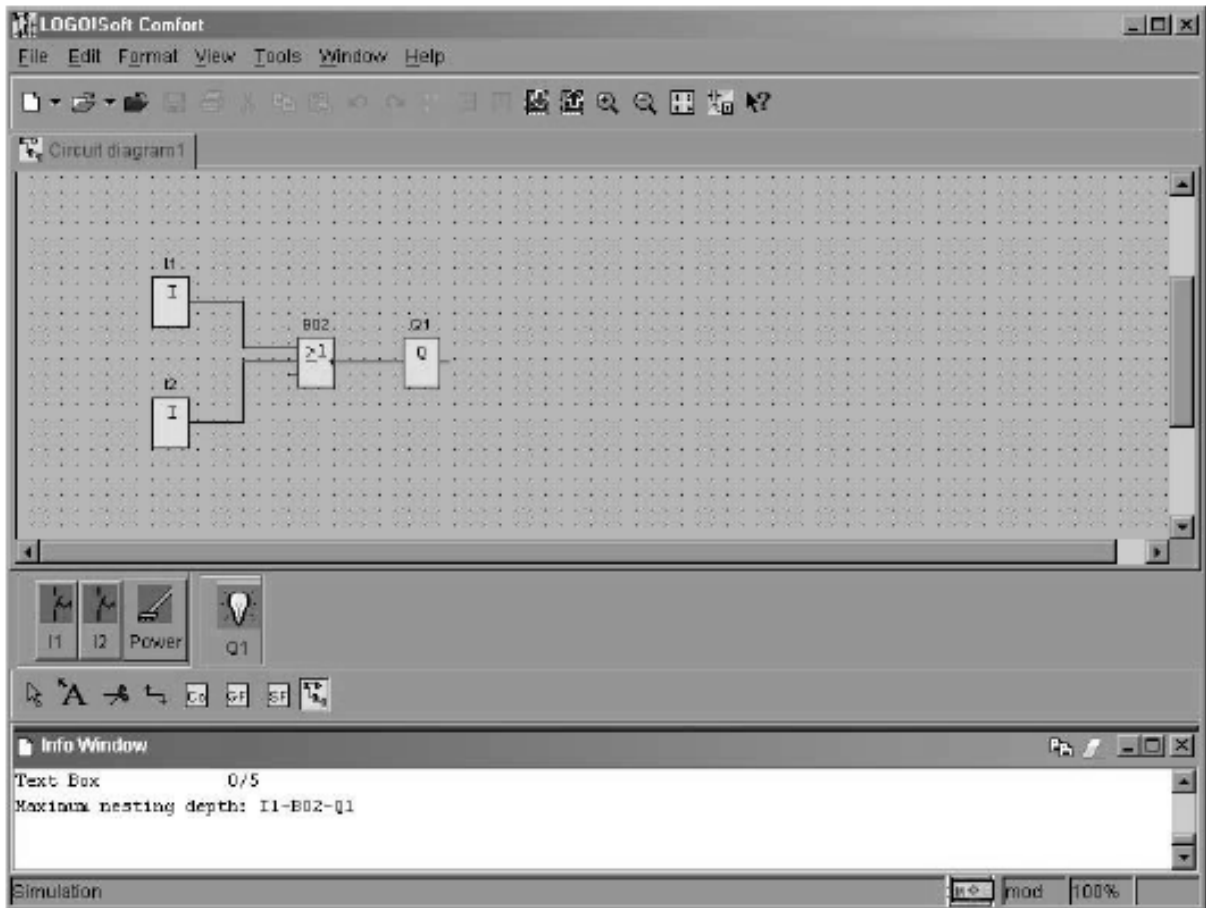


Рисунок 1.11 – Вікно LOGO!Soft Comfort в режимі симуляції

Частота частотного входу може бути заздалегідь установлена або змінена в режимі симуляції. Одиниця виміру встановлюваної частоти – Гц. Частотний вхід являє собою особливий випадок, тому що його використання має сенс тільки спільно зі спеціальною функцією «Частотний дискримінатор».

Якщо в режимі симуляції два рази клацнути по блоку, з'явиться вікно, у якому, як і в режимі програмування, ви можете заново робити коментарі й змінювати параметри. При цьому, у режимі симуляції відображаються також поточні значення параметрів. Завдяки можливості такого аналізу ви можете уявити поведінку вашої програми. Під час симуляції може бути відкрито багато вікон параметрування.

1.4 Завдання для самостійної роботи

Необхідно спроектувати систему автоматичного керування для заданого варіанта.

Проект повинен складатися з електричної схеми пристрою, блок-схеми алгоритму функціонування, програми для контролера та інструкції для користувача.

Варіант 1

Зрошення рослин в оранжерей

В оранжерей необхідно управляти зрошенням рослин за допомогою LOGO! При цьому розрізняють три типи рослин. У випадку рослин типу 1 мова йде про водорості в басейні, рівень води в якому повинен підтримуватися в певному діапазоні. Рослини другого типу повинні зрошуватися ранком і ввечері по 3 хвилини, а рослини третього типу – кожний другий вечір протягом 2 хвилин. Звичайно, автоматичне зрошення може й вимикатися.

Вирішення за допомогою LOGO!

Зрошення рослин – Тип 1: Через поплавкові вимикачі для максимального й мінімального значення (на I1 та I2) рівень води в басейні завжди підтримується в цьому заданому діапазоні.

Зрошення рослин – Тип 2: Зрошення включається через реле часу ранком з 6:00 до 6:03 і ввечері з 20:00 до 20:03 щораз на 3 хвилини (щодня).

Зрошення рослин – Тип 3: За допомогою функції імпульсного струму рослини зрошуються тільки кожний другий день; завжди ввечері протягом 2 хвилин, коли спрацьовує сутінковий вимикач на I3.

Використовувані компоненти:

LOGO!.

I1 – Поплавковий вимикач для максимального значення (розмикальний контакт).

I2 – Поплавковий вимикач для мінімального значення (замикаючий контакт).

I3 – Сутінковий вимикач (замикаючий контакт).

I4 – Вимикач для автоматично керованого зрошення (замикаючий контакт).

Q1 – Електромагнітний клапан для зрошення рослин типу 1.

Q2 – Електромагнітний клапан для зрошення рослин типу 2.

Q3 – Електромагнітний клапан для зрошення рослин типу 3.

Особливості:

Час зрошення ранком і ввечері може бути довільно змінено.

Поряд зі зрошенням рослин за допомогою LOGO! можна також управляти освітленням або провітрюванням оранжерей.

Варіант 2

Керування стрічковими транспортерами

За допомогою LOGO! необхідно управляти 3 транспортерами для транспортування деталей.

Установка, що перебуває перед транспортерами, кожних 30 секунд поставляє на стрічку деталі. Кожна деталь вимагає для переміщення по стрічці близько 1 хвилини. Тому що установка може мати багато часів простою, то

транспортери, залежно від того, повинні деталі транспортуватися чи ні, повинні автоматично запускатися або зупинятися.

Вирішення за допомогою LOGO!

Установка включається через кнопку ВКЛ на I2 і вимикається через кнопку ВИКЛ на I1. Кожний з 3 транспортерів починає рухатися двигуном (на Q1, Q2, Q3), а 3 реле близькості реєструють деталі на кожній зі стрічок (на I4, I5, I6). Через четверте реле близькості на I3 деталі реєструються на початку стрічки 1 (приходять деталі з установки). Якщо натиснуто кнопку ВКЛ, і деталі необхідно переміщати, то стрічки запускаються одна за іншою (послідовність - стрічка 1, стрічка 2, стрічка 3). Якщо до появи нової деталі проходить більше 1 хвилини, то стрічки зупиняються (у послідовності: стрічка 1, стрічка 2, стрічка 3). Якщо деталі з установки не надходять на транспортери більше 100 секунд, то настає час простою на 15 хвилин, про яке сигналізує лампа на Q4.

Використовувані компоненти:

LOGO! .

I1 – Кнопка ВИКЛ (замикаючий контакт).

I2 – Кнопка ВКЛ (замикаючий контакт).

I3 – Датчик для реєстрації деталей з установки (замикаючий контакт).

I4 – Датчик для реєстрації деталей на стрічці 1 (замикаючий контакт).

I5 – Датчик для реєстрації деталей на стрічці 2 (замикаючий контакт).

I6 – Датчик для реєстрації деталей на стрічці 3 (замикаючий контакт).

Q1 – Стрічка 1.

Q2 – Стрічка 2.

Q3 – Стрічка 3.

Q4 – Сигнальна лампа.

Особливості:

Інші часи перемикання можуть установлюватися довільно.

Простота зміни наявних установок.

Усі датчики підключаються безпосередньо до LOGO!.

Варіант 3

Керування згинальним верстатом

За допомогою LOGO! необхідно управляти процесом згинання вихлопних труб. Процес згинання запускається тоді, коли є в наявності як труба, так і деталь, що приєднується.

Якщо деталь несправна або відсутня, то про це повідомляється за допомогою сигнальної лампи.

Вирішення за допомогою LOGO!

За допомогою реле близькості на I1 реєструється наявність труби (для цього встановлюється вповільнення притягання в 1 секунду). Потім труба затискається за допомогою електромагнітного клапана на Q1. Якщо є в наявності й деталь, що приєднується (датчик на I2), то труба звільняється й

дається команда на деблокування згинання шляхом скидання деблокуючого реле на Q2 ($Q2 = 0$). Процес деблокування триває не більше 5 секунд. Це граничний час для деблокування. Якщо протягом цих 5 секунд труба не реєструється, то деблокування процесу згинання відміняється шляхом установки деблокуючого реле ($Q2 = 1$). Якщо деталь розпізнається як дефектна або неповна, то про це сповіщається за допомогою сигнальної лампи на Q3. Через I3 помилка може бути квитирувана, а несправна деталь вилучена. При цьому труба звільняється, і процес може початися знову.

Використовувані компоненти:

LOGO!.

I1 – Датчик наявності труби (замикаючий контакт).

I2 – Датчик наявності деталі, що приєднується (замикаючий контакт).

I3 – Кнопка для квитирування помилок (замикаючий контакт).

Q1 – Електромагнітний клапан для циліндра, що затискає трубу.

Q2 – Деблокуюче реле.

Q3 – Лампа для сигналізації про помилки.

Варіант 4

Освітлення вітрини

За допомогою LOGO! повинні автоматично освітлюватися товари, виставлені у вітрині. При цьому розрізняють 4 групи джерел світла. Одна для освітлення вдень, одна для додаткового освітлення ввечері, одна для мінімального освітлення вночі й одна для створення світлових плям, які повинні окремо освітлювати розміщені предмети.

Вирішення за допомогою LOGO!

Вітрина повинна освітлюватися з понеділка по п'ятницю з 8:00 до 22:00, у суботу з 8:00 до 24:00 і в неділю з 12:00 до 20:00. Протягом цих інтервалів через реле часу включається група джерел світла 1 на Q1. Крім того, по вечорах підключається група джерел світла 2, коли спрацьовує сутінкове реле на вході I1. Поза вищезгаданими інтервалами часу група джерел світла 3 на Q3 бере на себе мінімальне освітлення після деблокування сутінкового реле. Через сигналізатор переміщень на I4 протягом усього часу включаються або вимикаються світлові плями (група джерел світла 4 на Q4).

За допомогою тестової кнопки на I3 можна на 1 хвилину включити всі групи джерел світла, щоб, наприклад, перевірити їхнє функціонування або їх поправити.

Використовувані компоненти:

LOGO!.

I1 – Сутінкове реле (замикаючий контакт).

I2 – Вимикач ВКЛ(замикаючий контакт).

I3 – Тестовий вимикач (замикаючий контакт).

I4 – Сигналізатор переміщень(замикаючий контакт).

Q1 – Група джерел світла 1.

Q2 – Група джерел світла 2.

Q3 – Група джерел світла 3 (мінімальне освітлення).

Q4 – Група джерел світла 4 (світлові плями).

Варіант 5

Установка дзвінкової сигналізації, наприклад, у школі

Дзвінок у школі управляється за допомогою LOGO!. Дзвінок повинен дзенькати в певні моменти часу протягом 2 секунд (початок занять, зміни, кінець занять).

Вирішення за допомогою LOGO!

Через убудоване в LOGO! реле часу задаються часи для початку занять, змін і кінця занять. Дзвінок повинен дзенькати з понеділка по п'ятницю в 8:00, 9:45, 10:00, 12:45, 13:30 і 16:30. Правда, у п'ятницю заняття кінчаються вже в 15:30. Через затримку включення забезпечується, щоб дзвінок дзенькав тільки 2 секунди.

Використовувані компоненти

LOGO!

Q1 – Дзвінок.

Варіант 6

Контроль місць для стоянки автомобілів

На автостоянці є в розпорядженні певна кількість місць для стоянки автомобілів. Вхідний світлофор повинен автоматично перемикатися із зеленого на червоний, коли всі місця зайняті. Як тільки місця знову звільнюються, в'їзд знову забезпечується включенням зеленого сигналу.

Вирішення за допомогою LOGO!

Автомобілі, що в'їжджають і виїжджають підраховуються за допомогою фотодатчиків (на I1 та I2) убудованим в LOGO! лічильником. При в'їзді автомобіля (I1) лічильник збільшується на 1, а при виїзді автомобіля (I2) він знову зменшується на 1. Через I2 і імпульсну функцію задається напрямок рахунку (уперед/назад) на лічильнику. При досягненні встановленого параметра перемикається світлофорний пристрій на Q1. За допомогою кнопки на I3 значення лічильника й вихід Q1 можуть бути скинуті.

Використовувані компоненти:

LOGO!.

I1 – Фотодатчик "В'їзд" (замикаючий контакт).

I2 – Фотодатчик "Виїзд" (замикаючий контакт).

I3 – Кнопка скидання (замикаючий контакт).

Q1 – Реле світлофора (перемикаючий контакт).

Особливості:

Поточне значення лічильника може бути просто відображене через дисплей. Максимальне значення лічильника може змінюватися довільно.

Застосування може бути легко розширено: наприклад, перемикачі між двома значеннями лічильника (стояночні місця зарезервовані для підприємства).

Варіант 7

Зовнішнє освітлення

За допомогою LOGO! необхідно управляти зовнішнім освітленням будинку. При цьому розрізняють основне й допоміжне освітлення з ручним і автоматичним режимом. Основне освітлення постійно включене протягом установленого інтервалу часу, допоміжне освітлення, навпроти, тільки на певний час, коли спрацьовує також сигналізатор переміщення. Освітлення в загальному випадку включається тільки при настанні сутінків.

Вирішення за допомогою LOGO!

Основне освітлення (на Q1) в автоматичному режимі включається тільки з 6:00 до 24:00, якщо також спрацьовує сутінкове реле на I1. Допоміжне освітлення (на Q2) включається за допомогою сигналізатора переміщення на I2 на 90 секунд (в інтервалах з 6:00 до 8:00 і з 17:00 до 24:00). Через I4 (положення перемикача – ручний режим) основне й допоміжне освітлення включаються незалежно від реле часу й сутінкового реле, наприклад, для перевірки.

Використовувані компоненти:

LOGO! .

I1 – Сутінкове реле (замикаючий контакт).

I2 – Сигналізатор переміщення (замикаючий контакт).

I3 – Положення перемикача „автоматика" (замикаючий контакт).

I4 – Положення перемикача „ручний режим" (замикаючий контакт).

Q1 – Основне освітлення.

Q2 – Допоміжне освітлення.

Особливості

Інтервали часу можуть встановлюватися індивідуально; наприклад, у робочі й вихідні дні або інші тривалості інтервалів часу. Освітлювальна установка може бути легко розширена; наприклад, додаткові сигналізатори переміщення або інші групи освітлення, щоб чіткіше диференціювати окремі діапазони.

Варіант 8

Керування жалюзі

За допомогою LOGO! необхідно управляти жалюзі житлового будинку. За допомогою селекторного перемикача може бути обрані ручний режим або автоматичне керування. Залежно від часу, темряви й дня тижня жалюзі автоматично закриваються або відкриваються.

Вирішення за допомогою LOGO!

Ручний режим: за допомогою вимикачів на I2 (ВІДКРИТИ) і I3 (ЗАКРИТИ) жалюзі можна відкривати й закривати вручну. Передумовою для цього є те, що селекторний перемикач на I6 не стоїть в положенні “Автоматика”.

Автоматичний режим: для переходу в автоматичний режим селекторний перемикач (I6) повинен перебувати в положенні “Автоматика”. Коли спрацьовує сутінкове реле на I1, жалюзі закриваються на період з 18:00 до 7:00. Відкриваються вони протягом дня між 7:00 і 18:00.

Через кінцеві вимикачі на I4 і I5 робиться опитування, відкриті жалюзі або закриті.

Використовувані компоненти:

LOGO!.

I1 – Сутінкове реле (замикаючий контакт).

I2 – Вимикач для відкриття в ручному режимі (замикаючий контакт).

I3 – Вимикач для закриття в ручному режимі (замикаючий контакт).

I4 – Кінцевий вимикач “Жалюзі відкриті”(розмикальний контакт).

I5 – Кінцевий вимикач “Жалюзі закриті” (розмикальний контакт).

I6 – Селекторний перемикач “Автоматика”.

Q1 – Відкрити жалюзі.

Q2 – Закрити жалюзі.

Особливості

Часи можуть бути просто пристосовані до індивідуальних умов; наприклад, різні часи в робочі дні й наприкінці тижня або під час відпустки. Різне керування для двох діапазонів/ситуацій через ще вільні виходи.

Варіант 9

Зовнішнє й внутрішнє освітлення житлового будинку

За допомогою LOGO! необхідно управляти зовнішнім і внутрішнім освітленням житлового будинку. При цьому у випадку відсутності хазяїв або в темний час повинне сигналізуватися наближення людей. Через сигналізатор переміщення й контакт тривоги установки тривожної сигналізації включається зовнішнє й внутрішнє освітлення.

Вирішення за допомогою LOGO!

Зовнішнє освітлення розділене на три ділянки (на Q1, Q2, Q3). Для кожної ділянки використовується власний сигналізатор переміщення (на I2, I3, I4). Якщо на певному інтервалі часу спрацьовує один із цих сигналізаторів, то відповідне зовнішнє освітлення включається на 90 секунд. Діапазон часу задається через убудоване в LOGO! реле часу (з 17:00 до 7:00). Завдяки сутінковому реле на I1 гарантується, що включення відбувається тільки в темний час доби. На I5 підключений четвертий сигналізатор переміщення, що незалежно від часу й темряви включає всі три зовнішні освітлення на 90 секунд. Зовнішні освітлення включаються також на 90 секунд через контакт тривоги установки тривожної сигналізації на I6. Крім того, після відключення зовнішнього освітлення на 90 секунд включається внутрішнє освітлення.

Через сигналізатор переміщення на I5 і контакт тривоги внутрішнє освітлення включається на 90 секунд негайно.

Використовувані компоненти:

LOGO!.

I1 – Сутінкове реле (замикаючий контакт).

I2 – Сигналізатор переміщення 1 (замикаючий контакт).

I3 – Сигналізатор переміщення 2 (замикаючий контакт).

I4 – Сигналізатор переміщення 3 (замикаючий контакт).

I5 – Сигналізатор переміщення 4 (замикаючий контакт).

I6 – Контакт тривоги установки тривожної сигналізації (замикаючий контакт).

Q1 – Зовнішнє освітлення 1.

Q2 – Зовнішнє освітлення 2.

Q3 – Зовнішнє освітлення 3.

Q4 – Внутрішнє освітлення.

Особливості:

Простота зміни встановлених часів; наприклад, іншого діапазону реле часу або іншої тривалості освітлення.

Варіант 10

Керування мішалкою для молока й вершків

За допомогою LOGO! необхідно управляти мішалкою для молока й вершків на молочній фермі. За допомогою перемикача режимів роботи можуть бути обрані автоматичний режим або режим безпосереднього керування. Несправності сигналізуються за допомогою лампи й аварійного звукового сигналу.

Вирішення за допомогою LOGO!

Якщо перемикач режимів роботи перебуває в положенні “Автоматика” (I1), то мішалка (на Q1) запускається негайно. Автоматичний режим означає, що мішалка включається й вимикається через задані інтервали часу (15 секунд включена, 10 секунд – пауза). Мішалка працює із цими інтервалами, поки перемикач режимів роботи не буде переведений у положення “0”. У режимі прямого керування (I2 – положення “Пряме керування”) мішалка працює без обліку інтервалів часу. При спрацьовуванні автомата захисту двигуна (на I3) активізуються лампа сигналізації про несправності (Q2) і аварійний звуковий сигнал (Q3). Інтервали, з якими подається звуковий сигнал, встановлюються за допомогою датчика тактових імпульсів на 3 секунди. Звуковий сигнал може бути перерваний за допомогою кнопки скидання на I4. Якщо несправність усунута, то сигнальна лампа й звуковий сигнал знову скидаються. За допомогою кнопки “Контроль аварійної сигналізації” на I5 можна перевірити як сигнальну лампу, так і звуковий сигнал.

Використовувані компоненти:

LOGO! .

- I1 – Перемикач режимів роботи в положення “Автоматика” (замикаючий контакт).
- I2 – Перемикач режимів роботи в положення “Пряме керування” (замикаючий контакт).
- I3 – Аварійний контакт автомата захисту двигуна (замикаючий контакт).
- I4 – Кнопка скидання звукового сигналу (замикаючий контакт).
- I5 – Кнопка перевірки функціонування аварійної сигналізації (замикаючий контакт).
- Q1 – Мішалка.
- Q2 – Сигнальна лампа.
- Q3 – Аварійний звуковий сигнал.

Особливості:

Довільна зміна інтервалів перемішування.

Варіант 11

Освітлення спортзалу

За допомогою LOGO! управляється освітлення спортзалу й роздягалень у школі. Тому що вечорами спортзал знімали також різні спортивні об'єднання, за допомогою LOGO! реалізоване також примусове відключення, щоб не перевищувався час використання. За допомогою центрального вимикача освітлення може включатися й вимикатися зовсім незалежно.

Ворішення за допомогою LOGO!

Освітлення спортзалу (на Q1 і Q2) може включатися й вимикатися за допомогою кнопкового вимикача на I1. Через кнопковий вимикач на I2 включається й вимикається освітлення роздягалень. Примусове відключення вечорами реалізується через убудоване реле часу. У 21:45 протягом 5 секунд подається звуковий сигнал, який вказує користувачам, що час користування залом минуло. Спортзал звільняється, і світло вимикається. У 22:00 у спортзалі виключається перша група освітлення (Q1), а в 22:15 – друга група (Q2). Освітлення роздягалень потім вимикається в 22:25. Потім світло вже не можна більше включити. Через центральний вимикач світло можна включати й виключати зовсім незалежно (наприклад, завгоспом).

Канікулярний час може бути відмежовано блокуючим вимикачем на I4 вручну.

Використовувані компоненти

LOGO!.

- I1 – Кнопка освітлення спортзала (замикаючий контакт).
- I2 – Кнопка освітлення роздягалень (замикаючий контакт).
- I3 – Центральний вимикач (замикаючий контакт).
- I4 – Блокуючий вимикач для канікул (замикаючий контакт).
- Q1 – Група освітлення 1, спортзал.
- Q2 – Група освітлення 2, спортзал.

Q3 – Освітлення роздягалень.

Q4 – Звуковий сигнал.

Варіант 12

Рівномірне завантаження трьох споживачів

За допомогою LOGO! реалізується групове з'єднання трьох однакових споживачів. Із цих трьох споживачів два завжди повинні працювати. Щоб забезпечити рівномірне зношування, всі три споживачі повинні включатися й вимикатися поперемінно. Кожний споживач має аварійний вихід, що підведений до загальної аварійної сигналізації. Як тільки споживач повідомляє про несправності, він відключається, а інші два споживачі працюють.

Вирішення за допомогою LOGO!

Процес рівномірного завантаження споживачів виглядає в такий спосіб: Спочатку працюють споживачі 1 і 2 (на Q1 і Q2), потім споживачі 2 і 3 (на Q2 і Q3), потім споживачі 1 і 3 (на Q1 і Q3). Ця послідовність увесь час повторюється (починаючи з Q1 і Q2). Споживачі щораз працюють протягом установленого часу (наприклад, 3 секунди). Початок процесу реалізується за допомогою інвертованого реле із самоблокуванням. Установка сама запускається також після відновлення напруги (початковий стан).

Якщо в споживача 1 виникає несправність, то він відключається через аварійний вхід I1, і включається третій споживач. Несправність сигналізується через загальний аварійний сигнал на Q4. Якщо несправність усунута й натиснута квітуюча кнопка на I4, то LOGO! переходить у початковий стан, і процес знову починається з Q1 і Q2. Те ж саме справедливо також і для споживачів 2 і 3 (повідомлення про несправності споживача 2 на I2, повідомлення про несправності споживача 3 на I3).

Використовувані компоненти:

LOGO! .

I1 – Аварійний вхід споживача 1 (замикаючий контакт).

I2 – Аварійний вхід споживача 2 (замикаючий контакт).

I3 – Аварійний вхід споживача 3 (замикаючий контакт).

I4 – Кнопка квітання несправності (замикаючий контакт).

Q1 – Споживач 1.

Q2 – Споживач 2.

Q3 – Споживач 3.

Q4 – Вихід загального аварійного сигналу.

Особливості

Часи роботи споживачів можна міняти довільно.

Варіант 13

Керування послідовністю операцій верстата для зварювання кабелю великого перерізу

Технологічний процес на верстатах для зварювання кабелю повинен точно дотримуватися. Керування здійснюється тільки через педальний перемикач. У випадку помилкового керування цикл негайно переривається й повинен бути початий знову.

Вирішення за допомогою LOGO!

Процес зварювання запускається педальним перемикачем на П1. Будь-яке невірне керування й зрушення в часі з боку оператора повинні бути виключені. При натисканні педального перемикача кінець кабелю підводить до упору. Ножний перемикач повинен бути знову натиснутий протягом 3 секунд для затиску кінців кабелю (клапан на Q2). Інтервал часу в 3 секунди відображається світловим сигналом на Q1. Якщо педальний перемикач протягом 3 секунд був натиснутий другий раз, то відбувається процес зварювання. При новому натисканні педального перемикача кабель звільняється й простягається далі (клапан на Q3). Якщо перевищені 3 секунди після першого натискання педального перемикача, то затискний клапан негайно звільняє кабель, і він не зварюється. Цикл повинен бути початий знову.

Використовувані компоненти

LOGO!.

П1 – Педальний перемикач.

Q1 – Світловий сигнал часу попередньої установки (3 с).

Q2 – Клапан для затиску кабелю.

Q3 – Клапан для звільнення кабелю.

Особливості

Час попередньої установки в будь-який момент може бути пристосований до поточних умов.

Варіант 14

Східчастий вимикач, наприклад, для вентиляторів

За допомогою LOGO! повинне бути реалізоване східчасте перемикання чотирьох рівнів потужності вентилятора.

Вирішення за допомогою LOGO!

За допомогою кнопочного перемикача на П1 вентилятор запускається на рівні 1. Кожним наступним натисканням кнопки вентилятор перемикається на один рівень потужності вище. Це можливо до 4 разів (Q1, Q2, Q3 і Q4). Цей 4-східчастий перемикач був реалізований за допомогою убудованого лічильника. Залежно від того, скільки разів була натиснута кнопка П1, деблокується відповідний контактор (П1 натиснутий двічі -> активний вихід Q2). Щоб працював точно один контактор, перемикання між окремими контакторами відбувається тільки через короткий час очікування в 2 секунди. За допомогою кнопки П2 вентилятор може ступінь за ступенем перемикатися у зворотному напрямку.

Використовувані компоненти

LOGO!.

I1 – Кнопка “Збільшення” (замикаючий контакт).

I2 – Кнопка “Зменшення” (замикаючий контакт).

Q1 – Контактор рівня 1.

Q2 – Контактор рівня 2.

Q3 – Контактор рівня 3.

Q4 – Контактор рівня 4.

Особливості

Кількість ступенів перемикача може бути довільно змінена (2-, 3- або 4-східчастий).

Східчастий перемикач можна легко розширити, наприклад, вентилятор відключати негайно, якщо кнопка I2 натиснута тривалий час.

Варіант 15

Послідовне керування опалювальними казанами

За допомогою LOGO! повинно бути забезпечено, щоб чотири газових опалювальних казани не могли запускатися одночасно. Керування казанами деблокується за допомогою головного термостата.

Вирішення за допомогою LOGO!

Кожний із чотирьох опалювальних казанів має два ступені потужності. Кожний ступінь потужності поставлений у відповідність до виходу (з Q1 по Q8). До I1 підключений головний термостат. Через цей термостат може встановлюватися температура, при якій опалювальні казани повинні включатися або вимикатися. Якщо температура падає нижче 70 градусів, то через I1 запускається на нагрівання перший ступінь потужності казана 1 (Q1). Через 5 хвилин запускається ступінь потужності 2 казана 1 (Q2).

Поки не досягнута кінцева температура, кожних 5 хвилин для нагрівання деблокується наступний ступінь потужності (з Q3 до Q8). При досягненні кінцевої температури в 80 градусів казани знову послідовно відключаються. Спочатку ступені потужності 1 і 2 казана 1, через 5 хвилин - казан 2 і т.д. Після охолодження відбувається новий запуск, починаючи з 1.

Використовувані компоненти

LOGO!.

I1 – Головний термостат.

Q1 – Ступінь потужності 1 казана 1.

Q2 – Ступінь потужності 2 казана 1.

Q3 – Ступінь потужності 1 казана 2.

Q4 – Ступінь потужності 2 казана 2.

Q5 – Ступінь потужності 1 казана 3.

Q6 – Ступінь потужності 2 казана 3.

Q7 – Ступінь потужності 1 казана 4.

Q8 – Ступінь потужності 2 казана 4.

Особливості

Часи можуть бути просто пристосовані до потужності й режиму роботи.

Варіант 16

Відрізний пристрій, наприклад, для вогнепровідних шнурів

За допомогою LOGO! реалізовано відрізний пристрій для вогнепровідних шнурів піротехнічних зарядів. З умов безпеки, сповільнюючі вогнепровідні шнури довжиною 5 м повинні бути по можливості швидко розрізані на короткі шматки. Для цього шнур повинен просуватися на певну відстань до позиції відрізання. При досягненні заданої кількості шматків процес автоматично зупиняється.

Вирішення за допомогою LOGO!

Переміщення й відрізання вогнепровідного шнура реалізується за допомогою циліндрів, які, висуваючись, переміщують або відрізають шнур. До Q2 підключений електромагнітний клапан, що служить для того, щоб циліндри переходили у початкове положення. Для запуску повинна бути натиснута пускова кнопка на I1. Потім починає працювати електромагнітний клапан на Q2, і циліндри переходять у початкове положення. Відповідне повідомлення про знаходження циліндра для транспортування у початковому положенні видається індуктивним перемикачем на I3. Якщо він спрацював, то циліндр для транспортування на Q3 починає працювати. Цей циліндр висувається й переміщує весь шнур. Величина переміщення задається величиною ходу циліндра при висуванні. При досягненні кінцевого положення спрацьовує індуктивний перемикач на I4 для індикації стану “Циліндр транспортування висунутий”. Тепер починає працювати циліндр на Q4 для процесу відрізання. Він висувається й відрізає вогнепровідний шнур. Коли він досягає кінцевого положення, то про це повідомляється за допомогою індуктивного перемикача на I5 “Циліндр для відрізання висунутий”. Потім Q3 і Q4 скидаються, і процес відрізання починається знову. За допомогою убудованого в LOGO! лічильника кількості виробів може здійснюватися підрахунок окремих процесів відрізання. При кожному відрізання лічильник збільшує своє значення на 1. Після досягнення заданого числа виробів (80) новий цикл не починається. Це відображається за допомогою сигнальної лампи на Q1.

Для початку нового циклу й скидання лічильника кнопка вимикання на I2 повинна бути натиснута протягом більш ніж 2,5 секунд. Якщо кнопка вимикання або кнопка аварійного вимикання на I6 натиснута під час процесу відрізання, то цей процес переривається, і відбувається перехід у початкове положення. Стан лічильника зберігається, а електромагнітний клапан на Q2 відключається.

Використовувані компоненти

LOGO! .

I1 – Пускова кнопка (замикаючий контакт).

I2 – Кнопка вимикання (замикаючий контакт).

- I3 – Індуктивний перемикач, циліндр транспортування всунутий (замикаючий контакт).
- I4 – Індуктивний перемикач, циліндр транспортування висунутий (замикаючий контакт).
- I5 – Індуктивний перемикач, циліндр відрізання висунутий (замикаючий контакт).
- I6 – Кнопка аварійного відключення (розмикальний контакт).
- Q1 – Сигнальна лампа “Кількість виробів досягнута”.
- Q2 – Електромагнітний клапан (для початкового положення).
- Q3 – Циліндр для транспортування.
- Q4 – Циліндр для відрізання.

Варіант 17

Контроль тривалості використання, наприклад, у сонячній енергетичній установці

За допомогою LOGO! повинно бути забезпечено, щоб споживачі могли бути включені тільки на певний інтервал часу. Якщо заданий час перевищений, то LOGO! автоматично відключає споживачів. Це, наприклад, дуже корисно для сонячних енергетичних установок, тому що при цьому вдається уникнути глибокого розряду батарей.

Вирішення за допомогою LOGO!

LOGO! контролює тривалість включення приєднаних споживачів. Для окремих споживачів може бути заданий різний час. Кожному виходу ставиться у відповідність вхід: якщо натиснуто вимикач на вході I1 то негайно включається споживач на Q1. Протягом заданого інтервалу часу споживач може довільно часто включатися й вимикатися. Але якщо інтервал включення перевищений, то LOGO! автоматично відключає цього споживача. Інші входи й виходи (I2, I3 і Q 2, Q3) зв'язані один з одним у такий же спосіб. Завершення деблокування реалізоване в такий спосіб: через убудований в LOGO! лічильник встановлюється поточна тривалість включення шляхом подачі щохвилини на лічильник імпульсу датчиком тактових імпульсів. Тим самим може бути підрахована кількість минулих хвилин. Задане граничне значення відповідає максимальній тривалості включення (наприклад, 120 = 120 хвилинам для Q1). Коли це значення лічильника досягається, споживач відключається. Споживач залишається заблокованим доти, поки через реле часу не буде дана команда на деблокування (наприклад, щодня в 6:00).

Щоб дати сигнал про те, що максимальний час включення незабаром мине, до виходу Q4 підключена додаткова сигнальна лампа, що починає мигати за 15 хвилин до витікання часу.

Використовувані компоненти

LOGO! .

- I1 – Вимикач для споживача 1 (замикаючий контакт).
- I2 – Вимикач для споживача 2 (замикаючий контакт).

I3 – Вимикач для споживача 3 (замикаючий контакт).

Q1 – Споживач 1.

Q2 – Споживач 2.

Q3 – Споживач 3.

Q4 – Сигнальна лампа.

Особливості

Часи включення можуть бути просто змінені й адаптовані до тієї або іншої ситуації.

Момент деблокування може довільно змінюватися для кожного споживача, наприклад, тільки раз на тиждень. Крім того, через реле часу робота споживачів може бути обмежена певними інтервалами часу.

Варіант 18

Інтелектуальний педальний перемикач, наприклад, для вибору швидкостей

На робочому місці у верстата за допомогою педального перемикача можна вибирати різні швидкості обертання двигуна або відключати верстат. LOGO! бере на себе це керування.

Вирішення за допомогою LOGO!

Педальний перемикач верстата має два контакти, які підключені до LOGO! у такий спосіб: педаль “натиснута наполовину” на I1 і педаль “натиснута повністю” – на I2.

Для нормального робочого процесу досить 1-ї швидкості двигуна, що запускається включенням I1. 1-а швидкість двигуна починає діяти із затримкою в 2 секунди через вихід Q1. Якщо для робочого процесу потрібна більш висока швидкість, то 2-а швидкість двигуна може бути обрана подальшим натисканням I1. Друга швидкість двигуна також починає діяти після затримки в 2 секунди через вихід Q2. Якщо знову задіяти I1, то швидкість знову зменшується. Тобто при кожному включенні I1 по черзі починають діяти 1- й 2-а швидкості (щораз із затримкою в 2 секунди).

Якщо верстат працює на підвищеній швидкості, то це відображається за допомогою сигнальної лампи на Q3. Щоб зупинити верстат, педальний перемикач повинен бути натиснутий повністю. Тоді верстат відключається через I2.

Використовувані компоненти

LOGO!.

I1 – Педальний перемикач натиснутий наполовину, зміна швидкості.

I2 – Педальний перемикач натиснутий повністю, вимикання установки.

Q1 – 1-а швидкість двигуна.

Q2 – 2-я швидкість двигуна.

Q3 – Сигнальна лампа про роботу на 2-й швидкості.

Варіант 19

Керування піднімальною платформою

За допомогою LOGO! реалізовано керування піднімальною платформою. Для контролю зони переміщення піднімальної платформи встановлено декілька паралельно включених датчиків (наприклад, ультразвукових).

Вирішення за допомогою LOGO!

Піднімальна платформа за допомогою кнопок може переміщатися нагору або вниз. Для цього до I1 підключена кнопка “Нагору”, а до I3 – кнопка “Униз”. Відповідне кінцеве положення розпізнається кінцевим вимикачем. Кінцевий вимикач на I2 – для верхнього положення платформи, кінцевий вимикач на I4 – для нижнього положення. Якщо кінцеве положення досягнуто, то рухатися можна тільки в протилежному напрямку. Напрямок переміщення задається через кнопки на I1 та I3. За допомогою кнопки “Стоп” на I7 платформу можна зупинити. Ультразвукові датчики для контролю зони переміщення платформи підключені до I5. Якщо датчиками розпізнається перешкода, то платформа зупиняється. Але її можна переміщати в ручному режимі, якщо клавіша напрямку натиснута довше 2 секунд. Однак, якщо натиснуто кнопку аварійної зупинки на I7, то платформа зупиняється негайно й не може більше переміщатися за допомогою кнопок напрямку, поки не відпущена кнопка аварійної зупинки. Для кращого розпізнавання того, що платформа рухається, активізується попереджувальний сигнальний пристрій на Q3. Якщо платформа рухається нагору або вниз, то мигає попереджувальний світловий сигнал на Q3.

Використовувані компоненти:

LOGO!.

I1 – Кнопка “Нагору” (замикаючий контакт).

I2 – Верхній кінцевий вимикач (замикаючий контакт).

I3 – Кнопка “Униз” (замикаючий контакт).

I4 – Нижній кінцевий вимикач (замикаючий контакт).

I5 – Датчики (розмикальні контакти).

I6 – Кнопка «Стоп» (замикаючий контакт).

I7 – Кнопка аварійної зупинки (замикаючий контакт).

Q1 – Платформа нагору.

Q2 – Платформа вниз.

Q3 – Попереджувальний світловий сигнал.

Варіант 20

Просочення текстильних виробів, керування стрічковими нагрівачами й транспортерами

LOGO! використовується при просоченні текстильних виробів. Для цього рулони текстилю розкочуються, простягаються через просочувальну ванну й висушуються на стрічкових нагрівачах. При цьому LOGO! бере на себе керування стрічковими транспортерами для просочення й сушіння.

Вирішення за допомогою LOGO!

Процес просочення запускається автоматично через убудоване в LOGO! реле часу. Кожний робочий день в 03:00 у першу чергу запускаються

вентилятори на Q1 для провітрювання. Через 15 хвилин запускається перший стрічковий нагрівач на Q2. Через кожні 5 хвилин – інші на Q3, Q4 і Q5. Тому що стрічкові нагрівачі вимагають дуже багато часу для розігріву, то спочатку запускаються вони, й тільки через 3,5 години включається перший транспортер для просочувальної ванни. Другий і третій на Q7 і Q8 – через кожні 5 хвилин. Коли всі транспортери працюють, текстильні матеріали за допомогою стрічкових транспортерів пропускаються через просочувальну ванну, а потім висушуються на стрічкових нагрівачах. Якщо цей процес закінчений, то транспортери для нагрівання й просочення можуть бути негайно зупинені за допомогою кнопки на П1. Вентилятори ще працюють додатково протягом 1 години.

Використовувані компоненти:

LOGO!.

П1 – Кнопка вимикання (замикаючий контакт).

Q1 – Вентилятор

Q2 – Стрічковий нагрівач 1.

Q3 – Стрічковий нагрівач 2.

Q4 – Стрічковий нагрівач 3.

Q5 – Стрічковий нагрівач 4.

Q6 – Стрічковий транспортер 1 для просочення.

Q7 – Стрічковий транспортер 2 для просочення.

Q8 – Стрічковий транспортер 3 для просочення.

Варіант 21

Керування завантажувальним пристроєм бункера

LOGO! використовується для керування й контролю завантажувальних пристроїв бункерів. Бункери заповнюються через завантажувальний шланг вантажного автомобіля вапном або цементом.

Вирішення за допомогою LOGO!

Процес завантаження може бути початий тільки тоді, коли включений деблокуючий вимикач на П1 и завантажувальний шланг належним чином підключений. Язичковий контакт на завантажувальному штуцері сигналізує, чи правильно завантажувальний шланг з'єднаний з бункером. Це сигнал зчитується в LOGO! через вхід I2. Потім відкривається запірний клапан на Q2. Одночасно починає діяти випускний фільтр на Q1. Він повинен бути включений протягом усього процесу заповнення. Тепер вапно або цемент може накачуватися в бункер. Якщо бункер повний, то це сигналізується через граничний вимикач рівня заповнення на I3. Звуковий сигнал повідомляє операторові, що до автоматичного завершення процесу ще залишилося 99 секунд. Протягом цього часу повинен бути закритий клапан на вантажному автомобілі, щоб ще звільнявся від умісту завантажувальний шланг. Звуковий сигнал може бути достроково відключений за допомогою квитируючої кнопки на I6. Або він буде автоматично виключений через 25 секунд. Якщо

шланг не вдалося вчасно звільнити, то через кнопку на I5 можна виконати аварійне заповнення протягом 30 секунд.

Контроль надлишкового тиску в бункері також автоматично відключає процес заповнення. Це відображається за допомогою сигнальної лампи на Q4.

Використовувані компоненти

LOGO!.

I1 – Деблокуючий вимикач (замикаючий контакт).

I2 – Язичковий контакт завантажувального штуцера (замикаючий контакт)

I3 – Граничний вимикач рівня заповнення (замикаючий контакт).

I4 – Вимикач надлишкового тиску (розмикальний контакт).

I5 – Кнопка аварійного заповнення (замикаючий контакт)

I6 – Кнопка квитирювання звукового сигналу (замикаючий контакт).

Q1– Фільтр.

Q2– Запірний клапан.

Q3– Звуковий сигнал.

Q4– Світловий сигнал надлишкового тиску.

Варіант 22

Знищення збудників хвороб за допомогою фумігації

LOGO! використовується на птахофабриках для обробки яєць, призначених для вирощування курчат. Там LOGO! бере на себе фумігацію (окурювання газом) курячих яєць, щоб звільнити їх від збудників хвороб. Газ утворюється в газовій камері за допомогою електронагрівального приладу, що повинен перебувати в камері певний час. Потім він знову відсмоктується вентилятором.

Вирішення за допомогою LOGO!

Коротким натисканням на кнопку I1 починається процес фумігації. негайно починає діяти газогенератор на Q1. За допомогою затримки вимикання він вимикається через 10 хвилин. Час фумігації залежить від розміру камери. Тепер газ повинен певний час перебувати в приміщенні, щоб мати можливість знищити збудників хвороб. Через 10 хвилин включається вентилятор на Q2, щоб витягнути газ. Вентилятор також працює 10 хвилин, перш ніж він буде відключений за допомогою затримки вимикання. Через індикатор режиму роботи на Q3 сигналізується, що процес фумігації йде. Процес може бути зупинений у будь-який момент, якщо нажати I1 більше, ніж на 3 секунди. Виходи з Q1 по Q3 і всі таймери скидаються. Після кожного прогону й кожного переривання програма може бути знову запущена через I1.

Незалежно від програми через I2 можна окремо включати й виключати вентилятор.

Використовувані компоненти:

LOGO!.

I1 – Кнопка включення/виключення (замикаючий контакт).

I2 – Включення/вимикання вентилятора (замикаючий контакт).

Q1 – Газогенератор.

Q2 – Вентилятор.

Q3 – Індикатор режиму роботи.

2 ЛОГІЧНІ КОНТРОЛЕРИ SIMATIC S7-200

Програмувальні логічні контролери сімейства SIMATIC S7-200 застосовуються для побудови ефективних систем автоматичного керування при мінімальних витратах на придбання, встаткування й розробку системи. Контролери здатні працювати в реальному масштабі часу й можуть бути використані як для побудови вузлів локальної автоматики, так і систем розподіленого вводу-виводу з організацією обміну даними по PPI або MPI інтерфейсі, мережі PROFIBUS-DP, Industrial Ethernet або AS-інтерфейсу.

Відмінні риси сімейства SIMATIC S7-200:

- час виконання 1К логічних інструкцій не перевищує 0.37мс;
- наявність швидкісних лічильників зовнішніх подій;
- наявність швидкодіючих входів апаратних переривань;
- можливість нарощування кількості входів-виходів (за винятком систем на основі CPU 221);
- наявність імпульсних виходів (ШИМ або ЧІМ);
- потенціометри аналогового завдання цифрових параметрів;
- годинники реального часу (убудовані або встановлювані у вигляді знімного модуля);
- потужний набір інструкцій мови програмування;
- послідовний PPI-інтерфейс, що може бути використаний у якості вільно програмованого порту;
- дружня оболонка програмування STEP 7 Micro/Win;
- трирівневий парольний захист програм користувача;
- можливість роботи із пристроями людино-машинного інтерфейсу.

Усі моделі центральних процесорів S7-200 оснащені убудованим PPI (Point To Point Interface) інтерфейсом (RS 485), що може бути використаний для підключення програматора, пристроїв людино-машинного інтерфейсу, інших контролерів S7-200. Канали зв'язку виконуються крученою парою. У такій мережі один центральний процесор здатний підтримувати зв'язок з декількома центральними процесорами S7-200, текстовими дисплеями або панелями оператора. Швидкість передачі даних може досягати 187.5 Кбит/с.

Сімейство SIMATIC S7-200 поєднує у своєму складі 4 типи центральних процесорів, що відрізняються обсягами пам'яті, кількістю убудованих входів-виходів, набором убудованих функцій, можливостями розширення системи та широкий спектр модулів вводу-виводу дискретних і аналогових сигналів і комунікаційних модулів.

2.1 Знайомство з SIMATIC S7-200 CPU 221

CPU 221 найбільш простий і дешевий центральний процесор сімейства, призначений для побудови автономних вузлів локальної автоматики. Оснащений 6 убудованими дискретними входами й 4 дискретними виходами. CPU 221 має 2 виконання, що відрізняються напругою живлення й типами виходів.

Технічні дані

Пам'ять програм:

- обсяг 4Кбайт/ 1.3К інструкцій;
- тип – енергонезалежна, EEPROM.

Обсяг пам'яті даних 2048 слів.

Час збереження даних:

- типовий 50 годин (без буферної батареї); 200 днів (з буферною батареєю);
- мінімальний 8 годин при 40°C.

Мова програмування – STEP 7 Micro/Win.

Методи подання програм – STL, LAD і FBD.

Кількість рівнів вкладення підпрограм – до 64.

Кількість лічильників – 256.

Кількість прапорів – 256.

Кількість таймерів – 256.

Парольний захист програми – 3-рівневий.

Набір команд:

- основний: логічні операції, адресація результату, збереження, рахунок, завантаження, передача, порівняння, зрушення, обертання, виклик підпрограм з передачею параметрів.
- розширений: інструкції керування ШІМ і ЧІМ, інструкції переходів, циклів, перетворення типів даних, арифметичні інструкції додавання, вирахування, множення, добування квадратного кореня.

Час виконання логічної команди 0.37мкс

Убудовані функції:

- кількість входів переривань – 4, програмувальні на обробку наростаючого або спадаючого фронту;
- кількість швидкісних лічильників – 4, із вхідною частотою до 30кГц;
- імпульсні швидкісних виходи – 2, здатні формувати переривання, частота проходження вихідних сигналів до 20кГц, ШІМ або ЧІМ.

Система вводу-виводу:

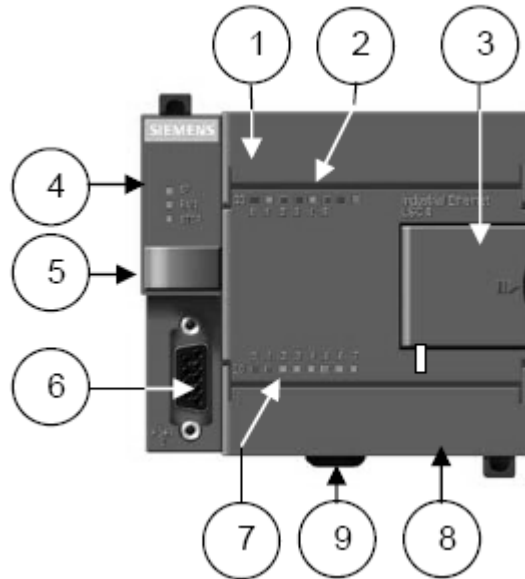
- убудованих входів – 6 дискретних входів, включаючи 4 входи переривань і 4 канали швидкісного рахунку;
- убудованих виходів – 4 дискретних виходи, включаючи 2 для виконання убудованих функцій.

У вхідних каскадах дискретних входів застосовуються транзисторні оптрони із зустрічно-паралельним включенням світлодіодів. Це забезпечує гальванічний поділ зовнішніх і внутрішніх кіл контролера, а також

дозволяє використовувати для живлення групи входів напругу будь-якої полярності.

У вихідних каскадах дискретних виходів гальванічний поділ зовнішніх і внутрішніх кіл забезпечується оптроном або електромагнітним реле.

Основні елементи CPU 221 показані на рисунку 2.1, а схема підключення на рисунку 2.2.



- 1– Захисна кришка термінального блоку вихідних кіл.
- 2– Світлодіоди індикації станів вихідних кіл.
- 3– Захисна кришка з'єднувача для підключення модулів введення-виводу.
- 4– Світлодіоди індикації режимів роботи центрального процесора.
- 5– Відсік для установки модуля EEPROM або модуля буферної батареї й годинника.
- 6– З'єднувач PPI інтерфейсу.
- 7– Світлодіоди індикації станів вхідних кіл.
- 8– Захисна кришка термінального блоку вхідних кіл.
- 9– Засувка фіксації на 35мм профільній шині DIN.

Рисунок 2.1 – Основні елементи CPU 221

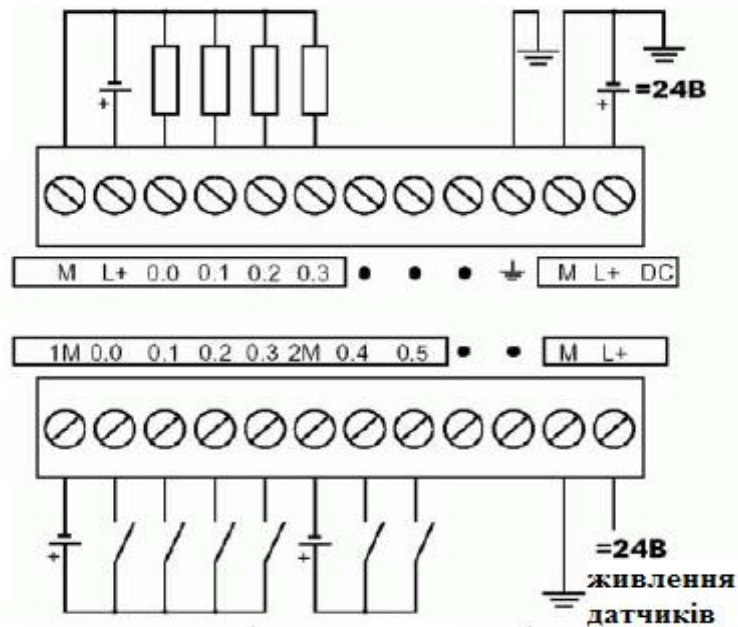


Рисунок 2.2 – Схема підключення CPU 221

S7-200 обробляє логіку керування у програмі циклічно, зчитуючи й записуючи дані і ставить програму у відповідність фізичним входам і виходам.

Основний принцип дії S7-200 дуже простий:

- S7-200 зчитує стан входів.
- Програма використовує ці входи для аналізу логіки керування.
- Під час обробки програми S7-200 оновлює дані.
- S7-200 записує дані на виходи.

На рисунку 2.3 показаний зв'язок між простою комутаційною схемою й S7-200. У цьому прикладі стан вимикача для запуску двигуна логічно пов'язаний зі станами інших входів. Оцінки цих станів визначають потім сигнальний стан виходу для виконавчого пристрою, що запускає двигун.

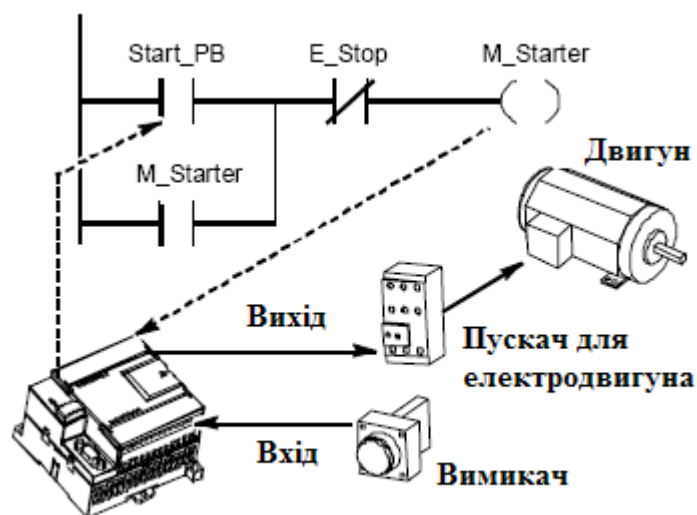


Рисунок 2.3 – Керування входами й виходами

S7-200 виконує послідовність завдань в циклі, неодноразово. Ця регулярна обробка завдань називається циклом. Як показано на рисунку 2.4, S7-200 виконує в циклі більшість або всі з наступних завдань:

1 Читання входів: S7-200 копіює стан фізичних входів у регістр входів образу процесу.

2 Виконання логіки керування в програмі: S7-200 виконує команди програми й зберігає значення даних в різних чарунках пам'яті.

3 Обробка запитів на обмін даними: S7-200 виконує всі завдання, необхідні для обміну даними.

4 Самодіагностика CPU: S7-200 перевіряє, щоб убудоване програмне забезпечення, програмна пам'ять і всі модулі розширення працювали належним чином.

5 Запис у виходи: значення, що зберігаються в регістрі виходів образу процесу, записуються у фізичні виходи.

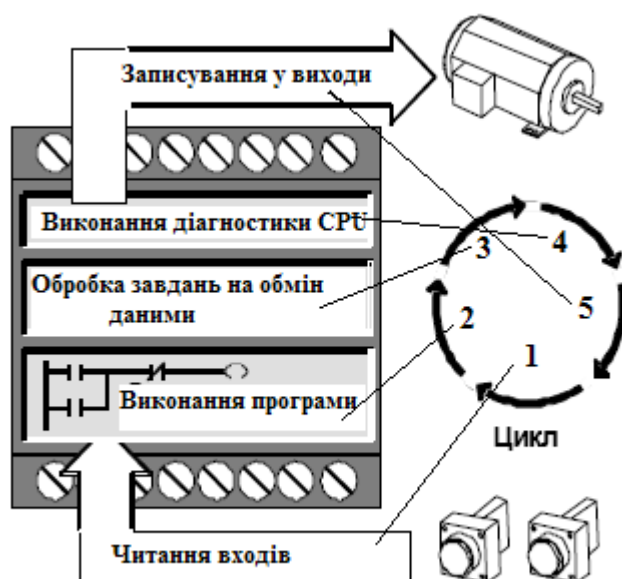


Рисунок 2.4 – Цикл S7-200

2.2 Програмування S7-200

S7-200 послідовно обробляє програму для керування завданням або процесом. Програма створюється за допомогою STEP 7-Micro/WIN, а потім завантажується в S7-200. STEP 7-Micro/WIN надає у ваше розпорядження різні інструментальні засоби й функції для розробки, реалізації й тестування програми.

2.2.1 STEP 7-Micro/WIN

Для відкриття STEP 7-Micro/WIN двічі клацніть на символі STEP 7-Micro/WIN або виберіть команду меню Пуск > SIMATIC >STEP 7 MicroWIN 32 .

Відкриється вікно проекту в STEP 7-Micro/WIN, показане на рисунку 2.5, яке надає зручну робочу область для створення програми керування.

На панелях інструментів є кнопки для часто використовуваних команд меню. Ви можете ці панелі інструментів окремо показувати або приховувати.

Навігаційна панель пропонує групи символів для доступу до різних функцій програмування.

Дерево команд відображає всі об'єкти проекту й команди, необхідні для створення програми керування. Ви можете буксирувати окремі команди з цього дерева у свою програму або вставляти команду подвійним щикликом у поточне положення курсору в редакторі програм.

Редактор програм містить логікові програми й таблицю локальних змінних, у якій ви можете призначити символічні імена для тимчасових локальних змінних.

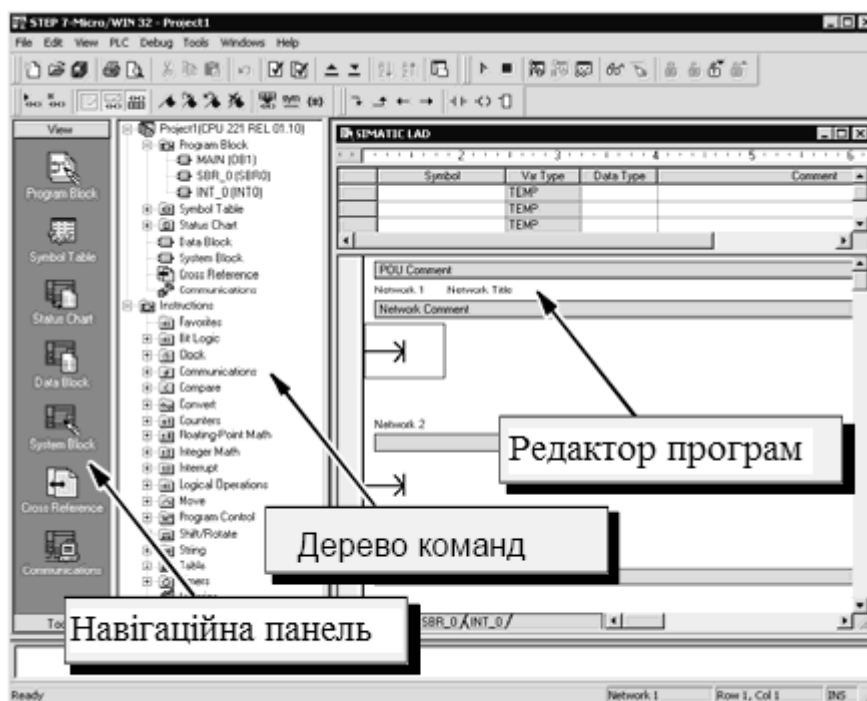


Рисунок 2.5 – Вікно проекту в STEP 7-Micro/WIN

Підпрограми й програми обробки переривань з'являються як закладки в нижній частині вікна редактора програм. Для переміщення між підпрограмами, програмами обробки переривань і основною програмою клацайте по цих закладках.

STEP 7-Micro/WIN має у своєму розпорядженні три редактори для створення програм: ланцюгова логічна схема (LAD), називана також контактним планом (KOP), список команд (STL або AWL) і функціональна блок-схема (FBD), називана також функціональним планом (FUP). З деякими обмеженнями програми, написані в кожному із цих редакторів програм, можуть відображатися й редагуватися за допомогою інших редакторів програм.

Функції редактора STL

Редактор STL відображає програму мовою, що має текстову основу і дає можливість створювати програми керування, уводячи мнемонічні позначення команд. У редакторі STL можна створювати програми, які неможливо створити в редакторі LAD або FBD. Це пояснюється тим, що, використовуючи STL, ви програмуєте на "рідній мові" S7-200, а не в графічному редакторі, у якому є деякі обмеження, щоб можна було правильно зображувати схеми з'єднань. Програмування в текстовому редакторі дуже схоже на програмування на мові Асемблера.

Приклад:

```
LD I0.0 //Прочитати вхід
```

```
A I0.1 //Виконати логічне "І"-
```

```
    //сполучення з іншим входом
```

```
= Q1.0 //Записати значення на вихід 1
```

S7-200 виконує команди в порядку, обумовленому програмою, зверху вниз, а потім починає спочатку. У STL логіка керування реалізується за допомогою логічного стека. У STL ви повинні вводити команди для обробки стекових операцій.

При виборі редактора STL візьміть до уваги наступне:

- STL найкраще пасує досвідченим програмістам.
- STL іноді дозволяє вирішувати проблеми, які ви не можете досить легко вирішити за допомогою редактора LAD або FBD.
- Ви можете використати редактор STL тільки із системою команд SIMATIC.
- Тоді як ви завжди можете використати редактор STL для перегляду або редагування програми, створеної за допомогою редактора LAD або FBD, зворотне не завжди можливо. Ви не завжди можете використати редактор LAD або FBD для відображення програми, написаної за допомогою редактора STL.

Функції редактора LAD

Редактор ланцюгових логічних схем LAD відображає програму у вигляді графічного подання, що має подібність із електричною монтажною схемою. Ланцюгові логічні схеми дозволяють програмі імітувати протікання електричного струму від джерела напруги через ряд логічних умов на входах, які, у свою чергу, активізують логічні умови на виходах.

LAD-програма містить у собі шину, яка розташована ліворуч, що перебуває під напругою, і є джерелом потоку сигналу. Замкнуті контакти дозволяють потоку сигналу протікати через ці контакти до наступного елемента, а розімкнуті контакти перешкоджають протіканню потоку сигналу.

Логіка підрозділяється на сегменти. Програма виконується сегмент за сегментом зліва направо і зверху вниз. На рисунку 2.6 показаний приклад програми у вигляді ланцюгової логічної схеми і основні її елементи.

Різні команди представляються графічними символами, що мають три основні форми:

1 Контакти – представляють логічні стани входів, наприклад, вимикачів, кнопок або внутрішніх умов.

2 Котушки – звичайно представляють логічні результати виходів, наприклад, ламп, пускачів електродвигунів, проміжних реле або внутрішніх вихідних умов.

3 Блоки представляють додаткові команди, наприклад, таймери, лічильники або математичні команди.

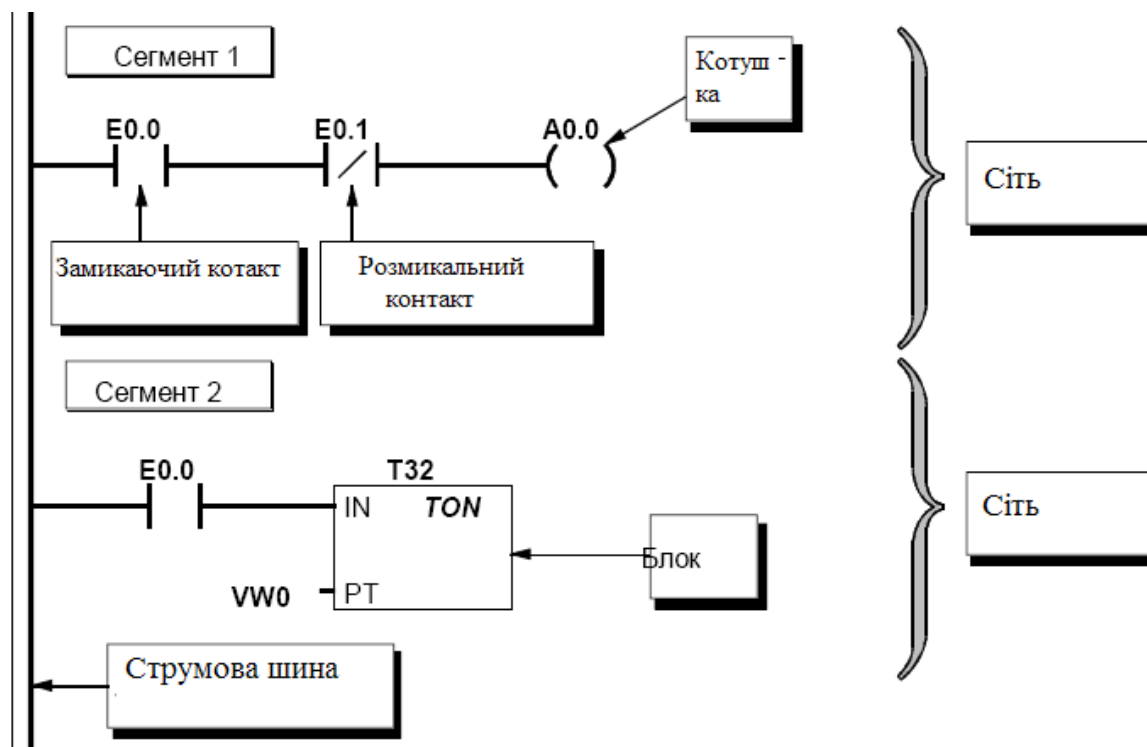


Рисунок 2.6 – Приклад програми у вигляді ланцюгової логічної схеми

При виборі редактора LAD візьміть до уваги наступне:

- Ланцюгова логічна схема проста у використанні для починаючих програмістів.
- Графічне подання легко розуміється й популярно в усьому світі.

- Редактор LAD можна використати й із системою команд SIMATIC, і із системою команд IEC 1131-3.

- Для відображення програми, створеної за допомогою редактора SIMATIC LAD, завжди можна використати редактор STL.

Функції редактора FBD

Редактор функціонального плану FBD відображає програму у вигляді графічного подання, що нагадує звичайні логічні схеми. Немає ніяких контактів і котушок, як у редакторі LAD, але є еквівалентні команди, подані у виді блоків. Ці блоки нагадують команди LO GO!.

FBD не використовує поняття лівої й правої струмових шин; тому поняття «потік сигналу» виражає аналогічне поняття потоку керування через логічні блоки FBD.

Із цієї причини шлях стану «1» через елементи FBD називається потоком сигналу. Походження потоку сигналу й місце призначення його виходу ставляться у відповідність безпосередньо операнду. Логіка програми впливає зі зв'язків між функціональними блоками, що позначають команди. Вихід однієї команди (наприклад, блок логічного І (AND)) може бути використаний для розблокування іншої команди (наприклад, таймера), формуючи необхідну логіку керування. Ця концепція дозволяє вирішувати широкий спектр завдань керування.

При виборі редактора FBD візьміть до уваги наступне:

- Графічне подання функціонального плану добре відбиває процес виконання програми.

- Редактор FBD можна використати й із системою команд SIMATIC, і із системою команд IEC 1131-3.

- Для відображення програми, створеної за допомогою редактора SIMATIC FBD, завжди можна використати редактор STL.

Структура програми для CPU S7-200 показана на рисунку 2.7 і складається із трьох основних елементів: головна програма, підпрограми (необов'язкові) і підпрограми обробки переривань (необов'язкові).

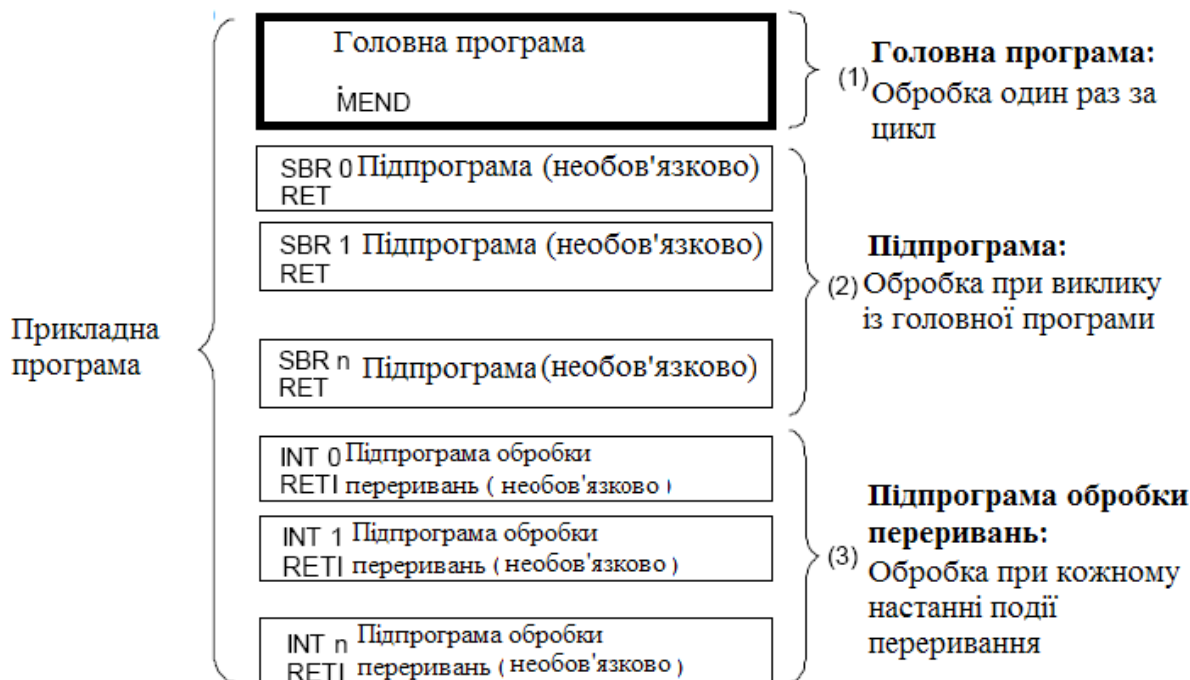


Рисунок 2.7 – Структура програми для CPU S7-200

Підпрограми й програми обробки переривань розташовуються за катушкою абсолютного завершення в LAD або за операцією MEND в STL у головній програмі.

S7-200 зберігає інформацію в різних місцях пам'яті, які мають однозначні адреси. Ви можете явно вказати адресу в пам'яті, до якої ви хочете звернутися. Завдяки цьому програма має прямий доступ до інформації. Таблиця 2.1 показує діапазон цілих значень, які можуть бути подані за допомогою даних різної довжини.

Для звертання до біта в деякій чарунці пам'яті ви повинні вказати адресу біта. Ця адреса складається з ідентифікатора чарунки пам'яті, адреси байта й номера біта.

Таблиця 2.1 – Десяткові й шістнадцятиричні діапазони для даних різної довжини

Подання	Байт (B)	Слово (W)	Подвійне слово (D)
Ціле без знака	Від 0 до 255 Від 0 до FF	Від 0 до 65 535 Від 0 до FFFF	Від 0 до 4 294 967 295 Від 0 до FFFF FFFF
Ціле зі знаком	Від -128 до +127 Від 80 до 7F	Від -32 768 до +32 767 Від 8000 до 7FFF	Від -2 147 483 648 до +2 147 483 647 Від 8000 0000 до 7FFF FFFF

Дійсне IEEE 32-бітове з плаваючої крапкою	Незастосовно	Незастосовно	Від +1.175495E-38 до +3.402823E+38 (позитивне) Від -1.175495E-38 до -3.402823E+38 (негативне)
---	--------------	--------------	--

На рисунку 2.8 показаний приклад звертання до біта (адресація у форматі «байт.біт»).

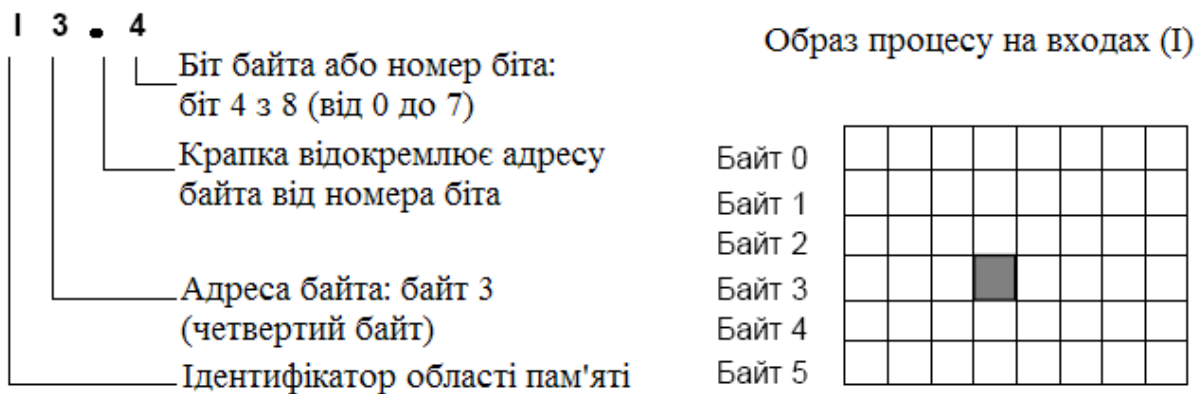


Рисунок 2.8 – Адресація байт.біт

У цьому прикладі за чарункою пам'яті й адресою байта (I = input [вхід], 3 = байт 3) необхідна крапка («.»), щоб відокремити адресу біта (біт 4).

Коли ви застосовуєте формат байт.біт, ви можете звернутися до даних у більшості чарунк пам'яті (V, I, Q, M, S, L і SM) як до байтів, слів або подвійних слів. Якщо ви хочете звернутися до байта, слова або подвійного слова даних у пам'яті, то ви повинні вказати ці адреси подібно адресі біта. Ви вказуєте ідентифікатор чарунки, позначення довжини даних і початкову адресу байта, слова або подвійного слова, як показано на рисунку 2.9.

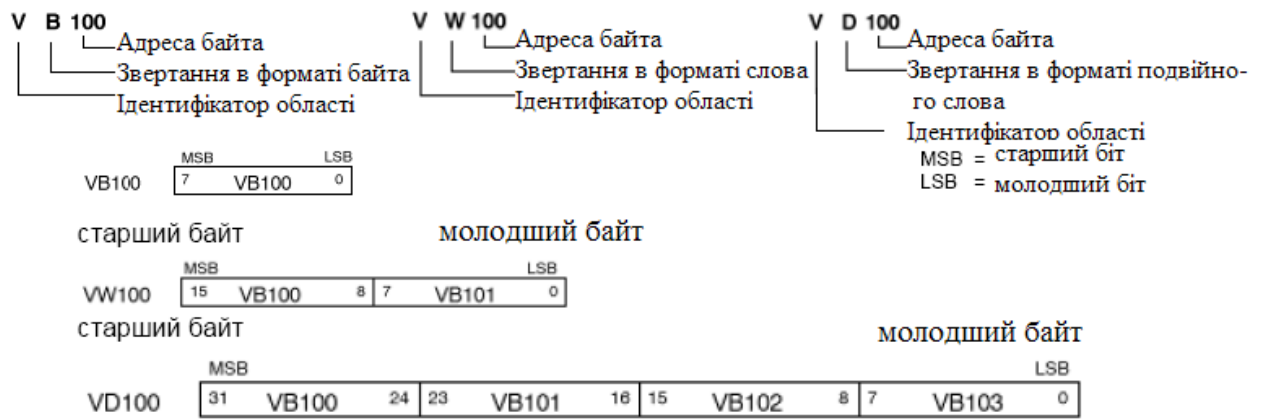


Рисунок 2.9 – Звертання до тієї ж самої адреси у форматі байта, слова й подвійного слова

До даних в інших чарунках пам'яті (напр., T, C, HC і акумулятори) ви звертаєтесь, указуючи як адресу ідентифікатор чарунки й номер елемента.

Існують наступні області пам'яті:

Регістр входів образу процесу: I

На початку кожного циклу S7-200 опитує фізичні входи й записує отримані значення в регістр входів образу процесу. До образу процесу можна звернутися у форматі біта, байта, слова й подвійного слова:

Біт: I[адреса байта].[адреса біта] – I0.1

Байт, слово або подвійне слово: I[довжина][початкова адреса байта] – IB4

Регістр виходів образу процесу: Q

Наприкінці циклу S7-200 копіює значення, що зберігаються в регістрі виходів образи процесу, в фізичні виходи. До образу процесу можна звернутися у форматі біта, байта, слова й подвійного слова:

Біт: Q[адреса байта].[адреса біта] – Q1.1

Байт, слово або подвійне слово: Q[довжина][початкова адреса байта] – QB5

Чарунка пам'яті змінних: V

Пам'ять змінних можна використати для зберігання проміжних результатів операцій, виконуваних у вашій програмі. У пам'яті змінних ви можете зберігати також інші дані, що мають відношення до процесу або до рішення вашого завдання автоматизації. До пам'яті змінних можна звернутися у форматі біта, байта, слова й подвійного слова.

Чарунка бітової пам'яті: M

Біти пам'яті (маркери) можна використати як керуючі реле для зберігання проміжних результатів операцій або іншої керуючої інформації. До бітів пам'яті можна звернутися у форматі біта, байта, слова й подвійного слова.

2.2.2 Приклад програмування

Електрична схема прикладу показана на рисунку 2.10.

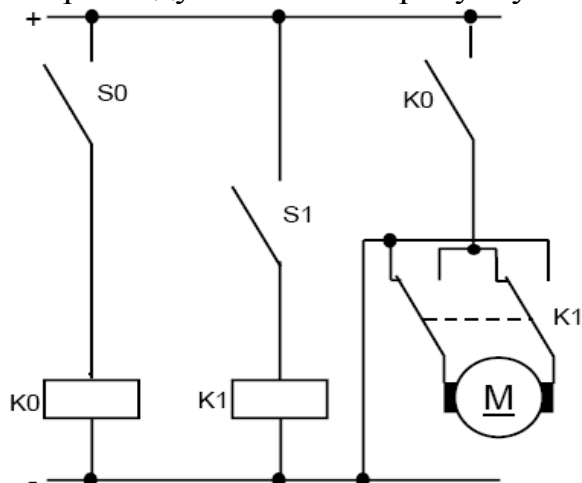


Рисунок 2.10 – Схема керування двигуном постійного струму

Ключ S0 використається для включення мотора. Ключ S1 використається для керування напрямком обертання мотора.

Схема навчального стенду наведена на рисунку 2.11.

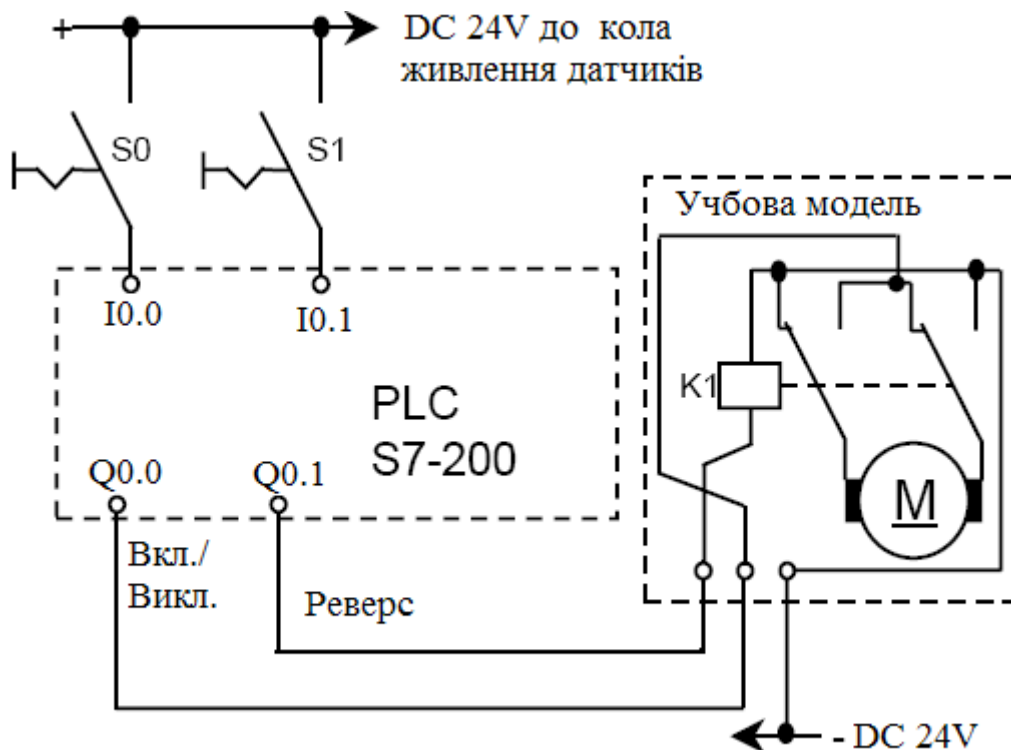


Рисунок 2.11 – Схема навчального стенду S 7-200

У навчальному комплекті перемикачі S0 і S1 перебувають на симуляторі. Симулятор підключає 24 V до входів I0.0 і I0.1. Навчальна модель підключена до виходів контролера Q0.0 (Мотор вкл./викл.) і Q0.1

(Керування реверсом). Стан сигналу входу І0.0 зв'язано в програмі з виходом Q0.0. Стан сигналу входу І0.1 пов'язаний з виходом Q0.1.

На рисунку 2.12 показано перехід від електричної схеми до програми в редакторі LAD.

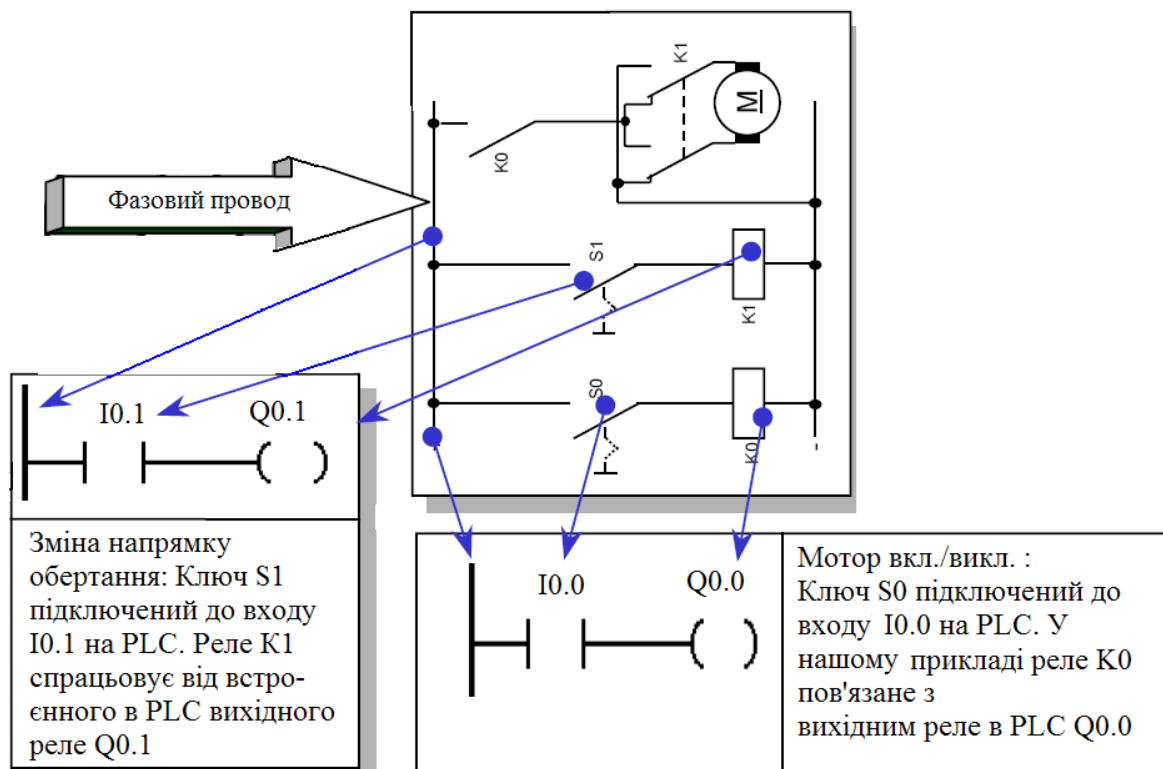


Рисунок 2.12 – Перехід від схеми до програми

Схемна логіка (наприклад, реле часу або тригери), що використовується як інструмент, записується тепер разом з перемикачами, контактами й керуючими реле і управляється від PLC. Елементи керування, такі як перемикачі входів, перемикачі напрямків і т.п., зі сторони входу й силові ключі (такі як контакти мотора, перемикачі реверса, і т.п.) зі сторони виходу не можуть бути описані в PLC.

Структура PLC-програми, поданої в контактному плані мовою LAD, (КОР) показана на рисунку 2.13. Цей тип подання найбільше близько відповідає електричній схемі. На рисунку 2.13 детально розібрані основні елементи програми.

На рисунку 2.14 наведена структура елемента LAD-програми.

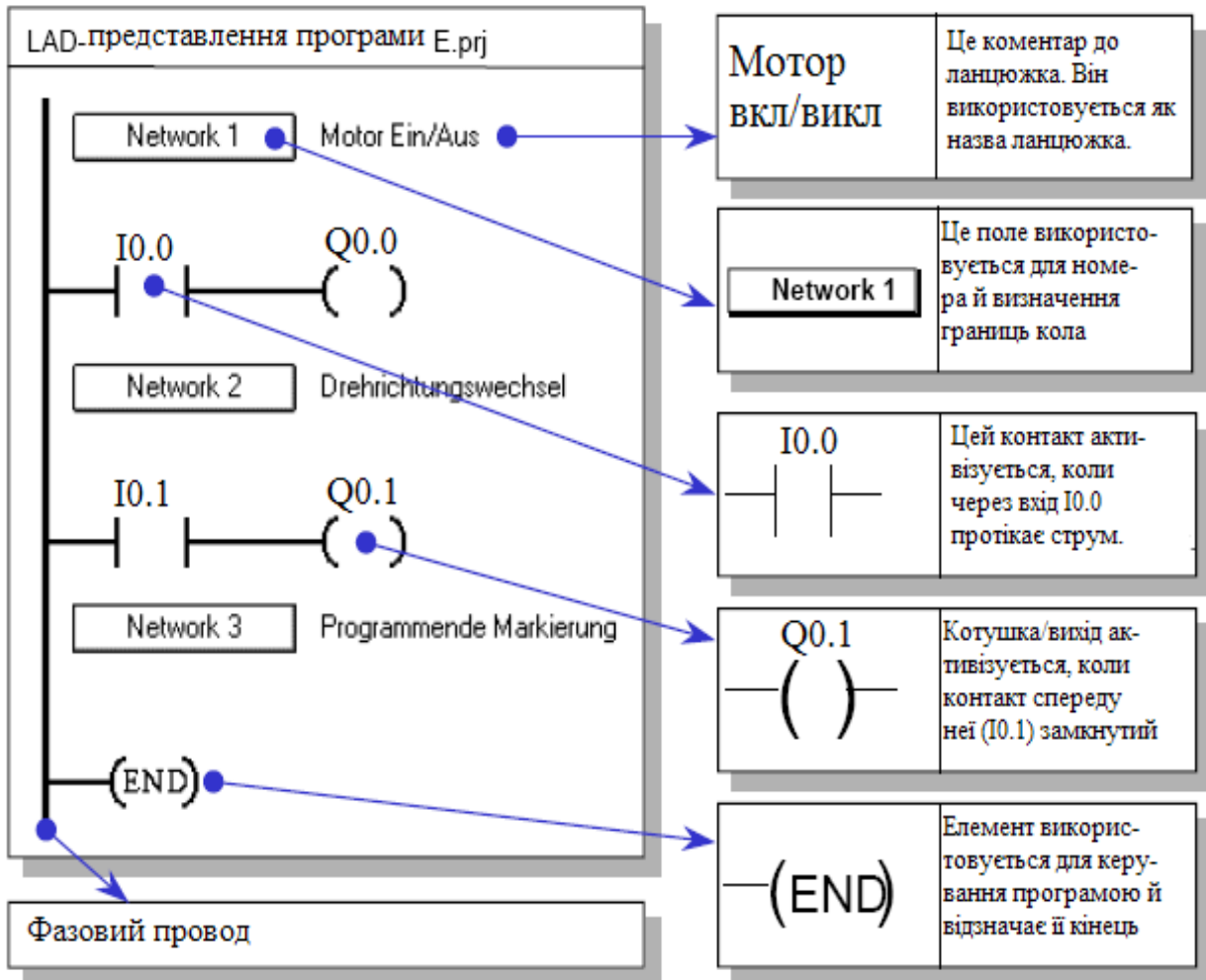


Рисунок 2.13 – Програма в редакторі LAD

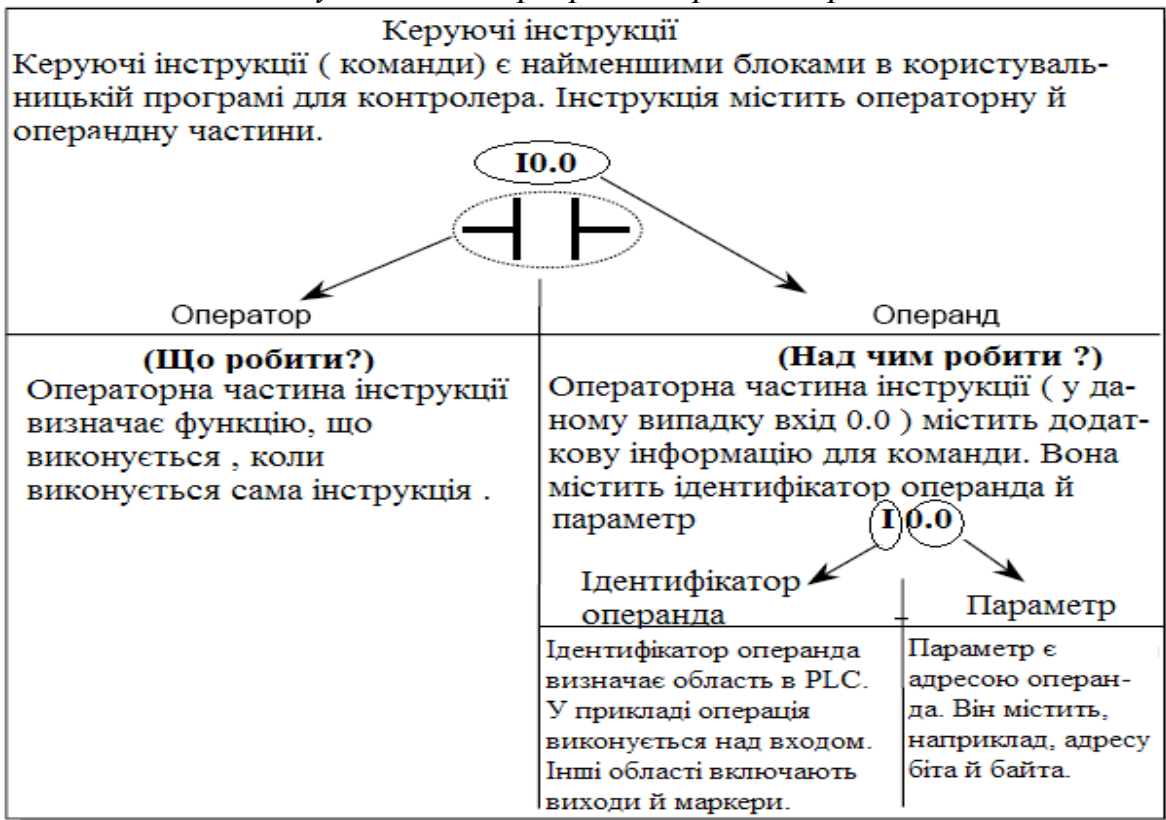


Рисунок 2.14 – Структура елемента LAD-програми

Тепер необхідно увести свою програму в комп'ютер, для цього необхідно мати контейнер, щоб "покласти" туди файл програми й пов'язані з ним файли. В STEP 7-Micro/WIN цей контейнер називається проектом. S7-200-проект містить програмний файл, наприклад, *.OB1 і файл для символів *.sym. S7-200- проект є також приймачем додаткових файлів, пов'язаних з програмою. Послідовність створення нового проекту показана на рисунку 2.15.

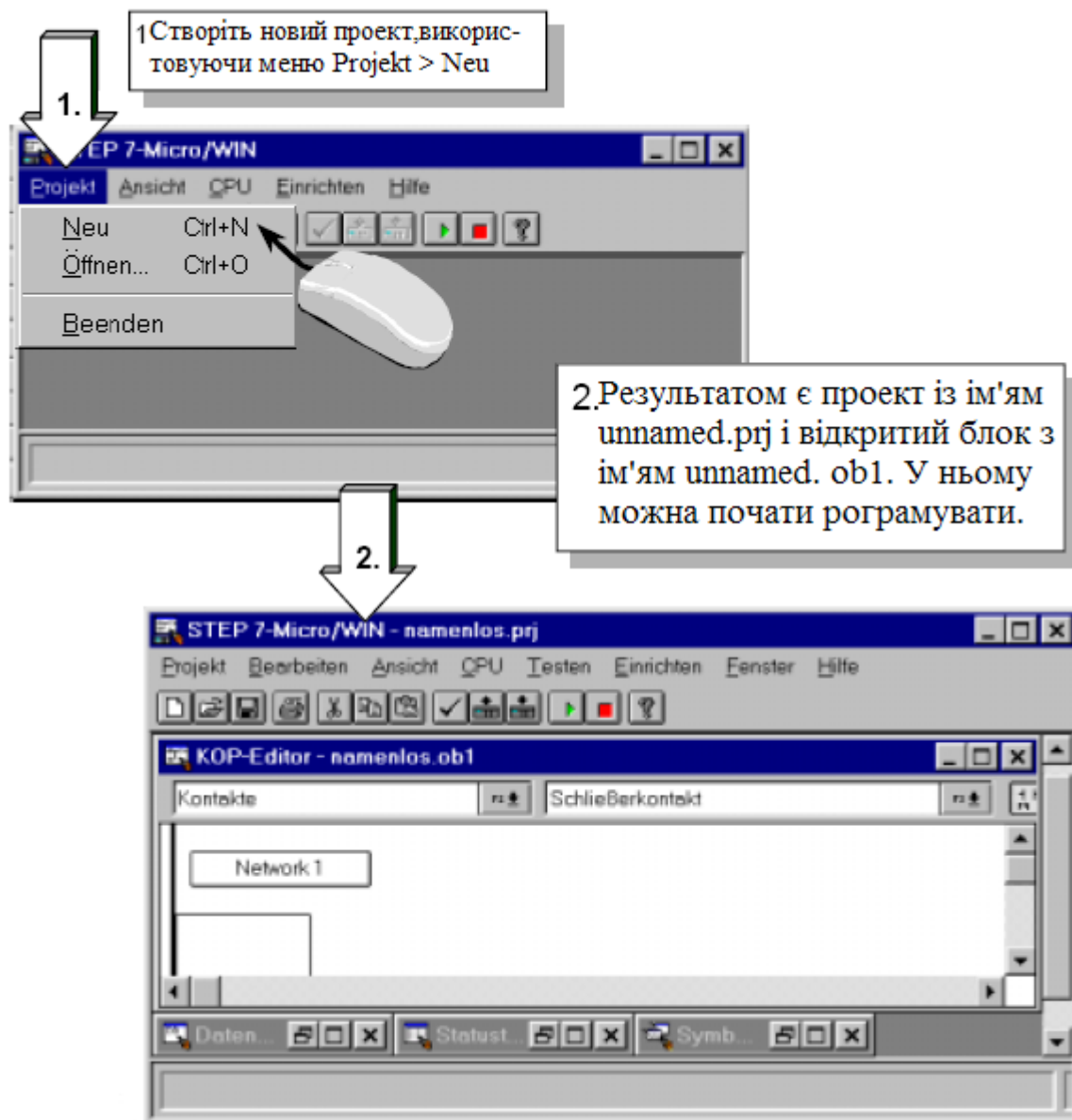


Рисунок 2.15 – Послідовність створення нового проекту

Збережіть проект Namenlos.prj відразу, після або під час програмування під ім'ям, що що-небудь означає для вас. За допомогою меню Projekt > Save as...

Для тестування програми виберіть розділ меню Debug > Ladder Status On для активації режиму спостереження за станом програми в LAD-виді. Ви можете тепер спостерігати стан операндів у контролері.

У прикладі контакт S0 з'єднаний з входом 0.0. Якщо ви активуєте ключ і будете спостерігати за LAD-програмою в режимі статусу, то операції, які пропускають струм, тобто істинні (=“1”), заповнюються синіми кольорами як це показано на рисунку 2.16.

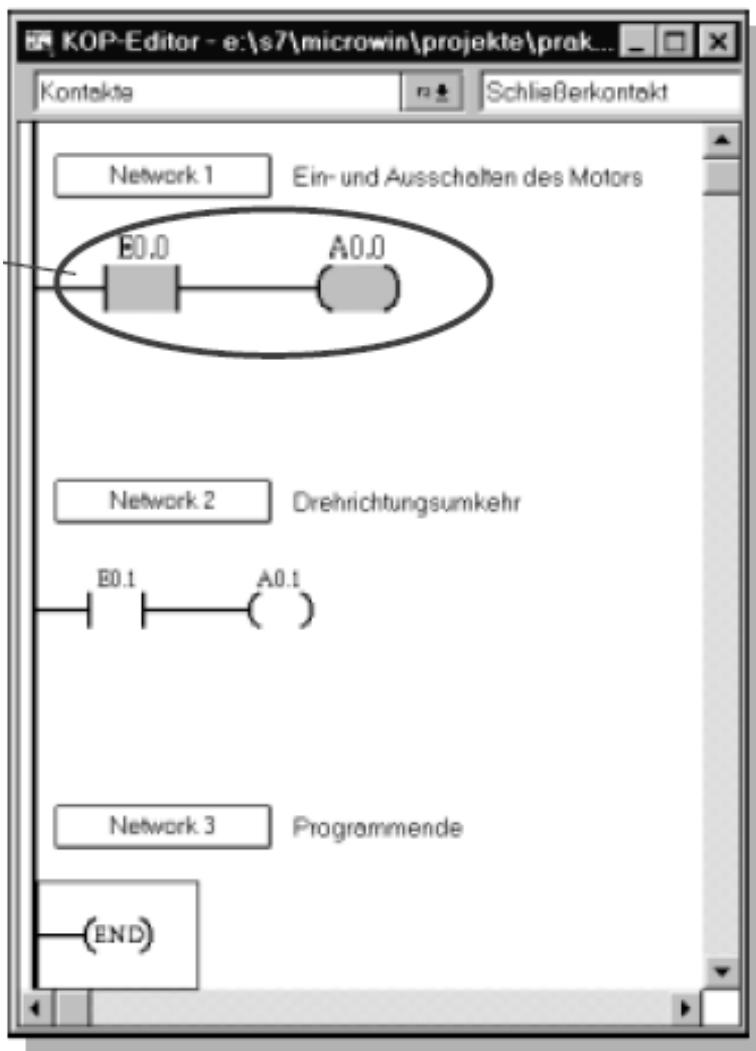


Рисунок 2.16 – Вікно редактора LAD в режимі тестування програми

Статус операцій циклічно читається з контролера й оновлюється на екрані. У цьому контексті, засоби Online-зв'язку дозволяють глянути на контролер зсередини за допомогою PC/PG, спостерігаючи за поточним станом і циклічними змінами операндів. Дуже швидкі події не можуть бути відслідковані тому, що час, необхідний для передачі вихідних даних на екран, досить великий.

2.3 Завдання для самостійної роботи

Необхідно спроектувати систему автоматичного керування для заданого варіанта.

Проект повинен складатися з електричної схеми пристрою, блок-схеми алгоритму функціонування, програми для контролера та інструкції для користувача.

Варіант 1. У термопечь ТВЧ з транспортера поступають деталі двох типів. Режим обробки деталей відрізняється за часом. Тип деталі визначається за її конфігурацією під час надходження до печі. Транспортер не працює поки працює піч. Нагрів включається під час надходження деталі.

Варіант 2. Температура води в робочій машині регулюється шляхом релейної зміни витрати через теплообмінник. При зростанні температури вище 90° витрата збільшується шляхом відкриття додаткового клапана. При зменшенні температури нижче 90° цей клапан закривається через 1 хвилину. Включення і виключення насоса проводиться кнопками.

Варіант 3. У термопечі ТВЧ обробляються 3 види деталей. Режим обробки вибирається перемикачем. Включення нагріву проводиться кнопкою, виключення – за часом, залежно від типу деталі. Після виключення нагріву на 30 секунд включається інтенсивне повітряне охолодження, потім протягом 3 секунд звучить сигнал, що дозволяє відкриття печі.

Варіант 4. Вантажна платформа перевозить вантаж з точки А в точку В або точку С. Управління платформою ручне, з пульта, встановленого в точці А. Вибір кінцевого пункту за допомогою перемикача. Якщо протягом 5 хвилин після прибуття вантаж не знятий з платформи, відбувається автоматичне повернення в точку А. Якщо вантаж знятий протягом вказаного часу, то відправка платформи в точку А відбувається через 1 хвилину після зняття вантажу. У будь-якому випадку перед відправленням протягом 5 сек. лунає дзвінок.

Варіант 5. Ліфт обслуговує 2 поверхи. Управління ручне. У середині кабіни – кнопки з номерами поверхів, на поверхах – кнопки виклику. При перевантаженні ліфт не рушає з місця і спалахує сигнальна лампа.

Варіант 6. Робоча машина має три приводи. Другий привід не можна включити, поки не включений перший, третій не можна включити, поки не включений другий. Виключення проводиться від однієї кнопки

"СТОП", причому спочатку вимикається перший привід, через 30 секунд – другий, ще через 30 секунд – третій.

Варіант 7. Температура повітря в приміщенні регулюється шляхом включення і виключення додаткового вентилятора. Вентилятор повинен включатися, якщо температура перевищує 45°. Виключення вентилятора відбувається при температурі 30° після витримки 1 хвилину.

Варіант 8. Рольганг з електроприводом подає сталевий лист до ножиць-гільйотин. За ножицями встановлений упор з кінцевим вимикачем, що подає сигнал на зупинку рольганга і відрізування листа. Далі подача листа повторюється. Включення приводу рольганга відбувається при виявленні листа на першому ролику, виключення – від кнопки або за відсутності листа перед гільйотиною протягом 30 с.

Варіант 9. Рольганг з електроприводом подає деталі двох типів на кантователь, який деталь першого типу перевертає ліворуч, а деталь другого типу – праворуч. Деталі завжди подаються по черзі. Включення приводу рольганга відбувається при попаданні чергової деталі на перший ролик, виключення – при попаданні деталі в кантователь. Схема повинна управляти рольгангом і кантователем з дотриманням необхідних блокувань.

Варіант 10. Подача свердлувальної головки вниз проходить на двох швидкостях – спочатку на прискореній подачі, потім, за сигналом путьового датчика – на робочій подачі. Після закінчення свердління відбувається підйом вгору на прискореній подачі. Схема повинна мати кнопки "ПУСК" і "АВАР. СТОП".

Варіант 11. Рука маніпулятора здійснює наступний цикл: вперед, захоплювач затиснути, назад, праворуч, вперед, захоплювач розтиснути, вверх, ліворуч. Цикл починається, якщо є сигнал дозволу.

Варіант 12. Рука маніпулятора здійснює наступний цикл: вниз, захоплювач затиснути, вгору, вправо, вниз, захоплювач розтиснути, вгору, вліво. Цикл починається, якщо є сигнал дозволу.

Варіант 13. Рука маніпулятора здійснює наступний цикл: вперед, захоплювач затиснути, назад, праворуч, вниз, захоплювач розтиснути, вгору, ліворуч. Цикл починається, якщо є сигнал дозволу.

Варіант 14. Ворота цеху відкриваються при натисненні на кнопку і закриваються після витримки 50 секунд. При наявності якого-небудь предмету в створі воріт закриття припиняється, вони повертаються в

початкове положення, лунає дзвінок (5 сек.), і ворота залишаються відкритими до натиснення кнопки.

Варіант 15. Температура води, що проходить через теплообмінник, регулюється шляхом стрибкоподібної зміни витрати, для чого передбачені додаткові клапани, що збільшують загальний переріз теплообмінника. Перший клапан збільшує переріз удвічі, якщо температура перевищує 80° . Якщо через 3 хвилини температура не впала нижче 70° , другий клапан збільшує переріз втричі. Закриття клапанів відбувається при температурі 70° .

Варіант 16. Рольганг з електроприводом подає деталі двох типів на кантователь: деталь першого типу перевертає ліворуч, а деталь другого типу – праворуч. Деталі завжди йдуть в наступній послідовності: ААБААБ... Включення приводу рольганга відбувається при попаданні чергової деталі на перший ролик, виключення – при заході деталі в кантователь. Схема повинна управляти рольгангом і кантователем з дотриманням необхідних блокувань.

Варіант 17. Ліфт обслуговує 2 поверхи. Управління ручне. Всередині кабіни – кнопка "Пуск", на поверхах – кнопки виклику. При перевантаженні ліфт не рушає з місця, і спалахує сигнальна лампа.

Варіант 18. Вантажна платформа перевозить вантаж з точки А у точку В. Управління платформою ручне, з пульта, встановленого в точці А. Якщо протягом 5 хвилин після прибуття вантаж не знятий з платформи, відбувається автоматичне повернення в точку А. Якщо вантаж знятий протягом вказаного часу, то відправка платформи в точку А відбувається через 1 хвилину після зняття вантажу. Перед відправленням платформи протягом 5 сек. лунає дзвінок.

Варіант 19. У термопечі ТВЧ обробляються 4 види деталей. Режим обробки вибирається перемикачем. Включення нагріву проводиться кнопкою, виключення – за часом, залежно від типу деталі. Після виключення нагріву на 20 секунд включається інтенсивне повітряне охолодження, потім протягом 4 сек. лунає сигнал, що дозволяє відкривання печі.

Варіант 20. Ворота цеху відкриваються під час надходження сигналу від фотодатчика або при натисканні на кнопку і закриваються після витримки 30 секунд. За наявності якого-небудь предмету в створі воріт закриття припиняється, лунає дзвінок (5 сек.), і ворота залишаються в напіввідкритому положенні. Продовження закриття – тільки від кнопки.

Додаток А

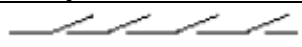
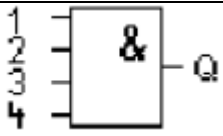
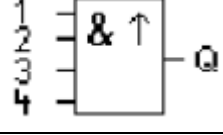
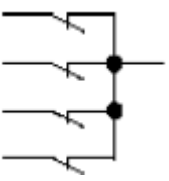
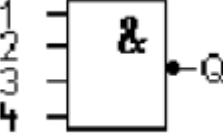
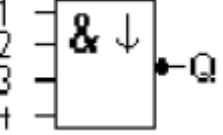
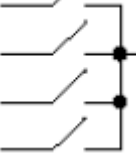
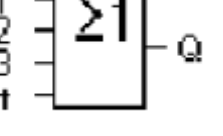
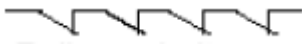
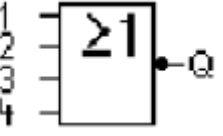
Функції LOGO!

LOGO! у режимі програмування надає у ваше розпорядження різні елементи, ці елементи поділяються на списки. Цими списками є:

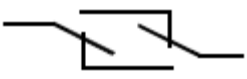
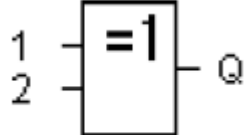

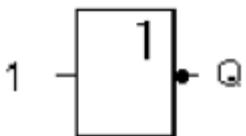
- ↓Co: список елементів з'єднання (Connector),
- ↓GF: список основних функцій AND [І], OR [АБО],
- ↓SF: список спеціальних функцій ,
- ↓BN: список готових до використання в комутаційній програмі блоків.

Список основних функцій – GF

Таблиця А.1

Подання на комутаційній схемі	Подання в LOGO!	Найменування основної функції
 Послідовне з'єднання замикаючих контактів		AND (І)
		AND з аналізом фронту
 Паралельне з'єднання розмикаючих контактів		NAND (І-НІ)
		NAND з аналізом фронту
 Паралельне з'єднання замикаючих контактів		OR (АБО)
 Послідовне з'єднання розмикальних контактів		NOR (АБО-НІ)

Продовження таблиці А.1

Подання на комутаційній схемі	Подання в LOGO!	Найменування основної функції
 Подвійний перекидний контакт		XOR (виключаючи АБО)
 Розмикальний контакт		NOT (заперечення, інверсія)

Список спеціальних функцій – SF

Опис елементів з'єднання, які ви можете використати для створення логічного зв'язку з іншими блоками або із входами пристрою LOGO!.

- **S (Set = установити):**
Сигнал на вході S установлює на виході логічну «1».
 - **R (Reset = скинути):**
Вхід скидання R має пріоритет над всіма іншими входами й перемикає виходи в "0".
 - **Trg (Trigger= запустити):**
Цей вхід використовується для запуску функції на виконання.
 - **Cnt (Count = рахувати):**
Цей вхід використовується для рахунку імпульсів.
 - **Fre (Frequency = частота):**
До входу із цим описом прикладаються частотні сигнали, що підлягають аналізу.
 - **Dir (Direction = напрямок):**
Цей вхід використовується, наприклад, для установлення напрямку, в якому повинен рахувати лічильник.
 - **En (Enable = дозволити):**
Цей вхід розблокує функцію, виконувану блоком. Якщо на цьому вході "0", то інші сигнали блоком ігноруються.
 - **Inv (Invert = інвертувати):**
Вихідний сигнал блоку інвертується, коли цей вхід активізований.
 - **Ral (Reset all = скинути все):**
Скидаються всі внутрішні значення.
- Є кілька входів, до яких ви не прикладаєте сигнали, а параметризуєте відповідний блок певними значеннями. Приклади:
- **Par (Parameter = параметр):**
Цей вхід не підключається. Тут для блоку встановлюються параметри (часи, пороги включення й вимикання й т.д.).
 - **No (Nocken = шаблон):**

Цей вхід не підключається. Тут установлюється шаблон часу.

- **P (Priority = пріоритет):**

Цей вхід не підключається. Тут установлюються пріоритети, а також визначається, повинне чи повідомлення квітнутися в режимі RUN.

У таблиці А.1 зазначено, чи володіє відповідна функція параметруючою збереженістю (Rem).

Є наступні спеціальні функції (рис.А.1).

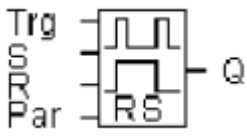
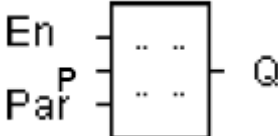
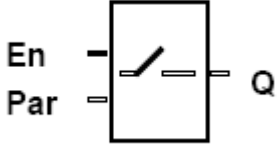
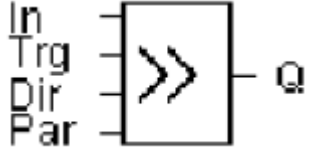
Таблиця А.2

Подання в LOGO!	Назва спеціальної функції	Rem
Таймери		
	Затримка включення	Rem
	Затримка вимикання	Rem
	Затримка включення/ вимикання	Rem
	Затримка включення із запам'ятовуванням	Rem
	Інтервальне реле часу (вивід імпульсу)	Rem
	Інтервальне реле часу, що запускається фронтом	Rem
	Асинхронний генератор імпульсів	Rem

Продовження таблиці А.2

Подання в LOGO!	Назва спеціальної функції	Rem
	Генератор випадкових імпульсів	
	Вимикач світла на сходовій клітці	Rem
	Двофункціональний вимикач	Rem
	Семиденний годинний вимикач	
	Дванадцятимісячний вартовий вимикач	
<i>Лічильники</i>		
	Реверсивний лічильник	Rem
	Лічильник робочого часу	Rem
	Граничний вимикач	
<i>Різне</i>		
	Реле, що самоблокується	Rem

Продовження таблиці А.2

Подання в LOGO!	Назва спеціальної функції	Rem
	Імпульсне реле	Rem
	Тексти повідомлень	
	Програмний вимикач	Rem
	Регістр зрушення	Rem

Додаток Б

Команди контролера S7-200

1 Бітові логічні операції

1.1 Контакти

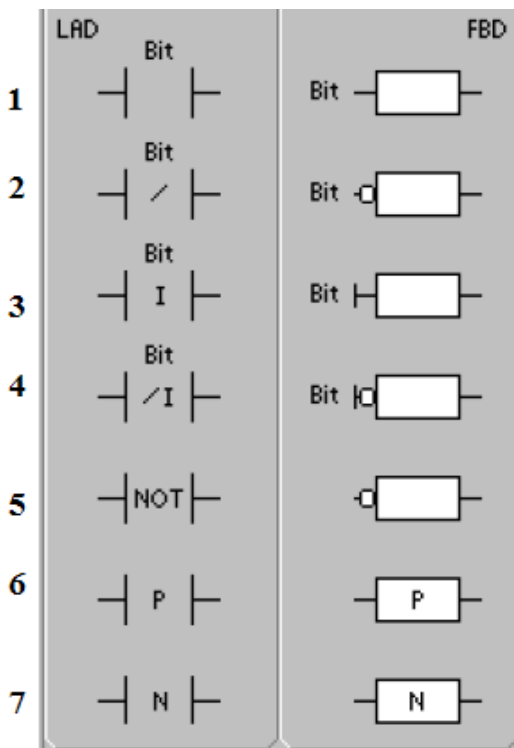


Рисунок Б.1

1.1.1 Стандартні контакти

Команди "Нормально відкритий контакт" (1) і "Нормально замкнутий контакт" (2) одержують вихідне значення з пам'яті або з регістра образу процесу. Стандартні контакти одержують вихідне значення з пам'яті (або з регістра образу процесу, якщо типом даних є I або Q). Нормально відкритий контакт замкнутий (включений), коли біт дорівнює 1, а нормально замкнутий контакт замкнутий (включений), коли біт дорівнює 0. В FBD до блоків "I" та "АБО" може бути підключено не більше 32 входів.

1.1.2 Безпосередньо керовані контакти

Безпосередньо керований контакт при своїй актуалізації не залежить від циклу S7-200, його значення оновлюється негайно. Команди "Безпосередньо керований нормально відкритий контакт" (3) і "Безпосередньо керований нормально замкнутий контакт" (4) при виконанні команди одержують значення фізичного входу, однак, регістр

образа процесу не обновляється. Безпосередньо керований нормально відкритий контакт замкнутий (включений), коли фізичний вхід (біт) перебуває в стані 1, а безпосередньо керований нормально замкнутий контакт замкнутий (включений), коли фізичний вхід (біт) перебуває в стані 0.

1.1.3 Команда NOT [NI]

Команда заперечення (NOT) (5) змінює стан входу потоку сигналу (тобто вона змінює значення у вершині стека з 0 на 1 або з 1 на 0).

1.1.4 Позитивний і негативний фронт

Контакт "Позитивний фронт" (6) пропускає потік сигналу протягом одного циклу при кожній появі позитивного фронту. Контакт "Негативний фронт" (7) пропускає потік сигналу протягом одного циклу з кожною появою негативного фронту.

Так як команди виявлення позитивного й негативного фронту вимагають перемикання, відповідно, з 0 на 1 або з 1 на 0, то немає можливості виявити наростаючий або падаючий фронт у першому циклі. У першому циклі стан зазначеного в цих командах біта встановлює S7-200. У наступних циклах ці команди самі можуть розпізнати фронти для зазначеного біта.

1.2 Котушки

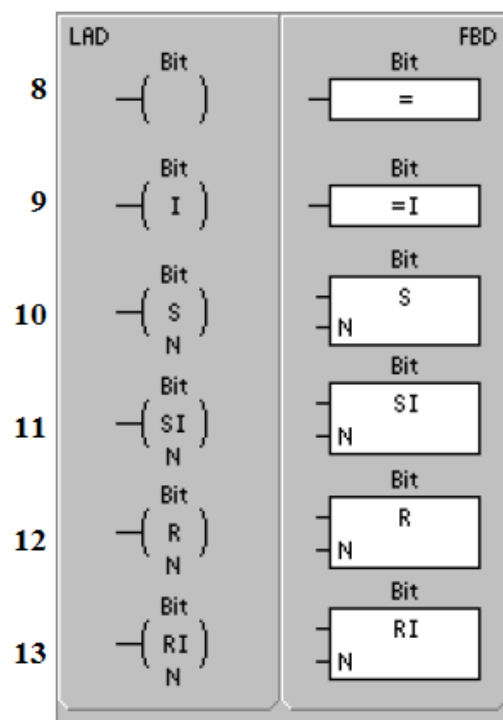


Рисунок Б.2

1.2.1 Присвоювання

Команда присвоювання (8) (=) записує нове значення для вихідного біта в регістр образу процесу. При виконанні команди присвоювання S7-200 установлює або скидає вихідний біт у регістрі образа процесу. В LAD і FBD зазначений бітустановлюється рівним потоку сигналу.

1.2.2 Безпосереднє присвоювання бітового значення

Команда безпосереднього присвоювання бітового значення (9) (=I) при своєму виконанні записує нове значення як у фізичний вихід, так і в образ процесу. Коли виконується команда безпосереднього присвоювання бітового значення, фізичний вихід (біт) негайно встановлюється відповідно до стану потоку сигналу. Символ “I” означає безпосередній доступ; при виконанні команди нове значення записується у фізичний вихід і у відповідний осередок регістра образу процесу. Тут є відмінність від інших видів доступу, які записують нове значення тільки в регістр образу процесу.

1.2.3 Установка й скидання

Команди установки (S) (10) і скидання (R) (12) установлюють (включають) або скидають (виключають) зазначену кількість входів або виходів (N), починаючи зі зазначеної адреси (біта). Ви можете встановити або скинути від 1 до 255 входів і виходів. Якщо команда скидання вказує на біт таймера (T) або лічильника (C), то команда скидає біт таймера або лічильника й стирає поточне значення таймера або лічильника.

1.2.4 Безпосередня установка й безпосереднє скидання

Команди безпосередньої установки (11) й безпосереднього скидання (13) безпосередньо установлюють (включають) або безпосередньо скидають (виключають) зазначену кількість входів або виходів (N), починаючи із зазначеної адреси (біта). Ви можете безпосередньо й негайно встановити або скинути від 1 до 128 входів і виходів. Символ “I” означає безпосередній доступ; при виконанні команди нове значення записується у фізичний вихід і у відповідний осередок регістра образу процесу. Тут є відмінність від інших видів доступу, які записують нове значення тільки в регістр образу процесу.

2 Функціональний блок із двома стійкими станами: перевага установлення й перевага скидання

Функціональний блок із двома стійкими станами й перевагою установки (14) являє собою тригер, у якого домінує установка. Якщо сигнал установки (S1) і сигнал скидання (R) одночасно набувають значення *істина*, то вихід (OUT) набуває значення *істина*.

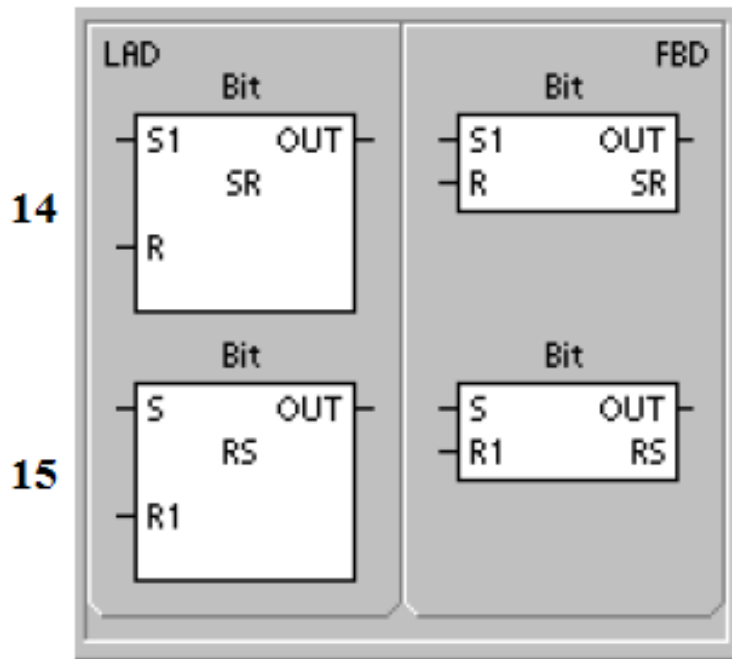


Рисунок Б.3

Функціональний блок із двома стійкими станами й перевагою скидання (15) являє собою тригер, в якого домінує скидання. Якщо сигнал установки (S) і сигнал скидання (R1) одночасно набувають значення *істина*, то вихід (OUT) набуває значення "0". Параметр Bit являє собою булевий параметр, який установлюється або скидається. Додатковий вихід відбиває сигнальний стан параметра Bit.

3 Порівняння числових величин

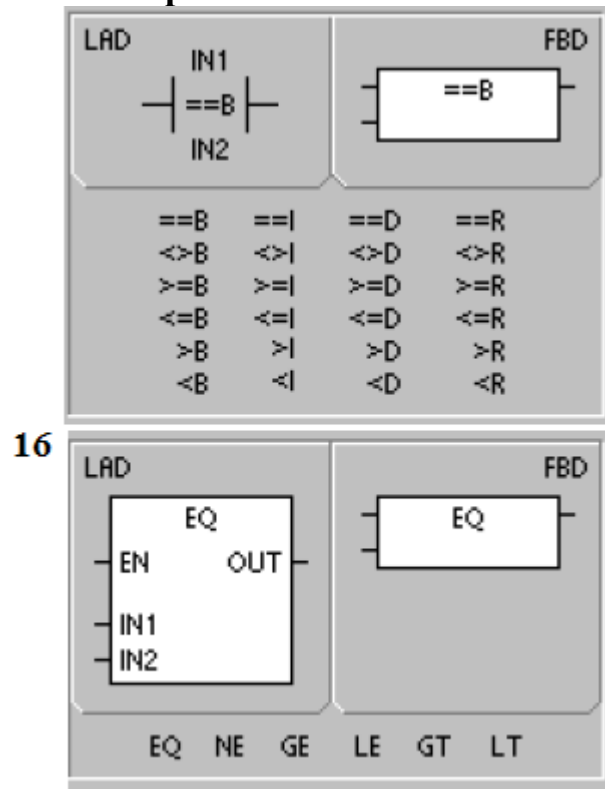


Рисунок Б.4

Команди порівняння (16) використовуються для порівняння двох величин:

IN1 = IN2 IN1 >= IN2 IN1 <= IN2

IN1 > IN2 IN1 < IN2 IN1 <> IN2

Операції порівняння байтів не враховують знака.

Операції порівняння цілих враховують знак.

Операції порівняння подвійних слів враховують знак.

Операції порівняння дійсних чисел враховують знак.

Для LAD і FBD: Якщо порівняння істинно, команда порівняння включає контакт (LAD) або вихід (FBD).

Лічильник SIMATIC

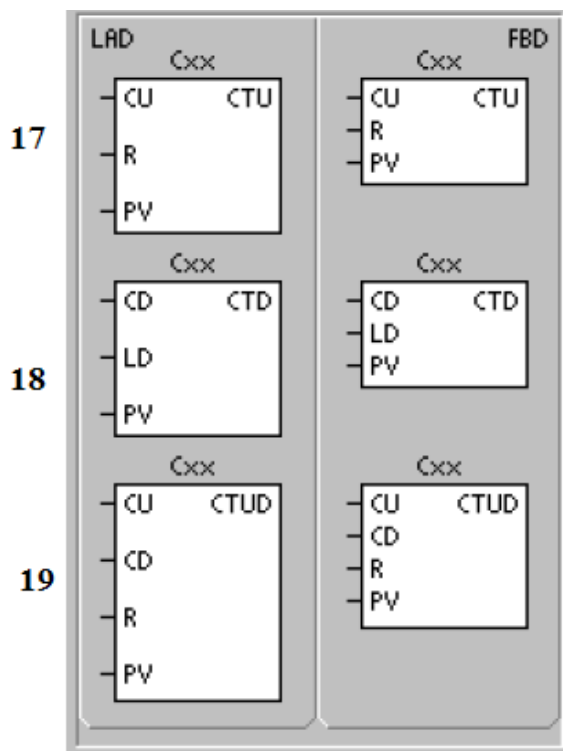


Рисунок Б.5

Команда прямого рахунку (17) (CTU) збільшує поточне значення лічильника з появою наростаючого фронту на вході (CU). Коли поточне значення Cxx більше або дорівнює попередньо встановленому значенню PV, біт лічильника Cxx устанавлюється. Лічильник скидається, коли включається вхід скидання (R), або коли виконується команда скидання. Лічильник припиняє рахунок при досягненні максимального значення (32767).

Команда зворотного рахунку (18) (CTD) зменшує поточне значення лічильника з появою наростаючого фронту на вході (CD). Коли поточне значення Cxx дорівнює нулю, біт лічильника Cxx включається. Лічильник скидає свій біт Cxx і завантажує поточне значення попередньо встановленим значенням PV, коли включається вхід завантаження LD. Лічильник зупиняється, коли він досягає нуля, і біт лічильника Cxx включається.

Реверсивний лічильник (19) виконує рахунок у прямому напрямку при наростаючому фронті на вході прямого рахунку (CU). При наростаючому фронті на вході зворотного рахунку (CD) лічильник виконує рахунок у зворотному напрямку. Якщо поточне значення (Cxx) \geq попередньо встановленому значенню (PV), то включається біт лічильника (Cxx). Лічильник скидається, якщо активізується вхід скидання.

Тому що для кожного лічильника є тільки одне поточне значення, не призначайте один і той же номер більш ніж одному лічильнику (прямі, реверсивні й зворотні лічильники з тим самим номером звертаються до того ж самого поточного значення).

Якщо ви скидаєте лічильник за допомогою команди “Скидання”, то біт лічильника скидається, а поточне значення встановлюється в нуль. Номер лічильника використовується для звертання як до поточного значення, так і до біта лічильника.

Швидкісні лічильники

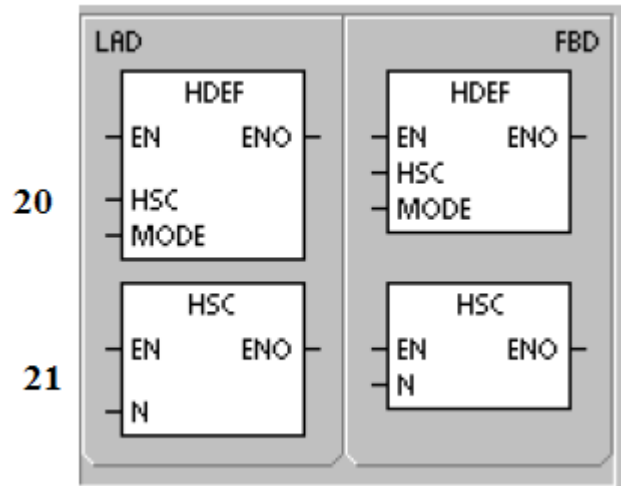


Рисунок Б.6

Команда визначення режиму роботи швидкісного лічильника (20) (HDEF) устанавлює режим роботи для певного швидкісного лічильника (HSCx). Вибором режиму визначаються датчик тактових імпульсів, напрямом і функції запуску й скидання швидкісного лічильника. Для кожного швидкісного лічильника використається одна команда визначення режиму роботи.

Команда активізації швидкісного лічильника (21) (HSC) конфігурує й управляє режимом роботи швидкісного лічильника через сигнальні стани бітів спеціальної пам'яті HSC. Параметр N визначає номер швидкісного лічильника. Швидкісні лічильники можуть бути сконфігуровані на дванадцять різних режимів роботи. Кожний лічильник має спеціалізовані входи, які підтримують такі функції, як датчик тактових імпульсів, керування напрямком, скидання й запуск. Для двофазних лічильників обидва датчики тактових імпульсів можуть працювати зі своєю максимальною швидкістю.

Швидкісні лічильники ведуть лік подій, що відбуваються з високою швидкістю, які не можуть управлятися при швидкостях сканування S7-200. Максимальна частота рахунку швидкісного лічильника залежить від типу вашого CPU S7-200.

5 Команда виводу імпульсів

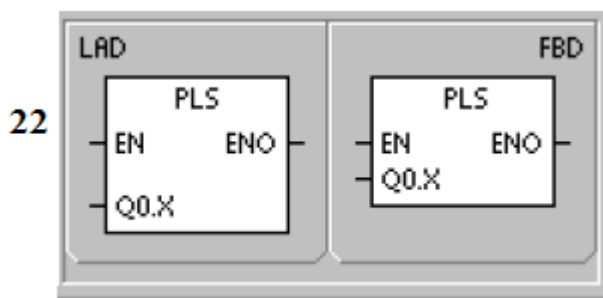


Рисунок Б.7

Команда виводу імпульсів (22) (PLS) використовується для керування функціями "Вивід послідовності імпульсів" (PTO) і "Широтно-імпульсна модуляція" (PWM), наявними в розпорядженні швидкісних виходів (Q0.0 і Q0.1). Удосконалений "Майстер керування позиціонуванням" створює команди, настроєні на вашу програму, що спрощує програмування ваших завдань і дозволяє використати переваги додаткових функцій CPU S7-200.

PTO надає в розпорядження вихідний сигнал в виді прямокутних імпульсів (з відносною тривалістю імпульсів 50%), період проходження яких і кількість визначається користувачем.

PWM надає в розпорядження вихідний імпульсний сигнал з постійним періодом проходження й змінною відносною тривалістю імпульсів, причому період проходження імпульсів і їхню ширину визначає користувач.

S7-200 має два генератори PTO/PWM, які формують або швидку послідовність імпульсів, або широтно-модульований імпульсний сигнал. Один генератор поставлений у відповідність цифровому виходу Q0.0, інший генератор -цифровому виходу Q0.1. Певна адреса спеціальної пам'яті (SM) зберігає для кожного генератора наступні дані: керуючий байт (8-бітове значення), кількість імпульсів (32-бітове значення без знака), а також період проходження імпульсів і значення ширини імпульсу (16-бітове значення без знака). Генератори PTO і PWM і регістр образу процесу спільно використовують Q0.0 і Q0.1. Коли функція PTO або PWM активна на Q0.0 або Q0.1, то виходом управляє генератор PTO або PWM, а нормальне використання виходу заблоковане. На форму вихідного сигналу не впливає ні стан регістра образу процесу, ні примусове присвоєння значень виходам, ні виконання команд безпосереднього виводу. Коли генератор PTO/PWM не активний, керування виходом вертається регістру образу процесу. Регістр образу процесу визначає початковий і кінцевий стан імпульсного вихідного сигналу, викликаючи його початок і завершення на високому або низькому рівні.

6 Таймерні команди

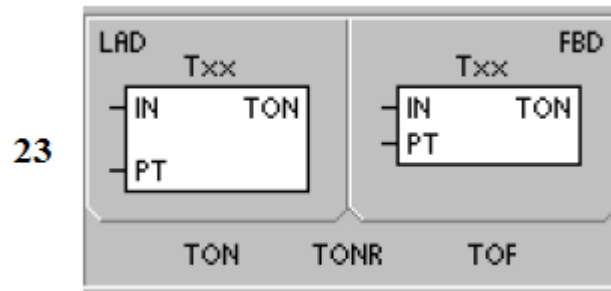


Рисунок Б.8

Команди "Таймер із затримкою включення" (23) (TON) і "Таймер із затримкою включення із запам'ятовуванням" (TONR) відраховують час, коли включений дозвіл на вході. Номер таймера (Тхх) визначає його розв'язну здатність, і ця розв'язна здатність тепер відображається в блоці команди.

Таймер із затримкою вимикання (TOF) використовується для затримки вимикання виходу на фіксований інтервал часу після вимикання входу. Номер таймера (Тхх) визначає його розв'язну здатність, і ця розв'язна здатність тепер відображається в блоці команди.

- Таймер із затримкою включення TON може використатися для відліку окремого інтервалу.
- Таймер із затримкою включення із запам'ятовуванням TONR може використатися для нагромадження декількох відлічених інтервалів часу.
- Таймер із затримкою вимикання TOF може використатися для збільшення інтервалу часу після вимикання (або збоя) наприклад, для охолодження двигуна після його відключення. Команди TON і TONR відраховують час, коли включений вхід дозволу. Коли поточне значення стає більше або дорівнює попередньо встановленому значенню часу, біт таймера устанавлюється.
- Поточне значення таймера TON скидається, коли вимикається вхід дозволу, тоді як поточне значення таймера TONR зберігається, коли цей вхід вимикається.
- Ви можете використати таймер TONR для накопичення часу, коли цей вхід включається й вимикається. Для стирання поточного значення TONR використовується команда "Скидання" (R).
- Таймери TON і TONR продовжують рахунок після досягнення попередньо встановленого значення, вони зупиняють рахунок при досягненні максимального значення, яке дорівнює 32767.

Команда TOF використовується для затримки вимикання виходу на фіксований інтервал часу після вимикання входу. Коли включається вхід

дозволу, негайно включається біт таймера, а поточне значення встановлюється в 0. Коли вхід вимикається, таймер веде відлік часу, поки час не досягне попередньо встановленого значення.

7 Арифметичні операції

7.1 Операції додавання, вирахування, множення й ділення

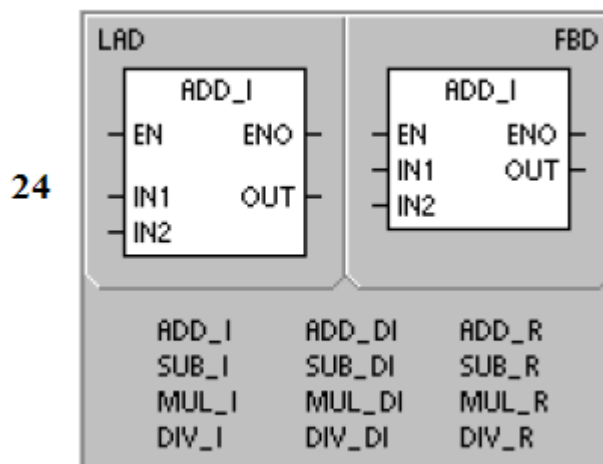


Рисунок Б.9

Додавання

$IN1 + IN2 = OUT$
 $IN1 + OUT = OUT$

Вирахування

$IN1 - IN2 = OUT$ *LAD i FBD*
 $OUT - IN1 = OUT$ *STL*

Команди додавання цілих чисел (+I) і вирахування цілих чисел (-I) складають або віднімають два 16-бітових цілих числа й дають 16-бітовий результат. Команди додавання подвійних цілих чисел (+D) і вирахування подвійних цілих чисел (-D) складають або віднімають два 32-бітових цілих числа й дають 32-бітовий результат. Команди додавання дійсних чисел (+R) і вирахування дійсних чисел (-R) складають або віднімають два 32-бітових дійсних числа й дають результат у вигляді 32-бітового дійсного числа.

Множення

$IN1 * IN2 = OUT$
 $IN1 * OUT = OUT$

Ділення

$IN1 / IN2 = OUT$ *LAD i FBD*
 $OUT / IN1 = OUT$ *STL*

Команди множення цілих чисел (*I) і ділення цілих чисел (/I) перемножують або ділять два 16-бітових цілих числа й дають 16-бітовий результат (для ділення остача не зберігається.) Команди множення подвійних цілих чисел (*D) і ділення подвійних цілих чисел (/D) перемножують або ділять два 32-бітових цілих числа й дають 32-бітовий результат (для ділення остача не зберігається.) Команди множення дійсних

чисел (*R) і ділення дійсних чисел (/R) перемножують або ділять два 32-бітові дійсні числа й дають у результаті 32-бітове дійсне число.

Біти спеціальної пам'яті й ENO

SM1.1 указує на помилки переповнення й неприпустимі значення. Якщо біт SM1.1 установлений, то стан SM1.0 і SM1.2 не має значення, і первісні вхідні операнди не міняються. Якщо SM1.1 і SM1.3 не встановлені, то арифметична операція завершилася із припустимим результатом, і SM1.0 і SM1.2 містять припустимий статус. Якщо в час операції ділення встановлюється SM1.3, то всі інші біти стану арифметичної операції залишаються незмінними.

7.2 Множення цілих чисел з поданням результату у вигляді подвійного цілого числа й розподіл цілих чисел із залишком

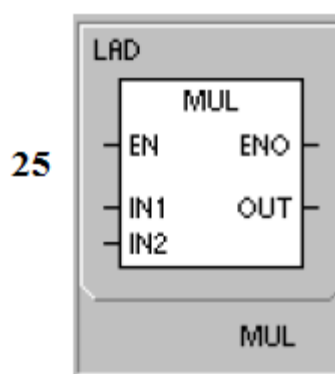


Рисунок Б.10

7.2.1 Множення цілих чисел з поданням результату у вигляді подвійного цілого числа

$IN1 * IN2 = OUT$ LAD і FBD

$IN1 * OUT = OUT$ STL

Команда множення цілих чисел з поданням результату у вигляді подвійного цілого числа (MUL) перемножує два 16-бітових цілих числа й дає 32-бітовий добуток. У команді множення STL молодше слово (16 бітів) 32-бітового виходу OUT використовується як один зі співмножників.

7.2.2 Розподіл цілих чисел із залишком

$IN1 / IN2 = OUT$ LAD і FBD

$OUT / IN1 = OUT$ STL

Команда розподілу цілих чисел із залишком (DIV) ділить два 16-бітових цілих числа й дає 32-бітовий результат, що складається з 16-бітового залишку (старше слово) і 16-бітової частки (молодше слово). У команді

розподілу STL молодше слово (16 бітів) 32-бітового виходу OUT використовується як ділене.

Біти спеціальної пам'яті й ENO

Для обох команд, біти спеціальної пам'яті (SM) указують на помилки й неприпустимі значення. Якщо під час операції розподілу встановлюється SM1.3 (розподіл на нуль), то всі інші біти стану арифметичної операції залишаються незмінними. У протилежному випадку всі підтримувані біти стану арифметичної операції містять після завершення арифметичної операції припустимий статус.

7.3 Числові функції синус, косинус і тангенс

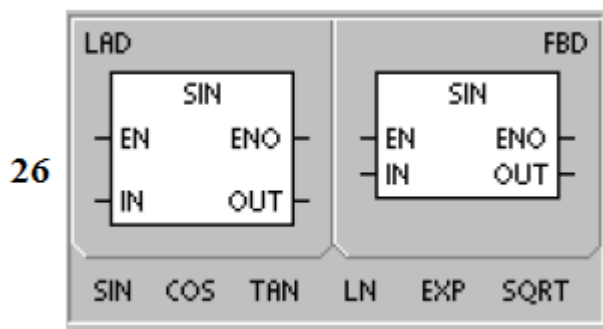


Рисунок Б.11

Команди обчислення синуса (SIN), косинуса (COS) і тангенса (TAN) обчислюють тригонометричну функцію кутової величини IN і поміщають результат в OUT. Вхідний кут задається в радіанах.

$$\text{SIN (IN) = OUT} \quad \text{COS (IN) = OUT} \quad \text{TAN (IN) = OUT}$$

Для перетворення кута із градусів у радіани:

Використайте MUL_R (*R) для множення кута в градусах на 1.745329E-2 (приблизно на $\pi/180$).

Команда обчислення натурального логарифма знаходить натуральний логарифм числа в IN і поміщає результат в OUT. Команда обчислення експоненти виконує експонентну операцію з піднесення e в ступінь, заданий значенням в IN, і поміщає результат в OUT.

$$\text{LN (IN) = OUT} \quad \text{EXP (IN) = OUT}$$

Для одержання десяткового логарифма з натурального: розділіть натуральний логарифм на 2,302585 (приблизно натуральний логарифм 10).
Для піднесення будь-якого дійсного числа в ступінь, заданий іншим дійсним числом, включаючи дроби.

Об'єднаєте команду обчислення експоненти з командою обчислення натурального логарифма. Наприклад, для піднесення X у ступінь Y уведіть наступну команду: EXP (Y * LN (X)).

Команда обчислення квадратного кореня (SQRT) витягає квадратний корінь із дійсного числа (IN) і дає результат у вигляді дійсного числа OUT.

$$\text{SQRT (IN)} = \text{OUT}$$

Для одержання інших коренів:

$$5 \text{ у кубі} = 5^3 = \text{EXP}(3 * \text{LN}(5)) = 125$$

$$\text{Кубічний корінь із } 125 = 125^{(1/3)} = \text{EXP}((1/3) * \text{LN}(125)) = 5$$

$$\text{Квадратний корінь із } 5 \text{ у кубі} = 5^{(3/2)} = \text{EXP}(3/2 * \text{LN}(5)) = 11.18034$$

7.4 Команди інкрементування й декрементування

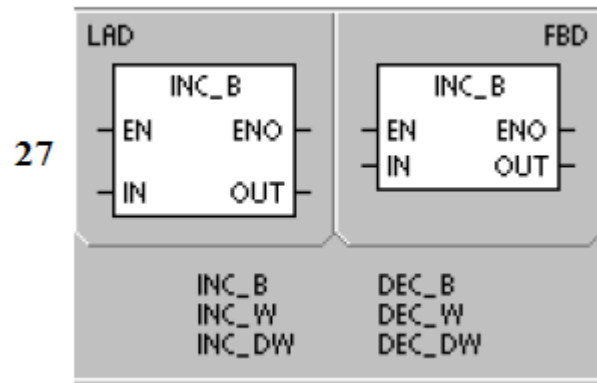


Рисунок Б.12

Збільшення на 1

$$\text{IN} + 1 = \text{OUT} \quad \text{LAD i FBD}$$

$$\text{OUT} + 1 = \text{OUT} \quad \text{STL}$$

Зменшення на 1

$$\text{IN} - 1 = \text{OUT} \quad \text{LAD i FBD}$$

$$\text{OUT} - 1 = \text{OUT} \quad \text{STL}$$

Команди інкрементування й декрементування додають 1 до входу IN або віднімають із нього 1 і поміщають результат у змінну OUT.

Операції збільшення байта на 1 (INCB) і зменшення байта на 1 (DECB) є беззнаковими.

Операції збільшення слова на 1 (INCW) і зменшення слова на 1 (DECW) ураховують знак.

Операції збільшення подвійного слова на 1 (INCD), збільшення й зменшення подвійного слова на 1 (DECD) ураховують знак.

Навчальне видання

Методичні вказівки
до самостійної роботи
з дисципліни “Мікропроцесорні системи керування”
(для спеціальності 7.092203 “Електромеханічні системи автоматизації та
електропривід”)

НАЛИВАЙКО Олександр Михайлович

Редактор І.І.Дьякова

Комп’ютерна верстка О.П.Ордіна

133/2008. Підп. до друку

Формат 60x84/16.

Папір офсетний. Ум. друк. арк.

Обл.-вид. арк.

Тираж прим. Зам. №

Видавець і виготівник

«Донбаська державна машинобудівна академія»

84313, м. Краматорськ, вул. Шкадінова, 72

Свідоцтво про внесення суб’єкта видавничої справи до державного реєстру
серія ДК № 1633 від 24.12.2003