

Міністерство освіти і науки України

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

Кафедра комп'ютерно-інтегрованих технологічних процесів і виробництв

**Методичні вказівки
до виконання лабораторної роботи
“Програмування ПЛК.
Вивчення мови релейних діаграм (*LD*)”
з дисципліни
“Мікропроцесорні та програмні засоби автоматизації”**

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

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

Одеса – 2016

Рецензент – Стопакевич О.А.

Тігарєв А.М. Методичні вказівки до виконання лабораторної роботи “Програмування ПЛК. Вивчення мови релейних діаграм (*LD*)” / Тігарєв А.М. – Одеса: ОНАЗ ім. О.С. Попова, 2016. – с.

Розглянуто реалізацію вузлів автоматики за допомогою релейно-контактних схем. Наведено приклади застосування типових релейно-контактних схем для програмування на мові *LD* за стандартом МЭК 61131.

УХВАЛЕНО

на засіданні кафедри
КІТ П і В і рекомендовано
до друку.
Протокол № 6 від 22.02.2016 р.

ЗАТВЕРДЖЕНО

методичною радою
академії зв'язку.
Протокол № 6 від 23.02.2016 р.

ЗМІСТ

Лабораторна робота № 2. ПРОГРАМУВАННЯ ПЛК. ВИВЧЕННЯ МОВИ РЕЛЕЙНИХ ДІАГРАМ (<i>LD</i>).....	4
1 Мета роботи	4
2 Основні положення	4
2.1 Застосування типових релейно-контактних схем	4
2.2 Мова <i>LD</i>	8
3 Контрольні питання	9
4 Домашнє завдання	10
5 Лабораторне завдання	10
6 Зміст протоколу	11
7 Перелік рекомендованої літератури	11
 Лабораторная работа № 2. ПРОГРАММИРОВАНИЕ ПЛК. ИЗУЧЕНИЕ ЯЗЫКА РЕЛЕЙНЫХ ДИАГРАММ (<i>LD</i>)	12
1 Цель работы	12
2 Основные положения	12
2.1 Применение типовых релейно-контактных схем	12
2.2 Язык <i>LD</i>	16
3 Контрольные вопросы	17
4 Домашнее задание	18
5 Лабораторное задание	18
6 Содержание протокола	18
7 Перечень рекомендованной литературы	20

ПРОГРАМУВАННЯ ПЛК. ВИВЧЕННЯ МОВИ РЕЛЕЙНИХ ДІАГРАМ (*LD*)

1 Мета роботи

Метою роботи є ознайомлення з принципами побудови принципових електричних схем автоматичного керування та сигналізації, а також набуття навичок розробки програм для програмованих логічних контролерів мовою *LD* за стандартом МЭК 61131.

2 Основні положення

2.1 Застосування типових релейно-контактних схем

Будь-який технологічний процес повинен виконуватися відповідно до певних правил, послідовностей операцій, тобто відповідно до його "Алгоритму функціонування" й "Алгоритму управління".

Складні процеси можуть бути поділені на елементарні операції, взаємодія яких здійснюється за елементарною логікою.

При автоматизації технологічних процесів виникає необхідність у керуванні процесами без безпосереднього втручання людини, в цьому випадку спеціальні технічні засоби автоматизації повинні забезпечити ведення процесу відповідно до його алгоритму керування. Схема, що пояснює, яким чином взаємодіють технічні засоби автоматичного керування, називається структурною схемою керування.

Найбільш широке застосування при автоматизації процесів знаходять технічні засоби, що використовують електричну енергію. У цьому випадку схема керування буде називатися принциповою електричною схемою регулювання або керування, сигналізації тощо.

Для виконання елементарної логіки широко використовуються релейно-контактні пристрої (електромагнітні реле, кнопки, перемикачі, пускачі, крокові пускачі тощо).

До основних елементарних логічних операцій відносяться:

- а) операції "АБО";
- б) операції "І";
- в) операції "НІ";
- г) операції "Пам'ять" тощо.

Складні схеми керування складаються з набору типових схем, що реалізують елементарні логічні операції:

1. Схема, що реалізує операцію "АБО" (рис. 2.1, а). У цій схемі проміжне електромагнітне реле К1 включається при замиканні кожної із кнопок (або *SB1*,

або $SB2$, або $SB3$) і замикає нормально розімкнутий контакт $K1$ реле, який включає сигнальну лампу $HL1$.

Замість кнопок можуть використовуватися технологічні контактні датчики, кінцеві вимикачі, реле тощо.

Логічна операція "АБО" може бути записана в алгебраїчній формі у вигляді:

$$K1 = SB1 + SB2 + SB3 = SB1 \vee SB2 \vee SB3,$$

де $K1$ – обмотка реле; $SB1, SB2, SB3$ – кнопки, $HL1$ – сигнальна лампа, L – шина живлення, N – шина нуля.

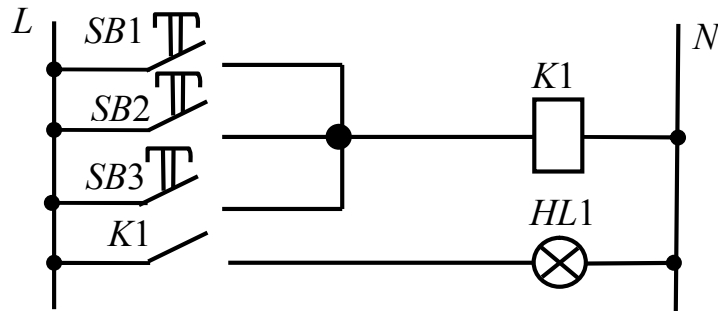


Рисунок 2.1, а

2. Схема, що реалізує операцію "І" (рис. 2.1, б). У цій схемі реле $K1$ включається тільки при замиканні всіх кнопок $SB1, SB2, SB3$.

Логічна операція "І" може бути записана в алгебраїчній формі у вигляді

$$K1 = SB1 * SB2 * SB3 = SB1 \wedge SB2 \wedge SB3.$$

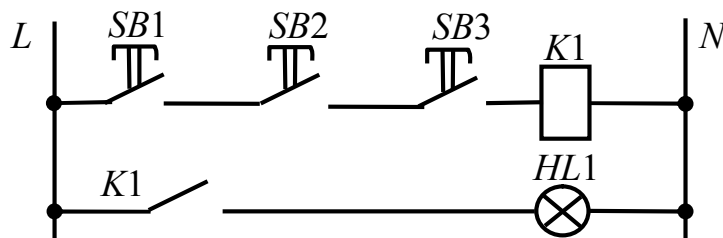


Рисунок 2.1, б

3. Схема, що реалізує операцію "НІ" ("заперечення", "інверсія") (рис. 2.1, в).

У цій схемі при замиканні кнопки подається живлення на обмотку реле $K1$, яке включається, та своїм розімкнутим контактом $K1$ включає лампу $HL1$ і, навпаки, при розімкненні $SB1$, лампа $HL1$ виключається.

Операція "НІ" алгебраїчно записується у вигляді:

$$HL1 = \overline{K1}$$

(риска над символом – знак заперечення, інверсії).

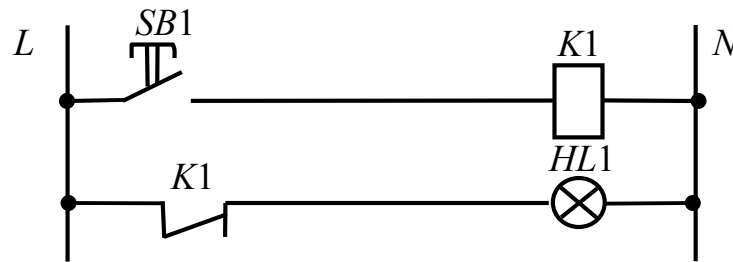


Рисунок 2.1, в

4. Схема, що реалізує операцію "затримка" (рис. 2.1, г). У цій схемі при замиканні кнопки $SB1$ подається живлення на обмотку реле часу $KT1$, однак лампа $HL1$ включається тільки після замикання контактів $KT1$ з витримкою τ до замикання.

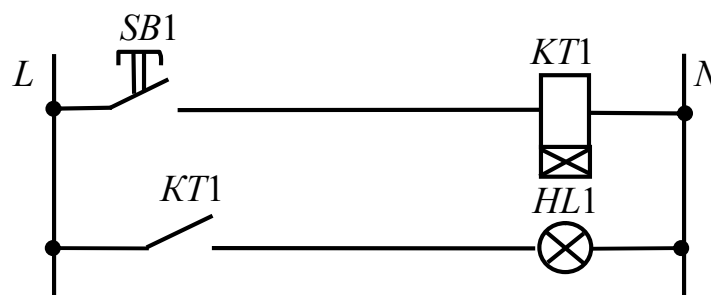


Рисунок 2.1, г

Операція, яка реалізована схемою, записується у вигляді:

$$HL1 = SB1(t - \tau) = KT1(\tau),$$

де t – момент замикання кнопки $SB1$; τ - час затримки реле часу $KT1$.

5. Операція "Пам'ять" (рис. 2.1, д).

Схема використовується за необхідності перетворення командного імпульсу малої діяльності в довгостроково діючий вплив. При короткочасному замиканні кнопки $SB1$ реле $K1$ включається. При цьому замикається його контакт $K1$, що шунтує кнопку $SB1$, тому при розмиканні $SB1$ реле $K1$ залишається включеним необмежено довго (поки є живлення на шинах схеми).

Дана операція записується у вигляді:

$$K1 = (SB1 + K1),$$

$$HL1 = K1.$$

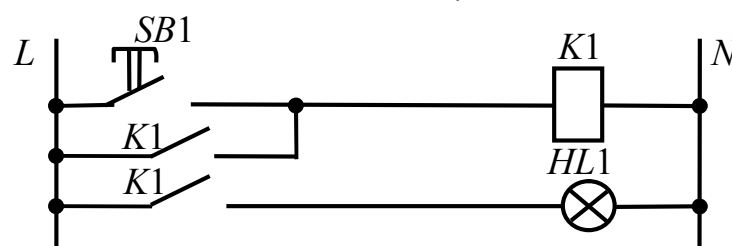


Рисунок 2.1, д

Розглянемо декілька практичних типових схем сигналізації і керування:

1. Типова схема керування електромагнітним виконавчим механізмом за допомогою реле $K2$ (рис. 2.2, а).

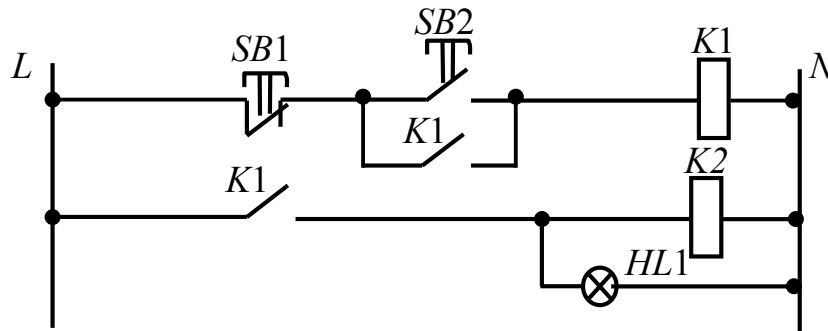


Рисунок 2.2, а

При впливі на кнопку $SB2$ реле $K1$ стає на "самопідхоплення" (реалізація логіки "Пам'ять"). При цьому подається живлення на обмотку електромагніта K , який включає виконавчий механізм і сигнальну лампу $HL1$. Відключення ланцюга виконується кнопкою $SB1$.

2. Типова схема керування електродвигуновим виконавчим механізмом – реверсивним двофазним двигуном змінного струму (рис. 2.2, б).

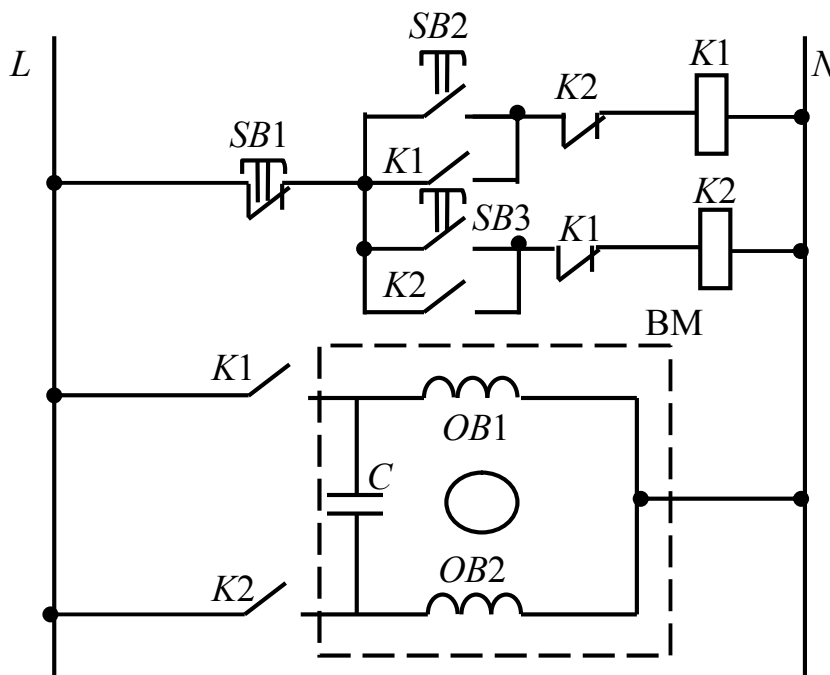


Рисунок 2.2, б

При впливі на кнопку $SB2$ включається реле $K1$, що своїм замикаючим контактом включає обмотку $OB1$ (це забезпечує обертання двигуна, наприклад, за годинниковою стрілкою) виконавчого механізму BM .

Після натиснення на кнопку $SB3$ забезпечується підключення обмотки $OB2$ (це забезпечує обертання двигуна, навпаки, проти годинникової стрілки) виконавчого механізму BM .

Замкнуті контакти $K2$ і $K1$ в вихідному стані виключають можливість одночасного включення реле $K1$ і $K2$ при одночасному впливі на обидві пускові кнопки.

3. Схема автоматичної передпускової сигналізації (рис. 2.2, в).

При подачі оператором пускового імпульсу на включення технологічного маршруту, останній повинен включатися автоматично тільки після дії попереджувального звукового сигналу протягом заданого часу, який оповіщає персонал про включення технологічного обладнання.

Після натискання на кнопку $SB1$ включається реле часу $KT1$ і реле-повторювач $K1$. За допомогою замикаючого контакту $K1$ ланцюг стає на самопідхоплення (реалізується логіка "Пам'ять").

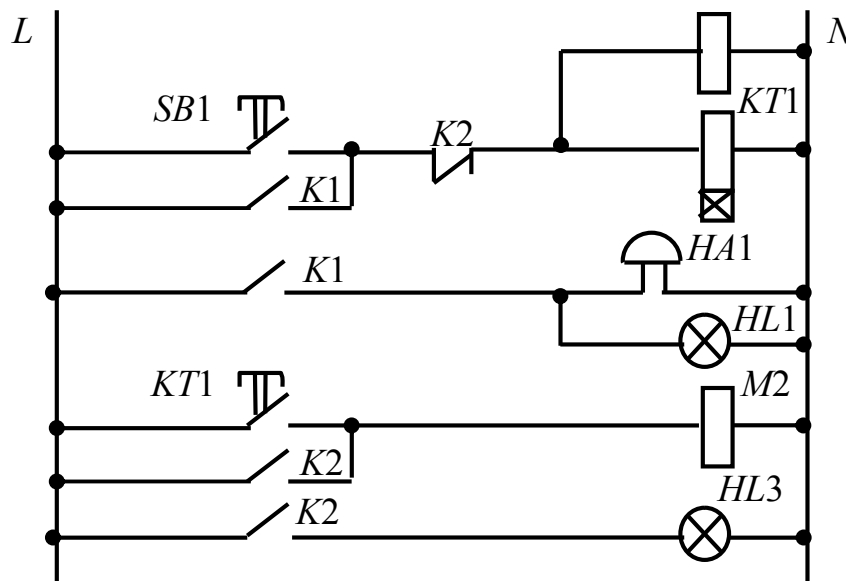


Рисунок 2.2, в

Після закінчення заданого часу, який обумовлений налаштуванням реле часу $KT1$, замикається контакт реле $KT1$, з витримкою на замикання, і включається проміжне реле $K2$, останнє забезпечує:

- у ланцюзі $KT1$ розмикає ланцюг за допомогою свого розмикаючого контакту;
- у ланцюзі $K2$ – реалізує логіку "Пам'ять";
- у ланцюзі $HL2$ (ланцюг $HL2$ імітує ланцюг керування першої черговості включення машини поточно-транспортної системи) – забезпечує імпульс на включення (на щиті оператора загориться світлове табло).

2.2 Мова LD

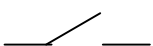
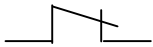
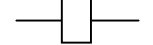
Мова релейних діаграм LD (*Ladder Diagram*) або релейно-контактних схем (РКС) – графічна мова, що реалізує структури електричних ланцюгів. РКС – це американський винахід. На початку 70-х рр. XX ст. релейні автомати складальних конвеєрів почали поступово витискуватися програмованими контролерами. Якийсь час ті й інші працювали одночасно й обслуговувалися тими самими людьми.

Так з'явилося завдання прозорого перенесення релейних схем у ПЛК. Різні варіанти програмної реалізації релейних схем створювалися практично всіма провідними виробниками ПЛК. Завдяки простоті подання РКС знайшов заслужену популярність, що і стало основною причиною включення його у стандарт МЕК.

Слова «релейна логіка» звучать сьогодні досить архаїчно, майже як «ламповий комп'ютер». Тим більше у зв'язку зі створенням численних швидкодіючих і надійних безконтактних (зокрема, оптоелектронних) реле і потужних перемикаючих приладів, таких як потужні польові транзистори, керовані тиристори і прилади *IGBT*. Але, незважаючи на це, релейна техніка усе ще дуже широко застосовується.

Графічно *LD*-діаграма показана у вигляді двох вертикальних шин живлення. Між ними розташовані ланцюги, утворені з'єднанням контактів. Навантаженням кожного ланцюга служить реле. Кожне реле має контакти, які можна використати в інших ланцюгах (табл. 2.1).

Таблиця 2.1 – Зіставлення позначень базових елементів *LD* і позначень ЄСКД

<i>LD</i>	ЄСКД	Позначення
		Нормально розімкнений контакт
		Нормально замкнений контакт
		Обмотка реле

Логічно послідовне (І), паралельне (АБО) з'єднання контактів та інверсія (НІ) утворюють базис Буля. У результаті *LD* ідеально підходить не тільки для побудови релейних автоматів, але й для програмної реалізації комбінаційних логічних схем. Завдяки можливості включення в *LD* функцій і функціональних блоків, виконаних іншими мовами, сфера застосування мови практично не обмежена.

3 Контрольні питання

3.1 Які основні елементарні логічні операції?

3.2 Які технічні засоби автоматизації можуть виконувати елементарні логічні операції?

3.3 Яку задачу реалізує "затримка"?

3.4 Яку задачу виконує типова схема передпускової сигналізації?

3.5 Як реалізується схема "Пам'ять"?

3.6 Яким пристроєм забезпечується необхідна тривалість дії звукового сигналу?

4 Домашнє завдання

4.1 Підготувати протокол згідно з розд. 6. Обґрунтувати вимоги й розробити систему пуску-зупинки ділянки технологічного процесу згідно з темою комплексного завдання. Виконати її текстовий опис, розробити блок-схему алгоритму. На підставі цього скласти його програмну реалізацію мовою *LD*. Програму забезпечити необхідними коментарями.

4.2 Програма виконується по закінченню лабораторного завдання.

5 Лабораторне завдання

5.1 Запустити середовище розробки *Codesys*. Якщо в організаторі об'єктів автоматично відкривається проект, потрібно закрити його за допомогою меню *File* → *Close*. Виконати побудову ("синтез") електричних принципових схем автоматичного керування й сигналізації та перевірку їх алгоритмів функціонування ("аналіз").

5.2 Порядок виконання роботи (використовувати для розробки програм опис редактора мови *LD* з файла *Codesys_V23_RU.pdf*).

Зібрати у робочій області *Codesys* схему, яка реалізує елементарну операцію "АБО", і перевірити її працездатність (рис. 2.1, а).

Те саме, операція "І" (рис. 2.1, б).

Те саме, операція "НІ" (рис. 2.1, в).

Те саме, операція "Пам'ять" (рис. 2.1, д).

Те саме, операція "затримка" (рис. 2.1, г) використовуючи таймер з бібліотеки *Standart.lib*.

Зібрати схему керування приводом через реле *K2* (рис. 2.2, а)

Зібрати схему попереджувальної сигналізації (рис. 2.2, б),

Зібрати схему керування виконавчим механізмом ВМ (рис. 2.2, в).

5.3 Включити режим симуляції за допомогою меню *Online* → *Simulation Mode*.

5.4 Під'єднати середовище до симуляції у ПЛК за допомогою меню *Online* → *Login* (Підключення). Зверніть увагу на те, що під час приєднання виконується компіляція проекту.

5.5 Запустити проект на виконання за допомогою меню *Online* → *run* (Старт).

5.6 Виконати імітацію включення відповідних кнопок, підтверджуючи це за допомогою меню *Online* (Підключення) опцією *Write value* (Записати значення).

5.7 Під час функціонування проекту дослідіть процес зміни змінних в усіх блоках.

5.8 Виконати контрольний приклад.

Приклад. Розробити програму керування відповідно алгоритму, показаному на рис. 5.1.

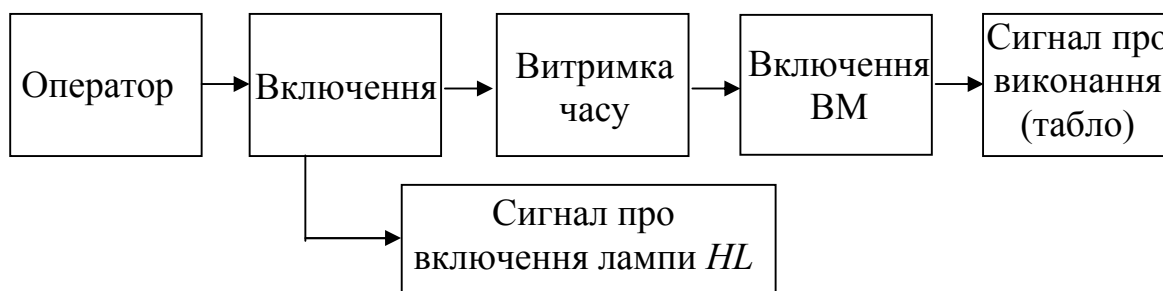


Рисунок 5.1 – Алгоритм включення механізмів оператором

Увага!

При запуску програмного проекту на тестування або налагодження необхідно виконати підключення бібліотек (*Standart.lib*, *UTIL.LIB* і ін.) [5]. Стандартна бібліотека *Standart.lib* містить строкові функції, перемикачі, детектори імпульсів, лічильники, таймери. Бібліотека *UTIL.LIB* містить регулятори, генератори сигналів, аналогові компаратори й ін. Для підключення бібліотек необхідно в меню *Window* (Вікно) ("*Window*"→"*Library Vanager*"→"*Additional library*") вибрати мишею опцію *Manager library* (Менеджер бібліотек) і далі подвійним клацанням правої кнопки у верхньому порожньому вікні з'являється список бібліотек, з яких вибираємо необхідну.

Зміст звіту

Обробка результатів дослідження. Всі схеми викреслити в протоколі.

Навести рівняння ланцюгів і назвати елементарні операції, виконувані елементами схеми.

6 Зміст протоколу

Протокол лабораторної роботи "Програмування ПЛК. Вивчення релейно-контактних схем (*LD*)" оформлюється в окремому зошиті для лабораторних робіт. Протокол має містити назву лабораторної роботи та її мету, відповіді на контрольні питання, хід виконання домашнього та лабораторного завдання, коментарі до програм, розроблених та налагоджених під час виконання лабораторної роботи, блок-схеми програм, висновки.

7 Перелік рекомендованої літератури

1. Петров И.В. «Программируемые контроллеры. Стандартные языки и приемы прикладного программирования». – М.: ООО «СОЛОН-Пресс», 2004.
2. Методичні посібники для лабораторних робіт даного курсу.
3. Конспект лекцій з курсу МП і ПЗА.
4. Веб-сторінка фірми *Smart Software Solutions GmbH* виробника середовища *CoDeSys* <http://www.3s-software.com/>
5. Веб-сторінка ПК "Пролог", підтримка середовища *CoDeSys* російською мовою <http://www.codesys.ru/>

ПРОГРАММИРОВАНИЕ ПЛК. ИЗУЧЕНИЕ ЯЗЫКА РЕЛЕЙНЫХ ДИАГРАММ (LD)

1 Цель работы

Целью работы является ознакомления с принципами построения принципиальных электрических схем автоматического управления и сигнализации, а также приобретение навыков разработки программ для программированных логических контролеров на языке *LD* по стандарту МЭК 61131.

2 Основные положения

2.1 Применение типовых релейно-контактных схем

Любой технологический процесс должен выполняться соответственно определенным правилам, последовательностям операций, т.е. соответственно его "Алгоритму функционирования" и "Алгоритму управления".

Сложные процессы могут быть поделены на элементарные операции, взаимодействие которых осуществляется по элементарной логике.

При автоматизации технологических процессов возникает необходимость в управлении процессами без непосредственного вмешательства человека, в этом случае специальные технические средства автоматизации должны обеспечить ведение процесса соответственно его алгоритму управления. Схема, которая объясняет, каким образом взаимодействуют технические средства автоматического управления, называется структурной схемой управления.

Наиболее широкое применение при автоматизации процессов получили технические средства, которые используют электрическую энергию. В этом случае схема управления будет называться принципиальной электрической схемой регулирования или управления, сигнализации и т.п.

Для выполнения элементарной логики широко используются релейно-контактные устройства (электромагнитные реле, кнопки, переключатели, пускатели, шаговые пускатели и т.п.).

К основным элементарным логическим операциям относятся:

- а) операции "ИЛИ";
- б) операции "И";
- в) операции "НЕ";
- г) операции "Память" и т.п.

Сложные схемы управления состоят из набора типичных схем, которые реализуют элементарные логические операции:

1. Схема, которая реализует операцию "ИЛИ" (рис. 2.1, а). В этой схеме промежуточное электромагнитное реле $K1$ включается при замыкании каждой из кнопок (или $SB1$, или $SB2$, или $SB3$) и замыкает нормально разомкнутый контакт $K1$ реле, который включает сигнальную лампу $HL1$.

Вместо кнопок могут использоваться технологические контактные датчики, конечные выключатели, реле и т.п.

Логическая операция "ИЛИ" может быть записана в алгебраической форме в виде:

$$K1 = SB1 + SB2 + SB3 = SB1 \vee SB2 \vee SB3,$$

где $K1$ – обмотка реле; $SB1$, $SB2$, $SB3$ – кнопки, $HL1$ – сигнальная лампа, L – шина питания, N – шина нуля.

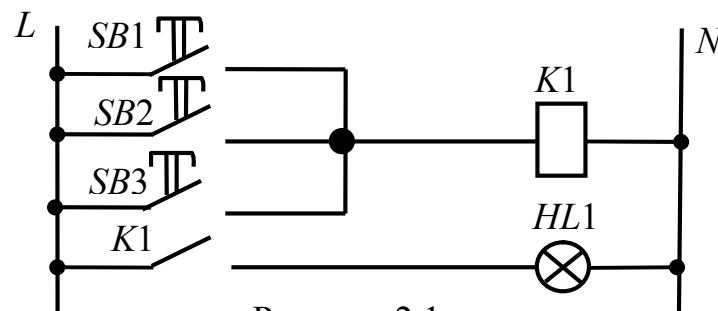


Рисунок 2.1, а

2. Схема, которая реализует операцию "И" (рис. 2.1, б). В этой схеме реле $K1$ включается только при замыкании всех кнопок $SB1$, $SB2$, $SB3$.

Логическая операция "И" может быть записана в алгебраической форме в виде

$$K1 = SB1 * SB2 * SB3 = SB1 \wedge SB2 \wedge SB3.$$

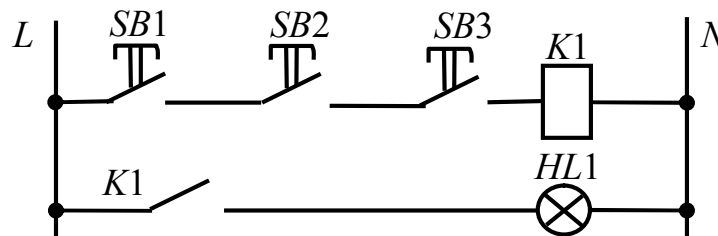


Рисунок 2.1, б

3. Схема, которая реализует операцию "НЕ" ("отрицание", "инверсия") (рис. 2.1, в).

В этой схеме при замыкании кнопки подается питание на обмотку реле $K1$, которое включается, и своим разомкнутым контактом $K1$ включает лампу $HL1$ и, наоборот, при размыкании $SB1$, лампа $HL1$ отключается.

Операция "НЕ" алгебраически записывается в виде:

$$HL1 = \overline{K1}$$

(черточка над символом – знак отрицания, инверсии).

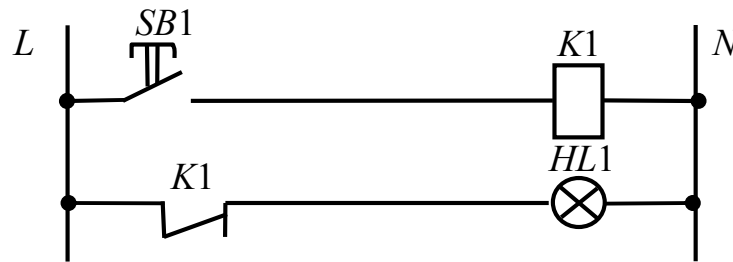


Рисунок 2.1, в

4. Схема, которая реализует операцию "задержка" (рис. 2.1, г). В этой схеме при замыкании кнопки $SB1$ подается питание на обмотку реле времени $KT1$, однако лампа $HL1$ включается только после замыкания контактов $KT1$ с выдержкой τ к замыканию.

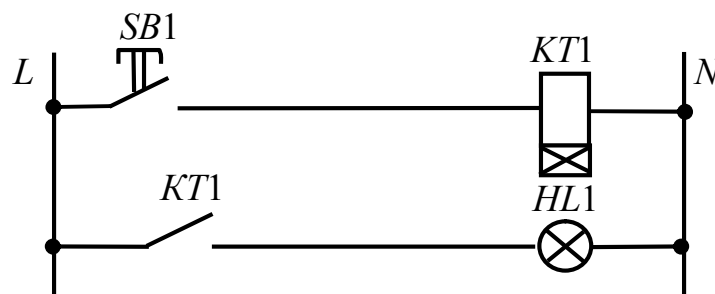


Рисунок 2.1, г

Операция, которая реализована схемой, записывается в виде:

$$HL1 = SB1 (t - \tau) = KT1(\tau)$$

где t – момент замыкания кнопки $SB1$; τ - время задержки реле времени $KT1$.

5. Операция "Память" (рис. 2.1, д).

Схема используется при необходимости преобразования командного импульса малой длительности в длительно действующее воздействие. При кратковременном замыкании кнопки $SB1$ реле $K1$ включается. При этом замыкается его контакт $K1$, что шунтирует кнопку $SB1$, поэтому при размыкании $SB1$ реле $K1$ остается включенным неограниченно долго (пока есть питание на шинах схемы).

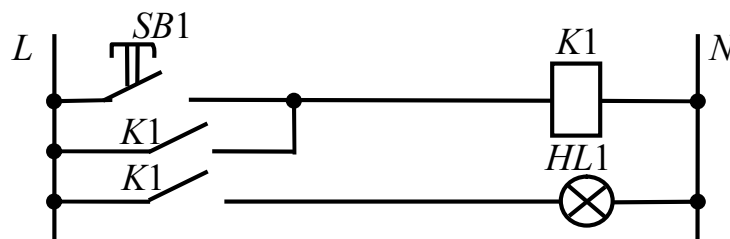


Рисунок 2.1, д

Данная операция записывается в виде:

$$K1 = (SB1 + K1),$$

$$HL1 = K1.$$

Рассмотрим несколько типичных практических схем сигнализации и управления:

1. Типичная схема управления электромагнитным исполнительным механизмом с помощью реле $K2$ (рис. 2.2, а).

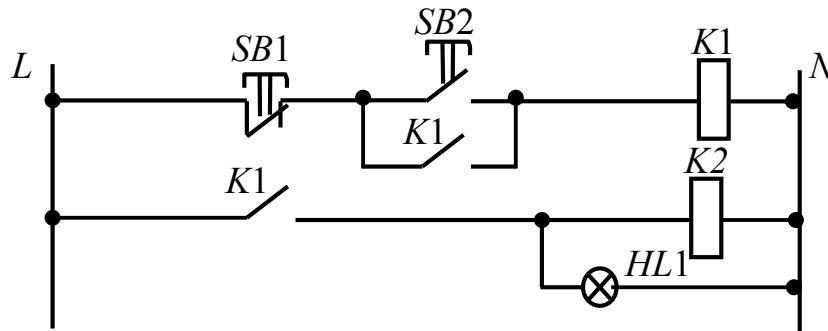


Рисунок 2.2, а

При воздействии на кнопку $SB2$ реле $K1$ становится на "самоподхват" (реализация логики "Память"). При этом подается питание на обмотку электромагнита $K2$, включающего исполнительный механизм $K2$ и сигнальную лампу $HL1$. Отключение цепи выполняется кнопкой $SB1$.

2. Типичная схема управления электродвигательным исполнительным механизмом – реверсивным двухфазным двигателем переменного тока (рис. 2.2, б).

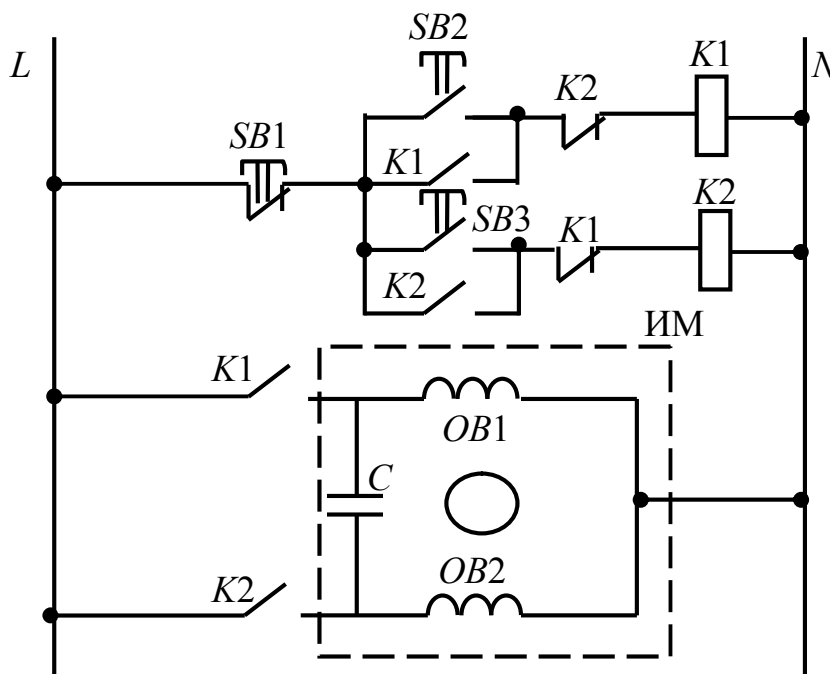


Рисунок 2.2, б

При воздействии на кнопку $SB2$ включается реле $K1$, которое своим замыкающим контактом включает обмотку $OB1$ (это обеспечивает вращение двигателя, например, по часовой стрелке) исполнительного механизма ИМ.

После нажатия на кнопку *SB3* обеспечивается подключение обмотки *OB2* (это обеспечивает вращение двигателя, наоборот, против часовой стрелки) исполнительного механизма ИМ.

Замкнутые контакты *K2* и *K1* в исходном состоянии исключают возможность одновременного включения реле *K1* и *K2* при одновременном влиянии на обе пусковые кнопки.

3. Схема автоматической предпусковой сигнализации (рис. 2.2, в).

При подаче оператором пускового импульса на включение технологического маршрута, последний должен включаться автоматически только после действия предупредительного звукового сигнала на протяжении заданного времени, которое оповещает персонал о включении технологического оборудования.

После нажатия на кнопку *SB1* включается реле времени *KT1* и реле-повторитель *K1*. С помощью замыкающего контакта *K1* цепь становится на самоподхват (реализуется логика "Память").

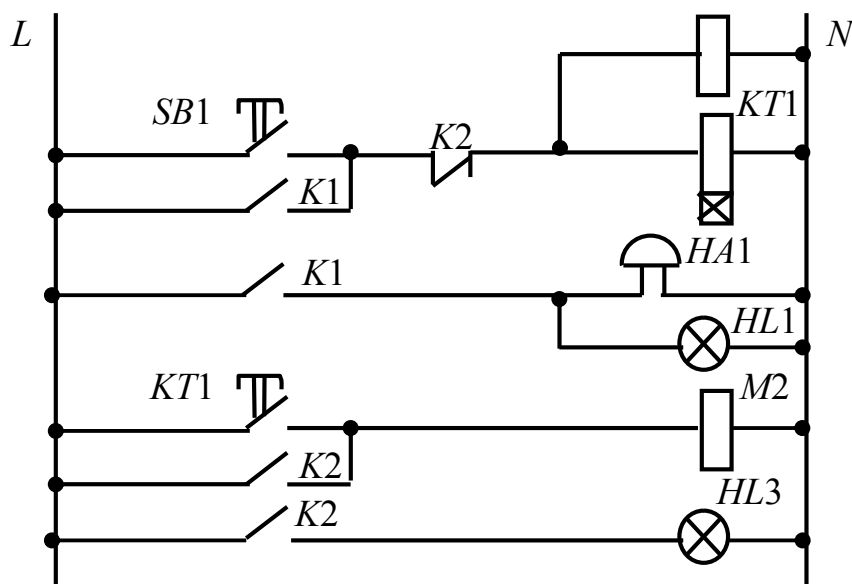


Рисунок 2.2, в

После окончания заданного времени, определяемого настройкой реле времени *KT1*, замыкается контакт реле *KT1*, с выдержкой на замыкание, и включается промежуточное реле *K2*, последнее обеспечивает:

- а) в цепи *KT1* размыкает цепь с помощью своего размыкающего контакта;
- б) в цепи *K2* – реализует логику "Память";
- в) в цепи *HL2* (цепь *HL2* имитирует цепь управления первой очереди включения машины поточно-транспортной системы) – обеспечивает импульс на включение (на щите оператора загорится световое табло).

2.2 Язык LD

Язык релейных диаграмм *LD* (*Ladder Diagram*) или релейно-контактных схем (РКС) – графический язык, который реализует структуры электрических цепей. РКС – это американское изобретение. В начале 70-х гг. XX в. релейные

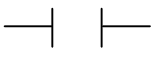
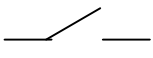
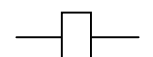
автоматы сборочных конвейеров начали постепенно вытесняться программируемыми контролерами. Некоторое время те и другие работали одновременно и обслуживались теми же самыми людьми.

Так появилась задача прозрачного переноса релейных схем в ПЛК. Разные варианты программной реализации релейных схем создавались практически всеми ведущими производителями ПЛК. Благодаря простоте представления РКС нашел заслуженную популярность, которая и стало основной причиной включения его в стандарт МЕК.

Слова «релейная логика» звучат сегодня довольно архаично, почти как «ламповый компьютер». Тем более в связи с созданием многочисленных быстродействующих и надежных бесконтактных (в частности, оптоэлектронных) реле и мощных переключающих приборов, таких как мощные полевые транзисторы, управляемые тиристоры и приборы *IGBT*. Но, несмотря на это, релейная техника все еще очень широко применяется.

Графически *LD*-диаграмма представлена в виде двух вертикальных шин питания. Между ними расположенные цепи, образованные соединением контактов. Нагрузкам каждой цепи служит реле. Каждое реле имеет контакты, которые можно использовать в других цепях (табл. 2.1)

Таблица 2.1 – Сопоставление обозначений базовых элементов *LD* и обозначений ЕСКД

<i>LD</i>	ЕСКД	Обозначение
		Нормально разомкнутый контакт
		Нормально замкнутый контакт
		Обмотка реле

Логически последовательное (И), параллельное (ИЛИ) соединение контактов и инверсия (НЕ) образуют базис Буля. В результате *LD* идеально подходит не только для построения релейных автоматов, но и для программной реализации комбинационных логических схем. Благодаря возможности включения в *LD* функций и функциональных блоков, выполненных на других языках, сфера применения языка практически не ограничена.

3 Контрольные вопросы

3.1 Какие основные элементарные логические операции?

3.2 Какие технические средства автоматизации могут выполнять элементарные логические операции?

3.3 Какую задачу реализует "задержка"?

3.4 Какую задачу выполняет типичная схема предпусковой сигнализации?

3.5 Как реализуется схема "Память"?

3.6 Каким устройством обеспечивается необходимая продолжительность действия звукового сигнала?

4 Домашнее задание

4.1 Подготовить протокол согласно разд. 6. Обосновать требования и разработать систему пуска-остановки участка технологического процесса согласно темы комплексного задания. Выполнить ее текстовое описание, разработать блок-схему алгоритма. На основании этого составить его программную реализацию на языке *LD*. Программу снабдить необходимыми комментариями.

4.2 Программа выполняется по окончании лабораторного задания.

5 Лабораторное задание

5.1 Запустить среду разработки *CoDeSys*. Если в организаторе объектов автоматически приоткрывается проект, нужно закрыть его с помощью меню *File* → *Close*. Выполнить построение ("синтез") электрических принципиальных схем автоматического управления и сигнализации, а также проверку их алгоритмов функционирования ("анализ").

5.2 Порядок выполнения работы (использовать для разработки программ описание редактора языка *LD* из файла *CoDeSys_V23_RU.pdf*)

Собрать у рабочей области *CoDeSys* схему, которая реализует элементарную операцию "ИЛИ", и проверить ее работоспособность (рис. 2.1, а).

То же, операция "И" (рис. 2.1, б).

То же, операция "НЕ" (рис. 2.1, в).

То же, операция "Память" (рис. 2.1, д).

То же, операция "задержка" (рис. 2.1, г) используя таймер из библиотеки *Standart.lib*.

Собрать схему управления приводом через реле *K2* (рис. 2.2, а)

Собрать схему предупредительной сигнализации (рис. 2.2, б),

Собрать схему управления исполнительным механизмом ИМ (рис. 2.2, в).

5.3 Включить режим симуляции с помощью меню *Online* → *Simulation Mode*.

5.4 Подсоединить среду для симуляции в ПЛК с помощью меню *Online* → *Login* (Подключение). Обратите внимание на то, что во время подсоединения выполняется компиляция проекта.

5.5 Запустить проект на выполнение с помощью меню *Online* → *Run* (Старт).

5.6 Выполнить имитацию включения соответствующих кнопок, подтверждая это с помощью меню *Online* (Подключение) опцией *Write value* (Записать значение).

5.7 Во время функционирования проекта исследуйте процесс изменения переменных во всех блоках.

5.8 Выполнить контрольный пример.

Пример. Разработать программу управления соответственно алгоритму приведенному на рис. 5.1.

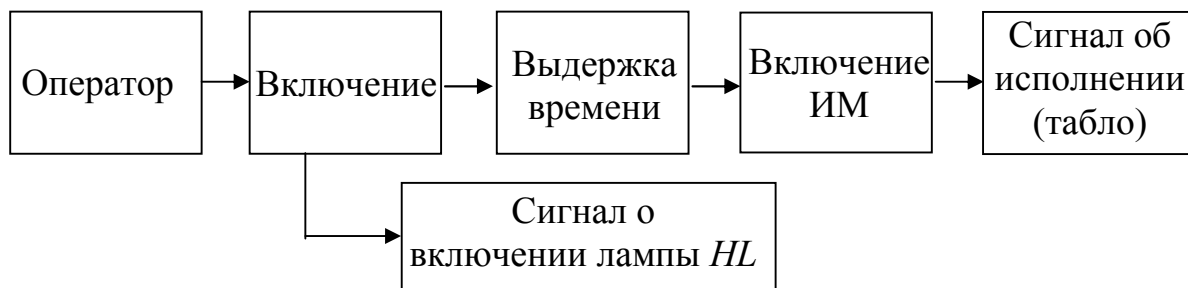


Рисунок 5.1 – Алгоритм включения механизмов оператором

Внимание!

При запуске программного проекта на тестирование или отладку необходимо выполнить подключение библиотек (*Standart.lib*, *UTIL.LIB* и др.) [5]. Стандартная библиотека *Standart.lib* содержит строковые функции, переключатели, детекторы импульсов, счетчики, таймеры. Библиотека *UTIL.LIB*, содержит регуляторы, генераторы сигналов, аналоговые компараторы и др. Для подключения библиотек необходимо в меню *Window* (Окно) (“*Window*”→”*Library Vanager*”→”*Additional library*”) выбрать мышью опцию *Manager library* (Менеджер библиотек) и далее двойным щелчком правой кнопки в верхнем пустом окне появляется список библиотек, из которых выбираем необходимую.

Содержание отчета

Обработка результатов исследования. Все схемы вычертить в протоколе.

Привести уравнение цепей и указать элементарные операции, выполняемые элементами схемы.

6 Содержание протокола

Протокол лабораторной работы “Программирование ПЛК. Изучение релейно-контактных схем (*LD*)” оформляется в отдельной тетради для лабораторных работ. Протокол должен содержать название лабораторной работы и ее цель, ответы на контрольные вопросы, ход выполнения домашнего и лабораторного заданий, комментарии к программам, разработанным и отлаженным во время выполнения лабораторной работы, блок-схемы программ, выводы.

7 Перечень рекомендованной литературы

1. Петров И.В. «Программируемые контроллеры. Стандартные языки и приемы прикладного программирования». – М.: ООО «СОЛОН-Пресс», 2004.
2. Методические пособия для лабораторных работ данного курса.
3. Конспект лекций по курсу МП и ПСА.
4. Веб-страница фирмы *Smart Software Solutions GmbH* производителя среды *CoDeSys* <http://www.3s-software.com/>
5. Веб-страница ПК "Пролог", поддержка среды *CoDeSys* на русском языке <http://www.codesys.ru/>