

Міністерство освіти і науки України
Запорізька державна інженерна академія

Мікропроцесорні та програмні засоби автоматизації

**Методичні вказівки
до лабораторного практикуму, контрольних і курсових робіт**

для студентів ЗДІА спеціальності 6.092500
“Автоматизоване управління технологічними процесами”

Рекомендовано до видання
на засіданні кафедри АУТП,
протокол № 8 від 12.11.2017 р.

Запоріжжя
2018

Мікропроцесорні та програмні засоби автоматизації. Методичні вказівки до лабораторного практикуму, контрольних і курсових робіт для студентів ЗДІА спеціальності 6.092500 “Автоматизоване управління технологічними процесами” /Укл.: А.М. Ніколаєнко, В.В. Довгаль – Запоріжжя, Видавництво ЗДІА 2018.- 153с.

Укладачі:

А.М.Ніколаєнко - канд.техн.наук, проф., В.В. Довгаль - канд.техн.наук, доц.

Відповідальний за випуск :

зав. кафедрою АУТП

професор Пазюк М.Ю.

Рецензія:

Алексієвський Д.Г.- доц., канд. техн. наук, каф. Електронних систем ЗДІА

Зміст

Передмова.....	4
1. Лабораторна робота №1. Програмування і дослідження роботи контролера Durus.....	5
2. Лабораторна робота №2. Програмування і дослідження роботи контролера Logo!Basic.....	12
3. Лабораторна робота №3. Програмування і дослідження роботи контролера MIK-51.....	23
4. Лабораторна робота №4. Програмування і дослідження роботи контролера Modicon TSX Micro.....	38
5. Лабораторна робота №5. Програмування і дослідження роботи контролера VIPA Spread 7 300.....	46
6. Лабораторна робота №6. Програмування і дослідження роботи контролера ADAM 5511.....	54
7. Лабораторна робота №7. Програмування і дослідження роботи контролера Констар К 202.....	63
8. Лабораторна робота №8. Програмування і дослідження роботи контролера WAGO 750 841.....	78
9. Лабораторна робота №9. Програмування і дослідження роботи контролера ADAM 5510KW.....	87
10. Лабораторна робота № 10 Програмування та дослідження роботи контролера Реміконт Р-130.....	96
11. Лабораторна робота №11. Створення проектів у системі програмування Concept.....	105
12. Лабораторна робота №12. Створення проектів у системі програмування Unity PRO XL.....	112
13. Лабораторна робота №13. Створення проектів у системі програмування CX-One.....	117
14. Лабораторна робота №14. Створення проектів у системі програмування RSLogix 5000.....	124
15. Лабораторна робота №15. Створення проектів у системі програмування Proficy Machine Edition	131
16. Лабораторна робота №16. Створення проектів у системі програмування Step7.....	136
17. Лабораторна робота №17. Створення проектів у системі програмування ISaGRAF.....	143
18. Лабораторна робота №18. Створення проектів у системі програмування GX IEC Developer.....	149
Література.....	153

Передмова

Програмовані мікропроцесорні контролери є основними технічними засобами автоматизації, які використовуються на всіх рівнях керування виробництвом. Технічне та програмне забезпечення цих пристроїв оптимізоване для створення, головним чином, систем керування обладнанням і технологічними процесами.

На ринку промислової автоматики великою популярністю користуються PC-сумісні контролери виробництва всесвітньовідомих фірм – Siemens, Schneider Electric, WAGO, Advantech, та інших. Не менш розповсюджені і вітчизняні контролери підприємств Мікрол, Констар. Усі вони мають загальні принципи будови та функціонування, програмуються мовами високого рівня «C», «C++» або технологічними мовами LD, FBD, IL, ST, SFC, які визначені міжнародним стандартом IEC61131-3.

Проте знання цих мов і вміння розробляти програми, ще не означає наявності у студентів–автоматників навичків створення проектів для промислових контролерів та їх налагодження. Окрім вміння розробляти користувальницькі додатки, потрібно знати конструкцію контролерів, вміти працювати з відповідними інструментальними середовищами.

Дисципліна «Мікропроцесорні та програмні засоби автоматизації» викладається студентам на четвертому курсі у першому і другому напівсеместрах. Згідно з навчальним планом об'єм лабораторних занять складає 56 годин. Протягом цього часу студентам пропонується виконати 17 лабораторних робіт, дев'ять з яких виконуються на діючих лабораторних стендах, а вісім на базі пакетів програмного забезпечення мікропроцесорних контролерів відомих закордонних виробників.

Окрім того, методики створення проектів для програмованих логічних контролерів, що надаються у методичних вказівках, застосовуються студентами при виконанні контрольних та курсових робіт.

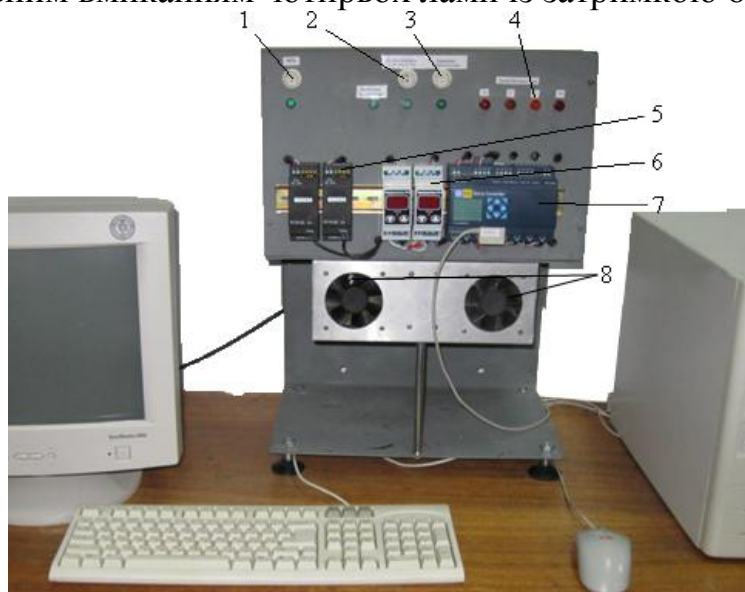
Лабораторна робота №1

Програмування і дослідження роботи контролера Durus.

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

Лабораторна установка, рис.1.1, 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.



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 1.1 – Лабораторна установка

Контролер DURUS

DURUS - контролер виробництва GE Fanuc, який розрахований на нескладні завдання логічного управління, рис.1.2, має розміри 72x90x59,6мм і установлюється на DIN-рейку. Робочий температурний діапазон – в межах 0-55 градусів за Цельсієм.

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.



Рисунок 1.2 – Контролер DURUS

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 використовуються 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.

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 сегментів 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 0,01с 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 налагодити програму, що створена інженерними мовами, навидь 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 Durus 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 OK. У підменю, що відкриється, вибрати мову LADDER і підтвердити кнопкою OK. Очистити пам'ять контролера. Для цього в головному меню перевести курсор на CLEAR PROG., натиснути послідовно OK і YES.

Щоб створити перший сегмент програми, який за сигналами з позиційних терморегуляторів вмикає в роботу вентилятор, необхідно увійти у меню LADDER, встановити курсор у крайнє ліве положення і натиснути кнопку SEL. З'явиться символічне ім'я I1 прямого контакту. Натиснути OK, 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 OK. 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 знаомісце і натиснути кнопку SEL. Кнопкою ↑ встановити час затримки 6 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. Коли 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 клавішею миші натиснути на кнопку New і у вікні Selekt Model Type, що з'явиться, вибрати тип контролера IC210MDR024, натиснути

ОК. 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:

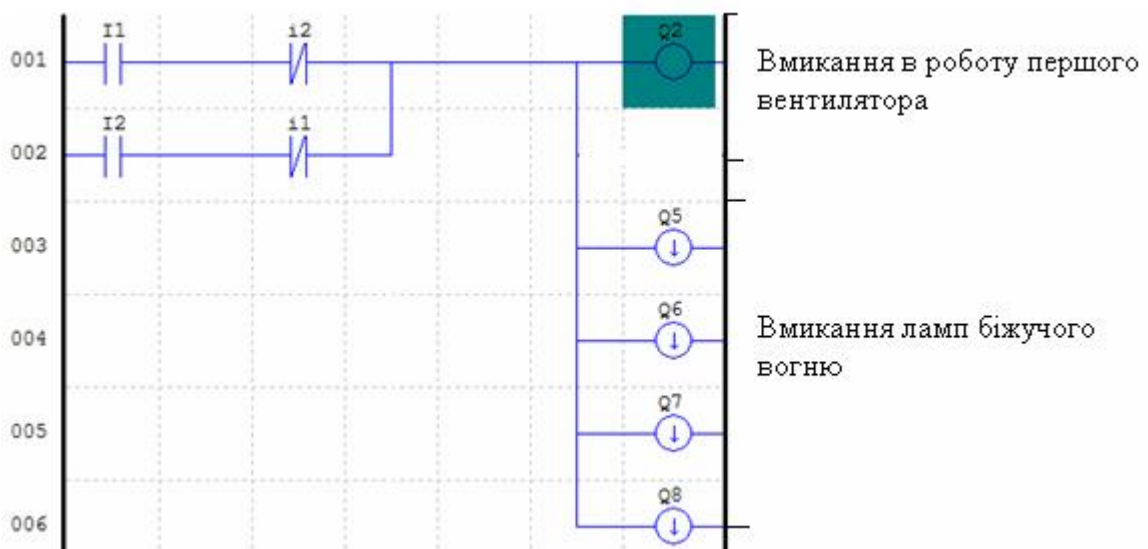


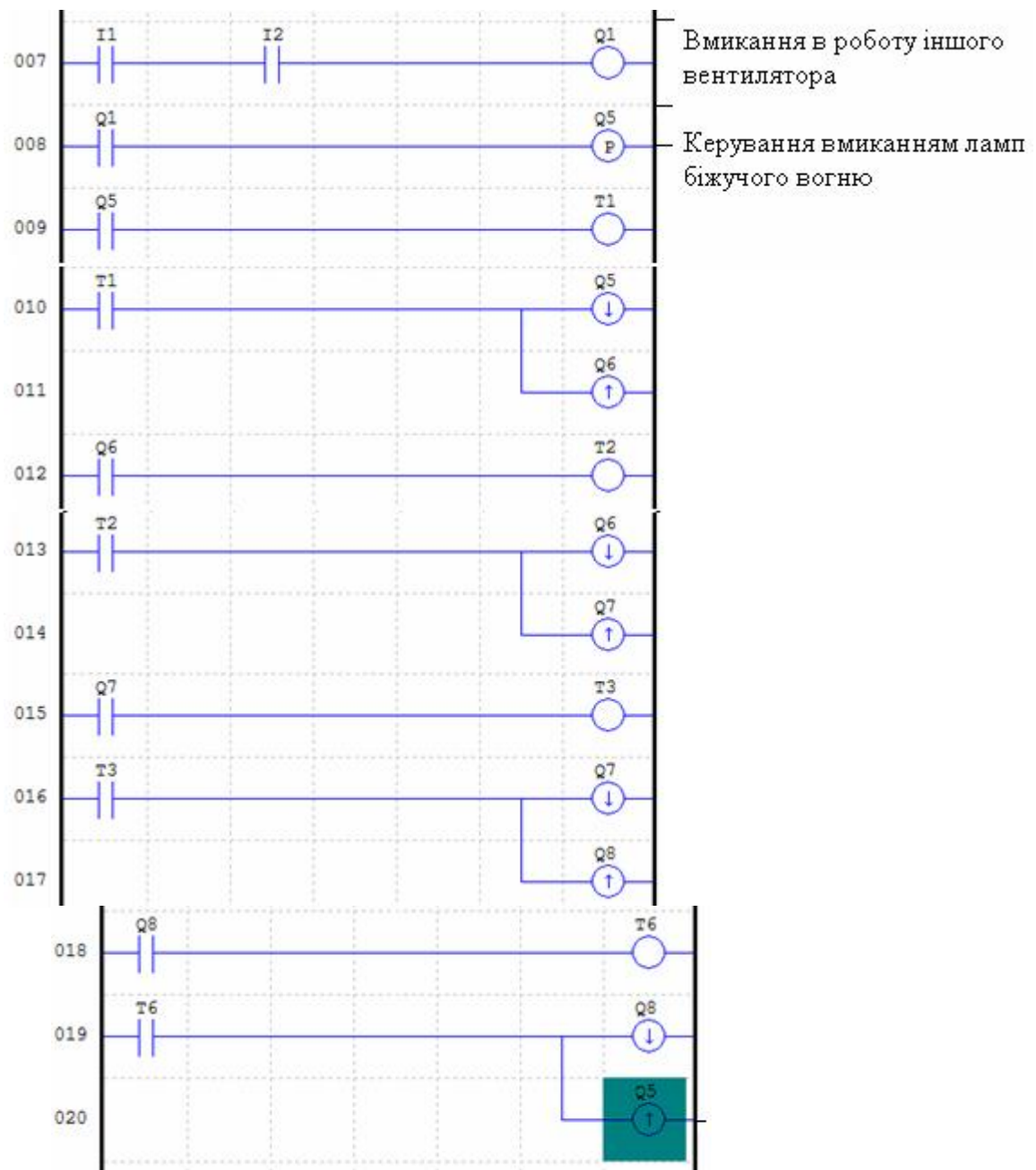
Для створення першого сегменту програми, який забезпечить вмикання першого вентилятора, в панелі інструментів вибрати кнопку I-контакта  і натиснути ліву клавішу миші. Перевести курсор у ліву крайню клітинку першого сегменту програми і повторно клацнути. У вікні Edit Contact, що з'явиться, призначити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


Натиснути ОК і контакт I1 з'явиться у першому сегменті, а зелений квадрат зміститься праворуч. Щоб послідовно за прямим контактом установити інверсний, необхідно клацнути лівою клавішею 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 прямую котушку Q, позначивши у вікні Edit Contact/Coil індекс змінної -2 і її тип ”-(”:

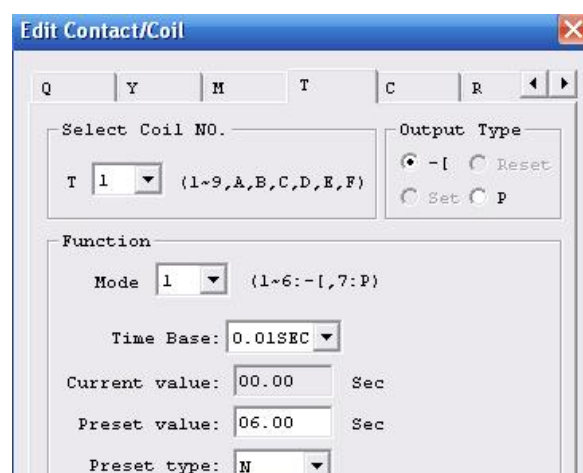
Створити таку саму послідовність контактів у другому сегменті програми, тільки прямому контакту призначити символічне ім'я I2, а інвертованому – i1. За допомогою кнопки  у панелі інструментів приєднати праву сполучну лінію інвертованого контакту i1 до першого сегменту програми, забезпечивши таким чином 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:




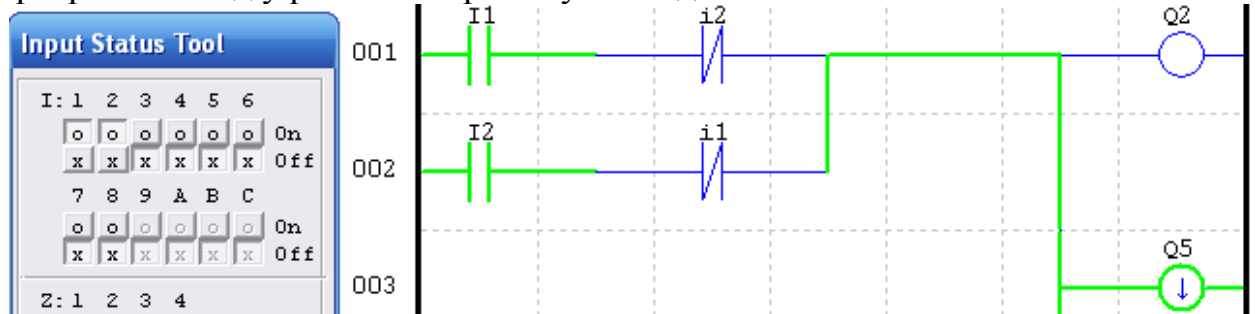


При установленні таймера скористатися в панелі інструментів кнопкою . Відкриється діалогове вікно Edit Contact/Coil, в якому призначити відповідний індекс для таймера T і час затримки Preset value: 06.00sec:




Решту параметрів залишити не змінними.

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



Перемикаючи лівою клавішею 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 клавішу миші на Link Com Port. У вікні даного пункту вибрати COM1 PORT і натиснути клавішу Link. Для завантаження програми у контролер, увійти в меню Operation → Write, а щоб запустити її необхідно натиснути лівою клавішею миші кнопку  в панелі інструментів.

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

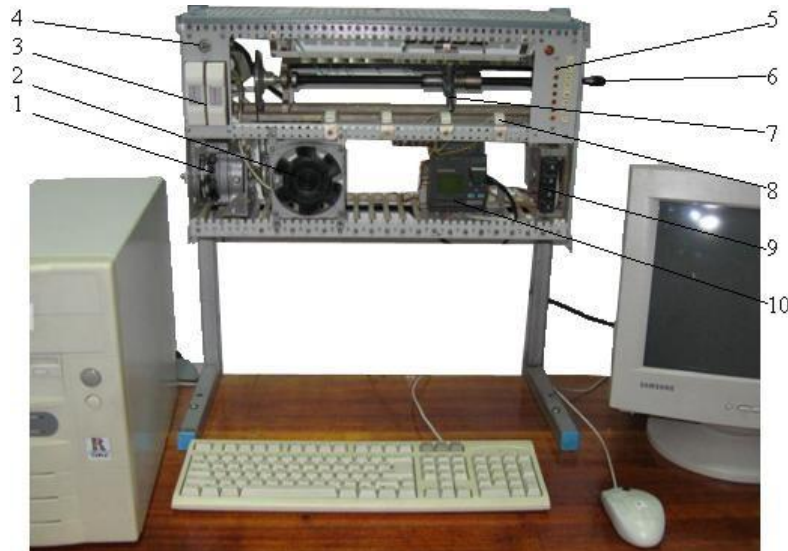
1. Яка конструкція контролера?
2. Якими мовами програмується контролер DURUS?
3. Яке призначення мають клавіші на лицевій панелі DURUS?
4. Як здійснюється програмування контролера з EOM?
5. Що містить бібліотека контролера?
6. Які можливості має пакет програмування Durus Development Software?
7. Для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

Лабораторна робота №2

Програмування та дослідження роботи контролера Logo!Basic.

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

Лабораторна установка, рис.2.1, 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 2.1 – 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 $\approx 24\text{В}$ постійного струму.

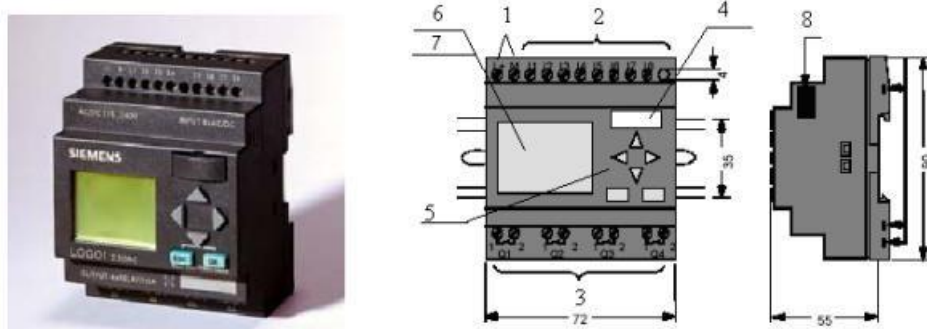
При досягненні кареткою крайнього положення спрацьовують кінцеві вимикачі, які зупиняють двигун. У крайньому лівому положенні каретки вмикається вентилятор.

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.2.2. Контролер має 8 дискретних входів, 4 дискретних виходи, вбудований рідкокристалічний дисплей і клавіатуру, 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  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 ЕОМ; 5 - панель управління; 6 - рідкокристалічний дисплей; 7 - індикатор стану RUN/STOP; 8 - інтерфейс розширення.

Рисунок 2.2 – Контролер LOGO! Basic

Для підтвердження вибраного сполучного елемента або блока використовується клавіша ОК. 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 ЕОМ 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 програмі.

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

При програмуванні контролера 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 встановлено «0», то програма змінює його на логічну «1». Список деяких основних функцій наведено у табл.2.1.

Таблиця 2.1 Основні функції – GF

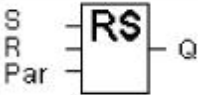
Зображення на комутаційній схемі	Зображення в LOGO!	Найменування
Послідовне з'єднання замикаючих контактів. Якщо який-небудь вхід цього блоку не підключений (х), то для цього входу $x=1$. 		AND (І)
Паралел. з'єднання замикаючих контактів. Якщо який-небудь вхід цього блоку не використовується (х), то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

Спеціальні функції – SF містять таймери, лічильники і інші функціональні блоки. Вони можуть зберігати і мають можливості параметризації (Rem). Опис деяких спеціальних функцій наведено у табл.2.2.

Таблиця 2.2 Спеціальні функції – SF

Символ в LOGO!	Підключення	Опис
Затримка вмикання і запам'ятовування За вхідним імпульсом починається відлік встановленого при параметризації інтервалу часу, після закінчення якого вихід встановлюється в 1.	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.
Реверсивний лічильник При отриманні вхідного імпульсу внутрішнє	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.

значення лічильника, залежно від параметризації, збільшується або зменшується на 1. Досягнувши заданих порогових значень вихід встановлюється або скидається. Напрямок рахунку може бути змінений сигналом на вході Dir.	Параметр	Оп: поріг вмикання Діапазон значень 0.999999 Оф: поріг вимкнення Діапазон значень 0.999999 Збереження для внутрішнього розрахункового значення Cnt: /= Збереження відсутнє R = Стан зберігається.
	Вихід Q	Q - встановлюється або скидається залежно від поточного значення Cnt і встановлених порогових значень.

	Вхід S	Вихід Q встановлюється сигналом на вході S
	Вхід R	Скидання виходу Q встановлюється сигналом на вході R. Якщо S та R = 1, вихід Q скидається.
	Параметр	Збереження: /= збереження відсутнє R = стан зберігається
	Вихід Q	Вихід Q встановлюється сигналом на вході S та скидається сигналом на вході R

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

Виконання лабораторної роботи

Запрограмувати керування рухом 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:
 - таймер із «затримкою вмикання та запам'ятовуванням»;
 - блок «RS»;
 - блок логічний «АБО»;
 - блок логічний «І».

- входи контролера:

1-І4, до яких підключені герконові контакти відповідних датчиків позиції каретки 8;

- І5 - І8, 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:

No Program
Press ESC

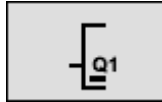
якщо ні, тоді:

>Program..
Card..
Clock..
Start

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 OK. LOGO! перейде в режим програмування >Edit.

Для створення FBD-програми вибрати «Edit Prg» і натиснути OK. На дисплеї LOGO! з'явиться перший вихід контролера Q1.



Після цього можна вводити FBD-програму керування роботою каретки за заданим алгоритмом, рис.2.3.

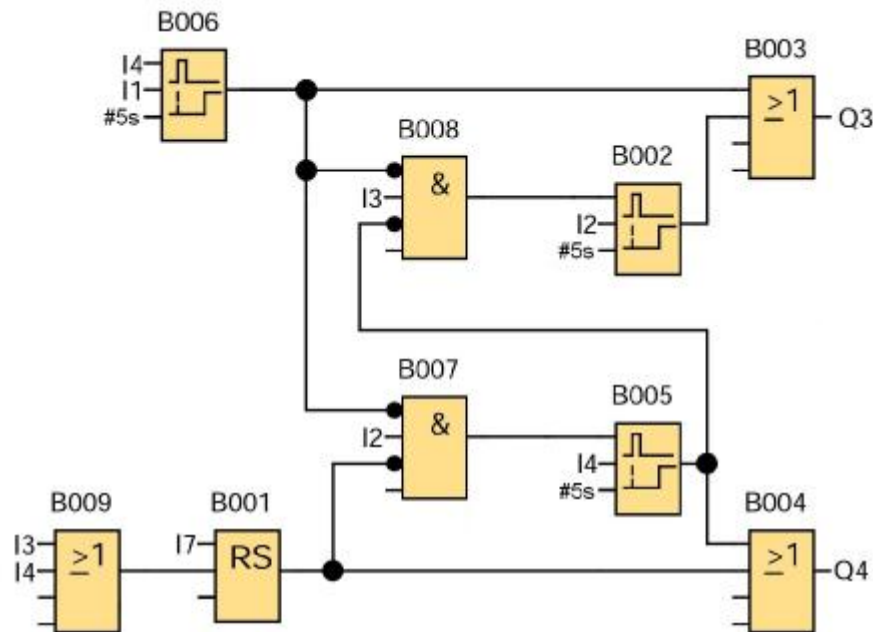


Рисунок 2.3 – FBD-програма керування роботою каретки

Для початку руху каретки необхідно встановити блок «реле з блокуванням(RS)», який при подачі сигналу на його вхід вмикає двигун. Для цього:

- кнопкою OK перейти у режим вводу, курсор отримає вигляд миготливого суцільного прямокутника;
- кнопкою ▼ знайти список SF (спеціальні функції) і натиснути OK, LOGO! відобразить перший блок зі списку спеціальних функцій;
- кнопкою ▲ або ▼ вивести на дисплей блок RS;
- натиснути OK для підтвердження вибору. За замовчуванням блок отримає номер B1.

Після створення блоку курсор у вигляді знака підкреслення знаходиться на першому вході RS - S. Натиснути ОК і кнопкою ▲ або ▼ вибрати список Co. Натиснути ОК і кнопкою ▲ або ▼ вибрати вхід I7. Натиснути ОК, перейти на другий вхід – R і за аналогією з I7 вибрати список BN та встановити B9. При чому встановити його можливо тільки якщо блок B9 вже оголошений.



Третій вхід залишається без зміни.

Після завершення параметризації блока натиснути кнопку Esc і в меню Edit Prg кнопкою перейти на вихід контролера Q2, натиснути ОК.

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

- 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»;

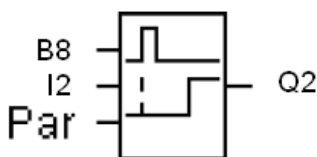
- натиснути ОК і блок автоматично отримає номер B2.

Курсор у вигляді знака підкреслення знаходиться на першому вході Trg.

Для підключення входу Trg блока таймера B2:

- натиснути ОК;
- користуючись ▼ або ▲, вибрати список BN і натиснути ОК;
- за допомогою клавіші ▼ вибрати B8 і натиснути ОК.

Другий вхід R сконфігурувати з I2 (для скидання таймера).



Встановити на третій вхід Par значення часу затримки 5с.

Для цього:

- перейти на вхід Par і натиснути ОК;
- враховуючі те, що формат часу представлений у вигляді «секунди: мілісекунди», клавішею ► перевести курсор на друге знаомісце і клавішею ▲ встановити значення часу - 5 с.

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

Для подачі на вихід контролера Q3 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 В3.

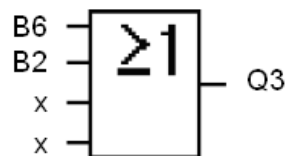
Щоб сконфігурувати перший вхід блока В3 з виходом блока В6 треба натиснути ОК;

- кнопкою ▼ або ▲ вибрати список створених блоків ВN і натиснути ОК;

- кнопкою ▼ вибрати блок В6 і натиснути ОК.

Сконфігурувати другий вхід з виходом В2.

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 подачі керуючого сигналу на вихід контролера Q4, за аналогією з В3 блоком АБО, створити ще один блок логічного АБО. Даному блоку автоматично присвоюється номер В4.

На вільному виході Q5, за аналогією з В2 таймером «із затримкою вмикання і запам'ятовуванням», створити другий таймер, який автоматично отримає номер В5, і сконфігурувати його входи. На перший вхід блока встановити В7, на другий – І4, а на вході Par встановити 5с. Вихід лічильника В5 сконфігурувати з першим входом блока В4.

Для цього:

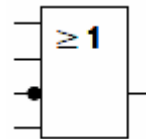
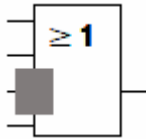
- натиснути кнопку Esc і в меню Edit Prg кнопкою ▼ знайти блок В4;

- 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 В4, вибрати таймер «із затримкою вмикання і запам'ятовуванням» з виходом Q6. Йому автоматично присвоїться номер В6. На перший вхід встановити І4, на другий І1, а час затримки - 5с.

Зверніть увагу на те, що блоки В7 та В8 мають інверсні входи. Щоб встановити такий тип входу треба при конфігуруванні входів блоку встановити курсор на межу блока та входу(у графічному представленні). Кнопками ▼ та ▲ змінити тип входу.



Завершити конфігурування інших блоків самостійно.

Після завершення конфігурування тричі натиснути Esc, перевести курсор на «Start» і натиснути OK.

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 OK.

Програмування у пакеті LOGO! Soft Comfort

Відкрити програму LOGO! Soft Comfort V5.0 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 EOM з'являться два поля - панель інструментів і робоче поле проекту.


Встановити на робочому полі блок RS. Для цього:

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

- курсором вибрати блок RS  і натиснути ліву клавішу миші;

- повторно клацнути лівою клавішею миші на робочому полі, де має бути встановлено цей блок. Йому автоматично присвоюється номер B001.


Для конфігурування входів блока B001 з входом контролера I7, необхідно:

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

- зі списку вибрати блок «I» і встановити його на робочому полі, натискаючи лівою клавішею миші спочатку на вибраному елементі, потім у місцях де має бути встановлено 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 та на вкладці «Емуляція» обрати тип перемикача «Кнопка без фіксації».

Щоб з'єднати вхід R блока B001 з входом контролера I7:


- натиснути кнопку , що зліва на панелі інструментів, потім клацнути лівою клавішею миші на вході I7 і не відпускаючи її підвести курсор до входу R B001.

Для встановлення блока АБО:

- зліва на панелі інструментів натиснути кнопку GF і на робочому

полі встановити блок , якому автоматично надається номер. Входа приєднуються за аналогією до блоку B001. Щоб змінити тип входу на інверсний, треба двічі клацнути лівою кнопкою миші по входу.


Для підключення програми до виходу контролера Q4, який забезпечує рух каретки вправо, 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 відповідного блоку.

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. Номер блока присвоюється автоматично.

- виконати конфігурування входів та виходів відповідно схемі;
- двічі клацнути лівою клавішею миші на таймері, відкриється діалогове вікно, в якому встановити час затримки вмикання - 5с.

Аналогічно на робочому полі встановити ще 2 таймери.

Загальний вигляд програми керування рухом каретки наведено на рис.2.4.

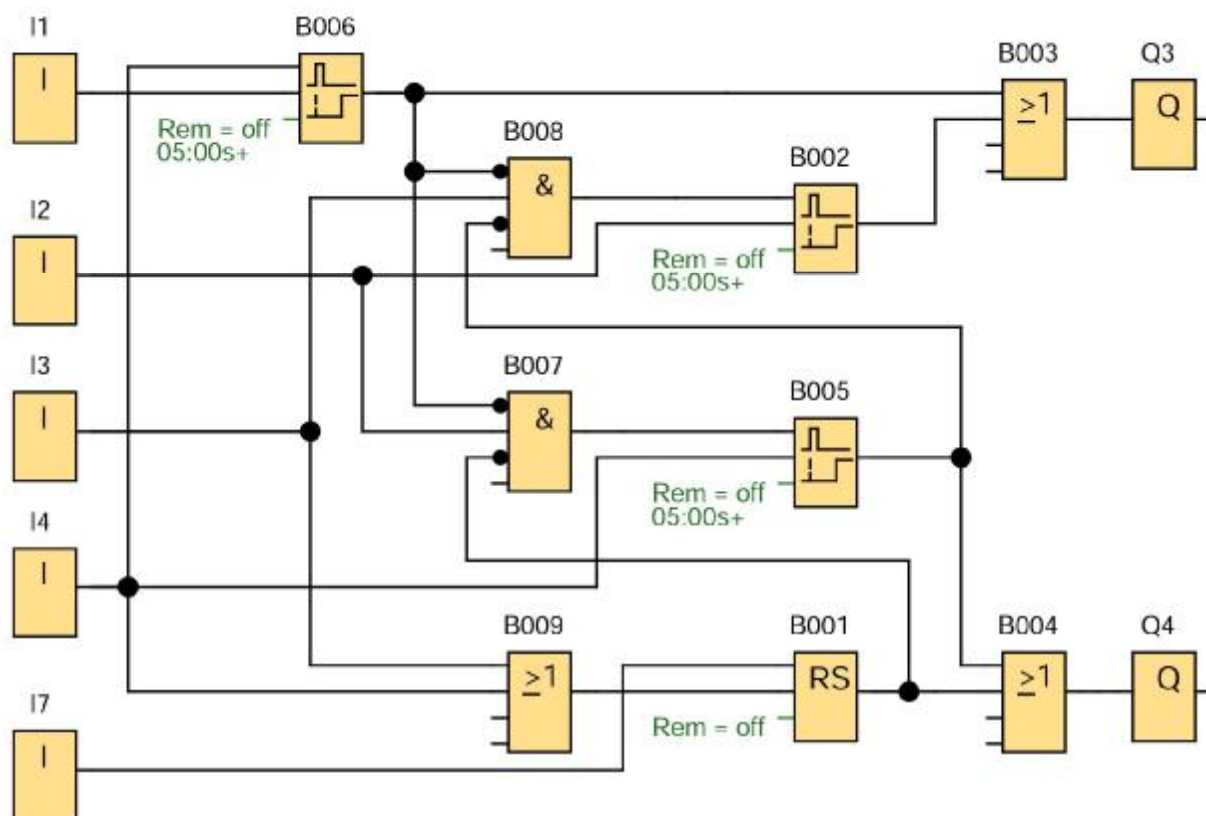



Рисунок 2.4 – Програма керування рухом каретки

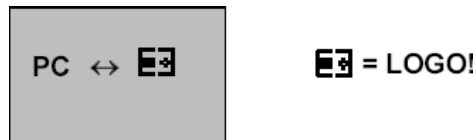
Для перевірки роботи програми в панелі інструментів натиснути кнопку «Емуляція» . Активізуючи входи контролера у програмі, за допомогою їх символів, що унизу робочого поля, і лівої клавіші миші, переконатися в правильності виконання завдання. При цьому сполучні лінії між блоками відображаються різними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

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

Для запуску програми необхідно 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.

Для зупинення програми необхідно на дисплеї контролера перевести курсор у позицію «Stop» і натиснути ОК та підтвердити операцію, обрати «Yes» та натиснути «ОК».

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



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 рухів каретки від позиції до позиції.

Зміст звіту

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

Контрольні запитання

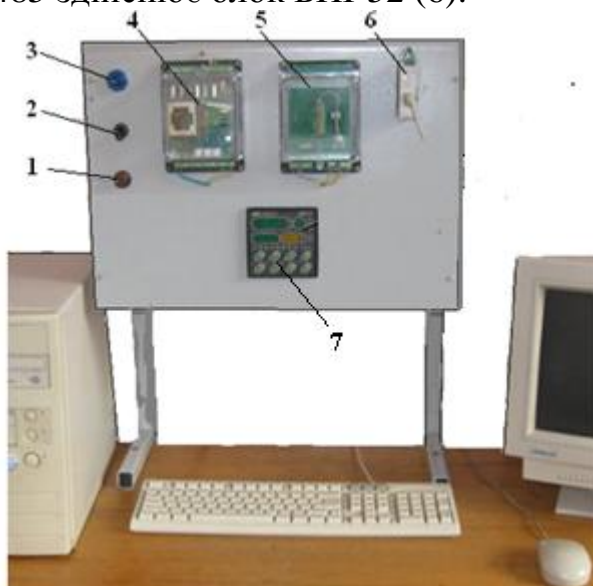
1. Для чого призначений контролер LOGO!?
2. Яка конструкція контролера?
3. Якими мовами програмується контролер LOGO!?
4. Яке призначення мають клавіші на лицевій панелі LOGO!?
5. Як здійснюється програмування контролера з ЕОМ?
6. Що містить бібліотека контролера?
7. Чим характеризуються логічні елементи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

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.3.1, призначена для програмування та дослідження роботи контролера МІК-51. У її складі контролер МІК-51 (7), який, відповідно до програми користувача за ПІД-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).



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 3.1 – Лабораторна установка

Контролер МІК-51

Контролер МІК-51, вироблений на підприємстві «Мікрол», м. Івано-Франківськ, 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.

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 - налаштування кількості блоків.

Другий рівень - програмування функціональних блоків.

Третій рівень - програмування зв'язків функціонального блока.

Четвертий рівень - налаштування параметрів функціональних блоків.

П'ятий рівень - тестування і калібрування входів-виходів

Шостий рівень - системні параметри контролера.

Сьомий рівень - контроль відмов.

Восьмий рівень - збереження програми користувача і налаштувань в енергонезалежній пам'яті.

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

Зміна значень виведених на різні дисплеї (цифрові індикатори) представлено на рис.3.2:

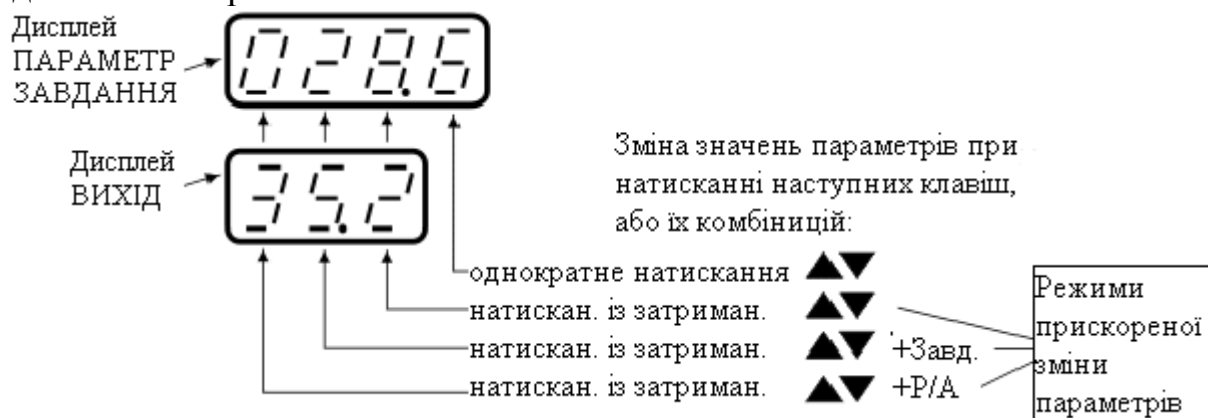


Рисунок 3.2 – Зміна значень на різних дисплеях

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 клавішу Р/А.

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

Для переходу в автоматичний локальний режим керування необхідно на лицевій панелі двічі натиснути клавішу 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 [Р/А] 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 лицевої панелі приладу.

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

- Для зміни уставки, МІК-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 ВИ←ІД при аналоговому вихідному сигналі, або світлодіодні індикатори ▲ чи ▼ - при дискретному.

Виконання лабораторної роботи

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

2. Запрограмувати за допомогою інструментальної системи «Альфа» і дослідити роботу контролера, який керує виконавчим механізмом МЕО за ПІД-імпульсним законом.

Завдання 1

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.3.1.

Таблиця 3.1 Програмування функціональних блоків

Алгоритми (2-й рівень)					Конфігурація (3-й рівень)					Настроювання (4-й рівень)		
№ 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 з лицевій панелі

1. Подати напругу живлення на контролер МК-51.
2. Перевести контролер у режим РАБОТА (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 дисплей 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-й і 2-й розряди дисплея ПАРАМЕТР. Знову натиснути кнопку ”Ввід” і обнулити 3-й і 4-й 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 задатчиків - 0000, кількість панелей користувача - 000.

7. Установити другий рівень - «Програмування функціональних блоків» і відповідно до табл.3.1 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.3.2.

Таблиця 3.2 Програмування функціонального блока АІN(05)

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

Аналогічно виконати програмування функціональних блоків РІD(60) і АОТ(08), користуючись табличним записом програми, табл.3.1.

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 РІD-блока приведено у табл.3.3.

Таблиця 3.3 Програмування зв'язків для блока РІD

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

Аналогічно, користуючись табличним записом програми, табл.3.1, 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.3.1.

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

У функціональному блоці РІD 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 АІN наведений у табл.3.4.

Таблиця 3.4 Настроювальні параметри алгоблока АІN

Дисплей «№» КАНАЛ, Номер рівня	Дисплей ПАРАМЕТР		Дисплей ЗАВДАННЯ	Дисплей ВИХІД
	1,2-й розряд	3,4-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. Тому п'ятий рівень програмування можна не виконувати.

10. Встановити шостий рівень – «системні параметри» і занести відповідні данні мережного обміну: мережну адресу, наприклад 10 та швидкість мережного обміну – 0009 (115200біт/с), табл.3.5.

Таблиця 3.5 Системні параметри

Дисплей «№» КАНАЛ, Номер рівня	Дисплей ПАРАМЕТР		Дисплей ЗАВДАННЯ	Дисплей ВИХІД
	1,2-й розряд	3,4-й розряд		
6	02	01	10	000
		02	0009	000

Сьомий рівень – «контроль відмов» не програмується.

11. 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.3.6.

Таблиця 3.6 Збереження програми користувача

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

12. Перевести контролер у режим РОБОТА (за допомогою комбінації клавіш [↺] Меню +[№ Кн ↑]), переконавшись, що немає помилок і відмов (індикатор ОШ не світиться) і кнопкою "Меню" 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 ВИ←ІД– значення вихідного сигналу регулятора.


13. 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.

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

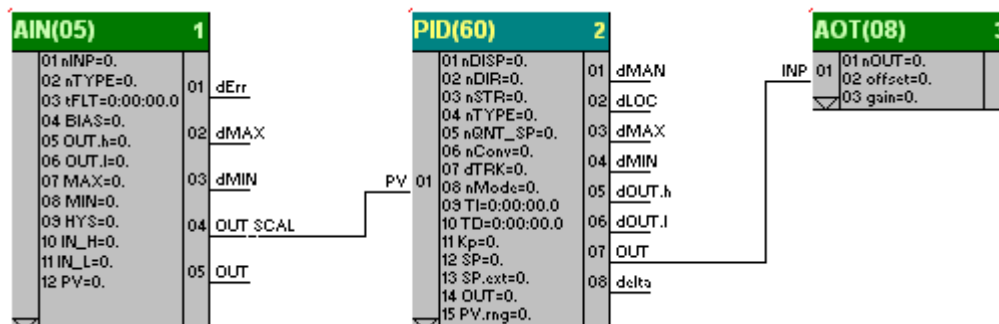
Програмування за допомогою інструментальної системи «Альфа»

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

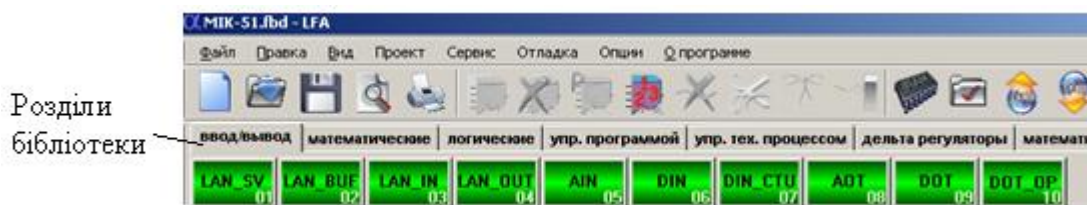
2. Запустити програму «Альфа» використовуючи на ЕОМ меню «Пуск – > Все программы» -> «Альфа».

3. Для створення проекту, у головному меню екрана вибрати «Файл» → «Новый» або в панелі інструментів натиснути кнопку «Новый» .

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:



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


Сполучення блоків здійснюється за допомогою миші:


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


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


Для налаштування 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 налаштування, користуючись табличною формою програми, табл.3.1.

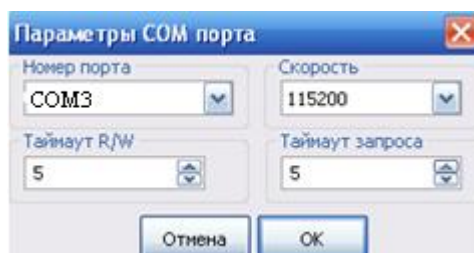
Нумерація блоків виконується в порядку їх установлення. При необхідності можна змінити нумерацію автоматично, за допомогою кнопки  (Пронумерувати блоки автоматично) на панелі інструментів. При цьому виконується послідовна нумерація блоків у порядку їх установлення;


Розподіл параметрів блока в реєстровій області пам'яті здійснюється автоматично за допомогою кнопки  (Автоматичний розподіл пам'яті) на панелі інструментів;


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


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

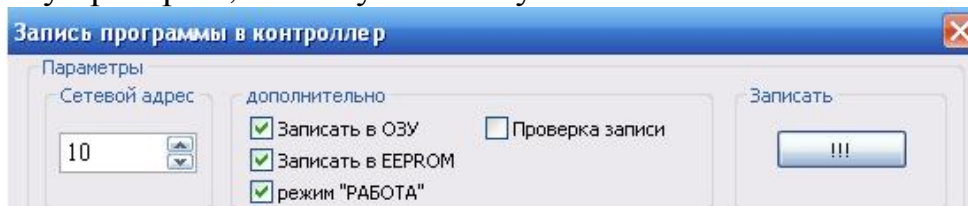
За допомогою кнопки  на панелі інструментів визначити параметри COM-порту:



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

Натиснути кнопку , у діалоговому вікні «Компиляция проекта» активізувати усі опції і натиснути ОК.

Повторно натиснути кнопку , у діалоговому вікні «Запись программы в контроллер» активізувати необхідні пункти запису і виконання програми та, указавши адресу пристрою, натиснути кнопку «Записать».



Після завершення процесу, на дисплеї контролера висвітляться символи «ЗП». Коли це повідомлення зникне, контролер почне регулювання температури.

За показаннями дисплеїв ПАРАМЕТР і ВИ←ІД відслідковувати роботу системи автоматичного регулювання.

Завдання 2

За допомогою інструментальної системи «Альфа» запрограмувати і дослідити роботу контролера, який керує виконавчим механізмом МЕО за ПІД-імпульсним законом. Вхідний сигнал регулятора програмується відповідно до графіка на рис 3.3 за допомогою функціональних блоків ТМ_М(56), ТМ_PRG(57) і SP_М(54), а керувальний вплив на виконавчий механізм формує функціональний блок PID_IMP(62), використовуючи блоки дискретного

виводу DOT(09) та дискретної уставки dSET(83). TM_PRG(57) формує кусково-лінійну функцію часу, яка складається з відрізків, тривалість яких і кінцева ордината визначаються функціональними блоками, відповідно TM_M(56) і SP_M(54).

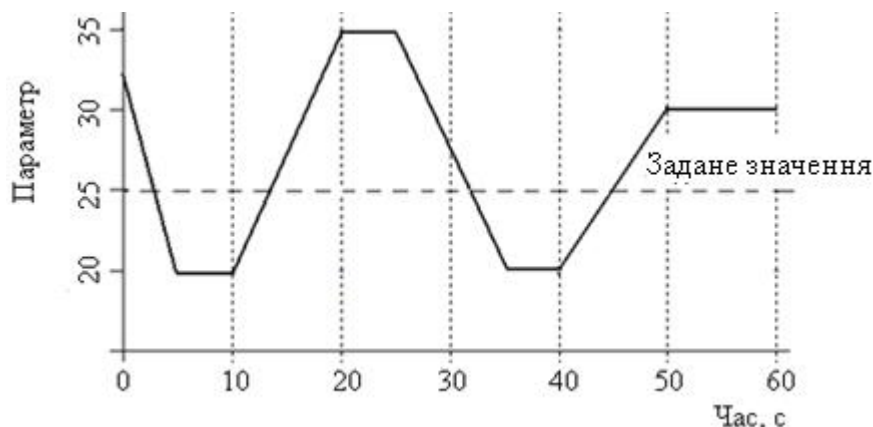


Рисунок 3.3. – Траєкторія вхідного сигналу регулятора відносно заданого значення

Лабораторна установка для програмування і дослідження роботи контролера приведена на рис.3.4.



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.

Рисунок 3.4– Лабораторна установка

Створення проекту

Табличний варіант програмного коду приведено у табл.3.7, користуючись відповідними розділами бібліотеки і описом функціональних блоків у керівництві по експлуатації МІК-51.

Таблиця 3.7

Програма керування роботою виконавчого механізму

Алгоритми (2-й рівень)					Конфігурація (3-й рівень)					Настроювання (4-й рівень)		
№ блока	Шифр	№ алгоритму	Базова адреса	Модифікатор	№ блока	№ входу	№ блока	№ виходу	Ім'я входу	№ параметра	Ім'я параметра	Значення
1	2	3	4	5	6	7	8	9	10	11	12	13
01	TM_M	56	0000	01						01	TM1	0:00:05
										02	TM2	0:01:00
										03	TM3	0:00:10
										04	TM4	0:01:00
										05	TM5	0:00:10
										06	TM6	0:01:00
										07	TM7	0:00:10
										08	TM8	0:01:00
										09	TM9	0:00:00
										10	TM10	0:00:00
										11	TM11	0:00:00
										12	TM12	0:00:00
02	TM_PRG	57	0012	01	02	01	07	01	eSTART	01	nDISP	1
					02	02	02	02	eSTOP	02	qREP	99
					02	03	07	01	dRESET	03	qSTEP	6
										04	B.T1	1
										05	B.SP1	3
										06	B.T2	0
										07	B.SP2	0
										08	B.T3	0
										09	B.SP3	0
										10	B.T4	0
										11	B.SP4	0

1	2	3	4	5	6	7	8	9	10	11	12	13
										12	nMODE	3
										13	nSTEP	6
										14	TIME	0:00
										15	SP0	33
03	SP_M	54	0027	01						01	SP1	20
										02	SP2	20
										03	SP3	35
										04	SP4	35
										05	SP5	20
										06	SP6	20
										07	SP7	30
										08	SP8	30
										09	SP9	0
										10	SP10	0
										11	SP11	0
										12	SP12	0
04	PID_IMP	62	0040	01	04	01	02	09	PV	01	nDISP	1
										02	nDIR	0
										03	nSTR	1
										04	nTYPE	0
										05	SP_Q	1
										06	nCONV	0
										07	dTRK	0
										08	nMODE	1
										09	tCTRL	0:00:22
										10	tMIN	0:00:02
										11	tDELAY	0:00:01
										12	TI	0:00:10
										13	TD	0:00:02

1	2	3	4	5	6	7	8	9	10	11	12	13
										14	KP	1
										15	SP	25
										16	SP_EXT	30
										17	OUT	0
										18	PV_R	60
										19	SP_H	100
										20	SP_L	0
										21	SP_R	0
										22	POZ	2
										23	TRK_V	0
										24	MAX	40
										25	MIN	15
										26	HYS	0
										27	FF_V	0
1	2	3	4	5	6	7	8	9	10	11	12	13
										28	FF_G	0
										29	FF_H	0
										30	FF_L	0
										31	COR_V	0
										32	COR_G	0
										33	COR_H	0
										34	COR_L	0
										35	FB	0
05	DOT	09	0075	01	05	01	04	05	dINT	01	nOUT	1
06	DOT	09	0076	01	06	01	04	06	dINT	01	nOUT	2
07	dSET	83	0077	01						01	dSET	0

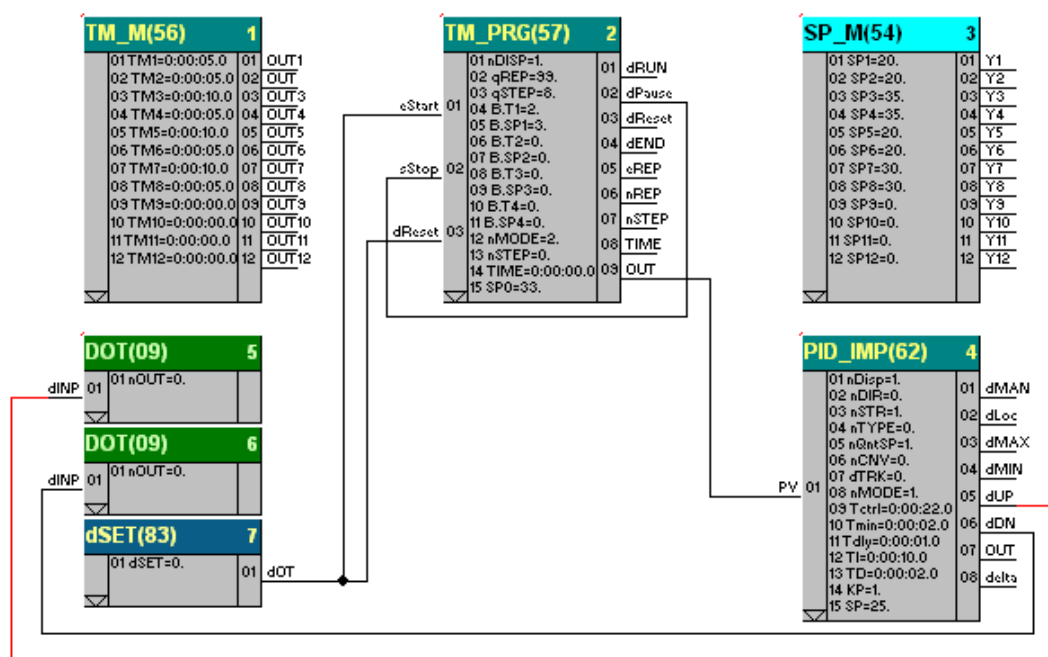
1. Подати напругу живлення на контролер МІК-51.

2. Перевести контролер у режим «Програмування», натиснути кнопку "Ввід" і клавішами ▲,▼ обнулити 1-й і 2-й розряди дисплея ПАРАМЕТР. Знову натиснути кнопку "Ввід" і обнулити 3-й і 4-й розряди дисплея



ПАРАМЕТР. Натиснути кнопку "Ввід" і обнулити розряди решти цифрових дисплеїв (після кожного обнуління результат підтверджується натисненням клавіші "Ввід").

3. Запустити програму «Альфа» і створити новий проект.

Програмний код алгоритму, за яким здійснюється дослідження динаміки ПІД-імпульсного регулятора, має наступний вигляд:



Здійснити налаштування параметрів функціональних блоків, користуючись табл.3.7.

Перевірити програму на відповідність існуючим правилам за допомогою кнопки  і виконати налагодження кнопкою .

За допомогою кнопки  в панелі інструментів визначити параметри COM-порту:

Параметры COM порта


Номер порта: COM3


Скорость: 115200

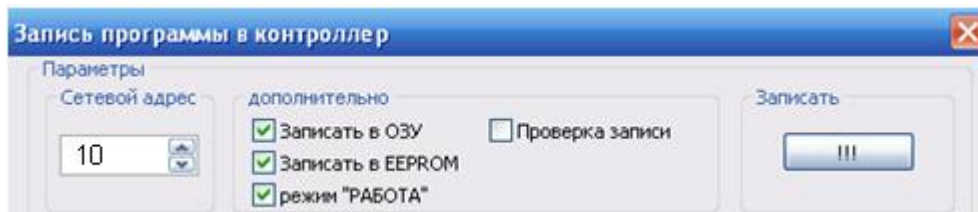
Таймаут R/W: 5

Таймаут запроса: 5

Отмена OK

Натиснути кнопку  і в діалоговому вікні «Компіляція проекту», активізувати усі опції, натиснути ОК.

Повторно натиснути кнопку  (Записать программу в контроллер), в діалоговому вікні зробити наступні установки і натиснути кнопку «Записать».



Після завершення процесу, на дисплеї контролера висвітляться символи «ЗП». Коли це повідомлення зникне, контролер почне керувати.

Щоб зупинити виконання програми, кнопкою "Меню" вибрати панель індикації "Прг", кнопкою "Р/А", перевести індикатор на позицію "Стоп" (червоний квадрат миготить) і натиснути кнопку "Ввід".

Для відновлення виконання програми кнопкою Р/А перевести індикатор на позицію «Програмний задавач працює» (зелений трикутник миготить). Кнопкою "Ввід" підтвердити дану команду і за допомогою дисплеїв контролера порівняти характер зміни вхідного сигналу регулятора з характеристикою, що наведена на рис.3.3. Кнопкою «Меню» перейти на індикатор «Рег» і відслідковувати виконання програми на дисплеях контролера та роботу виконавчого механізму МЕО.

Завдання для самостійної роботи

Познайомитися з описом функціональних блоків ТМ_М(56) і SP_М(54) і розробити програму керування температурою в печі за часовою діаграмою.

Зміст звіту

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

Контрольні запитання

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»?

Лабораторна робота №4

Програмування і дослідження роботи контролера Modicon TSX Micro

Мета: Ознайомитися з конструкцією, набути навичок програмування і роботи з контролером.

Дана лабораторна робота є комп'ютерною і призначена для вивчення конструкції і дослідження можливостей роботи контролера Modicon TSX Micro виробництва Schneider Electric. В якості об'єкта автоматизації використовується мнемосхема, що створена у середовищі PL7 PRO, на якій імітується технологічний процес дроблення породи і пульт керування, рис.4.1. При виконанні лабораторної роботи необхідно розробити програму керування роботою дробарки і налагодити її за допомогою візуалізованих об'єкта автоматизації та пульта управління.

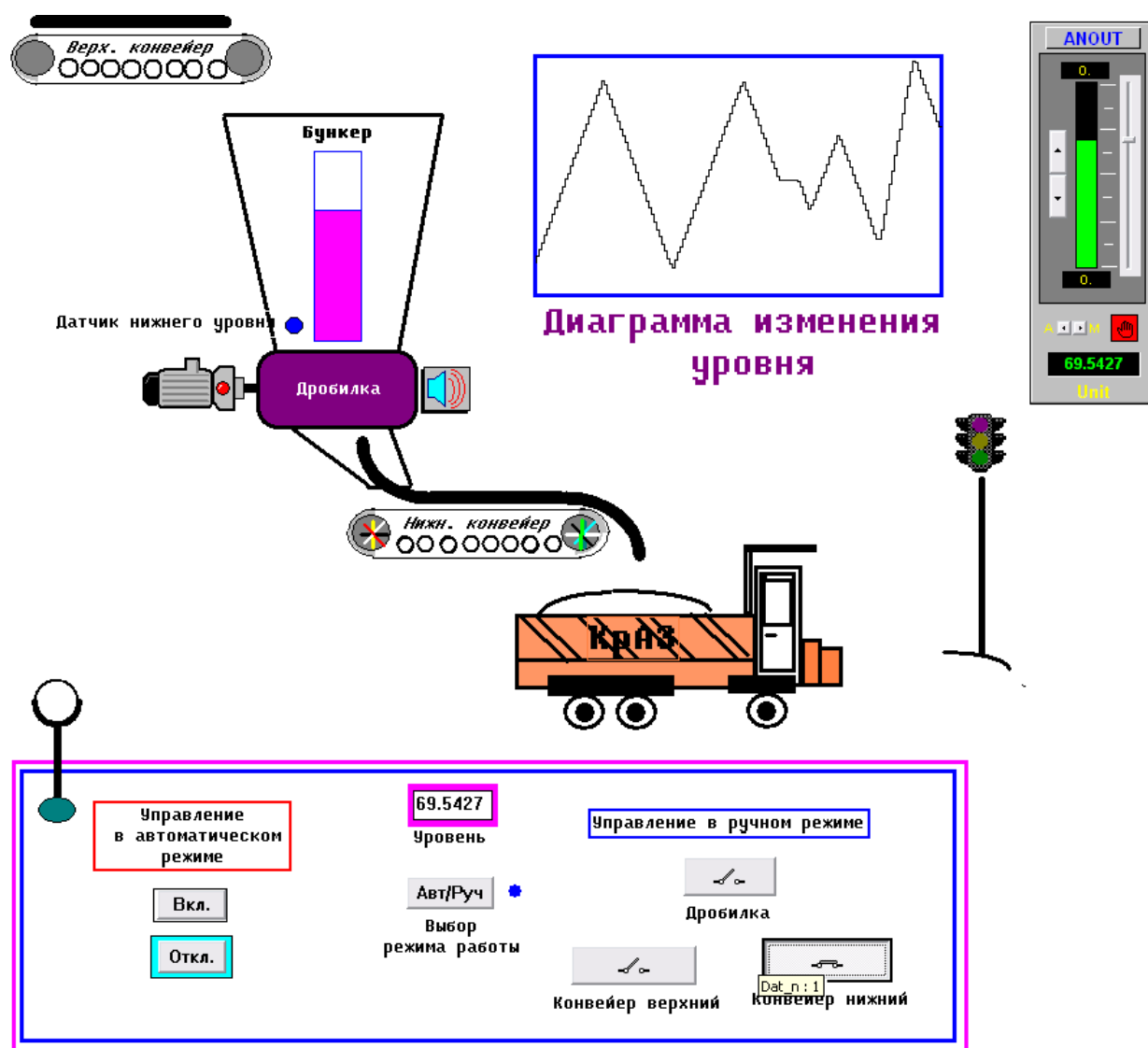


Рисунок 4.1– Візуалізація технологічного процесу дроблення породи і пульта управління

Контролер Modicon TSX Micro

Контролер Modicon TSX Micro призначений для керування об'єктами, для роботи з якими достатньо кілька сотень дискретних входів-виходів та до сорока аналогових входів-виходів у різних галузях промисловості.

Програмований логічний контролер Modicon TSX Micro випускається в декількох модифікаціях: TSX 3705, TSX 3708, TSX 3710 и TSX 3721/22.

ПЛК TSX 3705 і TSX 3708 мають шасі з вбудованим блоком живлення на 100/240 В змінного струму, процесор з об'ємом пам'яті 11 Кслів (програма, дані і константи), 1 резервну флеш-СППЗУ, 1 і 2, відповідно, модулі дискретного вводу/виводу TSX DMZ 28DR (16 входів і 12 релейних виходів), а також один вільний слот. У вільний слот можна встановити: 1 модуль дискретного вводу/виводу будь-якого типу стандартної висоти; 2 модулі дискретного вводу/виводу, захисту, аналогового вводу/виводу або рахунку половинної висоти.

ПЛК TSX 3721/22 містять шасі на 3 слоти з вбудованим блоком живлення (24 В DC або 100/240 В AC), процесор з ОЗУ на 20 Кслів (програма, дані і константи), 1 резервне флеш-СППЗУ, 2 слоти для PC-карт (1 карта зв'язку і 1 карта розширення пам'яті об'ємом не більше 64 Кслів) і годинник реального часу. Компактне шасі розширення TSX RKZ 02 дозволяє збільшити кількість слотів на 2 (4 установчих місця). ПЛК TSX 37-21/22 можна підключити до мережі Ethernet TCP/IP або модему за допомогою зовнішнього автономного модуля TSX ETZ 410/510.



1-базове шасі на два слоти; 2- центральний індикаторний блок; 3- порт для терміналу (TER) (протоколи Uni –Telway або Modbus); 4- кришка клем блока живлення; 5- кнопка скидання; 6- кришка відсіку для додаткової батареї; 7- модуль дискретного вводу/виводу на 28 або 64 канали; 8- вільний слот.

Рисунок 4.2–Програмований логічний контроллер TSX 3710

Програмований контроллер TSX 3710, рис.4.2, який використовується в лабораторній роботі, містить шасі з вбудованим блоком живлення 24В постійного струму (є модифікації з блоком живлення на 100/240В змінного струму), процесор з ОЗУ на 14 Кслов (програма, дані і константи), флеш-СППЗУ об'ємом 16 Кслов для резервного копіювання прикладної програми, годинник реального часу, 28-канальний модуль дискретного вводу/виводу

TSXDMZ 28DT і вільний слот зі встановленими на 3-му і 4-му посадочних місцях, відповідно, дискретними 12-канальним модулем входу TSXDEZ 12D2 і 8-канальним модулем виходу TSXTSZ 08T2.

Всі модифікації ПЛК Мікро мають інтегровані засоби зв'язку, що забезпечують обмін даними через термінальний порт RS 485 за протоколами Uni-Telway (провідний/підпорядкований), Modbus (підпорядкований) або по ASCII- протоколу. Через цей же порт за допомогою EOM програмується контролер.

Modicon TSX Мікро має багатозадачну операційну систему, яка дає можливість швидко відпрацьовувати декілька подій тих, що надійшли в прикладну програму. Контролер TSX Мікро витрачає не більше 0.15μ на виконання однієї логічної операції.

ПЛК Мікро оснащені індикаторним блоком, який забезпечує відображення всіх даних, необхідних для контролю, діагностики і техобслуговування ПЛК і всіх його модулів.

Центральний індикаторний блок TSX 3710 містить три групи по 32 світлодіоди, які відповідають слотам модулів, що встановлені на шасі базового блока або компактного шасі розширення; інформаційний рядок зі світлодіодними індикаторами режиму роботи індикаторного блока; кнопку управління, що забезпечує доступ до різних режимів роботи і п'ять світлодіодів: RUN (робота) – вмикання і вимикання ПЛК;

TER (термінал) – обмін даними через порт для терміналу;

I/O (вхід-вихід) – несправність входу/виходу;

ERR (помилка) – несправність процесора або програми;

BAT (батарея) – несправність або відсутність батареї.

З ПЛК Мікро може працювати більше 40 різних спеціалізованих модулів, включаючи швидкі лічильники, модулі виміру і регулювання, входи/виходи на 24 В постійного. струму або 220 В змінного струму. Модулі можуть бути на різну кількість входів/виходів (8, 16, 32, 64). Дискретні модулі випускаються двох форматів: стандартний, який займає один слот (2 установчих місця) і напівформат, який займає тільки одне установче місце. Решта всіх модулів (аналогові, рахункові і ін.) - напівформати. За допомогою мінішасі розширення, яке може бути безпосередньо підключене до базового шасі, можна збільшити кількість слотів для додаткових модулів.

Проектування і розробка додатків для ПЛК TSX-Мікро виконується на 4 мовах за допомогою програмного забезпечення PL7 PRO:

- сходових діаграм (Ladder Diagram (LD));
- список інструкцій (Instruction List (IA));
- структурований текст (Structured Text (ST));
- Grafset (аналог SFC).

Виконання лабораторної роботи

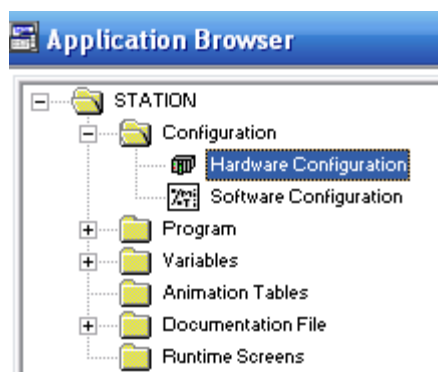
Скласти програму керування роботою дробарки, бункер якої заповнюється пороною верхнім транспортером, а розвантаження її здійснюється на нижній транспортер. Керування роботою дробарки відбувається в

автоматичному або ручному режимах. В автоматичному режимі після дворазового натискання кнопки "Вкл." спрацьовує сирена протягом 5с, яка після 3-секундної паузи повторно вмикається на 5с і після цього починає працювати дробарка. Якщо в бункері є залишки породи, про що свідчить сигнал з датчика нижнього рівня, вмикається нижній конвеєр. Після спорожнювання бункера вмикається верхній конвеєр, а на пульту оператора загоряється сигнальна лампа "Норма". Коли рівень досягає положення верхнього датчика верхній конвеєр вимикається, а коли порода знаходиться на рівні нижнього датчика, верхній конвеєр поновлює свою роботу. В ручному режимі дробарка, нижній та верхній конвеєри вмикаються окремими кнопками на пульту оператора. Вибір режимів роботи здійснюється відповідним перемикачем.

1. Запустити в роботу середовище програмування дворазовим натисканням лівої клавіші миші на логотипі PL7 PRO.

2. Користуючись меню File, у вікні Open позначити проект Object і створити його копію під своїм прізвищем.

3. Запустити свій проект, у його дереві увійти до теки Configuration і відкрити закладку Hardware Configuration. У вікні Configuration вибрати процесор TSX 3710 v.2.0.



4. За допомогою правої клавіші мишки з курсором на Add Module додати в слотах 1 і 2 стандартний модуль дискретного вводу/виводу TSXDMZ 28DT, а в слотах 3 і 4, відповідно, напівформатні модулі дискретного вводу TSXDEZ 12D2 і дискретного виводу TSXTSZ 08T2, що встановлені в контролері Micro.

5. Відкрити теку змінних Variables і у таблицю Memory Objects записати змінні типів EBOOL і REAL, а у таблицю Predefined FB блоки лічильника і таймерів, рис.2, що використовуються в програмі. Оскільки налагодження програми виконуватиметься на візуалізованому об'єкті всі змінні внутрішні %M0-%M17, %MF0.

Variables				
Parameters MEMORY EBOOL Entry field				
Address	Type	Symbol	Comment	
%M0	EBOOL	Drob	Дробилка	
%M1	EBOOL	Dat_v	Датчик верхнего уровня	
%M2	EBOOL	Avt_ruch	Переключатель Авт/Руч	
%M3	EBOOL	Dat_n	Датчик нижнего уровня	
%M4	EBOOL	Sirena	Сирена	
%M5	EBOOL	Norma	Норма	
%M6	EBOOL	Line_2	Минимальная амплитуда отклонения уровня при автоматическом регулировании	
%M7	EBOOL	Line_1	Максимальная амплитуда отклонения уровня при автоматическом регулировании	
%M8	EBOOL	Otkl	Остановить дробилку в автоматическом режиме	
%M9	EBOOL	Vkluch	Включить дробилку в автоматическом режиме	
%M10	EBOOL	Kon_v	Конвейер верхний	
%M11	EBOOL	Kon_n	Конвейер нижний	
%M12	EBOOL		Дополнительная1	
%M13	EBOOL		Дополнительная2	
%M14	EBOOL		Дополнительная3	
%M15	EBOOL		Вкл. дробилку (Ручное)	
%M16	EBOOL		Вкл верх. кон. (Ручное)	
%M17	EBOOL		Вкл. ниж. кон. (Ручное)	

Variables				
Parameters MEMORY REAL Entry field				
Address	Type	Symbol	Comment	
%MF0	REAL	Uroven	Уровень	

Variables						
Parameters PREDEFINED FE C Entry field						
Address	Type	Symbol	Preset	Req	Comment	
%C0	C	Counter	2	<input checked="" type="checkbox"/>	Підпицьник натисків кнопки Vkl	

Variables								
Parameters PREDEFINED FE TM Entry field								
Address	Type	Symbol	Preset	Mode	TB	Req	Comment	
%TM0	TM	Timer0	5	TP	1s	<input checked="" type="checkbox"/>	Двукратное предупреждение сирены	
%TM1	TM	Timer1	3	TOF	1s	<input checked="" type="checkbox"/>	пауза между активизациями сирены	
%TM2	TM	Timer2	13	TON	1s	<input checked="" type="checkbox"/>	Прекращение работы сирены	

Рисунок 2– Таблиці оголошення змінних і функціональних блоків

6. Щоб програму можна було запускати в роботу за допомогою перемикача «Пуск» на лабораторному стенді, та вивести на світлодіод панелі «Сирена» стан змінної Sirena, необхідно у вкладці I/O теки Variables оголосити зовнішні змінні %I3.1 для модуля TSXDEZ 12D2 і %Q4.1 для модуля TSXDMZ 28DT.

7. Увійти до теки Mast і за допомогою правої клавіші відкрити робоче поле Main, вказавши мову програмування LD.

8. Використовуючи панель інструментів LD-редактора, що розташована в нижній частині екрану, зконфігурувати програму управління роботою дробарки, лістинг якої приведено на рис.4.3. Для установлення функціональних блоків, операторів і компараторів використовуються кнопки, відповідно, «F7», «Oper.F3», «Comp.F4». Компаратор призначений для перевірки умов, написаних ST-

мовою з використанням знаків порівняння (<,>,<=,>=,">). Якщо ST-вираз – TRUE компаратор пропускає сигнал, інакше ні. Оператор використовується для виконання однієї ST-інструкції. Відбувається це, коли всі умови перед ним знаходяться у стані TRUE.


Примітки: 1. Створення фрагментів програми відбувається на сітчастих сторінках в режимі редагування (всі елементи секції у червоному кольорі). Щоб вийти з нього і підтвердити те, що зроблено, необхідно перевести курсор на вільне місце робочої сторінки і на клавіатурі натиснути Enter.

2. Для того, щоб додати наступну сторінку для чергового фрагменту програми, необхідно в меню Edit вибрати Insert Rung і підтвердити клавішею Enter.

9. Для перевірки правильності роботи програми необхідно прив'язати оголошені змінні до візуалізованих об'єкта автоматизації та пульта керування. Для цього в дереві проекту клацнути правою клавішею миші на теці Animation Tables, і в таблицю, що з'явиться, записати всі змінні проекту з їх адресами, іменами і коментарями (%MF0, %M0-%M17).

10. Виконати передачу програми в порт комп'ютера, використовуючи в панелі інструментів меню PLC→Transfer Program, і вказати операцію PC→PLC, у вікні, що з'явиться. Підтвердити зроблений вибір у запитанні, що з'явиться у наступному вікні.

11. У панелі інструментів за допомогою меню PLC→Connect з'єднати комп'ютер з контролером і кнопкою RUN в панелі інструментів вибрати режим виконання програми.

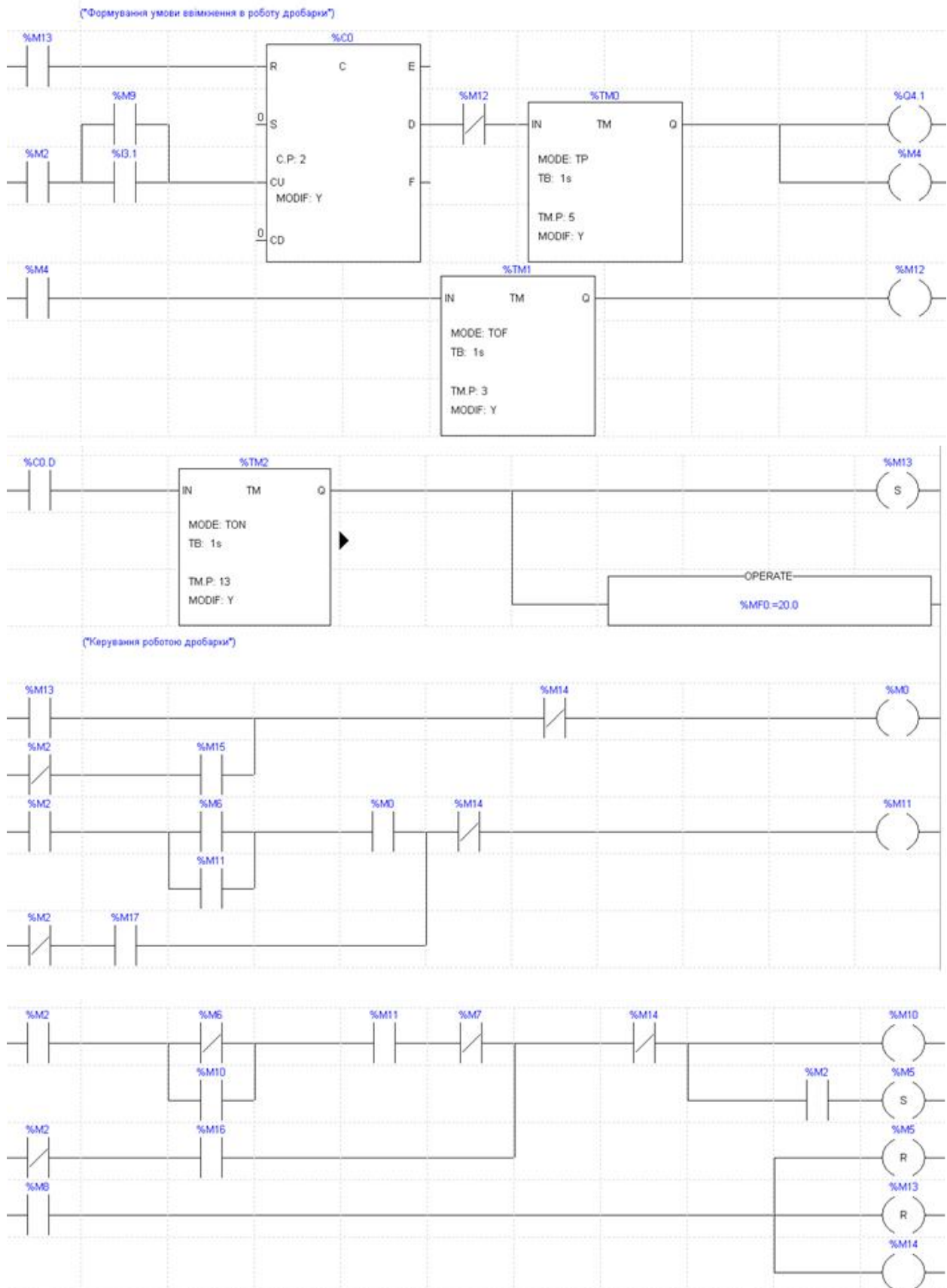
12. У панелі інструментів, користуючись клавішею  Application Browser, викликати дерево проекту і відкрити теку Animation Tables зі змінними анімаційної таблиці. У початковому стані проти кожної змінної записані нулі. Вибрати автоматичний режим управління дробаркою, помінявши стан FALSE (0) змінної Avt_Ruch в стан (TRUE) (1).

13. Двічі активізувати одиницею змінну Vkluch і за станом решти змінних (1 або 0) відстежувати роботу програми управління дробаркою відповідно до заданого алгоритму. Зупинити виконання програми, за рахунок активізації одиницею змінної Otkl. Вибрати ручний режим управління за рахунок переводу змінній Avt_Ruch в стан FALSE і відстежувати виконання програми, довільно активізуючи змінні Drob, Kon_n, Kon_v. Якщо в роботі програми виявляться помилки, знайти і виправити їх, переглянувши таблицю змінних і LD-код.

14. Вивести на екран EOM візуалізацію об'єкта автоматизації і пульта управління, двічі клацнувши лівою клавішею миші на теці Runtime Screens в дереві проекту. За допомогою клавіш на пульту управління дослідити роботу програми в автоматичному і ручному режимах, згідно приведеного алгоритму.

15. Зупинити відповідними кнопками на пульту візуалізації режим керування роботою дробарки. На панелі інструментів PL7 PRO натиснути кнопку STOP і через меню PLC-> Disconnect роз'єднати EOM і контролер.

Програма керування роботою дробарки



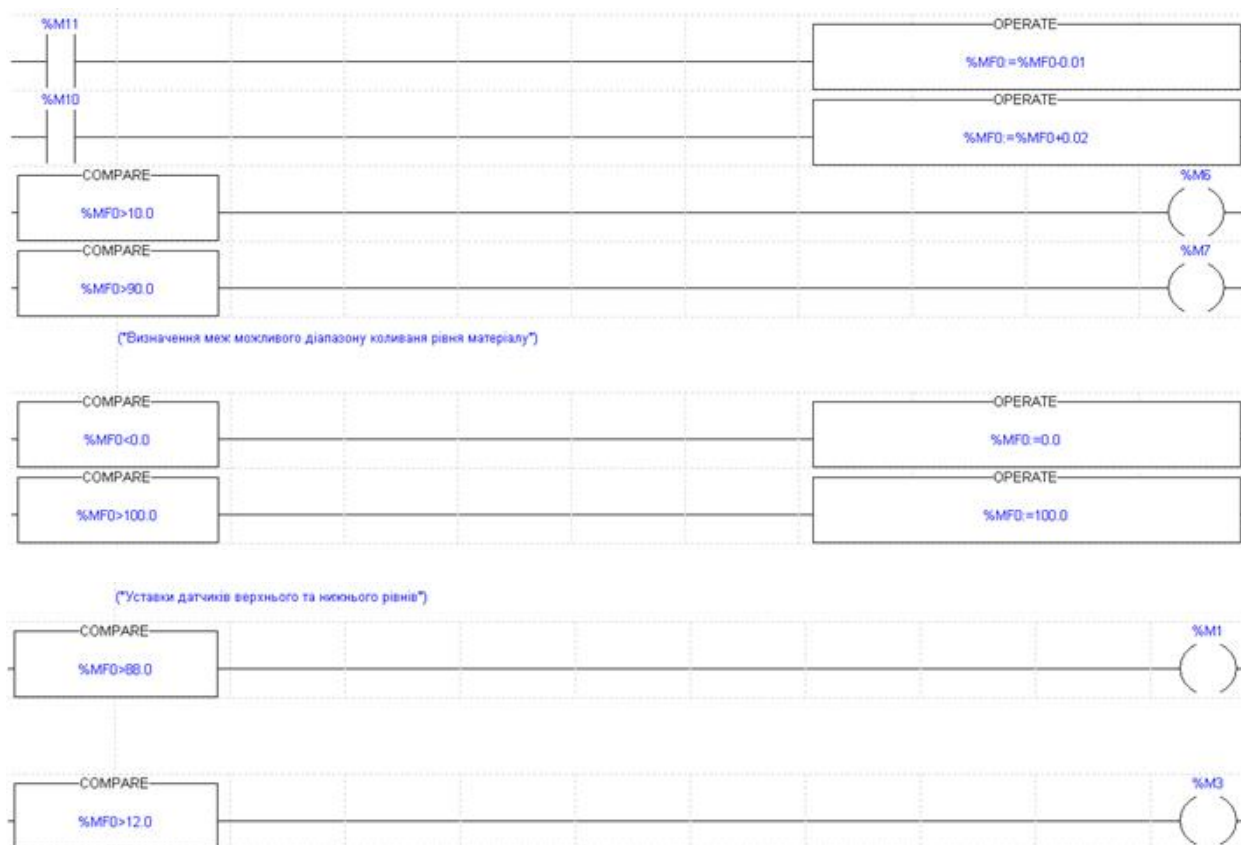


Рисунок 4.3 – Лістинг програми

Завдання для самостійної роботи

Скоригувати, приведену в інструкції програму так, щоб вмикання кожного елемента на мнемосхемі супроводжувалося світінням окремого світлодіода на лабораторному стенді. При оголошенні зовнішніх змінних користуватися таблицею I/O у теці Variables. При цьому враховувати, що світлодіод автоматичного режиму роботи підключений до нульового виходу модуля TSX DSX 08T2, світлодіод сигналізації – до першого, світлодіод дробарки – до другого, світлодіод верхнього конвеєра – до третього, а світлодіод нижнього конвеєра – до четвертого виходу цього ж модуля.

Зміст звіту

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

Контрольні запитання

1. Для чого призначено контролер Modicon TSX Micro і його модифікації?
2. Яка компоновка контролера на лабораторному стенді?
3. Які можливості має середовище програмування PL7 PRO?
4. Як виконується в проекті конфігурування контролера?
5. Як оголошуються змінні?
6. Як розробляється прикладна програма?
7. Як здійснюється налагодження проекту?
8. Як завантажується програма у контролер и здійснюється її запуск?

Лабораторна робота №5

Програмування та дослідження роботи контролера VIPA Speed 7 300

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

Лабораторна установка, рис.5.1, 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 5.1 – 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, доукомплектовує 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 нагоді, 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 МРА 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, ST←/ET←, 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.

Решта інформації про модулі контролера знаходиться у навчальному посібнику, де надаються відомості про Simatic S7-300.

Виконання лабораторної роботи

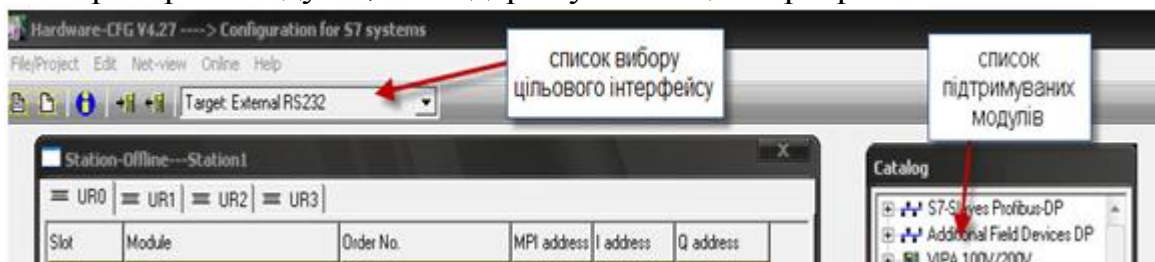
За допомогою програмного пакету WinPLC7 виконати конфігурування та програмування контролера VIPA Speed 7 300 FBD-мовою, який здійснює креслення рівнобедреного трикутника, керуючи рухом каретки графобудовника.

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.

Створення проекту

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 OK.

Для конфігурування контролера, в панелі інструментів натиснути ліву клавішу 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.

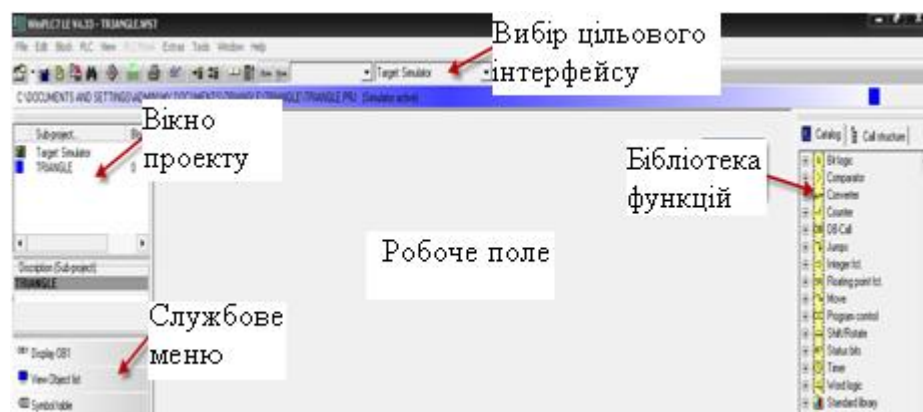
Щоб додати модуль живлення у слот 1, треба позначити його лівою клавішею миші та зі списку PS 300 вікна Catalog вибрати модуль живлення 6ES7307-1BA00-0AA0 і двічі клацнути лівою клавішею миші.

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, вказати мережну адресу контролера 192 140 80 01 і з нижнього списку вибрати активний мережний адаптер Realtek RTL8029(AS) Ethernet Adapter. Натиснути кнопку Determining accessible nodes і 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.

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 «OB» 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 OB1. Таким самим чином створити блок функції FC1.

Для заповнення таблиці символьних 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 стовпчику «адрес таблиці змінних» з'являться їх адреси 0.0 і 2.0.

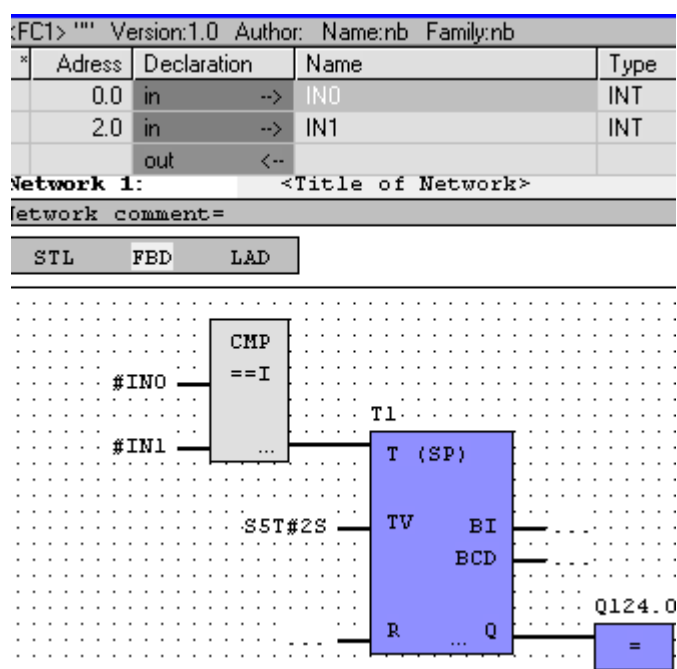
Вибрати FBD-мову. З каталогу Comparator додати у поле Network 1 оператор CMP ==I, двічі клацнувши по ньому лівою клавішею миші. Через нього подаватиметься сигнал на вмикання двигуна.

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.

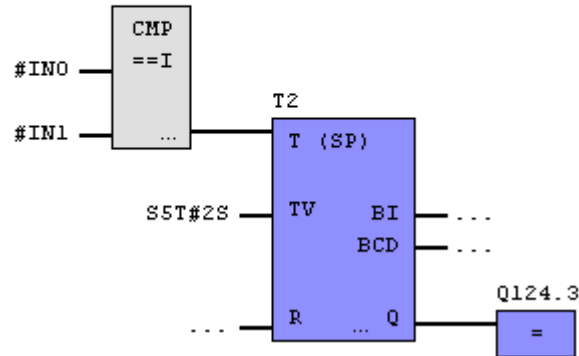
Увага: сигнали на входах таймерів T3, T4 і T5 інвертуються за допомогою кнопки інверсії , що у панелі інструментів.



Network 2: Network title=

Network comment=

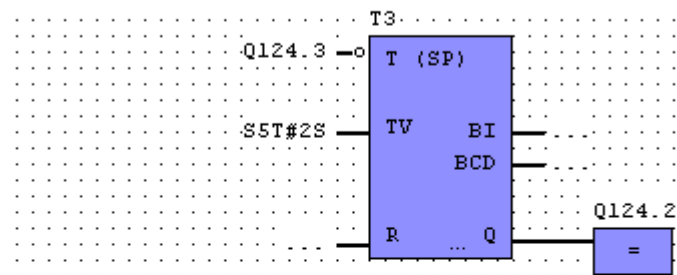
STL	FBD	LAD
-----	-----	-----



Network 3: Network title=

Network comment=

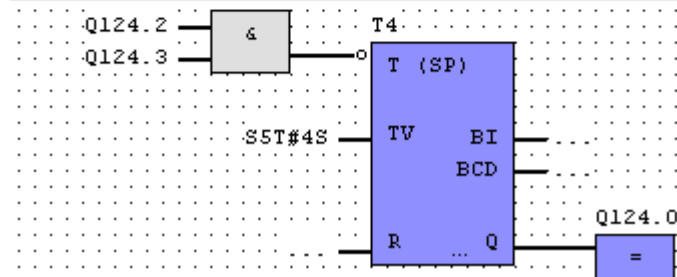
STL	FBD	LAD
-----	-----	-----



Network 4: Network title=

Network comment=

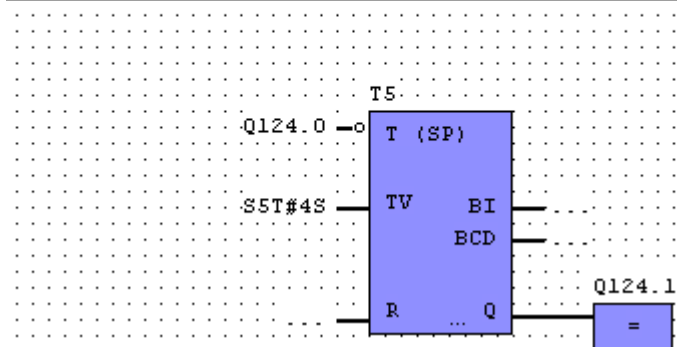
STL	FBD	LAD
-----	-----	-----



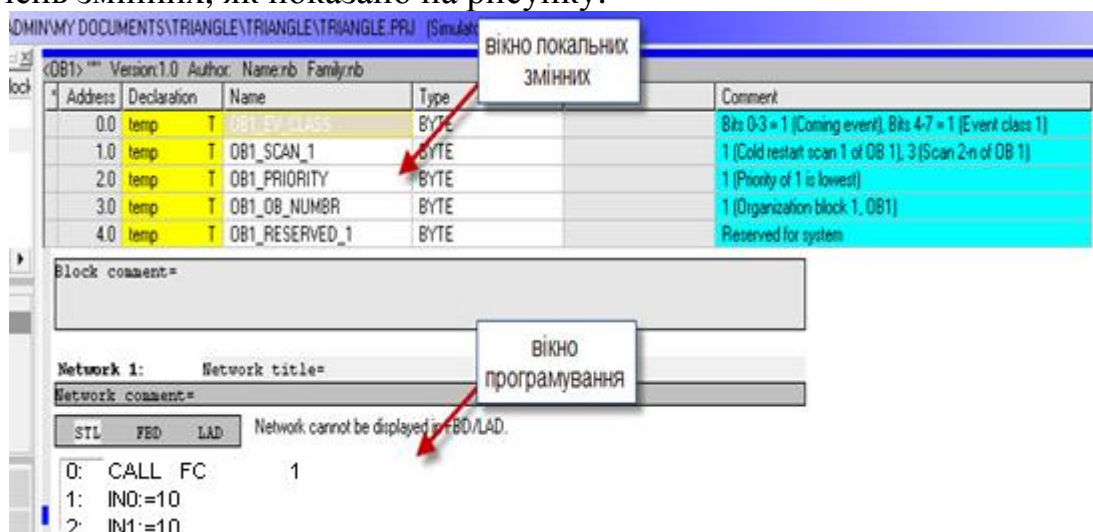
Network 5: Network title=

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:

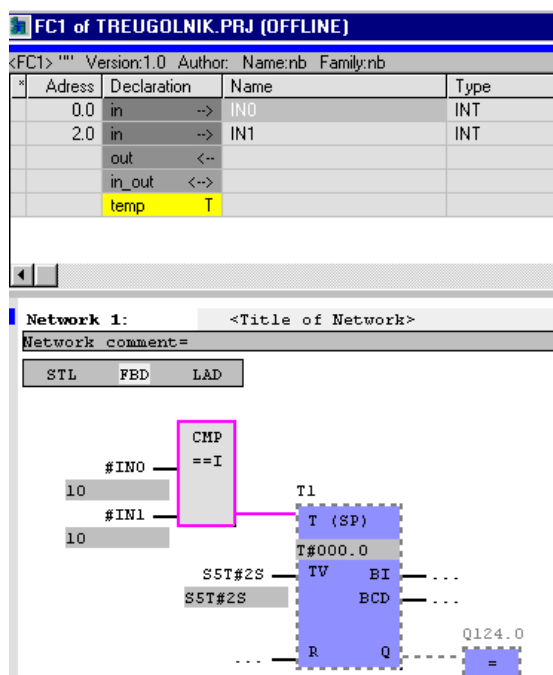



У вікні локальних змінних змінити назву усіх змінних на «TEMP0»... «TEMP9». Закрити блоки OB1 і FC1 та зберегти проект.

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 натиснути Enter.

Якщо помилки є, то їх потрібно виправити.

Щоб запустити 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:





Для завантаження проекту вимкнути режим моніторингу, натиснувши повторно кнопку . В меню «PLC» вибрати пункт «Operating»

mode» і 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  («Sendallblocks»). 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.

Зміст звіту

Звіт має містити назву роботи, короткі відомості про контролер, програму керування графобудовником, результати самостійної роботи, висновки.

Контрольні питання

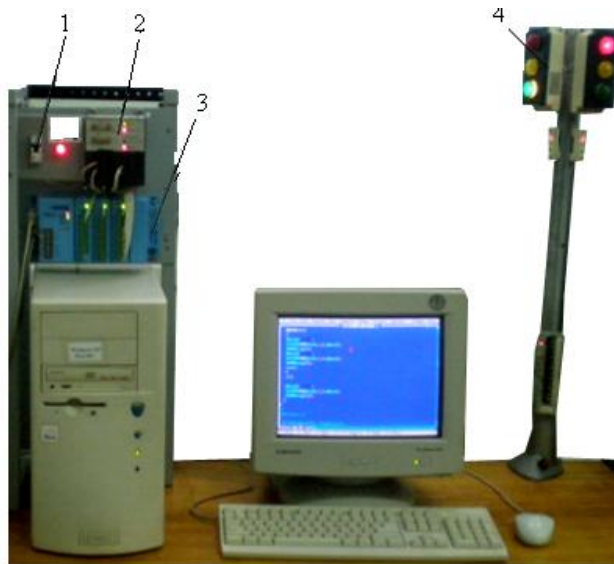
1. Аналогом якого ПЛК є контролер VIPA 300?
2. Якою перевагою характеризується контролер VIPA SPEED 7-300?
3. Як можна збільшити оперативну пам'ять ПЛК VIPA SPEED 7-300?
4. Який інтерфейс у центральних процесорів основний?
5. Як здійснюється програмне конфігурування контролера?
6. Що відбувається, коли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

Лабораторна робота № 6

Програмування та дослідження роботи контролера ADAM-5511

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

Лабораторна установка, рис.6.1, призначена для автоматичного управління роботою світлофора за допомогою мікропроцесорного PC-сумісного контролера ADAM 5511. Вихідні канали модуля дискретного вводу-виводу ADAM 5050, встановленого у слоті з адресою 2, підключені до світлофору.



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

Рисунок 6.1 – Лабораторна установка

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.

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 батареї «BATT».

У правій частині ADAM 5511 знаходяться DIP перемикачі, які призначені для установлення 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.

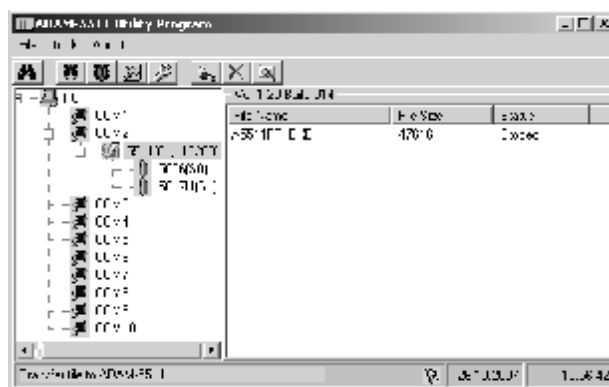
ADAM 5511 підтримує всі дискретні і аналогові ПЗО серії ADAM 5000.

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.



Проте у лабораторній роботі використовувалася не утиліта «ADAM 5511 Windows Utility», а утиліта «ADAM-5510 Series Utility», яка також дозволяє встановити зв'язок з вбудованим інтерпретатором командного рядка ROM-DOS.

Після встановлення утиліти на ПК та її запуску, з'являється вікно, яке складається з трьох панелей:

- на системній панелі відображаються віртуальні диски інструментального комп'ютера;
- на панелі контролера відображається вміст віртуального диска **D:** контролера;
- на консольній панелі відображаються повідомлення, що видаються працюючими програмами контролера.

Написання і компіляція програми для ADAM 5511 здійснюється за допомогою інструментальної системи програмування (ІСП) Borland C++ v.3.1.

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 програм, у пам'яті контролера використовуються відповідні адреси регістрів.

Адресація регістрів для аналогових і дискретних модулів встановлених в ADAM 5511, залежить від номера слота і показана на рис. 6.2.



Рисунок 6.2 – Адресація регістрів модулів ПЗО серії «ADAM5000»

Виконання лабораторної роботи

Розробити програму керування роботою автомобільно–пішохідного світлофору, який має перебувати в чотирьох станах: зеленому, червоному, жовтому і жовто-червоному. Тривалість кожного стану 2с. На першому етапі

загоряється зелене світло правого світлофора та 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 й червоне світло відповідно до автомобільного. Циклограма роботи світлофора та номери відповідних керувальних дискретних виходів «ADAM-5050» приведена на рис.6.3.

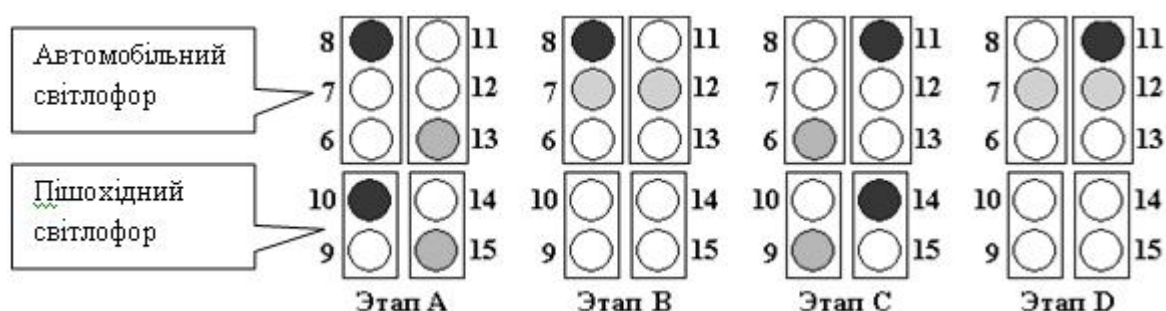


Рисунок 6.3 – Циклограма роботи світлофора та номери відповідних керувальних дискретних виходів «ADAM-5050»

На етапі «А», коли програма запускається, повинні спрацювати лампочки світлофора, що підключені до виходів 8, 10, 13, 15 «ADAM-5050».

На етапі «В» лампочки, що підключені до виходів 7, 8, 12.

На етапі «С» лампочки, , що підключені до виходів 6, 9, 11, 14.

На етапе «D» лампочки, , що підключені до виходів 7, 11, 12.

Перемикання між станами світлофора повинно відбуватися із затримкою у 2с.

Розробка програмного забезпечення контролера

Для створення початкового тексту програми мовою C++, скористаємося таблицею станів дискретних виходів модуля ПЗО, табл.6.1, побудованої відповідно до логіки роботи світлофору:

Таблиця 6.1 Стани дискретних виходів

Сигнал світлофора		ЗПП	ЧПП	ЗПА	ЖПА	ЧПА	ЧЛП	ЗЛП	ЧЛА	ЖЛА	ЗЛА	Не використовують						Значення, HEX
Дискретний вхід/вихід		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Етап «А»	bin	1	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0xA500	
	hex	A				5				0			0					
Етап «В»	bin	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0x1180	
	hex	1				1				8			0					
Етап «С»	bin	0	1	0	0	1	0	1	0	0	1	0	0	0	0	0	0x4A40	
	hex	4				A				4			0					
Етап «D»	bin	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0x1880	
	hex	1				8				8			0					

де

ЗПП- зелений правий пішохідний; ЗЛА- зелений лівий автомобільний;

ЖПА- жовтий правий автомобільний; ЗЛП- зелений лівий пішохідний;
 ЧПП- червоний правий пішохідний; ЖЛА- жовтий лівий автомобільний;
 ЧПА- червоний правий автомобільний; ЗПА- зелений правий автомобільний;
 ЧЛП- червоний лівий пішохідний; ЧЛА- червоний лівий автомобільний.

Для розробки програмного коду необхідно виконати наступні дії:

1. У кореневому каталозі диска D: створити папку Lab6.
2. Запустити на виконання ІСП Borland C++ v3.1. Ярлик для доступу до файлу ВС.EXE розміщений на робочому столі ПК. При необхідності, закрити вікно редактора з попередньою програмою, користуючись кнопкою у лівому верхньому куті. Загальний вигляд вікон ІСП Borland C++ v3.1 показаний на рис. 6.4.

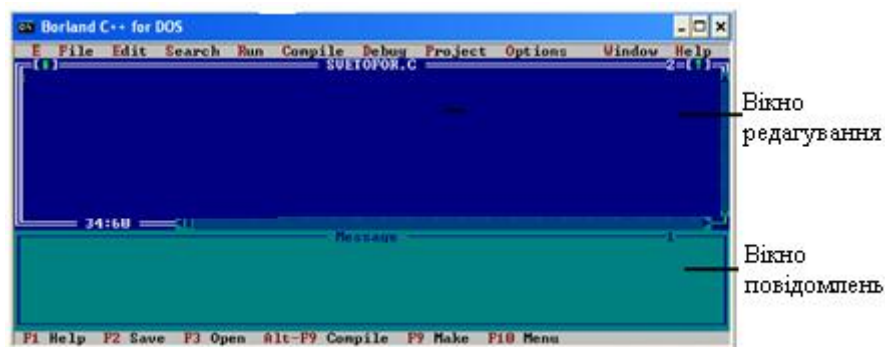


Рисунок 6.4 – Вікна ІСП Borland C++ v3.1

3. В ІСП Borland C++ змінити поточний каталог. Для цього виконати команду меню File → Change dir..., в діалоговому вікні в рядку Name надрукувати D: \ Lab6 і послідовно натиснути Chdir і Ок.
4. Створити новий файл з початковим текстом програми. Для цього виконати команду меню File → New. Відкриється вікно редактора з ім'ям NONAME00.CPP.
5. Зберегти створений файл. Для цього виконати команду меню File → Save as ... і в діалоговому вікні, у полі "Save File As" ввести рядок D: \ LAB6 \ LAB6.C і натиснути Ок.
6. У вікні редагування з ім'ям "LAB6.C" ввести початковий текст програми:

```
//-----
// Підключення бібліотек і файлів
//-----
#include <stdio.h>
#include <conio.h>
#include <dos.h>
#include <time.h>
#include "d:\ADAM\Library\5511drv.h"

// Оголошення та опис функції затримки часу
static void s_delay(void)
{
    time_t t1, t2; // Оголошення змінних
```

```

t1 = time(NULL); // Визначення поточного часу
t2 = time(NULL);
for (;;) // Нескінченний цикл
{
    if((t2-t1) >= 2) // Затримка на 2 сек
        return;
    else
        t2 = time(NULL);
}
}
//-----
// Початок роботи програми. Оголошення змінних. Вхід у нескінченний цикл.
//-----
void main(void)
{
    unsigned int WordData; // Оголошення 2-байтної змінної без знака
    int NSlot=2; // Оголошення знакової 2-байтної змінної

    clrscr(); // Очищення екрана
    printf("\n Press any key to exit \n"); // Видача повідомлення

    // Цикл виконується, поки не буде натиснута якась клавіша
    while(!kbhit())
    {
        //-----
        // Етап А
        // Установлення значень дискретних виходів модуля на етапі А, що
        // забезпечують вмикання ЗПП, ЗПА, ЧЛП, ЧЛА.
        //-----
        WordData=0xA500; // Ініціалізація значенням для Етапу «А»
        Set5050(&WordData,NSlot,0,AWord); // Установлення дискретних виходів
        s_delay(); // пауза 2 секунди
        Get5050(NSlot,0,AWord,&WordData); // Читання стану дискретних виходів
        printf("Etap A. AWord=%x\n",WordData); // Видача налагоджувального
                                                // повідомлення
        //-----
        // Етап В
        // Установлення значень дискретних виходів модуля на етапі В, що
        // забезпечують вмикання ЖПА, ЧЛА, ЖЛА.
        //-----
        WordData=0x1180;
        Set5050(&WordData,NSlot,0,AWord);
        s_delay();
        Get5050(NSlot,0,AWord,&WordData);
        printf("Etap B. AWord=%x\n",WordData);
        //-----
        // Етап С
        // Установлення значень дискретних виходів модуля на етапі С, що
        // забезпечують вмикання ЧПП, ЧПА, ЗЛП, ЗЛА.
        //-----

```

```

WordData=0x4A40;
Set5050(&WordData,NSlot,0,AWord);
s_delay();
Get5050(NSlot,0,AWord,&WordData);
printf("Etap C. AWord=%x\n",WordData);
//-----
// Етап D
// Установлення значень дискретних виходів модуля на етапі D, що
// забезпечують вмикання ЖПА, ЧПА, ЖЛА.
//-----
WordData=0x1880;
Set5050(&WordData,NSlot,0,AWord);
s_delay();
Get5050(NSlot,0,AWord,&WordData);
printf("Etap D. AWord=%x\n",WordData);
} // Кінець циклу
// Зупинка світлофора
WordData=0x0;
Set5050(&WordData,NSlot,0,AWord);
printf("\nSvetoфор ostanovlen.");
printf("\nPrikladnaja programma zavershena.");
}

```

7. Зберегти створений файл. Для цього виконати команду меню File→ Save.
 8. При розробці програми використовуються функції 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 установлений 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).
 - Установлення дискретних виходів ПЗО «ADAM-5050»
 Синтаксис: void Set5050(void *pValue, int Board, int Bit, int Size);
 Параметри функції аналогічні функції «Get5050(...)».
- Тому треба створити проект і додати в нього файли бібліотек «IO.lib», «Utility.lib». Для цього виконується наступна процедура:
- 8.3. Увійти в пункт меню «Project → Open project». У вікні, що з'явиться, в рядку «Open Project File» надрукувати повний шлях та і'мя проекту з розширенням «D:\LAB6\LAB6.PRJ». На клавіатурі ПК натиснути кнопку Enter. У нижній частині екрану з'явиться вікно зі списком файлів проекту, що створюється.
 - 8.4. Активізувати лівою клавішею миші вікно зі списком файлів проекту і увійти в пункт меню «Project → Add Item».
 - 8.5. У діалоговому вікні, «Add to Project List», у рядку «Name» надрукувати маску «D:\ADAM\Library *.LIB» і натиснути Enter.
 - 8.6. З переліку файлів бібліотек, що з'явилися, кнопкою «Add» по черзі додати у список файлів проекту бібліотеки Utility.lib, IO.lib.

- 8.7. У діалоговому вікні «Add to Project List» у рядку «Name» надрукувати маску «D:\LAB6*.C» і натиснути Enter.
- 8.8. Користуючись кнопкою «Add» додати у список файлів проекту LAB6.C.
- 8.9. У діалоговому вікні «Add to Project List» натиснути кнопку «Done».
9. Відкомпілювати програму. Для цього виконати команду меню Compile → Build all. Повідомлення про помилки, виявлені в процесі компіляції, відображаються у вікні "Message". При необхідності усунути помилки, виявлені компілятором.
10. Вийти з ІСП Borland C++ v3.1, користуючись пунктами меню File–Quit.

Завантаження програми в контролер і її запуск

1. Запустити програму «ADAM-5510 Series Utility». Ярлик для доступу до файлу програми розміщений на робочому столі ПК. Вікно програми показано на рис. 6.5.

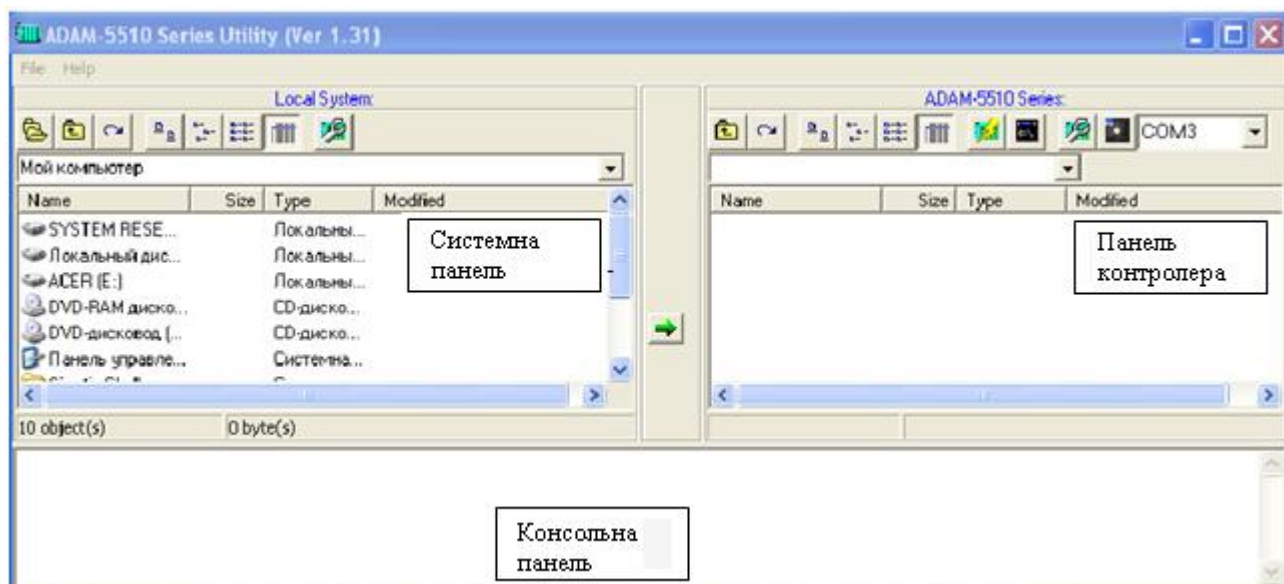




Рисунок 6.5 – Вікно програми "ADAM-5510 Series Utility"


2. На панелі контролера натиснути кнопку "Refresh folder" .

Встановлюється зв'язок з контролером.

Примітки:

1. Зв'язок з контролером встановлюється за умови, що завантажена раніше програма користувача не виконується контролером (світлофор не перемикається).
2. Якщо програма користувача працює до натискання кнопки , то з'являється вікно повідомлення про помилку "Error". Програма користувача зупиниться і світлофор не буде перемикається.
3. Повторно натиснути кнопку . Зв'язок з контролером встановлюється. На панелі контролера з'являється список файлів, розміщених на диску D: контролера. На консольній панелі виводяться повідомлення.

3. На системній панелі перейти в папку D:\Lab6 і сфокусувати файл LAB6.EXE.

4. Натиснути кнопку "Copy file to ADAM" . Відбувається завантаження програми користувача на диск D: контролера.

5. Запустити на виконання програму користувача. Для цього на панелі контролера вибрати LAB6.exe і двічі натиснути ліву клавішу миші. Світлофор почне працювати. Переконайтеся в тому, що логіка його роботи відповідає циклограмі, приведеної на рис. 6.3.

6. Пред'явити працюючу програму викладачу.

7. Після того, як викладач прийме лабораторну роботу, зупинити програму користувача натисканням будь-якої клавіші на клавіатурі (світлофор згасне). Для повторного запуску програми в консольній панелі ввести команду lab6.exe і натиснути клавішу Enter.

8. Видалити файл lab6.exe з диска D: контролера. Для цього на консольній панелі ввести команду ROM-DOS: del lab6.exe і натиснути клавішу Enter.

9. Закрити вікно утиліти "ADAM-5510 Series Utility".

10. Видалити всі файли з папки D: \ Lab6 інструментального комп'ютера.

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, текст програми керування світлофором, результати самостійної роботи і висновки.

Контрольні запитання

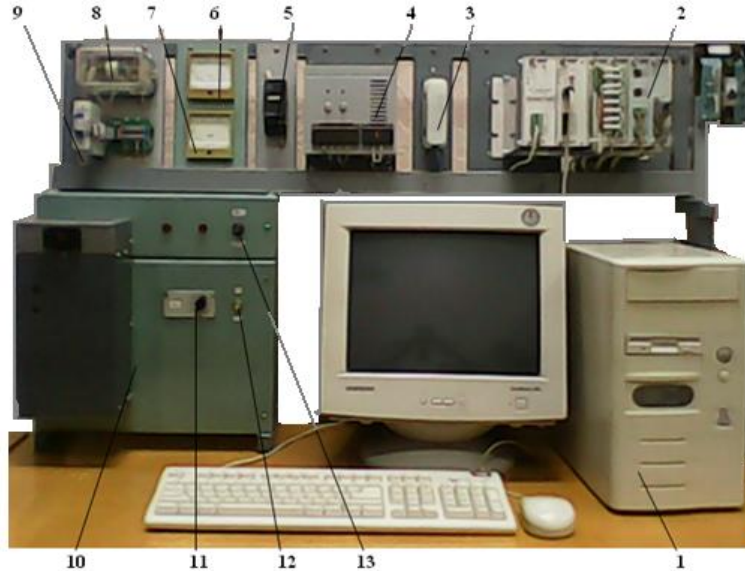
1. Що являє собою контролер ADAM5511?
2. Чим характеризуються модулі ПЗО, що використовуються у роботі?
3. Яке призначення і основні можливості утиліти «ADAM-5511 Windows Utility»?
4. Для чого призначені послідовні порти контролера?
5. Що є інструментом розробки користувальницького програмного забезпечення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

Лабораторна робота №7

Програмування і дослідження роботи контролера Констар К 202

Мета: Ознайомитися з конструкцією, набути навичок програмування і роботи з контролером.

Лабораторна установка, рис.7.1, призначена для вивчення конструкції і дослідження можливостей роботи контролера Констар К 202 виробництва АТ «КОНСТАР» м. ←архів що має наступну компоновку:



1 – комп'ютер; 2 – контролер Констар К202; 3 – перехідний клемник; 4- блок живлення світловим табло; 5- блок живлення виходу аналогового модуля; 6– показчик температури печі; 7 – прилад, що показує величину вихідного сигналу; 8 – емулятор печі; 9– перетворювач; 10 - підсилювач потужності керувального сигналу печі; 11– задатчик температури; 12–тумблер живлення печі; 13– мережний тумблер живлення стенду.

Рисунок 7.1 – Лабораторна установка

- каркас компоновальний СК10.02-01, виготовлений в конструктивах стандарту «Євромеханіка 19'» формату 3U, з кількістю місць для встановлення модулів вводу-виводу рівному 4;

- модуль електроживлення СВ91.06-02, вхідна напруга ~85- 264В, вихідна напруга $5\text{ В} \pm 2\%$;

- модуль мікропроцесорний СР59.15-01, об'єм пам'яті коду робочої програми (РП) 384К, об'єм пам'яті тексту робочої програми (початковий проект) 576К, об'єм пам'яті таблиці даних (ТД) (незалежне ОЗУ) 640К, сервісний канал зв'язку RS-232, комунікаційний канали зв'язку RS-485, ModBus RTU;

- модуль вводу-виводу дискретних сигналів СР36.01 8-канальний, вхідна напруга 24В;

- модуль виводу дискретних сигналів релейний СР35.21-01 8-канальний, вихідна напруга $=30\text{В}$, ~250В;

- модуль вводу безперервних сигналів постійного струму СР31.02-01 4-канальний, діапазон виміру 0-20мА;

- модуль виводу безперервних сигналів постійного струму СМ32.03-03
4-канальний, діапазон зміни вихідного сигналу 0-5В, 0-10В/-5–5В, 0-20мА

Як об'єкти автоматизації використовується табло ілюмінації і нагрівальна піч.

Програмований логічний контролер K202

Програмований логічний контролер K202 – це універсальний технічний засіб для створення на його базі пристроїв управління різним технологічним устаткуванням і АСУ ТП будь-якої складності, рис.7.2.

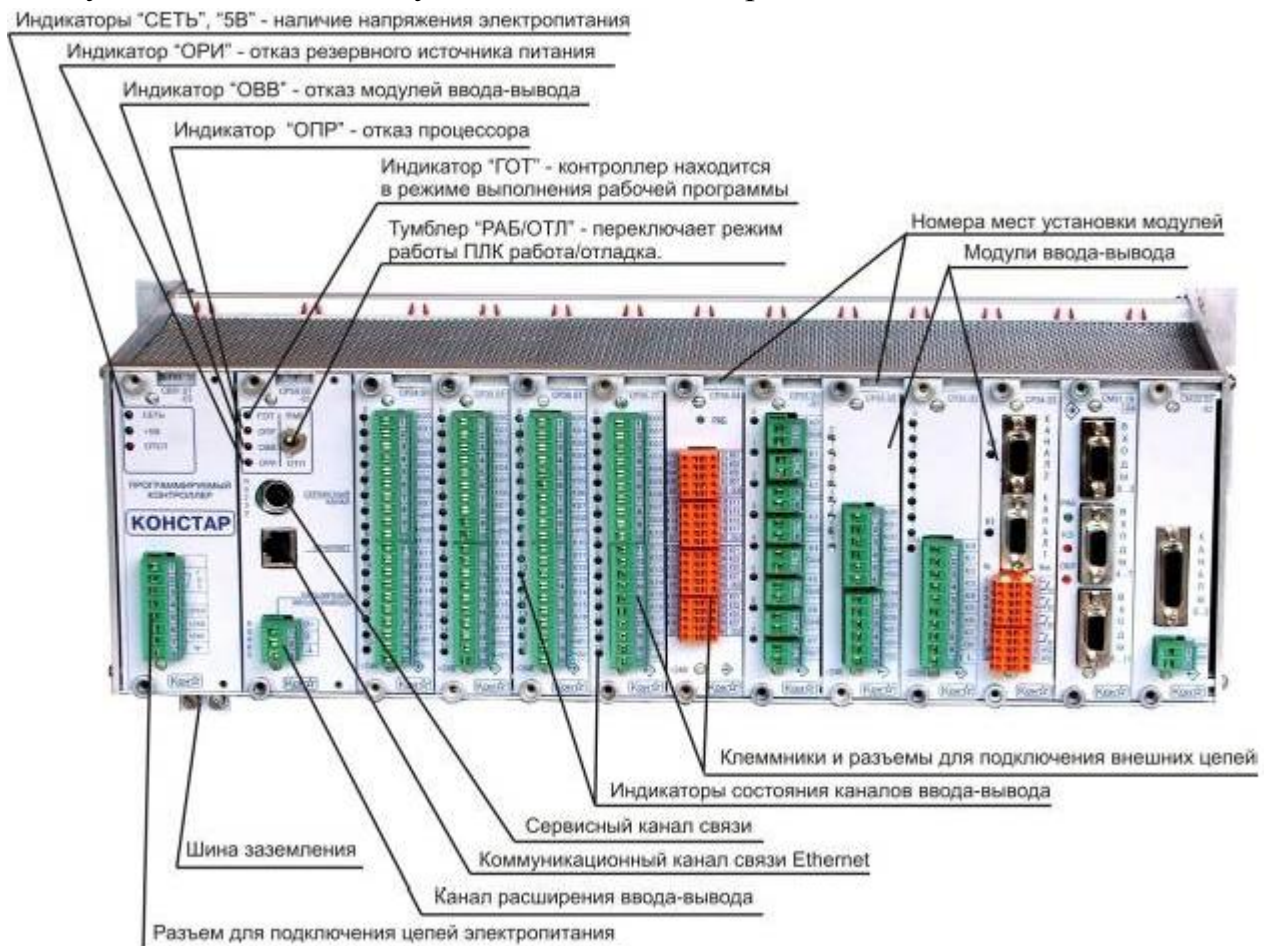


Рисунок 7.2– Программований логічний контролер K202

Каркас, з мікропроцесорним модулем, модулями електроживлення і вводу-виводу є базовим блоком. Модуль мікропроцесорний встановлюється в каркас на спеціальне місце «ПР», і має резервне джерело живлення (РЖ), що забезпечує збереження інформації протягом одного місяця в незалежній пам'яті при відключеному основному електроживленні.

До базового блока можна підключити до чотирьох блоків розширення вводу-виводу, які комплектуються модулем електроживлення, модулем розширення і модулями вводу-виводу. Кількість модулів вводу-виводу в одному блоці може бути 4, 8, 11 або 16.

Базовий блок контролера здійснює обмін інформацією з блоками розширення через порт «Канал розширення» по інтерфейсу RS485. Електричне живлення здійснюється від нестабілізованого джерела постійного струму напругою 18-36В або від мережі змінним однофазним струмом напругою 85-264В 50 гц.

Комунікаційні можливості деяких ПЛК K202 розширені за рахунок каналу Ethernet 10/100 Мбіт/с з протоколом Modbus/TCP, що дозволяє створювати на базі K202 АСУ ТП з розвиненою локальною мережею. На їх базі можлива побудова систем управління з «гарячим» резервуванням.

Програмування ПЛК K202 здійснюється мовами LD і ST стандарту МЕК 61131-3 за допомогою системи програмування контролерів «Констар» K748v2.

Виконання лабораторної роботи

Після запуску системи програмування відкривається головне вікно. Створення прикладного програмного забезпечення відбувається за концепцією проекту який містить прикладну програму, словник проекту (опис змінних, констант і буквено-цифрових повідомлень), конфігурацію контролера, реквізити (найменування проекту, Ф.І.О розробника, версію проекту і ін.), список об'єктів, вікна динаміки змінних та ін. Весь проект зберігається в єдиному файлі з розширенням “.ppc”.

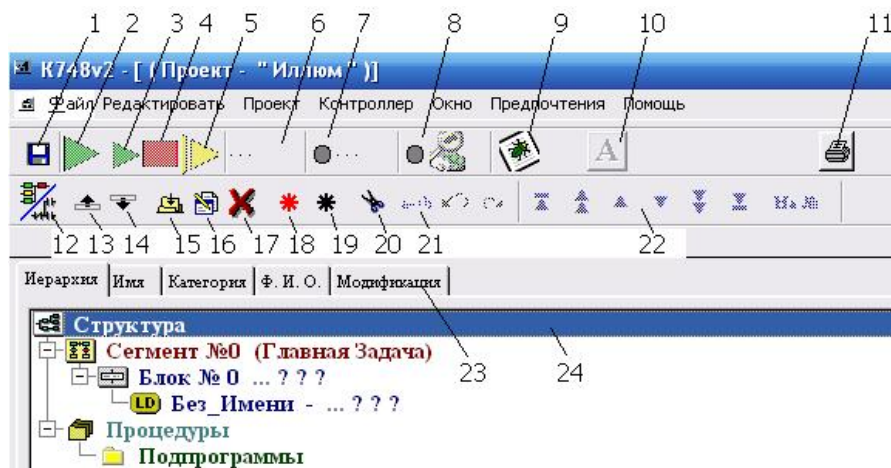
Створення проекту

Для створення нового проекту (відкриття іншого) поточний відкритий проект необхідно закрити, користуючись меню “Файл \ Закрить проект”.

Увійдіть до головного меню “Файл \ Новый проект”. У вікні, що відкриється, задайте реквізити проекту: найменування, Ф.І.О. розробника, зі списку виберіть тип процесорного модуля контролера, що встановлений на лабораторному стенді, – СР 59.12/15.

Після натискання на кнопку “Да” створюється заготовка нового проекту, яка містить в конфігурації контролера тільки інформацію про процесорний модуль (каркаси порожні), і прикладну програму, у складі якої є сегмент з блоком №0 і секцією “Без имени“, рис.7.3. Встановіть курсор на секцію “Без имени“ і, натискаючи праву клавішу миші, з контекстного меню виберіть «Свойства». У відповідних полях вкажіть ім'я «Илюминация», LD-мову, категорію - «Прочее» і натисніть кнопку ОК.

Конфігурування контролера виконується в меню “Проект\ Конфигурация”. Для цього необхідно відкрити вікно «Конфигурация контроллера» і на закладці “Основные” заповнити всі поля необхідною інформацією: тип – СР59. 12/15, час циклу – 300, початок зони пам'яті, що зберігається, – 5000, довжина зони, що зберігається, – 2000.



1 Кнопка - Зберегти проект; 2 Кнопка - “Холодний” ПУСК; 3 Кнопка - “Гарячий” ПУСК; 4 Кнопка – СТОП; 5 Кнопка – ЦИКЛ; 6 Табло індикації ключа (Робота\ Відладка); 7 Табло індикації Підрежиму; 8 Індикатор і кнопка Відмов контролера; 9 Кнопка повторного виклику вікна повідомлень про помилки; 10 Кнопка перекл. (Анімація\ Редагування); 11 Кнопка – Друкувати <Ctrl+P>; 12 Кнопка перемикавання рівня представлення проекту (Дерево\ Схема); 13 Кнопка – Секція/Ланцюг вгору <Shift+Up>; 14 Кнопка – Секція/Ланцюг вниз <Shift+Down>; 15 Кнопка – Додати <Insert>; 16 Кнопка – Редагувати <Enter>; 17 Кнопка – Вкл./Откл. Ланцюг <Ctrl+O>; 18 Кнопка – Маркіровки\ Демаркіровки секції; 19 Кнопка – Зняти маркіровку; 20 Кнопка – Видалити <Delete>; 21 Кнопка автоматичної заміни ідентифікаторів в ланцюгах; 22 Група навігаційних кнопок; 23 Набор закладок редактора структури; 24 Бар-показчик

Рисунок 7.3 – Головне вікно системи при відкритому проекті

На закладці “Периферія” натисніть кнопку “Показати компоновку каркасов” – відкриється вікно “Компоновка контролера”. Лівою клавішею миші активізуйте «Основной каркас» і натисніть кнопку “Добавить”. У вікні, що відкрилося, «Добавление модуля» за допомогою кнопки виберіть із запропонованого списку тип модуля, відповідно до місця установки на контролері і натисніть кнопку ОК. (Місце 00–модуль вводу-виводу дискретних сигналів СР 36.01, місце 01– модуль виводу дискретних сигналів релейний СР 35.21-01, місце 02–модуль вводу безперервних сигналів постійного струму СР 31.02-01, місце 03–модуль виводу безперервних сигналів постійного струму СМ 32.03-03).

Завдання до лабораторної роботи

1) Розробити програму керування ілюмінацією, відповідно до якої на світловому табло, яке складається з 8 променів і двох кіл, відбувається почергове вмикання із затримкою 1с лампочок зовнішнього кола за стрілкою годинника, а лампочок внутрішнього кола проти стрілки годинника. При цьому почергово спочатку вмикаються, а потім вимикаються промені світлового табло.

2) Розробити програму керування температурою печі за ПІД-законом.

Фундаментом прикладної програми контролера є словник проекту. У ньому призначаються символічні імена і визначаються властивості основних

об'єктів, з якими оперує прикладна програма. До них відносяться: входи модулів вводу, виходи модулів виводу, внутрішні змінні (елементи пам'яті), лічильники, таймери, масиви констант і літеро-цифрові повідомлення (БЦС).

У системі К748 реалізована концепція ізольованого простору імен, згідно якої всі об'єкти, що мають символічні імена, діляться на дві групи – глобальні і локальні. Глобальні об'єкти доступні користувачеві з будь-якої секції прикладної програми, а локальні об'єкти доступні тільки в межах однієї секції, до якої вони належать. Для обслуговування цих двох типів об'єктів в системі є глобальний словник, а також кожна секція прикладної програми має локальний словник. При формуванні словників необхідно прагнути до мінімізації числа глобальних ідентифікаторів, проте повністю без глобальних ідентифікаторів в реальному закінченому проекті обійтися не можна за наступними причинами:

- входи і виходи контролера не можуть бути локальними, це виключно глобальні об'єкти;
- літеро-цифрові повідомлення також є глобальним об'єктом;
- часто при побудові складних АСУТП потрібний зв'язок ПЛК з персональним комп'ютером. Для цього в словнику оголошують внутрішні змінні (реєстри) з жорстко виділеною адресою (номером), а об'єкти з фіксованою адресою можуть бути тільки глобальними.

Ім'я глобального об'єкта обов'язково починається з символу “_” (підкреслення). Ім'я локального об'єкта обов'язково починається з букви (латинської або російської). Ця вимога відстежується системою автоматично.

Локальний словник секції може бути викликаний з вікон редагування контактів, котушок, блоків порівняння і вікон додавання ідентифікаторів у вікно “динаміки змінних” натисканням на кнопку “...” (чорного кольору), а глобальний словник – з головного меню і з вікон “редагування.”, натисканням на кнопку “:::” (червоного кольору).

Для формування словника глобальних змінних виберіть в меню Проект – Словарь (Глобальн.), а у вікні, що з'явиться, вкладку «Все» і натисніть «Добавить». Задайте ім'я «_Вход», яке відповідає змінній контролера, що характеризує температуру печі, виберіть тип – «Внутр», каркас №0, місце 02, регістр 04, що працює з другим каналом модуля СМ3102-01, і натисніть ОК.

Решту глобальних змінних занесить у словник, користуючись табл.7. 1.

Після оголошення глобальних змінних необхідно встановити курсор на секцію «Илюминация» і двічі клацнути лівою клавішею миші. З'явиться робоче поле проекту. Натисніть праву клавішу миші на робочому полі і з контекстного меню виберіть «Лок. Словарь». Використовуючи кнопку «Добавить», треба оголосити таймери з іменами «Т1-Т26» і «Пуск» з параметрами: тип – Таймер, визначення для мови «ST» - TON, дискретність – Секунди.

Таблиця 7.1

Глобальні змінні проекту

Ім'я	Коментар	Тип	Адреса В/В			Характер/ Розрядність
			Каркас	місце	Канал (Регістр)	
_Вход	Температура	Вхід.	0	02	03	Аналог.
_Виход	Сигнал керування	Вихід.	0	03	00	Аналог.
_Задатчик	Сигнал задатчика	Вхід.	0	02	02	Аналог.
_Вход1		Внутр.	0			Дійсне
_Виход1		Внутр.	0			Дійсне
_Задатчик1		Внутр.	0			Дійсне
_a_Control		Внутр.	0			Аналог.
_Massive		Внутр.	0			Аналог.
_Огонь1	Вих. сигнал на світлодіод 1	вихід	0	00	0	Дискрет.
_Огонь11	Вих. сигнал на світлодіод 11	вихід	0	00	1	Дискрет.
_Огонь12	Вих. сигнал на світлодіод 12	вихід	0	00	2	Дискрет.
_Огонь13	Вих. сигнал на світлодіод 13	вихід	0	00	3	Дискрет.
_Огонь14	Вих. сигнал на світлодіод 14	вихід	0	00	4	Дискрет.
_Огонь15	Вих. сигнал на світлодіод 15	вихід	0	00	5	Дискрет.
_Огонь16	Вих. сигнал на світлодіод 16	вихід	0	00	6	Дискрет.
_Огонь17	Вих. сигнал на світлодіод 17	вихід	0	00	7	Дискрет.
_Огонь2	Вих. сигнал на світлодіод 2	вихід	0	01	0	Дискрет.
_Огонь3	Вих. сигнал на світлодіод 3	вихід	0	01	1	Дискрет.
_Огонь4	Вих. сигнал на світлодіод 4	вихід	0	01	2	Дискрет.
_Огонь5	Вих. сигнал на світлодіод 5	вихід	0	01	3	Дискрет.
_Огонь6	Вих. сигнал на світлодіод 6	вихід	0	01	4	Дискрет.
_Огонь7	Вих. сигнал на світлодіод 7	вихід	0	01	5	Дискрет.
_Огонь8	Вих. сигнал на світлодіод 8	вихід	0	01	6	Дискрет.
_Огонь9	Вих. сигнал на світлодіод 9	вихід	0	01	7	Дискрет.

Після формування словника розробляється прикладна програма, побудова якої здійснюється за допомогою трьох редакторів - редактора структури, редактора секції і редактора сегмента.

Редактор структури є надбудовою над редакторами секції і редактором сегмента, тобто є редактором більш високого рівня. Об'єктами, з якими він оперує, є сегменти, блоки, підпрограми і секції.

Перемикання між редакторами структури і секції виконується за допомогою кнопки 12, рис.7.3. Основні функції редактора структури наступні:

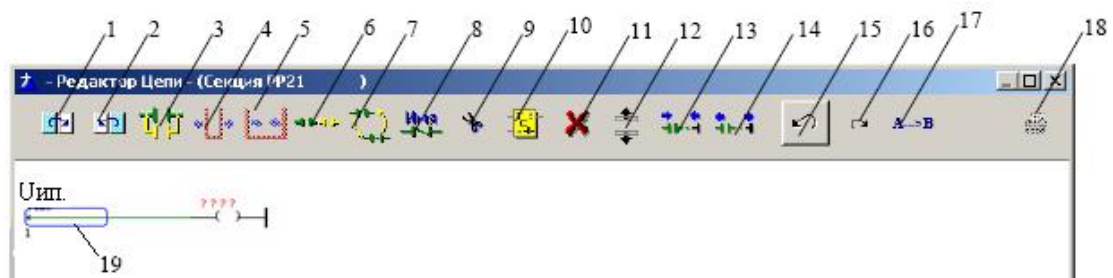
- деревовидне представлення структури проекту;
- переміщення по сегментах, блоках і секціях ;
- редагування структури;
- анімація структури (використовується при налагодженні)

Редактор структури може працювати в двох режимах, анімації і редагування. Перемикання режимів здійснюється кнопкою № 10, рис.7.3, яка активна при встановленому зв'язку з контролером. Будь-яка зміна проекту можлива тільки в режимі редагування. Режим анімації використовується для спостереження за поведінкою сегментів і блоків в динаміці (при виконанні контролером прикладної програми).

Завдання 1

У редакторі структури проекту встановити курсор на секцію «Илюминация» і двічі клацнути лівою клавiшею миші. З'явиться робоче поле проекту:



Щоб створити LD-мережу, необхідно правою клавiшею миші натиснути на робочому полі і вибрати вкладку Додати –Цепь. Дворазовим натисканням лівої клавiші на сегменті відкриється вікно редактора, рис.7.4, що містить панель інструментів для додавання контактів і вибору їх типа, лічильників, таймерів і т.ін.



1 Кнопка – Наступний сегмент <PgDwn> (Вперед) 2 Кнопка – Попередній сегмент <PgUp> (Назад)
3 Кнопка – Додати Паралель. гілку <Shift+Ins>; 4 Кнопка – Розширити Гілку <"Сірий"+>; 5 Кнопка – Звузити Гілку <"Сірий"->; 6 Кнопка – Додати Контакт <Ins>; 7 Кнопка – Змінити Тип Контакту < Shift+Enter>; 8 Кнопка – Призначити Ім'я (Редагуван.) <Enter>; 9 Кнопка – Видалити контакт/Котушку <Delete>; 10 Кнопка – Змінити тип.сегмента <Ctrl+Enter>) 11 Кнопка – Вкл./Вимк. сегмент <Ctrl+O> 12 Кнопка переміщення котушок <Shift+Up(Dwn)> 13 Кнопка – Укоротити сегмент; 14 Кнопка – Подовжити сегмент; 15 Кнопка – Відмінити Операцію <Ctrl+BakSpase>; 16 Кнопка – Повторити Операцію <Ctrl+A>; 17 Автоматич. заміна ідентифікаторів в сегменті; 18 Кнопка – Друкувати сегмент на Принтері <Ctrl+P>; 19 Показчик позиції (Курсор)


Рисунок 7.4 – Вікно редактора сегмента




Редактор сегмента дозволяє переміщатися як усередині сегмента так і міжсегментами. Для переходу від одного сегмента до іншої використовуються кнопки 1 і 2 в панелі інструментів редактора. Навігація усередині сегмента здійснюється за допомогою покажчика позиції, який відображається яскраво синім прямокутником 19, рис.7.4, і вказує на елемент сегмента, з яким буде проводиться та або інша операція редагування.



Для створення програми перевести покажчик на початок сегмента і в панелі інструментів двічі натиснути кнопку  «Добавить контакт». У сегменті з'являться 2 контакти. Двічі клацнути лівою клавiшею миші по кожному з них. У меню, що з'явиться, натиснути кнопку  і з глобальних змінних для цих контактів вибрати імена _Огонь2, ←8, і їх тип – нормально замкнений.

Двічі клацнути по котушці, вказати тип S, з глобального словника вибрати змінну _Огонь2 і закрити редактор сегмента.


На робочому полі встановити покажчик на початок сегмента і додати наступний сегмент, використовуючи кнопку 15, рис.7.3.

В панелі інструментів двічі натиснути кнопку  «Добавить контакт». У сегменті з'являться 2 контакти. З глобальних змінних для цих контактів вибрати імена _Огонь2, ←8, і їх типи, відповідно, нормально розімкнений і нормально замкнений, аналогічно діям у першому сегменту.

Поставити покажчик на початок сегмента і кнопкою  у панелі інструментів, додати паралельну гілку. Щоб охопити нею перший контакт необхідно клацнути по ньому лівою клавiшею миші. Щоб охопити два контакти паралельною гілкою, необхідно в панелі інструментів клацнути по кнопці  «Расширить ветвь» і ще раз натиснути кнопку . Двічі клацнути по контакту паралельного сегмента, що з'явився. З локального словника вибрати ім'я контакту таймера T17, тип–нормально розімкнений і активізувати «Использовать выход», натиснути ОК.

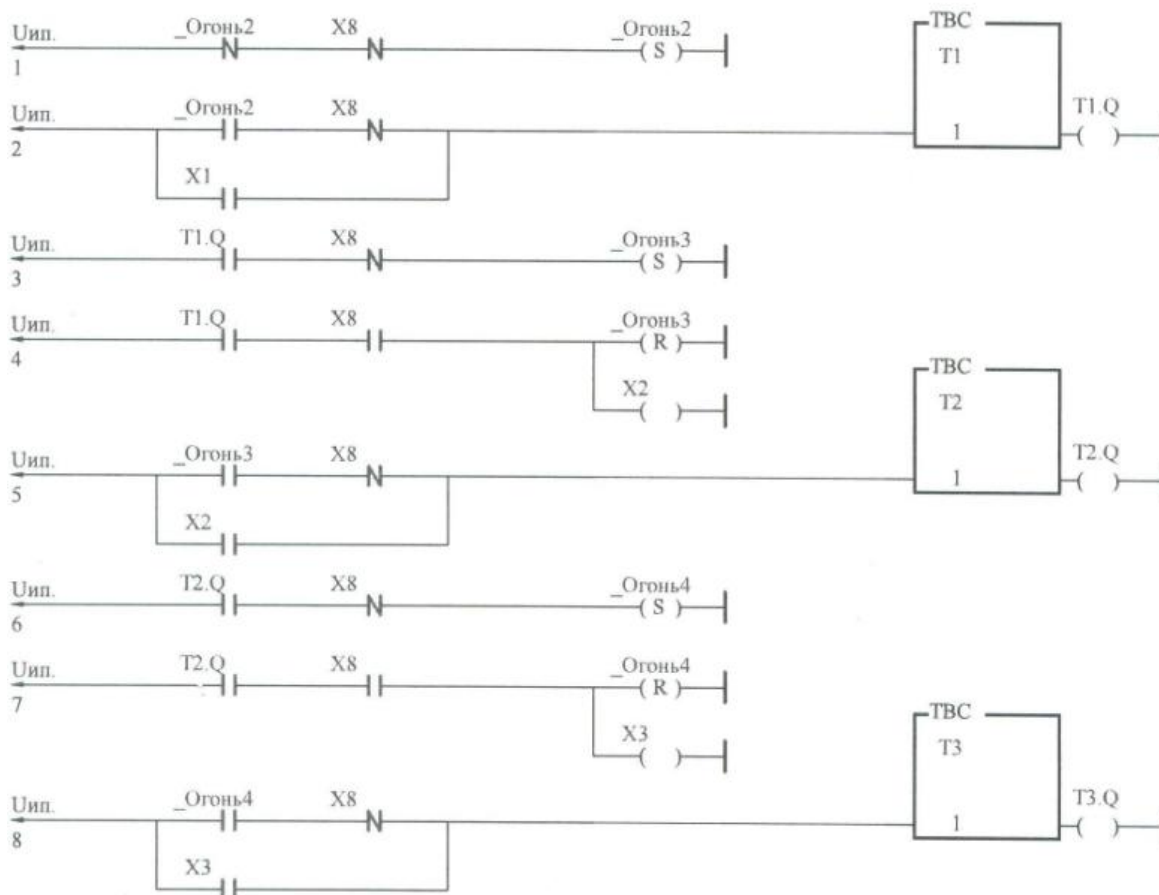
Для третього контакту необхідно увійти до словника локальних змінних, натиснути кнопку , вибрати ім'я T16 і тип контакту – нормально замкнений. Активізувати альтернативу «Использовать выход», натиснути ОК. На робочому полі встановити покажчик на початок сегмента і додати наступний сегмент, використовуючи кнопку 15, рис.7.3. Перевести покажчик на котушку, двічі клацнувши лівою клавiшею миші, увійти в редактор сегмента, і в панелі інструментів, використовуючи кнопку , вибрати таймер. Лівою клавiшею миші двічі клацнути на таймері, із словника локальних змінних задати ім'я Пуск, вибрати тип «На включение», дискретність – секунди і уставку константи -1.

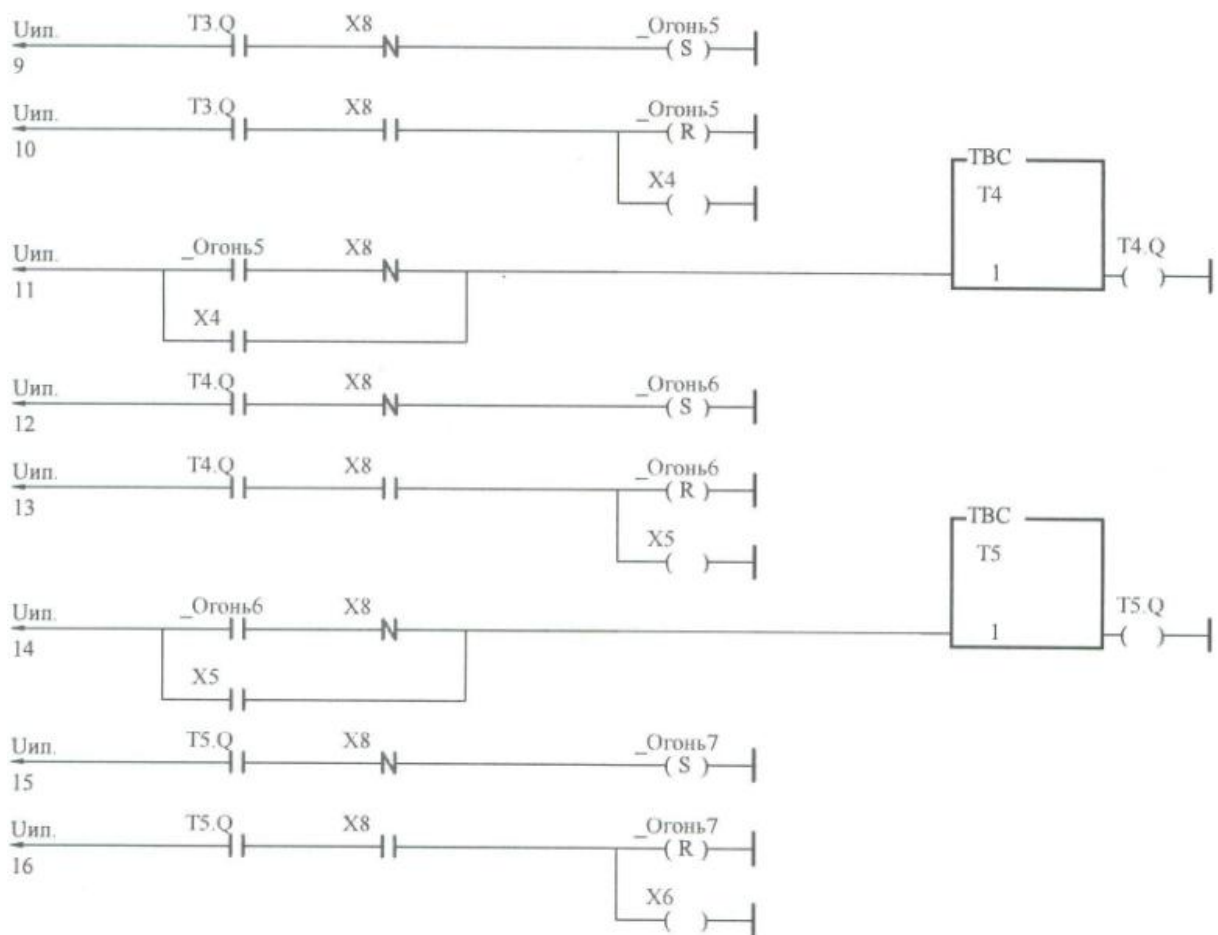
Закрити редактор кнопкою ОК. Видалити котушку з питаннями, використовуючи клавішу “Del”.

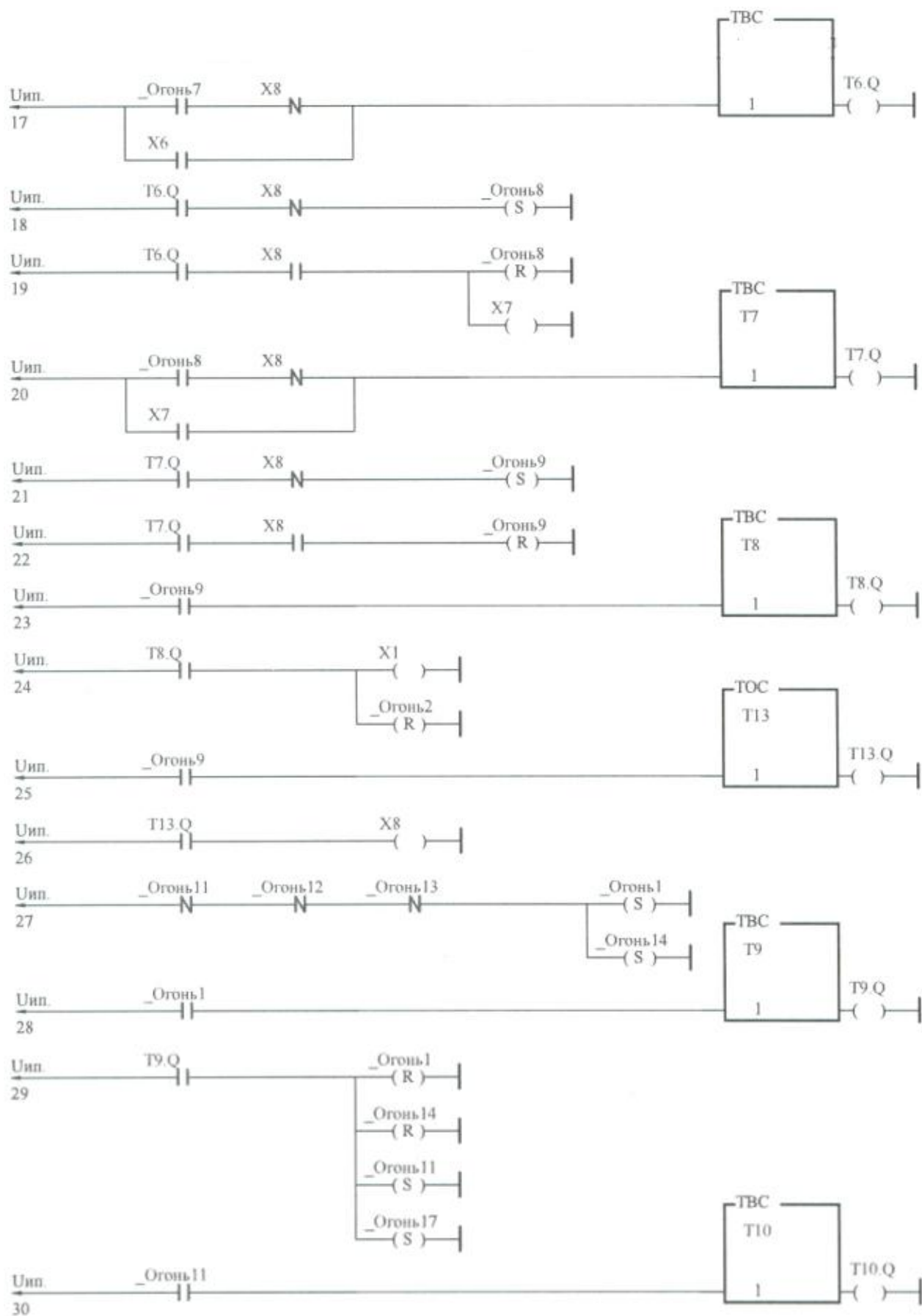
Якщо таймер накладається на попередній сегмент, необхідно повернутися в редактор сегмента і за допомогою кнопки  у панелі інструментів розтягнути сегмент з таймером.

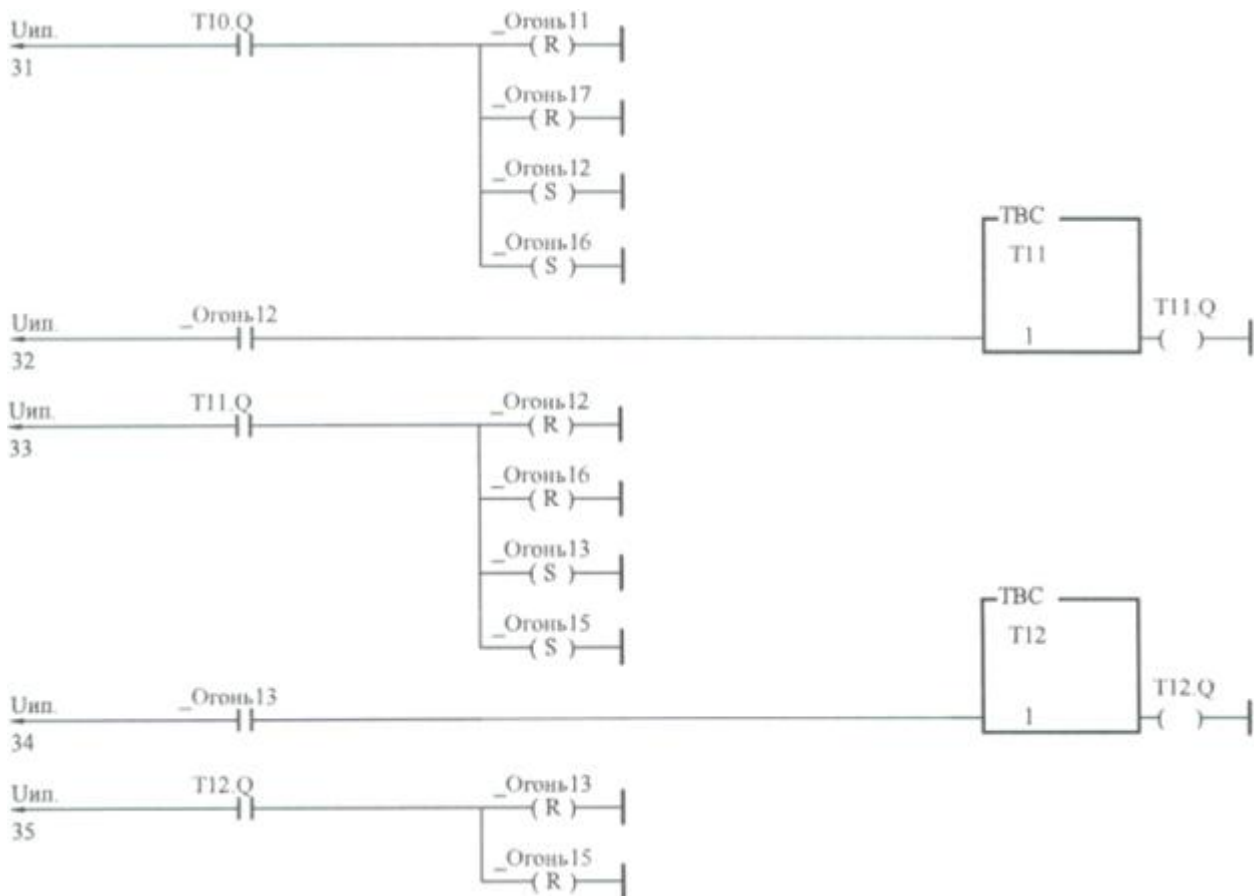
Встановити покажчик на початок сегмента і за аналогією з першим сегментом додати нормально розімкнений контакт з ім'ям _Огонь2.

Продовжити розробку програми, відповідно до приведеного роздруку:









Налагодження проектів в системі K748 відбувається в так званому комплексному режимі, тобто при встановленому зв'язку з контролером і тумблері на центральному процесорі переведеному в положення налагодження «Отл». Для того, щоб контролер міг взаємодіяти з ЕОМ використовується FLASH-ключ, розташований усередині системного блоку ЕОМ.

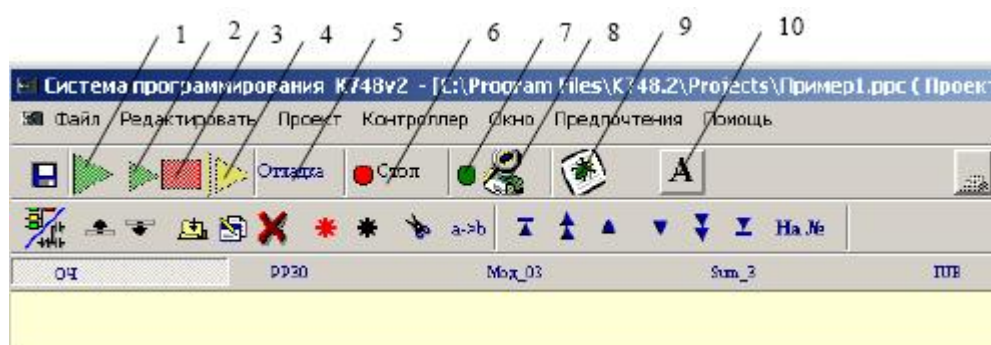
Подати тумблером 13 напругу на лабораторний стенд і увімкнути блок живлення світлового табло, рис.7.1. У головному меню вибрати підміну «Контроллер», активізувати позицію «Установить связь ». Після з'єднання в панелі інструментів зміниться вид деяких позицій, рис.7.5. Перемикач тумблера–5 на процесорному модулі контролера, показує положення тумблера «Робота/Отладка».

При положенні тумблера «Отладка» контролер знаходиться в однойменному режимі, який характеризується наявністю трьох підрежимів: «Пуск», «Стоп», «Цикл».

Для входу в режим анімації треба скористатися кнопкою-10, яка доступна після установа зв'язку з контролером.

Основне завдання режиму анімації в редакції сегмента, це динамічне відображення стану контактів, котушок, лічильників, таймерів. Для прямого контакту зелений колір означає що біт, асоційований з даним контактом, знаходиться в одиничному стані, а червоний в нульовому. Для інверсного контакту навпаки, зелений колір означає нуль, червоний одиницю. Зелений колір

котушки відповідає одиничному значенню біта, червоний нульовому. Якщо сегмент, що анімується, містить лічильник або таймер, то окрім відображення стану котушки виходу також динамічно відображається накопичене значення і значення уставки.




1,2,3,4 – Кнопки Пульта УПРАВЛЕНИЯ; 1- Кнопка - “Холодный” ПУСК; 2- Кнопки - “Горячий” ПУСК; 3- Кнопка – СТОП; 4- кнопка – ЦИКЛ; 5- Табло индикации ключа (Работа\Відладка); 6- Табло индикации Пидрежиму; 7- Индикатор Відмов Контролера; 8- Кнопка – Диагностика; 9- Кнопка повторного вызова окна уведомлений про помилки; 10-Кнопка переключения режимів Анімація/редагування)

Рисунок 7.5 – Панель инструментов при установленном зв'язку с контроллером


Анімація секції дозволяє відображати стан тільки дискретних об'єктів, а також лічильників і таймерів. Для спостереження за поведінкою будь-яких об'єктів, а також для примусової зміни їх значень служить вікно, що викликається з меню «Главное меню/ Проект/ Динамика переменных».


Після цього увійти в підміну «Контроллер – Загрузить проект». При наявності помилок на екрані з'явиться вікно з повідомленнями, якщо помилок немає – розпочнеться процес завантаження. Після його завершення – здійснити пуск контролера, користуючись відповідним запитанням на робочому полі.


Щоб управляти роботою контролера за допомогою кнопок на панелі інструментів , необхідно заздалегідь їх активувати, скориставшись меню «Контроллер – Пульт управления».


Після перемикавання тумблера «Работа/Отладка» в положення «Работа» на центральному процесорі, контролер переходить в режим самостійної роботи з об'єктом управління. Зупинити контролер можна за допомогою позиції «Контроллер – СТОП».

Завдання2

Після прогону програми управління ілюмінацією, в панелі інструментів натиснути кнопку  «Переключить режим Дерево/Схема» і встановити курсор на секцію «Иллюминация». Клацнути правою клавішею миші і з контекстного меню «Добавить» вибрати нову секцію. У вікні, що з'явиться, присвоїти ім'я секції, наприклад «Печь_ПИД», вибрати мову – LD і категорію – «Прочее».

Аналогічно попередньому завданню відкрити робоче поле секції «Печь_ПВД». Створити перший сегмент, встановити на ньому показчик і двічі клацнути лівою клавiшею миші. Відкриється редактор сегмента. Перевести показчик на котушку і лівою клавiшею миші відкрити вікно «Операции». Щоб настроїти реєстр другого каналу модуля CP31.02-01 на виведення інформації у внутрішню шину контролера, у вкладці «Встроенные функции» вибрати ЗПРМ «Записать в реєстр модуля» і встановити параметри: № місця модуля в каркасі – 02; № реєстра другого каналу – 04; число 0. Натиснути ОК. Перевести показчик на функцію ЗПРМ і в панелі інструментів за допомогою кнопки  додати котушку. Перевести показчик на котушку і лівою клавiшею миші відкрити вікно «Операции». Вибрати функцію ЧТРМ «Читает реєстр модуля» і встановити параметри: № місця модуля в каркасі – 02; № реєстра другого каналу модуля CP31.02-01 – 04; зчитані дані – _Вход. Натиснути ОК.

Перевести показчик на функцію ЧТРМ і в панелі інструментів за допомогою кнопки  додати котушку. Двічі клацаючи на ній лівою клавiшею миші, у вікні «Операции», що з'явиться, відкрити вкладку «Поразрядные» і вибрати тип операції – NOT, із словника глобальних змінних вибрати ім'я _Вход і натиснути ОК.

Створити наступну котушку за допомогою кнопки . Двічі клацнувши по неї відкрити вікно «Операции» і у вкладці «Встроенные функции» вибрати FLOAT «Преобразование целого (со знаком) А в вещественное В». У вікні «Операции» встановити параметри: із словника глобальних змінних, вибрати ціле А – _Вход і дійсне В – _Вход1.

Додати наступну котушку, у вкладці «Встроенные функции» вибрати ЗПРМ «Записать в реєстр модуля» і встановити параметри: № місця модуля в каркасі – 02; № реєстра третього каналу модуля CP31.02-01 – 06; число 0. Натиснути ОК.

Додати наступну котушку і у вкладці «Встроенные функции» вибрати ЧТРМ з параметрами: номер місця - 02; № реєстра третього каналу модуля CP31.02-01 – 06; зчитані дані – глобальна змінна _Задатчик.

Створити нову котушку і у вкладці «Поразрядные» активізувати тип NOT і вибрати ім'я _Задатчик.

Додати наступну котушку. У вкладці «Встроенные функции» вибрати FLOAT. У вікні «Операции» встановити параметри: із словника глобальних змінних вибрати ціле А – _Задатчик і дійсне В – _Задатчик1.

Створити наступну котушку, у вкладці «Пересылки» вибрати тип операції MVss і встановити параметри: Приемник – _a_Control, Источник – 0.

Додати котушку і у вкладці «Встроенные функции» вибрати PIDP1 «пропорційно-інтегральний регулятор 1». Встановити параметри, що приведені на рис.7.6.

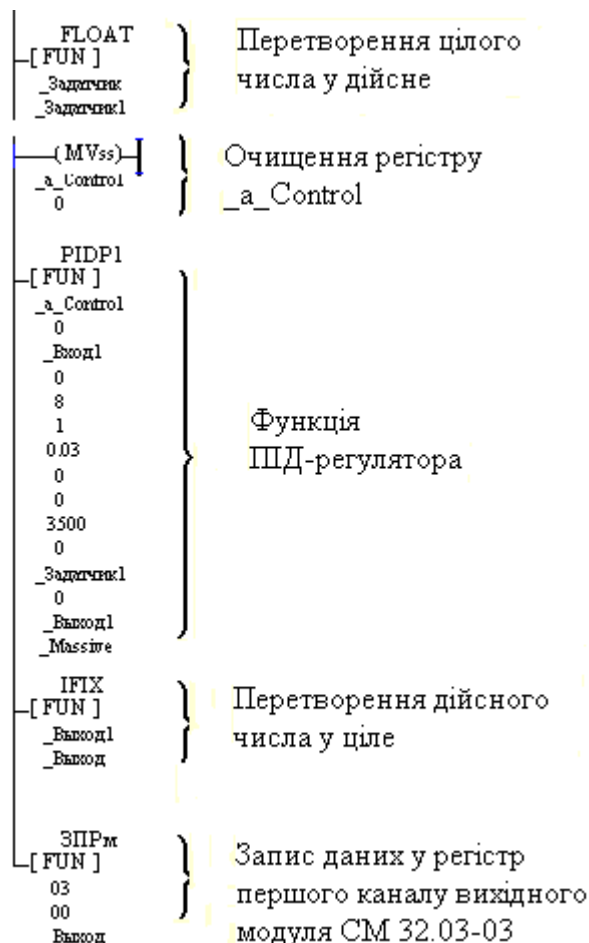
Додати котушку і у вкладці «Встроенные функции» вибрати IFIX «Преобразование действительного А в целое (со знаком) В». Як дійсне вибрати _Выход1, а ціле із знаком – _Выход.

Параметр	Значение	Описание
_a_Control	0	Control : Регистр Управления (Вх & Вых)
0	0	Номер контура регулятора (0...499)
_Вход1	0	Pt - контролируемый параметр
0	0	Revers : 0 отклонение (Sp-Pt); 1 (Pt-Sp)
8	8	Kp : кфт пропорциональности (Kp>=0)
1	1	Ki : интегральный коэффициент (Ki>=0)
0.03	0.03	Kd : дифференциальный кфт (Kd>=0)
0	0	Td : время запаздывания дифф. комп., сек
0	0	Umin : мин. значение вых. параметра Ut
3500	3500	Umax : макс. значение вых. параметра Ut
0	0	BIAS : вход смещения
_Задатчик1	0	Sp : уставка контролируемого параметра
0	0	Utman : заданное Ut для реж. Manual
_Выход1		Ut : выходной параметр
_Massive		Массив для копии данных PIDP1


Рисунок 7.6 – Параметры ПД-регулятора

Додати котушку і у вкладці «Встроенные функции» вибрати ЗПРм з параметрами: № місця – 03; № регістра першого каналу модуля виводу– 00; записані дані – _Выход1. Повністю програма має вигляд:





Виконати з'єднання з контролером і завантажити програму у ПЛК. За відсутності помилок увімкнути тумблер живлення печі 12, рис. 7.1, і задатчиком 11 встановити для регулятора задане значення температури, наприклад 30°C.

За допомогою кнопок на панелі інструментів , запустити контролер на виконання програми. Після завершення процесу регулювання зупинити контролер і розірвати з ним зв'язок, скориставшись пунктом меню «Контроллер». Вимкнути лабораторну установку.

Завдання для самостійної роботи

Модифікувати код програми керування ілюмінацією так, щоб на табло по черзі послідовно вмикалися то коло, то промінь.

Зміст звіту

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

Контрольні запитання

1. Для чого призначено контролер Констар К202 і його можливості?
2. Яка компоновка контролера на лабораторному стенді?
3. Які можливості має середовище програмування К748v2?
4. Як виконується в проєкті конфігурування контролера?
5. Як оголошуються змінні?
6. Як розробляється прикладна програма?
7. Як здійснюється налагодження проєкту?

Лабораторна робота №8

Програмування та дослідження роботи контролера WAGO 750-841

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

Лабораторна установка, рис.8.1, призначена для оволодіння методикою програмування і роботи з контролером 750-841 фірми WAGO.

У лабораторній установці контролер WAGO 750-841 забезпечує рух каретки вторинного приладу шляхом формування керуючого впливу на реле, які згідно з програмою підключаються 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 - мілівольтметр напруги термопари в нагрівальній печі; 7 - вторинний прилад РП-160.

Рисунок 8.1 – 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.8.2.

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.

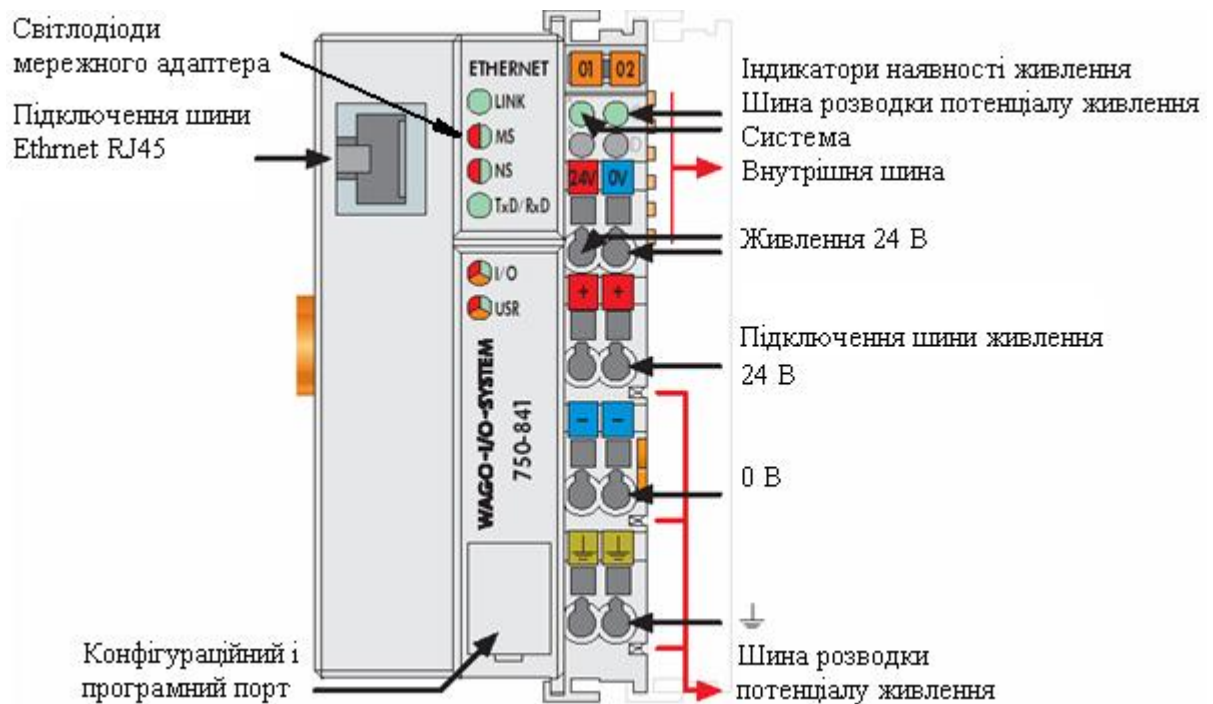


Рисунок 8.2 – Fieldbus контролер ETHERNET TCP/IP

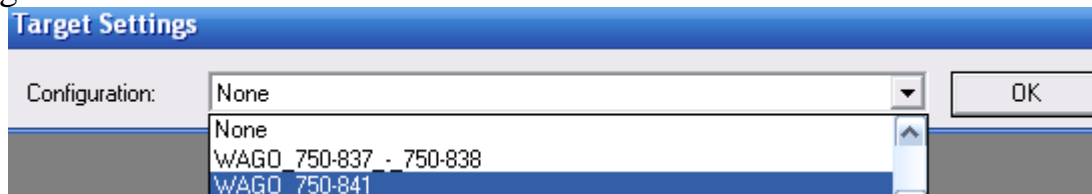
Fieldbus-вузол лабораторної установки складається з адаптера WAGO 750-841, модуля дискретного вводу WAGO 750-402, модуля дискретного виводу WAGO 750-504, модуля аналогового вводу WAGO 750-461, модуля аналогового виводу WAGO 750-554 і кінцевого термінального модуля WAGO 750-600.

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 OK.

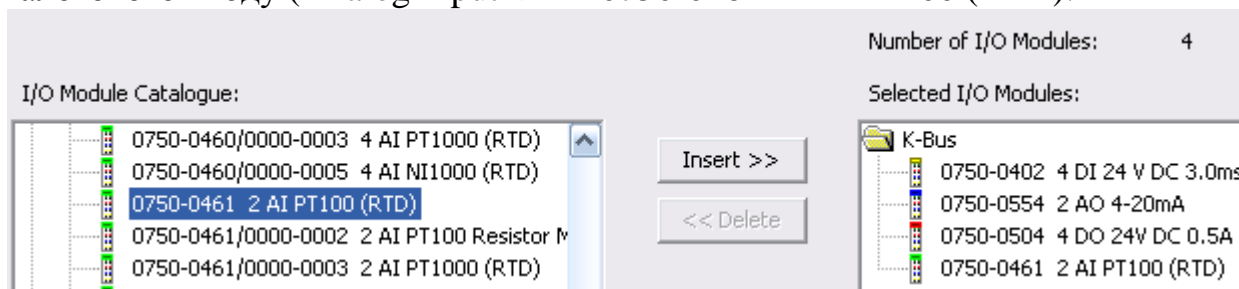
На екрані з'являться два вікна, ліворуч вікно організатора проекту POU's, а праворуч робоче поле для створення програмного коду.


Щоб здійснити конфігурування контролера, у нижній частині організатора проекту POU's 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)).

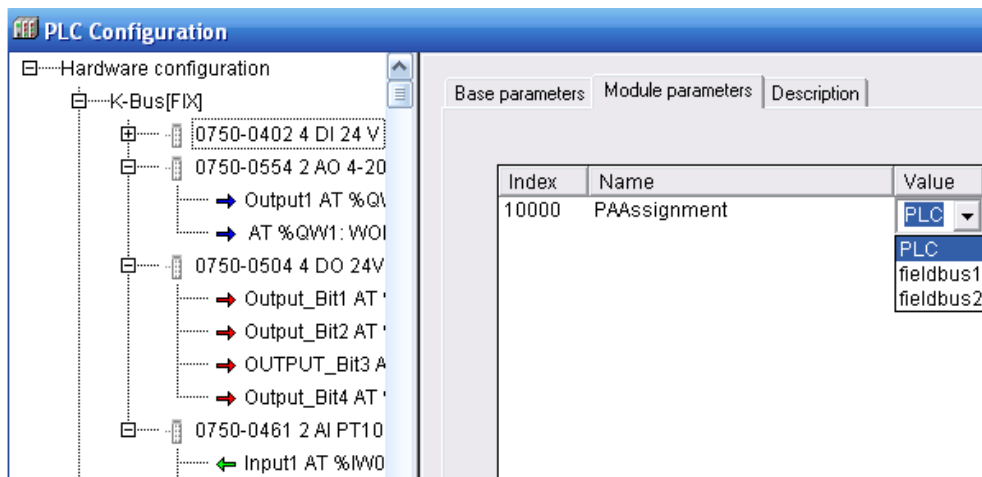


Якщо необхідно видалити деякі модулі, треба вибрати відповідний модуль в правій частині вікна конфігурації і натиснути кнопку “Delete”. Можна змінити позицію модуля, вибираючи і переміщаючи модулі, використовуючи кнопку , що знаходиться у правій частині вікна конфігурації. Кінцевий модуль 750-600 не 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.

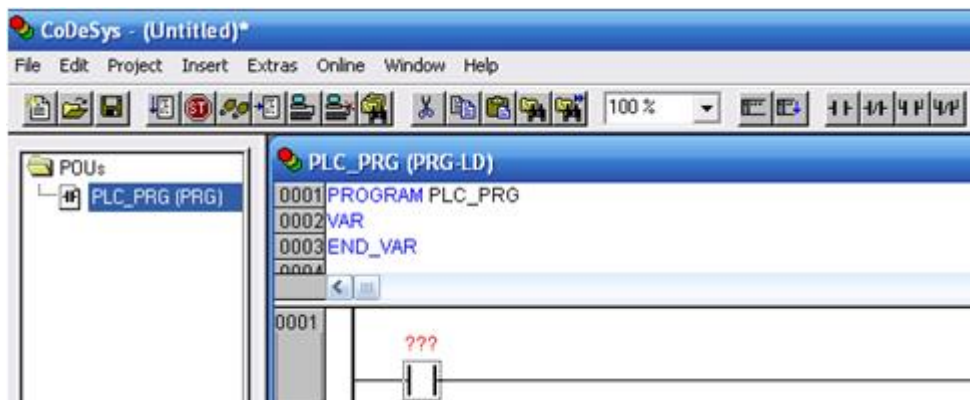
Після цього необхідно присвоїти імена вхідним і вихідним модулям контролера. Оскільки при керуванні використовується модуль дискретного виводу 0750-0504 двічі натиснути лівою клавішею по його назві.

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 POU's і подвійно клацнути лівою клавішею миші на PLC_PRG (PRG). Відкриється вікно з першою LD-мережею і відповідною панеллю інструментів. Щоб встановити в мережу перший контакт, натиснути лівою клавішею 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 OK. Змінна Start з'явиться у вікні глобальних змінних проекту.

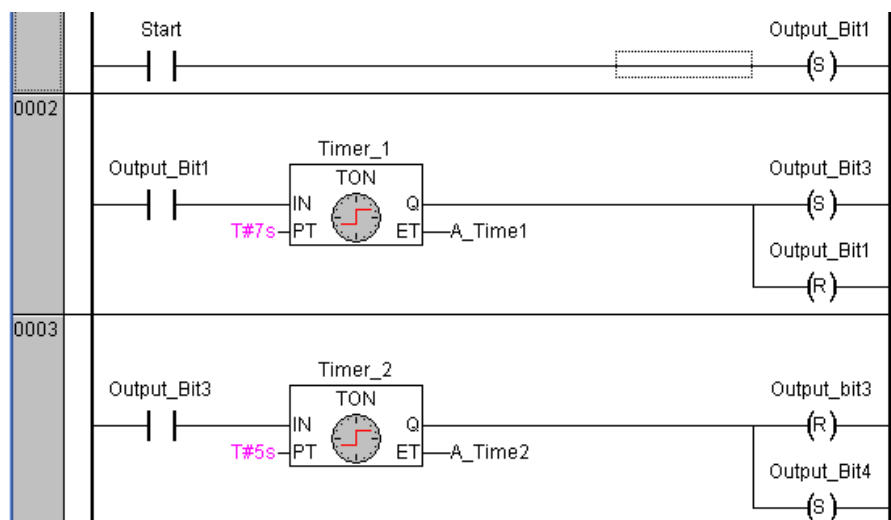
Щоб додати Set-котушку, позначити лівою клавішею миші LD-мережу і натиснути лівою клавішею миші на кнопку  (coil), яка знаходиться на панелі інструментів. Змінити ім'я котушки на Output_Bit1 (котушка буде зв'язана з першим каналом модуля дискретного виводу).

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 OK. 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 OK. 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 клав'яшею 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.

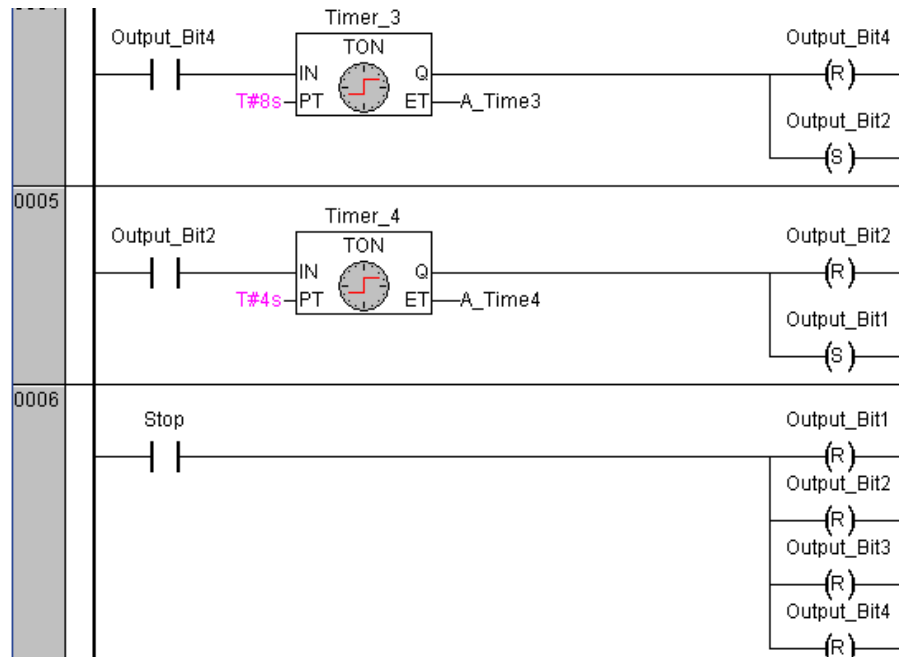
Створити ще одну мережу кнопкою . Додати контакт (contact) Output_Bit3, встановити таймер TON з ім'ям Timer_2. Замінити три знаки питання на вході PT константою T#5s, а на виході ET встановити змінну A_time2 типу TIME. Розмістити на цій мережі Reset- і Set-котушки з іменами Output_Bit3 і Output_Bit4 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:


```

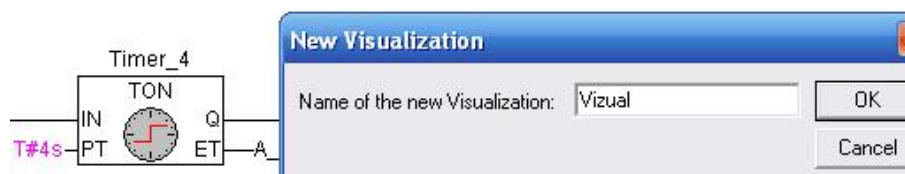
PLC_PRG (PRG-LD)
0001 PROGRAM PLC_PRG
0002 VAR
0003   Start: BOOL;
0004   Timer_1: TON;
0005   A_Time1: TIME;
0006   A_Time2: TIME;
0007   Output_bit1 AT %QX2.0: BOOL;
0008   Output_bit2 AT %QX2.1: BOOL;
0009   Output_bit3 AT %QX2.2: BOOL;
0010   Output_bit4 AT %QX2.3: BOOL;
0011   Timer_3: TON;
0012   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. 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.

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.

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 клавішею миші по прямокутнику. З'явиться діалогове вікно для настройки елементу візуалізації. У Category вибрати альтернативу Text і заповнити поле Contents ім'ям змінної Start:

Перейти в категорію змінних (Variable), клацнути лівою клавішею 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 елементу (колір елементу 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 відображатиметься 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_4:%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.

У меню Online відкрити діалог Communication parameters. Натиснути кнопку New для налаштування нового сполучення, 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.

Переконалися 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. Фрагмент робочого поля під час перевірки роботи програми приведений на рис.8.3.

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.

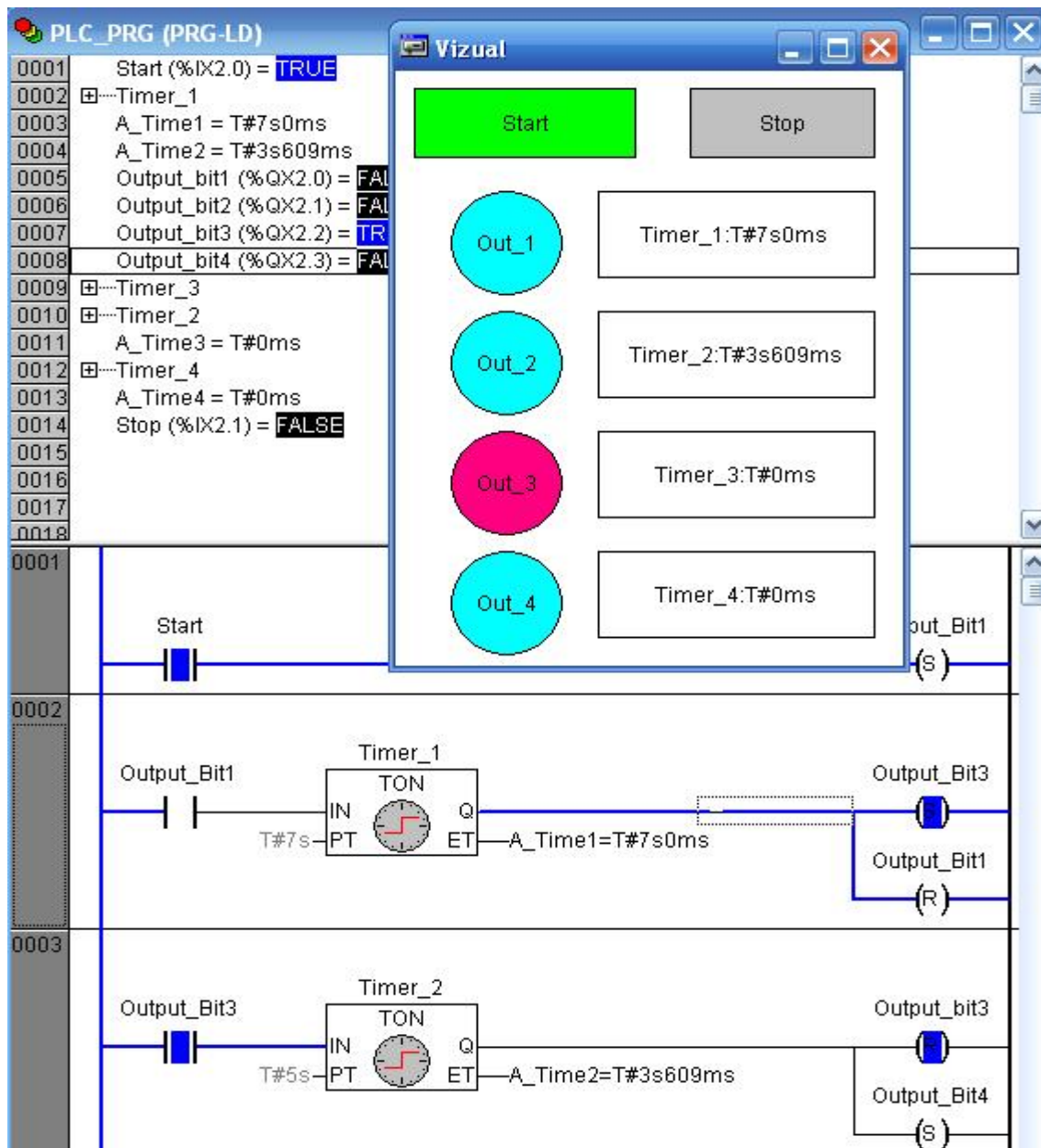


Рисунок 8.3 – Фрагмент робочого поля під час перевірки роботи програми

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 налаштування 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?

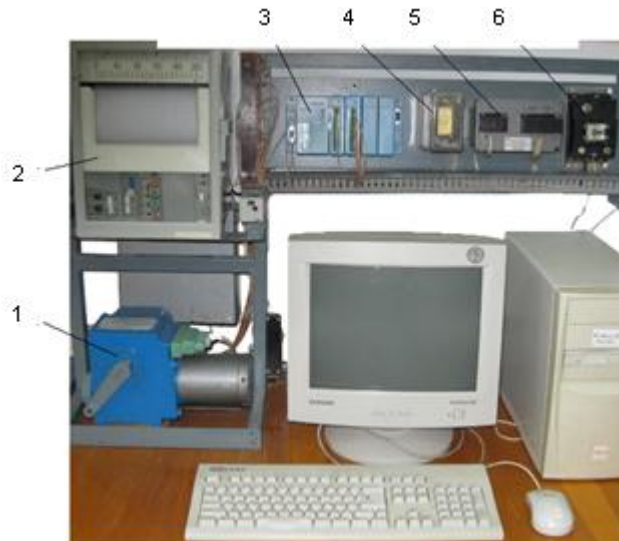
Лабораторна робота № 9

Програмування та дослідження роботи контролера ADAM-5510KW

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

Лабораторна установка, рис.9.1, 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 9.1 – Лабораторна установка

Контролер ADAM 5510KW

Мікроконтролер ADAM 5510KW складається з двох основних частин- базового блока 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 установлення 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».

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 OK.

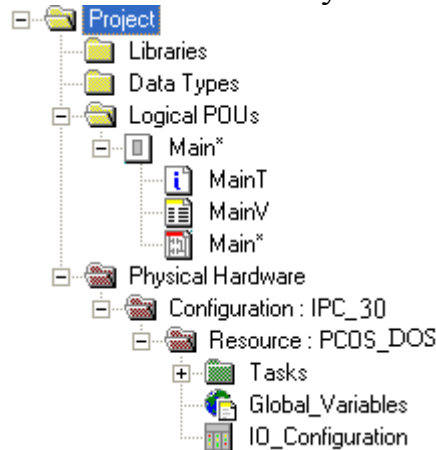
Натиснути кнопку Next і у рядок Name of POU (Ім'я POU) ввести ім'я першого організатора програми (POU), наприклад Main (Головна). Вибрати FBD-мову. Натиснути кнопку Next і ввести ім'я конфігурації, наприклад, Configuration.

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 OK.

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 клавішею по Ressours: PCOS_ DOS, а у підменю, що з'явиться, клацнути по "Properties..." і змінити ім'я на ADAM 5510KW, натиснути OK.

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

Клацнути правою клавішею 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	
<input type="checkbox"/> 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 установлюється 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 термодатчиків.

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

Щоб установити серійний порт треба натиснути кнопку “ADAM5510KW system setup” і вибрати для модуля вводу формат даних (Data Format) – NOPARITY,8,1. Клацнути OK 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 установити серійний порт треба натиснути кнопку “ADAM5510KW system setup” і в Data Format (Формат даних) вибрати NOPARITY,8.1 для модуля вводу-виводу. Натиснути кнопку OK 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 клавішею 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 клавішею 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 клавішею 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; ← – означає одинбітний розмір.

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

Двічі клацнути по котушці C001, і у діалоговому вікні Contact/Coil Properties (Властивості Контакт/котушка), 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; ←— означає однобітний розмір.

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

Для установлення нової мережі, клацнути лівою клавішею 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 клавішею миші. Натиснути OK 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 клавішею миші, з'явиться діалогове вікно Variable properties (Властивості змінної). У рядку Name (Ім'я) ввести M_Time і послідовно натиснути кнопки Apply (Застосувати) і OK, підтверджуючи завершення діалогу.

Для визначення часу роботи таймера лівою клавішею 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).

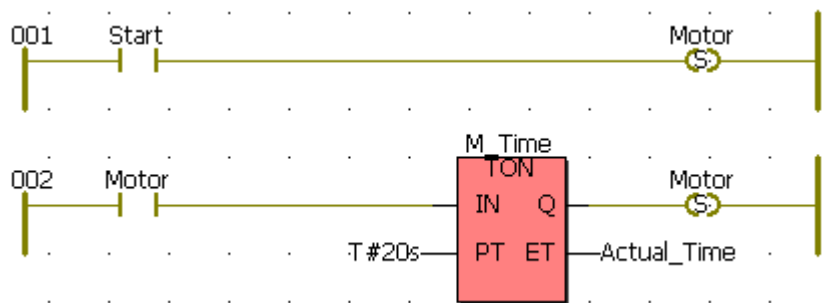
Клацнути OK і на RT-вході лічильника з'явиться константа часу.

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

Натиснути кнопку Apply (Застосувати), відкриється сторінка діалогу Common (Загальний). Зі списку Data_Type (Тип даних) вибрати Time і натиснути OK. На ET-виході таймера з'явиться Actual_Time.

Двічі клацнути по котушці C003, 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 OK.

На робочому полі залишиться код програми:



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

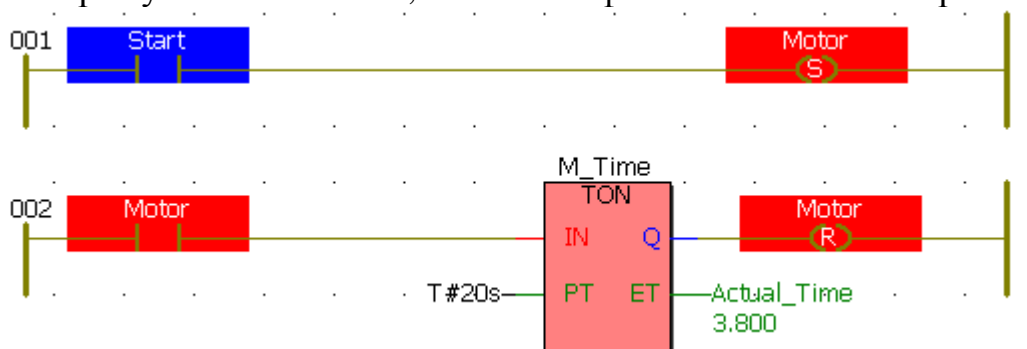
Завантаження проекту до ПЛК

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 (←олоднийстарт), Warm start(Теплий старт) або Hot start(Гарячий старт) у вікні Resource (Ресурс).

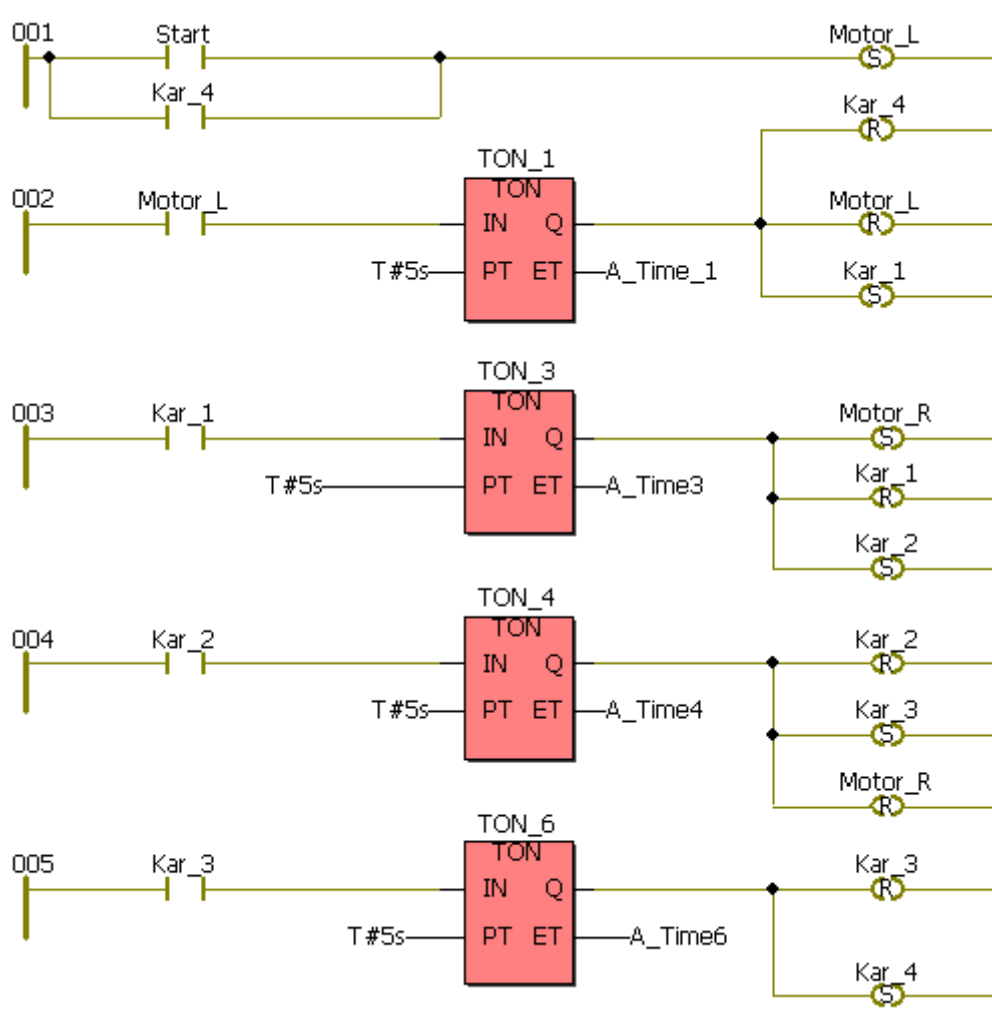
Щоб відстежувати роботу програми, можна скористатися оперативним налагоджувальним режимом. Для цього в панелі інструментів активізуйте ікону Debug on/of (Налагодження on/of). Всі змінні проектного коду, що знаходяться у хибному стані отримують синій фон. Коли змінна протягом виконання програми отримує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



Завдання2

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

- де: Kar_1(%QX1.3), Kar_2(%QX1.4), Kar_3(QX1.5), Kar4_(%QX1.6) – вихідні змінні, що визначають положення каретки на вторинному приладі;
 Start(%IX0.0) - 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.



Завдання для самостійної роботи

Модифікувати програму таким чином, щоб після п'яти циклів роботи вона зупинялась.

Зміст звіту

Звіт повинен містити назву роботи, короткі відомості про контролер, програми керування виконавчим механізмом і кареткою приладу та 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 система програмування MULTIPROG?
5. Як створюється проектний код?
6. Яке призначення Майстра проекту?
7. З яких етапів складається програмування контролера?
8. Як завантажується проект у контролер?
9. Що таке ресурс програми і для чого він потрібен?
10. Як здійснюється налагодження програми?

Лабораторна робота №10

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.10.1, призначена для дослідження роботи регулювального мікропроцесорного контролера Реміконт Р-130, який підтримує температуру в печі за ПІД – аналоговим алгоритмом.



1 - пульт настройки ПН-1; 2 - блок контролера БК-1; 3 - задатчик 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 10.1– Лабораторна установка

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».

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 силову напругу на електричній нагрівальній печі. Струм навантаження показує амперметр.

Реміонт Р-130

Р-130 – це комплекс технічних засобів, радянського виробництва, 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 лицевої панелі ПЛ, доставляє потрібні алгоритми із ПЗП. Ці алгоритми об'єднуються у систему заданої конфігурації, і в них встановлюються необхідні параметри настройки. За допомогою лицевої 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 лицевої 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.

Реміонт 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.

Реміонт 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.

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

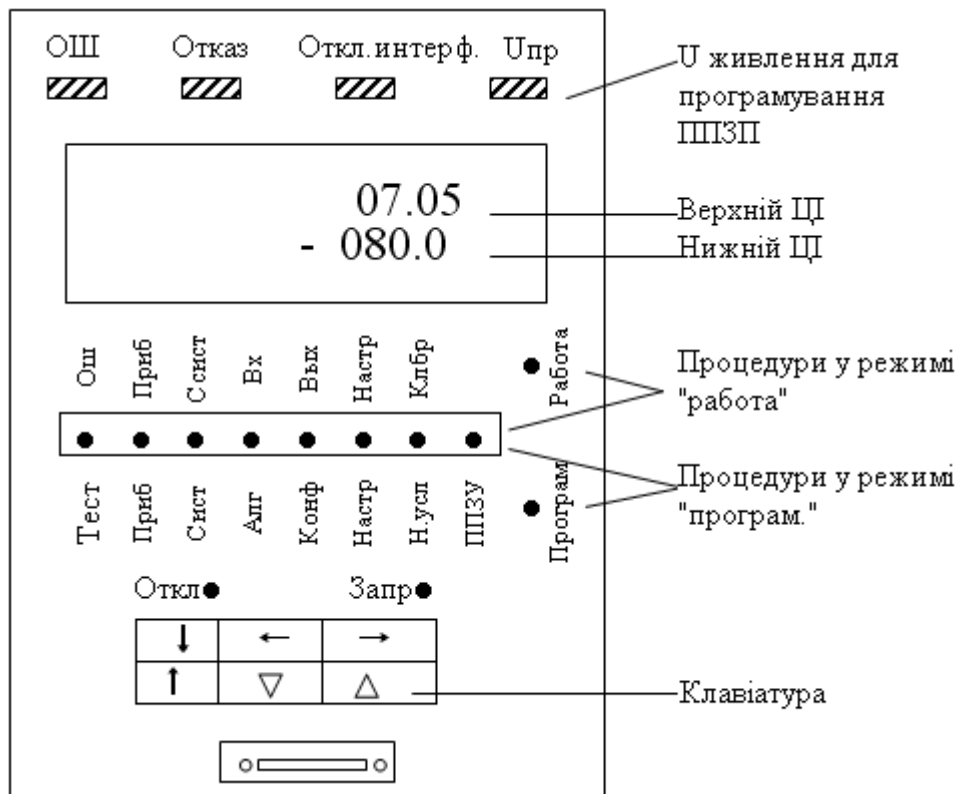


Рисунок 10.2 – Пульт настройки ПН-1

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 встановлюються 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, встановлюється 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”) - встановлюється 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”) - встановлюються значення параметрів настройки.

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

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

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 установленні 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.10.1.

Таблиця 10.1

Приладові параметри

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

У процедурі “сист” (системні параметри) задаються параметри, що визначають особливості взаємодії контролера з іншими пристроями за інтерфейсним каналом, табл. 10.2. 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.

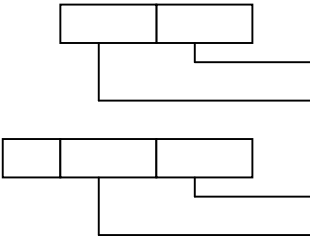
Таблиця 10.2

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 пам’яті 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.). Зазвичай у перші чотири алгоблоки розміщуються блоки оперативного контролю. Після четвертого алгоблока відкривається доступ до будь-якого алгоблока. Правила установлення алгоритму приведені у табл. 10.3.

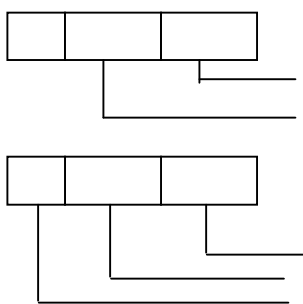
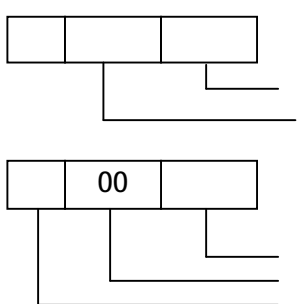
Таблиця 10.3 Параметри алгоблоків

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

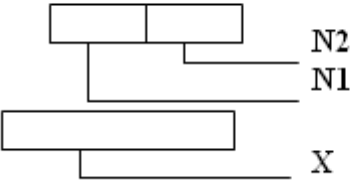
Правила конфігурування наведені у табл. 10.4.

Таблиця 10.4 Параметри конфігурування

Стан входу	Зображення інформації	Параметри
Зв’язане		N1 – номер алгоблока-приймача N2 – номер входу алгоблока-приймача N3≠00 - номер алгоблока-джерела N4 – номер виходу алгоблока-джерела C – признак інверсії C=“темне поле” – без інверсії C= “.” – інверсія
Вільне		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

У процедурі “настр.” (настройка) установлюються значення параметрів настройки - констант і коефіцієнтів. Ця процедура виконується лише для входів алгоблока, які у процедурі конфігурування були визначені як вільні. Якщо параметри настройки не задаються, вони приймають початкові значення, які залежать від виду алгоритму. Правила установлення параметрів настройки наведені у табл.10.5.

Таблиця 10.5 Параметри настройки

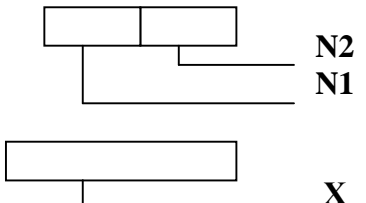
Зображення інформації	Параметри
	<p>N1 – номер алгоблока N2 – номер входу X – значення константи або коефіцієнта</p>

У процедурі “н.усл” (початкові умови) установлюються 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 установлення початкових умов наведені у табл.10.6.

Таблиця 10.6 Початкові умови

Зображення інформації	Параметри
	<p>N1 – номер алгоблока N2 – номер входу X – 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. 10.1.
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".
4. Запрограмувати перший канал контролера відповідно до схеми на рис. 10.3

4.1. На пульту ПН-1 за допомогою клавіші "→" вибрати процедуру "приб." і, користуючись табл. 10.1, виконати усі запропоновані операції.

4.2. Перейти у режим "работа" і переконаватися, 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 установити процедуру "сист" і виконати операції, відповідно до табл.10.2.

4.4. На пульту ПН-1 установити процедуру "алг"(алгоритми) і по-черзі, починаючи з першого алгоблоку, ввести алгоритми і їх реквізити, табл. 10.3.

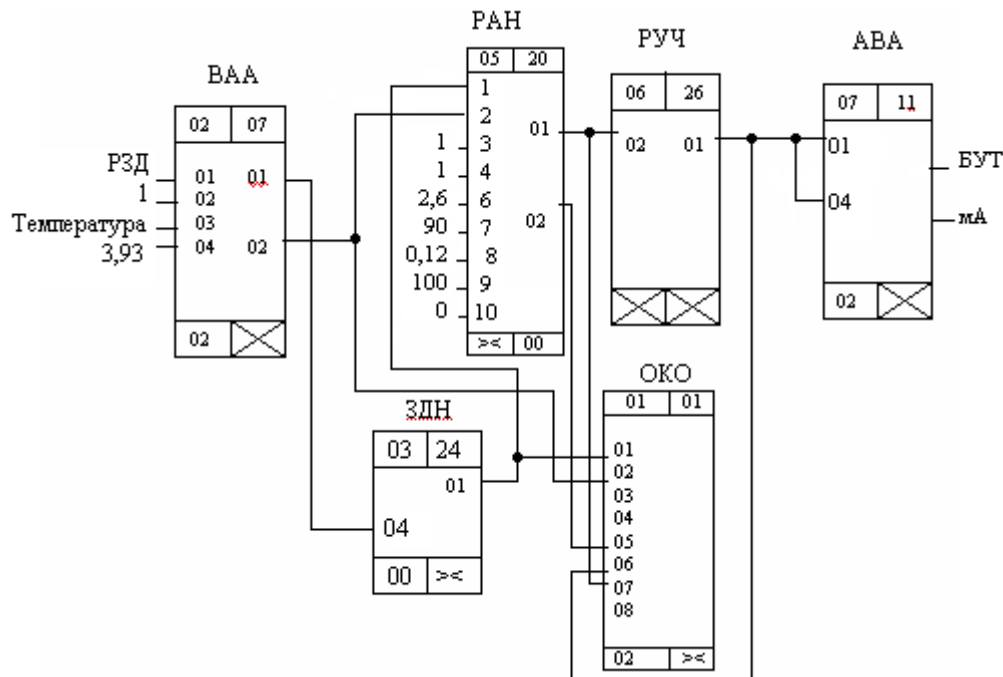


Рисунок 10.3 – Схема конфігурації Реміонта Р-130

4.5. На пульту ПН-1 установити процедуру "конф"(конфігурування) і для входів усіх алгоблоків, що повинні бути зв'язаними, установити конфігурацію, табл.10.4.

4.6. На пульту ПН-1 установити процедуру "настр"(настройка) і для входів усіх алгоблоків, де повинні задаватися коефіцієнти, установити ці параметри, табл.10.5.

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 "работа" одночасним натисканням на клавіші "↓" і "←" і переконаватися, 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 "работа" 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 БК-1, і

вторинним приладом РП160, який показує температуру печі. Змінюючи коефіцієнти настройки алгоблока РАН дослідити якість регулювання.

Клавіші лицевої панелі контролера:

- “←, →” – клавіші вибору параметра контролю;
- “Δ, ∇” – клавіші зміни параметрів або завдання, “більше”, “менше”;
- “N” – номер контуру;
- “↪” – клавіша переходу на зовнішнє завдання;
- “O” – клавіша переходу на один з режимів САР (локальний, каскадний, дистанційний), разом з клавішами “↑”, “↓”, “▲”, “▼”;
- “↙↘” – програмний задавач; разом з “▲”, “▼” змінюється завдання або номер програми;
- “□” – ручний задавач; разом з “▲” або “▼” змінюється завдання;
- “⏏” – режим ручного керування.

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

Завдання для самостійної роботи

Запрограмувати і дослідити роботу Р-130, який керує температурою печі за ПІД-законом, відповідно до наступного алгоритму:

Протягом 10 хвилин здійснюється поступовий нагрів печі до 60 градусів, після 5-ти хвилинної витримки температури протягом 8 хвилин зменшується до 50 градусів і надалі підтримується на цьому рівні 5 хвилин.

Зміст звіту

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

Контрольні запитання

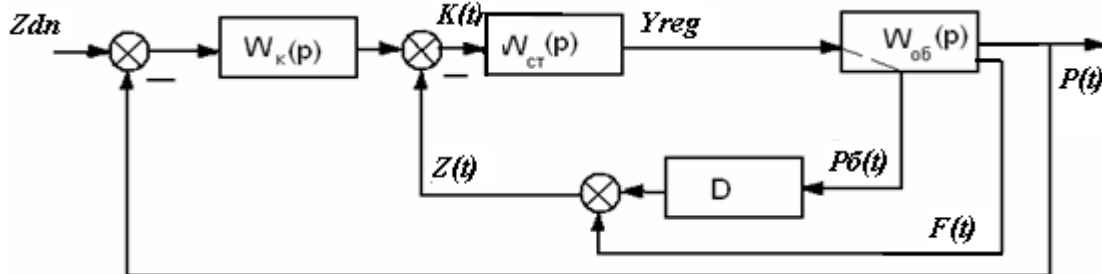
1. Яка фізична структура Р-130?
2. Що таке віртуальна структура Р-130 і з чого вона складається?
1. Якими реквізитами характеризується алгоблок?
4. Яке призначення алгоритму ОКО?
5. Якою мовою програмується Р-130 і в чому особливості програмування?
7. Як пульт настройки перевести у робочий режим, а як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

Лабораторна робота №12

Створення проектів у системі програмування Concept

Мета: ознайомитися з методикою створення проектів для контролерів компанії Schneider Elcttric у системі програмування Concept.

Виконати конфігурування і FBD-програмування контролера Modicon TSX Quantum відповідно до алгоритму:



Регулювання тиску за парогенератором $P(t)$, відповідно до заданого значення Zdn , здійснює коригуючий регулятор $Wk(p)$, який формує завдання стабілізуючому регулятору $Wcm(p)$. Вихідний сигнал стабілізуючого регулятора $Yreg$ змінює витрату палива в топку для підтримки на заданому значенні кількості тепла $Z(t)$, сприйманого поверхнями нагріву котла. Кількість тепла $Z(t)$ визначається за сумою витрати пари $F(t)$ і похідної від тиску в барабані котла $P6(t)$.

Створення проекту

Запустити програму Concept і в головному меню вибрати шлях File-> New project. Після появи на моніторі екрану нового проекту увійти в підміню File->New section. У діалоговому вікні New Program Section вибрати FBD-редактор програмування і присвоїти ім'я секції. Після натискання кнопки ОК відкриється робоче поле у вигляді логічної сітки для редагування програми користувача.

Для конфігурування ПЛК увійти в меню Project->Configurator і в діалоговому вікні PLC Configuration в кожному полі зробити необхідні уставки.

Поле активується після установалення на ньому курсору і подвійного клацання лівої клавіші миші. Уставки поля PLC є допуском для роботи з рештою полями.

Після активації поля PLC відкриється вікно PLC Selection.

У рядку PLC Family (Серія контролера) вибрати контролер – Quantum, а в CPU/Executive (ЦПУ/Система виконання) - тип процесора, який за власними характеристиками відповідає вимогам задачі, що розв'язується, наприклад 140CPU 113 03. Runtime (Робочий цикл) вибрати Enable (Дозволений) і підтвердити встановлені параметри кнопкою ОК.

У полі Ranges (Інтервали) відкрити діалогове вікно PLC Memory Partition (Розподіл пам'яті ПЛК) і визначити достатні адресні інтервали для розміщення вхідних-вихідних бітів/маркерів дискретних (Discretes) сигналів і вхідних регістрів, регістрів зберігання аналогових (Registers) сигналів. За замовчуванням у вікні пропонуються максимальні діапазони, які можна залишити не змінними.

Вікно поля Loadables (Завантажування) містить спеціальні функції, що доступні для завантаження в ПЛК. Оскільки вибраний центральний процесор

140CPU 113 03 не має математичного співпроцесора програма автоматично завантажить дві функції @1SE V196 і @2IE V196.

Поле Segment scheduler (Диспетчер сегментів) відкривається при конфігуруванні віддалених каналів, щоб встановити послідовність і тип обробки програмних секцій, створених мовою LL984. Кожному каналу призначається секція. Тому диспетчер сегментів не може мати кількість секцій менше кількості віддалених каналів. Оскільки проект створюється для локальної конфігурації, кількість секцій дорівнює 1.

Поле Specials (Спеціальні) відображає функціонально незалежні уставки. У його діалоговому вікні:

- Battery Coil. Уставка визначає адресу біта в області пам'яті, яка характеризує стан батареї.

- Timer Register. Уставка визначає адресу регістра таймера, вміст якого збільшується на 1 кожні 10 мс.

- Time of Day. Уставка реального часу призначена тільки для додатків MMI-інтерфейсу.

- Watchdog Timeout. Уставка в інтервалі часу від 2 до 255мс формує сторожовий імпульс таймера.

- Online Ending Timeslice. Уставка в інтервалі 3-30мс забезпечує час спостереження за комунікаційними зв'язками між пристроями мережі.

У даному проекті приведені опції не встановлюються.

У діалоговому вікні поля Configuration Extensions (Конфігураційні розширення) дозволяються або блокуються додаткові позиції меню конфігурації:

- Data Protection- уставка активує захист даних від доступу;

- Peer Cop- уставка активує режим постійного зв'язку між вузлами однієї мережі;

- Hot Standby- активується, коли для підвищення надійності використовуються два дубльованих ПЛК.

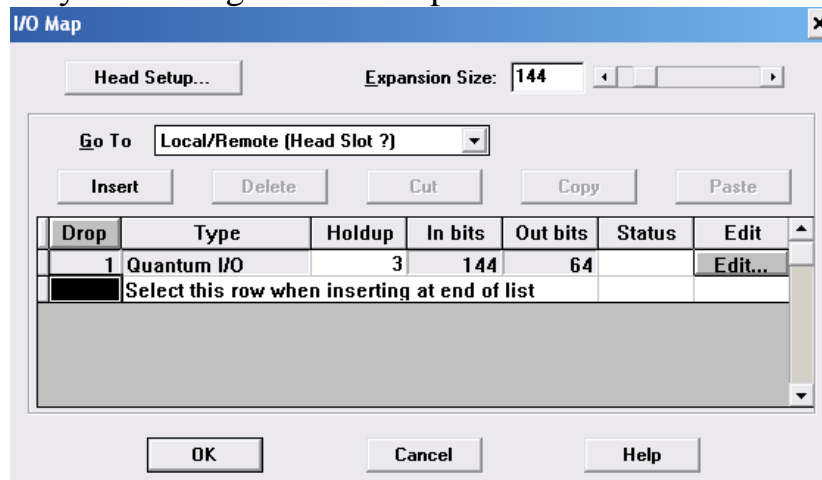
Оскільки ці опції в проекті не використовуються, активувати їх немає потреби.

У вікні поля ASCII встановлюються уставки, що регламентують ASCII-повідомлення, при програмуванні мовою LL984.

Остаточне вікно PLC Configuration має вигляд:

PLC Configuration			
PLC			
Type:	140 CPU 113 03	Available Logic Area:	1100
Exec Id:	871	Extended Memory:	—
Memory Size:	16K		
Ranges		Loadables	
Coils:	000001 - 001536	Number installed:	2
Discrete Inputs:	100001 - 100512		
Input Registers:	300001 - 300512		
Holding Registers:	400001 - 401872		
Specials		Segment Scheduler	
Battery Coil:	001500	Segments:	2
Timer Register:	400100		
Time of Day:	—		
Config Extensions		ASCII	
Data Protection:	Disabled	Number of Messages:	0
Peer Cop:	Disabled	Message Area Size:	0
Hot Standby:	Disabled	Number of Ports:	0
Ethernet:	0		
Profibus DP:	0		

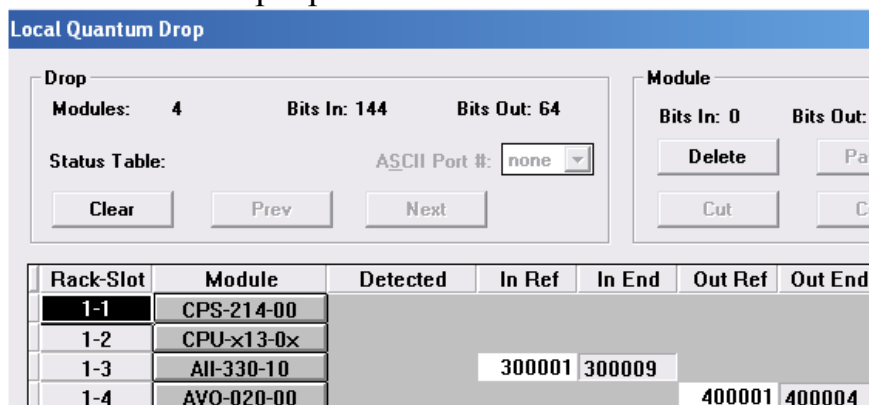
Конфігурування модулів вводу-виводу, а також задіяних каналів здійснюється за допомогою I/O Map... карти, яка викликається в меню вікна PLC Configuration через пункт Configure / I/O Map...:




Перед вибором модулів необхідно у списку Go To визначитися з видом конфігурації (локальна, RIO або DIO). При ініціалізації області пам'яті контролера для даних вводу-виводу кожного виду модуля використовуються відповідні посилання:

- посилання, що починаються з цифри 3 (3-посилання), вводяться для модулів аналогового вводу;
- посилання, що починаються з цифри 4 (4-посилання), вводяться для модулів аналогового виводу;
- посилання, що починаються з цифр 3 або 1 (3-посилання або 1 - посилання), вводяться для модулів дискретного вводу;
- посилання, що починаються з цифр 4 або 0 (4-посилання або 0-посилання), вводяться для модулів дискретного виводу.

Карта апаратного забезпечення формується у діалоговому вікні Local Quantum Drop, яке з'являється після натиснення кнопки Edit. Відкриваючи список модулів відповідного слота, вибираємо по черзі модуль живлення CPS-214-00, центральний процесор CPU-x13-0x, модуль аналогового вводу АП-330-10, призначений для роботи з вхідними сигналами датчиків витрати і тиску пари, а також аналогового виводу AVO-020-00 для впливу на виконавчий пристрій, що змінює витрату палива. Модулям вводу-виводу вказати початкові адреси. Кінцеві визначаються програмою автоматично.




Натиснути кнопку  і по кожному каналу модуля вводу-виводу визначити характеристику вхідного або вихідного сигналу.

Для збереження проекту з вибраною конфігурацією у меню натиснути File->Save projectas.


Після завершення редагування полів діалогового вікна PLC Configuration воно закривається і на екрані EOM залишається робоче поле для розробки програмного коду.


Створення програми

Для роботи в FBD-редакторі призначена панель інструментів  і декілька бібліотек, в яких по групах розподілені функціональні блоки.


Відповідно до наведеної структурної схеми, для розробки програмного коду потрібні такі функціональні блоки:

- PI (ПІ-регулятор) і блок DERIV (диференціатор) з бібліотеки CONT_CTL;
- SUB_REAL (відняти значення типу REAL), ADD_REAL(скласти 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.

Для установлення функціонального блока PI у робоче поле секції необхідно кнопкою  відкрити діалогове вікно FFBs from Library і за допомогою кнопки Library... вибрати бажану бібліотеку CONT_CTL:

Щоб витягнути блок з бібліотеки потрібно позначити у вікні Group групу CLC_PRO, а у вікні EFB Type необхідний блок PI і закрити вікно. Після цього вибрати на екрані EOM місце установлення блока на робочому полі і натиснути ліву клавішу миші. Для відновлення покажчика слід скористатися кнопкою .

Таким самим чином установити інші функціональні блоки. При цьому розташування їх на робочому полі має бути зручним для роботи з програмою.

Для з'єднання блоків необхідно скористатися кнопкою  і за допомогою миші накреслити сполучні лінії поміж відповідними входами і виходами.

Щоб оголосити змінні, необхідно мишею встановити курсор на вхід або вихід, розміщеного на робочому полі функціонального блока і двічі натиснути ліву клавішу. Відкриється вікно Connect FFB:

Необхідно активувати Variable (Змінна), Literal (Число) або Direct address (Пряма адреса) і вказати ім'я змінної, числове значення або пряму адресу.

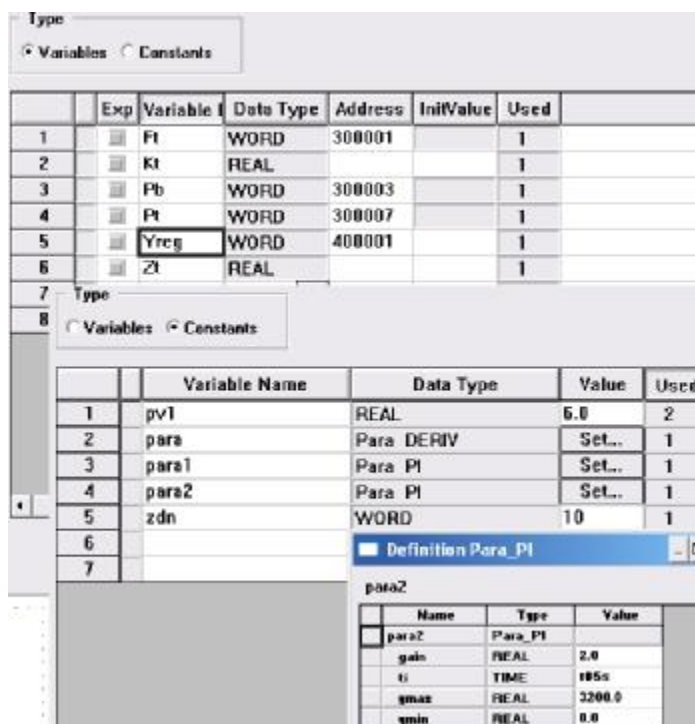
Після цього натиснути кнопку Variable declaration (Опис змінної) і у вікні, що відкриється, встановити вид (змінна або константа) і тип даних, а в колонці Adress вказати адреси пам'яті для запису значень змінних відповідно до умови:

- 4 – аналогові виходи.
- 3 – аналогові входи.

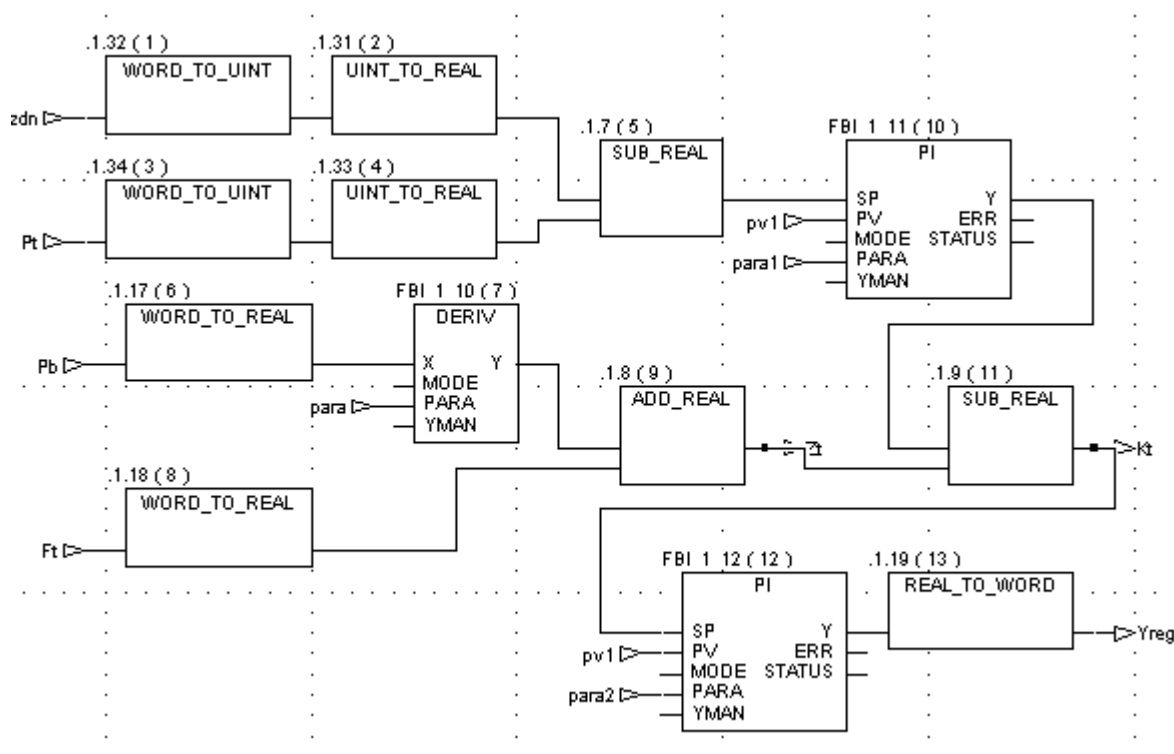
Щоб додатково оголосити параметри комплексних змінних функціональних блоків DERIV і PI, необхідно клацнути лівою клавішею миші

по кнопці Set і у вікні Definition, що відкриється, записати їх значення, керуючись бажаною якістю регулювання.

Після оголошення усіх змінних діалогове вікно Variable Editor має вигляд:



Реалізація програми FBD-мовою має наступний вигляд:



Для перевірки синтаксису програми у головному меню пройти шлях Project->Analyze section. Якщо помилки виявляться – виправити їх.

Запустити 32-бітовий симулятор через Пуск/Все программы/Concept V2.2 XL EN/ Simulator 32-Bit. (або він автоматично відкривається після запуску контролера).

У назві вікна симулятора, що з'явиться, показане ім'я утиліти PLC Sim32 і адреса TCP/IP плати інтерфейсу ПК. Вікно симулятора складається з 3 закладок:

- State RAM - показує статус буферної оперативної пам'яті;
- I/O modules - показує завантажену конфігурацію;
- Connections - показує поточні сполучення з модельованим ПЛК.

Щоб з'єднати проект з PLC, необхідно увійти в меню Online->Connect. У вікні, що з'явиться, вибрати тип протоколу IEC Simulator (32-bit), позначити альтернативу Change Configuration і підтвердити зроблені уставки.

Після встановлення логічного з'єднання ПК з ПЛК здійснюється тестування на еквівалентність параметрів конфігурації системи Concept з реальними апаратурними ресурсами ПЛК. У разі їх невідповідності з'явиться повідомлення про помилку.

За допомогою меню Online->Download відкрити вікно завантаження проекту, позначити лівою клавішею миші позиції «IEC program section», «State RAM», «Initial values only» і натиснути кнопку Download. Програма запропонує запустити контролер, натиснувши кнопку «Так». Запустити і зупинити контролер можна також через меню Online – Online control panel.

Щоб підготувати проект до симуляції на клавіатурі ЕОМ натиснути клавіші «Ctrl» і «A». При цьому на екрані в програмному коді зміниться колір ліній сполучення на синій. Використовуючи в меню шлях Online->Animate selected перейти в режим анімації. Як наслідок, у програмному коді на жовтому фоні ліній сполучення з'являться значення сигналів. Лівою клавішею миші клацнути по назві вікна симуляції PLC Sim32, що заховане у нижній частині екрану.

Повзунком на симуляторі №3 регістру пам'яті дискретних змінних задати певний вхідний сигнал. У програмному коді, значення вихідної змінної почнуть змінюватися відповідно до налаштувань функціонального блока ПІ-регулятора. Для вибору потрібного регістра треба клацнути лівою клавішею миші по одному з двох номерів симулятора №3 і у вікні Input Register, що з'явиться, записати потрібний номер регістра.

Зовнішній вигляд екрана проекту в процесі симуляції наведено на рис.12.1

Закриття симулятора Simulator 32-Bit під час роботи контролера (у режимі Connect...) приведе до збою системи і несподіваних наслідків.

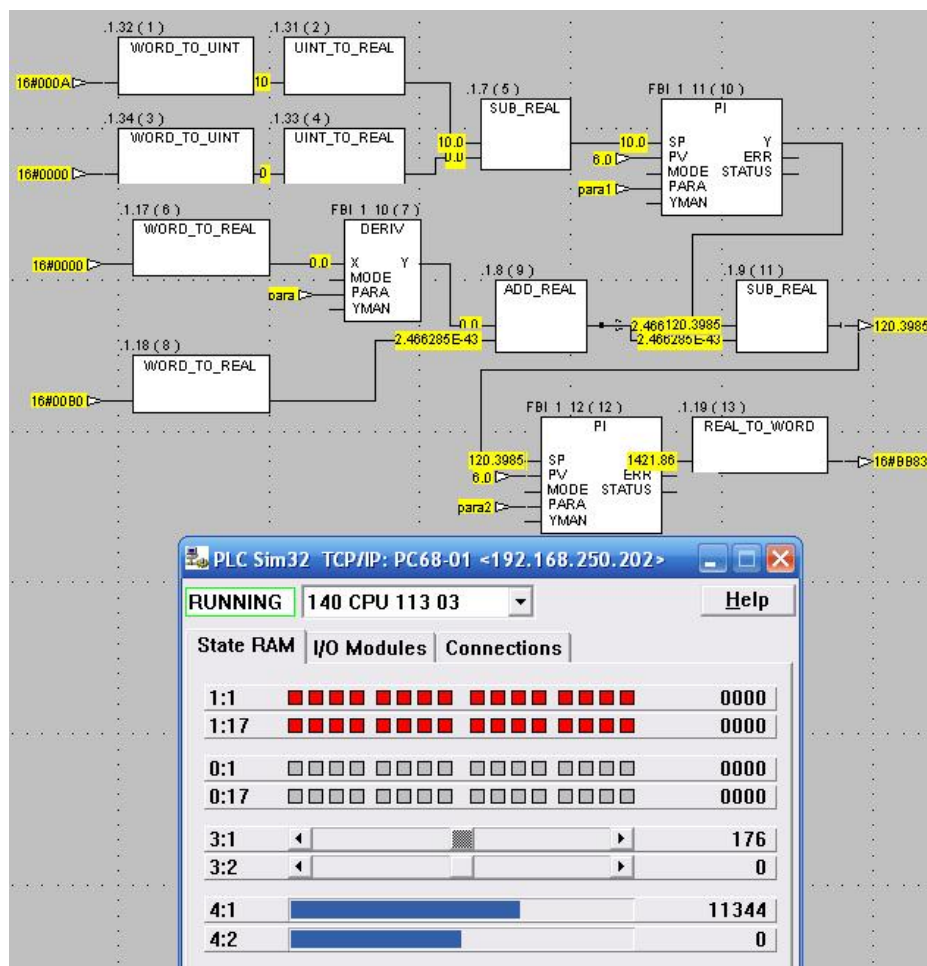


Рисунок 12.1– Екран проекту в процесі симуляції

Щоб виконати зупинку контролера необхідно в меню Online control panel натиснути кнопку Stop controller....

А після підтвердження рішення про зупинення контролера закрити вікноOnline Control Panel кнопкою Close.

Для вивантаження програми з контролера використати меню Online –> Upload... – і у вікні, що з'явиться, позначити позицію Configuration.

Після натиснення кнопки Upload, зв'язок між ПЛК і ПК зникне.

Завдання для самостійної роботи

Розробити програму послідовного, із затримкою у три секунди, вмикання в роботу трьох двигунів, які зупиняються одночасно за командою «Стоп».

Зміст звіту

Звіт має містити проект для контролера Modicon TSX Quantum, створений відповідно до завдання на самостійну роботу.

Контрольні запитання

1. Яка послідовність дій при розробці нового проекту?
2. Як здійснюється конфігурування контролера?
3. Як створюється програмний код?
4. Як здійснюється відладка програми?
5. Як відбувається завантаження проекту у контролер?

Лабораторна робота №13

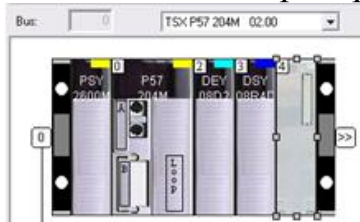
Створення проектів у системі програмування Unity PRO XL

Мета: ознайомитися з методикою створення проектів для контролерів компанії Schneider Elcttric у системі програмування Unity PRO XL.

Розробити ST-мовою проект керування роботою механізму, який автоматично по черзі рухається у протилежних напрямках. Час роботи у лівий бік складає 3с, а у правий - 2с. Кожне вмикання двигуна підраховує лічильник, а 5-те, 10-те і 15-те супроводжується повідомленням. Після 20 вмикань лічильник скидається і підрахунок вмикань механізму поновлюється. Схема має вимикач напруги і дві пускові кнопки, які забезпечують початковий пуск механізму праворуч або ліворуч.

Створення проекту


Створення проекту здійснюється після запуску в роботу Unity PRO XL і входу в меню File->New.... У вікні New Project, що відкриється вибрати контролер Premium і тип центрального процесора TSX P57 204M. Після підтвердження зроблених уставок на екрані EOM з'явиться вікно Project Browser з деревом проекту. Встановити курсор на теку Configuration і двічі клацнути лівою клавішею миші. Відкриється вікно X Bus із зображенням контролера і позначеними типами центрального процесора і автоматично вставленого модуля живлення PSY 2600M. Решта слотів вільна. Для їх заповнення модулями вводу-виводу потрібно по вибраному за номером слоту двічі клацнути лівою клавішею миші і в каталозі модулів Discrete, що з'явиться, позначити модуль вводу DEY 08D2D і модуль виводу SY08R4D. Після закриття каталогу вибраний модуль потрапить у відповідний слот мнемонічного контролера:



Коли контролер буде повністю скомпонований необхідно закрити вікно X Bus, щоб внести зроблені уставки у теку Configuration проектного дерева.




Створення програми

Для створення програмної секції подвійним клацанням лівої клавіші миші відкрити у дереві проекту теку Program, а потім теку MAST. Правою клавішею клацнути по теці Section і у New Section, що з'явиться, створити нову секцію, надати ім'я, вибрати ST-редактор і натиснути ОК. Відкриється сформоване дерево проекту і робоче поле для створення програмного коду.

ST- редактор надає користувачеві панель інструментів , за допомогою якої можна викликати відформатовані структури виразів:

- умовної дії - IF...;
- ітеративної умовної дії – WHILE...;
- дії, що повторюється – FOR...;
- вибіркової дії CASE....

Завдання користувача полягає у правильному заповненні структури виразів, що викликаються на робоче поле, виразами, операторами та операндами програми, що створюється.

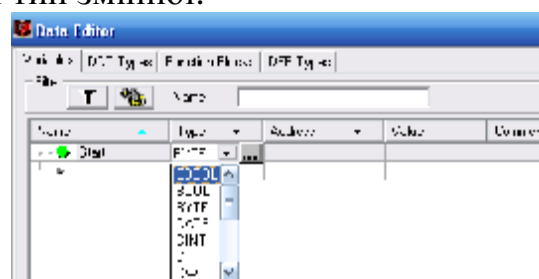
При потребі використання функціональних блоків необхідно звертатися до бібліотеки за допомогою асистента  (FFB Input Assistant), що знаходиться у панелі інструментів. Після натиснення лівою клав'яшею миші по кнопці  відкриється вікно Function Input Assistant, в якому у рядку FFB Type за допомогою кнопки  викликати бібліотеку блоків і операторів FFB Type Selection. Лівою клав'яшею миші відкрити теку Libset і двічі клацнути по вибраному блоку:



З'явиться вікно з характеристиками блока, типом і попереднім ім'ям його екземпляра (Instance).

Запропоноване ім'я екземпляра можна залишити, а можна і замінити на нове. Після натиснення OK, функціональний блок вважається оголошеним і потрапляє у вкладку Elementary FB Instance теки Variables&FB Instance проектного деревана, а на робочому полі залишається назва блока і дужки, в яких користувачем ST-мовою програмно конфігуруються входи функціонального блока. Далі відповідно до синтаксису ST-мови виконується конфігурування виходів блока.

Для декларування змінних проекту необхідно у проектному дереві відкрити вкладку Elementary Variables і у вікні Data Editor двічі клацнути лівою клав'яшею миші на вільному рядку колонки Name і вписати ім'я змінної. Після натиснення клав'яші Enter на клавіатурі EOM ім'я в рядку зафіксується і за замовчуванням запропонується тип змінної. Якщо він не відповідає змінній, то після подвійного клацання лівою клав'яшею миші по ньому відкриється список, з якого вибрати необхідний тип змінної:



Подвійне натиснення лівої клав'яші миші на рядку у колонці Address відкриє його для запису адреси зовнішніх вхідної або вихідної змінної. Адреса записується у вигляді %I0.2.2.0 для булевого входу або %Q0.3.1.0 – для булевого виходу. Тут I(Q) – вхід(вихід), 0 номер шасі, 2(3) – номер слота, 2.0(1.0) номер каналу вводу – виводу.

Програмний код, розроблений відповідно поставленої задачі має вигляд:







```

FBI_0 (S:=(on_le AND NOT mot_ri) OR time_ri,R1:=(NOT start OR time_le));
mot_le:=FBI_0.Q1;
COU_LE (CU:=mot_le,R:=RES,PV:=INT#20);
RES:=COU_LE.Q;
VL:=COU_LE.CV;
FBI_1 (IN:=FBI_0.Q1,PT:=t#3s);
time_le:=FBI_1.Q;
et_le:=FBI_1.ET;
FBI_2 (S:=(on_ri AND NOT mot_le) OR time_le,R1:=(NOT start OR time_ri));
mot_ri:=FBI_2.Q1;
COU_RI (CU:=mot_ri,R:=RES,PV:=INT#20);
RES:=COU_RI.Q;
VR:=COU_RI.CV;
FBI_3 (IN:=FBI_2.Q1,PT:=t#2s);
time_ri:=FBI_3.Q;
et_ri:=FBI_3.ET;
IF NOT start
  THEN mot_le:=FALSE; mot_ri:=mot_le;
  END_IF;
WHILE RES=TRUE
  DO COU_RI(CU:=FALSE, R:=TRUE, PV:=INT#20);
  STRL:=' '; STRR:=' ';
  RES:=FALSE;
END_WHILE;
CASE VL OF
5: STRL:='Left >5';
10: STRL:='Left >10';
15: STRL:='Left >15';
20: RES:=TRUE;
END_CASE;
CASE VR OF
5: STRR:='Right >5';
10: STRR:='Right >10';
15: STRR:='Right >15';
20: RES:=TRUE;
END_CASE;

```


Функціональні блоки і змінні, що оголошені у програмі, наведені у табл.13.1 і 13.2.




Таблиця 13.1 Функціональні блоки, що оголошені у програмі





Name ▲	no.	Type ▼	Value	Comment
 COU_LE		CTU		Лічильник рухів механізму ліворуч
 COU_RI		CTU		Лічильник рухів механізму праворуч
 FBI_0		RS		Керує рухом механізму ліворуч
 FBI_1		TON		Таймер руху механізму праворуч
 FBI_2		RS		Керує рухом механізму праворуч
 FBI_3		TON		Таймер руху механізму ліворуч


Таблиця 13.2. Вхідні і вихідні змінні, що оголошені у програмі

Name	Type	Address
et_le	TIME	
et_ri	TIME	
mot_le	EBOOL	%Q0.3.1.0
mot_ri	EBOOL	%Q0.3.3.0
on_le	EBOOL	%I0.2.2.0
on_ri	EBOOL	%I0.2.3.0
RES	BOOL	
Start	EBOOL	%I0.2.1.0
STRL	STRING	
STRR	STRING	
time_le	EBOOL	%Q0.3.2.0
time_ri	EBOOL	%Q0.3.4.0
VL	INT	
VR	INT	
Comment		
Відображає поточний час руху механізму ліворуч		
Відображає поточний час руху механізму праворуч		
Забезпечує рух механізму ліворуч		
Забезпечує рух механізму праворуч		
Вмикає рух механізму ліворуч		
Вмикає рух механізму праворуч		
Скидає значення лічильників		
Вмикає систему керування рухом механізму		
Показує рядкове повідомлення про кількість рухів механізму ліворуч		
Показує рядкове повідомлення про кількість рухів механізму праворуч		
Вмикає час руху механізму ліворуч		
Вмикає час руху механізму праворуч		
Відображає поточну кількість рухів механізму ліворуч		
Відображає поточну кількість рухів механізму праворуч		

Після створення програмного коду необхідно проаналізувати його на помилки за допомогою кнопки  (Analyze project) в панелі інструментів. У разі наявності помилок у нижній частині екрану з'являться відповідні повідомлення. Помилки необхідно виявити і виправити.

Для перевірки роботи програми увімкнути режим симуляції кнопкою  (Simulation mode) в панелі інструментів, кнопкою  (Connect) з'єднати проект з симулятором, а кнопкою  (Download Project) завантажити його. Фон екрану і текст програми отримують кольоровий вигляд.

У головному меню Service->Initialize Animation Table відкрити анімаційну таблицю. Для її заповнення двічі клацнути лівою клавішею миші у першому рядку колонки Name і натиснути кнопку , що з'явиться. З вікна Animation Table послідовно вибрати змінні для анімації і подвійним клацанням лівої клавіші миші перенести їх в анімаційну таблицю, де до них додаються нульові вихідні значення. В панелі інструментів анімаційної таблиці натиснути кнопку Force, установити курсор на змінну Start і клацнути лівою клавішею миші - активізуються кнопки  (Forces Value to 0) і  (Forces Value to 1). Натиснути кнопку  (Forces Value to 0) і у рядку Value замість нуля з'явиться F1. Таким самим чином активізувати змінну on_le. В панелі інструментів

натиснути кнопку  (Run) і програма почне працювати відповідно до алгоритму, а в колонці Value анімаційної таблиці будуть відображатися поточні зміни параметрів:






```

FBI_0 (S:=(on_le AND NOT mot_ri) OR time_ri, R1:=(NOT start OR time_le));
mot_le:=FBI_0.Q1;
COU_LE (CU:=mot_le, R:=RES, PV:=INT#20);
RES:=COU_LE.Q;
VL:=COU_LE.CV;
FBI_1 (IN:=FBI_0.Q1, PT:=t#3s);
time_le:=FBI_1.Q;
et_le:=FBI_1.ET;
FBI_2 (S:=(on_ri AND NOT mot_le) OR time_le, R1:=(NOT start OR time_ri));
mot_ri:=FBI_2.Q1;
COU_RI (CU:=mot_ri, R:=RES, PV:=INT#20);
RES:=COU_RI.Q;
VR:=COU_RI.CV;
FBI_3 (IN:=FBI_2.Q1, PT:=t#2s);
time_ri:=FBI_3.Q;
et_ri:=FBI_3.ET;
IF NOT start
THEN mot_le:=FALSE; mot_ri:=mot_le;
END_IF;
WHILE RES=TRUE
DO COU_LE (CU:=FALSE, R:=TRUE, PV:=INT#20);
STRL:=' '; STRR:=' ';
RES:=FALSE;
END_WHILE;
CASE VL OF
5: STRL:='Left >5';
10: STRL:='Left >10';
15: STRL:='Left >15';
20: RES:=TRUE;
END_CASE;
CASE VR OF
5: STRR:='Right >5';
10: STRR:='Right >10';
15: STRR:='Right >15';
20: RES:=TRUE;
END_CASE;

```

Name	Value	Type
et_le	2s_550ms	TIME
et_ri	0s	TIME
mot_le	1	EB00L
mot_ri	0	EB00L
on_le	F1	EB00L
on_ri	0	EB00L
Start	F1	EB00L
STRL	'Left >15'	STRING
STRR	'Right >10'	STRING
VL	15	INT
VR	14	INT

Для зупинення програми необхідно скористатися кнопкою  (Stop).

Після перевірки програми на якість роботи її можна завантажити у контролер. Для цього кнопкою  (Animation) вийти з режиму анімації, а кнопкою  (Disconnect) раз'єднатися з симулятором. Кнопкою  (Standard mode) перейти у режим звичайної роботи, кнопкою  (Connect) з'єднати проект з ПЛК і здійснити завантаження його натиснувши кнопку  (Download Project).

Завдання для самостійної роботи

Розробити FBD-мовою проект для контролера Modicon TSX Quantum, який керує механізмом за алгоритмом приведеним у даній роботі.

Зміст звіту

Звіт має містити проект, що створений відповідно до завдання на самостійну роботу.

Контрольні запитання

1. Яка послідовність дій при розробці нового проекту?
2. Як здійснюється конфігурування контролера?
3. Як створюється програмний код?
4. Як здійснюється відладка програми?
5. Як відбувається завантаження проекту у контролер?

Лабораторна робота №14

Створення проектів у системі програмування CX-One

Мета: ознайомитися з методикою створення проектів для контролерів корпорації Omron у системі програмування CX-One.

Розробити для контролера CS1 LD-програму керування температурою рідини у технологічному апараті, алгоритм якої надається релейно-контактною схемою, рис.14.1.

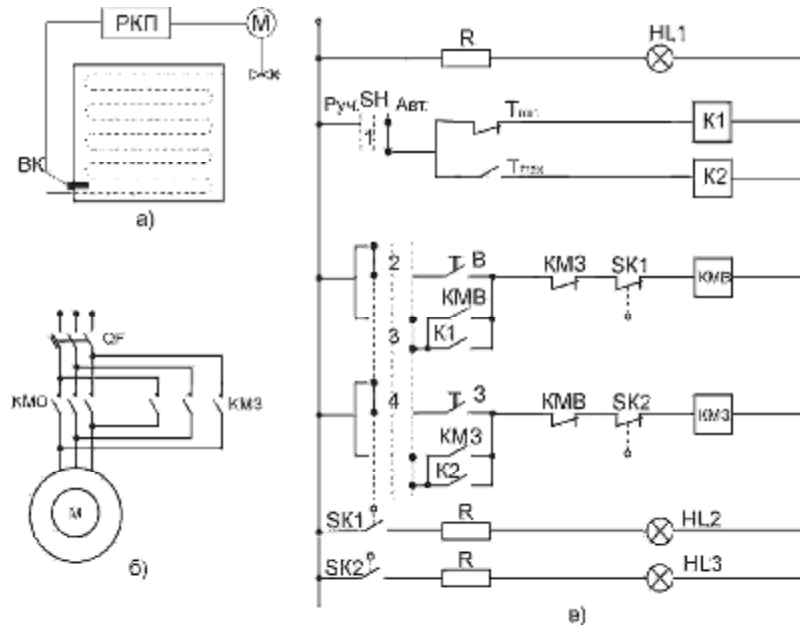


Рисунок 14.1– Схема автоматичного регулювання температури
а) - функціональна; б) – силових кіл виконавчого механізму М;
в) – релейно-контактна схема керування двигуном

Нагрівання відбувається парою. Датчиком температури ВК є електроконтактний манометричний термометр, виконавчим механізмом - заслінка з електроприводом. Керування приводом здійснюється релейно-контактним пристроєм РКП. Контакти T_{\min} і T_{\max} манометричного термометру малопотужні, тому в схемі передбачені проміжні реле K1 і K2, що використовуються як підсилювальні елементи. Універсальний перемикач SH перемикає кола керування з ручного режиму на автоматичний. Кінцеві вимикачі SK1 і SK2 вмикають контактори KM3 і KMB при повному закритті або відкритті заслінки і вмикають світлову сигналізацію HL2, HL3. У схемі передбачено електричне блокування, яке виключає одночасне вмикання контакторів KMB і KM3 (послідовно з KMB увімкнений робочий контакт KM3, а послідовно з KM3 - робочий контакт KMB).

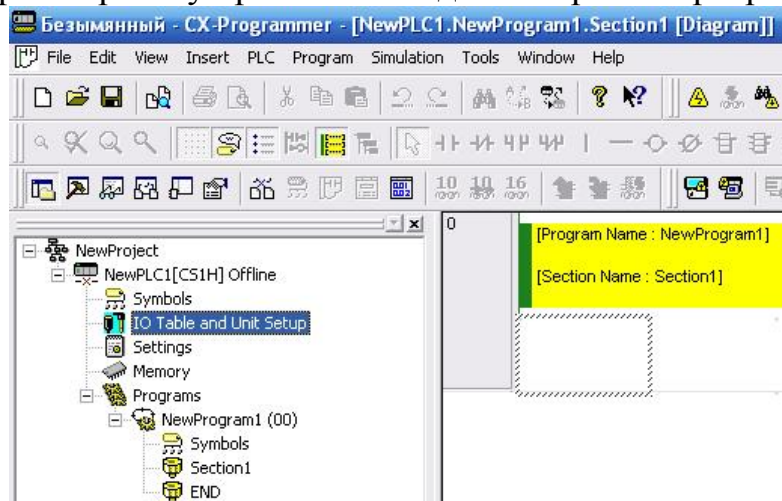
В автоматичному режимі перемикач SH знаходиться у положенні "Авт." Якщо температура рідини в апараті дорівнює або менше за нижню межу, то контакт термометра T_{\min} замкнений і на пульту оператора горить сигнальна лампа нижнього рівня температури. При вмиканні схеми спрацьовує реле K1, яке через контакти 3 перемикача SH, контакт KM3 і кінцевий вимикач SK1 подає сигнал на вмикання KMB. Вал виконавчого механізму М при

спрацьовуванні КМВ починає обертатися у бік відкриття заслінки, забезпечуючи приплив теплоносія в апарат. При повному відкритті заслінки кінцевий вимикач SK1 вмикає виконавчий механізм М і вмикає світлову сигналізацію HL2. Час руху вала від одного кінцевого вимикача до іншого складає 10с. При нагріванні рідини вище мінімальної температури розмикається контакт T_{min} і гасне сигнальна лампа мінімальної температури на пульту оператора, а при досягненні температурою верхньої встановленої межі замикається контакт T_{max} , який подає керувальний сигнал на закриття заслінки і на пульту оператора вмикається сигнальна лампа верхнього рівня температури.

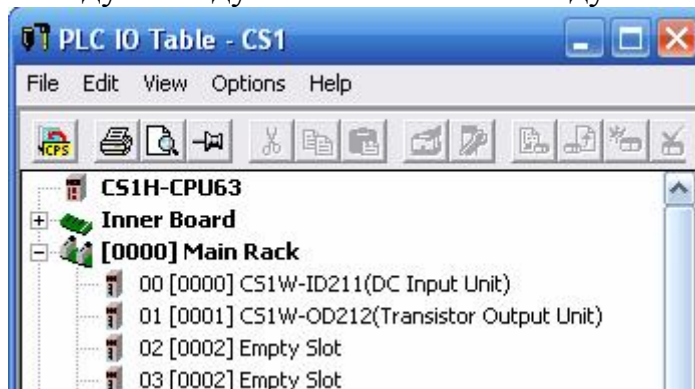
Створення проекту

У пакеті Cx-One запустити систему програмування Cx-Programmer. Для створення проекту вибрати меню File->New і у вікні Change PLC, що з'явиться, вказати перший номер контролера NewPLC1(у проекті може бути декілька ПЛК), його тип CS1H, тип мережі, за якою відбувається сполучення між програмою і контролером. За допомогою кнопок Settings... вибрати тип ЦПУ, відповідно до задачі керування (CPU63), параметри мережі і підтвердити зроблені уставки кнопкою OK.

З'явиться дерево проекту і робоче поле для створення програмного коду:











Для конфігурування контролера двічі натиснути ліву клавішу миші на вкладці дерева проекту IO Table and Unit Setup. У вікні PLC IO Table, що з'явиться, відкрити вкладку панелі центрального процесора Main Rack. Двічі натискаючи ліву клавішу миші, спочатку на адресі першого слота 00 [0000] Empty Slot, а потім другого 01 [0000] Empty Slot, зі списку Basic I/O вікна Select Unit вибрати дискретні модулі вводу CS1W-ID211 і виводу CS1W-OD212.



Для перевірки правильності конфігурування модулів вводу-виводу в меню «Options» вікна PLC IO Table вибрати «Check». При безпомилковому конфігуруванні з'явиться підтвердження, після якого можна закрити вікно PLC IO Table і почати розробку програмного коду.

Створення програми


Робоче поле пропонує користувачеві нульовий блок секції для створення першої мережі програмного коду, а LD-редактор – відповідні інструменти, серед яких:

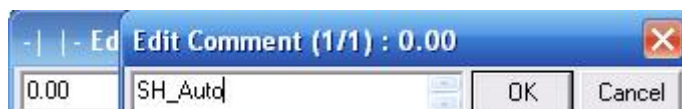
-  (Selection Mode)- курсор редактора;
-  (New contact)- прямий контакт;
-  (New closed contact) –інвертований контакт;
-  (New vertical) – вертикальна з'єднувальна лінія;
-  (New horizontal) – горизонтальна з'єднувальна лінія;
-  (New coil) –пряма котушка;
-  (New closed coil) –інвертована котушка;
-  (New PLC Instruction)- інструкція ПЛК.

Відповідно до алгоритму керування температурою рідини, програмний код проекту повинен мати дві основні мережі, серед яких перша забезпечувала б керування виконавчим механізмом при мінімальній температурі рідини, а друга - при максимальній температурі. Тому в роботі програми, як при ручному, так і при автоматичному управлінні, визначальними мають бути сигнали з електроконтактного датчика температури T_{min} і T_{max} . Тільки від їх стану (замкнений чи розімкнений) залежить напрям руху вала виконавчого механізму – на збільшення чи зменшення подачі теплоносія у технологічний апарат.

Щоб реалізувати заданий алгоритм керування кожна з двох LD-мереж має містити змінні, які реагують на режим керування (ручний чи автоматичний), на стани датчика температури, кінцевого вимикача, кнопки ручного керування, а також змінну, що забезпечує вмикання в роботу виконавчого механізму.

З урахуванням цього, реалізація програмного коду має відбуватися з використанням прямих та інвертованих контактів, прямих котушок, а також таймерів, визначаючих час роботи виконавчого механізму.

При створенні фрагмента програми, який керуватиме відкриттям заслінки, треба встановити прямий контакт у нульовому блоці секції, змінна якого буде реагувати на стан універсального перемикача режимів роботи. Для цього навести курсор на кнопку  (New contact), клацнути лівою клавішею миші, і повторно натиснути її, коли курсором буде вибрана потрібна точка на робочому полі секції. У вікні, що з'явиться, записати адресу вхідної змінної, наприклад 0.00, і натиснути ОК, а у вікні Edit Comment – вказати її ім'я SH_Auto:




Після закриття вікон прямий контакт з назвою і адресою залишиться у нульовому блоці секції.

При адресації змінних необхідно керуватися тим, що:



- слот 0 (завжди зліва) містить 32-х канальний дискретний модуль вводу, для якого резервуються слова 00-01;
- слот 1 містить 32-х канальний дискретний модуль виводу, для якого резервуються слова 02-03;
- слот 2 містить аналоговий модуль вводу-виводу, для якого резервуються слова 2000-2039;
- слот 3 містить аналоговий модуль вводу-виводу, для якого резервуються слова 2040-2049.

Адреса змінної складається з номера слова і номера біта, який відповідає каналу модуля. Ці номери, обов'язково розділяються крапкою.


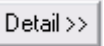
Послідовно з прямим контактом, аналогічно, установити три інверсних контакти  (New closed contact) з іменами і адресами змінних:

- мінімальної температури - T_{min} (0.01);
- керувального сигналу на закриття заслінки - $Dvig_Close$ (0.02);
- кінцевого вимикача - $Konc1$ (0.03).

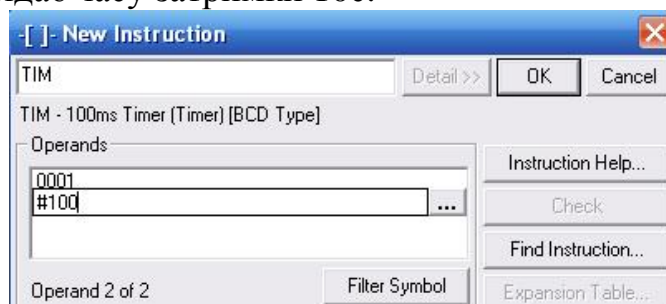
Оскільки при підвищенні температури відносно мінімальної уставки контакт датчика температури T_{min} розмикається, необхідно передбачити в програмі прямий контакт, який би протягом істинного стану вихідної змінної, що керує роботою виконавчого механізму у бік відкриття заслінки $Dvig_Open$, шунтував інвертований контакт T_{min} у нульовому блоці секції і не давав виконавчому механізму, при автоматичному керуванні, передчасно зупинитися.

У програмному коді шунтування інвертованого контакту T_{min} прямим контактом $Dvig_Open$ реалізується їх паралельним розташуванням, яке здійснюється за допомогою кнопок  (New vertical) і  (New contact) за вже описаною методикою.


При ручному режимі керування виконавчим механізмом здійснюється за станом сигнальних ламп на пульту оператора. Коли горить лампа нижнього рівня температури, збільшення подачі теплоносія забезпечується кнопкою «більше», а коли горить лампа верхнього рівня подача теплоносія зменшується кнопкою «менше». Тому в програмі мають бути гілки зі змінними $Open$ і $Close$, які забезпечують ручне керування, в обхід контактам T_{min} і T_{max} , що беруть участь при автоматичному керуванні.

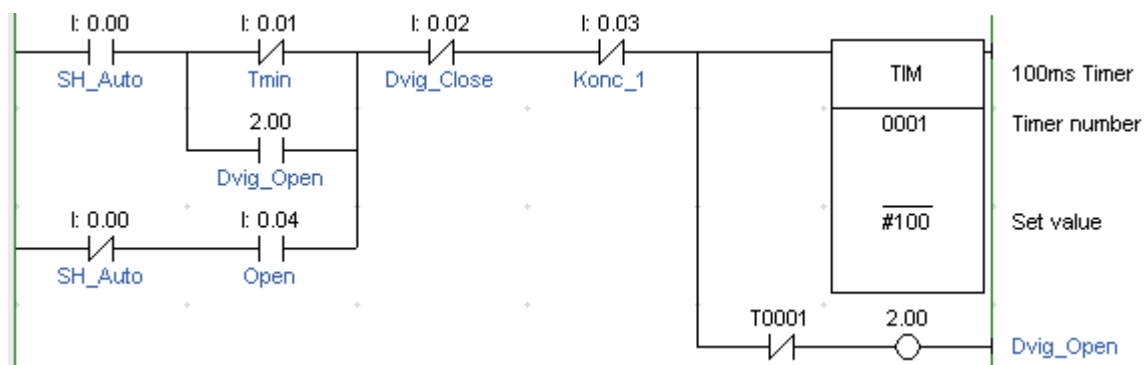
Останнім елементом LD-мережі має бути екземпляр таймера, який забезпечить вимикання виконавчого механізму за 10с роботи. Щоб його установити необхідно навести курсор на кнопку  (New PLC Instruction), клацнути лівою клавішею миші і повторно натиснути її у місці встановлення блока. У вікні New Instruction натиснути кнопку , а потім кнопку Find Instruction. Зі списку Groups вибрати бібліотеку Timer/Counter і позначити інструкцію TIM:

Після вибору інструкції закрити вікно Find Instruction, а у вікні New Instruction, де з'явиться назва інструкції TIM, записати її операнди – номер таймера 0001, і константу часу затримки #100. При виборі константи, треба враховувати, що відлік часу таймером здійснюється з інтервалом 0,1с. Тобто константа #100 відповідає часу затримки 10с.





Закрити вікно New Instruction і у сегменті програми з'явиться таймер, який необхідно пересунути до правої шини живлення за допомогою курсору і лівої клавіші миші. Булевий вихід таймера оформлюється у програмі інвертованим контактом з номером таймера і встановлюється перед прямою котушкою, змінна якої Dvig_Open керує вмиканням виконавчого механізму у бік відкриття заслінки.

Для встановлення прямої котушки навести курсор на кнопку  (New coil), клацнути лівою клавішею миші і повторно натиснути її у місці розташування. У вікні, що з'явиться, записати адресу, а у вікні Edit Comment - назву змінної Dvig_Open, яка буде керувати виконавчим механізмом на відкриття заслінки, і закрити діалогові вікна. Програмний код LD-мережі, що забезпечує керування температурою у бік нагрівання має вигляд:

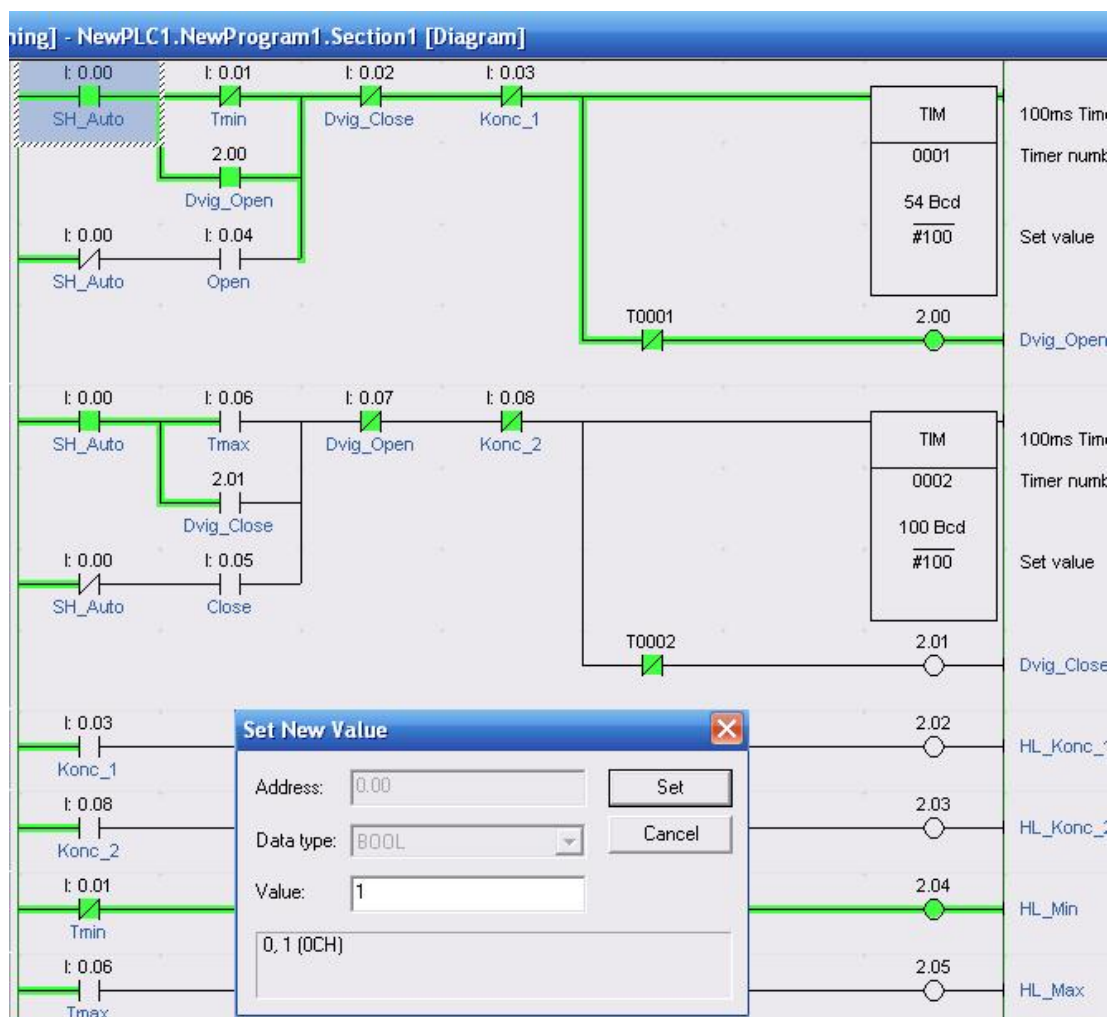


Аналогічно в окремому блоці секції створити LD-мережу на керування температурою у бік її зменшення, а також передбачити блоки секції з LD-мережами, що забезпечують сигналізацію при спрацьовуванні кінцевих вимикачів виконавчого механізму і при замиканні контактів T_{min} і T_{max}.

Розроблена програма компілюється за допомогою кнопки  (Compile PLC programs) у панелі інструментів. При наявності помилок у нижній частині екрану з'являються відповідні повідомлення. Після коригування програми і відсутності будь-яких помилок здійснюється її налагодження у режимі симуляції. Для завантаження програми ПЛК у симулятор натиснути кнопку  (Work Online Simulator) у панелі інструментів. Коли з'явиться повідомлення про

успішне завантаження, фоновий колір екрану зміниться. Після закриття вікна Download кнопкою ОК, всі елементи програми, які мають статус TRUE (Істина) підсвічуються зеленим кольором.

Для перевірки роботи програми необхідно активізувати вхідні змінні відповідно заданому алгоритму роботи і відстежувати стани вихідних. Для цього двічі клацнути лівою клавішею миші на вибраному контакті, наприклад SH_Auto, і у вікні Set New Value, що з'явиться, встановити 0 чи 1 і натиснути кнопку Set. На екрані відбуваються зміни відповідно до роботи схеми керування.



В програмі використовуються такі змінні:

- **вхідні:**

SH_Auto: BOOL – реагує на стан перемикача режимів роботи Авт/Руч;

Tmin: BOOL – реагує на електричний контакт манометричного термометра, що спрацьовує при мінімальній температурі рідини;

Tmax: BOOL – реагує на електричний контакту манометричного термометра, що спрацьовує при максимальній температурі рідини;

Open: BOOL –реагує на кнопку ручного керування виконавчим механізмом у бік відкриття регульовального органа;

Close: BOOL – реагує на кнопку ручного керування виконавчим механізмом у бік закриття регулювального органа;

Konc_1: BOOL – реагує на стан кінцевого вимикача виконавчого механізму при відкритому регулювальному органі;

Konc_2: BOOL – реагує на стан кінцевого вимикача виконавчого механізму при закритому регулювальному органі;

• вихідні:

Dvig_Open: BOOL – вмикає виконавчий механізм на відкриття регулювального органа;

Dvig_Close: BOOL – вмикає виконавчий механізм на закриття регулювального органа;

HL_Konc_1: BOOL – вмикає сигнальну лампу положення першого кінцевого вимикача;

HL_Konc_2: BOOL – вмикає сигнальну лампу положення другого кінцевого вимикача;

HL_Min: BOOL – вмикає сигнальну лампу мінімального рівня температури;

HL_Max: BOOL – вмикає сигнальну лампу максимального рівня температури;

• внутрішні:

T0001: BOOL – булевий вихід таймера 0001;

T0002: BOOL – булевий вихід таймера 0002.

Для завантаження програми у контролер необхідно вийти з режиму симуляції і здійснити з'єднання з ПЛК, використовуючи в меню проекту шлях PLC->Work Online. При успішному з'єднанні фоновий колір екрану зміниться. Перевести контролер у стан Stop PLC і здійснити завантаження програми з меню PLC->Transfer to PLC.

Завдання для самостійної роботи

Розробити проект LD-мовою для контролера CS1, який керує роботою світлофора.

Зміст звіту

Звіт має містити проект, що створений відповідно до завдання на самостійну роботу.

Контрольні запитання

1. Яка послідовність дій при розробці нового проекту?
2. Як здійснюється конфігурування контролера?
3. Як створюється програмний код?
4. Як здійснюється відладка програми?
5. Як відбувається завантаження проекту у контролер?

Лабораторна робота №15

Створення проектів у системі програмування RSLogix 5000

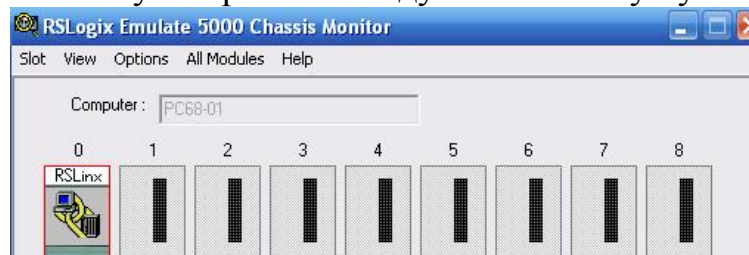
Мета: ознайомитися з методикою створення проектів для контролерів корпорації Rockwell Automation у системі програмування RSLogix 5000.

Розробити мовою релейної логіки проект керування роботою апарата, в якому відбувається нагрів і змішування компонентів. Якщо натиснута кнопка "Пуск" і не натиснута кнопка "Стоп", то відкривається клапан і суміш наповнює апарат. Після досягнення верхнього рівня клапан суміші закривається. Одночасно відкривається клапан подачі пари і починається стадія нагріву. Після досягнення сумішню заданої температури - клапан подачі пари закривається і на 2 хвилини вмикається мішалка. Після закінчення часу перемішування, мішалка вимикається і продукт зливається. Коли спрацьовує датчик нижнього рівня, технологічний цикл завершується.

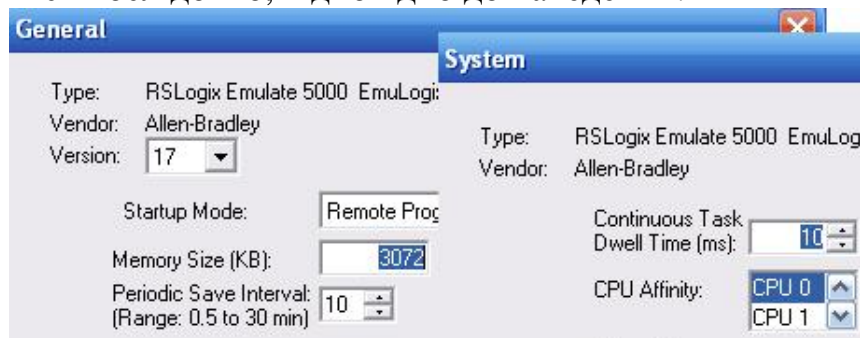
Конфігурування емулятора шасі

Для перевірки роботи створеної програми необхідно конфігурувати емулятор Монітор Шасі (Chassis Monitor). Щоб запустити відповідну програму, користуючись лівою клавішею миші, пройти шлях Пуск->Все программы->Rockwell Software-> RSLogix Emulate5000-> RSLogix Emulate Chassis Monitor

На екрані з'явиться емулятор шасі з модулем RSLinx у нульовому слоті:

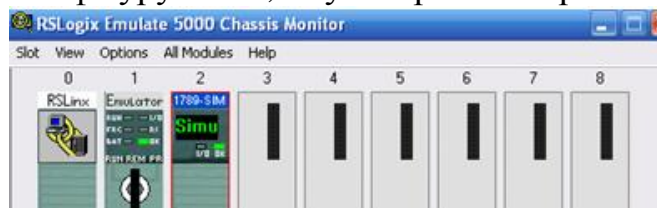


Щоб додати у слот №1 модуль центрального процесора, з меню Slot емулятора шасі вибрати Create Module (Создать модуль), а у вікні Selekt Module, що з'явиться вибрати тип модуля Emulator RSLogix Emulate 5000 Controller. Якщо після цього з'явиться повідомлення про попередні конфігурації, помітити альтернативу Reset the Configuration to Default Values і натиснути «Далее». Встановити параметри емулятора центрального процесора у вікнах General і System, що з'являться послідовно, відповідно до наведених:

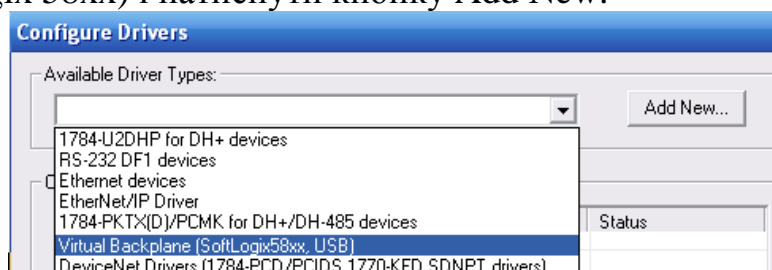


Щоб додати у слот №2 емулятор модуля вводу-виводу, треба пройти той самий шлях у меню Slot ->Create Module (Создать модуль), тільки вибрати у вікні Selekt Module модуль 1789-SIM 32 Point Input/Output Simulator, а у вікні General надрукувати для нього мітку IO_emulate.

Після завершення конфігурування, емулятор Монитор Шасі отримує вигляд:



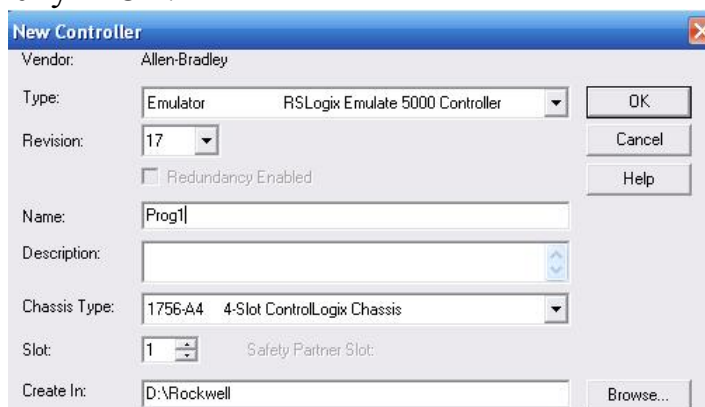
Для створення драйвера зв'язку емулятора Монитор Шасі з іншими прикладними програмами запустити програму RSLogix, користуючись меню Пуск> Все програми> Rockwell Software > RSLogix > RSLogix Classic, або натиснувши праву клавішу миші на модулі RSLinx в емуляторі RSLogix Emulate Chassis Monitor, викликати кнопку Open RSLinx і лівою клавішею миші відкрити вікно RSLogix Classic Gateway. З меню Communications вибрати Configure Drivers. З'явиться вікно Configure Drivers. Якщо у вікні вже показане ім'я драйвера, треба позначити його лівою клавішею миші і видалити кнопкою Delet. Після цього з меню Available Driver Types (Тип драйвера) вибрати Virtual Backplane (SoftLogix 58xx) і натиснути кнопку Add New.



У вікні Add New RSLogix Classic Driver, що з'явиться призначити ім'я драйверу, наприклад Connect, і натиснути OK. Після цього назва драйвера з'явиться у списку Configured Drivers. Закрити вікно натиснувши Close.

Створення проекту

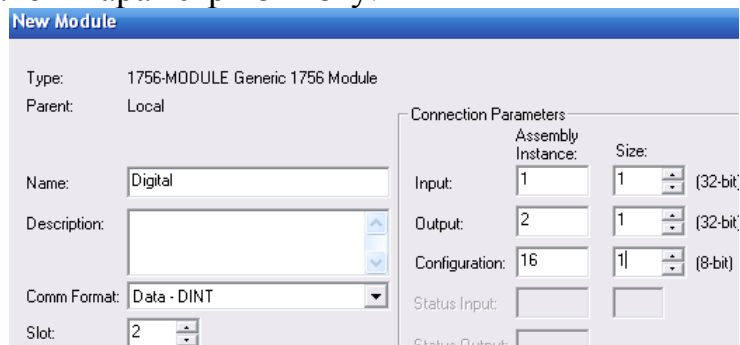
Запустити програму RSLogix (Меню Пуск > Все програми > Rockwell Software > RSLogix 5000 Enterprise Series > RSLogix 5000). З'явиться вікно Start Page зі вставкою Quick Start. Позначити альтернативу New Project. У вікні New Controller, що з'явиться, вибрати тип (Type) контролера - RSLogix Emulate 5000 Controller, версію мікропрограмного забезпечення (Revision) –17, призначити назву проекту, наприклад Prog1, тип шасі Chassis Type – 1756-A4 4-Slot ControlLogix Chassis, слот центрального процесора (Slot) –1 і директорію знаходження файлу проекту (можна залишити директорію, що пропонується за замовчуванням), натиснути OK.



Відкриється вікно проекту з організатором програми:



Для конфігурування контролера необхідним модулем дискретного вводу-виводу правою клавішею миші натиснути на теці I/O Configuration і відкрити меню New Module. У діалоговому вікні Select Module з каталогу Module вибрати групу Other і двічі натиснути ліву клавішу миші. З'явиться тип модуля 1756-Module. Позначити його лівою клавішею миші і натиснути OK. З'явиться вікно New Module, в якому ввести ім'я модуля, наприклад Digital, номер слота -2, де він знаходиться, а також параметри зв'язку:



Після підтвердження зроблених уставок кнопкою OK, з'явиться вікно Module Properties, в якому треба встановити інтервал відліку часу при виконанні програми Requested Packet Interval - 50ms і натиснути OK.

Створення програмного коду

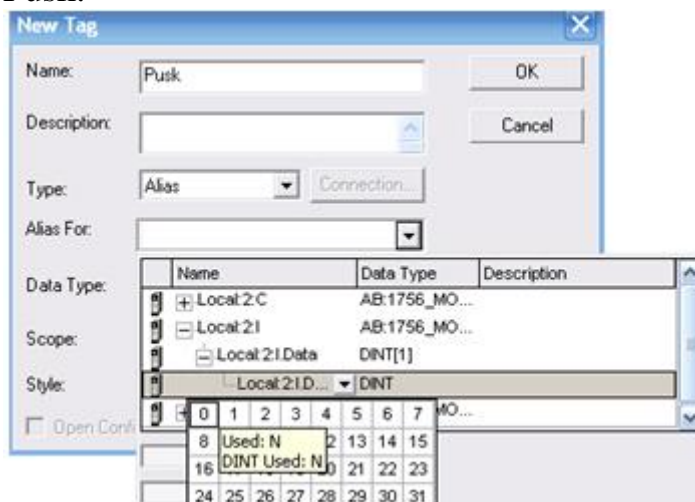
Для створення програмного коду в проектному дереві відкрити теки Task> MainTask> MainProgram, і двічі клацнути лівою клавішею миші по вкладці MainRoutine. Відкриється робоче поле для створення коду програми з вихідною LD-мережею:



В панелі інструментів лівою клавішею миші вибрати альтернативу Favorites, щоб активізувати кнопки елементів графічної мови.

Відповідно до алгоритму управління процесом виготовлення суміші створити LD-мережу, яка керує клапаном подачі суміші. Для цього у панелі інструментів послідовно вибрати курсором кнопку прямого контакту, потім L-катушку і натиснути на кожному з цих елементів ліву клавішу миші. Вибрані елементи з'являться у першій LD-мережі. Щоб оголосити змінні, навести курсор на знак питання, що знаходиться над кожним елементом мережі, і правою клавішею меню відкрити меню. Вибрати альтернативу New Tag...і у вікні, що з'явиться, призначити ім'я нового тегу і вибрати його тип – Alias (Псевдонім).

Щоб призначити адресу псевдоніму змінній, треба в меню Alias For: лівою клавішею миші відкрити вкладку адресів входу Local:2:I і вибрати нульовий біт змінної Pusk:



Після цього у рядку Alias For: з'явиться адреса змінної Pusk, формат якої має вигляд:

Local:2:I.Data[0].0 – (Location:Slot:Type.Member.SubMember.Bit),

де:

Location –місце розташування в мережі;

Slot- номер слота модуля вводу-виводу;

Type – Тип даних (I – ввід, O – вивід, C – конфігурація, S - статус);

Member – Конкретні дані, що надходять від модуля вводу-виводу;

SubMember – Конкретні дані, що відносяться до Member;

Bit – Конкретний канал в модулі вводу-виводу.

Тип змінної BOOL визначається автоматично, а контекст (Scope) і стиль (Style) вибрати, відповідно, як Prog1(Назва програми) і Binary, натиснути кнопку OK. Ім'я та адреса змінної з'являться над відповідним елементом програми.

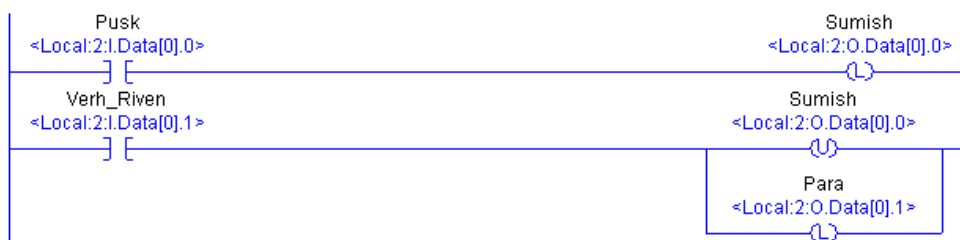
Для створення другої LD-мережі, яка забезпечить керування клапанами, що подають суміш і пару при спрацьовуванні датчика верхнього рівня, клацнути лівою клавішею миші спочатку по кінцевій мережі (End) в робочому полі, а потім по кнопці Rung в панелі інструментів проекту.

За аналогією з першою мережею встановити прямий контакт і паралельні гілки з U-катушкою і L-катушкою та оголосити змінні з назвами, відповідно, Verh_Riven, Sumish і Para.

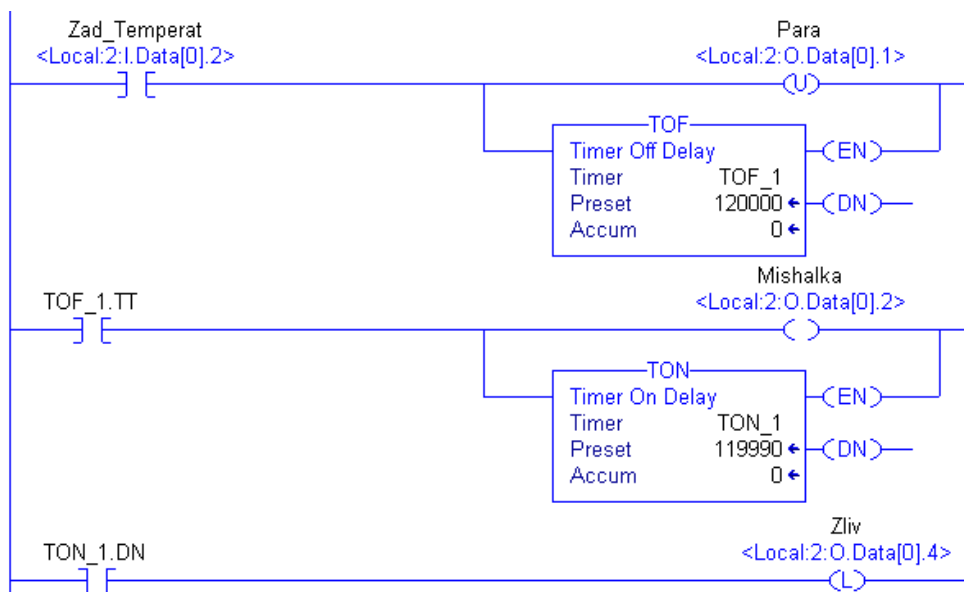
Щоб побудувати паралельну гілку необхідно позначити лівою клавiшею миші відповідний елемент програми і натиснути лівою клавiшею миші кнопку Branch в панелі інструментів. Якщо паралельна гілка не охоплює U-котушку, треба встановити курсор на неї, натиснути лiву клавiшу миші і передягнути кiнець гілки до правої шини живлення.

Для присвоєння котушки вже оголошеної назви Sumish, треба двічі клацнути лівою клавiшею миші по знаку питання, що над нею. Відкриється доступ до списку змінних, з якого лівою клавiшею миші вибрати потрібну назву і повторно клацнути на робочому полі. Назва змінної і її адреса з'являться над котушкою.

Програмний код отримує вигляд:



Третя і четверта мережі повинні забезпечити закриття клапана на паропроводі, і увімкнення в роботу на 2 хвилини мішалки, а п'ята – відкриття клапана на злив суміші. Фрагмент цієї частини програми отримає вигляд:



Створення третьої мережі з прямим контактом і U-котушкою, а також оголошення змінних Zad_Temperat і Para, здійснюється за попередньою методикою. Для встановлення таймера, треба позначити лівою клавiшею миші попередній елемент мережі (прямий контакт), в панелі інструментів активізувати каталог Timer/Counter і натиснути лівою клавiшею миші кнопку TOF. Коли таймер з'явиться у мережі, клацнути правою клавiшею миші по рядку Timer, відкриється вікно New Tag..., в якому треба оголосити цей функціональний блок, надрукувавши ім'я TOF_1, вибрати тип – Base, тип даних TIMER і назву проекту Prog1. Після натиснення кнопки ОК відкриється доступ

до рядків Preset і Accum, в які записати час затримки 120000 і початкове значення відліку 0.

Щоб забезпечити вплив таймера на котушки, що керують клапанами мішалки та зливу суміші, треба створити додаткову четверту LD-мережу з вихідним прямим контактом таймера TOF_1.TT і прямою котушкою.


Для оголошення вихідної змінної таймера, двічі клацнути лівою клавішею миші по знаку питання, що над прямим контактом у новій LD-мережі. Відкриється доступ до списку вже оголошених змінних і функціональних блоків. Зі списку виходів блока TOF_1 позначити лівою клавішею миші вихід TOF_1.TT, перевести курсор у будь-яке місце робочого поля і повторно клацнути лівою клавішею. Над прямим контактом з'явиться його назва.

Назву прямої котушки Mishalka присвоїти за аналогією оголошення попередніх змінних.

Щоб забезпечити відкриття клапана на злив суміші після зупинення роботи мішалки, треба користуючись каталогом Timer/Counter, що у панелі інструментів, встановити таймер TON паралельно прямій котушці Mishalka і за аналогією з таймером TOF оголосити його, як TON_1. Час затримки цього таймера встановити трохи меншим ніж у таймера TOF_1. Це забезпечить можливість своєчасного увімкнення виходу таймера TON_1.DN і активізації L-котушки Zliv.

Наступні дві мережі мають забезпечити завершення технологічного циклу і можливість вимкнення подачі суміші:



Для перевірки роботи програми у верхньому рядку інструментів натиснути кнопку  (Verify Routine). Якщо інформаційне вікно (відкривається і закривається кнопками на клавіатурі Alt+1), покаже помилки – виправити їх.

Для завантаження проекту в головному меню Communications вибрати Who Activ, позначити канал зв'язку з емулятором 01,RSLogix5000 Emulator, Prog1 і натиснути Download:

Після цього у лівому верхньому куту екрану активізується меню оперативного керування режимами роботи програми і увімкнеться режим Run, а шини живлення у програмі змінять колір на зелений.

Це свідчить про те, що інструментальна система для перевірки роботи програми готова. Якщо режим Run не увімкнувся, треба увійти в меню Rem Run, що у лівому верхньому куті, і вибрати альтернативу Run Mode.

Змінюючи стан контактів, відповідно до алгоритму роботи технологічного апарата, можна відслідковувати хід виконання програми по зеленому підсвічуванні відповідних елементів і зміні вмісту рядка Accum в блоках таймерів.

Якщо потрібно вимкнути режим емуляції Run , треба увійти в меню Rem Run, що у лівому верхньому куті, і вибрати альтернативу Program Mode. Скид програми здійснюється за допомогою альтернативи Go Offline того самого меню.

[illegible]

Розробити FBD-мовою проект для контролера ControlLogix, який керує технологічним апаратом, що працює за алгоритмом, приведеним у даній роботі.

Звіт має містити проект, що створений відповідно до завдання на самостійну роботу.

1. Яка послідовність дій при розробці нового проекту?
2. Як здійснюється конфігурування контролера?
3. Як створюється програмний код?
4. Як здійснюється відладка програми?
5. Як відбувається завантаження проекту у контролер?

Лабораторна робота №16

Створення проектів у системі програмування Proficy Machine Edition

Мета: ознайомитися з методикою створення проектів для контролерів корпорації GE Fanuc у системі програмування Proficy Machine Edition.

Розробити LD-мовою програму керування роботою насоса, що подає споживачам господарську воду, рис.16.1.

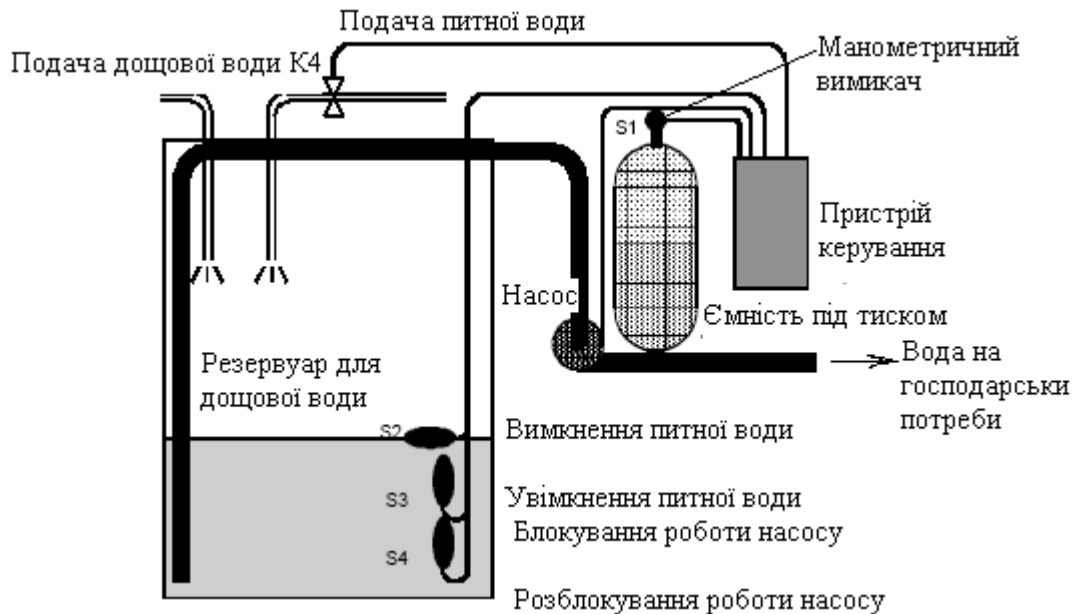


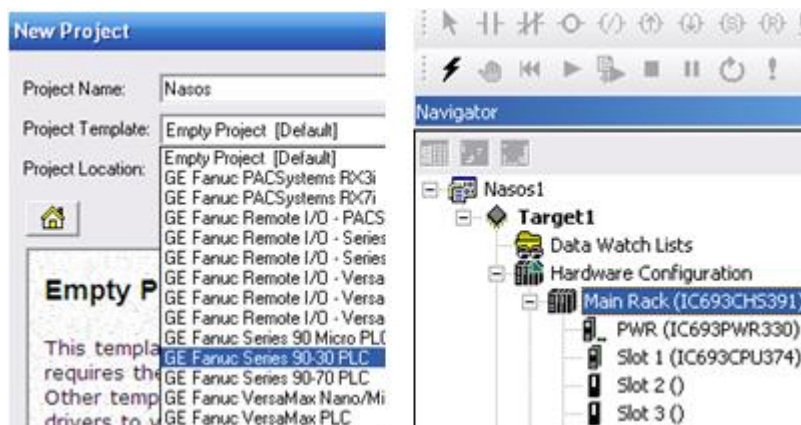
Рисунок 16.1 – Система постачання води на господарські потреби

Дощова вода збирається у резервуар і закачується насосом у спеціальну ємність, звідки вона відбирається споживачами. Щоб резервуар не спустошувався, в нього доливається питна вода. Насос і електромагнітний клапан питної води управляються за допомогою манометричного (S1-нормально відкритий контакт) і трьох поплавкових вимикачів (S2, S3- нормально закриті контакти, S4-нормально відкритий контакт), що знаходяться у резервуарі дощової води. Насос вмикається, коли тиск в ємності падає нижче за мінімальний рівень. При досягненні робочого тиску, насос вимикається із затримкою на 20с, що зменшує частоту його вмикань при тривалому відборі води. Насос господарської води не може вмикатися, коли в резервуарі недостатньо води.

Створення проекту

Для створення проекту запустити програму через меню Пуск->_Все программы->GE Fanuc-> Proficy Machine Edition -> Proficy Machine Edition. На екрані з'явиться вікно Machine Edition, в якому позначити альтернативу Machine Edition Template і натиснути кнопку ОК.

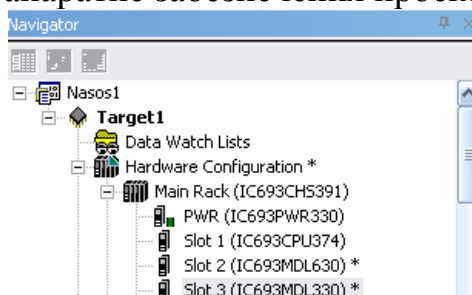
У вікні New Project, що відкриється, надрукувати ім'я проекту, наприклад Nasos1, вибрати шаблон проекту GE Fanuc Series 90-30 PLC, рис., і натиснути кнопку ОК. На екрані з'явиться Navigator проекту, в якому лівою клавішею миші відкрити вкладку Hardware Configuration, а потім головний крейт Main Rack:



За замовчуванням головний крейт пропонується з блоком живлення і центральним процесором.

Для конфігурування модуля вводу двічі клацнути лівою клав'яшею миші по вільному слоту 2, відкриється каталог модулів Modul Catalog, з якого вибрати дискретний модуль вводу на 24В вхідної напруги (IC693MDL630). Після підтвердження вибору кнопкою ОК, відкриється вікно з параметрами вибраного модуля, де показана адреса першого біту в області пам'яті %I00081, яка використовується при оголошенні змінних. Після закриття вікна вибраний тип модуля заповнює слот 2.

Для конфігурування модуля виводу у слоті 3 аналогічно з каталогу вибрати дискретний модуль на 240В змінного струму (IC693MDL330). Після конфігурування контролера апаратне забезпечення проекту має вигляд:

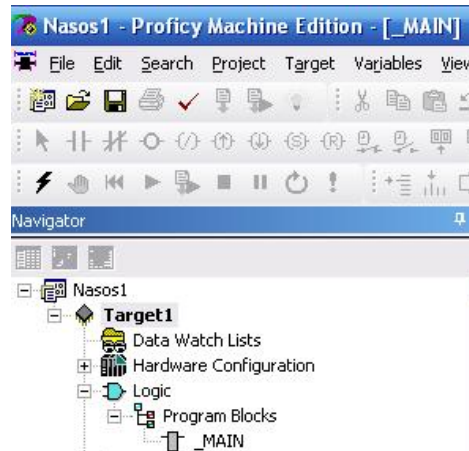


Для визначення властивостей зконфігурованої цілі (контролера або віддаленого вводу-виводу), натиснути правою клав'яшею миші на Target1(Ціль) і з меню, що випало, вибрати Properties. З'явиться Inspector, в якому властивості цілі, при необхідності, можна відкоригувати, якщо клацнути лівою клав'яшею миші по відповідному рядку.


Inspector	
Target	
Name	Target1
Type	GE Fanuc Controller
Description	
Documentation Address	
Family	Series 90-30 PLC
Controller Target Name	Nasos11
Update Rate (ms)	250
Sweep Time (ms)	Offline
Controller Status	Offline
Enable Shared Variable	False
Physical Port	COM1
Additional Configuration	

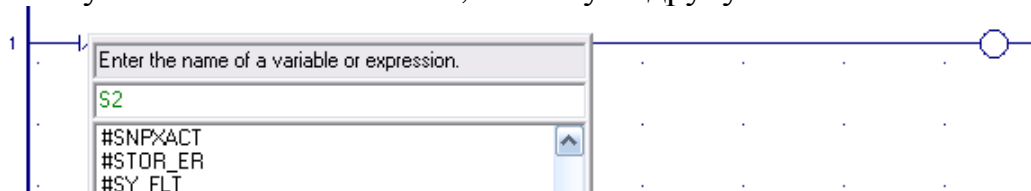
Створення програмного коду

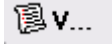
Для створення програмного коду у навігаторі проекту відкрити вкладку Logic і лівою клавішею миші увійти у робоче поле LD-редактора програмного блока Main.

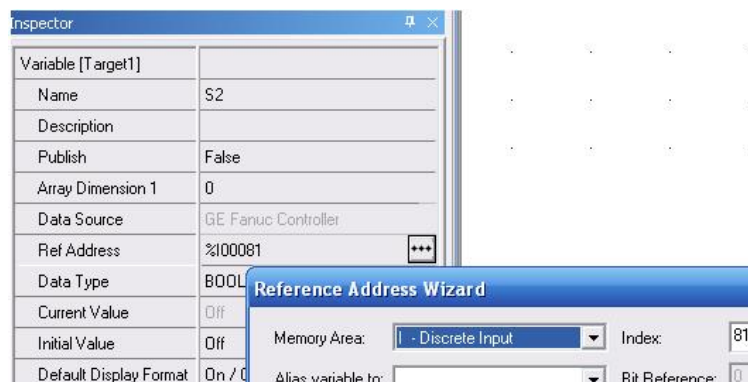


Щоб створити першу LD-мережу програмного коду, яка, відповідно до алгоритму задачі, забезпечить подавання питної води, треба на робочому полі приєднати до шин живлення послідовно два інвертованих контакти і пряму котушку. Для цього в панелі інструментів вибрати по черзі лівою клавішею миші відповідні елементи (⌘, ⚡) і повторним клацанням поставити їх на робоче поле.




Для оголошення змінних в панелі інструментів активізувати стандартний вказівець , навести його на перший інвертований контакт і двічі натиснути ліву клавішу миші. З'явиться вікно, в якому надрукувати ім'я змінної S2:



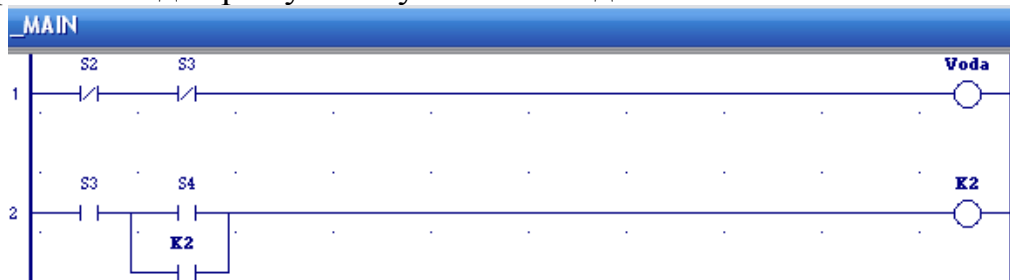
Натиснути клавішу Enter на клавіатурі, щоб ім'я змінної записалося над контактом і було занесене у список на вкладці  навігатора. Клацнути правою клавішею миші по контакту S2, з'явиться меню, в якому увійти у Properties. Лівою клавішею миші відкрити рядок Ref Address у інспекторі (Inspector), що з'явився, надрукувати адресу %I00081, присвоїти тип BOOL і натиснути кнопку з трьома крапками. У діалоговому вікні Reference Address Wizard, з визначеною змінною в області пам'яті, додати номер індексу 81 і закрити вікно.








Аналогічно оголосити змінні іншого інвертованого контакту, як S3 з адресою %I00082 і прямої котушки, як Voda з адресою %Q00001.

Для створення другої LD-мережі, яка забезпечить блокування роботи насосу при критично мінімальному рівні води у резервуарі, клацнути лівою клавішею миші на робочому полі нижче першої мережі. Панель інструментів активізується. Вибрати по черзі лівою клавішею миші елементи ,  і повторним клацанням поставити їх на робоче поле у відповідній послідовності. За допомогою кнопки  додати у мережі гілку з прямим контактом паралельно контакту S4. Нові змінні елементів програмного коду S4 (%I00083) і K2 (%Q00002) оголосити відповідно до попередній методики, а змінну S3 викликати зі списку, як вже оголошену.

Програмний код отримує наступний вигляд:

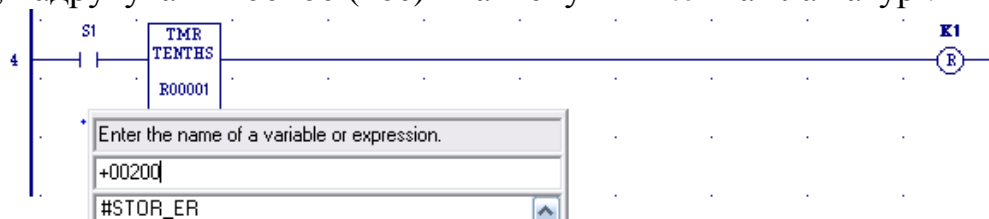


Для створення мережі, яка забезпечить вмикання насосу і затримку при його вимиканні, необхідно активізувати панель інструментів, клацнувши лівою клавішею миші на вільному місці робочого поля. Вибрати по черзі лівою клавішею миші елементи , , , ,  і повторним клацанням поставити їх на робоче поле у відповідній послідовності. Оголосити змінні Nasos (%Q00004), K1(%Q00003), S1(%I00084) нової LD-мережі за попередньою методикою, а K2 викликати, як вже оголошену.

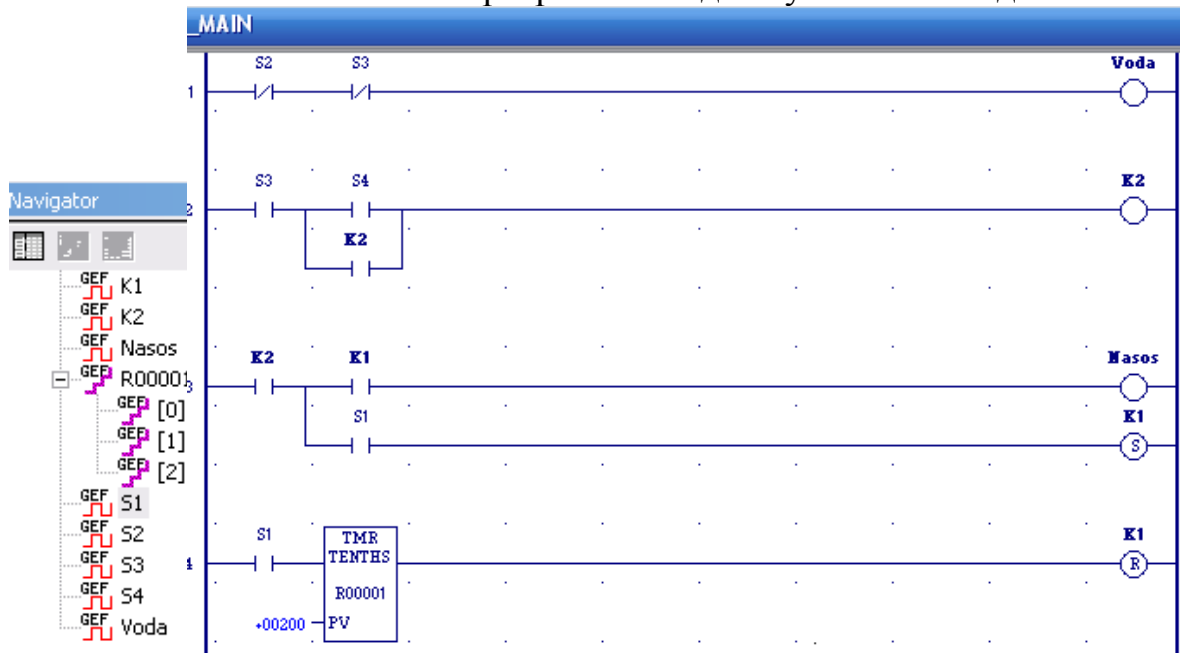
Затримка вимикання двигуна програмно реалізується за допомогою таймера. Щоб встановити таймер, необхідно в мережі праворуч від прямого контакту натиснути праву клавішу миші. Відкриється меню, в якому увійти у Place Instruction Ins і з каталогу, що відкриється, вибрати таймер TMR_TENTHS і двічі клацнути лівою клавішею миші. Таймер з'явиться у мережі. Щоб призначити йому адресу, треба двічі клацнути лівою клавішею миші по знакам питання, що показані усередині. З'явиться вікно у якому надрукувати адресу %R00001 і натиснути кнопку ОК.

Для оголошення параметрів таймера у списку змінних, що у навігаторі, відкрити вкладку таймера R00001, з'являться три слова (регістра) пам'яті %R для зберігання поточного (0) і заданого значення (1) параметрів таймера, а також керуючого слова. Натиснути ліву клавішу миші на першому параметрі і в інспекторі (Inspektor), що з'явиться, у рядку Initial Value надрукувати початкове значення 0 і клацнути клавішею Enter на клавіатурі. Так саме відкрити інспектор другого параметру і у тому ж рядку надрукувати задане значення затримки 200 (20с) і клацнути клавішею Enter на клавіатурі. Третій параметр залишити незмінним, оскільки таймер у LD-мережі вже з'єднаний з контактом S1. Щоб другий параметр таймера з'явився на робочому полі перед PV-входом таймера,

потрібно двічі клацнути лівою клавiшею миші перед ним і у вікні, що відкриється, надрукувати +00200 (20с) і натиснути Enter на клавіатурі.



Остаточно список змінних і програмний код набувають вигляд:



Для перевірки синтаксису створеного програмного коду клацнути правою клавiшею миші по цілі Target1, що у навігаторі проекту, і з меню, що розкриється, вибрати Validate.

При наявності помилок, про що свiдчить відповідне повідомлення, здійснити їх виправлення, а при відсутності можна з'єднати проект з контролером, використовуючи меню Target1->Go-Online, у навігаторі проекту. Завантаження програми здійснюється через меню Target1-> Download to Controller... після відповідних позначок у вікні Selections, що відкриється.

Для запуску ПЛК вибрати Start PLC у меню Target1->Offline Commands.

Завдання для самостійної роботи

Розробити FBD-мовою проект для контролера PACSystems RX7i, який керує насосом, що працює за алгоритмом, приведеним у даній роботі.

Зміст звіту

Звіт має містити проект, що створений відповідно до завдання на самостійну роботу.

Контрольні запитання

1. Яка послідовність дій при розробці нового проекту?
2. Як здійснюється конфігурування контролера?
3. Як створюється програмний код?
4. Як здійснюється відладка програми?
5. Як відбувається завантаження проекту у контролер?

Лабораторна робота №17

Створення проектів у системі програмування Step7

Мета: ознайомитися з методикою створення проектів для контролерів корпорації Siemens у системі програмування Step7.

Розробити LAD-мовою проект автоматичного керування роботою дробарки. При натисненні кнопки ПУСК спрацьовує попереджувальна сирена. За 10с сирена вимикається і після 30секундної паузи знову вмикається на 10с. Після другого попередження дробарка починає працювати. Якщо у приймальному бункері залишилася руда, вмикається в роботу конвеєр, який подає руду з бункера на дроблення. При цьому спрацьовує сигнальна лампа НОРМА на пульту оператора. Коли рівень руди у приймальному бункері стає нижче встановленого мінімального, про що свідчить сигнал позиційного датчика нижнього рівня і сигналізує лампа на пульту оператора, вмикається верхній конвеєр, який завантажує приймальний бункер дробарки рудою. Коли рівень руди в бункері досягне верхнього позиційного датчика, вмикається інша сигнальна лампа на пульту оператора і конвеєр, що подає руду у приймальний бункер, зупиняється. Оскільки нижній конвеєр працює, рівень в бункері зменшується і при досягненні встановленого мінімального, знову верхній конвеєр вмикається і подає руду у бункер. Вимикається дробарка з роботи після натиснення оператором кнопки ОСТАНОВ.

Створення проекту

Використовуючи ярлик SIMATIC Manager, відкрити вікно майстра проекту - STEP 7 Wizard: “New Project”:




Натиснути кнопку Next і у другій сторінці STEP 7 Wizard: “New Project” вибрати зі списку тип центрального процесора CPU314.

Перейти на третю сторінку, натиснувши кнопку Next, позначити лівою клавішею миші організаційний блок OB1 і мову програмування LAD.

Кнопкою Next підтвердити зроблені уставки, на останній сторінці вікна призначити ім'я проекту Drobarka і натиснути кнопку Finish. Відкриється екран проекту:



Щоб конфігурувати контролер, двічі натиснути ліву клавішу миші на етикетці  Hardware, що у правій панелі екрану. Відкриється вікно HW Config-SIMATIC з екраном конфігурування контролера SIMATIC 300 Station. На екрані знаходяться умовна стійка (0) UR з центральним процесором CPU314 у слоті 2 і конфігураційна таблиці (0) UR. Щоб встановити інші модулі на центральну стійку, у каталозі апаратури, що у правій частині конфігураційного вікна, відкрити вкладку SIMATIC 300, а потім теку модулів живлення PS 300 і вибрати курсором PS307 2A. Натиснути ліву клавішу миші і модуль живлення з'явиться у монтажній стійці (0) UR, а також у конфігураційній таблиці (0) UR. Щоб вибрати сигнальні модулі потрібно послідовно відкрити теки SM-300 і DI-300. За допомогою миші і натиснутої лівої клавіші миші перетягнути модуль вводу DI16xDC24V, а потім з теки DO-300 перетягнути модуль виводу DO16xDC24V/0,5A, відповідно у 4-й і 5-й слоти стійки. Автоматично вибрані модулі з'являються і у конфігураційній таблиці з призначеними адресами.

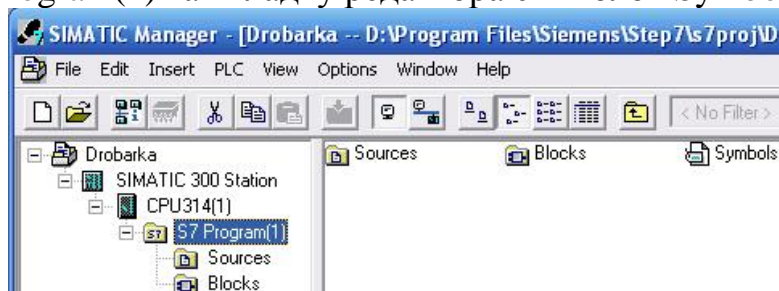
Слот третій, призначений для інтерфейсного модуля, залишається вільним.



The screenshot shows the 'SIMATIC 300 Station (Configuration)' window. On the left, a rack diagram shows slots 1 to 5 with modules: PS 307, CPU 314, and two DO16xDC modules. The main table below lists the configured modules:

Slot	Module	Order number	Firmware	MPI address	I address	Q address
1	PS 307 2A	6ES7 307-1BA00-0AA0				
2	CPU314(1)	6ES7 314-1AE04-0AB0		2		
3						
4	DI16xDC24V	6ES7 321-1BH01-0AA0			0...1	
5	DO16xDC24V/0.5A	6ES7 322-1BH01-0AA0				4...5

Після закриття додатку HW Config-SIMATIC на екрані EOM відкриється вікно проекту з двома панелями. Ліва панель містить структуру проекту, а права- зміст теки S7 Program(1) та вкладку редактора символів Symbols.



Тека Sources використовується для зберігання програм у вигляді початкових файлів. Тека Blocks призначена для зберігання блоків програми. Вкладка Symbols використовується для надання адресам символічних імен.

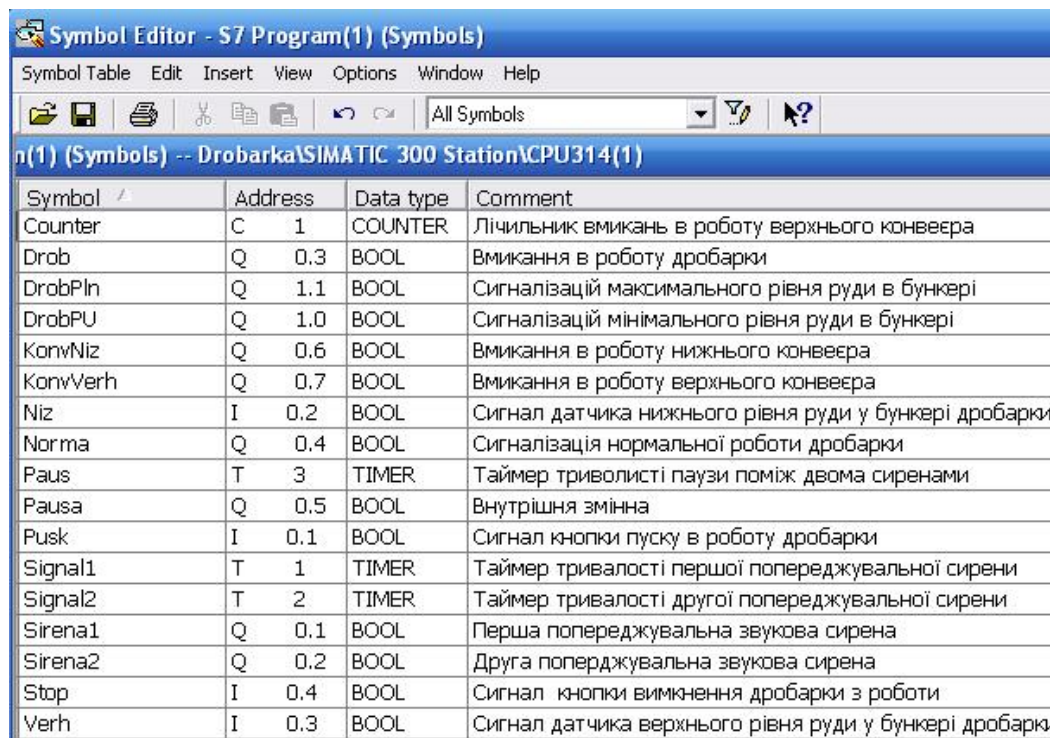
Створення програмного коду

Створення програмного коду починається з призначення символічних імен і адрес вхідним та вихідним змінним, які будуть використовуватися при

створенні програми. Для цього двічі натиснути ліву клавішу миші на етикетці Symbols і у вікні, що відкриється, заповнити таблицю символів. При наданні змінним їх адрес дотримуватися таких позначень:

I - вхідний біт; Q - вихідний біт; T – таймер; C – лічильник.

Таблиця Symbol Editor після заповнення її всіма змінними має вигляд:

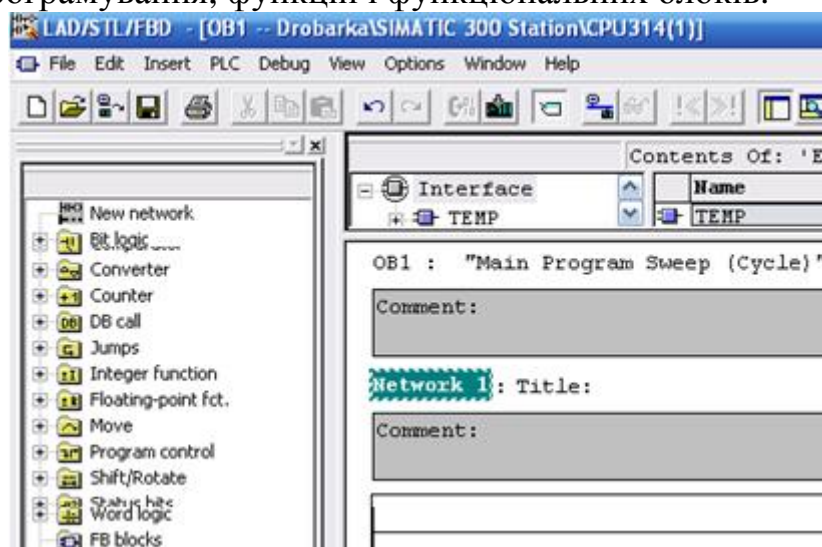


Symbol	Address	Data type	Comment
Counter	C 1	COUNTER	Лічильник вмикань в роботу верхнього конвеєра
Drob	Q 0.3	BOOL	Вмикання в роботу дробарки
DrobPln	Q 1.1	BOOL	Сигналізацій максимального рівня руди в бункері
DrobPU	Q 1.0	BOOL	Сигналізацій мінімального рівня руди в бункері
KonvNiz	Q 0.6	BOOL	Вмикання в роботу нижнього конвеєра
KonvVerh	Q 0.7	BOOL	Вмикання в роботу верхнього конвеєра
Niz	I 0.2	BOOL	Сигнал датчика нижнього рівня руди у бункері дробарки
Norma	Q 0.4	BOOL	Сигналізація нормальної роботи дробарки
Paus	T 3	TIMER	Таймер триволістї паузи поміж двома сиренами
Pausa	Q 0.5	BOOL	Внутрішня змінна
Pusk	I 0.1	BOOL	Сигнал кнопки пуску в роботу дробарки
Signal1	T 1	TIMER	Таймер тривалості першої попереджувальної сирени
Signal2	T 2	TIMER	Таймер тривалості другої попереджувальної сирени
Sirena1	Q 0.1	BOOL	Перша попереджувальна звукова сирена
Sirena2	Q 0.2	BOOL	Друга попереджувальна звукова сирена
Stop	I 0.4	BOOL	Сигнал кнопки вимкнення дробарки з роботи
Verh	I 0.3	BOOL	Сигнал датчика верхнього рівня руди у бункері дробарки



Для збереження змісту таблиці, у панелі інструментів натиснути кнопку .




Після створення символної таблиці у правій панелі екрану відкрити теку Blocks подвійним клацанням лівої клавіші миші, а потім об'єкт OB1.

Відкриється робоче поле для створення програмного коду і бібліотека символів мови програмування, функцій і функціональних блоків.



Для створення першої LAD-мережі (Network1), яка забезпечить вмикання попереджувальної сирени, необхідно позначити її логічну шину лівою клавішею миші. У бібліотеці New network, що у лівій панелі екрану відкрити теку Bit

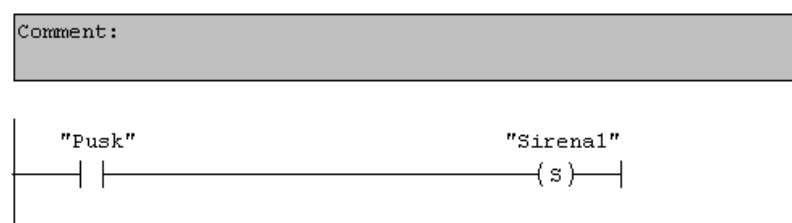
Logic і подвійним клацанням лівої клавіші миші по символу прямого контакту  встановити його на логічну шину. Над елементом у місці розташування його імені знаходяться знаки питання. Лівою клавішею миші позначити логічну шину праворуч від встановленого контакту і двічі клацнути на S - котушці  у теці Bit Logic. В LAD-мережі (Network1) з'явиться S-котушка. Щоб призначити елементам мережі імена змінних потрібно клацнути лівою клавішею миші по символам питання над елементами і у прямокутнику, що з'явився, відкрити таблицю символів, натиснувши клавішу пробілу на клавіатурі EOM. Після дворазового натиснення лівої клавіші миші на вибраному імені, воно з'являється над елементом мережі.

Для створення другої LAD-мережі (Network2), яка забезпечить тривалість попереджувального сигналу, потрібно в панелі інструментів натиснути ліву клавішу миші на кнопці . За аналогією з першою мережею встановити прямий контакт, R-котушку  і призначити їм ім'я Sirena1. Для встановлення паралельної S-котушки, яка має сформувати сигнал початку паузи, треба позначити логічну шину поміж прямим контактом і R-котушкою і в панелі інструментів на кнопці  Open Branch натиснути ліву клавішу миші. Коли у мережі з'явиться нова гілка двічі натиснути ліву клавішу миші на S-котушці у теці Bit_logic бібліотеки New network і з таблиці символів призначити їй ім'я Pausa.

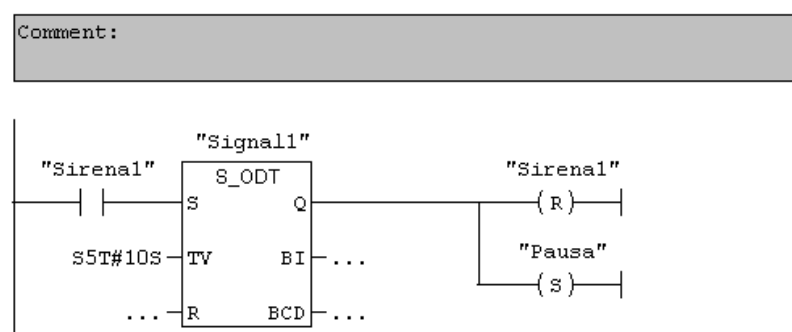
Щоб встановити таймер першої сирени в LAD-мережу (Network2) треба лівою клавішею миші позначити логічну шину поміж контактом і точкою розгалуження логічної шини. В теці Timers бібліотеки New network подвійно клацнути по функціональному блоку S_PULSE:

Після установа таймера у другу LAD-мережу (Network2) замість знаків питання присвоїти йому ім'я Signal1, а біля входу TV записати s5T#10s - 10с затримки. Тоді програмний код перших двох фрагментів отримує вигляд:

Network 1 : Перша попереджувальна звукова сирена



Network 2 : Таймер тривалості першої попереджувальної сирени



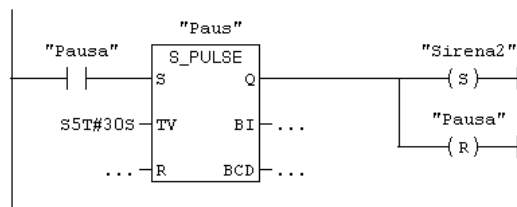
Наступні LAD-мережі, що забезпечують паузу поміж сиренами, тривалість другої попереджувальної сирени, вмикання і вимикання в роботу

дробарки, світлової індикації, верхнього та нижнього конвеєрів. а також автоматичне регулювання рівнем руди у приймальному бункері, створюються за тою самою методикою, що і попередні LAD-мережі.

Решта LAD-мереж програмного коду управління роботою дробарки має вигляд:

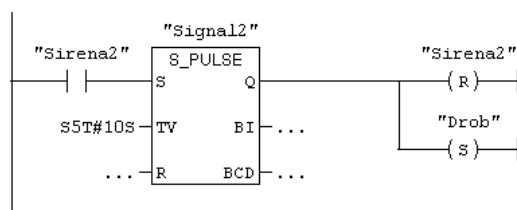
Network 3: Таймер паузи поміж сиренами

Comment:



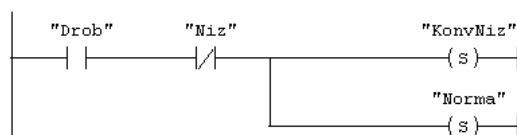
Network 4: Таймер тривалості другої попереджувальної сирени

Comment:



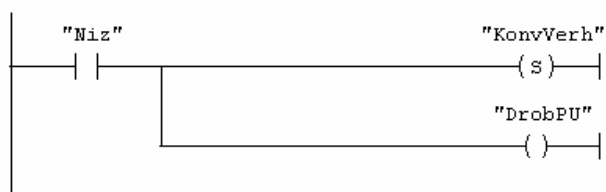
Network 5: Увімкнення в роботу нижнього конвеєра і світової сигналізації

Comment:



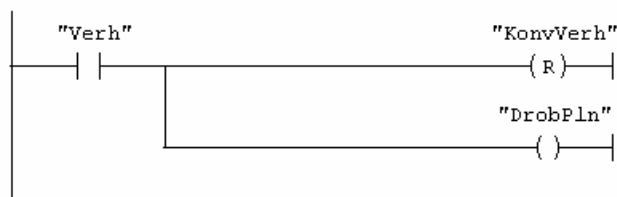
Network 6: Увімкнення в роботу верхнього конвеєра

Comment:

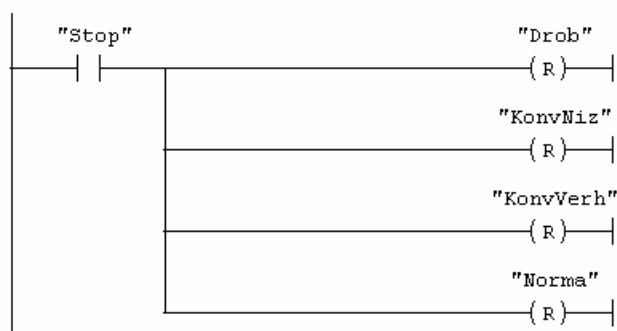






Network 7 : Вимкнення з роботи верхнього конвеєра

Comment:

**Network 8 : Вимкнення з роботи дробарки, конвеєрів і світової сигналізації**

Comment:



Для перевірки роботи створеної програми треба перейти у режим симуляції, натиснувши кнопку  Simulation on/off у вікні SIMATIC Manager. Відкриється вікно S7-PLCSIM-SimView1 з симулятором ЦПУ. Додатково відкрити симулятор модуля вводу, використовуючи в меню шлях Insert -> Input Variable. У вікні SIMATIC Manager натиснути кнопку  Download, щоб завантажити проект до симулятора, і відкрити програмний блок OB1, двічі натиснувши ліву клавішу миші на кнопці  OB1 Organization Block. Кнопкою  Monitor(on/off) запустити режим симуляції і у вікні S7-PLCSIM-SimView1 активізувати ЦПУ прапорцем біля RUN. Враховуючи те, що пуск в роботу дробарки, здійснюється активізацією змінної Pusk, яка має адресу I0.1, необхідно поставити прапорець на першому каналі симулятора модуля вводу. Програма почне працювати відповідно до алгоритму роботи дробарки. Щоб перевірити вплив решти вхідних змінних на хід виконання програми, необхідно після запуску її в роботу змінювати стан відповідних бітів у симуляторі модуля вводу.

Програма у режимі симуляції при активізованому біті змінної Pusk має вигляд:

Лабораторна робота №18

Створення проектів у системі програмування ISaGRAF

Мета: ознайомитися з методикою створення проектів для контролерів, що програмуються у системі ISaGRAF.

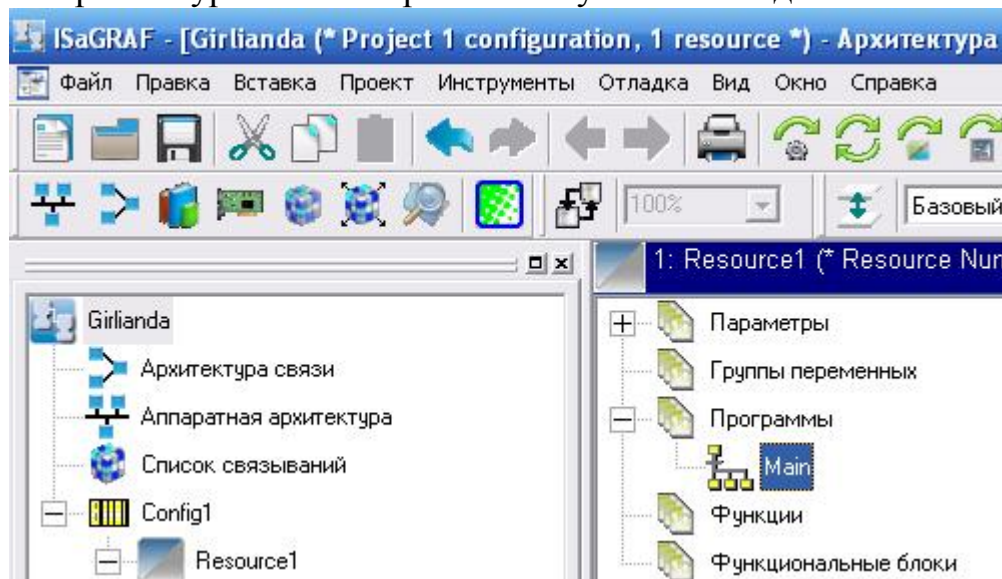
Розробити проект SFC-мовою, що керує послідовним вмиканням 4-х ламп гірлянди. При цьому кожний цикл в роботі гірлянди фіксується лічильником, який після 15 циклів скидається. Початок роботи гірлянди визначається вимикачем напруги.

Розроблення проекту

Запустити програму ISaGRAF 5.2, використовуючи меню ПУСК ->Все программы-ISaGRAF Products-> ISaGRAF 5.2. Увійти в меню «Файл->Создать проект/библиотеку» і у вікні «Новый проект», що відкриється, призначити ім'я проекту, наприклад Girlianda, а також вибрати шаблон Primonoresource. Після підтвердження назви кнопкою ОК, на екрані з'явиться організатор проекту, а його вкладка Resource1, з відповідними параметрами, продублюється праворуч.

Якщо вкладки Resource1 немає, двічі натиснути ліву клавішу миші на «Архитектура связи» в організаторі програми Girlianda.

Щоб додати в організаторі програми, SFC-модуль для створення програмного коду з назвою, наприклад Main, необхідно клацнути правою клавішею миші по вкладці «Программы» і вибрати відповідний програмний модуль. Вікно «Архитектура связи» отримає наступний вигляд:



Для оголошення змінних, в панелі інструментів кнопкою «Словарь» відкрити вікно словника змінних і перейти на вкладку «Переменные». Натиснути ліву клавішу миші на позначці «Глобальные переменные», щоб відкрити відповідну таблицю.

Для її заповнення треба перейти у режим Строка. Для цього клацнути правою клавішею миші на вільному рядку вікна «Глобальные переменные», вибрати «Переключить в режим Строка» і підтвердити лівою клавішею миші.

Після цього двічі клацнути лівою клавішею миші на вільному рядку таблиці. З'явиться вікно для оголошення змінної:

Lamp1

Имя: Алиас: Тип:

Нач. значение: Размерность:

Группа: Атрибут: Видимость:

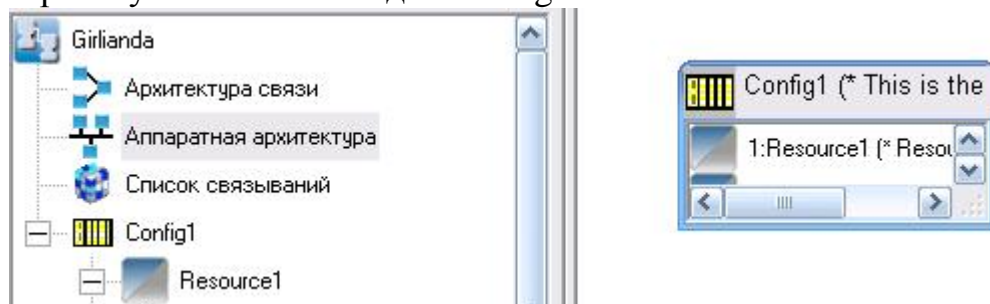
Направление: Сохранение: Монтаж:

Адрес: Комментарий:



Використовуючі таке вікно для оголошення усіх вихідних змінних Lamp1, Lamp2, Lamp3, Lamp4, Counter, а також вхідної Power, сформується таблиця глобальних змінних:

Глобальные переменные									
Имя	Алиас	Тип	Группа	Атрибут	Видимо...	Направ...	Сохран...	Адрес	Комме...
Lamp1		BOOL	None	Free	Global	Выход	No		Перша л...
Lamp2		BOOL	None	Free	Global	Выход	No		Друга л...
Lamp3		BOOL	None	Free	Global	Выход	No		Третья ла...
Lamp4		BOOL	None	Free	Global	Выход	No		Четверт...
Power		BOOL	None	Read	Global	Вход	No		Вмикання
Counter		INT	None	Free	Global	Выход	No		Лічильник
...									

Для створення апаратної архітектури в організаторі проекту двічі натиснути ліву клавішу миші на вкладці «Аппаратная архитектура». На екрані праворуч від організатора проекту з'явиться вкладка Config1:



Двічі клацнути лівою клавішею миші по вкладці Config1. У вікні «Конфигурация: Свойства», що відкриється, вибрати цільову систему SIMULATOR і натиснути OK.

Щоб з'єднати оголошені змінні з пристроями вводу-виводу треба в панелі інструментів кнопкою  «Монтаж ВВ» відкрити відповідне вікно. Кнопкою  «Добавить устройство» відкрити вікно «Выбор устройства», вибрати в ньому драйвер Drv_:Sim_:SimInBoo(*Simulation Boolean Input device*) і визначити один вхідний канал для змінній Power, рис.18а. Натиснути кнопку OK і у лівій частині вікна «Монтаж ВВ» з'явиться пристрій вводу на один канал, а у правій частині – назва вхідній змінній Power, рис. 18.1б.

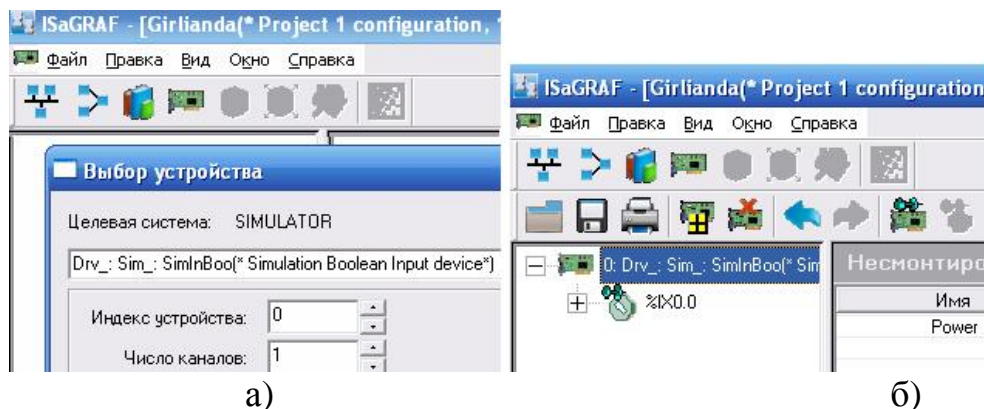
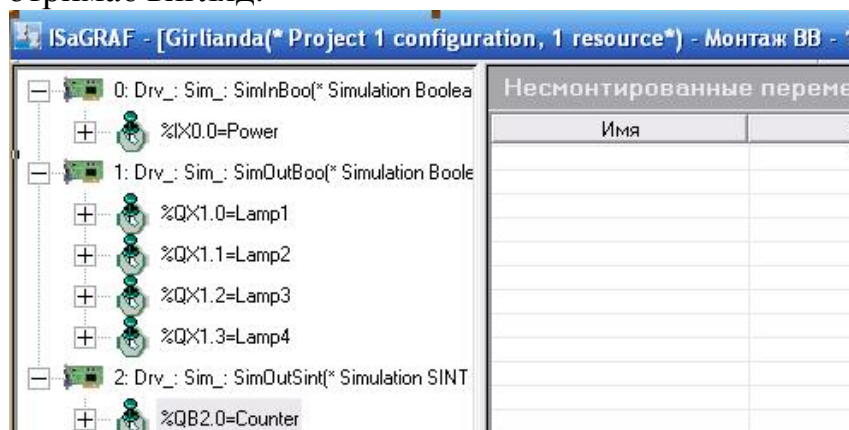


Рисунок 18.1 Вибір цільової системи (а) і з'єднання змінної з каналом пристрою вводу-виводу (б)

Позначити лівою клавішею миші канал з адресою %IX0.0 і подвійно клацнути по змінній Power. Після цього змінна стає змонтованою і переходить у ліву половину вікна з адресою вхідного каналу.

Аналогічно змонтувати вихідні булеві змінні, тільки у вікні «Выбор устройства» вибрати драйвер Drv_:Sim_:SimOutBool(*Simulation Boolean Input device*) і встановити 4 канали виводу для булевих змінних Lamp1, Lamp2, Lamp3, Lamp4. Для підключення цілочислової вихідної змінної Counter до пристрою виводу, у вікні «Выбор устройства», вибрати драйвер Drv_:Sim_:SimOutSint(*Simulation SINT Output device*) і встановити 1 канал виводу. Після приєднання змінної Counter до каналу пристрою виводу, вікно «Монтаж ВВ» отримає вигляд:






А у словнику змінних з'являться монтажні адреси змінних:

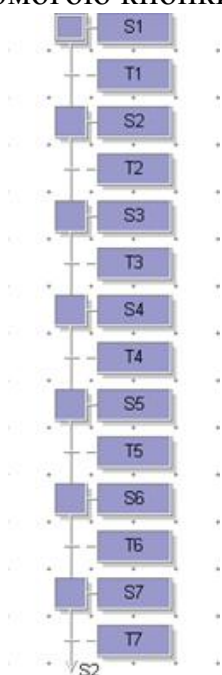
Имя	Алиас	Тип	Группа	Атрибут	Видимо...	Направ...	Сохран...	Монтаж
Lamp1		BOOL	None	Free	Global	Выход	No	%QX1.0
Lamp2		BOOL	None	Free	Global	Выход	No	%QX1.1
Lamp3		BOOL	None	Free	Global	Выход	No	%QX1.2
Lamp4		BOOL	None	Free	Global	Выход	No	%QX1.3
Power		BOOL	None	Read	Global	Вход	No	%IX0.0
Counter		SINT	None	Free	Global	Выход	No	%QB2.0

Створення програмного коду

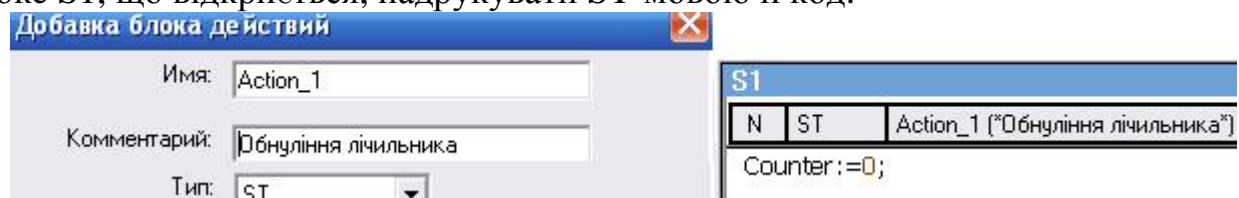
В організаторі програми двічі клацнути лівою клавішею миші по назві програми Main. Відкриється робоче поле SFC-редактора з початковим кроком

програмного коду. Якщо його немає, в панелі інструментів клацнути лівою клав'яшею миші по кнопці початкового кроку  і перевести курсор на робоче поле. У зручному місці клацнути лівою клав'яшею миші і початковий крок залишиться на екрані.

Клацнути лівою клав'яшею миші по кнопці  «Переход», перевести курсор під початковий крок і натиснути ліву клав'яшу миші. На екрані залишиться перша зв'язка «крок-перехід». Відповідно до алгоритму програми програмний код повинен мати сім кроків і сім переходів. Причому після останнього кроку має відбуватися перехід на другий крок, щоб забезпечити циклічність у роботі програми. Здійснюється це за допомогою кнопки  «Связь»:



Щоб визначити дії кроків необхідно на відповідному кроці двічі клацнути лівою клав'яшею миші. Праворуч на екрані з'явиться поле на якому потрібно клацнути правою клав'яшею миші, щоб додати блок дій. Після заповнення усіх рядків вікна «Добавка блока действий», щодо назви дії, коментарів, мови програмування та специфікатору, натиснути кнопку ОК і у робочому полі дії кроке S1, що відкриється, надрукувати ST-мовою її код:




Що стосується переходів, то тільки умови переходу T2 залежать від стану змінної Power, решта переходів здійснюється без будь-яких умов. Методика створення умов переходів така сама, як і для дій кроків.

Остаточно код програми, а також дій кроків і умов переходів має наступний вигляд:



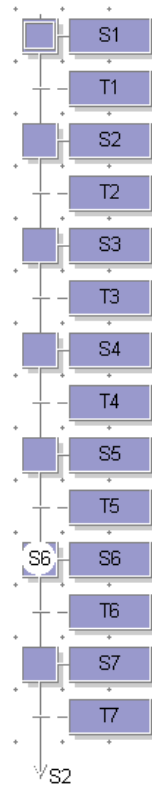
Для компіляції проекту у панелі інструментів натиснути кнопку «Компоновать». Якщо робочий код програми створений відповідно до синтаксису мов, що використовувалися, у нижній частині екрану з'явиться повідомлення про відсутність помилок та попереджень.

У протилежному випадку усі помилки необхідно виправити і провести повторну компіляцію проекту.

Після успішної компіляції можна моделювати роботу програми, натиснувши в панелі інструментів кнопку  «Моделировать». У нижній частині екрану з'явиться етикетка прихованого імітатора вводу-виводу. Відкрити його лівою клавішею миші і на екрані з'явиться вікно «Имитатор ВВ», на якому показані – відкрита вкладка CONFIG1 проекту GIRLIANDA, а також стани змінних усіх пристроїв вводу-виводу симулятора. Якщо подвійним клацанням лівої клавіші миші змінити стан вхідної змінної Power з FALS на TRUE, то відповідно до алгоритму роботи програми почнуть змінюватися стани вихідних змінних Lamp1, Lamp2, Lamp3 і Lamp4, а лічильник Counter відстежує кількість циклів у роботі гірлянди, рис.18.2а. Роботу програми можна також побачити на робочому полі, де крок, що знаходиться у стані виконання, позначається білою міткою, рис.18.2б.



а)



б)

Рисунок 18.2– Емуляція виконання програми у вікні «Имитатор ВВ (а) і на робочому полі (б)

Щоб завантажити програму у ПЛК, необхідно приєднатися до контролера і змінити конфігурацію апаратної структури. Для цього у вікні «Конфигурация: Свойства» у якості цільової системи вибрати не Simulator, а WIN32-TGT_L і монтування змінних здійснити відповідно до вибраних типів модулів вводу-виводу.

Завдання для самостійної роботи

Розробити FBD-мовою проект, що працює за алгоритмом, приведеним у даній роботі.

Зміст звіту

Звіт має містити проект, що створений відповідно до завдання на самостійну роботу.

Контрольні запитання

1. Яка послідовність дій при розробці нового проекту?
2. Як здійснюється конфігурування контролера?
3. Як створюється програмний код?
4. Як здійснюється відладка програми?
5. Як відбувається завантаження проекту у контролер?

Лабораторна робота №19

Створення проектів у системі програмування GX IEC Developer

Мета: ознайомитися з методикою створення проектів для контролерів корпорації Mitsubishi Electric у системі програмування GX IEC Developer.

Розробити LD-програму керування електродвигунами M1, M2, відповідно, горизонтального та похилого транспортерів, що перевантажують сміття у самохідний візок.

При пуску лінії першим вмикається двигун M2, а за 10с – двигун M1.

В нормальному режимі роботи похилий транспортер має працювати 10 хвилин, але першим за 20с до зупинення M2 має вимкнутися M1. У випадку малої кількості сміття оператор може достроково зупинити процес, натискаючи кнопку «Стоп». При цьому першим зупиняється M1, а за 20с – M2. При аварійній ситуації, одночасно натискаючи кнопки «Пуск» і «Стоп», можна зупинити одразу M1 і M2. Повторне (помилкове) натискання цих кнопок не вмикає транспортери. Нове вмикання транспортерів можливе тільки через 10 хвилин, коли двигуни M1 і M2 охолodяться.


Створення проекту

Запустити програму, використовуючи шлях MELSOFT Application GX IEC → Developer Ver.7.04 → GX IEC Developer Ver.7.04. Для створення нового проекту увійти в меню “Project” → “New...” і у вікні Select PLC Type, що з’явиться, вказати серію контролера Q та його тип Q00J.

Після підтвердження зроблених уставок у вікні New Project вибрати шлях зберігання проекту Transport на жорсткому диску і натиснути кнопку Create. Відкриється вікно «GX IEC Developer Ver.7.04 New Project Startup Options», в якому вибрати мову програмування Ladder Diagram (LD) і натиснути OK.



На екрані з’явиться вікно проекту Transport GX IEC Developer Ver.7.04.

Створення програмного коду


Перед тим, як створювати код програми необхідно оголосити глобальні змінні проекту: Pusk, Stop, Motor1, Motor2, які забезпечують запуск, зупинення і роботу двигунів двох транспортерів. Для цього у дереві проекту двічі клацнути лівою клавішею миші по «Global_Vars», щоб з’явилася таблиця глобальних змінних, яку можна збільшувати за допомогою кнопки  в панелі інструментів. У відповідних колонках надрукувати змінні, їх адреси і вибрати тип:

	Class	Identifier	MIT-Addr.	IEC-Addr.	Type	Initial
0	VAR_GLOBAL	Pusk	X1	%IX1	BOOL	FALSE
1	VAR_GLOBAL	Stop	X2	%IX2	BOOL	FALSE
2	VAR_GLOBAL	Motor1	Y1	%QX1	BOOL	FALSE
3	VAR_GLOBAL	Motor2	Y2	%QX2	BOOL	FALSE

Щоб відкрити редактор програмного коду, необхідно у дереві проекту двічі клацнути лівою клавішею миші по Body [LD].

У вихідному стані вікно редактора має одну ланку для створення програмного коду. Для збільшення кількості ланок призначені кнопки   у панелі інструментів, а для зменшення використовується клавіша Delete на клавіатурі ЕОМ.


Відповідно до алгоритму керування першим вмикається двигун М2, а двигун М1 вмикається за 10с.

Для створення першої мережі, яка імітує роботу кнопки з самоповерненням, в панелі інструментів вибрати прямий контакт  “Contact” і клацнути лівою клавішею миші. Для встановлення його у ланку робочого поля, повторно клацнути на потрібному місці. Щоб призначити контакту ім'я змінної Pusk, яка була оголошена, як глобальна, необхідно клацнути лівою клавішею миші по знаку питання над контактом, у віконці, що з'явиться, надрукувати це ім'я і натиснути клавішу Enter на клавіатурі ЕОМ. Щоб з'єднати прямий контакт з лівою шиною живлення клацнути правою клавішею миші біля контакту, відкриється меню. Активізувати альтернативу Interconnect Mode, замість курсору з'явиться олівець, яким з'єднати прямий контакт і ліву шину живлення. При цьому на початку і наприкінці з'єднання потрібно клацнути лівою клавішею миші.

Оскільки кнопка Пуск із самоповерненням потрібно послідовно з прямим контактом Pusk встановити таймер ТР, що формує імпульсний сигнал. Для цього натиснути праву клавішу миші, з меню, що відкриється, вибрати OP/FUN/FB і натиснути ліву клавішу миші. Відкриється вікно Function Block Selection, в якому вибрати бібліотеку Standart_Lib (Scope), функціональний блок ТР (Operators), його тип Function Blocks (Operators Type) і натиснути кнопку Apply:

Для встановлення вибраного блока ТР у мережу програмного коду, необхідно повторно клацнути лівою клавішею миші у потрібному місці. Коли функціональний блок з'явиться, треба його оголосити, як локальний. Для цього лівою клавішею миші активізувати типове ім'я Instance і натиснути праву клавішу миші. У меню, що відкриється, активізувати альтернативу List Operands, у вікні Variable Selection серед запропонованих Scop вибрати <Header>, у рядку Variables надрукувати ім'я ТР1, вказати тип функціонального блока – ТР і натиснути послідовно кнопки Define, Apply і Close.

Нова назва функціонального блока з'явиться над ним. Щоб приєднати вхід функціонального блока до контакту Pusk, треба скористатися послідовністю дій, що були наведені при приєднанні контакту до лівої шини живлення.

Для створення проміжної внутрішньої змінної К1, що керуватиме вмиканням двигуна другого транспортеру, необхідно в панелі інструментів лівою клавішею миші активізувати кнопку  «Coil» і повторно клацнути у потрібному місці мережі, що створюється. За аналогією з контактом Pusk приєднати котушку до функціонального блока ТР1 і оголосити її, як локальну змінну К1.

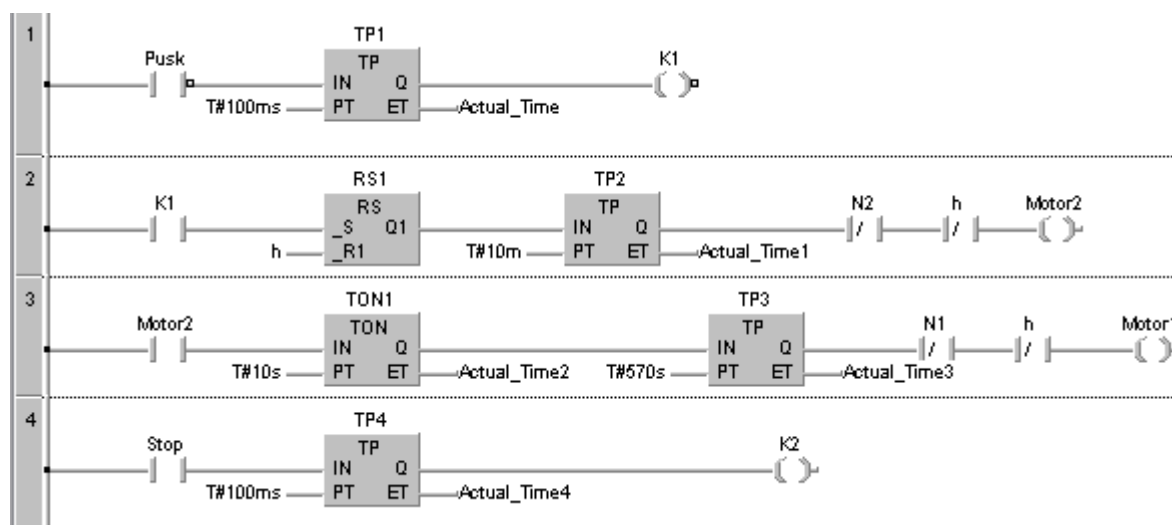
Щоб зафіксувати сигнал керування роботою двигуна від короткочасного спрацьовування кнопки Пуск, необхідно створити другу мережу з RS-тригером і

ТР-таймером, які забезпечують вмикання двигуна другого транспортеру за допомогою вихідної змінної Motor2 і його роботу на протязі 10 хвилин.

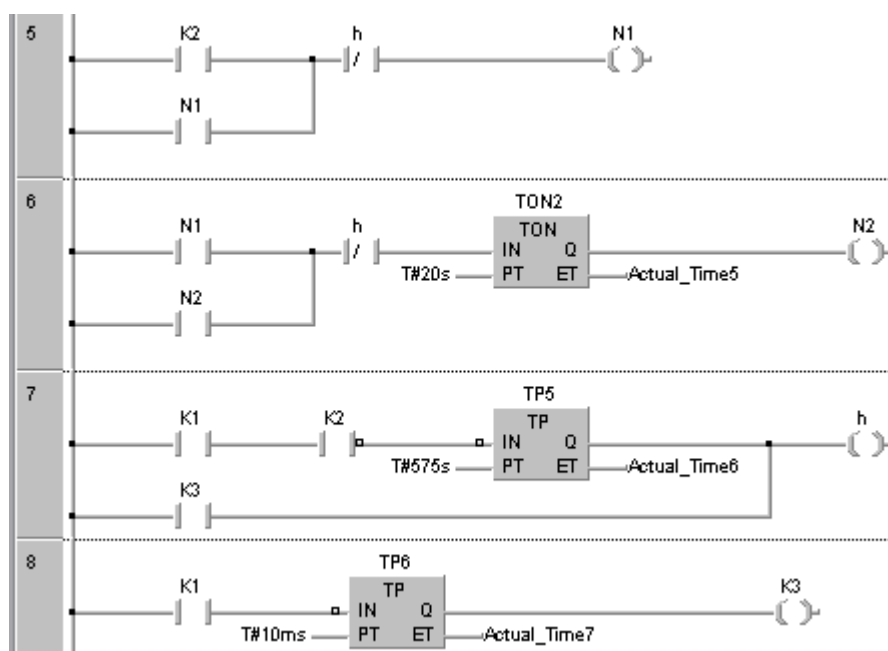
У третій мережі за допомогою TON-таймера забезпечити вмикання двигуна Motor1 першого транспортера із затримкою у 10с і його вимикання за 20с до зупинення двигуна Motor2 другого транспортера за допомогою ТР-таймера.

Для можливості дострокового зупинення роботи транспортерів у четвертій мережі необхідно передбачити змінну Stop, яка як і змінна Push, керується кнопкою з самоповерненням.

Початковий фрагмент програмного коду має наступний вигляд:




Тут h, N1, N2 додаткові внутрішні змінні, керування якими забезпечує решта програмного коду:

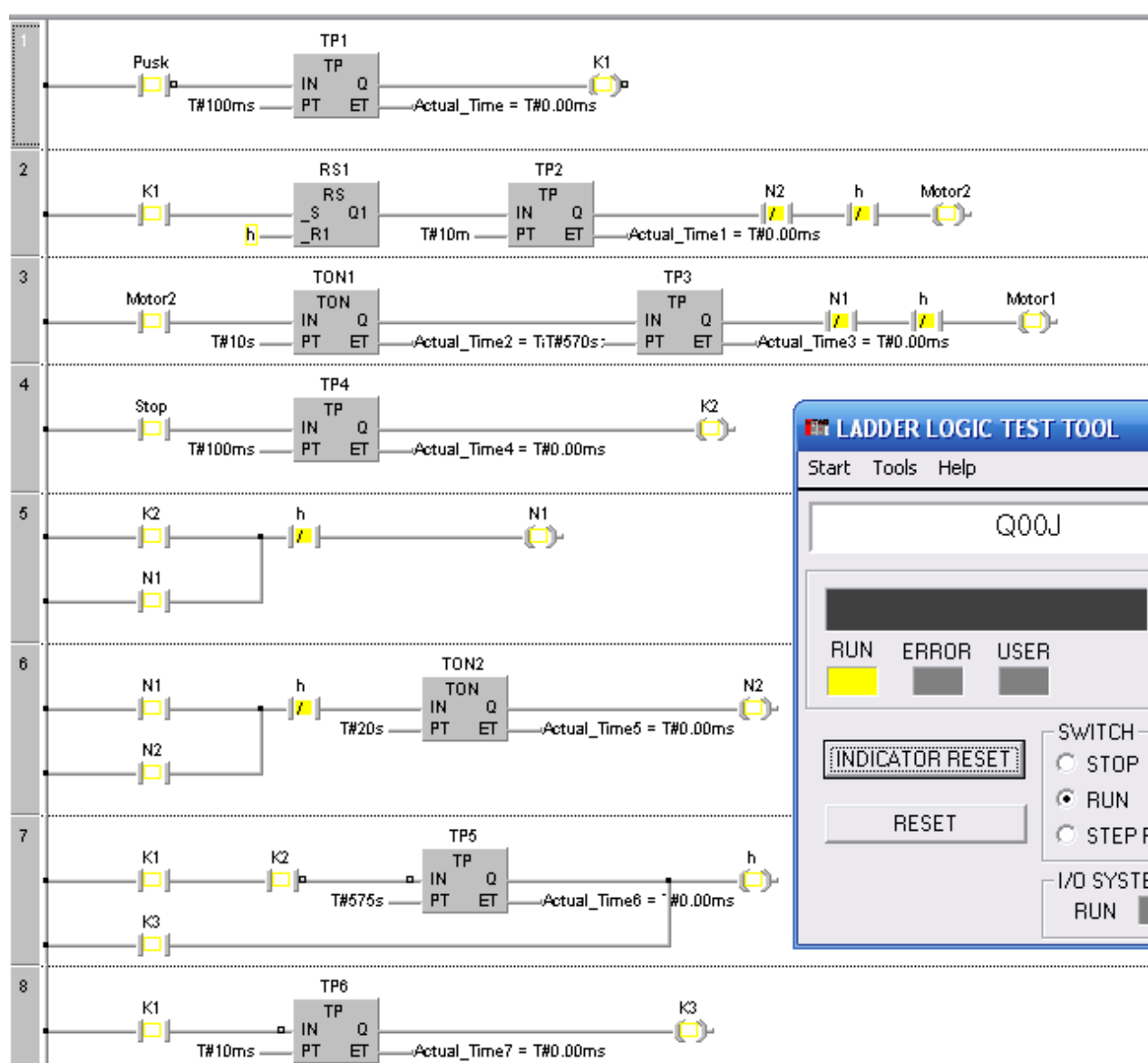


Щоб програмний код виконувався контролером, необхідно створити відповідну задачу. Для цього в дереві проекту треба клацнути правою клавішею

миші на Task_Pool і у меню, що відкриється, вибрати “New Task...”. У однойменному вікні, що з’явиться, надрукувати ім’я задачі Task_Trans і натиснути OK.

Для компіляції програмного коду, в панелі інструментів натиснути лівою клавішею миші кнопку  «Build». При наявності помилок, програма їх виділить червоним кольором. Після виправлення помилок ще раз здійснити компіляцію.

Для перевірки роботи програми в меню проекту Online лівою клавішею миші активізувати пункт GX Simulator. Коли симулятор завантажиться, з’явиться вікно Ladder Logic Test Tool з панеллю тестування віртуального контролера Q00J і активізуються елементи програмного коду, про що свідчить жовтий колір. При цьому пункт Start Monitoring у меню Online увімкнеться.




Для тестування роботи програми у вікні Ladder Logic Test Tool треба увійти в меню Start → Monitor Funktion → Device Memory Monitor. З’явиться

діалог Transfer setup, у якому встановити настройку передачі даних – Host Station і натиснути ОК.

Щоб моніторити вихідні змінні треба у вікні Device Memory Monitor, що з'явиться, відкрити вкладку Device Batch і в полі Device надрукувати адресу першого виходу контролера Y0. Натиснути кнопку Start monitor і у таблиці виходів з'являться нульові значення сигналів.

Для симуляції вхідних змінних перейти на вкладку Entry Device та натиснути кнопку Register Device. У однойменному вікні в полі Device надрукувати адресу першої змінної X1(Pusk) та натиснути кнопку Register. Так саме ввести адресу змінної X2(Stop). Коли змінні X1 і X2 з'являться у таблиці вікна Device Memory Monitor, натиснути кнопку Close. Якщо список у цьому вікні містить змінні, що залишилися від попередньої симуляції, то спочатку їх необхідно видалити натискаючи кнопку Clear Device.

Щоб керувати станом вхідних змінних типу Bool, необхідно у таблиці вікна Device Memory Monitor натиснути кнопку Start monitor. Після того, як проти вхідних змінних з'являться нулі, двічі клацнути лівою клавішею миші по змінній, вплив якої на роботу програми досліджується. У вікні Device Write, що з'явиться, скористатися кнопками – Force On (True), Force Off (False) або кнопкою Toggle force (Змінити стан) і натиснути кнопку Close. Проти змінної з'явиться 1 і програма почне працювати.

Після перевірки програми завантажити проект у контролер натиснувши лівою клавішею миші кнопку  Download Project в панелі інструментів.

У вікні Transfer to PLC, що з'явиться, задати та сконфігурувати порт передачі даних між контролером та комп'ютером (Transfer Setup Ports), а також параметри передачі проекту (Transfer Setup Project) і натиснути ОК.

Завдання для самостійної роботи

Розробити FBD-мовою проект, що працює за алгоритмом, приведеним у даній роботі.

Зміст звіту

Звіт має містити проект, що створений відповідно до завдання на самостійну роботу.

Контрольні запитання

1. Яка послідовність дій при розробці нового проекту?
2. Як здійснюється конфігурування контролера?
3. Як створюється програмний код?
4. Як здійснюється відладка програми?
5. Як відбувається завантаження проекту у контролер?

Література

1. Ніколаєнко А.М., Міняйло Н.О. Мікропроцесорні та програмні засоби автоматизації: Навчальний посібник/ – Запоріжжя, ЗДІА, 2011. – 444 с.