

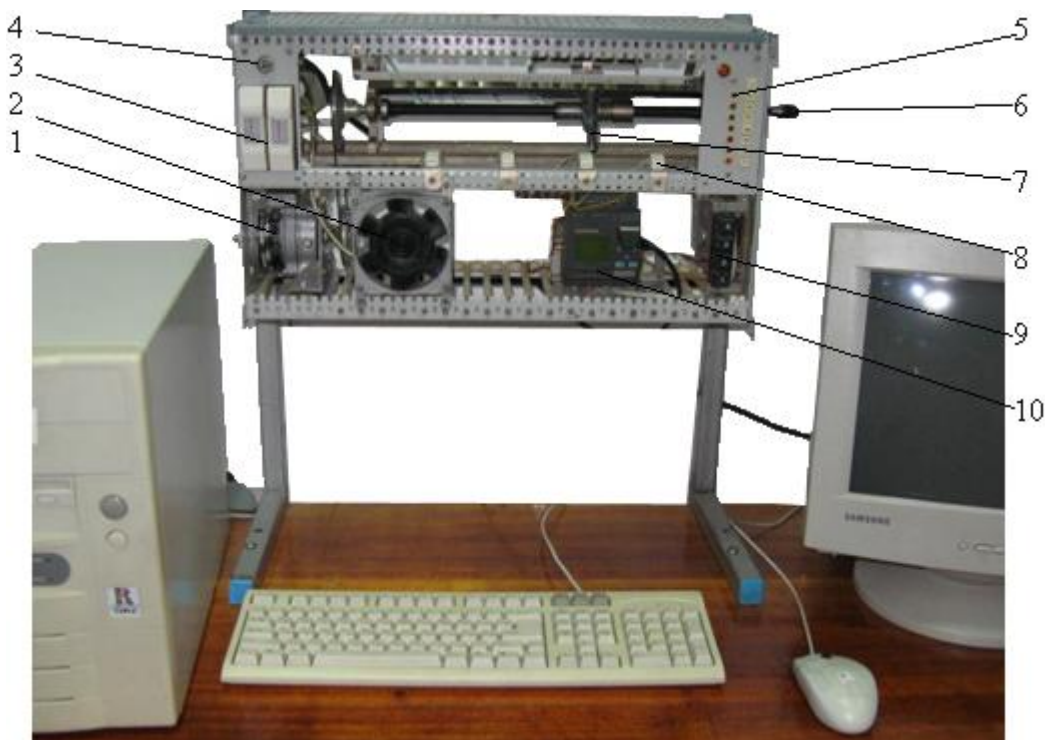
## 12 ЛАБОРАТОРНІ РОБОТИ

### 12.1 Лабораторна робота №1

Програмування та дослідження роботи контролера Logo!Basic.

**Мета:** ознайомитися з будовою LOGO!Basic, придбати навички програмування і роботи з контролером.

Лабораторна установка, рис.12.1, призначена для програмування і дослідження роботи контролера LOGO!Basic.



1 - електродвигун каретки; 2 – вентилятор; 3 - кнопки електричного управління двигуном каретки; 4-вимикач мережі 220В; 5 - світлові індикатори положення каретки; 6 - штурвал, ручного переміщення каретки; 7 - каретка з підсвічуванням і магнітним датчиком; 8 - датчики положення каретки; 9 - кнопки формування дискретного сигналу на входах контролера; 10 - контролер LOGO!Basic.

Рисунок 12.1 – Лабораторна установка

На лабораторній установці за допомогою кнопок 3 можна вручну керувати рухом каретки 7. Автоматичне управління здійснюється мікропроцесорним контролером LOGO!Basic. Відповідно до розробленої користувачем програми контролер керує роботою електродвигуна 1, який переміщує каретку уздовж чотирьох датчиків положення 8. При досягненні кареткою певного датчика на світловому табло 5 засвічується відповідна лампочка, а на один з входів контролера (I1-I4) подається сигнал. Кнопки 9 підключені до входів контролера (I5-I8). Натискаючи кнопки, на контролер

подається сигнал, який можна використовувати у програмі користувача. Живлення контролера здійснюється напругою  $=24\text{В}$  постійного струму.

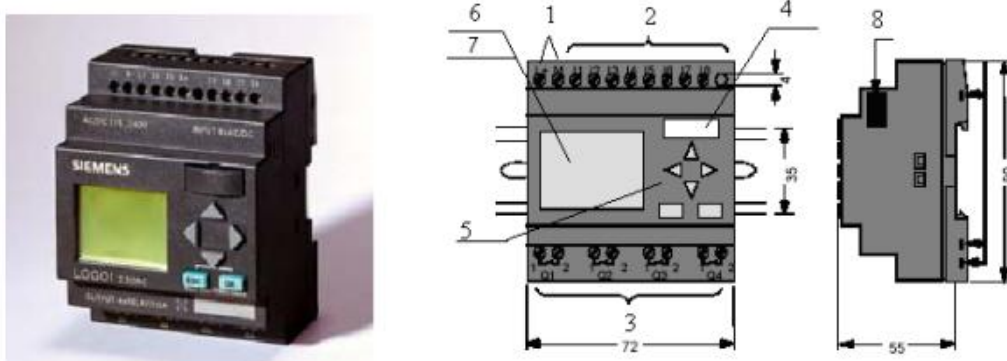
При досягненні кареткою крайнього положення спрацьовують кінцеві вимикачі, які зупиняють двигун. У крайньому лівому положенні каретки вмикається вентилятор 2 [60].

### **Контролер LOGO! Basic**

LOGO!Basic це універсальний логічний модуль фірми Siemens, який забезпечує інженерні рішення в діапазоні від нескладних систем регулювання до складних завдань автоматизації. Існує два типи контролерів - LOGO! Basic і LOGO! Pure.

В лабораторній установці використовується LOGO!Basic, рис.12.2. Контролер має 8 дискретних входів, 4 дискретних виходи, вбудований рідкокристалічний дисплей і клавіатура, а також гніздо для установлення модуля пам'яті або підключення кабелю ПК для програмування з комп'ютера.

На панелі управління розташовані клавіші. За допомогою клавіш ◀, ▶, ▼ і ▲ здійснюється управління курсором. Коли курсор у вигляді знака підкреслення – він переміщується по дисплею, коли курсор у вигляді суцільного прямокутника - здійснюється вибір сполучного елемента або блока.



1 - клеми для підключення джерела живлення; 2 - входи; 3 - виходи; 4 - гніздо для підключення EOM; 5 - панель управління; 6 - рідкокристалічний дисплей; 7 - індикатор стану RUN/STOP; 8 - інтерфейс розширення.

Рисунок 12.2 – Контролер LOGO! Basic

Для підтвердження вибраного сполучного елемента або блока використовується клавіша **OK**.

Для виходу з режиму програмування і повернення програми на один крок назад призначена клавіша **ESC**.

Окрім того в контролері є 30 вбудованих функцій, які згруповані у бібліотеці логічних (GF) і спеціальних (SF) функцій.

Програмується контролер з лицевої панелі, а також з EOM за допомогою спеціального пакета LOGO! Soft Comfort.

Пакет LOGO! Soft Comfort дозволяє розробляти на комп'ютері програми в графічній формі мовами - функціональних блокових діаграм (FBD) і релейно-

контактних схем (LAD), а також налагоджувати їх в режимі емуляції і документувати.

При програмуванні використовуються різні бібліотечні елементи, які згруповані у списки:

- ↓Co - список сполучних елементів;
- ↓GF - список основних функцій;
- ↓SF - список спеціальних функцій;
- ↓BN - список блоків готових до використання в комутаційній програмі.

**Константи і сполучні елементи - Co** - це входи, виходи, біти пам'яті, фіксовані рівні напруги (константи) і відкриті сполучні елементи.

При програмуванні завдання лабораторної роботи використовуються тільки входи, виходи і відкриті сполучні елементи.

Дискретні входи позначаються літерою **I**. Номери дискретних входів (I1, I2, I3, I4, I5, I6, I7, I8) відповідають номерам вхідних клем на LOGO! Basic.

Входи I7 і I8 можуть бути також запрограмовані для використання як аналогові входи. Тоді вони позначаються як AI1 і AI2.

Дискретні виходи позначаються літерою **Q**. Номери виходів (Q1, Q2, Q3, Q4) відповідають номерам вихідних клем на LOGO! Basic.

Відкриті сполучні елементи позначаються символом «х». Вони застосовуються, якщо входи і виходи не використовуються.

**Основні функції – GF** - це прості логічні елементи булевої алгебри, які використовуються в комутаційній програмі. Входи окремих основних функцій можна інвертувати, тобто комутаційна програма перетворить логічну «1» на відповідному вході в логічний «0». Якщо ж на вході встановлений «0», то програма змінює його на логічну «1». Список деяких основних функцій наведено у табл.12.1.

Таблиця 12.1 Основні функції – GF

Зображення на комутаційній схемі	Зображення в LOGO!	Найменування
<p>Послідовне з'єднання замкаючих контактів. Якщо який-небудь вхід цього блоку не підключений (х), то для цього входу <math>x=1</math>.</p> 		AND (І)
<p>Паралел. з'єднання замкаючих контактів. Якщо який-небудь вхід цього блоку не використовується (х), то для цього входу <math>x=0</math></p> 		OR (АБО)
<p>Контакт, що розмикається</p> 		NOT(заперечення, інверсія)

**Спеціальні функції – SF** включають у свій склад таймери, лічильники і інші функціональні блоки. Вони можуть зберігати і мають можливості параметризації (Rem). Опис деяких спеціальних функцій наведено у табл.12.2.

Таблиця 12.2

## Спеціальні функції – SF

Символ в LOGO!	Підключення	Опис
<p>Затримка вмикання і запам'ятовування</p>  <p>За входнім імпульсом починається відлік встановленого при параметризації інтервалу часу, після закінчення якого вихід встановлюється в 1.</p>	Вхід Trg	Сигналом на вході Trg (trigger = запустити) запускається відлік часу затримки включення
	Вхід R	Сигнал на вході R скидає час затримки включення і встановлює вихід в 0
	Параметр	T - це час, після закінчення якого вмикається вихід (стан виходу змінюється з 0 в 1). Збереження: /= Збереження відсутнє R = Стан зберігається
	Вихід Q	Q - включається після закінчення часу затримки T.
<p>Реверсивний лічильник</p>  <p>При отриманні входнього імпульсу внутрішнє значення лічильника, залежно від параметризації, збільшується або зменшується на 1. Досягши заданих порогових значень вихід встановлюється або скидається. Напрямок рахунку може бути змінений сигналом на вході Dir.</p>	Вхід R	Сигнал на вході R скидає внутрішнє рахункове значення і вихід в 0
	Вхід Cnt	На вході Cnt лічильник рахує зміни із стану 0 в стан 1. Зміни із стану 1 в стан 0 не рахуються.
	Вхід Dir	Напрямок рахунку встановлюється на вході Dir Dir = 0: прямий відлік Dir = 1: зворотний відлік.
	Параметр	On: поріг вмикання Діапазон значень 0.999999 Off: поріг вимкнення Діапазон значень 0.999999 Збереження для внутрішнього розрахункового значення Cnt: /= Збереження відсутнє R = Стан зберігається.
<p>Програмний вимикач</p>  <p>Ця спеціальна функція діє як механічна кнопка або як вимикач.</p>	Вхід En	Зміна стану на вході En (Enable = розблокувати) з 0 на 1 включає вихід Q, якщо настройка 'Switch=On в режимі параметризації була підтверджена натисканням ОК [Вимикач активізований]
	Параметр	<i>Режим програмування:</i> Вибір - функція використовується як кнопка, що вмикається на один цикл, або як вимикач. Start - стан увімкнений або вимкнений, який приймається в першому циклі при запусканні програми, якщо збереження вимкнене. Збереження: /= Збереження відсутнє R = Стан зберігається. Режим параметризації (режим RUN): Switch: вмикає/вимикає кнопку або вимикач.



	Вихід Q	<p>Вмикається, якщо <math>En = 1</math> і <math>Switch=On</math> було підтверджено натисненням ОК.</p> <p>Вихід скидається в 0 :</p> <ul style="list-style-type: none"> <li>• При зміні стану на вході <math>En</math> з 1 на 0.</li> <li>• Якщо функція параметризована як кнопка, і після вмикання пройшов один цикл.</li> <li>• Якщо в режимі параметризації для параметра 'Switch [Вимикач]' було вибрано положення Off [Деактивізований], і цей вибір був підтверджений натисненням ОК. Якщо збереження не включене, то після втрати живлення вихід Q ініціалізується відповідно до настройки параметра «Start».</li> </ul>
--	---------	--

Для створення логічних зв'язків між функціональними блоками призначені сполучні елементи, що вказані біля їх входів. Серед них:

- **S (Set = встановити)**- сигнал на вході S встановлює на виході логічну «1».
- **R (Reset = скинути)**- вхід скидання R має пріоритет над всією рештою входів і перемикає виходи в «0».
- **Trg (Trigger = запустити)**- цей вхід використовується для запуску функції на виконання.
- **Cnt (Count = вважати)**- цей вхід використовується для підрахунку імпульсів.
- **Dir (Direction = напрям)**- цей вхід використовується, наприклад, для установлення напрямку, в якому повинен рахувати лічильник.
- **En (Enable = вирішити)**- цей вхід розблоковує функцію, що виконується блоком. Якщо на цьому вході «0», то інші сигнали блоком ігноруються.
- **Inv (Invert = інвертувати)**- коли цей вхід активізований вихідний сигнал блока інвертується.
- **Ral (Reset all = скинути все)**- скидаються всі внутрішні значення.

Окрім зазначених, у блоках є параметричні входи, до яких сигнали не прикладаються:

- **Par (Parameter = параметр)**- на цьому вході встановлюються параметри (часи, пороги вмикання і вимикання і т.д.).
- **No (Nocken = шаблон)**- на цьому вході встановлюється шаблон часу.
- **P (Priority = пріоритет)**- на цьому вході встановлюються пріоритети.

У деяких спеціальних функціях є можливість параметризувати значення часу T. При завданні часу значення, що вводяться, залежать від встановленої бази часу- s (секунди), m (хвилини), h (години)[61].

### ***Виконання лабораторної роботи***

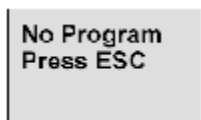
Розробити програму руху каретки за певними позиціями, які визначаються датчиками положення 8 (з першої на третю, з третьої на другу, з другої на четверту і знову на першу). При цьому на кожній позиції каретка має витримувати п'ятисекундну паузу .

Для програмування використовуються:

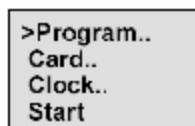
- функціональні блоки:
  - реверсивний лічильник;
  - таймер із «затримкою вмикання та запам'ятовуванням»;
  - блок логічний «АБО».
- входи контролера:
  - I1-I4, до яких підключені герконові контакти відповідних датчиків позиції каретки 8;
  - I5 - I8, до яких підключені кнопки вводу дискретних сигналів 9.

### *Програмування з лицевої панелі*

Для створення програми і вводу її в пам'ять контролера необхідно увімкнути лабораторну установку. Якщо пам'ять контролера вільна на дисплеї з'явиться повідомлення:



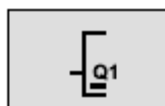
якщо ні, тоді:



Увійти в меню "Program." і очистити пам'ять контролера - "Clear Prg->Yes". Натиснути клавішу ESC, щоб перевести LOGO! у головне меню.

Клавішею ▲ або ▼ перемістити курсор «>» на «Program..» і натиснути ОК. LOGO! перейде в режим програмування >Edit.

Для створення FBD-програми перемістити курсор «>» на «Edit..» і натиснути ОК. Вибрати «Edit Prg» —> ОК і на дисплеї LOGO! з'явиться перший вихід контролера Q1.



Після цього можна вводити FBD-програму керування роботою каретки за заданим алгоритмом, рис.12.3.

Для початку руху каретки необхідно встановити реверсивний лічильник, який при подачі сигналу на його вхід вмикає двигун. Для цього:

- кнопкою ОК перейти у режим вводу, курсор отримає вигляд миготливого суцільного прямокутника;
- кнопкою ▼ знайти список SF (спеціальні функції) і натиснути ОК, LOGO! відобразить перший блок зі списку спеціальних функцій;
- кнопкою ▲ або ▼ вивести на дисплей блок реверсивного лічильника;
- натиснути ОК для підтвердження вибору.

За замовчуванням блок отримає номер B1.

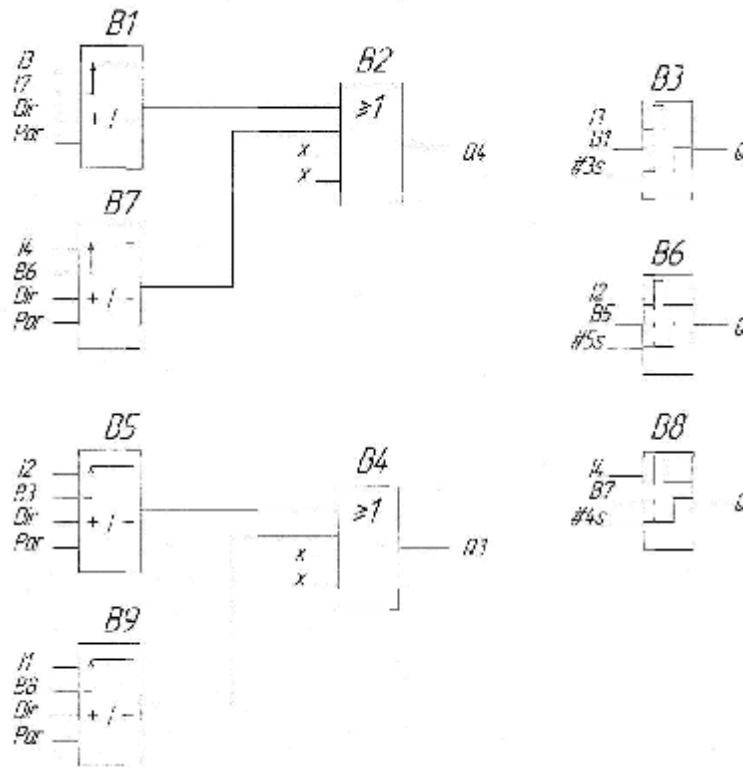
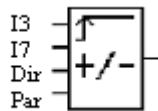


Рисунок 12.3 – FBD-програма керування роботою каретки

При створенні блока курсор у вигляді знака підкреслення знаходиться на першому вході реверсивного лічильника - R. Натиснути ОК і кнопкою ▼ вибрати список Co. Натиснути ОК і кнопкою ▼ вибрати вхід I3. Натиснути ОК, перейти на другий вхід – Cnt і за аналогією з I3 встановити вхід I7.



Третій вхід залишається без зміни.

Щоб зробити уставку порога вмикання лічильника, необхідно перейти до четвертого входу «Par» і натиснути ОК. Кнопкою ► перейти на останній знак «0» альтернативи “On” і за допомогою ▲ встановити значення 1. Після завершення параметризації блока натиснути кнопку Esc і в меню **Edit Prg** кнопкою ▼ перейти на вихід контролера Q4, натиснути ОК.

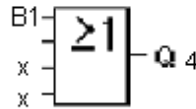
Для подачі на вихід контролера Q4 керувального сигналу необхідно створити інший блок – логічне АБО. Для цього:

- перейти в режим вводу, натиснувши кнопку ОК. Курсор має вигляд миготливого суцільного прямокутника;
- кнопкою ▼, вибрати список GF (основні функції) і натиснути ОК. LOGO! відобразить перший блок зі списку основних функцій;
- кнопкою ▼ або ▲ вибрати на дисплеї блок логічного АБО;
- натиснути ОК для підтвердження вибору і блок отримає номер B2.

Щоб сконфігурувати перший вхід блока В2 з виходом блока В1 треба натиснути ОК;

- кнопкою ▼ або ▲ вибрати список створених блоків ВN і натиснути ОК;
- кнопкою ▼ вибрати блок В1 і натисніть ОК.

Біля решти входів можна записати символ «х», вибравши його зі списку Со. На дисплеї з'явиться зображення:



Щоб повернутися в меню **Edit Prg**, необхідно натиснути кнопку Esc.

Для доповнення програми таймером «із затримкою вмикання і запам'ятовуванням» необхідно:

- кнопкою ▼ перейти на вільний вихід контролера Q2 і натиснути ОК;
- за допомогою ▼ вибрати список SF (спеціальні функції) і натиснути ОК.

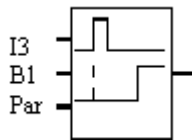
LOGO! відобразить перший блок у списку спеціальних функцій;

- натискати кнопку ▼ або ▲ поки на дисплеї не з'явиться блок таймера «із затримкою вмикання і запам'ятовуванням»;
- натиснути ОК і блок автоматично отримає номер В3. Курсор у вигляді знака підкреслення знаходиться на першому вході - Trg.

Для підключення входів блока таймера В3:

- натиснути ОК;
- користуючись ▼ або ▲, вибрати список Со і натиснути ОК;
- за допомогою клавіші ▼ вибрати вхід І3 і натиснути ОК.

Другий вхід R сконфігурувати з виходом блока В1 (для запуску таймера) відповідно до того, як це робилося для блока В2 .



Встановити на третій вхід Par значення часу затримки -3с.

Для цього:

- перейти на вхід Par і натиснути ОК;
- враховуючі те, що формат часу представлений у вигляді «секунди: мілісекунди», клавішею ► перевести курсор на друге знакомісце і клавішею ▲ встановити значення часу - 3 с.

Після завершення параметризації блока натиснути кнопку Esc. В меню **Edit Prg** кнопкою ▼ перейти на вихід контролера Q3 і натиснути ОК.

Для подачі керуючого сигналу на вихід контролера Q3, за аналогією з першим блоком АБО, створити ще один блок логічного АБО. Даному блоку автоматично присвоюється номер В4.

Для запуску двигуна після завершення роботи таймера В3, необхідно на вільному виході контролера Q5, за аналогією з першим реверсивним



лічильником, створити другий реверсивний лічильник, який автоматично отримає номер В5, і сконфігурувати його входи. На перший вхід блока встановити І2, на другий - В3, а на вході Par вибрати Оп=000001. Вихід лічильника В5 сконфігурувати з першим входом блока В4. Для цього:

- натиснути кнопку Esc і в меню Edit Prg кнопкою ▼ знайти блок В4;
- відкрити цей блок, двічі натискаючи кнопку ◀;
- на вході In1 зі списку ВN вибрати номер блока В5. Натиснути послідовно кнопки ОК і Esc.

Щоб забезпечити затримку у русі каретки при досягненні наступної позиції, за аналогією створення блока В3, вибрати таймер «із затримкою вмикання і запам'ятовуванням» з виходом Q6. Йому автоматично присвоїться номер В6. На перший вхід встановити І2, на другий - В5, а час затримки вибрати - 5с.

Для запуску двигуна після завершення роботи таймера В6, за аналогією з блоком В1, створити реверсивний лічильник з виходом Q7 і сконфігурувати його входи. Блок автоматично отримає номер В7.

На перший вхід блока встановити значення І4, на другий вхід - значення В6, а на вході Par вибрати Оп=000001. Вихід лічильника В7 сконфігурувати з другим входом В2 (блок логічне АБО), як це відбувалося при конфігуруванні блоків В5 і В4.

Щоб забезпечити затримку каретки на наступній позиції, вибрати таймер «із затримкою вмикання і запам'ятовуванням» з виходом Q8. Він автоматично отримає номер В8. Встановити на перший вхід - І4, на другий - В7, а час затримки - 5с.

Для завершення програми, створити блок - реверсивний лічильник В9 з виходом Q9, який забезпечить повернення каретки в початкове положення. До його входів приєднати І1 і В8, а на вході Par вибрати Оп=000001. За аналогією з блоками В5 і В4 сконфігурувати вихід блока В9 з другим входом блока В4 (логічне АБО).

Після завершення конфігурування тричі натиснути Esc, кнопкою ▼ перевести курсор на «Start» і натиснути ОК.

Для запуску програми необхідно, на лабораторній установці серед кнопок формування дискретного сигналу 9, натиснути кнопку 3, яка створює дискретний сигнал на вході контролера. Після цього каретка починає рух відповідно до наведеного алгоритму.


Для зупинення програми треба натиснути клавішу ESC і зі списку, що з'явиться на дисплеї, вибрати команду Stop і натиснути ОК[61].

### ***Програмування у пакеті LOGO! Soft Comfort***

Відкрити програму LOGO! Soft Comfort V4.0 подвійним клацанням по ярлику програми на робочому столі. Для створення проекту і вибору мови програмування, увійти в меню «Файл/Новый/Диаграмма (схема) функціональних блоків (FBD)) і на екрані ЕОМ з'являться два поля - панель інструментів і робоче поле проекту.

1. Встановити на робочому полі реверсивний лічильник. Для цього:

- в панелі інструментів, що ліворуч, натиснути кнопку SF, відкриється список спеціальних функцій;

- курсором вибрати блок реверсивного лічильника  і натиснути ліву клавішу миші;

- повторно клацнути лівою клавішею миші на робочому полі, де має бути встановлений цей блок. Йому автоматично присвоюється номер В001.

1.1 Для конфігурування входів блока В001 з входами контролера І3 і І7, необхідно:


- натиснути на кнопку , що зліва на панелі інструментів, відкриється список стандартних елементів;

- зі списку вибрати блок «Логічне І» і двічі встановити його на робочому полі, натискаючи лівою клавішею миші спочатку на вибраному елементі, потім у місцях де має бути встановлений цей блок.

При подвійному натисканні лівої клавіші миші на блоці з'явиться діалогове вікно, в якому на вкладці «Параметр» вибрати номер входу контролера, з якого даний блок отримує сигнал.

Для першого блока вибрати вхід контролера І3, для іншого - вхід І7.



1.2 Щоб з'єднати перший вхід блока В001 з входом контролера І3:

- натиснути кнопку , що зліва на панелі інструментів, потім клацнути лівою клавішею миші на вході І3 і не відпускаючи її підвести курсор до входу В001.

Так саме з'єднати вхід І7 з іншим входом блока В001.

Клацнути лівою клавішею миші 2 рази на блоці В001. Відкриється діалогове вікно, на вкладці «Параметр» в полі Оп встановити 1.


2. Для встановлення блока АБО:

- зліва на панелі інструментів натиснути кнопку GF і на робочому полі встановити блок , якому автоматично надається номер В002. До його першого входу приєднати вихід блока В001, користуючись кнопкою .

Для підключення програми до виходу контролера Q4, який забезпечує рух каретки вперед, необхідно встановити блок Q на робочому полі. Для цього:

- зліва на панелі інструментів натиснути кнопку  і вибрати блок Q;

- двічі натиснути ліву клавішу миші на цьому блоці і у діалоговому вікні, що з'явиться, вибрати Q4. Підключити вхід Q4 до виходу блока В002 (логічне АБО).

Для створення керуючого сигналу на каретку після завершення роботи таймера, треба встановити на робочому полі реверсивний лічильник, додати блок І4 і з'єднати його з першим входом блока В003 (див. п.1.1, 1.2). Вихід блока В003 з'єднати з другим входом блока В002 (логічний АБО), використовуючи кнопку .

3. Для забезпечення паузи між рухами каретки, встановити таймер «із затримкою включення і запам'ятовуванням». Для цього :

- зліва на панелі інструментів натиснути кнопку SF, з'явиться список спеціальних функцій;

- вибрати блок таймера «із затримкою включення і запам'ятовуванням»



, натиснути ліву клавішу миші на ньому і повторно клацнути на робочому полі, де має бути встановлений цей блок. Автоматично він отримає номер В004.

До входів таймера В004 підключити вихід контролера І3 і реверсивний лічильник В001. Для цього:

- натиснути праву клавішу миші на першому вході (Trg) таймера В004. У контекстному меню, що з'явиться, вибрати "Соединить блоки", а у його вікні - вхід контролера І3;

- аналогічно до другого входу таймера (R) приєднати лічильник В001;


- двічі клацнути лівою клавішею миші на таймері В004, відкриється діалогове вікно, в якому встановити час затримки вмикання - 3с.

Аналогічно на робочому полі встановити ще 2 реверсивних лічильника, яким автоматично присвояться номери В005 і В006. Щоб забезпечити рух каретки після паузи, необхідно приєднати до першого входу блока В005 вхід контролера І1, до першого входу лічильника В006 – вхід контролера І2 (див. п.1.1), а до другого входу лічильника В006 – вихід таймера В004.

Додати на робочому полі блок «АБО» (див. п.2), який автоматично отримає номер В007. З'єднати перші 2 входи з виходами лічильників В005 і В006 (див. п.1.2.), відповідно.

Для конфігурування програми з виходом контролера Q3, який керує рухом каретки в іншому напрямі, необхідно на робочому полі встановити блок Q.

Для цього:

- зліва на панелі інструментів натиснути на кнопку , вибрати блок Q і двічі клацнути лівою клавішею миші;

- для подачі сигналу на вихід контролера, у вікні, що з'явиться, вибрати Q3 і приєднати його вхід до виходу блока В007(логічне АБО) .

Для здійснення затримки у роботі каретки, на робочому полі встановити таймер (див. п.3), який автоматично отримає номер В008. До входу Trg приєднати вхід контролера І4, до входу R - вихід блока В003. Встановити час затримки - 5с. Вихід таймера приєднати до другого входу реверсивного лічильника В005.

Для здійснення затримки в іншому положенні каретки, встановити ще один таймер (див. п.3) з номером В009. До входу Trg приєднати вхід контролера І2, до входу R - вихід блока В006. Встановити затримку - 5 сек. Вихід таймера приєднати до другого входу реверсивного лічильника В003[62].

Загальний вигляд програми керування рухом каретки наведено на рис.12.4.

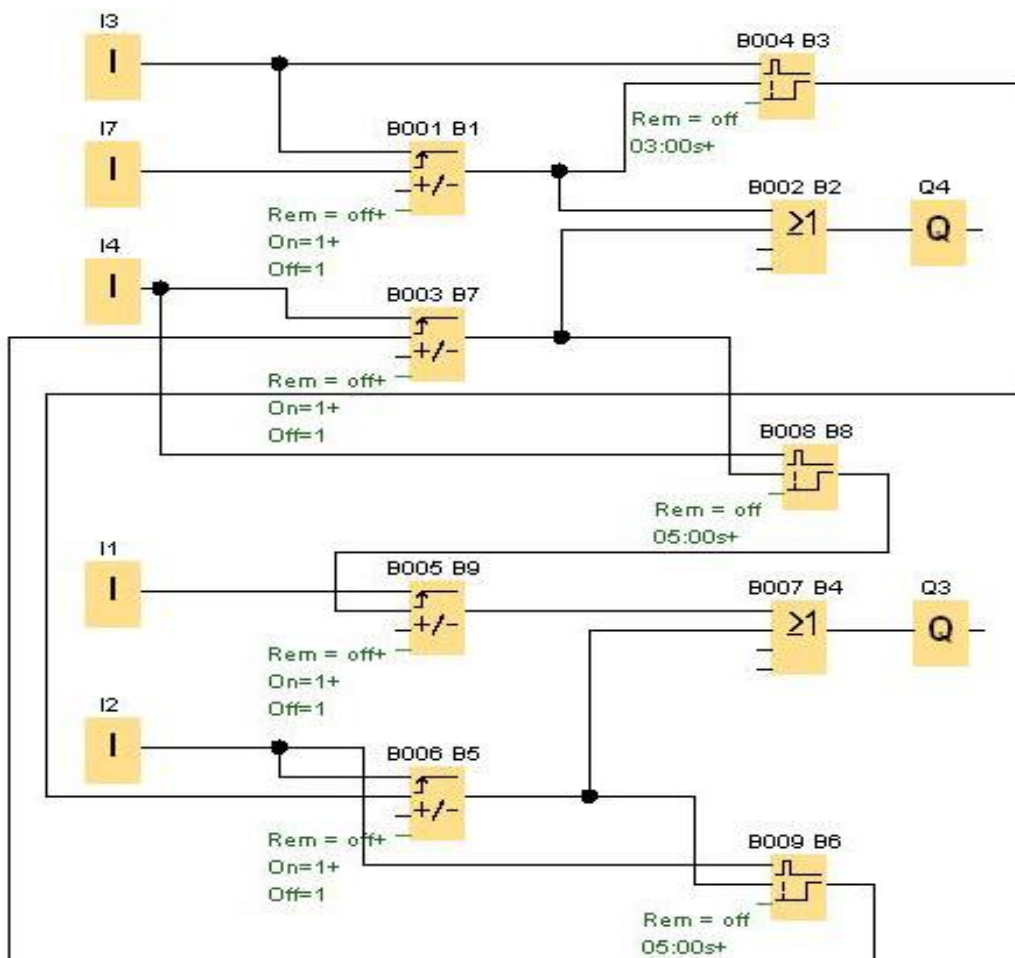



Рисунок 12.4 – Програма керування рухом каретки

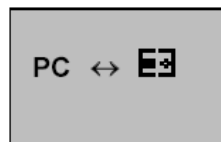
Для перевірки роботи програми в панелі інструментів натиснути кнопку «Емуляція» . Активізуючи входи контролера у програмі, за допомогою їх символів, що унизу робочого поля, і лівої клавіші миші, переконатися в правильності виконання завдання. При цьому сполучні лінії між блоками відображаються різними кольорами: активні - червоним кольором, неактивні - синім.

Щоб завантажити розроблену програму в контролер, необхідно на панелі інструментів натисніть кнопку .

Для запуску програми на дисплеї контролера перевести курсор у позицію «Start» і натиснути ОК, а на лабораторній установці серед кнопок формування дискретного сигналу 9, натиснути кнопку 3, яка створює дискретний сигнал на вході контролера. Після цього каретка починає рух відповідно до наведеного алгоритму.

При необхідності зупинення програми треба натиснути кнопку Online-тест . Перевод контролера у режим Stop здійснюється кнопкою .

Коли в стані STOP програма завантажується в контролер або вилучається з нього, на дисплеї автоматично з'являється зображення:



LOGO! = LOGO!

### ***Завдання для самостійної роботи***

Скоригувати розроблену програму, передбачивши екстрену зупинку каретки від натискання кнопки 4, яка створює дискретний сигнал на восьмому вході контролера, а також підрахунок кількості її рухів від позиції до позиції.

### ***Зміст звіту***

Звіт має містити назву роботи, опис лабораторної установки, короткі відомості про контролер, програму керування кареткою, результати самостійної роботи і висновки про виконану роботу.

### ***Контрольні запитання***

1. Для чого призначений контролер LOGO!?
2. Яка конструкція контролера?
3. Якими мовами програмується контролер LOGO!?
4. Яке призначення мають клавіші на лицевій панелі LOGO!?
5. Як здійснюється програмування контролера з ЕОМ?
6. Що містить бібліотека контролера?
7. Чим характеризуються логічні елементи «І» і «АБО»?
8. Як програмується часова затримка?
9. У чому суть FBD-мови?
10. Що собою являє лабораторна установка?

## **12.2 Лабораторна робота №2**

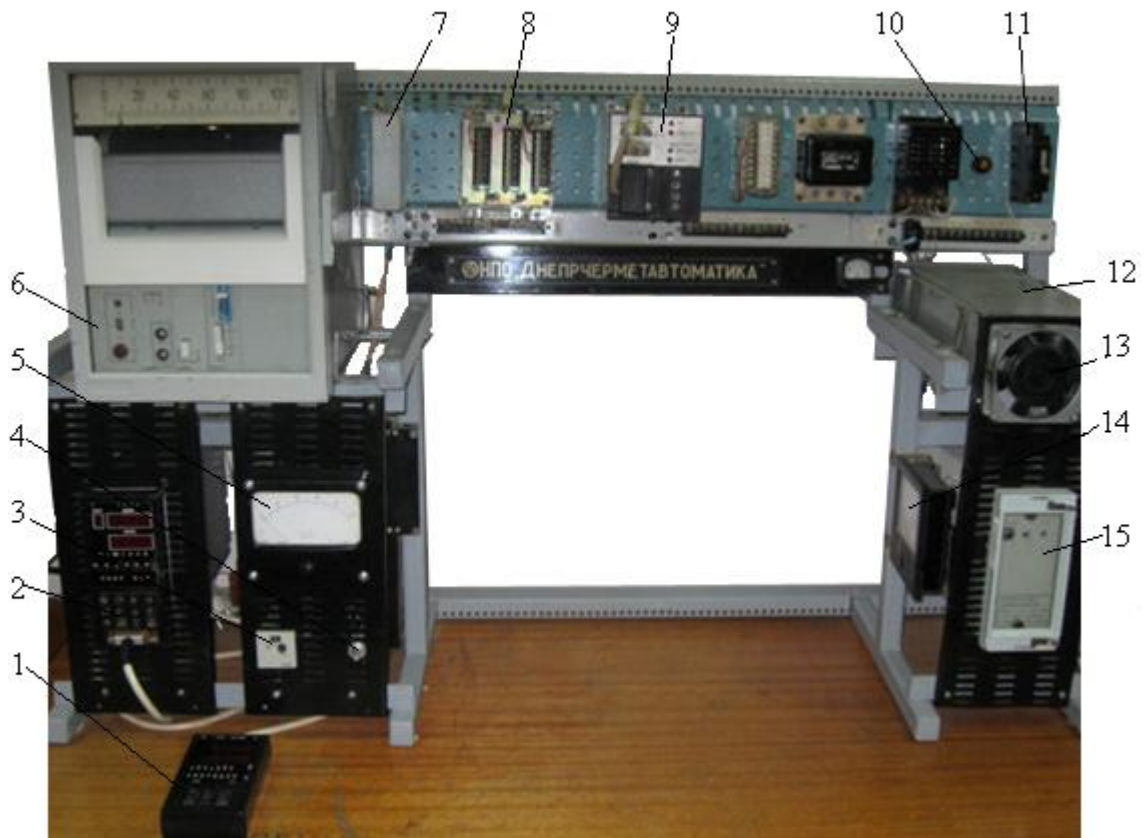
Програмування та дослідження роботи контролера Реміконт Р-130

**Мета:** ознайомитися з будовою Реміконта Р-130, придбати навички програмування і роботи з контролером.

Лабораторна установка, рис.12.5, призначена для дослідження роботи регульовального мікропроцесорного контролера Реміконт Р-130, який підтримує температуру в печі за ПД –аналоговим алгоритмом.

Блок контролера БК-1 живиться напругою 24В від джерела БП-1. Температура печі вимірюється двома термопарами ТП. Сигнал з однієї термопари надходить на вторинний прилад РП-160, а з другої термопари - на нормуючий підсилювач Ш-72, де перетворюється в уніфікований токовий сигнал 0-5 мА. З Ш-72 сигнал через клемно-блоковий з'єднувач КБС-3 надходить на аналоговий вхід контролера БК-1 (ПЗО А).

Є два режими дослідження роботи контролера, які вибираються за допомогою перемикача «Авт/Руч».



*1 - пульт настройки ПН-1; 2 - блок контролера БК-1; 3 - задавач РЗД; 4 - перемикач режимів роботи, Авт/Ручн; 5 - міліамперметр вихідного сигналу контролера; 6- прилад РП-160, що показує температуру в печі; 7 - клемно-блоковий з'єднувач КБС-1; 8- клемно-блоковий з'єднувач КБС-3; 9 - блок живлення БП-1; 10 - індикатор наявності вхідної напруги; 11 - автомат вхідної напруги; 12 – піч; 13 - вентилятор; 14 - амперметр струму навантаження; 15 - нормуючий перетворювач Ш-72 .*

Рисунок 12.5 – Лабораторна установка

При дослідженні ПІД-аналогового алгоритму управління в режимі «Руч» сигнал з РЗД в якості вхідного надходить на перший аналоговий вхід БК-1. Відповідно до цього сигналу БК-1 формує ПІД- аналоговий керувальний вплив, який через КБС-3 надходить до міліамперметра, що відслідковує часову характеристику ПІД- закону.

При автоматичному керуванні температурою печі «Авт» сигнал від термопари через нормуючий пристрій НУ надходить до КБС-3, а звідти до БК-1. Керуючий сигнал з БК-1 через КБС-3 подається на блок керування тиристорами, який змінює силу напругу на електричній нагрівальній печі. Струм навантаження показує амперметр.

### **Реміонт Р-130**

Р-130 – це комплекс технічних засобів, у складі якого є блок контролера БК-1 з лицевою панеллю і пультом настройки ПН-1, засоби зв'язку з об'єктом, а також блок живлення БП-1.



Блок БК-1 обробляє інформацію у цифровій формі, організує програму усіх алгоритмів керування, забезпечує програму обслуговування пульта настройки ПН-1 і лицевій панелі ПЛ, доставляє потрібні алгоритми із ПЗП. Ці алгоритми об'єднуються у систему заданої конфігурації, і в них встановлюються необхідні параметри настройки. За допомогою лицеві панелі ПЛ і пульта настройки ПН-1 забезпечується оперативне керування до 4 контурів автоматичного регулювання або логіко-програмного та покрокового дискретного управління.

Р-130 – виріб, що проектно-компонується. Він має базовий комплект апаратури, який поставляється завжди незалежно від об'єму розв'язуваної задачі і комплект, що проектно-компонується. Склад цього комплекту залежить від номенклатури та кількості каналів вводу-виводу інформації. Вигляд лицеві панелі, модифікація центрального блока і склад додаткових блоків визначаються картою замовлення.

Основна частина блока БК-01 містить:

- модуль контролю і програмування МКП;
- модуль процесора ПРЦ10;
- модуль стабілізованої напруги МСН10.

Змінна частина складається з двох посадкових місць для пристроїв зв'язку з об'єктом ПЗОА і ПЗОБ, куди можуть бути встановлені один або два з трьох модифікацій модулів ПЗО:

- модуль аналогових сигналів МАС;
- модуль аналогових і дискретних сигналів МДА;
- модуль дискретних сигналів МСД.

Реміконт Р-130 є програмованим пристроєм. Структура, що створюється в ньому програмно описує інформаційну організацію контролера і характеризує його як ланку системи керування, яка одержала назву віртуальної - тобто не існуючої як фізичне тіло. Ця віртуальна структура реалізується за допомогою, як апаратних, так і програмних засобів.

Апаратну складову віртуальної структури створюють апаратура вводу-виводу інформації, апаратура оперативного керування та настройки, апаратура інтерфейсного каналу, а програмну - алгоритмічні блоки і бібліотека алгоритмів.

Реміконт Р-130 має інтерфейсний канал, за допомогою якого контролери можуть об'єднуватися у локальну керувальну мережу "Транзит". Мережа "Транзит" має кільцеву конфігурацію. Контролери, що приєднані до мережі "Транзит", мають унікальну адресу, яка задається ПЛК у процесі його конфігурування, при заповненні системних параметрів.

Решта інформації про фізичну, віртуальну та мережну структуру контролера приведена у відповідному розділі цього навчального посібника.

### Програмування контролера

Програмування Р-130 здійснюється мовою функціональних блоків. Ця мова не процедурна, тобто при конфігуруванні контролера не задається послідовність виконання операцій.

Для технологічного програмування, настроювання і контролю Р-130 використовується виносний пульт настройки ПН-1, рис.12.6. Він вмикається до

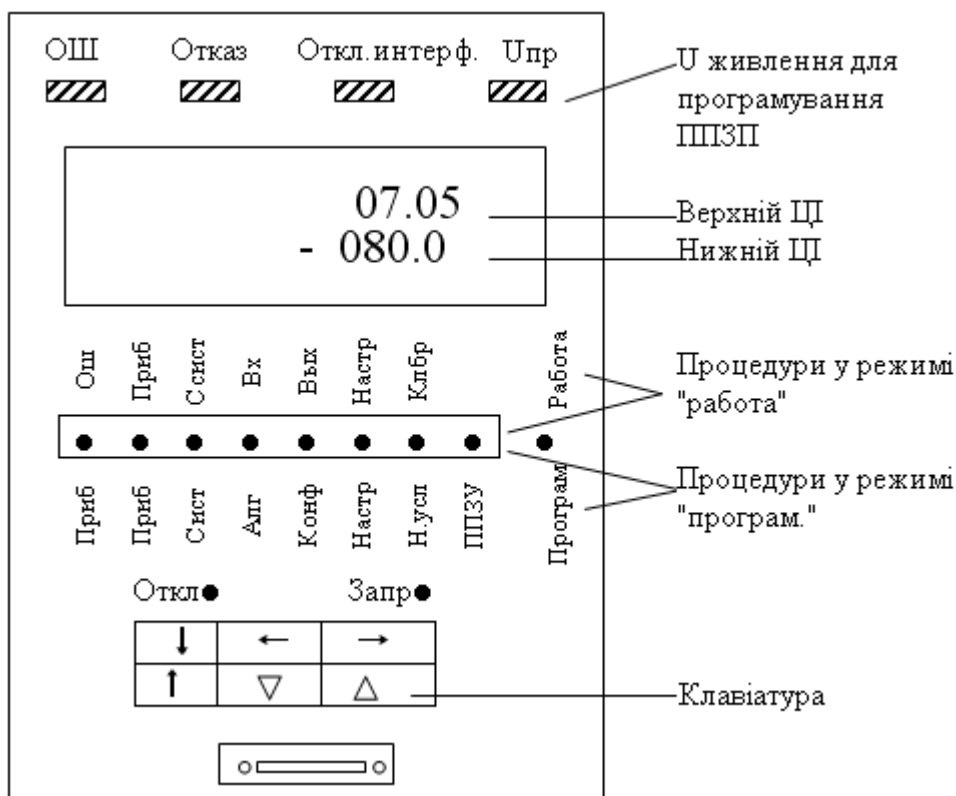


Рисунок 12.6 – Пульт настройки ПН-1

блока контролера БК-1 через рознім, що на його лицевій панелі. Контролер може знаходитися в одному з двох режимів - програмування і роботи.

У режимі роботи ламповий індикатор "робота" горить блимаючим світлом. При відсутності відмови частота блимання цього індикатора 3 рази на 2 секунди (при наявності – один раз на 1-2 секунди). У режимі "робота" можна контролювати параметри і сигнали, а також змінювати коефіцієнти.

У режимі програмування контролер вимкнений з контуру керування. У цьому режимі стан усіх його виходів, таймерів, лічильників і комірок накопичення заморожений, а на пульту настройки горить індикатор "прог". У режимі програмування можна змінювати усі параметри, - як контролера у цілому, так і алгоблоків.

У зв'язку з тим, що у режимі програмування контролер вимкнений з контуру управління на аварійних виходах контролера формується сигнал відмови і вимкнення інтерфейсу. Наявність цих сигналів у даному випадку

свідчить не про відмову контролера, а про те, що він вимкнений з контуру керування.

У режимі програмування задаються усі програмовані параметри контролера, що визначають його алгоритмічну структуру. Ці параметри у загальному випадку задаються “триступеневим” методом: спочатку вибирається та або інша процедура, в неї вибирається потрібна операція і у межах цієї операції устанавлюються потрібні параметри.

Контролер має 8 процедур програмування.

1. Тестування (“тест”) - виконується тестування пам’яті, інтерфейсу, сторожа циклу, клавіатури, індикаторів і вихідних ПЗО.

2. Приладові параметри (“приб”) - здійснюється обнуління або увід однієї із стандартних конфігурацій, устанавлюється комплектність, дозвіл або заборона на зміну алгоритмічної структури, діапазон часу, час циклу; контролюється ресурс ОЗП і номер бібліотеки алгоритмів у ПЗП.

3. Системні параметри (“сист”) - устанавлюється логічний номер контролера у локальній мережі і режим роботи інтерфейсу.

4. Алгоритми (“алг”) - алгоблоки заповнюються алгоритмами з показом модифікатора і масштабу часу.

5. Конфігурація (“конф”) - для входів алгоблоків-приймачів визначаються джерела сигналів.

6. Параметри настройки (“наст”) - устанавлюються значення параметрів настройки.

7. Початкові умови (“н.усл”) - устанавлюються початкові значення сигналів на виходах алгоблоків.

8. Робота з ПЗП (“ПЗУ”) - виконується запис у ПЗП, відновлення інформації із ПЗП в ОЗП, регенерація ПЗП, ПЗП.

**У процедурі “приб” (приладові параметри)** задаються і контролюються параметри, загальні для усіх алгоблоків контролера.

Після першого вмикання контролера необхідно зробити обнуління ОЗП, інакше може сформуватися сигнал відмови і виконання інших процедур (крім тестів) буде заблокованим.

При устанавленні комплектності задається код, рівний коду модифікації контролера. Цей код складається з двох цифр. Перша задає вид модуля ПЗО для входів-виходів групи А, друга – для групи Б.

У лабораторній роботі використовується контролер комплектності – 15. Тут 1 означає наявність модуля МАС, що має 8 аналогових входів і 2 аналогових виходи, а 5– наявність модуля МСД 8/8 (8 дискретних входів і 8 дискретних виходів).

У контролері необхідно встановити дозвіл на зміну алгоритмічної структури, щоб сталася можливість його конфігурування.

Для усіх алгоблоків задається молодший діапазон часу. Тоді в кожному блоці індивідуально можна вибрати один з двох масштабів часу: секунди або хвилини.

В операціях з приладовими параметрами спочатку задається номер операції, після чого встановлюються або контролюються самі приладові параметри, табл.12.3.

Таблиця 12.3 Приладові параметри

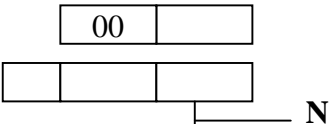
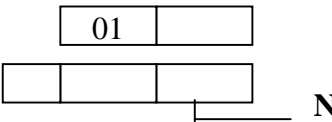
Операція	Формат індикації	Параметри
1. Обнуління або введення стандартної конфігурації		<b>N1=00</b> - заборона обнуління або введення стандартної конфігурації <b>N1=01</b> - дозвіл <b>N2=00</b> - обнуління <b>N2≠00</b> - номер стандарт.конфіг.
2. Комплектність		<b>N</b> - код комплектності
3. Заборона зміни алгоритмічної структури та діапазону часу		<b>N1=00</b> - заборона зміни алгоритмічної структури <b>N1=01</b> - дозвіл <b>N2=00</b> - молодший діапазон (с,хв.) <b>N2=01</b> - старший діапаз. (хв., год.)
4. Час циклу		Час циклу <b>T=0,2÷2,0 с</b>
5. Ресурс першої області ОЗП		<b>N</b> - ресурс у байтах
6. Ресурс другої області ОЗП		<b>N</b> - ресурс у байтах
7.Склад бібліотеки алгоритмів		<b>N1</b> - номер бібліотеки <b>N2</b> - номер версії (службова інформація)

У процедурі “сист” (системні параметри) задаються параметри, що визначають особливості взаємодії контролера з іншими пристроями за інтерфейсним каналом, табл. 12.4.

Системний номер задається у тому випадку, коли контролер працює у складі локальної мережі “Транзит”. Якщо контролер не приєднаний до мережі “Транзит”, для нього системним номером є 00.

Таблиця 12.4

Системні параметри

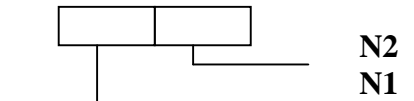
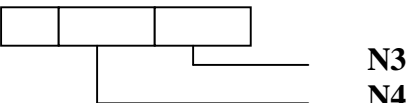
Операція	Формат індикації	Параметри
1. Системний номер		N0-15 – системний номер, (коли N=00 контролер не є у локальній мережі)
2. Режим інтерфейсу		N=01 – інформаційний режим N=02 – командний режим

У процедурі “алг” (алгоритми) відбувається “заповнення” алгоблоків алгоритмами. Після обнуління лише перший алгоблок доступний для введення алгоритму. Коли в нього буде уведений будь-який алгоритм (у тому числі і “простий” алгоритм з кодом 00), доступним стає другий алгоблок і т.д.). Якщо у будь-якому алгоблоці зміниться алгоритм, то усі раніше установлені зв’язки з ним анулюються.

Правила установлення алгоритму приведені у табл. 12.5.

Таблиця 12.5

Параметри алгоблоків

Формат індикації	Параметри		
	N1=01-99 – номер алгоблока; N2 – номер алгоритму; N3 – код модифікатора; N4 – масштаб часу		
	N4	Діапазон часу	Масштаб часу
	00	молодший	с.
	01		хв.
	00	старший	хв.
	01		год.

У процедурі “конф” (конфігурування) визначається стан кожного входу будь-якого алгоблока – зв’язаний чи вільний. На зв’язаний вхід алгоблока сигнал надходить з виходу того алгоблока, з яким даний вхід зв’язаний. На вільному вході сигнал може установлюватися оператором вручну і тоді він виконує роль параметра настройки.

Параметри настройки поділяються на дві групи – константи коефіцієнти.

Константи установлюються оператором тільки у режимі програмування і не можуть змінюватися у режимі роботи. Коефіцієнти можуть установлюватися як у режимі програмування, так і у режимі роботи.

При конфігуруванні визначається лише стан входів алгоблока-приймача, а для зв’язаних входів задається також номер алгоблока-джерела і номер його виходу.

Конкретні значення параметрів настройки на вільних входах установлюються у процедурі “настройка”.

У вихідному стані всі входи алгоблоків вільні і на них задані константи, початкові значення яких залежать від виду алгоритму.

Правила конфігурування наведені у табл. 12.6.

Таблиця 12.6 Параметри конфігурування

Стан входу	Зображення інформації	Параметри
Зв'язане		<p>N1 – номер алгоблока-приймача                      N2 – номер входу алгоблока-приймача                      N3≠00 - номер алгоблока-джерела                      N4 – номер виходу алгоблока-джерела                      С – признак інверсії                      С=“темне поле” – без інверсії                      С= “-” – інверсія</p>
Вільне		<p>N1,N2,C – те саме, що для зв'язаних входів                      N3=0 – признак вільного входу                      N4 – вид параметра настройки                      N4=00 – константа                      N4=01 – коефіцієнт</p>

У процедурі “настр.” (настройка) установлюються значення параметрів настройки, - як констант, так і коефіцієнтів. Ця процедура виконується лише для входів алгоблока, які у процедурі конфігурування були визначені як вільні.

Значення константи зберігається при переході у режим роботи і у цьому режимі не може бути змінено. Значення коефіцієнтів також зберігається при переході у режим роботи, але у цьому режимі їх можна змінювати.

Якщо параметри настройки не задаються, вони приймають початкові значення, які залежать від виду алгоритму.

Правила установлення параметрів настройки наведені у табл.12.7.

Таблиця 12.7 Параметри настройки

Зображення інформації	Параметри
	<p>N1 – номер алгоблока                      N2 – номер входу                      X – значення константи або коефіцієнта</p>



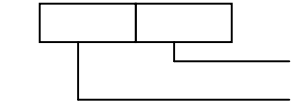
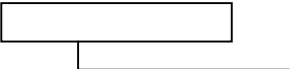
У процедурі “н.усл” (початкові умови) встановлюються значення сигналів на входах алгоблоків, з якими алгоблоки почнуть працювати при переході у режим “робота”.

Коли початкові умови не задаються, то після першого вмикання контролера вони приймають значення, яке залежить від виду алгоритму. У більшості випадків це нульові значення.

Правила встановлення початкових умов наведені у табл.12.8.

Таблиця 12.8

Початкові умови

Зображення інформації	Параметри
	<p><b>N1</b> – номер алгоблока  <b>N2</b> – номер входу  <b>X</b> – значення сигналу на виході (початкові умови)</p>
	

### **Настройка і контроль**

Усі процедури настроювання і контролю виконуються у режимі “робота”, тобто коли контролер увімкнений у контур управління. У лабораторній роботі контроль за параметрами не передбачається.

### **Виконання лабораторної роботи**

1. Ознайомитися зі змістом лабораторної роботи і зрозуміти принципи програмування контролера Р-130.

2. Увімкнути напругу лабораторної установки за допомогою автомата вхідної напруги 11, рис. 12.5.

3. До гнізда на лицевій панелі БК-1 підключити пульт настроювання. Якщо на панелі ПН-1 блимає індикатор “робота”, то натиснути разом клавіші “↓” і “→”, щоб перейти у режим “програм”. Якщо на ПН-1 індикатор “програм” горить, то виконувати цю операцію не треба.

4. Запрограмувати перший канал контролера відповідно до схеми на рис. 12.7.

4.1. На пульту ПН-1 за допомогою клавіші “→” вибрати процедуру “приб.” і, користуючись табл. 12.3, виконати усі запропоновані операції.

4.2. Перейти у режим “робота” і переконатися, що відмови та помилки відсутні, а інтерфейс вимкнений (на ПЛ-1 індикатори “ош” і “отказ” не горять, а “откл.интф” горить); повернутися у режим “програм”.

4.3. На пульту ПН-1 встановити процедуру “сист” і виконати операції, відповідно до табл.12.4.

4.4. На пульту ПН-1 встановити процедуру “алг”(алгоритми) і по-черзі, починаючи з першого алгоблоку, ввести алгоритми і їх реквізити, табл. 12.5.

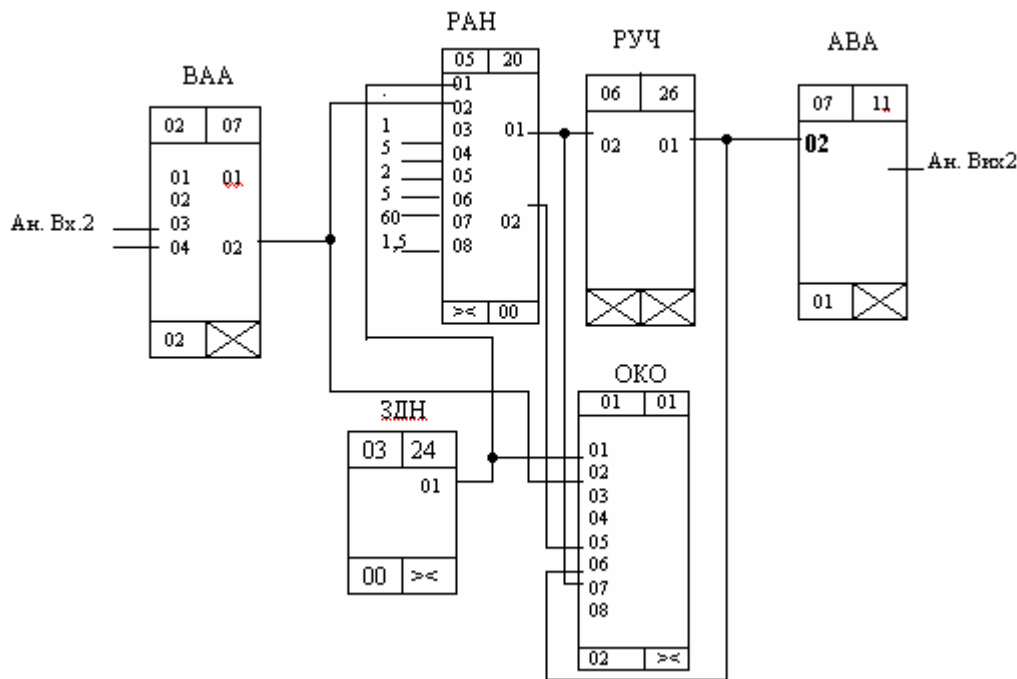


Рисунок 12.7 – Схема конфігурації Реміконта Р-130

4.5. На пульту ПН-1 установити процедуру “конф”(конфігурування) і для входів усіх алгоблоків, що повинні бути зв’язаними, установити конфігурацію, табл.12.6.

4.6. На пульту ПН-1 установити процедуру “настр”(настройка) і для входів усіх алгоблоків, де повинні задаватися коефіцієнти, установити ці параметри, табл.12.7.

4.7. На пульту ПН-1 установити процедуру “н.усл” (початкові умови). Оскільки в лабораторній роботі початкові умови алгоритмів не змінюються, цю процедуру можна не виконувати.

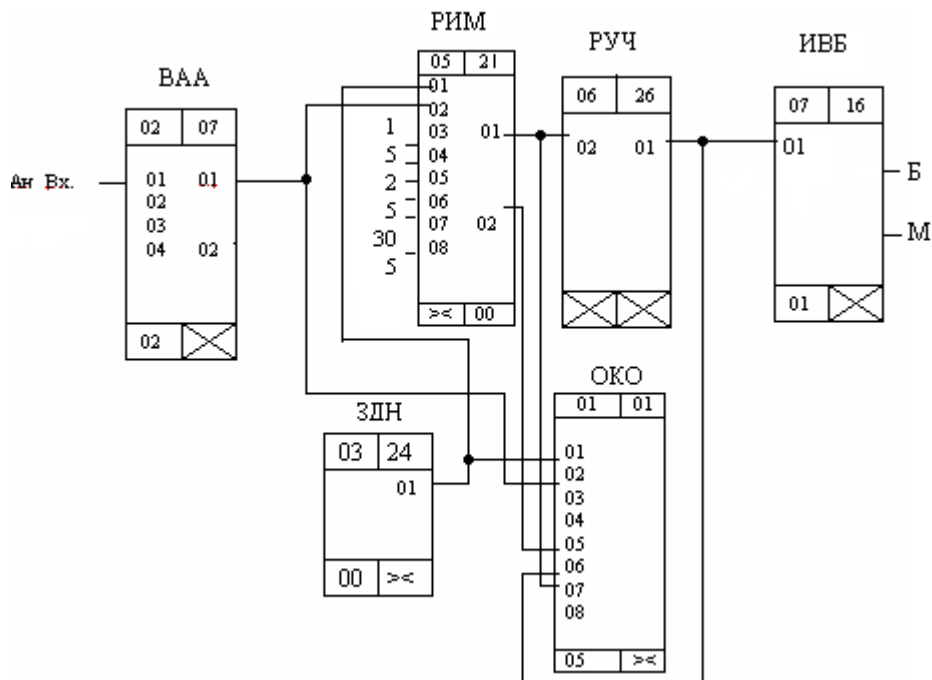
4.8. Перевести контролер у режим “робота” одночасним натисканням на клавіші “↓” і “←” і переконавшись, що немає помилок і відмов. Установити процедуру “приб” (приладові параметри) і визначити час, що витрачається; цей час мінімум на 10-20% має бути меншим за установлений час циклу;

4.9. У режимі “робота” вимкнути пульт настройки ПН-1. За допомогою відповідних клавіш лицевої панелі в цифровому індикаторі «Задание» установити 30% і виконати дослідження роботи системи автоматичного регулювання температури в печі, керуючись цифровим індикатором блока БК-1, і вторинним приладом РП160, який показує температуру печі. Змінюючи коефіцієнти настройки алгоблока РАН дослідити якість регулювання.

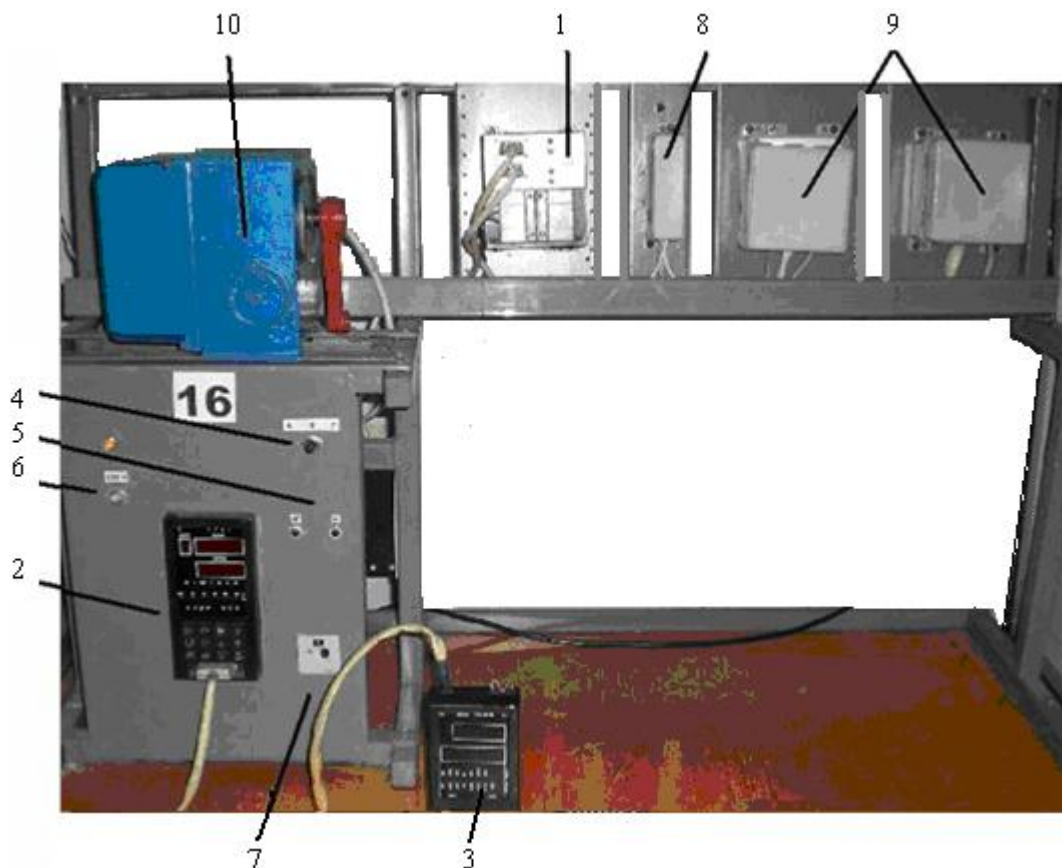
4.10. Опрацювати результати дослідження, зробити висновки і скласти звіт.

### ***Завдання для самостійної роботи***

Запрограмувати і дослідити роботу Р-130, який керує виконавчим механізмом за ПІД-імпульсним законом, відповідно до наведеної схеми конфігурації:



Лабораторна установка для програмування контролера і дослідження його роботи має вигляд:



1 - блок живлення БП-1; 2 - блок контролера БК-1; 3 - пульт настройки ПН-1; 4 - перемикач режимів роботи А/Р; 5 - кнопки ручного управління РУЧ; 6 - тумблер джерела живлення зі світловим індикатором; 7 - задавач вхідного сигналу РЗД; 8 - клемно-блоковий з'єднувач КБС-1; 9 - клемно-блоковий з'єднувач КБС-2; 10 - виконавчий механізм МЕО.

Комплектність контролера – 26.

### ***Зміст звіту***

Звіт має містити назву роботи, опис лабораторної установки, короткі відомості про контролер, програму керування температурою печі або результати самостійної роботи і висновки про виконану роботу.

### ***Контрольні запитання***

1. Яка фізична структура Р-130?
2. Що таке віртуальна структура Р-130 і з чого вона складається?
  1. Якими реквізитами характеризується алгоблок?
4. Яке призначення алгоритму ОКО?
5. Якою мовою програмується Р-130 і в чому особливості програмування ?
7. Як пульт настройки перевести у робочий режим, а як у режим програмування?
8. Як здійснюється конфігурування алгоблоків?
9. Як визначається комплектність контролера і в якій процедурі вона установлюється?
10. Яке призначення мають клемно-блокові з'єднувачі?

### **12.3 Лабораторна робота №3**

#### **Програмування та дослідження роботи контролера МІК-51**

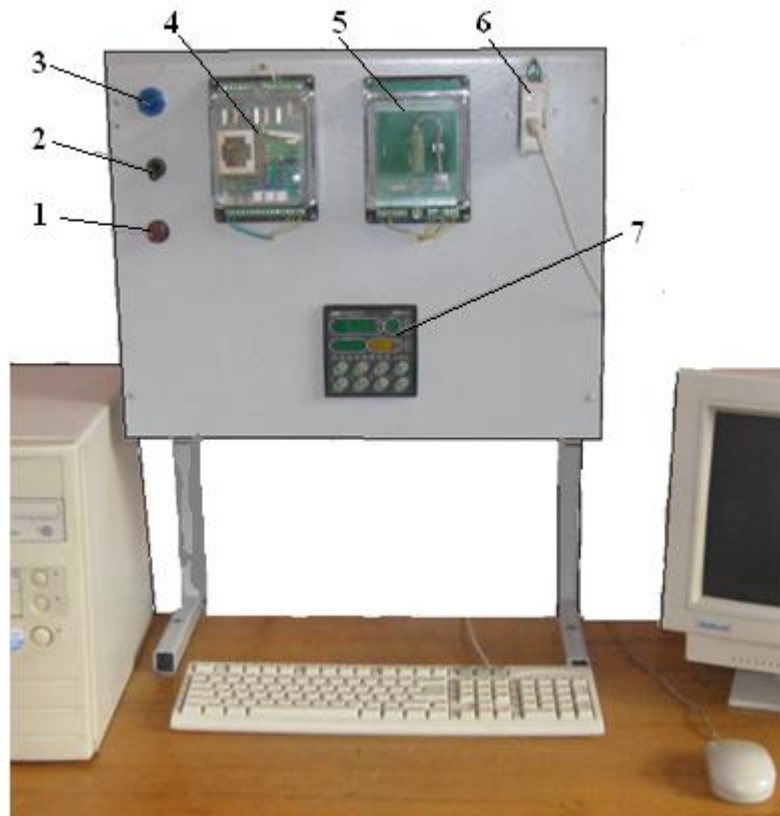
**Мета:** ознайомитися з будовою МІК-51, придбати навички програмування і роботи з контролером.

Лабораторна установка, рис.12.8, призначена для програмування та дослідження роботи контролера МІК-51. У її складі контролер МІК-51 (7), який, відповідно до програми користувача за ПД-аналоговим законом регулювання формує керувальний вплив на нагрівальну піч (5). Підсилення вихідного сигналу контролера здійснює блок підсилювання сигналу БУС (4). Програмування контролера може здійснюватися з лицевої панелі або за допомогою ЕОМ, використовуючи інструментальну систему АЛЬФА. Зв'язок комп'ютера з контролером відбувається через інтерфейсний канал, в якому перетворення RS232/RS485 здійснює блок БПІ-52 (6) [63].

### ***Контролер МІК-51***

Контролер МІК-51 являє собою комплекс технічних засобів, у складі якого - центральний мікропроцесорний блок і клемно-блоковий з'єднувач. Центральний блок перетворює аналогову і дискретну інформацію в цифрову форму, обробляє її і «видає» керуючі впливи. Формування імпульсних сигналів на виході контролера виконується програмно.

Клемно-блоковий з'єднувач призначений для зручної комутації контролера до інших технічних засобів.



*1 – запобіжник; 2 – вмикач напруги; 3 – індикатор наявності напруги; 4 – блок підсилювання вихідного сигналу контролера БУС; 5 – нагрівальна піч; 6 – блок перетворення інтерфейсу БПІ-52; 7 – контролер МІК-51*

Рисунок 12.8 – Лабораторна установка

МІК-51 містить засоби оперативного керування, які розташовані на лицевій панелі контролера. До 32 контролерів МІК-51 можуть поєднуватися в локальну керуючу мережу шинної конфігурації.

Віртуальна структура (архітектура) контролера МІК-51 характеризує його функціональні можливості.

Частина архітектурних елементів реалізована апаратно, частина - програмно. До складу архітектури контролера входять:

- Апаратура вводу-виводу інформації.
- Апаратура оперативного керування й налаштування.
- Апаратура інтерфейсного каналу.
- Функціональні блоки.
- Бібліотека функціональних блоків.

Програмування контролера виконується FBD-мовою за допомогою клавіш передньої панелі або по інтерфейсу за допомогою спеціального програмного забезпечення - візуального редактора FBD-Програм АЛЬФА. Бібліотека має більше 50 функціональних блоків.

В контролері є вісім процедур програмування, кожна з яких має свій рівень, який виводиться на дисплеї «№» (КАНАЛ).

- Перший рівень* - настроювання кількості блоків.
- Другий рівень* - програмування функціональних блоків.
- Третій рівень* - програмування зв'язків функціонального блока.
- Четвертий рівень* - настроювання параметрів функціональних блоків.
- П'ятий рівень* - тестування і калібрування входів-виходів. (В роботі не виконується).
- Шостий рівень* - системні параметри контролера.
- Сьомий рівень* - контроль відмов.
- Восьмий рівень* - збереження програми користувача і настройок в енергонезалежній пам'яті.

Контролер МІК-51 обладнаний активною чотирирозрядною цифровою індикацією. Окрім того лицева панель контролера містить необхідну кількість клавiш обслуговування і сигналізаційних світлодіодних індикаторів для різних статусних режимів і сигналів.

Зміна значень виведених на різні дисплеї (цифрові індикатори) представлено на рис.12.9:

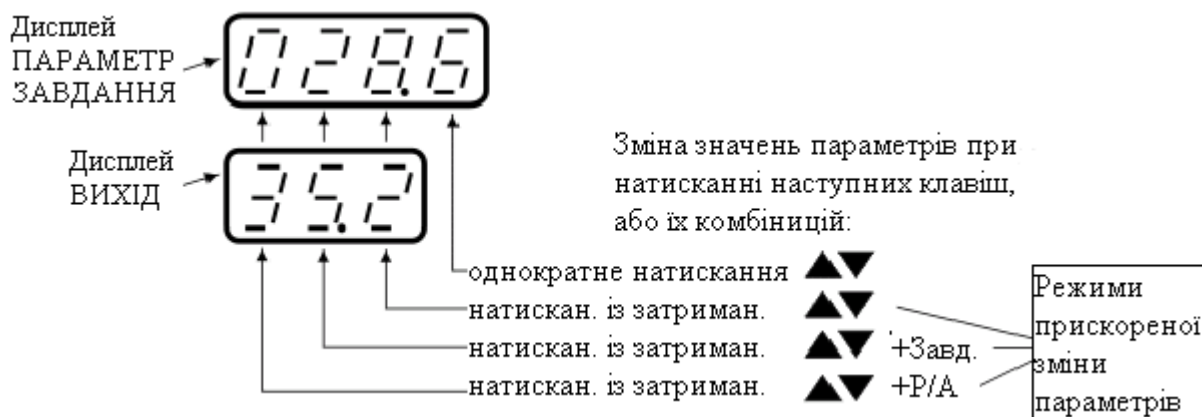


Рисунок 12.9 – Зміна значень на різних дисплеях

Оперативне керування контуром регулювання здійснюється у режимі РОБОТА для обраної панелі індикації. Індикатор РЕГ- світиться.

МІК-51 має три режими керування – два автоматичних (локальний -ЛУ, каскадний -КУ) і ручний - РУ.

**Для переходу в ручний режим** керування необхідно на лицевій панелі регулятора натиснути клавiшу Р/А.

- Індикатор РУ на лицевій панелі починає мигати.
- Для фіксації вибраного режиму треба натиснути із затриманням (приблизно 3-4 секунди) клавiшу [↻], індикатор РУ почне світитися постійно.
- У ручному режимі вихідний сигнал контролера змінюється за допомогою клавiш [▲] і [▼].

**Для переходу в автоматичний локальний** режим керування необхідно на лицевій панелі двічі натиснути клавiшу Р/А і коли індикатор ЛУ почне



мигати – натиснути клавішу [↻] (приблизно 3-4 секунди). Після фіксації вибраного режиму індикатор РУ згасне, а засвітиться індикатор ЛУ.

**Для переходу в автоматичний каскадний** режим керування необхідно тричі натиснути клавішу [P/A] на лицевій панелі регулятора і, коли індикатор КУ почне мигати, зафіксувати вибраний режим, натиснувши клавішу [↻] (приблизно 3-4 секунди). Індикатор ЛУ згасне і засвітиться індикатор КУ.

*Зміна заданої уставки регуляторів*

- Стосовно функціонального блока регулювання, задана уставка регулятора може бути внутрішньою або зовнішньою.

- Внутрішня задана уставка змінюється з лицеві панелі приладу.

- Зовнішня задана уставка регулятора може задаватися із зовні або формуватися іншим функціональним блоком. Змінити її значення з лицеві панелі регулятора неможливо.

- Для зміни уставки, МК-51 має бути переведений у режим РОБОТА, і вибрана відповідна панель індикації (індикатор РЕГ світиться). На дисплеї ПАРАМЕТР – показується значення вимірюваної величини, а на дисплеї ЗАВДАННЯ - значення уставки.

- На лицевій панелі натиснути клавішу ЗАВД, почне мигати дисплей ЗАВДАННЯ.

- За допомогою клавіш [▲] і [▼] на дисплеї ЗАВДАННЯ установити необхідне значення внутрішньої заданої уставки.

- Натиснути клавішу [↻] (приблизно 3-4 секунди) - регулятор перейде в режим керування з новим значенням внутрішньої заданої уставки.

*Зміна значення керуючого впливу*

- Контролер перевести в ручний режим керування. Індикатор РУ - світиться.

- За допомогою клавіш [▲] і [▼] управляти виходом регулятора.

- При зміні керуючого впливу після першого натискання кожної із клавіш ▲,▼ починає мигати дисплей ВИХІД при аналоговому вихідному сигналі, або світлодіодні індикатори ▲ чи ▼ - при дискретному.

### ***Виконання лабораторної роботи***

Виконати програмування контролера з лицеві панелі на прикладі ПІД-аналогового алгоритму і дослідити його роботу у системі автоматичного регулювання температури печі.

Для програмування контролера FBD – мовою необхідно використати наступні функціональні блоки:

- AIN(5) - аналоговий ввід,
- PID(60) - регулятор аналоговий,
- AOT(8) - аналоговий вивід.

Щоб виконати конфігурування і настроювання функціональних блоків необхідно розробити відповідну таблицю, табл.12.9.

Таблиця 12.9

## Програмування функціональних блоків

Алгоритми					Конфігурація					Настроювання		
№ блока	Шифр	№ алгоритму	Базова адреса	Модифікатор	№ блока	№ входу	№ блока	№ виходу	Ім'я входу	№ параметра	Ім'я параметра	Значення
1	AIN	05	000	1						1	nINP	1
										2	nTYPE	0
										3	tFLT	0:00:00.5
										4	BIAS	0.000
										5	OUT_H	100.0
										6	OUT_L	0.00
										7	MAX	15.0
										8	MIN	5.000
										9	HYS	0.000
										10	IN_H	0.000
										11	IN_L	0.000
										12	PV	0.000
2	PID	60	012	1	2	1	1	4	V	1	nDISP	1
										2	nDIR	0
										3	nSTR	0
										4	nTYPE	0
										5	SP_Q	1
										6	nCONV	0
										7	dTRK	0
										8	nMODE	1
										9	TI	0:03:00.0
										10	TD	0:00:00.0
										11	KP	5.000
										12	SP	20.000
										13	SP_EXTEN	0.000
										14	OUT	100.000
										15	PV_RANE	100.000
										16	SP_H	100.000
										17	SP_L	0.000
										18	SP_RATE	0.000
										19	OUT_H	100.000
										20	OUT_L	0.000
										21	TRK_VAL	0.000
										22	PV_MAX	80.000
										23	PV_MIN	30.000
										24	PV_HYS	0.000
										25	FF_VAL	0.000
										26	FF_GAIN	0.000
										27	FF_H	0.000
										28	FF_L	0.000
										29	COR_VAL	0.000

										30	COR_GAIN	0.000
										31	COR_H	0.000
										32	COR_L	0.000
										33	FB	0.000
3	AOT	08	045	1	3	1	2	7	NP	1	nOUT	1
										2	--	0.000
										3	--	0.000

### **Створення програмного коду**

1. Подати напругу живлення на контролер МІК-51.
2. Перевести контролер у режим «робота» (за допомогою комбінації клавіш Меню +№ Кн ↑) і клавішею “№Кн” вибрати перший контур програмування.
3. Перейти у режим ПРОГРАМУВАННЯ тривалим, більше 3-х секунд, одночасним натисканням клавіш “Меню + №кн ↓”. На дисплеї ПАРАМЕТР з’явиться PASS, а на дисплеї ЗАВДАННЯ - миготливе 0000.
4. За допомогою клавіші ▲ ввести пароль: 0002 і натиснути клавішу ”Ввід”.  
Якщо пароль уведений правильно, то прилад перейде в режим ПРОГРАМУВАННЯ і індикатор ПР засвітиться, якщо ні – в режим РОБОТА.
5. На лицевій панелі контролера натиснути кнопку ”Ввід” і клавішами ▲,▼ обнулити 1-ий і 2-ий розряди дисплея ПАРАМЕТР. Знову натиснути кнопку ”Ввід” і обнулити 3-ий і 4-ий розряди дисплея ПАРАМЕТР. Натиснути кнопку ”Ввід” і обнулити розряди решти цифрових дисплеїв (після кожного обнуління результат підтверджується натисканням клавіші ”Ввід”).
6. На дисплеї “№” установити перший рівень програмування – «Настроювання кількості блоків» і ввести кількість функціональних блоків, які використовуються програмою - 03, кількість панелей регулювання - 01, кількість програмних задавачів - 0000, кількість панелей користувача - 000.
7. Установити другий рівень - «Програмування функціональних блоків» і відповідно до табл.12.9 ввести в ОЗП функціональний блок АІН(05), указавши номер функціонального блока (01) в програмі користувача, тип блока (05), базову адресу параметрів(0000) і модифікатор кількості входів і/або параметрів (01), як це показано у табл.12.10.

Таблиця 12.10 Програмування функціонального блока АІН(05)

Дисплей «№» КАНАЛ, Номер рівня	Дисплей ПАРАМЕТР		Дисплей ЗАВДАННЯ	Дисплей ВИХІД
	1-й, 2-й розряд	3-й, 4-й розряд		
2	01	05	0000	01

Аналогічно виконати програмування функціональних блоків РІД(60) і АОТ(08), користуючись табличним записом програми, табл.12.9.

8. Установити третій рівень – «програмування сполучень» (конфігурування). І для всіх входів всіх блоків, які мають бути сполучені з

виходами інших блоків, установити необхідну конфігурацію. Програмування сполучень для PID-блоку приведено у табл.12.11.

Таблиця 12.11 Програмування зв'язків для блока PID

Дисплей «№» КАНАЛ, Номер рівня	Дисплей ПАРАМЕТР		Дисплей ЗАВДАННЯ	Дисплей ВИХІД
	1-й,2-й розряд	3-й,4-й розряд		
3	02	01	01	04

Аналогічно, користуючись табличним записом програми, табл.12.9, конфігуруються сполучення для блока АОТ.

9. Установити четвертий рівень - «настроювання параметрів» і занести в ОЗП настроювальні параметри алгоблоків програми, користуючись описом функціональних блоків у керівництві з експлуатації, а також табл.12.9.

У функціональному блоці АІН обов'язково вказується номер фізичного аналогового входу, тип датчика, діапазон вимірювання датчика.

У функціональному блоці PID обов'язково вказується № дисплея індикації в панелях регулятора, а також такі параметри як структура, напрям дії, режим роботи, коефіцієнти регулятора, діапазон зміни регульованого параметра, обмеження завдання, обмеження вихідного сигналу регулятора.

У функціональному блоці АОТ обов'язково вказується номер фізичного аналогового виходу.

Приклад програмування параметрів алгоблока АІН наведений у табл.12.12.

Таблиця 12.12 Настроювальні параметри алгоблока АІН

Дисплей «№» КАНАЛ, Номер рівня	Дисплей ПАРАМЕТР		Дисплей ЗАВДАННЯ	Дисплей ВИХІД
	1-й,2-й розряд	3-й, 4-й розряд		
4	01	01	1.000	000
		02	0000	000
		03	000.5	000
		04	0000	000
		05	100.0	000
		06	0000	000
		07	15.0	000
		08	0000	000
		09	0000	000
		10	0000	000
		11	0000	000
		12	0000	000


Оскільки вхідний сигнал контролера формується типовим термометром опору ТСМ 50, а в якості вихідного використовується стандартний струмовий сигнал 0-5мА, то немає необхідності робити тестування і калібрування входів-виходів. Немає потреби також встановлювати системні параметри, оскільки

контролер не знаходиться у мережі. Тому п'ятий, шостий і сьомий рівні програмування можна не виконувати.

10. Встановити восьмий рівень – “збереження програми користувача і настройок” і вказати задану кількість блоків і кількість задіяних блоків, а також запис в енергонезалежну пам'ять, відповідно до табл.12.13.

Таблиця 12.13 Збереження програми користувача

Дисплей«№» КАНАЛ, Номер рівня	Дисплей ПАРАМЕТР		Дисплей ЗАВДАННЯ	Дисплей ВИХІД
	1-й,2-й розряд	3-й, 4-й розряд		
8	3	3	0001	000

11. Перевести контролер у режим РОБОТА (за допомогою комбінації клавіш  Меню +[№ Кн ↑]), переконавшись, що немає помилок і відмов (індикатор ОШ не світиться) і перейти на панель “Рег”. На дисплеї ПАРАМЕТР відобразиться поточне значення регульованого параметра, на дисплеї “№” - номер регулятора –1– на дисплеї ЗАВДАННЯ - значення заданої уставки, на дисплеї ВИХІД – значення вихідного сигналу регулятора. При цьому зміна вихідного сигналу відбувається відповідно до форми динамічної характеристики ПІД-закону.

12. Дослідити роботу контролера і системи автоматичного регулювання температури печі за показаннями цифрових індикаторів «Вихід» і «Параметр», змінюючи завдання у бік збільшення та зменшення його величини.


13. Перейти у ручний режим роботи, та з лицевої панелі контролера здійснити керування виходом регулятора за допомогою клавіш “Знач▲” і “Знач▼”.

14. Виконати програмування контролера за допомогою інструментальної системи «Альфа».


### ***Програмування за допомогою інструментальної системи «Альфа»***

1. Перевести контролер у режим «Робота», натиснути кнопку ”Ввід” і клавішами ▲,▼ обнулити 1-ий і 2-ий розряди дисплея ПАРАМЕТР. Знову натиснути кнопку ”Ввід” і обнулити 3-ий і 4-ий розряди дисплея ПАРАМЕТР. Натиснути кнопку ”Ввід” і обнулити розряди решти цифрових дисплеїв (після кожного обнуління результат підтверджується натисненням клавіші ”Ввід”).

2. Запустити програму «Альфа» використовуючи на ЕОМ меню «Пуск -> Все программы» і вибрати «Альфа».

3. Для створення проекту, у головному меню екрана вибрати «Файл» -> «Новый» і в панелі інструментів екрану програми «Альфа» натиснути кнопку «Новый» .

4. При необхідності відкриття раніше створеного проекту:

- натиснути кнопку «Відкрити» або вибрати команду «Відкрити»  в меню «Файл»;

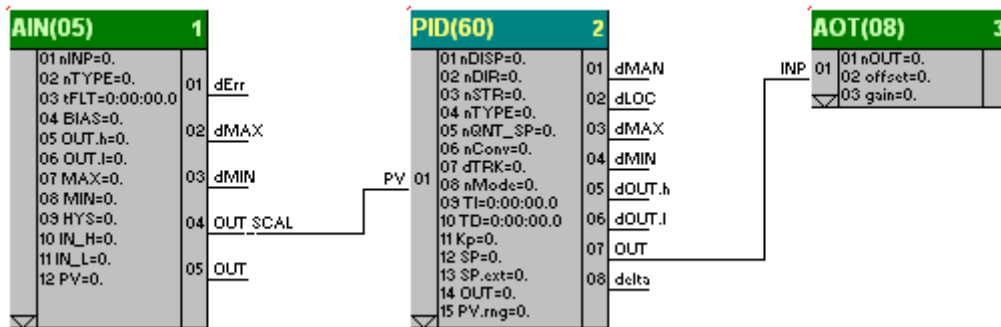
- у полі «Тип файлів» вибрати необхідний формат файлу \*.fbd;
  - щоб відкрити проект, збережений в іншій теці, знайти і відкрити потрібну теку;
  - клацнути по проекту, який потрібно відкрити в ALFA;
  - натиснути кнопку «Открыть».
5. Для збереження проекту:



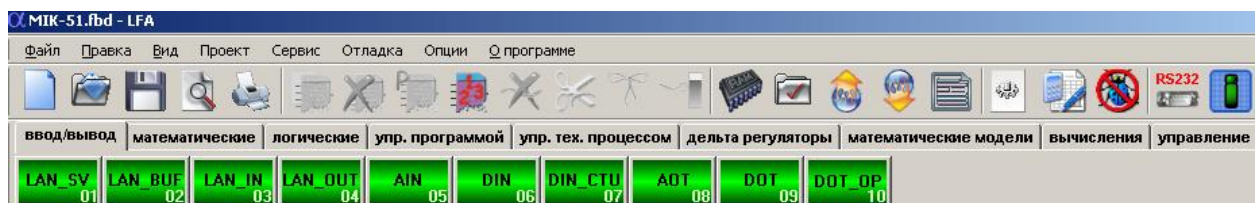
- натиснути кнопку «Зберегти» на панелі інструментів;
- Щоб зберегти файл в іншій теці, знайти і відкрити потрібну теку;
- Ввести ім'я документа в полі Ім'я файлу і натиснути кнопку «Сохранить».

### Створення програмного коду

Програмний код алгоритму, за яким здійснюється дослідження системи автоматичного регулювання температури печі, має наступний вигляд:



Вибір необхідних функціональних блоків здійснюється за допомогою панелі інструментів з відповідного розділу бібліотеки, табл. 12.14.



Щоб вибрати функціональний блок, необхідно увійти у відповідний розділ бібліотеки і клацнути лівою клавшею миші на потрібному блоці. Для установлення вибраного блока на робочому полі потрібно повторно клацнути лівою клавшею у вільній зоні екрану.

Сполучення блоків здійснюється за допомогою миші:

- установити курсор на відповідний вхід блока і після появи на вході кружка - натиснути лівою клавшею миші;
- пересунути курсор з натиснутою лівою клавшею миші на відповідний вихід потрібного блока (джерела сигналу) і після появи кружка відпустити клавшею;

- за необхідністю інверсії вхідного сигналу, встановити курсор миші на вхід блока і натиснути на клавіатурі ЕОМ клавішу CTRL та ліву кнопку миші.

Таблиця 12.14 Розділи бібліотеки функціональних блоків


Розділ	Найменування розділу	Перелік функціональних блоків
1	Функціональні блоки вводу-виводу	Інтерфейсний ввід-вивід, аналоговий ввід-вивід, дискретний ввід-вивід, імпульсний ввід
2	Математичні функціональні блоки	Множення, підсумовування з масштабуванням, розподіл, корінь квадратний і др.
3	Логічні функціональні блоки	Логічне І, логічне АБО, багатовходовое АБО, тригер, регістр, виділення фронту
4	Функціональні блоки керування програмою	Мінімум, максимум, екстремум, обмеження, перемикач по номеру, компаратор, таймер, лічильник, мультівібратор і др.
5	Функціональні блоки керування технологічним процесом	Фільтр, масштабування, кусково-лінійна функція, уставка аналогова, уставка часу, програмний задавач, регулятор аналоговий, регулятор каскадний, регулятор імпульсний, панель користувача
6	Функціональні блоки дельта-регуляторів	Регулятор аналоговий з розширеними функціями, дельта-регулятор

Для настроювання параметрів функціональних блоків:


- установити курсор на відповідний блок і натиснути ліву клавішу миші, при цьому блок підсвічується;
- натиснути праву клавішу миші на виділеному блоці і з меню, що з'явиться, вибрати пункт "Параметры блока":
- у діалоговому вікні редагування параметрів блока виконати відповідні настроювання, користуючись табличною формою програми, табл.12.9.


Нумерація блоків виконується в порядку їх установлення.

При необхідності можна змінити нумерацію:


- автоматично, за допомогою кнопки  (Пронумерувати блоки автоматично) на панелі інструментів. При цьому виконується послідовна нумерація блоків у порядку їх установлення;
- вручну при одночасному натисканні кнопок CTRL+D, установлення початкового і подальших номерів блоків у порядку зростання.

Розподіл параметрів блока в регістровій області пам'яті здійснюється:

- автоматично за допомогою кнопки  (Автоматичний розподіл пам'яті) на панелі інструментів;
- вручну із вказівкою базової адреси блока в регістровій області пам'яті.

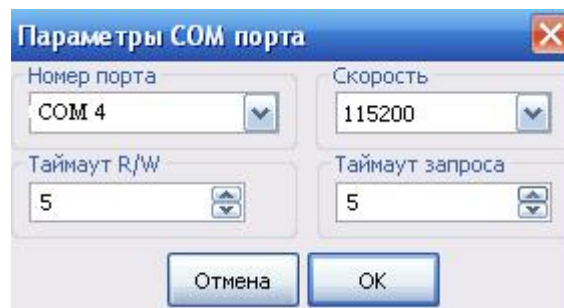
Перевірка програми здійснюється за допомогою кнопки  (Перевірити програму), що на панелі інструментів. При цьому виконується перевірка програми користувача на відповідність існуючим правилам.


### *Налагодження*


Налагодження у реальному часі запускається кнопкою  (Налагодження) на панелі інструментів і забезпечує читання даних з контролера з наступною візуалізацією станів всіх виходів блоків.

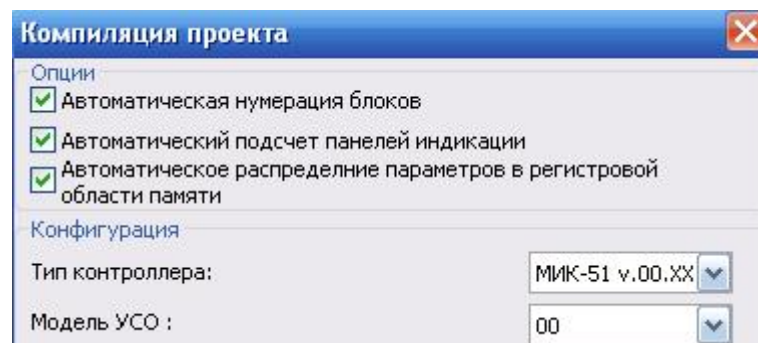
### *Запис і читання програми з контролера*

В меню «Опции» або за допомогою кнопки  на панелі інструментів визначити параметри СОМ-порту:



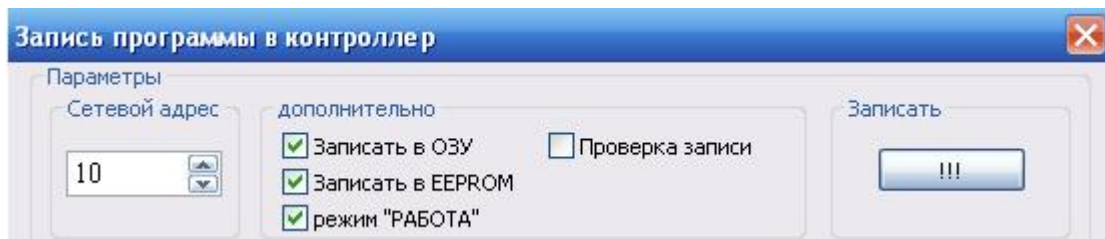
*Запис, читання програми* здійснюється за відповідною командою в меню «Действия» або, користуючись кнопками  на панелі інструментів.

Після натискання кнопки  з'явиться діалогове вікно «Компиляция проекта», в якому необхідно зробити уставки, як у наведену нижче фрагменті та натиснути ОК.



Повторно натиснути кнопку  (Записати програму в контролер), з'явиться діалогове вікно Запису програми у контролер:





Активізувати необхідні пункти запису і виконання програми та, указавши адресу пристрою, натиснути кнопку «Записать». Після завершення процесу запису, на дисплеї контролера висвітяться символи «ЗП». Коли це повідомлення зникне, контролер почне регулювання температури.

За показаннями дисплеїв ПАРАМЕТР і ВИХІД відслідковувати роботу системи автоматичного регулювання.

### ***Завдання для самостійної роботи***

Запрограмувати і дослідити роботу контролера, який керує виконавчим механізмом за ПІД-імпульсним законом, відповідно до алгоритму наведеному у розділі 4.4.1 даного посібника. Лабораторна установка для програмування контролера і дослідження його роботи має вигляд:



*1 - автомат вимикання напруги; 2 - контролер МІК-51; 3 - блок перетворювання інтерфейсів БПІ-52; 4 - клемно-блоковий з'єднувач КБЗ-29Р-01; 5- трубопровід із заслінкою; 6 - дистанційний вказівник положення ДУП; 7- виконавчий механізм МЕО.*

### ***Зміст звіту***

Звіт має містити назву роботи, опис лабораторної установки, короткі відомості про контролер, програму керування температурою печі або результати самостійної роботи та висновки про виконану роботу.

### **Контрольні питання**

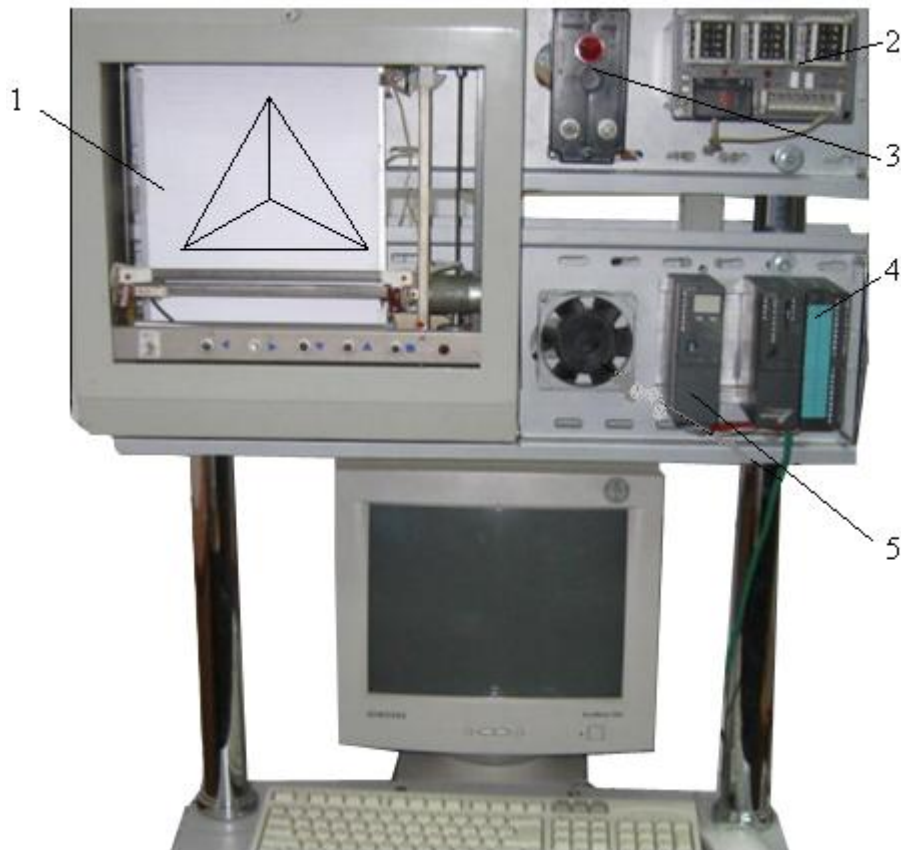
1. Що являє собою контролер МІК-51?
2. З чого складається лабораторна установка?
3. Яка архітектура контролера?
4. Які існують режими роботи і для чого вони використовуються?
5. В яких випадках використовується ПІД-аналоговий алгоритм, а в яких ПІД-імпульсний?
6. Як програмується контролер МІК-51 з лицевої панелі?
7. Як здійснюється оперативне керування?
8. Для чого призначена інструментальна система «Альфа»?
9. Чим характеризується бібліотека функціональних блоків?
10. Як програмується контролер МІК-51 за допомогою системи «Альфа»?

### **12.4 Лабораторна робота №4**

Програмування та дослідження роботи контролера VIPA Speed 7 300

**Мета:** ознайомитися з будовою, придбати навички програмування і роботи з контролером.

Лабораторна установка, рис.12.10, призначена для оволодіння методикою програмування і роботи з контролером VIPA Speed 7 300.



1- графобудовник; 2-джерело живлення контролера; 3-індикатор наявності напруги; 4-контролер VIPA7-Speed-300; 5-модуль живлення.

Рисунок 12.10 – Лабораторна установка

На лабораторній установці контролер VIPA Speed 7 300 забезпечує роботу графобудовника шляхом вмикання або вимикання двох двигунів, які рухають каретку у горизонтальному та вертикальному напрямках. Алгоритм роботи графобудовника визначається програмою користувача, яка створюється і завантажується у контролер, протягом виконання лабораторної роботи.

### ***Контролер VIPA Speed 7- 300***

Фірма VIPA випускає 5 типів контролерів — System 100V, System 200V, System 300V, System 300S, System 500V, які відрізняються кількістю каналів, функціональними та комунікаційними можливостями.

Однією з головних особливостей контролерів серії System 300 є універсальність. За допомогою модулів System 300 є можливість створювати системи для збирання даних та керування як з централізованою, так і розподіленою архітектурою.

Серія System 300 побудована за модульним принципом. Усі модулі вводу — виводу є універсальними, тобто їх можна застосовувати разом з будь-яким CPU цієї серії. Водночас є можливість вибору процесорного модуля з оптимальною продуктивністю для розв'язання конкретного завдання.

Контролери серії System 300 найбільш близькі до контролерів Simatic S7-300. Модулі цієї системи функціонально тотожні однотипним модулям S7-300 і можуть використовуватися з ними в одній стійці (номери для замовлення взаємозамінних модулів збігаються за символами). Модулі серії System 300 дешевші, тому споживач, використовуючи як процесорний модуль CPU Simatic, доукомплектує систему модулями розширення від VIPA.

Ще однією особливістю системи, побудованої на базі System 300 від VIPA, є те, що на одній стійці може бути встановлено до 32 модулів вводу-виводу. Це стає в пригоді, коли в системі більше 8 модулів і немає необхідності використовувати інтерфейсні модулі IM360, IM361 або IM365, як це відбувається в контролері Simatic S7-300.

Серед контролерів System 300 від VIPA особливої уваги заслуговує остання розробка фірми - серія високошвидкісних CPU, побудованих на базі спеціалізованого процесора PLC7000, заснованого на технології SPEED7.

Застосування цієї технології дало змогу суттєво збільшити швидкодію контролера. Як результат - операція з бітом або словом виконується лише за 14 нс, а з числом з плаваючою комою - за 84 нс. Час циклу виконання типового завдання, що складається з 100 тисяч інструкцій, може бути трохи більшим за 2 мс.

Як спеціальні модулі в контролері використовують дискретні вхідні-вихідні, аналогові вхідні, Profibus-Master, Interbus-Master, а також CANopen-Master.

RAM+FLASH-пам'ять для зберігання програми обсягом від 256 Кбайт до 2 Мбайт інтегрована в процесорний модуль.

За допомогою флеш-пам'яті у форматі Multimedia Card (MMC) обсяг робочої пам'яті збільшується з базових 512 Кбайт до 2 Мбайт в CPU 314S, з 1 до 2 Мбайт в CPU315S і з 2 до 8 Мбайт в CPU317S.

Програмування контролера може здійснюватися як за допомогою програмного забезпечення WinPLC7 від VIPA, так і за допомогою пакета STEP 7 від Siemens.

На всіх процесорних модулях основним є інтерфейс MP2I(MPI+PPI=MP2I) зі швидкістю обміну 187,5 Кбод. Кабель зв'язку від MPA об'єднує MPI з протоколом PC і PPI на одному розніму. Тому при програмуванні контролерів немає необхідності використовувати перетворювач інтерфейсу MPI – PPI, достатньо мати "зелений кабель" (green cable), призначений спеціально для контролерів VIPA. Як додатковий інтерфейс на процесорному модулі використовують Profibus Master із швидкістю обміну до 12 Мбод, при цьому допускається в мережі 125 Slave. Крім того, на цьому ж рознімі можна організувати PtP- інтерфейс на основі RS-485. Ці два інтерфейси гальванічно відділені від системної шини. Існує можливість обміну по ASCII, STX/ETX, 3964B, Modbus master або USS master протоколах.

Крім того, на кожному процесорі є ще один інтерфейс - Ethernet (10/100 Мбіт/с), що дає змогу сконфігурувати до 128 з'єднань. На деяких процесорних модулях Speed 7 є два Ethernet порти. Другий реалізований як вбудований комунікаційний процесор Ethernet CP 343, він підтримує S7-сумісні комунікації. Це дуже зручно використовувати, бо при цьому:

- один порт (протоколи PG/OP) є для зв'язку з програматорами і засобами операторського інтерфейсу, конфігурування, програмування, налагодження і діагностики контролера, а також для обміну з SCADA програмою;

- другий порт (за іншими протоколами) - для зв'язку з контролерами — партнерами.

В усіх моделях процесорів є годинники реального часу, а також літєві акумулятори, які забезпечують зберігання даних в незалежному ОЗП і роботу годинника протягом 30 днів[63].

Решта інформації про модулі контролера знаходиться у розділі даного навчального посібника, де надаються відомості про Simatic S7-300.

### ***Виконання лабораторної роботи***

За допомогою програмного пакету WinPLC7 виконати конфігурування та програмування контролера VIPA Speed 7 300 FBD-мовою, який здійснює креслення рівнобедреного трикутника, керуючи рухом каретки графобудовника.

### ***Створення проекту***

Увімкнути в роботу лабораторну установку та здійснити очищення пам'яті ПЛК. Для цього на блоці живлення контролера PS 307 перевести перемикач у положення On, а на контролері VIPA Speed 7 300 перемикач

режимів встановити в положення MRES. Приблизно через 3с світлодіод STOP перестане блимати і почне горіти постійно. Повернути перемикач у положення STOP і менш ніж за 3с переключити у положення MRES. Світлодіод почне блимати. Скидання контролера завершується коли світлодіод STOP почне горіти постійно.

Для створення проекту запустити інструментальну систему програмування WinPLC7, користуючись ярликом на робочому столі. У вікні, що відкриється, вибрати пункт «Create new project», ввести назву нового проекту (наприклад: «Triangle») і натиснути ОК.

Для конфігурування контролера, в панелі інструментів натиснути ліву клавiшу миші на «Hardwareconfigurator» або вибрати цей пункт з меню «PLC». Коли відкриється додаток «Hardware-CFG», з вікна «Station» вибрати цільову платформу контролера - «VIPA SPEED 7» і натиснути кнопку Create. У вікні Station відобразиться таблиця конфігурації контролера, а поряд каталог Catalog усіх контролерів і модулів, які підтримуються цим програмним пакетом.




Перед тим як зробити конфігурування контролера треба зберегти створену робочу станцію. Для цього у меню File/Project вибрати пункт Save active station, а у вікні, що з'явиться, ввести назву станції, наприклад Station1. Натиснути ОК і на всі наступні запитання, що з'являться на екрані, дати позитивну відповідь.

Щоб додати у слот 2 станції центральний процесор, у вікні Catalog послідовно розкрити списки S7-300, CPU Speed7, CPU 312SC і двічі клацнути по модифікації 6ES7 312-5BE03.

Щоб додати модуль живлення у слот 1, треба позначити його лівою клавiшею миші та зі списку PS 300 вікна Catalog вибрати модуль живлення 6ES7307-1BA00-0AA0 і двічі клацнути лівою клавiшею миші.

Оскільки контролер на лабораторному стенді підключений до комп'ютера за допомогою Ethernet-інтерфейсу, в панелі інструментів зі списку, що випадає, вибрати цільовий інтерфейс станції – Target: External TCP/IP direct.

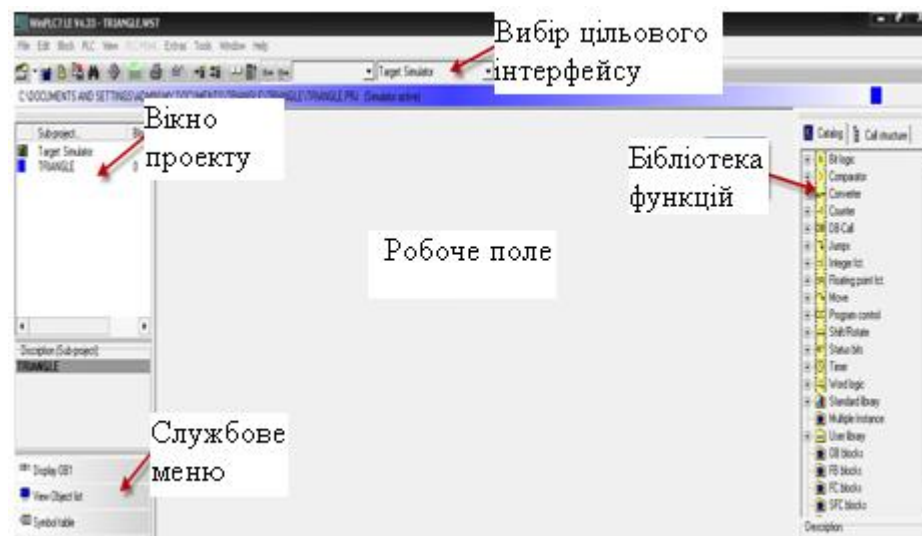
З головного меню в Online вибрати Select network interface of the PC, а у його вікні - активний мережний адаптер комп'ютера, натиснути ОК. Щоб створену конфігурацію відправити до ПЛК, з меню Online вибрати пункт Send configuration to the CPU або натиснути кнопку . З'явиться вікно Send Configuration, в якому для установлення мережних параметрів контролера натиснути кнопку Accessible Nodes. У вікні, що з'явиться, з нижнього списку вибрати активний мережний адаптер Realtek RTL8029(AS) Ethernet Adapter і



натиснути кнопку Determining accessible nodes. Програма здійснить з'єднання з контролером і виведе знайдений контролер у відповідне поле вікна. Позначити лівою клавішею миші знайдений контролер і послідовно натиснути кнопки Confirm і Send Configuration [64].

### *Розроблення програмного коду*

Закрити вікно додатку Hardware configurator і перейти до вікна основної програми:



Програма керування графобудовником має представляти собою функцію, яка визивається з головної програми – об'єкта. Тому треба створити організуючий (головний) блок «ОВ» та блок функції «FC». Для цього – активізувати кнопку WinPLC7 і у вікні проекту лівою клавішею миші позначити ярлик проекту, а у службовому меню активізувати View object list. Службове меню розшириться і у ньому з'явиться пункт New block. Натиснути на ньому ліву клавішу миші, з'явиться вікно, в якому ввести назву блока, наприклад ОВ1. Таким самим чином створити блок функції FC1.

Для заповнення таблиці символічних змінних у службовому меню активізувати пункт Symbol table і надрукувати у відповідній колонці вікна, що з'явиться, вхідні змінні IN0 і IN1 і їх адреси, відповідно I0.0 і I2.0 та надати до них коментар. Закрити вікно символів.

Відкрити вікно функції FC1. Щоб створити два рядка для вхідних змінних необхідно позначити рядок «in», натиснути праву клавішу миші і з меню, що випадає, вибрати Insert after the current line. Надрукувати у таблиці змінних імена IN0 і IN, їх тип INT та коментар.


Клацнути лівою клавішею миші у полі Network 1, у колонці адрес таблиці змінних з'являться їх адреси 0.0 і 2.0.

Вибрати FBD-мову. З каталогу Comparator додати у поле Network 1 оператор CMP ==I, двічі клацнувши по ньому лівою клавішею миші. Через нього подаватиметься сигнал на вмикання двигуна.

Позначити перший вхід суматора та надрукувати змінну #IN0. На другому вході суматора надрукувати вхідну змінну IN1. Позначити вихід суматора та двічі клацнути по таймеру S\_PULSE, що знаходиться у списку каталогу Timer бібліотеці функцій. Присвоїти йому ім'я «T1», на вході «TV» задати час 2 сек («S5T#2s»), а на виході задати безпосередню адресу дискретного виходу контролера Q124.0, який керує роботою двигуна, що зміщує каретку графобудовника по горизонталі.

Щоб додати у робочому полі чотири нових мережі, необхідно клацнути правою клавішею миші, і у вікні Insert new network, що з'явиться, вибрати пункт Insert after network та надрукувати кількість необхідних мереж.

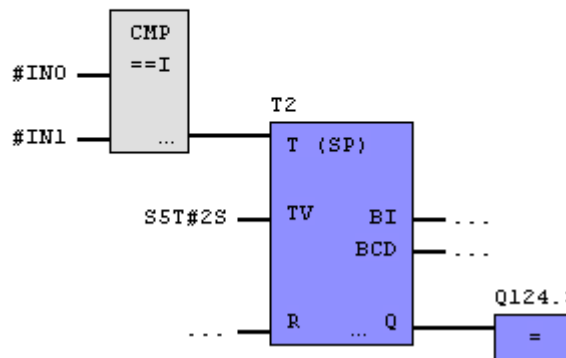
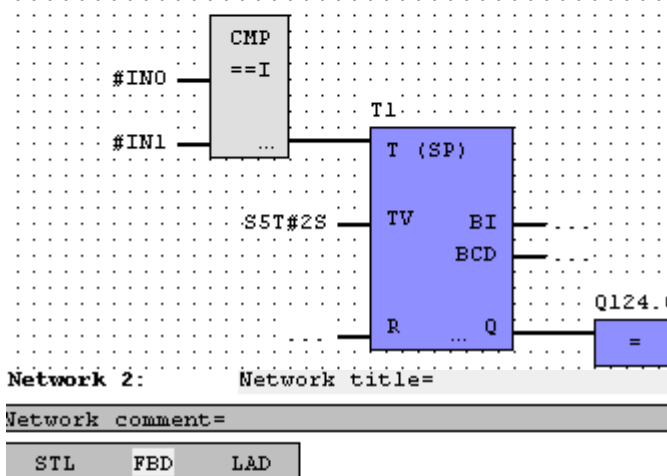
За аналогією з кодом програми, що створений у полі Network 1, сконфігурувати ще чотири поля, користуючись наведеним нижче рисунком.

Увага: сигнали на входах таймерів T3, T4 і T5 інвертуються за допомогою кнопки інверсії , що у панелі інструментів.

:FC1> "" Version:1.0 Author: Name:nb Family:nb				
*	Adress	Declaration	Name	Type
	0.0	in -->	IN0	INT
	2.0	in -->	IN1	INT
		out <--		

Network 1: <Title of Network>  
 Network comment=

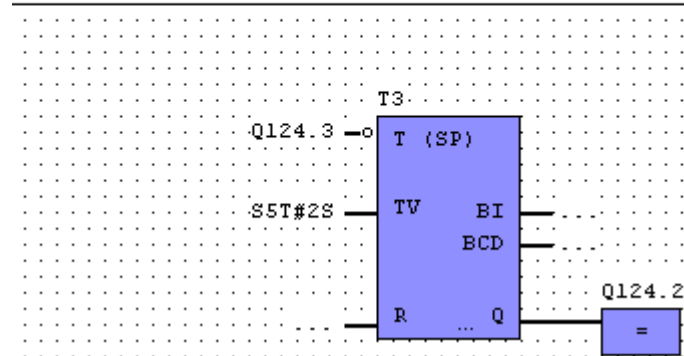
STL    FBD    LAD



Network 3: Network title=

Network comment=

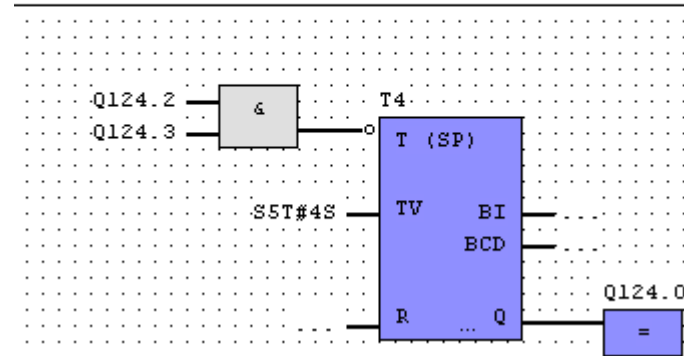
STL FBD LAD



Network 4: Network title=

Network comment=

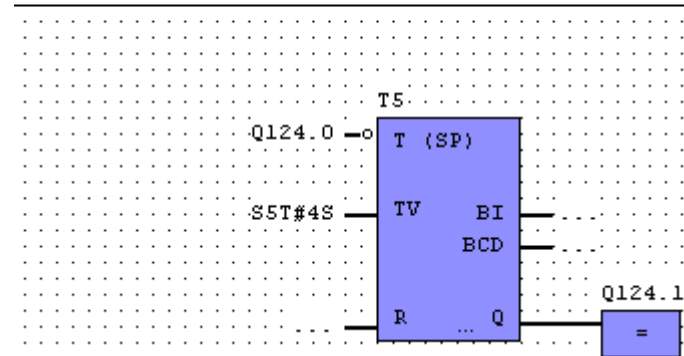
STL FBD LAD



Network 5: Network title=

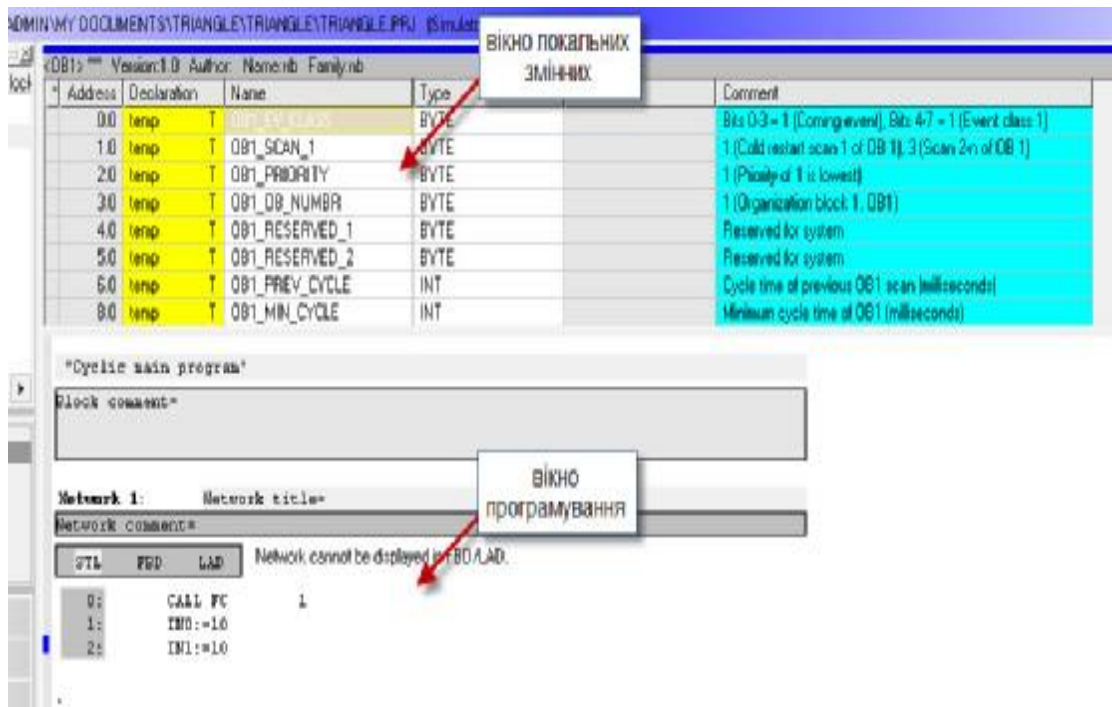
Network comment=

STL FBD LAD






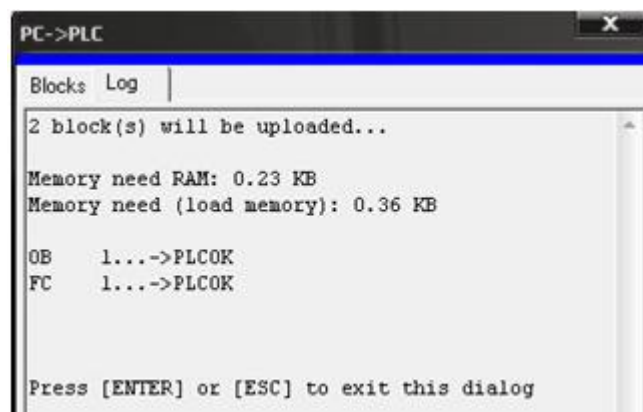
За допомогою кнопки на екрані відкрити блок «OB1» і перейти на STL-мову програмування. Набрати програму виклику функції «FC1» та передачі до неї значень змінних, як показано на рисунку:




У вікні локальних змінних змінити назву усіх змінних на «ТЕМР0»... «ТЕМР9». Закрити блоки OB1 і FC1 та зберегти проект.

### Перевірка коду програми

Зі списку «Вибір цільового інтерфейсу» вибрати «Target: Simulator» та завантажити код програми у стимулятор, натиснувши кнопку  («Send all blocks»). Якщо у коді програми немає помилок, то з'явиться вікно:



Якщо помилкиє, то з'явиться інформація про наявність помилок, які потрібно виправити. Щоб запусити симулятор для перевірки логіки програми, треба натиснути лівою клавішею миші кнопку . У робочому вікні активні


елементи виділяться рамкою та з'явиться інформація про значення, що передані на відповідні входи:

The screenshot displays the 'FC1 of TREUGOLNIK.PRJ (OFFLINE)' window. At the top, there is a table with the following data:

* Address	Declaration	Name	Type
0.0	in -->	INO	INT
2.0	in -->	IN1	INT
	out <--		
	in_out <-->		
	temp T		

Below the table, the 'Network 1' section is visible, showing a ladder logic diagram. The diagram includes a 'CMP ==I' instruction with inputs '#INO' and '#IN1' (both set to '10'). This instruction is connected to a timer 'T (SP)' with a preset value 'T#000.0'. The timer has outputs 'TV', 'BI', 'BCD', 'R', and 'Q'. The 'Q' output is connected to a coil 'Q124.0'.

### ***Завантаження програми до ПЛК***



Для завантаження проекту вимкнути режим моніторингу, натиснувши повторно кнопку . В меню «PLC» вибрати пункт «Operating mode» та у однойменному вікні натиснути кнопку «Stop», щоб зупинити контролер.

Вікно «Operating mode» також можна визвати, натиснувши на екран «Switch/Operatingmode», що у правому нижньому куті програми:



Щоб з'єднати ПЛК з комп'ютером за допомогою відповідного мережного адаптеру, зі списку «Вибір цільового інтерфейсу» вибрати «Target: TCP/IP-Direct». З меню «Extras» вибрати вкладку «TargetPropertiesofTarget: TCP/IP-

Direct» і у вікні, що з'явиться, натиснути кнопку «Accessiblenodes» та провести визначення контролера в мережі, як це робили на етапі конфігурування контролера.

Для завантаження проекту у контролер в панелі інструментів натиснути кнопку  («Send all blocks»). Якщо робоча програма змінена, то її можна завантажити у контролер без його скидання. Для цього натиснути на кнопку  («Send active block»), з'явиться вікно, яке свідчить про те, що код програми завантажено до контролера та похибок немає (як у пункті перевірки програми).

Після завантаження коду програми до контролера перевести його функціональний перемикач у режим «Run». Графобудовник почне працювати. Якщо виникне потреба припинити роботу контролера вручну, необхідно перевести перемикач у режим «Stop».

### ***Завдання для самостійної роботи***

Скоригувати програму таким чином, щоб графобудовник викреслював прямокутник поділений на 4 частини.

#### ***Зміст звіту***

Звіт повинен містити назву роботи, короткі відомості про контролер, програму, результати самостійної роботи, висновки.

#### ***Контрольні питання***

1. Аналогом якого ПЛК є контролер VIPA 300?
2. Якою перевагою характеризується контролер VIPA SPEED 7-300?
3. Як можна збільшити оперативну пам'ять ПЛК VIPA SPEED 7-300?
4. Який інтерфейс у центральних процесорів основний?
5. Яке призначення другого Ethernet порту центральних процесорів?
6. Як здійснюється програмне конфігурування контролера?
7. Що відбувається, коли перемикач на передній панелі контролера переведений у положення MRES?
8. Для яких цілей створюється FC1?
9. Як здійснюється оголошення змінних ?
10. Як здійснюється завантаження проекту у контролер?

### **12.5 Лабораторна робота №5**

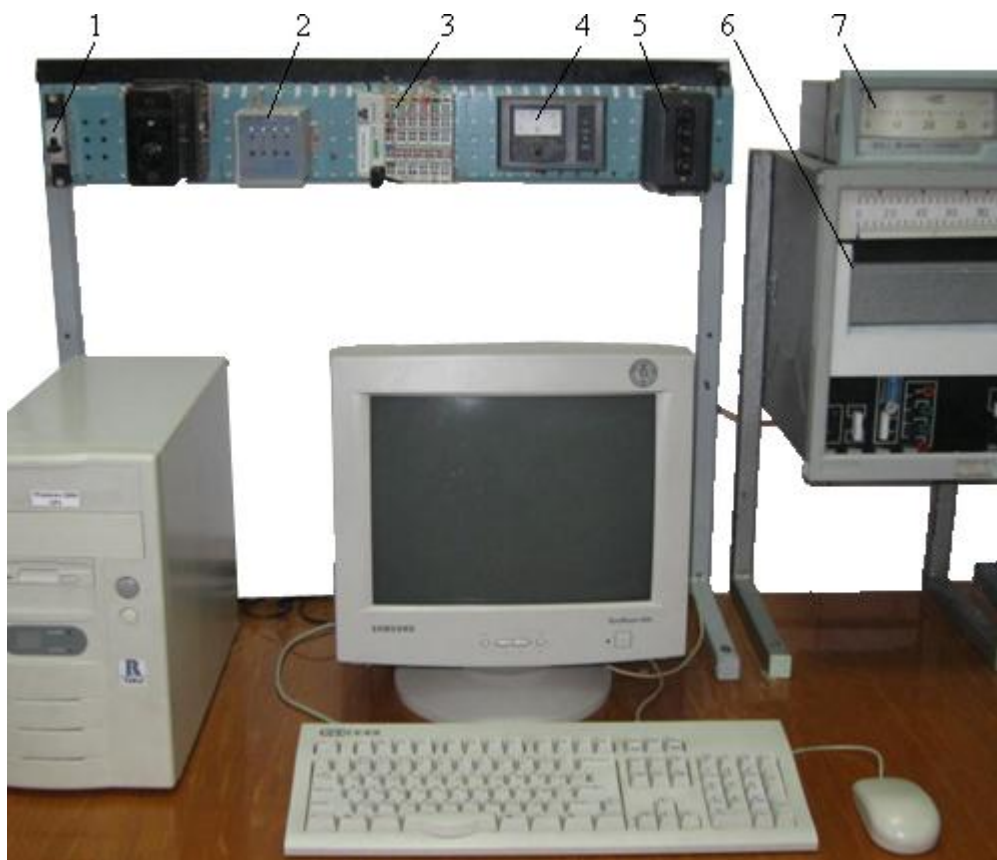
Програмування та дослідження роботи контролера WAGO 750-841

**Мета:** ознайомитися з будовою, придбати навички програмування і роботи з контролером.

Лабораторна установка, рис.12.11, призначена для оволодіння методикою програмування і роботи з контролером 750-841 фірми WAGO[64].

У лабораторній установці контролер WAGO 750-841 забезпечує рух каретки вторинного приладу шляхом формування керуючого впливу на реле,

які згідно з програмою підключають до входу РП 160 на певний відрізок часу, по-черзі, чотири опори різного номіналу.



1 - автомат вхідної напруги; 2 - блок живлення БП 96/36-4; 3 - контролер WAGO 750-841; 4 - міліамперметр вихідного струму контролера; 5 - кнопки управління положенням каретки вторинного приладу РП-160; 6 - вторинний прилад РП-160; 7 - мілівольтметр.

Рисунок 12.11 – Лабораторна установка

### **Контролер WAGO 750-841**

Контролер WAGO 750-841 є оновленою версією Ethernet TCP/IP програмованого контролера сімейства WAGO-I/O-SYSTEM. Оснащений 100-Mbit портом Ethernet, підтримує протоколи ModbusTCP і EtherNet/IP.

Зовнішній вигляд WAGO 750-841 приведено на рис.12.12.

Контролер має напругу живлення  $\approx 24V$  і USB-порт для підключення до мережі ETHERNET. Експлуатаційний режим контролера відображається за допомогою індикаторів у формі світлодіодів. В деяких випадках, вони є різноколірними (червоними/зеленими або червоними/зеленими/рожевими).

У нижній частині мережного адаптера під захисною стулкою знаходиться конфігураційний і програмний порт, а також операційний перемикач режимів.

У WAGO 750-841 відсутнє традиційне, практично для всіх ПЛК, об'єднувальне шасі. Механічним з'єднувачем для окремих модулів вводу-виводу є стандартна монтажна DIN-рейка.

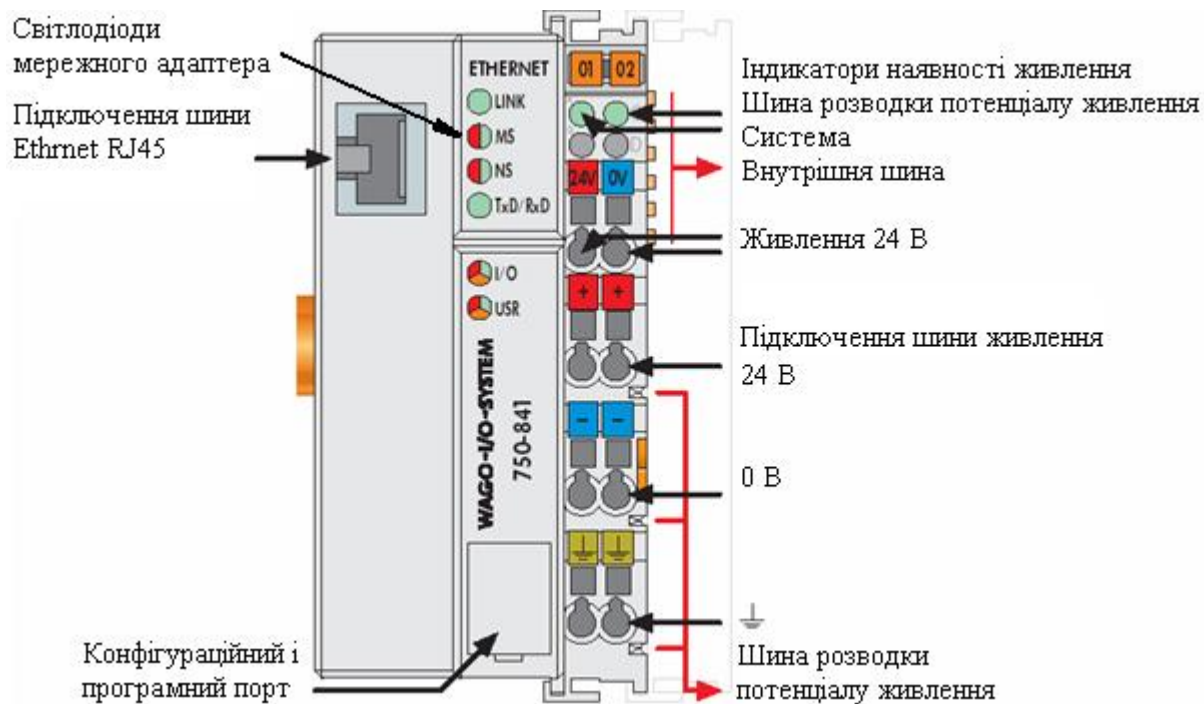


Рисунок 12.12 – Fieldbus контролер ETHERNET TCP/IP

Fieldbus-вузол лабораторної установки складається з адаптера WAGO 750-841, модуля дискретного вводу WAGO 750-402, модуля дискретного виводу WAGO 750-504, модуля аналогового вводу WAGO 750-461, модуля аналогового виводу WAGO 750-554 і кінцевого термінального модуля WAGO 750-600.

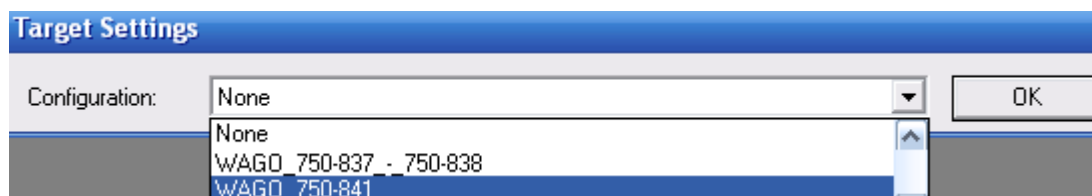
Решта інформації про контролер WAGO 750-841 надається у відповідному розділі даного посібника.

### ***Виконання лабораторної роботи***

Розробити LD-програму управління положенням каретки вторинного приладу РП-160, передбачивши затримку каретки у першому положенні на 7с, у другому на 5с, у третьому на 8с, у четвертому на 4с, а також аварійну зупинку.

### ***Створення проекту***


Запустити CoDeSys V2.3 і, використовуючи в меню шлях File-> New, лівою клавішею миші відкрити вікно налаштування цільової платформи Target Settings:



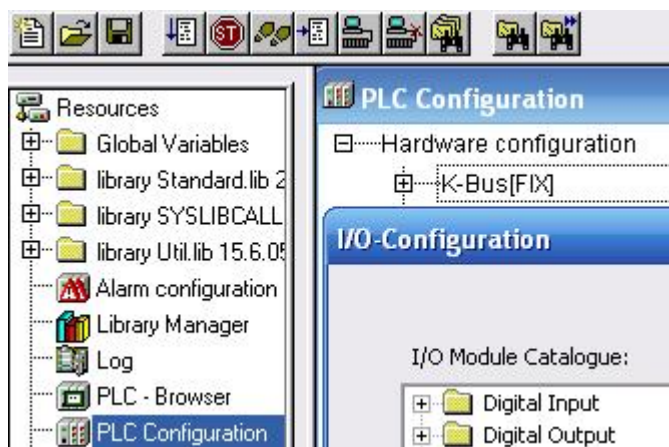


Зі списку Configuration вибрати контролер WAGO-750-841 і натиснути ліву клавішу миші. У діалоговому вікні New POU, що відкриється, позначити LD-мову, вибрати Program, як тип POU, залишити головний шаблон PLC\_PRG і натиснути ОК.

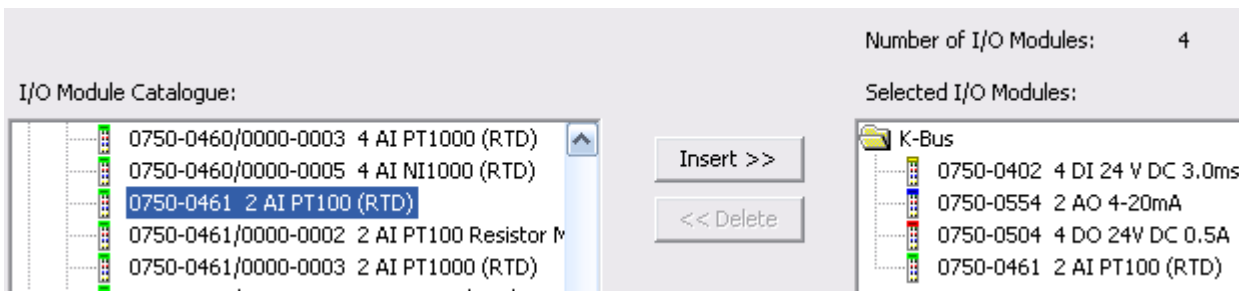
На екрані з'являться два вікна, ліворуч вікно організатора проекту POUs, а праворуч робоче поле для створення програмного коду.


Щоб здійснити конфігурування контролера, у нижній частині організатора проекту POUs треба активізувати вкладку  (Resources), і лівою клавішею миші позначити PLC Configuration, натиснути праву клавішу і вибрати Edit object. У діалоговому вікні конфігурації PLC, що з'явиться, розкрити Hardware configuration і вибрати K-Bus[FIX]. Натиснути праву клавішу миші, щоб відкрити контекстне меню і вибрати Append Subelement.

У вікні I/O-Configuration, що відкриється, знаходиться список усіх існуючих аналогових (Analog) і дискретних (Digital) модулів вводу (Input) і модулів виводу (Output).

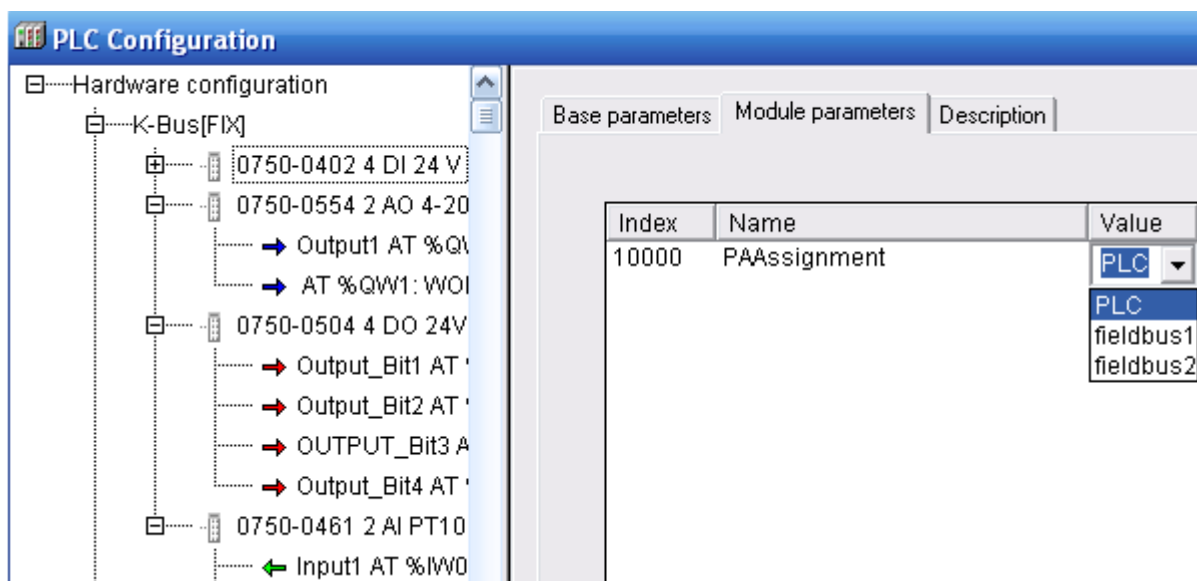


Для реалізації алгоритму управління положенням каретки вторинного приладу використовуються тільки дискретні виходи, проте необхідно при конфігуруванні PLC вказати всі модулі, що встановлені в слотах контролера. У каталозі модулів вводу – виводу, що в лівій частині вікна, двічі натиснути на “Digital Input”, щоб вибрати дискретний модуль DI 0750-0402 4 DI 24V DC 3.0ms, а потім пересунути його в праву частину вікна конфігурації, використовуючи кнопку “Insert”. Після цього, вибрати аналоговий модуль AO 0750-0554 2 AO 4-20mA з каталогу модулів вводу - виводу, натискаючи на “Analog Output”. І знову перемістити його в праву частину вікна конфігурації, використовуючи кнопку “Insert”. Аналогічні дії необхідно виконати з модулем дискретного виводу (Digital Output -> DO 0750-0504 4 DO 24V DC 0.5mA) і аналогового вводу (Analog Input -> AI 0750-0461 2 AI PT 100 (RTD)).



Якщо необхідно видалити деякі модулі, треба вибрати відповідний модуль в правій частині вікна конфігурації і натиснути кнопку "Delete". Можна змінити позицію модуля, вибираючи і переміщаючи модулі, використовуючи кнопку , що знаходиться у правій частині вікна конфігурації. Кінцевий модуль 750-600 не показують у вікні конфігурації вводу - виводу. При завершенні конфігурування – натиснути кнопку "OK".

Відкрити K-Bus[FIX] і натиснути ліву клавішу миші. Через Module parameters визначити, як здійснюється управління входами-виходами - за допомогою контролера чи протоколу Ethernet (fieldbus 1 = Modbus/TCP або fieldbus 2 = Ethernet/IP). Оскільки входами і виходами управлятиме контролер, вибрати PLC.




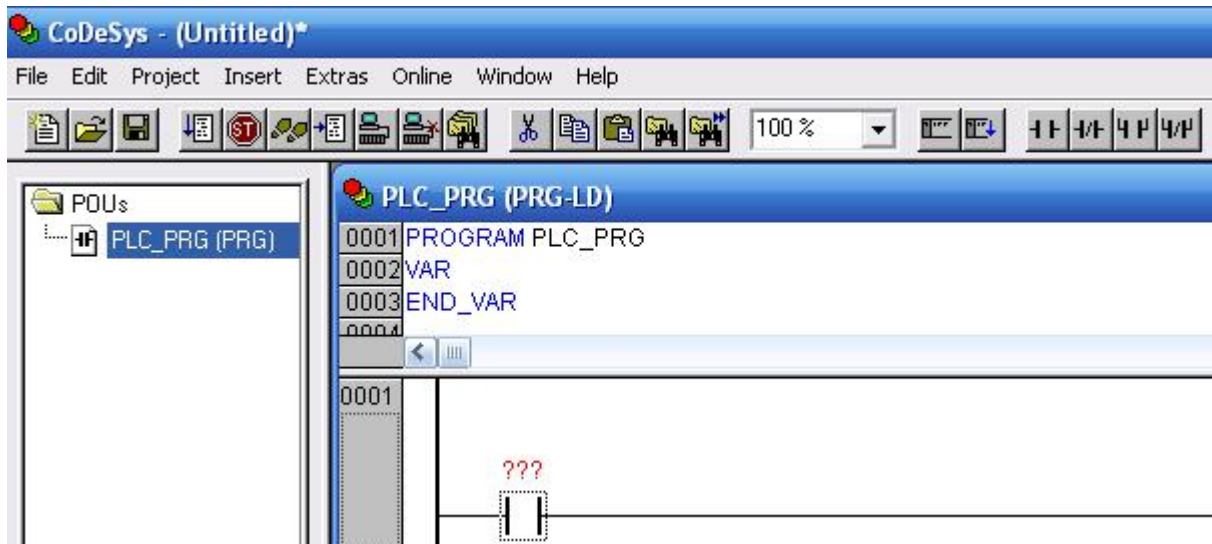
Після цього необхідно присвоїти імена вхідним і вихідним модулям контролера. Оскільки при керуванні використовується модуль дискретного виводу 0750-0504 двічі натиснути лівою клавішею по його назві.

Натискання стрілки, що поряд "AT" відкриє текстове поле, в якому ввести ім'я змінної для дискретного виходу - Output\_Bit1 (для адреси %QX2.0), Output\_Bit2 (для адреси %QX2.1), Output\_Bit3 (для адреси %QX2.2), Output\_Bit4 (для адреси %QX2.3), а також для аналогового виходу - Output\_1 (для адреси %QW0) і аналогового входу - Input\_1 (для адреси %IW0).





## Розроблення програмного коду



Для створення програми управління кареткою вторинного приладу у вікні організатора об'єктів перейти на вкладку POUs і подвійно клацнути лівою клав'яшею миші на PLC\_PRG (PRG). Відкриється вікно з першою LD-мережею і відповідною панеллю інструментів. Щоб встановити в мережу перший контакт, натиснути лівою клав'яшею миші на кнопку  в панелі інструментів:




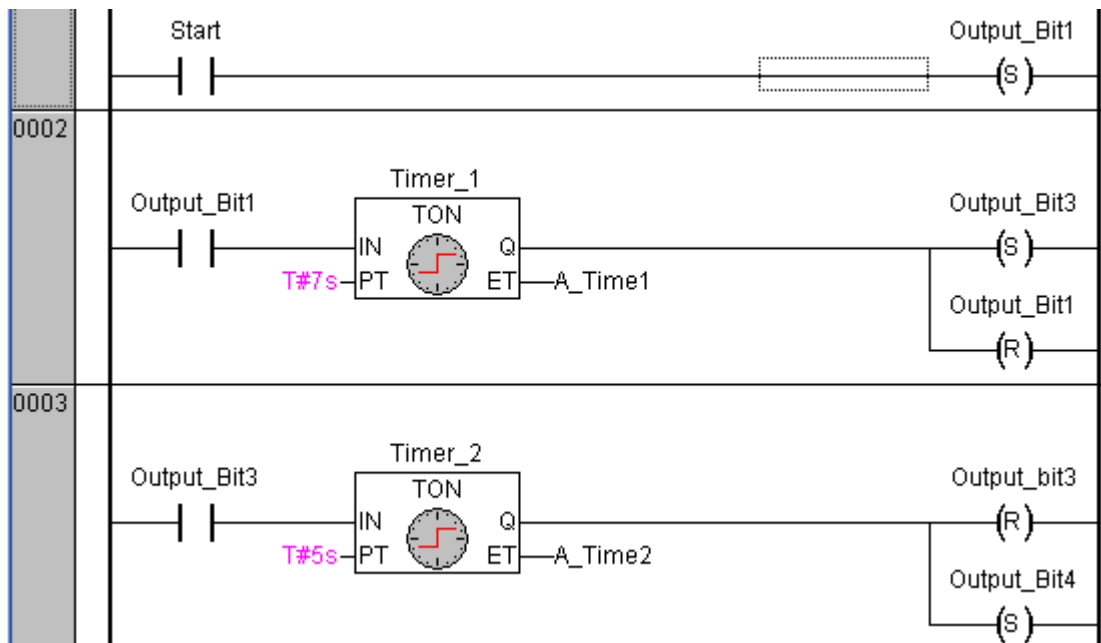
Замість знаків питань надрукувати ім'я контакту Start. Натиснути на клавіатурі Enter. У діалоговому вікні, що з'явиться, зберегти найменування (Name: Start) і логічний тип (Type: BOOL). Клас змінної - глобальний (Class:VAR\_GLOBAL), натиснути ОК. Змінна Start з'явиться у вікні глобальних змінних проекту.

Щоб додати Set-котушку, позначити лівою клав'яшею миші LD-мережу і натиснути лівою клав'яшею миші на кнопку  (coil), яка знаходиться на панелі інструментів. Змінити ім'я котушки на Output\_Bit1 (котушка буде зв'язана з першим каналом модуля дискретного виводу).

Створити нову мережу кнопкою  Network (after). Додати контакт (contact) і призначити йому ім'я Output\_Bit1. Далі додати функціональний блок. Для цього в панелі інструментів увійти в меню Insert->Function Blocks. У діалоговому вікні (зліва), що відкриється, вибрати категорію: стандартні функціональні блоки (Standard Function Blocks). З таймерів (timer) стандартної бібліотеки (standard.lib) вибрати TON (таймер із затримкою вмикання) і натиснути ОК. Щоб визначити ім'я нового екземпляра функціонального блока TON, клацнути лівою клав'яшею миші над зображенням таймера і надрукувати ім'я Timer\_1, натиснути Enter і закрити вікно оголошень кнопкою ОК. Замінити три знаки питання на вході PT константою T#7s, яка відповідає 7с. Цей час можна змінювати в процесі налагодження. Біля виходу ET задати змінну A\_time1, натиснути Enter і у вікні оголошення, що відкриється вибрати її тип (Type) – TIME.

На виході функціонального блока додати RESET-котушку кнопкою  і оголосити її, як Output\_Bit1. Позначити цю котушку лівою клавішею миші і в панелі інструментів натиснути кнопку , щоб додати паралельну SET-котушку для переміщення каретки в інше положення, і оголосити її змінну, як Output\_Bit3.

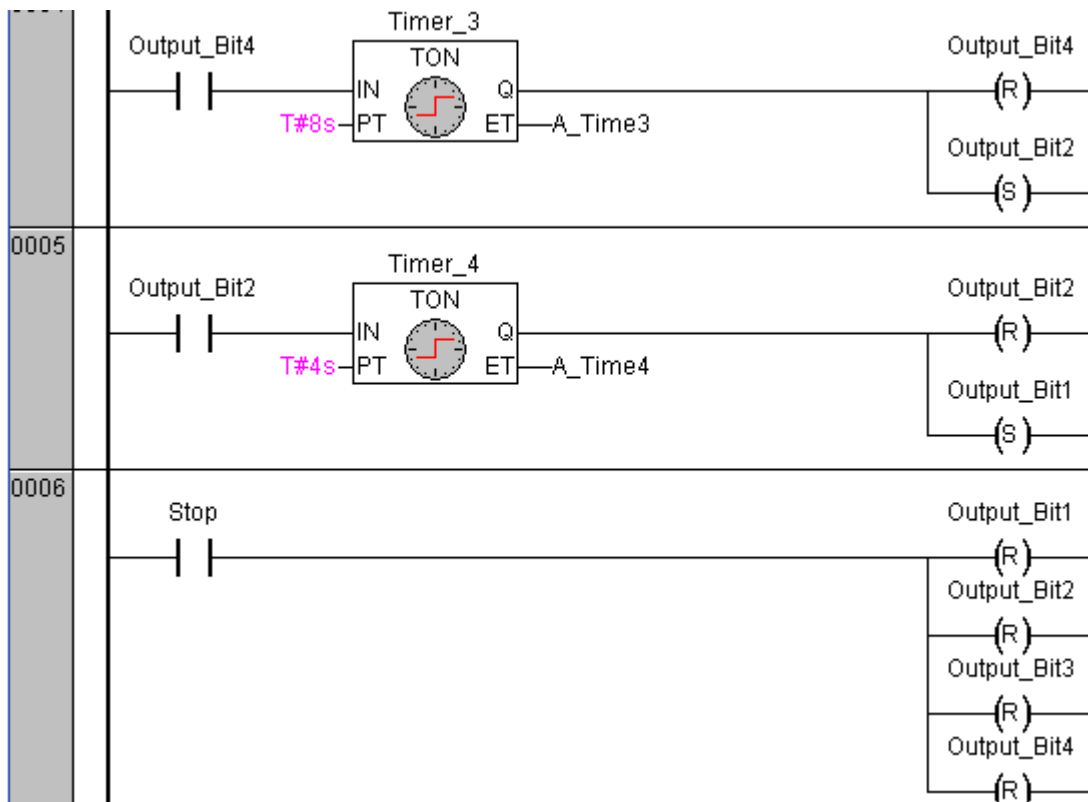
Створити ще одну мережу кнопкою . Додати контакт (contact) Output\_Bit3, встановити таймер TON з ім'ям Timer\_2. Замінити три знаки питання на вході PT константою T#5s, а на виході ET встановити змінну A\_time2 типу TIME. Розмістити на цій мережі Reset- і Set-котушки з іменами Output\_Bit3 і Output\_Bit4 відповідно. Перша половина програмного коду має вигляд:



Створити мережу 0004. Додати контакт (contact) з ім'ям Output\_Bit4. Встановити таймер TON і змінити його ім'я на Timer\_3. На вході PT надрукувати T#8s, а на виході ET - змінну A\_time4 типу TIME. Встановити Reset - і Set-котушки з іменами Output\_Bit4 і Output\_Bit2.

Створити мережу 0005, яка містить контакт Output\_Bit2, таймер TON - Timer\_4 з константою T#4s на вході PT і змінною A\_time5 типу TIME на виході ET, Reset- котушку (Output\_Bit2) і Set-котушку (Output\_Bit1).

Для аварійного зупинення програми створити мережу 0006, додати контакт Stop, який розмикатиме всі дискретні виходи, а також чотири Reset-котушки з іменами Output\_Bit1, Output\_Bit2, Output\_Bit3, Output\_Bit4. Друга половина програмного коду має вигляд:



Протягом створення програмного коду сформувався список оголошених змінних:

```


PLC_PRG (PRG-LD)
0001 PROGRAM PLC_PRG
0002 VAR
0003   Start: BOOL;
0004   Timer_1: TON;
0005   A_Time1: TIME;
0006   A_Time2: TIME;
0007   Output_bit1 AT %QX2.0: BOOL;
0008   Output_bit2 AT %QX2.1: BOOL;
0009   Output_bit3 AT %QX2.2: BOOL;
0010   Output_bit4 AT %QX2.3: BOOL;
0011   Timer_3: TON;
0012   Timer_2: TON;
0013   A_Time3: TIME;
0014   Timer_4: TON;
0015   A_Time4: TIME;
0016   Stop : BOOL;
0017 END_VAR

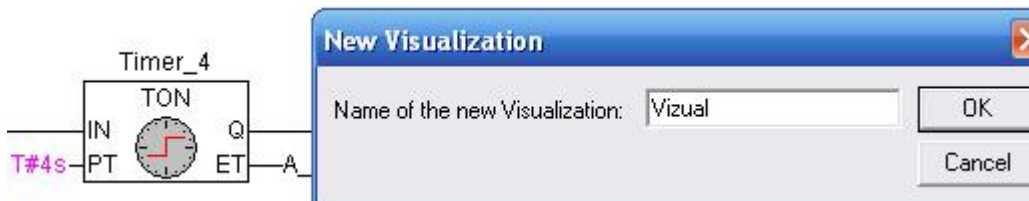
```

Для перевірки програми виконати компіляцію проекту командою меню Project -> Build або клавішею F11. CoDeSys виведе результати компіляції в нижній частині робочого екрану. При відсутності помилок та попереджень результат: 0 Error(s), 0 Warning(s). Якщо є помилки, то перейти у вікно повідомлень і клацнути кнопкою миші на тексті повідомлення. Це призведе до переміщення курсору в місце, де виникла помилка.

Повідомлення про помилки завжди необхідно аналізувати і виправляти, починаючи з першої. Попередження (warnings) вказують на потенційно небезпечні моменти, які не заважають нормальної компіляції проекту.

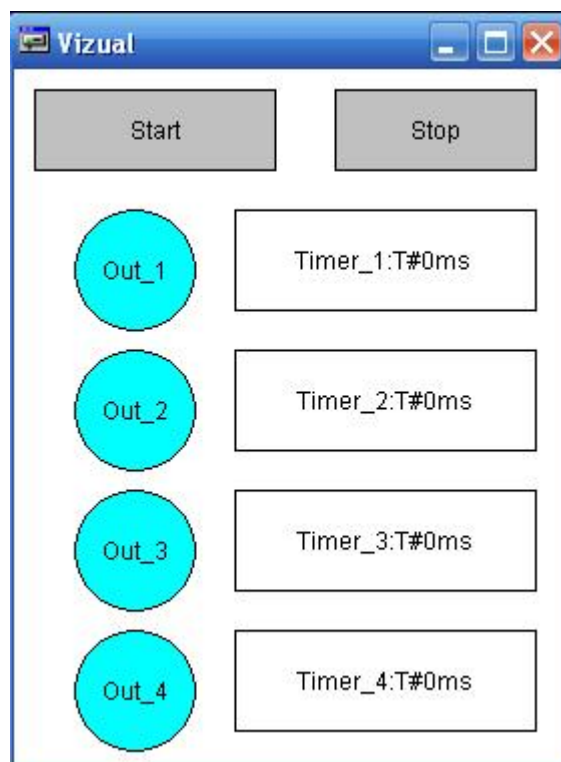
### *Візуалізація*


Щоб створити візуалізацію у проекті, необхідно кнопкою  вибрати вкладку Visualizations в організаторі об'єктів і увійти в меню Project Object Add:



Ввести ім'я візуалізації, наприклад Visual, і натиснути кнопку ОК.

Відкриється вікно, в якому створити візуалізацію. В кінці роботи, вікно візуалізації отримає такий вигляд:

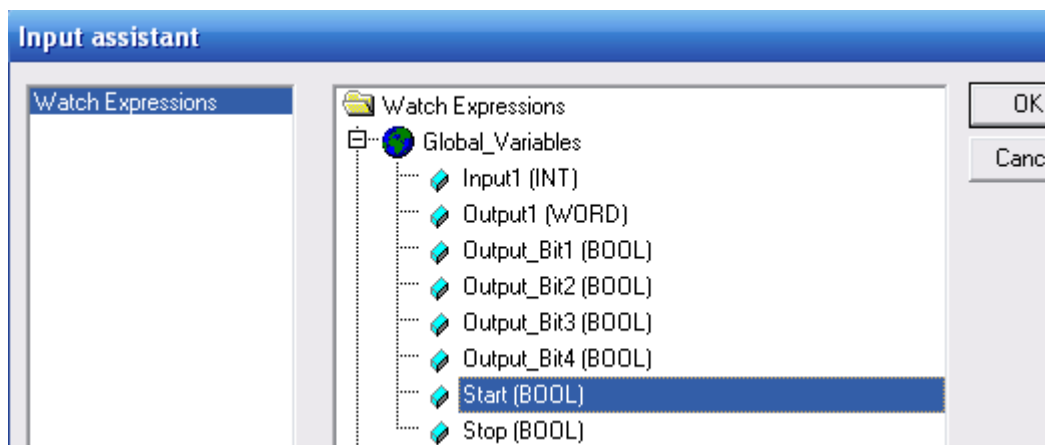


Для візуалізації кнопки Start, в панелі інструментів вибрати прямокутник (Rectangle) . У вікні редактора візуалізації натиснути ліву клавішу миші і розтягнути прямокутник до потрібної висоти і ширини, відпустити клавішу.

Двічі клацнути лівою клавiшею миші по прямокутнику. З'явиться діалогове вікно для настройки елемента візуалізації. У Category вибрати альтернативу Text і заповнити поле Contents ім'ям змінної Start:



Перейти в категорію змінних (Variable), клацнути лівою клавiшею миші в полі зміни кольору (Change color) і скористатися асистентом вводу - натиснути на клавіатурі <F2> і в діалоговому вікні, що відкриється, зі списку глобальних змінних (Global Variables) вибрати змінну Start:




Перейти в категорію колір (Colors). Натиснути кнопку Inside і задати сірий колір фарбування елемента (колір елемента при хибному значенні змінної Start). За допомогою кнопки Inside визначити колір активного стану змінної, наприклад зелений:



У категорії Input активізувати Toggle variable і ввести змінну Start.


В результаті, при значенні FALSE змінної Start прямокутник відобразатиметься сірим, а при значенні TRUE – зеленим. Значення змінюватиметься при кожному «натисненні» кнопки.

Решта елементів вікна візуалізації розробляється за аналогією з методикою створення кнопки Start.

За допомогою кнопки  в панелі інструментів намалювати коло. У настройках Text списку Category у вікні Contents надрукувати Out\_1, у настройці - Variable у рядку Change color надрукувати змінну .Output\_Bit1, у настройці - Colors кнопкою Inside визначити блакитний колір, а у настройці Alarm color кнопкою Inside - рожевий. Скопіювати створене коло командою в меню Edit-> Copy і вставити його кілька разів командою Edit->Paste для наступних положень механізму Out\_2, Out\_3, Out\_4.

Для кожного кола скоригувати настройки:

- Category-Text-> Contents для другого положення – Out\_2, для третього – Out\_3, для четвертого – Out\_4;
- Category-Variable ->Change color для другого положення - змінна .Output\_Bit2, для третього – .Output\_Bit3, для четвертого – .Output\_Bit4.

За допомогою кнопки  нарисувати прямокутник для клавіші Stop, з такими настройками:

- Category-Text , Contents - текст Stop;
- Category-Variable , Change color - змінна .Stop;
- Category-Input - прапорець Toggle variable активізований, змінна -.Stop;
- Category-Colors , Color - зафарбування Inside сірим кольором, Alarm color червоним кольором.

Намалювати білий прямокутник для таймерів з наступними настройками для положень Out\_1, Out\_2, Out\_3, Out\_4:

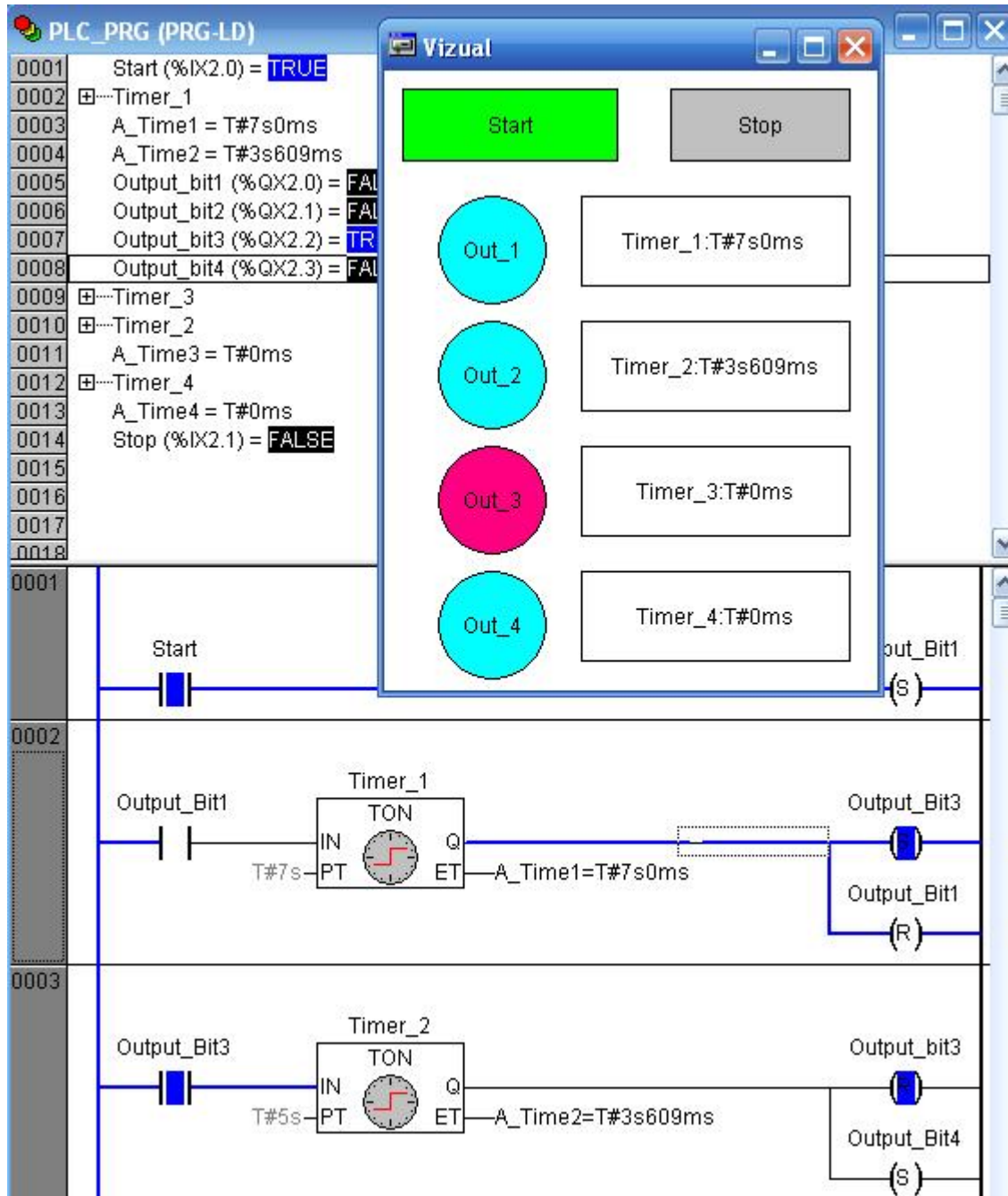
- Category-Text, Contents- текст Timer\_1:%s, Timer\_2:%s, Timer\_3:%s, Timer\_1:%s;
- Category-Variable, Textdisplay- змінні PLC\_PRG.A\_Time1, PLC\_PRG.A\_Time2, PLC\_PRG.A\_Time3, PLC\_PRG.A\_Time4.

### ***Налагодження програми***

Якщо проект без помилок, запустити його в режимі симуляції. Цей режим дозволяє перевірити роботу програми, не записуючи її в контролер. Для цього встановити прапорець в меню Online -> Simulation. Активізувати проект командою Online->Login або натисненням на панелі інструментів кнопку із зображення EOM і контролера. При цьому CoDeSys відразу виділить яскравим синім кольором активні з'єднання і контакти, тобто ті, які мають значення TRUE. Для безпосереднього запуску програми виконати команду Online -> Run або натиснути клавішу F5. Перейти у вікно візуалізації і натиснути кнопку Start. Кола Out\_1, Out\_2, Out\_3, Out\_4 почнуть змінювати колір відповідно до роботи програми, а у прямокутниках показується поточний час затримки таймерів.



Вимкнення всіх котушок здійснюється кнопкою Stop. Фрагмент робочого поля під час перевірки роботи програми має вигляд:



### ***Настроювання каналу і сполучення з контролером***

При першому підключенні контролера до CoDeSys необхідно виконати певні настройки. У меню Online відкрийте діалог Communication parameters. Натиснути кнопку New для настроювання нового сполучення. Якщо контролер запускається з даного комп'ютера вибрати драйвер RS232, і перевірити параметри вікна конфігурації, які містять наступні дані установки: Baudrate=19200, Parity=Even, Зупинка bits=1, Моторола byteorder=No. В меню Online вибрати Communication Parameters, а в меню Gatenay -> Local.



Переконайтесь в правильності настройки каналу сполучення. Зняти прапорець в меню Online -> Simulation. З'єднання з контролером здійснюється з середовища програмування CoDeSys командою Online->Login в панелі інструментів. Команда Online -> Run запускає проект. Перейти у вікно візуалізації і натиснути кнопку Start. Каретка вторинного приладу РП-160 почне переміщатися згідно з запрограмованим алгоритмом, затримуючись в певному положенні визначений час.

### ***Завдання для самостійної роботи***

Модифікуйте програму керування рухом каретки вторинного приладу, таким чином, щоб кожна зупинка каретки підраховувалась, а після 5 зупинок каретка поверталася на початок шкали.

### ***Зміст звіту***

Звіт повинен містити назву роботи, короткі відомості про контролер, програму керування кареткою, результати самостійної роботи і висновки.

### ***Контрольні запитання***

1. Що являє собою контролер WAGO 750-841?
2. Які модулі використовуються в лабораторній роботі?
3. Як здійснюється з'єднання контролера з ПК і настроювання каналу з'єднання?
4. Чим характеризується інструментальна система програмування CoDeSys?
5. Для чого призначене трасування в CoDeSys?
6. Як здійснюється візуалізація в CoDeSys і, яке її призначення?
7. З яких етапів складається програмування контролера?
8. Як здійснюється зв'язок написаної програми з фізичними входами-виходами контролера?
9. Яке призначення має кінцевий модуль?
10. Як здійснюється налагодження програми?

## **12.6 Лабораторна робота № 6**

### **Програмування та дослідження роботи контролера ADAM-5511**

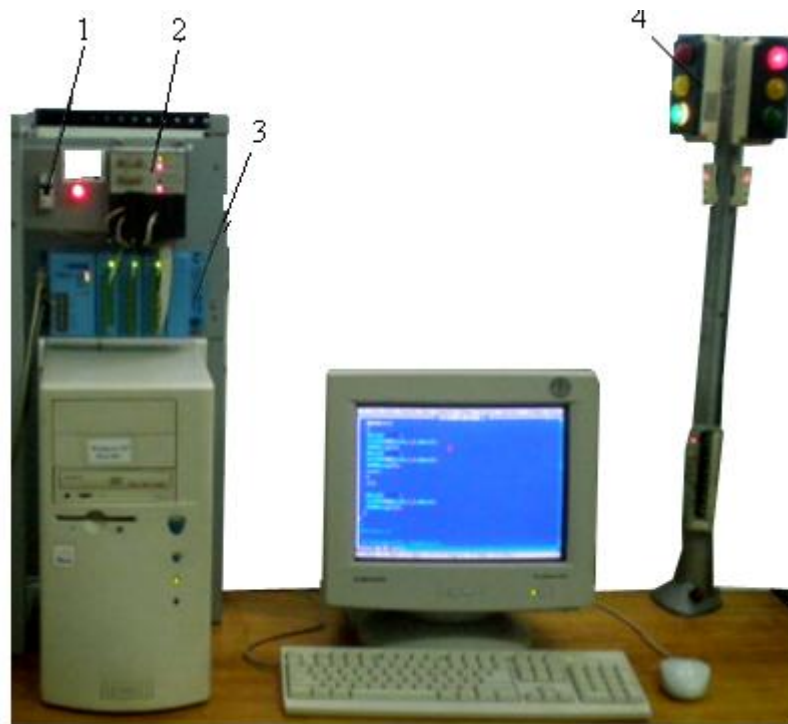
**Мета:** ознайомитися з будовою, особливостями програмування і роботи з контролером.

Лабораторна установка, рис.12.13, призначена для автоматичного управління роботою світлофора за допомогою мікропроцесорного контролера ADAM 5511. Вихідні канали модуля дискретного вводу-виводу ADAM 5050, встановленого в слоті з адресою 2, підключені до світлофору.

Для роботи з контролером за інтерфейсом RS-232 порти контролера COM1 та COM3 з'єднані з портами комп'ютера, відповідно, прямим і нуль-модемним кабелем.

У слотах контролера встановлено 3 модулі ПЗО. У слоті з адресою 0 встановлено семиканальний ПЗО ADAM 5018, призначений для роботи з термопарами.

У слоті з адресою 1 встановлено чотириканальний модуль аналогового виводу ADAM 5024. У слоті з адресою 2 встановлено шістнадцятиканальний модуль дискретних сигналів ADAM 5050. Канали модуля за допомогою DIP - перемикачів на платі модуля сконфігуровані незалежно один від одного як вихідні [65].



*1 - автомат вхідної напруги; 2 - блок живлення контролера БП-1; 3- контролер ADAM 5511; 4 - автомобільно-пішохідний світлофор.*

Рисунок 12.13 – Лабораторна установка

### ***Контролер ADAM 5511***

Контролер ADAM 5511 фірми Advantech виконаний у вигляді пластикової корзини з можливістю установлення модулів розширення. Процесор контролера IBM PC-сумісний, виконаний на базі процесора CPU 80188, працює під управлінням операційної системи «ROM-DOS» сумісної з «MS-DOS». Підтримує основні функції «MS-DOS», з можливістю виконання програм, написаних мовою «C».

Головний модуль контролера складається з центрального мікропроцесора, джерела живлення, чотирьох слотів для модулів ПЗО, трьох послідовних портів. «Flash-ROM» обсягом 256 Кбайт, «Flash-disk» обсягом 512 Кбайт, з них 400 Кбайт для користувальницького програмного забезпечення (ПЗ). «SRAM» обсягом 60 Кбайт. Контролер має годинник реального часу і сторожовий таймер. На лицевій стороні контролера розташовані індикатори стану живлення

«PWR», роботи мікропроцесора «RUN», наявності обміну по послідовному порту «COMM», стану батареїки «BATT».

У правій частині ADAM 5511 знаходяться DIP перемикачі, які призначені для встановлення робочої конфігурації контролера - адреси контролера для передачі даних відповідно до протоколу «ModBUS RTU» та швидкості передачі даних робочого COM- порту.

Три послідовних порти ADAM 5511 призначені для програмування контролера, конфігурування його модулів, відстеження зміни вхідних і вихідних сигналів.

ADAM 5511 підтримує всі дискретні і аналогові ПЗО серії ADAM 5000.

Решта інформації про ADAM 5511 надається у відповідному розділі даного навчального посібника.

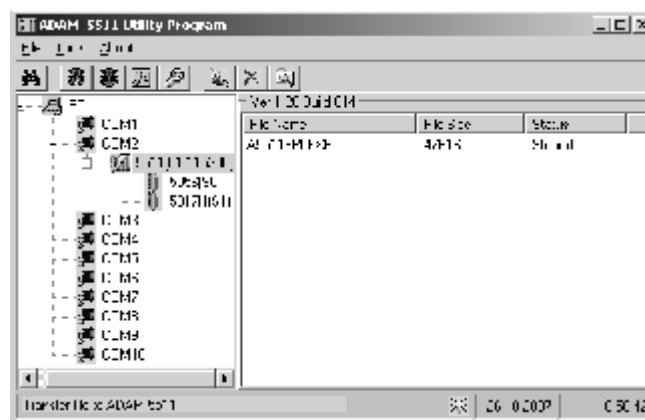
### ***Інструмент програмування контролера***

Інструментальне програмне забезпечення для роботи з контролером, що постачається разом з ADAM 5511, містить утиліту «ADAM 5511 Windows Utility», яка дозволяє встановити зв'язок з вбудованим інтерпретатором командного рядка ROM-DOS.

Ця утиліта реалізує:

- пошук підключених модулів ПЗО;
- читання значень каналів підключених модулів ПЗО;
- роботу з програмним забезпеченням (завантаження, видалення, запуск, зупинка).

Після встановлення утиліти на ПК та її запуску, з'являється вікно, яке складається з двох зон. У лівій зоні відображаються доступні послідовні порти ПК і інформація про підключені до контролера ADAM 5511 модулі ПЗО, а в правій - файлова система ADAM 5511 у вигляді віртуального диска D:\ контролера.



Програмування ADAM 5511 здійснюється за допомогою компілятора Borland Turbo C++ v.3.0.

При розробці прикладного програмного забезпечення для контролера ADAM 5511 використовуються спеціалізовані бібліотеки, що поставляються

виробником у комплекті з контролером. Для найбільш ефективного використання пам'яті контролера функції розділені на п'ять окремих бібліотек. Тому користувач повинен підключати тільки набір бібліотек, що містить необхідні для поточного проекту функції.

Щоб була можливість звернутися до каналів модулів ПЗО із зовнішніх додатків, у пам'яті контролера використовуються відповідні адреси регістрів.

Адресація регістрів для аналогових і дискретних модулів встановлених в ADAM 5511, залежить від номера слоту і показана на рис. 12.14.



Рисунок 12.14 – Адресація регістрів модулів ПЗО серії «ADAM5000»

### **Виконання лабораторної роботи**

Програмування логічної задачі виконати відповідно до алгоритму роботи автомобільно – пішохідного світлофору.

Світлофор має перебувати в чотирьох станах: зеленому, червоному, жовтому і жовто-червоному. Тривалість кожного стану 2с. На першому етапі загоряється зелене світло правого світлофора і червоне лівого, на другому етапі загоряється жовте на правому світлофорі і жовто-червоне на лівому, на третьому загоряється на правому світлофорі червоне і зелене на лівому, на четвертому етапі загоряється жовто-червоне на правому світлофорі і жовте на лівому. На пішохідному світлофорі загоряються зелене і червоне світло відповідно до автомобільного.

При створенні проектного коду використовуються функції:

- з бібліотеки «Utility.lib»:

- Перевірка наявності ознаки зупинки програми, яка отримана із програми «ADAM-5511 Windows Utility» при натисканні кнопки «stop» на панелі інструментів.

Синтаксис: `void check_prog_stop(void);`

- Пауза при виконанні програми.

Синтаксис: void ADAMDelay(unsigned int TC);

Параметр функції: кількість інтервалів часу по 55 мсек, [0 - 65535];

- з бібліотеки «IO.lib»:

- Читання дискретних входів / виходів ПЗО «ADAM-5050»

Синтаксис: void Get5050(int Board, int Bit, int Size, void \*pValue);

«Board» - номер слота [0 - 3], у який встановлений ПЗО «ADAM-5050»;

«Size» - розмір параметра, що зчитується або встановлюється функцією,

можливі значення параметра: ABit - побітовий запис, AByte - запис цілого байта, AWord - запис слова (двох байт).

- Установка дискретних виходів ПЗО «ADAM-5050»

Синтаксис: void Set5050(void \*pValue, int Board, int Bit, int Size);

Параметри функції аналогічні функції «Get5050(...)».

Тому додаємо в розроблюваний проект файли бібліотек «IO.lib», «Utility.lib», «A5050.lib».

Для створення проектного коду мовою C++, скористаємося таблицею станів дискретних виходів, табл.12.14, для модуля ПЗО, побудованою відповідно до логіки роботи світлофору:

Таблиця 12.14 Стани дискретних виходів

Сигнал світлофора	ЗПП	ЧПП	ЗПА	ЖПА	ЧПА	ЧЛП	ЗЛП	ЧЛА	ЖЛА	ЗЛА														
Дискретний вхід \ вихід	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					Значення HEX			
Етап «А»	bin	1	0	1	0	0	1	0	1	0	0	0	0	0	0	0								
	hex	A				5				0				0										0x500
Етап «В»	bin	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0								
	hex	1				1				8				0										0x1180
Етап «С»	bin	0	1	0	0	1	0	1	0	0	1	0	0	0	0	0								
	hex	4				A				4				0										0x4A40
Етап «D»	bin	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0								
	hex	1				8				8				0										0x1880

Де:

ЗПП- зелений правий пішохідний; ЖПА- жовтий правий автомобільний;

ЧПП- червоний правий пішохідний; ЧПА- червоний правий автомобільний;

ЧЛП- червоний лівий пішохідний; ЗЛА- зелений лівий автомобільний;

ЗЛП- зелений лівий пішохідний; ЖЛА- жовтий лівий автомобільний;

ЗПА- зелений правий автомобільний; ЧЛА- червоний лівий автомобільний.

Робота світлофора складається з 4 етапів. Керування здійснюється програмно командою розміром 2 байти, записаний у 16-річному коді. Значення параметра переданого командою перетворюється у 2-річну форму і управляє станом дискретних виходів модуля.

Структура програми складається з наступних блоків:

```
//-----
// Підключення бібліотек і файлів
//-----
#include <stdio.h> // підключення заголовного файлу, що входить у комплект
```

```

// IDE Turbo C++, що містить посилання на функції бібліотек
#include "c:\tc\adam\5511drv.h" // підключення заголовного файлу, що містить
    // прототипи бібліотечних функцій і константи
//-----
//Початок роботи програми. Оголошення змінних. Вхід у нескінченний цикл.
//-----
void main()
{
unsigned int WordData; // оголошення 2- байтної змінної без знака
int NSlot = 2; // оголошення знакової 2- байтної змінної
    while(1) // нескінченний цикл
    {
//-----
        // Установка значень дискретних виходів модуля на етапі А, що забезпечують
        //включення ЗПП, ЗПА, ЧЛП, ЧЛА.
//-----
        WordData = 0x500; // ініціалізація значенням для Етапу «А»
        Set5050(&WordData, NSlot, 0, AWord); // установка дискретних виходів
        ADAMDelay(36); // пауза 2 секунди
        Get5050(NSlot, 0, AWord, &WordData); // читання стану дискретних виходів
        printf("AWord = %x\n", WordData); // видача відладного повідомлення
        // через «COM3» контролера
//-----
        // Установка значень дискретних виходів модуля на етапі В, що забезпечують
        //включення ЖПА, ЧЛА, ЖЛА.
//-----
        WordData = 0x1180; // ініціалізація значенням для Етапу «В»
        Set5050(&WordData, NSlot, 0, AWord); // установка дискретних виходів
        ADAMDelay(36); // пауза 2 секунди
        Get5050(NSlot, 0, AWord, &WordData); // читання стану дискретних виходів
        printf("AWord = %x\n", WordData); // видача відладного повідомлення
        // через «COM3» контролера
//-----
        // Установка значень дискретних виходів модуля на етапі С, що забезпечують
        //включення ЧПП, ЧПА, ЗЛП, ЗЛА.
//-----
        WordData = 0x4A40; // ініціалізація значенням для Етапу «С»
        Set5050(&WordData, NSlot, 0, AWord); // установка дискретних виходів
        ADAMDelay(36); // пауза 2 секунди
        Get5050(NSlot, 0, AWord, &WordData); // читання стану дискретних виходів
        printf("AWord = %x\n", WordData); // видача відладного повідомлення
        // через «COM3» контролера
//-----
        // Установка значень дискретних виходів модуля на етапі D, що забезпечують
        //включення ЖПА, ЧПА, ЖЛА.
//-----
        WordData = 0x1880; // ініціалізація значенням для Етапу «D»
        Set5050(&WordData, NSlot, 0, AWord); // установка дискретних виходів
        ADAMDelay(36); // пауза 2 секунди
        Get5050(NSlot, 0, AWord, &WordData); // читання стану дискретних виходів
        printf("AWord = %x\n", WordData); // видача відладного повідомлення
    }
}

```

```
// через «COM3» контролера  
check_prog_stop(); // перевірка ознаки «зупинка програми»  
// якщо ознаки «зупинка програми» немає - перехід на Етап «А»  
}  
}
```

Завантаження виконуваного файлу у контролер здійснюється за допомогою утиліти «ADAM-5511 Windows Utility».

### ***Завдання для самостійної роботи***

Модифікувати код програми керування світлофором так, щоб зелене світло завершувало своє світіння трикратним миготінням.

### ***Зміст звіту***

Звіт має містити назву роботи, короткі відомості про контролер, структуру програми керування світлофором, результати самостійної роботи і висновки.

### ***Контрольні запитання***

1. Що являє собою контролер ADAM5511?
2. Чим характеризуються модулі ПЗО, що використовуються у роботі?
3. Яке призначення і основні можливості утиліти «ADAM-5511 Windows Utility»?
4. Для чого призначені послідовні порти контролера?
5. Що є інструментом розробки користувальницького програмного забезпечення і чим він характеризується?
6. З яких блоків складається програма керування світлофором і які функції використовуються?
7. Які кабелі використовуються для приєднання контролера до ЕОМ?
8. Яке призначення має область користувальницьких регістрів MODBUS?
9. Яке призначення має сторожовий таймер?
10. У чому відмінності контролерів ADAM5511 і ADAM5510?

## **12.7 Лабораторна робота № 7**

Програмування та дослідження роботи контролера ADAM-5510KW

**Мета:** ознайомитися з будовою, особливостями програмування і роботи з контролером.

Лабораторна установка, рис.12.15, призначена для програмування і дослідження роботи мікропроцесорного контролера ADAM 5510KW. Контролер ADAM 5510KW отримує напругу живлення від блока БПН-24. Через порт COM1 він підключений до комп'ютера.

В лабораторній роботі забезпечується керування рухом каретки вторинного приладу РП-160 і роботою виконавчого механізму МЕО, відповідно до алгоритмів запрограмованих у мікропроцесорному контролері ADAM 5510KW.





1 – виконавчий механізм МЕО; 2 – вторинний прилад РП-160; 3 – контролер ADAM 5510KW; 4 – плата з реле і опорами; 5 – джерело живлення контролера; 6 - автомат напруги живлення.

Рисунок 12.15 – Лабораторна установка

### ***Контролер ADAM 5510KW***

Мікроконтролер ADAM 5510KW складається з двох основних частин- базового блоку та модулів вводу - виводу. Базовий блок має у своєму складі процесорний модуль, перетворювач напруги, 4-слотову пасивну об'єднувальну панель та чотири серійні комунікаційні порти.

У якості CPU використовується 16-розрядний мікропроцесор 80188-40, з вбудованою Flash-пам'яттю на 1,5 Мбайт для завантаження операційної системи ROM-DOS і програмних додатків користувача. Статична оперативна пам'ять (SRAM) - 640 Кбайт.

На відміну від PC-сумісних контролерів ADAM 5510 або ADAM 5511, програмування яких відбувається мовами Асемблер або C++, ADAM-5510KW програмується зручними технологічними мовами стандарту IEC 61131-3 за допомогою інструментальної Softlogic-системи програмування KW MULTIPROG. У якості системи виконання реального часу, використовується операційна система KW PROCONOS, яка забезпечує виконання одночасно до 16 задач.

ADAM-5510KW має чотири серійні комунікаційні порти. Ці порти дозволяють програмувати та конфігурувати контролер, контролювати і об'єднувати віддалені пристрої вводу – виводу.

За допомогою DIP-перемикачів, що розміщуються за четвертим слотом, встановлюється мережний ідентифікатор пристрою, протоколи портів COM1, COM2 (MULTIPROG Protocol або MODBUS Protocol) та швидкість обміну даними від 9600 до 38400 біт/сек.

ADAM-5510KW має чотири слоти для встановлення модулів вводу – виводу, які нумеруються від 0 до 3.

В лабораторній установці використовується контролер з двома модулями. У слоті з адресою 0 встановлено семиканальний ПЗО «ADAM 5018», призначений для роботи з термopарами.

У слоті з адресою 1 встановлено шістнадцятиканальний модуль дискретних сигналів «ADAM 5050».

Решта інформації про контролер ADAM-5510KW наведена у відповідному розділі даного навчального посібника.

### ***Інструментальна система програмування KW MULTIPROG***

MULTIPROG – це SoftLogic-система програмування промислових логічних контролерів, яка відповідає стандарту IEC 61131-3. Система базується на віконній технології, яка використовує графічні можливості MS - Windows. Тому принципи стандарту здебільшого реалізуються графічно з символами і іконами або діалогами, в яких можуть бути визначені властивості елементів.

За допомогою MULTIPROG можна програмувати на двох текстових ST та IL і трьох графічних – LD, FBD та SFC мовах стандарту IEC 61131-3. Здійснюється це в діалоговому режимі, використовуючи, так званого, Project Wizard (Майстер Проекту).

Графічний редактор системи дозволяє вільне розміщення об'єктів на робочому полі проекту, вибирати мову програмування та використовувати бібліотеку операторів, функцій і функціональних блоків. Він підтримує змішування графічних мов в одному робочому листку, вставлення нових елементів в існуючі мережі, переміщення об'єктів або мереж.

Текстовий редактор дозволяє легко і швидко створювати програми, завдяки кольоровому зображенню ключових слів, використанню бібліотечних блоків і елементів.

Система програмування має могутній засіб налагодження із зручним інтерфейсом користувача. За допомогою MULTIPROG перевірка програми контролера здійснюється на симуляторі без участі самого контролера.

MULTIPROG працює разом з програмованою операційною системою реального часу ProConOs (Programmable Controller Operating System), яка вбудована в контролер.

### ***Виконання лабораторної роботи***

Розробити LD-мовою програму управління роботою виконавчого механізму, який починає працювати при натисканні кнопки «Пуск» і за 20с вимикається.

#### ***Створення нового проекту***

Запустити MULTIPROG V3.3, клацнувши лівою клавiшею миші на піктограмі MULTIPROG, з'явиться екран MULTIPROG. Увійти в меню File →New project.

Подвійно клацнути на піктограмі Project Wizard (Майстер проекту) і ввести ім'я проекту. Майстер проекту збереже проект у відповідному файлі з розширенням \*.mwt і створить підтеку того ж імені, в якій запам'ятовуються тіло коду, файли змінних і т.ін.

За замовчуванням шлях проекту вводиться автоматично. Для створення іншого шляху треба натиснути кнопку з трьома крапками.

У діалоговому вікні Select Directory вибрати теку для нового проекту і клацнути ОК.

Натиснути кнопку Next і у рядок Name of POU (Імя POU) ввести ім'я першого організатора програми (POU), наприклад Main (Головна). Вибрати FBD-мову. Натиснути кнопку Next і ввести ім'я конфігурації, наприклад, Configuration.

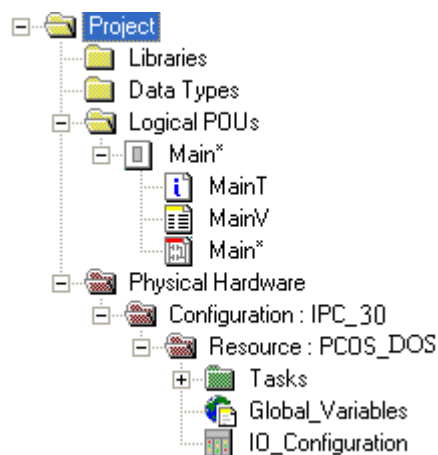
Вибрати тип конфігурації проекту IPC\_30, тому що компілятор генерує intel- код для системи виконання ProConOS 3.0. Натиснути кнопку Next, ввести ім'я ресурсу – Resource і вибрати тип ресурсу - PCOS\_DOS.

Натиснути кнопку Next, ввести ім'я задачі -Task і вибрати тип задачі - Cyclic.

Натиснути кнопку Next. На екрані з'явиться зміст зроблених установок. Якщо помилок немає натиснути Finish, в іншому разі треба скоригувати кроки проекту де є помилка, користуючись кнопкою Back.

Створений новий проект Main з'явиться у вікні проектного дерева.

В меню дерева проекту розкрити теку Physical Hardware.



Для модифікації імені конфігурації правою клавiшею миші клацнути по Configuration: IPC\_30, а потім по "Properties (Властивості)". У вікні, що

відкриється, надрукувати ім'я конфігурації (Name), наприклад, ADAM 5510KW\_Project і натиснути ОК.

Ім'я конфігурації в дереві проекту зміниться на ADAM 5510KW\_Project: IPC\_30, при цьому IPC\_30 додається автоматично.

В меню дерева проекту клацнути правою клав'яшею по Ressours: PCOS\_DOS, а у підменю, що з'явиться, клацнути по "Properties..." і змінити ім'я на ADAM 5510KW, натиснути ОК.

Для конкретизації цільового об'єкту завантаження - симулятор реального часу чи комунікаційний порт реального ADAM 5510KW, необхідно встановити ресурс.

Клацнути правою клав'яшею миші по ADAM 5510KW: PCOS\_DOS. У підменю, що відкриється активізувати "Settings...". На екрані з'явиться діалогове вікно: PCOS\_DOS, в якому:

COM- вказує у який серійний порт відправити програму;

Simulation- вказує в який симулятор відправити програму;

Baud- конкретизує швидкість компіляції у Бодах через вибраний порт;

Stopbits- конкретизує кількість стопових біт при комунікації через вибраний порт;

Databits- конкретизує кількість бітів, що передаються через порт;

Parity- вказує біт парності при передачі через вибраний порт;

Timeout- конкретизує інтервал часу (мс), протягом якого система програмування з'єднується з технічними засобами.

Для установлення порту контролера необхідно у каталозі «Port:» активізувати альтернативу COM1.

Для установлення швидкості компіляції у таблиці «Baud:» зберегти вказані уставки.

Натиснути кнопку "Data Area" і визначити області пам'яті для користувача, системних прапорців і резерву POUs, відповідно до даних наведених у діалоговому вікні "Data Area" :

Data Area	
Non retain	
Start user:	0
End user / Start system:	0
End system (max 63999):	63999
Reserve per POU: 500	
Start user:	64000
End user / Start system:	64000
End system (max 65530):	65530
Reserve per POU: 10%	

Для редагування робочого листка конфігурації модулів вводу-виводу (В/В) використовується діалог " I/O\_Configuration", який звичайно містить

початок і кінець адреси модулів В/В, декларацію пристрою (ім'я драйвера та адреса пам'яті) і т.ін.

Щоб відкрити діалогове вікно, двічі клацнути по "I/O\_Configuration" у проектному дереві.

Якщо у вікні "I/O\_Configuration" установлені непотрібні модулі вводу "IN" і виводу "OUT", то для видалення необхідно їх по черзі активізувати лівою клавішею миші і натиснути кнопку "Delete"(Видалити).

Відповідно до комплектації контролера, що використовується у лабораторній установці, у вікні "I/O\_Configuration" необхідно додати в слот 0 модуль аналогового вводу ADAM 5018, а в слот 1– модуль дискретного вводу-виводу ADAM-5050. Для цього необхідно:

- у діалоговому вікні "I/O\_Configuration" вибрати вкладку "INPUT" (Вхід) і натиснути кнопку "Add"(Додати). На екрані з'явиться діалогове вікно Add I/O Group.

У полі Name (Ім'я) присвоїти модулю ADAM 5018 ім'я AI\_5018 (Тут AI – скорочено аналоговий ввід), а у полі Task вибрати Task.

Для установлення логічної адреси модуля вводу-виводу існує три параметри:

- стартова адреса;
- місткість каналів модуля;
- кінцева адреса.

Для стартової адреси система сама знайде вільну адресу.

Місткість каналів модулів визначається їх призначенням:

Тип модуля	Кількість каналів	Місткість
Аналоговий ввід	8	16 бітів
Аналоговий вивід	4	8 бітів
Дискретний ввід	16	2 біти
Дискретний вивід	16	2 біти
Релейний вивід	8	1 біт

Кінцеву адресу система розраховує автоматично.

Оскільки місткість каналів аналогового модулю вводу ADAM-5018 дорівнює 16 бітам, у полі початкової адреси має бути 0, а у полі місткість – 16. При цьому кінцева адреса (End address) – 15 визначається автоматично після установлення Board I/O Module (Панелі модуля В/В).

При установленні Board I/O Module, вибрати ADAM-5510KW, щоб забезпечити підтримку модулів ADAM-5000. Якщо модулі вводу-виводу серії ADAM- 4000, – необхідно вибрати Read Modbus slave.

Після вибору Board I/O Module натиснути кнопку Driver parameter (Параметр драйвера), щоб забезпечити можливість установлення модулів серії ADAM-5000.

Оскільки вибрано ADAM-5510KW, то ім'я драйвера за замовчуванням – ADAMDrv.

Місце розташування модуля ADAM-5018 вказує на те, що адреса слота встановлюється як 0. У полі Module name (Ім'я модуля) вибрати ADAM-5018 [7T/C]. Тут 7- кількість каналів, "T/C"- скорочено визначає аналоговий ввід сигналів термопар.

Після вибору імені модуля система автоматично вибере тип даних – WORD.

Щоб встановити серійний порт треба натиснути кнопку "ADAM5510KW system setup" і вибрати для модуля вводу формат даних (Data Format) – NOPARITY,8.1. Клацнути ОК і конфігурація модуля вводу з'явиться у відповідному вікні.

Щоб додати модуль дискретного вводу-виводу треба у діалоговому вікні "Configuration I/O" вибрати OUTPUT і натиснути кнопку "Add...". З'явиться діалогове вікно Add I/O Group, в якому дискретний модуль ADAM-5050 ідентифікувати у графі Name, як DIO\_5050. У полі Task: вибрати Task і визначити логічну адресу (Logical addressed), як: Start address – 0, Length–2, а панель модуля вводу-виводу (Board I/O Module) і параметр драйвера (Driver parameter) за аналогією з модулем вводу.

При цьому у полі Name за замовчуванням з'явиться ADAMDrv.

Ідентифікатор пристрою (Device ID) дорівнює 1.

Оскільки ADAM5050 розташований у слоті 1, то у полі Slot надрукувати 1. Ім'я модуля вводу-виводу показати у полі Module name: ADAM 5050 [16UDIO]. Тут 16 – кількість каналів, DIO – дискретний вхід-вихід.

Щоб встановити серійний порт треба натиснути кнопку "ADAM5510KW system setup" і в Data Format (Формат даних) вибрати NOPARITY,8.1 для модуля вводу-виводу. Натиснути кнопку ОК і ADAM-5050 потрапить до списку модулів OUTPUT.

### ***Розроблення програмного коду***

В проектному дереві відкрити теку Logical POUs, вибрати останню вкладку Main і двічі клацнути лівою клав'яшею миші. Відкриється робоче поле для створення коду програми.

Для створення проектного коду LD-мовою, на робочому листку вибрати зручне місце для першої мережі і клацнути лівою клав'яшею миші.

В панелі інструментів клацнути по іконі Contact network (Мережа контактів), в установленому місці з'явиться LD- мережа 001 з контактом і котушкою.



Двічі клацнути лівою клав'яшею миші по контакту C000, щоб оголосити змінну, яка має запустити двигун. У діалоговому вікні Contact/Coil Properties (Властивості Контакт/Котушка), що з'явиться, замінити типове ім'я змінної C000 на Start.



Натиснути кнопку Apply (Застосувати) і відкрити сторінку діалогу Common (Загальний). У вікні списку Usage (Вжити) вибрати VAR, а у рядку Data Type (Тип даних) - BOOL.

Для призначення змінній Start фізичної адреси модуля дискретного вводу ПЛК, необхідно у полі I/O address (Адреса вводу/виводу) надрукувати %IX1.0, тут 1.0 – адреса нульового каналу модуля; I – фізичний вхід; X – означає однобітний розмір.

Відкрити по черзі сторінки діалогів Local scope (Локальна компетенція) і Global scope (Глобальна компетенція) і активізувати в них Default. Клацнути ОК, щоб змінна стала оголошеною. На робочому листку проекту з'явиться глобальна зміна Start замість C000.

Двічі клацнути по котушці C001, і у діалоговому вікні Contact/Coil Properties (Властивості Контакт/катушка), що з'явиться, замінити ім'я C001 на Motor. Оскільки двигун має працювати безперервно після його увімкнення, у рядку Type (Тип) вибрати –(S)–(Увімкнути вихід і зафіксувати) і натиснути кнопку Apply (Застосувати).

Відкрити сторінку діалогу Common (Загальний). У вікні списку Usage (Вжити) вибрати VAR, а тип змінної – BOOL вибрати у рядку Data Type.

Для призначення змінній Motor фізичної адреси, у рядку I/O Address ввести %QX1.1,

тут Q – позначає фізичний вихід; 1.1 – адреса восьмого каналу; X – означає однобітний розмір.

Відкрити по черзі сторінки діалогів Local scope і Global scope, щоб активізувати в них Default. Клацніть ОК і змінна буде оголошеною, а у LD-мережі замість катушки C001 з'явиться S-катушка з іменем Motor.

Для установлення нової мережі, клацнути лівою клав'яшею миші нижче першої LD- мережі, а в панелі інструментів активізувати ікону Contact Network (Мережа контактів).

Двічі клацнути по контакту C002, що у другій мережі. У діалоговому вікні Contact/Coil Properties зі списку вже оголошених змінних вибрати Motor і клацнути лівою клав'яшею миші. Натиснути ОК для зміни імені контакту C002 на Motor.

Клацнути по лінії сполучення між контактом Motor і C003, щоб позначити місце установлення таймера. В панелі інструментів клацнути по іконі Edit Wizard (Майстер редагування), на екрані з'явиться вікно Group (Група). З групи Function blocks курсором вибрати таймер TON і двічі клацнути лівою клав'яшею миші, з'явиться діалогове вікно Variable properties (Властивості змінної). У рядку Name (Ім'я) ввести M\_Time і послідовно натиснути кнопки Apply (Застосувати) і ОК, підтверджуючи завершення діалогу.

Для визначення часу роботи таймера лівою клав'яшею миші двічі клацнути по блакитній точці RT-входу. У діалоговому вікні Variable properties, що з'явиться, у рядку Name (Ім'я) ввести T#20s,



тут T – префікс часу; # - означає константу; 20s – час роботи двигуна (20с).

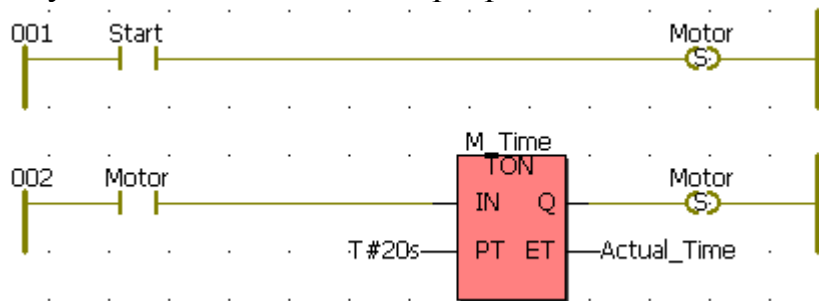
Клацнути ОК і на РТ-вході лічильника з'явиться константа часу.


Щоб визначити змінну для ідентифікації фактичного часу роботи таймера, двічі клацнути по зеленій точці ЕТ-виходу таймера. У діалоговому вікні Variable properties, що з'явиться, у рядку Name (Ім'я) ввести Actual\_Time, як ім'я локальної змінній.

Натиснути кнопку Apply (Застосувати), відкриється сторінка діалогу Common (Загальний). Зі списку Data\_Type (Тип даних) вибрати Time і натиснути ОК. На ЕТ-виході таймера з'явиться Actual\_Time.

Двічі клацнути по котушці С003, відкриється діалогове вікно Contact/Coil Properties. Для зупинки двигуна курсором зі списку змінних вибрати ім'я Motor, а у вікні Type - котушку –(R)– і клацнути ОК.

На робочому полі залишиться код програми:



Щоб перевірити програму на наявність помилок, необхідно кнопкою  Make у панелі інструментів здійснити компіляцію.

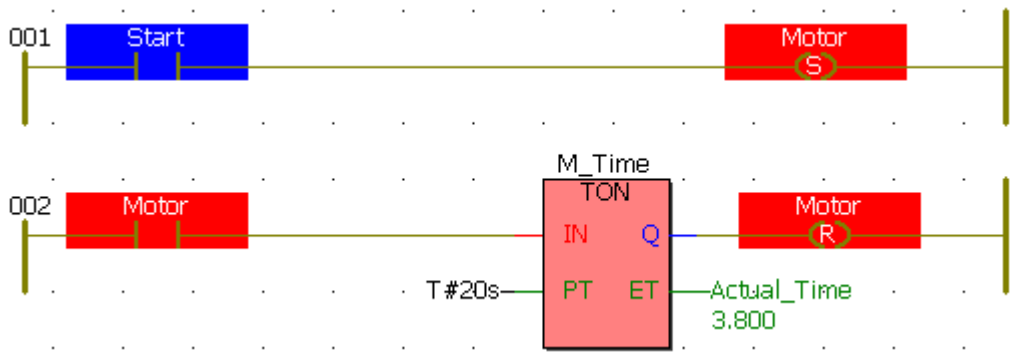
### ***Завантаження проекту до ПЛК***

Клацнути по іконі Project Control Dialog, з'явиться діалогове вікно Resource: ADAM 5510KW. Послідовно натиснути кнопки Stop, Reset і Download, з'явиться діалогове вікно, в якому в меню Bootproject (Початкове завантаження) натиснути кнопку Download . Проект завантажиться у пам'ять ПЛК, про що свідчить біг синьої стрічки внизу екрана.

У вікні Resource (Ресурс): (ADAM 5510KW) повторно натиснути Download. У вікні, що з'явилося, в меню Bootproject натиснути кнопку Activate.

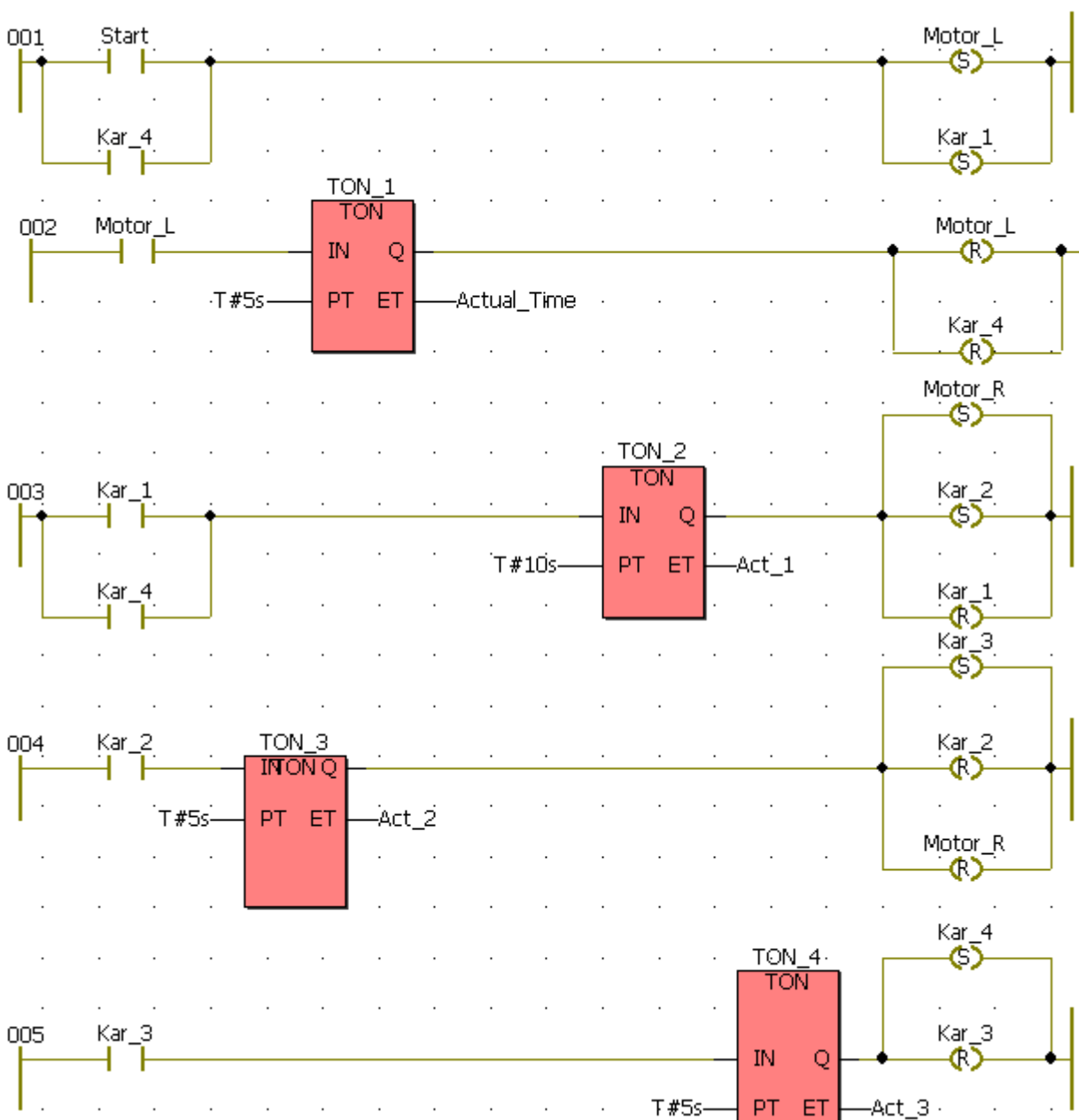
Після завантаження проекту і його активації, за умови відсутності помилок, можна запустити виконання програми, користуючись Cold start (Холодний старт), Warm start(Теплий старт) або Hot start(Гарячий старт) у вікні Resource (Ресурс).

Щоб відстежувати роботу програми, можна скористатися оперативним налагоджувальним режимом. Для цього в панелі інструментів активізуйте ікону Debug on/of (Налагодження on/of). Всі змінні проектного коду, що знаходяться у хибному стані отримують синій фон. Коли змінна протягом виконання програми отримує істинний стан, то її синій фон змінюється на червоний.



### Завдання для самостійної роботи

Запрограмувати контролер і дослідити роботу виконавчого механізму і каретки вторинного приладу відповідно до наведеної LD-програми:



. Тут: Kar1(%QX1.3), Kar2(%QX.4), Kar3(QX1.5), Kar4(QX1.6) – вихідні змінні, що визначають положення каретки на вторинному приладі;  
Start (%IX.0)- Вхідна змінна, яка запускає в роботу програму;  
Motor\_R (%QX1.1)– вихідна змінна, яка вмикає в роботу двигун праворуч;  
Motor\_L(%QX1.2) – вихідна змінна, яка вмикає в роботу двигун ліворуч;  
TON1...TON6 – таймери затримки руху каретки і роботи двигуна.

### ***Зміст звіту***

Звіт повинен містити назву роботи, короткі відомості про контролер, програми керування виконавчим механізмом і кареткою приладу і висновки.

### ***Контрольні запитання***

1. Що являє собою контролер ADAM 5510KW?
2. Які модулі використовуються в лабораторній роботі?
3. Як виконується конфігурування модулів вводу-виводу ?
4. Чим характеризується інструментальна система програмування MULTIPROG?
5. Як створюється проектний код?
6. Яке призначення Майстра проекту?
7. З яких етапів складається програмування контролера?
8. Як завантажуються проект у контролер?
9. Що таке ресурс програми і для чого він потрібен?
10. Як здійснюється налагодження програми?

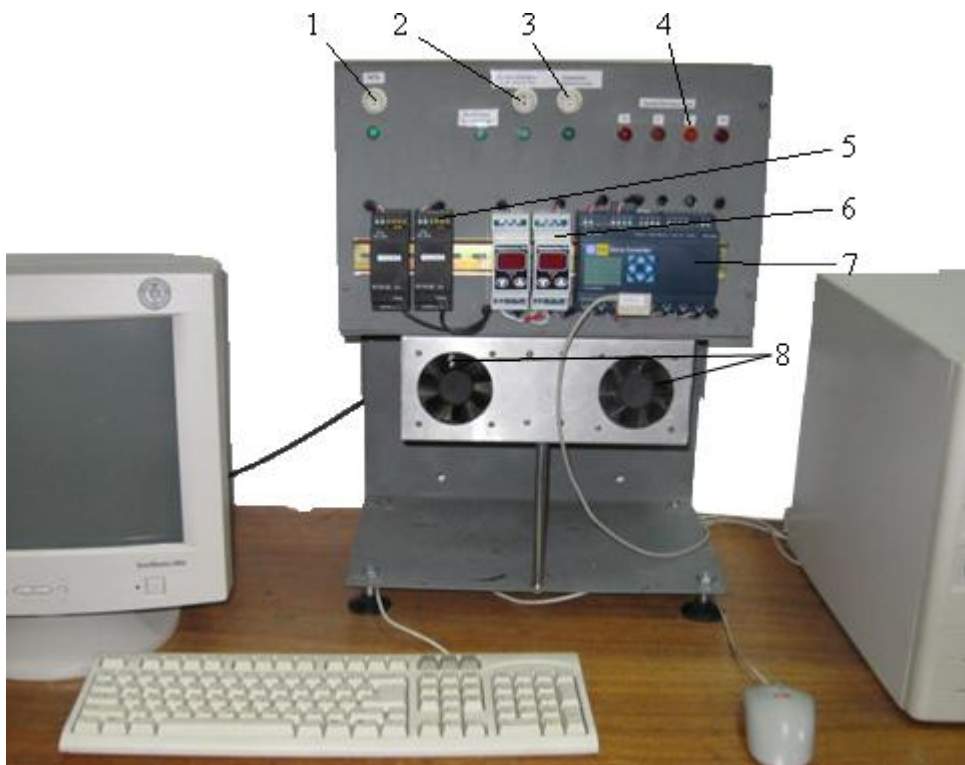
## **12.8 Лабораторна робота №8**

Програмування і дослідження роботи контролера Durus.

**Мета:** ознайомитися с будовою ПЛК Durus, придбати навички програмування і роботи з контролером.

Лабораторна установка, рис.12.16, призначена для програмування і дослідження роботи контролера Durus виробництва GE Fanuc.

Контролер Durus забезпечує трипозиційне регулювання температури в печі, значення якої вимірюють два позиційні датчики ТК-3. При досягненні температурою 30°C спрацьовує перший датчик, сигнал з якого передається на дискретний вхід контролера. Відповідно до програми користувача контролер вмикає вентилятор. Якщо температура продовжує підвищуватися і досягає значення 35°C спрацьовує другий датчик, по сигналу з якого контролер вмикає додатковий другий вентилятор. Одночасно з цим починає працювати програма керування послідовним вмиканням чотирьох ламп із затримкою 6с [66].



1- кнопка подачі живлення на лабораторну установку, 2- кнопка подачі напруги на нагрівальну піч, 3- кнопка для зміни потужності печі, 4-світлодіоди біжучого вогню, 5- джерело живлення, 6- датчики температури, 7 контролер, 8- вентилятори.

Рисунок 12.16 – Лабораторна установка

### ***Контролер DURUS***

DURUS - контролер виробництва GE Fanuc, який розрахований на нескладні завдання логічного управління, рис.12.17, має розміри 72x90x59,6 мм і встановлюється на DIN-рейку. Робочий температурний діапазон – в межах 0-55 градусів за Цельсієм.



Рисунок 12.17 – Контролер DURUS

DURUS є представником мікроконтролерів з вбудованим операторським інтерфейсом, комунікаційними опціями і різними комбінаціями каналів вводу-

виводу. Залежно від вибору моделі кількість входів-виходів може бути 10, 12, 20, 24. Є можливість підключення 3-х модулів розширення (доступні 3 моделі дискретного вводу-виводу і 1 аналогового вводу). З урахуванням модулів розширення загальна кількість зовнішніх сигналів може досягати 44-х.

Дискретні входи розраховані на 12 або 24В постійного, а також на діапазон 100- 240В змінного струму. Аналогові входи призначені для вводу сигналів напруги 0-10В і постійного струму 4-20 мА (модулі розширення). Залежно від моделі доступні релейні виходи до 8А, транзисторні виходи 24В/1А, лічильники імпульсів до 1кГц і виходи ШІМ.

Напруга живлення контролера залежить від вибраної моделі:

- моделі на постійний струм мають напругу 12В, від 20,4 до 28,8 або від 85 до 265 VDC;

- моделі на змінний струм - 24В або від 85 до 265 VAC.

У лабораторній роботі використовується малогабаритний контролер DURUS модифікації IC210MDR024, в якому є 12 дискретних входів на 24В постійного струму, 4 аналогових входа напругою 0-10В постійного струму та 8 релейних виходів на струм до 8А.

Контролер має операторський інтерфейс – LCD-дисплей на 4 рядки по 12 символів і 8 клавіш для програмування.

Пам'ять контролера здатна зберігати до 17 призначених для користувача екранів. Є вбудований годинник реального часу. Контролер здатний інтегруватися в розподілені системи за рахунок використання комунікаційних модулів розширення Modbus RTU Slave, Profibus\_DP або DeviceNet[67,68].

Програмується пристрій інженерними мовами релейноконтактних схем (Ladder) і функціональних блоків (Function Blocks). Спільний розмір призначеної для користувача програми у вбудованій флеш-пам'яті контролера може досягати 200 ланок LD або 99 блоків FBD, а швидкість її виконання складає 10мс на цикл.

Максимальна кількість таймерів, лічильників і компараторів у програмі обмежена 15 шт. Діапазон підрахунку часу від 0,01 секунди і до 9999хв. Програмування може здійснюватися як на комп'ютері, за допомогою програмного середовища Durus Development Software, так і за допомогою LCD дисплея. У контекстному меню програми Durus Development Software присутня кнопка допомоги, що значно спрощує роботу. Після вибору контролера у верхній частці контекстного меню з'являються кнопки. Серед них, окрім стандартних кнопок (Run, Stop, Quit, Save, Print і тому подібне), є кнопки симуляції і кнопка "Key Pad". Перша дозволяє відладити програму, що створена інженерними мовами, навіть не маючи самого контролера, а друга при цьому повністю емулює поведінку LCD дисплея контролера, у тому числі при маніпуляціях з кнопками управління. При роботі з мовою Ladder доступні не лише стандартні булеві операції, але і виклик функціональних блоків, які в тексті програми показуються у вигляді "котушок".

Бібліотека містить 7 різних типів таймерів (з окремим входом скидання і без, із затримкою по вмиканню і вимиканню та ін.), 8 типів лічильників

(інкрементний, декрементний, із збереженням та ін.), 5 типів компараторів (більше, менше, діапазон та ін.). Функції інформаційного обміну дозволяють з'єднуватися з 8-ма іншими контролерами і пересилати їм дані по мережі RS-485.

### ***Програмування Digus з лицевої панелі контролера***

Увімкнути напругу на лабораторному стенді. Для входу в головне меню, що на цифровому дисплеї контролера, натиснути кнопку ESC. Користуючись кнопками ↑ або ↓, вибрати INITIAL і натиснути ОК. У підменю, що відкриється, вибрати мову LADDER і підтвердити кнопкою ОК. Очистити пам'ять контролера. Для цього в головному меню перевести курсор на CLEAR PROG., натиснути послідовно ОК і YES.

Щоб створити перший сегмент програми, який за сигналами з позиційних терморегуляторів вмикає в роботу вентилятор, необхідно встановити курсор у крайнє ліве положення і натиснути кнопку SEL. З'явиться символічне ім'я I1 прямого контакту. Натиснути ОК, змінна I1 зафіксується, а курсор зміститься праворуч. Натиснути SEL, на дисплеї з'явиться другий прямий контакт з таким самим символічним ім'ям I1. Щоб з прямого отримати інвертований контакт з символічним ім'ям i2, необхідно залишити курсор на другому контакті I1 і двічі натиснути кнопку SEL. Велика літера «I» заміниться на прописну «i». Перевести курсор на індекс «1» і кнопкою ↑ збільшити до 2, натиснути ОК. Тут стан змінних I1 та i2 визначається вихідними сигналами терморегуляторів.

Щоб створити лінію сполучення тричі натиснути кнопку SEL. Перевести курсор у крайнє праве положення і натиснути кнопку SEL. З'явиться пряма котушка Q1. Оскільки контролер здійснює керування вентилятором з другого дискретного виходу, адресу прямої котушки треба змінити на Q2. Для цього курсор перевести на індекс біля літери Q і кнопкою ↑ збільшити його до 2.

Перевести курсор на нову строку і за попередньою аналогією створити послідовність контактів I2-i1. Щоб з'єднати паралельно перший і другий сегмент програми, поставити курсор за контактом i1 і натиснути кнопку SEL.

Оскільки лампи біжучого вогню спочатку роботи контролера мають бути вимкнуті, необхідно перевести курсор у праве крайнє положення і натиснути кнопку SEL, з'явиться пряма котушка Q1. Щоб надати їй статус Reset-котушки необхідно курсор перевести на дужку, що перед літерою Q, і вибрати кнопкою ↑ або ↓ ідентифікатор Reset-котушки – стрілку ↓. Для надання цій котушці адреси виходу контролера, що керує роботою першої лампи, перевести курсор на індекс біля літери Q і кнопкою ↑ збільшити його до 5. Для паралельного підключення Reset-котушки Q5 до котушки Q2 курсор установити попереду Set-котушки Q5 і натиснути кнопку SEL. Так само запрограмувати Reset-котушки Q6 - Q8, які керують роботою решти ламп.

Програма, яка забезпечує автоматичне регулювання температури в печі та керування послідовним вмиканням ламп має наступний вигляд:



Користуючись наведеною програмою і навичками отриманими при створенні перших сегментів, завершити самостійно створення програмного коду.

Для програмування таймера затримки вмикання лампи, курсор перевести у крайнє праве положення і натиснути кнопку SEL. З'явиться пряма котушка Q1. Встановити курсор на літеру Q і кнопкою ↑ або ↓ вибрати літеру T, натиснути ОК. З'явиться меню настройки таймера.

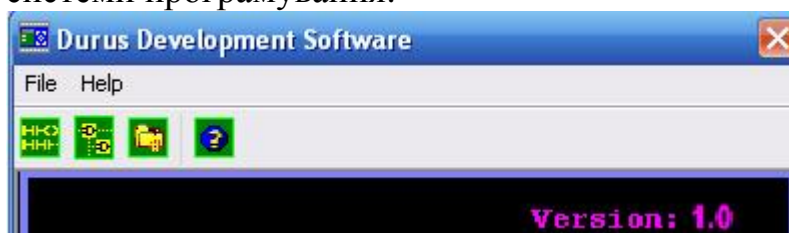
Для створення затримки при вмиканні лампи на бс, курсор перевести на друге знакомісце і натиснути кнопку SEL. Кнопкою ↑ встановити час затримки б і натиснути ОК. Для виходу з меню таймера, натиснути ESC.

Щоб запустити контролер, кнопкою ESC увійти в головне меню, вибрати альтернативу RUN і натиснути ОК. У підменю RUN PROG курсор перевести на YES і підтвердити вибір кнопкою ОК. Контролер почне працювати відповідно до створеної програми.

Для припинення роботи контролера у головному меню скористатися альтернативою STOP.

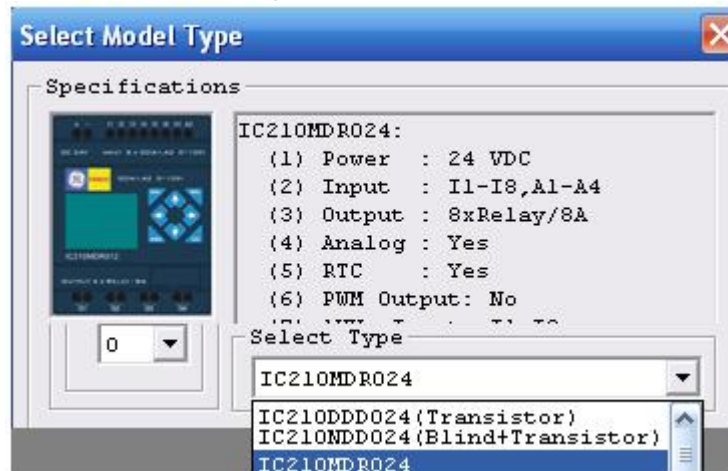
### ***Програмування контролера за допомогою інструментальної системи програмування Durus Development Software***

Запустити програму, користуючись її ярликом на моніторі EOM. З'явиться вікно системи програмування:






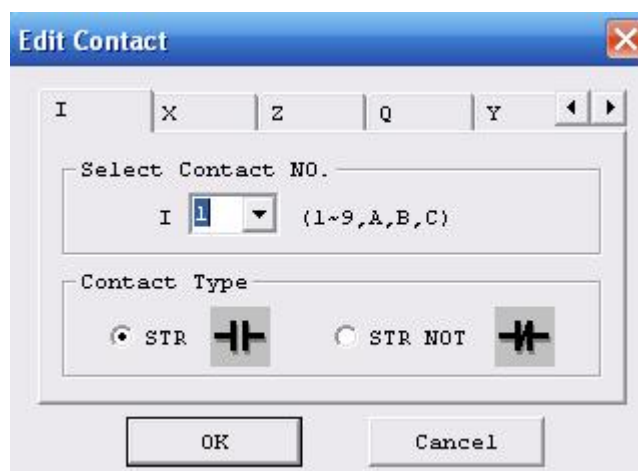
Увійти в меню File→New→NewLAD, щоб створити проект LAD-мовою. З'явиться екран LAD.Version:1.0 з панеллю інструментів. Лівою клавiшею миші натиснути на кнопку New і у вікні Select Model Type, що з'явиться, вибрати тип контролера IC210MDR024, натиснути OK:




Відкриється робоче поле для створення програмного коду. Поміж двома вертикальними шинами живлення знаходиться редакційна сітка, а у нижній частині екрану - панель інструментів, за допомогою якої у клітках редакційної сітки встановлюються мовні елементи:

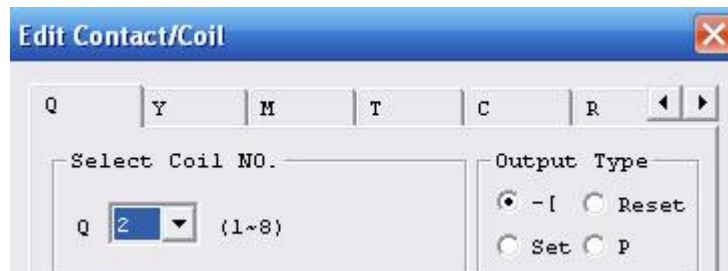



Для створення першого сегменту програми, який забезпечить вмикання першого вентилятора, в панелі інструментів вибрати кнопку I-контакту  і натиснути ліву клавiшу миші. Перевести курсор у ліву крайню клітинку першого сегменту програми і повторно клацнути. У вікні Edit Contact, що з'явиться, призначити контакту перший номер і вибрати тип STR (прямий контакт):



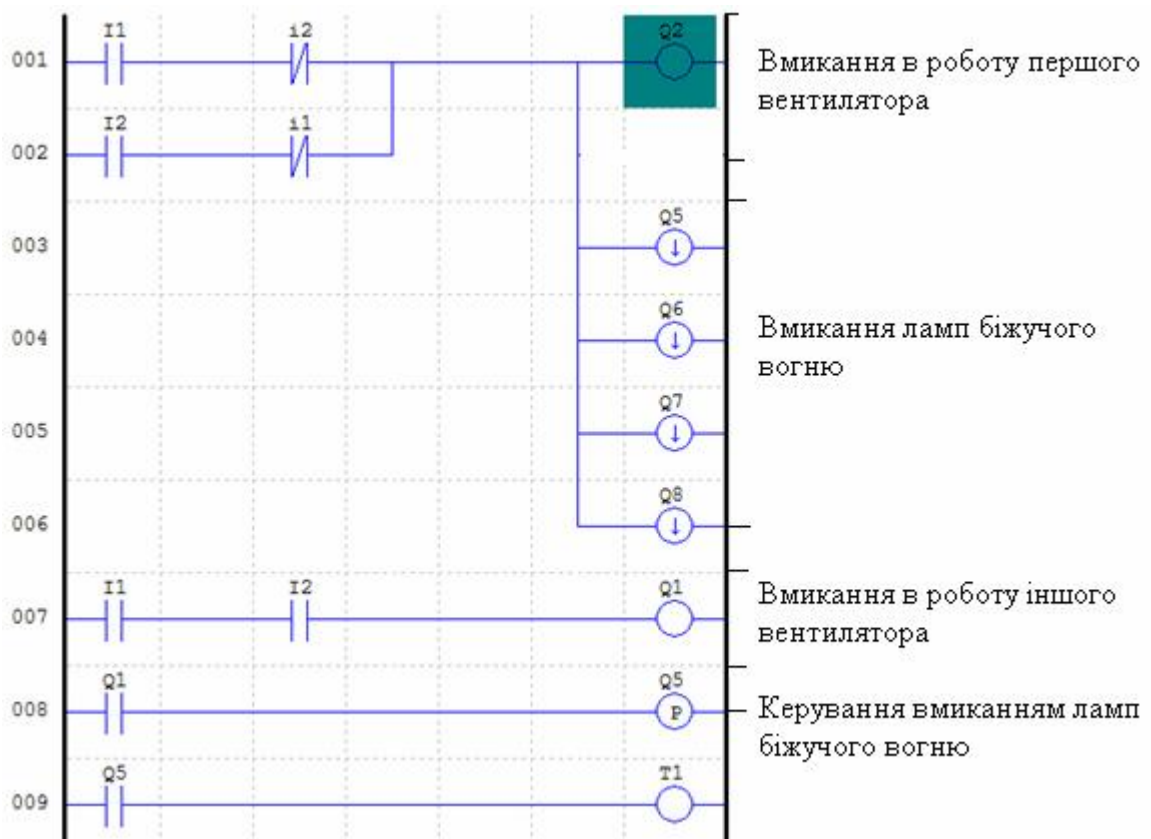
Натиснути OK і контакт I1 з'явиться у першому сегменті, а зелений квадрат зміститься праворуч. Щоб послідовно за прямим контактом установити інверсний, необхідно клацнути лівою клавiшею миші по зеленому квадрату. У вікні Edit Contact призначити змінній індекс 2, вибрати тип контакту STR NOT

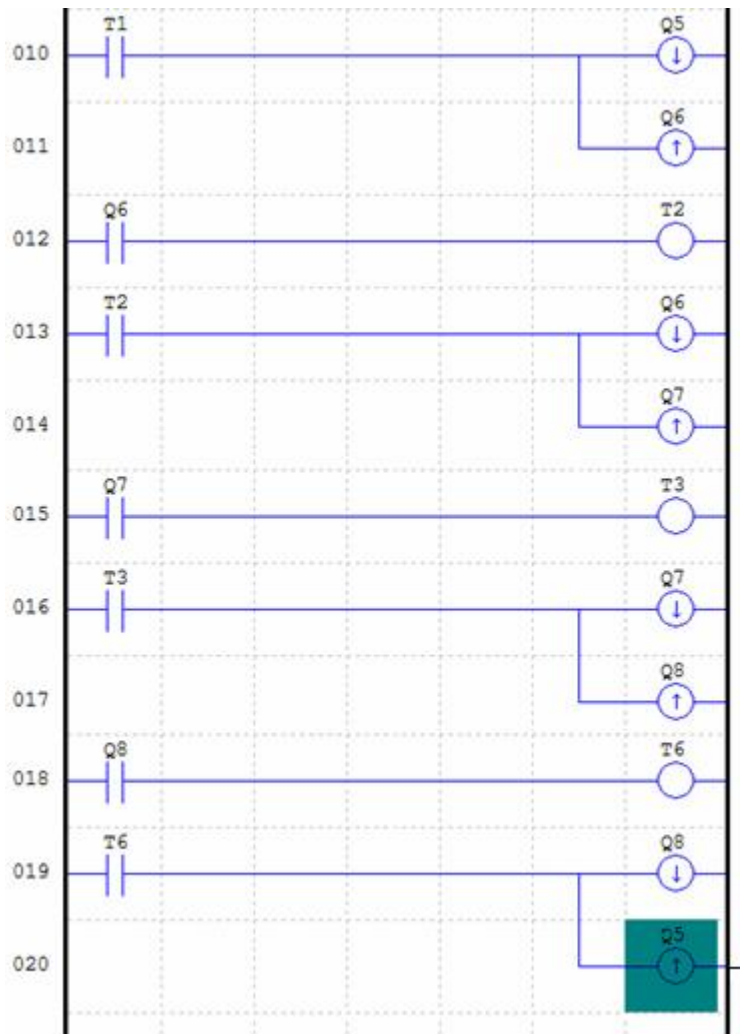
(інвертований контакт) і натиснути ОК. За допомогою кнопки  у панелі інструментів продовжити лінію сполучення до передостанньої клітинки редакційної сітки, а в останню встановити пряму котушку Q, позначивши у вікні Edit Contact/Coil індекс змінної -2 і її тип ”-( ”:



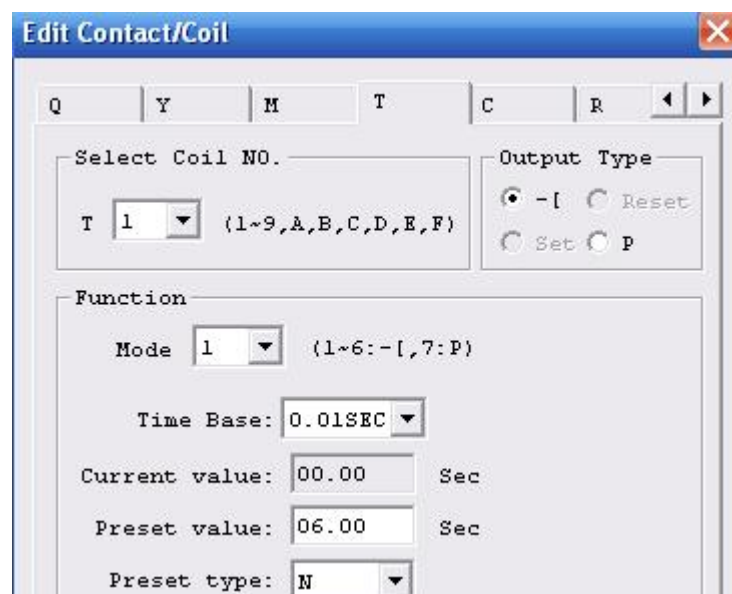
Створити таку саму послідовність контактів у другому сегменті програми тільки прямому контакту призначити символічне ім'я I2, а інвертованому – i1. За допомогою кнопки  у панелі інструментів приєднати праву сполучну лінію інвертованого контакту i1 до першого сегменту програми, забезпечивши таким чином паралельне сполучення контактів першого і другого сегментів. У сегментах програми 3-6 паралельно котушці Q2 встановити Reset-котушки Q5, Q6, Q7, Q8, які вимикають лампи.

Завершити створення програми, користуючись наведеним кодом:




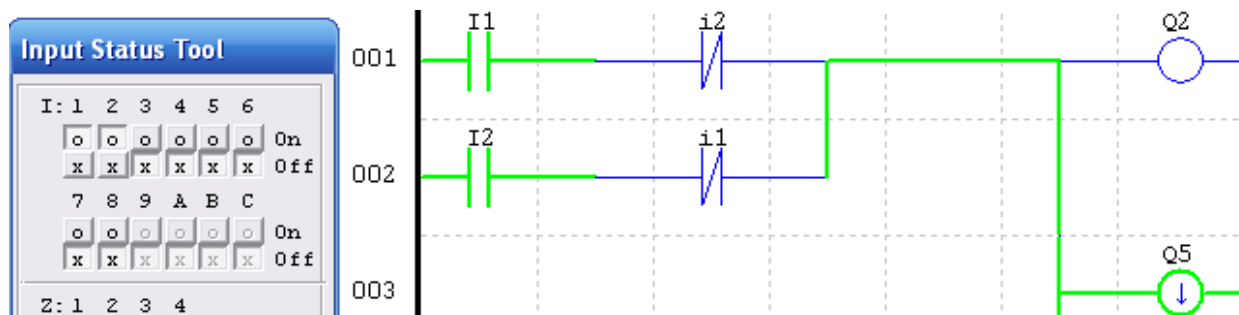


При установленні таймера скористатися в панелі інструментів кнопкою . Відкриється діалогове вікно Edit Contact/Coil, в якому призначити відповідний індекс для таймера T і час затримки Preset value: 06.00sec:




Решту параметрів залишити не змінними.

Щоб перевірити роботу програми, увійти в меню проекту Operation і активізувати Simulator. В панелі інструментів натиснути лівою клав'яшею миші кнопку . З'явиться симулятор вхідних каналів Input Status Tool і програмний код у різнокольоровому вигляді:



Перемикаючи лівою клав'яшею миші в емуляторі стан змінних I1 та I2 з положення Off у положення On, можна переглядати роботу програми. При цьому елементи програми, які набувають стан TRUE, фарбуються у зелений колір.

Коли програма відкоригована, необхідно зв'язатися з контролером. Відкрити меню Operation, активізувати альтернативу Stop і натиснути ліву клав'яшу миші на Link Com Port. У вікні данного пункту вибрати COM1 PORT і натиснути клав'яшу Link. Для завантаження програми у контролер, увійти в меню Operation → Write, а щоб запустити її необхідно натиснути лівою клав'яшею миші кнопку  в панелі інструментів[69].

### ***Завдання для самостійної роботи***

Модифікувати код програми регулювання температури печі так, щоб кількість спрацьовувань датчиків температури підраховував лічильник.

### ***Зміст звіту***

Звіт має містити назву роботи, короткі відомості про контролер, програму регулювання температури печі, результати самостійної роботи і висновки.

### ***Контрольні запитання***

1. Яка конструкція контролера?
2. Якими мовами програмується контролер DURUS?
3. Яке призначення мають клав'яші на лицевій панелі DURUS?
4. Як здійснюється програмування контролера з EOM?
5. Що містить бібліотека контролера?
6. Які можливості має пакет програмування Durus Development Software?
7. Для чого призначений годинник реального часу?
8. Який розмір може мати програма?
9. Як програмується таймер?
10. Як здійснюється завантаження програми у контролер?

## 13 КУРСОВА РОБОТА

**Мета.** Отримання знань і придбання навичок необхідних для створення мікропроцесорних систем автоматичного управління.

У процесі виконання курсової роботи, студент, відповідно до завдання, має зрозуміти технологічний процес і принципи управління його основними параметрами. На підставі отриманого розуміння, скласти схему інформаційних потоків, підібрати необхідні датчики контрольованих параметрів і виконавчі пристрої, якими керує контролер. Визначити кількість та тип вхідних і вихідних сигналів мікропроцесорного контролера. Здійснити проектне компонування мікропроцесорного контролера, розробити для нього програмний додаток користувача, перевірити і описати роботу створеної програми.

### ***Зміст курсової роботи.***

1. Завдання до курсового проекту.
2. Технологічний агрегат або процес, як об'єкт автоматизації.
3. Принципи управління (регулювання) технологічними параметрами.
4. Схема інформаційних потоків.
5. Визначення кількості та типу вхідних і вихідних сигналів мікропроцесорного контролера.
6. Проектне компонування контролера.
7. Розробка проекту, створення програмного коду і опис його роботи.
8. Висновки.
9. Використана література.

### **13.1 Перелік завдань для курсової роботи**

1. Здійснити проектне компонування контролера з децентралізованою переферією Simatic S7-300, який працює у мережі PROFIBUS, для АСУТП спікального відділення аглофабрики і створити в інструментальному середовищі Step-7 FBD-програму САР процесу спікання.

2. Здійснити проектне компонування контролера виробництва WAGO, який працює у мережі Ethernet, для системи автоматичного управління роботою грейферного крану на рудному дворі і в редакторах FBD та візуалізації інструментальної системи програмування CoDeSys створити відповідну програму та людино-машинний інтерфейс.

3. Здійснити проектне компонування контролера ADAM 5510KW для системи автоматичного управління роботою пневмотранспорту псевдоожиженого глинозему і розробити відповідну LD-програму у середовищі MULTIPROG.

4. Здійснити проектне компонування контролера OMRON SYSMAC CS1, який працює у мережі Controller Link, для АСУТП відділення змішування бетонного заводу і створити відповідний LD-додаток в інструментальному середовищі CX-Programmer.

5. Здійснити проектне компонування контролера Simatic S7-300 розширеної резервованої конфігурації, який працює у мережі PROFIBUS-DP, для системи автоматичного управління роботою преса відділення штамповки автозаводу і відповідну FBD-програму в інструментальному середовищі Step-7.

6. Здійснити проектне компонування дубльованого контролера ADAM 5510KW для системи автоматичного управління роботою роздавального бункера бетономішалки і відповідну FBD-програму у середовищі MULTIPROG.

7. Здійснити проектне компонування контролера Modicon TSX Quantum віддаленої конфігурації для АСУТП шихтового відділення та відповідний LD-додаток у середовищі Concept.

8. Здійснити проектне компонування контролера Реміконт P-130Isa, який працює у мережі Modbus, для системи автоматичного управління роботою ковпакової печі та створити відповідну FBD-програму у пакеті IsaGraf.

9. Здійснити проектне компонування контролера виробництва WAGO, який працює у мережі DeviceNET, для АСУТП дозувального відділення бетонозмішувального цеху і в інструментальній системі програмування CoDeSys створити відповідну LD-програму та людино-машинний інтерфейс.

10. Здійснити проектне компонування контролера ADAM 5510KW для системи автоматичного управління роботою пневмотранспорту твердого глинозему і відповідну LD -програму у середовищі KW MULTIPROG.

11. Здійснити проектне компонування контролера МІК-51 для системи автоматичного управління процесом тонкого помелу руди і відповідну FBD-програму у середовищі «Альфа».

12. Здійснити проектне компонування контролера розширеної конфігурації Simatic S7-300, який працює у мережі AS-Interface, для АСУТП парового котла і створити в інструментальному середовищі Step-7 FBD-програму САР навантаження котла.

13. Здійснити проектне компонування контролера МІК-51 для системи автоматичного управління процесом плавлення у дуговій печі та відповідну FBD-програму у середовищі «Альфа».

14. Здійснити проектне компонування контролера Реміконт P-130 для системи автоматичного управління роботою водогрійного котла і створити відповідну програму.

15. Здійснити проектне компонування контролера OMRON SYSMAC CS1, який працює у мережі Ethernet, для АСУ ТП регенеративного нагрівального колодязя і створити відповідну LD-програму у пакеті Cx-Programmer.

16. Здійснити проектне компонування контролера гарячого резервування Modicon TSX Quantum, який працює у мережі Modbus Plus АСУ ТП методичної печі та створити відповідну FBD-програму у пакеті Unity Pro.

17. Здійснити проектне компонування контролера ControlLogix розподіленої конфігурації, який працює у мережі ControlNet, для АСУ ТП доменної печі та створити у пакеті RSLogix 5000 LD-програму керування процесом завантаження шихти.

18. Здійснити проектне компонування контролера виробництва WAGO, який працює у мережі Profibus, для АСУ ТП камерної печі та створити відповідну програму керування у пакеті CoDeSys.

19. Здійснити проектне компонування контролера гарячого резервування PACSystemsRX7i розподіленої конфігурації для АСУ ТП синтезу продукту у реакторі та створити LD- програму керування у пакеті Proficy Machine Edition.

20. Здійснити проектне компонування контролера ControlLogix віддаленої конфігурації, який працює у мережі DH+, для АСУ ТП поточно-транспортної системи спікального відділення аглофабрики та створити відповідну LD-програму у пакеті RSLogix 5000.

21. Здійснити проектне компонування контролера OMRON SYSMAC CS1 розширеної конфігурації, який працює у мережі Controller Link, для АСУ ТП рекуперативного нагрівального колодязя і створити відповідну LD-програму у пакеті Cx-Programmer.

22. Здійснити проектне компонування контролера Modicon TSX Quantum розподіленої конфігурації для АСУТП повертальної вельц печі та LD-додаток автоматичного регулювання температури відхідних газів в інструментальному середовищі Concept.

23. Здійснити проектне компонування контролера гарячого резервування PACSystemsRX7i віддаленої конфігурації для АСУ ТП сталеплавильного конвертера та створити відповідну програму керування у пакеті Proficy Machine Edition.

24. Здійснити проектне компонування контролера ADAM 5510KW для системи автоматичного управління процесом термічної обробки деталей і відповідну FBD-програму у середовищі KW MULTIPROG.

25. Здійснити проектне компонування контролера МІК-51 для системи централізованого автоматичного контролю і управління трьома промисловими воротами і відповідну FBD-програму у середовищі «Альфа».

## **13.2 Приклади виконання курсової роботи**

### **13.2.1 Приклад 1**

**Завдання.** Зрозуміти технологію і принципи управління роботою багатопаливного парового котла. Скласти схему інформаційних потоків, виконати підбір необхідних датчиків і виконавчих пристроїв, здійснити проектне компонування контролера Modicon TSX Quantum для розподіленої АСУ ТП, розробити у середовищі Concept FBD-мовою програмний додаток до системи автоматичного регулювання температури перегрітої пари, виконати перевірку роботи програми і описати її роботу.

#### ***Паровий котлоагрегат, як об'єкт автоматизації***

Об'єктом автоматизації є багатопаливний котел призначений для вироблення перегрітої пари. Працює на трьох видах палива – доменний, коксовий і природний газ. Парова продуктивність котла при роботі на



доменному газі з підсвічуванням коксовим або природним газом становить 120 т/год, а на природному газі – 150 т/год. Пара має температуру 400°C і тиск 3,14 МПа. Основні види палива - доменний та коксовий газ. Резервний вид палива – природний газ.

Паровий котел складається з одного корпусу, виконаного за П-образним компонованням, і симетричним розташуванням поверхонь нагрівання.

Живильна вода надходить у котел з магістралі під тиском 4,3 МПа при температурі 103 °С. Проходить через вузол живлення і попадає у водяний економайзер, з якого підігріта вода надходить у барабан.

Котел обладнано одним барабаном і чотирма виносними циклонами. У котлі застосована природна циркуляція води та двохступінчаста схема випарювання. В процесі пароутворення рівень води в барабані підтримується в заданих межах для запобігання влучення води в парову магістраль котельні при високому рівні або порушення циркуляції в екранних трубах, при низькому рівні.

Котел обладнаний барабаном, має контури водяного і парового тракту.

Оскільки попередня підготовка живильної води не виключає присутності в ній солей, механічних домішок і газів, то в процесі випарювання води відбувається нагромадження солей у барабані котла. Для видалення солей і шламу з водяного тракту передбачається безперервна продувка, яка здійснюється шляхом регулювання витрати продувочної води за інформацією солеміра. З барабана котла пара надходить у систему пароперегрівника, який складається з двох ступеней. Для регулювання температури пари в пароперегрівнику використовується впорскувальний пароохолодник, який розташований поміж ступенями пароперегрівника. Конденсат на охолодження пари, надходить через регулювальний клапан. Після пароперегрівника пар надходить у парозборну камеру, з якої по двом однаковим паропроводам подається в магістраль котельній.

Паровий котел працює із зрівноваженою тягою. Тягодуттєва установка котла складається із двох димососів і двох дуттєвих вентиляторів.

Повітря для спалювання палива, подається в топку двома дуттєвими вентиляторами. Проходячи через рекуперативний трубчастий повітряпідігрівник, повітря надходить до двох пальників котла. Подача повітря в пальники здійснюється через верхні сопла (разом з доменним і коксовим газом) і нижні сопла (разом із природним газом).

У топці продукти згоряння надходять у конвективний газохід, де послідовно, по ходу газів, розташовуються конвективний перегрівник із вбудованим пароохолодником, випарна поверхня і водяний економайзер. Трубчастий повітропідігрівник і економайзер розташовані у винесеному газоході. Для створення розрідження і видалення димових газів з топки в димар, використовуються два димососи. Регулювання подачі повітря тягодуттєвими машинами здійснюється напрямними апаратами з поворотними лопатками, які установлені на вході робочого колеса.

Для спалювання доменного, природного і коксового газу, котел обладнаний двома пальниками, установленими на бічних стінах. Щоб отримати стабільний факел при спалюванні доменного газу, він підсвічується коксовим і природним газами, які подаються по окремих каналах в пальник. Витрата цих газів складає 7 - 10 % від загального тепловиділення.

### ***Управління роботою парового котла***

Управління роботою парового котла здійснюється за допомогою систем автоматичного регулювання температури пари, розрідження у топці котла, теплового навантаження котла, якості спалювання палива, рівня води в барабані-сепараторі та продувки барабана котла.

Регулювання температури пари здійснюється за сигналом з датчика температури, встановленого на паропроводі після другого ступеня пароперегрівника. Якщо цей сигнал не відповідає заданому значенню, регулятор збільшує або зменшує подачу охолодженої води в пароохолодник на впорскування. Через значне транспортне запізнення, регулювання температури за такою схемою, не забезпечує заданої якості. Тому в схемі регулювання використовується додатковий сигнал по швидкості зміни температури за пароохолодником. Сигнал формується за допомогою диференціатора і подається на інший вхід регулятора.

Показником балансу між кількістю газів, що утворюються у топці котла при спалюванні палива, і кількістю газів, що видаляються з неї димососом, є розрідження в топковому просторі. В усталеному режимі ця величина постійна. Підвищення або зменшення теплового навантаження котла призводить до коливання кількості газів, внаслідок чого величина розрідження також не постійна. Для відновлення рівноваги потрібно коригувати видалення газів димососом. Сигнал, пропорційний розрідженню у топці котла, подається на вхід регулятора, який збільшує або зменшує продуктивність димососу. Для підвищення якості регулювання додатково на вхід регулятора подається сигнал від датчика витрати повітря, який відчуває зміни у процесі горіння палива раніше, ніж виникнуть коливання розрідження у топці котла.

Системи автоматичного регулювання мають підтримувати продуктивність котлоагрегату у строгій відповідності зі споживанням пари.

Показником цієї відповідності є сталість тиску пари в паропроводі між котлом і споживачем, яка є основним збуренням у роботі системи автоматичного регулювання теплового навантаження котлоагрегату.

Щоб забезпечити сталий тиск у паропроводі, тобто мати стабільним процес пароутворення, необхідно при його коливаннях коригувати витрату палива. Проте зміна витрати палива не може відбуватися самостійно, оскільки це порушує умови спалювання палива.

У зв'язку з цим, одночасно зі зміною подачі палива коригується витрата повітря таким чином, щоб забезпечити підтримку заданого коефіцієнта надлишку повітря, при якому ККД котла має найбільше значення. Щоб забезпечити це, сигнали з датчиків витрати палива і повітря надходять на вхід

регулятора, де порівнюються між собою. Якщо не забезпечується необхідне співвідношення, регулятор змінює витрату повітря. Для підвищення якості регулювання, використовується додатковий коригуючий сигнал, пропорційний концентрації кисню у відхідних газах.

У процесі пароутворення в барабані-сепараторі відбувається безперервна циркуляція пароводяної емульсії. При різкому збільшенні споживання пари і незмінній подачі палива в котлоагрегаті знижується тиск і відбувається процес набрякання пароводяної суміші, що призведе до різкого підвищення рівня в барабані. У зв'язку з цим необхідне регулювання рівня води. Здійснюється це шляхом регулювання співвідношення витрат пари і живильної води з корекцією по величині рівня води в барабані котла.

Живильна вода приносить у котел певну кількість розчинених солей. У процесі пароутворення ці солі залишаються в котловій воді і згодом їх концентрація досягає неприпустимої величини. Щоб цього не допустити здійснюється продувка барабана котла. По сигналу, пропорційному концентрації солі, коригується постачання продувної води. Оскільки витрата продувної води має бути пропорційною паропродуктивності котла, то схема доповнюється коригувальним сигналом, що відповідає витраті пари.

### *Схема інформаційних потоків*

Щоб визначити всі входні і вихідні сигнали мікропроцесорної системи управління і здійснити проектне компонування контролера Modicon TSX Quantum для розподіленої АСУ ТП, необхідно на основі розглянутих алгоритмів автоматичного регулювання технологічних параметрів багатопаливного парового котла скласти схему інформаційних потоків, рис.13.1.

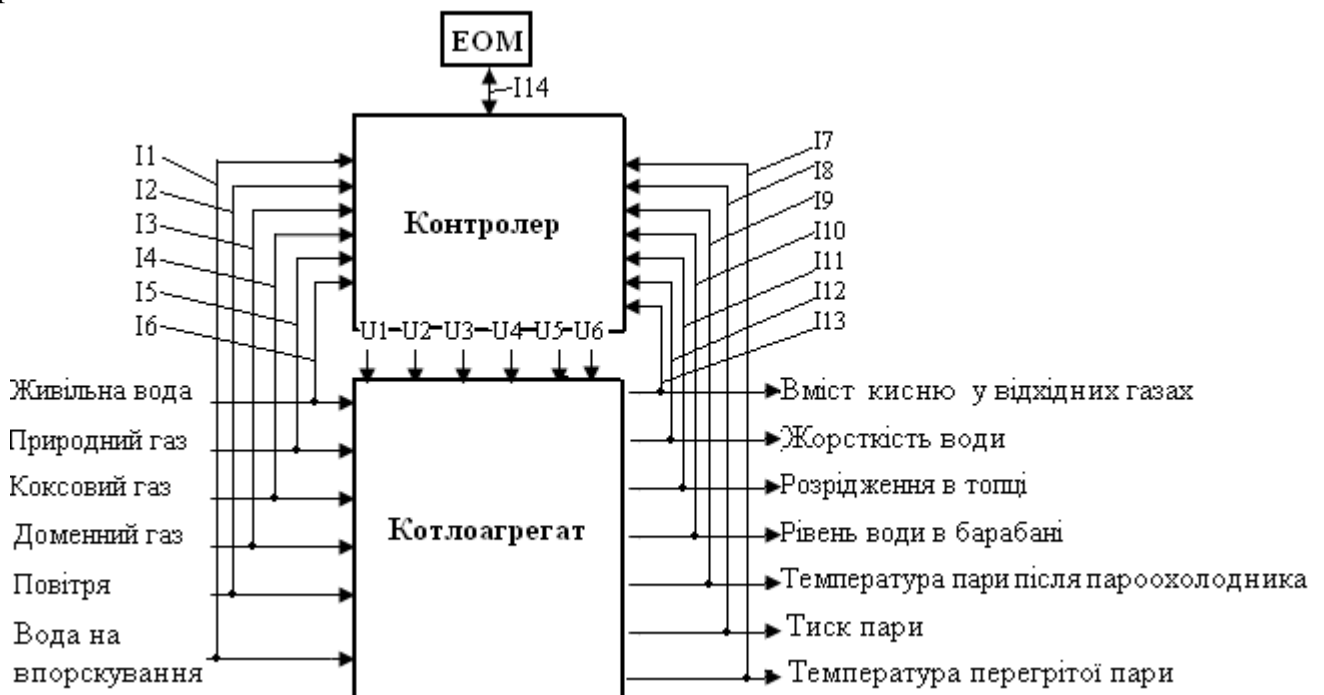


Рисунок 13.1 – Схема інформаційних потоків.

Потоки інформаційних сигналів:

- I1 – витрата води на впорскування; I8 – тиск пари;  
 I2 – витрата повітря; I9 – температура пари після пароохолодника;  
 I3 – витрата доменного газу; I10 – рівень води в барабані котла;  
 I4 – витрата коксового газу; I11 – розрідження у топці котла;  
 I5 – витрата природного газу; I12 – жорсткість води;  
 I6 – витрата живильної води; I13 – вміст кисню у відхідних газах;  
 I7 – температура перегрітої пари; I14 – обмін інформацією поміж контролером і ЕОМ.

Потоки керувальних сигналів:

- U<sub>1</sub> – вплив на витрату води в барабан котла;  
 U<sub>2</sub> – вплив на витрату природного газу в топку котла;  
 U<sub>3</sub> – вплив на витрату повітря для спалювання газу;  
 U<sub>4</sub> – вплив на витрату коксового газу в топку котла;  
 U<sub>5</sub> – вплив на витрату доменного газу в топку котла;  
 U<sub>6</sub> – вплив на витрату води на впорскування.

**Визначення кількості та типу вхідних і вихідних сигналів  
 мікропроцесорного контролера**

Визначення кількості та типу вхідних і вихідних сигналів мікропроцесорного контролера здійснюється на основі аналізу вихідних сигналів датчиків, що контролюють технологічні параметри і методів управління виконавчими механізмами, які отримують керувальний вплив від мікропроцесорного контролера.

Для реалізації розподіленої системи автоматичного управління роботою багатопаливного парового котла, необхідно здійснювати безперервний контроль технологічних параметрів, характеристика яких наведена у табл.13.1.

Таблиця 13.1

Технологічні параметри

№ п/п	Параметр	Одиниці ви- мірювання	min	Номінальна величина	max
1	Продуктивність	т/год.	120	130	150
2	Температура перегрітої пари	°С	390	400	420
3	Тиск у барабані котла	МПа	3,0	3,14	3,2
4	Температура пари після пароохолодника	°С	380	390	410
5	Витрата природного газу	нм <sup>3</sup> /год.	0	360	16000
6	Витрата доменного газу	нм <sup>3</sup> /год.	0	53000	160000
7	Витрата коксового газу	нм <sup>3</sup> /год.	0	2450	13000
8	Витрата повітря	нм <sup>3</sup> /год.	0	15000	56000
9	Жорсткість води	мг/кг	1500	2400	3000
10	Вміст кисню у відхідних газах	%	1,33	11,0	21,0
11	Розрідження у топці	мм.вд.ст.	-0,125	0,1	+0,125
12	Рівень води у барабані	мм	-315	0	+315
13	Витрата живильної води	нм <sup>3</sup> /год.	0	120	160
14	Витрата води на впорскування	нм <sup>3</sup> /год.	0	10	30

Керуючись даними, що наведені у табл.13.1, і каталогами провідних виробників технічних засобів автоматизації – «ЕЛЕМЕР», «ПЭК», «СВ АЛЬТЕРА», підібрані необхідні датчики та виконавчі механізми і визначені їх вихідні та вхідні сигнали, табл.13.2 і 13.3.

Таблиця 13.2

Датчики контролю технологічних параметрів

№ п/п	Параметр	Назва і тип датчика або виконавчого механізму	Вихідний сигнал датчика	Постачальник технічного засобу
1	Температура перегрітої пари	Термоперетворювач з уніфікованим вихідним сигналом ТСПУ-055	4-20мА	ЕЛЕМЕР м.Москва
2	Тиск у барабані котла	Інтелектуальний перетворювач надмірного тиску АІР-30-ТГ14	4-20мА	ЕЛЕМЕР м.Москва
3	Температура пари після пароохолодника	Термоперетворювач з уніфікованим вихідним сигналом ТСПУ-055	4-20мА	ЕЛЕМЕР м.Москва
4	Витрата природного газу	Перетворювач диференційного тиску АІР-30-СД13	4-20мА	ЕЛЕМЕР м.Москва
5	Витрата доменного газу	Перетворювач диференційного тиску АІР-30-СД15	4-20мА	ЕЛЕМЕР м.Москва
6	Вміст О <sub>2</sub> у відхідних газах	Газоаналізатор АКВТ-01	4-20мА	ЕЛЕМЕР м.Москва
7	Температура відхідних газів	Термоперетворювач з уніфікованим вихідним сигналом ТХКУ-205	4-20мА	ЕЛЕМЕР м.Москва
8	Витрата коксового газу	Перетворювач диференційного тиску АІР-30-СД15	4-20мА	ЕЛЕМЕР м.Москва
9	Витрата повітря	Перетворювач диференційного тиску АІР-30-СД13	4-20мА	ЕЛЕМЕР м.Москва
10	Розрідження у топці	Інтелектуальний перетворювач надмірного тиску – розрідження АІР-30-СВ6	4-20мА	ЕЛЕМЕР м.Москва
11	Рівень води у барабані	Інтелектуальний перетворювач гідростатичного тиску АІР-30-СЛ9	4-20мА	ЕЛЕМЕР м.Москва
12	Витрата живильної води	Перетворювач диференційного тиску АІР-30-СД8	4-20мА	ЕЛЕМЕР м.Москва
13	Жорсткість води	Індуктивний концентратомір АСІ-Z	4-20мА	СВ АЛЬТЕРА м.Київ

Таблиця 13. 3

Виконавчі пристрої, що змінюють параметр

№ п/п	Параметр	Назва і тип виконавчого пристрою, що змінює параметр	Сигнал керування виконавчим механізмом	Постачальник технічного засобу
1	Витрата води	Пускач ПБР з виконавчим механізмом МЕО-16/10-0,25-93	Імпульсний	ООО «ПЭК» м. Чебоксари
2	Витрата повітря	Пускач ПБР з виконавчим механізмом МЕО-40/63-0,63-93	Імпульсний	ООО «ПЭК» м. Чебоксари
3	Витрата природного, газу	Пускач ПБР з виконавчим механізмом МЕО-63/10-0,25-92С	Імпульсний	ООО «ПЭК» м. Чебоксари
4	Витрата доменного газу	Пускач ПБР з виконавчим механізмом МЕО-63/10-0,25-92С	Імпульсний	ООО «ПЭК» м. Чебоксари
5	Витрата коксового газу	Пускач ПБР з виконавчим механізмом МЕО-63/10-0,25-92С	Імпульсний	ООО «ПЭК» м. Чебоксари
6	Витрата води на впорскування	Пускач ПБР з виконавчим механізмом МЕО-63/10-0,25-92С	Імпульсний	ООО «ПЭК» м. Чебоксари

### ***Проектне компонування контролера Modicon TX Quantum***

Серія програмованих контролерів Modicon TX Quantum є платформою для розв'язання завдань в області промислової автоматизації. Завдяки модульній архітектурі контролер Modicon TX Quantum може забезпечувати керування як нескладними об'єктами автоматизації, так і найбільш відповідальними технологічними процесами на рівні цілого підприємства.

На базі Modicon TSX Quantum можна реалізувати локальну, віддалену або розподілену конфігурацію вводу-виводу, які компонуються певною комбінацією шасі, центральних процесорів, джерел живлення, модулів вводу-виводу та інших технічних засобів.

Проектне компонування контролера здійснюється на основі інформації про кількість та вид вхідних і вихідних сигналів, з якими працює контролер у процесі автоматичного управління технологічним процесом, бажану конфігурацію вводів-виводів та мережну структуру.

Відповідно табл.13.2, кількість інформаційних сигналів, що надходять від датчиків до контролера дорівнює 13. Усі вони уніфіковані струмові з діапазоном 4-20мА. На такий вхідний сигнал орієнтований 8-канальний аналоговий модуль вводу 140 АСІ 030 00, який споживає струм 240мА і має потужність 2Вт.

Кількість модулів такого типу, необхідних для перетворення 13 вхідних сигналів дорівнює:

$$13 \text{ вхідних сигналів} : 8 \text{ каналів вводу} = 1,63 \approx 2 \text{ модулі } 140 \text{ АСІ } 030 \text{ } 00$$

За результатами обробки вхідних сигналів контролер формує керувальний вплив на витрату води, повітря, природного, доменного та коксового газу. Здійснюється це за допомогою виконавчих механізмів сталої швидкості, які працюють від імпульсних сигналів управління. Оскільки при регулюванні технологічних параметрів виконавчий механізм має здійснювати компенсацію збурень шляхом збільшення або зменшення витрати води, пари, природного, доменного або коксового газу, при розрахунку кількості дискретних модулів виводу необхідно передбачити на кожний виконавчий механізм два виходи – «більше» і «менше». Як показано у табл.13.3, зміну витрат технологічних параметрів здійснюють шість виконавчих механізмів, а це означає, що для керування їх роботою потрібно дванадцять вихідних каналів дискретного модуля виводу.

Серед існуючих дискретних модулів виводу Quantum для керування роботою виконавчих механізмів найбільш підходить модуль 140 DRA 840 00, у якого 16 каналів оснащених вихідними реле з нормально відкритими контактами і який споживає струм 1100мА.

Кількість модулів такого типу, які забезпечать керування роботою виконавчих механізмів складе:

$$12 \text{ вихідних сигналів} : 16 \text{ каналів виводу} = 0,75 \approx 1 \text{ модуль } 140 \text{ DRA } 840 \text{ 00}$$

Для опитування 13 входів, обробки інформації, відповідно до алгоритмів керування окремими технологічними параметрами, оновлення станів 12 вихідних каналів, реалізації одної гілки розподіленого вводу-виводу та здійснення інших процедур у системі управління, можна використати центральний процесор 140CPU 113 03. Він має ОЗП –512 Кбайт, 1 порт Modbus Plus і споживаний струм шини – 790мА, а також підтримує локальний ввід-вивід– до 64(ввід)/64(вивід) і розподілений ввід-вивід – до 30 (ввід)/32 (вивід).

Оскільки, відповідно до завдання, необхідно сконфігурувати розподілений ввід-вивід, для віддаленої панелі потрібен адаптер вузлу DIO 140 CRA211 10, який одночасно використовується як блок живлення вузла. Цей адаптер має порт Modbus Plus для сполучання з центральним процесором, живиться напругою 85-276В змінного струму, має вихідний струм на шину – 3А.

Для функціонування контролера потрібен модуль живлення, який необхідно розрахувати, користуючись підсумковим струмом, що споживають усі модулі контролера. Враховуючі, що один аналоговий модуль вводу розташовується на розподіленому шасі, де в якості модуля живлення використовується адаптер вузла DIO, то для розрахунку споживаного струму потрібно враховувати модулі тільки центрального шасі.

Аналоговий модуль вводу споживає струм 240мА, дискретний модуль виводу – 560мА, центральний процесор – 790мА. Тобто сумарний струм, що споживають модулі центрального шасі дорівнює 1599мА. Щоб забезпечити такий струм споживання необхідно використати модуль живлення 140CPS 111

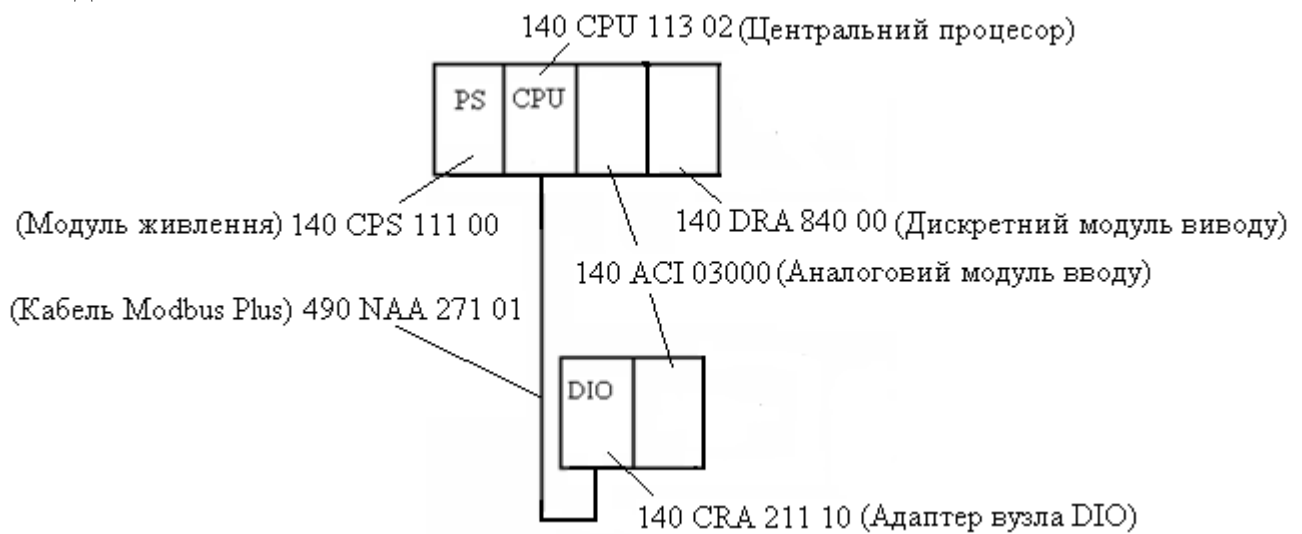


00, який має вхідну напругу 100-276В змінного струму, а вихідний струм шини дорівнює максимум 3А.

Для конфігурування розподіленого вводу-виводу потрібні два шасі. На центральному необхідно встановити модуль живлення, центральний процесор, аналоговий модуль вводу і дискретний модуль виводу, а на виносному – адаптер вузла DIO і аналоговий модуль вводу. У зв'язку з цим у якості центрального шасі необхідно використати 4-слотове 140 XBP 004 00, а для розподіленого вводу застосувати 2-слотове 140 XBP 002 00.

Для з'єднання порту Modbus Plus центрального процесора з адаптером вузла 140 CRA 211 10, який розташований на певній відстані, потрібен кабель 490 NAA 271 01 довжиною 30,5м.

Структурна схема розподіленої системи автоматичного управління має вигляд:



**Розробка проекту і створення програмного коду системи автоматичного регулювання температури перегрітої пари**

Структурна схема системи автоматичного регулювання температури перегрітої пари має наступний вигляд:



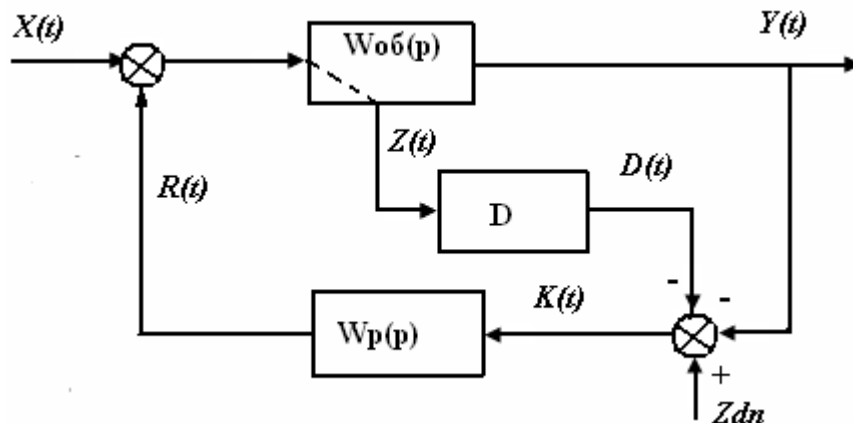
Перегрів пари до необхідної температури відбувається у пароперегрівнику, між першим і другим ступенями якого знаходиться

пароохолодник. Регулювання температури пари здійснюється шляхом зміни подачі охолодної води на впорскування в пароохолодник  $\mu(t)$ .

Основна регульована величина об'єкта - температура перегрітої пари  $y(t)$  вимірюється термопарою  $T\Pi y$ , сигнал від якої подається на регулятор. Крім того в схемі використовується додатковий сигнал  $z(t)$  від термопари  $T\Pi z$ , яка реагує на зміни температури пари, що викликані коливанням температури у першій ступені пароперегрівника або довільною зміною витрати води на впорскування. В результаті регулятор змінює витрату води на впорскування до початку відхилення температури пари на виході з II ступені пароперегрівника.

Оскільки в сталих режимах роботи котла регулятор має контролювати лише температуру на виході пароперегрівника  $y(t)$ , сигнал  $z(t)$  повинен з'являтися тільки у перехідних режимах. Тому перетворювач сигналу  $z(t)$  є диференціатором.

Для створення додатку користувача зобразимо вихідну схему автоматичного регулювання температури пари за пароперегрівником еквівалентною структурою з такими назвами змінних, які будуть використовуватися при створенні програми:



Програмування контролерів Modicon TX Quantum здійснюється за допомогою інструментальної системи програмування Concept.

Основними перевагами цього пакету є:

- легко доступні інтерфейси, досконалі графічні редактори і контекстна допомога, яка спрощує програмування Quantum;
- використання всіх п'яти мов МЕК-стандарту – LD, FBD, SFC, ST і IL;
- наявність бібліотек функціональних блоків за стандартом МЕК, а також групи елементів для регулювання, комунікацій та ін.;
- могутні функції пошуку змінних, помилок, визначення невикористаних змінних;
- можливість створення функціональних блоків, за допомогою мови «С»;
- програмний емулятор роботи контролера;
- можливість локального і віддаленого програмування з використанням інтерфейсів Modbus, Modbus Plus і TCP/IP Ethernet.

Відповідно до наведеної структурної схеми системи автоматичного регулювання температури пари, для розробки програмного коду потрібні такі функціональні блоки:

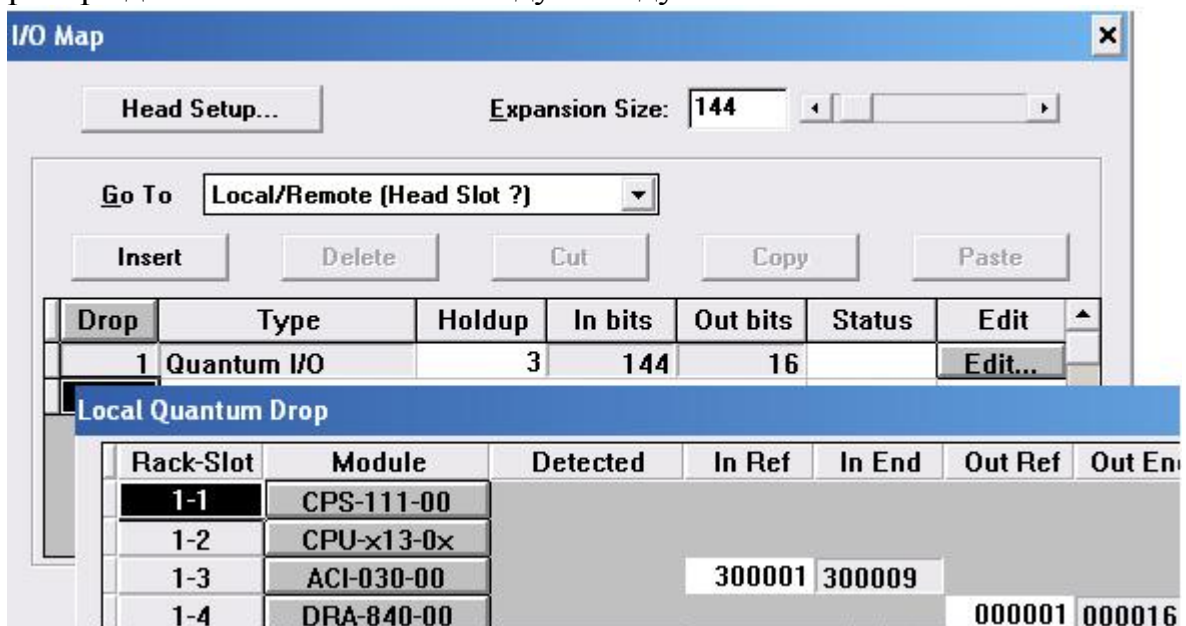
- PI1 (PII-регулятор), PDM (широко-імпульсний модулятор) і блок DERIV (диференціатор) з бібліотеки CONT\_CTL;

- ADD\_REAL(додати значення типу REAL) з бібліотеки IEC, група Arithmetic;

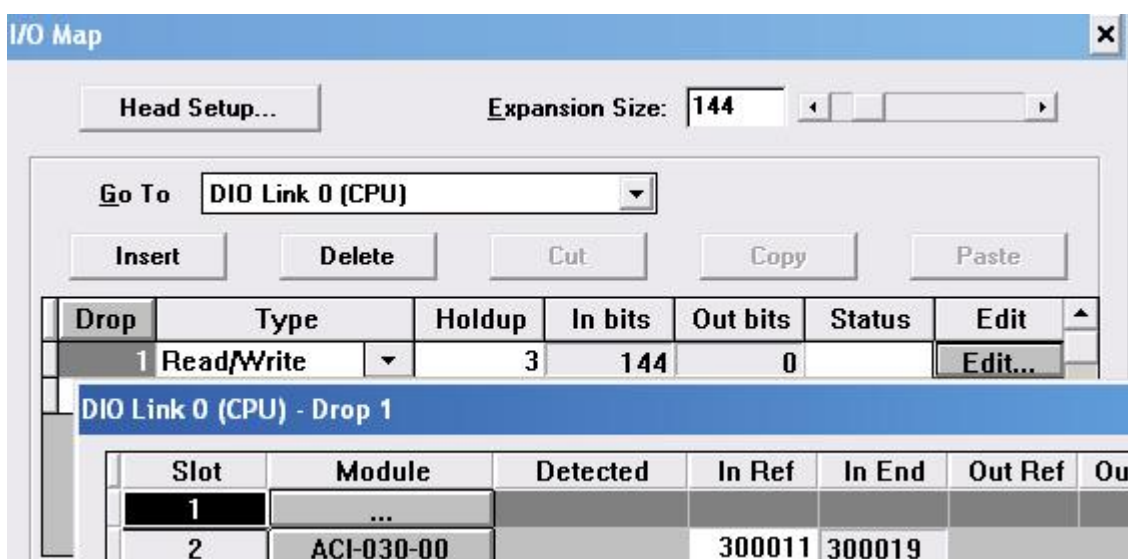
- UINT\_TO\_REAL (перетворити тип даних UINT в REAL) WORD\_TO\_UINT (перетворити тип даних WORD в UINT), REAL\_TO\_UINT (перетворити тип даних REAL в UINT), UINT\_TO\_WORD (перетворити тип даних UINT в WORD) з бібліотеки IEC, група Converter.

Створення додатку користувача у системі програмування Concept передбачає розробку проекту і створення програмного коду.

При розробці проекту здійснюється програмне конфігурування контролера для шасі локального вводу-виводу:



і розподіленого вводу-виводу:



Протягом програмування на підставі структурної схеми системи автоматичного регулювання температури пари оголошуються змінні у вікні Variable Editor і визначаються настроєні параметри функціональних блоків:

**Variable Editor**

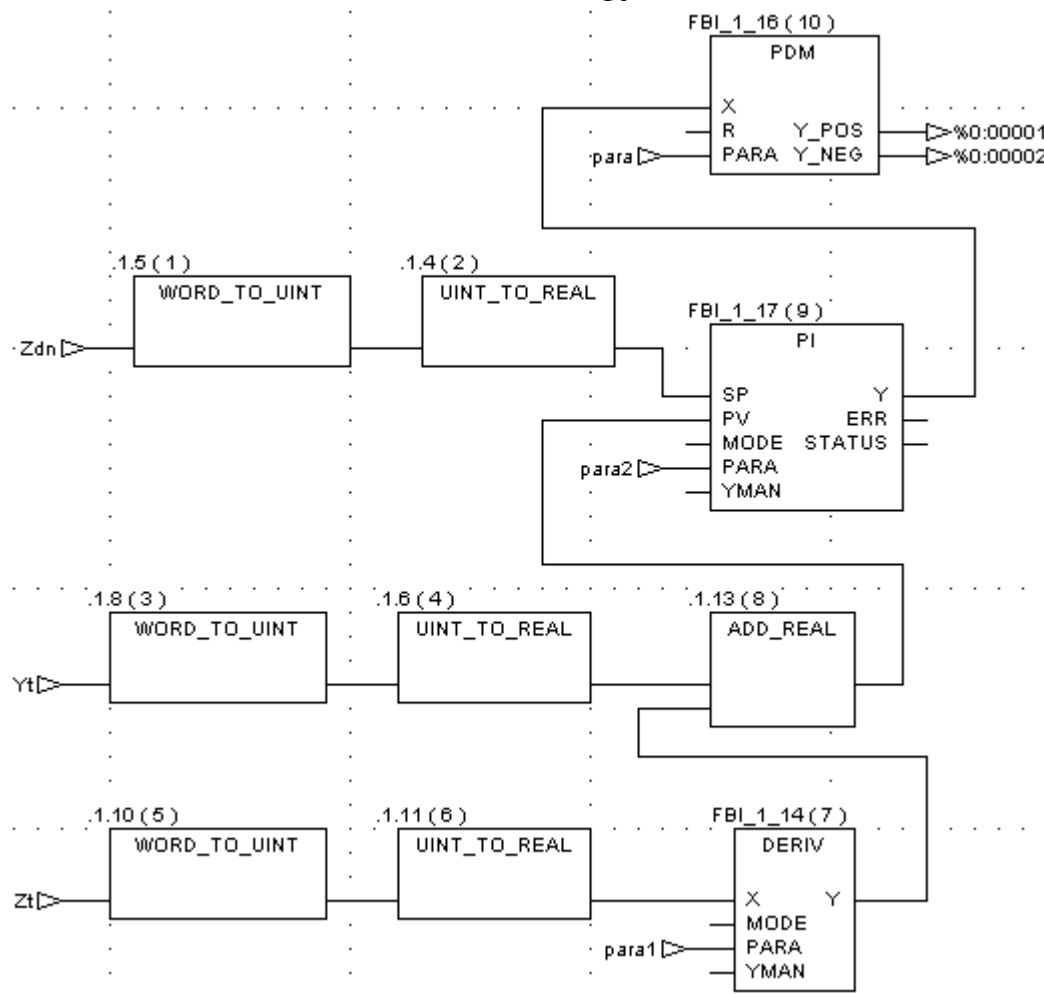
Type  
 Variables  Constants

	Exp	Variable Name	Data Type	Address
1	<input type="checkbox"/>	Yt	WORD	300002
2	<input type="checkbox"/>	Zdn	WORD	300001
3	<input type="checkbox"/>	Zt	WORD	300003

Variables  Constants

	Variable Name	Data Type	Value
1	para	Para PDM	Set...
2	para1	Para DERIV	Set...
3	para2	Para PI	Set...

А також створюється програмний код, перевірка роботи якого здійснюється за допомогою емулятора. При наданні нових значень змінній Yt, або Zdn, у програмному коді відбуваються зміни сигналу у сполучних лініях, що зв'язують функціональні блоки, а також активізуються у різних кольорах виходи Y\_POS і Y\_NEG (більше-менше) функціонального блока PDM:



Відповідно до створеного програмного коду, змінна Zdn, яка визначає задане значення регульованого параметра (температури пари), після перетворення її типу, подається на вхід SP функціонального блока PI (ПІ-регулятора). Поточне значення температури пари, яке в програмі відображає змінна Yt, подається на вхід PV того ж функціонального блока. Додатково на цей вхід через суматор ADD\_REAL подається похідна проміжної змінної Zt (температури пари після I-ї ступені пароперегрівника). Диференціювання Zt, після перетворення її типу відбувається у функціональному блоці DERIV.

Функціональний блок PI формує керувальний вплив, який з виходу Y у вигляді аналогового сигналу подається на вхід X функціонального блока широтно-імпульсного модулятора PDM. Завдяки властивостям цього блоку на виході контролера з адресами %0:00001 і %0:00002 подається модульований з аналогового імпульсний сигнал, який керує роботою виконавчого механізму сталої швидкості, що змінює подачу конденсату на охолодження перегрітої пари.

### ***Висновки***

1. На базі літературних джерел зроблено опис будови і технології виробництва пари у багатопаливному котлоагрегаті, а також наданий словесний алгоритм регулювання технологічних параметрів.

2. На основі розуміння роботи парогенератора та розглянутих алгоритмів автоматичного регулювання технологічних параметрів розроблена схема інформаційних потоків мікропроцесорної системи, яка складається з 13 вхідних сигналів і 6 вихідних.

3. По каталогах технічних засобів автоматизації підібрані датчики технологічних параметрів і виконавчі механізми, які отримують керувальний вплив від мікропроцесорного контролера. Оскільки усі датчики мають на виході уніфіковані струмові сигнали з діапазоном 4-20мА, то для роботи з ними вибрано два 8-канальні модулі аналогового вводу 140 ACI 030 00.

Для керування роботою шістьох виконавчих механізмів сталої швидкості, задіяні дванадцять виходів 16-канального дискретного модуля виводу 140 DRA 840 00, які оснащені вихідними реле з нормально відкритими контактами.

Для опитування 13 входів, обробки інформації, відповідно до алгоритмів керування окремими технологічними параметрами, оновлення станів 12 вихідних каналів, реалізації одної гілки розподіленого вводу-виводу та здійснення інших процедур у системі управління, використовується центральний процесор 140CPU 113 03.

На підставі сумарного струму – 1599мА, що споживають модулі центрального шасі підібрано модуль живлення 140CPS 111 00, який має вхідну напругу ~100-276В, а вихідний струм шини дорівнює максимум 3А.

Для реалізації розподіленого вводу-виводу використовуються два шасі – центральне 4-слотове 140 XBP 004 00 і розподілене 2-слотове 140 XBP 002 00. На центральному установлені модуль живлення, центральний процесор, аналоговий модуль вводу і дискретний модуль виводу, а на розподіленому –

адаптер вузлу 140 CRA 211 10 і аналоговий модуль вводу. Для з'єднання порту Modbus Plus центрального процесора з адаптером використовується кабель 490 NAA 271 01.

4. Для реалізації автоматичного регулювання температури перегрітої пари, розроблено додаток користувача FBD-мовою. При створенні програмного коду використані функціональні блоки PI1, PDM, DERIV, ADD\_REAL, UINT\_TO\_REAL, WORD\_TO\_UINT, REAL\_TO\_UNIT і UINT\_TO\_WORD. Перевірка роботи програми на симуляторі підтвердила правильність створеного додатку користувача.

### *Література*

1. Бузников Е.Ф. и др. Производственные и отопительные котельные. М. «Теплотехника», 1984.- 234 с.
2. Modicon Inc. Quantum Series Automation. Технические средства. Справочное руководство.-<http://www.schneider-electric.ua/>
3. Modicon Inc. Платформа автоматизации Modicon Quantum.-<http://www.schneider-electric.ua/>
4. Элемер. Каталог продукции 2010.
5. CD Альтера. Каталог для энергетиков и инженеров по автоматизации.
6. ПЭК. Исполнительные механизмы пусковые и управляющие устройства. Каталог продукции
7. Ніколаєнко А.М. Програмування систем реального часу. Методичні вказівки до лабораторного практикуму. Запоріжжя, 2010.-49с.

### **13.2.2 Приклад 2**

**Завдання.** Зрозуміти технологію завантаження шихти у приймальні бункери агломераційних машин і принципи управління роботою механізмів. Скласти схему інформаційних потоків, виконати підбір необхідних датчиків і виконавчих пристроїв, здійснити проектне компонування контролера ADAM 5510KW і розробити у середовищі MULTIPROG SFC-мовою програмний додаток для системи автоматичного управління процесом завантаження приймальних бункерів, виконати перевірку роботи програми і описати її роботу.

#### ***Технологія завантаження шихти у приймальні бункери агломераційних машин і принципи управління роботою механізмів***

На ділянці приймальних бункерів спікального відділення аглофабрики здійснюється розподіл потоку шихти між трьома паралельно розташованими бункерами. Як розподільний пристрій використовується автостела. Під кожним бункером є віброживильник, який безперервно розвантажує шихту у подальшому технологічному напрямі.

Алгоритм завантаження бункерів наступний: спочатку передбачається, що усі 3 бункери порожні, про що свідчить сигнал з датчика нижнього рівня

і автостела знаходиться над першим бункером. Оператор перемиканням ключа вибирає режим управління - “ручний” або “автоматичний”.

В автоматичному режимі завантаження бункерів відбувається за циклічною схемою, коли автостела рухається вперед (прямий цикл завантаження) і послідовно заповнює усі бункери до спрацьовування датчиків верхнього рівня. Після цього цикл завантаження повторюється у зворотному напрямі (зворотний цикл). Переміщення автостели від одного бункера до іншого здійснюється за часом.

У режимі ручного керування оператор має нагоду завантажити будь-який бункер натисненням відповідної кнопки. Якщо автостела при переході з автоматичного на ручне керування, рухалася від одного бункера до іншого, то вона зупиниться і відновить рух, коли оператор натисне кнопку завантаження потрібного бункера.

При переході з ручного на автоматичне керування завантаження продовжується у тому ж циклі, при якому воно було перерване при переході на ручний режим.

Якщо відбулася аварійна зупинка конвеєра подачі шихти до бункерів, то рух автостели у будь-якому напрямі блокується.

### *Схема інформаційних потоків*

Щоб визначити всі вхідні і вихідні сигнали мікропроцесорної системи і здійснити проектне компонування контролера ADAM 5510KW для системи автоматичного управління процесом завантаження приймальних бункерів, необхідно на основі розглянутих алгоритмів ручного і автоматичного управління роботою механізмів скласти схему інформаційних потоків, рис 13.2.

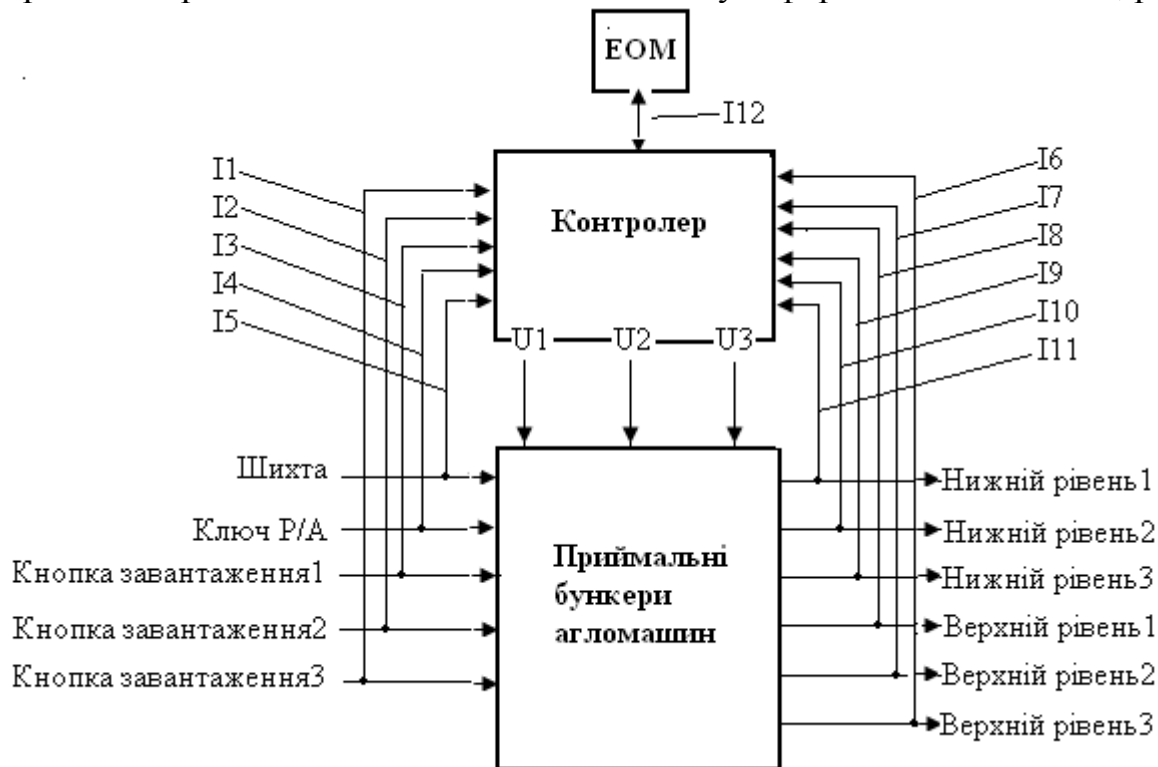


Рисунок 13.2 – Схема інформаційних потоків.



Потоки інформаційних сигналів:

- I1 –кнопка ручного керування завантаженням бункера №3;
- I2 –кнопка ручного керування завантаженням бункера №2;
- I3 –кнопка ручного керування завантаженням бункера №1;
- I4 –перемикач вибору режиму роботи;
- I5 –кнопка керування подачею шихти;
- I6 –верхній рівень шихти у бункері №3;
- I7 –верхній рівень шихти у бункері №2;
- I8 –верхній рівень шихти у бункері №1;
- I9 –нижній рівень шихти у бункері №3;
- I10 –нижній рівень шихти у бункері №2;
- I11–нижній рівень шихти у бункері №1;

Потоки керувальних сигналів:

- U<sub>1</sub> –рух автостели у прямому напрямі;
- U<sub>2</sub> –рух автостели у зворотному напрямі;
- U<sub>3</sub> –аварійна зупинка конвеєра подачі шихти.

### *Визначення кількості та типу вхідних і вихідних сигналів мікропроцесорного контролера*

Визначення кількості та типу вхідних і вихідних сигналів контролера здійснюється на основі аналізу роботи існуючих засобів вибору режимів роботи, ручного керування, вихідних сигналів датчиків, що контролюють технологічні параметри, а також виконавчих пристроїв, табл.13.4–13.7.

Таблиця 13.4

Вхідні інформаційні сигнали

№ п/п	Найменування вхідного сигналу	Стан сигналу	Тип сигналу
1	Ручне керування завантаженням бункера №1	Увімкнено/Вимкнено	Дискретний
2	Ручного керування завантаженням бункера №2	Увімкнено/Вимкнено	Дискретний
3	Ручного керування завантаженням бункера №3	Увімкнено/Вимкнено	Дискретний
4	Вибір режиму роботи	Авт/Руч	Дискретний
5	Подача шихти	Так/Ні	Дискретний
6	Верхній рівень шихти у бункері №1	Перевищення установленного значення	Дискретний
7	Верхній рівень шихти у бункері №2	Перевищення установленного значення	Дискретний
8	Верхній рівень шихти у бункері №3	Перевищення установленного значення	Дискретний
9	Нижній рівень шихти у бункері №1	Нижче установленного значення	Дискретний
10	Нижній рівень шихти у бункері №2	Нижче установленного значення	Дискретний
11	Нижній рівень шихти у бункері №3	Нижче установленного значення	Дискретний

Керуючись даними, що наведені у табл.13.4, і каталогами провідних виробників технічних засобів автоматизації – «СВ АЛЬТЕРА», «ВК Електро - Сервіс» підібрані необхідні датчики та виконавчі механізми і визначені їх вихідні та вхідні сигнали, табл.13.5 і 13.7.

Таблиця 13.5 Технічні засоби, що формують вхідні сигнали контролера

№ п/п	Параметр	Назва і тип датчика або виконавчого механізму	Вихідний сигнал	Постачальник технічного засобу
1	Рівень шихти	Сигналізатор граничного рівня Pointek ULS 200	Нормально відкритий контакт	«СВ Альтера»
2	Кнопка ручного керування	Кнопка XB2-BS542	Нормально відкритий контакт	«СВ Альтера»
3	Перемикач вибору режиму роботи	Перемикач VJ21 2P	Два положення	«СВ Альтера»
4	Кнопка керування подачею шихти	Кнопка XB2-BS542	Нормально відкритий контакт	«СВ Альтера»

Таблиця 13.6 Вихідні сигнали контролера

№ п/п	Найменування вихідного сигналу	Комутовані параметри	Тип сигналу
1	Рух автостели у прямому напрямі	~220В, 2А	Дискретний «сухий контакт»
2	Рух автостели у зворотному напрямі	~220В, 2А	Дискретний «сухий контакт»
3	Аварійна зупинка конвеєра подачі шихти	~220В, 2А	Дискретний «сухий контакт»

Таблиця 13.7 Виконавчі пристрої, що отримують вихідні сигнали

№ п/п	Параметр	Назва і тип виконавчого механізму, що змінює параметр	Сигнал керування виконавчим пристроєм	Постачальник технічного засобу
1	Рух автостели	Малогабаритний контактор КМИ з реверсивним електродвигуном	Дискретний - «сухий контакт»	ООО «ВК Електро-Сервіс»
2	Зупинка конвеєра подачі шихти	Реле проміжне РЭК 77	Дискретний - «сухий контакт»	ООО «ВК Електро-Сервіс»

Відповідно до технології завантаження бункерів, алгоритму управління механізмами, наведеної схеми інформаційних потоків, а також таблиць 13.4 – 13.7, задача автоматизації процесу завантаження приймальних бункерів

агломашин відноситься до класу логічних. Уся інформація про стан об'єкта автоматизації надається у дискретній формі. Вибір режиму роботи, керування механізмом завантаження бункерів в ручному та автоматичному режимах, а також роботою конвеєра подачі шихти, також здійснюється за допомогою дискретних сигналів. Тобто усі вхідні та вихідні сигнали, з якими працюватиме контролер, – дискретного типу.

### ***Проектне компонування контролера ADAM 5510KW***

ADAM 5510KW – це контролер фірми Advantech, який відповідає усім вимогам міжнародного стандарту IEC-61131-3 і може програмуватися п'ятьма технологічними мовами.

ADAM 5510KW призначений для використання в локальних розосереджених системах автоматизації у якості автономного контролера.

Контролер забезпечує прийом та видачу аналогових і дискретних сигналів, первинне перетворення їх відповідно до запрограмованих користувачем алгоритмів та обмін інформацією за послідовними каналами зв'язку на базі інтерфейсу RS-485. Мікроконтролер ADAM 5510KW має модульний принцип будови. Основою його конструкції є базовий блок, який складається з процесорного модуля, перетворювача напруги, 4-слотової пасивної панелі, трьох послідовних комунікаційних портів. В слоти контролера можуть бути установлені до 4-х модулів вводу-виводу серії ADAM 5000 із загальною кількістю каналів до 64. Живлення мікроконтролера здійснюється напругою  $=24В$ .

ADAM 5510KW має вбудовану Flash-пам'ять на 1,5Мбайт і статичну оперативну пам'ять на 640Кбайт для завантаження операційної системи та збереження даних.

Проектне компонування контролера здійснюється на основі інформації про кількість та вид вхідних і вихідних сигналів, з якими працює контролер у процесі автоматичного управління технологічним процесом.

Відповідно до схеми інформаційних потоків і табл.13.4, кількість сигналів, що надходять до контролера дорівнює 11 – усі вони дискретні.

Керувальні сигнали, які формуються контролером, всього їх 3, також мають бути дискретними, оскільки завданням ПЛК є своєчасне вмикання або вимикання з роботи двигунів автостели і транспортера шихти.

Серед дискретних модулів серії ADAM-5000 є 16-канальний модуль вводу ADAM-5051, який працює з сигналами типу «сухий контакт» і 6-канальний модуль виводу з релейними виходами ADAM-5056. Враховуючи те, що при розв'язанні задачі управління завантаженням бункерів використовуються 11 вхідних сигналів і 3 вихідних дискретних сигнали типу «сухий контакт», використання модулів ADAM-5051 і ADAM-5056 повністю задовольняє потребам системи управління.

Вибір процесорного модуля і джерела живлення для ADAM 5510KW не потрібен, оскільки його базовий блок комплектується вбудованим процесором і джерелом стабілізованої напруги, які розраховані на роботу з 4-ма модулями.

## ***Розробка проекту і створення програмного коду системи автоматичного управління процесом завантаження приймальних бункерів***

MULTIPROG – це SoftLogic-система програмування промислових логічних контролерів, яка відповідає стандарту IEC 61131-3. Система базується на віконній технології, яка використовує графічні можливості MS - Windows. Тому принципи стандарту здебільшого реалізуються графічно з символами і іконами або діалогами, в яких можуть бути визначені властивості елементів.

За допомогою MULTIPROG можна програмувати на двох текстових ST та IL і трьох графічних – LD, FBD та SFC мовах стандарту IEC 61131-3. Здійснюється це в діалоговому режимі, використовуючи, так званий, Project Wizard (Майстр Проекту).

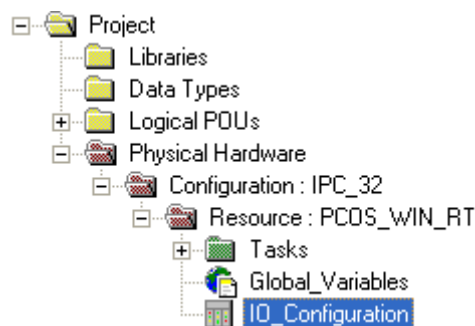
Графічний редактор системи дозволяє вільне розміщення об'єктів на робочому полі проекту, вибирати мову програмування та використовувати бібліотеку операторів, функцій і функціональних блоків. Він підтримує змішування графічних мов в одному робочому листку, встановлення нових елементів в існуючі мережі, переміщення об'єктів або мереж.

Текстовий редактор дозволяє легко і швидко створювати програми, завдяки кольоровому зображенню ключових слів, використанню бібліотечних блоків і елементів.

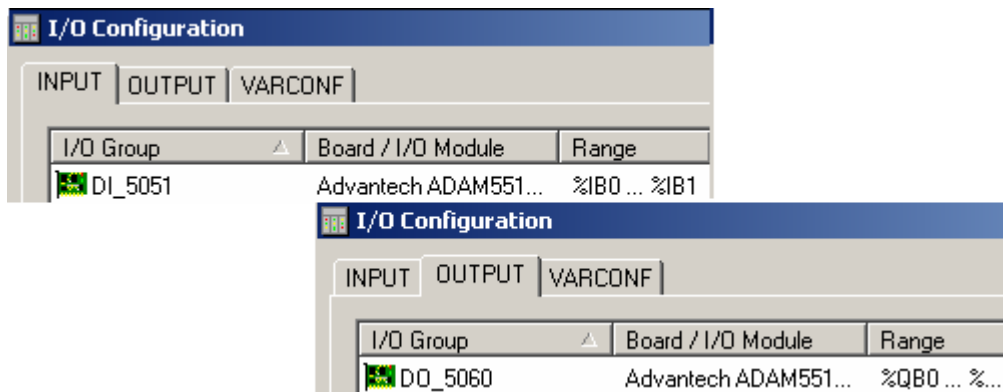
Система програмування має могутній засіб налагодження із зручним інтерфейсом користувача. За допомогою MULTIPROG перевірка програми контролера здійснюється на симуляторі без участі самого контролера.

Створення додатку користувача у системі програмування MULTIPROG передбачає розробку проекту і створення програмного коду.

Проект у MULTIPROG – це деревовидне меню, яке складається з бібліотеки(Libraries), типу даних(Data Type), програмних модулів (Logical POU's) і набору елементів конфігурації, які зображені у піддереві Physical Hardware (Фізична апаратура):



При програмному конфігуруванні контролера у вкладці I/O\_Configuration директорії Physical Hardware визначаються модулі вводу-виводу, якими компонується контролер:



Протягом програмування на підставі алгоритма управління процесом завантаження приймальних бункерів оголошуються змінні і створюється програмний код. Список оголошених змінних наведено у наступні таблиці:

Name	Type	Usage	Description	Address	Init
[-] Default					
Pusk_konveer	BOOL	VAR	Сигнал о включении конвейера подачи шихты	%IX0.0	
Ruchnoe	BOOL	VAR	Ключ управления в положении "Ручное"	%IX0.1	
Avtomat	BOOL	VAR	Ключ управления в положении "Автоматическое"	%IX0.2	
Bunker_1	BOOL	VAR	Кнопка ручного управления загрузки бункера №1	%IX2.0	
Bunker_2	BOOL	VAR	Кнопка ручного управления загрузки бункера №2	%IX2.1	
Bunker_3	BOOL	VAR	Кнопка ручного управления загрузки бункера №3	%IX2.2	
Bunker1_max	BOOL	VAR	Датчик максимального уровня шихты в бункере №1	%IX1.0	
Bunker2_max	BOOL	VAR	Датчик максимального уровня шихты в бункере №2	%IX1.1	
Bunker3_max	BOOL	VAR	Датчик максимального уровня шихты в бункере №3	%IX1.2	
Vpered	BOOL	VAR	Движение загрузочной тележки в направлении "Вперёд"	%QXD.0	
Nazad	BOOL	VAR	Движение загрузочной тележки в направлении "Назад"	%QXD.1	
Ostanov_konveer	BOOL	VAR	Сигнал в шихтовое отделение на остановку конвейера подачи шихты	%QXD.2	
Timer	TIME	VAR	Время передвижения тележки между бункерами (const)		t#5000ms
Tek_bunker	INT	VAR	Номер текущего бункера		
time_bunker	TIME	VAR	Текущее время передвижения тележки между бункерами		
Stop_konveer	BOOL	VAR	Промежуточная переменная		
TON_Stop_kon...	TON	VAR	Таймер остановки конвейера		
time_stop_konv...	TIME	VAR	Текущее время таймера остановки конвейера		
Direction_vp	BOOL	VAR	Текущее направление движения тележки "вперёд"		
Direction_nz	BOOL	VAR	Текущее направление движения тележки "назад"		
pr_cl	BOOL	VAR	Переменная перехода на прямой цикл загрузки		
obr_cl	BOOL	VAR	Переменная перехода на обратный цикл загрузки		
Ruchnoe_1	Ruc...	VAR	ФБ ручного управления		
Perehod_r	BOOL	VAR	Переменная перехода к ручному управлению		
Perehod_a	BOOL	VAR	Переменная перехода в автоматический режим		
Step	BOOL	VAR			
tn_1	BOOL	VAR			
TON_1	TON	VAR			
tn_2	BOOL	VAR			
R_TRIG_1	R_T...	VAR			
TON_2	TON	VAR			
TON_3	TON	VAR			
TON_4	TON	VAR			
Begin	BOOL	VAR			

Загальна структура алгоритму управління завантаженням бункерів розроблена SFC-мовою, рис.13.3. Вона реалізує прямий і зворотний цикли в роботі автостели, а також режим ручного керування. Програмування дій кроків і умов переходів SFC-структури виконано LD-, FBD- і ST-мовами.

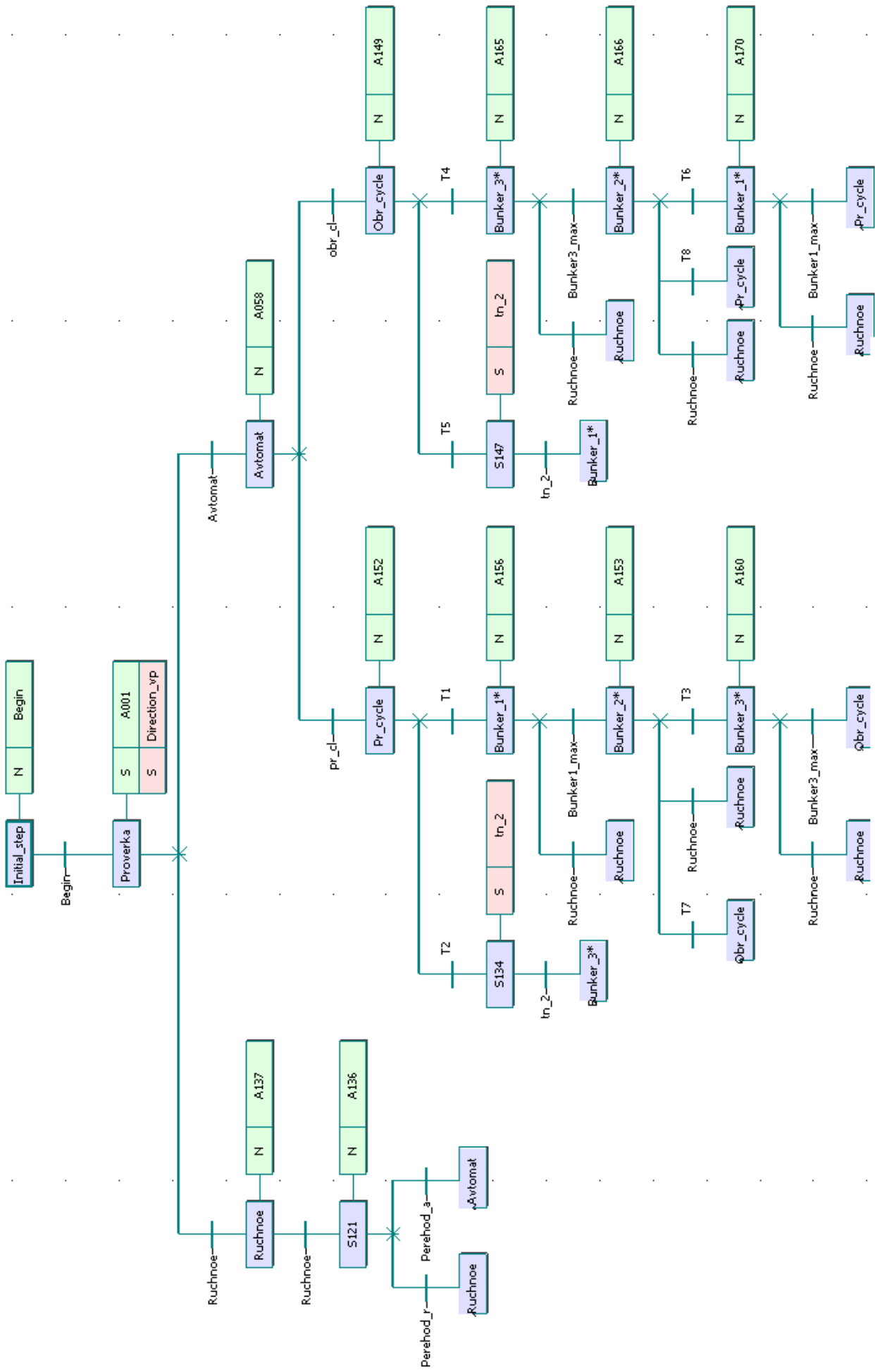
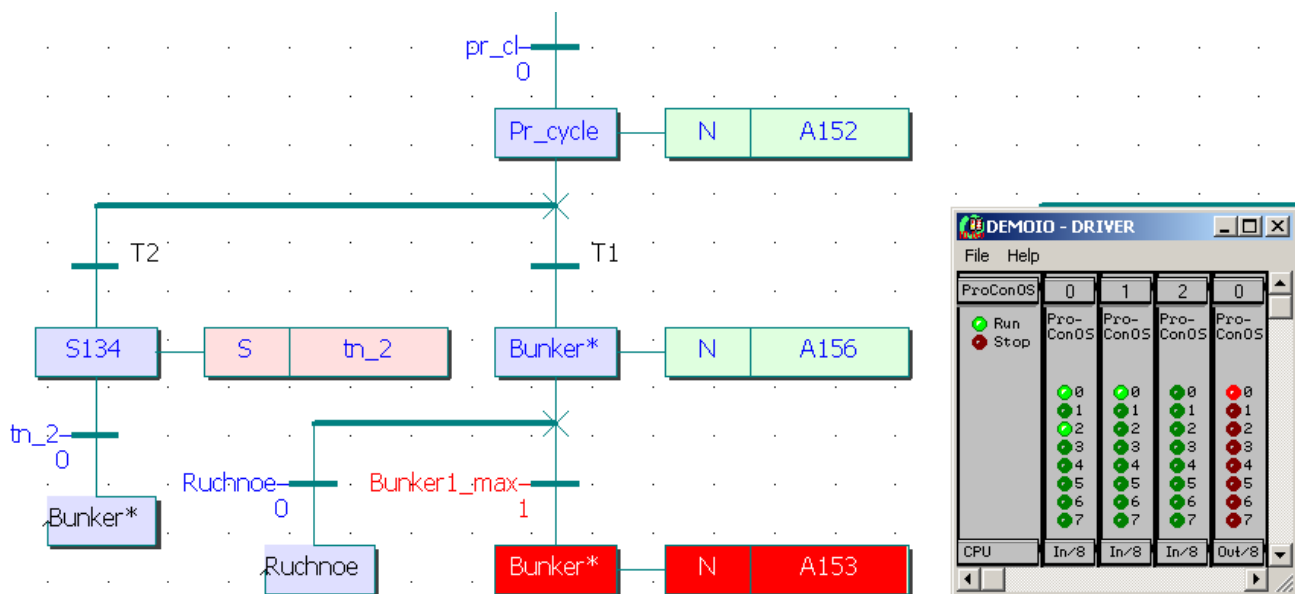


Рисунок 13.3 – SFC-структура алгоритму управління завантаженням бункерів

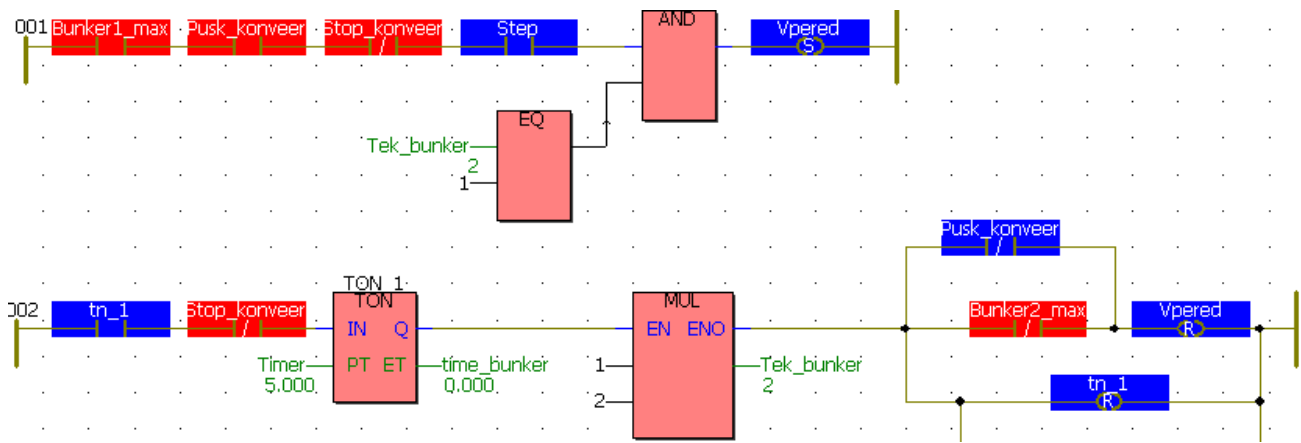
SFC-структура є комбінацією кроків і переходів, що поєднуються у певній послідовності. Альтернативні гілки, що йдуть після кроку Proverka, відповідають за ручний режим роботи автостели і автоматичний. Умова переходу визначається станом змінних Ruchnoe і Avtomat. В автоматичному режимі, залежно від стану змінних pr\_cl і obr\_cl, здійснюється перехід на одну з альтернативних гілок, що відповідають за прямий або зворотний цикл завантаження. У цих гілках SFC-програми шляхом послідовного виконання кроків і переходів, реалізується алгоритм прямого або зворотного циклу завантаження. При виконанні останньої умови переходу стає активним крок, що здійснює “стрибок” на гілку наступного циклу завантаження. Проміжні альтернативні гілки перевіряють умову переходу на ручний режим, який є пріоритетним по відношенню до інших умов переходу.

У вихідному стані всі бункери порожні і автостела знаходиться над першим бункером. При подачі сигналу про увімкнення конвеєра (0-ий вхід 0-го модуля дискретного вводу) і переведення оператором ключа управління в положення “автоматичне” (2-ій вхід 0-го модуля) починається прямий цикл завантаження. Заповнюється 1-ий бункер і після досягнення максимального рівня шихти, що фіксує датчик верхнього рівня, (0-ий вхід 1-го модуля дискретного вводу), автостела починає рух уперед до 2-го бункера, про що свідчить червоний колір 0-го виходу 0-го модуля дискретного виводу:

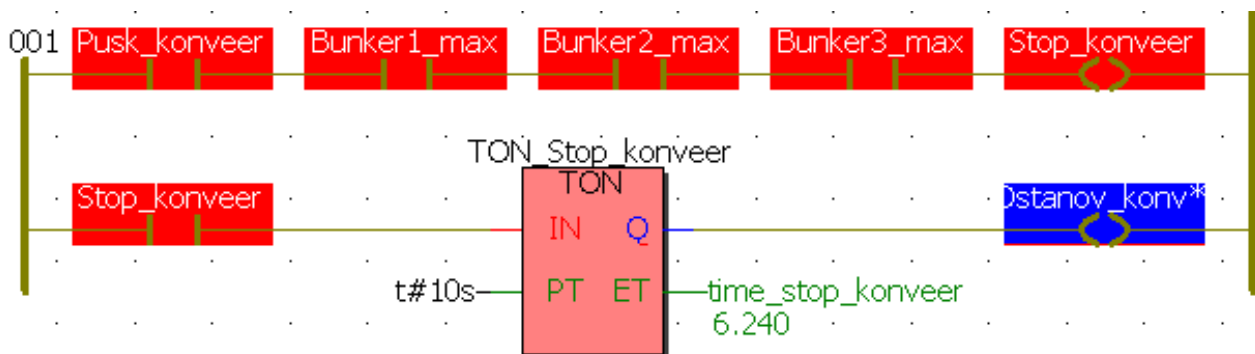


Дія A153 відповідає за переміщення автостели від 1-го бункера до 2-го, написана LD- і FBD-мовами. Фрагмент її має вигляд:

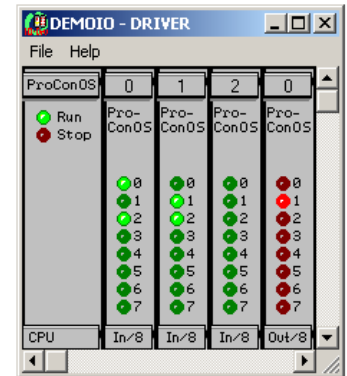
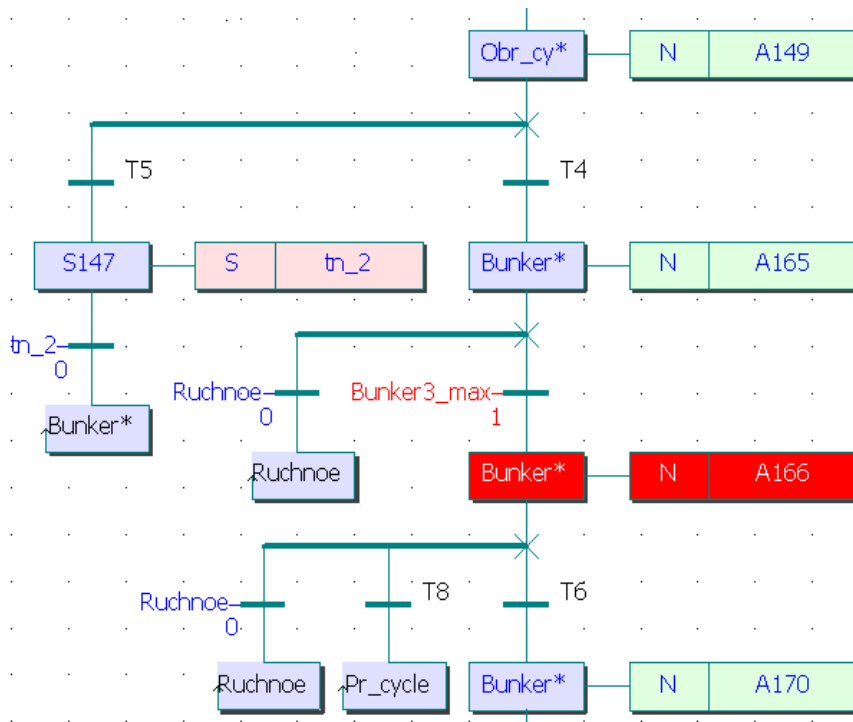




Коли автостела зупиниться над другим бункером, почнеться його завантаження. Після спрацювання датчика верхнього рівня 2-го бункера (1-ий вхід 1-го модуля вводу) автостела зміщується до 3-го бункера і починає його заповнювати. У разі, якщо пуск віброживильників бункерів не відбувся і усі бункери заповнені, щоб уникнути переповнення 3-го бункера, через інтервал часу 10с у шихтове відділення подається сигнал про виключення конвеєра подачі шихти на ділянку. (2-ий вихід 0-го модуля дискретного виводу). Цю перевірку виконує дія A001, яка є активною і незалежна від подальших кроків і переходів:

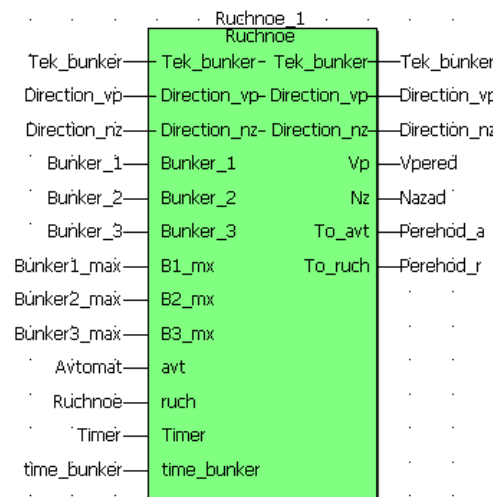


Якщо рівень у бункерах убуває, то починається цикл завантаження, при якому автостела рухається у зворотному напрямі і по черзі завантажує незаповнені бункери до максимального рівня. У наведеному нижче фрагменті програми, автостела рухається назад (1-ий вихід 0-го модуля дискретного виводу) від 3-го бункера до 1-го, не зупиняючись над заповненим 2-им. Час пересування дорівнює 10 с.



Після завантаження 1-го бункера цикл роботи змінюється, і автостела рухається уперед до найближчого незаповненого бункера. За змістом дії прямого і зворотного циклу аналогічні одна одній, відмінність полягає тільки в напрямі руху автостели.

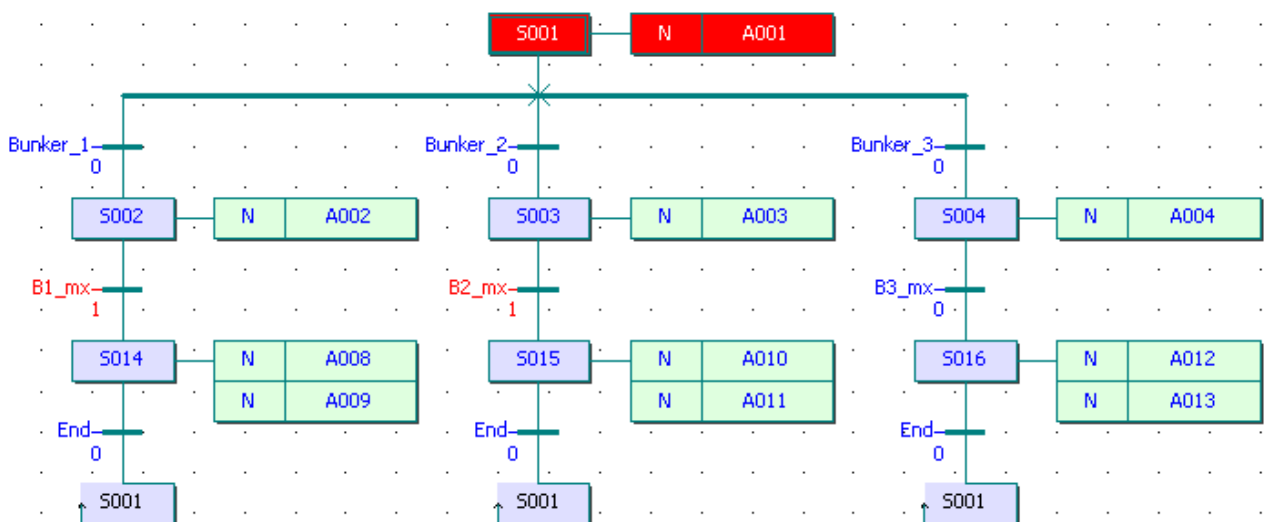
При переході у режим ручного керування, циклічне завантаження бункерів припиняється, і в програмі відбувається перехід до дії, яка відповідає за цей режим роботи. Дія містить функціональний блок користувача, який визначає номер поточного бункера, що завантажується, номер бункера визначеного оператором для завантаження, напрям і час руху автостели до нього:



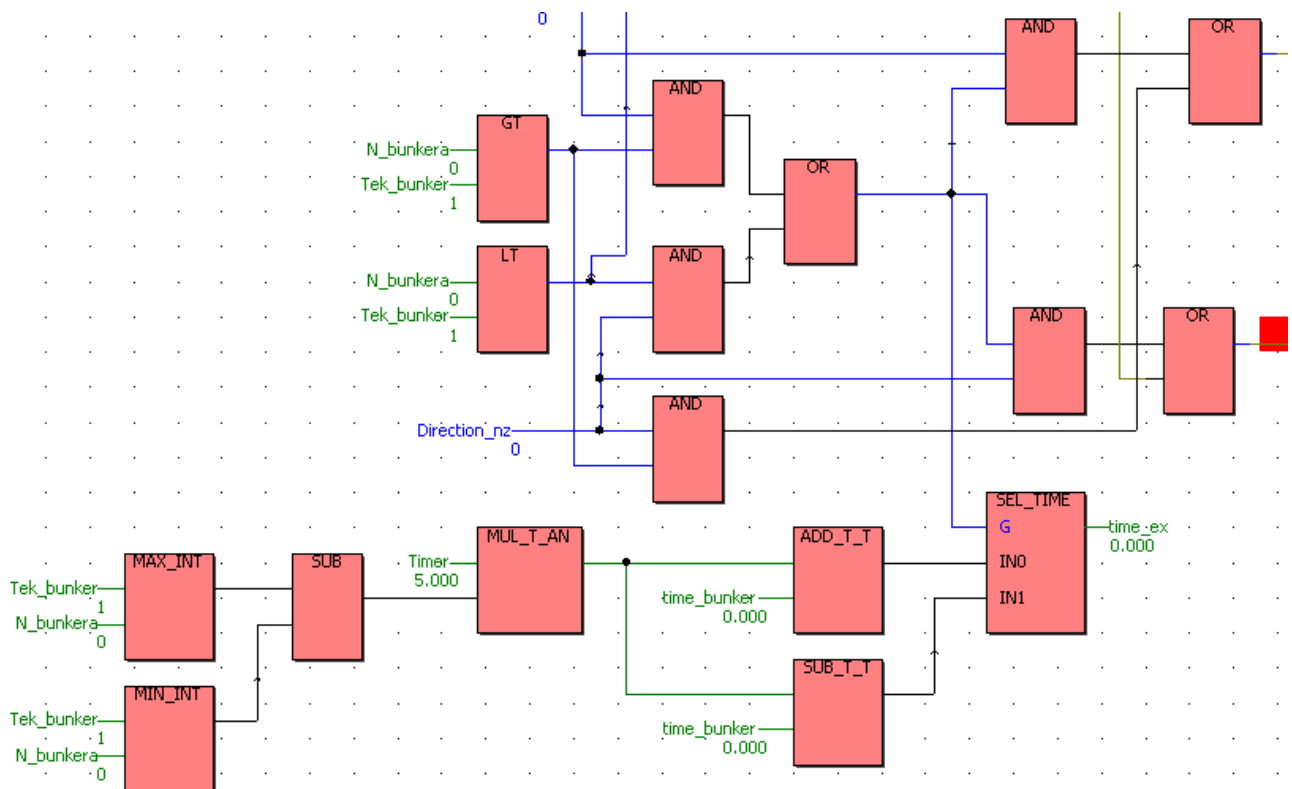
Список оголошених змінних, при створенні функціонального блока користувача, наведений у наступній таблиці:

Name	Type	Usage
<b>Default</b>		
N_bunkera	INT	VAR
Direction_vp	BOOL	VAR_IN_OUT
Bunker_1	BOOL	VAR_INPUT
Tek_bunker	INT	VAR_IN_OUT
Direction_nz	BOOL	VAR_IN_OUT ▾
Timer	TIME	VAR_INPUT
time_bunker	TIME	VAR_INPUT
time_ex	TIME	VAR
TP_Vpered	TP	VAR
TP_Nazad	TP	VAR
Vp	BOOL	VAR_OUTPUT
time_vpered	TIME	VAR
time_nazad	TIME	VAR
Nz	BOOL	VAR_OUTPUT
tp_vp	BOOL	VAR
B1_mx	BOOL	VAR_INPUT
avt	BOOL	VAR_INPUT
To_avt	BOOL	VAR_OUTPUT
ruch	BOOL	VAR_INPUT
To_ruch	BOOL	VAR_OUTPUT
tp_nz	BOOL	VAR
Bunker_2	BOOL	VAR_INPUT
B2_mx	BOOL	VAR_INPUT
Bunker_3	BOOL	VAR_INPUT
B3_mx	BOOL	VAR_INPUT
End	BOOL	VAR

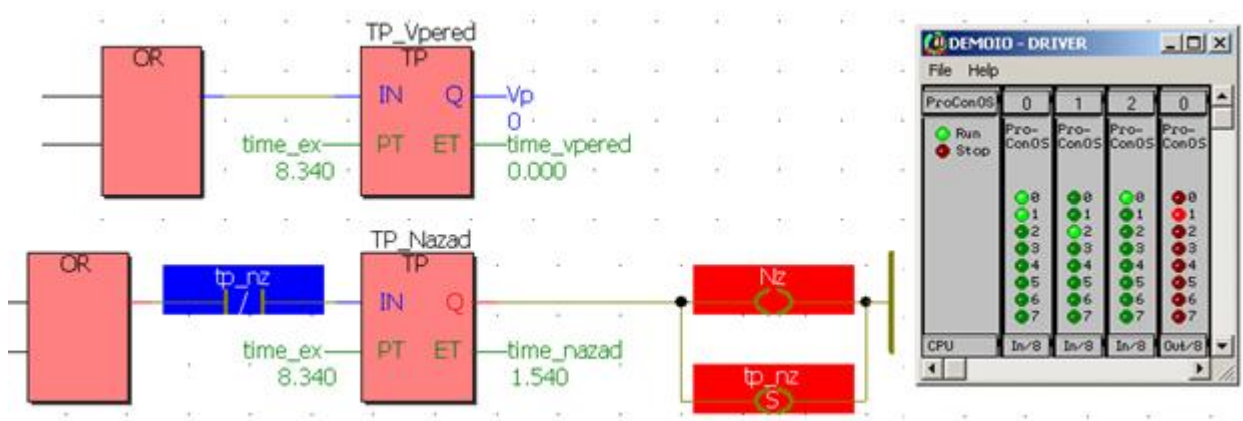
Якщо автостела, при переході з автоматичного на ручне керування, рухалася від одного бункера до іншого, то вона зупиняється. У цьому випадку час її руху до необхідного бункера у ручному режимі визначається, виходячи з поточного напрямку руху автостели, часу, що пройшов з моменту початку її руху і до переходу у ручний режим, а також напрямку, в якому автостела має рухатися при натисненні кнопки оператором. Загальний алгоритм функціонального блока ручного управління написаний SFC-МОВОЮ:



Основні дії – FBD-мовою, із застосуванням блоків порівняння, логічних операцій «І», «АБО», блоків арифметичних операцій, таймерів і блоку вибору SELECT. Фрагмент програми дії A002 має вигляд:

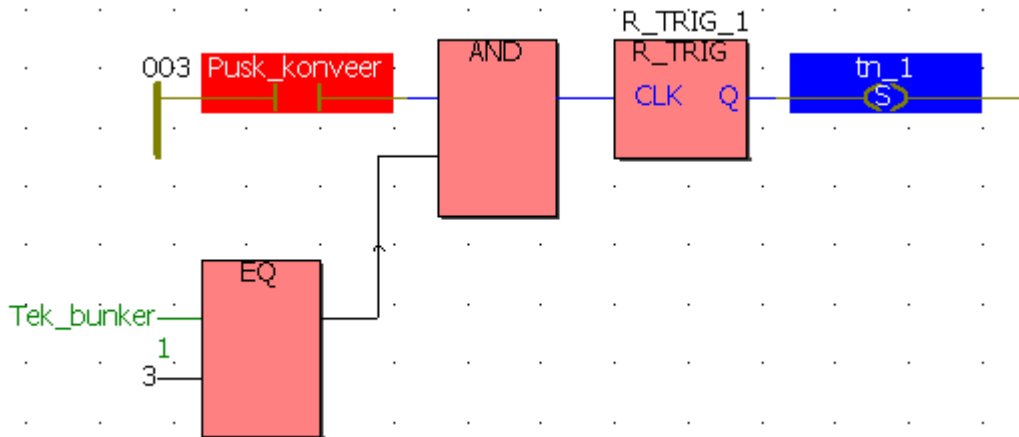


Промодельємо ситуацію, коли під час пересування від 3-го бункера до 2-го, оператору знадобилося завантажити 1-ий бункер. При переході з автоматичного режиму в ручний (1-ий вхід 0-го модуля вводу), автостела зупинилася. Коли оператор натиснув кнопку завантаження 1-го бункера (0-ий вхід 2-го модуля вводу), автостела почала рухатися уперед. Час необхідний для пересування автостели визначається, як час руху її від 3-го бункера до 1-го (10 сек.) мінус час, який автостела вже пройшла від 3-го бункера до 2-го (1.66 сек). Тобто загальний час пересування автостели до 1-го бункера після її зупинки дорівнює 8.34 с:



Після закінчення завантаження 1-го бункера і переходу назад в автоматичний режим, автостела повернеться у циклічний режим завантаження і почне рух уперед до найближчого незаповненого бункера, тобто до 2-го.

Якщо під час пересування автостели між бункерами відбулася раптова зупинка конвеєра подачі шихти, то автостела доїде до найближчого бункера і цикл завантаження припиниться до появи сигналу про включення конвеєра. Фрагмент програми, що відповідає за дану операцію має вигляд:



З наведеного фрагменту видно, що змінна «Pusk\_konveer» - у червоному кольорі, тобто знаходиться у стані TRUE (кнопка пуску конвеєра розімкнена). Як наслідок - транспортер не працює (змінна «tn\_1»- хибна).

### ***Висновки***

1. Зроблено опис технології завантаження приймальних бункерів шихтою, а також наданий словесний алгоритм роботи автостели.

2. Розроблена схема інформаційних потоків мікропроцесорної системи, яка складається з 11 вхідних сигналів і 3 вихідних.

3. Відповідно до технології завантаження бункерів, алгоритму управління механізмами, наведеної схеми інформаційних потоків, задача автоматизації процесу завантаження приймальних бункерів агломашин відноситься до класу логічних. Уся інформація про стан об'єкта автоматизації надається у дискретній формі. Вибір режиму роботи, керування механізмом завантаження бункерів в ручному та автоматичному режимах, а також роботою конвеєра подачі шихти, також здійснюється за допомогою дискретних сигналів.

4. По каталогам технічних засобів автоматизації підібрані датчики контролю граничних значень рівня шихти у приймальному бункері, засоби вибору режимів роботи, ручного керування, а також виконавчі пристрої, що отримують керувальний вплив від мікропроцесорного контролера.

5. Оскільки усі технічні засоби, що формують 11 вхідних сигналів контролера, мають на виході дискретний сигнал типу «сухий контакт», у якості модуля вводу вибрано 16-канальний ADAM-5051. Для дискретного управління

трьома двигунами використовується 6-канальний модуль з релейними виходами ADAM-5056.

6. Програмний код контролера створений SFC-мовою. Перевірка роботи програми на симуляторі підтвердила відповідність її роботи алгоритму керування автостелою.

### *Література*

1. Котов К.И., Шершевер М.А. Промышленные системы автоматизации металлургических агрегатов, М., «Металлургия», 1980.- 256с.
2. СВ Альтера. Каталог для энергетиков и инженеров по автоматизации.
3. <http://www.electro.ru/>
2. Advantech. ADAM 5510 KW User's Manual.
3. Ніколаєнко А.М. Програмування ПЛК у Softlogic-системі KW MULTIPROG: Навчальний посібник.-Запоріжжя: Видав.ЗДІА, 2008.-203с.

## ЛІТЕРАТУРА

1. Ніколаєнко А.М. Програмування ПЛК у Softlogic-системі KW MULTIPROG: навч.-посіб. [для студ. вищ. навч. закл.] / Ніколаєнко А.М. – Запоріжжя: Видав. ЗДІА, 2008. – 203с.
2. Фролов С.В. Тенденция развития систем управления технологическими процессами / С.В. Фролов // Приборы и системы. Управление, контроль, диагностика. – 1996 – №9. – с. 6-8.
3. Ельперін І. В. Промислові контролери: навч. посіб. [для студ. вищ. навч. закл.] / Ельперін І. В. – К.: НУХТ, 2003. – 320с.
4. Крутских Р.Ю. SCADA для непрерывных производств – Processlogix / Р.Ю. Крутских // Промышленные АСУ и контроллеры. – 2003 – №7. – с.38-42.
5. Фролов С.В. Тенденции развития программируемых логических контроллеров / С. В. Фролов // Приборы и системы. Управление, контроль, диагностика. – 1996 – №10. – с.17-21.
6. Бердычевский М. И. Конструктивы Евромеханики во встраиваемых системах / М. И. Бердычевский // СТА. – 2002 – №4. – с. 52-59.
7. Пономарев Д. И. Технология мезонинных контроллеров. – настоящее и будущее промышленных систем / Д. И. Пономарев, А.Н. Буданов // Приборы и системы. Управление, контроль, диагностика. – 1999 – №8, с.32-36.
8. Сысоев А.Д. Мезонины: что сегодня? / А. Д. Сысоев // Мир компьютерной автоматизации. – 2001 – №4. – с.19-24.
9. Патрахин В.В. Феномен РАС/ В. В Патрахин // Мир автоматизации. – 2008 – №2. – с.20-23.
10. Токхайм Р. Микропроцессоры: Курс и упражнения. / Роджер Токхайм. – М.: Энергоатомиздат, 1988.– 336 с.
11. Громов Д.И. Классификация современных контроллеров / Д. И. Громов // Контроавт: Контроль и автоматика. Методичка. – 2007 – №4. – с.14-23.
12. Микропроцессорные автоматические системы регулирования. Основы теории и элементы: уч. пос. [для студ. высш. уч. зав.] / Солодовников В. В., Коньков В. Г., Суханов В. А., Шевяков О. В. – М.: Высш. Шк. 1991 – 255с.
13. Андреев Е. Б. Технические средства систем управления технологическими процессами нефтегазовой промышленности: уч. пос. [для студ. высш. уч.зав.] / Е. Б.Андреев, В. Е. Попадько – Тюмень: ТНГИ. – 2007 – 270с. – Режим доступа до навчального посібника: <http://twirpx.com/>
14. Кругляк К. Д. Промышленные сети: цели и средства /К. Д. Кругляк// СТА. – 2002– №4.– с. 6-17.
15. Гусев С. П. Краткий экскурс в историю промышленных сетей /С. П. Гусев//СТА. – 2000 – №4. – с. 78-84.
16. Патрахин В. А. Средства программирования РС-совместимых контроллеров /В. А. Патрахин// ПиКАД. – 2003 – №1-2. – с.34-40.
17. Петров И. В. Программируемые контроллеры. Стандартные языки и приёмы прикладного проектирования /Петров И. В.–М.: СОЛОН –Пресс, 2004.– 256 с.



18. Зюбин В. Е. Программирование ПЛК: языки МЭК 61131-3 и возможные Альтернативы /В. Е. Зюбин// Промышленные АСУ и контроллеры. – 2005 – №11.– с. 31-35.
19. ЗЭиМ. Контроллер малоканальный многофункциональный регулирующий микропроцессорный Ремиконт Р-130. Техническое описание и инструкция по эксплуатации. г. Чебоксары. – 1989 – с.230.
20. ЗЭиМ. Контроллеры малоканальные многофункциональные регулирующие микропроцессорные Ремиконт Р-130 с блоком контроллера БК-1М/01. Руководство по эксплуатации. г. Чебоксары. – 2005– с.105.
21. Контроллеры КРОСС-500, ТРАСА-500, Р-130ISa. Руководство по эксплуатации. Система технологического программирования IsaGraf. – 2005 – с.117.
22. Микрол. Контроллер малоканальный многофункциональный микропроцессорный МІК-51. Руководство по эксплуатации. г. Ивано-Франковск.–2006 – 90с.
23. Программирование в пакете АЛЬФА.
24. Гусев С.П. ADAM 5510 как зеркало современных тенденций автоматизации / С. П. Гусев // СТА. – 1998 – №2. – с. 2–9.
25. Advantech. ADAM 5511User’s Manual. – 2000 – 286с.
26. Основы программирования для начинающих. Borland Turbo C++ v.3.1. – Режим доступа до довідкового порадника: <http://www.opita.net/node/69>.
27. Advantech. ADAM 5510 KW User’s Manual.
28. Кузнецов А.П. Устройство связи с объектом: Модули фирмы WAGO / Кузнецов А. П. // СТА. – 1997 – №3. – с.22-27.
29. Модульная система WAGO I/O System 750. – 2005 – 192с.
30. Руководство пользователя по программированию ПЛК в CoDeSys 2.3.- М.: СОЛОН-Пресс. 2004 – 253с.
31. SIMATIC S7 Программируемый контроллер S7-300. Справочное Руководство. – Режим доступа до довідкового порадника: <http://automation-drives.ru/>
32. SIMATIC S7 Компоненты систем комплексной автоматизации. Каталог ST70-2007 – 452с.
33. SIMATIC S7 Программирование с помощью STEP 7 V5.0 Руководство. – 1999–671с. – Режим доступа до довідкового порадника: [http:// automation -drives.ru/](http://automation-drives.ru/)
34. Schneider Electric.Серия программируемых контроллеров Modicon TSX Quantum. –1999–29с. – Режим доступа до довідкового порадника: [http:// asutpforum.spb.ru/](http://asutpforum.spb.ru/)
35. Modicon Inc. Quantum Series Automation. Технические средства. Справочное руководство. – 1994 – 135с. – Режим доступа до довідкового порадника: <http://www.schneider-electric.ua/>
36. Modicon Inc. Платформа автоматизации Modicon Quantum.– 2004– 264с. – Режим доступа до довідкового порадника: <http://www.schneider-electric.ua/>

37. Modicon Inc. Руководство по проектированию и монтажу системы удаленного ввода-вывода RIO. – Режим доступа до довідкового poradnika: <http://www.schneider-electric.ua/>
38. Руководство по проектированию и монтажу сети Modicon Modbus Plus. – Режим доступа до довідкового poradnika: <http://www.schneider-electric.ua/>
39. Schneider Electric. Платформа автоматизации Momentum. – 2004 – 42с. – Режим доступа до довідкового poradnika: <http://www.schneider-electric.ua/>
40. Деменков Н.П. Языки программирования промышленных контроллеров: уч. пос. [для студ. высш. уч. зав.] / Деменков Н. П. –М:Изд-во МГТУ им. Н.Э.Баумана, 2004. – 172 с.
41. Schneider Electric. Unity – Справочное руководство по языкам программирования. RUS.pdf. – Режим доступа до довідкового poradnika: <http://www.schneider-electric.ua/>
42. Omron. Программируемые контроллеры CS1. Руководство по эксплуатации. – Режим доступа до довідкового poradnika: <http://www.proenergo.ru/>
43. Omron. Модули Controller Link. Руководство по эксплуатации.–2001–308с. – Режим доступа до довідкового poradnika: <http://www.proenergo.ru/>
44. Omron. CX-Programmer V5. Вводный курс.–2001 – 47с.– Режим доступа до довідкового poradnika: <http://www.proenergo.ru/>
45. Omron. Программируемые контроллеры CS1. Руководство по программированию. – 2001 – 766с. – Режим доступа до довідкового poradnika: <http://www.proenergo.ru/>
46. Rockwell Automation. Руководство по выбору ControlLogix.–2005–76с. – Режим доступа до довідкового poradnika: <http://vdt-automation.com.ua/>
47. Rockwell Automation. Контроллеры ControlLogix, Руководство пользователя. – 2002 – 435с. – Режим доступа до довідкового poradnika: <http://vdt-automation.com.ua/>
48. Rockwell Automation. Шасси ControlLogix, Инструкция по установке. – 1997– 20с. – Режим доступа до довідкового poradnika: <http://vdt-automation.com.ua/>
49. Rockwell Automation. Источники питания ControlLogix, Инструкция по установке. – 1997– 14с. – Режим доступа до довідкового poradnika: <http://vdt-automation.com.ua/>
50. Rockwell Automation. Дискретные модули ввода-вывода Controllogix, Руководство пользователя. – 1998 – 65с. – Режим доступа до довідкового poradnika: <http://vdt-automation.com.ua/>
51. Rockwell Automation. Модули аналогового ввода-вывода Controllogix, Руководство пользователя. – 2003 – 352с. – Режим доступа до довідкового poradnika: <http://vdt-automation.com.ua/>
52. Rockwell Automation. Система резервирования ControlLogix. Руководство пользователя. – 2003 – 147с. – Режим доступа до довідкового poradnika: <http://vdt-automation.com.ua/>

53. Rockwell Automation. Общая методика для программируемых контроллеров Logix 5000. Руководство по программированию. – 2004 – 577с. – Режим доступа до довідкового порадника: <http://vdt-automation.com.ua/>
54. Rockwell Automation. Руководство по применению RSLogix 5000. – 1999 – 55с. – Режим доступа до довідкового порадника: <http://vdt-automation.com.ua/>
55. Rockwell Automation. RSLogix Emulate. Руководство пользователя. – 1999 – 39с. – Режим доступа до довідкового порадника: <http://vdt-automation.com.ua/>
56. Патрахин В.В. Обзор технических средств компании GE Fanuc для промышленной автоматизации / В. В. Патрахин // Автоматизация производства.– 2009– №2.– с. 61-65.
57. GE Fanuc Automation. Контроллеры и устройства ввода-вывода. Каталог продукции. – 2006 – 266с. – Режим доступа до каталогу: <http://www.technolink.net/ua/>
58. GE Fanuc Automation. Программируемые системы управления PACSystems RX7i/ Руководство по монтажу. – 2003 – 65с. – Режим доступа до довідкового порадника: <http://www.technolink.net/ua/>
59. GE Fanuc. С чего начать. Logic Developer–PLC. Инструментальное программное обеспечение ПЛК. – 202 – 148с. – Режим доступа до довідкового порадника: <http://www.technolink.net/ua/>
60. Коноваленко А.М., Серета Є.Ю., Ткачьева Н.В., студенты, Ніколаєнко А.М., проф. Особливості програмування логічних контролерів LOGO!. // Тези допов. XIII наук.-техніч. конф. студ., маг., асп. і викл. ЗДІА. Частина III Запоріжжя: Видавництво ЗДІА, 2008, с.156.
61. Бармин А. С. Устройства локальной автоматики. Микроконтроллеры /А.С. Бармин // СТА. – 2003 – №4.– с.38-42.
62. Siemens. SIMATIC LOGO! Руководство. – 2003 – 308с. – Режим доступа до довідкового порадника: <http://automation-drives.ru/>
63. Ткачук О.С., Вернідуб А.В., студенты, Ніколаєнко А.М., проф. Програмування і робота з контролером МІК-51.// Тези допов. XIII наук.-техніч. конф. студ., маг., асп. і викл. ЗДІА. Частина III Запоріжжя: Видавництво ЗДІА, 2008, с.136.
64. VIPA art of automation. Manual VIPA Systems 300S SPEED, 2007 – с.290.
65. Кандибка В.В., Колопов А.П., студенты, Ніколаєнко А.М., проф. Особливості програмування і роботи з контролером WAGO 750-841.// Тези доп. XIII наук.-техн. конф. студ., маг, асп. і викл. ЗДІА. Частина III Запоріжжя: Видавництво ЗДІА, 2008, с.143.
66. Машковський Р.А., Рибкін К.О., студенты, Ніколаєнко А.М., проф. Програмування РС-сумісних контролерів мовою «С».// Тези допов. XIII наук.-техніч. конф. студ., маг., асп. і викл. Частина III Запоріжжя: Видавництво ЗДІА, 2008, с.133
67. Балабуха В.Р. студент, Ніколаєнко А.М., проф. Розробка лабораторної роботи по програмуванню та дослідженню контролера Digus виробництва GE

Ганус.// Тези допов. XV наук.-техн. конф. студ., маг., асп. і викл. ЗДІА. Частина III Запоріжжя: Видавництво ЗДІА, 2010, с.121.

68. СВ АЛТЕРА. Каталог для енергетиків і інженерів по автоматизації. – 2008 – 232с. – Режим доступу до довідкового порадника: <http://www.svaltera.ua/>

69. Индасофт. Решения для промышленной автоматизации. Краткий каталог продукции и примеры внедрений. – 2007 – 64с. – Режим доступу до довідкового порадника: <http://www.indsoft.com/ua/>

69. Патрахин В. В. «Если мафию нельзя победить – её надо возглавить» / В.В. Патрахин, В. А. Патрахин // ПиКАД. – 2008 – №3. – с.22-26.

## СЛОВНИК ТЕРМІНІВ І СКОРОЧЕНЬ

АСУ ТП	Автоматизована система управління технологічними процесами
АЛП	Арифметично-логічний пристрій
АРМ	Автоматизоване робоче місце
АЦП	Аналогово-цифровий перетворювач
Вита пара	Фізичне середовище для передачі інформації, складається з двох ізольованих проводів, звитих між собою (для зменшення впливу зовнішніх електромагнітних полів). Швидкість передачі – до 1 Мбіт/с
Дуплексне коло	Фізичне коло, яке дозволяє одночасну двоспрямовану передачу даних (схема з чотирма проводами)
Дуплексний протокол	Режим роботи для доточкового сполучення з двома фізичними колами, в яких повідомлення або блоки передачі даних можуть посилатися в обидва напрями одночасно
ДЦП	Дискретно-цифровий перетворювач
ЕСППЗП (EEPROM)	Перепрограмований постійний запам'ятовуючий пристрій з електричним стиранням інформації
ІРПР	Інтерфейс радіальний паралельний
ІРПС	Інтерфейс радіальний послідовний
Коаксиальний кабель	Фізичне середовище для передачі інформації, складається з мідної жили, ізоляції, екрану і зовнішньої оболонки. Випускаються два типи коаксиальних кабелів: товстий і тонкий. Швидкість передачі – до 50 Мбіт/с
Майстер-станція	Пристрій (ПЛК з модулями вводу-виводу або робоча станція), який посилає і збирає дані в/з пристроїв, підключених до напівдуплексної мережі “точка-до-багатоточці”
Мережа Remote I/O	Мережа віддалених ввідів-виводів, що підтримує обмін даними в реальному масштабі часу для управління пристроями
Мережа ControlNet	Швидкодіюча мережа, яка використовується для передачі критичної за часом інформації
Мережа DeviceNet	Відкрита комунікаційна мережа нижнього рівня, яка забезпечує підключення польових пристроїв
Мережа MODBUS	Мережа з послідовним обміном типу провідний/підпорядкований

Мережа PROFIBUS-DP	Мережа, що адаптована для побудови швидких розподілених систем збору даних і управління з одним провідним вузлом
Мережа ASI	Польова шина нижнього рівня, яка забезпечує підключення польових пристроїв
Мережа FIPIO	Мережа, яка орієнтована на віддалений ввід-вивід
Мережа DirectNet	Підтримує передачу даних в режимі провідний/підпорядкований і забезпечує одноранговий зв'язок між процесорами і іншими пристроями
Мережа DH+	Підтримує передачу даних і віддалене програмування контролерів на додаток до однорангового зв'язку між іншими процесорами і пристроями
Мережа Ethernet	Локальна мережа для швидкого рівноправного (однорангова мережа) обміну інформацією між комп'ютерами та іншими пристроями
Мережа I/O Link	Високошвидкісна мережа забезпечує можливість праці за схемою провідний/підпорядкований для пакетів даних з фіксованими об'ємами через волоконно-оптичну лінію
Модем	Пристрій, який модулює цифрову інформацію з ПЛК або ПК в аналоговий сигнал, що транспортується по телефонних лініях, радіохвилях і супутниках, і демодулює аналогові дані в цифрові у місці отримання
МП	Мікропроцесор
ОЗП (RAM–Random access memory)	Оперативний запам'ятовуючий пристрій (Пам'ять з довільним доступом)
Оптоволоконний кабель	Фізичне середовище для передачі інформації, складається з скляної (пластикової) жили, оболонки і декількох шарів захисних покриттів. Швидкість передачі – 100 Мбіт/с і вище
ОП	Операційний пристрій
Паралельний порт	Електричне з'єднання комп'ютера, яке здатне передавати або отримувати два або більше бітів даних одночасно; порт зв'язку, до якого можуть бути підключені такі пристрої, як паралельний принтер
ПК	Персональний комп'ютер
ПЗО	Пристрій зв'язку з об'єктом
ПЗ	Програмне забезпечення

ПЗП (ROM–Read Only Memory)	Постійний запам'ятовуючий пристрій (Пам'ять тільки для читання)
ПЛК (PLC) (Programmable Logic Controller)	Програмований логічний контролер
Полудуплексне коло	Фізичне коло, яке забезпечує передачу даних у будь-якому напрямі, але не одночасно
Полудуплексний протокол	Режим роботи для доточкового або багатоточкового сполучення, в якому повідомлення або блоки передачі даних можуть посилатися в обох напрямках але не одночасно
Послідовний порт	Електричне з'єднання на комп'ютері, яке обробляє біти інформації один за одним; порт зв'язку (COM1 або COM2), до якого можуть бути приєднані пристрої типу модему, миші
ППЗП	Перепрограмований постійний запам'ятовуючий пристрій
Протокол	Набір правил для управління форматом і синхронізацією передачі даних поміж пристроями сполучення, у тому числі підтвердження зв'язку, виявлення помилок і відновлення при помилках
ПУ	Пристрій управління
Розміри контролера	Ширина, висота, глибина (Ш x В x Г)
РЧ	Реальний час
СППЗУ (EPROM)	Перепрограмований постійний запам'ятовуючий пристрій зі стиранням інформації за допомогою опромінювання чипу ультрафіолетовим або рентгенівським промінням
СУБД	Система управління базою даних
Термінатор	Кінцевий резистор
Формат 3U, 6U	Європейський стандарт на геометричні розміри конструктивів і плат контролерів (U=44,45мм)
ЦАП	Цифро-аналоговий перетворювач
ЦДП	Цифро-дискретний перетворювач
ША	Шина адресна
ШД	Шина даних
ШУ	Шина управління
12 біт, 16 біт	Розрядність АЦП у характеристиці аналогових модулів вводу-виводу
АІ/АО	Аналоговий ввід-вивід
АС, VАС	Напруга змінного струму



Bridge (міст)	Пристрій, що поєднує дві мережі, які використовують однакові методи передачі даних
COM/DCOM (Component Object Model)	Модель багатокомпонентних об'єктів, що дозволяє додатку маніпулювати програмними об'єктами. DCOM-Distributed (розподілена) COM
CPU	Центральний процесор
DC, VDC	Напруга постійного струму
DDE (Dynamic Data Exchange)	Динамічний обмін даними – протокол, створений компанією Microsoft для обміну даними поміж різними Windows –додатками. Цей протокол реалізує взаємозв'язки типу клієнт-сервер поміж двох програм, що одночасно виконуються
DF1 Half-Duplex	Асинхронний полудуплексний протокол Allen-Bradley
DF1 Full-Duplex	Дуплексний протокол Allen-Bradley для сполучення “точка – точка”
DI/DO	Дискретний ввід-вивід
16DI, 8AO і т. ін.	Модули вводу-виводу на 16 і 8 каналів, відповідно
DIO	Розподілений ввід-вивід
DCS	Distributed Control System (Розподілена система управління)
DRAM	Динамічний оперативний запам'ятовуючий пристрій
FBD	Мова функціональних блокових діаграм (набір типових програмних модулів)
Flash (Flash Erase EEPROM)	Перепрограмований постійний запам'ятовуючий пристрій, в якому електричне стирання вмісту комірок виконується або для всієї мікросхеми, або для певного блока
HMI (Human Machine Interface)	Людино – машинний інтерфейс
Host (хост)	Головний комп'ютер (контролер) мережі
IL	Мова список інструкцій (створення процедур)
IEC (International Electrotechnical Committee) 61131	Міжнародний стандарт, який визначає таке поняття, як «програмовані контролери»
I/O (Input/Output)	Ввід-вивід
LD	Мова крокових діаграм
Master/Slave	Провідний/підпорядкований
ODBC (Open DataBase Connectivity)	Протокол обміну з реляційними базами даних

OLE (Object Linking and Embedding)	Механізм включення та вбудовування об'єктів
OPC (OLE for Process Control)	Стандарт взаємодії поміж програмними компонентами SCADA и контролерами
OSI (Open Systems Interconnection)	Модель взаємозв'язку відкритих систем
PAC (Programmable Automation Controller)	Програмовані автоматизаційні контролери
PCI, CompactPCI	Стандарти локальної шини для внутрішньоплатних з'єднань
Point-to-point (PPI)	Мережа, в якій забезпечується сполучення тільки між двома вузлами(точка-до-точки)
Point-to-multipoint (MPI)	Мережа, в якій забезпечується сполучення між головною станцією и багатьма віддаленими вузлами (точка-до-багатоточки)
RIO	Віддалений ввід-вивід
RTU (Remoute Terminal Unit)	Віддалений термінальний пристрій
RS-485	Високошвидкісний та перешкодостійкий промисловий послідовний інтерфейс, який передає сигнали диференційними перепадами напруги (0,2...8)В у лінії зв'язку довжиною до 1км
SFC	Мова послідовних функціональних схем (процедури та транзакції)
Softlogic	Інструментальні засоби програмування контролерів, які базуються на вимогах міжнародного стандарту IEC 61131-3
SoftPLC	PC-сумісні контролери
ST	Мова структурованого тексту (типу Pascal)
SQL	Структурована мова запитів до СУБД
SRAM	Статичний оперативний запам'ятовуючий пристрій
VME	Стандарт шини, яка об'єднує у контролері мікропроцесор і модулі (32 або 64 розрядна шина з максимальною пропускнуною спроможністю 40 Мбіт/с).