

Міністерство освіти і науки, молоді та спорту України  
Запорізька державна інженерна академія

**А.М.Ніколаєнко**  
**Н.О.Міняйло**

# **МІКРОПРОЦЕСОРНІ ТА ПРОГРАМНІ ЗАСОБИ АВТОМАТИЗАЦІЇ**

*Рекомендовано*  
*Міністерством освіти і науки, молоді та спорту України*  
*як навчальний посібник*  
*для студентів вищих навчальних закладів*

Запоріжжя  
ЗДІА  
2011

УДК 621.382.049.77.004.4(075)

ББК 32973.2я7

Н 634

*А.М. Ніколаєнко, кандидат технічних наук, професор  
Н.О. Міняйло, кандидат технічних наук, доцент*

*Рекомендовано до друку:  
рішенням вченої ради Запорізької державної інженерної академії (протокол № 6  
від 03.02. 2011р.*

**Рецензенти:**

**О.М. Кукушкін**, Заслужений діяч науки та техніки України, Лауреат Державної премії УРСР в галузі науки і техніки, доктор технічних наук, професор кафедри автоматизації виробничих процесів Національної металургійної академії України (м. Дніпропетровськ);

**В.В. Ткачов**, доктор технічних наук, професор, завідувач кафедри автоматизації та комп'ютерних систем Державного вищого навчального закладу «Національний гірничий університет». (м. Дніпропетровськ);

**В.В. Зіновкін**, доктор технічних наук, професор кафедри електроприводу і автоматизації промислових установок Запорізького національного технічного університету. (м. Запоріжжя).

**Ніколаєнко А.М., Міняйло Н.О.**

**Н634 Мікропроцесорні та програмні засоби автоматизації: Навчальний посібник/**

**Ніколаєнко А.М., Міняйло Н.О.** – Запоріжжя, ЗДІА, 2011. – 444 с.

**ISBN 978-966-8462-51-1**

У навчальному посібнику описано тенденції розвитку систем управління та програмованих логічних контролерів, будова і організація роботи ПЛК, можливості та основні технічні характеристики, на підставі яких здійснюється вибір контролерів, в адаптованій формі надана технічна інформація про найбільш поширені ПЛК всесвітньо відомих і вітчизняних виробників, методика створення програмних додатків користувача за допомогою інструментальних пакетів програмування, а також приведені лабораторний практикум і курсова робота з прикладами виконання.

Навчальний посібник призначений для студентів, які навчаються за напрямом «Автоматизація та комп'ютерно-інтегровані технології», а також може бути корисний інженерно-технічним працівникам і розробникам АСУ ТП.

Рекомендовано Міністерством освіти і науки, молоді та спорту України як навчальний посібник для студентів вищих навчальних закладів. Гриф № 1/11-6462 від 21.07.2011 р.

**ISBN 978-966-8462-51-1**

**А.М. Ніколаєнко, Н.О. Міняйло, 2011**

## ЗМІСТ

|   |     |
|---|-----|
| <b>ПЕРЕДМОВА</b> .....  | 7   |
| <b>1 Загальні відомості про мікропроцесори та контролери</b> .....                            | 9   |
| 1.1 Етапи розвитку систем управління.....   | 9   |
| 1.2 Тенденції розвитку ПЛК.....   | 10  |
| 1.3 Нові технології у виробництві контролерів.....  | 12  |
| 1.4 Принципи побудови мікропроцесорних контролерів.....                                       | 14  |
| 1.5 Організація роботи контролера.....  | 19  |
| 1.6 Аналогово-цифрові та цифро-аналогові перетворювачі.....                                   | 21  |
| 1.7 Пристрої гальванічного розв'язання електричних кіл<br>контролера.....                     | 25  |
| <i>Тестові та контрольні запитання</i> .....  | 27  |
| <b>2 Основні характеристики та можливості контролерів</b> .....                               | 32  |
| 2.1 Характеристика центрального процесора.....  | 32  |
| 2.2 Характеристика каналів вводу-виводу.....  | 34  |
| 2.3 Конфігурація вводів-выводів.....  | 35  |
| 2.4 Комунікаційні можливості.....   | 37  |
| 2.5 Характеристика промислових мереж.....   | 40  |
| 2.6 Засоби програмування контролерів.....   | 47  |
| <i>Тестові та контрольні запитання</i> .....  | 49  |
| <b>3 Мікропроцесорний контролер Реміконт Р-130</b> .....                                      | 54  |
| 3.1 Загальна характеристика та фізична структура.....   | 54  |
| 3.2 Віртуальна структура Р-130.....   | 59  |
| 3.3 Мережна структура Р-130.....  | 63  |
| 3.4 Технологічне програмування контролера.....  | 66  |
| 3.5 Шляхи розвитку промислових контролерів Р-130.....   | 68  |
| 3.6 Програмування контролерів Р-130 Isa.....  | 68  |
| 3.6.1 Методика створення проекту у системі програмування<br>ISaGRAF.....                      | 69  |
| <i>Тестові та контрольні запитання</i> .....  | 74  |
| <b>4 Мікропроцесорний контролер МІК-51</b> .....  | 80  |
| 4.1 Загальна характеристика та фізична структура.....   | 80  |
| 4.2 Архітектура контролера.....   | 84  |
| 4.3 Мережна архітектура.....  | 90  |
| 4.4 Програмування, настройка і контроль.....  | 94  |
| 4.4.1 Методика програмування контролера за допомогою<br>інструментальної системи «Альфа»..... | 96  |
| <i>Тестові та контрольні запитання</i> .....  | 103 |
| <b>5 Програмовані контролери корпорації Advantech</b> .....                                   | 108 |
| 5.1 Мікропроцесорний контролер ADAM 5510.....   | 109 |
| 5.1.1 Оновлення контролера ADAM 5510.....   | 112 |
| 5.1.2 Програмування контролерів групи ADAM-5510.....  | 114 |
| 5.1.2.1 Методика створення проекту у середовищі   |     |

|   |     |
|---|-----|
| Borland Turbo C++ v.3.0.....  | 114 |
| 5.2 Мікропроцесорний контролер ADAM 5510KW.....                                 | 124 |
| 5.2.1 Програмування контролерів групи ADAM 5510KW.....                          | 127 |
| 5.2.1.1 Методика створення проекту у системі програмування<br>KW MULTIPROG..... | 128 |
| <i>Тестові та контрольні запитання.....</i>                                     | 140 |
| <b>6 Програмовані контролери корпорації WAGO.....</b>                           | 145 |
| 6.1 Будова WAGO I/O-System.....   | 145 |
| 6.2 Програмований контролер WAGO 750-841.....                                   | 148 |
| 6.3 Програмування контролерів WAGO.....   | 153 |
| 6.3.1 Методика створення проекту у системі програмування<br>CoDeSys.....        | 153 |
| <i>Тестові та контрольні запитання.....</i>                                     | 161 |
| <b>7 Програмовані логічні контролери SIMATIC.....</b>                           | 166 |
| 7.1 Програмовані контролери SIMATIC S7-300.....                                 | 166 |
| 7.1.1 Стійки контролера.....  | 167 |
| 7.1.2 Інтерфейсні модулі.....   | 168 |
| 7.1.3 Центральні процесори.....   | 169 |
| 7.1.4 Блоки живлення.....   | 172 |
| 7.1.5 Сигнальні модулі.....   | 173 |
| 7.1.6 Функціональні модулі.....   | 178 |
| 7.1.7 Система децентралізованої периферії SIMATIC ET 200.....                   | 179 |
| 7.1.8 Комунікації SIMATIC.....  | 180 |
| 7.1.9 Програмне резервування.....   | 182 |
| 7.2 Програмне забезпечення SIMATIC.....   | 183 |
| 7.2.1 Методика створення проекту у пакеті Step7.....                            | 183 |
| <i>Тестові та контрольні запитання.....</i>                                     | 190 |
| <b>8 Мікропроцесорні контролери MODICON TSX.....</b>                            | 195 |
| 8.1 Контролери Modicon TSX Quantum.....   | 195 |
| 8.1.1 Монтажні панелі.....  | 196 |
| 8.1.2 Центральні процесори.....   | 197 |
| 8.1.3 Джерела живлення.....   | 199 |
| 8.1.4 Модулі вводу-виводу.....  | 200 |
| 8.1.5 Конфігурації вводу-виводу.....  | 203 |
| 8.1.5.1 Конфігурація локального вводу-виводу.....                               | 204 |
| 8.1.5.2 Конфігурація віддаленого вводу-виводу (RIO).....                        | 204 |
| 8.1.5.3 Конфігурація розподіленого вводу-виводу(DIO).....                       | 206 |
| 8.1.6 Система гарячого резервування.....  | 209 |
| 8.1.7 Система управління Momentum.....  | 210 |
| 8.1.8 Мережні можливості.....   | 212 |
| 8.1.9 Програмування контролерів Modicon TSX Quantum.....                        | 214 |
| 8.1.9.1 Методика створення проекту у системі програмування<br>Concept.....      | 214 |
| 8.1.9.2 Методика створення проекту у системі програмування                      |     |

|   |            |
|---|------------|
| Unity PRO XL.....   | 222        |
| <i>Тестові та контрольні запитання.....</i>   | 227        |
| <b>9 Програмовані контролери корпорації OMRON.....</b>                                    | <b>233</b> |
| 9.1 Модульні програмовані контролери SYSMAC CS1.....                                      | 233        |
| 9.1.1 Базова конфігурація контролера.....   | 234        |
| 9.1.2 Модулі центрального процесора.....  | 236        |
| 9.1.3 Блоки живлення.....   | 239        |
| 9.1.4 Базові модулі вводу-виводу CS1.....   | 240        |
| 9.1.5 Спеціальні модулі CS1.....  | 241        |
| 9.1.6 Модулі шини центрального процесора.....   | 242        |
| 9.1.7 Розподіл адрес поміж модулями контролера.....                                       | 245        |
| 9.1.8 Пристрої програмування.....   | 247        |
| 9.2 Програмування контролерів OMRON.....  | 248        |
| 9.2.1 Методика створення проекту у системі програмування<br>CX-One.....                   | 249        |
| <i>Тестові та контрольні запитання.....</i>   | 256        |
| <b>10 Програмовані контролери корпорації Rockwell Automation.....</b>                     | <b>262</b> |
| 10.1 Контролери ControlLogix.....   | 262        |
| 10.1.1 Вибір контролерів.....   | 264        |
| 10.1.2 Шасі ControlLogix.....   | 264        |
| 10.1.3 Джерела живлення.....  | 266        |
| 10.1.4 Модулі вводу-виводу.....   | 266        |
| 10.1.5 Робота модулів вводу-виводу.....   | 270        |
| 10.1.6 Комунікаційні можливості.....  | 274        |
| 10.1.7 Резервування у ControlLogix.....   | 277        |
| 10.2 Програмування контролерів Rockwell Automation.....                                   | 279        |
| 10.2.1 Методика створення проекту у системі програмування<br>RSLogix 5000.....            | 279        |
| <i>Тестові та контрольні запитання.....</i>   | 290        |
| <b>11 Мікропроцесорні контролери виробництва GE Fanuc.....</b>                            | <b>296</b> |
| 11.1 Контролери PACSystems RX7i.....  | 296        |
| 11.1.1 Стійки PACSystems RX7i.....  | 298        |
| 11.1.2 Модулі центрального процесора.....   | 301        |
| 11.1.3 Блоки живлення.....  | 302        |
| 11.1.4 Модулі вводу-виводу.....   | 304        |
| 11.1.5 Засоби віддаленого вводу-виводу і комунікацій Genius і<br>Field Control.....       | 305        |
| 11.1.6 Гаряче резервування ПЛК.....   | 309        |
| 11.2 Програмування контролерів GE Fanuc.....  | 310        |
| 11.2.1 Методика створення проекту у системі програмування<br>Proficy Machine Edition..... | 311        |
| <i>Тестові та контрольні запитання.....</i>   | 317        |
| <b>12 Лабораторні роботи.....</b>   | <b>323</b> |
| 12.1 Лабораторна робота №1. Програмування та дослідження                                  |            |

|  |            |
|--|------------|
| роботи контролера Logo!Basic.....  | 323        |
| 12.2 Лабораторна робота №2. Програмування та дослідження<br>роботи контролера P-130.....         | 335        |
| 12.3 Лабораторна робота №3. Програмування та дослідження<br>роботи контролера MİK-51.....        | 346        |
| 12.4 Лабораторна робота №4. Програмування та дослідження<br>роботи контролера VIPA Spread 7..... | 358        |
| 12.5 Лабораторна робота №5. Програмування та дослідження<br>роботи контролера WAGO 750-841.....  | 367        |
| 12.6 Лабораторна робота №6. Програмування та дослідження<br>роботи контролера ADAM-5511.....     | 379        |
| 12.7 Лабораторна робота №7. Програмування та дослідження<br>роботи контролера ADAM-5510KW.....   | 385        |
| 12.8 Лабораторна робота №8. Програмування та дослідження<br>роботи контролера Durus.....         | 395        |
| <b>13 Курсова робота.....</b>  | <b>404</b> |
| 13.1 Перелік завдань для курсової роботи.....  | 404        |
| 13.2 Приклади виконання курсової роботи.....   | 406        |
| 13.2.1 Приклад 1.....  | 406        |
| 13.2.2 Приклад 2.....  | 419        |
| <b>Література.....</b>   | <b>434</b> |
| <b>Словник термінів і скорочень.....</b>   | <b>439</b> |

## ПЕРЕДМОВА

Наприкінці минулого і початку XXI – століття в промисловості України розпочалось активне впровадження розподілених АСУ ТП, в структурі яких під керівництвом спеціального програмного забезпечення працюють мікропроцесорні пристрої. На ринок автоматизації потрапила велика кількість програмованих логічних контролерів (ПЛК) всесвітньовідомих виробників, а на об'єктах автоматизації поряд з технічними засобами закордонних фірм все частіше використовується мікропроцесорна техніка українського походження.

У вищих навчальних закладах посилилась підготовка майбутніх спеціалістів по мікропроцесорним і програмним засобам автоматизації. Головним джерелом інформації для викладачів стали каталоги, технічні інструкції на прилади, часто англійською мовою, і статті у технічних журналах, а для студентів – лекції.

Незручність таких обставин спонукала авторів цього видання працювати над створенням відповідної навчальної літератури. У 2008 році була надрукована методична розробка, щодо вивчення стандартних мов і створення додатків користувача для ПЛК у середовищі KW MULTIPROG [1], а наразі пропонується навчальний посібник по мікропроцесорним і програмним засобам автоматизації.

На сторінках посібника розглянуто будова, організація роботи, основні технічні характеристики та можливості контролерів, а також найбільш поширені ПЛК світових і вітчизняних виробників, таких як Siemens, Schneider Electric, Advantech, Rockwell Automation, Omron, GE Fanuc, WAGO, ЗЕiМ і Мікрол. Додається методика створення проектів користувача за допомогою відповідних інструментальних пакетів програмування, а для отримання навичок роботи з контролерами – лабораторний практикум і курсова робота з переліком завдань і прикладами виконання.

Для перевірки знань наприкінці кожного розділу приведені тестові та контрольні запитання трьох рівнів складності з посиланнями на сторінки, де можна перевірити правильність відповідей.

Навчальний посібник «Мікропроцесорні та програмні засоби автоматизації» створений під однойменну дисципліну самостійного вибору студентів, яка має 7 кредитів ECTS загальним обсягом навантаження 252 години і викладається на четвертому курсі у двох напівсеместрах. При цьому використовуються знання, що отримані студентами при вивченні дисциплін «Мікропроцесорна техніка», «Метрологія та теплотехнічні вимірювання», «Технічні засоби автоматизації».

**Мета** дисципліни – навчити студентів орієнтуватися у програмованих логічних контролерах, що пропонуються світовими і вітчизняними виробниками; усвідомлено вибирати їх при розробці систем автоматичного управління і здійснювати проектне компонування; створювати додатки користувача за допомогою інструментальних середовищ програмування, працювати з контролерами і налагоджувати їх роботу.

**Завдання** дисципліни – ознайомити студентів з тенденціями розвитку систем управління та програмованих логічних контролерів, їх будовою і організацією роботи, з можливостями і основними технічними характеристиками, на підставі яких здійснюється вибір ПЛК; з поширеними в промисловості України контролерами світових і вітчизняних виробників, показати, що в них спільного і в чому їх відмінності; дати методику створення проектів у популярних інструментальних системах програмування сучасних мікропроцесорних контролерів і методику їх проектного компонування та налагодження роботи.

Після вивчення дисципліни студенти мають:

– **знати** будову і організацію роботи ПЛК, конструктивні особливості та функціональні можливості мікропроцесорних контролерів S7-300 (Siemens), TSX Quantum (Schneider Electric), SYSMAC CS1 (Omron), 750-841 (WAGO), ControlLogix (Rockwell Automation), PACSystems RX7i (GE Fanuc), ADAM 5510, ADAM 5510KW (Advantech), МІК-51(Мікрол), Ремиконт P-130; методику створення додатків користувача за допомогою інструментальних пакетів програмування Step 7, Concept, Multiprog, RSLogix 5000, Сх-Programmer, Proficy Machine Edition, CoDeSys, IsaGraf, WinPLC7, LOGO! Soft Comfort, Durus Development Software, Borland Turbo C++, Альфа; методику проектного компонування контролерів та налагодження їх роботи;

– **вміти** орієнтуватися на ринку промислових контролерів; вибирати необхідні ПЛК при розробці систем автоматичного управління і здійснювати підбір відповідних модулів; працювати з відомими інструментальними середовищами програмування, створювати за їх допомогою додатки користувача для ПЛК; працювати з контролерами та налагоджувати їх роботу.

Знання та навички, що отримують студенти протягом вивчення «Мікропроцесорних та програмних засобів автоматизації», необхідні при викладанні дисциплін «Монтаж, налагодження та експлуатація систем автоматизації», «Автоматизація технологічних процесів», «Програмно-технічні комплекси систем автоматизації», а також при підготовці дипломного проекту і у подальшій професійній діяльності.

Автори вдячні студентам спеціальності «Автоматизоване управління технологічними процесами» Запорізької державної інженерної академії, які приймали участь в освоєванні інструментальних середовищ програмування мікропроцесорних контролерів, створюванні стендів і методичних вказівок до лабораторних робіт.

Висловлюємо також подяку рецензентам посібника за корисні поради та пропозиції.

## **1 ЗАГАЛЬНІ ВІДОМОСТІ ПРО МІКРОПРОЦЕСОРНІ КОНТРОЛЕРИ**



## 1.1 Етапи розвитку систем управління

Історія виникнення і розвитку сучасних систем автоматизації відноситься до кінця 70-х - початку 80-х років ХХ століття і безпосередньо пов'язана з появою мікропроцесорів, що спричинило революційні зміни у галузі комп'ютерної техніки та технологіях розвитку засобів промислової автоматики.

Мікропроцесорні системи автоматизації прийшли на зміну автоматизованим системам управління технологічними процесами (АСУ ТП), які впроваджувалися на порозі 80-х років ХХ століття і будувалися на базі потужних (для тих часів) промислових ЕОМ, що централізовано керували складними неперервними технологічними процесами. Їх завданням було збирання й обробка сигналів від датчиків, підготовка та подання інформації оператору, накопичення і документування архівних даних, формулювання рекомендацій та реалізація керуючих дій і т.ін. Основним недоліком таких систем була їх надмірна централізація і надлишковість, що стримувало їх впровадження. Саме цим пояснюється той факт, що процеси, які були пов'язані із впровадженням АСУ ТП і активно розпочались на початку 70-х років, поступово стали уповільнювати свій розвиток.

Поява мікропроцесорів сприяла бурхливому розвитку багатопроцесорних і багатомашинних керуючих комплексів, які базувалися на нових принципах розподіленого керування, системних інтерфейсах і мережних технологіях. Нові системи, які отримали назву розподілених систем управління (Distributed Control System – DCS), розвивались у напрямі "від загального до часткового", тобто від постановки загальної інтегральної задачі керування технологічним процесом на всіх її рівнях до створення єдиного розподіленого програмно-технічного середовища для її розв'язання. Згідно з цією технологією задачі оперативного управління, збору даних і безпосереднього цифрового управління поділяються між декількома контролерами. При цьому виділяються два рівні – верхній (операторське управління, архівування, документування) і нижній (збір даних і безпосереднє цифрове управління). На кожному рівні використовується різна кількість комп'ютерів залежно від складності технологічного процесу і вимог до надійності та швидкодії.

Комп'ютери верхнього рівня отримали назву АРМ (Автоматизоване Робоче Місце), а нижнього – ПЛК (Програмовані Логічні Контролери), англійською PLC (Programmable Logic Controller). Спочатку на обох рівнях ніякої стандартизації в області апаратної платформи не було. Як правило, один виробник створював комплексні системи управління, в яких були і АРМ і ПЛК. Причому навіть в межах однієї системи DCS апаратні рішення для контролерів і АРМ були абсолютно різні і мали власні інструментальні середовища для програмування.

Для обміну даними між верхнім і нижнім рівнями кожний виробник DCS пропонував власні рішення – польові шини, що мали фізичну апаратуру і логічні протоколи передачі даних. Із-за відсутності стандартів на апаратуру та протоколи передачі даних неможливо було використовувати окремі компоненти

однієї DCS разом з компонентами іншої. Тому користувач був прив'язаний до одного виробника засобів управління.

Перші DCS були впроваджені на виробництвах зі складними неперервними технологічними циклами у хімічній і нафтопереробній промисловості, де і була потреба у використанні великих інтегрованих систем. Провідними фірмами в галузі розробки та впровадження DCS є Honeywell, Fisher-Rosemount, Westinghouse. В Україні прикладом створення таких систем став програмно-технічний комплекс Уніконт.

З подальшим розвитком технології офісної автоматизації більш надійніша і дешевша PC-платформа, а також існуюча стандартизація програмних засобів на рівні операційних систем та засобів комунікації практично витиснули з ринку інші рішення. Це сильно відбилося на архітектурі апаратних комплексів промислових засобів управління і більшість виробників перевели свої АРМ на PC-платформу.

Після того як PC-архітектура стала практично стандартом для верхнього рівня систем управління, з'явилася значна кількість виробників ПЛК, зорієнтованих на використання в АСУ ТП звичайних або промислових PC у якості АРМ. При цьому з контролерами поставлялися системи їх програмування і відкривалися протоколи для обміну даними. Це призвело до появи програм для розробки інтерфейсу користувача, які обмінювалися даними з контролерами через драйвери. Такі програми отримали назву SCADA (Supervisory Control AND Data Acquisition – Супервізорне керування та отримання даних). Вони відрізнялись можливостями в області розробки графічного інтерфейсу, математичній обробці, архівування та документування отриманої від ПЛК інформації. Але всі працювали за одною схемою – запрошували через драйвер дані у ПЛК і в той чи іншій формі представляли їх оператору.

Після розповсюдження PC-архітектури на рівні АРМ, розвиток автоматизованих систем управління пішов шляхом використання її у програмованих контролерах. Така уніфікація апаратних засобів на різних рівнях системи дозволяла суттєво скоротити витрати на їх інтеграцію і мала багато інших переваг. Проте розповсюдження PC-архітектури на рівні ПЛК йшло не так швидко, як в області АРМ. Щоб зрозуміти це необхідно проаналізувати розвиток ПЛК і те, які фактори сприяли, а які перешкоджали використанню в них PC-архітектури [2,3,4].

## **1.2 Тенденції розвитку ПЛК**

Поява ПЛК пов'язана насамперед з автоматизацією дискретних процесів і потребою заміни традиційних систем керування, побудованих на базі релейно-контактних або безконтактних логічних схем керування, які працювали за жорсткою логікою. На першому етапі ПЛК практично один до одного замінили ці схеми, але із суттєвою перевагою – можливістю зміни алгоритму керування шляхом перепрограмування. Звідси і назва – програмований логічний контролер. Перший ПЛК, який мав назву MoDiCon (Modular Digital Controller)

був використаний у 1968 році в автомобільній промисловості США саме для заміни шаф з релейною логікою. Але дуже швидко побачивши і зрозумівши переваги, які мають системи автоматизації, що побудовані на базі ПЛК, розробники цієї техніки почали розширювати функціональні та технічні можливості ПЛК. Сучасні ПЛК мають велику різноманітність модулів вводу-виводу, у тому числі дискретних, аналогових, для керування кроковими двигунами, вимірювання ваги та ін. Вони мають у складі свого програмного забезпечення алгоритми аналогової обробки сигналів і задач неперервного регулювання. Використання різноманітних мереж і польових шин, архітектури клієнт-сервер, нових технологій Microsoft (OPC, COM, DCOM, Active-X, Web-client), спеціальних SCADA-програм для створення операторського інтерфейсу АРМ перетворили побудовані на базі ПЛК АСУ ТП, у потужні розподілені системи, які конкурують із системами DCS у галузі автоматизації неперервних технологічних процесів.

Але при цьому ПЛК вдалося зберегти особливості, які на початку розвитку зумовили їх величезну популярність серед користувачів. Це насамперед зручності програмування та обслуговування. Наявність, так званих, "технологічних" мов програмування значно спростило процедуру реалізації і оперативної зміни алгоритмів керування програмним шляхом безпосередньо кінцевим користувачем.

Як недолік, ці ПЛК мають закриту архітектуру і досить велику ціну. Тому у боротьбі за клієнта виробники контролерів намагалися знизити ціни на свої вироби за рахунок підвищення продажу своїх контролерів і використання при їх виробництві комплектуючих, що випускалися великими тиражами і мали низьку ціну. Проте перший шлях вимагав дуже великих витрат на рекламу, а на другому шляху виробник неминуче використовував ті ж самі комплектуючі, що і у звичайних ЕОМ. У зв'язку з цим більшість виробників, які давно працювали на ринку ПЛК, стали поступово переходити на РС-платформу, а нові виробники засобів автоматизації одразу зорієнтувались на цю платформу.

Ця обставина дала можливість виробникам контролерів максимально зосередити свої зусилля на розробці більш широкої номенклатури модулів ПЗО і на архітектурних рішеннях, що підвищували надійність ПЛК. Тому на ринку з кожним роком з'являлося все більше контролерів з РС-архітектурою. Такі контролери отримали назву РС-сумісні контролери або SoftПЛК. Ця назва відображає той факт, що більшість функцій звичайних ПЛК, що раніше вирішувались на апаратному рівні, в цих контролерах розв'язується за допомогою програмного забезпечення (ПЗ). Проте, наявність вже відпрацьованої технології та діюче виробництво існуючих ПЛК, велика кількість користувачів, які вміють працювати зі старими контролерами, наявність в штаті виробників великої кількості співробітників, що вирішують проблеми, яких не буде при переході на РС-платформу, а також затрачені кошти на рекламу, суттєво стримують виробників контролерів перейти на РС-архітектуру.

З появою PC-сумісних контролерів тисячі користувачів персональних ЕОМ отримали можливість використати свої знання і досвід роботи у новій для них галузі. Проте залучення професійних програмістів ускладнило розробку, супроводження і модернізацію систем автоматизації.

Враховуючі те, що все-таки спеціалісти не в галузі комп'ютерних технологій, а в галузі автоматизації повинні бути розробниками сучасних АСУ ТП незалежні розробники ПЗ почали створювати інструментальні засоби програмування контролерів. Ядро реального часу таких програм було незалежним від марки контролера, а лише змінювалися драйвери для доступу до плат пристрою зв'язку з об'єктом (ПЗО). Ці програми базувалися на вимогах міжнародного стандарту IEC 61131-3 і отримали назву Softlogic. Завдяки їм при розробці прикладного програмного забезпечення сучасних ПЛК, багато задач розв'язується не на рівні програмування, а на рівні простого конфігурування.

Таким чином, перехід на PC-платформу став основною тенденцією розвитку засобів автоматизації. Головною причиною цього є бажання заказників використовувати стандартні рішення, які знижують вартість виробів, підвищують їх надійність і дозволяють легко інтегрувати їх у великі програмно апаратні комплекси.

Крім того, розвитку засобів автоматизації сприяло відокремлення задач створення ПЗ від виробництва апаратних засобів. ПЗ одного незалежного виробника може використовуватися на апаратурі різних виробників технічних засобів[3,4,5].

### **1.3 Нові технології у виробництві контролерів**

Значною перевагою PC-сумісних контролерів є те, що вони мають відкриту архітектуру, яка базується на одному з магістрально-модульних стандартів. При відкритій магістрально-модульній архітектурі центральний процесор сполучається з іншими модулями контролера через стандартний інтерфейс. Це надає можливість різним виробникам випускати у даному стандарті будь-які модулі, а розробникам контролерів – компонувати свої ПЛК модулями різних фірм. Широка кооперація фірм, що підтримують даний стандарт на шину і працюють у цьому стандарті, дозволяє користувачам модулів не орієнтуватися на одного постачальника, а мати широкий вибір необхідної продукції.

Зазвичай для задоволення потреб усіх споживачів, виробник випускає велику кількість модифікацій модулів стандартної моделі. Проте це не дозволяє йому істотно знизити вартість продукції. У той же час ринок вимагає функціонально гнучких контролерів, які забезпечували б можливість недорогого розширення або зміни їх конфігурації і функцій.

У кінці 90-х років виробники запропонували моделі контролерів, рівень функціональності та інтеграції яких може змінюватися. Досягається це застосуванням в контролерах з магістрально-модульною архітектурою наплатних мезонін-модулів, які можуть виконувати роль модулів вводу-виводу, інтерфейсів, пам'яті і т.ін. Мезонінні модулі – це додаткові плати, що

установлюються паралельно на основні для розширення їх функціональних можливостей. Використання наплатних мезонін-модулів надає можливість збирати з готових модулів контролери, структура і характеристики яких можуть відповідати усім вимогам автоматизації конкретного об'єкту. І, враховуючи просту заміну окремої плати в ході експлуатації, завжди залишається можливість переконфігурувати контролер відповідно до нових задач автоматизації і змінних властивостей об'єкту. Такий підхід дозволяє досягти максимально високого рівня інтеграції контролера при відносно малих витратах.

Останніми роками при виробництві контролерів широко розвиваються стандарти магістрально-модульної архітектури, такі як PCI, CompactPCI, VME. За найпоширенішим в Європі стандартом VME вже сотні фірм випускають різні типи плат. Серед них - центральні процесори, мережні контролери, блоки пам'яті, модулі вводу-виводу і т.ін.

VME - це 32 або 64 розрядна шина з максимальною пропускнуною спроможністю 40 Мбіт/с. Вона розроблена фірмою Motorola і згодом була стандартизована ІЕС, як ANSI/IEEE 1014-1987.

Контролери на базі шини VME виготовляються в конструктивах стандарту Євромеханіки форматів 3U і 6U. Базовим параметром цього стандарту є розмір 19 дюймів (482,6 мм) — це ширина передньої панелі субблока (монтажного каркаса, або крейта), в який вставляються печатні плати. Тому стандарт Євромеханіки часто кваліфікують як 19" стандарт МЭК-60297. Відповідно до цього стандарту висота монтажного каркасу визначається в одиницях U, яка дорівнює 1,75 дюйми, або 44,45 мм. Розробникам контролерів рекомендуються до впровадження субблока під плати висотою 3U і 6U.

Стандарти магістрально-модульної архітектури дозволяють збирати контролери з готових плат різних фірм, що випускають продукцію за цими стандартами. Збірка контролерів при такому підході зводиться до розміщення стандартної плати формату "Evrocard" в конструктиви (стійкі, шафи) такого ж стандарту. І конструктиви, і плати мають широкий діапазон виконання, який задовольняє різні умови експлуатації (температуру, вогкість, вібрацію, пил, електромагнітні перешкоди і т. д.).

Системи, що побудовані на базі VME-шини, набули широке поширення внаслідок великої продуктивності і можливості пристосовуватися до змінних умов експлуатації. В даний час ринок промислових систем розвивається у напрямі використання VME-плат/носіїв зі встановленими на них мезонінами.

Перспективність такого підходу безперечна. Проте висока вартість VME-контролерів (на 20-40% дорожче в порівнянні із звичними контролерами, що мають аналогічні характеристики) в даний час стримує процес їх розповсюдження. Серед фірм, що використовують магістрально-модульну архітектуру VME при виробництві контролерів, знаходяться такі відомі виробники, як GE Fanuc (сімейство контролерів 90-70, PACSystem RX7i), PER Modular Computers та ін..

Останніми роками розпочався випуск нових пристроїв промислової автоматики – програмованих автоматизаційних контролерів РАС (Programmable Automation Controller). РАС-пристрої мають архітектуру і операційну систему персональних комп'ютерів, потужні процесори, великий об'єм пам'яті але як і ПЛК, призначені для розв'язання завдань безпосереднього цифрового управління.

Відповідно до концепції РАС, програмовані автоматизаційні контролери мають багатозадачні функціональні можливості управління на єдиній платформі; єдине комплексне середовище розробки, що включає спільне програмне забезпечення і спільну базу даних змінних процесу; програмні засоби, які дозволяють розробляти додатки для технологічних ліній у взаємозв'язку з окремими машинами або технологічними процесами; відкриту модульну архітектуру, яка сприяє розробці комплексних додатків – від польового рівня до рівня управління виробництвом; підтримку існуючих стандартів для мережних інтерфейсів, мов програмування та ін., що дозволяє інтегрувати технічні рішення у системи інших виробників [6,7,8,9].

#### **1.4 Принципи побудови мікропроцесорних контролерів**

Програмований логічний контролер – це спеціалізований мікропроцесорний пристрій, який призначений для керування виробничими процесами в умовах промислового середовища у реальному масштабі часу і, програмування якого посильне неспеціалісту в галузі інформатики.

Центральною частиною контролерів є мікропроцесори, які можуть істотно розрізнятися між собою набором функціональних блоків і зв'язками між ними. Проте, у структурі кожного процесора можна виділити основні елементи, що визначають його специфіку, як керуючого центру ПЛК. Серед них, насамперед, пристрій управління та операційний пристрій.

Пристрій управління (ПУ) призначений для реалізації вибирання команд, їх дешифрування і на основі цього керування обміном та обробкою інформації шляхом створення керуючих сигналів.

Операційний пристрій (ОП) служить для обробки цифрової інформації (арифметичні та логічні операції, зсуви, аналіз чисел та ін.).

Узагальнена структура мікропроцесора приведена на рис 1.1.

Основними елементами для збереження інформації в середині мікропроцесора є регістри, які виконують функцію надоперативного запам'ятовуючого пристрою (ОЗП) з мінімальним часом запису та зчитування.

Регістр команд (РгК) використовується для фіксування коду команди після зчитування її з пам'яті. Як правило, у цьому регістрі фіксується лише код операції – частина коду команди, яка визначає виконувані дії і спосіб адресації операндів.

Регістри операндів призначені для зберігання даних у процесі їх обробки. Кількість регістрів операндів у сучасних процесорах може досягати 10-15 штук. По суті вони утворюють внутрішню пам'ять процесора. Дані, що були розміщені у регістрах операндів надходять на обробку в арифметично-логічний

пристрій (АЛП). У деяких типів процесорів один з регістрів операнда завжди є і приймачем результату операції в АЛП. Такий регістр називають регістром-акумулятором.

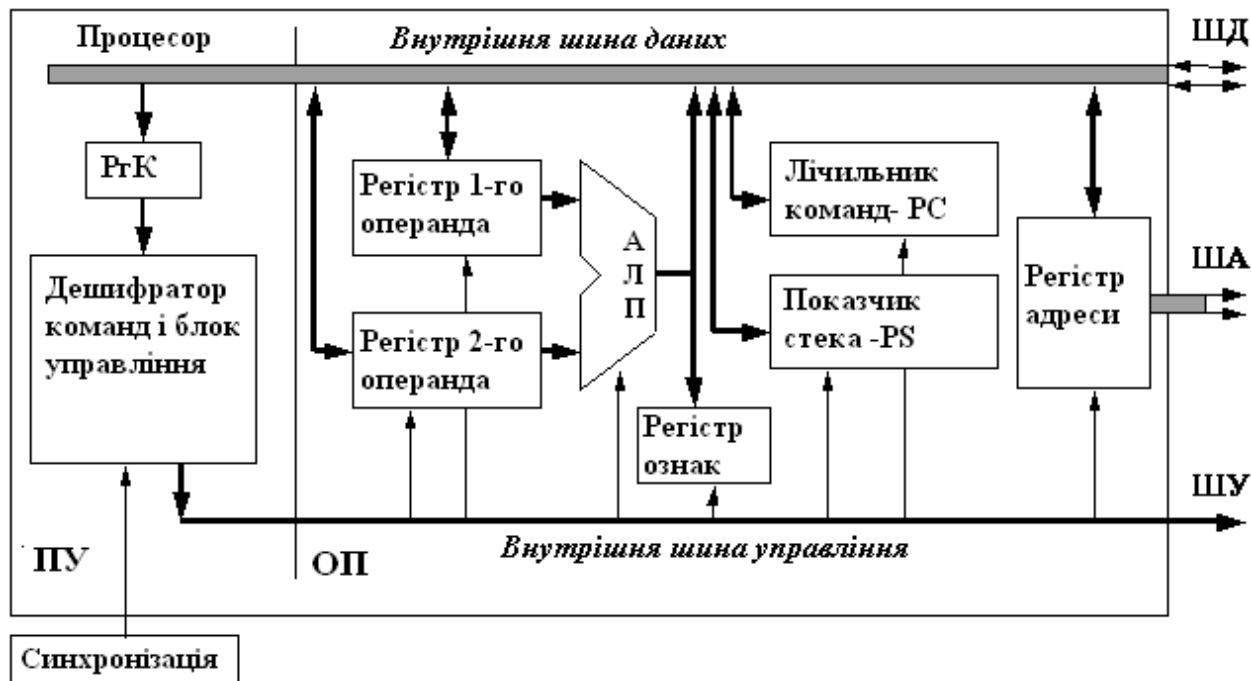


Рисунок 1.1 – Структурна схема універсального мікропроцесора

Лічильник команд містить адресу комірки пам'яті, в якій знаходяться байти виконуваної команди.

Показчик стека зберігає адресу останньої зайнятої комірки стека, яку називають вершиною. Число, що знаходиться в показчику, вказує, де знаходиться вершина стека. Коли в стек записується чергове слово, число, що в показчику стека, збільшується на одиницю. При добуванні слова із стека число, що заповнює його показчик, навпаки зменшується на одиницю.

Регістром ознак є набір тригерів – прапорців. Залежно від результатів операцій, виконуваних АЛП, кожен тригер встановлюється в стан 0 або 1. Біти прапорців, що визначають вміст регістра, відображають умовні ознаки: нульового результату, знака результату, переповнення і тому подібне. Ця інформація, важлива для вибору подальшого шляху обчислень.

Для сучасних контролерів характерна магістральна (шинна) фізична організація роботи, при якій обмін інформацією між блоками виконується в паралельному коді по магістралях, які називаються шинами. Шина – це кілька паралельних провідників із загальним функціональним призначенням.

У типовій структурі контролера, як правило реалізовані три інформаційні магістралі:

- шина адреси (ША) – використовується для передачі інформації, яка вказує пристрій або адресу комірки пам'яті, до яких здійснюється звернення з боку процесора;

- шина даних (ШД) – служить для передачі інформації (кодів команд або даних) між пристроями контролера;

- шина управління (ШУ)- набір сигналів, що визначають, які пристрої обмінюються даними, у якому напрямку (у процесор або з процесора) і, в якому режимі, а також виконують синхронізацію роботи мікропроцесора і інших блоків у складі контролера.

Усі пристрої контролера підключені до одних і тих самих шин адреси і даних. В кожний момент часу тільки два пристрої можуть обмінюватися інформацією по шині даних, а по шині адреси тільки один з пристроїв, що бере участь в обміні (частіше всього процесор), видає адреси абонента. Інші пристрої у складі системи відключені від шини даних.

В узагальненому вигляді функціонування мікропроцесора може бути подане, як циклічна послідовність двох етапів – вибирання (читання) команд з пам'яті та їх дешифрування і виконання команд.

Вибирання команд є автоматичним процесом, який відбувається під впливом імпульсів від генератора тактових імпульсів і не залежить від програміста.

Дешифрування команди – це процес формування послідовності керуючих сигналів для всіх вузлів процесора і інших блоків контролера на основі коду, що міститься в команді.

Дії, які виконуються відповідно до команди, являють собою арифметичну або логічну обробку даних визначених програмістом. Після виконання дії процесор автоматично переходить до вибору з пам'яті наступної команди.

Мікропроцесор, незважаючи на його універсальність, самостійно нездатний реалізувати конкретну задачу обробки інформації або керування об'єктом. Для виконання цих функцій необхідно його з'єднати з деякими пристроями, запрограмувати та забезпечити обмін інформації мікропроцесора з цими пристроями, тобто утворити мікропроцесорну систему, яка є суттю будь-якого ПЛК.

Узагальнена структура програмованого логічного контролера приведена на рис.1.2.

Програмований контролер окрім мікропроцесора (МП) додатково має оперативний запам'ятовуючий пристрій (ОЗП), постійний запам'ятовуючий пристрій (ПЗП), вхідні та вихідні модулі, що перетворюють аналогові, дискретні, вхідні сигнали у цифрову форму і навпаки, комунікаційні модулі для створення промислових мереж та ряд інших пристроїв, які призначені для виконання деяких чітко окреслених функцій з обробки цифрової інформації і керування процесом цієї обробки.

Оперативний запам'ятовуючий пристрій є основним видом пам'яті контролера. В ОЗП зберігається програма обробки інформації та дані. Пристрій керування мікропроцесора постійно звертається до пам'яті та крок за кроком виконує команди, які записані в ній.

Ємність пам'яті характеризують порціями, які кратні 1024 словам, або байтам. Скорочено ємність такої порції позначають 1 Кбайт.





Рисунок 1.2 – Узагальнена структура програмованого логічного контролера

Розрядність вибірки визначається кількістю розрядів інформації, яка записується до ОЗП чи береться з нього за одне звернення. Оперативну пам'ять називають також пам'яттю з довільним доступом, звідки і з'явилась абревіатура RAM (Random Acces Memory), в яку в довільний момент часу можна записати інформацію та у довільний час прочитати дані з неї. Незважаючи на велику швидкість і гнучкість у роботі, ОЗП має один суттєвий недолік - при вимкненні живлення дані, які записані в ОЗП, втрачаються. У деяких випадках паралельно живленню ОЗП приєднують батарейку (або акумулятор), яка підтримує працездатність ОЗП деякий час при вимкненні живлення. Такі ОЗП називають енергонезалежними.

Повністю уникнути втрати інформації при вимкненні живлення дає можливість постійний запам'ятовуючий пристрій. На початку розвитку мікропроцесорної техніки вважалось, що ОЗП призначене тільки для тимчасового зберігання інформації, а дані з ПЗП у процесі експлуатації можна тільки зчитувати.

Дані до ПЗП записують тільки один раз або при його виготовленні у виробника ПЗП на базі маски (картки) замовника, або безпосередньо користувачем.

Останнім часом значного поширення набули такі види ПЗП, які називають перепрограмованими постійними запам'ятовуючими пристроями (ППЗП). Вони допускають багаторазовий запис даних. Залежно від типу ППЗП час зберігання в них може бути від кількох тисяч годин до десятків років.

Інформація, яка призначена для обробки у мікропроцесорі, може бути як дискретною, так і аналоговою. Проте в обох випадках вона подається у МП однаково - у вигляді двійкових чисел. Тому для автоматичного збору

інформації про стан об'єкта керування, його обробки та формування керуючої дії, до складу ПЛК входить пристрій зв'язку з об'єктом керування (ПЗО), у складі якого є відповідні модулі вводу – аналогово-цифрові (АЦП), дискретно-цифрові (ДЦП), та виводу - цифро аналогові (ЦАП) і цифро дискретні (ЦДП). Завданням ПЗО є автоматичне перетворення сигналів від датчиків, встановлених на об'єктах керування у форму, необхідну для вводу у МП, а також перетворення керуючих сигналів, що надходять з МП, у форму, яка може сприйматися виконавчими механізмами для керування регулюючими органами, безпосередньо встановленими на об'єкті керування.

Модулі дискретних входів здійснюють перетворення дискретних сигналів в цифрові. Один дискретний вхід ПЛК здатний приймати один бінарний електричний сигнал, який характеризується двома станами – увімкнений або вимкнений. На рівні програми це один біт інформації – ІСТИНА або ХИБНІСТЬ.

Кнопки, вимикачі, контакти реле, датчики наявності предметів і багато приладів з виходом типу «сухий контакт» або «відкритий колектор» безпосередньо можуть бути підключені до дискретних входів ПЛК.

В модулях аналогового входу електричний сигнал відображає рівень напруги або струму пропорційний фізичній величині, яка вимірюється в кожний момент часу. Це може бути температура, тиск, вага, швидкість, частота і т.ін.

Оскільки ПЛК є цифровою обчислювальною машиною, аналогові вхідні сигнали обов'язково піддаються аналого-цифровому перетворенню. В результаті, утворюється дискретна змінна певної розрядності. Як правило, в ПЛК застосовуються 8-12 розрядні перетворювачі. АЦП вищої розрядності не виправдовують себе, в першу чергу через високий рівень індустріальних перешкод, характерних для умов роботи контролерів.

В модулях дискретних виходів відбувається перетворення цифрового управляючого сигналу в дискретний. Один дискретний вихід ПЛК здатний комутувати один електричний сигнал. Дискретний вихід, як і дискретний вхід, з погляду програми це один біт інформації, приймаючий стани ІСТИНА або ХИБНІСТЬ.

Навантаженням дискретних виходів можуть бути лампи, реле, соленоїди, силові пускачі, пневматичні клапани, індикатори і т.ін. Простий дискретний вихід ПЛК виконується у вигляді контактів реле. Такий вихід достатньо простий і зручний в застосуванні. Проте він має характерні недоліки реле – обмежений ресурс, низьку швидкодію, руйнування контактів при роботі на індуктивне навантаження. Альтернативним рішенням дискретного виходу є електронний силовий елемент. Схема ключа обов'язково містить гальванічну розв'язку, елементи захисту від помилкового вмикання навантаження і короткого замикання, а також часто індивідуальну світлодіодну індикацію.

У модулях аналогового виходу здійснюється перетворення цифрового сигналу, сформованого центральним процесором за програмою користувача, в аналоговий сигнал управління. Навантаженням аналогових вихідних модулів

можуть бути будь-які підсилювальні блоки тиристорів і симисторів, виконавчі пристрої пропорційного типу та ін.

Окрім сполучення з датчиками та виконавчими пристроями, модулі ПЗО можуть забезпечувати зв'язок з пристроями відображення інформації або з іншими мікропроцесорними автоматичними системами або ЕОМ, користуючись промисловими мережами.

Інформація як між окремими пристроями ПЛК, так і при обміні із зовнішніми (периферійними) пристроями передається по загальних інформаційних каналах - шинах.

Периферійні пристрої не мають безпосереднього контакту із системними шинами. Замість цього вони приєднані до спеціальних регістрів, які використовуються для тимчасового зберігання бінарної інформації і називаються портами. Кожний порт має свою адресу (по аналогії з комірками пристроїв пам'яті), причому в кожний момент часу до шини даних може бути підключений тільки один порт. Цим керує спеціальний контролер вводу-виводу.

У простих випадках зовнішні пристрої приєднуються до ПЛК за допомогою портів. При інтенсивному обміні інформацією використовується більш складна система спряження, яка має назву інтерфейс. Інтерфейс - це комплекс апаратних та алгоритмічних засобів уніфікованого спряження компонентів мікропроцесорної системи. До складу апаратних засобів входять система уніфікованих шин, уніфікованих сигналів і електричних схем. Алгоритмічна частина - це алгоритм (чи, як його ще називають, протокол) обміну, тобто сукупність правил взаємозв'язку цих компонентів у процесі обміну інформацією. Найбільш поширені інтерфейси паралельні - ІРПР, в яких інформація передається по паралельних, і послідовні - ІРПС, в яких інформація передається послідовно по лініях зв'язку обмеженої кількості[3,10,11].

### **1.5 Організація роботи контролера**

У зв'язку з тим що основною задачею запрограмованого мікропроцесорного контролера є керування виконавчими механізмами на підставі сигналів, які характеризують стан об'єкта автоматизації, контролер повинен весь час стежити за значеннями вхідних сигналів і виробляти керуючі дії відповідно до алгоритму керування. Тому процес роботи ПЛК складається з циклічної послідовності виконання кількох етапів, які становлять робочий цикл контролера. До них належать:

- читання входів, у процесі якого в оперативну пам'ять контролера записується інформація про стан фізичних входів;
- виконання прикладної програми, підготовленої користувачем відповідно до заданого алгоритму керування;
- відновлення виходів, у процесі якого вихідним каналам модулів встановлюються значення, отримані в результаті обробки прикладної програми. Окрім того, в процесі кожного робочого циклу виконуються процедури, які пов'язані не стільки з обробкою прикладної програми, як із задачами аналізу

стану апаратних засобів, самодіагностики, обробки запитів, які прийшли з пульта ПЛК або по мережі і т. ін. Тобто відбувається внутрішня обробка, яка записується у системну пам'ять контролера.

Прикладна програма має справу з одномоментною копією значень входів. У середині одного циклу виконання програми, значення входів можна вважати константами. Така модель спрощує аналіз і програмування складних логічних і послідовних алгоритмів. Очевидно, що час реакції на подію залежатиме від часу виконання однієї ітерації прикладної програми.

У технічних характеристиках ПЛК надається типовий час робочого циклу, який у сучасних контролерах вимірюється одиницями мілісекунд і менше. Події, що вимагають швидкої реакції, виділяються в окремі задачі, пріоритетність і період виконання яких можна змінювати.

При роботі контролера можливі дві форми організації робочого циклу – циклічне або періодичне виконання.

Для циклічного виконання характерно послідовне, один за одним, виконання етапів прикладної програми. Сумарне виконання окремих етапів програми визначає час робочого циклу.

У зв'язку з тим що в процесі виконання прикладної програми час виконання окремих її етапів може змінюватись, змінюватиметься і тривалість робочого циклу ПЛК.

За періодичного режиму виконання робочого циклу його тривалість задається при програмуванні контролера. У цьому разі при виконанні програми ПЛК також послідовно виконує всі етапи обробки програми. Проте, якщо останній етап робочого циклу виконався, а заданий час періоду ще не вичерпався, процесор заповнює час операційного циклу, що залишився, системними чи фоновими задачами доти, поки не закінчиться період, і тільки після цього почнеться новий робочий цикл.

Основним призначенням систем самодіагностики і тестування ПЛК є забезпечення оперативного виявлення несправностей апаратних засобів контролера, а також програмних помилок, які заважають подальшому виконанню програми користувача або роблять його неможливим. Процедури самодіагностики виконуються автоматично і не залежать від прикладної програми. Розвитку і функціональним можливостям цих систем розробники ПЛК приділяють велику увагу. Це пояснюється тим, що у разі побудови системи автоматизації на базі локальних засобів автоматизації показники надійності цієї системи рівномірно розподілені між окремими технічними засобами. У разі відмови одного з технічних засобів автоматизації інші контури керування зберігають свою роботоздатність. Якщо ж система автоматизації будується на базі ПЛК, її надійність в основному визначається надійністю роботи саме ПЛК, тому що в ньому зосереджені основні ресурси керування об'єктом. Безумовно, що у разі відмови датчиків або виконавчих механізмів нормальна робота контурів керування, пов'язаних з ними, також буде порушена, але якщо вийде з ладу ПЛК, то може втратити свою роботоздатність система в цілому. Тому своєчасне виявлення помилок у роботі ПЛК є дуже важливим. Це

дає можливість не тільки оперативно виявити відмови ПЛК, а й вжити заходи для їх усунення. Цьому сприяє також блоково-модульний принцип побудови ПЛК. Вважається, що час відновлення роботоздатності ПЛК не повинен перевищувати 1 години.

Розрізняють два типи порушення роботоздатності контролерів – "помилка" і "відмова". Помилка - це часткова втрата роботоздатності ПЛК, яка не порушує виконання більшої частини алгоритму керування (відмова окремих каналів модулів, вихід значень вхідних сигналів за допустимі межі і т.ін.). Відмова - це порушення роботоздатності контролера, яке не дає можливості подальшого виконання алгоритму керування. При виявленні "відмови" ПЛК має припинити виконання прикладної програми. Розрізняють апаратні і програмні засоби самодіагностики. Апаратні засоби виконуються, як правило, у вигляді світлових індикаторів, які розташовуються на лицевих панелях модулів. Їх стан дає можливість оператору візуально визначити роботоздатність ПЛК і окремих його модулів, а також визначити момент порушення в його роботі.

Програмна самодіагностика виконується в процесі кожного робочого циклу ПЛК. Її результати записуються у пам'ять контролера у відповідні системні слова. До цих слів можна звертатися в процесі виконання програми користувача і програмним шляхом реалізовувати алгоритми реагування системи на виявлення відповідних помилок. Крім того, ця інформація може використовуватись для аналізу причин відмов ПЛК.

Детальніше виявити причини порушення роботоздатності ПЛК дають змогу системи тестування і діагностики ПЛК, які запускаються з терміналів програмування або з пультів програмування. Ці системи дають можливість швидко визначити причини виникнення помилок і відмов контролера. У деяких випадках для відновлення роботоздатності ПЛК необхідно замінити модулі, а у деяких – відновити або внести зміни у програму користувача[3].

### **1.6 Аналогово-цифрові та цифро-аналогові перетворювачі**

Аналогово-цифрові перетворювачі перетворюють вхідні дані у аналоговій формі (звичайно у вигляді значень струму або напруги) на цифрову форму. Процес перетворення містить у загальному випадку дискретизацію за часом, квантування за рівнем і кодування безперервної вхідної величини. Необхідна швидкість АЦП визначається в основному швидкістю зміни вхідної величини.

Цифро-аналогові перетворювачі реалізують зворотний процес перетворення цифрових даних на аналогові сигнали. Їх будова ґрунтується на генерації у відповідних колах напруг або струмів, пропорційних вагам розрядів.

ЦАП є одним з елементів АЦП, тому спочатку треба розглянути принцип цифро-аналогового перетворення.

Керувальний вплив, сформований у мікропроцесорі, подається у цифрових кодах. Виконавчі механізми (клапан, затулка, електричний двигун) здійснюють повільне регулювання параметрів об'єкту (витрати, тиску, рівня, температури) за допомогою аналогового сигналу. При дискретизації за часом керувальні дії надходять з мікропроцесора до ЦАП у вигляді цифрового коду

один раз за період квантування. Тому в ЦАП передбачено регістр, що запам'ятовує цифровий код на весь період квантування до моменту надходження наступного коду.

Структурна схема ЦАП, побудованого за принципом цифрового ділення еталонного сигналу, приведена на рис.1.3.



Рисунок 1.3 – Структурна схема ЦАП

Цифровий код з мікропроцесора надходить через комутатор на регістр, що запам'ятовує. Відповідно до прийнятого цифрового коду змінюється коефіцієнт передачі цифрового дільника по каналу “джерело опорної напруги – вихід ЦАП”.

Для однозначного перетворення цифрового коду в аналоговий сигнал необхідно, щоб виконувалося таке рівняння:

$$U_{вих} = E_{ет} \cdot (q_1 \cdot 2^{-1} + q_2 \cdot 2^{-2} + \mathbf{K} + q_n \cdot 2^{-n}),$$

де  $E_{ет}$  – опорна напруга;  $(q_1, q_2, \dots, q_n)$  – цифровий код, у якому  $q_i$  приймає значення “0” або “1”,  $n$  – кількість розрядів цифрового коду.

Технічна реалізація попереднього рівняння може бути виконана в ЦАП, базуючись на методі паралельного сумування струмів, величина яких визначається значенням резисторного дільника типу R-2R, рис.1.4.

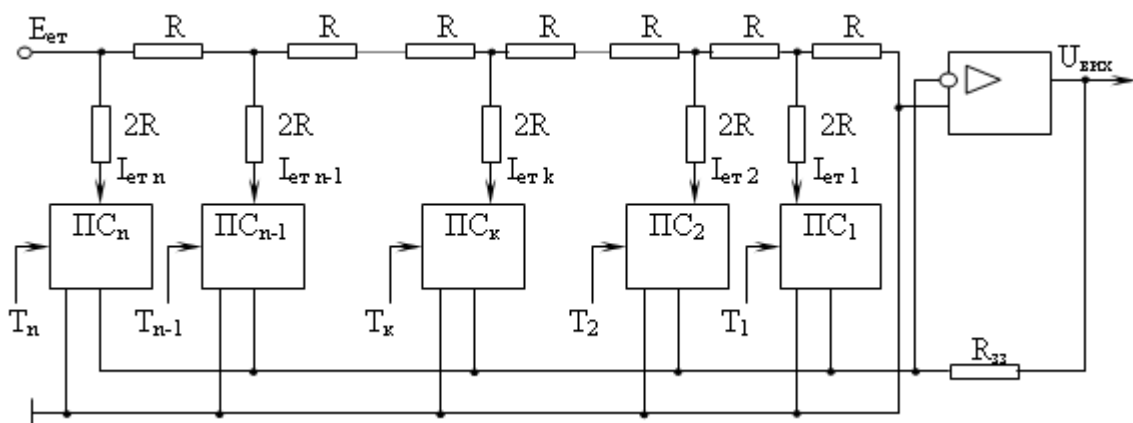


Рисунок 1.4 – Цифро-аналоговий перетворювач

На перемикачі струму ПС<sub>1</sub>, ПС<sub>2</sub>, ..., ПС<sub>n</sub> з відповідних розрядів T<sub>1</sub>, T<sub>2</sub>, ..., T<sub>n</sub> регістра, що запам'ятовує, подаються одиничні або нульові рівні сигналів двійкового цифрового коду. При надходженні на ПС<sub>k</sub> одиничного сигналу T<sub>k</sub> вихідний струм I<sub>ет k</sub> з резистора 2R подається через перемикач ПС<sub>k</sub> на

сигнальний вхід вихідного масштабного операційного підсилювача. Якщо сигнал  $T_k$  має нульовий рівень, то струм  $I_{ет k}$  проходить на землю. Струми  $I_{ет 1}$ ,  $I_{ет 2}$ , ...,  $I_{ет n}$ , які проходять через перемикачі струму на вхід підсилювача, визначаються за формулою:

$$I_{ет k} = 2^{-k} \cdot I_{ет 1},$$

де  $k$  – номер розряду перетворюваного коду;

$I_{ет 1}$  – найменше значення еталонного струму, який проходить через  $ПС_1$  і відповідає молодшому розряду перетворюваного коду.

$$I_{ет 1} = \frac{E_{ет}}{2^n \cdot R}.$$

Загальний струм ЦАП на вході підсилювача визначається як

$$I_{вх y} = \sum_{k=1}^n I_k \cdot I_{ет k} = \sum_{k=1}^n I_k \cdot \frac{2^{-k} \cdot E_{ет}}{2^n \cdot R},$$

де  $q_k$  – цифровий код  $k$ -того розряду.

А вихідна напруга підсилювача  $U_{вих}$  при  $K_y \rightarrow \infty$  ( $K_y$  – коефіцієнт підсилювання розімкненого підсилювача) дорівнює:

$$U_{вих} = -I_{вх y} \cdot R_{зз} = \frac{E_{ет} \cdot R_{зз}}{2^n \cdot R} \cdot \sum_{k=1}^n I_k \cdot 2^{-k},$$

де  $R_{зз}$  – опір зворотного зв'язку підсилювача. При  $R_{зз}/2^n R = 1$  останній вираз відповідає раніше наведеному рівнянню для  $U_{вих}$ .

Наведена схема ЦАП має ту перевагу, що струми  $I_{ет 1}$ ,  $I_{ет 2}$ , ...,  $I_{ет n}$ , які проходять через усі резистори дільника, практично не залежать від стану перемикачів  $ПС_1$ ,  $ПС_2$ , ...,  $ПС_n$ . Точність перетворення ЦАП визначається точністю еталонного джерела напруги, точністю виготовлення опорів і точністю коефіцієнта підсилювання операційного підсилювача. Швидкодія – декілька кілогерців. Час одного перетворення 3,5 мкс.

Існує декілька схем аналого-цифрових перетворювачів, які відрізняються принципами будови та способами реалізації. АЦП порозрядного зрівноваження, рис.1.5а, використовує принцип загального зворотного зв'язку, коли вхідна вимірювальна напруга  $U_x$  зрівноважується сумою еталонних напруг, що виробляє ЦАП. При цьому сумарна еталонна напруга наближується до вимірюваної величини  $U_x$  за траєкторією, що здійснює згасання коливань відносно  $U_x$ , рис.1.5б.

За сигналом “Пуск” генератор тактових сигналів ГТС генерує послідовність керуючих імпульсів, які переміщують за часом одиничний сигнал на виході регістра зсуву РГС. У результаті цього з'являється послідовність керуючих імпульсів  $U_{y1}$ ,  $U_{y2}$ , ...,  $U_{y n+1}$ . У першому такті імпульсом  $U_{y1}$  встановлюється тригер ТГ<sub>1</sub> старшого розряду вихідного регістру РГ<sub>вих</sub> у стан “1”, а решта тригерів (ТГ<sub>2</sub> ... ТГ<sub>n</sub>) імпульсом  $U_{y1}$  скидається у “0”.

Тригер ТГ<sub>1</sub> через компаратор цифро-аналогового перетворювача подає вихідну напругу  $U_{к1}$  до пристрою, що порівнює ПП (нуль-органу).  $U_{к1}$  виробляється цифро-аналоговим перетворювачем, установленим у колі

зворотного зв'язку АЦП. Ця напруга відповідає значенню старшого розряду вихідного регістра ЦАП  $U_{k1} = U_{ет} \cdot \theta \cdot 2^{-1}$ . Нуль-орган порівнює  $U_x$  і  $U_k$ .

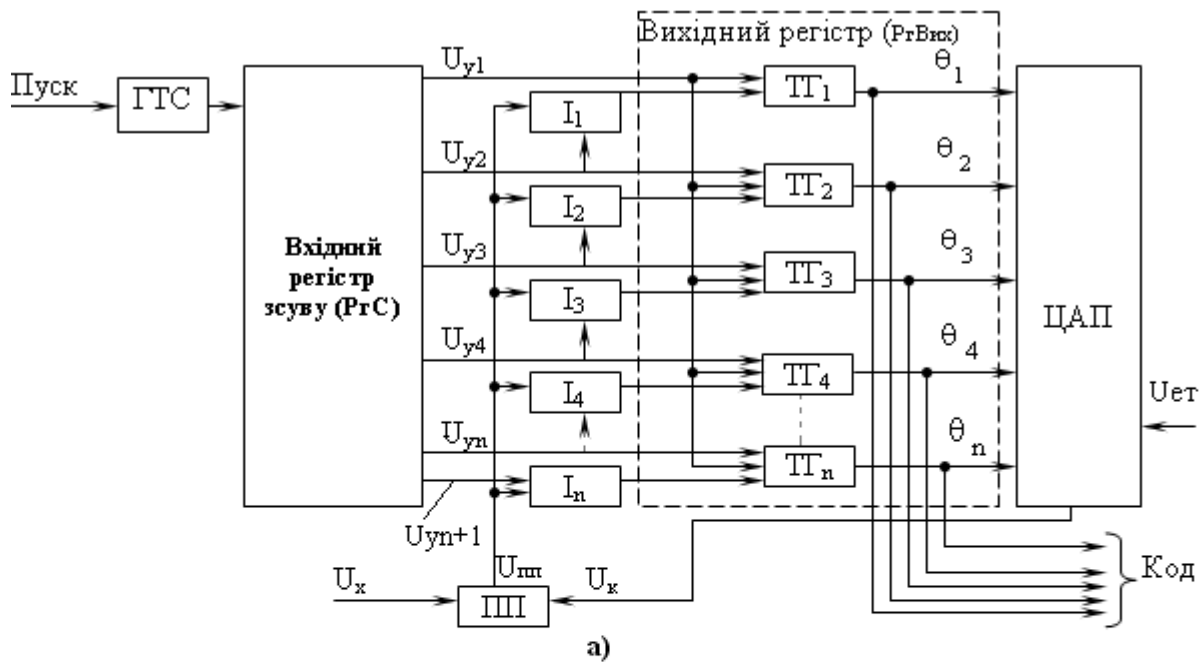


Рисунок 1.5 – АЦП порозрядного зрівноваження (а) і сумарна еталонна напруга (б)

Залежно від знаку різниці  $\Delta U = U_x - U_k$ , на виході ПП з'являється сигнал різного рівня. Якщо у першому такті  $\Delta U < 0$ , то на схемі збігу I з виходу ПП поступає сигнал високого рівня. Тоді в наступному такті імпульс  $U_{y2}$  з вхідного регістру зсуву проходить скрізь схему  $I_1$  і скидає тригер  $ТГ_1$  в стан "0", вимикаючи напругу  $U_{k1}$  від схеми порівняння. Якщо  $\Delta U > 0$ , то  $ТГ_1$  залишається у стані "1" і напруга  $U_{k1}$  подається до ПП. У другому такті тригер  $ТГ_2$  вихідного регістра імпульсом  $U_{y2}$  перекидається у стан "1" і подає до ПП додатково напругу  $U_{ет} \cdot \theta \cdot 2^{-2}$ . Таким чином, після двох тактів у ПП здійснюється порівняння вхідної напруги  $U_x$  з величиною  $U_{k2} = U_{ет} (\theta_1 \cdot 2^{-1} + \theta_2 \cdot 2^{-2})$ , де  $\theta_1$  і  $\theta_2$  відповідно вихідні сигнали  $ТГ_1$  і  $ТГ_2$ , що дорівнюють "1" або "0".



Третій імпульс  $U_{y3}$  з вхідного регістра зсуву встановлює тригер  $ТГ_3$  у стан “1”. Якщо у другому такті величина  $\Delta U = (U_x - U_{k2}) < 0$ , то на виході ПП з’явиться сигнал  $U_{пп}$  високого рівня. Внаслідок цього через схему  $I_2$  сигнал  $U_{y3}$  перекине тригер  $ТГ_2$  у стан “0”, який через комутатор вимкне від ПП напругу  $U_{ет} \cdot 2^{-2}$ . Якщо ж  $(U_x - U_{k2}) > 0$ , то сигнал  $U_{пп}$  має низький рівень, а значить напруга  $U_{ет} \cdot 2^{-2}$  залишиться. При цьому на вхід ПП з ЦАП буде подаватися компенсуюча напруга  $U_{k3} = U_{ет} \cdot (\theta_1 \cdot 2^{-1} + \theta_2 \cdot 2^{-2} + \theta_3 \cdot 2^{-3})$ , де  $q_1, q_2, q_3$ , відповідно, вихідні сигнали  $ТГ_1, ТГ_2$  і  $ТГ_3$ , що дорівнюють “1”.

Описаний процес “зрівняння та віднімання” здійснюється для всіх імпульсів  $U_{y1}, \dots, U_{yn}$ , що надходять з вхідного регістру зсуву, тобто стільки разів, скільки розрядів має вихідний код. Формування коду у молодшому розряді здійснюється імпульсом  $U_{yn+1}$ .

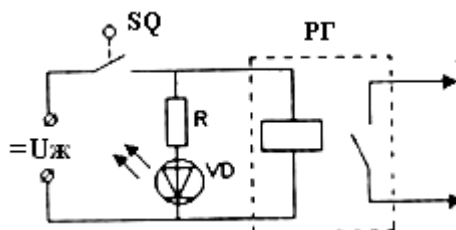
Код результату перетворення знімається з тригерів вихідного регістра  $РГ_{вих}$ . АЦП порозрядного зрівноваження з кількістю розрядів вихідного регістру  $n=12$  має час одного перетворення  $t_{пр}=130$  мкс, тобто у 1с виконується  $\approx 7600$  перетворювань аналогового сигналу в цифровий код. Точність роботи АЦП залежить від стабільності напруги  $U_{ет}$ , точності виготовлення резистивної матриці R-2R в ЦАП, точності нуль-органа в ПУ[12].

### 1.7 Пристрої гальванічного розв’язання електричних кіл контролера

Особливу увагу розробники ПЛК приділяють захисту інформації від впливу різноманітних перешкод, які можуть виникати під час експлуатації контролера в промислових умовах. Насамперед це схемотехнічні рішення, які дають змогу реалізувати надійне гальванічне розв’язання внутрішніх (низьковольтних і слабкострумових) електричних кіл контролера від зовнішніх (високовольтних та сильнострумових) електричних кіл об’єкта керування, які одночасно є джерелом потужних електричних та електромагнітних перешкод.

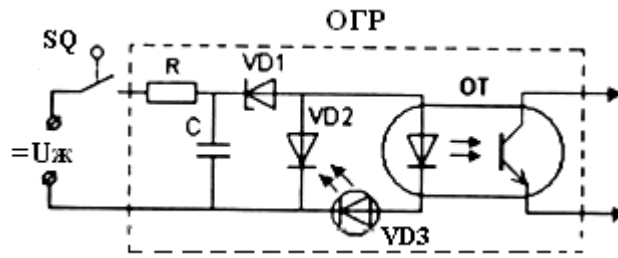
Для дискретних сигналів існує кілька варіантів технічної реалізації гальванічного розв’язання входів і виходів ПЛК.

Найпростішим і найдешевшим є релейний варіант гальванічного розв’язання входів, який виконаний за допомогою мініатюрних герконових реле (РГ), встановлених на платі модуля дискретних входів для кожного дискретного каналу окремо:



Живлення реле може здійснюватись через технологічний контакт SQ як постійним, так і змінним (через діод) струмом. Для індикації спрацьовування вхідного сигналу паралельно обмотці реле можна приєднати світлодіод VD. Недоліком такої схеми є обмежений ресурс спрацьовування герконового реле.

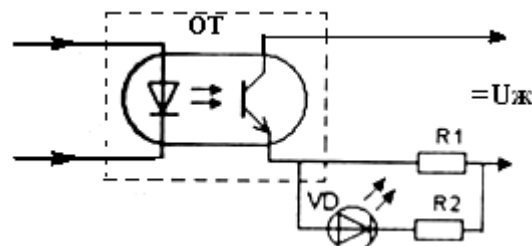
Найчастіше для вхідних ланок постійного струму використовують оптронне гальванічне розв'язання (ОГР):



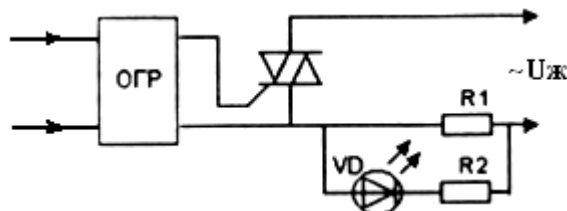
Основним елементом цієї схеми є транзисторна або діодна оптопара (ОТ), яка складається із світлодіода (опромінювача) і фототранзистора або світлодіода. Резистор  $R$  обмежує вхідний струм оптрона ( в межах 14-16 мА), стабілітрон  $VD1$  встановлює поріг спрацювання вхідного кола по напрузі (що досить важливо при приєднанні до ПЛК безконтактних датчиків з великим рівнем залишкової напруги),  $VD2$  забезпечує захист опромінювача оптопари і світлодіода  $VD3$  у разі помилкового підключення полярності вхідного сигналу.  $R$  і  $C$  утворюють фільтр для захисту вхідної ланки від короточасних імпульсних перешкод.

Для вхідних кіл змінного струму використовують таку саму схему але з введенням випрямляча змінного струму на постійний.

У вихідних колах постійного струму використовуються транзисторні оптопари, які виконують функції не тільки гальванічного розв'язання, а й підсилення сигналу:



Необхідно враховувати, що модулі дискретних виходів не «видають» дискретні сигнали, а комутують кола навантаження, які живляться від зовнішніх джерел живлення. Тому в технічних характеристиках дискретних виходів вказується струм, який може комутувати вихідний канал. У колах змінного струму може використовуватись така сама схема, але доповнена симистором:



Більш складно вирішується питання гальванічного розв'язання аналогових входів-виходів. При цьому використовуються або трансформаторні схеми, або схеми послідовного перетворення аналогового сигналу у послідовність імпульсів з подальшим їх підрахунком і перетворенням на цифрову форму[3].

## Тестові та контрольні запитання до розділу 1

**I-й рівень запитань дозволяє перевірити загальний рівень засвоєння матеріалу розділу і передбачає відповідь на запитання або твердження “Так” чи “Ні”.**

1. Розподілені системи керування (DCS) з'явилися завдяки появі мікропроцесорів.
2. Контролери з PC-архітектурою отримали назву Soft DCS.
3. Спеціалісти не в галузі комп'ютерних технологій, а в галузі автоматизації повинні бути розробниками сучасних АСУ ТП.
4. Чи сприяло розвитку засобів автоматизації відокремлення задач створення програмного забезпечення від виробництва апаратних засобів?
5. При відкритій магістрально-модульній архітектурі центральний процесор сполучається з іншими модулями контролера через стандартний інтерфейс.
6. Мезонінні модулі – це додаткові плати, що встановлюються паралельно на основні для розширення їх функціональних можливостей.
7. Softlogic – це інструментальні засоби програмування контролерів, які базуються на вимогах міжнародного стандарту IEC 61131-3.
8. Контролери, що створені на базі шини VME, конструктивно характеризуються можливістю збирання їх з готових плат різних фірм, які випускають продукцію за різними стандартами.
9. PAC-пристрої – це ті ж PC-сумісні контролери, але з іншими можливостями.
10. Пристрій управління в мікропроцесорі призначений для обробки цифрової інформації.
11. Основним елементом для збереження інформації в середині мікропроцесора є ПЗП.
12. Для сучасних контролерів характерна магістральна фізична організація роботи, при якій обмін інформацією між блоками виконується в паралельному коді по магістралях, які називаються шинами.
13. Мікропроцесор, незважаючи на його маленький розмір, здатний самостійно реалізувати конкретну задачу керування об'єктом.
14. Оперативна пам'ять працює за принципом «першим зайшов – останнім вийшов».
15. ОЗП має суттєвий недолік - при вимкненні живлення дані, які записані в ньому, втрачаються.
16. Один дискретний вхід ПЛК здатний приймати байтовий сигнал.
17. Типовий час робочого циклу у сучасних контролерах вимірюється одиницями мілісекунд.

18. При роботі контролера можливі три форми організації робочого циклу – циклічна, хаотична або періодична.

19. Шини в контролерах використовуються для обміну даними між ПЗО.

20. Гальванічне розв'язання електричних кіл контролера потрібне тільки для захисту інформації від впливу електричних перешкод.

*Перевірте відповіді на сторінках: 1 - 9 с., 2 - 11 с., 3 - 12 с., 4 - 12 с., 5 - 12 с., 6 - 12...13 с., 7 - 12 с., 8 - 13 с., 9 - 14 с., 10 - 14 с., 11 - 14 с., 12 - 15 с., 13 - 16 с., 14 - 16 с., 15 - 17 с., 16 - 18 с., 17 - 20 с., 18 - 20 с., 19 - 19 с., 20 - 25 с.*

**II-й рівень запитань (тестові запитання) дозволяє перевірити рівень засвоєння матеріалу розділу із заглибленням у технічні особливості будови та структури контролерів та передбачає вибір правильної відповіді з наведених трьох варіантів.**

**1. З чим пов'язана поява такого поняття, як DCS?**

- а) з впровадженням на виробництвах централізованих АСУ ТП;
- б) з впровадженням в АРМ РС-платформи;
- в) з появою мікропроцесорів.

**2. Яке призначення мезонін-модулів?**

- а) розширити функціональні можливості основних плат;
- б) покращити відмовостійкість контролерів;
- в) прискорити обробку інформації.

**3. Яке призначення в мікропроцесорі має пристрій управління?**

- а) керує обміном та обробкою інформації у ПЛК;
- б) обробляє цифрову інформацію;
- в) формує керуючі сигнали для управління виробничими процесами.

**4. До складу мікропроцесора входять:**

- а) арифметично-логічний пристрій, блок внутрішніх регістрів, пристрій керування, шина даних, шина адреси і магістральна шина;
- б) арифметично-логічний пристрій, блок внутрішніх регістрів, пристрій керування, шина управління, шина даних і шина адреси;
- в) арифметично-логічний пристрій, блок внутрішніх регістрів, шина управління, шина даних і шина адреси.

**5. Окрім мікропроцесора, до складу контролера входять:**

- а) ОЗП, ПЗП, вхідні та комунікаційні модулі;
- б) ОЗП, ПЗП, модулі ПЗО, комунікаційні модулі;
- в) ЗПО, ППЗ, модулі ОЗП, комунікаційні модулі.

**6. Якої розрядності, як правило, використовуються АЦП у ПЛК?**

- а) 8-12;
- б) 16-18;
- в) 2-6.

**7. В яких видах ПЗО носієм інформації є один біт?**

- а) ДЦП, ЦДП;
- б) тільки у ДЦП;
- в) в усіх видах ПЗО.

**8. Як часто виконується програмне самодіагностування контролера?**

- а) один раз за секунду;
- б) протягом кожного робочого циклу ПЛК;
- в) період самодіагностування задається програмно.

**9. Робота якого перетворювача сигналів базується на методі паралельного сумування струмів?**

- а) ЦАП;
- б) у жодному перетворювачі не здійснюється паралельне сумування струмів;
- в) ЦАП та АЦП.

**10. В яких АЦП при перетворенні сигналу використовується принцип загального зворотного зв'язку?**

- а) тільки в АЦП подвійного інтегрування;
- б) в АЦП подвійного інтегрування та порозрядного зрівноваження;
- в) тільки в АЦП порозрядного зрівноваження.

**11. Чим характеризується шина VME?**

- а) універсальністю;
- б) 32 або 64 розрядна з пропускнуою спроможністю до 40 Мбод;
- в) має формат 3U.

**12. В чому перевага РАС над ПЛК?**

- а) РАС-пристрої мають архітектуру і операційну систему персональних комп'ютерів, потужні процесори, великий об'єм пам'яті;
- б) РАС-пристрої більш дешеві;
- в) РАС-пристрої мають закриту архітектуру, унікальну операційну систему і великий об'єм пам'яті.

**13. Який вигляд має магістральна шина?**

- а) кілька паралельних провідників із загальним функціональним призначенням;
- б) профільована пластина для встановлення модулів контролера;
- в) мідна контактна пластина.

**14. В якому вигляді подається інформація у мікропроцесор?**

- а) у вигляді двійкових чисел;
- б) у вигляді дискретних сигналів;
- в) у вигляді аналогових сигналів.

**15. Яка схема гальванічного розв'язання входів для дискретних сигналів є найпростішою і найдешевшою ?**

- а) релейна;
- б) оптронна;
- в) транзисторна.

**16. Яким форматам відповідають конструктиви стандарту Євромеханіки?**

- а) формат 3D;

- б) формат 3U і 6U;
- в) форматам, визначеним стандартом ІЕС 61131.

**17. При якій організації роботи контролера тривалість циклу може змінюватися?**

- а) при періодичній;
- б) тривалість циклу завжди незмінна;
- в) при циклічній.

**18. Що ілюструє рівняння  $U_{вих} = E_{ет} \cdot (q_1 \cdot 2^{-1} + q_2 \cdot 2^{-2} + \mathbf{K} + q_n \cdot 2^{-n})$  ?**

- а) взаємозв'язок між вихідним сигналом і еталонним;
- б) роботу ЦАП;
- в) перетворення цифрового коду в аналоговий сигнал?

**19. Яке призначення має вихідний регістр в АЦП порозрядного зрівноваження?**

- а) формує цифровий сигнал;
- б) формує сигнал зворотного зв'язку;
- в) формує сигнал на виході порівняльного пристрою.

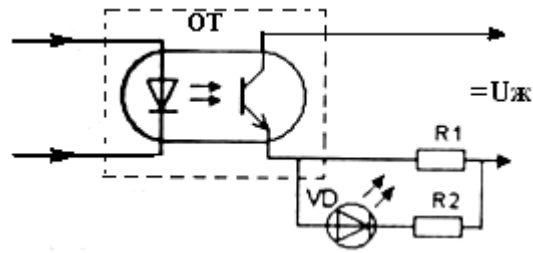
**20. Коли в роботі контролера відбувається самодіагностування?**

- а) на початку кожного робочого циклу ПЛК;
- б) в процесі кожного робочого циклу ПЛК;
- в) після опитування стану входів ПЛК.

*Перевірте відповіді на сторінках: 1 - 9 с., 2 – 12...13 с., 3 - 14 с., 4 – 14...16 с., 5 - 16 с., 6 - 18 с., 7 - 18 с., 8 - 21 с., 9 - 22 с., 10 - 23 с., 11 - 13 с., 12 - 14 с., 13 - 15 с., 14 - 16 с., 15 - 25 с., 16 - 13 с., 17 - 20 с., 18 - 22 с., 19 - 25 с., 20 - 21 с.*

**III-й рівень запитань (контрольні запитання) дозволяє перевірити рівень засвоєння матеріалу із систематизацією отриманої у розділі інформації та передбачає відповіді, що побудовані на логічному аналізі або порівнянні із приведенням конкретних технічних параметрів (даних або прикладів).**

1. Проаналізуйте тенденції розвитку ПЛК, і покажіть як змінювалися за часом конструкція та можливості контролерів.
2. Систематизуйте інформацію про види пам'яті контролерів у вигляді структурно-ієрархічної схеми.
3. Перерахуйте основні конструктивні особливості та можливі варіанти побудови контролерів.
4. Нарисуйте схему мікропроцесора та поясніть його роботу і призначення елементів.
5. Як організована робота контролера?
6. Якими шляхами удосконалюються сучасні контролери?
7. Нарисуйте схему ЦАП і поясніть, як цифровий код, що перетворюється, визначає величину вихідної напруги.
8. Прокоментуйте цю схему:



9. Які існують типи порушень працездатності контролерів і до чого вони призводять?

10. Для чого потрібне гальванічне розв'язання електричних кіл контролера?

*Перевірте відповіді на сторінках: 1 – 9...14 с., 2 – 14...19 с., 3 – 10...14 с., 4 – 14...16 с., 5 – 19...21 с., 6 – 12...14 с., 7 – 22...23 с., 8 – 25...26 с., 9 - 21 с., 10 – 25...26 с.*

## **2 ОСНОВНІ ХАРАКТЕРИСТИКИ ТА МОЖЛИВОСТІ КОНТРОЛЕРІВ**

Велика різноманітність ПЛК на ринку сучасних засобів автоматизації ставить перед розробниками АСУ ТП завдання правильного вибору контролера, вирішенням якого є компроміс між вартістю, технічним рівнем, надійністю, комфортністю, витратами на сервісне обслуговування, повнотою програмного забезпечення та багато чим іншим [11,13].

Зазвичай при побудові систем управління, як основні характеристики, на підставі яких здійснюється вибір контролерів, використовуються п'ять узагальнених показників:

- характеристика центрального процесора;
- характеристика каналів вводу-виводу, що підтримуються контролерами;
- комунікаційні можливості;
- умови експлуатації;
- програмне забезпечення.

### **2.1 Характеристика центрального процесора**

Центральні процесори характеризуються типом, розрядністю основної процесорної плати, робочою частотою, а також функціональними можливостями, основними з яких є:

- кількість шасі, які можуть входити до складу ПЛК;
- кількість входів-виходів, які може обробити контролер;
- наявність функції ПІД-регулювання та спеціальних модулів (лічильники, модулі вимірювання ваги та ін.);
- кількість і типи мереж, до яких може приєднуватись ПЛК;
- підтримка математики з плаваючою комою.

В значній мірі всі ці характеристики і можливості забезпечуються об'ємом і видом пам'яті (ОЗП (RAM), ПЗП (ROM), СППЗП (EPROM), ЕСППЗП (EEPROM), флеш (Flash)), яку мають процесори.

Оперативний запам'ятовуючий пристрій – ОЗП або RAM (random access memory - пам'ять з довільним доступом) є типом пам'яті, яка дозволяє читання і запис у будь-яку комірку без попереднього пошуку. У контролерах цей тип пам'яті використовується для зберігання програм і значень технологічних параметрів (даних). ОЗП втрачає інформацію при вимкненні живлення, проте існують "незалежні" модулі ОЗП, що містять вбудоване джерело автономного живлення.

Залежно від способу, який використовується для зберігання інформації, всі ОЗП можна поділити на статичні (SRAM) і динамічні (DRAM). У статичних ОЗП елементом зберігання інформації є тригер, а у динамічних - конденсатор. Час доступу у мікросхемах DRAM становить 60–100 нс, а у мікросхемах SRAM –10-15нс. Але пам'ять SRAM коштує значно більше, тому для зберігання



великих обсягів інформації використовують DRAM, а як швидкодіючу пам'ять обмеженого обсягу – SRAM.

Постійний запам'ятовуючий пристрій ПЗП або ROM (Read Only Memory - пам'ять тільки для читання) уявляється як масив адресованих комірок (матриця). Кожна комірка пам'яті може кодувати одиницю інформації. Дані на ROM записуються при її виготовленні шляхом нанесення на матриці алюмінієвих сполучних доріжок літографічним способом. Наявність або відсутність у відповідному місці такої доріжки кодується "1" або "0".

У контролерах пам'ять типу ПЗП використовується для зберігання програм користувача. Даний тип пам'яті не набув широкого поширення у зв'язку з тим, що сучасне програмне забезпечення часто має багато недоробок і вимагає оновлення, тоді як виробничий цикл виготовлення пам'яті достатньо тривалий (4-8 тижнів).

Перепрограмований постійний запам'ятовуючий пристрій зі стиранням записаної інформації (СППЗП) або EPROM (Erasable Programmable ROM), перепрограмований постійний запам'ятовуючий пристрій з електричним стиранням запису ЕСППЗП або EEPROM (Electrically EPROM), а також Flash (флеш-пам'ять) відносяться до класу незалежної перезаписуваної пам'яті (англійський еквівалент - nonvolatile read-write memory або NVRWM).

В EPROM перед записом необхідно здійснити стирання (для отримання можливості перезаписувати вміст пам'яті). Стирання комірок EPROM виконується відразу для всієї мікросхеми за допомогою опромінювання чипа ультрафіолетовим або рентгенівським промінням протягом декількох хвилин.

Коли відбувається стирання інформації в EPROM, усі біти області пам'яті переводяться в один стан (звичайно в одиницю, рідше - в нуль). Запис на EPROM здійснюється за допомогою програматорів. Великою перевагою такої пам'яті є можливість перезаписувати вміст мікросхеми.

EEPROM розроблена у 1979 році в компанії Intel. Головною відмінністю EEPROM (у т.ч. і Flash) від інших типів незалежної пам'яті є можливість перепрограмування при підключенні до стандартної системної шини мікропроцесорного пристрою. В EEPROM стирання окремої комірки здійснюється за допомогою електричного струму. Відбувається стирання автоматично при запису в комірку нової інформації, тобто можна змінити дані в будь-якій комірці, не зачіпаючи інші. Процедура стирання набагато триваліше процедури запису.

EEPROM в порівнянні з EPROM має збільшений ресурс роботи, простіша у використанні, але характеризується більш високою вартістю. У контролерах цей тип пам'яті використовується як для зберігання програм, так і для зберігання даних.

Flash (повна назва - Flash Erase EEPROM) вперше була розроблена компанією Toshiba у 1984 році, і вже наступного року розпочалось виробництво мікросхем flash-пам'яті у промислових масштабах. У 1988 році компанія Intel розробила власний варіант флеш-пам'яті.

У флеш-пам'яті використовується дещо відмінний від EEPROM тип комірки-транзистора. Технологічно флеш-пам'ять споріднена як з EPROM, так і з EEPROM. Основна відмінність флеш-пам'яті від EEPROM полягає у тому, що стирання вмісту комірок виконується або для всієї мікросхеми, або для певного блока (кластера, кадру або сторінки). Звичайний розмір такого блока складає 256 або 512 байт, проте в деяких видах флеш-пам'яті об'єм блока може досягати 256 Кбайт. Існують мікросхеми, що дозволяють працювати з блоками різних розмірів (для оптимізації швидкодії). Стирати можна як блок, так і вміст всієї мікросхеми відразу.

Флеш-пам'ять іноді називають "флеш-диском" тому що вона допускає багаторазовий (до 1 млн. циклів) перезапис даних і по суті виконує функції, які раніше виконували накопичувачі на жорстких та гнучких магнітних дисках. Конструктивно "флеш-пам'ять" це мікросхема або твердотільна картка, яка не має ніяких рухомих частин. Тому швидкість зчитування даних з флеш-пам'яті перебуває у межах 1-5 Мбайт за секунду. І хоча до останнього часу флеш-пам'ять як ОЗП не використовується, у зв'язку зі збільшенням її випуску і зменшенням ціни можна очікувати, що у майбутньому мікросхеми DRAM у ОЗП, можуть поступитися місцем мікросхемам або карткам флеш-пам'яті[13].

## **2.2 Характеристика каналів вводу-виводу**

Характеристика каналів вводу-виводу, що обслуговуються центральним процесором, найчастіше є визначальною при виборі контролерів. При цьому, важливою є не тільки кількість каналів, а і різноманітність модулів вводу-виводу за кількістю і рівнем комутованих сигналів (струм/напруга), способи підключення зовнішніх кіл до модулів вводу-виводу, кількість каналів локального, віддаленого і розподіленого вводу-виводу.

Більшість фірм-виробників поставляють на ринок засобів і систем автоматизації сімейства контролерів, кожне з яких розраховане на певний набір виконуваних функцій і об'єм оброблюваної інформації. Серед них є контролери невеликої обчислювальної потужності (мікро), що призначені для управління простими об'єктами і, які здатні підтримувати максимум декілька десятків ввідів-виводів, в основному, дискретних. До таких ПЛК відносяться, наприклад, контролери сімейства MicroLogix (Allen-Bradley), LOGO (Siemens), Nano (Schneider Electric), Durus (GE Fanuc).

Сімейства малих контролерів здатні підтримувати вже сотні ввідів-виводів, виконувати складніші функції. Ці контролери мають достатньо розвинений аналоговий ввід-вивід, виконують операції з плаваючою комою і функції ПІД-регулювання. До цієї групи контролерів можна віднести SLC 500 (Allen-Bradley), Simatic S7-200 (Siemens), Twido (Schneider Electric), Fanuc 90-30 (GE Fanuc).

Контролери середньої потужності, маючі достатню пам'ять і швидкодію, можуть обробляти вже тисячі змінних дискретного, аналогового і швидкісного типу. Це контролери PLC-5 (Allen-Bradley), Premium (Schneider Electric), CJ1 (Omron), Simatic S7-300 (Siemens) та ін.

Деякі крупні фірми випускають клас контролерів дуже високої обчислювальної потужності, які мають пам'ять, що вимірюється мегабайтами і десятками мегабайтів. Їх здатність обробляти десятки тисяч змінних визначила одну з областей застосування – у якості концентраторів інформації, що одержується від локальних контролерів. Обчислювальні можливості цього класу контролерів дозволяють реалізовувати складні алгоритми (адаптивне, оптимальне управління), що використовуються при автоматизації безперервних технологічних процесів. Найяскравішими представниками цієї групи контролерів є ControlLogix (Allen-Bradley), Simatic S7-400 (Siemens), PACSystems RX7i (GE Fanuc), Quantum (Schneider Electric).

Як закордонні, так і вітчизняні виробники контролерів комплектують свої вироби широкою гаммою модулів дискретного і аналогового вводу-виводу. По кількості сигналів, що обробляються, модулі розрізняють на 2-, 4-, 8-, 16-, 32- і 64-канальні. Така різноманітність модулів полегшує підбір необхідної конфігурації контролера і дозволяє мінімізувати вартість технічних засобів.

Сигнали, що комутуються модулями дискретного вводу-виводу, можуть мати різний рівень напруги. Серед них 12, 24 і 48 В постійного струму, а також 120 і 240 В змінного струму.

Рівні сигналів, що комутуються модулями аналогового вводу-виводу, можуть бути – 0-5В, 0-10В,  $\pm 5$ В,  $\pm 10$ В по напрузі і 0-20мА, 4-20мА по струму. Серед модулів вводу-виводу є модулі, які призначені спеціально для роботи з термопарами і термометрами опору різних градувань.

Розрізняються модулі вводу-виводу і за способом підключення зовнішніх кіл. До одних модулів зовнішні кола приєднуються за допомогою клем з гвинтовими затисками. Інші виробники пропонують системи вводу-виводу, в яких зовнішні низьковольтні кола підключаються за допомогою пружинних затисків.

Розповсюджене також приєднання зовнішніх кіл до модулів вводу-виводу через знімні термінальні блоки або фронтальні з'єднувачі. Це дозволяє здійснювати заміну модулів без демонтажу зовнішніх проводів.

На лицевій панелі дискретних модулів вводу-виводу можуть бути розташовані світлодіоди індикації стану їх каналів[13].

### **2.3 Конфігурація ввідів-виводів**

Однією з найважливіших характеристик контролерів є їх здатність підтримувати локальний, розширений, віддалений і розподілений ввід-вивід.

При локальному ввіді-виводі модулі розміщуються безпосередньо на тому самому шасі, на якому встановлений центральний процесор. Оскільки кількість слотів в шасі обмежена, то і кількість локальних ввідів-виводів також обмежена. Перевага локального вводу-виводу полягає у тому, що вони мають високу швидкість оновлення даних, оскільки взаємодіють з центральним процесором напряму.

Для підтримки більшої кількості змінних багато контролерів мають можливість розширення локального вводу-виводу за допомогою виносних шасі. Ці шасі розширення з розміщеними в них модулями вводу-виводу з'єднуються між собою спеціальним коротким кабелем і можуть бути віднесені максимум на декілька десятків метрів від центрального процесора. Деякі комплекси контролерів здатні підтримувати одне-два шасі розширення, інші – десятки шасі з дуже великою кількістю модулів вводу-виводу.

Наприклад, для збільшення кількості модулів в контролерах CS1 (Omron) використовуються панелі розширення, які приєднуються до базової панелі за допомогою кабелів довжиною – 0,3 м, 0,7 м, 2 м, 3 м, 5 м, 10 м, 12 м. Загальна протяжність з'єднувальних кабелів між панелями розширення не повинна перевищувати 50 м. До однієї панелі центрального процесора може бути приєднано до семи панелей розширення, що дозволяє довести загальну кількість модулів вводу-виводу в системі до 80.

Віддалений ввід-вивід застосовується для систем, в яких є велика кількість датчиків і інших польових пристроїв, що знаходяться на достатньо великій відстані (1000 і більше метрів) від центрального процесора. Такий підхід дозволяє зменшити вартість кабелів, що з'єднують датчики і виконавчі пристрої з контролерами, за рахунок того, що модулі вводу-виводу розміщуються поблизу польових пристроїв.

Виробники апаратних засобів автоматизації розв'язують проблему віддаленого вводу-виводу по-різному. Підтримка віддаленого вводу-виводу може здійснюватися за допомогою модулів, які називаються "віддалений провідний" і "віддалений підпорядкований". Провідний модуль розташовується в локальному каркасі контролера і з'єднується кабелем з "віддаленим підпорядкованим", який знаходиться у віддаленому каркасі (наприклад, контролер Quantum компанії Schneider Electric). Один провідний модуль може підтримувати 32, 64, 125 підпорядкованих. Таким чином, системи управління, що побудовані за технологією віддаленого вводу-виводу, здатні обробляти тисячі параметрів.

Інше рішення організації віддаленого вводу-виводу забезпечується вбудованим в процесор портом, що виконує роль "майстра", як це відбувається в контролерах PLC-5 фірми Allen-Bradley.

Розподілений ввід-вивід є різновидом віддаленого, з тією лише різницею, що кількість параметрів, яку потрібно обробляти, не велика (від декількох параметрів до десятків). Тому рішення із застосуванням каркасів віддаленого вводу-виводу, які розраховані на достатньо велику кількість параметрів, може виявитися досить дорогим. У зв'язку з цим деякі фірми пропонують спеціалізовані рішення для реалізації розподіленої системи. Наприклад, контролери серії Modicon TSX Quantum компанії Schneider Electric можуть приєднуватися до трьох мереж розподіленого вводу-виводу. Одна мережа реалізується через порт Modbus Plus, що є на ЦПУ, а дві інші - за допомогою додаткових модулів інтерфейсного зв'язку NOM, які мають порти Modbus і Modbus Plus та забезпечують розширені можливості зв'язку для системи

Quantum.

Для реалізації віддаленого і розподіленого вводу-виводу деякі виробники контролерів пропонують так звані системи децентралізованої периферії - SIMATIC ET 200 (Siemens), Genius і Field Control (GE Fanuc), Momentum (Schneider Electric), у складі яких є процесорні модулі, адаптери зв'язку і модулі вводу-виводу. Процесорні модулі по спеціальним каналам забезпечують безперешкодний обмін інформацією між периферією, яку вони контролюють, і ЦПУ контролера розташованого у центральному пункті [13].

## 2.4 Комунікаційні можливості

Основною тенденцією розвитку сучасних систем автоматизації є створення багаторівневих систем керування виробництвом, для яких характерним є інтенсивний обмін інформацією між окремими технічними засобами, робочими станціями і рівнями керування. Тому питанням мережних можливостей ПЛК приділяється особлива увага.

До параметрів контролерів, що характеризують їх здатність взаємодіяти з іншими пристроями системи управління, відносяться:

- кількість і різноманітність портів в процесорних модулях;
- широта набору інтерфейсних модулів і інтерфейсних процесорів;
- протоколи, що підтримуються;
- швидкість обміну даними і протяжність каналів зв'язку.

На рис.2.1 наведена мережна архітектура багаторівневої системи управління.

На рівні управління підприємством знаходяться звичайні РС-сумісні комп'ютери і файлові сервери, що об'єднані локальною мережею і обмінюються великими об'ємами інформації. Задача обчислювальних систем на цьому рівні — забезпечення візуального контролю основних параметрів виробництва, побудова звітів, архівація даних. Масив даних, що пересилається, може бути доступний не лише центральному пристрою, але і іншим вузлам мережі цього рівня. Це означає, що мережа є рівноправною (одноранговою), тобто визначається моделлю взаємодії peer-to-peer (рівний з рівним). Об'єми даних, що передаються між вузлами, вимірюються мегабайтами, а часові показники обміну інформацією не є критичними.

Мережі, що забезпечують інформаційний обмін на цьому рівні, називають інформаційними мережами. Найбільш яскравим представником мереж цього рівня є Ethernet з протоколом TCP/IP.

Мережі, що забезпечують інформаційний обмін між контролерами, датчиками і виконавчими пристроями, часто об'єднуються під спільною назвою – промислові мережі. Їх можна поділити на два рівні:

- управляючі промислові мережі, що забезпечують збір і обробку даних на рівні промислових контролерів та управління технологічним процесом;
- польові мережі або шини, завдання яких зводяться до опитування датчиків і управління роботою різноманітних виконавчих пристроїв.

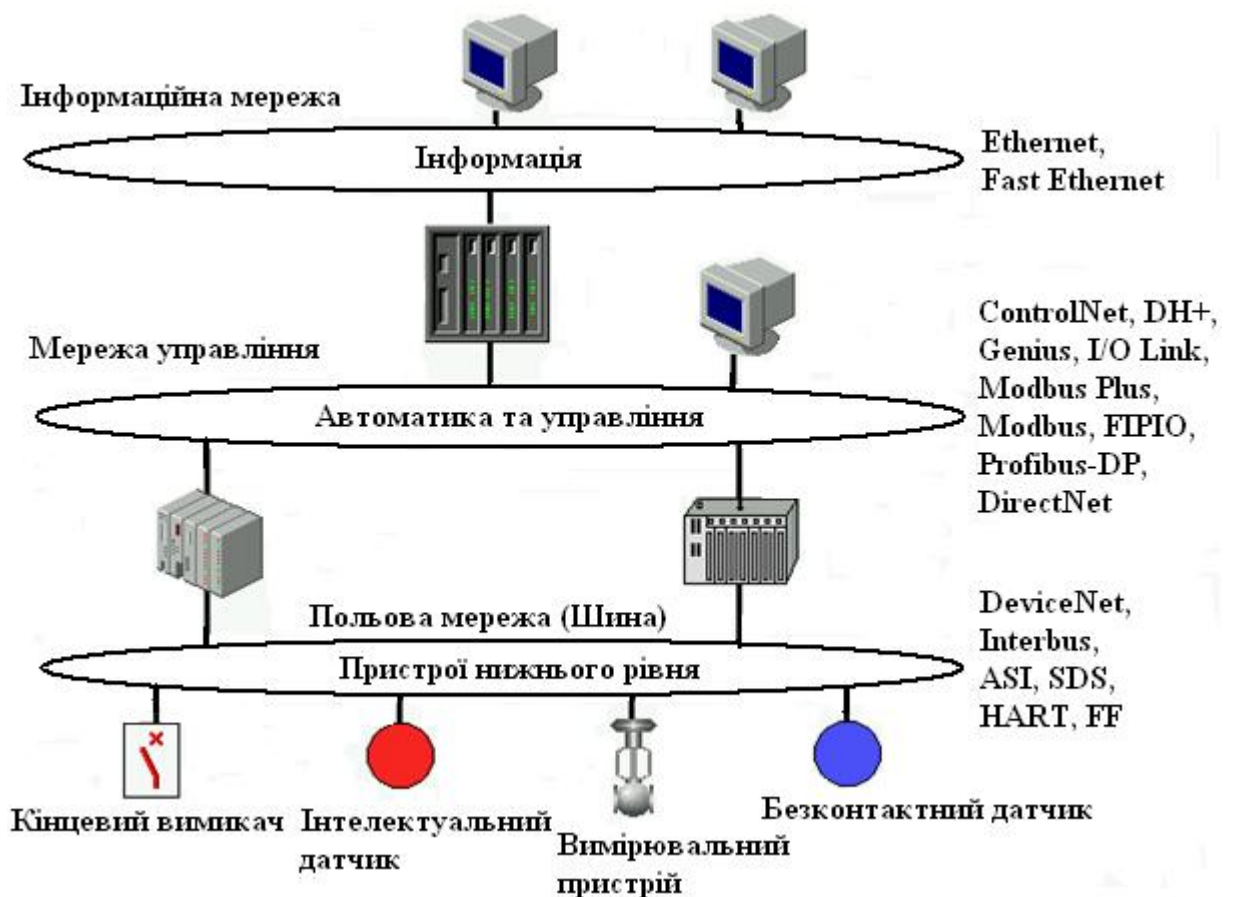


Рисунок 2.1 – Мережна архітектура АСУТП

Міжнародна організація зі стандартизації OSI (Open Systems Interconnect) розробила еталонну модель структури, яка забезпечує взаємодію відкритих систем, тобто спільну роботу в мережі технічних засобів різних фірм. Особливу роль у мережах ПЛК відіграють так звані протоколи зв'язку, що являють собою визначений набір правил, процедур і форматів повідомлень, які необхідні для організації і забезпечення інформаційних зв'язків між ПЛК, а також між ними і зовнішніми пристроями і системами.

Відповідно до еталонної моделі відкритий протокол сполучень включає сім рівнів, на кожному з яких виконується визначена функція:

- перший рівень фізичний — забезпечення фізичних характеристик приєднання пристроїв до фізичних каналів зв'язку (у разі потреби перетворення сигналів);
- другий рівень каналний - забезпечення передачі даних між двома пристроями (з'єднання і роз'єднання каналу зв'язку, захист від помилок у процесі передачі даних);
- третій рівень мережний - вибір, організація і оптимізація маршрутів передачі інформації;
- четвертий рівень транспортний - керування передачею (пересилкою) даних (масивів) між об'єктами мережі;

- п'ятий рівень сеансовий - організація і проведення сеансів зв'язку між прикладними програмами, включаючи розв'язання задач синхронізації і активізації окремих об'єктів або фрагментів програм;
- шостий рівень представницький - перетворення даних, які передаються, до вигляду зручного для кінцевого використання;
- сьомий рівень прикладний - виконання прикладних програм (адміністрування мережею, керування технологічними процесами і транспортними засобами різного призначення).

Перші три рівня забезпечують пересилання даних, а рівні четвертий - сьомий забезпечують керування першими трьома. Залежно від складності задачі керування і розмірів мереж, кількість рівнів може змінюватись. Тому досить часто у мережах ПЛК використовуються найпростіші рівні.

Функціонування системи відповідно до наведеного протоколу виконується за схемою «згори - донизу» — при передаванні даних і “знизу – догори” - при прийманні інформації.

Важливу роль у мережах ПЛК відіграють способи і засоби передачі інформації. Як фізичний канал у сучасних ПЛК можуть використовуватись, звичайні і спеціальні коаксіальні кабелі, виті пари, оптоволоконні канали сполучень.

Передача інформації по каналах сполучень здійснюється симплексним методом – тільки в одному напрямку, напівдуплексним – в обох напрямках, але не одночасно, дуплексним – одночасно в обох напрямках. При цьому використовують як послідовний, так і паралельний способи передачі інформації з синхронізацією передачі повідомлень і без неї.

Існують три основні режими обміну даними, ефективність використання яких залежить від конкретної задачі.

Режим «Провідний-підпорядкований». У цьому режимі один з вузлів промислової мережі є провідним пристроєм (master), який послідовно опитує підпорядковані вузли (slave). Залежно від змісту запиту підпорядкований вузол або виконує одержану команду, або передає провідному поточні дані з підключених кінцевих пристроїв. Slave не може посилати запити. Типовим прикладом мережі, побудованої за таким принципом, є мережа PROFIBUS. Як правило, ролі провідного і підпорядкованого закріплюються за контролерами жорстко і не змінюються в процесі функціонування мережі.

Режим «Клієнт-сервер». Даний режим має багато загального з попереднім і використовується у системах з гнучким розподілом функцій. Вузол клієнт запрошує дані, а вузол сервер їх надає. При цьому клієнт може запрошувати декілька вузлів, а сервер – мати декілька клієнтів. Також функції клієнта і сервера можуть поєднуватися на одному вузлі. Прикладом цього режиму роботи є промислова мережа Foundation Fieldbus.

Режим «Підписка». У цьому режимі вузол, що потребує регулярне надходження будь-якої інформації, підписується на її отримання від іншого вузла і після цього одержує регулярні розсилки даних без додаткових запитів. Режим має два варіанти – у першому дані передаються циклічно з певним

інтервалом незалежно від динаміки інформації, у другому дані передаються тільки у разі їх зміни. Даний режим також використовується в мережах Foundation Fieldbus.

Швидкість, з якою дані передаються в мережі, звичайно вимірюється в кілободах (КБод) або мегабодах (МБод). Швидкість передачі показує, скільки даних може бути передано протягом певного проміжку часу. Наприклад, швидкість передачі 19,2 КБод означає 19 200 біт за секунду. Всі пристрої, які обмінюються даними через певну мережу, повинні бути настроєні на передачу з однаковою швидкістю. Тому щонайвища швидкість передачі в мережі визначається найповільнішим приладом, підключеним до мережі. Мережна адреса - це унікальний номер, який призначається кожному пристрою в мережі і гарантує, передачу або прийом інформації потрібними засобами автоматизації[3,13,14].

## 2.5 Характеристика промислових мереж

Для функціонування багаторівневих систем керування виробництвом розроблена і впроваджена значна кількість закінчених апаратно-програмних рішень, які суттєво відрізняються своїми параметрами і сферами застосування.

Існує досить широкий спектр протоколів для промислових мереж рівня польових шин. Серед них AS-інтерфейс, CAN – протокол, SDS – протокол, DeviceNet – мережа, специфікація Interbus, HART – протокол та ін.

AS-інтерфейс у вигляді концепції з'явився у 1993 році. Основне завдання цього інтерфейсу – зв'язати в єдину інформаційну структуру пристрої нижнього рівня технологічного процесу, що автоматизується, (фотоелектричні датчики, виконавчі пристрої, реле, контактори, перемикачі, приводи і тому подібне) з системою контролерів. Це підтверджується і назвою мережі - ASI (Actuator Sensor Interface).

AS-інтерфейс дозволяє повністю виключити з АСУ ТП аналогові лінії зв'язку, кросові шафи та інше допоміжне устаткування. Максимальний час циклу опитування складає 5-10 мс, що відповідає циклу виконання програми в контролері, а максимальний об'єм даних з одного ASI-вузла дорівнює 4 біта. Завдяки цьому мережі на базі AS-інтерфейсу активно застосовуються в розподілених АСУ ТП реального часу.

AS-інтерфейс дозволяє через свої комунікаційні лінії не лише передавати дані, але і підводити живлення (24 VDC) до датчиків і виконавчих пристроїв. До одного контролера можна приєднати до 31 пристрою, а протяжність сегменту ASI - шини досягає 100м. За рахунок повторювачів довжину мережі можна збільшити до 300м, а кількість вузлів – до 62. Дані і напруга живлення передаються по одній парі проводів. Мережі на базі AS-інтерфейсу відрізняються економічністю і дуже великим набором засобів взаємодії з іншими промисловими мережами.

Протокол CAN (Controller Area Network) визначає тільки перші два рівні ISO/OSI – фізичний та рівень доступу до середовища передачі даних. З початку 90-х років, коли компанія Bosch розробила відповідну специфікацію, на основі



цього протоколу реалізована величезна кількість повнофункціональних мереж, зокрема таких як SDS, CANOpen, DeviceNet та ін. Кількість вузлів мережі, що працюють на основі CAN, обчислюється десятками мільйонів. Практично у кожного крупного виробника мікроконтролерів є виріб з CAN-інтерфейсом. Основними перевагами, що визначили високу популярність цього протоколу у розробників промислових систем, є висока швидкість (до 1 Мбод), можливість мати в мережі декілька провідних пристроїв, надійна система виявлення і виправлення помилок. Метод доступу до інформації поєднує мінімальну затримку передачі інформації з ефективним арбітражем ситуацій, коли декілька вузлів починають передавати дані одночасно. Завдяки цьому доставка повідомлення гарантується.

DeviceNet – відкрита комунікаційна мережа нижнього рівня. Ця мережа сполучає пристрої нижнього рівня безпосередньо з системою управління, зменшуючи кількість зв'язків вводу-виводу і проводів по відношенню до типових апаратних рішень. Довжина мережі DeviceNet визначається швидкістю передачі даних – 100м при швидкості 500 Кбод, 200м – 250 Кбод, 500м – 125 Кбод. Максимальна кількість вузлів-64.

DeviceNet забезпечує доступ до інтелектуальних датчиків різних виробників, зв'язок "провідний/підпорядкований" і рівноправний, конфігурацію датчиків, управління і збір даних.

SDS (Smart Distributed System) – протокол систем вводу-виводу, запропонований компанією Honeywell для побудови мереж, які об'єднують периферійні пристрої різних виробників.

Ця мережа дозволяє працювати з такими пристроями вводу-виводу, як кінцеві вимикачі, фотоелектричні і безконтактні датчики, позиціонери, і здійснювати обмін інформацією на високих швидкостях. Перевагою мережі є те, що вона призначена для контролерів і джерел інформації, живлення передається по проводах мережного кабелю, діагностика виконується на рівні фізичних пристроїв, а час проходження даних по мережі досягає 0,1 мс. Швидкість передачі інформації залежить від довжини шини. Коли відстань не перевищує 30м, швидкість передачі даних - 1Мбод. При довжині шини 487м – 125Кбод.

Модуль SDS замінює стандартні модулі вводу-виводу, забезпечуючи приєднання 64 дискретних входів/виходів (розподілений ввід-вивід).

Специфікація Interbus розроблена фірмою Phoenix Contact у 1984 році. Основне призначення протоколу Interbus - організація комунікацій на рівні датчиків і виконавчих механізмів. Interbus використовує процедуру доступу до шини за схемою «провідний/підпорядкований» (Master/Slave). Топологія Interbus - це фізичне і логічне кільце, у якого фізичний рівень побудований на основі стандарту RS-485. Для реалізації кільця використовуються дві виті пари (дуплексний режим). Така фізична структура дозволяє організувати мережу, яка працює між двома сусідніми вузлами на відстані 400м зі швидкістю 500 Кбод. Функція повторювача, яку має кожний мережний пристрій сигналу, дозволяє

розширити систему до 13 км. Загальна кількість пристроїв мережі обмежена і складає максимум 256 вузлів.

Протокол HART (Highway Addressable Remote Transducer), розроблений фірмою Rosemount Inc. у середині 80-х років, реалізує відомий стандарт BELL 202 FSK, заснований на технології 4 – 20 мА.

Стандарт BELL 202 FSK - це кодування сигналу методом частотного зсуву для обміну даними на швидкості 1,2Кбод. Сигнал накладається на аналоговий вимірювальний сигнал 4—20 мА. Оскільки середнє значення сигналу дорівнює 0, то він не впливає на аналоговий сигнал 4-20 мА.

Довжина польової шини – 1,5 км, а кількість приладів на одній шині не перевищує 16. Схема взаємостосунків між вузлами мережі заснована на принципі Master/Slave. У HART-мережі може бути присутнім до 2 Master-вузлів (звичайно один). Другий Master, як правило, звільнений від підтримки циклів передачі і використовується для організації зв'язку з якою-небудь системою контролю/відображення даних. Стандартна топологія – «зірка», але можлива і шинна організація. Для передачі даних по мережі використовуються два режими:

- асинхронний - по схемі «Master-запит/ Slave-відповідь» (один цикл укладається в 500 мс);

- синхронний - пасивні вузли безперервно передають свої дані майстер-вузлу (час оновлення даних в майстер-вузлі - 250 - 300 мс).

HART-протокол через польову шину дозволяє проводити віддалене налаштування датчиків на необхідний діапазон вимірювання; не підводити до датчиків окремі лінії електроживлення і не мати в них блоків живлення (електроживлення реалізується від блоків живлення контролерів через польову шину); збільшити інформаційний потік між контролером і приладами; за наявності самодіагностування в пристроях автоматизації передавати повідомлення про несправності.

Протокол MODBUS від компанії Modicon вважається найбільш поширеним протоколом в управляючих промислових мережах. Протокол працює за принципом Master/Slave (провідний/підпорядкований). Конфігурація на основі цього протоколу передбачає наявність одного Master-вузла і до 247 Slave-вузлів. Тільки Master ініціює цикли обміну даними. Протокол Modbus може застосовуватися в дуплексних і напівдуплексних лініях сполучення. Фізичним рівнем протоколу Modbus є лінія стандарту RS422/RS485, проте, якщо використовується сполучення «точка-точка», той же формат команд може використовуватися на будь-якому послідовному асинхронному фізичному інтерфейсі, у тому числі RS232. Закінчення відповіді від підпорядкованого пристрою майстер визначає за часовим інтервалом між закінченням прийому попереднього байту і початком прийому наступного. Якщо цей інтервал більше часу необхідного для прийому двох байтів на заданій швидкості передачі, прийом кадру відповіді вважається завершаним. Кадри запиту і відповіді по протоколу Modbus мають фіксований формат.

Протокол PROFIBUS (PROcess FIeld BUS) розроблений фірмою Siemens на початку 90-х років. Стандарт протоколу описує рівні 1, 2 і 7 OSI-моделі. У PROFIBUS використовується гібридний метод доступу – Master/Slave і децентралізована процедура передачі маркера. У якості передаючого середовища використовується екранована вита пара, скляний або пластиковий оптоволоконний кабель. Довжина кабелю в сегменті шини залежить від швидкості передачі даних; вона складає 100 м при найбільшій швидкості передачі (12 Мбод) і 1000 м при якнайменшій (9,6 Кбод). Дальність мережі може бути збільшена за допомогою повторювачів або оптичних модулів зв'язку.

Мережа PROFIBUS може складатися зі 122 вузлів, з яких 32 можуть бути Master-вузлами. У середовищі Master-вузлів за схемою зростаючих номерів передається маркер, який надає право вузлам здійснювати на шині цикли читання або запису. Активний вузол одержує права доступу до шини на певний інтервал часу, після закінчення якого права доступу передаються наступному вузлу. Якщо активному вузлу (майстру) призначені пасивні вузли (підпорядковані), майстер обмінюється з ними даними поки володіє правами доступу.

При побудові багаторівневих систем автоматизації часто виникають завдання організації інформаційного обміну поміж рівнями. В одному випадку за необхідним є обмін комплексними повідомленнями на середніх швидкостях, в іншому - швидкий обмін короткими повідомленнями з використанням спрощеного протоколу обміну (рівень датчиків), у третьому потрібна робота на небезпечних дільницях виробництва (нафтогазові технології, хімічне виробництво).

Для всіх цих випадків PROFIBUS пропонує три окремі протоколи - PROFIBUS-FMS, PROFIBUS-DP і PROFIBUS-PA.

Протокол PROFIBUS-FMS з'явився першим і був призначений для роботи на так званому цеховому рівні, де потрібна висока ступінь функціональності, і де цей критерій більш важливіший ніж критерій швидкості. Основне його призначення - передача великих об'ємів даних.

У завданнях управління, що вимагають реального часу, на перше місце висувається такий параметр, як тривалість циклу шини. Реалізація протоколу PROFIBUS-DP дає збільшення продуктивності шини (наприклад, для передачі 512 біт даних, розподілених по 32 станціям, потрібний всього 6 мс). Протокол PROFIBUS-DP підтримується пристроями різних виробників.

Протокол PROFIBUS-PA - це розширення DP-протоколу, відповідно до якого технологія передачі інформації, заснована не на RS-485, а спирається на стандарт IEC1158-2, що визначає організацію передачі у вибухонебезпечному середовищі. Для комутації пристроїв потрібна всього одна вита пара, яка може одночасно використовуватися для інформаційного обміну і для підведення живлення до пристроїв польового рівня.

Мережа DH+ виробництва Allen-Bradley підтримує передачу даних і віддалене програмування контролерів. Зв'язок між процесорами і пристроями

здійснюється за одноранговим принципом, який характеризується відсутністю "майстра", можливістю будь-якого контролера ініціалізувати зв'язок з будь-яким вузлом, мінімальним мережним трафіком і простотою нарощування контролерів в мережі.

Магістральна лінія мережі DN+ має протяжність до 3048м, а відгалуження – до 30м. До однієї мережі DN+ можна приєднати до 64 пристроїв. Швидкість передачі даних залежить від довжини шини і може настраюватися від 57,6 Кбод (3048м) до 230,4 Кбод (750м).

Мережа Genius фірми GE Fanuc призначена для об'єднання в закінчену систему контролери GE Fanuc серій 90-70 і 90-30 та віддалені периферії Genius і Field Control. Взаємодія різних пристроїв з мережею Genius здійснюється за допомогою контролерів шини Genius (GBC), інтерфейсних модулів (GCM) і блоків інтерфейсу з шиною Genius (BUI). Фізично пристрої об'єднуються в мережу екранованою витю парою. Мережа має топологію "шина", до якої можуть бути підключені до 32 пристроїв. Максимальна довжина шини складає 2,3 км при швидкості обміну 38,4 Кбод. Максимальна швидкість передачі даних 153,6 Кбод досягається при довжині лінії до 600 м.

Протокол BITBUS розроблений фірмою INTEL у 1984 році для побудови розподілених систем, в яких потрібно було забезпечити високу швидкість передачі, детермінізм і надійність. Фізичний інтерфейс заснований на RS-485. Інформаційний обмін організований за принципом "запит - відповідь" (Master /Slave).

Протокол BITBUS визначає два режими передачі даних по шині:

- синхронний режим використовується при необхідності роботи на великій швидкості, але на обмежених відстанях. У цьому режимі до шини можна підключити не більше 28 вузлів, але довжина шини обмежується 30 м. Швидкість може бути від 500 Кбод до 2,4 Мбод. Синхронний режим передачі передбачає використання двох пар проводів (однієї пари - для даних, іншої - для синхронізації);

- режим з самосинхронізацією дозволяє значно продовжити шину. Стандартом визначено дві швидкості передачі – 375 Кбод (до 300м) і 62,5 Кбод (до 1200м). Використовуючи повторювачі, можна об'єднувати послідовно декілька шинних сегментів (до 28 вузлів на сегмент). Тоді спільне число вузлів можна довести до 250, а довжину спільної шини - до декількох кілометрів. При цьому режимі передачі також використовуються дві пари провідників (одна для даних, інша для управління повторювачем).

Протокол FIP (Factory Information Protocol) забезпечує високі швидкості передачі і конкретні інтервали оновлення даних. Протокол має гібридний централізований/децентралізований контроль за шиною, заснований на принципі широкого мовлення. Використання режиму широкого мовлення не потребує присвоєння кожному пристрою унікальної мережної адреси.

Кожен вузол на шині повністю автономний. Всі вузли мають можливість отримувати дані, що призначені для них. Контроль здійснюється з боку центрального вузла мережі, який називається арбітром.

FIP протокол підтримує рівні 1, 2 і 7 стандарту OSI. Для його передачі використовується вита пара або оптоволоконний кабель. Максимальна протяжність мережі - 1000м без повторювачів (до 15 км з оптичними повторювачами) при швидкості обміну інформацією – 1 Мбод. Мережа підтримує до 128 пристроїв.

Foundation Fieldbus — стандарт промислових мереж, який з'явився у 1995 році як результат зусиль консорціуму крупних, в основному північноамериканських виробників. По багатьох параметрах ця система схожа з PROFIBUS-PA - можливість установа у вибухонебезпечних зонах, передача інформаційного сигналу разом з напругою живлення по одній парі проводів, дворівнева ієрархія і т.ін. У Foundation Fieldbus на верхньому рівні використовується високошвидкісна магістраль Ethernet, а на нижньому — технологія передачі за стандартом IEC 61158-2, як і в PROFIBUS-PA. Дві особливості виділяють Foundation Fieldbus серед інших промислових мереж. По-перше, була розроблена спеціальна мова опису кінцевих пристроїв (Device Description Language), використання якої дозволяє підключати нові вузли до мережі за технологією plug-and-play. Достатньо фізично підключити новий пристрій, і він одразу самовизначиться на підставі закладеного опису DD (Device Description). По-друге, на відміну від інших промислових мереж, Foundation Fieldbus орієнтований на забезпечення однорангового зв'язку між вузлами без центрального провідного пристрою. Цей підхід дає можливість реалізувати системи управління, розподілені не тільки фізично, але і логічно, що у багатьох випадках дозволяє підвищити надійність і живучість АСУ ТП. У Foundation Fieldbus реалізовані найскладніші технології обміну інформацією – підписка на дані, режим «клієнт-сервер», синхронізація розподіленого процесу.

Промисловий (Industrial) Ethernet – це мережа для рівня управління підприємством і цехом у відкритій, незалежній від виробника системі зв'язку, яка сполучає комп'ютери і програмовані контролери, для застосування в індустріальній галузі. Industrial Ethernet придатний для швидкої передачі великих об'ємів даних. Через міжмережні шлюзи він надає можливість з'єднання абонентів різних мереж.

Фізичним електронним з'єднанням є коаксіальний кабель з подвійним екрануванням або промислова вита пара. Оптичне з'єднання виконується на основі скляного оптоволоконного кабелю. Дальність електронної мережі складає до 1,5 км, оптичної – до 4,5 км. Швидкість передачі даних встановлена на рівні 10 Мбод. Більше 1000 вузлів може бути об'єднано в мережу за допомогою промислового Ethernet. Перед здійсненням доступу до шини кожен вузол перевіряє, чи передає в даний момент інший вузол. Якщо це так, то вузол чекає протягом довільного інтервалу часу перед наступною спробою дістати доступ. Всі вузли мають рівні права доступу.

Основним чинником, що забезпечив Ethernet переможний шлях у сфері АСУ, є наявність величезного вибору сумісних між собою апаратних і програмних засобів побудови мереж цього стандарту. Перехід від концентраторів (hub) до комутаторів (switch) і від напівдуплексних каналів

зв'язку до дуплексних, дозволив зняти питання про можливість блокування обміну по мережному каналу через численні колізії інформаційних кадрів. Завдяки своїм «інтелектуальним» можливостям комутатор направляє одержаний інформаційний кадр тільки на те підключення, де реально знаходиться абонент, а не ширококомовно в усю мережу. В результаті загальний об'єм трафіку в мережі на багато разів скорочується. Фактично топологія «загальна шина» на логічному рівні трансформується в топологію «кожен з кожним», забезпечуючи гарантовану доставку даних. Крім того, однією з основних перешкод до застосування Ethernet в АСУ ТП завжди була невідповідність виготовлення апаратних засобів умовам їх застосування в промисловості. Зараз ситуація змінилася – з'явився цілий ряд концентраторів і комутаторів, що виконані відповідно до вимог промислових умов експлуатації. Це дозволяє протягнути надійну лінію Ethernet з верхнього (офісного) «поверху» АСУ ТП підприємства на нижній — в цех[3,13,14,15].

Порівняльна характеристика мереж і шин наведена у табл.2.1.

Таблиця 2.1 Порівняльна характеристика мереж і шин

| Мережа    | Топологія, метод доступу | Фізичне середовище       | Довжина         | Швидкість               | Вузли |
|-----------|--------------------------|--------------------------|-----------------|-------------------------|-------|
| Ethernet  | Шина, зірка              | Коакс.кабель             | 100 м           | 10 Мбод                 |       |
|           |                          | Вита пара<br>Оптоволокно | 2800 м          | 100 Мбод                |       |
| FIP       | Шина                     | Вита пара<br>Оптоволокно | 15 км           | 1 Мбод                  | 128   |
| Profibus  | Шина, М/S*, маркер       | Вита пара<br>Оптоволокно | 9600 м<br>90 км | 12 Мбод                 | 125   |
| Bitbus    | Шина, М/S*               | Дві пари проводів        | 300м<br>1200 м  | 375 Кбод<br>62,5 Кбод   | 28    |
| MODBUS    | Шина, М/S*               | Вита пара                | 1200 м          | 115,2 Кбод              | 254   |
| DH+       | Шина                     | Вита пара                | 3048 м          | 57,6 Кбод               | 64    |
|           |                          |                          | 750 м           | 230,8 Кбод              |       |
| Genius    | Шина                     | Вита пара<br>Оптоволокно | 600 м<br>2300 м | 153,6 Кбод<br>38,4 Кбод | 32    |
| DeviceNet | Шина                     | Вита пара                | 100 м<br>500 м  | 500 Кбод<br>125 Кбод    | 64    |
| Interbus  | Кільце, М/S*             | Вита пара<br>Оптоволокно | 12.8 км         | 500 Кбод                | 256   |
| ASI       | Шина, зірка, кільце, М/S | Вита пара                | 100м - сегмент  | -                       | 31    |
| SDS       | Шина<br>М/S              | Вита пара                | 450 м<br>30 м   | 125 Кбод<br>1 Мбод      | 64    |
| HART      | Шина, М/S                |                          | 1500 м          | 1,2 Кбод                | 16    |
| FF        | Шина, зірка              |                          |                 | До 1 Мбод               | 16    |

\* М/S - master/slave

## 2.6 Засоби програмування контролерів

При виборі контролерів для автоматизації одним з основних критеріїв є наявність програмних засобів, які дозволяють ефективно реалізувати прикладні задачі користувача. Ефективна реалізація задач передбачає, перш за все, швидку розробку програмних додатків для ПЛК, а також наявність готових рішень для стандартних функцій контролю та управління. Кращий результат відбувається, коли спеціаліст з автоматизації технологічних процесів може своє розуміння задачі втілити у життя без посередників – програмістів. Але відсутність або низьке розповсюдження придатних для цього мов суттєво стримувало застосування програмованих РС – сумісних контролерів. Користувач, як правило, віддавав перевагу спеціалізованому пристрою, який мав обмежені можливості, тільки за те, що для його використання не потрібні програмісти.

Тривалий час, фактично з самої своєї появи в кінці 70-років ХХ – століття, програмовані контролери були пристроями одного з трьох типів – логічний контролер для управління дискретними процесами, регулювальний контролер для керування неперервними процесами та вільно програмований контролер для універсального застосування. Як для логічних, так і для регулювальних контролерів мови програмування остаточно сформувалися до середині 80 – років і були повністю адекватні тим задачам, для яких вони застосовувалися. Проте для складних систем автоматизації, які характеризуються багатоконтурними функціональними зв'язками, задовільного рішення не було знайдено. Спроба введення міжнародного стандарту для програмування контролерів показала, що саме багатофункціональна мова описана найменш чітко. Щоб зрозуміти причину цієї ситуації розглянемо особливості основних мов програмування.

Перші технологічні контролери були переважно логічними тому відповідну специфіку мала одна з перших технологічних мов. Рішення було найпростішим і безпомилковим – імітувати проектування той техніки, замість якої використовувалися контролери. Так з'явилася мова релейно – контактних схем (РКС). Завдяки зведенню нової сутті до старої звичної форми був забезпечений психологічно безболісний перехід на нову техніку. РКС – мова мала ще одну перевагу – процедури вводу та налагодження програм легко реалізувалися на примітивних пультах з однорядковим дисплеєм і півтора десятками кнопок.

Успіх програмованих логічних контролерів невдовзі розділили одноканальні та багатоканальні програмовані цифрові регулятори. Для програмування була створена мова функціональних блоків, яка повторювала методику створення систем регулювання на технічній базі, що використовувалася раніше, коли окремі електронні або пневматичні функціональні блоки з'єднували між собою для отримання більш складних функціональних можливостей.

В Україні РКС - мова і мова функціональних блоків були застосовані при програмуванні відповідно МікроДАТів і Реміконтів.

Проте, як показав досвід, РКС – мова виявилася незручною для опису управління такими кроковими процесами, де послідовність кроків порушувалася розгалуженнями, паралелізмом і рециклами. Багатоканальні схеми регулювання, мовою функціональних блоків, були занадто громіздкі і невиразно представлені на примітивних пультах. Змішування цих мов вирішувало задачу при невеликому відхиленні від функціонального призначення мови, але було безсилим у випадку їх дійсно багатфункціонального застосування. Тому почались пошуки, які дали декілька варіантів для розв’язання проблеми.

Перший успіх принесли спроби організації загальної структури програми. Такий підхід дозволив зробити декомпозицію задачі за рахунок введення модульності програм і їх ієрархічного представлення. Вдалим прикладом цього підходу вважається створення французькою фірмою “Telemechaniqu” мови Графсет, яка стала зразком для розробки подібних мов іншими фірмами. Основні властивості мови Графсет, яка практично застосувала розроблені в теорії автоматів мережі Петрі, це представлення програми у вигляді кроків і переходів, а також наявності декількох одночасно працюючих програм і організуючої програми, яка керує їх роботою. Причому, логіка виконання кроків і переходів в багатьох випадках може бути описана РКС – мовою. Наявність кроків, розгалужень і циклів, разом з структуризацією задачі та використанням РКС – мови забезпечили Графсет-мові достатню універсальність і можливість використання її для розв’язання задач управління більшістю дискретних і періодичних процесів.

Необхідність поєднання неперервного та дискретного управління на значній кількості промислових об’єктів спонукала створення універсальної технологічної мови. Початком шляху до неї стала спроба адаптації універсальних мов програмування. У перших вільно програмованих контролерах застосовувався асемблер, потім для програмування контролерів використовувалися мови високого рівня (Си, Бейсик), проте для програмування цими мовами запрошувалися професійні програмісти, а сам процес розробки програм був достатньо трудомістким. Використання пультів для програмування виключалося складністю мов програмування, а також великими ресурсами, необхідними для компіляції та рекомпіляції програм у випадку внесення змін.

З іншого боку стрімкий розвиток персональних комп’ютерів показав, що кращим пультом програмування для будь-якого контролера є звичайний офісний notebook. Тому обмеження на складність мови програмування були зняті й стало можливим здійснити перехід від програм – інтерпретаторів до компіляторів, що працюють на інструментальній машині і створюють код завантаження для виконання в контролері.

Принцип адаптації універсальних мов програмування був простим – в мову вводилися нові типи змінних – “входи” і “виходи” контролера, а також створювалася певна бібліотека підпрограм, що реалізують найбільш популярні алгоритми. Програмування спростилося, але ще було складним для звичайного технолога або “автоматчика – киповця”. І тоді був зроблений наступний крок –



створення на базі розповсюджених мов програмування процедурного типу спеціалізованих технологічних мов, орієнтованих саме на кінцевого користувача на об'єкті. Основною різницею текстових технологічних мов від універсальних стало, поперше, різьке спрощення синтаксису та семантики (скорочення типів операторів, складності виразів та ін.), а подруге, введення в мову спеціальних технологічних понять, які реалізують типові функції контролю та управління. У зв'язку з цим відкрився великий простір для творчості і кількість мов такого типу на першому етапі, майже, співпала з кількістю фірм, які бажали їх застосовувати.

Щоб уніфікувати численні технологічні мови, Міжнародний електротехнічний комітет МЕК (ІЕС) у 1993р опублікував стандарт ІЕС61131-3, який визначив п'ять мов програмування – LD (крокових діаграм), FBD (функціональних блокових діаграм), SFC (послідовних функціональних схем), ST (структурованого тексту) і PL(список інструкцій).

Досвід роботи з контролерами показав, що вибір мов програмування частіше за все визначається особистою схильністю користувачів і мало залежить від об'єкта автоматизації. Дійсно мови, що представлені у стандарті в більшості випадків взаємозамінні. Це означає, що при різному рівні підготовленості в області чистого програмування користувачі можуть створювати програми рівної функціональності. За різними оцінками, біля 80% ПЛК– ринку обслуговується програмними продуктами, які реалізують в той чи іншій мірі стандарт ІЕС-61131-3. Всі основні виробники інструментальних програмних систем для промислової автоматики прийняли нові правила діяльності. У списку інструментальних програмних систем, що реалізують цей стандарт, більш ніж два десятки найменованій. Серед них системи CoDeSys (Німеччина), PUMA (Австрія), PDS7 (Нідерланди), SELECONTROL (Швейцарія), ISaGRAF (Франція), MULTIPROG (Німеччина)[1,3,16,17,18].

## **Тестові та контрольні запитання до розділу 2**

**І-й рівень запитань дозволяє перевірити загальний рівень засвоєння матеріалу розділу і передбачає відповідь на запитання або твердження “Так” чи “Ні”.**

1. Чи відносяться пристрої EPROM, EEPROM і Flash до класу незалежної перезаписуваної пам'яті?
2. Чи правомірно флеш-пам'ять називати "флеш-дискон"?
3. Сімейства малих контролерів здатні підтримувати максимум декілька десятків каналів вводу-виводу.
4. Чи можливо в контролерах здійснювати заміну модулів з приєднаними зовнішніми проводами?
5. При віддаленому ввіді-виводі обробляється значно менша кількість параметрів ніж при розподіленому.

6. Мережі, що забезпечують інформаційний обмін між контролерами, датчиками і виконавчими пристроями називаються технічними мережами.
7. Швидкість передачі інформації вимірюється у Бодах.
8. Протокол PROFIBUS-PA призначений для організації передачі інформації у вибухонебезпечному середовищі.
9. Відкрита комунікаційна мережа DeviceNet призначена для приєднання контролерів до EOM.
10. Промисловий Ethernet дозволяє об'єднувати у мережу більше 1000 вузлів.
11. Мови ST і LD є текстовими?
12. EEPROM в порівнянні з EPROM має збільшений ресурс роботи, простіша у використанні, але характеризується більш високою вартістю.
13. Модулі аналогового вводу можуть працювати з сигналами 0-5В, 0-10В, 0-50В, 0-100В.
14. При локальному вводі-виводі модулі розміщуються на центральному і виносному шасі.
15. Загальна протяжність з'єднувальних кабелів між панелями розширення не повинна перевищувати 150 м.
16. Мережі на рівні управління підприємством називається управляючими промисловими мережами.
17. Передача інформації напівдуплексним методом здійснюється тільки в одному напрямку, а дуплексним – в обох.
18. У режимі «клієнт-сервер» провідний пристрій (master), послідовно опитує підпорядковані вузли (slave).
19. DeviceNet – відкрита комунікаційна мережа нижнього рівня.
20. Протокол MODBUS від компанії Modicon вважається найбільш поширеним протоколом в інформаційних мережах.

*Перевірте відповіді на сторінках: 1 - 33 с., 2 - 34 с., 3 - 34 с., 4 - 35 с., 5 - 36с., 6 - 37 с., 7 - 40 с., 8 - 43 с., 9 - 41 с., 10 - 45 с., 11 - 49 с., 12 - 33 с., 13 - 35 с., 14 - 35 с., 15 - 36 с., 16 - 37 с., 17 - 39 с., 18 - 39 с., 19 - 41 с., 20 - 42 с.*

**II-й рівень запитань (тестові запитання) дозволяє перевірити рівень засвоєння матеріалу розділу із заглибленням у технічні особливості будови та структури контролерів та передбачає вибір правильної відповіді з наведених трьох варіантів.**

### **1. Чим відрізняються SRAM і DRAM?**

- а) об'ємом інформації, що зберігається, та ціною;
  - б) елементом зберігання інформації; часом доступу до інформації;
- об'ємом інформації, що зберігається, ціною;
- в) часом доступу до інформації та об'ємом інформації, що зберігається.

- 2. В якому запам'ятовуючому пристрої вміст пам'яті стирається опромінюванням чипа ультрафіолетовим або рентгенівським промінням?**
- а) EEPROM;
  - б) EPROM;
  - в) EEPROM та EPROM.
- 3. Які сигнали можуть отримувати модулі дискретного вводу-виводу?**
- а) напругу 12, 24, 48 В постійного струму або 120 і 240 В змінного струму;
  - б) напругу 10 В постійного струму або 240 В змінного струму;
  - в) напругу 12, 24, 48 В постійного струму.
- 4. На яку відстань можуть бути віднесені шасі розширення з модулями вводу-виводу?**
- а) максимум на декілька десятків метрів від центрального процесора;
  - б) не більше ніж на десять метрів від центрального процесора;
  - в) не більше ніж на сотню метрів від центрального процесора.
- 5. Скільки рівнів має відкритий протокол сполучень?**
- а) шість рівнів;
  - б) сім рівнів;
  - в) вісім рівнів.
- 6. На якому рівні відкритого протоколу забезпечується передача даних між двома пристроями?**
- а) транспортному;
  - б) мережному;
  - в) каналному.
- 7. Які методи забезпечують передачу інформації по каналах в обох напрямках?**
- а) дуплексний та напівдуплексний;
  - б) дуплексний та симплексний;
  - в) симплексний та напівдуплексний.
- 8. Які характеристики AS-інтерфейсу дозволяють використовувати його у розподілених АСУ ТП реального часу?**
- а) максимальний час циклу опитування складає 5-10с; об'єм даних, що передається з одного ASI-вузла складає не більше 4 байтів;
  - б) максимальний час циклу опитування складає 5-10мс; об'єм даних, що передається з одного ASI-вузла складає не більше 4 байтів;
  - в) максимальний час циклу опитування складає 5-10мс; об'єм даних, що передається з одного ASI-вузла складає не більше 4 бітів.
- 9. Який протокол реалізує стандарт заснований на технології 4 – 20 мА?**
- а) CAN;
  - б) MODBUS;
  - в) HART.
- 10. В чому особливість EPROM?**
- а) дана пам'ять не є перезаписуваною;
  - б) перед записом у пам'ять нової інформації необхідно примусово стерти стару;

в) при записі у пам'ять нової інформації стара стирається повністю автоматично.

**11. Що дозволяє здійснювати заміну модулів без демонтажу зовнішніх проводів?**

- а) наявність пружинних клемників;
- б) наявність знімних клемників;
- в) відсутність кодування модулів.

**12. В яких одиницях вимірюється швидкість передачі інформації?**

- а) м/с;
- б) біт/с;
- в) байт/с.

**13. Який протокол PROFIBUS призначений для організації передачі інформації у вибухонебезпечному середовищі?**

- а) PROFIBUS-FMS;
- б) PROFIBUS-DP;
- в) PROFIBUS-PA.

**14. Контролери якої фірми об'єднує мережа Genius?**

- а) GE Fanuc;
- б) Siemens;
- в) Schneider Electric.

**15. Якою може бути довжина електронних та оптичних мереж?**

- а) до 1,5 км і до 4,5 км;
- б) 15м і 1200м;
- в) до 1,5км і вище 4,5км.

**16. Які мережі підтримують найбільшу кількість вузлів?**

- а) Modbus і Profibus;
- б) Interbus і Modbus;
- в) DH+ і ASI.

**17. Який з розділів стандарту IEC 6113 визначає мови програмування контролерів?**

- а) 5;
- б) 1;
- в) 3.

**18. Які з наведених мов програмування – LD, FBD, SFC, ST або IL є текстовими?**

- а) LD, ST;
- б) SFC, IL;
- в) IL, ST.

**19. Чим характеризуються центральні процесори контролерів?**

- а) кількістю шасі, кількістю входів-виходів, наявністю функції ПД-регулювання та спеціальних модулів, кількістю і типами мереж, підтримкою математики з плаваючою комою;
- б) мовою програмування та об'ємом пам'яті;

в) кількістю шасі, кількістю входів-виходів, наявністю функції ПД-регулювання та спеціальних модулів, мовою програмування.

**20. В яких системах використовується віддалений ввід-вивід?**

- а) в системах, де велика кількість датчиків і інших польових пристроїв знаходиться на відстані 1000 і більше метрів від центрального процесора;
- б) в системах, що використовують моноблочні контролери;
- в) коли реалізується розподілена конфігурація ввідів-виводів.

*Перевірте відповіді на сторінках: 1 - 32 с., 2 - 33 с., 3 - 35 с., 4 - 36 с., 5 - 38 с., 6 - 38 с., 7 - 39 с., 8 - 40 с., 9 - 42 с., 10 - с., 11 - 35 с., 12 - 40 с., 13 - 43 с., 14 - 44 с., 15 - 45 с., 16 - 46 с., 17 - 49 с., 18 - 49 с., 19 - 32 с., 20 - 36 с.*

**III-й рівень запитань (контрольні запитання) дозволяє перевірити рівень засвоєння матеріалу із систематизацією отриманої у розділі інформації та передбачає відповіді, що побудовані на логічному аналізі або порівнянні із приведенням конкретних технічних параметрів (даних або прикладів).**

1. Які фактори впливають на вибір способу і засобів передачі інформації у мережах ПЛК?
2. Чим визначається сфера застосування того чи іншого з видів контролерів?
3. Перерахуйте програмні засоби для ПЛК та мови програмування у порядку їх появи за часом.
4. За якими характеристиками здійснюється вибір контролерів для побудови систем управління?
5. Дайте порівняльну характеристику можливостей протоколів та їх особливості при роботі з контролерами.
6. У чому відмінність існуючих типів ПЗП?
7. Чим визначається вибір конфігурації вводу-виводу?
8. За яким принципом поділяються інформаційні мережі по рівням автоматизованих систем?
9. Проаналізуйте три основних режими обміну даними у промислових мережах.
10. Дайте характеристику інтерфейсів і протоколів мережі шинного рівня.

*Перевірте відповіді на сторінках: 1 – 37...46 с., 2 - 34-40 с., 3 – 47...49 с., 4 – 32...49 с., 5 – 37...46 с., 6 – 32...34 с., 7 – 35...37 с., 8 - 37 с., 9 – 39...40 с., 10 – 37...46 с.*

## 3 МІКРОПРОЦЕСОРНИЙ КОНТРОЛЕР РЕМІКОНТ Р-130

### 3.1 Загальна характеристика та фізична структура

Реміконт Р-130 – це розробка російських та українських спеціалістів ще радянських часів. На момент створення цей контролер нічим не поступався кращим закордонним аналогам, а по деяких позиціях був навіть кращим. Як підтвердження цього – використання Реміконтів в сучасних АСУ ТП поряд з виробами відомих закордонних виробників.

Компактний малоканалний багатофункціональний контролер Р-130, призначений для автоматичного регулювання та логічного керування технологічними процесами. Існує три моделі Р-130 – регулююча, логіко-командна та неперервно-дискретна.

Регулююча модель контролера дозволяє реалізовувати за ПД-законом до чотирьох незалежних або взаємозв'язаних контурів автоматичного регулювання.

Кожна з систем автоматизації може бути локальною або каскадною, мати ручний, програмний або зовнішній (супервізорний) задавач, аналоговий вихід для роботи з позиціонером або імпульсний вихід для роботи з виконавчим механізмом сталої швидкості. До 15 контролерів можуть бути об'єднані у локальну кільцеву мережу “Транзит”.

У складі комплексу технічних засобів Реміконт Р-130 є блок мікропроцесорного контролера БК-1 з лицевою панеллю, рис. 3.1, розрахований на щитовий монтаж, пульт настроювання ПН-1, блок живлення БП-1, засоби зв'язку з об'єктом – БУТ-10, БУС-10, БУМ-10, БПР-10, РН-1, а також клемно-блокові з'єднувачі КБС-0, КБС-1, КБС-2, КБС-3.



Рисунок 3.1 – Основний блок контролера БК-1

Блок контролера БК-1 розрахований на щитовий монтаж, має габаритні розміри 160×80×365мм (В×Ш×Г) і виконує наступні функції:

- перетворення аналогових і дискретних вхідних сигналів у цифрову форму;
- перетворення сигналів у цифровій формі в аналогові та дискретні вихідні сигнали;
- обробку інформації відповідно до вибраних алгоритмів керування;
- обробку дискретних аварійних сигналів;
- прийом і передачу сигналів через інтерфейсний канал цифрового зв'язку;
- оперативного контролю і керування за допомогою індикаторів і клавіш на лицевій панелі блока.

Панель лицева ПЛ блока контролера призначена для взаємодії оператора-технолога з Реміконтом Р-130 в процесі регулювання. ПЛ містить клавіатуру для подачі команд оператора та засоби індикації, які дозволяють оператору стежити за ходом регулювання.

У верхній частині лицевій панелі, рис.3.2, розташовані п'ять одиничних індикаторів, що контролюють помилки.



Рисунок 3.2 – Лицева панель ПЛ-1 регулювальної моделі

Однорозрядний цифровий індикатор (ЦІ) “Контур” показує номер контуру, з яким працює оператор. На 4-х розрядному ЦІ “Задание” показується сигнал завдання, а на третьому ЦІ “Контроль” висвічується значення контрольованого параметра.

Сім одиничних індикаторів у групі “Контрольовані параметри ” показують, яка інформація виводиться на ЦІ контрольованих параметрів:

Vx – вхідний сигнал (регульований параметр);

$\varepsilon$  – сигнал непогодження;

Vих – вихідний сигнал (керувальний вплив), %;

Z – довільний сигнал, призначення якого програмується;

Nn – номер програми, номер поточної частки програми;

$\tau_n$  – час, що залишився до закінчення поточної частки програми;

OK – помилки контуру, показується номер сигналу, що вийшов за допустимі значення.

Шкальний індикатор “Вихідний сигнал”, показує значення аналогового керувального сигналу на виході контуру в інтервалі від 0 до 100%, а два одиничних індикатори “▼” і “▲” у цій групі сигналізують при спрацьовуванні імпульсного регулятора у бік “менше” або “більше”.

Чотири одиничних індикатори “Режим керування” показують режим роботи контуру:

КУ – каскадне керування;

ЛУ – локальне керування;

ДУ – дистанційне керування;

РУ – ручне керування.

Три одиничних індикатори “Вид завдання” показують вид встановленого завдання у контурі:

ВЗ – зовнішнє завдання;

ПЗ – програмне завдання;

РЗ – ручне завдання.

Лицева панель має 12 клавiш для оперативного керування контурами регулювання:


“←” “→” – клавiші вибору параметра контролю;


“Δ” “∇” – клавiші зміни параметрів або завдання, “більше”, “менше”;


“N” – номер контуру;

“↶” – клавiша переходу на зовнішнє завдання;

“⊙” – клавiша переходу на один з режимів САР (локальний, каскадний, дистанційний), разом з клавiшами “↑”, “↓”, “▲”, “▼”;

“” – програмний задавач; разом з “▲”, “▼” змінюється завдання або номер програми;

“” – ручний задавач; разом з “▲” або “▼” змінюється завдання;

“” – режим ручного керування.

Пульт настроювання ПН-1 призначений для технологічного програмування, настроювання та контролю Реміонта Р-130. Пульт являє собою портативний блок калькуляторного типу, який приєднується до блока контролера тільки під час програмування або налагодження.



Двоканальний блок підсилювачів низького рівня БУТ-10 здійснює перетворення сигналів термопар у струмовий сигнал 0-5 мА та компенсацію термо-е.р.с. вільних кінців термопари.

Двоканальний блок підсилювачів сигналів резистивних датчиків БУС-10 перетворює зміну опору резистивних датчиків у струмовий сигнал 0-5 мА. БУС-10 застосовується лише у випадках, коли відсутні нормувальні перетворювачі.

Двоканальний блок підсилювача потужності БУМ-10 містить чотири сильнострумових герконових реле, контакти яких можуть комутувати постійну або змінну напругу. Обмотки реле вмикаються до дискретних виходів блока БК-1, при цьому для живлення реле необхідна зовнішня напруга 24 В, наприклад, від блоку БП-1.

Блок перемикання БПР-10 містить 8 перемикальних реле, обмотки яких вмикаються до виходів БК-1 через джерело живлення напругою 24 в БП-1. Контакти реле можуть використовуватися для вмикання і вимикання кіл автоматики.

Резистор нормувальний РН-1 перетворює сигнали 0-5 мА, 0(4)-20 мА і 0-10В у сигнал 0-2В, який є вхідним аналоговим сигналом блока БК-1. РН-1 застосовується лише у тому випадку, коли у складі Реміонта Р-130 немає клемно-блокових з'єднувачів КБС-3, які мають ті ж самі РН-1.

Клемно-блокові з'єднувачі КБС-0, КБС-1, КБС-2, КБС-3 призначені для підключення датчиків, засобів зв'язку з об'єктом і виконавчих пристроїв. Ці з'єднувачі являють собою, крім КБС-0, відрізок кабелю, на одному кінці якого змонтована вилка розніму, а на другому – клемна колодка. КБС-0 – використовується для розмноження клем у тих випадках, коли до однієї клеми повинні підключатися більше двох проводів. КБС-0 містить 8 вільних клем “під гвинт”. Конструкція КБС-1 аналогічна клемно-блоковому з'єднувачу КБС-0, але має кабель з рознімом. З'єднувачі КБС-2 і КБС-3 мають 3 колодки по 8 клем кожна. КБС-2 застосовують для підключення аналогових і дискретних кіл. КБС-3 застосовується для підключення аналогових кіл. З'єднувачі КБС застосовують у випадках, коли необхідно приєднати прилад з клемною колодкою до приладу з рознімом.

Блок БП-1 призначений для живлення БК-1, пристроїв зв'язку з об'єктом і вхідних кіл контролера. Блок живлення виконує ще одну дуже важливу функцію – через нього блок контролера приєднуються до мережі “Транзит”.

Для приєднання блока БК-1 до блока живлення БП1 використовується міжблоковий з'єднувач МБС, який являє собою відрізок кабелю, що має з обох сторін вилки розніму.

На рис.3.3 показана загальна схема приєднання до блока контролера різних технічних засобів, які є джерелами або приймачами вхідних і вихідних сигналів.

Р-130 – виріб, що проектно-компонується. У нього є базовий комплект апаратури, який поставляється завжди і комплект, що компонується під задану задачу.

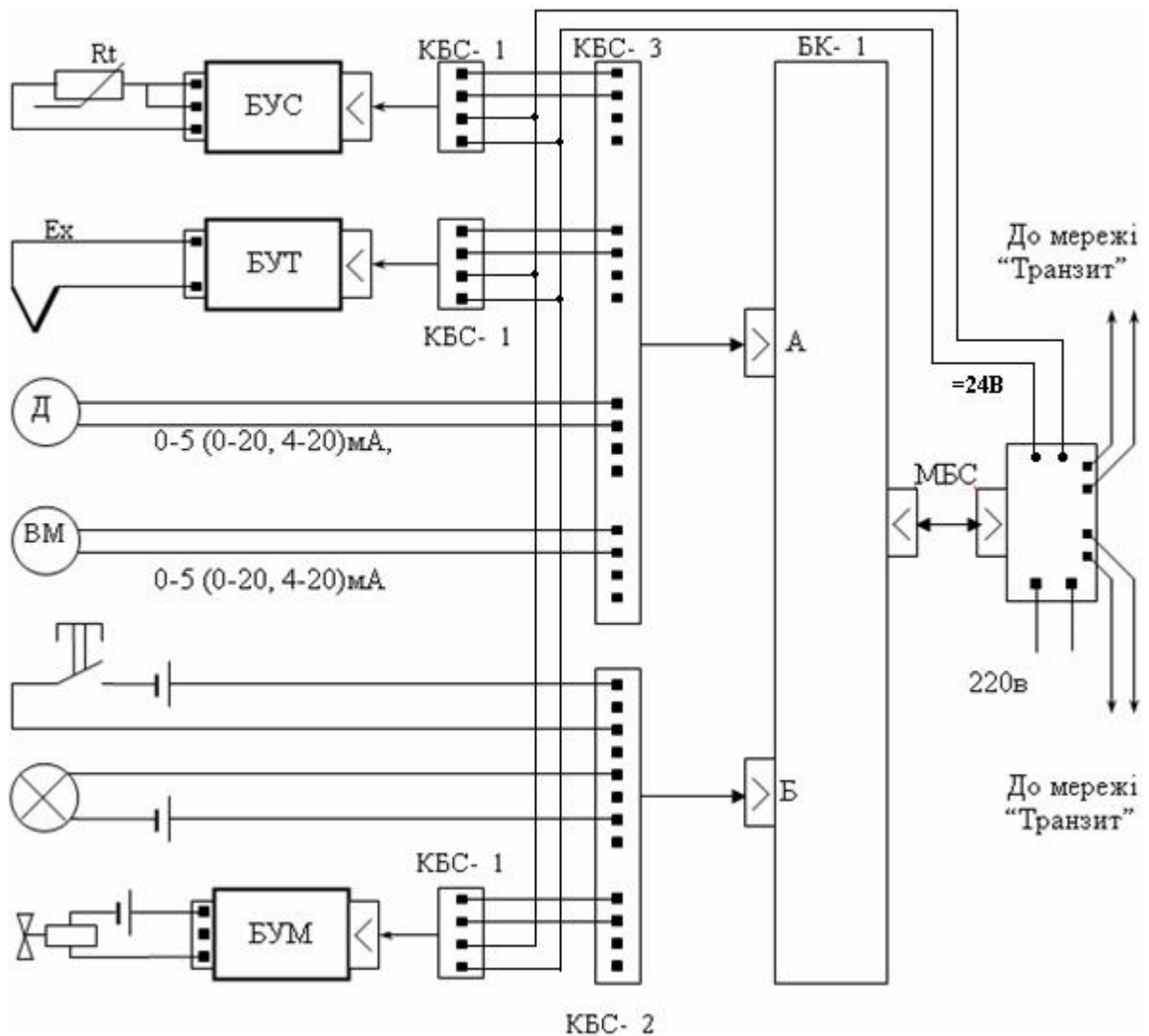


Рисунок 3.3 – Узагальнена схема приєднання засобів автоматизації до Р -130

Базова 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 установлені один або два з трьох модифікацій модулів ПЗО:

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

Модуль аналогових сигналів МАС виконує функції АЦП і ЦАП. Модуль аналогових і дискретних сигналів МДА виконує функції перетворення вхідних аналогових сигналів постійного струму на цифровий двійковий код – АЦП і цифровий код на дискретний вихідний сигнал – ЦДП.

Модуль дискретних сигналів МСД виконує функції перетворення вхідних дискретних сигналів на цифровий двійковий код – ДЦП і цифровий код на дискретний вихідний сигнал ЦДП.

Існує 30 модифікацій контролера, які відрізняються номенклатурою входів-виходів модулів ПЗО.

Модуль ПРЦ10 призначений для організації обміну даними між модулями Р-130, а також для логічної та арифметичної обробки цих даних відповідно до заданої програми. Модуль має безпосередній зв'язок з акумулятором, завдяки чому зберігається вміст ОЗП при вимкненні напруги.

Модуль контролю та програмування МКП є зв'язувальним ланцюгом між процесором і “панеллю оператора” у якості оперативного запам'ятовуючого пристрою ОЗП.

Модуль стабілізованої напруги МСН-10 забезпечує напругою блок контролера БК-1 і пульт настройки ПН-1[3,19].

### **3.2 Віртуальна структура Р-130**

Якщо фізична структура Р-130 пов'язана зі складом його фізичних елементів (блоків, модулів та ін.), то віртуальна – характеризує властивості контролера, як ланки системи керування. Фізична структура “перетворюється” у віртуальну за допомогою внутрішнього програмного забезпечення, “зашитого” у ПЗП

У загальному вигляді віртуальна структура Р-130 складається з апаратури вводу-виводу інформації, апаратури оперативного керування та настроювання, апаратури інтерфейсного каналу, а також алгоритмічних блоків і бібліотеки алгоритмів.

Апаратура вводу перетворює аналогові та дискретні сигнали, що надходять до входу контролера, у цифрову форму. Апаратура виводу здійснює зворотне перетворення.

Апаратура оперативного керування (лицева панель) призначена для оператора-технолога, який за допомогою клавіш, світлодіодних і цифрових індикаторів керує технологічним процесом – контролює його параметри, змінює режими керування та уставки, пускає, зупиняє або скидає програму і т. ін.

За допомогою пульта настроювання оператор-наладник програмує контролер, здійснює настроювання його параметрів, а також контролює сигнали у внутрішніх точках віртуальної структури.

Контролер має один інтерфейсний канал. Цей канал має пристрій, що перетворює вхідний потік послідовних біт інформації на цифрову інформацію у вигляді байтів (тобто перетворює послідовний код на паралельний), а також здійснює зворотне перетворення.

Основу віртуальної структури контролера складають алгоблоки, в які оператор розміщує з бібліотеки будь-які алгоритми, сполучає їх між собою і встановлює необхідні параметри настроювання. Всього в контролері 99 алгоблоків, а в бібліотеці 76 алгоритмів. Усе програмне забезпечення, що

формує віртуальну структуру, “зашиито” у ПЗП. Конфігурування віртуальної структури виконуються без участі професійних програмістів за допомогою спеціалізованого малогабаритного пульта ПН-1.

Алгоблоки реалізовані програмно. Вони створюють область управління контролера. У вихідному стані алгоблоки відсутні, тому ніякі функції, що до опрацювання сигналів, контролером не виконуються. Алгоблок “з’являється” після того, як у процесі технологічного програмування в ОЗП розташовується будь-який алгоритм з бібліотеки. У цьому випадку алгоблок – це алгоритм, що використовується, тому поняття алгоблока і поняття алгоритму ідентичні.

Бібліотека контролера містить алгоритми, за допомогою яких можна розв’язувати досить складні задачі автоматичного регулювання та логіко-програмного керування. Крім алгоритмів автоматичного регулювання та логіко-програмного керування у бібліотеці є набір алгоритмів, що виконують статичні, математичні, логічні та аналогово-дискретні перетворення сигналів. Кожний алгоритм має бібліотечний номер, шифр і найменування.

Частка бібліотечних алгоритмів здійснює особливу задачу – вона сполучає апаратуру контролера з основною кількістю функціональних алгоритмів. До цих алгоритмів відносяться алгоритми вводу та виводу аналогових і дискретних сигналів, алгоритми обслуговування лицевої панелі та алгоритми прийому і передачі сигналів через інтерфейсний канал. Апаратні елементи віртуальної структури починають виконувати свої функції лише після того, як до будь-якого алгоблока буде розміщено алгоритм, що зв’язує їх з основною кількістю функціональних алгоритмів.

У загальному випадку алгоритм має власні входи, виходи та функціональне ядро.

Входи алгоритму поділяються на дві групи – сигнальні та настроювальні. Сигнали, що надходять на сигнальні входи алгоритму, обробляються їм відповідно до його призначення, а сигнали, що надходять до настроювальних входів алгоритму, встановлюють його параметри настройки.

На виході алгоритму формуються сигнали, які є результатом обробки алгоритмом вхідних сигналів.

Кількість входів і виходів не фіксована і визначається його видом. Ні в одному алгоритмі кількість входів не перевищує 99, а кількість виходів – 25.

В окремому випадку в алгоритмі можуть бути відсутні входи або виходи.

Окрім входів-виходів, що можна вільно конфігурувати, деякі алгоритми мають входи-виходи спеціального призначення, які не доступні для конфігурування. До таких алгоритмів відносяться алгоритми вводу-виводу, прийому і передачі, оперативного керування.

Якщо один з таких алгоритмів розміщується у будь-який алгоблок, його неявні входи або виходи автоматично з’єднуються з апаратурою, яку має обслуговувати цей алгоритм. Наприклад, коли у будь-який алгоблок розміщується алгоритм аналогового вводу групи А, неявні входи цього алгоритму автоматично з’єднуються з АЦП, що обробляє сигнали групи А. На виходах цього алгоритму мають бути сформовані “загальнодоступні” сигнали,

що еквівалентні сигналам на аналогових входах контролера групи А. Тому, коли на вхід будь-якого функціонального алгоритму потрібно подати аналоговий сигнал, цей вхід при програмуванні необхідно з'єднати з відповідним виходом алгоритму аналогового вводу.

Деякі алгоритми мають обмеження на кратність їх використання у межах одного контролера. Наприклад, алгоритм аналогового вводу групи А можна використовувати лише один раз, оскільки він охоплює усі аналогові входи групи А. Такі самі обмеження мають інші алгоритми вводу-виводу інформації.

У загальному випадку бібліотечний алгоритм має три реквізити (параметри): бібліотечний номер, модифікатор, масштаб часу.

Бібліотечний номер – це двозначне десяткове число, яке є основним параметром, що характеризує властивості алгоритму.

Модифікатор задає додаткові властивості алгоритму. Звичайно – це кількість однотипних операцій, які може виконувати один алгоритм. Наприклад, у суматорі модифікатор задає кількість сумових входів, у програмному задавачі – кількість ділянок програми і т. ін.

В окремих випадках модифікатор задає набір певних параметрів. Наприклад, в алгоритмі оперативного контролю ОКО модифікатор задає параметри контуру – локальний контур чи каскадний, з аналоговим чи імпульсним регулятором і т. ін. Деякі алгоритми (інтегрування, множення та ін.) модифікатора не мають.

Масштаб часу є тільки в алгоритмах, робота яких пов'язана з реальним часом, наприклад, таких як регулювання, програмний задавач, таймер та ін. Масштаб часу задає одну з двох розмірностей для часових сигналів або параметрів. Якщо контролер настроєний на молодший діапазон, то масштаб часу індивідуально у кожному алгоблоці задає масштаб – “секунди” або “хвилини”. Для старшого діапазону масштаб часу задає “хвилини” або “години”.

Наявність модифікатора та масштабу часу суттєво поширює можливості алгоритмів. Наприклад, в одному алгоблоці може розташовуватися програмний задавач, який має декілька ділянок програми протяжністю декілька секунд, в той час як в іншому алгоблоці може працювати програмний задавач з декількома десятками ділянок і з протяжністю кожної ділянки в декілька сотень хвилин.

Коли алгоритми розміщуються по алгоблоках, у більшості випадках діють два правила:

- будь-який алгоритм можна вміщувати у будь-який (за номером) алгоблок;
- один і той алгоритм можна вміщувати у різні алгоблоки, тобто використовувати багаторазово.

З цих правил є винятки. Алгоритм ОКО, що формує інформацію, яка надходить на індикатори лицевій панелі, розміщується тільки у перших чотирьох блоках. Номер алгоблока, у якому розміщений алгоритм ОКО, визначає номер контуру, який обслуговуються даним алгоритмом. Коли

кількість контурів або програм менше за чотири, у вільних алгоблоках цієї групи можна розміщувати будь-які інші алгоритми, проте алгоритм ОКО має бути розміщений у молодших за номером алгоблоках один за одним (тобто між ними не можуть знаходитися інші алгоритми, за винятком алгоритму з кодом ОО, який резервує місце для можливого використання алгоритма ОКО в майбутньому).

Кожний вхід алгоблока, незалежно від того, який алгоритм у ньому, може знаходитися в одному з двох станів – зв'язаному або вільному.

Вхід зв'язаний, коли він з'єднаний з виходом якого-небудь алгоблока, в іншому випадку вхід вільний.

Сигнали на зв'язані входи надходять з виходів тих алгоблоків, з якими ці входи зв'язані. На вільних входах сигнали можуть установлюватися оператором вручну у процесі настроювання.

Сигнали на вільних входах можуть бути у якості констант або коефіцієнтів. Константи можна установлювати та змінювати у режимі програмування, коефіцієнти можна установлювати та змінювати як у режимі програмування, так і у режимі роботи, тобто не вимикаючи контролер з контуру керування.

Можливості конфігурування не залежать від алгоритму, що знаходиться у алгоблоці, а визначаються такими правилами:

- будь-який вхід будь-якого алгоблока можна зв'язувати з будь-яким виходом алгоблока або залишати вільним;
- на будь-якому вільному вході будь-якого алгоблока можна вручну задавати сигнал у вигляді константи або коефіцієнта;
- на будь-якому вході будь-якого алгоблока сигнал можна інвертувати.

Обслуговування алгоблоків здійснюється циклічно зі сталим часом циклу. Спочатку обслуговується перший алгоблок, потім – другий і т.д. до останнього. Коли час у межах установленого часу циклу закінчиться, програма знову перейде до обслуговування першого алгоблока.

Час циклу вибирається при програмуванні контролера і може змінюватися у діапазоні 0,2-2с. Час обслуговування усіх запрограмованих алгоблоків має бути менше установленого часу циклу. Залишок часу використовується для самодіагностування контролера.

Окрім часу на обслуговування алгоблоків і на самодіагностування, потрібен також певний час на приймання, передавання та оброблення повідомлень через інтерфейсний канал. Загальний час на обслуговування алгоблоків та інтерфейсного каналу має бути меншим за час циклу.

Мова програмування контролерів Р-130 не процедурна, тобто не задається послідовність виконання операцій. При програмуванні створюється віртуальна структура, що являє собою структурну схему системи автоматичного управління.

Програмування Реміонта Р-130, як процес створення віртуальної структури, здійснюється з виносної панелі настроювання ПН-1, що з'єднується з блоком контролера через рознім [19].

### 3.3 Мережна структура Р-130

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

Порядок розташування контролерів у мережі не має значення. Контролери, що входять до мережі "Транзит", можуть обмінюватися інформацією. Таким чином створюється розподілена система керування на базі сумісно працюючих контролерів. Це підвищує надійність системи в цілому, оскільки в разі виходу з ладу одного з контролерів система продовжує зберігати свою працездатність. Максимальна кількість контролерів, приєднаних до мережі "Транзит", не може перевищувати 15.

Мережа "Транзит" може бути закритою і відкритою. У закритій мережі контролери обмінюються інформацією один з одним. Відкрита мережа відрізняється від закритої тим, що контролери додатково можуть обмінюватися інформацією із зовнішнім абонентом.

Для побудови закритої мережі ніяких додаткових пристроїв, крім тих, що використовуються для керування процесом, не потрібно. Для створення відкритої мережі, закрита доповнюється спеціальним пристроєм, який називається шлюзом. Завдяки шлюзу абонент отримує доступ до будь-якого контролера, що є у мережі "Транзит".

У мережі "Транзит" інформація послідовно передається від одного контролера до іншого. Якщо ця інформація потрібна якомусь контролеру, він її використовує і передає наступному, якщо не потрібна вона одразу передається далі.

Мережа "Транзит" захищена від відмови окремих контролерів. Для цього блок контролера приєднується до мережі "Транзит" через блок живлення БП1. Саме тому при виходу з ладу одного з контролерів, його можна від'єднати від блока живлення, а мережа "Транзит" продовжує залишатись працездатною, рис.3.4.

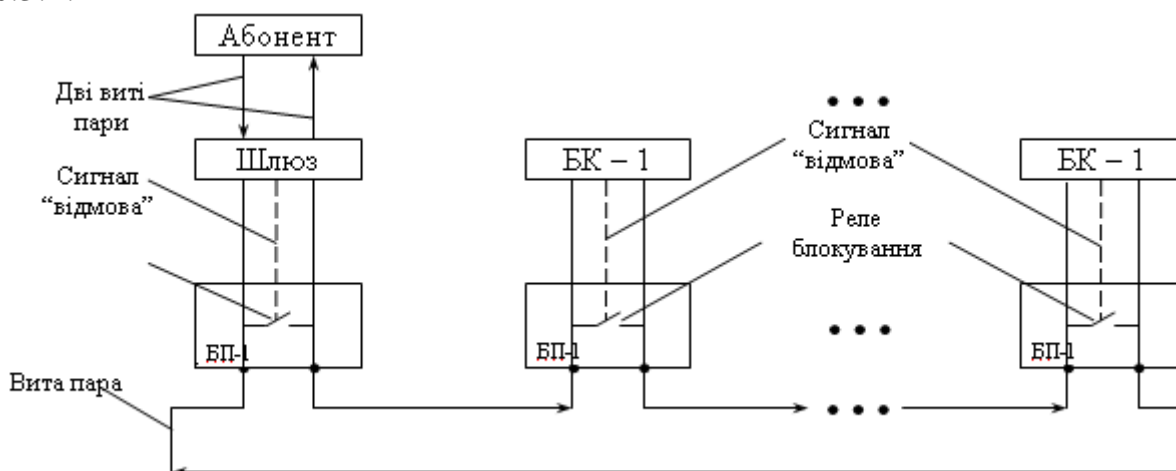


Рисунок 3.4 – Структурна схема відкритої мережі "Транзит"

Кожний контролер у мережі “Транзит” має системний номер, який призначається в процесі програмування від одного до п’ятнадцяти. Системні номери можуть призначатись у будь-якій послідовності або мати пропуски. Якщо контролеру надати нульовий номер він вимкнеться з мережі, проте цілісність мережі зберігається.

Закрита мережа “Транзит” призначена для обміну інформацією між алгоблоками, що знаходяться у різних контролерах. Принцип цього зв’язку такий як і при конфігуруванні алгоблоків одного контролера але є і деякі особливості.

1. Безпосередньо зв’язувати алгоблоки через мережу “Транзит” неможливо. Для такого зв’язку є два спеціальних алгоритми – інтерфейсного виводу ИНВ і вводу інтерфейсного ВИН.

2. Якщо вихідні сигнали будь – яких алгоблоків мають передаватись в інші контролери, то відповідні виходи цих алгоблоків, за стандартною процедурою конфігурування, зв’язуються із входами алгоритму ИНВ. Тільки у такому разі контролер починає передавати в мережу сигнали, що надходять на вхід алгоритму ИНВ, тобто вихідні сигнали алгоблоків, що зв’язані з алгоритмом ИНВ. Всьому “пакету” сигналів надається системний номер, а кожний сигнал у пакеті отримує свій власний номер, що дорівнює номеру відповідного входу алгоритму ИНВ.

Сигнали, що передаються через мережу “Транзит” не мають конкретної адреси. Їх може отримати будь-який контролер, або одночасно декілька контролерів.

3. Якщо будь-які алгоблоки одного контролера мають отримувати сигнали від іншого контролера, їх входи за стандартним конфігуруванням з’єднуються з виходами алгоритму ВИН. Один алгоритм ВИН створює зв’язок лише з одним контролером. Якщо даному контролеру необхідно зв’язатися з декількома контролерами, то необхідно запрограмувати в ньому кілька алгоритмів ВИН.

В алгоритмі ВИН встановлюється системний номер контролера – джерела. Крім того, для кожного виходу алгоритму ВИН призначається номер сигналу, що передається контролером – джерелом. Саме цей сигнал прийде на вхід інших алгоритмів, що з’єднані за конфігурацією з даним виходом алгоритму ВИН.

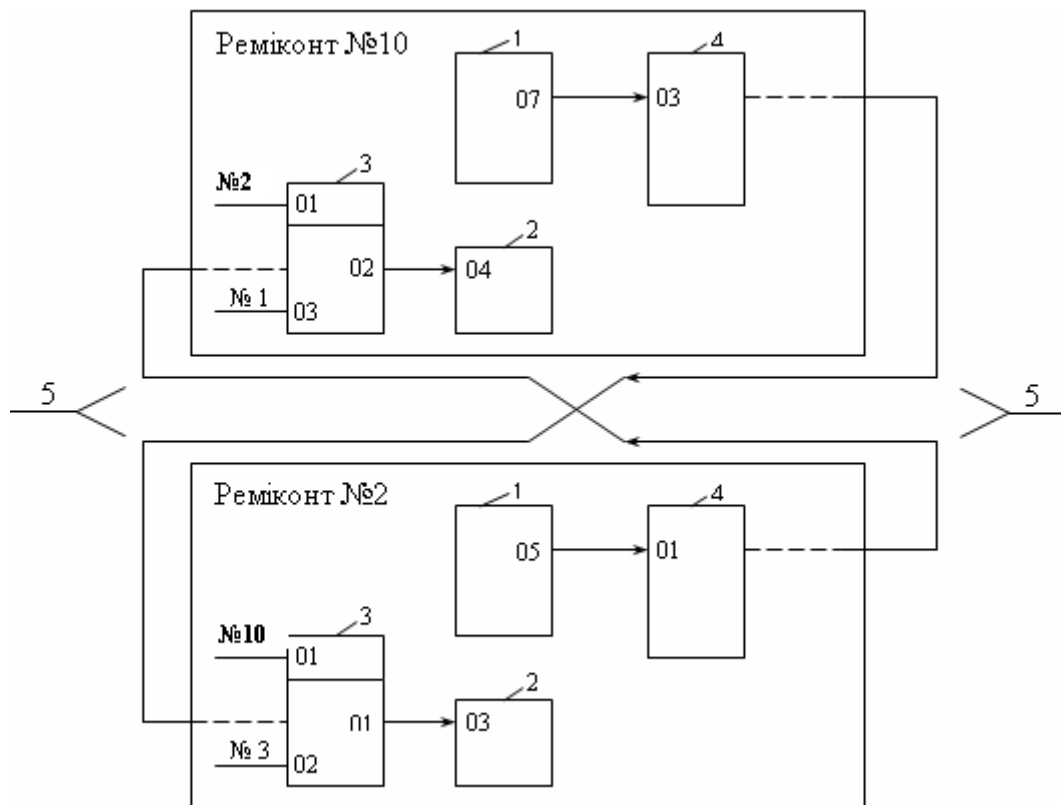
Таким чином, між алгоблоками різних контролерів за допомогою мережі “Транзит” і алгоритмів інтерфейсного вводу-виводу створюється віртуальний канал зв’язку.

Прикладом віртуального каналу між блоками різних контролерів є структурна схема на рис. 3.5.

Обмін інформацією по закритій мережі “Транзит” має певні обмеження.

1. Кожний контролер може передавати в мережу не більше як 30 сигналів. Ця передача завжди створюється за допомогою одного алгоритму ИНВ, причому кількість сигналів, що передається, задається модифікатором цього алгоритму. В одному контролері не може бути більше одного алгоритму ИНВ.





1 – алгоблоки джерела; 2 – алгоблоки приймачі; 3 – алгоритми вводу інтерфейсного ВИН; 4 – алгоритми інтерфейсного виводу ИНВ; 5 – мережа “Транзит”.

Рисунок 3.5 – Структурна схема віртуального каналу.

2. У кожному контролері може бути будь-яка кількість алгоритмів ВИН, при цьому кожний алгоритм ВИН створює зв’язок лише з одним контролером – джерелом.

Один алгоритм ВИН може приймати не більше 21 сигналу. Кількість вхідних сигналів задається модифікатором цього алгоритму.

Мережа “Транзит” перетворюється із закритої у відкриту за допомогою шлюзу. Шлюз має два інтерфейсних канали — мережний та абонентський. Через мережний канал шлюз вмикається у мережу як і будь – який контролер. Через абонентський канал шлюз приєднується до абонента.

В одну мережу “Транзит” може вмикатися будь-яка кількість шлюзів, але загальна сума шлюзів і контролерів має бути не більше п’ятнадцяти.

Абонент може працювати з контролером на рівні алгоблоків і на рівні контурів регулювання. Коли він обмінюється інформацією з алгоблоком вказується номер алгоблока, а при обміні з контуром регулювання – номер контуру.

Запит надсилається при необхідності отримання від контролера якої-небудь інформації.

Команда надсилається абонентом коли необхідно змінити параметри настройки алгоблока або оперативні параметри контуру.

Фізично усіма повідомленнями абонент обмінюється не з контролером, а із шлюзом. Шлюз є посередником між абонентом і мережею “Транзит”. Отримуючи від абонента запити і команди, шлюз по мережі передає їх до

потрібного контролера. Шлюз також транслює абоненту відповіді та підтвердження, що контролери передають по мережі.

Протокол зв'язку контролера з абонентом має три рівня: фізичний, інформаційний та транспортний.

Фізичний здійснює електричний зв'язок між абонентом і шлюзом за допомогою послідовного інтерфейсу ІРПС.

Інформаційний рівень пакує повідомлення, що передаються, у певний формат, додає до них заголовки, так звані, «кінцевик» повідомлення і контрольний байт. Коли повідомлення приймаються, інформаційний рівень відокремлює їх від заголовку та «кінцевика» і контролює відсутність помилок.

Повідомлення на інформаційному рівні не розшифровуються.

Транспортний рівень формує зміст повідомлення при передачі і розшифровує зміст повідомлення коли приймає його.

Фізичний рівень реалізується апаратно, а інформаційний та транспортний — програмно[3,19].

### **3.4 Технологічне програмування контролера**

Для програмування Реміконта використовується мова функціональних блоків, яка полягає в реалізації схеми керування об'єктом комбінацією алгоритмів, записаних у його постійну пам'ять (бібліотека алгоритмів).

Технологічне програмування Р-130 здійснюється за допомогою пульта настроювання ПН-1, який містить світлодіодні, цифрові індикатори та клавіатуру.

Контролер може знаходитися в одному з двох режимів – програмування і роботи. У режимі програмування контролер вимкнений з контуру керування. У цьому режимі стан усіх його виходів, таймерів, лічильників і комірок накопичення заморожений, а на пульту настроювання горить індикатор “прог”.

У режимі програмування можна змінювати усі параметри - як контролера у цілому, так і алгоблоків.

Контролер має 8 процедур програмування – тестування, приладові параметри, системні параметри, алгоритми, конфігурування, параметри настройки, початкові умови, робота з ППЗП.

При тестуванні виконується тестування пам'яті, інтерфейсу, сторожа циклу, клавіатури, індикаторів і вихідних ПЗО.

У процедурі «приладові параметри» задаються і контролюються параметри, загальні для усіх алгоблоків контролера.

У процедурі «системні параметри» задаються параметри, що визначають особливості взаємодії контролера з іншими пристроями за інтерфейсним каналом, - системний номер контролера і режим інтерфейсного каналу.

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

Інтерфейсний канал може працювати у двох режимах: інформаційному та командному.

В інформаційному – можна викликати значення усіх параметрів, передбачених протоколом інтерфейсу.

У командному – через інтерфейс можна викликати параметри і змінювати їх.

У процедурі «алгоритми» відбувається “заповнення” алгоблоків алгоритмами. У загальному випадку для кожного алгоблока задаються три параметри: сам алгоритм, його модифікатор і масштаб часу. У часному випадку модифікатор і (або) масштаб часу можуть бути відсутніми.

У процедурі “конф” (конфігурування) визначається стан кожного входу алгоблоків, який може бути в зв’язаному або вільному стані.

У процедурі «параметри настройки» встановлюються значення параметрів настройки – констант і коефіцієнтів. Ця процедура виконується лише для входів алгоблока, які у процедурі конфігурування були визначені як вільні.

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

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

Процедура «Робота з ППЗП» відбувається тільки при необхідності перепрограмування постійного запам’ятовуючого пристрою.

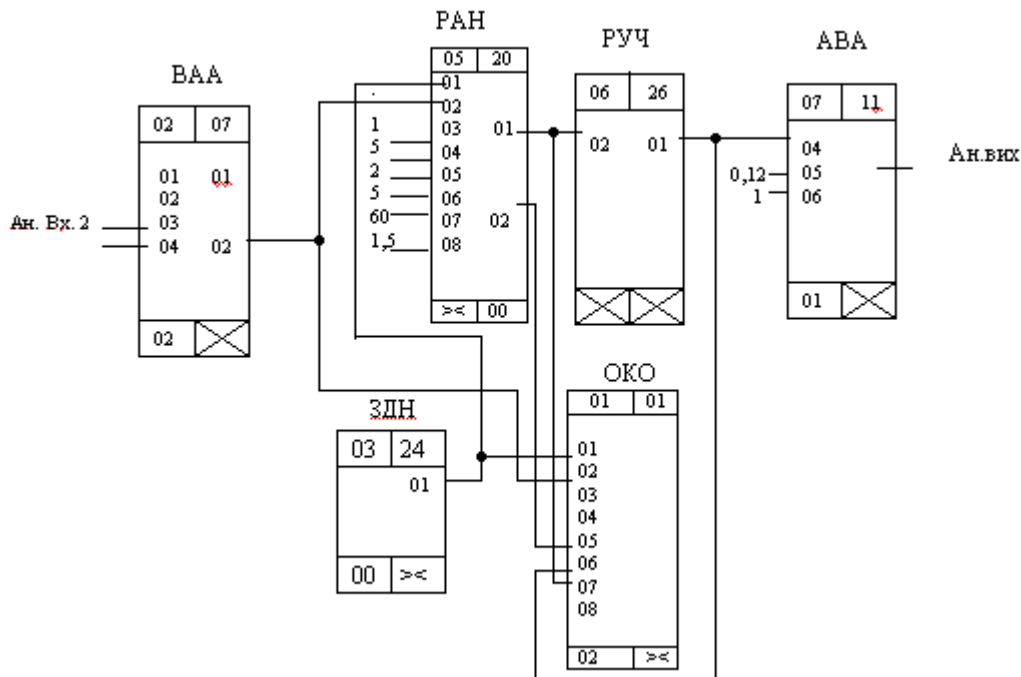
У режимі “робота” можна контролювати параметри і сигнали, а також змінювати коефіцієнти. При цьому засоби самодіагностики контролера фіксують несправності, що пов’язані з відмовами апаратури, порушеннями правил програмування та ін.

Усі несправності поділяються на відмови та помилки.

При відмові нормальна робота контролера неможлива, і на пульті настройки засвічується миготливим світлом індикатор “отказ”.

При наявності помилки, контролер продовжує нормально працювати, і не вимикається з контуру керування, а на пульті настройки засвічується миготливим світлом індикатор “ОШ” (помилка)[19].

Схема конфігурації Реміонта Р-130 складається з фрагментів, кожний з яких відповідає окремому, відносно автономному функціональному вузлу системи регулювання, наприклад, одному контуру регулювання. Схеми розробляються з використанням спрощених зображень алгоритмів, а їх розміщення на кресленику виконується з мінімальною кількістю перетинань ліній зв’язку. Біля кожного фрагменту (зверху) пишеться шифр алгоритму, а біля вільних входів значення коефіцієнтів і констант. У середині фрагменту у лівій клітинці показується номер алгоблока, у правій верхній – код алгоритму, у нижній лівій та правій, відповідно, модифікатор і масштаб часу, якщо алгоритм їх має. Окрім того біля лівої границі фрагменту пишуться номери входів, а біля правої – номери виходів. На рис.3.6 показана схема конфігурації аналогового ПД-закону регулювання.



*ОКО-алгоритм оперативного контролю; ВАА-алгоритм аналогового входу; ЗДН-алгоритм завдання; РАН-алгоритм аналогового регулювання; РУЧ-алгоритм ручного керування; АВА-алгоритм аналогового виходу.*

Рисунок 3.6 – Схема конфігурації аналогового ПІД-закону регулювання.

### 3.5 Шляхи розвитку промислових контролерів Р-130

Нова модель контролера Р-130Іsа призначена для заміни Р-130 на діючих АСУ ТП і створення нових АСУ ТП з великими функціональними можливостями. Контролери Р-130Іsа сумісні з Р-130 за технічними характеристиками, мають такі самі розміри і схеми електричних з'єднань.

Кількість входів-виходів визначається платами розширення, які вставляються в контролер. Вхідними сигналами можуть бути сигнали від термопар ТХК, ТХА, ТПР, ТВР, ТПП, термометрів опору ТСМ, ТСП, можуть бути уніфіковані аналогові сигнали постійного струму 0-5, 0-20, 4-20мА, 0-10В, дискретні сигнали 24В. Вхід може бути сконфігурований як частотний або як лічильник.

Вихідні сигнали формуються, як уніфіковані аналогові – 0-5, 0-20, 4-20 мА або дискретні у вигляді транзисторного виходу – “сухий контакт” з максимальною напругою комутації 40В при струмі навантаження 0,3А і сильнострумове релейного виходу з максимальною напругою комутації 220В при струмі навантаження 2А.

Контролер може обслуговувати 8 контурів регулювання і підтримує мережі Modbus RS232/485 і TCP/IP(Ethernet)[20].

### 3.6 Програмування контролерів Р-130 Іsа

Розширення сімейства контролерів Р-130 укладається в лінію нових виробів Челябінського ВАТ “ЗЭиМ ” - “Р-130Іsа” – КРОСС” – “ТРАССА”, які

мають єдині засоби технологічного програмування у вигляді широко відомого середовища ISaGRAF французької фірми "CJ International". В ISaGRAF закладена методологія структурного програмування, яка дає можливість користувачеві описати процес, що автоматизується, в найлегшій і зрозумілішій формі. Інтерфейс з користувачем системи ISaGRAF має багатовіконний режим роботи, графічні та текстові редактори, роботу з мишею і т.ін. Набір інструментів ISaGRAF дозволяє програмувати конфігурації автоматизованих систем будь-якої складності і будь-якою мовою стандарту IEC 61131-3 [21].

### 3.6.1 Методика створення проекту у системі програмування ISaGRAF

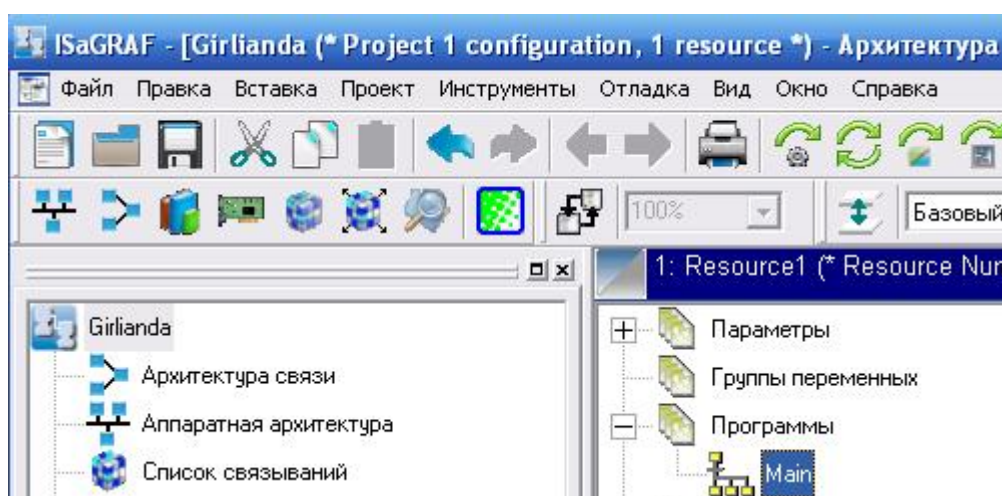
Розробити SFC-програму, що керує послідовним вмиканням 4-х ламп гірлянди. При цьому кожний цикл в роботі гірлянди фіксується лічильником, який після 15 циклів скидається. Початок роботи гірлянди визначається вимикачем напруги.


#### *Розробка 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 відкривається, призначити ім'я проекту, наприклад Girlianda, а також вибрати шаблон Primoporesource. 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.

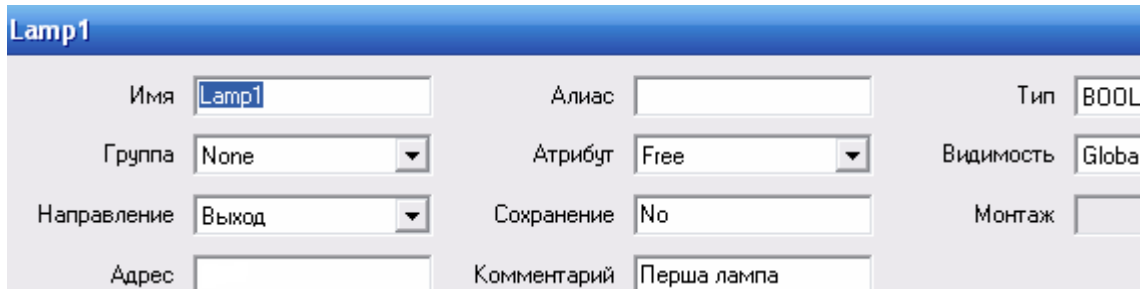
Щоб додати в організаторі програми, SFC-модуль для створення програмного коду з назвою, наприклад Main, необхідно клацнути правою клавішею миші по вкладці «Программы» і вибрати відповідний програмний модуль. Вікно «Архитектура связи» отримає наступний вигляд:



Для оголошення змінних, в панелі інструментів кнопкою  «Словарь» відкрити вікно словника змінних і перейти на вкладку «Переменные».

Натиснути ліву клавiшу миші на значку «Глобальные переменные», щоб відкрити відповідну таблицю.


Для її заповнення перейти у режим «Строка». Для цього клацнути правою клавiшею миші на вільному рядку вікна «Глобальные переменные», вибрати «Переключить в режим Строка» і підтвердити лівою клавiшею миші. Після цього двічі клацнути лівою клавiшею миші на вільному рядку таблиці. З'явиться вікно для оголошення змінної:




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

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

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

Щоб з'єднати оголошені змінні з пристроями вводу-виводу 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 змінної Power, рис.3.7а. Натиснути ОК,

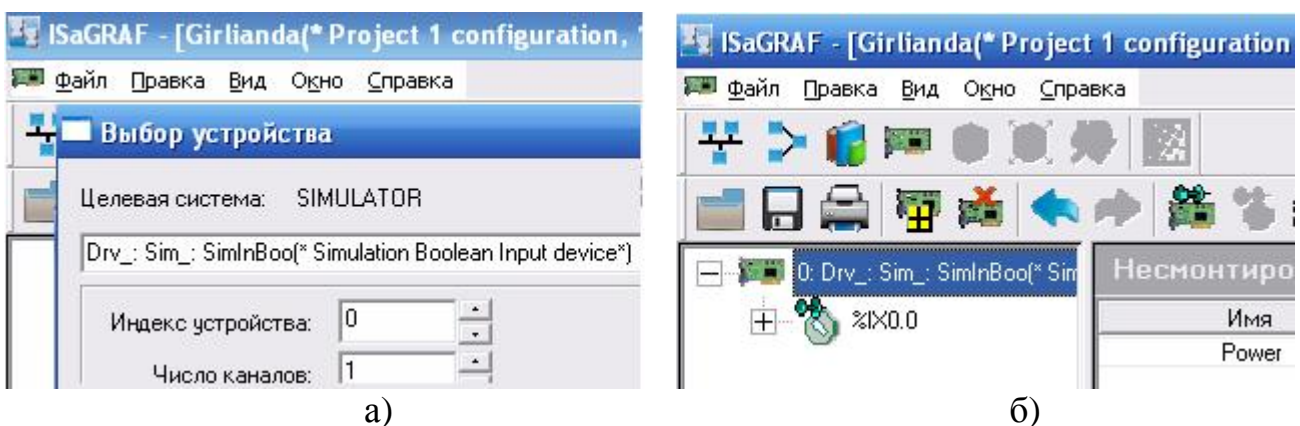


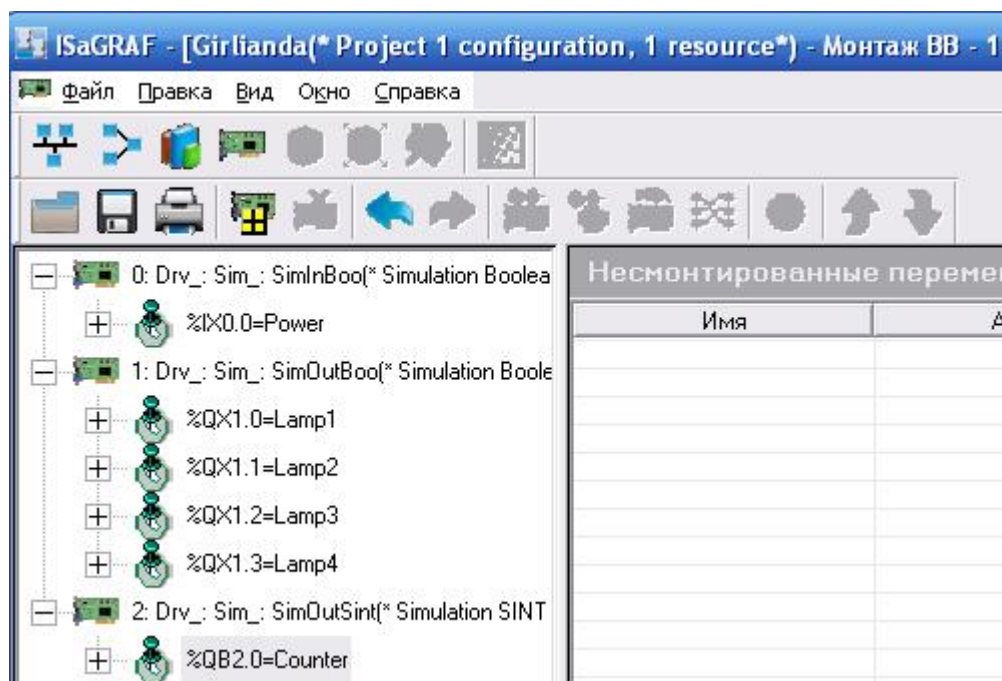
Рисунок 3.7 – Вибір цільової системи (а) і з'єднання змінної з каналом пристрою вводу-виводу (б)

у лівій частині вікна «Монтаж ВВ» з'явиться пристрій вводу на один канал, а у правій частині – назва вхідній змінній Power, рис.3.7б.

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:

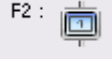


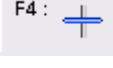
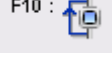
Аналогічно змонтувати вихідні булеві змінні, тільки у вікні «Выбор устройства» (Device Selection) вибрати драйвер Drv\_:Sim\_:SimOutBoo(\*Simulation Boolean Input device\*) і встановити 4 канали виводу для булевих змінних Lamp1, Lamp2, Lamp3, Lamp4. Для підключення цілочислової вихідної змінної Counter 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:



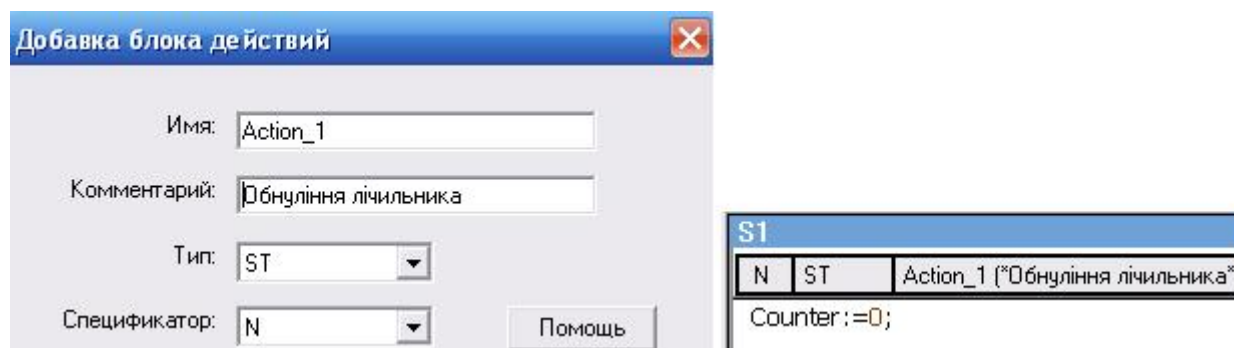
А у словнику змінних з'являться монтажні адреси змінних.

## Створення програмного коду

В організаторі програми двічі клацнути лівою клав'яшею 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 клацнути правою клав'яшею миші, щоб додати блок дій. Після заповнення усіх рядків вікна «Добавка блока действий», щодо назви дії, коментарів, мови програмування та специфікатора, натиснути кнопку ОК і у робочому полі дії кроку 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 приведені на рис. 3.8.

Для компіляції проекту в 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.



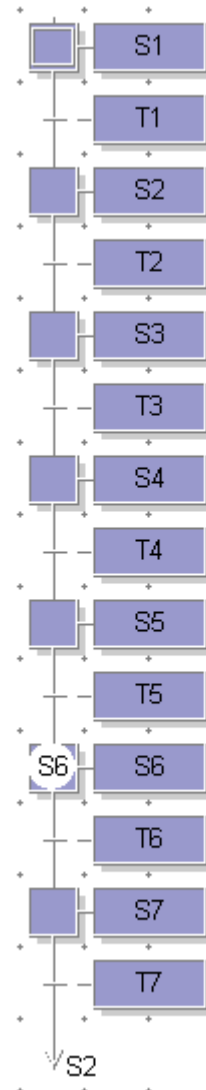
Рисунок 3.8 – Програмний код

Відкрити його лівою клавішею миші і на екрані з'явиться вікно «Імітатор 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.3.9а. Роботу програми можна також побачити на робочому полі, де крок, що знаходиться у стані виконання, позначається білою міткою (S6), рис.3.9б.

Щоб завантажити програму у ПЛК, необхідно приєднатися до контролера і змінити конфігурацію апаратної структури. Для цього у вікні «Конфігурація: Свойства» у якості цільової системи вибрати не Simulator, а



а)



б)

Рисунок 3.9 – Емуляція виконання програми у вікні «Имитатор ВВ (а) і на робочому полі (б)

WIN32-TGT\_L і оголошення змінних здійснити відповідно до вибраних типів модулів вводу-виводу.

### Тестові та контрольні запитання до розділу 3

**I-й рівень запитань і тверджень дозволяє перевірити загальний рівень засвоєння матеріалу розділу і передбачає відповідь на запитання або твердження “Так” чи “Ні”.**

1. БУМ-1 є основним блоком комплексу технічних засобів Р-130.
2. Взаємодія оператора-технолога з Реміконтом Р-130 здійснюється тільки через пульт настройки.
3. Клемно-блокові з'єднувачі призначені для підключення зовнішніх електричних кіл до блока БК-1.
4. Чи можна за допомогою КБС приєднати БУТ-10, БУС-10, БУМ-10, БПР-10 до БК-1?

5. За допомогою БУМ-10 та БУС-10 можна перетворити сигнали терморпар та термометрів опору в уніфіковані електричні сигнали 0-5мА та підключити їх до блока контролера БК-1.

6. Чи забезпечує БП-1 живлення контролера та вхідних кіл, а також приєднання контролера до мережі “Транзит”.

7. Віртуальна структура контролера Р-130 складається з алгоритмічних блоків і бібліотеки алгоритмів.

8. Алгоритми вводу-виводу аналогових і дискретних сигналів, алгоритми обслуговування лицевої панелі та алгоритми прийому і передачі сигналів через інтерфейсний канал сполучають апаратуру контролера з основною кількістю функціональних алгоритмів.

9. Кожен бібліотечний алгоритм має два реквізити: модифікатор та бібліотечний номер.

10. Чи задає масштаб часу алгоритму додаткові властивості?

11. Алгоритм оперативного контролю ОКО призначений для формування інформації, яка надходить на індикатори лицевої панелі.

12. Контролер Р-130 програмується будь-якою мовою стандарту МЕК 61131.

13. Р-130 програмується за допомогою пульта настроювання.

14. Мережа “Транзит” підтримує конфігурацію «загальна шина».

15. Контролери Р-130 обмінюються інформацією один з одним по закритій і відкритій мережі, а із зовнішніми абонентами – по відкритій.

16. У кожному контролері алгоритм ВИН можна використовувати тільки 4 рази.

17. Один контролер Р-130 забезпечує до 10 контурів автоматичного регулювання.

18. У процедурі “конф” (конфігурування) визначається стан кожного входу алгоритму.

19. Програмування контролерів Р-130 здійснюється за допомогою пульта настроювання шляхом натискання відповідних клавіш та вибору з бібліотеки контролера необхідних алгоритмів, конфігурування їх поміж собою та встановлення відповідних параметрів.

20. Контролер Р-130Isa програмується за допомогою програмного пакета IsaGraf.

*Перевірте відповіді на сторінках: 1 - 54 с., 2 - 55 с., 3 - 57 с., 4 – 57...58 с., 5 - 57 с., 6 - 57 с., 7 – 59 с., 8 - 60 с., 9 - 61 с., 10 - 61 с., 11 - 61 с., 12 - 66 с., 13 – 59...60 с., 14 - 63 с., 15 - 63 с., 16 - 65 с., 17 - 54 с., 18 - 67 с., 19 - 66 с., 20 - 69 с.*

**II-й рівень запитань (тестові запитання) дозволяє перевірити рівень засвоєння матеріалу розділу із заглибленням у технічні особливості будови та структури контролера Р-130 та передбачає вибір правильної відповіді з наведених трьох варіантів.**

**1. Який з перелічених блоків комплексу технічних засобів Р-130 є основним: ПН-1, БУТ-1, БК-1, БУМ-1, БП-1, БУС-1?**

- а) ПН – 1;
- б) БП – 1;
- в) БК-1.

**2. Які модулі входять в базову частину блока контролера БК-1?**

- а) МКП, ПРЦ-10, МСН-10, акумулятор;
- б) МКП, ПРЦ-10, ПН-1, акумулятор;
- в) МКП, ПРЦ-10, МСН-10.

**3. Які модулі блока БК-1 є змінними?**

- а) ПЗОА, ПЗОБ, МДА;
- б) АЦП, ДЦП, УСП;
- в) МАС, МДА, МСД.

**4. Коли використовується клемно-блоковий з'єднувач КБС-3?**

- а) коли до блока контролера необхідно підключити аналоговий сигнал;
- б) коли до блока контролера необхідно підключити аналоговий або дискретний сигнал;
- в) коли до блока контролера необхідно підключити дискретний сигнал.

**5. Для чого призначена лицева панель блока контролера?**

- а) для технологічного програмування, налагодження та контролю Р-130;
- б) для подачі команд оператора та індикації процесу регулювання;
- в) для подачі команд оператора та програмування Р-130.

**6. Скільки каналів мають БУТ-10, БУС-10, БУМ-10?**

- а) по одному каналу;
- б) БУТ-10 та БУМ-10 мають по два канали, а БУС-10 – три;
- в) по два канали.

**7. Яка апаратура відноситься до складу віртуальної структури Р-130?**

- а) апаратура вводу-виводу інформації та апаратура оперативного керування і налагодження;
- б) алгоритмічні блоки та бібліотека алгоритмів;
- в) апаратура вводу-виводу інформації, апаратура оперативного керування та налагодження, апаратура інтерфейсного каналу.

**8. Як визначається комплектність контролера і в якій процедурі вона здійснюється?**

а) комплектність контролера визначається номерами модифікацій плат ПЗОА і ПЗОБ, встановлюється вона у процедурі програмування «Приладові параметри»

б) комплектність контролера визначається візуально, встановлюється вона у процедурі програмування «Системні параметри»;

в) комплектність контролера визначається номерами модифікацій плат ПЗОА і ПЗОБ, встановлюється вона у процедурі програмування «Параметри настройки».

**9. В якій процедурі програмування здійснюється розміщення алгоритмів в алгоблоки?**

- а) у процедурі конфігурування;
- б) у процедурі «параметри настройки»;
- в) у процедурі «алгоритми».

**10. Скільки входів і виходів може мати алгоритм?**

- а) кількість входів у кожному алгоритмі визначається його функцією, а кількість виходів – один;
- б) кожен алгоритм може мати не більше 99 входів та 25 виходів;
- в) кожен алгоритм може мати не більше 25 входів та 99 виходів.

**11. Які алгоритми розміщуються тільки у перших чотирьох блоках?**

- а) алгоритми вводу-виводу;
- б) алгоритми вводу;
- в) алгоритми, що формують інформацію, яка надходить на індикатори лицевої панелі.

**12. Які алгоритми мають обмеження на кратність їх використання у межах одного контролера?**

- а) алгоритми, що формують інформацію, яка надходить на індикатори лицевої панелі;
- б) алгоритми вводу-виводу інформації;
- в) алгоритми не мають обмеження на кратність їх використання у контролері.

**13. Коли використовуються клавіші «←» «→» на лицевій панелі контролера Р-130?**

- а) при зміні параметрів або завдання;
- б) при переході на один з режимів САР (локальний, каскадний, дистанційний);
- в) при виборі параметрів контролера.

**14. Яке призначення інтерфейсного каналу?**

- а) для обміну інформацією із зовнішніми приладами;
- б) для живлення модулів МАС, МДА, МСД;
- в) для створення мережі “Транзит”.

**15. Якою може бути мережа “Транзит”?**

- а) мережа «Транзит» завжди є відкритою;
- б) мережа «Транзит» може бути відкритою та закритою;
- в) мережа «Транзит» завжди є закритою.

**16. Як мережа “Транзит” захищена від відмови окремих контролерів?**

- а) при відмові будь-якого контролера мережа виходить з ладу;
- б) блок контролера приєднується до мережі “Транзит” через блок живлення БП1;
- в) блок контролера приєднується до мережі “Транзит” через блок перемикачів БПР-10.

**17. Як сполучаються алгоблоки через мережу “Транзит”?**

- а) за допомогою пристроїв ИНВ та ВИН;
- б) за допомогою міжблокового з’єднувача МБС-20;
- в) за допомогою алгоритмів ИНВ та ВИН.

**18. Скільки процедур програмування має Р-130?**

- а) вісім;
- б) три;
- в) п'ять.

**19. Яка послідовність дій при розробці проекту в IsaGraf?**

а) відкрити новий проект, оголосити змінні, з'єднати оголошені змінні з пристроями вводу-виводу, створити апаратну архітектуру, додати програмний модуль в організаторі програми.

б) відкрити новий проект, створити апаратну архітектуру, оголосити змінні, з'єднати оголошені змінні з пристроями вводу-виводу, додати програмний модуль в організаторі програми.

в) відкрити новий проект, додати програмний модуль в організаторі програми, оголосити змінні, створити апаратну архітектуру, з'єднати оголошені змінні з пристроями вводу-виводу.

**20. Як створюється програмний код в IsaGraf?**

а) відкрити редактор, змоделювати роботу системи, розробити код програми, виконати компіляцію проекту;

б) відкрити редактор, розробити код програми, виконати компіляцію проекту, змоделювати роботу програми;

в) відкрити редактор, розробити код програми, змоделювати роботу програми, виконати компіляцію проекту.

*Перевірте відповіді на сторінках: 1 - 54 с., 2 - 58 с., 3 - 58 с., 4 - 57 с., 5 - 55 с., 6 - 57 с., 7 - 59 с., 8 – 66...67 с., 9 - 67 с., 10 - 60 с., 11 - 61 с., 12 - 61 с., 13 - 56 с., 14 - 63 с., 15 - 63 с., 16 - 63 с., 17 - 64 с., 18 - 66 с., 19 – 69...71 с., 20 – 72...74 с.*

**III-й рівень запитань (контрольні запитання) дозволяє перевірити рівень засвоєння матеріалу із систематизацією отриманої у розділі інформації та передбачає відповіді, що побудовані на логічному аналізі або порівнянні із приведенням конкретних технічних параметрів (даних або прикладів).**

1. Розкрийте призначення всіх пристроїв, що входять до апаратної частини Р-130.

2. Проаналізуйте які пристрої контролера Р – 130 мають бути задіяні для прийому інформації про стан зовнішніх реле.

3. Поясніть як взаємодіють апаратна частина та віртуальна структура контролера Р-130.

4. Дайте характеристику алгоритмічним блокам.

5. Як взаємодіють між собою алгоритми та алгоблоки, які параметри та реквізити слід обов'язково задавати?

6. Поясніть послідовність виконання процедур програмування Р-130 і які операції треба виконувати у кожній з процедур.

7. Як формується віртуальний канал в Р-130?

8. Які засоби є в контролері Р-130 для підвищення надійності його роботи у складі АСУ ТП?

9. Які додаткові можливості з'являються в нових модифікаціях контролера Р-130?

10. Розкрийте основні можливості та принцип роботи у середовищі IsaGraf.

*Перевірте відповіді на сторінках: 1 – 54...68 с., 2 – 54...59 с., 3 – 59...62 с., 4 – 59...66 с., 5 – 59...68 с., 6 – 66...68 с., 7 – 64...66 с., 8 – 55...68 с., 9 - 68 с., 10 – 68...74 с.*

## 4 МІКРОПРОЦЕСОРНИЙ КОНТРОЛЕР МІК-51

### 4.1 Загальна характеристика та фізична структура

МІК-51 – це компактний малоканалний багатофункціональний мікропроцесорний контролер, призначений для автоматичного регулювання і логічного управління технологічними процесами. Він спроектований, розроблений і вироблений на підприємстві «Мікрол», м. Івано-Франківськ, Україна. Завдяки малоканалності МІК-51 дозволяє економічно управляти невеликим агрегатом або забезпечити високу живучість крупних систем управління.

У контролері передбачено до 9 незалежних контурів регулювання, кожен з яких може бути локальним або каскадним, з аналоговим або імпульсним виходом, з ручним, програмним (зокрема багатопрограмним) або супервізорним задавачем.

МІК-51 містить засоби оперативного управління, які розташовані на лицевій панелі контролера. За допомогою їх можна вручну змінювати режими роботи, встановлювати завдання, управляти ходом виконання програми, управляти виконавчими пристроями, а також контролювати сигнали і відображати помилки. Стандартні аналогові і дискретні датчики і виконавчі пристрої приєднуються до контролера МІК-51 за допомогою індивідуальних кабельних з'єднувачів. У середині контролера сигнали обробляються у цифровій формі.

Контролери МІК-51 можуть об'єднуватися в локальну управляючу мережу шинної конфігурації і обмінюватися інформацією. Для такого об'єднання ніяких додаткових пристроїв не потрібно.

Програмування контролера здійснюється за допомогою клавіш передньої панелі або по інтерфейсу за допомогою спеціального програмного забезпечення - візуального редактора АЛЬФА. Система програмування реалізована відповідно до вимог стандарту Міжнародної Електротехнічної Комісії (МЕК) ІЕС 61131-3 і призначена для розробки прикладного програмного забезпечення збору даних і управління технологічними процесами мовою функціональних блокових діаграм Function Block Diagram (FBD).

МІК-51 має розвинені засоби самодіагностики, сигналізації і ідентифікації несправностей. Запрограмована інформація зберігається в ньому при вимкненому живленні завдяки вбудованій енергонезалежній пам'яті.

Контролер МІК-51 є комплексом технічних засобів. У його складі центральний мікропроцесорний блок, клемноблокові з'єднувачі і модулі розширення.

Базова модель контролера має 4 аналогові входи (2 універсальних, 2 уніфікованих), 1 аналоговий вихід, 3 дискретні входи і 5 дискретних виходів.

Модулі розширення мають 7 модифікацій – від МР-51-01 до МР-51-07, які відрізняються кількістю і видом входів-виходів.



Контролер МІК-51 це - проектно-компонований виріб, який дозволяє користувачеві вибирати комплект модулів розширення і клемноблокових з'єднувачів відповідно до кількості і виду вхідних-вихідних сигналів.

Центральний мікропроцесорний блок, рис.4.1, має габаритні розміри 96×96×189 (В×Ш×Г) і розрахований на втоплений монтаж у вертикальній панелі щита або пульта управління. Він перетворює аналогову і дискретну інформацію в цифрову форму, обробляє її та формує керувальний вплив.



Рисунок 4.1 – Центральний мікропроцесорний блок контролера МІК-51

МІК-51 має лицеву панель, за допомогою якої можна виконувати оперативне управління контурами регулювання і здійснювати програмування контролера.

Лицева панель в режимі оперативного управління контурами регулювання приведена на рис.4.2



Рисунок 4.2 – Лицева панель МІК-51 в режимі оперативного управління

Оперативне управління контурами регулювання здійснюється за допомогою клавіш лицевої панелі, двох чотирирозрядних, одного трирозрядного і одного одnorозрядного цифрових індикаторів, а також набору світлодіодів, які дозволяють змінювати режими, встановлювати завдання, управляти виконавчими механізмами, контролювати сигнали, відображати

аварійні ситуації. При програмному регулюванні засоби оперативного управління дозволяють вибирати необхідну програму, пускати, зупиняти, і скидати програму, переходити до наступної ділянки програми, а також контролювати хід виконання програми.

Лицева панель має 8 клавiш, призначення яких наведено у табл.4.1.

Таблиця 4.1 Призначення клавiш оперативного управління

| Клавiша   | Призначення клавiші  |
|-----------|--|
| [↻] Меню  | У режимі РОБОТА здійснюється по колу перемикання панелей режимів індикації позначених світлодіодами.<br>У режимі ПРОГРАМУВАННЯ дана клавiші виконує функцію скасування виконаних дій, операцій.  |
| [✓] Ввід  | Клавiша призначена для підтвердження виконуваних дій, операцій і для фіксації значень, що вводяться.   |
| [№кн ↑]   | Клавiша призначена для зміни індиційованого каналу. У режимі конфігурування використовується для налаштування умов переходів програмного задавача.   |
| [№кн ↓]   | Клавiша призначена для зміни індиційованого каналу (входу, регулятора, програмного задавача) у бік зменшення.  |
| [P/A]     | Кожне натискання клавiші викликає перехід регулятора з автоматичного режиму роботи в режим ручного керування й назад (разом з натисканням клавiші [✓], для підтвердження виконання операції переходу). Також, якщо обрано панель індикації стану програмного задавача, цією клавiшею можна перевести програмний задавач у режим "робота", "очікування" або "стоп". |
| [Завд.]   | Клавiша призначена для виклику значення внутрішньої заданої уставки для редагування або для перемикання режимів заданих величин.   |
| [▲] Знач. | Клавiша "більше". При кожному натисканні цієї клавiші здійснюється збільшення значень, заданої уставки, вихідного сигналу керування (керуючого впливу) або значення змінюваного параметра.   |
| [▼] Знач. | Клавiша "менше". При кожному натисканні цієї клавiші здійснюється зменшення значень, заданої уставки, вихідного сигналу керування (керуючого впливу) або значення змінюваного параметра.   |

Однорозрядний дисплей "№" показує номер контуру регулятора, з яким працює оператор. На чотирирозрядний дисплей ПАРАМЕТР виводиться поточне значення регульованого параметра. Два світлодіодні індикатори ▼ і ▲ у цій групі сигналізують про спрацьовування сигналізації виходу параметра, що відображається, за уставки "менше" або "більше". На чотирирозрядний дисплей ЗАВДАННЯ виводиться задане значення регульованого параметру. На трирозрядний дисплей ВИХІД виводиться значення сигналу на виході регулятора. Два світлодіодні індикатори ▼ і ▲ у цій групі сигналізують про спрацьовування виходів імпульсного регулятора в напрямі "менше" або "більше". Чотири світлодіодні індикатори режимів управління вказують, в якому режимі працює контур:

КУ – каскадне управління,  
 ЛУ – локальне управління,  
 РУ – ручне управління,  
 СУ – слідкувальне управління.

Призначення індикаторів панелі регуляторів наведені у табл.4.2.

Таблиця 4.2 Призначення індикаторів панелі регуляторів

| Індикатор | Призначення індикатора  |
|-----------|---|
| Рег       | Світиться, якщо вибрана панель індикації і керування регулятором (PID(60), PID_IMP(62) і др. Перемикання номера регулятора здійснюється за допомогою клавіш "Некн↑" і "Некн↓". Максимальна кількість регуляторів задається при програмуванні. |
| Прг       | Світиться, якщо обрано панель індикації і керування програмним задавачем. Перемикання номера програмного задавача здійснюється за допомогою клавіш "Некн↑" і "Некн↓".   |
| Інд       | Світиться, якщо обрано користувальницьку панель індикації і керування. Інформація і керування визначаються користувачем при програмуванні у функціональному блоці USER(63).   |
| Вх        | Світиться, якщо обрано панель індикації входів функціональних блоків.   |
| Вих       | Світиться, якщо обрано панель індикації виходів функціональних блоків.  |
| Прм       | Світиться, якщо обрано панель індикації й редагування параметрів функціональних блоків.   |
| ВІД       | Світиться тільки в режимі ПРОГРАМУВАННЯ, якщо існує системна або програмувальна ВІДМОВА.  |
| ОШ        | Світиться в режимі ПРОГРАМУВАННЯ і мигає в режимі РОБОТА при наявності помилок програмування.   |
| ▶         | Світиться, якщо програмний задавач перебуває в режимі РОБОТА.   |
|           | Світиться, якщо програмний задавач перебуває в режимі "очікування" або "зупинник".  |
| ■         | Світиться, якщо програмний задавач перебуває в режимі "стоп" або "кінець програми".   |
| РБ        | Світиться, якщо контролер МІК-51 перебуває в режимі РОБОТА  |
| ПР        | Світиться, якщо контролер МІК-51 перебуває в режимі ПРОГР.  |
| ІН        | Блимає, якщо відбувається передача даних по інтерфейсному каналу.   |

Для реалізації оперативного управління контролер МІК-51 має бути переведений у режим РОБОТА і вибрана панель індикації регуляторів (світиться індикатор РЕГ).

Оперативне управління ведеться за виборчим принципом: вибирається номер контуру і для нього контролюються і змінюються режими, параметри і сигнали. Виключення складає лише контроль помилок, який ведеться для всього контролера в цілому.

Всі зовнішні кола підключаються до контролера МІК-51 через 3 розніми, що розташовані на задній стінці приладу. Через верхній рознім підключаються кола аналогових і дискретних входів - виходів, кола живлення дискретних виходів і інтерфейс.

Через середній рознім підключаються модулі розширення МР-51.

Через нижній двоконтактний рознім-клему підключається електроживлення контролера МІК-51.

Кабелі, що з'єднують контролери МІК-51 з датчиками і виконавчими механізмами, приєднуються до них через клемно-блокові з'єднувачі КБЗ-29-01(КБЗ-29Р-01, КБЗ-29К-01, КБЗ-29С-01), рис.4.3, або КБЗ-24-10, які виконані на основі безгвинтових плоско-пружинних клем і встановлюються на рейку DIN35x7.5 EN50022[22].



Рисунок 4.3 – Клемноблокові з'єднувачі

## 4.2 Архітектура контролера

Архітектура контролера МІК-51 описує інформаційну організацію контролера і характеризує його як ланку системи управління. Частина архітектурних елементів структури реалізована апаратно, частина - програмно. Все програмне забезпечення, що формує архітектуру, зашите в постійному запам'ятовуючому пристрої і користувачеві недоступно. Незалежно від того, які реалізовані елементи архітектури – апаратні або програмні – користувач може уявляти контролер як виріб, в якому всі елементи реально існують у вигляді окремих вузлів.

До складу архітектури контролера входять:

- апаратура вводу-виводу інформації (базової моделі і модулів розширення);
- апаратура оперативного управління і настройки;
- апаратура інтерфейсного каналу;
- бібліотека функціональних блоків;
- функціональні блоки.

Апаратура вводу-виводу інформації розрахована на прийом і видачу двох видів сигналів- аналогових і дискретних. Формування імпульсних сигналів на виході імпульсного регулятора виконується програмно і ці сигнали поступають на виконавчі механізми через дискретні виходи контролера. Апаратура вводу (вхідні ПЗО) перетворює аналогові і дискретні сигнали, що поступають на вхід контролера, в цифрову форму, а апаратура виводу (вихідні ПЗО) здійснює зворотне перетворення. Всі аналогові і дискретні входи і виходи контролера

повністю універсальні. У початковому стані вони не «прив'язані» до яких-небудь обмежень контролера. Така прив'язка здійснюється користувачем і реалізується в процесі програмування.

Апаратура оперативного управління (лицева панель) розрахована на оператора-технолога і оператора-наладника. За допомогою набору клавіш, цифрових і світлодіодних індикаторів оператор-технолог «веде» технологічний процес - контролює його параметри, змінює режими управління та уставки, пускає, зупиняє, скидає програму і т.ін.

За допомогою лицевої панелі можна програмувати контролер, виконувати настроювання його параметрів, а також контролювати сигнали у внутрішніх точках структури. У загальному вигляді лицева панель можна розглядати як інтерфейс з людиною.

Апаратура інтерфейсного каналу забезпечує послідовний зв'язок декількох контролерів. Цей канал має приємопередавач, який перетворює вхідний потік послідовних біт інформації у паралельний код представлений у вигляді байтів, а також здійснює зворотне перетворення. Завдяки достатньо великій швидкості передачі побітної інформації через інтерфейсний канал, вважається, що всі сигнали передаються одночасно.

Бібліотека функціональних алгоритмів достатня для того, щоб можна було розв'язувати порівняно складні задачі автоматичного регулювання і логіко-програмного управління. У ПЗП «зашиито» більше 50 типів функціональних алгоритмів безперервної і дискретної обробки інформації, включаючи блоки ПД регулювання, функціональні блоки математичних, динамічних, нелінійних, аналого-дискретних і логічних перетворень.

Частина бібліотечних алгоритмів виконує особливу задачу – вона зв'язує апаратуру контролера з основною масою функціональних блоків. До цих «зв'язних» функціональних алгоритмів відносяться:

- функціональні алгоритми вводу-виводу аналогових і дискретних сигналів;
- функціональний алгоритм обслуговування панелі користувача;
- функціональні алгоритми прийому і передачі сигналів через інтерфейсний канал.

Апаратні елементи архітектури контролера (вхідні і вихідні ПЗО, лицева панель, інтерфейсний канал) починають виконувати свої функції лише після того, як будуть задіяні відповідні функціональні блоки.

Існує біля 99 блоків з вільним заповненням їх будь-якими функціональними алгоритмами бібліотеки і вільною конфігурацією між собою та входами-виходами контролера. У загальному випадку функціональний блок має свої входи, виходи, параметри настройки і функціональне ядро. Входи функціонального блока з'єднуються сполучними лініями з іншими блоками. Один або декілька функціональних блоків, що сполучені зв'язками між собою, утворюють FBD-програму. Функціональним блокам у програмі користувача присвоюються порядкові номери, які ідентифікують блок в системі, а також визначають черговість виконання блоків у програмі.

У загальному випадку бібліотечний функціональний блок має наступні реквізити:

- бібліотечний номер, який відповідає номеру функції, що виконується;
- номер алгоблока;
- модифікатор розміру;
- базова адреса регістрової області параметрів;

Бібліотечний номер є двозначним десятковим числом і основним параметром, що характеризує властивості функціонального блока.

Модифікатор звичайно задає кількість однотипних входів або операцій, які може виконувати функціональний блок. Наприклад, в суматорі модифікатор розміру задає кількість сумованих входів, а в програмному задавачі – кількість ділянок програми і т.ін. Модифікатор вказується для функціональних блоків, в яких він є, і не вказується для блоків, в яких модифікатор відсутній.

Базова адреса функціонального блока – це номер комірки регістрової області пам'яті контролера в якій повинен розміщуватися перший параметр цього функціонального блока. Параметри кожного функціонального блока розміщуються в регістровій області один за одним, починаючи з адреси 0000, вказаної у відповідному реквізиті функціонального блока.

Базова адреса регістрової області параметрів функціонального блока настроюється користувачем в режимі програмування на рівні конфігурування функціонального блока [7].

Кожний функціональний блок графічно має вигляд прямокутника, усередині якого є позначення функції, яка виконується блоком, модифікатор розміру, номер алгоблока і його параметри.

Програмна модель функціонального блока приведена на рис.4.4.

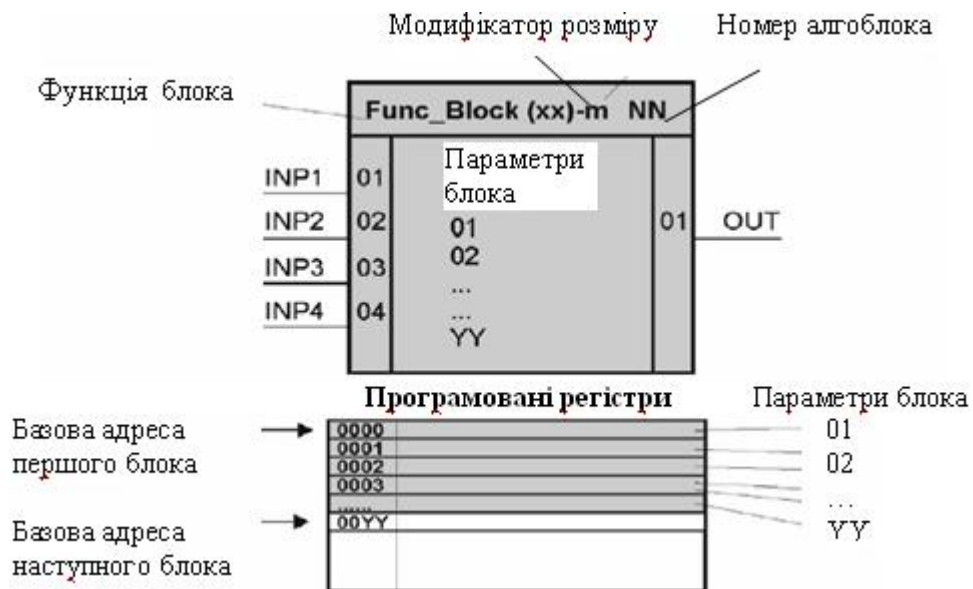


Рисунок 4.4 – Програмна модель функціонального блока

Кожен блок залежно від виконуваної їм функції має певну кількість входів і виходів. Входи (INP) завжди розташовані зліва, а виходи (OUT)-справа.

Параметри настройки функціонального блока задають йому різні властивості, наприклад, номер вхідного сигналу, коефіцієнт посилення, масштабний коефіцієнт, зсув сигналу, постійна часу фільтру, максимальні і мінімальні межі зміни будь-якого параметра і т.ін. Параметри настройки функціонального блока відрізняються від входу блока тим, що параметр настройки блока може залишатися не підключеним.

Перелік сигналів, які обробляються функціональними блоками, включає наступні типи:

- аналоговий (вимірювані параметри, значення аналогових входів-виходів, технічні одиниці, коефіцієнт, швидкість зміни параметра і т.ін.);
- дискретний, імпульсний;
- числовий;
- часовий (уставка часу, тривалість імпульсу).

Аналогові сигнали формуються на виході таких функціональних блоків, як регулятори, суматори, задавачі, інтегратори і т.ін. До аналогових сигналів відносяться такі параметри настройки, як межі спрацьовування компаратора, рівень обмеження і т.п. Не дивлячись на те, що на аналогових входах і виходах контролера сигнал змінюється в діапазоні 0-100% на виході функціональних блоків аналоговий сигнал може змінюватися в ширшому діапазоні – в діапазоні відповідних типів даних. Це дозволяє, наприклад, складати два числа, кожне з яких дорівнює 90%, і на виході суматора одержувати правильний результат. Якщо результат яких-небудь обчислень виявляється більше встановленого типу даних, то сигнал на виході функціонального блока обмежується значеннями відповідного типу даних.

Часові сигнали формуються на виході таймерів, програмних задавачів, одновібраторів і подібних функціональних блоків. До часових сигналів відносяться такі параметри настройки, як постійні часу, протяжність часу, час витримки і т.п.

Числові сигнали – це сигнали на виході лічильника або інших функціональних блоків, робота яких пов'язана з відліком подій. Числовими можуть бути і параметри настройки, наприклад, число може задати граничне значення сигналу на виході лічильника, номер етапу, до якого повинна перейти логічна програма і т.п.

Дискретні сигнали звичайно обробляються логічними функціональними блоками і функціональними блоками, пов'язаними з перемиканням сигналів. Проте дискретними можуть бути і параметри настройки. Наприклад, дискретні сигнали у функціональному блоці завдання визначають, чи повинне виконуватися статичне і динамічне балансування.

Масштабний коефіцієнт – це параметр настройки функціональних блоків, в яких потрібне масштабування сигналів. Так, цей коефіцієнт використовується у функціональних блоках аналогового вводу і виводу,

підсумовування з масштабуванням і т.ін. Коефіцієнт пропорційності використовується в основному у функціональних блоках регулювання.

Швидкість зміни аналогових сигналів – це параметр настройки, що задає, наприклад, швидкість зміни при динамічному балансуванні або задає обмеження швидкості у відповідному функціональному блоці.

Тривалість імпульсу – це параметр настройки функціонального блока імпульсного виводу. Цей параметр задає мінімальну тривалість імпульсу, який сформовано імпульсним регулятором. Тривалість імпульсу відлічується в десятих частках секунди.

Технічні одиниці – це параметри настройки функціональних блоків оперативного контролю. За допомогою цих параметрів задається формат числа, в якому контролювані параметри (наприклад, параметр, завдання, неузгодження і т. ін.) виводяться на індикатори лицевої панелі.

В результаті виконання блоком відповідної функції на його виходах формуються сигнали, які визначаються станом вхідних сигналів і значенням параметрів, що настроюються. Наприклад, якщо на вхід функціонального блока суматора поступає чотири сигнали, то ці сигнали підсумовуються з урахуванням відповідних масштабних коефіцієнтів, які вказані в параметрах і результируючий сигнал поступає на вихід функціонального блоку.

Кількість входів і виходів функціонального блока не фіксовані і визначаються видом функціонального блока. Ні в одному функціональному блоці число входів і виходів не перевищує 12. Деякі функціональні блоки мають неявні входи або виходи, які не доступні для конфігурування. До таких функціональних блоків відносяться функціональні блоки вводу-виводу, функціональні блоки приємопередачі, функціональний блок панелі користувача.

Як тільки один з таких функціональних блоків використовується, його неявні входи або виходи автоматично з'єднуються з апаратурою, яку вони призначені обслуговувати.

Наприклад, коли використовується функціональний блок аналогового вводу, його неявні входи автоматично з'єднуються з АЦП відповідного каналу. А на виходах цього функціонального блока будуть сформовані загальнодоступні сигнали, еквівалентні сигналам, що поступають на аналогові входи контролера. Тому, якщо на вхід будь-якого функціонального блока потрібно подати аналоговий сигнал, цей вхід при програмуванні необхідно з'єднати з відповідним виходом (масштабованим або не масштабованим) функціонального блока аналогового вводу АІН, рис.4.5.

Неявні входи і виходи на графічних схемах функціональних блоків не відображаються. Всі входи і параметри функціонального блока мають скрізну нумерацію від 1 до 99. Виходи функціонального блока також нумеруються двозначною десятковою цифрою від 1 до 99.



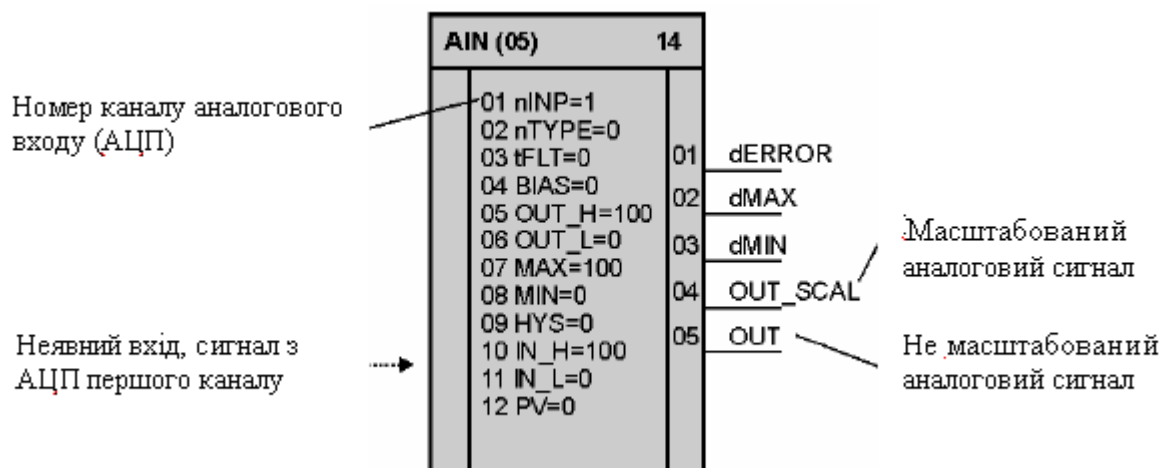


Рисунок 4.5 – Функціональний блок аналогового вводу з неявними входами

Якщо параметри функціонального блока є зв'язаними параметрами, тобто підключеними до яких-небудь виходів інших блоків, то їх нумерація здійснюється із зсувом, рівним максимальному номеру входу.

При програмуванні мовою функціональних блоків в більшості випадків діють правила:

- функціональним блокам привласнюються порядкові номери, які ідентифікують блок в системі, а також визначають черговість виконання блоків в програмі;
- функціональному блоку може бути привласнений будь-який порядковий номер;
- в одній програмі функціональні блоки одного і того ж типу можна використовувати багато разів;
- не може бути вільних (непідключених) входів функціонального блоку;
- виходи функціонального блока можуть залишатися вільними (непідключеними);
- допускаються сполучення входів і виходів будь-якого типу, оскільки перетворення типів здійснюється автоматично.

З цих правил є наступні виключення:

- для деяких функціональних блоків є обмеження на кратність їх використання в межах одного контролера. Так, функціональний блок аналогового вводу можна використовувати лише чотири рази - ці чотири функціональних блоки конфігурують всі аналогові входи. Аналогічні обмеження (з аналогічних причин) мають інші функціональні блоки вводу-виводу інформації.

При конфігуруванні для входів задається номер функціонального блока і номер виходу, до якого підключається даний вхід.

Всі можливості конфігурування однакові як для входів, так і для параметрів.

Зв'язок параметрів функціональних блоків дозволяє виконати автоматичну зміну будь-якого параметра настройки (автопідстроювання).

Сигнал на будь-якому вході при необхідності можна інвертувати. Для безперервних сигналів інверсія означає зміну знаку, а для дискретних - зміна стану (заміну 1 на 0 і 0 на 1).

Інвертування дозволяє, наприклад, віднімати сигнали за допомогою суматора, запускати або скидати таймер не переднім, а заднім фронтом сигналу і т.ін.

Можливості конфігурування не залежать від функціонального блока і визначаються наступними правилами.

Обслуговування функціональних блоків здійснюється послідовно за циклічною схемою. Спочатку обслуговується перший функціональний блок, потім другий і т.д. поки не буде обслужений останній функціональний блок. Коли час в межах встановленого циклу закінчиться, програма знову перейде до обслуговування першого функціонального блока. Час циклу виконання програми користувача фіксований і складає 0,1с. Якщо контролер включений в мережу, тоді залишок часу циклу після обробки програми використовується для обробки запитів, отриманих по інтерфейсному каналу. Мережний обмін носить випадковий характер по відношенню до циклу контролера. Якщо контролер не встиг завершити обмін інформацією в одному циклі, він доопрацьовує розпочатий обмін у наступних циклах.

Загальний час, що витрачається на обслуговування функціональних блоків  $T_{ФБ}$  і інтерфейсного каналу  $T_{ІН}$ , повинен бути менше ніж час циклу, тобто:

$$T_{ФБ} + T_{ІН} < 0,1 \text{ с.}$$

У кожному циклі функціональний блок одержує на свої входи сигнали, що обчислені в попередньому циклі функціональними блоками, до яких даний функціональний блок підключений по конфігурації.

Циклічність обслуговування функціональних блоків призводить до того, що затримка в обробці сигналів залежить від порядку програмування сполучених між собою функціональних блоків. Цю обставину слід враховувати при програмуванні функціональних блоків і для мінімізації затримки бажано, щоб функціональний блок - джерело мав менший номер, ніж функціональний блок-приймач (величина цієї різниці значення не має). Коли, наприклад, номер блока-джерела менше за номер блока-приймача, затримка складає один цикл, а коли номер блока-джерела більше за номер блока-приймача, затримка складає два цикли.

В контролері МК-51 все різноманіття типів сигналів і параметрів настройки обслуговується загальноприйнятим набором типів даних. При цьому забезпечується простий інтерфейс з базами даних інших мов програмування, що сприяє уніфікації і розвитку програмного забезпечення контролера[22].

### **4.3 Мережна архітектура**

Контролер МК-51 має інтерфейсний канал послідовного зв'язку RS-485, за допомогою якого контролери можуть об'єднуватися в локальну або розподілену управляючу мережу. У мережі контролери можуть обмінюватися

інформацією, як з комп'ютером, так і між собою. Всього в мережі може бути до 31 контролера, у тому числі інших моделей.

Контролери в мережі зв'язуються один з одним за допомогою витої пари проводів. Довжина лінії зв'язку може досягати 1200 м (на одному сегменті мережі) залежно від встановленої швидкості передачі даних. Для збільшення довжини лінії зв'язку, а також кількості сегментів - використовують магістральні підсилювачі.

Мережа має багатоточкову конфігурацію і шинну топологію. Для побудови мережі контролерів, призначених для обміну інформацією з комп'ютером, необхідний додатковий пристрій, який використовується для перетворення інтерфейсів RS-232 в RS-485 - БПІ-485 (або USB в RS-485 - БПІ-52).

Якщо контролери розміщуються недалеко один від одного (наприклад, в межах одного приміщення), екранувати виту пару не потрібно. При значних відстанях між контролерами і наявності перешкод на лінії треба використовувати екрановану виту пару.

До інтерфейсних входів контролерів, розташованих у крайніх точках сполучної лінії необхідно підключити два термінальних резистора опором 120 Ом (R1 і R2). Підключення резисторів до контролерів №№ 01-30 не потрібно.

Логічна організація мережі має наступні можливості мережного обміну інформацією:

- 1) обмін інформацією між контролерами (комп'ютер є диспетчером мережного обміну);
- 2) обмін інформацією між контролерами і комп'ютером;
- 3) комбінована організація обміну, об'єднуюча всі переваги двох перших способів мережного обміну.

Кожному контролеру МК-51, підключеному до мережі, привласнюється унікальний логічний номер - мережна адреса або номер приладу в мережі. Цей номер встановлюється в процесі програмування на рівні конфігурування в процедурі «системні параметри». Порядок мережних адрес в мережі ніяк не пов'язаний з фізичним порядком включення контролерів. Мережні адреси можуть встановлюватися в будь-якій послідовності або мати пропуски. При установленні мережних адрес повинні виконуватися наступні правила:

- значення мережної адреси встановлюється в межах 0-255;
- контролер, який бере участь в обміні, не повинен мати мережної адреси, яка б дорівнювала 0. При N=0 контролер програмно відключений від мережі, і не реагує ні на які запити зовнішніх абонентів, але при цьому цілісність мережі зберігається;
- максимальна кількість контролерів, об'єднаних одним сегментом мережі, не повинна перевищувати 31. При використуванні магістральних підсилювачів кількість контролерів не повинна перевищувати 250;
- в одній мережі не повинно бути двох або більш контролерів, що мають однакову мережну адресу.

Обмін інформацією між контролерами здійснюється на рівні сигналів і/або параметрів функціональних блоків. Мережні адреси можуть встановлюватися в будь-якій послідовності або мати пропуски, посилаючи в мережу спеціальну команду про початок мережного обміну. Комп'ютер (або інший контролер) будучи диспетчером мережного обміну проводить синхронізацію і контроль над обміном по мережі. Контролер (диспетчер мережного обміну) повинен використовувати функціональний блок LAN\_SV(1).

При сполученні функціональних блоків через мережу є ряд особливостей:

- безпосередньо зв'язувати функціональні блоки через мережу не можна - для такого зв'язку використовуються два спеціальні функціональні блоки – блок інтерфейсного виводу LAN\_OUT(04) і блок інтерфейсного вводу LAN\_IN(03);

- якщо вихідні сигнали яких-небудь функціональних блоків повинні передаватися в інші контролери, відповідні виходи функціональних блоків за стандартною процедурою конфігурування підключаються до входів функціонального блока інтерфейсного виводу LAN\_OUT(04);

- якщо які-небудь функціональні блоки одного контролера повинні приймати сигнали від іншого контролера, їх входи за стандартним конфігуруванням з'єднуються з виходами блока інтерфейсного вводу LAN\_IN(03);

- один функціональний блок інтерфейсного вводу LAN\_IN(03) забезпечує зв'язок тільки з одним функціональним блоком інтерфейсного виводу LAN\_OUT(04) іншого контролера. Якщо даному контролеру необхідно зв'язатися з декількома контролерами, в ньому необхідно запрограмувати декілька функціональних блоків інтерфейсного вводу LAN\_IN(03);

- у функціональному блоці інтерфейсного вводу LAN\_IN(03) встановлюється мережна адреса контролера-джерела, тобто того контролера, від якого будуть одержані дані;

- в кожному контролері може встановлюватися будь-яке число функціональних блоків інтерфейсного вводу-виводу. Кожен контролер може передавати в мережу довільну кількість сигналів.

Ініціалізацію обміну інформацією між контролерами здійснює майстер-пристрій мережі - комп'ютер. У мережі можливий зв'язок між будь-яким контролером і комп'ютером.

Комп'ютер може взаємодіяти з контролером на наступних рівнях:

- на рівні входів-виходів функціонального блока;

- на рівні програмованих реєстрів – параметрів функціонального блока.

При обміні інформацією з функціональним блоком вказується номер блока, а при обміні з програмованими реєстрами - номер реєстра. Комп'ютер посилає контролеру запит або команду і одержує від контролера відповідно відповідь або підтвердження. Запит посилається комп'ютером при необхідності одержання від контролера (функціонального блока) якої-небудь інформації.

Одержавши запит, контролер готує необхідну інформацію і передає її комп'ютеру у вигляді відповіді. Команда посилається комп'ютером при необхідності зміни параметра настройки блока.

Одержавши команду, контролер її виконує і передає комп'ютеру підтвердження.

Комп'ютер при обміні інформацією з контролерами має ряд можливостей:

- запитати значення сигналу на будь-якому вході або виході будь-якого блока в будь-якому контролері;
- запитати значення будь-якого параметра настройки будь-якого блока в будь-якому контролері;
- змінити значення будь-якого параметра будь-якого блока в будь-якому контролері;
- здійснювати запуск та зупинення програмного задавача;
- запитати наявність помилок в будь-якому контролері. При роботі з оперативними параметрами комп'ютер може запрошувати і змінювати ті ж параметри, які спостерігає і змінює оператора, працюючи з лицевою панеллю контролера.

Комбінована організація обміну, об'єднує всі переваги двох перших способів мережного обміну:

- обмін інформацією між контролерами (через комп'ютер);
- обмін інформацією між контролерами і комп'ютером.

Логічна організація комбінованого обміну по мережі має наступну нагоду:

- при обміні інформацією між контролерами (комп'ютер є диспетчером мережного обміну), - контролери передають інформацію по мережі від одного - іншому за допомогою функціональних блоків мережного обміну;
- інформація, що передається (між контролерами) по мережі виявляється доступною і комп'ютеру.

Таким чином, комбінована організація обміну надає можливість зробити доступною інформацію при сеансі обміну між контролерами відразу декільком абонентам мережі - комп'ютеру і одному або декільком контролерам. Це скорочує час доступу до даних і період обміну по мережі.

Зв'язок комп'ютера з мережею контролерів організовується на трьох рівнях: фізичному, інформаційному і транспортному.

Фізичний рівень призначений для електричного зв'язку між комп'ютером і контролерами. На фізичному рівні використовується інтерфейс RS-485.

Інформаційний рівень додає повідомленням певний формат кадру, що передається - на передаючій стороні до повідомлень додає адресу пристрою, функціональний код операції і байти контрольної суми, а на приймальній стороні виділяє дані і аналізує контрольну суму повідомлення. Смісл повідомлення на інформаційному рівні не розшифровується.

Транспортний рівень формує смисловий зміст повідомлення при передачі і розшифровує значення цього повідомлення при прийомі.

Інтерфейс контролера призначений для програмування і конфігурування приладу, а також для використання в якості віддаленого контролера, при роботі в сучасних мережах управління і збору інформації (прийому-передачі команд і даних), SCADA системах і т.ін.

Протоколом зв'язку по інтерфейсу RS-485 є протокол MODBUS режим RTU (Remote Terminal Unit).

Для роботи необхідно настроїти комунікаційні характеристики контролера так, щоб вони співпадали з настройками обміну даними комп'ютера і інших контролерів. При обміні по інтерфейсному каналу, якщо відбувається передача даних від контролера в мережу, на лицевій панелі контролера блимає індикатор ІНТ.

Кількість запрошуваних регістрів контролера не повинна перевищувати 16. Якщо в кадрі запиту замовлено більше 16 регістрів, контролер МІК-51 обмежує їх кількість у відповіді до перших 16-ти регістрів.

При програмуванні з комп'ютера необхідно контролювати діапазони зміни значень параметрів.

Для забезпечення мінімального часу реакції на запит від ЕОМ в контролері існує параметр - «Тайм-аут кадру запиту в системних тактах контролера 1 такт = 250 мкс».

Час передачі кадру – запиту, пакету з 8-ми байт, визначається співвідношенням:

$$T_{\text{передачі}} = 1000 * \frac{10 \text{ біт} * 8 \text{ байт}}{V \text{ біт/сек.}}, \text{ мс,}$$

де: один байт, що передається = 1 старт біт + 8 біт + 1 стоп біт = 10 біт.

Якщо спостерігаються часті збої при передачі даних від контролера, то необхідно збільшити значення його тайм-ауту. При цьому потрібно збільшити час повторного запиту від ЕОМ, оскільки завжди час повторного запиту повинен бути більше тайм-ауту контролера[22].

#### 4.4 Програмування, настройка і контроль

Програмування контролера МІК-51 здійснюється за допомогою клавіш і індикаторів передньої панелі або по інтерфейсу з використанням спеціального програмного пакету – візуального редактора FVD- програм «АЛЬФА».

Лицева панель контролера МІК-51 у режимі програмування і настройки представлена на рис 4.6.

Світлодіодні індикатори сигналізують про режим роботи, вибрану процедуру і помилки. На цифрові семисегментні індикатори виводяться номери і значення параметрів.

У режимі програмування задаються всі програмовані параметри контролера, що визначають його функціональну структуру.

На дисплеї ПАРАМЕТР індичується два параметри (П№1,П№2)



Рисунок 4.6 – Лицева панель контролера МІК-51 у режимі програмування

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

- Перший рівень* - налаштування кількості блоків;
- Другий рівень* - програмування функціональних блоків;
- Третій рівень* - програмування зв'язків функціонального блоку;
- Четвертий рівень* - налаштування параметрів функціональних блоків;
- П'ятий рівень* - тестування й калібрування входів-виходів;
- Шостий рівень* - системні параметри контролера;
- Сьомий рівень* - контроль відмов;
- Восьмий рівень* - збереження програми користувача і налаштувань в енергонезалежній пам'яті.

Процес програмування зводиться до того, що на кожному рівні шляхом послідовного натискання декількох кнопок з бібліотеки, записаної в постійній пам'яті, викликаються потрібні функціональні блоки, ці функціональні блоки об'єднуються в систему заданої конфігурації і в них встановлюються необхідні параметри налаштування.

За допомогою вбудованої незалежної пам'яті запрограмована інформація зберігається при відключенні живлення.

Коли використовується програмний пакет «АЛЬФА», FBD-програма створюється шляхом розміщення функціональних блоків у робочому полі і сполучення їх в одну діаграму. Функціональні блоки вибираються у бібліотеці,

здійснюється програмування базових адрес їх параметрів, міжблокових сполучень, а також налаштування параметрів блоків.

Налагодження розробленої програми може відбуватися на емуляторі. У цьому режимі програма користувача перераховується у неперервному режимі із заданим періодом.

При використанні програмного пакету для організації послідовного зв'язку між ЕОМ і контролером має бути підключений поміж ними перетворювач інтерфейсів RS 232/RS485[22,23].

#### 4.4.1 Методика програмування контролера за допомогою інструментальної системи «Альфа»

Програмування контролера за допомогою інструментальної системи «Альфа» здійснити на прикладі ПД-імпульсного алгоритму, який керує роботою виконавчого механізму МЕО. Контролер циклічно імітує вхідний аналоговий сигнал відповідно до графіку на рис 4.7 за допомогою функціональних блоків

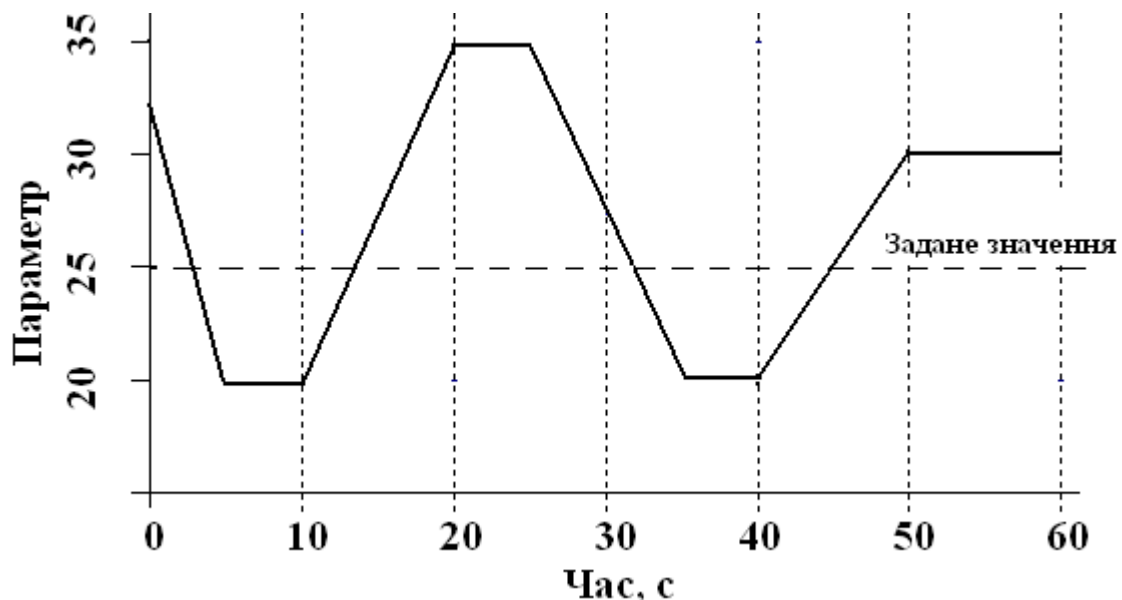


Рисунок 4.7 – Графік зміни параметру керування відносно заданого значення

TM\_M(56), TM\_PRG(57) і SP\_M(54) і здійснює керувальний вплив на виконавчий механізм за ПД-імпульсним PID\_IMP(62) законом регулювання, 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 зовнішніми функціональними блоками, відповідно TM\_M(56) і SP\_M(54).

Для створення програмного коду необхідно розробити табличний варіант програми, табл.4.3, користуючись відповідними розділами бібліотеки і описом функціональних блоків у керівництві по експлуатації МІК-51.



Таблиця 4.3

## Програма керування роботою виконавчого механізму


| Алгоритми (2 рівень) |         |    |           |      | Конфігурація (3 рівень) |      |      |       |            | Настроювання (4 рівень) |                 |           |
|----------------------|---------|----|-----------|------|-------------------------|------|------|-------|------------|-------------------------|-----------------|-----------|
| № Бло-ка             | шифр    | №  | Баз. адр. | Мод. | № бл                    | № вх | № бл | № вих | Ім'я входу | № пара-метра            | Ім'я пара-метра | Значен-ня |
| 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         |
|                      |         |    |           |      |                         |      |      |       |            | 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         |

| Алгоритми |      |    |           |      | Конфігурація |      |      |       |            | Настроювання |                 |           |
|-----------|------|----|-----------|------|--------------|------|------|-------|------------|--------------|-----------------|-----------|
| № Бло-ка  | шифр | №  | Баз. адр. | Мод. | № бл         | № вх | № бл | № вих | Ім'я входу | № пара-метра | Ім'я пара-метра | Значен-ня |
|           |      |    |           |      |              |      |      |       |            | 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   |
|           |      |    |           |      |              |      |      |       |            | 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         |
|           |      |    |           |      |              |      |      |       |            | 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. Перевести контролер у режим «Робота», натиснути кнопку "Ввід" і клавішами ▲,▼ обнулити 1-ий і 2-ий розряди дисплея ПАРАМЕТР. Знову натиснути кнопку "Ввід" і обнулити 3-ий і 4-ий розряди дисплея ПАРАМЕТР. Натиснути кнопку "Ввід" і обнулити розряди решти цифрових дисплеїв (після кожного обнуління результат підтверджується натисненням клавіші "Ввід").

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




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

4. При необхідності відкриття раніше створеного проекту:

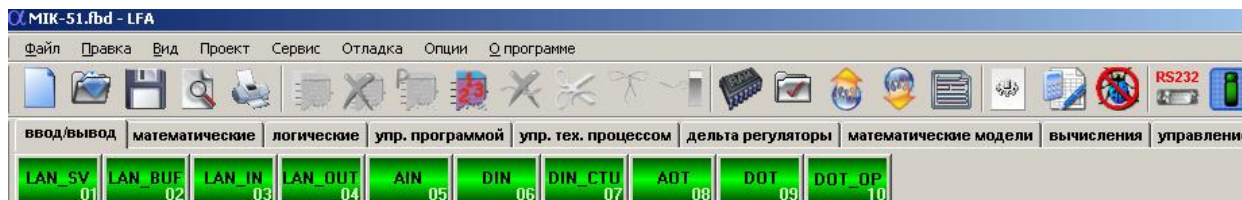
- натиснути кнопку «Відкрити» або вибрати команду «Відкрити»  в меню «Файл»;
- у полі «Тип файлів» вибрати необхідний формат файлу \*.fdb;
- щоб відкрити проект, збережений в іншій теці, знайти і відкрити потрібну теку;
- клацнути по проекту, який потрібно відкрити в ALFA;
- натиснути кнопку «Открыть».

5. Для збереження проекту:

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

### *Створення програмного коду*

Вибір необхідних функціональних блоків здійснюється на панелі інструментів з відповідного розділу бібліотеки, табл. 4.4.



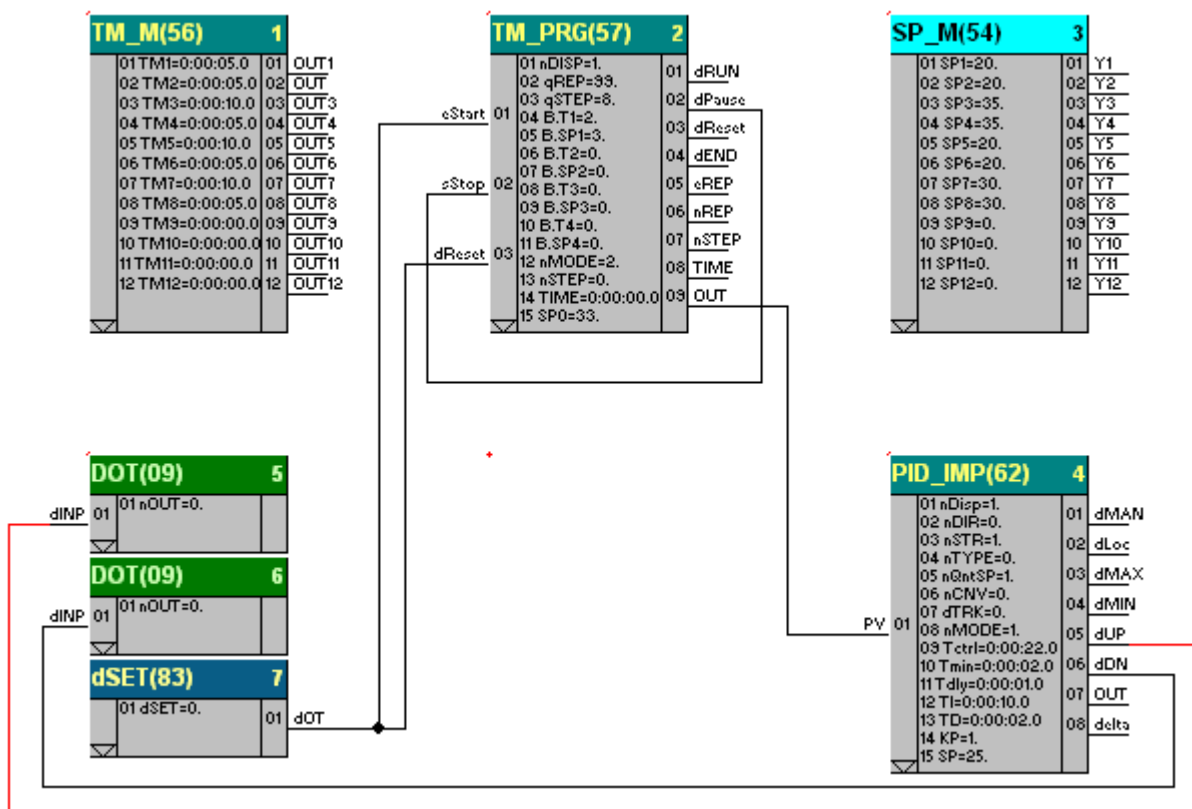
Щоб вибрати функціональний блок, необхідно увійти у відповідний розділ бібліотеки і клацнути ліву клавiшу миші на потрібному блоці. Для установлення вибраного блока на робочому полі потрібно повторно клацнути лівою клавiшею у вільній зоні екрану.

Сполучення блоків здійснюється за допомогою миші:

- установити курсор на відповідний вхід блока і після появи на вході кружка - натиснути ліву клавiшу миші;
- пересунути курсор з натиснутою лівою клавiшею миші на відповідний вихід потрібного блока (джерела сигналу) і після появи кружка відпустити клавiшу;
- за необхідністю інверсії вхідного сигналу, встановити курсор миші на вхід блока і натиснути на клавіатурі EOM клавiшу CTRL та ліву кнопку миші.

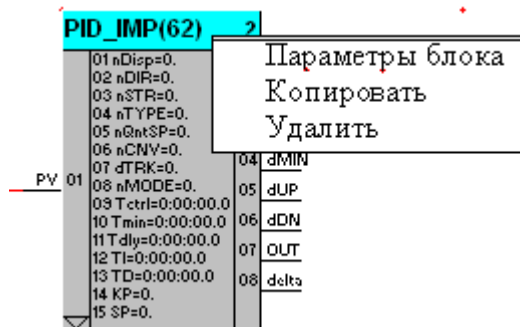
| Розділ | Найменування розділу                                 | Перелік функціональних блоків  |
|--------|--|--|
| 1      | Функціональні блоки вводу-виводу                     | Інтерфейсний ввід-вивід, аналоговий ввід-вивід, дискретний ввід-вивід, імпульсний ввід   |
| 2      | Математичні функціональні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

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



Для настроювання параметрів функціональних блоків:

- установити курсор на відповідний блок і натиснути ліву клавішу миші, при цьому блок підсвічується;
- натиснути праву клавішу миші на виділеному блоці і з меню, що з'явиться, вибрати пункт "Параметри блока":




- у діалоговому вікні редагування параметрів блока виконати відповідні настроювання, користуючись табличною формою програми.

| Параметры блока     |        |   |           |                  |   |
|---------------------|--------|---|-----------|------------------|---|
| PID_IMP(62)         |        |   |           |                  |   |
| Параметры алгоблока |        | Базовый адрес блока                       | 62        | Порядковый номер | 4 |
| Номер               | Шифр   | Название                                  | Значение  | Связанный п-р    |   |
| 1                   | nDisp  | Номер дисплея                             | 1.000     |                  |   |
| 2                   | nDIR   | Направление действия регулятора           | 0.000     |                  |   |
| 3                   | nSTR   | Структура регулятора(паралельный, п...    | 1.000     |                  |   |
| 4                   | nTYPE  | Тип регулятора: 0-без ОС, 1-с датчиком... | 0.000     |                  |   |
| 5                   | nQntSP | Количество используемых заданий (1-в...   | 1.000     |                  |   |
| 6                   | nCNV   | Тип преобразования                        | 0.000     |                  |   |
| 7                   | dTRK   | Команда установки на выходе регулят...    | 0.000     |                  |   |
| 8                   | nMODE  | Режим работы                              | 1.000     |                  |   |
| 9                   | Tctrl  | Время механизма                           | 0:00:22.0 |                  |   |
| 10                  | Tmin   | Минимальная длительность импульса         | 0:00:02.0 |                  |   |
| 11                  | Tdly   | Задержка между включениями                | 0:00:01.0 |                  |   |
| 12                  | TI     | Время интегрирования                      | 0:00:10.0 |                  |   |
| 13                  | TD     | Время дифференцирования                   | 0:00:02.0 |                  |   |
| 14                  | KP     | Кoeffициент усиления                      | 1.000     |                  |   |
| 15                  | SP     | Задание                                   | 25.000    |                  |   |


Аналогічно зробити настроювання параметрів інших блоків.

Нумерація блоків виконується в порядку їх установлення.

При необхідності можна змінити нумерацію:

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

Розподіл параметрів блока в регістровій області пам'яті здійснюється:

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

Перевірка програми здійснюється за допомогою кнопки




(Перевірити програму), що на панелі інструментів. При цьому виконується перевірка програми користувача на відповідність існуючим правилам.

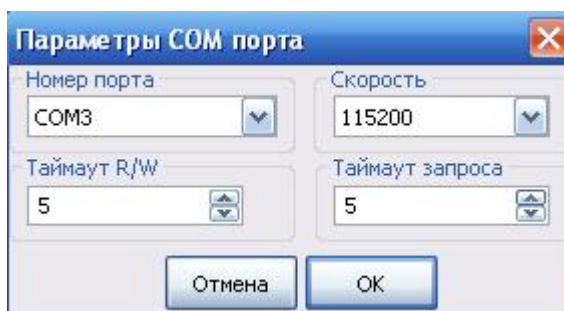
## Налагодження




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

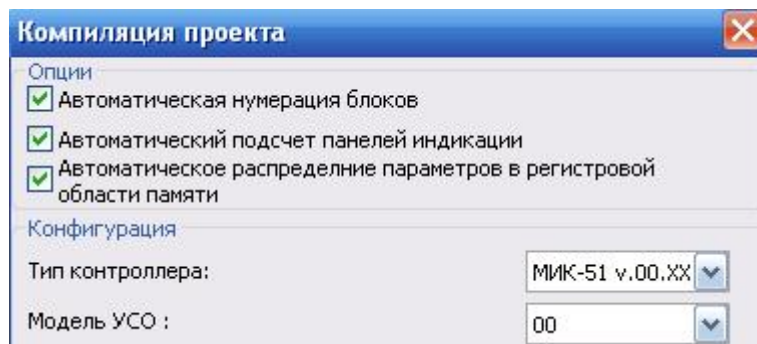
### Запис і читання програми з контролера


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

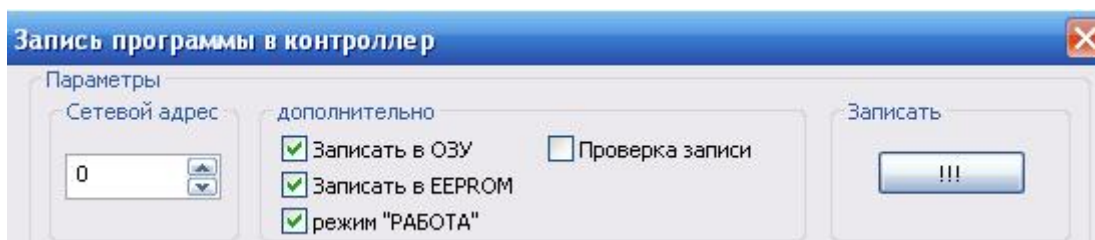


Запис, читання програми здійснюється за відповідною командою в меню «Действия» або, користуючись кнопками  на панелі інструментів.

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



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



Активізувати необхідні пункти запису і виконання програми та, указавши адресу пристрою, натиснути кнопку «Записать». Після завершення процесу запису, на дисплеї контролера висвітяться символи «ЗП».

Запуск контуру регулювання відбувається після переводу контролера у стан зупинення виконання програми.

Для зупинення роботи контролера, кнопкою "Меню" вибрати панель індикації "Прг". Натиснути кнопку "P/A", почне мигати індикатор "Стоп" (червоний квадрат). Натиснути кнопку "Ввід".

Для запуску програми натиснути кнопку P/A, почне миготіти червоний квадрат. Після 2-разового додатково натиснення кнопки P/A, миготіння перейде на зелений трикутник. Кнопкою "Ввід" підтвердити дану команду і відслідковувати виконання програми на дисплеях контролера та роботі виконавчого механізму МЕО [23].

#### **Тестові та контрольні запитання до розділу 4**

**I-й рівень запитань і тверджень дозволяє перевірити загальний рівень засвоєння матеріалу розділу і передбачає відповідь на запитання або твердження "Так" чи "Ні" .**

1. Виробником контролерів МІК-51 є українське підприємство "Мікрол".
2. Контролер МІК-51 являє собою компактний багатоканальний мікропроцесорний контролер, призначений тільки для логічного управління технологічними процесами.
3. Контролер МІК-51 складається з центрального мікропроцесорного блока, клемноблокових з'єднувачів та модулів розширення.
4. Базова модель контролера має 2 аналогові входи, 2 аналогові виходи, 4 дискретні входи та 4 дискретні виходи.
5. Модулі розширення контролера МІК-51 можуть бути різних модифікацій.
6. Для з'єднання контролерів МІК-51 з датчиками і виконавчими механізмами призначені клемноблокові з'єднувачі.
7. Контролери МІК-51 об'єднуються в мережу конфігурації «зірка».
8. У мережі може бути до 31-го контролера у тому числі інших моделей.
9. МІК-51 має інтерфейсний канал RS-232.
10. Обмін інформацією між контролерами ініціалізує комп'ютер.
11. Підключення контролера до ЕОМ здійснюється без перетворювача інтерфейсів.
12. Засоби оперативного управління МІК-51 розташовані на боковій панелі контролера.
13. Клавіша «Ввід» на панелі контролера призначена для підтвердження виконуваних дій, операцій і для фіксації значень, що вводяться.
14. У режимі «Програмування» здійснюється оперативне або автоматичне управління об'єктом автоматизації.
15. МІК-51 має 99 функціональних блоків.
16. Функціональним блокам автоматично присвоюються порядкові номери, які визначають послідовність їх виконання у програмі.
17. У кожному функціональному блоці використовується модифікатор.

18. Чи програмується контролер МІК-51 з лицевої панелі?

19. Контролер МІК-51 програмується за допомогою програмного пакета «Альфа».

20. Чи зберігається у пам'яті контролера запрограмована інформація коли живлення вимикається?

*Перевірте відповіді на сторінках: 1 - 80 с., 2 - 80 с., 3 - 80 с., 4 - 80 с., 5 - 80 с., 6 - 84 с., 7 - 91 с., 8 - 91 с., 9 - 90 с., 10 - 92 с., 11 - 91 с., 12 - 80 с., 13 - 82 с., 14 - 94 с., 15 - 85 с., 16 - 85 с., 17 - 86 с., 18 - 91 с., 19 - 94 с., 20 - 95 с.*

**II-й рівень запитань (тестові запитання) дозволяє перевірити рівень засвоєння матеріалу розділу із заглибленням у технічні особливості побудови та структури контролера МІК-51 та передбачає вибір правильної відповіді з наведених трьох варіантів.**

**1. Скільки незалежних контурів регулювання передбачено в МІК-51?**

- а) чотири;
- б) дев'ять;
- в) шість.

**2. Що входить до архітектури контролера МІК-51?**

а) апаратура вводу-виводу інформації, апаратура оперативного управління і настройки, апаратура інтерфейсного каналу, функціональні блоки і бібліотека функціональних блоків;

б) апаратура вводу-виводу інформації, апаратура оперативного управління і настройки, апаратура інтерфейсного каналу;

в) функціональні блоки і бібліотека функціональних блоків.

**3. За допомогою чого в МІК-51 можна виконувати оперативне управління та контроль за контурами регулювання?**

а) за допомогою клавіш лицевої панелі; 2-х чотирирозрядних, одного трирозрядного і одного однорозрядного цифрових індикаторів, а також набору світлодіодів;

б) за допомогою клавіш лицевої панелі та 2-х чотирирозрядних цифрових індикаторів;

в) за допомогою клавіш лицевої панелі; одного трирозрядного і одного однорозрядного цифрових індикаторів, а також набору світлодіодів.

**4. Скільки клавіш має лицева панель МІК-51?**

- а) дванадцять;
- б) вісім;
- в) шість.

**5. Яке управління може здійснювати контролер МІК-51?**

- а) каскадне та локальне, ручне та програмне, слідкувальне управління;
- б) каскадне, ручне та програмне, слідкувальне управління;
- в) каскадне та локальне, ручне та програмне управління.



**6. Які існують режими роботи контролера і для чого вони використовуються?**

а) режим «Робота» дозволяє здійснювати оперативне або автоматичне управління об'єктом автоматизації; режим «Програмування», в якому контролер програмується і встановлюються параметри алгоблоків та режим «Конфігурування», в якому здійснюються сполучення між алгоблоками та модулями розширення;

б) режим «Робота» дозволяє здійснювати оперативне або автоматичне управління об'єктом автоматизації і режим «Програмування», в якому контролер програмується і встановлюються параметри алгоблоків;

в) режим «Робота» дозволяє здійснювати тільки автоматичне управління об'єктом автоматизації; режим «Програмування», в якому контролер програмується та режим «Конфігурування», в якому забезпечуються сполучення між алгоблоками та модулями розширення і встановлюються параметри алгоблоків.

**7. Скільки і які входи має базова модель контролера?**

а) по 2 аналогових входи-виходи і по 4 дискретних входи-виходи;

б) 4 дискретні входи та 4 дискретні виходи;

в) 4 аналогові входи, 1 аналоговий вихід, 3 дискретні входи і 5 дискретних виходів.

**8. Скільки функціональних блоків «зашите» у ПЗП МК-51?**

а) 99;

б) більше 50;

в) більше 100.

**9. Що можна робити за допомогою лицевої панелі МК-51?**

а) виконувати настроювання параметрів контролера та контролювати сигнали у внутрішніх точках структури;

б) програмувати контролер, виконувати настроювання його параметрів, контролювати сигнали у внутрішніх точках структури;

в) тільки контролювати сигнали у внутрішніх точках структури.

**10. Для яких функціональних блоків МК-51 є обмеження на кратність їх використання в межах одного контролера?**

а) для блоків виводу;

б) для блоків вводу;

в) для блоків вводу-виводу.

**11. Яке правило діє при конфігуруванні функціональних блоків?**

а) для виходів задається номер функціонального блока і номер входу, до якого приєднується даний вихід;

б) для входів та виходів задаються номери функціональних блоків і номер виходу/входу, до якого вони приєднуються;

в) для входів задається номер функціонального блока і номер виходу, до якого приєднується даний вхід.

**12. Скільки контролерів може бути в одному сегменті мережі?**

а) 31;

- б) 12;
- в) 255.

**13. До яких контролерів мережі підключаються термінальні резистори?**

- а) до кожного контролера, розташованого у мережі;
- б) до контролерів, розташованих у крайніх точках сполучної лінії;
- в) до контролера, який розташований на найбільшій відстані від ЕОМ.

**14. В якій послідовності контролерам встановлюються мережні адреси?**

- а) у будь-який;
- б) відповідно до розташування у мережі;
- в) мережні адреси встановлюються автоматично при приєднанні контролера до мережі.

**15. Як сполучаються функціональні блоки через мережу?**

- а) за допомогою клемно – блокових з'єднувачів;
- б) за допомогою функціональних блоків інтерфейсного вводу LAN\_IN і виводу LAN\_OUT;
- в) функціональні блоки через мережу зв'язуються безпосередньо.

**16. Який протокол використовується для реалізації сполучення по інтерфейсу RS-485?**

- а) MODBUS режим RTU;
- б) MODBUS;
- в) TCP/IP.

**17. Як контролер МК-51 формує імпульсні вихідні сигнали?**

- а) формування імпульсних сигналів здійснюється програмно на виході імпульсного регулятора і ці сигнали поступають на виконавчі механізми через імпульсні виходи контролера;
- б) формування імпульсних сигналів здійснюється через імпульсні виходи контролера;
- в) формування імпульсних сигналів здійснюється програмно на виході імпульсного регулятора і ці сигнали поступають на виконавчі механізми через дискретні виходи контролера.

**18. Скільки процедур програмування має контролер МК-51?**

- а) чотири;
- б) шість;
- в) вісім.

**19. За допомогою якого пристрою забезпечується послідовний зв'язок між ЕОМ і контролером при використанні програмного пакету Альфа?**

- а) за допомогою перетворювача інтерфейсів RS 232/RS485;
- б) для підключення контролера до ЕОМ додаткового пристрою не потрібно;
- в) за допомогою перетворювача інтерфейсів Ethernet/RS485.

**20. Якою мовою створюється програма для МК-51?**

- а) FBD та LD;
- б) FBD та ST;
- в) FBD.

*Перевірте відповіді на сторінках: 1 - 80 с., 2 - 84 с., 3 - 81 с., 4 - 82 с., 5 - 80 с., 6 - 82...83 с., 94...95 с., 7 - 80 с., 8 - 85 с., 9 - 81 с., 10 - 89 с., 11 - 89 с., 12 - 91 с., 13 - 91 с., 14 - 91 с., 15 - 92 с., 16 - 94 с., 17 - 84 с., 18 - 95 с., 19 - 96 с., 20 - 94 с.*

**III-й рівень питань (контрольні питання) дозволяє перевірити рівень засвоєння матеріалу із систематизацією отриманої у розділі інформації та передбачає відповіді, що побудовані на логічному аналізі або порівнянні із приведенням конкретних технічних параметрів (даних або прикладів).**

1. Накресліть структурну схему контролера МІК-51 та поясніть призначення основних елементів апаратної частини.
2. Розкрийте призначення усіх елементів лицевої панелі МІК-51 та вкажіть який стан або режим контролера вони характеризують.
3. Які функції можна виконувати вручну при налагодженні та роботі МІК-51?
4. Поясніть як взаємодіють апаратна частина та віртуальна структура контролера МІК-51.
5. Розкрийте принцип налагодження роботи функціональних блоків контролера.
6. Поясніть особливості, які є при сполученні функціональних блоків через мережу.
7. Які мережні технології закладено в контролері МІК-51 для організації обміну інформацією у складі АСУ ТП.
8. Поясніть послідовність виконання процедур програмування МІК-51 і, які операції можна виконувати у кожній з процедур?
9. Дайте порівняльну характеристику між структурами контролерів Р-130 та МІК-51.
10. Розкрийте основні можливості та принцип роботи у середовищі Альфа.

*Перевірте відповіді на сторінках: 1 – 80...94 с., 2 – 81...83 с., 3 – 81...83 с., 4 – 84...90 с., 5 – 85...90 с., 6 - 92 с., 7 – 90...94 с., 8 - 95 с., 9 – 80...94, 54...68 с., 10 – 96...103 с.*

## 5 ПРОГРАМОВАНІ КОНТРОЛЕРИ КОРПОРАЦІЇ ADVANTECH

Тайванська корпорація Advantech утворена у 1983 році і спеціалізується на виробництві обчислювальної техніки для промислової автоматизації. На підприємствах машинобудування, металургії, легкої промисловості багатьох держав добре відомі промислові комп'ютери, програмовані логічні контролери, пристрої збору та передачі даних, що створені спеціалістами цієї фірми. При впровадженні автоматизованих систем управління технологічними процесами широко використовуються засоби автоматизації цієї фірми серій ADAM 4000, ADAM 5000, ADAM 6000, ADAM 8000.

Модулі серії ADAM 4000 є компактними і інтелектуальними пристроями обробки сигналів датчиків і призначені для побудови розподілених систем збору даних і управління на основі інтерфейсу RS-485.

До складу серії ADAM 5000 входять системи розподіленого вводу-виводу даних і управління, а також програмовані контролери. Всі вони забезпечують до 128 каналів дискретного вводу-виводу або 64 аналогові канали на один блок ADAM 5000. Пристрої серії ADAM 5000 можуть об'єднуватися у багатоточкову мережу на базі інтерфейсу Ethernet або RS-485.

Модулі серії ADAM 6000 призначені для побудови інтелектуальних розподілених систем збору даних і управління на основі інтерфейсу Ethernet. Використання Ethernet дозволяє легко інтегрувати системи на основі модулів ADAM 6000 в мережі Інтернет/інтранет. Здійснюється це шляхом організації Web-доступу до модулів збору даних у реальному часі за допомогою вбудованого в кожен з них Web-серверу. Для організації взаємодії з SCADA-системами верхнього рівня в модулях реалізована підтримка протоколу Modbus/TCP і обмін даними відбувається через OPC-сервер.

Вироби серії ADAM 8000 є новим кроком, зробленим корпорацією Advantech у напрямі освоєння ринку класичних ПЛК-систем. Нове покоління ADAM можуть з успіхом використовуватися у системах промислової автоматизації з підвищеними вимогами до надійності устаткування і до тимчасових параметрів контурів управління. Ці пристрої працюють в промислових мережах MPI, PROFIBUS-DP, ModBus TCP і CAN.

Контролери серії 8000 повністю програмно сумісні з популярними контролерами серії S7-300 фірми Siemens, причому функціональні можливості модулів ADAM 8000 не поступаються оригіналу, а конструктивно нові модулі набагато компактніше.

Серед програмованих логічних контролерів виробництва фірми Advantech найбільш поширеними у промисловості є ПЛК серії ADAM 5000, які складаються з трьох модульних компонентів - процесор, крос плата, модулі вводу-виводу. Кожен пристрій може містити до 4 або до 8 модулів вводу-виводу.

Контролери серії ADAM 5000 умовно можна поділити на дві групи – контролери, що програмуються мовами високого рівня – це контролери групи ADAM 5510, і контролери, програмування яких здійснюється за допомогою

Softlogic-системи KW MULTIPROG мовами міжнародного стандарту IEC 61131-3 – це контролери групи ADAM 5510KW. До першої групи відносяться контролери ADAM 5000, ADAM 5510, ADAM 5510M, ADAM 5510E, ADAM 5510/TCP, ADAM 5510E/TCP, ADAM 5511, а до другої - ADAM 5510KW, ADAM 5510E/KW, ADAM 5510 EKW/TP, ADAM 5510KW/TCP, ADAM 5510EKW/TCP, ADAM 5550. Контролери кожної групи мають багато спільного, різниця між ними визначається тільки об'ємом пам'яті, кількістю слотів та мережним інтерфейсом.

Окрему групу складають контролери ADAM 5510/HC, ADAM 5510/HCG, які мають вбудовану SCADA-систему Trace Mode. Ці контролери зберегли всі переваги серії ADAM 5000 і придбали додаткові можливості інтегрування систем верхнього і нижнього рівня. Вони підтримують всі функції, що характерні для Trace Mode і їх можна програмувати у тому ж середовищі, що і операторську робочу станцію.

### **5.1 Мікропроцесорний контролер ADAM 5510**

На початку 1998 року фірма Advantech приступила до випуску ADAM 5510, продовжуючи лінію інтелектуальних ПЗО серії ADAM-5000. До серії базових блоків ADAM 5000/485 і ADAM 5000/CAN, орієнтованих на роботу з управляючою машиною верхнього рівня в режимі «питання — відповідь» по одному з популярних промислових інтерфейсів, додався ще один — програмований ADAM 5510.

PC-сумісний програмований контролер ADAM 5510 призначений для використання в локальних розосереджених системах автоматизації у якості автономного контролера. ADAM 5510 забезпечує прийом та видачу аналогових і дискретних сигналів, первинне перетворення їх відповідно до запрограмованих користувачем алгоритмів та обмін інформацією за послідовними каналами зв'язку на базі інтерфейсу RS-485. Мікроконтролер ADAM 5510 має модульний принцип будови, рис.5.1. Основою контролера є базовий блок, в який можуть бути встановлені до 4-х модулів вводу-виводу серії ADAM 5000 із загальною кількістю каналів до 64. Габаритні розміри базового блока складають 110×231×75мм (В×Ш×Г).

Базовий блок контролера складається з процесорного модуля, перетворювача напруги, 4-слотової пасивної панелі, трьох послідовних комунікаційних портів.

Процесорний модуль створений на базі мікропроцесора 80188, який працює під керівництвом операційної системи ROM-DOS на рівні базових функцій, за виключенням BIOS, що дає можливість створення стандартного для IBM-PC програмного забезпечення або прикладних програм, написаних мовою високого рівня "C".

У складі процесорного модуля є флеш-ПЗП об'ємом 512 Кбайт і статичне ОЗП об'ємом 256 Кбайт. Для зберігання прикладної програми, її використання та зберігання даних призначено 170 Кбайт флеш-ПЗП. Крім того,

для виконання прикладних програм та пересилання файлів доступні 192 Кбайт оперативної пам'яті.

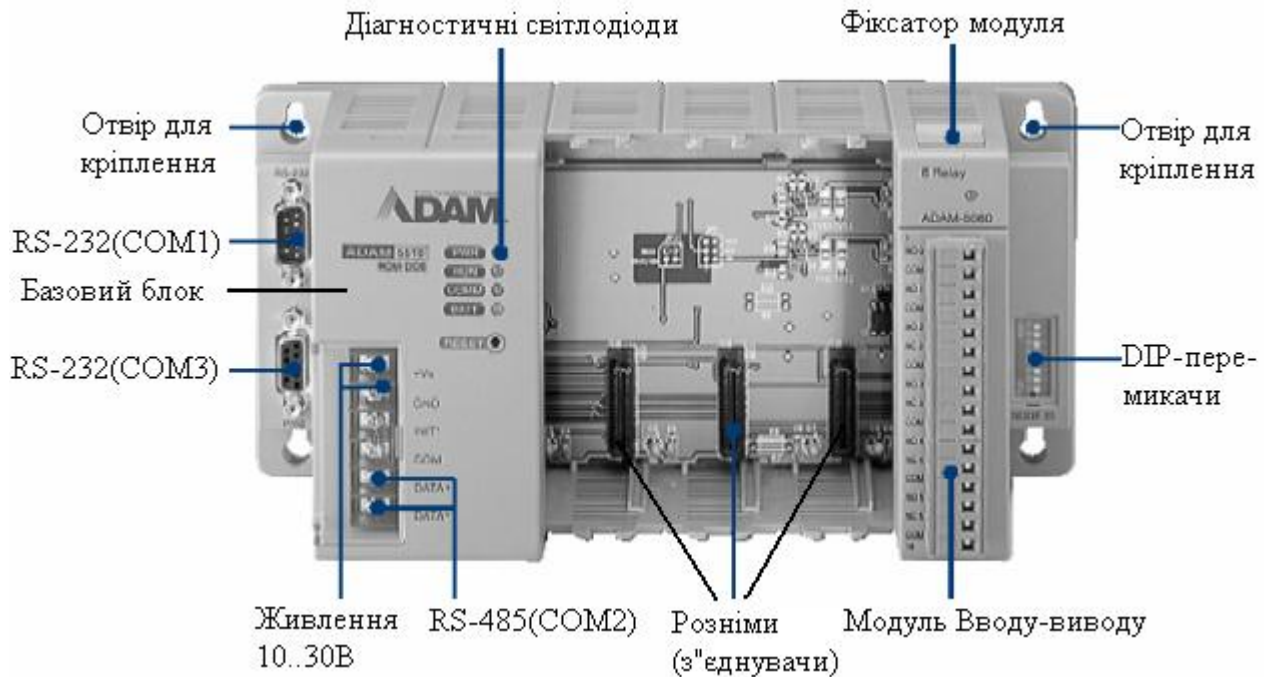


Рисунок 5.1 – Мікропроцесорний контролер ADAM-5510

Функціональна блок-схема контролера приведена на рис.5.2.

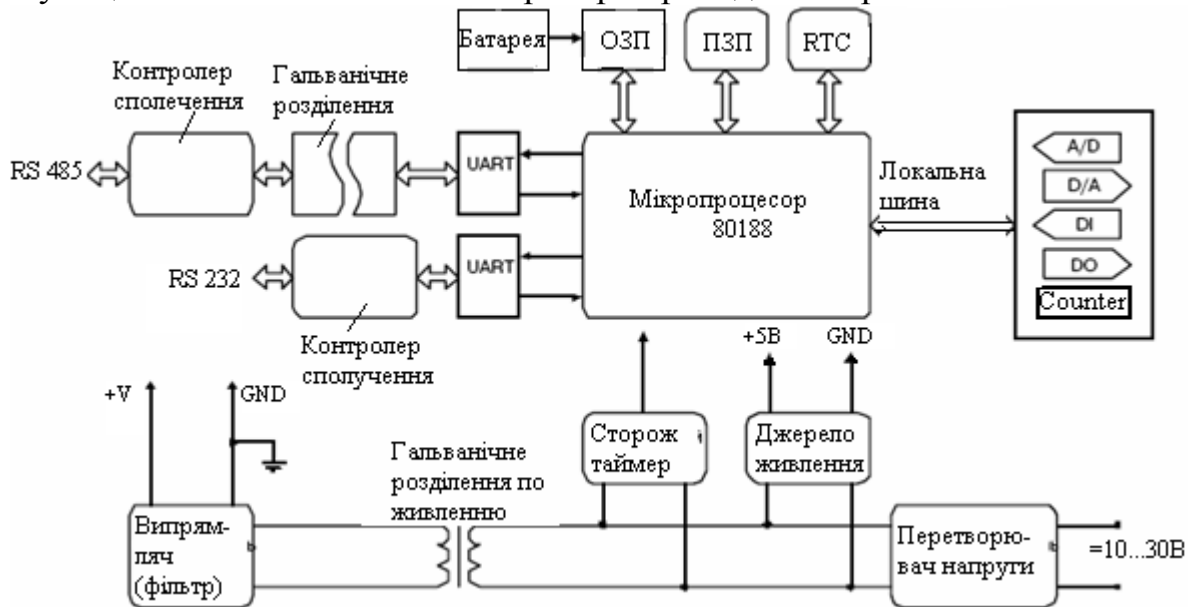


Рисунок 5.2 – Функціональна блок-схема ADAM 5510

Мікроконтролер ADAM 5510 містить три послідовних комунікаційних порти, за допомогою яких можлива взаємодія його з будь-якими пристроями, що мають послідовний доступ. Порт COM1 підтримує роботу в режимі інтерфейсу RS-232, а порт COM2 – роботу в режимі інтерфейсу RS-485. Порт COM3 призначений для здійснення програмування, конфігурування та діагностики контролера ADAM 5510 за допомогою керувального комп'ютера.

Крім того, він може бути використаним у якості комунікаційного порту загального призначення з інтерфейсом RS-232. Максимальна довжина лінії зв'язку, що обслуговують порти COM1 і COM3 (інтерфейс RS-232) – 15.2м; порт COM2 (інтерфейс RS-485) – 1220м. Швидкість обміну інформацією до 115200 біт/с.

Годинники реального часу, які є у складі мікроконтролера, забезпечують можливість прив'язки до часу будь-яких подій, що фіксуються у об'єкті керування. Сторожовий таймер призначений для здійснення повторного запуску мікроконтролера у випадку непередбаченого перебою у роботі керувальної програми.

Установлення мережної адреси контролера здійснюється 8-позиційним DIP-перемикачем, який розташовано праворуч на лицевій стороні базового блока. Адреса контролера може мати значення у межах від 0 до 255.

Живлення мікроконтролера ADAM 5510 здійснюється напругою  $\approx 24\text{В}$ . Для цього може використовуватися будь-яке джерело з вихідною напругою в діапазоні 10-30В постійного струму.

На лицевій панелі ADAM 5510 розташовані чотири світлодіодних індикатори, які призначені для сигналізації станів блока:

- світіння індикатора POWER червоного кольору – є напруга живлення;
- світіння індикатора RUN зеленого кольору – йде процес початкового завантаження;
- світіння індикатора COMM оранжевого кольору – відбувається обмін за послідовним каналом зв'язку;
- світіння індикатора BAT жовтого кольору – розряджений стан батареї.

Модулі вводу-виводу приєднуються до базового блока за допомогою рознімів, що закріплені в його слотах. Слоти нумеруються від 0 до 3, а нумерація каналів будь-якого модуля починається з 0. Дискретні модулі забезпечують ввід та вивід дискретних сигналів у діапазоні 10-30В постійного струму, а також релейну комутацію навантажень різної потужності. Модулі аналогового вводу завдяки АЦП перетворюють у цифровий код сигнали з термопар або термометрів опору, а також аналогові сигнали у вигляді напруги та струму. Цифрова форма аналогового сигналу потім перетворюється у формат інженерних одиниць. Модулі аналогового вводу мають оптичну гальванічну розв'язку по вхідних колах і трансформаторну гальванічну розв'язку по колах живлення з напругою ізоляції 3000В постійного струму. Існують аналогові модулі, що мають відповідно три канали вводу сигналів термометрів опору (ADAM-5013), 8 каналів аналогового вводу з диференційними входами (ADAM-5017), 7-каналів аналогового вводу сигналів термопар з компенсацією температури холодних кінців (ADAM- 5018), а також 4-канали аналогового виводу (ADAM-5024). Використовуючи сервісне програмне забезпечення, користувач може задати швидкість наростання вихідного сигналу, його початкове значення і тип (струм або напруга), установлення вхідного або вихідного робочого діапазону.

Серед дискретних є 8- (ADAM-5052) і 16- (ADAM-5051) каналні модулі вводу, 16- каналний модуль виводу (ADAM-5056), вихідні канали якого виконані у вигляді транзисторних ключів, є 6- (ADAM-5060) і 8- (ADAM-5068) каналні модулі з релейними виходами. Існує також 16-каналний модуль дискретного вводу-виводу (ADAM-5050), кожний канал якого може бути незалежно конфігурований в якості вхідного або вихідного за допомогою DIP-перемикачів, що розташовані на платі модуля. Вихідні каскади каналів виконані у вигляді транзисторних ключів за схемою з відкритим колектором, який забезпечує безпосереднє керування малопотужним навантаженням, а при застосуванні реле і керування виконавчими механізмами великої потужності.

Крім того ще існує 4-каналний модуль вводу частотних-імпульсних сигналів (ADAM- 5080).

Для підключення зовнішніх пристроїв до модулів контролера використовуються гвинтові клемні колодки. Завдяки тому, що вони роз'ємні суттєво спрощується монтаж проводів і налагоджувальні роботи.

ADAM 5510 має трирівневу гальванічну ізоляцію як для кіл живлення, так і для модулів вводу-виводу з напругою ізоляції 3000В постійного струму, а для портів послідовного зв'язку з напругою ізоляції 2500В постійного струму. Наявність гальванічного розділення дозволяє знизити вплив на систему електромагнітних перешкод, усунути гальванічний зв'язок з електрообладнанням об'єкта керування, а також запобігти несправностей, які можуть з'явитися під впливом випадкових змін напруги в колах живлення, а також перехідними процесами при комутації силового обладнання.

При монтажі мікроконтролер може бути установлений на вертикальну панель або на DIN-рейку[24].

### **5.1.1 Оновлення контролера ADAM 5510**

Оновленням контролера ADAM 5510 є контролер ADAM 5511, в якому реалізована програмна підтримка популярного протоколу ModBus, що дозволяє обмінюватися даними з будь-яким програмним забезпеченням верхнього рівня (SCADA) без використання спеціальних драйверів. Крім того, ADAM 5511 забезпечує можливість віддаленого завантаження, запуску, зупинення і завершення програм.

Контролер ADAM 5511 має такий самий вигляд як і ADAM 5510. Процесор контролера CPU 80188, працює під управлінням операційної системи «ROM-DOS» сумісної з «MS-DOS» і підтримує всі її основні функції з можливістю виконання програм, написаних мовою «С».

Головний модуль контролера складається з центрального мікропроцесора, джерела живлення, чотирьох слотів для модулів вводу-виводу, трьох послідовних портів. «Flash-ROM» має обсяг 256 Кбайт, а «Flash-disk» - 512Кбайт, з яких 400 Кбайт призначені для користувальницького ПЗ. Статична оперативна пам'ять «SRAM» складає 60 Кбайт, а її живлення забезпечується від батареї. ADAM 5511 містить у собі 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 батарейки «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», швидкості передачі даних і робочого COM-порту.

ADAM 5511 підтримує всі дискретні і аналогові модулі вводу-виводу серії ADAM 5000.

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

Для програмування ADAM 5511 призначений порт «COM1» контролера, що підтримує роботу в режимі інтерфейсу RS-232. Для використання «COM1» перемикач DIP6 на корпусі контролера необхідно встановити в положення «ON». Для підключення до ПК використовується перехресний нуль - модемний кабель, рис.5.3.

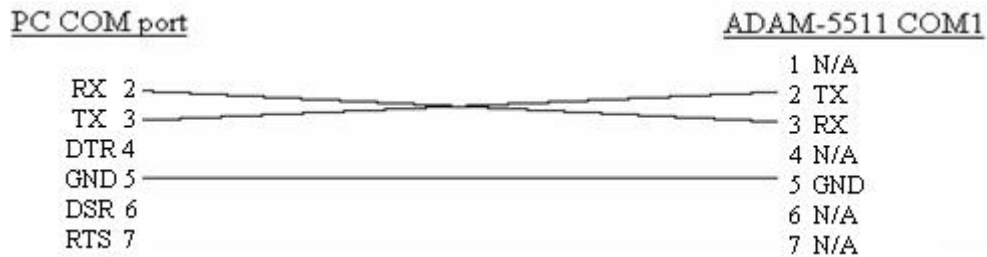


Рисунок 5.3 – Підключення ПК до «COM1» контролера

Порт «COM2» підтримує роботу в режимі інтерфейсу «RS-485». Функції, що виконуються «COM2» аналогічні функціям «COM1». Для підключення контролера за допомогою даного порту до ПК необхідно використовувати перетворювач інтерфейсу «RS-232/485». Клеми «DATA+» і «DATA-» призначені для підключення по інтерфейсу RS-485, рис.5.4.

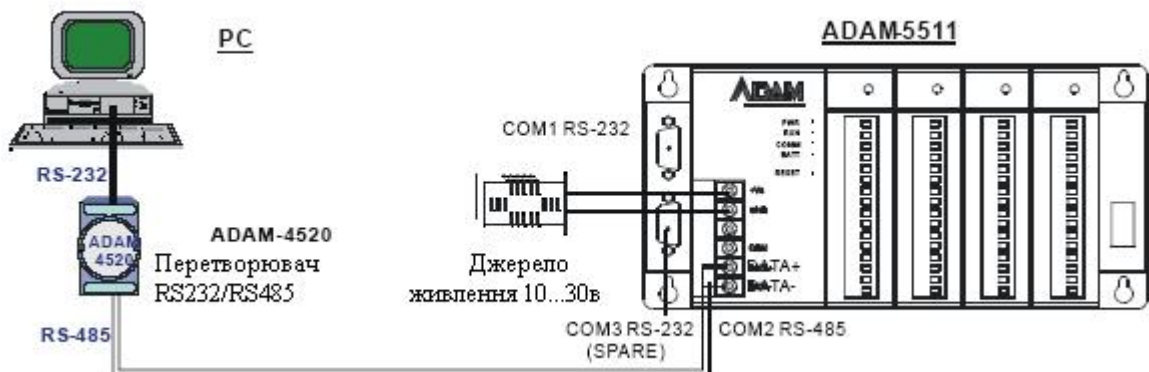


Рисунок 5.4 – Підключення ПК до «COM2» контролера

Для калібрування та конфігурування аналогових модулів контролера призначений порт «COM3», що працює у режимі інтерфейсу «RS-232». Перед використанням порту необхідно всі DIP - перемикачі на корпусі контролера встановити в положення «OFF». Для підключення до ПК використовується «прямий» кабель.

Порти «COM1(RS-232)» і «COM2(RS-485)» призначені для роботи за протоколом «Modbus». Через ці порти здійснюється програмування контролера[25].

### **5.1.2 Програмування контролерів групи ADAM-5510**

Контролери групи ADAM-5510 є PC-сумісними контролерами, тому програмування їх передбачає використання мов високого рівня. Здійснюється це за допомогою середовища розробки Borland Turbo C++ v.3.0.

При створенні прикладного програмного забезпечення використовуються спеціалізовані бібліотеки, що постачаються виробником у комплекті з контролером. Для найбільш ефективного використання пам'яті контролера функції, що створені мовою «C», розділені на п'ять окремих бібліотек. Тому користувач повинен підключати до проекту тільки ті бібліотеки, які містять необхідні для поточного проекту функції.

Серед бібліотечних функцій мікроконтролера ADAM 5510 є повний набір підпрограм, написаних мовою «C» для підтримки виконання різноманітних операцій вводу-виводу. Ці підпрограми використовуються розробниками при створенні додатків у середовищі Borland C++ 3.0.

Створення проекту здійснюється за допомогою EOM і програмного додатку до контролера «ADAM-5510/5511 Windows Utility». Ця утиліта дозволяє реалізувати пошук підключених модулів ПЗО, читання значень каналів підключених модулів ПЗО, роботу з програмним забезпеченням користувача - завантаження, видалення, запуск, зупинник.

Для роботи з зовнішніми вхідними та вихідними змінними використовуються не фізичні канали модулів вводу-виводу, а відповідні їм адреси регістрів у пам'яті контролера.

Після створення коду проекту він компілюється і створюється «\*.exe» - файл, який завантажується у контролер. Найбільш складним при створенні проекту для традиційного спеціаліста з автоматики є розробка коду програми мовою «C», що суттєво обмежує використання контролерів. Ця обставина, а також впровадження стандарту IEC 61131-3 спонукало виробника контролерів забезпечити можливість програмування їх технологічними мовами. Тому починаючи з ADAM 5510 програмна підтримка контролерів доповнена системою російського виробництва UltraLogic, яка забезпечує програмування мовою функціональних блокових діаграм FBD.

#### **5.1.2.1 Методика створення проекту у середовищі Borland Turbo C++ v.3.0**

Створити у середовищі Borland Turbo C++ v.3.0 проект системи автоматичного регулювання температури в печі за ПІД-аналоговим

алгоритмом, який починає працювати після отримання контролером ADAM 5511 дискретного сигналу про закриття кришки печі.

Для реалізації керувального впливу на піч скомпонувати контролер модулями аналогового вводу ADAM 5018, аналогового виводу ADAM 5024 та дискретного вводу ADAM 5050.

Для роботи з контролером по інтерфейсу RS-232, з'єднати порти контролера «COM1» та «COM3» з портами комп'ютера. Порт «COM1» з'єднується «прямим» кабелем, «COM3» нуль-модемним.

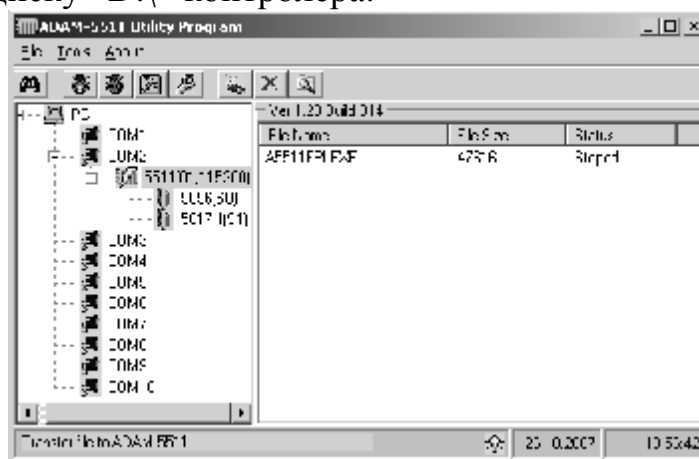
Інструментальне програмне забезпечення, що постачається разом з ADAM 5511, містить утиліту «ADAM5511 Windows Utility», яка дозволяє встановити зв'язок з вбудованим інтерпретатором командного рядка ROM-DOS.

Ця утиліта реалізує:

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

Після встановлення утиліти та її запуску на екрані EOM з'явиться вікно, яке містить панель інструментів і дві зони. В лівій відображається тека PC з доступними портами комп'ютера, а в правій знаходиться вікно для відображення списку користувальницьких програм на диску «D:\» контролера.

Розкривши теку PC, встановити курсор на послідовний порт EOM, до якого підключений «COM1» контролера. В панелі інструментів натиснути кнопку «Search» і в лівій частині вікна утиліти відобразиться ADAM 5511. Клацнувши лівою клавшею миші по назві контролера відкриється список встановлених у слоти модулів, розблокуються кнопки панелі інструментів і активізується вікно диску «D:\» контролера:

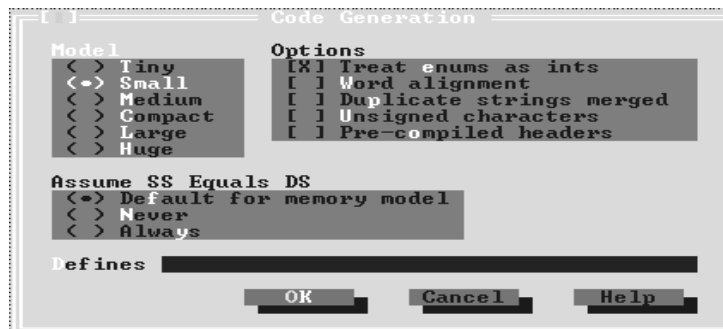


### ***Настроювання середовища розробки «Turbo C++ v.3.0»***

До початку розробки програмного коду необхідно виконати настроювання «Turbo C++ v.3.0».

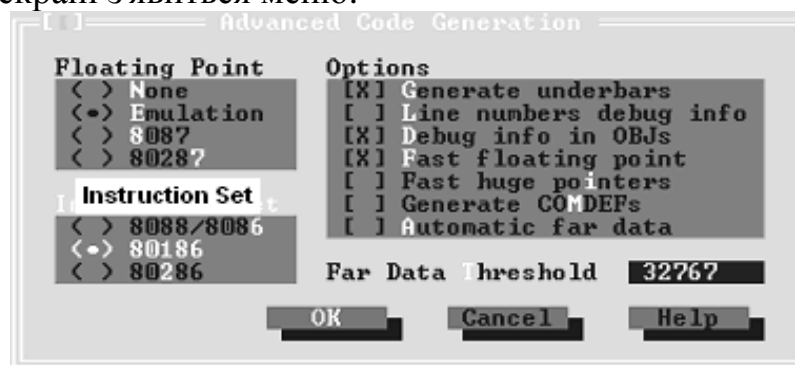
Для цього запустити на виконання файл «tc.exe» з каталогу «C:\TC\BIN», де встановлене середовище розробки, або запустити з ярлика на робочому столі. На екрані з'явиться вікно «Turbo C++ v.3.0».

Для створення виконавчого файлу вибрати шлях «Options ->Compiler->Code generation...» і в меню, що з'явиться, у групі «Model» активізувати «Small», щоб компілятор згенерував файл розміром 64 Кбайт:



Якщо необхідно одержати виконавчий файл за розміром більше 64 кбайт, треба активізувати «Medium» і натиснути кнопку ОК.

Щоб настроїти тип процесора, для якого компілятор генерує код, і забезпечити обробку операцій з плаваючою комою необхідно повернутися в головне меню програми і вибрати шлях «Options»->«Compiler->Advanced code generation...». На екрані з'явиться меню:



У групі «Floating point» установити «Emulation», а у групі «Instruction Set» вибрати процесор «80186» і натиснути кнопку ОК.

Для настроювання каталогів, в яких будуть міститися виконуваний файл і вибрані бібліотеки, необхідно згорнути вікно «Turbo C++ v.3.0» і за допомогою засобів Windows створити папку C:\TC\ім'я каталогу (наприклад, USER).

Перейти в середовище розробки «Turbo C++ v.3.0».

У головному меню програми вибрати шлях «Options»->«Directories...». З'явиться меню для настроювань каталогів:

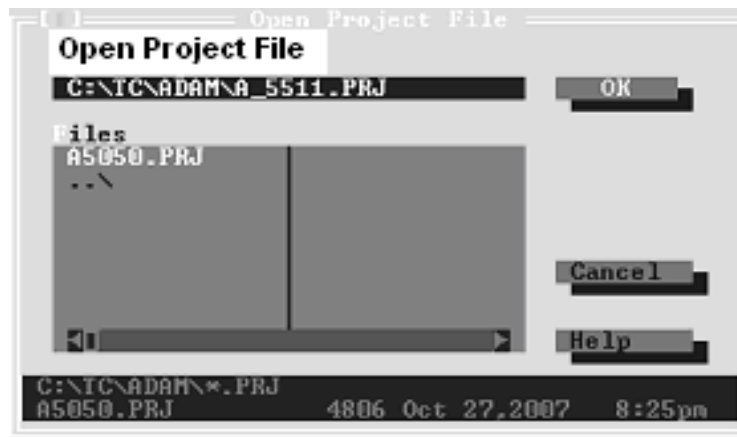


У рядках «Output Directory» і «Source Directories» установити «C:\TC\USER».

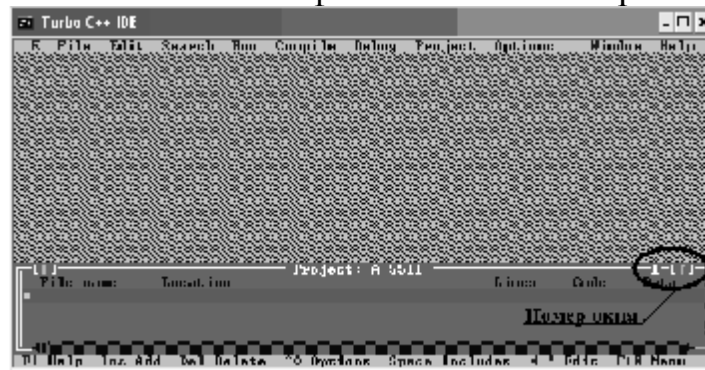
### **Створення проекту в «Turbo C++ v.3.0»**

Створення проекту в «Turbo C++ v.3.0» складається з компонування файлів бібліотеки, розробки коду програми та інших допоміжних файлів.

Для створення проекту в головному меню інтегрованого середовища розробки вибрати пункт «Project» і підменю «Open project...». У діалоговому вікні, що з'явиться:



в рядку «Open Project File» надрукувати C:\TC\ADAM (ім'я каталогу)\A\_5511.PRJ, тобто повний шлях та ім'я створюваного проекту з розширенням «PRJ». На клавіатурі EOM натиснути «Enter». У нижній частині екрану EOM з'явиться вікно зі списком файлів поточного проекту:

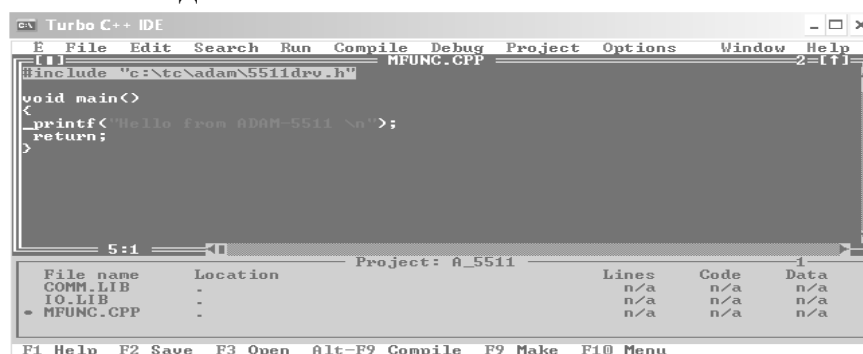


Для додавання файлів (бібліотеки, коду програми), що будуть використовуватися при розробці проекту, у головному меню вибрати меню «Project» і підменю «Add Item...».

При цьому вікно проекту має бути активним. Для активізації відповідного вікна використовується комбінація клавіш «Alt+N», де N номер вікна, якому передається фокус.

За допомогою діалогового вікна, що при цьому з'явиться, додати в поточний проект файли бібліотек: «Utility.lib», «IO.lib». Для цього в рядку «Name» указати «\*.lib». Перейти у верхню директорію, натиснувши ..\ . Перейти в каталог ADAM. Для вибору потрібної бібліотеки двічі натиснути лівою клавішею миші на її імені. Для завершення натиснути Done.

Для створення і додавання файлу, який містить виконуваний код, необхідно вибрати в головному меню шлях «File»->«New». Задати ім'я програми і зберегти новий файл з розширенням «\*.cpp» за допомогою пункту «Save As...» меню «File». Додати створений файл у проект (використовуючи вже зазначений шлях «Project» -> «Add Item...») і відкрити його для редагування. При додаванні файлу в проект, указати «\*.cpp». Вміст проекту має прийняти наступний вигляд:



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

Адресація регістрів для аналогових і дискретних модулів встановлених в ADAM 5511, залежить від номера слота і показана на рис.5.5.



Рисунок 5.5 – Адресація регістрів модулів ПЗО серії «ADAM 5000»

Для створення проектного коду потребують такі функції:

- з бібліотеки «Utility.lib»:
  - перевірка наявності ознаки зупинки програми, яка отримується з утиліти «ADAM-5511 Windows Utility» при натисненні кнопки «stop» у панелі інструментів.

Синтаксис: void check\_prog\_stop(void);

- пауза при виконанні програми.

Синтаксис: void ADAMDelay(unsigned int TC);

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

- читання користувальницьких регістрів «ModBUS».

Синтаксис: unsigned int read\_user\_ram(unsigned int index);

Параметр «index» - індекс пам'яті для доступу до двобайтових регістрів «ModBUS» у діапазоні [0 - 511]. Повертає значення регістра за заданим індексом пам'яті.

- з бібліотеки «IO.lib»:

- читання значення аналогових входів модуля ПЗО ADAM 5018.

Синтаксис: void Get5017/5018(int Board, int Bit, int Size, void \*pValue);

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

«Size» - розмір параметра, зчитуваного або встановлюваного функцією.

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

- устанавлення вихідних значень аналогового модуля виводу ADAM 5024.

Синтаксис: void Set5024(void \*pValue, int Board, int Channel);

«pValue» - устанавлюване вихідне значення модуля;

«Board» - слот [0 - 3], у який встановлене ПЗО;

«Channel» - номер каналу [0 - 3].

Додамо у проект файли бібліотек «IO.lib», «Utility.lib», «A5050.lib», а також файл з проектним кодом мовою C++.

Рівняння ПД-алгоритму регулювання має вигляд:

$$Y = Kp(x[n] + \frac{T_0}{T_u} \sum_{i=1}^n \frac{x[n] + x[n-1]}{2} + \frac{T_d}{T_0} (x[n] - x[n-1]))$$

Структура програми складається з наступних блоків:

```
//-----  
// Підключення бібліотек і файлів  
//-----  
#include "c:\tc\adam\5511drv.h"  
#include <stdio.h>  
// основні параметри  
//-----  
// Ініціалізація каналів для роботи регулятора, змінних і функцій, що  
//використовуються у програмі.  
//-----  
float RealVal = 60; // реальне значення регульованого параметра  
float Task = 50;    // завдання  
float Y = 0;        // вихід регулятора  
float dUprev = 0;   // значення величини неузгодженості на попередньому кроці  
виклику  
float IntTot = 0;   // сумарна накопичена величина для інтегральної складової  
// параметри регулятора
```

```

float IZ=1;          // зона нечутливості
float YHi = 100;    // верхнє обмеження виходу
float YLo = 0;      // нижнє обмеження виходу
// коефіцієнти ПІД-регулятора
float T0 = 1;       // час такту в секундах, тобто період виклику регулятора
float Kp = 1;       // коефіцієнт пропорційності
float Ti = 0.5;     // час інтегрування
float Td = 0.2;     // час диференціювання
//----- прототипи функцій ПІД-алгоритму-----
float Integr(float);
float Differ(float d);
float PID();
void PIDsetup();
//-----ініціалізація каналів-----
int Slot5018 = 0;    // номер слота для ADAM 5018
int Channel5018 = 0; // номер каналу зі значенням від термопар, термопара 1 (V0)
int Slot5050 = 2;    // номер слота для ADAM 5050
int Slot5024 = 1;    // номер слота для ADAM 5024
int Channel5024 = 0; // номер каналу з керуючим значенням
unsigned int Value5050; // значення дискретних входів/виходів ADAM 5050
unsigned int IsOpen = 1; // готовність об'єкта до роботи
unsigned int Value5018; // значення від термопар
int Value5024;       // вихідний вплив на блок управління тиристорний
int Restart;         // ознака рестарту програми
//-----
// Читання вхідних сигналів
//-----
// ГОЛОВНА ФУНКЦІЯ (точка входу в програму)
void main() // функція, що виконується при запуску програми
{
Init5024(Slot5024,0,0,0,0); // ініціалізація ADAM 5024
printf("Початок роботи\n");
while(1) // нескінченний цикл
{
Get5050(Slot5050,0,AWord,&Value5050); // читання дискретних входів ADAM 5050
if(!(Value5050 & IsOpen)) //перевірка стану кінцевого вимикача
{
Get501718(Slot5018, Channel5018, &Value5018); // читання значення від термопар
RealVal = (float)Value5018; // перетворення прийнятого значення для
// передачі у функцію керування
printf("Температура = %f\n",RealVal);
PID(); // виклик функції керування за ПІД - законом
Value5024 = (int)Y; // перетворення розрахункового значення в цілочислове
// для видачі вихідного впливу
Set5024(Value5024, Slot5024, Channel5024); // видача вихідного впливу
// через ADAM 5024
printf("Вихідний вплив = %d\n", Value5024);
PIDsetup(); // читання уставок регулятора
if(Restart) // перевірка ознаки "рестарт регулятора"
{
dUprev = 0; // обнуління значення величини неузгодженості на попередньому кроці
}
}
}

```



```

IntTot = 0; // обнуління суми накопиченої величини для інтегральної складової
Restart = 0; // обнуління ознаки "рестарт регулятора"
Set5024(0, Slot5024, Channel5024); // обнуління вихідного впливу
printf("Отримана команда РЕСТАРТ\n");
}
}
else
{
Set5024(0, Slot5024, Channel5024); // обнуління вихідного впливу

printf("Дверцята печі відкриті\n");
Restart = 1;
}
ADAMDelay((T0*1000.)/55.); // задана пауза в роботі регулятора
check_prog_stop(); // перевірка ознаки "зупинка програми"
}
}
//-----
// Читання настроювань алгоритму керування з користувальницьких регістрів
//-----
void PIDsetup()
{
int Kpid=0,C3=0;
// ТОЧНІСТЬ ЗНАЧЕНЬ ДО 0.1, тобто
// ПРИ ДЕШИФРУВАННІ ЗНАЧЕНЬ ВИХІДНА ВЕЛИЧИНА
// МНОЖИТЬСЯ НА 0.1
// аналіз ознаки "читати установки ПІД регулятора"
Kpid = read_user_ram(0);
if(Kpid==0x00A5) // десяткове значення 165
{
Kpid = 0;
// читання завдання
Task= (float)read_user_ram(1) * 0.1;
// читання значення зони нечутливості
IZ = (float)read_user_ram(2) * 0.1;
// верхнє обмеження виходу
YHi = (float)read_user_ram(3) * 0.1;
// нижнє обмеження виходу
YLo = (float)read_user_ram(4) * 0.1;
// період виклику регулятора
T0 = (float)read_user_ram(5) * 0.1;
// пропорційний коефіцієнт
Kp = (float)read_user_ram(6) * 0.1;
// коефіцієнт інтегрування
Ti = (float)read_user_ram(7) * 0.1;
// коефіцієнт диференціювання
Td = (float)read_user_ram(8) * 0.1;
// ознака перезапуску
Restart = read_user_ram(9);
}
// ознака виводу значень ПІД- регулятора в СОМ3

```

```

C3 = read_user_ram(10);
if(C3)
{
C3=0;
printf("Task =%f IZ = %f\n",Task,IZ);
printf("YHi =%f YLo = %f\n",YHi,YLo);
printf("T0 = %f Kp = %f Ti = %f Td = %f\n",T0,Kp,Ti,Td);
}
}
//-----
// Розрахунок складових ПІД-алгоритму керування
//-----
// розрахунок інтегральної складової
float Integr(float d)
{
float S;
if ((T0 > 0) && (Ti != 0)) S = (d + dUprev) / 2.0 * T0 / Ti;
else return 0;
return S;
}
//-----
// розрахунок диференціальної складової
float Differ(float d)
{
float D;
if ((T0 > 0) && (Td != 0)) D = (d - dUprev) * Td / T0;
else return 0;
return D;
}
//-----
// Реалізація функцій ПІД регулятора
//-----
// основна функція ПІД- регулятора
float PID()
{
float d, OutVal;
d = Task - RealVal; // неузгодженість регулювання
// якщо неузгодженість не виходить за зону нечутливості,
// немає необхідності рахувати далі
if (2*fabs(d) <= IZ) return 0.;
IntTot += Integr(d);
// перевірка вихідного значення
Y = Kp * (d + IntTot + Differ(d));
// обмеження вихідного значення
if (Y > YHi) Y = YHi;
if (Y < YLo) Y = YLo;
// збереження величини неузгодженості на попередньому кроці
dUprev = d;
return Y;
}

```

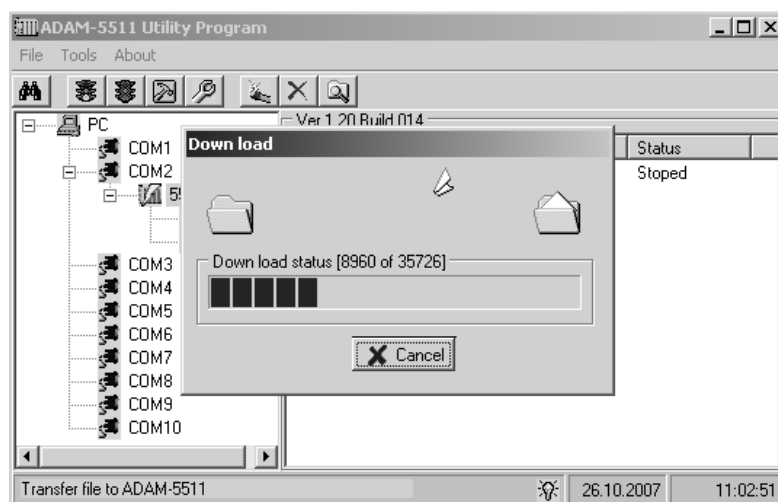
Для компіляції проекту та одержання «\*.exe» - файлу, необхідно вибрати в головному меню Turbo C++, пункт «Compile», підміню «Build». Про успішне закінчення компіляції буде свідчити повідомлення:



При появі повідомлень про помилки, треба усунути їх і повторити процес компіляції.

Кнопкою «Stop» в панелі інструментів «ADAM 5511 Windows Utility» зупинити виконувану в контролері програму, якщо така є.

Для завантаження програми необхідно в панелі інструментів «ADAM 5511 Windows Utility» натиснути кнопку «Transfer file to ADAM-5511». У діалоговому вікні, що відкриється, вибрати виконавчий файл розробленої програми і натиснути кнопку «ОК». Після цього почнеться процес передачі файлу в контролер:



Після закінчення процесу передачі - вибрати завантажений файл програми в правій частині вікна програми і в панелі інструментів «ADAM 5511 Windows Utility» натиснути кнопку «Run». Контролер почне працювати. Після запуску статус програми повинен змінитися на «Runing». Для зупинення програми на панелі інструментів «ADAM 5511 Windows Utility» натиснути кнопку «Stop» або «Terminate» .

Для видалення програми з пам'яті контролера необхідно зупинити виконання програми за допомогою кнопок «Stop» або «Reset» в панелі інструментів «ADAM 5511 Windows Utility». У правій частині вікна утиліти, де відображається список файлів на диску «D:\» контролера, вибрати необхідний

файл і натиснути кнопку «Delete» в панелі інструментів «ADAM 5511 Windows Utility»[25,26].

## 5.2 Мікропроцесорний контролер ADAM 5510KW

ADAM 5510KW – це базовий контролер групи контролерів фірми Advantech, які відповідають усім вимогам міжнародного стандарту IEC-61131-3 і можуть програмуватися п'ятьма технологічними мовами. Контролер має збільшений відносно ADAM 5510 розмір пам'яті для програмування та збереження проектів, багатозадачне ядро реального часу, вільно визначає бібліотеку, потужні налагоджувальні можливості, відкриту стандартну комунікаційну систему – Modbus, редагування у реальному масштабі часу, RS – 232/485 – комунікаційні можливості, вбудовані годинники реального часу та засоби безпеки, зручний інструментальний засіб конфігурування. Габаритні розміри базового блока складають 110×231× 75мм (В×Ш×Г).

Мікроконтролер ADAM 5510KW, як і всі контролери серії ADAM 5000 складається з двох основних частин - базового блока та модулів вводу – виводу, які є загальними для всіх контролерів серії ADAM 5000, рис.5.6.

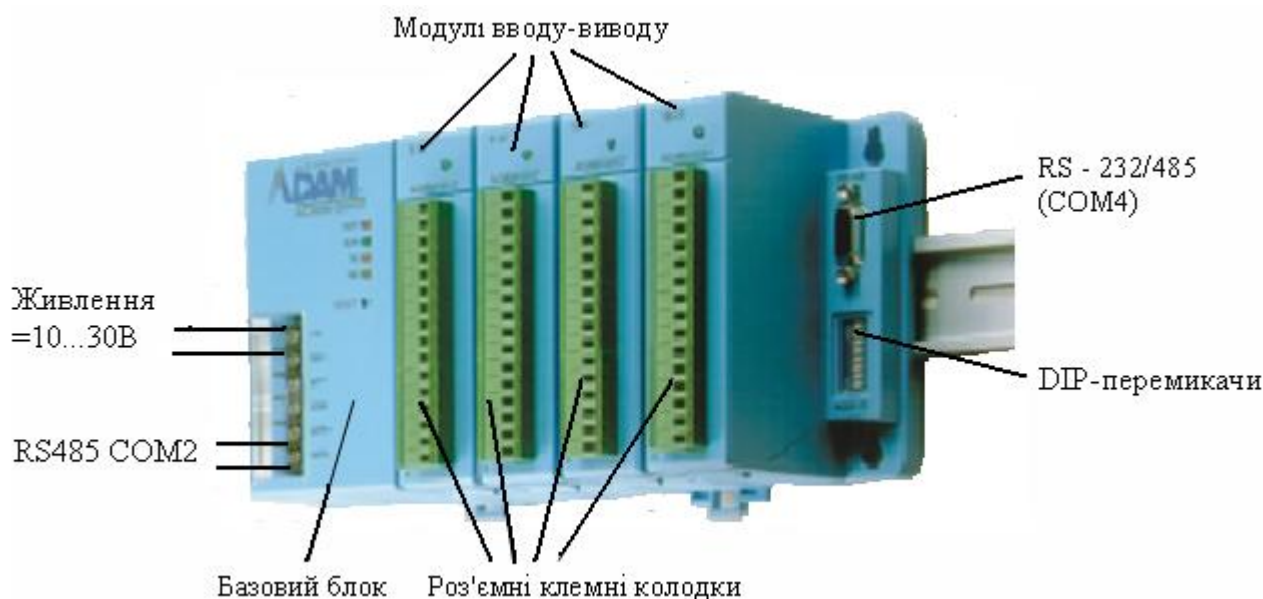


Рисунок 5.6 – Мікропроцесорний контролер ADAM 5510KW

Базовий блок має у своєму складі процесорний модуль, перетворювач напруги, 4-слотову пасивну об'єднувальну панель та серійні комунікаційні порти RS-232, RS-485.

Тип центрального процесора CPU - 80188-40, 16-розрядний. ADAM 5510KW має вбудовану Flash-пам'ять на 1,5MB і статичну оперативну пам'ять (SRAM) на 640KB для завантаження операційної системи та збереження даних.

Це більше ніж у традиційних PC-сумісних контролерів, тому дає змогу збільшувати розмір програми.

Операційна система – ROM-DOS.

ADAM 5510KW, в якості системи виконання реального часу, має програму KW PROCONOS, яка забезпечує виконання одночасно до 16 задач.

ADAM 5510KW має чотири серійні комунікаційні порти, три з яких розташовані так само, як і в ADAM 5510, а четвертий знаходиться з права над DIP-перемикачами.

Ці порти дозволяють програмувати та конфігурувати контролер, контролювати і об'єднувати віддалені пристрої вводу – виводу. Порт COM1 підтримує роботу в режимі інтерфейсу RS-232. До нього приєднується головний комп'ютер за допомогою з'єднувача DB-9.

Порт COM2 підтримує роботу в режимі інтерфейсу RS-485, приєднується до головного комп'ютера за допомогою проводів і клемника через RS-232/RS-485-перетворювач. Завдяки протоколу Modbus контролер можна інтегрувати в АСКТП і об'єднати з більшістю існуючих SCADA-систем.

Порт COM3 – порт налагодження, використовується тільки для системного обслуговування, його інтерфейс - RS-232, приєднувач DB-9. Цей порт дозволяє програмувати, конфігурувати та шукати несправності у контролері з головного комп'ютера.

Порт COM4 підтримує RS-232/485 і призначений для роботи з віддаленими модулями вводу-виводу серії ADAM-4000 (до 255 точок), рис.5.7.

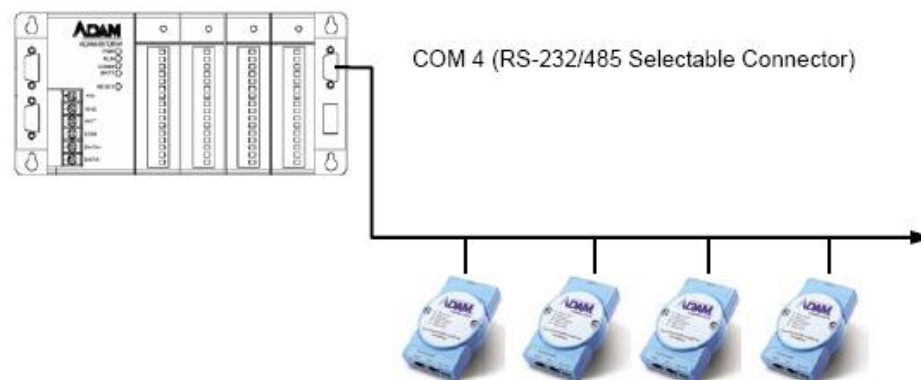


Рисунок 5.7 – Приєднання контролера до віддалених модулів вводу-виводу

Потрібний інтерфейс встановлюється за допомогою перемикача підключеного до цього порту. Порти ADAM 5510KW задовольняють всім вимогам комунікації та інтеграції і допомагають розширити системну структуру.

Інтерфейс RS-232 забезпечує швидкість передачі до 38400біт/с на відстані до 15.2 м. Режим роботи – асинхронний повний дуплекс, з'єднання точка - точка. Інтерфейс RS-485 має швидкість передачі до 38400 біт/с на відстані до 1220м. Режим – напівдуплексний (багатопрхідний).

За допомогою DIP – перемикачів і відповідно до табл.5.1-5.2 встановлюється ідентифікатор пристрою (перемикачі 1-5), протоколи портів COM1, COM2 (перемикач 6) та швидкість обміну даними (перемикачам 7 і 8).

Таблиця 5.1 Установлення швидкості обміну даними у бодах

| DIP 7 | DIP 8 | Baud Rate |
|-------|-------|-----------|
| ON    | ON    | 9600 bps  |
| OFF   | OFF   | 9600 bps  |
| ON    | OFF   | 19200 bps |
| OFF   | ON    | 38400 bps |

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

Таблиця 5.2 Установлення протоколів

| DIP 6 | COM 1             | COM 2             |
|-------|-------------------|-------------------|
| OFF   | MULTIPRO Protocol | MODBUS Protocol   |
| ON    | MODBUS Protocol   | MULTIPRO Protocol |

Канали модулів не залежно від місця їх розташування починають нумеруватися від 0.

ADAM 5510KW, як і ADAM 5510 має годинники реального часу та сторожовий таймер. Додатково у нього є три сторожові схеми – сторож операційної системи, сторож KW-додатку та сторож комунікаційного обслуговуючого пристрою Modbus. Це збільшує надійність пристрою і дає йому переваги при розв’язанні складних задач автоматизації.

ADAM 5510KW можна використовувати в якості дублюючого пристрою. Для цього необхідно іншій резервний контролер підключити паралельно основному і завантажити в нього таку саму програму, а також присвоїти йому відповідний ідентифікатор. Тоді у випадку помилки в комунікації або передачі даних в основному контролері, активізується резервний і продовжить функціонування системи управління.

ADAM 5510KW розроблений під напругу 24В нестабілізованого постійного струму, але підключати його можна до будь-якого джерела з напругою в межах +10 - +30В.

На передній панелі контролер має чотири індикатори:

PWR - індикатор напруги, коли контролер несправний індикатор блимає.

RUN – індикатор виконання програми, блимає безперервно, коли програма виконується.

COMM – індикатор комунікації, блимає кожного разу, коли головний комп’ютер обмінюється з контролером.

BAT – індикатор стану акумулятора, горить коли акумулятор резервування SRAM має низьку напругу.

ADAM 5510KW забезпечує багатозадачний та багатопрограмний режим роботи. Його комунікація, обробка даних і доступ до модулів вводу-виводу здійснюється незалежно одно від одного, тому якість функціонування та ефективність цього контролера краще ніж у традиційних PLC.

Монтаж контролерів здійснюється на вертикальну стінку за допомогою гвинтів або на DIN-рейку. Модулі вводу-виводу встановлюються у різні

вільних слотів базового блоку легким натисненням. Приєднання проводів до клемників модулів – гвинтове. Клемники на модулях мають фронтальне розташування. Для зручності монтажу вони приєднуються за допомогою рознімів[27].

### **5.2.1. Програмування контролерів групи ADAM 5510KW**

Контролери групи ADAM 5510KW програмуються технологічними мовами стандарту IEC- 61131-3. Це сталося завдяки купівлі фірмою Advantech прав на використання програмного продукту KW MULTIPROG у німецького розробника KW-Software GmbH.

KW MULTIPROG – це вбудована основа Softlogic – контролера, яка використовується при автоматизації об'єктів з 1991 року. Вона підтримує усі стандартні мови програмування, причому в одному проекті можуть використовуватися усі мови.

Завдяки системі програмування MULTIPROG, контролер має напередвизначену бібліотеку великої кількості функцій, за допомогою якої легко створити у проекті таймер, перетворити тип змінної або перетворити послідовність строкової змінної і т. ін. Крім того, користувач може створити власний функціональний блок, використовуючи функції бібліотеки.

Графічний редактор дозволяє вільне розміщення об'єктів на робочому полі проекту, вибирати мову програмування та використовувати бібліотеку функцій. Він підтримує змішування графічних мов в одному робочому листі, розміщення об'єктів в розмірній сітці, вставлення нових елементів в існуючі мережі, переміщення об'єктів або мереж, показ графіків функцій та функціональних блоків для уявлення їх поведінки, різні кольори бібліотечних функцій та функціональних блоків.

Текстовий редактор дозволяє легко і швидко створювати програми. Для цього існують різні функціональні можливості:

- підсвічування синтаксису вказує на ключові слова програми;
- майстер редагування полегшує редагування, вказуючи на доступні елементи і блоки.

MULTIPROG має могутній засіб налагодження, діагностування, моделювання та функціонального втручання із зручним інтерфейсом користувача. За допомогою MULTIPROG перевірка програми контролера здійснюється на емуляторі без участі самого контролера.

MULTIPROG підтримує оперативне редагування програми, завдяки чому можна вносити зміни і завантажувати їх в контролер не зупиняючи процесор.

MULTIPROG запускається з головного комп'ютера. Коли програму створено вона завантажується в контролер через порт COM1 і контролер починає працювати.

MULTIPROG працює разом з програмованою операційною системою реального часу ProConOs (Programmable Controller Operating System), яка вбудована у контролер[27].

### 5.2.1.1 Методика створення проекту у системі програмування KW MULTIPROG

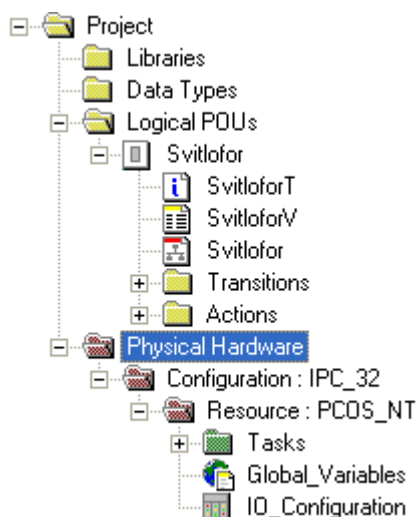
Створити SFC-мовою проект керування роботою світлофора, який встановлений на перехресті доріг. Світлофор має перебувати в чотирьох станах: зеленому, червоному, жовтому і жовто-червоному. Тривалість зеленого і червоного станів 10с, а жовтого і жовто-червоного – 3с. Світлофор вмикається в роботу дворазовим натисненням кнопки «Пуск». Цикли роботи світлофора підраховує лічильник. Після 10 циклів роботи світлофор вимикається а лічильник скидається. На першому етапі загоряється зелене світло правого світлофора і червоне лівого, на другому етапі загоряється жовте на правому світлофорі і жовто-червоне на лівому, на третьому загоряється на правому світлофорі червоне і зелене на лівому, на четвертому етапі загоряється жовто-червоне на правому світлофорі і жовте на лівому.

#### *Створення проекту*

Створення проекту здійснюється після запуску програми MULTIPROG V3.3. Використовуючи в меню шлях File→ New Project, двічі клацнути по піктограмі Project Wizard у вікні New Project.

За замовчуванням шлях створення проекту вводиться автоматично. Для вибору іншого шляху у вікні Project Wizard натиснути кнопку з трьома крапками і у діалозі Select Directory вибрати теку для нового проекту. Ввести ім'я першого організатора програми (POU), наприклад Svitlofor, вибрати SFC-мову. Ввести ім'я конфігурації, наприклад Configuration, і вибрати тип конфігурації проекту IPC\_32, тому що компілятор генерує intel- код для системи виконання ProConOS 3,2. Ввести ім'я ресурсу – Resource і вибрати його тип - PCOS\_NT. Ввести ім'я задачі –Task, і вибрати її тип – Cyclic.

Після натиснення кнопки Next на екрані з'явиться зміст зроблених установок. Якщо помилок немає натиснути Finish, в іншому разі, користуючись кнопкою Back, скоригувати кроки проекту де є помилки. Створений проект Svitlofor з'явиться у вікні проектного дерева у піддеревах Logical POU's і Physical Hardware. В меню дерева проекту розкрити теку Physical Hardware:





Для модифікації імені конфігурації правою клавішею миші клацнути по Configuration:IPC\_32 і у меню, що з'явиться вибрати Properties. Надрукувати ім'я конфігурації, наприклад ADAM 5510KW\_Project, у вікні, що відкриється, і натиснути ОК.

Після цього ім'я конфігурації в дереві проекту зміниться на ADAM 5510KW\_Project: IPC\_32, при цьому IPC\_32 додається автоматично.

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

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\_NT. У підменю, що відкриється, активізувати "Settings...". На екрані з'явиться діалогове вікно: PCOS\_NT, в якому:

COM- вказує у який серійний порт відправлено програму;

Simulation- вказує в який симулятор відправлено програму;

Baud- конкретизує швидкість компіляції у Бодах через порт;

Stopbits- конкретизує кількість стопових біт при комунікації через 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 встановлення порту контролера необхідно у каталозі «Port:» активізувати альтернативу COM1.

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

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

| Non retain               |       |
|--------------------------|-------|
| Start user:              | 0     |
| End user / Start system: | 0     |
| End system (max 63999):  | 63999 |
| Reserve per POU:         | 500   |

|                          |       |
|--------------------------|-------|
| Start user:              | 64000 |
| End user / Start system: | 64000 |
| End system (max 65530):  | 65530 |
| Reserve per POU:         | 10%   |

Для редагування робочого листка конфігурації модулів вводу-виводу (В/В) використовується діалог “ I/O\_Configuration”, який звичайно містить початок і кінець адреси модулів вводу-виводу, декларацію пристрою (ім'я драйвера та адреса пам'яті) і т.ін.

Щоб відкрити діалогове вікно, у проектному дереві двічі клацніть по “I/O\_Configuration”.

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

Припустимо, що комплектація контролера складається з модуля аналогового вводу ADAM 5018 у слоті 0 і модуля дискретного вводу-виводу ADAM-5050 у слоті 1. Тоді у діалоговому вікні “ I/O\_Configuration” необхідно вибрати вкладку “INPUT” (Вхід) і натиснути кнопку “Add”(Додати);

На екрані з'явиться діалогове вікно Add I/O Group. У полі Name 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, табл.5.3.

Кінцеву адресу система розраховує автоматично.

Таблиця 5.3 Місткість каналів модулів вводу-виводу

| 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     |

Оскільки у аналогового модуля вводу ADAM-5018 місткість каналів складає 16 бітів, 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.

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, щоб забезпечити можливість установлення модулів серії ADAM-5000.

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

Місце розташування модуля ADAM-5018 вказує на те, що адреса слота встановлюється як 0. У полі Module name вибрати ADAM-5018 [7T/C]. Тут 7-кількість каналів, “T/C”- скорочено визначає аналоговий ввід сигналів термопар.

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

Щоб установити серійний порт натиснути кнопку “ADAM5510KW system setup” і вибрати для модуля B/B Data Format : NOPARITY,8.1.

Натиснути ОК і конфігурація модуля B/B буде показана у відповідному вікні.

Щоб додати модуль дискретного вводу-виводу треба вибрати вкладку OUTPUT у діалоговому вікні “Configuration I/O” і натиснути кнопку “Add...”. З’явиться діалогове вікно Add I/O Group. Модифікувати в ньому дискретний модуль ADAM-5050 іменем: DIO\_5050, а в полі Task: вибрати Task і визначити логічну адресу (Logical addressed), як - Start address: 0, Length:2, а панель модуля B/B (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): 1.


Оскільки ADAM5050 розташований у слоті 1, то у полі Slot: записана 1. Ім’я модуля B/B показується у полі Module name: ADAM 5050 [16UDIO]. Тут 16 – кількість каналів, DIO – дискретний ввід-вивід.

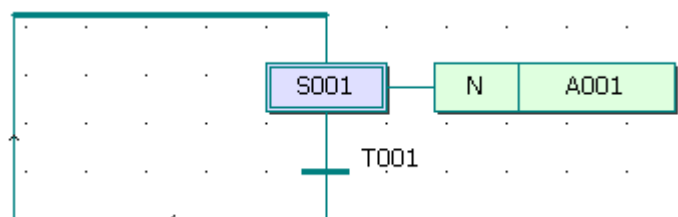
Щоб установити серійний порт натиснути кнопку “ADAM5510KW system setup” і в Data Format вибрати для модуля B/B - NOPARITY,8.1.

Натиснути кнопку ОК і ADAM-5050 потрапить до списку модулів OUTPUT.

Після визначення модулів контролера здійснюється розробка програмного коду.


### ***Розробка коду програми***

В проектному дереві відкрити теку Logical POU’s, вибрати курсором вкладку програмного коду Svitlofor і двічі натиснути ліву клавішу миші. Відкриється робоче поле для створення SFC-коду, на якому позначити курсором місце початку програми. В панелі інструментів натиснути ліву клавішу миші на кнопці  (Creat step transition sequence) і на робочому полі з’явиться SFC-схема, яка містить крок S001 з дією A001 і перехід T001, а також повернення назад:




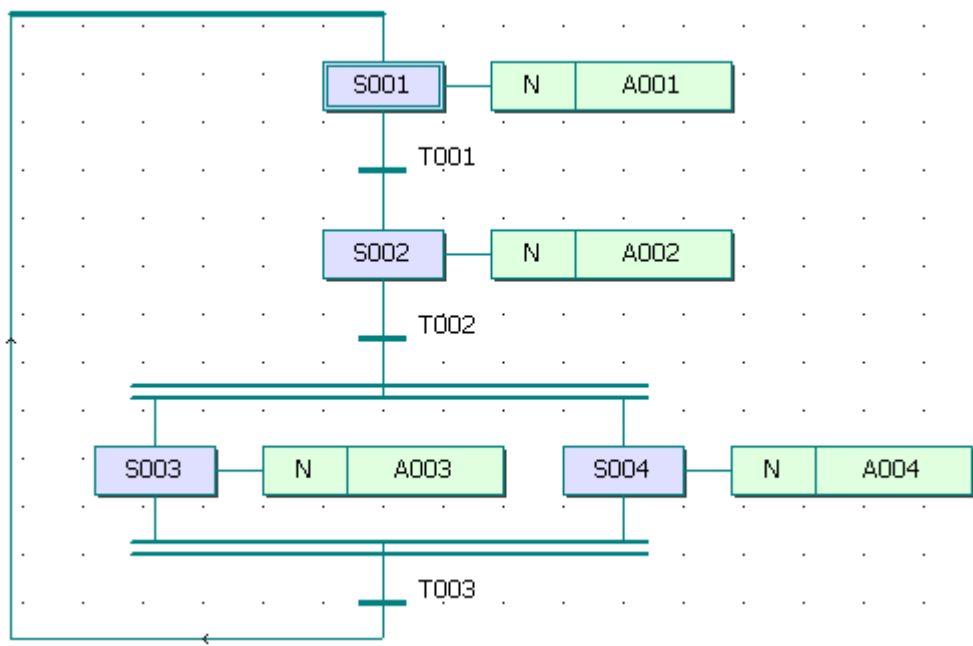
Якщо з’явився не початковий крок, а звичайний, тобто прямокутник кроку створений не подвійною лінією, а одинарною, то двічі натиснути ліву


клавішу миші на прямокутнику і у діалоговому вікні Step, що з'явиться, активізувати Initial step. Закрити вікно Step, натиснути ОК. На робочому полі звичайний крок перетвориться у початковий.


Щоб додати другу пару крок - перехід, призначену для програмування вихідної фази роботи світлофорів (в одному напрямі руху – зелене світло, а в іншому – червоне), натиснути ліву клавішу миші спочатку на переході, а потім в панелі інструментів на кнопці  (Creat step transition sequence). Існуюча SFC-схема збільшиться на один крок S002 і перехід T002.

Оскільки світлофор забезпечує регулювання руху на перехресті, послідовна мережа кроків і переходів повинна мати розгалуження на дві паралельні гілки, в яких будуть програмуватися фази роботи двох світлофорів.

Для створення паралельних гілок розгалуження позначити курсором другий перехід T002 SFC-схеми і в панелі інструментів клацнути по кнопці  (Insert Simultaneous/Alternative Divergence). У діалоговому вікні Divergence, що з'явиться, у полі Branches Count надрукувати кількість гілок - 2 і натиснути ОК. Програмний код у робочому полі набуває вигляд:



У кожній гілці SFC-схеми послідовно додати по дві пари кроків і переходів для програмування решти фаз роботи світлофорів. Здійснюється це за допомогою кнопки  (Creat step transition sequence) в панелі інструментів після позначення лівою клавішею миші відповідного кроку.

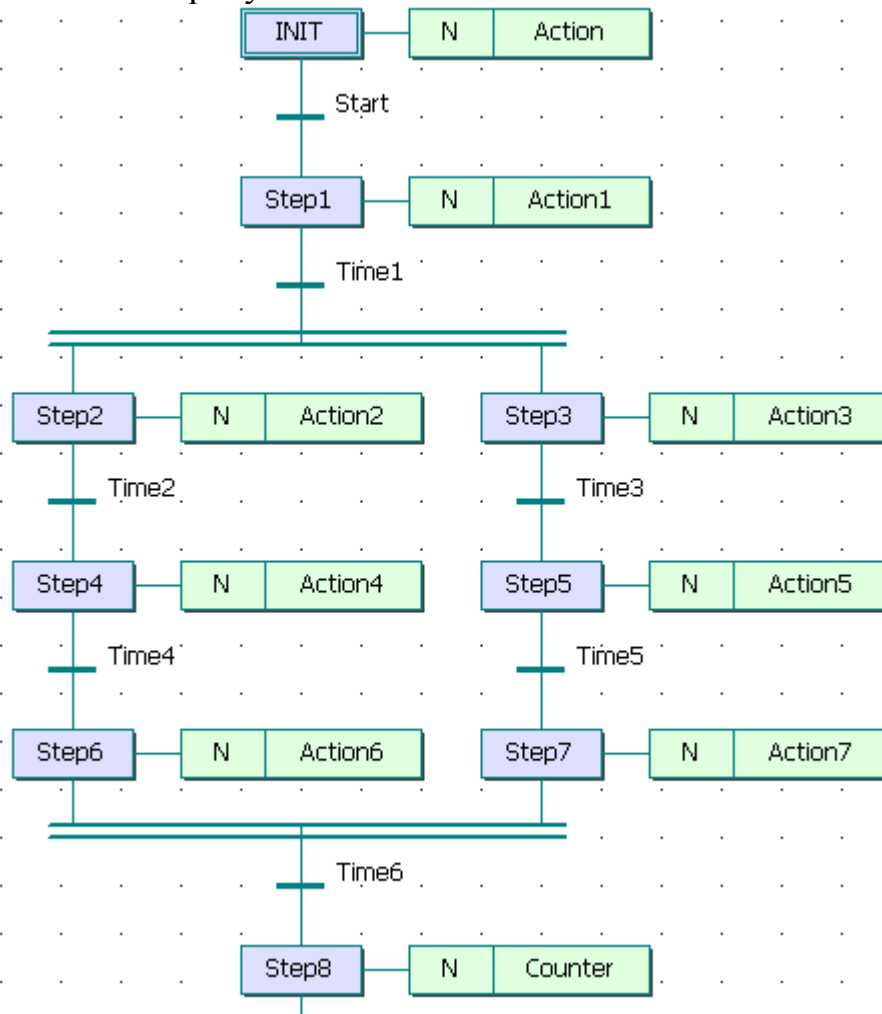
Для створення лічильника циклів роботи світлофорів позначити лівою клавішею миші перехід під подвійним сходженням паралельних гілок і додати нову пару крок - перехід, використовуючи кнопку  (Creat step transition sequence) в панелі інструментів.


Відповідно до умов роботи, світлофор повинен зробити 10 циклів і після цього вийти з циклу і вимкнутися, тобто в програмі має бути

передбачено альтернативне розгалуження зі стрибками до різних кроків. Тому компонентам SFC-схеми потрібно призначити імена.

Клацнути правою клавшею миші по кроку S001 і в меню, що з'явиться, лівою клавшею миші активізувати Object Properties... У вікні Step надрукувати ім'я INIT і натиснути ОК. Таке саме зробити послідовно з дією A001 та переходом T001, привласнюючи їм у діалоговому вікні Action Properties імена, відповідно Action з класифікатором (Qualifier) N та Start у діалоговому вікні Transition, при цьому в обох вікнах альтернатива Detail має бути активізована.

За аналогічною методикою призначити імена решти крокам і переходам, а також діям біля кожного кроку:



Для створення альтернативних гілок розгалуження необхідно після кроку Step8 SFC-схеми створити ще одну пару крок – перехід і позначити лівою клавшею миші останній крок. В панелі інструментів клацнути по кнопці  (Insert Simultaneous/Alternative Divergence) і у діалоговому вікні Divergence, що з'явиться, у полі Branches Count надрукувати 2. Натиснути ОК і у гілках альтернативного розгалуження, що з'являться, створити стрибки на початковий крок INIT і крок Step1. Для цього треба позначити курсором перехід в лівій гілці розгалуження і додати пару крок – перехід. Правою клавшею миші клацнути по кроку і в меню, що з'явиться, лівою клавшею

миші вибрати Object Properties... У полі Type активізувати альтернативу Jump і клацнути ОК. У діалоговому вікні Jump/End step insert control, що з'явиться, натиснути кнопку Change і в лівій гілці залишаться тільки перехід і безадресний стрибок у вигляді прямокутника зі стрілкою.


Таким самим чином зробити безадресний стрибок і у правій гілці альтернативного розгалуження.

Щоб призначити адресу стрибку лівою клавішею миші подвійно клацнути по ньому і у діалоговому вікні Step, що з'явиться, вказати ім'я адреси. Для лівого стрибка у якості адреси позначити початковий крок INIT, а для правого стрибка – крок Step1 у натиснути ОК.

Після створення SFC-схеми необхідно запрограмувати дії кроків і умови переходів відповідно до алгоритму роботи світлофора.


Відповідно до синтаксису SFC – мови, перший крок є ініціалізаційним. Тому змінним світлофора red1, yellow1, green1, red2, yellow2, green2, необхідно присвоїти значення FALSE. Для цього лівою клавішею миші двічі клацнути по дії початкового кроку Action, у діалоговому вікні Insert, що з'явиться, вибрати мову програмування, наприклад ST, і натиснути ОК. Відкриється робоче поле дії Action, у якому всім змінним присвоїти значення FALSE:

```
red1:=False;  
yellow1:=False;  
green1:=False;  
red2:=False;  
yellow2:=False;  
green2:=False;
```

Щоб забезпечити обнуління лічильника після 10 циклів роботи світлофора перейти на новий рядок і в панелі інструментів активізувати кнопку  (Edit Wizard). У таблиці Group вікна Edit Wizard, що з'явиться на екрані, вибрати меню Keywords. Подвійно клацнути по оператору IF і на робочому полі проекту з'явиться його синтаксис з відповідним коментарем:

```
IF (*EXPRESSION (must return a boolean value)*)  
  THEN (*If returned value of EXPRESSION = TRUE*)  
    (*STATEMENT*);  
END_IF;
```

Щоб доповнити цю заготовку необхідним змістом треба курсор спочатку установити після оператора IF і надрукувати Cycle\_Count>10, а потім установити після оператора THEN і надрукувати Cycle\_Count:=0.




Для оголошення змінних, у цьому ж полі, курсор по черзі установити на кожне з імен і в панелі інструментів лівою клавішею миші клацнути по кнопці  (Variable). У діалоговому вікні Variables, що з'явиться, у рядку Name відобразиться ім'я вибраної змінної. Натиснути кнопку Apply, відкрити сторінку діалогу Common і у рядку Usage вибрати статус змінної VAR, а у рядку Data Type її тип BOOL. У рядку I/O address надрукувати фізичну адресу вихідного каналу модуля виводу, наприклад %QX0.0 для red1,


%QX0.1 для yellow1, %QX0.2 для green1, %QX1.0 для red2, %QX1.1 для yellow2 і %QX1.2 для green2. Натиснути ОК і діалогове вікно закриється, а оголошена змінна автоматично буде занесена у список змінних проекту.




Після оголошення змінних закрити робоче поле дії Action і зберегти її код, натиснувши у діалоговому вікні MULTIPROG, що з'явиться, кнопку ОК.

Відповідно до заданого алгоритму, умовою початку роботи світлофора є дворазове натиснення стартової кнопки. Програмування цієї умови здійснюється LD-мовою, використовуючи прямий зв'язок, при якому перехід не має тіла коду, а безпосередньо сполучається з LD-мережею.

Клацнути правою клавішею миші по переходу Start, у діалоговому вікні Transition, що з'явиться, активізувати Direct Connection, зробити порожнім рядок Name і натиснути ОК. Діалогове вікно закриється, а у робочому полі SFC - схеми після початкового кроку INIT залишиться безіменна смужка переходу з блакитною крапкою. Для створення умови переходу LD-мовою клацнути лівою клавішею миші у зручному місці робочого поля зліва від безіменного переходу.

В панелі інструментів натиснути послідовно кнопки  (Add contact right) і  (Left powerrail). В установленому місці робочого поля з'явиться LD- мережа 001 з лівою шиною живлення і контактом. В панелі інструментів натиснути кнопку  (Connect objects) і за допомогою курсору і лівої клавіші миші з'єднати прямий контакт LD- мережі з блакитною крапкою смужки безіменного переходу.

Лівою клавішею миші позначити лінію, що сполучає прямий контакт і перехід. В панелі інструментів клацнути по кнопці  (Edit Wizard) і з таблиці Group (Група), що з'явиться на екрані, вибрати функціональний блок лічильника STU. Двічі клацнути по блоку STU і у діалоговому вікні Variable properties, що з'явиться, змінити типове ім'я STU-1 на Svit\_Count. Натиснути послідовно кнопки Apply і ОК і на позначеному місці LD- мережі з'явиться лічильник.

Щоб задати параметри лічильнику, клацнути на функціональному блоці STU по синій крапці Reset-входу. В панелі інструментів натиснути кнопку  (Add contact left) і на Reset - вході лічильника з'явиться контакт C002. В панелі інструментів активізувати кнопку  (Connect objects) і за допомогою курсору, а також лівої клавіші миші з'єднати прямий контакт з шиною живлення LD- мережі. В панелі інструментів натиснути кнопку  Mark, щоб деактивізувати кнопку Connect objects.

Для оголошення змінної, яка має увімкнути світлофор, двічі клацнути лівою клавішею миші по контакту C000. У діалоговому вікні Contact/Coil Properties, що з'явиться, замінити типове ім'я C000 на Start. Натиснути кнопку Apply (Застосувати) і на сторінці Common зі списку Usage вибрати статус змінної VAR, зі списку Data Type - тип BOOL, а у рядку I/O address надрукувати фізичну адресу змінної Start - %IX0.0. Після натиснення ОК

оголошена змінна потрапить у відповідний список, а над контактом C000 з'явиться ім'я Start.

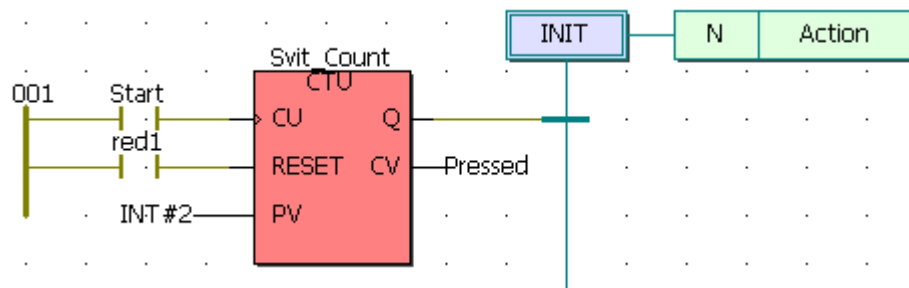
Щоб надати контакту C002 ім'я red1, який має скидати лічильник, необхідно двічі клацнути по контакту. У діалоговому вікні Contact/Coil Properties, що з'явиться, зі списку вже оголошених змінних вибрати red1, клацнути лівою клав'яшею миші і натиснути ОК.

Двічі клацнути по блакитній крапці PV-входу і у діалоговому вікні Variable properties, що з'явиться, у рядку Name надрукувати константу INT#2, щоб світлофор увімкнувся після дворазового натиснення стартової кнопки. Після натиснення ОК на PV-вході лічильника з'явиться INT#2.


Двічі клацнути по зеленій крапці CV-виходу лічильника і у діалоговому вікні Variable properties у рядку Name надрукувати Pressed - це внутрішня змінна, яка запам'ятовує поточне значення лічильника.

Натиснути кнопку Apply і на сторінці діалогу Common у рядку Usage вибрати VAR, а в рядку Date type (Тип даних) – INT. Після натиснення ОК у робочому полі поряд з CV-виходом з'явиться Pressed.

Програмний код умови першого переходу отримує наступний вигляд:



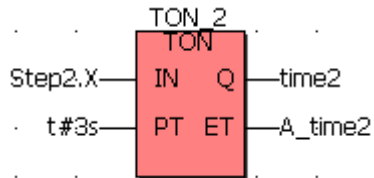
Умовами решти переходів є час роботи той чи іншої фази світлофорів. А дії кроків визначаються станами змінних світлофорів.

Щоб реалізувати програмування таймерів для кожного з переходів необхідно двічі клацнути лівою клав'яшею по відповідному переходу і у вікні Insert вибрати мову програмування. Після закриття діалогового вікна Insert відкриється робоче поле для створення програмного коду умови переходу. В панелі інструментів натиснути кнопку  і у вікні Edit Wizard знайти функціональний блок таймера TON. Для переносу екземпляра цього блока у робоче поле необхідно двічі клацнути по ньому лівою клав'яшею миші і у вікні Variables, що відкриється, призначити йому ім'я, або залишити запропоноване по замовчуванню. Після підтвердження імені, екземпляр блока зобразиться у робочому полі і йому потрібно визначити параметри, які забезпечать необхідний час затримки, відповідно до синтаксису вибраної мови.

Наприклад, для програмування умови переходу Time2 FBD-мовою, необхідно перенести екземпляр функціонального блока TON на її робоче поле і двічі клацнути по блакитній крапці входу IN. Відкриється діалогове вікно Variable Properties, в якому необхідно записати ім'я джерела вхідного сигналу (крок Step2) у вигляді Step2.X і натиснути ОК. Після цього двічі клацнути лівою клав'яшею миші по блакитній крапці входу RT. Відкриється вікно Variable Properties, в якому записати параметр таймера (3с) у вигляді T#3s і натиснути



ОК. Двічі клацнути по зеленій крапці виходу Q і у вікні Variable Properties записати ім'я переходу time2, що програмується. Двічі клацнути по зеленій крапці виходу ET і у вікні Variable Properties записати ім'я змінної, яка фіксує поточний час відліку таймера, і натиснути кнопку Apply (Застосувати). Відкриється сторінка діалогового вікна Common, в якій вибрати статус змінної VAR і її тип TIME. Після натиснення кнопки ОК на робочому полі залишиться запрограмована умова переходу Time2 у вигляді:



Умова переходу Time1 ST-мовою має вигляд:

```
TON_1(IN:=Step1.X(* BOOL *),PT:=t#10s(* TIME *));
time1(* BOOL *):=TON_1.Q;
A_time1(* TIME *):=TON_1.ET;
```

Умова переходу Time3 IL-мовою має вигляд :

```
LD Step3.X (* BOOL *)
ST TON_3.IN
LD t#3s (* TIME *)
ST TON_3.PT
CAL TON_3
LD TON_3.Q
ST time3 (* BOOL *)
LD TON_3.ET
ST A_time3(* TIME *)
```

Умова переходу Time4 ST-мовою має вигляд:

```
TON_4(IN:=Step4.X(* BOOL *),PT:=t#10s(* TIME *));
time4(* BOOL *):=TON_4.Q;
A_time4(* TIME *):=TON_4.ET;
```

Умова переходу Time5 ST-мовою має вигляд:

```
TON_5(IN:=Step5.X(* BOOL *),PT:=t#10s(* TIME *));
time5(* BOOL *):=TON_5.Q;
A_time5(* TIME *):=TON_5.ET;
```

Умова переходу Time6 ST-мовою має вигляд:

```
TON_6(IN:=Step6.X (* BOOL *),PT:=t#3s(* TIME *));
time6(* BOOL *):=TON_6.Q;
A_time6(* TIME *):=TON_6.ET.
```


Останній крок програми Step8 призначений для лічильника циклів роботи світлофора. Дія цього кроку запрограмована ST-мовою з використанням умовного оператора IF, як це здійснювалося при програмуванні початкового кроку INIT:

```

IF time6      (*EXPRESSION (must return a boolean value)*)
THEN Cycle_Count:=Cycle_Count+1    (*If returned value of EXPRESSION = TRUE*)

(*STATEMENT*);
END_IF;


```

В альтернативному розгалуженні умова переходу на крок INIT (при кількості циклів роботи світлофора>10) виконана ПЛ-мовою з використанням оператора GT з бібліотеки Operators вікна Edit Vizard, яке відкривається кнопкою :

```

LD Cycle_Count    (* IN1 as ELEMENTARY *)
GT 10    (* IN2 as ELEMENTARY *)
ST Reset    (* Result as BOOL *)

```


Умова переходу на крок Step1 (при кількості циклів роботи світлофора<10) запрограмована ПЛ-мовою з використанням оператора LE з бібліотеки Operators вікна Edit Vizard, яке відкривається кнопкою :

```


LD Cycle_Count    (* IN1 as ELEMENTARY *)
LE 10    (* IN2 as ELEMENTARY *)
ST Trans    (* Result as BOOL *)


```

Для перевірки програми використовується симулятор. Щоб його активізувати у дереві проекту клацнути правою клавішею миші по теці Resource і увійти в пункт меню Settings. З'явиться вікно Resource settings for IRC\_32, у якому позначити Simulation 1 і закрити діалог, клацнувши ОК.

Компіляція проекту відбувається після натиснення кнопки  (Make) в панелі інструментів. У вікні Build унизу екрана відобразяться помилки (Error) і попередження (Warning), якщо будуть знайдені протягом компіляції.

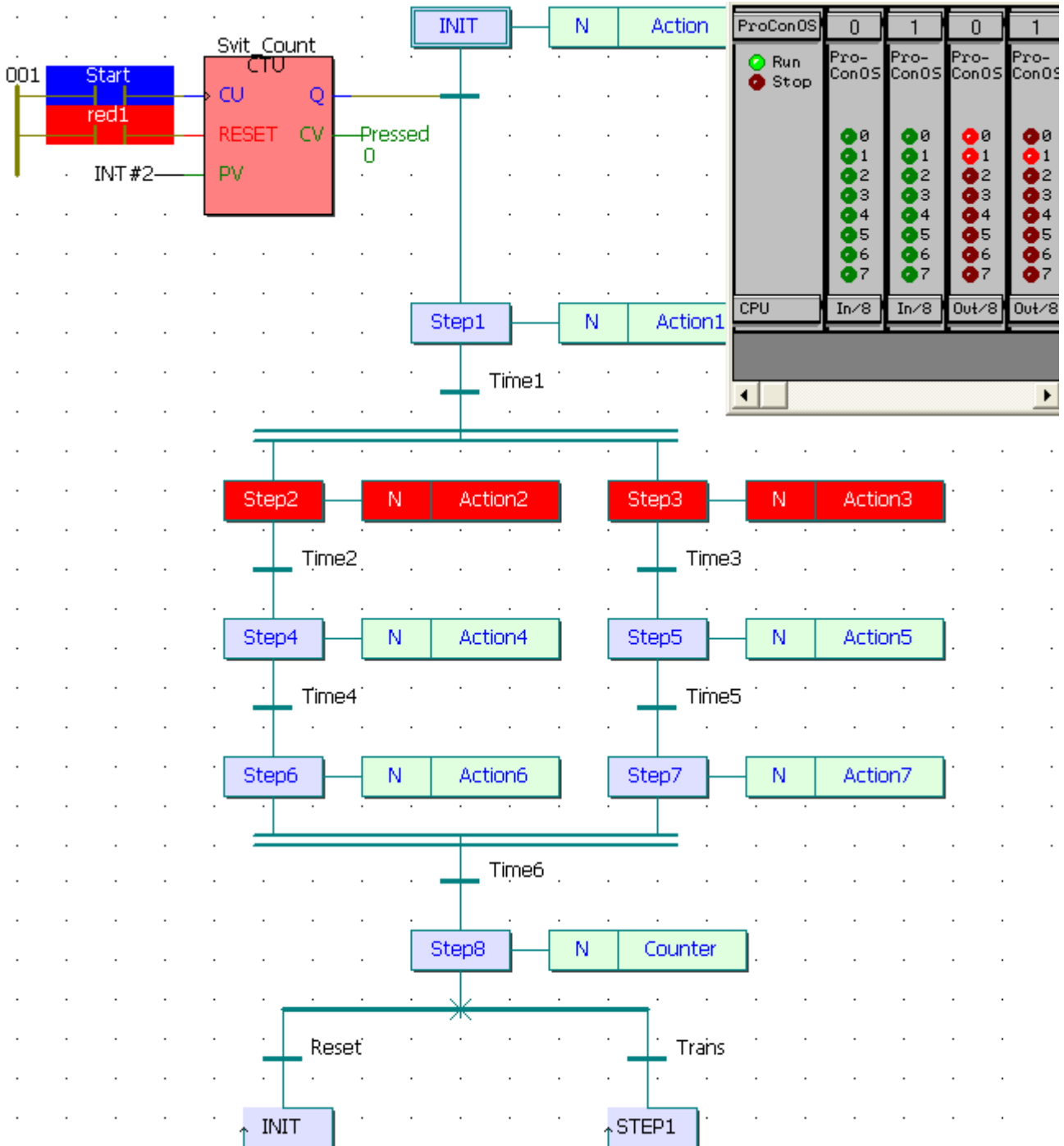
Щоб вивести на екран список помилок або попереджень необхідно клацнути по Error чи Warning. Після подвійного натиснення лівої клавіші миші на вказаній у списку помилці або попередженні, відкриється відповідна сторінка, яку необхідно скоригувати, а потім повторно компілювати проект.


Створений без помилок проект завантажується до симулятора за допомогою кнопки  (Project Control Dialog). У діалоговому вікні Resource, що відкриється, натиснути кнопку Download, а потім продублювати завантаження у частині Project вікна Download. Успішний процес завантаження проекту супроводжується бігом синьої стрічки унизу екрана.


У діалоговому вікні Resource натиснути кнопку Cold для холодного пуску симулятора, а в панелі інструментів клацнути по кнопці  (Debug on/off), щоб перейти в оперативний режим роботи. При цьому усі змінні позначаються відповідними кольорами. Істинний стан – червоним, а хибний – синім.

Лівою клавішею миші клацнути по DemoIo\_Driver унизу екрана, для відкриття симулятора I/O. Двічі подвійно клацнути лівою клавішею миші по

нульовому світлодіоду нульового модуля вводу In, оскільки саме цю адресу має змінна Start. Програма почне виконуватися, про що свідчить перехід червоного кольору від початкового кроку INIT до кроку STEP1 і далі, а також робота світлодіодів на симуляторі модулів 0 і 1 виводу Out, відповідно до адреси вихідних змінних:



Щоб завантажити проект до ПЛК необхідно зупинити роботу емулятора кнопкою Stop у вікні Resource і вийти з оперативного режиму за допомогою кнопки  (Debug on/off) в панелі інструментів. Клацнути правою клавішею миші по теці Resource, а потім по пункту меню Settings і у вікні Resource settings for IRC\_32 позначити COM1. Закрити діалог, клацнувши ОК. В панелі

інструментів натиснути кнопку  (Project Control Dialog), щоб з'явилося діалогове вікно Resource: ADAM 5510KW, і натиснути послідовно кнопки Stop, Reset і Download. У діалоговому вікні Download, що з'явиться, натиснути кнопку Download у частині Bootproject і проект завантажиться у пам'ять ПЛК, про що свідчить біг синьої стрічки вниз екрана.

У вікні Resource:(ADAM5510KW) знову натиснути кнопку Download, а потім натиснути кнопку Activate у частині Bootproject вікна Download. Після цього програму можна запустити, використовуючи Cold start у вікні Resource[1,27].

## Тестові та контрольні запитання до розділу 5

**I-й рівень запитань та тверджень дозволяє перевірити загальний рівень засвоєння матеріалу розділу і передбачає відповідь на запитання або твердження “Так” чи “Ні” .**

1. Виробником контролерів та модулів серії ADAM є корпорація Advantech.

2. Чи можна пристрої серій ADAM 4000 називати ПЛК?

3. Пристрої серії ADAM 5000 можуть об'єднуватися в багатоточкову мережу на базі інтерфейсу Ethernet або RS-485.

4. Чи потрібен модулям серії ADAM 6000 OPC–сервер для обміну даними зі SCADA-системами?

5. ПЛК серії ADAM 5000 складаються з центрального мікропроцесорного блока, клемноблокових з'єднувачів та модулів розширення.

6. Контролери серії ADAM 5000 умовно поділяють на три групи: ADAM 5510, ADAM 5511 та ADAM 5550.

7. ADAM 5000/485, ADAM 5000/CAN та ADAM 5510 є базовими блоками ПЛК серії ADAM-5000.

8. Чи має базовий блок контролера ADAM 5510 4-слотову пасивну панель, процесорний модуль, перетворювач напруги, три послідовні комунікаційні порти?

9. Контролер ADAM 5510 має два порти.

10. Усі лінії зв'язку, що обслуговують COM-порти контролера, мають однакову максимальну довжину.

11. Чи може використовуватися джерело з вихідною напругою 10...30В постійного струму для живлення контролера ADAM 5510?

12. Чи характеризують стан модулів вводу-виводу індикатори на лицевій панелі контролера ADAM5510KW?

13. Встановлення мережної адреси контролера ADAM 5510 здійснюється програмно.

14. В контролері ADAM 5511 на відміну від контролера ADAM 5510 реалізована програмна підтримка протоколу ModBus, що дозволяє

обмінюватися даними з будь-яким програмним забезпеченням верхнього рівня (SCADA) без використання спеціальних драйверів.

15. Чи можливо в ADAM 5511 за допомогою DIP-перемикачів встановлювати адресу контролера для передачі даних до протоколу «ModBUS RTU», задавати швидкість передачі даних та робочий COM-порт?

16. Контролер ADAM 5511 програмується 5 мовами стандарту MEK 6113.

17. Чи з'єднується контролер ADAM 5510KW з віддаленими модулями вводу – виводу через порт COM4 ?

18. На відміну від контролера ADAM 5511 контролер ADAM 5510KW відповідає усім вимогам міжнародного стандарту IEC-61131-3.

19. Чи програмується ADAM 5510KW за допомогою інструментальної системи KW MULTIPROG?

20. Чи передбачено в контролері ADAM 5511 «гаряче резервування» модулів вводу-виводу?

*Перевірте відповіді на сторінках: 1 - 108 с., 2 - 108 с., 3 - 108 с., 4 - 108 с., 5 - 108 с., 6 - 108 с., 7 - 109 с., 8 - 109 с., 9 - 110 с., 10 - 111 с., 11 - 111 с., 12 - 126 с., 13 - 111 с., 14 - 112 с., 15 - 113 с., 16 - 112 с., 17 - 125 с., 18 - 124 с., 19 - 127 с., 20 - 112...114 с.*

**II-й рівень запитань (тестові запитання) дозволяє перевірити рівень засвоєння матеріалу розділу із заглибленням у технічні особливості будови та структури контролерів Advantech та передбачає вибір правильної відповіді з наведених трьох варіантів.**

**1. Скільки і які порти має контролер ADAM 5510?**

- а) два послідовні комунікаційні порти та один порт мережі Ethernet;
- б) три послідовні комунікаційні порти;
- в) два послідовні комунікаційні порти.

**2. Як вибирається швидкість обміну даними для контролера ADAM5510 KW?**

- а) швидкість обміну даними є величиною постійною;
- б) швидкість обміну даними встановлюється програмно;
- в) за допомогою DIP – перемикачів.

**3. Для чого призначені годинники реального часу?**

- а) призначені для здійснення повторного запуску мікроконтролера у випадку виникнення позаштатних ситуацій;
- б) дозволяють вести облік загального часу роботи контролера у штатних та позаштатних ситуаціях;
- в) забезпечують можливість прив'язки до часу будь-яких подій, що фіксуються у об'єкті керування.

**4. Яку кількість каналів вводу-виводу забезпечують програмовані контролери серії ADAM 5000?**

- а) до 128 каналів дискретного вводу-виводу або 64 аналогові канали на один блок ADAM 5000;
- б) до 512 каналів дискретного вводу-виводу та 64 аналогові канали на один блок ADAM 5000;
- в) до 128 каналів дискретного вводу-виводу та 32 аналогові канали на один блок ADAM 5000.

**5. Скільки модулів вводу-виводу можна встановити в базовий блок контролера ADAM 5510?**

- а) вісім;
- б) чотири;
- в) необмежену кількість.

**6. Який порт контролера ADAM 5510 підтримує роботу в режимі інтерфейсу RS-485?**

- а) тільки COM 2;
- б) COM 2 та COM 3;
- в) тільки COM 1.

**7. Яке призначення має сторожовий таймер?**

- а) забезпечує можливість прив'язки до часу будь-яких подій, що фіксуються у об'єкті керування;
- б) забезпечує контроль за аварійними та нештатними ситуаціями, що фіксуються на об'єкті управління;
- в) для здійснення повторного запуску мікроконтролера при збоях у роботі програми керування.

**8. Пристрій ADAM-5018 – це ...**

- а) кінцевий модуль контролера;
- б) 7-канальний модуль аналогового вводу сигналів термопар;
- в) 16-канальний модуль дискретного виводу.

**9. Чим характеризується 16-канальний модуль дискретного вводу-виводу ADAM-5050?**

- а) кожному каналу модуля можна незалежно встановлювати діапазон вхідного або вихідного сигналу;
- б) кожному каналу модуля можна незалежно встановлювати статус вхідного або вихідного;
- в) у цьому модулі закладена функція дублювання каналів, що підвищує надійність його роботи.

**10. Завдяки чому суттєво спрощується підключення зовнішніх пристроїв до модулів контролера серії ADAM 5000?**

- а) завдяки тому, що зовнішні пристрої можна підключати до контролера через клемно-блокові з'єднувачі;
- б) завдяки тому, що клемні колодки дворядні;
- в) завдяки тому, що гвинтові клемні колодки роз'ємні.

**11. Для чого в ADAM 5511 призначений порт «COM3»?**

- а) для калібрування та конфігурування аналогових модулів вводу-виводу;
- б) для програмування контролера;

в) для передачі значень вхідних і вихідних сигналів на EOM.

**12. За допомогою якого програмного середовища програмується ADAM 5511?**

- а) за допомогою середовища розробки Borland Turbo C++ ;
- б) за допомогою середовища розробки KW MULTIPROG;
- в) за допомогою середовища розробки C++ Builder;

**13. Які способи кріплення має контролер ADAM 5510 при монтажі?**

- а) тільки на DIN-рейку;
- б) на вертикальну панель або на DIN-рейку;
- в) на вертикальну або горизонтальну панель.

**14. Скільки портів має контролер ADAM 5510KW?**

- а) чотири;
- б) три;
- в) два.

**15. Як створюється проект і код програми у середовищі KW MULTIPROG?**

- а) сконфігурувати контролер, виконати компіляцію, створити проектне дерево та розробити код програми, виконати компіляцію;
- б) сконфігурувати контролер, створити проектне дерево, розробити код програми, виконати компіляцію;
- в) створити проектне дерево, сконфігурувати контролер, розробити код програми, виконати компіляцію.

**16. Який інтерфейс підтримують модулі серії ADAM 6000 при створенні розподілених систем збору даних і управління?**

- а) RS-485;
- б) Ethernet;
- в) MPI, PROFIBUS-DP, ModBus TCP і CAN.

**17. Контролери якої серії Advantech програмно сумісні з контролерами серії S7-300 фірми Siemens?**

- а) контролери серії Advantech не сумісні з контролерами серії S7-300;
- б) контролери серії ADAM 5000;
- в) контролери серії ADAM 8000.

**18. За якою ознакою умовно поділяють контролери серії ADAM 5000?**

- а) за методом їх програмування;
- б) за конструктивними особливостями;
- в) за кількістю каналів вводу-виводу.

**19. На якому принципі побудовано програмований контролер ADAM 5510?**

- а) на блоковому принципі;
- б) на модульному принципі;
- в) на базі модулів розширення.

**20. На базі якого мікропроцесора створений процесорний модуль контролера ADAM 5510?**

а) на базі мікропроцесора 80188, який працює під керівництвом операційної системи ROM-DOS;

б) на базі мікропроцесора 80188, який працює під керівництвом операційної системи Windows;

в) на базі мікропроцесора 8188, який працює під керівництвом операційної системи DOS.

*Перевірте відповіді на сторінках: 1 - 110 с., 2 – 125...126 с., 3 - 111 с., 4 - 108 с., 5 - 109 с., 6 - 110 с., 7 - 111 с., 8 - 111 с., 9 - 112 с., 10 - 112 с., 11 - 114 с., 12 - 114 с., 13 - 112 с., 14 - 125 с., 15 – 128...140 с., 16 - 108 с., 17 - 108 с., 18 – 108...109 с., 19 - 109 с., 20 - 109 с.*

**III-й рівень запитань (контрольні запитання) дозволяє перевірити рівень засвоєння матеріалу із систематизацією отриманої у розділі інформації та передбачає відповіді, що побудовані на логічному аналізі або порівнянні із приведенням конкретних технічних параметрів (даних або прикладів).**

1. Накреслить структурно-логічну схему усіх засобів автоматизації фірми Advantech.

2. Виконайте порівняння контролерів ADAM-5510, ADAM-5510KW та ADAM 5511 за технічними характеристиками.

3. Систематизуйте інформацію по модулям вводу-виводу контролерів серії ADAM-5000 у вигляді таблиці з переліком усіх типів модулів, їх позначень та технічних характеристик.

4. Поясніть функції світлодіодів, що застосовуються у контролерах серії ADAM-5000.

5. Порівняйте призначення COM-портів в контролерах ADAM-5510, ADAM-5510KW, ADAM 5511 та поясніть принцип роботи з ними.

6. Визначте, які інтерфейси та протоколи підтримують ті або інші контролери фірми Advantech для організації обміну інформацією.

7. Перерахуйте, які параметри апаратної частини контролера треба знати для створення проекту та коду програми у середовищі KW MULTIPROG та Borland Turbo C++.

8. Поясніть принцип роботи у середовищі Borland Turbo C++ v.3.0 при створенні проекту для контролера ADAM 5511.

9. За рахунок яких технологій у контролерах фірми Advantech забезпечується висока надійність роботи пристроїв?

10. Поясніть принцип роботи у середовищі KW MULTIPROG при створенні коду програми для контролера ADAM 5510 KW.

*Перевірте відповіді на сторінках: 1 – 108...114 с., 2 – 108...114, 124...127 с., 3 – 111...112 с., 4 – 111...113, 126 с., 5 – 110...111, 113...114, 125 с., 6 – 108...114, 124...127 с., 7 – 114...123, 128...140 с., 8 – 114...123 с., 9 – 108...114, 124...127 с., 10 – 128...140 с.*



## 6 ПРОГРАМОВАНІ КОНТРОЛЕРИ КОРПОРАЦІЇ WAGO

Німецька корпорація WAGO, яка заснована у 1951 році, є світовим лідером в області термінальних технологій. Основною розробкою її вважається унікальна технологія з'єднання провідників Cage Clamp. Проте чимало уваги корпорація приділяє роботам в області I/O систем, що монтуються на DIN-рейку.

Розробка WAGO I/O-System призначена для організації віддаленого збору даних і управління на основі різних промислових мереж. Система дозволяє приймати і передавати дискретні, аналогові, числоімпульсні сигнали, а також обмінюватися даними з різними спеціальними пристроями.

Ідеологія WAGO I/O заснована на наданні розробнику максимальних можливостей в конфігуруванні, нарощуванні і обслуговуванні системи. З іншого боку, користувачеві надана можливість максимально гнучко змінювати кількість входів і виходів контролера використовуючи чотирьох, двох або одно-канальні модулі. Це дає значну економію засобів, за рахунок зменшення надмірності системи, у порівнянні з традиційними ПЛК що мають, як правило, 16/8 канальні модулі вводу-виводу.

### 6.1 Будова WAGO I/O-System

Конструктивно WAGO I/O-System, рис.6.1, складаються з модулів двох типорозмірів – мережного адаптера (51×65×100мм) і до 64 модулів вводу/виводу (12×64×100мм) для будь-якого типу сигналів.

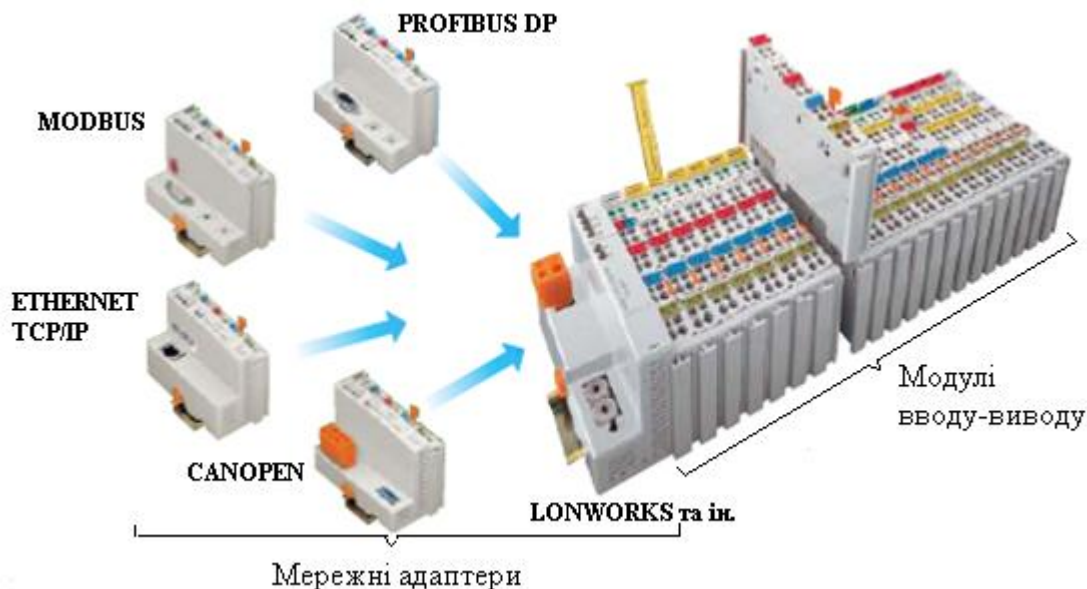


Рисунок 6.1 – Будова WAGO I/O-System

Мережні адаптери випускаються для роботи в мережах Profibus, Interbus-S, CANbus, CANopen, CAL, DeviceNET, ETHERNET і LIGHTBUS. Разом з модулями вводу-виводу, вони складають Fieldbus вузол, який завершується кінцевим модулем, рис.6.2. Для переводу вузла мережі з одного типу Fieldbus на

іншій немає необхідності міняти і перемонтовувати весь контролер – достатньо замінити лише мережний адаптер. Для мінімізації вартості мережні адаптери випускаються двох різновидів – з можливістю підключення дискретних та аналогових модулів і лише дискретних.

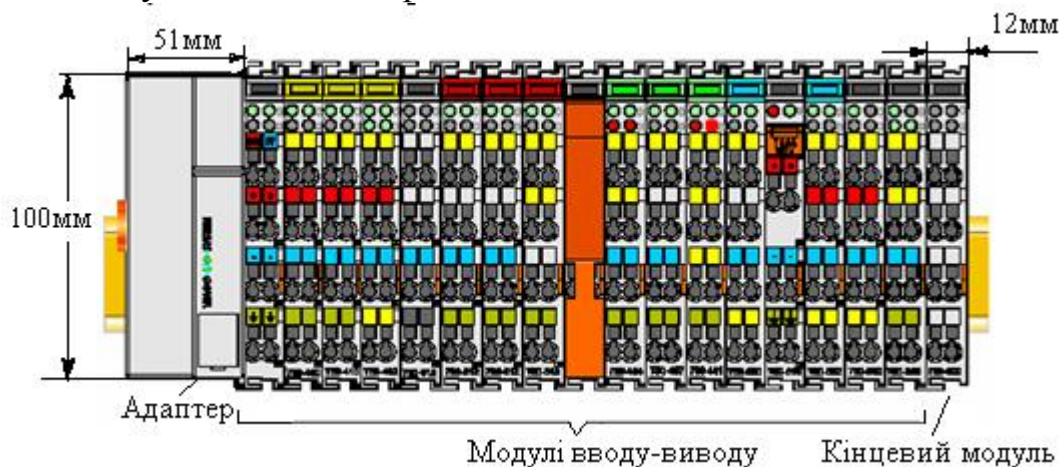


Рисунок 6.2 – Fieldbus вузол

Контролери WAGO I/O-System безініціативні, тому мають працювати під керуванням будь-якого інтелектуального пристрою, наприклад комп'ютера, на якому виконується SCADA-програма.

Вузлові контролери можуть бути пасивними або активними (програмованими). У промисловій мережі пасивні контролери виконують функції тільки підпорядкованих (Slave) і потребують підключення безпосередньо до робочої станції, виступаючої в ролі майстра мережі або до окремого контролера, який виконує роль майстра мережі. Активний контролер вузла мережі, окрім цього, може виконувати алгоритм, на підставі якого він і управляє станом своїх вихідних модулів, без участі комп'ютера верхнього рівня. В цьому випадку в мережі можна обійтися і без окремого управляючого майстер-контролера і без робочої станції.

Програмування контролерів здійснюється за допомогою технологічних мов програмування стандарту МЕК 61131-3. Завантаження програм може бути здійснено як локально, через діагностичний порт контролера WAGO I/O, так і дистанційно по мережі.

Основні технічні характеристики програмованих контролерів WAGO I/O:

- об'єм пам'яті програм: 32 кбайт;
- об'єм пам'яті даних: 32 кбайт;
- максимальне число програмних інструкцій: близько 3000;
- кількість одночасно виконуваних програм: 1;
- час циклу виконання програми: близько 3 мс для програми з 1000 інструкцій (включаючи час обміну з модулями вводу-виводу);
- система програмування - WAGO-IO-PRO 32, в стандарті МЕК 61131.3;
- підтримувані мови програмування: LD, FBD, SFC;
- вимоги по живленню: 24 В, 500 мА;
- діапазон робочих температур: 0-50°C.

Оскільки розподілені АСК ТП на основі контролерів з модулями вводу-виводу на 16 і 8 каналів мають значну надмірність (тобто частина каналів вводу-виводу залишається незадіяною), модулі контролера WAGO I/O-System розраховані на підключення тільки чотирьох, двох або одного каналу вводу-виводу. Наприклад, щоб реалізувати вузол збору даних в мережі CANBUS (DeviceNet) з трьома дискретними каналами вводу сигналів 0...24В, одним дискретним каналом виводу і одним каналом для підключення термопар, необхідно на монтажну DIN-рейку зліва направо послідовно встановити:

- адаптер мережі CANBUS (DeviceNet);
- два двоканальні модулі вводу дискретних сигналів і один одноканальний модуль виводу;
- один одноканальний модуль для підключення термопар;
- один кінцевий пасивний модуль (обов'язковий елемент контролера WAGO I/O-System).

В результаті такої конфігурації контролер має мінімальну надмірність по каналах (один канал дискретного вводу) в надзвичайно малих габаритах. При цьому, якщо дільниця модернізується і все устаткування необхідно підключити, наприклад, до мережі ProfiBus, то для цього потрібно буде тільки в кожному окремому вузлі мережі поміняти мережний адаптер на відповідний адаптер для ProfiBus з набору WAGO I/O-System.

У контролерів WAGO I/O-System відсутнє об'єднувальне шасі, яке присутнє практично в усіх ПЛК. Механічним з'єднувачем для окремих модулів вводу-виводу є стандартна монтажна DIN-рейка, рис.6.3. А шини живлення і управління реалізовані незвичайно.

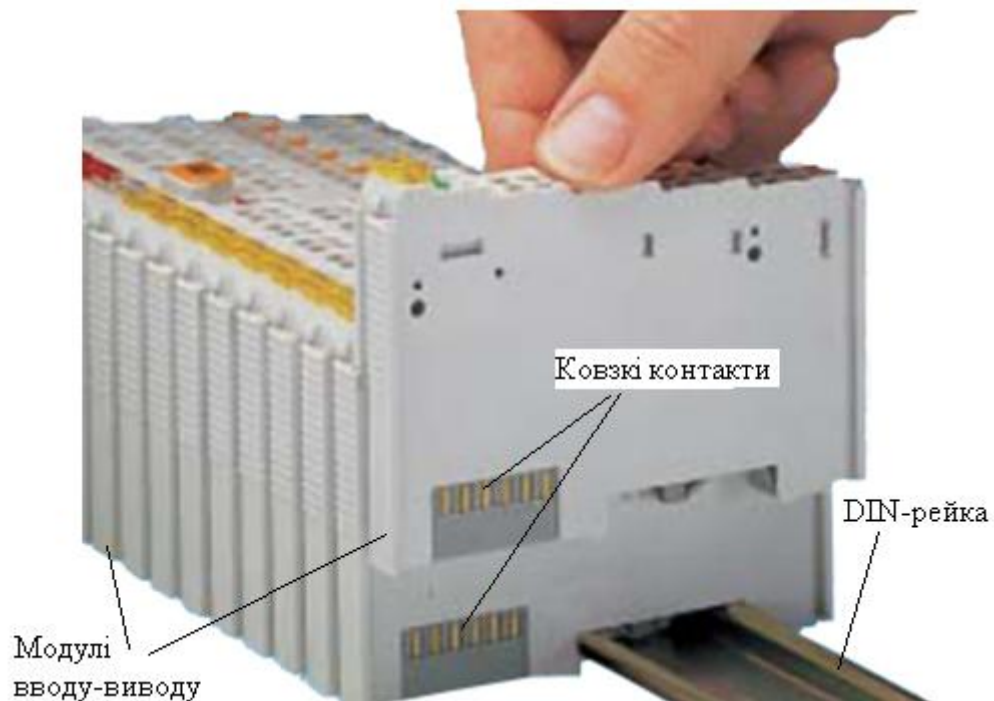


Рисунок 6.3 – Установлення модулів на 35мм DIN-рейку

Кожен модуль на лівому і правому боці має ковзкі контакти для з'єднання шини даних і шини напруги живлення. При послідовному установленні модулів на DIN-рейку ці шини надійно з'єднуються позолоченими, ковзкими контактами з самозачищенням і забезпечують скрізну передачу даних і живлення від модуля до модуля. За рахунок такого конструктивного рішення в контролері WAGO I/O-System практично відсутнє обмеження по кількості модулів вводу-виводу, що сполучаються разом. Мікропроцесор, що знаходиться усередині мережного адаптера, через послідовну шину даних безперервно сканує стан входів-виходів підключених модулів і за запитом майстер-контролера мережі передає їх в мережу або змінює стан виходів.

Ще однією конструктивною особливістю WAGO I/O-System є спосіб підключення зовнішніх кіл до контролера, який за рахунок використання пружинних клем цієї ж фірми, забезпечує стійке до вібрації, швидке і надійне підключення проводів перетином від 0,08 до 2,5 кв. мм.

Всі позитивні риси, що властиві клемам WAGO, повною мірою успадковані і контролером WAGO I/O-System[28].

## 6.2 Програмований контролер WAGO 750-841

WAGO 750-841, рис.6.4, є оновленою версією Ethernet TCP/IP програмованого контролера сімейства WAGO-I/O-SYSTEM, яка має значно збільшений об'єм пам'яті.

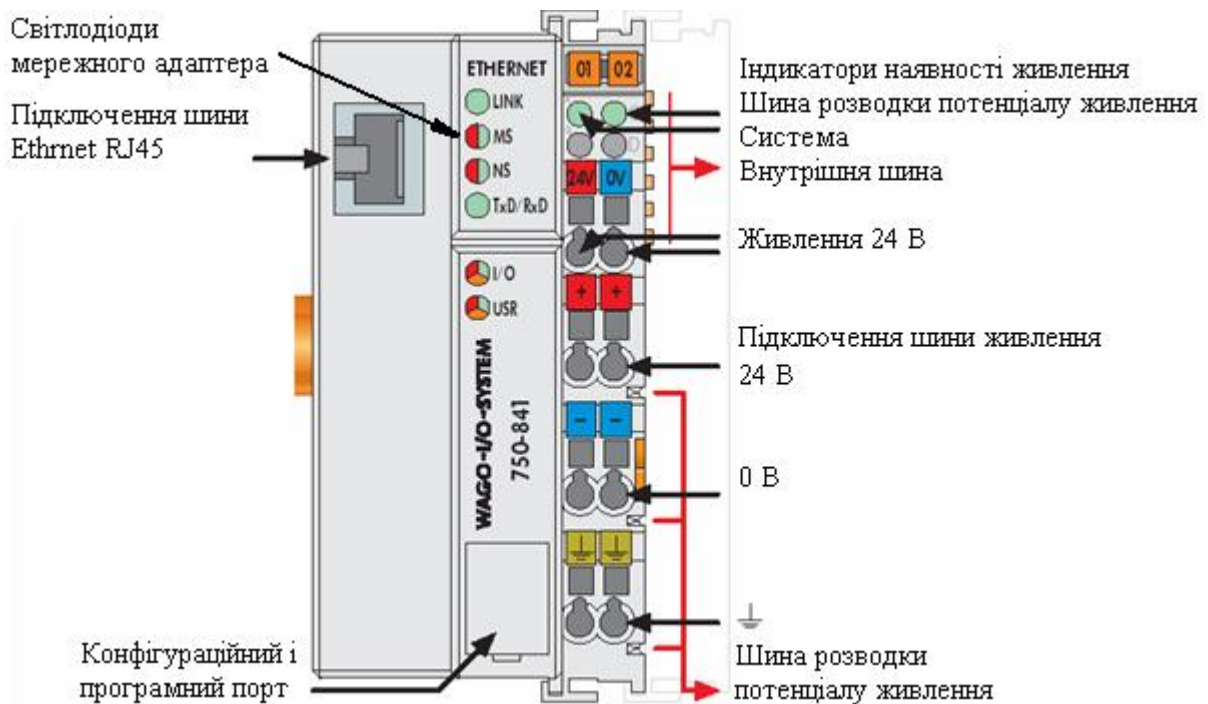


Рисунок 6.4 – Fieldbus контролер ETHERNET TCP/IP

Оснащений 100-Mbit портом Ethernet, контролер підтримує протоколи ModbusTCP і EtherNet/IP. WAGO 750-841 є безініціативним контролером, тому

працює під управлінням будь-якого інтелектуального пристрою, наприклад комп'ютера зі SCADA-програмою.

Він програмується інструментальною системою програмування WAGO-I/OPRO CAA (CoDeSys) відповідно до стандарту IEC 61131-3, використовуючи 512 Кбайт пам'яті програм, 128 Кбайт пам'яті даних і 24 Кбайт незалежної пам'яті.

Контролер побудований на базі 32-бітового процесора, який працює в мультитізадачному режимі і має вбудований годинник - таймер реального часу.

Контролер має напругу живлення =24В. Експлуатаційний режим контролера відображається за допомогою індикаторів у формі світлодіодів. В деяких випадках, вони є різноколірними (червоними/зеленими або червоними/зеленими/рожевими).

Від того, яким кольором в даний момент горить світлодіод можна визначити стан зв'язку. Характеристика свічення приведена в табл.б.1.

Таблиця 6.1 Характеристика світлодіодів лицевої панелі мережного адаптера

| Світлодіод     | Колір                    | Значення  |
|----------------|--------------------------|---|
| <b>LINK</b>    | зелений                  | Зв'язок фізичної мережі існує                                 |
| <b>MS</b>      | червоний/зелений         | MS-світлодіод указує статус модуля                            |
| <b>NS</b>      | червоний/зелений         | NS-світлодіод указує стан мережі                              |
| <b>TxD/RxD</b> | зелений                  | Відбувається обмін даними                                     |
| <b>IO</b>      | червоний/зелений/рожевий | Світлодіод В/В указує операцію вузла і повідомляє про помилки |
| <b>USR</b>     | червоний/зелений/рожевий | Світлодіод USR управляється програмою користувача             |
| <b>A</b>       | зелений                  | Статус наявності напруги в системі                            |
| <b>B or C</b>  | зелений                  | Статус надходження напруги на силовий контакт - перемичку     |

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

Конфігураційний і програмний порт зв'язку використовується для програмування контролера за допомогою середовища розробки WAGO-I/OPRO CAA (CoDeSys), а також для завантаження мікропрограм. Операційний вимикач режиму має три положення:

- у верхньому положенні - обробка програми активована;
- у середній позиції – програмований пристрій працює, додаток контролера зупинений;
- нижнє положення вимикача режиму - апаратні засоби скинуті, змінні встановлені в нуль або приведені в початковий стан.

Як частина WAGO-I/O-SYSTEM, контролер WAGO 750-841 на 100% сумісний з 750 серій модулів I/O. Тому функціональний набір модулів вводу-виводу є достатньо типовим - дискретний і аналоговий ввід-вивід, лічильники і

деякий набір спеціальних модулів. Дискретні модулі вводу-виводу працюють з сигналами від 5 до 250 В і можуть мати функцію діагностики. Серед дискретних модулів вводу одна модифікація чотириканальна (750-402), дві двоканальні і одна одноканальна, а серед модулів виводу сім модифікацій двоканальних і одна одноканальна. Наприклад, модуль дискретного вводу WAGO 750-402 має чотири канали і приймає із зовнішніх пристроїв (датчиків і т.ін.) керуючі сигнали постійного струму 24В. Кожен вхідний канал обладнаний фільтром. Для забезпечення гальванічної ізоляції між вхідними колами і внутрішньою шиною застосовані оптопари, рис.6.5.

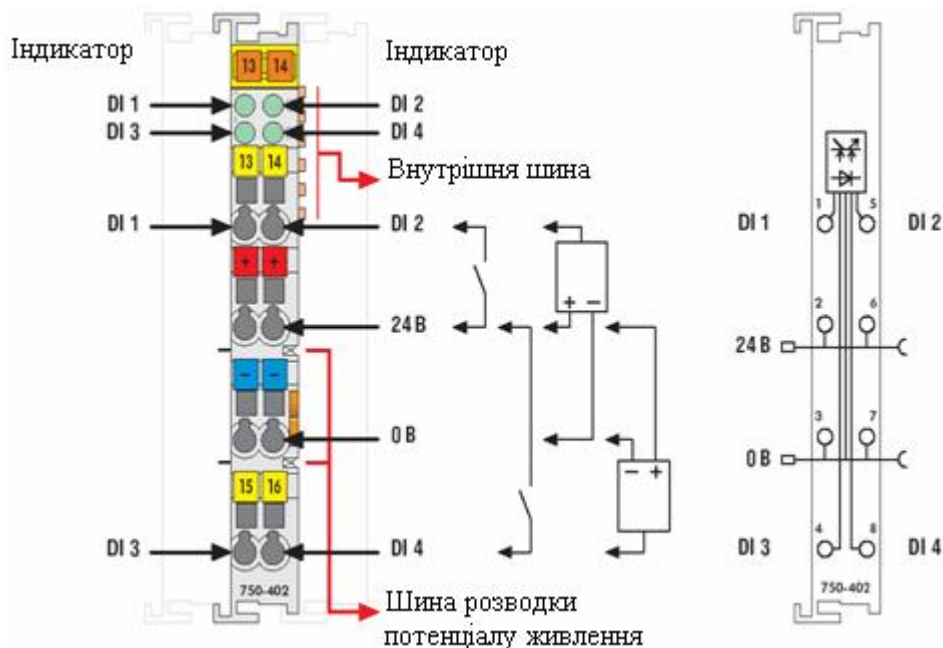


Рисунок 6.5 – Модуль дискретного вводу WAGO 750-402

Модуль дискретного виводу WAGO 750-504 призначений для управління навантаженнями постійного струму напругою 24 В. Всі виходи захищені від короткого замикання і гальванічно ізольовані від внутрішньої шини з допомогою оптопар.

Аналогові модулі підтримують всі стандартні види сигналів: 0 (4) –20 мА, 0–10 В, ±10В, термомпери, термометри опори. Серед аналогових модулів вводу дві модифікації чотириканальні і чотири двоканальні, а серед модулів виводу одна модифікація чотириканальна і дві двоканальні.

Наприклад, модуль аналогового вводу WAGO 750-461 має два канали і призначений для вводу сигналів з термометрів опору Pt або Ni. Можливе підключення 2 і 3-провідних датчиків. Лінеаризація характеристик виконується модулем автоматично. Аварії датчика відображаються червоним індикатором. Зелений індикатор відображає готовність даних і зв'язок з контролером. Екран заземляється на DIN рейку, рис.6.6.

Модуль аналогового виводу WAGO 750-554 має два канали і видає стандартні струмові аналогові сигнали 0-20 мА або 4-20 мА. Вихідний сигнал

гальванічно ізольований. Для живлення вихідного каскаду (струмової петлі) використовується потенціал шини розводки живлення.

Серед спеціалізованих модулів є квадратурний декодер (750-631), модулі інтерфейсів - RS232C (750-650), RS485 (750-653) і «струмова петля 20 мА» (750-651).

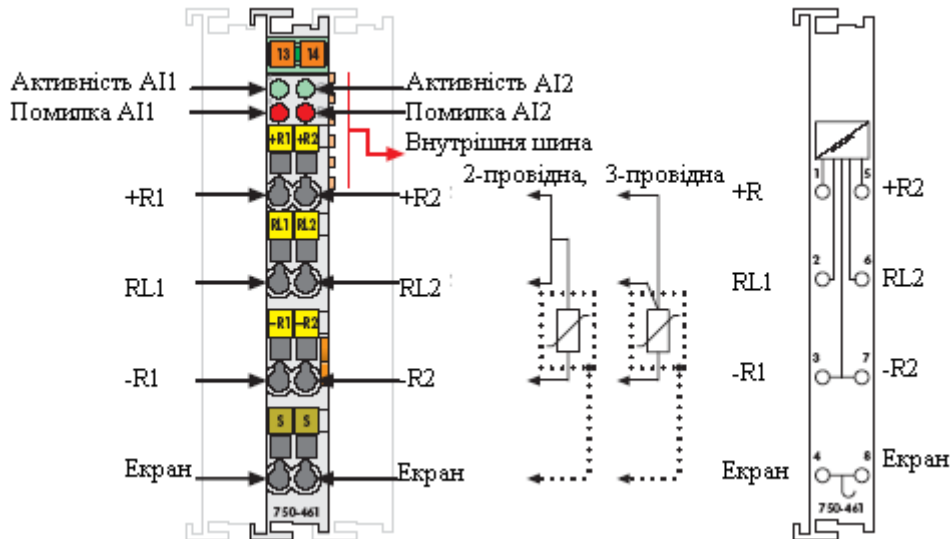


Рисунок 6.6 – Модуль аналогового вводу WAGO 750-461

Є також модулі живлення на різні напруги – 24В (750-601 і 750-602) і 230В (750-608) постійного або змінного струму, із запобіжником або без нього. Вони призначаються для живлення датчиків і виконавчих пристроїв.

Різні функції модулів позначені груповими прозорими кольоровими маркерами:

- жовтий – дискретні входи;
- червоний – дискретні виходи;
- зелений – аналогові входи;
- синій – аналогові виходи;
- безбарвний – живлення і модулі спеціальних функцій.

Всі модулі I/O SYSTEM 750 виконані на основі вібростійкого не обслуговуваного затиску CAGE CLAMP, який забезпечує системі найвищу надійність.

Кінцевий термінальний модуль WAGO 750-600 має обов'язково замикати Fieldbus вузол, тому що через нього забезпечується коректний обмін даними між контролером і модулями.

Коли є необхідність зробити Fieldbus-вузол розширеної конфігурації, то замість стандартного кінцевого модуля 750-600 використовують кінцевий модуль з розширенням внутрішньої шини 750-627, який встановлюється на несучу рейку в збірку вузла мережі останнім модулем. Разом із сполучним кабелем, і модулем 750-628 він дозволяє рознести збірку вузла мережі на декілька окремих складових, які мають один загальний контролер, рис.6.7.

Модуль для розширення внутрішньої шини 750-628 замінює контролер вузла мережі в наступній збірці модулів. Він є відповідною частиною модуля 750-627.

Електроніка модулів WAGO I/O може розміщуватися в корпусах трьох різних розмірів. Ширина модулів вводу-виводу і модулів спеціальних функцій - 12мм або 24мм. Електроніка контролерів віддаленого вводу-виводу і програмованих логічних контролерів встановлена в корпусі шириною 51 мм.

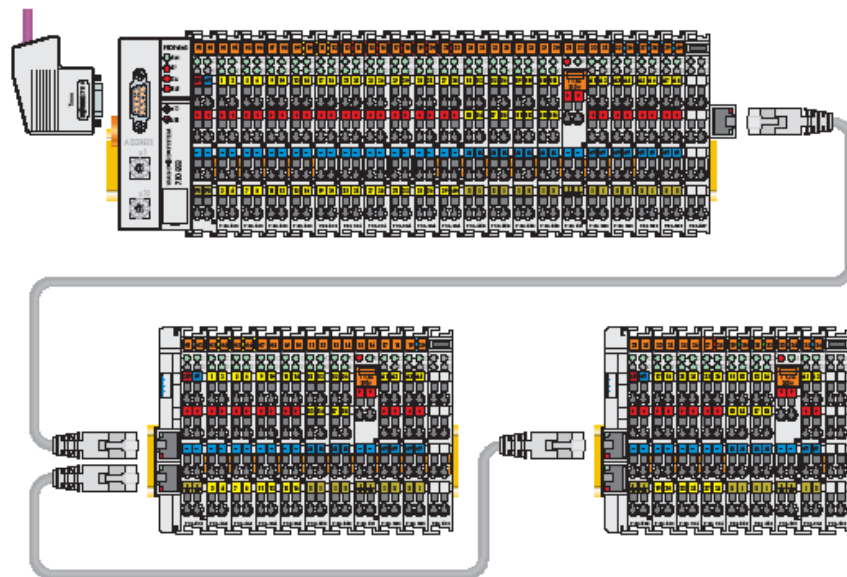


Рисунок 6.7 – Кінцевий модуль з розширенням внутрішньої шини 750-627

Модулі із спеціальними функціями можуть містити до трьох прохідних контактів шини розмноження живлення. Залежно від призначення, модулі можуть мати до чотирьох світлодіодних індикаторів (12мм корпус) або до восьми індикаторів (24мм корпус).

Модифікація WAGO - I/O - SYSTEM 753, рис. 6.8, виготовляється із

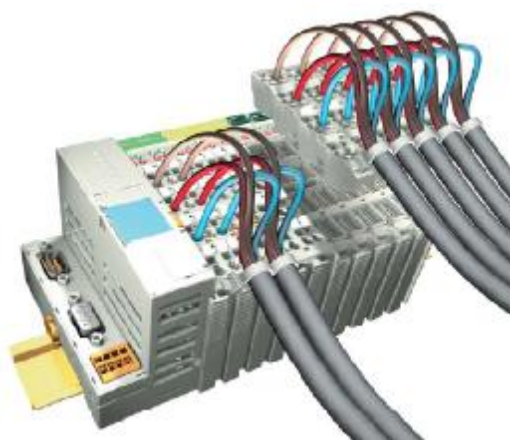


Рисунок 6.8 – Штекерне з'єднання WAGO-I/O-SYSTEM серії 753



штекерним сполученням, яке забезпечує заміну модулів вводу-виводу без порушення існуючої проводки, зняття штекера без застосування будь-якого інструменту, комутацію штекера окремо від модуля, що підвищує зручність при монтажі. Окрім цього контролер може бути протестований окремо від об'єкту, а також можливе комбінування модулів серій 750 і 753 в одній збірці контролера. Вбудований фіксатор дозволяє фіксувати проводи при монтажі.

При використанні контролерів у загазованих або заповнених приміщеннях використовуються корпуси, які виготовляються з неіржавіючої або листової сталі, алюмінію або поліестеру. Корпуси мають необхідну кількість отворів і кабельних сальників. Кожний тип корпусу має чотири розміри[29,30].

### **6.3 Програмування контролерів WAGO**

Програмування контролерів виробництва WAGO здійснюється за допомогою широко відомої інструментальної системи CoDeSys (скорочення від слів Controller Development System). Як і будь-яке хороше середовище, що задовольняє вимогам MEK 61131-3, CoDeSys включає 5 спеціалізованих редакторів для кожної із стандартних мов програмування: релейно-контактні схеми (LD), Функціональні блокові діаграми (FBD), Список Інструкцій (IL), Структурований текст (ST), Послідовні функціональні схеми (SFC). Редактори підтримуються великою кількістю допоміжних інструментів, які прискорюють програмування. Редактори FBD, LD і SFC автоматично розміщують свої графічні елементи і сполучення відповідно до структури діаграми. Це прискорює ввід, гарантує логічно узгоджене відображення і практично позбавляє від ручного вводу.

CoDeSys допомагає користувачеві при створенні оголошень змінних. Змінні і їх властивості задаються в діалоговому вікні, яке відкривається автоматично при введенні нового ідентифікатора. За бажанням текст коду і оголошення змінних автоматично форматується і виділяється кольором. Це полегшує сприйняття і позитивно впливає на якість і зрештою на ефективність роботи.

Коли необхідно ввести ім'я змінної, ключове слово, назву підпрограми з бібліотеки або з поточного проекту на допомогу приходять асистент вводу. Асистент вводу зводить до мінімуму ручну працю і пов'язані з цим помилки, оператор просто вибирає потрібні елементи із структурованого відсортованого списку.

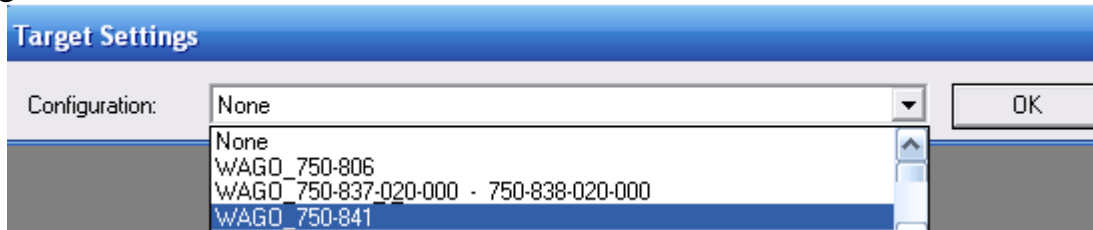
Вбудований компілятор створює швидкий машинний код безпосередньо з MEK-додатку. CoDeSys має багато передових засобів, що допомагають у налагодженні, тестуванні і супроводі додатків[30].

#### **6.3.1 Методика створення проекту у системі програмування CoDeSys**

Створити FBD-мовою проект автоматичного регулювання продуктивності живильника сипких матеріалів за ПД-аналоговим законом регулювання.

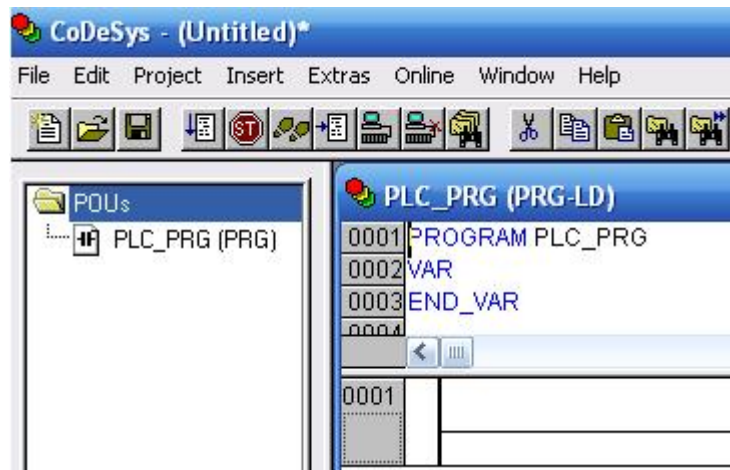
## *Розроблення проекту*


Запустити CoDeSys V2.3 і, використовуючи в меню шлях File-> New, лівою клавішею миші відкрити вікно настройки 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 FBD-мову, вибрати Program, як тип POU, залишити головний шаблон PLC\_PRG і натиснути ОК.

На екрані з'являться два вікна, зліва вікно організатора проекту POUs, а справа робоче поле для створення програмного коду:



Щоб здійснити конфігурування контролера, у нижній частині організатора проекту POUs треба відкрити вкладку  (Resources), лівою клавішею миші позначити в неї 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).

Хоча для реалізації регулювання продуктивності живильника сипких матеріалів потрібні тільки аналогові модулі вводу-виводу, при конфігуруванні контролера необхідно вказати всі скомпоновані в ньому модулі. Припустимо, що в контролері встановлені два модулі вводу – аналоговий AI 0750-0461 2 AI RT 100 (RTD) і дискретний DI 0750-0402 4 DI 24V DC 3.0ms, а також два модулі виводу - аналоговий AO 0750-0554 2 AO 4-20mA і дискретний DO 0750-0504 4 DO 24V DC 0.5mA.

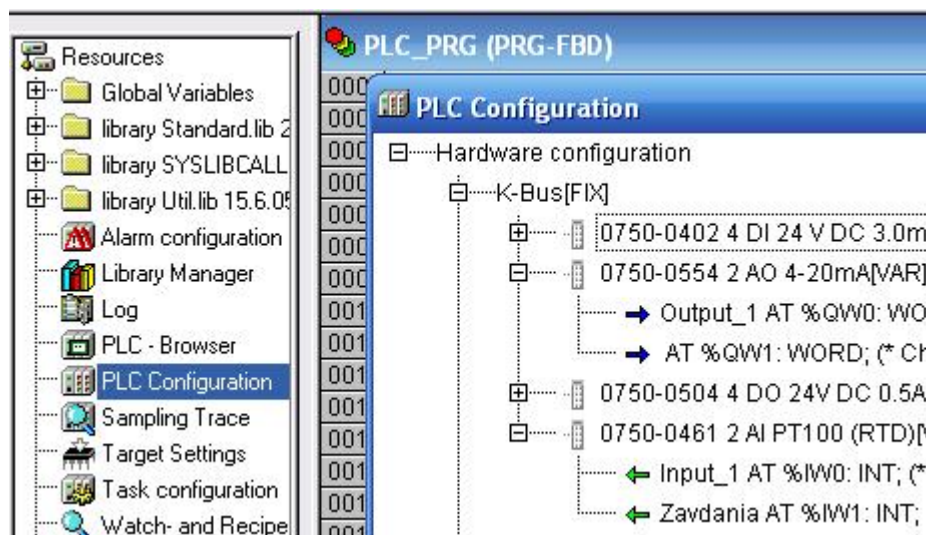
Щоб додати дискретний модуль вводу у проект, необхідно у лівому вікні відкрити каталог, двічі натиснувши ліву клавішу миші на теці “Digital Input”, і кнопкою Insert пересунути модуль DI 0750-0402 4 DI 24V DC 3.0ms у праву частину вікна конфігурації. Вибрати з теки “Analog Output” аналоговий модуль виводу АО 0750-0554 2 АО 4-20mA і за допомогою кнопки Insert пересунути його у праву частину вікна конфігурації. Таким самим чином пересунути у праве вікно дискретний модуль виводу з каталогу Digital Output і аналоговий модуль вводу з каталогу Analog Input.

Кінцевий модуль 750-600 при конфігуруванні модулів вводу – виводу не враховується. Закрити вікно, натиснувши кнопку “OK”.

Відкрити “K-Bus[FIX]” і натиснути ліву клавішу миші. Через вкладку 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 присвоїти ім'я змінним, яка відстежує сигнал з датчика витрати і, яка керує продуктивністю живильника.

Оскільки при керуванні використовується модуль аналогового виводу двічі клацнути лівою клавішею по його назві АО 0750-0554 2 АО 4-20mA, щоб відкрити канали з адресами. Для оголошення змінної навести курсор на адресу відповідного каналу модуля, натиснути ліву клавішу миші і адресу охопить рамка. Після повторного натиснення лівої клавіші відкриється текстове поле, в якому ввести ім'я вихідної змінної Output\_1 для адреси %QW0. Так само призначити ім'я вхідної змінної Input\_1 для адреси %IW0 і зовнішнього завдання Zavdania для адреси %IW1 модуля вводу AI 0750-0461 2 AI PT 100 (RTD):



### ***Створення програмного коду***

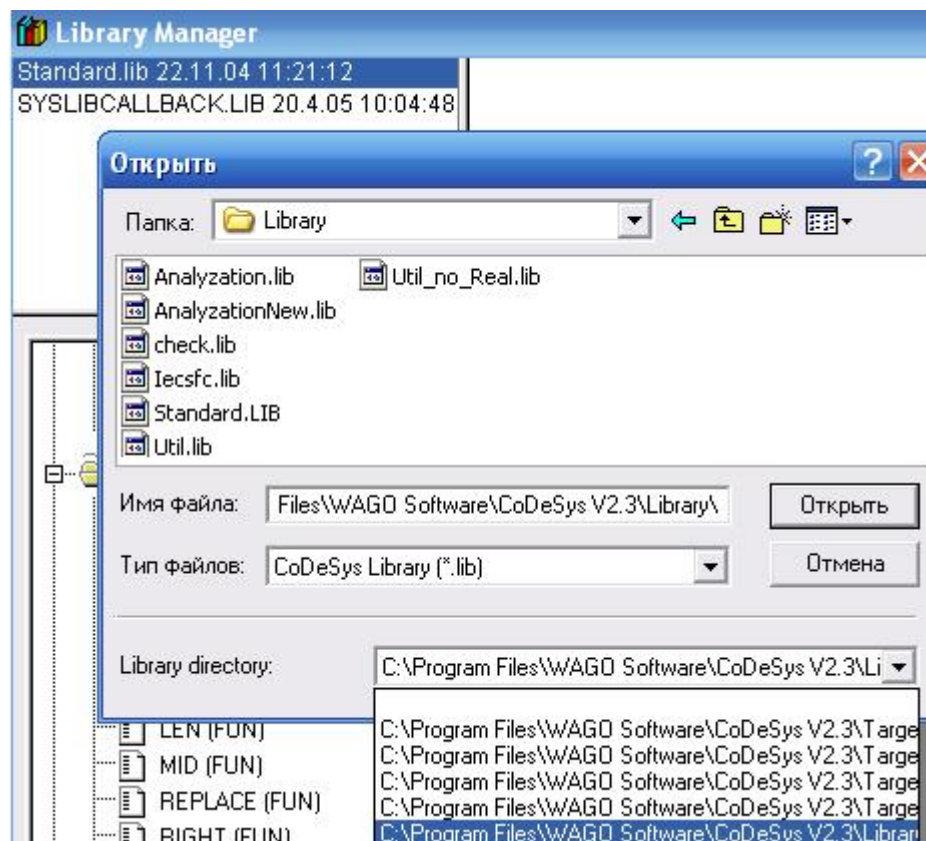
Пропорційно-інтегрально-диференційний закон (ПІД) – це найбільш широко вживаний закон в автоматичних регуляторах.

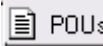
Передаточна функція ПІД – алгоритму керування має вигляд:

$$Y = Y\_OFFSET + KP \left( e(t) + \frac{1}{TN} \int_0^{TN} e(t) + TV \frac{de(t)}{dt} \right)$$

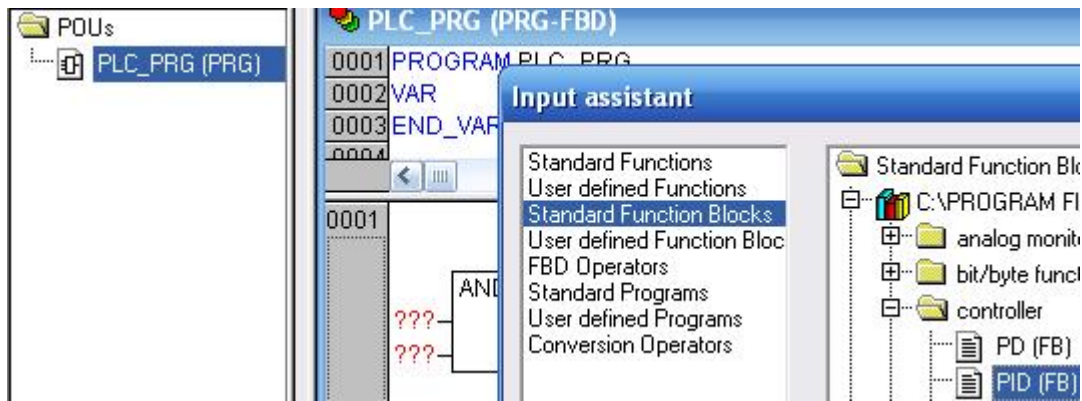
де  $Y\_OFFSET$  – стаціонарне значення вихідного сигналу,  $KP$  – коефіцієнт передачі,  $TN$  – постійна інтегрування,  $TV$  – постійна диференціювання,  $e(t)$  – сигнал неузгодження (SET\_POINT-ACTUAL).

Щоб створити функціональний блок ПІД-регулятора у вікні розробки проекту, необхідно в Library Manager (Менеджер бібліотеки), який знаходиться на вкладці Resources організатора проекту, додати бібліотеку UTIL.LIB. Для цього двічі натиснути ліву клавішу миші на Library Manager, щоб викликати його вікно, і натиснути на ньому праву клавішу миші. З меню, що відкриється, вибрати Additional Library і вибрати директорію C:\Program Files\WAGO-Software\CoDeSys V2.3\Library\UTIL.LIB. Після натиснення лівої клавіші миші, Util.Lib з'явиться у вікні бібліотеки. Позначити лівою клавішею миші цей файл і натиснути кнопку «Открыть», щоб його завантажити:



Для створення програми з функціональним блоком ПІД-регулятора, кнопкою  відкрити вкладку POU's організатора програми і двічі клацнути лівою клавішею миші по PLC\_PRG. Відкриється робоче поле, на якому установити курсор праворуч від ???, натиснути праву клавішу миші та в меню, що відкриється, виконати команду Vox. З'явиться блок AND.

Натиснути F2 на клавіатурі EOM, зі списку, що з'явиться, вибрати Standart Function Blocks->Controller->PID(FB) і натиснути клавішу ОК:



З'явиться блок ПІД-регулятора, натиснути клавішу Enter.

Клацнути лівою клавішею миші по знакам питання і задати ім'я – Regulator. Натиснути на клавіатурі клавішу Enter і у вікні Declare Variable, що з'явиться, вказати клас Var і тип PID змінної.

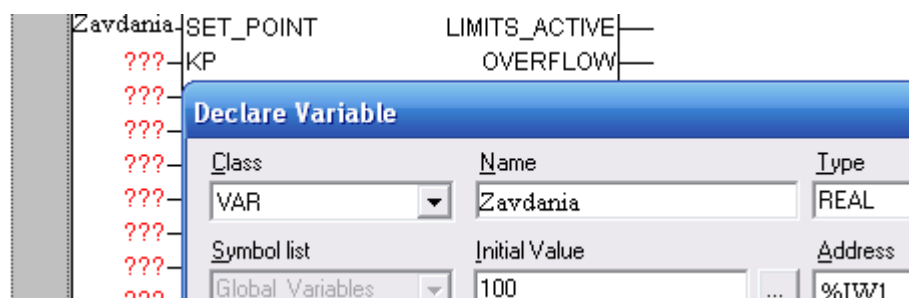
Натиснути ОК і для усіх входів і виходів функціонального блока оголосити змінні, керуючись інтерфейсом функціонального блока ПІД-регулятора:

|                  |              |                                |
|------------------|--------------|--------------------------------|
| <b>ACTUAL</b>    | <b>REAL</b>  | сигнал зворотного зв'язку      |
| <b>SET_POINT</b> | <b>REAL</b>  | завдання                       |
| <b>KP</b>        | <b>REAL</b>  | коефіцієнт                     |
| <b>TN</b>        | <b>DWORD</b> | постійна інтегрування          |
| <b>TV</b>        | <b>DWORD</b> | постійна диференціювання       |
| <b>Y_OFFSET</b>  | <b>REAL</b>  | стаціонарне значення           |
| <b>Y_MIN</b>     | <b>REAL</b>  | мінімальне допустиме значення  |
| <b>Y_MAX</b>     | <b>REAL</b>  | максимальне допустиме значення |
| <b>MANUAL</b>    | <b>BOOL</b>  | ручний режим                   |
| <b>RESET</b>     | <b>BOOL</b>  | скидання                       |

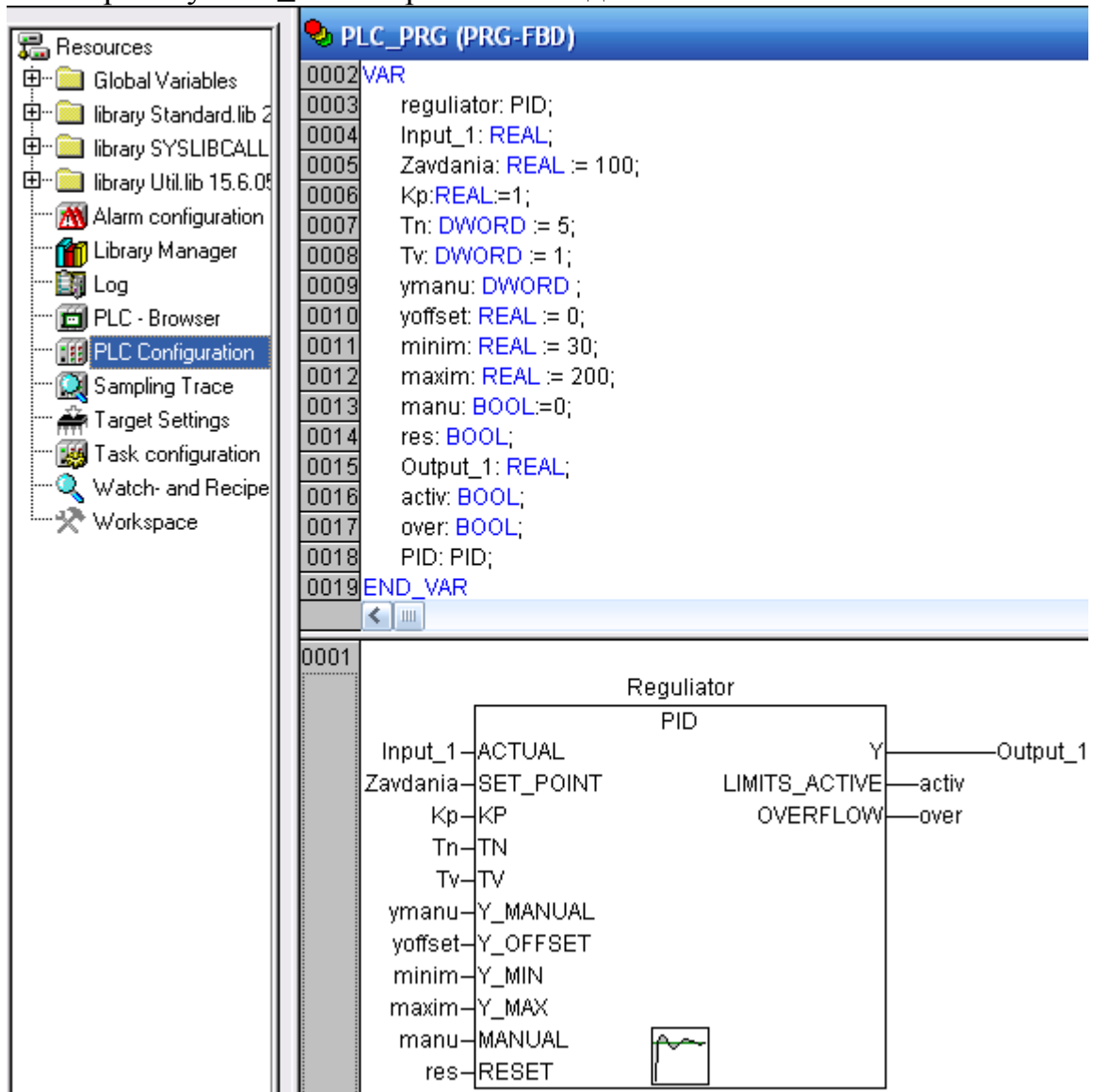
та виходами:

|                      |             |                                     |
|----------------------|-------------|-------------------------------------|
| <b>Y</b>             | <b>REAL</b> | керуюча дія                         |
| <b>LIMITS_ACTIVE</b> | <b>BOOL</b> | ознака досягнення порогових значень |
| <b>OVERFLOW</b>      | <b>BOOL</b> | ознака помилки переповнення         |

Щоб оголосити входні змінні, клацнути лівою клавішею миші по знакам питання біля відповідного входу, надрукувати ім'я і клацнути клавішею Enter, відкриється вікно Declare Variable, в якому заповнити вільні рядки:




Для оголошення вихідної змінної Y натиснути праву клавішу миші на квадраті біля цього виходу і з меню, що відкриється, вибрати Assign. З'являться знаки питання біля виходу. Натиснути на клавіатурі клавішу Enter і у вікні Declare Variable, що з'явиться, оголосити змінну Output\_1. Установити курсор біля іншого виходу, надрукувати актив і клацнути клавішею Enter для відкриття вікна оголошення змінної. Таке саме зробити з останнім виходом Overflow. Робоче поле проекту PLC\_PRG отримає вигляд:





Для перевірки програми виконати компіляцію проекту, використовуючи в меню шлях Project -> Build або клавішу F11 на клавіатурі ЕОМ. Результати компіляції з'являться у нижній частині робочого екрану. Кращий результат: 0 Error(s), 0 Warning(s). Якщо є помилки, то перейти у вікно повідомлень і клацнути кнопкою миші на тексті повідомлення. Це призведе до переміщення курсору в місце, де виникла помилка.

Для перегляду роботи функціонального ПІД-блока в панелі інструментів увійти у меню Online, позначити опцію Simulation Mode і натиснути ліву клавішу миші на опції Login. На робочому полі активізуються змінні. Логічні покажуть у відповідному кольорі початковий статус, а дійсні - початкове




значення. Щоб змінити, при необхідності, початкове значення змінної необхідно двічі клацнути по її імені і у вікні, що з'явиться, встановити нове значення. Для збереження встановленого значення виконати команду Online->Write Values.

Для запуску програми, в меню Online натиснути лівою клав'яшею миші опцію Run або кнопку  в панелі інструментів.

Значення вихідної змінної Output\_1 почнуть змінюватися відповідно до ПДД закону регулювання.


Щоб зупинити виконання програми треба лівою клав'яшею миші в панелі інструментів натиснути кнопку , або в меню Online опцію Stop. Для того щоб програму скинути, натиснути лівою клав'яшею миші кнопку  в панелі інструментів, або опцію Logout в меню Online.

Для наочного дослідження динаміки роботи функціонального блока необхідно провести трасування даних. Трасування дозволяє переглядати значення змінних у певному відрізку часу. Для переходу у вікно трасування на вкладці ресурсів організатора проекту вибрати Sampling Trace і двічі клацнути лівою клав'яшею миші. Для визначення списку трасованих змінних і настроювання параметрів трасування призначене діалогове вікно Trace Configuration, яке викликається з меню, що відкривається правою клав'яшею миші. Для додавання вихідної змінної «PLC\_PRG.Output\_1» ПДД-регулятора необхідно використати поле вводу Input of trace variable. Після завершення вводу натиснути кнопку Insert і назва змінної з'явиться у списку змінних Variables і у рядку Trace вікна Sampling Trace. Надати колір змінної, що трасується, вибравши її в одному з кольорових рядків вікна.

При роботі у вікні трасування на панелі інструментів відображаються кнопки    (Start Trace, Read Trace, Stop Trace), які призначені для управління накопиченням і відображенням даних. Аналогічний набір команд доступний в меню Extras:

- Extras->Start Trace - конфігурація трасування передається в ПЛК, і починається процес запису значень змінних в кільцевий буфер;
- Extras->Read Trace - значення змінних зчитуються з буфера трасування і відображаються у вигляді графіків;
- Extras->Auto Read Trace - якщо вибрана ця команда, то поточний буфер трасування читається автоматично і значення безперервно зображаються на екрані.

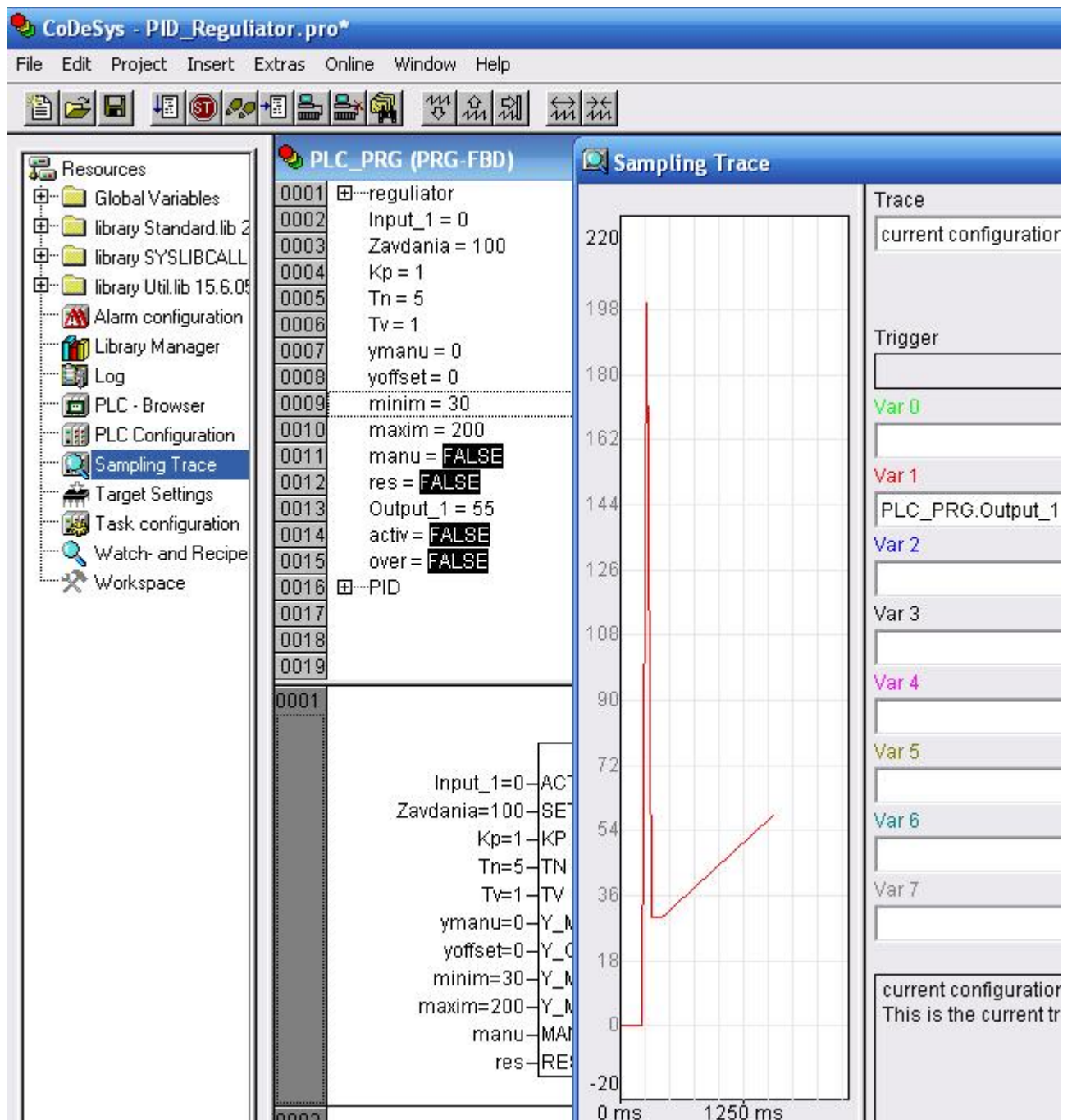
Запуск програми здійснюється в режимі симуляції, використовуючи Online->Simulation Mode.

Для запуску трасування у меню Online натиснути лівою клав'яшею миші Login, а в меню Extras спочатку - Start Trace, а потім Auto Read Trace. Щоб у вікні Sampling Trace розпочалось відображення графіка у панелі інструментів натиснути кнопку  (Run).

Щоб змінити початкове значення змінних необхідно після входу в меню Online-> Login двічі клацнути по змінній і у вікні, що з'явиться, встановити

нові значення. Для збереження встановлених значень виконати команду Online->Write Values [14].

Вкладка ресурсу проекту, робоче поле програмного коду і вікно трасування ПІД-алгоритму мають вигляд:



### *Сполучення з контролером*

При першому підключенні контролера до CoDeSys необхідно виконати певні настройки. У меню Online відкрийте діалог Communication parameters. Натиснути кнопку New для настроювання нового сполучення. Якщо контролер запускається з даного комп'ютера вибрати драйвер RS232, і перевірити параметри вікна конфігурації, які містять наступні дані установки:



Baudrate=19200, Parity=Even, Зупинка bits=1, Моторола byteorder=No. В меню Online вибрати Communication Parameters, а в меню Gatenay -> Local.

Переконалися в правильності настройки каналу сполучення. Зняти прапорець в меню Online -> Simulation. З'єднання з контролером встановлюється з середовища програмування CoDeSys командою Online->Login в панелі інструментів. Команда Online -> Run запускає проект [30].

## Тестові та контрольні запитання до розділу 6

**I-й рівень запитань та тверджень дозволяє перевірити загальний рівень засвоєння матеріалу розділу і передбачає відповідь на запитання або твердження “Так” чи “Ні” .**

1. Розробка WAGO I/O-System призначена для організації віддаленого збору даних і управління на основі різних промислових контролерів.
2. Чи дозволяє WAGO I/O змінювати кількість входів і виходів контролера, використовуючи чотири-, дво- або одно - канальні модулі?
3. Fieldbus вузол WAGO I/O-System завжди завершується кінцевим модулем.
4. Для живлення контролера потрібна напруга 24В.
5. Чи мають контролери WAGO I/O-System об'єднувальне шасі?
6. Модулі контролера забезпечують гальванічну ізоляцію між вхідними колами і внутрішньою шиною.
7. Кількість модулів вводу-виводу для одного контролера складає біля 128.
8. Зовнішні кола підключаються до контролера WAGO I/O- System за допомогою пружинних клем.
9. Модулі WAGO I/O-System позначені кольоровими маркерами для визначення їх поточного стану.
10. Контролер WAGO 750-841 сумісний з 750 серій модулів I/O.
11. Чи має контролер WAGO 750-841 можливість рознесення збірки вузла мережі на декілька окремих складових?
12. Операційний перемикач режимів на мережному адаптері дозволяє переводити контролер у два різні режими.
13. Чи підтримує контролер WAGO 750-841 мережу Ethernet ?
14. Контролери WAGO I/O-System у мережі можуть бути провідними та підпорядкованими.
15. Контролер WAGO 750-841 є безініціативним контролером.
16. Чи програмуються контролери WAGO-750 841 за допомогою інструментальної системи CoDeSys?
17. Програмування контролерів WAGO I/O-System здійснюється за допомогою мови програмування C++.
18. Чи може контролер WAGO 750-841 працювати під управлінням SCADA-системи?
19. Чи містить CoDeSys 5 спеціалізованих редакторів для кожної із стандартних мов програмування?

20. Чи має середовище CoDeSys засоби розробки у проекті підсистеми візуалізації даних?

*Перевірте відповіді на сторінках: 1 - 145 с., 2 - 145 с., 3 - 145 с., 4 - 146 с., 5 - 147 с., 6 - 150 с., 7 - 145 с., 8 - 148 с., 9 - 151 с., 10 - 150 с., 11 - 152 с., 12 - 149 с., 13 - 149 с., 14 - 146 с., 15 - 149 с., 16 - 153 с., 17 - 153 с., 18 - 146 с., 19 - 153 с., 20 - 159 с.*

**II-й рівень запитань (тестові запитання) дозволяє перевірити рівень засвоєння матеріалу розділу із заглибленням у технічні особливості будови та структури контролерів WAGO I/O-System та передбачає вибір правильної відповіді з наведених трьох варіантів.**

**1. З чого складається Fieldbus вузол WAGO I/O-System?**

- а) тільки з модулів вводу-виводу;
- б) з мережного адаптера та до 64 модулів вводу-виводу;
- в) з мережного адаптера, до 64 модулів вводу-виводу і кінцевого модуля.

**2. Чим відрізняються активні базові контролери WAGO I/O-System від пасивних?**

- а) активні базові контролери можуть бути підключені до промислової мережі;
- б) активні базові контролери додатково виконують алгоритми керування станом своїх вихідних модулів;
- в) активні базові контролери можуть обмінюватись інформацією з комп'ютером верхнього рівня.

**3. На яку кількість каналів розраховані модулі контролера WAGO I/O-System?**

- а) модулі контролера WAGO I/O- System можуть мати до 16 каналів;
- б) модулі контролера WAGO I/O- System розраховані на 8 і 16 каналів;
- в) чотири, два і один.

**4. Навіщо кожний модуль контролера WAGO I/O- System має контакти на боковій грані?**

- а) для створення шин живлення і управління;
- б) для фізичного приєднання модулів вводу-виводу один до одного;
- в) для створення мережі ETHERNET.

**5. Що знаходиться у нижній частині мережного адаптера?**

- а) конфігураційний і програмний порт, операційний перемикач режимів;
- б) мережний порт;
- в) рознім для підключення шини живлення.

**6. За рахунок чого виробники контролерів WAGO I/O-System уникають надмірності каналів вводу- виводу?**

- а) за рахунок випуску малоканальних контролерів;
- б) за рахунок випуску малоканальних модулів вводу-виводу не більше ніж на чотири входи-виходи;

в) за рахунок випуску малоканалних модулів вводу-виводу не більше ніж на вісім входів-виходів.

**7. Для чого потрібен кінцевий термінальний модуль WAGO 750-600?**

- а) для забезпечення коректного обміну даними між центральним процесором і модулями;
- б) для захисту останнього модуля від впливу навколишнього середовища;
- в) для забезпечення обміну даними між контролерами у мережі.

**8. Яке призначення має модуль WAGO 750-628?**

- а) дозволяє об'єднати контролери у мережу;
- б) дозволяє рознести збірку вузла мережі на декілька окремих складових, які мають один загальний контролер;
- в) це термінальний модуль, що забезпечує коректний обмін даними між модулями та контролером.

**9. Яка модифікація WAGO-I/O-SYSTEM має штекерне з'єднання із зовнішніми колами?**

- а) 841;
- б) жодна з модифікацій WAGO-I/O-SYSTEM не виробляється із штекерним з'єднанням зовнішніх кіл;
- в) 753.

**10. Контролер WAGO 750-841 є оновленою версією програмованого контролера сімейства WAGO-I/O-SYSTEM, який:**

- а) має значно збільшений набір модулів вводу/виводу;
- б) має нові комунікаційні можливості;
- в) має значно збільшений об'єм пам'яті.

**11. Дискретні модулі вводу-виводу WAGO-I/O-SYSTEM можуть працювати з сигналами:**

- а) від 5 до 250 В;
- б) від 0 до 24В;
- в) від 0 до 220В.

**12. WAGO I/O-System – це:**

- а) назва корпорації;
- б) система, яка призначена для організації віддаленого збору даних і управління на основі різних промислових мереж;
- в) модифікація програмно-логічного контролера серії WAGO.

**13. Мережні адаптери WAGO I/O-System можуть підтримувати :**

- а) Profibus, Interbus-S, CANbus, CANopen, CAL, DeviceNET, ETHERNET і LIGHTBUS мережі;
- б) тільки польові шини;
- в) Profibus, CANbus, DeviceNET, ETHERNET мережі.

**14. Як здійснюється завантаження програм у контролер WAGO I/O-System?**

- а) локально, через діагностичний порт контролера;
- б) за допомогою функціональних клавіш на панелі контролера;

в) локально, через діагностичний порт контролера та дистанційно по мережі.

**15. У контролерах WAGO I/O-System може одночасно виконуватися...**

- а) одна програма;
- б) стільки програм, скільки контурів управління;
- в) дві програми.

**16. При якій температурі навколишнього середовища може працювати контролер WAGO I/O?**

- а) від  $-20$  до  $+50$  °С;
- б) від  $0$  до  $+50$  °С;
- в) від  $-5$  до  $+80$  °С.

**17. Світлодіоди лицевої панелі мережного адаптера відображають:**

- а) статус модуля, наявність мережного зв'язку та обміну даними, помилки, наявність напруги живлення;
- б) статус модуля та наявність напруги живлення;
- в) наявність мережного зв'язку та обміну даними, помилки.

**18. У складі контролерів WAGO-I/O-SYSTEM є модулі, які забезпечують живлення датчиків і виконавчих пристроїв напругою:**

- а) 24В і 230В постійного або змінного струму;
- б) 24В і 36В постійного струму;
- в) 36В і 230В постійного або змінного струму.

**19. Які редактори CoDeSys мають засоби прискорення процесу створення програм?**

- а) IL, LD та ST;
- б) FBD, LD та ST;
- в) FBD, LD та SFC.

**20. Яка послідовність створення проекту у середовищі CoDeSys?**

- а) запустити середовище CoDeSys, вибрати контролер, здійснити конфігурування контролера, присвоїти імена вхідним і вихідним змінним та визначити, як здійснюється управління входами-виходами;
- б) відкрити вікно настройки цільової платформи, вибрати контролер, тип ROU, здійснити конфігурування контролера, визначити, як здійснюється управління входами-виходами, присвоїти імена вхідним і вихідним змінним, виконати компіляцію проекту;
- в) запустити середовище CoDeSys, вибрати контролер та тип ROU, здійснити конфігурування контролера, визначити, як здійснюється управління входами-виходами, виконати компіляцію проекту.

*Перевірте відповіді на сторінках: 1 - 145 с., 2 - 146 с., 3 - 147 с., 4 - 148 с., 5 - 149 с., 6 - 147 с., 7 - 151 с., 8 - 152 с., 9 - 152...153 с., 10 - 148 с., 11 - 150 с., 12 - 145 с., 13 - 145 с., 14 - 146 с., 15 - 146 с., 16 - 146 с., 17 - 149 с., 18 - 151 с., 19 - 153 с., 20 - 153...155 с.*

**III-й рівень запитань (контрольні запитання) дозволяє перевірити рівень засвоєння матеріалу із систематизацією отриманої у розділі інформації та передбачає відповіді, що побудовані на логічному аналізі або порівнянні із приведенням конкретних технічних параметрів (даних або прикладів).**

1. Визначте основні позиції ідеології WAGO I/O-System.
2. Накресліть загальну структурну схему контролера WAGO I/O-System та поясніть призначення основних елементів апаратної частини.
3. Поясніть функції світлодіодів, що застосовуються у контролерах WAGO I/O-System.
4. Які мережні технології закладено в контролері WAGO-I/O-SYSTEM для організації обміну інформацією.
5. Перерахуйте типи модулів вводу-виводу контролерів WAGO I/O-System та їх основні технічні характеристики.
6. Наведіть конструктивні особливості контролерів WAGO I/O-System при їх компонуванні та підключенні до об'єкту управління та до складу АСУТП.
7. Розкрийте основні функції мережного адаптера.
8. Визначте технічні характеристики контролера за якими він порівнюється з контролерами інших виробників.
9. Наведіть перелік типів та рівнів сигналів, які може приймати та видавати контролер WAGO 750-841.
10. Поясніть принцип роботи у середовищі CoDeSys при створенні проекту для контролерів WAGO I/O-System.

*Перевірте відповіді на сторінках: 1 – 145...153 с., 2 – 145...153 с., 3 – 149...151 с., 4 - 145, 149 с., 5 – 150...153 с., 6 – 145...153 с., 7 - 149 с., 8 – 145...153 с., 9 – 150...151 с., 10 – 153...155 с.*

## 7 ПРОГРАМОВАНІ ЛОГІЧНІ КОНТРОЛЕРИ SIMATIC

Торгова марка SIMATIC німецької фірми Siemens відома у світі, як синонім технічних і програмних засобів для автоматизованих систем управління. Найбільш популярною розробкою є «Система комплексної автоматизації SIMATIC», яка об'єднує окремі рішення системної автоматизації в єдине ціле - від апаратури "польового" рівня до управління процесом. Це досягається за допомогою інтегрованих в систему засобів конфігурування і програмування - мікропроцесорних контролерів (SIMATIC S7), засобів управління даними в системі комунікацій за допомогою спеціалізованих комп'ютерів (SIMATIC M7) і системами управління (SIMATIC C7).

За допомогою програмованих контролерів S7-200, S7-1200, S7-300 і S7-400, що випускаються під маркою SIMATIC S7, перекриваються всі області їх застосування при розв'язанні задач автоматизації. Проте найбільше поширення отримали модульні функціонально розширювані вироби серій S7-300 і S7-400, які використовуються в системах низької і високої продуктивності.

Програмним забезпеченням системи автоматизації SIMATIC є пакет STEP7. Розробка проектів в STEP 7 здійснюється за допомогою мов програмування, відповідних міжнародному стандарту DIN EN 6.1131-3: STL ("statement list" - список мнемонік, Assembler-подібна мова), LAD ("ladder diagram" - "контактний план", у вигляді логічних схем), FBD ("function block diagram" - "функціональний план", мова функціональних блок-схем) і пакет SCL, що поставляється за окремим замовленням ("Structured Control Language" - "структурована мова управління", Pascal-подібна мова високого рівня).

### 7.1 Програмовані контролери SIMATIC S7-300

Контролери SIMATIC S7-300, рис.7.1, широко використовуються при автоматизації технологічних процесів.



Рисунок 7.1 – Програмований контролер SIMATIC S7-300

Модульна структура, простота реалізації на їх основі децентралізованих систем управління, дружній інтерфейс користувача і можливість

експлуатації без примусового охолодження, перетворюють пристрої сімейства SIMATIC S7-300 в привабливий засіб розв'язання самих різних задач автоматизації як на середньому виробничому рівні, так і у складі великої розподіленої автоматизованої системи.

Як проектно-компонований виріб, SIMATIC S7-300 у своєму складі має:

- стійки (Racks) - для розміщення модулів і з'єднання їх між собою;
- модулі центрального процесора (Central Processing Unit, CPU) - для зберігання і обробки програми користувача;
- модулі блоків живлення (Power Supply, PS) - для подачі електроенергії до внутрішніх пристроїв від мережі змінного струму напругою 120/230В або від джерела постійного струму напругою 24/48/60/110В;
- сигнальні модулі (Signal Modules, SM) для вводу-виводу дискретних і аналогових сигналів з різними електричними і часовими параметрами;
- комунікаційні процесори (Communications Processors, CP) - для підключення до мереж PROFIBUS, Industrial Ethernet, AS-Interface або організації зв'язку за PtP (point to point) інтерфейсом;
- функціональні модулі (Function Modules, FM) – для виконання складної або критичної за часом обробки сигналів незалежно від CPU;
- інтерфейсні модулі (Interface Modules, IM) – для підключення до базового блока (стійка з CPU) стійок розширення вводу-виводу;
- шинний кабель PROFIBUS - для сполучення абонентів підмережі MPI або PROFIBUS між собою;
- повторювач RS 485 - для посилення сигналів в підмережі MPI або PROFIBUS, а також для з'єднання сегментів підмережі MPI або PROFIBUS;
- пристрій програмування (PG або PC) - для конфігурування, параметризації та програмування з пакетом тестування S7-300 програмного забезпечення STEP 7;
- кабель PG - для з'єднання програматора PG/PC з CPU.

Конструкція контролера характеризується високою гнучкістю і зручністю обслуговування. Всі модулі встановлюються на стійку контролера S7-300 і фіксуються в робочих положеннях гвинтами. Розмір конструктиву контролера — 200x125x120 мм, маса — до 1,2 кг[31].

### 7.1.1 Стійки контролера

Програмований контролер (або станція) SIMATIC S7-300 складається з центрального пристрою і, залежно від потреб, одного або декількох пристроїв розширення. Стійка, що містить центральний процесор (CPU), модуль електроживлення (PS), модулі SM, FM і CP називається центральним пристроєм (CU). Коли в центральній стійці недосить місця для модулів вводу-виводу або необхідно деякі I/O-модулі, а може і всі винести за межі центральної стійки, використовуються стійки розширення (EM). Для S7-300 у якості стійки застосовуються профільні шини, які бувають двох видів:

- готові до монтажу з чотирма отворами для кріпильних болтів і одним болтом для проводу заземлення (4- типорозміри);

- двометрова профільна шина, яку можна довільно скорочувати для конструкцій, що вимагають спеціальної довжини. У неї немає ні отворів для кріпильних болтів, ні болта для заземлення. Все це виготовляється на місці.

Модулі на профільній шині розміщуються, починаючи зліва, в наступній послідовності:

- блок живлення (PS);
- центральний процесор (CPU);
- інтерфейсні модулі (IM);
- сигнальні модулі (SM), функціональні модулі (FM), комунікаційні модулі (CP).

Зазвичай модулі в стійках сполучаються за допомогою двох шин - шини входів-виходів (I/O- шина, периферійна шина або Р-шина) і комунікаційної шини (або К-шина). Р-шина розроблена для високошвидкісного обміну вхідними і вихідними сигналами, а К-шина – для обміну великими об'ємами даних. Комунікаційна шина зв'язує CPU і інтерфейс пристрою програмування (MPI) з функціональними модулями і комунікаційними процесорами.

Об'єднання модулів в єдину систему виконується за допомогою шинних з'єднувачів, що входять в комплект поставки і встановлюються з тильної сторони, як перемички поміж суміжними модулями.

У центральну стійку контролера S7-300 може бути встановлено до 8 модулів. Якщо ця однорівнева конфігурація виявляється недостатньою, то можна реалізувати багаторівневу.

Для монтажу на декількох стійках (до чотирьох) потрібні інтерфейсні модулі (IM), які продовжують шину S7-300 від однієї стійки до іншої. CPU завжди знаходиться на нульовій стійці[31].

### **7.1.2 Інтерфейсні модулі**

Інтерфейсні модулі (IM) призначені для підключення стійок розширення до базового блока контролера SIMATIC S7-300. Це дозволяє використовувати в системі локального вводу-виводу до 32 модулів різного призначення. Існують три модулі розширення - IM 365, IM 360, IM 361 і пара інтерфейсних модулів IM 365.

Інтерфейсний модуль IM 360 забезпечує інтерфейс для нульової стійки контролера S7-300. Інтерфейсний модуль IM 361 створює інтерфейс для стійок 1, 2 і 3 і має джерело живлення 24В постійного струму. Струм, на задній шині S7-300, не перевищує 0,8 А. Передача даних з IM 360 в IM 361 або з IM 361 в IM 361 здійснюється через сполучний кабель 368. Максимальна відстань між інтерфейсними модулями не перевищує 10 м.

Пара інтерфейсних модулів IM 365 забезпечує інтерфейсний зв'язок між стійками 0 і 1. Вона має загальне джерело живлення на 1,2 А, з яких до 0,8 А може бути використано в одній стійці. Між модулями постійно підключений сполучний кабель завдовжки 1м. У стійку 1 встановлюються тільки сигнальні модулі.



Модулі ІМ 365 дозволяють створювати 2-рядні конфігурації, а модулі ІМ 360 і ІМ 361 – 2-, 3- і 4-рядні конфігурації.

На рис.7.2 показано приклад максимальної конфігурації, який передбачає розміщення модулів в конструкції S7-300 на 4 стійках.

У разі потреби в стійках розширення встановлюються додаткові блоки живлення [31].

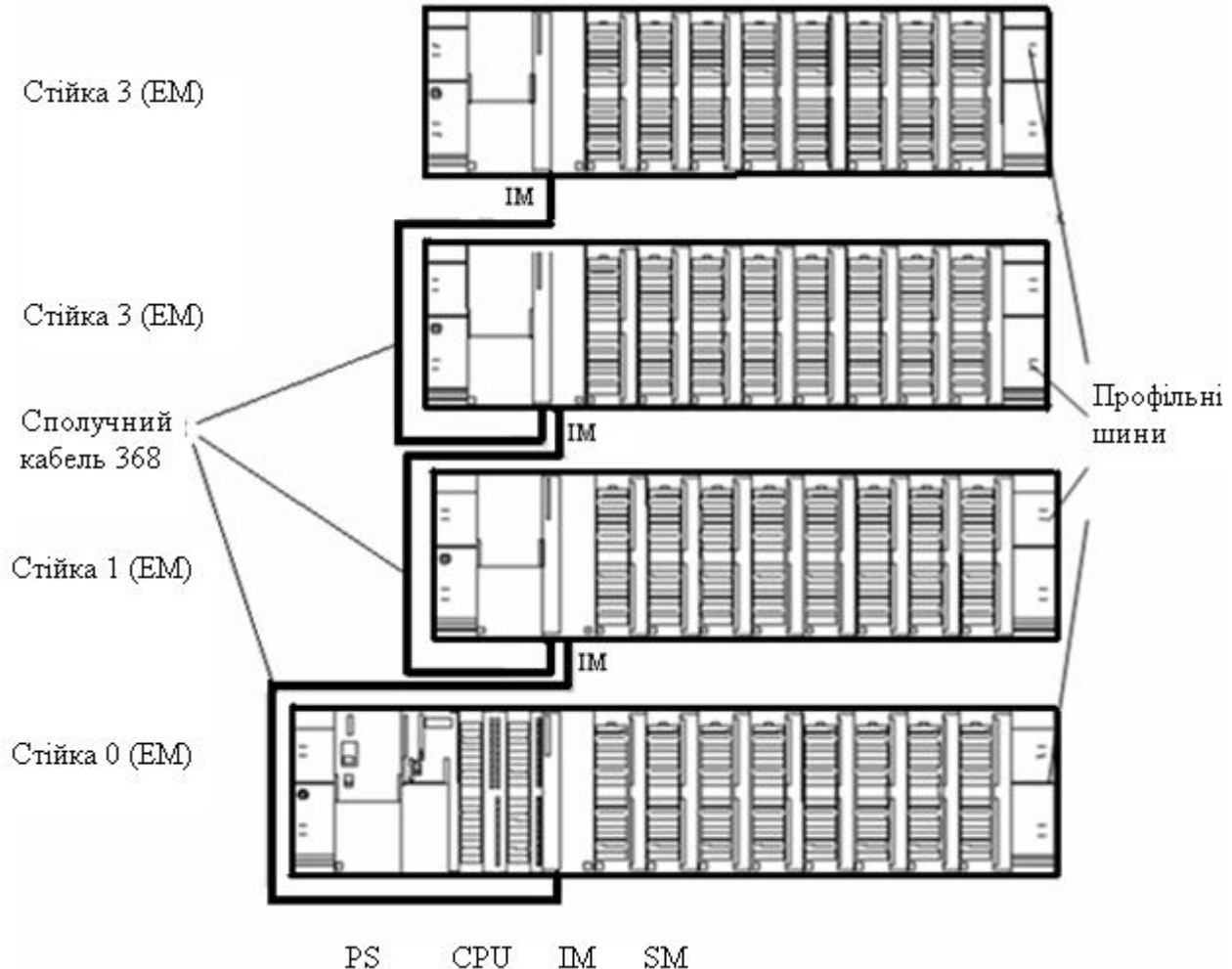


Рисунок 7.2 – Максимальна конфігурація S7-300

### 7.1.3 Центральні процесори

Залежно від ступеня складності задачі автоматичного управління, в контролерах SIMATIC S7-300 використовуються різні типи центральних процесорів, які відрізняються продуктивністю, об'ємом пам'яті, наявністю або відсутністю вбудованих входів-виходів і спеціальних функцій, кількістю і видом вбудованих комунікаційних інтерфейсів і т.ін. Всього пропонується сім типів центральних процесорів:

- CPU 312 IFM - для автоматизації невеликих установок з використанням або без використання аналогових входів - виходів;
- CPU 313 - для автоматизації установок з підвищеними вимогами до об'єму програми управління;
- CPU 314 - для автоматизації процесів, що пред'являють високі вимоги до об'єму програми управління і швидкості її обробки;

- CPU 314 IFM - компактний центральний процесорний модуль з вбудованими входами і виходами для автоматизації процесів, що пред'являють високі вимоги до об'єму програми управління і швидкості її обробки;

- CPU 315/315-2-DP, CPU 316-2DP, CPU 317-2DP і CPU 318-2DP- для розв'язання складних задач автоматизації з великим об'ємом програми управління і для побудови систем управління з децентралізованою структурою на базі PROFIBUS.

Центральні процесори нового покоління CPU 312, CPU 314, CPU 315/315-2DP відрізняються від аналогічних попередніх моделей збільшеними об'ємами робочої пам'яті (6 Кбайт у CPU 312 і 48 Кбайт у CPU 315/315-2DP), а також підвищеною кількістю входів-виходів, що обробляються, (аналогових 32 у CPU 312 і 128 у CPU 315/315-2DP, дискретних відповідно 128 і 1024). Вони працюють без буферної батареї, використовують як пам'ять, що завантажується, мікрокартку пам'яті MMC (Micro Memory Card) місткістю до 8 Мбайт. Крім того, MMC використовується для збереження даних при зникненні живлення CPU, а також зберігання символічної таблиці та всіх коментарів до проекту.

Наймогутнішими центральними процесорами сімейства S7-300 є - CPU 315/315-2DP, CPU 316-2DP, CPU 317-2DP і CPU 318-2DP. Ці процесори застосовуються для розв'язання задач, що пред'являють підвищені вимоги до об'єму програми, швидкості обробки даних і використання розвинених систем розподіленого вводу-виводу, побудованих на основі мережі PROFIBUS-DP. Всі CPU використовують для своєї роботи 5-ти вольтову картку пам'яті Flash-EEPROM. У CPU 318-2DP можуть використовуватися картки пам'яті RAM. Для збереження даних при зникненні живлення в CPU установлюється буферна батарея.

Наявність вбудованого інтерфейсу MPI (Multi Point Interface/ RS 485) в усіх типах CPU, дозволяє об'єднувати контролери в мережу, підключати до мережі пристрої людино-машинного інтерфейсу, виконувати дистанційне або локальне програмування і діагностування контролерів.

Зовнішній вигляд центрального процесору CPU 315-2DP такий саме як у CPU 316-2DP, CPU 317-2DP і CPU 318-2DP, рис.7.3.

Верхню половину передньої частини можна вважати лицевою панеллю центрального процесора. На ній знаходяться індикатори стану контролера, перемикач режимів роботи і гніздо для картки пам'яті. Перемикач режимів роботи та індикаторні елементи, а також їх призначення і функції однакові у всіх CPU контролера.

Індикатори SF, BATT, DC 5V, DC 5V, RUN, STOP, які створюють перший ряд світлодіодів, характеризують стан контролера, помилки програмування і параметризації, а індикатори SFDP і BUSF, що знаходяться у другому ряді світлодіодів, характеризують роботу CPU відповідно як Master-пристрою DP і як Slave-пристрою DP.

Перемикач режимів роботи CPU має чотири положення:

RUN-P (RUN-PROGRAM) - режим роботи, при якому CPU обробляє прикладну програму. Ключ у цьому положенні неможна витягти;

RUN - режим роботи, при якому CPU обробляє прикладну програму. Ключ у цьому положенні можна витягти, для того, щоб ніхто не зміг без дозволу змінити режим роботи;

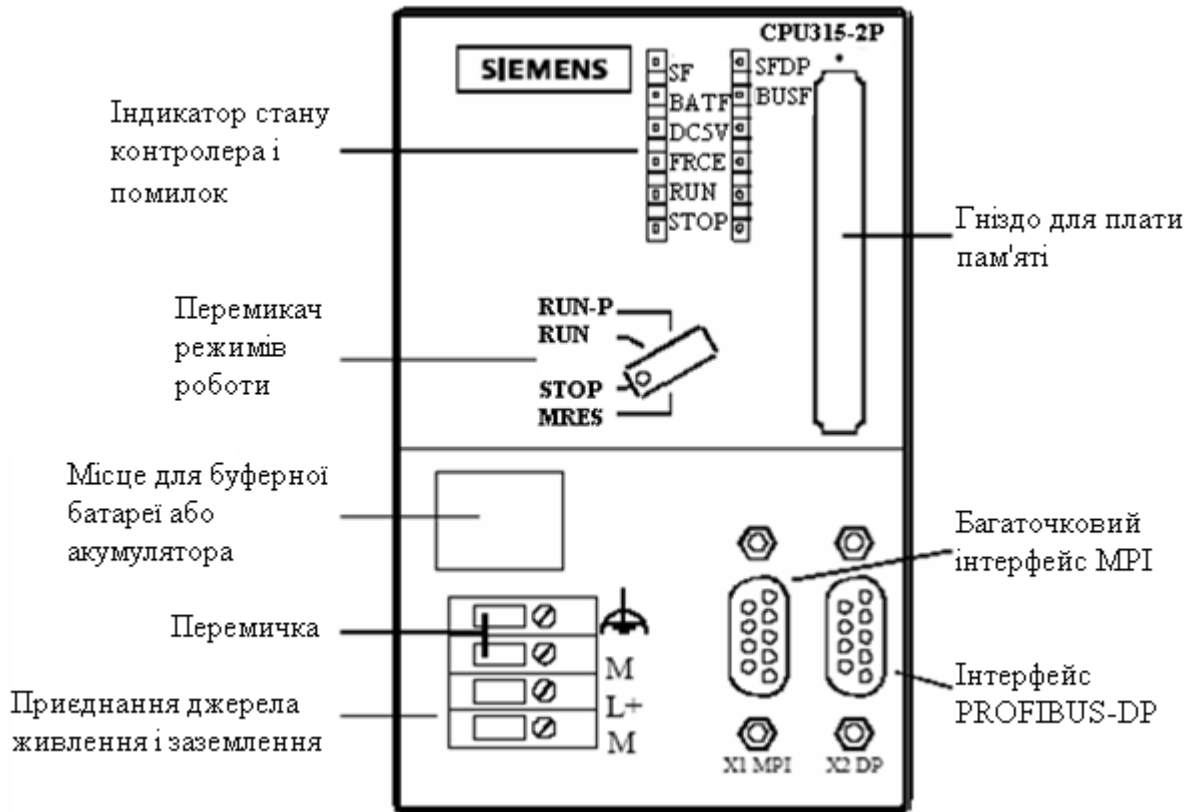


Рисунок 7.3 – Елементи CPU 315-2 DP

STOP - режим роботи, при якому CPU не обробляє прикладну програму. Ключ у цьому положенні можна витягти, для того, щоб ніхто не зміг без дозволу змінити режим роботи;

MRES - режим роботи для загального стирання CPU.

У нижній частині процесора під ступкою знаходяться місце для буферної батареї, різні багаточковий MPI і додатковий PROFIBUS-DP – інтерфейсів, а також клеми для підключення джерела живлення і заземлення.

Контролер за допомогою додаткового інтерфейсу можна використовувати у PROFIBUS-DP - мережі як DP-Master або як DP-Slave. Причому у якості DP-Master він може працювати не більше ніж з 64 DP-Slave пристроями.

CPU 315-2 DP має вбудованими 48 Кбайт робочої і 80 Кбайт завантажувальної пам'яті. У завантажувальній пам'яті розташовується програма користувача, а робоча пам'ять містить її дані та програмний код. Робоча пам'ять є високошвидкісною RAM і має випадковий доступ. Завантажувальна пам'ять може бути RAM або EEPROM. Якщо

завантажувальна пам'ять - RAM, то для зберігання програми користувача потрібна резервна батарея. Там, де завантажувальна пам'ять реалізована як інтегрована EEPROM, CPU може працювати без резервної батареї.

При необхідності завантажувальна пам'ять може бути розширена до 512 Кбайт за допомогою плати пам'яті, яка вставляється у спеціальне гніздо на лицевій панелі. Використовуються два типи плат - RAM-картка і EPROM-картка. Картка пам'яті RAM використовується тільки для розширення завантажувальної пам'яті. При відключенні ця картка втрачає свій вміст. Флешкартки EPROM використовуються коли необхідно захистити програму користувача від збоїв електроживлення, включаючи конфігураційні дані і параметри модулів[31].

#### 7.1.4 Блоки живлення

Для живлення програмованого контролера, датчиків і виконавчих пристроїв напругою 24В постійного струму, в S7-300 є декілька блоків живлення: PS 305(2 A), PS 307(2 A), PS 307(5 A), PS 307(10 A).

Основна відмінність їх полягає в характері вхідної напруги і величині вихідного струму. Так, блок живлення PS 305 (2 A) характеризується:

- підключенням до мережі живлення (номінальна вхідна напруга 24/48/72/96/110 В постійного струму);
- вихідним струмом 2 А;
- вихідною напругою 24В постійного струму; захистом від короткого замикання і обриву електричного кола;
- надійною гальванічною розв'язкою відповідно до стандарту EN 60 950;
- можливістю використання у якості джерела живлення навантаження.

А блоки живлення PS 307 (2 A),(5 A),(10A), рис.7.4, мають:

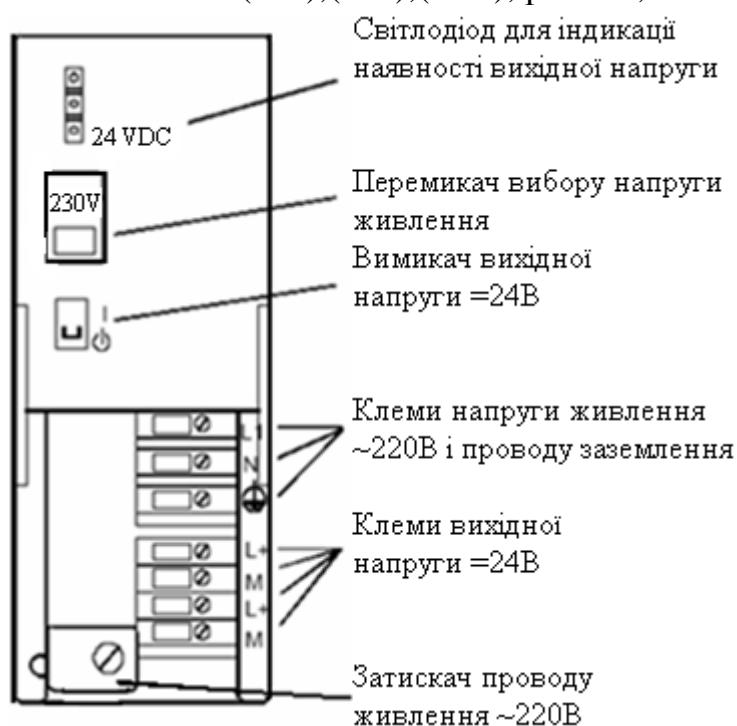


Рисунок 7.4 – Фронтальна частина блока живлення PS 307 (2 A)

- підключення до однофазної системи живлення (вхідна напруга 120/230В змінного струму, 50/60 Гц);
- вихідний струм відповідно 2 А, (5 А), (10А);
- вихідну напругу 24В постійного струму, захист від короткого замикання і обриву кола;
- надійну гальванічну розв'язку відповідно до стандарту EN 60 950, а також можуть бути використані як джерела живлення навантаження.

При виборі блока живлення і кількості розміщуваних модулів (SM, FM, CP) треба враховувати, що загальне споживання струму модулями не може перевищувати 1,2 А [31].

### 7.1.5 Сигнальні модулі

Для ПЛК S7-300 розроблена широка номенклатура сигнальних модулів, серед яких дискретні та аналогові модулі вводу-виводу. При виборі модулів для реалізації конкретної задачі керування об'єктом користувач повинен аналізувати їхні функціональні можливості, технічні характеристики і конструктивні особливості. Це дає можливість підібрати конфігурацію контролера, яка найбільш повно відповідає вимогам до системи керування, забезпечуючи при цьому оптимальну її вартість.

Аналізуючи вибір дискретного модуля за кількістю каналів, необхідно враховувати, що існує три типи модулів - модулі змішаних каналів вхід-виходів (SM 323), у яких на одному модулі присутні як вхідні, так і вихідні дискретні канали, модулі для приєднання винятково вхідних (SM 321) і модулі вихідних кіл (SM 322).

Дискретні модулі можуть працювати з різними типами дискретних сигналів, серед яких сигнали змінного (VAC) і постійного (VDC) струму, напругою  $\pm 24$  В VDC або 120/230 В VAC.

Вихідні дискретні канали можуть мати транзисторні або релейні виходи, які комутують сигнали змінної чи постійної напруги зі струмом комутації від 0,5 до 2 А.

У технічних характеристиках модулів вказується також сумарний струм виходів, гальванічна розв'язка між каналами і задньою шиною, допустима різниця потенціалів між різними колами струму.

Властивості деяких модулів можна встановлювати за допомогою параметризації. До таких модулів відносяться:

- дискретний модуль вводу SM 321; DI 16 x 24 VDC;
- дискретний модуль виводу SM 322; DO 8 x 24 VDC/0.5 А.

Параметризація дискретних модулів здійснюється програмно в пакеті STEP 7, коли центральний процесор CPU знаходиться в режимі STOP .

До параметрів, що встановлюються користувачем, відносяться затримка вводу, діагностика, деблокування апаратних переривань, деблокування діагностичних переривань, апаратне переривання при зростаючому фронті, переривання від процесу при спадаючому фронті.

Аналогові модулі відрізняються призначенням, кількістю каналів, характеристиками і діапазоном вимірювання. Так модулі групи SM 331 мають два або вісім тільки аналогових входів. Одні з них призначені для вимірювання температури, інші - температури та опору, треті напруги та струму, а четверті – усі види названих входних сигналів.

Модулі аналогового виходу групи SM 332 – двох або чотирьох каналні. Вихідні сигнали – напруга та струм.

Аналогові модулі групи SM 334 - модулі змішаних каналів. Вони мають чотири входних канали і два вихідних. Залежно від модифікації працюють з будь-якими видами входних сигналів, а на вихідних каналах формують напругу або напругу і струм.

До одного модуля можна підключити 4 термопари типів J, K, T, E, R, S, N. Модуль для термометрів опору допускає безпосереднє підключення двох датчиків температури однакового типу. Датчики можуть підключатися по 2-, 3- або 4-провідної схемі. Підключення датчиків до модуля здійснюється без використання проміжних підсилювачів.

Звичайно аналогові модулі мають за замовчуванням певні установки щодо виду і діапазону вимірювання. Проте за бажанням користувача вид і діапазон вимірювання каналів вводу-виводу можна встановити власноруч за допомогою модуля установки діапазону, а також пакету програмування STEP 7.

Аналогові модулі, в яких вид і діапазон вимірювання встановлюється не програмно, комплектуються відповідними модулями установлення діапазону. Ці модулі розташовуються у спеціальному гнізді на боковій стороні модуля вводу. Щоб змінити вид і діапазон вимірювання, необхідно модуль для установлення діапазону установити у відповідне положення – “A”, “B”, “C” або “D”. Ці положення вказані на його корпусі. Якщо потрібно переустановити модуль вибору діапазону вимірювання, його необхідно вилучити з аналогового модуля викруткою, вибрати нове положення і знову вставити в аналоговий модуль. Вибраним діапазоном є той, на який вказує маркіровочна точка на модулі вводу.

Час перетворення каналів аналогового вводу складається з основного часу перетворення і додаткового часу, який надається для вимірювання опору і контролю наявності пориву проводу.

Користуючись STEP 7, в деяких аналогових модулях вводу можна забезпечити згладжування аналогових сигналів. Заміряні значення згладжуються за допомогою цифрової фільтрації. Згладжування реалізується шляхом розрахунку модулем середніх значень з певної кількості перетворених аналогових значень. Користувач призначає параметри згладжування не більше ніж на чотирьох рівнях (відсутність згладжування, низьке, середнє, високе). Рівень визначає кількість аналогових сигналів, що використовуються для усереднювання. Чим вище вибраний рівень згладжування, тим стабільніше згладжене аналогове значення і тим більше вимагається часу для його отримання.

Усі сигнальні модулі мають адреси їх установлення на стійках. Вони потрібні для правильної адресації каналів сигнальних модулів в програмі користувача.

Адресація модулів є фіксованою і визначається місцем їх установлення. Тому кожному номеру слота відповідає певна початкова адреса модуля.

На рис.7.5 зображена конструкція S7-300 на 4 стійках і слоти з початковими адресами їх модулів. У модулях вводу-виводу адреси входів і виходів починаються з однієї і тієї ж початкової адреси модуля. Залежно від типу модуля адреса може бути дискретною або аналоговою. Адреса входу або виходу дискретного модуля складається з літери, адреси байта та адреси біта. Літера перед адресою вказує на походження сигналу. Якщо вхідний, то перед

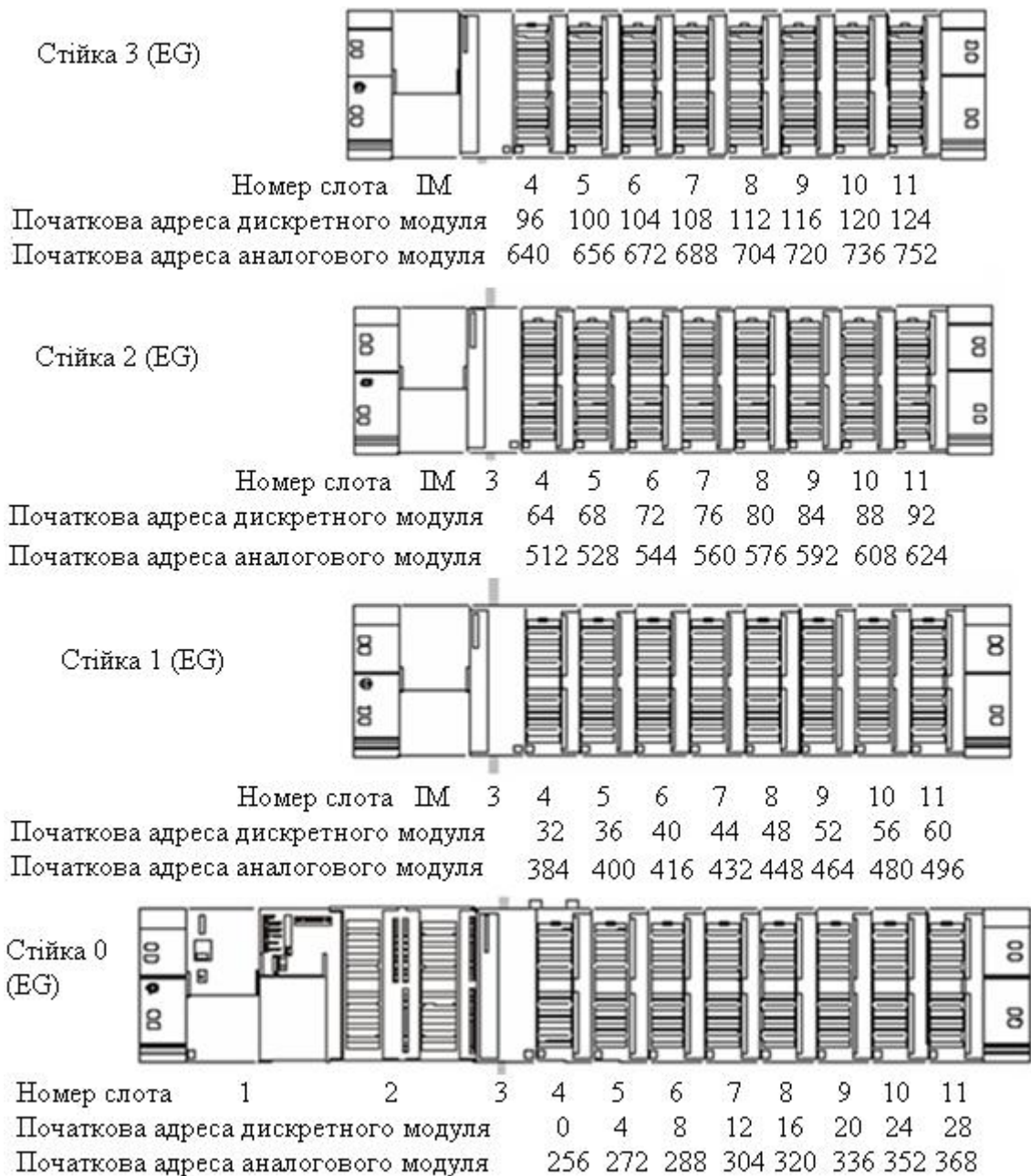


Рисунок 7.5 – Слоти S7-300 і відповідні початкові номери модулів

адресою вказується літера I, якщо вихідний - літера Q. Наступні за літерою цифри відображають адресу байта і адресу біта, наприклад, I1.2. Адреса байта зв'язана з початковою адресою модуля. Адреса біта зчитується з модуля.

Якщо перший дискретний модуль установлений у слоті 4, то за замовчуванням його початкова адреса дорівнює 0. Початкова адреса кожного наступного дискретного модуля збільшується від слоту до слоту на 4.

Адреси окремих каналів дискретного модуля формуються згідно зі схемою на рис.7.6.

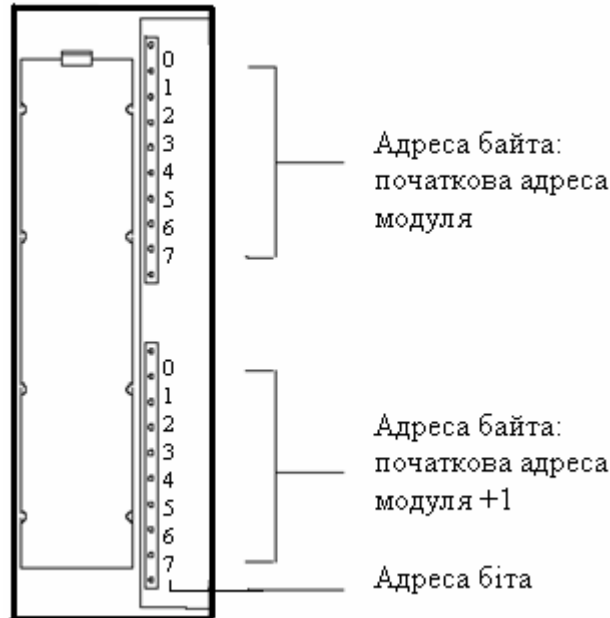


Рисунок 7.6 – Адреси входів і виходів дискретних модулів

Тобто, якщо дискретний модуль розміщується у слоті 4, коли початкова адреса його дорівнює 0, то за замовчуванням адреси каналів формуються відповідно до рис.7.7.

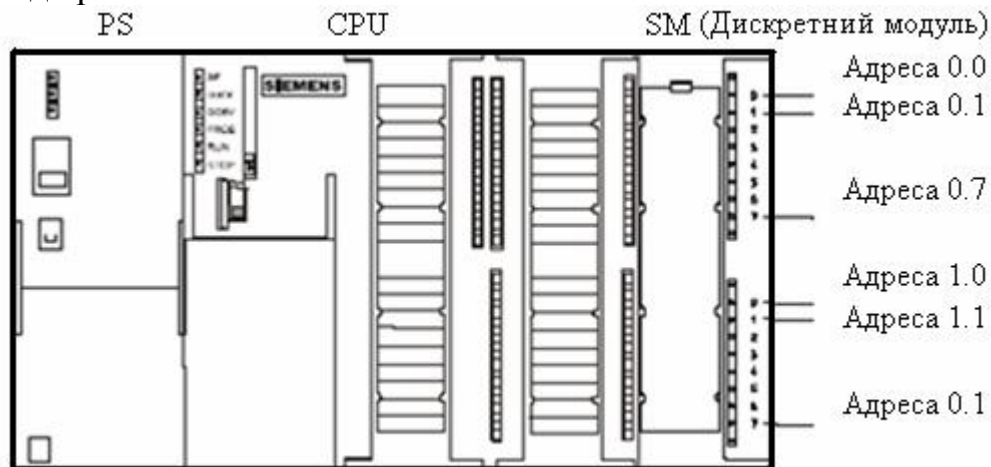


Рисунок 7.7 – Адреси каналів дискретного модуля

Тут номер слота 3 залишається вільним, оскільки в прикладі відсутній інтерфейсний модуль.



Адреса каналу аналогового входу або виходу завжди є адресою слова. Адреса каналу орієнтована на початкову адресу модуля. Якщо перший аналоговий модуль розташовується у слоті 4, то його початкова адреса за замовчуванням - 256. Початкова адреса кожного наступного аналогового модуля з кожним подальшим слотом підвищується на 16.

Аналоговий модуль має для аналогових каналів вводу і виводу однакові початкові адреси. На рис.7.8 показано які адреси отримує канал за замовчуванням у тому випадку, коли аналоговий модуль розташовується у слоті 4. Номер слота 3 також залишається незайнятим, оскільки в прикладі відсутній інтерфейсний модуль.

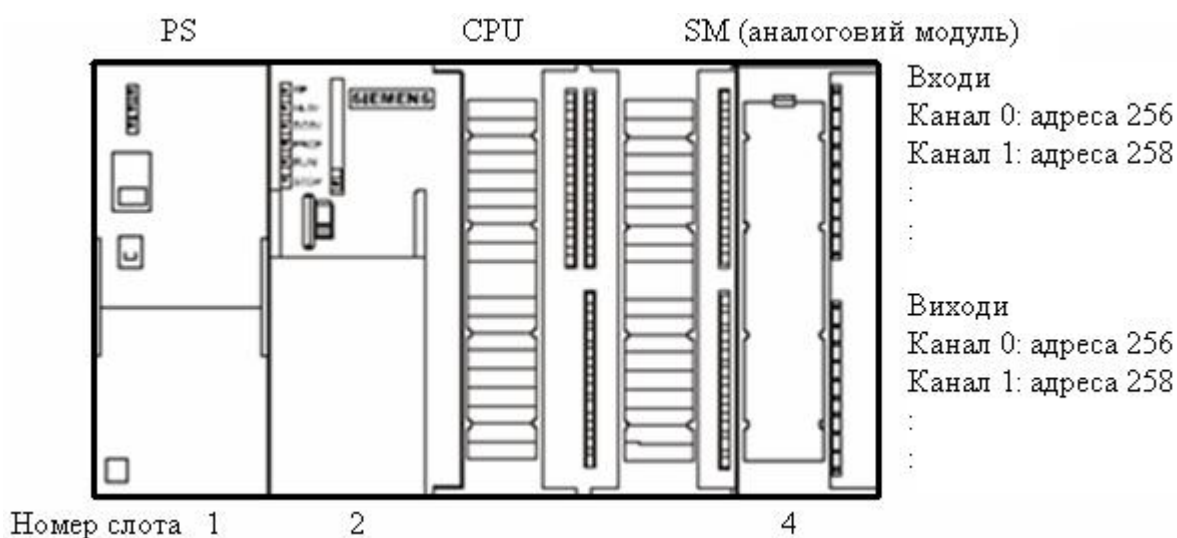


Рисунок 7.8 – Адреси входів і виходів аналогового модуля у слоті 4

Окрім адресації за замовчуванням, користувач може присвоїти кожному модулю (SM/FM/CP) адресу за власним бажанням (вільна адресація). Це призначення здійснюється в пакеті програмування STEP 7. При цьому задається початкова адреса модуля, на якому базуються всі наступні адреси цього модуля.

Адреси входів і виходів, які використовуються у програмі користувача, знаходяться у системній пам'яті (system memory) CPU. Ці адреси об'єднані в спеціальні області – так звані «образи процесу» модулів входу-виходу. У системній пам'яті також знаходяться таймери, лічильники, пам'ять меркерів, діагностичний буфер та тимчасові локальні дані.

Підключення зовнішніх кіл до дискретних і аналогових модулів здійснюється за допомогою компонентів "SIMATIC TOP connect" і відповідно "SIMATIC TOP connect TPA".

SIMATIC TOP connect і SIMATIC TOP connect TPA складаються з фронтштекера А з плоским стрічковим з'єднанням, одного або декількох клемних блоків В і одного або декількох сполучних кабелів Б з штепсельними з'єднувачами на кінцях, рис.7.9.

Використання для монтажу SIMATIC TOP connect і SIMATIC TOP connect TRA забезпечує швидке, дешеве підключення, простий монтаж компонентів (фронтштекер, сполучний кабель, клемний блок), значне скорочення помилок при підключенні, а також підключення джерела живлення не до модулів, а до компонентів SIMATIC TOP connect і SIMATIC TOP connect TRA.

Довжина сполучного плоского кабелю між SIMATIC S7 і клемними блоками повинна бути не більш 30 м. До обох кінців кабелю необхідно приєднати штепсельні розніми для підключення до фронтштекера і клемного блока. Фронтштекер необхідний для підключення сполучного кабелю до модуля.

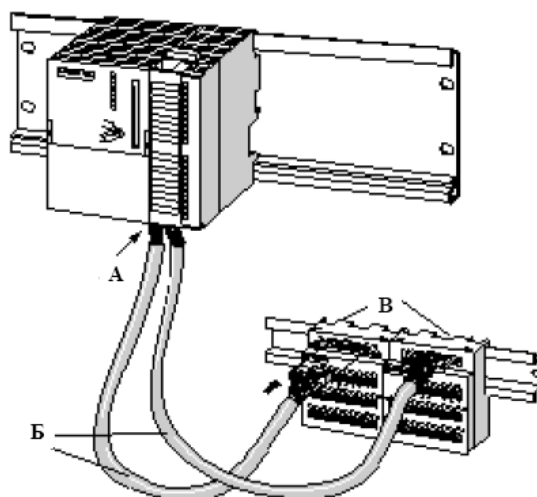


Рисунок 7.9 – Сполучення SIMATIC TOP connect на S7–300

Є 20-контактний і 40-контактний фронтштекер, кожен з гвинтовими або з пружинними контактами. 40-контактний фронтштекер потрібен для CPU 31xC і для 32-канальних сигнальних модулів.

Оскільки модулі S7-300 є відкритими пристроями, контролери SIMATIC S7 завжди монтуються тільки в кожухах, шафах або в апаратних приміщеннях, причому доступ до них повинен бути можливий тільки за наявності ключа або спеціальних інструментів [31].

### 7.1.6 Функціональні модулі

Функціональні модулі (FM) - це інтелектуальні модулі для розв'язання задач швидкісного підрахунку, позиціонування, автоматичного регулювання та інших.

Функціональні модулі забезпечені вбудованим мікропроцесором і здатні виконувати покладені на них задачі навіть у разі відмови центрального процесора ПЛК.

Функціональний модуль FM 355 – це універсальний 4-х канальний модуль автоматичного управління. Він має два варіанти виконання—С-регулятор безперервного управління з аналоговими виходами і S-регулятор покрокового або імпульсного управління з дискретними виходами. Модуль може працювати

в декількох режимах — автоматичному, ручному, відмовостійкому, слідкувальному, резервному, а також в режимі прямого дискретного управління.

Обидві модифікації FM 355 мають 4 аналогових і 8 дискретних входів. FM 355C має 4 аналогових виходи, а FM 355 S – 8 дискретних.

Функціональний модуль FM 350–1 – це швидкісний лічильник, який використовуються там, де необхідно рахувати сигнали, що поступають з великою частотою, і потрібна швидка реакція на заздалегідь заданий стан лічильника.

Обмін даними в межах одного ряду модулів S7–300 відбувається через шинний з'єднувач.

Перед монтажем FM 350–1 на профільній шині потрібно встановити в правильне положення кодуєчий штекер. Положення А кодуєчого штекера відповідає 5-ти вольтовому вхідному сигналу, а положення D – 24-вольтовому.

При використанні в контролері S7–300 модулів FM 355 і FM 350–1 потрібна їх програмна параметризація, яка здійснюється через екранні форми за допомогою інструментальної системи програмування STEP-7[31].

### **7.1.7 Система децентралізованої периферії SIMATIC ET 200**

Якщо об'єкт автоматизації знаходиться на значній відстані від контролера, то для отримання їм вхідних сигналів знадобляться проводи та кабелі великої довжини, а електромагнітні перешкоди можуть зменшити надійність системи. У таких випадках доцільно використовувати систему децентралізованої периферії SIMATIC ET 200, яка завдяки великій швидкості передачі даних забезпечує безперешкодний обмін інформацією між периферією і CPU контролера розташованого у центральному пункті.

Система децентралізованої периферії ET 200 складається з активних (Master) і пасивних (Slave) абонентів, зв'язаних польовою магістраллю PROFIBUS–DP. Master-пристроями в системі децентралізованої периферії можуть бути, наприклад, контролери S7–300 з центральним процесором CPU 315–2 DP або S7–400 з процесорами CPU 413–2 DP, 414–2 DP, 416–2 DP, а Slave-абонентом - станція розподіленого вводу-виводу ET 200M системи децентралізованої периферії SIMATIC ET 200.

Пристрій децентралізованої периферії ET 200M складається з наступних компонентів:

- профільної шини;
- джерела живлення (PS);
- веденого інтерфейсного модуля IM 153–1, IM 153–2 або IM 153–3
- до 8 сигнальних модулів (SM), функціональних модулів (FM) або комунікаційних процесорів (CP), причому FM і CP використовуються тільки з S7–DP–Master'ом.

Станція ET 200M може бути змонтована не більш, ніж на одному носії модулів (профільній шині), оскільки з'єднання через інтерфейсні модулі з іншими носіями модулів неприпустимо. Потужність джерела живлення

визначається струмом, що споживають модулі децентралізованої периферії і потужністю втрат всієї структури ET 200M.

SIEMENS пропонує широку гаму станцій розподіленого вводу-виводу SIMATIC ET 200, серед яких:

- станція ET 200S має в своєму складі інтерфейсний модуль IM151 і до 64 модулів різного призначення із загальним адресним простором вводу-виводу 128 байт. Інтерфейсний модуль може бути пасивним, виконуючим функцію веденого пристрою PROFIBUS-DP, або активним, який володіє можливостями програмованого логічного контролера. Великий спектр 2- і 4-канальних модулів дозволяє оптимальним чином адаптувати станції ET 200S для розв'язання широкого кола задач автоматичного управління. Всі модулі станції встановлюються в термінальні модулі. Термінальні модулі включають в свій склад всю розводку сполучних електричних мереж і монтуються на профільні шини 35x15 мм або 35x7.5 мм.

Заміна модулів вводу-виводу може здійснюватися без вимкнення живлення;

- станція ET 200X має ступінь захисту IP 65/67 і призначена для застосування поза шафами управління. Станція включає в свій склад один базовий і до 7 модулів розширення. Базовий модуль може бути пасивним, виконуючим функцію веденого пристрою PROFIBUS-DP, або активним, що додатково володіє можливостями програмованого логічного контролера для розв'язання широкого кола задач автоматичного управління;

- станція ET 200iS призначена для встановлення у вибухонебезпечних (Ex) зонах. Станція виконує функції стандартного веденого пристрою PROFIBUS-DP. Вона може вмонтовуватися у шафи управління, які розміщуються у вибухонебезпечних зонах класу 1 або 2, і допускає безпосереднє підключення датчиків і виконавчих пристроїв, що розташовуються в зоні класу 0. Кожна станція здатна обслуговувати до 128 дискретних або до 64 аналогових каналів вводу-виводу. Забезпечується підтримка HART-протоколу. Блок живлення і модулі вводу-виводу можуть бути замінені без вимкнення живлення станції (гаряча заміна). Ступінь захисту IP30[32].

### **7.1.8 Комунікації SIMATIC**

Комунікації (обмін даними між програмованими модулями) є вбудованим компонентом SIMATIC S7. Управління майже всіма комунікаційними функціями здійснює операційна система. Можна здійснювати обмін даними між двома CPU без будь-якого додаткового апаратного забезпечення за допомогою одного лише сполучного кабелю.

SIMATIC МЕРЕЖА (SIMATIC NET) – це апаратне з'єднання між комунікаційними вузлами. Мережа складається з однієї або більш ідентичних або різних підмереж, сполучених разом. У підмережі всі комунікаційні вузли (абоненти) сполучені за допомогою апаратного з'єднання з однорідними фізичними характеристиками і параметрами передачі, такими як швидкість передачі даних. SIMATIC- підмережа може працювати як MPI, PROFIBUS,

промисловий Ethernet (Industrial Ethernet) і з'єднання «точка-до-точці» (point-to-point – PtP).

Щоб усі абоненти могли підтримувати зв'язок один з одним, необхідно присвоїти їм адреси:

- у підмережі MPI - Адреса MPI;
- у підмережі PROFIBUS - Адреса PROFIBUS.

Ці адреси MPI/PROFIBUS можна встановити для кожного абонента окремо за допомогою програматора PG (у деяких підпорядкованих PROFIBUS DP – за допомогою перемикача на Slave-пристрої). При цьому у підмережі MPI може бути до 127 пристроїв, у підмережі PROFIBUS-DP – до 126, а у PtP – 2.

Для об'єднання в мережу окремих S7–300 потрібно з'єднати між собою їх CPU шинними кабелями SINEC L2. Контролери S7–300 приєднуються до мережі зв'язку через багатоточковий інтерфейс (MPI) CPU .

Відрізок шини між двома кінцевими резисторами (термінаторами) характеризується як сегмент. Сегмент може містити до 32 абонентів. У сегменті підмережі MPI можна реалізувати довжини кабелів до 50 м. У сегменті підмережі PROFIBUS довжина кабелів залежить від швидкості передачі.

Якщо необхідно реалізувати великі довжини кабелів, які перевищують допустиму довжину в одному сегменті, то треба використовувати повторювач RS 485. Повторювач RS 485 підсилює сигнали з даними в шинних кабелях і зв'язує між собою шинні сегменти.

Послідовно можна приєднати до 9 повторювачів RS 485.

Повторювач RS 485 має бути врахованим у загальній кількості всіх об'єднаних абонентів як абонент підмережі, хоча він і не одержує власної адреси MPI/PROFIBUS.

Шинний кабель PROFIBUS - це витий екранований мідний кабель. Він підключається до інтерфейсу MPI або PROFIBUS DP за допомогою шинного штекера. Якщо необхідно за допомогою польової шини охопити великі відстані незалежно від швидкості передачі, або обмін даними по шині не повинен піддаватися дії зовнішніх перешкод, то замість мідного кабелю використовується волоконно-оптичний кабель.

Максимальна довжина кабелю для сегмента має бути не більше 1000м при швидкості передачі даних від 9,6 до 187,5 КБод, не більше 400м при швидкості передачі даних 500 КБод, не більше 200м при швидкості передачі даних 1,5МБод і не більше 100м при швидкості передачі даних від 3 до 12 МБод. Якщо шина містить повторювач RS 485, то максимальна довжина кабелю між двома абонентами не повинна перевищувати 10000м при швидкості передачі від 9,6 до 187,5 КБод, 2000м при швидкості передачі від 1,5МБод і 1000 при швидкості передачі від 3 до 12 МБод.

Якщо необхідно налагодити обмін даними між станціями SIMATIC або пристроями інших виробників, то потрібні комунікаційні процесори.

Комунікаційні процесори (CP) – це інтелектуальні модулі, які виконують автономну обробку комунікаційних задач для промислових мереж AS-Interface, PROFIBUS, Industrial Ethernet і інтерфейс PtP.

Застосування драйверів для CP 341 дозволяє розширити комунікаційні можливості контролера підтримкою обміну даними в мережах MODBUS RTU і Data Highway. Для організації модемного зв'язку у складі програмованого контролера SIMATIC S7-300 можуть використовуватися комунікаційні модулі сімейства SINAUT ST7[31].

### 7.1.9 Програмне резервування

Для підвищення надійності в роботі системи за допомогою програмного забезпечення можна створити резервовану систему з головною станцією (master station), яка управляє процесом і резервною станцією (standby station), що приймає управління у разі збою головної станції.

В той же час відмовостійкість, яка досягається за допомогою програмного резервування, забезпечується тільки для повільних процесів, оскільки передача управління в резервну станцію відбувається протягом декількох секунд.

Більш відмовостійким є контролер SIMATIC S7-400H, який має резервовану конфігурацію, що включає дві центральні стійки з процесорами H-CPU і модулями синхронізації для порівняння даних. Передача інформації здійснюється через оптико-волоконний кабель. Обидва контролери працюють в режимі «гарячого резервування». У разі збою працюючий коректно контролер продовжує функціонувати самостійно.

Призначена для користувача програма розташована в двох областях пам'яті CPU, які називаються - завантажувальна пам'ять (load memory) і робоча пам'ять (work memory).

Завантажувальна пам'ять може бути інтегрована в CPU або в картку пам'яті, що підключається (plugin). Вся програма користувача, включаючи конфігураційні дані, розташовується у завантажувальній пам'яті.

Робоча пам'ять є високошвидкісною RAM (пам'ять з випадковим доступом), повністю вбудованою в CPU. Вона містить програмний код і призначені для користувача дані.

Програмуючий пристрій (програматор) переносить всю програму, включаючи дані конфігурації, в завантажувальну пам'ять. Потім операційна система CPU копіює «зв'язаний» програмний код і призначені для користувача дані в робочу пам'ять.

Якщо необхідно розширити завантажувальну пам'ять, використовують RAM - картку пам'яті. RAM-картка дозволяє модифікувати всю програму користувача в режимі онлайн (інтерактивному). Проте картки пам'яті RAM при відключенні втрачають свій вміст.

Якщо необхідно захистити програму, включаючи конфігураційні дані і параметри модулів, від збоїв електроживлення, використовують флеш-картки EPROM. В цьому випадку (у режимі офлайн) вся програма завантажується автономно на флеш-картку EPROM, підключену до пристрою програмування. Програму можна завантажити і в режимі онлайн, але для цього потрібно відповідні CPU з підключеною карткою пам'яті[32].

## 7.2 Програмне забезпечення SIMATIC

Для програмування і конфігурування програмованих контролерів SIMATIC S7-300/400 використовується пакет STEP 7 Professional. Цей пакет дозволяє отримати найвищу продуктивність при розробці комплексних проектів автоматизації.

Для створення програм у пакеті STEP 7 можна використовувати мови:

- список інструкцій (Statement List - STL);
- діаграми сходової логіки (Ladder Diagram - LAD);
- функціональних блоків (Function Block Control - FBD).

Більш того, для розв'язання спеціальних задач можуть бути використані додаткові мови програмування високого рівня і технологічно орієнтовані мови.

STEP 7 містить набір комфортабельних функцій для виконання всіх фаз розробки проекту:

- конфігурування і налаштування апаратури;
- установлення комунікаційних сполучень;
- програмування;
- тестування, введення в експлуатацію і обслуговування;
- документування проекту і архівування даних;
- діагностичні функції.

Всі функції підтримуються могутньою системою інтерактивної допомоги.

У файлах STEP 7 всі програми користувача і всі необхідні для їх виконання дані оформляються у вигляді блоків. Можливість використання одних блоків усередині інших дозволяє створювати чіткі і зрозумілі програмні структури, підвищувати наочність програми, полегшувати перенесення блоків з програми в програму[33].

### 7.2.1 Методика створення проекту у пакеті Step7



Розробити LAD-мовою проект автоматичного керування роботою дробарки. При натисненні кнопки ПУСК спрацьовує попереджувальна сирена. За 10с сирена вимикається і після 30 секундної 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.


### *Розробка проекту*

Використовуючи ярлик SIMATIC Manager пакету програмування STEP 7 V5.3, відкрити вікно майстра проекту – STEP 7 Wizard: “New Project”.

Натиснути кнопку Next і на 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 з двома полями. Зліва – вікно дерева проекту Drobarka, а справа поле з етикетками  Hardware і  CPU314(1).

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 таблиця (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 дискретні 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'являться і у конфігураційній таблиці з призначеними адресами.

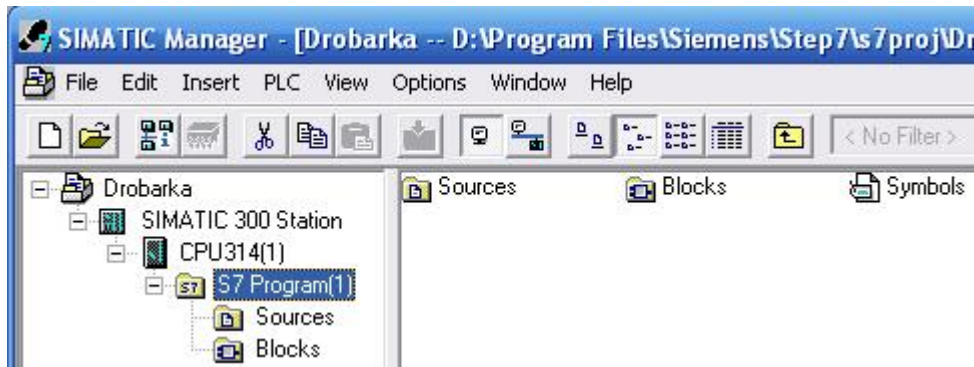
Слот третій, призначений для інтерфейсного модуля, залишається вільним:



| 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 321-1BH81-0AA0 |          |             | 0...1     |           |
| 5    | DO16xDC24V/0.5A | 6ES7 322-1BH81-0AA0 |          |             |           | 4...5     |

Після закриття додатку HW Config-SIMATIC на екрані ЕОМ 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 використовуються при створенні програми. Для цього двічі натиснути ліву клавішу миші на ярлику Symbols і у вікні, що відкривається, заповнити таблицю символів. При наданні змінним їх адрес дотримуватися таких позначень:

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

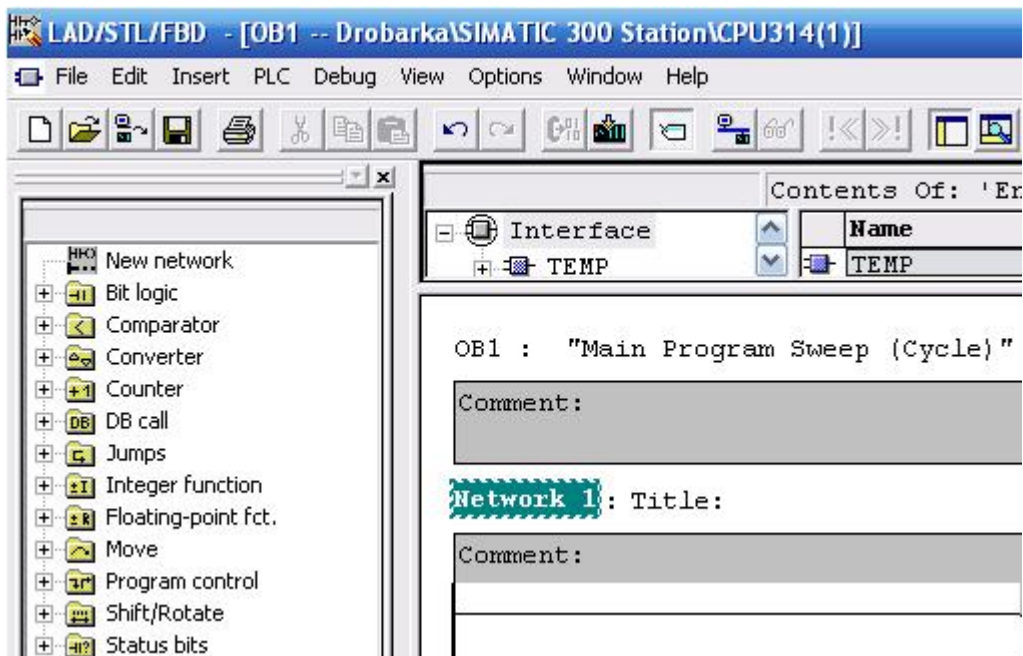
Таблиця Symbol Editor після заповнення її всіма змінними має вигляд:



| Symbol Editor - S7 Program(1) (Symbols)                  |         |           |  |
|--|---------|-----------|--|
| Symbol Table Edit Insert View Options Window Help        |         |           |  |
| n(1) (Symbols) -- Drobarka\SIMATIC 300 Station\CPU314(1) |         |           |  |
| 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 триволісті 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 триволісті першої попереджувальної сирени       |
| Signal2  | T 2     | TIMER     | Таймер триволісті 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'явиться, відкрити таблицю символів, натиснувши клавішу пробілу на клавіатурі EOM. Після дворазового натиснення лівої клавіші миші на вибраному імені, воно з'явиться над елементом мережі.

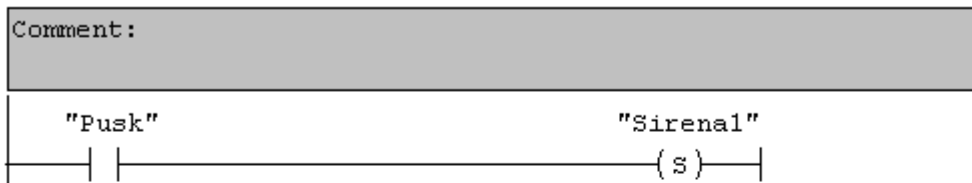
Для створення другої LAD-мережі (Network2), яка забезпечить тривалість попереджувального сигналу, потрібно в панелі інструментів натиснути ліву клавішу миші на кнопці . За аналогією з першою мережею встановити прямий контакт, R-котушку  і призначити їм ім'я Sirena1. Для встановлення паралельної S-котушки, 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 клавiшу миші на S-катушці у теці Bit\_logic бібліотеки New network і з таблиці символів призначити їй ім'я Pausa.

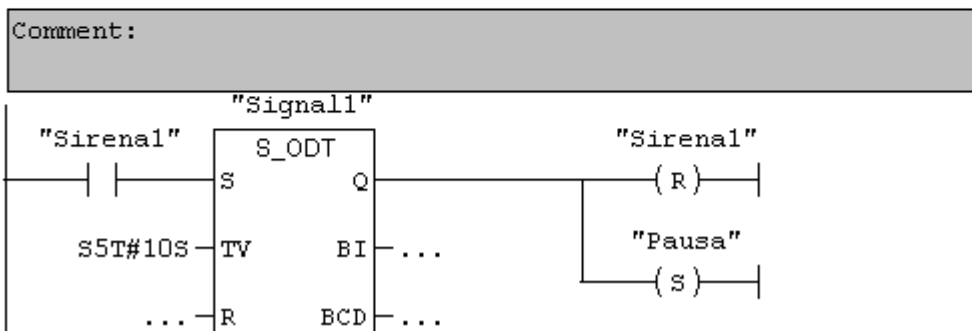
Щоб встановити таймер першої сирени в LAD-мережі (Network2) треба лівою клавiшею миші позначити логічну шину поміж контактом і точкою розгалуження логічної шини. В теці Timers бібліотеки New network подвійно клацнути по функціональному блоку S\_PULSE.

Після встановлення 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



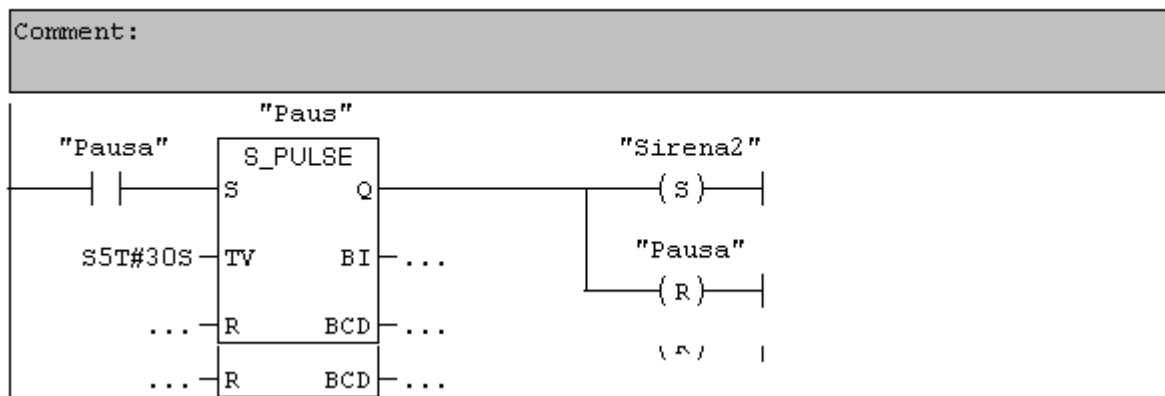
**Network 2** : Таймер тривалості першої попереджувальної сирени



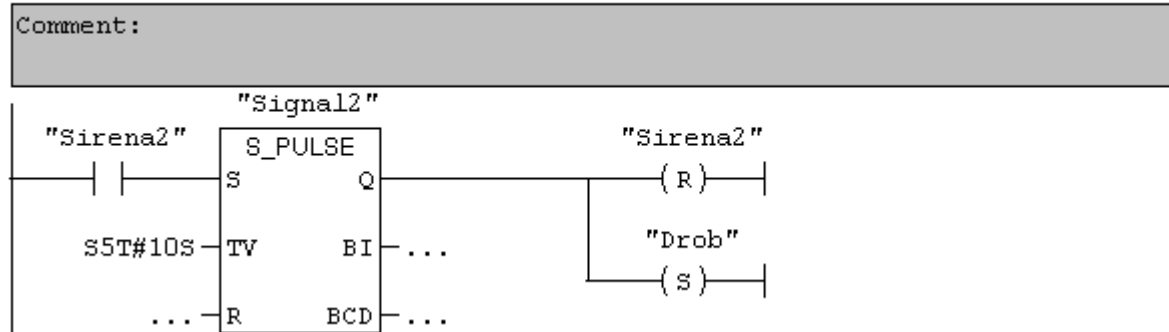
Наступні LAD-мережі, що забезпечують паузу поміж сиренами, тривалість другої попереджувальної сирени, вмикання і вимикання в роботу дробарки, світлової індикації, верхнього та нижнього конвеєрів, а також автоматичне регулювання рівнем руди у приймальному бункері, створюються за тою самою методикою, що і попередні LAD-мережі.

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

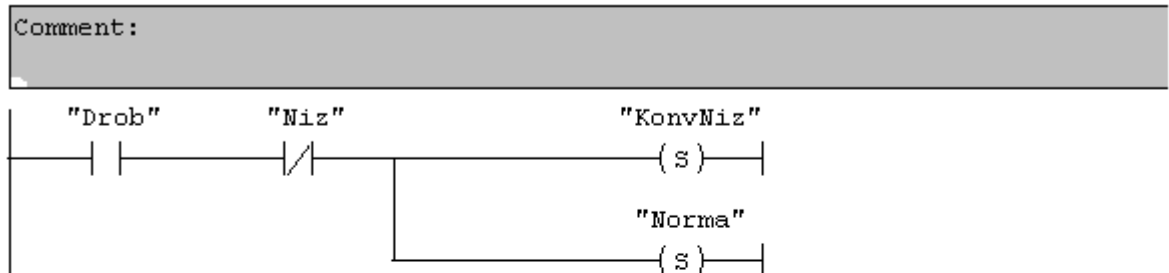
**Network 3** : Таймер паузи поміж сиренами



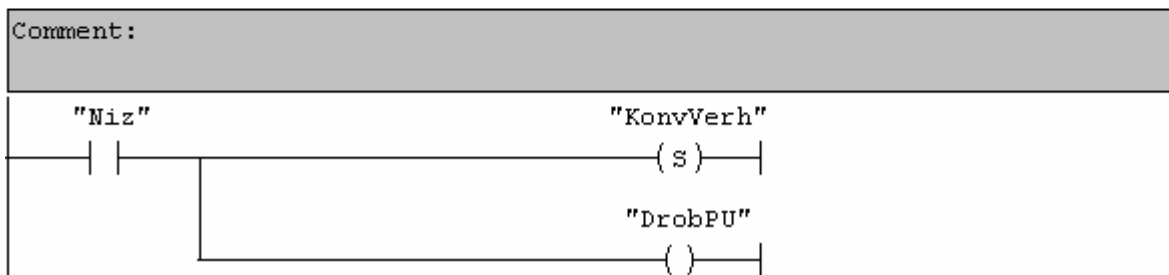
**Network 4 :** Таймер тривалості другої попереджувальної сирени



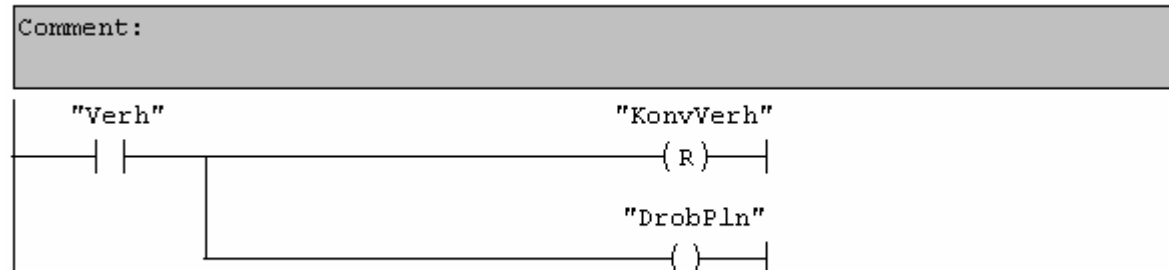
**Network 5 :** Увімкнення в роботу нижнього конвеєра і світової сигналізації



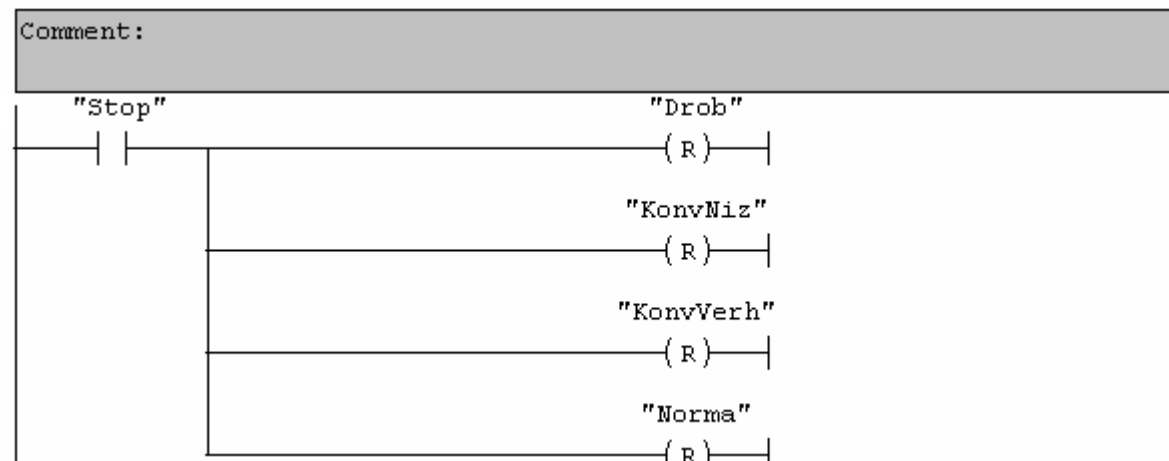
**Network 6 :** Увімкнення в роботу верхнього конвеєра







**Network 7 :** Вимкнення з роботи верхнього конвеєра

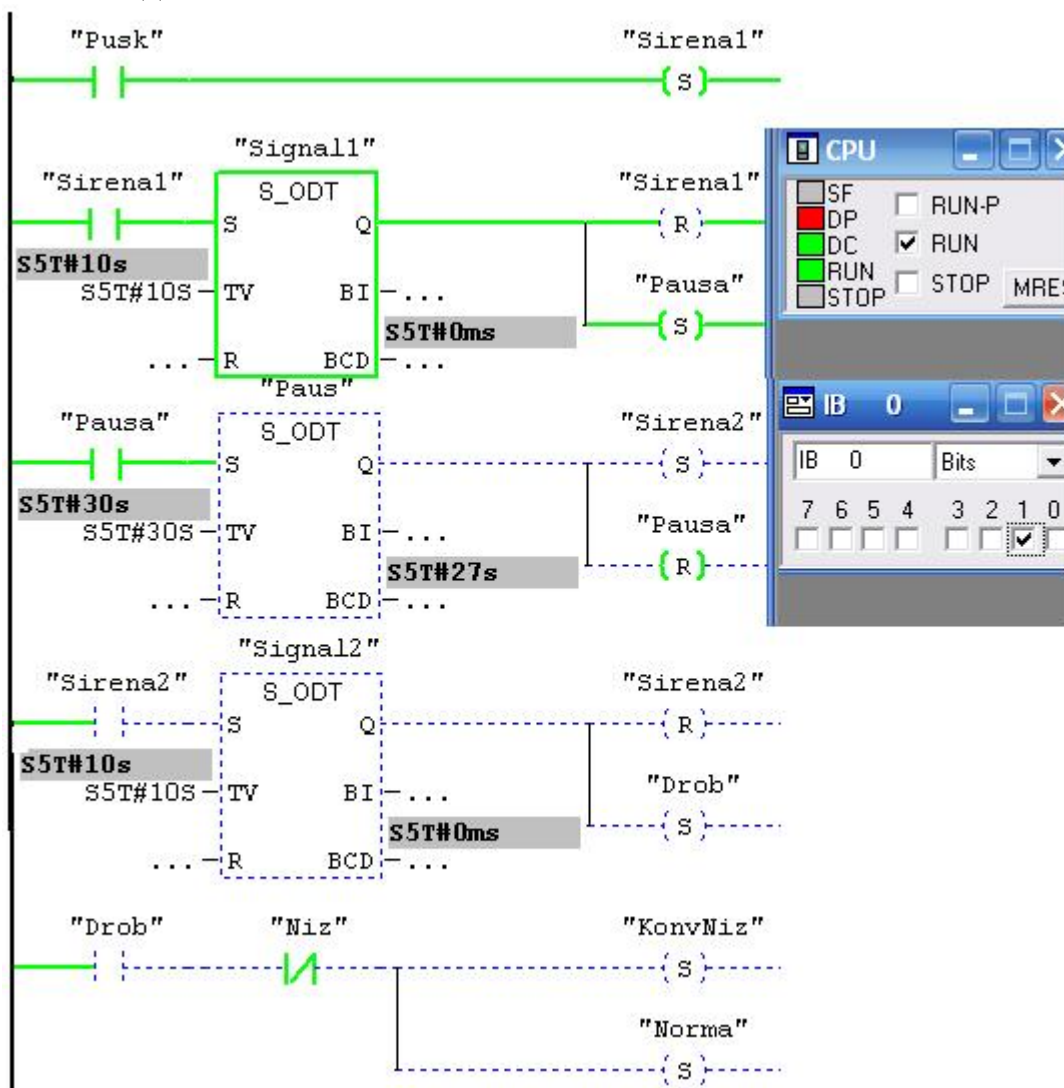


**Network 8 :** Вимкнення з роботи дробарки, конвеєрів і світової сигналізації



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[33].

Частина програми у режимі симуляції при активізованому біті змінної Pusk має вигляд:



## Тестові та контрольні запитання до розділу 7

**I-й рівень запитань та тверджень дозволяє перевірити загальний рівень засвоєння матеріалу розділу і передбачає відповідь на запитання або твердження “Так” чи “Ні” .**

1. SIMATIC – це назва торгової марки фірми Siemens та контролерів.
2. Контролер S7-300 має чотири типи центральних процесорів.
3. Контролер SIMATIC S7-300 має блокову структуру.
4. Чи усі типи CPU мають вбудований MPI-інтерфейс?
5. Чи є в контролері S7-300 можливість розширення пам'яті?
6. Блок живлення контролера S7-300 ідентифікується аббревіатурою SP.
7. У номенклатуру сигнальних модулів входять дискретні та аналогові модулі вводу-виводу.
8. Аналогові модулі вводу-виводу відрізняються між собою тільки кількістю каналів?
9. Для підключення зовнішніх кіл до дискретних модулів вводу-виводу призначено пристрій “SIMATIC TOP connect TRA”.
10. Функціональні модулі призначені для розв'язання задач швидкісного підрахунку, позиціонування та автоматичного регулювання.
11. Чи здатні функціональні модулі виконувати покладені на них задачі у разі відмови центрального процесора ПЛК?
12. Інтерфейсні модулі призначені для об'єднання базових блоків контролерів SIMATIC S7-300 у єдину мережу.
13. Довжина сполучного кабелю між парою інтерфейсних модулів IM 365 складає 10м.
14. Чи можна профільні шини в S7-300 застосовувати у якості стійки?
15. Модулі в стійках S7-300 сполучаються тільки за допомогою комунікаційної К-шини.
16. Чи можна у центральну стійку контролера S7-300 встановити 10 модулів?
17. Чи є позначення Q1.3 адресою входу дискретного модуля?
18. Комунікаційні процесори призначені для автономної обробки комунікаційних задач для промислових мереж AS-Interface, PROFIBUS, Industrial Ethernet та PtP.
19. Сегмент – це відрізок шини між двома кінцевими резисторами мережі контролерів.
20. Для створення програм у пакеті STEP 7 використовуються всі мови стандарту MEK 61131-1.

*Перевірте відповіді на сторінках: 1 - 166 с., 2 - 169 с., 3 - 166 с., 4 - 170 с., 5 - 172 с., 6 - 172 с., 7 - 173 с., 8 - 174 с., 9 - 177 с., 10 - 178 с., 11 - 178 с., 12 - 168 с., 13 – 168...169 с., 14 - 167 с., 15 - 168 с., 16 - 168 с., 17 – 175...176 с., 18 - 181 с., 19 - 181 с., 20 - 183 с.*

**II-й рівень запитань (тестові запитання) дозволяє перевірити рівень засвоєння матеріалу розділу із заглибленням у технічні особливості будови та структури контролерів SIMATIC та передбачає вибір правильної відповіді з наведених трьох варіантів.**

**1. Чим відрізняються процесори контролера S7-300?**

а) тільки видом інтерфейсу для організації мережі;  
б) продуктивністю та об'ємом пам'яті;  
в) продуктивністю, об'ємом пам'яті, наявністю вбудованих входів-виходів і спеціальних функцій, кількістю і видом вбудованих комунікаційних інтерфейсів.

**2. SM – це позначення...**

а) функціональних модулів;  
б) сигнальних модулів;  
в) інтерфейсних модулів.

**3. Які центральні процесори сімейства S7-300 є найпотужнішими?**

а) CPU 312, CPU 314, CPU 315/315-2DP;  
б) CPU 313 IFM та CPU 314 IFM,  
в) CPU 315/315-2DP, CPU 316-2DP і CPU 318-2DP.

**4. Що знаходиться на верхній половині лицевої панелі центрального процесора?**

а) індикатори стану контролера, перемикач режимів роботи і гніздо для картки пам'яті;  
б) тільки перемикач режимів роботи;  
в) тільки індикатори стану контролера.

**5. У чому основна відмінність блоків живлення контролерів S7-300 ?**

а) в характері вхідної напруги і величині вихідного струму;  
б) у кількості вихідних каналів живлення;  
в) тільки у характері вхідної напруги.

**6. З якими типами сигналів можуть працювати дискретні модулі?**

а) з сигналами “сухий контакт”;  
б) з сигнали змінного 120/230В і постійного  $\pm 24$ В струму;  
в) тільки з сигналами постійного  $\pm 24$ В струму.

**7. Чим комплектуються аналогові модулі, в яких вид і діапазон вимірювання встановлюється не програмно?**

а) DIP – перемикачами;  
б) чотирьохпозиційним перемикачем;  
в) модулями установлення діапазону вимірювання.

**8. З чого складається час перетворення каналів аналогового вводу?**

а) з основного часу перетворення, додаткового часу обробки модуля для вимірювання опору і контролю наявності пориву проводу;  
б) з основного часу перетворення та додаткового часу обробки модуля для вимірювання опору;  
в) з основного часу перетворення та контролю наявності пориву проводу.

**9. З чого складаються «SIMATIC TOP connect» і «SIMATIC TOP connect TRA»?**

- а) тільки з фронтштекера та одного або декількох клемних блоків;
- б) з фронтштекера, одного або декількох клемних блоків і одного або декількох сполучних кабелів з штепсельними з'єднувачами на кінцях;
- в) тільки з одного або декількох клемних блоків.

**10. Яка довжина сполучного плоского кабелю між SIMATIC S7 і клемними блоками?**

- а) не більше 50 м;
- б) не більше 30 м;
- в) не більше 20 м.

**11. Який функціональний модуль призначений для автоматичного управління?**

- а) функціональний модуль FM 350 –1;
- б) функціональний модуль FM 350 – 2;
- в) функціональні модулі FM 355-C та FM 355-S.

**12. Який інтерфейсний модуль забезпечує інтерфейс для нульової стійки S7-300?**

- а) IM 361;
- б) IM 365;
- в) IM 360.

**13. Через який кабель і якої довжини здійснюється передача даних з IM 360 в IM 361 або з IM 361 в IM 361?**

- а) через сполучний кабель 368 довжиною до 10м;
- б) через внутрішню шину довжиною до 5м;
- в) через кабель інтерфейсу RS-485 довжиною до 15м.

**14. В якій послідовності розміщуються модулі на профільній шині?**

- а) центральний процесор, функціональні модулі, сигнальні модулі, інтерфейсні модулі, комунікаційні модулі, блок живлення;
- б) блок живлення, центральний процесор, інтерфейсні модулі, сигнальні модулі, функціональні модулі, комунікаційні модулі;
- в) центральний процесор, функціональні модулі, сигнальні модулі, комунікаційні модулі, блок живлення, інтерфейсні модулі.

**15. Скільки стійок може мати багаторівнева конфігурація S7-300?**

- а) до чотирьох;
- б) до восьми;
- в) не більше двох.

**16. Яку початкову адресу мають дискретні і аналогові модулі вводу-виводу, що знаходяться на нульовій стійці?**

- а) 0 і 256 відповідно;
- б) адреси перших дискретних та аналогових модулів вводу-виводу дорівнюють 0;
- в) адреси перших дискретних та аналогових модулів вводу-виводу дорівнюють 256.



**17. Яке головне призначення шинного кабелю PROFIBUS?**

- а) для зв'язку програмованих контролерів з комп'ютерами;
- б) для зв'язку програмованих контролерів з первинними перетворювачами та датчиками;
- в) для об'єднання програмованих контролерів між собою у мережу.

**18. Чим визначається максимальна довжина кабелю для сегмента?**

- а) кількістю абонентів у сегменті;
- б) швидкістю передачі даних;
- в) об'ємом інформації, що передається по мережі.

**19. Як здійснюється параметризація дискретних модулів вводу-виводу сімейства S7-300?**

- а) програмно в пакеті STEP 7, коли центральний процесор CPU знаходиться в режимі STOP;
- б) за допомогою модулів установлення діапазону вимірювання;
- в) в контролерах S7-300 параметризація дискретних модулів вводу-виводу не виконується.

**20. Для створення програм у пакеті STEP 7 можна використовувати мови:**

- а) STL, LAD, FBD та мови високого рівня і технологічно орієнтовані;
- б) тільки STL, LAD, FBD;
- в) тільки мови стандарту MEK 61131-3.

*Перевірте відповіді на сторінках: 1 - 169 с., 2 - 167 с., 3 - 170 с., 4 - 170 с., 5 - 172 с., 6 - 173 с., 7 - 174 с., 8 - 174 с., 9 - 177 с., 10 - 178 с., 11 - 178 с., 12 - 168 с., 13 - 168 с., 14 - 168 с., 15 - 168 с., 16 - 175 с., 17 - 167 с., 18 - 181 с., 19 - 173 с., 20 - 183 с.*

**III-й рівень запитань (контрольні запитання) дозволяє перевірити рівень засвоєння матеріалу із систематизацією отриманої у розділі інформації та передбачає відповіді, що побудовані на логічному аналізі або порівнянні із приведенням конкретних технічних параметрів (даних або прикладів).**

1. Охарактеризуйте SIMATIC S7-300 як проектно-компонований виріб.
2. Чим відрізняються центральні процесори різних типів?
3. Перерахуйте технічні характеристики аналогових модулів вводу-виводу SIMATIC та їх параметри, що задаються програмно.
4. Як правильно підібрати функціональні модулі SIMATIC для задач автоматизації.
5. Розкрийте особливості компонування стійки контролера.
6. Скомпонуйте стійку SIMATIC S7-300, що включає: 2 модулі аналогового вводу, 2 модулі аналогового виводу, 2 модулі дискретного вводу-виводу. Визначте адреси аналогових та дискретних каналів.
7. Які мережні технології закладено у контролері SIMATIC S7-300.

8. Розкрийте призначення та характеристики інтерфейсів і протоколів, що підтримує контролер SIMATIC S7-300.

9. Як здійснюється створення проекту у програмному пакеті STEP 7?

10. Проведіть порівняльний аналіз створення проектів для контролера ADAM-5510KW у середовищі KW MULTIPROG та для контролера S7-300 у програмному пакеті STEP 7.

*Перевірте відповіді на сторінках: 1 – 166...179 с., 2 – 169...172 с., 3 – 173...178 с., 4 – 178...179 с., 5 – 167...168 с., 6 – 166...179 с., 7 – 180...182 с., 8 - 167, 180...182 с., 9 – 183...189 с., 10 – 128...140, 183...189 с.*

## 8 МІКРОПРОЦЕСОРНІ КОНТРОЛЕРИ MODICON TSX

Компанія Schneider Electric (SE) – це міжнародний концерн, який об'єднав провідних виробників засобів автоматизації США, Франції та Німеччини (Modicon, Telemecanique, Merlin Gerin, Square) і є одним з лідерів у галузі засобів промислової автоматизації.

Концерном Schneider Electric випускається велика гама ПЛК, які відрізняються кількістю портів і мережних інтерфейсів, розрядністю та швидкодією центрального процесора, об'ємом різних видів пам'яті. За кількістю входів - виходів ці контролери можна поділити на дві гілки, які, незважаючи на те, що вони об'єднані загальною ідеологією розвитку та модернізації і з кожним роком наближаються одна до одної, мають деякі особливості. До першої гілки можна віднести ПЛК Modicon TSX Nano, Modicon TSX Twido, Modicon TSX Micro, Modicon TSX Premium, Modicon TSX M340 а до другої - надвеликі контролери Modicon TSX Quantum.

Тривалий час при автоматизації складних агрегатів і виробничих процесів, керуванні великими інфраструктурними об'єктами контролери Modicon TSX Premium і Modicon TSX Quantum демонструють високу конкурентоздатність. Тому ці вироби компанії Schneider Electric отримали широке застосування на багатьох підприємствах всього світу.

### 8.1 Контролери Modicon TSX Quantum

Серія програмованих контролерів Modicon TSX Quantum, рис.8.1, є могутньою платформою для розв'язання будь-яких задач у галузі промислової



Рисунок 8.1 – Контролери Modicon TSX Quantum

автоматизації. Завдяки розширеній модульній архітектурі - від одиночного контролера до глобальної системи автоматизації, контролер Quantum може розв'язувати найвідповідальніші задачі на рівні цілого підприємства.

Контролери Quantum програмно, а також на мережному рівні сумісні з молодшими серіями контролерів -Compact і Momentum, що дозволяє будувати ще гнучкішу і ефективнішу архітектуру управління.

При розробці контролерів серії Modicon TSX Quantum збережена повна спадкоємність з попереднім сімейством контролерів Modicon 984, а також передбачено ряд удосконалень, що дозволяють покращити функціональні можливості і знизити загальну вартість прикладних систем автоматизації.

Системи управління, що використовують програмовані контролери серії Modicon TSX Quantum, підтримують різні варіанти рішень - від одиночної панелі вводу-виводу (до 448 вводів-виводів) до резервованих процесорів з розгалуженою системою вводів-виводів (до 64000 вводів-виводів). Цьому сприяє і об'єм пам'яті (від 256 Кбайт до 4 Мбайт) – достатній для реалізації найскладніших схем управління. Усі модулі серії Modicon TSX Quantum мають однакові розміри – 250×103,85×40,34мм, (В×Ш×Г).

В цих контролерах використовуються високопродуктивні математичні співпроцесори для забезпечення відповідної швидкості виконання алгоритмів і математичних обчислень, необхідних для реалізації якісного безперервного управління.

Можливість підключення до мереж підприємства і польових шин реалізована для восьми типів мереж від Ethernet до INTERBUS-S.

Усі модулі серії Modicon TSX Quantum мають однакові розміри – 250×103,85×40,34мм, (В×Ш×Г).

Quantum підтримує засоби розробки програмного забезпечення, відповідно до стандарту IEC 61131-3[34,35].

### **8.1.1 Монтажні панелі**

Монтажні панелі розроблені таким чином, щоб усі модулі, що використовуються в контролері, механічно фіксувалися на них і електрично сполучалися між собою. Монтажна панель містить пасивну схемну плату, яка дозволяє модулям через різні шини зв'язуватися один з одним і ідентифікувати номери їх слотів без додаткових призначень перемикачем. Випускаються панелі з 2, 3, 4, 6, 10 і 16 встановлювальними місцями. Кожна з них має свою модифікацію. Так 2-слотове шасі кваліфікується як 140 XBP 002 00, 4-слотове – як 140 XBP 004, а 16-слотове – як 140 XBP 016.

Для зменшення номенклатури запасних частин в усіх основних конфігураціях вводу-виводу (локального, віддаленого і розподіленого) використовуються однакові панелі. Вибір панелі зводиться до вибору моделі з достатнім числом місць для встановлення необхідної кількості модулів і місця для подальшого розширення. Монтажна панель на 10 слотів приведена на рис.8.2.

Програмовані контролери серії Modicon TSX Quantum використовують принцип загальної шини, тобто кожне місце в кожній панелі ідентичне за електричними можливостями. Модулі можуть бути вставлені в будь-який слот (місце) будь-якої монтажною панелі. Проте модулі джерел живлення

повинні встановлюватися в першій або останній слоти монтажної панелі, щоб забезпечити їм хорошу тепловіддачу. Для монтажу модуля на панелі необхідно видалити кришки з рознімів панелі, начепити модуль під кутом на два гачки, що розташовані зверху панелі, і притиснути його до розніму шини панелі, щоб забезпечити електричне підключення.

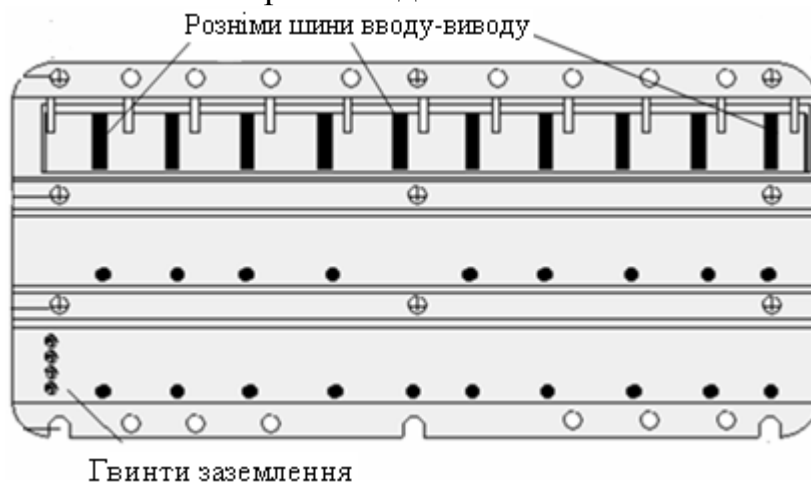


Рисунок 8.2 – Монтажна панель на 10 слотів 140 XBP 010

Після цього необхідно встановити рознім в модуль і затягнути кріпильні гвинти. Гаряча заміна модулів вводу-виводу може бути зроблена тільки при знятих рознімах зовнішніх підключень. Така конструкція модулів дозволяє легко вмонтовувати їх в спеціалізовані шафи управління NEMA.

Комунікації по шині, що вбудована в монтажну панель, здійснюються у трьох режимах. Для запуску і конфігурування активізується канал конфігурації, який дозволяє легко налагодити інтелектуальні модулі. Паралельно каналу конфігурації функціонують канал вводу-виводу даних і окремий канал зв'язку. Така організація роботи шини забезпечує високу загальну пропускну спроможність – 80 млн. біт/с. Враховуючи сучасні підвищені вимоги до комунікацій, така поліпшена архітектура забезпечує пріоритетне обслуговування модулів з мінімальним впливом на процес логічної обробки.

Для сполучення центрального шасі з шасі розширення, використовуються модулі розширення 140ХВЕ 100 00, які встановлюються на обох шасі, причому у будь-якому слоті. Зв'язок поміж модулями забезпечується кабелями розширення. Кабель 140ХСА 717 03 має довжину 1м, 140ХСА 717 06 – 2м, а 140ХСА 717 09–3м. [36].

### 8.1.2 Центральні процесори

Існує п'ять модифікацій процесорів - 140 CPU 113 02, 140 CPU 113 03, 140 CPU 213 04, 140 CPU 434 12 і 140 CPU 534 14, які відрізняються розмірами ОЗП (від 256 Кбайт до 4 Мбайт) і FLASH-ПЗП (від 256 Кбайт до 1 Мбайт), швидкістю обробки інформації на одному каналі (від 0,3 мс до 0,09 мс) кількістю регістрів доступних для користувача, струмом споживання та ін.

Центральний процесор Quantum обов'язково знаходиться на головній монтажній панелі вводу-виводу і як господар системної шини забезпечує управління локальним, віддаленим, і розподіленим вводом-виводом системи автоматизації.

Модуль CPU займає одне місце на панелі і містить системну пам'ять, пам'ять прикладних програм і порти зв'язку. Вся пам'ять вбудована, тому для конфігурування не потрібні додаткові мікросхеми або плати.

Для зберігання даних і інструкцій в системній пам'яті використовується технологія перезаписуваної постійної flash-пам'яті. Ця технологія дозволяє проводити модернізацію системи автоматизації, шляхом завантаження нової програми через порт Modicon Modbus або Modbus Plus замість заміни мікросхеми при електрично програмуваном ПЗП або плати. Flash-ПЗП містить програми, які визначають мову програмування додатку користувача, здійснюють керування статичною та динамічною пам'яттю, встановлюють порядок обробки та зберігання даних в ПЛК.

В CPU використовується статичне ОЗП з резервним підживленням від батареї, яка розташована в передній частині модуля і доступна для обслуговування під час роботи контролера. Пам'ять програми користувача призначена для зберігання таблиць конфігурації системи (режим роботи ПЛК, параметри комунікаційних портів, таблиці параметрів модулів вводу-виводу, що підтримуються центральним процесором і т.ін.), спеціальних функцій пакета Concept, а також сегментів і секцій проекту.

Усі входи-виходи ПЛК мають відповідне відображення у пам'яті даних користувача, яке організоване у вигляді окремих біт і 16-бітових регістрів.

Зовнішній вигляд центрального процесора приведений на рис.8.3.

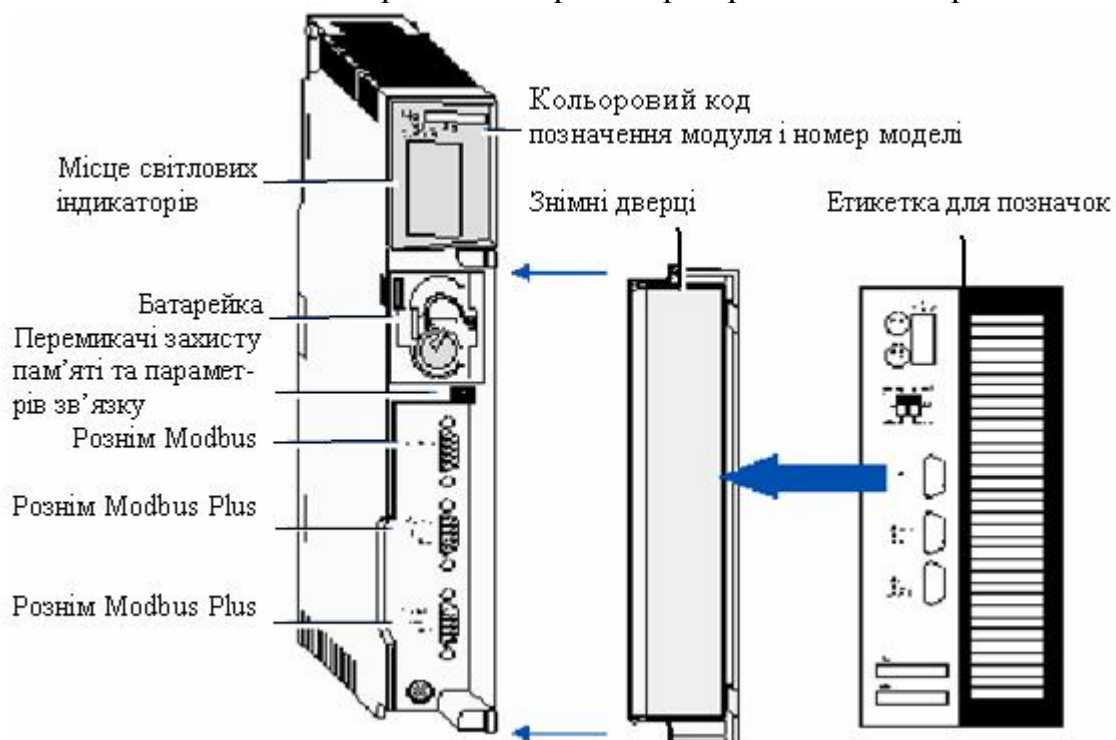


Рисунок 8.3 – Модуль центрального процесора

На передній панелі процесора безпосередньо під батареєю розташовані два трипозиційні перемикачі, положення яких відображається на світлодіодних індикаторах.

Зліва розташований перемикач захисту пам'яті. У верхньому положенні він забороняє перезапис програми користувача в процесорі з програматора або інших пристроїв.

Справа знаходиться перемикач параметрів зв'язку, який використовується для встановлення параметрів зв'язку через порт Modbus відповідно до вимог по швидкості передачі інформації і кількості біт даних.

На задній панелі процесора розташовані два кругові перемикачі, які використовуються для встановлення адреси вузла в мережі Modbus Plus (від 1 до 64) і для встановлення порту Modbus. Верхній перемикач (SW1) визначає десятки в значенні адреси, а нижній (SW2) - одиниці.

ЦПУ має стандартні порти зв'язку з підтримкою мережних протоколів Modbus і Modbus Plus. Додаткові порти зв'язку є на інших інтерфейсних модулях.

Світлодіодні індикатори на передній частині модуля, полегшують усунення несправностей, тому що відображають стан самого ЦПУ, і портів зв'язку.

Для реалізації додатків, що інтенсивно використовують математичні розрахунки, в моделях 140 CPU 213 04, 140 CPU 434 12 і 140 CPU 534 14 встановлено апаратний математичний співпроцесор. Використання цього пристрою значно знижує час виконання програм PCFL (бібліотека функцій управління процесами), а також програм, створених мовами МЕК Concept[34,35,36].

### **8.1.3 Джерела живлення**

Модулі живлення Quantum виконують дві функції – вони є джерелом живлення шасі системи і захищають її від перешкод і стрибків напруги.

Джерела живлення використовуються, для забезпечення стабілізованою напругою 5В усі модулі, що встановлені на монтажній панелі, включаючи модулі центрального процесора, модулі інтерфейсу для мережної передачі даних і модулі вводу-виводу.

В системах управління Quantum використовуються модулі живлення трьох типів – малопотужні автономні модулі живлення, нарощувані модулі живлення великої потужності і резервовані модулі живлення великої потужності.

Автономний модуль забезпечує тільки одне шасі Quantum стабілізованою напругою живлення 5В зі струмом навантаження 3А. Він використовується у системах, що не потребують значної потужності.

Нарощувані модулі забезпечують шасі струмом 8А. Ці модулі можуть працювати в автономному режимі і нарощуваному. Якщо на шасі встановлений один нарощуваний модуль він працює в автономному режимі і забезпечує шину струмом 8А. Якщо встановлені два модулі, то вони автоматично

перемикаються у нарощуваний режим, забезпечуючи шасі живленням стабілізованою напругою 5В зі струмом 16А. Нарощуваний режим застосовується для конфігурацій, які споживають більше, ніж номінальний струм одного джерела. В цьому випадку на одній монтажній панелі можуть бути встановлені два джерела живлення.

Резервованій модуль, забезпечує шасі струмом 8А. В системах підвищеної надійності два резервованих модуля забезпечують шасі струмом 8А. Якщо один модуль виходить з ладу, інший забезпечує необхідне живлення шасі струмом 8А.

Модулі 140 CPS 111 00 (автономний) і 140 CPS 114 00 (нарощуваний) мають вхідну напругу ~115-230В і забезпечують шину живленням 5В при струмі навантаження відповідно – 3А і 8А, а джерела 140 CPS 211 00 (автономний) і 140 CPS 214 00 (нарощуваний) – живляться напругою =24В при тих самих струмах навантаження. До резервованих відносяться модулі 140 CPS 114 00 з напругою живлення ~115-230В і 140 CPS 224 00 з напругою живлення =24В. Усього існує 10 модифікацій модулів живлення. Серед них – по три модифікації автономного і нарощуваного типу і чотири – резервованого [34,35,36].

#### **8.1.4 Модулі вводу-виводу**

Серія програмованих контролерів Modicon TSX Quantum підтримує повний спектр високопродуктивних модулів вводу-виводу, які розроблені для взаємодії з широким колом периферійних пристроїв.

Серед модулів дискретного вводу серій 140 DAI і 140 DDI є 16-ти і 32-х каналні на струм споживання - 180, 200, 225, 250 і 330 мА при напрузі живлення =24В, 60В, ~115В, 230В. Серед модулів дискретного виводу серій 140 DAO і 140 DDO і 140 DRA є модулі на 8,16 і 32 канали зі струмом навантаження від 100 до 560 мА при напрузі живлення =24В, 60В, або ~24В, 115В, 230В. Модулі серії 140 DRA на виході мають нормально відкриті або нормально закриті контакти реле.

Модулі аналогового вводу серій 140 ACI призначені для роботи зі струмовим сигналом 4-20 мА або напругою 1-5В, серії 140 AVI – зі струмовим сигналом 0-20мА або напругою ±10В, серії 140 ARI – з платиновим термометром опору, а серії 140 ATI з термопарами. Всі модулі 8-каналні.

Серед 4-х каналних модулів аналогового виводу модуль 140 ACO 030 00 має вихідний сигнал 4-20 мА, а модуль 140 AVO 030 00 може забезпечити вихідний сигнал по напрузі в інтервалах 0-10; ±10В; 0-5В і ±5В.

Існують також комбіновані модулі 140 DAM 590 001, 140 DDM 390 001(дискретні) і 140 АММ 090 001(аналогові), які об'єднують в певній пропорції канали вводу і канали виводу.

Для підвищення надійності і працездатності системи автоматизації передбачена можливість "гарячої заміни" модулів вводу-виводу, коли витягання або встановлення модулів в монтажні плати під напругою відбувається без створення перешкод для інших робочих елементів систем



управління. При порушенні зв'язку всі модулі можуть бути налаштовані на перехід в один з трьох аварійних станів - відключення всіх виводів, утримання останнього значення, перехід в передумовлений стан.

Кожен модуль не залежно від типу може розташовуватися на будь-якому місці монтажної панелі. Адресація їх і параметри настройки визначаються за допомогою програмного забезпечення Modicon. Після закінчення настройки програмне забезпечення системи здатне знайти відсутність або несправність модуля і передати повідомлення про це контролеру. Усуненню несправностей також допомагає світлодіодна індикація, що є у кожного модуля.

Зовнішній вигляд модуля вводу-виводу приведено на рис.8.4, а опис світлових індикаторів для аналогових модулів вводу-виводу і дискретних модулів на 16 каналів приведено у табл.8.1.

На правій і лівій сторонах клемного розніму і відповідного йому гнізда в корпусі модуля знаходяться виїмки для кодування вставними ключами.

Таблиця 8.1 Опис світлових індикаторів

| Індикатор                     | Колір    | Свічення                             |
|-------------------------------|----------|--------------------------------------|
| Active                        | Зелений  | Модуль підключений до шини           |
| F                             | Червоний | Знайдена несправність (поза модулем) |
| 1 ... 16 (2 лівих стовбчика)  | Зелений  | Активність вузла або каналу          |
| 1 ... 16 (2 правих стовбчика) | Червоний | Несправність вузла або каналу        |

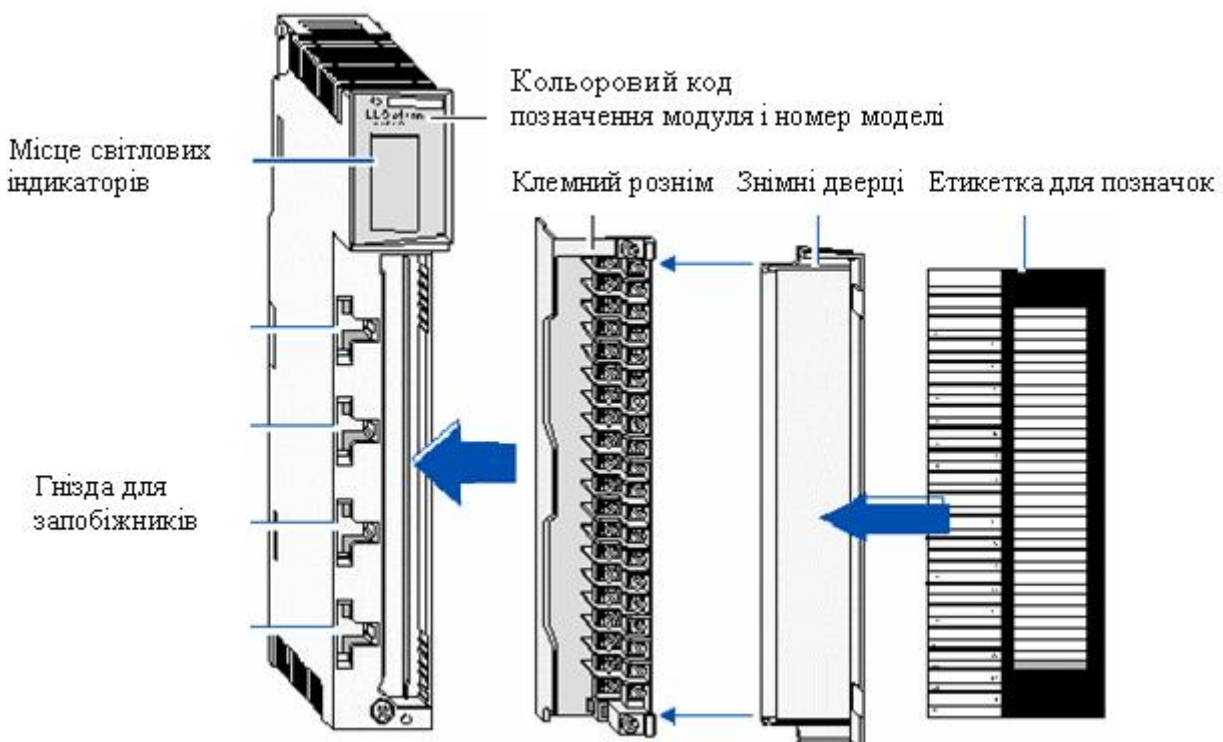


Рисунок 8.4 – Модуль вводу-виводу

Мета кодування полягає в тому, щоб після підключення до розніму проводів усунути можливість помилкового установлення його не на той модуль.

Для можливості кодування всі розніми, що призначені для модулів вводу-виводу, поставляються з 12 первинними ключами (по 6 жовтих ключів для модуля і для розніму) і з 6 вторинними ключами (по 3 білі ключі для модуля і для розніму).

Первинні ключі розташовуються з правого боку модуля і позначаються буквами А - F (верхні і нижні позиції кодуються однаково). Ці ключі забезпечують кодування класу модуля. Первинні коди заздалегідь визначені Modicon. У табл.8.2 наведено фрагмент кодування первинними ключами модуля і розніму з вказівкою розташування ключів.

Таблиця 8.2 Кодування первинними ключами

| Клас модуля                | Позначення модуля   | Кодування модуля | Кодування розніму |
|----------------------------|---|------------------|-------------------|
| <b>=24В пост.напр.</b>     | <b>140 DDI 353 00</b><br><b>140 DDO 353 00</b>                          | <b>ABE</b>       | <b>CDF</b>        |
| <b>=10-60В пост. напр.</b> | <b>140 DDI 841 00</b><br><b>140 DDI 853 00</b><br><b>140 DDO 843 00</b> | <b>ABF</b>       | <b>CDE</b>        |

Вторинні ключі розташовуються з лівого боку модуля і позначаються цифрами 1- 6. Вторинні ключі визначаються користувачем і можуть використовуватися для того, щоб ідентифікувати розміщення модуля на монтажній панелі. Для забезпечення користувача можливістю кодування вторинними ключами (для попередження плутанини рознімів в ідентичних модулях) в рознімі і в модулі передбачено 17 гнізд. Для визначення унікального для кожної позиції коду модуля і коду розніму рекомендується користуватися спеціальною таблицею, фрагмент якої наведено у табл.8.3

Таблиця 8.3 Кодування вторинними ключами

| Позиції на панелі | Кодування модуля | Кодування розніму |
|-------------------|------------------|-------------------|
| <b>1</b>          | <b>123</b>       | <b>456</b>        |
| <b>2</b>          | <b>124</b>       | <b>356</b>        |
| <b>3</b>          | <b>125</b>       | <b>346</b>        |
| <b>4</b>          | <b>126</b>       | <b>345</b>        |
| <b>5</b>          | <b>134</b>       | <b>256</b>        |
| <b>6</b>          | <b>135</b>       | <b>246</b>        |

На рис.8.5 показане розташування первинних і вторинних ключів в рознімі, яке відображає рекомендоване кодування для модуля на =24В в слоті №6 монтажної панелі.

Для монтажу модулів розроблена система Modicon CableFast, яка об'єднує рознім модуля вводу-виводу і клемну колодку за допомогою системного кабелю. Системний кабель заздалегідь розведений на обох кінцях і випускається різної довжини. Для додатків, що потребують кабель спеціальної довжини або спеціальну клемну колодку, випускаються додаткові кабелі з відкритою кіскою. Система Modicon CableFast дозволяє знизити витрати на прокладку проводів в щиті майже на 50%. При використанні системи CableFast відпадає необхідність продзвонювання проводів, які сполучають модуль вводу-виводу і клемну колодку.

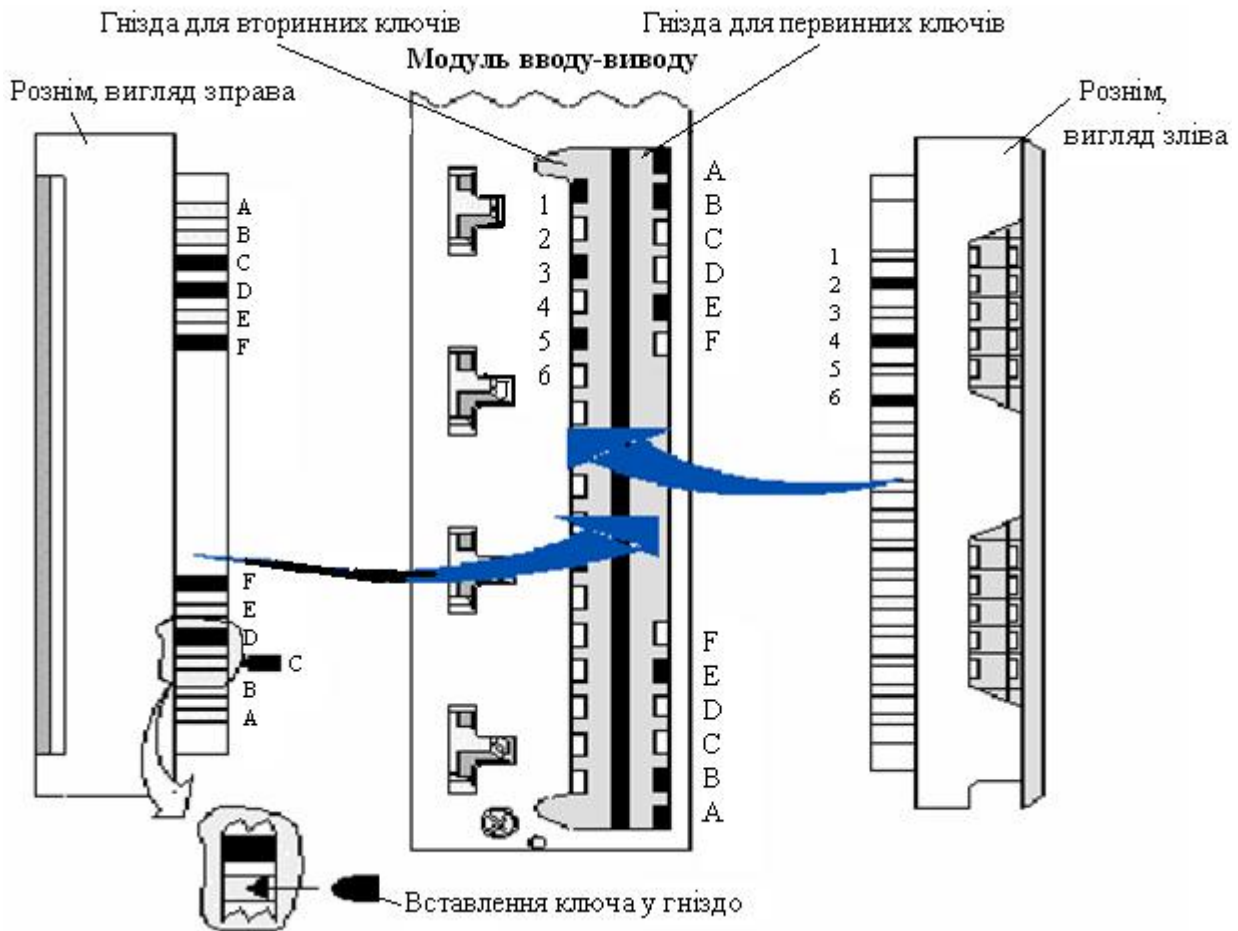


Рисунок 8.5 – Розташування первинних і вторинних ключів в рознімі

Унікальною новиною у системі CableFast є двопровідний електричний інтерфейс для периферійних пристроїв з клемною колодкою. Такий інтерфейс дозволяє приєднувати тільки один провід до кожної клеми, зменшуючи кількість підключень[34,35,36].

### 8.1.5 Конфігурації вводу-виводу

Modicon TSX Quantum може реалізувати одну або більшу кількість наступних конфігурацій:

- локальний ввід-вивід;

- віддалений ввід-вивід;
- розподілений ввід-вивід.

Ці конфігурації можуть компонуватися комбінацією центральних процесорів Quantum, джерел живлення, модулів мережних інтерфейсів і модулів вводу-виводу.

#### **8.1.5.1 Конфігурація локального вводу-виводу**

Програмовані контролери серії Modicon TSX Quantum мають конфігурацію локального вводу-виводу, коли найефективнішим способом сполучення є прокладка кабелю з периферії до центральної шафи управління. Конфігурація локального вводу-виводу може здійснюватися модулями вводу-виводу в кількості від 1 до 14 разом з центральним процесором і джерелом живлення, зібраними на монтажній панелі з 16 слотами. Така архітектура дозволяє мати до 448 дискретних каналів вводу-виводу.

У разі потреби, в панелі локального вводу-виводу можуть встановлюватися додаткові системні модулі. Для їх використання необхідна наявність локального центрального процесорного пристрою (ЦПП), за винятком блока управління багатокординатним переміщенням, який може працювати автономно.

Як додаткові системні модулі можуть використовуватися різні модулі мережного інтерфейсу, які складаються з процесорів віддаленого вводу-виводу (RIO) (не більш одного) або модулів розширення мережі Modbus Plus – NOM (не більше двох), модуль управління багатокординатним переміщенням, а також контролер гарячого резерву.

Типова система локального вводу-виводу містить PS-джерело живлення, CPU (ЦПП), DI- дискретний ввід, DO- дискретний вивід, AI- аналоговий ввід, AO- аналоговий вивід, Spare- вільне встановлювальне місце. [35,36].

#### **8.1.5.2 Конфігурація віддаленого вводу-виводу (RIO)**

Якщо програмний додаток вимагає великої кількості каналів вводу-виводу або необхідно підключити контролер до вже існуючих віддалених вузлів вводу-виводу виробництва Modicon, система Quantum здатна реалізувати конфігурацію віддаленого вводу-виводу (RIO). Для цього використовуються модулі інтерфейсного обміну даними, які встановлюються на платі центрального процесора і на платі віддаленого вводу-виводу RIO. Модулі RIO 140 CRP 931 00 і 140 CRP 932 00 з одинарним і здвоєним каналом відповідно, які встановлюються в ту саму монтажну панель, що і модуль центрального процесора - утворюють головний (місцевий) канал RIO і називаються процесорами. Ці ж модулі, що встановлені на віддалених панелях – утворюють підканали RIO.

Модуль (процесор) віддаленого вводу-виводу приведений на рис.8.6. Для двонаправленої передачі даних між головним каналом і підканалами на лицевій панелі модуля існує рознім, до якого приєднується коаксіальний кабель завдовжки до 5250 м. Максимальна кількість вузлів віддаленого вводу-виводу

не повинна перевищувати 31. Така високопродуктивна мережа забезпечує передачу даних вводу-виводу зі швидкістю 1,544 Мбіт/с.

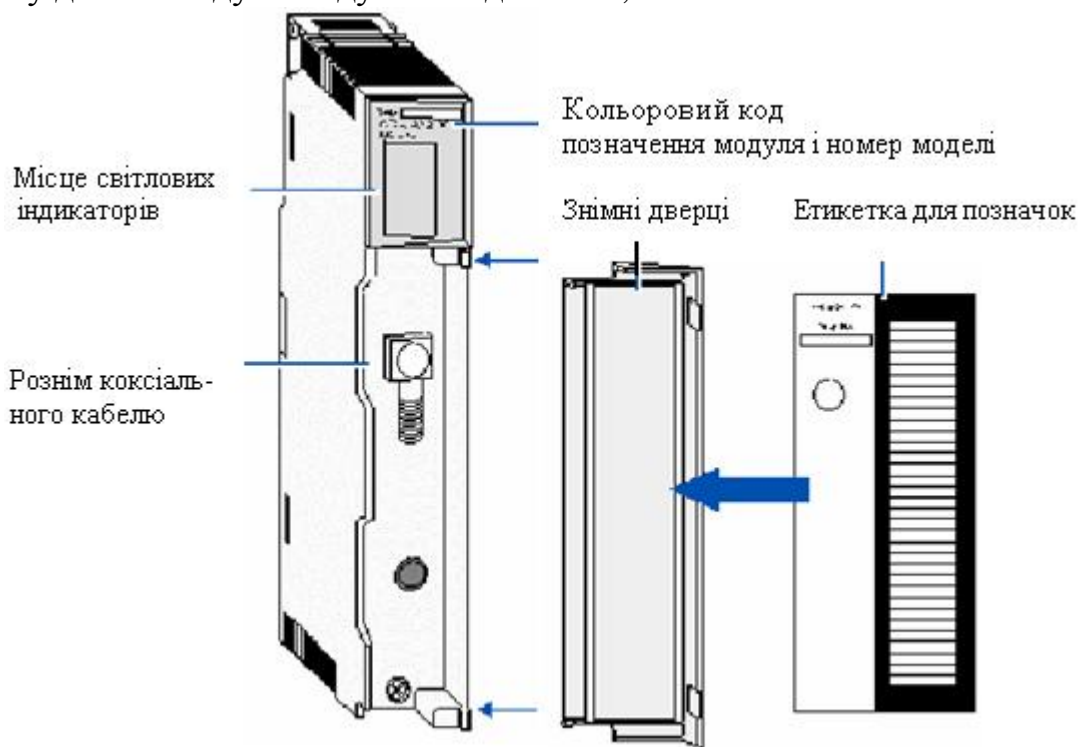
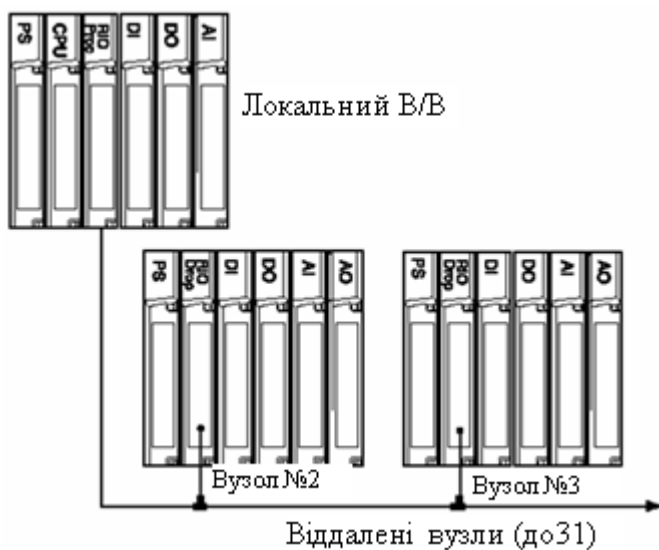


Рисунок 8.6 – Модуль віддаленого вводу-виводу RIO

Типова система віддаленого вводу-виводу приведена на рис.8.7. Кожному вузлу в мережі RIO має бути надана унікальна адреса. Якщо головний вузол використовує локальні модулі вводу-виводу, то перший віддалений вузол вважається вузлом №2. У випадку, якщо використовуваний контролер не підтримує локального вводу-виводу, то перший віддалений вузол вважається вузлом №1. Процесор RIO використовує цю адресу для посилення даних у відповідний підканал. Фізичне розташування вузла в мережі не впливає на його адресу або на пропускну спроможність каналу, що робить мережу RIO справжньою шиною передачі даних.



### Рисунок 8.7 – Типова система віддаленого вводу-виводу

Для установлення унікальної адреси вузла, адаптери RIO мають перемикачі. На адаптерах типу 984 використовуються DIP- перемикачі, а на адаптерах серії Quantum використовуються два поворотні перемикачі на задній панелі процесора. Верхній перемикач (SW1) встановлює десятки в значенні адреси, а нижній (SW2)- одиниці.

Повідомлення, що ініційоване головним процесором, передається по кабельній мережі і приймається всіма адаптерами RIO. Адаптер RIO з адресою, співпадаючою з адресою, що міститься в повідомленні, може потім за певний проміжок часу передати відповідь у процесор RIO . Якщо адаптер не відповідає, те саме повідомлення посилається знову. Цей процес називається перепосиланням повідомлень. Якщо адаптер не відповідає після декількох перепосилань, то вузол вважається несправним. Після кожного циклу обробки програми, головний процесор RIO намагається відновити комунікації з адаптером, при цьому робиться одна спроба на кожен цикл сканування, доти поки вузол не повернеться в робочий стан.

Магістральний коаксіальний кабель в конфігурації віддаленого вводу-виводу починається від головного процесора і продовжується уздовж всієї довжини мережі. На магістральному кабелі встановлюються відгалужувачі, за допомогою яких він приєднується до вузлів, використовуючи кабелі відгалуження. Цей кабель приєднується до відгалужувача і адаптера вузла RIO за допомогою спеціальних конекторів.

Для правильного узгодження імпедансу кабельної системи використовуються 75-омні резистори, які знаходяться усередині більшості адаптерів, і автоматично погоджують кабельні підключення.

Якщо головний процесор і адаптери вузлів мають два порти, то для систем, що потребують особливої надійності, можливий варіант конфігурації віддаленого вводу-виводу з подвійним кабелем. Це оберігає систему від наслідків обриву одного з кабелів або пошкодження рознімів. Тестування кабелю і перевірка його цілісності проводяться автоматично як контролером, так і кожним з вузлів. У разі обриву кабелю система Modicon TSX Quantum повідомляє користувача за допомогою сигнального світлодіода і внутрішньої програми перевірки працездатності.

Для підвищення перешкодостійкості мережі і збільшення її довжини до 16 км розроблені ретранслятори на волоконно-оптичному кабелі. Ретранслятори здійснюють перехід від кабелю типу "вита пара" або коаксіального кабелю до стандартного волоконно-оптичного кабелю діаметром 62,5-125 мкм, із збереженням динамічного діапазону мережі, і підвищують гнучкість системи віддаленого вводу-виводу[35,36,37].

#### **8.1.5.3 Конфігурація розподіленого вводу-виводу(DIO)**

Для додатків, що вимагають наявності просторово розподілених невеликих вузлів, система Quantum може реалізувати схему розподіленого вводу-виводу (DIO), що заснована на технології Modbus Plus. Ця схема

дозволяє розташувати модулі вводу-виводу ближче до робочої ділянки і зменшити, таким чином, витрати на підключення датчиків і виконавчих механізмів. При цьому в схемі розподіленого вводу-виводу використовуються ті самі модулі, що і в системах локального і віддаленого вводу-виводу, і недорогий кабель "вита пара".

Для оптимізації топології системи, підвищення її продуктивності, або збільшення кількості модулів вводу-виводу контролери серії Modicon TSX Quantum можуть приєднуватися до трьох мереж розподіленого вводу-виводу. Одна мережа реалізується через порт Modbus Plus, що є на ЦПУ. Дві інші - за допомогою додаткових модулів інтерфейсного зв'язку NOM. У систему можуть бути встановлені до двох модулів NOM, які мають додаткові порти Modbus і Modbus Plus і забезпечують розширені можливості зв'язку для системи Quantum.

Модуль розширення 140 NOM 211 00 має один канал зв'язку, а модуль 140 NOM 212 00 – два канали зв'язку. Світлові індикатори на лицевій стороні характеризують результати діагностики живлення, показують в якому режимі знаходиться модуль, сигналізують стан зв'язку через порти Modbus і Modbus Plus.

На передній панелі модуля розташовані два трипозиційні пересувні перемикачі. Лівий перемикач не використовується. Правий перемикач необхідний для встановлення параметрів зв'язку порту Modbus (RS-232) модуля розширення мережі Modbus Plus.

Верхнє положення перемикача призначає для порту режим ASCII, середнє положення перемикача призначає для порту режим віддаленого пристрою з параметрами зв'язку RTU, а нижнє положення перемикача дає можливість програмно призначити параметри порту зв'язку.

На задній панелі модуля розташовані два кругові перемикачі. Вони використовуються для встановлення адреси вузла даного модуля в мережі Modbus Plus.

Обертанням верхнього перемикача (SW1) встановлюються десятки в значенні адреси, а обертанням SW2 - одиниці в адресі вузла.

Мережа розподіленого вводу-виводу DIO в мінімальній конфігурації складається з одного контролера (ЦПУ) і одного або більш вузлів розташованих поблизу з об'єктами. Кожен вузол складається зі спеціального адаптера DIO з вбудованим блоком живлення або блока Momentum. Адаптери DIO призначені для сполучення модулів вводу-виводу з головним процесором за допомогою екранованої виті пари. Адаптер забезпечує модулі живленням від внутрішнього джерела напругою =24В (140CRA 211 20, 140CRA 212 20) або ~115/230В (140CRA 211 10, 140CRA 212 10). Вузол DIO менше за вузол RIO, проте можливе використання більшої кількості вузлів DIO на більшій площі. RIO підтримує лінійну конфігурацію довжиною до 4500м, а архітектура DIO- до трьох головних мережних вузлів на один ЦПУ при довжині до 450м на одну мережу. На рис.8.8 одна мережа DIO містить процесор (CPU), адаптер DIO і модуль Momentum. Дві інші мережі DIO

складаються з додаткових мережних модулів (NOM), адаптерів DIO і блоків Momentum. Мережа складається з однієї або декількох кабельних секцій. Кожна секція підтримує до 32 вузлів і має максимальну довжину кабелю 450м. Для збільшення довжини мережі до 1800м і підтримки до 64 вузлів секції можуть бути сполучені пристроями повторювачами Modicon. Мінімальна довжина кабелю між будь-якою парою вузлів повинна бути принаймні 3м. Максимальна довжина кабелю між двома вузлами така сама, як і максимальна довжина секції 450м.

Для великих відстаней до 15750 м застосовуються оптоволоконні кабелі.

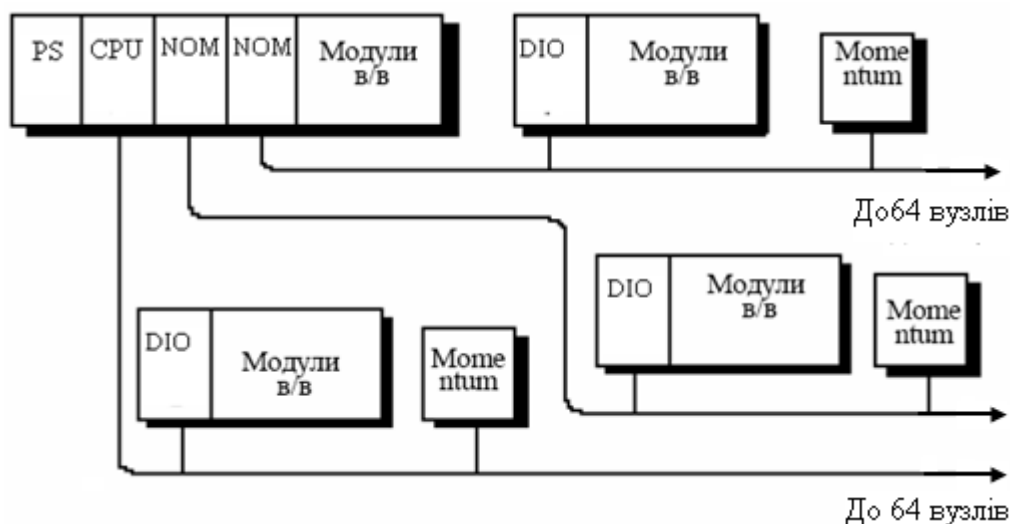


Рисунок 8.8 – Мережа розподіленого вводу-виводу DIO

Вузли мережі визначені адресами, які встановлює користувач. Кожна адреса вузла не залежить від його фізичного розташування. Адреси можуть бути визначені десятковими числами від 1 до 64 і не зобов'язані бути послідовними. Однакові адреси не дозволені.

Для систем з підвищеними вимогами до надійності є варіант реалізації з подвійним кабелем. Якщо відбувається обрив кабелю, то за допомогою сигнального світлодіода і засобів програмної діагностики, можна локалізувати вузол, де відбуваються збої, і визначити пошкоджений кабель.

У мережах з подвійним кабелем, кабелі звичайно позначаються як А і В. Кожний кабель може бути завдовжки до 450 м (між двома крайніми пристроями). Різниця в довжині кабелів А і В не повинна перевищувати 150 м між будь-якими двома вузлами на одній секції. Вузли приєднуються до кабелю за допомогою відгалужувачів, що поставляються фірмою Modicon. Вони забезпечують з'єднання магістрального кабелю, відгалуження для пристрою і заземлення.

Для підвищення загальної надійності в роботі контролера передбачена система гарячого резервування. Дві ідентичні системи Quantum зв'язані між собою. Один з контролерів основний, а інший працює як гарячий резерв, готовий перейняти на себе управління в разі відмови основного контролера.

[35,36,38].



### 8.1.6 Система гарячого резервування

Система гарячого резервування забезпечує високий рівень експлуатаційної готовності ЦПУ Quantum, завдяки наявності двох шасі ПЛК – основного і резервного, апаратна конфігурація яких повністю співпадає. Основним елементом кожного шасі є процесор 140 CPU 671 60 спеціально розроблений для архітектури гарячого резерву з програмним забезпеченням Unity Pro. Цей процесор є двослотовим модулем, який в одному корпусі об'єднує функції центрального процесора з функціями резервного сопроцесора, рис. 8.9.

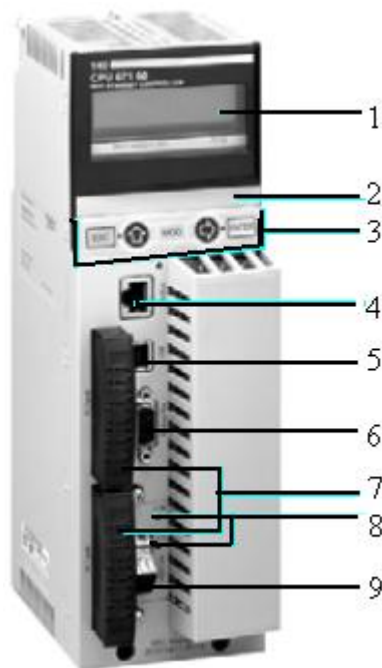


Рисунок 8.9 – Процесор гарячого резерву 140 CPU 671 60

На передній панелі ЦПУ знаходиться мінітермінал з ЖК-дисплеєм (1) і клавіатурою 3, який дозволяє контролювати стан ПЛК без програматора, відсік для батареї живлення 2, рознім для підключення до шини Modbus (4), рознім для підключення ПК-програматора (5), рознім для підключення до мережі Modbus Plus (6), два слота для карток розширення пам'яті (7), рознім для підключення волоконно-оптичного кабелю, що з'єднує первинний і вторинний ПЛК в архітектурі гарячого резервування (9) і два світлодіоди, які відображають стан вузлів гарячого резерву (8).

Основний ПЛК виконує прикладну програму і контролює входи-виходи. Резервний ПЛК залишається у фоновому режимі і готовий у будь-яку мить прийняти на себе управління. Резервний ПЛК з'єднаний з основним високошвидкісним волоконо-оптичним каналом зв'язку (100Мбод), що вбудований в ЦПУ. Довжина цього волоконо-оптичного каналу без додаткового спеціального обладнання може сягати 2км. За допомогою цього

каналу здійснюється періодичне оновлення даних користувальницького додатку у резервному ПЛК.

Коли основний контролер відмовляється працювати, система резервування автоматично переводить виконання програми і управління входами-виходами на резервний ПЛК, який стає основним. Після відновлення роботоздатності контролера, що вийшов з ладу, і повторного підключення його до системи резервування він стає резервним. Приклад архітектури системи гарячого резерву наведений на рис. 8.10.

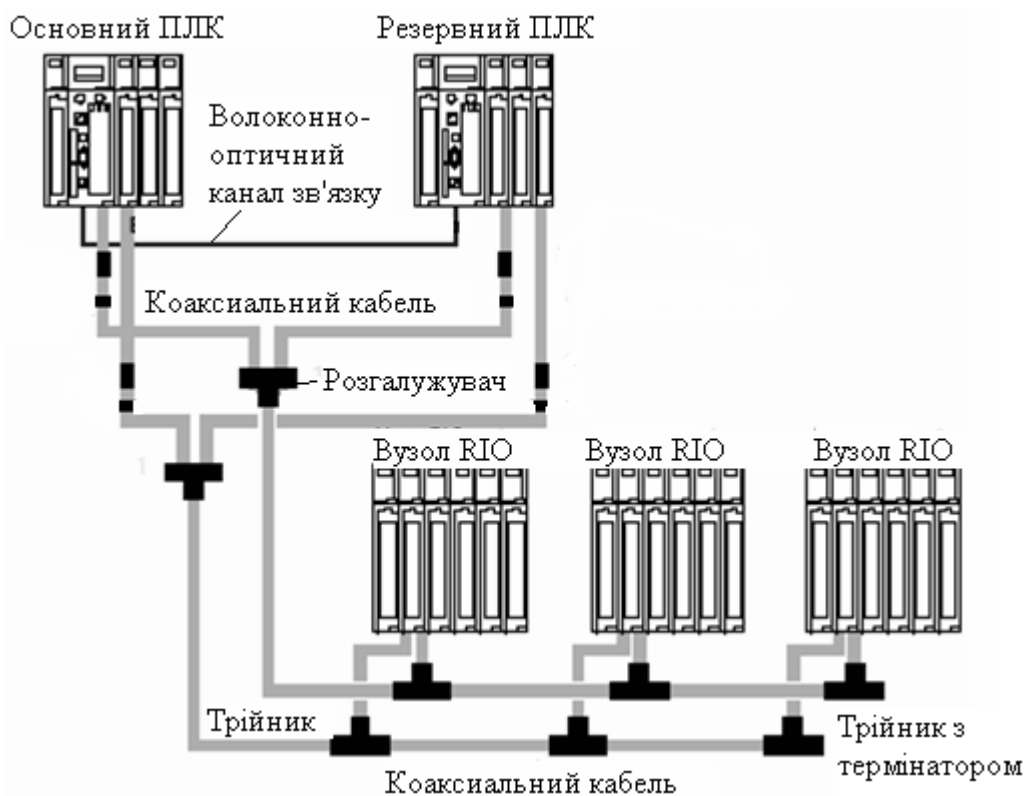


Рисунок 8.10 – Приклад архітектури системи гарячого резерву

Використання системи гарячого резерву з програмним забезпеченням Unity Pro забезпечує безударне перемикання з основного на резервний ПЛК, що збільшує продуктивність і мінімізує час простою [36].

### 8.1.7 Система управління Momentum

Система Modicon TSX Momentum створена для реалізації на базі потужних контролерів виробництва Schneider Electric віддаленого та розподіленого управління за допомогою поширених відкритих промислових мереж - Modbus Plus, FIPIO, Ethernet, Interbus, Profibus, DeviceNet ControlNet. Modicon TSX Momentum складається з 4 основних компонентів, які легко з'єднуються один з одним в різних поєднаннях:

- адаптери зв'язку;
- базові блоки вводу-виводу;
- процесорні адаптери;

- додаткові адаптери.

Конструкція пристроїв серії Momentum – модульна. Базові блоки вводу-виводу Momentum є основою для всієї системи управління. До базових блоків вводу-виводу підключаються адаптери зв'язку, процесорні і додаткові адаптери, утворюючи віддалений вузол вводу-виводу.

Процесорні адаптери Momentum M1 є повноцінними ПЛК з ЦПУ, ОЗП і флеш-пам'яттю. Вони сумісні з ПЛК Quantum і підключаються до базових блоків вводу-виводу, як і адаптери зв'язку. Наявність процесорних адаптерів у складі Momentum дозволяє будувати самостійну невелику розподілену систему управління.

Адаптери зв'язку і процесорні адаптери спроектовані у вигляді функціональних модулів, які встановлюються на базові блоки вводу-виводу Momentum.

Додаткові адаптери забезпечують для процесорних адаптерів додаткові мережні функції, годинник реального часу і резервне батарейне живлення.

Базові блоки вводу-виводу встановлюються в стандартних корпусах, які монтуються на DIN-рейку або панель в шафі.

Підключення до базових блоків вводу-виводу польових пристроїв і живлення здійснюється за допомогою 18-контактних клемних колодок, а також додаткових 1, 2 або 3-рядних шин.

Базові блоки аналогового вводу Momentum призначені для підключення різних промислових аналогових входних сигналів, а базові блоки аналогового виводу застосовуються для управління аналоговими польовими пристроями.

На лицевій панелі базового блока аналогового вводу-виводу 170 AAI є внутрішній інтерфейсний з'єднувач для підключення модуля зв'язку або процесорного модуля, світлодіодні індикатори стану (кількість індикаторів залежить від числа каналів), два з'єднувача для знімних клемних колодок, слот для шини живлення і захисна кришка.

Спеціалізовані базові блоки вводу-виводу Momentum призначені для розв'язання специфічних задач, що розширюють сферу застосування пристроїв Momentum. До спеціалізованих базових блоків відносяться:

- базовий блок 2-канального високошвидкісного лічильника;
- базовий блок дискретного вводу-виводу з портом зв'язку Modbus;
- базовий блок, що забезпечує взаємодію між Momentum і мережею компонентів SERIPLEX.

Базовий блок високошвидкісного лічильника 170 AEC 920 00 має 2 незалежні лічильники, а також 6 дискретних входів і 4 дискретні виходи. Цей блок можна напряму приєднувати до багатьох стандартних мереж шляхом установлення на базовому блоці знімних адаптерів зв'язку Momentum.

Базовий блок дискретного вводу-виводу 170 ADM 540 80 має 6 дискретних входів і 3 дискретні виходи, а також порт зв'язку Modbus для підключення до зовнішніх пристроїв. Цей блок можна використовувати як базовий блок вводу-виводу для програмованого контролера з автономною або

розподіленою конфігурацією вводу-виводу шляхом установалення одного знімного процесорного адаптера Momentum M1.

Базовий блок 170 ANM 050 10 є інтерфейсом SERIPLEX для контролерів Momentum і забезпечує обмін даними вводу-виводу по мережі компонентів SERIPLEX. Цей блок є підпорядкованим свого хост-комп'ютера і провідним пристроєм шини SERIPLEX. До базового блока SERIPLEX можна підключити процесорний адаптер Momentum або адаптер зв'язку, що дозволяє підтримувати могутню трирівневу архітектуру управління.

Спеціалізовані базові блоки вводу-виводу містять внутрішній з'єднувач інтерфейсу для модуля зв'язку, світлодіодні індикатори стану, два різні для знімних клемних колодок, слот для монтажу шини заземлення, захисну кришку для запобіжників (170 ADM 54080) або з'єднувач для знімної клемної колодки.

Адаптер зв'язку моделі Ethernet 170 ENT 110 00 забезпечує пряме приєднання всієї серії модулів вводу-виводу Momentum до мереж, заснованих на Ethernet. Подібна можливість підключення дозволяє організувати зв'язок з широким спектром пристроїв управління, сумісних з Ethernet TCP/IP.

В адаптері мережі Ethernet поверх протоколу TCP/IP використовується стандартна структура повідомлень і команди управління Modbus. Це дозволяє спростити упродовження силами фахівців АСУ і одержати дані, які можуть передаватися по стандартному мережному фізичному середовищу на всі системи підприємства.

Адаптери зв'язку Modbus Plus для пристроїв серії Momentum можна підключати до будь-якого базового блока вводу-виводу Momentum з метою створення функціонального вузла вводу-виводу на шині Modbus Plus і забезпечення прямого приєднання всієї серії модулів вводу-виводу Momentum до мережі Modbus Plus. Для розширення можливостей мережі Modbus Plus з архітектурою розподіленого вводу-виводу адаптери зв'язку дозволяють підключити до мережі до 64 модулів вводу-виводу Momentum без використання повторювачів сигналу.

Кожен модуль вводу-виводу Momentum є окремим вузлом в мережі Modbus Plus зі своєю адресою, вибраною користувачем за допомогою подвійного поворотного перемикача на лицевій панелі адаптера зв'язку [39].

### **8.1.8 Мережні можливості**

Система Quantum надає можливість підключення до відкритих стандартних мереж і польових шин, які вона підтримує. Серед них Modbus, Modbus Plus, Ethernet, InterBus-S, LonWorks і Profibus.

Локальна мережа Modicon Modbus Plus дозволяє організувати одноранговий обмін інформацією між управляючими ЕОМ, контролерами і іншими джерелами даних через виту пару (модуль 140 NOM 21100) або волоконно-оптичний кабель (модуль 140 NOM 25200). Мережа Modbus Plus забезпечує доступ до даних процесу зі швидкістю 1 Мбод. Зручність цієї мережі полягає у тому, що вона дозволяє управляти пристроями реального

часу, такими, як модулі вводу-виводу і приводи, без зниження продуктивності, викликаного завантаженням мережі.

Для підвищення надійності в роботі системи автоматизації, використовується варіант мережі з резервним кабелем (модуль 140 NOM 21200). Це дозволяє здійснювати зв'язок по двох незалежних кабелях, перевіряючи цілісність обох кабелів при кожному посиленні.

Всі контролери Quantum мають вбудований порт мережі Modbus з інтерфейсом RS-232, через який здійснюється оперативне програмування або збір даних. Зв'язок між мережами Modbus і Modbus Plus в контролері відбувається автоматично при включенні режиму моста за допомогою перемички або шляхом налаштування ПЛК. Як наслідок, послання в мережі Modbus перенаправляються в мережу Modbus Plus.

Мережні модулі Quantum Ethernet надають ще один спосіб зв'язку між контролерами Quantum і/або інформаційними системами. Підтримується промисловий стандарт TCP/IP, а також протокол Sy/Max. Фізичними носіями можуть бути вита пара 10Base-T (модуль 140 NOE 21100) або волоконна оптика 10Base-FL (модуль 140 NOE 25100). Модуль Ethernet підтримує як одноранговий зв'язок, так і зв'язок «контролер – хост». Для зв'язку між хост-системами і мережними модулями Ethernet TCP/IP системи Quantum використовується протокольний рівень Modbus, розроблений для застосування у системах управління реального часу.

Модулі Ethernet Sy/Max (140 NOE 31100 і 140 NOE 35100) дозволяють користувачам пристроїв серії Sy/Max легко і безболісно підключати наявні мережі Sy/Max до платформи Quantum.

Рішення Modicon в області зв'язку забезпечують сумісність і можливість об'єднання мереж Ethernet TCP/IP, Modbus Plus і Modbus, оскільки вони використовують Modbus, як протокол.

Серед комунікаційних модулів системи Quantum є модуль 140 NOA 61100, який призначений для обслуговування до 256 пристроїв із загальною кількістю точок вводу-виводу 4096 у мережі InterBus-S. Польова шина InterBus-S використовує топологію "провідний/підпорядкований" і за допомогою витой пари обслуговує модулі вводу-виводу на відстані до 13 км. Пропускна спроможність InterBus-S – 500 Кбод, причому дані для логічної обробки передаються в контролер Quantum кожен цикл.

У мережі LonWorks, що підтримує контролер Quantum, використовується принцип розподіленого управління. Мережна технологія LonWorks, розроблена компанією Echelon, дозволяє об'єднувати пристрої автоматизації за допомогою однорангової мережі, що використовує протокол моделі OSI (взаємодія відкритих систем). Для задач збору даних і управління, пристроям призначаються стандартні або задаються мережні змінні, які передаються в мережний модуль Quantum LonWorks (FTT 10140 NOL 91100, TPT/XF-78140 NOL 91110 або TPT/XF-1250140 NOL 91120) зі швидкістю 1,25 Мбод. Технологія Quantum LonWorks надає відкритий стандарт підключення,

необхідний для розширення традиційної області застосування програмованих контролерів.

### 8.1.9 Програмування контролерів Modicon TSX Quantum

Програмування контролерів Modicon TSX Quantum може здійснюватися за допомогою інструментальної системи програмування Unity Pro або Concept.

Unity Pro — єдине середовище розробки і налагодження додатків користувача, а також операційна система для контролерів Modicon Premium, Atrium і Quantum. П'ять мов ІЕС 61131-3 програмування, інструменти налагодження і діагностики, дозволяють Unity Pro підвищити продуктивність розробки і спростити обслуговування систем автоматизації.

У процесі розроблення програми користувача можна використовувати спеціальні засоби для її налагоджування - покрокове виконання, таблиці анімації (Animation Tables) та багато інших.

Структура програми користувача ПЛК може бути однозадачною чи багатозадачною. В однозадачній структурі є тільки одна основна задача (MAST), що може виконуватись у циклічному або періодичному режимах. У багатозадачній структурі, окрім задачі MAST (яка присутня завжди), можуть виконуватися так звана швидка задача FAST і кілька задач обробки подій.

Кожна з цих типів задач має різний пріоритет виконання. Найвищий пріоритет мають задачі обробки подій, найнижчий - задача MAST.

Concept - це сучасний інструмент для ОС Windows, який надає єдине багатомовне середовище для програмування систем управління на базі контролерів серії Modicon TSX Quantum.

Основними перевагами цього пакету є досконалі графічні редактори і контекстна допомога, можливість використання всіх п'яти мов МЕК-стандарту і наявність групи елементів для регулювання, нечіткого управління, системного конфігурування та діагностики, могутні функції пошуку змінних, помилок, визначення невикористаних змінних і перевірки роботи програми за допомогою емулятора.

Проект користувача складається з однієї або декількох секцій, за допомогою яких визначається специфіка функціонування окремих компонентів і системи в цілому. У програмі вказується порядок виконання секцій, характеризуються змінні, константи, підтримуються пряма адресація і декілька різних типів даних[40].

#### 8.1.9.1 Методика створення проекту у системі програмування Concept

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

Регулювання тиску за парогенератором  $P(t)$ , рис.8.11, відповідно до заданого значення  $Z_{dn}$ , здійснює коригуючий регулятор  $W_k(p)$ , який формує завдання стабілізуючому регулятору  $W_{ст}(p)$ . Вихідний сигнал стабілізуючого регулятора  $Y_{рег}$  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  $P\delta(t)$ .

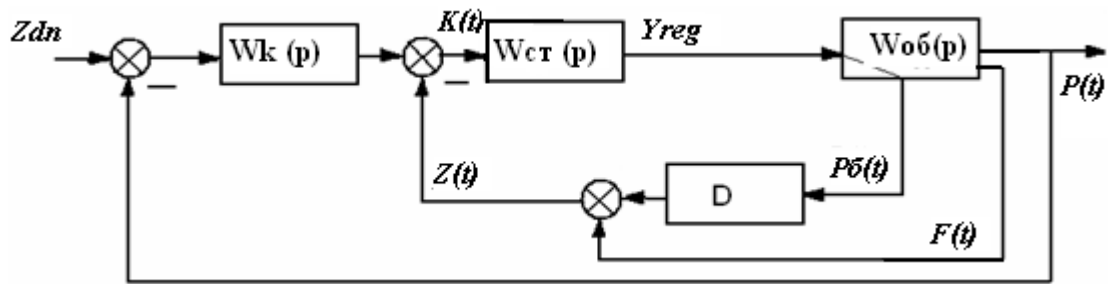


Рисунок 8.11 – Структурна схема системи автоматичного регулювання тиску за парогенератором

### *Розробка проекту*

Запустити програму Concept V 2.2 і в головному меню вибрати шлях File > New project. Після появи на моніторі нового проекту увійти в підміну File > New section. У діалоговому вікні New Program Section вибрати FBD-редактор програмування і присвоїти ім'я секції. Після натиснення 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 установлення 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 адресні інтервали для розміщення вхідних-вихідних бітів/маркерів дискретних (Discretes) сигналів і 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 мережі;
- Hot Standby - активується, коли для підвищення надійності використовуються два дубльованих ПЛК.

Ці опції в проекті не використовуються, тому і не активуються.

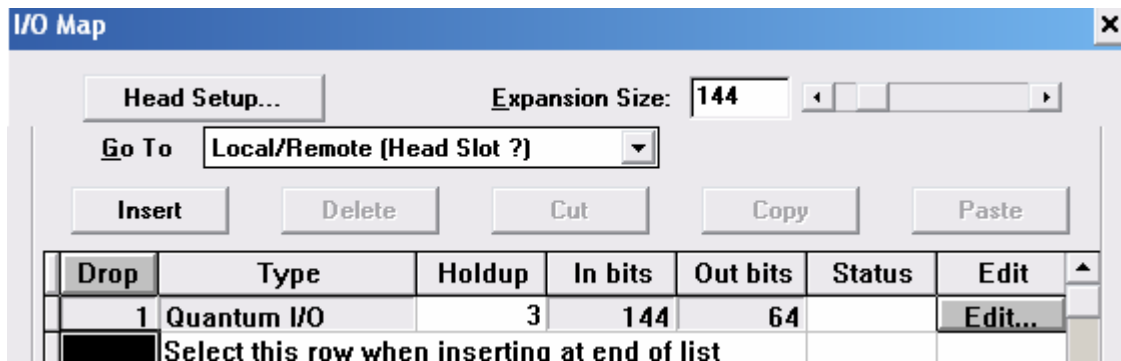
У вікні поля ASCII встановлюються уставки, що регламентують ASCII-повідомлення, при програмуванні мовою LL984.

Остаточно вікно PLC Configuration має вигляд:

| PLC Configuration  |                 |                       |   |
|--------------------|-----------------|-----------------------|---|
| PLC                |                 |                       |   |
| Type:              | 140 CPU 113 03  | Available Logic Area: | 0 |
| 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:             | 1 |
| 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 |



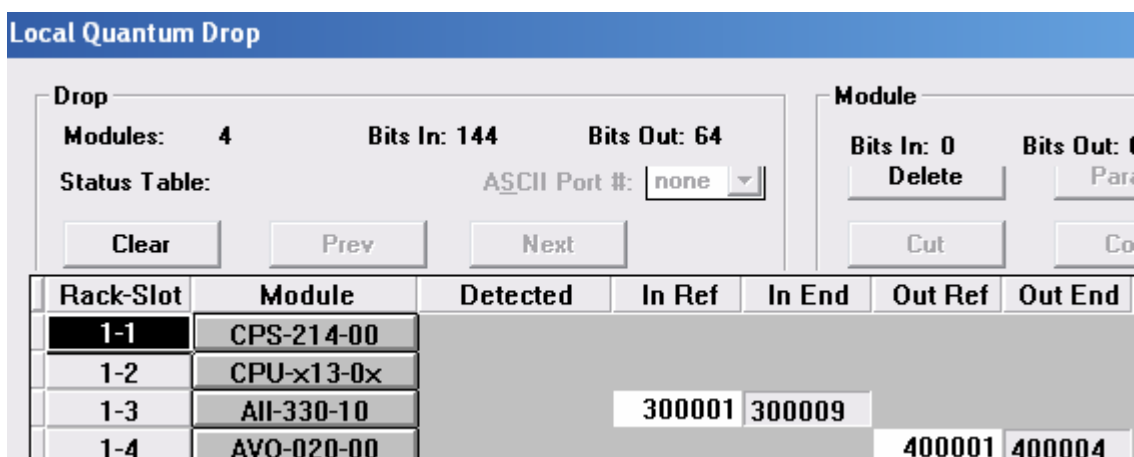
Конфігурування модулів вводу-виводу, а також задіяних каналів здійснюється за допомогою I/O Map... карти, яка викликається в меню вікна PLC Configuration через пункт Configure / I/O Map...:




Перед вибором модулів необхідно у списку Go To визначитися з видом конфігурації (локальна, RIO або DIO). При використанні відповідні послання:

- послання, що починаються з цифри 3 (3-посилання), вводяться для модулів аналогового вводу;
- послання, що починаються з цифри 4 (4-посилання), вводяться для модулів аналогового виводу;
- послання, що починаються з цифр 3 або 1 (3-посилання або 1 - послання), вводяться для модулів дискретного вводу;
- послання, 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, вибрати по черзі модуль живлення CPS-214-00, центральний процесор CPU-x13-0x, модуль аналогового вводу AI-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 адреси 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 ЕОМ залишається робоче поле для розробки програмного коду.


### ***Створення програми***


Відповідно до структурної схеми системи автоматичного регулювання, 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 встановлення 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 ЕОМ місце встановлення блока на робочому полі і натиснути ліву клавішу миші. Для відновлення покажчика слід скористатися кнопкою .

Таким самим чином встановити 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:

The Variable Editor dialog box is shown with the following data:

| Exp                      | Variable Name | Data Type | Address | InitValue | Used |
|--------------------------|---------------|-----------|---------|-----------|------|
| <input type="checkbox"/> | Ft            | WORD      | 300001  |           | 1    |
| <input type="checkbox"/> | Kt            | REAL      |         |           | 1    |
| <input type="checkbox"/> | Pb            | WORD      | 300003  |           | 1    |
| <input type="checkbox"/> | Pt            | WORD      | 300007  |           | 1    |
| <input type="checkbox"/> | Yreg          | WORD      | 400001  |           | 1    |
| <input type="checkbox"/> | Zt            | REAL      |         |           | 1    |

| Variable Name | Data Type  | Value  | Used |
|---------------|------------|--------|------|
| pv1           | REAL       | 6.0    | 2    |
| para          | Para DERIV | Set... | 1    |
| para1         | Para PI    | Set... | 1    |
| para2         | Para PI    | Set... | 1    |
| zdn           | WORD       | 10     | 1    |

**Definition Para\_PI**

| Name  | Type    | Value  |
|-------|---------|--------|
| para2 | Para_PI |        |
| gain  | REAL    | 2.0    |
| ti    | TIME    | t#5s   |
| ymax  | REAL    | 3200.0 |
| ymin  | REAL    | 0.0    |

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

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

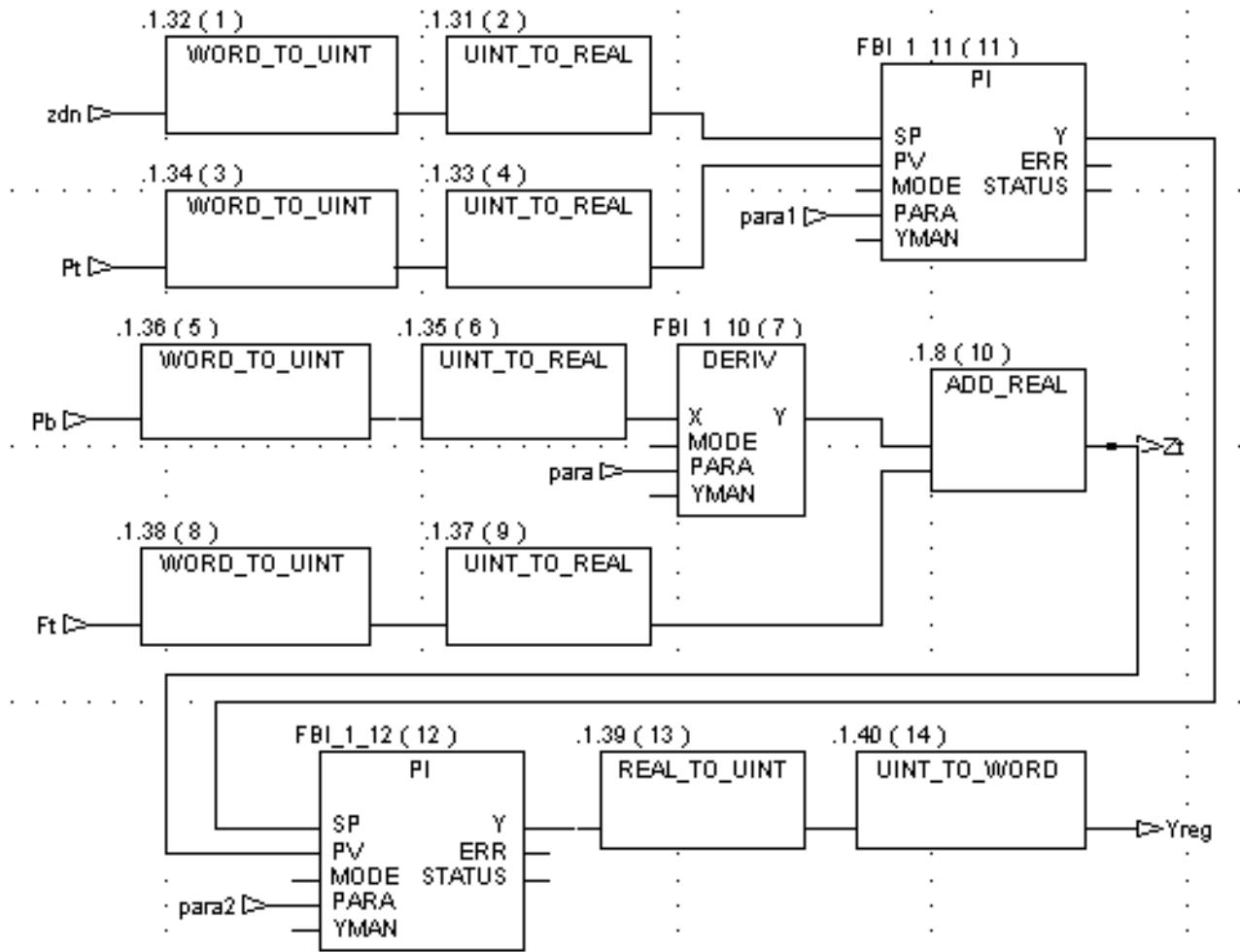
У назві вікна симулятора, що з'явиться, показане ім'я утиліти PLC Sim32 і адреса TCP/IP плати інтерфейсу ПК. Вікно симулятора складається з 3 закладок:

- 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.

Реалізація програми FBD-мовою має наступний вигляд:



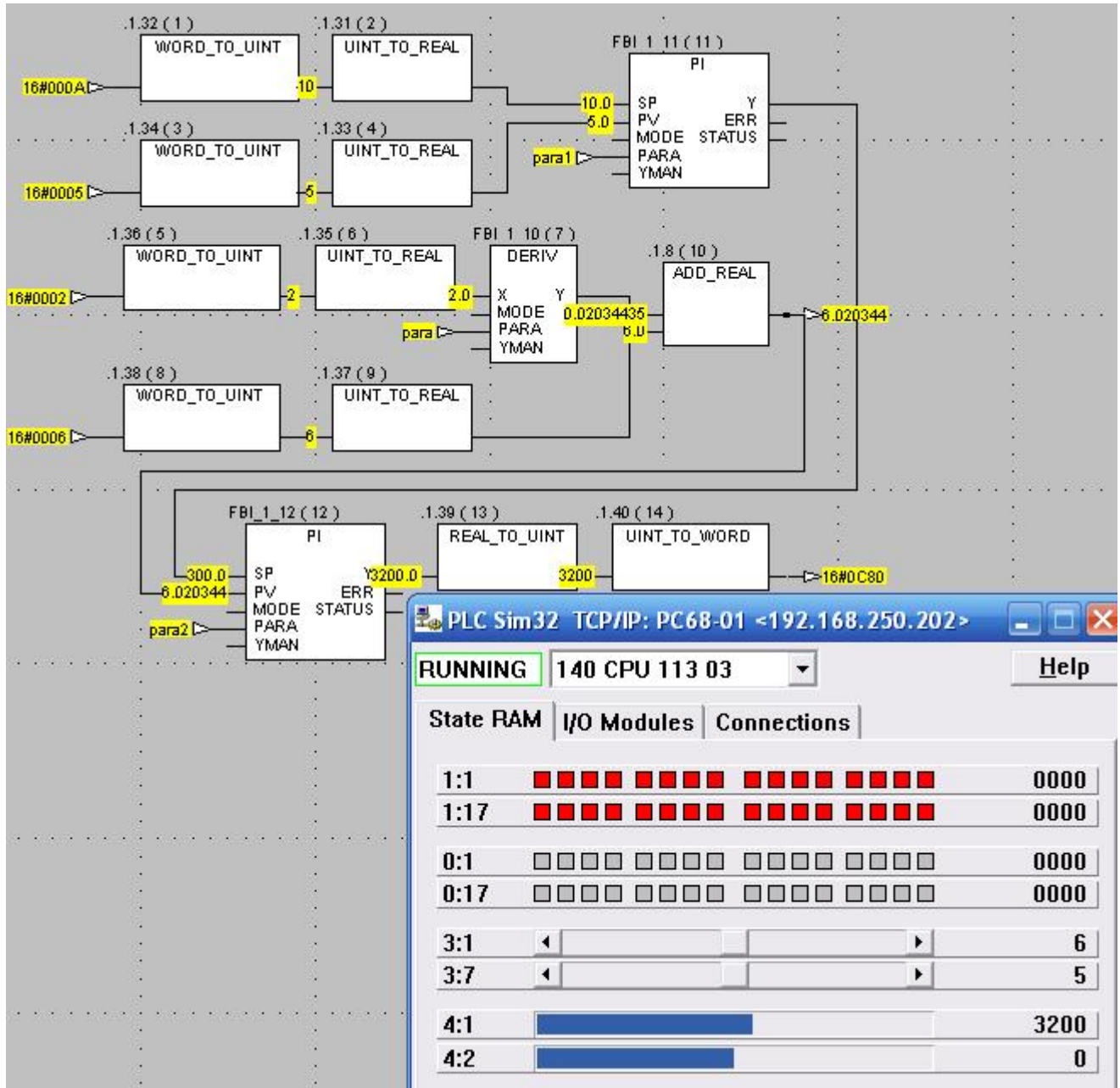
За допомогою меню Online->Download відкрити вікно завантаження проекту, позначити лівою клавішею миші позиції «IEC program section», «State RAM», «Initial values only» і натиснути кнопку Download. Програма запропонує запусити контролер, натиснувши кнопку «Так». Запустити і зупинити контролер можна також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

Щоб підготувати проект до симуляції на клавіатурі EOM 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.

Повзунцем на симуляторі №3 регістру пам'яті дискретних змінних задати певний вхідний сигнал. У програмному коді, значення вихідної змінної почнуть

змінюватися відповідно до настройок функціонального блока PI-регулятора. Для вибору потрібного регістра треба клацнути лівою клавшею миші по одному з двох номерів симулятора №3 і у вікні Input Register, що з'явиться, записати потрібний номер регістра.

Екран проекту в процесі симуляції має наступний вигляд:



Після підтвердження рішення про зупинення контролера закрити вікно Online Control Panel кнопкою Close.

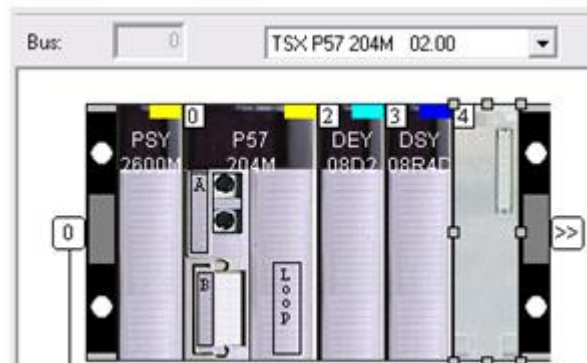
Для вивантаження програми з контролера використати меню Online → Upload...-і у вікні Upload Controller, що з'явиться, позначити позицію Configuration. Після натиснення кнопки Upload, зв'язок між ПЛК і ПК зникне [40].

### 8.1.9.2 Методика створення проекту у системі програмування Unity PRO XL

Для ПЛК TSX Premium 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.

#### *Розробка 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 повністю скомпонований, необхідно закрити вікно X Bus, щоб внести зроблені уставки у теку Configuration проектного дерева.




#### *Створення програми*

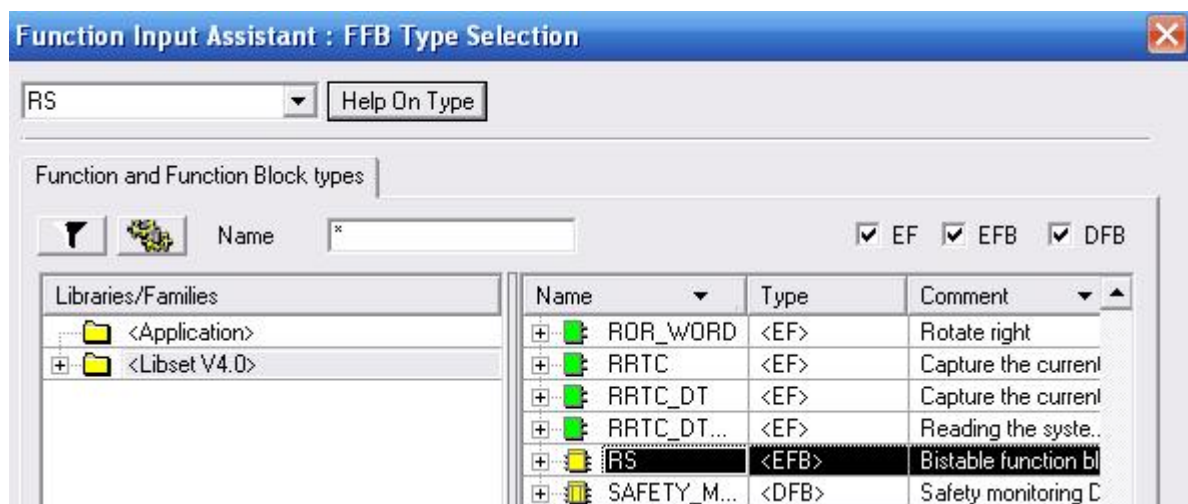
Для створення програмної секції подвійним клацанням лівої клавіші миші відкрити у дереві проекту теку Program, а потім теку MAST. Правою клавішею клацнути по теці Section і у New Section, що з'явиться, створити нову секцію, надати ім'я, вибрати ST-редактор і натиснути ОК. Відкриється сформоване дерево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

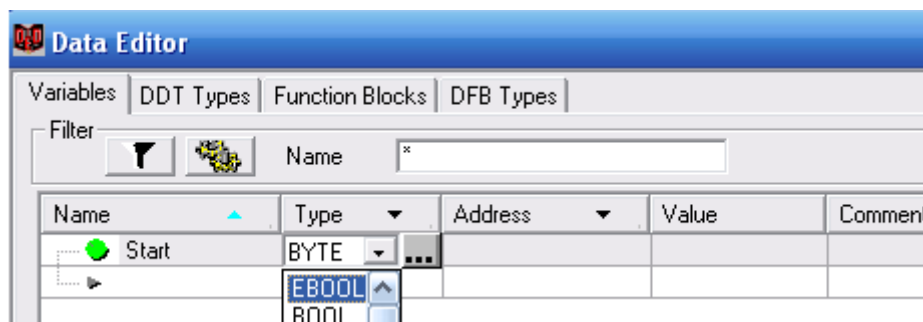


З'явиться вікно з характеристиками блока, типом і попереднім ім'ям його екземпляра (Instance).

Запропоноване ім'я екземпляра можна залишити, а можна і замінити на нове. Після натиснення ОК, функціональний блок вважається оголошеним і потрапляє у вкладку Elementary FB Instance теки Variables&FB Instance проектного дерева, 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 Elementari Variables і у вікні Data Editor двічі клацнути лівою клавшею миші на вільному рядку колонки Name і вписати ім'я змінної. Після натискання клавіші Enter на клавіатурі EOM, ім'я в рядку зафіксується і за

замовчуванням пропонується тип змінної. Якщо він не відповідає змінній, то після подвійного клацання лівою клавшею миші по ньому відкриється список, з якого вибрати необхідний тип змінної:



Подвійне натискання лівої клавшею миші на рядку у колонці Address відкриє його для запису адреси зовнішніх вхідної або вихідної змінних. 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.

Змінні та функціональні блоки, що оголошені у програмі наведені у табл.8.1 і табл. 8.2.

Таблиця 8.1 Вхідні і вихідні змінні, що оголошені у програмі

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

Таблиця 8.2 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      |







Програмний код, що 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;
```

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

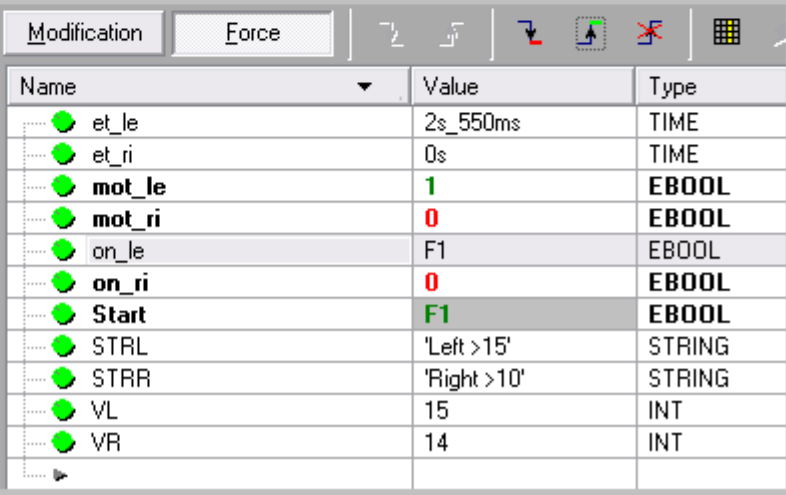
У головному меню Service->Initialize Animation Table відкрити анімаційну таблицю. Для її заповнення двічі клацнути лівою клавшею миші у першому рядку колонки Name і натиснути кнопку , що з'явиться. З вікна Animation Table послідовно вибрати змінні для анімації і подвійним клацанням

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 0). 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, рис.8.12.

```

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;

```








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

Рисунок 8.12 – Анімація роботи програмного коду

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)[41].

## Тестові та контрольні запитання до розділу 8

**I-й рівень запитань та тверджень дозволяє перевірити загальний рівень засвоєння матеріалу розділу і передбачає відповідь на запитання або твердження “Так” чи “Ні” .**

1. Центральний процесор Modicon TSX Quantum займає два місця на монтажній панелі.
2. Центральний процесор контролера Modicon TSX Quantum має стандартні порти зв'язку з підтримкою мережних протоколів Modicon Modbus та TCP/IP Ethernet.
3. Чи забезпечують джерела живлення 140 CPS 111 00 та 140 CPS 114 00 модулі контролера Modicon TSX Quantum напругою 24В постійного струму?
4. Нарощувані модулі забезпечують шасі струмом 16А.
5. Дискретні модулі серії 140 DAM 590 001, 140 DDM 390 001 та аналогові 140 АММ 090 001 є комбінованими, які об'єднують в певній пропорції канали вводу і канали виводу.
6. Для підвищення надійності і працездатності системи автоматизації у контролері Modicon TSX Quantum передбачається можливість "гарячої заміни" окремих модулів вводу-виводу.
7. Кожен модуль контролера Modicon TSX Quantum розташовується на монтажній панелі в залежності від його типу у певній послідовності.
8. Чи входять до складу модуля вводу-виводу ключі, які кодують клас модуля та ідентифікують розміщення модуля на монтажній панелі?
9. Чи підтримує конфігурація локального вводу-виводу контролера Modicon TSX Quantum 512 дискретних каналів вводу-виводу?
10. Конфігурація віддаленого вводу-виводу контролерів серії Modicon TSX Quantum використовується коли програмний додаток вимагає великої кількості каналів вводу-виводу або необхідно підключити контролер до вже існуючих віддалених вузлів вводу-виводу виробництва Modicon.
11. Чи є на лицевій панелі модуля віддаленого вводу-виводу перемикач для установлення параметрів зв'язку між головним каналом і підканалами?
12. Для визначення унікальної адреси вузла в мережі RIO використовується програма виробництва Modicon.
13. Чи використовується система розподіленого вводу-виводу (DIO) Quantum для створення просторово розподілених невеликих вузлів мережі?

14.Мережа розподіленого вводу-виводу DIO складається з одного або більше контролерів та вузлів, розташованих поблизу з об'єктами.

15.Чи здатні адаптери забезпечувати створення додаткових кабельних секцій та збільшувати довжину мережі?

16.Підключення польових пристроїв вводу-виводу і джерел живлення до базових блоків у системі Modicon TSX Momentum здійснюється за допомогою 18-контактних клемних колодок, а також додаткових 1, 2 або 3-рядних шин.

17.Для реалізації локальної, віддаленої і розподіленої конфігурації вводу-виводу використовують різні монтажні панелі.

18.Чи мають модулі джерел живлення особливі рекомендації щодо їх розміщення в слоти монтажної панелі?

19.Чи дозволяє система Modicon CableFast при монтажі модулів об'єднувати різні модуля вводу-виводу і клемну колодку за допомогою системного кабелю?

20.Контролери Modicon TSX Quantum програмуються тільки за допомогою інструментальної систем програмування Unity PRO XL, а контролери TSX Premium – у програмному середовищі Concept.

*Перевірте відповіді на сторінках: 1 - 198 с., 2 - 199 с., 3 - 200 с., 4 – 199...200 с., 5 - 200 с., 6 - 200 с., 7 - 201 с., 8 – 201...202 с., 9 - 204 с., 10 - 204 с., 11 - 204 с., 12 - 206 с., 13 - 206 с., 14 – 206...207 с., 15 - 211 с., 16 - 211 с., 17 - 196 с., 18 - 196 с., 19 - 203 с., 20 – 214...227 с.*

**II-й рівень запитань (тестові запитання) дозволяє перевірити рівень засвоєння матеріалу розділу із заглибленням у технічні особливості будови та структури контролерів Modicon TSX Quantum та передбачає вибір правильної відповіді з наведених трьох варіантів.**

**1. Для розв'язання яких задач призначений контролер Modicon TSXQuantum ?**

а) для розв'язання задач тільки логічного управління технологічними процесами;

б) для розв'язання задач тільки автоматичного регулювання;

в) для розв'язання будь-яких задач у галузі промислової автоматизації, навіть на рівні цілого підприємства.

**2. Скільки існує модифікацій центрального процесора контролера Modicon TSX Quantum?**

а) вісім;

б) п'ять;

в) шість.

**3. Скільки перемикачів знаходиться на передній панелі процесора Modicon TSX Quantum і яке їх призначення?**

а) два перемикача, перемикач зліва забороняє перезапис програми користувача у процесорі, другий - для установлення параметрів сполучення через порт Modbus;

б) два перемикача, перемикач зліва використовується для установлення порту Modbus, другий – для установлення адреси вузла в мережі Modbus Plus;

в) один перемикач, який дає можливість програмно призначити параметри порту сполучення .

#### **4. Автономний модуль живлення в системах управління Quantum забезпечує...**

а) модулі, що розташовані в одному шасі, напругою зі струмом навантаження 3А;

б) модулі, що розташовані в одному шасі, напругою зі струмом навантаження 8А ;

в) модулі, що розташовані в одному шасі, напругою зі струмом навантаження 16А .

#### **5. В якому випадку на одній монтажній панелі можуть бути встановлені два джерела живлення?**

а) коли до складу контролера входить більш ніж два модуля вводу-виводу;

б) коли до складу контролера входять центральні процесори з великою продуктивністю обробки інформації;

в) коли конфігурація контролера споживає більше, ніж номінальний струм одного джерела.

#### **6. До якої групи модулів контролера Modicon TSX Quantum належать модифікації 140 DAI і 140 DDI?**

а) до модулів дискретного та аналогового вводу;

б) до модулів дискретного вводу;

в) до модулів дискретного вводу-виводу.

#### **7. Які виходи мають модулі контролера Modicon TSX Quantum серії 140 DRA?**

а) нормально відкриті або нормально закриті контакти реле;

б) тільки нормально відкриті контакти реле;

в) тільки нормально закриті контакти реле.

#### **8. Як визначається адресація і параметри настройки модулів контролера Modicon TSX Quantum?**

а) за допомогою перемикачів на модулях контролера;

б) задаються при оформленні замовлення на контролер;

в) за допомогою програмного забезпечення Modicon.

#### **9. За допомогою чого здійснюється кодування модулів вводу-виводу?**

а) за допомогою програмного забезпечення Modicon;

б) за допомогою програмного забезпечення Modicon або за допомогою 12 первинних та 6 вторинних ключів;

в) за допомогою 12 первинних та 6 вторинних ключів.

#### **10. Які модулі можуть використовуватися в конфігурації локального вводу-виводу як додаткові системні?**

а) тільки модулі мережного інтерфейсу, які складаються з процесорів RIO або модулів розширення NOM;

б) модулі мережного інтерфейсу, які складаються з процесорів RIO або модулів розширення NOM, модуль управління багатокординатним переміщенням, процесор гарячого резерву;

в) будь-які модулі.

**11. За допомогою яких модулів реалізується конфігурація віддаленого вводу-виводу?**

а) за допомогою додаткових модулів інтерфейсного зв'язку 140 NOM 211 00 та 140 NOM 212 00, які встановлюються на платі центрального процесора і на платі віддаленого вводу-виводу;

б) за допомогою додаткових модулів інтерфейсного зв'язку 140 NOM 211 00 та 140 CRP 931 00, які встановлюються на платі центрального процесора і на платі віддаленого вводу-виводу;

в) за допомогою модулів інтерфейсу обміну даними RIO 140 CRP 931 00 і 140 CRP 932 00, які встановлюються на платі центрального процесора і на платі віддаленого вводу-виводу.

**12. Якщо головний вузол в мережі RIO використовує локальні модулі вводу-виводу, то який номер має перший віддалений вузол?**

а) кількість локальних модулів вводу-виводу + 1;

б) №2;

в) №1.

**13. Навіщо для RIO–мережі розроблені ретранслятори на волоконно-оптичному кабелі?**

а) для підвищення перешкодостійкості мережі і збільшення її довжини до 16 км;

б) для підвищення динамічного діапазону мережі і збільшення її довжини до 20 км;

в) для збільшення довжини мережі до 15 км.

**14. Скільки мереж розподіленого вводу-виводу можуть підтримувати контролери серії Modicon TSX Quantum?**

а) не більше трьох, одна мережа реалізується через порт Modbus Plus, що на ЦПУ, дві інші - за допомогою додаткових модулів інтерфейсного зв'язку NOM;

б) не більше двох, одна мережа реалізується через порт Modbus Plus, що на ЦПУ, одна – за допомогою додаткових модулів інтерфейсного зв'язку NOM;

в) не більше однієї, яка реалізується через порт Modbus Plus, що на ЦПУ.

**15. Яке призначення мають два кругові перемикачі на задній панелі модуля розширення мережі Modbus Plus?**

а) для установлення параметрів зв'язку порту Modbus (RS-232);

б) кругові перемикачі не використовуються для мережі Modbus Plus;

в) використовуються для встановлення адреси вузла в мережі Modbus Plus.

**16. Яка конструкція пристроїв серії Momentum?**

- а) модульна;
- б) блокова;
- в) розподілена.

**17. В якому діапазоні призначаються адреси вузлів мережі DIO?**

- а) від 0 до 128;
- б) від 1 до 64;
- в) від 1 до 32.

**18. До чого підключаються адаптери зв'язку і процесорні адаптери системи Momentum?**

- а) до шин базових блоків вводу-виводу Momentum;
- б) до спеціалізованих базових блоків вводу-виводу Momentum;
- в) до базових блоків вводу-виводу Momentum.

**19. Яку кількість модулів можна встановити на найбільшу монтажну панель контролерів серії Modicon TSX Quantum?**

- а) 32 модуля;
- б) 64 модуля;
- в) 16 модулів.

**20. Яку функцію виконує базовий блок 170 ANM 050 10?**

- а) цей блок є інтерфейсом SERIPLEX для контролерів Momentum і забезпечує обмін даними вводу-виводу по мережі компонентів SERIPLEX;
- б) цей блок є інтерфейсом Ethernet для контролерів Momentum і забезпечує обмін даними вводу-виводу по мережі Ethernet TCP/IP;
- в) цей блок можна використовувати як базовий блок вводу-виводу для програмованого контролера, як в автономній, так і в розподіленій конфігурації вводу-виводу.

*Перевірте відповіді на сторінках: 1 - 195 с., 2 - 197 с., 3 - 199 с., 4 - 199 с., 5 - 200 с., 6 - 200 с., 7 - 200 с., 8 - 201 с., 9 - 201...202 с., 10 - 204 с., 11 - 204 с., 12 - 205 с., 13 - 206 с., 14 - 207 с., 15 - 207 с., 16 - 211 с., 17 - 208 с., 18 - 211 с., 19 - 196 с., 20 - 212 с.*

**III-й рівень запитань (контрольні запитання) дозволяє перевірити рівень засвоєння матеріалу із систематизацією отриманої у розділі інформації та передбачає відповіді, що побудовані на логічному аналізі або порівнянні із приведенням конкретних технічних параметрів (даних або прикладів).**

1. Накреслить загальну структурну схему контролера Modicon TSX Quantum з використанням максимальної кількості різних типів модулів, що забезпечують його повнофункціональність.
2. Охарактеризуйте види конфігурацій вводу-виводу в контролерах Modicon TSX Quantum за їх призначенням та апаратною реалізацією.
3. Чим відрізняється вузол DIO від вузла RIO?

4. Роз'ясніть призначення ключів, перемикачів та інших видів пристроїв, що дозволяють змінювати параметри процесорів, модулів та ін.

5. Які мережні технології і за допомогою яких засобів реалізовано у контролерах Modicon TSX Quantum?

6. Якими особливості монтажу модулів та зовнішніх пристроїв має контролер Modicon TSX Quantum?

7. Доведіть який з контролерів, що розглянуті у розділах посібнику 3 – 6 найбільш близький до контролерів Modicon TSX Quantum за технічними характеристиками, конфігурацією, функціями та організацією мереж.

8. Охарактеризуйте контролери Modicon TSX Quantum з точки зору їх програмування.

9. Які засоби застосовуються для підвищення надійності роботи контролерів Modicon TSX Quantum?

10. Виконайте порівняння методик створення проекту у системах програмування Concept та Unity PRO XL?

*Перевірте відповіді на сторінках: 1 – 195...214 с., 2 – 203...208 с., 3 - 204-208 с., 4 – 195...214 с., 5 – 210...214 с., 6 – 196...214 с., 7 – 195...214с., 54...66с., 80...94с., 108...114с., 124...127с., 145...153с., 8 – 214...222 с., 9 – 200...201с., 209...210 с., 10 – 214...227с.*



## 9 ПРОГРАМОВАНІ КОНТРОЛЕРИ КОРПОРАЦІЇ OMRON

Японська компанія OMRON заснована у 1933 році в місті Кіото, а до початку 1955 р. вона затвердилася, як один зі світових виробників електронних компонентів для управління. В даний час OMRON є міжнародною корпорацією і визнаним світовим лідером у розробці і виробництві всіх необхідних компонентів для створення систем АСУТП. Значну частку у виробничій програмі фірми OMRON займає продукція, яка орієнтована на середній і нижній рівні ієрархії систем АСУ ТП. У цьому секторі фірма пропонує найсучасніше устаткування, у тому числі і програмовані логічні контролери.

ПЛК корпорації OMRON діляться на чотири класи - Micro (мікро), Small (малі), Medium (середні) і Large (великі). Серед представників двох останніх класів найвідомішими є контролери серій SYSMAC CS1 і SYSMAC CJ1. Наявність в їх складі великої різноманітності модулів вводу-виводу і модулів спеціального призначення, а також двопроцесорних пристроїв, з функцією контролю циклу, дозволяє назвати їх універсальними програмованими контролерами, призначеними для розв'язання широкого кола задач підвищеної складності, у тому числі і створення відмовостійких систем для складних об'єктів.

Програмуються контролери мовою крокових діаграм і структурованого тексту стандарту IEC 61131-3 за допомогою пакету програмування CX-One.

### 9.1 Модульні програмовані контролери SYSMAC CS1

Програмовані логічні контролери серії CS1, рис.9.1, це найчисельніше сімейство програмованих контролерів компанії Omron. Вони здатні обробляти 5120 точок вводу-виводу, мають збільшений об'єм пам'яті (448000 слів), високу

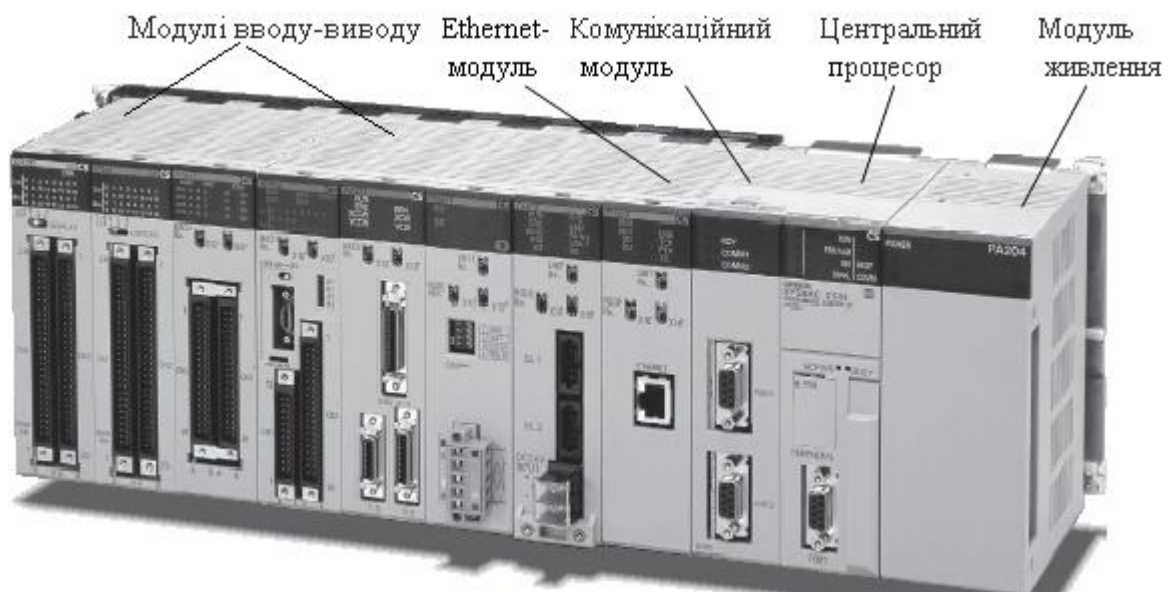


Рисунок 9.1 – Програмований логічний контролер серії SYSMAC CS1

швидкість виконання операцій (0,04 мкс), володіють здатністю створення макропротоколів для одночасної підтримки декількох портів, а також здатні здійснювати безперервний комунікаційний обмін через три рівні мережного обміну.

OMRON CS1 має понад 200 моделей різноманітних модулів дискретних і аналогових входів-виходів, модулів регулювання і модулів зв'язку. З контролерами серії CS1 також використовується широка різноманітність модулів серії C200H, що випускалися для попередніх моделей ПЛК.

Підвищення надійності системи автоматичного управління досягається за рахунок застосування здвоєних центральних процесорів і здвоєних джерел живлення. Можливість "гарячої" заміни модулів ЦПУ, джерел живлення і модулів вводу-виводу дозволяє здійснювати обслуговування системи практично без переривання технологічного процесу.

Контролери CS1 можуть виконувати додаткові, не властиві багатьом іншим контролерам функції - здійснювати розширену обробку даних і архівацію. Нові інструкції дозволяють обробляти файловою пам'ять, текстові рядки, індексні реєстри і багато інших. Як базова концепція запропоноване структурне багатозадачне програмування (Task Programming).

Габаритні розміри (висота × ширина) - 130 × 123 мм.

### **9.1.1 Базова конфігурація контролера**

Конфігурація контролерів серії CS1 складається з панелі центрального процесора і при необхідності панелі розширення. Панель центрального процесора зображає собою базову панель центрального процесора з установленими на неї блоком живлення, модулем центрального процесора, базовими модулями вводу-виводу, спеціальними модулями і модулями шини центрального процесора. Базова панель центрального процесора, рис.9.2, може мати 2 (CS1W-BC023), 3 (CS1W-BC033), 5 (CS1W-BC053), 8 (CS1W-BC083) або 10 (CS1W-BC0103) слотів з рознімами для установлення модулів, а також містить розніми для підключення блока живлення, модуля центрального процесора і панелі розширення.

Для збільшення кількості модулів в контролері використовують панелі розширення, які компонуються на базовій панелі. Існує два типи базових панелей розширення – панель CS1 і панель C200H. Базова панель розширення CS1 аналогічна панелі центрального процесора, проте має конструкції тільки для установлення 3 (CS1W-BI033), 5 (CS1W-BI053), 8 (CS1W-BI083) або 10 (CS1W-BI0103) модулів. Крім того на ній розташований рознім для підключення блока живлення, а також два розніми для підключення за допомогою кабелів панелі центрального процесора і іншої панелі розширення. Панелі розширення C200H також випускають на 3,5,8 і 10 модулів вводу-виводу, але до базової панелі центрального процесора може підключатися тільки панель розширення CS1. Подальше збільшення конфігурації контролера може відбуватися як за допомогою панелі розширення CS1, так і C200H. Кріпляться базові панелі центрального процесора і розширення на DIN-рейці.

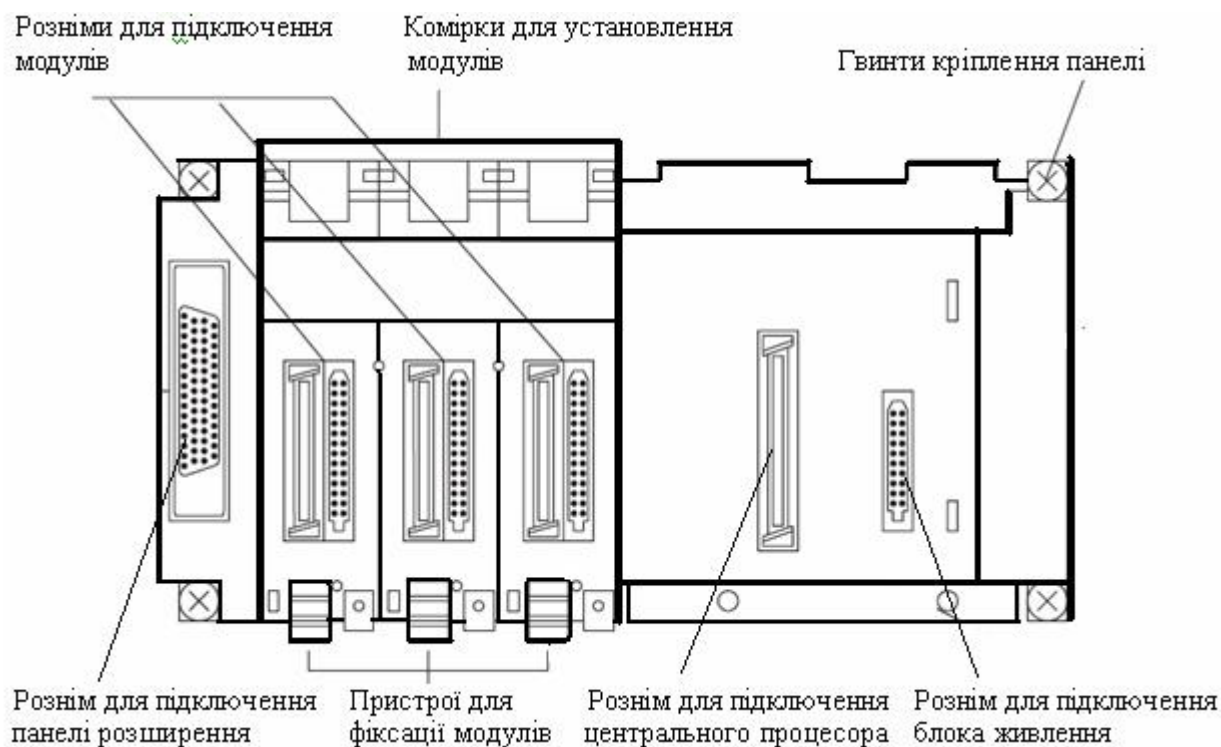


Рисунок 9.2 – Базова панель центрального процесора CS1W-BC033

Для з'єднання базових панелей випускається три типи сполучних кабелів – CS1→ S1, CS1→ C200H, C200H→ C200H. Кабель типа CS1→ CS1 є відрізком проводу, на кінцях якого прості розніми, що захищуються. Він застосовується для з'єднання панелі ЦПУ з панеллю розширення CS1, а також для з'єднання між собою панелей розширення CS1.

Кабель типа CS1→ C200H є відрізком проводу, на одному кінці якого простий рознім, що захищується, а на іншому рознім з двома кріпильними гвинтами. Він застосовується для з'єднання панелі розширення CS1 з панеллю розширення C200H.

Кабель типа C200H→ C200H є відрізком проводу, на кінцях якого рознім з двома кріпильними гвинтами. Він застосовується для з'єднання панелей розширення C200H.

Сполучні кабелі випускаються довжиною – 0,3 м, 0,7 м, 2 м, 3 м, 5 м, 10 м, 12 м. Загальна протяжність сполучних кабелів між панелями розширення не повинна перевищувати 50 м.

До однієї панелі центрального процесора може бути приєднано до семи панелей розширення, що дозволяє довести загальну кількість модулів вводу-виводу в системі до 80. При цьому панелі розширення не можуть підключатися до панелі центрального процесора, яка має два слоти.

Щоб установити на базову панель модулі центрального процесора CS1, блок живлення, базові модулі вводу-виводу, спеціальні модулі, модулі шини центрального процесора, необхідно спочатку навісити верхню частину модуля на відповідне місце базової панелі, а потім затягнути гвинти в нижній її частині, рис. 9.3. При видаленні модулів необхідно ослабити гвинт кріплення в нижній частині модуля і повернути модуль у гору.

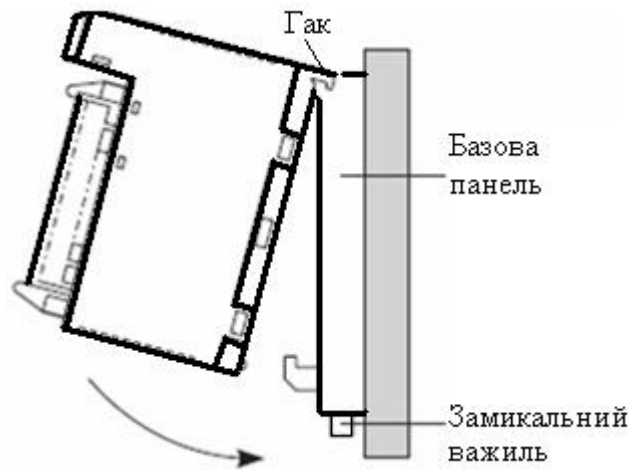


Рисунок 9.3 – Установлення модуля на панель

Для виконання базового комунікаційного обміну центрального процесора з віддаленими вводами-виводами використовуються панелі Slave-модулів SYSMAC BUS (C200H-BC101-V2 , C200H-BC081-V2 і C200H-BC051-V2 ), в які допускається встановлювати до 10, 8 і 5 базових і спеціальних модулів серії C200H. Панелі Slave -модулей SYSMAC BUS дозволяють здійснювати обмін порівняно невеликим об'ємом даних (максимум 512 біт). Підключення базових панелей до панелей Slave -модулей SYSMAC BUS здійснюється за допомогою Master і Slave модулів витою парою або оптоволоконним кабелем.

До одного віддаленого Master-модуля вводу-виводу SYSMAC BUS може підключатися до п'яти панелей Slave - модулів SYSMAC BUS. Панелі Slave -модулів можуть підключатися до панелей розширення модулів вводу-виводу C200H за допомогою кабелів сполучення вводу-виводу C200H [41].

### 9.1.2 Модулі центрального процесора

Модулі центрального процесора серії CS1 здійснюють обмін даними з базовими модулями вводу-виводу, спеціальними модулями і модулями шини центрального процесора. Всього випускається 9 моделей центральних процесорів, які розрізняються кількістю панелей розширення, що входять до складу ПЛК, кількістю входів-виходів, які оброблюються контролером, видом і об'ємом пам'яті, а також швидкістю обробки команд. CPU контролера можна умовно поділити на дві групи - CS1H-CPUxx-E і CS1G-CPUxx-E, схожість і відмінності яких приведені у табл. 9.1.

Не залежно від модифікації всі моделі центрального процесора мають рознім для встановлення картки Flash-пам'яті, периферійний і RS-232C порти, а також слот для встановлення плати послідовного інтерфейсу або плати циклічного управління з безпосереднім підключенням до шини ЦПУ.

Модулі центрального процесора можуть працювати в режимах:

- програмування - використовується для редагування програми або виконання інших підготовчих операцій (зміна налаштувань ПЛК, завантаження і перевірка програми та ін.);

- монітора - використовується для виконання пробних запусків та інших налагоджувальних операцій (редагування в режимі Online, зміна значень в пам'яті вводу-виводу та ін.);

- виконання - використовується для звичайного виконання програми.

Модулі центрального процесора серії CS1 обробляють дані в послідовних циклах, тривалість яких не перевищує 1,45 мкс.

Таблиця 9.1                      Характеристики модулів центрального процесора.

| Модель       | Біти вводу-виводу                          | Об'єм програми | Пам'ять даних | Швидкість обробки команд |
|--------------|--|----------------|---------------|--------------------------|
| CS1H-CPU67-E | 5120 бітів<br>До 7 панелей<br>розширення   | 260К кроків    | 448К слів     | 0.04 мкс                 |
| CS1H-CPU66-E |  | 120К кроків    | 256К слів     |                          |
| CS1H-CPU65-E |  | 60К кроків     | 128К слів     |                          |
| CS1H-CPU64-E |  | 30К кроків     | 64К слів      |                          |
| CS1H-CPU63-E |  | 20К кроків     | 32К слів      |                          |
| CS1G-CPU45-E | 5120 бітів<br>До 7 панелей<br>розширення   | 60К кроків     | 128К слів     | 0.08 мкс                 |
| CS1G-CPU44-E | 1280 бітів<br>До 3-х панелей<br>розширення | 30К кроків     | 64К слів      |                          |
| CS1G-CPU43-E | 980 бітів                                  | 20К кроків     | 32К слів      |                          |
| CS1G-CPU42-E | До 2-х панелей<br>розширення               | 10К кроків     | 32К слів      |                          |

Пам'ять програмованих контролерів серії CS1 складається з блока пам'яті і блока пам'яті користувача. Блок пам'яті має область запам'ятовування параметрів, де зберігається системна інформація центрального процесора, і область пам'яті входів -виходів. Блок пам'яті користувача призначений для розробленої програми управління.

Блок пам'яті і блок пам'яті користувача мають резервне живлення, яке забезпечує набір батарей CS1 W-BAT01. Дані в цих областях пам'яті можуть бути втрачені при зниженні напруги живлення батареї.

Всі програми користувача, вміст області пам'яті вводу-виводу, а також область параметрів можуть зберігатися і оброблятися у вигляді файлів. Для них призначена частина області пам'яті ЕМ і плата пам'яті. ЕМ пам'ять файлів має тип оперативної пам'яті (RAM) об'ємом до 832 Кбайт, а плата пам'яті - Flash-пам'ять об'ємом від 8 до 48 Мбайт залежно від моделі. Плата пам'яті може перезаписуватися до 100 000 разів.

Окрім базових моделей ЦПУ CS1H/G до складу серії входить модель CS1D, яка призначена для підвищення надійності систем автоматизації і, яка підтримує "гарячу" заміну модулів вводу-виводу. CS1D забезпечує безвідмовне функціонування системи за рахунок здвоєних центральних процесорів, які безперервно стежать один за одним і дублюють один одного при виникненні

збоїв. Можливість "гарячої" заміни модулів ЦПУ, джерел живлення і модулів вводу-виводу дозволяє здійснювати обслуговування системи без переривання технологічного процесу.

Зовнішній вигляд модуля центрального процесора серії CS1H приведений на рис. 9.4.

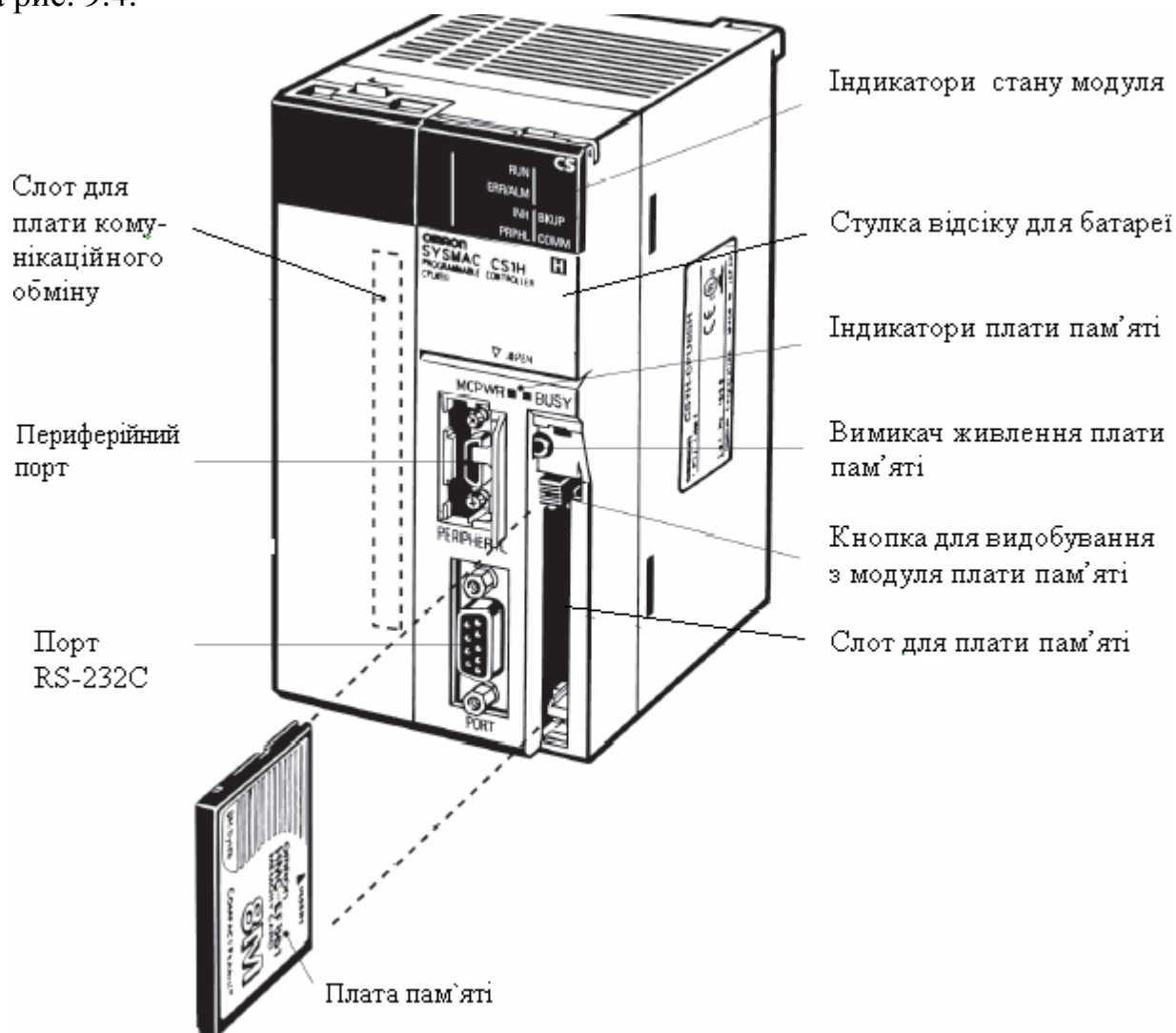


Рисунок 9.4 – Зовнішній вигляд модуля центрального процесора серії CS1H

Зверху на передній панелі модуля центрального процесора розташовані 7 світлодіодних індикаторів, які сигналізують різними кольорами про стан контролера, режим його роботи, доступ до плати пам'яті, наявність напруги живлення на платі пам'яті та комунікаційному обміну через відповідні порти.

У спеціальному відсіку, який закривається стулкою, встановлюється батарея резервного живлення. Там же знаходяться 8 двопозиційних DIP-перемикачів для установлення основних робочих параметрів модуля (запис програми користувача в пам'ять, її автоматичний запуск, вибір мови повідомлень на пульт програмування, використання комунікаційних портів).

Плата розширення пам'яті встановлюється і фіксується в спеціальному слоті з рознімом.

Індикатор плати пам'яті MCPWR миготить зеленим світлом, коли на плату подається напруга живлення, а індикатор BUSY миготить оранжевим світлом при здійсненні доступу до плати пам'яті.

Вимикач живлення плати пам'яті призначений для зняття з плати напруги перед витяганням її із спеціального слота. При натисканні кнопки, розташованої під вимикачем живлення, плата пам'яті звільняється із затиску слота і стає доступною для витягання.

Порт RS-232C призначений для підключення центрального процесора до пристроїв програмування (окрім пульта програмування), головного комп'ютера або зовнішніх пристроїв загального призначення.

За допомогою периферійного порту до CPU може підключатися пульт програмування або головний комп'ютер.

Зліва на передній панелі модуля центрального процесора під стулкою знаходиться слот з рознімом для підключення вбудованої плати послідовного комунікаційного обміну [41].

### 9.1.3 Блоки живлення

Блоки живлення призначені для подачі напруги на модулі контролера. У кожній базовій панелі центрального процесора і розширення повинен бути встановлений блок живлення, розрахований на споживаний струм. Пропонуються різні типи блоків живлення (C200HW-PA204, C200HW-PA204S, C200HW-PA204R, C200HW-PA209R і C200HW-PD024), рис.9.5, які відрізняються

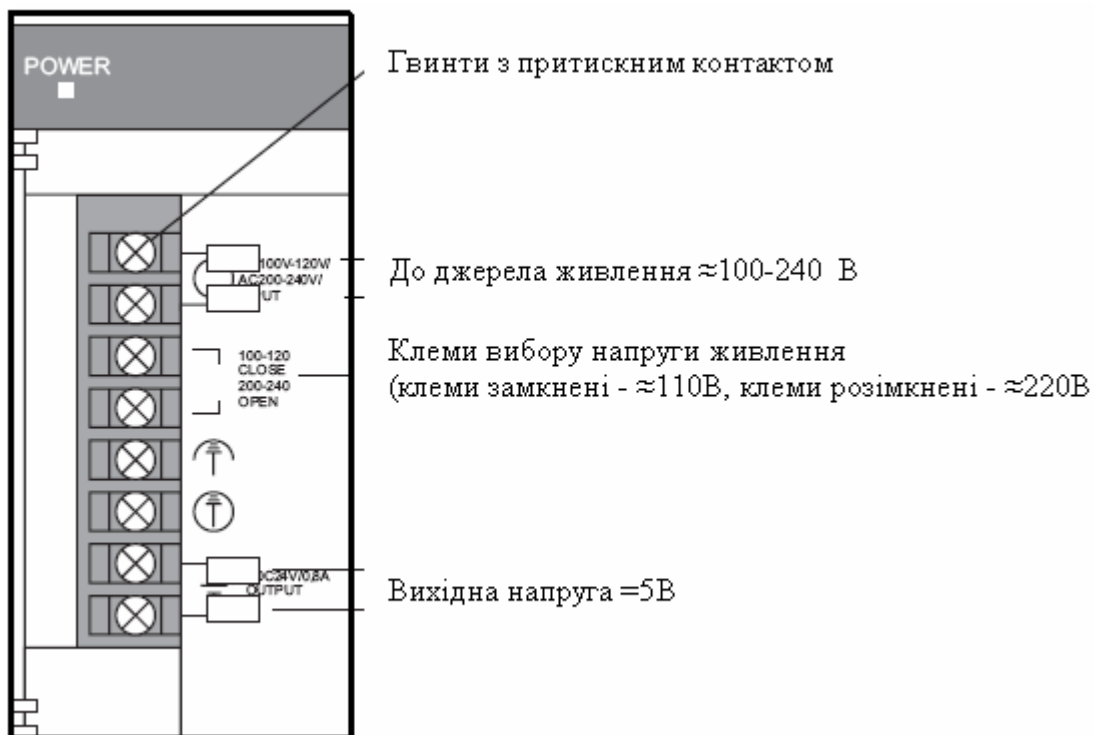


Рисунок 9.5 – Лицева панель блоків живлення C200HW-PA204R і C200HW-PA209R

напругою живлення змінного (~ 100...120В, ~200...240В) або постійного струму (=24В), потужністю (30, 45Вт), а також значенням струму на виході (4,6А, 9А).

Вибір напруги живлення здійснюється замиканням або розмиканням клем 3 і 4 на лицевій поверхні модуля. Коли клеми замкнені напруга живлення дорівнює ~110В, при розімкнених - ~220В. Кожен блок живлення має індикатор POWER, який спалахує, коли на виході блока з'являється напруга =5В. Вихід блока переводиться у стан ON, коли модуль центрального процесора працює у робочому режимі або режимі монітора. Коли модуль центрального процесора переходить у режим програмування або відбувається критична помилка, вихід блока живлення переводиться у стан OFF.

Блок живлення C200HW-PA204S має додатковий вихід допоміжної напруги =24В постійного струму, який звичайно використовується для подачі напруги на модулі вводу постійного струму. Сумарне споживання потужності по виходах 5В і 24В не повинне перевищувати 30Вт[41].

#### **9.1.4 Базові модулі вводу-виводу CS1**

До базових модулів вводу-виводу відносяться модулі дискретних входів-виходів, які дозволяють реалізувати швидке і надійне послідовне управління, а також модулі інтерфейсного вводу-виводу В7А.

Широкий асортимент базових модулів, від швидкісних входів постійної напруги до релейних виходів, дозволяє гнучко конфігурувати ПЛК CS1 під конкретну задачу управління.

Сімейство CS1 містить модулі з різною кількістю входів-виходів і різними способами підключення. Так, модулі дискретного вводу CS1WID211, CS1WID231, CS1WID261, CS1WID291, працюють з вхідним сигналом =24В і мають відповідно 16, 32, 64 і 96 каналів. Модулі вводу CS1WINT01 і CS1WIDP01 - це 16-канальні модулі переривання і високошвидкісних сигналів напругою =24В. Серед модулів виводу існує група транзисторних модулів CS1WOD211, CS1WOD231, CS1WOD261, CS1WOD291 на струм від 0,1 до 0,5А і група транзисторних модулів CS1WOD212, CS1WOD232, CS1WOD262, CS1WOD292 на вихідний струм 5А. У кожній з цих груп є пристрої на 16, 32, 64 і 96 вихідних каналів. Модулі дискретного вводу/транзисторного виводу CS1WMD261, CS1WMD262, CS1WMD291, CS1WMD292 поєднують в собі можливості вхідних і вихідних каналів дискретного сигналу.

Окрім приведених базових модулів вводу-виводу сімейства CS1 при проектному компонуванні контролерів даної серії можна використовувати ще 47 різних за характеристиками модулів сімейства C200H, які випускалися для попередньої серії програмованих контролерів C200H, C200HS, C200HX/HG/HE і повністю відповідають вимогам контролерів серії CS1.

Модулі інтерфейсного вводу-виводу В7А – C200HB7AI1 (16 входів), C200H0B7AO1 (16 виходів), C200HB7AI2 (32 входи), C200HB7AO2 (32 виходи), C200HB7A21 (16входів/16виходів) і C200HB7A22 (32 входи/32виходи) дозволяють обробляти до 64 сигналів і передавати інформацію по двох



проводах послідовного каналу зв'язку на відстань до 500 м. Настроювання або програмування для цього не потрібне.

До модулів з кількістю входних-вихідних каналів 16 і менше зовнішні проводи можна підключати за допомогою знімних гвинтових клемних колодок. А модулі з більшою кількістю входів-виходів (на 32 і 64 канали) підключаються за допомогою стандартних рознімів на 40 виводів. Для простого підключення кіл до модулів з великою кількістю входів-виходів пропонуються готові кабелі і клеми.

Модулі вводу-виводу забезпечені знімними клемними блоками, які кріпляться за допомогою клямок. Для зняття клемного блока з модуля вводу-виводу не потрібне від'єднання проводів, що підведені до нього[41].

### 9.1.5 Спеціальні модулі CS1

Спеціальні модулі CS1 об'єднують модулі аналогового вводу-виводу і модулі переміщення.

До складу серії SYSMAC CS входять модулі аналогового вводу CS1WAD041-V1, CS1W-AD081-V1 і CS1W-AD161, модулі аналогового виводу CS1W-DA041, CS1W-DA08V і CS1W-DA08C, а також модуль аналогового вводу-виводу CS1W-MAD44.

Модулі аналогового вводу CS1WAD041-V1, CS1W-AD081-V1 і CS1W-AD161 мають відповідно 4, 8 і 16 входів з діапазонами  $-10...+10\text{В}$ ,  $0...10\text{В}$  і  $0...5\text{В}$ ,  $1-5\text{В}$  і  $4...20\text{ мА}$ . У модулях CS1WAD041-V1, CS1W-AD081-V1 канал вводу аналогового сигналу можна використовувати як вхід напруги або як вхід струму. В усіх модулях передбачене масштабування і фільтрація сигналів, а також сигналізація аварій.

Модулі аналогового виводу CS1W-DA041, CS1W-DA08V і CS1W-DA08C мають, відповідно, 4, 8 і 8 виходів, з набором діапазонів вихідних сигналів по напрузі  $-10...+10\text{В}$ ,  $0...10\text{В}$ ,  $1...5\text{В}$ ,  $0...5\text{В}$  (CS1W-DA041, CS1W-DA08V), і по струму  $4...20\text{ мА}$  (CS1W-DA041, CS1W-DA08C).

Модуль аналогового вводу-виводу CS1W-MAD44 має 4 входи і 4 виходи з діапазонами сигналів аналогічними модулям вводу і виводу.

В усіх модулях вводу-виводу кожен канал гальванічно розв'язаний від решти каналів.

Окрім перелічених до групи спеціальних модулів входять ще 56 модулів контролерів серії С200Н. Серед них не тільки модулі аналогового вводу-виводу, а і модулі керування температурою, нагрівом/охолодженням, модулі датчиків, ПІД-регулювання, позиціонування, нечіткої логіки та ін.

При використанні модулів аналогового вводу здійснюються спеціальні настроювання при їх звичайному режимі роботи. За допомогою перемикачів, що знаходяться під знімним клемним блоком, по кожному каналу вибирається необхідний тип сигналу (напруга/струм). Після встановлення модуля у стійку йому присвоюється номер (у межах 0-95) за допомогою поворотних перемикачів на лицевій панелі модуля і програмно здійснюється настроювання діапазонів входних сигналів. Номер потрібен для того, щоб в пам'яті

центрального процесора визначилась адреса слів області модулів вводу-виводу, через яку CPU обмінюється даними з модулем аналогового вводу.

При необхідності по кожному каналу модуля вводу-виводу можна виконати регулювання коефіцієнта підсилення та зміщення сигналу. Здійснюється це у режимі регулювання, який встановлюється за допомогою перемикача на задній панелі модуля.

При використанні модулів аналогового виводу їм також присвоюється номер за допомогою поворотних перемикачів і програмно здійснюється налаштування діапазонів вихідних сигналів.

Стан модулів у процесі роботи відображають чотири індикатори у верхній частині лицевої панелі[41].

### **9.1.6 Модулі шини центрального процесора CS1**

Модулі шини центрального процесора - це модулі сполучення, які забезпечують як стандартні відкриті мережні інтерфейси, так і приватні високошвидкісні мережні рішення. Сполучення між окремими ПЛК або між ПЛК і вищестоящою інформаційною системою може бути організоване за послідовним інтерфейсом, за мережею Ethernet або за простою і зручною мережею Controller Link.

Мережа Controller Link є основною в системах промислової автоматизації з використанням ПЛК корпорації OMRON. Підключення модуля Controller Link до мережі забезпечує встановлення обміну даними між програмованими контролерами і персональними комп'ютерами без спеціального програмування. Крім того, корпорація OMRON розробила протокол FINS (Factory Intelligent Network Service), який дозволяє мати доступ до всіх контролерів в межах трирівневої мережі. Комунікаційний обмін FINS-повідомленнями забезпечує при необхідності самостійне управління і переміщення даних великого об'єму з великою пропускнуною спроможністю по шині центрального процесора і в мережах різних типів.

Зовнішній вигляд лицевій панелі модуля Controller Link (CS1W-CLK21) приведений на рис.9.6. У верхній частині модуля знаходяться індикатори стану модуля, нижче - перемикач номера модуля, який призначається для ідентифікації модуля шини CPU в контролері, два перемикача для встановлення адреси вузла (десятки та одиниці), DIP-перемикачі для встановлення швидкості передачі даних, яка має бути однаковою для всіх вузлів мережі, а також перемикач для увімкнення чи вимкнення кінцевого опору для модулів, що знаходяться на границях мережі. Кінцевий опір - є погоджувальним навантаженням і сприяє придушенню паразитних сигналів і зниженню рівня перешкод.

На панель центрального процесора або на панель розширення CS можна встановити до 4-х модулів Controller Link, проте на панелі розширення C200H модуль Controller Link не можна використовувати.

Якщо декілька модулів шини CPU серії CS одночасно встановлюються на одну панель, максимальна загальна кількість модулів шини CPU серії CS не може перевищувати 16.

**Номер модуля**

|          |                    |
|----------|--------------------|
| Діапазон | Вузли              |
| 01...F   | Усі вузли у мережі |

**Адреса вузла**

|          |                    |
|----------|--------------------|
| Діапазон | Вузли              |
| 01...32  | Усі вузли у мережі |

**Швидкість передачі**

| Перемикачі |            | Швидкість передачі  | Макс. відстань | Вузли                       |
|------------|------------|---------------------|----------------|-----------------------------|
| Перемикач1 | Перемикач2 |                     |                |                             |
| Вимк.      | Вимк.      | 2Мбіт/с             | 500м           | швидкість передачі однакова |
| Вимк.      | Увімк.     | 1Мбіт/с             | 800м           |                             |
| Увімк.     | Вимк.      | 500 Кбіт/с          | 1км            |                             |
| Увімк.     | Увімк.     | Не використовується |                |                             |

Перемикачі 3 і 4 – у положенні «Вимк.»

**Кінцевий опір**

| Перемикач | Кінцевий опір | Вузли  |
|-----------|---------------|--|
| Вимк.     | Не увімкнено  | Кінцевий опір вмикати тільки у вузлах, що розташовані на границях мережі |
| Увімк.    | Увімкнено     |  |

Клемна колодка для кабелю зв'язку

Рисунок 9.6 – Лицева панель модуля CS1W-CLK21

Через контролери CS1 мережу Controller Link можна приєднати до Ethernet- мережі. Для цього на панелі ПЛК встановлюються модуль Controller Link і комунікаційний модуль Ethernet.

Ethernet-модуль CS1W-ETN21 призначений для підключення ПЛК серії CS1 в Ethernet-мережу 100Base-TX, а CS1W-ETN01- в Ethernet-мережу 100Base-5. На лицевій панелі Ethernet-модуля (вузла мережі) знаходяться індикатори робочого стану модуля, перемикачі номера модуля і адреси вузла, а також гніздо для підключення до Ethernet. Номер модуля (в діапазоні 0-F) призначається для ідентифікації окремих модулів шини CPU, які

використовуються в складі одного ПЛК. Для встановленого номера автоматично резервуються адреси слів в області СІО і в області DM. Ethernet-модуль використовує ці слова для прийому даних управління від модуля CPU, а також для повідомлення центральному процесору про стан Ethernet-модуля і зв'язку.

Вузли в Ethernet-мережі ідентифікуються за IP-адресами, які встановлюються за допомогою восьми перемикачів на задній панелі модуля.

Коли використовується комунікаційний протокол FINS і до мережі підключено декілька Ethernet-модулів, то вони ідентифікуються за адресою вузла, яка встановлюється в діапазоні 01-7E.

Ethernet-модулі можна встановлювати в будь-які слоти базової центральної процесорної панелі або базової панелі розширення, але кількість слотів, в які вони можуть бути встановлені, залежить від об'єднувальної шини (задньої шини). В один ПЛК може бути встановлено до чотирьох Ethernet-модулів. Якщо разом з Ethernet-модулями встановлюються інші модулі шини CPU (наприклад, модулі Controller Link), то в цьому випадку може бути встановлено не більше 16 модулів шини CPU.

Ethernet-мережа створюється за допомогою спеціального екранованого (STP) або неекранованого (UTP) кабелю (витої пари) з модульним штекером RJ45 з обох боків, а також концентратора (хаба) - пристрою ретрансляції, об'єднуючого в інформаційних мережах декілька вузлів.

Наявність різноманітних протоколів дозволяє створювати широкий спектр додатків, що використовують мережу Ethernet. За допомогою протоколів TCP/IP або UDP/IP можна передавати і приймати дані, а протоколи SMTP/POP3 використовуються для передачі і прийому електронної пошти. Автоматичне коректування вбудованого годинника ПЛК здійснюється за допомогою протоколу SNTP. Завдяки вбудованій в Ethernet-модуль функції FTP-сервера, файли даних, що зберігаються в ПЛК, можуть бути прочитані з робочої станції або комп'ютера, як FTP-клієнтів, і навпаки. Використовуючи комунікаційний протокол FINS, з інформаційної мережі Ethernet можна зв'язуватися з пристроями в мережі Controller Link, що відноситься до рівня систем управління. Завдяки цьому ПЛК в мережі Controller Link можуть контролюватися з ПЛК в мережі Ethernet, а також ці ПЛК можуть обмінюватися між собою даними.

Комунікаційний обмін на основі послідовного інтерфейсу може бути реалізований за допомогою портів, вбудованих в модулі центрального процесора (периферійний порт і порт RS-232C), двох портів RS-232C плати послідовного комунікаційного обміну CS1W-SCB21 або портів RS-232C і RS-422/485 плати послідовного комунікаційного обміну CS1W-SCB41, портів модуля послідовного комунікаційного обміну (два- RS-232C) і портів модуля ASCII (два- RS-232C або RS-422/485). До одного модуля центрального процесора допускається підключати до 16 модулів послідовного комунікаційного обміну і до 16 модулів ASCII.

Для обміну даними вводу-виводу на польовому рівні компанія Omron підтримує дві основні мережі - DeviceNet і PROFIBUS-DP, а для підтримки високошвидкісного обміну пропонує власну мережу CompoBus/S, яка характеризується простотою монтажу і настроювання.

Модуль провідного пристрою PROFIBUS-DP Master (CS1W-PRM21) – це модуль шини CPU, який може бути встановлений на базову плату ПЛК CS1. Модуль підпорядкованого пристрою PROFIBUS-DP Slave (C200HW-PRT21)- це модуль для використання на платі розширення ПЛК серії CS1.

На лицевій панелі модуля CS1W-PRM21 – провідного пристрою PROFIBUS-DP, є 7 світлодіодів, які індицирують режими роботи, а також стан модуля і мережі PROFIBUS, перемикач номера модуля і рознім для підключення мережі до PROFIBUS-DP. Номер модуля ідентифікує окремі модулі шини CPU у тому випадку, коли в ПЛК встановлено декілька модулів шини CPU.

Для кожного модуля провідного пристрою PROFIBUS-DP в області пам'яті CIO відводиться 25 слів управління і стану. Розташування слів визначається номером модуля. За допомогою бітів управління з програми ПЛК можна перемикати режими роботи модуля, які відповідають основним режимам роботи PROFIBUS-DP - OFF-LINE (Немає зв'язку), STOP (Стоп), CLEAR (Обнуління) і OPERATE (Робота). Використовуючи слова управління, користувач також може ініціювати передачу повідомлення загального управління (Global-Control) будь-якій групі відомих пристроїв за мережею PROFIBUS. В решті слів області CIO міститься інформація про стан модуля і діагностична інформація самого модуля, мережі PROFIBUS і підпорядкованих пристроїв, призначених модулю провідного пристрою PROFIBUS-DP.

Модуль провідного пристрою PROFIBUS-DP підтримує обмін FINS-повідомленнями з модулем CPU ПЛК, що дозволяє передавати діагностичні дані підпорядкованих пристроїв або протокол помилок. Користувач також може дозволити або заборонити обмін даними з певними підпорядкованими пристроями, призначеними провідному пристрою, завдяки чому їх можна тимчасово виключати з роботи мережі, наприклад, для проведення технічного огляду, уникаючи при цьому формування повідомлень про помилки модулем провідного пристрою PROFIBUS-DP.

Функціонування мережі DeviceNet забезпечують модулі CS1W-DRM21-V1 і C200HW-DRT21. CAN і CANopen комунікацію здійснює модуль C200HW-CORT21-V1, а мережа CompoBus/S працює завдяки використанню, модуля C200HW-SRM21-V1 [42].

### **9.1.7 Розподілення адрес поміж модулями контролера**

Контролери серії CS1 в процесі виконання програми користувача працюють не з фізичними входами-виходами, а з бітами, що відповідають їм в області пам'яті вводу-виводу. Оперативна пам'ять модуля центрального процесора умовно розділена на 3 частини - пам'ять програми користувача, область пам'яті вводу-виводу (CIO) і область параметрів (даних) (DM). В

області пам'яті програми користувача знаходиться програмний додаток контролера. Область пам'яті вводу-виводу (СІО) використовується для обміну центрального процесора даними з іншими модулями. Область параметрів пам'яті містить початкові уставки програмованого контролера, таблиці вводу-виводу, таблиці маршрутизації, а також уставки для модуля шини центрального процесора, які визначаються тільки за допомогою пристроїв програмування.

Розподіл пам'яті базовим модулям вводу-виводу, спеціальним модулям і модулям шини центрального процесора здійснюється різними способами.

Для базових модулів вводу-виводу, встановлених в панелі центрального процесора, слова в області пам'яті розподіляються за порядком розташування модулів – зліва направо. Базовим модулям вводу-виводу розташованим на панелі центрального процесора і панелі розширення CS1 в області пам'яті відповідають слова з адресами (СІО 0000 - СІО 0319). Модулям, що мають від 1 до 16 каналів вводу-виводу, розподіляється 16 бітів, а модулям, що мають від 17 до 32 каналів вводу-виводу, розподіляється 32 біти. Наприклад, модулю дискретного вводу, що має 8 каналів, розподіляється 16 бітів (1 слово), а біти від 00 до 07 цього слова розподіляються для восьми каналів модуля.

Слова вводу-виводу не розподіляються незаповненим коміркам модуля. Розподіл слів незаповненим коміркам можна здійснити за допомогою пристрою програмування.

Номери панелей від 0 до 7 визначаються за чергою, в якій ці панелі підключені сполучними кабелями. Панель центрального процесора завжди має номер 0, а панелі розширення нумеруються від 1 до 7. Номери панелей не можуть змінюватися на послідовність, яка не співпадає з чергою підключення панелей.

Для панелей, в яких перше слово встановлене, модулям розподіляються слова по черзі, в якій ці модулі встановлені в панель (зліва направо), починаючи з СІО 0000. Слова не розподіляються незаповненим коміркам.

В панелях, де адреса першого слова не встановлена, слова розподіляються за порядком номерів панелей (від меншого до більшого), продовжуючи з останнього номера попередньої панелі.

Кожному із спеціальних модулів вводу-виводу згідно із заданим їм номером автоматично розподіляється 10 слів в області з адресою СІО 2000 – СІО 2959. Номер модуля визначається положенням відповідного перемикача на передній панелі. Кожному зі спеціальних модулів в області параметрів пам'яті DM надається 100 слів у діапазоні адрес D20000 – D29599. Ці слова звичайно використовуються для зберігання початкових уставок спеціальних модулів. Спеціальні модулі можуть встановлюватися на панелі центрального процесора і панелі розширення CS1.

Кожному з модулів шини CPU, відповідно до заданих їм номерам, розподіляється по 25 слів в області модулів шини центрального процесора CS1 (СІО 1500 – СІО 1899). Для завдання початкових уставок модулям шини CPU в області пам'яті даних DM розподіляється 100 слів у діапазоні адрес D30000 – 31599.

Модулі шини CPU можуть встановлюватися на панелі центрального процесора або на панелі розширення CS1.

У табл.9.2 наведений приклад розподілу слів вводу-виводу для базових, спеціальних і модулів шини центрального процесора в панелі центрального процесора.

Модулям, розташованим в панелях Slave-модулів, слова в області вводу-виводу не розподіляються. Проте кожній з панелей розподіляється по 25 слів в області панелей Slave-модулів SYSMAC BUS (CIO 3000 – CIO 3049) відповідно до заданим їм номерам (від 0 до 4). Кожній з комірок в панелях Slave-модулів присвоюється одне з десяти слів, що призначене панелі.

Таблиця 9.2 Розподіл слів в панелі центрального процесора для базових і спеціальних модулів, а також модулів шини центрального процесора

| Ко-мірка | Модуль   | Необхідна кількість слів | Розподілювані слова | Номер модуля | Група              |
|----------|--|--------------------------|---------------------|--------------|--------------------|
| 0        | Модуль дискретного вводу C200H-ID212, 16(точок)        | 1                        | CIO0000             | -            | Базовий модуль     |
| 1        | Модуль ASCII C200H-ASC02                               | 10                       | CIO2000...CIO2009   | 0            | Спеціальний модуль |
| 2        | Модуль послідовного комунікаційного обміну C200H-SCU21 | 25                       | CIO1500...CIO1524   | 0            | Модуль шини ЦПУ    |
| 3        | Транзисторний модуль виводу C200H-OD21A , 16 точок     | 1                        | CIO 0001            | -            | Базовий модуль     |
| 4        | Модуль послідовного комунікаційного обміну C200H-SCU21 | 25                       | CIO1525...CIO1549   | 1            | Модуль шини ЦПУ    |
| 5        | Модуль аналогового вводу C200H-AD002                   | 10                       | CIO2020...CIO2019   | 1            | Спеціальний модуль |

Слова розподіляються зліва направо. Одне слово присвоюється кожній з комірок, навіть якщо вона не заповнена. Останні два слова панелі не використовуються, оскільки панель містить не більше 8 комірок.

Відповідну область CIO мають також комунікаційні плати, термінали вводу-виводу, сполучення і т.ін[41].

### 9.1.8 Пристрої програмування

Існує два типи пристроїв програмування, які можуть застосовуватися для створення додатків користувача – це портативний пульт програмування, який

має вигляд калькулятора або СХ-програматор, що працює на персональному комп'ютері в оболонці Windows, рис. 9.7.

Пульт програмування приєднується до периферійного порту CPU, а підключення контролера до комп'ютера здійснюється через порт RS-232 за допомогою кабелів CS1W-CN118 довжиною 0,1м, CS1W-CN226 довжиною 2м, CS1W-CN616 довжиною 6м.

При підключенні до контролера пульта програмування, четвертий DIP-перемикач переводиться у положення OFF. У випадку використання комп'ютера в якості програматора п'ятий DIP-перемикач переводиться у положення ON.

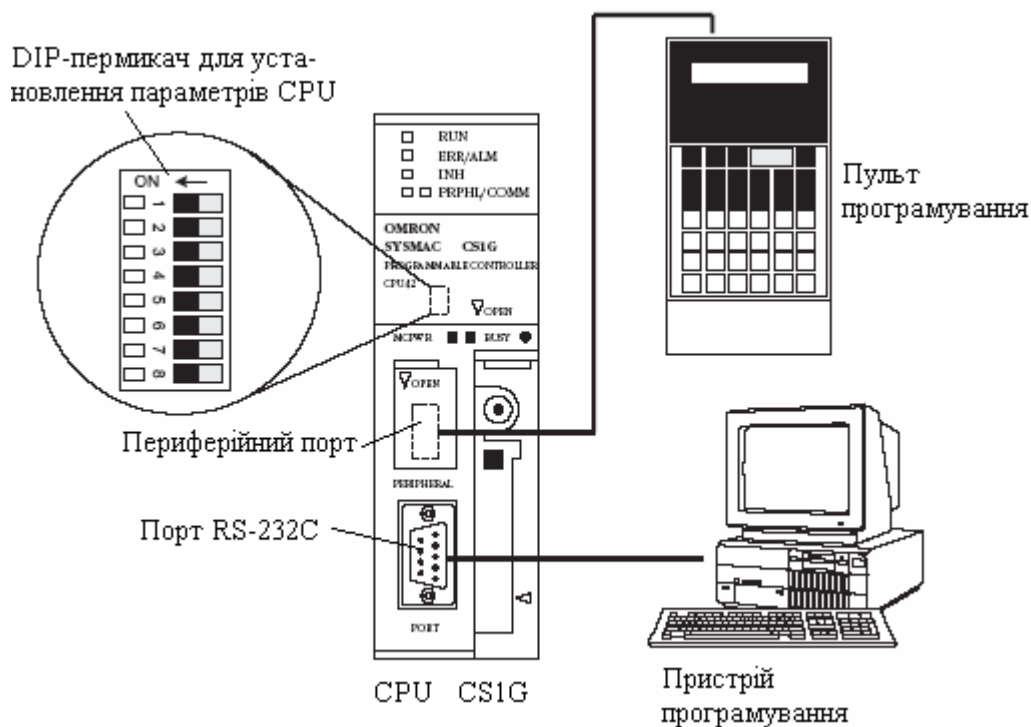


Рисунок 9.7 – Підключення пульта програмування і програматора до центрального процесора

Пульт програмування використовується для зміни режимів роботи, редагування програм, а також здійснення моніторингу обмеженої кількості точок.

СХ-програматор звичайно застосовується для розробки програм, створення таблиць вводу-виводу, моніторингу стану модулів вводу-виводу, причому не тільки для безпосередньо підключеного програмованого контролера, а також і для будь-якого з ПЛК, підключених до ЕОМ через мережі Controller Link або Ethernet[41].

## 9.2 Програмування контролерів OMRON

Корпорацією Omron створений єдиний програмний пакет CX-One, за допомогою якого реалізується мережне програмне забезпечення систем автоматизації, програмування ПЛК і симуляція його роботи, програмування



систем динамічного управління, а також регулювання і комутація. Складова цього пакету - CX-Programmer створює єдину платформу програмного забезпечення для ПЛК Omron всіх типів - від програмованих логічних мікроконтролерів до двопроцесорних систем. CX-Programmer дозволяє програмувати і редагувати додаток користувача, конфігурувати контролери і периферійне устаткування, а також виконує роль інструменту настройки параметрів ПЛК, спеціальних модулів входів-виходів і модулів шини ЦПУ.

ПЛК серії CS1 програмується мовою релейно-контактної логіки або «сходових діаграм», яка відповідає міжнародному стандарту IEC 61131-3. Великий набір команд і бібліотека функцій та функціональних блоків дозволяють створювати дуже компактні програми. За рахунок здатності розділення програми на задачі контролери забезпечують підвищену ефективність їх виконання.

### 9.2.1 Методика створення проекту у системі програмування CX-One

Розробити для контролера CS1 LD-програму керування температурою рідини у технологічному апараті, алгоритм якої надається релейно-контактною схемою, рис.9.8.

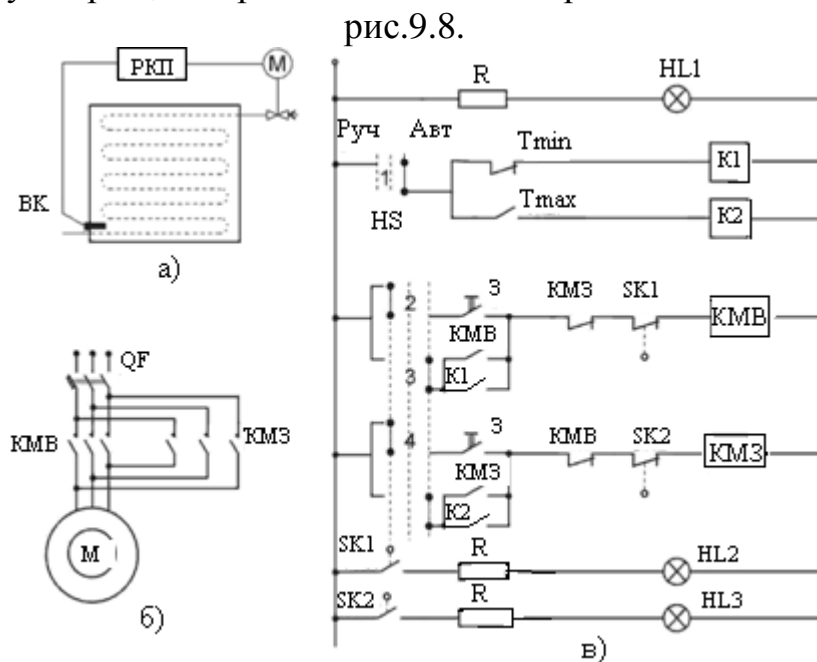


Рисунок 9.8 – Схема автоматичного регулювання температури  
 а) - функціональна; б) – силових кіл виконавчого механізму М;  
 в) – релейно-контактна схема керування двигуном

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 К1 і К2, що використовуються як підсилювальні елементи. Універсальний перемикач SH

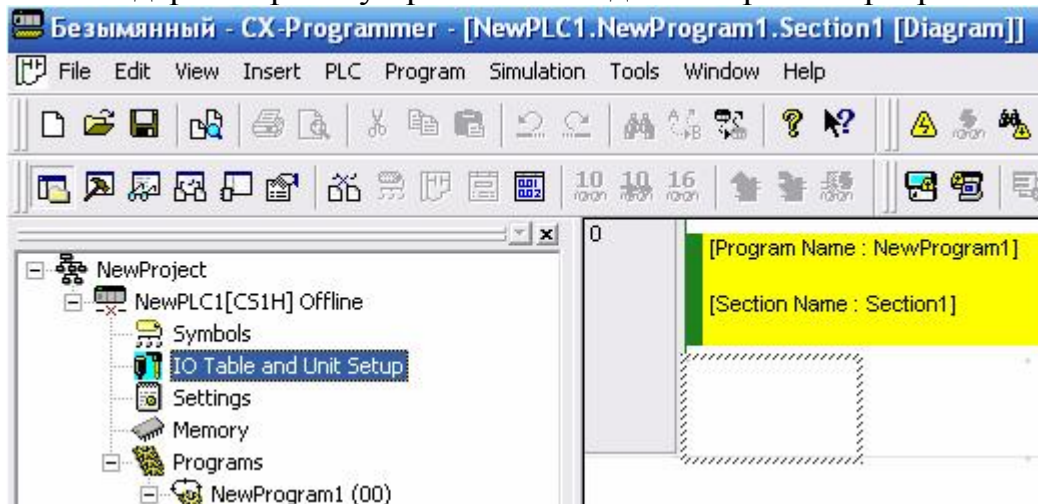
перемикає кола керування з ручного режиму на автоматичний. Кінцеві вимикачі SK1 і SK2 вмикають контактори КМЗ і КМВ при повному закритті або відкритті заслінки і вмикають світлову сигналізацію HL2, HL3. У схемі передбачено електричне блокування, яке виключає одночасне вмикання контакторів КМВ і КМЗ (послідовно з КМВ увімкнений робочий контакт КМЗ, а послідовно з КМЗ - робочий контакт КМВ).

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 К1, яке через контакти 3 перемикача SH, контакт КМЗ і кінцевий вимикач SK1 подає сигнал на вмикання КМВ. 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.

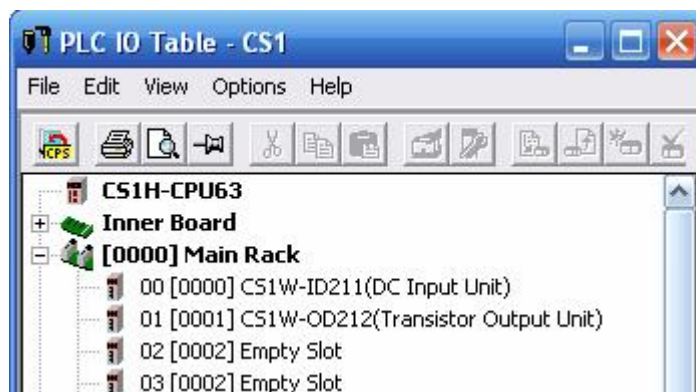
### ***Розроблення проекту***

У пакеті Sx-One V 3.iso запустити систему програмування Sx-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 ОК.

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











Для конфігурування контролера двічі натиснути ліву клавiшу 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 клавiшу 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 забезпечує 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 клав'яшею 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<sub>min</sub> (0.01);
- керувального сигналу на закриття заслінки - Dvig\_Close (0.02);
- кінцевого вимикача - Конс1 (0.03).

Оскільки при підвищенні температури відносно мінімальної уставки контакт датчика температури T<sub>min</sub> розмикається, необхідно передбачити в програмі прямий контакт, який протягом істинного стану вихідної змінної, що керує роботою виконавчого механізму у бік відкриття заслінки Dvig\_Open,

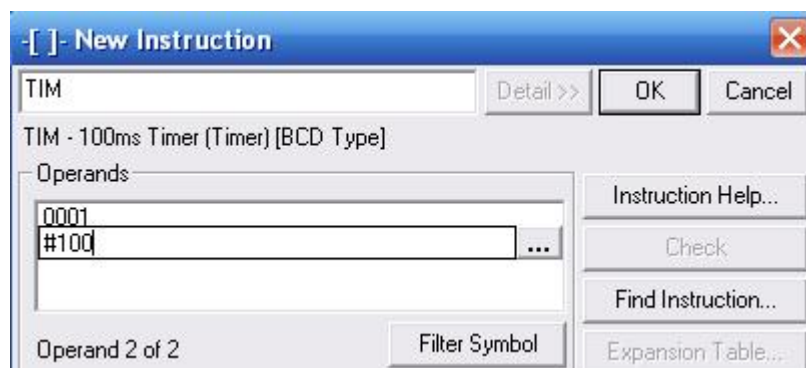
шунтує інвертований контакт  $T_{min}$  у нульовому блоці секції і не дає виконавчому механізму, при автоматичному керуванні, передчасно зупинитися.

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 раніше 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.

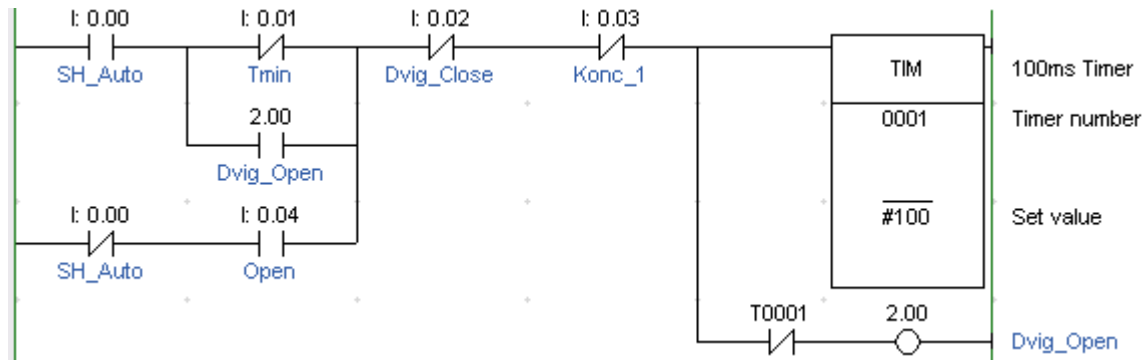
Останнім елементом LD-мережі має бути екземпляр таймера, який забезпечить вимикання виконавчого механізму за 10с роботи. Щоб його установити необхідно навести курсор на кнопку  (New PLC Instruction), клацнути лівою клавiшею миші і повторно натиснути її у місці встановлення блока. У вікні New Instruction 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 час затримки #100. При виборі константи часу, 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 клавiші 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 клав'яшею 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'являться 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 клав'яшею 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, рис.9.9.

В програмі використовуються такі змінні:

- вхідні:

SH\_Auto: BOOL – реагує на стан перемикача режимів роботи Авт/Руч;

$T_{min}$ : BOOL – реагує на електричний контакт манометричного термометра, що спрацьовує при мінімальній температурі рідини;

$T_{max}$ : 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 регульовального органа;

Konc\_1: BOOL-реагує на стан кінцевого вимикача виконавчого механізму при відкритому регульовальному органі;

Konc\_2: BOOL-реагує на стан кінцевого вимикача виконавчого механізму при закритому регульовальному органі;

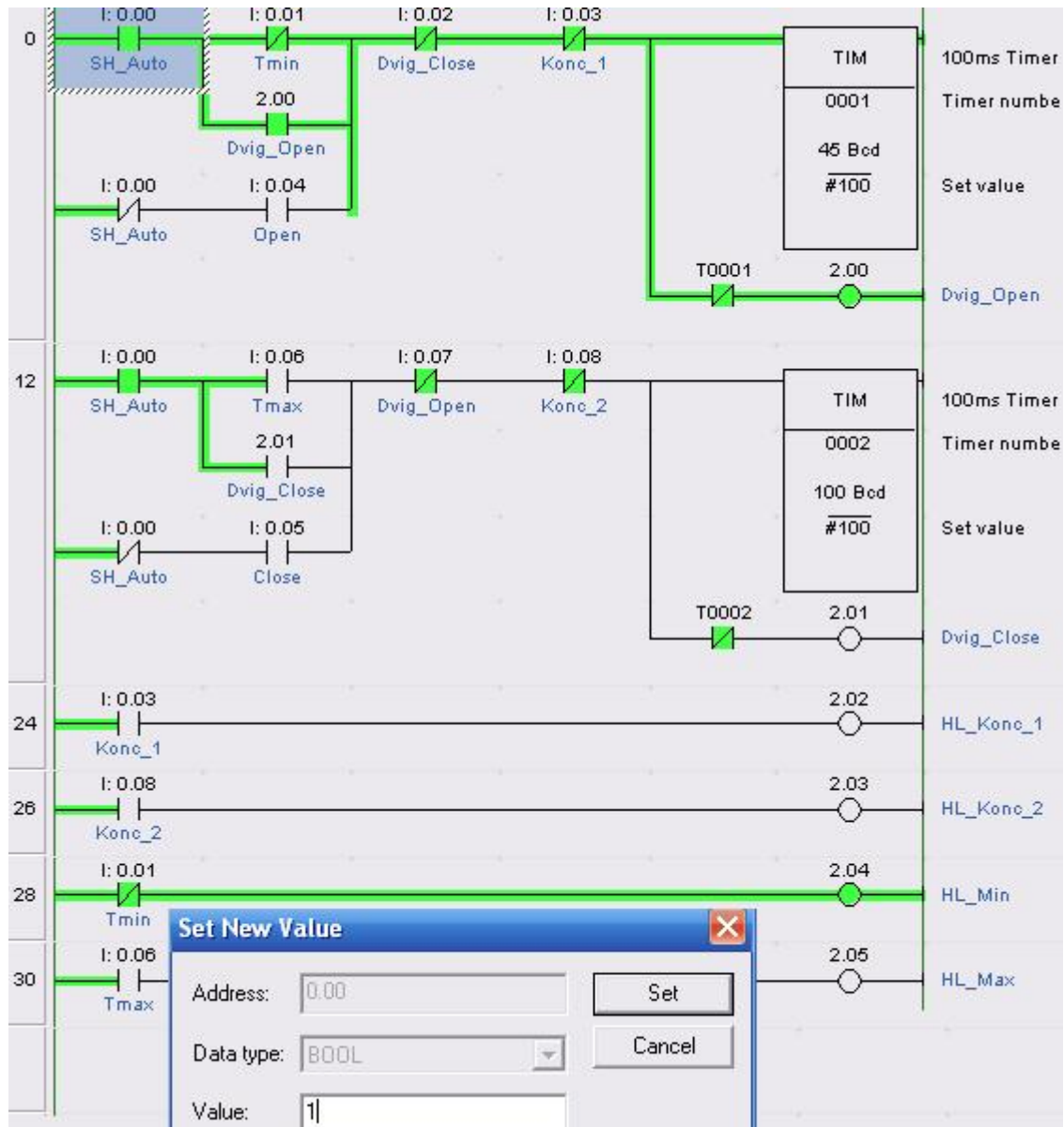


Рисунок 9.9 – Вид екрану при перевірці роботи програми

- вихідні:

Dvig\_Open: BOOL-вмикає виконавчий механізм на відкриття регульовального органа;

Dvig\_Close: BOOL–вмикає виконавчий механізм на закриття регульовального 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[43,44].

## Тестові та контрольні запитання до розділу 9

**I-й рівень запитань та тверджень дозволяє перевірити загальний рівень засвоєння матеріалу розділу і передбачає відповідь на запитання або твердження “Так” чи “Ні”.**

1. Чи здатні ПЛК серії SYSMAC CS1 обробляти 5120 входів-виходів?

2. Тільки CS1G-CPUxx-E група центральних процесорів ПЛК серії CS1 мають рознім для встановлення картки Flash–пам'яті, периферійний і RS-232C порти, слот для встановлення плати послідовного інтерфейсу, плати циклічного управління з безпосереднім підключенням до шини CPU.

3. Пам'ять програмованих контролерів серії SYSMAC CS1 складається з блока пам'яті і блока пам'яті користувача.

4. Чи має центральний процесор серії CS1H порт USB та Ethernet?

5. Блок живлення C200HW-PA204S має додатковий вихід допоміжної напруги 36В постійного струму, який використовується для подачі напруги на датчики або первинні перетворювачі.

6. Чи відносяться модулі дискретного та інтерфейсного вводу-виводу до базових модулів вводу-виводу в контролерах серії CS1?

7. Зовнішні проводи підключаються до модулів на 32 і 64 канали за допомогою знімних гвинтових клемних колодок.

8. Для модулів аналогового вводу здійснюються такі спеціальні налаштування: присвоюється номер модуля за допомогою поворотних перемикачів; для окремого каналу модуля вибирається необхідний тип сигналу



за допомогою двопозиційних перемикачів; програмно настроюються діапазони вхідних сигналів.

9. Номер модуля встановлюється для визначення його у мережі при обміну даними?

10. Чи є Controller Link модифікацією спрощеної версії контролера серії CS1, який використовується тільки для логічного управління технологічними процесами?

11. Чи забезпечує протокол FINS доступ до всіх контролерів в межах трирівневої мережі та в мережах різних типів?

12. В один ПЛК серії CS1 можна встановити до чотирьох Ethernet-модулів.

13. Чи здійснюється в контролерах сімейства CS1 за допомогою портів плат CS1W-SCB21 та CS1W-SCB41 комунікаційний обмін на основі послідовного інтерфейсу?

14. Модуль провідного пристрою PROFIBUS-DP підтримує обмін даними з модулем CPU ПЛК за допомогою протоколу FINS.

15. Область параметрів призначена для зберігання початкових значень змінних, що використовуються у програмі управління.

16. Панель центрального процесора завжди має номер 1, а панелі розширення нумеруються від 0 до 16.

17. Область пам'яті для модулів шини CPU розподіляється так: кожному модулю виділяється по 25 слів в області модулів шини CPU та по 100 слів в області пам'яті даних.

18. Базова панель центрального процесора CS1 містить: блок живлення, модуль центрального процесора, базові модулі вводу-виводу, спеціальні модулі та модулі шини центрального процесора.

19. Чи може загальна протяжність сполучних кабелів між панелями розширення складати понад 100 м?

20. Чи можна до одного віддаленого Master-модуля вводу-виводу SYSMAC BUS підключати чотири панелі Slave - модулів SYSMAC BUS?

*Перевірте відповіді на сторінках: 1 - 233 с., 2 - 236 с., 3 - 237 с., 4 - 238 с., 5 - 240 с., 6 - 240 с., 7 - 241 с., 8 - 241 с., 9 - 241...242 с., 10 - 242 с., 11 - 242 с., 12 - 244 с., 13 - 244 с., 14 - 245 с., 15 - 246 с., 16 - 246 с., 17 - 246 с., 18 - 234 с., 19 - 235 с., 20 - 236 с.*

**II-й рівень запитань (тестові запитання) дозволяє перевірити рівень засвоєння матеріалу розділу із заглибленням у технічні особливості будови та структури контролерів серії SYSMAC CS1 корпорації OMRON та передбачає вибір правильної відповіді з наведених трьох варіантів.**

**1. В яких режимах можуть працювати модулі центрального процесора ПЛК серії CS1?**

а) програмування та виконання;

б) конфігурування, програмування та виконання;

в) програмування, моніторингу та виконання.

**2. Для чого призначений блок пам'яті користувача в програмованих контролерах серії SYSMAC CS1?**

- а) для зберігання розробленої програми управління;
- б) для зберігання налаштувань ПЛК;
- в) для зберігання даних, що надходять з об'єкта управління.

**3. Скільки слотів і для чого розташовані на передній панелі модуля центрального процесора серії CS1H?**

- а) два слоти, один для підключення вбудованої плати послідовного комунікаційного обміну, а інший для плати розширення пам'яті;
- б) три слоти, один для підключення модулю Ethernet, а два для плат розширення пам'яті;
- в) три слоти, один для підключення модулю Ethernet, один для плати розширення пам'яті та один для модуля розширення.

**4. Що необхідно зробити перед витяганням плати пам'яті зі спеціального слота процесора серії CS1H?**

- а) зняти напругу живлення з контролера;
- б) зняти напругу живлення з плати через CX-програмактор;
- в) зняти напругу живлення з плати за допомогою вимикача, що розташований на платі.

**5. Як здійснюється вибір напруги в блоках живлення змінного струму контролерів серії CS1?**

- а) вибір напруги живлення здійснюється двопозиційним перемикачем на модулі центрального процесора;
- б) вибір напруги живлення здійснюється замиканням або розмиканням клем на лицевій поверхні модуля живлення;
- в) вибір напруги живлення здійснюється двопозиційним перемикачем на модулі живлення.

**6. З яким вхідним сигналом працюють модулі дискретного вводу CS1WID211, CS1WID231, CS1WID261 та CS1WID291 і скільки каналів вони мають?**

- а) з вхідним сигналом  $\leq 24\text{В}$  і мають, відповідно, 16, 32, 64 і 96 каналів;
- б) з вхідним сигналом  $\leq 36\text{В}$  і мають, відповідно, 8, 16, 32 і 64 канали;
- в) з вхідним сигналом  $\sim 220\text{В}$  і мають, відповідно, 8, 16 та 32 канали.

**7. Яку максимальну кількість сигналів можуть обробляти модулі інтерфейсного вводу-виводу В7А і на яку відстань передавати інформацію?**

- а) до 32 сигналів і передавати інформацію на відстань до 100 м;
- б) до 64 сигналів і передавати інформацію на відстань до 100 м;
- в) до 64 сигналів і передавати інформацію на відстань до 500 м.

**8. Які модулі сімейства CS1 відносяться до спеціальних модулів?**

- а) модулі аналогового та дискретного вводу-виводу;
- б) модулі аналогового вводу-виводу і модулі переміщення;
- в) модулі інтерфейсного вводу-виводу і модулі переміщення.

**9. До якої групи відноситься модуль CS1W-MAD44 і скільки у нього каналів?**

- а) це спеціальний модуль аналогового вводу-виводу, у якого 4 вхідних і 4 вихідних канали;
- б) це базовий модуль дискретного вводу-виводу, у якого 16 вихідних каналів;
- в) це спеціальний модуль інтерфейсного вводу-виводу, у якого 16 вхідних каналів.

**10. Що відображає стан модулів аналогового вводу-виводу сімейства CS1?**

- а) один індикатор у верхній частині лицевої панелі;
- б) індикатори, що розташовані на лицевій панелі модуля навпроти кожного каналу;
- в) чотири індикатори у верхній частині лицевої панелі.

**11. Яке призначення мають модулі шини центрального процесора CS1?**

- а) забезпечують підключення модулів вводу-виводу до CPU;
- б) забезпечують як стандартні відкриті мережні інтерфейси так і приватні високошвидкісні мережні рішення;
- в) забезпечують підключення ПЛК до мережі Ethernet.

**12. Скільки модулів Controller Link можна встановити на панелі центрального процесора або на панелі розширення CS?**

- а) на кожній з панелей можна встановити до 4-х модулів Controller Link;
- б) на панелі центрального процесора можна встановити до 4-х модулів Controller Link, а на панелі розширення CS – жодного;
- в) на кожній з панелей можна встановити до 16 модулів Controller Link.

**13. Як підключається мережа Controller Link до Ethernet – мережі?**

- а) ці дві мережі не сумісні;
- б) через модулі інтерфейсного вводу-виводу;
- в) через контролер CS1 у якому встановлені модуль Controller Link і комунікаційний модуль Ethernet.

**14. Які мережі підтримують контролери компанії Omron для обміну даними на польовому рівні?**

- а) мережі CAN, PROFIBUS-DP та власну мережу Controller Link;
- б) мережі DeviceNet, PROFIBUS-DP та власну мережу CompoBus/S;
- в) мережі HART, CAN та власну мережу Controller Link.

**15. За допомогою яких модулів забезпечується мережа PROFIBUS-DP?**

- а) за допомогою провідного модуля PROFIBUS-DP Master (CS1W-PRM21) та модуля підпорядкованого пристрою PROFIBUS-DP Slave (C200HW-PRT21);
- б) достатньо тільки провідного модуля PROFIBUS-DP Master (CS1W-PRM21);
- в) достатньо тільки модуля підпорядкованого пристрою PROFIBUS-DP Slave (C200HW-PRT21).

**16. На які частини умовно поділена оперативна пам'ять модуля центрального процесора ПЛК CS1?**

- а) пам'ять програми користувача і область параметрів (даних);
- б) область пам'яті вводу-виводу та Flash-пам'ять;
- в) пам'ять програми користувача, область пам'яті вводу-виводу і область параметрів (даних).

**17. Для чого призначена область пам'яті вводу-виводу (CІO)?**

- а) для обміну центрального процесора даними з іншими модулями;
- б) для обміну центрального процесора даними з програмами управління;
- в) для обміну центрального процесора даними з модулями, які розташовані на панелі розширення.

**18. Як розподіляються слова в області пам'яті для базових модулів вводу-виводу, що встановлені в панелі центрального процесора ?**

- а) слова з адресами CІO 0000 - CІO 0319 розподіляються згідно порядку розташування модулів – зліва направо;
- б) слова з адресами CІO 2000 - CІO 2959 розподіляються автоматично згідно із заданим модулю номером;
- в) слова з адресами CІO 0000 - CІO 2000 розподіляються автоматично згідно із заданим модулю номером.

**19. Як центральний процесор здійснює базовий комунікаційний обмін з віддаленими вводами-виводами?**

- а) за допомогою панелі Master -модулей SYSMAC BUS;
- б) за допомогою панелей Master та Slave -модулей SYSMAC BUS;
- в) за допомогою панелі Slave-модулів SYSMAC BUS.

**20. Які існують пристрої програмування контролерів сімейства CS1?**

- а) програмування контролерів сімейства CS1 здійснюється тільки з EOM;
- б) програмування контролерів сімейства CS1 здійснюється через портативний пульт програмування та СХ-програмактор;
- в) програмування контролерів сімейства CS1 здійснюється через лицеву панель контролера та через та СХ-програмактор.

*Перевірте відповіді на сторінках: 1 – 236...237 с., 2 - 237 с., 3 - 238 с., 4 - 239 с., 5 - 240 с., 6 - 240 с., 7 – 240...241 с., 8 - 241 с., 9 - 241 с., 10 - 242 с., 11 - 242 с., 12 - 242 с., 13 - 243 с., 14 - 245 с., 15 - 245 с., 16 - 245 с., 17 - 246 с., 18 - 246 с., 19 - 236 с., 20 – 247...248 с.*

**ІІІ-й рівень запитань (контрольні запитання) дозволяє перевірити рівень засвоєння матеріалу із систематизацією отриманої у розділі інформації та передбачає відповіді, що побудовані на логічному аналізі або порівнянні із приведенням конкретних технічних параметрів (даних або прикладів).**

1. Якої концепції дотримувалася корпорація OMRON при проектуванні та розробці контролерів серій SYSMAC CS1?

2. Розкрийте особливості роботи модулів центрального процесора.
3. Роз'ясніть поняття базові та спеціальні модулі контролера серії SYSMAC CS1.
4. Виконайте порівняння всього ряду модифікацій аналогових модулів вводу-виводу контролера серії SYSMAC CS1.
5. Скомпонуйте контролер серії SYSMAC CS1 для повнофункціональної роботи та розкрийте принцип розподілу адреси слів для кожного модуля.
6. Які засоби в конструкції контролера та модулів передбачені для зручності роботи з ними та контролю за їх станом?
7. Основні види, можливості та призначення модулів шини центрального процесора контролера серії SYSMAC CS1.
8. Роз'ясніть правила компоновання панелі центрального процесора та панелі розширення.
9. Приведіть перелік кабельної продукції, яка необхідна для здійснення різного роду підключень у контролері серії SYSMAC CS1.
10. Засоби програмування контролера та їх функції.

*Перевірте відповіді на сторінках: 1 – 233...249 с., 2 – 236...239 с., 3 – 240...242 с., 4 – 241...242 с., 5 – 233...247 с., 6 – 233...247 с., 7 – 242...245 с., 8 – 234...237 с., 9 – 234...248 с., 10 – 247...249 с.*

## 10 ПРОГРАМОВАНІ КОНТРОЛЕРИ КОРПОРАЦІЇ ROCKWELL AUTOMATION

Американська корпорація Rockwell Automation, історія якої почалася у 1880 році, є одним з провідних світових виробників і постачальників засобів промислової автоматизації. У 1985 році, тоді ще, компанія Rockwell, що займалася розробками для військово-промислового комплексу і космічної галузі, придбала фірму Allen-Bradley, яка мала досвід виробництва програмованих контролерів, і почала розвивати відділення промислової автоматизації під назвою Rockwell Automation. Надалі компанія придбала ще понад 50 фірм, що працювали в цій області, а власні відділення іншого спрямування - продала. Назви придбаних найвідоміших фірм були збережені в якості торгових марок, під якими випускається відповідна продукція. Серед них:

- Allen-Bradley (промислові контролери, мережі, панелі візуалізації, частотні приводи, датчики, компоненти);
- Rockwell Software (засоби програмування контролерів і розробки SCADA);
- Reliance Electric (частотні приводи і електродвигуни);
- Dodge (інтелектуальні механізми).

В даний час корпорація випускає три сімейства контролерів MicroLogix/SLC (MicroLogix 1000, MicroLogix 1200, MicroLogix 1500, SLC-500), що орієнтовані на розв'язання невеликих задач управління, PLC - 5 – для реалізації задач середньої складності і Logix (ControlLogix, FlexLogix і CompactLogix) - призначених для побудови великих систем управління.

Для програмування ПЛК спеціалістами корпорації Rockwell Automation розроблені інструментальні середовища для всіх сімейств контролерів. Програмування мікропроцесорних пристроїв групи Micrologix і SLC здійснюється за допомогою пакета RSLogix 500 (Rockwell Software Logix), контролерів сімейства PLC-5 за допомогою RS Logix 5, а контролерів ControlLogix – за допомогою RSLogix 5000. Розробка проектів здійснюється мовами релейно-контактних схем, функціональних блоків, структурованого тексту та послідовних функцій, побудованих на основі стандарту IEC 61131-3. Разом з пакетом програмної емуляції RSLogix Emulate код проекту можна запускати і налагоджувати на комп'ютері без використання контролерів.

### 10.1 Контролери ControlLogix

Контролери ControlLogix є найпотужнішими серед тих, що випускаються корпорацією Rockwell Automation, і являють собою широкий набір різноманітних модулів, які поєднуються за допомогою шасі вводу-виводу. ControlLogix забезпечує дискретне управління, управління безперервними процесами, приводами і сервоприводами, в поєднанні з комунікаціями і сучасним вводом-виводом. Відмінною рисою контролерів ControlLogix є

архітектура, що заснована на базі мережі ControlNet. Шасі контролера є сегментом мережі ControlNet, де кожний модуль - це станція. Обмін даними поміж модулями в шасі, як і у мережі ControlNet, здійснюється за принципом постачальник-споживач. Така архітектура забезпечує контролеру декілька переваг:

- усі місця в шасі рівноцінні, завдяки чому модулі вводу-виводу, процесори та інші модулі можуть бути встановлені на будь-яке місце;
- в одне шасі може бути встановлене будь-яка кількість процесорів, що дозволяє створити повноцінну багатопроцесорну систему управління;
- для підключення до «зовнішніх» мереж ControlNet, Ethernet і ін. використовуються мережні модулі.

Кількість мережних модулів в шасі не обмежена. Кожний з них є вузлом мережі і може самостійно, без участі процесора, передавати і приймати інформацію. Встановивши в шасі два адаптери, що сполучаються з різними мережами, можна організувати міст між цими мережами (наприклад, між двома локальними мережами ControlNet, між ControlNet і Ethernet і з ін.). В одному шасі можна організувати декілька мостів, їх кількість обмежується тільки кількістю місць в шасі.

У системі управління на базі мережі ControlNet, процесорів і шасі ControlLogix, кожен модуль є самостійним вузлом мережі, що, зокрема, дозволяє вільно розподіляти процесори по різних шасі системи управління. Можна використовувати декілька процесорів в одному шасі або декілька процесорів, об'єднаних через різні мережі. Один процесор контролера підтримує до 250 віддалених шасі, на яких може бути розміщено по декілька модулів процесора.

Завдяки таким відмінностям і широким можливостям виробники ControlLogix називають цю розробку системою, а окремий процесор – контролером.

Система в мінімальній конфігурації складається з одного автономного контролера (процесора) і модулів вводу-виводу встановлених у шасі, рис.10.1.

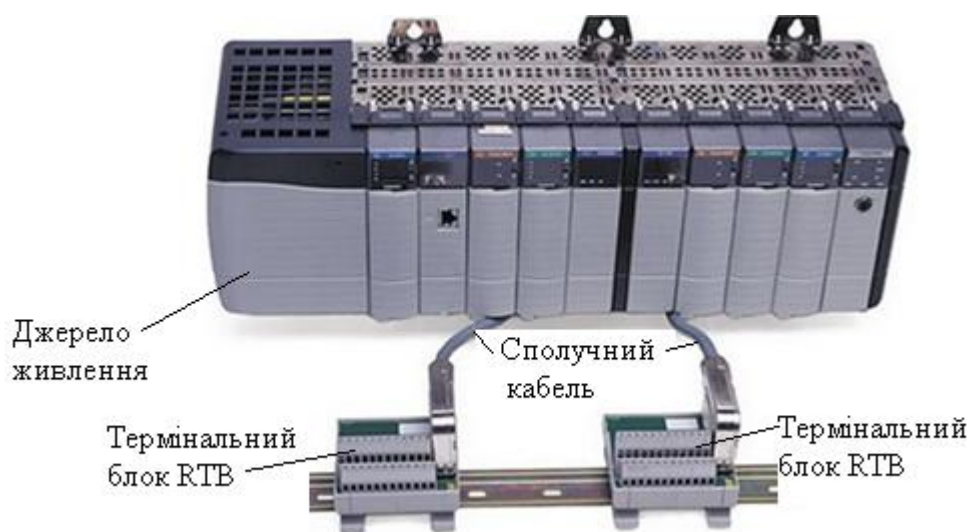


Рисунок 10.1 – Система ControlLogix

Вона здатна управляти як локальними, так і віддаленими входами-виводами із загальною кількістю аналогових каналів – 4000 і дискретних - 128000[45,46].

### **10.1.1 Вибір контролерів**

Контролери (процесори) ControlLogix можна поділити на дві групи – контролери модифікації 1756-L55Mxx (7 різновидів) і контролери 1756-L6x (три різновиди). Перша група контролерів має об'єм призначеної для користувача пам'яті від 750 (1756-L55M12) до 7680 Кбайт (1756-L55M16), а друга від 2048 (1756-L61) до 8192 Кбайт (1756-L63). Об'єм пам'яті вводу-виводу у контролерів другої групи складає 478 Кбайт, що більше ніж у 2 рази перевищує об'єм пам'яті контролерів першої групи.

Контролери 1756-L55M2x мають внутрішню фіксовану незалежну пам'ять, а пам'ять контролерів 1756-L6x підтримується змінною карткою пам'яті CompactFlash.

Контролери групи 1756L6x виконують сканування релейної логіки майже в два рази швидше, ніж контролери групи 1756L55Mxx, а виконання функціональних блоків, інструкцій управління рухом і математичних інструкцій з даними типу REAL – швидше в 4-5 разів.

Особливе місце серед наведених контролерів займає пристрій 1756-L60M03SE, який об'єднує в собі модуль групи 1756L6x і модуль управління рухом. Цей контролер займає два слоти в шасі і є ідеальним для малих систем управління рухом по 3-х осях.

Контролери ControlLogix можуть встановлюватися у будь-який слот шасі. При цьому, в одному шасі може бути декілька незалежно працюючих контролерів, зв'язок між якими, а також контроль і управління вводом-виводом здійснюється через задню шину. Взаємодія з комп'ютерами та іншими контролерами реалізується за інтерфейсами RS-232-C (протокол DF1/DH-485), DeviceNet, DH+(Data Highway Plus), ControlNet і EtherNet/IP за допомогою відповідних модулів (мережних адаптерів), встановлених у шасі.

Багатозадачна операційна система контролера підтримує 32 задачі, які виконуються за пріоритетами. Одна задача може бути безперервною, інші повинні бути періодичними або обробниками подій. Кожна задача може включати до 32 програм і необмежену кількість функцій і процедур. Виконання програми здійснюється з високою швидкістю. Повний час сканування з урахуванням мережного обміну складає 2 мс[45,46].

### **10.1.2 Шасі ControlLogix**

Система ControlLogix є модульною системою і використовує шасі для встановлення модулів вводу-виводу серії 1756. Шасі випускаються на 4 (1756-A4), 7 (1756-A7), 10 (1756-A10), 13 (1756-A13) і 17 (1756-A17) слотів. Задня шина шасі, за своїми властивостями, являє собою сегмент мережі ControlNet, тому у системах управління, що створені на базі цієї мережі, усі шасі ControlLogix є ланцюгами мережі, а кожен модуль в такій системі є самостійним вузлом мережі. У будь-якому слоті шасі можна розміщувати



будь-який модуль. Декілька контролерів в одному шасі мають можливість через задню шину обмінюватися повідомленнями один з одним. А маючи декілька модулів сполучення в одному шасі, можна послати повідомлення на комунікаційний порт одного модуля, перенаправити його через задню шину на порт іншого модуля і переслати по іншій лінії сполучення кінцевому адресату.

На рис.10.2 показане шасі 1756-А10 з установленим джерелом живлення.

Шасі являє собою, так звану, корзину, яка складається з відсіків на 4, 5 або 6 модулів. У верхній половині шасі розташовані розніми для підключення модулів до задньої шини.

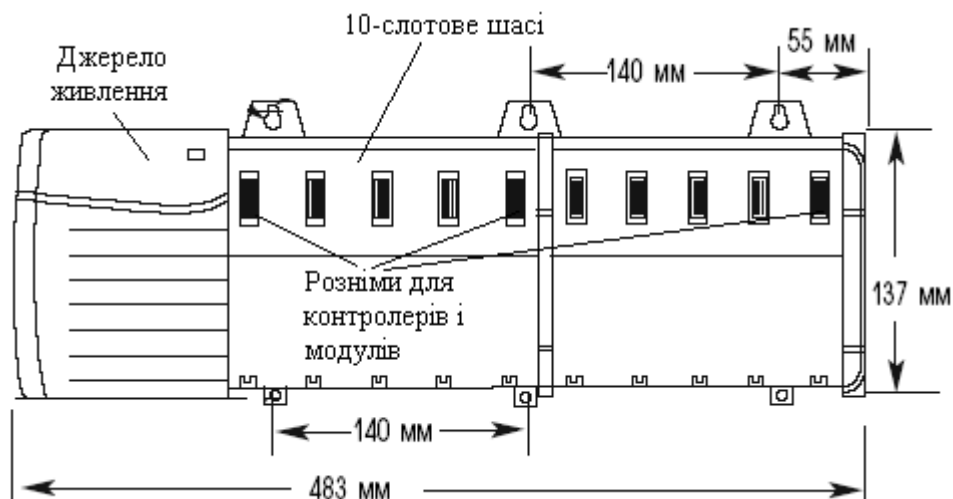


Рисунок 10.2 – Шасі 1756-А10 з установленим джерелом живлення (1756-РА72/-РВ72)

Для приєднання джерела живлення до шасі на його боковій стороні розташований рознім і спеціальні напрямні, по яким джерело живлення всувається доки його рознім не з'єднається з рознімом задньої шини шасі, рис.10.3.

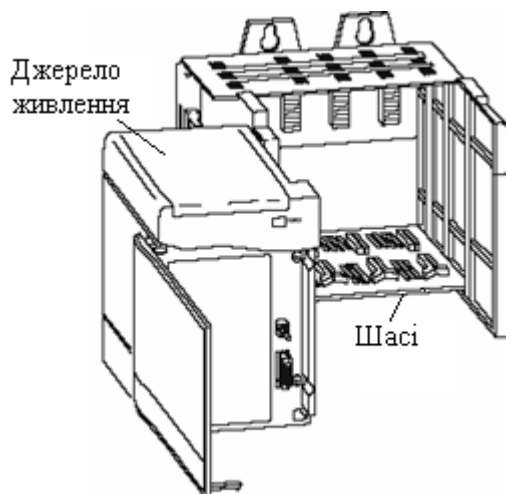


Рисунок 10.3 – Приєднання джерела живлення до шасі

Всі шасі монтуються на панель задньою стінкою. Для закриття порожніх слотів використовується заглушка 1756-N2[47].

### 10.1.3 Джерела живлення

Джерела живлення ControlLogix, що використовуються з шасі 1756, залежно від модифікації, подають напругу 1,2В, 3,3В, 5В або 24В постійного струму безпосередньо на задню шину шасі. Існують нерезервовані (1756-PA72, 1756-PB72, 1756-PA75, 1756-PB75, 1756-PC75, 1756-PH75) і резервовані (1756-PA75R, 1756-PB75R) джерела живлення.

Номінальна вхідна напруга джерел живлення модифікацій 1756-PA72/С і 1756-PA75/В дорівнює ~120/220В змінного струму, модифікацій 1756-BA72/С і 1756-PB75/В – = 24В, модифікації 1756-PC75/В – =48В, а модифікації 1756-PH75 – =125В постійного струму. Вихідна потужність блоків усіх модифікацій складає 75Вт.

Джерела живлення, які в номері серії мають літеру «С» сумісні з шасі серії «А» і «В», а ті, що мають літеру «В» - сумісні з шасі серії «В».

Стандартне джерело живлення, рис.10.4, монтується на лівій стороні шасі, де воно підключається безпосередньо до задньої шини. Для кожного шасі призначено по одному джерелу живлення розміром 112×140×145мм, (Ш×В×Г).

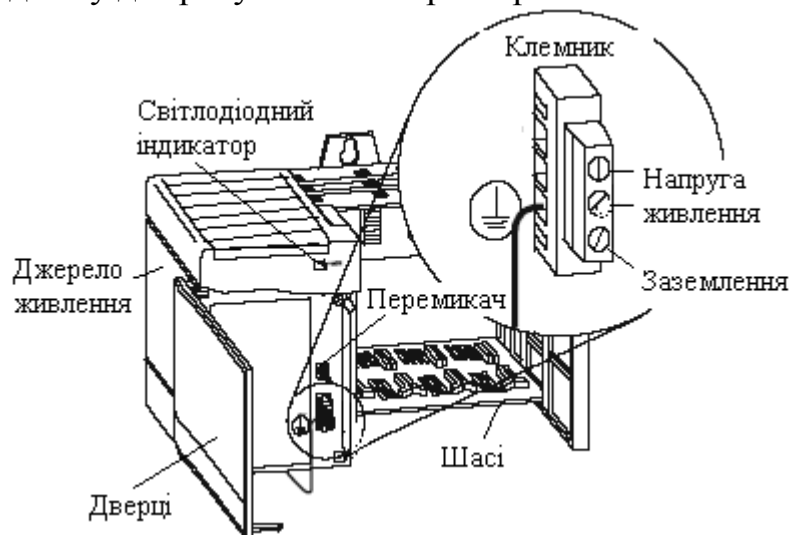


Рисунок 10.4 – Джерело живлення ContrlLogix

На передній стороні джерела живлення під дверцятами розташовані дві клєми для підключення напруги і одна для заземлення. Над клемніком знаходиться перемикач, за допомогою якого джерело живлення можна увімкнути в роботу або вимкнути. У верхній частині джерела живлення знаходиться зелений світлодіодний індикатор, який світиться протягом його нормального функціонування[48].

### 10.1.4 Модулі вводу-виводу

Архітектура ControlLogix забезпечує широкий вибір модулів вводу-виводу. Це дозволяє охопити багато різних задач автоматизації, від

високошвидкісного дискретного управління до управління безперервними процесами. Архітектура ControlLogix використовує технологію постачальник - споживач, яка дозволяє використовувати вхідні дані і стан виходів одночасно декількома контролерами ControlLogix.

Дискретні модулі вводу-виводу ControlLogix є модулями, які контролюють стан вхідних кіл (ON/OFF) і передають реакцію контролера на їх зміни.

Використовуючи мережну модель постачальник - споживач, вони можуть без запиту здійснювати інтелектуальний обмін даними між модулями і іншими пристроями системи. Крім того дискретні модулі вводу-виводу формують повідомлення про несправності на рівні модуля і діагностують помилки підключення електроустаткування. При відновленні працездатності системи допускається видалення і встановлення модулів в шасі не вимикаючи напругу живлення. Монтаж вхідних і вихідних кіл модулів здійснюється за допомогою знімних клемників, рис.10.5.

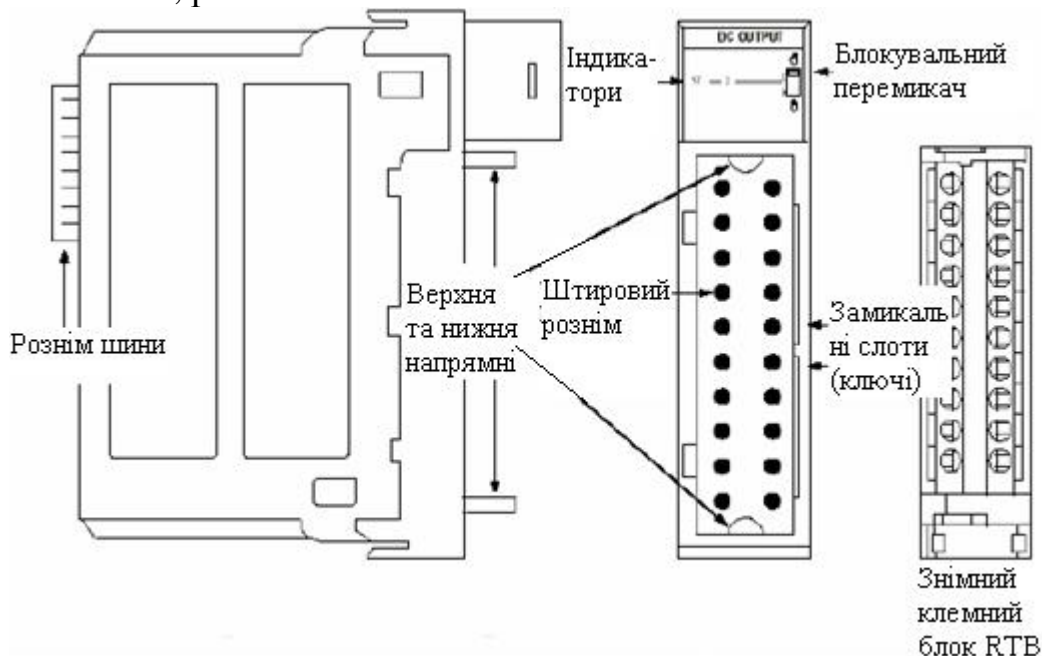


Рисунок 10.5 – Дискретний модуль вводу-виводу ControlLogix

Основними елементами модуля вводу-виводу є його корпус з рознімами на задній і передній панелі, індикатори, блокувальний перемикач і знімний клемний (термінальний) блок RTB.

Рознім на задній панелі - це рознім для підключення модуля до системи ControlLogix за допомогою шини ControlBus.

Через штировий рознім, що на лицевій панелі, здійснюється зв'язок модуля з колами вводу-виводу, живлення і заземлення при використанні термінального блока RTB.

Альтернативою клемним блокам RTB і самостійному підключенню зовнішніх проводів може бути монтажна система, в яку входять модулі інтерфейсу (IFM) з вихідними клемними блоками для дискретних модулів вводу-виводу. Причому модулі інтерфейсу і відповідні модулі вводу-виводу

мають бути заздалегідь з'єднані між собою кабелем. Окрім цих пристроїв для монтажу зовнішніх кіл до модулів вводу-виводу можуть використовуватися готові кабелі, на одному кінці якого приєднаний блок RTB, а інший кінець має кодований кольором стандартний рознім.

Блокувальний перемикач закріплює блок RTB або IFM- кабель на модулі.

Замикальні слоти з правого боку модуля призначені для вставлення в них механічних ключів, які запобігають від неправильного монтажу зовнішніх проводів при помилковому встановленні знімного клемного блока.

Індикатори стану відображають стан зв'язку, справність модуля і пристроїв вводу-виводу.

Верхні і нижні направляючі на передній панелі допомагають приєднувати блок RTB або IFM- кабель до штирового розніму, що на модулі.

Дискретні модулі вводу-виводу серії 1756 мають декілька варіантів виготовлення – стандартне, з ізольованими каналами, з діагностикою і електронним захистом.

На відміну від модулів стандартного виготовлення, модулі з електронним захистом додатково забезпечують захист від короткого замикання, а модулі з діагностикою – визначення обриву лінії, короткого замикання і електронний захист від нього, а також контроль спрацьовування вихідного каскаду.

Модулі мають оптичну розв'язку між каналами і шиною, яка забезпечує захист по напрузі до 2550В постійного струму, протягом 1 с.

Дискретні модулі вводу-виводу за вхідними та вихідними сигналами можна поділити на модулі змінного струму, модулі постійного струму та дискретні релейні модулі виводу.

Серед стандартних модулів змінного струму є 16 – каналні модулі вводу 1756-IN16 і 1756 - IA16 з напругою вхідного сигналу, відповідно ~24 і ~120В, а також 32-каналний модуль 1756-IA32 з вхідною напругою ~120В. Восьмиканальний модуль вводу 1756-IA8D також орієнтований на вхідну напругу ~120В, але він має можливість діагностування вхідних сигналів. 16-каналні модулі 1756-IA16I і 1756-IM16I мають індивідуальну ізоляцію входів, вхідний сигнал першого з них складає ~120В, другого ~240В.

Серед 11 модифікацій модулів вводу постійного струму також є стандартні (1756-IB16, 1756-IB32, 1756-IV16, 1756-IV32) з вхідним сигналом =12/24В, з діагностичними входами (1756-IB16D) на таку саму напругу, а також з індивідуально-ізольованими входами (1756-IB16I, 1756-IN16I та ін.) на напругу =12/24В і =125В.

Дискретні модулі виводу змінного струму є 8- і 16- каналні. Серед 8-каналних, модулі 1756-OA8 і 1756-ON8 мають стандартне виготовлення з вихідною напругою, відповідно ~120/240В і ~240В. Модулю 1756-OA8D з вихідною напругою ~120В надані можливості діагностування вихідних каналів, а модуль 1756-OA8E має внутрішній електронний захист від надмірного струму і вихідну напругу – ~ 120/240В.

Найбільшу кількість модифікацій – 12, мають модулі виводу постійного струму. Серед них є стандартні (1756-OB8, 1756-OC8, 1756-OG16 і 1756-OB32),

з можливістю діагностування (1756-OB16D), з індивідуально ізольованими каналами (1756-OB16I, 1756-OB16IS, 1756-ON8I), з електронним захистом (1756-OB8EI, 1756-OB16E, 1756-OV16E і 1756-OV32E). Більшість з них має вихідну напругу  $=12/24V$ , а модулі 1756-OG16, 1756-OC8 і 1756-ON8I призначені для живлення напругою, відповідно  $=5, 48$  і  $=120V$ .

Модулі 1756-OX8I і 1756-OW16I мають, відповідно 8 і 16 індивідуально ізольованих каналів з релейними виходами.

Усі модифікації модулів вводу-виводу оснащені знімними клемними блоками для спрощення підключення до зовнішніх кіл, а також заміни модулів і мають робочу напругу  $\sim 10-265V$  змінного струму або  $=5-150V$  постійного струму[49].

Модулі аналогового вводу-виводу зовні такі самі, як і дискретні модулі. Тому встановлення їх в шасі та підключення до зовнішніх електричних кіл нічим не відрізняється. Як і дискретні, аналогові модулі ControlLogix здійснюють обмін даними поміж модулями та іншими пристроями системи без спеціального опитування. Дані, з якими працює аналоговий модуль можуть бути представлені у форматі 16-розрядного цілого числа або 32-розрядного цілого числа з плаваючою комою. Сигнали, що надходять до модулів можна масштабувати до інженерних одиниць, а коли їх значення перевищують встановлені межі спрацьовує система попередження.

В системі ControlLogix пропонується 7 аналогових модулів вводу і 4 модуля виводу. Модифікації модулів вводу 1756-IF6I, 1756-IF16 і 1756-IF8 мають, відповідно 6, 8 і 16 входних каналів на струм/напругу, а у модуля 1756-IF6I канали ще і ізольовані.

6-канальні модулі 1756-IR6I та 1756-IT6I призначені для роботи з термометрами опору та термопарами. А модифікація 1756-IT6I2 – є вдосконаленим термопарним модулем з ізольованими шістьма каналами.

4-і 8-канальні неізольовані аналогові модулі 1756-OF4 та 1756-OF8 мають конфігурований вихідний сигнал – струм/напруга, а модифікації 1756-OF6CI та 1756-OF6VI – є 6-канальними ізольованими модулями виводу, відповідно за струмом і напругою.

Серед спеціальних модулів є сімейство аналогових сервомодулів ControlLogix (1756-M02AE, 1756-HYD02, 1756-M02AS і ін.) призначених для управління рухом із замкненим і розімкненим контуром управління для пристроїв, що підтримують аналоговий інтерфейс. Аналогові сервомодулі мають аналоговий вихідний сигнал  $\pm 10V$  і підтримують різні пристрої зворотного зв'язку. Один модуль підтримує до двох осей. В системі на один контролер ControlLogix можуть використовуватися декілька модулів, контролюючих до 32 осей.

Модуль вимірювання потоку 1756-CFM, що настраюється, забезпечує вимірювання і підсумовування витрат на двох каналах.

Модуль високошвидкісного лічильника 1756-HSC на двох каналах рахує кількість імпульсів та їх частоту. Модуль має 4 швидкодіючі вихідні контакти з

установками вмикання-вимикання, працює з імпульсними пристроями і енкодерами.

Модуль програмованого порогового вимикача 1756-PLS призначений для реалізації складних додатків по здійсненню пакування.

Усі модулі вводу-виводу ControlLogix можуть встановлюватися та зніматися без вимикання напруги [50].

### 10.1.5 Робота модулів вводу-виводу

В традиційних системах вводу-виводу контролери опитують модулі вводу аби отримати стан їх входів. В системі ControlLogix модулі контролером не опитуються. Для організації роботи кожний модуль вводу-виводу повинен мати власника - контролер ControlLogix. Цей контролер-власник може бути локальним або віддаленим відносно модуля вводу-виводу. На основі конфігурацій кожного модуля, якими контролер-власник програмно завантажується, він задає модулю поведінку роботи у складі системи управління. Для нормального функціонування модулі вводу-виводу ControlLogix повинні безперервно підтримувати зв'язок з власником. Звичайно кожний модуль в системі має одного власника. Проте вхідні модулі на відміну від вихідних можуть мати більше одного власника, рис.10.6.



Рисунок 10.6 – Робота модулів вводу-виводу

Якщо декілька контролерів конфігуровані як власники одного і того ж модуля, відбуваються наступні події:

- коли контролери починають завантажувати конфігурацію, вони обидва намагаються встановити з'єднання з вхідним модулем;
- встановлює з'єднання той контролер, дані якого прибувають першими;
- коли поступають дані другого контролера, модуль порівнює їх з поточною конфігурацією (даними, одержаними і прийнятими від першого контролера);
- якщо конфігурація, що послана іншим контролером, співпадає з конфігурацією, посланою першим контролером, з'єднання також приймається;

- якщо хоч один параметр другої конфігурації відрізняється від першої, модуль відхилює з'єднання.

Перевага конфігурації з декількома власниками у тому, що при втраті з'єднання з модулем будь-якого контролера, модуль продовжує працювати і передавати дані в систему, оскільки з'єднання буде підтримуватися іншим контролером-власником.

Модулі вводу-виводу у випадку зміни сигналів передають дані колективно або періодично. Частота передачі залежить від опцій, вибраних в процесі конфігурування модулів, і від того, де в системі управління знаходиться модуль.

Така технологія спілкування називається моделлю постачальник-споживач. Модуль вводу є постачальником вхідних даних, а контролер – споживачем даних.

Поведінка модуля змінюється залежно від того, функціонує він у локальному чи віддаленому шасі.

Якщо модуль вводу-виводу дискретний і знаходиться у тому самому шасі, що і контролер-власник (контролер, що має конфігураційні дані модуля вводу-виводу), передача модулем даних визначається двома параметрами конфігурації – запрошеним інтервалом пакета (RPI) і зміною стану (COS).

Запрошений інтервал пакета RPI визначає швидкість, з якою модуль передає дані. Він завантажується у модуль разом з усіма параметрами конфігурації. Коли RPI (визначений цикл) спливає, модуль передає колективні дані.

Зміна стану COS інструктує модуль передавати дані кожного разу, коли певні входи змінюють свій стан з ON в OFF або навпаки.

Метод COS ефективніший за RPI, тому, що дані передаються, коли відбувається зміна. Проте, якщо входи модуля не змінили свій стан протягом інтервалу RPI, модуль передаватиме дані зі швидкістю визначеною RPI.

Якщо модуль вводу знаходиться в іншому шасі ніж контролер-власник, тобто вони сполучаються через мережу ControlNet, то роль RPI і поведінка функції COS визначають тільки момент передачі модулем колективних даних у власному шасі. Отримання цих даних контролером-власником визначає RPI мережі.

Коли контролер-власник працює з модулем виводу у локальному шасі, він посилає йому вихідні дані у кінці кожного циклу сканування програми і/або з частотою, визначеною в RPI модуля, рис.10.7.

Коли модуль виводу знаходиться у віддаленому шасі, рис.10.8, контролер-власник посилає дані до модуля виводу тільки з частотою визначеною RPI модуля. Оновлення даних контролера-власника у кінці циклу сканування програми - не відбувається.



Рисунок 10.7 – Модуль виводу у локальному шасі

Кожного разу, коли модуль отримує дані з контролера, він миттєво здійснює колективну передачу команд виводу. Дійсні вихідні дані повторюються модулем виводу як вхідні дані і колективно передаються назад у мережу. Це називається відлунням вихідних даних, до якого також може додаватися діагностична інформація і інформація про помилки. Таким чином, у моделі постачальник-споживач, модуль виводу є споживачем вихідних даних контролера і постачальником відлуння даних.

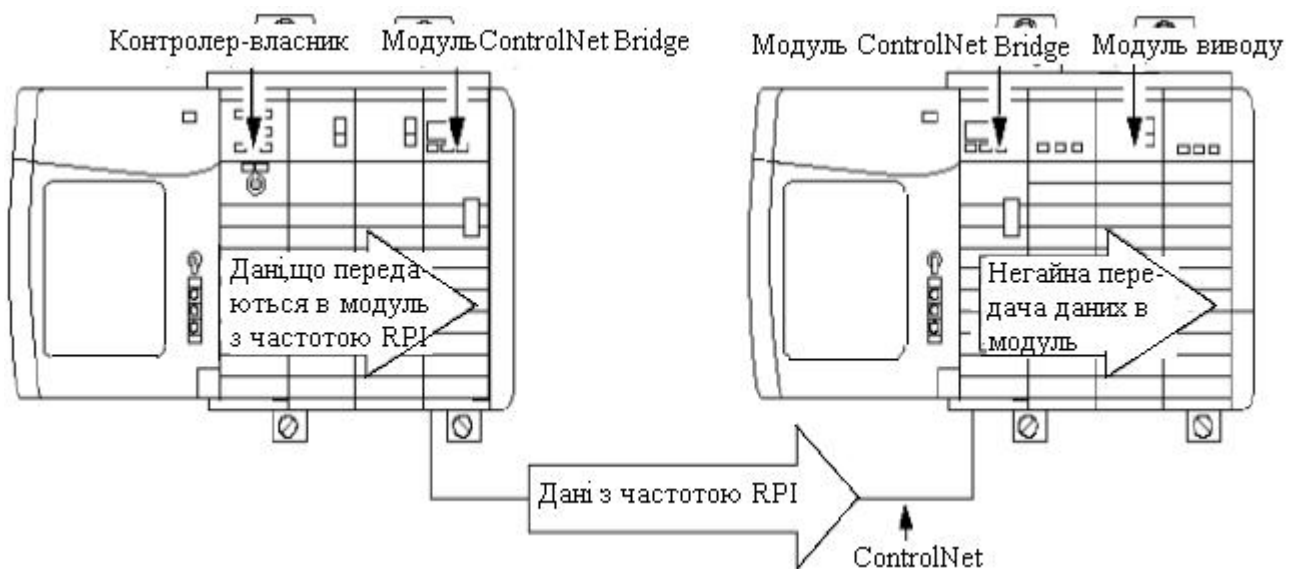


Рисунок 10.8 – Модуль виводу у віддаленому шасі

Залежно від значення RPI довжини циклу сканування програми, модуль виводу може приймати дані і передавати «відлуння» даних багато разів протягом одного сканування програми.

Коли значення RPI визначене для модуля виводу у віддаленому шасі, контролер-власник інструктується відносно колективної передачі вихідних даних усередині його власного шасі і RPI «резервує» комірку у потоці даних, що передаються у мережі ControlNet.



Якщо модуль вводу-виводу аналоговий і знаходиться в тому ж шасі, що і контролер-власник, обмін даними базується на вибірці у реальному часі (RTS) і запитаному інтервалі пакетів (RPI).

Вибірка у реальному часі RTS настроюється і вказує модулю частоту сканування вхідних каналів і збору всіх доступних даних. Після того, як канали проскануються, модуль здійснює групову передачу зібраних даних через задню панель локального шасі.

Запитаний інтервал пакетів RPI також настроюється і за аналогією роботи дискретних модулів визначає, з яким циклом аналоговий модуль поширюватиме дані каналів через задню панель локального шасі. RPI вказує модулю поширювати поточний зміст його вбудованої пам'яті у момент закінчення RPI.

Модуль скидає таймер RPI кожний раз, коли виконується RTS. Це означає, що від цих параметрів залежить, як і коли контролер-власник в локальному шасі одержить оновлені дані каналу. Якщо значення RTS - менше або дорівнює RPI, кожна передача даних з модуля несе оновлену інформацію каналів. Фактично, модуль передає дані з частотою RTS.

Якщо значення RTS більше ніж RPI, модуль передає дані з частотою і RTS і RPI. Їх значення вказують, як часто контролер-власник одержує дані, і як багато передач модуля містять оновлені дані каналів.

Якщо вхідний аналоговий модуль розташований у віддаленому шасі, вплив RPI і RTS модуля на отримання даних контролером-власником зміниться ненабагато, і залежатиме від типу мережі, яка використовується для підключення модуля.

Якщо віддалений аналоговий модуль вводу підключений до контролера-власника через мережу ControlNet, інтервали RPI і RTS все ще визначають, коли модуль передаватиме дані усередині його власного шасі. Проте тільки значення RPI мережі визначає, як часто контролер-власник одержуватиме їх через мережу.

В модуль аналогового виводу контролер-власник посилає дані тільки з періодом, визначеним у RPI, а не в кінці сканування програми контролера. Коли модуль аналогового виводу одержує нові дані, значення яких відповідають аналоговому сигналу на вихідних контактах контролера-власника (тобто кожен RPI), він автоматично передає або «відображає» дані в решту частини системи управління. Відлуння вихідних даних існує незалежно від того, в локальному або у віддаленому шасі розташований модуль. Залежно від значення RPI і тривалості сканування програми контролера, вихідний модуль може одержувати дані і передавати "відлуння" багато разів протягом одного скана програми. Фактично, якщо RPI менше, ніж тривалість сканування програми, контролер дозволяє вихідним канали модуля неодноразово змінювати значення протягом одного скана програми, оскільки вихідний модуль не вимагає досягнення кінця програми для передачі даних.

Якщо вихідний модуль розташований у віддаленому шасі, вплив RPI на отримання даних від контролера-власника ненабагато зміниться, і залежатиме від типу мережі, до якої підключений модуль.

Якщо віддалений аналоговий модуль підключений до контролера-власника через мережу ControlNet, то на додаток до настройки модуля для передачі даних усередині його власного шасі, RPI також "резервує" місце в потоці даних, що передаються через мережу ControlNet.

Існує два методи спілкування контролера з модулями вводу-виводу – пряме і рекове. При прямому спілкуванні передача даних між контролером-власником і модулем вводу-виводу, що займає окремий слот, відбувається відповідно до RPI окремих модулів вводу-виводу. Якщо дискретні модулі вводу-виводу розташовані у віддаленому шасі, то користувач під час початкового конфігурування модулів може у форматі сполучення вибрати рекову оптимізацію. Відносно прямого сполучення рекове реалізується швидше, оскільки контролер-власник працює з RPI не окремих модулів, а з RPI окремого река, яке збирає всі модулі вводу-виводу віддаленого шасі в одному рековому сполученні, рис.10.9.

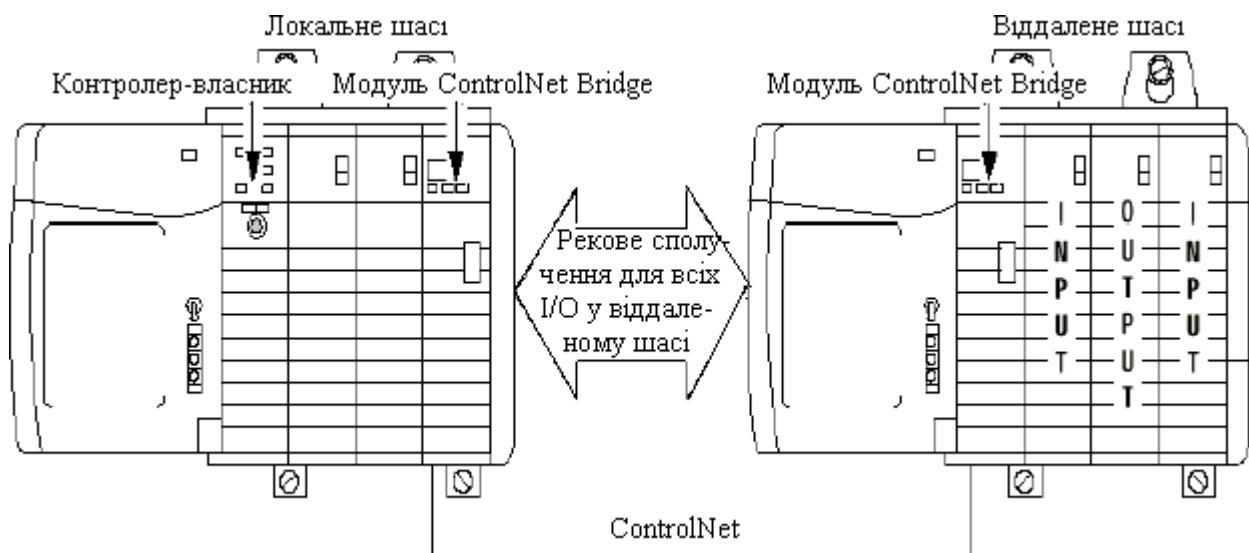


Рисунок 10.9 – Рекове сполучення з вводом-виводом у віддаленому шасі

Рекове сполучення використовується коли контролер працює на нижньому рівні системи автоматизації і використовуються дискретні модулі вводу-виводу без діагностики та запобіжників. Таке обмеження пояснюється тим, що додаткові функції модулів при рековому сполученні не передаються[45].

### 10.1.6 Комунікаційні можливості

Контролери ControlLogix можуть підтримувати зв'язок з пристроями автоматизації, використовуючи мережі EtherNet/IP, ControlNet, DeviceNet, DH+ (Data Highway Plus), RS-232 (протокол DF1/DH-485). Для цього система ControlLogix має у своєму розпорядженні набір інтерфейсних модулів.

Модуль EtherNet/IP-сполучення 1756-ENBT призначений для управління вводом-виводом. Він працює як адаптер для розподіленого вводу-виводу у віддаленій мережі EtherNet/IP, зв'язується з іншими пристроями EtherNet/IP, маршрутизує повідомлення пристроям в інших мережах.

Вдосконалений модуль EtherNet/IP-сполучення 1756-ENEB підтримує доступ до даних контролера ControlLogix (читання і запис), міжмережну передачу повідомлень, замовлення веб-сторінки, електронну пошту.

Модулі EtherNet/IP- сполучення мають 1 порт, через який забезпечується передача інформації зі швидкістю 10 або 100 Мбіт/с.

Модулі сполучення 1756-CNBR і 1756-CNB підтримують управляючу мережу ControlNet, яка відповідає вимогам високопродуктивних додатків, що працюють в режимі реального часу. На базі мережі ControlNet можна створювати розподілені системи управління протяжністю до 6 км (до 30 км на оптоволоконному кабелі), причому швидкість роботи мережі не залежить від довжини лінії. Мережа ControlNet використовує загальний промисловий протокол CIP (Common Industrial Protocol), для об'єднання функціональних можливостей мережі вводу-виводу і однорангової інформаційної мережі, забезпечуючи високошвидкісну роботу обох мереж. Модуль має 1 порт, через який ведеться обмін інформацією зі швидкістю 5 Мбод.

Модуль 1756-DNB підтримує відкриту мережу нижнього рівня DeviceNet, яка забезпечує сполучення між датчиками і виконавчими механізмами та пристроями вищого рівня (такими, як програмовані контролери і комп'ютери) зі швидкістю 125, 250, 500Кбод (за вибором). Мережа DeviceNet також використовує CIP-протокол. Це забезпечує для промислових пристроїв можливість управління, настроювання і збору даних.

Модуль мереж DH+ і Remote I/O 1756-DHRIO підтримує передачу повідомлень між пристроями в мережах DH+. Можливість Remote I/O дозволяє модулю виконувати роль сканера для передачі дискретних сигналів і блоків даних від/для пристроїв віддаленого вводу-виводу. Модуль має 2 порти і здійснює обмін інформацією зі швидкістю передачі даних – 57,6 Кбод для DH+ і 57,6, 115, 230 Кбод для Remote I/O.

За допомогою модулів, що підтримують мережі DeviceNet, ControlNet, EtherNet/IP можна створити, так звану, відкриту мережну архітектуру NetLinx (NetLinx Open Network Architecture). Оскільки всі мережі в архітектурі NetLinx використовують загальний промисловий протокол CIP, це дозволяє виконати наскрізне об'єднання всіх компонентів системи автоматизації, від декількох пристроїв однієї мережі, до багатьох пристроїв в декількох мережах, включаючи доступ в Internet. Для зменшення витрат на установлення компонентів системи, збільшення продуктивності її роботи і полегшення зміни мережі або конфігурації системи, необхідно віддалені вводи-виводи кожного контролера підключати в окрему мережу.

При цьому треба враховувати, що:

- інтерфейсний модуль 1756 - ENET/ENBT підтримує в мережі Ethernet модулі вводу-виводу серій 1756, 1794 і 1797;

- інтерфейсний модуль 1756 - CNB/CNBR підтримує в мережі ControlNet модулі вводу-виводу серій 1756, 1771, 1794 і 1797;
- інтерфейсний модуль 1756 - DNB підтримує в мережі DeviceNet модулі вводу-виводу серій 1791 і 1794;
- інтерфейсний модуль 1756 - DHRIO підтримує в мережі Remote I/O модулі вводу-виводу серій 1746, 1771, 1791 і 1794.

Інтерфейсні модулі можна використовувати в мережах без контролера ControlLogix. У цьому випадку з декількох модулів сполучення, що установлені на шасі, конфігурується шлюз для міжмережної передачі або маршрутизації управляючих і інформаційних даних. Повідомлення від одного інтерфейсного модуля до іншого пересилаються через задню шину. При цьому маршрутизувати повідомлення можна максимум через 4 шасі (8 комунікаційних ділянок).

Комунікаційні можливості контролерів ControlLogix поширюють також інтерфейсні модулі 1788-CN2FF, 1757-FFLD2, 1757-FFLD4, які підтримують інформаційну мережу FOUNDATION Fieldbus. Пристрої, що об'єднані мережею FOUNDATION Fieldbus, можуть використовуватися для складного, сильно розподіленого управління безперервними процесами.

Модулі MVI56-HART, 1756sc-IF8H, 1756sc-OF8H, 1794 FLEX I/O і 1797 FLEX Ex I/O підтримують HART - протокол, розроблений для підключення аналогових датчиків.

Модуль 1756-MVI, -MVID має послідовний порт RS-232, через який він взаємодіє з іншими пристроями послідовного сполучення, використовуючи протокол DF1.

Модулі інтерфейсу SERCOS (1756-M03SE, 1756-M08SE, 1756-M016SE) призначені для сполучення системи ControlLogix з інтелектуальними сервоприводами. SERCOS – це послідовний протокол системи сполучення реального часу. Модулі інтерфейсу SERCOS використовують одиночну цифрову оптоволоконну лінію сполучення, яка замінює до 18 окремих проводів на кожну керовану вісь.

На рис.10.10 наведений приклад інтерфейсних сполучень. Контролер 1756 ControlLogix управляє локальними (розташованими в тому самому шасі) дискретними модулями вводу-виводу, управляє віддаленими пристроями вводу-виводу по мережі DeviceNet, посилає і приймає повідомлення для/від контролера CompactLogix в мережі EtherNet/IP, створює один тег, що споживає контролер 1794 FlexLogix, програмується програмним забезпеченням RSLogix 5000.

Контролер 1756 ControlLogix використовує сполучення «оптимізований рек» з локальними модулями вводу-виводу, модулем 1756-ENBT та прямі сполучення з модулем 1756-DNB, з програмним забезпеченням RSLogix5000, а також повідомлення в контролер CompactLogix і тег, що створюється для контролера 1794 FlexLogix [46].

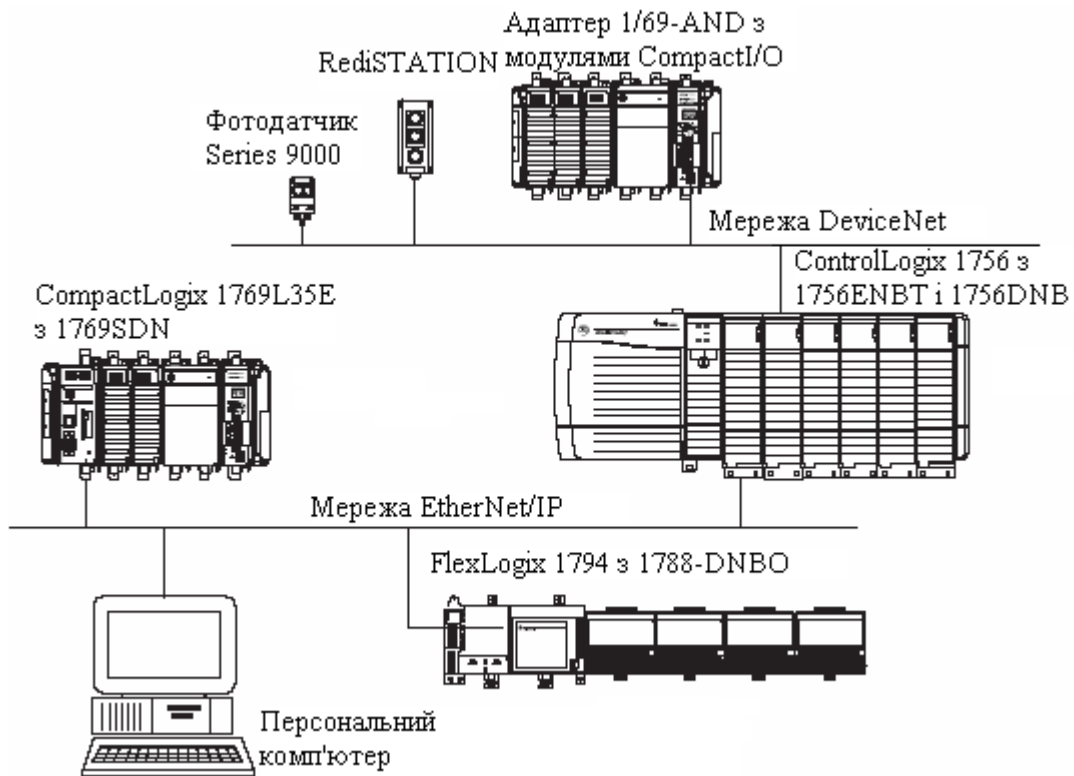


Рисунок 10.10 – Інтерфейсні сполучення у системі Logix

### 10.1.7 Резервування у ControlLogix

Резервування є однією із суттєвих можливостей ControlLogix, без якої неможливе створення надійної системи управління. Резервування може бути реалізоване апаратним і програмним шляхом.

При апаратному резервуванні передбачається резервування:

- шасі контролера;
- блоків живлення шасі контролерів;
- ліній зв'язку мережі ControlNet.

Резервування кабельної системи ControlNet потребує модулі ControlNet 1756-CNBR і дві ідентичні лінії ControlNet, рис10.11:

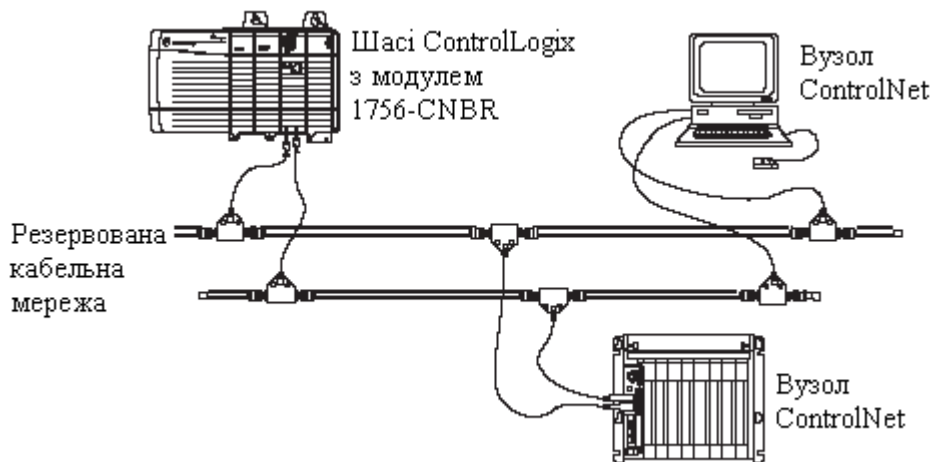


Рисунок 10.11 – Резервування кабельної системи ControlNet

Для резервування джерел живлення потрібні два джерела живлення модифікації 1756-PA75R або 1756-PB75R, модуль адаптера шасі 1756-PSCA2, два кабелю 1756-CPR2 для приєднання джерел живлення до адаптерів 1756-PSCA2, а також сигнальні проводи для підключення джерел живлення до входів модуля, рис.10.12:

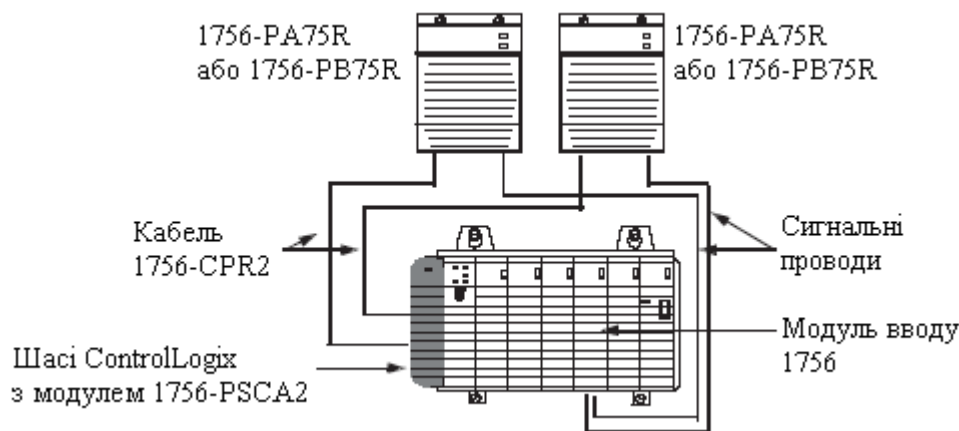


Рисунок 10.12 – Резервування джерел живлення

При резервуванні шасі контролера потрібно таке саме шасі з однаковим розташуванням модулів, по одному з контролерів 1756-L55, 1756-L61, 1756-L62 або 1756-L63 на шасі, максимум 5 модулів сполучення, якими можуть бути 1-5 модулів 1756-CNB або 1-2 модуля 1756-ENBT, і один модуль 1757-SRM на кожне резервоване шасі, рис.10.13:

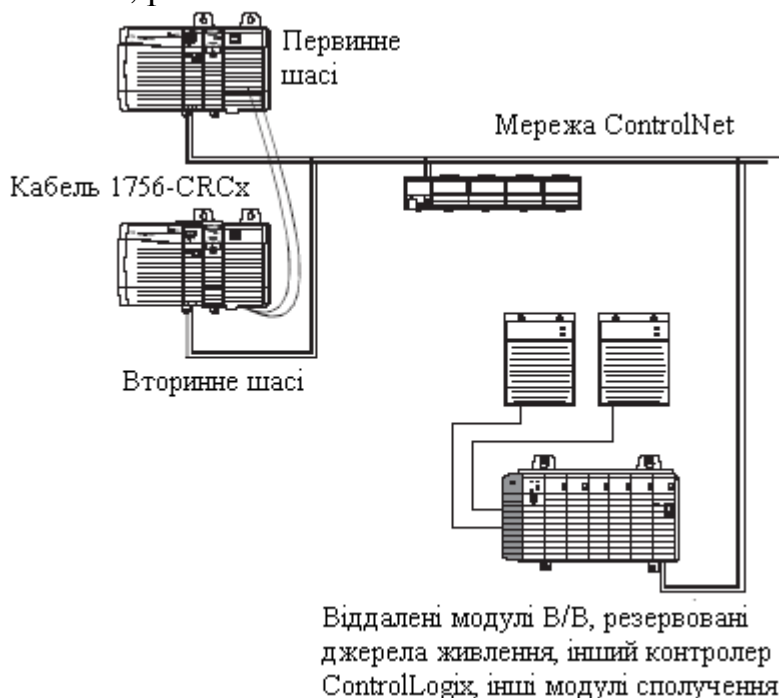


Рисунок 10.13 – Резервування шасі контролера

При апаратному резервуванні завантаження програми здійснюється тільки в один процесор, в другий - вона перезаписується. Підтримка ідентичності при внесенні змін у програму першого процесора, забезпечується модулями

резервування. Завдяки принципу передачі (постачальник-споживач), що застосовується в мережі ControlNet, обидва процесори одночасно одержують дані від модулів входів. Завдяки цьому забезпечується синхронізація і ідентичність виконання програм обома процесорами. Модулям резервування залишається тільки стежити за працездатністю процесорів і у разі потреби перемкнути управління виходами з одного на інший.

При програмному резервуванні функції модулів резервування виконуються спеціальною програмою, що вимагає від розробника додаткової роботи. Проте це суттєво знижує вартість системи управління, навидь у відносно невеликих задачах[51].

## **10.2 Програмування контролерів Rockwell Automation**

Програмування контролера ControlLogix здійснюється мовами релейно-контактних схем, функціональних блоків, структурованого тексту та послідовних функцій за допомогою пакету RSLogix 5000, побудованого на основі стандарту IEC 61131-3. Пакет RSLogix 5000 призначений також для конфігурування модулів вводу-виводу та інтерфейсних модулів.

Мови програмування підтримують символну адресацію, роботу з масивами і структурами, надають повний набір математичних і логічних функцій. Стандартний набір команд включає понад 200 інструкції програмування, а в мову функціональних блоків включені додаткові інструкції по управлінню процесами і позиціонуванням. Кількість внутрішніх змінних, контурів регулювання і інших інструкцій в робочій програмі обмежена тільки об'ємом пам'яті процесора. У пакет вбудована функція графічного відображення стану до 16 змінних у вигляді трендів.

Всі функції контролера (обробка всіх модулів, ПД - інструкції) реалізуються стандартними засобами мови, ніякі спеціальні підпрограми не потрібні. Відсутні обмеження на число внутрішніх змінних, ПД і інших інструкцій. Завантаження і коригування програм може здійснюватися без зупинки процесора.

RSLogix 5000 разом з пакетом програмної емуляції RSLogix Emulate 5000 дозволяють на комп'ютері запускати і налагоджувати код проекту[52].

### **10.2.1 Методика створення проекту у системі програмування RSLogix 5000**

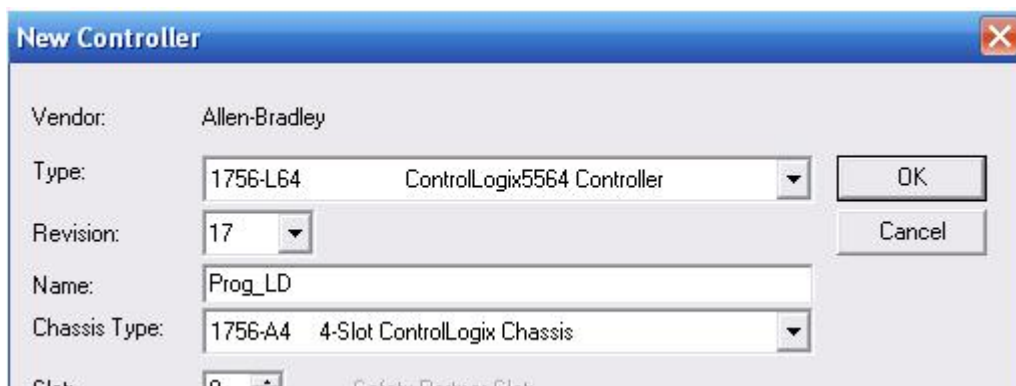
Розробити мовою релейної логіки проект керування роботою апарата, в якому відбувається нагрів і змішування компонентів. Якщо натиснута кнопка "Пуск" і не натиснута кнопка "Стоп", то відкривається клапан і суміш наповнює апарат. Після досягнення верхнього рівня клапан суміші закривається. Одночасно відкривається 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.

### *Розробка 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 і у вікні New Controller, що з'явиться, вибрати тип (Type) контролера -1756-L64:



ControlLogix 5564 Controller, версію мікропрограмного забезпечення (Revision) -17, призначити назву проекту, наприклад Prog\_LD, тип шасі з необхідною кількістю слотів Chassis Type – 1756-A4 4-Slot ControlLogix Chassis, слот центрального процесора (Slot) -0 і директорію знаходження файлу проекту (можна залишити директорію, що пропонується за замовчуванням), натиснути OK. Відкриється вікно проекту з організатором програми:



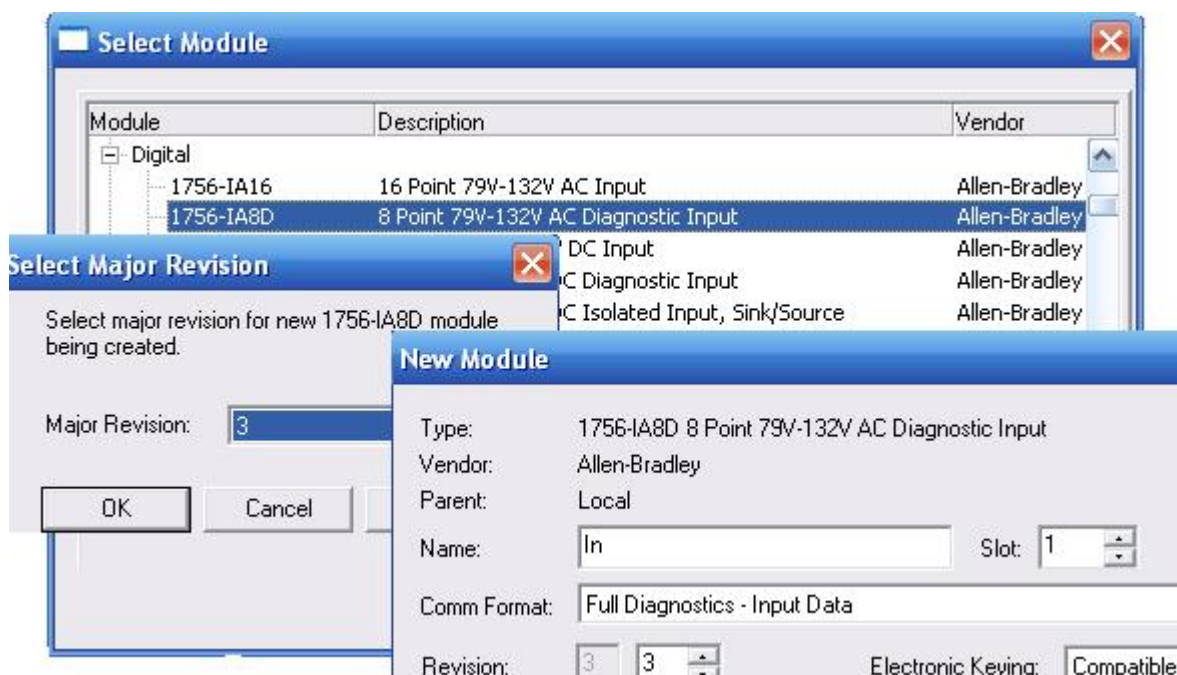


Для конфігурування контролера з меню Edit вибрати Controller Properties. З'явиться діалогове вікно із закладкою General, в якому можна, при необхідності, змінити інформацію, яка була введена у вікні New Controller:



Відкриваючи по черзі закладки Serial Port, Sistem Protocol, Advanced, здійснити необхідні зміни контролеру і натиснути ОК.

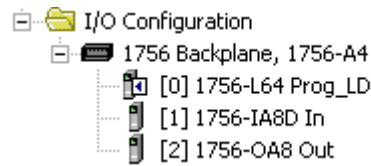
Для конфігурування контролера необхідними модулями вводу-виводу в організаторі проекту правою клавішею миші натиснути на теці I/O Configuration і відкрити меню New Module. У діалоговому вікні Select Module з каталогу Module вибрати групу модулів Digital і відкрити її. Зі списку модулів, що з'явиться, вибрати потрібний модуль вводу, наприклад 1756-IA8D, і натиснути кнопку ОК, а у вікні Select Major Revision залишити 3. Після вибору модуля з'явиться вікно, у якому присвоїти йому ім'я, наприклад In, і вказати слот розташування - 1:



Натиснути ОК і у вікні Module Properties, що з'явиться, виставити зручний інтервал часу, наприклад 50ms.

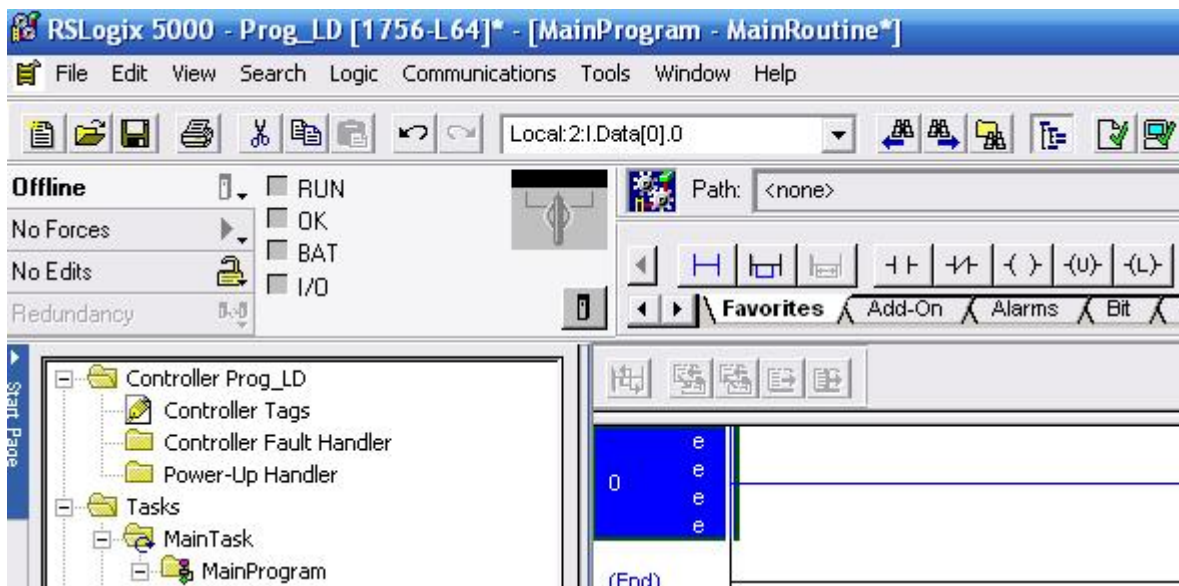
Аналогічно з групи Digital вибрати модуль виводу, наприклад 1756-OA8, і встановити його у слот 2.

Після конфігурування контролера тека I/O Configuration отримає наступний вигляд:



### Створення програмного коду

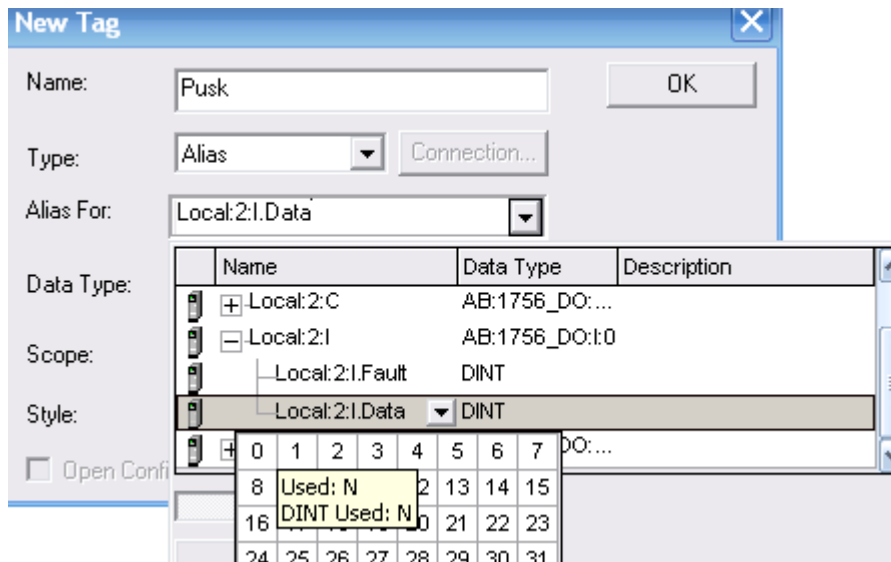
Для створення програмного коду в проектному дереві відкрити теки Task> MainTask> MainProgram, і двічі клацнути лівою клавішею миші по вкладці MainRoutine. Відкриється робоче поле для створення коду програми з вихідною LD-мережею:



В панелі інструментів лівою клавішею миші вибрати альтернативу Favorites, щоб активізувати кнопки елементів графічної LD-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-котушку 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 кожным елементом мережі, і правою клавішею меню відкрити меню. Вибрати альтернативу New Tag...і у вікні, що з'явиться, призначити ім'я новому 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 Push:



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

Local:2:I.Data.0 – (Location:Slot:Type.Member. 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;

Bit – Конкретний канал в модулі вводу-виводу.

Тип змінної BOOL визначається автоматично, а контекст (Scope) і стиль (Style) вибрати, відповідно, як Prog\_LD(Назва програми) і Binary. Натиснути кнопку ОК, ім'я та адреса змінної з'являться над відповідним елементом програми.

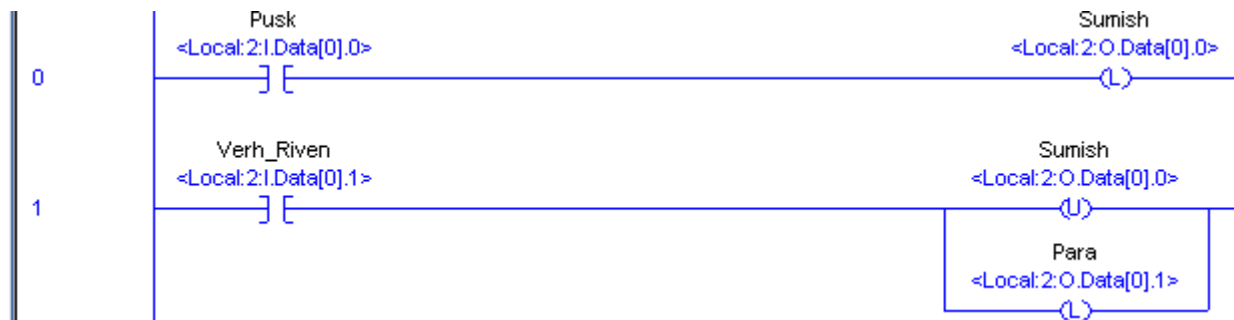
Для створення другої LD-мережі, яка забезпечить керування клапанами, що подають суміш і пару при спрацьовуванні датчика верхнього рівня, клацнути лівою клавішею миші спочатку по кінцевій мережі (End) в робочому полі, а потім по кнопці  Rung в панелі інструментів проекту.

За аналогією з першою мережею встановити прямий контакт і паралельні гілки з U-котушкою і L-котушкою та оголосити змінні з назвами, відповідно, Verh\_Riven, Sumish і Para.

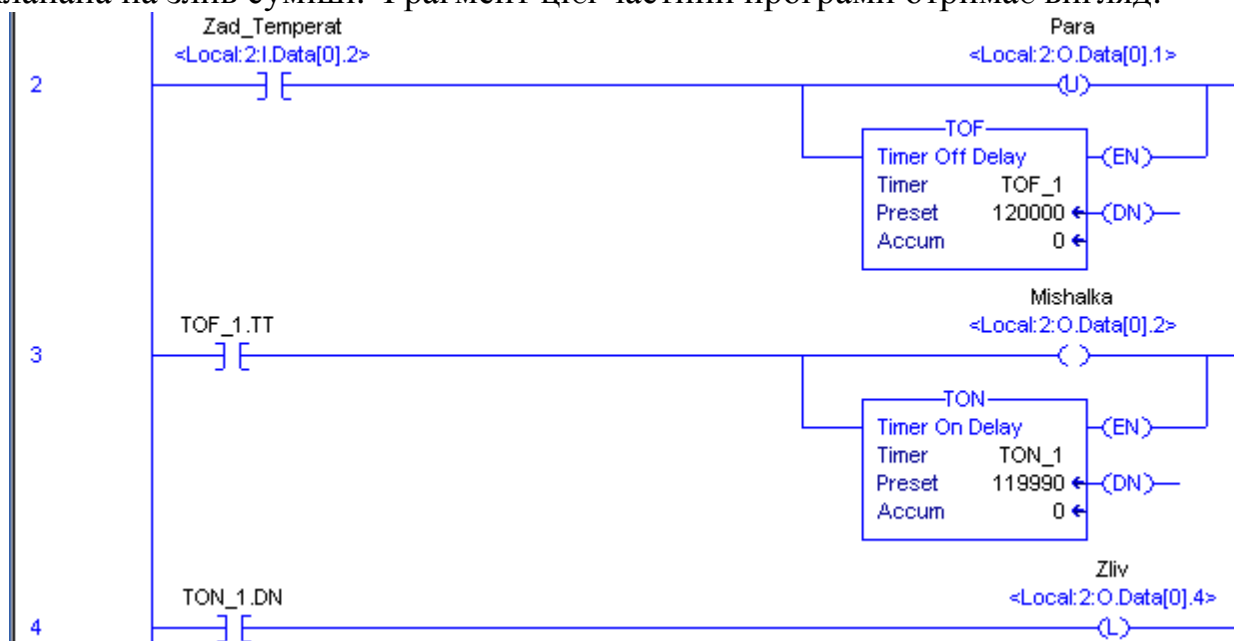
Щоб побудувати паралельну гілку необхідно позначити лівою клавішею миші відповідний елемент програми і в панелі інструментів натиснути лівою клавішею миші кнопку Branch . Якщо паралельна гілка не охоплює U-котушку, треба встановити курсор на неї, натиснути ліву клавішу миші і передягнути кінець гілки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

Для присвоєння котушки вже оголошеної назви Sumish, треба двічі клацнути лівою клавішею миші по знаку питання, що над нею. Відкриється доступ до списку змінних, з якого лівою клавішею 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 клавішею миші попередній елемент мережі (прямий контакт), в панелі інструментів активізувати каталог Timer/Counter і натиснути лівою клавішею миші кнопку TOF. Коли таймер з'явиться у мережі, клацнути правою клавішею миші по рядку Timer, відкриється вікно New Tag..., в якому оголосити цей функціональний блок, надрукувавши ім'я TOF\_1, вибрати тип – Base, тип даних TIMER і назву проекту Prog\_LD. 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 зливом суміші, треба створити додаткову четверту LD-мережу з вихідним прямим контактом таймера TOF\_1.TT і прямою котушкою.

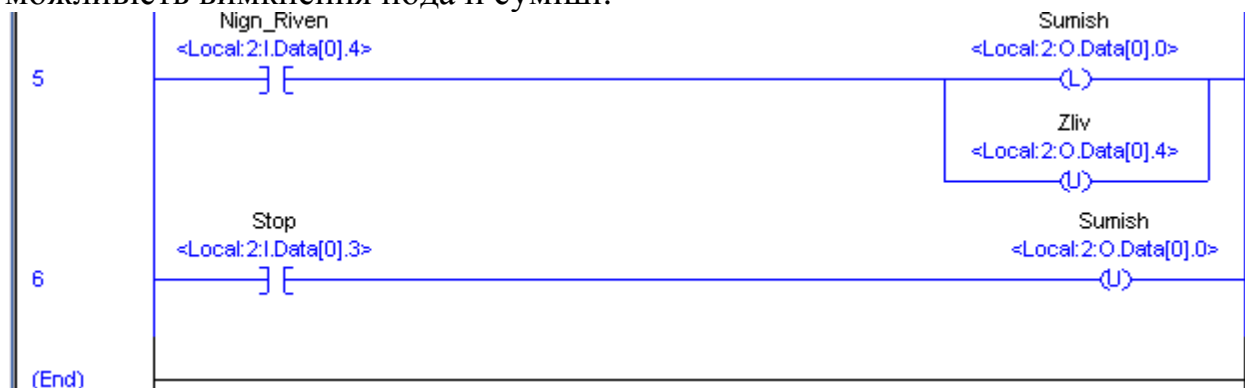
Для оголошення вихідної змінної таймера, двічі клацнути лівою клавішею миші по знаку питання, що над прямим контактом у новій LD-мережі.


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, встановити таймер TON паралельно прямій котушки 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:

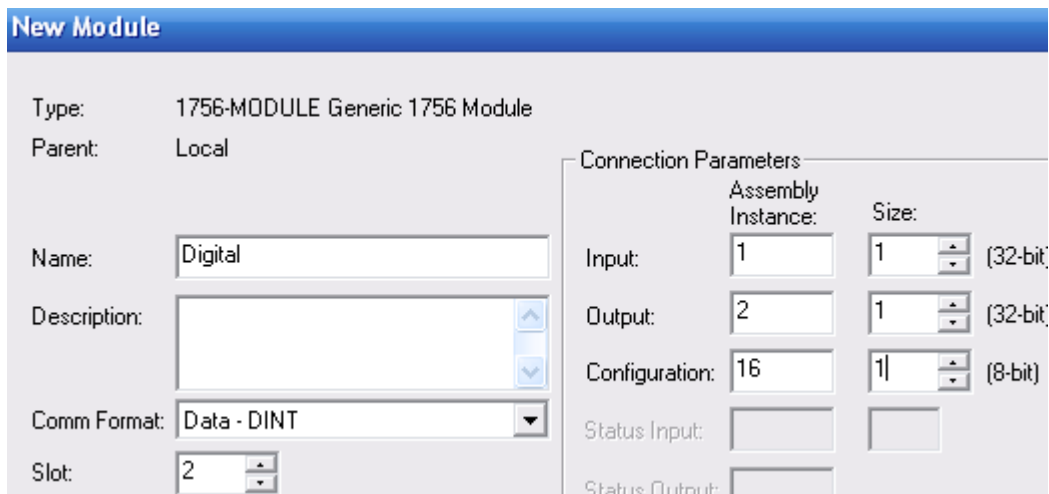


Для перевірки синтаксису програми у верхньому рядку інструментів натиснути кнопку  (Verify Routine). Якщо інформаційне вікно (відкривається і закривається кнопками на клавіатурі Alt+1), покаже помилки – виправити їх.

### ***Перевірка роботи програми***

Для перевірки роботи створеної програми необхідно переконфігурувати контролер на відповідні емулятори. Для цього в організаторі проекту у теці I/O Configuration вилучити модулі вводу-виводу і замінити контролер 1756-L64 ControlLogix 5564 Controller емулятором Emulator RSLogic Emulate 5000 Controller, встановити модуль вводу-виводу 1756 MODULE, а також додатково сконфігурувати емулятор шасі (Chassis Monitor). Для видалення модулів вводу-виводу з теки I/O Configuration необхідно позначити їх лівою клавішею миші та натиснути на клавіатурі клавішу Delet. Для зміни типу контролера в організаторі проекту треба двічі клацнути по ньому лівою клавішею миші і, користуючись клавішею Change Controller у вікні Controller Properties, що з'явиться, зробити необхідні коригування. Щоб встановити модуль 1756 MODULE, треба в організаторі проекту позначити лівою клавішею миші теку I/O Configuration і натиснути праву клавішу. Вибрати меню New module... і з каталогу модулів у вікні Select Module, що з'явиться, відкрити групу Other, позначити модуль 1756-Module лівою клавішею миші і натиснути ОК.

З'явиться вікно New Module, в якому ввести ім'я модуля, наприклад Digital, номер слота -2, де він знаходиться, а також параметри зв'язку, які наведені у фрагменті вікна:



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

Щоб прив'язати код розробленої програми до емулятора, треба формат, в якому оголошені адреси змінних, скоригувати відповідно до наведеного:

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;

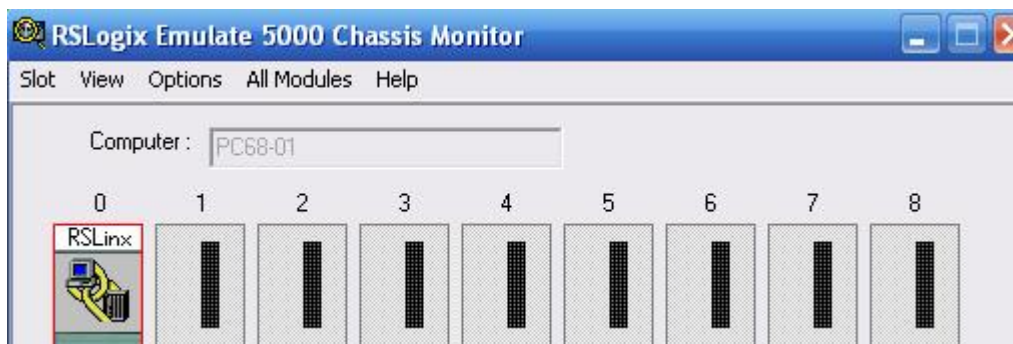
SubMember – Конкретні дані, що відносяться до Member;

Bit – Конкретний канал в модулі вводу-виводу.

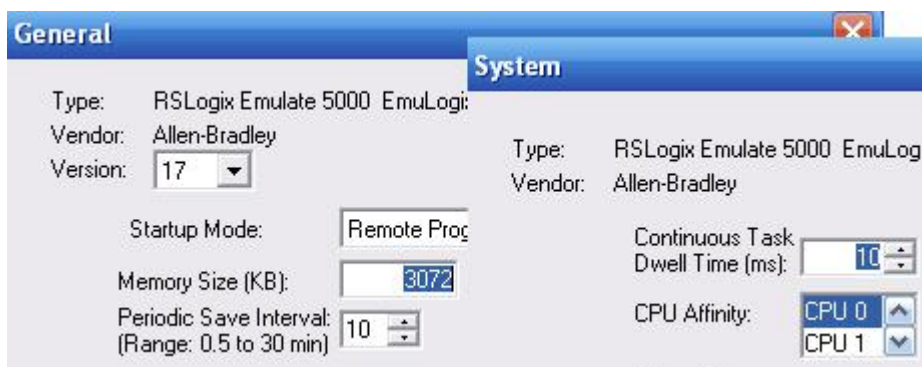
Для цього на програмному коді позначити лівою клавшею миші змінну і натиснути праву клавшею. З меню, що з'явиться, вибрати Edit“змінна”Properties і у вікні Tag Properties скоригувати відповідним чином рядок Alias For:.

Щоб сконфігурувати емулятор шасі, потрібно запустити програму RSLogix Emulate Chassis Monitor, використовуючи шлях Пуск->Все програми->Rockwell Software-> RSLogix Emulate5000-> RSLogix Emulate Chassis Monitor.

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

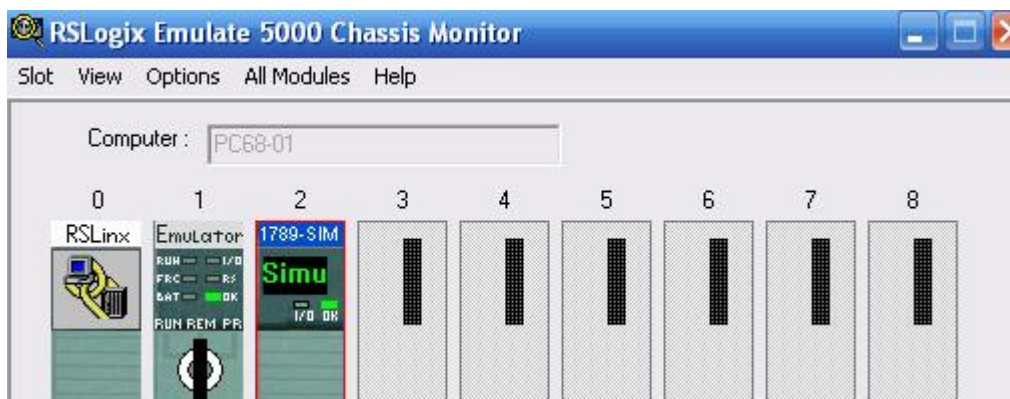


Щоб додати у слот №1 емулятор модуля центрального процесора, з меню Slot вибрати Create Module (Создать модуль), а у вікні Selekt Module, що з'явиться, вибрати тип модуля Emulator RSLogix Emulate 5000 Controller. Якщо після цього з'явиться повідомлення про попередні конфігурації, помітити альтернативу 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\_emulete.

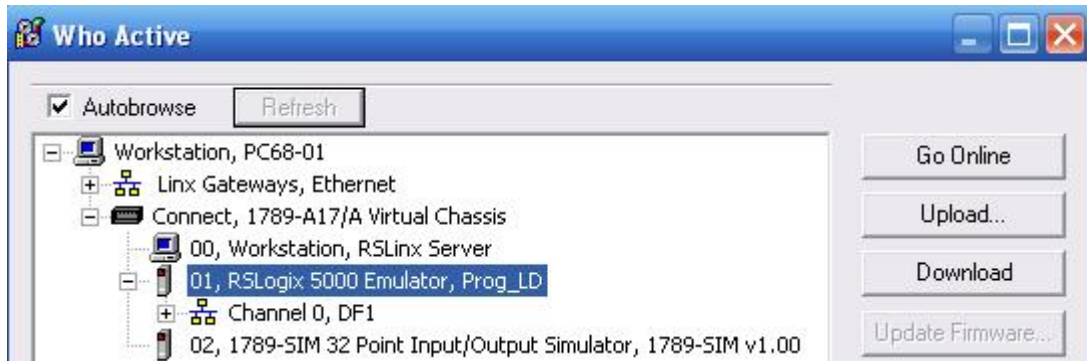
Після завершення конфігурування, емулятор Монітор Шасі отримує вигляд:



Для створення драйвера зв'язку емулятора Монітор Шасі з іншими прикладними програмами запуснути програму RSLinx, користуючись меню Пуск> Все программы> Rockwell Software > RSLinx > RSLinx Classica, або натиснувши праву клавішу миші на модулі RSLinx в емуляторі RSLogix Emulate Chassis Monitir, викликати кнопку Open RSLinx і лівою клавішею миші відкрити вікно RSLinx Classic Gateway. З меню Communications вибрати Configure Drivers. З'явиться вікно Configure Drivers, якщо у вікні вже показане ім'я драйвера, 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 RSLinx Classic Driver, що з'явиться, призначити ім'я драйверу, наприклад Connect, і натиснути ОК. Після цього назва драйвера з'явиться у списку Configured Drivers. Закрити вікно кнопкою Close.

Для завантаження проекту, в головному меню Communications вибрати Who Active, позначити канал зв'язку з емулятором 01,RSLogix5000 Emulator, Prog\_LD 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.

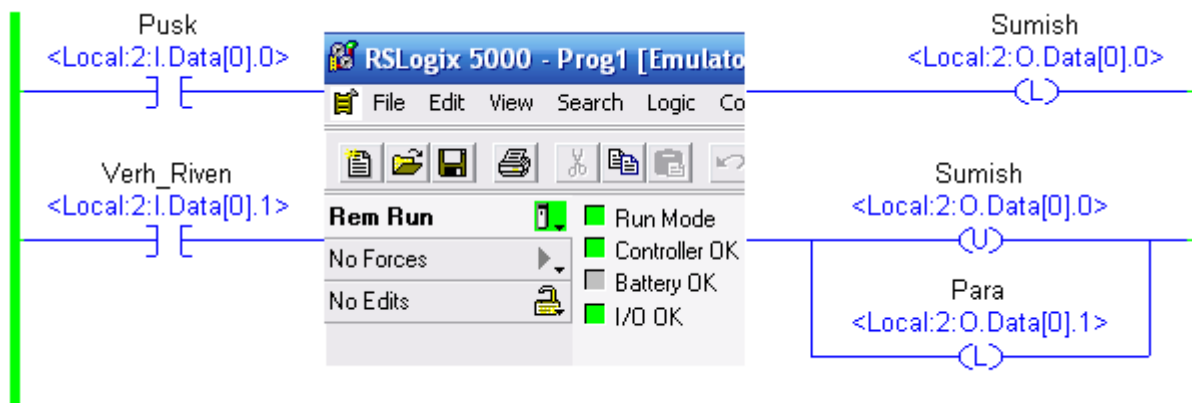
Це свідчить про те, що інструментальна система для перевірки роботи програми готова. Якщо режим Run не увімкнувся, треба увійти в меню Rem Run, що у лівому верхньому куті, і вибрати альтернативу Run Mode.

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

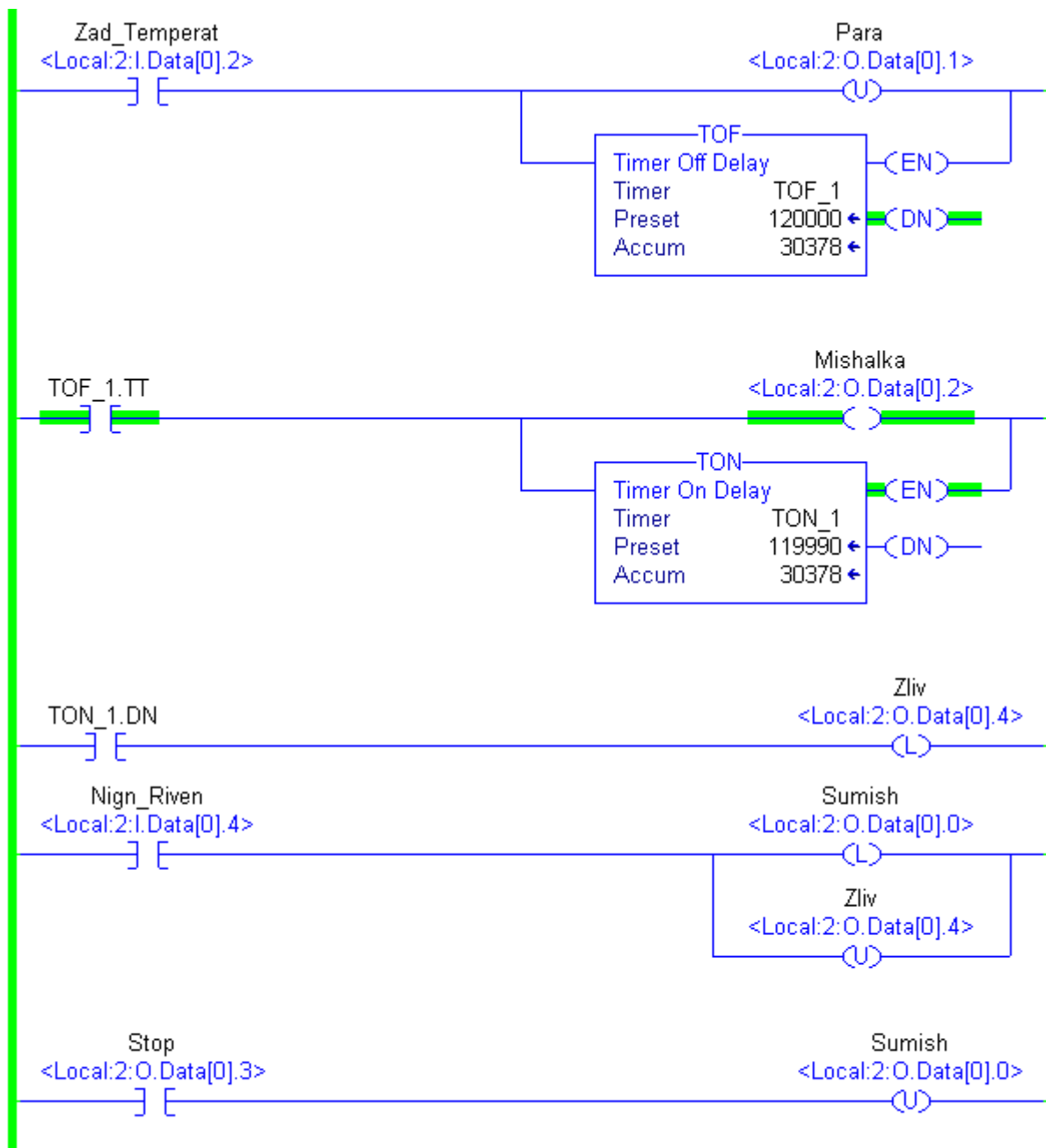
Для зміни стану контакту необхідно позначити його лівою клавішею миші, а правою клавішею відкрити меню, з якого вибрати Toggle Bit і знову натиснути ліву клавішу миші.

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.

Програмний код у режимі Run Mode має вигляд:







### ***Завантаження проекту у контролер***

Після перевірки роботи програми повернути проект, до того стану, який він мав перед симуляцією, тобто в проектному дереві у теці I/O Configuration змінити емулятор на реальний процесор з усіма потрібними уставками, встановити попередні модулі вводу і виводу, а також адреси змінних на проектному коді програми.

Для завантаження проекту в контролер необхідно в меню Communications вибрати Who Active і у відповідному діалоговому вікні, вибрати канал зв'язку з контролером. Якщо RSLinx ще не працює, він запуститься, аби можна було переглянути топологію мережі. Вибрати з дерева контролер. Клацнути лівою клавшею миші по кнопці Recent, у діалоговому вікні Select Recent Communication Path вибрати канал зв'язку і натиснути кнопку Download. Після підтвердження бажання, завантаження проекту в контролер відбудеться

негайно і з'являється повідомлення компілятора і інформація про завантаження[52,53].

## Тестові та контрольні запитання до розділу 10

**I-й рівень запитань та тверджень дозволяє перевірити загальний рівень засвоєння матеріалу розділу і передбачає відповідь на запитання або твердження “Так” чи “Ні”.**

1. Чи випускає корпорація Rockwell Automation у даний час три сімейства контролерів: MicroLogix/SLC, що орієнтовані на розв'язання невеликих задач управління, PLC-5 – для реалізації задач середньої складності і Logix - призначених для побудови великих систем управління?

2. При конфігуруванні контролера ControlLogix модулі вводу-виводу, процесори та інші модулі необхідно розташовувати в шасі у заданій послідовності.

3. У системі управління, яка створена на базі процесорів і шасі ControlLogix, а також мережі ControlNet, кожен модуль є самостійним вузлом мережі. Це дозволяє вільно розподіляти процесори по різних шасі системи управління.

4. Для приєднання джерела живлення до шасі на його боковій стороні розташований рознім і спеціальні напрямні, по яким джерело живлення вставляється доки його рознім не з'єднається з рознімом задньої шини шасі.

5. Чи будуть джерела живлення, які в номері серії мають літеру «С» сумісними з шасі серії «А» і «В»?

6. Особливістю мережної моделі постачальник – споживач є те, що модулі вводу-виводу можуть за запитом здійснювати інтелектуальний обмін даними з іншими пристроями системи.

7. Чи мають модулі вводу-виводу ControlLogix функції формування повідомлень про несправності на рівні модуля і діагностування помилки підключення електроустаткування?

8. Блокувальний перемикач призначений для блокування роботи модуля вводу-виводу.

9. Чи може бути альтернативою клемним блокам RTB монтажна система, в яку входять модулі інтерфейсу?

10. Чи призначені замикальні слоти з правого боку модуля вводу-виводу для встановлення в них механічних ключів, за допомогою яких задається адреса модулів у мережі?

11. Чи забезпечують дискретні модулі вводу-виводу з ізольованими каналами визначення обриву лінії, короткого замикання і електронний захист від нього, а також контроль за спрацьовуванням вихідного каскаду?

12. Модулі аналогового вводу можуть здійснювати масштабування сигналів до інженерних одиниць та формувати попередження при перевищенні їх значень встановленим межам.

13. Чи відносяться до спеціальних модулів в системі ControlLogix дискретні модулі вводу-виводу та інтерфейсні модулі?

14. Коли модулі вводу-виводу конфігуровані з декількома власниками, то при втраті будь-яким контролером сполучення з модулем, останній продовжить працювати і передавати дані в систему, оскільки він підтримує сполучення з іншим контролером-власником.

15. Якщо модуль вводу знаходиться в іншому шасі ніж контролер-власник, то передача та отримання модулем даних визначається двома параметрами конфігурації - запрошеним інтервалом пакета (RPI) і зміною стану (COS).

16. Якщо вхідний аналоговий модуль розташований у віддаленому шасі, то отримання даних контролером-власником залежатиме від типу мережі, яка використовується для підключення модуля.

17. При рековому спілкуванні контролера-власника з модулями вводу-виводу обмін даними здійснюється з RPI окремих модулів.

18. Чи можуть контролери ControlLogix працювати у різних мережах завдяки широкому набору інтерфейсних модулів?

19. Чи дозволяє мережна архітектура NetLinx виконувати наскрізне об'єднання всіх компонентів системи автоматизації: від декількох пристроїв однієї мережі до багатьох пристроїв в декількох мережах?

20. Інтерфейсні модулі в мережах без контролера ControlLogix не використовуються.

*Перевірте відповіді на сторінках: 1 - 262 с., 2 - 263 с., 3 - 263 с., 4 - 265 с., 5 - 266 с., 6 - 267 с., 7 - 267 с., 8 - 268 с., 9 - 267 с., 10 - 268 с., 11 - 268 с., 12 - 269 с., 13 - 269 с., 14 - 271 с., 15 - 271 с., 16 - 274 с., 17 - 274 с., 18 - 274 с., 19 - 275 с., 20 - 276 с.*

**II-й рівень запитань (тестові запитання) дозволяє перевірити рівень засвоєння матеріалу розділу із заглибленням у технічні особливості будови та структури контролерів ControlLogix і передбачає вибір правильної відповіді з наведених трьох варіантів.**

### **1. В чому основна відмінність контролерів ControlLogix?**

а) основною відмінністю є забезпечення контролерами дискретного управління та управління безперервними процесами, приводами і сервоприводами у поєднанні з комунікаціями і сучасним вводом-виводом;

б) основною відмінністю є архітектура контролера, що заснована на базі мережі ControlNet;

в) основною відмінністю є набір різноманітних модулів, які поєднуються за допомогою шасі вводу-виводу.

### **2. На які групи можна поділити контролери (процесори) ControlLogix?**

а) контролери MicroLogix/SLC – що орієнтовані на розв'язання невеликих задач управління; PLC - 5 – для реалізації задач середньої складності і Logix - призначених для побудови великих систем управління;

б) контролери модифікації 1756-L55Mxx і контролери 1756-L6x, які відрізняються об'ємом призначеної для користувача пам'яті та пам'яті вводу-виводу, швидкістю виконання програм управління;

в) контролери, що виконані на базі процесорів Intel Celeron та Pentium III.

**3. Як контролери 1756-L55Mxx і 1756-L6x відрізняються за швидкістю?**

а) контролери групи 1756L55Mxx виконують сканування релейної логіки в два рази швидше, ніж контролери групи 1756L6x, а виконання функціональних блоків, інструкцій управління рухом і математичних інструкцій з даними типу REAL –швидше в 4-5 разів;

б) контролери групи 1756L55Mxx виконують сканування релейної логіки в 4-5 разів швидше, ніж контролери групи 1756L6x, а виконання функціональних блоків, інструкцій управління рухом і математичних інструкцій з даними типу REAL –швидше в 2 рази;

в) контролери групи 1756L6x виконують сканування релейної логіки в два рази швидше, ніж контролери групи 1756L55Mxx, а виконання функціональних блоків, інструкцій управління рухом і математичних інструкцій з даними типу REAL –швидше в 4-5 разів.

**4. На скільки слотів розраховані шасі ControlLogix?**

а) шасі випускаються на 4, 7, 10, 13 та 17 слотів;

б) шасі випускаються на 2, 4, 8, 16 та 32 слота;

в) шасі випускаються на 4, 10, 16 та 32 слота.

**5. Яку напругу мають джерела живлення ControlLogix, що використовуються з шасі 1756?**

а) 120/220В змінного струму;

б) 120/220В змінного струму та 1,2В, 3,3В, 5В або 24В постійного струму;

в) 1,2В, 3,3В, 5В або 24В постійного струму.

**6. Що розташоване на передній стороні джерела живлення?**

а) дві клеми для підключення напруги і одна для заземлення, перемикач для підключення/відключення джерела живлення та один світлодіодний індикатор;

б) дві клеми для підключення напруги і одна для заземлення, круговий перемикач режимів роботи джерела живлення та два світлодіодних індикатори;

в) дві клеми для підключення напруги і одна для заземлення та світлодіодні індикатори.

**7. Як здійснюється установлення модулів вводу-виводу і підключення зовнішніх проводів?**

а) видалення і установлення модулів в шасі можливо тільки при вимкнені напруги живлення; монтаж вхідних і вихідних кіл модулів здійснюється за допомогою клемно-блокових з'єднувачів;

б) видалення і установлення модулів в шасі допускається без вимикання напруги живлення; монтаж вхідних і вихідних кіл модулів здійснюється за допомогою знімних клемників;

в) видалення і встановлення модулів в шасі можливо тільки при вимкненні напруги живлення; монтаж вхідних і вихідних кіл модулів здійснюється за допомогою знімних клемників.

**8. Що відноситься до основних елементів модуля вводу-виводу?**

а) корпус з рознімами на задній і передній панелі та знімний клемний (термінальний) блок RTB;

б) корпус з рознімами на задній і передній панелі, блокувальний перемикач і знімний клемний (термінальний) блок RTB;

в) корпус з рознімами на задній і передній панелі, індикатори, блокувальний перемикач і знімний клемний (термінальний) блок RTB.

**9. Які існують варіанти виготовлення дискретних модулів вводу-виводу?**

а) стандартне виготовлення та з діагностикою і електронним захистом;

б) стандартне виготовлення, з ізольованими каналами, з діагностикою і електронним захистом;

в) з ізольованими каналами, з діагностикою і електронним захистом.

**10. Які дискретні модулі мають найбільшу кількість модифікацій?**

а) модулі виводу змінного струму;

б) модулі виводу постійного струму;

в) модулі вводу постійного струму.

**11. Якщо дискретний модуль вводу-виводу знаходиться у тому ж самому шасі що і контролер-власник, то як може бути організована модулем передача даних?**

а) передача модулем даних визначається двома параметрами конфігурації – запрошеним інтервалом пакета (RPI) і зміною стану (COS);

б) передача модулем даних визначається одним параметром конфігурації – запрошеним інтервалом пакета (RPI) мережі;

в) передача модулем даних визначається одним параметром конфігурації – зміною стану (COS).

**12. За що відповідає параметр COS при передачі даних?**

а) параметр COS перевіряє цілісність переданих модулем даних і при необхідності ініціалізує повтор їх передачі;

б) параметр COS визначає швидкість, з якою модуль передає дані;

в) параметр COS інструктує модуль передавати дані кожного разу, коли певні входи змінюють свій стан з ON в OFF або ні.

**13. З якою частотою контролер-власник може передавати дані модулю виводу?**

а) тільки з частотою кожного циклу сканування програми;

б) у кінці кожного циклу сканування програми або з частотою, визначеною в RPI модуля;

в) тільки з частотою, визначеною в RPI модуля.

**14. Яка поведінка модуля виводу, коли він отримує дані з контролера-власника?**

- а) модуль миттєво здійснює колективну передачу команд виводу та одночасно передає отримані дані назад у мережу;
- б) модуль миттєво здійснює тільки колективну передачу команд виводу;
- в) миттєво передає контролеру-власнику інформацію про цілісність отриманих даних.

**15. Яку мережу підтримує модуль 1756-DNB?**

- а) модуль підтримує мережу DH+, яка дозволяє модулю виконувати роль сканера для передачі дискретних сигналів і блоків даних від/для пристроїв віддаленого вводу-виводу;
- б) модуль підтримує управляючу мережу ControlNet, яка відповідає вимогам високопродуктивних додатків, що працюють в режимі реального часу, та дозволяє створювати розподілені системи управління протяжністю до 6 км;
- в) модуль підтримує відкриту мережу нижнього рівня DeviceNet, яка забезпечує сполучення між датчиками і виконавчими механізмами та програмованими контролерами і комп'ютерами.

**16. Які модулі ControlLogix можуть мати більше одного контролера-власника?**

- а) вхідні та вихідні модулі;
- б) тільки вхідні модулі;
- в) тільки вихідні модулі.

**17. Які мережі підтримують контролери ControlLogix?**

- а) Ethernet/IP, CAN, DeviceNet, RS-485;
- б) Ethernet/IP, ControlNet, DeviceNet, DH+, RS-232;
- в) Ethernet/IP, ControlNet, DeviceNet.

**18. Для чого призначені модулі інтерфейсу SERCOS (1756-M03SE, 1756-M08SE, 1756-M016SE)?**

- а) для сполучення системи ControlLogix з інтелектуальними польовими пристроями;
- б) для сполучення контролерів ControlLogix з віддаленими пристроями вводу-виводу;
- в) для сполучення системи ControlLogix з інтелектуальними сервоприводами.

**19. Як реалізується резервування у ControlLogix?**

- а) апаратним та програмним шляхом;
- б) тільки апаратним шляхом;
- в) тільки програмним шляхом.

**20. За допомогою якого пакета програмуються контролери ControlLogix?**

- а) за допомогою пакета RSLogix 5000;
- б) за допомогою пакета RSLogix 5;
- в) за допомогою пакета RSLogix 500.

*Перевірте відповіді на сторінках: 1 – 262...263 с., 2 - 264 с., 3 - 264 с., 4 - 264 с., 5 - 266 с., 6 - 266 с., 7 – 269...270 с., 8 - 267 с., 9 - 268 с., 10 - 268 с., 11 -*

271 с., **12** - 271 с., **13** - 271 с., **14** - 272 с., **15** - 275 с., **16** - 270 с., **17** - 274 с., **18** - 276 с., **19** - 277 с., **20** - 279 с.

**III-й рівень запитань (контрольні запитання) дозволяє перевірити рівень засвоєння матеріалу із систематизацією отриманої у розділі інформації та передбачає відповіді, що побудовані на логічному аналізі або порівнянні із приведенням конкретних технічних параметрів (даних або прикладів).**

1. Роз'ясніть принцип конфігурування контролерів ControlLogix.
2. Розкрийте особливості архітектури контролерів ControlLogix у порівнянні з контролерами інших виробників.
3. Чим відрізняються принципи роботи контролера-власника ControlLogix з модулями вводу-виводу, які розташовані у віддаленому або локальному шасі?
4. Який принцип підбору джерела живлення ControlLogix під шасі?
5. Розкрийте поняття шасі ControlLogix і, як воно співвідноситься з поняттям шасі контролерів інших виробників.
6. Поясніть призначення параметрів RPI і COS при різних методах обміну даними.
7. Охарактеризуйте види та функції спеціальних модулів ControlLogix.
8. Поясніть призначення різних видів мереж, які підтримує контролер ControlLogix.
9. Які методи резервування підтримуються у ControlLogix?
10. Поясніть з яких етапів складається розробка проекту для контролера ControlLogix у середовищі RSLogix 5000.

*Перевірте відповіді на сторінках: 1 – 264...276 с., 2 – 262...264 с., 3 – 270...274 с., 4 - 266 с., 5 – 264...266 с., 109...112 с., 145...153 с., 166...168 с., 195...197с., 233...236с., 6 – 270...274 с., 7 – 269...270 с., 8 – 274...277 с., 9 – 277...279 с., 10 – 279...290 с.*

## **11 МІКРОПРОЦЕСОРНІ КОНТРОЛЕРИ ВИРОБНИЦТВА GE FANUC**

Одним з лідерів на ринку засобів автоматизації є GE Fanuc - спільне підприємство відомого концерну General Electric (США) і японської корпорації Fanuc. Продукцією GE Fanuc і його служб підтримки є програмні продукти SIMPLICITY, високоякісні програмовані логічні контролери, широкий діапазон пристроїв ЧПУ, пристроїв контролю переміщення, інтерфейси оператора, промислові комп'ютери, лазери і широкий перелік послуг з сервісного обслуговування споживачів.

Гама контролерів, що розробила GE Fanuc складається з ПЛК VersaMax, VersaMax Micro, ПЛК Series 90-30 і 90-70, а також контролерів нового покоління PACSystems RX3i та PACSystems RX7i.

Контролери ПЛК VersaMax, VersaMax Micro мають просте управління, мінімальні вимоги до об'єму пам'яті і обслуговують менше 100 каналів вводу-виводу.

ПЛК Series 90-30 – контролери середнього рівня, мають управління помірної складності, підтримують 100-512 каналів вводу-виводу і здатні працювати у розподілених АСУТП.

ПЛК Series 90-70 і прості конфігурації контролерів PACSystems RX3i, PACSystems RX7i-у призначені для виконання складних задач. Вони використовують високошвидкісну передачу даних, мають 2000-5000 каналів вводу-виводу і високу швидкість реагування на зміну інформації, що надходить.

Складні конфігурації PACSystems RX3i, PACSystems RX7i-у використовуються для дуже складних задач, таких як автоматизація автомобільного цеху фарбування та системи вимкнення атомного реактора.

Як середовище програмування для всіх контролерів, операторських панелей, систем збору даних і промислових комп'ютерів виробництва GE Fanuc застосовується універсальний засіб Proficy Machine Edition (ME). Цей пакет є одним з компонентів загальної програмної технології Proficy, призначеної для комплексної автоматизації підприємства від рівня контролерів до MES.

### **11.1 Контролери PACSystems RX7i**

Останніми роками деякі виробники ПЛК розпочали випуск нових пристроїв промислової автоматики – програмованих автоматизаційних контролерів PAC (Programmable Automation Controller). Пристрої PAC, як і PC-сумісні контролери мають відкриту архітектуру, а тому багато в чому схожі. Проте потужні виробники засобів автоматизації, до яких відноситься компанія GE Fanuc, не дуже звертали увагу на PC-сумісний напрям у створенні програмованих логічних контролерів, доки з розвитком одноплатних комп'ютерів не з'явилась нагода створювати відносно не дорогі потужні контролери на базі Інтеловських процесорів, комп'ютерних системних шин, стандартних операційних систем і т.ін. Тобто поки ПЛК з відкритою



архітектурою реально не наблизилися до можливостей управління за допомогою промислових комп'ютерів.

PAC-пристрої мають архітектуру ПК, потужні процесори, великий об'єм пам'яті, але як і ПЛК, призначені для розв'язання завдань безпосереднього цифрового управління.

Новітніми розробками компанії GE Fanuc, що створені саме на основі стандартної відкритої архітектури PAC є контролери PACSystems RX3i і PACSystems RX7i. В цих засобах автоматизації використовуються високопродуктивні процесори Intel з частотою 1800/700/300 МГц, а також високошвидкісні PCI (PACSystems RX3i) 27МГц та інтеграційні VME64 (PACSystems RX7i) шини. Ці контролери прийшли на зміну популярним моделям Series 90-70/90-30. При цьому була забезпечена спадкоємність з модулями вводу-виводу контролерів попереднього покоління.

Поява контролерів PACSystems ознаменувала собою революційний перелом в підході до розробки систем автоматизації, забезпечивши інтегроване управління замість механічного складання окремих компонентів і вузлів. За допомогою стандартного блока управління і єдиного інструментарію користувачі дістали можливість використовувати ефективно середовище для проектування різноманітних застосувань.

Контролер RX7i є першим з сімейства програмованих контролерів PACSystems. Модельний ряд цього контролера орієнтований на розв'язання задач середнього і високого технічного рівня, які потребують великих ресурсів пам'яті, пропускну здатності і розподілених засобів вводу-виводу. PACSystems – це рішення «3 в 1» – управління, візуалізація пересування.

PACSystems RX7i, рис.11.1, як і Series 90-70 створені за найбільш

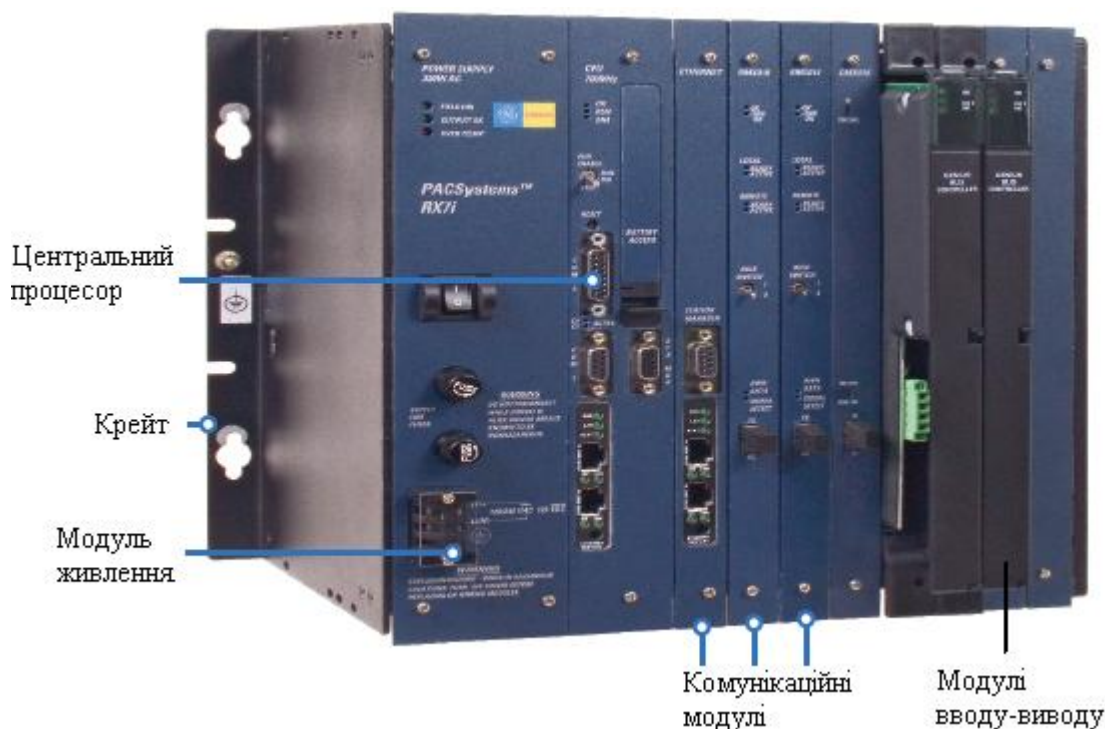


Рисунок 11.1 – Контролер PACSystems RX7i

поширеним у Європі стандартом VME, відповідно до якого при створенні контролерів використовується магістрально-модульна архітектура. Контролери на базі шини VME виготовляються в конструктивах <Евромеханіка 19"> форматів 3U и 6U. Цей стандарт дозволяє збирати контролери з готових плат різних фірм, які випускають продукцію за цим стандартам.

За допомогою PACSystem RX7i можна розробляти системи високої надійності HSB (Hot Standby). Основа таких систем – синхронізація даних основного і резервного ЦП за допомогою технології CMX (Control Memory eXchange). Таке рішення є «спільною пам'яттю по проводу», при якій обмін між процесорами відбувається по оптоволоконному кабелю зі швидкістю більше ніж 2Гбод, на відстані до 300м і без уповільнення виконання програми управління [8,54,55].

### 11.1.1 Сійки PACSystems RX7i

Базовою основою системи управління PACSystems RX7i є сійка, яка має 9 (IC698CHS009, C698CHS109), 17 (C698CHS217) або 18 (C698CHS017, C698CHS117) слотів. Якщо кількість модулів перевищує можливості типових конструктивів, то для розширення системи вводу-виводу використовуються стандартні сійки контролерів Series 90-70 (максимум 7 сійок) на 5 (IC697CHS750) або 9 (IC697CHS790, IC697CHS791) слотів, а також сійки інтегратора VME ємністю 17 (IC697CHS782, IC697CHS783) слотів.

Сійка RX7i і всі сійки розширення мають верхню об'єднувальну панель J1, яка розрахована на 16- і 24-бітові адреси і передачу даних з розрядністю 8 і 16 бітів. Окрім того, основна сійка PACSystems додатково має нижню об'єднувальну панель J2, яка надає контролеру шини для 32-бітової адресації, 32-бітових даних і додаткового блока живлення постійного струму. Якщо в сійці розширення необхідно встановити модулі, для яких потрібна нижня об'єднувальна панель J2, то її вмонтовують. Усі об'єднувальні панелі мають різні (слоти) для встановлення модулів. Сійки контролерів PACSystems RX7i підтримують нові потужні модулі живлення і високопродуктивні центральні процесори, а вмонтована шина VME64 забезпечує пропускання інформації в 4 рази більше, ніж шина VME, що в Series 90-70. Завдяки цьому ввід-вивід даних здійснюються значно швидше. Шина VME64 підтримує всі стандартні модулі VME, у т.ч. модулі вводу-виводу Series 90-70 і модулі VMI.

Сійка на 18 слотів має висоту - 283 мм, ширину - 483 мм і глибину - 184 мм. Ширина всіх слотів – 20,3 мм, за винятком слота блока живлення, ширина якого – 60,96 мм, рис.11.2.

У нульовий слот сійки RX7i встановлюється блок живлення, а у слоти 1 і 2 – модуль центрального процесора з дочірньою платою Ethernet. Решту слотів (3 - 17) можна використовувати для однієї з наступних комбінацій вводу-виводу:

- 15 модулів одинарної ширини;
- 8 модулів подвійної ширини;

- комбінація модулів подвійної і одинарної ширини.

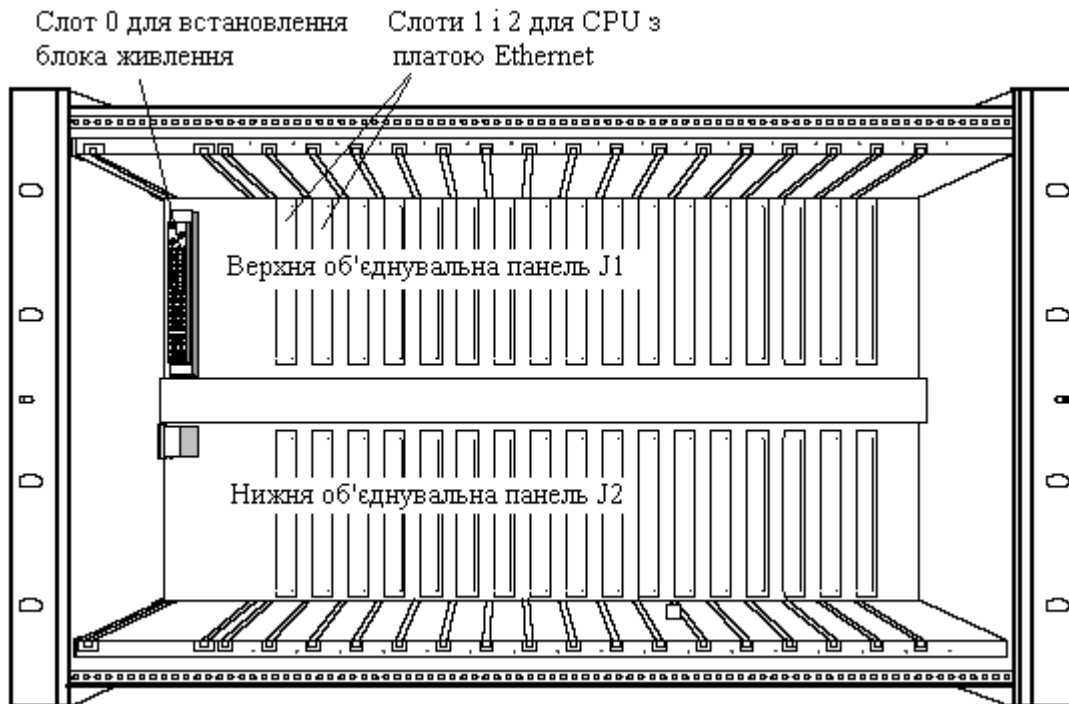


Рисунок 11.2 – Стійка PACSystems RX7i

Стійка з встановленими блоком живлення і центральним процесором з платою Ethernet кваліфікується, як крейт. Основний крейт системи PACSystems RX7i може використовуватися для всіх комбінацій центральних процесорів, модулів RX7i і модулів VME одинарної ширини, а також стандартних модулів Series 90-70, які в основному крейті використовують два слоти. Коли модуль Series 90-70 встановлюють в стійку перший раз, його замкова колодка автоматично фіксується на центральній направляючій об'єднувальній платі і залишається там після видалення модуля. У такий слот інший модуль установити не можна.

Для адресації модулів ніякі перемички на стійці або DIP- перемикачі в модулях вводу-виводу не потрібні.

Незайняті слоти між модулями вводу-виводу в основному крейті PACSystems автоматично ретранслюють сигнали шини VME, тому установлення перемичок на пустих слотах не потрібне.

Для охолодження модулів в стійку можна встановити комплект вентиляторів. Зовнішній вигляд компонованого контролера приведений на рис.11.3.

У крейтах розширення Series 90-70 можуть встановлюватися ті ж типи модулів GE Fanuc, що і в основному крейті. Крейти інтегратора VME Series 90-70 забезпечують просту інтеграцію в систему модулів VME інших виробників.

У стандартному крейті Series 90-70 можуть бути розміщені модулі подвійної ширини, наприклад, стандартні модулі вводу-виводу Series 90-70 і модулі VME подвійної висоти формату 6U. Модулі VME одинарної висоти (3U)

можуть використовуватися, якщо до цього модуля приєднана перехідна лицева панель.

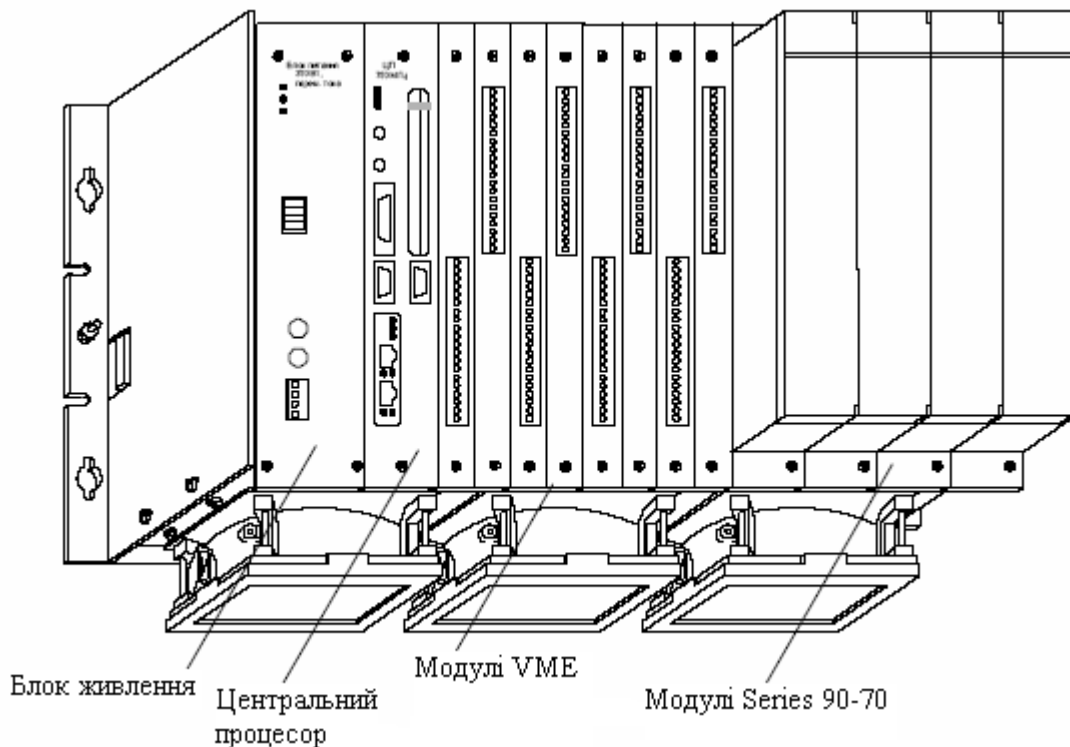


Рисунок 11.3 – Стійка RX7i з установленими модулями VME і модулями Series 90-70

Модулі типів 3U і 6U підключаються до об'єднувальної панелі J1. Деякі модулі типу 6U можуть підключатися також до об'єднувальної панелі J2. Рознім модуля, який підключається до об'єднувальної панелі J1, називається P1, а рознім модуля, який підключається до об'єднувальної панелі J2, називається P2.

Кількість модулів в стійці обмежується потужністю блока живлення.

У крейті не повинно бути пустих слотів між модулями GE Fanuc, а також невикористаних слотів зліва від будь-якого модуля VME, який формує переривання VME. Якщо слот не використовується, він має бути закритий рознімом, який передає сигнали переривання по шлейфу. Всі модулі VME інших виробників повинні бути встановлені праворуч від модулів GE Fanuc.

Крейти розширення VME в системі зв'язуються з основним крейтом і ЦПУ за допомогою модулів передавача і приймача шини[56,57].

### 11.1.2 Модулі центрального процесора

У якості CPU контролери PACSystems RX7i використовують процесори Intel Celeron і Pentium III, які оснащені годинниками реального часу. Для збереження даних і програм CPU контролерів разом з незалежною флеш-пам'яттю об'ємом до 10Мбайт мають ще 10 (IC698CPE010, IC698CPE010) або

64(IC698CPE0400) Мбайт оперативної пам'яті і працюють з тактовою частотою, відповідно, 300 і 700 МГц. Додатково користувачі можуть зберігати в пам'яті опис алгоритмів і програми, а також машинну документацію у форматах Word, Excel, PDF, систем автоматизованого проектування CAD та інші файли.

Центральний процесор через об'єднувальну плату VME64, що змонтована в стійці, обмінюється даними з модулями вводу-виводу і інтелектуальними модулями. Забезпечуючи у 4 рази вищу швидкість системної шини (до 14 Мбайт за секунду), порівняно з існуючими системами на базі VME, і до 64 Мбайт пам'яті, контролери PACSystems RX7i, підтримують всі стандартні модулі VME, у тому числі модулі вводу-виводу Series 90-70 і модулі VME, що випускаються іншими виробниками. Один процесорний модуль здатний обслуговувати до 32000 каналів вводу-виводу

Модуль центрального процесора, рис.11.4, має три незалежних вбудованих послідовних порти, доступ до яких здійснюється через розніми на передній панелі модуля.



Рисунок 11.4 – Модуль центрального процесора PACSystems RX7i

Два порти використовуються для модернізації вбудованих в процесор мікропрограм і для приєднання зовнішніх пристроїв. Порт №1 підтримує інтерфейс RS-232, а порт №2 - інтерфейс RS-485. Третій послідовний порт RS-232 використовується для управління інтерфейсом Ethernet з боку станції диспетчера. Він має здвоєні порти RJ-45, які послідовно приєднують контролер до інших пристроїв Ethernet через комутатор, оснащений функцією визначення необхідності завантаження процесора чи передачі інформації у мережу. Кожний

порт автоматично визначає швидкість передачі даних (10 Мбіт/с або 100 Мбіт/с), тип дуплексного режиму (напівдуплекс або повний дуплекс) і тип кабельного з'єднання (пряме або перехідне з'єднання) лінії. Інтерфейс Ethernet центрального процесора забезпечує можливість дистанційного контролю за роботою системи управління за допомогою Web-браузера.

Для забезпечення енергонезалежної пам'яті ОЗП на лицевій поверхні модуля центрального процесора під відкидною кришкою знаходиться слот, де розташована батарейка.

Після перевірки програми кулісний перемикач, що знаходиться поряд зі слотом батарейки переводиться в одне з положень, відповідно до вибраного режиму роботи - RUN EN (режим роботи з активізованими виходами), RUN DIS (режим роботи із заблокованими виходами) або STOP (зупинник).

Центральний процесор має 5 світлодіодних індикатори: ОК – стан процесора, ENA – активізація процесора, RUN – робочий режим і два індикатори станів портів 1 і 2. Вбудований інтерфейс Ethernet – 7 індикаторів, які призначені для індикації стану різних функцій [56,57].

### 11.1.3 Блоки живлення

У складі ПЛК RX7i є два блоки живлення на 85 - 264В змінного струму. Один з них (IC698PSA100) має вихідну потужність 100 Вт, а інший (IC698PSA350) – 350Вт. Окрім того є блок живлення (C698PSD300) на 24В постійного струму з вихідною потужністю 300Вт. Вихідна напруга першого блока дорівнює 5В постійного струму 0-20А, -12В постійного струму 0-1А і 12В постійного струму 0-2А. Напруга блока живлення другої модифікації також 5В, але постійного струму 0-60А, -12В постійного струму 0-4А і 12В постійного струму 0-12А. Третій блок живлення характеризується струмом у 50А при вихідній напрузі 5В, струмом у 10А при вихідній напрузі 12В і струмом у 4А при вихідній напрузі -12В.

Блок живлення IC698PSA100 забезпечує повну потужність 100 Вт без примусового повітряного охолодження при температурі навколишнього середовища 0 - 60°C, рис.11.5. Блоки живлення IC698PSA350 і C698PSD300, що мають потужність більше 300 Вт, використовуються при збільшеній кількості модулів. Для цих блоків потрібне примусове повітряне охолодження, яке забезпечується вентиляторами, що монтуються на шасі у нижній частині стійки.

Блоки живлення IC698PSA350 і C698PSD300 мають датчики їх внутрішньої температури і наявності повітряного потоку від вентилятора. Коли вони перегріваються, вмикається індикатор OVER TEMP і всі виходи блокуються. Якщо при перегріві виявляється також відсутність повітряного потоку, всі виходи блокуються навіть, коли він вимкнений.

Вмикання каналів виконується автоматично, після повернення температури середини блока у заданий діапазон.

У верхній частині передньої панелі блока живлення розміщені світлодіодні індикатори:

FIELD OK - горить зеленим, якщо вхідна напруга змінного струму відповідає заданому діапазону;

OUTPUT OK – горить зеленим, якщо робочі параметри всіх трьох вихідних каналів постійного струму знаходяться в заданих межах. Гасне у разі відмови будь-якого з трьох вихідних каналів;

OVER TEMP (тільки у блоці живлення IC698PSA350) – горить червоним, якщо перевищена гранична температура або датчик наявності повітряного потоку зафіксував збій в подачі повітря.



Рисунок 11.5 – Блок живлення IC698PSA100

У середній частині знаходиться двопозиційний перемикач On/Off, який перемикає логічні рівні вихідної напруги за рахунок активізації або блокування вихідних каналів. Вимкнути вхідну напругу змінного струму за допомогою цього перемикача неможливо.

Змінні плавкі запобіжники, що знаходяться нижче перемикача, встановлені на активному і нейтральному входах лінії змінного струму.

У нижній частині лицевій панелі розташовані клеми L1 і L2/N для приєднання проводів напруги живлення 120/240В змінного струму і клемна колодка заземлення GND.

Для стійок розширення призначені спеціальні модулі живлення (IC697PWR710, IC697PWR711, IC697PWR724, IC697PWR748), які встановлюються в крейт ПЛК, як звичайні модулі вводу-виводу. Вони мають різні напруги живлення і потужності. Один такий блок можна одночасно використовувати для двох стійок розширення однієї і тієї ж головної стійки RX7i. Для приєднання блока живлення застосовується подовжувальний кабель IC697CBL700. Живити дві стійки від одного блока живлення можна, тільки якщо в іншій стійці використовується напруга не більше 5В постійного струму і струм не більше 5,2А [56,57].

#### **11.1.4 Модулі вводу-виводу**

Система управління PACSystems RX7i пропонує велику кількість модулів вводу-виводу, серед яких універсальні модулі аналогового вводу, де кожний канал може бути конфігурований на струм, напругу, резистивний термодатчик, термопару або опір. Є модулі високошвидкісних лічильників і модулі управління переміщенням, модулі аналогового і дискретного вводу-виводу високої щільності. Серед модулів є модулі одинарної і подвійної ширини.

Модулі дискретного вводу PACSystems забезпечують сполучення контролера із зовнішніми пристроями вводу, у тому числі з безконтактними датчиками, кнопками, перемикачами і маніпуляторами двійково-десятькового коду. Серед модулів дискретного вводу є 16-ти канальні (IC697MDL240, IC697MDL251) і 32-х канальні (IC697MDL252, IC697MDL253, IC697MDL254, IC697MDL250). Усі вони працюють з вхідними сигналами змінного струму. Модулі IC697MDL252, IC697MDL253, IC697MDL254 орієнтовані на вхідний сигнал 12В, 24В і 48В, решта – 120В.

Модуль дискретного вводу IC697MDL671 має 14 каналів, що працюють за перериваннями, а 64-х канальний модуль дискретного вводу IC697VDD100 має багатофункційний інтелектуальний контролер.

Усі модулі вводу, окрім одинарного 64-х канального, мають конструкції подвійної ширини.

Модулі дискретного виводу забезпечують сполучення контролера із зовнішніми пристроями виводу, у тому числі з контакторами, проміжними реле, дисплеями з відображенням у двійково-десятьковому коді і індикаторними лампами.

Модулі дискретного виводу поділяються на транзисторні модулі з вихідною напругою змінного струму (IC697MDL350, IC697MDL340, IC697MDL341), модулі з вихідною напругою постійного струму (IC697MDL753, IC697MDL752, IC697MDL750, IC697MDL740, IC697VDQ120) і модулі з релейними виходами (IC697MDL940, IC697VDR150, IC697VDR151). Серед них є 8-ми канальні (IC697VDQ120), 12-ти канальні (IC697MDL341), 16-ти канальні (IC697MDL340, IC697MDL740, IC697MDL940), 32-х канальні (IC697MDL350, IC697MDL753, IC697MDL752, IC697MDL750, IC697VDR150) і



64-х каналні (IC697VDR151). Модулі IC697MDL350, IC697MDL340, IC697MDL341 і IC697MDL753 мають подвійну ширину.

Для керування безперервними процесами в контролерах PACSystems RX7i пропонуються велика кількість легких у використанні аналогових модулів, які мають широкий вибір по кількості і типу вхідних і вихідних сигналів. Серед аналогових модулів вводу є 8-ми (IC697ALG230, IC697VRD008), 16-ти (IC697ALG440, IC697ALG441, IC697VAL216), 32-х (IC697VAL132 IC697VAL134, IC697VAL232)- і 64-х (IC697VAL264) каналні. Модуль IC697ALG230 конфігурується на струм або напругу, модулі IC697VAL132 і IC697VAL134 кваліфікуються, як ізольована плата 12-бітного 32-канального струмового аналогоцифрового перетворювача з клемною колодкою під гвинт і функцією тестування, а модуль IC697VRD008 - є інтелектуальною платою вводу аналогового сигналу тензорезисторів і тензомостів з клемною колодкою під гвинт.

При виборі модуля аналогового виводу треба враховувати, що серед них є 4-х каналні (IC697VAL304 і IC697VAL324), перший з них має на виході біполярну напругу, а інший – уніполярну), 8-ми каналні (IC697VAL308, IC697VAL348 і IC697VAL328, перші два мають біполярну напругу, а третій уніполярну) і 32-х каналний модуль аналогового виводу по напрузі з функцією тестування. 4-х каналний IC697VAL314 і 8-ми каналний IC697VAL318 модулі мають на виході струмовий уніфікований сигнал 4-20мА, а вихідні модулі IC697ALG320 і IC697VAL306 можуть конфігуруватися як на струм так і на напругу, перший з них 4-х каналний, а другий-16-ти каналний.

Для адресації модулів не потрібно використовувати перемички або DIP-перемикачі. Адресація модулів визначається номером слота в стійці, де вони встановлені.

Система управління PACSystems RX7i підтримує більшість існуючих модулів вводу-виводу Series 90-70, які мають подвійну ширину і оснащені знімними рознімами з клемми для монтажу зовнішньої проводки. Це дозволяє зручним способом приєднувати проводи до пристроїв вводу-виводу і замінювати модулі на місці експлуатації, не порушуючи існуючу проводку. Знімні розніми поставляються з вбудованими фіксаторами, які забезпечують їх надійне з'єднання з модулями вводу-виводу.

Окрім модулів вводу-виводу контролери PACSystems мають декілька спеціальних модулів, серед яких високошвидкісний лічильник IC697HSC700, програмований модуль співпроцесора IC697PCM711, однослотовий модуль жорсткого диска для шини VME IC697VHD001[56,57].

### **11.1.5 Засоби віддаленого вводу-виводу і комунікацій Genius і Field Control**

При створенні АСУТП в масштабах виробництва використовуються промислові мережі з високою надійністю передачі даних. Для розв'язання подібних завдань корпорація GE Fanuc розробила систему віддаленої периферії

Genius і Field Control, а також мережу Genius, яка об'єднує її і контролери Series 90-70/90-30 в закінчену систему.

Оскільки PACSystems RX7i/3i прийшла на заміну контролерам Series 90-70/90-30, то і всі напрацювання, що пов'язані зі створенням промислової мережі, даною системою підтримуються. Додатково для реалізації віддаленого вводу-виводу в системі управління PACSystems RX7i розроблені контролери шини Genius (IC697BEM731, IC697BEM734), Genius-сканер (IC697BEM733) та передаючий (IC697BEM713) і приймальний (IC697BEM711) модулі розширення.

Контролер шини Genius і сканер віддаленого вводу-виводу забезпечують інтегрування ПЛК GE Fanuc в мережу Genius і взаємодію з віддаленою периферією Genius і Field Control, рис.11.6, а передавальний і приймальний модулі - для розширювання центральної стійки, рис.11.7.

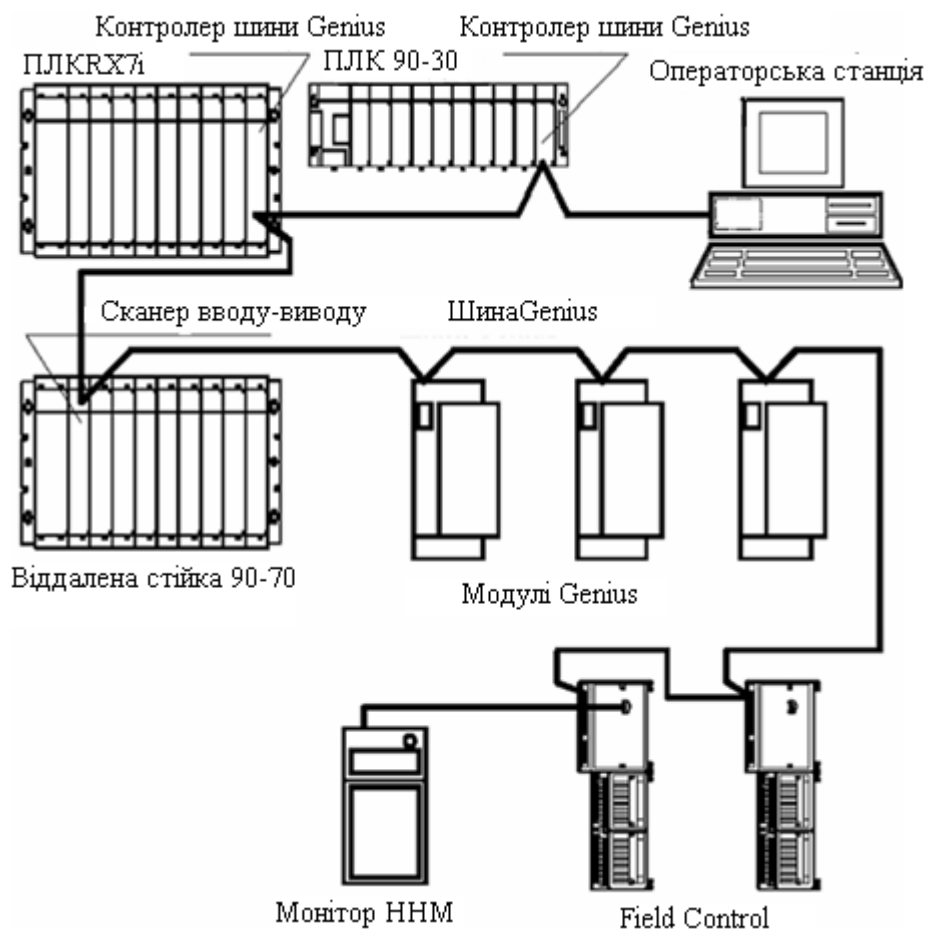
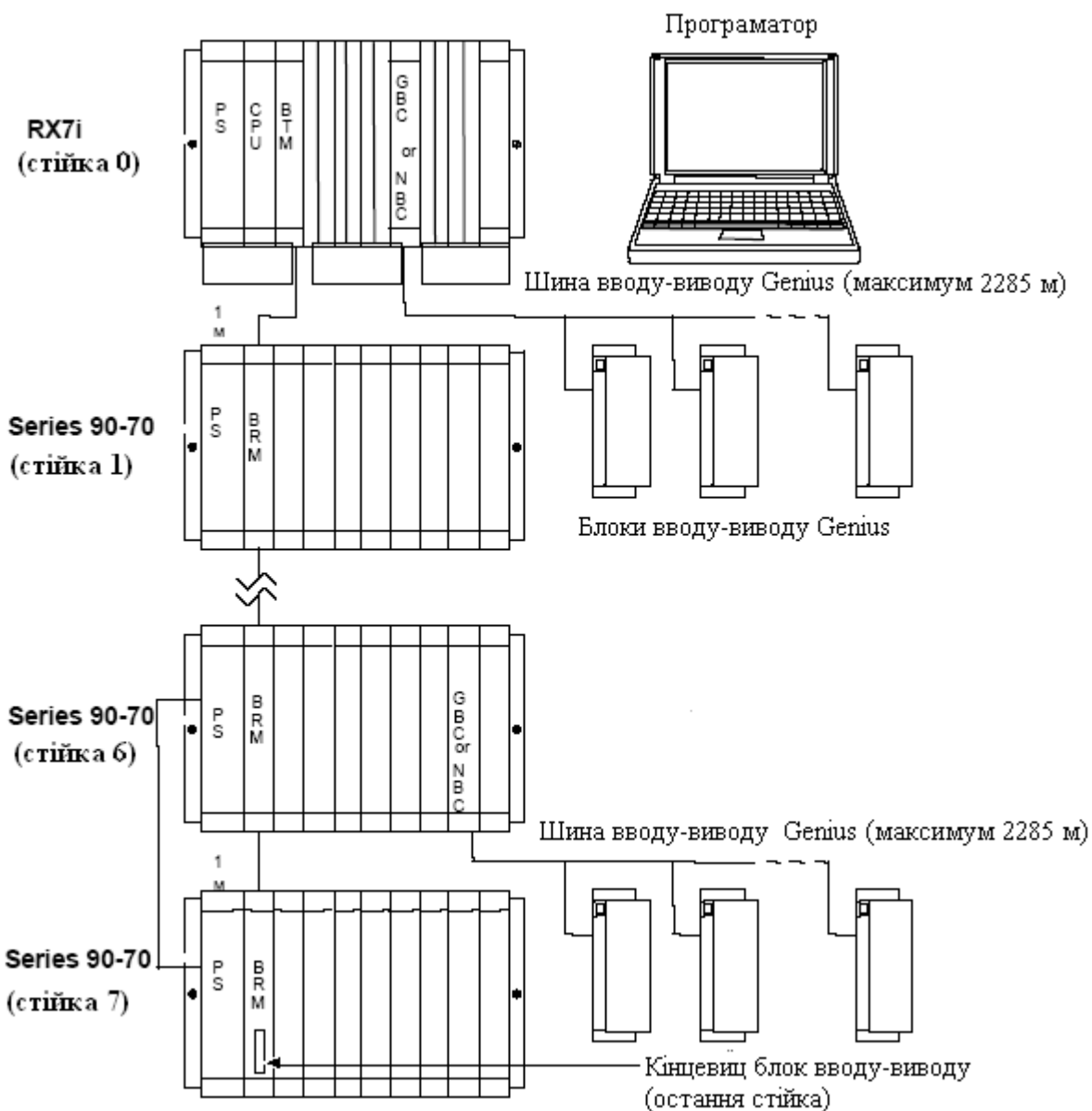


Рисунок 11.6 – Основні складові мережі Genius

Мережа Genius має топологію "шина". Пристрої в мережі об'єднуються за допомогою екранованої витії пари або волоконно-оптичного кабелю. Максимальна довжина шини - 2,3 км, швидкість обміну - 38,4 КБод. При довжині лінії до 600 м максимальна швидкість передачі даних досягає 153,6 КБод.



*CPU – центральний процесор; BRM- приймальний модуль шини, VEM711;  
 BTM – передавальний модуль шини, VEM713; GBC/NBC – шинний контролер, VEM731/734;  
 PS – блок живлення.*

Рисунок 11.7 – Конфігурація системи управління

До однієї шини Genius можна підключити до 32 пристроїв. До складу одного ПЛК можуть бути включені декілька контролерів шини, кожен з яких підтримує окремий сегмент мережі.

Коли компанія GE Fanuc випустила інтелектуальну систему вводу-виводу Genius, експерти назвали її першим крупним проривом в області технології вводу-виводу з часу винаходу програмованого контролера. Модулі вводу-виводу Genius здатні не тільки знімати інформацію з об'єкту, а і додатково надавати діагностичну інформацію про зовнішні підключення, умови електроживлення і навантаження, а також про стан мереж зв'язку, модулів і електричних кіл. Засоби діагностики Genius різко зменшують час, необхідний

на початкову конфігурацію і налагодження системи. При підключенні до резервованої системи, коли два або більше центральних процесорів працюють одночасно, модулі Genius можуть автоматично перемикатися на резервний CPU при збої основного центрального процесора. Всі ці властивості системи Genius зробили її незамінною в системах підвищеної надійності і протиаварійного захисту.

У мережі Genius підтримується обмін даними між контролерами за допомогою Global Data і Datagram. Для обміну даними за Global Data кожному контролеру у складі мережі виділяється частина адресного простору, куди він передає дані, що вказані при конфігурації його шини. Передача даних здійснюється без адресації контролера, який повинен їх отримати, оскільки адресний простір доступний для читання всім підключеним до шини PLC. Як результат, для всієї мережі створюється єдиний набір даних для обміну. Один контролер шини забезпечує передачу до 128 байтів Global Data і прийом до 128 байтів Global Data від кожного з вузлів.

Datagram є направленим посиланням даних від одного контролера до іншого. Прийом і передача Datagram відбувається під управлінням програми користувача. Момент відправлення Datagram може бути заданий з необхідною періодичністю або після виникнення якої-небудь події. Можливе спільне використання для зв'язку Global Data і Datagram.

Система вводу-виводу Field Control, як і Genius, працює з аналоговими і дискретними сигналами, сигналами від термопар і термометрів опору. Field Control має модульну конструкцію і складається з блока інтерфейсу VIU (Bus Interface Unit), блока польових контактних пристроїв (шасі вводу-виводу) і польових модулів вводу-виводу. У складі VIU є інтерфейс для сполучення з хост-шинами Genius і FIP. Універсальні блоки польових контактних пристроїв, які встановлюються на DIN-рейку або на панель існують в різних конфігураціях. Один блок інтерфейсу може підтримувати до 4 блоків контактних пристроїв або 8 модулів вводу-виводу, загальною кількістю 128 точок.

У номенклатуру Field Control також входить Micro Field Processor (MFP) - модуль процесора, який виконує логічну обробку сигналів вводу-виводу за програмою користувача, що дозволяє реалізувати управління підключеним до пристрою Field Control агрегатом без участі CPU ПЛК. Модуль MFP встановлюється в шасі вводу-виводу.

Поряд з VIU, призначеними для сполучення з контролерами GE Fanuc по мережі Genius, випускаються VIU для роботи в мережі Profibus, Interbus. Це дозволяє використовувати модулі вводу-виводу Field Control у складі АСУТП на базі контролерів третіх фірм.

Для конфігурування і діагностування мережі Genius, віддаленої периферії Genius і Field Control застосовується портативний монітор (ННМ). Підключивши його до одного з вказаних пристроїв в мережі, можна конфігурувати всі її складові. Живлення ННМ здійснюється від адаптера мережі ~220В, або від вбудованої акумуляторної батареї.

Серія RX7i характеризується також різноманітністю додаткових пристроїв зв'язку для розподіленого управління і/або вводу-виводу, таких як Ethernet EGD, Profibus-DP, Genius і DeviceNet.

Модуль інтерфейсу Ethernet IC698ETM001 встановлюється у стійку центрального процесора RX7i додатково до інтерфейсу Ethernet, вбудованого в процесор. Цих модулів у стійці може бути не більше трьох.

Модуль має два порти Ethernet і один послідовний - RS232. Порти Ethernet- дуплексні 10BaseT/100BaseT/TX (рознім RJ-45) з вбудованим мережним комутатором, який для спрощення обміну даними між вбудованим портом Ethernet і двома зовнішніми портами автоматично визначає і узгоджує швидкості передачі.

Наявність порту BaseT дозволяє напряму приєднуватися до концентратора або ретранслятора мережі BaseT (вита пара) без використання зовнішнього приймача-передавача і здійснювати обмін даними у форматі глобальних даних Ethernet (EGD) або відповідно до протоколу TCP/IP за допомогою таблиць жорсткої маршрутизації(SRTP);

Послідовний порт – це порт диспетчера станції, через який надається повний набір послуг з програмування і конфігурування контролера[56,57].

### 11.1.6 Гаряче резервування ПЛК

Гаряче резервування ПЛК дозволяє продовжувати роботу, у випадку якщо відмовив один з двох ПЛК. Система гарячого резервування, рис.11.8,

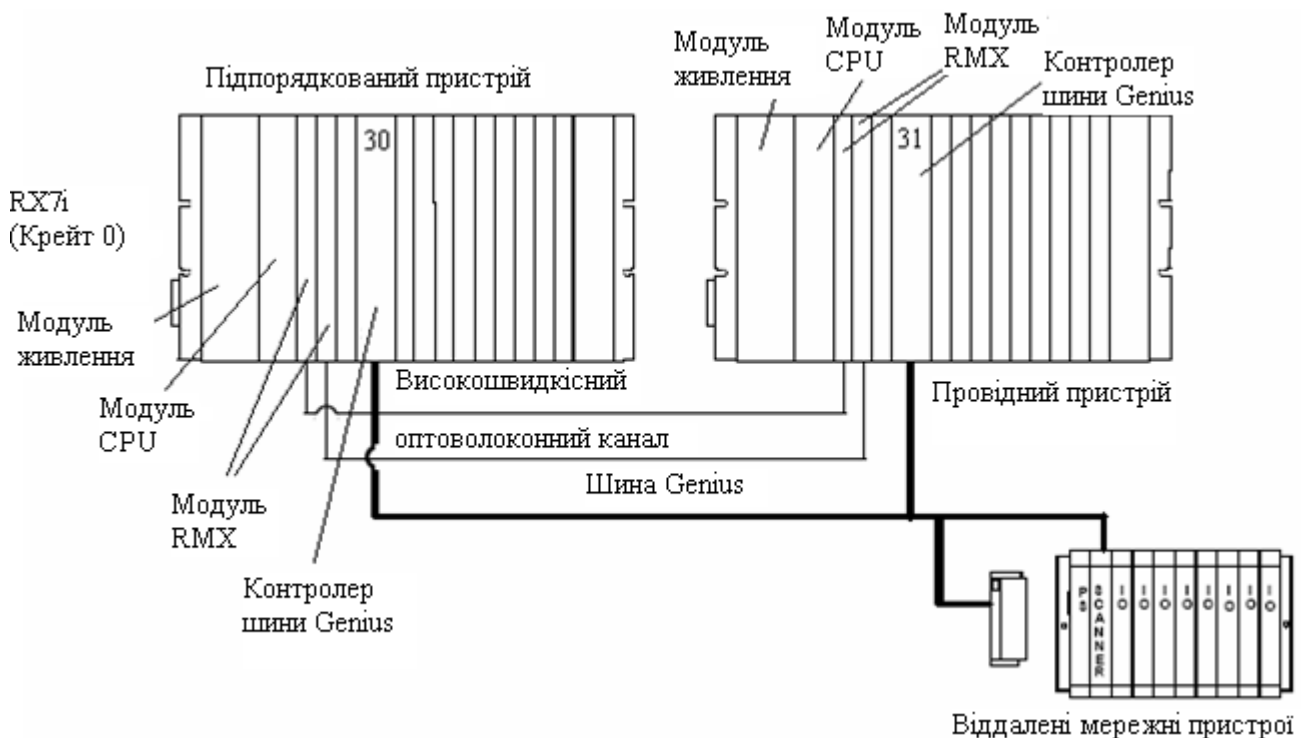


Рисунок 11.8 – Система гарячого резервування використовує два центральних процесора (ЦП) - активний ЦП, який безпосередньо управляє процесом і резервний ЦП, який синхронізований з активним ЦП і може «перехопити» управління при необхідності.

Кожен контролер (провідний і підпорядкований) повинен мати у своєму складі спеціальний резервований центральний процесор (IC698CRE020), який встановлений в перший слот нульового крейта, і один або два модулі RMX (IC698RMX016), що забезпечують канали зв'язку між двома контролерами.

Резервований центральний процесор IC698CRE020, як і стандартні CPU RX7i має 10Мб пам'яті (RAM), яка підтримується батареєю, 10Мб флеш-пам'яті, 2 порти, що конфігуруються - COM1 (RS-232) і COM2 (RS-485), вбудовану інтерфейсну плату Ethernet з двома портами 10 BASE T/100 BASE TX і порт станції управління оператора (RS-232). На лицевій панелі цього модуля, розташовані 5 світлодіодів центрального процесора і 7 світлодіодів, що показують стан інтерфейсу Ethernet. Функціонуванням резервованого центрального процесора можна управляти за допомогою трипозиційного перемикача «RUN/STOP» або дистанційно - за допомогою програмуючого пристрою і відповідного програмного забезпечення.

Модулі RMX забезпечують канал для передачі даних між двома резервованими ЦП. В цілому канал зв'язку складається з одного модуля RMX провідного пристрою, одного модуля RMX у підпорядкованому пристрої і двох високошвидкісних оптоволоконних кабелів, що з'єднують їх між собою. Максимальна довжина кабелю у складі каналу зв'язку для резервування, складає 30м. Модуль RMX має перемикач каналів і вісім світлодіодів, які відображають стан модуля.

Система управління автоматично перемикається на резервний ПЛК у разі виходу з роботи активного ПЛК. Перемикнути управління можна вручну, використовуючи перемикач на модулі RMX або програмно. При ручному перемиканні активний ПЛК стає резервним, а резервний ПЛК стає активним.

Система гарячого резервування центрального процесора підтримує два типи мереж Genius - мережі з одною шиною і мережі з дубльованою шиною.

При використуванні мережі Genius з однією шиною в системі гарячого резервування центрального процесора, один контролер шини Genius повинен бути розташований у провідному пристрої і один у підпорядкованому. Усі резервовані контролери шини Genius у провідному ПЛК мають адресу 31, а у підпорядкованому - 30.

Коли в системі з гарячим резервуванням центрального процесора використовуються мережі з дубльованою шиною Genius, то для кожної пари шин необхідно встановити два контролера шини Genius з адресою 31 у провідному пристрої, і два з адресою 30 - у підпорядкованому.

При цьому до першого модуля вводу-виводу Genius, що приєднуються до дубльованої шини необхідно підключити модуль перемикання шини (BSM)[56,57].

## **11.2 Програмування контролерів GE Fanuc**

Для програмування всіх контролерів виробництва GE Fanuc застосовується універсальний засіб Proficy Machine Edition (ME). Цей пакет є

одним з компонентів загальної програмної технології Proficy, призначеної для комплексної автоматизації підприємства від рівня контролерів до MES.

Proficy ME складається з чотирьох частин, серед яких Proficy Logic Developer-PLC – компонент, призначений для програмування і конфігурування всіх ПЛК, контролерів PACSystems і віддалених модулів вводу-виводу. При цьому користувачеві доступна зручна русифікована контекстно-залежна система допомоги, яка не просто містить описи процедур, функціональних блоків, використовуваних типів змінних і тому подібне, але і безліч прикладів їх використання.

У системі передбачені зручні режими наладки проектів в реальному часі шляхом завантаження додатку користувача в цільовий контролер в режимі Run, тестування ПО в режимі Online і можливість посліпної заміни в логіці (яка не міняє розмір програми) без переривання процесу керування. Передбачена перевірка синтаксису логіки і конфігурації апаратної частини проекту перед його компіляцією і завантаженням в контролер. Система блокує завантаження проектів, що містять критичні помилки. Окрім цього доступні засоби контролю даних, у тому числі перегляд змінних (Data Watches), посилань (Reference View) і адрес, що дозволяє зручно контролювати виконання додатку в контролері.

ME забезпечує ефективне програмування задач контролю і управління інженерними мовами стандарту MEK 61131-3 (IL, LD, FBD, SFC, ST), а також мовою «С». Нажаль пакет програмування не має симулятора для перевірки програми без наявності контролера.

### 11.2.1 Методика створення проекту у системі програмування Proficy Machine Edition

Розробити LD-мовою програму керування роботою насоса, який постачає господарську воду споживачам, рис.11.9. Дощова вода збирається у резервуар і

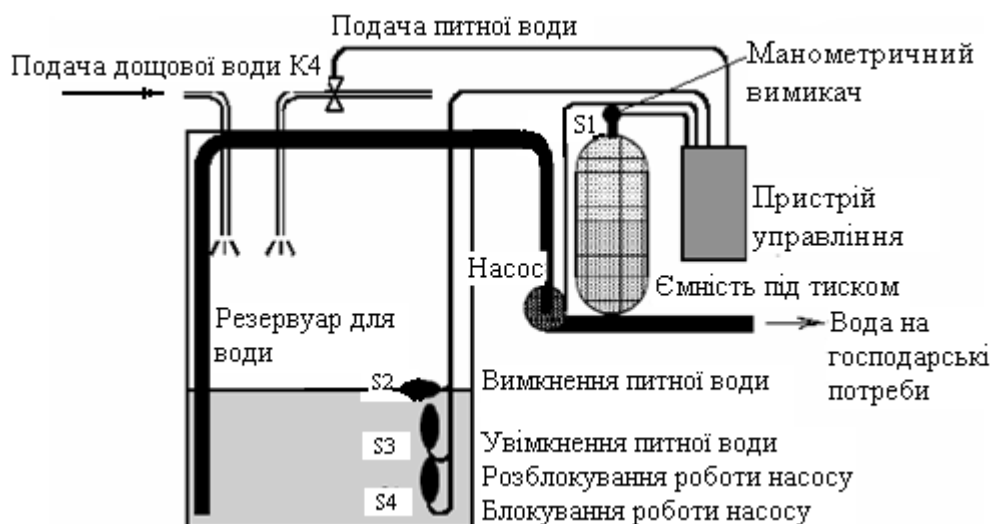


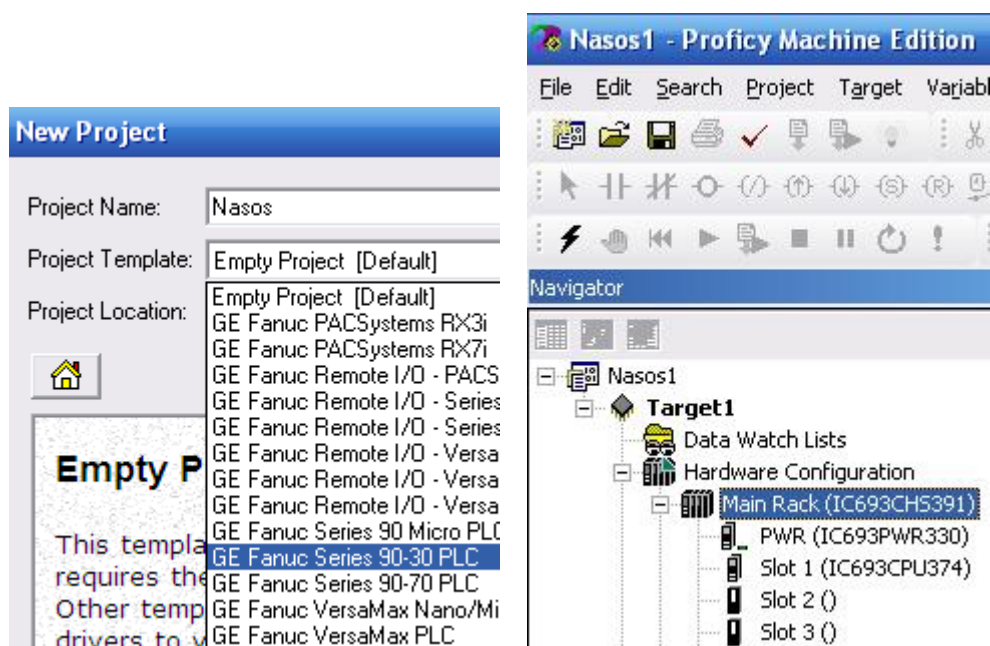
Рисунок 11.9 – Система постачання води на господарські потреби закачується насосом у спеціальну ємність, звідки вона відбирається споживачами. Щоб резервуар не спустошувався, в нього доливається питна вода. Насос і електромагнітний клапан питної води управляються за допомогою

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 вмикається із затримкою на 20с, що зменшує частоту його вмикань при тривалому відборі води. Насос господарської води не може вмикатися, коли в резервуарі недостатньо води.

### ***Розробка проекту***

Для створення проекту запустити програму через меню Пуск->\_Все програми->GE Fanuc-> Proficy Machine Edition V5.9-> 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 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:

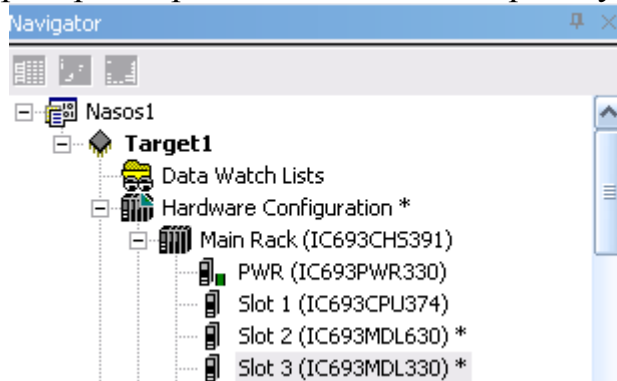


За замовчуванням головний крейт пропонується з блоком живлення і центральним процесором.

Для конфігурування модуля вводу двічі клацнути лівою клавішею 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 оголошені змінних. Після закриття вікна вибраний тип модуля заповнює слот 2.



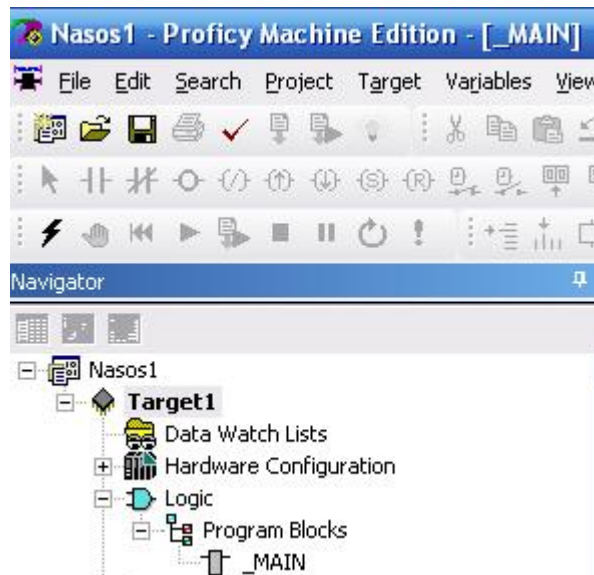
Для конфігурування модуля виводу у слоті 3 аналогічно з каталогу вибрати дискретний модуль на 240В змінного струму (IC693MDL330). Після конфігурування контролера апаратне забезпечення проекту має вигляд:




Для визначення властивостей сконфігурованої 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.

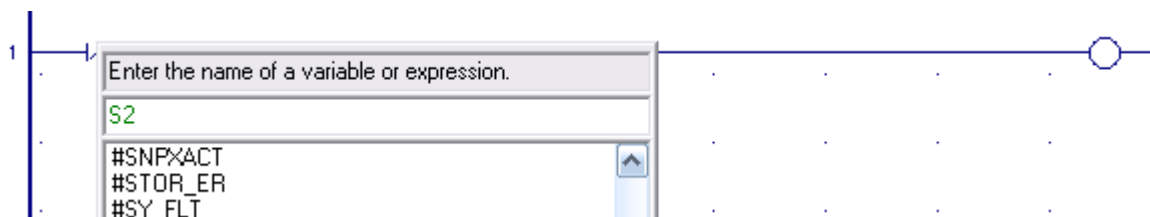
### ***Створення програмного коду***


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

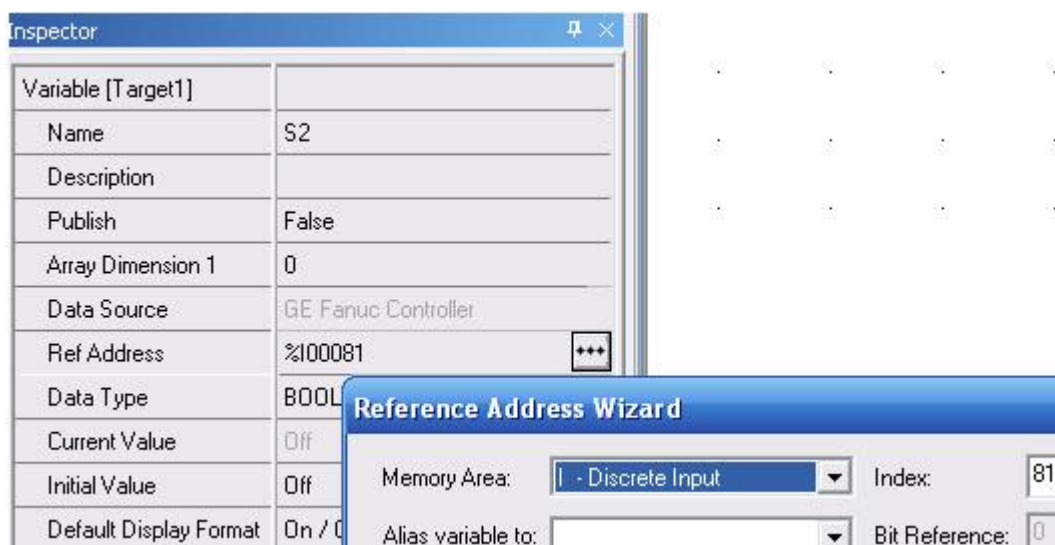


Щоб створити першу LD-мережу програмного коду, яка, відповідно до алгоритму задачі, забезпечить подавання питної води, треба на робочому полі приєднати до шин живлення послідовно два інвертованих контакти і пряму котушку. Для цього в панелі інструментів вибрати по черзі лівою клав'яшею миші відповідні 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:



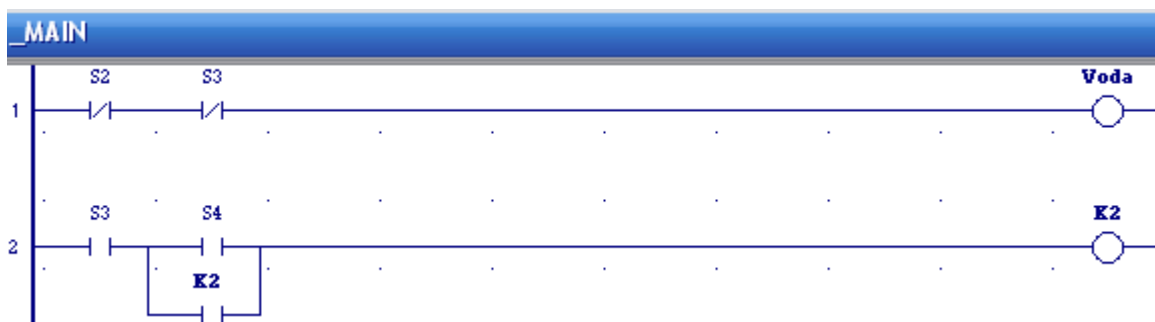
Натиснути клавішу Enter на клавіатурі, щоб ім'я змінної записалося над контактом і було занесене у список на вкладці ... навігатора. Клацнути правою клавішею миші по контакту S2, з'явиться меню, в якому увійти у Properties. Лівою клавішею миші відкрити рядок Ref Address у інспекторі (Inspector), що з'явиться, 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.

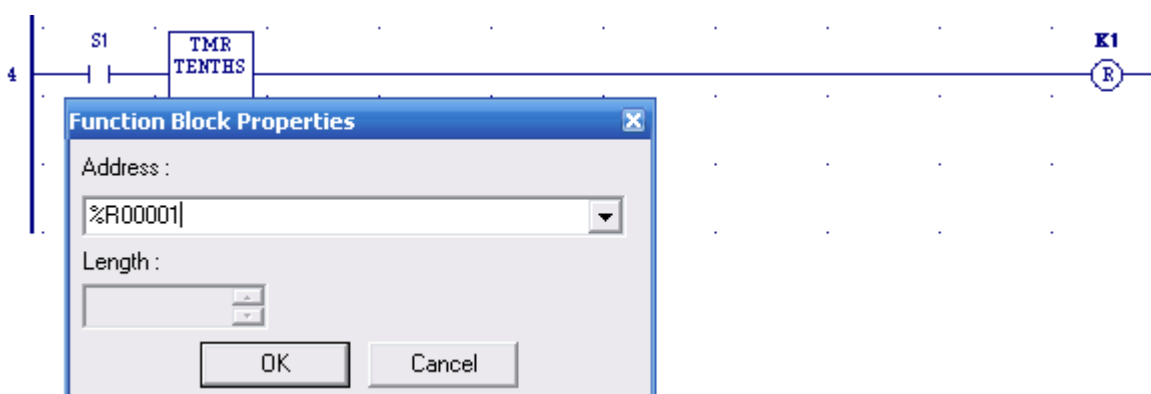
Для створення другої LD-мережі, яка забезпечить блокування роботи насосу при критично мінімальному рівні води у резервуарі, клацнути лівою клавішею миші на робочому полі нижче першої мережі. Панель інструментів активізується. Вибрати по черзі лівою клавішею миші елементи ,  і повторним клацанням поставити 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:



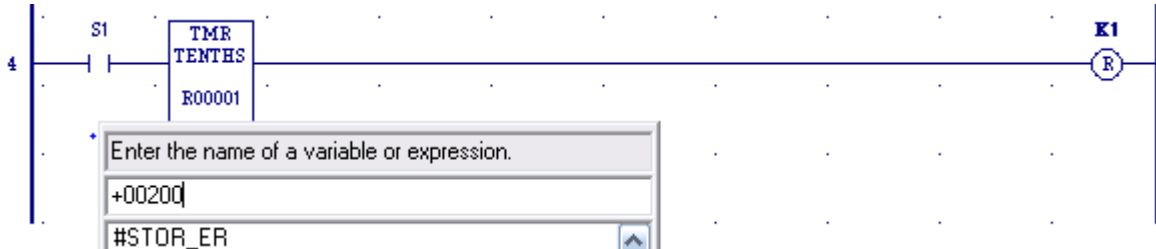
Для створення мережі, яка забезпечить вмикання насосу і затримку при його вимиканні, необхідно активізувати панель інструментів, клацнувши лівою клавішею миші на вільному місці робочого поля. Вибрати по черзі лівою клавішею миші елементи  $\text{H}$ ,  $\text{O}$ ,  $\text{S}$ ,  $\text{R}$ ,  $\text{T}$  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:

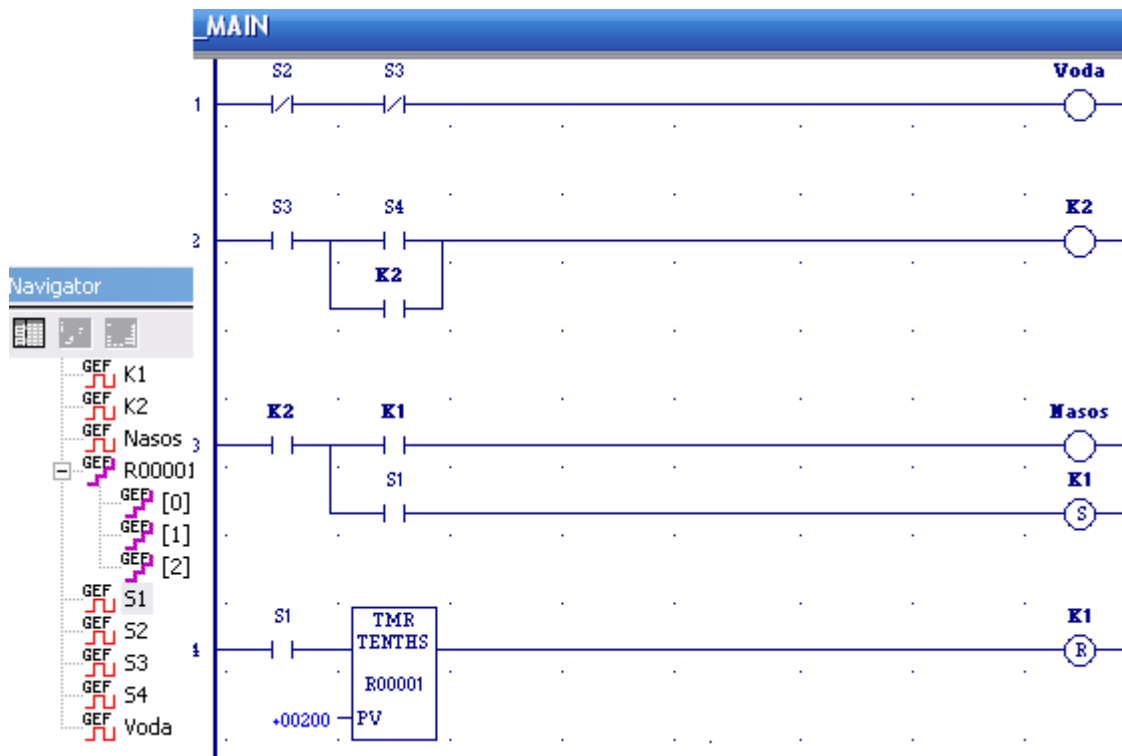


Для оголошення параметрів таймера у списку змінних, що у навігаторі, відкрити вкладку таймера R00001, з'являться три слова (регістри) пам'яті %R для зберігання поточного (0) і заданого значення (1) параметрів таймера, а також керуючого слова. Натиснути ліву клавішу миші на першому параметрі і в інспекторі (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 клавішею 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 клавішею миші по цілі Target1, що у навігаторі проекту, і з меню, що розкриється, вибрати Validate.

При наявності помилок, про що свідчить відповідне повідомлення, здійснити їх виправлення, а при відсутності можна з'єднати проект з контролером, використовуючи меню Target1->Go-Online, у навігаторі проекту. На екрані з'явиться вікно Connecting і пунктирна голуба стрічка. Коли стрічка заповнює вікно повністю з'являється попередження про успішне з'єднання з контролером:



Завантаження програми здійснюється через меню Target1-> Download to Controller... після відповідних позначок у вікні Selections, що відкриється.

Для запуску ПЛК потрібно вибрати Start PLC у меню Target1->Offline Commands і клацнути лівою клавiшею миші[58].

## Тестові та контрольні запитання до розділу 11

**І-й рівень запитань та тверджень дозволяє перевірити загальний рівень засвоєння матеріалу розділу і передбачає відповідь на запитання або твердження “Так” чи “Ні”.**

1. Чи відповідають конструктиви контролерів PACSystems RX7i стандарту VME.

2. Чи можливе розширення системи вводу-виводу PACSystems RX7i за допомогою стійки інтегратора VME ємністю 17 слотів?

3. Стійка RX7i являє собою об'єднувальну панель на 10 слотів, яка підтримує модулі живлення, центральні процесори та модулі вводу-виводу.

4. В основний крейт системи PACSystems RX7i можуть встановлюватися центральні процесори, модулі RX7i та модулі VME одинарної ширини, а також стандартні модулі Series 90-70.

5. Чи використовуються в контролерах PACSystems RX7i такі процесори, як AMD Duron та AMD K6, які мають великий об'єм пам'яті, забезпечують швидке виконання команд і працюють з тактовою частотою, відповідно, 200 і 400 МГц.

6. Чи має модуль центрального процесора PACSystems RX7i на передній панелі три незалежних вбудованих послідовних порти?

7. У складі ПЛК RX7i є два блоки живлення на 110 - 230В змінного струму потужністю 100 Вт, а також на 36В постійного струму з вихідною потужністю 350Вт.

8. Вихідні канали модуля живлення RX7i автоматично вимикаються у разі їх перегріву та вмикаються при охолодженні до заданої температури.

9. Модулі аналогового вводу-виводу системи управління PACSystems RX7i мають конструкцію подвійної ширини.

10. Чи можна кожний канал модулів дискретного вводу контролера PACSystems RX7i конфігурувати на необхідний рівень напруги?

11. Чи відносяться до спеціальних модулів в контролері PACSystems RX7i модуль високошвидкісного лічильника, програмований модуль співпроцесора та однослотовий модуль жорсткого диска для шини VME?

12. Genius і Field Control – це протоколи, що розроблені корпорацією GE Fanuc для обміну даними між контролерами PACSystems RX7i та польовими пристроями.

13. Контролер шини Genius і сканер віддаленого вводу-виводу забезпечують інтегрування ПЛК GE Fanuc в мережу Genius і взаємодію з віддаленою периферією Genius і Field Control.

14. Чи здатні модулі вводу-виводу Genius знімати інформацію з об'єкта, а також додатково надавати діагностичну інформацію про зовнішні підключення, умови електроживлення і навантаження та про стан мереж зв'язку, модулів і електричних кіл?

15. Система вводу-виводу Field Control працює тільки з уніфікованими аналоговими і дискретними сигналами.

16. Чи здатні контролери серії RX7i реалізовувати розподілене управління за допомогою пристроїв зв'язку Ethernet, Profibus-DP, Genius і DeviceNet?

17. Гаряче резервування у ПЛК серії RX7i може реалізовуватись шляхом додавання окремих резервних модулів вводу-виводу.

18. Чи підтримує система гарячого резервування центрального процесора мережу Genius?

19. Чи для всіх контролерів виробництва GE Fanuc можна застосовувати універсальний засіб програмування Proficy Machine Edition?

20. Чи правильною надана послідовність дій, якої дотримуються при створенні проекту у системі програмування Proficy Machine Edition: запуск програми Logic Developer PLC, вибір шаблону проекту GE Fanuc, визначення властивості сконфігурованої цілі, конфігурування модулів вводу-виводу, з'єднання проекту з контролером, оголошення змінних, створення програмного коду, перевірка синтаксису програмного коду, завантаження програми у ПЛК, запуск ПЛК у роботу.

*Перевірте відповіді на сторінках: 1 – 297...298 с., 2 - 298 с., 3 - 298 с., 4 - 299 с., 5 - 301 с., 6 - 301 с., 7 - 302 с., 8 - 302 с., 9 - 304 с., 10 – 304...305 с., 11 - 305 с., 12 - 306 с., 13 - 306 с., 14 - 308 с., 15 - 308 с., 16 - 309 с., 17 – 309...310 с., 18 - 310 с., 19 - 311 с., 20 – 311...317 с.*

**II-й рівень запитань (тестові запитання) дозволяє перевірити рівень засвоєння матеріалу розділу із заглибленням у технічні особливості будови та структури контролерів PACSystems RX7i та передбачає вибір правильної відповіді з наведених трьох варіантів.**

**1. Що є базовою основою системи управління PACSystems RX7i?**

а) базовий модуль контролера RX7i з 10 слотами;

- б) об'єднувальна панель, яка має 16 слотів;
- в) стійка на 9, 17 або 19 слотів.

**2. Як використовуються слоти стійки RX7i?**

- а) у 0 та 1 слот встановлюється модуль центрального процесора з дочірньою платою Ethernet, у слот 2–блок живлення, у слоти 3 -17 інші модулі;
- б) у стійці в 1 та 2 слоти розміщується модуль центрального процесора з дочірньою платою Ethernet, а інші модулі розміщуються у довільному порядку;
- в) у 0 слот встановлюється блок живлення, у слоти 1 і 2– модуль центрального процесора з дочірньою платою Ethernet, у слоти 3 -17 інші модулі.

**3. Як називається стійка з встановленими блоком живлення і центральним процесором з платою Ethernet?**

- а) крейт;
- б) панель центрального процесора;
- в) об'єднувальна панель.

**4. Як називаються розніми модулів, які підключаються до об'єднувальних панелей?**

- а) P1 – рознім модуля, який підключається до верхньої об'єднувальної панелі, P2 – рознім, який підключається до нижньої об'єднувальної панелі;
- б) J1 – рознім модуля, який підключається до верхньої об'єднувальної панелі, J2 –рознім, який підключається до нижньої об'єднувальної панелі;
- в) RJ-45-1 – рознім модуля, який підключається до верхньої об'єднувальної панелі, RJ-45-2 –рознім, який підключається до нижньої об'єднувальної панелі.

**5. Де у крейті встановлюються модулі VME інших виробників?**

- а) всі модулі VME інших виробників повинні бути встановлені ліворуч від модулів GE Fanuc;
- б) всі модулі VME інших виробників повинні бути встановлені праворуч від модулів GE Fanuc;
- в) всі модулі VME інших виробників повинні бути встановлені в кінці крейта.

**6. З якою швидкістю центральний процесор обмінюється даними з модулями вводу-виводу і інтелектуальними модулями в контролерах PACSystems RX7i?**

- а) до 30 Мб за секунду;
- б) до 14 Мб за секунду;
- в) до 10 Мб за секунду.

**7. Що знаходиться під відкидною кришкою на лицевій поверхні модуля центрального процесора PACSystems RX7i?**

- а) порт для підключення контролера до мережі Ethernet;
- б) слот для розташування плати розширення пам'яті;
- в) слот для розташування батарейки, яка забезпечує енергонезалежну пам'ять ОЗП.

**8. При яких умовах використовують блоки живлення IC698PSA350 і C698PSD300, що мають потужність більше 300 Вт?**

а) при збільшеній кількості модулів в умовах наявності примусового повітряного їх охолодження;

б) при необхідності додатково забезпечити живлення датчиків і виконавчих пристроїв в умовах відсутності примусового повітряного їх охолодження;

в) при необхідності додатково забезпечити живлення датчиків і виконавчих пристроїв в умовах наявності примусового повітряного їх охолодження.

**9. Що знаходиться у верхній частині передньої панелі блока живлення RX7i?**

а) клеми L1 і L2/N для приєднання проводів напруги живлення 120/240В змінного струму та клемма заземлення GND;

б) світлодіодні індикатори для відображення стану та режиму роботи блока;

в) двопозиційний перемикач, який перемикає логічні рівні вихідної напруги за рахунок активізації або блокування вихідних каналів.

**10. Які модулі живлення призначені для стійок розширення RX7i?**

а) спеціальні модулі живлення (IC698PSA350 і C698PSD300), які встановлюються в крейт ПЛК, як звичайні модулі вводу-виводу;

б) спеціальні модулі живлення (IC698PWR100 і C698PWR300), які встановлюються в крейт ПЛК, як звичайні модулі вводу-виводу;

в) спеціальні модулі живлення (IC697PWR710, IC697PWR711, IC697PWR724, IC697PWR748), які встановлюються в крейт ПЛК, як звичайні модулі вводу-виводу.

**11. Як поміж собою поділяються модулі дискретного виводу PACSystems RX7i?**

а) транзисторні модулі з вихідною напругою змінного струму, модулі з вихідною напругою постійного струму і модулі з релейними виходами;

б) модулі сполучення з безконтактними датчиками, кнопками, перемикачами і маніпуляторами двійково-десятькового коду;

в) 4-, 16-, 32- каналні модулі дискретного виводу.

**12. На яку кількість каналів випускаються аналогові модулі вводу-виводу PACSystems RX7i?**

а) 2-, 4-, 16- та 32- каналні;

б) 8-, 16-, 32- та 64- каналні;

в) 2-, 4-, 12-, 16- та 32- каналні.

**13. Як здійснюється адресація модулів вводу-виводу PACSystems RX7i?**

а) визначається положенням кругового перемикача, який розташований на модулі вводу-виводу;

б) визначається номером слота в стійці, де вони встановлені;

в) встановлюється за допомогою DIP-перемикачів, які розташовані на модулі вводу-виводу.



**14. Які пристрої розроблені в системі управління PACSystems RX7i для реалізації віддаленого вводу-виводу?**

а) контролери шини Genius, Genius-сканер, передаючий та приймальний модулі розширення;

б) модулі інтерфейсного обміну даними Genius, які встановлюються на стійці системи управління PACSystems RX7i та на стійці віддаленого вводу-виводу;

в) модулі інтерфейсного обміну даними Genius, які встановлюються на стійці системи управління PACSystems RX7i та на стійці розширення.

**15. Яку топологію має мережа Genius?**

а) шина;

б) зірка;

в) кільце.

**16. Як здійснюється інтегрування пристрою Field Control в мережу Genius?**

а) пристрої Field Control не інтегруються в мережу Genius;

б) інтегрування пристрою Field Control в мережу Genius забезпечується за допомогою Genius-сканера;

в) інтегрування пристрою Field Control в мережу Genius забезпечує модуль сполучення ВІУ.

**17. Що собою являє модуль інтерфейсу Ethernet?**

а) модуль має тільки два дуплексні Ethernet порти;

б) модуль має тільки один дуплексний Ethernet порт з вбудованим мережним комутатором;

в) модуль має два дуплексні Ethernet порти з вбудованим мережним комутатором і один послідовний порт RS 232.

**18. Як у системі управління PACSystems RX7i можна здійснювати перемикання на резервний ПЛК у разі виходу з роботи активного ПЛК?**

а) автоматично, вручну, використовуючи перемикач на модулі RMX або програмно;

б) автоматично за допомогою модуля RMX або програмно;

в) автоматично або вручну, використовуючи перемикач на модулі RMX.

**19. Як забезпечується система гарячого резервування при використуванні мережі з дубльованою шиною Genius?**

а) завдяки розташуванню одного контролера шини Genius у провідному пристрої і одного у підпорядкованому;

б) завдяки розташуванню двох контролерів шини Genius у провідному пристрої і двох у підпорядкованому;

в) завдяки розташуванню двох контролерів шини Genius у провідному пристрої і одного у підпорядкованому.

**20. Чим обмежується кількість модулів у стійці?**

а) кількістю слотів у стійці;

б) потужністю блока живлення;

в) конструктивними параметрами стійки.

*Перевірте відповіді на сторінках: 1 - 298 с., 2 - 299 с., 3 - 299 с., 4 - 298, 300 с., 5 - 300 с., 6 - 301 с., 7 - 302 с., 8 - 302 с., 9 - 303 с., 10 - 304 с., 11 – 304...305 с., 12 - 305 с., 13 - 305 с., 14 - 306 с., 15 - 307 с., 16 - 308 с., 17 - 309 с., 18 - 310 с., 19 - 310 с., 20 - 300 с.*

**III-й рівень запитань (контрольні запитання) дозволяє перевірити рівень засвоєння матеріалу із систематизацією отриманої у розділі інформації та передбачає відповіді, що побудовані на логічному аналізі або порівнянні з приведенням конкретних технічних параметрів (даних або прикладів).**

1. Поясніть взаємодію всіх типів модулів, що входять до складу системи управління PACSystems RX7i.
2. Розкрийте поняття стійкі системи управління PACSystems RX7i.
3. Роз'ясніть принцип роботи із засобами віддаленого вводу-виводу в системі управління PACSystems RX7i.
4. Види та технічні характеристики модулів центрального процесора контролера RX7i.
5. Які типи модулів живлення використовуються в системі управління PACSystems RX7i та в чому їх особливість?
6. Які засоби закладено в систему управління PACSystems RX7i для підвищення надійності її роботи у складі АСУ ТП?
7. Поясніть поняття Genius та Field Control.
8. В чому складається принцип обміну даними між контролерами за допомогою Global Data і Datagram.
9. Виконайте порівняння між модифікаціями модулів дискретного вводу-виводу контролерів серії SYSMAC CS1 та PACSystems RX7i.
10. Засоби програмування контролерів GE Fanuc.

*Перевірте відповіді на сторінках: 1 – 296...310 с., 2 – 298...300 с., 3 – 306...309 с., 4 – 301...302 с., 5 – 302...304 с., 6 – 296...310 с., 7 – 306...309 с., 8 - 308 с., 9 – 267...269, 304...305 с., 10 – 311...317 с.*

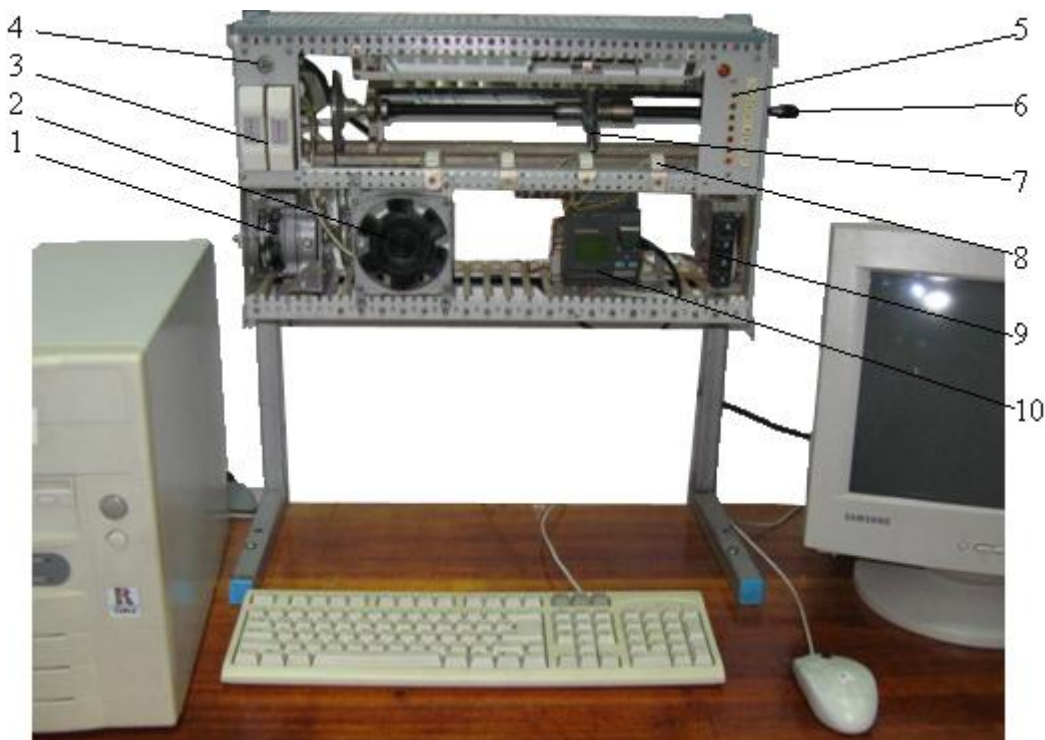
## 12 ЛАБОРАТОРНІ РОБОТИ

### 12.1 Лабораторна робота №1

Програмування та дослідження роботи контролера Logo!Basic.

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

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



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

Рисунок 12.1 – Лабораторна установка

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

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

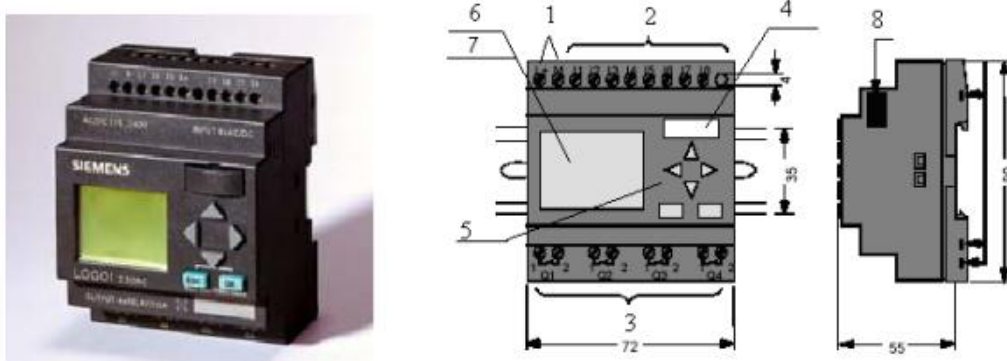
При досягненні кареткою крайнього положення спрацьовують кінцеві вимикачі, які зупиняють двигун. У крайньому лівому положенні каретки вмикається вентилятор 2 [60].

### **Контролер LOGO! Basic**

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

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

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



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

Рисунок 12.2 – Контролер LOGO! Basic

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

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

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

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

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

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

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

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

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

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

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

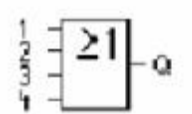
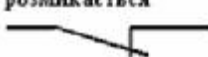
Входи I7 і I8 можуть бути також запрограмовані для використання як аналогові входи. Тоді вони позначаються як AI1 і AI2.

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

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

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

Таблиця 12.1 Основні функції – GF

| Зображення на комутаційній схемі  | Зображення в LOGO!   | Найменування               |
|---|--|----------------------------|
| Послідовне з'єднання замкнених контактів. Якщо який-небудь вхід цього блоку не підключений (х), то для цього входу $x=1$ .   |  | AND (І)                    |
| Паралел. з'єднання замкнених 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** включають у свій склад таймери, лічильники і інші функціональні блоки. Вони можуть зберігати і мають можливості параметризації (Rem). Опис деяких спеціальних функцій наведено у табл.12.2.

Таблиця 12.2

## Спеціальні функції – SF

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

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

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

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

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

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

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

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

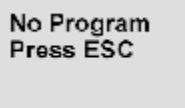
Розробити програму руху каретки за певними позиціями, які визначаються датчиками положення 8 (з першої на третю, з третьої на другу, з другої на четверту і знову на першу). При цьому на кожній позиції каретка має витримувати п'ятисекундну паузу .

Для програмування використовуються:

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

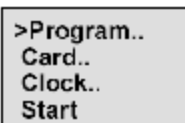
### *Програмування з лицевої панелі*

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



No Program  
Press ESC

якщо ні, тоді:

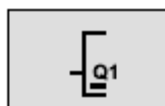


>Program..  
Card..  
Clock..  
Start

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 ОК. LOGO! перейде в режим програмування >Edit.

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



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

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

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

За замовчуванням блок отримає номер В1.



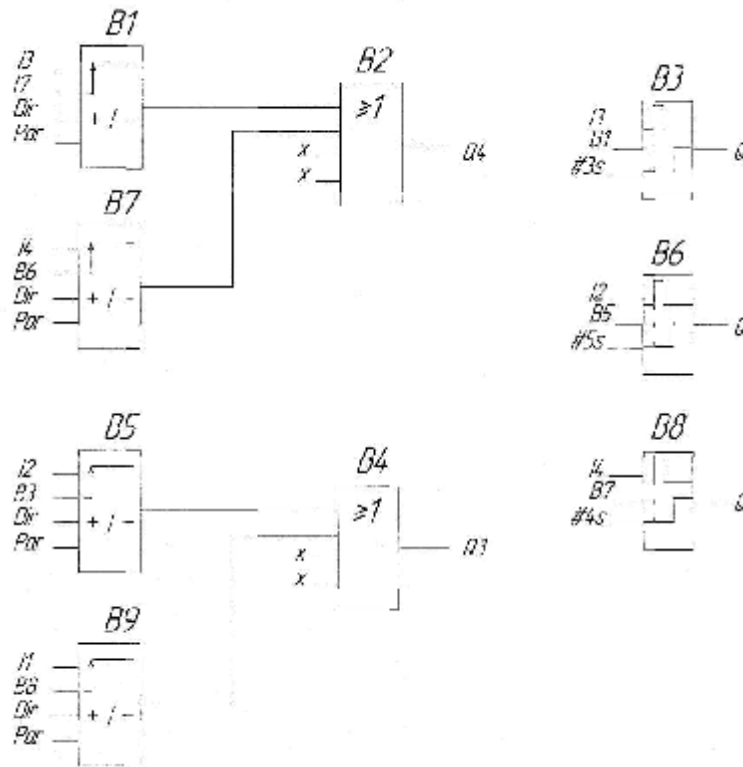
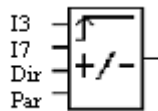


Рисунок 12.3 – FBD-програма керування роботою каретки

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



Третій вхід залишається без зміни.

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

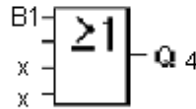
Для подачі на вихід контролера Q4 керувального сигналу необхідно створити інший блок – логічне АБО. Для цього:

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

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

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

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



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

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

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

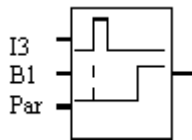
LOGO! відобразить перший блок у списку спеціальних функцій;

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

Для підключення входів блока таймера В3:

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

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



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

Для цього:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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


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

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

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

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



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

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

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

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


2. Для встановлення блока АБО:

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

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

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

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

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

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

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

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



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

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

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

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


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

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

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

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

Для цього:

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

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

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

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

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

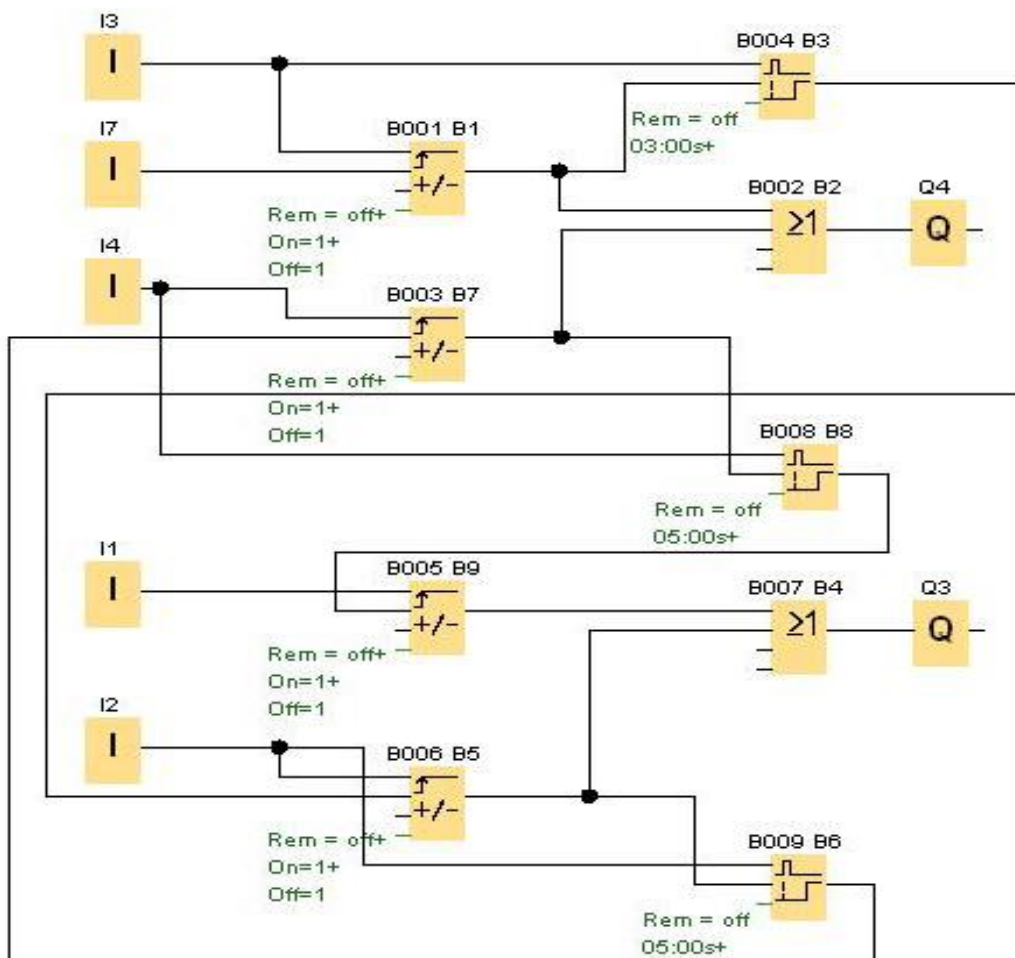



Рисунок 12.4 – Програма керування рухом каретки

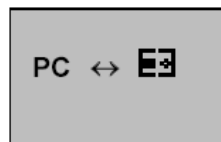
Для перевірки роботи програми в панелі інструментів натиснути кнопку «Емуляція» . Активізуючи входи контролера у програмі, за допомогою їх символів, що унизу робочого поля, і лівої клавіші миші, переконатися в правильності виконання завдання. При цьому сполучні лінії між блоками відображаються різними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

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

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

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

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



LOGO! = LOGO!

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

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

### ***Зміст звіту***

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

### ***Контрольні запитання***

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

## **12.2 Лабораторна робота №2**

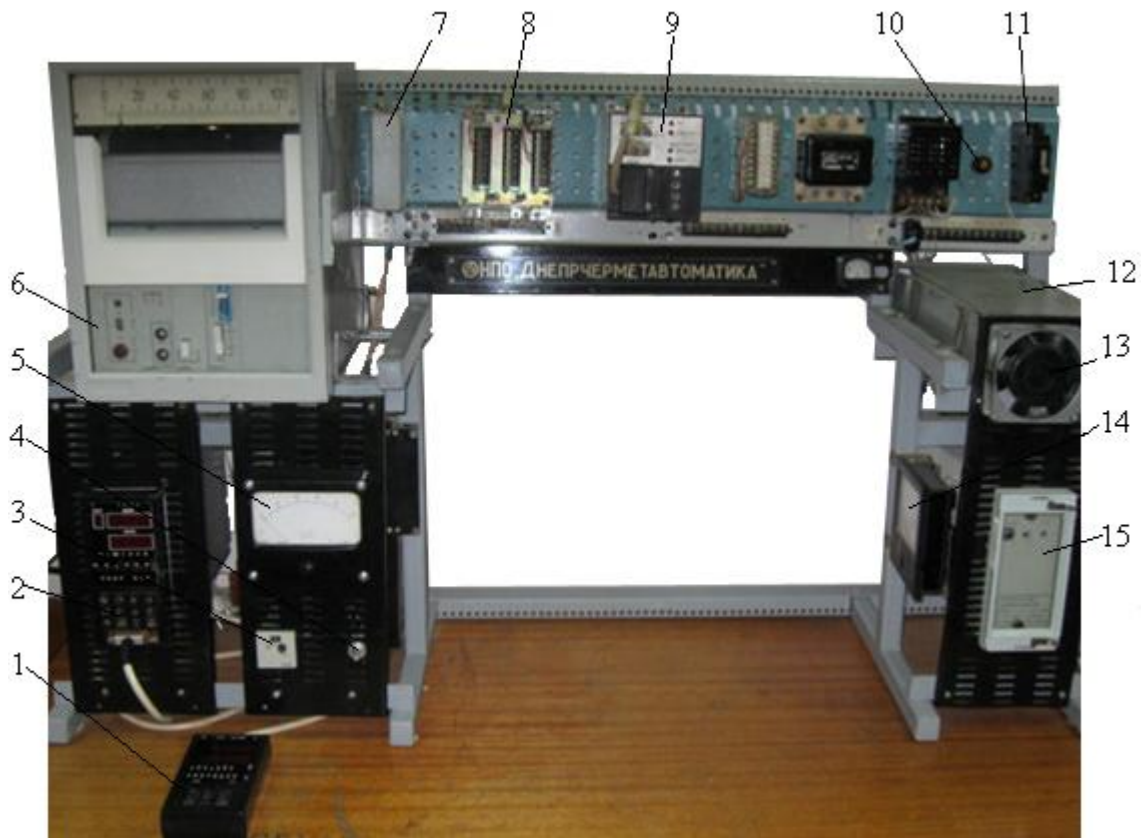
Програмування та дослідження роботи контролера Реміконт Р-130

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

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

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

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



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

Рисунок 12.5 – Лабораторна установка

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

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

### **Реміонт Р-130**

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



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

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

Основна частина блока БК-01 містить:

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

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

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

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

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

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

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

### Програмування контролера

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

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

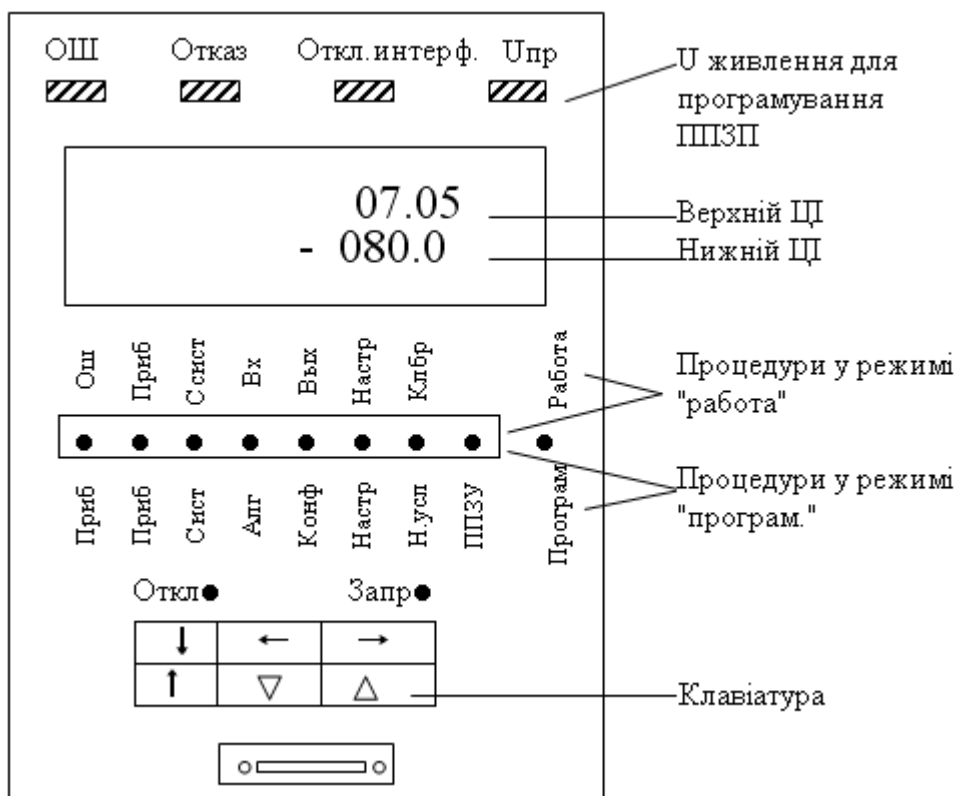


Рисунок 12.6 – Пульт настройки ПН-1

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

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

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

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

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

У режимі програмування задаються усі програмовані параметри контролера, що визначають його алгоритмічну структуру. Ці параметри у загальному випадку задаються “триступеневим” методом: спочатку вибирається та або інша процедура, в неї вибирається потрібна операція і у межах цієї операції 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 ПЗП (“ППЗУ”) - виконується запис у ПЗП, відновлення інформації із ПЗП в ОЗП, регенерація ПЗП, ПЗП.

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

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

При 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.12.3.

Таблиця 12.3 Приладові параметри

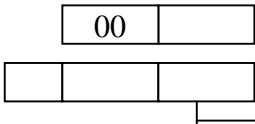
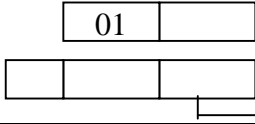
| Операція  | Формат індикації | Параметри  |
|---|------------------|--|
| 1. Обнуління або введення стандартної конфігурації          |                  | <b>N1=00</b> - заборона обнуління або введення стандартної конфігурації<br><b>N1=01</b> - дозвіл<br><b>N2=00</b> - обнуління<br><b>N2≠00</b> - номер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

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

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

Таблиця 12.4

Системні параметри

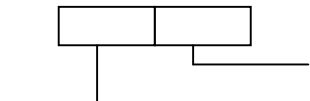
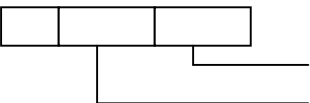
| Операція            | Формат індикації  | Параметри  |
|---------------------|---|--|
| 1. Системний номер  |  | N0-15 – системний номер, (коли N=00 контролер не є у локальній мережі) |
| 2. Режим інтерфейсу |  | N=01 – інформаційний режим<br>N=02 – командний режим                   |

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

Правила установлення алгоритму приведені у табл. 12.5.

Таблиця 12.5

Параметри алгоблоків

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

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

Параметри настройки поділяються на дві групи – константи коефіцієнти.

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

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

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

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

Правила конфігурування наведені у табл. 12.6.

Таблиця 12.6 Параметри конфігурування

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

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

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

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

Правила установлення параметрів настройки наведені у табл.12.7.

Таблиця 12.7 Параметри настройки

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

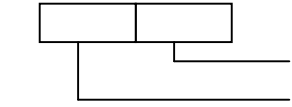
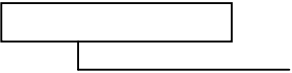
У процедурі “н.усл” (початкові умови) встановлюються значення сигналів на входах алгоблоків, з якими алгоблоки почнуть працювати при переході у режим “робота”.

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

Правила встановлення початкових умов наведені у табл.12.8.

Таблиця 12.8

Початкові умови

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

### **Настройка і контроль**

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

### **Виконання лабораторної роботи**

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

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

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

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

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

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

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

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

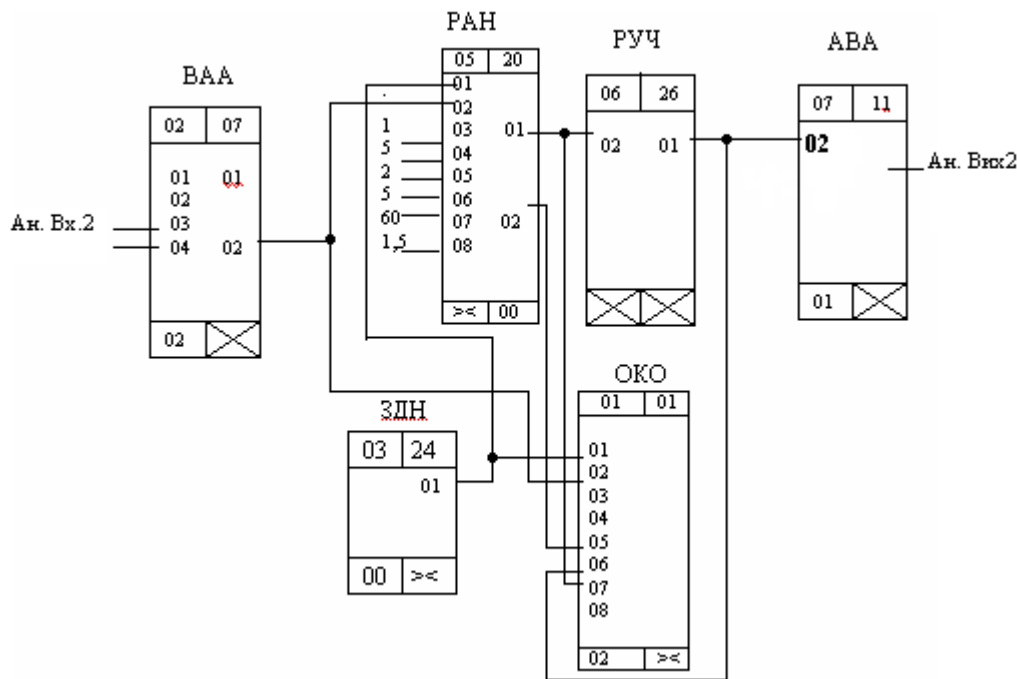


Рисунок 12.7 – Схема конфігурації Реміконта Р-130

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

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

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

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

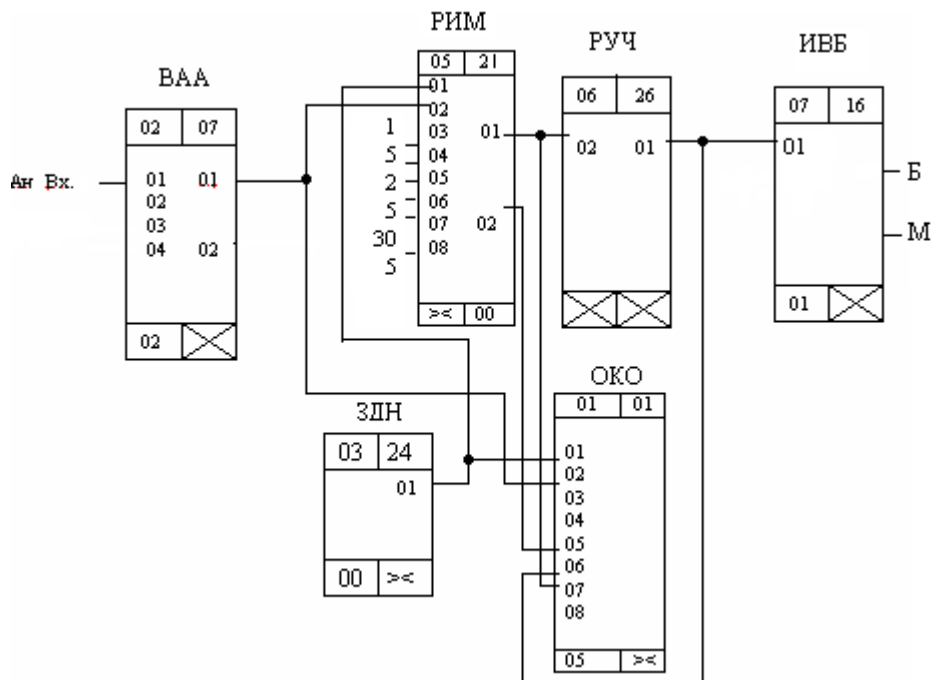
4.9. У режимі “робота” вимкнути пульт настройки ПН-1. За допомогою відповідних клавіш лицевої панелі в цифровому індикаторі «Задание» установити 30% і виконати дослідження роботи системи автоматичного регулювання температури в печі, керуючись цифровим індикатором блока БК-1, і вторинним приладом РП160,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

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

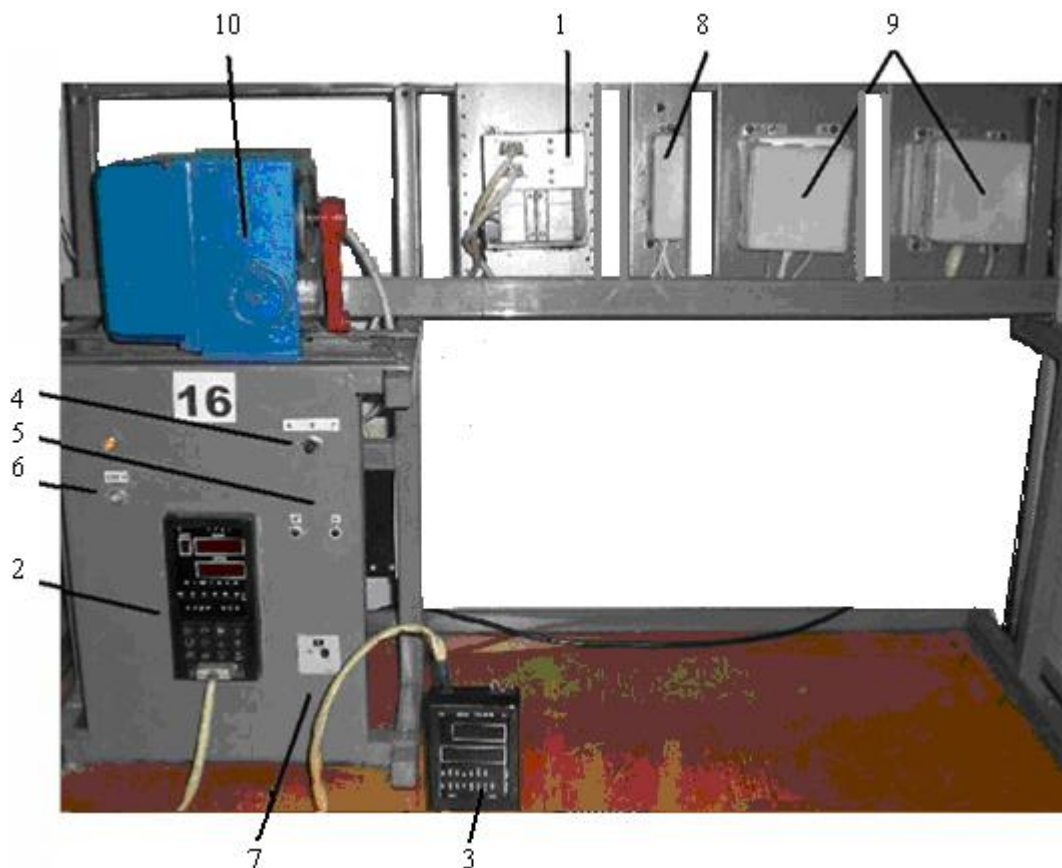
### ***Завдання для самостійної роботи***

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





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



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

Комплектність контролера – 26.

### ***Зміст звіту***

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

### ***Контрольні запитання***

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

### **12.3 Лабораторна робота №3**

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

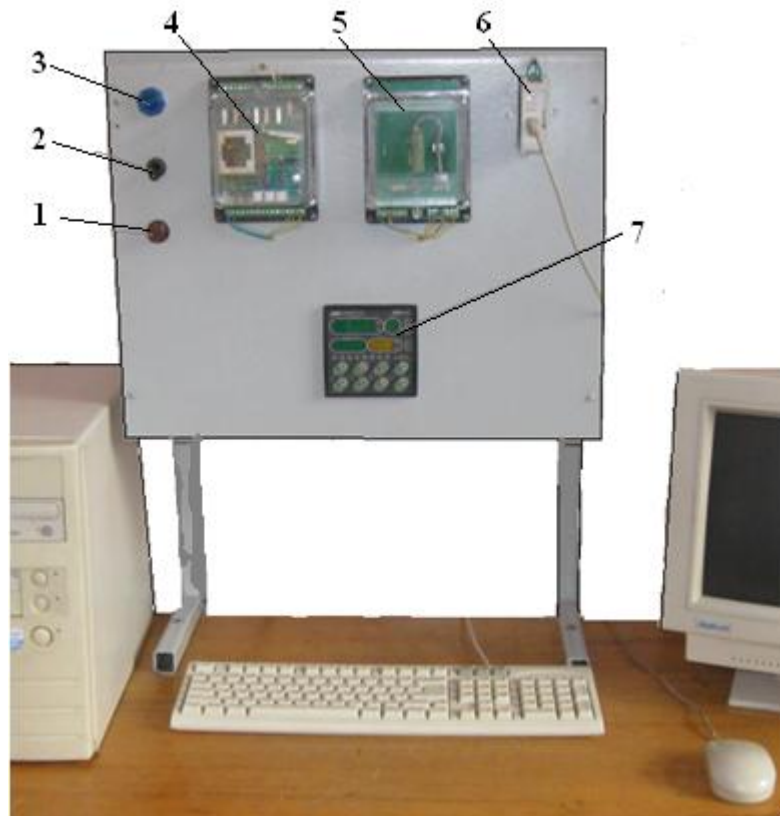
**Мета:** ознайомитися з будовою МІК-51, придбати навички програмування і роботи з контролером.

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

### ***Контролер МІК-51***

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

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



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

Рисунок 12.8 – Лабораторна установка

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

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

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

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

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

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

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

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

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

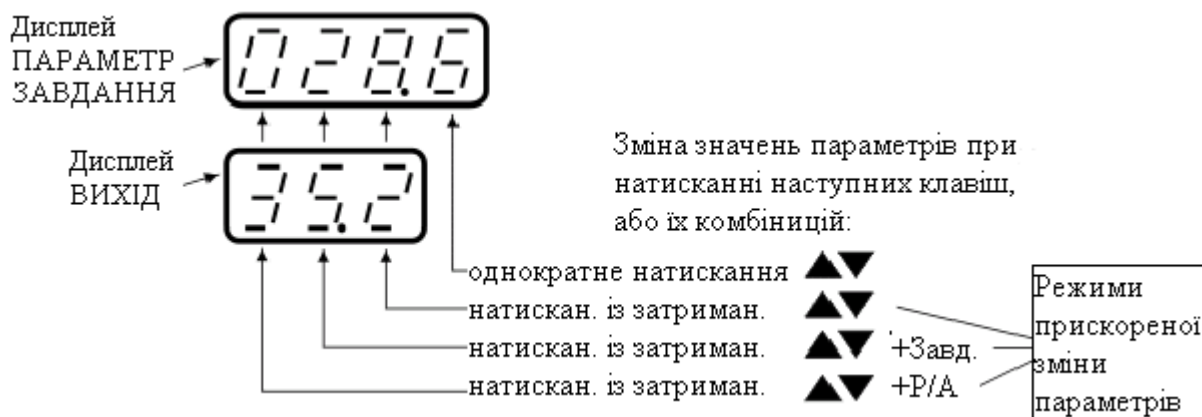


Рисунок 12.9 – Зміна значень на різних дисплеях

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

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

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

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

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

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

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

*Зміна заданої уставки регуляторів*

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

- Внутрішня задана уставка змінюється з лицеві панелі приладу.

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

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

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

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

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

*Зміна значення керуючого впливу*

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

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

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

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

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

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

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

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

Таблиця 12.9

## Програмування функціональних блоків

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

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

### **Створення програмного коду**

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

Таблиця 12.10 Програмування функціонального блока АІН(05)

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

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

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

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

Таблиця 12.11 Програмування зв'язків для блока PID

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

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

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

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

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

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

Приклад програмування параметрів алгоблока АІН наведений у табл.12.12.

Таблиця 12.12 Настроювальні параметри алгоблока АІН

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

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




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

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

Таблиця 12.13 Збереження програми користувача

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

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

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


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

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


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

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

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

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

4. При необхідності відкриття раніше створеного проекту:

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

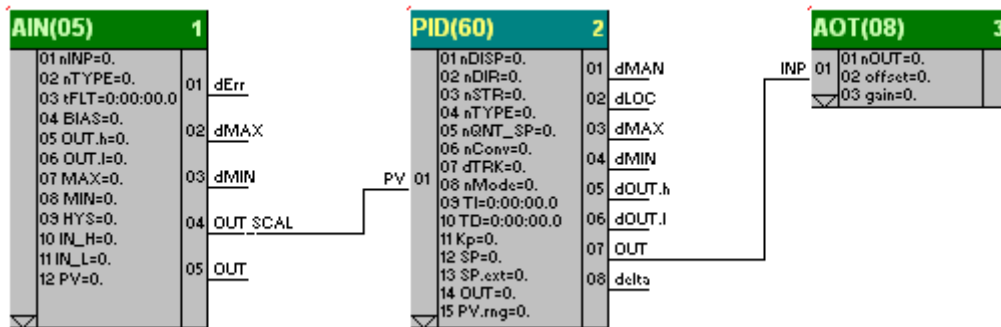
- у полі «Тип файлів» вибрати необхідний формат файлу \*.fdb;
  - щоб відкрити проект, збережений в іншій теці, знайти і відкрити потрібну теку;
  - клацнути по проекту, який потрібно відкрити в ALFA;
  - натиснути кнопку «Открыть».
5. Для збереження проекту:



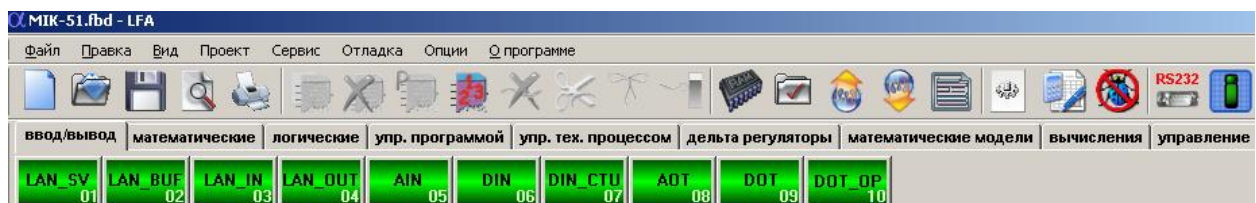
- натиснути кнопку «Зберегти» на панелі інструментів;
- Щоб зберегти файл в іншій теці, знайти і відкрити потрібну теку;
- Ввести ім'я документа в полі Ім'я файлу і натиснути кнопку «Сохранить».

### Створення програмного коду

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



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



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

Сполучення блоків здійснюється за допомогою миші:

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

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

Таблиця 12.14 Розділи бібліотеки функціональних блоків


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

Для настроювання параметрів функціональних блоків:


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


Нумерація блоків виконується в порядку їх установлення.

При необхідності можна змінити нумерацію:


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

Розподіл параметрів блока в регістровій області пам'яті здійснюється:

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

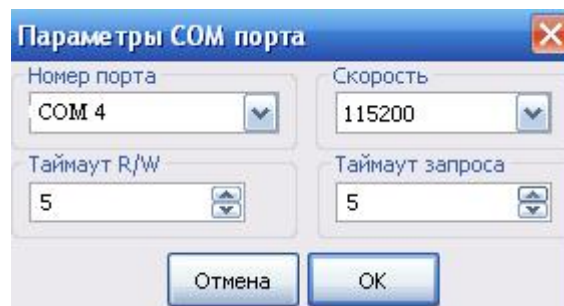
Перевірка програми здійснюється за допомогою кнопки  (Перевірити програму), що на панелі інструментів. При цьому виконується перевірка програми користувача на відповідність існуючим правилам.


### *Налагодження*


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

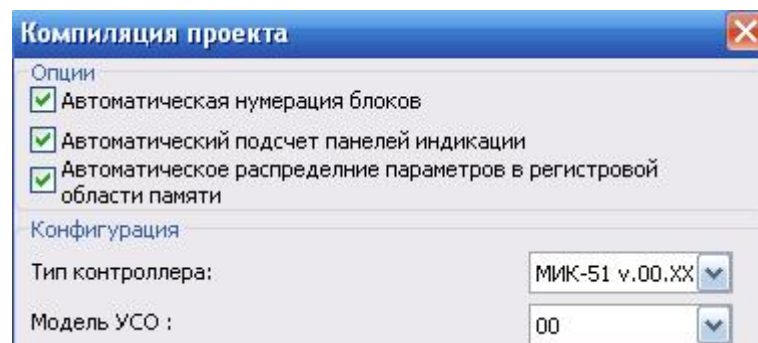
### *Запис і читання програми з контролера*

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

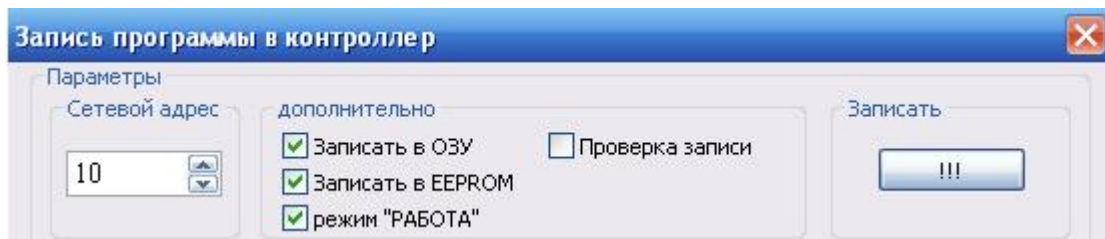


*Запис, читання програми* здійснюється за відповідною командою в меню «Действия» або, користуючись кнопками  на панелі інструментів.

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



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



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

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

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

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



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

### ***Зміст звіту***

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

### **Контрольні питання**

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

### **12.4 Лабораторна робота №4**

Програмування та дослідження роботи контролера VIPA Speed 7 300

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

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



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

Рисунок 12.10 – Лабораторна установка

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

### ***Контролер VIPA Speed 7- 300***

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

### ***Створення проекту***

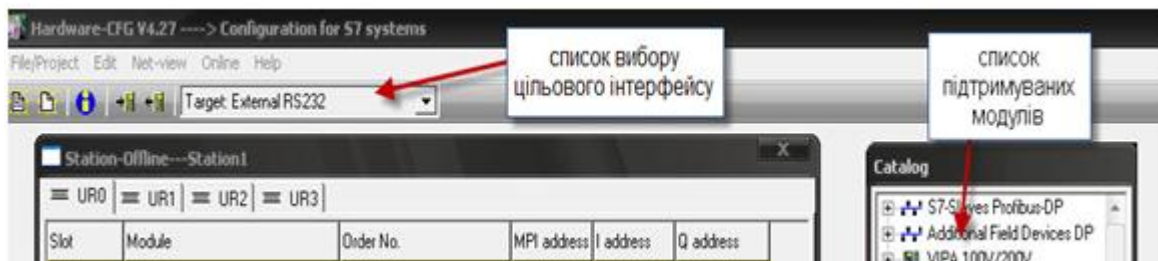
Увімкнути в роботу лабораторну установку та здійснити очищення пам'яті ПЛК. Для цього на блоці живлення контролера PS 307 перевести перемикач у положення On, а на контролері VIPA Speed 7 300 перемикач



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

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

Для конфігурування контролера, в панелі інструментів натиснути ліву 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.

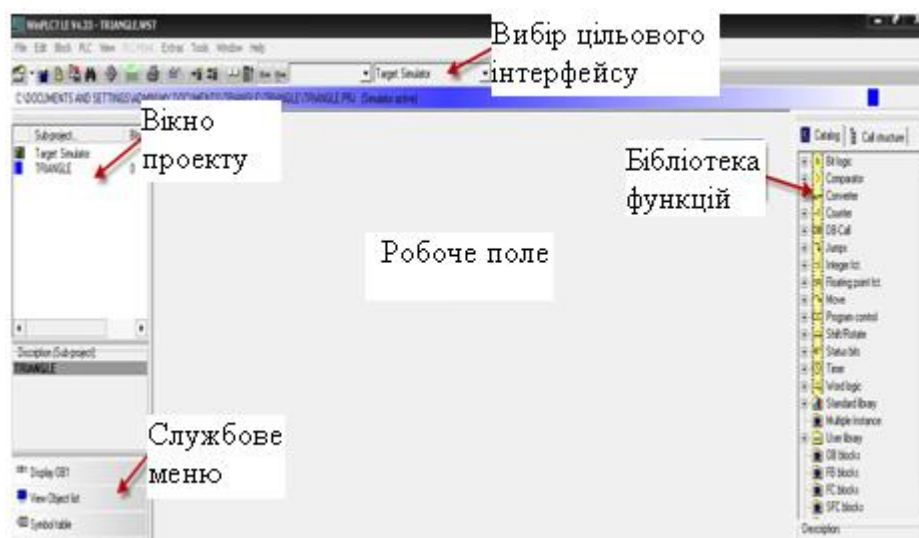
Оскільки контролер на лабораторному стенді підключений до комп'ютера за допомогою Ethernet-інтерфейсу, в панелі інструментів зі списку, що випадає, вибрати цільовий інтерфейс станції – Target: External TCP/IP direct.

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, з нижнього списку вибрати активний мережний адаптер Realtek RTL8029(AS) Ethernet Adapter і

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

### *Розроблення програмного коду*

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



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

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

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


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

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

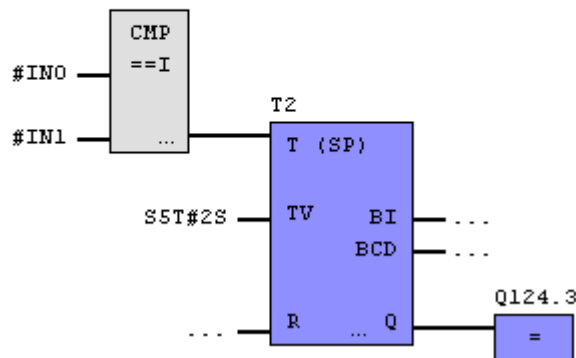
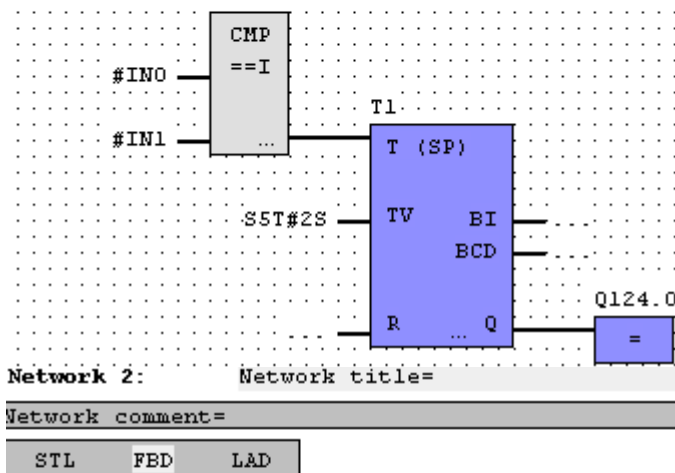
Позначити перший вхід суматора та надрукувати змінну #IN0. На другому вході суматора надрукувати вхідну змінну IN1. Позначити вихід суматора та двічі клацнути по таймеру S\_PULSE, що знаходиться у списку каталогу Timer бібліотеці функцій. Присвоїти йому ім'я «T1», на вході «TV» задати час 2 сек («S5T#2s»),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

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

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

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

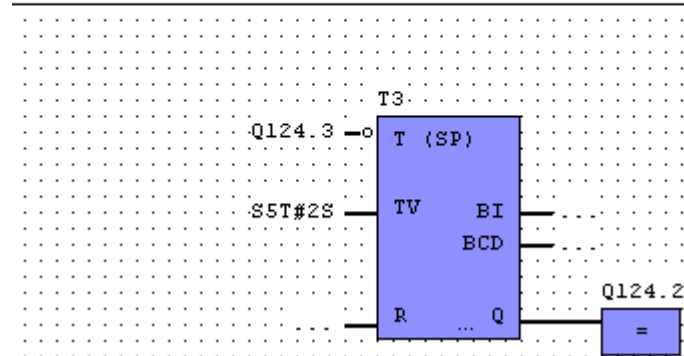
| :FC1> "" Version:1.0 Author: Name:nb Family:nb  |        |             |      |      |
|---|--------|-------------|------|------|
| *   | Adress | Declaration | Name | Type |
|   | 0.0    | in -->      | IN0  | INT  |
|   | 2.0    | in -->      | IN1  | INT  |
|   |        | out <--     |      |      |
| Network 1: <Title of Network>   |        |             |      |      |
| Network comment=  |        |             |      |      |
| <div style="display: flex; justify-content: space-around;"> <span>STL</span> <span>FBD</span> <span>LAD</span> </div> |        |             |      |      |



Network 3: Network title=

Network comment=

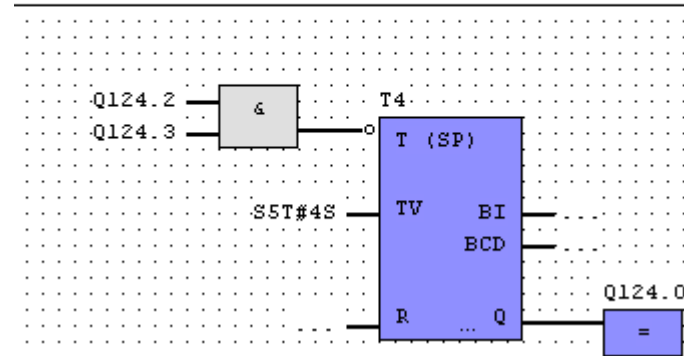
STL FBD LAD



Network 4: Network title=

Network comment=

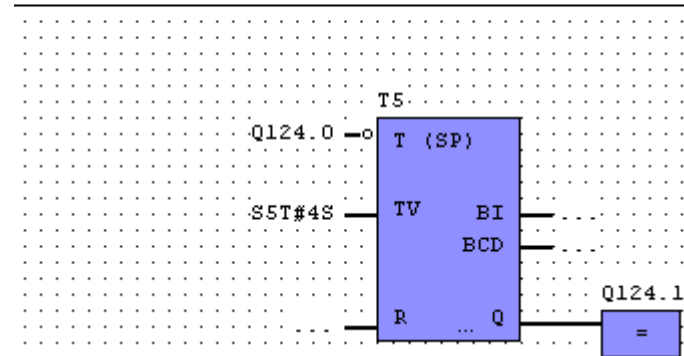
STL FBD LAD



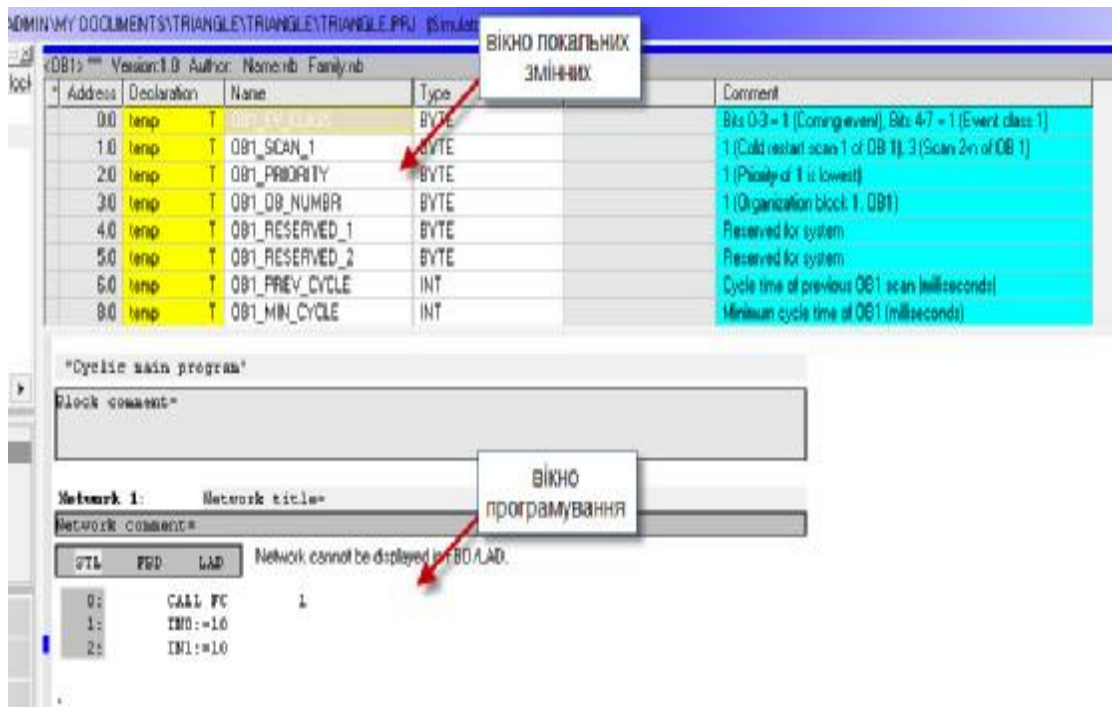
Network 5: Network title=

Network comment=

STL FBD LAD




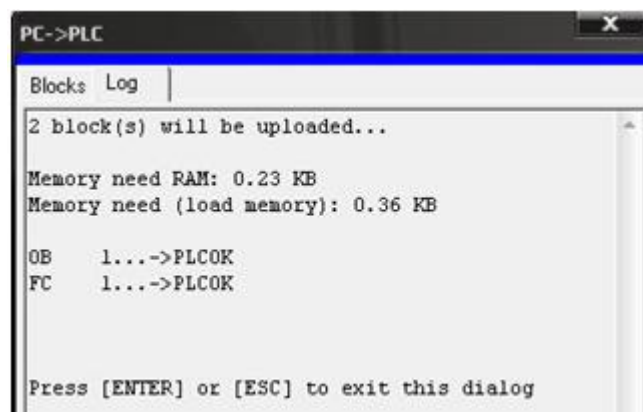
За допомогою кнопки на екрані відкрити блок «OB1» і перейти на STL-мову програмування. Набрати програму виклику функції «FC1» та передачі до неї значень змінних, як показано на рисунку:




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

### Перевірка коду програми

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



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

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

FC1 of TREUGOLNIK.PRJ (OFFLINE)

<FC1> "" Version:1.0 Author: Name:nb Family:nb

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

Network 1: <Title of Network>

Network comment=

STL FBD LAD

The diagram shows a network with the following components and connections:

- Inputs: #INO (value 10) and #IN1 (value 10).
- Block: CMP ==I (Comparison block).
- Block: T1 T (SP) T#000.0 (Timer block).
- Block: TV BI (Timer block).
- Block: S5T#2S (Timer block).
- Block: S5T#2S (Timer block).
- Block: R Q (Reset block).
- Block: Q124.0 (Output block).

### Завантаження програми до ПЛК



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

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



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

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

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

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

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

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

#### ***Зміст звіту***

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

#### ***Контрольні питання***

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

### **12.5 Лабораторна робота №5**

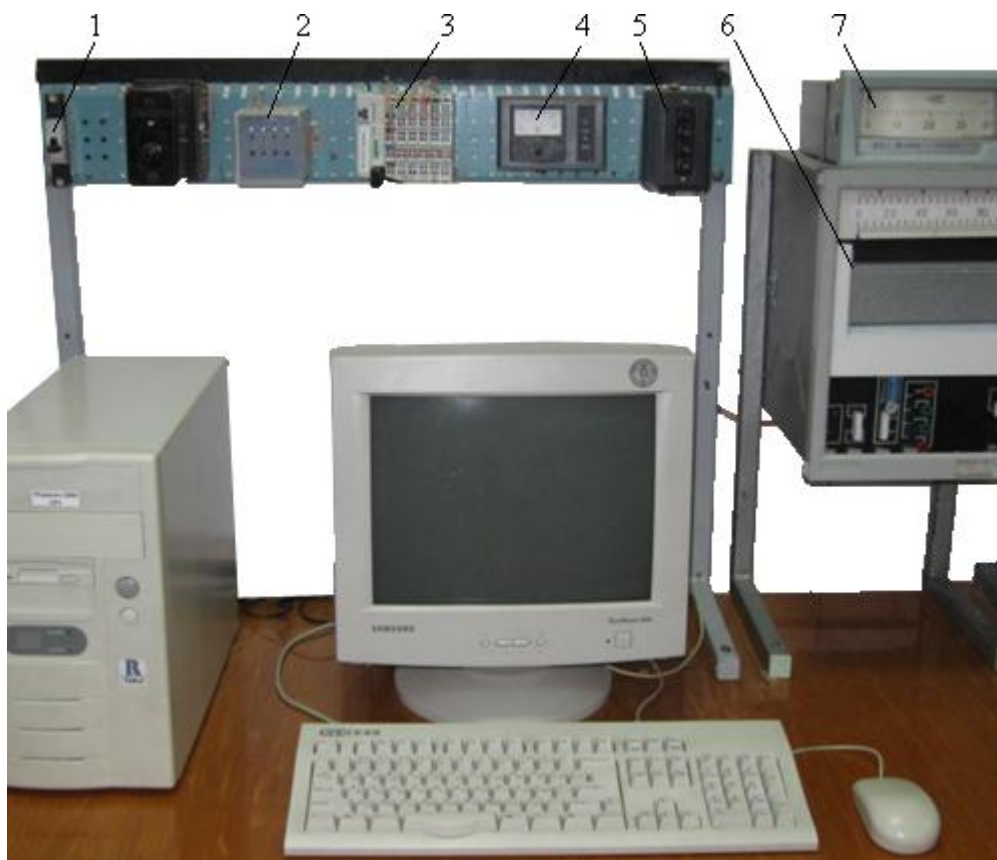
Програмування та дослідження роботи контролера WAGO 750-841

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

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

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

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



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

Рисунок 12.11 – Лабораторна установка

### ***Контролер WAGO 750-841***

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

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

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

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

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



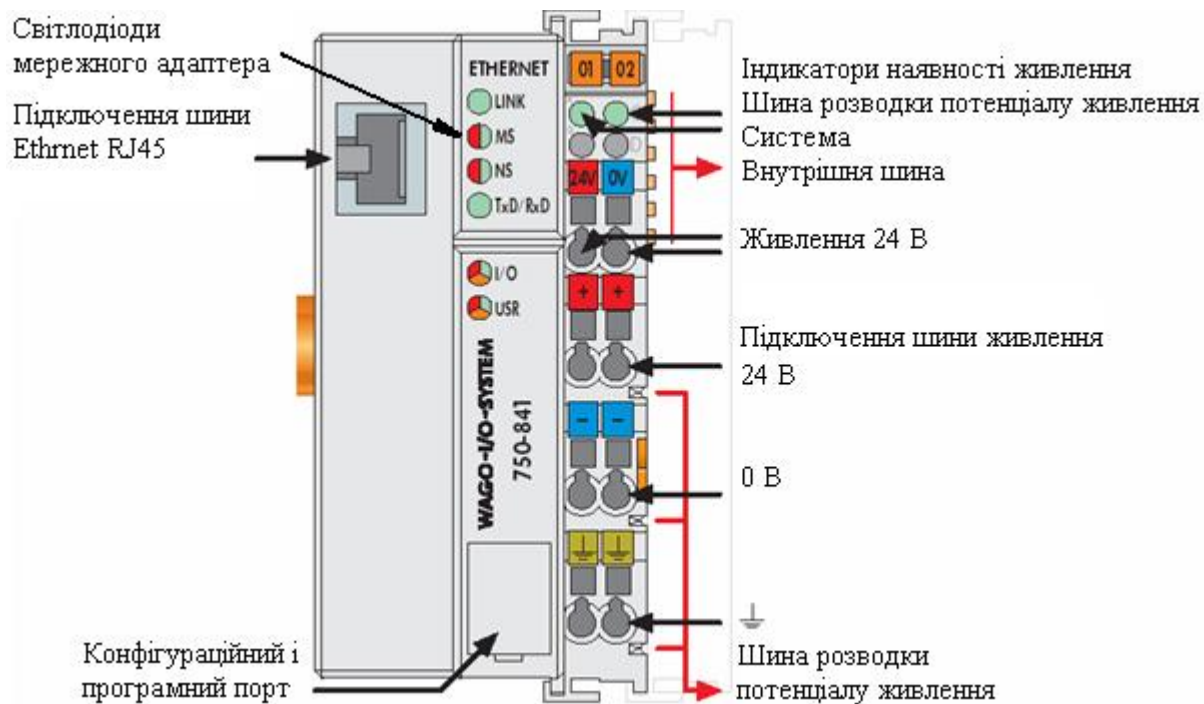


Рисунок 12.12 – Fieldbus контролер ETHERNET TCP/IP

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

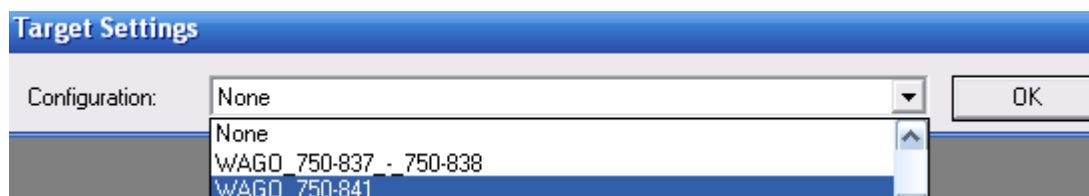
Решта інформації про контролер WAGO 750-841 надається у відповідному розділі даного посібника.

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

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


### ***Створення проекту***

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

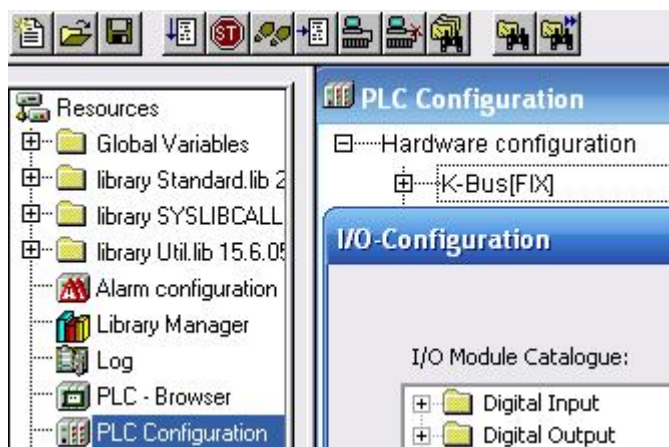


Зі списку Configuration вибрати контролер WAGO-750-841 і натиснути ліву клавішу миші. У діалоговому вікні New POU, що відкриється, позначити LD-мову, вибрати Program, як тип POU, залишити головний шаблон PLC\_PRG і натиснути ОК.

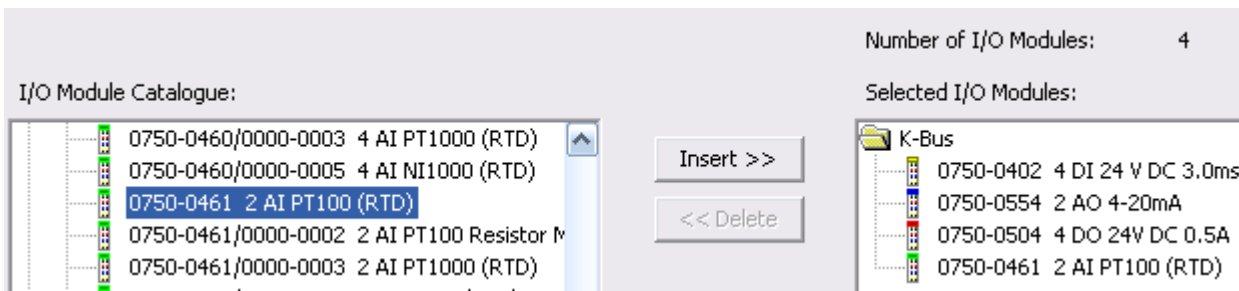
На екрані з'являться два вікна, ліворуч вікно організатора проекту POUs, а праворуч робоче поле для створення програмного коду.


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

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

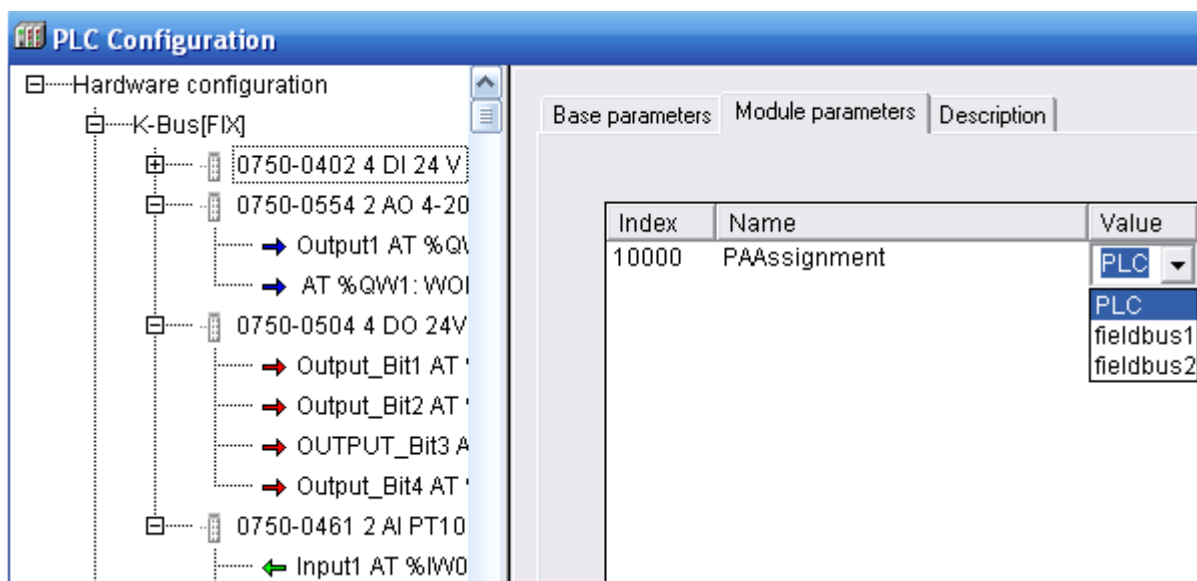


Для реалізації алгоритму управління положенням каретки вторинного приладу використовуються тільки дискретні виходи, проте необхідно при конфігуруванні PLC вказати всі модулі, що встановлені в слотах контролера. У каталозі модулів вводу – виводу, що в лівій частині вікна, двічі натиснути на “Digital Input”,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



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


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

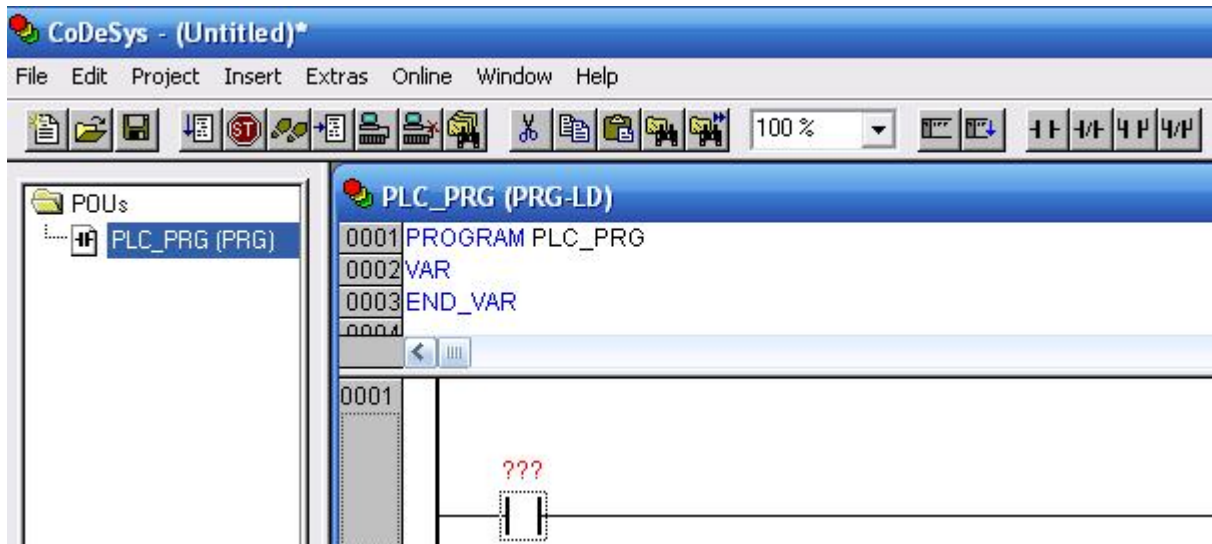


Після цього необхідно присвоїти імена вхідним і вихідним модулям контролера. Оскільки при керуванні використовується модуль дискретного виводу 0750-0504 двічі натиснути лівою клавішею по його назві.


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


## Розроблення програмного коду



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




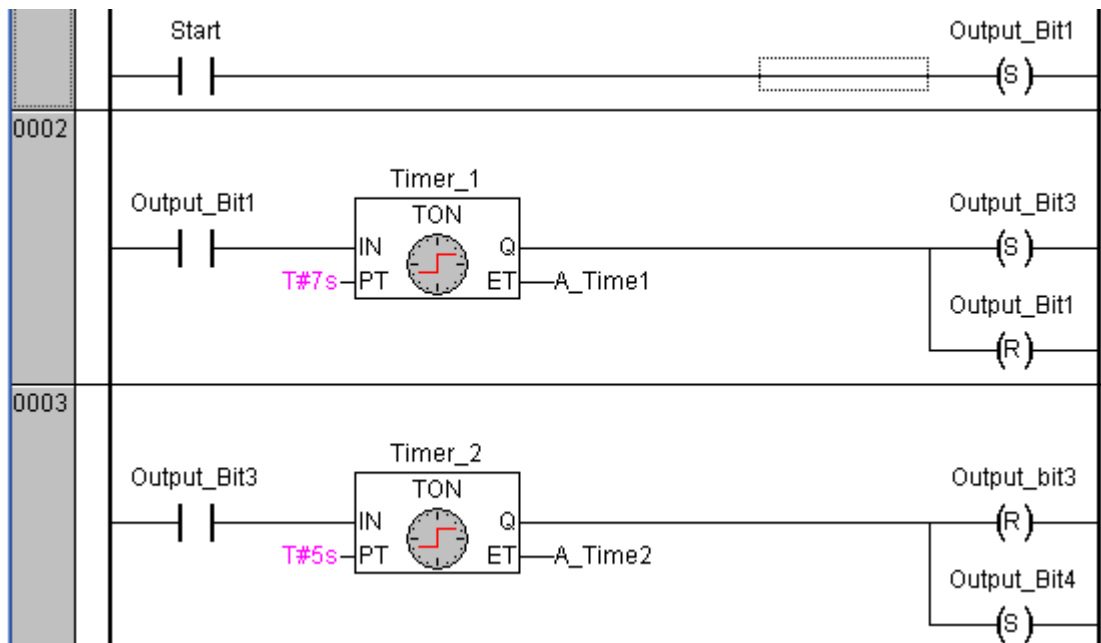
Замість знаків питань надрукувати ім'я контакту Start. Натиснути на клавіатурі Enter. У діалоговому вікні, що з'явиться, зберегти найменування (Name: Start) і логічний тип (Type: BOOL). Клас змінної - глобальний (Class:VAR\_GLOBAL), натиснути ОК. Змінна Start з'явиться у вікні глобальних змінних проекту.

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

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

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

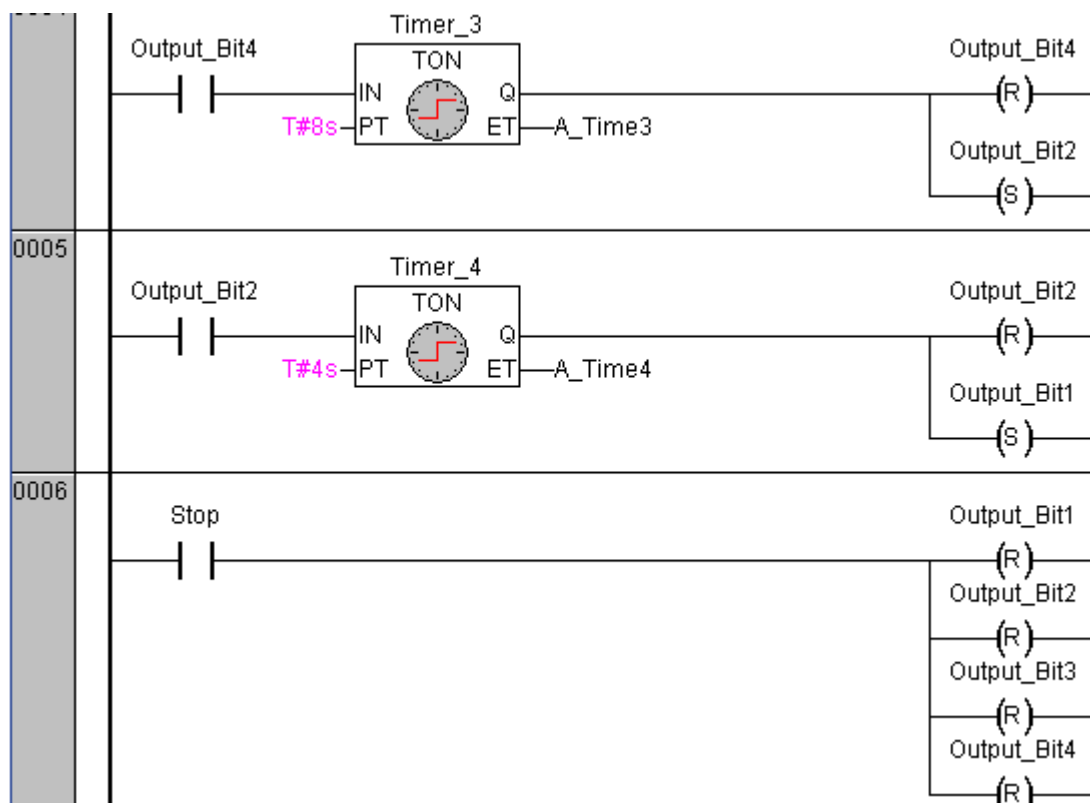
Створити ще одну мережу кнопкою . Додати контакт (contact) Output\_Bit3, встановити таймер TON з ім'ям Timer\_2. Замінити три знаки питання на вході PT константою T#5s, а на виході ET встановити змінну A\_time2 типу TIME. Розмістити на цій мережі Reset- і Set-котушки з іменами Output\_Bit3 і Output\_Bit4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



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

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

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



Протягом створення програмного коду сформувався список оголошених змінних:

```


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

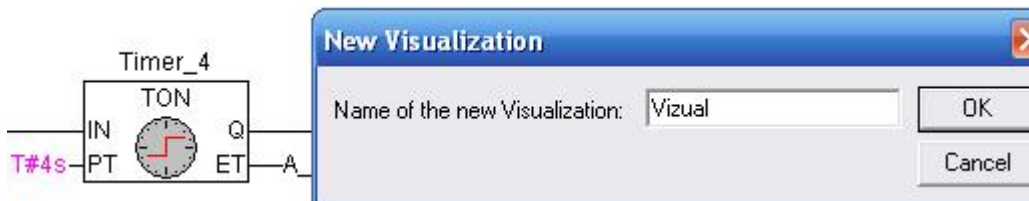
```

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

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

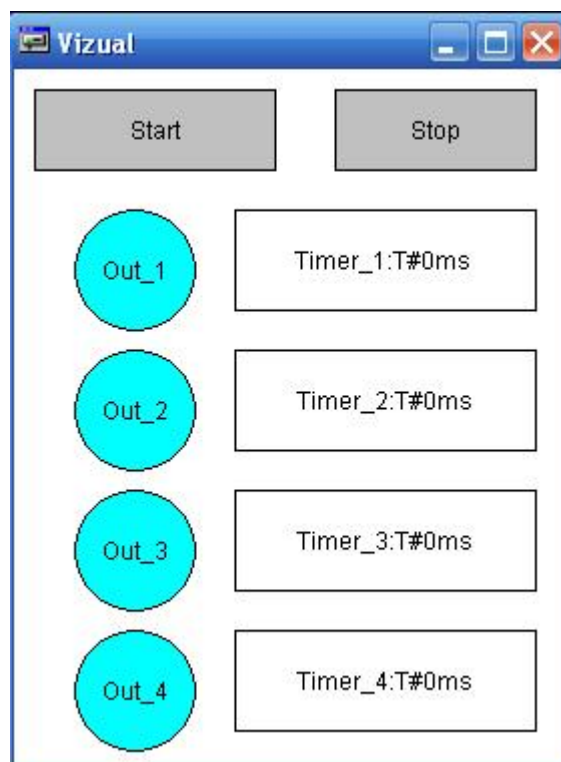
### *Візуалізація*


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



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

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

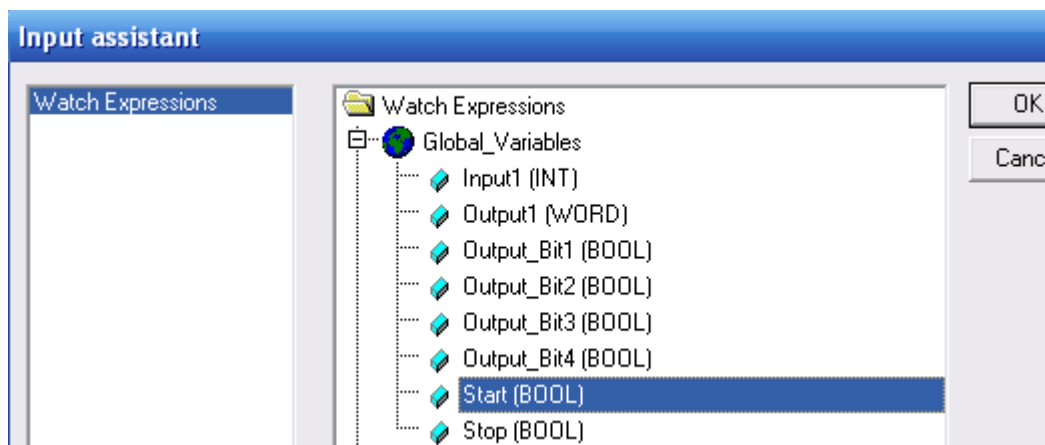


Для візуалізації кнопки Start, в панелі інструментів вибрати прямокутник (Rectangle) . У вікні редактора візуалізації натиснути ліву клавішу миші і розтягнути прямокутник до потрібної висоти і ширини, відпустити клавішу.

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



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



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




У категорії Input активізувати Toggle variable і ввести змінну Start.




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

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

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

Для кожного кола скоригувати настройки:

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

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

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

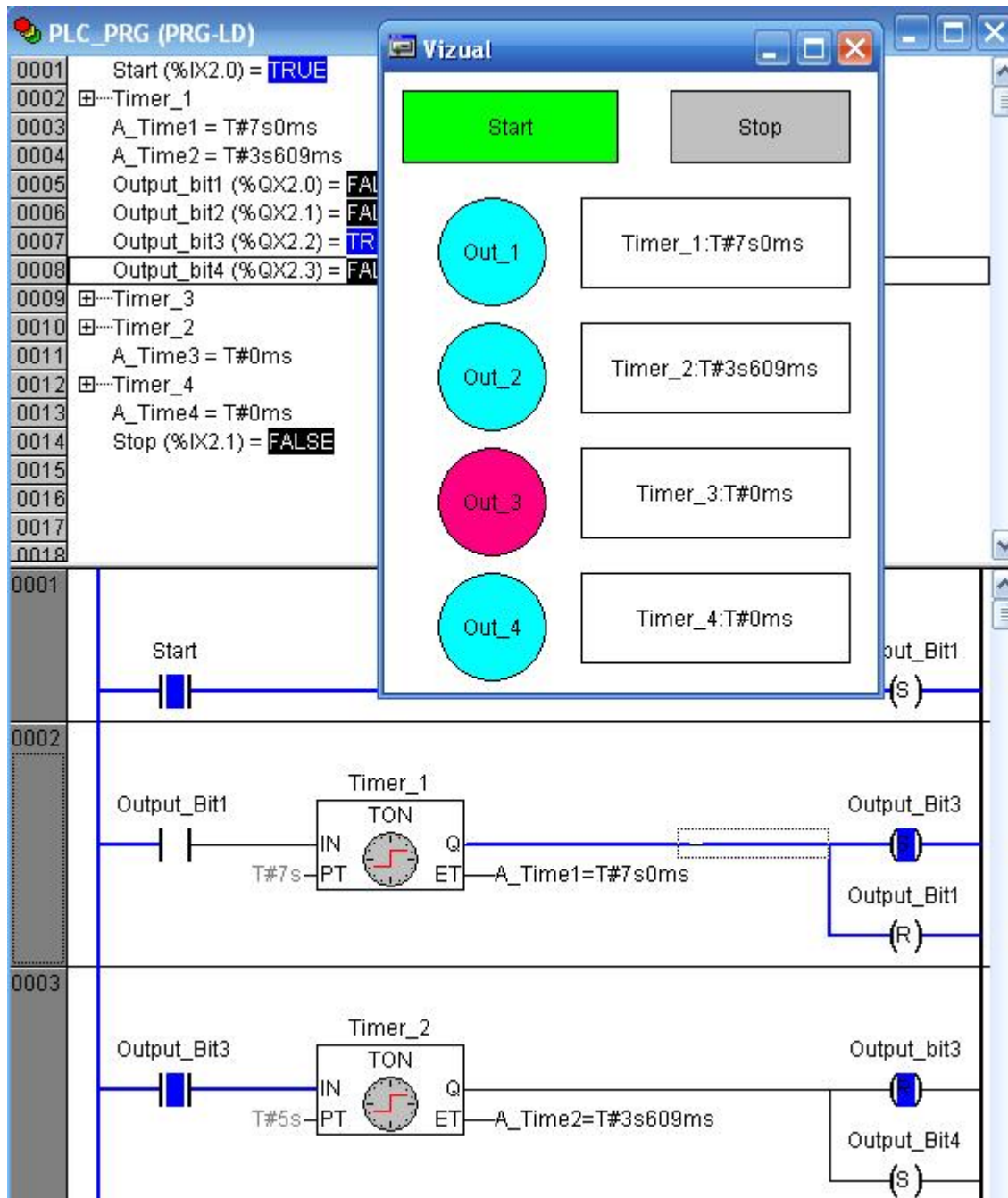
Намалювати білий прямокутник для таймерів з наступними настройками для положень Out\_1, Out\_2, Out\_3, Out\_4:

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

### ***Налагодження програми***

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

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



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

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

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

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

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

### ***Зміст звіту***

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

### ***Контрольні запитання***

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

## **12.6 Лабораторна робота № 6**

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

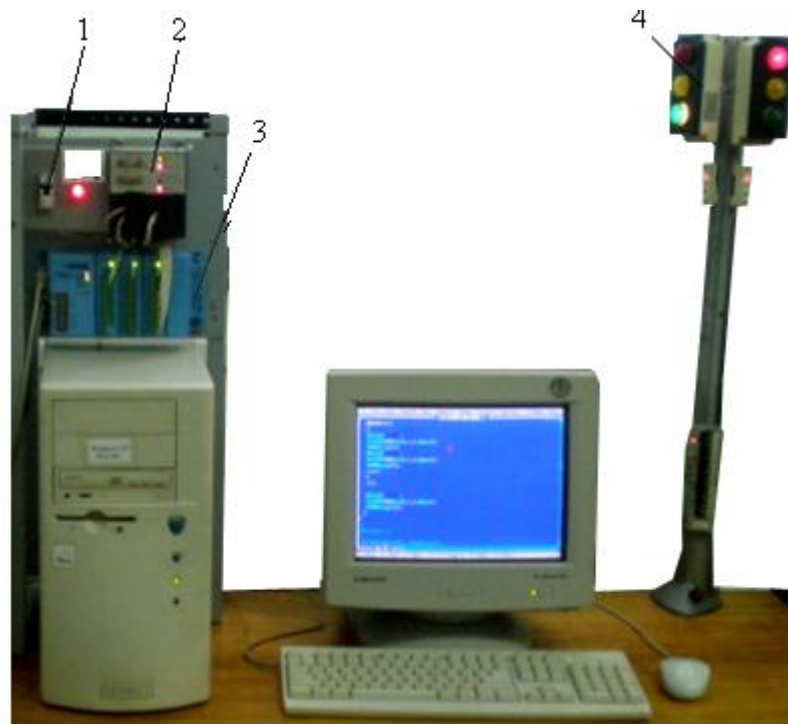
**Мета:** ознайомитися з будовою, особливостями програмування і роботи з контролером.

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

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

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

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



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

Рисунок 12.13 – Лабораторна установка

### ***Контролер ADAM 5511***

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

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

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

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

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

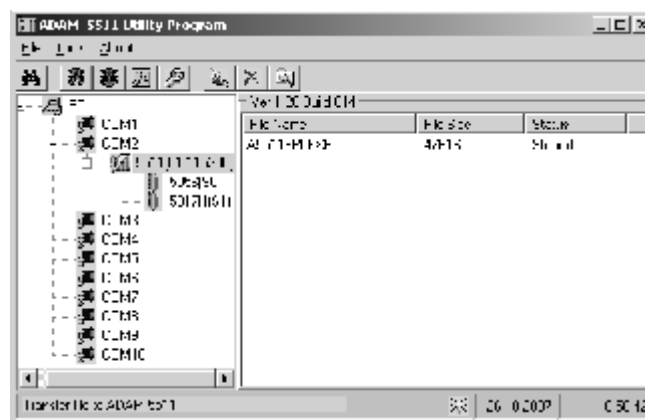
### ***Інструмент програмування контролера***

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

Ця утиліта реалізує:

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

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



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

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

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

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

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



Рисунок 12.14 – Адресація регістрів модулів ПЗО серії «ADAM5000»

### **Виконання лабораторної роботи**

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

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

При створенні проектного коду використовуються функції:

- з бібліотеки «Utility.lib»:

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

Синтаксис: `void check_prog_stop(void);`

- Пауза при виконанні програми.

Синтаксис: void ADAMDelay(unsigned int TC);

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

- з бібліотеки «IO.lib»:

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

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

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

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

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

- Установка дискретних виходів ПЗО «ADAM-5050»

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

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

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

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

Таблиця 12.14

Стани дискретних виходів

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

Де:

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

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

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

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

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

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

Структура програми складається з наступних блоків:

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

```

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

```



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

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

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

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

### ***Зміст звіту***

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

### ***Контрольні запитання***

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

## **12.7 Лабораторна робота № 7**

Програмування та дослідження роботи контролера ADAM-5510KW

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

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

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



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

Рисунок 12.15 – Лабораторна установка

### ***Контролер ADAM 5510KW***

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

#### ***Створення нового проекту***

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

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

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

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

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

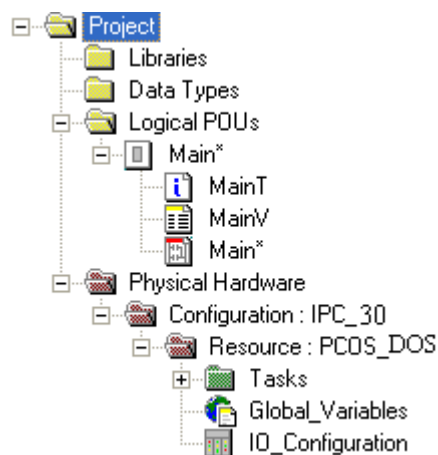
Вибрати тип конфігурації проекту IPC\_30, тому що компілятор генерує intel- код для системи виконання ProConOS 3.0. Натиснути кнопку Next, ввести ім'я ресурсу – Resource і вибрати тип ресурсу - PCOS\_DOS.

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

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

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

В меню дерева проекту розкрити теку Physical Hardware.



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

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

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

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

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

Клацнути правою 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 POU, відповідно до даних наведених у діалоговому вікні "Data Area" :

| Field                   | Value |
|-------------------------|-------|
| Start user              | 0     |
| End user / Start system | 0     |
| End system (max 63999)  | 63999 |
| Reserve per POU         | 500   |
| Start user              | 64000 |
| End user / Start system | 64000 |
| End system (max 65530)  | 65530 |
| Reserve per POU         | 10%   |

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

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

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

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

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

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

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

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

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

Для стартової адреси система сама знайде вільну адресу.

Місткість каналів модулів визначається їх призначенням:

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

Кінцеву адресу система розраховує автоматично.

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

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

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

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

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

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

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

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

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

Ідентифікатор пристрою (Device ID) дорівнює 1.

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

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

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

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

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

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



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

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

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

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

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

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

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

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

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

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

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

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

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



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

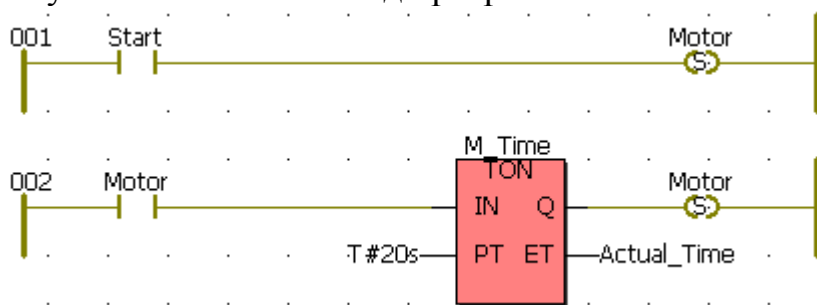
Клацнути ОК і на РТ-вході лічильника з'явиться константа часу.


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

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

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

На робочому полі залишиться код програми:



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

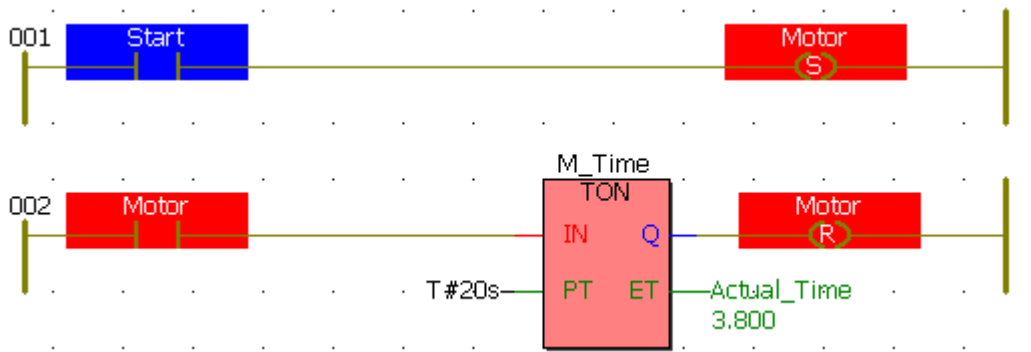
### ***Завантаження проекту до ПЛК***

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

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

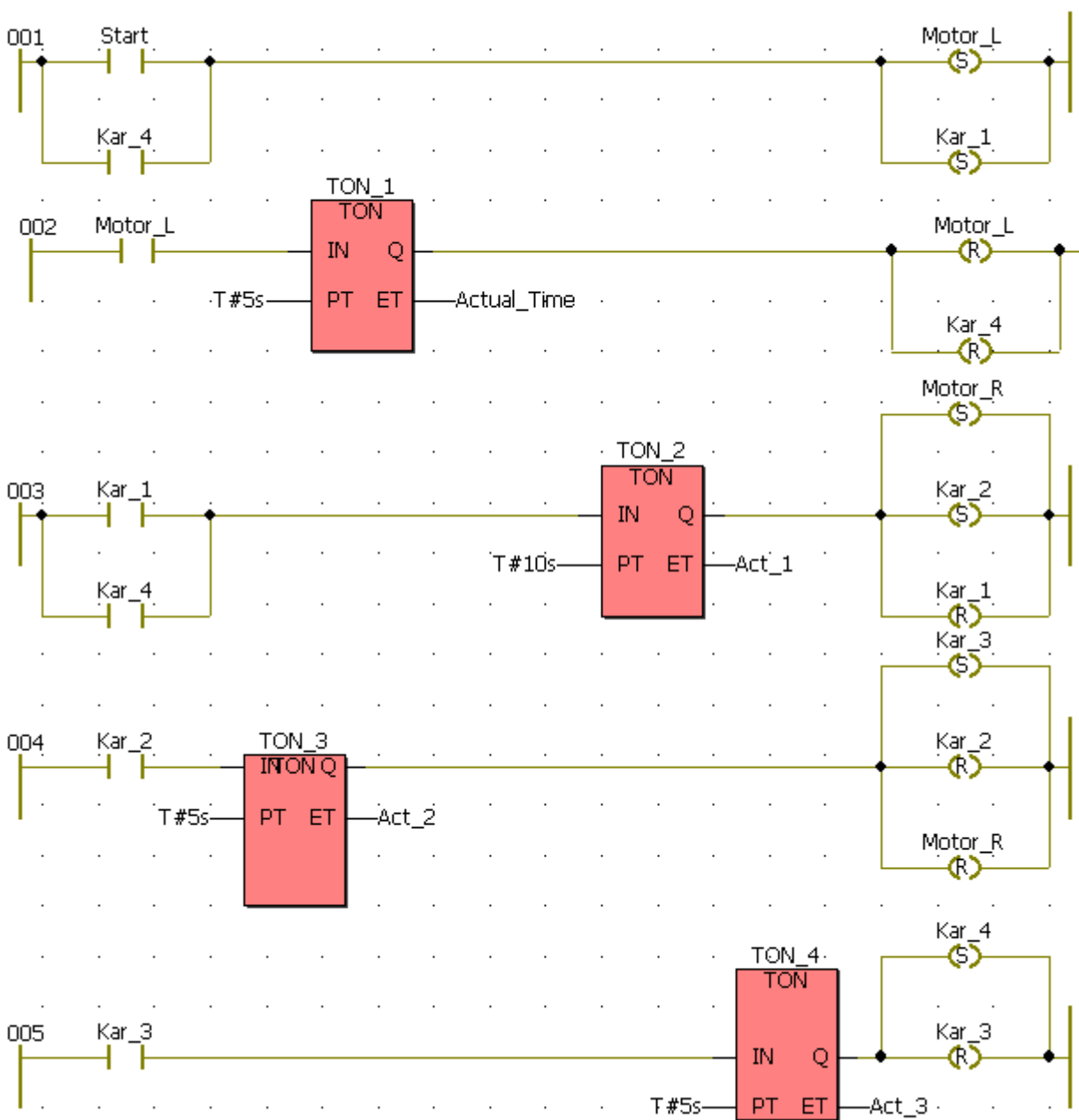
Після завантаження проекту і його активації, за умови відсутності помилок, можна запустити виконання програми, користуючись Cold start (Холодний старт), Warm start(Теплий старт) або Hot start(Гарячий старт) у вікні Resource (Ресурс).

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



### Завдання для самостійної роботи

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



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

### ***Зміст звіту***

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

### ***Контрольні запитання***

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

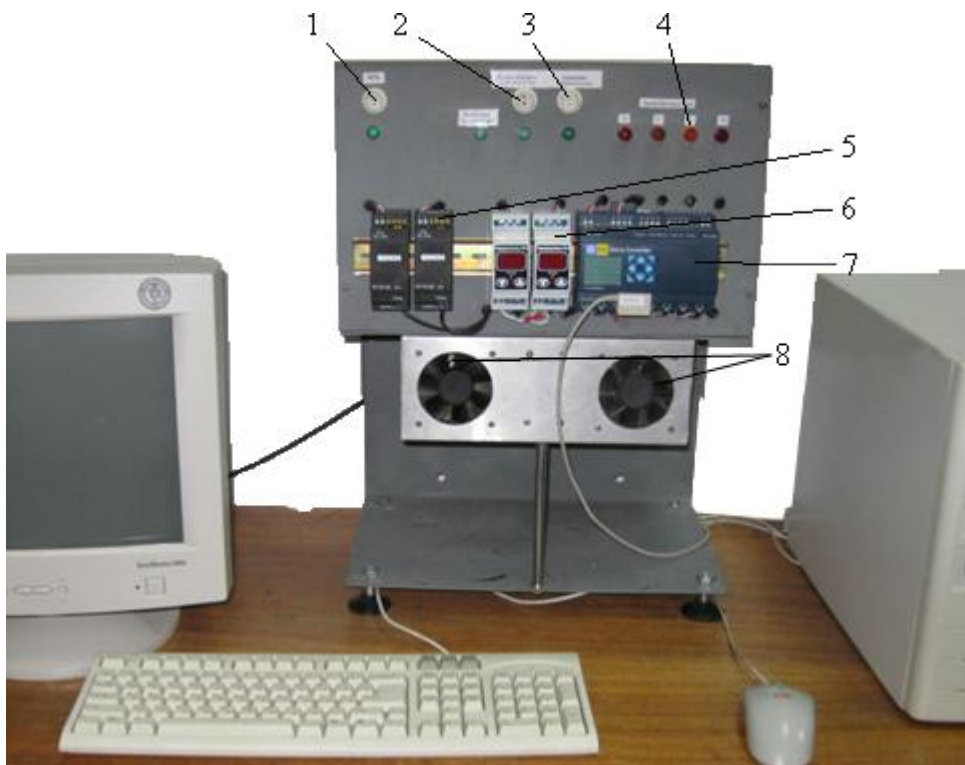
## **12.8 Лабораторна робота №8**

Програмування і дослідження роботи контролера Durus.

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

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

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



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

Рисунок 12.16 – Лабораторна установка

### ***Контролер DURUS***

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



Рисунок 12.17 – Контролер DURUS

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

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

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

Напруга живлення контролера залежить від вибраної моделі:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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



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

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

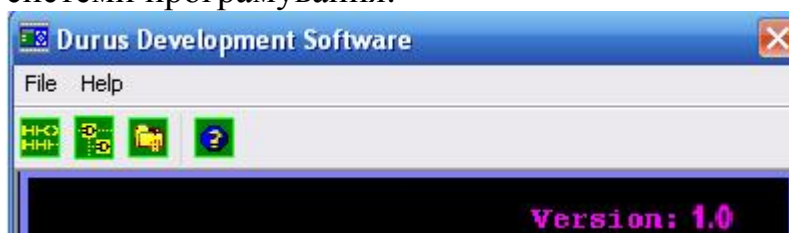
Для створення затримки при вмиканні лампи на бс, курсор перевести на друге знакомісце і натиснути кнопку SEL. Кнопкою ↑ встановити час затримки б і натиснути ОК. Для виходу з меню таймера, натиснути ESC.

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

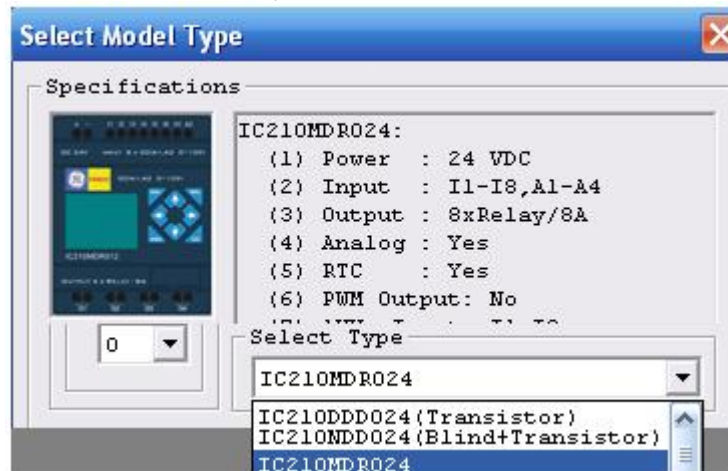
Для припинення роботи контролера у головному меню скористатися альтернативою STOP.

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

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




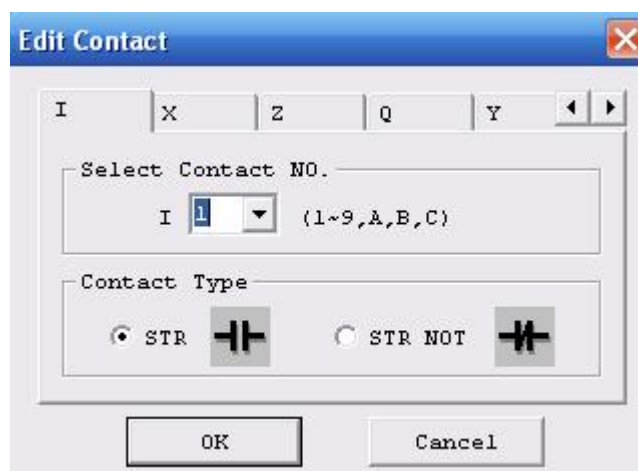
Увійти в меню File→New→NewLAD, щоб створити проект LAD-мовою. З'явиться екран LAD.Version:1.0 з панеллю інструментів. Лівою клавiшею миші натиснути на кнопку New і у вікні Selekt Model Type, що з'явиться, вибрати тип контролера IC210MDR024, натиснути OK:



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




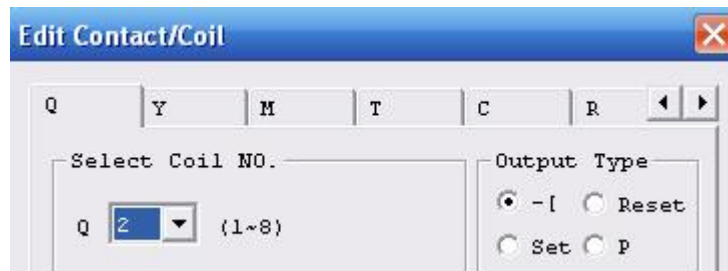
Для створення першого сегменту програми, який забезпечить вмикання першого вентилятора, в панелі інструментів вибрати кнопку I-контакту  і натиснути ліву клавiшу миші. Перевести курсор у ліву крайню клітинку першого сегменту програми і повторно клацнути. У вікні Edit Contact, що з'явиться, призначити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




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

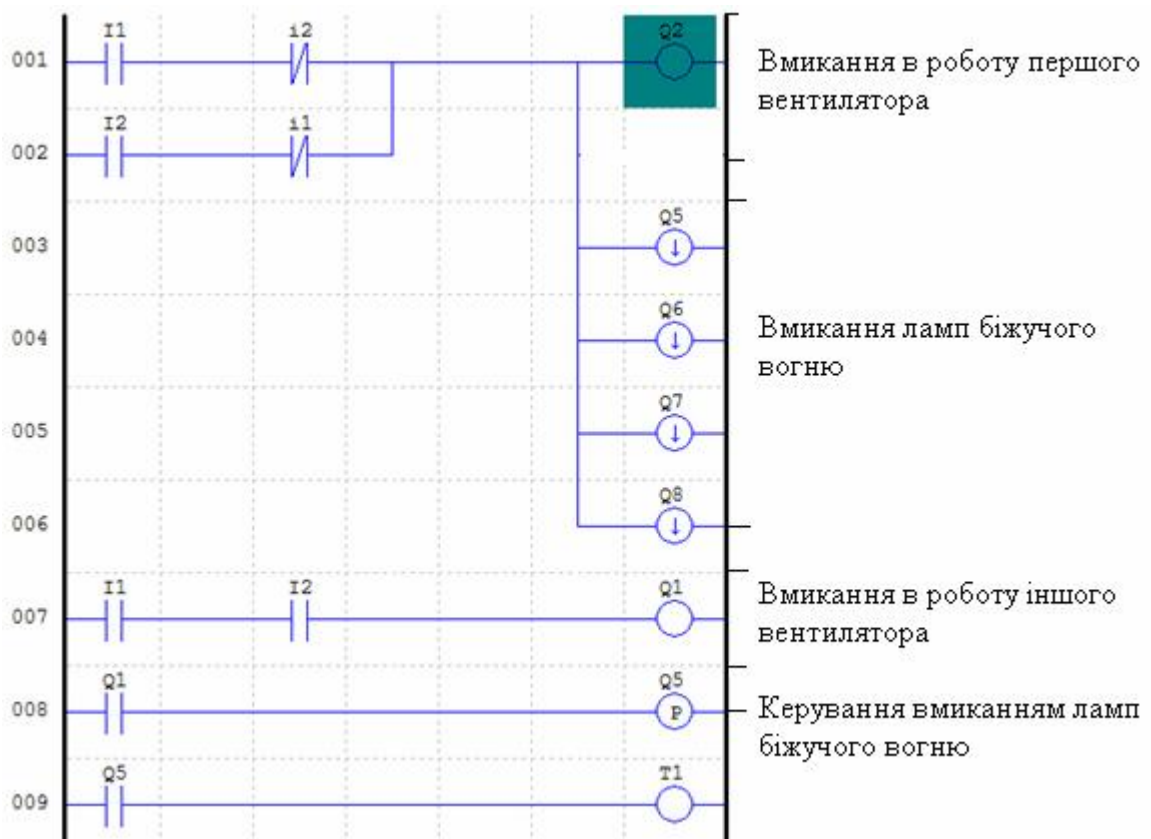


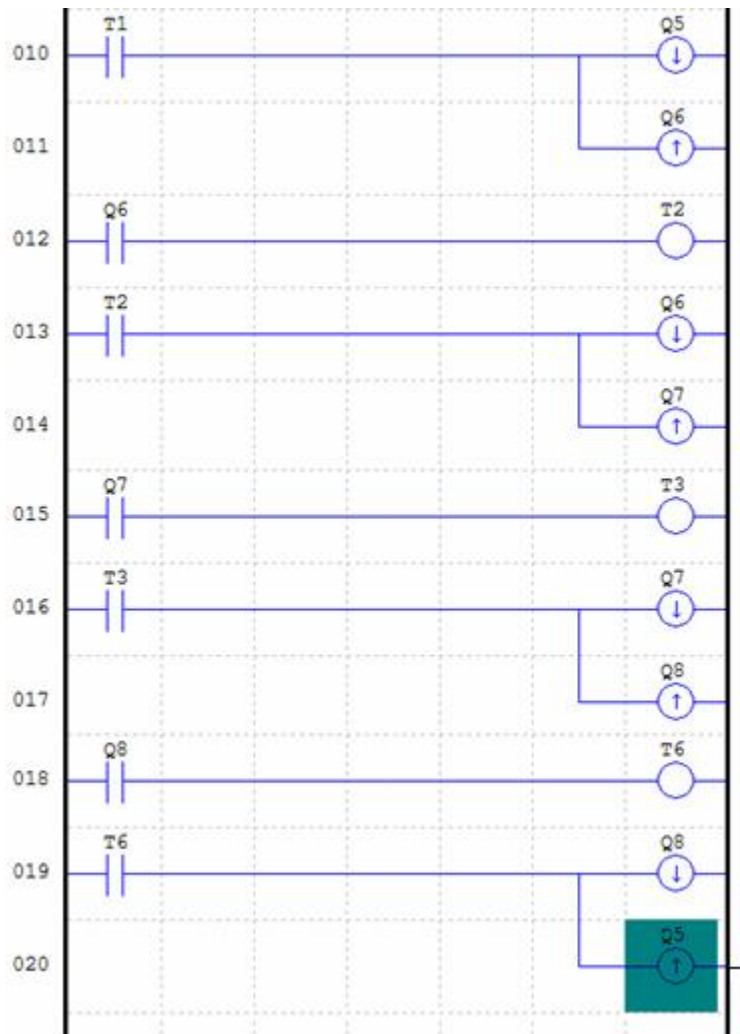
(інвертований контакт) і натиснути ОК. За допомогою кнопки  у панелі інструментів продовжити лінію сполучення до передостанньої клітинки редакційної сітки, а в останню встановити пряму котушку Q, позначивши у вікні Edit Contact/Coil індекс змінної -2 і її тип ”-( ”:



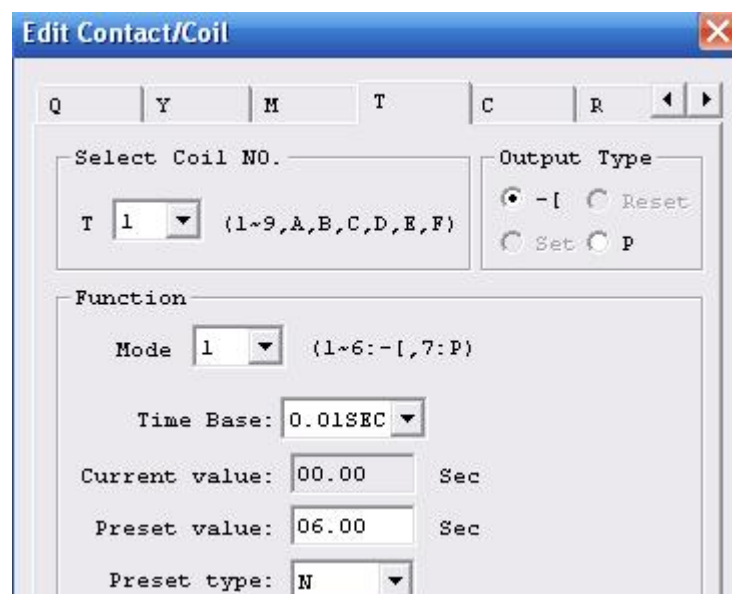
Створити таку саму послідовність контактів у другому сегменті програми тільки прямому контакту призначити символічне ім'я I2, а інвертованому – i1. За допомогою кнопки  у панелі інструментів приєднати праву сполучну лінію інвертованого контакту i1 до першого сегменту програми, забезпечивши таким чином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

Завершити створення програми, користуючись наведеним кодом:




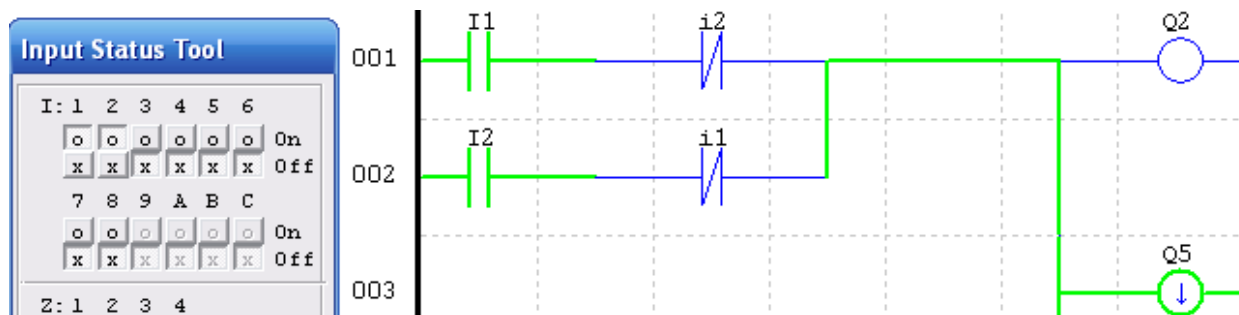


При установленні таймера скористатися в панелі інструментів кнопкою . Відкриється діалогове вікно Edit Contact/Coil, в якому призначити відповідний індекс для таймера T і час затримки Preset value: 06.00sec:




Решту параметрів залишити не змінними.

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



Перемикаючи лівою клавiшею 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 лiву клавiшу миші на Link Com Port. У вікні данного пункту вибрати COM1 PORT і натиснути клавiшу Link. Для завантаження програми у контролер, увійти в меню Operation → Write, а щоб запустити її необхідно натиснути лівою клавiшею миші кнопку  в панелі інструментів[69].

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

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

### ***Зміст звіту***

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

### ***Контрольні запитання***

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

## 13 КУРСОВА РОБОТА

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

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

### *Зміст курсової роботи.*

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

## **13.2 Приклади виконання курсової роботи**

### **13.2.1 Приклад 1**

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

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

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

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

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

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

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

Котел обладнаний барабаном, має контури водяного і парового тракту.

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

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

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

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

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

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

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

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

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

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

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

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

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



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

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

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

### *Схема інформаційних потоків*

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

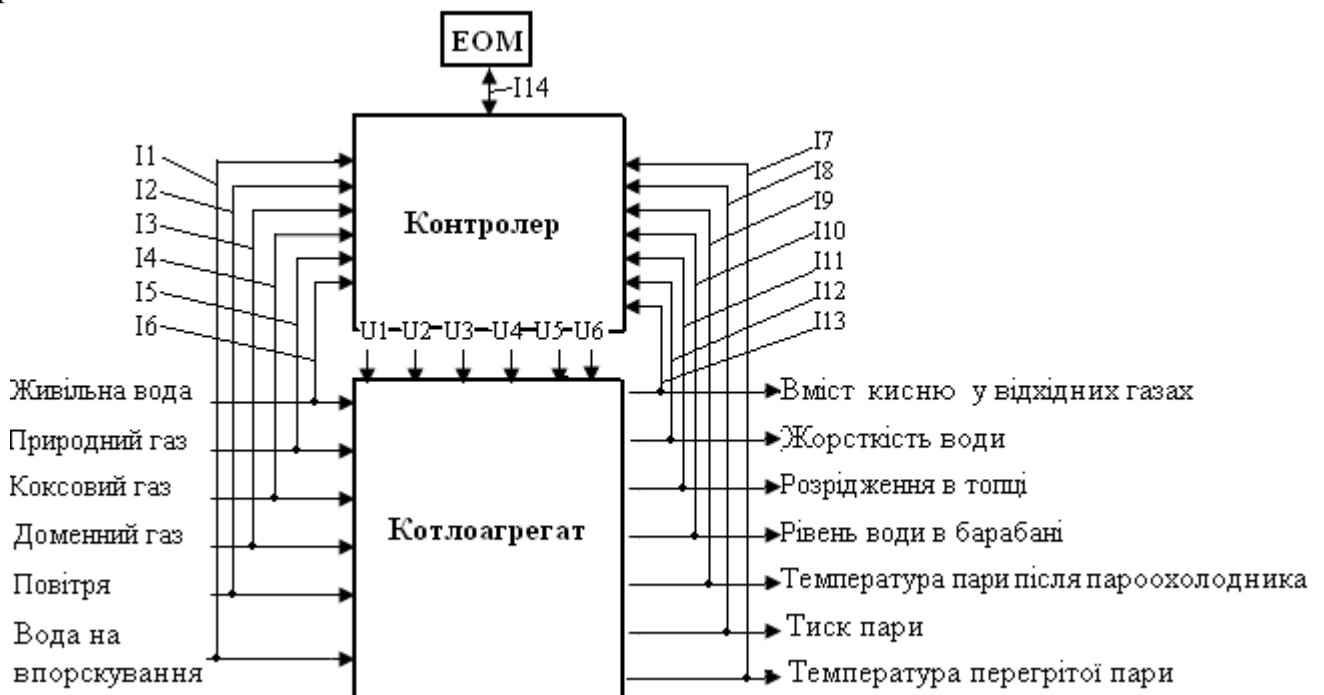


Рисунок 13.1 – Схема інформаційних потоків.

Потоки інформаційних сигналів:

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

Потоки керувальних сигналів:

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

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

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

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

Таблиця 13.1

Технологічні параметри

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

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

Таблиця 13.2

Датчики контролю технологічних параметрів

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

Таблиця 13. 3

Виконавчі пристрої, що змінюють параметр

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

### *Проектне компонування контролера Modicon TX Quantum*

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

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

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

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

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

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

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

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

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

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

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

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

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

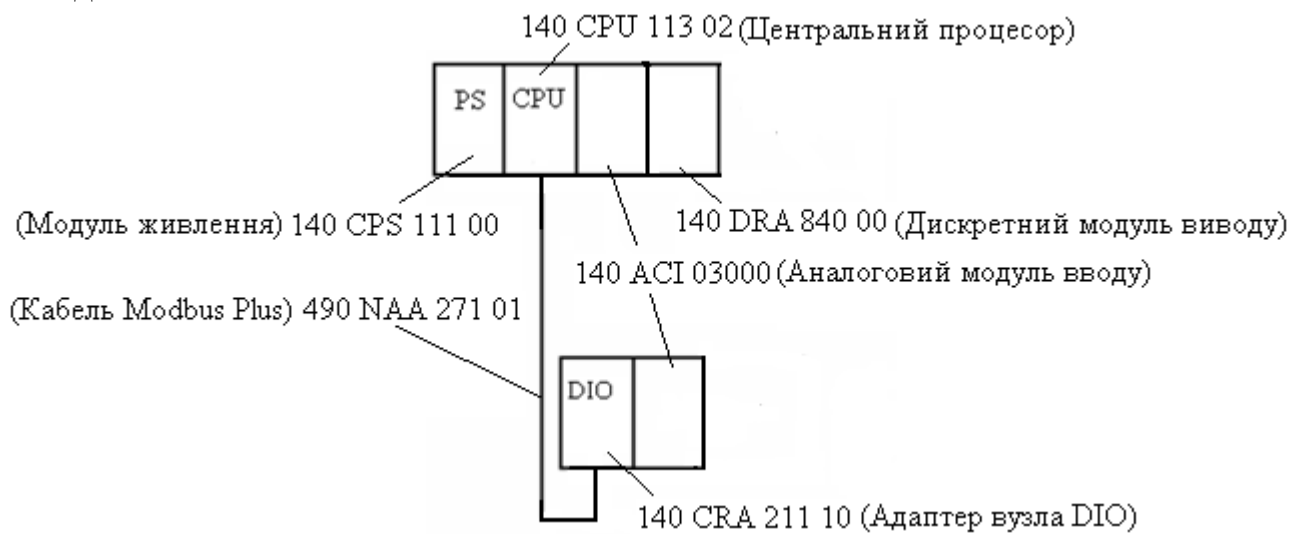
Аналоговий модуль вводу споживає струм 240мА, дискретний модуль виводу – 560мА, центральний процесор – 790мА. Тобто сумарний струм, що споживають модулі центрального шасі дорівнює 1599мА. Щоб забезпечити такий струм споживання необхідно використати модуль живлення 140CPS 111

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

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

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

Структурна схема розподіленої системи автоматичного управління має вигляд:



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

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



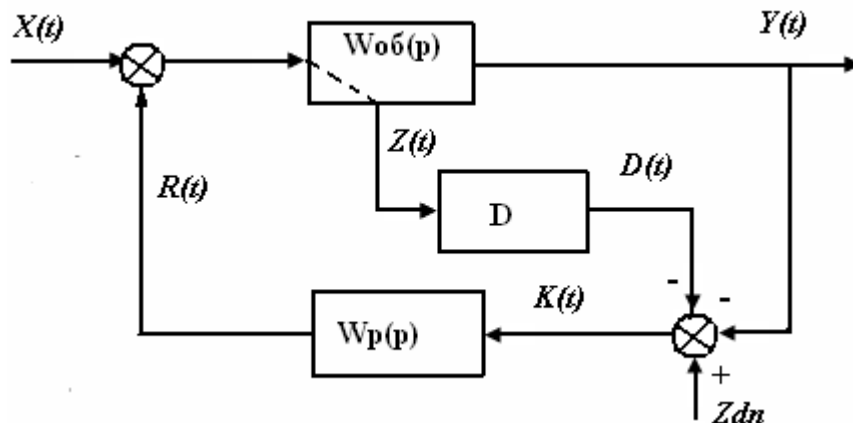
Перегрів пари до необхідної температури відбувається у пароперегрівнику, між першим і другим ступенями якого знаходиться

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

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

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

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



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

Основними перевагами цього пакету є:

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

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

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

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

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

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

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

The screenshot shows the 'I/O Map' window with 'Expansion Size' set to 144. The 'Go To' dropdown is set to 'Local/Remote (Head Slot ?)'. Below the main table, a 'Local Quantum Drop' configuration is shown with the following data:

| Rack-Slot | Module     | Detected | In Ref | In End | Out Ref | Out End |
|-----------|------------|----------|--------|--------|---------|---------|
| 1-1       | CPS-111-00 |          |        |        |         |         |
| 1-2       | CPU-x13-0x |          |        |        |         |         |
| 1-3       | ACI-030-00 |          | 300001 | 300009 |         |         |
| 1-4       | DRA-840-00 |          |        |        | 000001  | 000016  |

і розподіленого вводу-виводу:

The screenshot shows the 'I/O Map' window with 'Expansion Size' set to 144. The 'Go To' dropdown is set to 'DIO Link 0 (CPU)'. Below the main table, a 'DIO Link 0 (CPU) - Drop 1' configuration is shown with the following data:

| Slot | Module     | Detected | In Ref | In End | Out Ref | Out End |
|------|------------|----------|--------|--------|---------|---------|
| 1    | ...        |          |        |        |         |         |
| 2    | ACI-030-00 |          | 300011 | 300019 |         |         |

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



**Variable Editor**

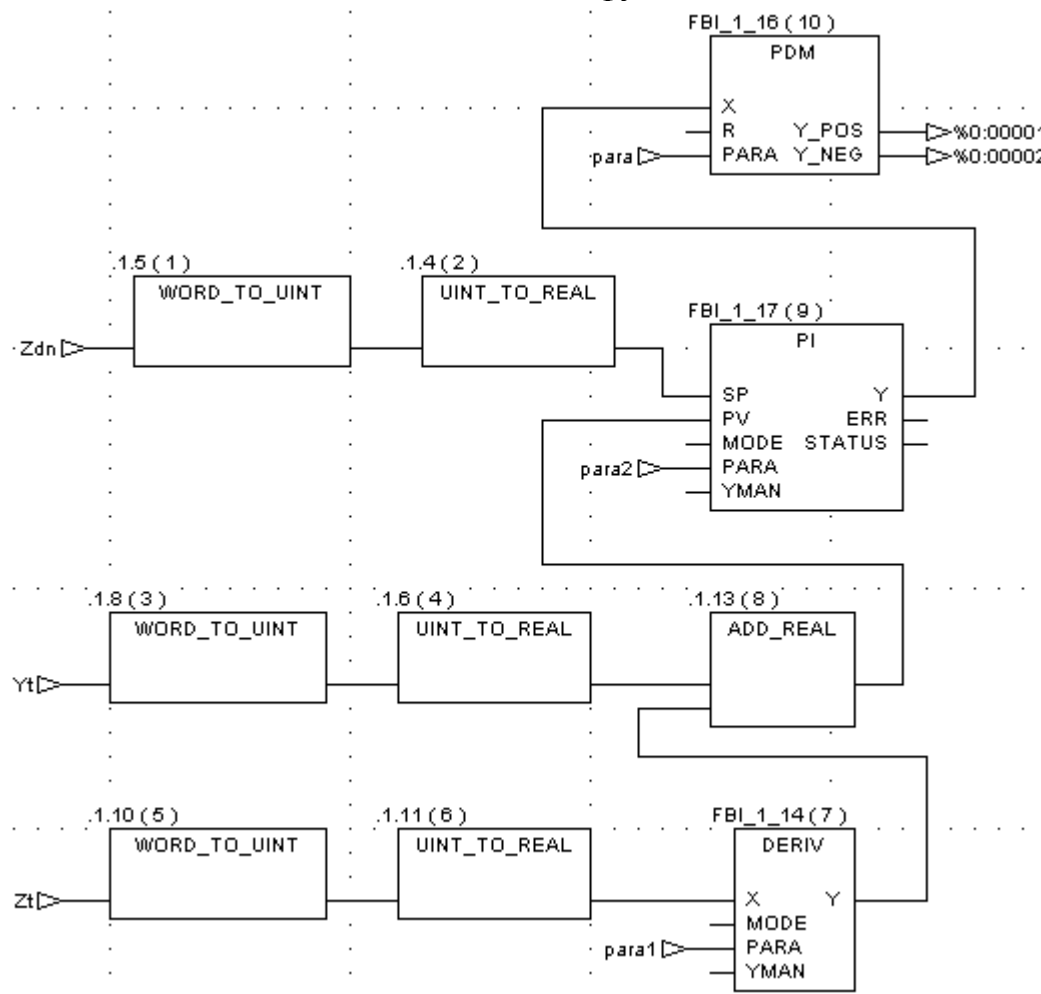
Type  
 Variables  Constants

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

Variables  Constants

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

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



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

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

### ***Висновки***

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

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

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

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

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

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

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

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

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

### *Література*

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

### **13.2.2 Приклад 2**

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

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

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

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

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

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

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

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

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

### *Схема інформаційних потоків*

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

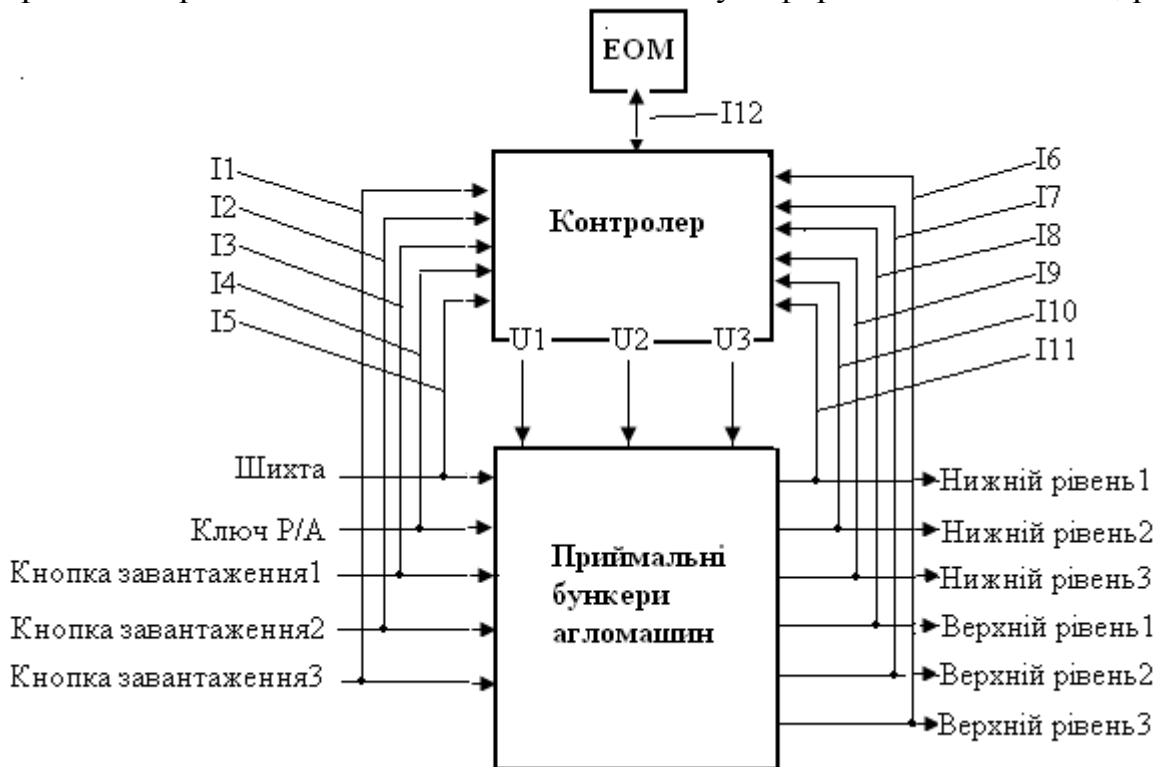


Рисунок 13.2 – Схема інформаційних потоків.

Потоки інформаційних сигналів:

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

Потоки керувальних сигналів:

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

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

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

Таблиця 13.4

Вхідні інформаційні сигнали

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

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

Таблиця 13.5 Технічні засоби, що формують вхідні сигнали контролера

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

Таблиця 13.6 Вихідні сигнали контролера

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

Таблиця 13.7 Виконавчі пристрої, що отримують вихідні сигнали

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

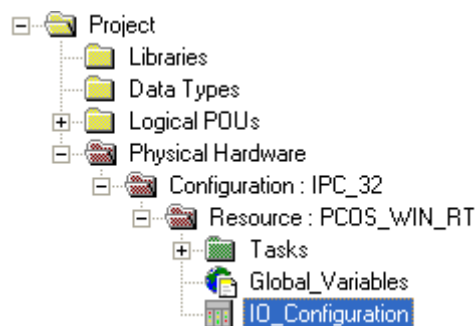
Графічний редактор системи дозволяє вільне розміщення об'єктів на робочому полі проекту, вибирати мову програмування та використовувати бібліотеку операторів, функцій і функціональних блоків. Він підтримує змішування графічних мов в одному робочому листку, встановлення нових елементів в існуючі мережі,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

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

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

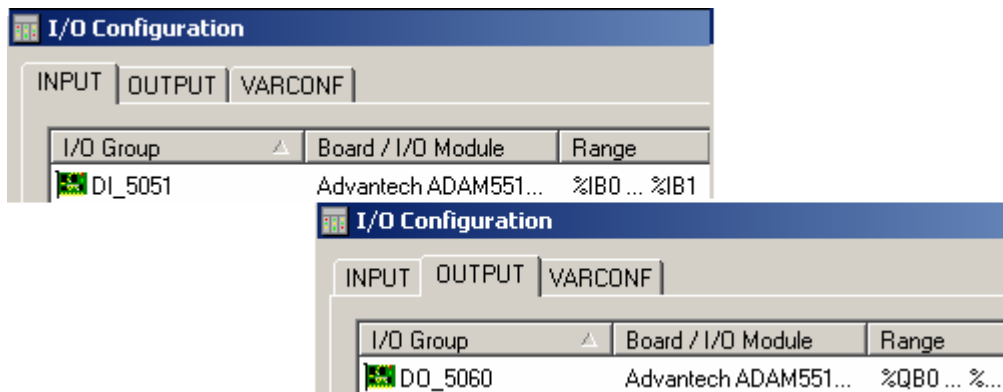
Створення додатку користувача у системі програмування MULTIPROG передбачає розробку проекту і створення програмного коду.

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



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





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

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

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

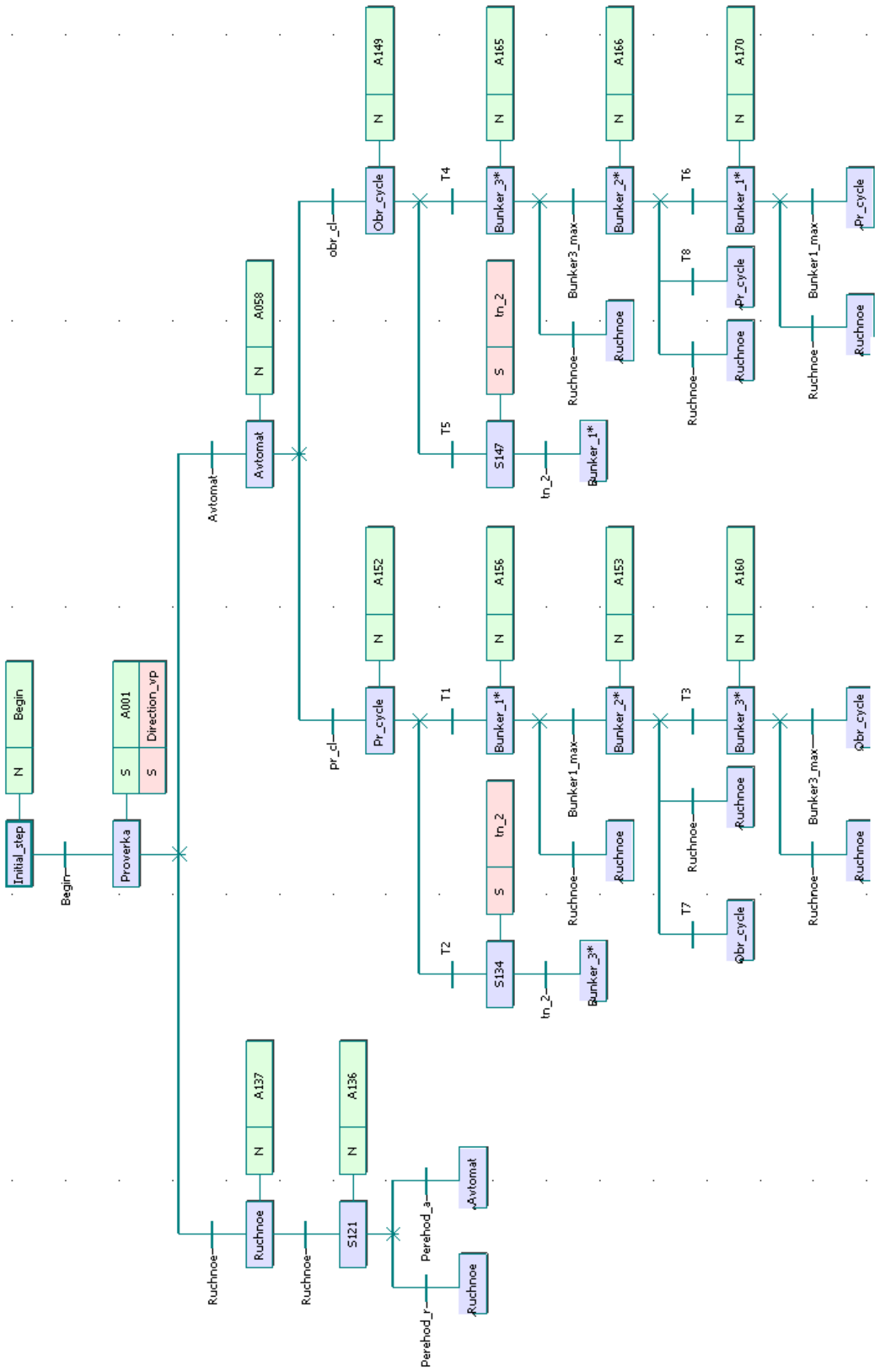
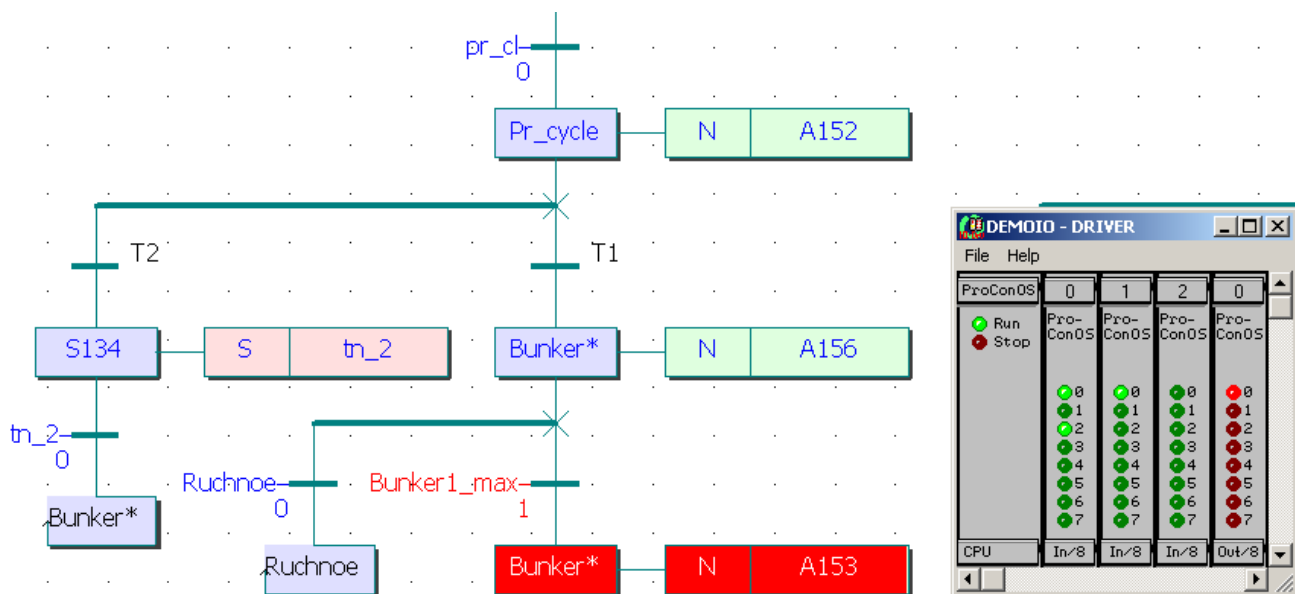


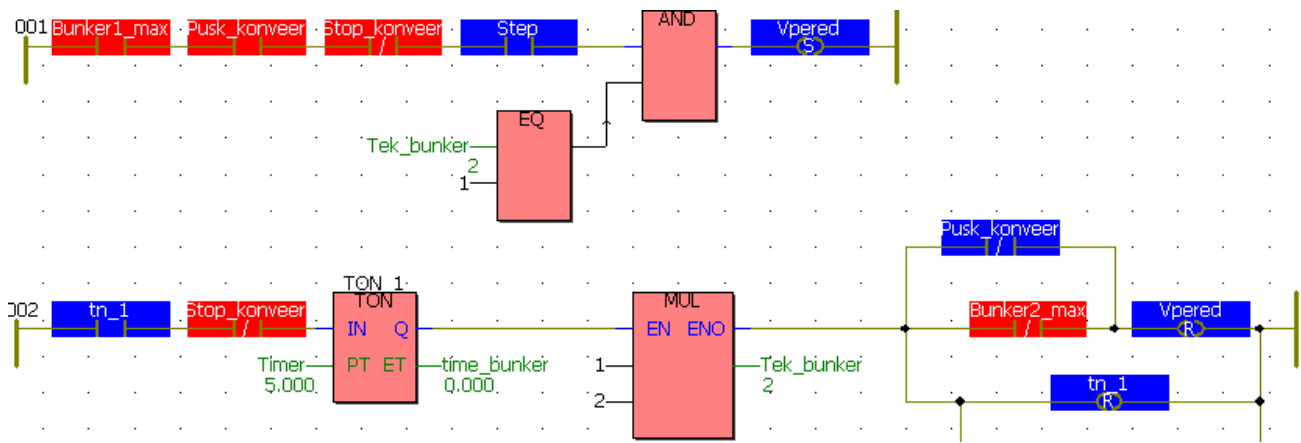
Рисунок 13.3 – SFC-структура алгоритму управління завантаженням бункерів

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

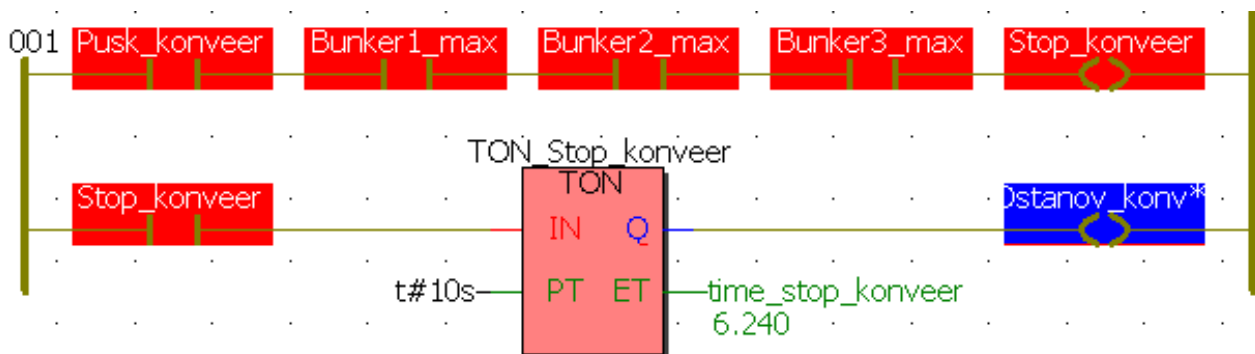
У вихідному стані всі бункери порожні і автостела знаходиться над першим бункером. При подачі сигналу про увімкнення конвеєра (0-ий вхід 0-го модуля дискретного вводу) і переведення оператором ключа управління в положення “автоматичне” (2-ій вхід 0-го модуля) починається прямий цикл завантаження. Заповнюється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



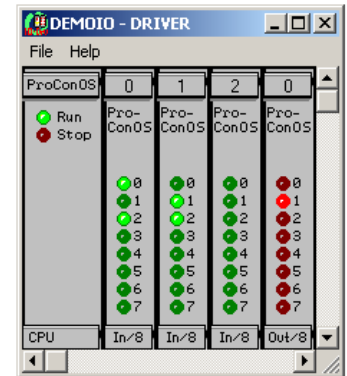
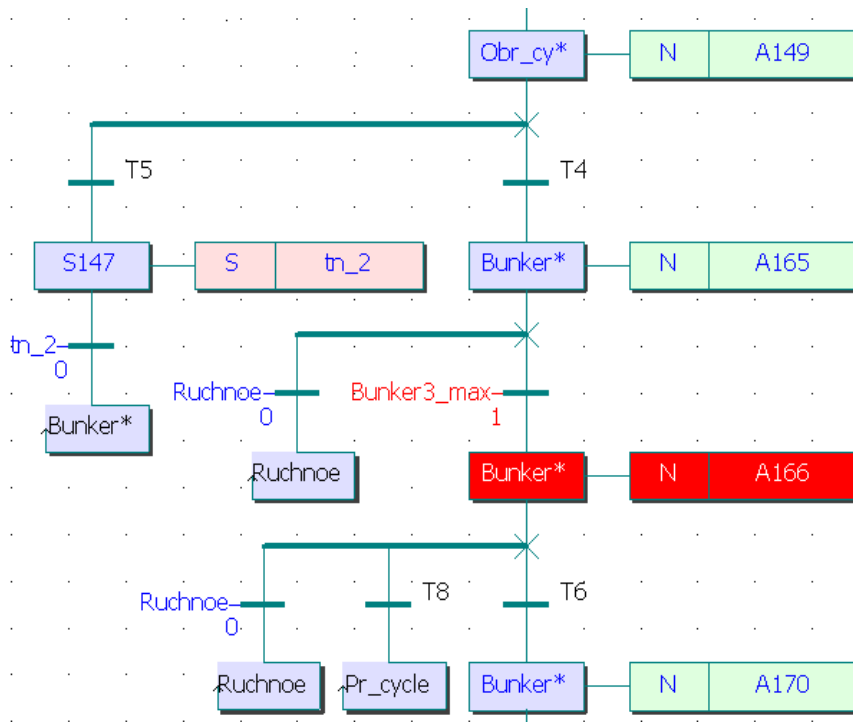
Дія A153 відповідає за переміщення автостели від 1-го бункера до 2-го, написана LD- і FBD-мовами. Фрагмент її має вигляд:



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

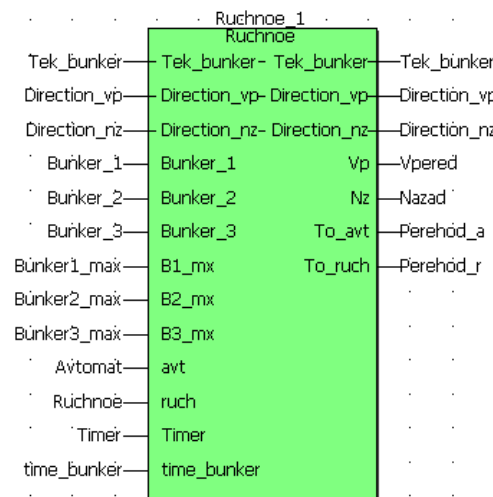


Якщо рівень у бункерах убуває, то починається цикл завантаження, при якому автостела рухається у зворотному напрямі і по черзі завантажує незаповнені бункери до максимального рівня. У наведеному нижче фрагменті програми, автостела рухається назад (1-ий вихід 0-го модуля дискретного виводу) від 3-го бункера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



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

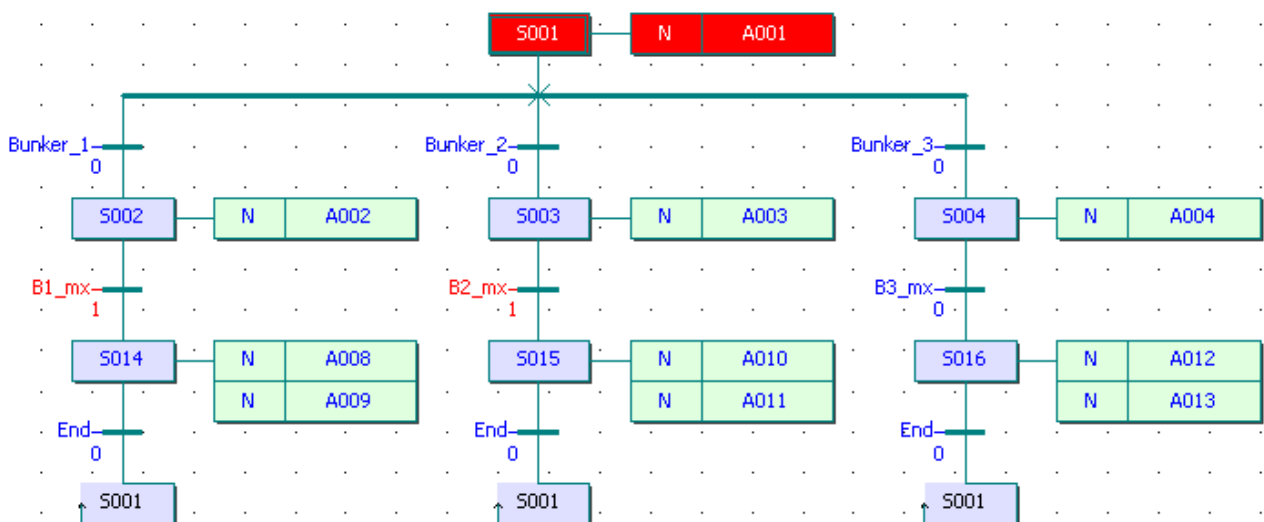
При переході у режим ручного керування, циклічне завантаження бункерів припиняється, і в програмі відбувається перехід до дії, яка відповідає за цей режим роботи. Дія містить функціональний блок користувача, який визначає номер поточного бункера, що завантажується, номер бункера визначеного оператором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



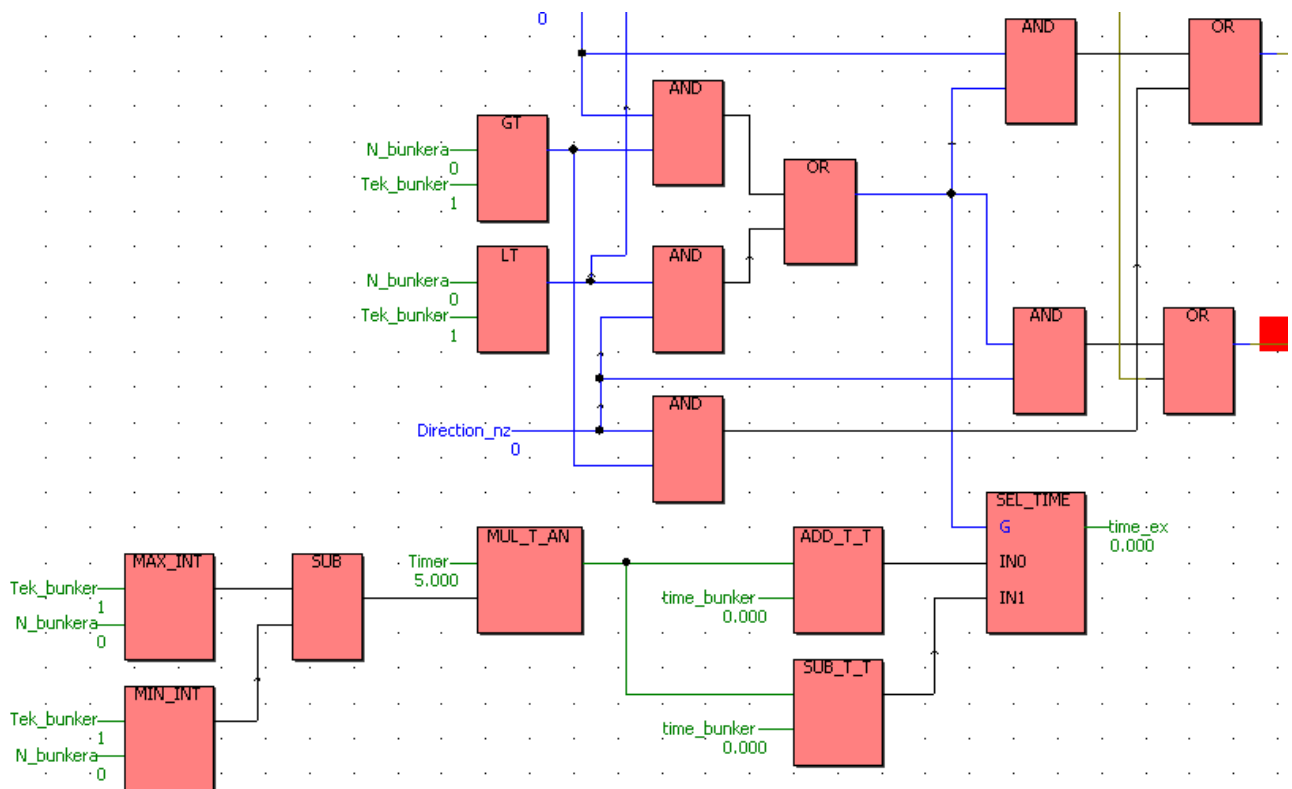
Список оголошених змінних, при створенні функціонального блока користувача, наведений у наступній таблиці:

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

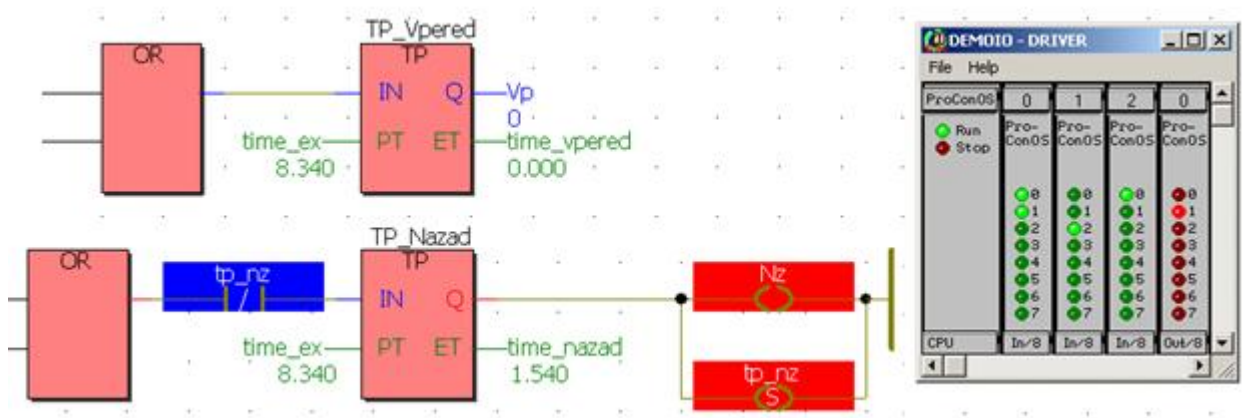
Якщо автостела, при переході з автоматичного на ручне керування, рухалася від одного бункера до іншого, то вона зупиняється. У цьому випадку час її руху до необхідного бункера у ручному режимі визначається, виходячи з поточного напрямку руху автостели, часу, що пройшов з моменту початку її руху і до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



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

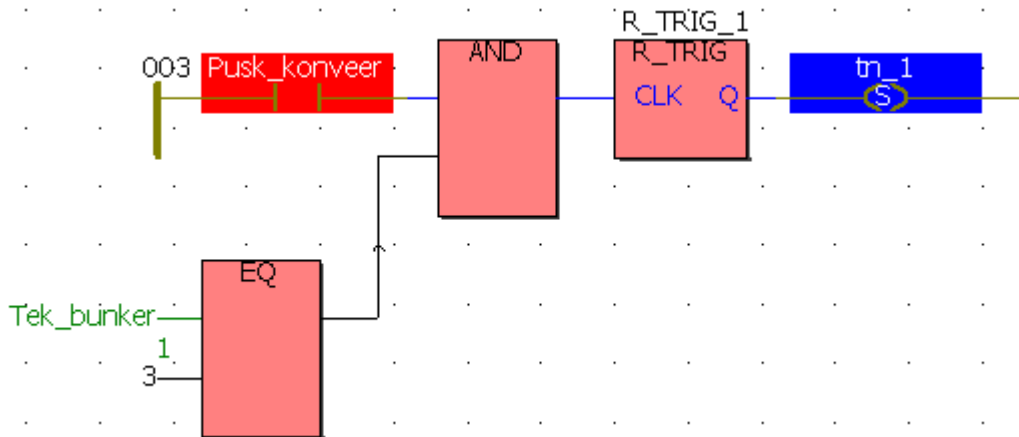


Промодельюємо ситуацію, коли під час пересування від 3-го бункера до 2-го, оператору знадобилося завантажити 1-ий бункер. При переході з автоматичного режиму в ручний (1-ий вхід 0-го модуля вводу), автостела зупинилася. Коли оператор натиснув кнопку завантаження 1-го бункера (0-ий вхід 2-го модуля вводу),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



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

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



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

### ***Висновки***

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

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

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

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

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



трьома двигунами використовується 6-канальний модуль з релейними виходами ADAM-5056.

6. Програмний код контролера створений SFC-мовою. Перевірка роботи програми на симуляторі підтвердила відповідність її роботи алгоритму керування автостелою.

### *Література*

1. Котов К.И., Шершевер М.А. Промышленные системы автоматизации металлургических агрегатов, М., «Металлургия», 1980.- 256с.
2. СВ Альтера. Каталог для энергетиков и инженеров по автоматизации.
3. <http://www.electro.ru/>
2. Advantech. ADAM 5510 KW User's Manual.
3. Ніколаєнко А.М. Програмування ПЛК у Softlogic-системі KW MULTIPROG: Навчальний посібник.-Запоріжжя: Видав.ЗДІА, 2008.-203с.

## ЛІТЕРАТУРА

1. Ніколаєнко А.М. Програмування ПЛК у Softlogic-системі KW MULTIPROG: навч.-посіб. [для студ. вищ. навч. закл.] / Ніколаєнко А.М. – Запоріжжя: Видав. ЗДІА, 2008. – 203с.
2. Фролов С.В. Тенденция развития систем управления технологическими процессами / С.В. Фролов // Приборы и системы. Управление, контроль, диагностика. – 1996 – №9. – с. 6-8.
3. Ельперін І. В. Промислові контролери: навч. посіб. [для студ. вищ. навч. закл.] / Ельперін І. В. – К.: НУХТ, 2003. – 320с.
4. Крутских Р.Ю. SCADA для непрерывных производств – Processlogix / Р.Ю. Крутских // Промышленные АСУ и контроллеры. – 2003 – №7. – с.38-42.
5. Фролов С.В. Тенденции развития программируемых логических контроллеров / С. В. Фролов // Приборы и системы. Управление, контроль, диагностика. – 1996 – №10. – с.17-21.
6. Бердычевский М. И. Конструктивы Евромеханики во встраиваемых системах / М. И. Бердычевский // СТА. – 2002 – №4. – с. 52-59.
7. Пономарев Д. И. Технология мезонинных контроллеров. – настоящее и будущее промышленных систем / Д. И. Пономарев, А.Н. Буданов // Приборы и системы. Управление, контроль, диагностика. – 1999 – №8, с.32-36.
8. Сысоев А.Д. Мезонины: что сегодня? / А. Д. Сысоев // Мир компьютерной автоматизации. – 2001 – №4. – с.19-24.
9. Патрахин В.В. Феномен РАС/ В. В Патрахин // Мир автоматизации. – 2008 – №2. – с.20-23.
10. Токхайм Р. Микропроцессоры: Курс и упражнения. / Роджер Токхайм. – М.: Энергоатомиздат, 1988.– 336 с.
11. Громов Д.И. Классификация современных контроллеров / Д. И. Громов // Контроавт: Контроль и автоматика. Методичка. – 2007 – №4. – с.14-23.
12. Микропроцессорные автоматические системы регулирования. Основы теории и элементы: уч. пос. [для студ. высш. уч. зав.] / Солодовников В. В., Коньков В. Г., Суханов В. А., Шевяков О. В. – М.: Высш. Шк. 1991 – 255с.
13. Андреев Е. Б. Технические средства систем управления технологическими процессами нефтегазовой промышленности: уч. пос. [для студ. высш. уч.зав.] / Е. Б.Андреев, В. Е. Попадько – Тюмень: ТНГИ. – 2007 – 270с. – Режим доступа до навчального посібника: <http://twirpx.com/>
14. Кругляк К. Д. Промышленные сети: цели и средства /К. Д. Кругляк// СТА. – 2002– №4.– с. 6-17.
15. Гусев С. П. Краткий экскурс в историю промышленных сетей /С. П. Гусев//СТА. – 2000 – №4. – с. 78-84.
16. Патрахин В. А. Средства программирования РС-совместимых контроллеров /В. А. Патрахин// ПиКАД. – 2003 – №1-2. – с.34-40.
17. Петров И. В. Программируемые контроллеры. Стандартные языки и приёмы прикладного проектирования /Петров И. В.–М.: СОЛОН –Пресс, 2004.– 256 с.

18. Зюбин В. Е. Программирование ПЛК: языки МЭК 61131-3 и возможные Альтернативы /В. Е. Зюбин// Промышленные АСУ и контроллеры. – 2005 – №11.– с. 31-35.
19. ЗЭиМ. Контроллер малоканальный многофункциональный регулирующий микропроцессорный Ремиконт Р-130. Техническое описание и инструкция по эксплуатации. г. Чебоксары. – 1989 – с.230.
20. ЗЭиМ. Контроллеры малоканальные многофункциональные регулирующие микропроцессорные Ремиконт Р-130 с блоком контроллера БК-1М/01. Руководство по эксплуатации. г. Чебоксары. – 2005– с.105.
21. Контроллеры КРОСС-500, ТРАСА-500, Р-130ISa. Руководство по эксплуатации. Система технологического программирования IsaGraf. – 2005 – с.117.
22. Микрол. Контроллер малоканальный многофункциональный микропроцессорный МІК-51. Руководство по эксплуатации. г. Ивано-Франковск.–2006 – 90с.
23. Программирование в пакете АЛЬФА.
24. Гусев С.П. ADAM 5510 как зеркало современных тенденций автоматизации / С. П. Гусев // СТА. – 1998 – №2. – с. 2–9.
25. Advantech. ADAM 5511User’s Manual. – 2000 – 286с.
26. Основы программирования для начинающих. Borland Turbo C++ v.3.1. – Режим доступа до довідкового порадника: <http://www.opita.net/node/69>.
27. Advantech. ADAM 5510 KW User’s Manual.
28. Кузнецов А.П. Устройство связи с объектом: Модули фирмы WAGO / Кузнецов А. П. // СТА. – 1997 – №3. – с.22-27.
29. Модульная система WAGO I/O System 750. – 2005 – 192с.
30. Руководство пользователя по программированию ПЛК в CoDeSys 2.3.- М.: СОЛОН-Пресс. 2004 – 253с.
31. SIMATIC S7 Программируемый контроллер S7-300. Справочное Руководство. – Режим доступа до довідкового порадника: <http://automation-drives.ru/>
32. SIMATIC S7 Компоненты систем комплексной автоматизации. Каталог ST70-2007 – 452с.
33. SIMATIC S7 Программирование с помощью STEP 7 V5.0 Руководство. – 1999–671с. – Режим доступа до довідкового порадника: [http:// automation -drives.ru/](http://automation-drives.ru/)
34. Schneider Electric.Серия программируемых контроллеров Modicon TSX Quantum. –1999–29с. – Режим доступа до довідкового порадника: [http:// asutpforum.spb.ru/](http://asutpforum.spb.ru/)
35. Modicon Inc. Quantum Series Automation. Технические средства. Справочное руководство. – 1994 – 135с. – Режим доступа до довідкового порадника: <http://www.schneider-electric.ua/>
36. Modicon Inc. Платформа автоматизации Modicon Quantum.– 2004– 264с. – Режим доступа до довідкового порадника: <http://www.schneider-electric.ua/>

37. Modicon Inc. Руководство по проектированию и монтажу системы удаленного ввода-вывода RIO. – Режим доступа до довідкового poradnika: <http://www.schneider-electric.ua/>
38. Руководство по проектированию и монтажу сети Modicon Modbus Plus. – Режим доступа до довідкового poradnika: <http://www.schneider-electric.ua/>
39. Schneider Electric. Платформа автоматизации Momentum. – 2004 – 42с. – Режим доступа до довідкового poradnika: <http://www.schneider-electric.ua/>
40. Деменков Н.П. Языки программирования промышленных контроллеров: уч. пос. [для студ. высш. уч. зав.] / Деменков Н. П. –М:Изд-во МГТУ им. Н.Э.Баумана, 2004. – 172 с.
41. Schneider Electric. Unity – Справочное руководство по языкам программирования. RUS.pdf. – Режим доступа до довідкового poradnika: <http://www.schneider-electric.ua/>
42. Omron. Программируемые контроллеры CS1. Руководство по эксплуатации. – Режим доступа до довідкового poradnika: <http://www.proenergo.ru/>
43. Omron. Модули Controller Link. Руководство по эксплуатации.–2001–308с. – Режим доступа до довідкового poradnika: <http://www.proenergo.ru/>
44. Omron. CX-Programmer V5. Вводный курс.–2001 – 47с.– Режим доступа до довідкового poradnika: <http://www.proenergo.ru/>
45. Omron. Программируемые контроллеры CS1. Руководство по программированию. – 2001 – 766с. – Режим доступа до довідкового poradnika: <http://www.proenergo.ru/>
46. Rockwell Automation. Руководство по выбору ControlLogix.–2005–76с. – Режим доступа до довідкового poradnika: <http://vdt-automation.com.ua/>
47. Rockwell Automation. Контроллеры ControlLogix, Руководство пользователя. – 2002 – 435с. – Режим доступа до довідкового poradnika: <http://vdt-automation.com.ua/>
48. Rockwell Automation. Шасси ControlLogix, Инструкция по установке. – 1997– 20с. – Режим доступа до довідкового poradnika: <http://vdt-automation.com.ua/>
49. Rockwell Automation. Источники питания ControlLogix, Инструкция по установке. – 1997– 14с. – Режим доступа до довідкового poradnika: <http://vdt-automation.com.ua/>
50. Rockwell Automation. Дискретные модули ввода-вывода Controllogix, Руководство пользователя. – 1998 – 65с. – Режим доступа до довідкового poradnika: <http://vdt-automation.com.ua/>
51. Rockwell Automation. Модули аналогового ввода-вывода Controllogix, Руководство пользователя. – 2003 – 352с. – Режим доступа до довідкового poradnika: <http://vdt-automation.com.ua/>
52. Rockwell Automation. Система резервирования ControlLogix. Руководство пользователя. – 2003 – 147с. – Режим доступа до довідкового poradnika: <http://vdt-automation.com.ua/>

53. Rockwell Automation. Общая методика для программируемых контроллеров Logix 5000. Руководство по программированию. – 2004 – 577с. – Режим доступа до довідкового порадника: <http://vdt-automation.com.ua/>
54. Rockwell Automation. Руководство по применению RSLogix 5000. – 1999 – 55с. – Режим доступа до довідкового порадника: <http://vdt-automation.com.ua/>
55. Rockwell Automation. RSLogix Emulate. Руководство пользователя. – 1999 – 39с. – Режим доступа до довідкового порадника: <http://vdt-automation.com.ua/>
56. Патрахин В.В. Обзор технических средств компании GE Fanuc для промышленной автоматизации / В. В. Патрахин // Автоматизация производства.– 2009– №2.– с. 61-65.
57. GE Fanuc Automation. Контроллеры и устройства ввода-вывода. Каталог продукции. – 2006 – 266с. – Режим доступа до каталогу: <http://www.technolink.net/ua/>
58. GE Fanuc Automation. Программируемые системы управления PACSystems RX7i/ Руководство по монтажу. – 2003 – 65с. – Режим доступа до довідкового порадника: <http://www.technolink.net/ua/>
59. GE Fanuc. С чего начать. Logic Developer–PLC. Инструментальное программное обеспечение ПЛК. – 202 – 148с. – Режим доступа до довідкового порадника: <http://www.technolink.net/ua/>
60. Коноваленко А.М., Серета Є.Ю., Ткачьева Н.В., студенты, Ніколаєнко А.М., проф. Особливості програмування логічних контролерів LOGO!. // Тези допов. XIII наук.-техніч. конф. студ., маг., асп. і викл. ЗДІА. Частина III Запоріжжя: Видавництво ЗДІА, 2008, с.156.
61. Бармин А. С. Устройства локальной автоматики. Микроконтроллеры /А.С. Бармин // СТА. – 2003 – №4.– с.38-42.
62. Siemens. SIMATIC LOGO! Руководство. – 2003 – 308с. – Режим доступа до довідкового порадника: <http://automation-drives.ru/>
63. Ткачук О.С., Вернідуб А.В., студенты, Ніколаєнко А.М., проф. Програмування і робота з контролером МІК-51.// Тези допов.XIII наук.-техніч. конф. студ., маг., асп. і викл. ЗДІА. Частина III Запоріжжя: Видавництво ЗДІА, 2008, с.136.
64. VIPA art of automation. Manual VIPA Systems 300S SPEED, 2007 – с.290.
65. Кандибка В.В., Колопов А.П., студенты, Ніколаєнко А.М., проф. Особливості програмування і роботи з контролером WAGO 750-841.// Тези доп. XIII наук.-техн. конф. студ., маг, асп. і викл. ЗДІА. Частина III Запоріжжя: Видавництво ЗДІА, 2008, с.143.
66. Машковський Р.А., Рибкін К.О., студенты, Ніколаєнко А.М., проф. Програмування РС-сумісних контролерів мовою «С».// Тези допов. XIII наук.-техніч. конф. студ., маг., асп. і викл. Частина III Запоріжжя: Видавництво ЗДІА, 2008, с.133
67. Балабуха В.Р. студент, Ніколаєнко А.М., проф. Розробка лабораторної роботи по програмуванню та дослідженню контролера Digus виробництва GE

Ганус.// Тези допов. XV наук.-техн. конф. студ., маг., асп. і викл. ЗДІА. Частина III Запоріжжя: Видавництво ЗДІА, 2010, с.121.

68. СВ АЛТЕРА. Каталог для енергетиків і інженерів по автоматизації. – 2008 – 232с. – Режим доступу до довідкового порадника: <http://www.svaltera.ua/>

69. Индасофт. Решения для промышленной автоматизации. Краткий каталог продукции и примеры внедрений. – 2007 – 64с. – Режим доступу до довідкового порадника: <http://www.indsoft.com/ua/>

69. Патрахин В. В. «Если мафию нельзя победить – её надо возглавить» / В.В. Патрахин, В. А. Патрахин // ПиКАД. – 2008 – №3. – с.22-26.

## СЛОВНИК ТЕРМІНІВ І СКОРОЧЕНЬ

|                     |  |
|---------------------|--|
| АСУ ТП              | Автоматизована система управління технологічними процесами   |
| АЛП                 | Арифметично-логічний пристрій  |
| АРМ                 | Автоматизоване робоче місце  |
| АЦП                 | Аналогово-цифровий перетворювач  |
| Вита пара           | Фізичне середовище для передачі інформації, складається з двох ізольованих проводів, звитих між собою (для зменшення впливу зовнішніх електромагнітних полів). Швидкість передачі – до 1 Мбіт/с                |
| Дуплексне коло      | Фізичне коло, яке дозволяє одночасну двоспрямовану передачу даних (схема з чотирма проводами)  |
| Дуплексний протокол | Режим роботи для доточкового сполучення з двома фізичними колами, в яких повідомлення або блоки передачі даних можуть посилатися в обидва напрями одночасно  |
| ДЦП                 | Дискретно-цифровий перетворювач  |
| ЕСППЗП (EEPROM)     | Перепрограмований постійний запам'ятовуючий пристрій з електричним стиранням інформації  |
| ІРПР                | Інтерфейс радіальний паралельний   |
| ІРПС                | Інтерфейс радіальний послідовний   |
| Коаксиальний кабель | Фізичне середовище для передачі інформації, складається з мідної жили, ізоляції, екрану і зовнішньої оболонки. Випускаються два типи коаксиальних кабелів: товстий і тонкий. Швидкість передачі – до 50 Мбіт/с |
| Майстер-станція     | Пристрій (ПЛК з модулями вводу-виводу або робоча станція), який посилає і збирає дані в/з пристроїв, підключених до напівдуплексної мережі “точка-до-багатоточці”  |
| Мережа Remote I/O   | Мережа віддалених ввідів-виводів, що підтримує обмін даними в реальному масштабі часу для управління пристроями  |
| Мережа ControlNet   | Швидкодіюча мережа, яка використовується для передачі критичної за часом інформації  |
| Мережа DeviceNet    | Відкрита комунікаційна мережа нижнього рівня, яка забезпечує підключення польових пристроїв  |
| Мережа MODBUS       | Мережа з послідовним обміном типу провідний/підпорядкований  |

|                                |   |
|--------------------------------|---|
| Мережа PROFIBUS-DP             | Мережа, що адаптована для побудови швидких розподілених систем збору даних і управління з одним провідним вузлом  |
| Мережа ASI                     | Польова шина нижнього рівня, яка забезпечує підключення польових пристроїв  |
| Мережа FIPIO                   | Мережа, яка орієнтована на віддалений ввід-вивід  |
| Мережа DirectNet               | Підтримує передачу даних в режимі провідний/підпорядкований і забезпечує одноранговий зв'язок між процесорами і іншими пристроями   |
| Мережа DH+                     | Підтримує передачу даних і віддалене програмування контролерів на додаток до однорангового зв'язку між іншими процесорами і пристроями  |
| Мережа Ethernet                | Локальна мережа для швидкого рівноправного (однорангова мережа) обміну інформацією між комп'ютерами та іншими пристроями  |
| Мережа I/O Link                | Високошвидкісна мережа забезпечує можливість праці за схемою провідний/підпорядкований для пакетів даних з фіксованими об'ємами через волоконно-оптичну лінію   |
| Модем                          | Пристрій, який модулює цифрову інформацію з ПЛК або ПК в аналоговий сигнал, що транспортується по телефонних лініях, радіохвилях і супутниках, і демодулює аналогові дані в цифрові у місці отримання |
| МП                             | Мікропроцесор   |
| ОЗП (RAM–Random access memory) | Оперативний запам'ятовуючий пристрій (Пам'ять з довільним доступом)   |
| Оптоволоконний кабель          | Фізичне середовище для передачі інформації, складається з скляної (пластикової) жили, оболонки і декількох шарів захисних покриттів. Швидкість передачі – 100 Мбіт/с і вище                           |
| ОП                             | Операційний пристрій  |
| Паралельний порт               | Електричне з'єднання комп'ютера, яке здатне передавати або отримувати два або більше бітів даних одночасно; порт зв'язку, до якого можуть бути підключені такі пристрої, як паралельний принтер       |
| ПК                             | Персональний комп'ютер  |
| ПЗО                            | Пристрій зв'язку з об'єктом   |
| ПЗ                             | Програмне забезпечення  |



|   |  |
|---|--|
| ПЗП (ROM–Read Only Memory)                | Постійний запам'ятовуючий пристрій (Пам'ять тільки для читання)  |
| ПЛК (PLC) (Programmable Logic Controller) | Програмований логічний контролер   |
| Полудуплексне коло                        | Фізичне коло, яке забезпечує передачу даних у будь-якому напрямі, але не одночасно   |
| Полудуплексний протокол                   | Режим роботи для доточкового або багатоточкового сполучення, в якому повідомлення або блоки передачі даних можуть посилатися в обох напрямках але не одночасно                     |
| Послідовний порт                          | Електричне з'єднання на комп'ютері, яке обробляє біти інформації один за одним; порт зв'язку (COM1 або COM2), до якого можуть бути приєднані пристрої типу модему, миші            |
| ППЗП                                      | Перепрограмований постійний запам'ятовуючий пристрій   |
| Протокол                                  | Набір правил для управління форматом і синхронізацією передачі даних поміж пристроями сполучення, у тому числі підтвердження зв'язку, виявлення помилок і відновлення при помилках |
| ПУ  | Пристрій управління  |
| Розміри контролера                        | Ширина, висота, глибина (Ш x В x Г)  |
| РЧ  | Реальний час   |
| СППЗУ (EPROM)                             | Перепрограмований постійний запам'ятовуючий пристрій зі стиранням інформації за допомогою опромінювання чипу ультрафіолетовим або рентгенівським промінням                         |
| СУБД                                      | Система управління базою даних   |
| Термінатор                                | Кінцевий резистор  |
| Формат 3U, 6U                             | Європейський стандарт на геометричні розміри конструктивів і плат контролерів (U=44,45мм)  |
| ЦАП                                       | Цифро-аналоговий перетворювач  |
| ЦДП                                       | Цифро-дискретний перетворювач  |
| ША  | Шина адресна   |
| ШД  | Шина даних   |
| ШУ  | Шина управління  |
| 12 біт, 16 біт                            | Розрядність АЦП у характеристиці аналогових модулів вводу-виводу   |
| АІ/АО                                     | Аналоговий ввід-вивід  |
| АС, VАС                                   | Напруга змінного струму  |

|  |   |
|--|---|
| Bridge (міст)  | Пристрій, що поєднує дві мережі, які використовують однакові методи передачі даних  |
| COM/DCOM<br>(Component Object Model)                 | Модель багатокомпонентних об'єктів, що дозволяє додатку маніпулювати програмними об'єктами.<br>DCOM-Distributed (розподілена) COM   |
| CPU  | Центральний процесор  |
| DC, VDC  | Напруга постійного струму   |
| DDE (Dynamic Data Exchange)                          | Динамічний обмін даними – протокол, створений компанією Microsoft для обміну даними поміж різними Windows –додатками. Цей протокол реалізує взаємозв'язки типу клієнт-сервер поміж двох програм, що одночасно виконуються |
| DF1 Half-Duplex                                      | Асинхронний полудуплексний протокол Allen-Bradley   |
| DF1 Full-Duplex                                      | Дуплексний протокол Allen-Bradley для сполучення “точка – точка”  |
| DI/DO  | Дискретний ввід-вивід   |
| 16DI, 8AO і т. ін.                                   | Модули вводу-виводу на 16 і 8 каналів, відповідно   |
| DIO  | Розподілений ввід-вивід   |
| DCS  | Distributed Control System (Розподілена система управління)   |
| DRAM   | Динамічний оперативний запам'ятовуючий пристрій   |
| FBD  | Мова функціональних блокових діаграм (набір типових програмних модулів)   |
| Flash (Flash Erase EEPROM)                           | Перепрограмований постійний запам'ятовуючий пристрій, в якому електричне стирання вмісту комірок виконується або для всієї мікросхеми, або для певного блока  |
| HMI (Human Machine Interface)                        | Людино – машинний інтерфейс   |
| Host (хост)  | Головний комп'ютер (контролер) мережі   |
| IL   | Мова список інструкцій (створення процедур)   |
| IEC (International Electrotechnical Committee) 61131 | Міжнародний стандарт, який визначає таке поняття, як «програмовані контролери»  |
| I/O (Input/Output)                                   | Ввід-вивід  |
| LD   | Мова крокових діаграм   |
| Master/Slave   | Провідний/підпорядкований   |
| ODBC (Open DataBase Connectivity)                    | Протокол обміну з реляційними базами даних  |

|  |  |
|--|--|
| OLE (Object Linking and Embedding)       | Механізм включення та вбудовування об'єктів  |
| OPC (OLE for Process Control)            | Стандарт взаємодії поміж програмними компонентами SCADA и контролерами   |
| OSI (Open Systems Interconnection)       | Модель взаємозв'язку відкритих систем  |
| PAC (Programmable Automation Controller) | Програмовані автоматизаційні контролери  |
| PCI, CompactPCI                          | Стандарти локальної шини для внутрішньоплатних з'єднань  |
| Point-to-point (PPI)                     | Мережа, в якій забезпечується сполучення тільки між двома вузлами(точка-до-точки)  |
| Point-to-multipoint (MPI)                | Мережа, в якій забезпечується сполучення між головною станцією и багатьма віддаленими вузлами (точка-до-багатоточки)   |
| RIO                                      | Віддалений ввід-вивід  |
| RTU (Remoute Terminal Unit)              | Віддалений термінальний пристрій   |
| RS-485                                   | Високошвидкісний та перешкодостійкий промисловий послідовний інтерфейс, який передає сигнали диференційними перепадами напруги (0,2...8)В у лінії зв'язку довжиною до1км |
| SFC                                      | Мова послідовних функціональних схем (процедури та транзакції)   |
| Softlogic                                | Інструментальні засоби програмування контролерів, які базуються на вимогах міжнародного стандарту IEC 61131-3  |
| SoftPLC                                  | PC-сумісні контролери  |
| ST                                       | Мова структурованого тексту (типу Pascal)  |
| SQL                                      | Структурована мова запитів до СУБД   |
| SRAM                                     | Статичний оперативний запам'ятовуючий пристрій   |
| VME                                      | Стандарт шини, яка об'єднує у контролері мікропроцесор і модулі (32 або 64 розрядна шина з максимальною пропускнуою спроможністю 40 Мбіт/с).                             |