



Національний технічний університет України  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»



Кафедра автоматизації  
електромеханічних  
систем та електроприводу  
ФЕА

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»

# СИСТЕМИ АВТОМАТИЗАЦІЇ ЛАБОРАТОРНИЙ ПРАКТИКУМ. ЧАСТИНА 2

**Навчальний посібник**

Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського  
як навчальний посібник для здобувачів ступеня бакалавра  
за освітньою програмою «Електромеханічні системи автоматизації, електропривод та  
електромобільність»  
спеціальності 141 «Електроенергетика, електротехніка та електромеханіка»

Укладачі: С.О. Бур'ян, Г.Ю. Землянухіна, Р.С. Волянський

Електронне мережне навчальне видання

Київ  
КПІ ім. Ігоря Сікорського  
2023

Рецензент

*Троценко Є.О., канд. техн. наук, доц. каф. теоретичної  
електротехніки КПІ ім. Ігоря Сікорського*

Відповідальний  
редактор

*Красношанка Н.Д., канд. техн. наук, доц.*

*Гриф надано Методичною радою КПІ ім. Ігоря  
Сікорського (протокол № 5 від 23.02.2023 р.)  
за поданням Вченої ради факультету електроенерготехніки та автоматики  
(протокол № 7 від 30.01.2023 р.)*

Навчальний посібник включає методичні матеріали щодо виконання та оформлення лабораторних робіт з дисципліни “Системи автоматизації. Частина 2” для освітньої програми "Електромеханічні системи автоматизації, електропривод та електромобільність" всіх форми навчання спеціальності 141 "Електроенергетика, електротехніка та електромеханіка". Розглядаються питання виконання та оформлення лабораторних робіт, наведені приклади синтезу, програми та шляхи технічної реалізації систем автоматизації за заданими умовами роботи.

Реєстр. № 22/23-512. Обсяг 9,5 авт. арк.

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»  
проспект Перемоги, 37, м. Київ, 03056  
<https://kpi.ua>

Свідоцтво про внесення до Державного реєстру видавців,  
виготовлювачів і розповсюджувачів видавничої продукції ДК №  
5354 від 25.05.2017 р.

**ЗМІСТ**

Вступ.....	4
Лабораторна робота № 10.....	5
Лабораторна робота № 11.....	20
Лабораторна робота № 12.....	49
Лабораторна робота № 13.....	88
Лабораторна робота № 14.....	99
Лабораторна робота № 15.....	130
Лабораторна робота № 16.....	150
Лабораторна робота № 17.....	167
Лабораторна робота № 18.....	180
Заходи безпеки під час виконання лабораторних робіт в лабораторії «Автоматизації технологічних процесів, установок і комплексів» .....	205
Список рекомендованої літератури.....	207
Додаток А.....	208

## ВСТУП

Навчальний посібник містить опис лабораторних робіт, впровадження яких у освітній процес пов'язане з безперервним вдосконаленням елементної бази систем автоматизації, впровадженням нових технічних засобів, подальшим розвитком методів синтезу дискретних схем автоматики.

В навчальному посібнику, відповідно до силябусу освітнього компонента «Системи автоматизації-2», наведені методичні вказівки для виконання 9-ти лабораторних робіт. Особливістю даних робіт є те, що вони можуть проводитися як на реальному обладнанні лабораторії, так і в режимі симуляції у відповідному програмному забезпеченні.

Мета лабораторних робіт – закріпити теоретичні знання методів синтезу дискретних схем промислової автоматики та сучасної елементної бази пристроїв систем автоматизації, навчитися вирішувати практичні задачі автоматизації методами логічного синтезу, отримати навички реалізації та дослідження схем автоматики на сучасній елементній базі, що включає програмовані логічні контролери різної складності програмування.

У навчальному посібнику до кожної лабораторної роботи поданні основні теоретичні відомості, необхідні для підготовки, виконання та наступного захисту виконаної роботи. Крім того, посібник містить програму роботи, вказівки для її виконання, опис лабораторної установки, зміст звіту та контрольні запитання. Всі лабораторні роботи виконуються на сучасному обладнанні, що використовується у промислових системах автоматизації.

Всі лабораторні роботи виконуються класичним «бригадним» методом. Звіти з лабораторних робіт оформлюються згідно діючих стандартів до технічної документації.

## Лабораторна робота №10

### РІШЕННЯ ЛОГІЧНИХ ЗАДАЧ АВТОМАТИЧНОГО КЕРУВАННЯ НА БАЗІ ПРОГРАМОВАНОГО КОНТРОЛЕРА LOVATO LRD20RA024

Тривалість лабораторної роботи – 4 години.

Тривалість домашньої роботи – 4 години.

**Мета роботи** – вивчити будову та принцип дії програмованого контролера LOVATO LRD20RA024, практично відпрацювати правила програмування, запису та відпрацювання програм.

#### 10.1. Основні теоретичні відомості

Логічний програмований контролер призначений для реалізації логічних, часових та лічильних функцій керування виробничими механізмами та обладнанням за програмами, що записані у блоці пам'яті контролера.

Принцип дії контролера полягає у тому, що його процесор зчитує та виконує команди, які записано у запам'ятовуючому блоці, згідно з якими він приймає та опрацьовує сигнали, що надходять на входи, та забезпечує спрацьовування виходів.

Програмування контролера полягає у перетворенні алгоритму роботи схеми автоматичного керування у послідовність команд, що записуються у контролері. Система команд з їх умовними позначеннями, а також правила використання команд складають мову програмування контролера.

В ПЛК використовуються наступні позначення змінних: великі літери позначають нормально розімкнуті контакти, а маленькі – нормально замкнуті (інверсні), I1 – I8 – входи, M1 – M8 – проміжні змінні, T1 – T8 – таймери, Q1 – Q8 – виходи. Про подачу/зняття вхідних сигналів кнопками SB1-SB8 сигналізують лампи, які знаходяться над ними HL1-HL8. На виходах ПЛК знаходяться: Q1-Q4 – сигнальні лампи зеленого кольору, Q5-Q7 – котушки

магнітних контакторів (винесені на передню панель стенда), до контактів яких своєю чергу ввімкнені асинхронні двигуни, Q8 – звуковий сигнал. Прямий сигнал позначається великою літерою, інверсний – малою. При роботі з контролером потрібно пам'ятати, що записана на мові LD релейно-контактна схема не буде працювати, якщо рядок команди не заповнений з початку до кінця (три нормально замкнуті/нормально розімкнуті контакти та вихід). Роботу схеми можна бачити, як на стенді (засвічування сигнальних ламп, спрацьовування контакторів, подача звукового сигналу) так і на дисплеї ПЛК (замикаються/розмикаються відповідні контакти схеми).

Для того, щоб запрограмувати таймер, на 5 с наприклад, на ньому потрібно натиснути кнопку ОК, після цього на екрані з'явиться меню таймера :

$$1 \left[ \begin{array}{cc} 1 & \\ 00. & 00 \end{array} \right] T1$$

потрібно його змінити таки чином, щоб отримати наступне :

$$2 \left[ \begin{array}{cc} 1 & \\ 005. & 0 \end{array} \right] T1$$

Для цього на будь-якій цифрі потрібно натиснути кнопку SEL, і за допомогою кнопок позиціонування вибрати потрібні значення.

Для прикладу запишемо рівняння  $T_4 = p_1 p_2 p_3$  на мові Ladder Diagram, затримку часу встановлюємо на 3с.:

$$M1 - M2 - M3 - T4$$

Меню таймера:

$$2 \left[ \begin{array}{cc} 1 & \\ 003. & 0 \end{array} \right] T4$$

Для того, щоб записати в програму інформацію, про вмикання/вимикання тригера потрібно біля його позначення встановити стрілку ввєрх/вниз, що буде відповідати вмиканню/вимиканню тригера відповідно. Для прикладу запишемо рівняння  $S_{p_2} = I1 \cdot \bar{p}_1 \cdot \bar{p}_3$  на мові Ladder Diagram :

$$I1 - m1 - m3 - \uparrow M2$$

Де тригери р позначені проміжними змінними М. Стрілка ввєрх вказує, що це сигнал на вмикання тригера.

Для початку внесення програми в ПЛК потрібно ввімкнути автомат QF1, при цьому на дисплеї з'явиться меню, де буде відображатися стан входів (наявність/відсутність сигналу), стан програми ПЛК (RUN/STOP). Якщо вона внесена попередньо та поточний час. Для початку програмування потрібно натиснути кнопку ESC на панелі прибору, на екрані з'явиться меню, навігація по якому здійснюється за допомогою кнопок  $\uparrow$ ,  $\downarrow$ ,  $\leftarrow$ ,  $\rightarrow$ , якщо в третьому рядку стоїть стан STOP, то його потрібно змінити на RUN (натиснути кнопку ОК, та вибрати параметр YES), тому що в іншому випадку робота з програмою буде неможлива. Наступним кроком потрібно очистити пам'ять ПЛК, для цього потрібно вибрати пункт CLEAR PROG, та очистити пам'ять. Для початку внесення програми в пам'ять ПЛК потрібно вибрати пункт LADDER, натиснути кнопку ОК, після цього з'явиться чистий екран, це масив для введення програми.

## 10.2. Програма роботи

За варіантом, наведеним у таблиці 10.1, виконати наступні завдання:

1. Виконати логічний синтез схеми керування, та отримати логічні вирази, що описують схему (**виконується вдома під час СРС**).
2. За отриманими алгебричними виразами у пункті 1 скласти, релейно-контактної схеми для логічного контролера на мові *LD*.
3. Виконати адресацію вхідних, вихідних і проміжних змінних.
4. Записати програму до контролера.
5. Перевірити відпрацювання заданих умов роботи (**викладач підтверджує своїм підписом у протоколі працездатність програми**).
6. Оформити звіт по лабораторній роботі та зробити висновки.

Таблиця 10.1 – Варіанти завдань

Бригада	Номер завдання
1	1, 7, індивідуальне
2	2, 8, індивідуальне
3	3, 9, індивідуальне
4	4, 10, індивідуальне
5	5, 11, індивідуальне
6	6, 12, індивідуальне

Згідно номеру бригади вибрати номери завдань. **Після виконання завдань 1 та 2 необхідно отримати у викладача індивідуальне завдання.**

### Завдання № 1

Три лампочки HL1, HL2, HL3 вмикаються та вимикаються трьома вимикачами SA1, SA2, SA3. Лампочка HL1 повинна світитися тільки у тому разі, коли замкнутий будь-який один вимикач, HL2 – коли замкнуті перший та третій або перший та другий вимикачі, HL3 – якщо замкнуті усі три вимикачі або другий та третій вимикачі.

### Завдання № 2

Схема перетворює трирозрядний код Грея у двійковий код згідно з таблицею істинності (таблиця 10.2).

Таблиця 10.2 – Таблиця істинності коду Грея

Десяткове число	Вхідні сигнали (код Грея)			Вихідні сигнали (двійковий код)		
	$a_3$	$a_2$	$a_1$	$b_3$	$b_2$	$b_1$
0	0	0	0	0	0	0
1	0	0	1	0	0	1
2	0	1	1	0	1	0
3	0	1	0	0	1	1
4	1	1	0	1	0	0
5	1	1	1	1	0	1
6	1	0	1	1	1	0
7	1	0	0	1	1	1

**Завдання № 3**

Схема порівнює за модулем два дворозрядних двійкових числа  $A = a_2a_1$  та  $B = b_2b_1$ , де  $a_i, b_i$  – цифри (0 або 1) відповідальних розрядів чисел  $A$  і  $B$ . Вихідні сигнали схеми:  $f_1 = 1$ , якщо  $A > B$ ;  $f_2 = 1$ , якщо  $A < B$ ;  $f_3 = 1$ , якщо  $A = B$ .

**Завдання № 4**

Схема має три вхідні ( $a, b, c$ ) і два вихідні ( $f_1, f_2$ ) сигнали. Сигнал  $f_1 = 1$ , якщо парна кількість вхідних сигналів дорівнює одиниці, сигнал  $f_2 = 1$ , якщо непарна. Нуль вважати парним числом.

**Завдання № 5**

Схема виконує вибір “за більшістю” з чотирьох сигналів  $a, b, c, d$ . Значення сигналу на виході схеми зберігається із значенням більшості вхідних сигналів. Якщо два вхідних сигнали дорівнюють нулю, а два інших – одиниці, то вихідний сигнал дорівнює нулю.

**Завдання № 6**

Схема має три вимикачі  $SA1, SA2, SA3$  і чотири лампочки  $HL1, HL2, HL3, HL4$ . Лампочка  $HL1$  повинна світитись тільки у тому разі коли замкнутий будь-який один вимикач; лампочка  $HL2$  – якщо замкнуті будь-які два вимикачі; лампочка  $HL3$  – якщо замкнуті усі три вимикачі; лампочка  $HL4$  – якщо замкнуті будь-які два або усі три вимикачі.

**Завдання № 7**

Керування трьома двигунами  $M1, M2, M3$  здійснюється за допомогою кнопок «Пуск» і «Стоп». При натисненні кнопки «Пуск» вмикається на 5 с. звукова сигналізація, потім із затримкою часу 2 с. вмикається двигун  $M1$ , потім через 4 с вмикається двигун  $M2$  та через 8 с – двигун  $M3$ . При натисненні кнопки «Стоп» знову вмикається звукова сигналізація на 5 с., потім двигун  $M1$  вмикається без затримки часу, потім через 3 с вмикається двигун  $M2$  та через 4 с – двигун  $M3$ .

### **Завдання № 8**

Двигун М вмикається через 5 с після натиснення кнопки “Пуск 1” і вимикається через 3 с після натиснення кнопки “Стоп 1”. Кнопками “Пуск 2” і “Стоп 2” двигун вмикається і вимикається без затримки часу. Зміна режиму роботи здійснюється тумблером. Ввімкнений стан тумблера – відповідає першому режиму, вимкнений – другому. Передбачити світлову та звукові сигналізації включення та виключення двигуна.

### **Завдання № 9**

Схема керування двигунами М1, М2, М3 працює наступним чином. При натисненні першої кнопки вмикається двигун М1, при натисненні другої кнопки – вмикається двигун М2 і вимикається двигун М1, при натисненні третьої – вмикається двигун М3 і вимикається М2. При натисненні четвертої кнопки вимикається двигун М3. Кнопки із самоповерненням і натискаються по черзі. Схема не повинна реагувати на порушення черговості натискання кнопок. Передбачити звукову сигналізацію циклу вмикання/вимикання двигунів.

### **Завдання № 10**

Схема керування двома двигунами М1 і М2 працює в двох режимах: перший – при натисненні кнопки “Пуск 1” вмикається двигун М1 без затримки часу, а потім через 5 с вмикається двигун М2; при натисненні кнопки “Стоп 1” вимикається двигун М2 без затримки часу, а потім через 4 с – двигун М1; другий – двигун М1 ввімкнений тільки під час натиснення кнопки “Пуск 2”, а двигун М2 – тільки під час натиснення кнопки “Пуск 3”. При відпусканні цих кнопок обидва двигуни вимикаються. Перехід з першого режиму в другий здійснюється тумблером, замкнутий стан якого відповідає першому режиму, розімкнутий – другому. Передбачити звукову сигналізацію циклу вмикання/вимикання двигунів.

### **Завдання № 11**

Трьома механізмами з двигунами М1, М2, М3 керують за допомогою кнопок “Пуск” і “Стоп”. При натисненні кнопки “Пуск” спочатку вмикається звукова попереджувальна сигналізація, а потім через час  $t_0$  – двигуни в

послідовності М1 – М2 – М3 з затримкою 2 с між вмиканнями. Після запуску останнього двигуна сигналізація вимикається. При натисненні кнопки “Стоп” двигуни вимикаються без затримки часу в зворотному порядку: М3 –М2 – М1 затримкою 3 с між вмиканнями. Передбачити звукову сигналізацію циклу вмикання/вимикання двигунів.

### **Завдання № 12**

Трьома механізмами з двигунами М1, М2, М3 керують за допомогою кнопок “Пуск” і “Стоп”. При натисненні кнопки “Пуск” вмикається перший механізм, набирає швидкість і спрацьовує пов’язане з ним відцентрове реле, яке подає команду на пуск двигуна М2. Другий механізм набирає швидкість і його відцентрове реле подає команду на пуск двигуна М3. Якщо через час  $\Delta t$  після надходження команди “Пуск” усі механізми не наберуть потрібної швидкості (не спрацюють усі три відцентрові реле), то усі двигуни вимикаються. Нормальна зупинка здійснюється кнопкою “Стоп”. Після її натиснення вимикаються двигуни усіх механізмів без затримки часу. Передбачити звукову сигналізацію циклу вмикання/вимикання двигунів.

## **10.3. Опис лабораторної установки**

Функціональну схему лабораторного стенду представлено на рисунку 10.1. Даний лабораторний стенд має відповідати наступним вимогам: простота та зручність експлуатації; наочність монтажу; відсутність потреби в додатковому обладнанні. За допомогою вбудованого в програмований логічний контролер інтерфейсу та мови програмування Ladder Diagram складається та відлагоджується робоча програма, яка описує роботу системи відповідно до заданої умови роботи релейно-контактної схеми.

В залежності від завдання, за допомогою кнопок SB1..SB8 подаються вхідні сигнали на ПЛК. Про подачу даних сигналів сповіщають сигнальні лампи HL1-HL8, які розміщені над кнопками.

Після подачі сигналів від ПЛК можна бачити роботу схеми, як на стенді (засвічування сигнальних ламп, спрацьовування контакторів, подача звукового сигналу) так і на дисплеї програмованого логічного контролера.

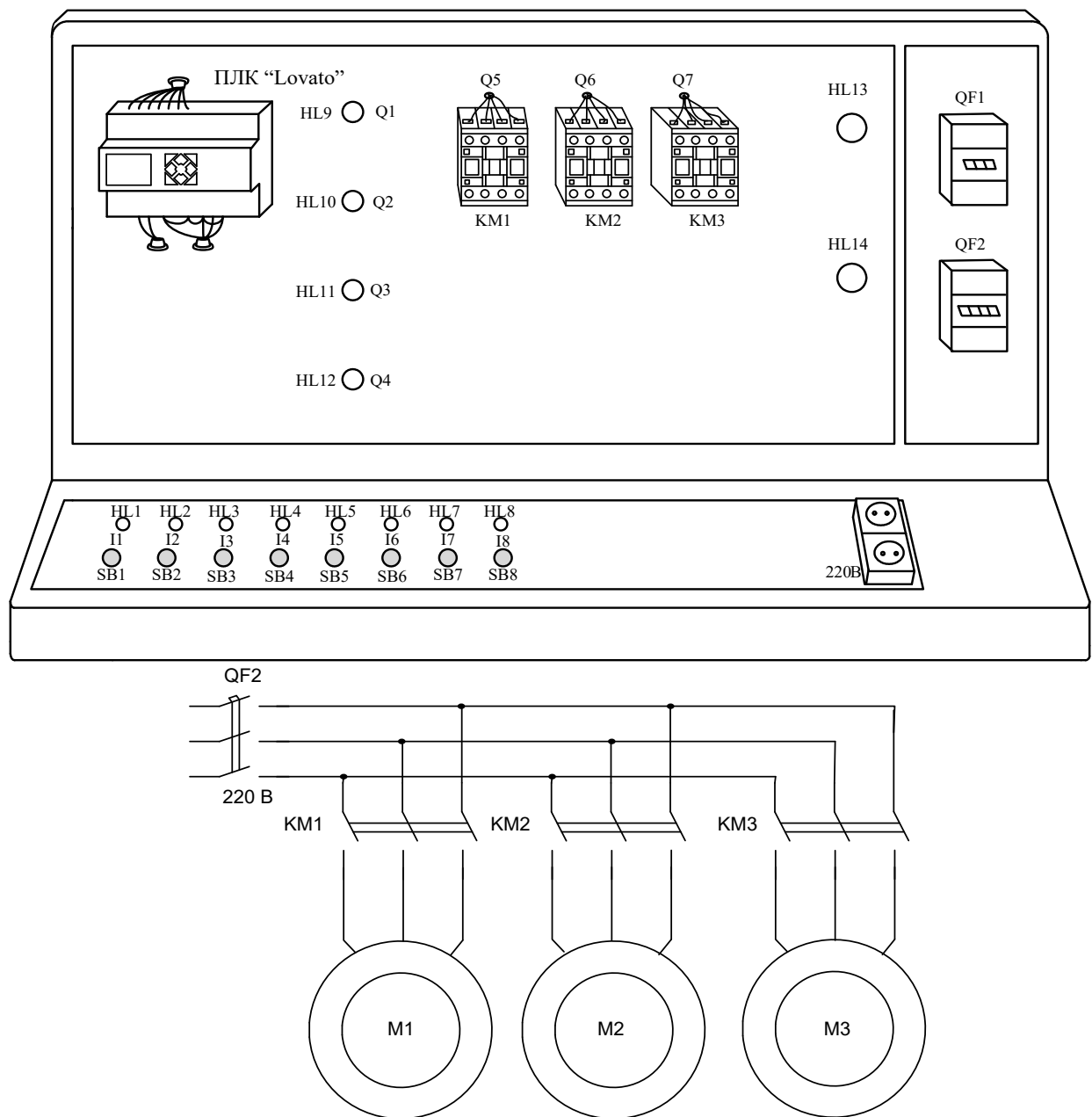


Рисунок 10.1 – Функціональна схема лабораторного стенду

На панелі стенду представлено: програмований логічний контролер «Lovato», сигнальні лампи 220 та 24 В, асинхронні двигуни М1, М2, М3, магнітні контактори КМ1–КМ3, автоматичні вимикачі QF1, QF2 та кнопки з самоблокуванням SB1-SB8.

Використаний у лабораторному стенді програмований логічний контролер «Lovato» зображений на рисунку 10.2.

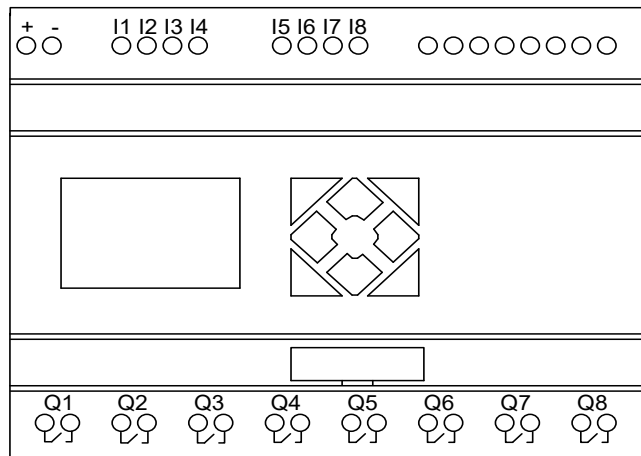


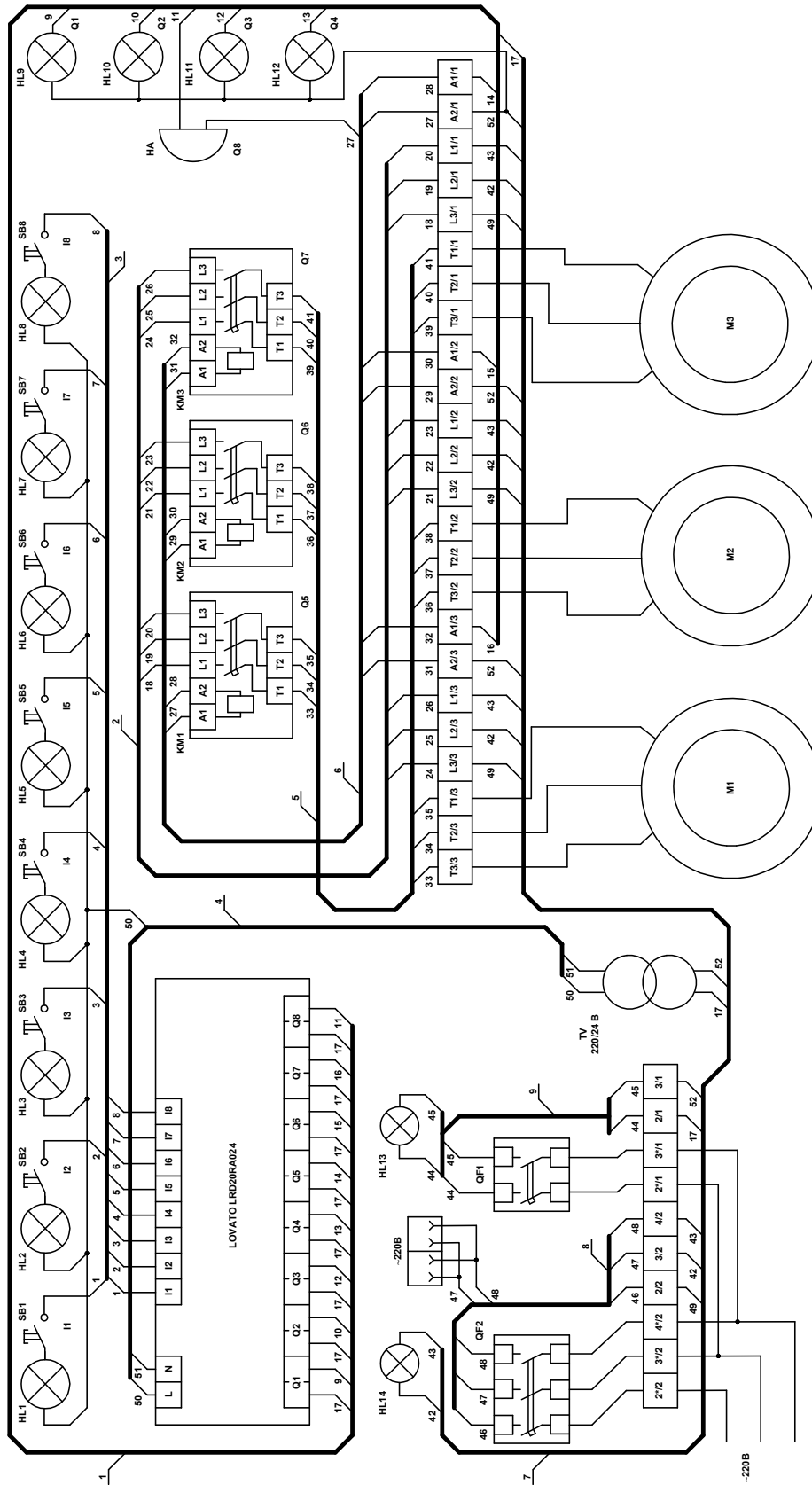
Рисунок 10.2 – Програмований логічний контролер «Lovato»

На клеми «+» «-» подається живлення ПЛК, на входи I1-I8 подаються вхідні сигнали (їх формують кнопки SB1..SB8). До виходів контролера підключаються сигнальні лампи (до виходів Q1-Q4 підключені лампи HL9-HL12), магнітні контактори (до виходів Q5-Q7 підключені контактори КМ1-КМ3) та звукова сигналізація (до виходу Q8 підключений звукова сигналізація НА).

Схема підключень стенда зображена на рисунку 10.3, електрична принципова схема зображена на рисунку 10.4. Через автоматичний вимикач QF1 подається живлення на двигуни M1, M2, M3. QF2 подає живлення на трансформатор напруги (TV), магнітні контактори (КМ1-КМ3), сигнальні лампи (HL9-HL14) та кнопки (SB1-SB8). При вмиканні тумблерів спрацьовує світлова сигналізація (HL13-HL14). З трансформатора напруга протікає до ПЛК та сигнальних ламп (HL1-HL8). До ПЛК на входи I1-I8 підключені кнопки SB1-SB8 послідовно з сигнальними лампами HL1-HL8, тобто вхідним сигналом ПЛК є натиснення кнопки, при якому засвічується відповідна сигнальна лампа. До виходів Q1-Q4 підключені сигнальні лампи HL9-HL12, тобто вони засвічуються при поданні сигналів від ПЛК на виходи Q1-Q4. При активізації виходів Q5-Q7 спрацьовують магнітні контактори КМ1-КМ3, що в свою чергу замикають контакти в колі двигуна і він починає обертатися. До виходу Q8

підключена звукова сигналізація НА. Монтаж всього обладнання здійснений через клеми.

Схема підключень лабораторного обладнання



- 1. Дажут, 9 проводів типу МГШВ 0,5 кв.мм
- 2. Дажут, 9 проводів типу МГШВ 1,5 кв.мм
- 3. Дажут, 8 проводів типу МГШВ 0,5 кв.мм
- 4. Дажут, 3 провора типу МГШВ 0,5 кв.мм
- 5. Дажут, 9 проводів типу МГШВ 1,5 кв.мм
- 6. Дажут, 6 проводів типу МГШВ 1,5 кв.мм
- 7. Дажут, 5 проводів типу МГШВ 1,5 кв.мм
- 8. Дажут, 3 провора типу МГШВ 1,5 кв.мм
- 9. Дажут, 2 провора типу МГШВ 1,5 кв.мм

Рисунок 10.3 – Схема підключень лабораторного стенду

Схема електрична принципова

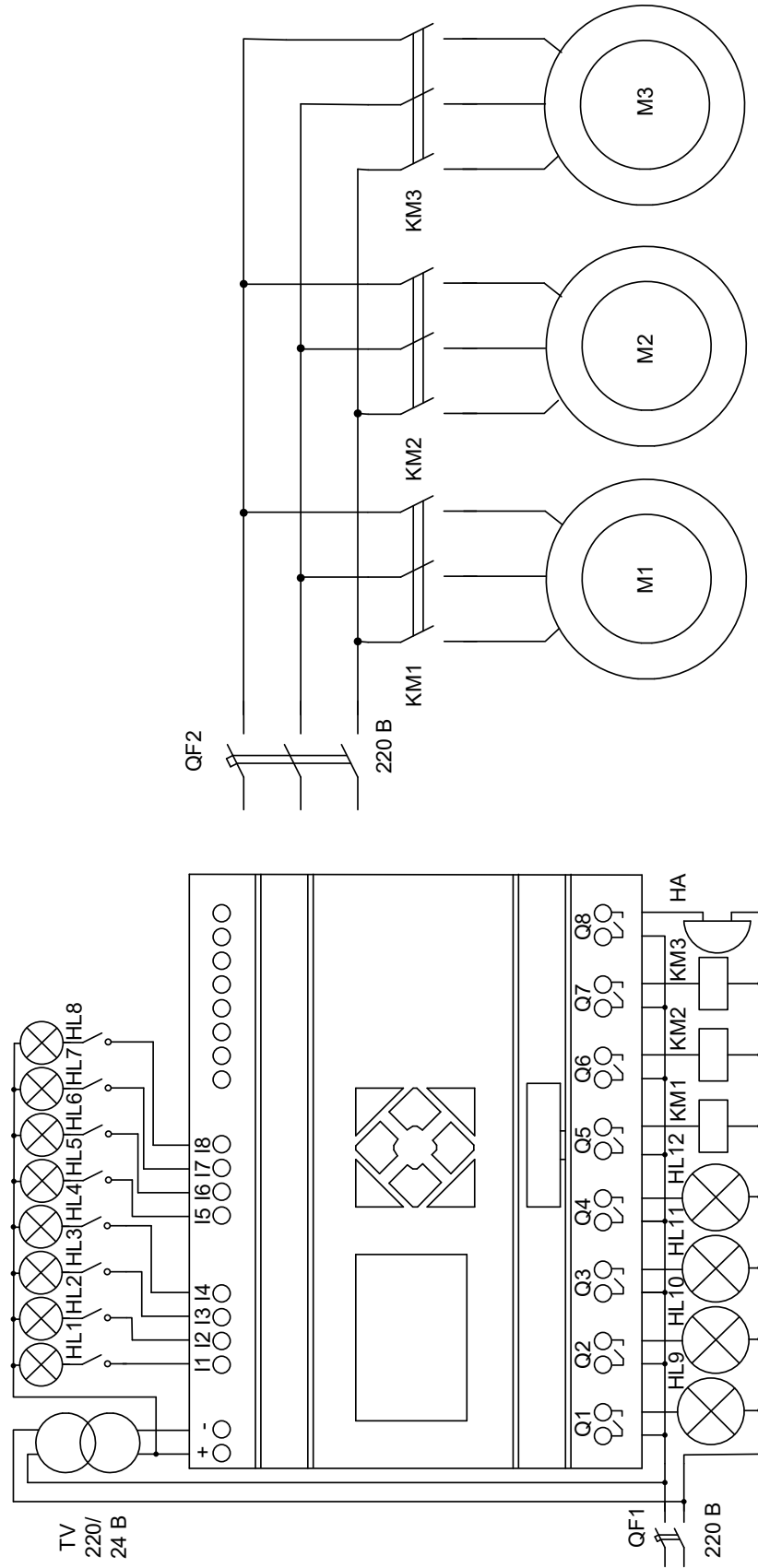


Рисунок 10.4 – Електрична принципова схема лабораторного стану

### 10.4. Методичні вказівки до виконання роботи

Приклад 1. Керування трьома двигунами М1, М2, М3 здійснюється за допомогою кнопок «Пуск» і «Стоп». При натисненні кнопки «Пуск» вмикається двигун М1 без затримки часу, потім через 3 с вмикається двигун М2 та через 6 с – двигун М3. При натисненні кнопки «Стоп» двигун М1 вимикається без затримки часу, потім через 4 с вимикається двигун М2 та через 8 с – двигун М3. При складанні програми передбачити звукову сигналізацію ввімкнення/вимкнення циклу на 1с.

Вводимо такі позначення : Q8 – звуковий сигнал, Q5 – Q7 – двигуни М1 – М3 відповідно. Графоперехід за умовами роботи схеми представлений на рисунок 10.5.

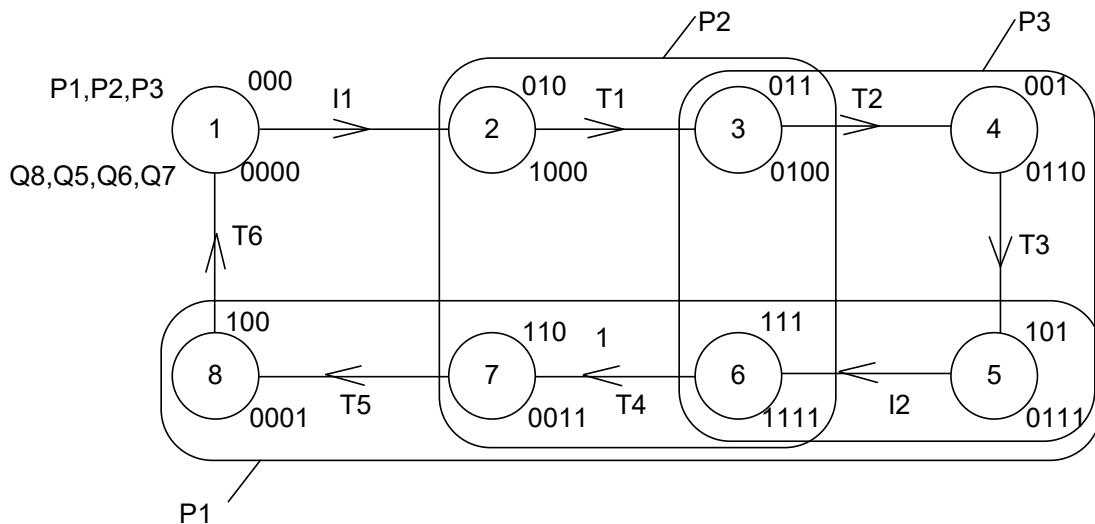


Рисунок 10.5 – Графоперехід за умовами роботи схеми

Запишемо рівняння:

$$S_{p1} = T3 \cdot p_3 \cdot \bar{p}_2; \quad R_{p1} = T6 \cdot \bar{p}_2 \cdot \bar{p}_3;$$

$$S_{p2} = I1 \cdot \bar{p}_1 \cdot \bar{p}_3 + I2 \cdot p_1 \cdot p_3; \quad R_{p2} = T5 \cdot p_1 \cdot \bar{p}_3 + T2 \cdot p_3 \cdot \bar{p}_1;$$

$$S_{p3} = T1 \cdot p_2 \cdot \bar{p}_1; \quad R_{p3} = T4 \cdot p_1 \cdot p_2;$$

$$f_1 = \bar{p}_1 p_2 \bar{p}_3 + p_1 p_2 p_3; \quad f_2 = \bar{p}_1 p_2 p_3 + \bar{p}_1 \bar{p}_2 p_3 + p_1 \bar{p}_2 p_3 + p_1 p_2 p_3;$$

$$f_3 = \bar{p}_1 \bar{p}_2 p_3 + p_1 \bar{p}_2 p_3 + p_1 p_2 p_3 + p_1 p_2 \bar{p}_3; \quad f_4 = p_1 \bar{p}_2 p_3 + p_1 p_2 p_3 + p_1 p_2 \bar{p}_3 + p_1 \bar{p}_2 \bar{p}_3;$$

$$T_1 = \bar{p}_1 p_2 \bar{p}_3; \quad T_2 = \bar{p}_1 p_2 p_3; \quad T_3 = \bar{p}_1 \bar{p}_2 p_3; \quad T_4 = p_1 p_2 p_3; \quad T_5 = p_1 p_2 \bar{p}_3;$$

$$T_6 = p_1 \bar{p}_2 \bar{p}_3.$$

Проведемо адресацію змінних :

$$\begin{array}{ll} f1 \rightarrow Q8 & I1 \rightarrow I1 \\ f2 \rightarrow Q5 & I2 \rightarrow I2 \\ f3 \rightarrow Q6 & f4 \rightarrow Q7 \end{array}$$

Запишемо дані рівняння в ПЛК:

$T3 - M3 - m2 - \uparrow M1$	$m1 - m2 - M3 \square (Q6)$
$T6 - m2 - m3 - \downarrow M1$	$M1 - m2 - M3 \square$
$I1 - m1 - m3 \square \uparrow M2$	$M1 - M2 - M3 \square$
$I2 - M1 - M3 \square$	$M1 - M2 - m3 \square$
$T5 - M1 - m3 \square \downarrow M2$	$M1 - m2 - M3 \square (Q7)$
$T2 - m1 - M3 \square$	$M1 - M2 - M3 \square$
$T1 - M2 - m1 - \uparrow M3$	$M1 - M2 - m3 \square$
$T4 - M1 - M2 - \downarrow M3$	$M1 - m2 - m3 \square$
$m1 - M2 - m3 \square (Q8)$	$m1 - M2 - m3 - T1$
$M1 - M2 - M3 \square$	$m1 - M2 - M3 - T2$
$m1 - M2 - M3 \square (Q5)$	$m1 - m2 - M3 - T3$
$m1 - m2 - M3 \square$	$M1 - M2 - M3 - T4$
$M1 - m2 - M3 \square$	$M1 - M2 - m3 - T5$
$M1 - M2 - M3 \square$	$M1 - m2 - m3 - T6$

Приклад 2. Три лампочки HL1, HL2, HL3 вмикаються та вимикаються трьома вимикачами SA1, SA2, SA3. Лампочка HL1 повинна світитися тільки у тому разі, коли замкнутий будь-який один вимикач, HL1 – коли замкнуті будь-які два вимикачі, HL3 – якщо замкнуті усі три вимикачі.

Таблиця 10.3 – Таблиця істинності

SA1	SA2	SA3	HL1	HL2	HL3
0	0	0	0	0	0
0	0	1	1	0	0
0	1	0	1	0	0
0	1	1	0	1	0
1	0	0	1	0	0
1	0	1	0	1	0
1	1	0	0	1	0
1	1	1	0	0	1

Запишемо рівняння :

$$HL1 = \overline{SA1} \cdot \overline{SA2} \cdot SA3 + \overline{SA1} \cdot SA2 \cdot \overline{SA3} + SA1 \cdot \overline{SA2} \cdot \overline{SA3}$$

$$HL2 = \overline{SA1} \cdot SA2 \cdot SA3 + SA1 \cdot \overline{SA2} \cdot SA3 + SA1 \cdot SA2 \cdot \overline{SA3}$$

$$HL3 = SA1 \cdot SA2 \cdot SA3$$

Проведемо адресацію змінних :

$$HL1 \rightarrow Q1 \quad SA1 \rightarrow I1$$

$$HL2 \rightarrow Q2 \quad SA2 \rightarrow I2$$

$$HL3 \rightarrow Q3 \quad SA3 \rightarrow I3$$

Запишемо рівняння на мові LD :

$$Q1 = \overline{I1} \cdot \overline{I2} \cdot I3 + \overline{I1} \cdot I2 \cdot \overline{I3} + I1 \cdot \overline{I2} \cdot \overline{I3}$$

$$Q2 = \overline{I1} \cdot I2 \cdot I3 + I1 \cdot \overline{I2} \cdot I3 + I1 \cdot I2 \cdot \overline{I3}$$

$$Q3 = I1 \cdot I2 \cdot I3$$

Запишемо програму, яку необхідно занести в ПЛК :

$$i1 - i2 - I3 \square (Q1)$$

$$i1 - I2 - i3 \square$$

$$I1 - i2 - i3 \square$$

$$i1 - I2 - I3 \square (Q2)$$

$$I1 - i2 - I3 \square$$

$$I1 - I2 - i3 \square$$

$$I1 - I2 - I3 - (Q3)$$

#### **10.4. Зміст звіту**

- 1) Титульний аркуш.
- 2) Мета роботи, програма роботи.
- 3) Опис та схема лабораторного стенду.
- 4) Умови роботи схем.
- 5) Логічний синтез схем керування.
- 6) Адресування входів, виходів, таймерів, комірок пам'яті.
- 7) Програма на мові LD.
- 8) Висновок про відповідність роботи заданим умовам.

#### **10.5 Контрольні запитання**

1. Яке призначення логічного програмованого контролера?
2. У чому полягає принцип дії контролера?
3. Які позначення використовуються в ПЛК?
4. Яким чином живиться контролер на лабораторному стенді?
5. Якою мовою здійснюється програмування ПЛК?

## Лабораторна робота №11

# ДОСЛІДЖЕННЯ СИСТЕМ ВІДПРАЦЮВАННЯ ЗАДАНИХ ТРАЕКТОРІЙ НА ОСНОВІ ПЕРЕТВОРЮВАЧА ЧАСТОТИ ALTIVAR ТА ПРОГРАМОВАНОГО ЛОГІЧНОГО КОНТРОЛЕРА ZELIO LOGIC

Тривалість лабораторної роботи – 4 години.

Тривалість домашньої роботи – 4 години.

**Мета роботи** – вивчити будову та принцип дії програмованого контролера ZELIO LOGIC SR3B102BD, навчитися програмувати у програмному забезпеченні Zelio Soft 2 та складати і налагоджувати програми для відпрацювання заданих траєкторій.

### 11.1. Основні теоретичні відомості

Складання та відлагодження робочої програми відбувається в середовищі *Zelio soft 2*, та використовуючи програмування за допомогою релейної логіки LD.

*Zelio soft 2* – це інструмент програмування промислових комп'ютерів, який має режим моделювання та моніторингу, тестування програми у реальному часі з підключенням інтелектуальних реле до ПК або без підключення.

LD (скорочення від *Ladder Diagram*) – мова релейної логіки. Призначена для програмування програмованих логічних контролерів (ПЛК). Синтаксис мови зручний для заміни логічних схем, виконаних на релейній техніці. Орієнтований на інженерів з автоматизації, які працюють на промислових підприємствах. Забезпечує наочний інтерфейс логіки роботи контролера, який полегшує не лише завдання для програмування і введення в експлуатацію, але і швидкий пошук неполадок в обладнанні, що підключається до контролера.

Програма мовою релейного логіки має наочний і інтуїтивно зрозумілий інженерам-електрикам графічний інтерфейс, який представляє логічні операції, як електричний ланцюг із замкнутими і розімкненими контактами. Протікання або відсутність струму в цьому ланцюзі відповідає результату логічної операції (*true* - якщо струм тече; *false* - якщо струм не тече).

Основними елементами мови є контакти, які можна образно уподібнити парі контактів реле або кнопки. Пара контактів ототожнюється з логічною змінною, а стан цієї пари - зі значенням змінної.

Розрізняються нормально замкнуті і нормально розімкнуті контактні елементи, які можна зіставити з нормально замкнутими і нормально розімкнутими кнопками в електричних ланцюгах. Нижче вказані вигляди контактів, як вони виглядають на дисплеї ПЛК.

—| |— Нормально розімкнений контакт розімкнений при значенні *false*, призначеної йому змінної і замикається при значенні *true*.

—|/|— Нормально замкнутий контакт, навпаки, замкнутий, якщо змінна має значення *false*, і розімкнений, якщо змінна має значення *true*.

—( )— Підсумок логічного кола копіюється в цільову змінну, яка називається котушка (англ. *coil*). Це слово має узагальнений образ виконавчого пристрою, тому в російськомовній документації зазвичай говорять про вихід кола, хоча можна зустріти і приватні значення терміна, наприклад котушка реле.

### **Створення нового проекту.**

Даний пункт призначений для ознайомлення з середовищем програмування *Zelio soft 2*, основами програмування *Zelio Logic* і містить опис послідовності дій для програмування контролера. Приклад розроблений на базі *Zelio Logic SR* і дозволяє перевірити в роботі програму не тільки в режимі симуляції, але і автономно – в самому контролері.

*Створення проекту програми.* При створенні проекту використовується мова релейних діаграм *LD (Ladder Diagram)*, який реалізує структури, подібні електричним ланцюгам в комутаційній автоматичі.

Користувач запускає *Zelio soft 2* послідовним вибором додатків:

Новий проект відкривається з головного меню: Create new program. У вікні (рис 11.1) вибирається тип модуля, *10 I/O with extensions* лівою кнопкою миші натискаємо на необхідний модуль.

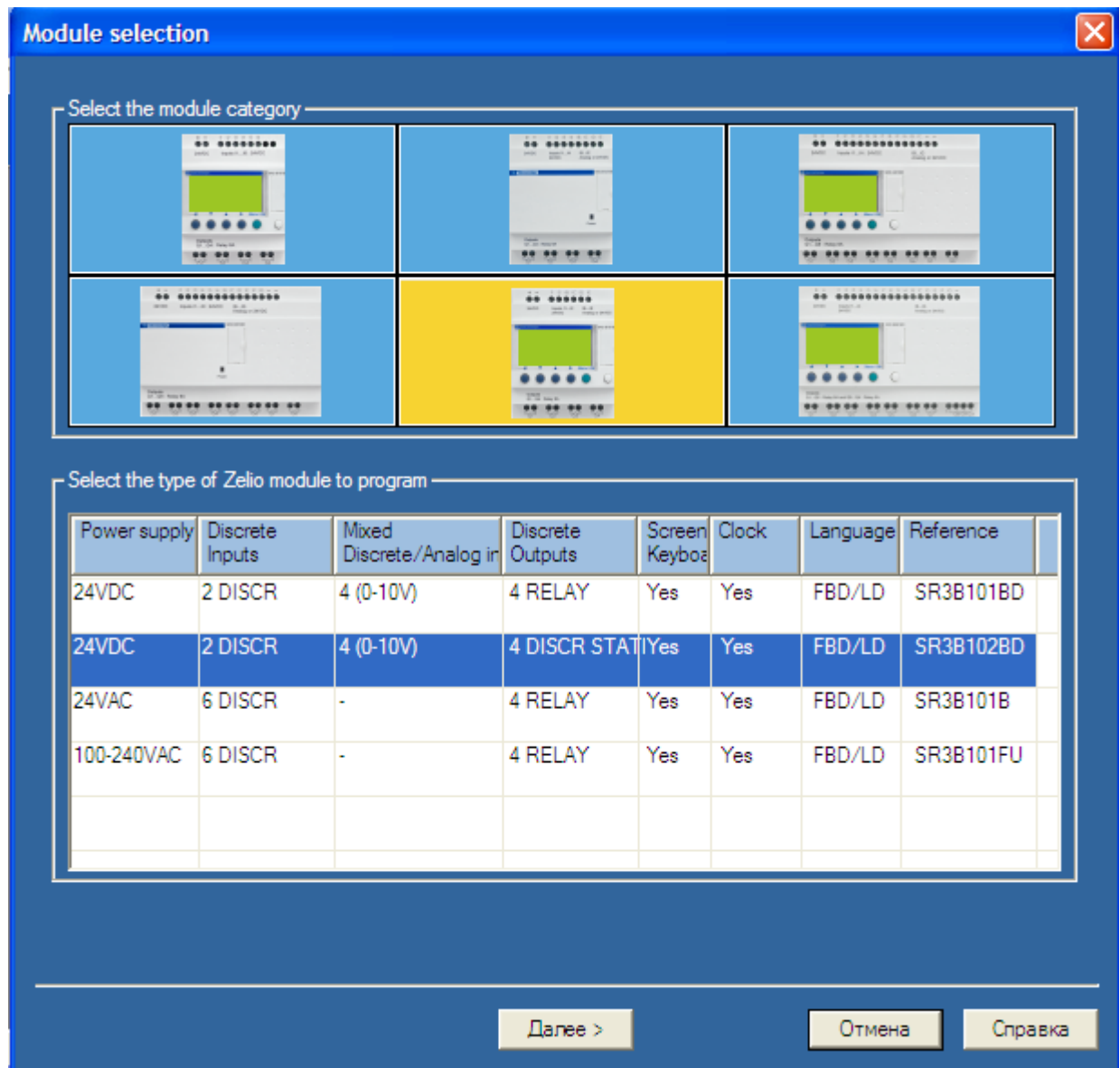


Рисунок 11.1 – Вибір типу модуля у вікні «Module Selection»

Далі вибираємо тип контролера, а саме SR3B102BD, та натискаємо кнопку далі. Після цього переходимо до меню вибору модуля розширення. Оскільки у данному випадку модуль розширення відсутній, то вибираємо перший та натискаємо далі.

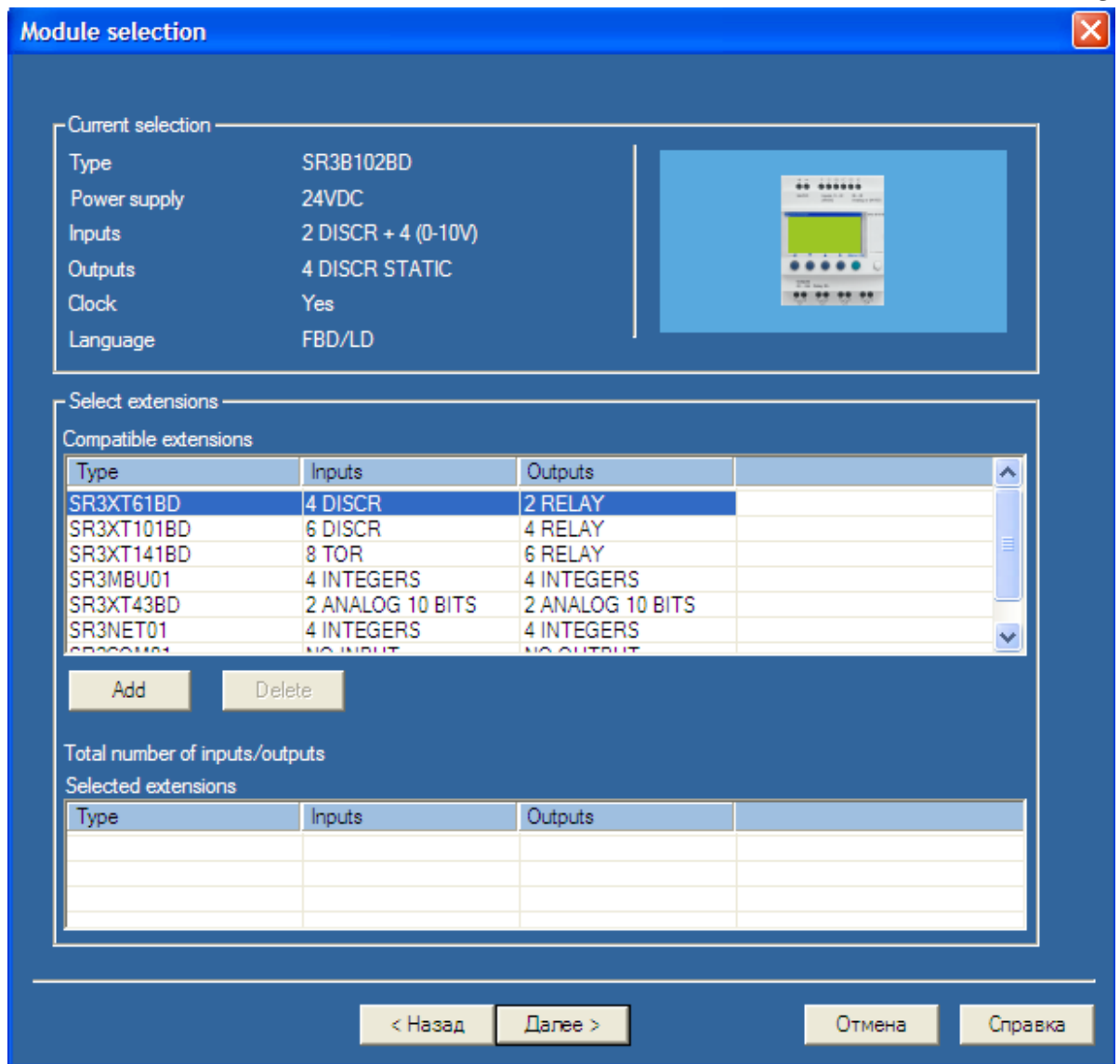


Рисунок 11.2 – Вибір модуля розширення

Далі і переходимо до вибору типу програмування FBD або LD, як показано на рис 11.3.

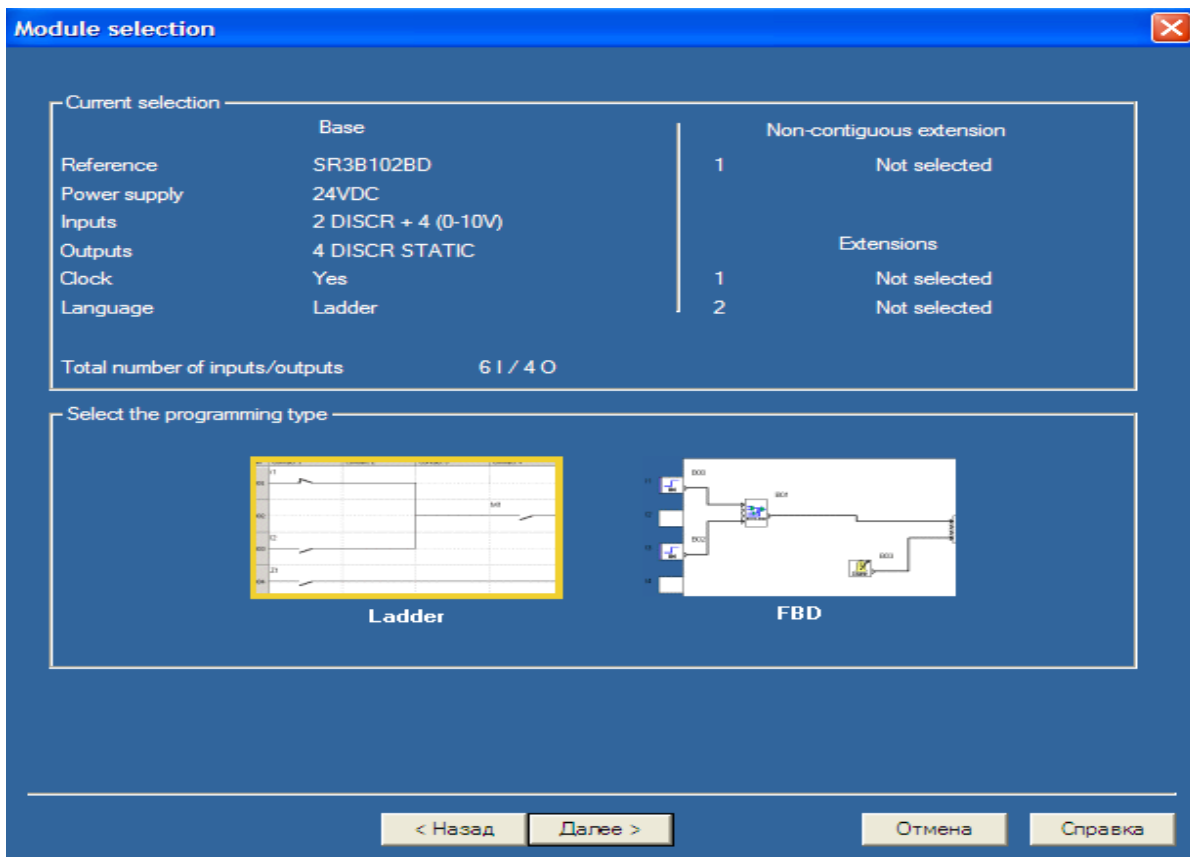


Рисунок 11.3 – Вибір типу програмування

На рисунках 11.4 – показано як виглядає середовище програмування LD.

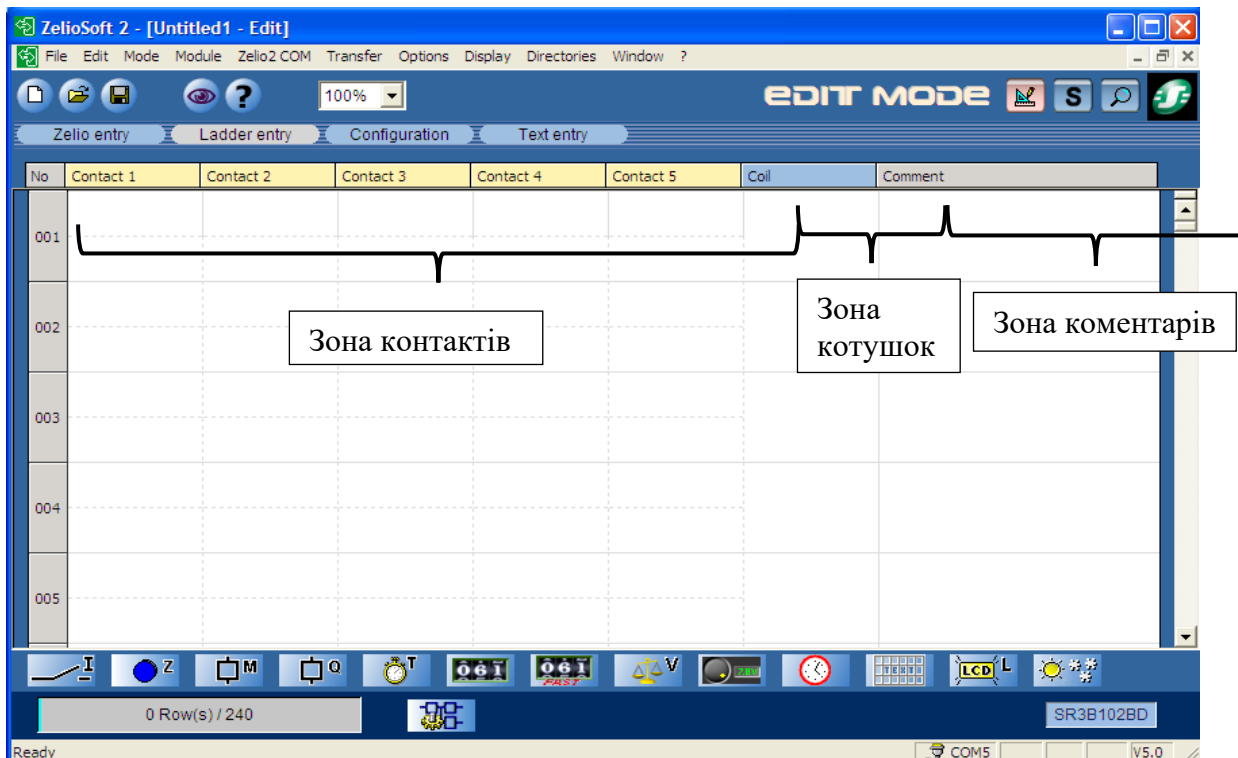


Рисунок 11.4 – Вигляд робочого середовища LD

Робоче поле поділяється на декілька зон, як показано на рис 11.4. На зону Contact призначена для контактів. У робочому середовищі може бути

максимум 5 контактів в одному рядку. Coil – це зона для створення катушок. В одному рядку може бути розміщена одна катушка. Також є зона для коментарів Comment у якій можна писати коментарі.

**Створення програми на мові LD.** При написанні програми в робочій зоні вкладки LD складається релейно-контакторна схема, як показано на рисунку 11.5. При виборі елемента схеми необхідно вказувати до якого вхідного порту він буде прив'язаний. Як показано на рис 11.6. при програмуванні у середовищі zelio soft 2 використовується наступні операнди:

I - входи

Q - виходи

M – комірки пам'яті

T - таймер

S – катушка Set

R - катушка Reset

Інверсні виходи позначаються малими літерами. Перед програмуванням логічних рівнянь у контролер необхідно зробити заміни змінних у відповідності до операндів, наприклад, вираз  $x = (a + x)\bar{b}$  після відповідних замін вираз матиме вигляд  $Q_1 = (I_1 + Q_1)i_2$

Відповідна програма на мові LD представлена на рис 11.5.

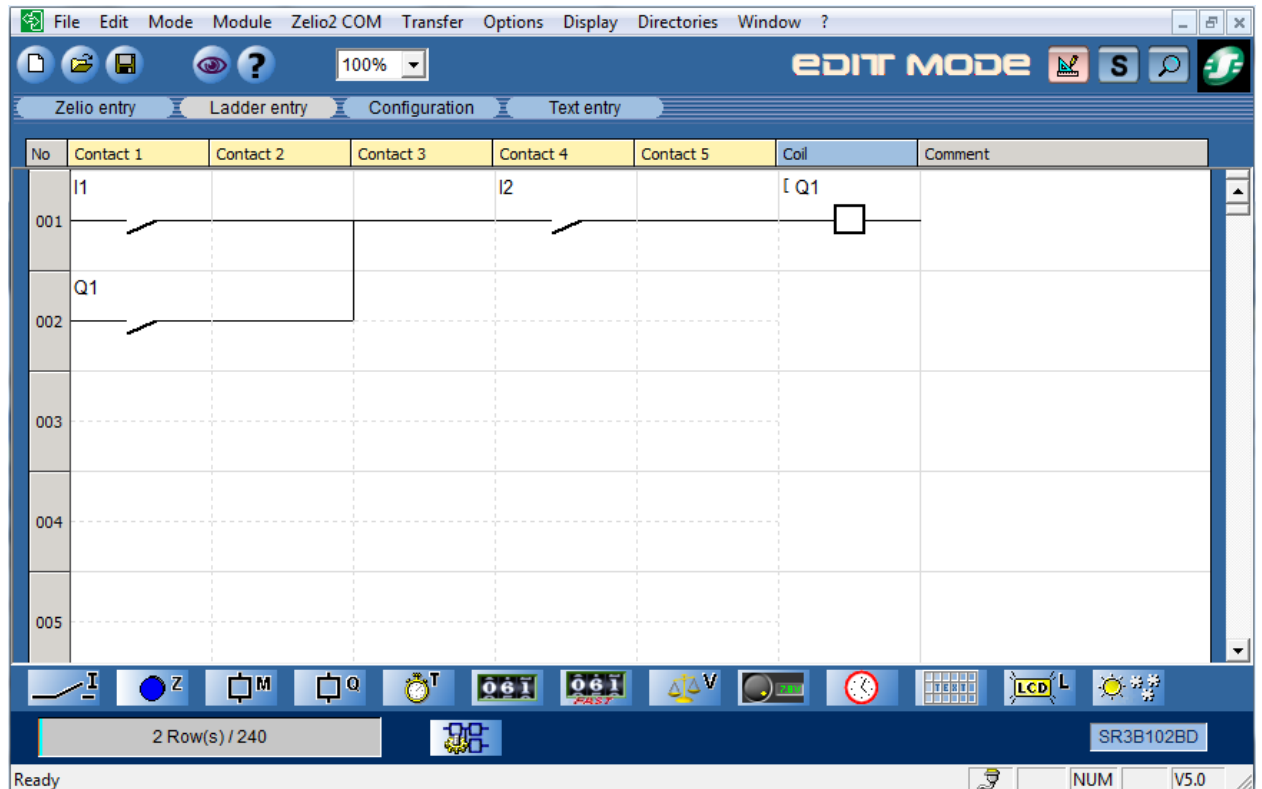



Рисунок 11.5 – Приклад програми на мові LD

Для створення контакту необхідно натиснути кнопку  на панелі інструментів. При цьому обирається до якого входу буде прив'язаний контакт, як показано на рис 11.6.

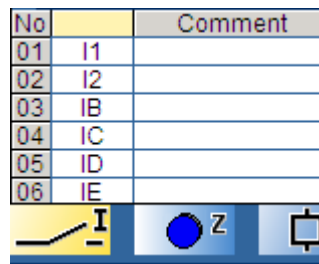


Рисунок 11.6 – Створення контакту.

Для того щоб з'явився на робочій площі необхідно навести на нього курсором миші, натиснути на ліву кнопку і утримуючи її перенести контакт на робочу площу.

Щоб вибрати нормально розімкнений контакт або нормально замкнений контакт необхідно навести курсор на контакт у робочій площі та натиснути праву кнопку миші. У вікні яке з'явилося необхідно обрати Normally open або Normally closed відповідно.

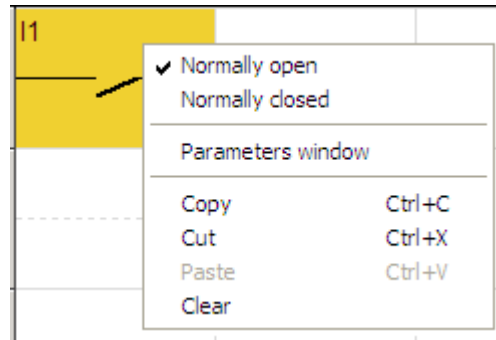




Рисунок 11.7 – Обрання типу

Для створення котушки проміжної змінної та її контакту у панелі інструментів обираємо Auxiliary relays або натискаємо на  та обираємо символ . Для створення контакту цієї котушки необхідно у меню вибору натиснути на змінну M1 та перенести її у робоче поле.


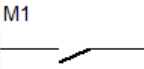


No	Contact 1	Contact 2	Contact 3	Contact 4	Contact 5	Coil	Comment
001							
002							

Рисунок 11.8 – Створення котушки та її контакту

Елемент Discrete outputs відповідає за вихідні котушки. Для їх вибору необхідно навести на значок  та вибрати символ . Щоб вибрати контакт котушки Q необхідно перенести на робоче поле відповідну змінну аналогічно як для проміжної.






















No					Comment	
01	Q1			 S	 R	
02	Q2			 S	 R	
03	Q3			 S	 R	
04	Q4			 S	 R	

Рисунок 11.9 – Меню вибору вихідного контакту та котушки

Для створення котушок Set і Reset необхідно навести на значок  та вибрати символ . Щоб вибрати контакт котушки Reset необхідно навести на значок  та вибрати символ .

Для створення котушки таймера необхідно навести курсор на значок

Timers.  Навівши курсор миші на Timers відкривається вікно, як показано на рис 11.10.

No				Comment	No				Comment
01	T1	T	R		15	TF	T	R	
02	T2	T	R		16	TG	T	R	
03	T3	T	R		17	TH	T	R	
04	T4	T	R		18	TJ	T	R	
05	T5	T	R		19	TK	T	R	
06	T6	T	R		20	TL	T	R	
07	T7	T	R		21	TN	T	R	
08	T8	T	R		22	TP	T	R	
09	T9	T	R		23	TQ	T	R	
10	TA	T	R		24	TR	T	R	
11	TB	T	R		25	TS	T	R	
12	TC	T	R		26	TT	T	R	
13	TD	T	R		27	TU	T	R	
14	TE	T	R		28	TV	T	R	

Рисунок 11.10 – Меню вибору контакту таймера та його котушки

T1-TV відповідають контактам таймера. T відповідає котушці таймера. Для того щоб задати час для таймера, необхідно навести курсор на котушку таймера у робочому полі та лівою кнопкою миші натиснути на неї. Після цього відкриється вікно у якому можна вибрати тип затримки та задати час у спеціальному полі зліва знизу, як показано на рис 11.11.

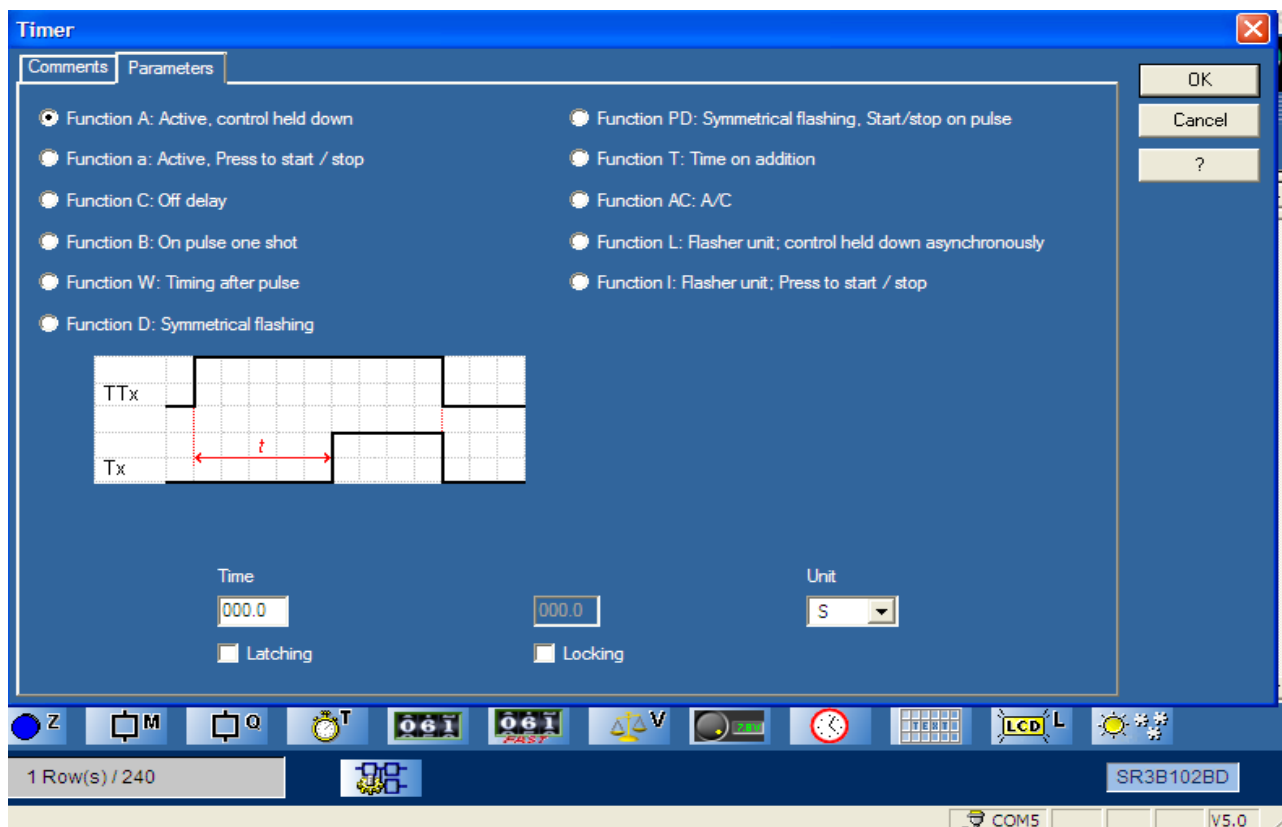


Рисунок 11.11 – Меню налаштування таймера із затримкою на вмикання

**Запис програми в контролер.** Налаштування з'єднання ПК з Zelio для завантаження і перевірки роботи програми в автономному режимі проводиться наступним чином.

1. Підключити інтерфейсний кабель до контролера та комп'ютера. Для інформаційного обміну ПК з Zelio використовується кабель програмування. Їм з'єднуються COM - порт комп'ютера і порт контролера (на лицьовій панелі).
2. Виконати наступні налаштування у програмному забезпеченні Zelio Soft  
З основного меню *Zelio soft* вибирається команда *Transfer* ► *Communication configuration*. У діалоговому меню необхідно вибрати порт який з'єднує ПК та контролер. Після цього необхідно натиснути кнопку *Test*, після цього необхідно закрити вікно клавішею *OK*.
3. Записати програму у контролер.

Програмне з'єднання з Zelio включається з головного меню *Zelio soft* командою *Transfer* ► *Transfer program* ► *PC>module* .

Після завершення запису проекту в оперативну пам'ять Zelio, запуск роботи програми здійснюється вибором команди з дисплею контролера *Menu* ► *Run/Stop* ► *Run*.

## 11.2. Опис лабораторної установки

Даний лабораторний стенд складається з наступного обладнання: Zelio Logic SR3B102BD, ПЧ *Schneider Electric Altivar 312*, асинхронного двигуна *M АОЛ-11/4*, тахогенератора *BR ТМГ-30ПУ3* та інших елементів (автоматичний вимикач *SA*, тумблери *S1..S3*, кнопки *SB1..SB3*, світлодіодні індикатори *HL1..HL10*), зображених на рис. 11.7.

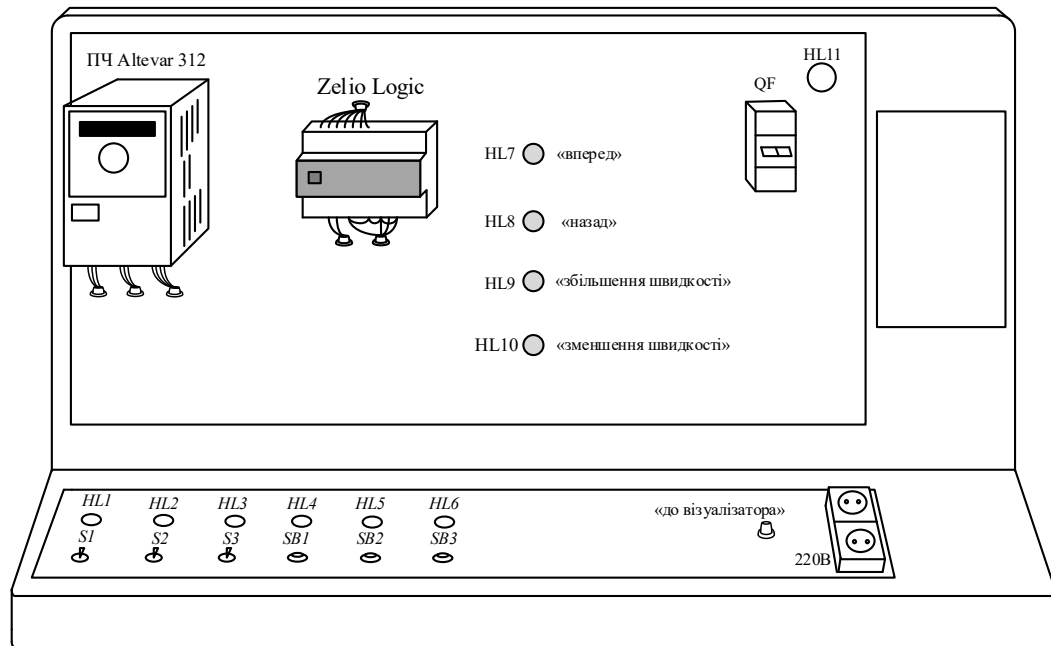


Рисунок 11.12 – Зовнішній вигляд лабораторного стенду

Даний лабораторний стенд, електрична принципова схема якого зображена на рис. 11.12, працює наступним чином. За допомогою системного програмного забезпечення *CoDeSys* складається та відлагоджується робоча програма, яка описує роботу системи відпрацювання заданих траєкторій для керування перетворювачем частоти та надалі асинхронним двигуном. Після відладки, через зовнішній *COM* інтерфейс персонального комп'ютера та порту програмованого логічного контролера *RS-232*, програму завантажують в ПЛК, після чого система стає працездатною за умовами закладених в програмі.

В залежності від завдання, робота перетворювача частоти маніпулюється тумблерами *S1..S3* та кнопками *SB1..SB3*, сигнали яких відображається на світлодіодних індикаторах *HL1..HL6*, та прямують до дискретних входів контролера, а вихідні сигнали контролера, які з'єднані з входами перетворювача частоти, відображаються на світлодіодних індикаторах *HL7..HL10*.

Після подачі сигналів від ПЛК, ПЧ скалярним законом частотного керування, керує швидкістю наростання частоти обертання ротора асинхронного двигуна, статичними та динамічними режимами роботи, реверсом.



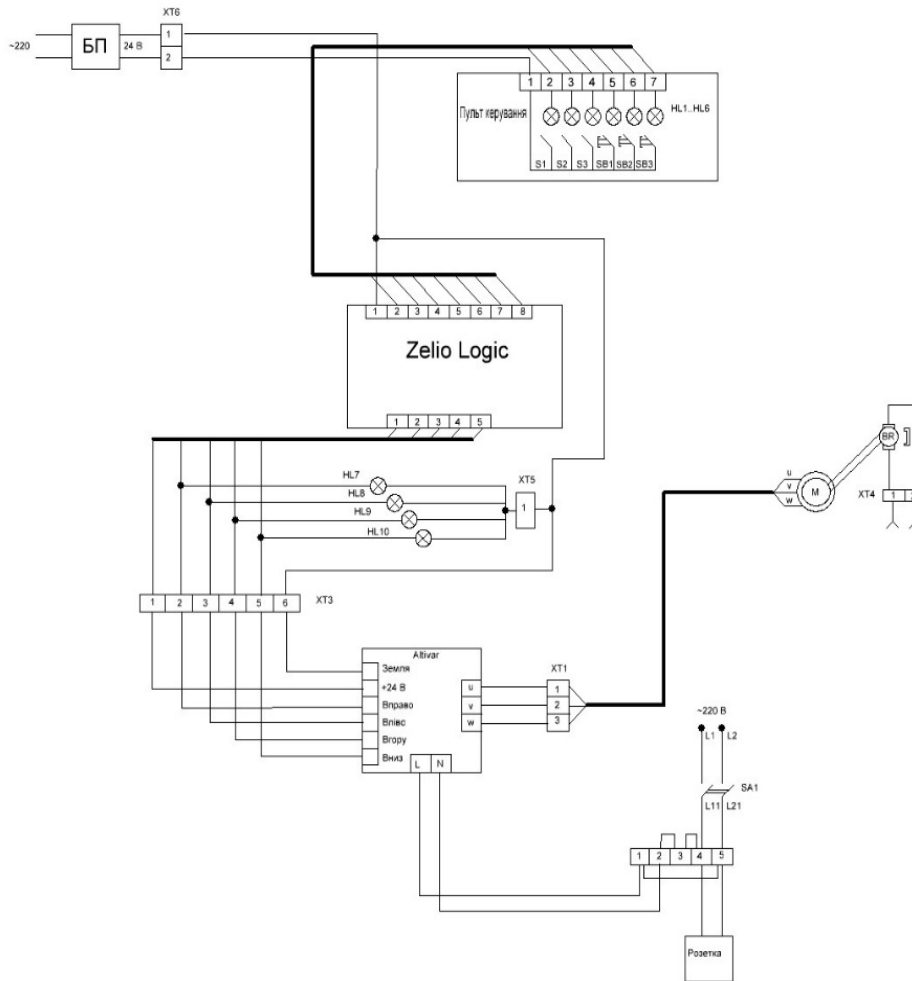


Рисунок 11.14 – Схема підключень лабораторного обладнання

Тумблери  $S1..S3$  та кнопки  $SB1...SB3$  знаходяться на пульту керування та з'єднані між собою загальним проводом. Проводи від них прямують до клемних затискачів, закріплених на  $DIN$ -рейці, через світлодіодні індикатори  $HL1..HL6$ . Після чого провони з'єднані з дискретними входами програмованого логічного контролера 2..7 та загальним проводом, який підключений з 1 входом контролера, з'єднання яких зображено на рис. 11.14. Виноска 1 на схемі підключень позначає джгут із 7 проводів.

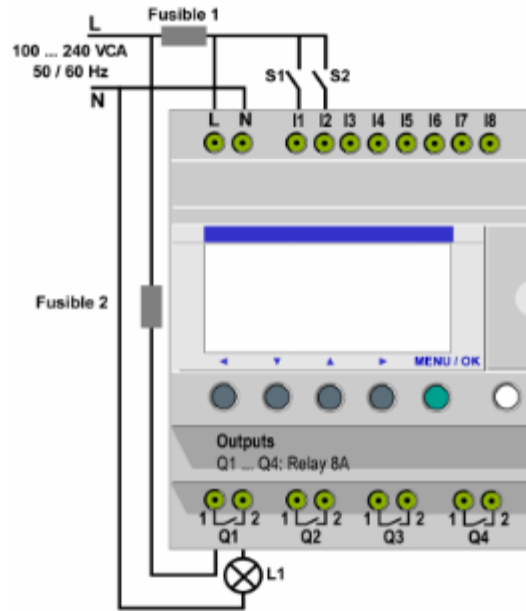


Рисунок 11.15 – Схеми підключення до Zelio Logic SR3B102BD

З виходів перетворювача, частоти  $U$ ,  $V$ ,  $W$  через силові проводи, живлення поступає на монтажні клеми  $XT1$ , а з них на асинхронний двигун  $M$ , як зображено на рис. 11.15. Сам ПЧ заземлений проводом до корпусу лабораторного стенду.

Вал двигуна гнучко з'єднаний з тахогенератором  $BR$ . Клеми аналогових виходів генератора підключаються до входів СТЕП-2 через монтажні клеми  $XT4$ .

У ПЧ використовується режим «цифрового потенціометра». Входи 1 та 2 працюють в режимі «Вперед» та «Назад» відповідно, а 3 та 4 збільшення і зменшення швидкості.

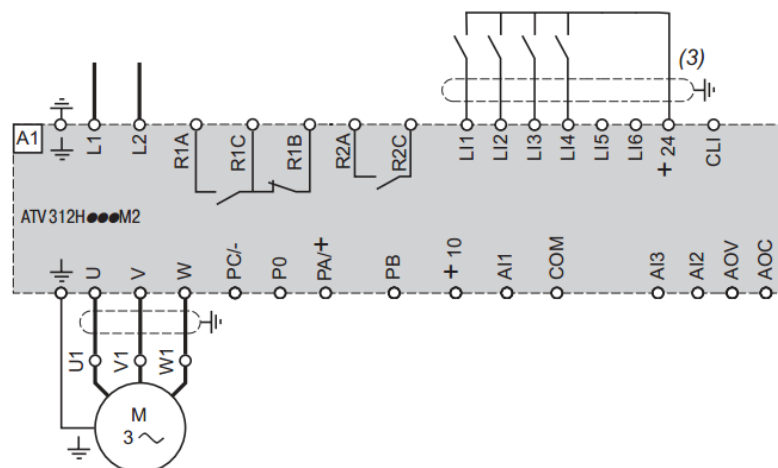


Рисунок 11.16 – Схеми підключення до перетворювача частоти Altivar 312

### 11.3 Програма роботи

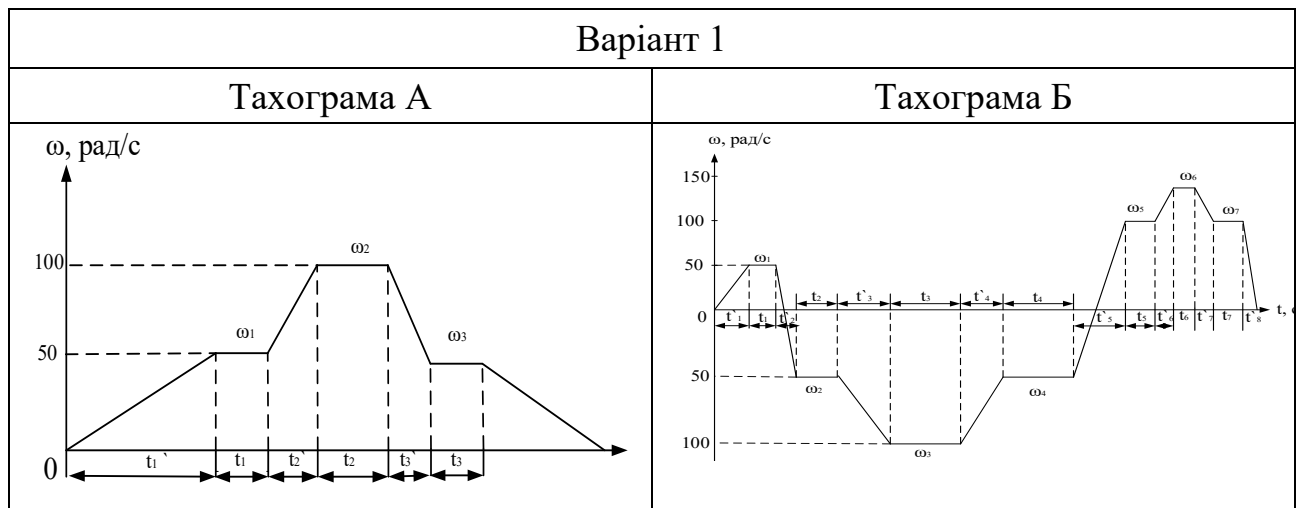
Задана тахограма роботи асинхронного двигуна у деякому технологічному процесі. За варіантом, наведеним у таблиці (див. табл. 11.1), виконати наступні завдання:

1. Розрахувати тахограму роботи двигуна, що подана у завданні відповідно до номера бригади (**виконується вдома під час СРС**).
2. Виконати логічний синтез схеми керування двигуном (**виконується вдома під час СРС**).
3. За отриманими алгебраїчними виразами у пункті 2 скласти релейно-контактну схему для логічного контролера в середовищі *CoDeSys* на мові *LD*.
4. Відлагодити програму та записати її до контролера.
5. Експериментально зняти тахограму руху, відфільтрувати її, побудувати у програмному середовищі *MatLab* та порівняти з розрахованою.
6. Оформити звіт по лабораторній роботі та зробити висновки.

Таблиця 11.1 – Дані для розрахунку тахограми руху\*

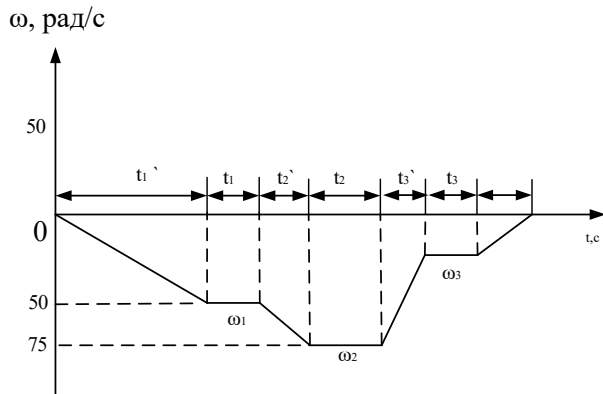
№	Час розгону/сповільнення до частоти 50 Гц, с
1	1.8/2.3
2	2.1/1.5
3	1.7/2.2
4	2.2/1.8
5	1.9/2.4
6	2.3/1.6

Згідно заданого викладачем варіанту, вибрати тахограму роботи двигуна:

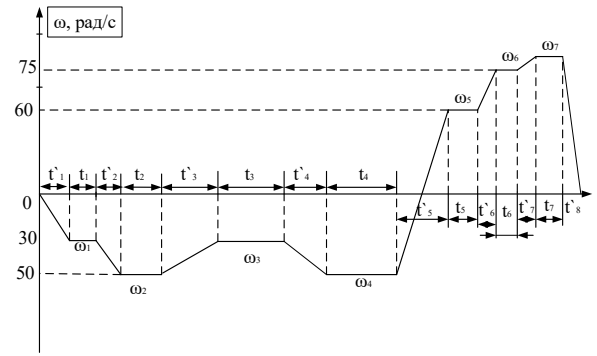


Вариант 2

Тахограма А

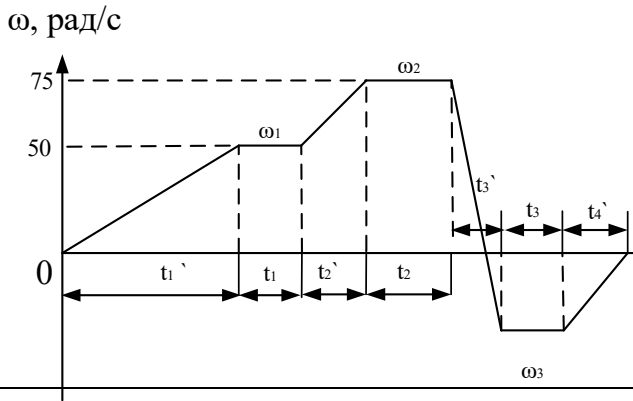


Тахограма Б

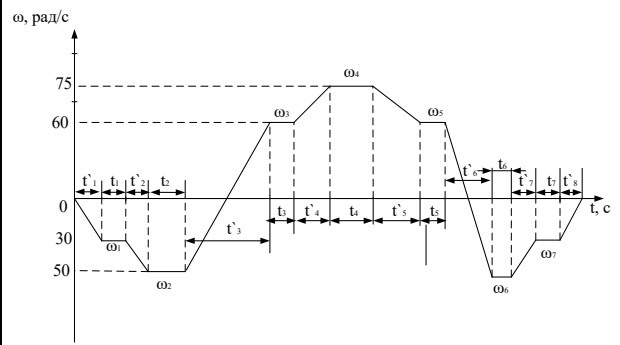


Вариант 3

Тахограма А

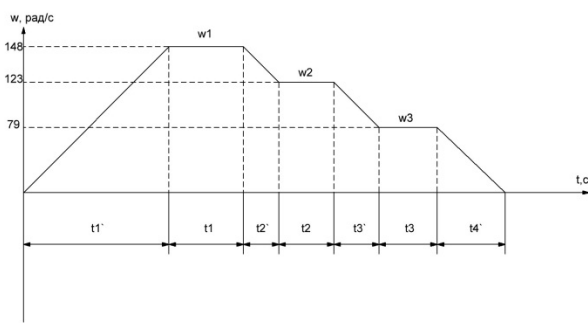


Тахограма Б

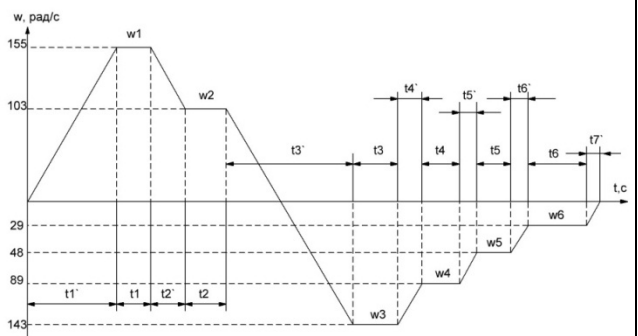


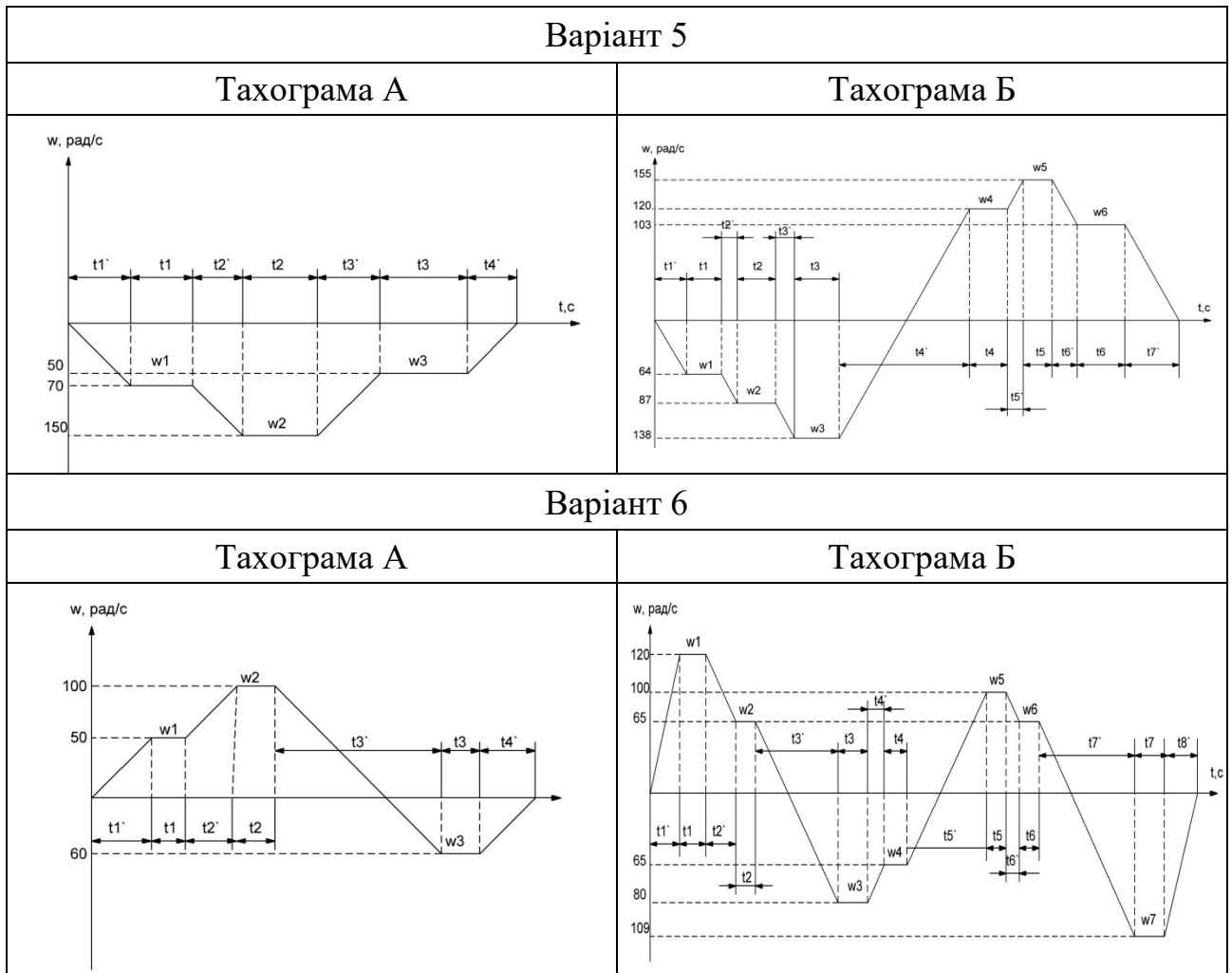
Вариант 4

Тахограма А



Тахограма Б





**Час усталеної роботи  $t_1$ - $t_7$  бригади вибирають довільно, виходячи з того, що СТЕП-1, що використовується для візуалізації результатів, має обмеження за часом перехідного процесу – 24 секунди.**

#### 11.4. Методичні вказівки до виконання роботи

Тахограмою (діаграмою швидкості) називається залежність швидкості робочого органу від часу (залежність кутової швидкості робочого органу від часу). Розглянемо приклад розрахунку тахограми роботи двигуна. Нехай задана тахограма має вигляд, зображений на рисунку 11.17.

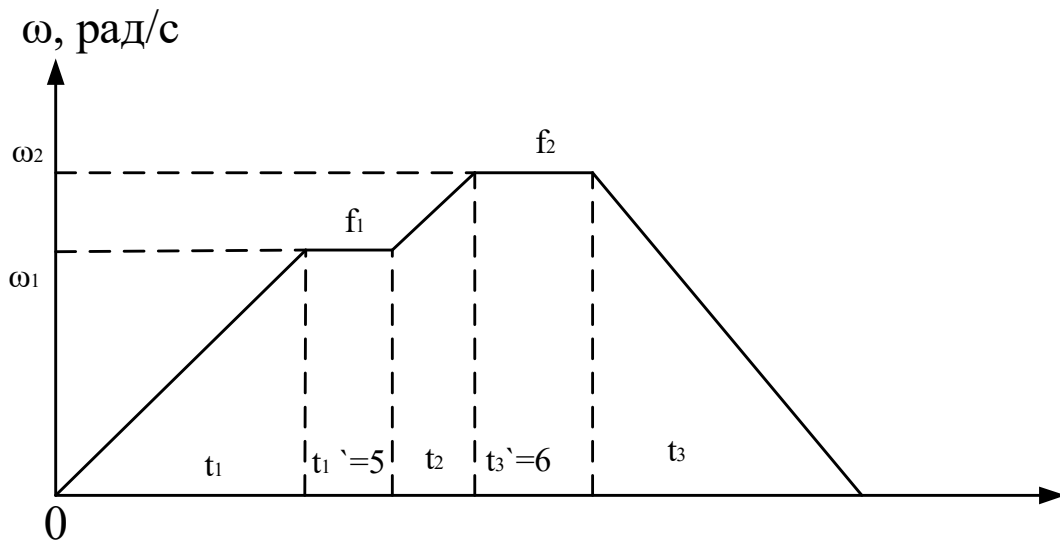


Рисунок 11.17 – Тахограма швидкості асинхронного двигуна деякого технологічного процесу

**Двигун, що використовується у технологічному процесі має дві пари полюсів, тому його синхронна швидкість обертання становить 157 рад/с.**

Вихідні дані для розрахунку:

$$\omega_0 = 50 \text{ рад/с}; \omega_1 = 123 \text{ рад/с}; t_p = 3\text{с}; t_c = 4\text{с}.$$

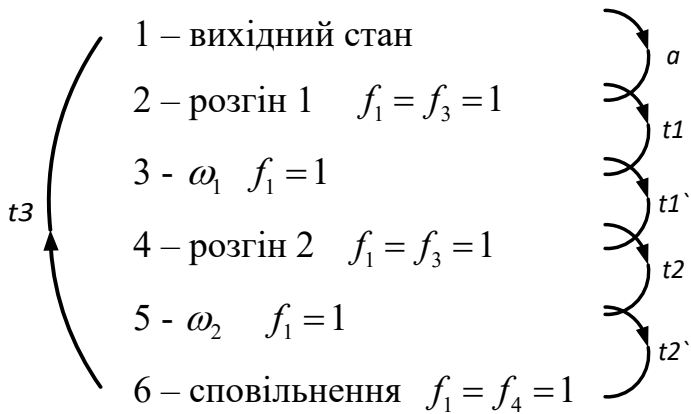
Час роботи на окремій ділянці визначається як

$$t = \frac{t_p * (\omega_1 - \omega_0)}{\omega_c}$$

де  $t_p$  – час розгону/сповільнення до синхронної частоти згідно таблиці 8.1;  $\omega_0$  – початкова швидкість ділянки;  $\omega_1$  – кінцева швидкість ділянки;  $\omega_c$  – синхронна частота обертання.

- 1) Час роботи двигуна на першій ділянці руху  $t_1 = \frac{3 * (50 - 0)}{157} = 0.955\text{с}$
- 2) Час роботи двигуна на другій ділянці руху  $t_2 = \frac{3 * (123 - 50)}{157} = 1.395\text{с}$
- 3) Час роботи двигуна на третій ділянці  $t_3 = \frac{4 * (123 - 0)}{157} = 3.134\text{с}$

Визначаємо кількість станів:



$f_1 = \text{вперед}$

$f_2 = \text{назад}$

$f_3 = \text{розгін}$

$f_4 = \text{сповільнення}$

$t_{ACC} = 3c$

$t_{DEC} = 4c$ .

**Логічний синтез схеми керування двигуном.** Розроблення схем промислової автоматики починається з визначення алгебраїчних виразів (логічних формул), що описують роботу схеми. Процес визначення логічних формул в інженерній практиці називають логічним синтезом. Це найвідповідальніший етап розробки схем автоматики.

В результаті логічного синтезу за формулюванням умов роботи схеми мають бути визначені логічні формули, що описують її роботу, яка задовольняє усі задані умови автоматизації, особливості технологічного процесу та керованого об'єкту.

Переважає більшість промислових схем автоматики – це багатотактні схеми, стан виходів яких залежить не тільки від комбінації вхідних сигналів у будь – який момент часу, але й від послідовності їх надходження або від внутрішнього попереднього стану схем. З цією метою у даній розрахунковій роботі для логічного синтезу вибраний метод синтезу схем на основі тригерів.

Тригер – це логічна схема зі зворотними зв'язками, яка має релейну характеристику керування і може перебувати в одному з двох усталених станів, забезпечуваних цими зв'язками. Тригер характеризується властивістю зберігати інформацію, тому використовуючи вихідні сигнали тригерів як проміжні змінні, можна будувати багатотактні схеми.

*Послідовність виконання синтезу.*

1) За даними умовами роботи схеми будуємо граф переходів. Згідно даних правил, виконуємо логічний синтез методом синтезу схем на тригерах. Для відпрацювання двигуном заданої тахограми роботи на рисунку 8.14 побудований граф переходів, відповідно до заданих умов роботи.

Приймемо такі позначення вхідних і вихідних сигналів, а також сигналів таймерів, які необхідно розглядати як вхідні сигнали для графа переходів.

Вхідні сигнали:  $a$  – команда "Пуск";  $T1 - T3$  – сигнали таймерів, що дають затримки  $t_1 - t_3$ .

Вихідні сигнали:  $f1$  – команда "дозволу роботи",  $f3$  – команда на розгін,  $f4$  – команда на сповільнення.

Графопереходів заданого технологічного процесу, зображений за рис. 11.18

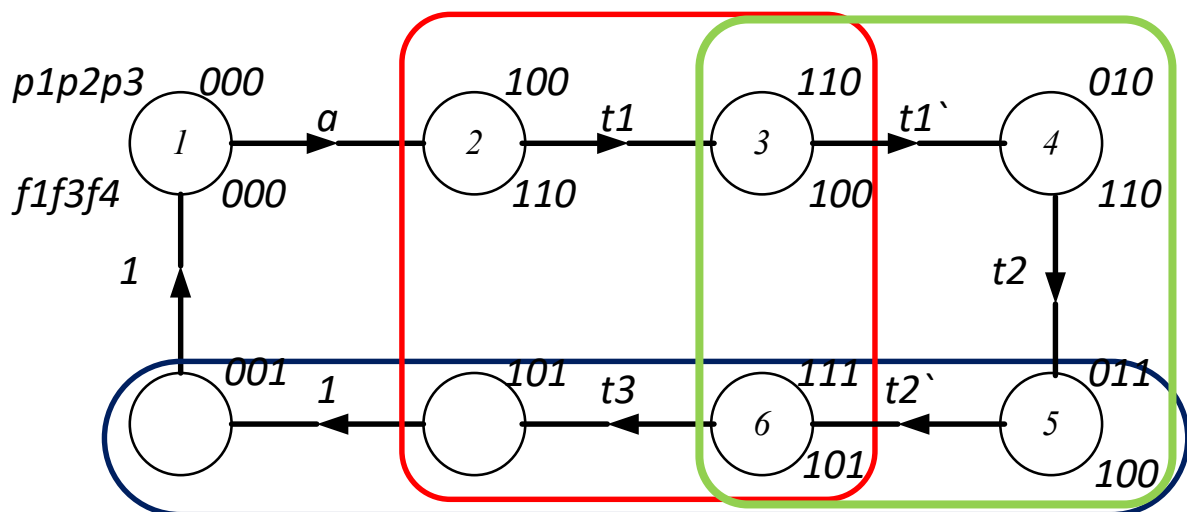


Рисунок 11.18 – Графоперехід заданого технологічного процесу

2) Записуємо умови вмикання і скидання кожного тригера

Синтез схеми полягає у записі умов вмикання і скидання кожного тригера. Ця процедура також описана в третьому розділі дипломного проекту «Методи синтезу логічних схем автоматики».

Запишемо умови вмикання і скидання для першого тригера:

$$S_{P_1} = a \bar{p}_2 \bar{p}_3 + t_2 p_2 p_3; \quad R_{P_1} = t_1 p_2 \bar{p}_3 + \bar{p}_2 p_3;$$

Умови вмикання і скидання для другого тригера:

$$S_{P_2} = t_1 p_1 \bar{p}_3; \quad R_{P_2} = t_3 p_1 p_3;$$

Умови вмикання і скидання для третього тригера:

$$S_{P_3} = t_2 \bar{p}_1 p_2; \quad R_{P_3} = \bar{p}_1 \bar{p}_2;$$

3) Записуємо вирази для вихідних сигналів схеми. Формули для вихідних сигналів  $f_1$ ,  $f_2$  і  $f_4$  записуються як комбінаційні функції вихідних сигналів тригерів  $p_1$ ,  $p_2$  та  $p_3$

$$f_1 = p_1 \bar{p}_2 \bar{p}_3 + p_1 p_2 \bar{p}_3 + \bar{p}_1 p_2 \bar{p}_3 + \bar{p}_1 p_2 p_3 + p_1 p_2 p_3 = p_2 + \bar{p}_1 \bar{p}_3;$$

$$f_3 = p_1 \bar{p}_2 \bar{p}_3 + \bar{p}_1 p_2 \bar{p}_3;$$

$$f_4 = p_1 p_2 p_3;$$

4) В схемі присутні технологічні затримки, тому складаємо вирази вхідних сигналів таймерів як комбінаційні функції вихідних сигналів тригерів.

Запишемо рівняння для таймерів:

$$T_1 = p_1 \bar{p}_2 \bar{p}_3;$$

$$T_1' = p_1 p_2 \bar{p}_3;$$

$$T_2 = \bar{p}_1 p_2 \bar{p}_3;$$

$$T_2' = \bar{p}_1 p_2 p_3;$$

$$T_3 = p_1 p_2 p_3;$$

5) За отриманими логічними формулами складають схему або програму для програмованого логічного контролера.

Програму представлено у вигляді релейно-контактної схеми у середовищі ZelioSoft2

Приймемо такі позначення:

$p_1, p_2, p_3 - M1, M2, M3; a - I1; \bar{p}_1, \bar{p}_2, \bar{p}_3 - m1, m2, m3;$   
 $T_1, T_2, T_3, -T1, T3, T5; T_1, T_2 - T2, T4$

Програма буде мати наступний вигляд:

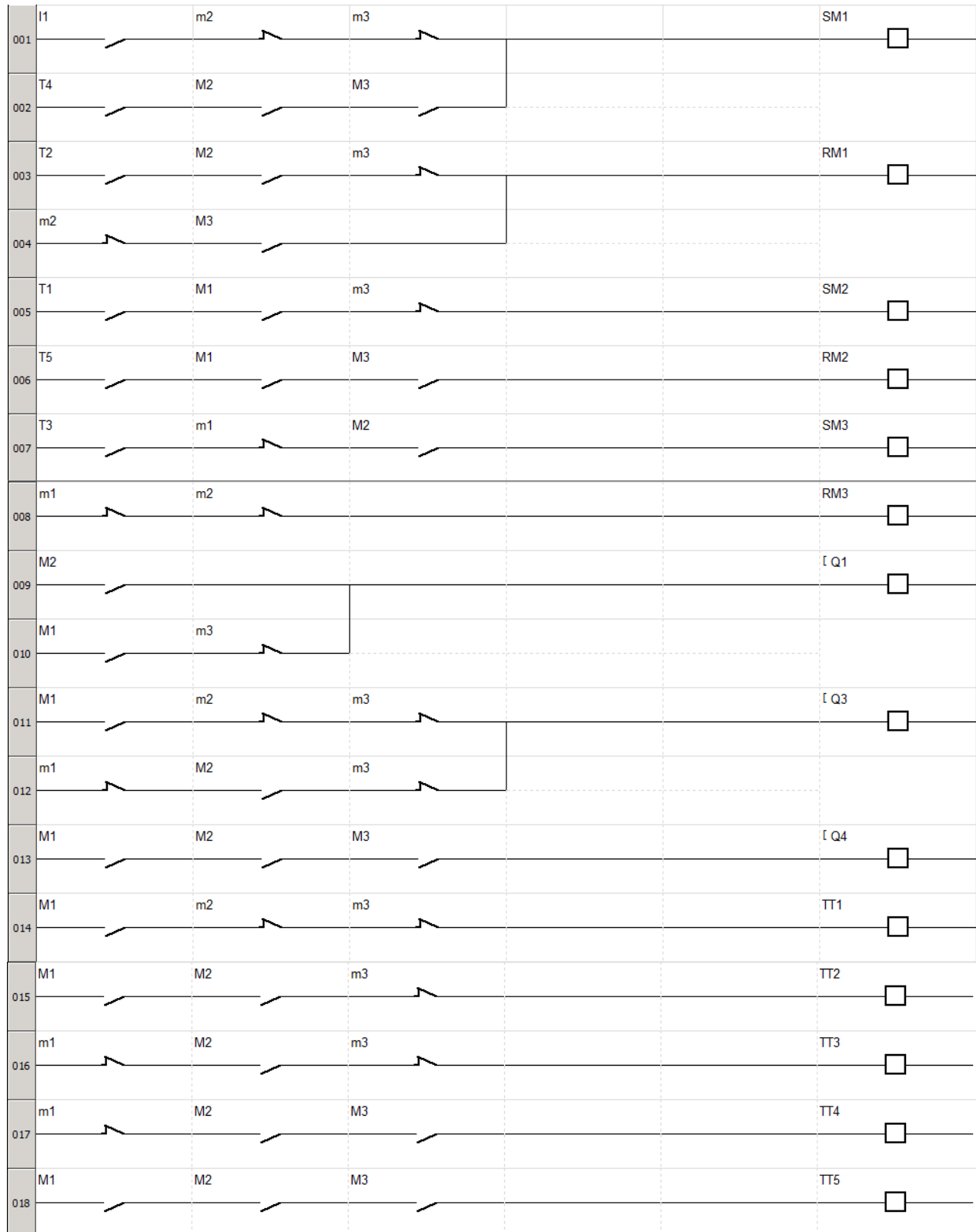


Рисунок 11.19 – Релейно-контактна схема у середовищі ZelioSoft2

Після відлагодження та програмування контролера отримано реальну тахограму руху, яка зображена на рис.11.20

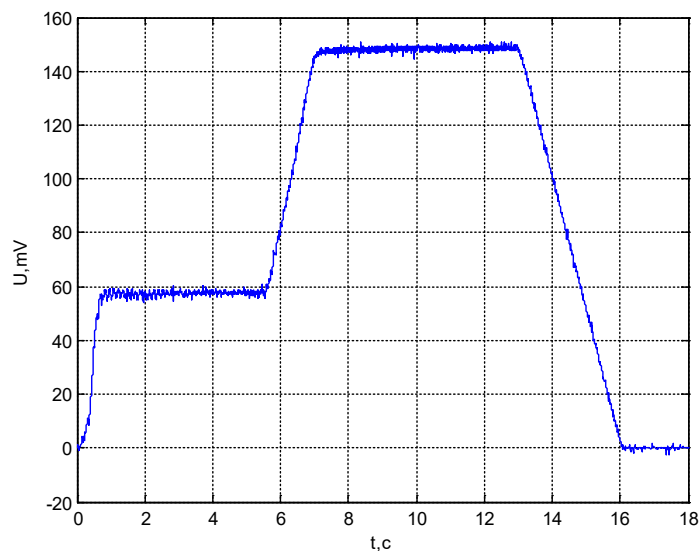


Рисунок 11.20 – Отримана реальна тахограма руху двигуна

Із отриманої тахограми руху видно, що система відпрацьовує задану траєкторію згідно завдання, відповідно синтез схеми автоматики на основі тригерів дозволяє отримати потрібні результати.

#### Приклад 2.

Розглянемо приклад розрахунку тахограми роботи двигуна. Нехай задана тахограма має вигляд, зображений на рисунку 11.21

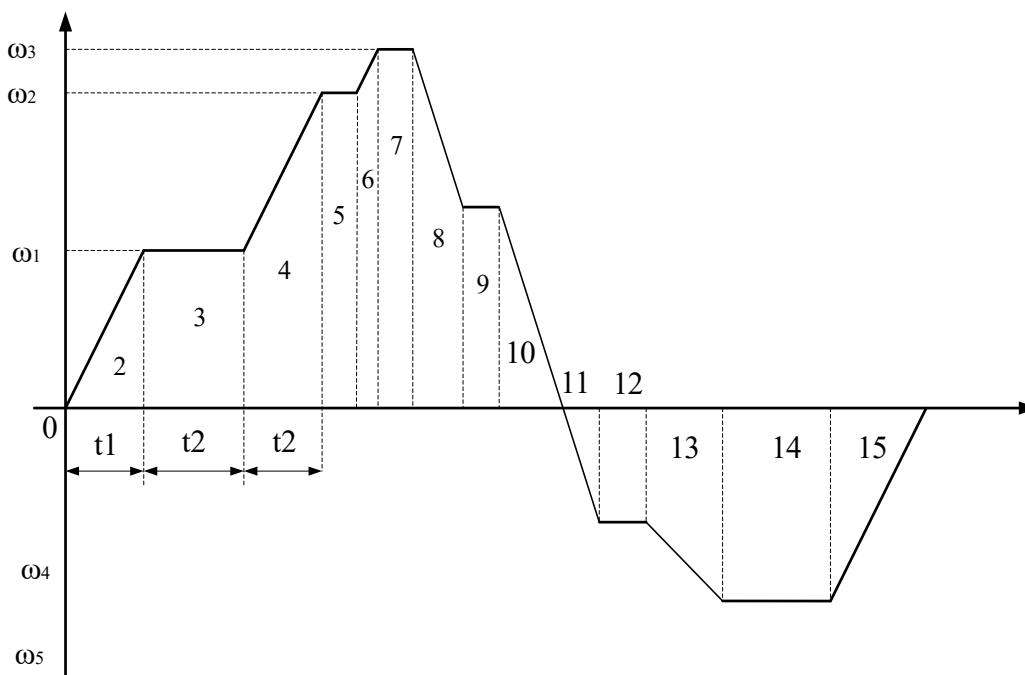


Рисунок 11.21 – Задана траєкторія руху з реверсом двигуна

$$t_{ACC} = 3c; \quad t_{BACC} = 2c$$

Визначимо час роботи двигуна на всіх ділянках руху :

$$t_1 = \frac{3 * 30}{157} = 0.573c$$

$$t_2 = 1c$$

$$t_3 = \frac{3 * (105 - 30)}{157} = 1.433c$$

$$t_4 = 1c$$

$$t_5 = \frac{3 * (157 - 105)}{157} = 0.993c$$

$$t_6 = 1c$$

$$t_7 = \frac{2 * (157 - 30)}{157} = 1.62c$$

$$t_8 = 1c$$

$$t_9 = \frac{2 * 30}{157} = 0.382c$$

$$T_A = \frac{3 * (-75)}{157} = -1.43c$$

$$T_B = 1c$$

$$T_C = \frac{3 * (-75 + 140)}{157} = 1.24c$$

$$T_D = 1c$$

$$T_E = \frac{3 * (-140)}{157} = -1.78c$$

- 1) Кутова швидкість обертання на першій ділянці  $\omega_1 = 30 \text{ рад} / \text{с}$ ;
- 2) Кутова швидкість обертання на другій ділянці  $\omega_2 = 105 \text{ рад} / \text{с}$ ;
- 3) Кутова швидкість обертання на третій ділянці  $\omega_3 = 157 \text{ рад} / \text{с}$ ;
- 4) Кутова швидкість обертання на четвертій ділянці  $\omega_4 = -70 \text{ рад} / \text{с}$ ;

5) Кутова швидкість обертання на п'ятій ділянці  $\omega_5 = -140 \text{ рад / с}$ ;

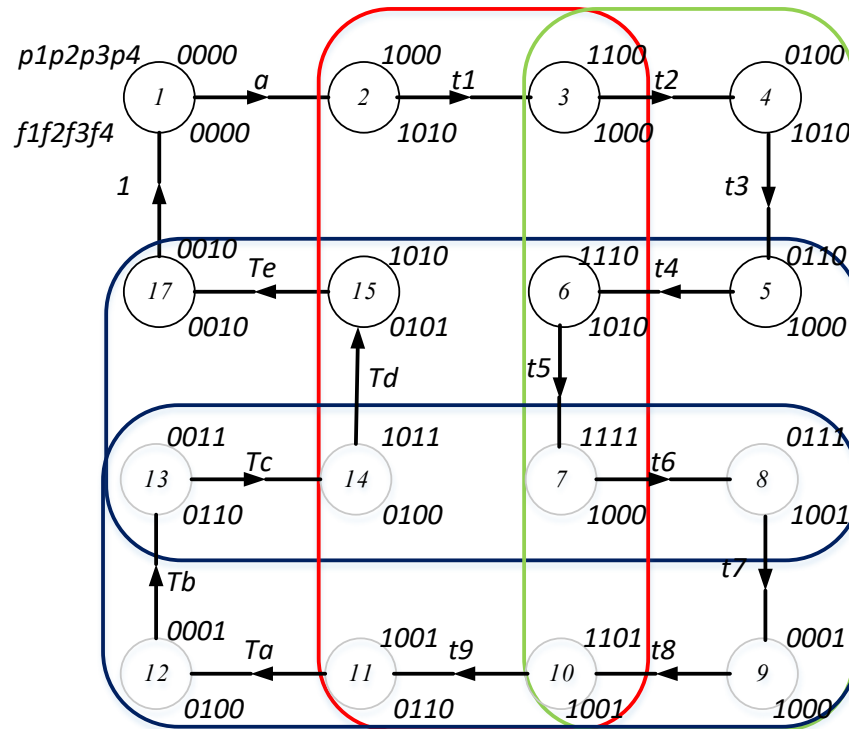


Рисунок 11.22 – Граф переход заданого технологічного процесу

2) Записуємо умови вмикання і скидання кожного тригера

Синтез схеми полягає у записі умов вмикання і скидання кожного тригера. Ця процедура також описана в третьому розділі дипломного проекту «Методи синтезу логічних схем автоматики».

Запишемо умови вмикання і скидання для першого тригера:

$$S_{P_1} = a \bar{p}_2 \bar{p}_3 \bar{p}_4 + t_4 p_2 p_3 \bar{p}_4 + t_8 p_2 \bar{p}_3 p_4 + t_c \bar{p}_2 p_3 p_4;$$

$$R_{P_1} = t_2 p_2 \bar{p}_3 \bar{p}_4 + t_6 p_2 p_3 p_4 + T_A \bar{p}_2 \bar{p}_3 p_4 + T_E \bar{p}_2 p_3 \bar{p}_4;$$

Умови вмикання і скидання для другого тригера:

$$S_{P_2} = t_1 p_1 \bar{p}_3 \bar{p}_4, \quad R_{P_2} = t_9 p_1 \bar{p}_3 \bar{p}_4;$$

Умови вмикання і скидання для третього тригера:

$$S_{P_3} = t_3 \bar{p}_1 p_2 \bar{p}_4 + t_B \bar{p}_1 \bar{p}_2 p_4, \quad R_{P_3} = t_7 \bar{p}_1 p_2 p_4 + \bar{p}_1 \bar{p}_2 \bar{p}_4;$$

Умови вмикання і скидання для четвертого тригера:

$$S_{P_4} = t_5 p_1 p_2 p_3, \quad R_{P_4} = t_D p_1 \bar{p}_2 p_3;$$

3) Записуємо вирази для вихідних сигналів схеми. Формули для вихідних сигналів  $f_1$ ,  $f_2$ ,  $f_3$  і  $f_4$  записуються як комбінаційні функції вихідних сигналів тригерів  $p_1$ ,  $p_2$ ,  $p_3$  та  $p_4$  :

$$f_1 = p_2 + p_1 \bar{p}_3 \bar{p}_4;$$

$$f_2 = \bar{p}_2 p_4 + p_1 p_3 \bar{p}_2;$$

$$f_3 = \bar{p}_1 p_2 \bar{p}_3 \bar{p}_4 + p_1 \bar{p}_2 \bar{p}_3 \bar{p}_4 + p_1 p_2 p_3 \bar{p}_4 + p_1 \bar{p}_2 \bar{p}_3 p_4 + \bar{p}_1 \bar{p}_2 p_3;$$

$$f_4 = \bar{p}_1 p_2 p_3 p_4 + p_1 \bar{p}_2 p_3 \bar{p}_4 + p_1 p_2 \bar{p}_3 p_4;$$

4) В схемі присутні технологічні затримки, тому складаємо вирази вихідних сигналів таймерів як комбінаційні функції вихідних сигналів тригерів.

Запишемо рівняння для таймерів:

$$T_1 = p_1 \bar{p}_2 \bar{p}_3 \bar{p}_4;$$

$$T_2 = p_1 p_2 \bar{p}_3 \bar{p}_4;$$

$$T_3 = \bar{p}_1 p_2 \bar{p}_3 \bar{p}_4;$$

$$T_4 = \bar{p}_1 p_2 p_3 \bar{p}_4;$$

$$T_5 = p_1 p_2 p_3 \bar{p}_4;$$

$$T_6 = p_1 p_2 p_3 p_4;$$

$$T_7 = \bar{p}_1 p_2 p_3 p_4;$$

$$T_8 = \bar{p}_1 p_2 \bar{p}_3 p_4;$$

$$T_9 = p_1 p_2 \bar{p}_3 p_4;$$

$$T_A = p_1 \bar{p}_2 \bar{p}_3 p_4;$$

$$T_B = \bar{p}_1 \bar{p}_2 \bar{p}_3 p_4;$$

$$T_C = \bar{p}_1 \bar{p}_2 p_3 p_4;$$

$$T_D = p_1 \bar{p}_2 p_3 p_4;$$

$$T_E = p_1 \bar{p}_2 p_3 \bar{p}_4;$$

5) За отриманими логічними формулами складають схему або програму для програмованого логічного контролера.

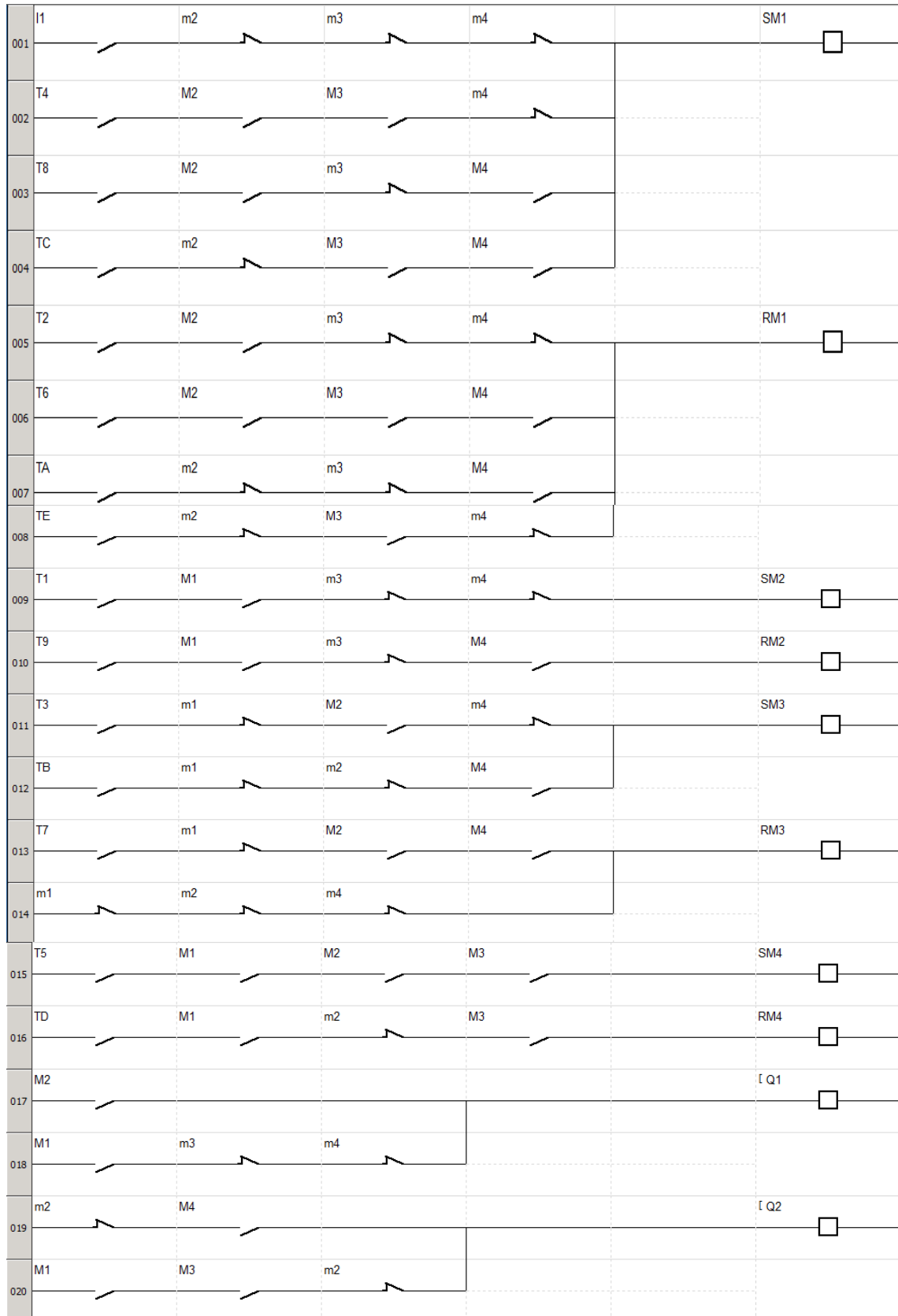
Програму представлено у вигляді релейно-контактної схеми у середовищі ZelioSoft2

Прийємо такі позначення:

$p_1, p_2, p_3, p_4 - M1, M2, M3, M4; a - I1; \bar{p}_1, \bar{p}_2, \bar{p}_3, \bar{p}_4 - m1, m2, m3, m4;$

$T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8, T_9 - T1, T2, T3, T4, T5, T6, T7, T8, T9;$

$T_A, T_B, T_C, T_D, T_E - TA, TB, TC, TD, TE$



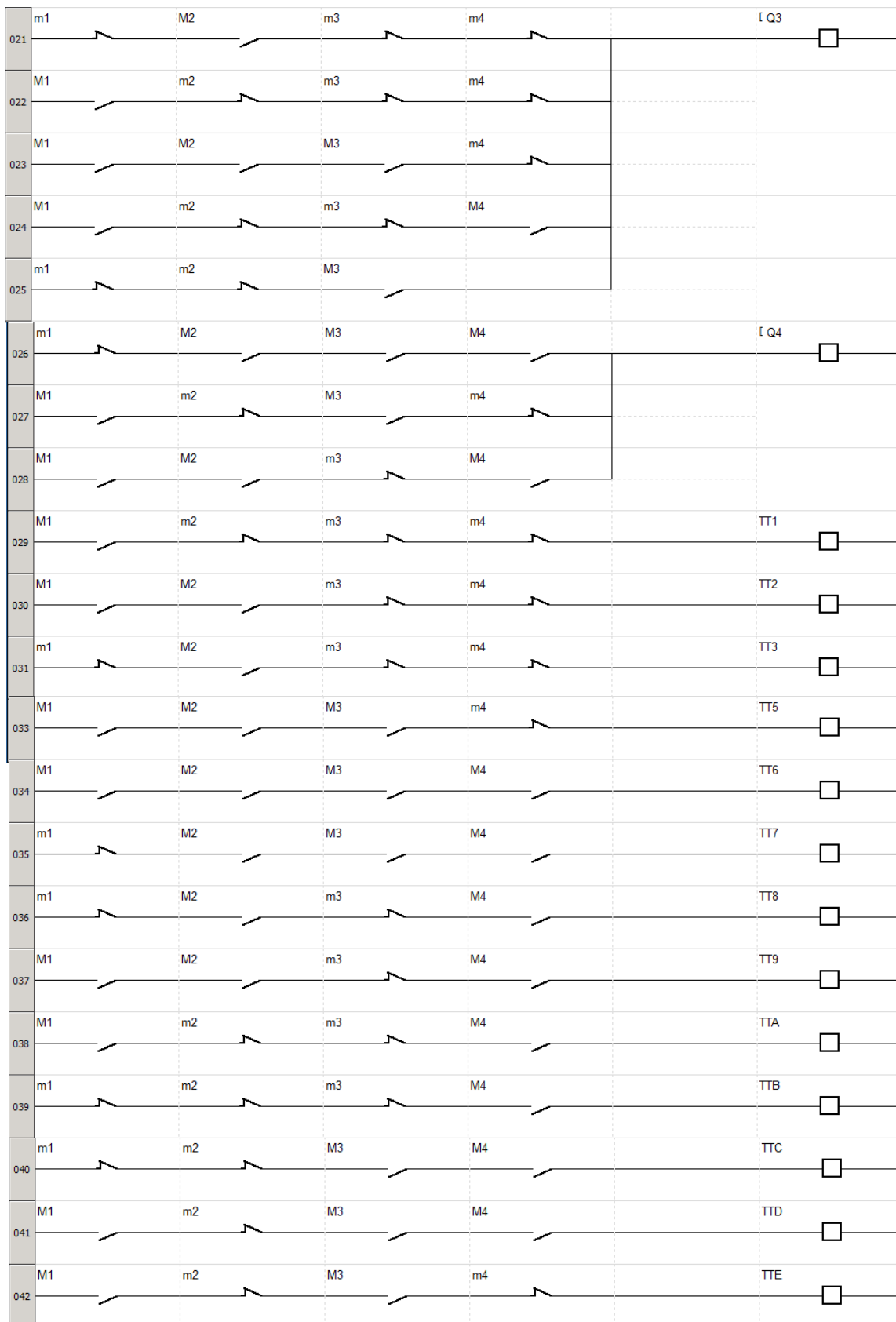


Рисунок 11.23 – релейно-контактна схема у середовищі ZelioSoft2

Після відлагодження та програмування контролера отримано реальну тахограму руху, яка зображена на рис. 11.24

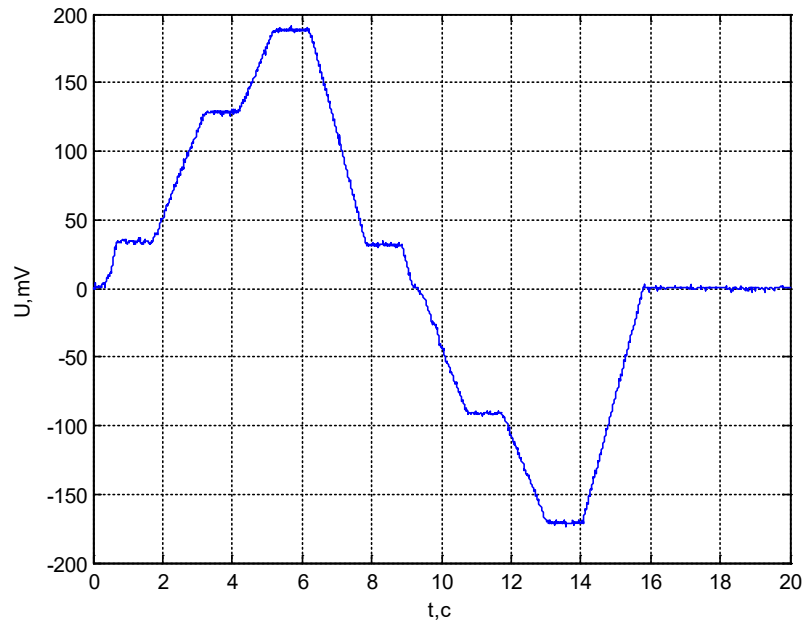


Рисунок 11.24 – Отримана реальна тахограма руху із реверсом двигуна

### 11.5. Зміст звіту

- 1) Титульний аркуш.
- 2) Мета роботи, програма роботи.
- 3) Опис стенда та його схема.
- 4) Умова роботи схем згідно варіанту.
- 5) Логічний синтез схеми керування.
- 6) Адресування входів, виходів, таймерів, комірок пам'яті.
- 7) Програма на мові LD у середовищі ZelioSoft
- 8) Тахограми, що підтверджують відпрацювання заданих траєкторій.
- 9) Висновок про відповідність роботи заданим умовам.

### 11.6 Контрольні запитання

1. Що таке тахограма?
2. Які операнди використовуються у середовищі ZelioSoft?
3. Навести алгоритм завантаження програми у ПЛК.
4. Навіщо використовується пристрій СТЕП-2?
5. Якими мовами може здійснюватися програмування у середовищі ZelioSoft?

## Лабораторна робота №12

### ДОСЛІДЖЕННЯ СИСТЕМИ АВТОМАТИЗАЦІЇ НА ОСНОВІ ПРОМИСЛОВОГО РОБОТА РФ-202М

Тривалість лабораторної роботи – 4 години.

Тривалість домашньої роботи – 4 години.

**Мета роботи** – вивчити будову та принцип дії промислового робота РФ-202м, навчитися працювати у програмному середовищі Lectus OPC та SimpLight, скласти програми для відпрацювання роботом заданих траєкторій.

#### 12.1. Основні теоретичні відомості

Функціональна схема стану із усіма взаємозв'язками показано на рисунку 12.1.

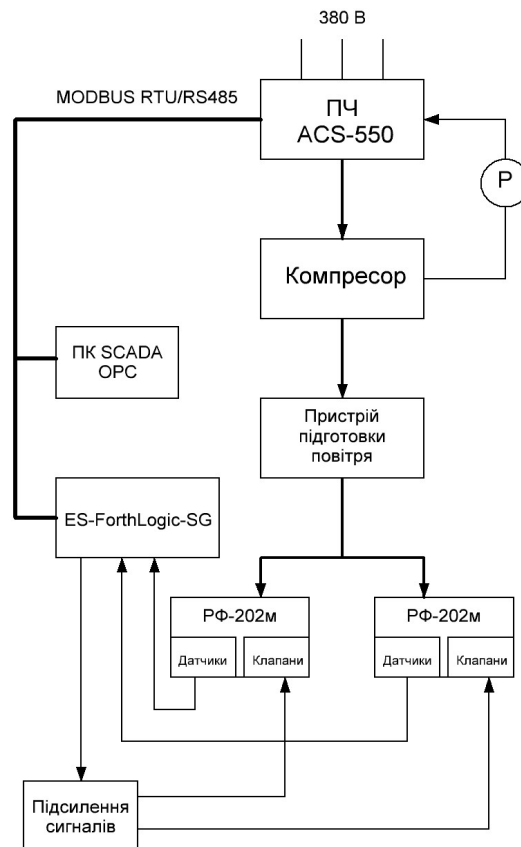


Рисунок 12.1 – Функціональна схема стану

До складу схеми входить ПЧ ACS-550 який керує компресором, що приводиться в рух асинхронним двигуном з фазним ротором. На виході компресора стоїть датчик тиску (P), який реалізує зворотній зв'язок за тиском. Від компресора повітря надходить до пристрою підготовки повітря, де відбувається очистка повітря від вологи та важких сухих домішок (пил і т.п.), а також додавання під тиском у повітря масла для змащення поршнів та циліндрів маніпулятора. Далі повітря потрапляє до розподільних блоків, що направляють його у потрібні пневматичні циліндри.

Робот виконує кожний рух від енергії стисненого повітря що надходить з відкритих пневматичних клапанів. Поворот та підйом колони, висування та втягування рук, затискання МЗ здійснюється від пневматичних циліндрів. Обертання рук навколо осі здійснюється від пневматичних серводвигунів, вал яких здійснює обертальні рухи.

Кожний робот керується блоком із 12 клапанів. Щоб клапан знаходився в закритому стані потрібно подати на обмотку керування напругу величиною 27 В. Тому для забезпечення нерухомості роботів, на кожний клапан має бути прикладена напруга. Сигнали керування клапанами надходять від ПЛК до блоку підсилення сигналу, а потім подаються до кожного клапану. Для визначення положення промислового робота, використовується 6 кінцевих герконових датчиків. Сигнали з датчиків надходять до ПЛК і відображуються в SCADA-системі, де використовуються для реалізації автоматичної програми керування.

ПЛК являється необхідною ланкою, що з'єднує між собою SCADA-систему та промисловий робот, як у режимі локального керування так і автоматичного. Передача даних відбувається за допомогою протоколу MODBUS RTU та інтерфейсу RS485. OPC-сервер, що знаходиться на комп'ютері, отримує дані від ПЛК та надає до них доступ SCADA-системі.

OPC-сервер може працювати в різних режимах, а саме: Master, Slave, Master&Slave, Master&Slave (віддалене підключення). Опитування Modbus контролерів відбувається за допомогою використання протоколів Modbus RTU, Modbus ASCII, Modbus TCP.

1. Режим Master зображений на рисунку 12.2.

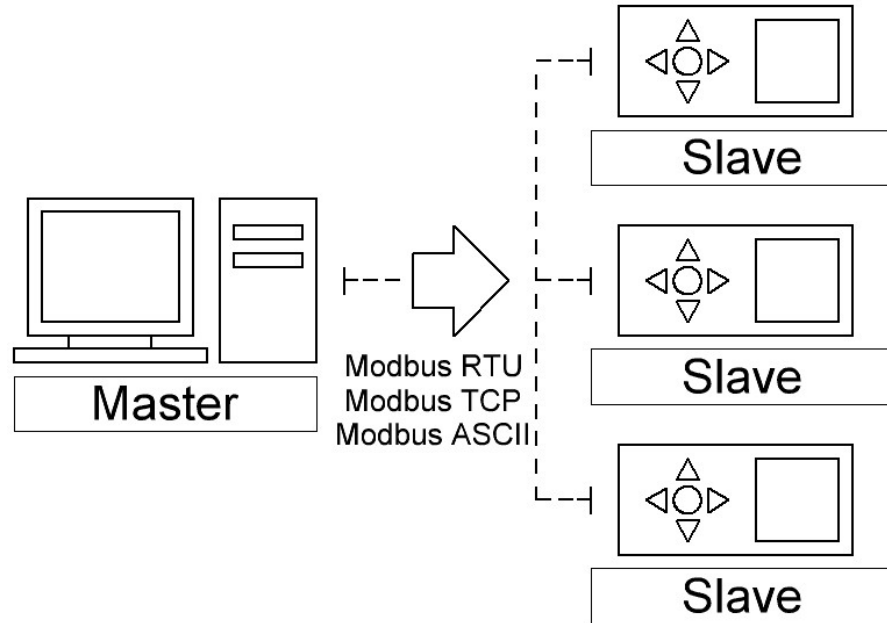


Рисунок 12.2 – Режим Master

В режимі Master сервер самостійно опитує Modbus контролери за протоколами Modbus RTU, Modbus ASCII, Modbus TCP.

2. Режим Slave зображений на рисунку 12.3.

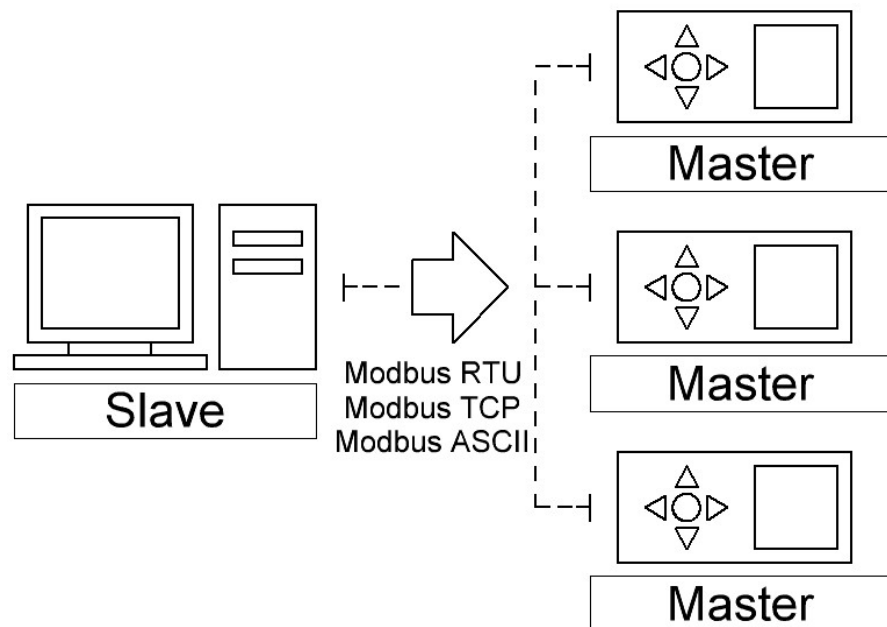


Рисунок 12.3 – Режим Slave

В режимі Slave сервер відповідає на Modbus запити від Master-ів по протоколам Modbus RTU, Modbus ASCII, Modbus TCP.

3. Режим Master&Slave зображений на рисунку 12.4.

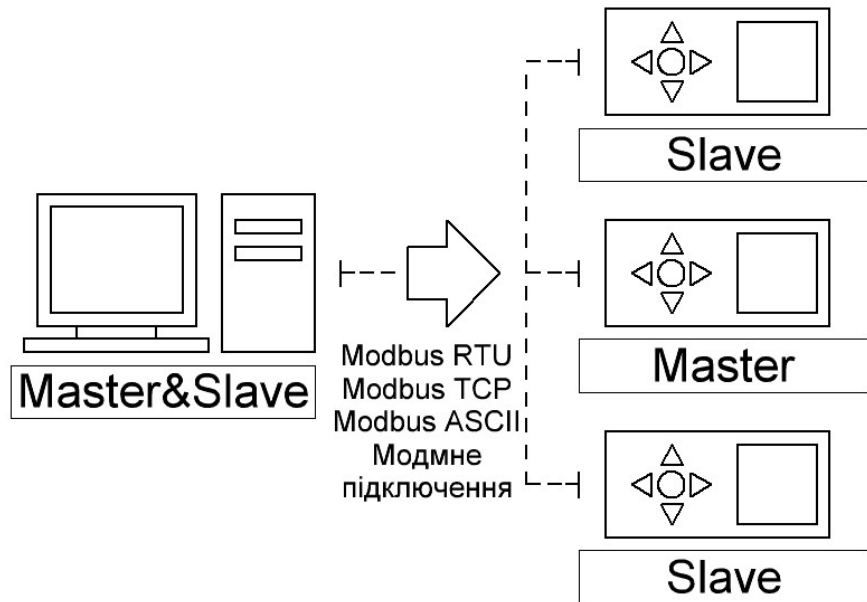


Рисунок 12.4 – Режим Master&Slave

В режимі Master&Slave сервер може як опитувати Modbus контролери по протоколам Modbus RTU, Modbus ASCII, Modbus TCP, так і відповідати на Modbus запити від Master-ів.

4. Режим віддаленого підключення Master&Slave зображено на рисунку 12.5.

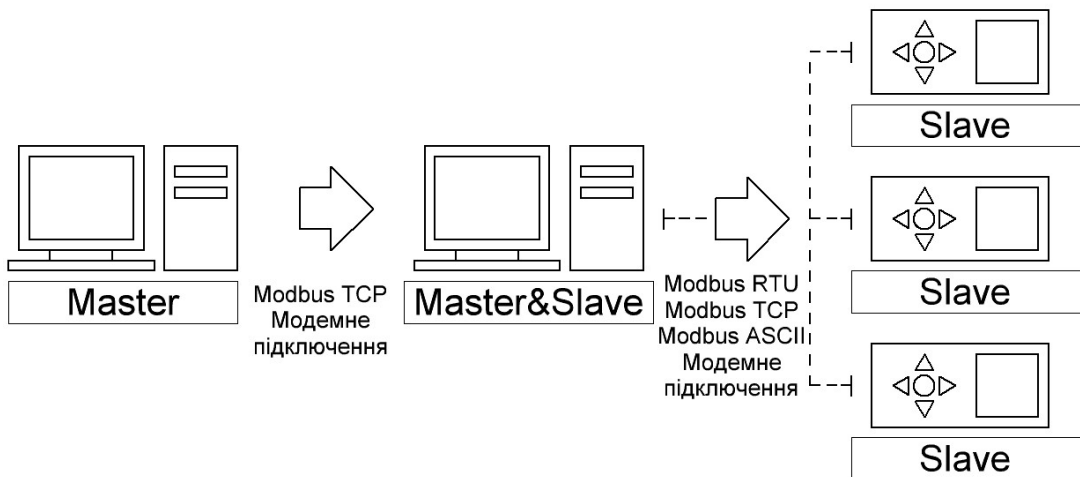


Рисунок 12.5 – Режим віддаленого підключення Master&Slave

В режимі віддаленого підключення один сервер працює в режимі Master&Slave, а інший сервер здійснює віддалене підключення до першого за допомогою Modbus TCP або модемного підключення. Даний режим можна використовувати для отримання даних від віддалених точок доступу.

Встановлена на комп'ютері програма Lectus OPC/DDE сервер автоматично виконує реєстрацію серверу (запис відповідної інформації в системний реєстр).

ПЛК являється ключовим елементом даної системи. В даному випадку використовується ПЛК ES-ForthLogic фірми Електросвіт, програмований логічний контролер модульної архітектури, використовується для побудови замкнутих, інтегрованих контурів, що можуть бути інтегровані до інших систем, керування довільним обладнанням малого та середнього рівня складності.

Схема розташування входів/виходів ПЛК зображено на рисунку 12.6.

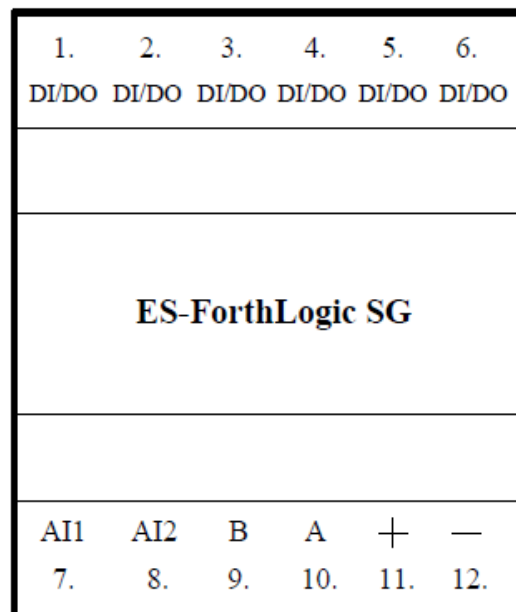


Рисунок 12.6 – Розташування входів/виходів ПЛК ES-ForthLogic

Позначення окремих контактів наведено в таблиці 12.1.

Таблиця 12.1 – Позначення контактів ПЛК

Номер контакту	Назва контакту	Призначення
1-6	D1-D6	Універсальні входи/виходи: - Входи для приєднання сигналів типу «сухий контакт» відносно «-»: - Виходи типу «відкритий колектор» відносно контакту «-»:

7, 8	AI1, AI2	Аналогові входи по струму відносно контакту «-»
9	B	Сигнал B (DATA-) послідовної шини RS485
10	A	Сигнал A (DATA+) послідовної шини RS485
11	+	Плюс постійної напруги живлення ПЛК
12	-	Мінус постійної напруги живлення ПЛК

Протокол MODBUS RTU і інтерфейс RS485 дає можливість обміну даними із зовнішніми пристроями.

Для комутації електричних клапанів, що відповідають за рухи роботів необхідна напруга живлення  $U=27$  В та струм  $I=0,3$  А. Оскільки у роботі використовується два роботи, кожний з яких керується 12 клапанами, то максимальний струм блоку живлення для відкривання/закривання клапанів складатиме 7.2 А. Для живлення ПЛК, його модулів розширення та керування клапанами використовується блок живлення фірми F&F ZI-240-24.

Так як, кожен із маніпуляторів має по 12 клапанів для подачі стисненого повітря, та 6 геконових датчиків, що відповідають конкретному положенню робота, то разом із ПЛК ES-ForthLogic використовуються модулі розширення цифрових входів/виходів ES-DIO-1M.

ES-DIO-1M обладнаний шістьма універсальними контактами. Кожний контакт може розглядатись як вхід для сигналу «сухий контакт» відносно мінуса напруги живлення або як вихід типу «відкритий колектор», який комутує мінус напруги живлення. В ES-DIO-1M реалізована технологія локальної пам'яті, її принцип полягає в тому, що довільний вихід можна запрограмувати на довільний стан, який буде автоматично включений при подачі живлення на ES-DIO-1M навіть при відсутності інформаційного обміну по послідовному інтерфейсу. Цей стан називається локальним станом вихідного контакту, на відміну від поточного стану, що може варіюватися динамічно в процесі інформаційного обміну з контролером. Нові локальні і поточні значення виходів а також станів входів і виходів можна установити або

прочитати по послідовному інтерфейсу RS485 по протоколу MODBUS RTU. Більшість параметрів обміну являються фіксованими за виключенням власного мережевого адресу – його можна встановити за допомогою багатопозиційного перемикача схованого під передньою панеллю. ES-DIO-1M обладнаний двома світлодіодами розташованими на передній панелі, вони відображують наступну інформацію: зелений – наявність напруги живлення, жовтий – вдалий обмін по протоку MODBUS RTU.

Схема розташування виводів ES-DIO-1M показано на рисунку 12.7.

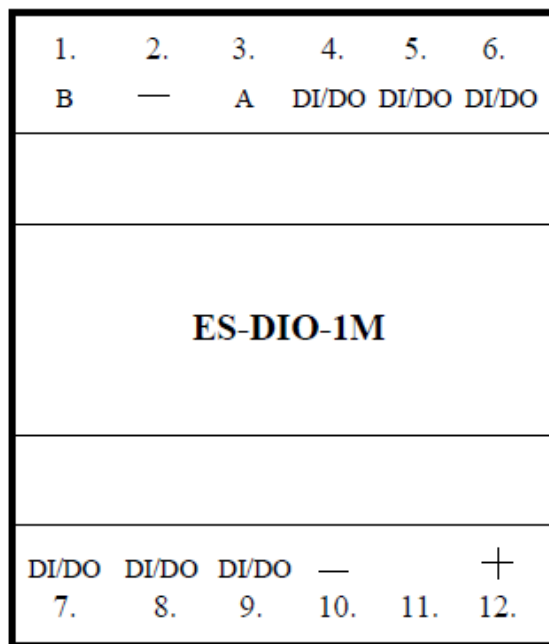


Рисунок 12.7 – Схема виводів модуля розширення ES-DIO-1M

В таблиці 12.2 наведено позначення контактів модуля розширення (MP) ES-DIO-1M.

Таблиця 12.2 – Позначення контактів модуля розширення

Номер контакту	Назва контакту	Призначення
4-9	D1-D6	Універсальні входи/виходи: <ul style="list-style-type: none"> <li>- Входи для приєднання сигналів типу «сухий контакт» відносно контакту «-»</li> <li>- Виходи типу «відкритий колектор» відносно контакту «-&gt;»</li> </ul>
1	B	Сигнал B (DATA-) послідовної шини RS485

3	A	Сигнал A (DATA+) послідовної шини RS485
12	+	Плюс постійної напруги живлення МР
10	-	Мінус постійної напруги живлення МР

### Характеристики робота РФ-202м

Промисловий робот РФ-202м призначений для автоматизації процесів завантаження розвантаження технологічного устаткування (металорізальних верстатів, конвеєрних ліній, ливарних машин, пресів і т.д.).

Робот складається із маніпулятора автоматичного (МА), системи керування та пристрою підготовки повітря (ППП).

Робот призначений для переміщення, знімання та встановлення предметів, що обробляються на технологічному устаткуванні. Під час переміщення предмети затискаються в механізмі затиску (МЗ) автоматичного маніпулятора. Маніпулятор дозволяє здійснювати вертикальне переміщення (підйом чи опускання), горизонтальне (робота модуля горизонтального переміщення МГП), поворот МГП у горизонтальній площині відносно вертикальної осі і обертання МЗ відносно горизонтальної осі. Всі переміщення МА відбуваються за рахунок подачі стисненого повітря від ППП у відповідні циліндри МА. Подачею повітря до модулів МА керує система керування за допомогою електронних клапанів.

Технічні характеристики промислового робота РФ-202м наведені в таблиці 12.3.

Таблиця 12.3 – Технічні характеристики РФ-202м

Найменування	Значення
Напруга живлення	220 В ± 15%
Частота струму	50 Гц ± 1Гц
Номінальний тиск подачі стиснутого повітря	0.4 Мпа
Вантажопідйомність кожної руки	10 Н
Величина обертання	180°

Номінальна величина висування руки	200 мм
Похибка повторюваності позиціонування	$\pm 0.5$ мм
Номінально величина вертикального переміщення	20 мм

Маніпулятор робота представляє собою механічну систему, що виконує маніпуляції з предметами, один із елементів якої нагадує руку людини.

На рисунку 12.8 представлена структурно-кінематична схема робота РФ-202М. Вона дає наглядне представлення про кінематичні можливості робота. Робота РФ-202М має 5 ступенів рухомості, тобто може виконувати 5 різних незалежних рухів.

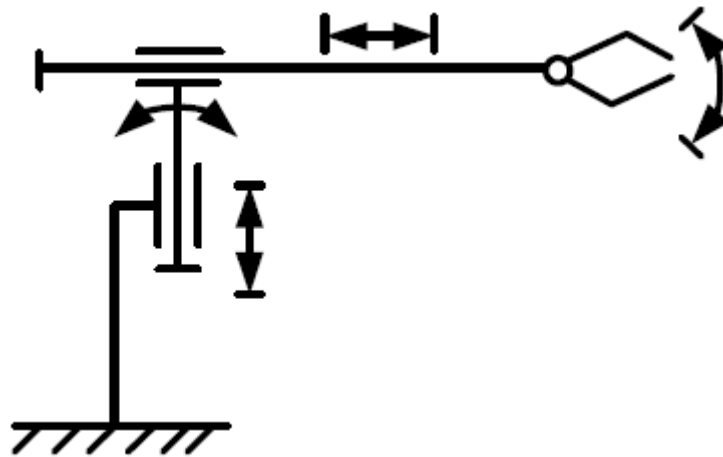


Рисунок 12.8 – Структурно-кінематична схема робота РФ-202М

Руки робота за допомогою кронштейнів закріплені на колоні, що спирається на основу робота. Основа робота закрита кожухом, під яким розміщено два пневмоциліндри, що повертають колону робота, а також електроклапани, для керування подачі стиснутого повітря до всіх циліндрів.

Всі рухи робота виконує за рахунок енергії стисненого повітря від пневматичних виконуючих механізмів. Висування рук, підйом колони, поворот колони, закриття захватів відбувається від пневмоциліндрів з прямолінійним рухом поршня, а обертання рук – від пневмодвигунів, вал яких здійснює неповні обертальні рухи.

На рисунку 12.9 показана пневматична схема приводу робота. Стиснуте повітря через вхідний штуцер 1, запірний вентиль 2, вологовідділювач 3,

регулятор тиску 4, маслорозпилювач 6 по повітряним каналам поступає до розподільвача повітря 7. Всі ці елементи об'єднані в один блок – блок підготовки стиснутого повітря. Розподільвачі направляють стиснене повітря до пневмоциліндрів.

Вологовіділювач видаляє з повітря вологу. За допомогою регулятора тиску відбувається налаштування тиску стисненого повітря, що поступає до пневмоциліндрів. Маслорозпилювач насичує повітря маслом, необхідним для змащення елементів, що труться, пневмоциліндрів. Контроль тиску стисненого повітря, що надходить до пневмоциліндрів, відбувається візуально по манометру 5.

Блок підготовки стисненого повітря виконується автономним і встановлюється біля маніпулятора.

Розподільвачі стисненого повітря 7 клапанного типу з електрокеруванням виконують відкриття або закриття доступу стисненого повітря в робочі зони пневмоциліндрів. На кожний рух в роботі встановлений автономний електроклапан.

Повернення руки робота навколо її осі відбувається за допомогою пневмодвигуна 8, висувні руки – від пневмоциліндра двосторонньої дії 9, закриття захвату від пневмоциліндра односторонньої дії 10, підйом колони – від пневмоциліндра 11, поворот колони – від двох пневмоциліндрів 12.

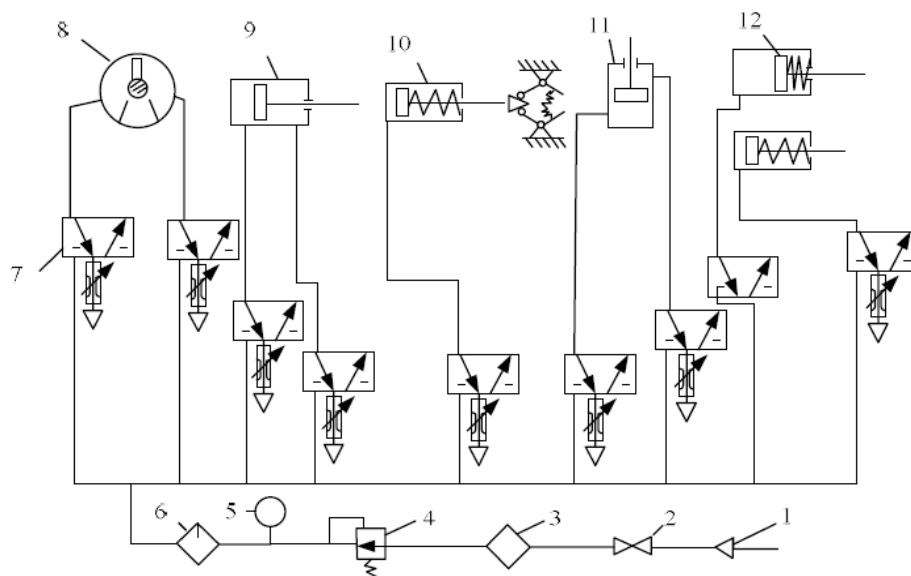


Рисунок 12.9 – Схема пневмоприводу робота РФ-202М

## Компресор

В якості двигуна для приводу компресора використовується асинхронний крановий двигун МТФ 012-6У2 з фазним ротором. Його паспортні дані наведені в таблиці 12.4

Таблиця 12.4 – Паспортні дані двигуна МТФ 012-6У2

Найменування	Значення
Номінальна потужність	$P_n = 2.2 \text{ кВт}$
Номінальна фазна напруга	$U_n = 380 \text{ В}$
Синхронна частота обертання	$n_c = 895 \text{ об/хв}$
Номінальний струм статора	$I_{1n} = 6 \text{ А}$
Номінальний струм ротора	$I_{2n} = 11.5 \text{ А}$
Номінальний ККД	$\eta_o = 87.0\%$
Номінальний коефіцієнт потужності	$\cos(\varphi) = 0.68$
Пусковий момент	$m_n = 1.7 \text{ в.о.}$
Кратність пускового струму	$I_n = 9 \text{ в.о.}$
Критичний момент	$m_k = 2.3 \text{ в.о.}$
Момент інерції ротора	$J = 0.005 \text{ кг} \cdot \text{м}^2$
Номінальне ковзання	$s_H = 2.1\%$
Критичне ковзання	$s_K = 12\%$
Число пар полюсів	$p_n = 3$

## Датчик тиску

Для забезпечення зворотного зв'язку за тиском системи автоматичного регулювання тиску обираємо датчик Dwyer 673-4. Перетворювач тиску має точність 0.25%, являється гарним вибором для відображення тиску рідин і газів. Виготовлений із сталі 17-4 РН, що дозволяє проводити виміри в більшості промислових процесів.

Технічні характеристики наведені в таблиці 12.5.

Таблиця 12.5 – Технічні характеристики давача тиску

Найменування	Значення
Точність	0.25%
Температура навколишнього середовища	-40..+85
Температура підконтрольного середовища	-40..+80
Граничне значення тиску	x2
Час відклику	5 мс
Напруга живлення	9-30 В=
Вихідний сигнал	4-20 мА
Діапазон тиску	0-10 ар

## 12.2. Програма роботи

1. Ознайомитися з функціональною схемою стенду та чітко розуміти призначення кожного блоку системи.
2. Виконати налаштування комп'ютера згідно з методичними вказівками.
3. За таблицею 1 обрати свій варіант.
4. Виконати логічний синтез завдань методом графопереходів.
5. У програмному середовищі SimpLight записати у вигляді скрипту отримане рішення завдань.
6. Перевірити правильність виконання завдання.
7. Оформити протокол по лабораторній роботі та зробити висновки.

Таблиця 12.6 – Варіанти завдань

Номер бригади	Номер завдання
1	1, 7
2	2, 8
3	3, 9
4	4, 10
5	5, 11
6	6, 12

### **Завдання до лабораторної роботи**

1. Відпрацювати наступний алгоритм: робот №1 повернути в право, висунути ліву руку, повернути робот вліво, втягнути руку, повернути робот вправо одночасно висунувши ліву руку, втягнути руку, повернути робот вліво.

2. Відпрацювати наступний алгоритм (на будь якому роботі): поворот вправо, висунути руку (затиснути МЗ), повернути робот вліво (МЗ розтиснути), повернути робот вправо, висунути другу руку (затиснути МЗ), втягнути обидві руки (розтиснути МЗ), повернути робот вліво.

3. Відпрацювати наступний алгоритм: поворот робота вправо, права рука вперед, поворот вліво, ліва рука вперед, поворот вправо, руки назад.

4. Забезпечити переміщення деталей з права на ліво (використовуючи обидві руки маніпулятора).

5. Відпрацювати наступний алгоритм: повернути робот №1 вправо, робот №2 вліво, висунути праві руки двох роботів та затиснути МЗ, повернути роботи в початкове положення.

6. Відпрацювати наступний алгоритм: робот №1 вліво, робот №2 поворот вліво, висунути руки робота №1 та повернути його вправо, висунути руки робота №2, повернути роботи в початкове положення.

7. Забезпечити автоматичне відпрацювання наступної задачі: Маніпулятор №1 встановити в початкове положення (крайне праве), витягнути ліву руку і затиснути МЗ, втягнути руку та повернути колону вліво. Витягнути обидві руки Маніпулятора №1. Маніпулятор №2 повернути вправо витягнути обидві руки. Повернути обидва робота в початкові крайні положення.

8. Забезпечити автоматичне відпрацювання задачі: два маніпулятори в початкове крайне положення, висунути праві руки, повернути навколо горизонтальної осі та висунути ліві руки. Повернути маніпулятори на зустріч один одному. Втягнути руки у наступній послідовності: права рука маніпулятора №1, ліва рука маніпулятора №2, ліва рука маніпулятора №1, права рука маніпулятора №2. Повернути по чергово маніпулятори в крайне початкове положення.

9. Забезпечити автоматичне відпрацювання задачі: поставити маніпулятори на зустріч один одному. Висунути дві руки маніпулятора №1 одночасно, руки маніпулятора №2: спочатку ліву потім праву. Маніпулятора №1 вправо втягнути ліву руку, повернути праву навколо горизонтальної осі та втягнути її разом із лівою рукою маніпулятора №2. Повернути маніпулятор №2 вліво та втягнути праву руку.

10. Використовуючи Маніпулятор №2 виконати наступну задачу: забезпечити автоматичне перенесення трьох деталей (одна деталь за один оберт). Використовуючи Маніпулятор №1 забрати ці деталі.

11. Використовуючи Маніпулятор №1 виконати наступну задачу: забезпечити автоматичне перенесення шести деталей. Використовуючи Маніпулятор №2 перевернути деталі та перемістити їх.

12. Використовуючи маніпулятор виконати подачу нових заготовок до станку та знімання готових. Подача та знімання виконуються в крайніх положеннях, та по чергово (спочатку забрати заготовку, потім подати нову, повернути маніпулятор і знову).

### 12.3. Методичні вказівки

Перед початком роботи необхідно налаштувати комп'ютер для коректної роботи із системою SCADA. Для цього необхідно встановити програму SimpLight, Lectus OPC та необхідні драйвери для адаптера гальванічної розв'язки RS-485. Інструкція із послідовністю встановлення прикладена разом із установочним файлом. Зверніть увагу, що програму SimpLight потрібно встановлювати в режимі сервера (встановити галочку «Сервер»).

Після встановлення всіх вище перерахованих програм перейдемо до налаштування портів комп'ютера. Підключивши USB-провід до комп'ютера потрібно відкрити диспетчер пристроїв Пуск/ Панель управління/ Система/ Оборудование/ Диспетчер устройств/ Порты (COM и LTP)/ USBSerialPort. Нажимаємо правою кнопкою та відкриваємо «Свойства». Потрібно відкрити вкладку «Параметры порта» та встановити всі параметри як показано на

рисунку 12.10 (Скорость – 57600, Биты данных – 8, Четность – нет, стоп биты – 2, Управление потоком – нет).

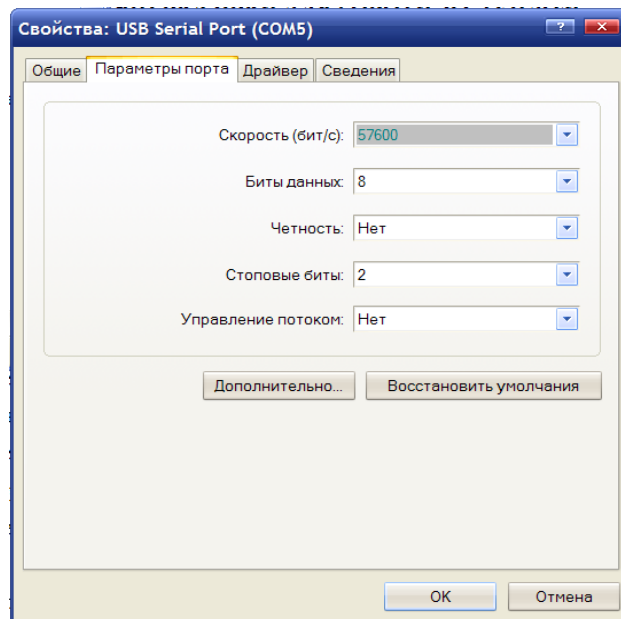


Рисунок 12.10 – Налаштування порту комп'ютера

Натиснувши кнопку «Дополнительно» вибираємо COM 5 серед переліку портів. Бажано перевірити чи не використовується цей порт іншими пристроями або програмами, щоб не виникло конфлікту при обміні даними з ПЛК.

Після налаштування портів перейдемо до запуску програм. Lectus OPC програма, яка призначена для забезпечення «спілкування» комп'ютера та ПЛК, представлена на рисунку 12.11.

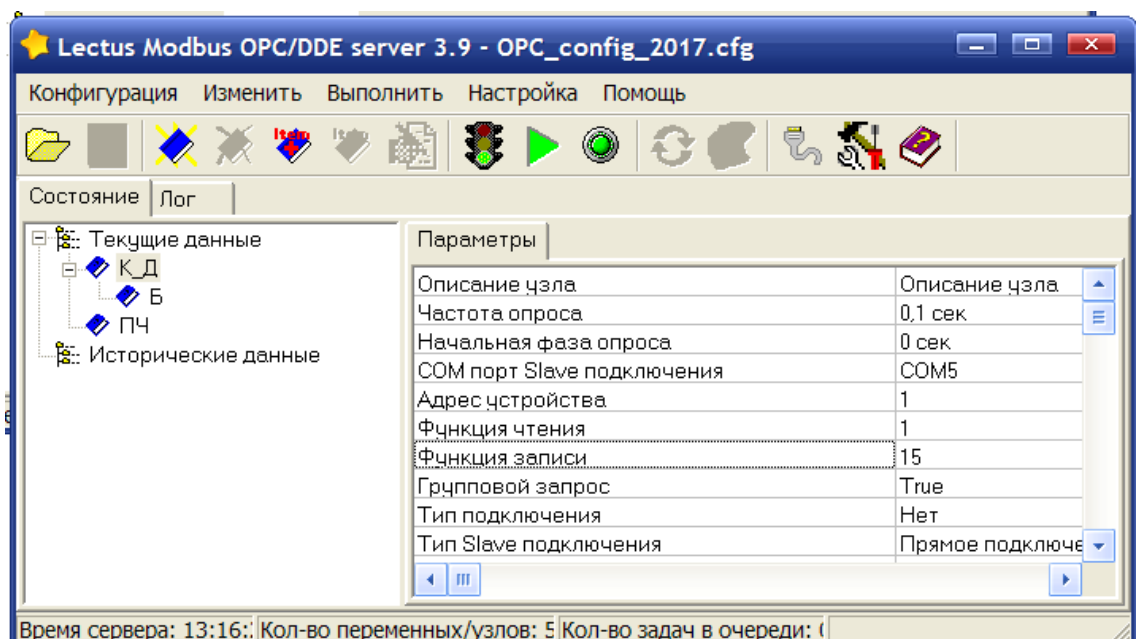



Рисунок 12.11 – Програма Lectus OPC

Перед початком роботи потрібно завантажити файл конфігурації до Lectus OPC. Для цього натискаємо , далі знаходимо потрібний файл конфігурації (OPC\_config\_2017.cfg) та натискаємо «Открыть».

Слід переконатись що налаштування COM порту в Lectus OPC співпадає із налаштуванням на комп'ютері. Для цього відкриваємо вкладку «Настройка»/COM порт. Вибираємо з переліку COM 5 та перевіряємо налаштування (рисунок 12.12).

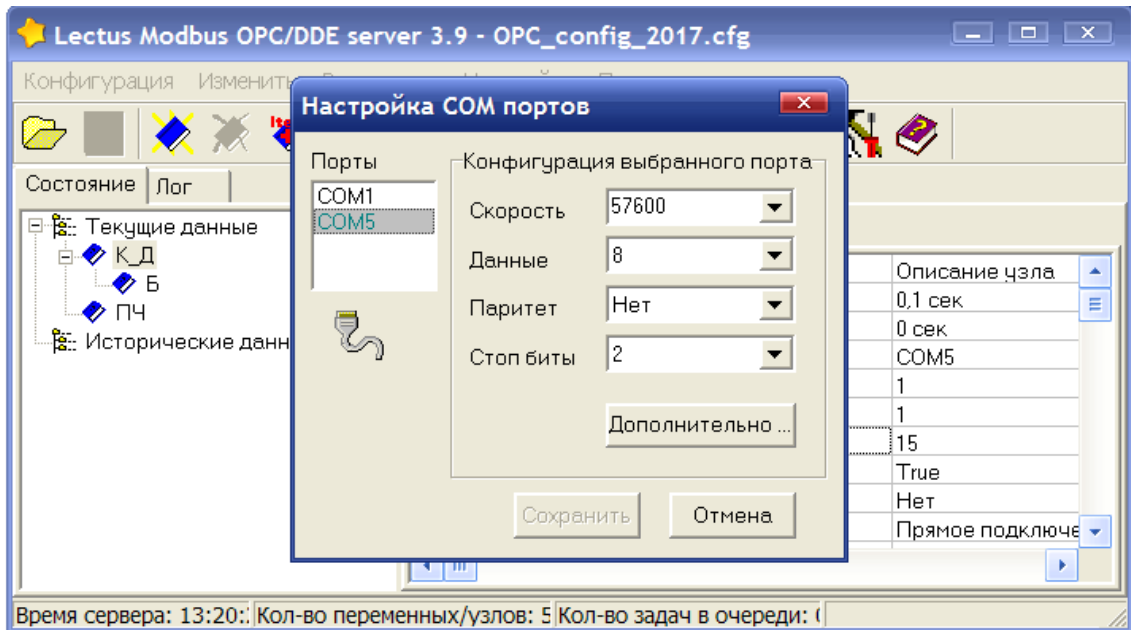


Рисунок 12.12 – Налаштування COM порту в програмі

Далі потрібно перевірити правильні налаштування вузлів. Для цього натискаємо правою кнопкою миші на К\_Д вибираємо «Свойства». Відкриється наступне вікно.

The screenshot shows the 'Добавить узел' (Add Node) dialog box for a Modbus node. The node name is 'К\_Д'. The description is 'Описание узла'. The connection is set to 'Нет' (None). The data type is 'Текущие данные' (Current data). The polling period is 50 ms and the initial phase is 0 ms. The device settings are: Protocol: Modbus RTU, Address: 1, Read function: 01, Write function: 15, and Group request is checked. There is a 'Настройка...' (Settings...) button in the connection section and a 'Параметры...' (Parameters...) button in the additional section. The 'Изменить' (Change) and 'Отмена' (Cancel) buttons are at the bottom.



Рисунок 12.13 – Вікно налаштувань вузла К\_Д

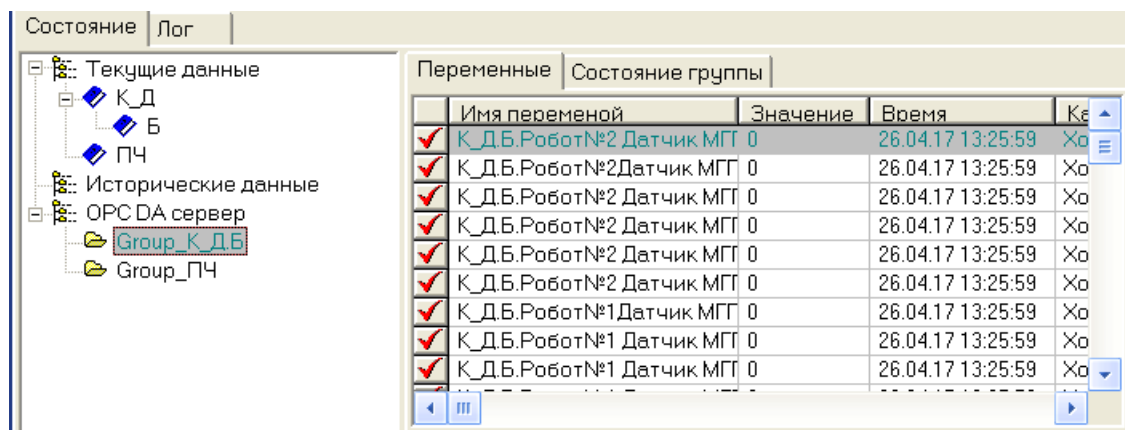
Виставляємо всі параметри згідно з рисунком 12.13. При відкритті «Свойств» вузла ПЧ потрібно виставити наступні параметри (рисунок 12.14).

The screenshot shows the 'Добавить узел' (Add Node) dialog box for a Modbus node named 'ПЧ'. The description is empty. The connection is set to 'Нет' (None). The data type is 'Текущие данные' (Current data). The polling period is 100 ms and the initial phase is 0 ms. The device settings are: Protocol: Modbus RTU, Address: 1, Read function: 03, Write function: 16, and Group request is checked. There is a 'Настройка...' (Settings...) button in the connection section and a 'Параметры...' (Parameters...) button in the additional section. The 'Изменить' (Change) and 'Отмена' (Cancel) buttons are at the bottom.

Рисунок 12.14 – Вікно налаштувань вузла ПЧ

При «Параметры» у полі «Допольнительно» потрібно виставити в «Тип подключения» - пряме підключення.

Опитування ПЛК відбувається шляхом натиснення . Якщо горить зелений це значить що відбувається опитування серверу, якщо червоний – ні. При вимкненому ПЛК можна запустити тестове опитування (рисунок 12.15) натисненням . Цей режим не бажано використовувати при роботі з роботами. При цьому з'явиться наступний пункт в дереві.



Имя переменной	Значение	Время	К...
К_Д_Б.Робот№2 Датчик МГТ	0	26.04.17 13:25:59	Хо
К_Д_Б.Робот№2 Датчик МГТ	0	26.04.17 13:25:59	Хо
К_Д_Б.Робот№2 Датчик МГТ	0	26.04.17 13:25:59	Хо
К_Д_Б.Робот№2 Датчик МГТ	0	26.04.17 13:25:59	Хо
К_Д_Б.Робот№2 Датчик МГТ	0	26.04.17 13:25:59	Хо
К_Д_Б.Робот№1 Датчик МГТ	0	26.04.17 13:25:59	Хо
К_Д_Б.Робот№1 Датчик МГТ	0	26.04.17 13:25:59	Хо
К_Д_Б.Робот№1 Датчик МГТ	0	26.04.17 13:25:59	Хо

Рисунок 12.15 – Работа програми при тестовому опитувані

Група «OPC DA сервер» відображає усі підключені клапани та датчики. Також програма Lectus OPC контролює якість зв'язку та сповіщає про помилки.

Такі ж пункти з'являться якщо увімкнути ПЛК, ПЧ та запустити опитування серверу.

Далі перейдемо до роботи із програмою SimpLight. Відкриваємо програму «Управление проектами» (рисунок 12.16). За допомогою цієї програми нам потрібно імпортувати файл з проектом для роботи з системою керування промисловими роботами РФ-202м.

Для імпортування відкриваємо вкладку «Проект»/ Имортировать проект. Далі вибираємо файл «magister.slz» та натискаємо «Открыть». В дереві проектів з'явиться вкладка «Текущий проект» із назвою «magister».

У вкладці «Другие проекты» будуть знаходитись усі раніше створенні проекти. Можна також створювати резервні копії проектів, які будуть сортуватись по року розробки та місяцю.

Далі відкриваємо вкладку «Запустить» та відкриваємо «Редактор каналов» (рисунок 12.16). Разом із ним має відкритись Lectus OPC (якщо він був закритий).

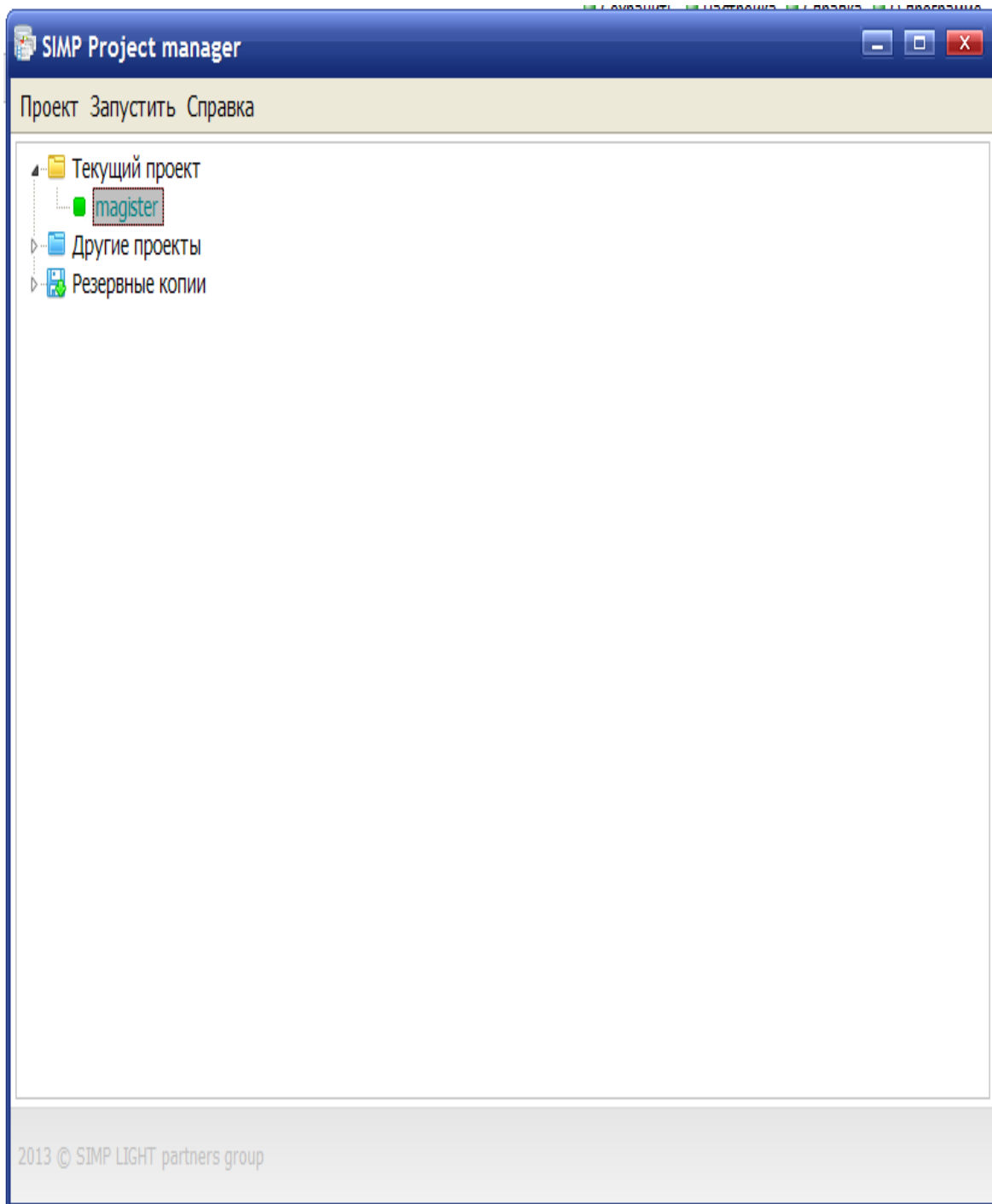


Рисунок 12.16 – Імпортування проекту.

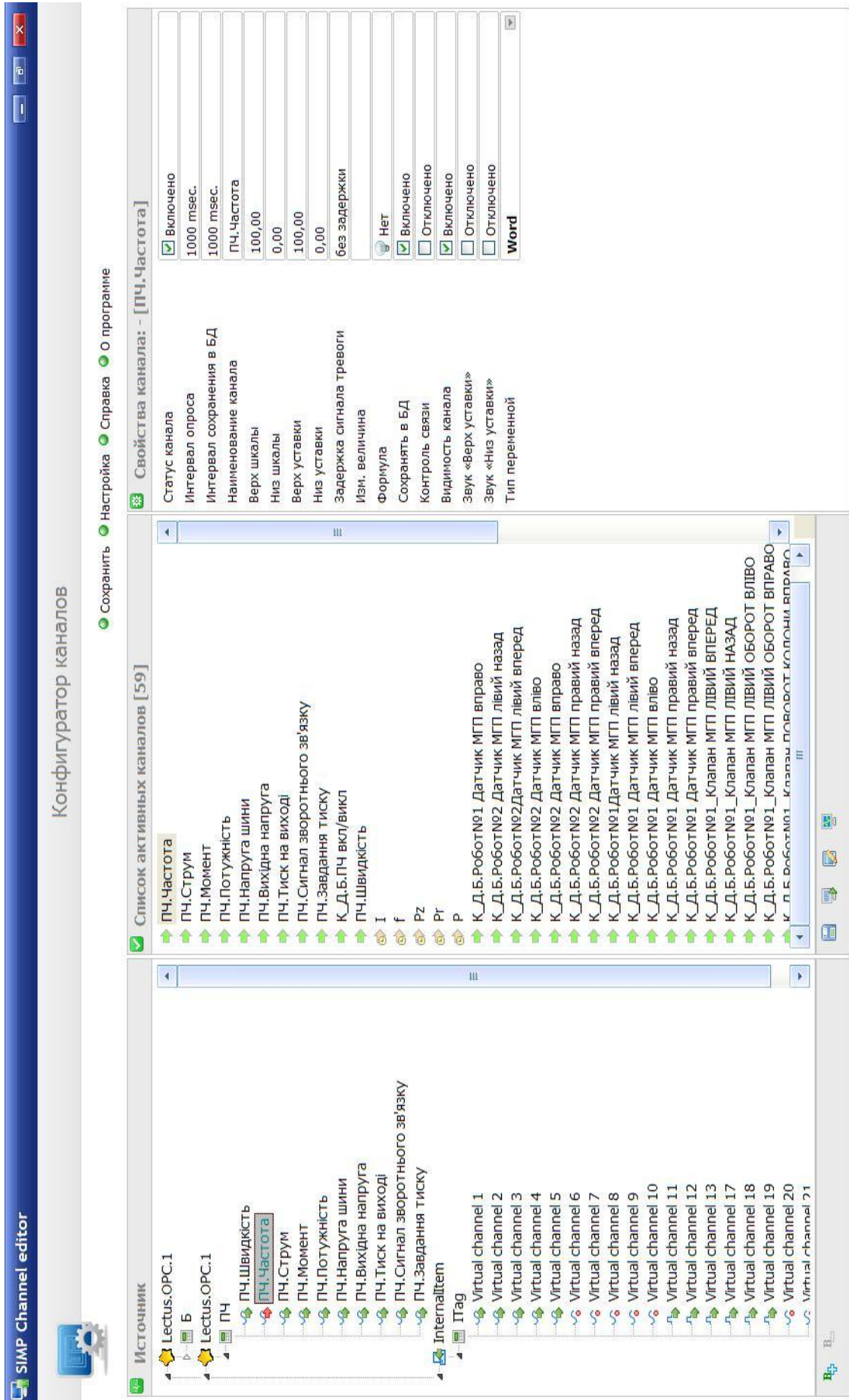


Рисунок 12.17 – Конфігуратор каналів

Для того щоб отримати інформацію від Lectus OPC потрібно відкрити вкладку «Настройка» та натиснути «Список OPC серверов». У відкритому переліку поставити галочку тільки навпроти «Lectus.OPC.1» та натиснути ОК.

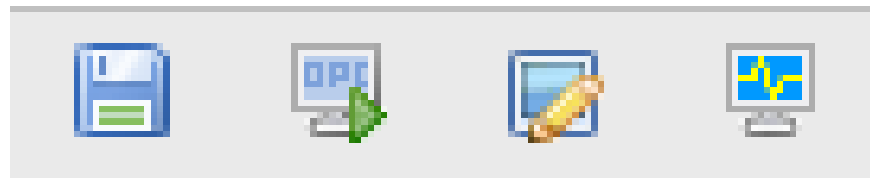
В полі під назвою «Источник» відображаються список OPC серверів, що підключені, та доступні канали. Розкривши дерево ми побачимо всі ті ж канали що і бачили в Lectus OPC.

Окрім OPC серверів в дереві присутня вкладка «Internal Item». В цій вкладці знаходиться список віртуальних каналів, які використовуються для створення програми автоматичного керування роботами за допомогою скрипту.

Для того щоб зробити канал активним його потрібно перемістити в поле «Список активних каналов» (стрілочка навпроти назви каналу стане зеленою). В цьому полі відображаються канали з якими будемо далі працювати.

В полі «Свойства канала» буде відображатись налаштування та параметри кожного каналу. Також тут відображується чи має канал скрипт, його часовий інтервал опитування, назва, тип даних та інше.

Нижче знаходиться меню швидкого доступу, яке має вигляд як на рисунку 12.18.



*Рисунок 12.18 – Меню швидкого доступу*

Меню виконує наступні функції: зберегти, запустити тест OPC каналів, запустити редактор мнемосхем, запустити монітор відповідно.

Програма переходить в режим роботи при відкритті монітора. Проте перед запуском радимо відкрити «Редактор мнемосхем» та ознайомитись із панеллю керування роботами. Мнемосхема представлена на рисунку 12.19.

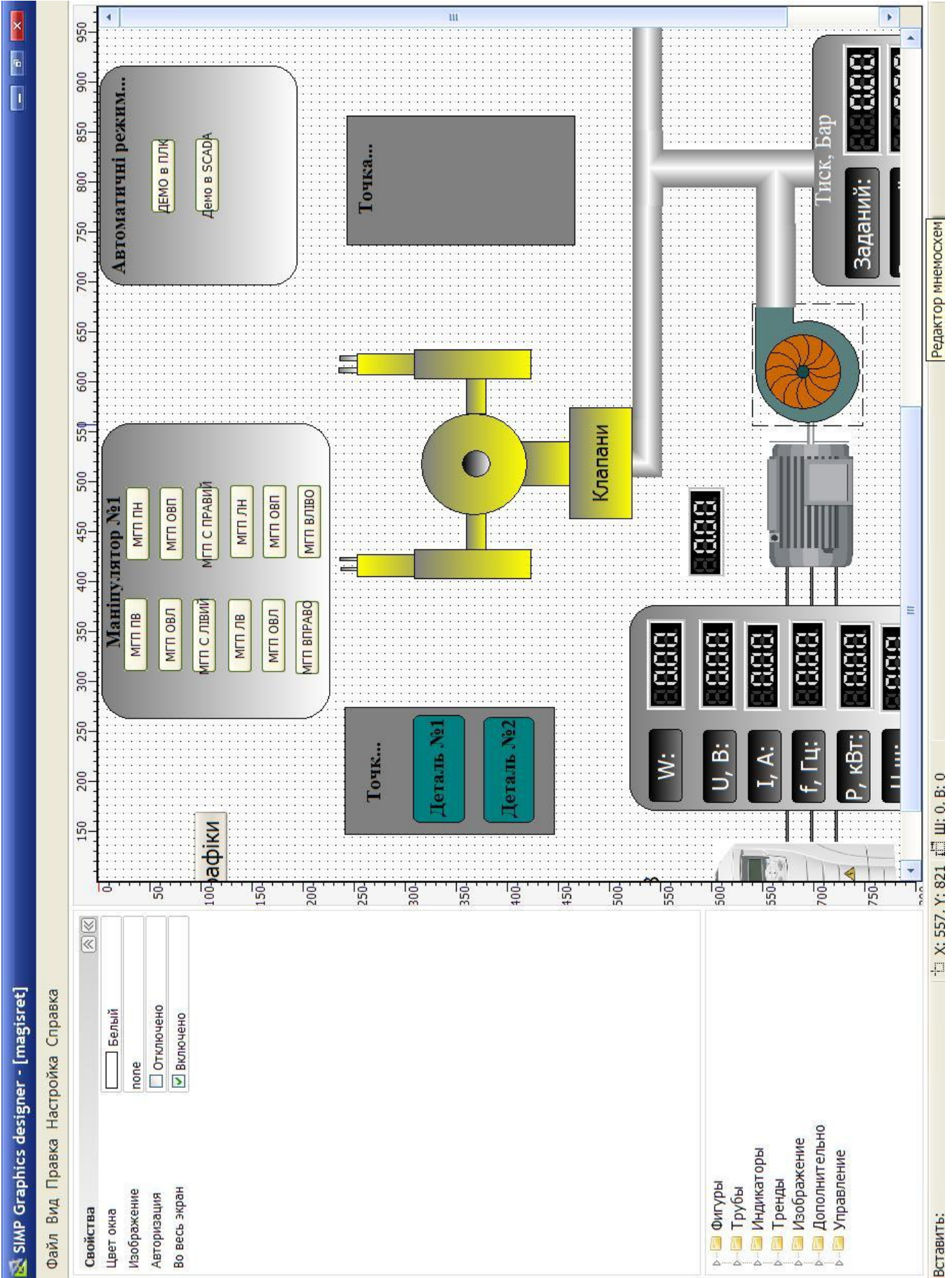


Рисунок 12.19 – Редактор мнемосхем

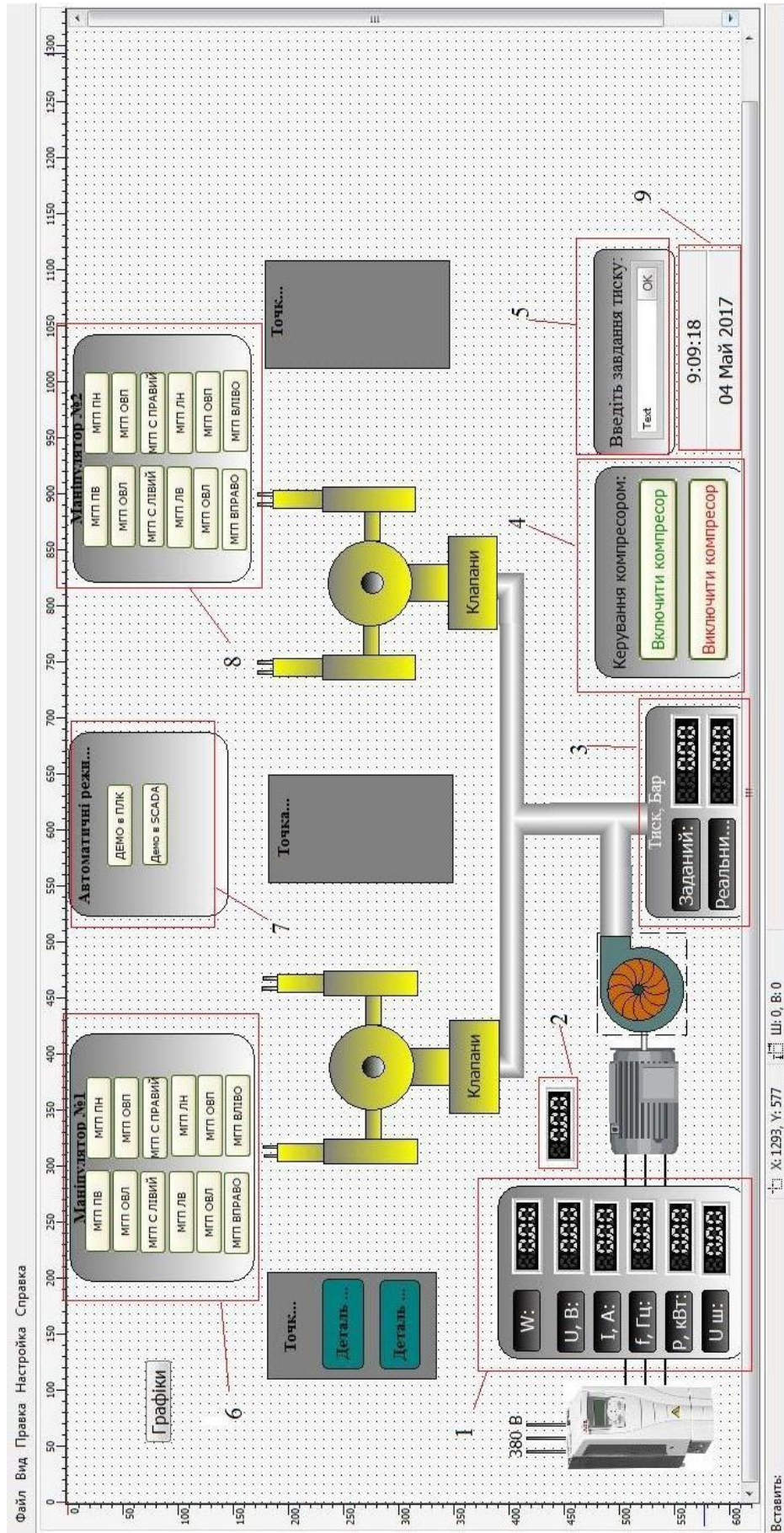


Рисунок 12.20 – Мнемосхема технологічного процесу

Для того щоб відкрити мнемосхему потрібно відкрити вкладку «Файл» далі «Открыть», вибрати файл з назвою «magister», натиснути «Открыть».

В колонці «Свойства» відображується налаштування кожного елементу мнемосхеми та показує яка кнопка, за який сигнал відповідає.

Дана мнемосхема відображує весь технологічний процес керування, та схему послідовності підключень (рисунки 12.21).

1. Параметри перетворювача частоти;
2. Зворотній зв'язок за тиском;
3. Завдання тиску та реальний тиск на виході компресора;
4. Кнопки керуванням компресором;
5. Поле для вводу завдання тиску;
6. Панель ручного керування першим маніпулятором;
7. Панель запуску автоматичного режиму роботи маніпуляторів (ДЕМО в ПЛК – запускає програму, що записана в ПЛК, Демо в SCADA – запускає програму, що створено в редакторі каналів);
8. Панель ручного керування другим маніпулятором;
9. Реальні дата та час;

Перейдемо до створення програми автоматичного керування маніпуляторами в SCADA. Кожному студенту буде дана задача, яка вирішується за допомогою методу графа переходів. Розглянемо на прикладі простої задачі, яка покаже принцип і основну ідею програмування задачі.

Умова задачі: запрограмувати один маніпулятор для виконання обертових рухів: спочатку вправо потім вліво.

Вхідними сигналами будуть сигнали керування геконовими датчиками, які представлені в таблиці 12.5.

Таблиця 12.5 – Вхідні сигнали

Сигнал	Датчик
$a_1$	К_Д.Б. Робот №1 Датчик МГП вправо
$a_2$	К_Д.Б. Робот №1 Датчик МГП вліво

Наведемо вихідні функції в таблиці 12.6.

Вихідна функція	Сигнал переходу	Опис стану
$K_1$	$a_1$	Робот №1 поворот вправо
$K_2$	$a_2$	Робот №1 поворот вліво

Для синтезу задач на тригерах умови роботи схеми подаються у вигляді графів переходів. Граф переходів – це графічне зображення послідовності роботи багатоактної схеми. Елементами графа є вершини і ребра. Вершини відповідають станам схеми і позначаються кружками. Ребра – це лінії із стрілками, що з'єднують вершини і показують напрям переходу з одного стану схеми в інший.

Кількість вершин графа при синтезі асинхронних схем на RS - тригерах визначається з умови  $2^n > S$ , де  $S$  – кількість станів схеми;  $2^n$  – кількість вершин графа;  $n$  – кількість тригерів.

Вершини графа рекомендується розміщувати так, щоб при  $n=2$  вони створювали конфігурацію  $2 \times 2$ , при  $n=3$  – конфігурацію  $4 \times 2$ , а при  $n=4$  – конфігурацію  $4 \times 4$ .

Вихідні стани тригерів виконують роль проміжних змінних, тому тригери та їх вихідні сигнали позначаються буквами  $P_1, P_2, \dots$ . Ці позначення записують зверху ліворуч від графа.

Кожна вершина графа кодується набором значень вихідних сигналів тригерів. Коди вибираються так, щоб для сусідніх вершин вони відрізнялися значенням тільки однієї змінної. У вихідному стані схеми (стан очікування) звичайно приймають, що усі проміжні змінні дорівнюють нулеві. Значення проміжних змінних для кожної вершини записуються над кружками у тій черговості, в якій записані позначення тригерів.

Вершини, між якими повинні відбуватися переходи, з'єднують ребрами із стрілками. Над стрілками або праворуч від них, якщо ребра спрямовано вертикально, записують позначення вхідних сигналів, що спричинюють ці переходи.

При побудові схем на асинхронних RS – тригерах переходи можна робити тільки між сусідніми вершинами. Якщо ця умова не виконується, то необхідно передбачити так звані природні переходи (за рахунок подавання вхідного сигналу одиниця) через проміжні нестійкі стани.

Побудову графа переходів починаємо з визначення кількості станів, в яких може перебувати схема автоматичного керування.

Виходячи із кількості станів схеми, визначаємо кількість тригерів і кількість вершин графа переходів. Позначаємо проміжні змінні  $P_1, P_2, P_3$ . Граф переходів зображено на рисунку 12.22.

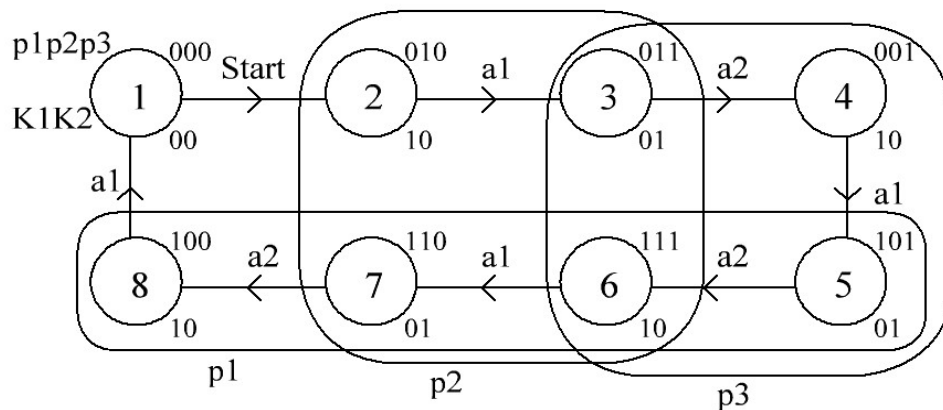


Рисунок 12.22 – Граф переходів

Кожному станіві схеми ставимо у відповідність одну з вершин графа, причому стани, між якими повинен відбуватися перехід згідно з умовами роботи схеми, розміщуємо у сусідніх вершинах.

Синтез схеми полягає у записі умов вмикання і скидання кожного тригера. Для цього охопили замкнутою лінією всі стани на графі переходів, в яких значення вихідного сигналу даного тригера дорівнює одиниці. Вхідні сигнали схеми, позначення яких стоять на ребрах, що заходять в одержану замкнуту область, встановлюють тригер в стан 1, а вихідні сигнали на ребрах, що виходять з цієї області, скидають тригер в стан 0.

Умови вмикання тригера записані у вигляді добутку сигналу на ребрі, що заходить в область, і сигналів решти тригерів, стан яких не змінюється при переході, позначеному ребром.

Якщо у замкнену область входить кілька ребер, то умова вмикання тригера записується у вигляді суми добутків відповідних сигналів, складених для кожного ребра.

Умова скидання тригера записується аналогічно для кожного ребра, що виходять з даної області, і подається у вигляді формули  $R_{pi}$ . Описану процедуру виконують для кожного тригера і визначають для них умови вмикання і скидання.

Застосувавши описану процедуру визначення умов вмикання і скидання тригерів, для графа переходів на рисунку отримаємо:

$$S_{p1} = a_1 \bar{p}_2 p_3; R_{p1} = a_1 \bar{p}_2 \bar{p}_3;$$

$$S_{p2} = start \cdot \bar{p}_1 \bar{p}_3; R_{p2} = a_2 p_1 \bar{p}_3 + a_2 \bar{p}_1 p_3;$$


$$S_{p3} = a_1 \cdot \bar{p}_1 p_2; R_{p3} = a_1 p_1 p_2;$$

Запишемо вирази для вихідних функцій:

$$K_1 = \bar{p}_1 p_2 \bar{p}_3 + \bar{p}_1 \bar{p}_2 p_3 + p_1 p_2 p_3 + p_1 \bar{p}_2 \bar{p}_3;$$

$$K_2 = \bar{p}_1 p_2 p_3 + p_1 \bar{p}_2 p_3 + p_1 p_2 \bar{p}_3;$$

Далі отримані рівняння потрібно записати у вигляді скрипту до віртуальних каналів в програмі SimpLight в редакторі каналів. Для створення скрипту у віртуальному каналі потрібно:

1. Створити віртуальний канал натиснувши  та перетягнути його у поле «Список активних каналов»;
2. В налаштуваннях каналу виставити параметри, як на рисунку 12.23;




Статус канала	<input checked="" type="checkbox"/> Включено
Интервал опроса	100 msec.
Интервал сохранения в БД	1000 msec.
Наименование канала	p2
Задержка сигнала тревоги	без задержки
Скрипт	 Java Script 
Сохранять в БД	<input checked="" type="checkbox"/> Включено
Видимость канала	<input checked="" type="checkbox"/> Включено
Контроль состояния «Вкл.»	<input type="checkbox"/> Отключено
Контроль состояния «Откл.»	<input type="checkbox"/> Отключено
Тип переменной	<b>Boolean</b> 

Рисунок 12.23 – Параметри віртуального каналу

3. Натиснути на поле «Скрипт» та відкрити налаштування програмування шляхом натиснення кнопки що виділена квадратом (рисунок 12.24);



Рисунок 12.24 – Відкриття скрипту каналу

Налаштування скрипту має вигляд, як на рисунку 12.25

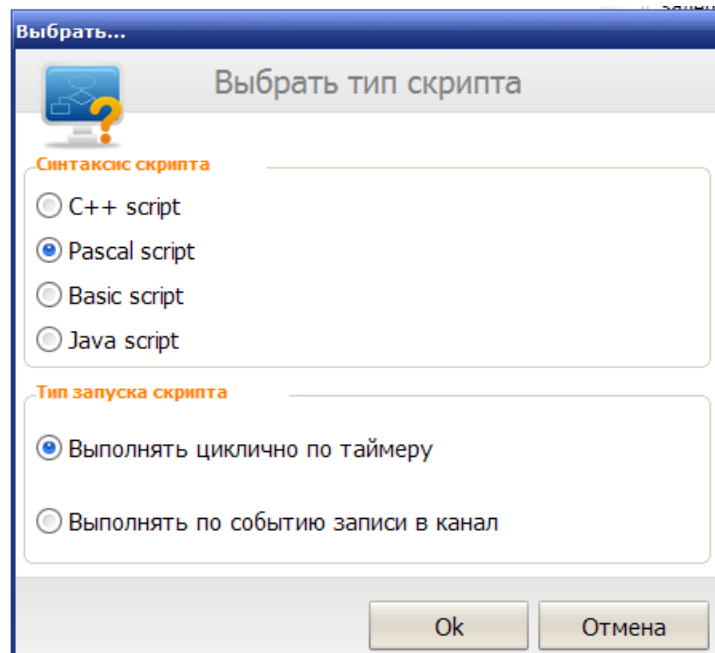


Рисунок 12.25 – Налаштування скрипта

Користувачеві пропонують на вибір чотири мови програмування. В даній моделі використовується мова програмування Java. В цьому випадку програмування не потребує великих знань в даній мові. Детально розібравши приклад програмування все буде зрозуміло.

Тип запуску скрипта вибираємо «Выполнять циклично по таймеру». Натиснувши ОК ми перейдемо безпосередньо в середовище програмування (рисунок 12.26).

В вікні, що знаходиться зліва буде знаходитися текст програми з рівняннями, які були отримані в ході рішення графа переходу. В вікні, що справа під назвою «Каналы» будуть відображатись список всіх активних каналів, що нам доступні.

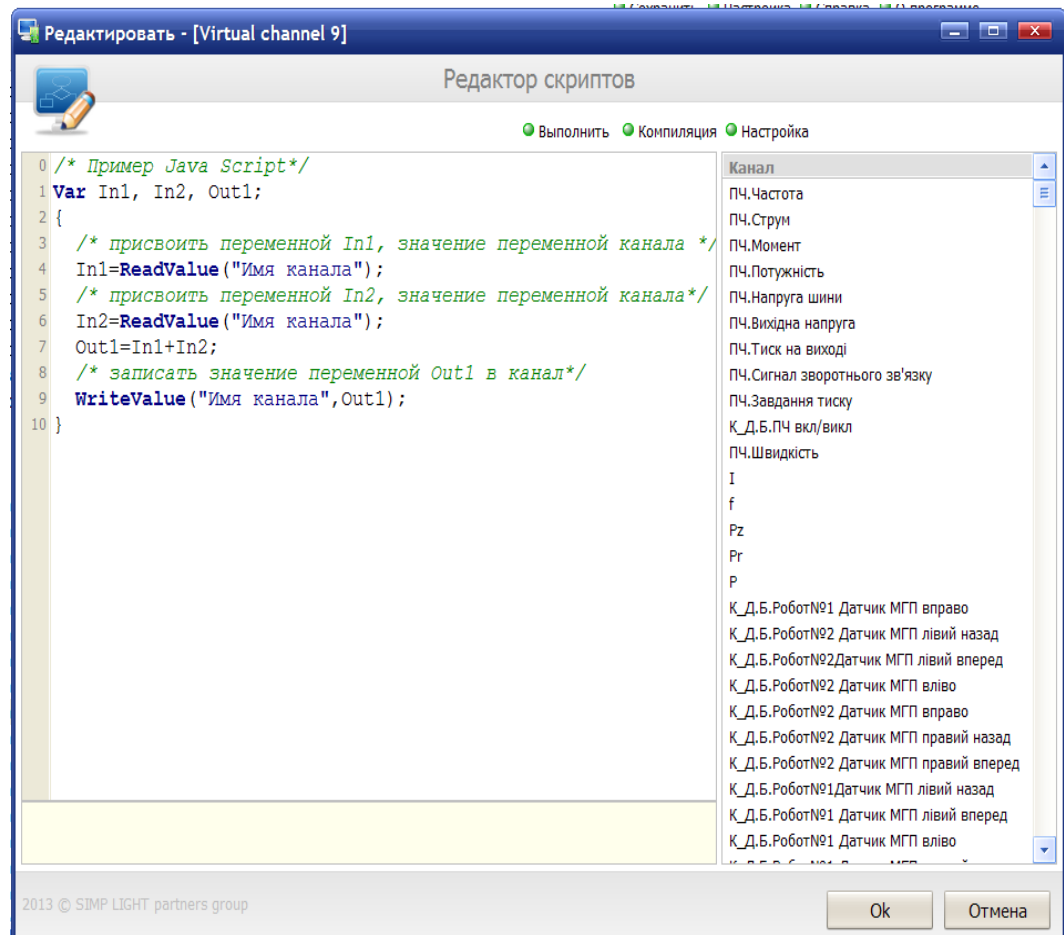


Рисунок 12.26 – Середовище програмування

Запишемо наші рівняння у вигляді Java script:

Рівняння для першого тригера:

```

/* Пример JavaScript*/
Var p1, p2, p3, a1; //оголошуємо змінні які будемо використовувати
{
  p2=ReadValue("p2"); // зчитуємо величину проміжної змінної p2
  p3=ReadValue("p3"); // зчитуємо величину проміжної змінної p3
  a1=ReadValue("К_Д.Б.Робот№1 Датчик МГП вправо"); // зчитуємо сигнал з датчика
повороту платформи
  if (a1 && (!p2) && p3){
    p1 = 1; // записуємо рівняння вмикання першого тригера
    WriteValue("p1",p1);
  }
  elseif (a1 && (!p2) && (!p3)){

```

```

    p1 = 0; // записуємо рівняння скидання першого тригера
WriteValue("p1",p1);
} }

```

Запис рівнянь для другого тригера. Оскільки сигналом початку роботи слугує натиснення кнопки старт, то вводимо відповідну змінну «start». Необхідно також створити віртуальний канал, який буде відповідати сигналу з кнопки.

```

/* Пример JavaScript*/
Var p1, p2, p3, a2, start;
{
    p1=ReadValue("p1");
    p3=ReadValue("p3");
    a2=ReadValue("К_Д.Б.Робот№1 Датчик МГП вліво");
    start=ReadValue("start"); // зчитуємо сигнал натиснення кнопки

    if (start&& (!p1) && (!p3) || a2 && p1 && p3){
        p2 = 1;
        WriteValue("p2",p2);
    }
    elseif (a2 && (!p1) && p3 || a2 && p1 && (!p3)){
        p2 = 0;
        WriteValue("p2",p2);
    }
}

```

Для третього тригера текст програми буде по аналогії до першого. Запишемо рівняння для вихідних функцій. Текст програми для першої вихідної функції.

```

/* Пример JavaScript*/
Var p1, p2, p3, k1;
{
    p1=ReadValue("p1");
    p2=ReadValue("p2");
    p3=ReadValue("p3");
    start=ReadValue("start");
    if (start == "1") // умова що перевіряє чи натиснута кнопка «старт»
    {

```

```

k1 = (!p1) && p2 && (!p3) || (!p1) && (!p2) && p3 || p1 && p2 && p3 || p1 && (!p2) && (!p3);
WriteValue("К_Д.Б.Робот№1_Клапан ПОВОРОТ КОЛОНИ ВПРАВО",k1);
}
}

```

Текст програми для другої вихідної функції:

```

/* Пример JavaScript*/
Var p1, p2, p3, k2;
{
  p1=ReadValue("p1");
  p2=ReadValue("p2");
  p3=ReadValue("p3");
  start=ReadValue("start");

  if (start == "1"){

    k2 = ((!p1) && p2 && p3 || p1 && (!p2) && p3 || p1 && p2 && (!p3));
    WriteValue("К_Д.Б.Робот№1_Клапан ПОВОРОТ КОЛОНИ ВЛІВО",k2);
  }
}

```

Слід звернути увагу на те, що перехід у наступний стан відбувається по сигналу датчика. Відповідно в тексті скрипту перемінним, що відповідають сигналам переходу потрібно присвоювати канал відповідного датчика. Для вихідних функцій потрібно присвоювати канал відповідного клапану.

Після запису всіх рівнянь до скрипту відповідного каналу перейдемо до виконання даного завдання. Для цього відкриваємо в редакторі каналів монітор. На стартовій сторінці буде відображатись список всіх клапанів та датчиків, а також список каналів що вийшли за межі уставки. Вони будуть блимати жовтим, сигналізуючи про вихід за межі уставки, до тих пір поки не натиснути на них.

Щоб відкрити мнемосхему в робочому режимі потрібно натиснути «Мнемосхема» та вибрати файл з назвою «magister». Відкриється вікно що зображена на рисунку 12.27.

В поле завдання тиску вводимо значення 4 Бар та натискаємо ENTER. Далі вмикаємо компресор, про ввімкнення буде слугувати збільшення значення зворотного зв'язку.

Для ввімкнення режиму автоматичного керування потрібно натиснути на панелі автоматичного режиму керування «Демо в SCADA». Програма буде повторюватись циклічно раз за разом. Щоб вимкнути режим автоматичного керування потрібно натиснути на кнопку «Start» та вимкнути в ручну всі ввімкненні клапани на палені ручного керування маніпуляторами.

Після завершення лабораторної роботи потрібно відімкнути стенд. Спочатку виставляємо в полі завдання тиску значення нуль та натискаємо кнопку вимкнути компресор (про вимкнення компресора буде свідчити зниження значення зворотного зв'язку та спадання значення швидкості до нуля). Якщо цього не зробити може виникнути помилка: при вимкненому живленні ПЛК, при спаданні значення тиску нижче заданого, може запуснитися компресор (потрібно знову запускати всі програми та вимикати компресор). Вимикаємо живлення ПЛК та вимикаємо комп'ютер.

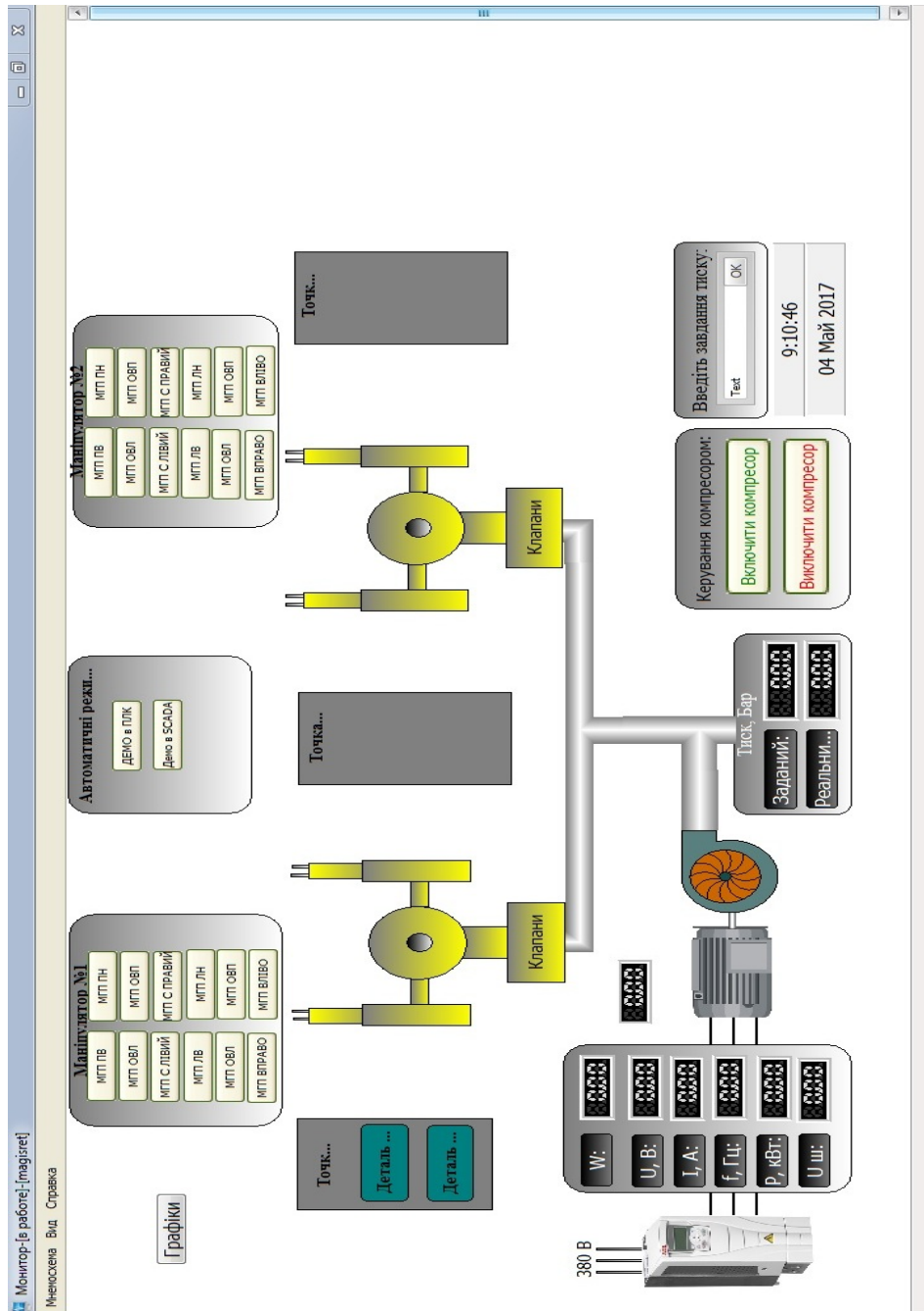


Рисунок 12.27 – Мнемосхема в робочому режимі

### Приклад рішення задачі

Для прикладу розглянемо наступну задачу

1. Відпрацювати наступний алгоритм (на будь якому роботі): поворот вліво, висунути руку (затиснути МЗ), повернути робот вправо (МЗ розтиснути), повернути робот вліво, висунути другу руку, втягнути обидві руки, повернути робот вправо.

В таблиці 12.7 наведено вхідні сигнали з герконових датчиків.

Таблиця 12.7 – Вхідні сигнали герконових датчиків

Сигнал	Координата
a <sub>1</sub>	Маніпулятор №1 МГП поворот вліво
a <sub>2</sub>	Маніпулятор №1 МГП поворот вправо
b <sub>1</sub>	Маніпулятор №1 МГП права рука вперед
b <sub>2</sub>	Маніпулятор №1 МГП права рука назад
c <sub>1</sub>	Маніпулятор №1 МГП ліва рука вперед
c <sub>2</sub>	Маніпулятор №1 МГП ліва рука назад

Вихідними сигналами керуються клапани. Перелік вихідних сигналів наведено в таблиці 12.8.

Таблиця 12.8 – Вихідні сигнали

Сигнал	Координата
K <sub>1</sub>	Маніпулятор №1 МГП поворот вліво
K <sub>2</sub>	Маніпулятор №1 МГП поворот вправо
K <sub>3</sub>	Маніпулятор №1 МГП правий затиснення МЗ
K <sub>4</sub>	Маніпулятор №1 МГП ліва рука вперед
K <sub>5</sub>	Маніпулятор №1 МГП ліва рука назад
K <sub>6</sub>	Маніпулятор №1 МГП права рука вперед
K <sub>7</sub>	Маніпулятор №1 МГП права рука назад

Граф переходів для цієї задачі зображено на рисунку 12.28.

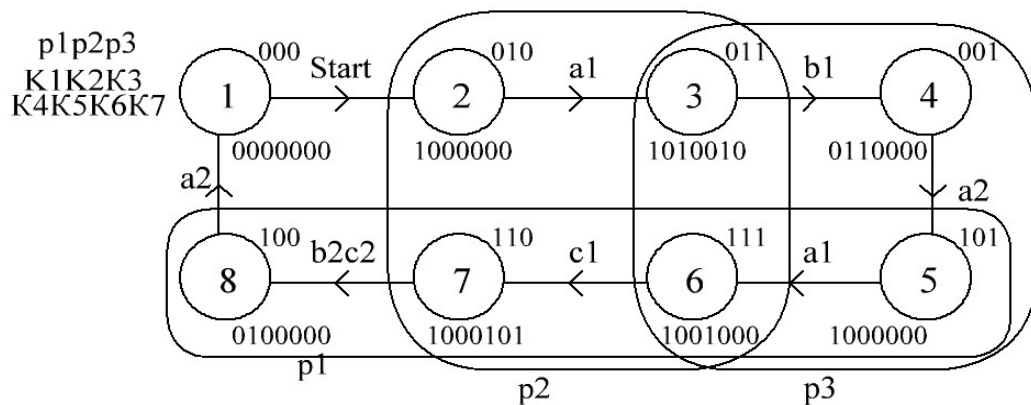


Рисунок 12.28 – Граф переходів

Рівняння встановлення та скидання тригерів наступні:

$$S_{p1} = a_2 \bar{p}_2 p_3; R_{p1} = a_2 \bar{p}_2 \bar{p}_3;$$

$$S_{p2} = \text{Start } \bar{p}_1 \bar{p}_3 + a_1 p_1 p_3; R_{p2} = b_2 c_2 p_1 \bar{p}_3 + b_1 \bar{p}_1 p_3;$$

$$S_{p3} = a_1 \bar{p}_1 p_2; R_{p3} = c_1 p_1 p_2;$$

Рівняння вихідних сигналів мають наступний вигляд:

$$K_1 = \bar{p}_1 p_2 + p_1 p_3 + p_1 p_2 \bar{p}_3;$$

$$K_2 = \bar{p}_1 \bar{p}_2 p_3 + p_1 \bar{p}_2 \bar{p}_3;$$

$$K_3 = \bar{p}_1 p_2 p_3 + \bar{p}_1 \bar{p}_2 p_3;$$

$$K_4 = p_1 p_2 p_3;$$

$$K_5 = p_1 p_2 \bar{p}_3;$$

$$K_6 = \bar{p}_1 p_2 p_3;$$

$$K_7 = p_1 p_2 \bar{p}_3;$$

Напишемо рівняння для встановлення та скидання першого тригера у вигляді скрипту:

```
/* Пример JavaScript*/
Var p1, p2, p3, a2;
{
  p2=ReadValue("p2");
  p3=ReadValue("p3");
  a2=ReadValue("К_Д.Б.Робот№1 Датчик МГП вправо");
  // d2=ReadValue("К_Д.Б.Робот№1 Датчик МГП лівий назад");
  /*b1=ReadValue("К_Д.Б.Робот№1 Датчик МГП вправо");*/
  if (a2 && (!p2) && p3){
    p1 = 1;
    WriteValue("p1",p1);
  }
  elseif (a2 && (!p2) && (!p3)){
    p1 = 0;
    WriteValue("p1",p1);
  }
}
```

Скрипт встановлення та скидання другого тригера:

```
/* Пример JavaScript*/
Var p1, p2, p3, b2, c2, a1, b1, start;
{
  p1=ReadValue("p1");
  p3=ReadValue("p3");
```

```

b1=ReadValue("К_Д.Б.Робот№1 Датчик МГП правий вперед");
a1=ReadValue("К_Д.Б.Робот№1 Датчик МГП вліво");
c2=ReadValue("К_Д.Б.Робот№1 Датчик МГП лівий назад");
b2=ReadValue("К_Д.Б.Робот№1 Датчик МГП правий назад");
start=ReadValue("start");

if (start&& (!p1) && (!p3) || a1 && p1 && p3){
    p2 = 1;
    WriteValue("p2",p2);
}
elseif (b2 && c2 && p1 && (!p3) || b1 && (!p1) && p3){
    p2 = 0;
    WriteValue("p2",p2);
}
}

```

### Для p3

```

/* Пример JavaScript*/
Var p1, p2, p3, a1, c1;
{
    p1=ReadValue("p1");
    p2=ReadValue("p2");
    c1=ReadValue("К_Д.Б.Робот№1 Датчик МГП лівий вперед");
    a1=ReadValue("К_Д.Б.Робот№1 Датчик МГП вліво");
    if (a1 && (!p1) && p2){
        p3 = 1;
        WriteValue("p3",p3);
    }
    elseif (c1 && p1 && p2){
        p3 = 0;
        WriteValue("p3",p3);
    }
}

```

### Скрипт для вихідного сигналу K1

```

/* Пример JavaScript*/
Var p1, p2, p3, k1;
{
    p1=ReadValue("p1");
    p2=ReadValue("p2");
    p3=ReadValue("p3");
    start=ReadValue("start");

    if (start == "1"){

        k1 = (!p1) && p2 || p1 && p3 || p1 && p2 && (!p3);
        WriteValue("К_Д.Б.Робот№1_Клапан ПОВОРОТ КОЛОНИ ВЛІВО",k1);
    }
}

```

### Скрипт для вихідного сигналу K2

```

/* Пример JavaScript*/
Var p1, p2, p3, k2;
{
  p1=ReadValue("p1");
  p2=ReadValue("p2");
  p3=ReadValue("p3");
  start=ReadValue("start");

  if (start == "1"){

    k2 = ((!p1) && (!p2) && p3 || p1 && (!p2) && (!p3));
    WriteValue("К_Д.Б.Робот№1_Клапан ПОВОРОТ КОЛОНИ ВПРАВО",k2);
  }
}

```

### Скрипт для вихідного сигналу К3

```

/* Пример JavaScript*/
Var p1, p2, p3, k3;
{
  p1=ReadValue("p1");
  p2=ReadValue("p2");
  p3=ReadValue("p3");
  start=ReadValue("start");

  if (start == "1"){

    k3 = (!p1) && p2 && p3 || (!p1) && (!p2) && p3;

    WriteValue("К_Д.Б.Робот№1_Клапан МГП ПРАВИЙ ЗАТИСНЕННЯ",k3);
  }
}

```

### Скрипт для вихідного сигналу К4

```

/* Пример JavaScript*/
Var p1, p2, p3, k4;
{
  p1=ReadValue("p1");
  p2=ReadValue("p2");
  p3=ReadValue("p3");
  start=ReadValue("start");

  if (start == "1"){

    //k1 = (!p1) && p2 && (!p3) || (!p1) && (!p2) && p3 || p1 && p2 && p3 || p1 && (!p2) && (!p3);
    k4= p1 && p2 && p3;
    WriteValue("К_Д.Б.Робот№1_Клапан МГП ЛІВИЙ ВПЕРЕД",k4);
  }
}

```

### Скрипт для вихідного сигналу К5

```

/* Пример JavaScript*/
Var p1, p2, p3, k5;
{
  p1=ReadValue("p1");
  p2=ReadValue("p2");
  p3=ReadValue("p3");
  start=ReadValue("start");

  if (start == "1"){

    //k1 = (!p1) && p2 && (!p3) || (!p1) && (!p2) && p3 || p1 && p2 && p3 || p1 && (!p2) && (!p3);
    k5= p1 && p2 && (!p3);
    WriteValue("К_Д.Б.Робот№1_Клапан МГП ЛІВІЙ НАЗАД",k5);
  }
}

```

### Скрипт для вихідного сигналу К6

```

/* Пример JavaScript*/
Var p1, p2, p3, k6;
{
  p1=ReadValue("p1");
  p2=ReadValue("p2");
  p3=ReadValue("p3");
  start=ReadValue("start");

  if (start == "1"){

    //k1 = (!p1) && p2 && (!p3) || (!p1) && (!p2) && p3 || p1 && p2 && p3 || p1 && (!p2) && (!p3);
    k6 = (!p1) && p2 && p3;
    WriteValue("К_Д.Б.Робот№1_Клапан МГП ПРАВИЙ ВПЕРЕД",k6);
  }
}

```

### Скрипт для вихідного сигналу К7

```

/* Пример JavaScript*/
Var p1, p2, p3, k7;
{
  p1=ReadValue("p1");
  p2=ReadValue("p2");
  p3=ReadValue("p3");
  start=ReadValue("start");

  if (start == "1"){

    //k1 = (!p1) && p2 && (!p3) || (!p1) && (!p2) && p3 || p1 && p2 && p3 || p1 && (!p2) && (!p3);
    k7= p1 && p2 && (!p3);
    WriteValue("К_Д.Б.Робот№1_Клапан МГП ПРАВИЙ НАЗАД",k7);
  }
}

```

## 12.4. Зміст звіту

- 1) Титульний аркуш.
- 2) Мета роботи, програма роботи.
- 3) Функціональна схема стенду (рисунок 12.1).
- 4) Умова роботи схем згідно варіанту.
- 5) Логічний синтез схеми керування.
- 6) Скрипти для проміжних і вихідних сигналів.
- 7) Таблиці для вхідних/вихідних сигналів.
- 8) Висновок про відповідність роботи заданим умовам.

## 12.5 Контрольні запитання

1. Що таке SCADA?
2. По якому інтерфейсу спілкується SCADA з контролером?
3. В яких режимах може працювати OPC-сервер?
4. Що таке мнемосхема?
5. Яку кількість ступенів рухомості має робот РФ-202М?

## Лабораторна робота №13

### ДОСЛІДЖЕННЯ СИСТЕМИ АВТОМАТИЗАЦІЇ НАСОСНИХ УСТАНОВОК

Тривалість лабораторної роботи – 4 години.

Тривалість домашньої роботи – 4 години.

**Мета роботи** – навчитися вирішувати задачі автоматизації насосних установок для перекачування рідин на основі дискретних давачів рівня та складати відповідні програми на мові LD для програмованого контролера Lovato.

#### 13.1. Основні теоретичні відомості

Лабораторна установка складається з трьох мембранних насосів потужністю 9 Вт, які можна за допомогою вентилів переключати у будь-яке гідравлічне з'єднання між собою, двох баків об'ємом 15 літрів з трьома дискретними датчиками рівня в кожному, програмованого логічного реле Lovato LRD20RA024 та кнопок керування. Стенд також має два імпульсні давачі продуктивності, які будуть застосовані у подальшій роботі.

Схема електричних з'єднань стенду, представлена на рис. 13.1. На схемі зображено 3 насоси M1, M2, M3; 2 кнопки керування SB1, SB2 – «Пуск» та «Стоп», та контакти дискретних датчиків SQ1-SQ6. SQ1 та SQ4 встановлені знизу баків, SQ2 та SQ5 – посередині, а SQ3 та SQ6 – зверху, причому ці датчики мають інверсний контакт. Двигуни живляться від джерела живлення постійної напруги 12В, ПЛК – від джерела живлення змінної напруги 24 В. Живлення на стенд подається вимикачем QF1.

На рисунку 13.2 зображено гідравлічну схему підключень лабораторної насосної установки з наступними умовними позначеннями: Б1 та Б2 – баки 1 та 2 відповідно; ДР1-ДР6 – датчики рівня; Н1-Н3 – насоси; ДП1, ДП2 – датчики

продуктивності. Переключення гідравліки відбувається вручну на основі вентилів К1-К11.

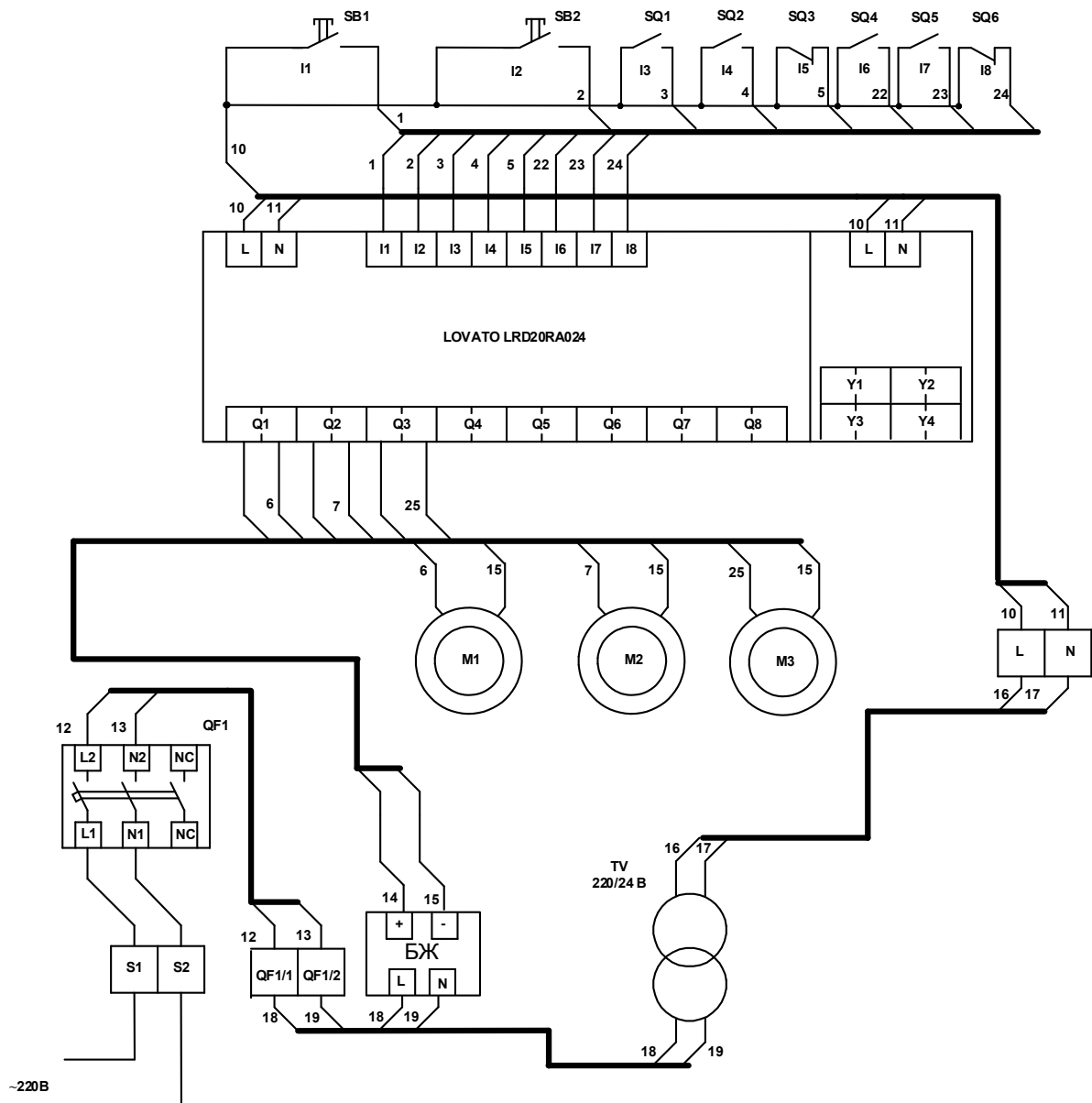


Рисунок 13.1 – Схема електричних з'єднань лабораторного стенду

Дана схема дозволяє виконувати різноманітні варіанти включення насосів в контур гідросистеми в залежності від поставленої задачі, а саме:

- окреме включення кожного насосу для перекачування води із баку 1 в бак 2 (Н1 – відкриваються К1, К6; Н2 – відкриваються К1, К3, К11; Н3 – відкриваються К2, К8);
- послідовна робота всіх трьох насосів ( відкриваються К2, К4, К7, К11);

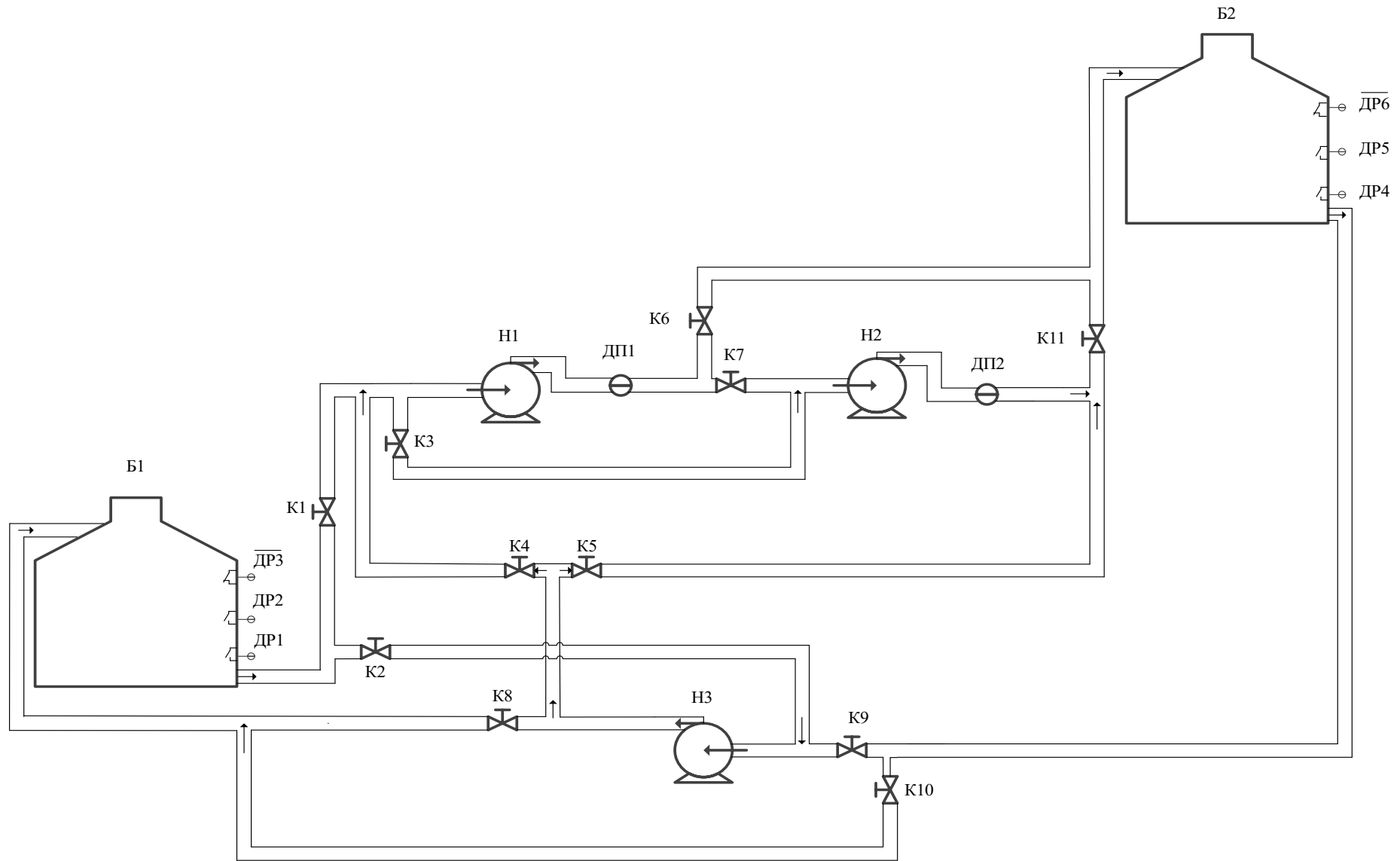


Рисунок 13.2 – Гідравлічна схема підключень наносної установки

- - паралельна робота всіх трьох насосів (відкриваються К1, К2, К3, К4, К5, К6, К11);
- паралельна робота двох насосів (відкриті К1, К3, К6, К11);
- послідовна робота двох насосів (відкриті К1, К7, К11);
- послідовно-паралельна робота насосів (відкриті К2, К3, К4, К6, К11);
- відкачування води третім насосом (відкриті К8, К9);
- відкачування води третім насосом при роботі двох насосів паралельно на закачування (відкриті К1, К3, К6, К8, К9, К11).

### 13.2 Програма роботи для лабораторної роботи

1. Виконати логічний синтез схеми керування, та отримати логічні вирази, що описують схему.
2. За отриманими алгебричними виразами у пункті 1 скласти програму на мові LD для програмованого контролера Lovato.
3. Виконати адресацію вхідних, вихідних і проміжних змінних.
4. Записати програму до контролера.
5. Перевірити відпрацювання заданих умов роботи.
6. Оформити звіт та зробити висновки.

Таблиця 13.1 – Варіанти завдань

Бригада	Номер завдання
1	1
2	2
3	3
4	4
5	2
6	3

### Задача №1

Схема виробляє сигнали  $f_1, f_2, f_3$ , які керують насосами Н1, Н2, Н3. Насоси Н1, Н2 закачують рідину в бак. В баці розміщено три датчики рівня:  $dr_4$  – датчик нижнього рівня,  $dr_5$  – датчик середнього рівня,  $dr_6$  – інверсний датчик верхнього рівня. При досягненні рідиною відповідного рівня, вихідний сигнал набуває значення логічної одиниці. Схема починає роботу після натиснення кнопки «Пуск», вмикається світловий індикатор, потім після затримки часу  $t_1=5c$ , починають вмикатися насоси. Якщо бак пустий ( $dr_4=0$ ), вмикаються одночасно насоси Н1, Н2. Після досягнення рідини датчика  $dr_5$  ( $dr_5=1$ ), насос Н2 вимикається, а насос Н1 продовжує роботу. Коли рідина досягла датчика  $dr_6$  ( $dr_6=0$ , так як він інверсний), насос Н1 вимикається. Далі після затримки часу  $t_2=10c$ , вмикається насос Н3, який відкачує рідину з бака. При подальшому зниженні рівня рідини ( $dr_4=0$ ), насос Н3 вимикається.

Після подачі сигналу на «Пуск», схема повинна працювати по заданому алгоритму, незалежно від вихідного рівня рідини в баці. Також необхідно передбачити кнопку «Аварійний стоп», яка вимикає систему, незалежно від стану в якому вона знаходиться.

### Задача №2

Схема виробляє сигнали  $f_1, f_2, f_3$ , які керують насосами Н1, Н2, Н3. Насоси Н1, Н2, Н3 закачують рідину в бак. В баці розміщено три датчики рівня:  $dr_4$  – датчик нижнього рівня,  $dr_5$  – датчик середнього рівня,  $dr_6$  – інверсний датчик верхнього рівня. При досягненні рідиною відповідного рівня, вихідний сигнал набуває значення логічної одиниці. Схема починає роботу після натиснення кнопки «Пуск1», вмикається світловий індикатор, потім після затримки часу  $t_1=5c$ , починають вмикатися насоси. Якщо бак пустий ( $dr_4=0$ ), вмикаються одночасно насоси Н1, Н2, Н3. Після досягнення рідини датчика  $dr_4$  ( $dr_4=1$ ), насос Н3 вимикається, а насоси Н1 та Н2 продовжують роботу. Після досягнення рідини датчика  $dr_5$  ( $dr_5=1$ ), насос Н2 вимикається, а насос Н1 продовжує роботу. Коли рідина досягла датчика  $dr_6$  ( $dr_6=0$ , так як він інверсний), насос Н1 вимикається. Відкриваємо кран К10 для зливу води з баку Б2, в бак Б1, після чого

натискаємо кнопку «Пуск2». В баці Б1 встановлені три датчики рівня:  $dr1$  – датчик нижнього рівня,  $dr2$  – датчик середнього рівня,  $dr3$  – інверсний датчик верхнього рівня. При досягненні рідини датчика  $dr3$  ( $dr3=0$ , так як він інверсний) та при повністю пустому баці Б2, тобто датчик  $dr4=0$ , насоси запускаються по заданому вище алгоритму.

Після подачі сигналу на «Пуск», схема повинна працювати по заданому алгоритму, незалежно від вихідного рівня рідини в баці. Також необхідно передбачити кнопку «Аварійний стоп», яка вимикає систему, незалежно від стану в якому вона знаходиться.

### **Задача №3**

Схема виробляє сигнали  $f1$ ,  $f2$ ,  $f3$ , які керують насосами Н1, Н2, Н3. Насоси Н1, Н2, одночасно закачують рідину в бак. В баці розміщено три датчики рівня:  $dr4$  – датчик нижнього рівня,  $dr5$  – датчик середнього рівня,  $dr6$  – інверсний датчик верхнього рівня. При досягненні рідиною відповідного рівня, вихідний сигнал набуває значення логічної одиниці. Схема починає роботу після натиснення кнопки «Пуск», вмикається світловий індикатор, потім після затримки часу  $t1=5c$ , починають вмикатися насоси. Якщо бак пустий ( $dr4=0$ ), вмикаються одночасно насоси Н1, Н2. Після досягнення рідини датчика  $dr5$  ( $dr5=1$ ), насос Н3 вмикається на відкачку рідини з бака Б2 в бак Б1. Коли рідина досягла датчика  $dr6$  ( $dr6=0$ , так як він інверсний), насос Н1 та Н2 вимикаються, а насос на викачку Н3 продовжує працювати, поки рідина не впаде нижче рівня датчика  $dr4$  ( $dr4=0$ ). Схема продовжує працювати в циклічному режимі по заданому алгоритму.

Після подачі сигналу на «Пуск», схема повинна працювати по заданому алгоритму, незалежно від вихідного рівня рідини в баці. Також необхідно передбачити кнопку «Аварійний стоп», яка вимикає систему, незалежно від стану в якому вона знаходиться.

### **Задача №4**

Схема виробляє сигнали  $f1$ ,  $f2$ ,  $f3$ , які керують насосами Н1, Н2, Н3. Насоси Н1, Н2, викачують рідину з бака Б1. В баці розміщено три датчики рівня:  $dr1$  – датчик нижнього рівня,  $dr2$  – датчик середнього рівня,  $dr3$  – інверсний датчик

верхнього рівня. При досягненні рідиною відповідного рівня, вихідний сигнал набуває значення логічної одиниці. Схема починає роботу після натиснення кнопки «Пуск», вмикається світловий індикатор, потім після затримки часу  $t_1=5\text{с}$ , починають вмикатися насоси. Якщо бак Б1 повний ( $dr_3=0$ , так як він інверсний), вмикаються одночасно два насоси Н1, Н2. При подальшому зниженні рівня рідини ( $dr_2=0$ ) вимикається насос Н2, одночасно з ним вмикається насос Н3 на закачку води в бак Б1. Потім після затримки часу  $t_2=5\text{с}$ , насос Н3 вимикається, а насос Н1 продовжує роботу поки рівень води в баці не впаде нижче датчика  $dr_1$  ( $dr_1=0$ ), після досягнення цього рівня насос Н1 вимикається і через час  $t_3=3\text{с}$ , вмикається насос Н3, який закачує рідину в бак до рівня датчика  $dr_3$  ( $dr_3=0$ , так як він інверсний). Далі схема працює циклічно по заданому алгоритму.

Після подачі сигналу на «Пуск», схема повинна працювати по заданому алгоритму, незалежно від вихідного рівня рідини в баці. Також необхідно передбачити кнопку «Аварійний стоп», яка вимикає систему, незалежно від стану в якому вона знаходиться.

### 13.3 Методичні вказівки до виконання роботи

Розглянемо приклад виконання роботи.

Перекачати воду з Бака№1 в Бак№2, за умови, що Бак№1 заповнений повністю. Після подачі сигналу на страт системи кнопкою Робота через час  $7\text{с}$  вмикається насос Н1, який перекачує воду в Бак№2. При спрацюванні середнього датчика рівня води вмикається Н2 паралельно до Н1, через час  $20\text{с}$  вмикається Н3 який перекачує воду в Бак№1. Коли Бак№2 буде заповнений, система вимикається. Всі спрацювання датчиків супроводжувати світловими індикаторами. Робота світлових індикаторів повинна відповідати станам, які наведені нижче:

$$HL1 = 1, \text{ коли } a = 1;$$

$$HL2 = 1, \text{ коли } b = 1;$$

$$HL3 = 1, \text{ коли } c = 0;$$

$$HL1 = 0, \text{ коли } c_1 = 0;$$

$$HL2 = 0, \text{ коли } b_1 = 1;$$

$$HL3 = 0, \text{ коли } a_1 = 1.$$

**Умовні позначення:**

У контролері використовуються такі позначення:

I – вхідні сигнали;

Q—вихідні сигнали;

M – проміжна змінна, комірка пам'яті;

T – затримка часу;

“ - “ – контактне з'єднання.

**Всі входи і виходи підключені так, як наведено нижче:**

I1, I2, I3 – датчики рівня води Бака№1;

I4, I5, I6 – датчики рівня води Бака№2;

Q4, Q5, Q6 – увімкнення насосів №1, №2, №3;

I7 – кнопка "Робота";

I8 – кнопка "Стоп".

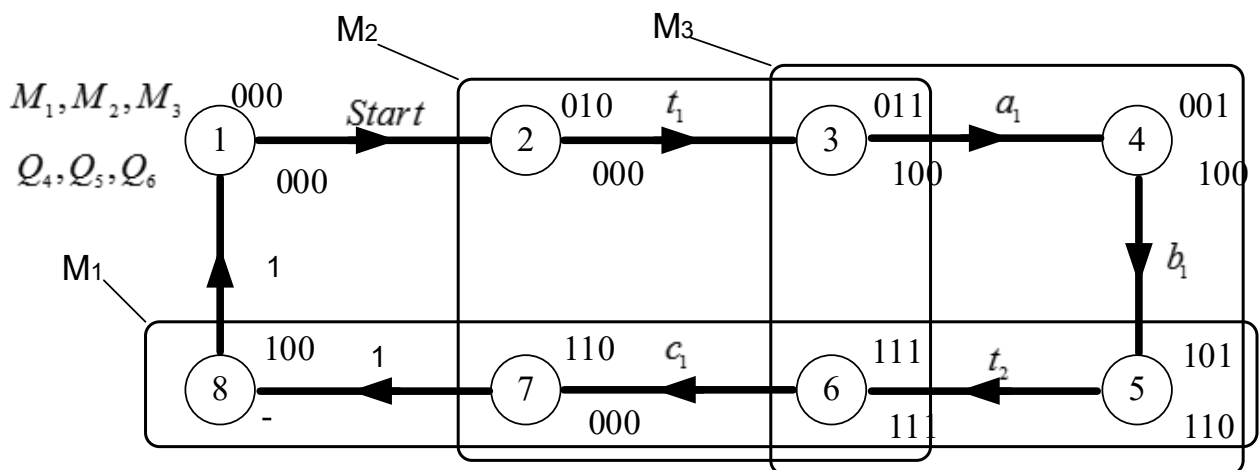


Рисунок 13.3 – Графоперехід на основі якого був зроблений синтез даної програми

**Запишемо рівняння:**

Умови встановлення тригерів

$$S_{M1} = I_5 m_2 M_3;$$

$$S_{M2} = I_7 m_1 m_3 + T_2 M_1 M_3;$$

$$S_{M3} = T_1 m_1 M_2;$$

Умови скидання тригерів

$$R_{M1} = m_2 m_3;$$

$$R_{M2} = I_4 m_1 M_3 + M_1 m_3;$$

$$R_{M3} = I_6 M_1 M_2.$$

Рівняння для таймерів:

$$T_1 = m_1 M_2 m_3;$$

$$T_2 = M_1 m_2 M_3.$$

Рівняння для вихідних сигналів:

$$Q_4 = M_3;$$

$$Q_5 = M_1 M_3;$$

$$Q_6 = M_1 M_2 M_3.$$

Запишемо дані рівняння в ПЛК:

I5 — m2 — M3 — ↑M1

— — m2 — m3 — ↓M1

I7 — m1 — m3 Т ↑M2

T2 — M1 — M3 Ј

I4 — m1 — M3 Т ↓M2

M1 — m3 — — Ј

T1 — m1 — M2 — ↑M3

I6 — M1 — M2 — ↓M3

M3 — — — — (Q4

M1 — M3 — — — (Q5

M1 — M2 — M3 — (Q6

m1 — M2 — m3 — (T1

M1 — m2 — M3 — (T2

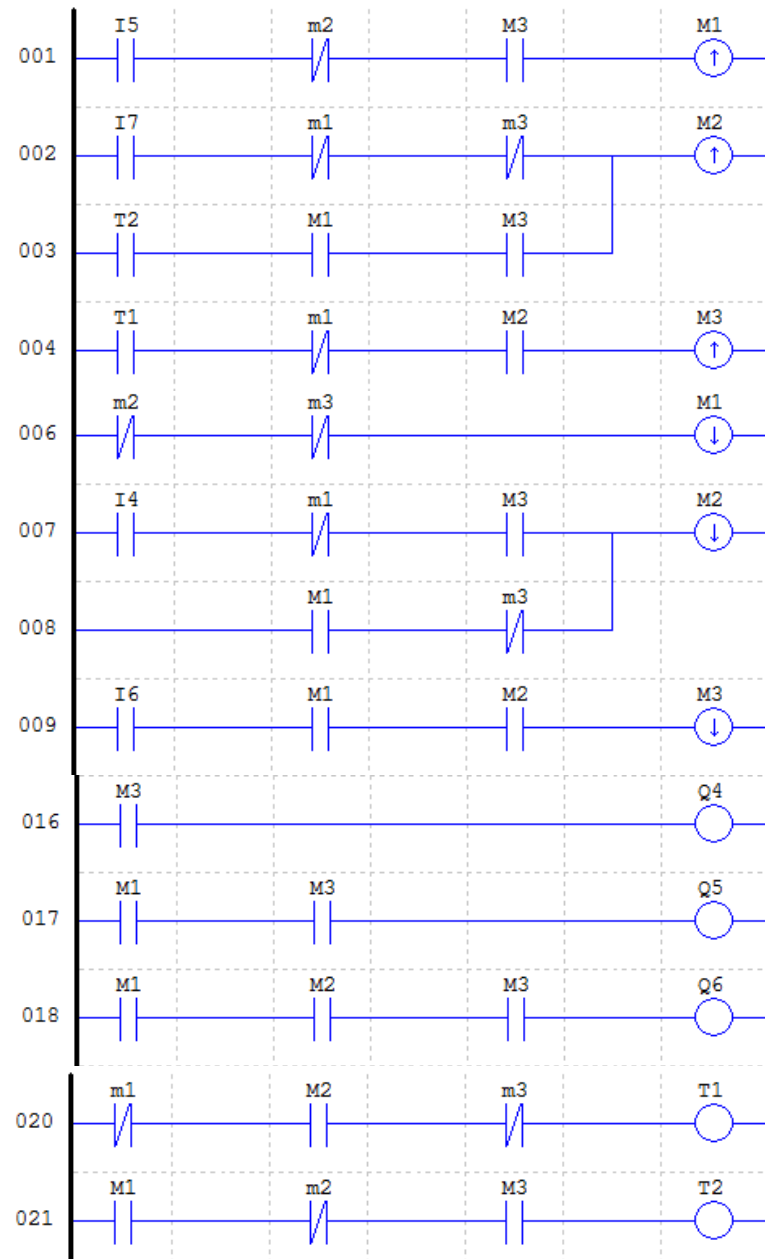


Рисунок 13.4 – Програма роботи прикладу на мові LD

### 13.4 Зміст звіту

- 1) Титульний аркуш.
- 2) Мета роботи, програма роботи.
- 3) Опис та схема лабораторного стенду.
- 4) Умови роботи схем.
- 5) Логічний синтез схем керування.
- 6) Адресування входів, виходів, таймерів, комірок пам'яті.
- 7) Програма на мові LD.
- 8) Висновок про відповідність роботи заданим умовам.

### 13.5 Контрольні запитання

1. Яке живлення двигунів на лабораторному стенді?
2. Чим відрізняють виходи контролера Q1-Q8 від виходів Y1-Y4?
3. Які умовні позначення використовуються у контролері?
4. Яка максимальна кількість послідовноз'єднаних контактів можлива при програмуванні?
5. Як позначається Set та Reset котушки?

## Лабораторна робота №14

### СИНТЕЗ ТА ДОСЛІДЖЕННЯ ГЕНЕРАТОРІВ ДЕСЯТКОВИХ ЧИСЕЛ НА ОСНОВІ ПРОГРАМОВАНОГО ЛОГІЧНОГО КОНТРОЛЕРА НІТАСНІ-A23DRP ТА СВІТЛОДІОДНОГО ІНДИКАТОРА

Тривалість лабораторної роботи – 4 години.

Тривалість домашньої роботи – 4 години.

**Мета роботи** – навчитися синтезувати логічні вирази для генераторів десяткових чисел методом RS-тригерів, розробити на основі цих виразів програму для програмованого логічного контролера та перевірити роботу генератора.

#### 14.1. Основні теоретичні відомості

Тригер – це логічна схема зі зворотними зв'язками, яка має релейну характеристику керування і може перебувати в одному з двох усталених станів, забезпечуваних цими зв'язками. Тригер характеризується властивістю зберігати інформацію, тому, використовуючи вихідні сигнали тригерів як проміжні змінні, можна будувати багатотактні схеми.

За способом запису інформації тригери поділяють на несинхронізовані (асинхронні) та синхронізовані (синхронні). Багатотактні схеми за принципом дії також поділяють на асинхронні та синхронні. Для побудови асинхронних схем застосовують несинхронізовані RS-тригери. Синхронні схеми звичайно будують на синхронізованих JK-тригерах.

Для синтезу багатотактної схеми на тригерах умови її роботи подають у вигляді графопереходів.

*Графоперехід* – це графічне зображення послідовності роботи багатотактної схеми. Елементами графу є *вершини* і *ребра*. Вершини відповідають станам схеми

і позначаються кружками. Ребра – це лінії зі стрілками, що з'єднують вершини і показують напрям переходу з одного стану схеми в інший.

Кількість вершин графу при синтезі асинхронних схем на  $RS$ -тригерах визначають з умови  $2^n \geq S$ , де  $S$  – кількість станів схеми;  $2^n$  – кількість вершин графу;  $n$  – кількість тригерів. Вершини графу рекомендується розміщувати так, щоб коли  $n = 2$ , вони створювали конфігурацію  $2 \times 2$ , якщо  $n = 3$ , – конфігурацію  $4 \times 2$ , а якщо  $n = 4$ , – конфігурацію  $4 \times 4$ . Вигляд графопереходу переходів для  $n = 3$  зображено на рис. 14.1.

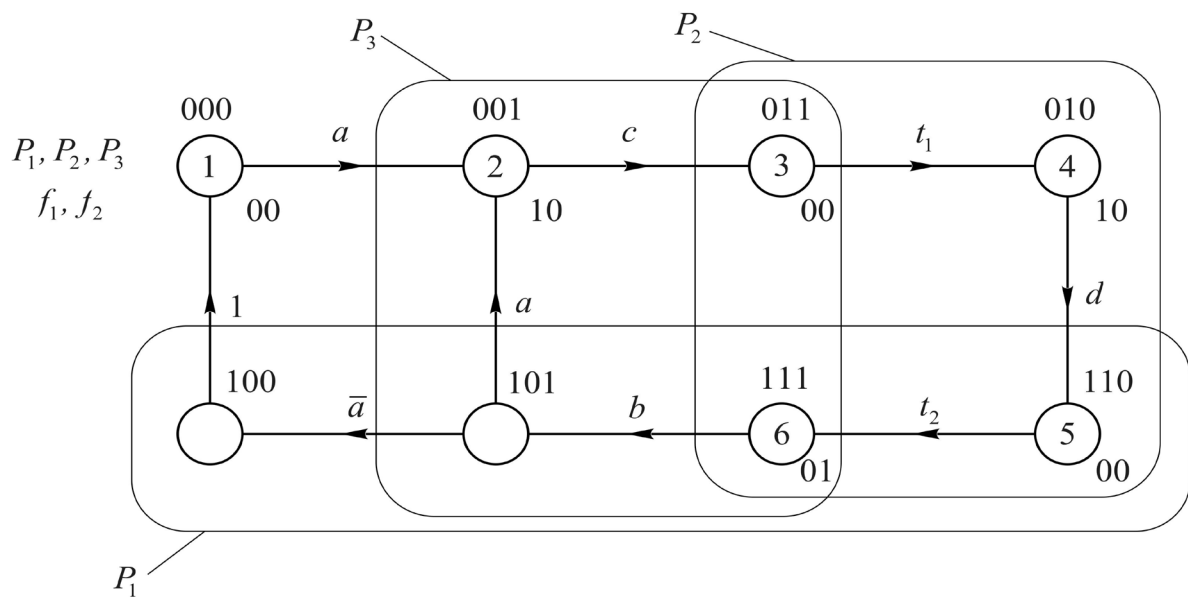


Рисунок 14.1 – Графоперехід для  $n = 3$

Вихідні сигнали тригерів виконують роль проміжних змінних, тому тригери та їх вихідні сигнали позначають буквами  $P_1, P_2, P_3$ . Ці позначення записують зверху ліворуч від графу.

Кожна вершина графу кодується набором значень вихідних сигналів тригерів. Коди обирають так, щоб для сусідніх вершин вони розрізнялися значенням тільки однієї змінної. У вихідному стані схеми (стан очікування) зазвичай припускають, що всі проміжні змінні дорівнюють нулеві. Значення проміжних змінних для кожної вершини записують над кружками у тій черговості, у якій записано позначення тригерів.

Всередині вершин записують номери станів схеми. Стани, між якими має відбуватися перехід згідно з умовами роботи схеми, розміщуються в сусідніх вершинах, якщо це можливо. Під вершинами з номерами станів записують значення вихідних змінних, що відповідають цим станам. Позначення вихідних змінних  $f_1, f_2, \dots, f_n$  записують ліворуч від графу під позначеннями тригерів.

Вершини, між якими повинні відбуватися переходи, з'єднують ребрами зі стрілками. Над стрілками або праворуч від них, якщо ребра спрямовано вертикально, записують позначення вхідних сигналів, що спричиняють ці переходи.

У схемах на асинхронних *RS*-тригерах переходи можна робити тільки між сусідніми вершинами. Якщо ця умова не виконується, то необхідно передбачити природні переходи (за рахунок подавання вхідного сигналу 1) через проміжні нестійкі стани.

Розглянемо методику побудови схеми генератора чисел на основі *RS*-тригерів на такому прикладі.

*Приклад 1.* Виконати синтез генератора наступної послідовності чисел: 1, 4, 7, 1, 4, 7, 1... із затримкою часу між числами 1 секунда. При натисненні кнопки «Пуск» генератор починає роботу, при натисненні кнопки «Стоп» схема вимикається. Передбачити кнопку «Пауза», при натисненні якої схема зупиняє генерацію на тому числі, яке висвітилося на генераторі; при відпусканні кнопки «Пауза» генератор продовжує свою роботу.

Синтез генератора будемо робити виходячи з того, що виходи схеми підключаються до світлодіодів семисегментного індикатора, як показано на рис. 14.2.

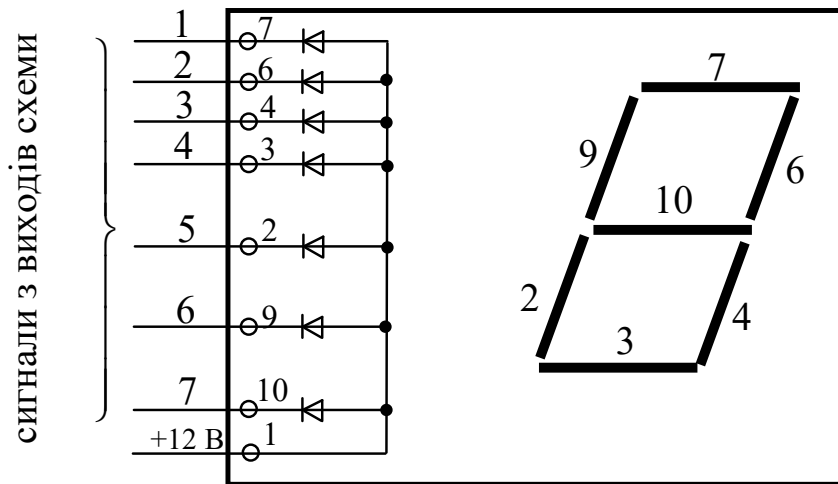


Рисунок 14.2 – Підключення виходів схеми до світлодіодного індикатора

Зробимо такі позначення вхідних і вихідних сигналів, а також сигналів таймерів, які потрібно розглядати як вхідні сигнали для графу переходів. Вхідні сигнали:  $a$  – команда «Пуск»;  $b$  – команда «Стоп»;  $c$  – команда «Пауза»;  $t_1, t_2, t_3$  – сигнали таймерів, кожний з яких забезпечує затримки часу по 1 секунді. Вихідні сигнали:  $f_1, f_2, f_3, f_4, f_5, f_6, f_7$  – сигнали, що забезпечують вмикання окремих світлодіодів, відповідно до рис. 14.2.

Побудову графу переходів починаємо з визначення кількості станів, у яких може перебувати схема автоматичного керування. Таких станів чотири: 1 – вихідний стан; 2 – світиться число «1» протягом часу 1 секунда; 3 – світиться число «4» протягом часу 1 секунда; 4 – світиться число «7» протягом часу 1 секунда.

Виходячи з кількості станів схеми  $4 < 2^3$ , визначаємо кількість тригерів  $n = 3$  і кількість вершин графу переходів  $2^3 = 8$ . Позначаємо проміжні змінні  $P_1, P_2, P_3$ , будуємо вісім вершин графу і кодуємо їх комбінаціями значень проміжних змінних (рис. 14.3).

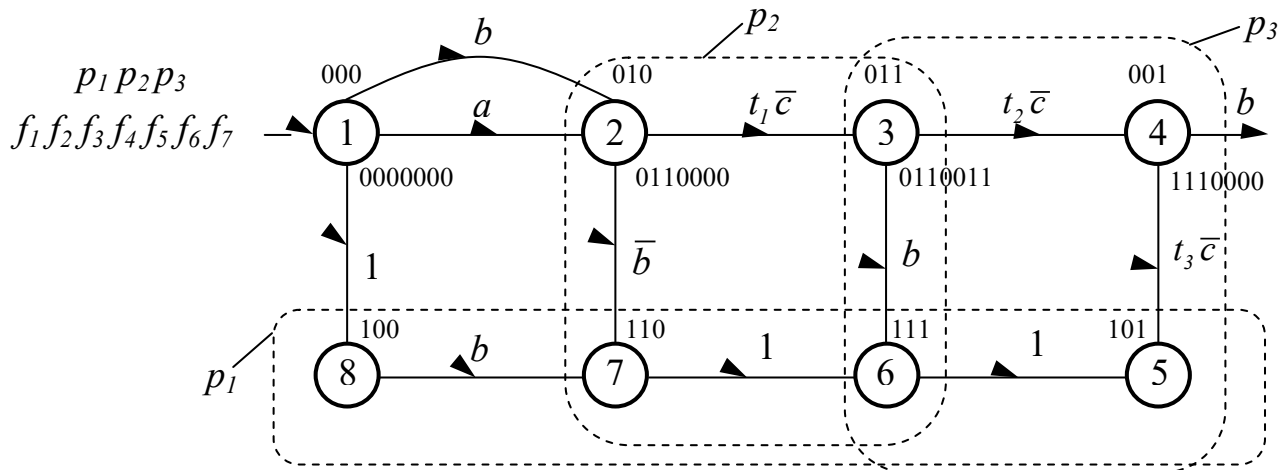


Рисунок 14.3 – Графоперехід до прикладу

У сусідніх вершинах розміщуємо стани, між якими потрібен перехід згідно з умовами роботи схеми. Номером 1 позначимо стан очікування ( $a = 0, b = 0, c = 0$ ). Із надходженням сигналу  $a$  відбувається перехід у стан 2 ( $f_2 = 1, f_3 = 1$ ) і світиться цифра «1». Зі стану 2 з надходженням сигналу  $b$  схема повертається у вихідний стан, а при надходженні сигналу таймера  $t_1$  за відсутності сигналу паузи  $c$  схема переходить у стан 3 ( $f_2 = 1, f_3 = 1, f_6 = 1, f_7 = 1$ ) і світиться цифра «4». Зі стану 3 можливе повернення в стан 1 у разі надходження сигналу  $b$  або перехід у стан 4 ( $f_1 = 1, f_2 = 1, f_3 = 1$ ) з надходженням сигналу таймера  $t_2$  за відсутності сигналу паузи  $c$ . У цьому стані світиться цифра «7». Зі стану 4 також можливі два переходи: повернення в стан 1 у разі надходження сигналу  $b$  або перехід у стан 2 при надходженні сигналу  $t_3$ . Вершини 3 і 1 не є сусідніми. Тому перехід зі стану 3 у стан 1 здійснюється за рахунок природних переходів через нестійкі стани 6, 7 та 8. За безпосереднього переходу зі стану 3 у стан 1 змінюються одночасно три проміжні змінні. Це є неприпустимим через те, що це може спричинювати змагання в колах проміжних змінних. За рахунок уведення природних переходів забезпечується змінювання проміжних змінних по черзі: спочатку  $P_1$  набуває значення 1, потім  $P_3 = 0$ , потім  $P_2 = 0$ , і нарешті  $P_1 = 0$ . Таким чином схема переходить у стан 1 ( $P_1 = 0; P_2 = 0; P_3 = 0$ ). Всі переходи, необхідні за умовами роботи схеми, здійснено. На цьому побудова графу переходів завершується.

Синтез схем на  $RS$ -тригерах виконують у такій послідовності.

1. За заданими умовами роботи схеми будують граф переходів.

2. Записують умови вмикання і скидання кожного тригера. Для цього охоплюють замкнутою лінією всі стани на графі переходів, у яких значення тієї ж самої проміжної змінної (вихідного сигналу тригера) дорівнює одиниці. Вхідні сигнали схеми, позначення яких розміщено на ребрах, що заходять в одержану замкнену область, встановлюють тригер у стан 1, а вхідні сигнали на ребрах, що виходять з цієї області, скидають тригер у стан 0.

Умови вмикання тригера записують у вигляді добутку сигналу на ребрі, що заходить в область, і сигналів решти тригерів, стан яких не змінюється під час переходу, позначеному ребром. Наприклад, якщо сигнал на ребрі, що заходить в область з одиничним значенням вихідного сигналу тригера  $P_1$ , дорівнює  $a$ , а тригери  $P_2$  і  $P_3$  не перемикаються, а зберігають стани  $P_2 = 1$ ,  $P_3 = 0$ , то умову вмикання тригера  $P_1$  записують у вигляді

$$S_{P_1} = ap_2\bar{p}_3.$$

Якщо в замкнену область входить кілька ребер, то умову вмикання тригера записують у вигляді суми добутків відповідних сигналів, складених для кожного ребра.

Умову скидання тригера записують аналогічно для кожного ребра, що виходить з цієї області, і подають у вигляді формули  $R_{P_i}$ . Уведення сигналів решти тригерів в умови вмикання і скидання цього тригера гарантує здійснення переходів у потрібній послідовності.

Описану процедуру виконують для кожного тригера і визначають для них умови вмикання і скидання.

3. Записують вирази для вихідних сигналів схеми у вигляді комбінаційних функцій вихідних сигналів тригерів.

4. Якщо умовами роботи схеми передбачено технологічні затримки, то складають вирази для вхідних сигналів таймерів як комбінаційні функції вихідних сигналів тригерів.

5. За отриманими логічними формулами складають схему.

Проілюструємо викладену методику синтезу для побудованого графу переходів.

Охоплюємо замкнутою лінією ті стани на графі переходів, для яких  $P_1 = 1$ . У замкнену область заходить два ребра із сигналами  $t_3 \bar{c}$  та  $b$ . На першому переході  $P_2 = 0, P_3 = 1$ , а на другому  $P_2 = 1, P_3 = 1$ . Тому умова вмикання тригера  $P_1$

$$S_{P_1} = t_3 \bar{c} \bar{p}_2 p_3 + b p_2 p_3.$$

Із замкненої області виходять два ребра – із сигналами  $\bar{b}$  та  $1$ , причому для першого переходу  $P_2 = 1, P_3 = 0$ , для другого –  $P_2 = 0, P_3 = 0$ . Тому умову скидання тригера записують у вигляді суми добутку  $\bar{b} p_2 \bar{p}_3$  для першого переходу і  $\bar{p}_2 \bar{p}_3$  – для другого (сигнал  $1$  у добутку  $\bar{p}_2 \bar{p}_3$  можна не записувати). Отже,

$$R_{P_1} = \bar{b} p_2 \bar{p}_3 + \bar{p}_2 \bar{p}_3 = \bar{b} \bar{p}_3 + \bar{p}_2 \bar{p}_3.$$

Охоплюємо тепер замкненою лінією стани, для яких  $P_2 = 1$ . В область заходить два ребра із сигналами  $a$  та  $1$ . На першому переході  $P_1 = 0$  та  $P_3 = 0$ , а на другому  $P_1 = 1$  та  $P_3 = 1$ , тому

$$S_{P_2} = a \bar{p}_1 \bar{p}_3 + p_1 p_3.$$

Умову скидання тригера визначено сигналами  $t_2 \bar{c}$  та двома ребрами с сигналом  $b$ , при цьому у першому випадку  $P_1 = 0, P_3 = 1$ , у другому  $P_1 = 1, P_3 = 0$ , а у третьому  $P_1 = 0, P_3 = 0$ , отже,

$$R_{P_2} = t_2 \bar{c} \bar{p}_1 p_3 + b p_1 \bar{p}_3 + b \bar{p}_1 \bar{p}_3 = t_2 \bar{c} \bar{p}_1 p_3 + b \bar{p}_3.$$

Нарешті охоплюємо замкненою лінією стани, для яких  $P_3 = 1$ . В отриману область заходить одне ребро із сигналом  $t_1 \bar{c}$  і виходять два ребра із сигналами  $b$  та  $1$ . Отже, з урахуванням станів тригерів  $P_1$  і  $P_2$ , отримаємо

$$S_{P_3} = t_1 \bar{c} \bar{p}_1 p_2;$$

$$R_{P_3} = b \bar{p}_1 \bar{p}_2 + p_1 p_2.$$

Формули для вихідних сигналів  $f_1 \dots f_7$  записуємо як комбінаційні функції вихідних сигналів тригерів  $P_1, P_2, P_3$ , причому для реалізації заданої послідовності чисел функції  $f_4$  та  $f_5$  не використовуються, тобто вирази для них складати не треба. Дійсно, функція  $f_1 = 1$  у стані  $4$ , тобто в стані, якому

відповідають такі комбінації значень вихідних сигналів тригерів:  $p_1 = 0$ ,  $p_2 = 0$ ,  $p_3 = 1$ . Тому

$$f_1 = \bar{p}_1 \bar{p}_2 p_3.$$

Аналогічно, для функцій  $f_2, f_3, f_6$  та  $f_7$

$$f_2 = \bar{p}_1 p_2 \bar{p}_3 + \bar{p}_1 p_2 p_3 + \bar{p}_1 \bar{p}_2 p_3 = \bar{p}_1 p_2 + \bar{p}_1 p_3;$$

$$f_3 = \bar{p}_1 p_2 \bar{p}_3 + \bar{p}_1 p_2 p_3 + \bar{p}_1 \bar{p}_2 p_3 = \bar{p}_1 p_2 + \bar{p}_1 p_3 = f_2;$$

$$f_6 = \bar{p}_1 p_2 p_3;$$

$$f_7 = \bar{p}_1 p_2 p_3 = f_6.$$

Таймер  $T_1$  вмикається у стані 2 ( $p_1 p_2 p_3 = 010$ ), таймер  $T_2$  – у стані 3 ( $p_1 p_2 p_3 = 011$ ), а таймер  $T_3$  у стані 4 ( $p_1 p_2 p_3 = 001$ ) тому

$$T_1 = \bar{p}_1 p_2 \bar{p}_3;$$

$$T_2 = \bar{p}_1 p_2 p_3;$$

$$T_3 = \bar{p}_1 \bar{p}_2 p_3.$$

На основі отриманих логічних виразів необхідно скласти програму на мові LD для подальшого програмування логічного контролера Hitachi-A23DRP.

Складання та відлагодження робочої програми відбувається в середовищі *Pro-H*, та використовуючи сучасну мову програмування для логічних програмованих контролерів (програмованих реле) – *LD (Ladder Diagram)*.

*LD* (скорочення від *Ladder Diagram*) – мова релейної логіки. Призначена для програмування промислових контролерів (ПЛК). Синтаксис мови зручний для заміни логічних схем, виконаних на релейній техніці. Орієнтований на інженерів з автоматизації, які працюють на промислових підприємствах. Забезпечує наочний інтерфейс логіки роботи контролера, який полегшує не лише завдання для програмування і введення в експлуатацію, але і швидкий пошук неполадок в підключається до контролера обладнанні.

Програма мовою релейного логіки має наочний і інтуїтивно зрозумілий інженерам-електрикам графічний інтерфейс, який представляє логічні операції, як електричний ланцюг із замкнутими і розімкненими контактами. Протікання або відсутність струму в цьому ланцюзі відповідає результату логічної операції (*true* – якщо струм тече; *false* – якщо струм не тече).

Основними елементами мови є контакти, які можна образно уподібнити парі контактів реле або кнопки. Пара контактів ототожнюється з логічної змінної, а стан цієї пари – зі значенням змінної.

Розрізняються нормально замкнуті і нормально розімкнуті контактні елементи, які можна зіставити з нормально замкнутими і нормально розімкнутими кнопками в електричних ланцюгах.

$\neg \mid \neg$  – Нормально розімкнений контакт розімкнений при значенні *false*, призначеної йому змінної і замикається при значенні *true*.

$\neg \mid \mid$  – Нормально замкнутий контакт, навпаки, замкнутий, якщо змінна має значення *false*, і розімкнений, якщо змінна має значення *true*.

$\neg ( ) \neg$  – Підсумок логічного кола копіюється в цільову змінну, яка називається котушка (англ. *coil*). Це слово має узагальнений образ виконавчого пристрою, тому в російськомовній документації зазвичай говорять про вихід кола, хоча можна зустріти і приватні значення терміна, наприклад котушка реле.

**Створення нового проекту.** Даний пункт призначений для ознайомлення з середовищем програмування *Pro-H*, основами програмування Hitachi ПЛК і містить опис послідовності дій для програмування ПЛК. Приклад розроблений на базі Hitachi EH-A23DPR і дозволяє перевірити правильність роботи програми на самому контролері.

*Створення проекту програми.* При створенні проекту використовується мова релейних діаграм *LD (Ladder Diagram)*, який реалізує структури, подібні електричним ланцюгам в комутаційній автоматичі.

Користувач запускає *CoDeSys* послідовним вибором додатків:

Пуск ► Усі програми ► *Pro-H* ► *Pro-H*.

Новий проект відкривається з головного меню: *File* ► *New*. У вікні (рис 14.4) вибирається тип контролера, MICRO\_23, вибір підтверджується натисканням клавіші ОК.

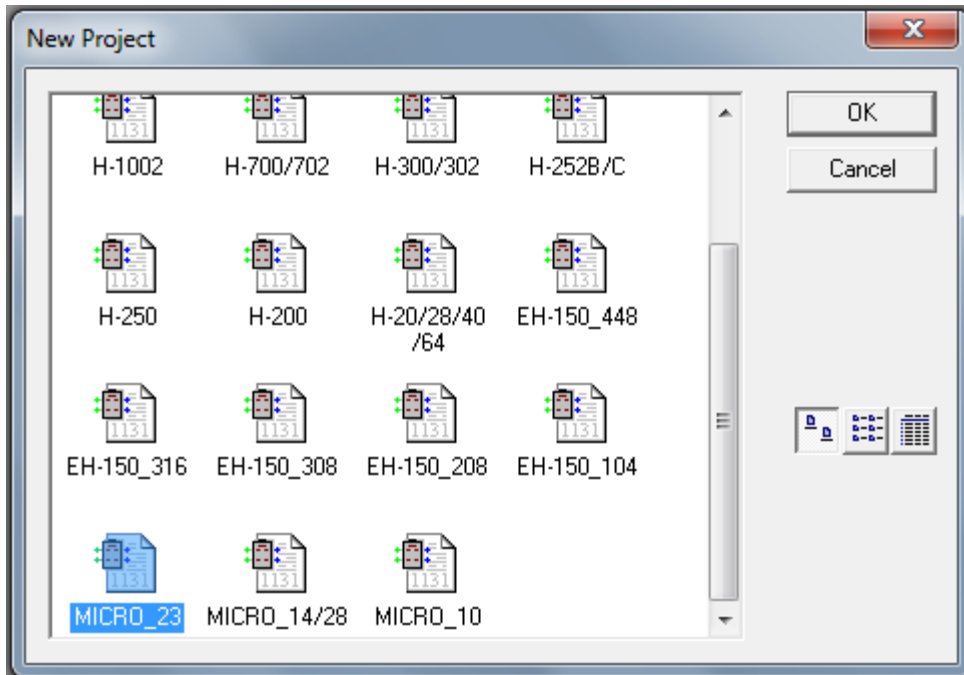


Рисунок 14.4 – Вибір типу контролера у вікні New Project.

Наступним етапом є створення файлу завдання і файлу програми.

Для цього необхідно натиснути правою кнопкою мишу на папку Tasks і вибрати пункт Insert, як це показано на рис. 14.5:

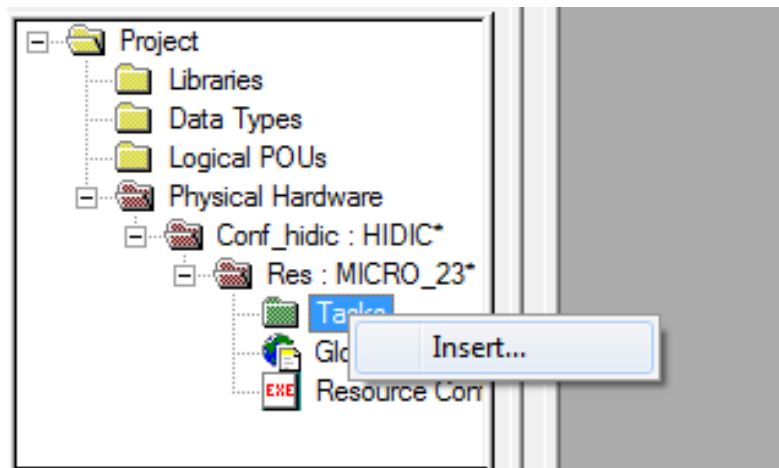


Рисунок 14.5 – Створення файлу завдання

Після цього на екрані з'явиться вікно в якому необхідно ввести ім'я та натиснути OK (рис. 14.6). Тип файлу завдання так і залишити NORMAL.

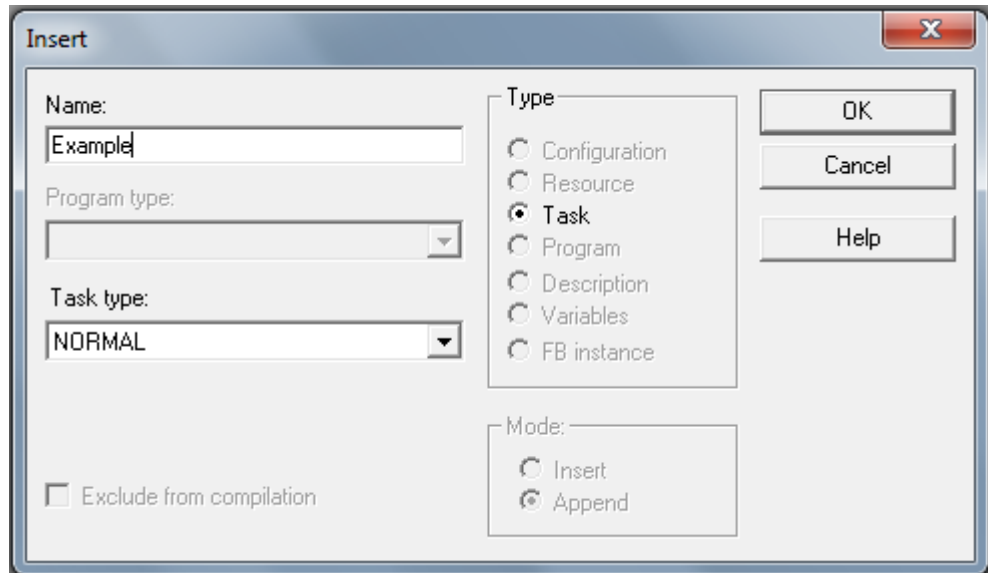



Рисунок 14.6 – Вікно Insert

Наступним етапом є створення вікна для написання програми. Щоб це зробити необхідно повторити таку послідовність дій:

- зверху зліва натискаємо кнопку Add Program  ;
- у вікні, що з'явилося (рис. 14.7) необхідно вказати ім'я, та обрати в Type►Program та Language►LD і натиснути ОК;
- на раніше створеному файлі завдання (Task) (рис. 14.8) натискаємо правою кнопкою мишу і обираємо Insert. З'являється вже знайоме нам вікно (рис. 14.9.). Необхідно вказати ім'я, та обрати в Type►Program і натиснути ОК.

На даному етапі від вікна написання програми нас відділяє один крок. Залишилося просто його вибрати. Для цього потрібно двічі натиснути на файл Program створений кнопкою Add Program (рис. 14.7).

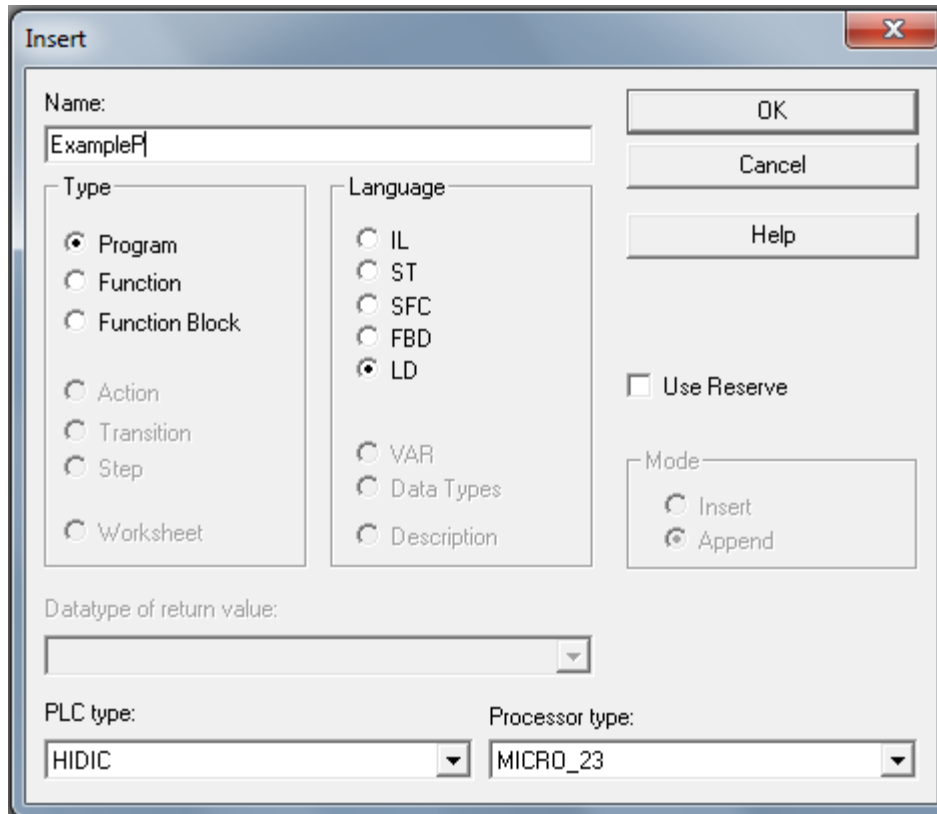


Рисунок 14.7 – Вікно створення програми

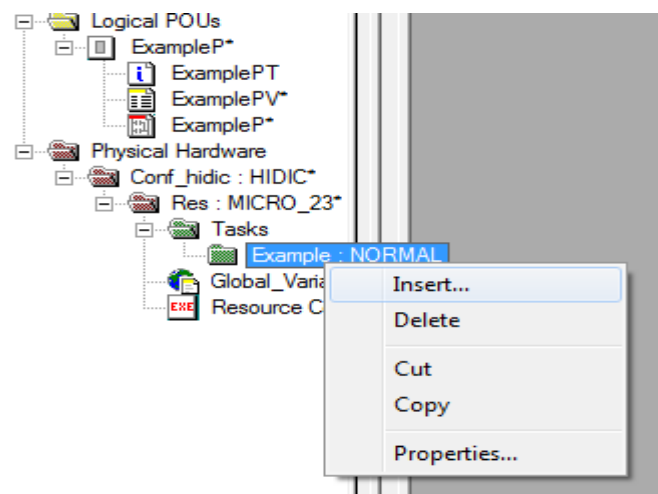


Рисунок 14.8 – Вікно меню

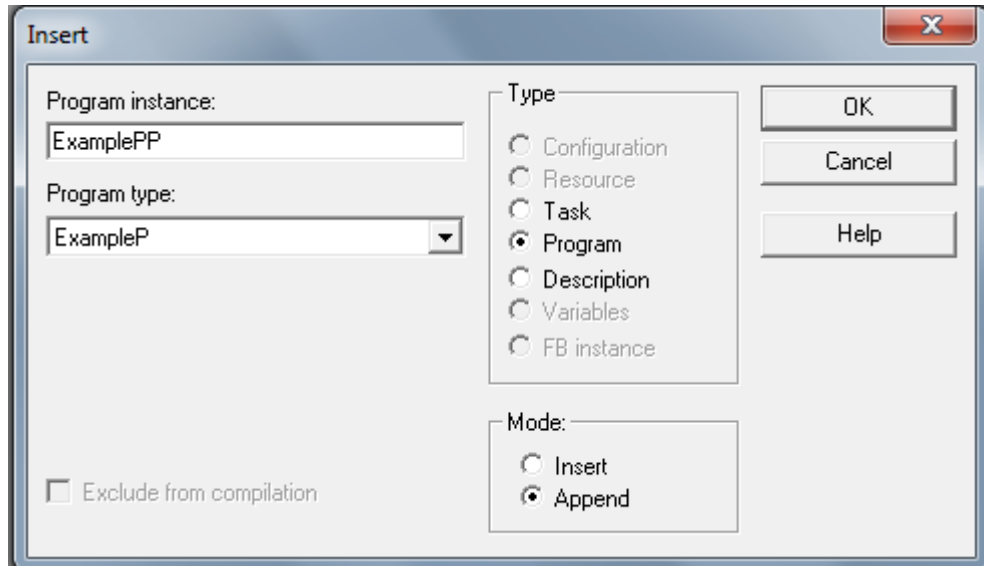


Рисунок 14.9 – Вікно Insert

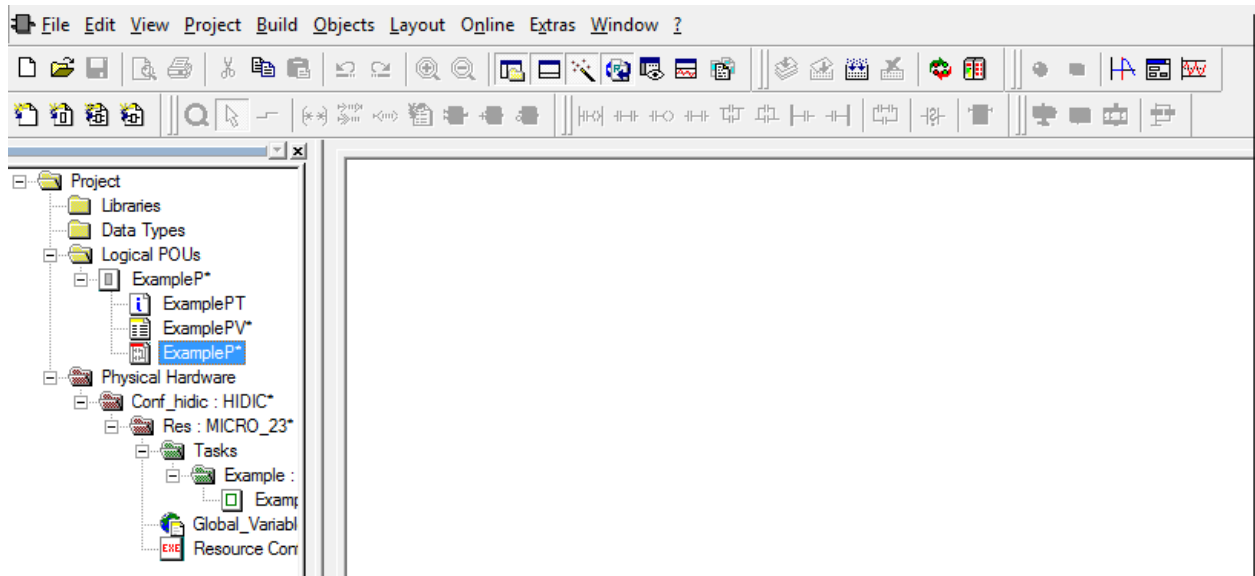


Рисунок 14.10 – Робоче поле

Тепер на робочому полі (рис. 14.10) складаємо контактно-релейну схему відповідно складених рівнянь (програми). У верхньому меню вікна вибираємо тип об'єкт де в третій частині меню є дев'ять об'єктів за допомогою яких складається схема. Основні це Contact Network, Contact Right та Coil Right, які створюють на робочому полі відповідно контактну вітку, контакт або котушку. Всі ці об'єкти продубльовані кнопками на робочій області



Розглянемо створення схеми для першого тригера з графопереходу рис. 14.3, яка представлена на рис. 14.11:

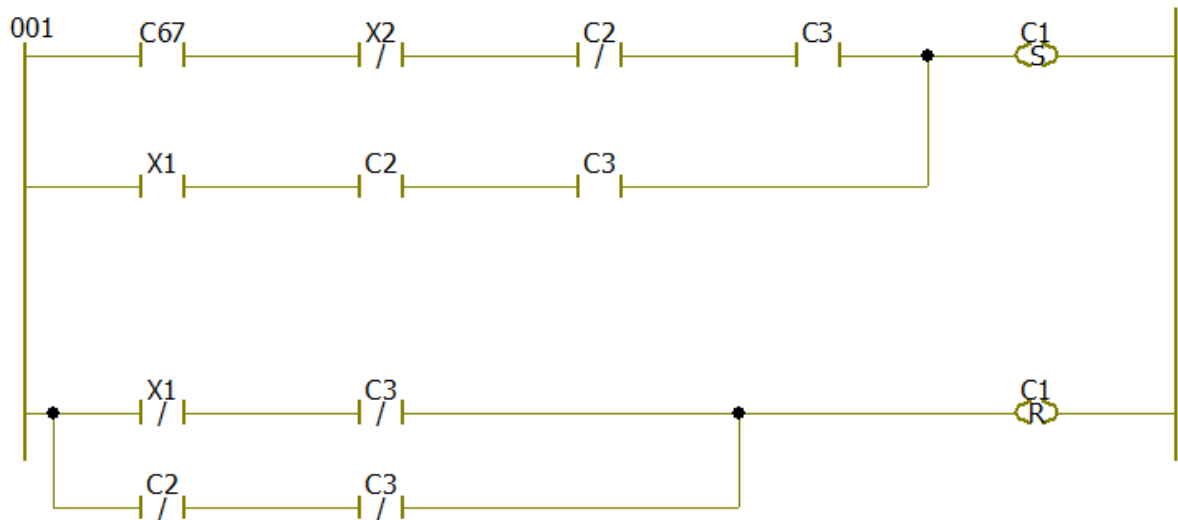


Рисунок 14.11 – Програма на мові LD для першого тригера

Всі проміжні змінні тригера та таймерів позначаються буквами С, наприклад С67– таймер  $t_1$ , С1– тригер  $P_1$ . Вхідні та вихідні змінні доцільно позначати  $X\#$  та  $Y\#\#\#$  відповідно, наприклад  $X1$ ,  $Y101$ . Кожний елемент має меню яке викликається подвійним натисканням лівої клавіші миші (рис. 14.12) :

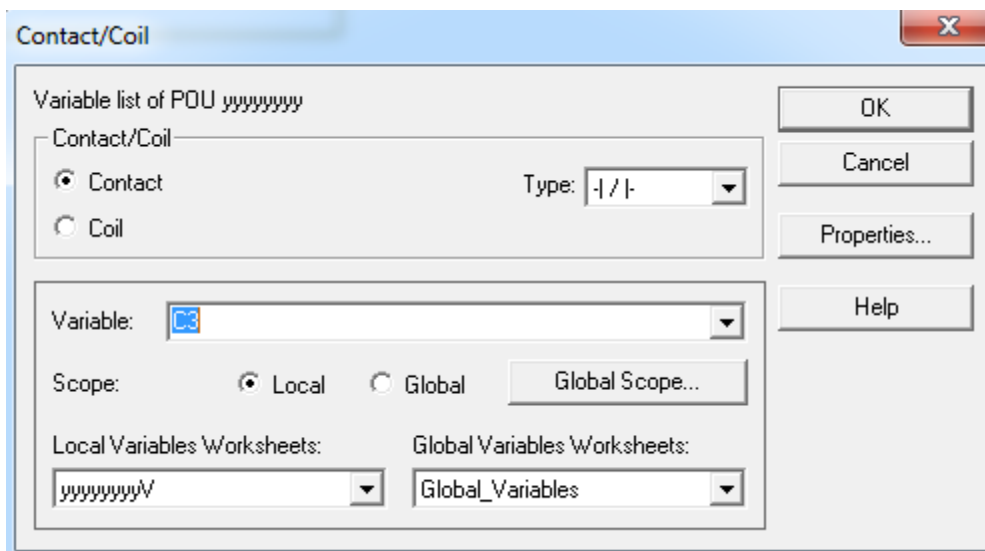


Рисунок 14.12 – Вікно налаштування змінних

В пункті Variable прописуємо назву контакту/котушки. Також обираємо тип, для контактів це:

- | | - контакт;
- |/| - контакт з інверсією;

Для котушки:

- ( ) - котушка;

-(/)- котушка з інверсією;

-(S)- котушка SET;

-(R)- котушка RESET;

Кнопка Properties викликає ще одне діалогове вікно (рис. 14.13):

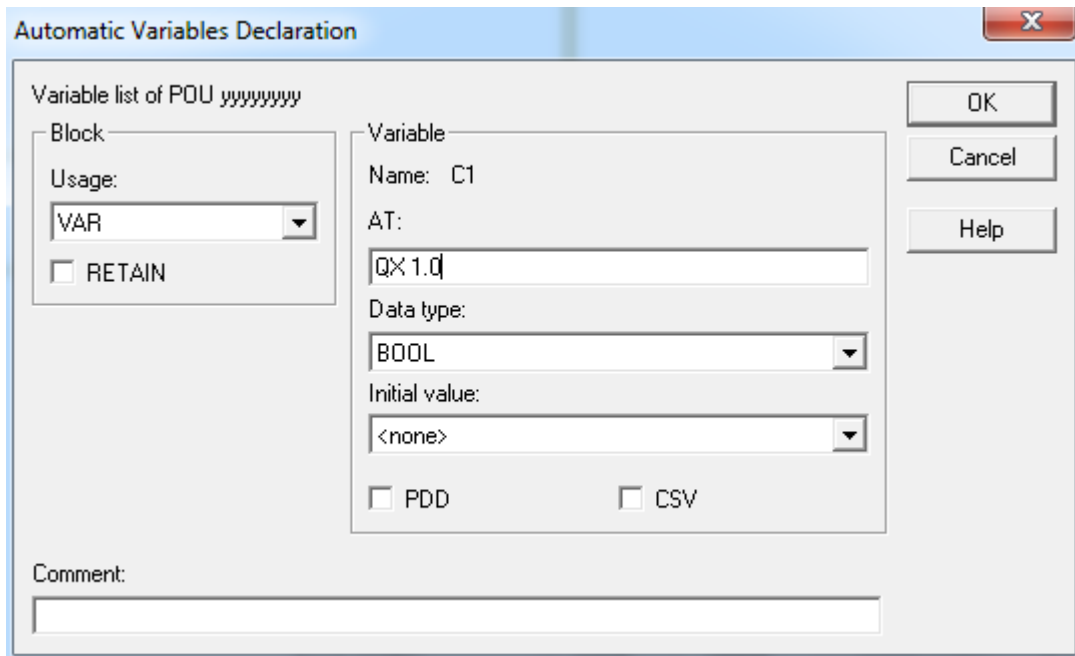


Рисунок 14.13 – Адресація змінних












В полі AT записуємо тип змінної:

- IX–вхід;
- QX– вихід;

Наприклад:

Вихідна змінна	Позначення
X1	IX 0.0
X0	IX 0.1
Y101	QX 1.1
Y102	QX 1.2
Y103	QX 1.3
Y106	QX 1.6

Далі через пробіл вказується номер порту контролера звідки/куди подається сигнал. Для проміжних змінних це поле залишаємо пустим. Кожний наступний тригер а також рівняння для вихідних змінних доцільно створювати в окремій контактній вітці.

Розглянемо створення вітки для таймера. В діалоговому вікні кнопки Insert function or FB  із меню кнопок           обираємо функціональний блок типу TD\_1, TD\_01 або TD\_001 в залежності від необхідної дискрети часу (1, 0.1 та 0.01 секунди відповідно). Номер таймера по порядку обирається у вкладці Instance (рис. 14.14):

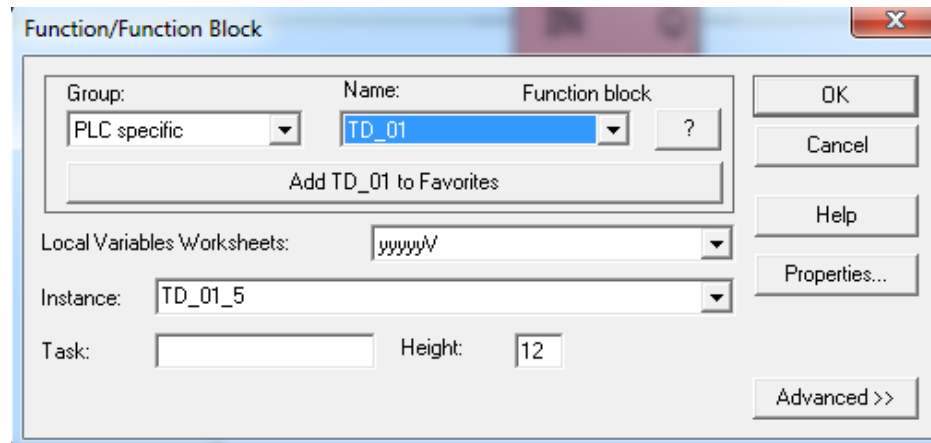


Рисунок 14.14 – Налаштування таймерів

Схема для рівняння таймера  $t_1$  має вигляд (рис. 14.15):

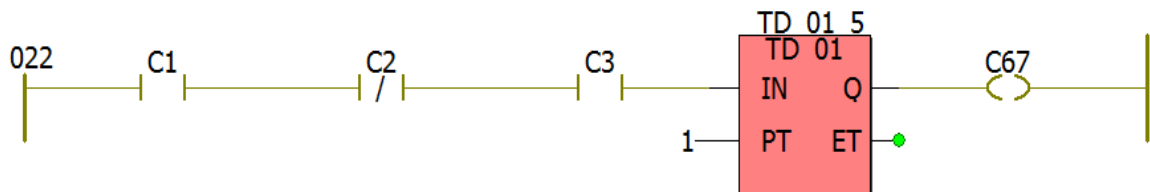


Рисунок 14.15 – Коло таймера із затримкою часу 0,1 секунда

На вхід IN подається умова включення початку відліку таймера, на вхід PT прописується час затримки в секундах, який множиться на дискрету часу.

У випадку коли програма працює у двох і більше режимах на мові LD можна реалізувати за допомогою переходів (рис. 14.16):

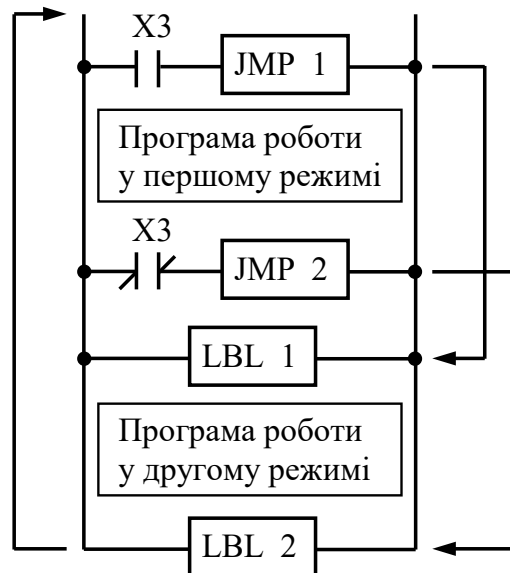



Рисунок 14.16 – Приклад програми із переходами

Для вибору міток і переходів натискаємо кнопку Connector/Jump or Label . У діалоговому вікні обираємо мітку або перехід і вказуємо його назву (рис. 14.17):

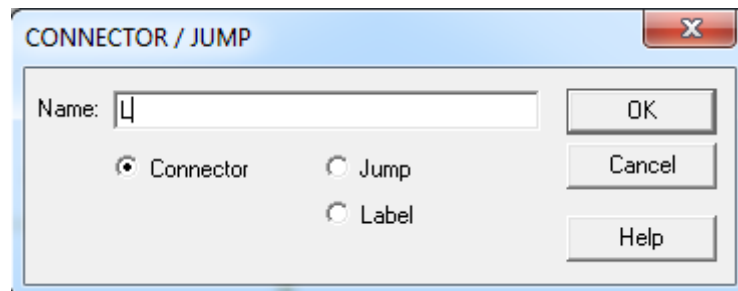


Рисунок 14.17 – Налаштування переходів

Також замість правої контактної рейки відразу йде з'єднання на перехід, мітка ставиться перед підпрограмою роботи в іншому режимі (рис. 14.18):

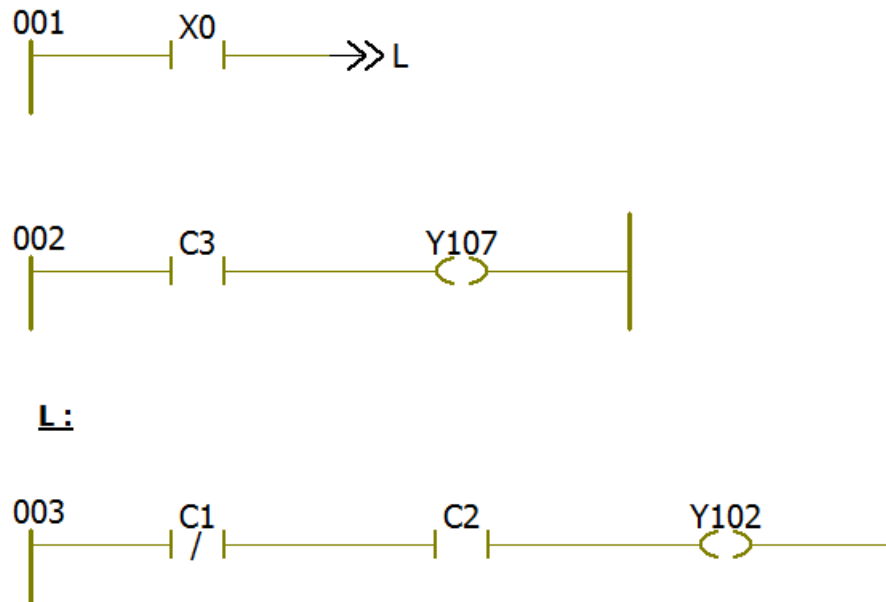





Рисунок 14.18 – Приклад програмування переходів на LD

Наступним кроком є компіляція програми. Вона здійснюється за допомогою кнопки Make . Якщо в програмі не знайдеться помилок, то її  кна сміливо записувати в контролер. Спочатку потрібно натиснути кнопку  і в вікні, що з'явилося натиснути кнопку Download (D) (рис. 14.19).

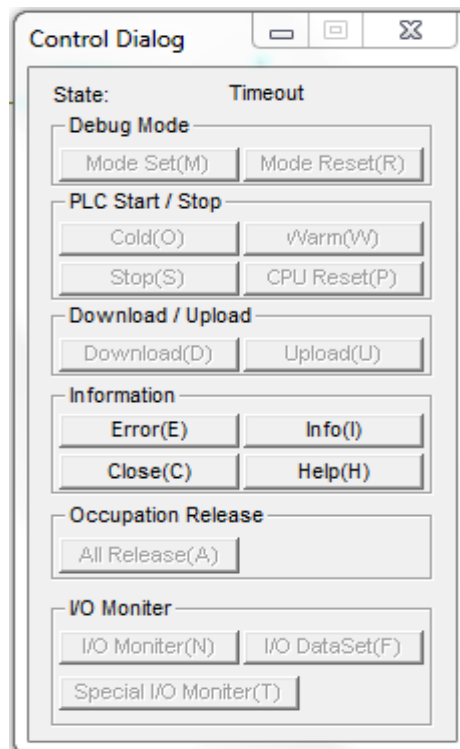
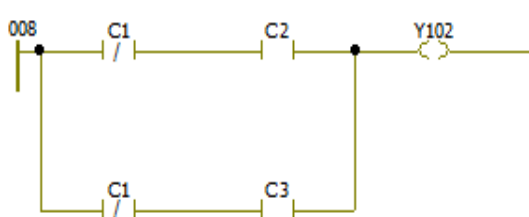
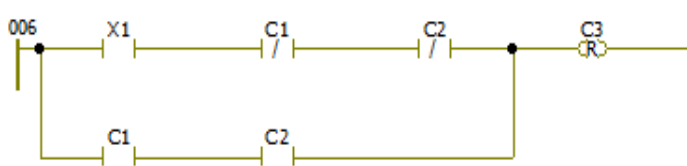
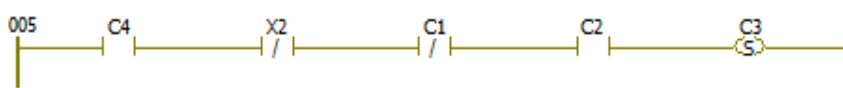
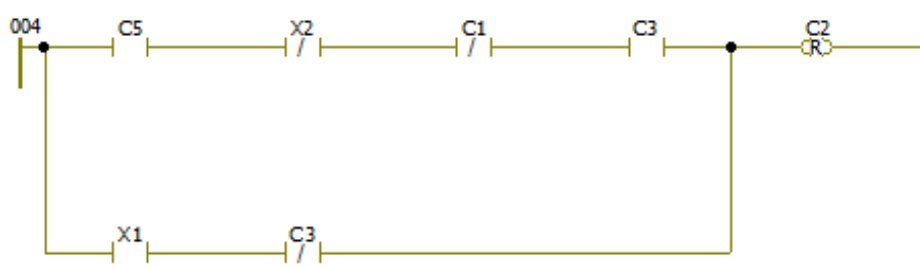
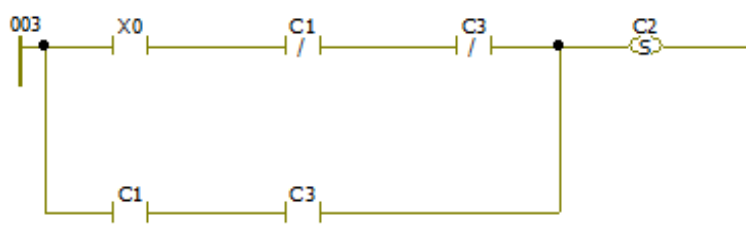
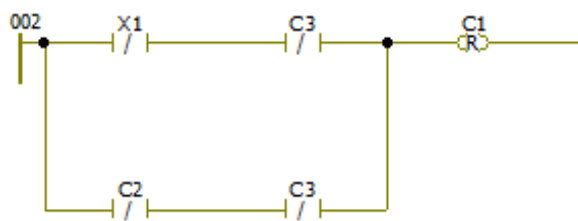
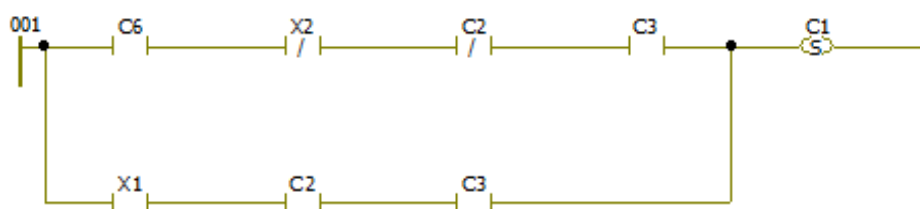


Рисунок 14.19 – Меню завантаження у контролер

Релейно-контактна схема для задачі з прикладу 1 (рис. 14.20):



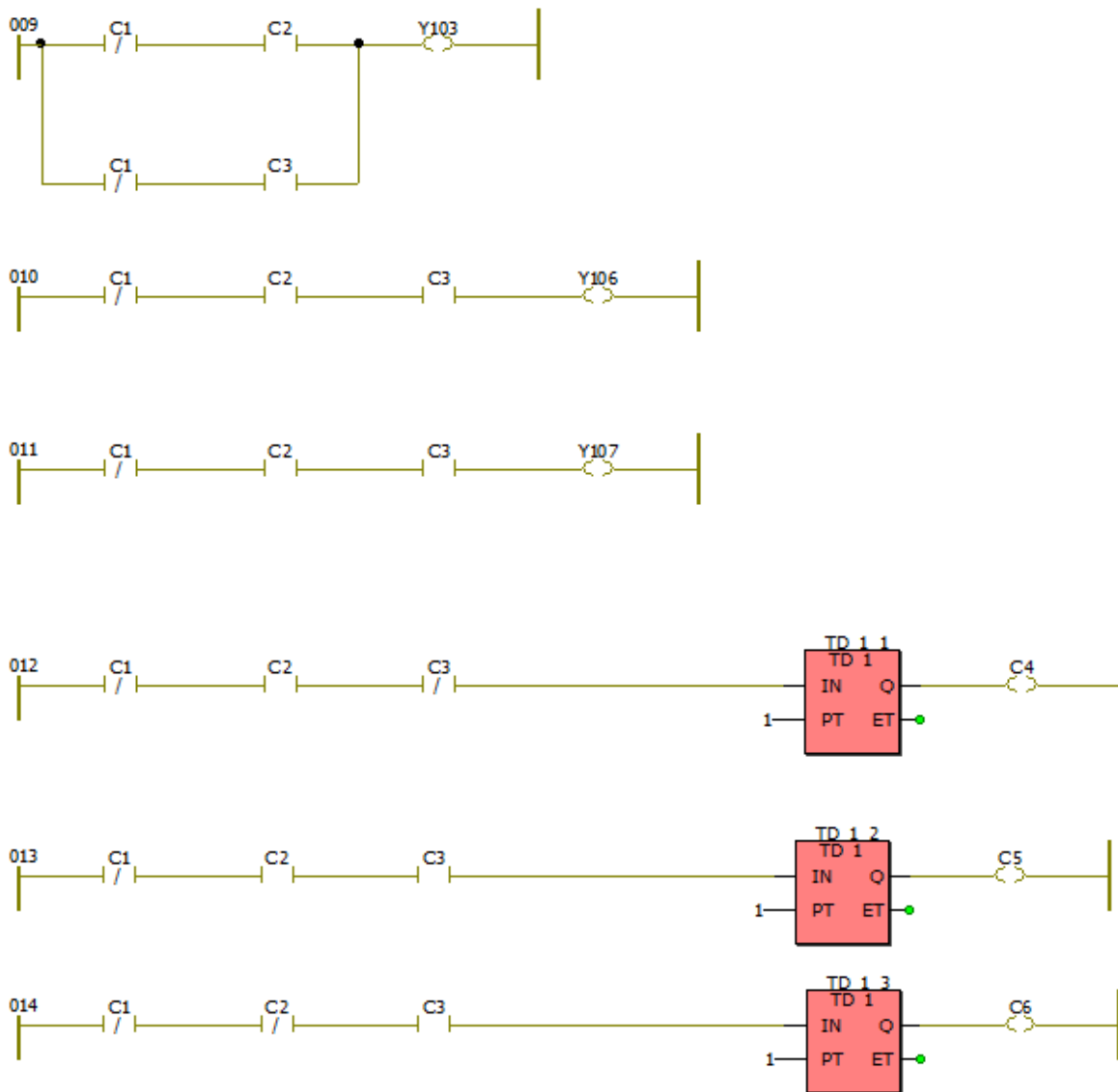


Рисунок 14.20 – Приклад програми на мові LD

Програма за отриманими виразами на мові IL має наступний вигляд

«Circuit №1»	ORB	ORB	LD Y106
LD TD67	SET M2	RES M3	OUT Y107
ANI X2	«Circuit №4»	«Circuit №7»	«Circuit №12»
ANI M2	LD TD66	LDI M1	LDI M1
AND M3	ANI X2	ANI M2	AND M2
LD X1	ANI M1	AND M3	ANI M3
AND M2	AND M3	OUT Y101	OUT TD65 .1S 10
AND M3	LD X1	«Circuit №8»	«Circuit №13»
ORB	ANI M3	LDI M1	LDI M1
SET M1	ORB	AND M2	AND M2

«Circuit №2»	RES M2	LDI M1	AND M3
LDI X1	«Circuit №5»	AND M3	OUT TD66 .1S 10
ANI M3	LD TD65	ORB	«Circuit №14»
LDI M2	ANI X2	OUT Y102	LDI M1
ANI M3	ANI M1	«Circuit №9»	ANI M2
ORB	AND M2	LD Y102	AND M3
RES M1	SET M3	OUT Y103	OUT TD67 .1S 10
«Circuit №3»	«Circuit №6»	«Circuit №10»	
LD X0	LD X1	LDI M1	
ANI M1	ANI M1	AND M2	
ANI M3	ANI M2	AND M3	
LD M1	LD M1	OUT Y106	
AND M3	AND M2	«Circuit №11»	

Розглянемо наступний приклад.

*Приклад 2.* Умови роботи схеми задані у прикладі 1. Різниця полягає в тому, що схема працює у двох режимах, які перемикаються за допомогою кнопки  $d$ . В режимі 1, коли  $d = 0$  схема працює як у прикладі 1. Якщо  $d = 1$ , схема генерує тільки числа 1 та 4 із затримкою 3 секунди.

В даному випадку необхідно окремо виконати синтез кожного режиму роботи, а потім об'єднати їх у програмі за допомогою команд переходів.

Для реалізації умови роботи режиму 2 зобразимо граф переходів, показаний на рис. 14.21.

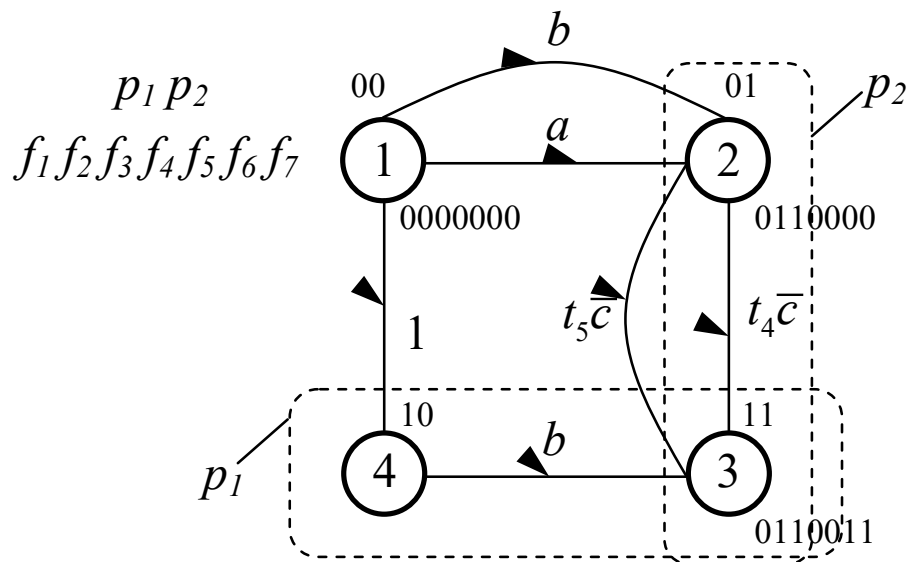


Рисунок 14.21 – Граф переходів до 2 режиму роботи

Як видно з графу переходів, у другому режимі роботи використано тільки 2 тригери, так як в схемі 3 стани. Також, для реалізацій генератора чисел «1» та «4» використані тільки чотири вихідні сингали  $f_2, f_3, f_6, f_7$ .

На основі графу переходів запишемо умови вмикання та скидання тригерів

$$S_{p_1} = t_4 \bar{c} p_2;$$

$$R_{p_1} = \bar{p}_2 + t_5 \bar{c} p_2 = \bar{p}_2 + t_5 \bar{c};$$

$$S_{p_2} = a \bar{p}_1;$$

$$R_{p_2} = b \bar{p}_1 + b p_1 = b.$$

Рівняння для вихідних функцій та таймерів

$$f_2 = \bar{p}_1 p_2 + p_1 p_2 = p_2;$$

$$f_3 = \bar{p}_1 p_2 + p_1 p_2 = p_2 = f_2;$$

$$f_6 = p_1 p_2;$$

$$f_7 = p_1 p_2 = f_6;$$

$$T_4 = \bar{p}_1 p_2;$$

$$T_5 = p_1 p_2.$$

Позначимо нові змінні відповідно до мови програмування

$d - X3; T_4 - TD68; T_5 - TD69.$

Алгоритм, за яким працюватиме програма, наведений на рис. 14.22.

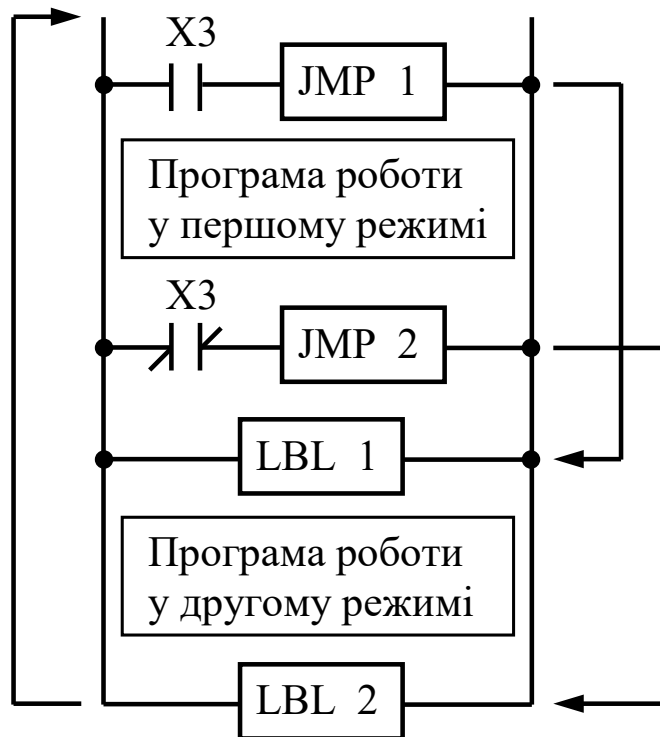


Рисунок 14.22 – Алгоритм роботи програми

Як видно з рис. 14.22, при відсутності сигналу X3 на початку роботи, алгоритм ігнорує перший перехід (JMP 1) і схема працює у першому режимі. Далі за допомогою переходу JMP 2 схема повертається у початковий стан. При подачі сигналу X3 на початку роботи алгоритм відразу переходить на мітку LBL 1 і працює у другому режимі.

Програма на мові ПЛ має наступний вигляд. Спробуйте самостійно перетворити її на мову LD.

<b>«Circuit №1»</b>	<b>«Circuit №5»</b>	AND M2	OUT TD67 .1S 10
LD X3	LD TD66	LDI M1	<b>«Circuit №16»</b>
[	ANI X2	AND M3	LDI X3
JMP 1	ANI M1	ORB	[
]	AND M3	OUT Y102	JMP 2
<b>«Circuit №2»</b>	LD X1	<b>«Circuit №10»</b>	]
LD TD67	ANI M3	LD Y102	<b>«Circuit №17»</b>
ANI X2	ORB	OUT Y103	[
ANI M2	RES M2	<b>«Circuit №11»</b>	LBL 1
AND M3	<b>«Circuit №6»</b>	LDI M1	]
LD X1	LD TD65	AND M2	<b>«Circuit №18»</b>
AND M2	ANI X2	AND M3	LD TD68
AND M3	ANI M1	OUT Y106	ANI X2
ORB	AND M2	<b>«Circuit №12»</b>	AND M2
SET M1	SET M3	LD Y106	SET M1
<b>«Circuit №3»</b>	<b>«Circuit №7»</b>	OUT Y107	<b>«Circuit №19»</b>
LDI X1	LD X1	<b>«Circuit №13»</b>	LDI M2
ANI M3	ANI M1	LDI M1	LD TD69
LDI M2	ANI M2	AND M2	ANI X2
ANI M3	LD M1	ANI M3	ORB
ORB	AND M2	OUT TD65 .1S 10	RES M1
RES M1	ORB	<b>«Circuit №14»</b>	<b>«Circuit №20»</b>
<b>«Circuit №4»</b>	RES M3	LDI M1	LD X0
LD X0	<b>«Circuit №8»</b>	AND M2	ANI M1
ANI M1	LDI M1	AND M3	SET M2
ANI M3	ANI M2	OUT TD66 .1S 10	<b>«Circuit №21»</b>
LD M1	AND M3	<b>«Circuit №15»</b>	LD X1
AND M3	OUT Y101	LDI M1	RES M2
ORB	<b>«Circuit №9»</b>	ANI M2	<b>«Circuit №22»</b>
SET M2	LDI M1	AND M3	LD M2

OUT Y102	«Circuit №25»	LD M1
«Circuit №23»	LD Y106	AND M2
LD Y102	OUT Y107	OUT TD69 .1S 30
OUT Y103	«Circuit №26»	«Circuit №28»
«Circuit №24»	LDI M1	[
LD M1	AND M2	LBL 2
AND M2	OUT TD68 .1S 30	]
OUT Y106	«Circuit №27»	

## 14.2. Програма роботи

За варіантом, наведеним у таблиці 14.1, виконати наступні завдання:

1. Виконати логічний синтез схеми керування та отримати логічні рівняння (**виконується вдома під час СРС**).
2. Скласти програму на мові LD згідно отриманих логічних виразів та зробити адресацію всіх змінних.
3. Записати програму в контролер.
4. З'єднати контролер з лабораторним пультом, перейти до режиму безперервного відпрацювання програми (RUN) та перевірити відповідність роботи заданим умовам.
5. Показати працездатність програми викладачу та продемонструвати її роботу (**викладач підтверджує своїм підписом у протоколі працездатність програми**).
6. Експортувати програми у картинки для звіту.
7. Оформити звіт по лабораторній роботі та зробити висновки.

Таблиця 14.1 – Варіанти завдань

Бригада	Номер завдання, яке визначає умову роботи
1	1; 7
2	2; 8
3	3; 9
4	4; 10
5	5; 11
6	6; 12

### Умови роботи схем

1. Виконати синтез генератора наступної послідовності чисел: 1, 2, 3, 1, 2, 3, 1... із затримкою часу між числами 2 секунди. При натисненні кнопки «Пуск» генератор починає роботу, при натисненні кнопки «Стоп» схема вимикається.

2. Виконати синтез генератора наступної послідовності чисел: 5, 6, 9, 5, 6, 9, 5... із затримкою часу між числами 3 секунди. При натисненні кнопки «Пуск» генератор починає роботу, при натисненні кнопки «Стоп» схема вимикається.

3. Виконати синтез генератора наступної послідовності чисел: 0, 2, 4, 0, 2, 4, 0... із затримкою часу між числами 4 секунди. При натисненні кнопки «Пуск» генератор починає роботу, при натисненні кнопки «Стоп» схема вимикається.

4. Виконати синтез генератора наступної послідовності чисел: 0, 2, 4, 6, 0, 2, 4, 6, 0... із затримкою часу між числами 1 секунда. При натисненні кнопки «Пуск» генератор починає роботу, при натисненні кнопки «Стоп» схема вимикається.

5. Виконати синтез генератора наступної послідовності чисел: 1, 3, 5, 7, 1, 3, 5, 7, 1... із затримкою часу між числами 2 секунди. При натисненні кнопки «Пуск» генератор починає роботу, при натисненні кнопки «Стоп» схема вимикається.

6. Виконати синтез генератора наступної послідовності чисел: 9, 7, 5, 3, 9, 7, 5, 3, 9... із затримкою часу між числами 2 секунди. При натисненні кнопки «Пуск» генератор починає роботу, при натисненні кнопки «Стоп» схема вимикається.

7. Схема працює у двох режимах, перемикання між якими здійснюється за допомогою тумблера. У першому режимі схема генерує послідовність парних чисел, починаючи з нуля. Час затримки між числами повинен, починаючи з 1 секунди, збільшуватися на 1 секунду з кожним новим парним числом. При поверненні схеми у вихідний стан затримка часу знову починається з 1 секунди. Запуск схеми робити за допомогою кнопки «Пуск 1»,

зупинку за допомогою кнопки «Стоп 1». У другому режимі схема генерує одиниці, які світяться впродовж 3 секунд із паузою 4 секунди. Для керування пуском/зупинкою використовувати кнопки «Пуск 2» та «Стоп 2».

8. Схема працює у двох режимах, перемикання між якими здійснюється за допомогою тумблера. У першому режимі схема генерує послідовність непарних чисел, починаючи з одиниці. Час затримки між числами повинен, починаючи з 1 секунди, збільшуватися на 1 секунду з кожним новим непарним числом. При поверненні схеми у вихідний стан затримка часу знову починається з 1 секунди. Запуск схеми робити за допомогою кнопки «Пуск 1», зупинку за допомогою кнопки «Стоп 1». У другому режимі схема генерує послідовність чисел 1, 2, 3, 1, 2, 3, 1... із затримкою часу 5 секунд. Для керування пуском/зупинкою використовувати кнопки «Пуск 2» та «Стоп 2».

9. Схема працює у двох режимах, перемикання між якими здійснюється за допомогою тумблера. У першому режимі схема генерує послідовність чисел 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0.... Час затримки між числами повинен бути 2 секунди. Запуск схеми робити за допомогою кнопки «Пуск 1», зупинку за допомогою кнопки «Стоп 1». У другому режимі схема генерує послідовність чисел 1, 3, 7, 1, 3, 7... із затримкою часу 5 секунд. Для керування пуском/зупинкою використовувати кнопки «Пуск 2» та «Стоп 2».

10. Схема працює у двох режимах, перемикання між якими здійснюється за допомогою тумблера. У першому режимі схема генерує послідовність чисел 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Після надходження числа 9 схема зупиняє роботу. Для продовження необхідно натиснути кнопку «Продовжити». Час затримки між числами повинен бути 2 секунди. Запуск схеми робити за допомогою кнопки «Пуск 1», зупинку за допомогою кнопки «Стоп 1». У другому режимі схема генерує послідовність чисел 1, 3, 7, 1, 3, 7... із затримкою часу 5 секунд. Для керування пуском/зупинкою використовувати кнопки, що і для першого режиму.

11. Схема працює у двох режимах, перемикання між якими здійснюється за допомогою тумблера. У першому режимі схема генерує

послідовність чисел 9, 8, 7, 6, 5. Після надходження числа 5 для продовження роботи необхідно натиснути кнопку «Продовжити 1», при цьому генерація продовжується числами 4, 3, 2, 1, 0. Після надходження числа 0 схема знову зупиняє роботу. Для запуску повторної генерації необхідно натиснути кнопку «Продовжити 2». Час затримки між числами повинен бути 1 секунда. Запуск схеми робити за допомогою кнопки «Пуск 1», зупинку за допомогою кнопки «Стоп 1». У другому режимі схема генерує нулі, які світяться впродовж 2 секунд із паузою 1 секунди. Для керування пуском/зупинкою використовувати кнопки, що і для першого режиму.

12. Схема працює у двох режимах, перемикання між якими здійснюється за допомогою тумблера. У першому режимі схема генерує послідовність парних чисел в зворотному порядку 8, 6, 4, 2, 0, 8, 6, 2.... Після надходження чисел 6 та 2 схема зупиняється. Для продовження її роботи необхідно натискати кнопку «Продовжити». Час затримки між числами повинен бути 2 секунди. Запуск схеми робити за допомогою кнопки «Пуск 1», зупинку за допомогою кнопки «Стоп 1». У другому режимі схема генерує ту саму послідовність чисел, але без зупинки. Для керування пуском/зупинкою використовувати кнопки, що і для першого режиму.

### 14.3. Опис лабораторної установки

Схема електрична принципова лабораторної установки показана на рис. 14.23. Установка складається з пульта, на якому розміщений програмований логічний контролер HITACHI-A23DRP, що має 12 дискретних входів та 10 виходів, світлодіодний індикатор BS-AE16RD, що має 7 світлодіодів (адреси Y101-Y107), які утворюють собою десяткову цифру, автоматичний вимикач ІЕК С16, 4 тумблери (адреси X0-X3) та 4 кнопки з самоповерненням (адреси X4-X7). Відповідність виходів контролера до окремих світлодіодів індикатора також показаний на рис. 14.23. Живлення стенда здійснюється від мережі 220 В змінного струму, а світлодіодного індикатора від блоку живлення 12В постійного струму.

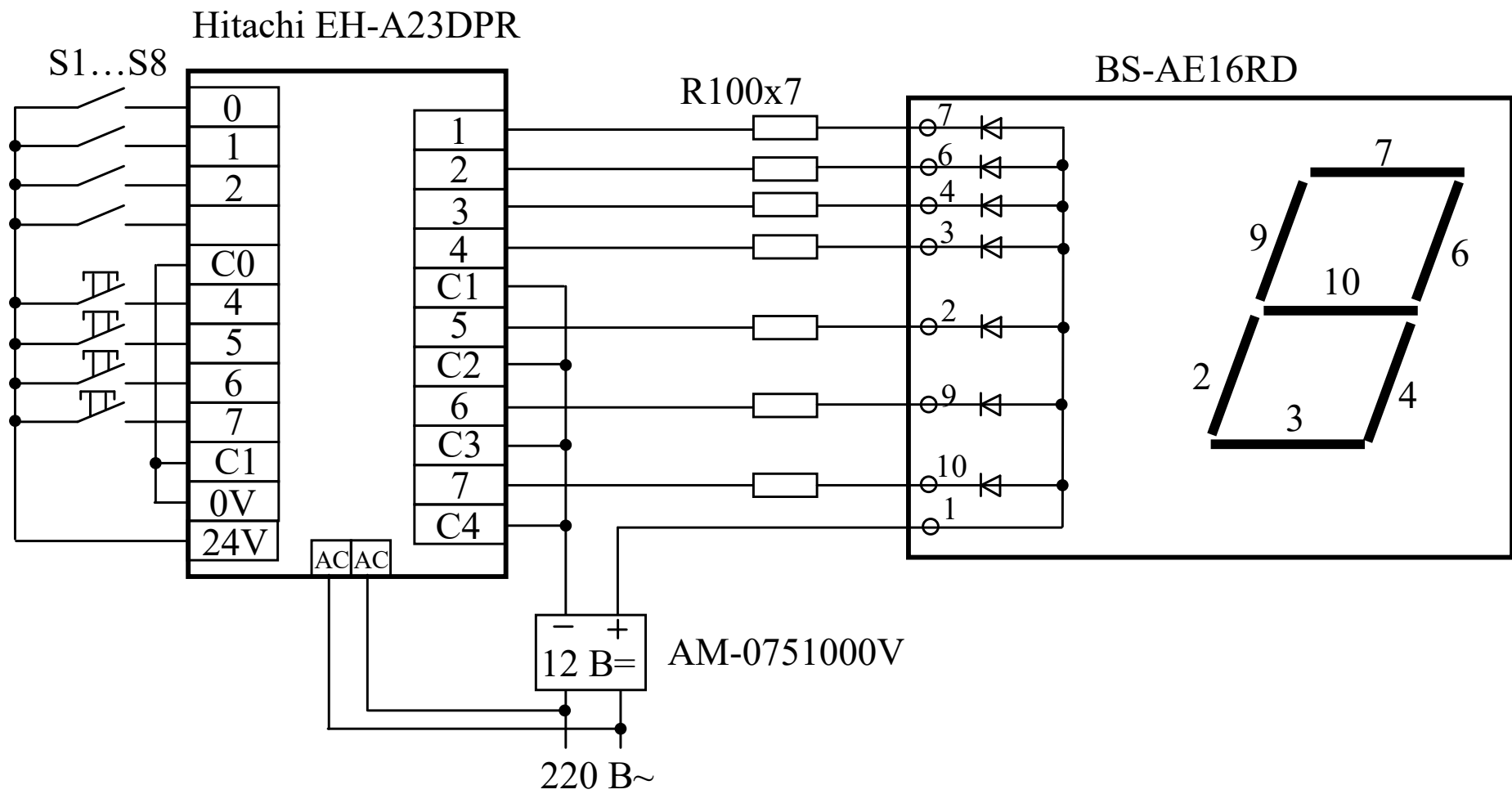


Рисунок 14.23 – Схема електрична принципова лабораторної установки

#### 14.4. Методичні вказівки

1. Після написання програми на мові LD необхідно відкрити ліву кришку логічного контролера та перевірити, щоб він не знаходився у робочому режимі (перемикач знаходиться у положенні STOP). Після цього запустити програму HLadder, виконати набір програми та записати її у контролера.

2. Після запису програми у контролер переключити перемикач на контролері у режим RUN, після чого перевірити працездатність програми.

3. Якщо у роботі програми виникли помилки необхідно переключити перемикач у положення STOP, відредагувати програму та знову записати її до контролера.

4. При написанні виразів до таймерів на мові IL слід використовувати дискрети часу 0.1S із відповідним множником. Наприклад, якщо потрібно реалізувати затримку часу 3 секунди на таймері TD65, у програму необхідно записати

```
OUT TD65 .1S 30
```

Це пов'язане із особливістю програмного забезпечення HLadder, яке після компіляції програми із дискретами 1S автоматично переводить їх у .1S.

5. У випадку, якщо при роботі схеми таймери не спрацьовують, необхідно замінити їх номери на інші, наприклад TD65 на TD5 і т.д.

#### 14.5. Зміст звіту

- 1) Титульний аркуш.
- 2) Мета роботи, програма роботи.
- 3) Опис та схема лабораторного стенду.
- 4) Умови роботи схем.
- 5) Логічний синтез схем керування.
- 6) Програми на мові LD для контролера та адресація всіх змінних.
- 7) Висновок про відповідність роботи заданим умовам.

### 14.6. Контрольні запитання

1. Що таке тригер? Які його особливості?
2. Що таке граф переходів? Як на основі графу переходів зобразити умови роботи схеми?
3. Як записуються умови включення тригера?
4. Як записуються умови скидання тригера?
5. Як на основі графу переходів записати вирази для вихідних змінних і для таймерів?
6. Як підключаються світлодіоди індикатора до виходів логічного контролера?

## Лабораторна робота № 15

### ДОСЛІДЖЕННЯ СПІЛЬНОЇ РОБОТИ ПРОГРАМОВАНИХ ЛОГІЧНИХ КОНТРОЛЕРІВ LOVATOLRD20RA024 З МОДУЛЯМИ РОЗШИРЕННЯ ТА СВІТЛОДІОДНИМИ ІНДИКАТОРАМИ

Тривалість лабораторної роботи – 4 год.

Тривалість домашньої роботи – 4 год.

**Мета роботи** – навчитися спільно програмувати контролер LOVATOLRD20RA024 з модулями розширення для опрацювання багатотактних схем.

#### 15.1. Основні теоретичні відомості

Методика синтезу схем на тригерах представлений у лабораторній роботі № 14.

Синтез схем на *RS*-тригерах виконують у такій послідовності.

1. За заданими умовами роботи схеми будують граф переходів.
2. Записують умови вмикання і скидання кожного тригера. Для цього охоплюють замкнутою лінією всі стани на графі переходів, у яких значення тієї ж самої проміжної змінної (вихідного сигналу тригера) дорівнює одиниці. Вхідні сигнали схеми, позначення яких розміщено на ребрах, що заходять в одержану замкнену область, встановлюють тригер у стан 1, а вхідні сигнали на ребрах, що виходять з цієї області, скидають тригер у стан 0.

Умови вмикання тригера записують у вигляді добутку сигналу на ребрі, що заходить в область, і сигналів решти тригерів, стан яких не змінюється під час переходу, позначеному ребром. Наприклад, якщо сигнал на ребрі, що заходить в область з одиничним значенням вихідного сигналу тригера  $P_1$ ,

дорівнює  $a$ , а тригери  $P_2$  і  $P_3$  не перемикаються, а зберігають стани  $P_2 = 1$ ,  $P_3 = 0$ , то умову вмикання тригера  $P_1$  записують у вигляді

$$S_{P_1} = a p_2 \overline{p_3}.$$

Якщо в замкнену область входить кілька ребер, то умову вмикання тригера записують у вигляді суми добутків відповідних сигналів, складених для кожного ребра.

Умову скидання тригера записують аналогічно для кожного ребра, що виходить з цієї області, і подають у вигляді формули  $R_{P_i}$ . Уведення сигналів решти тригерів в умови вмикання і скидання цього тригера гарантує здійснення переходів у потрібній послідовності.

Описану процедуру виконують для кожного тригера і визначають для них умови вмикання і скидання.

3. Записують вирази для вихідних сигналів схеми у вигляді комбінаційних функцій вихідних сигналів тригерів.

4. Якщо умовами роботи схеми передбачено технологічні затримки, то складають вирази для вхідних сигналів таймерів як комбінаційні функції вихідних сигналів тригерів.

5. За отриманими логічними формулами складають схему.

## 15.2. Програма роботи

1) Виконати логічний синтез за умовами роботи схеми для заданного варіанту (**виконується вдома, під час СРС**).

2) Скласти програму на мові LD згідно отриманих логічних виразів.

3) Записати програму в контролер.

4) Перевірити відповідність роботи заданим умовам (**викладач своїм підписом засвідчує правильність роботи програми**).

5) Оформити звіт та зробити висновки про результати роботи.

### Умови роботи схем

1. Розробити програму відображення цифр: при натисканні кнопки — спрацьовують світлодіоди, для отримання відповідних цифр вказаних у табл.15.1.

Таблиця 15.1 – Варіант завдань

№ Бригади	Індикатор №1		Індикатор №2		Індикатор №3	
	SB3	SB4	SB1	SB2	SB5	SB6
1	7	5	6	8	3	2
2	7	6	2	9	4	5
3	3	8	8	3	9	2
4	9	4	6	7	1	5
5	1	2	3	4	5	6
6	7	8	9	0	1	2

2. Розробити програму керування світлодіодними індикаторами: при натисканні кнопки SB1 — запуск програми послідовного спрацьовування світлодіодного індикатора з затримкою 0.5с по траєкторії вказаної у табл. 15.2, а при натисканні SB2 — схема переходить у вихідний стан. Позначення світлодіодів вказані на рис. 15.3.

Таблиця 15.2 – Варіант завдань

№ Бригади	Траєкторія
1	1.7,2.7,3.7→1.6,2.6,3.6→1.4,2.4,3.4→1.3,2.3,3.3→1.2,2.2,3.2→ →1.9,2.9,3.9→1.7,2.7,3.7→...
2	1.7→2.7→3.7→3.10→2.10→1.10→1.3→2.3→3.3→1.7→...
3	1.7→1.10→1.3→2.7→2.10→2.3→3.7→3.10→3.3→1.7→...
4	1.9,3.4→1.6,3.2→2.9,2.4→2.6,2.2→3.9,1.4→3.6,1.2→1.9,3.4→...
5	1.6,2.6,3.6→1.7,2.7,3.7→1.3,2.3,3.3→1.4,2.4,3.4→1.9,2.9,3.9→ →1.2,2.2,3.2→1.6,2.6,3.6→...
6	1.7→3.10→1.3→2.7→2.10→3.3→3.7→1.10→2.3→1.7→...

3. Розробити наступну програму: при натисканні кнопки SB1 — запуск програми послідовного спрацьовування світлодіодного індикатора з

затримкою 1.5 с по траєкторії указаної у табл. 15.3 та згідно з рис. 15.3, та на третьому світлодіодному індикаторі висвітлювався підрахунок кількості опрацьованих циклів, а при натисканні SB2 — виходила у вихідний стан:

Таблиця 15.3 – Варіант завдань

№ Бригади	Траєкторія
1	1.9,1.2→1.9,1.2,1.6,1.4→1.9,1.2,1.6,1.4,2.9,2.2→1.9,1.2,1.6,1.4,2.9,2.2,1.7→1.9,1.2,1.6,1.4,2.9,2.2,1.7,1.3→1.9,1.2,1.6,1.4,2.9,2.2,1.7,1.3,2.7→1.9,1.2,1.6,1.4,2.9,2.2,1.7,1.3,2.7,2.3→...
2	1.9→1.9,1.6→1.9,1.6,2.9→1.9,1.6,2.9,1.2→1.9,1.6,2.9,1.2,1.4→1.9,1.6,2.9,1.2,1.4,2.2→1.9,1.6,2.9,1.2,1.4,2.2,2.7,2.3→...
3	1.9→1.9,1.2→1.9,1.2,1.6→1.9,1.2,1.6,1.4→1.9,1.2,1.6,1.4,2.9→1.9,1.2,1.6,1.4,2.9,2.4→1.9,1.2,1.6,1.4,2.9,2.4,2.7,2.3→...
4	1.9→1.9,1.2→1.9,1.2,1.7→1.9,1.2,1.7,1.10→1.9,1.2,1.7,1.10,1.3→→1.9,1.2,1.7,1.10,1.3,2.7→1.9,1.2,1.7,1.10,1.3,2.7,2.3→...
5	1.9→1.9,1.2→1.9,1.2,1.3→1.9,1.2,1.3,2.3→1.9,1.2,1.3,2.3,2.4→1.9,1.2,1.3,2.3,2.4,2.6→1.9,1.2,1.3,2.3,2.4,2.6,2.7,1.7→...
6	1.7→1.7,1.6→1.7,1.6,2.9→1.7,1.6,2.9,2.7,2.6→1.7,1.6,2.9,2.7,2.6,2.4→1.7,1.6,2.9,2.7,2.6,2.4,2.3→1.7,1.6,2.9,2.7,2.6,2.4,2.3,2.2→...

### 15.3. Опис лабораторної установки

У даний стенд входить 2 логічних контролера LOVATOLRD20RA024 з модулями розширення та 3 світлодіодних індикатора, автоматичний вимикач, блок живлення (10V), 6 кнопок, трансформатор (220/24V).Схема електрична принципова зображена на рис. 15.1. Схема підключення зображена на рис. 15.2.

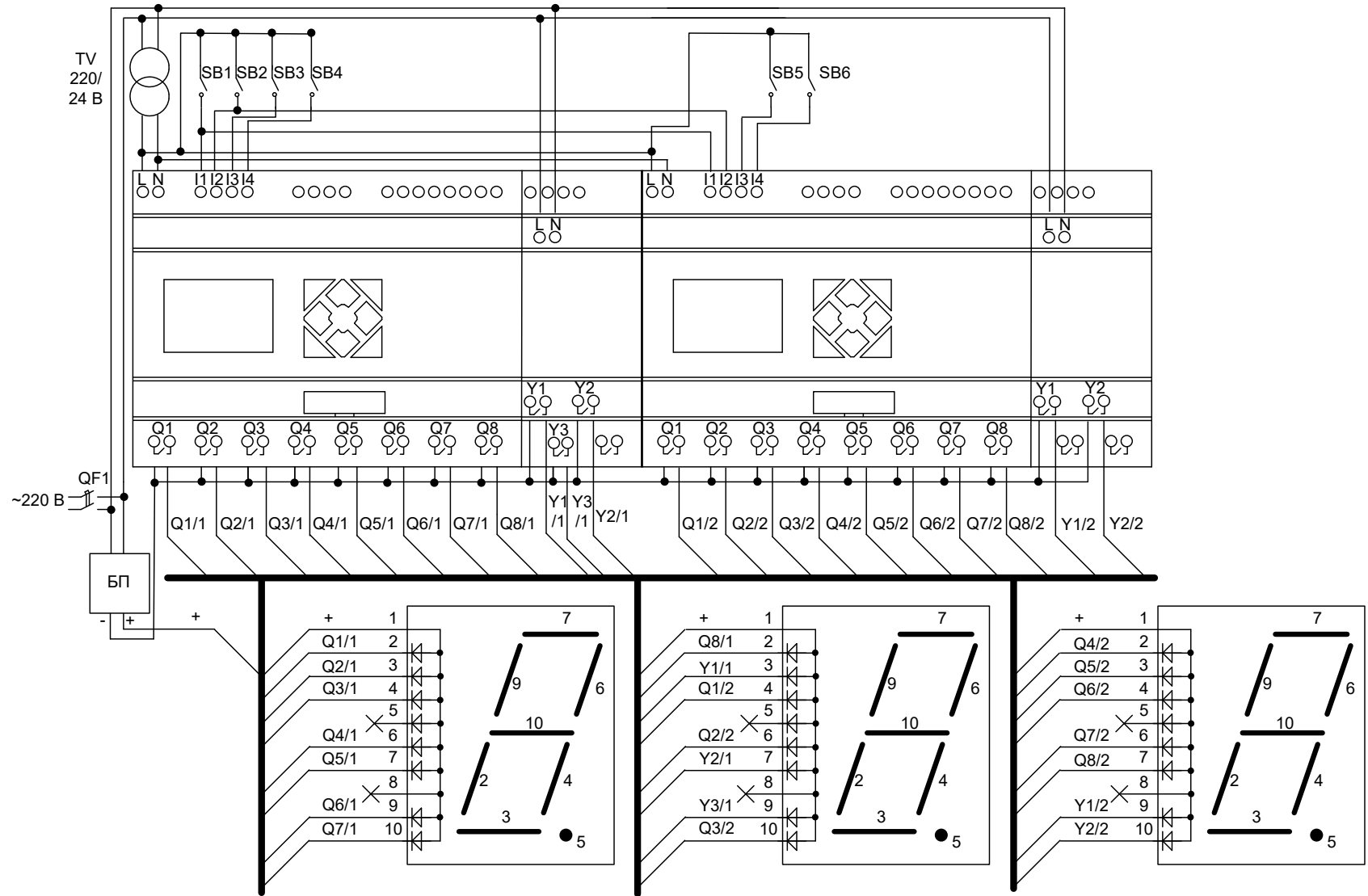


Рисунок 15.1 – Схема електрична принципова.

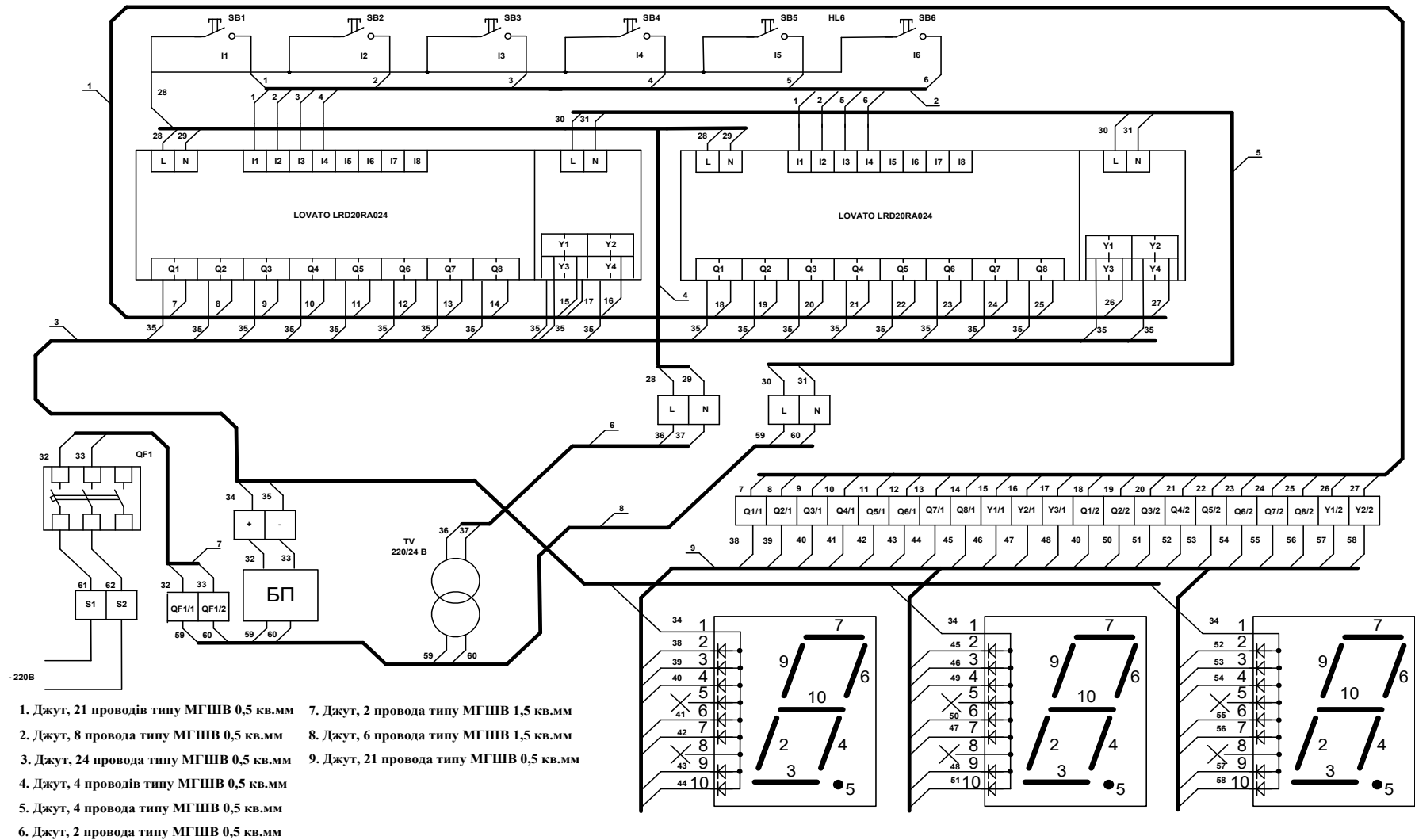


Рисунок 15.2 – Схема підключення

Схеми світлодіодних індикаторів зображена на рис. 15.3.

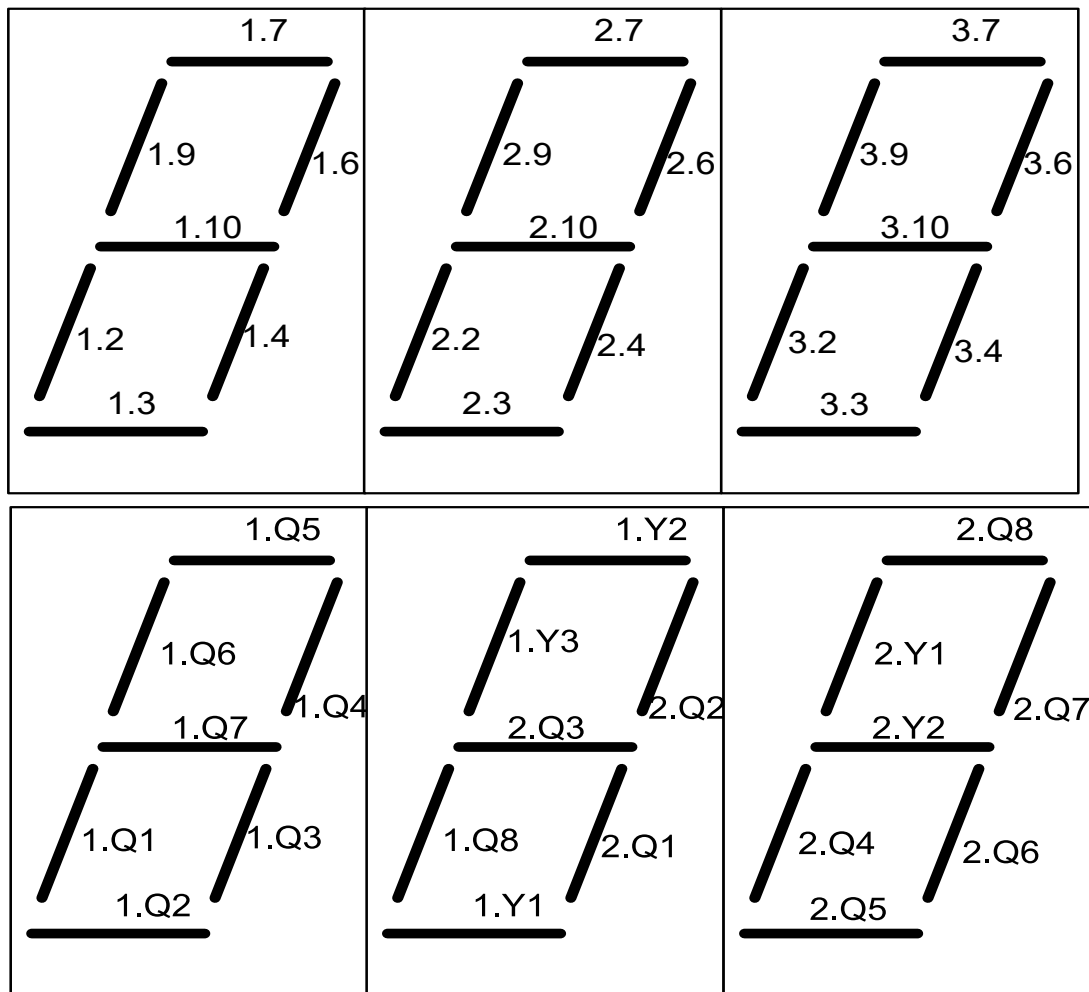


Рисунок 15.3 – Схеми світлодіодних індикаторів

Умовні позначення:

SB1, SB2, SB3, SB4 — кнопки (вхідні сигнали), під'єднані до першого контролера;

SB1, SB2, SB5, SB6— кнопки (вхідні сигнали), під'єднані до другого контролера;

1.Q1,1.Q2 ...1.Q8, 1.Y1, 1.Y2, 1.Y3 —вихідні сигнали першого контролера, що відповідають певним сегментам світлодіодних індикаторів.

2.Q1, 2.Q2 ...2.Q8, 2.Y1, 2.Y2 —вихідні сигнали другого контролера, що відповідають певним сегментам світлодіодних індикаторів.

У контролері використовуються такі позначення:

I –вхідні сигнали;

Q—вихідні сигнали;

M – проміжна змінна, комірка пам'яті;

T – затримка часу;

“ - “ – контактне з'єднання.

Для першого контролера:

I1, I2, I3, I4 – кнопки SB1, SB2, SB3, SB4 відповідно;

Q1, Q2,...,Q8, Y1, Y2,Y3—вихідні сигнали Q1, Q2,...,Q8, Y1, Y2,Y3 відповідно;

Для другого контролера:

I1, I2, I3, I4 – кнопки SB1, SB2, SB5, SB6 відповідно;

Q1, Q2,...,Q8, Y1, Y2,Y3—вихідні сигнали Q1, Q2,...,Q8, Y1, Y2,Y3 відповідно;

Для початку програмування контролера вмикаємо автомат QF1. Після загоряння екранів контролера натискаємо кнопку “SEL”, з'явиться меню контролера. Якщо на екрані є рядок “STOP”, вибираємо її та натискаємо “OK”. Для початку написання програми заходимо у меню контролера у строку “LADDER”. Для вибору потрібного контакту натискаємо “SEL” та стрілками вибираємо відповідне позначення входу, виходу, таймеру, проміжної змінної чи контактної з'єднання, потім натискаємо кнопку «вправо» та вибираємо потрібний номер входу, виходу, таймеру, проміжної змінної.

Розглянемо роботу стенду на прикладі, розглянутому нижче:

*Приклад 1.* Розробити програму відображення цифр: при натисканні кнопки — спрацьовують світлодіоди, для отримання відповідних цифр вказаних у табл. 15.4.

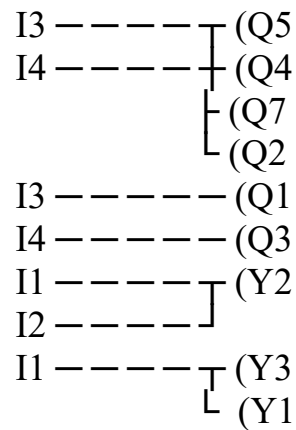
*Таблиця 15.4 – Завдання*

Індикатор №1		Індикатор №2		Індикатор №3	
SB3	SB4	SB1	SB2	SB5	SB6
2	3	5	7	6	8

Кожен контролер відповідає за свої світло світлодіоди, тому розіб'ємо задачу на дві(для кожного контролера окремо).

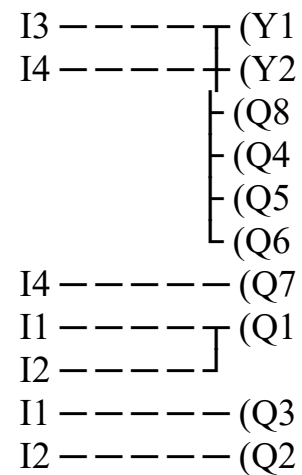
Перший контролер:

Записуємо текст програми у контролер



Другий контролер:

Записуємо текст програми у контролер



*Приклад 2.* Розглянемо наступний приклад: при натисканні кнопки SB1 — запуск програми послідовного спрацьовування світлодіодного індикатора з затримкою 0.5с по траєкторії указаній у табл. 15.5, а при натисканні SB2 — схема повертається у вихідний стан:

Таблиця 15.5 – Завдання

Траєкторія
1.7→2.7→3.7→3.6→3.4→3.3→2.3→1.3→1.2→1.9→1.7→2.7...

Кожен контролер відповідає за свої світло світлодіоди, тому розіб'ємо задачу на 2 (для кожного контролера окремо).

Перший контролер:

1) Побудуємо графо-перехід (рис. 15.4):

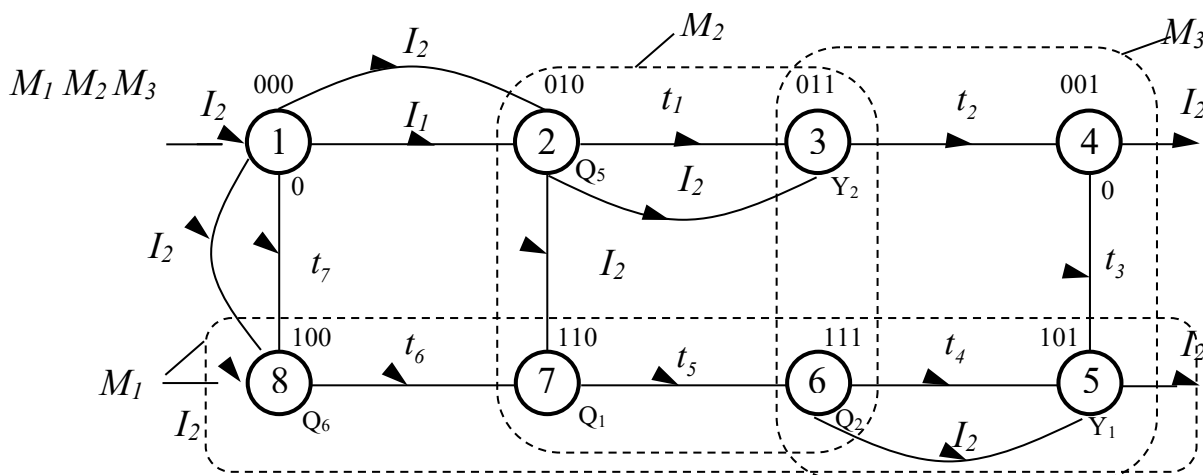


Рисунок 15.4 – Графоперехід для першого контролера

2) Складаємо рівняння для тригерів, таймерів та вихідних змінних:

Умови встановлення тригерів:

$$S_{M1} = T_3 \cdot \bar{M}_2 \cdot M_3;$$

$$S_{M2} = I_1 \cdot \bar{M}_1 \cdot \bar{M}_3 + T_4 \cdot M_1 \cdot M_3;$$

$$S_{M3} = T_1 \cdot \bar{M}_1 \cdot M_2.$$

Умови скидання тригерів:

$$R_{M1} = T_7 \cdot \bar{M}_2 \cdot \bar{M}_3 + I_2 \cdot (M_2 \cdot \bar{M}_3 + \bar{M}_2 \cdot \bar{M}_3) = T_7 \cdot \bar{M}_2 \cdot \bar{M}_3 + I_2 \cdot \bar{M}_3;$$

$$R_{M2} = T_2 \cdot \bar{M}_1 \cdot M_3 + T_6 \cdot M_1 \cdot \bar{M}_3 + I_2 \cdot (M_1 \cdot M_3 + \bar{M}_1 \cdot \bar{M}_3);$$

$$\begin{aligned} R_{M3} &= T_5 \cdot M_1 \cdot M_2 + I_2 \cdot (M_1 \cdot \bar{M}_2 + \bar{M}_1 \cdot \bar{M}_2 + \bar{M}_1 \cdot M_2) = \\ &= T_5 \cdot M_1 \cdot M_2 + I_2 \cdot (M_1 \cdot \bar{M}_2 + \bar{M}_1). \end{aligned}$$

Рівняння для таймерів:

$$T_1 = \bar{M}_1 \cdot M_2 \cdot \bar{M}_3 \quad (0.5c);$$

$$T_2 = \bar{M}_1 \cdot M_2 \cdot M_3 \quad (0.5c);$$

$$T_3 = \bar{M}_1 \cdot \bar{M}_2 \cdot M_3 \quad (2.0c);$$

$$T_4 = M_1 \cdot \bar{M}_2 \cdot M_3 \quad (0.5c);$$

$$T_5 = M_1 \cdot M_2 \cdot M_3 \quad (0.5c);$$

$$T_6 = M_1 \cdot M_2 \cdot \bar{M}_3 \quad (0.5c);$$

$$T_7 = M_1 \cdot \bar{M}_2 \cdot \bar{M}_3 \quad (0.5c).$$

Рівняння для вихідних сигналів:

$$Q_1 = M_1 \cdot M_2 \cdot \bar{M}_3;$$

$$Q_2 = M_1 \cdot M_2 \cdot M_3;$$

$$Q_5 = \bar{M}_1 \cdot M_2 \cdot \bar{M}_3;$$

$$Q_6 = M_1 \cdot \bar{M}_2 \cdot \bar{M}_3;$$

$$Y_1 = M_1 \cdot \bar{M}_2 \cdot M_3;$$

$$Y_2 = \bar{M}_1 \cdot M_2 \cdot M_3.$$

3) Записуємо отриманні рівняння у контролер (у вигляді релейно-контактної схеми):

T3	—	m2	—	M3	—	↑M1
T7	—	m2	—	m3	┘	↓M1
I2	—	—	—	m3	┘	
I1	—	m1	—	m3	┘	↑M2
T4	—	M1	—	M3	┘	
T2	—	m1	—	M3	┘	↓M2
T6	—	M1	—	m3	┘	
I2	┘	m1	—	m3	┘	
		M1	—	M3	┘	
T1	—	m1	—	M2	—	↑M3
T5	—	M1	—	M2	┘	↓M3
I2	┘	m1	—	M2	┘	
		m1	—	—	┘	
m1	—	M2	—	m3	—	(T1
m1	—	M2	—	M3	—	(T2
m1	—	m2	—	M3	—	(T3
M1	—	m2	—	M3	—	(T4
M1	—	M2	—	M3	—	(T5
M1	—	M2	—	m3	—	(T6
M1	—	m2	—	m3	—	(T7
M1	—	M2	—	m3	—	(Q1
M1	—	M2	—	M3	—	(Q2
m1	—	M2	—	m3	—	(Q5
M1	—	m2	—	m3	—	(Q6
M1	—	m2	—	M3	—	(Y1
m1	—	M2	—	M3	—	(Y2

Другий контролер:

1) Побудуємо графо-перехід (рис. 15.5):

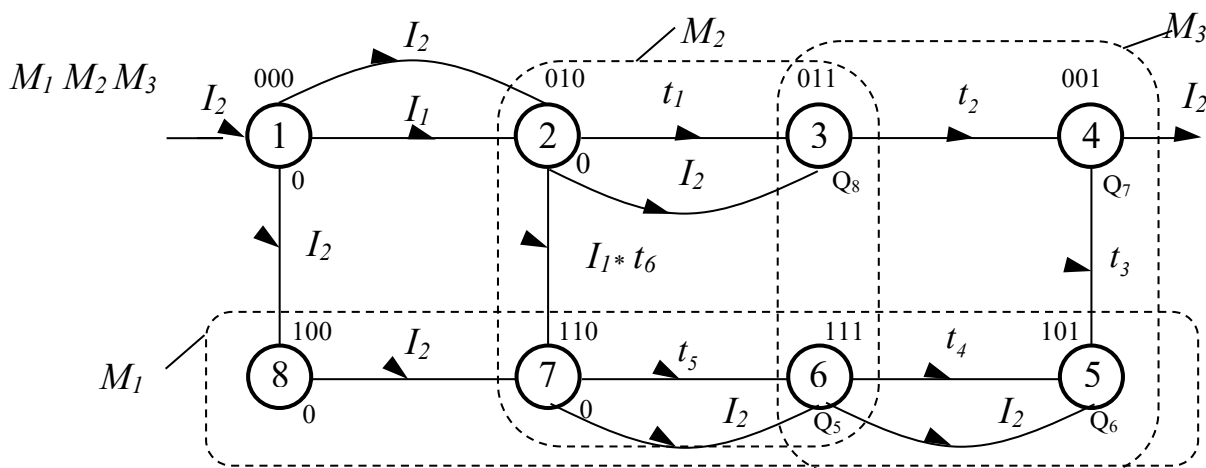


Рисунок 15.5 – Графоперехід для другого контролера.

2) Складаємо рівняння для тригерів, таймерів та вихідних змінних:

Умови встановлення тригерів:

$$S_{M1} = T_3 \cdot \bar{M}_2 \cdot M_3;$$

$$S_{M2} = I_1 \cdot \bar{M}_1 \cdot \bar{M}_3 + T_4 \cdot M_1 \cdot M_3 + I_2 \cdot M_1 \cdot M_3;$$

$$S_{M3} = T_1 \cdot \bar{M}_1 \cdot M_2.$$

Умови скидання тригерів:

$$R_{M1} = T_6 \cdot I_1 \cdot M_2 \cdot \bar{M}_3 + I_2 \cdot \bar{M}_2 \cdot \bar{M}_3;$$

$$R_{M2} = T_2 \cdot \bar{M}_1 \cdot M_3 + I_2 \cdot (M_1 \cdot \bar{M}_3 + \bar{M}_1 \cdot \bar{M}_3) = T_2 \cdot \bar{M}_1 \cdot M_3 + I_2 \cdot \bar{M}_3;$$

$$\begin{aligned} R_{M3} &= T_5 \cdot M_1 \cdot M_2 + I_2 \cdot (M_1 \cdot M_2 + \bar{M}_1 \cdot \bar{M}_2 + \bar{M}_1 \cdot M_2) = \\ &= T_5 \cdot M_1 \cdot M_2 + I_2 \cdot (M_2 + \bar{M}_1 \cdot \bar{M}_2). \end{aligned}$$

Рівняння для таймерів:

$$T_1 = \bar{M}_1 \cdot M_2 \cdot \bar{M}_3 \quad (1.0c);$$

$$T_2 = \bar{M}_1 \cdot M_2 \cdot M_3 \quad (0.5c)$$

$$T_3 = \bar{M}_1 \cdot \bar{M}_2 \cdot M_3 \quad (0.5c);$$

$$T_4 = M_1 \cdot \bar{M}_2 \cdot M_3 \quad (0.5c)$$

$$T_5 = M_1 \cdot M_2 \cdot M_3 \quad (0.5c);$$

$$T_6 = M_1 \cdot M_2 \cdot \bar{M}_3 \quad (2.0c).$$

Рівняння для вихідних сигналів:

$$Q_5 = M_1 \cdot M_2 \cdot M_3;$$

$$Q_6 = M_1 \cdot \bar{M}_2 \cdot M_3;$$

$$Q_7 = \bar{M}_1 \cdot \bar{M}_2 \cdot M_3;$$

$$Q_8 = \bar{M}_1 \cdot M_2 \cdot M_3.$$

3) Записуємо отриманні рівняння у контролер (у вигляді релейно-контактної схеми):

T3	—	m2	—	M3	—	↑M1
T6	—	I1	—	--	—	(M4
M4	—	M2	—	m3	┘	↓M1
I2	—	m2	—	m3	┘	
I1	—	m1	—	m3	┘	↑M2
T4	—	M1	—	M3	┘	
I2	—	M1	—	M3	┘	
T2	—	m1	—	M3	┘	↓M2
I2	—	—	—	m3	┘	
T1	—	m1	—	M2	—	↑M3
T5	—	M1	—	M2	┘	↓M3
I2	┘	m1	—	m2	┘	
		—	—	M2	┘	
m1	—	M2	—	m3	—	(T1
m1	—	M2	—	M3	—	(T2
m1	—	m2	—	M3	—	(T3
M1	—	m2	—	M3	—	(T4
M1	—	M2	—	M3	—	(T5
M1	—	M2	—	m3	—	(T6
M1	—	M2	—	M3	—	(Q5
M1	—	m2	—	M3	—	(Q6
m1	—	m2	—	M3	—	(Q7
m1	—	M2	—	M3	—	(Q8

*Приклад 3.* Розробити програму: при натисканні кнопки SB1 — запуск програми послідовного спрацьовування світлодіодного індикатора з затримкою 1.5 с по траєкторії вказаної у табл.15.6 та згідно з рис. 15.3, та рахувало кількість індикаторів, що світяться, а при натисканні SB2 — виходила у вихідний стан:

Таблиця 15.6 – Завдання

Траєкторія
1.9, 1.2 → 1.9, 1.2, 1.6, 1.4 → 1.9, 1.2, 1.6, 1.4, 2.2, 2.9 → 1.9, 1.2, 1.6, 1.4, 2.2, 2.9, 2.7, 2.3 → ...

Кожен контролер відповідає за свої світлодіоди, тому розіб'ємо задачу на 2 (для кожного контролера окремо).

Перший контролер:

2) Побудуємо графо-перехід (рис. 15.6):

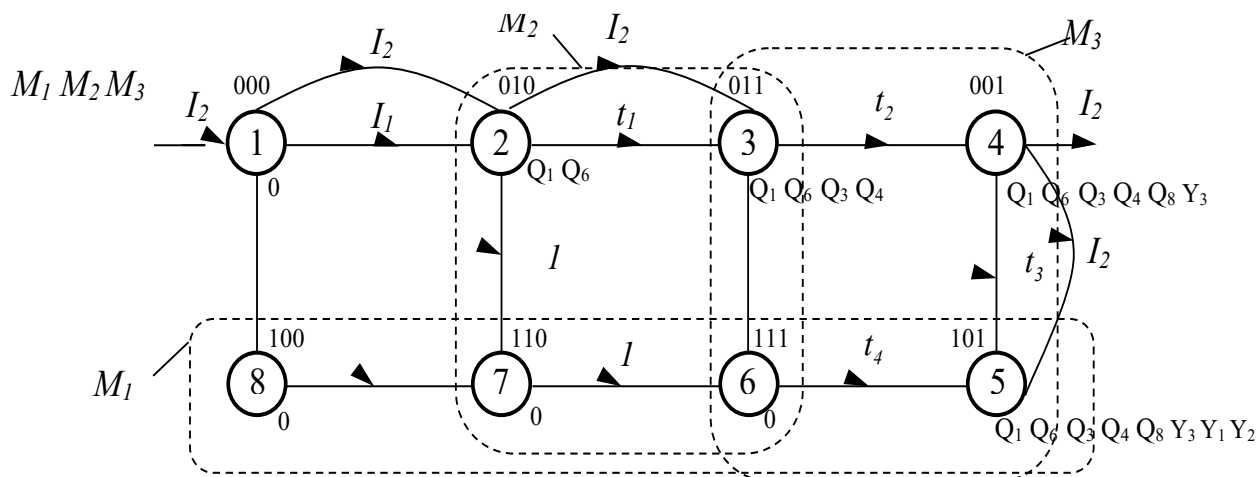


Рисунок 15.6 – Графоперехід для першого контролера

2) Складаємо рівняння для тригерів, таймерів та вихідних змінних:

Умови встановлення тригерів:

$$S_{M1} = T_3 \cdot \bar{M}_2 \cdot M_3;$$

$$S_{M2} = I_1 \cdot \bar{M}_1 \cdot \bar{M}_3 + T_4 \cdot M_1 \cdot M_3;$$

$$S_{M3} = T_1 \cdot \bar{M}_1 \cdot M_2.$$

Умови скидання тригерів:

$$R_{M1} = M_2 \cdot \bar{M}_3 + I_2 \cdot \bar{M}_2 \cdot M_3;$$

$$R_{M2} = T_2 \cdot \bar{M}_1 \cdot M_3 + I_2 \cdot \bar{M}_1 \cdot \bar{M}_3;$$

$$R_{M3} = M_1 \cdot M_2 + I_2 \cdot \bar{M}_1 \cdot \bar{M}_2 + I_2 \cdot \bar{M}_1 \cdot M_2 = M_1 \cdot M_2 + I_2 \cdot \bar{M}_1;$$

Рівняння для таймерів:

$$T_1 = \bar{M}_1 \cdot M_2 \cdot \bar{M}_3 \quad (1.5c);$$

$$T_2 = \bar{M}_1 \cdot M_2 \cdot M_3 \quad (1.5c);$$

$$T_3 = \bar{M}_1 \cdot \bar{M}_2 \cdot M_3 \quad (1.5c);$$

$$T_4 = M_1 \cdot \bar{M}_2 \cdot M_3 \quad (1.5c).$$

Рівняння для вихідних сигналів:

$$Q_1 = Q_6 = \bar{M}_1 \cdot M_2 \cdot \bar{M}_3 + \bar{M}_1 \cdot M_2 \cdot M_3 + \bar{M}_1 \cdot \bar{M}_2 \cdot M_3 + M_1 \cdot \bar{M}_2 \cdot M_3 = \\ = \bar{M}_1 \cdot M_2 + \bar{M}_2 \cdot M_3;$$

$$Q_3 = Q_4 = \bar{M}_1 \cdot M_2 \cdot M_3 + \bar{M}_1 \cdot \bar{M}_2 \cdot M_3 + M_1 \cdot \bar{M}_2 \cdot M_3 = \bar{M}_1 \cdot M_2 \cdot M_3 + \bar{M}_2 \cdot M_3;$$

$$Q_8 = Y_3 = \bar{M}_1 \cdot \bar{M}_2 \cdot M_3 + M_1 \cdot \bar{M}_2 \cdot M_3 = \bar{M}_2 \cdot M_3;$$

$$Y_1 = Y_2 = M_1 \cdot \bar{M}_2 \cdot M_3.$$

3) Записуємо отриманні рівняння у контролер (у вигляді релейно-контактної схеми):

T3	—	m2	—	M3	—	↑M1
I2	—	m2	—	M3	⊥	
m3	—	M2	—	—	⊥	↓M1
I1	—	m1	—	m3	⌋	↑M2
T4	—	M1	—	M3	⌋	
T2	—	m1	—	M3	⌋	↓M2
I2	—	m1	—	m3	⌋	
T1	—	m1	—	M2	—	↑M3
M1	—	—	—	M2	⌋	↓M3
I2	—	m1	—	—	⌋	
m1	—	M2	—	m3	—	(T1
m1	—	M2	—	M3	—	(T2
m1	—	m2	—	M3	—	(T3
M1	—	m2	—	M3	—	(T4
m1	—	M2	—	—	⊥	(Q1
m2	—	M3	—	—	⊥	(Q6
m1	—	M2	—	M3	⊥	(Q3
m2	—	—	—	M3	⊥	(Q4
m2	—	—	—	M3	⌋	(Q8
					⌋	(Y3
M1	—	m2	—	M3	⌋	(Y2
					⌋	(Y1

Другий контролер:

1) Побудуємо графо-перехід (рис. 15.7):

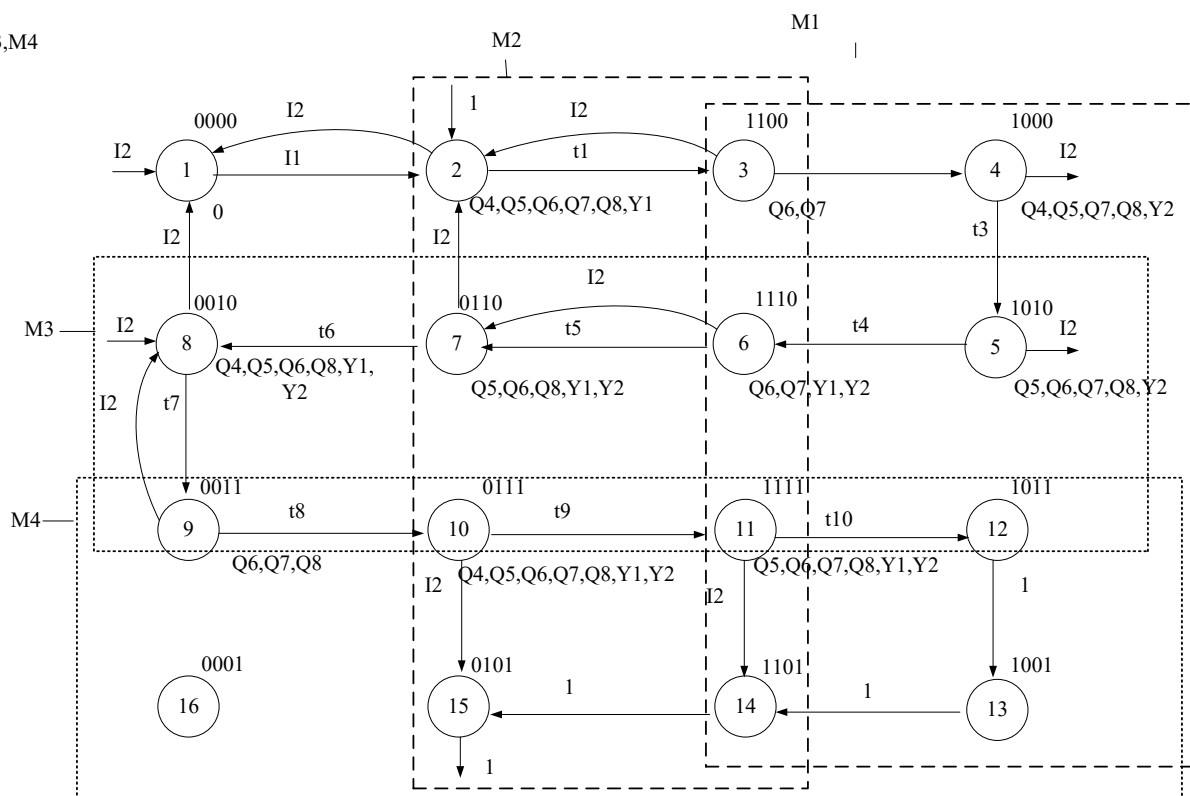


Рисунок 15.7 – Графоперехід для другого контролера

2) Складаємо рівняння для тригерів, таймерів та вихідних змінних:

Умови встановлення тригерів:

$$S_{M1} = T_1 M_2 \bar{M}_3 \bar{M}_4 + T_9 M_2 M_3 M_4;$$

$$S_{M2} = I_1 \bar{M}_1 \bar{M}_3 \bar{M}_4 + T_4 M_1 M_3 \bar{M}_4 + T_8 \bar{M}_1 M_3 M_4 + M_1 \bar{M}_3 M_4;$$

$$S_{M3} = T_3 M_1 \bar{M}_2 \bar{M}_4;$$

$$S_{M4} = T_7 \bar{M}_1 \bar{M}_2 M_3.$$

Умови скидання тригерів:

$$R_{M1} = I_2 \bar{M}_2 \bar{M}_3 \bar{M}_4 + I_2 \bar{M}_2 M_3 \bar{M}_4 + I_2 M_2 M_3 \bar{M}_4 + T_5 M_2 M_3 \bar{M}_4 + \\ + M_2 \bar{M}_3 M_4 + I_2 M_2 \bar{M}_3 \bar{M}_4;$$

$$R_{M2} = I_2 \bar{M}_1 \bar{M}_3 \bar{M}_4 + T_2 M_1 \bar{M}_3 \bar{M}_4 + T_6 \bar{M}_1 M_3 \bar{M}_4 + T_{10} M_1 M_3 M_4;$$

$$R_{M3} = I_2 \bar{M}_1 \bar{M}_2 \bar{M}_4 + I_2 \bar{M}_1 M_2 \bar{M}_4 + I_2 M_1 M_2 M_4 + I_2 \bar{M}_1 M_2 M_4 + M_1 \bar{M}_2 M_4;$$

$$R_{M4} = I_2 \bar{M}_1 \bar{M}_2 M_3 + \bar{M}_1 M_2 \bar{M}_3.$$

Рівняння для таймерів:

$$T_1 = \bar{M}_1 M_2 \bar{M}_3 \bar{M}_4 \quad (1.5c);$$

$$T_2 = M_1 M_2 \bar{M}_3 \bar{M}_4 \quad (1.5c);$$

$$T_3 = M_1 \bar{M}_2 \bar{M}_3 \bar{M}_4 \quad (1.5c);$$

$$T_4 = M_1 \bar{M}_2 M_3 \bar{M}_4 \quad (1.5c);$$

$$T_5 = M_1 M_2 M_3 \bar{M}_4 \quad (1.5c);$$

$$T_6 = \bar{M}_1 M_2 M_3 \bar{M}_4 \quad (1.5c);$$

$$T_7 = \bar{M}_1 \bar{M}_2 M_3 \bar{M}_4 \quad (1.5c);$$

$$T_8 = \bar{M}_1 \bar{M}_2 M_3 M_4 \quad (1.5c);$$

$$T_9 = \bar{M}_1 M_2 M_3 M_4 \quad (1.5c);$$

$$T_{10} = M_1 M_2 M_3 M_4 \quad (1.5c).$$

Рівняння для вихідних сигналів:

$$Q_4 = \bar{M}_1 M_2 \bar{M}_3 \bar{M}_4 + M_1 \bar{M}_2 \bar{M}_3 \bar{M}_4 + \bar{M}_1 \bar{M}_2 M_3 \bar{M}_4 + \bar{M}_1 M_2 M_3 M_4;$$

$$Q_5 = \bar{M}_1 M_2 \bar{M}_3 \bar{M}_4 + M_1 \bar{M}_2 \bar{M}_3 \bar{M}_4 + M_1 \bar{M}_2 M_3 \bar{M}_4 + \bar{M}_1 M_2 M_3 \bar{M}_4 + \\ + \bar{M}_1 M_2 M_3 M_4 + M_1 M_2 M_3 M_4 + \bar{M}_1 \bar{M}_2 M_3 \bar{M}_4;$$

$$Q_6 = \bar{M}_1 M_2 \bar{M}_3 \bar{M}_4 + M_1 M_2 \bar{M}_3 \bar{M}_4 + M_1 \bar{M}_2 M_3 \bar{M}_4 + M_1 M_2 M_3 \bar{M}_4 + \\ + \bar{M}_1 M_2 M_3 \bar{M}_4 + \bar{M}_1 \bar{M}_2 M_3 \bar{M}_4 + \bar{M}_1 \bar{M}_2 M_3 M_4 + \bar{M}_1 M_2 M_3 M_4 + M_1 M_2 M_3 M_4;$$

$$Q_7 = \bar{M}_1 M_2 \bar{M}_3 \bar{M}_4 + M_1 M_2 \bar{M}_3 \bar{M}_4 + M_1 \bar{M}_2 \bar{M}_3 \bar{M}_4 + M_1 \bar{M}_2 M_3 \bar{M}_4 + \\ + M_1 M_2 M_3 \bar{M}_4 + \bar{M}_1 \bar{M}_2 M_3 M_4 + \bar{M}_1 M_2 M_3 M_4 + M_1 M_2 M_3 M_4;$$

$$Q_8 = \bar{M}_1 M_2 \bar{M}_3 \bar{M}_4 + M_1 \bar{M}_2 \bar{M}_3 \bar{M}_4 + M_1 \bar{M}_2 M_3 \bar{M}_4 + \bar{M}_1 M_2 M_3 \bar{M}_4 + \\ + \bar{M}_1 \bar{M}_2 M_3 \bar{M}_4 + \bar{M}_1 \bar{M}_2 M_3 M_4 + \bar{M}_1 M_2 M_3 M_4 + M_1 M_2 M_3 M_4;$$

$$Y_1 = \bar{M}_1 M_2 \bar{M}_3 \bar{M}_4 + M_1 M_2 M_3 \bar{M}_4 + \bar{M}_1 M_2 M_3 \bar{M}_4 + \bar{M}_1 \bar{M}_2 M_3 \bar{M}_4 + \\ + \bar{M}_1 M_2 M_3 M_4 + M_1 M_2 M_3 M_4;$$

$$Y_2 = M_1 \bar{M}_2 \bar{M}_3 \bar{M}_4 + M_1 \bar{M}_2 M_3 \bar{M}_4 + M_1 M_2 M_3 \bar{M}_4 + \bar{M}_1 M_2 M_3 \bar{M}_4 + \\ + \bar{M}_1 \bar{M}_2 M_3 \bar{M}_4 + \bar{M}_1 M_2 M_3 M_4 + M_1 M_2 M_3 M_4.$$

Після спрощення виразів отримуємо вирази:

$$Q_4 = \bar{M}_1 M_2 \bar{M}_3 \bar{M}_4 + M_1 \bar{M}_2 \bar{M}_3 \bar{M}_4 + \bar{M}_1 \bar{M}_2 M_3 \bar{M}_4 + \bar{M}_1 M_2 M_3 M_4;$$

$$Q_5 = M_1 \bar{M}_2 \bar{M}_4 + M_2 M_3 M_4 + \bar{M}_1 M_2 \bar{M}_4 + \bar{M}_1 \bar{M}_2 M_3 \bar{M}_4;$$

$$Q_6 = M_2 \bar{M}_4 + M_3 \bar{M}_4 + M_2 M_3 + \bar{M}_1 M_3;$$

$$Q_7 = M_2 \bar{M}_3 \bar{M}_4 + M_1 \bar{M}_2 \bar{M}_4 + \bar{M}_1 M_3 M_4 + M_1 M_2 M_3;$$

$$Q_8 = \bar{M}_1 \bar{M}_2 M_3 + M_1 \bar{M}_2 \bar{M}_4 + \bar{M}_1 M_2 \bar{M}_4 + M_2 M_3 M_4;$$

$$Y_1 = M_2 M_3 + \bar{M}_1 M_3 \bar{M}_4 + \bar{M}_1 M_2 \bar{M}_4;$$

$$Y_2 = M_2 M_3 + M_1 \bar{M}_2 \bar{M}_4 + \bar{M}_1 M_3 \bar{M}_4.$$

3) Записуємо отриманні рівняння у контролер (у вигляді релейно-контактної схеми):

m3	—	m4	—	--	—	(M5
m2	—	M3	—	--	—	(M6
M2	—	M3	—	--	—	(M7
m1	—	M4	—	--	—	(M8
M1	—	m4	—	--	—	(M9
m1	—	m4	—	--	—	(MA
M1	—	M4	—	--	—	(MB
M2	—	M4	—	--	—	(MC
T1	—	M2	—	M5	┌	↑M1
T9	—	M4	—	M7	└	
I2	┌	m2	—	M5	┌	↓M1
	└	m4	—	M6	└	
		M2	—	M5	└	
T5	—	M7	—	m4	└	
M2	—	m3	—	M4	└	
I1	—	m1	—	M5	┌	↑M2
T4	—	M3	—	M9	└	
T8	—	M3	—	M8	└	
m3	—	MB	—	--	└	
I2	—	m1	—	M5	┌	↓M2
T2	—	M1	—	M5	└	
T6	—	M3	—	MA	└	
TA	—	M3	—	MB	└	
T3	—	m2	—	M9	—	↑M3
I2	┌	m2	—	MA	┌	↓M3
	└	M2	—	MA	└	
		M2	—	MC	└	
		M2	—	M8	└	
M1	—	m2	—	M4	└	
T7	—	m1	—	M6	—	↑M4
I2	—	m1	—	M6	└	↓M4
m1	—	M2	—	m3	└	

m1	—	M2	—	M5	—	(T1
M1	—	M2	—	M5	—	(T2
M1	—	m2	—	M5	—	(T3
M1	—	m4	—	M6	—	(T4
M1	—	m4	—	M7	—	(T5
m1	—	m4	—	M7	—	(T6
m1	—	m4	—	M6	—	(T7
m1	—	M4	—	M6	—	(T8
m1	—	M4	—	M7	—	(T9
M1	—	M4	—	M7	—	(TA
m1	—	M2	—	M5	┌	(Q4
M1	—	m2	—	M5		
m1	—	m4	—	M6		
M1	—	M4	—	M7		
m1	—	m2	—	M3	—	(MD
MD	—	m4	—	—	┌	
M1	—	m2	—	m4		(Q5
M2	—	M3	—	M4		
m1	—	M2	—	m4		
M2	—	m4	—	--	┌	(Q6
M3	—	m4	—	--		
M2	—	M3	—	--		
m1	—	M3	—	--		
M2	—	m3	—	m4	┌	(Q7
M1	—	m2	—	m4		
m1	—	M3	—	M4		
M1	—	M2	—	M3		
m1	—	M3	—	--	┌	(Q8
M1	—	m2	—	m4		
m1	—	M2	—	m4		
M2	—	M3	—	M4		
M2	—	M3	—	--	┌	(Y1
m1	—	M3	—	m4		
m1	—	M2	—	m4		
M2	—	M3	—	--	┌	(Y2
M1	—	m2	—	m4		
m1	—	M3	—	m4		

### 15.5. Зміст звіту

- 1) Умова роботи схеми.
- 2) Логічний синтез та блок-схема алгоритму роботи.
- 3) Адресування входів, виходів, таймерів, комірок пам'яті.
- 4) Програма на мові LD.

5) Висновок про відповідність роботи заданим умовам.

### **15.6. Контрольні запитання**

1. Описати методику синтезу задач методом графопереходів.
2. Чому у семисегментних індикаторів на лабораторному стенді не підключаються ніжки 5?
3. Навіщо використовується блок живлення (10 В)?
4. Які позначення використовуються у контролері?
5. Чим відрізняють виходи контролера Q1-Q8 від виходів Y1-Y4?

## Лабораторна робота № 16

### ДОСЛІДЖЕННЯ СИСТЕМИ АВТОМАТИЗАЦІЇ СТРІЧКОВОГО КОНВЕЄРА

Тривалість лабораторної роботи – 4 год.

Тривалість домашньої роботи – 4 год.

**Мета роботи** – отримати практичні навички роботи з логічним контролером та перетворювачем частоти в умовах керування стрічковим конвеєром

#### 16.1 Основні теоретичні відомості

##### 16.1.1 Опис лабораторної установки

Стенд являє собою діючий макет транспортної системи з стрічковим конвеєром. Установка складається з двох самостійних вузлів з барабанами, головного з механізмом приводу і хвостового, переміщення якого використовується для натягу робочого органу. Ескіз конвеєра показаний на рис. 16.1.

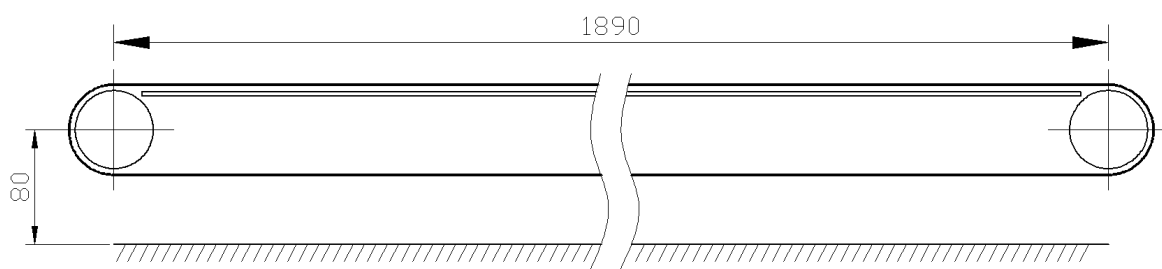


Рисунок 16.1 – Ескіз конвеєрної установки

В якості конвеєрної стрічки для макету використано стрічку для бігових доріжок, вона має достатньо високу міцність, розрахована на роботу в порівняно важких умовах, може бути виготовлена з будь-якими необхідними розмірами. Складається з текстильної нейлонової основи та зовнішнього шару з цупкого полімеру (рис. 16.2).



*Рисунок 16.2 – Загальний вигляд стрічки*

Для горизонтального розташування вантажної гілки конвеєра був виготовлений новий середній проліт установки. В якості конструкційного матеріалу використано березову фанеру товщиною 10 мм (рис. 16.3). Деталі оброблювались на фрезерному ЧПК верстаті, що забезпечило їх високу розмірну точність. Після складання нові деталі було покрито шаром нітрофарби для захисту від вологи та надання декоративного вигляду.



*Рисунок 16.3 – Вигляд прольоту конвеєрної установки*

Для використання в якості давачів положення вантажу безконтактних сенсорів моделі E18-D80NK, були розроблені та виготовлені спеціальні кронштейни двох видів для їх кріплення на установку. Різновид, зображений на рис. 16.4, використовується для кріплення давачів у крайніх кінцевих позиціях конвеєра, на рис. 16.5 – для кріплення на середині прольоту конвеєра. Два середніх кронштейни можна переміщувати, що дозволяє збільшити різноманітність задач, виконуваних на стенді. Наприклад, можна втілити

алгоритм керування, який буде відрізняти короткі вантажі від довгих (довгі вантажі викличуть спрацювання обох датчиків, рознесених на деяку відстань, див. рис. 16.6).

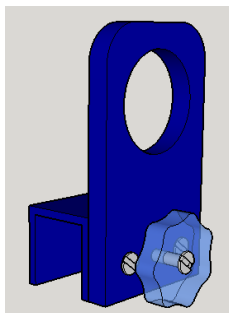


Рисунок 16.4 – Кронштейн для кріплення датчиків на середині конвеєра

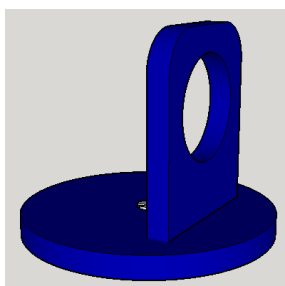


Рисунок 16.5 – Кронштейн для кріплення датчиків у крайніх точках конвеєра

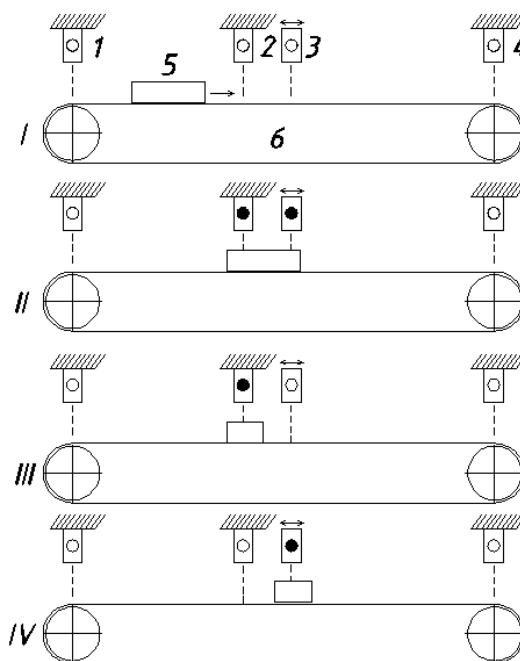


Рисунок 16.6 – Приклад алгоритму роботи сенсора довжини вантажу на датчиках з регульованим положенням: 1-4 – оптичні датчі, 5 – вантаж, 6 – конвеєр

На діаграмі II довгий вантаж при переміщенні викликає одночасне спрацювання сенсорів 2 і 3. На діаграмі III показано, що короткий вантаж не може викликати спрацювання обох давачів.

Оскільки напруга живлення нових давачів – 5 В постійного струму, а існуюча схема керування станда працює з сигналами 24 В постійного струму, для інтеграції сенсорів в існуючу електричну схему станда необхідне використання схеми-перетворювача сигналів. Вона являє собою набір сигнальних реле, котушки яких керуються давачами, а контакти під'єднані до відповідних входів програмованого контролера, та імпульсного перетворювача напруги, який перетворює напругу мережі станда 24 В в напругу живлення давачів і котушок реле 5В. Зовнішній вигляд готової плати наведено на рисунку 16.7. Принципова схема наведена в складі загальної електричної схеми станду в графічних додатках до проекту.

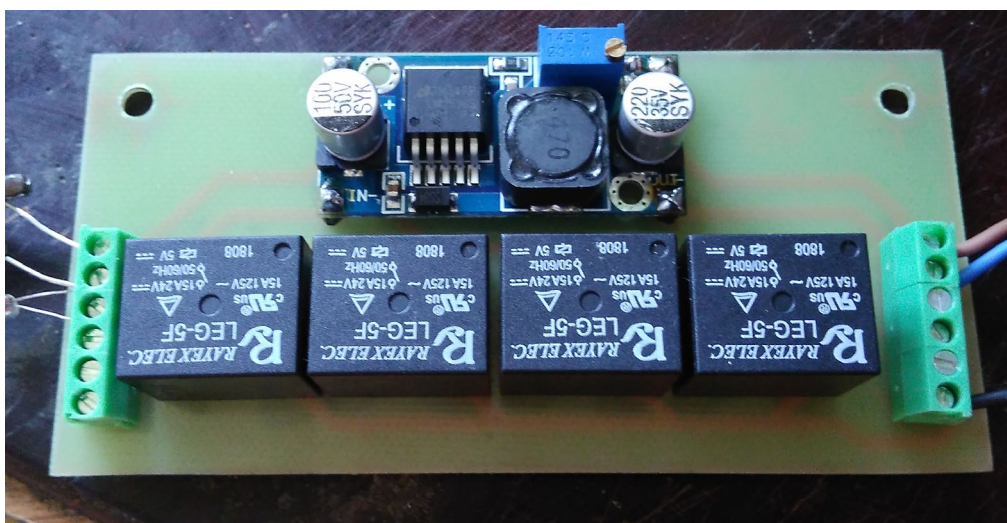
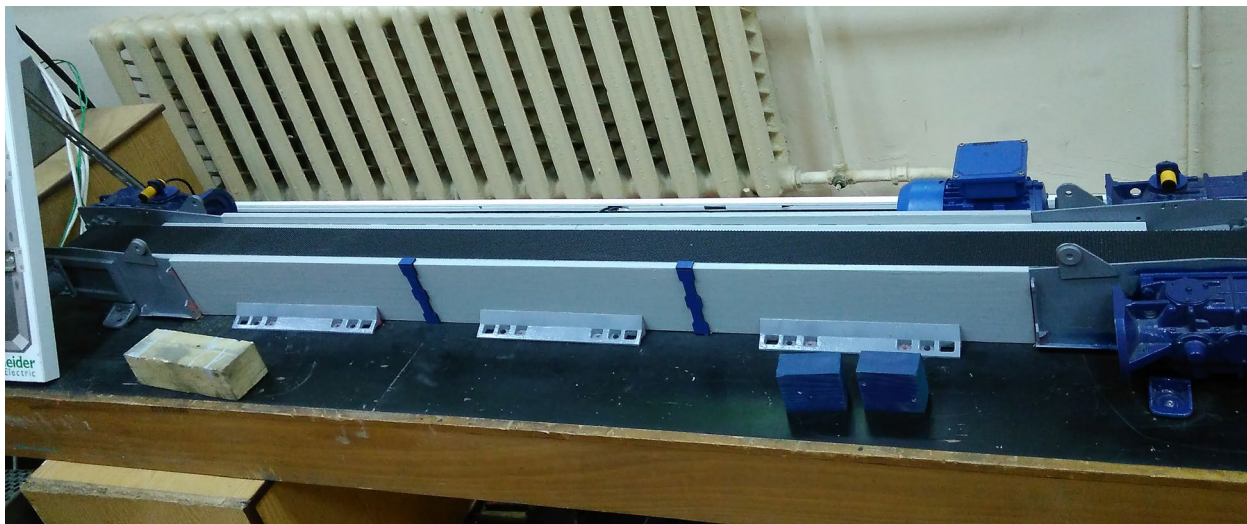


Рисунок 16.7 – Плата перетворювача сигналів

Готова установка з встановленою транспортною стрічкою і оптичними кінцевими вимикачами показана на рисунку 16.8.



*Рисунок 16.8 – Загальний вигляд лабораторної установки*

### **16.1.2 Програмоване логічне реле Zelio**

Програмоване логічне реле Zelio призначене для реалізації систем автоматизації малого рівня інтеграції. Застосовуються в промисловості і комунальному господарстві для використань, від яких не залежить безпека і життя людей.

Основні напрямки використання в промисловості:

1) Автоматизація невеликих виробничих ліній обробки, складання і пакування продукції.

2) Децентралізована автоматизація допоміжного обладнання в великих і середніх виробничих циклах текстильної промисловості, при виготовленні виробів з пластмас, переробленні вторинних матеріалів та ін.

3) Автоматизація сільськогосподарських машин (насосні установки, конвеєри овочебаз, теплиці, системи роздачі кормів та ін.)

В комунальному господарстві і в будинках:

1) Автоматизація шлагбаумів, воріт, систем контролю доступу і освітлення.

2) Автоматизація вентиляції і систем кондиціонування повітря.

Програмування логічного реле Zelio SR2 B122BD виконується на базі візуальних мов програмування (Ladder Diagram programming), що значно

спрощує роботу спеціалістів по налаштуванні систем автоматизації, та може здійснюватися інженерами-електриками після короткого інструктажу.

Програмування може здійснюватись:

1) Автономно за допомогою вбудованого графічного інтерфейсу програмованого реле.

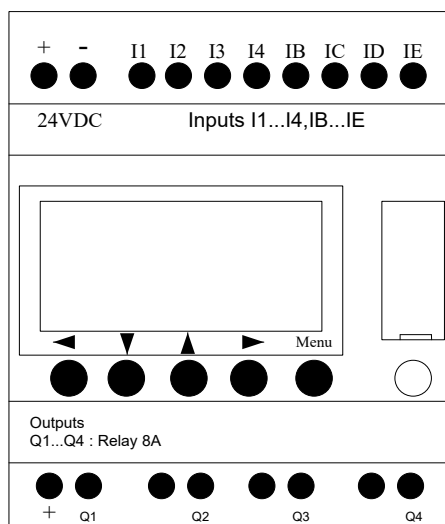
2) Через персональний комп'ютер за допомогою візуального середовища програмування "Zelio Soft 2".

Програмування з комп'ютера можна здійснювати як на мові східчастих діаграм (LD), так і на мові функціональних блок-схем (FBD). На мові LD можна виконати програму за принципами побудови класичних схем релейної автоматизації з віртуальними контактами, котушками і змінними.

#### **Функції ПЛК Zelio SR2 B122BD:**

- 16 Текстових функціональних блоків.
- 16 Функціональних блоків затримки часу, кожному з яких можна задати параметри 11 різних типів від 1/10 секунди до 9999 годин )
- 16 Функціональних блоків лічильник прямого/зворотного відліку від 0 до 32767
- 1 Швидкий лічильник (1 кГц)
- 16 Функціональних блоків аналогового компаратора
- 8 Функціональних блоків годинників, кожен з 4 каналами
- 28 реле керування
- 8 компараторів-лічильників

Зовнішній вигляд програмованого логічного реле Zelio logic SR2B122BD зображено на рисунку 16.9.



*Рисунок 16.9* – Зовнішній вигляд програмованого логічного реле Zelio logic SR2B122BD

### 16.1.3 Оптичні давачі E18-D80NK

Оптичні кінцеві перемикачі моделі E18-D80NK є доцільною заміною традиційних кінцевих перемикачів у випадках, коли механічний контакт відстежуваного об'єкту та штанги перемикача неможливі (велика відстань, великий діапазон руху об'єкта) або недопустимий (крихкий чи небезпечний об'єкт).



*Рисунок 16.10* – Зовнішній вигляд давача

Конструктивно, давач представляє собою поєднані в завершену технологічну одиницю модульований інфрачервоний випромінювач,

інфрачервоний оптичний сенсор та схему керування для налаштування чутливості давача і транзисторним ключем, виконаним за схемою «відкритий колектор». Функціональну схему з прикладом підключення показано на рисунку 16.11, технічні параметри наведено в таблиці 16.1.

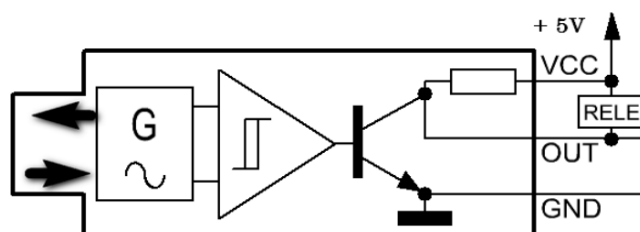


Рисунок 16.11 – Функціональна схема давача

Таблиця 16.1 – Технічні параметри давача

Напруга живлення, В	5
Струм споживання, мА	7
Діапазон чутливості давача, см	3-80
Максимально допустимий струм навантаження, мА	200
Габарити, мм	50x18x18

Схема електрична принципова лабораторної установки показана на рис. 16.12, а зовнішній вигляд та функціональна схема на рис. 16.13.

**Слід зауважити, що 4 безконтактні давачі розміщені вздовж конвеєрної лінії, а їх замикання відповідає логічній 1, що подається на відповідний вхід контролера згідно рис. 16.12. Перетворювач частоти, так само, як і у лабораторній роботі № 11 працює у режимі «цифрового потенціометра», де виходи контролера Q1 та Q2 відповідають за прямий та зворотній напрямок обертання двигуна відповідно, а виходи Q3 та Q4 за прискорення та сповільнення відповідно.**

**Це зауваження необхідно використовувати при синтезі та при адресації вхідних/вихідних сигналів.**

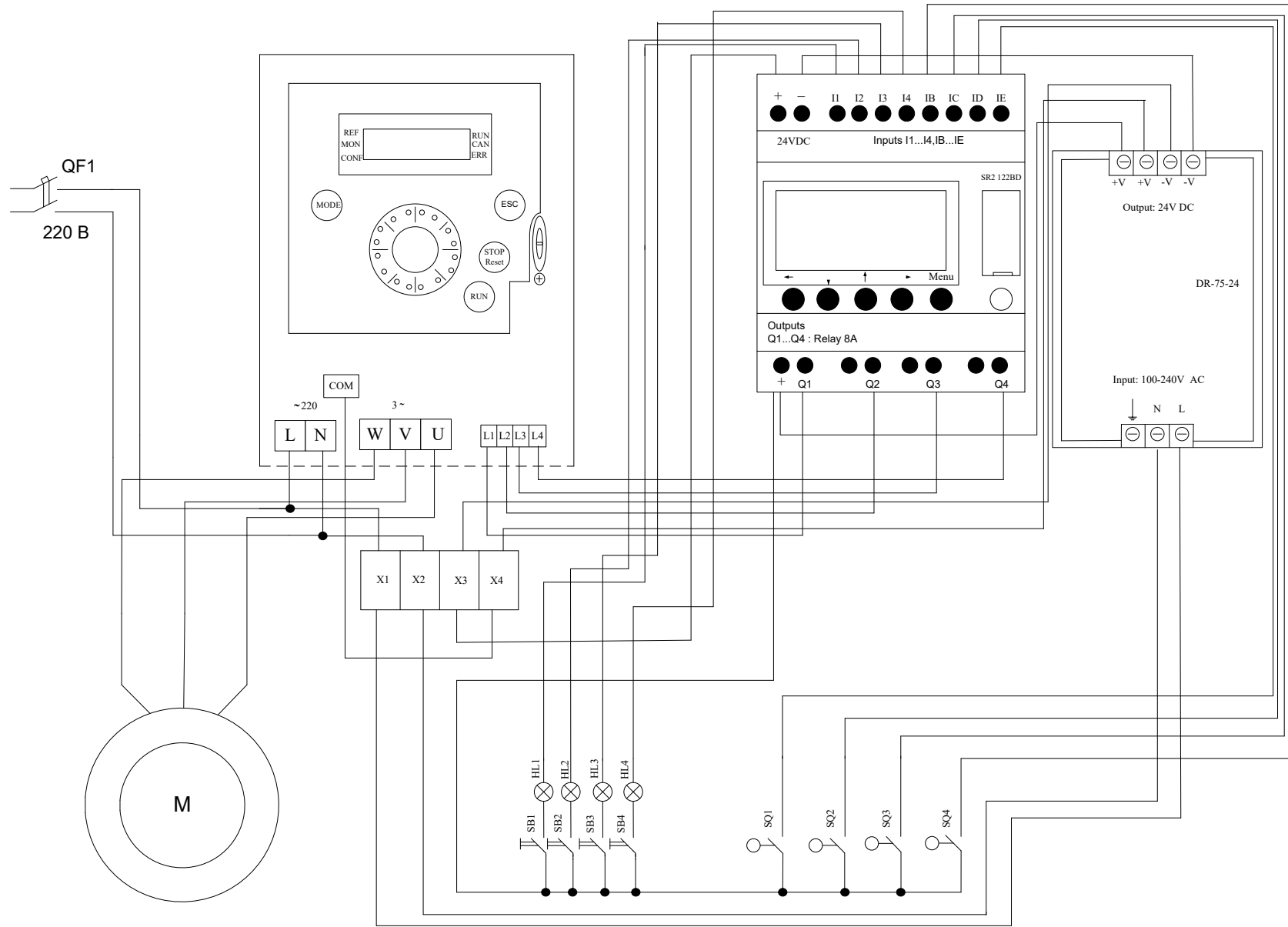
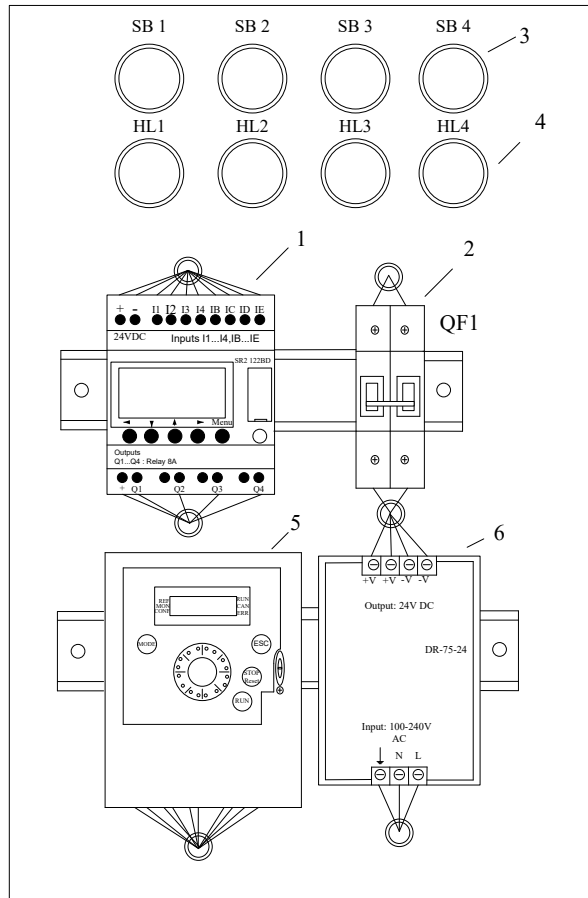
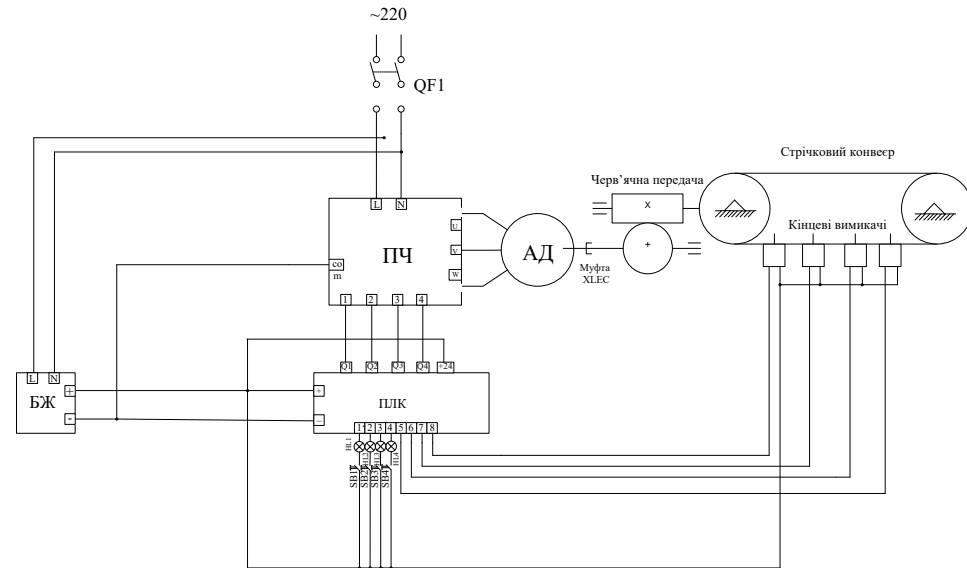


Рисунок 16.12 – Схема електрична принципова лабораторної установки



Зовнішній вигляд панелі керування

- 1-програмоване логічне реле Zelio
- 2-автоматичний вивикач
- 3-кнопки з саблокуванням
- 4-світлодіодні індикатори
- 5-Перетворювач частоти Altivar 312
- 6-Блок живлення



Функціональна схема лабораторної установки

Рисунок 16.13 – Зовнішній вигляд та функціональна схема лабораторної установки

## 16.2. Програма роботи

- 1) Виконати логічний синтез за умовами роботи схеми для заданого варіанту, поданого у таблиці 16.2 (**виконується вдома, під час СРС**).
- 2) Скласти програму на мові LD згідно отриманих логічних виразів.
- 3) Записати програму в контролер.
- 4) Перевірити відповідність роботи заданим умовам (**викладач своїм підписом засвідчує правильність роботи програми**).
- 5) Отримати у викладача індивідуальне завдання, виконати синтез, написати програму, записати її у контролер та перевірити її працездатність.
- 6) Оформити звіт та зробити висновки про результати роботи.

Таблиця 16.2 – Розподілення завдань за бригадами

Бригада	Номер завдання, яке визначає умову роботи
1	1; індивідуальне
2	2; індивідуальне
3	3; індивідуальне
4	4; індивідуальне
5	5; індивідуальне
6	6; індивідуальне

### Умови роботи схем

1. Конвеєр працює за наступним алгоритмом: запуск установки здійснюється кнопкою «ПУСК», при цьому двигун розганяється до номінальної частоти обертання за 6 секунд. Рух вантажу на конвеєрі обмежується кінцевими перемикачами SQ1, SQ2 та SQ4. При досягненні вантажу SQ1 конвеєр зупиняється на 2 секунди, потім продовжує працювати, доки вантаж не досягне SQ2. При досягненні SQ2 конвеєр знову зупиняється. Для продовження роботи необхідно натиснути кнопку «Продовження», при цьому вантаж іде до SQ4, реверсується, прямує до SQ1 і зупиняється. При натисненні кнопки «СТОП» у будь-якому стані конвеєр уповільнюється до зупинки за 4 секунди.

2. Конвеєр працює за наступним алгоритмом: запуск установки здійснюється кнопкою «ПУСК», при цьому двигун розганяється до номінальної

частоти обертання за 3 секунди. Рух вантажу на конвеєрі обмежується кінцевими перемикачами SQ1, SQ2 та SQ4. При досягненні вантажу SQ1 конвеєр зупиняється на 5 секунд, потім продовжує працювати, доки вантаж не досягне SQ2. При досягненні SQ2 конвеєр знову зупиняється на 3 секунди, потім знову продовжує працювати. При досягненні SQ4 вантаж зупиняється. Для продовження роботи необхідно натиснути кнопку «Продовження», при цьому вантаж іде до SQ2, зупиняється на 5 секунд, прямує до SQ1 і зупиняється. При натисненні кнопки «СТОП» у будь-якому стані конвеєр уповільнюється до зупинки за 2,5 секунди.

3. Конвеєр працює за наступним алгоритмом: запуск установки здійснюється кнопкою «ПУСК», при цьому двигун розганяється до номінальної частоти обертання за 2,8 секунди. Рух вантажу на конвеєрі обмежується кінцевими перемикачами SQ1, SQ2, SQ3 та SQ4. При досягненні вантажу SQ1 конвеєр зупиняється на 10 секунд, потім продовжує працювати, доки вантаж не досягне SQ3 (минаючи SQ2). При досягненні SQ3 конвеєр знову зупиняється на 3 секунди, реверсується і повертає вантаж до SQ2, потім знову реверсується і т.д. Для виходу з цього режиму необхідно натиснути кнопку «Продовження», коли вантаж прямує до SQ3 (якщо він прямує до SQ2, ця команда не повинна спрацювати). При досягненні SQ4 конвеєр реверсується, вантаж іде до SQ2, зупиняється на 2 секунди, прямує до SQ1 і зупиняється. При натисненні кнопки «СТОП» у будь-якому стані конвеєр уповільнюється до зупинки за 2 секунди.

4. Конвеєр працює за наступним алгоритмом: запуск установки здійснюється кнопкою «ПУСК», при цьому двигун розганяється до номінальної частоти обертання за 3,2 секунди. Рух вантажу на конвеєрі обмежується кінцевими перемикачами SQ1, SQ2, SQ3 та SQ4. При досягненні вантажу SQ1 конвеєр зупиняється на 8 секунд, потім продовжує працювати, доки вантаж не досягне SQ2. При досягненні SQ2 конвеєр знову зупиняється на 4 секунди, потім знову продовжує працювати. При досягненні SQ4 (минаючи SQ3) конвеєр реверсується і вантаж прямує до SQ2 (минаючи SQ3). При досягненні SQ2 конвеєр знову реверсується, вантаж прямує до SQ3, там знову здійснюється реверс і т.д. Для виходу із цього стану необхідно натиснути

«Продовження», при цьому, в незалежності де знаходився вантаж, він починає рухатися до SQ1 і там зупиняється. При натисненні кнопки «СТОП» у будь-якому стані конвеєр уповільнюється до зупинки за 2,5 секунди.

5. Конвеєр працює за наступним алгоритмом: запуск установки здійснюється кнопкою «ПУСК», при цьому двигун розганяється до номінальної частоти обертання за 4 секунди. Рух вантажу на конвеєрі обмежується кінцевими перемикачами SQ1, SQ2, SQ3 та SQ4. При досягненні вантажу SQ1 конвеєр зупиняється на 8 секунд, потім продовжує працювати, доки вантаж не досягне SQ2. При досягненні SQ2 конвеєр реверсується і вантаж прямує до SQ1. Там він знову реверсується і вантаж прямує до SQ3 (минаючи SQ2). Далі конвеєр знову реверсується і вантаж знову прямує до SQ1 (минаючи SQ2), де він зупиняється. Для продовження необхідно натиснути кнопку «Продовження», при цьому вантаж прямує до SQ4 (минаючи SQ1 та SQ2), зупиняється на 10 секунд, потім реверсується і вантаж повертається до SQ1, де остаточно зупиняється. При натисненні кнопки «СТОП» у будь-якому стані конвеєр уповільнюється до зупинки за 2,1 секунди.

6. Конвеєр працює за наступним алгоритмом: запуск установки здійснюється кнопкою «ПУСК», при цьому двигун розганяється до номінальної частоти обертання за 2 секунди. Рух вантажу на конвеєрі обмежується кінцевими перемикачами SQ1, SQ3 та SQ4. При досягненні вантажу SQ1 конвеєр зупиняється на 5 секунд, потім продовжує працювати, доки вантаж не досягне SQ3 (минаючи SQ2). При досягненні SQ3 конвеєр зупиняється на 6 секунд, реверсується і вантаж прямує до SQ1 (минаючи SQ2). Там він знову зупиняється на 6 секунд, реверсується і вантаж знову прямує до SQ3 (минаючи SQ2). Далі конвеєр зупиняється і чекає на команду «Продовжити». Якщо цю команду не надати за 20 секунд, конвеєр реверсується, вантаж повертається до SQ1 і зупиняється. Якщо команда «Продовжити» надійде до того, як пройшло 20 секунд, вантаж спочатку рухається до SQ4, зупиняється там на 15 секунд, реверсується і повертається до SQ1. При натисненні кнопки «СТОП» у будь-якому стані конвеєр уповільнюється до зупинки за 2,1 секунди.

### 16.3 Методичні вказівки до виконання роботи

Розглянемо приклади виконання завдань.

*Приклад 1.* Конвеєр працює за наступним алгоритмом: запуск установки здійснюється кнопкою А, при цьому двигун розганяється до номінальної частоти обертання за три секунди. Рух вантажа на конвеєрі обмежується кінцевими перемикачами SQ1 та SQ4. При досягненні деталлю променя вимикачів, здійснюється реверс конвеєра. При натисненні кнопки В конвеєр уповільнюється до зупинки за 3 секунди.

Програма для рішення задачі синтезується за допомогою графопереходу:

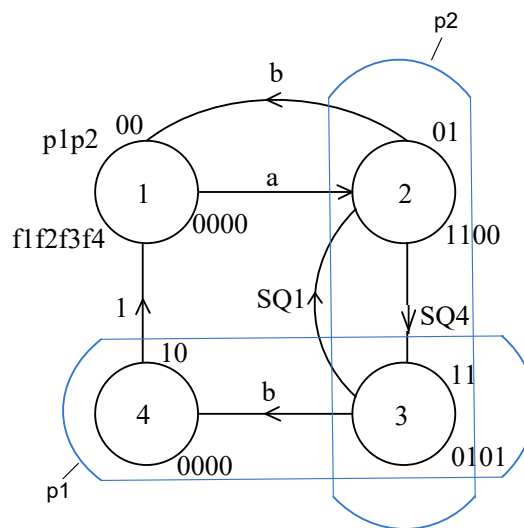


Рисунок 16.14 – Графоперехід до задачі 1

За яким складаються логічні рівняння:

$$\begin{aligned}
 Sp_1 &= SQ4 \cdot p_2; & Sp_2 &= a \cdot \overline{p_1}; \\
 Rp_1 &= SQ1 \cdot p_1 + \overline{p_2}; & Rp_2 &= b \cdot \overline{p_1} + b \cdot p_1 = b; \\
 f_1 &= \overline{p_1} \cdot p_2; & f_2 &= p_1 \cdot p_2; \\
 f_3 &= \overline{p_1} \cdot p_2 + p_1 \cdot p_2 = p_2; & f_4 &= 0.
 \end{aligned}$$

За отриманими рівняннями складається програма на мові LD:

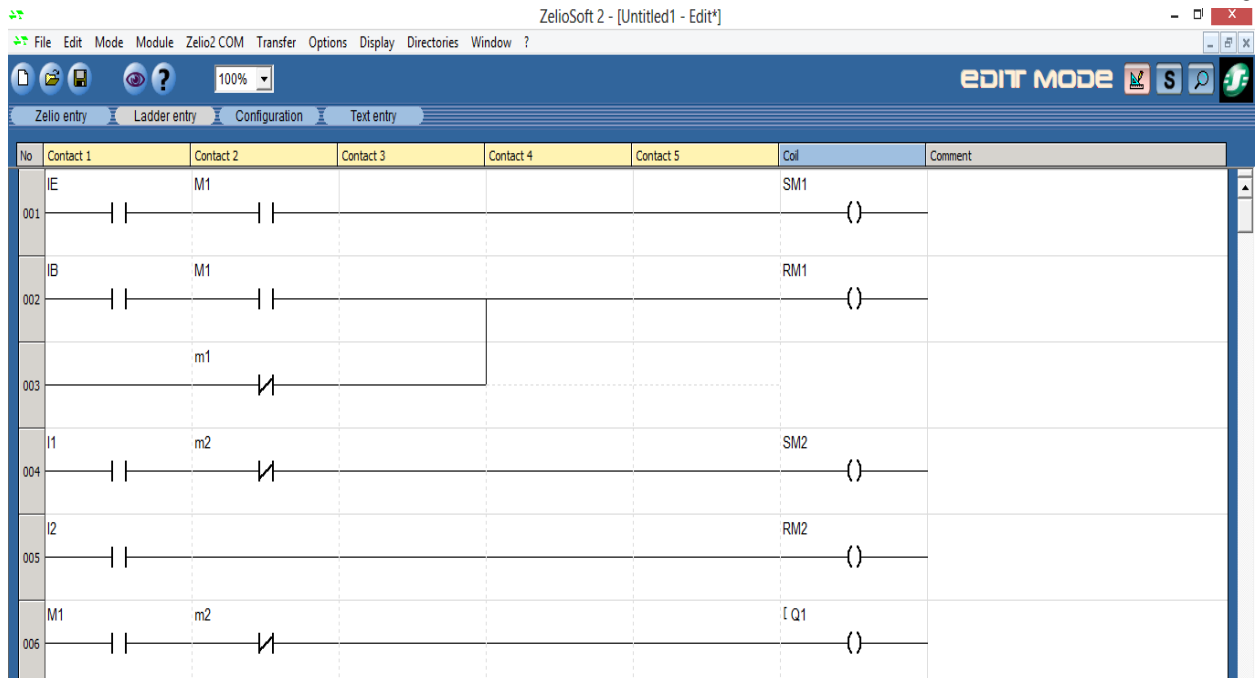


Рисунок 16.15 – Вирішення задачі в середовищі Zelio soft 2.

Приклад 2. Конвеєр працює за наступним алгоритмом: запуск установки відбувається кнопкою А, при цьому двигун розганяється за 3 секунди до номінальної частоти обертання. Рух вантажа конвеєра обмежується кінцевими перемикачами SQ1 та SQ2. При досягненні вантажем променя перемикачів конвеєр зупиняється, стоїть 20 секунд і реверсується. Зупинка здійснюється кнопкою В. За умовами задачі складаємо графо перехід.

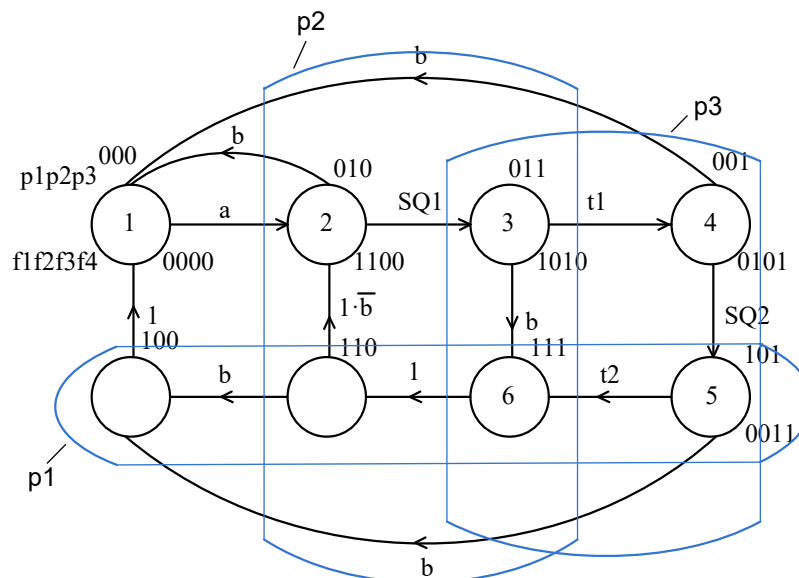


Рисунок 16.16 – Графоперехід до задачі 2

$$T_1 = \bar{p}_1 p_2 p_3$$

$$T_2 = p_1 \bar{p}_2 p_3$$

$$f_1 = \bar{p}_1 p_2 \bar{p}_3 + \bar{p}_1 p_2 p_3$$

$$f_2 = \bar{p}_1 p_2 \bar{p}_3 + p_1 p_2 \bar{p}_3$$

$$f_3 = p_1 \bar{p}_2 p_3 + p_1 \bar{p}_2 \bar{p}_3$$

$$f_4 = p_1 p_2 \bar{p}_3 + p_1 \bar{p}_2 p_3$$

$$Sp_1 = SQ2 \cdot \bar{p}_2 p_3 + bp_2 p_3$$

$$Rp_1 = \bar{p}_2 \bar{p}_3 + \bar{b} \bar{p}_3 p_2$$

$$Sp_2 = a \bar{p}_1 \bar{p}_3 + t_2 p_1 p_3$$

$$Rp_2 = t_1 \bar{p}_1 p_3 + bp_1 \bar{p}_3 + b \bar{p}_1 \bar{p}_3$$

$$Sp_3 = SQ1 \bar{p}_1 p_2$$

$$Rp_3 = p_1 p_2 + b \bar{p}_1 \bar{p}_2 + bp_1 \bar{p}_2$$

$$p_1 = M1, p_2 = M2, p_3 = M3$$

На основі отриманих рівнянь складаємо програму на мові LD

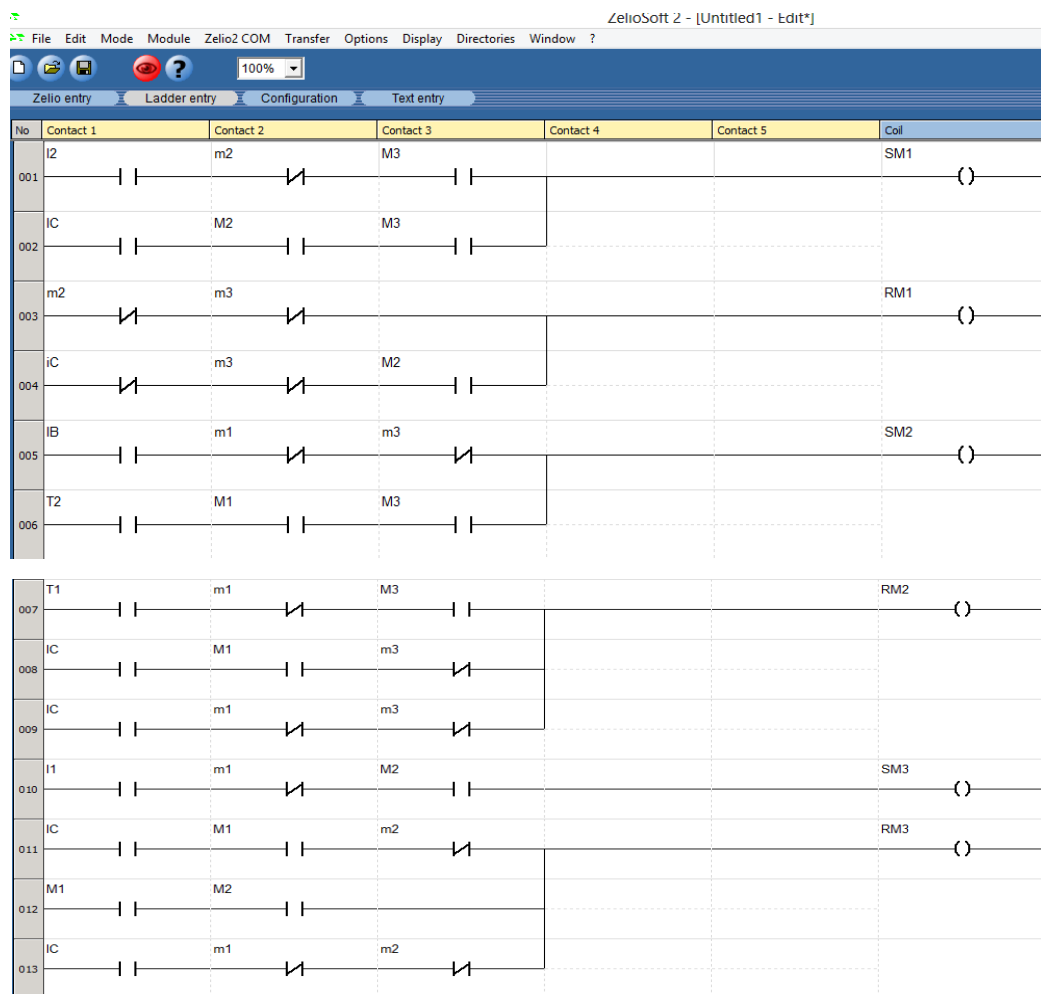


Рисунок 16.17 – Реалізація програми в середовищі Zelio soft 2

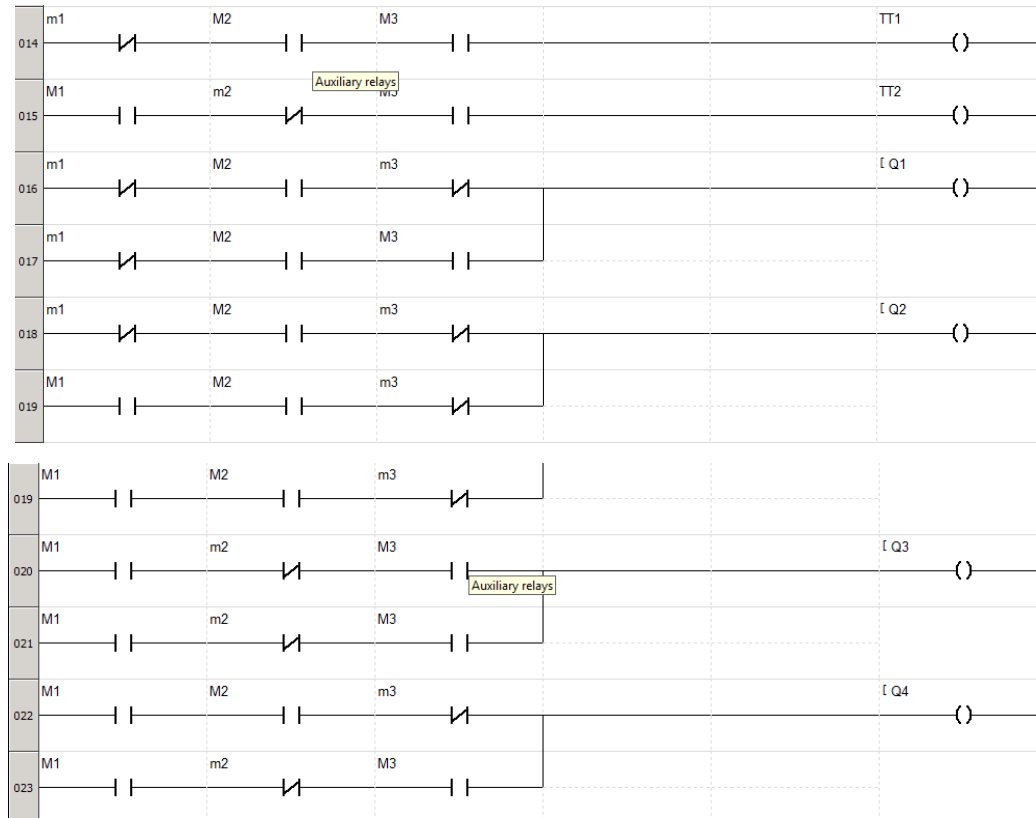


Рисунок 16.18 – Реалізація програми в середовищі Zelio soft 2 (продовження)

## 16.5. Зміст звіту

- 1) Умова роботи схеми.
- 2) Логічний синтез та блок-схема алгоритму роботи.
- 3) Адресування входів, виходів, таймерів, комірок пам'яті.
- 4) Програма на мові LD.
- 5) Висновок про відповідність роботи заданим умовам.

## 16.6. Контрольні запитання

1. Яка причина використання схеми-перетворювачів сигналів?
2. На яких мовах можливо програмувати у програмному середовищі ZelioSoft?
3. Як працює перетворювач частоти?
4. Навіщо необхідна черв'ячна передача?
5. Описати входи контролера та їх особливості використання.

## Лабораторна робота № 17

### ЕАЛІЗАЦІЯ СИСТЕМ АВТОМАТИЗАЦІЇ НА ОСНОВІ КОНТРОЛЕРА ALFA

Тривалість лабораторної роботи – 4 год.

Тривалість домашньої роботи – 4 год.

**Мета роботи** – ознайомитись з принципами розробки і тестування систем автоматизації за допомогою програмного середовища «ALPHA Programming» на практиці освоїти процес створення та налагодження програм на мові функціональних блоків (FBD).

#### 17.1. Основні теоретичні відомості

##### 17.1.1. Програмовані логічні контролери

Програмовані логічні контролери (ПЛК) – це модуль зі змінною (програмованою) логікою роботи, який керує якоюсь машиною або пристроєм.

ПЛК, функціональна схема якого зображена на рис. 17.1, складається з:

1. Блоку узгодження входів на певну кількість вхідних сигналів певного типу.
2. Блоку підсилення виходів на певну кількість каналів (підсилення сигналів, за допомогою яких здійснюється вплив на об'єкт керування).
3. Мікропроцесорного модуля з операційною системою, яка зчитує вхідні сигнали (їх значення на вході), розраховує необхідні за алгоритмом роботи стани вхідних сигналів та передає їх на вихідні лінії.

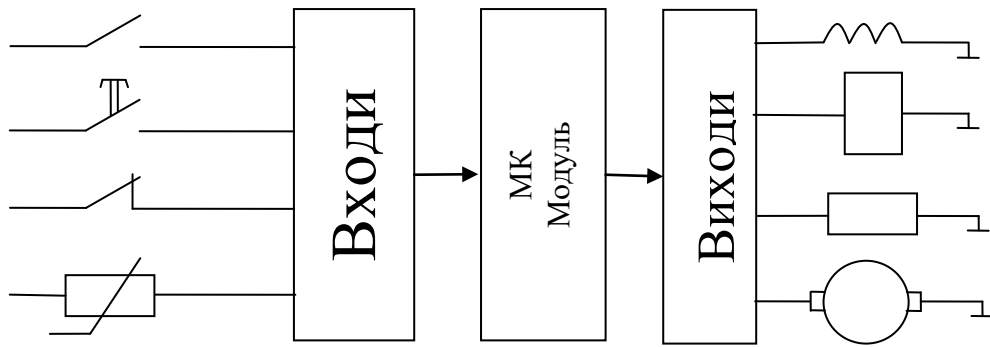


Рисунок 17.1 – функціональна схема ПЛК.

На початку програмовані логічні контролери (ПЛК) розроблялися для реалізації алгоритмів логічного керування, з метою заміни релейно-контактних схем, зібраних на дискретних компонентах, реле, лічильниках і таймерах. Принципова відмінність ПЛК від таких схем полягає в тому, що всі його функції реалізуються програмно. На одному контролері можна реалізувати схему, еквівалентну сотням чи навіть тисячам логічних елементів. При цьому надійність роботи системи майже не залежить від її складності.

ПЛК має набір вхідних і вихідних сигналів. Зміна стану вхідних сигналів призводить до зміни вихідних сигналів у відповідності до заздалегідь запрограмованого алгоритму. До входів ПЛК, як правило, підключаються датчики, кнопкові перемикачі та інші пристрої, що можуть надавати інформацією про стан системи. До виходів ПЛК зазвичай підключаються реле, що виконують функції керування, виконавчі механізми та інші об'єкти керування.

Перевагою застосування ПЛК є гнучкість, універсальність, висока надійність і помірна ціна. Застосування ПЛК дає можливість реалізувати додаткові функції, не ускладнюючи і не збільшуючи вартість готової продукції, які допоможуть повніше використовувати можливості устаткування.

Умовно всі ПЛК можна поділити на три групи: контролери початкового рівня, середнього рівня і на контролери, що призначені для виконання складних задач автоматизації.

Контролери для виконання задач автоматизації початкового рівня складності відносяться до нижнього цінового діапазону, мають центральний процесор невисокої швидкодії, об'єм пам'яті до 128 Кб, не велику кількість

дискретних входів (8-24) та виходів (6-24), аналогові входи та виходи присутні дуже рідко, а комунікаційні можливості або відсутні, або обмежуються найпростішими інтерфейсами.

Області застосування:

- керування освітлювальними приборами (зовнішнім і внутрішнім освітленням, освітленням вітрин);
- керування комутаційною апаратурою;
- керування технологічним устаткуванням (насосами, вентиляторами, компресорами, пресами);
- системи опалювання і вентиляції;
- системи керування дорожнім рухом;
- автоматизоване керування системами кондиціонування повітря,
- автоматизовані системи поливу рослин;
- відкривання і закривання воріт;
- прості системи безпеки;
- систем розподілу кормів в тваринництві.
- керування підйомниками;
- системи очищення;
- керування деревообробними станками;
- керування ліфтами та підйомниками;
- керування конвеєрними лініями;
- харчова промисловість та ін.

Характерні особливості контролерів початкового рівня:

4. Мають можливість відображати на рідкокристалічному дисплеї поточний стан контролера і різноманітні сервісні повідомлення.
5. Годинник реального часу, який має велику кількість перемикачів, що значно розширює можливості створення часових затримок.
6. Високошвидкісний лічильник 1кГц.

7. Вбудований програмований постійний запам'ятовуючий пристрій, що очищується електричним шляхом, виключає необхідність використовувати акумулятора для зберігання даних.

Стандарт ІЕС-1131-3 передбачає п'ять різних мов, найбільш популярна з яких - мова функціональних блоків і діаграм (FBD). Це графічна мова, яка дозволяє створювати програму по функціональній схемі електронного пристрою, замінюючи позначення реальних елементів програмними функціональними блоками, що реалізують функції елементів програмно.

Програмування контролерів початкового рівня може виконуватись через ПК зі спеціалізованим програмним забезпеченням або за допомогою передньої панелі контролера.

В данній роботі вивчається метод створення і відпрацювання програми на мові FBD за допомогою ПК.

**Послідовність розв'язання задач автоматизації на основі ПЛК.** На початковому етапі необхідно поділити завдання на якомога простіші задачі, потім виконати синтез логічних функцій чи алгоритмів керування для кожної простої задачі одним з методів і намалювати схему з використанням стандартних блоків мови FBD. На кінцевому етапі блок-схеми, отримані в результаті розв'язання простих задач, об'єднуються в одну для розв'язання всього завдання, переносяться в програмне середовище «ALPHA Programming» і тестуються.

### 17.1.2. Синтез логічних функцій

**Приклад.** Виконати синтез схеми, що має три вхідні сигнали  $a_1 \dots a_3$  і два вихідних  $X$  і  $Y$ . Якщо три вхідні сигнали  $a_1 \dots a_3$  дорівнюють одиниці, причому першим надходить  $a_1$ , то  $X = 1$ , а якщо  $a_1$  надходить не першим, то  $Y = 1$ .

Синтез виконуємо за допомогою таблиць переходів. Складемо таблицю для визначення кількості станів.

Таблиця 17.1 – Визначення кількості станів схеми для задачі

Номер стану схеми	Комбінації вхідних сигналів			Значення вихідних сигналів		Примітка
	$a_1$	$a_2$	$a_3$	X	Y	
1	0	0	0	0	0	-
2	0	0	1	0	0	-
3	0	1	0	0	0	-
4	0	1	1	0	0	-
5	1	0	0	0	0	-
6	1	0	1	0	0	$a_1$ - перший
7	1	1	0	0	0	$a_1$ - перший
8	1	1	0	0	0	$a_1$ - не перший
9	1	1	1	1	0	$a_1$ - перший
10	1	1	1	0	1	$a_1$ - не перший
11	1	0	1	0	1	$a_1$ - не перший

Визначивши кількість станів, складемо таблицю переходів.

Таблиця 17.2 – Таблиця переходів

Номер вихідного стану	Наступні стани								Вихідні змінні	
	$a_1$	$a_2$	$a_3$	$a_1$	$a_2$	$a_3$	$a_1$	$a_2$	X	Y
1	<1>	5	-	3	-	-	-	2	0	0
2	1	-	-	-	4	-	11	<2>	0	0
3	1	-	8	<3>	4	-	-	-	0	0
4	-	-	-	3	<4>	10	-	2	0	0
5	1	<5>	7	-	-	-	6	-	0	0
6	-	5	-	-	-	9	<6>	2	0	0
7	-	5	<7>	3	-	9	-	-	0	0
8	-	5	<8>	3	-	10	-	-	0	0
9	-	-	7	-	4	<9>	6	-	1	0
10	-	-	8	-	4	<10>	11	-	0	1
11	-	5	-	-	-	10	<11>	2	0	0

Складемо стиснену таблицю переходів. Стиснена таблиця переходів зображена в таблиці 17.3.

Таблиця 17.3 – Стиснена таблиця переходів

Номери вихідних станів	Наступні стани							
	$a_1$		$a_2$			$a_3$		
1, 5, 6, 7, 9	<1>	<5>	<7>	3	4	<9>	<6>	2
2, 3, 4, 8, 10, 11	1	5	<8>	<3>	<4>	<10>	<11>	<2>

Складемо карти Карно для проміжної змінної і вихідної величини, які наведемо на рис. 17.2. Значення проміжної величини для станів 1, 5, 6, 7, 9 оберемо рівним одиниці, а для станів 2, 3, 4, 8, 10, 11 – нулю.

Користуючись картами Карно, запишемо вирази для проміжної змінної та вихідних сигналів. Отже, маємо:

$$\begin{aligned}
 P &= \bar{a}_2 \bar{a}_3 + a_1 p; \\
 X &= a_2 a_3 p; \\
 Y &= a_1 a_2 a_3 \bar{p}.
 \end{aligned}
 \tag{17.1}$$

У відповідності до синтезованих виразів створимо схему на мові FBD і пронумеруємо блоки на схемі у порядку їх підключення від входу до виходу.

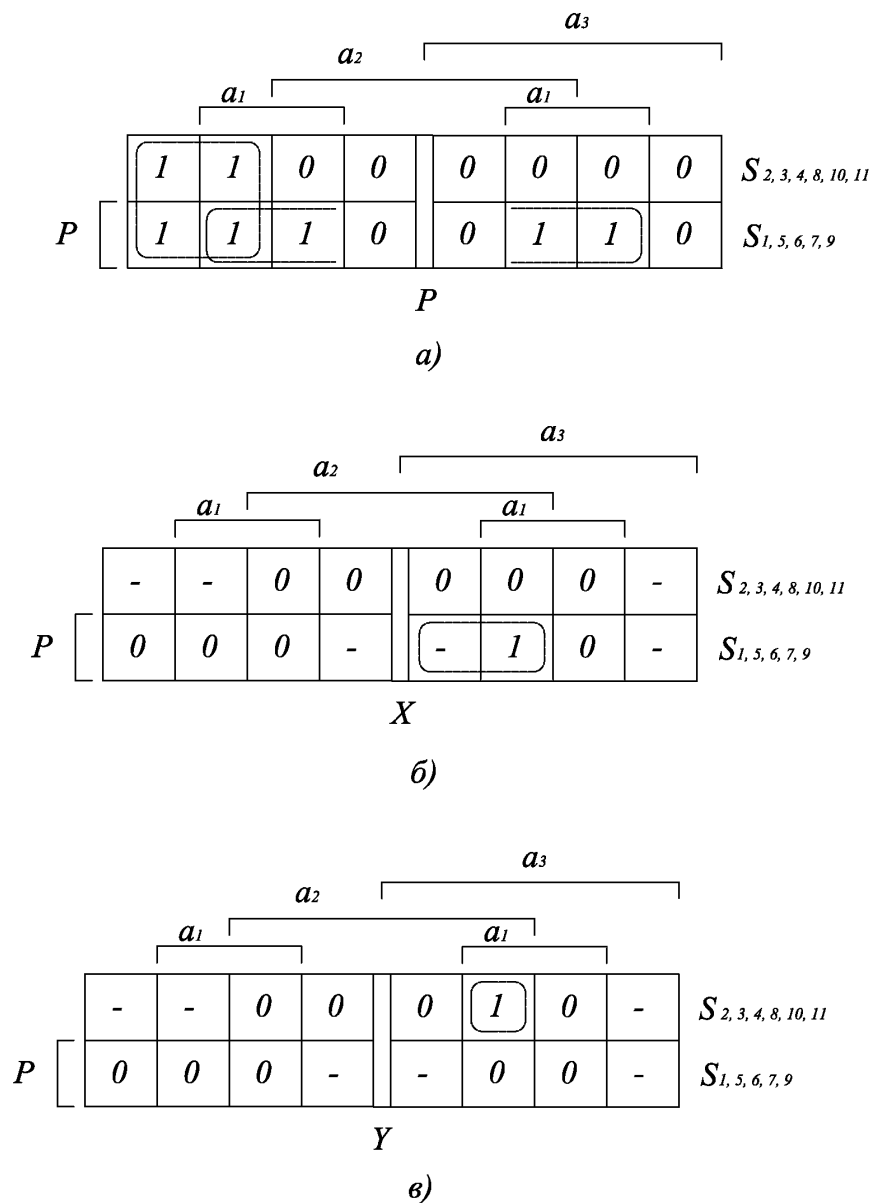


Рисунок 17.2 – Карти Карно: а) для проміжної змінної; б) для вихідного сигналу X; в) для вихідного сигналу Y

### 17.1.3. Складання програми на мові FBD

Програма на стандартній мові FBD складається у вигляді схеми з'єднань типових функціональних блоків на основі попередньо синтезованих логічних виразів.

Складання програми на мові FBD базується на комбінуванні різних функціональних блоків. З метою спрощення процедури програмування функціональні блоки вже мають певні стандартні налаштуваннями, які за допомогою діалогових вікон для кожного функціонального блоку можуть бути встановлені параметри, що необхідні для його адаптації до конкретного

випадку. Програма для ПЛК може включати до 200 блоків, які реалізують логічні функції, тригери, таймери, лічильники, формують імпульси та інше, загалом 38 типів основні з яких наведені нижче.

**Логічні блоки.** Значення виходу логічного блоку визначається результатом логічної дії, яку реалізує функціональний блок над вхідними сигналами. В більшості ПЛК реалізовано 6 типів логічних блоків. Властивості даних блоків і символи, якими вони позначаються на LCD-дисплеї, наведені у таблиці 17.4.

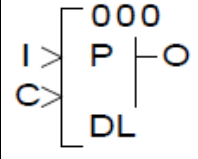
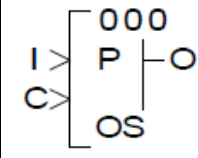
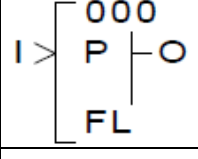
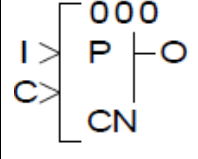
Таблиця 17.4 – Логічні функціональні блоки

Логічна функція	AND	OR	NOT	XOR	NAND	NOR
Зображення логічного блоку						

**Дискретні функціональні блоки** реалізують тригери, лічильники, таймери, формувачі імпульсів та інші різноманітні функції, які часто використовуються в системах автоматичного керування. Деякі функціональні блоки можуть бути налаштовані відповідно до вимог алгоритму. Опис основних дискретних функціональних блоків, їх входи і виходи, налаштувальні параметри, а також їх зображення на LCD-дисплеї, наведено в в таблиці 17.5.

Таблиця 17.5 – Дискретні функціональні блоки

Назва функціонального блоку	Зображення функціонального блоку	Опис
Set/Reset [SR] (Установка/Скид)		Функціональний блок типу Set/reset реалізує функцію тригера, де пріоритет може бути наданий будь-якому з вхідних контактів.

Delay [DL] (Затримка переключення)		Функціональний блок Delay має таймери затримки сигналів «ON» і «OFF». Для кожного з цих таймерів може бути встановлений свій інтервал затримки з кроком 10 мс, 100 мс або 1 с.
One Shot [OS] (Генератор одиничного імпульсу)		Функціональний блок One Shot встановлює вихідний сигнал після надходження вхідного сигналу і утримує його протягом деякого часу. Час утримання вихідного сигналу задається параметром часу і залежить від пріоритету входів.
Flicker [FL] (Тактовий генератор)		Функціональний блок Flicker видає серію імпульсів зміни стану виходу з «OFF» на «ON» і назад.
Counter [CN] (Лічильник)		Функціональний блок Counter збільшує своє поточне значення на одиницю під час надходження сигналу на вхід. Коли поточне значення лічильника досягає заданого значення, стан виходу блоку встановлюється в «ON». Поточне значення лічильника скидається, коли подається сигнал на вхідний контакт очищення «C».

Використовуючи вище наведені доступні логічні та функціональні блоки можна класти програму на мові FBD, що реалізує вираз (17.1) і має вигляд зображений на рис. 17.3.

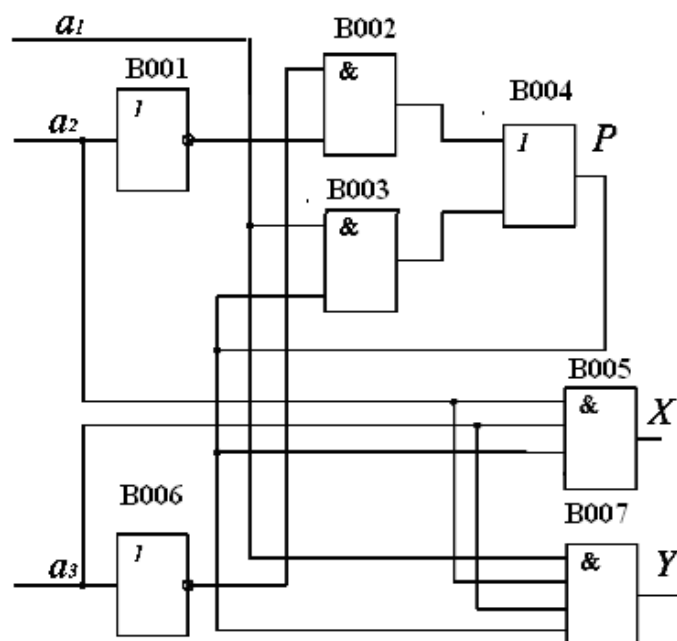


Рисунок 17.3 – Програма на мові FBD

### 17.1.4. Запис і перевірка розробленої програми.

Програмне середовище «ALPHA Programming» використовується для перевірки правильності розробленої програми і її налагодження. Основне вікно програми показано на рис 17.3. Більшу частину робочого вікна займає «Display Manager» – це поле для набору програми, яку ще називають областю з'єднань. Програмне середовище «ALPHA Programming» дозволяє застосовувати при програмуванні 5 наборів функціональних блоків: «IN», «FUNC», «LOGIC», «OUT», «USER» які можна обрати відповідними кнопками на панелі інструментів зліва від вікна з'єднань (на рис. 17.4 позначено 1). Ці набори включають різні типи вхідних сигналів, математичні функції, логічні блоки, функціональні блоки користувача і виходи відповідно.

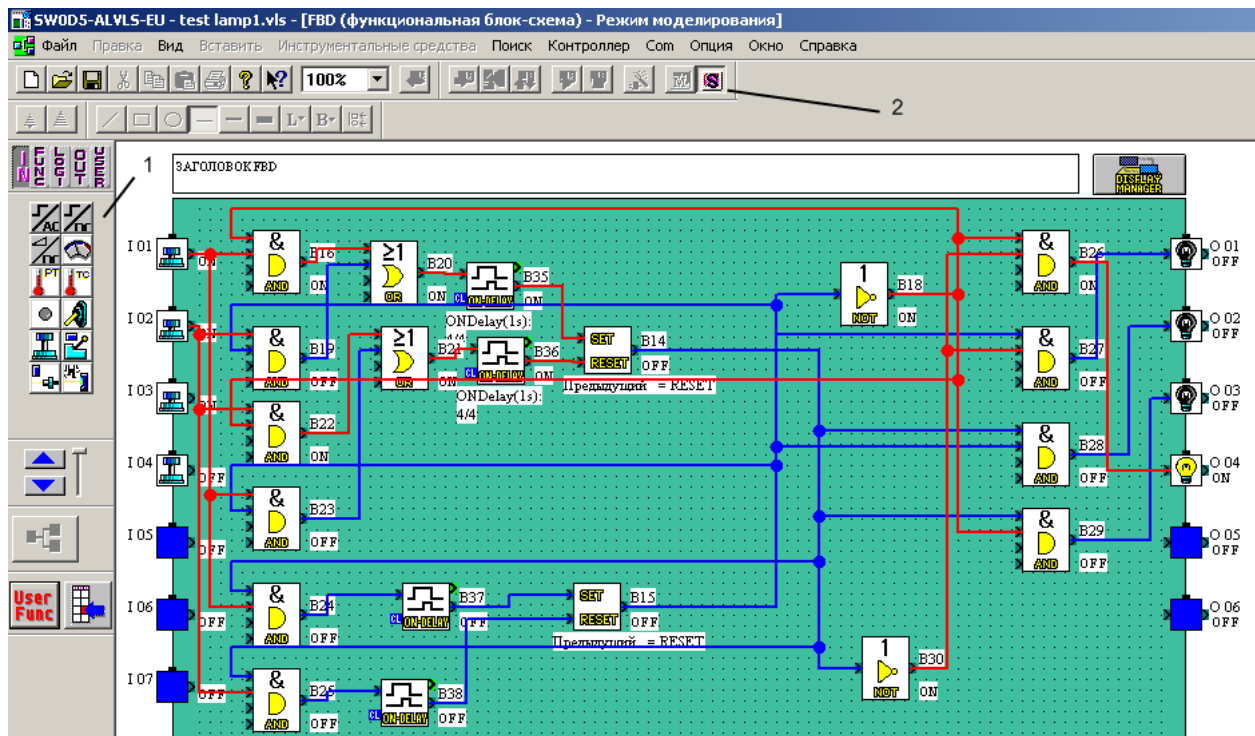


Рисунок 17.4 – Програмне середовище «ALPHA Programming»

Входи контролера знаходяться в правій частині вікна з'єднань, позначаються I01 - I08 і представлені у вигляді квадратів, в які необхідно вставити блок з бібліотеки «IN», що буде формувати необхідний вхідний сигнал. Аналогічно необхідно налаштувати виходи контролера, які розміщені – справа і позначаються O01 – O06.

Налаштування входів і створення програми в «ALPHA Programming» виконується шляхом перетягування мишкою значків блоків з бібліотеки на поле для набору програми і з'єднання цих блоків між собою. Прокладання лінії з'єднання виконується автоматично після послідовного натискання на вході і виході елементів, які необхідно з'єднати. Відображення з'єднань на екрані спрощує процес програмування як для початківців так і для досвідчених користувачів.

Після з'єднання всіх блоків між собою, підключення їх до входів і виходів необхідно виконати моделювання роботи схеми натиснувши кнопку «запуск моделювання» (на рис. 17.4 позначено 2). Під час моделювання можна змінювати вхідні сигнали за допомогою вхідних блоків і спостерігати зміну сигналів на виході за допомогою ламп на виходах.

## 17.2. Програма роботи

Для завдання, номер якого співпадає з номером бригади:

1. Розбити завдання на прості задачі і виконати логічний синтез для кожної з них.
2. Відповідно до синтезованих виразів скласти програму на мові FBD.
3. Набрати програму в середовищі «ALPHA Programming».
4. Налаштувати та протестувати програму в режимі емуляції.

**Задача № 1.** Виконати синтез схеми, що має три вхідних сигналів  $a_1 \dots a_3$  і чотири вихідних  $b_1 \dots b_4$ . Якщо три вхідні сигнали  $a_1 \dots a_3$  дорівнюють одиниці і спочатку надходить  $a_1$ , потім  $a_2$  і останнім  $a_3$ , то на виході тричі повторюється послідовність вмикання  $b_1 \rightarrow b_2 \rightarrow b_3$  з тривалістю ввімкнутого/вимкнутого стану  $2c$ , на подальші надходження сигналів  $a_1 \dots a_3$  система не реагує. Якщо послідовність надходження вхідних сигналів  $a_1 \dots a_3$  неправильна, то вихідний сигнал  $b_4$  мигає з періодом  $2c$ .

**Задача №2.** Виконати синтез схеми, що має три вхідних сигналів  $a_1 \dots a_3$  і п'ять вихідних  $b_1 \dots b_5$ . Якщо три вхідні сигнали  $a_1 \dots a_3$  дорівнюють одиниці, причому першим надходить  $a_1$ , то на 10 с запускається послідовність появи вихідних сигналів  $b_1, b_4 \rightarrow b_2, b_3$  з тривалістю ввімкнутого/вимкнутого стану 2 с, на подальші надходження сигналів  $a_1 \dots a_3$  система не реагує протягом 20 с. Якщо послідовність надходження вхідних сигналів  $a_1 \dots a_3$  неправильна, то з'являється вихідний сигнал  $b_5$ .

**Задача №3.** Виконати синтез схеми, що має три вхідних сигналів  $a_1 \dots a_3$  і чотири вихідних  $b_1 \dots b_4$ . Сигнали  $a_1$  і  $a_2$  представляють дворозрядне двійкове число. Якщо число збільшується, то запускається послідовність появи вихідних сигналів  $b_1 \rightarrow b_2 \rightarrow b_3$  з тривалістю ввімкнутого/вимкнутого стану 1 с. Якщо число зменшується, то сигнал  $b_4 = 1$ .

**Задача №4.** Виконати синтез схеми, що має чотири вхідних сигналів  $a_1 \dots a_4$  і чотири вихідних  $b_1 \dots b_4$ . Система відпрацьовує наступну послідовність із затримкою часу 3 с вхідних і вихідних сигналів:

$a_1 a_2 a_3 a_4$ : 0000  $\rightarrow$  1000  $\rightarrow$  0000  $\rightarrow$  0100  $\rightarrow$  0000  $\rightarrow$  0010  $\rightarrow$  0000  $\rightarrow$  0001  $\rightarrow$  0000  
 $b_1 b_2 b_3 b_4$ : 0000  $\rightarrow$  0001  $\rightarrow$  0011  $\rightarrow$  0111  $\rightarrow$  1111  $\rightarrow$  1110  $\rightarrow$  1100  $\rightarrow$  1000  $\rightarrow$  0000

### 17.3. Зміст звіту

Звіт має містити такі матеріали:

- 1) заданні умови роботи схеми;
- 2) завдання для простих задач і логічний синтез для кожної з них;
- 3) програму на мові FBD, що реалізує всі задані функції;
- 4) висновки про відповідність роботи схеми заданим умовам і основні проблеми при налаштуванні.

### 17.4. Контрольні запитання

1. Основні складові програмованих логічних контролерів та їх призначення.
2. Визначення програмованих логічних контролерів переваги та недоліки.
3. Области застосування програмованих логічних контролерів.
4. Характерні особливості контролерів початкового рівня.
5. Основні етапи розв'язання задач автоматизації на основі ПЛК.
6. Правила складання програм на мові FBD, опис блоків, що реалізують логічні функції.
7. Опишіть функції основних дискретних функціональних блоків.

## Лабораторна робота №18

### СИНТЕЗ ТА ДОСЛІДЖЕННЯ СИСТЕМ АВТОМАТИЗАЦІЇ НА БАЗІ ЛОГІЧНИХ КОНТРОЛЕРІВ SIEMENS LOGO

Тривалість лабораторної роботи – 4 години.

Тривалість домашньої роботи – 4 години.

**Мета роботи** – вивчити будову та принцип дії програмованого контролера Siemens LOGO! 12/24RCE, принцип роботи двох контролерів разом по технології Master-Slave та практично відпрацювати правила програмування, запису та відпрацювання програм.

#### 18.1 Основні теоретичні відомості

Логічний програмований контролер призначений для реалізації логічних, часових та лічильних функцій керування виробничими механізмами та обладнанням за програмами, що записані у блоці пам'яті контролера.

Принцип дії контролера полягає у тому, що його процесор зчитує та виконує команди, які записано у запам'ятовуючому блоці, згідно з якими він приймає та опрацьовує сигнали, що надходять на входи, та забезпечує спрацьовування виходів.

Програмування контролера полягає у перетворенні алгоритму роботи схеми автоматичного керування у послідовність команд, що записуються у контролері. Система команд з їх умовними позначеннями, а також правила використання команд складають мову програмування контролера.

В ПЛК SIEMENS LOGO! використовуються наступні позначення змінних: великою літерою «I» позначаються вхідні сигнали (I1 – I8), великою літерою «Q» позначаються вихідні сигнали (Q1 – Q4), всі інші елементи програми позначаються «BO» – це тригери, лічильники, логічні елементи, таймери та інші елементи. Даний контролер програмується за допомогою

програмного забезпечення LOGO!Soft Comfort на мовах програмування FBD (Function Block Diagram) або LD (Ladder Diagram).

Якщо система автоматизації керується не одним контролером, при цьому необхідно керувати всіма контролерами одночасно та мати від них зворотні зв'язки, то для вирішення даної проблеми існує технологія **Master-Slave**. В якості «**Master**» може виступати як сам контролер, так і сервер.

В даній лабораторній роботі використовується два контролери Siemens LOGO які будуть працювати разом по технології **Master-Slave**. **Master-Slave** – це модель асиметричної взаємодії чи комунікації, де один прилад чи процес контролює один чи більше інших пристроїв чи процесів і служить їх комунікаційним концентратором. У деяких системах "Master" обирається з групи придатних пристроїв чи процесів, а інші пристрої чи процеси з цієї групи є "Slaves". Зв'язок для даної технології реалізується через інтерфейс RJ45. Тобто в кожного контролера є своя IP адреса, яка необхідна для розпізнання пристрою з яким необхідно працювати. IP **Master 192.168.000.003**, IP **Slave 192.168.000.004**. Також є можливість поміняти IP адресу контролера, нижче буде описаний спосіб як це зробити. Дані контролери уже налаштовані на режим роботи **Master-Slave**. Докладніше про дане налаштування описано нижче.

Для початку внесення програми в ПЛК потрібно увімкнути автомат QF1, після того, як контролери увімкнутися необхідно підключитися до контролера через інтерфейс з'єднання RJ45. Відкрити програму LOGO!Soft Comfort та вибрати готову програму, так як показано на рис. 18.1

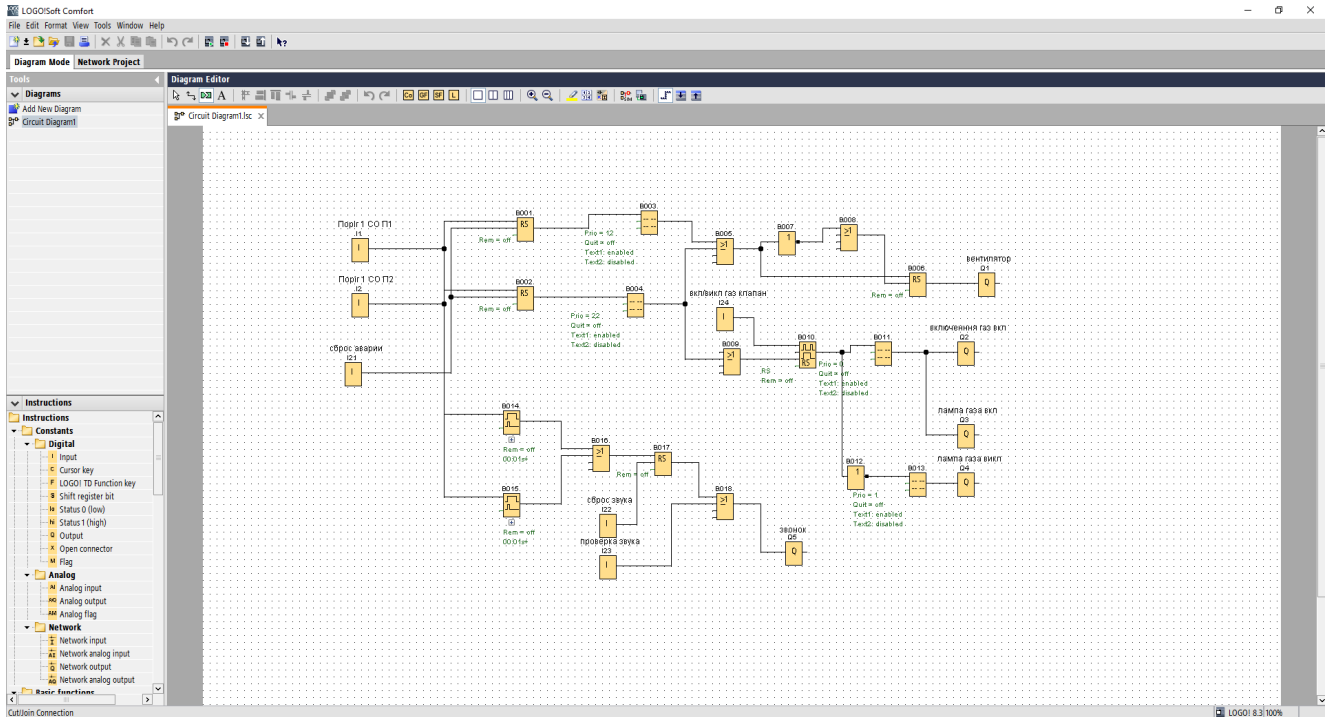


Рисунок 18.1 – Готова програма в LOGO!Soft Comfort

Після відкриття програми необхідно завантажити її до контролера. Це можна реалізувати декількома способами. Один із них це вибрати на панелі інструментів меню **Tools**, потім вибрати **Transfer** та вибрати **PC->LOGO!**, або застосувати комбінацію клавіш **Ctrl+D**. Також можна вибрати на панелі інструментів кнопку загрузки програми в контролер, рис. 18.3.

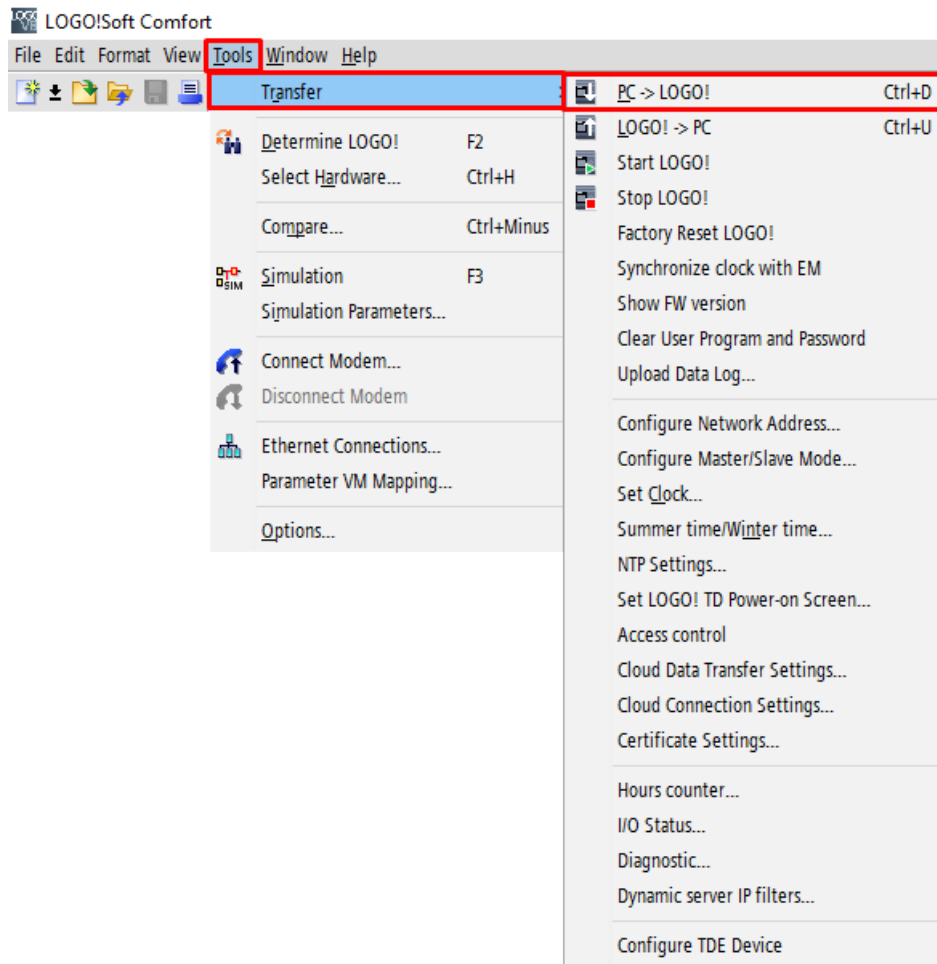


Рисунок 18.2 – Завантаження програми до контролера

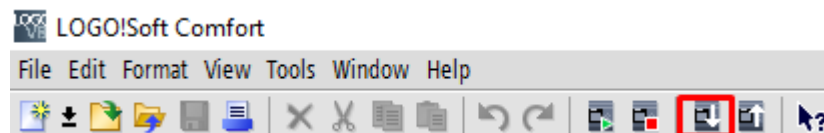


Рисунок 18.3 – Завантаження програми до контролера через панель інструментів

Потім з'являється наступне вікно, рис. 18.4, в якому необхідно вибрати інтерфейс, по якому буде відбуватися завантаження програми в контролер. На рис. 18.4 показано який повинен бути вибраний інтерфейс зв'язку. В полі **Target IP address** необхідно вказати IP адресу контролера, на який буде завантажуватися програма, рис. 18.4.

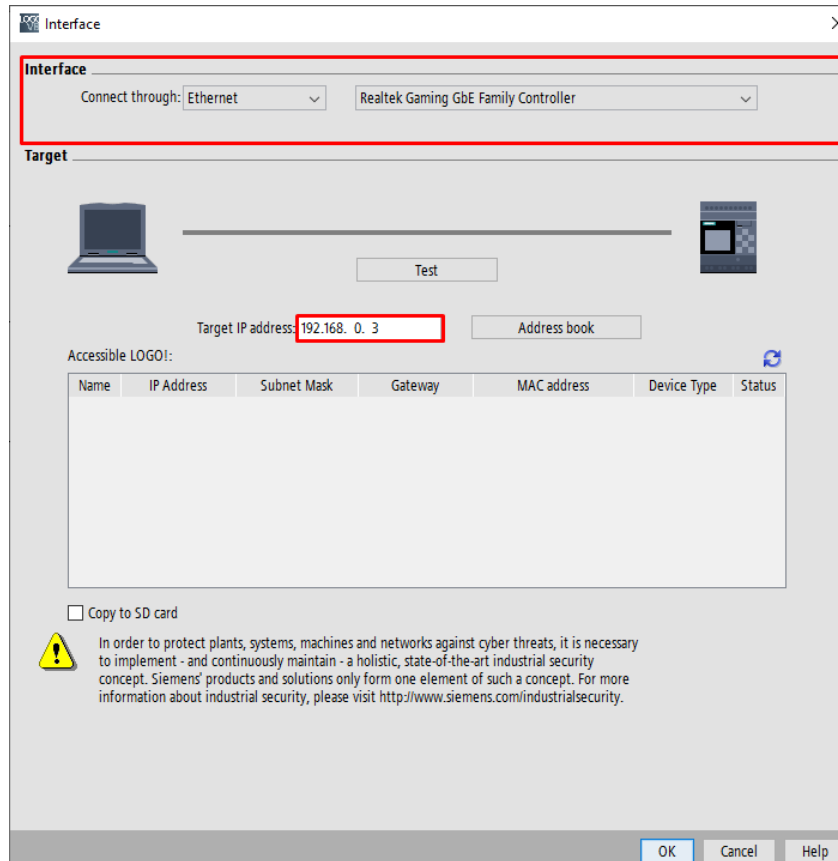


Рисунок 18.4 – Налаштування завантаження програми в контролер

Після налаштування необхідно натиснути кнопку **ОК** і почнеться завантаження програми в контролер, якщо програма в контролері виконувалась, то з'явиться вікно, яке запропонує призупинити виконання даної програми, необхідно натиснути **ОК**. Потім відбувається завантаження програми в контролер. Після завантаження програми в контролер з'являється вікно, яке пропонує запусити програму, і необхідно вибрати чи одразу її запусити чи запусити пізніше на самому контролері.



*Рисунок 18.5 – Меню контролера при зупиненій програмі*

На рис. 18.5 зображено меню контролера при зупиненій програмі. Пункт **Start** відповідає за запуск програми, тобто при натисненні на даний пункт програма, яка знаходиться в контролері, починає працювати. Наступний пункт меню це – **Program**. В даному пункті знаходиться налаштування програми. На рис. 18.6 показано параметри, які можна змінювати в цьому пункті. В даний контролер можна записати програму, яка буде складатися з не більше ніж з 400 блоків.



*Рисунок 18.6 – Підпункт меню **Program***

В підпункті меню **Program** можна змінювати програму або створити її, налаштувати програму, поставити пароль на програму, тобто захистити її від редагування або від зчитування. Даний контролер підтримує режим вивантаження програми на пристрій збереження (flash-card) або на комп'ютер за допомогою відповідного програмного забезпечення. Можна давати ім'я програми або його змінювати. Можна очистити програму або перевірити кількість пам'яті, яку займає програма.

В підпункті меню **Setup** знаходяться налаштування самого контролера. В даному пункті можна налаштовувати початковий екран, затримку, таймери годинник, екран програми, змінювати мову меню та інше. На рис. 18.7 зображено пункт меню **Setup**.



Рисунок 18.7 – Підпункт меню **Setup**

В пункті меню **Network** знаходиться налаштування контролера для налаштування його роботи в мережі. Також в даному пункті відбувається налаштування роботи технології **Master-Slave**, рис. 18.8.



Рисунок 18.8 – Підпункт меню **Network**

Налаштування режиму **Master-Slave** відбувається наступним чином: спочатку необхідно налаштувати IP-адрес на обох контролерах. Потім необхідно для одного контролера вибрати режим **Master**, для іншого контролера – режим **Slave** та вказати IP-адрес **Master** контролера, рис. 18.9.



Рисунок 18.9 – Налаштування режиму **Master-Slave**

В меню пункту **Diagnostics** можна дізнатися інформацію до якого програмного забезпечення підключено контролер, перевірити наявність додаткових модулів та перевірити налаштування, рис. 18.10.



Рисунок 18.10 – Підпункт меню **Diagnostics**

В меню пункту **Card** можна зберегти програму з контролера на flash-card або навпаки з flash-card завантажити програму на контролер, також є можливість зробити копію проекту, рис. 18.11.



Рисунок 18.11 – Підпункт меню **Card**

## 18.2 Програма роботи

За варіантом, наведеним у таблиці 18.1, виконати наступні завдання:

1. Виконати логічний синтез схеми керування, та отримати логічні вирази, що описують схему (**виконується вдома під час СРС**).
2. За отриманими алгебричними виразами у пункті 1 скласти, програму в програмному забезпечення для логічного контролера на мові *FBD*.
3. Виконати адресацію вхідних, вихідних і проміжних змінних.
4. Записати програму до контролера.
5. Перевірити відпрацювання заданих умов роботи (**викладач підтверджує своїм підписом у протоколі працездатність програми**).
6. Оформити звіт по лабораторній роботі та зробити висновки.

Таблиця 18.1 – Варіанти завдань

Бригада	Номер завдання
1	1, індивідуальне
2	2, індивідуальне
3	3, індивідуальне
4	4, індивідуальне
5	5, індивідуальне
6	6, індивідуальне

Номер завдання відповідає номеру бригади. **Після виконання завдання необхідно отримати у викладача індивідуальне завдання.**

### Завдання №1

При натисненні кнопки SB1 на семисегментному індикаторі виводяться числа від 0 до 9 по черзі через кожні 1.5 с. При натисненні кнопки SB2 індикація скидається і на семисегментнику світиться цифра 0. Якщо знову натиснути SB1, то схема повторює роботу.

### Завдання №2

Схема працює у двох режимах, які перемикаються за допомогою SA1, у першому режимі при натисненні кнопки SB1 через кожну секунду на семисегментному індикаторі по черзі з'являються парні числа, якщо натиснути

SB2 то схема скидається і повертається у вихідне положення, а у другому – по черзі парні числа із затримкою в 1 секунду при натисненні SB4, при натисненні SB5 схема скидається і повертається у вихідне положення.

### **Завдання №3**

Три мотори M1, M2, M3 вмикаються та вимикаються трьома вимикачами SB1, SB2, SB3. мотор M1 повинен працювати тільки у тому разі, коли замкнутий будь-який один вимикач, M2 – коли замкнуті перший та третій або перший та другий вимикачі, M3 – якщо замкнуті усі три вимикачі.

### **Завдання №4**

Керування трьома двигунами M6, M7, M8 здійснюється за допомогою кнопок «Пуск» і «Стоп». При натисненні кнопки «Пуск» вмикається на 5 с. M1, M3, M5, потім із затримкою часу 2 с вмикається двигун M6, потім через 4 с вмикається двигун M7 та через 8 с – двигун M8. При натисненні кнопки «Стоп» знову вмикається M2 та M4 на 5 с., потім двигун M6 вимикається без затримки часу, потім через 3 с вимикається двигун M7 та через 4 с – двигун M8.

### **Завдання №5**

Схема працює у двох режимах, які перемикаються за допомогою SA2. У першому режимі при спрацюванні кінцевого вимикача на семисегментному індикаторі з'являється відповідний номер кінцевого вимикача і зберігається поки не спрацює інший, при натисненні кнопки SB1 схема скидається і на семисегментному індикаторі з'являється нуль. У другому режимі комбінація працюючих SQ1-SQ3 відповідає двійковому коду і відповідну цифру відображає на семисегментному індикаторі.

### **Завдання №6**

При натисненні SB1 вмикається M1 і починає працювати впродовж 6 секунд. Якщо по закінченню 6 секунд перемикач SA1 знаходився в положенні 1 то вмикається M2, а M1 вимикається. M2 працює до тих пір, поки не спрацює

SQ1. Після спрацювання SQ1, M2 вимикається і вмикається M3. При спрацюванні SQ2, M3 зупиняється і чекатиме натиснення SB2. Після натиснення SB2, M3 знову працює і при спрацюванні SQ3 зупиняється. Якщо по закінченню 6 секунд перемикач SA1 знаходився в положенні 2, то вмикається M4, а M1 вимикається. M4 працює до тих пір, поки не спрацює SQ4. Після спрацювання SQ4, M4 вимикається і вмикається M5. При спрацюванні SQ5, M5 зупиняється і чекатиме натиснення SB3. Після натиснення SB3, M5 знову працює і, при спрацюванні SQ6, зупиняється. При натисненні SB4 мотори працюють поки не спрацює найближчий кінцевий вимикач.

В усіх завданнях на екрані контролера повинно відображатися відповідне повідомлення про стан роботи завдання.

### **18.3. Опис лабораторної установки**

Функціональну схему лабораторного стенду представлено на рис. 18.12. Даний лабораторний стенд має відповідати наступним вимогам: простота та зручність експлуатації; наочність монтажу; відсутність потреби в додатковому обладнанні. Програма до контролера складається в програмуному забезпеченні LOGO!Soft Comfort на мові програмування FBD або LD.

В залежності від завдання, за допомогою кнопок SB1 – SB6, двопозиційних кнопок SA1 – SA2, кінцевих вимикачів SQ1 – SQ6 подаються вхідні сигнали на ПЛК.

Після подачі сигналів від ПЛК можна бачити роботу схеми, як на стенді (засвічування сигнальних ламп, спрацьовування магнітних контакторів, робота мікродвигуни, робота АД, засвідчування сегментів на семисегментному індикаторі) так і на дисплеї програмованого логічного контролера, якщо це зробити в програмі.

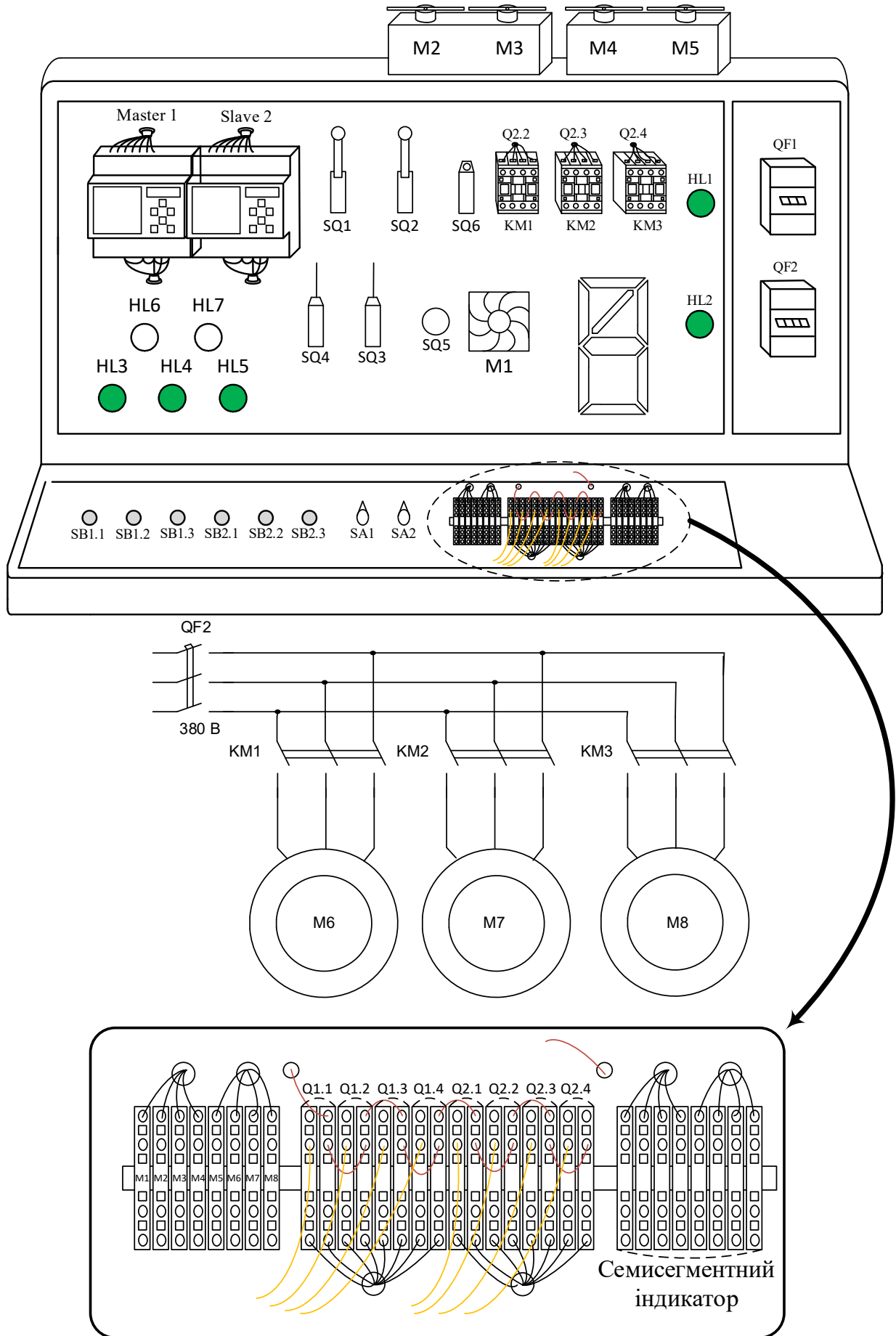


Рисунок 18.12 – Функціональна схема лабораторного стенду

На панелі стенду представлено: програмовані логічні контролери «Siemens LOGO», сигнальні лампи на 24 В (HL3–HL7) та на 220 В (HL1, HL2), мікродвигуни М1 – М5, асинхронні двигуни М6 – М8, магнітні контактори КМ1–КМ3, автоматичні вимикачі QF1, QF2, кнопки без фіксації SB1-SB6, двопозиційних кнопок SA1 – SA2, кінцевих вимикачів SQ1 – SQ6, та семисегментний індикатор.

Використаний у лабораторному стенді програмований логічний контролер «Siemens LOGO» зображений на рисунку 18.13.

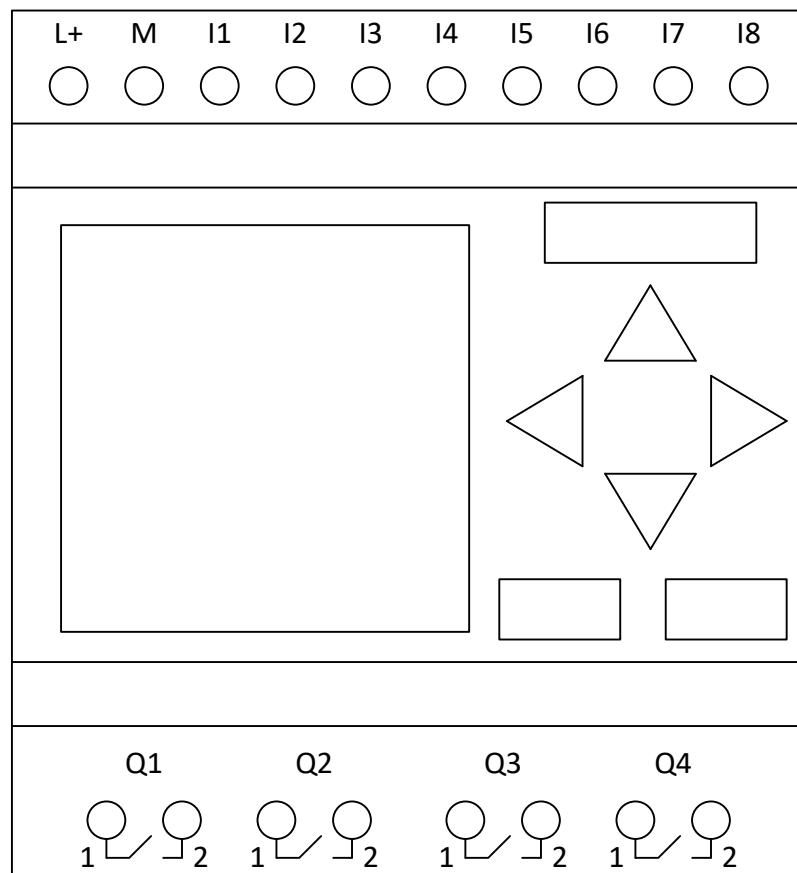


Рисунок 18.13 – Програмований логічний контролер «Siemens LOGO»

На клеми «L+» «M» подається живлення ПЛК, на входи I1-I8 подаються вхідні сигнали (їх формують кнопки, перемикачі, датчики). До виходів контролера підключаються сигнальні лампи, магнітні контактори, семисегментний індикатор, мікродвигуни.

Схема підключень стенда зображена на рисунку 18.15, електрична принципова схема зображена на рисунку 18.16. Через автоматичний вимикач QF1 подається живлення на мікродвигуни М1-М5, ПЛК(Master, Slave), магнітні

контактори (KM1-KM3), кнопки (SB1-SB6, SA1, SA2), семисегментний індикатор, кінцеві вимикачі (SQ1-SQ6). QF2 подає живлення на асинхронні двигуни через магнітні контактори.

Наявність живлення після увімкнення QF1 або QF2 сигналізує відповідна лампа HL1 або HL2.

При вмиканні тумблерів автомата QF1 спрацьовує світлова сигналізація (HL1). В подальшому напруга протікає на два блоки живлення, один блок живлення видає DC 24 В, а інший DC 12 В. 12 В постійної напруги потрібна для керування семисегментним індикатором. 24 В постійної напруги необхідно для всього іншого.

При вмиканні тумблерів трифазного вимикача QF2 спрацьовує світлова сигналізація (HL2). В подальшому напруга протікає до магнітних контакторів і, коли вони спрацьовують, то напруга далі передається до асинхронних двигунів.

Всі кінцеві вимикачі, окрім SQ5, мають по два типи контактів NC та NO. Кінцевий вимикач SQ5 працює на ефекті Холла. Зона спрацьовування датчика 0-10 мм. На задній торцевій частині датчика встановлено червоний світлодіод індикації роботи. Датчик являється NPN типу та є уніполярним. Коли датчик Холла спрацьовує, то на його виході з'являється високий рівень сигналу.

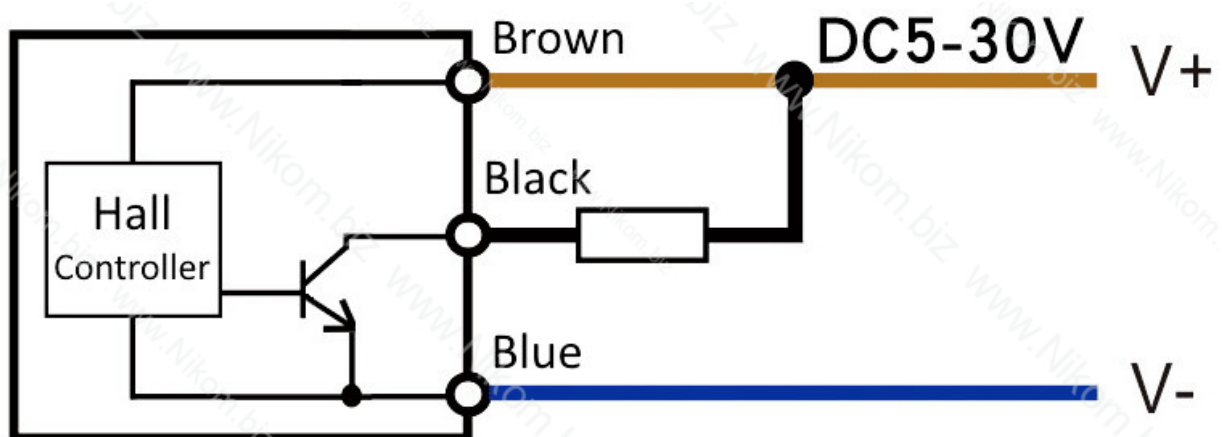


Рисунок 18.14 – Схема роботи датчика Холла

Кінцеві вимикачі SQ1, SQ2, SQ6 мають нормально відкриті контакти. Кінцеві вимикачі SQ3, SQ4 мають нормально закриті контакти. SQ5 має нормально відкритий контакт, але на контролер приходять інверсний сигнал.

Таблиця 18.2 – Підключення вхідних сигналів до ПЛК (Master)

Входи контролера	Вхідні сигнали
I1	SB1.1
I2	SB1.2
I3	SB1.3
I4	SA1.1
I5	SA1.2
I6	SQ1
I7	SQ3
I8	SQ5

Таблиця 18.3 – Підключення вхідних сигналів до ПЛК (Slave)

Входи контролера	Вхідні сигнали
I1	SB2.1
I2	SB2.2
I3	SB2.3
I4	SA2.1
I5	SA2.2
I6	SQ2
I7	SQ4
I8	SQ6

Вихідні сигнали можна підключити до будь-якого виходу контролера. Потрібно пам'ятати, що контролер має релейні виходи, тобто до одного контакту необхідно подати живлення, а до іншого підключається пристрій. На рис. 18.17 показана схема підключення до входів та виходів контролера.

Схема підключень лабораторного обладнання

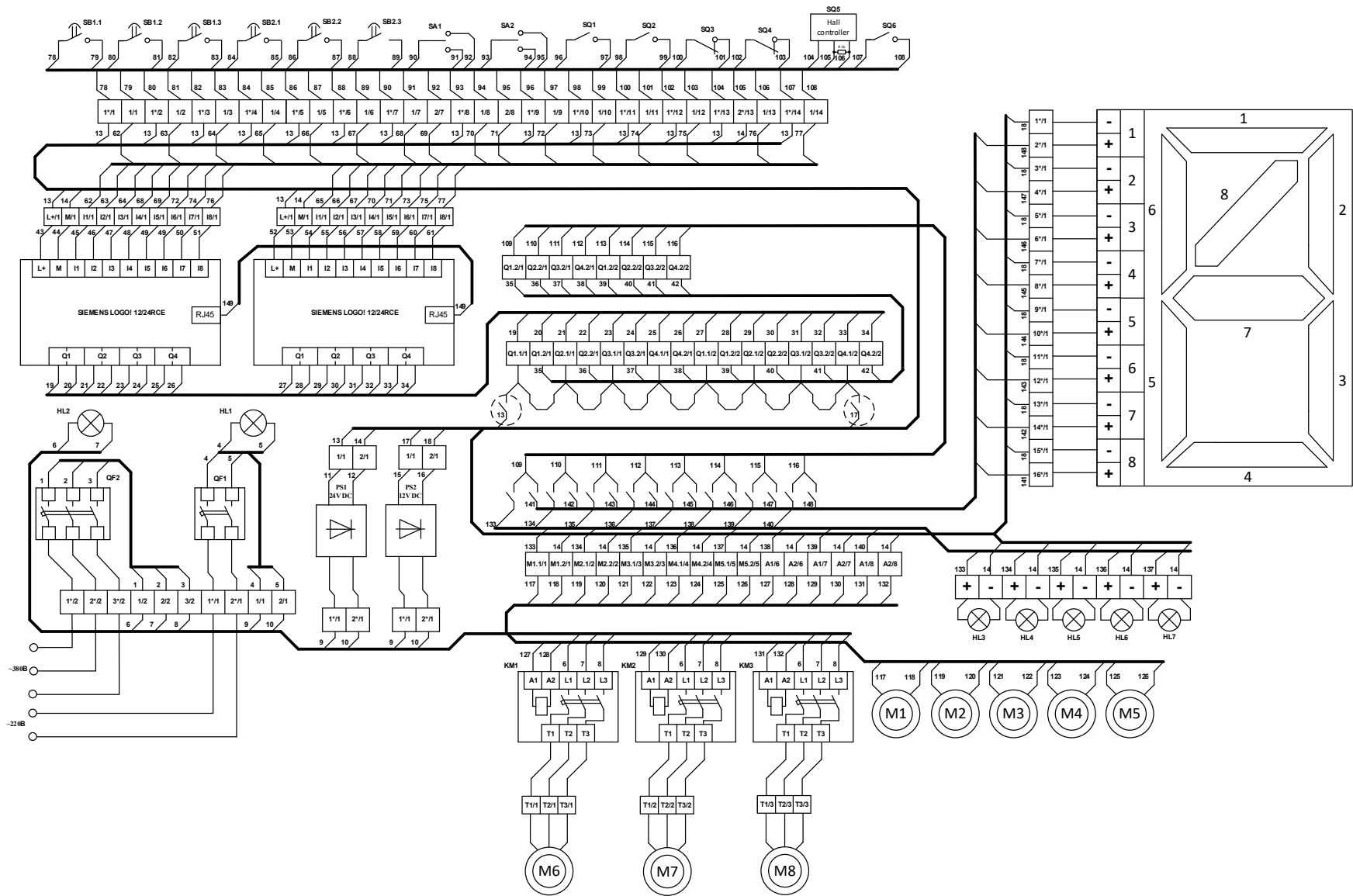


Рисунок 18.15 – Схема підключень лабораторного стенду

### Схема електрична принципова

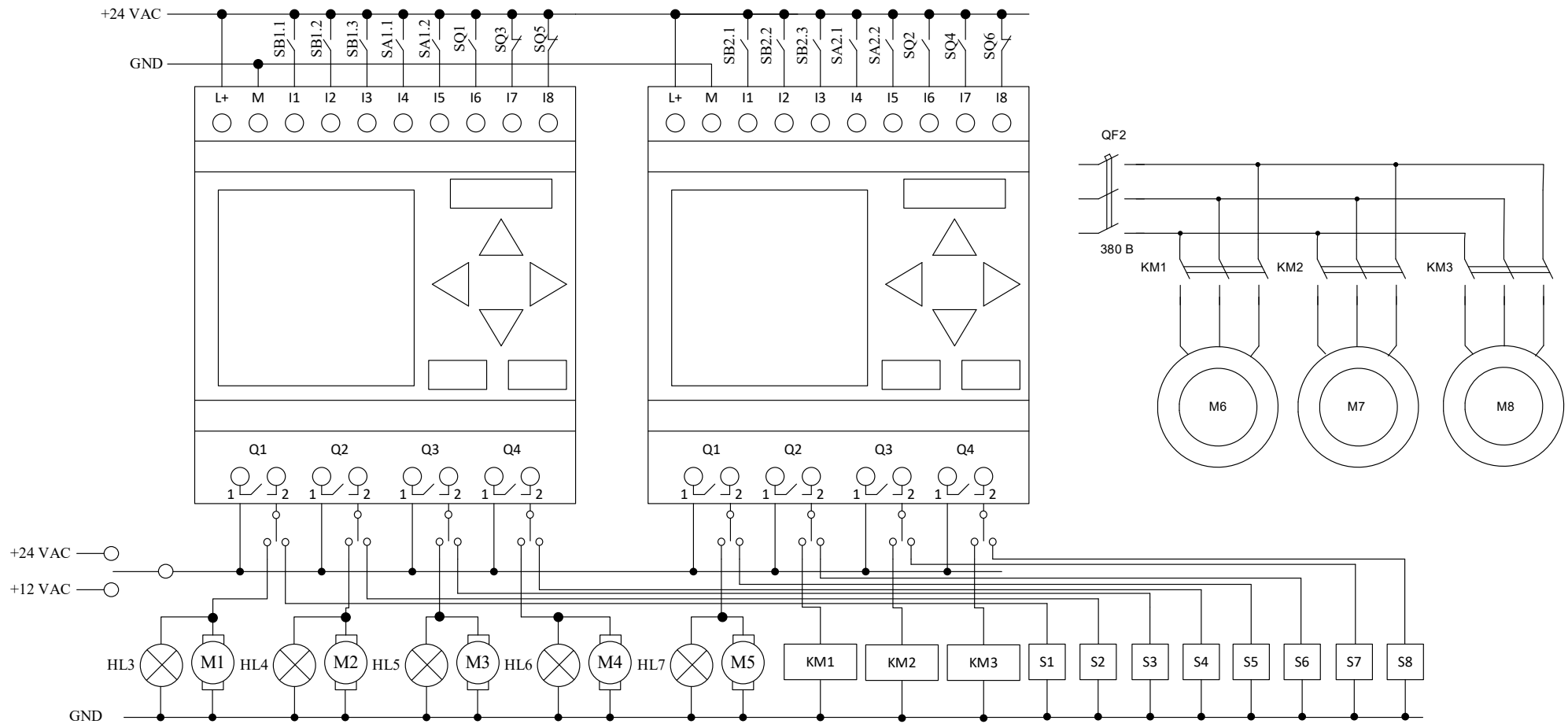


Рисунок 18.16 – Електрична принципова схема лабораторногo станду

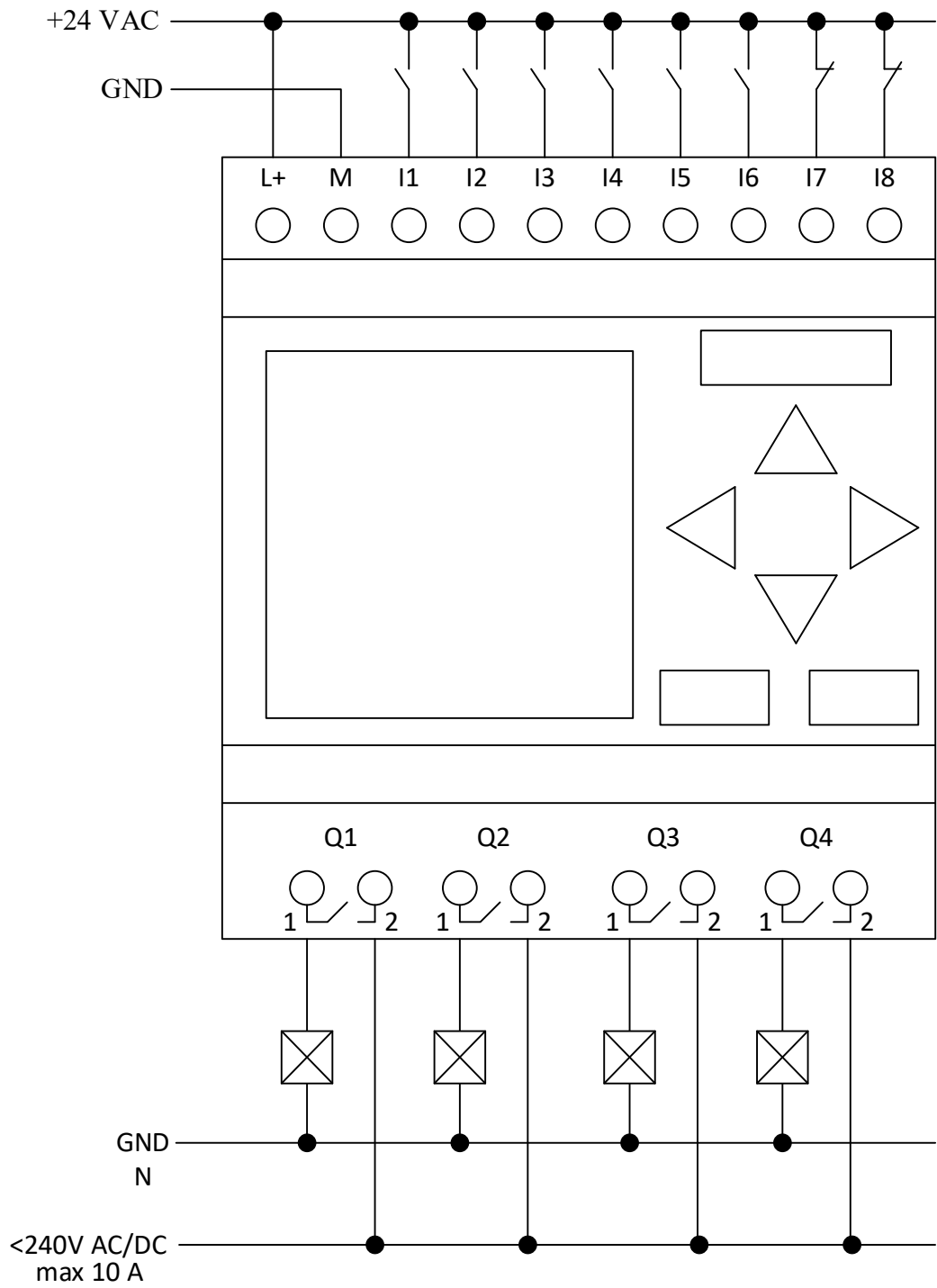


Рисунок 18.17 – Схема підключення контролера

## 18.4. Методичні вказівки до виконання роботи

Для встановлення програмного забезпечення необхідно спочатку встановити файл під назвою Step1\_LOGOV7\_Setup. Після встановлення необхідно запустити дану програму один раз та вимкнути її. Потім необхідно встановити Step2\_LOGOV8.3\_Upgrade. Після встановлення можна запускати та створювати програму для ПЛК.

Ім'я	Дата змінення	Тип	Розмір
Step1_LOGOV7_Setup	07.10.2011 18:12	Застосунок	60 637 КБ
Step2_LOGOV8.3_Upgrade	21.09.2020 6:50	Застосунок	482 781 КБ

Рисунок 18.18 – Файли для встановлення LOGO!Soft Comfort

Розглянемо приклад розв'язку завдання.

Трьома механізмами з двигунами M1, M2, M3 керують за допомогою кнопок “Пуск” і “Стоп”. При натисненні кнопки “Пуск” спочатку вмикається звукова попереджувальна сигналізація, а потім через час  $\Delta t$  – двигуни в послідовності M1 – M2 – M3. Після запуску останнього двигуна сигналізація вимикається. При натисненні кнопки “Стоп” двигуни вимикаються без затримки часу в зворотному порядку : M3 – M2 – M1. Зробити індикацію роботи моторів. Скласти схему керування.

Таблиця 18.4 – Сигнали програми

Вхідні сигнали	Вихідні сигнали
Кнопка «Пуск», А	Мотор 1, M1
Кнопка «Стоп», В	Мотор 2, M2
	Мотор 3, M3
	Лампа Індикації 1, HL1
	Лампа Індикації 2, HL2
	Лампа Індикації 3, HL3

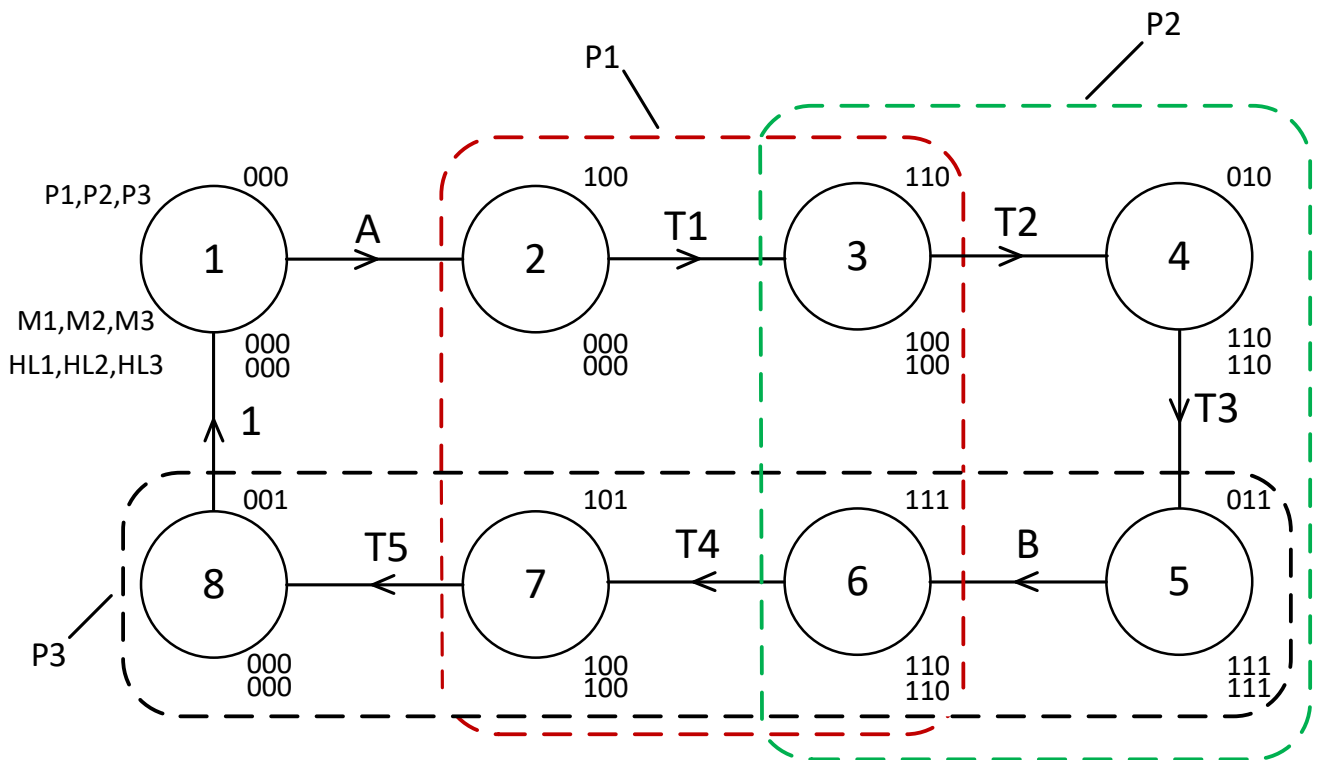


Рисунок 18.19 – Графоперехід відповідно до завдання

Відповідні позначення: Кнопка «Пуск» – А; кнопка «Стоп» – В;  $T1 = 10$  с.,  $T2 = 3$  с.,  $T3 = 3$  с.,  $T4 = 3$  с.,  $T5 = 3$  с. – таймери.

Складемо рівняння для графа переходу який зображено на рис. 18.19.

$$Sp_1 = A \cdot \bar{p}_2 \cdot \bar{p}_3 + B \cdot p_2 \cdot p_3; \quad (18.1)$$

$$Rp_1 = T2 \cdot p_2 \cdot \bar{p}_3 + T5 \cdot \bar{p}_2 \cdot p_3; \quad (18.2)$$

$$Sp_2 = T1 \cdot p_1 \cdot \bar{p}_3; \quad (18.3)$$

$$Rp_2 = T4 \cdot p_1 \cdot p_3; \quad (18.4)$$

$$Sp_3 = T3 \cdot \bar{p}_1 \cdot p_2; \quad (18.5)$$

$$Rp_3 = \bar{p}_1 \cdot \bar{p}_2; \quad (18.6)$$

$$T1 = p_1 \cdot \bar{p}_2 \cdot \bar{p}_3; \quad (18.7)$$

$$T2 = p_1 \cdot p_2 \cdot \bar{p}_3; \quad (18.8)$$

$$T3 = \bar{p}_1 \cdot p_2 \cdot \bar{p}_3; \quad (18.9)$$

$$T4 = p_1 \cdot p_2 \cdot p_3; \quad (18.10)$$

$$T5 = p_1 \cdot \bar{p}_2 \cdot p_3; \quad (18.11)$$

$$M1 = p_2 + p_1 \cdot p_3; \quad (18.12)$$

$$M2 = \overline{p_1} \cdot p_2 + p_2 \cdot p_3; \quad (18.13)$$

$$M3 = \overline{p_1} \cdot p_2 \cdot p_3; \quad (18.14)$$

$$HL1 = p_2 + \overline{p_1} \cdot p_3; \quad (18.15)$$

$$HL2 = \overline{p_1} \cdot p_2 + p_2 \cdot p_3; \quad (18.16)$$

$$HL3 = \overline{p_1} \cdot p_2 \cdot p_3. \quad (18.17)$$

Таблиця 18.5 – Адресація сигналів

Сигнал в рівнянні	Сигнал в програмі	Сигнал на стенді
A	I1	SB1.1
B	I2	SB1.2
M1	Q1	Q1.1
M2	Q2	Q1.2
M3	Q3	Q1.3
HL1	Q1	Q1.1
HL2	Q2	Q1.2
HL3	Q3	Q1.3

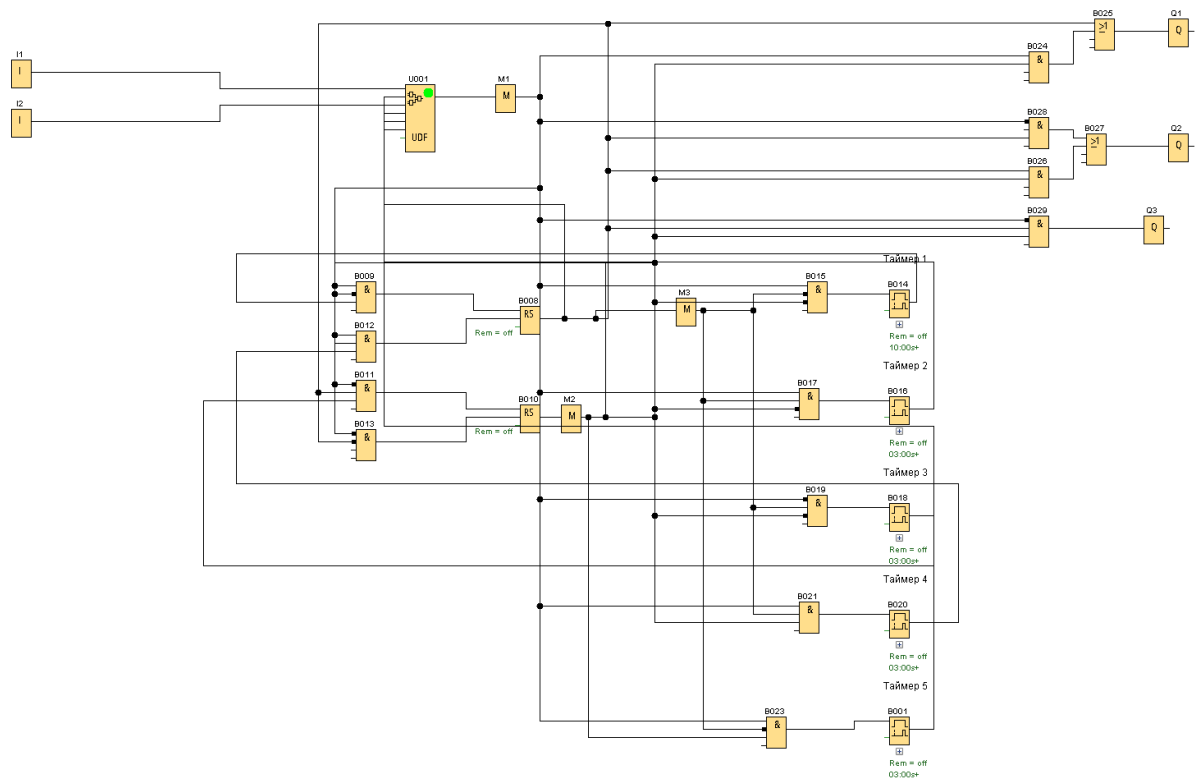


Рисунок 18.20 – Програма, яка відповідає граф переходу на мові FBD

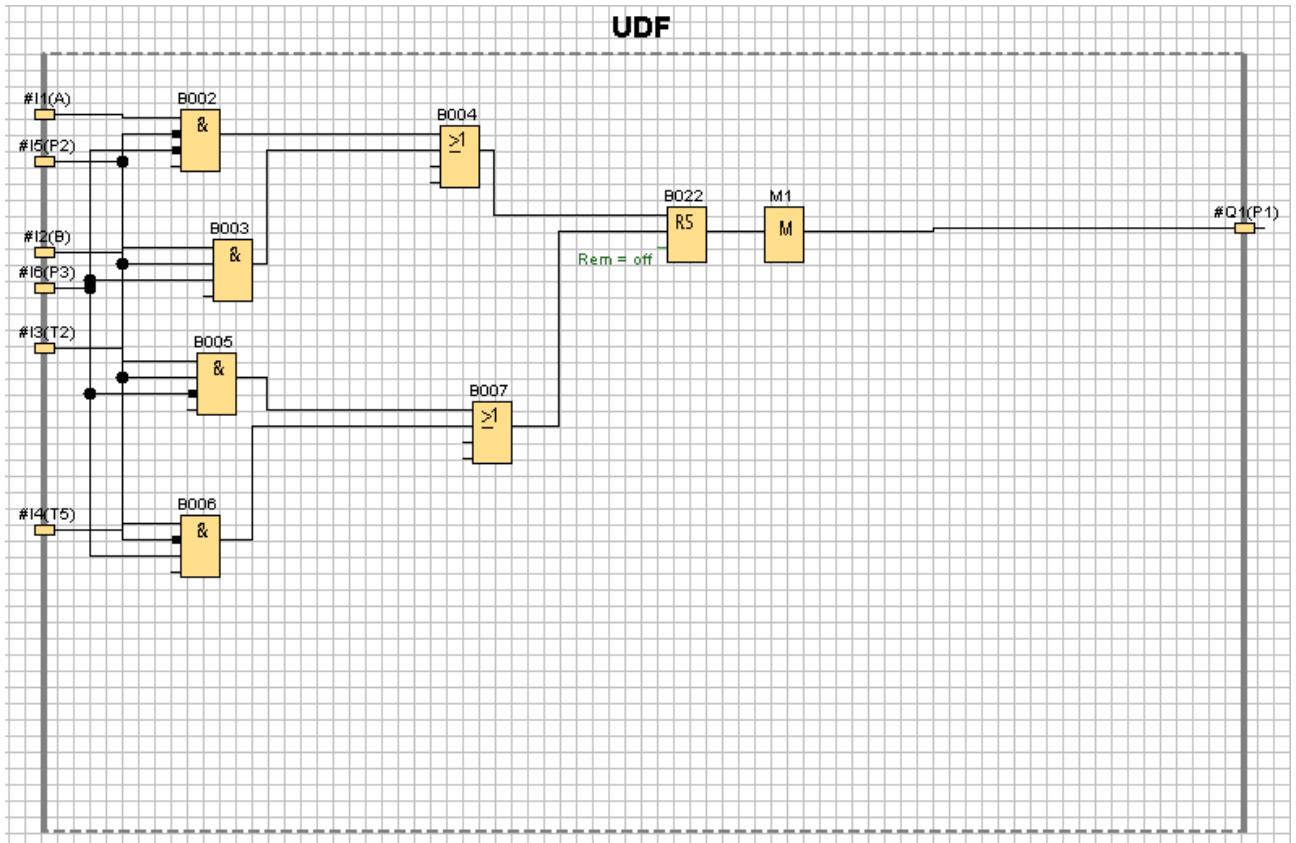


Рисунок 18.21 – Блок вмикання та скидання тригера P1

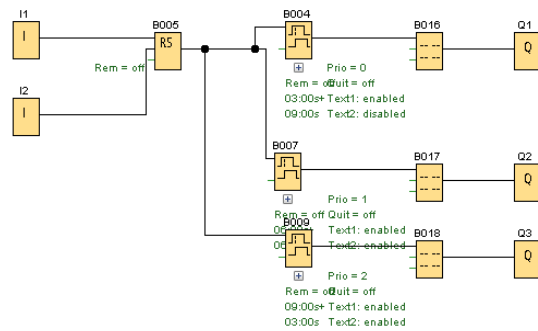


Рисунок 18.22 – Мінімізована програма на FBD

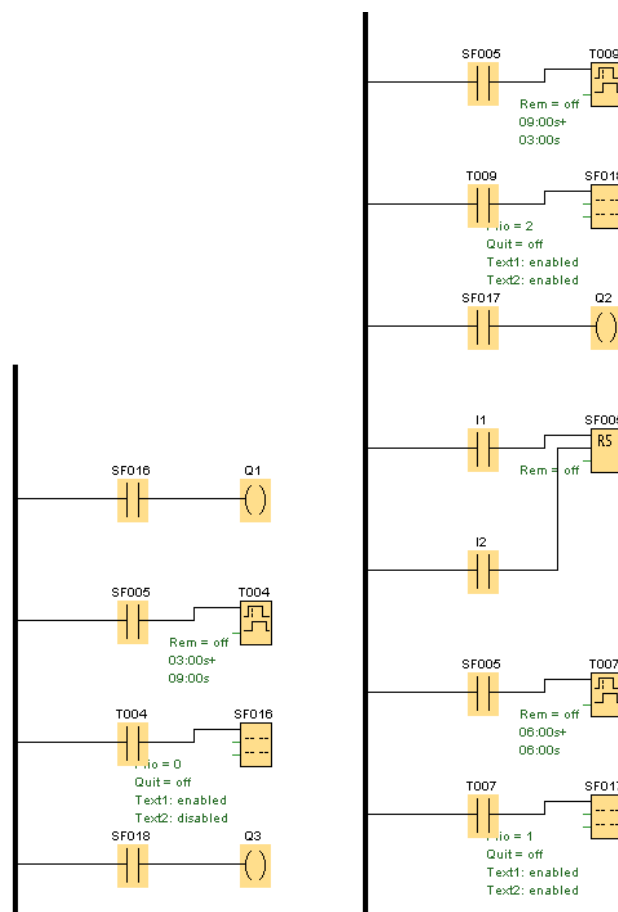


Рисунок 18.22 – Мінімізована програма на мові LD

#### 18.4. Зміст звіту

- 1) Титульний аркуш.
- 2) Мета роботи, програма роботи.
- 3) Опис та схема лабораторного стенду.
- 4) Умови роботи схем.
- 5) Логічний синтез схем керування.
- 6) Адресація входів, виходів, таймерів, комірок пам'яті.
- 7) Програма на мові FBD. Якщо в програмі присутні блоки, які зроблено користувачем, то необхідно їх окремо показати – з чого вони складаються.
- 8) Висновок про відповідність роботи заданим умовам.

### 18.5. Контрольні запитання

1. Які позначення використовуються в ПЛК Siemens LOGO!?
2. На яких мовах можливе програмування у середовищі LOGO!Soft Comfort?
3. Пояснити основні принципи технології Master-Slave.
4. Який принцип роботи датчика Холла?
5. Яка напруга живлення контролерів і який пристрій живить їх?

## **Заходи безпеки під час виконання лабораторних робіт в лабораторії «Автоматизації технологічних процесів, установок і комплексів»**

Перед початком лабораторних робіт проводиться інструктаж з правил безпеки згідно з інструкцією з охорони праці, затвердженою розпорядженням декана ФЕА. Після вивчення і перевірки знань цієї інструкції прізвище кожного студента заноситься до контрольного листа, де він ставить свій особистий підпис.

Під час виконання лабораторної роботи студентам забороняється:

- заходити за огороження робочих місць, торкатися неізольованих струмопровідних частин обладнання, а також конструкцій обладнання, які обертаються;
- збирати схеми або робити перемикання в них, усувати недоліки без відключення установки;
- підніматися на фундаменти машин;
- стояти поряд з незахищеними кожухами, муфтами, дисками та частинами машин, що обертаються;
- підключати та знімати з'єднувальні провідники під напругою;
- розбирати схеми не викручуючи клеми, виривати з'єднувальні провідники із клем;
- приєднувати провідники до приладів, машин та апаратів без напаяних наконечників;
- робити переключення на головних (розподільних) щитах;
- вмикати вимикачі для включення напруги до перевірки викладачем правильності з'єднань у схемі;
- виконувати лабораторну роботу одному.

Студенти зобов'язані:

- перед включенням схеми перевірити, в якому положенні знаходиться решта членів бригади (чи торкається хто-небудь до

струмопровідних частин або до частин машин, які можуть обертатися) та попередити голосом: «Обережно, вмикаю!»;

– не залишати лабораторну установку без нагляду, постійно спостерігати за її справною роботою.

Після закінчення роботи необхідно відключити лабораторну установку, розібрати схему, навести порядок на робочому місці, повідомити викладача про всі неполадки (якщо вони є), виявлені під час роботи.

## Список рекомендованої літератури

1. Ковальчук О.В. Логічний синтез дискретних схем автоматики: навчальний посібник – К.: НТУУ «КПІ», 2008. – 168 с. ISBN 978-966-622-294-0.
2. І.М. Бондаренко, О.В. Бородін, В.П. Карнаушенко Сучасна компонентна база електронних систем: навч. посібник для студентів ЗВО. / І.М. Бондаренко, О.В. Бородін, В.П. Карнаушенко. – Харків: ХНУРЕ, 2020. – 268 с. (доступ за посиланням [https://openarchive.nure.ua/bitstream/document/14062/3/SKB\\_2020.pdf](https://openarchive.nure.ua/bitstream/document/14062/3/SKB_2020.pdf)).
3. Rabbie, Max (2018). Programmable Logic Controllers: hardware and programming. ISBN: 9781631269325 (доступ за посиланням <https://cutt.ly/IChInHV>).
4. Petruzella, Frank D. (2017). Programmable logic controllers. ISBN 978-0-07-337384-3. (доступ за посиланням <https://cutt.ly/1ChIL4e>).
5. Hanssen, Dag Hakon (2015). Programmable Logic Controllers: A Practical Approach to IEC 61131-3 using CODESYS. ISBN 978-1-118-94924-5 (доступ за посиланням <https://cutt.ly/ZChUZus>).
6. Ковальчук, О. В., Бур'ян, С. О. (2010). Застосування різних методів в синтезу для складних програм для логічних програмованих контролерів. Інформаційний збірник Промелектро. «Промислова електроніка та електротехніка». (4). 51–53 (доступ за посиланням <https://ela.kpi.ua/bitstream/123456789/38235/1/09.pdf>).
7. Бур'ян С.О. Логічний синтез дискретних систем автоматичного керування при використанні програмованих реле низького рівня / С.О. Бур'ян, М.В. Печеник, Г.Ю. Землянухіна, І.С. Єпіфанцев // Збірник наукових праць національного університету кораблебудування імені адмірала Макарова. – 2021 - №1 (484). – С. 54-60 (доступ з посиланням [https://doi.org/10.15589/znp2021.1\(484\).7](https://doi.org/10.15589/znp2021.1(484).7)).

**ДОДАТОК А**

Міністерство освіти і науки України  
Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»

Лабораторна робота №10

з дисципліни: «Системи автоматизації-2»

на тему:

**«РІШЕННЯ ЛОГІЧНИХ ЗАДАЧ АВТОМАТИЧНОГО КЕРУВАННЯ НА  
БАЗІ ПРОГРАМОВАНОГО КОНТРОЛЕРА LOVATO LRD20RA024»**

Бригада №1

Виконали:

студенти групи ЕП-32

Шевченко Т.Г.

Франко І.Я.

Перевірила:

ас. Землянухіна Г.Ю.