

Грищенко В.О., Степаненко С.П., Грищенко І.Ю.

КОМП'ЮТЕРНО-ІНТЕГРОВАНІ ТЕХНОЛОГІЇ АВТОМАТИЗАЦІЇ

Навчальний посібник

Київ

2025

УДК 681.5:621.3.049.77

Г 82

*Рекомендовано до друку видання рішенням вченої ради Національного університету біоресурсів і природокористування України
(Протокол № 4 від 22 жовтня 2025 р.)*

Рецензенти:

Алієв Е., доктор технічних наук, професор кафедри інжинірингу технічних систем Дніпровського державного аграрно-економічного університету;

Миرونенко В., доктор технічних наук, професор, завідувач відділу агронавігації та автоматизації мобільних процесів Інституту механіки та автоматизації агропромислового виробництва Національної Академії Аграрних наук України;

Горобець В., доктор технічних наук, професор кафедри інженерії енергосистем Національного університету біоресурсів і природокористування.

Г 82 Комп'ютерно-інтегровані технології автоматизації: навчальний посібник / В. Грищенко, С. Степаненко, І. Грищенко. Київ: НУБіП України, 2025. 240 с.

ISBN 978-617-8598-87-7

Навчальний посібник розроблено згідно «Положення про порядок підготовки, оформлення та видання навчальної літератури в НУБіП України» затвердженого вченою радою НУБіП України 27.02.2025 р. протокол № 8.

Зміст навчального посібника відповідає навчальній програмі дисципліни «Комп'ютерно-інтегровані технології», «Інформаційні комп'ютерні технології для інженерії систем керування» та «Інтелектуальні системи керування електротехнічними комплексами».

Посібник буде корисний студентам, аспірантам та викладачам закладів вищої освіти.

© Грищенко В., Степаненко С., Грищенко І., 2025

ВІДОМОСТІ ПРО АВТОРІВ



Грищенко Володимир Олександрович

Кандидат технічних наук, доцент кафедри автоматичної та робототехнічних систем ім. акад. І.І. Мартиненка Національного університету біоресурсів і природокористування. Викладає дисципліни «Автоматизований облік енерго- і матеріальних ресурсів»; «Комп'ютерно-інтегровані технології»; «Ідентифікація та

моделювання технологічних об'єктів»; «Особливості моделювання та ідентифікація біотехнічних об'єктів». Наукові інтереси пов'язані з моделюванням динамічних режимів типових технологічних об'єктів агропромислового виробництва. Автор 79 наукових праць, з яких 2 монографії, 1 навчальний посібник. Електронна адреса: vlgr@nubip.edu.ua



Степаненко Сергій Петрович

Доктор технічних наук, старший науковий співробітник, завідувач відділу перспективних технологій і технічних засобів для збирання, обробки та зберігання врожаю зернових і олійних культур Інститут механіки та автоматичної агропромислового виробництва Національної академії аграрних наук України. Наукові інтереси пов'язані з

механізації, автоматизації, роботизації технологічних процесів в галузі післязбиральної обробки зерна. Автор 414 наукових праць, з яких 6 монографій, 15 рекомендацій, 1 підручник, 2 навчальні посібники, 48 патентів. Електронна адреса: stepanenko_s@ukr.net



Грищенко Ірина Юріївна

Кандидат технічних наук, доцент навчальної геометрії, комп'ютерної графіки та дизайну Національного університету біоресурсів і природокористування. Викладає дисципліни «Комп'ютерна графіка», «Дизайн і ергономіка», «Інженерна графіка». Наукові інтереси пов'язані з геометричним моделюванням об'єктів. Автор 63 наукових праць, з яких 1 монографії, 1 підручник. Електронна адреса: hryshchenko@nubip.edu.ua

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1. ПРОГРАМОВАНІ ЛОГІЧНІ КОНТРОЛЕРИ. ЗАГАЛЬНИЙ ОГЛЯД	9
1.1. Програмовані логічні контролери.	9
1.2. Архітектура ПЛК.....	12
1.3. Секція вводу/виводу.	15
1.4. Способи підключення пристроїв до ПЛК.....	27
1.5. Типові схеми підключення пристроїв до ПЛК.	28
Висновки за розділом.....	32
Питання для самоперевірки.	33
РОЗДІЛ 2. МОВИ ПРОГРАМУВАННЯ ПЛК.....	34
2.1. Схеми релейно-контактної логіки (Ladder Diagrams).	34
2.1.1. Логічна операція «І» (AND).....	38
2.2.2. Логічна операція «АБО» (OR).	40
2.2.3. Логічна операція «НІ» (NOT).	41
2.2.4. Логічна операція NAND.....	41
2.2.5. Логічна операція NOR.	43
2.2.6. Логічна операція XOR.....	44
2.2.7. Фіксація (Latching).....	45
2.2.8. Використання декількох виходів.....	46
2.2.9. Приклади програмування.....	48
2.2. Список інструкцій (Instruction Lists).	53
2.2.1. Програми схем релейно-контактної логіки та списку інструкцій.	55
2.2.2. Розгалужені та багаторядкові структури.....	58
2.2.3. Приклади програмування.....	60
2.3. Структурований текст (Structured Text).....	61
2.3.1. Умовні оператори.....	63
2.3.2. Оператори циклу.	64
2.3.3. Оголошення типів, змінних та функціональних блоків у мові структурованого тексту.	65
2.3.4. Приклади програмування.....	67
2.4. Функціональні блокові діаграми (Function Block Diagram).....	71
2.4.1. Логічні елементи.	72
2.4.2. Зв'язок та порядок обробки елементів.	73
2.4.3. Приклади програмування.....	74
2.5. Послідовні функціональні схеми (Sequential Function Charts).	75
2.5.1. Розгалуження і об'єднання.	79
2.5.2. Дії.....	80
2.5.3. Приклади програмування.....	83
2.6. Безперервна функціональні схеми (Continuous Function Chart).....	84
2.6.1. Базові елементи мови.....	86
2.6.2. З'єднання елементів та порядок їх обробки.....	88
2.6.3. Приклади програмування.....	89
Висновки за розділом.....	92

Питання для самоперевірки.	93
РОЗДІЛ 3. ПРОЕКТУВАННЯ СИСТЕМ АВТОМАТИЗАЦІЇ	94
3.1. Розробка програми.	94
3.2. Заходи безпеки при проектуванні систем автоматизації.	97
3.2.1. Системи з ПЛК та безпека.	99
3.2.2. Реле аварійної зупинки.....	100
3.2.3. Функції безпеки.....	102
3.2.4. Контролери безпеки.....	103
3.3. Введення в ПЛК експлуатацію.	104
3.4. Пошук несправностей.....	108
Висновки за розділом.....	111
Питання для самоперевірки.	112
РОЗДІЛ 4. ЕЛЕКТРОПНЕВМАТИЧНІ СИСТЕМИ АВТОМАТИЗАЦІЇ	113
4.1. Пристрої для отримання та транспортування стисненого повітря.	114
4.1.1. Компресори.....	114
4.1.2. Ресивери (повітряні баки).	115
4.1.3. Підготовка повітря.....	117
4.1.4. Регулятори тиску.....	119
4.2. Пристрої пневматики та електропневматики.....	122
4.2.1. Пневматичні циліндри.....	122
4.2.2. Клапани.	126
4.2.3. Соленоїди та електромагнітні клапани.....	128
Висновки за розділом.....	130
Питання для самоперевірки.	130
РОЗДІЛ 5. ПРОГРАМУВАННЯ ПРИСТРОЇВ ПЛАВНОГО ПУСКУ ТА ПЕРЕТВОРЮВАЧІВ ЧАСТОТИ	131
5.1. Пристрої плавного пуску.....	131
5.2. Перетворювачі частоти серії Altivar Machine 320.	137
5.2.1. Апаратні та функціональні можливості перетворювачів частоти серії ATV320.....	137
5.2.2. Програмування ATV Logic в перетворювачі частоти серії ATV320.	141
5.2.3. Функціональні блоки ATV Logic в перетворювачі частоти серії ATV320.....	143
5.2.4. Приклад застосування ATV Logic в перетворювачі частоти серії ATV320.....	153
5.3. Перетворювачі частоти Altivar Process 630.....	154
5.3.1. Апаратні та функціональні можливості перетворювачів частоти серії ATV630.....	154
5.3.2. Налаштування функцій перетворювачів частоти серії ATV630.....	159
5.4. Перетворювачі частоти Altivar Process 930.....	167
5.4.1. Апаратні та функціональні можливості перетворювачів частоти серії ATV930.....	167
5.4.2. Налаштування функцій перетворювачів частоти серії ATV930.....	171
5.5.3 Приклад використання MultiDrive Link.....	174
Висновки за розділом.....	187

Питання для самоперевірки.	187
РОЗДІЛ 6. КЕРУВАННЯ ПРОЦЕСАМИ, МЕРЕЖЕВІ СИСТЕМИ ТА SCADA-СИСТЕМИ.....	188
6.1. Типи процесів.	188
6.2. Структура систем керування.....	191
6.3. Керування за принципом «Вмикання/Вимикання».	194
6.4. Керування типу ПІД.....	196
6.5. Керування рухом.	200
6.6. Передача даних.....	202
6.6.1. Послідовна передача даних (Serial Communication).....	209
6.6.2. EtherNet/IP.....	210
6.6.3. Modbus.....	211
6.6.4. Польова шина (Fieldbus).....	212
6.6.5. CANOpen.....	213
6.7. СКАДА.	214
Висновки за розділом.....	216
Питання для самоперевірки.	217
РОЗДІЛ 7. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ АВТОМАТИЗОВАНОГО ПРОЄКТУВАННЯ ЕЛЕКТРИЧНИХ СХЕМ ТА СИСТЕМ АВТОМАТИЗАЦІЇ	218
7.1. Інтерфейс користувача в SEE Electrical.	219
7.2. Базові операції в SEE Electrical.....	228
7.3. Створення нового проекту в SEE Electrical.....	238
Висновки за розділом.....	239
Питання для самоперевірки.	239
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	240

ВСТУП

Сучасне виробництво неможливо уявити без автоматизованих систем керування, які забезпечують ефективність, точність і надійність технологічних процесів. Автоматизація стала невід'ємною складовою індустріального розвитку, сприяючи зниженню енергоспоживання, підвищенню продуктивності та безпеки праці. Основою цих систем є програмовані логічні контролери (ПЛК), перетворювачі частоти, пристрої плавного пуску та комп'ютерні засоби моніторингу і керування.

Програмовані логічні контролери займають центральне місце у системах автоматизації завдяки своїй гнучкості, надійності та простоті у зміні алгоритмів роботи. На відміну від традиційних релейно-контактних схем, ПЛК дозволяють швидко вносити зміни до програм керування без суттєвого втручання у фізичну структуру обладнання. Завдяки використанню стандартних мов програмування, визначених міжнародним стандартом ІЕС 61131-3, можливим стало створення уніфікованих рішень, зрозумілих фахівцям різних спеціальностей.

Важливою складовою сучасних систем керування є перетворювачі частоти, які забезпечують плавне регулювання швидкості обертання електродвигунів, оптимізацію енергоспоживання та підвищення терміну служби обладнання. Перетворювачі серій Altivar Machine та Altivar Process це приклад сучасних інтелектуальних пристроїв, здатних не лише забезпечувати керування двигунами, а й виконувати діагностику, зберігати параметри, обмінюватися даними через мережеві протоколи та інтегруватися у комплексні системи автоматизації.

У структурі промислових систем автоматизації значне місце посідають також електропневматичні компоненти, що поєднують електричне керування з пневматичним приводом. Вони широко використовуються у машинобудуванні, харчовій промисловості, сільському господарстві та в інших галузях, де необхідна швидкодія, точність позиціонування та довговічність механізмів.

Проектування, налагодження та експлуатація автоматизованих систем вимагають глибоких знань не лише з електротехніки, а й з комп'ютерних технологій, системного аналізу, мікропроцесорної техніки та програмування. Тому сучасна підготовка інженера-автоматника передбачає комплексне вивчення як теоретичних основ, так і практичних аспектів створення систем керування.

Опанування методів програмування ПЛК, налаштування перетворювачів частоти, побудови схем керування та роботи з системами SCADA дає змогу майбутнім фахівцям ефективно впроваджувати технології автоматизації у різних галузях виробництва. Застосування програмних засобів для автоматизованого проектування електричних схем (зокрема, SEE Electrical) спрощує розробку технічної документації, скорочує час проектування і зменшує ймовірність помилок.

Навчальний посібник має на меті формування у студентів комплексного уявлення про структуру, принципи роботи та способи програмування сучасних систем автоматизації. Матеріал подано у логічній послідовності, від базових

понять до прикладних рішень. В основі викладу лежить поєднання теоретичних положень із практичними прикладами, що дає змогу краще зрозуміти принципи функціонування систем та освоїти навички їхнього налаштування.

Посібник орієнтований на студентів технічних спеціальностей, зокрема з галузей автоматизації, електротехніки та комп'ютерно-інтегрованих технологій, а також може бути корисним інженерам-практикам, які займаються проектуванням і впровадженням систем автоматичного керування.

Автори висловлюють глибоку подяку докторам технічних наук Е. Алієву, В. Мироненку та В. Горобцю, за велику працю з рецензування навчального посібника і цінні зауваження, які враховано при доопрацюванні.

РОЗДІЛ 1. ПРОГРАМОВАНІ ЛОГІЧНІ КОНТРОЛЕРИ. ЗАГАЛЬНИЙ ОГЛЯД

1.1. Програмовані логічні контролери.

На даний час для керування технологічними процесами застосовують програмовані логічні контролери (рис. 1.1). Програмований логічний контролер (ПЛК) або Programmable Logic Controller (PLC) – це промисловий клас комп'ютера, який можна запрограмувати на виконання функцій керування технологічним процесом [1, 5, 6, 7, 8, 13, 17].

ПЛК замінили звичайні релейні схеми керування, що забезпечило:

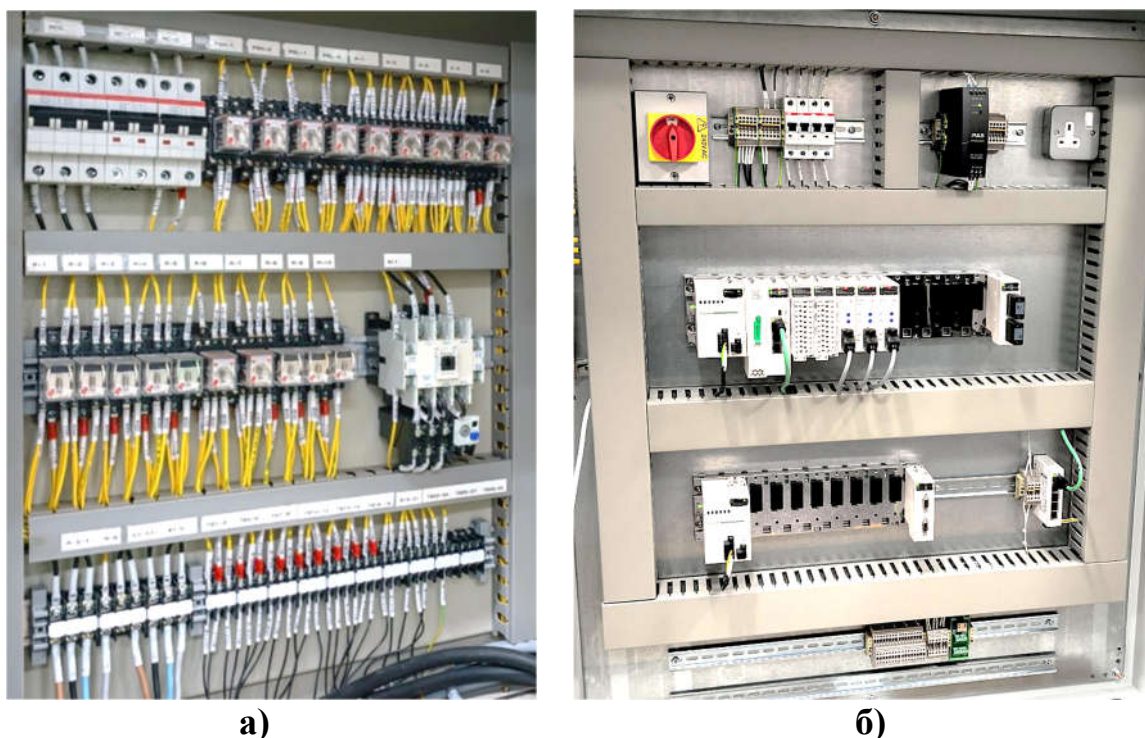
- спрощення монтажу систем автоматики;
- високу швидкість керування;
- високу надійність (ПЛК проходять випробування та отримують відповідні сертифікати);
- легке програмування, тестування та усунення несправностей.



Рис. 1.1. Програмовані логічні контролери Modicon M221 (а) та M580 (б) від компанії Schneider Electric

ПЛК можна назвати цифровим пристроєм розробленим для керування технологічним процесом в реальному часі в залежності від зовнішніх керуючих впливів, але на відміну від персонального комп'ютера він пристосований для використання в промисловому середовищі (нечутливий до: вібрацій, електричних шумів, ударів, збільшений діапазон робочих температур), обладнаний спеціальними інтерфейсами введення/виведення і програмується спеціальними мовами програмування. Програми зберігаються в енергонезалежній пам'яті та/або обладнані акумуляторними батареями. ПЛК можна назвати системами реального часу, бо вихід системи керування відповідає певним умовам на вході. Структура ПЛК базується на тих самих принципах, що і архітектура комп'ютера, вона здатна не тільки виконувати завдання перемикання реле, але й виконувати інші програми, такі як синхронізація, підрахунок, обчислення, порівняння і обробка даних аналогових сигналів.

Використання ПЛК в порівнянні з реле керування надає значні переваги (рис. 1.2): немає жорстких зв'язків, які неможливо змінити без демонтажу всієї системи; невисока ціна в порівнянні з еквівалентними релейними схемами; невеликий розмір.



а) б)
Рис. 1.2. Схеми систем керування релейна (а) та з використанням ПЛК (б)

В сучасних системах керування релейні елементи вже не використовують в якості елементів логіки та мають багато інших переваг:

Підвищена надійність. Після того, як програма була написана і перевірена, її можна легко завантажити до інших ПЛК. Оскільки вся логіка міститься в пам'яті ПЛК, немає шансів зробити помилку при підключенні (рис. 1.3). Програма зменшує кількість необхідного монтажу в порівнянні з релейними схемами але залишається необхідність в монтажі релейних елементів для зв'язку з силовими пристроями. Використання твердотільних елементів в ПЛК також підвищує надійність.

Більша гнучкість. Простіше створити і замінити програму в ПЛК, ніж провести монтаж та переналаштування схеми. Взаємозв'язки між входами та виходами визначаються програмою, а не послідовністю підключення (рис. 1.4). Програмне забезпечення, яке може містити захист (блокування паролем або апаратним ключем) може оновлюватись виробником шляхом завантаження та перепрограмування ПЛК.

Нижча вартість. Спочатку ПЛК були розроблені для заміни логіки побудованої на релейних елементах, але економія коштів була такою, що використання реле стає недоцільним, за винятком керування силовими пристроями.



Рис. 1.3. Програма знаходиться в пам'яті ПЛК

Можливість зв'язку. PLC може зв'язатись з іншими контролерами, панелями оператора або комп'ютерним обладнанням для виконання таких функцій, як диспетчерський контроль, збір даних, моніторинг пристроїв та параметрів процесу, а також завантаження та вивантаження програм (рис. 1.5).

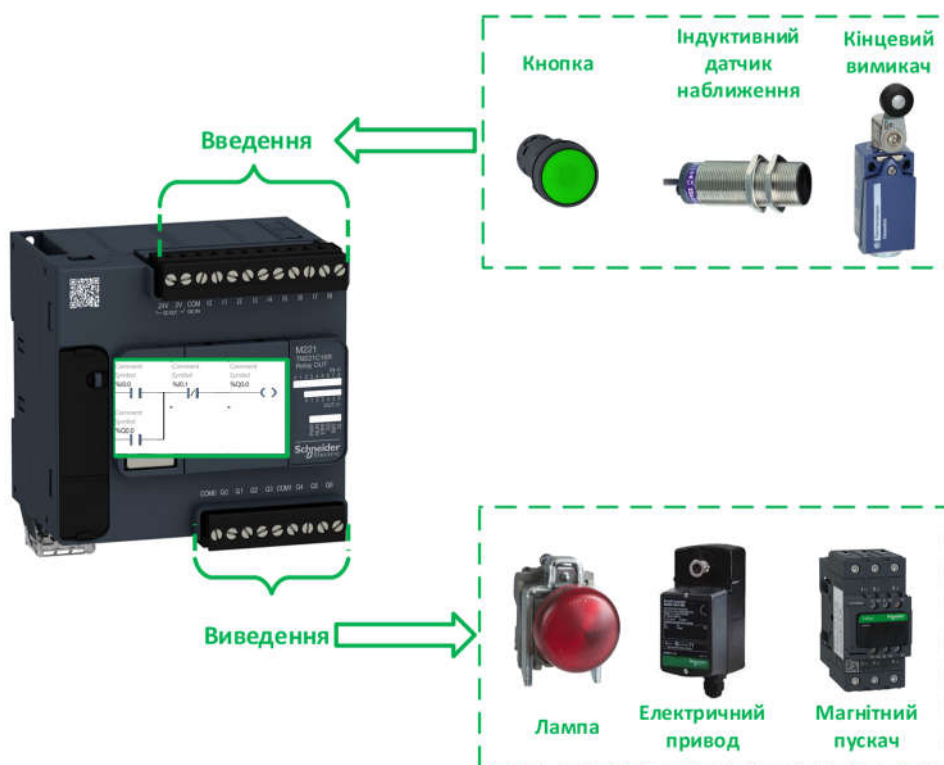


Рис. 1.4. Залежність між вводами та виводами які визначаються в користувацькій програмі

Зменшений час відгуку. ПЛК призначені для високошвидкісних і реальних додатків (рис. 1.6). Програмований контролер працює в режимі реального часу, що означає, що дія, яка відбувається в даний час, призведе до виконання операції або зупинки. Машини, які обробляють тисячі елементів в секунду і об'єкти, які

проходять лише частку секунди перед датчиком, потребують використання ПЛК з швидкодіючими входами і виходами.

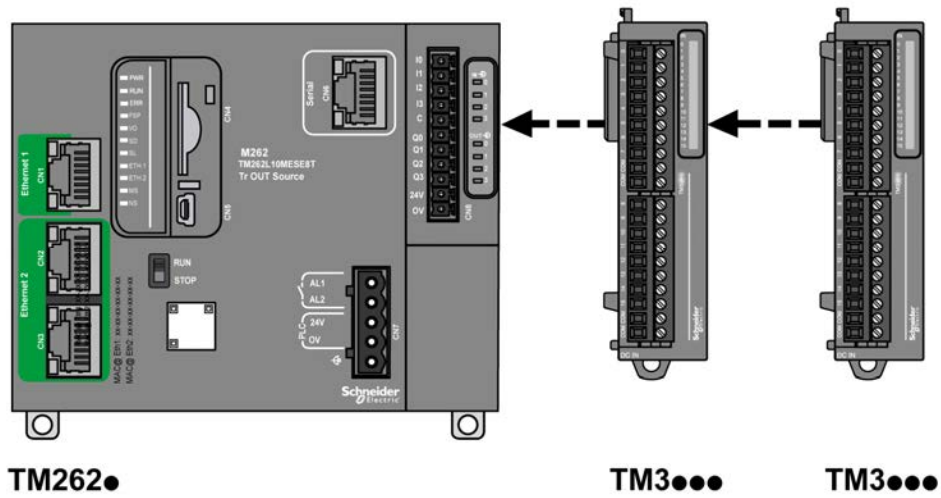


Рис. 1.5. ПЛК M262 з модулями розширення TM3



Рис. 1.6. Використання швидкісного лічильника

Простіше усунути неполадки. ПЛК мають вбудовані функції симуляції та діагностики, які дозволяють користувачам легко відстежувати та виправляти проблеми програмного забезпечення та апаратних засобів. Для пошуку та виправлення цих проблем користувачі можуть відображати керуючу програму на моніторі та дивитися її в реальному часі, коли вона виконується (рис. 1.7).

1.2. Архітектура ПЛК.

Термін «архітектура» може відноситися до апаратного забезпечення ПЛК, до програмного забезпечення ПЛК або до їх комбінації. Відкрита архітектура дозволяє легко підключати систему до пристроїв і програм інших виробників. Відкриті архітектури використовують готові компоненти, які відповідають затвердженим стандартам. Система із закритою архітектурою – це система, яка має власну розробку, що ускладнює її підключення до інших систем.

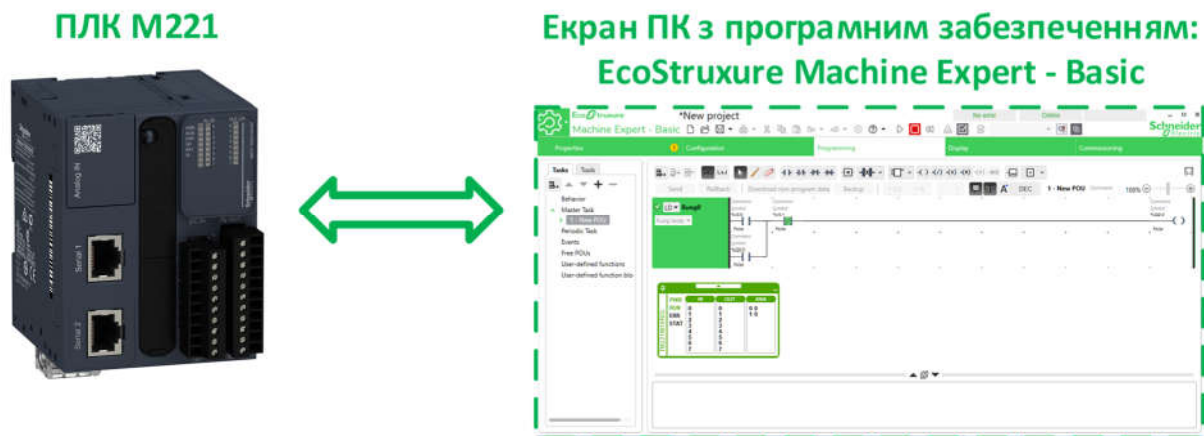


Рис. 1.7. Виконання програми в ПЛК можна перевіряти на ПК в реальному часі

Більшість систем ПЛК є запатентованими, тому будь-яке загальне обладнання або програмне забезпечення, яке використовується, сумісне тільки з конкретним ПЛК. Крім того, хоча основні концепції однакові у всіх методах програмування, можуть бути невеликі відмінності в адресації, розподілі пам'яті, пошуку та обробці даних для різних моделей. Отже, програми ПЛК різних виробників не можуть бути взаємозамінними.

Зазвичай система ПЛК має наступні типові функціональні вузли: процесор, пам'ять, інтерфейс введення/виведення, інтерфейс зв'язку, пристрій програмування, блок живлення (рис. 1.8).

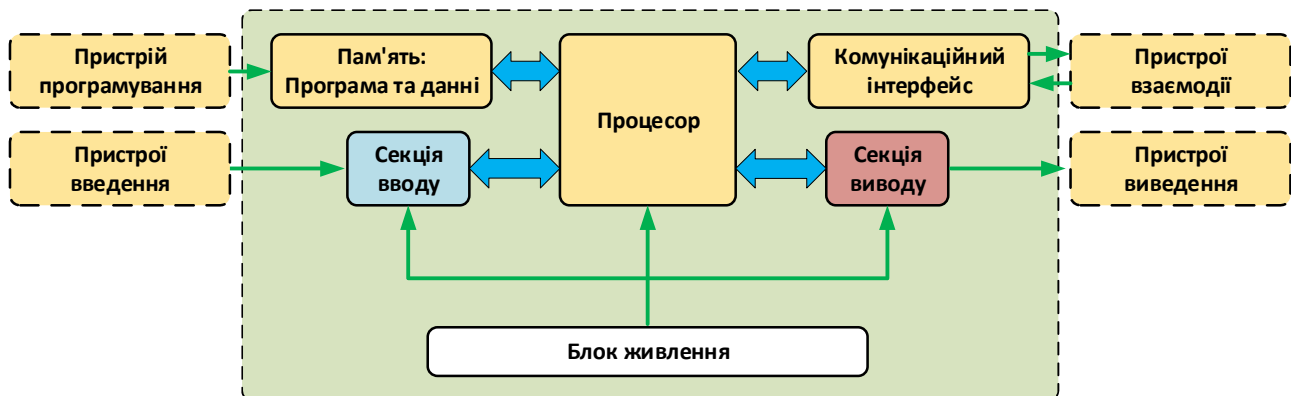


Рис. 1.8. Будова ПЛК

Складовими елементами є:

Процесор або *центральний процесор* (CPU) є блок, що містить мікропроцесор. Він інтерпретує вхідні сигнали і виконує керуючі дії відповідно до програми, що зберігається в його пам'яті, передаючи рішення як сигнали дії на виходи. Цей блок містить велику кількість окремих реле, лічильників, таймерів і даних одиниць зберігання, однак, вони не існують фізично в ПЛК, бо моделюються програмно.

Пам'ять, це місце де зберігається програмний код, який містить керуючі дії для виконання мікропроцесором та дані для обробки станів зі входів/виходів, зазвичай використовується EPROM або EEPROM та RAM.

Пристрій програмування використовується для введення необхідної програми в пам'ять процесора. Програма розробляється в пристрої, а потім переноситься в блок пам'яті ПЛК.

Секція Введення/Виведення це форма інтерфейсу завдяки якій процесор отримує інформацію від зовнішніх пристроїв і передає інформацію зовнішнім пристроям. Наприклад входи можуть отримувати інформацію від кнопок, фотоелементів, датчиків температури, датчиків потоку та інш. До виходів можуть бути підключені реле, магнітні пускачі, електромагнітні клапани, лампи та інш. Пристрої вводу та виводу можна класифікувати на дискретні, цифрові та аналогові (рис. 1.9). Пристрої, що подають *дискретні* або *цифрові сигнали* – це пристрої, в яких сигнали мають два стани: вимкнений та увімкнений (FALSE або TRUE). Таким чином, перемикач – це пристрій, що подає дискретний сигнал, або відсутність напруги, або напругу. *Цифрові пристрої* можна розглядати по суті, як дискретні пристрої, що видають послідовність сигналів увімкнення/вимкнення. *Аналогові пристрої* дають сигнали, розмір яких пропорційний розміру змінної, що контролюється. Наприклад, датчик температури може видавати напругу, пропорційну температурі.

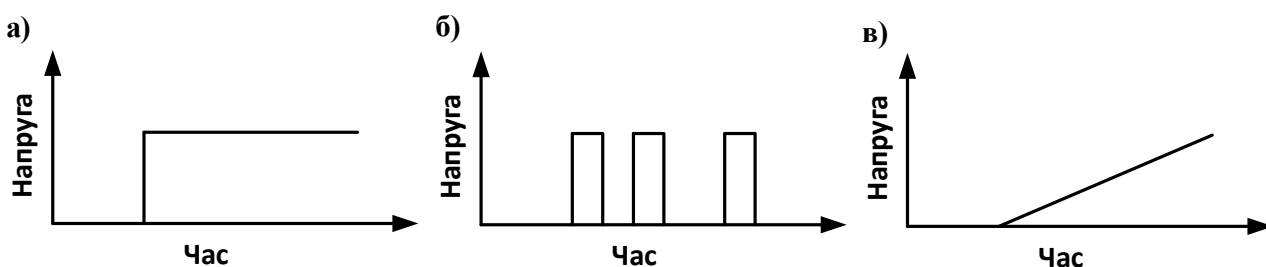


Рис. 1.9. Типи сигналів: (а) дискретні, (б) цифрові та (в) аналогові

Існує два способи апаратної реалізації секції введення/виведення в ПЛК: фіксований та модульний. *Фіксоване введення/виведення* (рис. 1.1а) є типовим для невеликих ПЛК, які поставляються в одному корпусі без окремих знімних блоків. Процесор і входи/виходи упаковані разом, а клеми вводу/виводу мають фіксовану кількість вбудованих з'єднань для входів і виходів. Основною перевагою такого типу апаратної реалізації є нижча вартість. Кількість доступних точок вводу/виводу варіюється і зазвичай її можна розширити, придбавши додаткові блоки фіксованого введення/виведення. Недоліком фіксованого введення/виведення є його недостатня гнучкість; ви обмежені в тому, що можете отримати в тих кількостях і типах, які продиктовані апаратною реалізацією. Крім того, для деяких моделей, якщо будь-яка частина пристрою виходить з ладу, доводиться замінювати весь пристрій, замінити весь пристрій.

Модульне введення/виведення (рис. 1.1б) є апаратною реалізацією ПЛК при якій він розділений на відсіки в які можна вставляти окремі модулі. Ця функція значно розширює апаратні можливості та підвищує гнучкість пристрою. Вона дозволяє вибрати один з модулів, доступних від виробника та комбінувати їх на свій розсуд. Базовий модульний контролер складається зі стійки, блоку живлення, процесорного модуля (CPU), модулів вводу/виводу (I/O), що вставляються в стійку та операторського інтерфейсу для програмування та

моніторингу. Коли модуль вставляється в стійку, він встановлює електричне з'єднання з серією контактів, які називаються об'єднувальною платою, розташованою в задній частині стійки. Процесор ПЛК також підключений до задньої панелі та може обмінюватися даними з усіма модулями в стійці.

Комунікаційний інтерфейс використовується для прийому і передачі даних по комунікаційних мережах від або до інших віддалених ПЛК, НМІ та інш. Він відповідає за такі дії, як перевірка пристрою, збір даних, синхронізація між користувацькими програмами та керування з'єднанням.

Блок живлення необхідний для перетворення мережевої змінної напруги в низьку постійну (за стандартом для ринку України 24 В) необхідну для роботи процесора та ланцюгів інтерфейсу модулів вводу-виводу.

1.3. Секція вводу/виводу.

Секція вводу/виводу (вводу/виводу) є основою будь-якої реалізації ПЛК, слугуючи критично важливим інтерфейсом між контролером і фізичним світом. Без належного функціонування секцій вводу/виводу навіть найскладніші алгоритми керування залишаються відірваними від промислових процесів, якими вони покликані керувати. Вона забезпечує необхідну схему для взаємодії системи ПЛК із зовнішніми пристроями, встановлюючи з'єднання через канали вводу/виводу з польовими пристроями, такими як датчики, перемикачі, двигуни, електромагніти та інші виконавчі елементи. Кожна система керування потребує такого дво-направленого потоку інформації – входів, що надають інформацію про стан процесів центральному процесору, та виходів, що передають робочі сигнали від центрального процесора до пристроїв, якими він керує.

Кожна точка входу і виходу в системі ПЛК має унікальну адресу, до якої може отримати доступ центральний процесор. Цю схему адресації можна уявити як ряд будинків уздовж дороги; наприклад, адреса 10 може бути призначена для входу від певного датчика наближення, тоді як адреса 45 може бути призначена для виходу, що керує певним магнітним пускачем двигуна.

Система адресації дозволяє ПЛК організувати інформацію про всі підключені пристрої та отримати доступ до неї. Як показано на рис. 1.10, адресація надає процесору систематизований метод визначення місцезнаходження кожного пристрою для ефективного моніторингу або керування ним.

Сучасні ПЛК реалізують декілька підходів до адресації, зокрема:

Адресація на основі стійок/слотів – ця система ідентифікує пристрої на основі їх фізичного розташування в архітектурі керування. Адреса зазвичай включає:

- ідентифікатор типу (вхід або вихід);
- номери стійок і слотів, що вказують на фізичне розташування модуля;
- позначення слів та бітів, що ідентифікують конкретне клемне з'єднання.

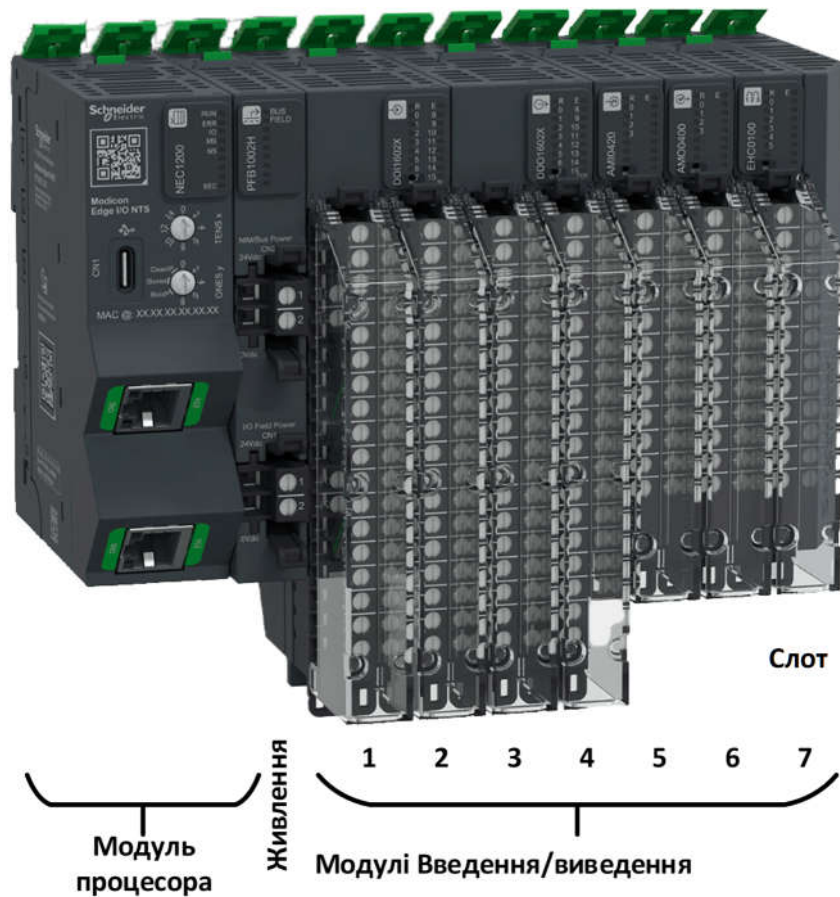


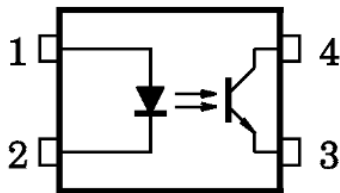
Рис. 1.10. Промисловий контролер Modicon Edge

Адресація на основі тегів – застосовується в нових системах, вона використовує алфавітно-цифрові імена (теги), а не фіксовані числові формати. Ім'я тегу саме по собі ідентифікує дані, забезпечуючи більш інтуїтивно зрозуміле програмування та можливості пошуку та усунення несправностей.

Адресація на базі ПК – використовується в програмних ПЛК, які імітують функції ПЛК на персональних або промислових комп'ютерах. Ці системи використовують плати вводу/виводу в поєднанні з ПК як інтерфейс для польових пристроїв.

Важливим аспектом систем вводу/виводу є електрична ізоляція між внутрішньою схемою ПЛК і зовнішніми польовими пристроями. Ця ізоляція захищає чутливі електронні компоненти контролера від потенційно небезпечних напруг і струмів, присутніх у промисловому середовищі.

Ізоляція зазвичай досягається за допомогою оптоізоляторів (також званих оптронами). На рис. 1.11 показано принцип роботи оптоізолятора, який складається зі світлодіода та фототранзистора. Коли цифровий імпульс проходить через світлодіод, він виробляє імпульс інфрачервоного випромінювання. Цей імпульс виявляється фототранзистором, генеруючи напругу в приймальному ланцюзі. Фізичний проміжок між світлодіодом і фототранзистором забезпечує електричну ізоляцію, дозволяючи при цьому передавати цифрові сигнали між ланцюгами. Внутрішні схеми керування ПЛК зазвичай працюють при напрузі 5 В постійного струму або менше.



**Рис. 1.11. Оптрон: 1 – анод; 2 – катод;
3 – емітер; 4 – колектор**

Формування сигналу у вхідних каналах дозволяє ПЛК приймати широкий спектр вхідних сигналів, перетворюючи їх у напругу, сумісну з вимогами мікропроцесора. Залежно від моделі та можливостей ПЛК, входи можуть підтримувати різні рівні напруги, такі як 5 В, 24 В, 110 В і 240 В для цифрових (дискретних) сигналів. Великі ПЛК зазвичай пропонують більше варіантів входів, тоді як менші моделі можуть бути обмежені одним типом входу, наприклад 24 В.

Спрощена схема для одного дискретного входу змінної напруги 240 В модуля введення наведено на рис. 1.12. Принцип роботи схеми полягає в наступному:

- фільтр вхідного шуму, що складається з конденсатора і резисторів R1 і R2, видаляє помилкові сигнали, які викликані відскоком контакту або електричними перешкодами;
- коли кнопка натиснута, 240 В змінного струму подається на вхід мостового випрямляча;
- це призводить до низького рівня вихідної напруги постійного струму, яка подається на світлодіод оптичного ізолятора;
- номінальна напруга стабілітрона (Z_D) встановлює мінімальний пороговий рівень напруги, який можна виявити;
- коли світло від світлодіода потрапляє на фототранзистор, він переходить у провідний стан, і логічний стан кнопки передається процесору;
- оптрон не тільки відокремлює високу вхідну напругу змінного струму від логічних схем, але також запобігає пошкодженню процесора через перехідні процеси мережевої напруги. Крім того, ця ізоляція також допомагає зменшити вплив електричного шуму та електричних перешкод, які можуть спричинити нестабільну роботу процесора;
- для діагностики несправностей є світлодіодний індикатор стану входу, який світиться, коли кнопка входу замкнута. Цей індикатор може бути підключений по обидва боки оптрона;
- вхідний модуль типу АС/DC використовується для обох типів входів для входів змінного і постійного струму, оскільки полярність входу не має значення;
- вхідний модуль ПЛК має або всі входи ізолювані один від одного без спільних вхідних з'єднань, або групи входів, які мають спільне з'єднання.

Для *аналогових* сигналів додаткові процеси перетворення перетворюють безперервні фізичні вимірювання (наприклад, температуру, тиск, витрату) в цифрові значення, які ПЛК може обробити, як правило, за допомогою аналого-

цифрових перетворювачів, інтегрованих в спеціалізовані модулі вводу/виводу, які підтримують сигнали за напругою 0...10 В та/або за струмом 0...20 мА.

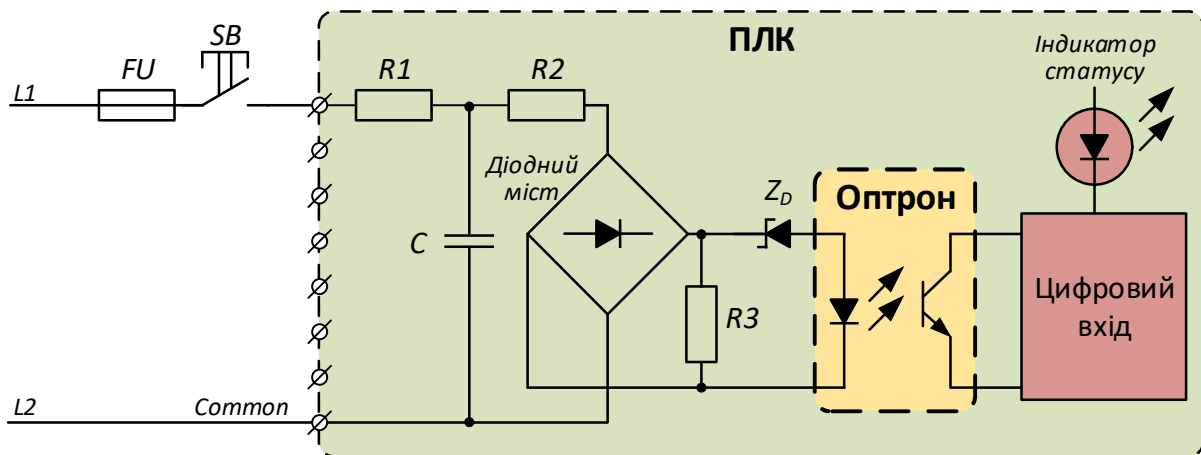


Рис. 1.12. Спрощена схема для одного входу за дискретним входом змінного струму

Вхідні інтерфейси приймають сигнали від машин або технологічних пристроїв і перетворюють їх у сигнали, придатні для використання контролером. Вхідні схеми повинні відповідати різним типам датчиків і характеристикам сигналів, а також забезпечувати їх передачу:

- захист від стрибків напруги та електричних перешкод;
- індикацію стану за допомогою світлодіодів для полегшення пошуку та усунення несправностей;
- фільтрацію для запобігання помилкових спрацьовувань через перехідні сигнали;
- сумісність зі стандартними виходами датчиків.

Вхідні модулі доступні в різних конфігураціях, включаючи вхід постійного струму (з виходом або без нього), вхід змінного струму, аналоговий вхід (напруга або струм), а також спеціальні модулі для конкретних типів датчиків, таких як термомпари або резистивні датчики температури.

Вихідні інтерфейси перетворюють сигнали контролера на зовнішні сигнали, які використовуються для керування машинами або процесами. У системах ПЛК зазвичай використовуються три основні типи вихідних каналів (рис 1.13):

- релейний вихід;
- транзисторний вихід;
- симісторний вихід.

Релейні виходи використовують електромеханічні реле для перемикання струмів у зовнішніх ланцюгах. Основні характеристики включають:

- здатність перемикати струми в кілька ампер;
- сумісність з навантаженнями як змінного, так і постійного струму;
- відмінна ізоляція між ПЛК і зовнішніми ланцюгами;
- стійкість до імпульсних струмів і перехідних процесів напруги;
- відносно повільна реакція перемикання (зазвичай 5...15 мс);
- механічний знос, що обмежує термін служби.

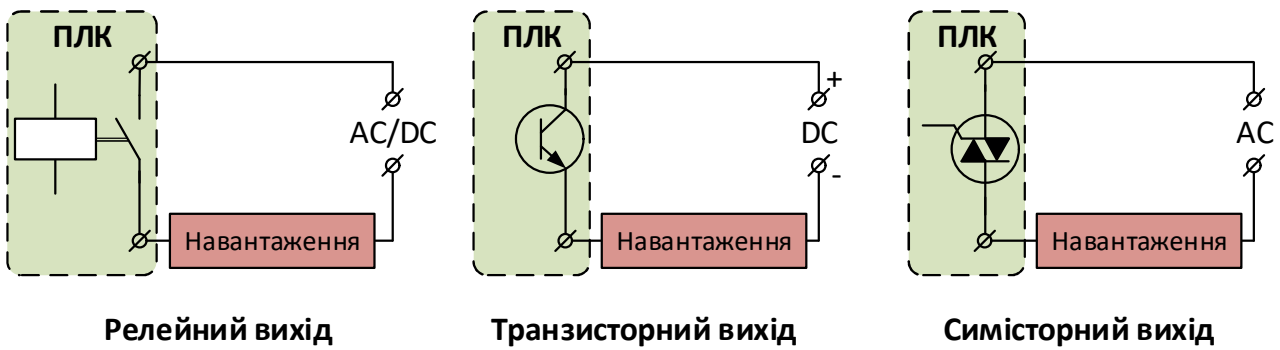


Рис. 1.13. Вихідні інтерфейси ПЛК

Релейні виходи добре підходять для застосувань, що вимагають надійної ізоляції, перемикачів змішаної напруги або там, де швидкість перемикачів не є критичною. Але максимальне струмове навантаження для кожного модуля виходів визначено в технічній специфікації та зазвичай складає до 1...2 А. Для захисту ланцюгів вихідних модулів не слід перевищувати вказані номінальні значення струму. Тому для керування більшими навантаженнями, наприклад, потужними магнітними пускачами, до вихідного модуля підключається стандартне реле керування. Контакти реле можна використовувати для керування більшим навантаженням або магнітним пускачем, як показано на рис. 1.14. Коли реле керування використовується таким чином, воно називається проміжним реле.

Транзисторні виходи використовують напівпровідникові пристрої для перемикачів передбачають:

- значно швидше перемикачів (зазвичай мікросекунди);
- відсутність механічного зносу, що забезпечує довший термін служби;
- виключно перемикачів тільки за постійним струмом;
- вразливість до перенапруги та високої зворотної напруги;
- необхідність використання захисту за допомогою запобіжників або вбудованих електронних схем;
- електричну ізоляцію на основі оптрона.



Рис. 1.14. Використання проміжного реле

Ці виходи ідеально підходять для високошвидкісних застосувань, керування широтно-імпульсною модуляцією (ШІМ) та ситуацій, що вимагають частих перемикань.

У *симісторних виходах* використовуються напівпровідникові пристрої, спеціально розроблені для комутації змінного струму:

- суворо обмежені для роботи зі змінним струмом;
- відсутність рухомих частин, що забезпечує тривалий термін служби;
- висока чутливість до пошкоджень від перевантаження по струму;
- практично універсальне до застосування захисних запобіжників;
- ізоляція на основі оптрона від схеми ПЛК.

Симісторні виходи особливо добре підходять для керування навантаженнями змінного струму, такими як обігрівачі, освітлення та певні типи двигунів, де перевага надається твердотільній комутації.

Спрощена схема для одного виходу дискретного вихідного модуля змінного струму показана на рис. 1.15. Робота схеми може бути підсумована наступним чином:

- при нормальній роботі цифрові логічні схеми процесора встановлюють стан виходу відповідно до програми;
- коли програма вимагає подачу живлення, на світлодіод оптрону подається напруга;
- світлодіод випромінює світло, яке перемикає фототранзистор у провідний стан;
- це, в свою чергу, запускає напівпровідниковий симістор змінного струму, який переходить у провідний стан, дозволяючи струму текти до вихідного навантаження;
- оскільки симістор проводить струм в обох напрямках, на виході до навантаження подається змінний струм;
- симістор, замість того, щоб мати стан увімкненого та вимкненого, насправді має низький і високий рівні опору, відповідно. У вимкненому стані (високий опір) невеликий струм витoku в кілька міліампер все ще протікає через симістор;
- як і у випадку вхідних ланцюгів, вихідний інтерфейс зазвичай обладнаний світлодіодами, які показують стан кожного виходу;
- для вихідного модуля зазвичай потрібні запобіжники, і вони є для кожного ланцюга, що дозволяє захищати і експлуатувати кожен ланцюг окремо. Деякі модулі також мають візуальні індикатори стану запобіжників;
- симістор не можна використовувати для перемикання навантаження постійного струму.

Аналогові входи представляють фізичні величини, які можуть мати нескінченну кількість значень. Типові аналогові входи і виходи варіюються 0...20 мА, 4...20 мА або 0...10 В. На рис.1.16 показано, як модулі аналогових входів і виходів ПЛК використовуються для вимірювання та відображення рівня рідини в резервуарі. Модуль інтерфейсу аналогового входу містить схему, необхідну для прийому аналогового сигналу напруги або струму від польового

пристрою датчика рівня. Цей вхід перетворюється з аналогового в цифрове значення для використання процесором. Схема модуля аналогового виходу приймає цифрове значення від процесора і перетворює його назад в аналоговий сигнал, який керує польовим вимірювачем рівня в резервуарі.

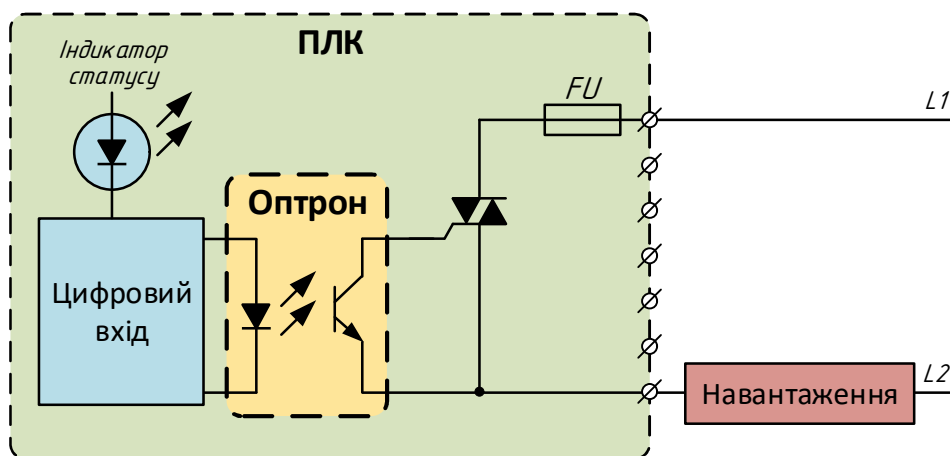


Рис. 1.15. Спрощена схема одного дискретного виходу змінного струму

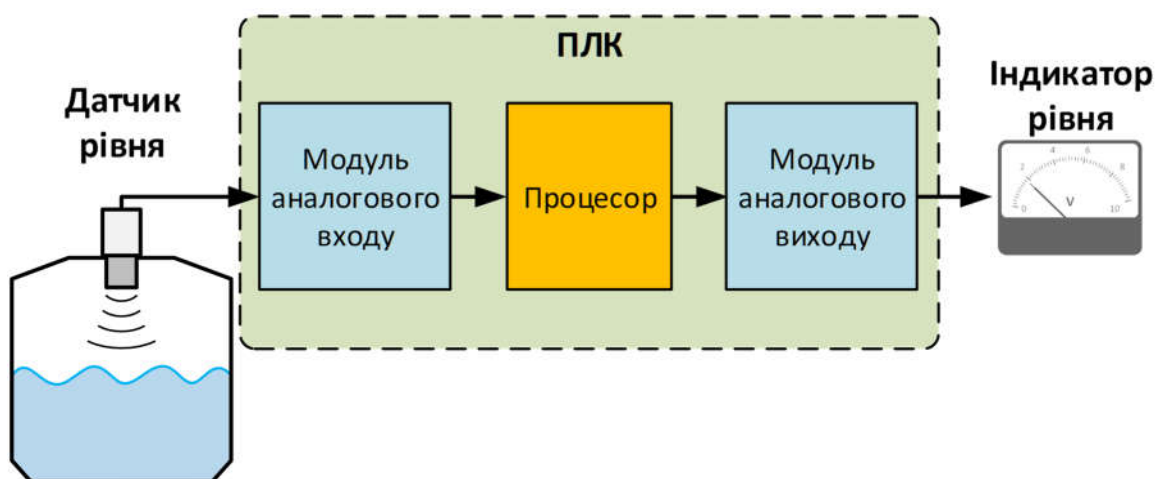


Рис. 1.16. Аналоговий вхід та вихід ПЛК

Модулі аналогового вводу зазвичай мають декілька вхідних каналів, які дозволяють підключати до ПЛК 2...16 пристроїв. Два основних типи модулів аналогового вводу – це датчики напруги та датчики струму. Аналогові датчики вимірюють різні фізичні величини в певному діапазоні та генерують відповідний сигнал напруги або струму. До поширених фізичних величин, які вимірює аналоговий модуль ПЛК, належать температура, вологість, швидкість, рівень, витрата, вага, тиск і положення. Наприклад, датчик може вимірювати температуру в діапазоні 0...500 °С і видавати відповідний сигнал напруги, який змінюється 0...50 мВ.

На рис. 1.17 наведено приклад аналогового модуля вхідного сигналу вимірювання напруги, який використовується для вимірювання температури.

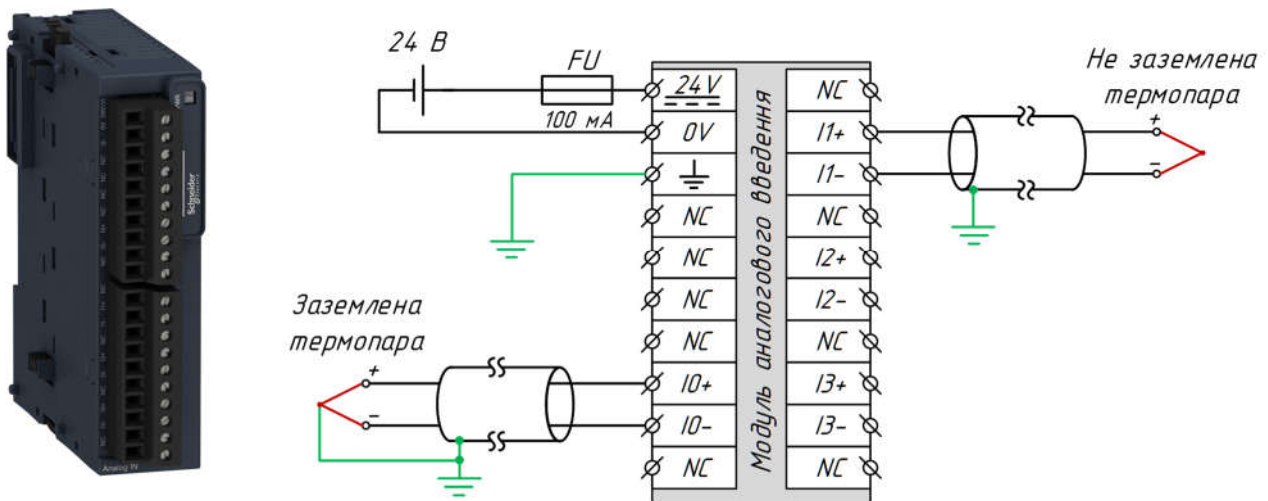


Рис. 1.17. Чотирьох каналний модуль розширення аналогових входів TM3TI4D/G

Схема підключення застосовується до 4-канального аналогового модуля вводу термопари TM3TI4D/G компанії Schneider Electric. Термопара виробляє змінну напругу постійного струму в діапазоні мілівольт, яка пропорційна контрольованій температурі. Ця напруга підсилюється і оцифровується модулем аналогового вводу, а потім надсилається до процесора. Через низький рівень напруги вхідного сигналу, для зменшення небажаних електричних шумів, які можуть бути індуковані в провідниках від іншої проводки, в схемі використовується вита екранована пара. При використанні незаземленої термопари екран повинен бути підключений до заземлення на клеммах модуля. Для отримання точних показань з кожного з каналів необхідно компенсувати температуру між проводом термопари і вхідним каналом. Для цього в клемну колодку вбудовано терморезистор з компенсацією холодного спаю (Cold junction accuracy ± 4.0 °C для модуля TM3TI4D/G).

Перетворення аналогового сигналу в цифрові данні здійснюється за допомогою аналого-цифрового (АЦП) перетворювача, основного елемента модуля аналогового вводу. Модулі аналогового вводу за напругою бувають двох типів: уніполярні та біполярні. Уніполярні модулі можуть приймати вхідний сигнал, який змінюється тільки в позитивному напрямку. Наприклад, якщо зовнішній пристрій видає напругу 0...10 В, то використовуються уніполярні модулі. Біполярні сигнали змінюються між максимальним від'ємним і максимальним додатним значенням. Наприклад, якщо первинний пристрій видає сигнал -10 В...+10 В, використовується біполярний модуль. Роздільна здатність аналогового вхідного каналу відноситься до найменшої зміни значення вхідного сигналу, який може отримати, і базується на кількості бітів, що використовуються в цифровому представленні. Модулі аналогового вводу повинні створювати діапазон цифрових значень між максимальним і мінімальним значенням для представлення аналогового сигналу в усьому його діапазоні.

При підключенні входів для вимірювання напруги важливо чітко дотримуватися визначених вимог щодо довжини проводів, щоб мінімізувати

погіршення сигналу та вплив електромагнітних перешкод, які індукуються вздовж з'єднувальних провідників. Вхідні сигнали струму, які не настільки чутливі до перешкод, як сигнали напруги, зазвичай не мають обмежень щодо відстані. Вхідні модулі для вимірювання струму зазвичай приймають аналогові дані в діапазоні 4...20 мА шляхом пропускання його через опір 250 Ом для отримання вхідного сигналу 1...5 В (рис. 1.18).

Таким чином, наприклад, датчик, який використовується для контролю рівня рідини в діапазоні висот від 0 до 1 м, матиме рівень 0, представлений 4 мА, а рівень 1 м – 20 мА. Використання 4 мА для представлення нижньої межі аналогового діапазону слугує для того, щоб відрізнити, коли датчик показує нуль і коли датчик не працює і з цієї причини дає нульову реакцію. У цьому випадку струм буде дорівнювати 0 мА. Струм 4 мА також часто є достатнім для роботи датчика і тому усуває необхідність в окремому джерелі живлення (рис. 1.19). Для підключення будь-якого типу аналогового вхідного сигналу зазвичай рекомендується використовувати екрановану виту пару.

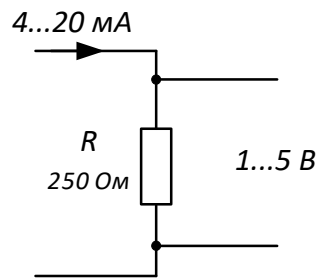


Рис. 1.18. Перетворення аналогового сигналу в ПЛК

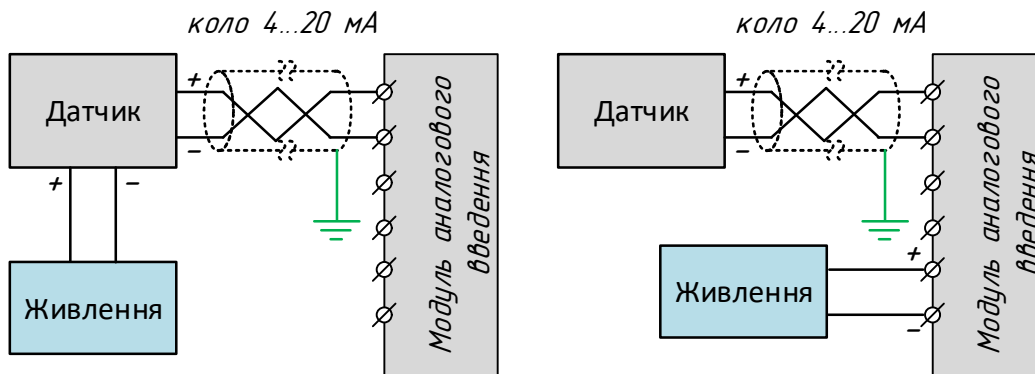


Рис. 1.19. Живлення датчика та модуля аналогового введення

Модуль інтерфейсу *аналогового виходу* отримує від процесора цифрові дані, які перетворюються в пропорційну напругу або струм для керування аналоговим польовим пристроєм.

Перетворення цифрового сигналу в аналогові значення здійснюється за допомогою цифро-аналогового (ЦАП) перетворювача, основного елемента модуля аналогового виходу. *Аналоговий вихідний сигнал* – це безперервний і змінний сигнал, який змінюється під контролем програми ПЛК. До типових пристроїв, якими керує модуль аналогового виходу ПЛК, належать:

регулювальні клапани, самописець, електронні приводи та інші типи керуючих пристроїв, які реагують на аналогові сигнали.

Розглянемо, як приклад використання модулів аналогового вводу/виводу в типовому технологічному процесі контролю кількості рідини, розміщеної в резервуарі, рівень якої регулюється відсотком (кутом) відкриття клапану (рис. 1.20). Аналоговий вихід ПЛК використовується для управління потоком рідини шляхом регулювання величини (кута) відкриття клапану. Спочатку клапан відкритий на 100 відсотків. Коли рівень рідини в резервуарі наближається до заданого значення, процесор змінює вихідний сигнал, який регулює клапан для підтримання заданого значення.

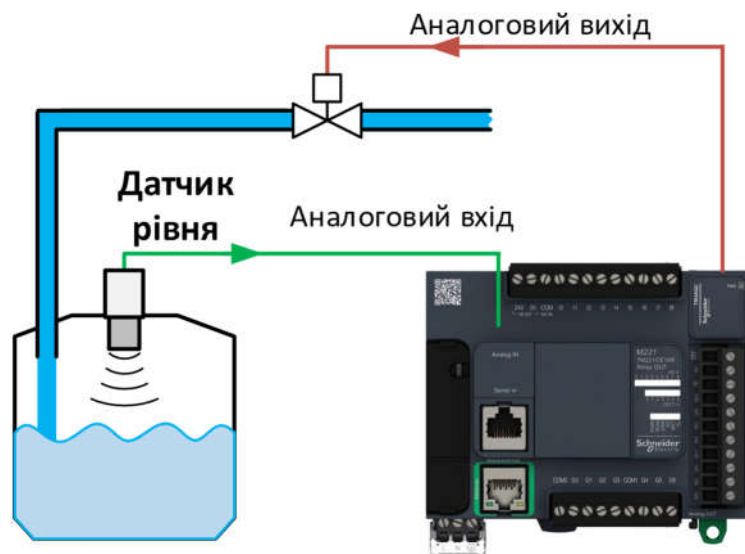


Рис. 1.20. Спрощена схема керування рівнем рідини в резервуарі

Для зменшення напруги з датчика до необхідного рівня (рівень вихідної напруги U_{out} можна використовувати дільник потенціалу (рис. 1.21):

$$U_{out} = \frac{R_2}{R_1 + R_2} U_{in}.$$

Для підвищення рівня напруги можна використовувати підсилювачі. На рис. 1.22 показано основні схеми, які можна використовувати з операційним підсилювачем 741.

В інвертуючому підсилювачі вихідна напруга U_{out} розраховується як:

$$U_{out} = -\frac{R_2}{R_1} U_{in},$$

а з неінвертуючим підсилювачем:

$$U_{out} = \frac{R_1 + R_2}{R_1} U_{in}.$$

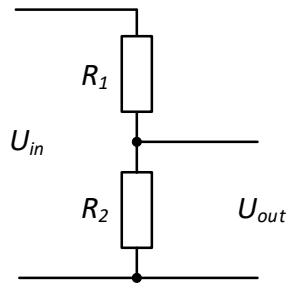


Рис. 1.21. Принципова схема дільника потенціалу

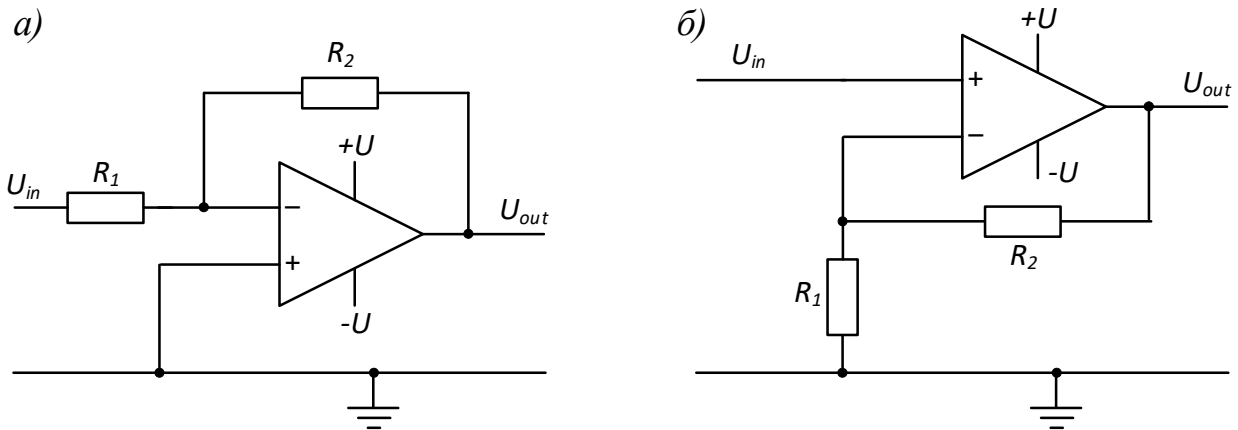


Рис. 1.22. Принципові схеми інвертуючого (а) та неінвертуючого (б) підсилювачів

Часто диференціальний підсилювач потрібен для підсилення різниці між двома вхідними напругами. Це відбувається, коли датчик – наприклад, тензорезистор – підключений до моста Уїтстона, і на виході отримують різницю між двома напругами, або з термопарою, де потрібна різниця напруги між гарячим і холодним спаями. На рис. 1.23 показано спрощену схему операційного підсилювача для цієї мети.

Вихідна напруга U_{out} дорівнює:

$$U_{out} = \frac{R_2}{R_1} (U_2 - U_1).$$

Розглянемо, як приклад підключення тензорезистивного датчика до ПЛК (рис. 1.24). Датчик підключений до моста Уїтстона, а різниця потенціалів, що вийшла з рівноваги, підсилюється диференціальним підсилювачем перед подачею на блок аналого-цифрового перетворювача, який є частиною аналогового вхідного порту ПЛК.

Вихід операційного підсилювача насичується при напрузі близько 12 В, такі значення зазвичай досягаються при напрузі на вході близько 10 мВ. Таким чином, операційні підсилювачі широко використовуються для подачі сигналів увімкнення/вимкнення на основі відносної величини двох вхідних сигналів. Один сигнал підключається до неінвертуючого виводу, а інший – до інвертуючого. Операційний підсилювач визначає, чи є сигнал на інвертуючій клемі вищим або нижчим за сигнал на неінвертуючій клемі, тобто він є

компаратором. Помінявши місцями підключення опорної та вхідної напруги, можна змінити полярність вихідного сигналу. Такі компаратори можна використовувати як основу для систем керування вмиканням/вимиканням. Таким чином, можна одержати опорну напругу і порівнювати з нею напругу з датчика, і таким чином отримувати вихід увімкнення/вимкнення в залежності від того, чи напруга з датчика вища або нижча за опорну напругу.

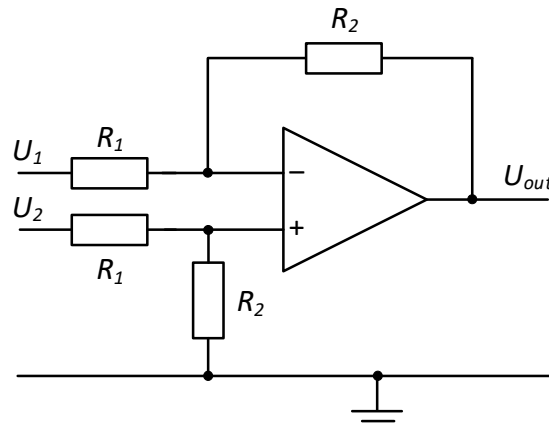


Рис. 1.23. Принципова схема диференціального підсилювача

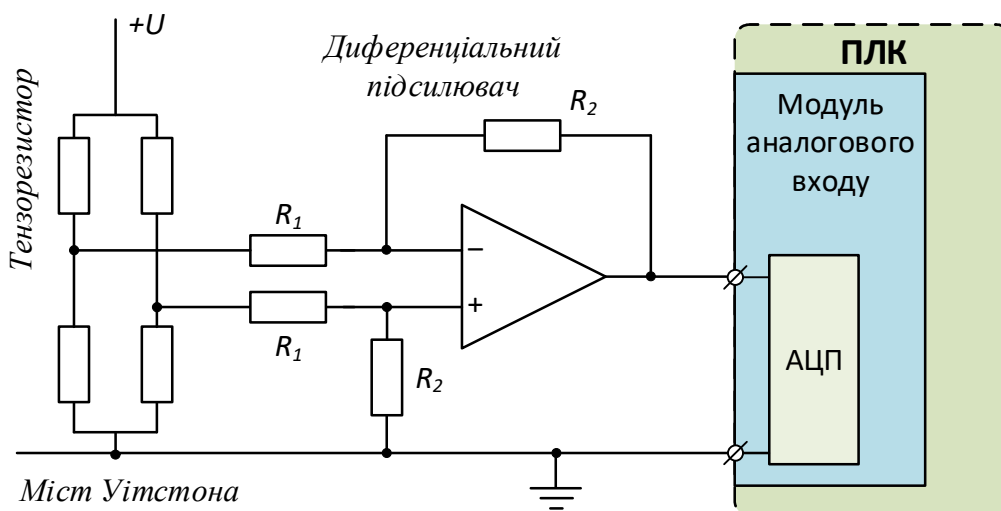


Рис. 1.24. Принципова схема підключення тензорезистора до ПЛК

Котушки, соленоїди та двигуни які приєднано до виходів ПЛК є індуктивними елементами. При роботі цих пристроїв в момент вимикання створюється зворотна ЕРС, яка може бути досить великою. Найпростішим методом захисту від цієї зворотної ЕРС є додавання діода паралельно котушці. Діод підключений таким чином, що струм не може протікати через нього, коли струм живить котушку, але він закорочує котушку і таким чином гасить струм, що виникає від зворотної ЕРС. Такий діод часто називають зворотним діодом.

Деякі вихідні пристрої можуть потребувати послідовних струмообмежувальних резисторів. Наприклад, світлодіодний індикатор зазвичай має максимальний номінальний струм 10...30 мА. При 20 мА падіння напруги на ньому може становити 2.1 В. Таким чином, якщо ми подаємо на нього вхідну

напругу 5 В, через послідовний резистор падає 2.9 В. Це означає, що послідовний опір становить $2.9/0.020=145$ Ом, тому можна використовувати стандартний резистор на 150 Ом. Деякі світлодіоди вже мають вбудовані резистори.

1.4. Способи підключення пристроїв до ПЛК.

Для підключення пристроїв постійного струму (DC) до ПЛК використовуються два способи Sourcing і Sinking.

При Sourcing (джерело), якщо вважати напрям струму від «плюса» до «мінуса», вхідний пристрій отримує струм від вхідного модуля – тобто вхідний модуль є джерелом струму (рис. 1.25а).

При Sinking (споживач), струм тече від вхідного пристрою до модуля, тобто вхідний модуль приймає струм і є його «споживачем» (рис. 1.25б). Якщо струм тече від вихідного модуля до навантаження, вихід вважається Sourcing (рис. 1.26а). Якщо ж струм тече в зворотному напрямку, від навантаження до вихідного модуля то це Sinking (рис. 1.26б).

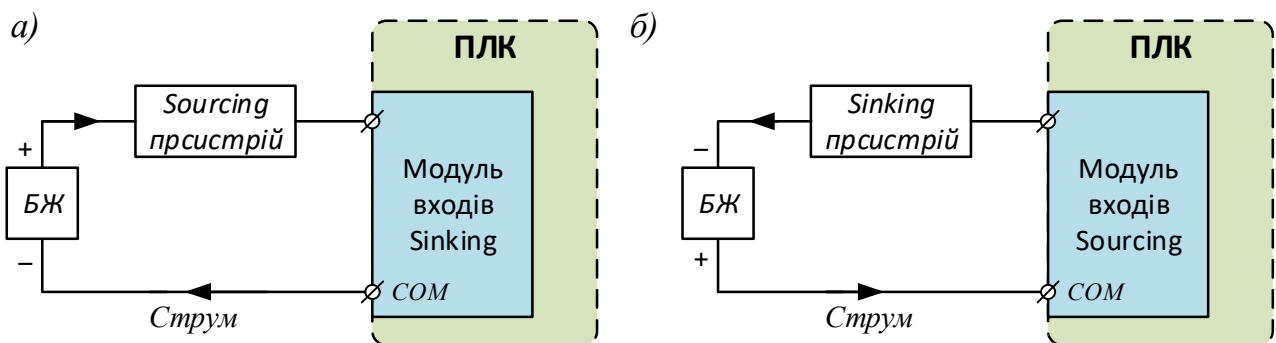


Рис. 1.25. Схеми підключення входів ПЛК для Sinking (а) та Sourcing (б)

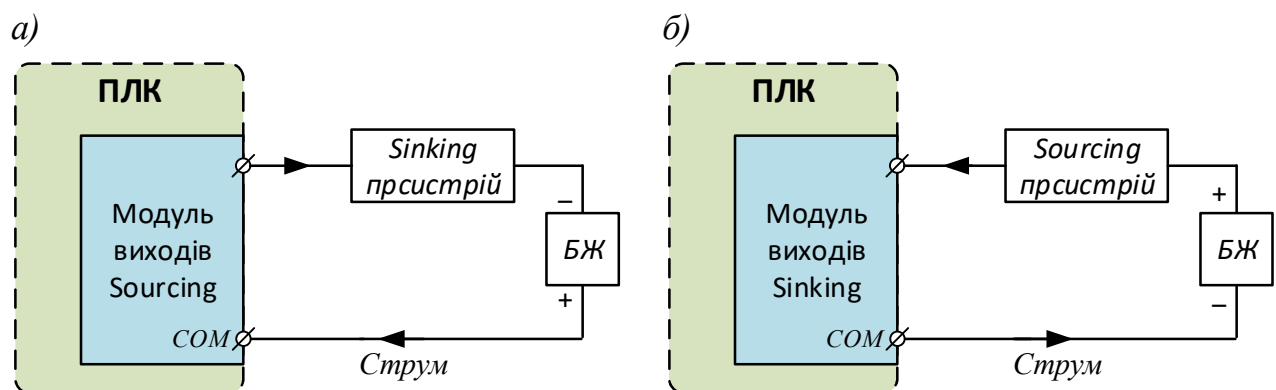


Рис. 1.26. Схеми підключення виходів ПЛК для Sourcing (а) та Sinking (б)

Важливо знати, який саме тип входу або виходу використовується, щоб правильно під'єднати пристрої до ПЛК. Наприклад, датчики з виходом типу Sourcing мають підключатися до входів ПЛК типу Sinking, а датчики з Sinking виходом – до Sourcing входів ПЛК.

1.5. Типові схеми підключення пристроїв до ПЛК.

Специфікації виробників надають інформацію про те, як правильно та безпечно використовувати інтерфейсний пристрій. Ці специфікації накладають певні обмеження не тільки на модуль вводу/виводу, але й на зовнішнє обладнання, з яким він може працювати. Деякі системи ПЛК підтримують гарячу заміну модулів вводу/виводу, призначених для заміни при увімкненому живленні та працюючому ПЛК.

Для дискретних модулів важливо знати, з якою напругою вони працюють. Наприклад, це може бути 24 В постійного струму (рис. 1.27) або 230 В змінного (рис. 1.28). Вхідні модулі зазвичай витримують невеликі відхилення – плюс-мінус 10 % від номіналу. В технічній документації це часто вказано як допустимий діапазон.

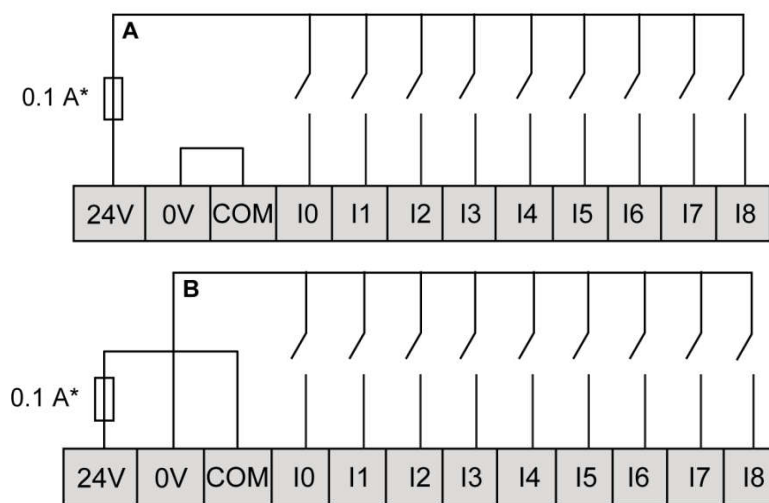


Рис. 1.27. Схема підключення ПЛК TM221C16R з позитивною (А) та негативною (В) логікою

Для спрацьовування входу потрібен певний мінімальний струм. Це захищає від хибних спрацювань через перешкоди або слабкі сигнали. Також важливо знати порогові значення напруги при яких відбувається спрацювання дискретного входу, тобто перехід «Увімкнено» – «Вимкнено» (рис. 1.29).

Умови навколишнього середовища також мають значення. Наприклад, модулі розраховані на роботу лише до певної температури повітря, перевищення якої призводить до пониження продуктивності вбудованих цифрових входів/виходів ПЛК, а значне перевищення може спричинити збої в роботі (рис. 1.30).

При проектуванні системи автоматизації також треба враховувати час реакції. Це затримка між тим, коли на вхід подали сигнал, і тим, коли модуль його «побачив». Вона потрібна, щоб згладжувати шум, наприклад – від дрібних «спрацювань» контактів. Ця затримка залежить від типу входу та його номеру. Для ПЛК TM221C вона становить для входів: I0, I1, I6, I7 – 5 мкс (швидкі входи); I2, I3, I4, I5 – 35 мкс; I8...I23 – 100мкс, з додаванням часу фільтрації сигналу. Для інтелектуального реле Zelio SR2 A121FU затримка становить: 50...255 мс.

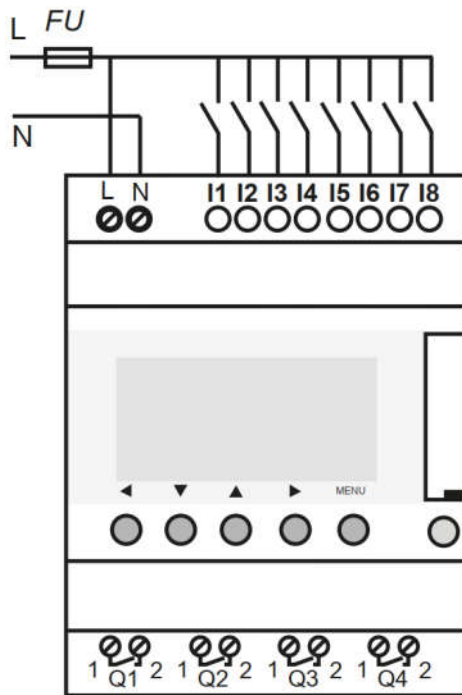


Рис. 1.28. Схема підключення інтелектуального реле Zelio SR2 A121FU

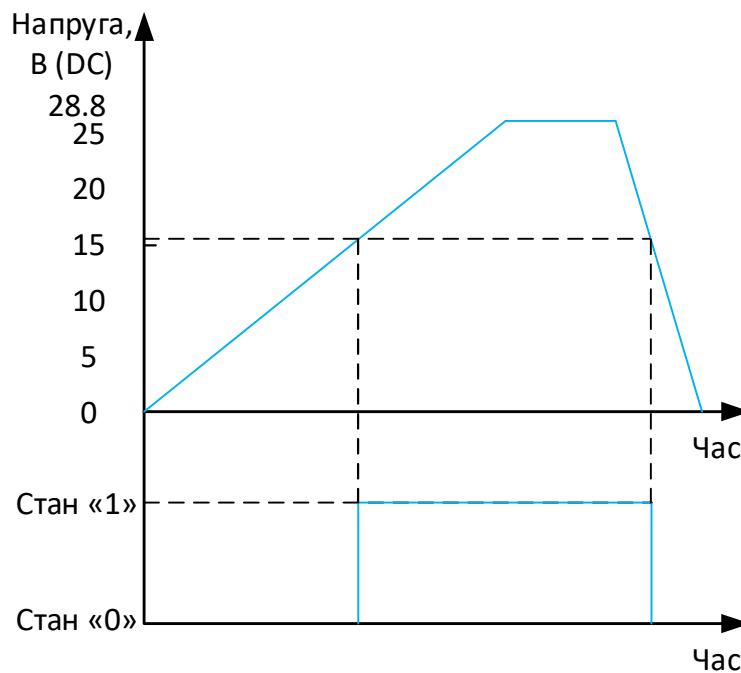
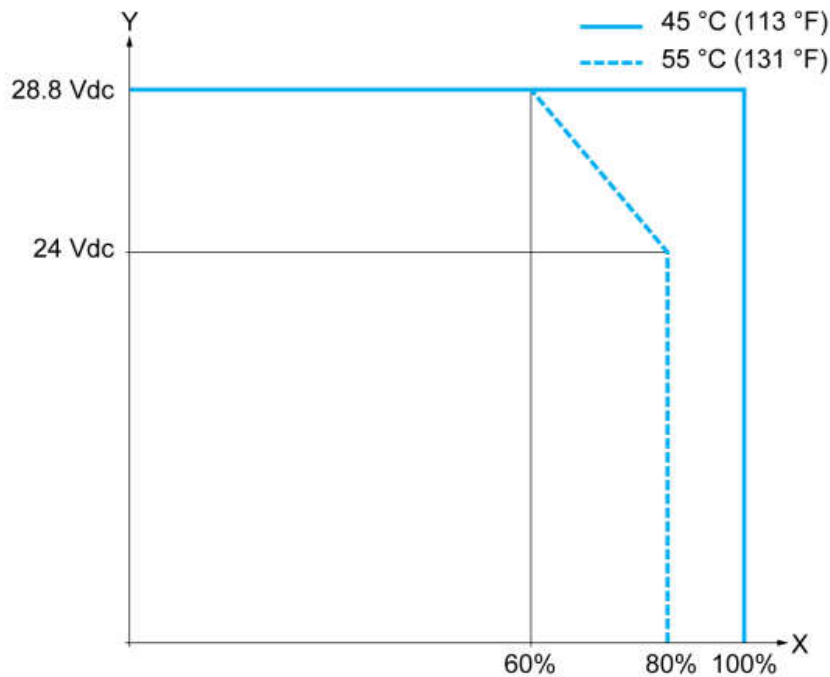


Рис. 1.29. Порогове спрацювання дискретних входів ПЛК ТМ221С

Секція виходів має свої обмеження, які зазначаються в технічній документації та залежать від марки та модифікації ПЛК. До основних вимог до пристроїв які приєднуються до виходів є напруга та струм. Технічні особливості вихідного інтерфейсу залежать переважно від типу його типу (рис. 1.13). Так релейні виходи більш універсальні до напруги (24...250 В для змінної, 5...30 В для постійної) але в свою чергу не забезпечують швидкісне перемикання, та мають свої обмеження за струмом (до 8 А для інтелектуальних реле та до 2 А для

ПЛК). Для запобігання перевантаженню вихідних модулів в коло рекомендують встановлювати запобіжники або інші захисти.



**Рис. 1.30. Криві зниження продуктивності вбудованих цифрових входів ПЛК TM221C без модуля розширення:
X – частка вхідного сигналу; Y – вхідна напруга**

На рис. 1.31 наведено схеми підключень вихідних контактів ПЛК M221C16R до навантаження. При підключенні необхідно враховувати, що клеми COM1 і COM2 не з'єднані між собою. Для збільшення терміну служби контактів і захисту від потенційного пошкодження індуктивних навантажень необхідно підключити діод з вільним ходом паралельно кожному індуктивному навантаженню постійного струму або RC-обмежувач паралельно кожному індуктивному навантаженню змінного струму.

Також існує поняття пускового струму – це короткий сильний струм, який виникає при запуску, скажімо, ламп чи двигунів. Модуль повинен бути розрахований на таке навантаження, навіть якщо воно короткочасне.

Деякі модулі мають захист від коротких замикань – він може бути як для кожного виходу окремо, так і для групи виходів. А іноді, навіть після вимкнення виходу, невеликий струм усе ще протікає. Це нормальне явище для твердотільних елементів і зазвичай не заважає, але на це потрібно зважати при підключенні чутливих пристроїв.

Існує також поняття електричної ізоляції – щоб захистити внутрішню логіку ПЛК від випадкового високого сигналу ззовні (наприклад короткого замикання або статичного розряду на датчик). Наприклад, номінальна імпульсна витримувана напруга для інтелектуального реле Zelio SR2 A121FU до 4 кВ відповідно до МЕК 60947-1 та МЕК 60664-1.

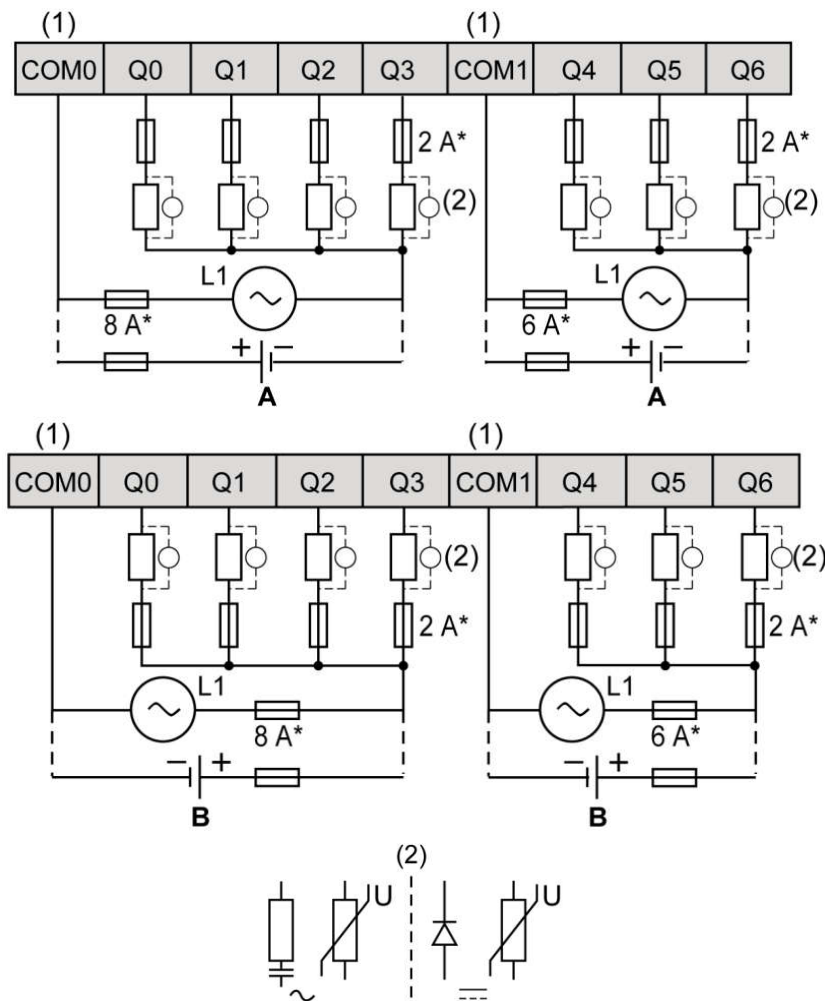


Рис. 1.31. Принципова схема підключення вихідних контактів ПЛК M221C16R: А – підключення Source (позитивна логіка); В – підключення Sink (негативна логіка); 1 – клеми COM; 2 – додатковий захист клем

Кількість входів/виходів в ПЛК – ще один важливий параметр. Найчастіше в ПЛК є 8, 16, 32 або навіть 64 входи/виходи. Наприклад, в ПЛК M221, M241, M251 і M262 від компанії Schneider Electric, передбачено можливість підключення модулів розширення ТМ3 (рис. 1.32), які дозволяють адаптувати ці ПЛК до різноманітних технічних завдань. В залежності від марки ПЛК та типів модулів можна підключати до 14 терміналів, не використовуючи жодних інструментів. Термінали Modicon ТМ3 дозволяють додати:

- цифрові модулі вводу-виводу для створення конфігурацій, розрахованих на максимум 264 цифрові входи/виходи;
- аналогові модулі вводу-виводу для створення конфігурацій, розрахованих на максимум 114 аналогових входів/виходів;
- експертні модулі для керування стартерами двигунів TeSys, які спрощують дротове з'єднання зони керування завдяки використанню кабелів RJ45;
- модулі розширення для дистанційних модулів у коробці чи іншій шафі (на відстані до 5 метрів), для яких використовується система розширення шини;

- модулі функціональної безпеки, які спрощують дротове з'єднання, налаштовані в програмному забезпеченні EcoStruxure Machine Expert – Basic.



Рис. 1.32. Модулі розширення введення/виведення ТМ3 для ПЛК Modicon M221, M241, M251 та M262

Аналогові модулі працюють з безперервними значеннями (наприклад, температурою, вологістю або тиском) та їхні виходи називаються каналами. Модулі можуть мати від 1 до 16 каналів. Підключення може бути однополюсним (спільна земля) або диференціальним (окремий плюс і мінус на кожний канал) (рис. 1.17). Важливо правильно вибрати діапазон сигналу: модуль може працювати, наприклад, з 0...10 В або з 4...20 мА. Деякі входи мають захист від перевищення сигналу – наприклад, якщо випадково подати 20 В замість 10 В. Також є параметр роздільної здатності – тобто наскільки точно ПЛК розпізнає аналоговий сигнал. Чим більше бітів у розрядності АЦП, тим точніші значення отримуються. Наприклад, 12-бітний модуль може розрізнити 4096 рівнів сигналу. При аналогових підключенні датчиків важливо враховувати вхідного опору та ємності. Для диференціальних каналів необхідно враховувати можливість виникнення шумів. Для зменшення цього впливу (наприклад, від електромагнітного випромінювання) при підключенні використовують виту пару, для придушення спільного шуму.

Висновки за розділом.

1. Програмовані логічні контролери (ПЛК) – це основа сучасних систем автоматизації, призначені для керування технологічними процесами шляхом виконання логічних операцій, збору та обробки сигналів від датчиків, а також формування керуючих впливів на виконавчі механізми. Вони забезпечують надійність, гнучкість і простоту модернізації систем.

2. Архітектура ПЛК включає центральний процесор (CPU), модулі пам'яті, інтерфейси зв'язку та модулі вводу/виводу. Така структура дозволяє здійснювати послідовну обробку сигналів, виконання програми користувача та обмін даними з іншими пристроями і системами.
3. Секція вводу/виводу (I/O) є проміжною ланкою між ПЛК і технологічним обладнанням. Модулі вводу приймають сигнали від датчиків, а модулі виводу передають команди на виконавчі пристрої. Від правильного вибору та налаштування секції I/O залежить точність і стабільність роботи системи.
4. Способи підключення пристроїв до ПЛК визначаються типом сигналів (дискретних або аналогових) і протоколами зв'язку. У промислових мережах використовуються як локальні з'єднання, так і мережеві інтерфейси, що дає змогу інтегрувати контролери у складні автоматизовані системи.
5. Типові схеми підключення пристроїв до ПЛК відображають принципи побудови систем автоматизації – правильне маркування, узгодження рівнів сигналів, застосування захисних елементів і стандартних позначень. Розуміння цих схем є основою для розробки, налагодження та експлуатації автоматизованих систем керування.

Питання для самоперевірки.

1. Що таке програмований логічний контролер (ПЛК) і які основні завдання він виконує в системах автоматизації?
2. Які переваги ПЛК порівняно з традиційними релейними схемами керування?
3. Які основні частини (архітектурні модулі) входять до складу ПЛК? Опишіть їхню роль.
4. Що являє собою центральний процесор (CPU) ПЛК і які функції він виконує?
5. Які типи пам'яті використовуються у ПЛК та для чого кожна з них призначена?
6. Що таке секція вводу/виводу (I/O) і яку роль вона відіграє у взаємодії ПЛК із зовнішніми пристроями?
7. Які типи сигналів можуть надходити на входи ПЛК і видаватися з його виходів? Наведіть приклади.
8. Які існують способи підключення пристроїв до ПЛК? Поясніть відмінності між дискретними та аналоговими каналами.
9. Наведіть приклади типових схем підключення датчиків і виконавчих механізмів до ПЛК.
10. Як забезпечується безпека та надійність роботи ПЛК при підключенні та експлуатації в промислових умовах?

РОЗДІЛ 2. МОВИ ПРОГРАМУВАННЯ ПЛК

2.1. Схеми релейно-контактної логіки (Ladder Diagrams).

У мікропроцесорних системах програми завантажуються у вигляді машинного коду – послідовності бінарних чисел, що відповідають конкретним інструкціям. Втім, щоб полегшити роботу, можна використовувати мову асемблеру, яка спирається на мнемонічні позначення. Наприклад, команда LD сигналізує про завантаження даних, що йдуть після неї, а спеціальна програма – асемблер – перетворює такі мнемоніки у машинний код [1, 2, 5, 6].

Ще простіше програмувати з використанням мов високого рівня, таких як C, BASIC, Pascal чи Python. Вони оперують готовими функціями, які позначаються простими словами чи символами, що відображають суть дії. Так, у мові C логічну операцію AND позначає символ &. Проте навіть для таких мов потрібні певні навички програмування. Оскільки програмовані логічні контролери (ПЛК) створені для інженерів, які не обов'язково мають глибокі знання програмування, було розроблено більш інтуїтивно зрозумілий підхід – релейно-контактну логіку, або LAD (Ladder Programming). За допомогою відповідного програмного забезпечення такі схеми перетворюються у машинний код, який виконується мікропроцесором ПЛК.

Метод Ladder набув широкого поширення серед виробників ПЛК, але кожен з них мав власне бачення його реалізації. Щоб уніфікувати підходи, у 1993 році було запроваджено міжнародний стандарт ІЕС 61131-3, який визначає правила не лише для LAD, а й для всіх основних методів програмування ПЛК.

Щоб ознайомитися з принципом побудови релейно-контактних схем (Ladder Diagrams), варто спочатку розглянути просту електричну схему керування двигуном (рис. 2.1). На типовому кресленні зображено, як за допомогою вимикача можна вмикати чи вимикати електродвигун (рис. 2.1а). Цю ж схему можна зобразити інакше: між двома вертикальними лініями, які символізують силові шини, розміщують усі інші елементи (рис. 2.1б). У такому вигляді, як це показано на прикладі, схема набуває вигляду «драбини» – звідси й назва «Діаграма драбин» або «Релейно-контактна схема (логіка)».

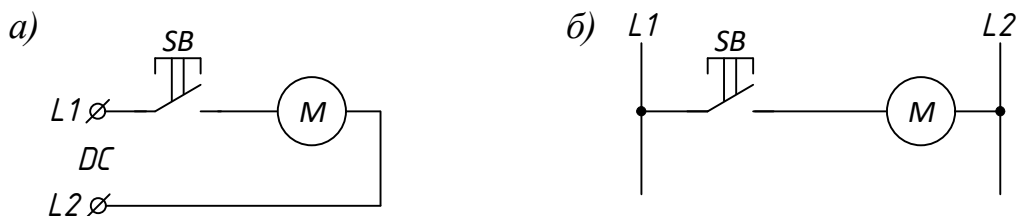


Рис. 2.1. Зображення електричних схем

У подібному зображенні вертикальні лінії завжди позначають джерело живлення, а між ними, у вигляді горизонтальних ліній (перекладин драбини), розміщуються компоненти схеми. Кожна горизонтальна ділянка відображає лише керуючу частину – у найпростішому випадку, це просто вимикач, з'єднаний

попередньо з електродвигун. На відміну від звичайних схем, де можна побачити фізичне розміщення елементів та їх з'єднання, Ladder Diagrams (LD) зосереджені лише на логіці керування, без прив'язки до реального розташування.

Ще один приклад – схема, що дозволяє запускати та зупиняти електродвигун за допомогою кнопок (рис. 2.2). У початковому стані кнопка «пуск» розімкнута, а кнопка «стоп» замкнута. Коли користувач натискає кнопку «пуск», замикається електричне коло, електродвигун починає працювати та спрацьовує проміжне реле контакти якого з'єднані паралельно з кнопкою «Пуск», та залишаються замкнутими, поки електродвигун увімкнений. Завдяки цьому, навіть коли кнопку «Пуск» відпускають, коло не розривається, та електродвигун продовжує працювати. Щоб зупинити електродвигун, потрібно натиснути кнопку «Стоп», яка розриває коло, вимикає електродвигун і розмикає контакти проміжного реле. Відпускання кнопки «Стоп» вже не змінює стан живлення до електродвигуна. Таким чином, запуск відбувається натисканням однієї кнопки, а зупинка іншою.

Тобто створення програми фактично на мові Ladder Diagrams зводиться до побудови схеми електричного керування. Діаграма складається з двох вертикальних ліній, які символізують силові шини, а між ними розміщуються горизонтальні відрізки «сходинки» драбини, що й утворюють логіку керування.

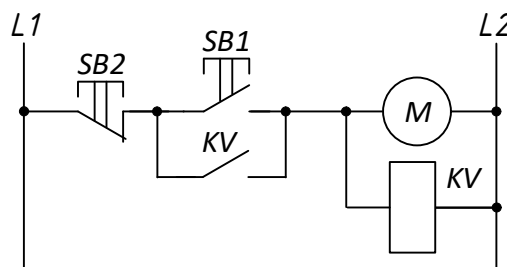


Рис. 2.2. Схема пуска та зупинки двигуна

Побудова Ladder Diagrams базується на певних умовних правилах. Вертикальні лінії завжди символізують джерела живлення, а рух електричного потоку прийнято вважати таким, що відбувається зліва направо по кожному рядку (сходинці). Кожна горизонтальна лінія відповідає окремій операції в системі керування. Схема читається (оброблюється ПЛК) зліва направо та зверху вниз: спочатку верхній рядок (сходинка), потім наступний нижче і так далі (рис. 2.3). У режимі "RUN" (виконання) ПЛК послідовно сканує всю програму до кінця, після чого негайно повертається на початок. Такий повний прохід через усі рядки (сходинки) називають циклом. Кінець програми позначають спеціальним блоком із написом END або RET (return), який сигналізує про повернення на початок. Тривалість одного циклу залежить від обсягу програми: у середньому близько 1 мс на 1000 байт коду. Типові значення часу сканування коливаються від 10 до 50 мс.

Структура кожного рядка (сходинки) повинна починатися з одного чи кількох «Входів» і завершуватися хоча б одним «Виходом». Під «Входом» розуміють дію керування, наприклад замикання контактів вимикача. «Вихід» –

це пристрій, підключений до ПЛК, такий як проміжне реле (магнітний пускач до якого під'єднаний електродвигун чи інший виконавчий механізм). Під час сканування програми виходи не змінюються одразу – результати виконання зберігаються в пам'яті, а всі виходи оновлюються одночасно лише після завершення проходження всіх рядків (сходинок).

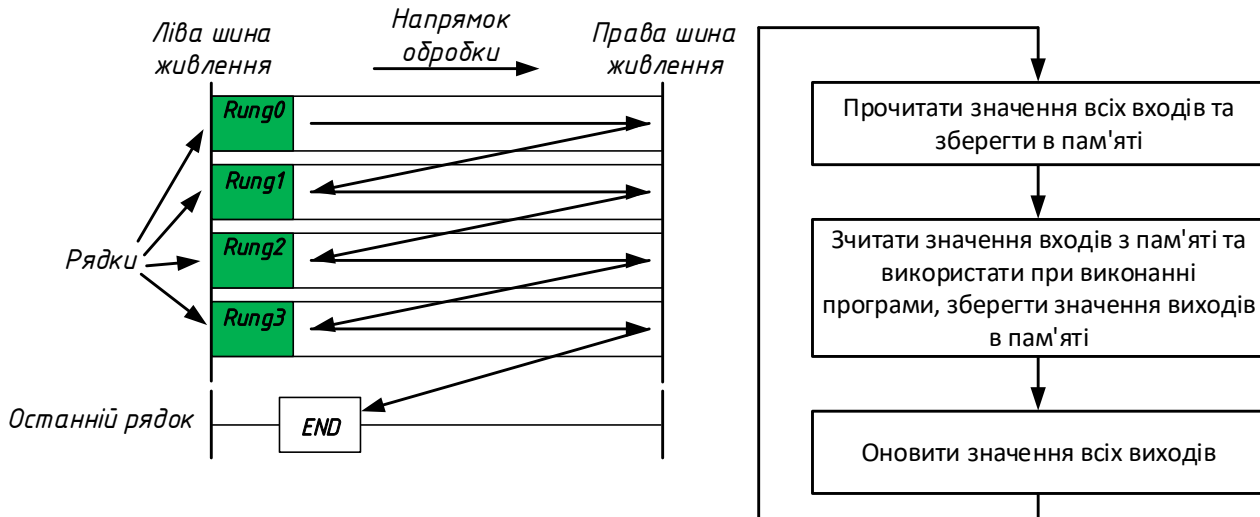


Рис. 2.3. Порядок обробки програми на мові Ladder Diagrams

На діаграмі всі елементи зображуються у їхньому нормальному стані. Тобто вимикач, який зазвичай розімкнутий і замикається лише при натисканні або впливі, показується відкритим, а вимикач, що зазвичай замкнутий, позначається замкнутим. Один і той самий пристрій може бути використаний на кількох рядках схеми. Наприклад, реле, яке керує декількома приладами, може бути відображене в різних місцях, але завжди з тим самим маркуванням. Усі входи та виходи мають адреси у пам'яті ПЛК, і їхнє позначення залежить від виробника контролера.

У стандарті ІЕС 61131-3 подано систему умовних позначень, що використовується для відображення вхідних і вихідних пристроїв у схемах програмованих логічних контролерів (рис. 2.4). Існують два варіанти цих символів: напівграфічні, які утворюються звичайним введенням із клавіатури, та повністю графічні, виконані за допомогою спеціальних інструментів креслення. Вхідні елементи зображуються у вигляді символів нормально розімкнутих або нормально замкнутих контактів, що за своєю дією нагадують звичайний вимикач, який замикає або розриває коло. Вихідні ж елементи – так звані котушки – мають єдине усталене позначення. Кожен символ супроводжується назвою змінної та її адресом, наприклад: кнопка «Пуск» може мати адресу **%I0.0**, а контакти котушки – **%Q0.0**.

Розгляньмо просту ситуацію: запуск електродвигуна здійснюється натисканням кнопки «Пуск», яка у вихідному стані є розімкнутою (рис. 2.5а). У такій діаграмі кнопка виступає вхідним елементом, а котушка реле (двигун) – вихідним. На схемі символ розімкнутого контакту з'єднаний з символом вихідної котушки. Коли кнопку натискають, коло замикається, і котушка реле (двигун)

отримує живлення. Але варто відпустити кнопку – і котушка реле (двигун) одразу розімкнеться (зупиниться). Якщо використати нормально замкнутий контакт, ситуація буде протилежною: котушка реле (двигун) буде працювати доти, доки цей контакт не розімкнеться (рис. 2.5б).

	Напівграфічне зображення	Графічне зображення
Горизонтальний зв'язок за яким передається живлення	-----	—————
Додаткові з'єднання за яким передається живлення: горизонтальні та вертикальні	-----+----- -----+----- 	—————+————— —————+—————
Лівостороннє приєднання до живлення в рядку	+----- 	—————
Правостороннє приєднання до живлення в рядку	-----+ 	—————+
Нормально розімкнутий контакт	— —	————— —————
Нормально замкнутий контакт	— / —	————— / —————
Вихідна котушка: якщо живлення надходить на котушку реле переключачється в стан «Включено»	— () —	————— () —————

Рис. 2.4. Базові позначення мови Ladder Diagrams в IEC 61131-3

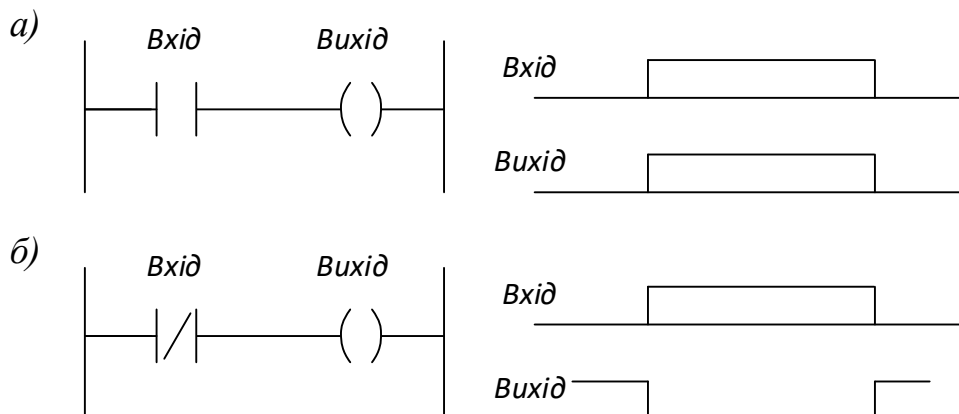


Рис. 2.5. Схема рележно-контактної логіки

Створюючи такі діаграми, важливо не обмежуватися сухими коментарями «Вхід» чи «Вихід». Значно виразніше й зрозуміліше звучать назви на кшталт «Вимикач керування насосом» або «Електродвигун насоса». В залежності від виробника та програмного забезпечення візуальне представлення елементів та позначень буде мати інший вигляд (рис. 2.6). Під час практичного підключення елементів до контролера обов'язково слід дотримуватися правил адресування у відповідності до програми виробника ПЛК, щоб забезпечити правильну роботу схеми.

При автоматизації технологічних процесів часто трапляються ситуації, коли певна дія повинна виконуватися лише за наявності визначеної комбінації умов. Наприклад, в автоматичному свердлильному верстаті може бути передбачено, що електродвигун свердла вмикається тільки тоді, коли спрацювали кінцеві вимикачі, які сигналізують про наявність заготовки та про те, що свердло розташоване саме на її поверхні. Такий випадок є прикладом реалізації логічної операції «І» (AND): для отримання вихідного сигналу необхідно, щоб одночасно виконувалася умова А та умова В.

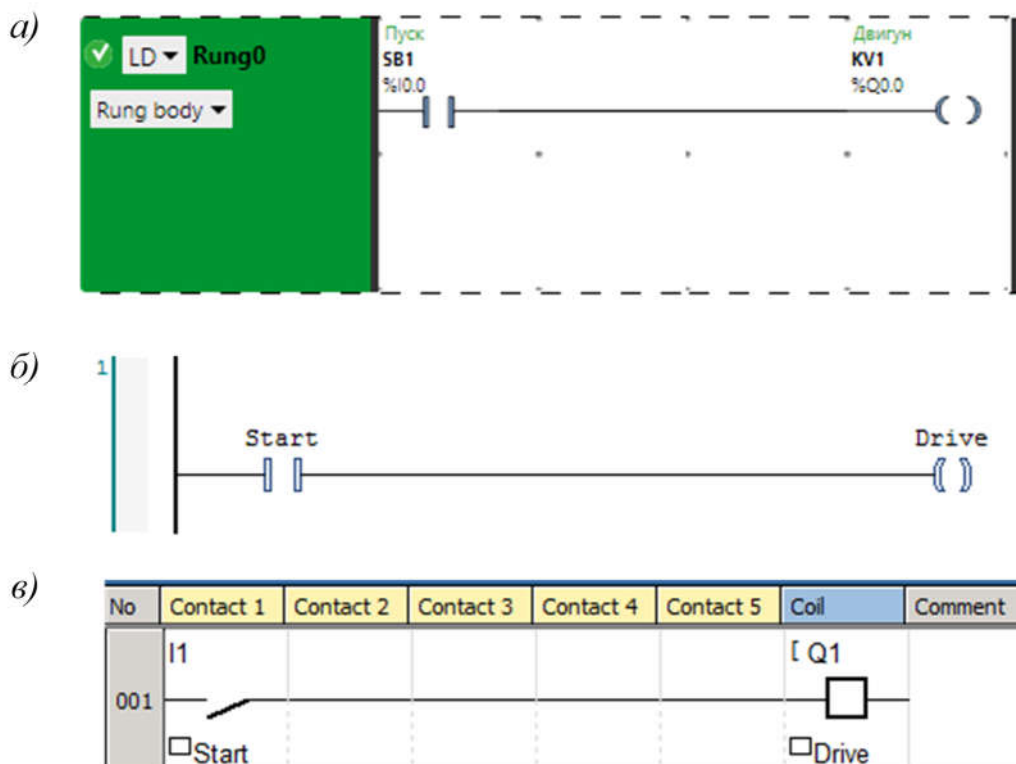


Рис. 2.6. Позначення елементів в програмному забезпеченні фірми Schneider Electric: а) Machine Expert – Basic; б) SoMachine; в) Zelio Soft 2

2.1.1. Логічна операція «І» (AND).

На рис. 2.7а зображено схему, у якій вихід не буде активований доти, доки обидва вимикачі А та В залишаються розімкнутими. Для появи вихідного сигналу необхідно, щоб обидва перемикачі були замкнені. Таким чином реалізується операція «І» (AND).

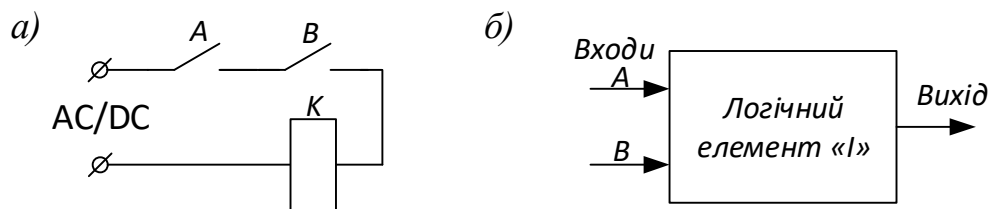


Рис. 2.7. Ланцюг з логікою «І» та елемент логіки «І» (AND)

Це можна уявити як систему керування з двома входами А та В (рис. 2.7б). Лише тоді, коли обидва входи одночасно перебувають у стані «Увімкнено» (1), формується вихідний сигнал. У всіх інших випадках вихід дорівнює нулю (0).

Подібна робота описується через логічні елементи, а співвідношення між вхідними та вихідними сигналами подається у вигляді таблиці істинності. Для елемента «І» (AND) ця таблиця має такий вигляд:

Входи		Вихід
A	B	
0	0	0
0	1	0
1	0	0
1	1	1

Прикладом використання логічного елемента «І» (AND) є система блокувань у верстатах. Така система забезпечує, що обладнання може працювати лише за умови, коли захисний кожух знаходиться у правильному положенні, а живлення ввімкнене.

На рис. 2.8а показано реалізацію логічної операції «І» (AND) у вигляді схеми релейно-контактної логіки. Вона починається з пари нормально розімкнених контактів, позначених як **Вхід А**, що відповідає вимикачу **А**, після чого послідовно розташовано ще одну пару контактів, **Вхід В**, який представляє вимикач **В**. Лінія закінчується символом «()», що позначає вихід.

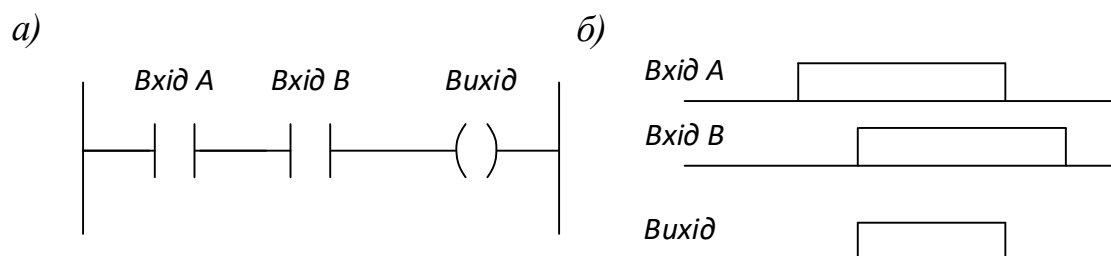


Рис. 2.8. Використання логічної операції «І» (AND) в схемі релейно-контактної логіки

Для появи вихідного сигналу необхідно, щоб обидва входи були замкнені одночасно тобто контакти **А** і **В** повинні перебувати у стані «Увімкнено» (рис. 2.8б).

Загальне визначення: сигнал на виході з'являється тільки тоді, коли одночасно на всіх входах присутні сигнали.

2.2.2. Логічна операція «АБО» (OR).

На рис. 2.9а наведено електричну схему, в якій вихід активується, якщо замкнений хоча б один з двох нормально розімкнених вимикачів А або В. Це відповідає логічному елементу «АБО» (OR) (рис. 2.9б): для появи виходу достатньо сигналу від будь-якого з входів. Відповідні комбінації вхідних та вихідних станів зведені до таблиці істинності:

Входи		Вихід
A	B	
0	0	0
0	1	1
1	0	1
1	1	1

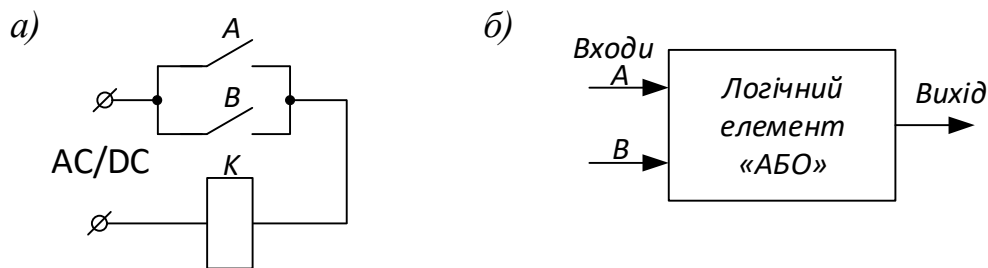


Рис. 2.9. Ланцюг з логікою «АБО» та елемент логіки «АБО» (OR)

На рис. 2.10а представлена реалізація логічного елемента «АБО» (OR) у вигляді схеми релейно-контактної логіки; на рис. 2.10б наведено альтернативний (еквівалентний) спосіб її зображення. Схема починається з контакту який позначає нормально розімкнений контакт **Вхід А**, який відповідає вимикачу А. Паралельно до нього розташовано ще один контакт **Вхід В**, для вимикача В. У результаті вихід спрацює, якщо замкнено хоча б один із входів (рис. 2.10в). Лінія закінчується символом «()», що позначає вихідний елемент.

Узагальнено принцип роботи можна описати так: «Альтернативні шляхи, що створюються паралельними вертикальними гілками до основного рядка діаграми, реалізують логічну операцію «АБО» (OR)».

Прикладом застосування «АБО» (OR) у автоматичному керуванні є транспортерна лінія для подачі пляшок на пакування. Якщо вага пляшки виходить за межі допустимих значень або на ній відсутня кришка, спеціальна відвідна пластина автоматично спрацьовує, відправляючи такий виріб у відбракований контейнер.

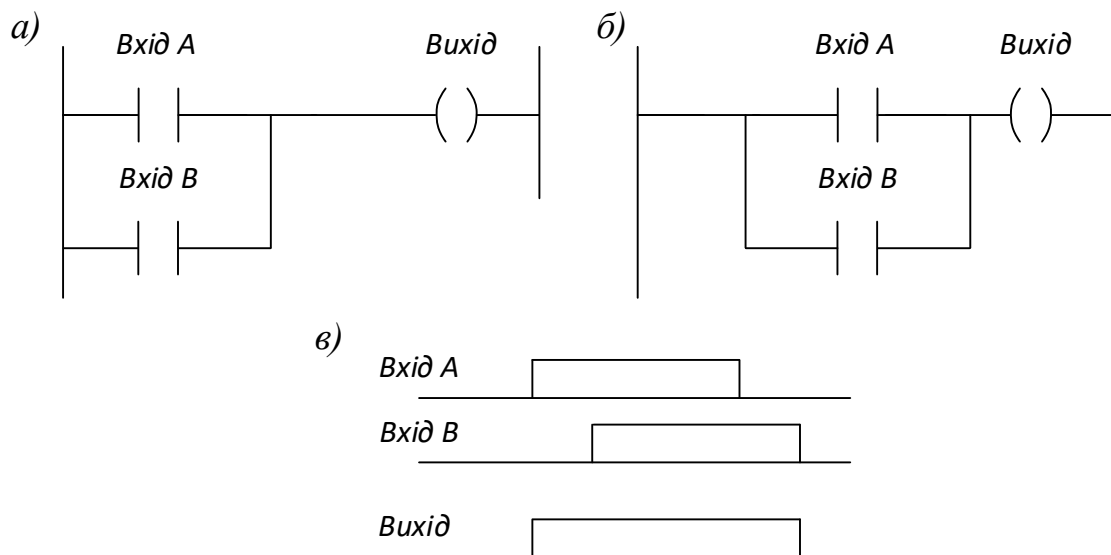


Рис. 2.10. Використання логічної операції «АБО» (OR) в схемі релейно-контактної логіки

2.2.3. Логічна операція «НІ» (NOT).

На рис. 2.11а зображено електричне коло, кероване вимикачем, що перебуває у нормально замкненому стані. Коли на нього подається вхідний сигнал, контакти розмикаються, і струм у колі припиняється. Така ситуація є прикладом роботи логічного елемента «НІ» (NOT), тобто сигнал на виході з'являється лише тоді, коли відсутній сигнал на вході, та зникає, щойно на вхід надходить сигнал (рис. 2.11в). Через цю особливість даний елемент часто називають інвертором, адже він «перевертає» сигнал на протилежний.

Вхід	Вихід
А	
0	0
1	1

На схемі релейно-контактної логіки (рис. 2.11б) елемент «НІ» (NOT) відображається контактами **Входу А**, виконаними у вигляді нормально замкнених. Вони підключені послідовно з виходом «()». За відсутності сигналу на **Вході А** контакти залишаються замкненими, і в колі є сигнал на виході. Якщо на вхід подати сигнал, контакти розімкнуться, і сигнал на виході зникне.

Як приклад застосування можна навести систему керування освітленням, коли лампа вмикається лише за настання темряви. Тобто, поки датчик не отримує світлового сигналу на вихід надходить сигнал та світильник загоряється.

2.2.4. Логічна операція NAND.

Якщо після логічного елемента AND встановити елемент NOT (рис. 2.12а), вихідний сигнал буде повністю інвертований. Іншими словами, будь-який результат роботи елемента AND одразу ж змінюється на протилежний.

До того ж існує й інший спосіб отримати такий самий ефект. Якщо кожен вхід попередньо пропустити через інвертор, а вже потім підключити їх до елемента OR, ми знову отримаємо ідентичний результат (рис. 2.12б).

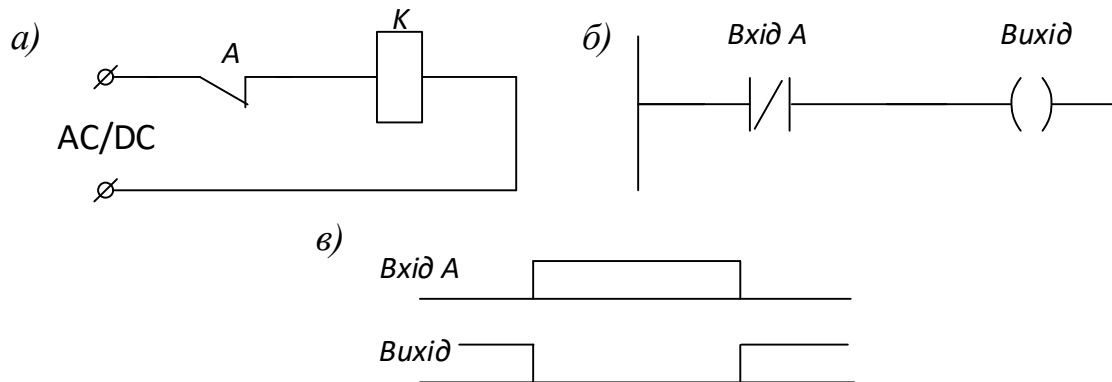


Рис. 2.11. Використання логічної операції «НІ» (NOT) в схемі релейно-контактної логіки

Таким чином, обидва варіанти формують однакову таблицю істинності, яка демонструє роботу елемента NAND: вихід дорівнює нулю лише тоді, коли всі входи одночасно активні; в усіх інших випадках формується одиниця:

Входи		Вихід
A	B	
0	0	1
0	1	1
1	0	1
1	1	0

Для отримання сигналу на виході в логічному елементу типу NAND достатньо, аби хоча б один із входів **A** або **B** перебував у стані «0». Це означає, що вихід формується завжди, коли хоча б один із сигналів неактивний. Лише в ситуації, коли обидва входи водночас перебувають у стані «1», вихідний сигнал зникає і дорівнює нулю.

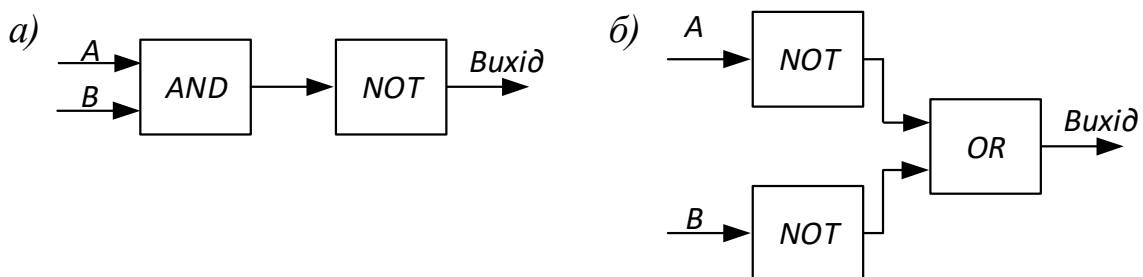


Рис. 2.12. Логічний елемент NAND

На рис. 2.13 наведено приклад реалізації логічного елементу типу NAND у вигляді схеми релейно-контактної логіки. В якій при будь-якій комбінації, де **Вхід А** або **Вхід В** (або обидва) дорівнюють нулю, на виході виводиться одиниця. Проте якщо одночасно **Входи А** та **В** дорівнюють **1**, вихідна лінія розривається, та на сигнал **Виході** буде дорівнювати **0**.

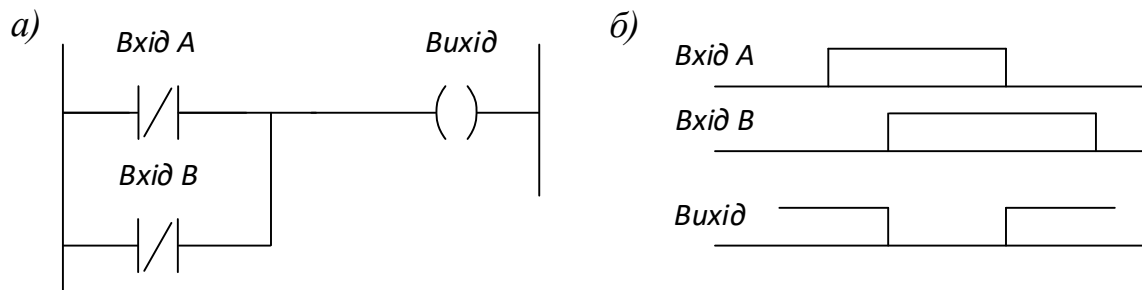


Рис. 2.13. Використання логічної операції NAND в схемі релейно-контактної логіки

Прикладом застосування логіки NAND може слугувати система індикації небезпеки на верстаті. У такій схемі попереджувальна лампа загоряється тоді, коли не спрацював вимикач захисного кожуха або не зафіксовано сигнал від кінцевого вимикача, що підтверджує наявність заготовки. Інакше кажучи, світлове попередження з'являється щоразу, коли хоча б одна з умов безпеки не виконується.

2.2.5. Логічна операція NOR.

Якщо після логічного елемента OR під'єднати елемент NOT, вихідний сигнал інвертується, і вся схема працює вже за іншими правилами (рис. 2.14а). Тобто результат операції OR змінюється на протилежний. Того самого ефекту можна досягти й іншим шляхом: достатньо інвертувати кожен із вхідних сигналів за допомогою елементів NOT, а потім об'єднати їх через логіку AND (рис. 2.14б). Обидва варіанти дають ідентичний результат, що підтверджується відповідною таблицею істинності:

Входи		Вихід
А	В	
0	0	1
0	1	0
1	0	0
1	1	0

Поєднання логічних елементів OR і NOT утворює так званий елемент NOR. Його робота ґрунтується на простому правилі: вихідний сигнал з'являється лише тоді, коли жоден із вхідних сигналів не активний. Іншими словами, якщо **Вхід А**, і **Вхід В** перебувають у стані **0**, на **Виході** формується **1**. Натомість **активація**

хоча б одного з **Входів** миттєво призводить до появи **0** на **Виході**. Схематично принцип дії такої логіки можна відобразити за допомогою схеми релейно-контактної логіки (рис. 2.15), де наочно видно залежність між відсутністю сигналів на входах та формуванням єдиного позитивного результату.

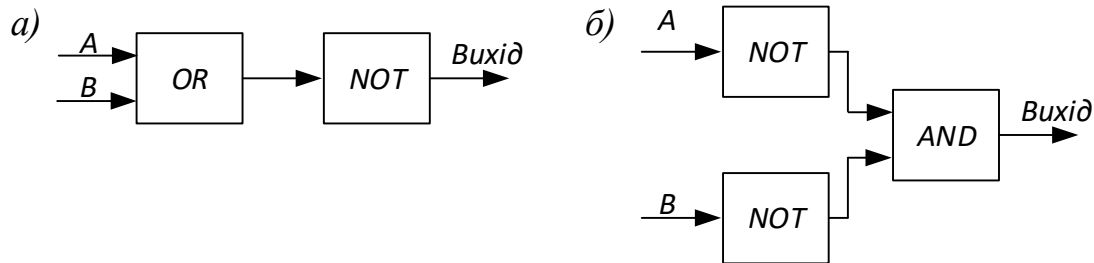


Рис. 2.14. Логічний елемент NOR

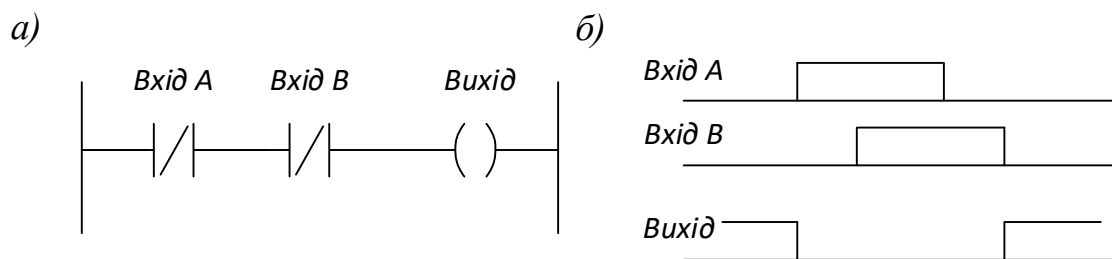


Рис. 2.15. Використання логічної операції NOR в схемі релейно-контактної логіки

2.2.6. Логічна операція XOR.

Якщо розглянути комбінацію логічних елементів, можна побудувати схему, яка реалізує операцію виключного АБО (XOR). У цьому випадку вихідний сигнал з'являється лише тоді, коли активним є один із входів, але не обидва одночасно. Іншими словами, коли на входах різні стани (0 і 1 або 1 і 0), на виході формується 1. Якщо ж обидва входи перебувають у однаковому стані або обидва 0, або обидва 1 на виході з'являється 0. Такий результат можна отримати кількома способами: наприклад, поєднуючи елементи OR, AND та NOT у певній послідовності. Незалежно від схеми побудови, результат роботи завжди залишається однаковим і чітко відображається у таблиці істинності для функції XOR:

Входи		Вихід
A	B	
0	0	0
0	1	1
1	0	1
1	1	0

Поєднання кількох базових логічних елементів дає змогу побудувати схему, яка реалізує функцію виключного АБО, або XOR. Принцип її дії полягає в тому, що вихідний сигнал з'являється лише тоді, коли активним є один із входів, але не обидва одночасно. Якщо **Вхід А** перебуває у стані **1**, а **Вхід В** у стані **0**, на **Виході** формується сигнал **1**. Те саме відбувається й у зворотній ситуації: коли **Вхід А** дорівнює **0**, а **Вхід В** дорівнює **1**. Однак, якщо обидва входи перебувають у стані **0** або одночасно у стані **1**, результатом роботи схеми буде **0**.

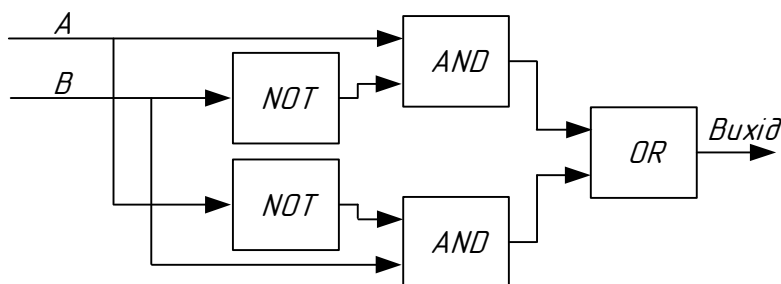


Рис. 2.16. Логічний елемент XOR

Таким чином, елемент XOR реагує саме на різницю станів входів, і саме ця властивість робить його особливо зручним у системах, де необхідно фіксувати зміну або взаємовиключність подій. Схематично роботу такої логіки можна відобразити за допомогою схеми релейно-контактної логіки (рис. 2.17), де чітко видно, що вихід активується лише тоді, коли входи перебувають у протилежних станах.

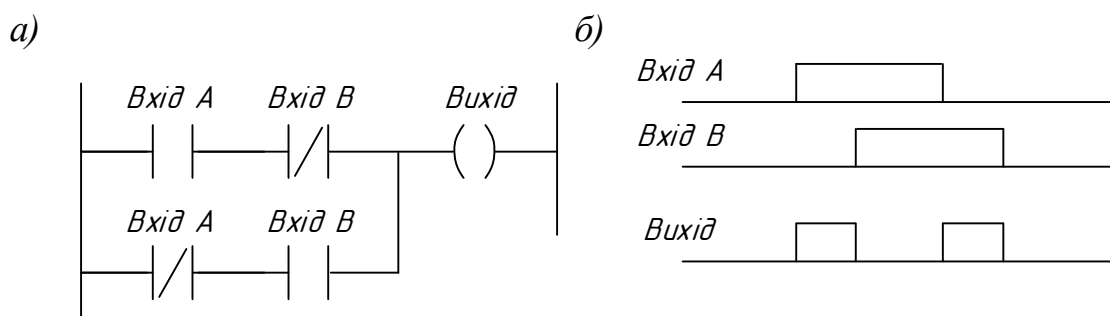


Рис. 2.17. Використання логічної операції XOR в схемі релейно-контактної логіки

Наприклад, якщо на конвеєрі стоять два датчики (А і В), які мають фіксувати наявність виробу у двох контрольних точках. Потрібно вмикати сигнал тривоги (лампку/сирену) тільки тоді, коли датчики дають різні сигнали (тобто один бачить виріб, інший ні). Це класичне застосування XOR: тривога при невідповідності.

2.2.7. Фіксація (Latching).

У практиці керування часто виникають ситуації, коли необхідно зберегти вихід у ввімкненому стані навіть після зникнення вхідного сигналу. Класичним прикладом є пуск електродвигуна за допомогою кнопкового вимикача: хоча контакти кнопки пружинами автоматично роз'єднуються (кнопка нормально

розімкнута без фіксації «Старт»), двигун повинен продовжувати працювати доти, доки не буде натиснута кнопка «Стоп». Для виконання такої функції застосовується *фіксує коло* (Latch Circuit). Це самопідтримувальна схема, яка після початкового увімкнення утримує свій стан аж до моменту надходження іншого, керуючого сигналу.

Приклад фіксує коло наведено на рис. 2.18. Коли замикаються вхідні контакти **A**, формується вихідний сигнал. Водночас з його появою замикається ще один комплект контактів, асоційованих із **Виходом**. Ці контакти утворюють із вхідними контактами систему логіки **OR**. Таким чином, навіть якщо **Вхід А** розімкнеться, коло продовжує утримувати **Вихід** в увімкненому стані. Єдиним способом скинути **Вихід** є спрацювання нормально замкненого контакту **B**.

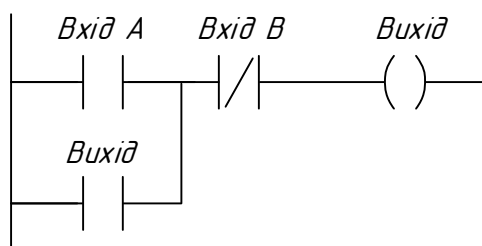


Рис. 2.18. Фіксує коло (Latch Circuit)

Застосування фіксує схем добре ілюструє приклад керування електродвигуном, для якого передбачені кнопки «Стоп» і «Пуск», а також два сигнальні індикатори: один має світитися при подачі живлення на двигун, інший коли двигун знеструмлений. На рис. 2.19 наведено схему релейно-контактної логіки з використанням позначень Schneider Electric для адрес. Контакт %I0.0 замикається для запуску програми. Коли короткочасно замикається %I0.0, відбувається подача сигналу на вихід %Q0.0 і замикання його контактів. У результаті формується стан фіксації, а також відбувається вимикання %Q0.1 і вмикання %Q0.2. Для вимкнення двигуна натискають кнопку, яка розмикає контакт %I0.1. При цьому контакти %Q0.0 розмикаються в Rung0 та Rung2, але замикаються в Rung1. Таким чином, вмикається %Q0.1 і вимикається %Q0.2.

Фіксація знаходить широке застосування у системах запуску, оскільки дозволяє перетворити короткочасне увімкнення у стабільний стан роботи пристрою.

2.2.8. Використання декількох виходів.

У схемах релейно-контактної логіки можливим є підключення кількох вихідних елементів до одного й того самого контакту. Так, на рис. 2.20 наведено приклад програми у вигляді схема релейно-контактної логіки, де до вхідних контактів приєднано дві вихідні котушки. У момент замикання вхідних контактів обидві котушки формують сигнали на виходах.

Інший варіант наведено на рис. 2.21: вихід А активується одночасно із замиканням входу А, тоді як вихід В з'являється лише за умови, якщо одночасно замкнені вхід А, та вхід В.

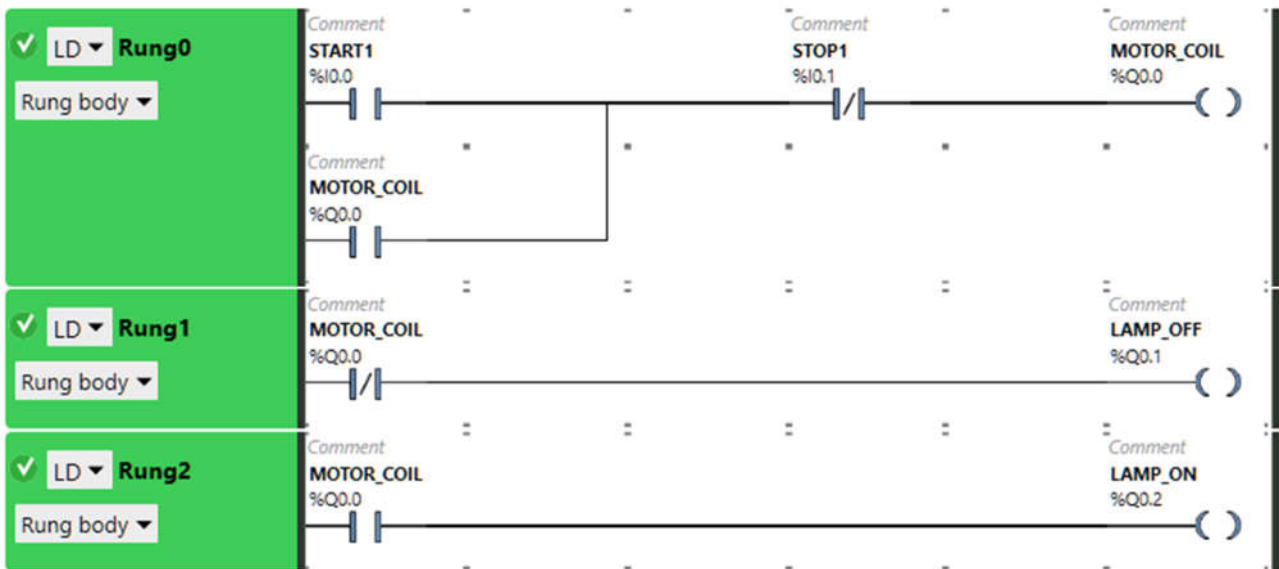


Рис. 2.19. Схема релейно-контактної логіки керуванням електродвигуном в програмі Machine Expert – Basic від Schneider Electric

Подібні схеми розширюють можливості побудови послідовності вихідних сигналів, які формуються відповідно до порядку замикання контактів. Наприклад на рис. 2.22 наведено схема релейно-контактної логіки де виходи А, В і С послідовно активуються в такому ж порядку, в якому замикаються відповідні входи А, В та С. Допоки вхід А не замкнений, жоден із наступних виходів не може бути активований. Замикання входу А приводить до появи виходу А; після цього замикання входу В спричиняє появу виходу В. Нарешті, замикання входу С активує вихід С. Прикладом такої послідовності є транспортерна лінія для подачі сировини, обладнана трьома вузлами: живильником, основним конвеєром і вентилятором системи відсмоктування пилу. Логіка роботи такої системи передбачає, що всі механізми мають вмикатися в чітко визначеній послідовності, інакше запуск може призвести до перевантаження або аварійної ситуації.

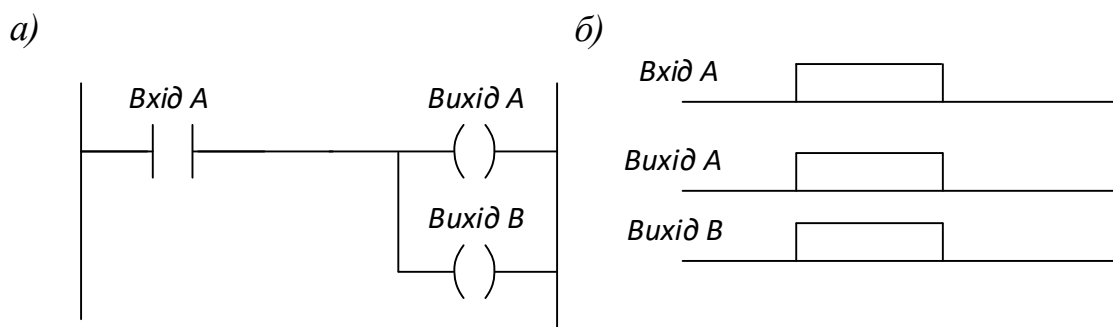


Рис. 2.20. Використання двох виходів у схемі релейно-контактної логіки

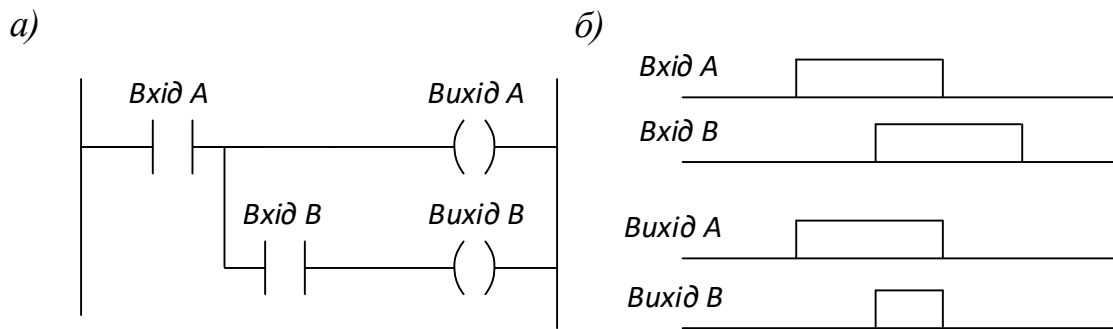


Рис. 2.21. Використання двох входів і виходів у схемі релейно-контактної логіки

На першому етапі оператор натискає кнопку «Пуск». Це еквівалент замикання вхідного контакту А, у результаті чого вмикається перший вихід – живильник. Далі, лише після замикання наступного вхідного контакту В (що може бути реалізовано кінцевим вимикачем, який сигналізує про стабільну роботу живильника), активується вихід В – конвеєр. Нарешті, при замиканні входу С, що підтверджує нормальний рух конвеєрної стрічки, подається сигнал на вихід С, яким вмикається вентилятор аспіраційної системи.

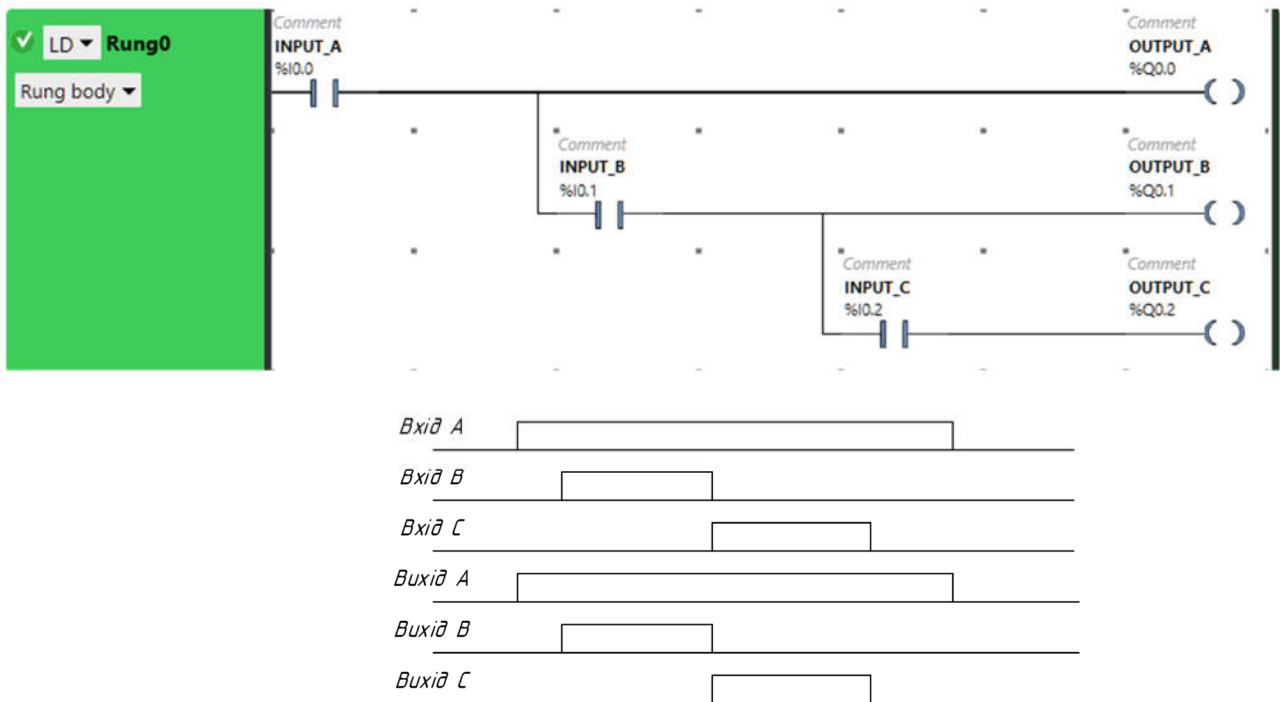


Рис. 2.22. Схема релейно-контактної логіки послідовного ввімкнення в програмі Machine Expert – Basic від Schneider Electric

2.2.9. Приклади програмування.

Розглянемо прикладну задачу керуванням наповненням двох резервуарів за пріоритетом заповнення. У пневматичній системі транспортування матеріалу використовуються два накопичувальні резервуари (Tank1 та Tank2) (рис. 2.23). Іноді виникає ситуація, коли обидва резервуари одночасно порожні, тому визначаємо пріоритет заповнення за першим резервуаром. Визначимо за схемою

технологічного процесу транспортування матеріалу необхідні входи та виходи (табл. 2.1).

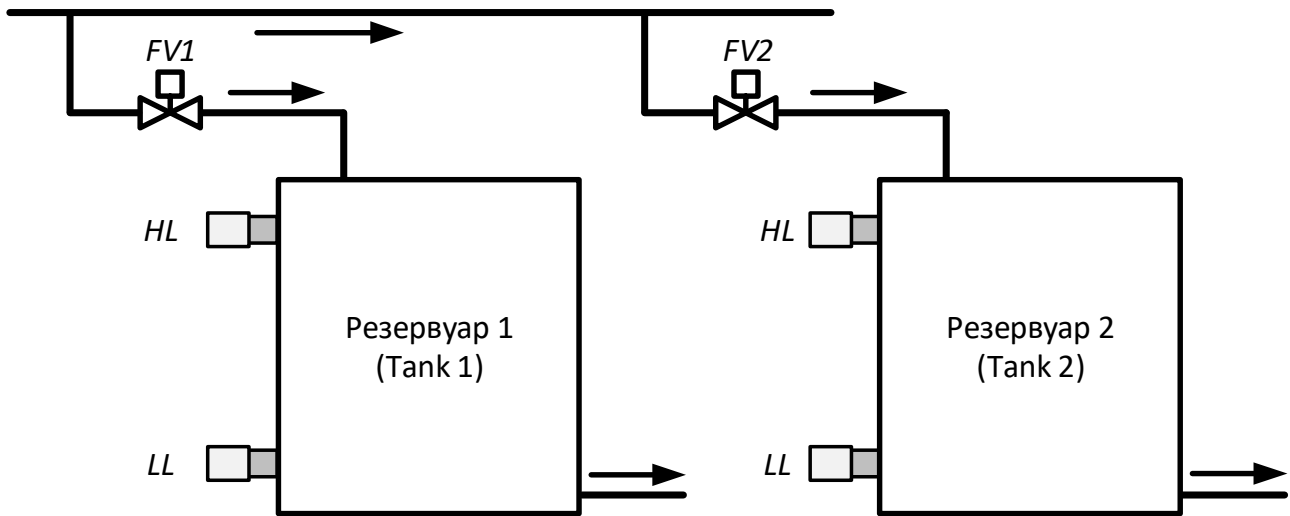


Рис. 2.23. Схема технологічного процесу транспортування матеріалу

Таблиця 2.1

Список входів та виходів технологічного процесу транспортування матеріалу

Опис	Позначення	
	Символьне	Адресне
Список входів		
Кнопка ПУСК	START_PB	%I0.0
Кнопка СТОП	STOP_PB	%I0.1
Датчик нижнього рівня резервуара 1	LL T1	%I0.2
Датчик нижнього рівня резервуара 2	LL T2	%I0.3
Датчик верхнього рівня резервуара 1	HL T1	%I0.4
Датчик верхнього рівня резервуара 2	HL T2	%I0.5
Список виходів		
Клапан резервуара 1	FV1	%Q0.0
Клапан резервуара 2	FV2	%Q0.1
Комірки пам'яті		
Засувка виконання програми	LATCH	%M0

Використовуючи список входів та виходів технологічного процесу наповнення резервуарів, складемо програму на мові LD (рис. 2.24).

Розглянемо прикладну задачу керуванням двигуна стрічкового транспортеру в прямому та зворотному напрямку з кінцевими вимикачами. Заготовка починає рухатись з лівого боку та рухається праворуч при натисканні кнопки запуску (рис. 2.25). Коли вона досягне крайньої правої межі, привідний двигун змінює напрямок руху на «Реверс» і знову повертає заготовку в крайнє ліве положення, і процес повторюється (табл. 2.2). Кнопки прямого та зворотного руху забезпечують запуск двигуна вперед або назад, щоб кінцеві вимикачі могли

брати на себе автоматичне керування. Керування двигуном здійснюється з використанням двох магнітних пускачів, що ввімкнені через проміжні реле.

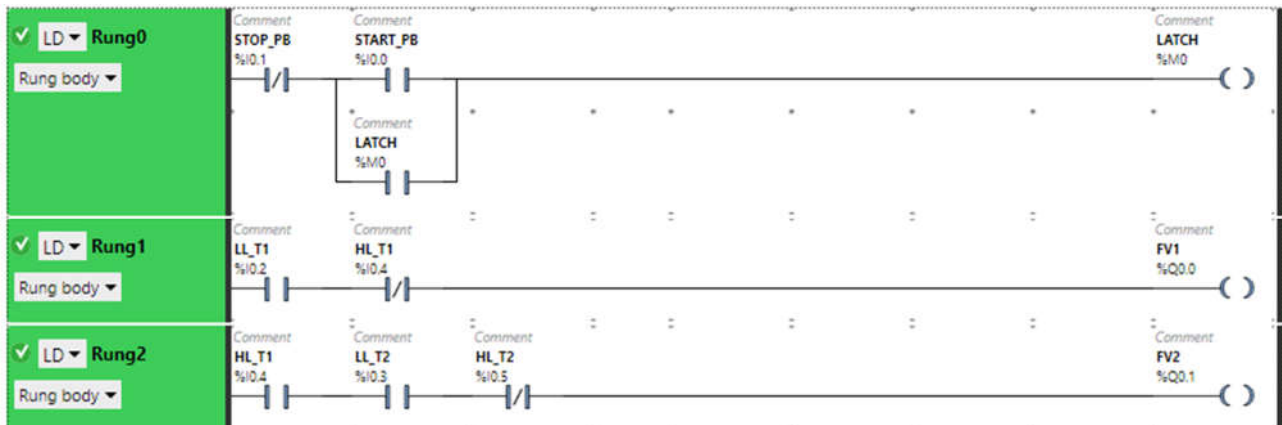


Рис. 2.24. Схема релейно-контактної логіки транспортування матеріалу в програмі Machine Expert – Basic від Schneider Electric

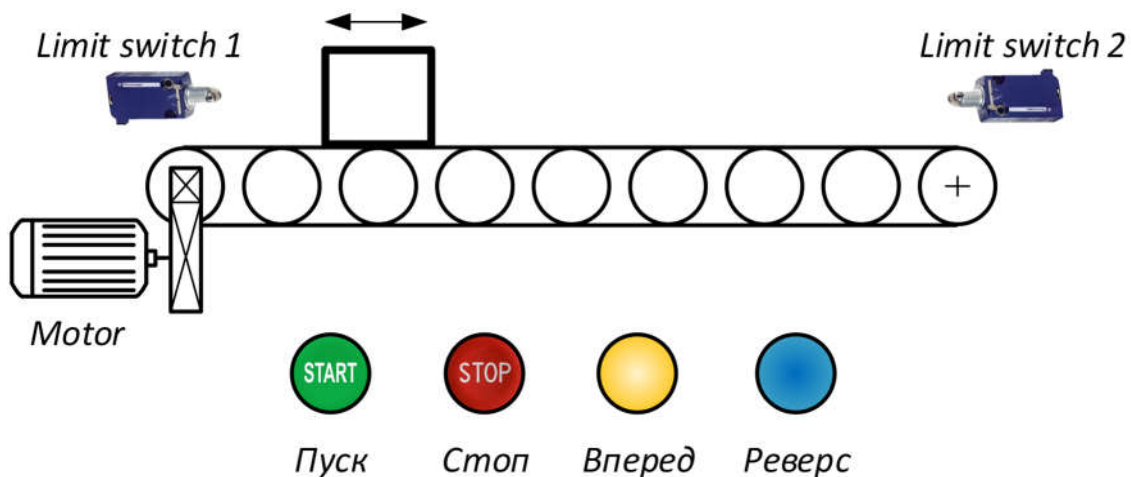


Рис. 2.25. Схема керуванням двигуна стрічкового транспортеру

Використовуючи список входів та виходів опис роботи конвеєрного транспортеру, складемо програму на мові LD (рис. 2.26).

Розглянемо прикладну задачу керуванням кормороздавачем, який має три накопичувальних бункери та розподільчим бункером (рис. 2.27). Робота кормороздавача починається з моменту, коли датчики рівня у бункері фіксують нестачу корму: нижній сигналізує про його критично малу кількість, а верхній підтверджує, що ємність ще не переповнена. Саме за цих умов формується команда на запуск технологічного процесу. Вибір джерела подачі здійснюється за визначеною пріоритетністю. Насамперед у роботу включається перший силос: відкривається його пневматична заслінка, та корм потрапляє на транспортерний механізм (стрічковий транспортер та норії). Якщо виявляється, що вміст першого силосу вичерпано, система автоматично переключається на другий, а в разі його спорожнення на третій. Таким чином забезпечується безперервність роботи навіть у ситуаціях, коли один або два резервуари порожні.

Таблиця 2.2

Список входів та виходів керуванням двигуна стрічкового транспортеру

Опис	Позначення	
	Символьне	Адресне
Список входів		
Кнопка ПУСК	START_PB	%I0.0
Кнопка СТОП	STOP_PB	%I0.1
Кнопка ВПЕРЕД	F_PB	%I0.2
Кнопка РЕВЕРС	R_PB	%I0.3
Кінцевий вимикач 1	LS 1	%I0.4
Кінцевий вимикач 2	LS 2	%I0.5
Список виходів		
Двигун прямий хід	MF	%Q0.0
Двигун реверсний хід	MR	%Q0.1
Комірки пам'яті		
Засувка виконання програми	LATCH	%M0

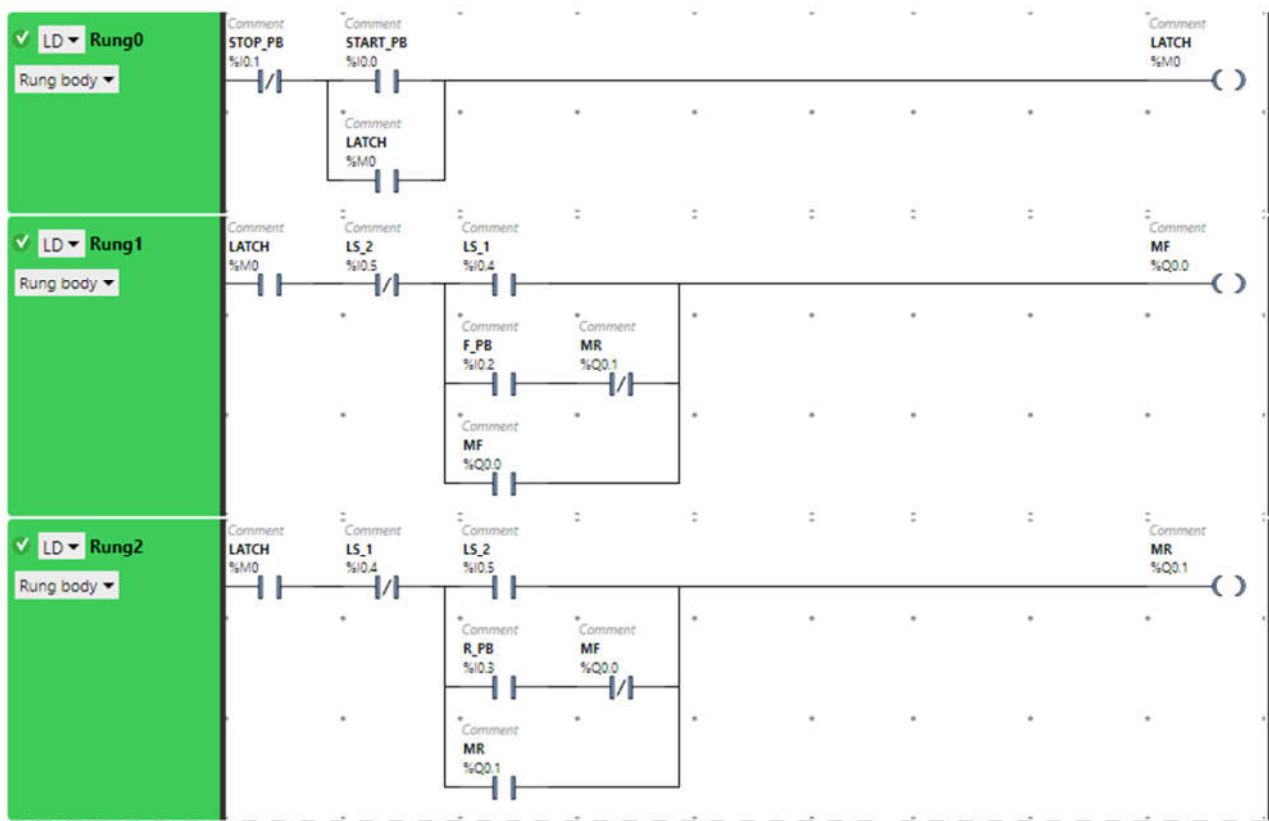


Рис. 2.26. Схема релейно-контактної логіки керуванням двигуна стрічкового транспортеру в програмі Machine Expert – Basic від Schneider Electric

Разом із відкриттям однієї з заслінок запускається транспортерний механізм, що переміщує корм безпосередньо в розподільчий бункер. Він діє доти, доки верхній датчик рівня не зафіксує наповнення ємності. У цей момент

контролер дає команду на закриття клапанів і зупинку двигуна, переводячи систему в режим очікування.

Визначимо за схемою керуванням кормороздавачем необхідні входи та виходи (табл. 2.3).

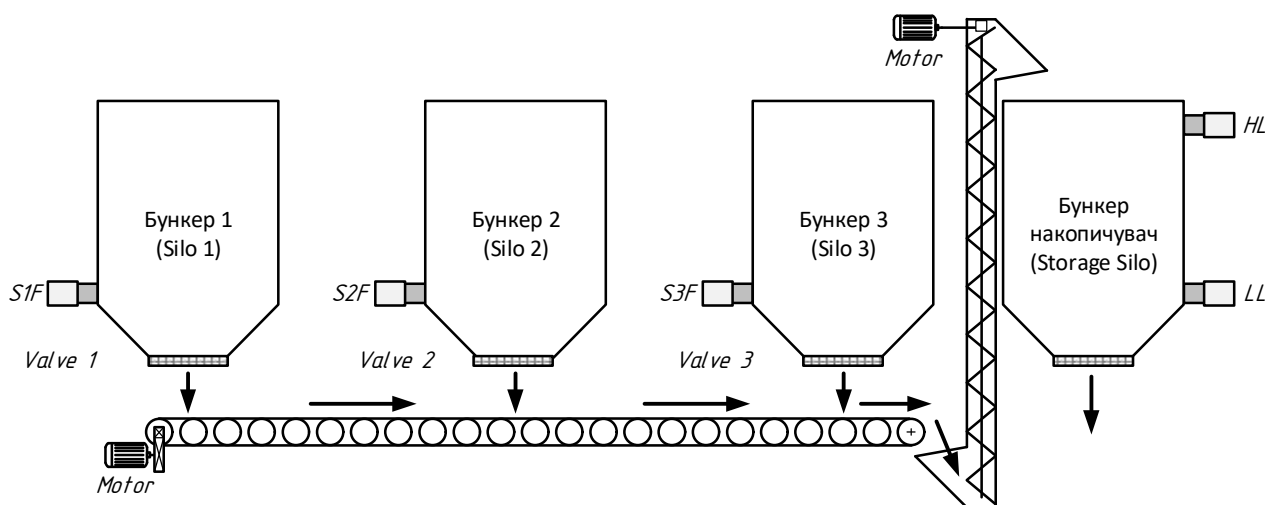


Рис. 2.27. Схема керуванням кормороздавачем

Таблиця 2.3

Список входів та виходів керуванням кормороздавачем

Опис	Позначення	
	Символьне	Адресне
Список входів		
Кнопка ПУСК	START PB	%I0.0
Кнопка СТОП	STOP PB	%I0.1
Нижній рівень в бункері накопичувачі	LL	%I0.2
Верхній рівень в бункері накопичувачі	HL	%I0.3
Статус накопичувального бункера 1	S1F	%I0.4
Статус накопичувального бункера 2	S2F	%I0.5
Статус накопичувального бункера 3	S3F	%I0.6
Список виходів		
Заслінка накопичувального бункера 1	VALVE1	%Q0.0
Заслінка накопичувального бункера 2	VALVE2	%Q0.1
Заслінка накопичувального бункера 3	VALVE3	%Q0.2
Транспортерний механізм	MOTOR	%Q0.3
Комірки пам'яті		
Засувка виконання програми	LATCH	%M0
Перевантаження двигуна або заклинювання заслінок	FAULT	%M1
Процес завантаження	LOAD	%M2
Аварійний сигнал	ALARM	%M3

Особлива увага приділяється безпеці роботи. У разі перевантаження двигуна або заклинювання будь-якої з заслінок система миттєво зупиняє подачу та подає аварійний сигнал. Після усунення несправності повторний запуск

можливий лише після натискання кнопки скидання. Завдяки такому алгоритму робота комплексу стає не лише безперервною, а й захищеною від аварійних ситуацій, що забезпечує стабільність виробничого процесу та зменшує потребу у втручанні оператора.

Використовуючи список входів та виходів (табл. 2.3) опис роботи кормороздавача, складемо спрощену програму на мові LD (рис. 2.28).

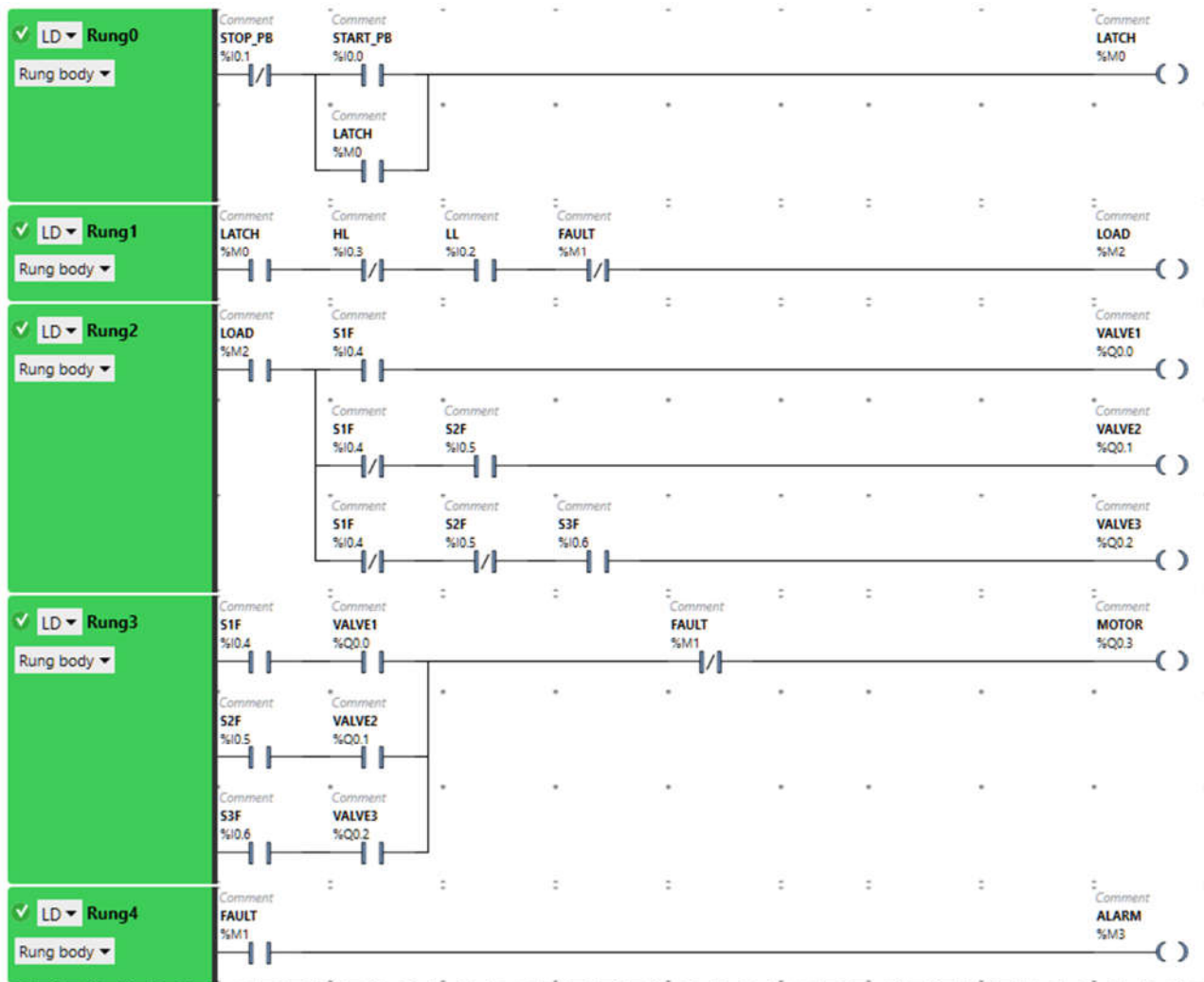


Рис. 2.28. Схема релейно-контактної логіки керування кормороздавачем в програмі Machine Expert – Basic від Schneider Electric

2.2. Список інструкцій (Instruction Lists).

Одним із методів програмування, який можна розглядати як введення схем релейно-контактної логіки у вигляді тексту, є мова визначена стандартом IEC 61131-3 *список інструкцій* (Instruction Lists, IL) [1, 5, 6]. Ця мова програмування подається у вигляді послідовності команд, кожна з яких розташована в новому рядку. Кожна інструкція складається з оператора та одного або кількох операндів, тобто об'єктів, до яких застосовується дія оператора.

Наприклад, інструкція:

LD A

означає завантаження операнду А у пам'ять (регістр результатів або акумулятор), де LD виступає оператором завантаження. У термінах схем релейно-контактною логіки оператор можна трактувати як елемент сходової логіки, а команда LD А відповідає початку гілки зі замикаючим контактом, що відображає вхід А.

Інша інструкція,

OUT Q

означає, що результат потрібно вивести на вихід Q.

Для операторів застосовуються мнемонічні коди, кожен із яких відповідає певній операції або елементу схем релейно-контактною логіки. Хоча конкретні коди можуть різнитися залежно від виробника, запропоновано єдиний стандарт у межах IEC 61131-3, який нині набув широкого поширення. У таблиці 2.4 наведено приклади кодів, що використовуються різними виробниками, та рекомендовані стандартні позначення для інструкцій.

Списки інструкцій є низькорівневою текстовою мовою, яку легко реалізувати і яка застосовується низкою виробників ПЛК, переважно для малих і середніх контролерів. Вона особливо зручна для створення компактних, нескладних програм. Водночас частина виробників відмовляється від IL, віддаючи перевагу більш високорівневій мові структурованого тексту (ST).

Для ілюстрації використання операторів стандарту IEC 61131-3 розглянемо фрагмент програми:

```
LD    A    (*Завантажити значення А*)
AND   B    (*логічне AND з В*)
ST    Q    (*Зберегти вміст акумулятора в Q*)
```

У першому рядку оператором виступає LD, операндом – А, а текст у дужках зі зірочками є коментарем, доданим для пояснення дії й не входить до власне інструкцій, які виконує ПЛК. Таким чином, команда LD А завантажує змінну А у регістр пам'яті (або акумулятор), де вона може бути використана для подальших операцій. У наступному рядку виконується булева операція логічного множення (AND) між А та В. Завершальний рядок передбачає збереження отриманого результату у вихідній змінній Q.

Для позначення різних точок входу до програми можуть використовуватися мітки, що особливо важливо при реалізації переходів (jumps). Такі мітки розташовуються перед інструкцією та відокремлюються двокрапкою. Наприклад:

```
MOTOR_ON: LD M (*Завантажити значення М*)
```

раніше у програмі може бути команда переходу на мітку MOTOR_ON, якщо виконується певна умова.

Особливістю операторів IEC 61131-3 є можливість інверсії значень за допомогою додавання символу N після оператора. Наприклад:

```
LD    A    (*Завантажити значення А*)
ANDN  B    (*Логічне AND NOT В*)
```

оператор ANDN інвертує значення вхідного сигналу В і виконує логічну операцію AND із попереднім значенням.

Мнемоніка кодів мови списку інструкцій

IEC 61131-3	Schneider	Siemens	Mitsubishi	OMRON	Операція	Мова LD
LD	LD	A	LD	LD	Завантаження значення операнду до акумулятора	Рядок старту з розімкнутим контактом
LDN	LDN	AN	LDI	LD NOT	Завантаження інверсного значення операнду до акумулятора	Рядок старту з замкнутим контактом
AND	AND	A	AND	AND	Булеве AND	Послідовне з'єднання контактів
ANDN	ANDN	AN	ANI	AND NOT	Булеве AND з інверсним операндом	Послідовне з'єднання замкнутих контактів
OR	OR	O	OR	OR	Булеве OR	Паралельне з'єднання контактів
ORN	ORN	ON	ORI	OR NOT	Булеве OR з інверсним операндом	Паралельне з'єднання замкнутих контактів
ST	ST	=	OUT	OUT	Збереження значення акумулятора в операнд	Вихід

2.2.1. Програми схем релейно-контактної логіки та списку інструкцій.

В схемах релейно-контактної логіки кожен їх рядок (Rung) повинен починатися з команди «Почати рядок». Для цього можуть застосовуватися оператори LD, або, залежно від виробника, A чи L, які вказують на початок рядка з нормально розімкненими контактами. Якщо ж рядок має починатися з нормально замкнених контактів, використовуються оператори LDN, або ж їхні варіанти LDI, LD NOT, AN, LN (в програмах від інших виробників ПЛК).

Завершення кожного рядка повинно містити інструкцію виходу чи збереження результату. Для цього призначені оператори OUT, або альтернативні позначення ST чи «=».

Для прикладу розглянемо, яким чином у вигляді списку інструкцій заноситься один рядок програми на схемах релейно-контактної логіки з використанням логіки І (AND) (рис. 2.29). При конвертації програми зі схем релейно-контактної логіки до списку інструкцій, використовуючи вбудовану функцію в програмі Machine Expert – Basic від Schneider Electric, відбувається автоматична заміна прямої адресації (наприклад %I0.0) на символічне (наприклад INPUT_A) позначення (за наявності) в програмі списку інструкцій.

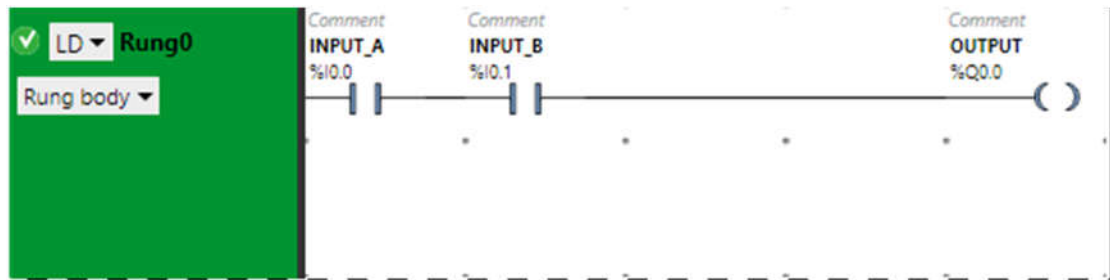


Рис. 2.29. Схема релейно-контактної логіки з використанням логіки І (AND) в програмі Machine Expert – Basic від Schneider Electric

Рядок розпочинається інструкцією LD, що відповідає відкритим контактам. Оскільки у наведеній схемі вхід позначений адресом %I0.0, перший оператор матиме вигляд:

```
LD    %I0.0
```

Далі додається ще один вхід із відкритими контактами, що вимагає застосування інструкції AND із зазначенням адреси елемента. Таким чином, наступний рядок програми:

```
AND   %I0.1
```

Завершується рядок виходом, тож використовується оператор OUT, у нашому випадку з адресом виходу %Q0.0:

```
OUT   %Q0.0
```

Таким чином, уся послідовність команд для одного рядка релейно-контактної програми набуває вигляду (при адресному на символічному позначенні):

```
LD    %I0.0          LD  INPUT_A
AND   %I0.1          AND INPUT_B
ST    %Q0.0          ST  OUTPUT
```

Розглянемо приклад використання логіки АБО (OR) (рис. 2.30).

Програма мовою списку інструкцій для схеми релейно-контактної логіки наведеного рядка, починається з відкритого контакту, що задається командою LD INPUT_A (для символічного адресування). Наступним елементом є паралельне приєднання контактів INPUT_B, для чого використовується команда OR INPUT_B. Завершальним кроком виступає формування вихідного сигналу, яке реалізується оператором ST OUTPUT. Отже, повний перелік інструкцій має вигляд (при адресному на символічному позначенні):

```
LD    %I0.0          LD  INPUT_A
OR    %I0.1          OR  INPUT_B
```

ST %Q0.0 ST OUTPUT

Програма мовою списку інструкцій для схеми релейно-контактної логіки наведеного рядка, починається з нормально замкнених контактів, тому використовується інструкція LDN. У системі команд літера N вказує на інверсію відповідної операції. Наступним кроком є послідовність нормально замкнених контактів, до якої застосовується інструкція ANDN; тут знову суфікс N означає, що інструкція AND реалізується в інверсному вигляді. Крім того, сама літера N може виступати оператором логічного заперечення (NOT).

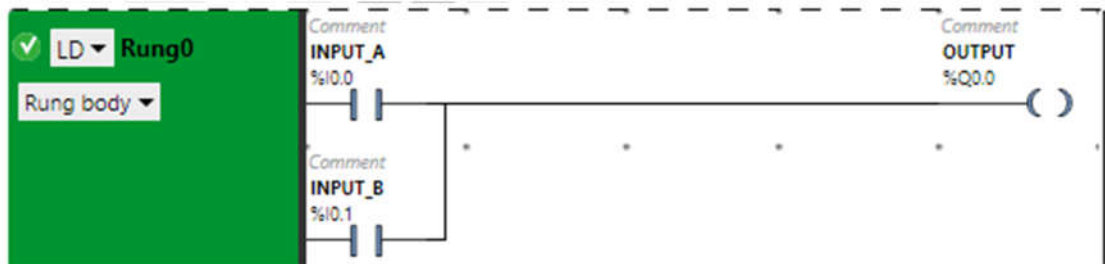


Рис. 2.30. Схема релейно-контактної логіки з використанням логіки АБО (OR) в програмі Machine Expert – Basic від Schneider Electric

Розглянемо приклад використання логіки НІ-АБО (NOR) (рис. 2.31).

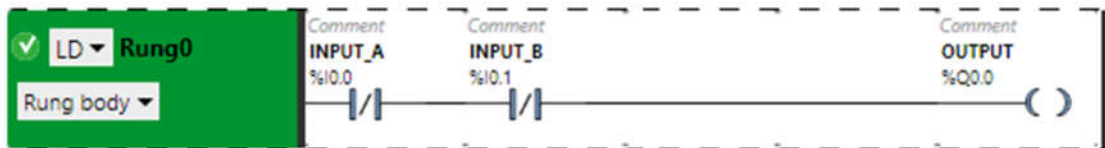


Рис. 2.31. Схема релейно-контактної логіки з використанням логіки НІ-АБО (NOR) в програмі Machine Expert – Basic від Schneider Electric

Таким чином, перелік інструкцій для реалізації драбинного рядка елемента NOR має вигляд (при адресному на символічному позначенні):

LDN	%I0.0	LDN	INPUT_A
ANDN	%I0.1	ANDN	INPUT_B
ST	%Q0.0	ST	OUTPUT

Розглянемо приклад використання логіки НІ-І (NAND) (рис. 2.32).

Програма мовою списку інструкцій для схеми релейно-контактної логіки наведеного рядка, починається з нормально замкнених контактів, тому використовується інструкція LDN. Наступним кроком є паралельне приєднання нормально замкнених контактів, до якого застосовується інструкція ORN; тут знову суфікс N означає, що інструкція OR реалізується в інверсному вигляді.

Таким чином, список інструкцій для реалізації схеми рядку релейно-контактної логіки елемента NOR має вигляд (при адресному на символічному позначенні):

LDN	%I0.0	LDN	INPUT_A
ORN	%I0.1	ORN	INPUT_B
ST	%Q0.0	ST	OUTPUT

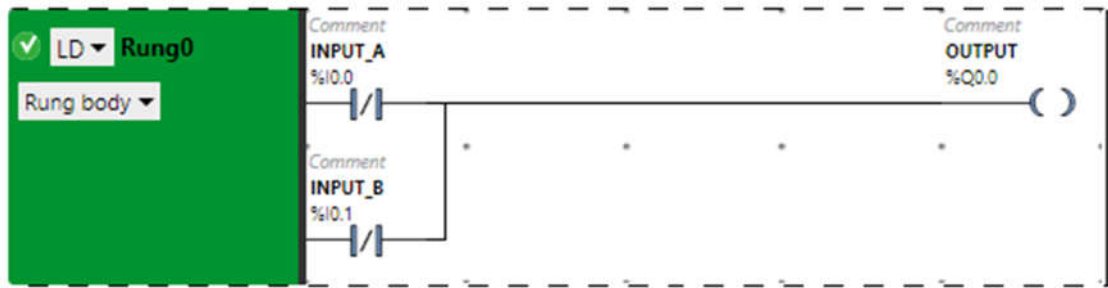


Рис. 2.32. Схема релейно-контактної логіки з використанням логіки NI-I (NAND) в програмі Machine Expert – Basic від Schneider Electric

2.2.2. Розгалужені та багаторядкові структури.

Елемент «Виключне АБО» (XOR), наведений на рис. 3.33, має дві паралельні гілки, кожна з яких містить умову типу AND.

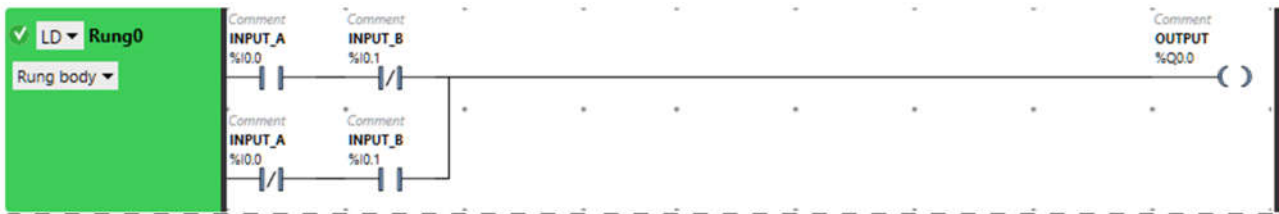


Рис. 2.33. Схема релейно-контактної логіки з використанням логіки «Виключне АБО» (XOR) в програмі Machine Expert – Basic від Schneider Electric

На рис. 2.33 перша інструкція відповідає послідовному з'єднанню нормально відкритого контакту INPUT_A та нормально закритому контакту INPUT_B (отримується інверсією N значення), тобто реалізується логічний елемент АБО. та Наступна паралельна (OR) до першої інструкції виконується, яка відповідає послідовному з'єднанню нормально закритому контакту INPUT_A (отримується інверсією N значення) та нормально відкритого контакту INPUT_B, тобто реалізується логічний елемент АБО. Таким чином, список інструкцій для цього XOR виглядає так:

```
LD    %I0.0
ANDN  %I0.1
OR(N  %I0.0
AND   %I0.1
)
ST    %Q0.0
```

Тут використовуються дужки для позначення того, що певні інструкції мають виконуватися як єдиний блок. Їх застосування аналогічне використанню дужок у математичних виразах. Наприклад, у виразі $(3 + 5)/2$ операція додавання виконується першою, а результат ділиться на 2. У стандарті IEC 61131-3 передбачено саме такий підхід до програмування, де дужки використовуються аналогічно до їх застосування у звичайній арифметиці. Це дає можливість відкладати виконання інструкцій, що знаходяться всередині дужок, аж до

моменту їх закриття. Таким чином, програма у форматі списку інструкцій IES має вигляд:

```
LD    Z
ADD(  A
MUL(  B
AND   C
)
)
```

дає формулу: $Z+(B \cdot (C+D))$.

На рис. 2.34 представлено електричну схему, яка може розглядатися як дві розгалужені групи елементів AND.

Список інструкцій для даної схеми відповідно має такий вигляд:

```
LD    %I0.0
OR    %I0.2
AND(  %I0.1
OR    %I0.3
)
ST    %Q0.0
```



Рис. 2.34. Схема релейно-контактної логіки з використанням логіки з двома гілками елементів АБО (OR) в програмі Machine Expert – Basic від Schneider Electric

На рис. 2.35 представлено приклад схеми релейно-контактної логіки, яка містить два окремі рядки (Rungs). Під час формування списку інструкцій програма записується построчно: інструкції кожного рядка подаються послідовно, одна за одною. Ключовим маркером початку нового рядка є оператор LD та LDN, за допомогою яких контролер (ПЛК) ідентифікує початок нового логічного рядка (гілки). У цьому випадку список інструкцій має такий вигляд:

```
LD    %I0.0
ST    %Q0.0
LDN   %I0.0
ST    %Q0.1
```

Принцип дії системи полягає в наступному: якщо вхід %I0.0 перебуває у неактивному стані, то формується вихідний сигнал на %Q0.1, тоді як вихід %Q0.0 залишається вимкненим. У протилежному випадку, коли %I0.0 активується, на виході %Q0.0 з'являється сигнал, тоді як %Q0.1 переходить у неактивний стан.



Рис. 2.35. Схема релейно-контактної логіки з використанням логіки перемикачів контурів в програмі Machine Expert – Basic від Schneider Electric

2.2.3. Приклади програмування.

Розглянемо декілька прикладів з використанням мов схем релейно-контактної логіки та списку інструкцій.

В першому прикладі передбачається, щоб сигнальна лампа вмикалася у випадку, якщо насос перебуває в роботі та тиск знаходиться в допустимих межах, або ж коли замкнений випробувальний вимикач лампи (рис. 2.36).

Другий приклад стосується клапана, який призначений для підйому вантажу. Клапан повинен спрацювати за умови, що насос працює і при цьому або активовано підйомний вимикач, або ж подається сигнал від іншого вимикача, який підтверджує, що вантаж ще не піднятий і перебуває в нижній позиції свого підйомного каналу (рис. 2.37).

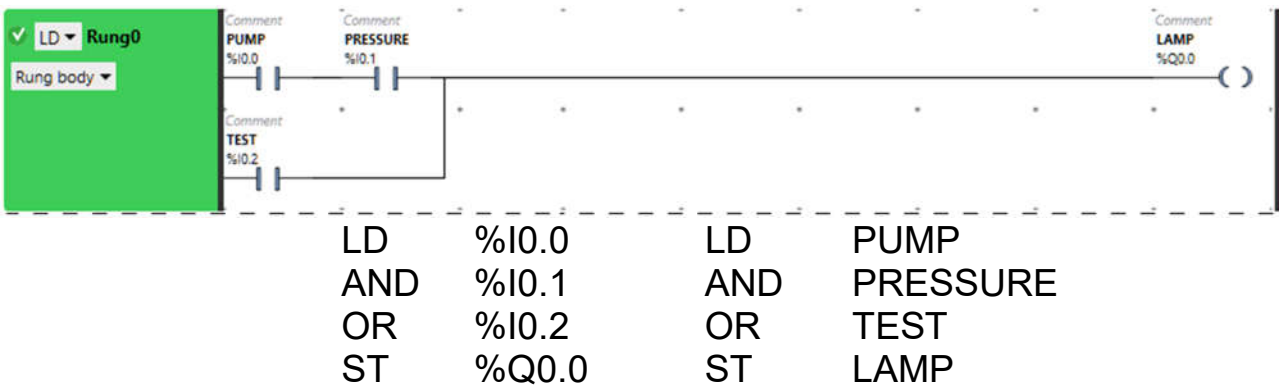


Рис. 2.36. Приклад програми з вмикання сигнальної лампи

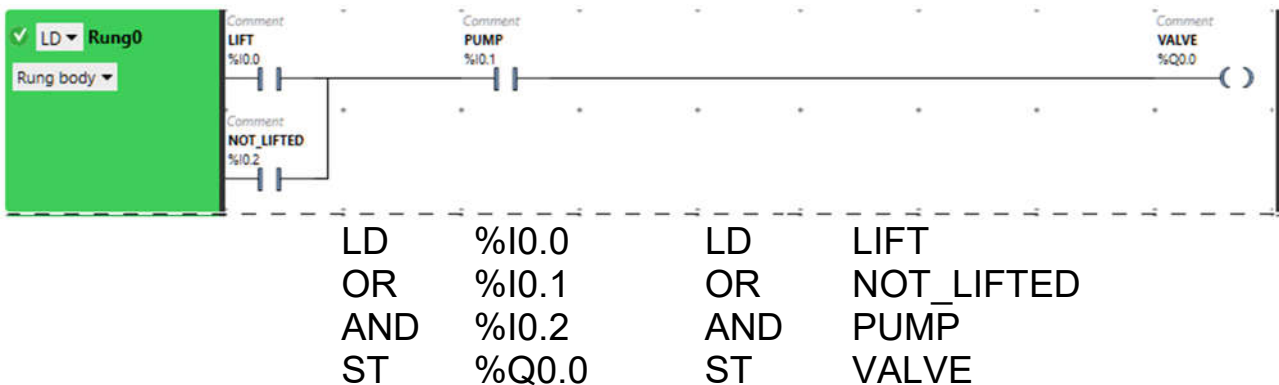


Рис. 2.37. Приклад програми з роботою клапану

В третьому прикладі розглядається система, у якій не повинно бути вихідного сигналу в разі спрацювання будь-якого з чотирьох датчиків. Якщо ж усі датчики не подають сигналів, тоді формується вихід (рис. 2.38).

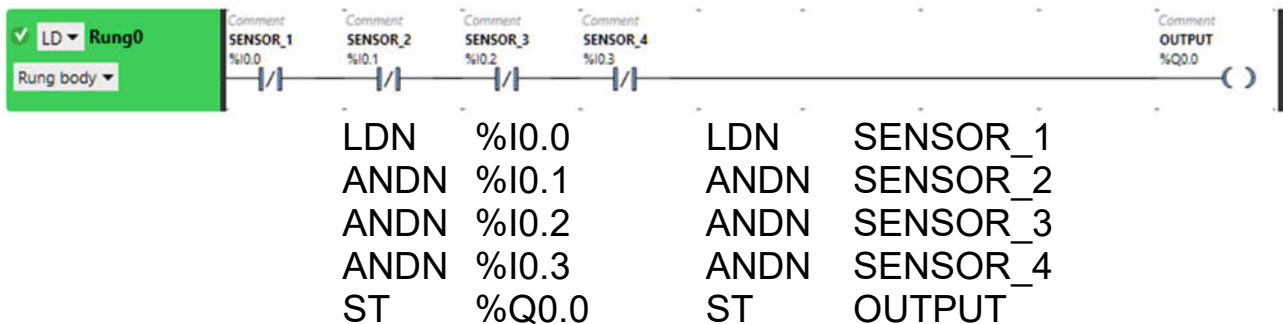


Рис. 2.38. Приклад програми перевірки спрацювання датчиків

2.3. Структурований текст (Structured Text).

Мова програмування Структурований текст – Structured Text (ST) за своїми особливостями значною мірою подібна до мови програмування Pascal [1, 3, 5, 6]. Програма в ній записується у вигляді послідовності операторів, розділених крапкою з комою. Такі оператори використовують заздалегідь визначені інструкції та підпрограми для зміни значень змінних. Під змінними при цьому розуміють заздалегідь задані параметри, внутрішньо збережені значення або ж вхідні та вихідні сигнали.

Для опису зміни значення змінної застосовуються оператори присвоєння. Наприклад, запис

Light := SwitchA;

означає, що змінна Light (сигнальна лампа) набуває нового значення та вмикається чи вимикається залежно від стану SwitchA (вимикача), тобто його ввімкнення або вимкнення.

Загальний формат оператора присвоєння має вигляд:

X := Y;

де Y позначає вираз, який визначає нове значення змінної X, а символ := виступає знаком присвоєння. Змінна зберігає отримане значення доти, доки інший оператор присвоєння не змінить його.

Інші приклади використання мають вигляд:

Light := SwitchA OR SwitchB;

такий запис вказує, що лампа (Light) вмикається у випадку замикання або вимикача A, або вимикача B.

Застосовуючи логічну операцію AND, можемо отримати:

Start := Steam AND Pump;

що означає, що запуск (Start) здійснюється лише тоді, коли одночасно активні Steam і Pump.

У табл. 2.5 наведені деякі оператори (зокрема OR та AND, які використано у наведених вище прикладах), що застосовуються у програмах мовою Structured Text, а також їхня відносна пріоритетність під час обчислення виразів. Для задання потрібної послідовності виконання виразів використовуються дужки.

Наприклад:

```
InputA := 6;  
InputB := 4;  
InputC := 2;  
OutputQ := InputA/3 + InputB/(3 - InputC);
```

Таблиця 2.5

Оператори Structured Text

Оператор	Опис	Пріоритет
(...)	Вираз у дужках (квадратних дужках)	1
Function(...)	Список параметрів функції	2
**	Піднесення до степеню	3
-, NOT	Заперечення, булеве NOT	4
*, /, MOD	Множення, ділення, операція модуля	5
+, -	Додавання, віднімання	6
<, >, <=, >=	Менше, більше, менше або дорівнює, більше або дорівнює	7
=, <>	Рівність, нерівність	8
AND, &	Булеве AND	9
XOR	Булеве XOR	10
OR	Булеве OR	11

У наведеному виразі обчислення відбувається з дотриманням правил пріоритету операцій. Спочатку обчислюється різниця (3 - InputC), і вже її значення використовується як дільник. Таким чином, друга частина інструкції для OutputQ набуває вигляду:

$$4 / (3 - 2) = 4 / 1 = 4.$$

Оскільки операція ділення має вищий пріоритет, ніж додавання, перша частина інструкції виконується перед операцією додавання, тобто:

$$6 / 3 = 2.$$

У підсумку для змінної OutputQ отримуємо значення: $2 + 4 = 6$.

Мова Structured Text є нечутливою до регістру: програміст може застосовувати як малі, так і великі літери, залежно від потреби підвищення читабельності. Аналогічно, пробіли та відступи не є обов'язковими, однак їх використання сприяє кращому сприйняттю коду.

Ідентифікатори змінних, що безпосередньо відображають пам'ять, починаються зі символу %, після якого слідує однолітерний або дволітерний код. Цей код визначає, чи пов'язана пам'ять із входами, виходами або внутрішніми осередками, а також уточнює розрядність (біт, байт, слово тощо). Наприклад:

```
%IX100 (вхідна пам'ять, біт 100),  
%ID200 (вхідна пам'ять, слово 200),  
%QX100 (вихідна пам'ять, біт 100).
```

Перша літера вказує на область пам'яті:

I – входи (Input memory location),

Q – виходи (Output memory location),
M – внутрішня пам'ять (Internal memory).

Друга літера визначає розрядність:

X – біт,
B – байт (8 біт),
W – слово (16 біт),
D – подвійне слово (32 біти),
L – довге слово (64 біти).

Оператор AT використовується для закріплення змінної за конкретною адресою пам'яті. Наприклад:

```
Input1 AT %IX100;
```

в цьому випадку змінна Input1 фіксується у комірці вхідної пам'яті, що відповідає біту з адресою 100.

2.3.1. Умовні оператори.

Оператор IF.

Оператор IF використовується для виконання певних дій залежно від логічного стану змінної. Наприклад:

```
IF fluid_temp THEN
```

означає, що якщо змінна fluid_temp активна, тобто має значення 1, наступні після цього рядка команди в програмі виконуються.

Якщо потрібно діяти за умови, що змінна fluid_temp неактивна (0), застосовується конструкція:

```
IF NOT fluid_temp THEN
```

Коли потрібно перевіряти декілька умов, наприклад, якщо fluid_temp1 або fluid_temp2 активні, використовується логічне OR:

```
IF fluid_temp1 OR fluid_temp2 THEN
```

Оператор IF ... THEN ... ELSE застосовується для вибору різних дій залежно від умов. Наприклад:

```
IF (Limit_switch1 AND Workpiece_Present) THEN
```

```
Gate1 := Open;
```

```
Gate2 := Close;
```

```
ELSE
```

```
Gate1 := Close;
```

```
Gate2 := Open;
```

```
END_IF;
```

У цьому випадку, якщо одночасно активні Limit_switch1 та Workpiece_Present, Gate1 відкривається, а Gate2 закривається. В іншому випадку Gate1 закривається, а Gate2 відкривається. Зауважимо, що кінець оператора IF обов'язково позначається командою END_IF.

З прикладом використання адрес ПЛК:

```
IF (%IX0.0 = 1) THEN
```

```
%QX0.0 := 1;
```

```
ELSE
```

```
%QX0.1 := 0;
```

END_IF;

Якщо на вході %IX0.0 з'являється сигнал 1, на виході QX0.0 встановлюється 1; у протилежному випадку 0. Використання прямого адресування для звернення в операторах можливо але програмне забезпечення буде рекомендувати використовувати символну адресацію, як більш зрозумілішу при розробці та редагуванні програм, бо можливий ризик руйнування обладнання, травм персоналу та його загибелі.

Оператор CASE.

Оператор CASE дозволяє виконувати різні дії залежно від цілочисельного значення змінної. Наприклад, для керування температурою можна записати:

```
CASE (Temperature) OF
    0..40: Furnace_switch := 1;
    41..100: Furnace_switch := 0;
ELSE
    Furnace_switch := 0;
END_CASE;
```

У цьому випадку залежно від значення Temperature виконується відповідна дія: включення або відключення перемикача печі. Як і в операторі IF, кінець CASE обов'язково позначається END_CASE.

Інший приклад використання для керування швидкістю двигуна з вентиляторами:

```
CASE speed_setting OF
    1: speed := 5;
    2: speed = 10;
    3: speed = 15; fan1 = ON;
    4: speed = 20; fan2 = ON;
ELSE
    speed = 0; speed_fault = TRUE;
END_CASE;
```

У цьому прикладі змінна speed_setting визначає швидкість двигуна, а також включає відповідні вентилятори при певних значеннях. У випадку, якщо значення не входить у визначений діапазон, швидкість встановлюється на 0, а сигнал помилки (speed_fault) активується.

2.3.2. Оператори циклу.

Оператори циклу застосовуються у випадках, коли необхідно багаторазово повторювати одну чи кілька інструкцій залежно від значення певної змінної.

Оператор FOR ... DO дозволяє виконувати серію команд визначену кількість разів, використовуючи цілочисельну змінну-лічильник. Наприклад:

```
FOR Input := 10 TO 0 BY -1
DO
    Output := Input;
END_FOR;
```

У цьому випадку значення Output зменшується на 1 на кожному кроці, відповідно до змінної Input, яка поступово змінюється від 10 до 0 з кроком -1.

Оператор WHILE ... DO дозволяє виконувати серію інструкцій доти, доки заданий булевий вираз залишається істинним. Наприклад:

```
OutputQ := 0;
WHILE InputA AND InputB
DO
OutputQ := OutputQ + 1;
END_WHILE;
```

У цьому випадку змінна OutputQ щоразу збільшується на 1, поки обидва вхідні сигнали InputA та InputB залишаються активними (в програмі необхідно передбачити умову зупинки для уникання помилки переповнення значення).

Оператор REPEAT ... UNTIL забезпечує виконання серії інструкцій принаймні один раз, після чого перевіряється умова завершення циклу. Інструкції повторюються доти, доки логічний вираз залишається істинним. Приклад:

```
OutputQ := 0;
REPEAT
OutputQ := OutputQ + 1;
UNTIL (Input1 = Off) OR (OutputQ > 5)
END_REPEAT;
```

У цьому прикладі змінна OutputQ поступово збільшується, доки Input1 не вимкнено або доки значення OutputQ не перевищить 5.

Ці три оператори FOR, WHILE та REPEAT ... UNTIL надають гнучкі можливості для організації циклічних процесів у мові структурованого тексту, що є особливо важливим для моделювання повторюваних операцій у системах керування технологічними процесами в промисловості.

2.3.3. Оголошення типів, змінних та функціональних блоків у мові структурованого тексту.

Перед початком написання програми можна визначити користувацькі типи даних, які використовуватимуться для представлення змінних. Це можуть бути як перелічувані типи, так і числові. Наприклад:

```
TYPE Motor: (Stopped, Running); (*Мотор: зупинений або працює*)
END_TYPE
```

```
TYPE Valve: (Open, Shut); (*Клапан: відкритий або закритий*)
END_TYPE
```

```
TYPE Pressure: REAL; (*Тиск як аналогова величина*)
END_TYPE
```

Далі оголошуються змінні – вхідні та вихідні сигнали, що надходять від датчиків чи передаються на виконавчі механізми. Наприклад:

```
VAR_IN (*Вхідні змінні*)
PumpFault : BOOL; (*Сигнал несправності насоса - логічна змінна*)
END_VAR;
```

```
VAR_OUT (*Вихідні змінні*)
Motor_speed : REAL; (*Швидкість двигуна - дійсне число*)
END_VAR;
```

```
VAR_IN
Value : INT; (*Цілочисельна змінна*)
END_VAR;
```

Також на цьому етапі можуть призначатися початкові значення для змінних, наприклад:

```
VAR
Temp : REAL := 100; (*Початкове значення дорівнює 100*)
END_VAR;
```

Лише після цього переходять до формування інструкційної частини програми.

Розглянемо структуру функціонального блоку на прикладі. Цей функціональний блок може бути складовою більш складної програми та використовується для перевірки рівнів напруги. Окремо визначаються вхідні та вихідні параметри та їх тип. Програма складається з типових команд мови ST. В цьому прикладі, якщо будь-яке зі значень VOLTS1, VOLTS2 чи VOLTS3 перевищує 12, вихідна змінна OVERVOLTS встановлюється у стан TRUE, а оператор RETURN забезпечує дострокове завершення виконання функціонального блоку. У головній програмі встановлення змінної OVERVOLTS = TRUE може використовуватися як сигнал для ініціювання захисних або аварійних дій (наприклад, вимкнення навантаження чи подачі сигналу тривоги).

```
FUNCTION_BLOCK TEST_VOLTAGE
VAR_INPUT
    VOLTS1, VOLTS2, VOLTS3 : REAL;
END_VAR
VAR_OUTPUT
    OVERVOLTS : BOOL;
END_VAR
(* Текст програми функціонального блоку*)
IF VOLTS1 > 12 THEN
    OVERVOLTS := TRUE; RETURN;
END_IF;
IF VOLTS2 > 12 THEN
    OVERVOLTS := TRUE; RETURN;
END_IF;
IF VOLTS3 > 12 THEN
    OVERVOLTS := TRUE;
END_IF;
```

2.3.4. Приклади програмування.

Розглянемо приклад в якому автоматичне керування температурою повітря в камері, яка обладнана нагрівачем та вентилятором. Потрібно підтримувати температуру в межах 18...22 °С. Якщо температура повітря знижуються менше за 18 °С, вмикається нагрівач. Якщо температура повітря в камері вища за 22 °С, вмикається вентилятор для охолодження. Коли температура повітря в камері знаходиться в межах норми, виконавчі механізми вимкнено (нагрівач і вентилятор).

Розглянемо алгоритм роботи системи автоматичного керування температурою повітря в камері:

Зчитати температуру з аналогового датчика (REAL).

Якщо $T < 18$ присвоюємо змінній Heater = TRUE.

Якщо $T > 22$ присвоюємо змінній Fan = TRUE.

Інакше присвоюємо змінним Heater = FALSE, Fan = FALSE.

Розробимо програму на мові ST для системи автоматичного керування температурою повітря в камері.

```
VAR
    TempSensor : REAL; (* температура в камері *)
    Heater : BOOL; (* нагрівач *)
    Fan : BOOL; (* вентилятор *)
END_VAR
(* Текст програми *)
IF TempSensor < 18.0 THEN
    Heater := TRUE;
    Fan := FALSE;
ELSIF TempSensor > 22.0 THEN
    Heater := FALSE;
    Fan := TRUE;
ELSE
    Heater := FALSE;
    Fan := FALSE;
END_IF;
```

В іншому прикладі при циклічному дозуванні рідини використовуємо оператор CASE. Для дозування рідини використовується клапан, який повинен 10 разів відкритися і закритися для подачі порцій рідини. Після 10 циклів подача завершується.

Розглянемо алгоритм роботи системи циклічного дозування рідини:

На першому кроці запускаємо таймер на 2 секунди.

Далі чекаємо закінчення таймера ($Trg.Q = TRUE$), тоді закриваємо клапан, інкрементуємо та переходимо до наступної ітерації.

Коли $i > 10$, встановлюється $Done := TRUE$, і цикл завершується.

Розробимо програму на мові ST для системи циклічного дозування рідини:

```
VAR
    i : INT := 1; (* лічильник ітерацій *)
    Valve : BOOL := FALSE; (* стан клапана *)
```

```

    Done : BOOL := FALSE; (* прапорець завершення *)
    Step : INT := 0; (* крок алгоритму *)
    Tmr : TON; (* таймер *)
END_VAR
(* Текст програми *)
CASE Step OF
    0: (* Початковий стан *)
        IF i <= 10 THEN
            Valve := TRUE; (* відкрили клапан *)
            Tmr(IN := TRUE, PT := T#2s); (* таймер на 2 сек *)
            Step := 1;
        ELSE
            Done := TRUE; (* завершення циклу *)
            Valve := FALSE;
            Step := 99;
        END_IF;
    1: (* Чекаємо, поки таймер відпрацює *)
        Tmr(IN := TRUE, PT := T#2s);
        IF Tmr.Q THEN
            Valve := FALSE; (* закрили клапан *)
            Tmr(IN := FALSE); (* скинули таймер *)
            i := i + 1; (* наступна ітерація *)
            Step := 0; (* назад на початок *)
        END_IF;
    99: (* Кінець циклу, нічого не робимо *)
END_CASE;

```

Розглянемо приклад створення функціональних блоків. В прикладі необхідно контролювати рівень в двох баках. Коли в обох баках рівень нижче за мінімальний включається насос. Якщо в будь-якому баку рівень досяг максимального насос зупиняється. Для цього створимо функціональний блок.

Розглянемо алгоритм роботи контролю рівня в баках:

Зчитати рівні Tank1 та Tank2 (REAL).

Якщо Tank1 < Min AND Tank2 < Min то Pump = TRUE.

Якщо Tank1 ≥ Max OR Tank2 ≥ Max то Pump = FALSE.

Розробимо функціональний блок на мові ST контролю рівня в баках.

```
FUNCTION_BLOCK TANK_CONTROL
```

```
VAR_INPUT
```

```
    Tank1 : REAL; (* поточний рівень у баку 1 *)
```

```
    Tank2 : REAL; (* поточний рівень у баку 2 *)
```

```
    MinLevel : REAL;
```

```
    MaxLevel : REAL;
```

```
END_VAR
```

```
VAR_OUTPUT
```

```
    Pump : BOOL;
```

```
END_VAR
```

```
(* Текст програми функціонального блоку*)
IF (Tank1 < MinLevel) AND (Tank2 < MinLevel) THEN
    Pump := TRUE;
ELSIF (Tank1 >= MaxLevel) OR (Tank2 >= MaxLevel) THEN
    Pump := FALSE;
END_IF;
END_FUNCTION_BLOCK
```

Для керування технологічними об'єктами з розподіленими параметрами (наприклад температура повітря в теплиці) необхідно робити усереднення показів масиву датчиків температури. Розглянемо камеру в якій встановлено 5 датчиків температури. Щоб уникнути впливу випадкових шумів та похибок одного датчика, потрібно: зібрати значення усіх датчиків у масив; обчислити середнє значення; приймати рішення про вмикання/вимикання вентиляування/нагріву за цим середнім значенням.

Розглянемо алгоритм роботи САК температурою повітря в камері. Спочатку оголошуємо масив TempSensors[1..5] для зберігання температур. Використовуючи оператор FOR для елементів масиву підсумовуємо їх значення та обчислюємо середнє значення: Average := Sum / 5. Проводимо порівняння з уставками: якщо Average > 25 °C то увімкнути вентилятор (Fan = TRUE) або Average < 18 °C то увімкнути нагрівач (Heater = TRUE). Коли температура повітря знаходиться в заданому діапазоні обидва пристрої вимкнені.

Розробимо програму на мові ST для САК температурою повітря в камері.

```
VAR
    TempSensors : ARRAY[1..5] OF REAL; (* масив з 5 датчиків
    температури *)
    i : INT; (* лічильник циклу *)
    Sum : REAL := 0.0; (* сума значень *)
    Average : REAL := 0.0; (* середня температура *)
    Fan : BOOL := FALSE; (* вентилятор *)
    Heater : BOOL := FALSE; (* нагрівач *)
```

```
END_VAR
```

```
(* Текст програми *)
(* Розрахунок середньої температури *)
Sum := 0.0;
FOR i := 1 TO 5 DO
    Sum := Sum + TempSensors[i];
END_FOR;
Average := Sum / 5.0;
(* Прийняття рішення *)
IF Average > 25.0 THEN
    Fan := TRUE;
    Heater := FALSE;
ELSIF Average < 18.0 THEN
    Heater := TRUE;
    Fan := FALSE;
```

ELSE

Heater := FALSE;

Fan := FALSE;

END_IF;

У цьому прикладі використано: ARRAY для зберігання кількох сигналів; FOR-цикл для підсумовування; умовні оператори для логіки вмикання/вимикання обладнання.

Розглянемо приклад холодильного складу з контролем температури повітря у зонованому складі (2D ARRAY). Холодильний склад розділено на 3 ряди та 4 зони у кожному ряду (усього 12 датчиків температури).

Завдання САК:

- опитати всі датчики, які зберігаються у двовимірному масиві Temp[1..3, 1..4];
- знайти середню температуру по кожному ряду;
- знайти максимальну температуру серед усіх датчиків;
- увімкнути охолодження, якщо середня температура в будь-якому ряду > 5 °C або максимальна > 8 °C.

Алгоритм роботи САК процесом регулювання температурою повітря на холодильному складі: оголошуємо двовимірний масив Temp[Row, Zone]; ініціалізуємо змінні: SumRow, AvgRow, MaxTemp; виконуємо подвійний цикл FOR для проходження по рядках та зонах; для кожного ряду підраховуємо середнє значення температури повітря; знаходимо максимальне значення серед усіх датчиків та визначаємо стан охолодження (Cooling := TRUE/FALSE).

Розробимо програму на мові ST для САК процесом регулювання температурою повітря на холодильному складі.

VAR

Temp : ARRAY[1..3, 1..4] OF REAL; (* масив температур: 3 ряди по 4 зони *)

Row, Zone : INT;

SumRow : REAL;

AvgRow : ARRAY[1..3] OF REAL; (* середня температура по рядках *)

MaxTemp : REAL := -999.0; (* початково мінімальне значення *)

Cooling : BOOL := FALSE;

END_VAR

(* Текст програми *)

(* Обчислення середніх та пошук максимальної температури *)

FOR Row := 1 TO 3 DO

SumRow := 0.0;

FOR Zone := 1 TO 4 DO

SumRow := SumRow + Temp[Row, Zone];

(* Пошук максимуму *)

IF Temp[Row, Zone] > MaxTemp THEN

MaxTemp := Temp[Row, Zone];

END_IF;

```

    END_FOR;
    AvgRow[Row] := SumRow / 4.0;
END_FOR;
(* Логіка керування охолодженням *)
Cooling := FALSE;
FOR Row := 1 TO 3 DO
    IF AvgRow[Row] > 5.0 THEN
        Cooling := TRUE;
    END_IF;
END_FOR;
IF MaxTemp > 8.0 THEN
    Cooling := TRUE;
END_IF;

```

У цьому прикладі використано: двовимірний масив Temp[Row, Zone]; масив середніх значень AvgRow; подвійний цикл FOR; умовні оператори для логіки охолодження.

2.4. Функціональні блокові діаграми (Function Block Diagram).

Термін «Функціональні блокові діаграми» (FBD) застосовується для позначення програм ПЛК, що описуються у вигляді графічних блоків [1, 5, 6]. Це графічна мова для відображення потоків сигналів та даних через блоки, які є багаторазово використовуваними програмними елементами. Функціональний блок це програмна інструкційна одиниця, яка при виконанні формує одне або кілька вихідних значень. Блок зображується у вигляді прямокутника з назвою функції всередині (рис. 2.39).

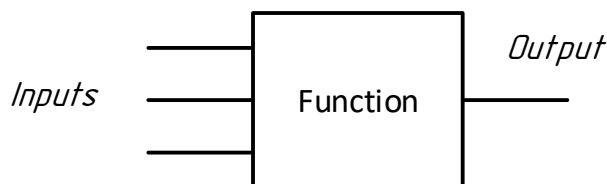


Рис. 2.39. Функціональний блок

Стандарт ІЕС 61131-3 для побудови таких блоків наведено на рис. 2.40. Вхідні сигнали подаються ліворуч, вихідні праворуч. У прямокутнику зазначається тип функціонального блоку, наприклад AND, а вище його системне ім'я, наприклад Timer1. Позначення входів і виходів вказуються біля відповідних точок на блоці. Для уникнення перевантаження схеми застосовуються міждіаграмні з'єднувачі, які показують використання виходу в іншій частині схеми. Функціональні блоки можуть мати стандартні функції (логічні елементи, лічильники, таймери) або визначатися користувачем, наприклад блок для обчислення середнього значення сигналів.

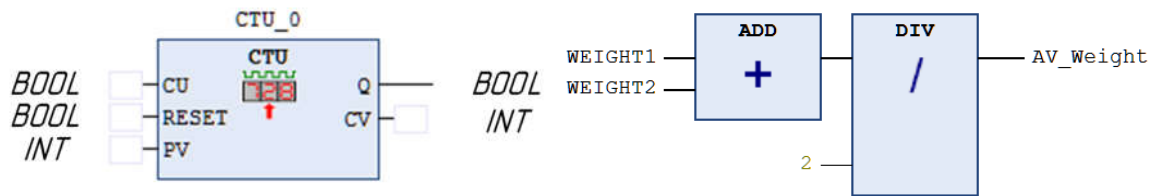


Рис. 2.40. Представлення функціональних блоків

2.4.1. Логічні елементи.

У програмах часто застосовуються логічні елементи. Використовуються дві системи стандартних графічних символів: американська та міжнародна (IEEE/ANSI), де логічний елемент позначається прямокутником із зазначенням функції всередині. Позначка «1» у прямокутнику вказує на наявність вихідного сигналу при вході «1». Функція OR позначається як ≥ 1 , оскільки вихід активний, якщо будь-який вхід дорівнює 1. Інверсія входу позначається маленьким кружком на вході, інверсія виходу – кружком на виході (рис. 2.41). В мові FBD зазвичай використовується позначення за системою IEEE/ANSI (рис. 2.42).

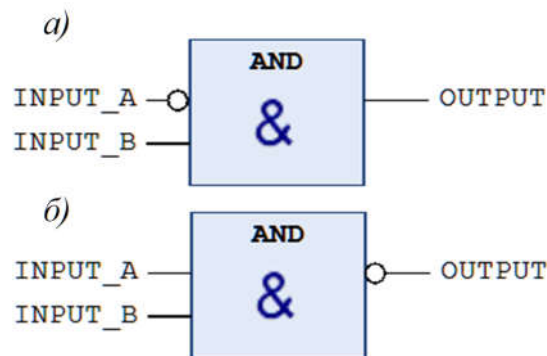


Рис. 2.41. Логічне ТА з одним інверсним входом (а) та інверсним виходом (б)

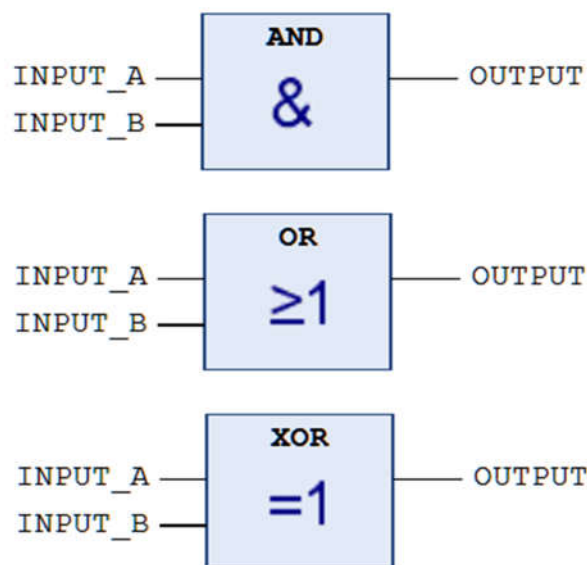


Рис. 2.42. Елементи логіки в програмі Machine Expert від Schneider Electric

2.4.2. Зв'язок та порядок обробки елементів.

Зв'язки це лінії які зв'язують між собою елементи мови FBD. Виходи елементів мови FBD можуть мати кілька зв'язків. Зв'язки можуть з'єднуватись та місце такого з'єднання позначаються жирною крапкою. Входи/виходи, які будуть з'єднані, повинні мати відповідні типи даних. Зв'язки можуть бути відредаговані в режимі "Вибору". Дозволяється перекриття зв'язків іншими об'єктами. Зв'язки не можуть використовуватися для конфігурації контурів, тому що неможливо ясно визначити порядок обробки в секції. Контури повинні вирішуватися за допомогою фактичних параметрів.

Якщо вхід елемента мови FBD невизначено то його значення за замовчуванням призначається рівним "0".

В першу чергу порядок обробки визначається розміщенням елементів мови FBD. Якщо елементи мови FBD згодом з'єднуються за допомогою графічних зв'язків, порядок обробки визначається потоком даних (рис. 2.44).

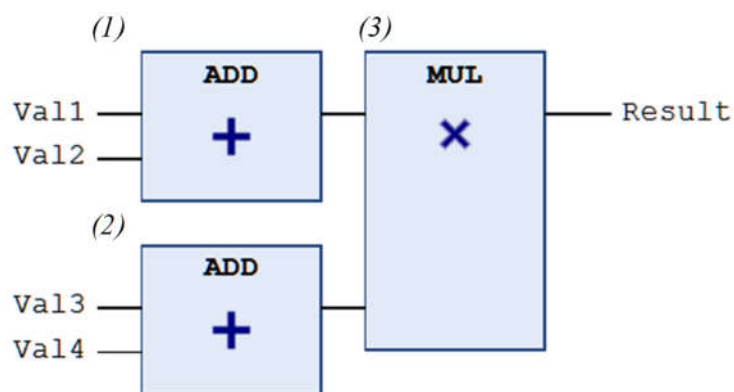


Рис. 2.43. Порядок обробки елементів мови FBD в програмі Machine Expert від Schneider Electric

Елементи мови FBD всередині однієї секції виконуються згідно зумовленої порядку та використовують фактичні параметри на входах і яким не були розподілені ніякі значення, працюють з початковими значеннями цих фактичних параметрів.

Для створення замкнутого контуру елементів мови FBD використовується завдання фактичних параметрів (рис. 2.44).

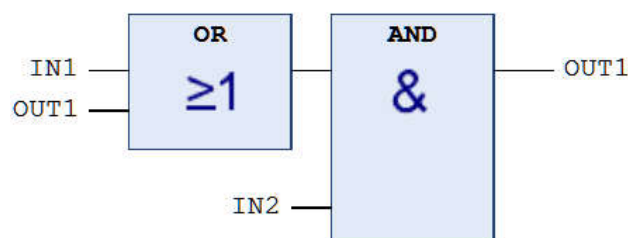


Рис. 2.44. Замкнутий контур обробки елементів мови FBD в програмі Machine Expert від Schneider Electric

2.4.3. Приклади програмування.

Для прикладу, потрібно керувати насосом, який вмикається і подає рідину в бак, якщо замкнено пусковий вимикач, рівень рідини в баку нижчий за необхідний і є рідина в резервуарі. Це задача вирішується з застосуванням логічного елемента AND між сигналом пускового вимикача та датчиком, який спрацьовує при низькому рівні рідини. Додатково враховується перемикач, що показує наявність рідини в резервуарі. Якщо є рідина, цей перемикач дає сигнал на вхід (рис. 2.45). Відповідні FBD та LD наведені на рис. 5.31.

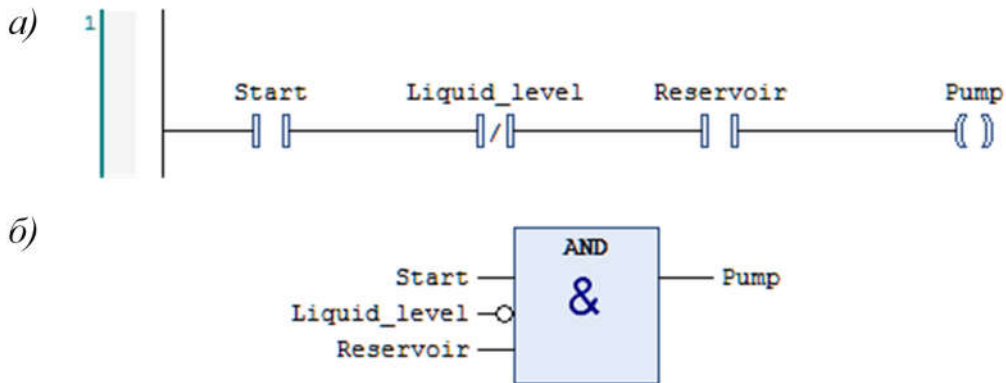


Рис. 2.45. Програма керування насосом на мові LD (а) та FBD (б) в програмі Machine Expert від Schneider Electric

В наступному прикладі необхідно, щоб сигнальна лампа вмикалася у випадку, якщо насос працює та тиск знаходиться у допустимих межах, або якщо замкнено вимикач перевірки лампи. Для сигналів від насоса та датчика тиску маємо ситуацію логіки AND, оскільки обидва умови необхідні для появи вихідного сигналу лампи. Проте, з урахуванням вимикача перевірки, маємо логіку OR, адже при його замиканні лампа повинна засвітитися незалежно від сигналу від системи AND. Відповідні функціональна блокова діаграма та схема релейно-контактної логіки показані на рис. 2.46.

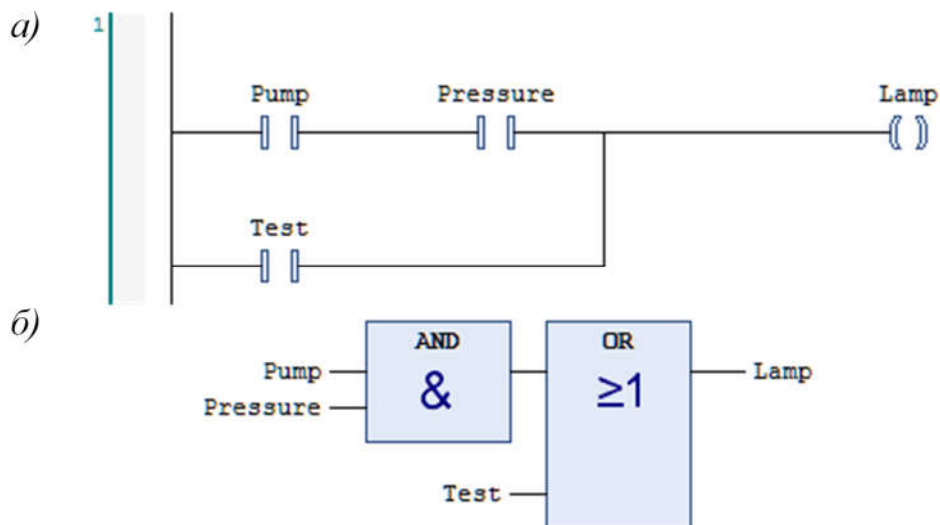


Рис. 2.45. Програма керування сигналізацією на мові LD (а) та FBD (б) в програмі Machine Expert від Schneider Electric

Розглянемо приклад, коли клапан має бути задіяний для підйому вантажу за умови, що насос працює, і при цьому спрацьовує або перемикач підйому, або перемикач, що сигналізує про те, що вантаж ще не піднятий і знаходиться внизу підйомного каналу. Тут маємо ситуацію логіки OR для двох перемикачів та логіку AND, яка об'єднує обидва перемикачі разом із насосом. Можливий варіант програми наведено на рис. 2.46.

В наступному прикладі розроблено програму керування мікрокліматом теплиці з урахуванням наступних контролю температури та вологості повітря. Контроль температури повітря здійснюється за мінімальним ($T_{min}=18\text{ }^{\circ}\text{C}$) і максимальними ($T_{max}=28\text{ }^{\circ}\text{C}$) значеннями при перевищенні яких вмикається або нагрівач або вентилятор. Якщо вологість нижча за нижній поріг ($H_{min}=40\%$) вмикається зволожувач. При спрацюванні аварійного сигналу ($Emergency=TRUE$) всі виконавчі механізми (обігрівач, вентилятор, зволожувач) вимикаються.

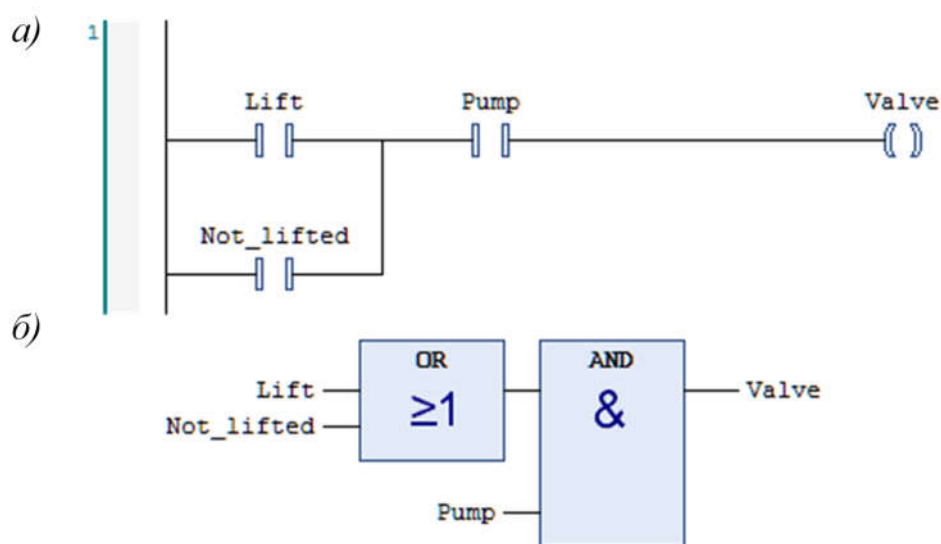


Рис. 2.46. Програма керування клапаном на мові LD (а) та FBD (б) в програмі Machine Expert від Schneider Electric

2.5. Послідовні функціональні схеми (Sequential Function Charts).

Послідовні функціональні схеми (Sequential Function Chart (SFC)) з'явилася в 1979 році як мова, що дозволяє ясно й однозначно визначити бажану поведінку системи управління [1, 5, 6]. Відрізняється строгим математичним визначенням, кожний операційний стан системи може бути декодовано з дуже невеликим аналізом.

Мова SFC зручна для опису, як послідовних процесів, так і пакетних або паралельних процесів, легко комбінується з іншими мовами (мова специфікацій), володіє розвиненими механізмами синхронізації, має прості динамічні правила. Якщо необхідно описати послідовність перемикання світлофора «червоний – зелений», одним зі способів може бути представлення цього процесу як послідовності функцій або станів наприклад, стан червоного сигналу та стан

зеленого сигналу разом із відповідними входами та виходами кожного стану (рис. 2.48).

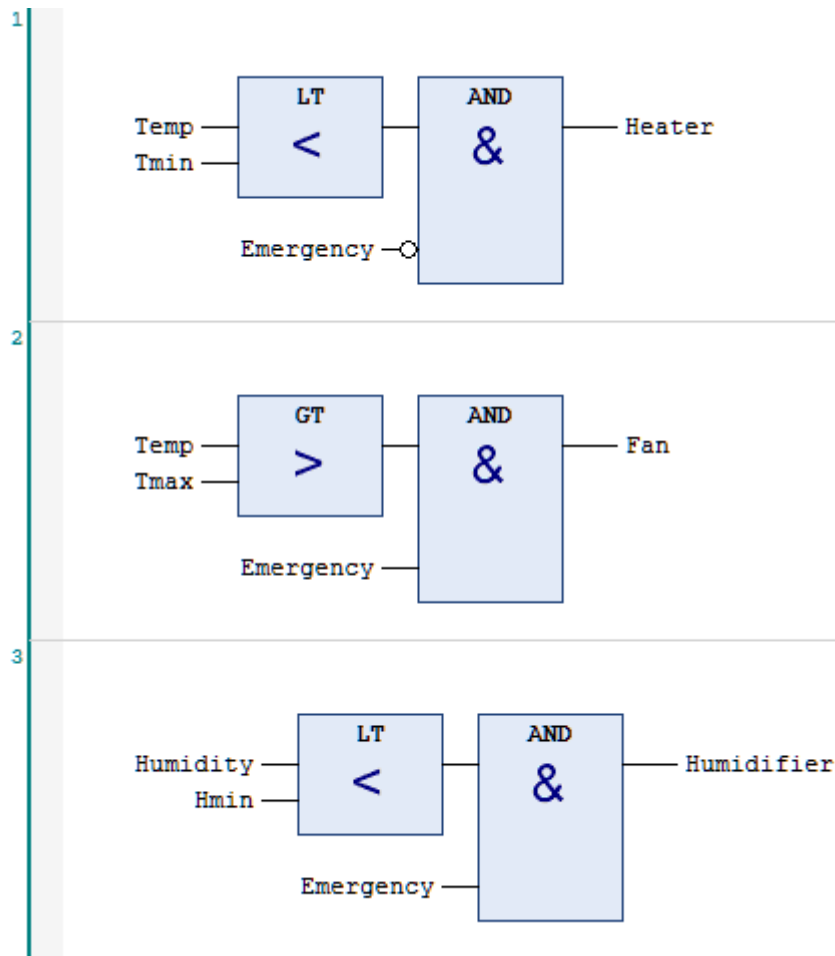


Рис. 2.47. Програма керування мікрокліматом в теплиці на мові FBD в програмі Machine Expert від Schneider Electric

Стан 0 має вхід, який активується після того, як зелений сигнал горів протягом 1 хвилини, та вихід у вигляді включення червоного сигналу, тобто умова переходу від зеленого сигналу задається часовим інтервалом 1 хвилина. Стан 1 має вхід, який активується після того, як червоний сигнал горів 1 хвилину, і вихід у вигляді включення зеленого сигналу, тобто умова переходу від червоного сигналу також 1 хвилина. Після того, як зелений сигнал горів 1 хвилину, відбувається перехід назад до стану 0.

Мова SFC включає такі елементи:

1. Робота описується низкою окремих послідовно з'єднаних станів (кроків), що позначаються прямокутниками. Кожний стан відображає певний режим роботи керованої системи, у якому виконується деяка дія. Початковий крок програми виділяється подвійними лініями, що відрізняє його від інших (рис. 2.49).
2. Кожна лінія з'єднання між станами має горизонтальну риску, яка позначає умову переходу, що повинна бути виконана для перемикання системи з одного стану до іншого. Два стани не можуть бути з'єднані

безпосередньо їх завжди розділяє умова переходу. Так само дві умови переходу не можуть іти підряд безпосередньо між ними завжди має бути крок.

3. Програма постійно перевіряє умови переходу, і щойно вони виконуються, відбувається перехід до наступного стану.
4. Процес триває від одного стану до іншого доти, доки не буде завершений повний робочий цикл машини.
5. Виходи або дії для кожного стану/кроку зображуються прямокутниками, розташованими горизонтально поруч із кроком, і реалізуються тоді, коли даний стан активний. Це можуть бути, наприклад, «Очікувати 1 хвилину», «Відкрити клапан 1», «Закрити клапан 1» тощо.

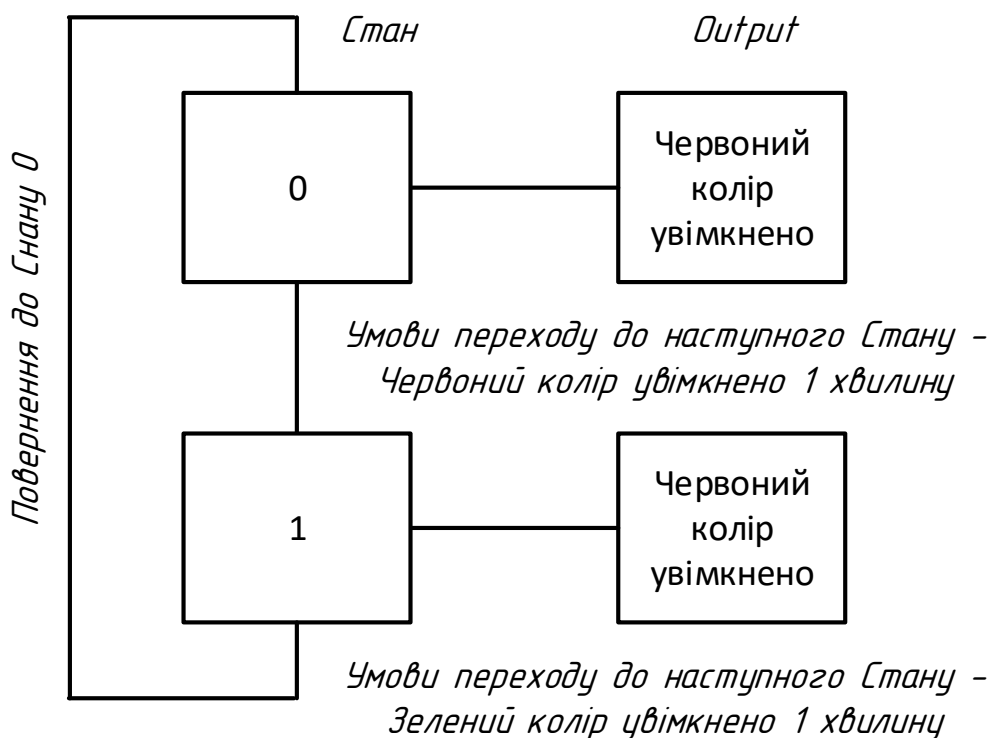


Рис. 2.48. Послідовність перемикання світлофора

Розглянемо приклад програми на мові SFC та її еквівалент у вигляді програми на мові LD (рис. 2.50). Програма запускається при спрацюванні входу IN 1, що активує вихід кроку 1. Після завершення дії цього виходу настає перехід і активується крок 2 із виходом 2. По його завершенні виконується умова переходу, яка веде до кінцевого кроку End.

Як приклад принципів SFC можна розглянути частину циклу роботи бункера активного вентилявання зерна. На першому етапі бункер заповнюється зерном. Коли бункер повний, вмикається вентилятор, який працює до досягнення заданої температури зерна. Потім зерно вивантажується з бункера. Послідовність таких станів подана на рис. 2.51.

Перехід між кроками визначається умовами, які мають бути виконані, щоб програма виконувалась далі. До моменту їхнього виконання програма продовжує виконувати поточний крок. Прикладами умов можуть бути:

Temperature > 50 Pump = On
 Valve_Open AND Compressor_On

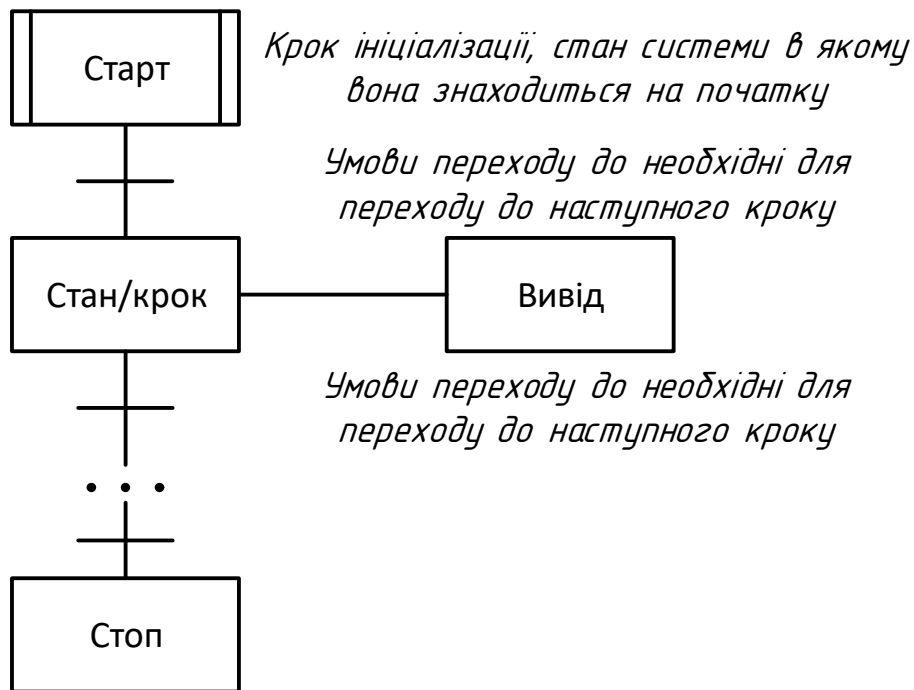


Рис. 2.49. Елементи мови SFC

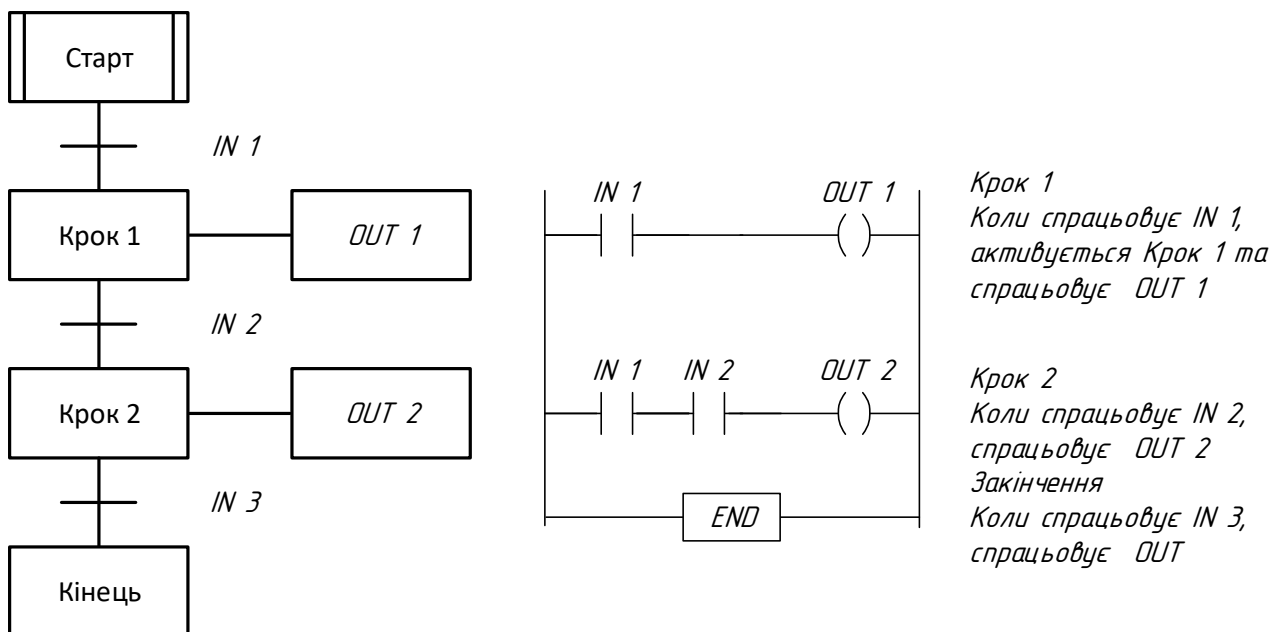


Рис. 2.50. Еквівалентна програма на мовах SFC та LD

При програмуванні такі переходи можуть бути задані через логічні (Boolean) вирази у мові ST, які перевіряють істинність умови (TRUE, FALSE). Як альтернатива, можна викликати підпрограму для перевірки умови; якщо вона виконується, повертається значення 1 (TRUE), інакше 0 (FALSE). В кінці підпрограми використовується інструкція End of Transition (EOT) для встановлення стану переходу у відповідне логічне значення.

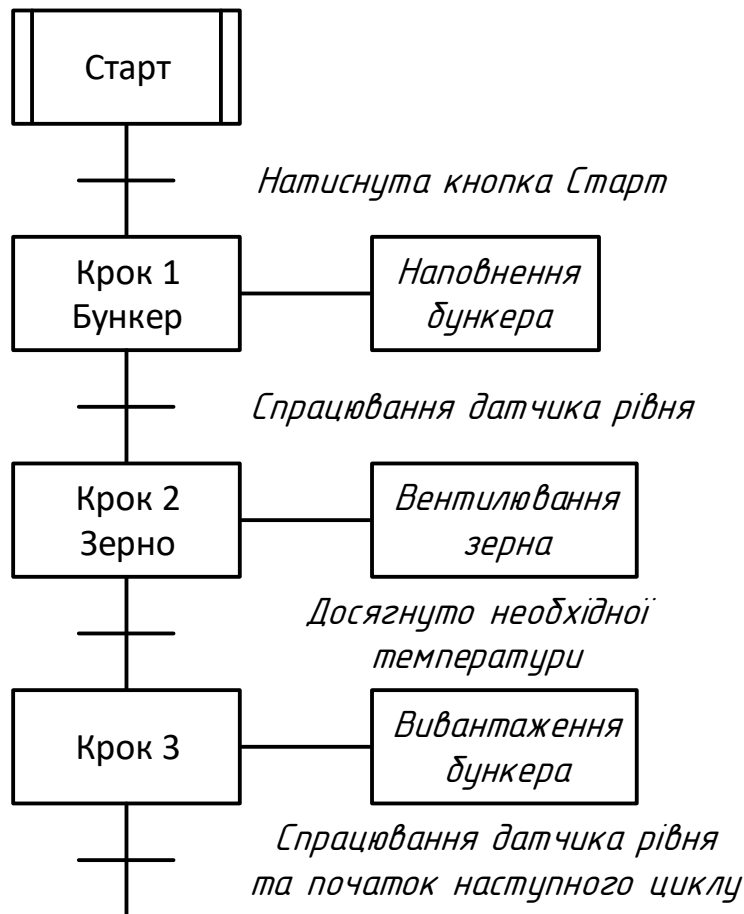


Рис. 2.51. Програма на мови SFC

Дії додаються до кроків і вказують, які функції виконує даний крок, наприклад: увімкнення двигуна, відкриття клапана чи виклик підпрограми.

Мова Sequential Function Chart (SFC) є потужною графічною технікою опису послідовної поведінки програми. Графічні мови цього типу застосовуються вже багато років, зокрема Grafset, що використовується у Європі. Стандарт ІЕС 61131-1 відтворює більшість принципів Grafset.

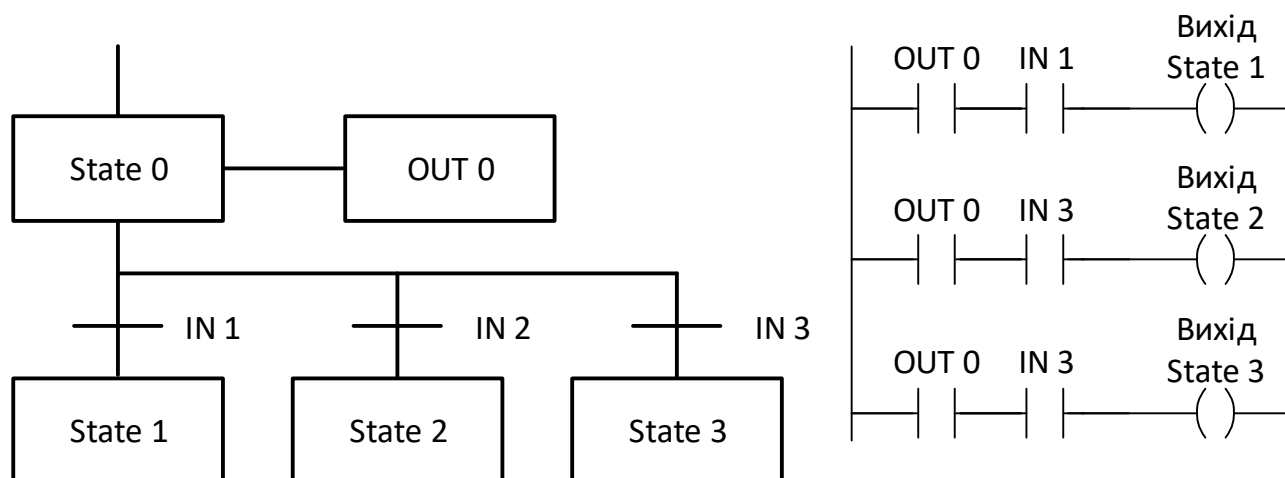
2.5.1. Розгалуження і об'єднання.

Вибіркове (селективне) розгалуження дає можливість реалізації різних станів (кроків) залежно від виникаючих умов переходів (рис. 2.52). Для цього необхідні переходи, розташовані під горизонтальною лінією, які вказують на умови, що мають бути виконані для переходу від кроку, розташованого над лінією, до одного з кроків, розташованих під нею.

Паралельне розгалуження (рис. 2.53), яке зображується парою горизонтальних ліній, дозволяє одночасно реалізувати два чи більше різних станів. Умова переходу в такому випадку розміщується поза парою горизонтальних ліній і визначає, за яких умов група наступних кроків виконується одночасно.

При вибіркового об'єднання гілок послідовність може перейти зі State 2 до State 4, якщо виконується умова IN 4, або зі State 3 до State 4, якщо виконується

умова IN 5 (рис. 2.54). При цьому переходи розташовані над горизонтальною лінією, що вказує на умови, які необхідні для того, щоб кроки над лінією могли перейти до кроку під нею.



Виконання State 1 можливо при виконанні умови переходу IN 1 але якщо IN 2 тоді State 2 або якщо IN 3 тоді State 3

Рис. 2.52. Вибіркове розгалуження: наступний за State 0 стан залежить від виконання умов IN 1, IN 2 або IN 3

В паралельному об'єднанні гілок тільки якщо виконується умова IN 4 та виконанні State 2 та/або State 3 відбувається виконання State 4 (рис. 2.55). В цьому випадку умова переходу розташована під парою горизонтальних ліній і визначає умову, за якої відбувається перехід до наступного кроку (стану).

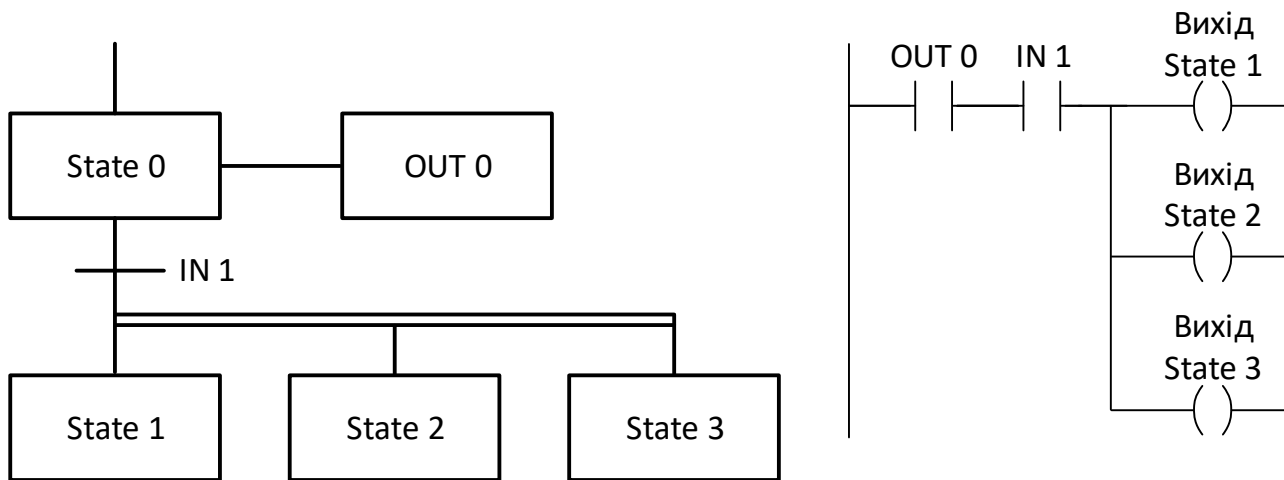
Як приклад застосування описаних вище принципів, на рис. 2.56 наведено фрагмент програми, представлений на мовах SFC та LD.

2.5.2. Дії.

Разом із кроками завжди пов'язане виконання певної дії чи набору дій. Такі дії, наприклад вихідні сигнали, наведені в попередньому прикладі, зображаються у вигляді прямокутних блоків, приєднаних до відповідного стану (рис. 2.57а). Поведінку дії можна описати за допомогою схем релейно-контактної логіки (LD), функціональних блокових діаграм (FBD), списку інструкцій (IL) або структурованого тексту (ST). Таким чином, якщо застосовується мова LD, логіка реалізації дії подається у вигляді схеми, розміщеної всередині блока дії. Дія активується тоді, коли в блок надходить живлення (перехід стає активним) (рис. 2.57б).

Блоки дій можуть мати префікси (кваліфікатори), які уточнюють умови їх виконання. Перелік кваліфікаторів дій, визначено стандартом ІЕС 61131 (табл. 2.6). За відсутності кваліфікатора або при використанні кваліфікатора N, дія не зберігається та виконується безперервно доти, доки відповідний стан активний, і припиняється одразу після його деактивації. Кваліфікатор P застосовується для імпульсних дій, які виконуються лише один раз у момент

активації кроку і відразу після цього деактивуються. Кваліфікатор D визначає дію із затримкою за часом: вона запускається тільки після завершення встановленого інтервалу і припиняється при деактивації кроку. Кваліфікатор L позначає дію з обмеженою тривалістю: вона починається при активації кроку й завершується автоматично після закінчення заданого часу (рис. 2.57в).



При виконанні умови переходу IN 1 одночасно активуються State 1, 2 та 3

Рис. 2.53. Паралельне розгалуження: State 1, 2 та 3 активується одночасно після виконання умови IN 1

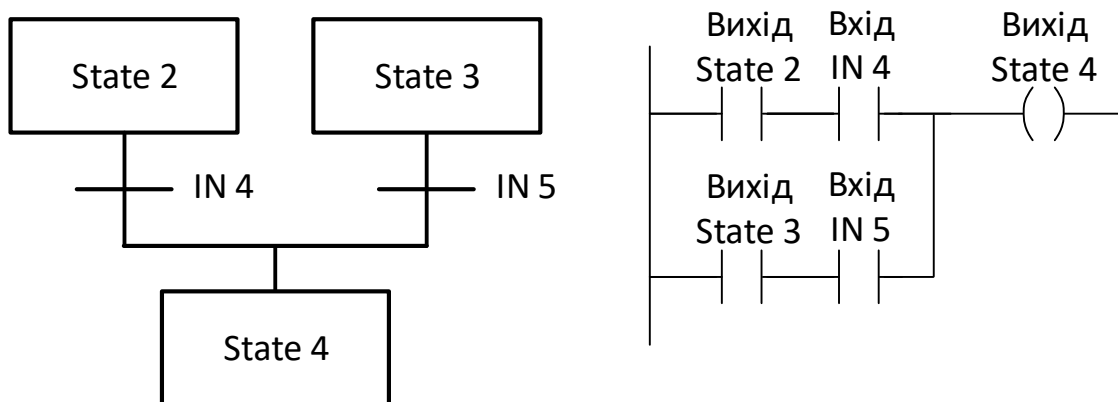


Рис. 2.54. Вибіркове об'єднання гілок

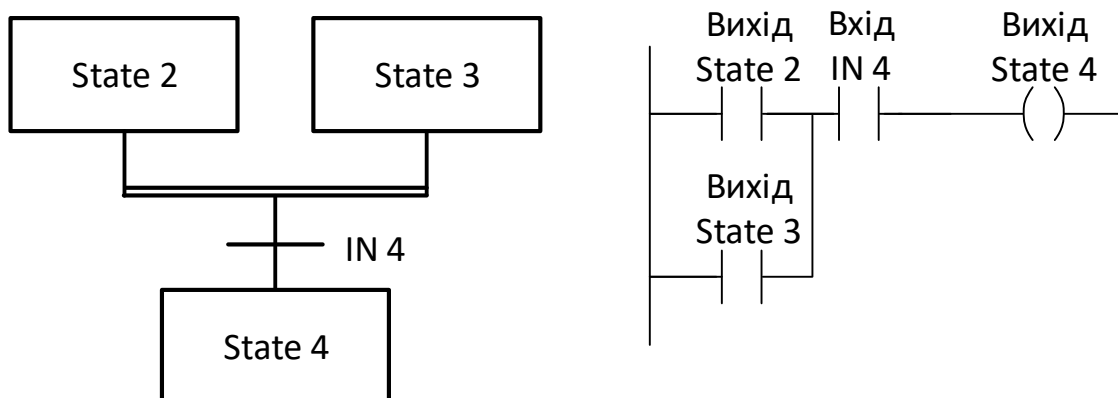


Рис. 2.55. Паралельне об'єднання гілок

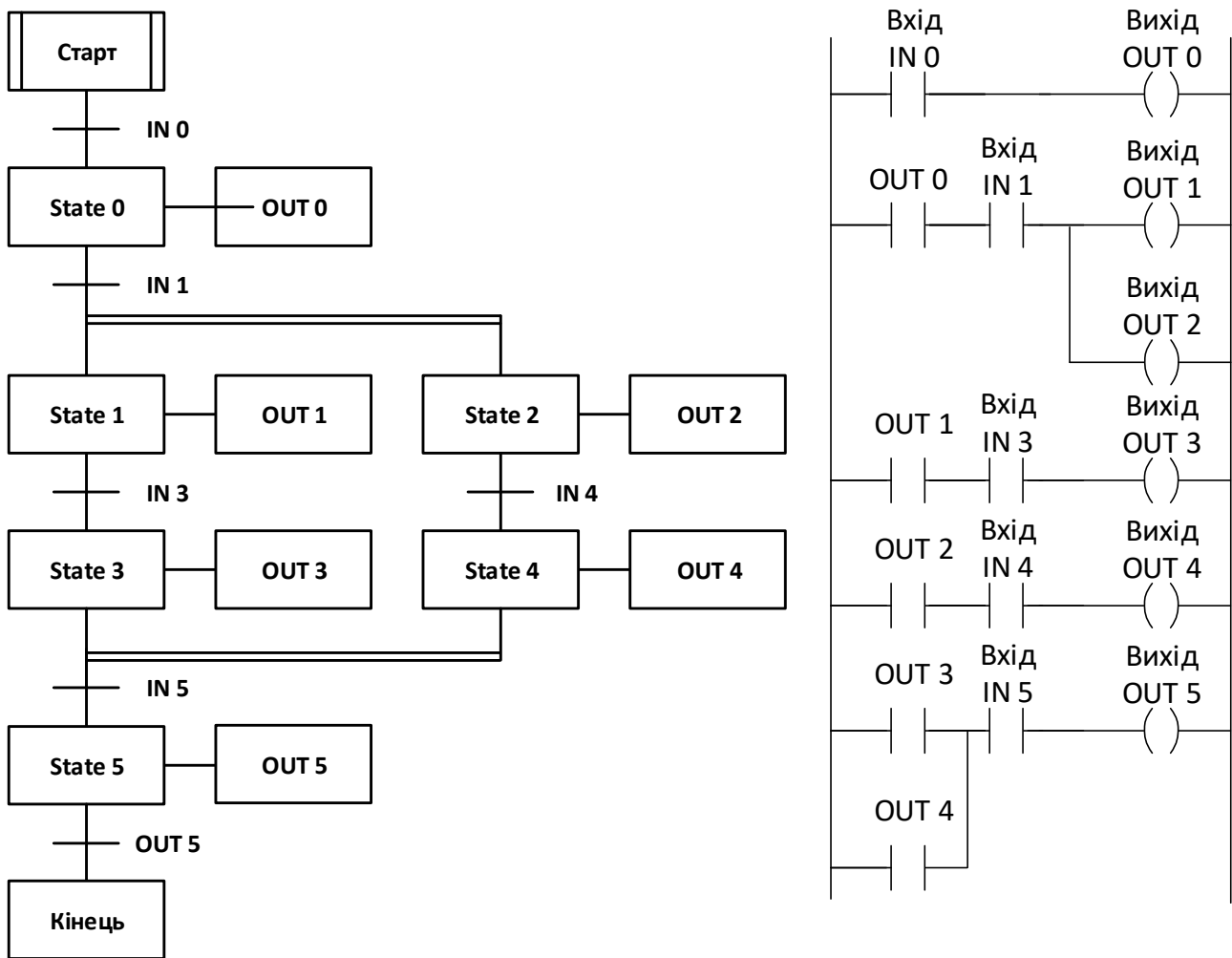


Рис. 2.56. Фрагмент програми еквівалентних програм на мовах SFC та LD

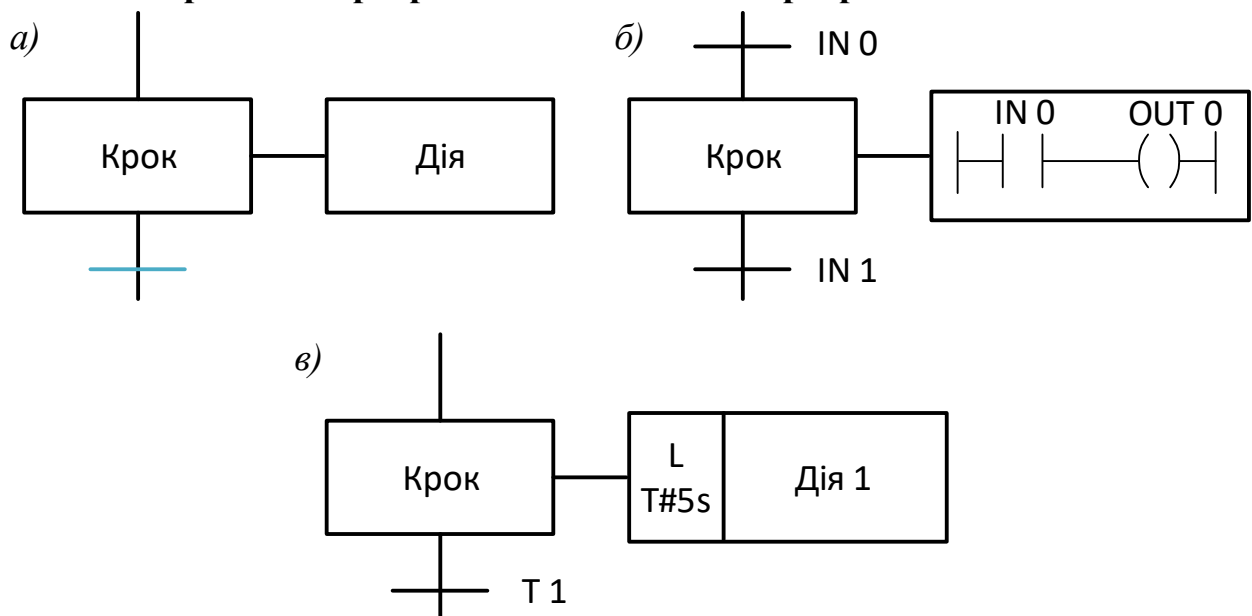


Рис. 2.57. Кроки та дії в мові SFC

Коли умова переходу виконується і один крок завершується, а наступний активується, необхідно враховувати, у якому стані слід залишити дію цього кроку. Може виникнути потреба вимкнути пристрій, який був активований у

попередньому кроці. Це реалізується програмуванням відповідної дії для такого пристрою. Якщо ж потрібно повернути його у початковий стан, тоді в програмі слід задати повернення до вихідної позиції кроку.

Таблиця 2.6

Кваліфікатори дій

Кваліфікатор	Значення	Характеристика виконання
N	Non-stored (незбережена дія)	Дія виконується безперервно, доки активний відповідний крок, і припиняється після його деактивації
S	Stored (збережена дія)	Дія виконується безперервно після активації кроку й триває навіть після його завершення, доки інша дія явно не деактивує її
R	Reset (скидання)	Використовується для припинення дії, яка була раніше активована кваліфікатором S
P	Pulse (імпульсна дія)	Дія виконується одноразово в момент активації кроку й одразу деактивується
L	Limited (обмежена за часом дія)	Дія запускається при активації кроку та автоматично припиняється після закінчення заданого проміжку часу, навіть якщо стан ще активний
D	Delayed (затримана дія)	Дія запускається лише після завершення заданої затримки й припиняється одразу після деактивації кроку
SD	Stored Delayed (збережена із затримкою)	Дія активується після закінчення заданої затримки й триває безперервно, навіть після завершення кроку, доки не буде явно скинута.
DS	Delayed Stored (затримка перед зберіганням)	Дія зберігається й виконується безперервно, але активується лише після закінчення встановленої затримки
SL	Stored Limited (збережена з обмеженням часу)	Дія запускається при активації кроку, триває обмежений проміжок часу, але зберігається й може підтримуватися навіть після завершення кроку

2.5.3. Приклади програмування.

Розглянемо приклад в якому лампа повинна горіти 5 секунд після натискання кнопки. Розглянемо змінні які необхідні для реалізації цього завдання: Button (BOOL) – стан кнопки; Lamp (BOOL) – стан лампи; Timer (TON) – виклик функціонального блоку затримки часу. В програмі на мові SFC для

програмування дій використовуємо мову ST та умови переходів відповідають значенням змінних Button та Timer.Q (рис. 2.58):

```
ACT1:    Lamp := TRUE;  
ACT2:    Timer(IN := TRUE, PT := T#5s);  
ACT3:    Lamp := FALSE; Timer(IN:=FALSE);
```

Завданням наступного прикладу є керування температурою повітря в теплиці. Необхідно підтримувати температуру повітря в межах 20...25 °С. Використовуємо в програмі наступні змінні: Temp – температура повітря; Heater – нагрівач; Vent – вентилятор охолодження (рис. 2.59). В програмі на мові SFC для програмування дій використовуємо мову ST:

```
ACT1:    Heater := TRUE; Vent := FALSE;  
ACT2:    Vent := TRUE; Heater := FALSE;  
ACT3:    Vent := FALSE; Heater := FALSE;
```

Розглянемо, як приклад систему автоматичного зволоження повітря в теплиці. Зволоження повітря активується при низькій вологості, триває 10 секунд і повторюється через 1 хвилину, якщо вологість повітря не зростає. Тобто необхідно підтримувати вологість повітря в теплиці вищою за 40 %. Для підвищення вологості повітря в теплиці використовується зволожувач повітря, який складається з розпилювачів та насоса для створення відповідного тиску води в форсунках.

Використовуємо в програмі наступні змінні: Humidity – вологість повітря; WaterPump – нагрівач (рис. 2.60). В програмі на мові SFC для програмування дій використовуємо мову ST:

```
ACT1:    WaterPump := TRUE; Delay(IN := FALSE);  
ACT2:    Timer(IN := TRUE, PT := T#10s);  
ACT3:    WaterPump := FALSE; Timer(IN:=FALSE);  
ACT4:    Delay(IN := TRUE, PT := T#20s);
```

2.6. Безперервна функціональні схеми (Continuous Function Chart).

У сучасних системах автоматизованого керування застосовується низка графічних мов програмування, що реалізують концепції стандарту IEC 61131-3.

Одним із важливих його розширень є мова безперервних функціональних схем (Continuous Function Chart, CFC), яка поєднує наочність блокової структури з можливістю формування складних зворотних зв'язків у логіці керування.

Завдяки цьому CFC знаходить широке застосування у процесному керуванні, де необхідно відображати взаємопов'язані сигнали, зокрема в системах регулювання тиску, температури, рівня чи витрати.

Мова CFC є розширенням стандарту IEC 61131-3 і базується на мові Function Block Diagram (FBD) але на відміну від неї у CFC відсутня фіксована мережева структура. Мова CFC дозволяє вільне розміщення графічних елементів, що, у свою чергу, забезпечує можливість створення петльових (зворотних) зв'язків.

Створення програмних об'єктів CFC у середовищі Machine Expert здійснюється за допомогою спеціального редактора CFC. Для створення

програми мовою SFC її обирають під час додавання нового програмного об'єкта ROU (Program Organization Unit) у проєкт. Для масштабних програм рекомендується сторінкова версія редактора.

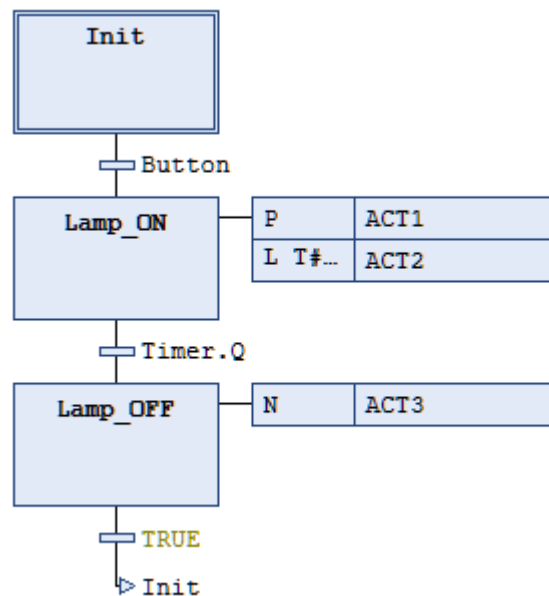


Рис. 2.58. Програма на мові SFC керування лампою в програмі Machine Expert від Schneider Electric

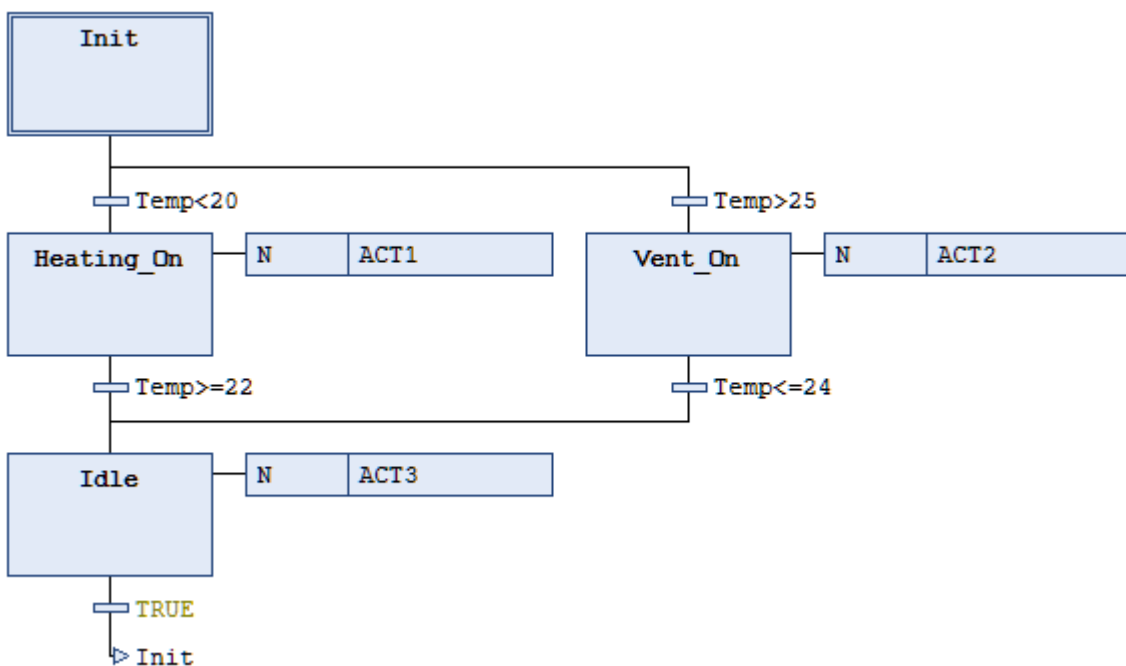


Рис. 2.59. Програма на мові SFC керування температурою повітря в програмі Machine Expert від Schneider Electric

На відміну від редакторів FBD або LD, редактор SFC забезпечує вільне розташування елементів, що дозволяє безпосередньо формувати зворотні петлі у схемі. Послідовність виконання задається списком елементів, який може бути змінений вручну.

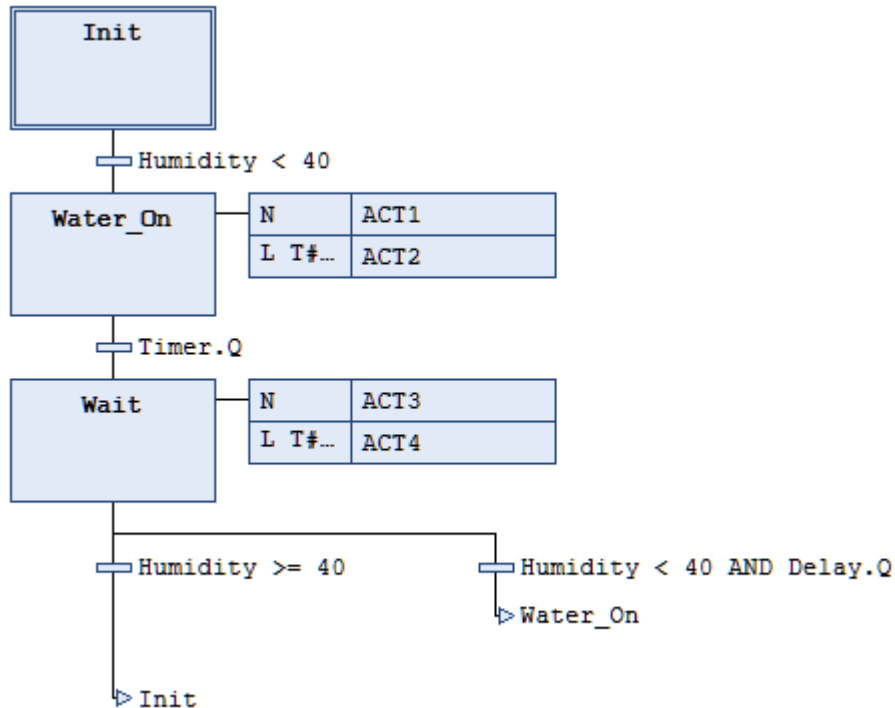


Рис. 2.60. Програма на мові SFC керування системою автоматичного зволоження повітря в теплиці в програмі Machine Expert від Schneider Electric

2.6.1. Базові елементи мови.

В панелі інструментів (Toolbox) доступні такі об'єкти, які можна вставляти перетягуванням (drag and drop) (рис. 2.61).

Елементи **Input** (вхідні сигнали)/**Output** (вихідні сигнали) використовуються для задання змінних або констант.

Елемент **Box** можна використовувати для представлення операторів, функцій, функціональних блоків або програм. Текст, позначений ???, при додаванні елемента, можна замінити назвою відповідного оператора, функції, функціонального блоку чи програми.

Якщо вставляється функціональний блок, над його зображенням з'являється ще одне поле з ???, яке потрібно замінити на ім'я екземпляра функціонального блоку.

Якщо функціональний блок містить постійні (константні) вхідні параметри, у нижньому лівому куті блока з'являється поле Parameters... Натиснувши цю кнопку, можна відкрити діалогове вікно для редагування параметрів введення.

Якщо замінити існуючий блок іншим (змінивши введене ім'я), і новий блок має іншу кількість мінімальних або максимальних вхідних чи вихідних контактів (pins), то ці контакти буде автоматично скориговано. У разі, якщо деякі контакти потрібно видалити, спочатку видаляється нижній контакт.

Для переходу використовується елемент **Jump**, та місце з якого має продовжуватися виконання програми визначається за допомогою мітки **Label**.

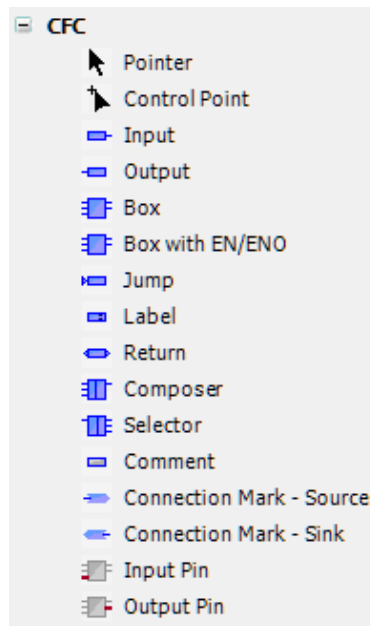


Рис. 2.61. Базові елементи мови CFC в програмі Machine Expert від Schneider Electric

У режимі онлайн елемент **Return** генерується автоматично – на початку програми, у першому стовпці редактора, а також після її завершального елемента. Під час покрокового виконання система самостійно здійснює перехід до цього елемента безпосередньо перед виходом із програмної організаційної одиниці, забезпечуючи коректне завершення виконання програми.

Елемент **Composer** застосовується у випадках, коли вхід блоку має структурований тип даних. Завдяки цьому елементу всі компоненти структури відображаються в CFC, що надає зручний доступ до їхніх полів.

Елемент **Selector** призначений для роботи з вихідними даними блоку, що мають структурований тип. Він розкриває поля структури та забезпечує доступ до кожного з них безпосередньо у CFC.

Елемент **Comment** призначений для вставки коментарів безпосередньо у схему CFC. Текст коментаря вводиться у відповідне поле та для створення нового рядка використовується комбінація клавіш CTRL+ENTER.

Замість з'єднувальної лінії між елементами можна використовувати мітки з'єднань (**Connection Marks**). Це особливо корисно для впорядкування складних схем, роблячи їх більш читабельними. Щоб створити коректне з'єднання, необхідно призначити мітку-джерело (**Source Mark**) на виході одного елемента та мітку-приймач (**Sink Mark**) на вході іншого. Обидві мітки повинні мати однакову назву (регістр символів не враховується). Перша мітка-джерело, яку вставляється в CFC за замовчуванням, має ім'я C-1, яку можна змінити вручну. У відповідній мітці-приймачі слід замінити символи ??? тим самим іменем, яке використано для мітки-джерела. Редактор контролює унікальність назв міток. Якщо змінюється ім'я мітки-джерела, то ім'я пов'язаної мітки-приймача автоматично оновлюється. Однак, якщо змінити ім'я приймача, джерело збереже

попереднє ім'я – це дозволяє гнучко переконфігувувати з'єднання. Аналогічно, видалення однієї мітки не видаляє її пару. Щоб використати мітку з'єднання у схемі, перетягніть її з панелі інструментів у вікно редактора, а потім з'єднайте її контактний вивід із входним або вихідним виводом відповідного елемента. Альтернативно можна перетворити вже існуючу звичайну лінію зв'язку на мітку за допомогою команди **Connection Mark**. Цією ж командою можна виконати зворотню дію – перетворити мітки назад у звичайні лінії.

Залежно від типу блока, у схемі CFC можна додати додатковий вхід (**Input Pin**). Для цього виберіть потрібний елемент-блок у мережі CFC та додайте контакт входу, «намалювавши» (перетягнувши) вхідний контакт безпосередньо на елемент блока. Щоб змінити положення вже існуючого входу або виходу на блоці, утримуйте клавішу Ctrl і перетягніть з'єднання у нове місце.

Аналогічно, залежно від типу блока, можна додати додатковий вихід вхід (**Output Pin**). Для цього також виберіть блок у CFC-мережі та додайте вихідний контакт безпосередньо на ньому. Як і у випадку з входами, щоб перемістити вхід або вихід у межах блока, утримуйте Ctrl під час перетягування з'єднання у бажане положення.

2.6.2. З'єднання елементів та порядок їх обробки.

Вхідні та вихідні контакти елементів можна з'єднувати лінією, проведеною мишею (рис. 2.62). Шлях з'єднувальної лінії створюється автоматично та проходить найкоротшим маршрутом. Під час переміщення елементів з'єднання коригуються автоматично.

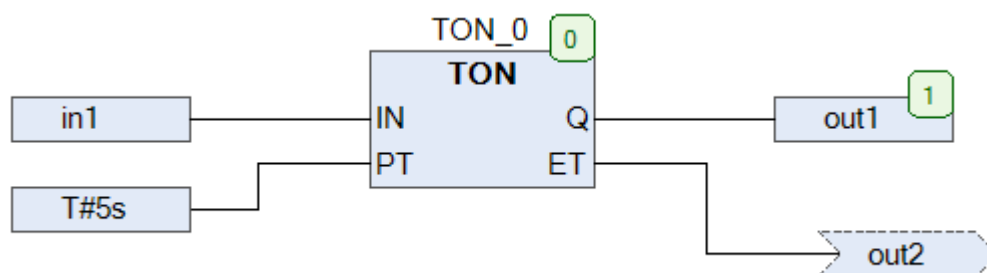




Рис. 2.62. З'єднання елементів програми на мові SFC в програмі Machine Expert від Schneider Electric

Для складних схем рекомендується використовувати позначки з'єднань (Connection Marks) замість звичайних ліній, а також можна налаштовувати маршрути з'єднань.

У разі, якщо розташування елементів призводить до перекриття існуючих з'єднань, такі колізії позначаються червоними лініями. Коли у схемі наявні колізії, кнопка в правому верхньому куті редактора отримує червону рамку. Щоб переглядати та усувати колізії послідовно, слід натиснути цю кнопку  й виконати команду «Show next collision», після чого виділяється наступне проблемне з'єднання. Для складних схем можна використовувати позначки з'єднань або перейти до сторінкової версії редактора CFC.

Функція масштабування (Zoom) дозволяє змінювати розмір вікна редактора. Для цього потрібно натиснути кнопку  в нижньому правому куті вікна та вибрати потрібний коефіцієнт масштабування зі списку. Альтернативно можна обрати пункт «...», щоб відкрити діалогове вікно і ввести довільний масштаб вручну.

Команди для роботи з редактором CFC доступні через контекстне меню або через меню CFC, яке стає активним після відкриття відповідного редактора.

Порядок виконання елементів у межах програмних організаційних одиниць (POU) однозначно визначений у текстових і мережевих редакторах. Однак у редакторі CFC (Continuous Function Chart) елементи можна розміщувати вільно, тому початково порядок їх виконання не є фіксованим.

У середовищі EcoStruxure Machine Expert послідовність виконання визначається на основі потоку даних (Data Flow), а у випадку кількох мереж – також за топологічним розташуванням елементів. Для забезпечення єдиного порядку виконання елементи впорядковуються згори донизу та зліва направо.

Хронологічну послідовність виконання елементів у схемі можна тимчасово відобразити в CFC-редакторі, увімкнувши показ порядку виконання.

Для мереж із зворотним зв'язком користувач може визначити початковий елемент петлі зворотного зв'язку.

В EcoStruxure Machine Expert передбачено можливість явного редагування порядку виконання в програмі CFC, на вкладці CFC→Execution Order у властивостях об'єкта CFC.

Загалом термін «потік даних» (Data Flow) означає хронологічну послідовність читання чи запису даних у певному програмному коді, а також способи їх обробки. Одна програмна одиниця (POU) може опрацьовувати кілька потоків даних, які, своєю чергою, можуть виконуватися незалежно один від одного.

За замовчуванням у властивостях об'єкта CFC у вкладці CFC→Execution Order активовано режим Order by Data Flow, який забезпечує автоматичне визначення порядку виконання.

Щоб задати початкову точку для визначення порядку виконання у мережах із зворотним зв'язком, необхідно виділити необхідний елемент та у вкладці CFC→Execution Order обрати Set Execution Order і задати New Execution Order (Рис. 2.63). При цьому відповідальність за правильність та наслідки визначення такого порядку виконання покладається на розробника.

2.6.3. Приклади програмування.

Розглянемо приклад автоматизації насосної станції є резервуаром, яка забезпечує подачу рідини в технологічну лінію. Необхідно підтримувати рівень рідини в межах допустимого діапазону. При зниженні рівня нижче L_{min} – відкривається клапан наповнення, при перевищенні L_{max} – клапан закривається. У разі спрацювання датчика L_{alarm} (перелив) – система формує аварію.

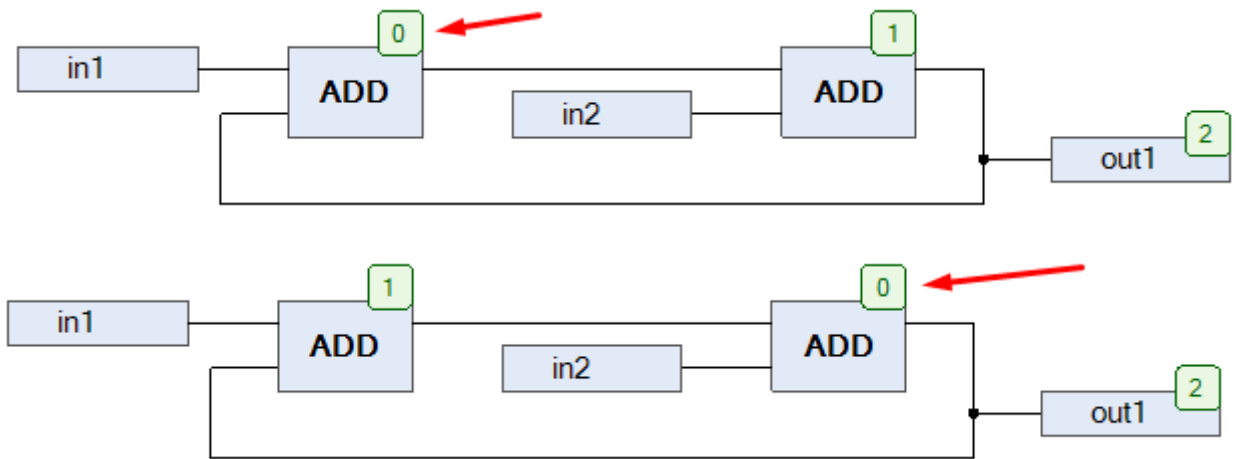


Рис. 2.63. Зміна порядку виконання елементів у програми на мові SFC в програмі Machine Expert від Schneider Electric

В автоматичних системах керування насосом необхідно передбачати систему захистів і автоперезапуск. Вона підвищує надійність роботи обладнання та здійснює автоматичне відновлення роботи після усунення несправності.

Розробимо, як приклад, для насосної станції програму на мові SFC з системою захисту (рис. 2.65). Керування пуском/зупинкою двигуна насоса здійснюється з урахуванням сигналів захисту: перевантаження, обриву фази, сухого ходу. Після відновлення нормальних умов насос має перезапуститися через 10 секунд.

Розглянемо циклічну роботу технологічного устаткування на прикладі змішувача. Необхідно реалізувати циклічну подачу енергії до приводу для перемішування суміші, з автоматичним повторенням циклів без участі оператора та можливістю оперативної зупинки. Змішувальний вузол, який періодично переміщує сировину у баку за наступним циклом роботи: перемішування протягом 30 секунд – пауза 10 секунд – повтор. Робота триває до натискання кнопки STOP (рис. 2.66).

Напря́м	Позначення	Тип	Опис
Вхідні	Lmin	BOOL	Датчик нижнього рівня
	Lmax	BOOL	Датчик верхнього рівня
	Lalarm	BOOL	Датчик переливу
	Start	BOOL	Кнопки керування
	Stop	BOOL	Кнопки керування
Вихідні	Valve_Fill	BOOL	Клапан подачі
	Run_LED	BOOL	Індикатор “Система в роботі”
	ALARM	BOOL	Аварійна сигналізація

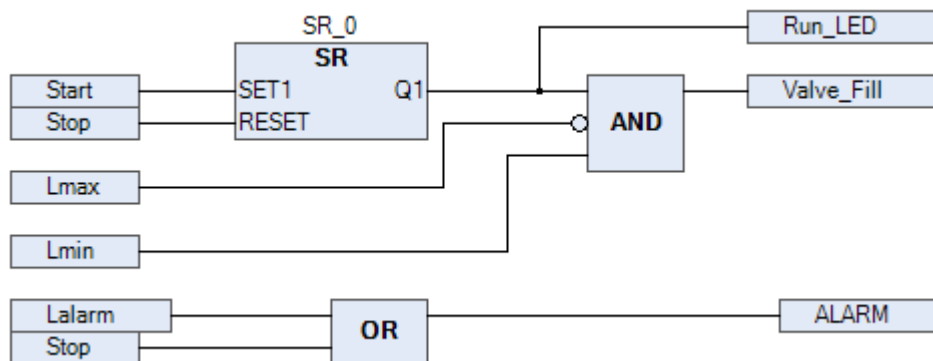


Рис. 2.64. Програма на мові SFC керування насосною станцією в програмі Machine Expert від Schneider Electric

Напря́м	Позначення	Тип	Опис
Вхідні	Start	BOOL	Кнопка Пуск
	Stop	BOOL	Кнопка Стоп
	Overl	BOOL	Перевантаження
	PH_Loss	BOOL	Втрата фази
	Dry_RUN	BOOL	Сухий хід
Вихідні	Motor_ON	BOOL	Контактор двигуна
	Fault_LED	BOOL	Індикація аварії
	Restart_timer	BOOL	Індикатор перезапуску

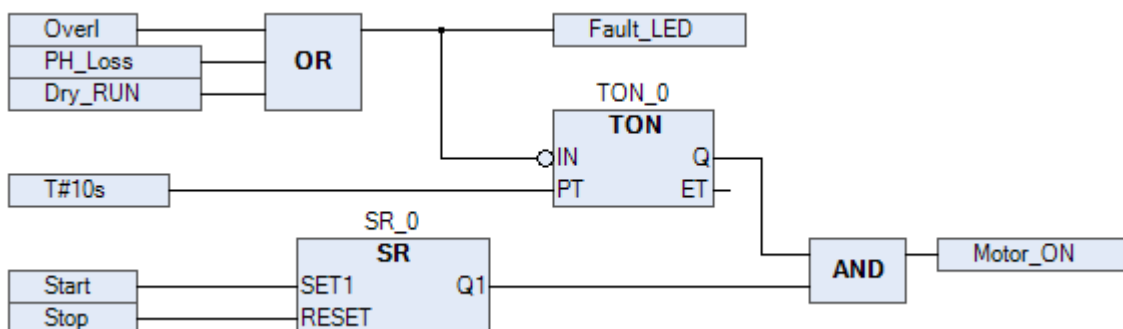


Рис. 2.65. Програма на мові SFC для керування насосом в програмі Machine Expert від Schneider Electric

Цей приклад демонструє реалізацію циклічного процесу в SFC з використанням двох послідовних таймерів та може бути використаний для керування мішалками, дозаторами або компресорами з періодичним режимом роботи.

Напря́м	Позначення	Тип	Опис
Вхідні	Start	BOOL	Кнопка Пуск
	Stop	BOOL	Кнопка Стоп
	Time_RUN	TIME	Тривалість роботи
	Time_Pause	TIME	Тривалість паузи
Вихідні	Motor	BOOL	Привід змішувача
	Cycle	BOOL	Індикатор активності циклу

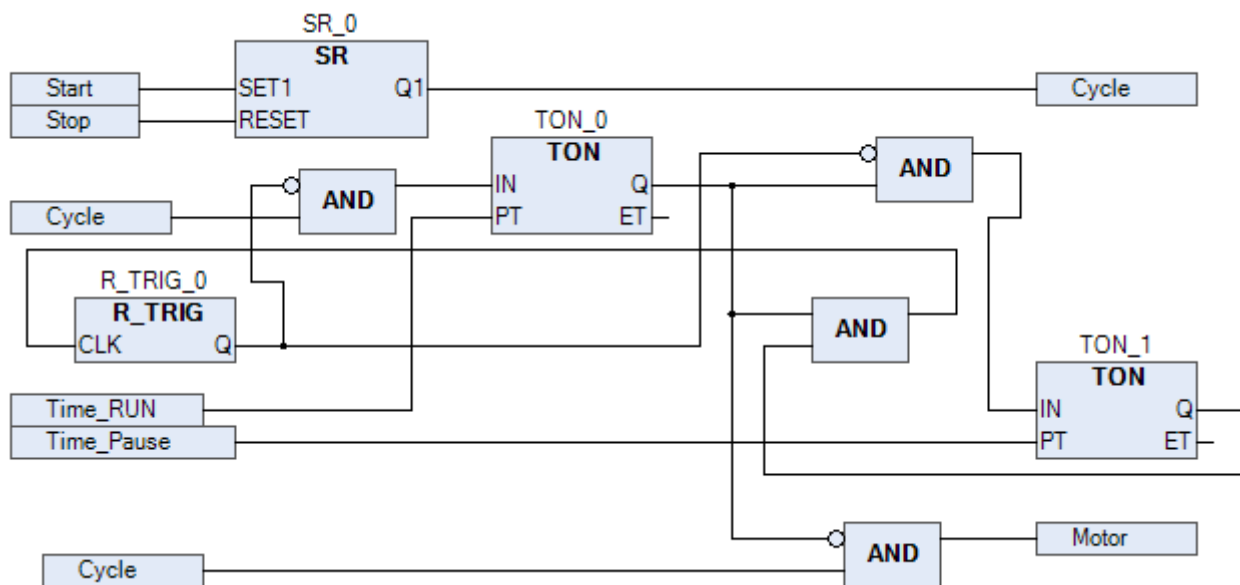


Рис. 2.66. Програма на мові SFC для керування циклічною роботою насосу в програмі Machine Expert від Schneider Electric

Висновки за розділом.

1. Релейно-контактна логіка (Ladder Diagrams, LD) – це найбільш наочна мова програмування ПЛК, побудована за аналогією з електричними схемами. Вона використовує символи контактів і котушок для реалізації логічних операцій («І», «АБО», «НІ», «NAND», «NOR», «XOR») і широко застосовується для створення простих алгоритмів керування технологічними процесами.
2. Список інструкцій (Instruction List, IL) – мова програмування низького рівня, схожа на асемблер. Вона дозволяє виконувати послідовні логічні та арифметичні операції, реалізовувати розгалуження та цикли, забезпечуючи більш компактне подання програми, хоча є менш наочною порівняно з LD.
3. Структурований текст (Structured Text, ST) – це мова високого рівня, що базується на алгоритмічних принципах програмування. Вона підтримує умовні оператори, цикли, оголошення типів, змінних та функціональних

блоків, що робить її зручною для розробки складних і гнучких програм керування.

4. Функціональні блокові діаграми (Function Block Diagram, FBD) дозволяють створювати програми у вигляді з'єднаних блоків, кожен з яких виконує певну функцію. Такий підхід забезпечує зручність у візуальному відображенні процесів і підвищує швидкість розробки програм завдяки повторному використанню функціональних елементів.
5. Послідовні функціональні схеми (Sequential Function Charts, SFC) описують процес керування як послідовність кроків і переходів. Ця мова особливо ефективна для багатостадійних процесів, де необхідно чітко визначати умови переходів між етапами та дії, що виконуються на кожному з них.
6. Безперервні функціональні схеми (Continuous Function Chart, CFC) дозволяють реалізовувати безперервні алгоритми регулювання, поєднуючи блокову логіку та часову послідовність виконання. Ця мова широко використовується в системах автоматичного регулювання, де важливо забезпечити плавність змін параметрів і стабільність процесів.

Питання для самоперевірки.

1. Які основні мови програмування ПЛК визначені стандартом IEC 61131-3 і в чому полягає їхня відмінність?
2. У чому полягає принцип побудови мови релейно-контактних схем (Ladder Diagram, LD) і чому вона зручна для електромонтажників?
3. Як реалізуються логічні операції AND, OR, NOT, NAND, NOR, XOR у середовищі мови LD? Наведіть приклади їхнього використання.
4. Що таке фіксація (Latching) у програмах на LD і для чого вона застосовується?
5. Як виглядає різниця між програмами LD і списку інструкцій (Instruction List, IL)? У яких випадках доцільно застосовувати IL?
6. Що таке структурований текст (Structured Text, ST) і які основні конструкції в ньому використовуються (умовні оператори, цикли, оголошення змінних)?
7. Які переваги мови функціональних блокових діаграм (Function Block Diagram, FBD) і як відбувається обробка сигналів між блоками?
8. Що таке послідовна функціональна схема (Sequential Function Chart, SFC) і як у ній реалізуються послідовність, розгалуження та об'єднання дій?
9. Які особливості має мова безперервних функціональних схем (Continuous Function Chart, CFC) порівняно з іншими мовами програмування ПЛК?
10. Як вибір мови програмування ПЛК впливає на зручність розробки, читання та супровід програми автоматизації?

РОЗДІЛ 3. ПРОЕКТУВАННЯ СИСТЕМ АВТОМАТИЗАЦІЇ

3.1. Розробка програми.

Незалежно від мови програмування, у якій створюється програма, системний підхід до постановки задачі значно підвищує ймовірність отримання якісного результату за мінімально можливий час [4, 5, 6, 8, 10, 12, 13, 14, 15]. Типова методика проектування передбачає такі етапи:

1. Визначення вимог – опис того, що саме має виконувати програма, із чітким переліком усіх вхідних і вихідних сигналів.
2. Формулювання алгоритму – розроблення покрокової послідовності дій для розв’язання задачі. Алгоритм зазвичай подають у вигляді блок-схеми або псевдокоду, використовуючи службові слова BEGIN, DO, END, IF-THEN-ELSE, WHILE-DO.
3. Переклад алгоритму у програмний код ПЛК.
4. Оскільки реальні програми часто є громіздкими та складними для сприйняття у вигляді суцільного блоку, доцільно ділити їх на функціональні частини, які згодом деталізуються до логічно завершених модулів. Такий підхід називається ієрархічним (або каскадним) проектуванням згори вниз (top-down design).
5. Тестування та налагодження (debugging) – покрокова перевірка працездатності програми з усуненням логічних і синтаксичних помилок.
6. Документування програми – опис її структури, функціонування, вхідних і вихідних параметрів для подальшого супроводу, модифікацій чи передавання іншому фахівцеві.

На рис. 3.1а наведено основні символи, що використовуються в блок-схемах. Кожен етап алгоритму позначається певним символом і з’єднується лініями, які відображають логіку виконання програми (рис. 3.1б).

Псевдокод – це спосіб опису кроків алгоритму у вільній текстовій формі, максимально наближеній до мови програмування, але без прив’язки до конкретного синтаксису.

Послідовні процеси (Sequential). Розглянемо приклад, коли подія А має передувати події В. У блок-схемі (рис. 3.2б) це подано як послідовність операцій. У псевдокоді запис виглядає так:

```
BEGIN A
      DO A
END A
BEGIN B
      DO B
END B
```

В схемі релейно-контактної логіки (Ladder Diagram) (рис. 13.2б) ця логіка реалізується так: коли надходить сигнал «Пуск», активується вихід А; після виконання дії А спрацьовує реле, яке зумовлює активацію виходу В.

В SFC (Sequential Function Chart) (рис. 13.2в) таку ж послідовність відображено у вигляді двох кроків, що виконуються один за одним.

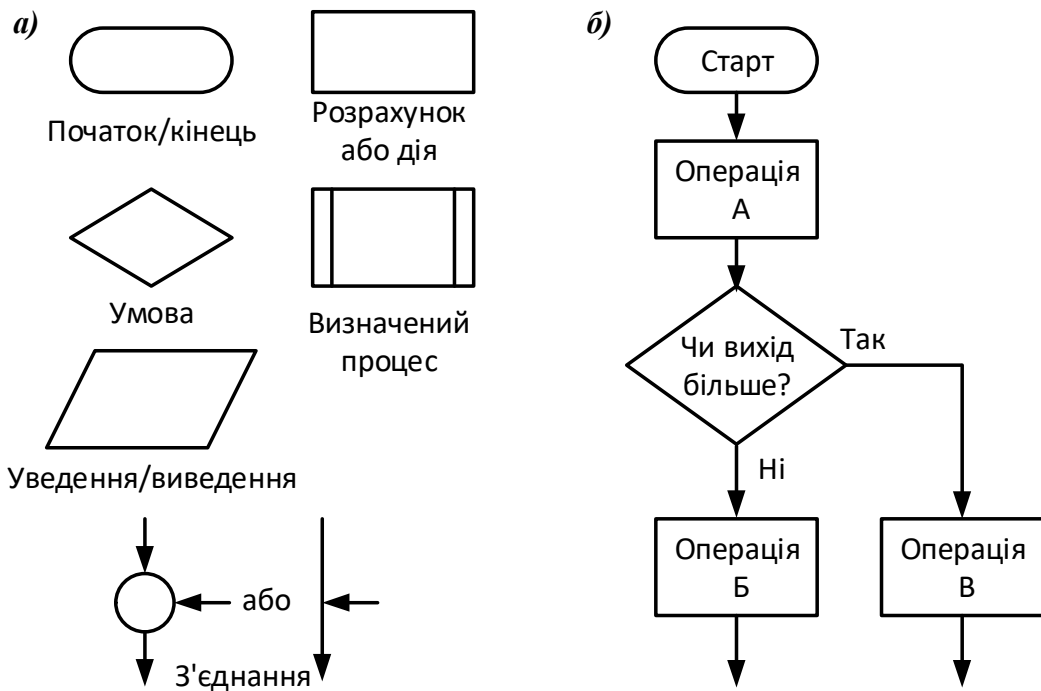


Рис. 3.1. Символьні позначення (а) та приклад алгоритму (б)

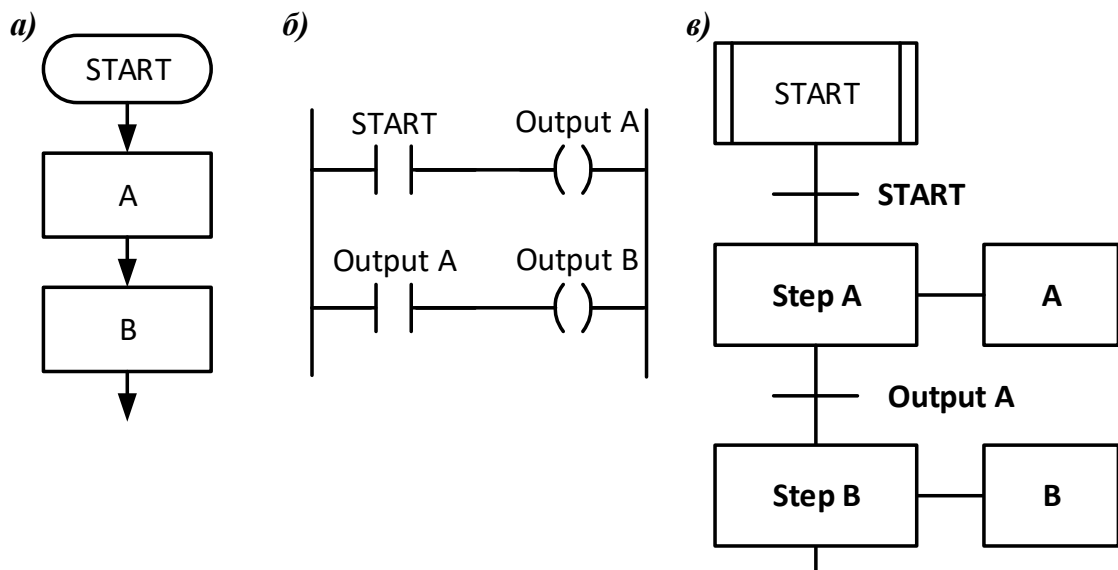


Рис. 3.2. Послідовність

Умовне розгалуження (Conditional). На рис. 3.3а показано алгоритм, у якому виконується дія А або В, залежно від умови X (TRUE/FALSE). Псевдокод має вигляд:

```

IF X
  THEN
    BEGIN A
      DO A
    END A
  
```

```

ELSE
    BEGIN B
        DO B
    END B
ENDIF X

```

В схемі релейно-контактної логіки (рис. 3.3б) умова реалізується шляхом комбінації контактів: якщо сигнал X активний, спрацьовує вихід А, інакше вихід В. Використовуючи мову SFC (рис. 3.3в) це має структуру селективного розгалуження – один із двох шляхів активується залежно від логічного стану X.

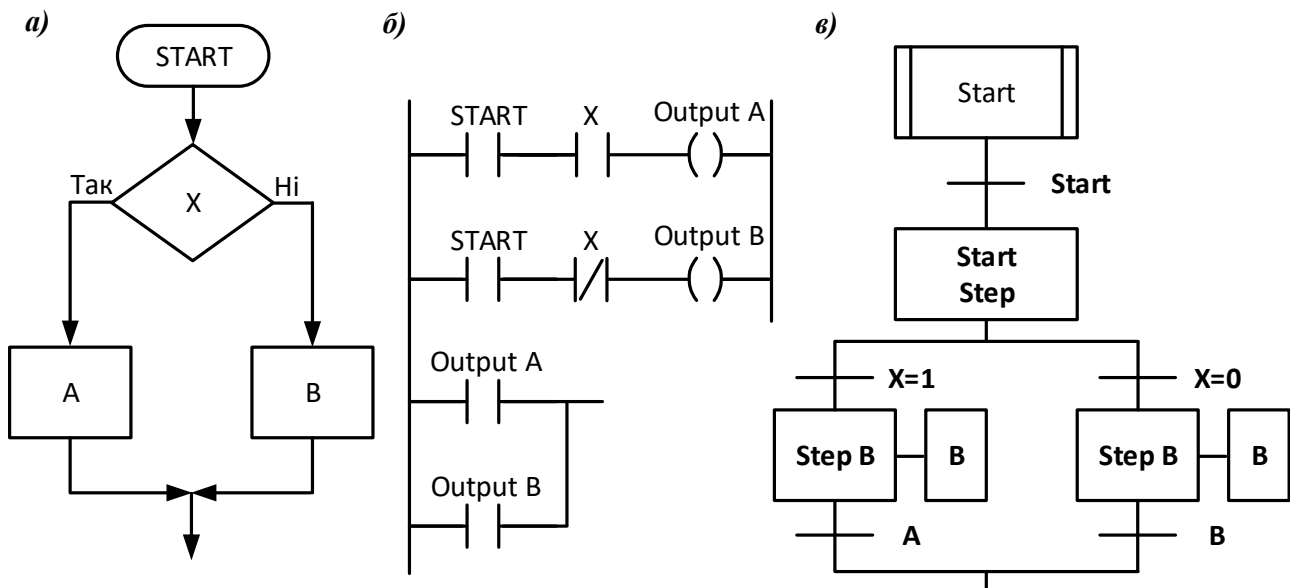


Рис. 3.3. Умови

Циклічні процеси (Looping). Цикл означає повторення певної послідовності дій, поки виконується умова X. В блок-схемі (рис. 3.4а) повторюється послідовність A → B, доки умова X = TRUE. Коли X стає FALSE, цикл завершується. Псевдокод має вигляд:

```

WHILE X
    BEGIN A
        DO A
    END A
    BEGIN B
        DO B
    END B
ENDWHILE X

```

В схемі релейно-контактної логіки (рис. 3.4б) це може бути реалізовано за допомогою внутрішнього реле (internal relay), яке зберігає стан виконання циклу.

В SFC (рис. 13.4в) цикл утворюється замиканням ланцюга між останнім і першим кроком. Якщо цикл має повторюватися певну кількість разів, у схему вводять лічильник (counter), який отримує імпульс щоразу, коли цикл завершується, і виходить із циклу після досягнення встановленого значення (рис. 3.5).

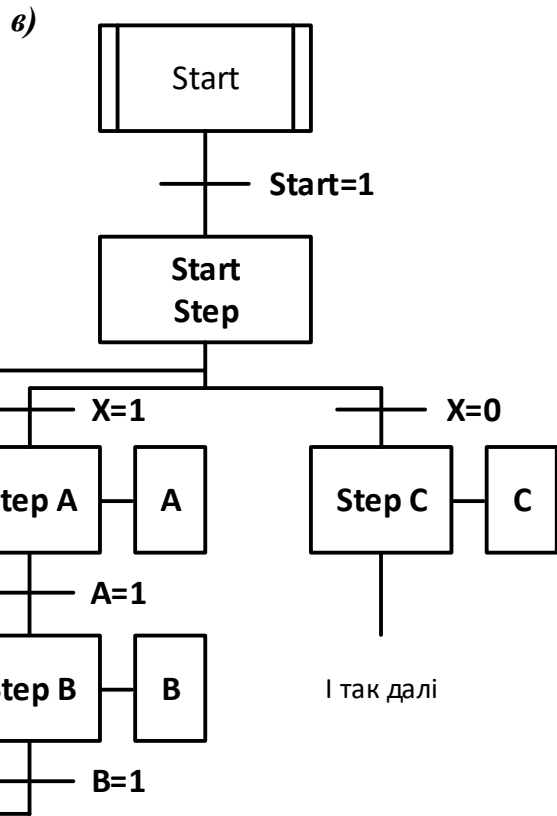
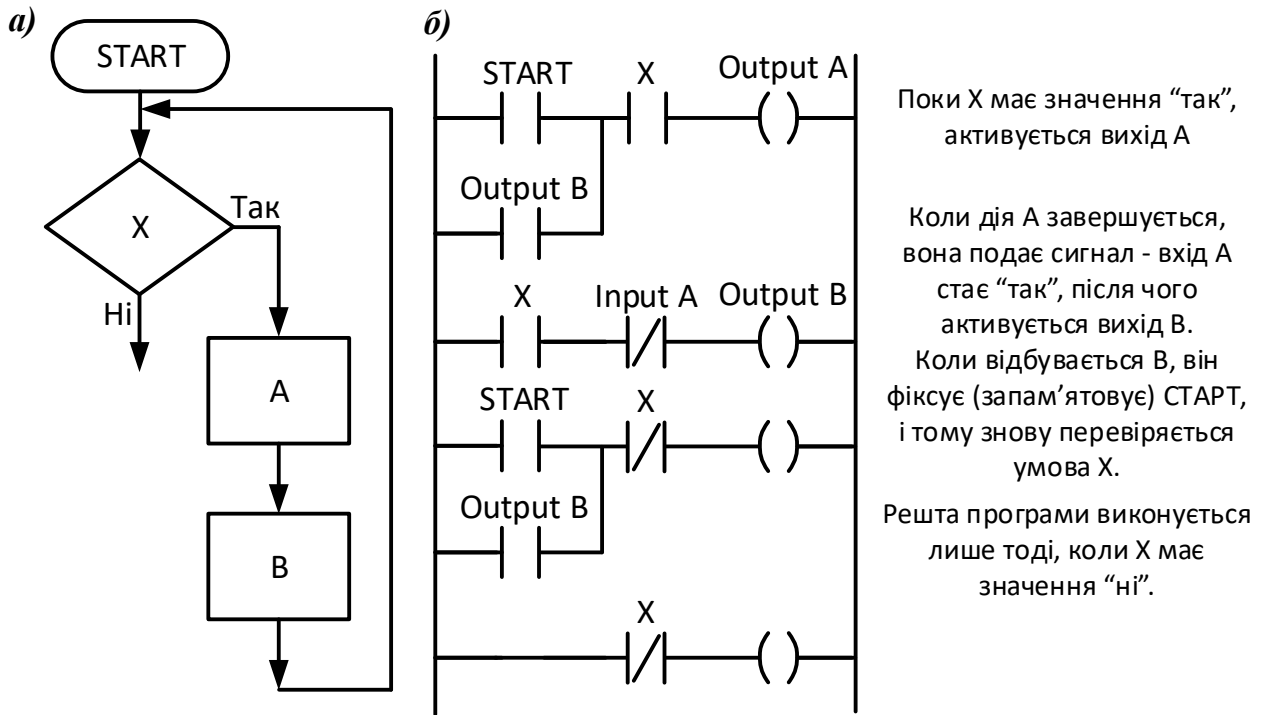


Рис. 3.4. Цикл

3.2. Заходи безпеки при проектуванні систем автоматизації.

Сучасне законодавство у сфері безпеки покладає на роботодавців обов'язки, які включають забезпечення безпечних умов праці, усунення ризиків для

здоров'я працівників, гарантування безпечного стану обладнання та впровадження безпечних методів виконання робіт.

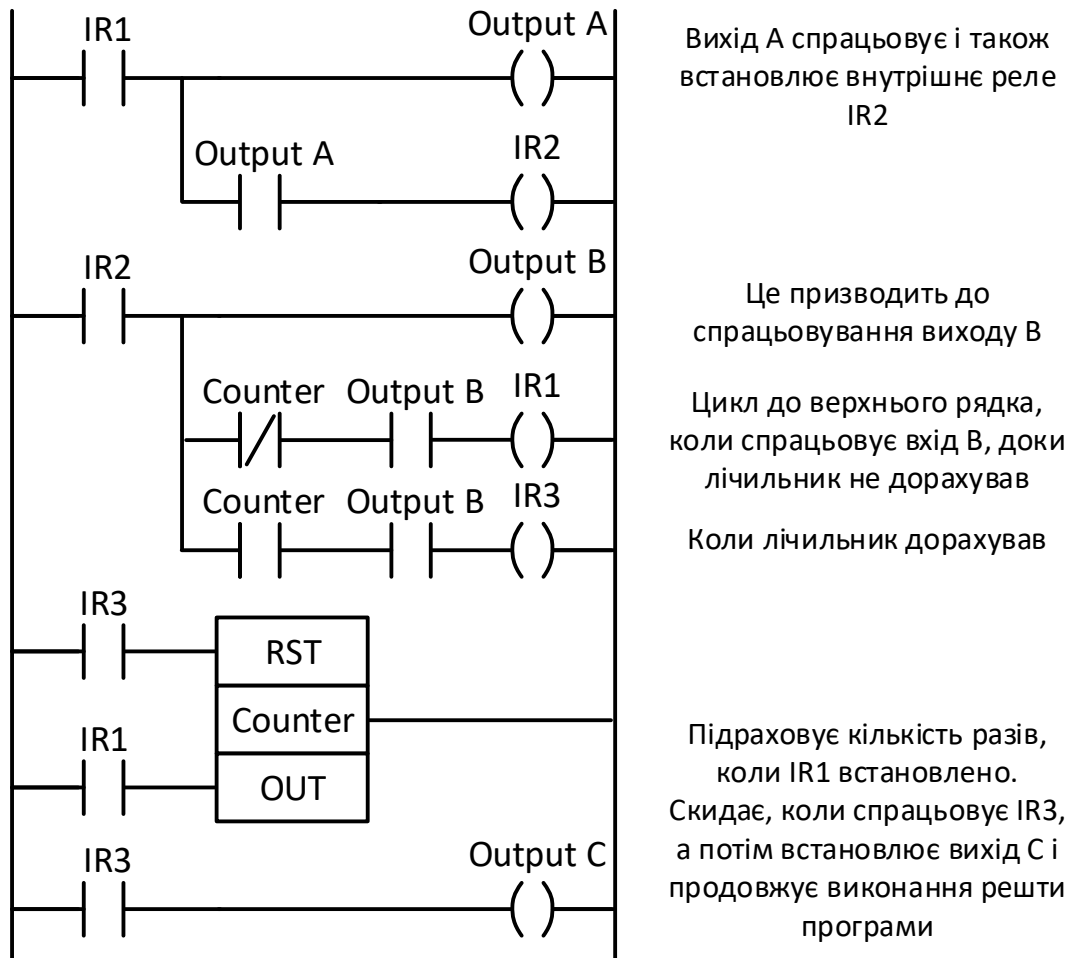


Рис. 3.5. Циклічне виконання

Це вимагає оцінки ризиків на робочому місці, що передбачає:

- виявлення потенційних небезпек (усіх факторів, здатних завдати шкоди);
- визначення осіб, які можуть постраждати, і способів, якими це може статися;
- оцінювання рівня ризику – ймовірності та тяжкості можливої шкоди;
- аналіз наявних засобів захисту й ухвалення рішення, чи достатні вони, чи потрібно вжити додаткових заходів;
- фіксування результатів аналізу;
- перегляд і коригування оцінки у разі змін у виробничому процесі чи виявлення нових ризиків.

Наприклад, під час проектування виробничого обладнання необхідно враховувати питання аварійних зупинок і захисних дверцят. Для цього виконують оцінку ризику та створюють безпечні схеми управління.

Зокрема, на дверцятах можуть встановлюватися кінцеві вимикачі, які зупиняють роботу системи, якщо доступ до небезпечних частин обладнання відкрито або захисні огороження не зафіксовані.

Розглянемо міжнародні стандарти безпеки. Ключовим нормативним документом у цій сфері є ІЕС 61508 – «Функціональна безпека електричних, електронних та програмованих електронних систем, пов'язаних із безпекою».

Він складається із семи частин:

1. Загальні вимоги;
2. Вимоги до електричних/електронних/програмованих електронних систем безпеки (E/E/PE);
3. Вимоги до програмного забезпечення;
4. Терміни та скорочення;
5. Приклади методів визначення рівнів цілісності безпеки (SIL);
6. Рекомендації щодо застосування частин ІЕС 61508-2 та ІЕС 61508-3;
7. Огляд технік і заходів щодо забезпечення безпеки.

Цей стандарт безпеки призначено для оцінки рівня якого досягає функціональна безпека машини чи установки система керування або захисту, пов'язана з безпекою, яка повинна:

- працювати коректно за нормальних умов;
- забезпечувати безпечний стан об'єкта у випадку відмови або несправності, тобто здійснювати автоматичне аварійне відключення чи переведення системи у безпечний режим.

Таким чином, концепція безпечних систем передбачає не лише апаратний і програмний захист, а й цілісну стратегію – від ідентифікації ризиків до впровадження надійних функцій безпеки відповідно до міжнародних стандартів.

3.2.1. Системи з ПЛК та безпека.

Безпека повинна бути першочерговим пріоритетом під час проєктування системи з використанням ПЛК (програмованого логічного контролера).

Зокрема, кнопки аварійної зупинки та вимикачі захисних огорожень мають бути жорстко підключені (Hardwired), а не реалізовані програмно через ПЛК. Це гарантує, що навіть у випадку відмови контролера або помилки програми система залишиться у безпечному стані.

Іншими словами, система має бути відмовобезпечною (Fail-Safe) – тобто, у разі будь-якої несправності вихідні сигнали переходять у безпечний режим, що виключає можливість травмування персоналу.

Наприклад, захисні дверцята на устаткуванні не повинні відкриватися і не можуть бути відчиненими, якщо ПЛК вийшов з ладу.

Розглянемо небезпечну схему керування зупинкою. У найпростішому випадку зупинка може реалізовуватись через звичайний перемикач (рис. 3.6). Для запуску натискається кнопка «Пуск» вона короткочасно замикає коло, вмикаючи внутрішнє реле керування двигуном. Реле «Запам'ятовує» стан і утримує вихід у ввімкненому положенні. Для зупинки потрібно короткочасно натиснути кнопку «Стоп» – вона розмикає коло та знімає утримання реле. Однак така схема небезпечна: якщо кнопка «Стоп» зламається або її контакт не розімкнеться, зупинка стане неможливою. У разі відмови елемента система продовжить працювати, створюючи небезпеку для оператора.

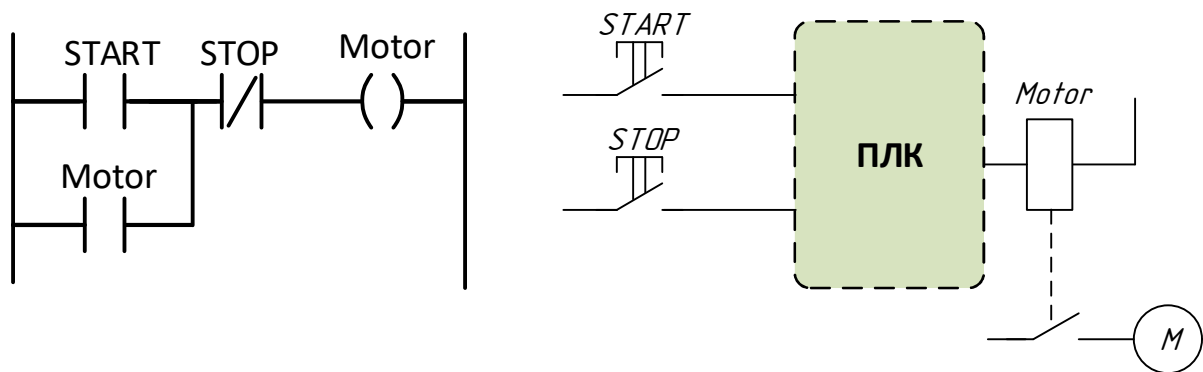


Рис. 3.6. Небезпечна зупинка системи

Розглянемо алгоритм відмовобезпечної схеми зупинки (рис. 3.7). У цій схемі кнопка «Стоп» має нормально замкнені контакти (NC). Програма ПЛК інтерпретує цей стан як «Дозвіл роботи». Коли кнопку «Стоп» натискають – коло розмикається, програма втрачає сигнал дозволу і виконує зупинку обладнання.

Перевага такого рішення полягає в тому, що навіть у разі обриву проводу, відмови кнопки або втрати живлення, система автоматично переходить у режим зупинки.

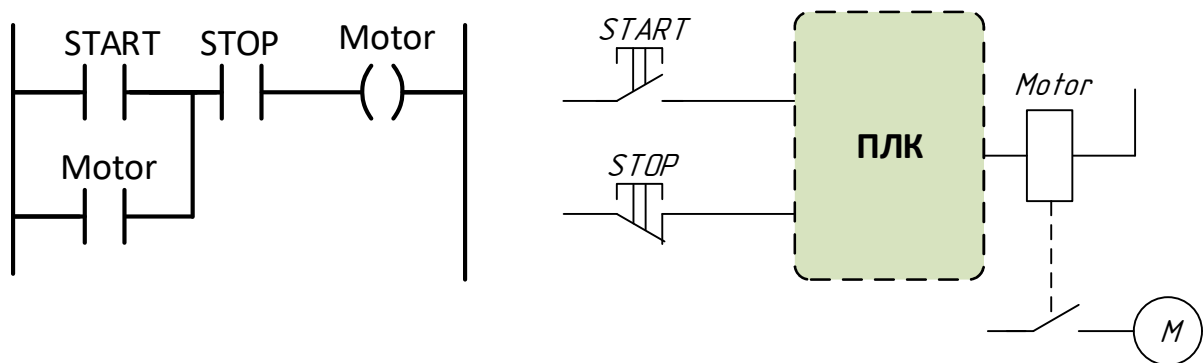


Рис. 3.7. Безпечна зупинка системи

3.2.2. Реле аварійної зупинки.

Реле аварійної зупинки широко застосовуються для організації ланцюгів безпечної зупинки обладнання. Прикладом може слугувати модуль Harmony XPS від компанії Schneider Electric (рис. 3.8-9).

Такі пристрої мають світлодіодні індикатори (LED) для відображення стану: вхідних і вихідних кіл, кола скидання (reset circuit), живлення, а також можливих несправностей. Базовий модуль може бути з'єднаний із інтерфейсним модулем, який дозволяє зчитувати стан реле аварійної зупинки ПЛК. З'єднання виконується через оптоізолятори, які електрично розділяють вихід реле від вхідних кіл ПЛК.

Таким чином, аварійна зупинка функціонує незалежно від ПЛК, але при цьому ПЛК може отримати сигнал про її спрацювання – наприклад, щоб ініціювати контрольоване завершення технологічного процесу або безпечне вимкнення обладнання (рис 3.10).



Рис. 3.8. Контролер безпеки, Harmony XPS MC від компанії Schneider Electric



Рис. 3.9. Захисний модуль Harmony XPS UAT та реле безпеки XPSBAT12A1AC від компанії Schneider Electric

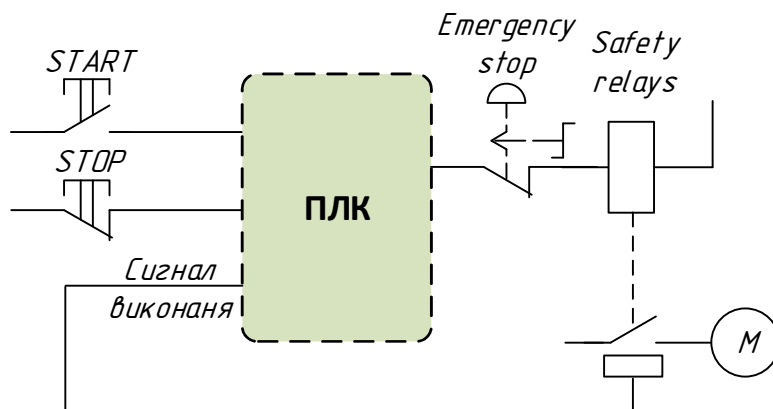


Рис. 3.10. Реле аварійної зупинки з контуром зворотного зв'язку

Принцип роботи простого реле аварійної зупинки, тобто коли кнопка Emergency Stop розриває коло керування реле (рис. 3.11а), що призводить до вимкнення живлення силового кола.

Проте така схема має серйозний недолік: якщо контакти реле зваряться (залипнуть), то навіть після натискання кнопки аварійної зупинки живлення не припиниться, і система залишиться небезпечною.

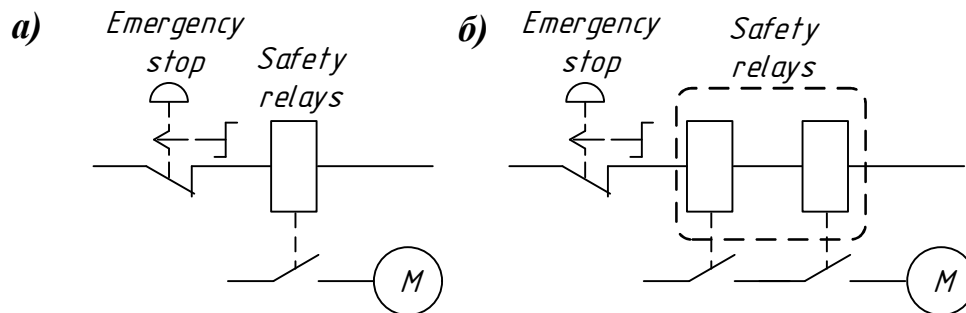


Рис. 3.11. Реле аварійної зупинки: а) одноканальний режим або двоканальний режим (б)

Для підвищення безпеки застосовують двоканальний принцип роботи (Dual-channel Mode), який виключає виникнення відмови спрацювання, шляхом встановлення в коло керування послідовно ввімкнених двох нормально замкнених (NC) контактів (рис. 3.11б). Під час натискання кнопки «Аварійної зупинки» обидва кола одночасно розриваються, і для відновлення роботи необхідне повне ручне скидання системи (Manual Reset). Це знижує ймовірність того, що відмова одного контакту призведе до небезпечної ситуації.

Ще більшої безпеки можна досягти, якщо застосувати три контакти в серії, причому: один – нормально замкнений (NC), два інші нормально розімкнені (NO). У такому разі для забезпечення аварійного вимкнення необхідно, щоб одна група контактів розімкнулася, а інші дві замкнулися (при знеструмленні реле). Це гарантує, що навіть у випадку залипання одного контакту або часткової відмови реле, система все одно перейде в безпечний стан.

Реле аварійної зупинки є ключовим елементом функціональної безпеки систем автоматизації та їх використання забезпечує:

- Апаратну незалежність від ПЛК.
- Миттєву реакцію на натискання кнопки «Аварійної зупинки».
- Контроль стану контактів і каналів (дво/три канальний).
- Безпечне відновлення роботи після ручного скидання.

Завдяки цьому аварійні реле Harmony XPS, є обов'язковим компонентом сучасних відмовобезпечних систем керування технологічними установками.

3.2.3. Функції безпеки.

Під час проектування систем керування першочерговою вимогою є захист персоналу від контакту з рухомими частинами машин під час їх роботи. Для цього використовують різні інженерні засоби безпеки, що забезпечують

автоматичне вимкнення обладнання при порушенні захисних умов. Основні з них такі:

Дво-ручне керування. У цьому способі для запуску або підтримання роботи машини потрібне одночасне натискання двох кнопок обома руками. Якщо оператор забирає одну руку (наприклад, щоб доторкнутися до робочої зони), система негайно зупиняє установку. Цей принцип запобігає можливості контакту людини з небезпечними частинами під час роботи обладнання.

Контроль захисних дверей (Protective Door Monitoring). Щоб запобігти доступу до устаткування під час її роботи, використовують захисні дверцята або огорожі, обладнані блокувальними вимикачами (Safety Interlocks). При відкритті дверей або воріт спрацьовує кінцевий вимикач (Limit Switch), який розриває ланцюг керування та зупиняє машину. У простих системах такі вимикачі можна обійти (наприклад, вставивши викрутку у замок), тому сучасні рішення передбачають: індуктивні або магнітні датчики наближення, замкові системи з ключем (Key Interlocks), які унеможливають обман системи без розблокування ключем.

Світлові завіси (Light Curtains). Світлові завіси – це оптоелектронні бар'єри, що створюють мережу інфрачервоних променів навколо небезпечної зони устаткування (наприклад, пакувального автомата). Якщо хтось перетинає промінь, система миттєво подає команду безпечної зупинки (Safe Shutdown). Такі системи особливо ефективні для високошвидкісних процесів, де механічні огороження непридатні.

Безпечні килимки (Safety Mats). Сенсорні килимки розміщуються навколо устаткування. Якщо оператор або інша особа ступає на килимок, замикається контакт, який зупиняє роботу устаткування. Цей метод простий, надійний і ефективний для зон обслуговування або завантаження обладнання.

Реле аварійної зупинки (Emergency Stop Relays). Вони забезпечують миттєве відключення живлення устаткування у разі виникнення небезпечної ситуації. Такі реле виконуються з подвійним або потрійним каналом для підвищення надійності. «Аварійна кнопка» дозволяє оператору миттєво припинити роботу обладнання вручну.

На робочому місці персоналу всі перелічені заходи використовують комплексно, наприклад: захисні огорожі з дверцятами, що розмикають реле зупинки при відкритті; світлові завіси навколо небезпечних зон; «Аварійні кнопки» у доступних місцях. Усі ці елементи інтегруються в єдину систему функціональної безпеки, що гарантує негайне зупинення небезпечного руху у разі порушення будь-якої з умов захисту.

3.2.4. Контролери безпеки.

Контролери безпеки (Safety PLCs) – це спеціально розроблені контролери, які забезпечують виконання функцій безпеки в автоматизованих системах.

На відміну від звичайних ПЛК, у конструкції контролера безпеки передбачено два або три мікропроцесори, які одночасно виконують ідентичну логіку керування, постійно перевіряють результати один одного та формують вихідні сигнали лише у разі повної узгодженості результатів.

Одним із поширених рішень є двоканальна архітектура, що містить два ідентичні підсистеми, які обмінюються даними через оптоволоконний канал зв'язку.

Сигнали з датчиків подаються одночасно на обидві підсистеми. Під час роботи контролерів між підсистемами безперервно передаються дані для порівняння – це стосується вхідних і вихідних сигналів, результатів логічних операцій, лічильників тощо.

У разі будь-якої розбіжності результатів або виявлення внутрішньої несправності система автоматично переходить у безпечний (зупинений) стан.

Для безпечних цифрових виходів використовується принцип подвійного керування виконавчими механізмами – кожна підсистема може незалежно відключити обладнання. Це означає, що навіть у разі відмови одного каналу інший забезпечить негайну зупинку процесу, запобігаючи аварійній ситуації.

Висока надійність завдяки надлишковій структурі (Redundancy) та самодіагностика апаратних і програмних помилок надає можливість їх сертифікації за міжнародними стандартами безпеки (наприклад, SIL або PL). Окремі модулі дозволяють програмувати функції безпеки без потреби у додаткових реле чи апаратних блоках. Прикладом може слугувати контролери безпеки Modicon MCM від компанії Schneider Electric (рис. 3.12).



Рис. 3.12. Контролери безпеки Modicon MCM від компанії Schneider Electric

3.3. Введення в ПЛК експлуатацію.

Введення програмованого логічного контролера (ПЛК) в експлуатацію є завершальним етапом створення системи автоматизації, під час якого перевіряється правильність монтажу, надійність електричних з'єднань і коректність роботи програмного забезпечення. Перед запуском системи здійснюється детальний огляд усіх кабельних ліній, що з'єднують контролер з елементами об'єкта керування, з метою переконатися у відповідності монтажу технічним вимогам і нормам безпеки. Особливу увагу приділяють перевірки відповідності напруги живлення налаштуванням контролера та стану захисних

пристроїв, які мають бути відрегульовані на необхідні параметри спрацьовування.

Обов'язковою частиною випробувань є перевірка працездатності аварійних вимикачів і кнопок екстреної зупинки, що гарантують безпечне відключення обладнання у разі небезпечної ситуації. Далі перевіряють правильність підключення усіх вхідних та вихідних сигналів, відповідність фізичних адрес у схемі налаштуванням контролера, а також коректність обміну даними між усіма пристроями системи.

Завершальним етапом є завантаження керуючої програми у пам'ять ПЛК та її послідовне тестування. Під час цього перевіряється, чи реагує система на всі команди відповідно до алгоритму, чи відображаються сигнали на індикаторах правильно, і чи спрацьовують виконавчі механізми у заданій послідовності. Лише після успішного проходження всіх цих процедур система може вважатися повністю готовою до промислової експлуатації.

Перевірка вхідних і вихідних сигналів у системі ПЛК є одним із ключових етапів наладки, що забезпечує достовірність обміну інформацією між контролером і обладнанням. Для тестування вхідних пристроїв, таких як перемикачі або датчики, створюють умови замикання та розмикання контактів і спостерігають за індикацією на модулі входів. Світлодіод має світитися при замкненому вході та гаснути при розімкненому. Якщо індикатор не реагує, причиною може бути несправність самого датчика, помилка у схемі підключення, відсутність живлення на пристрої або дефект модуля введення.

Перевірка вихідних сигналів зазвичай здійснюється за допомогою ручних кнопок або вимикачів, що дозволяють короткочасно вмикати виконавчі механізми, не створюючи небезпечних ситуацій. Таким чином перевіряється, чи подається сигнал із ПЛК на потрібний вихід і чи коректно реагує на нього обладнання.

Окрім фізичних перевірок, існує програмний метод контролю – форсування. Він полягає у примусовому змінюванні стану вхідних або вихідних сигналів без механічного втручання. Через програмний інтерфейс оператор задає логічне значення наприклад, TRUE або FALSE для конкретної адреси входу чи виходу. Це дозволяє перевірити не лише правильність адресації, а й реакцію системи на подану команду.

В програмних середовищах, наприклад EcoStruxure Machine Expert - Basic для цього використовують меню (команду) Force (Unforce), де оператор задає адресу потрібного сигналу та обирає значення, тобто дозволяє примусово встановити стан входу або виходу. Наприклад, при форсуванні виходу %Q0.1 може автоматично активуватись інший вихід %Q0.3, або при поданні примусового сигналу на вхід %I0.1 зміниться стан логічної схеми (рис. 3.13).



Рис. 3.13. Схема релейно-контактної логіки з форсованим виходом (а), форсованим входом (б)

У більш складних програмах, зокрема в тих, що реалізовані у вигляді послідовних функціональних схем (SFC), форсування може застосовуватися для перевірки переходів між етапами (рис. 3.14). Якщо примусово встановити перехід як хибний, програма зупиниться на поточному кроці, не переходячи далі; якщо як істинний навпаки, буде змушена перейти до наступного стану незалежно від умов.

Через потенційну небезпеку така перевірка вимагає особливої обережності: примусові сигнали можуть спричинити неочікуване ввімкнення механізмів, що здатне пошкодити обладнання або створити ризик для людей. Крім того, після завершення випробувань необхідно обов'язково скасувати всі активні форсування, інакше окремі елементи системи можуть залишитись у хибному стані навіть після відновлення нормального режиму роботи.

Більшість програмованих логічних контролерів оснащено спеціальними програмами перевірки коректності коду. Таке програмне забезпечення здійснює автоматичний аналіз завантаженої програми, виявляючи помилки у використанні адрес пристроїв, некоректні налаштування таймерів і лічильників, а також дублювання вихідних сигналів. Результати перевірки можуть відображатися на екрані або виводитися у вигляді звіту, який містить повний перелік усіх використаних входів, виходів, допоміжних реле, таймерів і

лічильників. Наприклад, система може повідомити, що певна вихідна адреса використовується в програмі більше одного разу, лічильник не має значення скидання або таймер працює без попередньо заданого часу.

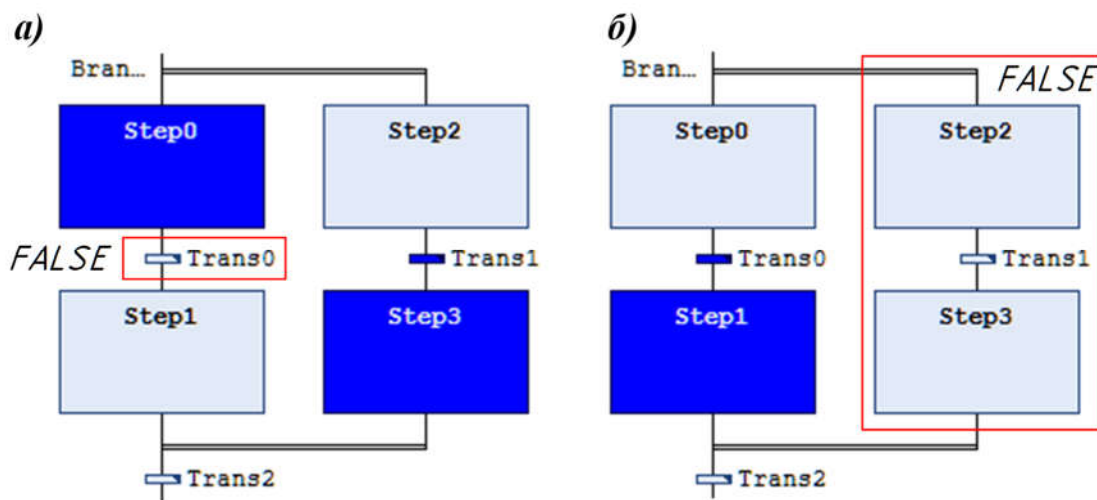


Рис. 3.14. Послідовні функціональні схеми з форсованим виходом (а), форсованим входом (б)

Сучасні ПЛК часто мають вбудований симулятор, який дозволяє перевіряти роботу програми без підключення до реального обладнання. Такий симулятор безпосередньо записує та зчитує інформацію з пам'яті входів і виходів, імітуючи їхні сигнали. Це дає змогу запустити програму, протестувати логіку її роботи, перевірити правильність змін станів та коректність усіх попередньо заданих параметрів.

Щоб виконати симуляцію, оператор повинен перевести термінал у спеціальний режим. В контролерах та програмному забезпеченні від компанії Schneider Electric цей режим називається Simulation. в цьому режимі можна в реальному часі спостерігати за роботою програми, станом логічних елементів і сигналів.

Так, наприклад, у моніторинговому режимі на екрані відображається частина програми у вигляді схеми релейно-контактної логіки, де видно, які входи відкриті або замкнені, а які виходи активні чи неактивні. На певному етапі роботи (як показано умовно на рис. 3.15) видно, що для гілки з входами IN0, IN1 і IN2 сигнал відсутній, оскільки допоміжний біт M0 не активований, і, відповідно, вихід OUT0 не спрацьовує. У наступному рядку таймер T0 відпрацював свій інтервал часу, тому його контакти замкнені, але вихід OUT1 залишається вимкненим, бо OUT0 ще не активний. Якщо ж примусово встановити сигнал на вході M0, програма одразу реагує вихід OUT0 переходить у стан ON, що спричиняє активацію OUT1.

Таким чином, засоби симуляції ПЛК є ефективним інструментом для налагодження, тестування та перевірки логіки програми ще до її впровадження на реальному об'єкті, зменшуючи ризик помилок і підвищуючи надійність системи керування.

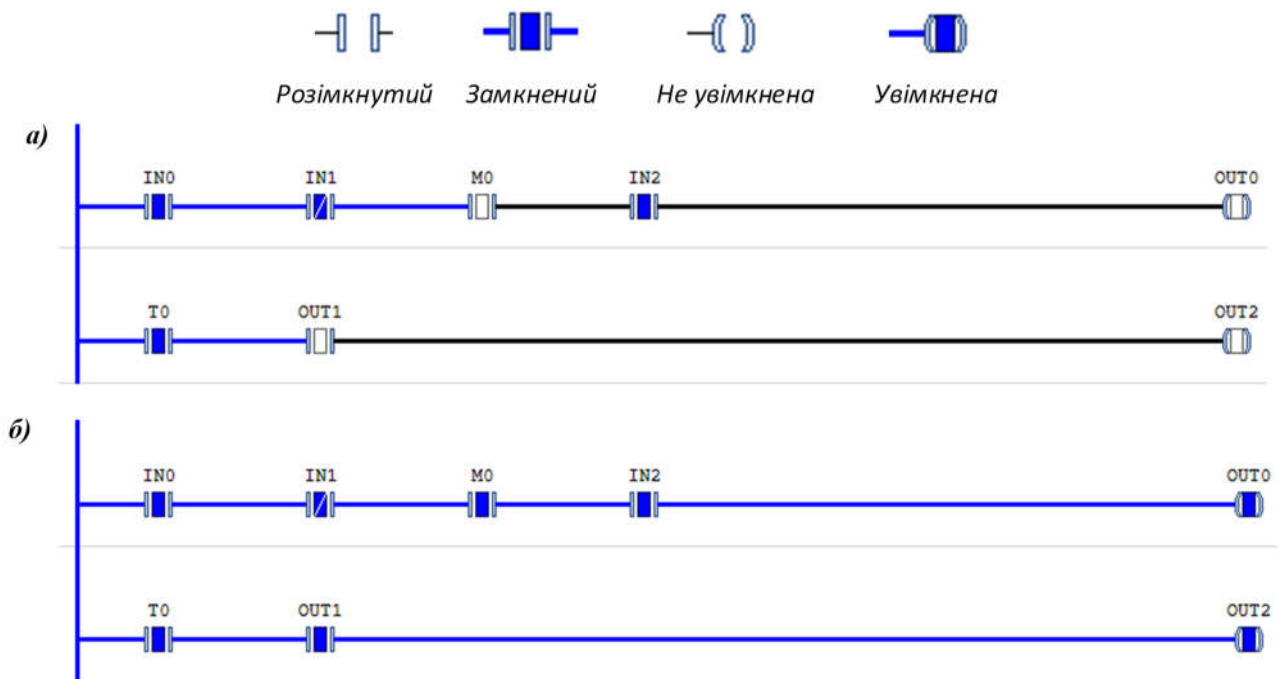


Рис. 3.15. Моніторинг виконання схем релейно-контактної логіки

3.4. Пошук несправностей.

У будь-якій системі, що керується ПЛК, більшість відмов зазвичай пов'язана не з самим ПЛК, а з периферійними елементами: датчиками, виконавчими механізмами або проводкою. Серед внутрішніх несправностей контролера найчастіше виходять з ладу модулі введення/виведення чи джерело живлення, тоді як процесор (CPU) зазвичай залишається справним.

Наприклад, якщо вихідний пристрій не вмикається, хоча індикатор виходу на ПЛК горить, спочатку перевіряють напругу на виході. Якщо вона в нормі, то ймовірною причиною є обрив або коротке замикання в проводці, або ж несправність самого пристрою. Якщо ж напруга присутня і на клеммах пристрою, тоді очевидно, що проблема саме у виконавчому елементі. Інша типова ситуація – відмова всіх входів одночасно. Таке явище часто свідчить про коротке замикання чи замикання на землю в одному з ланцюгів. У цьому випадку для пошуку дефекту входні сигнали по черзі від'єднують, доки несправний не буде знайдений. Якщо ж перестав працювати вся система, перш за все перевіряють живлення бо можливе його вимкнення, спрацювання автоматичного вимикача або відмова мережі.

Багато сучасних ПЛК мають вбудовані засоби діагностики, які виконують самоперевірку, виявляють помилки й виводять на екран код несправності разом із коротким описом. Розшифровка коду здійснюється за допомогою довідкової таблиці, яка вказує на джерело проблеми (наприклад, певний модуль) і можливі шляхи усунення: замінити модуль або перезапустити живлення системи.

Методи виявлення несправностей.

Одним із ключових засобів контролю є таймінгові перевірки, або так звані WatchDog Timers (WDT). Вони стежать за тим, щоб програма виконувала певні дії за нормальний проміжок часу. Якщо цикл перевищує допустиму тривалість,

вважається, що система вийшла з ладу, і спрацьовує аварійне відключення або сигнал тривоги. У самих ПЛК таймер спрацьовує на основі порівняння фактичного часу виконання сканування з очікуваним. Якщо цикл затримується, контролер автоматично переходить у безпечний режим. У програмі користувач також може додати таймер для контролю роботи окремих функцій – наприклад, руху поршня у пневмоциліндрі (рис. 3.16). Якщо хід поршня триває довше встановленого часу, система зупиняє процес і подає сигнал про аварію.

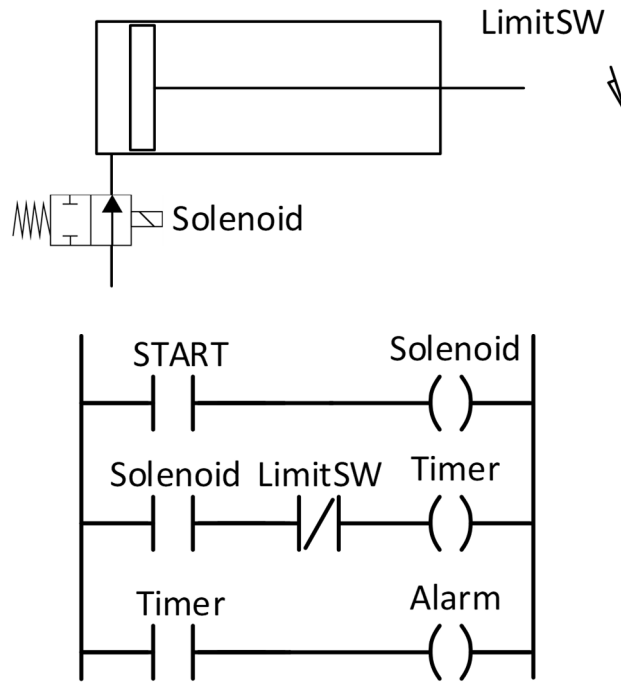


Рис. 3.16. WatchDog Timers

Інший прийом індикація останнього виходу, або Last Output Set. У цьому випадку кожна вихідна операція супроводжується увімкненням сигнальної лампи або світлодіода, що дозволяє швидко визначити, на якому етапі процес зупинився (рис. 3.17). Це особливо корисно для послідовних механізмів, наприклад у пневмосистемі, де декілька циліндрів працюють поетапно (рис. 3.18). У разі несправності, скажімо, заклинювання штока, можна миттєво побачити, на якому етапі послідовності сталася зупинка.

Для систем підвищеної надійності застосовують метод реплікації – дублювання процесів або цілих контролерів. Найпростіша форма подвійне виконання кожної операції з порівнянням результатів: якщо вони збігаються, система вважається справною. У більш складних і дорогих рішеннях використовують два незалежні ПЛК, які працюють паралельно; при найменшій розбіжності результатів система автоматично переходить у безпечний стан.

Ще один метод – перевірка очікуваних значень (Expected Value Checks). Програма аналізує, чи відповідає результат дії заданим параметрам. Якщо сигнал або результат не відповідає очікуваному, система реєструє відмову.

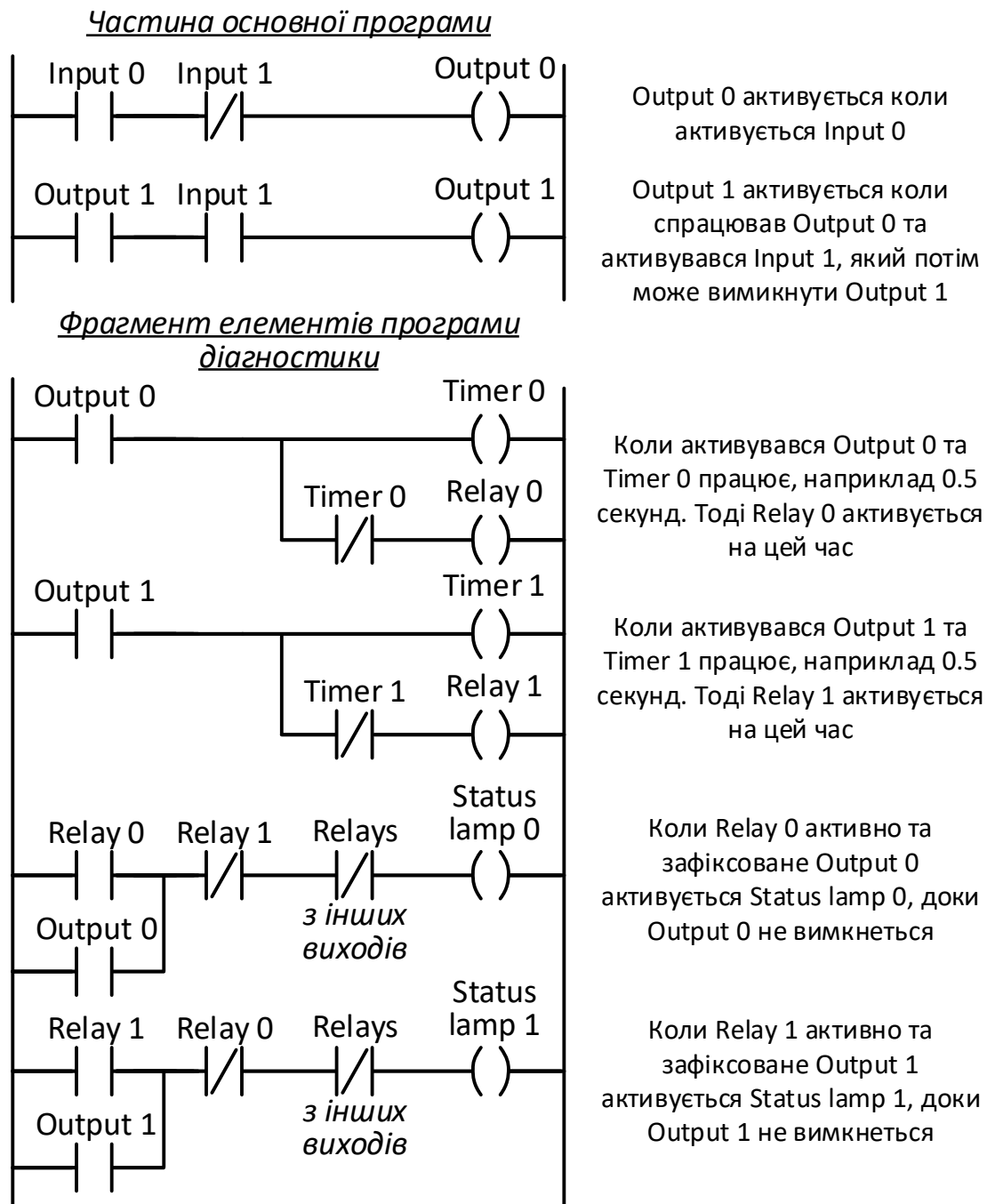


Рис. 3.17. Діагностика програми – індикація останнього виходу

Важливим аспектом надійності є збереження програми. Програмне забезпечення ПЛК зазвичай зберігається в оперативній пам'яті з батарейним резервом (Battery-Backed RAM). У разі розряду або відмови батареї програма повністю стирається. Надійнішою альтернативою є EPROM, де дані залишаються незмінними навіть після втрати живлення. Однак і в цьому випадку передбачено створення резервної копії програми, яка зберігається на жорсткому диску, USB-Flash або у вигляді знімного модуля пам'яті (SD). У разі збою вона може бути швидко перезаписана в контролер без потреби повторного програмування.

Таким чином, система технічної діагностики ПЛК охоплює апаратні, програмні та часові перевірки, що разом забезпечують високу надійність, зручність обслуговування та швидке відновлення роботи після збоїв.

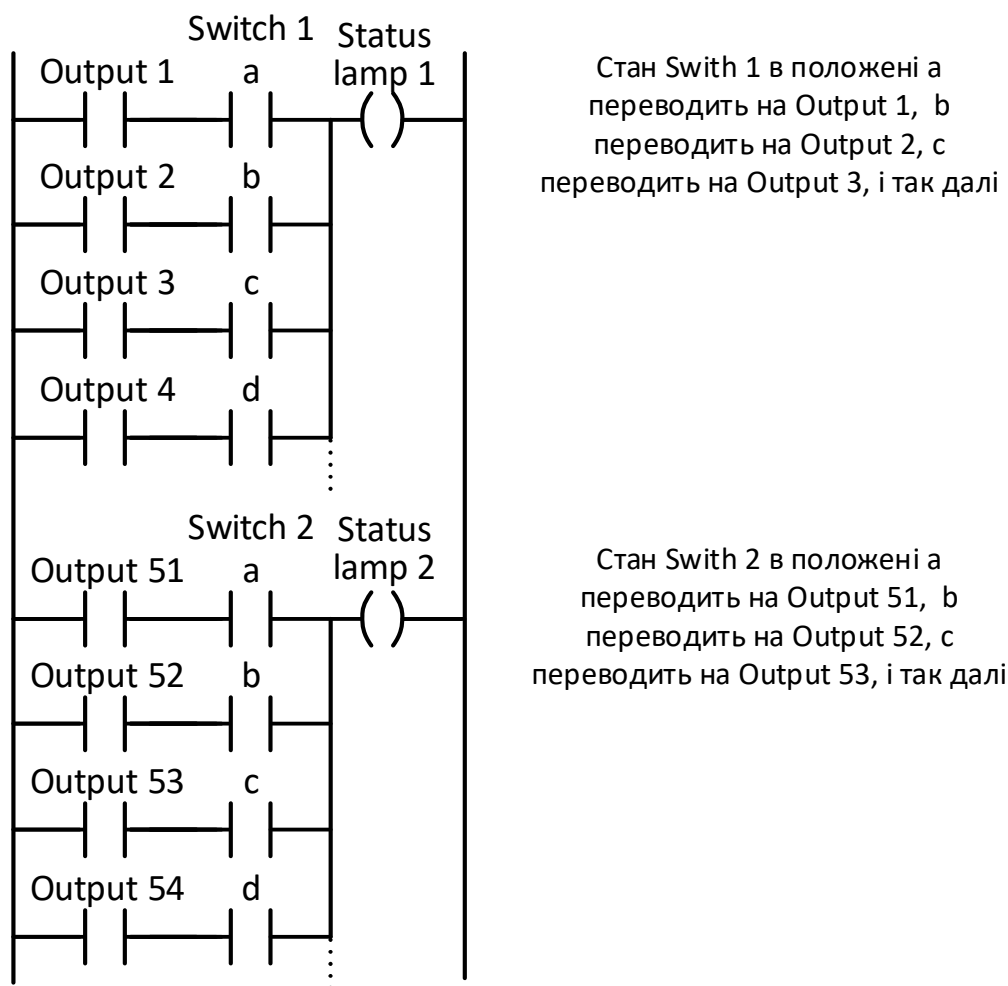


Рис. 3.18. Сигналізація виконання групи команд

Висновки за розділом.

1. Проєктування систем автоматизації базується на створенні програми, що визначає послідовність технологічних операцій та логіку керування обладнанням. Важливим є правильний вибір архітектури ПЛК, мови програмування та структуроване представлення алгоритмів для забезпечення надійності роботи системи.
2. Безпечність системи автоматизації забезпечується комплексом технічних і програмних рішень. До них належать застосування аварійних реле, реалізація функцій безпеки, використання контролерів безпеки та дотримання стандартів, що регламентують проєктування і експлуатацію промислових систем.
3. Після завершення проєктування проводиться етап налаштування, перевірки взаємодії компонентів і тестування програми в реальних умовах. Це дозволяє

забезпечити стабільну роботу системи, перевірити її відповідність технічним вимогам і мінімізувати ризики помилок.

4. Ефективне функціонування системи автоматизації передбачає постійний моніторинг і діагностику стану обладнання. Використання вбудованих засобів ПЛК, журналів подій і програмних інструментів налагодження сприяє швидкому виявленню відхилень та підтриманню безперервності технологічного процесу.

Питання для самоперевірки.

1. Які основні етапи включає процес розробки програми для ПЛК і які вимоги висуваються до структури та логіки програми?
2. Які принципи потрібно враховувати при проектуванні систем автоматизації з точки зору безпеки?
3. У чому полягає роль ПЛК у забезпеченні безпеки технологічного процесу?
4. Яке призначення реле аварійної зупинки та як воно інтегрується в систему керування?
5. Що розуміють під функціями безпеки (Safety Functions) у системах автоматизації та які приклади таких функцій можна навести?
6. Чим контролери безпеки відрізняються від стандартних ПЛК і в яких випадках їх доцільно використовувати?
7. Які основні дії виконуються під час введення ПЛК в експлуатацію? Які перевірки здійснюються на цьому етапі?
8. Як проводиться пошук несправностей у системах автоматизації, які засоби діагностики використовуються?
9. Яке значення має технічна документація при проектуванні системи автоматизації і які документи входять до її складу?
10. Як взаємопов'язані етапи проектування, налагодження, експлуатації та технічного обслуговування систем автоматизації?

РОЗДІЛ 4.

ЕЛЕКТРОПНЕВМАТИЧНІ СИСТЕМИ АВТОМАТИЗАЦІЇ

Електропневматика сьогодні широко застосовується в багатьох галузях промислової автоматизації, особливо там, де потрібні недорогі та гнучкі рішення. Такі системи активно використовуються у виробництві, збиранні, фармацевтичній, хімічній та пакувальній промисловості.

Електропневматичне керування поєднує електричну систему управління з пневматичною системою виконавчих механізмів. Соленоїдні клапани при цьому виступають інтерфейсом між електричною та пневматичною частинами. Датчики положення, кінцеві вимикачі та індуктивні/ємнісні сенсори використовуються як елементи зворотного зв'язку.

Цей тип керування інтегрує електричні й пневматичні технології, що робить його придатним для великих і складних систем.

У таких системах сигнали керування передаються електричним струмом (постійним або змінним), тоді як робочим середовищем залишається стиснене повітря. Робоча напруга зазвичай становить від 12 до 220 В.

Кінцевий елемент керування це клапан, який спрацьовує за допомогою соленоїда.

Повернення клапана у вихідне положення відбувається або пружиною (в односоленоїдних клапанах), або другим соленоїдом (у двосоленоїдних). Часто застосовується пілотне керування, що дозволяє зменшити розміри та вартість клапанів.

Керування електропневматичною системою може здійснюватися двома способами:

- за допомогою реле та контакторів, які формують і перетворюють сигнали від датчиків;
- за допомогою ПЛК, який дозволяє реалізувати необхідну логіку, часові затримки та послідовність операцій.

Отримані вихідні сигнали подаються на соленоїди, які активують клапани, керуючи рухом пневмоциліндрів або іншого виконавчого обладнання.

Найбільша перевага електропневматики полягає у можливості інтеграції різних типів електричних датчиків і ПЛК, що забезпечує високу швидкість, точність і надійність керування. Електричні сигнали передаються значно швидше, ніж пневматичні, тому зменшується цикл роботи системи та збільшується дистанція передавання сигналів.

Основні етапи роботи електропневматичної системи:

- введення сигналу – генерація сигналів від перемикачів, кнопок, датчиків чи контакторів;
- обробка сигналу – здійснюється комбінацією реле, контакторів або за допомогою ПЛК;
- вихідні сигнали – після обробки подаються на соленоїди, світлові індикатори або звукові сповіщувачі.

Пневматичні пристрої поділяються на дві основні категорії:

- пристрої для отримання та транспортування стисненого повітря;
- пристрої, що споживають стиснене повітря.

Робочі пневматичні елементи, або виконавчі механізми, є кінцевою ланкою пневматичної системи, оскільки саме вони виконують механічну роботу та реалізують задані операції. Ті з них, що здійснюють переміщення або обертальні рухи зворотно-поступального характеру, називаються циліндрами.

4.1. Пристрої для отримання та транспортування стисненого повітря.

4.1.1. Компресори.

Сьогодні стиснене повітря є невід'ємною складовою більшості промислових процесів. Машини, що його виробляють, називаються *компресорами*. Вони, приводячись у дію двигунами, всмоктують атмосферне повітря, стискають його та подають споживачам. Хоча фахівці з автоматизації зазвичай не займаються безпосередньо питаннями компресії чи виробництва повітря, базові знання в цій галузі необхідні для розуміння принципів пневмоавтоматизації.

Перед розглядом типів компресорів слід окреслити основні пневматичні параметри:

- тиск на всмоктуванні (P_a),
- тиск на нагнітанні (P_e),
- об'ємна витрата при всмоктуванні (Q),
- об'ємна витрата на виході (Q_l),
- ступінь стискання ($r = P_e/P_a$).

Поняття «нормальний об'єм» (позначення n) означає повітряний об'єм, приведений до стандартних умов – температури 20 °C та атмосферного тиску. Потужність, необхідну для стискання повітря до певного тиску, можна визначити за відповідною формулою:

$$N = Q \times P_a \times 3.5 \times (r^{2.85} - 1),$$

а ефективність роботи компресора зменшується зі збільшенням ступеня стискання. За нормальних умов коефіцієнт корисної дії не повинен бути нижчим за 0.7 при $r < 5$.

Компресори поділяються на *об'ємні* та *динамічні*. У пневматичних системах найчастіше використовують *об'ємні компресори*, які бувають *поршневими* та *ротаційними*.

Поршневі компресори застосовують для створення низького, середнього й високого тиску. Для підвищених тисків використовують багатоступеневі схеми. Робота компресора базується на зворотно-поступальному русі поршня в циліндрі, який приводиться в дію кривошипно-шатунним механізмом (рис. 4.1). Потік повітря контролюють два клапани, на всмоктуванні та нагнітанні. Охолодження здійснюється повітрям або рідиною. Через переривчастий режим роботи компресор обов'язково під'єднують до ресивера (повітряного бака), який

накопичує повітря. Коли тиск у баку досягає верхньої межі, реле тиску вимикає двигун; при зниженні тиску до нижнього порогу знову запускає.

Ротаційні компресори поділяють на *лопатеві*, *гвинтові (спіральні)* та *ротаційно-поршневі*. У *лопатевому компресорі* (рис. 4.2) ротор (2) обертається ексцентрично в циліндричному корпусі (статорі) (1), а в його пазах (3) розміщені тонкі сталеві пластини (4), які під дією відцентрової сили висуваються й утворюють камери стиснення. Повітря засмоктується (5), стискується під час обертання ротора, а потім виштовхується через вихідний отвір (6). Такі компресори працюють безперервно навіть коли споживання повітря відсутнє, вони переходять у режим холостого ходу, закриваючи впускний клапан. Завдяки системі автоматичного регулювання всмоктування ці компресори можуть працювати без ресивера.

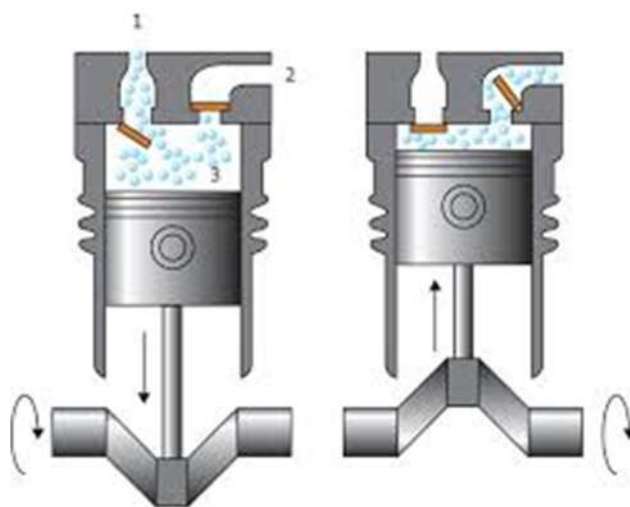


Рис. 4.1. Поршневий компресор

Гвинтові компресори мають схожий принцип, але відрізняються низьким рівнем шуму через мінімальну кількість деталей, що контактують. Вони містять два паралельні ротори зі спіральними виступами, які обертаються назустріч один одному (рис. 4.3). Повітря стискується завдяки зменшенню об'єму між витками гвинтів, переміщуючись від входу до виходу. Для запобігання витокам і охолодження необхідне інтенсивне мастило. Як і лопатеві, гвинтові компресори можуть працювати в режимі холостого ходу. Їхні робочі тиски зазвичай не перевищують 10 бар.

Ротаційно-поршневі компресори мають нижчу ефективність і використовуються рідше. Вони створюють тиск до 3 бар, але з невеликою подачею. Два жорстко з'єднані ротори з лопатями при обертанні переміщують повітря з входу до виходу (рис. 4.4). Стиснення відбувається лише при виході, тому такі компресори мають обмежену здатність створювати високий тиск.

4.1.2. Ресивери (повітряні баки).

Ресивер виконує функцію накопичення стисненого повітря та забезпечення його подачі до споживачів під час пауз у роботі компресора. Він також сприяє

осадженню домішок: пилу, масла, конденсату, які осідають на дні й періодично видаляються автоматичним дренажним клапаном (рис. 4.5).

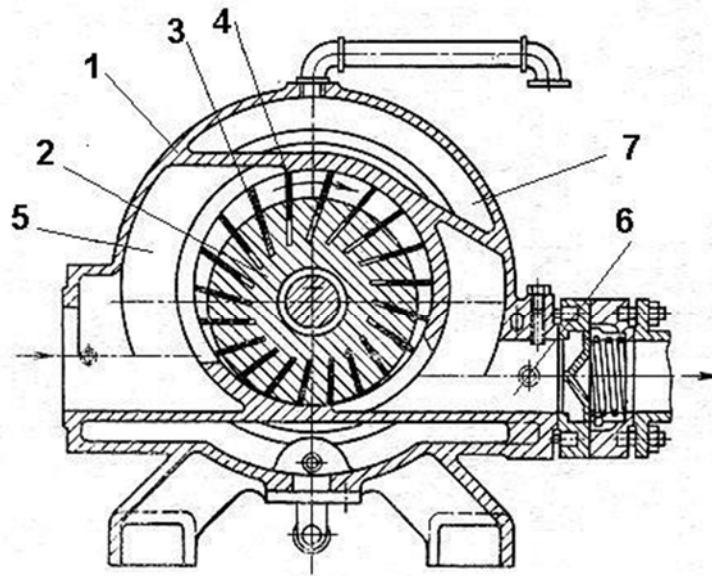


Рис. 4.2. Роторно-лопатевий компресор

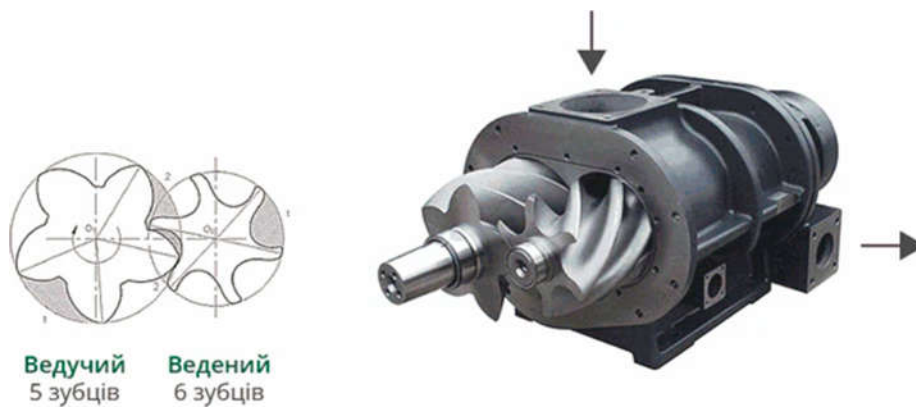


Рис. 4.3. Гвинтовий компресор



Рис. 4.4. Ротаційно-поршневий компресор

Об'єм ресивера визначається залежно від продуктивності компресора:

$$C = \frac{Q}{60},$$

де C – об'єм бака (м^3);
 Q – подача повітря ($\text{м}^3/\text{год}$).



Рис. 4.5. Ресивери

Наприклад, для поршневого компресора з витратою $50 \text{ м}^3/\text{год}$ потрібен бак об'ємом приблизно 0.83 м^3 (830 л), отже вибирають стандартний бак 1000 л.

Для ротаційного компресора, який працює з регульованим всмоктуванням, розрахунок виконується за формулою:

$$C = \frac{Q}{600},$$

і за такої ж витрати достатньо бака ≈ 100 л.

Баки об'ємом понад 25 л підлягають обов'язковому технічному випробуванню на міцність і безпечність.

4.1.3. Підготовка повітря.

Під час стискання повітря його температура зростає, і воно може вміщувати більше водяної пари. Якщо таке гаряче повітря подати безпосередньо в мережу, при охолодженні у трубопроводах відбудеться конденсація вологи, що призведе до появи води в пневмосистемі.

Для запобігання цьому між компресором і ресивером встановлюють фінальні охолоджувачі (aftercoolers), які знижують температуру повітря та осаджують вологу. Вони бувають: водяним або повітряним охолодженням, холодильним циклом.

Надалі стиснене повітря потребує додаткової підготовки, щоб стати придатним для роботи пневматичного обладнання, йому необхідне очищення від твердих домішок і пилу за допомогою фільтрації, а також про зниження та стабілізацію тиску до постійного значення, нижчого за той, що є в розподільній

мережі. За потреби до потоку додають дрібнодисперсну масляну мікротуманність для змащування рухомих елементів пристроїв.

Типовий пристрій підготовки повітря складається з фільтра, регулятора тиску з манометром і, за необхідності, змащувача (рис. 4.6). Повітря, крім водяної пари, часто містить тверді частинки та випари оливи, що утворюються під час роботи компресора. Завдання фільтра остаточно очистити повітря після попереднього грубого очищення у вхідних фільтрах системи.

Повітря потрапляє у верхню частину фільтра, де спеціальні лопатки надають йому обертального руху. Під дією відцентрової сили краплі води та великі домішки відкидаються до стінок склянки й осідають унизу. Нижня частина колби відділена перегородкою, що запобігає підсосу вологи назад у потік. Перед виходом очищене повітря проходить через фільтруючий елемент, який затримує дрібні частинки. Ступінь очищення залежить від пористості картриджа: наприклад, фільтр із порами 50 мікрон затримує частинки діаметром 50 мікрон і більше, а для більш тонкої фільтрації застосовують елементи з порами 20 або навіть 5 мікрон.



Рис. 4.6. Фільтр-регулятор Syntesi FR SY1 від компанії Metal Work

Корпуси фільтрів виготовляють із прозорого ударостійкого полікарбонату чи нейлону, а для більших або спеціальних установок із металевим захистом. Внизу колби розташовують пристрій для видалення конденсату ручний або автоматичний.

Для особливо чистого повітря використовують коалесцентні мікрофільтри, які видаляють залишкові частинки оливи та вологи. Вони працюють за принципом коалесценції: дрібні аерозольні краплі злипаються у більші, що осідають унизу. Такі фільтри мають пористість 0.1 мікрона й забезпечують очищення повітря на 99.97 %, але потребують встановлення попереднього фільтра з пористістю 5 мікрон, щоб уникнути передчасного забруднення картриджа.

Фільтри потребують регулярного обслуговування – заміни картриджів і зливу накопиченої рідини. Забитий фільтр збільшує втрати тиску, тому його потрібно підбирати відповідно до витрати повітря та допустимого перепаду тиску. Для стабільної роботи фільтра достатньо падіння тиску близько 0.1 бар. У

технічних довідниках виробники подають графіки залежності витрати від перепаду тиску, які допомагають правильно вибрати тип пристрою (рис. 4.7).

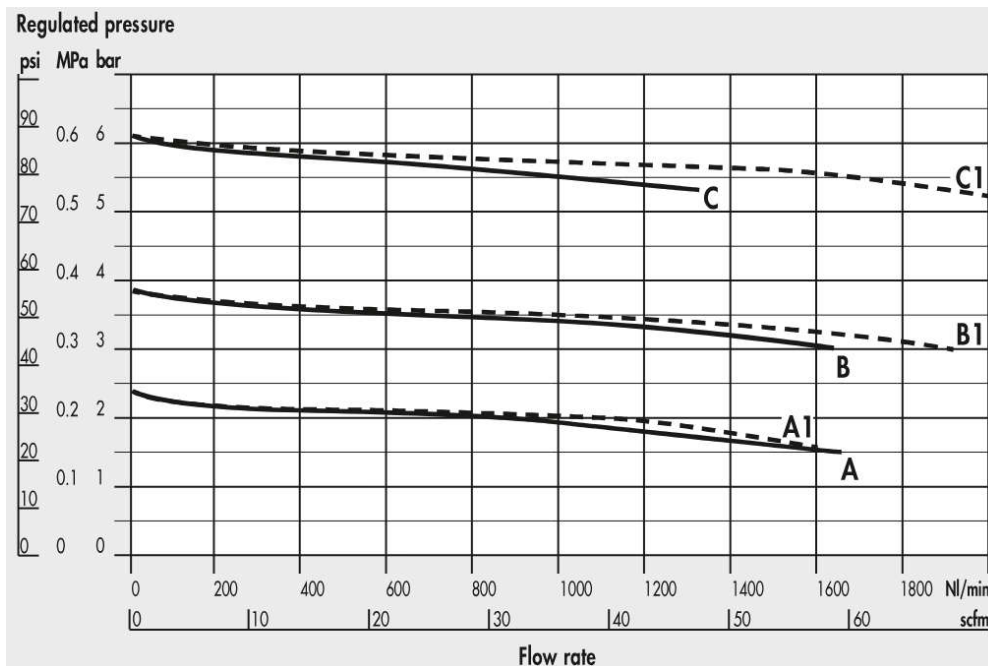


Рис. 4.7. Залежності витрати від перепаду тиску для фільтр-регулятора Syntesi FR SY1: А – 7(2.5) бар; В – 7(4) бар; С – 7(6.3) бар; А1 – 10(2.5) бар; В1 – 10(4) бар; С1 – 10(6.3) бар

4.1.4. Регулятори тиску.

Редуктор тиску – це пристрій, призначений для зниження та стабілізації повітряного тиску в системі. Його принцип дії ґрунтується на підтриманні вихідного тиску, пропорційного опорному сигналу. Такий елемент є необхідним у будь-якій пневматичній системі, адже забезпечує стабільну подачу повітря на споживачів, що гарантує коректну роботу обладнання.

Зазвичай опорний сигнал створюється силою, яка виникає при стисканні пружини регулювальним гвинтом. Чим сильніше натискає пружина, тим вищим є вихідний тиск (рис. 4.8).



Рис. 4.8. Регулятор потоку Push-Lock серії On Line від компанії Metal Work

Під час налаштування редуктора тиску (рис. 4.9) оператор обертає гвинт, стискаючи регулювальну пружину, яка через мембрану тисне на шток, з'єднаний

із клапаном-заслінкою. Це відкриває канал між входом і виходом повітря. Вхідний тиск позначають як P_1 , а вихідний – як P_2 . Коли тиск P_2 підвищується, він починає діяти на нижню поверхню мембрани, урівноважуючи зусилля пружини. Досягнувши рівноваги, клапан закривається. Якщо ж під час роботи споживання повітря зростає, тиск P_2 падає, і клапан знову відкривається, компенсуючи витрати. У стані спокою, без споживання, клапан залишається закритим.

Якщо з певних причин тиск на виході P_2 перевищує задане значення, мембрана піднімається, відкриваючи отвір для стравлення надлишку повітря в атмосферу. Така функція називається розвантаженням (relieving).

Коли витрата повітря значно зростає, клапан відкривається ширше, і пружина розтягується. У цьому випадку сила, яку вона чинить, стає меншою, ніж у початковому положенні, і баланс між зусиллям пружини та тиском P_2 зміщується вихідний тиск зменшується. Щоб компенсувати це падіння, у конструкції редуктора застосовується ефект Вентурі.

Згідно з принципом Вентурі, коли рідина або газ проходить крізь звуження трубопроводу, швидкість потоку збільшується, а тиск у цій зоні зменшується. Після звуження, коли перетин знову розширюється, тиск повертається до попереднього значення. У редукторі аналогічно створюють додаткову камеру P_3 , сполучену з вихідним каналом P_2 через малий отвір у звуженій частині. Завдяки цьому тиск під мембраною стає дещо нижчим, ніж фактичний тиск P_2 , і пружина відкриває клапан ширше, забезпечуючи більшу витрату повітря. Для підсилення ефекту Вентурі в центр потоку іноді вставляють трубку з косим зрізом, спрямовану до виходу.

Щоб коливання вхідного тиску P_1 не впливали на стабільність вихідного P_2 , у редукторі застосовують балансування клапана. Річ у тім, що при зміні P_1 змінюються сили, які діють на обидві сторони клапана, тому необхідно зрівноважити тиск на поверхнях, щоб уникнути впливу коливань мережі на регульований тиск.

Підбираючи редуктор, враховують необхідну витрату та допустиме падіння тиску. Здатність пристрою пропускати повітря залежить від його розміру ці дані вказані у технічних характеристиках виробників. Типова характеристика редуктора поділяється на три ділянки:

1. початкова, коли клапан лише злегка відкритий, регулювання майже не відбувається, реакція на зміну витрати повітря відсутня.
2. робоча зона це область стабільного регулювання, де пристрій працює коректно.
3. критична коли клапан повністю відкритий, витрата досягає максимуму, а вихідний тиск P_2 різко падає.

Саме друга зона є робочою й використовується при виборі редуктора. Першу та третю частини діаграми не враховують, оскільки вони не забезпечують стабільної роботи.

Діапазон регулювання залежить від характеристик пружини: щоб працювати в широкому діапазоні, потрібна жорсткіша пружина. Наприклад, якщо необхідно отримати вихідний тиск 1.5 бар, доцільно використати пружину

з діапазоном 0...4 бар, а не 0...12 бар. У першому випадку пружина забезпечує достатній хід клапана і стабільну роботу в зоні 2. Якщо ж застосувати занадто жорстку пружину, регулятор працюватиме в зоні 1 або 3, де стабільність регулювання погіршується.

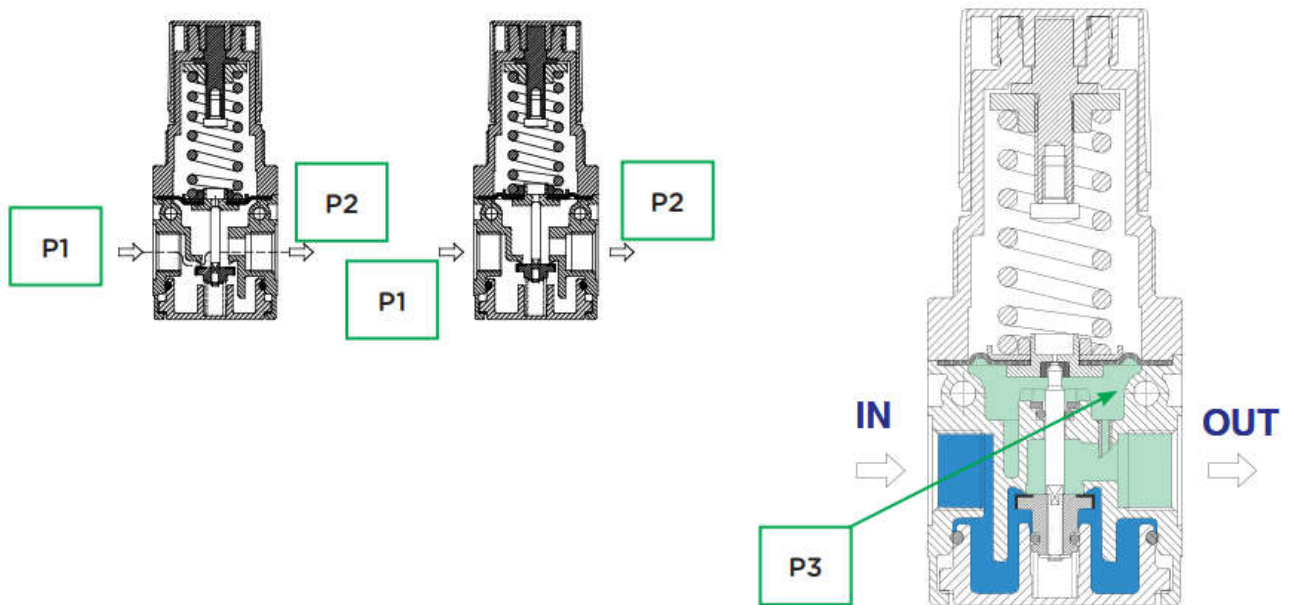


Рис. 4.9. Регулятор тиску

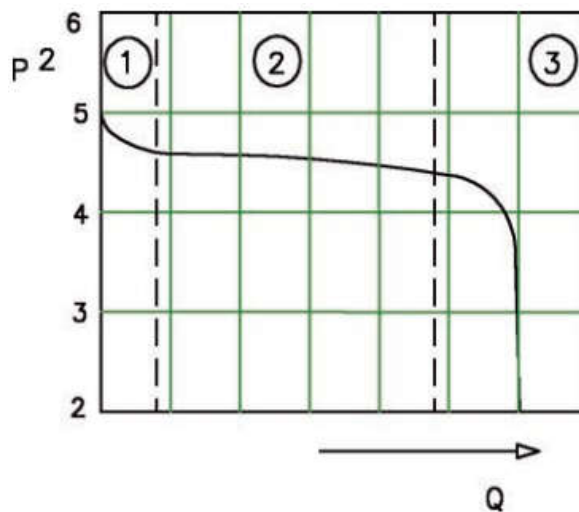


Рис. 4.10. Характеристика редуктора

Характеристичні криві, для кожного редуктора наведені в технічних довідниках, показують залежність витрати від падіння тиску P_2 при різних робочих тисках, а також зміну вихідного тиску при коливаннях тиску на вході. Саме за цими графіками визначають оптимальний робочий режим редуктора (рис. 4.11-12).

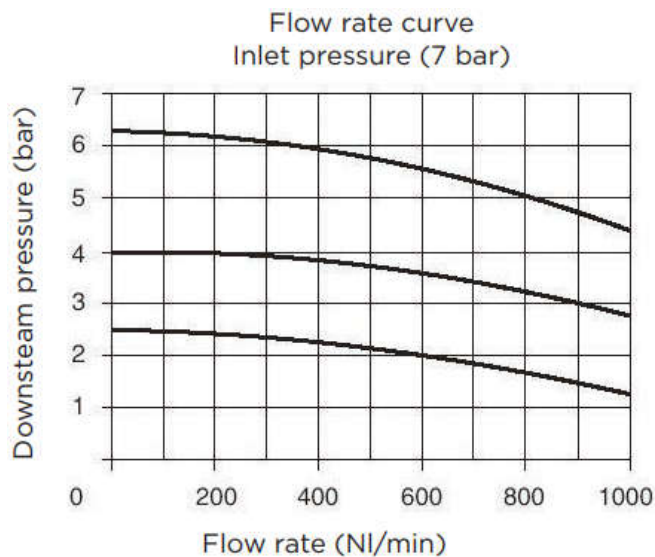


Рис. 4.11. Криві витрати та відповідні падіння P2 при різних робочих тисках

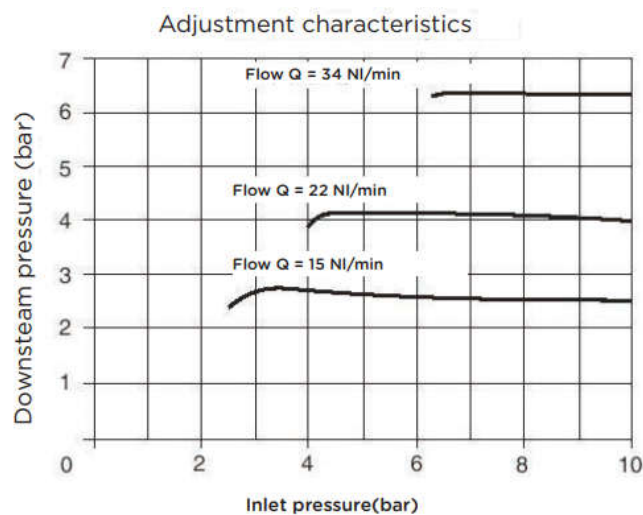


Рис. 4.11. Зміна тиску P2 на виході залежно від змін тиску P1

4.2. Пристрої пневматики та електропневматики.

Робочі пневматичні елементи, або виконавчі механізми, є кінцевою ланкою пневматичної системи, оскільки саме вони виконують механічну роботу та реалізують задані операції. Ті з них, що здійснюють переміщення або обертальні рухи зворотно-поступального характеру, називаються циліндрами.

4.2.1. Пневматичні циліндри.

Пневматичні циліндри, за умови правильного підбору їхніх параметрів, є надійними виконавчими пристроями, що не піддаються перевантаженням, забезпечують високі швидкості руху, дозволяють швидко змінювати напрямок переміщення, не впливають негативно на робоче середовище, а також створюють

зусилля та швидкості, які легко регулювати. Крім того, вони відзначаються простою конструкцією та невибагливим технічним обслуговуванням.

Конструктивно циліндр (рис. 4.12) складається з корпусу (гільзи) (2), внутрішня поверхня якого має циліндричну форму, поршня (5), з'єднаного зі штоком (3), і ущільнювальних манжет (8), що забезпечують герметичність. З обох боків корпус закритий кришками (1,7), одна з яких має центральний отвір для виходу штока (1). Кришки (або головки) механічно з'єднані з гільзою, утворюючи герметичну камеру.

Рух штока в обох напрямках здійснюється шляхом подачі стисненого повітря поперемінно у передню або задню порожнину циліндра через різьбові отвори у кришках, що сполучені з відповідними камерами. Така схема забезпечує зворотно-поступальний рух виконавчого елемента та точне керування його положенням.

Лінійні пневматичні циліндри здійснюють прямолінійний рух штока від втягнутого положення до повністю висунутого й назад. У процесі цього руху вони виконують механічну роботу, прикладаючи певну силу до точки дії.

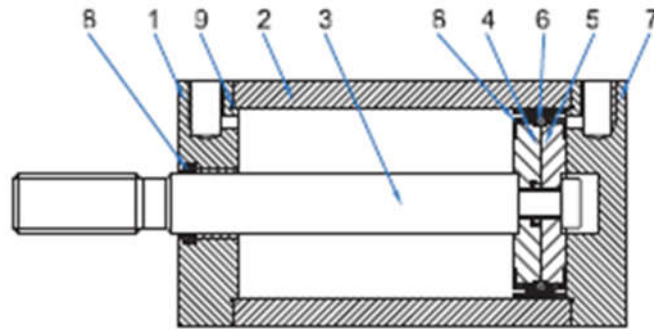


Рис. 4.12. Пневматичний циліндр

Основний принцип їхньої роботи полягає у співвідношенні між силою, тиском і площею поршня:

$$F = P \times S,$$

де F – сила;

P – тиск повітря в поршні;

S – площа поршня.

Проте сила, яку створює циліндр, різниться залежно від напрямку руху. При зворотному ході (коли шток повертається у вихідне положення) ефективна площа поршня зменшується на площу, яку займає сам шток, а отже, і створювана сила стає меншою.

Окрім того, розрахована за цією формулою сила є лише теоретичною реальна сила зменшується через необхідність подолання сил тертя та ваги рухомих частин, тобто самого поршня та штока. Тертя виникає в місцях контакту ущільнювальних манжет поршня і штока з внутрішньою поверхнею циліндра. При цьому розрізняють тертя зрушення, що пов'язане з подоланням початкової адгезії між поверхнями, та динамічне тертя, яке виникає під час руху.

Якщо поршень тривалий час залишається у певному положенні, ущільнення стискають мастильний шар між собою та стінками циліндра, частково витісняючи його. Через це під час початку руху поверхні ковзають майже без змащення, що потребує більших зусиль для старту. Згодом, коли гідродинамічні умови змащення відновлюються, величина тертя різко зменшується. Зазвичай тертя зменшується із зростанням швидкості руху.

Крім того, під час початку ходу циліндр зазнає невеликих еластичних деформацій, які створюють додатковий опір руху та знижують ефективність роботи. У середньому втрати сили через тертя та деформації становлять приблизно 15 % від теоретичного значення.

За принципом дії лінійні циліндри поділяються на два основних типи: *односторонньої* та *двосторонньої* дії.

Конструктивно будь-який циліндр складається з корпусу, двох кришок, поршня зі штоком, прямої втулки та ущільнювальних елементів. На штоку також розміщено ущільнювальне кільце зі знімачем пилу, що запобігає потраплянню забруднень у робочі зони механізму.

Пневмоциліндри односторонньої дії створюють робоче зусилля лише в одному напрямку (рис. 4.13). Повернення штока у вихідне положення відбувається за рахунок вбудованої пружини або під дією зовнішньої сили.

За принципом роботи такі циліндри поділяють на штовхаючі (які створюють зусилля під час висунання штока) та тягучі (які працюють під час втягування штока). Їх використовують у тих випадках, коли необхідно виконати операції типу затискання, виштовхування або пресування тобто дії, що не потребують закріплення навантаження на різьбі штока.



Рис. 4.13. Пневмоциліндр односторонньої дії

Пружина, яка встановлюється всередині циліндра, розрахована лише на відновлення рівноваги між поршнем і штоком після закінчення робочого циклу, а не на створення значного робочого зусилля.

Залежно від конструкції, розрізняють дві основні схеми циліндрів односторонньої дії: штовхаючого типу та тягучого типу, кожна з яких має своє умовне графічне позначення.

Особливістю таких циліндрів є обмежений робочий хід, адже наявність пружини не дозволяє реалізувати великі довжини переміщення. Пружина займає внутрішній простір корпусу, що технічно обмежує довжину ходу, хоча у випадках малих діаметрів циліндра та коротких переміщень це не становить проблеми.

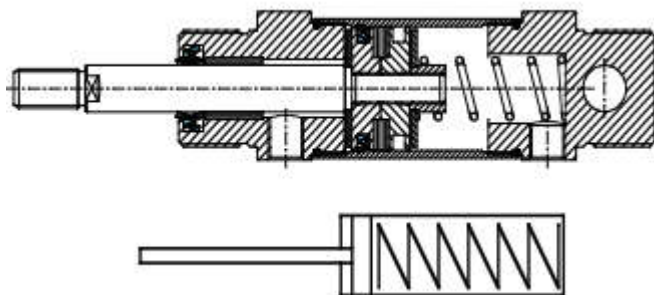


Рис. 4.13 Схема пневмоциліндру односторонньої дії

Пневмоциліндри двосторонньої дії належать до виконавчих елементів, здатних створювати як зусилля штовхання, так і зусилля тяги залежно від того, до якого боку поршня підведено стиснене повітря (рис. 4.14). Як уже зазначалося, значення цих сил різняться через різницю площ дії тиску: при русі штока вперед вона більша, ніж при його зворотному ході.

Такі циліндри застосовують у найрізноманітніших механізмах, де необхідне як висування, так і втягування штока, причому до нього може бути безпосередньо приєднано навантаження. Регулюючи швидкість подачі повітря за допомогою спеціальних дросельних пристроїв, можна точно контролювати швидкість переміщення навантаження.

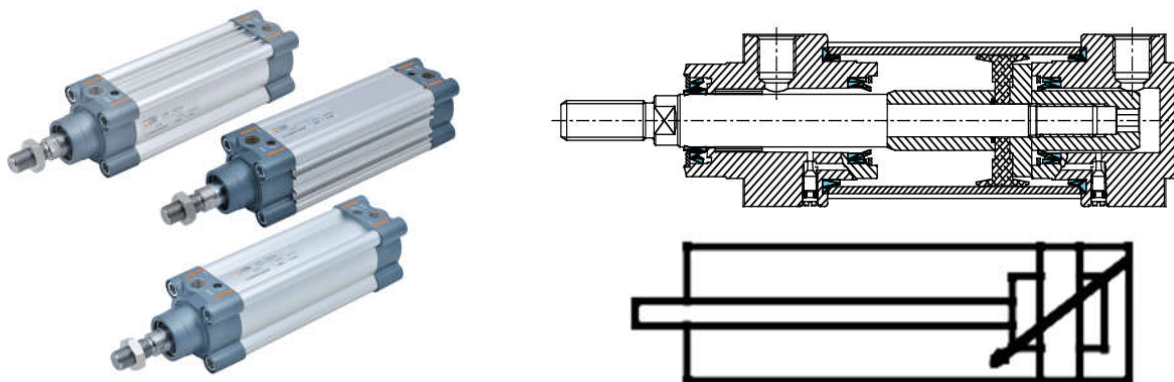


Рис. 4.14 Пневмоциліндри двосторонньої дії з повітряним гальмуванням

Зазвичай корпус циліндра сприймає кінцеве навантаження при зупинці рухомих частин, тому на його торцевих кришках передбачають обмежувачі ходу своєрідні упори. Щоб уникнути пошкодження деталей під час удару в кінці робочого циклу, у конструкції передбачають системи поглинання кінетичної енергії.

Найефективнішим способом є пневматичне демпфування: у кінцевій частині ходу поршня утворюється повітряна подушка, що плавно гальмує рух. У циліндрах малих розмірів або при невеликих швидкостях замість цього іноді застосовують пружинні шайби, розташовані біля торців поршня (рис. 4.15).



Рис. 4.15 Пневмоциліндри двосторонньої дії з еластичним/гумовим відбійником

Хід штока у двосторонньо-діючих циліндрах може бути досить великим за умови, що це відповідає вимогам механічної системи, в яку циліндр інтегровано.

Циліндри з магнітними кільцями. Існує кілька способів визначення положення поршня в пневмоциліндрі. Один із найпоширеніших полягає у виявленні магнітного поля, створюваного магнітним кільцем, установленим на поршні всередині циліндра. Принцип дії такого циліндра полягає в наступному, коли поршень переміщується всередині циліндра, зовнішній датчик положення (геркон або датчик Холла), розташований зовні корпусу, фіксує магнітне поле від кільця. У момент його виявлення формується електричний сигнал, який відповідає певному положенню поршня. Таким чином можна відстежувати його рух або подавати сигнали керуванню для зупинки, реверсу тощо. Але оскільки магнітне поле повинно проходити крізь стінку циліндра без спотворення, корпус виготовляють із немагнітних матеріалів таких як алюміній, латунь або нержавіюча сталь. Найпоширенішим матеріалом є алюміній, оскільки він легкий, корозійностійкий і має добрі механічні властивості.

4.2.2. Клапани.

Розподільчі клапани призначені для зміни напрямку потоку стисненого повітря у пневматичній системі. Це відбувається завдяки зовнішнім впливам на органи керування, які переміщують робочі елементи всередині клапана.

Основними характеристиками таких клапанів є:

- кількість ходів (шляхів);
- кількість положень;
- тип керування (механічне, електричне, пневматичне тощо).

Кількість ходів і положень.

Щоб визначити кількість ходів клапана, достатньо порахувати кількість отворів (з'єднань) у його корпусі, не враховуючи ті, що призначені для елементів керування.

Наприклад, клапан із двома отворами називається двоходовим (2-way), з трьома трьохходовим (3-way) тощо.

Кількість положень визначається кількістю фіксованих станів, у яких клапан може перебувати під дією керування, включно зі станом спокою. Наприклад, 2/2-клапан має два ходи та два положення.

Для клапанів типу 2/2 або 3/2 важливо знати, чи перекривається повітря в положенні спокою. Якщо при відсутності сигналу подача повітря заблокована, клапан вважається нормально закритим (N.C.). Якщо навпаки у положенні спокою повітря проходить вільно, а при подачі сигналу перекривається, клапан є нормально відкритим (N.O.).

Умовні графічні символи клапанів відображають кількість ходів, положень і тип керування (рис. 4.16). Кожен квадрат у символі відповідає одному положенню клапана. У середині квадратів зображується напрямок руху потоку повітря.



Рис. 4.16 Загальне схематичне позначення клапанів

Наприклад, 2/2 N.C. клапан у стані спокою має подачу (позначену числом 1) перекритою, а вихід (2) закритим. При активації сигналу повітря проходить від 1 до 2.

Для 3/2 N.C. (рис. 4.17) клапана додається третій хід злив (3). У стані спокою отвір 2 з'єднаний зі зливом 3, а при активації із подачею 1.

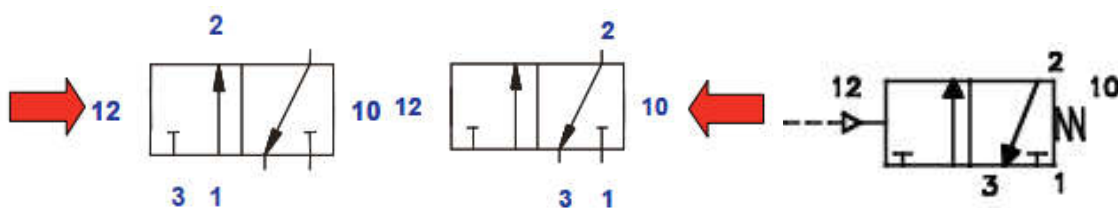


Рис. 4.17. Схема позначення клапана 3/2 N.C.

Позначення керування у схемах стандартизовані: 10 – керування, що відповідає положенню спокою; 12 – керування для другого положення.

Якщо клапан автоматично повертається у вихідне положення за допомогою пружини, він називається *моностабільним* (однопозиційним) – сигнал на виході діє лише стільки, скільки триває керуючий імпульс.

Клапани, які можуть залишатися в одному з двох положень до подачі протилежного сигналу, називають *бістабільними* (з пам'яттю), оскільки вони «запам'ятовують» останнє керування.

Типи клапанів за кількістю ходів:

- 2/2 клапани – застосовуються як прості запірні елементи (N.C. або N.O.).
- 3/2 клапани – використовуються для керування односторонньодіючими циліндрами або для передавання пневмосигналів іншим клапанам.
- 5/2 клапани – призначені для керування двосторонньодіючими циліндрами. У них дві робочі лінії (2 і 4), одна лінія подачі (1) та два виходи на злив (3 і 5).

У системах з трипозиційними клапанами (наприклад, 5/3) центральне положення має спеціальну функцію:

- «із закритими центрами» усі канали перекриті;
- «з відкритими центрами» усі канали сполучені між собою;
- «з підживленням центру» тиск подається одночасно на обидва виходи.

Центральне положення фіксується механічно або за допомогою пружин, а крайні положення досягаються подачею керуючих сигналів. Використовують стандартні графічні символи згідно з ISO 1219 або DIN ISO 1219 для позначення типів керування (пружина, електромагніт, кнопка тощо) (рис. 4.18).

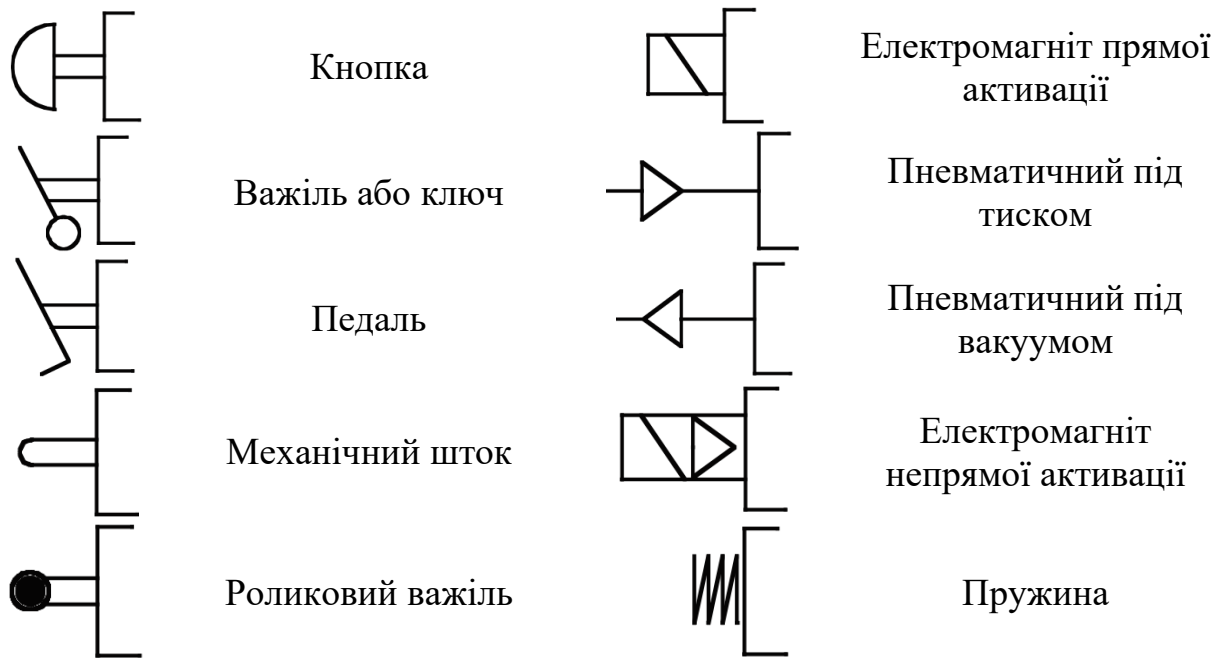


Рис. 4.18. Схема позначення клапана 3/2 N.C.

4.2.3. Соленоїди та електромагнітні клапани.

Соленоїд являє собою котушку з провідного дроту, намотаного на немагнітний трубчастий осердя. Коли через неї проходить електричний струм, утворюється магнітне поле, силові лінії якого концентруються вздовж осі намотування. У місцях входу й виходу силових ліній з котушки утворюються полюси подібно до магніту. Якщо ж у порожнину котушки вставити залізне осердя, магнітний потік посилюється у тисячу разів, адже залізо проводить магнітне поле значно краще за повітря.

Принцип дії електромагнітного клапана подібний до роботи підйомного соленоїда: він має нерухоме осердя, рухоме осердя та котушку. У середині немагнітного корпусу розміщується нерухомий осердя (контр-ядро) і рухоме осердя, яке притискається пружиною. Після подачі струму котушка створює магнітне поле, яке притягує рухому частину до нерухомої, відкриваючи або закриваючи прохід для повітря. У місцях контакту встановлюються ущільнювачі, що забезпечують герметичність і точність роботи. Величина

магнітної сили залежить від розміру повітряного зазору між рухомими частинами, чим він менший, тим більша сила притягання.

Електромагнітні клапани прямої дії забезпечують безпосередній прохід повітря через корпус від входу до виходу. Типовим прикладом є клапан типу 3/2 N.C., який у стані спокою перекриває подачу повітря, одночасно з'єднуючи вихід із атмосферою (рис. 4.19). Після подачі струму рухоме осердя піднімається, перекриваючи отвір розрядження та відкриваючи шлях для повітря до виходу. Такі клапани застосовують у системах із невеликим витратним потоком, оскільки при великих перерізах сила тиску зростає, і магнітному полю важко її подолати без значного збільшення потужності котушки.

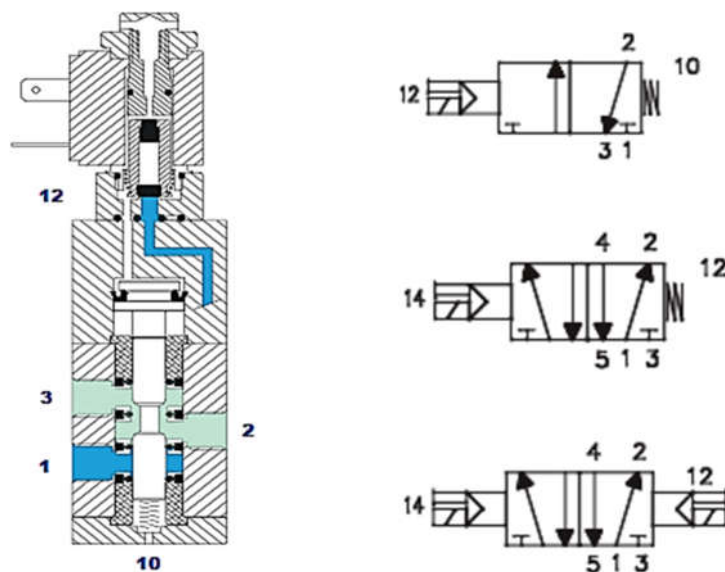


Рис. 4.19. Електромагнітний клапан типу 3/2

Для роботи з великими потоками застосовують непряму дію, коли електромагнітний клапан прямої дії керує більшим пневматичним клапаном – підсилювачем. Комбінація цих двох елементів утворює електромагнітний клапан непрямої дії, який може бути моностабільним або бістабільним, три- або п'ятиходовим, дво- чи трипозиційним.

Розвиток автоматизації привів до підвищення вимог до надійності, компактності та енергоефективності електроклапанів. Вони повинні легко інтегруватися з електронними системами керування, споживати мінімальний струм і відповідати стандартам захисту. Клас захисту пристроїв визначається типом електричного з'єднання котушки та позначається кодом IP. Наприклад, IP40 означає захист від твердих частинок діаметром до 1 мм без захисту від води, а IP65 забезпечує повний захист від пилу та струменів води.

У сучасних системах для зменшення монтажних витрат застосовують модульні блоки електроклапанів (так звані «острови»), де кілька клапанів об'єднані в один корпус з готовими електричними з'єднаннями (рис. 4.20). Це значно скорочує час складання та підвищує надійність. Сьогодні все частіше використовують серійні підключення, коли один дво жильний кабель забезпечує живлення і передачу сигналів для всієї групи клапанів.



Рис. 4.20. Електропневматична система EB 80 від компанії Metal Work

Висновки за розділом.

1. В розділі розглянуто, як утворюється та зберігається стиснене повітря за допомогою компресорів і ресиверів. Пояснено, що перед подачею у систему повітря потрібно очистити, осушити та стабілізувати його тиск, щоб забезпечити надійну роботу обладнання.
2. Описано основні робочі елементи: пневмоциліндри та клапани. Циліндри перетворюють тиск повітря у рух, а клапани керують його подачею та напрямком. Показано, як електричне керування допомагає точніше контролювати роботу пневмосистем.
3. Пояснено, як поєднання електричних і пневматичних елементів дозволяє створювати зручні та ефективні системи автоматичного керування.

Питання для самоперевірки.

1. Що таке електропневматична система автоматизації і які основні її складові?
2. Яке призначення пневматичного циліндра в системах автоматичного керування?
3. Назвіть основні типи пневматичних циліндрів та поясніть принцип їхньої дії.
4. Чим відрізняються односторонньо-діючі та двосторонньо-діючі пневмоциліндри?
5. Які параметри визначають технічні характеристики циліндра (хід, діаметр поршня, тиск тощо)?
6. Яку роль у пневмосистемі відіграють розподільні клапани?
7. За якими ознаками класифікують клапани (за кількістю отворів, положень, принципом дії тощо)?
8. Як здійснюється електричне керування клапанами у електропневматичних схемах?
9. Поясніть призначення зворотних, дросельних та регулювальних клапанів.
10. Як забезпечується узгодження роботи електричних та пневматичних елементів у єдиній системі автоматизації?

РОЗДІЛ 5. ПРОГРАМУВАННЯ ПРИСТРОЇВ ПЛАВНОГО ПУСКУ ТА ПЕРЕТВОРЮВАЧІВ ЧАСТОТИ

5.1. Пристрої плавного пуску.

Пристрій плавного пуску – це пристрій, який застосовується в електроприводах для зменшення пускового струму та крутного моменту двигуна під час запуску. Його принцип дії полягає у поступовому підвищенні напруги, що подається на електродвигун, завдяки чому забезпечується плавне прискорення обертання двигуна та навантаження, з яким він з'єднаний. Такий режим роботи створює лінійне зростання потужності, усуває різкі поштовхи при запуску, зменшує струмові кидки й знижує механічне зношення рухомих частин. Пристрій дозволяє налаштовувати початковий крутний момент і тривалість розгону, що дає змогу адаптувати його до різних технологічних процесів.

Пристрої плавного пуску широко застосовуються у металургійній промисловості (прокатні стани, лінії обробки металу), гумовій та пластмасовій галузях, целюлозно-паперовому виробництві, цукровій промисловості, системах водопостачання, верстатобудуванні, цементній та текстильній індустріях. Найчастіше вони використовуються для керування насосами, вентиляторами, компресорами, конвеєрами, а також дробарками, міксерами, подрібнювачами тощо.

До основних переваг пристроїв плавного пуску належать:

- забезпечують плавний і рівномірний пуск електродвигуна шляхом точного регулювання крутного моменту, що запобігає ривкам під час розгону (особливо в трифазних системах), що істотно подовжує термін служби механічних елементів приводу і зменшує потребу в технічному обслуговуванні;
- значне зниження пускового струму, необхідного для зрушення двигуна з місця, із подальшим контролем струму протягом усього періоду розгону, що зменшується механічне, електричне та теплове навантаження на двигун, трансформатори й комутаційне обладнання;
- використання мікропроцесорного керування, тобто мають програмно контрольований вихід на номінальну швидкість, що дозволяє моніторити енергоспоживання і забезпечує стабільність роботи системи.
- покращення коефіцієнта потужності ($\cos \varphi$), завдяки вбудованій системі самоконтролю: при роботі двигуна з неповним навантаженням реактивна складова струму, що витрачається на намагнічування, стає надмірною, однак плавне регулювання напруги зменшує ці втрати, що сприяє підвищенню енергоефективності системи.

Серед провідних виробників таких пристроїв варто відзначити Schneider Electric. Основна лінійка пристроїв плавного пуску від цієї компанії орієнтована на три сегмента застосувань:

- прості машини та HVAC (системи вентиляції, опалення та кондиціонування);
- стандартні промислові машини (насоси та вентилятори, компресори);
- процеси та інфраструктура (відцентрові машини, компресори, конвеєри (довгі), насоси та вентилятори).

В залежності від типу застосувань та від потужності найпростішим є пристрій плавного пуску ATS01 (рис.5.1а) або його оновлений варіант ATS130. Розглянемо його основні характеристики:

1. Напруга живлення:
 - 110 В...230 В, 1 фаза;
 - 200 В ... 480 В, 3 фази;
2. Струм від 3 А до 32 А.
3. Потужність двигуна від 0.37 кВт до 15 кВт.
4. Поєднання з пускачем TeSys U (рис. 5.1б).
5. З вбудованим байпасом (рис. 5.1в).
6. Просте налаштування:
 - легке введення в експлуатацію: лише 3 потенціометра;
 - простий у використанні та встановленні (монтаж на DIN-рейку чи панель).
7. Найкомпактніший дизайн: ширина 22.5 та 45 мм.



Рис. 5.1. Пристрій плавного пуску ATS01

На відміну від попередньої серії пристрій плавного пуску ATS130 (рис. 5.2) має більший діапазон потужності (струм від 38 до 105 А).

Збільшено функціональні можливості за рахунок додавання двох цифрових входів, які дозволяють реалізувати схему «Пуск/Стоп» та «Пуск-Буст» використовуючи тільки пристрій плавного пуску (без застосування зовнішньої схеми керування). Додаткове вихідне реле стану (NO) дозволяє використати його в додаткових схема керування.



Рис. 5.2. Пристрій плавного пуску ATS130

Пристрій плавного пуску ATS22 (рис. 5.3) має наступні основні технічні характеристики:

- напруга живлення: 230 В ... 600 В, 3 фази;
- струм від 17 А до 590 А;
- потужність двигуна від 4 кВт до 500 кВт;
- ступінь захисту IP00/IP20;
- просте підключення;
- вбудований 7-сегментний екран з кнопками керування;
- вбудований байпас;
- розширені захисти двигуна (контроль за трьома фазами);
- робота з двигунами зі з'єднанням у зірку;
- вбудований Modbus;
- робота у важких умовах (стійке покриття).



Рис. 5.3. Пристрій плавного пуску ATS22

Розглянемо типові стандартні застосування пристрою плавного пуску ATS22 в залежності від типу механізму та додаткових функцій які наявні в ньому:

- відцентровий насос (керування гальмуванням та зупинкою (зниження гідравлічних ударів), захищає від недовантаження і неправильного чергування фаз);
- поршневий насос (керування заливкою насоса та напрямом обертання);
- вентилятор (визначення перевантаження або недовантаження (проблеми з передавальним механізмом між електродвигуном/вентилятором), гальмівний момент під час зупинки);
- турбіна (оцінка теплового стану електродвигуна за допомогою електрично ізольованого датчика РТС);
- холодильний компресор (керування пусковими характеристиками, налаштування автоматичного перезапуску);
- гвинтовий компресор (захист від неправильного чергування фаз, контакт для команди автоматичного розвантаження при зупинці);
- відцентровий компресор (захист від неправильного чергування фаз, контакт для команди автоматичного розвантаження при зупинці);
- конвеєр (контроль перевантаження для виявлення інцидентів та контроль недовантаження для виявлення поломок);
- стрічковий конвеєр (можливість роботи з двома комплектами параметрів двигуна залежно від навантаження на стрічку конвеєра);
- такелажний підйомник (контроль перевантаження для виявлення жорстких ділянок та контроль недовантаження для виявлення поломок);
- змішувач (відображення струму як індикатор в'язкості матеріала);
- міксер (відображення струму як індикатор в'язкості матеріала, форсування при старті);
- очищувач (керування моментом при пуску та зупинці).

Для сигналізації режимів роботи пристрою плавного пуску ATS22 на лицевій панелі розміщено 4х сегментний індикатор та 4 світлодіоди. На 4х сегментному індикаторі за замовченням виводиться частота під час пуску/гальмуванні.

Перший зелений світлодіод **Rdy** має три статуси: включений – живлення плати керування та силове живлення подано; виключений – немає живлення на платі керування; блимає – відсутнє силове живлення.

Другий зелений світлодіод **Com** сигналізує про стан зв'язку по мережі Modbus.

Третій жовтий світлодіод **Run** має три статуси: включений – на двигун подається напруга мережі, байпасний конактор включено; вимкнено – двигун зупинено; блимає – здійснюється розгін/гальмування.

Четвертий червоний світлодіод **Trip** має три стани: включено – аварія (зупинка двигуна); вимкнено – аварія відсутня; блимає – попередження.

Пристрій плавного пуску ATS22 має вхід RJ-45 для підключення Modbus RTU, який дозволяє проводити його налаштування в програмі SoMove, підключати для керування ПЛК та виносний термінал (рис. 5.5).

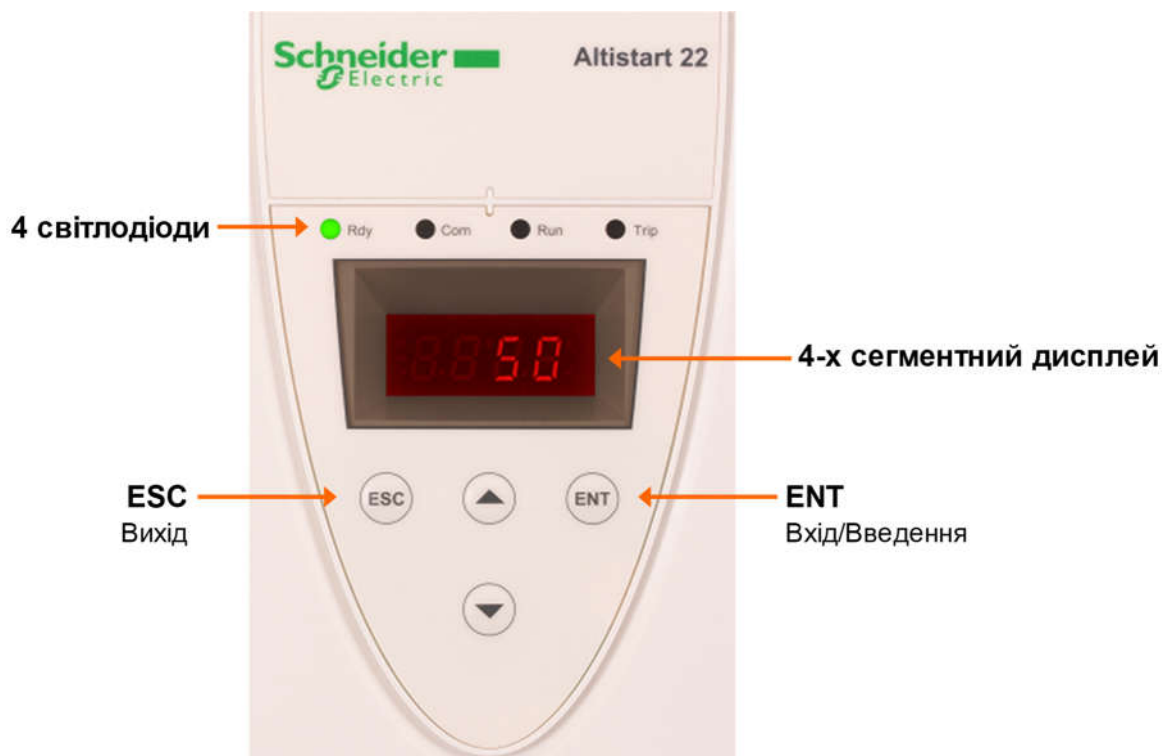


Рис. 5.4. Елементи сигналізації та керування пристрою плавного пуску ATS22

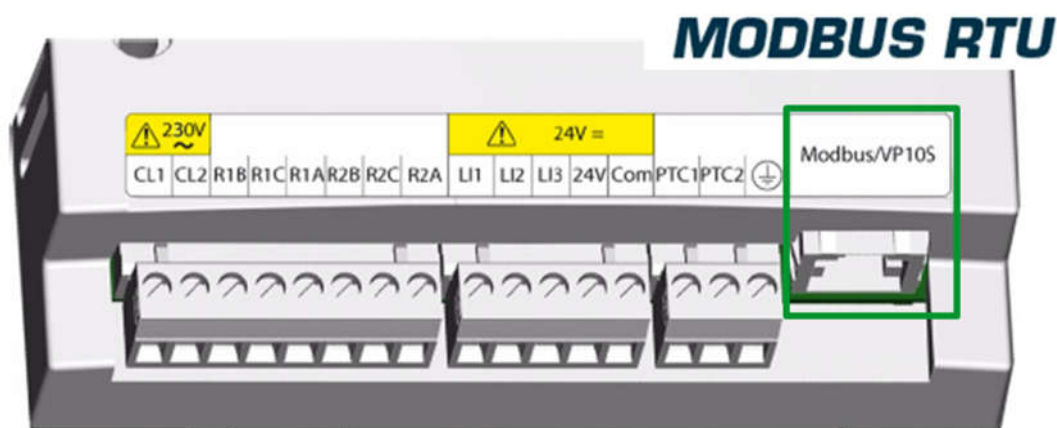


Рис. 5.5. Клемна колодка керування пристрою плавного пуску ATS22

Програмування пристрою плавного пуску ATS22 можна проводити використовуючи кнопки на передній панелі, а також з використанням програми SoMove. Процес налаштування пристрою, за допомогою кнопок керування, складається з трьох основних етапів. Спочатку необхідно за допомогою прокручування перейти до меню параметрів і підтвердити вибір клавішею ENT. Потім у цьому меню знайти потрібний параметр і знову натиснути ENT. На третьому етапі обрати бажане значення, після чого ще раз натиснути ENT, щоб зберегти його. Вибране значення параметра починає діяти одразу після його

зміни, ще до підтвердження клавішею ENT. Наприклад, якщо під час запуску двигуна збільшити обмеження струму, то сила струму зросте миттєво (але не довше ніж на 15 секунд). Після підбору оптимального значення користувач може або зберегти його, натиснувши ENT, або повернутися до попереднього значення, натиснувши ESC, або просто зачекати 15 секунд, після чого пристрій автоматично повернеться до попередніх налаштувань.

Структура меню складається з двох рівнів (рис.5.6).

Перший рівень – «Easy start up» (простий запуск) встановлений на заводі за замовчуванням. Він забезпечує доступ до основних параметрів, що визначають роботу застосування, зокрема налаштування часу розгону (acceleration ramp) та рівня підсилення (boost level) (рис. 5.6а).

Другий рівень – «Advanced level» (розширений рівень) відкриває доступ до додаткових параметрів, які визначають характеристики захисту двигуна, параметри інтерфейсу, комунікації тощо. Після вибору цього рівня з'являються додаткові меню, а в меню моніторингу розширений набір параметрів (рис. 5.6б).

Щоб обрати потрібний рівень меню («Easy start up» або «Advanced»), слід виконати такі дії:

1. за допомогою клавіш прокручування (вперед/назад) перейти до меню conF і натиснути клавішу ENT, щоб увійти в меню конфігурації;
2. далі прокрутити до параметра LAC у меню conF і знову натиснути ENT;
3. вибрати потрібний рівень: oFF – для простого запуску, або On – для розширеного режиму, після чого натиснути ENT.

При встановленні нового значення параметра (натиснути ENT), він набуває чинності одразу після зміни, але якщо натиснути ESC, пристрій відновить попереднє значення, збережене у пам'яті EEPROM.

Канал керування може бути локальним або дистанційним. Канал керування визначає спосіб подачі команд на пристрій плавного пуску (пуск, зупинка тощо) та можливість зчитування чи зміни параметрів. У локальному режимі керування налаштування та команди здійснюються безпосередньо з вбудованої панелі пристрою за допомогою чотирьох клавіш, що дозволяють переходити по меню та змінювати параметри. У дистанційному режимі керування відбувається з виносної панелі (Remote Keypad), яка повністю дублює функціонал вбудованої її інтерфейс та логіка роботи ідентичні. Деякі канали керування, окрім виконання команд, можуть також забезпечувати читання або запис параметрів пристрою, що підвищує гнучкість у налаштуванні системи.

Якщо пристрій плавного пуску буде керуватися з ПЛК його необхідно налаштувати на керування за комунікаційним каналом (Modbus). Для цього в пункті меню COP (Advanced communication menu) в підпункті CtrlL (Command channel) встановити параметр dbS. В пункті меню COP також налаштовуються основні параметри обміну даними Modbus: Add – налаштовується Modbus адрес; tbr – встановлюється швидкість обміну даними (19.2 Кб/с за замовчанням); For – обирається формат Modbus повідомлень (8E1 за замовчанням); ttO – Тайм-аут Modbus (5.0 с за замовчанням).

Пристрій плавного пуску низку стандартних функцій обміну за протоколом Modbus, які забезпечують зчитування, запис і ідентифікацію пристрою:

Read Holding Registers (03 / 16#03) зчитування заданої кількості вихідних регістрів. Максимальна довжина PDU становить 63 слова.

Write Single Register (06 / 16#06) запис одного вихідного регістру.

Write Multiple Registers (16 / 16#10) запис декількох вихідних регістрів одночасно. Максимальна довжина PDU 61 слово.

Read Device Identification (підфункція) зчитування ідентифікаційних даних пристрою, таких як тип, версія прошивки та виробник.

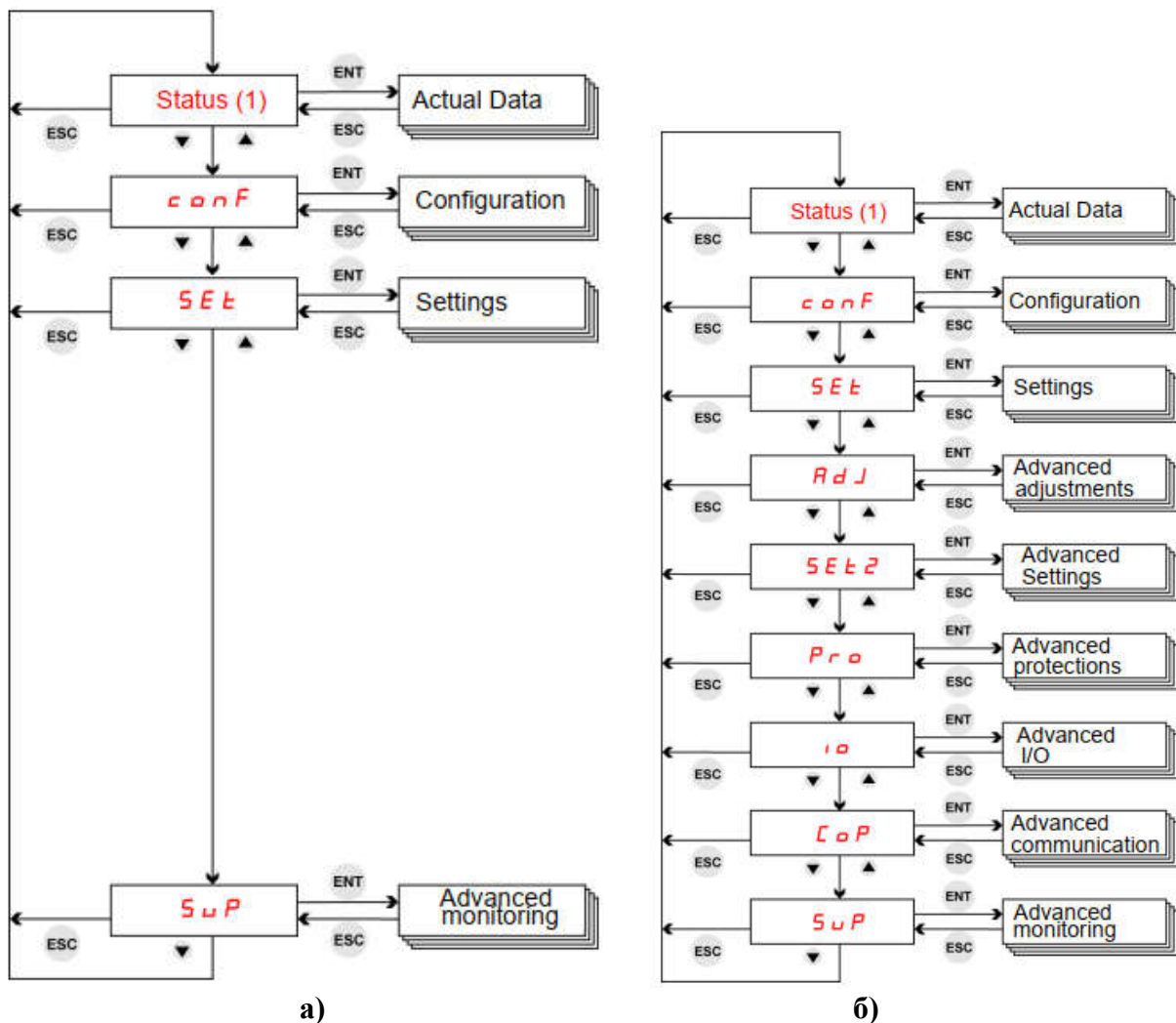


Рис. 5.6. Структура меню програмування пристрою плавного пуску ATS22

Ці функції дозволяють інтегрувати Altistart 22 у системи автоматизації з підтримкою Modbus, забезпечуючи контроль, моніторинг і конфігурацію пристрою через мережу.

5.2. Перетворювачі частоти серії Altivar Machine 320.

5.2.1. Апаратні та функціональні можливості перетворювачів частоти серії ATV320.

Перетворювачі частоти серії Altivar Machine ATV320 компанії Schneider Electric призначені для керування асинхронними та синхронними електродвигунами у промислових машинах та автоматизованих системах. Вони

поєднують компактність, високу надійність та широкий спектр функціональних можливостей, що забезпечує їх ефективне використання у сучасних технологічних процесах.

З апаратної точки зору, перетворювачі цієї серії охоплюють діапазон потужностей від 0.18 до 22 кВт. Виробник передбачив два конструктивних виконання: компактне та у форматі «книга» (рис. 5.7), що дозволяє здійснювати монтаж у шафах керування або безпосередньо на корпусі машини. Пристрої забезпечують стабільну роботу при температурі навколишнього середовища до 50 °С без зниження навантаження, а при відповідних умовах охолодження до 60 °С. Плати електронних модулів покриті захисним шаром, який відповідає вимогам стандарту IEC 60721-3-3 класу 3С3, що гарантує стійкість до впливу пилу, вологи та агресивних середовищ. Залежно від модифікації, ступінь захисту корпусу становить IP21, IP65 або IP66.

У схемі інвертора використано сучасні транзистори IGBT з широтно-імпульсною модуляцією (ШІМ), які забезпечують ефективне перетворення енергії та плавне регулювання швидкості обертання двигуна. Конструкцією передбачено вбудовані захисти від перевантаження, перегріву, перевищення струму та напруги, що підвищує надійність експлуатації та подовжує термін служби обладнання.



Рис. 5.7. Перетворювачі частоти серії Altivar Machine ATV320 компанії Schneider Electric

Програмно-функціональні можливості перетворювачів ATV320 охоплюють кілька режимів керування електродвигунами, зокрема скалярне керування U/f, безсенсорне векторне керування, а також спеціальні режими для вентиляторів і насосів:

- за типом двигуна:
 - синхронний, асинхронний, PM, BLDC;
- за циклом керування двигуном:
 - відкритий цикл (крутний момент і швидкість);
- за методом керування двигуном:
 - безсенсорне векторне керування;
 - стандартне співвідношення швидкість / частота (V/F);
 - 5-точкове V/F;
 - синхронний двигун;
 - керований крутний момент для вентилятора / насоса;
 - режим енергозбереження, герметичні реле (782H) для небезпечних ділянок.

Завдяки цьому пристрої можуть використовуватися у системах із підвищеними вимогами до точності регулювання швидкості або моменту.

Вбудована апаратно-програмна частина реалізує наступні функції:

- 5 простих макросів (запуск / зупинка, обробка, підйомність, загальне використання, ПД-регулювання, шина зв'язку);
- вбудований ПЛК (ATVLogic);
- 16 попередньо заданих швидкостей;
- можливість задавати кілька параметрів через переключення наборів параметрів;
- зовнішня конструкція без вентилятора, 50 °C без зниження потужності;
- 3 аналогових входи, привод зі змінною частотою обертання (VSD) може діяти як монітор сигналів;
- гальмування постійним струмом для швидкої зупинки;
- спільна шина постійного струму з ATVRegen;
- поштовховий режим;
- ПД для вентилятора, насоса;
- позиціонування завдяки кінцевим вимикачам або датчикам;
- 4 набори для перемикачів наборів параметрів;
- автоматичний перезапуск;
- відкритий цикл керування двигуном PM.

Особливу увагу приділено безпеці: у перетворювачах реалізовано вбудовані функції STO (Safe Torque Off), SS1 (Safe Stop 1), SLS (Safe Limited Speed), SMS (Safe Maximum Speed) та GDL (Guard Door Locking). Таке рішення спрощує інтеграцію частотних перетворювачів у машини, що відповідають вимогам європейських директив з безпечності.

Для організації комунікацій передбачено стандартні інтерфейси Modbus RTU та CANopen, а також можливість підключення додаткових модулів для роботи в мережах EtherNet/IP, Modbus TCP, Profinet, Profibus, DeviceNet чи

EtherCAT. Це дає змогу легко інтегрувати перетворювач у системи автоматизації на базі ПЛК різних виробників.

Для налаштування, діагностики та моніторингу роботи застосовується програмне забезпечення SoMove, яке надає зручний інтерфейс для параметрування, збереження конфігурацій, а також спостереження за динамічними характеристиками приводу у реальному часі. Крім того, у пристрої передбачено вбудовану функцію «осцилографа» для відображення осцилограм струму, напруги та температури двигуна.

У конструкції перетворювачів передбачено набір аналогових і дискретних входів та виходів, які забезпечують гнучкість у побудові схем керування (рис. 5.8). Пристрій підтримує реалізацію кількох фіксованих швидкісних режимів, а також PID-регулювання, що дозволяє використовувати його для стабілізації параметрів технологічного процесу наприклад, тиску, температури або витрати.

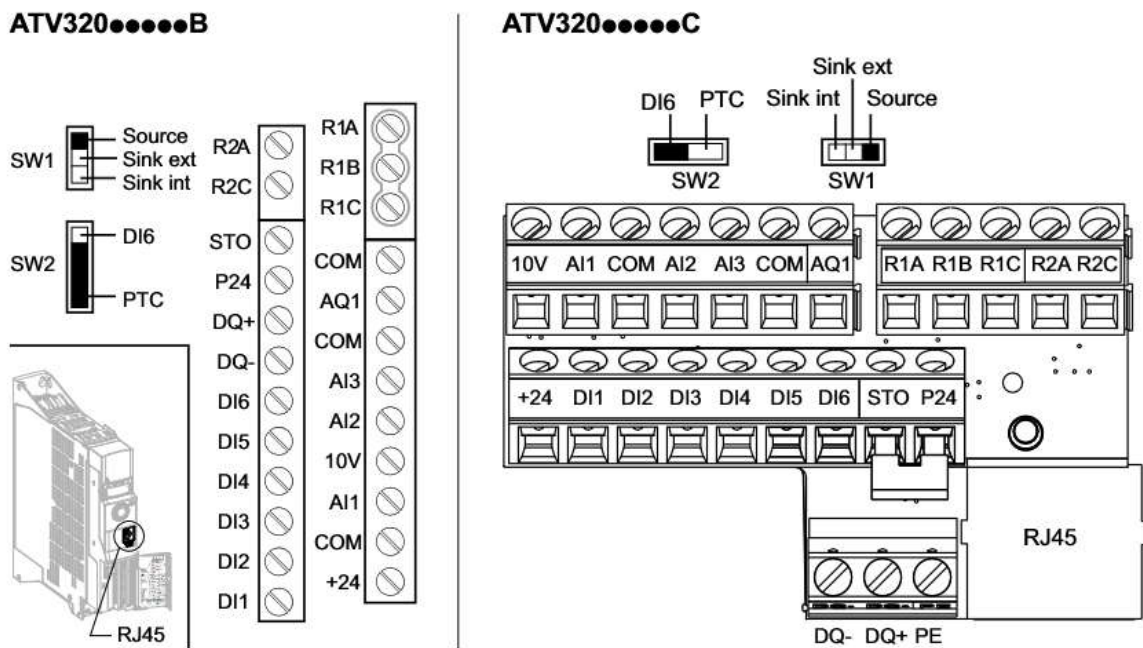


Рис. 5.8. Клемний блок керування перетворювача частоти ATV320 (В – Book, С – Compact)

Інтерфейс перетворювача частоти забезпечує зручний доступ до параметрів, налаштувань і режимів роботи пристрою. Основні елементи панелі керування включають дисплей і три основні органи керування (рис. 5.9).

Клавіша ESC (1) призначена для навігації у зворотному напрямку в меню та для скасування змін параметрів. Натискання цієї клавіші дозволяє вийти з поточного пункту меню або відмінити введене значення без його збереження.

Обертовий енкодер (Jog dial) (2) використовується для переміщення по пунктах меню (вгору або вниз) та зміни значень параметрів (збільшення або зменшення числових величин, вибір елемента). Окрім цього, енкодер може виконувати роль віртуального аналогового входу 1, що використовується для завдання частотної команди перетворювача.

Клавіша ENT (натискання на енкодер) (3) служить для підтвердження вибору або переходу вперед у структурі меню. При роботі з параметрами натискання цієї клавіші фіксує нове значення та активує його застосування у роботі перетворювача.

На панелі відображення частотного перетворювача відображається поточна інформація про режими роботи, параметри, одиниці вимірювання та стан системи (рис. 5.9).

- A – Режим роботи. На дисплеї можуть відображатися три основні режими:
 - rEFB – режим відображення опорного значення (Reference mode);
 - MONC – режим моніторингу параметрів (Monitoring mode);
 - CONF – режим конфігурування параметрів (Configuration mode).
- D – Десяткові точки (1/100 одиниці). Світлова точка в цій позиції показує, що на дисплеї виведено значення параметра з точністю до сотих частин одиниці.
- E – Десяткові точки (1/10 одиниці). Точка у цьому положенні вказує, що значення параметра подано з точністю до десятих частин одиниці.
- F – Індикатори стану та помилок CANopen. Світлодіодні індикатори зліва направо показують:
 - наявність виявленої помилки перетворювачем;
 - стан роботи мережі CANopen (RUN);
 - стан помилки CANopen (Error).
- G – Відображення значення параметра. Основна частина дисплея показує числове значення активного параметра, яке може змінюватися або моніторитися користувачем.
- H – Одиниці вимірювання параметра. Вказує одиницю вимірювання (наприклад, Гц, А, В, %), що відповідає відображеному параметру.

Для керування перетворювачем частоти за допомогою ПЛК необхідно встановити відповідний канал керування: DRI → CONF → FULL → STL змінюємо на MDB – для керування з використанням протоколу обміну даними Modbus або CAN – для використання вбудованого інтерфейсу CANopen. Повні налаштування комунікації за протоколом обміну даними Modbus знаходиться: DRI → CONF → FULL → FLT → MD1 та для вбудованого інтерфейсу CANopen: DRI → CONF → FULL → FLT → CNO.

5.2.2. Програмування ATV Logic в перетворювачі частоти серії ATV320.

Однією з переваг серії ATV320 є наявність функції ATV Logic, що фактично виконує роль вбудованого логічного контролера. Користувач може створювати прості програми керування без потреби у зовнішньому ПЛК, використовуючи графічні блоки з бібліотеки логічних, математичних та часових елементів.

Система ATV Logic базується на концепції пользовацьких функціональних блоків (Function Blocks), які з'єднуються між собою подібно до схеми на мові FBD (Function Block Diagram).

У функціональних блоках/ATV Logic є три типи завдань: PRE, POST і AUX. Правило використання завдання таке (рис. 5.10):

- завдання PRE/POST: підключення тільки до параметрів швидкого доступу;
- завдання AUX: підключення до параметрів повільного та швидкого доступу.

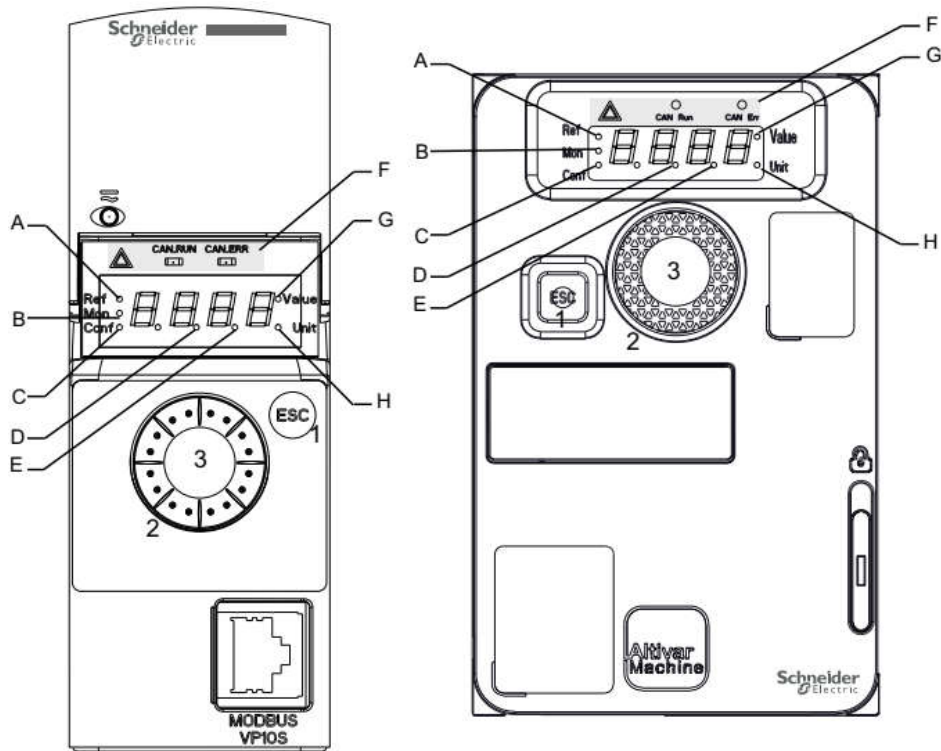


Рис. 5.9. Зовнішній вигляд та елементи індикації перетворювача частоти ATV320

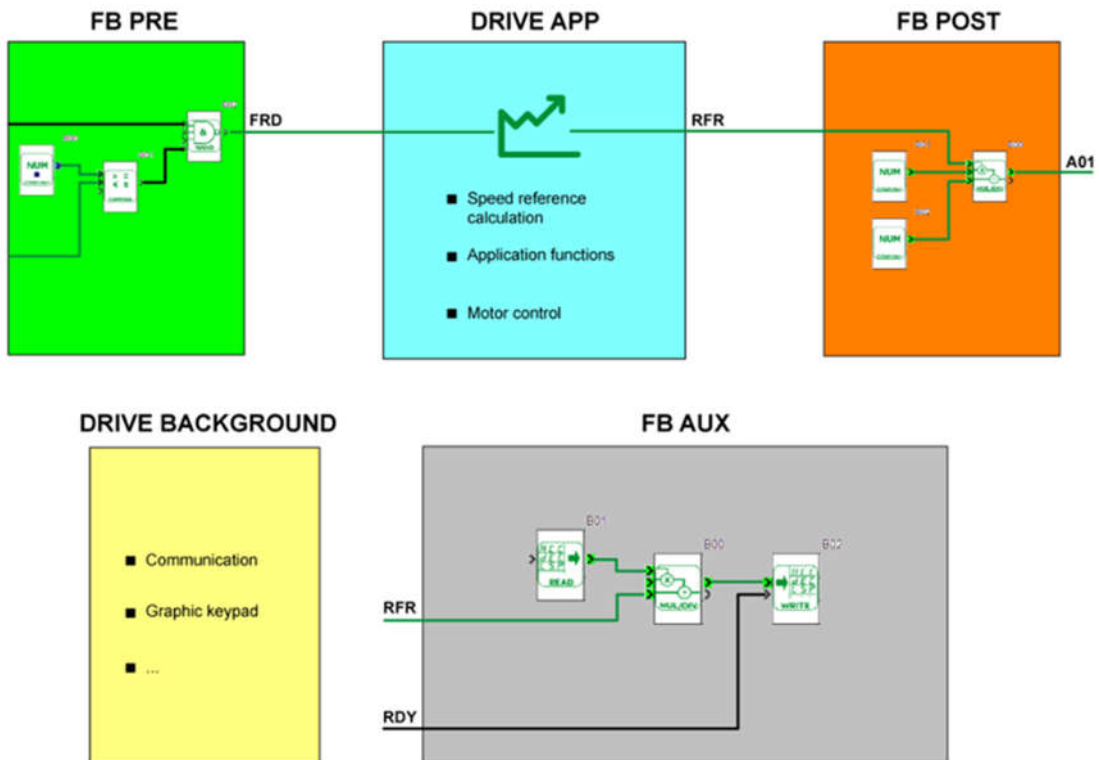


Рис. 5.10. Обробка завдань в ATV Logic

Завдання FB PRE і POST виконуються в прикладному завданні приводу, яке саме виконується кожні 2 мс. Тривалість прикладної задачі приводу становить менше 2 мс. Це залишає від 500 мкс до 1 мс для виконання прикладної задачі приводу. Однак фонове завдання диска може перериватися щоразу, коли має виконуватися прикладне завдання диска. Фонове завдання ПЧ може завантажуватися і, отже, виконуватися кілька разів між двома прикладними завданнями ПЧ.

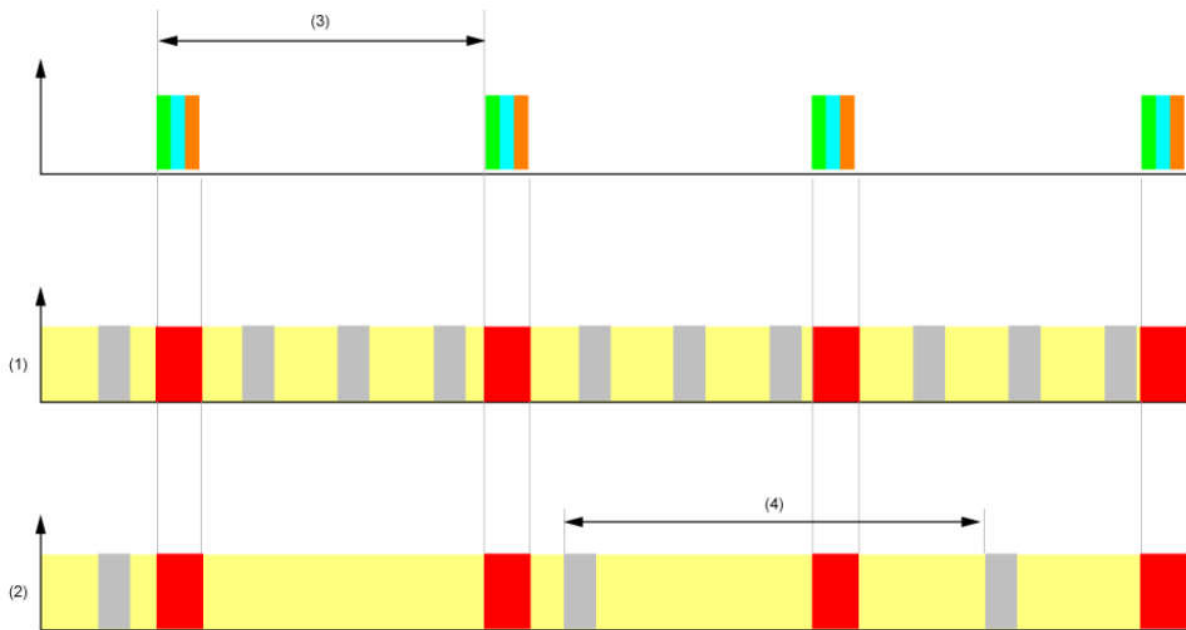


Рис. 5.11. Взаємодія завдань функціональних блоків із завданнями електроприводу: 1 – фонове завдання приводу < 1 мс; 2 – фонове завдання приводу > 1 мс; 3 – завдання 2 мс; 4 – $t >$ тривалості періоду AUX = помилка FBE TOAU

Використання ATV Logic доцільне у таких випадках:

- коли потрібно автоматизувати просту послідовність дій (наприклад, запуск насоса після спрацювання датчика рівня);
- коли система має обмежену кількість сигналів вводу/виводу;
- для оптимізації роботи приводу без додаткового програмного контролера;
- для покращення реакції системи – ATV Logic виконується безпосередньо в перетворювачі, без затримки, пов'язаної з передачею даних до ПЛК.

Програмування ATV Logic здійснюється з використанням панелі оператора (інтерфейс HMI) або програмного забезпечення SoMove (рис. 5.12).

5.2.3. Функціональні блоки ATV Logic в перетворювачі частоти серії ATV320.

ATV Logic містить набір стандартних логічних елементів і спеціалізованих блоків для роботи з приводом. До основних груп відносяться:

1. Логічні блоки:

AND, OR, XOR, NOT – базові логічні операції для обробки сигналів;

RS (Set/Reset) – тригери для запам'ятовування стану.

2. Таймери:

TON (Timer ON Delay) – затримка на включення;

TOF (Timer OFF Delay) – затримка на вимкнення;

TP (Pulse Timer) – формування імпульсу заданої тривалості.

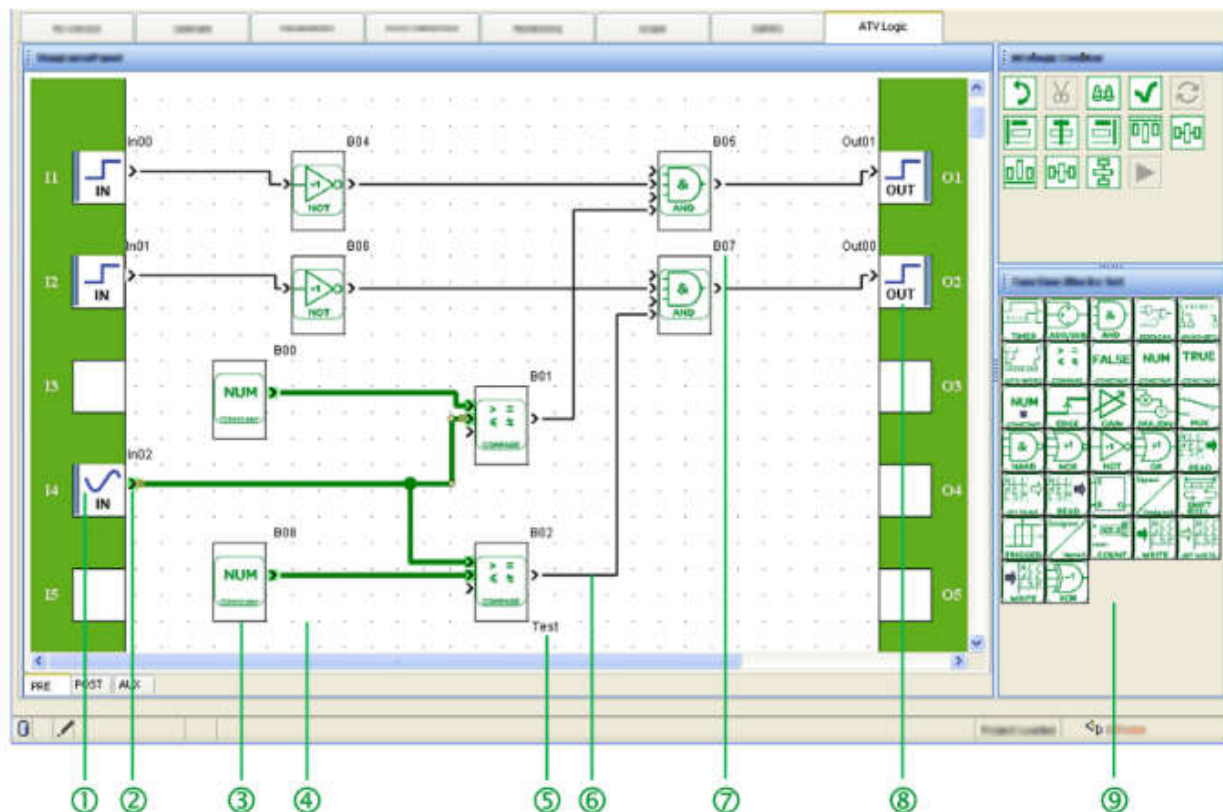


Рис. 5.12. Вікно ATV Logic в програмному забезпеченні Somove:
1 – віртуальний вхідний контакт; 2 – роз'єм; 3 – функціональний блок;
4 – поле для схеми; 5 – коментарі; 6 – з'єднання між блоками;
7 – номер функціонального блоку; 8 – віртуальний вихідний контакт;
9 -- панелі інструментів: зедагування та інструментарій

3. Лічильники:

CTU (Count Up) – лічильник наростання імпульсів;

CTD (Count Down) – лічильник зменшення.

4. Аналогові блоки:

Comparator – порівняння аналогових сигналів (наприклад, струму або частоти);

Scaler – масштабування вхідних сигналів.

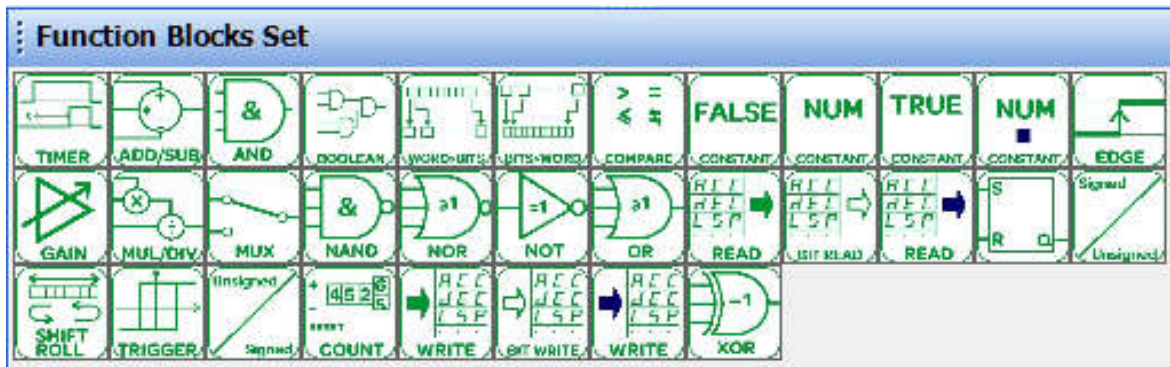
5. Спеціальні блоки

Motor Status – аналіз стану двигуна (робота, помилка, перевантаження);

Command Block – формування команд пуску, зупинки, реверсу;

Analog Output Control – керування аналоговими виходами для сигналізації або регулювання.

Розглянемо принцип роботи функціональних блоків ATV Logic в перетворювачі частоти ATV320.



Функція *Таймер* використовується для затримки, подовження та керування діями протягом заданого часу.

Таймер має три функції:

- Функція А: Затримка ввімкнення таймера або активний таймер;
- Функція С: Затримка вимкнення таймера, або таймер бездіяльності;
- Функція А/С: Комбінація функцій А і С.

Функція має Входи/виходи:

- [IN]: 1 булевий вхід;
- [Q]: 1 булевий вихід.

Арифметична функція *ADD-SUB* використовується для виконання простих операцій над цілими числами:

- Додавання;
- Віднімання.

Функція має Входи/виходи:

Опис вхідних даних:

- [A]: Аналоговий вхід;
- [B]: Аналоговий вхід;
- [C]: Аналоговий вхід.

Опис виходів:

- [Q]: Аналоговий вихід: це значення виведеної розрахункової формули;
- [Переповнення]: Цей вихід, який має булевий тип, вказує на наявність помилок переповнення (результат не входить в інтервал [-32768, +32767]).

Формула обчислення: РЕЗУЛЬТАТ ОБЧИСЛЕННЯ $[Q] = [A] + [B] - [C]$.

Функція *AND*. Якщо всі входи активні або не підключені, вихід активний. Якщо хоча б один вхід підключено, вихід неактивний.

Функція має Входи/виходи:

- [IN1] - [IN4]: 4 логічні входи;
- [Q]: 1 булевий вихід.

In1	In2	In3	In4	Q
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

Функція *BOOLEAN* повертає значення виходу відповідно до комбінації входів. Функція має чотири входи, а отже, 16 комбінацій. Ці комбінації можна знайти у таблиці істинності; для кожної з них можна налаштувати значення виходу. Кількість комбінацій, які можна налаштувати залежить від кількості входів, підключених до функції. Непідключені входи встановлюються на 0.

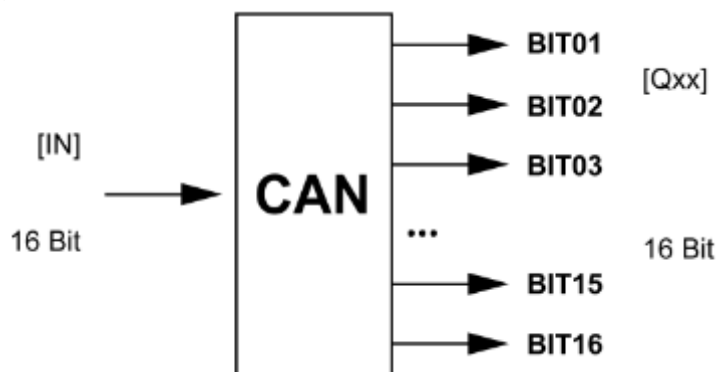
Функція має Входи/виходи:

[IN1], [IN2], [IN3] та [IN4]: 4 булеві входи;
[OUT]: 1 булевий вихід.

Функція *CAN* (перетворення слова в біт). Функція Word to Bit Conversion розбиває вхідні дані регістрового типу (16-біт) на 16-розрядні виходи.

Функція має Входи/виходи:

[IN]: 1 вхід регістра;
Q01 - Q16]: 16 булевих виходів: Біт01 (молодший байт) ... Біт16 (старший байт).



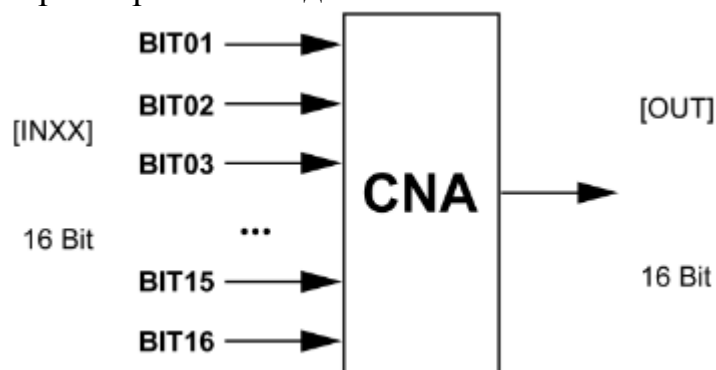
Функція *CNA* (перетворення біт в слово).

Функція перетворення біт у слово створює регістровий (16-розрядний) вихід з 16-розрядних входів типу входів.

Функція має Входи/виходи:

[IN01] - [IN10]: 16 логічних входів: Біт01 (молодший байт) ... Біт16 (старший байт);

[OUT]: 1 регістровий вихід.



COMPARE (Порівняння двох значень) ця функція використовується для порівняння двох аналогових значень. Вихід [Q] активний, якщо результат порівняння між [IN1] і [IN2] істинний, і якщо вхід функції Enable функції активний або не підключений. Вихід не змінює стан, якщо вхід Enable functions змінюється з активного на неактивний.

Функція має Входи / Виходи:

[IN1]: 1 аналоговий вхід;

[IN2]: 1 аналоговий вхід;

[ENABLE]: 1 логічний вхід. Якщо вхід [IN1] або [IN2] не підключений, значення встановлюється на 0. Примітка: Вихід функціонального блоку не вихід функціонального блоку не оновлюється, тому що якщо вихід функціонального блоку встановлений на 1, а вхід Enable перемикається на 0, то вихід залишиться рівним 1;

[Q]: 1 Булевий вихід.

TRUE – ця константа може бути використана для встановлення входів функції на 1.

FALSE – ця константа може бути використана для встановлення входів функції на 0.

NUM (аналог) – числова константа NUM є аналоговим цілим числом зі значенням від -32768 до +32767. Значення константи можна встановити у вікні Параметри.

NUM (регістр) – числова константа NUM є регістровим цілим числом зі значенням від 0 до 65535. Значення константи можна встановити у вікні Параметри.

Функція *EDGE* – ця функція виявляє спадний або наростаючий фронт сигналу.

Функція має Входи/виходи:

– Тестування сигналу.

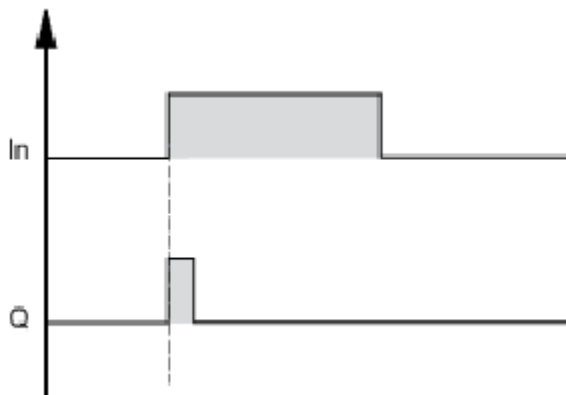
[Q] 1 Булевий вихід.

– Імпульс при виявленні фронту.

[IN]: 1 булевий вхід.

Конфігурація параметрів:

- Падаючий;
- Висхідний;
- Двічі.



Функція *GAIN* дозволяє перетворювати аналогові значення шляхом зміни масштабу і зсуву. Формула розрахунку коефіцієнта посилення: $[Q] = (A / B) \times \text{INPUT} + [C]$.

Функція має Входи:

[IN]: 1 аналоговий вхід. Це ціле число між -32768 і 32767;

[ENABLE]: 1 логічний вхід. Це команда входу функції підсилення.

Стан цього входу визначає роботу блоку: якщо вхід Enable Function неактивний, на виході Calculation зберігається останнє розраховане значення.

Функція має Виходи:

[Q]: 1 аналоговий вхід. Це вихідне значення функції підсилення. Це значення залежить від стану входу Enable функції.

[OVERFLOW]: 1 булевий вихід. Якщо він є:

Активний: вихід Calculation дорівнює результату формули розрахунку коефіцієнта підсилення.

Неактивний: вихід Calculation дорівнює нулю.

Арифметична функція *MUL/DIV* використовується для виконання операцій над цілими числами за формулою обчислення $\text{MUL/DIV}: [Q] = [A] \times [B] / [C]$.

Функція має Входи:

[A]: 1 аналоговий вхід;

[B]: 1 аналоговий вхід;

[C]: 1 аналоговий вхід.

Функція має Виходи:

[Q]: 1 аналоговий вхід. Це значення виведеної розрахункової формули. [OVERFLOW]: Цей вихід, який має булевий тип,

вказує на наявність помилок насичення.

Функція *MUX* (Multiplexing) здійснює мультиплексування двох вхідних каналів на виході.

Функція має Входи:

[SEL]: 1 булевий вхід. Цей вхід використовується для вибору вхідного каналу, який буде застосовано до виходу.

[IN1]: 1 аналоговий вхід. Це вхід мультиплексора [IN1].

[IN2]: 1 аналоговий вхід. Це вхід мультиплексора [IN2]

Функція має Виходи:

[Q]: 1 аналоговий вхід. Це вихід мультиплексора [Q].

Функція *NAND* – якщо хоча б один вхід неактивний, вихід активний. Якщо всі входи активні або не підключені, вихід неактивний.

Функція має Входи/Виходи:

[IN1] - [IN4]: 4 булеві входи;

[Q]: 1 булевий вихід.

Функція *NOR* – якщо всі входи неактивні або не підключені, вихід активний. Якщо хоча б один вхід активний, вихід неактивний.

Функція має Входи/Виходи:

[IN1] - [IN4]: 4 логічних входи;

[Q]: 1 булевий вихід.

Функція *NOT* – якщо вхід неактивний або не підключений, вихід активний. Якщо вхід активний, вихід неактивний.

Функція має Входи/Виходи:

[IN]: 1 логічний вхід;

[Q]: 1 булевий вихід.

Функція *OR* – якщо хоча б один вхід активний, вихід активний. Якщо всі входи неактивні або не підключені, вихід неактивний.

Функція має Входи/Виходи:

[IN1] - [IN4]: 4 логічні входи;

[Q]: 1 булевий вихід.

Функціональний блок *READ Ana Param* використовується для зчитування значення.

Функціональний блок має Входи/Виходи:

1 булевий вхід;

1 аналоговий вихід.

Для...	Використовувати	Коментар
Зчитування параметрів ПЧ	з ADL-контейнерів для зв'язування параметрів ПЧ	[ADL CONTAINERS] (FAd-)
Зчитування параметрів протоколу зв'язку	з параметрами M00x для зберігання даних. Зберігати значення для повторного використання функціональними блоками та протоколом зв'язку.	[FB PARAMETERS] (FbP-)

Функціональний блок *BIT READ* використовується для зчитування одного біта параметрів.

Функціональний блок має Входи/Виходи:

[ENABLE]: 1 булевий вхід;

[Q]: 1 булевий вихід Readbit.

Для...	Використовувати	Коментар
Зчитування одного біта параметрів ПЧ	з ADL-контейнерів для зв'язування параметрів ПЧ	[ADL CONTAINERS] (FAd-)
Зчитування параметрів протоколу зв'язку	з параметрами M00x для зберігання даних. Зберігати значення для повторного використання функціональними блоками та протоколом зв'язку.	[FB PARAMETERS] (FbP-)

Функціональний блок *READ Reg Param* використовується для зчитування значення.

Функціональний блок має Входи/Виходи:

1 булевий вхід;

1 регістровий вихід.

Функція SET RESET працює наступним чином:

Активація входу SET активує вихід, який залишається в цьому стані, навіть якщо вхід SET потім деактивується:

- Активація входу RESET деактивує вихід;
- Непідключені входи встановлюються в стан Inactive.

Функція має Входи/Виходи:

[SET]: 1 логічний вхід;

[RESET]: 1 логічний вхід;

[Q]: 1 булевий вихід.

Функціональний блок STU (16-bit analog TO 16-bit register) використовується для перетворення аналогового сигналу в регістровий сигнал шляхом обмеження. Діапазон аналогового входу: від -32768 до +32767. Діапазон вихідного сигналу регістра: від 0 до 32767.

Функціональний блок має Входи/Виходи:

[IN]: 1 аналоговий вхід

[Q]: 1 регістровий вихід

[OVERFLOW]: 1 вихід переповнення

[OVERFLOW] – це булевий вихід. Якщо значення аналогового входу [IN] від'ємне, [OVERFLOW] = 1. Якщо значення аналогового входу [IN] позитивне, [OVERFLOW] = 0.

Функція *SHIFT/ROLL*. *ROLL* переміщує біти у вибраному напрямку і замінює нові порожні біти бітами, вилученими з регістра. Це круговий регістр. Вихід *CARRY* містить рівень (0/1) останнього переміщеного біта. Функція використовується для зсуву або повороту значення [IN] на фіксовану кількість бітів вліво/вправо. При кожному циклі, якщо встановлений сигнал *ENABLE*, логічний блок буде виконувати зсув/обертання вліво/вправо на Номер біта запиту щодо конфігурації параметра.

SHIFT переміщує біти у вибраному напрямку і замінює порожні біти на 0.

ROLL переміщує біти у вибраному напрямку і замінює нові порожні біти на біти, що викидаються з регістра. Це круговий регістр. Вихід *CARRY* містить рівень (0/1) останнього переміщеного біта.

Функція має Входи:

[IN]: 1 вхід регістра;

[ENABLE]: 1 вхід булевої функції.

Функція має Виходи:

[Q]: 1 вихід регістра;

[CARRY]: 1 булевий вихід. Вихід переносу показує значення останнього зміщеного біта.

Функція *TRIGGER* (Schmitt Trigger) – дозволяє контролювати аналогову величину відносно двох порогових значень. Вихід змінює стан, якщо:

- Значення [IN] менше мінімального значення;
- Значення [IN] більше максимального значення;

Якщо вхід [IN] знаходиться між цими двома значеннями, вихід [Q] не змінює стан. Дві уставки - «Увімкнено - Вимкнено» і «Вимкнено-Увімкнено» – можуть бути встановлені як мінімальне або максимальне значення. Це передбачає зворотну роботу функції. Ці дві операції показані на схемах. Якщо вхід [ENABLE] неактивний, вихід [Q] залишається неактивним. Вихід [Q] не змінює стан, якщо вхід [ENABLE] змінюється з активного на неактивний.

Функція має Входи/Виходи:

[IN]: 1 аналоговий вхід уставки;

[ENABLE]: 1 логічний вхід;

[Q]: 1 булевий вихід.

Функціональний блок *UTS* (16-Bit Register Input to 16-Bit Analog Output) використовується для перетворення 16-розрядного входу регістра в 16-розрядний аналоговий вихід (з обмеженням). Діапазон входу регістра: від 0 до 65535. Діапазон виходу регістра: від -32768 до +32767.

Функціональний блок має Входи/Виходи:

[IN]: 1 вхід регістра;

[Q]: 1 аналоговий вихід;

[OVERFLOW]: 1 вихід переповнення.

Вихід [OVERFLOW] має булевий тип. Цей вихід активний, якщо значення на вході регістра перевищує 32767; значення аналогового виходу буде встановлено на 32767.



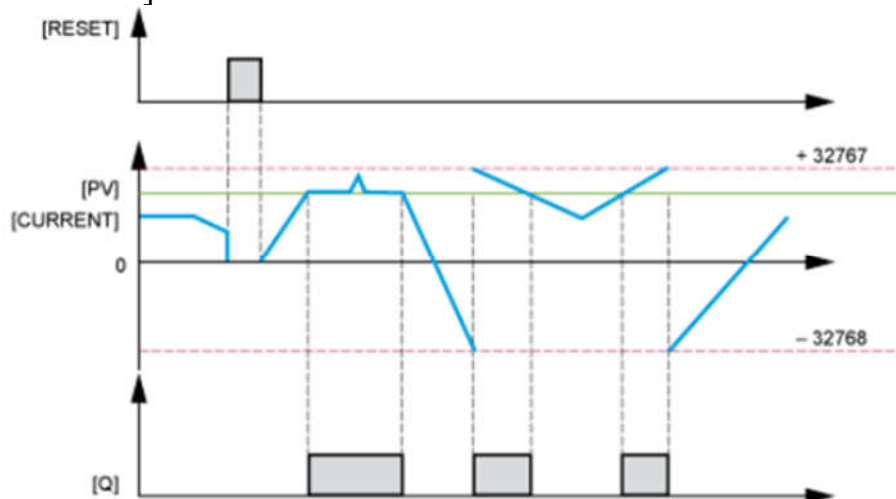
Функція *COUNT* (Up/Down Counter) (лічильник вгору/вниз) використовується для відліку вгору/вниз від заданого значення, отриманого в результаті обчислення за межами функції.

Функція має Входи:

- [UP]: 1 булевий вхід лічильника вгору;
- [DOWN]: 1 булевий вхід зворотного відліку;
- [RESET]: 1 логічний вхід;
- [LOAD]: 1 логічний вхід;
- [PV]: 1 аналоговий вхід попередньо встановленого значення;
- [ENABLE]: 1 вхід.

Функція має Виходи:

- [Q]: 1 Булевий вихід;
- [CURRENT]: Поточне значення лічильника.



Функціональний блок *WRITE Ana Param* використовується для запису одного біта параметрів. Для запису параметрів ПЧ: необхідно використовувати ADL-контейнери для зв'язування параметрів електропривода ([ADL CONTAINERS] (FAd-)). Щоб зберегти або повторно використати значення: Параметри M00x легко використовувати для зберігання даних. Можна зберігати значення для повторного використання за допомогою функціональних блоків і протоколу зв'язку ([FB PARAMETERS] (FbP-)).

Функціональний блок має Входи/Виходи:

- 1 аналоговий вхід;
- 1 логічний вхід.

Функціональний блок *BIT WRITE* (WriteBitParam) використовується для запису одного біта параметрів. Для запису одного біта параметрів ПЧ: необхідно використовувати ADL-контейнери для зв'язування параметрів ПЧ ([ADL

КОНТЕЙНЕРІВ] (FAd-)). Щоб зберегти або повторно використати значення: Параметри M00x легко використовувати для зберігання даних. Можна зберігати значення для повторного використання за допомогою функціональних блоків і протоколу зв'язку ([FB PARAMETERS] (FbP-)).

Функціональний блок має Входи/Виходи:

[IN]: 1 булевий вхід WriteBit;

[ENABLE]: 1 булевий вхід.

Функціональний блок *WRITE* Reg Param використовується для запису одного біта параметрів. Для запису параметрів ПЧ: необхідно використовувати ADL-контейнери для зв'язування параметрів електропривода ([ADL CONTAINERS](FAd-)). Щоб зберегти або повторно використати значення: Параметри M00x легко використовувати для зберігання даних. Можна зберігати значення для повторного використання за допомогою функціональних блоків і протоколу зв'язку ([FB PARAMETERS](FbP-)).

Функціональний блок має Входи/Виходи:

1 регістровий вхід;

1 логічний вхід.

Функція XOR – якщо один вхід неактивний, а інший активний або не підключений, вихід активний. Якщо обидва входи активні або неактивні, або не підключені, вихід неактивний.

Функція має Входи/Виходи:

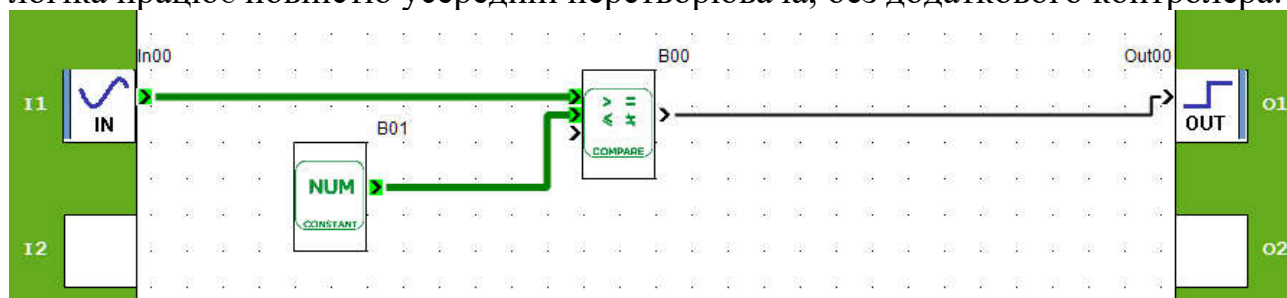
[IN1] та [IN2]: 2 булеві входи;

[Q]: 1 булевий вихід.

Основні переваги ATV Logic: зменшення вартості системи (немає потреби у ПЛК); простота створення та модифікації алгоритмів; висока швидкість реагування; можливість тестування логіки без підключення двигуна; збереження програми в пам'яті пристрою (не втрачається при вимкненні живлення).

5.2.4. Приклад застосування ATV Logic в перетворювачі частоти серії ATV320.

Розглянемо приклад застосування ATV Logic для системи автоматичного керування температурою повітря. При перевищенні температури повітря вище 40 °C автоматично запускається вентилятор. Для цього в ATVLogic з'єднуємо вхідний аналоговий сигнал від датчика температури на блок Comparator. Який при перевищенні порогу 40 °C активує вихід блоку до якого приєднано блок Command Forward, який запускає двигун вентилятора. При зниженні температури нижче порогу вентилятор автоматично зупиняється. Таким чином, логіка працює повністю усередині перетворювача, без додаткового контролера.



5.3. Перетворювачі частоти Altivar Process 630.

5.3.1. Апаратні та функціональні можливості перетворювачів частоти серії ATV630.

Перетворювачі частоти серії Altivar Process ATV630 компанії Schneider Electric призначені для керування асинхронними та синхронними електродвигунами у промислових установках, де потрібна висока енергоефективність, точність регулювання та надійність роботи в складних умовах експлуатації. Ці перетворювачі належать до серії Altivar Process, що орієнтована на процесне керування у промислових і комунальних системах: насосних станціях, вентиляційних агрегатах, компресорних установках, міксерних, конвеєрних і технологічних лініях, тобто для обладнання, що працює у безперервному циклі, потребує моніторингу енергоефективності та захисту технологічного середовища.

З апаратної точки зору перетворювачі ATV630 охоплюють широкий діапазон потужностей від 0.75 до 315 кВт (при напрузі живлення 380...480 В) та забезпечують стабільну роботу у важких промислових умовах. Конструктивно передбачено кілька виконань корпусу: компактне (IP21) для монтажу у шафах керування та високозахищене (IP54/IP55) для безпосереднього встановлення у виробничих приміщеннях (рис. 5.13). Робоча температура без зниження навантаження сягає 50 °С, а за умови ефективного охолодження до 60 °С. Плати електронних компонентів покриті спеціальним захисним лаком, що відповідає вимогам ІЕС 60721-3-3 класу 3С3, забезпечуючи стійкість до пилу, вологи та агресивних середовищ.

Інверторна частина побудована на базі IGBT-транзисторів із широтно-імпульсною модуляцією (ШІМ), що забезпечує високу ефективність перетворення енергії, плавність пуску та зупинки двигуна, а також зменшення рівня електромагнітних перешкод. Перетворювачі оснащені вбудованими фільтрами гармонік (RFI), що дозволяє їх використовувати у промислових мережах без додаткових фільтруючих пристроїв.

Перетворювачі ATV630 підтримують кілька режимів керування двигунами:

- U/F VC Standard motor law – скалярне частотне керування з підтримкою співвідношення U/f.
- U/F VC 5 points voltage/frequency – скалярне частотне керування (U/f) 5-точкового профілю;
- U/F VC Quadratic – скалярне частотне керування (U/f), момент навантаження якого квадратично залежить від швидкості обертання;
- SYN_U VC – синхронними з постійними магнітами зі змінним крутним моментом (розімкнутий контур);
- Reluctance Motor – синхронними реактивними;
- U/F VC Energy Saving – енергозберігаючий режим для вентиляторів і насосів із функцією оптимізації енергоспоживання (Energy Saving Mode).



Рис. 5.13. Перетворювачі частоти серії Altivar Process ATV630 компанії Schneider Electric: а) IP21; б) IP55

Усі ці режими дозволяють використовувати ATV630 як у простих системах водопостачання чи вентиляції, так і у складних технологічних комплексах, де необхідно забезпечити стабільність тиску, витрати або швидкості потоку.

Завдяки розширеній функціональності перетворювачі ATV630 виконують роль не лише частотного регулятора, а й інтелектуального контролера технологічного процесу. Апаратно-програмна частина реалізує:

- вибір режиму застосування;
- одиниці застосування;
- функції керування насосами (цільове призначення): моніторинг насосів, функція сну / пробудження, обмеження потоку, оцінка витрати без датчиків, керування жокей насосом, функція захисту від заклинювання, функція заповнення труби, компенсація втрат на гідравлічне тертя, керування zalивним насосом, захист від відсутності або низького потоку, захист насоса від кавітації, захист насоса від сухого ходу, захист насоса низького тиску, захист від перепадів тиску на вході/виході, захист від циклічного запуску насоса, захист від підвищеного потоку;
- ПІД-регулятор з моніторингом зворотного зв'язку;
- моніторинг спожитої енергії;
- пропуск частоти;
- блокування несправностей.

Перетворювач має розвинену систему захисту:

- контроль струму, напруги та температури;
- захист від короткого замикання, обриву фази, перевантаження;
- виявлення дисбалансу фаз та блокування двигуна;
- контроль заземлення і температури навколишнього середовища;
- попереджувальні повідомлення про можливі несправності.

У системі безпеки реалізовано вбудовані функції SIL2/SIL3 рівня, такі як:

- STO (Safe Torque Off) – безпечно зняття крутного моменту;
- SS1 (Safe Stop 1) – безпечна зупинка з контрольованим гальмуванням;
- SLS (Safe Limited Speed) – обмеження швидкості у безпечному діапазоні;
- SMS (Safe Maximum Speed) – контроль максимально допустимої швидкості;
- GDL (Guard Door Locking) – блокування при відкритті захисних кожухів.

Це дозволяє інтегрувати ATV630 у системи, що відповідають вимогам європейських директив із машинної безпеки (Machinery Directive 2006/42/EC).

У комунікаційному плані ATV630 має широкі можливості підключення до промислових мереж. Базово підтримуються Modbus RTU і EtherNet/IP, а також опціонально Modbus TCP, CANopen, Profibus DP, Profinet, DeviceNet, EtherCAT, ВАСnet, що забезпечує сумісність із більшістю промислових контролерів і систем SCADA. Передбачено можливість віддаленого моніторингу та діагностики у режимі реального часу.

Для налаштування, тестування та обслуговування використовується програмне забезпечення SoMove або веб-інтерфейс Web Server (рис. 5.14-15), який дозволяє підключатися до перетворювача через Ethernet без додаткового програмного забезпечення. Вбудовані засоби діагностики включають осцилограф, який відображає параметри струму, напруги, частоти та температури в графічному вигляді, що спрощує аналіз роботи приводу.

Конструкцією передбачено широкий набір аналогових та дискретних входів/виходів, що дає змогу гнучко реалізовувати різноманітні схеми керування: з локальними командами, зовнішніми датчиками або віддаленим керуванням (рис. 5.16). ATV630 підтримує збереження кількох наборів параметрів (setpoints), між якими можна швидко перемикатися під час роботи, що зручно для багаторежимних процесів.

Передня панель приводу ATV630 оснащена низкою світлодіодних індикаторів (LED), які сигналізують про стан пристрою, наявність помилок, стан підключення до мережі Ethernet та активність зв'язку Modbus (рис. 5.17).

Світлодіод STATUS (1) сигналізує про загальний стан приводу. Якщо індикатор вимкнений, це означає, що пристрій знеструмлений і не отримує живлення. Коли зелений сигнал блимає з певною періодичністю, привід перебуває у стані готовності до запуску, але наразі не працює. Якщо зелений колір блимає частіше, це вказує на перехідний стан наприклад, прискорення або гальмування. Постійне зелене світло означає, що привід запущений і працює в нормальному режимі.

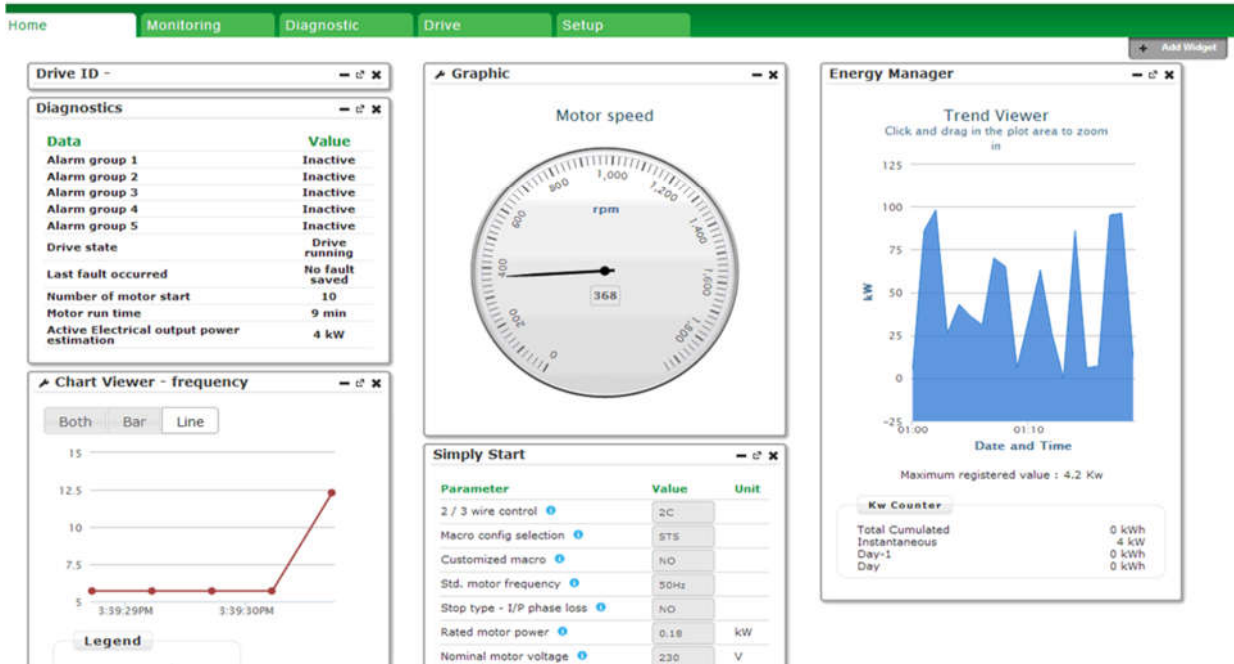


Рис. 5.14. Моніторинг роботи перетворювача частоти серії Altivar Process ATV630 в Web Server

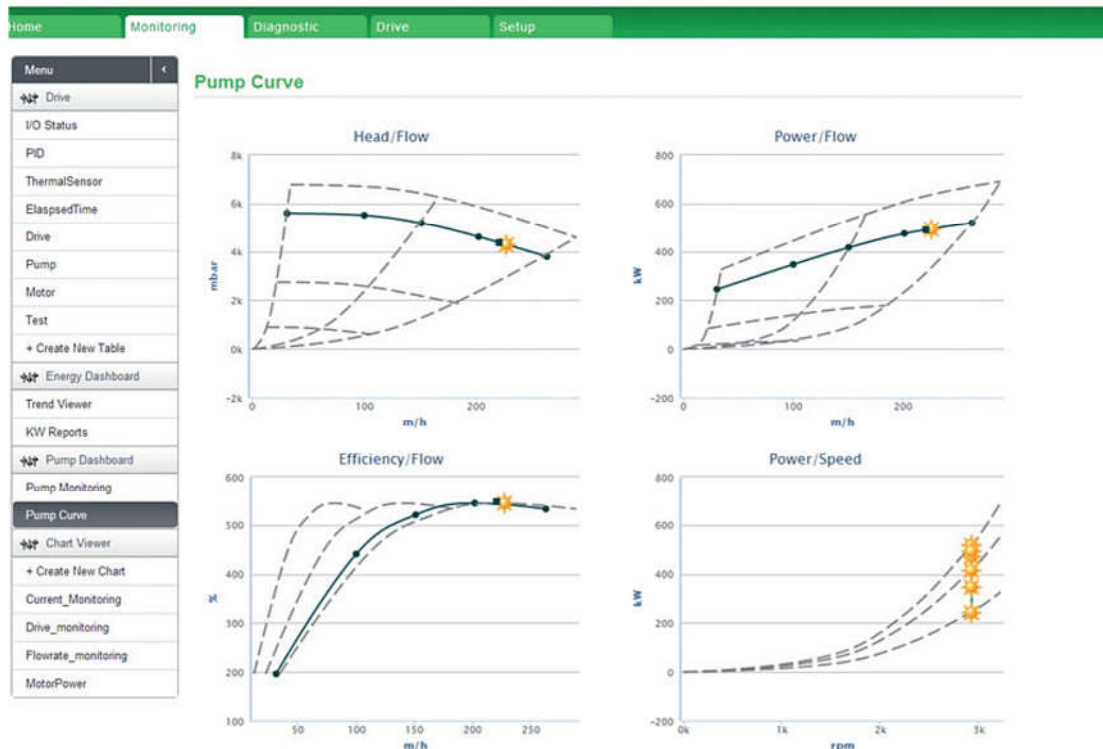


Рис. 5.15. Введення даних кривих насоса в Web Server перетворювача частоти серії Altivar Process ATV630

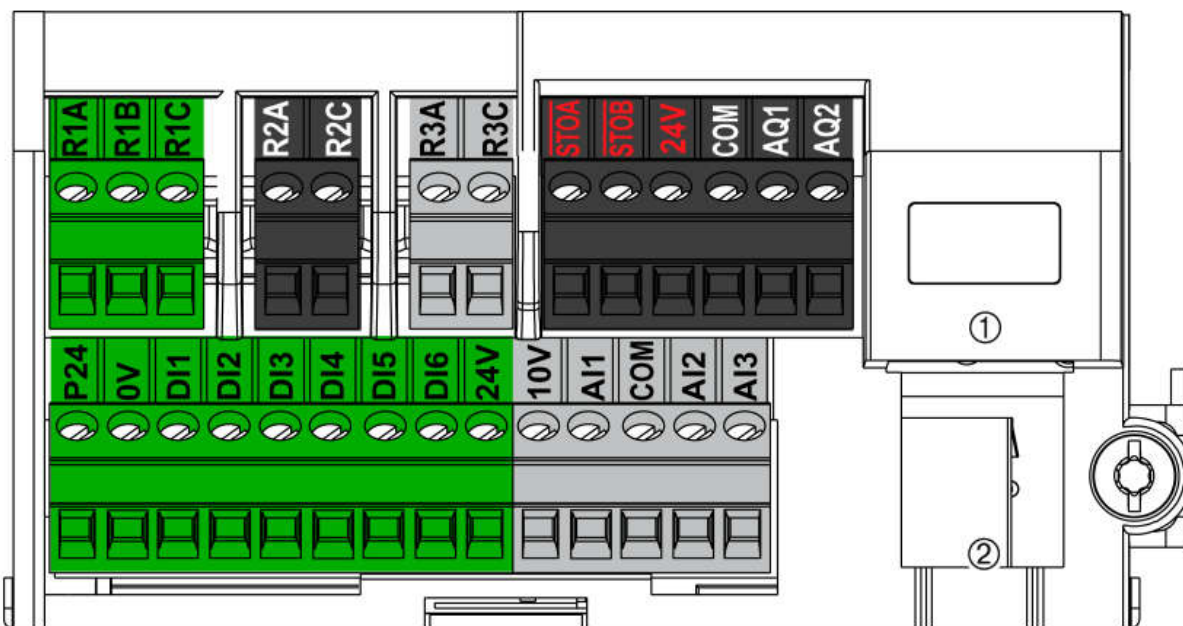


Рис. 5.16. Клеми блоку управління перетворювача частоти серії Altivar Process ATV630: 1 – Ethernet Modbus TCP, 2 – Serial Modbus

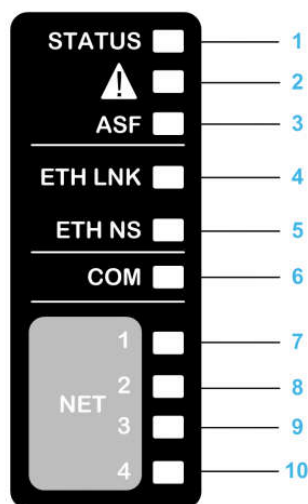


Рис. 5.17. Світлодіодні індикатори на передній панелі перетворювача частоти серії Altivar Process ATV630

Індикатор Warning/Error (2) попереджає про збої або критичні помилки. Миготіння червоного світла означає, що пристрій зафіксував попередження чи ситуацію, яка ще не потребує зупинки, але вимагає уваги. Якщо ж червоне світло горить постійно, це свідчить про серйозну помилку, через яку привід заблоковано або його роботу призупинено.

Світлодіод ASF (Safety Function) (3) відображає стан системи безпеки. Жовтий сигнал цього індикатора означає, що функція безпеки була активована й обмежила або зупинила роботу приводу, забезпечуючи захист обладнання чи персоналу.

У нижній частині панелі розташовані індикатори, що відповідають за з'єднання Ethernet. Світлодіод ETH LNK (4) показує, чи встановлене з'єднання і

з якою швидкістю воно працює. Якщо він вимкнений або зв'язок відсутній. Зелений колір означає активне підключення на швидкості 100 Мбіт/с, а жовтий на 10 Мбіт/с. Коли індикатор блимає, це свідчить про обмін даними в мережі з відповідною швидкістю.

Світлодіод ETH NS (5) інформує про стан мережі Ethernet і з'єднання за протоколом Modbus TCP. Якщо індикатор не горить, пристрій ще не отримав IP-адресу. Миготіння зеленого та червоного сигналів одночасно означає, що пристрій проходить самотестування після вмикання живлення. Коли світиться зеленим, з'єднання Modbus TCP встановлене, і привід успішно підключений до командного слова. Якщо зелений сигнал блимає, це означає, що IP-адреса дійсна, але з'єднання ще не створено. Постійне червоне світло вказує на дублювання IP-адреси в мережі, а червоний сигнал без миготіння вказує на розрив з'єднання або закінчення часу очікування зв'язку.

Індикатор COM (6) відображає активність обміну даними по вбудованому послідовному інтерфейсу Modbus. Коли він блимає жовтим, це свідчить про активну комунікацію між приводом і підключеними пристроями.

Додаткові індикатори (7 – 10) використовуються при встановленні додаткових плат розширення комунікаційних можливостей.

Потужні функціональні можливості, енергоефективність та простота інтеграції роблять Altivar Process ATV630 універсальним рішенням для автоматизації систем водопостачання, вентиляції, перекачування рідин, керування компресорами, транспортними механізмами та інших процесів, де важливе поєднання точності, надійності та економічності.

5.3.2. Налаштування функцій перетворювачів частоти серії ATV630.

Графічні термінали VW3A1111 та VW3A1113 призначені для керування, налаштування та моніторингу перетворювачів частоти серії Altivar (рис. 5.18).

Обидва виконують однакові базові функції відображення робочих параметрів, зміну налаштувань, збереження конфігурацій і перенесення даних між пристроями. Проте між ними існують суттєві відмінності у рівні захисту, якості відображення, температурній стійкості та функціональних можливостях, що визначає їх різне призначення у промислових умовах.

Модель VW3A1113 є простішою та економічнішою у використанні. Вона має ступінь захисту IP21, що робить її придатною для встановлення всередині шафи або у приміщеннях з контрольованими умовами. Її текстовий дисплей із роздільною здатністю 128×64 пікселі забезпечує достатній рівень інформативності для стандартних операцій налаштування та моніторингу. Робоча температура обмежується +50 °С, що також свідчить про орієнтацію на менш агресивні умови експлуатації. Основна перевага VW3A1113 це простота, надійність і зручність приладового монтажу, адже термінал легко встановлюється безпосередньо на передню панель перетворювача або на дверцята шафи (через додатковий комплект IP43). Його достатньо для локального обслуговування одного пристрою, зберігання кількох конфігурацій і оперативного дублювання налаштувань без підключення до зовнішнього ПК.



Рис. 5.18. Графічний термінал: а) VW3A1113, б) VW3A1111

На відміну від нього, VW3A1111 це розширена версія, орієнтована на інтенсивне промислове застосування та роботу у важких умовах (рис. 5.19). Ступінь захисту IP65 дозволяє експлуатувати термінал у запилених, вологих або виробничих середовищах, що особливо важливо при монтажі на дверцятах шаф, встановлених у цехах або на відкритих об'єктах. Дисплей має підвищену роздільну здатність 240×160 пікселів і здатен відображати більше інформації одночасно завдяки 8-рядковому графічному інтерфейсу. Це покращує зручність читання даних, спрощує діагностику помилок та аналіз роботи двигуна в реальному часі.

Функціонально VW3A1111 переважає молодшу модель також завдяки підтримці багатомовного інтерфейсу (24 мови, включно з українською), вбудованій пам'яті 16 МБ для зберігання численних конфігурацій і можливості підключення до кількох перетворювачів одночасно. Це дозволяє використовувати його як централізований термінал для налаштування або обслуговування кількох приводів у системах з каскадним або мережевим керуванням. Для зберігання мітки часу для попереджень, помилок та аварій, в терміналі є годинник реального часу (стандартна батарея, термін 10 років). Крім того, цей термінал може підключатися безпосередньо до ПК для обміну файлами, що робить його ефективним інструментом для технічного персоналу під час налагодження або модернізації обладнання.

В центральній частині графічного терміналу розташовано графічний дисплей (5), який відображає робочі параметри, статуси системи, повідомлення про помилки, а також меню налаштувань і допоміжну інформацію. Під екраном знаходяться кнопки F1...F4 (4), що забезпечують швидкий доступ до ID-адреси перетворювача, QR-коду пристрою, вкладок меню або інших функцій, які можуть бути перепрограмовані відповідно до потреб користувача. В верхній частині панелі розміщено кнопку STOP/RESET (1), що виконує подвійну функцію: зупинку роботи перетворювача частоти та скидання аварій або несправностей. Поруч із нею розташована кнопка LOCAL/REMOTE (2), яка дозволяє перемикатися між локальним режимом керування (через термінал) і

дистанційним (через мережу або зовнішні сигнали керування). Для навігації по меню використовується кнопка ESC (3), яка дозволяє виходити з поточного параметра, підменю або режиму довідки, а також скасовувати відображене значення з поверненням до попереднього збереженого стану. Кнопка Home (6) забезпечує швидкий перехід до кореневого каталогу меню, спрощуючи орієнтацію у структурі налаштувань, а кнопка Information (7) відкриває довідкові відомості про поточне меню, підменю або обраний параметр. В нижній частині розташована кнопка RUN (8), яка запускає виконання призначеної функції або командного сценарію, якщо він попередньо заданий у конфігурації. Основним елементом управління є навігаційний джойстик (9) універсальний інструмент для переміщення по меню та редагування параметрів. Обертання джойстика дозволяє швидко прокручувати списки, стрілки вгору та вниз забезпечують покрокову навігацію, а стрілки вліво та вправо використовуються для зміни значень параметрів.

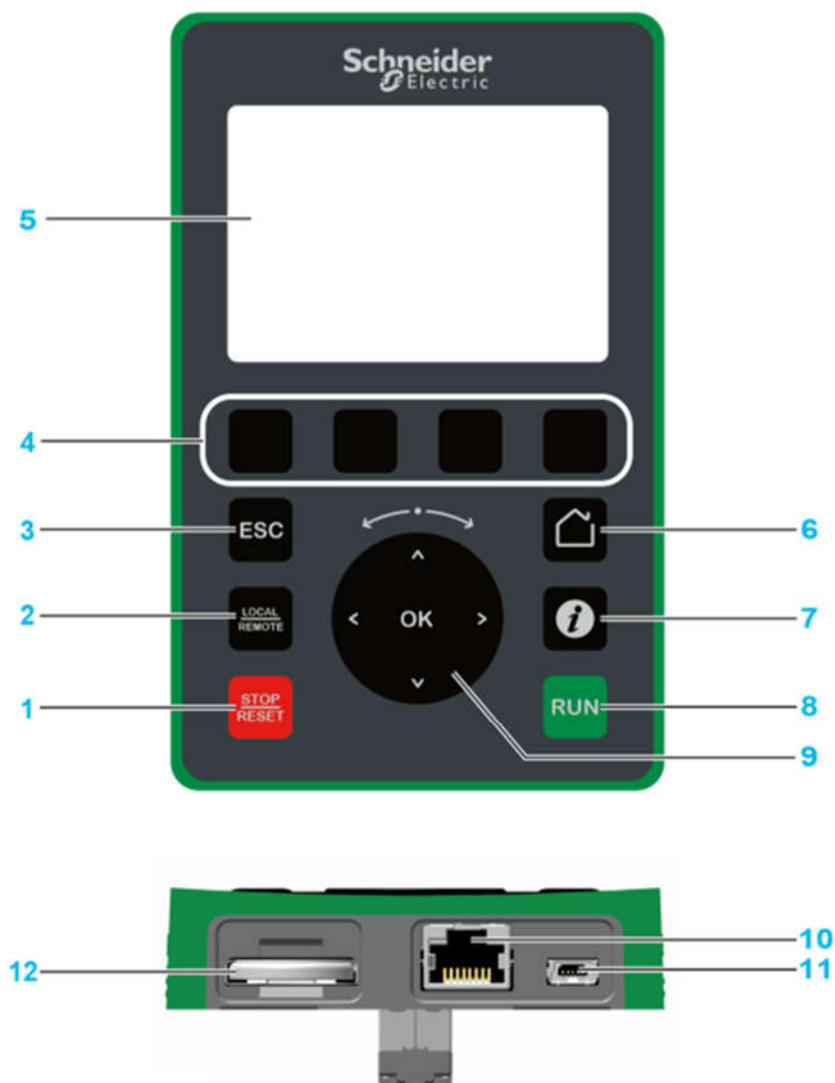


Рис. 5.19. Графічний термінал VW3A111

Натискання на центральну частину джойстика (кнопка OK) підтверджує вибір або зберігає внесені зміни. Для організації зв'язку термінал оснащено двома

інтерфейсами: RJ45 Modbus-портом (10), який використовується для підключення графічного терміналу до перетворювача частоти під час дистанційного керування, та MiniB USB-портом (11), що забезпечує з'єднання з комп'ютером для передавання файлів конфігурацій або оновлення прошивки. Живлення внутрішнього годинника реального часу та збереження системних налаштувань забезпечується вбудованою батареєю типу CR2032 (12) із терміном служби до десяти років. Позитивний полюс батареї спрямований до передньої панелі терміналу, що спрощує її заміну під час технічного обслуговування.

Графічний дисплей терміналу має чітку логічну структуру відображення інформації, що забезпечує зручність роботи користувача та швидкий доступ до необхідних параметрів і налаштувань (рис. 5.20).

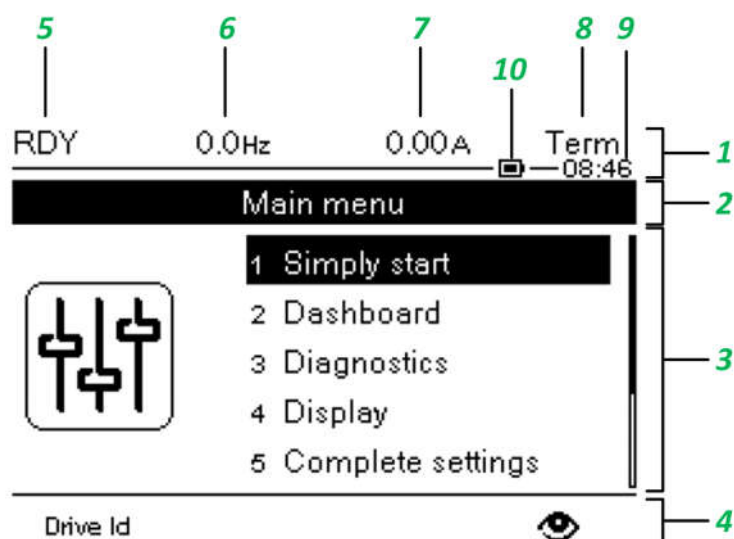


Рис. 5.20. Графічний дисплей терміналу VW3A1111

У верхній частині розташований рядок стану (1), який може бути повністю сконфігурований відповідно до потреб оператора або системи. У цьому рядку відображається основна інформація про поточний стан перетворювача частоти або двигуна (5): ACC – розгін; DEC – уповільнення; RUN – двигун працює або отримано команду пуску при нульовому завданні; RDY – пристрій готовий до роботи; NST – зупинка на вибігу; FLT – виявлено несправність; STO – активована функція безпеки STO; FA, EP – помилки автотестування або пам'яті EEPROM; NLP, USA – відсутність або зниження напруги мережі; OBR, DCP, DCB – процеси гальмування чи керування енергією; TUN, ST, YES – режими автоналаштування; SOC, CTL, CLI, FLU, IDLE – додаткові режими контролю, обмеження струму, намагнічування чи очікування. Користувацькі поля (6-7), що призначені для відображення даних, які визначаються користувачем (наприклад, частота, навантаження, температура, лічильники тощо). Обраний активний канал керування (8), звідки надходять команди керування приводом: TERM – через клемник; HMI – з графічного дисплейного терміналу; MDB – через інтегрований послідовний інтерфейс Modbus; CAN – через мережу CANopen; NET – за допомогою зовнішньої комунікаційної карти; ETH – через інтегрований Ethernet

Modbus TCP. Також відображається системний час реального годинника (9), що зберігається незалежно від живлення та рівень заряду батареї (10).

Під рядком стану (1) розміщується рядок заголовка (2), у якому зазначається назва поточного меню, підменю або конкретного параметра, з яким працює користувач. Цей елемент допомагає орієнтуватися у структурі меню та чітко розуміти, у якій частині системи налаштувань перебуває оператор у даний момент.

Основну частину екрана займає область відображення вмісту (3). Вона виконує роль робочого вікна для перегляду й редагування інформації, що може містити списки меню та підменю, параметри з числовими значеннями, текстові повідомлення, а також індикаторні шкали або барграфи, які відображають поточні значення струму, швидкості, температури двигуна чи інших змінних. Ця зона реалізована у вигляді прокручуваного вікна висотою до п'яти рядків, що забезпечує компактність та водночас повноцінне відображення інформації.

У нижній частині дисплея розташовані мітки поточних функцій клавіш F1...F4 (4). Їхній зміст змінюється залежно від відкритого меню або активного режиму. Тут можуть відображатися вкладки (від однієї до чотирьох), ярлики швидкого переходу до екрана відображення змінних, кнопки виклику довідки або інформації про перетворювач (Drive ID) тощо. Така організація елементів дозволяє оперативно виконувати дії без необхідності багаторазового перемикання між меню.

Головне меню графічного терміналу забезпечує швидкий доступ до всіх основних функцій перетворювача частоти, дозволяючи здійснювати налаштування, моніторинг, діагностику та обмін даними (рис. 5.21). Навігація по меню виконується за допомогою джойстика та клавіш керування, що забезпечує інтуїтивну роботу оператора.

Головне меню:

1. Simply Start (Прискорений Запуск). Призначене для швидкого введення перетворювача в експлуатацію. У цьому підменю користувач може задати базові параметри двигуна (номінальна напруга, струм, частота, кількість полюсів), напрям обертання, тип керування, джерела команд пуску та завдання частоти. Меню орієнтоване на швидке налаштування без заглиблення у повну структуру параметрів.
2. Dashboard (Екран Відображення). Відображає основні робочі параметри в реальному часі: частоту виходу, струм, потужність, стан двигуна, температуру, активний режим; можливість створення власної панелі моніторингу для візуалізації потрібних змінних у вигляді чисел, графіків або індикаторних шкал.
3. Diagnostic (Діагностика). Меню призначене для контролю справності системи та аналізу подій. У цьому розділі можна переглядати: журнал помилок і попереджень; історію останніх спрацьовувань; інформацію про стан силових модулів, температуру, мережеві сигнали; рекомендації щодо усунення несправностей.

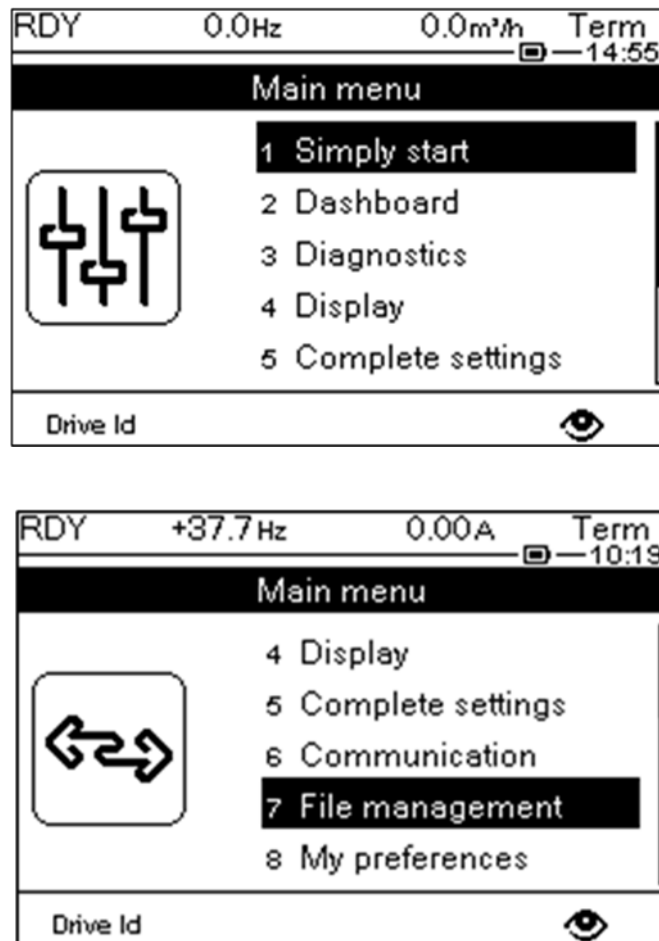


Рис. 5.21. Головне меню графічного терміналу

4. Display (Відображення). Призначений для детального моніторингу параметрів роботи перетворювача частоти, двигуна, а також підключених систем і периферійних пристроїв. Це меню дозволяє користувачеві відображати та контролювати в реальному часі: енергетичні параметри, параметри застосування, параметри насоса, параметри двигуна, параметри перетворювача, тепловий моніторинг, відображення ПД-регулятора, керування лічильниками, карта комунікацій, журнал даних.
5. Complete Setting (Повне Налаштування). Повний доступ до всіх параметрів перетворювача. Використовується для детального налаштування всіх режимів роботи, комунікацій, захистів, функцій керування двигуном, аналогових і цифрових входів/виходів, логічних блоків, фільтрів, оптимізації енерговитрат. Це меню призначене для досвідчених користувачів і технічних спеціалістів.
6. Communication (Комунікація). Параметри налаштування зв'язку з іншими пристроями системи автоматизації. Дозволяє обрати протокол обміну (Modbus, CANopen, Ethernet Modbus TCP, інше), задати адресу, швидкість передачі, тайм-аути та перевірку зв'язку.
7. Files Management (Керування Файлами). Служить для збереження, копіювання, відновлення або передавання конфігураційних файлів між терміналом, перетворювачем і комп'ютером. Можливе створення

резервних копій параметрів або перенесення налаштувань на інші пристрої.

8. My Preferences (Індивідуальні Налаштування). Призначене для персоналізації роботи оператора. Тут користувач може налаштувати вигляд меню, мову, рівень доступу, швидкість прокручування, попередження, комбінації клавіш, параметри безпеки тощо.

Розглянемо налаштування для швидкого старту. Для цього переходимо в меню Simply start (рис. 5.22). Задаємо технічні характеристики двигуна (за табличкою) та встановлюємо тип керування (дво- або три провідне). Проводимо автопідстроювання двигуна використовуючи Autituning. Під час виконання цієї процедури перетворювач частоти вимірює основні характеристики двигуна: опір та індуктивність обмоток статора.

Головною особливістю приводів серії ATV6xx є наявність вбудованих прикладних функцій (або макроконфігурацій), оптимізованих під конкретні типи навантаження: насоси, вентилятори, компресори тощо (рис. 5.23).

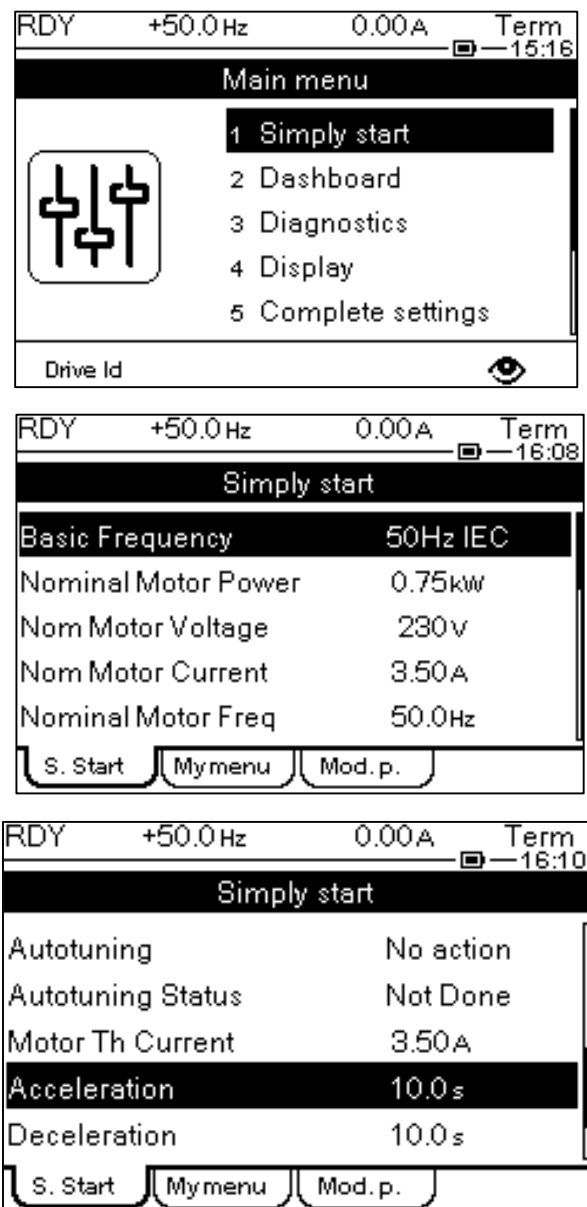


Рис. 5.22. Меню Simply start графічного терміналу

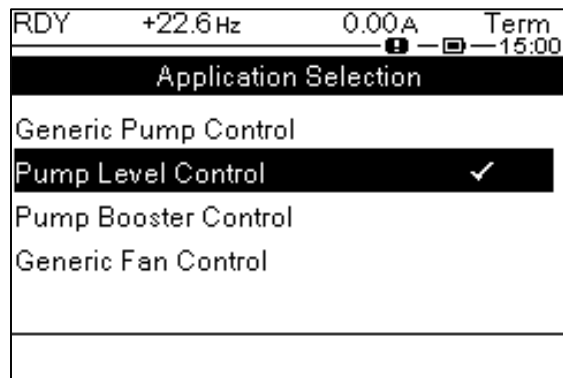


Рис. 5.23. Меню вибору макроконфігурацій (застосування) графічного терміналу

Розглянемо детальний опис і параметри налаштування чотирьох базових функцій, що найчастіше застосовуються у практиці.

Generic Pump Control (Загальне керування насосом). Функція *Generic Pump Control* використовується для керування одним або кількома насосами, забезпечуючи плавний пуск, регулювання швидкості та захист від аварійних режимів. Вона дозволяє підтримувати необхідні параметри тиску, витрати або рівня у системах з постійним або змінним навантаженням.

Перетворювач змінює частоту обертання електродвигуна залежно від сигналу з датчика зворотного зв'язку (наприклад, тиску чи витрати). Використовуючи вбудований ПІД-регулятор, пристрій порівнює реальне значення із заданим і змінює швидкість насоса для підтримання сталого рівня параметра. Додатково передбачено функції антизаклинювання, захисту від сухого ходу, автоматичного чергування насосів і енергозбереження при частковому навантаженні.

Основні параметри налаштування: [PPr] Process Setpoint – задане значення параметра (тиску, витрати, рівня); [Fr1] Reference Channel 1 – джерело завдання (аналоговий вхід, Modbus, НМІ тощо); [FLt] Flow Threshold – мінімальна витрата, нижче якої вмикається захист від сухого ходу; [tFd] Detection Delay – затримка перед спрацьовуванням захисту; [nCr] Minimum Speed і [nFr] Maximum Speed – мінімальна та максимальна частота обертання двигуна; [AtP] Auto Tuning PID – автоматичне налаштування параметрів ПІД-регулятора; [rP1]/[rP2] Rotation Time – час розгону та гальмування.

Pump Level Control (Керування рівнем у резервуарі). Функція *Pump Level Control* застосовується для керування насосами, що підтримують рівень рідини у резервуарах, басейнах чи водонапірних вежах. Вона забезпечує автоматичне вмикання та вимикання насоса залежно від сигналів від датчиків рівня, а також плавне регулювання подачі при необхідності.

Система використовує сигнал від аналогового або дискретного датчика рівня. При досягненні верхнього порогу рівня насос зупиняється, а при зниженні нижче мінімального запускається. У разі наявності аналогового датчика рівня можливе плавне регулювання подачі для стабілізації рівня.

Основні параметри налаштування: [LL1] Low Level Threshold – нижня межа рівня; [LL2] High Level Threshold – верхня межа рівня; [LLH] Hysteresis – різниця

між рівнями вмикання/вимикання для уникнення частих циклів; [FLL] Flow Loss Detection – виявлення втрати потоку (сухий хід); [tOFF] Stop Delay – затримка вимкнення після досягнення рівня; [rSt] Restart Delay – час затримки повторного запуску насоса після зупинки.

Pump Booster Control (Каскадне або підсилювальне керування насосами). Функція Pump Booster Control (або Multi-Pump Control) використовується у системах з кількома насосами, де необхідно підтримувати постійний тиск або витрату шляхом автоматичного підключення та відключення додаткових насосів. Типовими прикладами є насосні станції, системи охолодження, зрошення та підживлення котлів.

Один перетворювач частоти керує основним насосом з регулюванням швидкості, тоді як додаткові насоси можуть вмикатись або вимикатись за сигналом від ПЧ або зовнішнього контролера. Система підтримує оптимальний баланс між продуктивністю і споживанням енергії. Реалізується функція чергування насосів, що забезпечує рівномірний знос обладнання.

Основні параметри налаштування: [nPS] Number of Pumps – кількість насосів у системі; [PSM] Pump Switching Mode – логіка перемикавання між насосами (за часом або навантаженням); [rPr] Regulation Pressure – значення тиску, яке необхідно підтримувати; [tPS] Pump Switching Delay – затримка між увімкненням додаткових насосів; [rOt] Rotation Time – інтервал чергування насосів; [dEL] Delay On Stop – час затримки перед вимкненням насосів.

Generic Fan Control (Загальне керування вентилятором). Функція Generic Fan Control застосовується для регулювання швидкості обертання вентиляторів у системах вентиляції, кондиціонування, охолодження та витяжки. Вона забезпечує плавне керування продуктивністю вентилятора відповідно до потреб процесу, знижуючи рівень шуму та споживання енергії.

Перетворювач регулює швидкість вентилятора залежно від сигналу з датчика тиску або температури. У системі може бути використаний ПД-регулятор для стабілізації параметра, наприклад тиску в повітроводі або температури в приміщенні. Передбачено функції зворотного обертання, очищення фільтра, плавного пуску та захисту від перевантаження.

Основні параметри налаштування: [tSP] Temperature Setpoint або [PrS] Pressure Setpoint – задане значення температури (тиску); [FAn] Fan Type – тип вентилятора (осьовий, відцентровий тощо); [FLt] Filter Cleaning Function – активація очищення фільтра (реверсивний оберт); [nCr]/[nFr] – мінімальна та максимальна швидкість вентилятора; [PID] PID Settings – коефіцієнти ПД-регулятора (P, I, D); [Stt] Start Mode – тип запуску (плавний, прискорений, автоматичний); [rEv] Reverse Run – дозвіл на зворотний напрям обертання.

5.4. Перетворювачі частоти Altivar Process 930.

5.4.1. Апаратні та функціональні можливості перетворювачів частоти серії ATV930.

Перетворювачі частоти серії Altivar Process ATV930 компанії Schneider Electric є високопродуктивними інтелектуальними пристроями для керування електроприводами у найвідповідальніших технологічних процесах

промисловості. Вони належать до лінійки Altivar Process, подібно до серії ATV630, проте мають розширені апаратні та програмні можливості, орієнтовані не лише на регулювання швидкості, а й на повноцінне управління технологічними об'єктами із високими динамічними навантаженнями, змінним моментом та складною логікою керування.

Якщо ATV630 призначений передусім для систем з вентиляторними, насосними та компресорними механізмами, тобто процесів із плавною динамікою, то ATV930 створено для високопродуктивних і технологічно складних машинних застосувань – приводів з підйомними механізмами, екструдерами, дробарками, млинами, змішувачами, конвеєрними системами, пресами, приводами у гірничо-металургійній, хімічній, нафтохімічній та енергетичній промисловості.

Загальні характеристики серії ATV930 перевищують можливості ATV630 за потужністю, швидкодією та гнучкістю налаштувань. Пристрої охоплюють діапазон потужностей від 0.75 до 500 кВт і доступні у версіях для напруги 380...690 В (рис. 5.24). Це дозволяє використовувати їх у потужних системах середньої напруги та великих електроприводах із високим моментом інерції.

Конструктивно ATV930 випускається у кількох варіантах виконання корпусу – стандартному (IP21) для монтажу в шафах керування та пиловологозахищеному (IP54) для роботи у відкритих або агресивних промислових середовищах. Робоча температура без зниження потужності сягає 50 °С, а при належному охолодженні – до 60 °С. Захисне покриття електронних плат відповідає вимогам IEC 60721-3-3 класу 3С3, що забезпечує стабільність роботи у середовищах із підвищеною запиленістю, вологістю чи наявністю хімічних домішок.

Серія ATV930 підтримує всі режими керування, властиві ATV630, і додатково реалізує розширені векторні алгоритми зворотного зв'язку, що забезпечують точне керування моментом навіть при дуже малих швидкостях або у стані утримання навантаження. Завдяки цьому перетворювач може замінювати системи з енкадером, забезпечуючи високий крутний момент при нульовій швидкості (standstill torque control).

Параметр [Motor control type] (СТТ selection) визначає алгоритм керування, що застосовується для конкретного типу двигуна. Вибір типу контролю залежить від вимог до точності швидкості, стабільності моменту, динамічної реакції та енергоефективності.

ATV930 підтримує вісім типів алгоритмів керування, що дозволяють адаптувати систему до будь-якого типу електричної машини:

- [SVC V] Voltage Vector Control (VVC) – векторне керування напругою з компенсацією ковзання для асинхронних двигунів; використовується у стандартних промислових установках.
- [U/F VC 5pts] UF5 – універсальний закон U/f із п'ятьма точками налаштування; призначений для вентиляторів і насосів.
- [Energy Sav.] NLD – енергозберігаючий режим; автоматично оптимізує подачу енергії в двигун для зменшення втрат.

- [Sync. mot.] SYN – керування синхронними двигунами з постійними магнітами для високоефективних процесів.
- [SYN_U VC] SYNU – режим для синхронних двигунів зі змінним моментом навантаження.
- [Reluctance Motor] SRVC – алгоритм для двигунів із синхронним реактивним ротором, що забезпечує високу енергоефективність без постійних магнітів.
- [FVC] Field Vector Control – векторне керування струмом (закритий контур) для високої точності моменту і швидкості.
- [Sync.CL] FSY – режим для синхронних двигунів із зворотним зв'язком, що забезпечує сервоподібне керування.



Рис. 5.24. Перетворювач частоти серії Altivar Process ATV930 компанії Schneider Electric

Це робить ATV930 універсальним рішенням для систем, де потрібна висока точність, повторюваність і стабільність динаміки – наприклад, у роботизованих лініях, пресах, позиціонувальних механізмах або підйомних пристроях.

На відміну від ATV630, який орієнтований на енергозбереження і просте керування насосами та вентиляторами, ATV930 має розширений набір апаратних та програмних можливостей, зокрема:

- точне керування моментом та векторне керування зворотного зв'язку (Closed-Loop);
- підтримку різних типів двигунів (асинхронних, синхронних, реактивних);
- гнучкі параметри динамічного регулювання (розгін, гальмування, утримання моменту);
- висока перевантажувальна здатність до 150% номінального струму.

Перетворювач ATV930 має розширену систему діагностики, яка не лише фіксує помилки, а й прогнозує можливі відмови, аналізуючи тенденції зміни параметрів у часі (Predictive Maintenance) з використанням хмарних технологій.

Частотний перетворювач ATV930 серії Altivar Process оснащений сучасними вбудованими функціями безпеки та системами моніторингу, які забезпечують ефективний захист персоналу і обладнання від аварійних режимів роботи. Вбудована функція STO (Safe Torque Off) – безпечне відключення моменту, що відповідає вимогам стандартів SIL3/PLe. Забезпечує негайне знеструмлення двигуна без необхідності відключення живлення всього перетворювача. Функції моніторингу стану привода та двигуна, які допомагають попереджати передчасне зношення, перегрів, перевантаження або інші аномальні ситуації. Це дозволяє ATV930 застосовувати у відповідальних механізмах, де безпека руху має критичне значення, наприклад, у підйомних кранах або транспортних системах.

З точки зору комунікаційних можливостей, перетворювач Altivar Process ATV930 підтримує весь набір промислових протоколів, що й ATV630, але додатково має вбудовану підтримку Ethernet-сервісів (веб-інтерфейс, FTP, SNMP, DHCP, Modbus TCP, Ethernet/IP, Profinet), а також опцію Dual Port Ethernet для реалізації резервування зв'язку (Redundant Ring).

Керування, параметрування і моніторинг можуть здійснюватися за допомогою: програмного забезпечення SoMove; веб-сервера (через браузер); інтерфейсу EcoStructure Machine Expert / Process Expert; панелі Human Machine Interface (HMI) із сенсорним дисплеєм.

ATV930 має більшу кількість аналогових та дискретних входів і виходів, що дозволяє підключати додаткові датчики, енкодери, температурні елементи (Pt100, Pt1000, термопари) (рис. 5.25). Підтримується робота з енкодерами типів HTL, TTL, SinCos, SSI, що забезпечує надвисоку точність позиціонування.

Таким чином, Altivar Process ATV930 можна розглядати як розширену та потужнішу версію ATV630, оптимізовану для машинних і технологічно складних систем, де необхідні швидке реагування, точне керування моментом, підтримка зворотного зв'язку, високий рівень безпеки та інтеграція з цифровими системами керування підприємства.

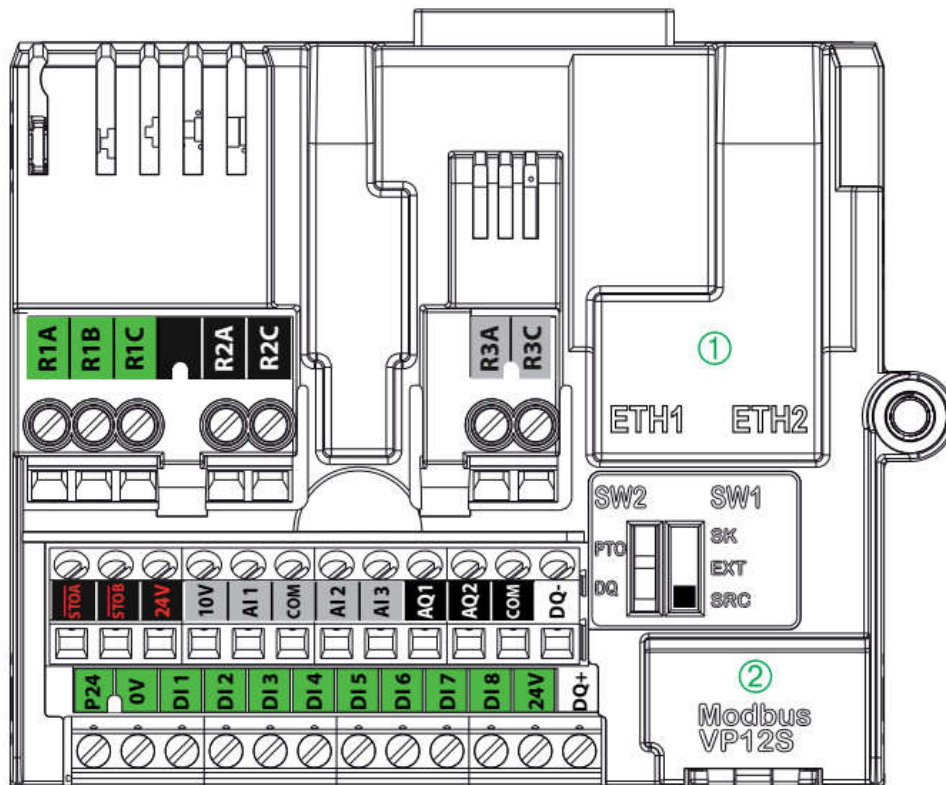


Рис. 5.25. Клеми блоку управління перетворювача частоти серії Altivar Process ATV930: 1 – Ethernet Modbus TCP, 2 – Serial Modbus

5.4.2. Налаштування функцій перетворювачів частоти серії ATV930.

Налаштування ATV930 здійснюється аналогічно до ATV630, оскільки обидва приводи належать до серії Altivar Process та мають однакову логіку роботи графічного терміналу VW3A1111. Процес налаштування виконується через головне меню терміналу, яке містить такі основні пункти:

1. Simply Start (Прискорений запуск) – базове налаштування параметрів двигуна та запуск системи.
2. Dashboard (Екран відображення) – відображення основних робочих параметрів і стану приводу.
3. Diagnostic (Діагностика) – перегляд активних попереджень, аварій і журналу подій.
4. Display (Відображення) – вибір параметрів, що відображаються на екрані, таких як струм, напруга, потужність, PID-регулювання тощо.
5. Complete setting (Повне налаштування) – розширені параметри, що охоплюють режими керування, налаштування входів/виходів, PID-регулятор, захист, розширені параметри налаштування векторного керування двигунами різних типів (асинхронні, синхронні з постійними магнітами, реактивні).
6. Communication (Комунікація) – параметри обміну даними через Modbus, Ethernet, CANopen та інші інтерфейси.
7. Files management (Керування файлами) – збереження, копіювання або відновлення конфігураційних файлів.

8. My Preferences (Індивідуальні налаштування) – налаштування інтерфейсу користувача (мова, яскравість екрана, вибір відображуваних одиниць).

Частотний перетворювач ATV930 відрізняється від ATV630 розширеними можливостями, орієнтованими на високопродуктивні технологічні процеси, де потрібне точне керування рухом і швидка реакція системи. У цьому пристрої реалізовано спеціальні робочі режими: Generic Pump, Hoisting, Conveyor та All Applications, які дозволяють швидко адаптувати привід під конкретне завдання, будь то робота з насосами, підйомними механізмами чи конвеєрними лініями.

Розглянемо детально функціональне призначення цих режимів, логіку їх налаштування та визначальні параметри, що впливають на роботу приводу в кожному конкретному застосуванні.

Режим *Generic Pump* (загальне керування насосами) орієнтований на керування насосними системами з урахуванням змінного навантаження, гідравлічних втрат і потреби в енергоефективності. У цьому режимі перетворювач реалізує функції плавного пуску та зупинки, захисту від гідродару, підтримання тиску або витрати шляхом регулювання частоти обертання двигуна. Налаштування виконується через вибір типу насоса (центробіжний, осьовий або шнековий), введення робочих характеристик системи та задання уставок тиску або швидкості. Основними параметрами є межі частоти, час розгону й гальмування, коефіцієнти ПД-регулятора та логіка аварійного відключення при перевантаженні або кавітації.

Режим *Hoisting* (підйомні механізми) призначений для систем підйому, наприклад кранів, ліфтів, тельферів і вантажопідіймальних механізмів, де критичними є стабільність моменту, утримання навантаження та точність зупинки. Перетворювач у цьому режимі використовує векторне керування з замкненим контуром моменту, що забезпечує миттєву реакцію на зміни навантаження. Логіка налаштування передбачає визначення типу двигуна, способу вимірювання моменту та режиму гальмування, а також активацію функцій динамічного гальмування й компенсації моменту утримання. Ключовими параметрами виступають максимальний крутний момент, обмеження швидкості, контроль перевантаження та налаштування ПД-регулятора для плавного керування підйомом і спуском вантажу.

Режим *Conveyor* (транспортери) оптимізований для приводу транспортерних систем, що потребують синхронізації з іншими механізмами лінії. У цьому режимі частотний перетворювач забезпечує рівномірний рух із заданою швидкістю, плавне прискорення і зупинку, а також компенсує коливання навантаження. При налаштуванні користувач задає тип транспортера, довжину лінії, характер динаміки та можливість роботи в реверсивному напрямку. Важливими параметрами є прискорення, уповільнення, обмеження струму, а також функції синхронізації з іншими приводами через комунікаційні протоколи, такі як Modbus TCP або EtherNet/IP.

Режим *All Applications* (універсальні налаштування) є універсальним і надає повний доступ до всіх функцій перетворювача, дозволяючи гнучко адаптувати його під будь-який тип промислового обладнання, від вентиляторів до

змішувачів чи дозаторів. У цьому режимі користувач може вручну визначати алгоритм керування, вибрати тип векторного або скалярного управління, встановлювати граничні параметри струму, моменту та частоти. Основною перевагою є можливість тонкого налаштування динаміки приводу під специфічні умови експлуатації без обмежень типової конфігурації.

В частотних перетворювачах Altivar Process ATV930 реалізовано низку розширених режимів взаємодії між приводами, які забезпечують ефективну роботу в системах із кількома двигунами або складною кінематичною структурою. До таких режимів належать Load Sharing, Master/Slave та MultiDrive Link, вони призначені для узгодження моменту, швидкості та навантаження між декількома приводами, що працюють на один спільний механізм або процес.

Режим *Load Sharing* (розподіл навантаження) застосовується в системах, де кілька двигунів одночасно обертають спільний вал або механічно з'єднані елементи, наприклад, у насосних установках, конвеєрних секціях великої довжини, змішувачах або екструдерах. Основна мета цього режиму полягає у рівномірному розподілі механічного навантаження між усіма двигунами, щоб запобігти перевантаженню одного з них і підвищити загальну ефективність системи.

Логіка роботи полягає в тому, що один із перетворювачів визначається як ведучий і задає швидкість або момент обертання, а решта пристроїв регулюють свій вихідний момент відповідно до сигналу навантаження, який передається між приводами через комунікаційний канал (наприклад, Modbus, CANopen або Ethernet). Визначальними параметрами для налаштування є режим обміну даними, коефіцієнти розподілу моменту, межі синхронізації швидкості та фільтри компенсації динаміки, які гарантують стабільний і плавний розподіл потужності між двигунами.

Режим *Master/Slave* (ведучий/ведений) передбачає організацію ієрархічного керування між кількома приводами, де один з них Master формує основну команду швидкості, моменту або положення, а інші Slave повторюють її, синхронізуючи свої дії. Такий принцип особливо ефективний у транспортних системах, підійомно-транспортному обладнанні, а також у процесах, де необхідно зберігати точне співвідношення швидкостей між механізмами.

В режимі Master/Slave перетворювач ведучого двигуна безпосередньо зчитує технологічні параметри, а потім передає їх іншим перетворювачам по шині зв'язку. Ведені приводи здійснюють корекцію своїх параметрів, використовуючи замкнений контур регулювання, що враховує відхилення між фактичною та командною швидкістю. Основними параметрами при налаштуванні є тип синхронізації (швидкість, момент або положення), коефіцієнти регуляторів узгодження, а також час реакції на зміни в системі. Такий підхід дозволяє зберігати точність руху навіть у разі зміни навантаження або незначних механічних розбіжностей між приводами.

Режим *MultiDrive Link* є розширенням принципів попередніх двох режимів і використовується у комплексних системах з декількома взаємопов'язаними приводами, які можуть одночасно взаємодіяти, обмінюватися даними та координувати дії на основі єдиної логіки керування. Цей режим забезпечує

повноцінну інтеграцію приводів у єдину мережеву структуру, де кожен частотний перетворювач може виконувати роль і ведучого, і веденого залежно від умов технологічного процесу.

Комунікація між пристроями здійснюється через високошвидкісні промислові протоколи (EtherNet/IP, Modbus TCP, CANopen), що гарантують мінімальні затримки та високу точність синхронізації. Логіка налаштування передбачає визначення ролі кожного приводу в мережі, налаштування параметрів пріоритету команд, затримки реакції та механізмів захисту від конфліктів команд. Визначальними параметрами є топологія мережі, інтервали оновлення даних, синхронізаційна частота та тип переданих змінних (швидкість, момент, положення).

Такий режим дозволяє створювати інтегровані системи керування, де декілька приводів працюють як єдиний логічний об'єкт, що підвищує точність керування, скорочує час відгуку та спрощує реалізацію складних технологічних алгоритмів.

Режими Load Sharing, Master/Slave та MultiDrive Link у частотних перетворювачах ATV930 розширюють стандартні функції приводу, забезпечуючи синхронну та збалансовану роботу кількох двигунів, підвищуючи стабільність і надійність технологічного процесу, зменшуючи механічний знос і надаючи системам автоматизації гнучкість та масштабованість.

5.5.3 Приклад використання MultiDrive Link.

Розглянемо процес керування рівнем спорожнення резервуара за допомогою 3 ПЧ Altivar Process ATV600 з регульованою швидкістю.

Для прикладу розглянемо наступну схему (рис. 5.26):

- резервуар з корисною висотою 5 м;
- 3 ПЧ Altivar Process ATV600, підключені через MultiDrive Link;
- 3 Модулі польової шини VW3A3721 Ethernet/IP / ModbusTCP
- 3 насоси та відповідні електродвигуни;
- датчик рівня (4-20 мА, 0-10 м).

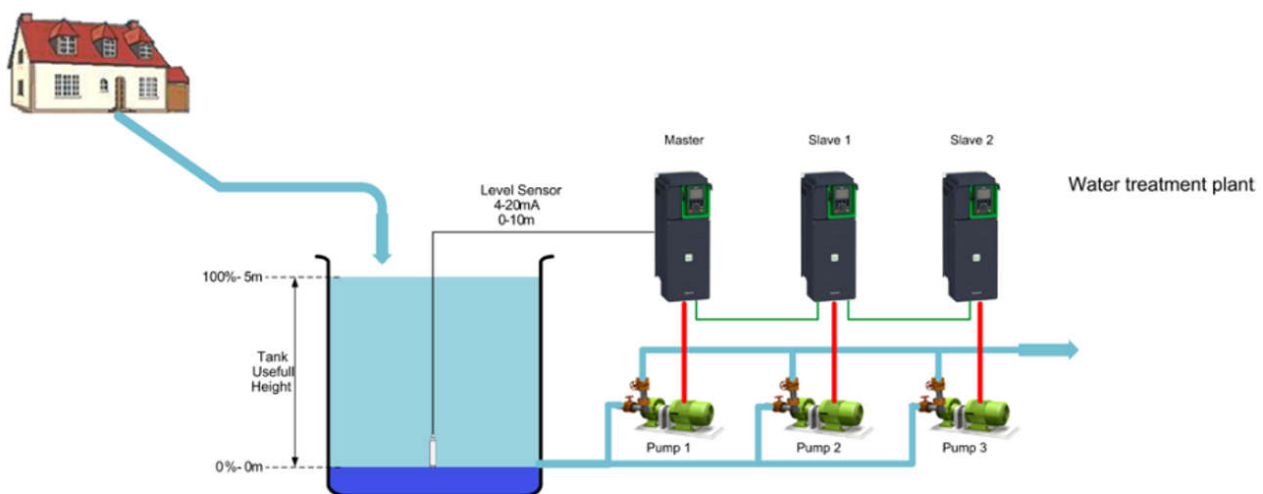


Рис. 5.26. Схема процесу керування рівнем спорожнення резервуара за допомогою 3 ПЧ Altivar Process ATV600 з регульованою швидкістю



Рис. 5.27. Модулі польової шини VW3A3721 Ethernet/IP / ModbusTCP

Використовуючи технічне завдання та довідкову літературу зі схем підключення ПЧ розроблюємо схему підключення ПЧ (рис. 5.28).

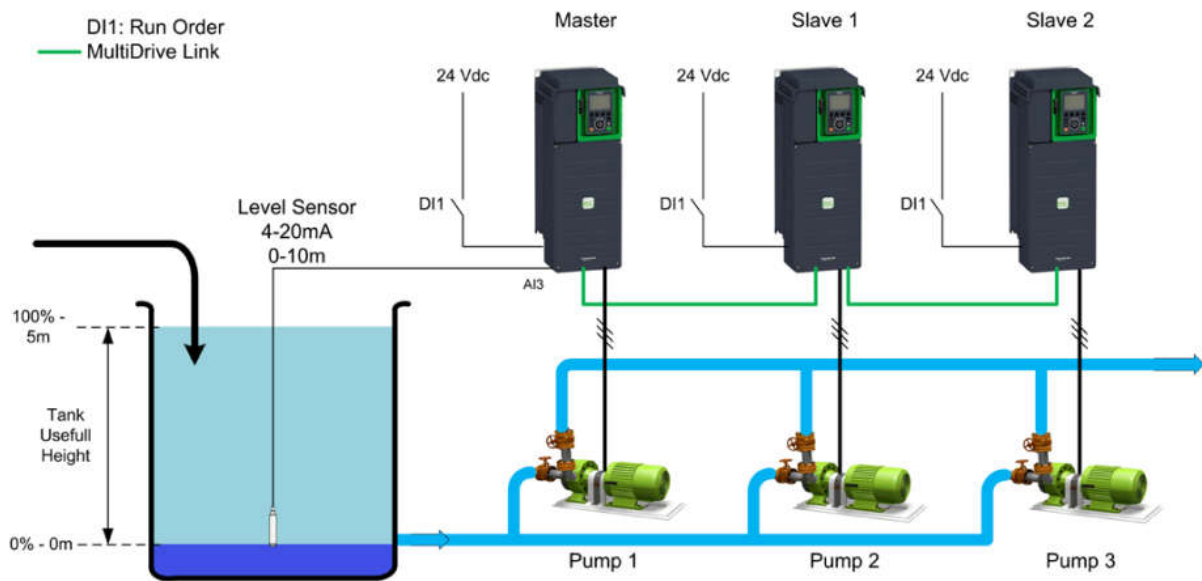


Рис. 5.28. Схема підключення 3 ПЧ Altivar Process ATV600 та датчика аналогового введення (AI3)

В меню макроконфігурацій обираємо Pump Level Cotrol (рис. 5.29), щоб отримати доступ до відповідних функцій програми і пов'язаних з ними параметрів.

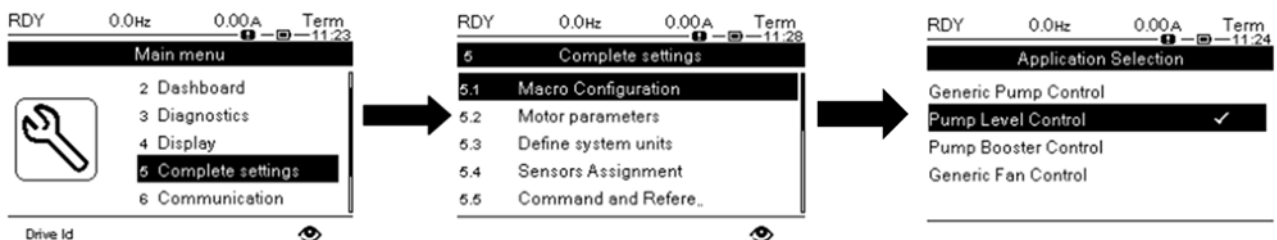


Рис. 5.29. Налаштування макроконфігурації Pump Level Cotrol

Наступним кроком необхідно налаштувати параметри двигуна (рис. 5.30), наприклад:

[Motor Standard] bFr: [50 Hz IEC] 50 (Hz)

[Nominal Motor Power] nPr: 0.37 kW
 [Nom Motor Voltage] UnS: 230 Vac
 [Nom Motor Current] nCr: 2.25 A
 [Nominal Motor Freq] FrS: 50.00 Hz
 [Nominal Motor Speed] nSP: 1340 rpm
 [2/3-Wire Control] tCC: 2C
 [Max Frequency] tFr: 50 Hz

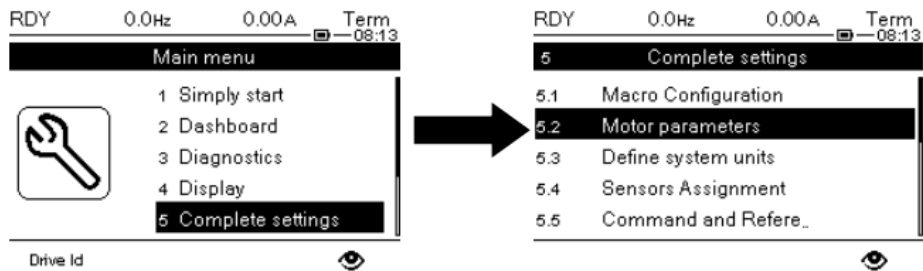


Рис. 5.30. Налаштування параметрів двигуна

Обов'язково необхідно перевірити напрямок обертання ротора двигуна:
 [Complete settings] CST → [Motor parameters] MPA → [Motor control] DRC,
 встановити [Output Ph Rotation] PHR в [ACB] ACB

Встановлюємо системні одиниці вимірювання (рис. 5.31).

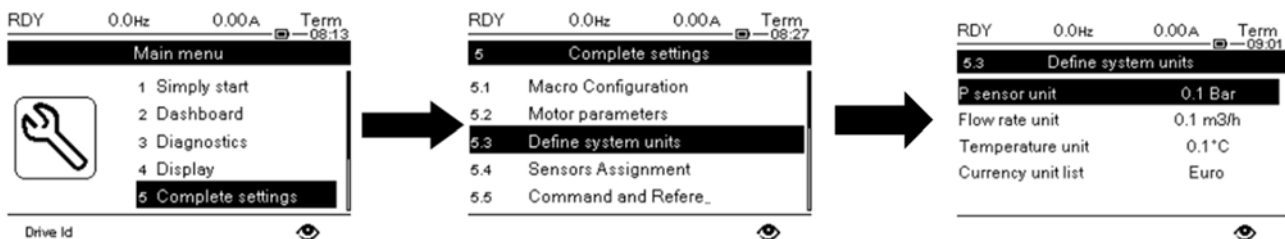


Рис. 5.31. Налаштування системних одиниць вимірювання

Привід Altivar Process ATV600 має кілька командних і додаткових каналів. В цьому прикладі використовуємо заводську конфігурацію: команда і опорне значення надходять з клем (рис. 5.32).

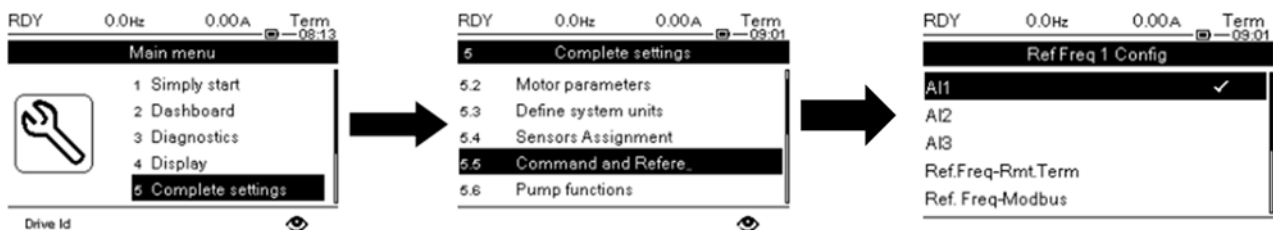


Рис. 5.32. Налаштування заданого та керуючого сигналу

Функція MultiDrive Link в Altivar Process ATV600, які оснащені модулями Ethernet VW3A3721, дозволяє керувати приводами в нашому прикладі за допомогою Ethernet-з'єднання між ПЧ. Для цього ПЧ повинні бути підключені до однієї мережі Ethernet (рис. 5.33).



Рис. 5.33. Налаштування Ethernet

ПЧ Altivar Process ATV600 можна використовувати в декількох архітектурах насосів. System Architecture використовується для вибору та налаштування однієї з наступних архітектур:

- Single Drive: один ПЧ Altivar Process ATV600 і до 6 насосів з фіксованою швидкістю;
- Multi Drives: один головний Altivar Process ATV600 і до 6 ведених Altivar Process ATV600;
- Multi Masters: один головний Altivar Process ATV600 і до 6 ведених Altivar Process ATV600, які можуть діяти як ведучі або ведені.

Обираємо архітектуру Multi Drives (рис. 5.34).

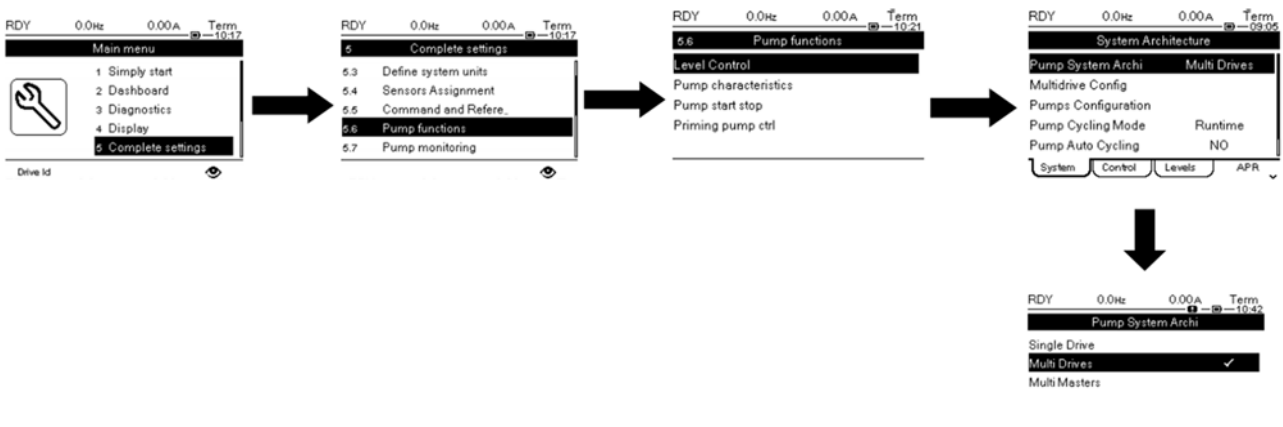


Рис. 5.34. Налаштування архітектури системи

MultiDrive зв'язок потрібно налаштувати на кожному ПЧ, щоб визначити, який пристрій у програмі працює як ведучий, а який як ведений. В прикладі один ПЧ виконує роль ведучого, а два – ведених.

Встановити параметри:

- Встановити [M/P Device Role] MPDT в [Master] MAST;
- Встановити [Nb of Devices] MPGN на 3,
- Встановити [M/P Device ID] MGID в 1.

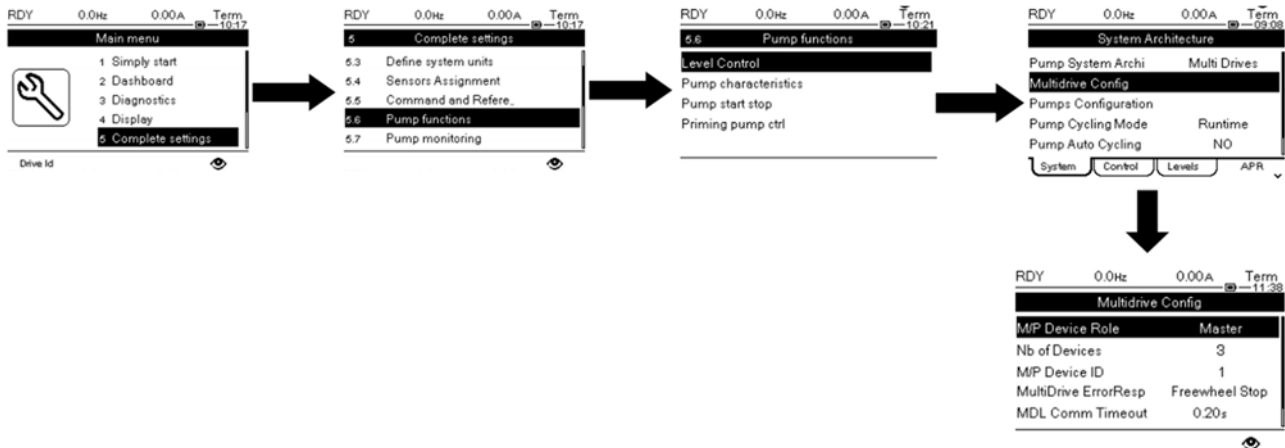


Рис. 5.35. Налаштування MultiDrive Link - MASTER

Налаштовуємо два ведених ПЧ (рис. 5.36).

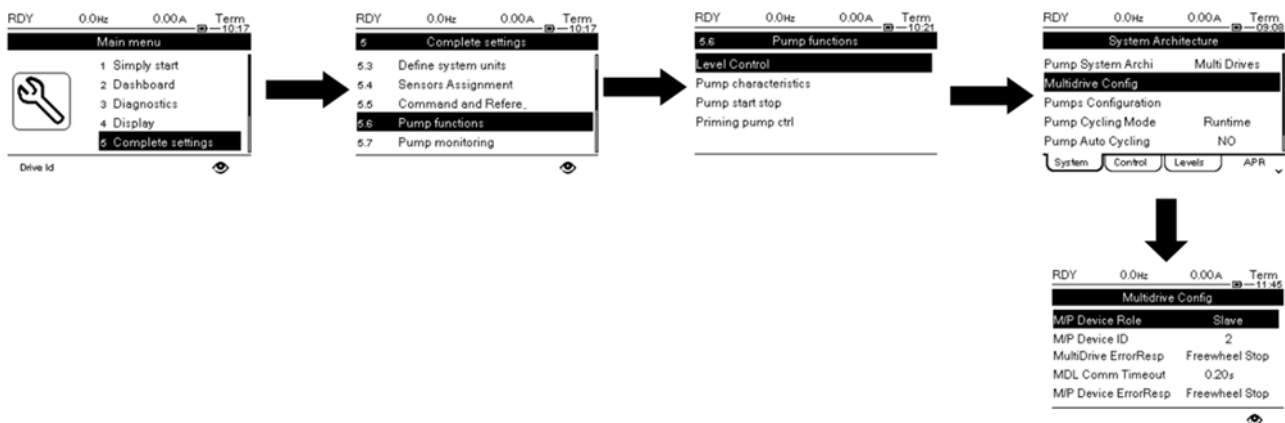


Рис. 5.36. Налаштування MultiDrive Link - SLAVES

На Altivar Process ATV600 можна вибрати два режими контролю рівня:

- наповнення;
- спорожнення.

За прикладом обираємо режим контролю рівня – спорожнення (рис. 5.37).

ПЧ Altivar Process ATV600 підтримує три типи датчиків для управління рівнем в резервуарі в системі контролю рівня:

- перемикачі рівня;
- датчик рівня;
- датчик тиску.

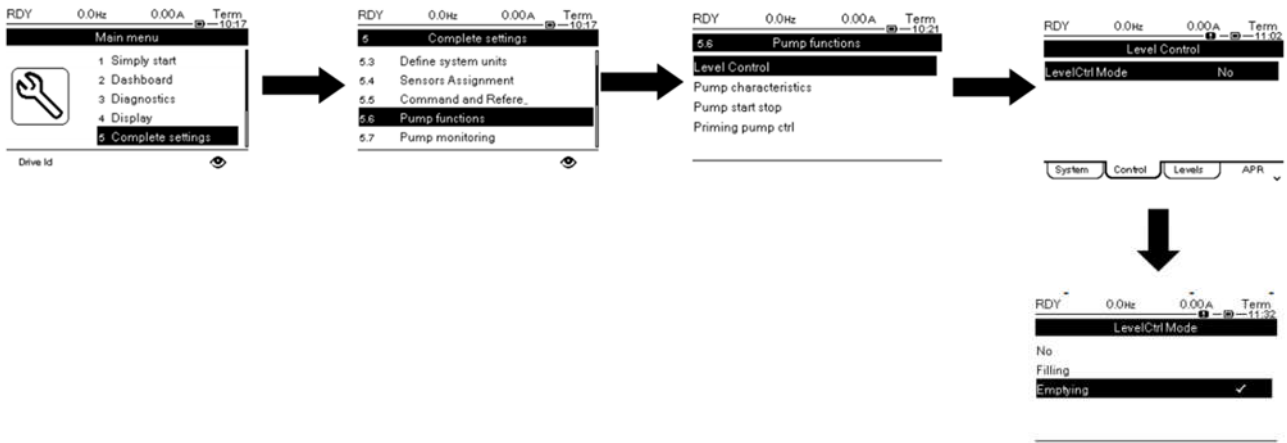


Рис. 5.37. Налаштування Level Control

За прикладом використовується датчик рівня з наступними характеристиками: 4...20 мА для 0..10 м (рис. 5.38).

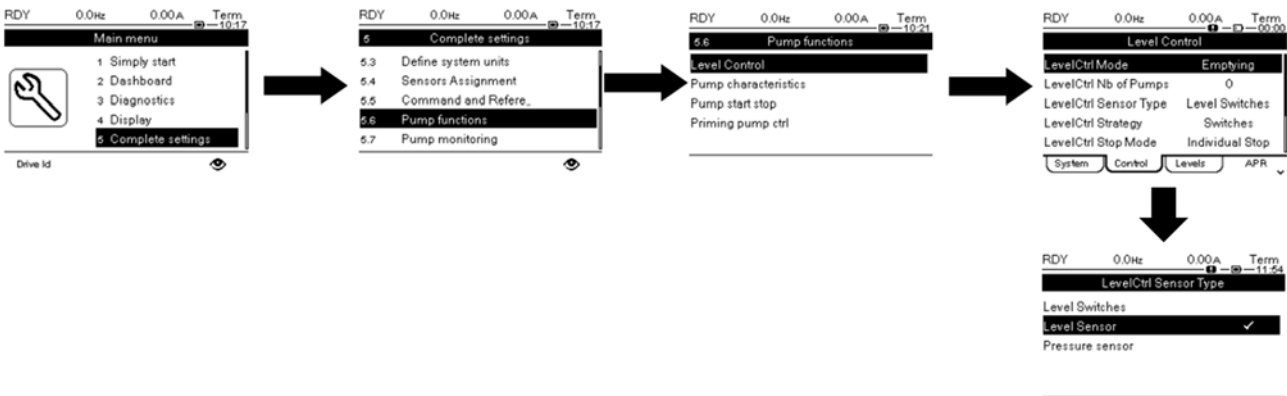


Рис. 5.38. Вибір типу датчика рівня

Конфігуруємо обраний датчик рівня (рис. 5.39). Найнижчі та найвищі введені значення процесу відповідають мінімальному та максимальним значенням датчика.

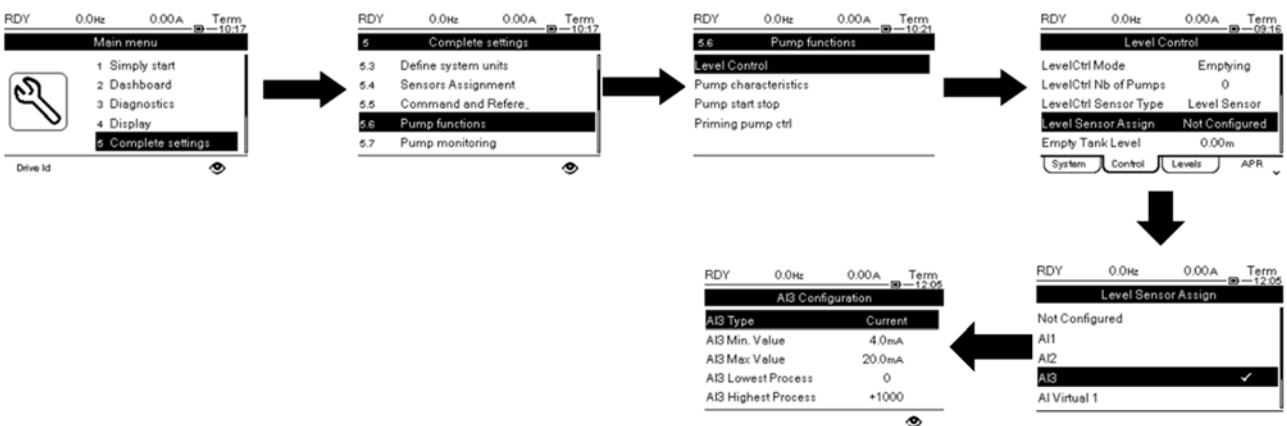


Рис. 5.39. Налаштування характеристик датчика рівня

Конфігурація висоти резервуара повинна бути налаштована таким чином, щоб встановити відповідність між конфігурацією датчика та реальною висотою резервуара. За прикладом висота резервуара становить 5 м, що нижче максимального значення датчика рівня, який використовується.

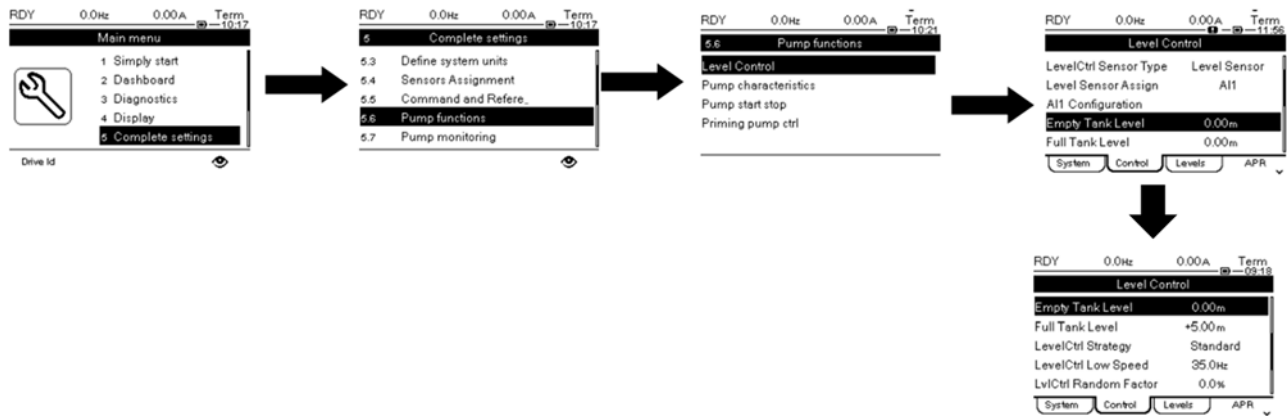


Рис. 5.40. Налаштування висоти резервуара

Для того, щоб використовувати функцію контролю рівня, необхідно налаштувати рівні запуску та зупинки кожного насосів, що використовуються в системі (за прикладом на рис. 5.41-42).

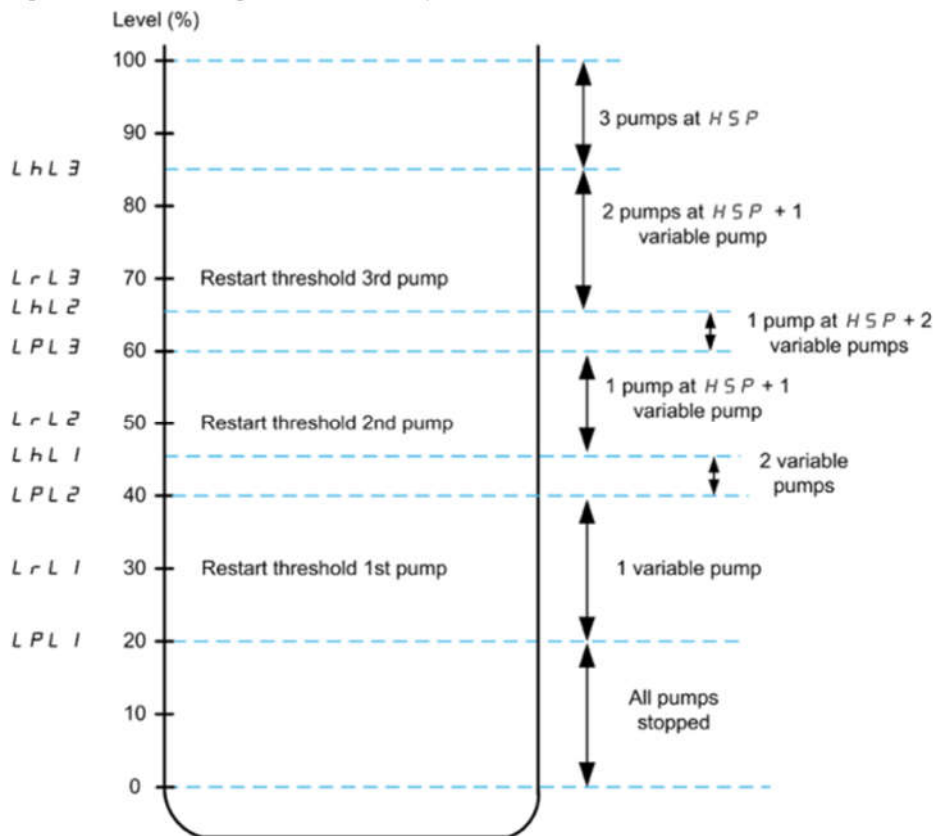


Рис. 5.41. Швидкість відповідно до рівня

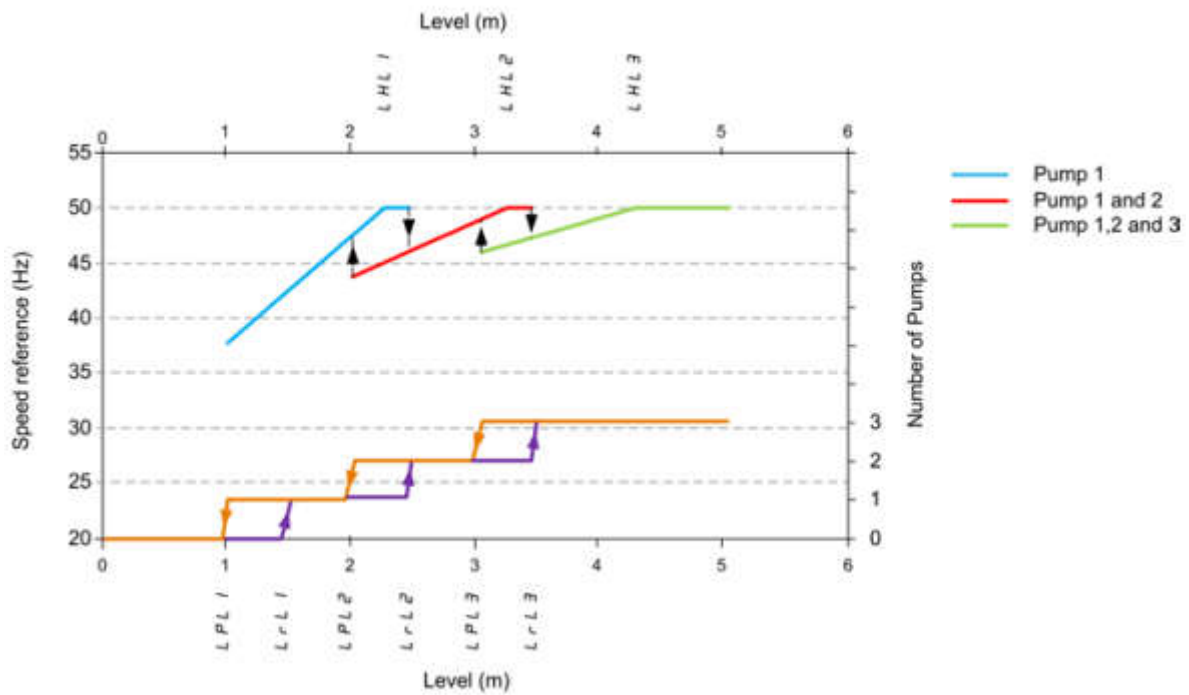


Рис. 5.42. Швидкість і кількість насосів відповідно до рівня

Налаштовуємо відповідні рівні спрацювання ПЧ (рис. 5.43).

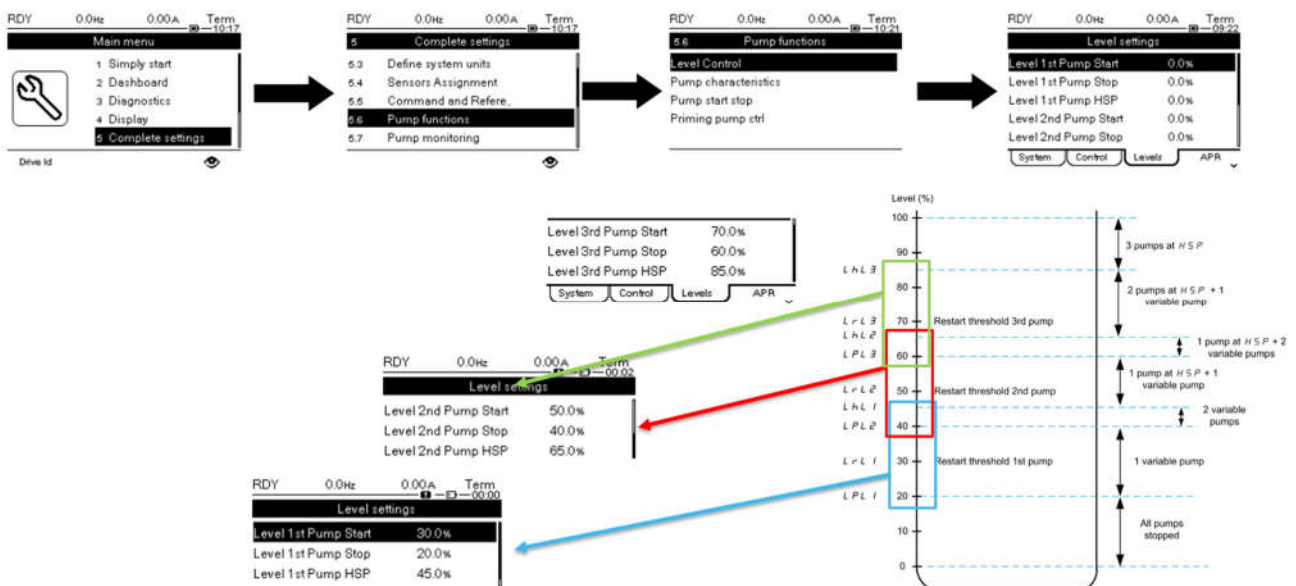


Рис. 5.43. Налаштування рівнів Пуску/Зупинки ПЧ

Перед запуском ПЧ необхідно перевірити конфігурацію параметрів у меню «Дисплей». В таблиці 5.1 наведено покрокову конфігурацію для перевірки налаштувань програми (рис. 5.44).

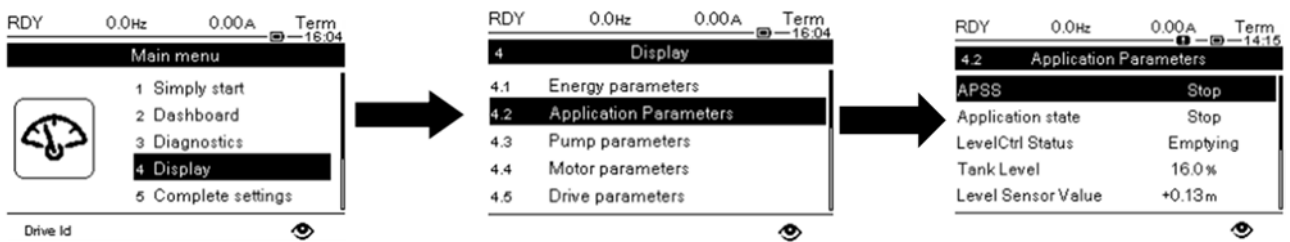


Рис. 5.44. Перевірка налаштувань програми

Таблиця 5.1

Порядок перевірки налаштувань

Налаштування	Необхідно перевірити
[System App State] APSS	Стан [Stop]. Немає команди на запуск на ведучому або ведених.
[Application state] APPS	Стан - [Stop]. На поточному пристрої немає команди на запуск.
[LevelCtrl Status] LCS	Стан узгоджується відповідно до рівня води в баку: [Emptying] : рівень між високим і низьким значенням [Low Level]: порожній бак [High Level]: повний бак
[Tank Level] LCTL	Задане значення повністю відповідає застосуванню.
[Level Sensor Value] LCSV	Значення зворотного зв'язку повністю відповідає застосуванню.

В таблиці 5.2 наведено покрокову конфігурацію для перевірки параметрів насоса (рис. 5.45).

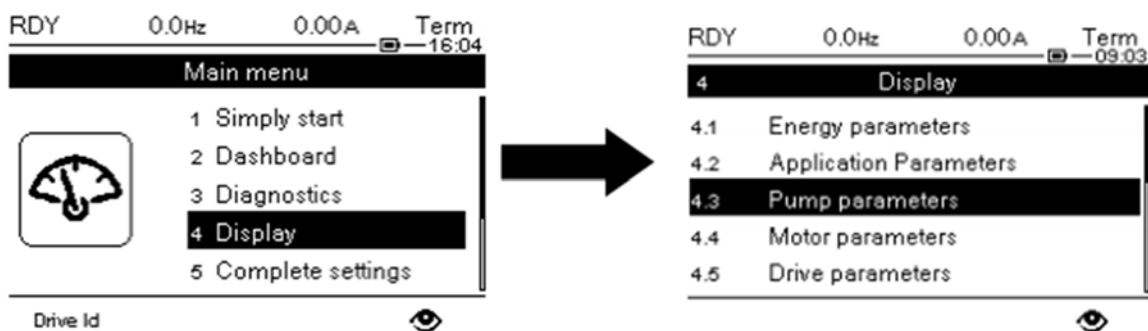


Рис. 5.45. Перевірки параметрів насоса

Для перевірки, що насоси запускаються і зупиняються на очікуваних рівнях необхідно виконати цикл, використовуючи меню [Application Parameters] APR і [Multipump System] MPS

Можна налаштувати реакцію на помилки ПЧ, які можна виявити на MultiDrive Link архітектурі MultiDriveLink.

Також можна налаштувати тайм-аут зв'язку MultiDrive Link відповідно до навантаження на мережу програми.

За замовчуванням для цих реакцій на помилки встановлена швидка зупинка, щоб уникнути ефекту гідроудару.

Таблиця 5.2

Порядок перевірки параметрів насоса

Налаштування	Необхідно перевірити
[MultiPump State] MPS	Стан [Ready]
[Available Pumps] MPAN	Кількість доступних насосів - 3
[Number of Staged Pump] MPSN	Немає жодного насоса зі ступінчастим режимом
[Pump 1 State] P1S	Стан [Ready].
[Pump 1 Type] P1T	Стан [Auxiliary Variable].
[Pump 2 State] P2S	Стан [Ready]
[Pump 2 Type] P2T	Стан [Auxiliary Variable].
[Pump 3 State] P3S	Стан [Ready]
[Pump 3 Type] P3T	Стан [Auxiliary Variable].

Відповідно до прикладу конфігурацію реакції на помилку змінено на зупинку вільного ходу, а таймаут встановлено на 0.20 с (рис. 5.46-47).

Помилка [MultiDrive Link Error] MDLF активна, якщо архітектура MultiDrive Link не є узгодженою (кілька ведучих, кілька ведених з однаковим ідентифікатором) під час виконання команди запуску.

Відповідь ПЧ на помилку [MultiDrive Link Error] MDLF задається параметром [MultiDrive ErrorResp] MDLB.

Помилка MPDF [M/P Device Error] може бути активною тільки на пристрої, який працює як ведений.

ПЧ на помилку [M/P Device Error] MPDF задається параметром [M/P Device ErrorResp] MPDB.

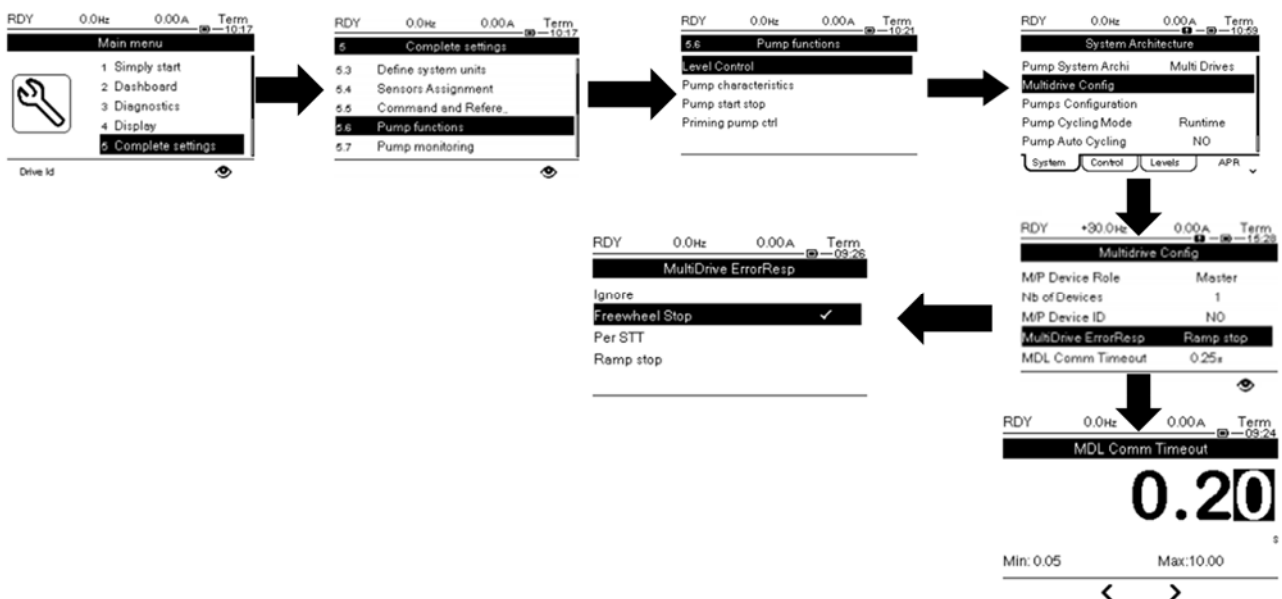


Рис. 5.46. Конфігурація обробки помилок та попереджень для Ведучого

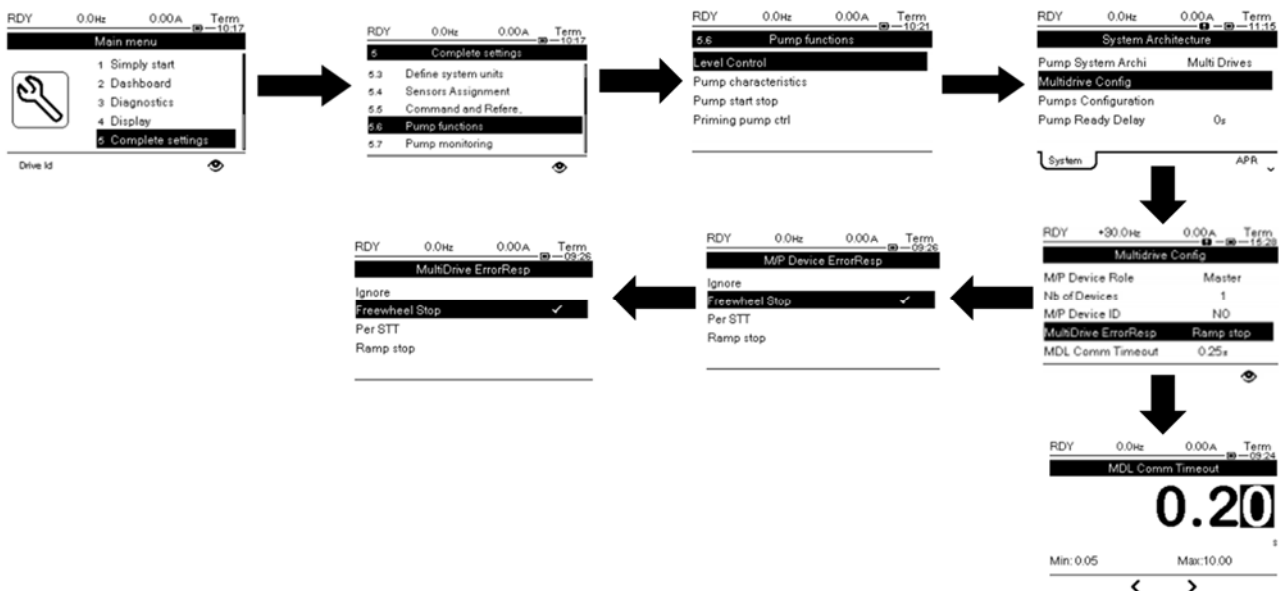


Рис. 5.47. Конфігурація обробки помилок та попереджень для Веденого

Для технічного обслуговування ПЧ можна локально заблокувати насос, активувавши цифровий вхід. Розглянемо налаштування цієї функції для цифрового входу DI3 (рис. 5.48).

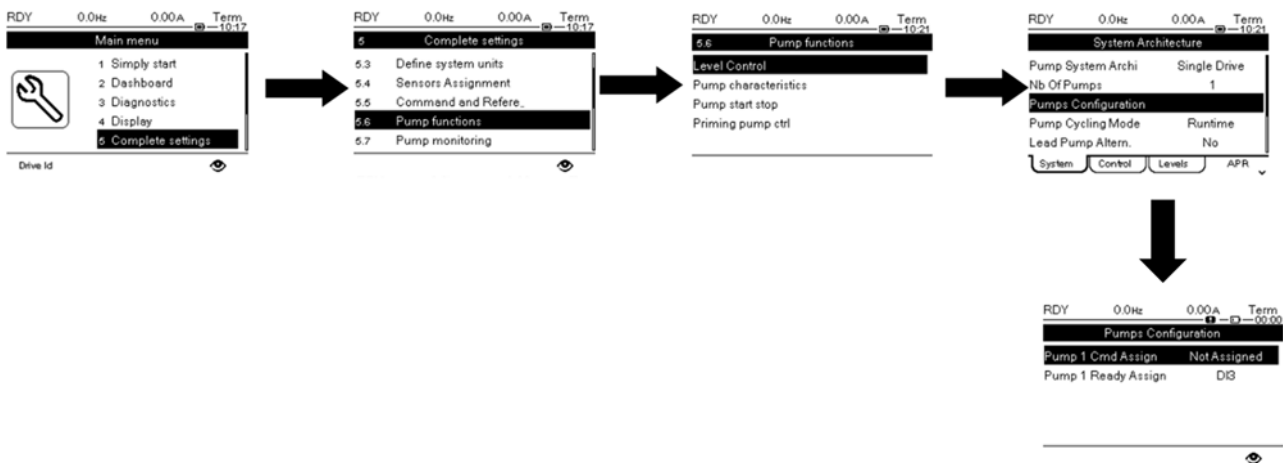


Рис. 5.49. Налаштування конфігурації блокування роботи ПЧ

Використаємо функцію яка дозволяє змінювати порядок запуску всіх доступних насосів, щоб керувати їхнім зносом. Згідно прикладу режим циклічності базується на часі їх роботи (рис. 5.50).

Доступний насос з найнижчим з найменшим часом роботи запускається першим, а насос з найбільшим часом роботи зупиняється першим.

Циклічність на основі порядку роботи насосів:

- [FIFO] Режим FIFO: насоси запускаються та зупиняються в порядку зростання;
- [LIFO] Режим LIFO: насоси запускаються в порядку зростання, а зупиняються в порядку спадання.

Циклічність на основі часу роботи:

- [Runtime] RTIME: доступний насос з найменшим часом роботи запускається першим, а працюючий насос з найменшим часом роботи запускається першим, а зупиняється насос з найбільшим часом роботи першим.
- [Runtime&LIFO] RTLF: циклічність на основі комбінації часу роботи і режиму LIFO. Доступний насос з найменшим часом роботи запускається першим, а насос, який запустився найпізніше, зупиняється першим.

Пункт меню недоступний, якщо для параметра [Pump System Archi] MPSA встановлено значення на [Multi Drives] NVSD.

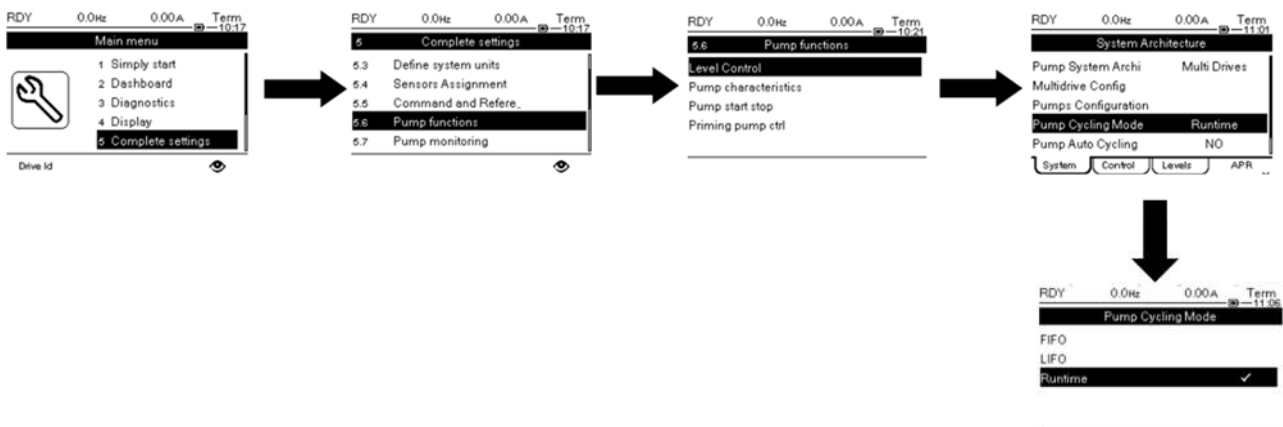


Рис. 5.50. Налаштування циклічності роботи насосів

В архітектурі MultiPump можна мати більше насосів, ніж потрібно. Така конфігурація необхідна, наприклад, якщо в архітектурі MultiDrive Link задекларовано X насосів але менше ніж X насосів можуть задовольнити потреби програми контролю рівня. Якщо кількість насосів, задекларованих у MultiDrive Link, дорівнює кількості насосів, що використовуються в додатку Level Control, цей параметр можна встановити на 0. Для нашого прикладу конфігурації максимальна кількість насосів, що використовуються для функції контролю рівня, встановлена дорівнює 3 (рис. 5.51).

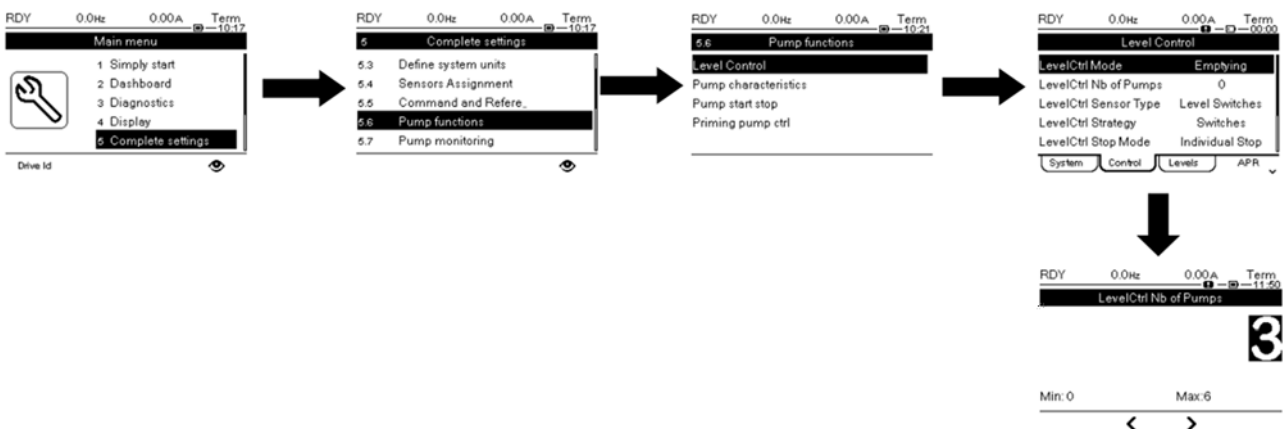


Рис. 5.51. Налаштування кількості насосів

Випадковий фактор Level Control дозволяє додавати випадковий рівень запуску та зупинки, щоб уникнути накопичення осаду в резервуарі. Цей параметр є відсотковим значенням від параметрів [Рівень запуску насоса 1-го рівня] LRL1 і [Рівень зупинки насоса 1-го рівня] LPL1. Для нашого прикладу конфігурації ми встановили випадковий коефіцієнт 5%. Під час кожного запуску та зупинки електропривод буде вибирати випадкове значення в межах цього діапазону.

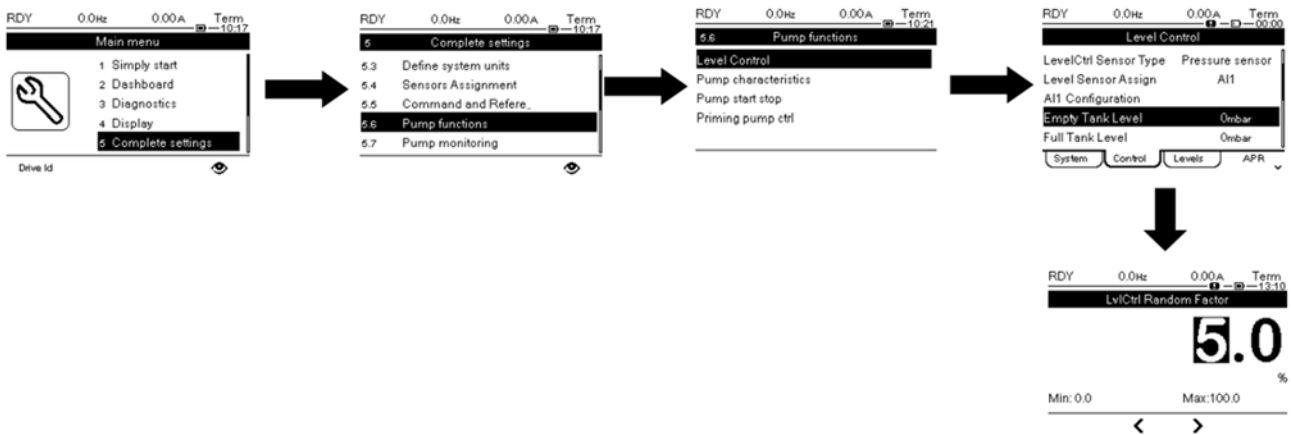


Рис. 5.52. Налаштування випадкового фактору

ПЧ Altivar Process ATV600 керують декількома попередженнями та помилками залежно від вибору режиму контролю рівня. Для нашого прикладу можливі такі попередження: [High Level Warning] LCHA: це попередження активне, якщо перемикач максимального рівня, визначений у параметрі [Max Level Switch Assign] (Призначення максимального перемикача рівня) LCWL активний. Можлива помилка: [Low Level Error] (Помилка низького рівня): [Low Level Error] LCLF : ця помилка активується, якщо перемикач мінімального рівня, визначений у параметрі [Min Level Switch Assign] LCWH активний. В заводській конфігурації реакція на помилку контролю рівня встановлена на зупинку темпу і може бути відрегульована залежно від потреб застосування. В цьому прикладі конфігурації перемикач мінімального рівня підключено до цифрового входу DI5, а перемикач максимального рівня перемикач максимального рівня підключено до цифрового входу DI6. Реакція на помилку змінюється на зупинку за вибігом (рис. 5.53).

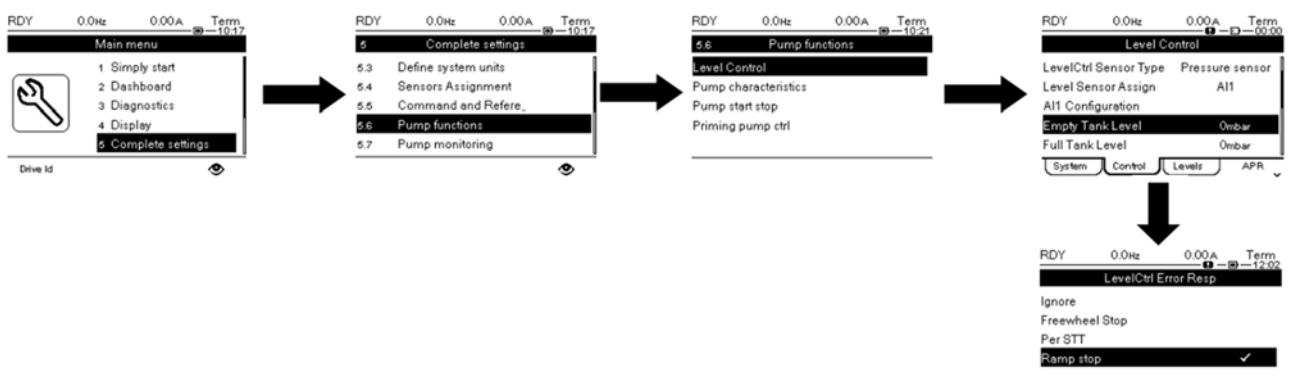


Рис. 5.53. Обробка помилок та попереджень

Висновки за розділом.

1. Розглянуто принципи роботи та функціональні можливості пристроїв плавного пуску, які забезпечують м'який розгін та зупинку електродвигунів. Зазначено, що застосування таких пристроїв дає змогу зменшити механічні навантаження, струмові удари та продовжити термін служби електроприводів і механічних вузлів.
2. Узагальнено апаратні характеристики та можливості ПЧ ATV320, які підтримують локальне програмування логіки керування (ATV Logic). Показано, що використання функціональних блоків і вбудованих алгоритмів дозволяє реалізувати автономні системи керування без додаткового ПЛК.
3. Визначено особливості сучасних промислових ПЧ серій ATV630 та ATV930, які відрізняються розширеними функціями моніторингу, енергозбереження та комунікації. Наголошено, що їх налаштування орієнтоване на керування складними технологічними процесами з високими вимогами до стабільності та ефективності.
4. Розглянуто приклади практичного застосування перетворювачів частоти у багатопривідних системах через MultiDrive Link, що забезпечує синхронізовану роботу кількох двигунів. Підкреслено, що сучасні ПЧ не лише виконують функцію регулювання швидкості, а й є інтелектуальними вузлами автоматизованої системи керування.

Питання для самоперевірки.

1. Яке основне призначення пристроїв плавного пуску в електроприводах?
2. У чому полягає принцип роботи пристрою плавного пуску, і які основні параметри підлягають налаштуванню?
3. Які переваги використання перетворювачів частоти порівняно з традиційними системами керування двигунами?
4. Які апаратні можливості має перетворювач частоти серії Altivar Machine 320?
5. Для чого використовується функція ATV Logic у перетворювачах частоти серії ATV320?
6. Які типові способи підключення і комунікації підтримує серія Altivar Process 630?
7. Які основні параметри налаштовуються при введенні в експлуатацію перетворювача ATV630?
8. Чим відрізняються апаратні та функціональні можливості перетворювачів серій ATV630 і ATV930?
9. Які функції захисту реалізовані в перетворювачах частоти Altivar Process 930?
10. Як реалізується взаємодія пристроїв Altivar із системами автоматизації на базі програмованих логічних контролерів?

РОЗДІЛ 6. КЕРУВАННЯ ПРОЦЕСАМИ, МЕРЕЖЕВІ СИСТЕМИ ТА SCADA-СИСТЕМИ

У цьому розділі розглядаються типи промислових процесів, які можуть керуватися за допомогою програмованих логічних контролерів (ПЛК). Одним із таких прикладів є системи SCADA. Для складних виробничих процесів застосовуються різні типи систем керування. Поряд із ПЛК це можуть бути роботизовані системи, термінали збору даних або комп'ютери. Щоб усі ці пристрої могли ефективно працювати разом, між ними має бути налагоджений обмін інформацією. У цьому розділі докладно розглядаються основні види промислових процесів і способи, за допомогою яких здійснюється взаємодія та комунікація між системами керування.

6.1. Типи процесів.

Під *керуванням* процесом розуміють автоматизоване управління перебігом технологічних операцій. Такі системи зазвичай працюють із аналоговими сигналами, що надходять від датчиків. Завдяки можливості виконання математичних обчислень і обробки аналогових даних ПЛК є особливо придатним для цього виду керування. Виробництво базується на послідовності процесів, які застосовуються до сировини. Типовими прикладами систем керування процесами є складання автомобілів, виробництво нафтохімічних продуктів, нафтопереробка, енергетика та харчова промисловість.

Безперервні процеси – це такі, у яких сировина безперервно надходить на один кінець системи, а готова продукція виходить на іншому. Процес відбувається постійно, без зупинок. Наприклад, на складальній лінії автомобільних двигунів деталі послідовно встановлюються на різних етапах виробництва – частково автоматизовано, частково вручну (рис. 6.1).

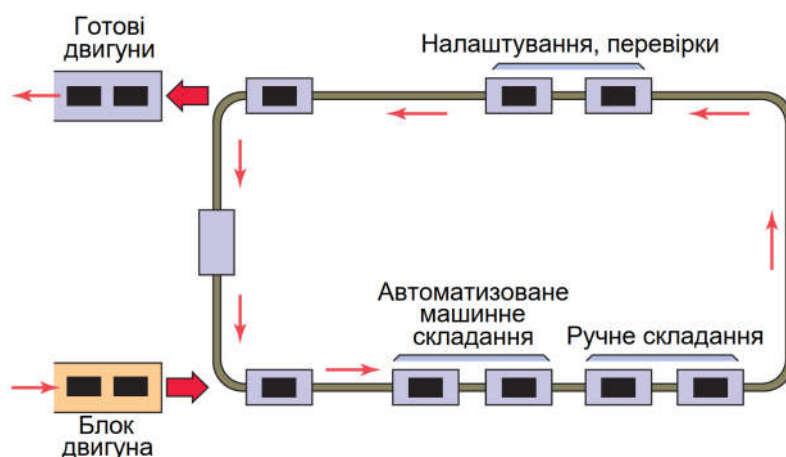


Рис. 6.1. Безперервні процеси

Пакетна обробка передбачає, що матеріали не переміщуються безперервно між ділянками виробництва. Натомість у процес подається певна кількість компонентів, які змішуються чи піддаються обробці, утворюючи завершену

партію продукту. Такий підхід характерний для виробництва харчових продуктів, напоїв, фармацевтичної продукції, фарб і добрив. Кожна партія виготовляється за певним «рецептом», який визначає її властивості (рис. 6.2).

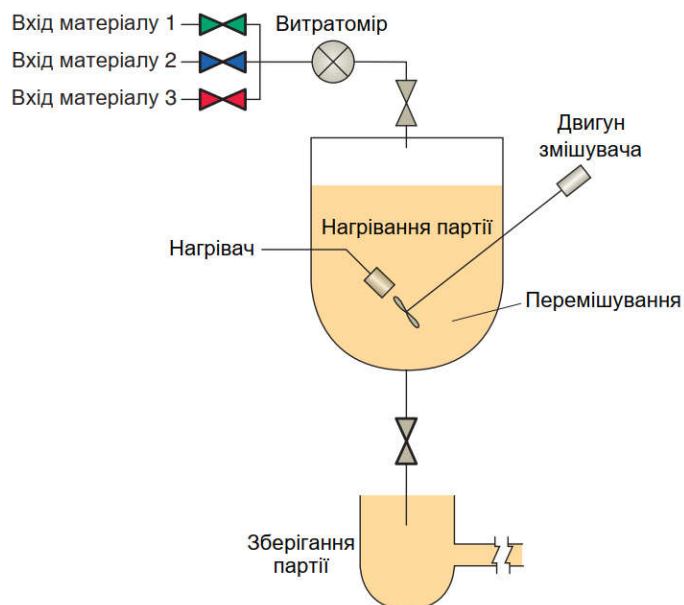


Рис. 6.2. Пакетна обробка

Дискретне виробництво – це процес, у якому продукція виготовляється у вигляді окремих, індивідуальних одиниць. Виробничі операції виконуються послідовно, поки не буде отримано готовий виріб. Такі системи зазвичай використовують цифрові входи ПЛК для керування двигунами та роботизованими пристроями. Кожна деталь обробляється окремо. Прикладом є виробництво елементів автомобільного інтер'єру, де кожна частина проходить власний цикл операцій (рис. 6.3).

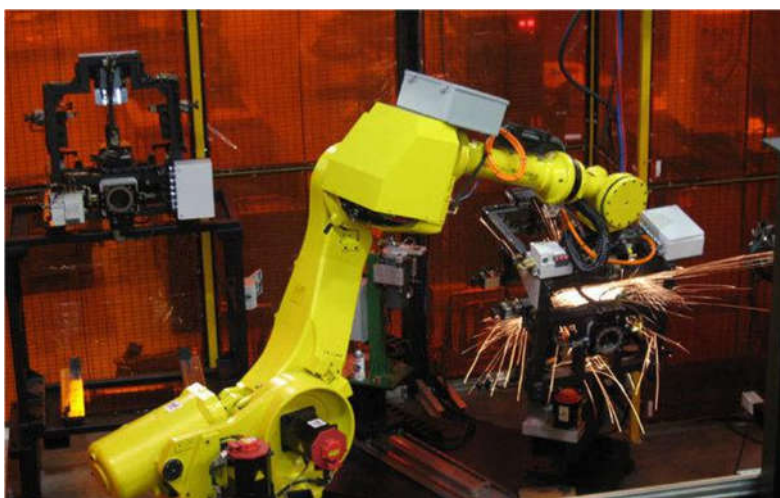


Рис. 6.3. Дискретне виробництво

Системи керування можуть мати *індивідуальну, централізовану* або *розподілену* конфігурацію.

Індивідуальне керування застосовується для роботи однієї машини, що не потребує зв'язку з іншими контролерами. Наприклад, у системі різання матеріалу на відрізки оператор задає довжину подачі й кількість відрізів через панель керування, після чого натискає кнопку запуску (рис. 6.4).



Рис. 6.4. Індивідуальне керування

Централізоване керування передбачає, що кількома машинами або процесами керує один центральний контролер. Така система охоплює всі етапи виробництва, але має певні недоліки: усі дії виконуються центральним контролером; інформація між іншими контролерами не обмінюється; вихід з ладу головного контролера призводить до зупинки всього процесу (рис. 6.5).

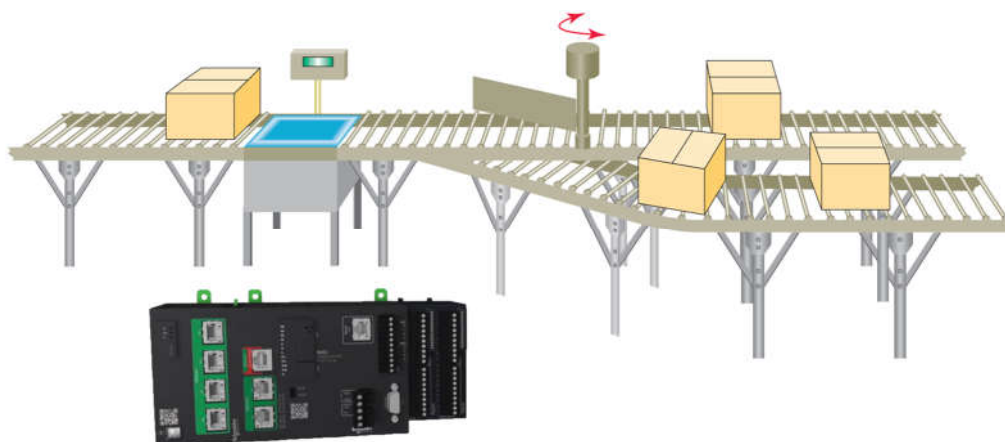


Рис. 6.5. Централізоване керування

Розподілена система керування (Distributive control system (DCS)) працює на основі мережевої взаємодії кількох ПЛК. Кожен контролер керує певним локальним процесом, водночас усі вони постійно обмінюються даними через комунікаційні лінії – це можуть бути кабелі CAT-5 або CAT-6, коаксіальні лінії, волоконно-оптичні з'єднання чи мережа Ethernet. Така архітектура дозволяє розподілити обчислювальні завдання між кількома контролерами, зменшити

кількість польових проводів і підвищити ефективність, оскільки контролер розташований безпосередньо біля об'єкта керування. У разі відмови одного з контролерів система зазвичай продовжує роботу. Над роботою розподіленої системи керування зазвичай здійснює нагляд головний комп'ютер, який виконує функції моніторингу, зберігання даних і формування звітів (рис. 6.6).

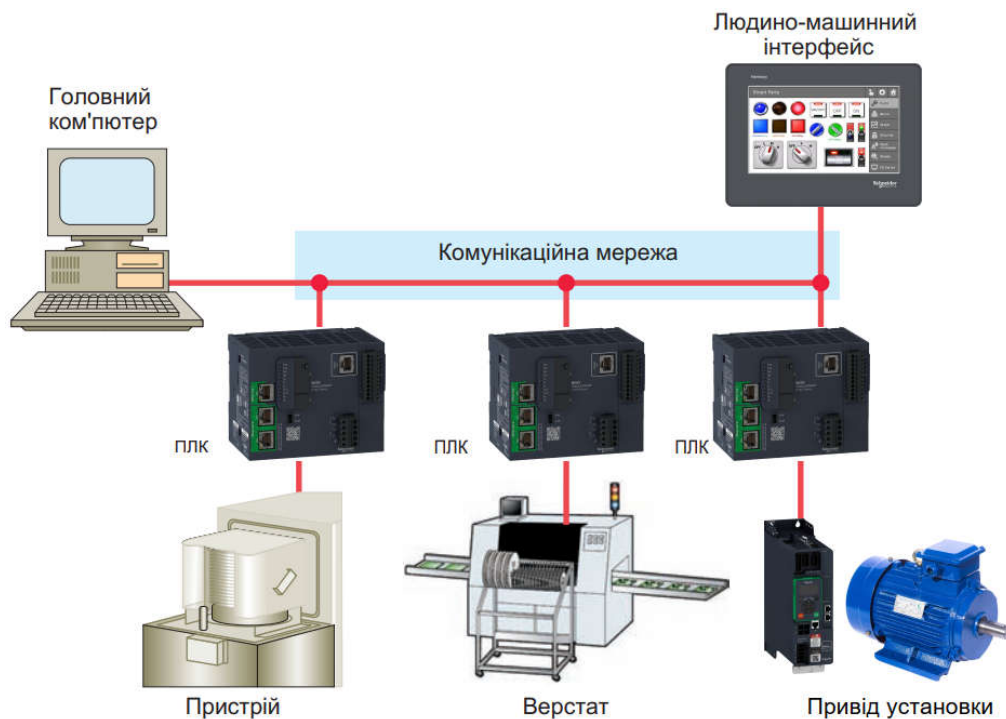


Рис. 6.6. Розподілена система керування (DCS)

6.2. Структура систем керування.

Керування процесами найчастіше застосовується у промисловому виробництві та переробці різних продуктів. У разі використання програмованого логічного контролера робота машини або технологічного процесу відбувається під керуванням користувацької програми, яка контролює всі етапи операцій.

Основними складовими системи керування процесом є:

Датчики – отримують сигнали з технологічного процесу або навколишнього середовища, перетворюючи фізичні параметри, такі як тиск, температура, витрата чи положення, в електричні сигнали.

Людино-машинний інтерфейс або графічний термінал (Human machine interface (HMI)) – забезпечує взаємодію оператора з системою через перемикачі, кнопки, сенсорні панелі чи клавіатури. За допомогою людино-машинного інтерфейсу оператор може задавати початкові умови, змінювати параметри або контролювати хід процесу (рис. 6.7).

Узгодження сигналів (Signal Conditioning) – процес перетворення сигналів у форму, придатну для обробки контролером. Це може включати підсилення, ослаблення, фільтрацію, масштабування та перетворення аналогових сигналів у цифрові (A/D) або навпаки (D/A).

Виконавчі пристрої – перетворюють електричні сигнали керування в механічну дію. До них належать клапани регулювання потоку, насоси, приводи

позиціонування, варіатори швидкості, зчеплення, гальма, соленоїди, крокові двигуни та потужні реле.

Контролер – основний елемент системи, який на основі вхідних сигналів приймає рішення та формує вихідні сигнали для керування виконавчими механізмами.



Рис. 6.7. Людино-машинний інтерфейс або графічний термінал

Інтерфейс НМІ виконує роль засобу керування та візуалізації, що поєднує оператора з технологічним процесом. Через нього оператор може запускати й зупиняти систему, змінювати режими роботи, стежити за параметрами та вчасно реагувати на відхилення чи аварійні ситуації.

Сучасні графічні термінали НМІ замінили традиційні панелі з фізичними перемикачами. Вони оснащені сенсорними екранами з візуальними елементами керування – кнопками, індикаторами, графічними діаграмами. Такі екрани можуть відображати:

- операційний огляд, що показує загальний стан процесу;
- екрани налаштування, де вводяться параметри роботи;
- журнал тривоги, який містить активні сигнали з мітками часу;
- історію подій, що фіксує послідовність дій системи;
- графіки трендів, які відображають зміну параметрів (наприклад, температури чи витрати) у часі;
- режим ручного керування, доступний технічному персоналу для обслуговування;
- діагностичні вікна, що допомагають виявляти несправності обладнання.

Комплекти НМІ зазвичай постачаються разом із необхідним апаратним, програмним забезпеченням і засобами комунікації (рис. 6.8). Під час створення НМІ-застосунку потрібно налагодити зв'язок із ПЛК, створити базу даних змінних (тегів), спроектувати графічні елементи, додати анімацію та логіку взаємодії.



Рис. 6.8. Графічні панелі (термінали)

Більшість сучасних систем керування є замкненими контурами – тобто працюють за принципом зворотного зв'язку. У таких системах вихід процесу впливає на вхідний сигнал керування. Контролер безперервно порівнює фактичний результат із заданим значенням (Set-Point) і автоматично коригує дію доти, доки відхилення не стане мінімальним або не зникне зовсім (рис. 6.9).

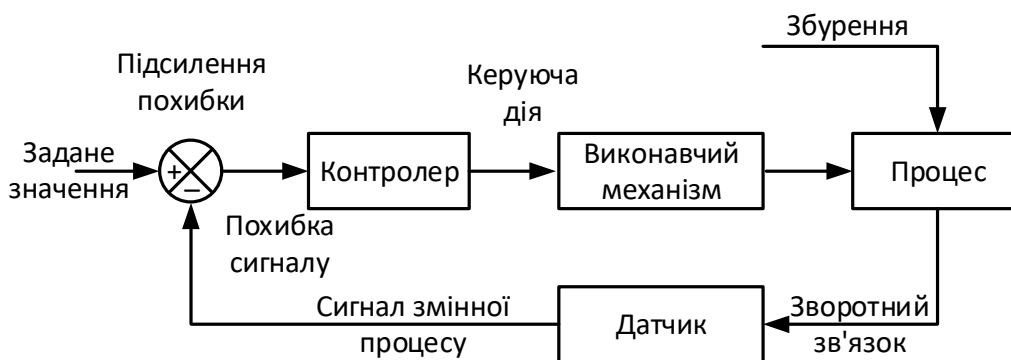


Рис. 6.9. Замкнутий контур системи керування

У замкненому контурі задане значення визначає бажаний режим роботи, процесна змінна – це сигнал зворотного зв'язку, який показує фактичний стан процесу, підсилювач помилки обчислює різницю між цими сигналами, а контролер формує коригуючий вихідний сигнал для виконавчого механізму наприклад, двигуна, нагрівача, вентилятора чи соленоїда.

Класичним прикладом замкненої системи є процес автоматичного наповнення коробок продуктом до заданої ваги (рис. 6.10). Порожня коробка подається під дозатор, починається наповнення, і датчик ваги постійно передає сигнал до контролера. Коли фактична вага досягає встановленої норми, контролер подає команду на закриття клапана.

Таким чином, усі системи зі зворотним зв'язком працюють за єдиним принципом – контролер постійно відстежує різницю між бажаним і фактичним станом процесу та автоматично усуває відхилення, забезпечуючи стабільність і точність керування (рис. 6.11).

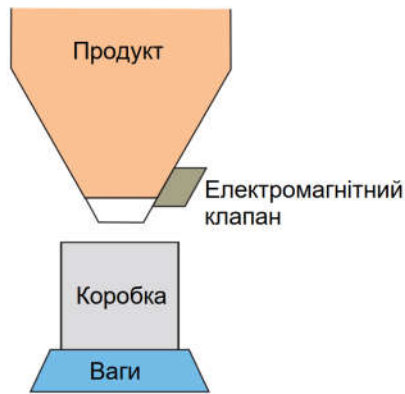


Рис. 6.10. Автоматична система наповнення контейнера

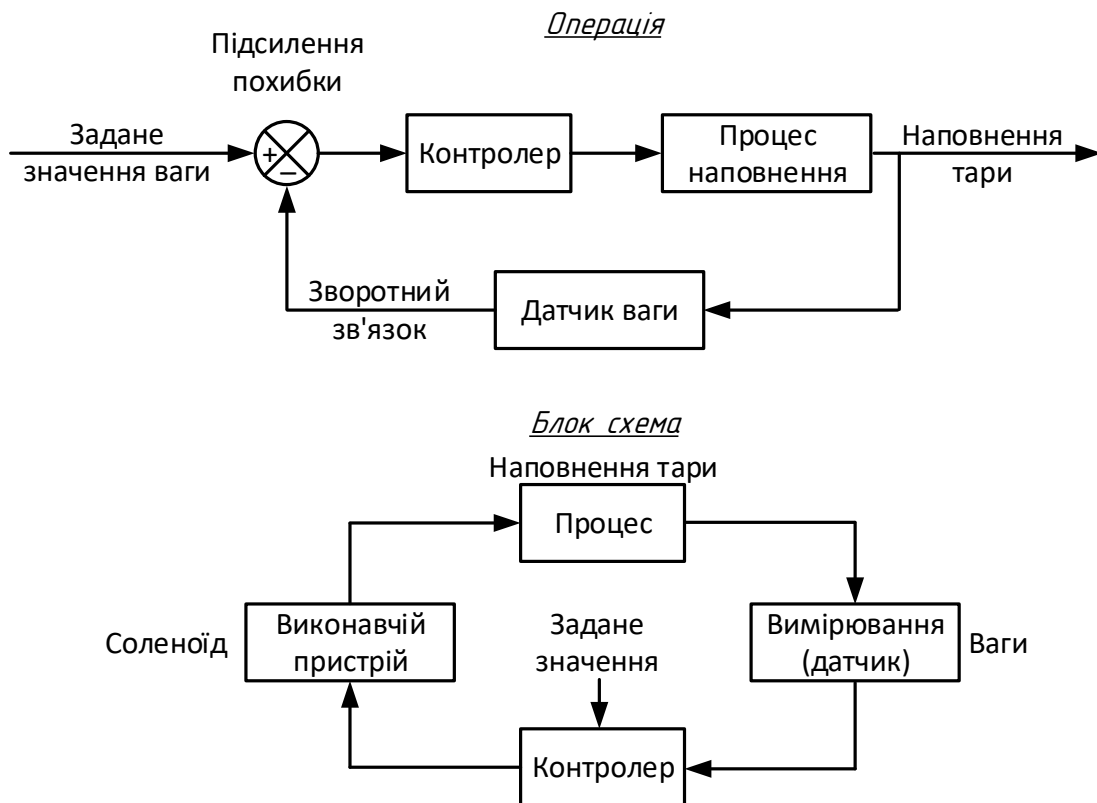


Рис. 6.11. Схеми системи керування та операцій процесу наповнення контейнера

6.3. Керування за принципом «Вмикання/Вимикання».

У системах керування типу «Увімкнено/Вимкнено» (*двопозиційне*) (On/Off Control) кінцевий елемент керування може перебувати лише у двох станах: увімкнений або вимкнений. Це означає, що коли вимірюване значення процесної змінної (наприклад, температури) падає нижче заданої точки (Set-Point), контролер подає команду на увімкнення виконавчого пристрою. Коли ж значення перевищує заданий рівень, контролер вимикає цей пристрій. У проміжному положенні керуючий елемент ніколи не залишається, регулювання відбувається лише за рахунок періодичного циклу «Вмикання/Вимикання».

Розглянемо приклад системи, у якій рідина нагрівається за допомогою пари. Якщо температура рідини падає нижче встановленої межі, паровий клапан відкривається, і подача пари починається (рис. 6.12). Коли температура

перевищує встановлену точку, клапан закривається, і нагрів припиняється. Цей цикл «Вмикання/Вимикання» повторюється доти, доки система працює.

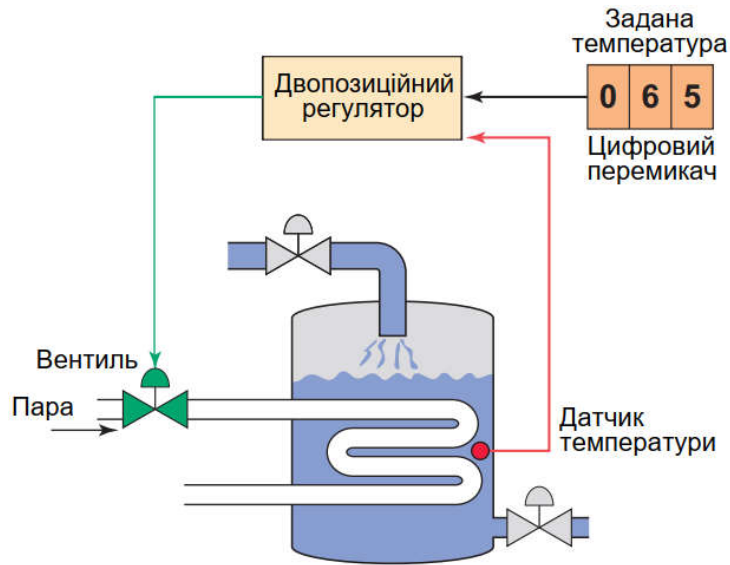


Рис. 6.12. Схема керування за принципом «Вмикання/Вимикання» в системі підігріву рідини

Графік реакції такої системи показує, що вихідний сигнал вмикається, коли температура падає нижче встановленого значення, і вимикається, щойно вона його досягає або перевищує (рис. 6.13). Такий тип керування є простим у реалізації, однак має певні недоліки. Через інерційність процесу температура (або інший параметр) часто перевищує або не досягає заданого значення, утворюючи коливання навколо точки налаштування. Частота й амплітуда цих коливань залежать від об'єму системи та швидкості її реакції.

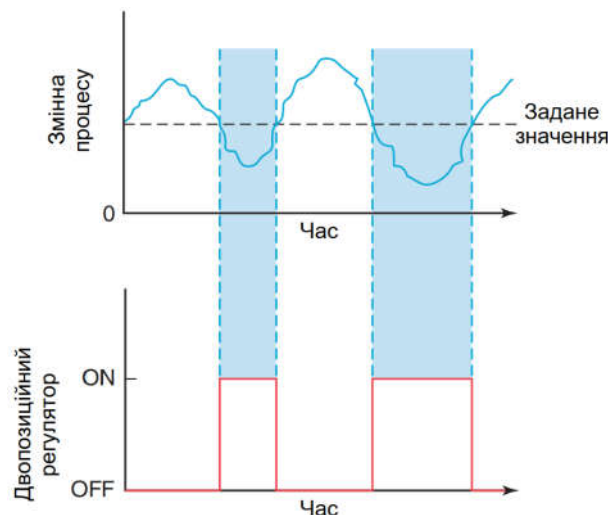


Рис. 6.13. Графік реакції системи керування за принципом «Вмикання/Вимикання»

Зменшити коливання можна, підвищивши чутливість контролера, але тоді пристрій буде частіше перемикатися між станами, що може призвести до передчасного зношування обладнання або небажаних коливань у системі. Тому

керування типу «Вмикання/Вимикання» зазвичай застосовується там, де висока точність не потрібна, наприклад у побутових термостатах або простих нагрівальних системах.

Щоб уникнути постійного «полювання» системи навколо точки налаштування, у контролерах передбачено так звану зону нечутливості (Deadband). Вона задає допустиме відхилення від заданого значення – невеликий діапазон вище та нижче точки налаштування, у межах якого контролер не змінює свій стан. Завдяки цьому дрібні коливання сигналу не призводять до безперервного вмикання та вимикання, що забезпечує більш стабільну роботу системи.

6.4. Керування типу ПІД.

Системи пропорційно-інтегрально-диференційного (ПІД) керування становлять основу сучасної автоматизації технологічних процесів. Їхня поява стала логічним розвитком після двопозиційного та суто пропорційного керування, оскільки саме ПІД-алгоритм дає змогу забезпечити високу точність, стабільність і гнучкість регулювання в умовах змінних навантажень і зовнішніх збурень.

Пропорційні контролери були створені для усунення коливань і нестабільності, характерних для систем із двопозиційним («Вмикання/Вимикання») керуванням. Вони дозволяють виконавчому механізму займати проміжні положення між повністю «увімкненим» і «вимкненим» станом, забезпечуючи плавне регулювання подачі енергії в процес залежно від ступеня відхилення вимірюваної величини від заданого значення.

На відміну від двопозиційних систем, у пропорційному керуванні вихідний сигнал може набувати будь-якого значення в межах від мінімального до максимального, що забезпечує точніше підтримання параметра процесу. Наприклад, у системі керування клапаном, привід отримує аналоговий струмовий сигнал у діапазоні від 4...20 мА: 4 мА відповідає повністю закритому положенню клапана, а 20 мА максимально відкритому (рис. 6.14). Такий тип сигналу є стійким до перешкод і змін опору з'єднувальних проводів, що підвищує надійність системи.

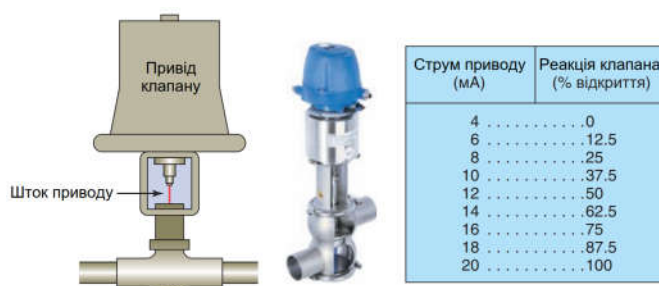


Рис. 6.14. Аналоговий мотор-привод з пропорційним керуванням вентилля

Існує також часовий варіант пропорційного керування – широко-імпульсна модуляція (ШІМ) (Pulse Width Modulation, PWM). У цьому випадку елемент

вмикається та вимикається з високою частотою, а середня тривалість увімкненого стану визначає ефективну потужність. Так, якщо нагрівальний елемент потужністю 200 Вт працює половину часу, фактична потужність становитиме 100 Вт, а при 25% часу лише 50 Вт (рис. 6.15).

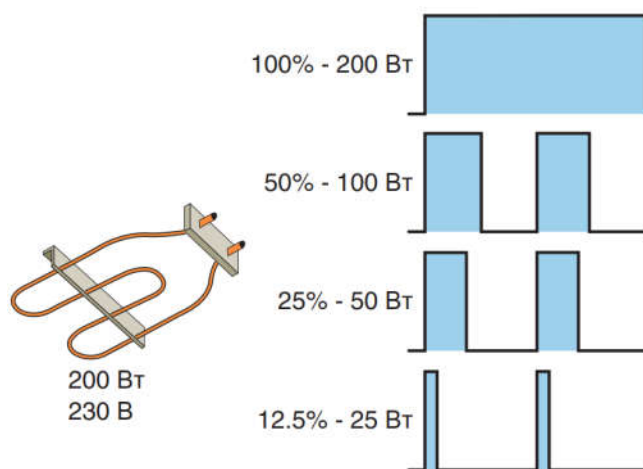


Рис. 6.15. Пропорційне керування нагрівальним елементом з використанням ШІМ

У пропорційних системах існує так звана пропорційна зона – діапазон температур чи інших параметрів, навколо точки налаштування, у межах якого контролер змінює свій вихід поступово. Якщо температура нижча за межі зони, вихід буде повністю активний; якщо вища повністю вимкнений. У самій зоні регулятор змінює сигнал плавно, зменшуючи амплітуду коливань (табл. 6.1).

Таблиця 6.1

Пропорційна зона для системи нагріву

Пропорційна за часом			Температура, °С	Пропорційна 4-20 мА	
Проценти	Час увімкнення	Час вимкнення		Рівень виходу	Процент виходу
0.0	0.0	20.0	більше 280	4 мА	0.0
0.0	0.0	20.0	280	4 мА	0.0
12.5	2.5	17.5	275	6 мА	12.5
25.0	5.0	15.0	270	8 мА	25.0
37.5	7.5	12.5	265	10 мА	37.5
50.0	10.0	10.0	260.0	12 мА	50.0
62.5	12.5	7.5	255.0	14 мА	62.5
75.0	15.0	5.0	250.0	16 мА	75.0
87.5	17.5	2.5	245.0	18 мА	87.5
100.0	20.0	0.0	240.0	20 мА	100.0
100.0	20.0	0.0	менше 240	20 мА	100.0

Попри точність, пропорційні контролери мають недолік – сталу похибку. Це означає, що реальне значення параметра стабілізується не точно на заданому рівні, а дещо нижче або вище нього (рис. 6.16). Для усунення цього ефекту до пропорційної дії додають інтегральну (І) та диференційну (Д) складові, які компенсують недоліки одна одної.

Інтегральна дія усуває сталу похибку, накопичуючи інформацію про розмір і тривалість відхилення. Якщо різниця між заданим і фактичним значенням не зникає, інтегральна складова поступово збільшує вихідний сигнал, доки система не досягне рівноваги. Її вплив вимірюється у хвилинах на повторення (Minutes Per Repeat) або повторях за хвилину (Repeats Per Minute), що характеризує швидкість реакції контролера на похибку в часі.

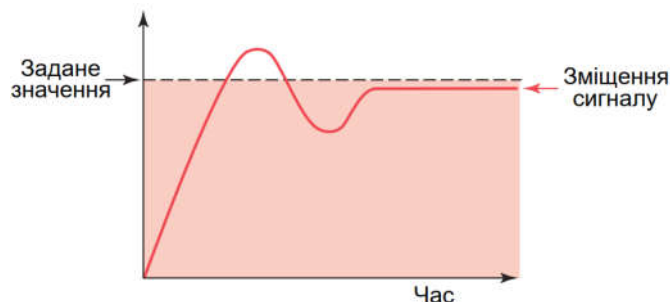


Рис. 6.16. Стала похибка пропорційного регулювання

Диференційна дія реагує не на величину, а на швидкість зміни похибки. Вона виконує роль динамічного демпфера, «гальмуючи» надто швидкі зміни вихідного сигналу. Це дозволяє запобігати перевищенню заданого значення, стабілізуючи процес у моменти різких збурень.

Об'єднання трьох типів дії – пропорційної, інтегральної та диференційної формує *ПІД-контур керування*, який забезпечує найвищу якість регулювання (рис. 6.17). Пропорційна частина забезпечує плавність, інтегральна усуває систематичну похибку, а диференційна забезпечує швидку реакцію на перехідні процеси.

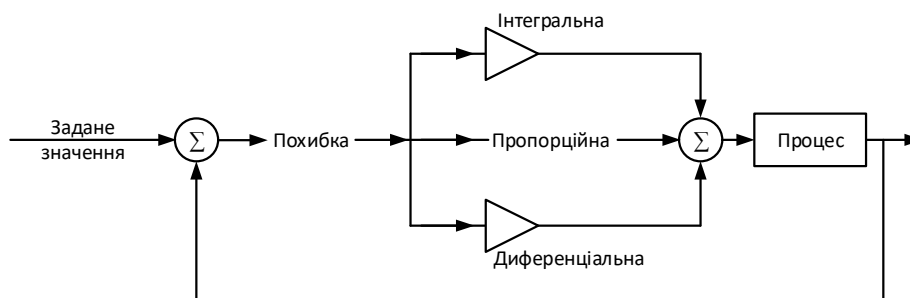


Рис. 6.17. Замкнутий контур керування з ПІД-регулятором

Під час налаштування контролера задаються такі параметри, як точка регулювання, ширина пропорційної зони, час інтегрування, час диференціювання та межі вихідного сигналу. Ці значення можуть змінюватися в процесі роботи системи для досягнення оптимальної динаміки.

Наприклад, якщо ПІД-контролер використовується для підтримання температури печі на рівні 50 °С, він автоматично збільшує потужність нагріву, коли температура знижується, і поступово зменшує її, коли параметр наближається до встановленого значення. Це дозволяє уникнути різких коливань і досягти стабільного теплового режиму.

Під час налаштування ПІД-керування задаються параметри заданого значення, ширини пропорційної зони, часу інтеграції, часу диференціювання та

меж вихідного сигналу. Ці значення можуть змінюватися під час роботи системи для точнішого «підстроювання» процесу. Наприклад, у випадку регулювання температури печі: якщо температура падає нижче встановленої, потужність нагріву поступово збільшується; коли температура наближається до заданої, контролер плавно знижує подачу енергії, запобігаючи перегріву.

Основна мета будь-якої системи досягти балансу між подачею та відведенням енергії або маси. Оскільки умови процесу постійно змінюються, важливим параметром є час реакції системи. ПІД-керування враховує як величину похибки, так і швидкість її зміни, що дозволяє адаптувати його до процесів із різною динамікою від дуже швидких до повільних.

Сучасні ПЛК можуть виконувати ПІД-регулювання за допомогою спеціальних модулів введення/виведення або математичних функцій у програмі. Принцип дії полягає у безперервному вимірюванні контрольованого параметра, порівнянні його із заданим значенням, обчисленні похибки та формуванні вихідного сигналу, який коригує процес (рис. 6.18).

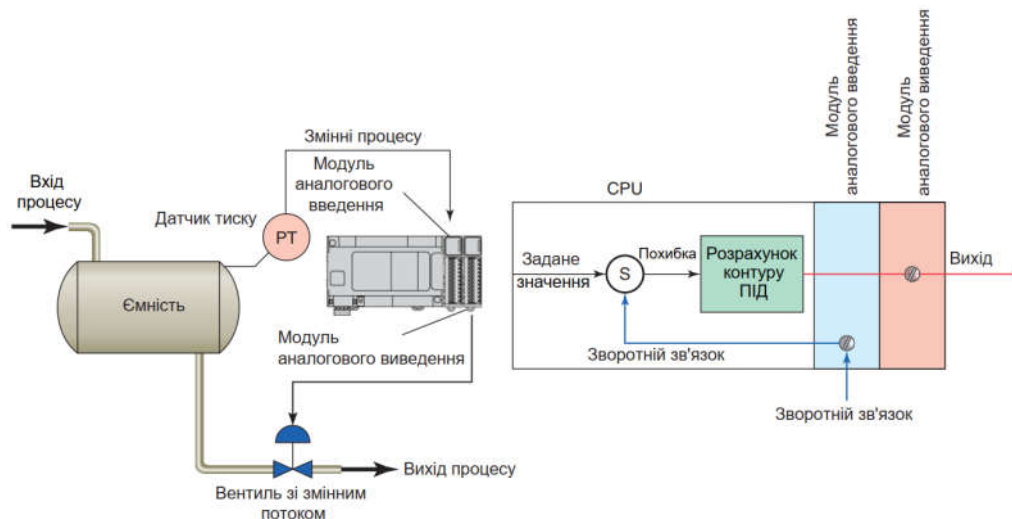


Рис. 6.18. Система автоматичного керування з використанням ПЛК з ПІД-регулятором

Точність і стабільність системи залежать від правильного налаштування трьох коефіцієнтів: пропорційного, інтегрального та диференційного. Якщо збільшити пропорційний коефіцієнт, система реагуватиме швидше, але може стати нестабільною. Інтегральна складова стабілізує процес і підтримує необхідний рівень потужності навіть без поточної похибки. Диференційна складова, навпаки, уповільнює реакцію при занадто швидких змінах, запобігаючи коливанням.

Налаштування ПІД може виконуватися вручну (підбором параметрів оператором методом спроб і помилок), напівавтоматично (режим AutoTune, коли контролер сам розраховує параметри) або повністю автоматично за допомогою нечіткого керування (Fuzzy Logic). У цьому випадку алгоритм не використовує жорстку математичну формулу, а оцінює ситуацію за допомогою набору правил типу «якщо-то». Помилки перетворюються на «нечіткі» лінгвістичні змінні (наприклад, «помилка велика негативна» або «зміна мала позитивна»), а

результат потім «дефазифікується» у конкретне значення наприклад, ступінь відкриття клапана.

Таким чином, ПД-керування є універсальним інструментом сучасної автоматизації. Завдяки поєднанню трьох регулювальних дій він забезпечує стабільність, точність і швидку реакцію системи, що робить його незамінним у керуванні температурою, тиском, рівнем рідини, витратою, швидкістю та багатьма іншими параметрами промислових процесів.

6.5. Керування рухом.

Система керування рухом забезпечує точне регулювання положення, швидкості та крутного моменту в широкому діапазоні застосувань, пов'язаних із переміщенням механізмів. Такі системи особливо важливі там, де необхідна висока точність і синхронізація рухів, як у лінійних, так і в обертальних процесах. ПЛК добре підходять для реалізації керування рухом, оскільки здатні ефективно працювати як із лінійними, так і з ротаційними типами переміщення.

Одним із поширених прикладів застосування систем керування рухом є машини типу «Взяти та перемістити» (Pick and Place), що широко використовуються у виробництві споживчих товарів. Такі пристрої виконують завдання з перенесення продукції з однієї точки в іншу. Типова ситуація коли виріб автоматично переміщується до рухомого конвеєра, де він потрапляє до наступного етапу технологічного процесу (рис. 6.19).

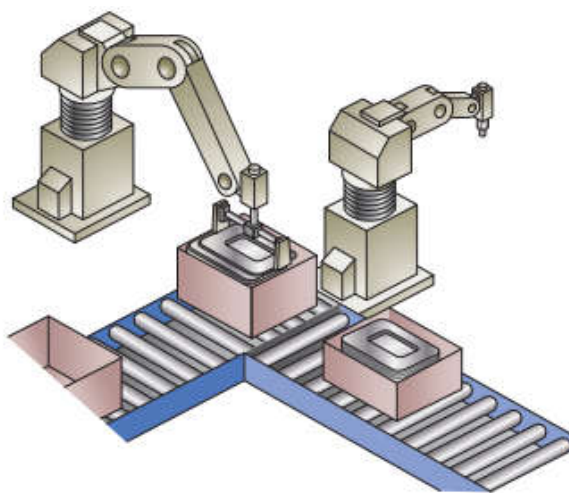


Рис. 6.19. Роботи типу «Взяти та перемістити»

Основна структура системи керування рухом на базі ПЛК включає контролер, модуль керування рухом, сервопривід, один або кілька двигунів з енкодерами та механічне обладнання, яким здійснюється управління. Кожен двигун, що бере участь у системі, визначається як окрема вісь руху. Наприклад, у процесі наповнення пляшок рідиною система має дві осі руху: одна відповідає за механізм наповнення, а інша за регулювання швидкості конвеєрної стрічки (рис. 6.20).

Контролер у такій системі є головною складовою, що зберігає та виконує користувацьку програму, яка визначає логіку всього процесу. У програмі

містяться інструкції керування рухом, які задають переміщення осей. Коли контролер стикається з командою руху, він виконує розрахунки, формуючи необхідні команди для осей, що визначають положення, швидкість або момент, яких має досягти сервомотор у конкретний момент часу.

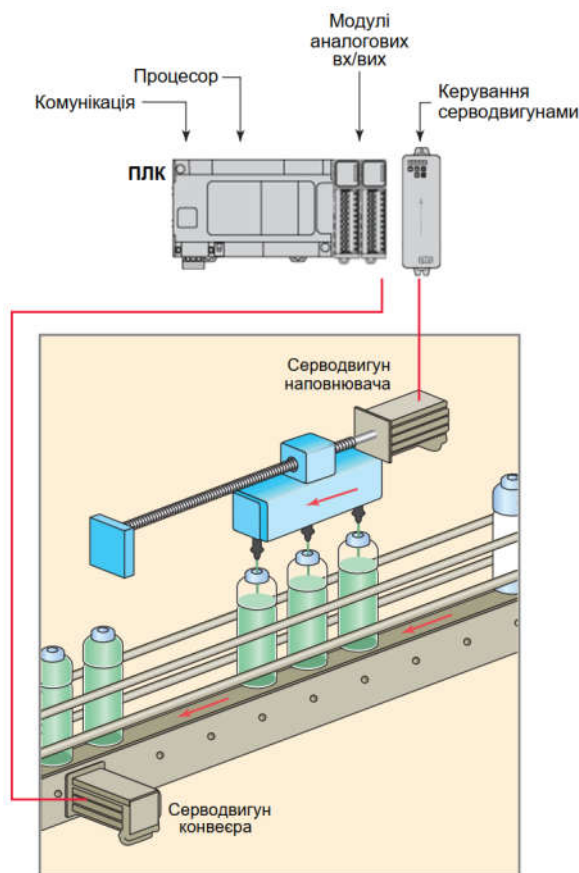


Рис. 6.20. Система контролю процесом наповнення пляшок

Модуль керування рухом приймає ці команди від контролера і перетворює їх у формат, зрозумілий для сервопривода. Крім того, він постійно передає контролеру дані щодо стану двигунів і привода, що дозволяє здійснювати моніторинг ефективності та стабільності роботи системи.

Сервопривід, у свою чергу, отримує сигнали від модуля руху і перетворює їх у команди керування двигуном. Ці команди можуть стосуватися положення вала, швидкості обертання або величини крутного моменту. Сервопривід також відповідає за подачу електричної потужності на двигун, забезпечуючи його точну реакцію на отримані команди. Для контролю положення та швидкості двигуна використовується енкодер, змонтований на його валу. Сигнали зворотного зв'язку від енкодера дозволяють сервоприводу постійно коригувати рух, підтримуючи задану точність і стабільність.

Сервомотор є безпосереднім виконавчим елементом, який здійснює рух осі. Він отримує електроживлення від сервопривода, який визначає його швидкість і положення. У процесі розливу пляшок, наприклад, двигун наповнювача має синхронізувати свій рух із швидкістю конвеєра, щоб точно відповідати положенню кожної пляшки під час заповнення. Після завершення циклу розливу

двигун зупиняє механізм, змінює напрям обертання і повертає його у вихідну позицію для початку нового циклу.

Роботизовані системи є природним продовженням технологій керування рухом. По суті, робот складається з ряду механічних ланок, які з'єднані між собою і приводяться в дію сервомоторами. Найпоширеніший тип промислового робота це маніпулятор або роботизована рука, що здійснює переміщення для виконання різних виробничих операцій (рис. 6.21).

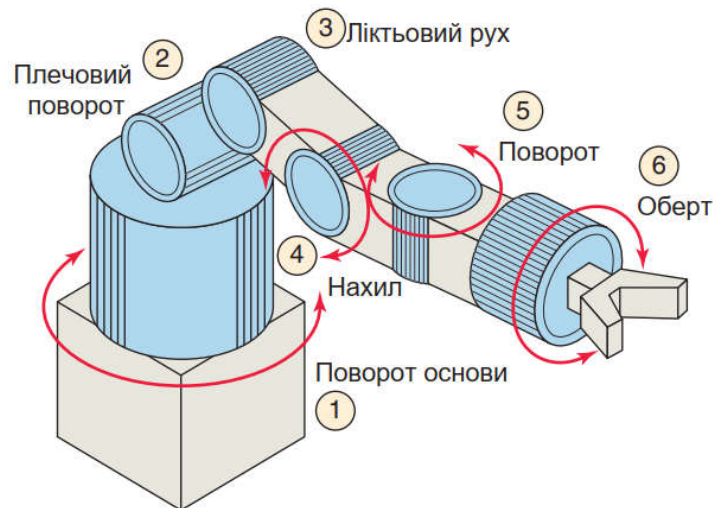


Рис. 6.21. Шести осьова рука робота

Типовий промисловий робот має шість осей руху, кожна з яких працює як замкнена система з сервоконтуром зворотного зв'язку. Три останні осі зазвичай утворюють так зване зап'ястя робота, яке забезпечує рухи типу нахилу, повороту та обертання – відомі як Roll-Pitch-Yaw (RPY). Така структура дозволяє роботу виконувати складні просторові маніпуляції з високою точністю.

Керування промисловими роботами може здійснюватися двома основними способами: за допомогою ПЛК або за допомогою комп'ютерних контролерів. Вибір системи залежить від складності технологічного завдання. Якщо робот виконує порівняно прості дії, таких як перенесення або позиціонування об'єктів, достатньо системи на базі ПЛК. У випадках, коли робот виконує складні координовані рухи, що вимагають високої швидкості обчислень і точної синхронізації декількох осей, використовується спеціалізований робот-контролер або ПК-орієнтована система.

Таким чином, система керування рухом є складним багаторівневим механізмом, у якому кожен компонент від контролера до сервомотора виконує важливу роль у досягненні високої точності, стабільності та узгодженості дій усіх елементів виробничого процесу.

6.6. Передача даних.

Під передачею даних розуміють різноманітні способи обміну інформацією між мікропроцесорними системами на базі ПЛК та іншими пристроями. Такий

обмін є невід'ємною частиною роботи сучасних автоматизованих систем керування. Існують два основні типи зв'язку, що можуть бути встановлені між контролером і зовнішніми пристроями точка-точка та мережеве з'єднання.

Найпростішою формою є послідовний зв'язок типу «точка-точка» (рис. 6.22), який забезпечує обмін даними між двома пристроями, наприклад між ПЛК і ПК, операторською станцією, електроприводом, зчитувачем штрихкодів, комп'ютером або іншим контролером. Послідовний інтерфейс може бути вбудований безпосередньо в процесорний модуль або виконаний у вигляді окремого комунікаційного модуля. Для створення з'єднання між двома ПЛК одного виробника зазвичай достатньо встановити по одному такому модулю в кожен контролер.

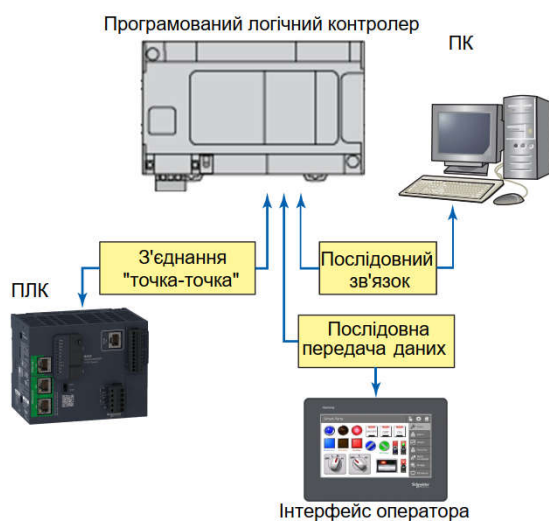


Рис. 6.22. Послідовне з'єднання типу точка-точка

З розвитком промислової автоматизації системи керування стають дедалі складнішими, і для їх ефективного функціонування потрібні розвиненіші комунікаційні схеми. У таких випадках використовується локальна обчислювальна мережа (LAN) система, що з'єднує елементи передавання даних у межах обмеженої території, зазвичай до кількох кілометрів (рис. 6.23). Мережа LAN дозволяє організувати обмін інформацією між численними ПЛК та іншими пристроями, забезпечуючи централізований контроль, спільне використання даних і координацію дій різних елементів системи.

Такі мережі дають змогу здійснювати моніторинг стану системи з одного центру, завантажувати або оновлювати програми керування, а також забезпечувати спільну роботу кількох контролерів для досягнення єдиної технологічної мети.

Основним середовищем передавання даних у мережах є кабелі, через які проходять сигнали керування та інформаційні потоки. У промислових системах найчастіше використовуються коаксіальні кабелі, вита пара або волоконно-оптичні лінії. Вибір типу кабелю залежить від вимог до швидкості передавання, електричних характеристик і умов експлуатації. Поряд із кабельними мережами сьогодні все ширше застосовуються бездротові технології, наприклад Wi-Fi Ethernet, які передають дані за допомогою радіохвиль.



Рис. 6.23. З'єднання з використанням локальної мережі

У промисловій автоматизації локальні мережі часто є основою розподілених систем керування (DCS), де окремі контролери відповідають за керування певними підсистемами, що відрізняє цей підхід від централізованого управління, коли один контролер контролює всю установку. Іншим поширеним напрямом використання LAN є системи диспетчерського керування та збору даних (SCADA), які забезпечують централізований збір, обробку й аналіз даних від групи контролерів через один головний комп'ютер.

Мережеві системи промислової автоматизації зазвичай мають три рівні функціональності (рис. 6.24). Рівень пристроїв включає датчики, вимикачі, двигуни, клапани та інші елементи, безпосередньо пов'язані з технологічним процесом. Рівень керування охоплює ПЛК, роботизовані контролери та інші пристрої, що координують роботу виконавчих механізмів і здійснюють обмін вхідними та вихідними даними. Інформаційний рівень охоплює загальнозаводську мережу, до якої входять комп'ютери, офісні системи та корпоративні бази даних, що забезпечують управлінські, планувальні і аналітичні функції.

Кожен пристрій, підключений до мережі, називається вузлом або станцією. Сигнали, що передаються кабелем, поступово слабшають у процесі, який називають згасанням. Якщо кабель надто довгий, сигнал може спотворитися настільки, що стане нерозпізнаваним. Для компенсації цього ефекту використовуються повторювачі, які підсилюють сигнал і відновлюють його первісну форму, дозволяючи передавати дані на більші відстані.

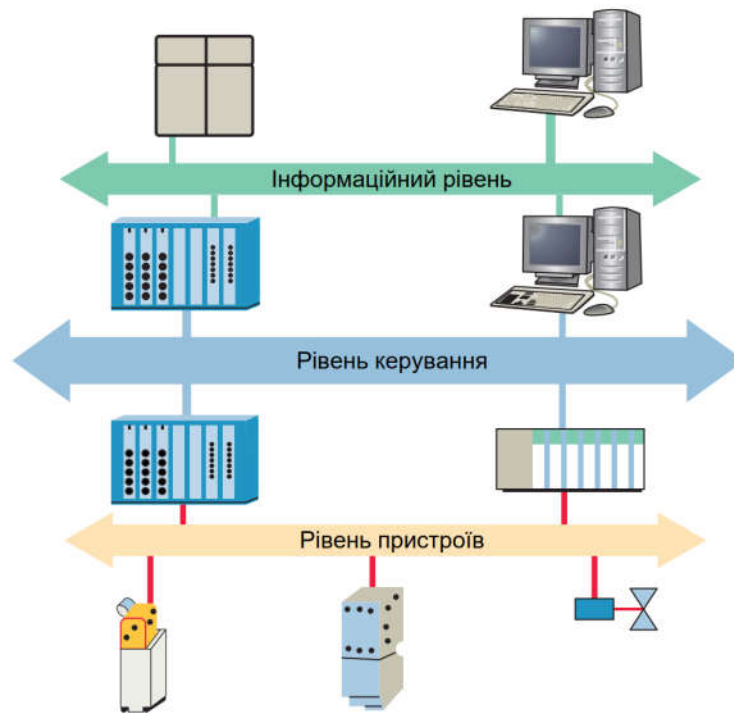


Рис. 6.24. Рівні функціональності промислових мереж

Фізичне розташування пристроїв у мережі називають топологією. Найпоширенішою є *топологія типу зірка*, у якій усі ПЛК з'єднані з центральним комутатором або концентратором (рис. 6.25). У сучасних системах частіше використовуються комутатори (switch), що забезпечують більшу швидкість і пропускну здатність порівняно з концентраторами (hub). Ця конфігурація забезпечує двосторонній обмін даними між центральним пристроєм і кожним контролером. Проте її недоліком є залежність усієї системи від центрального вузла: у разі відмови комутатора робота всієї мережі припиняється.

Іншим типом організації є *шинна топологія*, у якій усі пристрої підключаються до спільного магістрального кабелю (рис. 6.26). Кожен ПЛК отримує дані, що передаються мережею, але приймає лише ті, що адресовані саме йому. На кінцях магістралі встановлюються термінальні резистори, які запобігають відбиттю сигналу. Перевагою шинної структури є простота монтажу, менша витрата кабелю й можливість легкого підключення або від'єднання пристроїв. Проте розрив головної шини може призвести до зупинки значної частини мережі.

Мережі введення-виведення (I/O-bus) поділяються на пристроєві та процесні. Пристроєві мережі обслуговують низькорівневі елементи, такі як кнопки, вимикачі або малі аналогові пристрої, і передають інформацію про стан «увімкнено/вимкнено» або інші базові параметри. Вони можуть бути побітовими, коли передається менше 8 бітів даних, або побайтовими, здатними обробляти десятки байтів інформації. Процесні шини, навпаки, забезпечують обмін сотнями байтів даних за одну передачу та використовуються переважно для аналогових пристроїв наприклад, інтелектуальних клапанів і витратомірів, які регулюють потік чи температуру. Через великі пакети даних такі мережі

працюють повільніше, але дозволяють передавати детальну інформацію про технологічний процес.

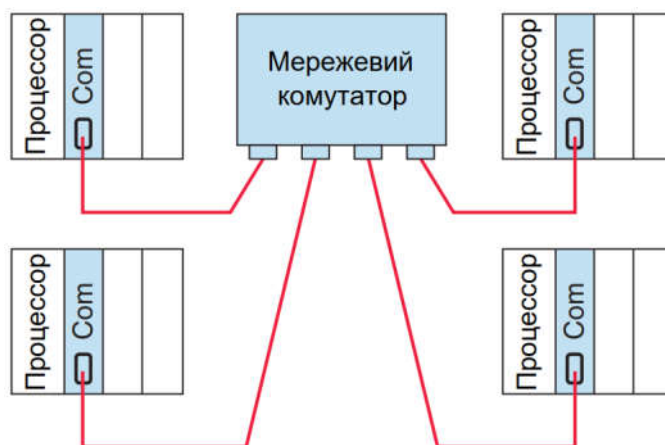


Рис. 6.25. Топологія типу зірка

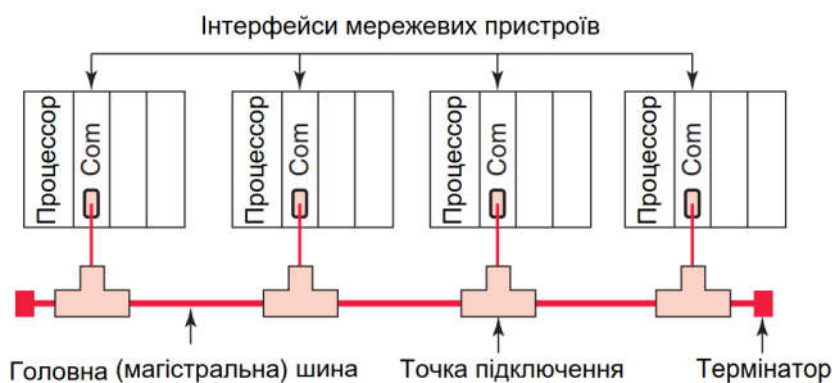


Рис. 6.26. Топологія типу шина

Взаємодія пристроїв у мережі регулюється протоколом обміну даними набором правил, що визначає формат, структуру, порядок і контроль передавання інформації. Для успішної комунікації всі пристрої мають підтримувати один і той самий протокол, так само як люди мають знати спільну мову, щоб порозумітися. Раніше більшість промислових мереж були закритими й прив'язаними до стандартів конкретного виробника, тому користувачі були змушені купувати всі компоненти системи у одного постачальника. Сьогодні ж домінує тенденція до створення відкритих мережевих систем, побудованих на основі міжнародних стандартів.

Для забезпечення взаємодії між різними мережами використовуються шлюзи (gateways), які перетворюють і узгоджують дані, що передаються між системами з різними протоколами (рис. 6.27). Якщо пристрій лише забезпечує трансляцію адрес між мережами, його називають мостом (bridge), а якщо він додатково переформатує дані чи контролює передавання шлюзом.

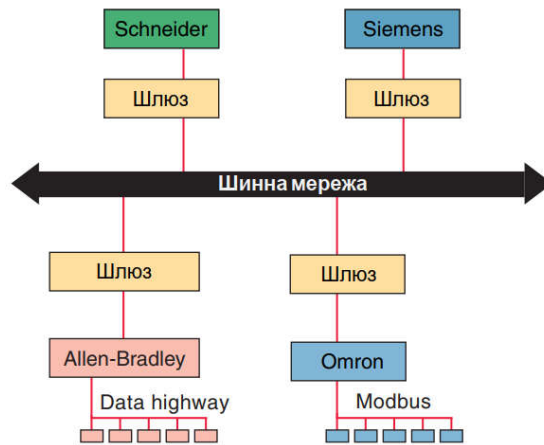


Рис. 6.27. Трансляцію від одної схеми мережевого доступу до іншого

У мережах із шинною топологією необхідно визначати спосіб доступу пристроїв до спільного каналу. Існують три основні методи передавання токена, виявлення колізій та опитування. У системах із токеном, право на передавання даних надається лише тому вузлу, який володіє спеціальним пакетом токеном (рис. 6.28). Після завершення передачі токен передається наступному пристрою за певною послідовністю. Такий метод гарантує впорядкований доступ і запобігає конфліктам сигналів.

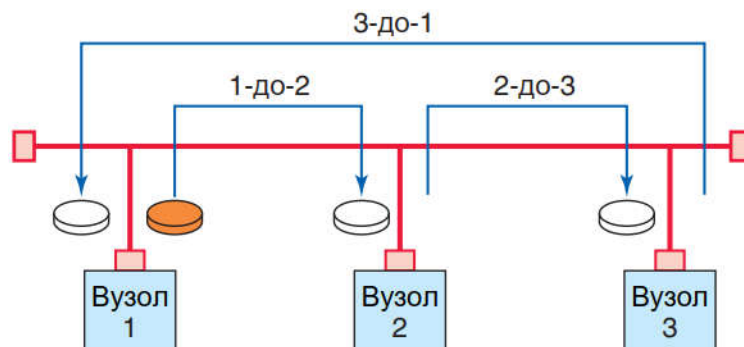


Рис. 6.28. Приклад передачі токена

В Ethernet-мережах використовується метод виявлення колізій: кожен вузол перед передаванням перевіряє, чи вільний канал. Якщо кілька вузлів одночасно починають передавати дані, виникає колізія, після чого вони припиняють передавання, очікують випадковий проміжок часу й повторюють спробу.

Інший підхід метод опитування, характерний для мереж типу «Ведучий/Ведений» (Master/Slave) (рис. 6.29). У такій системі головний контролер ініціює всі обміни, надсилає команди підлеглим пристроям і запитує від них необхідні дані. Прямий обмін між підлеглими вузлами не допускається будь-яка інформація передається через головний вузол.

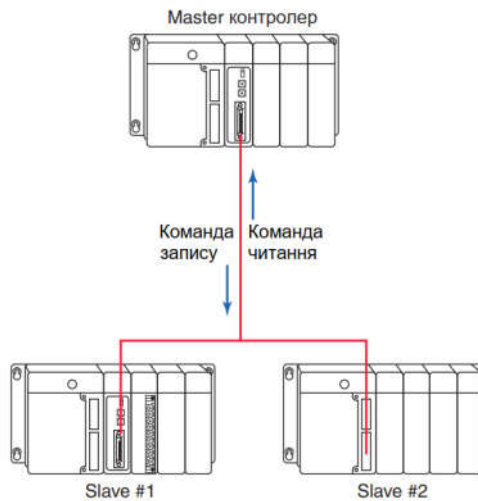


Рис. 6.29. Мережа «Ведучий/Ведений»

Передача цифрових даних у ПЛК може здійснюватися *паралельно* або *послідовно*. При *паралельній* передачі всі біти байта передаються одночасно окремими лініями, що забезпечує високу швидкість, але потребує багатожильного кабелю (рис. 6.30). *Послідовна* передача передає біти один за одним по одному каналу, що зменшує кількість проводів і дозволяє здійснювати обмін на більших відстанях, хоча зі зниженою швидкістю (рис. 6.31). Для маркування початку та кінця пакета використовуються спеціальні біти стартовий і стоповий, а додатковий біт парності може забезпечувати контроль помилок.

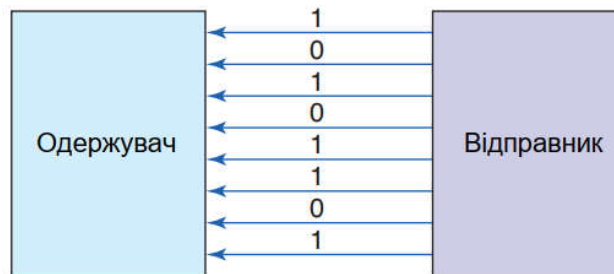


Рис. 6.30. Паралельна передача даних

Комунікаційні системи можуть бути напівдуплексними, коли обмін відбувається в обох напрямках, але не одночасно, або повнодуплексними, які дозволяють одночасну двосторонню передачу наприклад, у мережах рівноправних пристроїв.

Мережеві технології значно скорочують кількість проводів у системах керування, замінюючи традиційні схеми жорсткого дротового з'єднання. Це зменшує витрати на монтаж, спрощує обслуговування і полегшує пошук несправностей. Замість численних кабельних трас усі пристрої можуть бути підключені до єдиної лінії зв'язку.

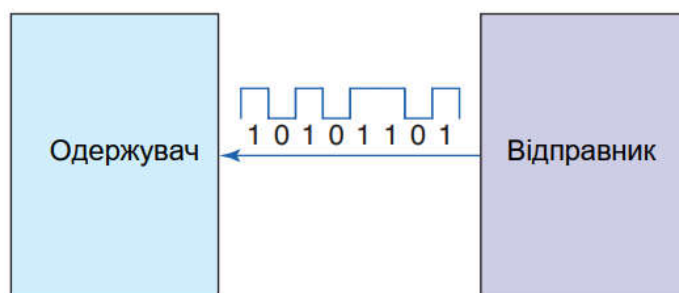


Рис. 6.31. Послідовна передача даних

Сучасні високошвидкісні промислові мережі пропонують широкий спектр методів підключення обладнання. Конфігурації мереж ПЛК можуть бути як відкритими, побудованими на загально визначених стандартах, так і власницькими, розробленими конкретними виробниками. У будь-якому разі промислові мережі відіграють ключову роль у сучасних системах автоматизації, забезпечуючи швидкий, надійний та ефективний обмін інформацією між усіма рівнями керування.

6.6.1. Послідовна передача даних (Serial Communication).

Послідовна передача даних у промислових системах реалізується відповідно до міжнародних стандартів, таких як RS-232, RS-422 та RS-485. Аббревіатура RS означає Recommended Standard «рекомендований стандарт», що визначає електричні, механічні та функціональні характеристики інтерфейсів послідовної комунікації.

Інтерфейси послідовного зв'язку можуть бути вбудовані безпосередньо в процесорний модуль контролера або виконані як окремі модулі зв'язку, що під'єднуються до ПЛК залежно від вимог системи (6.32).

Найпростішим варіантом послідовного підключення є порт RS-232. Він використовується для обміну даними між одним комп'ютером і одним контролером. Такий тип зв'язку забезпечує передачу даних із достатньо високою швидкістю, проте має обмеження по довжині кабелю до 50 футів (приблизно 15 метрів).

Інтерфейси RS-422 і RS-485 розроблені для більш складних систем, у яких один комп'ютер може взаємодіяти з кількома контролерами одночасно. Вони відрізняються вищою завадостійкістю, що дозволяє використовувати їх у промисловому середовищі з підвищеним рівнем електромагнітних перешкод. Максимальна довжина лінії передачі для стандарту RS-485 зазвичай становить близько 650 футів (приблизно 200 метрів), а для RS-422 до 1650 футів (приблизно 500 метрів).

Такі інтерфейси широко застосовуються для підключення пристроїв, які потребують оперативного обміну великими обсягами інформації, зокрема систем машинного зору, сканерів штрих-кодів, операторських терміналів і різноманітних периферійних пристроїв, що працюють у зв'язці з ПЛК.

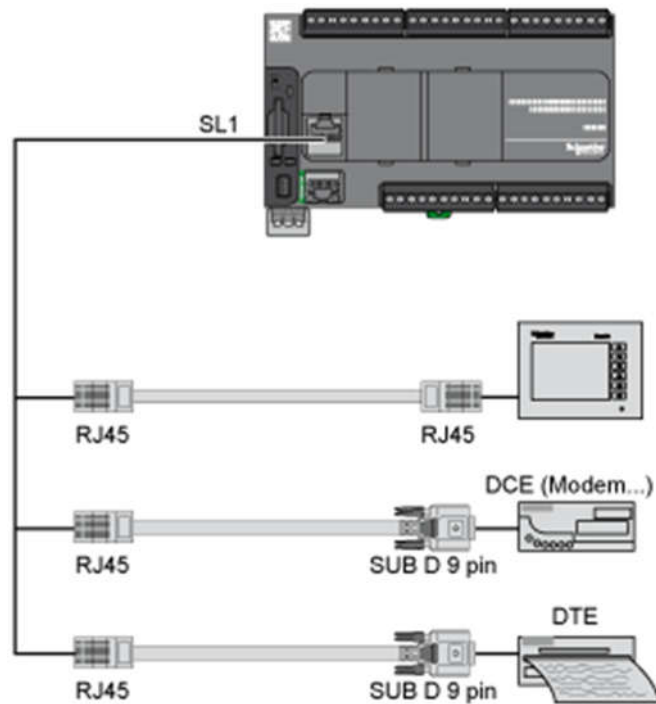


Рис. 6.32. Послідовна передача даних ПЛК Modicon M221

6.6.2. EtherNet/IP.

EtherNet/IP (Ethernet Industrial Protocol) це відкритий протокол промислової комунікації, побудований на основі Common Industrial Protocol (CIP). Завдяки цьому EtherNet/IP забезпечує повну сумісність і можливість безперешкодного обміну даними між різними пристроями, що працюють за одним протоколом, без необхідності застосування спеціального апаратного забезпечення (рис. 6.33).

Однією з головних переваг EtherNet/IP є спільний рівень прикладного протоколу (CIP) для всіх трьох систем це ControlNet, DeviceNet та EtherNet/IP. Це дає змогу досягати повної взаємодії між обладнанням різних виробників, реалізуючи концепцію plug-and-play. Завдяки цьому нові пристрої можуть автоматично розпізнаватися системою після підключення, без необхідності ручного налаштування перемикачів, джамперів чи інших параметрів. Такий підхід значно спрощує введення в експлуатацію та інтеграцію нових елементів у вже існуючу мережу.

Ще однією важливою характеристикою EtherNet/IP є підтримка стандартного повнодуплексного режиму передачі даних. Це означає, що вузол у з'єднанні типу «peer-to-peer» може одночасно передавати й приймати дані, використовуючи повну пропускну здатність каналу. Такий принцип роботи забезпечує максимально можливу продуктивність мережі, оскільки весь доступний обсяг пропускну здатності (bandwidth) виділяється конкретному з'єднанню. Ширина пропускну каналу визначає швидкість передачі даних (у бітах за секунду), і чим вона більша тим вищою є загальна ефективність системи.

EtherNet/IP також має важливу перевагу універсальність інтеграції. Він дозволяє одночасно об'єднувати промислові пристрої автоматизації та звичайні інформаційні системи підприємства в єдину мережу. Це означає, що через одну інфраструктуру можна забезпечити як керування технологічними процесами, так

і доступ до бізнес-додатків, корпоративних баз даних чи навіть Інтернету. Така гнучкість робить EtherNet/IP потужним засобом побудови інтегрованих систем керування підприємством, де обмін інформацією між виробничим рівнем і адміністративним рівнем відбувається в реальному часі.

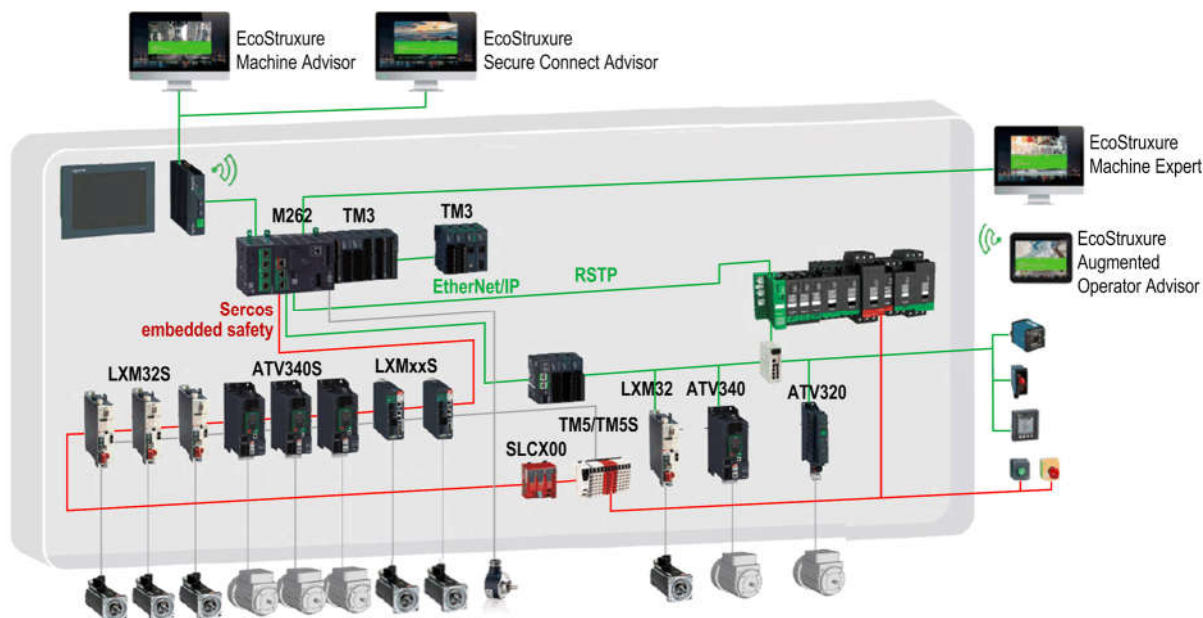


Рис. 6.33. Інформаційні зв'язки EtherNet/IP

6.6.3. Modbus.

Modbus це послідовний комунікаційний протокол, який був розроблений компанією Modicon спеціально для використання з її програмованими логічними контролерами (ПЛК). Його основне призначення полягає у передачі даних по послідовних лініях зв'язку між різними електронними пристроями. Принцип роботи протоколу ґрунтується на моделі «Ведучий/Ведений» (Master/Slave): пристрій, який ініціює запит інформації, виконує роль Modbus Master, тоді як пристрої, що надають дані або відповідають на запити, виступають як Modbus Slaves.

Однією з ключових переваг Modbus є те, що це відкритий протокол, тобто його можна вільно впроваджувати у будь-яке обладнання без сплати ліцензійних відрахувань. Завдяки цій відкритості Modbus швидко став галузевим стандартом у промисловості, забезпечуючи простий і надійний спосіб зв'язку між різними типами промислових електронних пристроїв контролерами, датчиками, приводами, вимірювальними системами тощо.

Сьогодні Modbus є одним із найпоширеніших протоколів промислової комунікації, який використовується для обміну даними між пристроями різних виробників. Його популярність пояснюється простотою реалізації, надійністю, стабільністю та широкою підтримкою в системах автоматизації. Протокол забезпечує передачу даних через стандартні послідовні інтерфейси RS-232C, RS-422 або RS-485, що дозволяє легко інтегрувати його в існуючі системи управління.

Наприклад ПЛК Modicon M241 від компанії Schneider Electric (рис. 6.34), який підтримує мережеву комунікацію за протоколом Modbus RTU через порти RS-232C та RS-422/485. Це демонструє гнучкість і універсальність Modbus, який може функціонувати як у простих локальних системах, так і в складних промислових мережах, забезпечуючи стабільний обмін інформацією між усіма компонентами автоматизованого виробничого процесу.



Рис. 6.34. ПЛК Modicon M241 від компанії Schneider Electric

6.6.4. Польова шина (Fieldbus).

Польова шина (Fieldbus) це відкрита, послідовна, двонаправлена система зв'язку, призначена для об'єднання вимірювального та керуючого обладнання таких пристроїв, як датчики, виконавчі механізми, контролери та інші елементи систем автоматизації. Ця технологія функціонує на базовому рівні ієрархії виробничих мереж, забезпечуючи ефективну взаємодію між польовими пристроями, які виконують ключові функції у процесному керуванні.

Основна мета Fieldbus полягає у створенні єдиного цифрового середовища обміну даними між усіма пристроями системи управління, що дозволяє зменшити кількість окремих з'єднань, спростити монтаж і підвищити надійність комунікацій. На відміну від традиційних аналогових систем, де кожен датчик або виконавчий механізм підключається до контролера окремим кабелем, Fieldbus дає змогу передавати сигнали по спільній шині зв'язку, що суттєво скорочує витрати на прокладання та обслуговування кабельної інфраструктури.

Для побудови Fieldbus-мереж існує кілька можливих топологій підключення пристроїв. Однією з найпоширеніших є послідовна (daisy-chain) топологія, зображена на рис. 6.35. У такій конфігурації кабель Fieldbus прокладається послідовно від одного пристрою до іншого, утворюючи єдину лінію зв'язку. При цьому система має бути змонтована таким чином, щоб від'єднання одного пристрою не порушувало роботу всієї мережі. Це досягається використанням спеціальних роз'ємів або схем підключення, які забезпечують безперервність сигналу навіть у разі тимчасового відключення окремих вузлів.

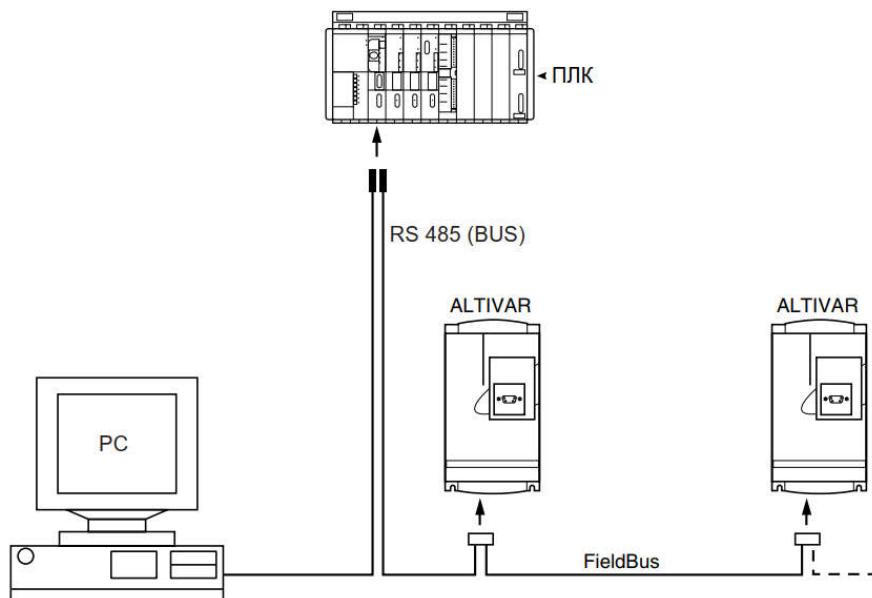


Рис. 6.35. Послідовне з'єднання польовою шиною

Таким чином, Fieldbus виступає фундаментом сучасних розподілених систем управління (DCS), забезпечуючи швидкий, надійний і гнучкий обмін даними між усіма компонентами автоматизованого виробничого процесу.

6.6.5. CANOpen.

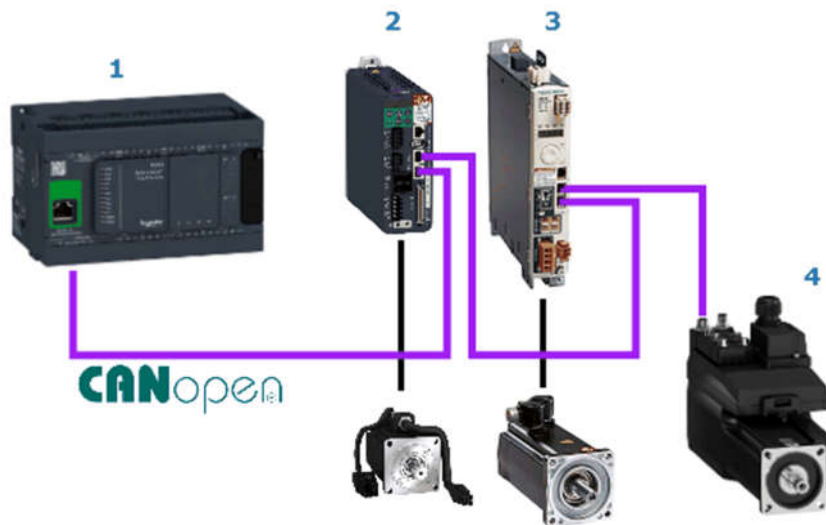
CANopen це відкритий протокол промислової комунікації, розроблений організацією CiA (CAN in Automation) на базі шини Controller Area Network (CAN). Його основне призначення це забезпечення взаємодії між контролерами, сенсорами, приводами, вимірювальними модулями та іншими пристроями автоматизації у реальному часі (рис. 6.36).

На відміну від базового рівня CAN, який визначає лише фізичний і каналний рівні передачі даних, CANopen описує повний набір комунікаційних і прикладних сервісів, що дозволяють реалізувати гнучку структуру керування. Цей протокол задає стандартизований об'єктний словник (Object Dictionary), де всі параметри пристрою мають унікальні ідентифікатори. Завдяки цьому будь-який контролер може автоматично розпізнати підключений вузол, зчитати його характеристики і налаштувати зв'язок без додаткового програмування.

Однією з ключових переваг CANopen є висока надійність і детермінованість обміну даними, що робить його придатним для систем із жорсткими вимогами до часу відгуку, наприклад, у сервоприводах, системах керування рухом, робототехніці або медичному обладнанні. Передача повідомлень у CANopen організована за подієвим принципом, що дозволяє оптимізувати навантаження мережі та зменшити кількість непотрібних передач.

Архітектура CANopen підтримує декілька типів комунікаційних об'єктів:
 PDO (Process Data Object) – для швидкого обміну даними в реальному часі;
 SDO (Service Data Object) – для конфігурації та обміну параметрами;
 NMT (Network Management) – для керування станом вузлів;

SYNC і EMCY – для синхронізації дій і передачі аварійних повідомлень відповідно.



**Рис. 6.36. Структура інформаційних зв'язків у мережі CANopen:
1 – Modicon M241; 2 – Lexium 28; 3 – Lexium 32A;
4 – Integrated Lexium 32i**

CANopen дозволяє будувати мережі з до 127 пристроїв на одній шині, використовуючи двожильний екранований кабель з диференціальним сигналом, що забезпечує стійкість до електромагнітних завад і високу надійність зв'язку. Максимальна швидкість обміну даними може досягати 1 Мбіт/с при довжині лінії до 40 метрів, або менше при більшій відстані – відповідно до вимог конкретної системи.

Завдяки своїй модульності та простоті інтеграції, CANopen став одним із найпоширеніших промислових протоколів у галузях, де потрібна стабільна взаємодія між мікроконтролерами, ПЛК і приводами. Його підтримують багато виробників, серед яких Schneider Electric, Siemens, Beckhoff, Bosch Rexroth, Omron та інші.

6.7. СКАДА.

У деяких виробничих процесах, окрім стандартних функцій керування, ПЛК може виконувати додаткові завдання збір, обробку та структурування даних з метою подальшого аналізу та створення звітів. Наприклад, контролер може підраховувати кількість виготовлених деталей і автоматично передавати отриману інформацію безпосередньо до електронної таблиці на комп'ютері оператора.

Цей процес суттєво спрощується завдяки використанню системи диспетчерського керування та збору даних (SCADA – Supervisory Control and Data Acquisition) (SCADA) (рис. 6.37). Основна функція СКАДА полягає в обміні даними між рівнем виробництва та центральним комп'ютером. Це забезпечує широкий спектр можливостей: ведення журналів даних, відображення

параметрів процесу, аналіз тенденцій, завантаження рецептур, налаштування параметрів роботи обладнання та загальний моніторинг виробничої інформації. Крім того, СКАДА надає засоби диспетчерського керування, які дозволяють оператору тонко коригувати параметри процесу для досягнення максимальної ефективності.

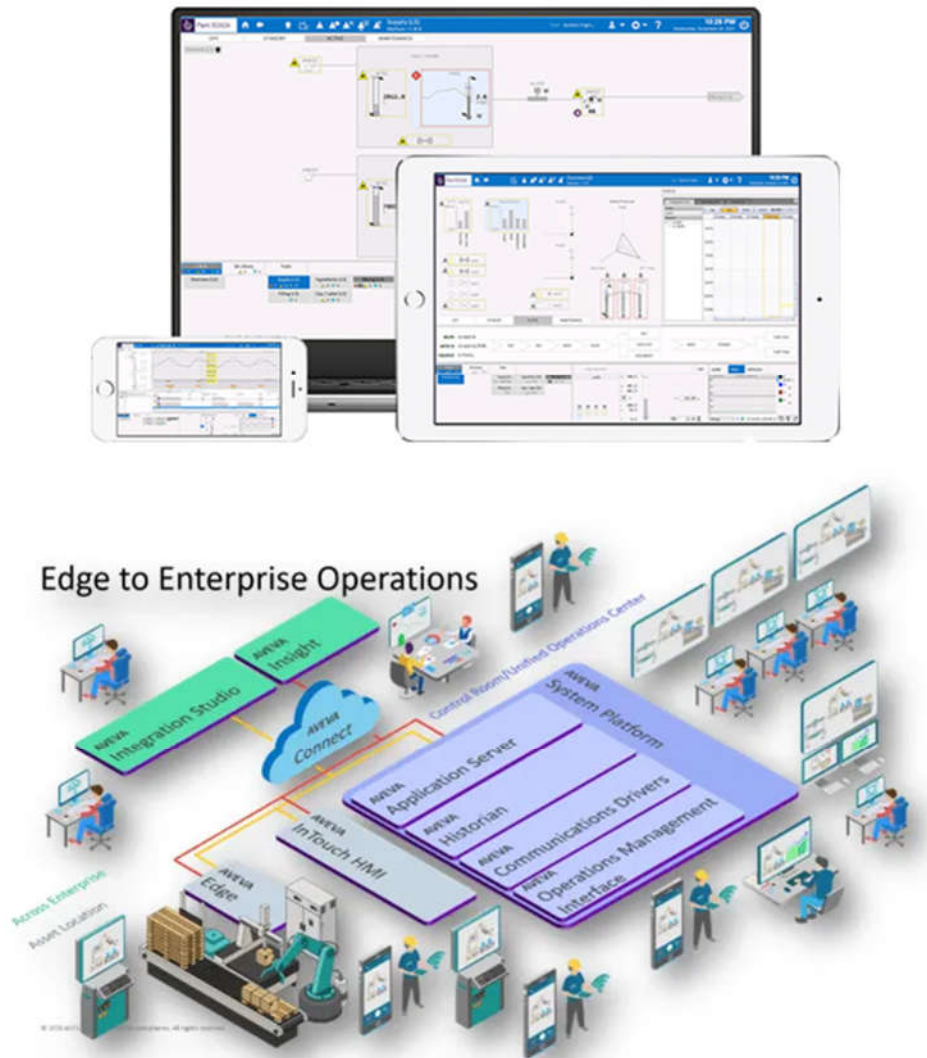


Рис. 6.37. Принципова схема СКАДА системи від компанії AVEVA

На відміну від розподілених систем керування (DCS), СКАДА, як правило, не виконує керування в реальному часі, а лише координує роботу окремих підсистем, збираючи та обробляючи інформацію про стан технологічного процесу.

У типовій СКАДА-архітектурі окремі ПЛК контролюють роботу польових пристроїв (датчиків, виконавчих механізмів тощо), тоді як центральний комп'ютер із програмним забезпеченням СКАДА/ЛМІ здійснює моніторинг, візуалізацію та координацію цих процесів (рис. 6.38). Оператор спостерігає за станом обладнання, аналізує параметри, а за необхідності надсилає команди керування безпосередньо до ПЛК.

Важливою перевагою СКАДА є автоматизоване збереження всіх отриманих даних у цифровому вигляді, що забезпечує точність, доступність та зручність подальшого аналізу. Дані збираються під контролем процесора, виводяться на екран у реальному часі та можуть бути надруковані або збережені у звіт. Такий підхід дозволяє отримувати максимально точні результати вимірювань без механічних обмежень, що значно підвищує ефективність контролю та аналітики на виробництві.

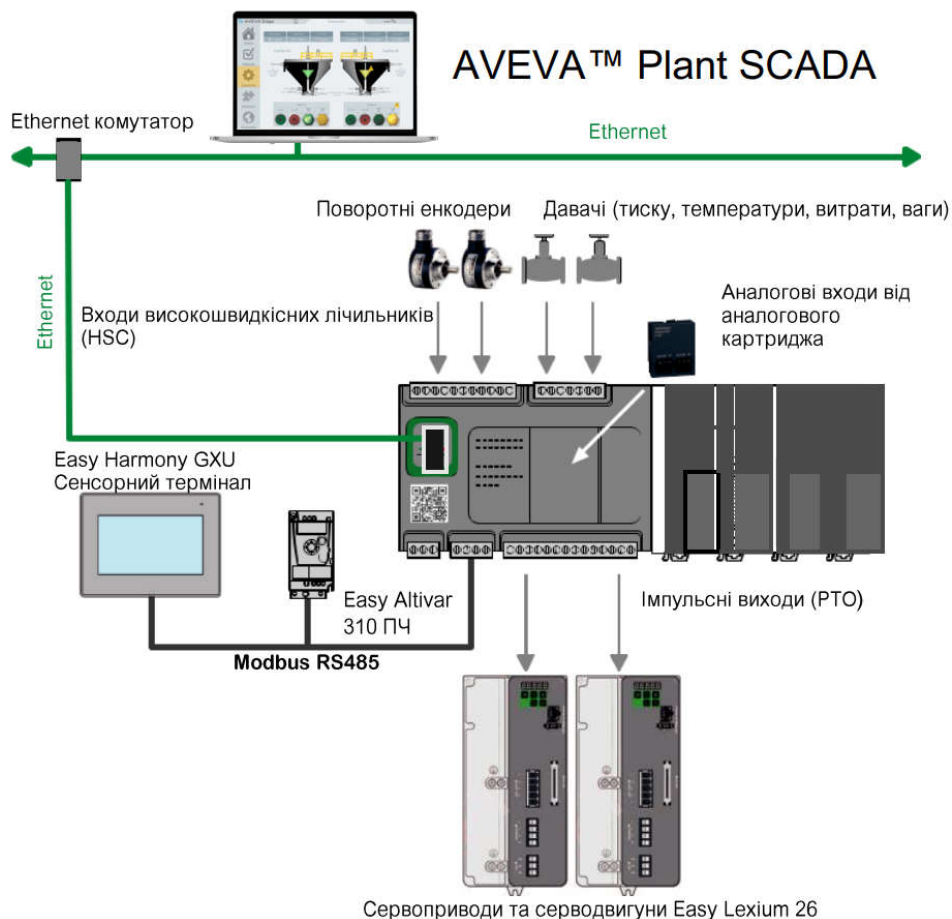


Рис. 6.38. Типовий склад СКАДА системи ПЛК M200

Висновки за розділом.

1. В розділі розглянуто класифікацію технологічних процесів за їх динамікою: безперервні, дискретні та змішані. Визначено, що для кожного типу необхідні свої підходи до автоматизації, вибору алгоритмів керування та технічних засобів.
2. Узагальнено типову побудову автоматизованої системи керування, яка складається з рівнів вимірювання, регулювання, логічного керування, а також операторського рівня. Наголошено на ролі зворотних зв'язків і контролю параметрів у забезпеченні стабільності процесів.
3. Розглянуто найпростіший метод регулювання, який базується на пороговій зміні стану виконавчих механізмів. Зроблено висновок, що хоча цей метод не забезпечує високої точності, він залишається ефективним у побутових і технічних системах з невисокими вимогами до стабільності.

4. Проаналізовано принцип дії ПІД-регулятора як універсального інструменту для точного керування безперервними процесами. Визначено, що налаштування коефіцієнтів пропорційної, інтегральної та диференціальної складових дозволяє досягти оптимальної динамічної реакції системи.
5. Висвітлено принципи побудови систем керування електричними приводами та сервомеханізмами. Підкреслено важливість точного позиціонування, регулювання швидкості й моменту, а також використання зворотного зв'язку для забезпечення стабільної роботи механізмів.
6. Узагальнено відомості про сучасні промислові протоколи обміну даними: EtherNet/IP, Modbus, Fieldbus, CANopen. Визначено їх переваги: сумісність між пристроями, гнучкість масштабування, висока швидкість і надійність зв'язку. Особливо підкреслено роль стандартизації в інтеграції систем різних виробників.
7. Зроблено висновок, що SCADA-технології є ключовою складовою сучасної автоматизації, оскільки забезпечують моніторинг, керування та архівацію даних у реальному часі. Вони дозволяють реалізувати централізоване керування технологічними процесами та підвищити ефективність виробництва за рахунок аналітики та візуалізації параметрів.

Питання для самоперевірки.

1. Які основні типи технологічних процесів виділяють у системах автоматичного керування?
2. У чому полягає різниця між безперервними, дискретними та змішаними процесами?
3. З яких основних елементів складається структура типової системи керування процесом?
4. Які функції виконують датчики, виконавчі механізми та контролери у системах керування?
5. У чому полягає принцип дії системи керування за схемою «Вмикання/Вимикання»?
6. Які недоліки має релейне керування порівняно з ПІД-регулюванням?
7. Поясніть основні складові ПІД-регулятора та їх вплив на точність і швидкодію системи.
8. Як реалізується керування рухом у сучасних автоматизованих системах?
9. Яке значення мають мережеві системи обміну даними для організації SCADA-систем?
10. Які функції виконує SCADA-система у структурі промислової автоматизації?

РОЗДІЛ 7.

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ АВТОМАТИЗОВАНОГО ПРОЄКТУВАННЯ ЕЛЕКТРИЧНИХ СХЕМ ТА СИСТЕМ АВТОМАТИЗАЦІЇ

Сучасне проектування електричних схем і систем автоматизації вимагає високої точності, швидкості та узгодженості між усіма етапами створення документації та від концепції до реалізації. Тому у промисловій автоматизації все ширше застосовується спеціалізоване програмне забезпечення (САПР), яке дозволяє не лише створювати електричні принципові схеми, а й автоматично генерувати специфікації, кабельні журнали, маркування проводів, креслення панелей, 3D-моделі шаф управління, а також забезпечує інтеграцію з системами ERP, PLM чи механічним САД.

До загальних вимог, яким має відповідати сучасне ПЗ для електротехнічного проектування, належать:

- підтримка міжнародних стандартів (IEC, ISO, NFPA, ГОСТ);
- наявність розширених бібліотек символів і компонентів;
- автоматизація рутинних операцій, наприклад нумерації, формування звітів, крос-посилань, перевірок помилок;
- гнучкість у роботі з багатосторінковими схемами;
- інтеграція з системами 3D-моделювання та управління проектами;
- зручний інтерфейс і можливість адаптації під специфіку підприємства.

Розглянемо найпоширеніші програмні продукти, що відповідають цим вимогам.

SolidWorks Electrical (Electrical Schematic Designer) це потужний інструмент від Dassault Systèmes, орієнтований на інтеграцію електричної частини проекту з 3D-моделлю механічних елементів у SolidWorks. Перевагою є повна асоціативність між схемою та 3D-моделлю, що дозволяє уникати помилок при компоновці шаф, трасуванні кабелів та виборі компонентів. Програма підтримує стандарти IEC, ANSI, ICS і має велику бібліотеку символів. До переваг цієї програми можна віднести: інтеграцію з SolidWorks 3D CAD, автоматичне трасування кабелів, зручне управління проектами. Основними недоліками є: висока вартість ліцензії, складність для новачків, потреба у потужному обладнанні.

AutoCAD Electrical це спеціалізована версія AutoCAD від Autodesk, створена для електротехнічних проєктів. Має понад 65 000 бібліотечних символів, автоматичну нумерацію проводів, перевірку логічних зв'язків та генерацію звітів. До переваг цієї програми можна віднести: знайомий інтерфейс для користувачів AutoCAD, широка підтримка стандартів, можливість спільної роботи в корпоративних мережах. Основними недоліками є: обмежені можливості інтеграції з механічним САД, складність у налаштуванні шаблонів, менша глибина автоматизації порівняно з EPLAN.

EPLAN Electric P8 є один із найпотужніших інструментів для електротехнічного проектування. Підтримує автоматичне створення списків з'єднань, кабельних журналів, клемних планів, 3D-проектування шаф (через EPLAN Pro Panel), інтеграцію з базами даних компонентів. До переваг цієї програми можна віднести: висока ступінь автоматизації, підтримка стандартів IEC, можливість розширення через макроси та API, інтеграція з ERP/PLM. Основними недоліками є: висока ціна, складність у навчанні, потреба в потужній технічній підтримці.

Zuken E3.series (E3.schematic) є продуктом від компанії Zuken, що охоплює електричне, кабельне та гідравлічне проектування. Відзначається модульною структурою та гнучкою інтеграцією з механічним САД. До переваг цієї програми можна віднести: підтримку великих проектів, інтеграцію з SolidWorks, CATIA, NX, потужну перевірку зв'язків між схемами. Основними недоліками є: обмежена кількість навчальних матеріалів, складне ліцензування, відносно висока вартість.

SEE Electrical (IGE+XAO Group, Schneider Electric) це інтуїтивно зрозуміле програмне забезпечення для створення електричних схем, проектування шаф керування, розробки систем автоматизації. Програма пропонується у трьох рівнях: Basic, Standard та Advanced, що дозволяє адаптувати її під різні потреби користувача. До переваг цієї програми можна віднести: простий у використанні інтерфейс, швидке навчання, підтримка стандартів IEC і EN 81346, автоматизація рутинних операцій, доступна ціна. Основними недоліками є: менша кількість інтеграцій з 3D САД.

Попри значну кількість сучасних програм для електротехнічного проектування, SEE Electrical займає особливе місце завдяки своїй простоті, стабільності роботи та гнучкій модульності. Вона ідеально підходить для середніх підприємств, навчальних закладів і фахівців, яким необхідне ефективно та зручне рішення без надлишкової складності.

У подальшому розгляді ми зосередимо увагу саме на SEE Electrical, детально проаналізуємо її інтерфейс, структуру проектів, рівні функціональності, а також основні етапи створення електричних схем і документації.

7.1. Інтерфейс користувача в SEE Electrical.

Основне вікно програми SEE Electrical має логічну структуру, що забезпечує швидкий доступ до всіх інструментів і функцій, необхідних для розроблення електричних схем (рис. 7.1). Його інтерфейс складається з кількох ключових елементів: заголовка вікна, рядка меню, панелей інструментів, вікна Electrical Data Explorer, палітри вставки (Insertion Palette), області креслення, списку відкритих аркушів та рядка стану.

Заголовок вікна відображає назву поточної програми та тип активного аркуша, наприклад, Schematic, Terminal Strip, Layout, Block Diagram, Parameter Sheet, Title Block, Symbol або Illustration (рис. 7.2).

Рядок меню містить основні команди та інструменти, згруповані за категоріями (від File до Options), що дає змогу виконувати всі доступні операції

у SEE Electrical (рис. 7.3). Меню відкривається простим натисканням на його назву.

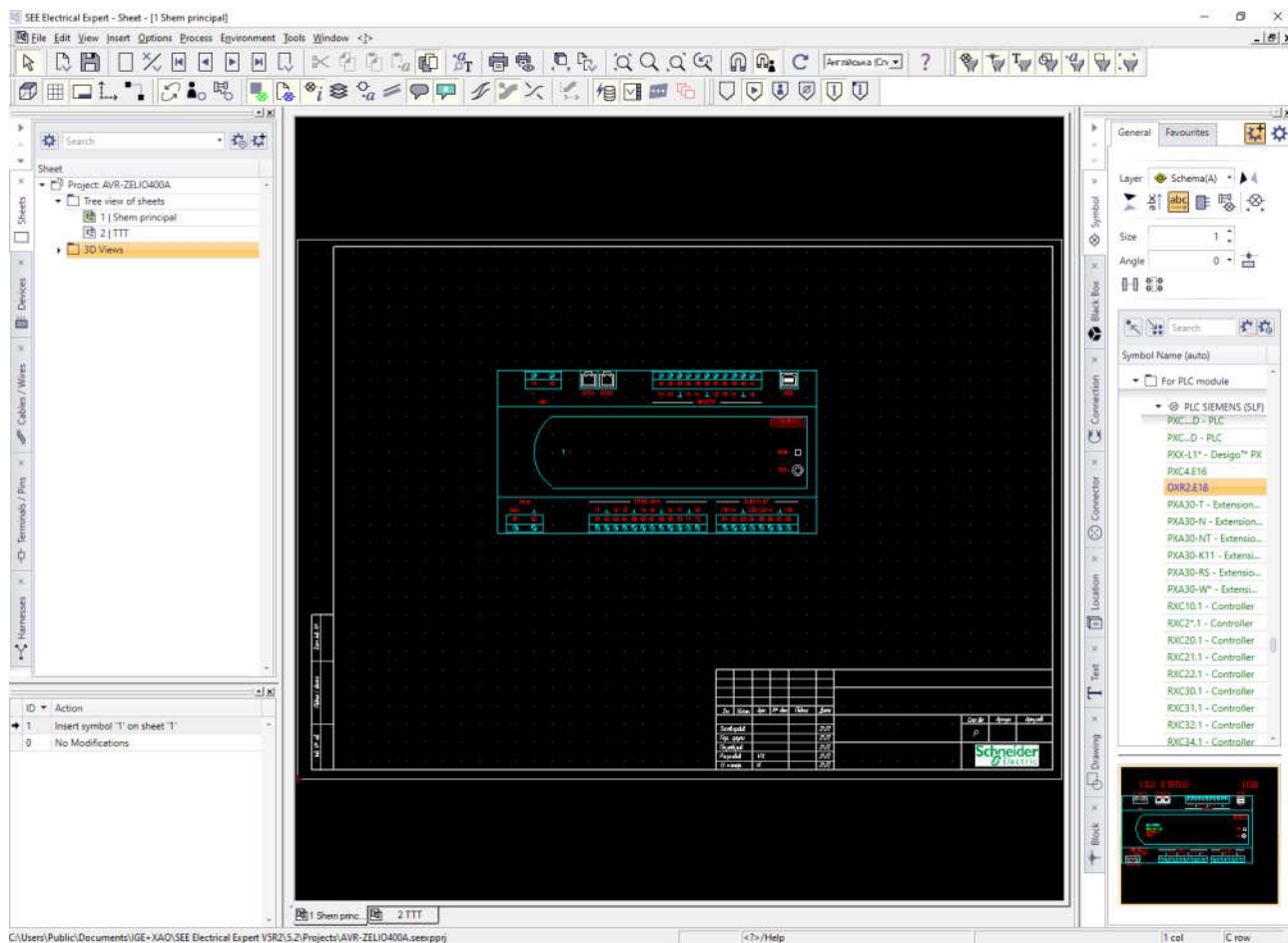


Рис. 7.1. Основне вікно програми SEE Electrical

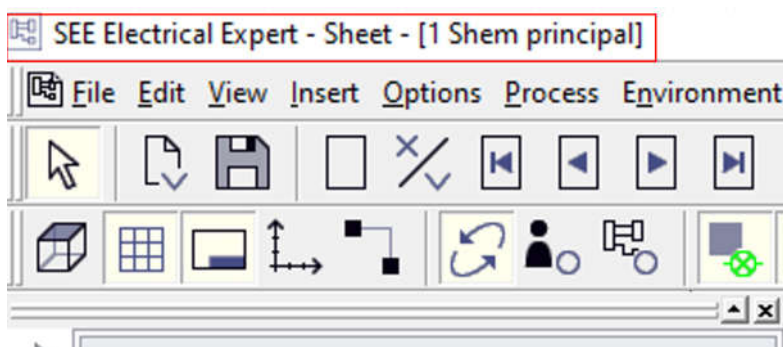


Рис. 7.2. Заголовок вікна програми SEE Electrical

Панелі інструментів (Toolbars) забезпечують швидкий доступ до основних функцій програми (рис. 7.4). При наведенні курсора на іконку з'являється підказка з коротким описом дії, а у рядку стану виводиться детальніша інформація. Панелі можуть бути закріпленими (anchored) тобто розташованими по краях робочого поля, або плаваючими (floating) пересувними в межах області креслення. За замовчуванням доступні панелі Standard, Visibility, Selection Mode і Tagging Mode. Їх відображення, переміщення та розмір можна змінювати через

меню Options → Customize або контекстне меню, викликане правою кнопкою миші.

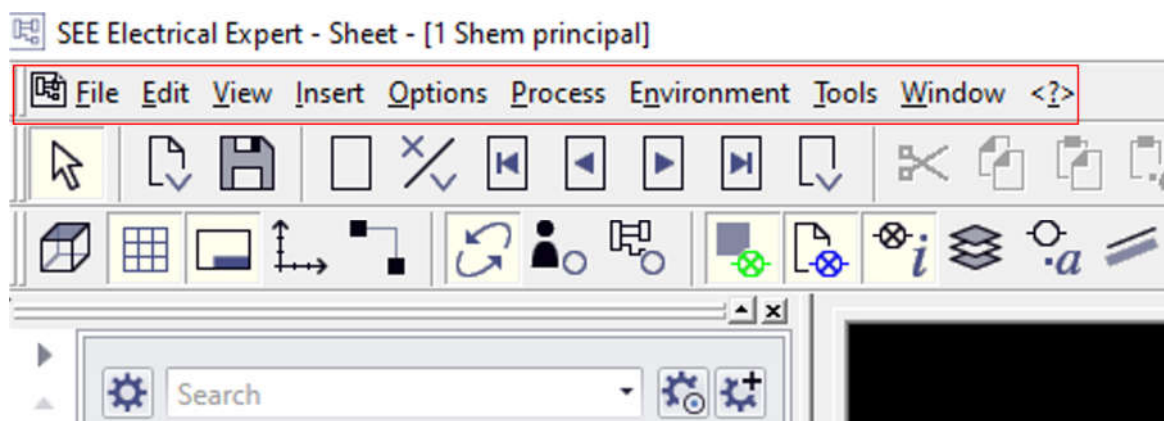


Рис. 7.3. Рядок меню програми SEE Electrical

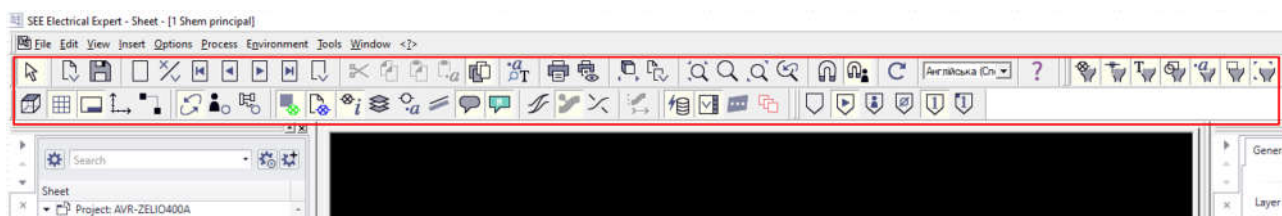


Рис. 7.4. Панель інструментів (Toolbars) програми SEE Electrical

Electrical Data Explorer і пов'язані з ним допоміжні вікна зазвичай розташовуються зліва та відкриваються автоматично (рис. 7.5а). Вони використовуються для навігації по проекту, пошуку елементів, доступу до параметрів та об'єктів.

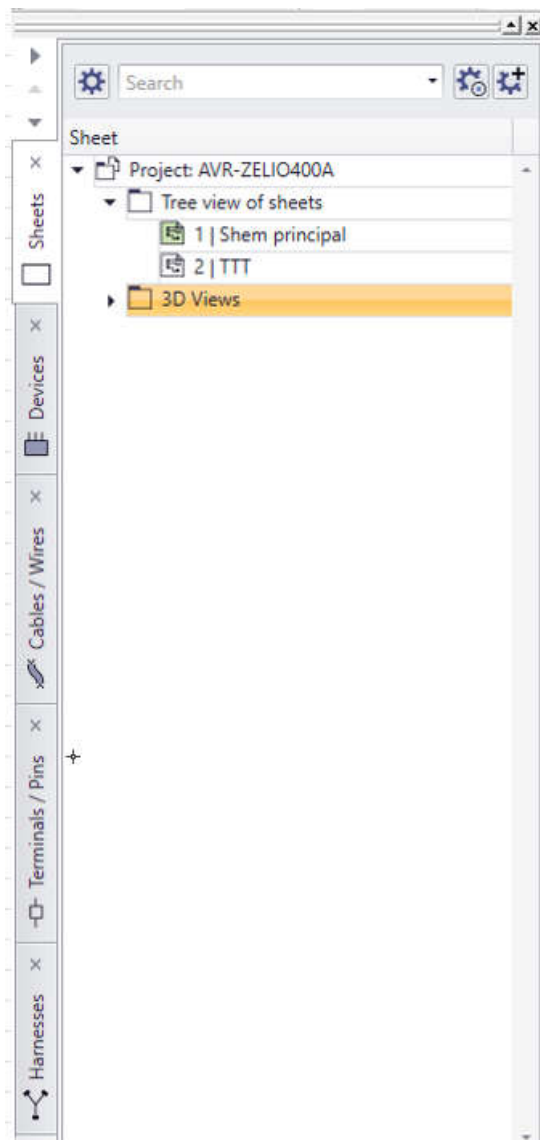
Палітра вставки (Insertion Palette) забезпечує швидке додавання компонентів, символів або елементів схем на аркуш, що дозволяє ефективно працювати з проектом без переходу між меню (рис. 7.5б).

Область креслення (Drawing Area) це основний простір, де відображається активний аркуш і виконується проектування: створення схем, таблиць, макетів або креслень.

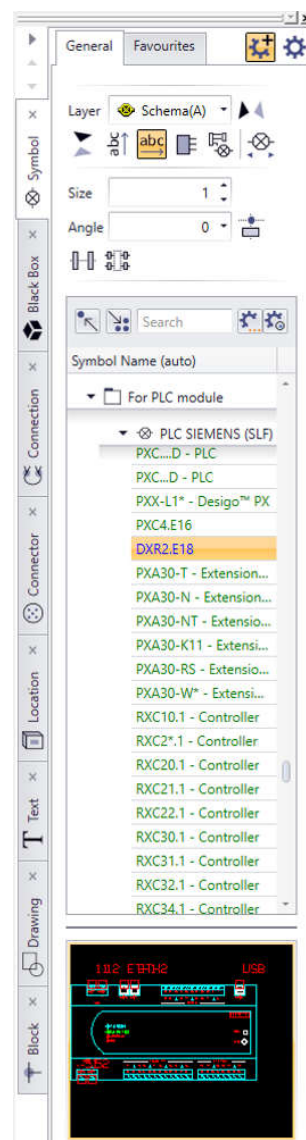
Рядок стану (Status Bar) містить важливу оперативну інформацію: шлях до поточного проекту, назву останньої виконаної команди, етапи поточної операції, а також координати курсора (X, Y). Одиниці вимірювання координат можуть змінюватися залежно від обраного режиму відображення.

Контекстні меню (Pop-up Menus) надають швидкий доступ до найбільш уживаних команд і властивостей. Їхній вміст залежить від контексту тобто від того, який об'єкт або область вибрано. Меню можна викликати клацанням правої кнопки миші на (рис. 7.6):

- панелях інструментів,
- областях навігаторів (explorers),
- області креслення (з вибраним або без вибраного об'єкта).



а)



б)

Рис. 7.5. Вікна Electrical Data Explorer з допоміжними вікнами (а) та палітра вставки (б)

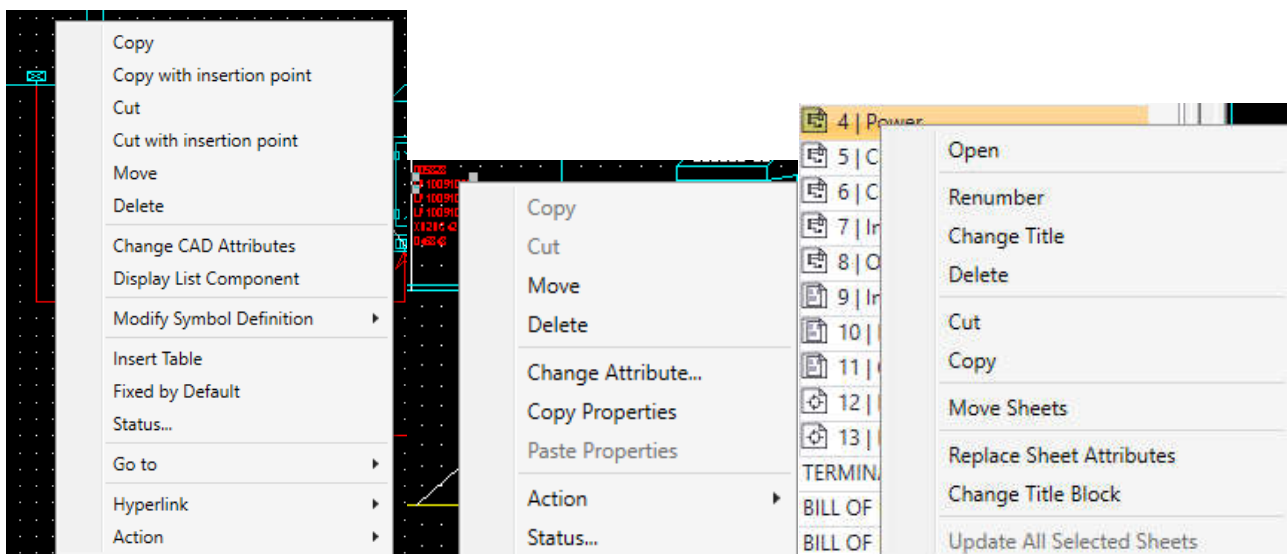


Рис. 7.6. Контекстні меню

У налаштуваннях програми можна обрати два режими вигляду контекстних меню: Classic (списком ліворуч або праворуч від точки клацання) та Round (радіальне меню, де команди розташовані по колу, а підменю концентричними дугами) (рис. 7.7).

Для зміни одиниць вимірювання відображення координат у рядку стану необхідно:

1. Навести курсор на координати X/Y у рядку стану.
2. Натиснути праву кнопку миші.
3. У спливаючому меню обрати потрібний режим (рис. 7.8).

Після зміни параметрів усі значення координат автоматично перераховуються, а в рядку стану відображаються нові одиниці.

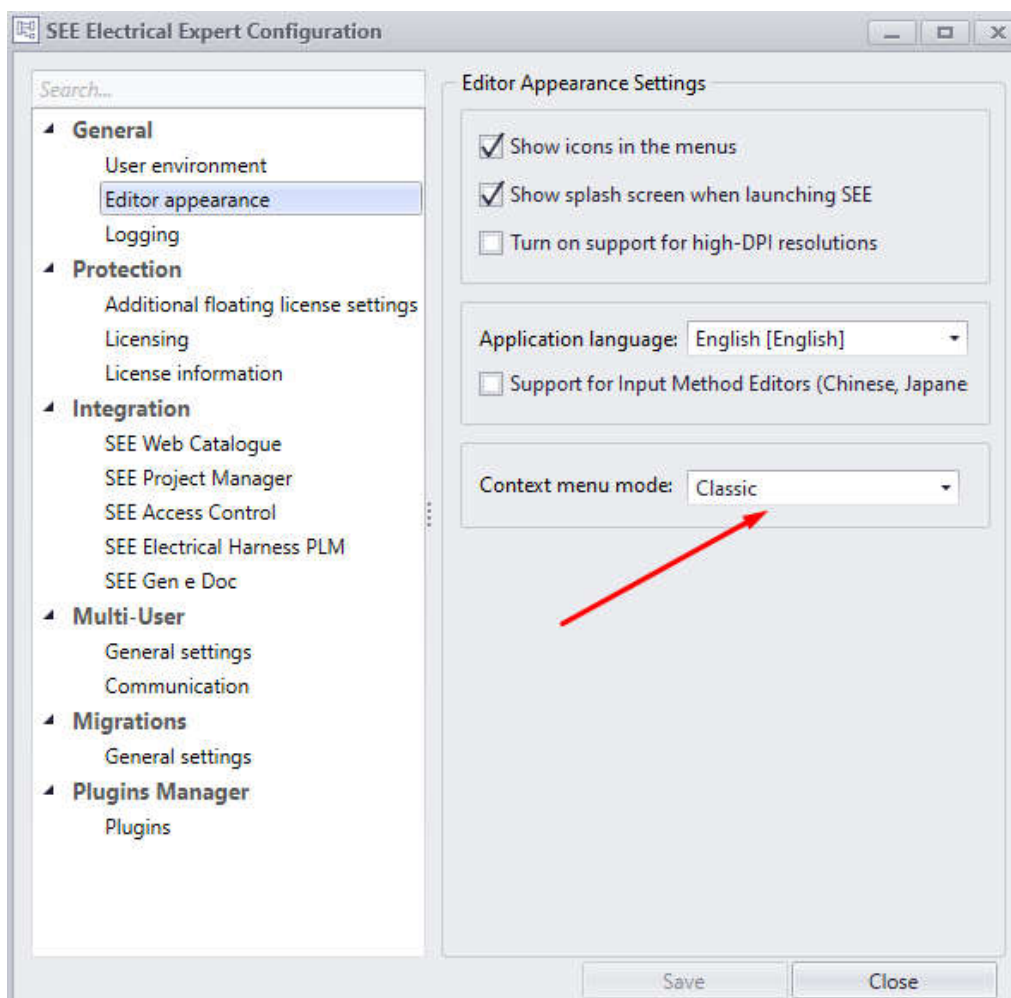


Рис. 7.7. Вікно програми налаштувань SEE Electrical Expert Configuration

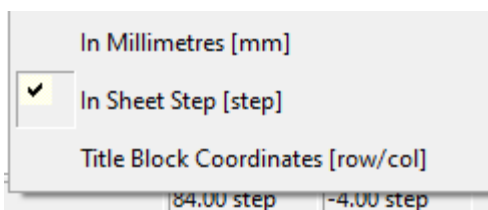
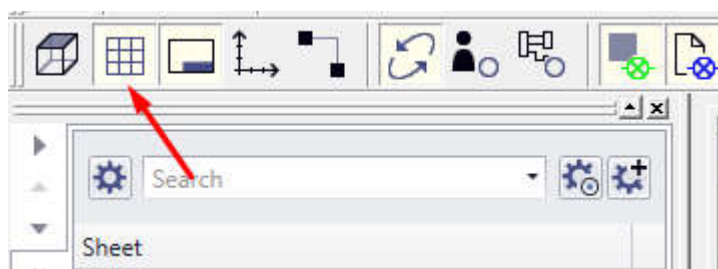


Рис. 7.8. Контекстне меню зміни одиниць вимірювання відображення координат

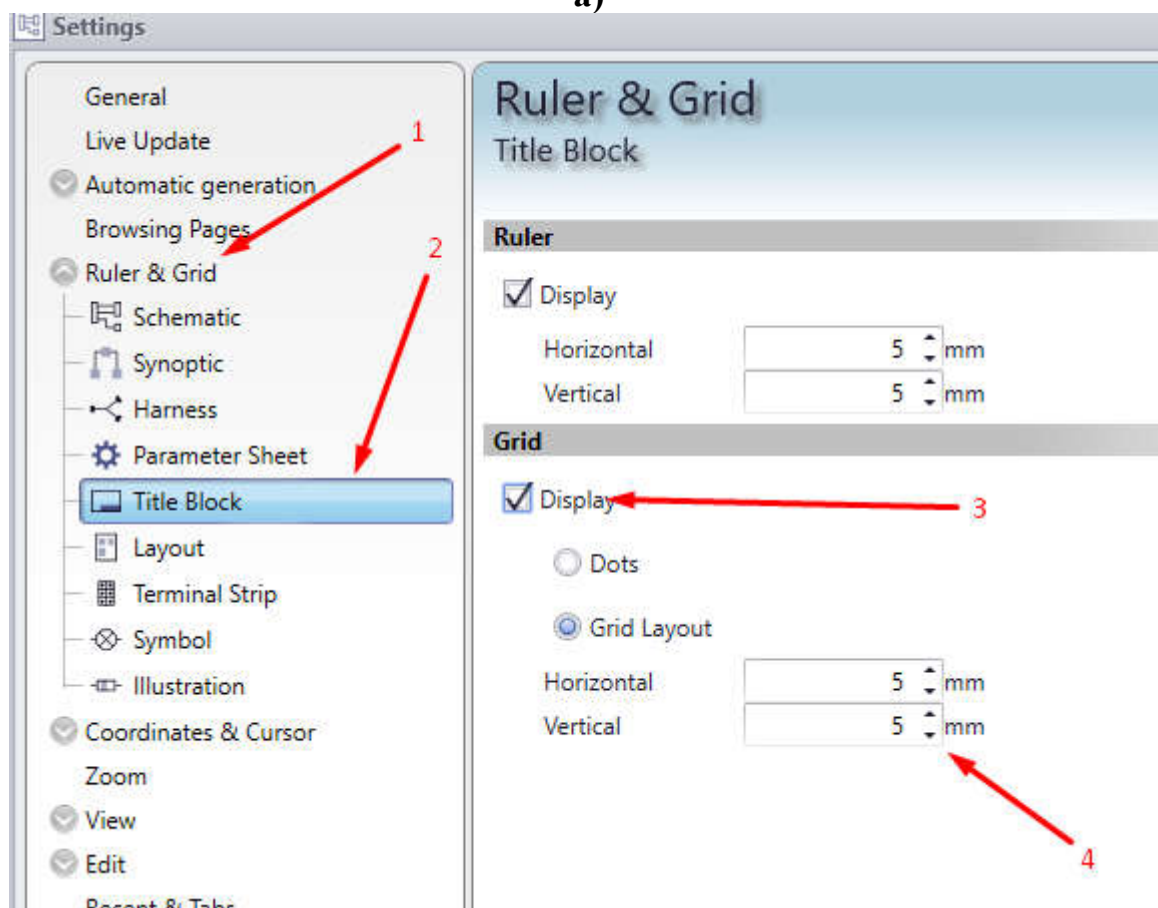
Сітка (Grid) в області креслення допомагає вирівнювати елементи та точно розміщувати компоненти схеми. Вона є візуальною допомогою і не друкується. Її можна вмикати або вимикати за допомогою іконки Grid на панелі Visibility (рис. 7.9а).

Для налаштування сітки необхідно відкрити Options → Settings обрати розділ Ruler / Grid та увімкнути параметр Display для потрібного типу аркуша. Задати горизонтальний та вертикальний крок сітки (за замовчуванням встановлено 5 мм) та підтвердити вибір ОК (рис. 7.9б).

Крім того, клавіша G прив'язує або відв'язує рух курсора до вузлів сітки, полегшуючи точне розміщення елементів.



а)



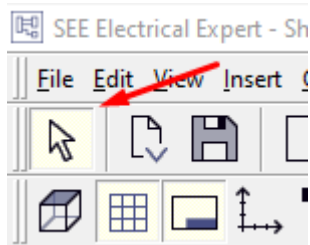
б)

Рис. 7.9. Налаштування параметрів сітки

Лінійка (Ruler) розташована у нижньому та лівому краї області креслення і відображає координати об'єктів по осях X та Y. Вона вмикається або вимикається через іконку Ruler у панелі Visibility. Для налаштування лінійки

необхідно відкрити Options → Settings → Ruler / Grid та увімкнути параметр Display (рис. 7.9б). Задати горизонтальний і вертикальний крок шкали (5 мм у метричній системі або 0.125 дюйма в імперській). Вибрати спосіб масштабування відповідно до координат титульного блока або заданого кроку аркуша.

Для вибору об'єктів необхідно переконатися, що активовано режим вибору.



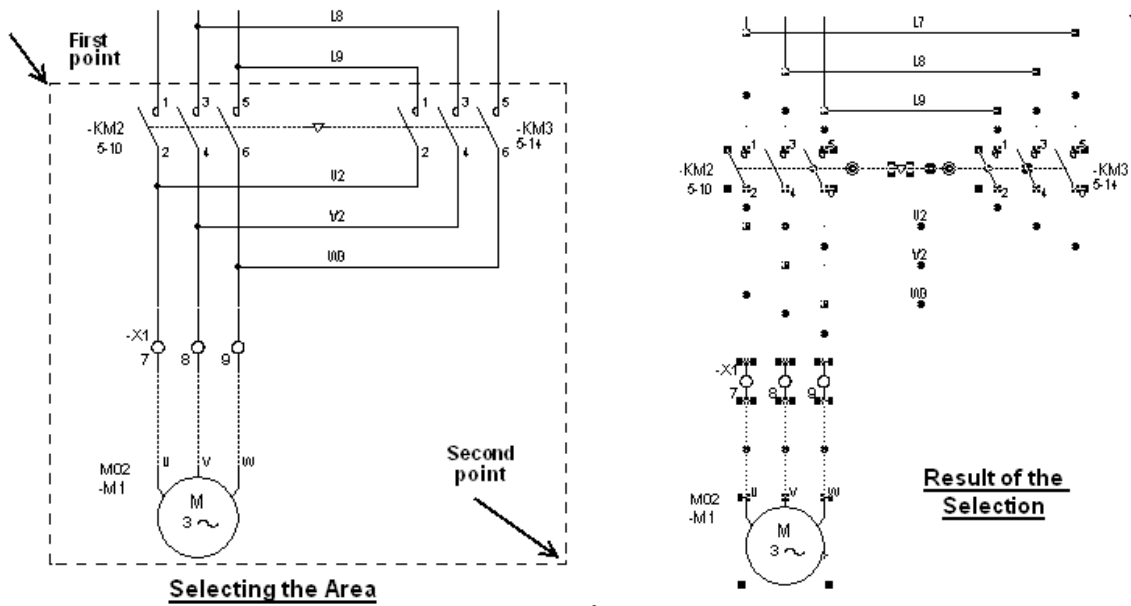
Вибір елементів здійснюється за допомогою окреслення області, у межах якої розташовані потрібні об'єкти. Кожен об'єкт має опорну (референсну) точку: для з'єднань це дві крайні точки, для текстів і символів це точка вставлення, а для креслень це точки побудови. Саме положення цих опорних точок визначає, чи потрапить елемент до вибору.

Існує два режими вибору залежно від положення опорних точок: розширений (Extended) (рис. 7.10а) і обмежений (Restricted) (рис. 7.10б). У розширеному режимі вибираються ті елементи, у яких хоча б одна опорна точка потрапляє до виділеної області. У разі обмеженого режиму вибираються лише ті об'єкти, усі опорні точки яких розміщені всередині області вибору. У кожному випадку користувач позначає ліву верхню точку області, потім перетягує курсор для задання правої нижньої межі.

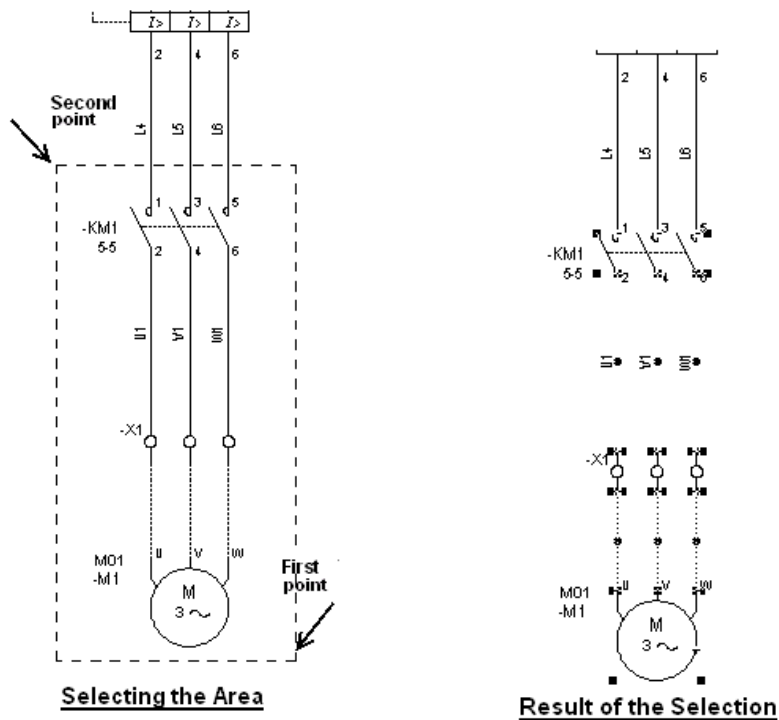
Під час виділення можна застосовувати фільтри, що дозволяють виключати певні типи об'єктів. Набір доступних фільтрів залежить від типу поточного аркуша. Наприклад, у схемах можна виключати символи, з'єднання, тексти, креслення або атрибути; у монтажних планах це обладнання, рейки, лотки, розміри тощо; у структурних аркушах це компоненти, гілки, захисти кабелів; у термінальних це клеми, провідники, перемички й розділювачі. Для цього в панелі інструментів потрібно обрати піктограми об'єктів, які не мають брати участь у виборі, а потім окреслити потрібну область.

Якщо потрібно вибрати кілька не суміжних елементів, достатньо клацнути перший об'єкт, утримуючи клавішу Shift, і по черзі обрати інші.

Під час вибору або переміщення графічних об'єктів автоматично активуються точки прив'язки (snap points), які виконують роль своєрідних «магнітів». Це значно полегшує точне позиціонування елементів коли курсор наближається до будь-якої опорної точки об'єкта, він автоматично «прилипає» до неї. Аналогічно, якщо об'єкт вставляється або переміщується поблизу іншого, відбувається автоматичне з'єднання за найближчими точками прив'язки.



a)



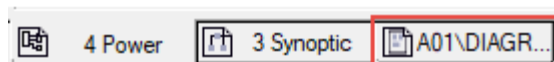
б)

Рис. 7.10. Вибір елементів розширений (Extended) (а) і обмежений (Restricted) (б)

Таким чином, процес вибору об'єктів у середовищі проектування передбачає використання різних режимів, фільтрів та інструментів прив'язки, що забезпечують точність і зручність роботи з графічними елементами.

У середовищі SEE Electrical Expert користувач може одночасно відкривати кілька аркушів, проте робота здійснюється лише з одним активним аркушем у певний момент. Активним вважається той аркуш, на якому виконуються поточні дії. Відкрити новий аркуш можна за допомогою відповідної команди меню File → Open Sheet або натискання відповідної піктограми на панелі інструментів.

Кожен відкритий аркуш відображається окремою вкладкою в нижній частині головного вікна програми.



Залежно від параметрів, вибраних у розділі Recent & Tabs вікна налаштувань, назва вкладки може містити лише номер і заголовок аркуша, або ж додатково ще й назву групи, до якої цей аркуш належить. Таким чином, користувач може самостійно визначити, наскільки деталізовано відображатиметься підпис вкладки. Попри можливість одночасного відкриття кількох аркушів, у стандартному режимі відображення на екрані показується лише один. Щоб перейти до іншого вже відкритого аркуша, достатньо клацнути відповідну вкладку внизу вікна.

Додатково, при натисканні правою кнопкою миші на вкладці аркуша з'являється контекстне меню, яке дозволяє перейменувати, видалити, перемістити або закрити вибраний аркуш. Це забезпечує зручну навігацію між документами та швидке керування їхнім розташуванням у робочому середовищі.

Також в програмі SEE Electrical Expert користувач має змогу одночасно відображати всі відкриті аркуші, розташовуючи їх у межах головного вікна в окремих робочих областях. Це дозволяє порівнювати схеми, перемикатися між ними або закривати окремі чи всі аркуші одночасно. Усі дії з візуалізацією аркушів виконуються за допомогою команд меню Window.

Для роботи з кількома відкритими аркушами доступні три режими розташування: каскадне, горизонтальне та вертикальне.

Window → Cascade – аркуші відображаються у вигляді каскаду, кожен у своєму вікні, при цьому вікна частково перекривають одне одного.

Window → Horizontal Tile – усі аркуші розташовуються горизонтально поруч без перекриття, що зручно для порівняння документів по вертикалі.

Window → Vertical Tile – аркуші розташовуються поруч вертикально, що дозволяє переглядати їх одночасно на вузькому екрані.

Кожне з відкритих вікон можна вільно переміщати або змінювати його розміри, підлаштовуючи під потреби користувача.

Для зручного керування відкритими документами використовується команда Window → List of opened sheets. Після її вибору відкривається діалогове вікно, де наведено перелік усіх відкритих аркушів.

У цьому списку можна:

- закрити окремий аркуш, вибравши його та натиснувши Close;
- закрити кілька аркушів одночасно, утримуючи клавішу Shift під час вибору;
- активувати потрібний аркуш натисканням кнопки Activate (доступна лише для одного вибраного аркуша);
- закрити всі відкриті аркуші натисканням кнопки Close All.

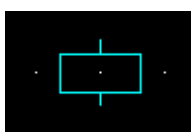
Таким чином, інтерфейс SEE Electrical побудований з урахуванням логіки проєктувальника та усі необхідні інструменти згруповані за функціями, а робоче середовище можна гнучко налаштувати, що спрощує роботу з великою кількістю схем і підвищує ефективність роботи користувача.

7.2. Базові операції в SEE Electrical.

Розглянемо операцію вставки умовних графічних позначень (символів). Вона здійснюється у режимі вставки в середовищі електротехнічного проєктування вставки графічних елементів (символів).

Символ це графічне зображення елемента або пристрою, який використовується в електричних схемах.

Наприклад, на рисунку подано умовне позначення реле.



Під час роботи із символами у середовищі проєктування користувач має можливість орієнтувати їх під різними кутами: 0°, 90°, 180° або 270°. З'єднання елементів у схемі здійснюється лише через спеціальні контактні точки символів, що забезпечує правильність побудови електричних з'єднань. Графічна частина символів створюється в окремому редакторі символів і не підлягає редагуванню безпосередньо у схемі.

Бібліотека символів постачається разом із середовищем Elec_IЕС і складається з набору файлів, що містять групи символів. Кожна група об'єднує символи одного типу пристроїв, таких як автоматичні вимикачі, контактори чи електродвигуни. Позначення групи формується з літерного префікса (тега) та назви типу пристрою. Наприклад, відповідно до стандарту ІЕС, літера Q використовується для позначення роз'єднувачів, тому група символів може мати опис "Q-Disconnecter 1-2-3-4P".

Під час наведення курсора на символ з'являється підказка (рис. 7.11), у якій відображається основна інформація про елемент: тег і його стан (вільний або заблокований), код обладнання, назва та родина, функціональне призначення або поведінка символу, а також наявність гіперпосилання, якщо воно передбачене.

Усі символи, що використовуються в середовищі Elec_IЕС програми SEE Electrical, відповідають останнім вимогам міжнародних стандартів ІЕС. Більшість із них побудовано на основі публікацій ІЕС 60617 та ІЕС 1082-1, які визначають правила графічного відображення електротехнічних елементів.

Для кожного символу у властивостях вказується посилання на відповідний стандарт у полі Description 2. Якщо у полі Description 1 назва починається зі знака зірочки (*), це означає, що такий символ є поширеним у практиці, однак не стандартизованим. Літерне позначення (тег), яке використовується для кожного символу, відповідає вимогам стандарту ІЕС 61346-2.

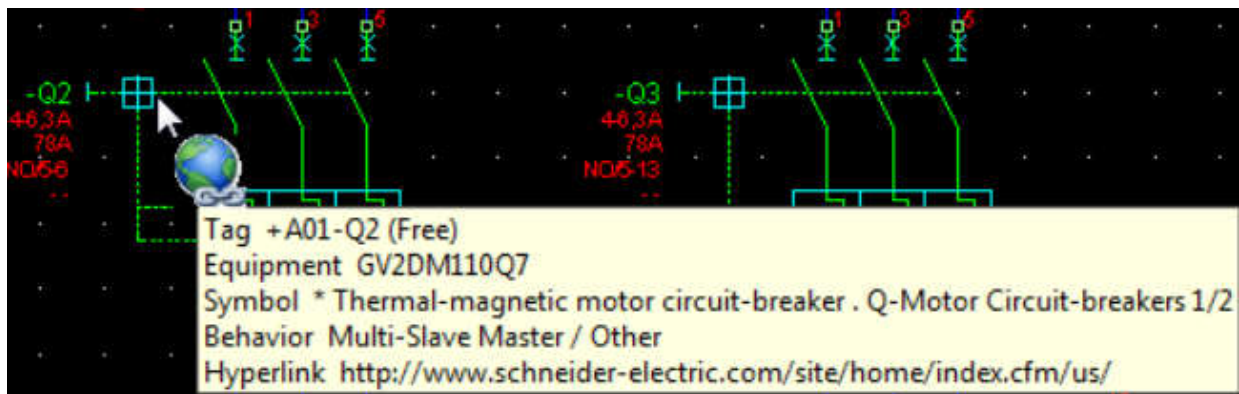


Рис. 7.11. Інформаційна підказка про елемент

Стандарт ІЕС 1082-1, який регламентує правила створення електротехнічної документації, вніс певні зміни у спосіб представлення елементів на електричних схемах. Зокрема, у статті 4.1.5 «Writing and writing direction» зазначено, що маркування клем пристроїв (наприклад, 13-14, А1-А2) тепер має розміщуватися вертикально й читатися знизу вгору (тобто під кутом 90°). Проте символи, що постачаються у стандартній бібліотеці програми, поки не враховують цю зміну.

Інша частина стандарту, стаття 3.3 «Documentation structure», визначає правила формування тегів для позначення функцій і розташування пристроїв. Відповідно до цього стандарту, позначення електричного пристрою має починатися зі знака «-», тоді як символи «+» і «=» використовуються на вищих рівнях: символ «+» – для позначення місця встановлення (location), а «=» – для позначення обладнання або машини (function).

Слід зазначити, що лише шаблон проекту “Tagging by ІЕС” у програмі SEE Electrical підтримує застосування цих правил повністю.

Крім того, програма дозволяє дотримуватися інших міжнародних стандартів, зокрема: ІЕС 60757 – колірне кодування провідників; ІЕС 60050-195 – позначення провідників і клем; EN 61082 – підготовка електротехнічної документації; EN/ІЕС 61346 та EN/ІЕС 61666 – правила позначення установок і систем обладнання.

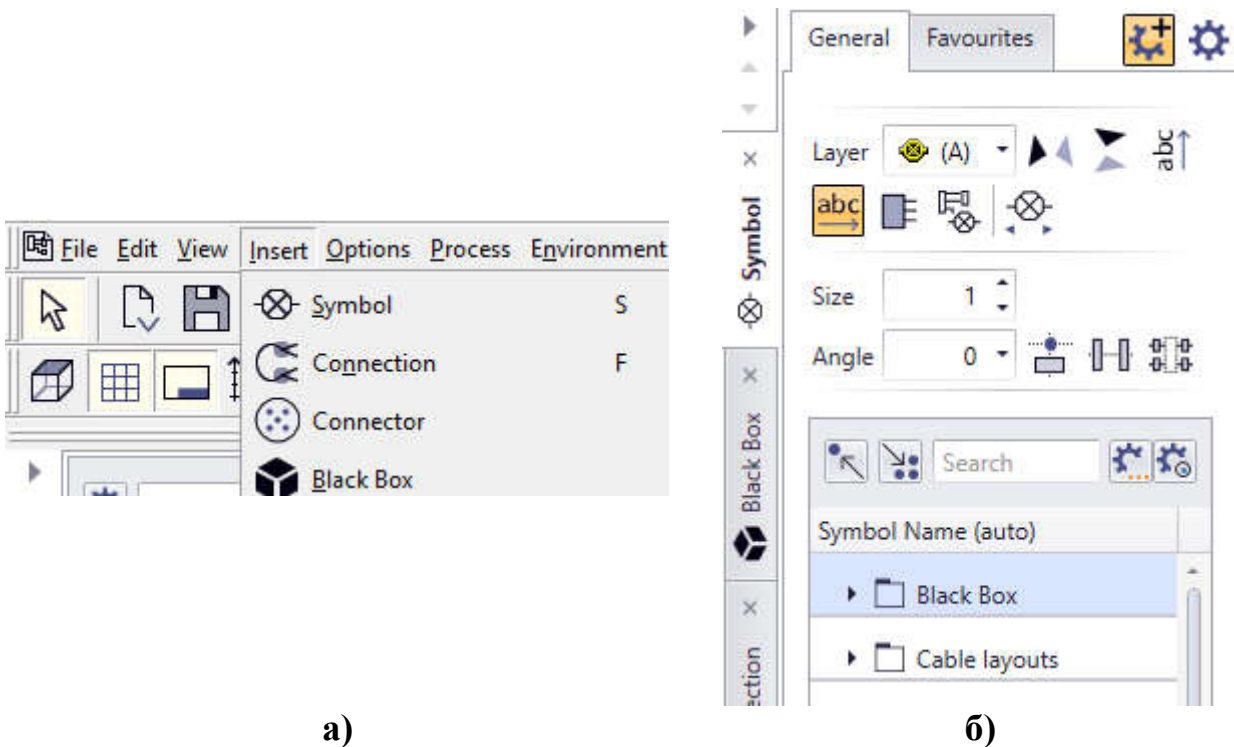
Для вставка символів у схему використовується команда Insert → Symbol (або клавіша S) з головного меню або відповідна піктограма на панелі інструментів (рис. 7.12а).

Вибір потрібного символу здійснюється за допомогою панелі вставка (Insertion Palette) (рис. 7.12б), у якій відображаються графічні зображення та короткі описи символів, доступних у проекті або середовищі Elec_ІЕС. Щоб відкрити цю панель, можна скористатися командою меню View → Insertion Palette. Перед вставленням символу у схему користувач має можливість змінити його основні параметри: розмір, кут повороту, шар вставлення, а також відобразити його у дзеркальному вигляді: горизонтально або вертикально. Усі ці властивості задаються у вкладці Symbol панелі Insert.

Основний порядок вставки символу:

1. Відкрити палітру вставки (View → Insertion Palette).
2. Після цього з’явиться вікно Insertion Palette, яке містить вкладки для вибору елементів.

3. Перейти на вкладку Symbol.
4. Вибрати потрібний символ у вкладці General або Favourites.
5. У вкладці General містяться стандартні бібліотеки елементів, а у Favourites – користувацькі або часто використовувані символи.
6. Задати властивості символу (положення, орієнтацію, масштаб, параметри з'єднань тощо).
7. Вказати місце вставки на аркуші, клацнувши у потрібній точці.
8. Ввести CAD-властивості символу (позначення, опис, нумерацію тощо).



а) б)
Рис. 7.12. Операція вибору символу

Вставка символу можливе лише у тому випадку, якщо він не захищений.

Символи у SEE Electrical можуть вставлятися двома способами: *вільно* на аркуші або *поверх* уже накреслених з'єднань. Якщо елемент розміщується безпосередньо на лінії з'єднання, характер його взаємодії з нею залежить від режиму вставки проєкту, типу символу, а також типу та групування з'єднувальних точок, які контактують із цією лінією.

Під час вставки символ автоматично розриває сегмент з'єднання, створюючи логічний розрив у схемі (рис. 7.13а). Однак, залежно від налаштувань, розрив може стосуватися лише певних елементів наприклад, дротів, але не сигналів.

Якщо точка вставки символу збігається з лінією з'єднання, програма автоматично орієнтує символ у правильному напрямку відносно цієї лінії, що полегшує роботу з розміщення елементів на схемі (рис. 7.13б).

Нейтральні символи це елементи, у яких атрибут Wire Connection має значення *neutral*. Такі символи не розривають сигнали або проводи, а зберігають їхню безперервність у схемі.

Якщо з'єднувальні точки належать до однієї групи, вони також зберігають безперервність сигналів між собою.

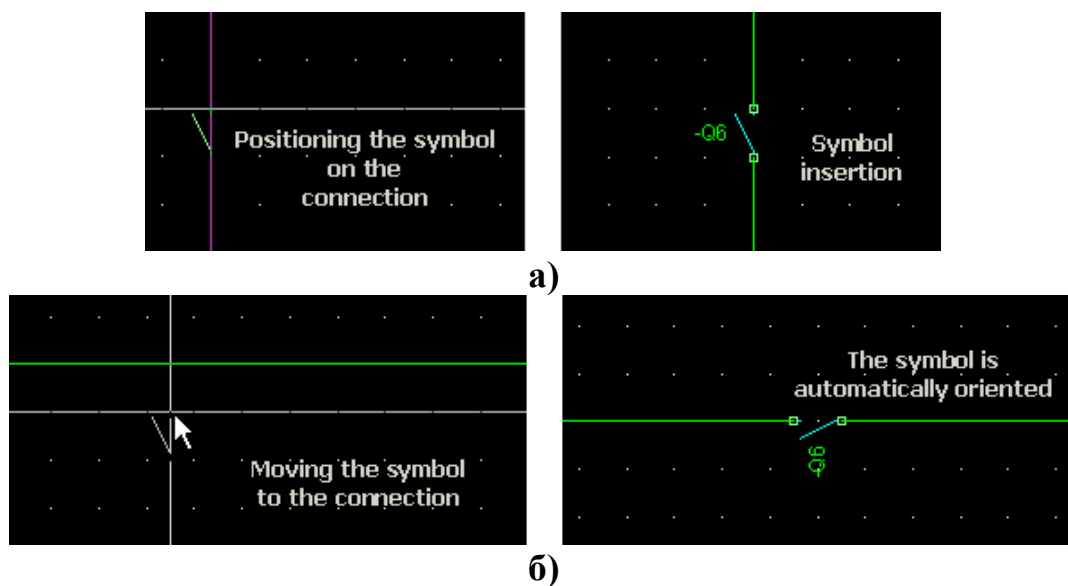


Рис. 7.13. Вставка символу

У разі, коли атрибут Wire Connection точки має значення «Wire goes through the connection point» (тобто «провід проходить через точку з'єднання»), безперервність дроту зберігається, і така точка не створює розриву. Зазвичай такі точки також групуються між собою. Якщо ж атрибут має значення «Wire extremity is connected at connection point» («кінець дроту під'єднано до точки з'єднання»), тоді дріт переривається в цій точці, формуючи нове електричне з'єднання.

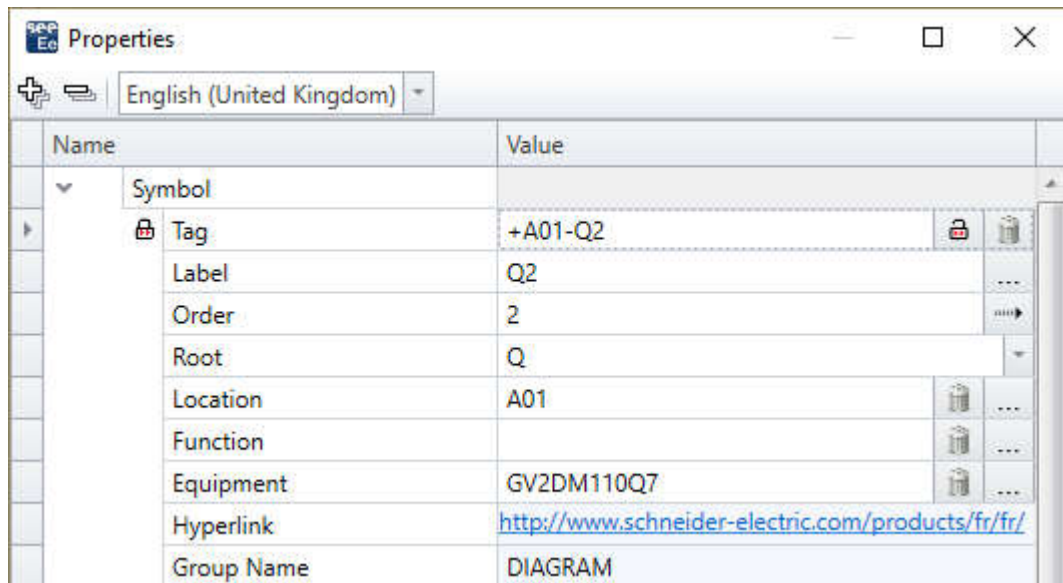
Функція Auto-connection mode дозволяє автоматично підключати символи до існуючих з'єднань під час вставки або переміщення. При цьому програма автоматично визначає контактну точку і тип з'єднання відповідно до заданих характеристик символу. Для активації режиму Auto-connection необхідно обрати: Insertion Palette → Advanced Settings → Auto-connection mode, або Options → Settings → Edit → Schematic sheet → Symbol auto-connection = Yes.

Кожен символ у SEE Electrical має набір **CAD-атрибутів**, які визначають як технічні характеристики елемента, так і правила його відображення на електричній схемі. Ці атрибути виконують подвійну функцію, вони не лише містять інформацію про сам символ, а й керують його поведінкою під час автоматизованих процесів проектування.

Видимість та доступність властивостей у вікні Properties (Властивості) регулюється параметрами CAD Settings Method. Змінювати ці параметри можуть лише користувачі, які мають відповідні права доступу та редагування.

Коли користувач вставляє символ у схему, і активним є режим маркування «User», автоматично відкривається вікно Properties, де можна задати всі необхідні атрибути елемента (рис. 7.14). Для вже розміщених символів це вікно викликається через команду Edit → Change CAD Attributes, або за допомогою

контекстного меню (Change CAD Attributes) або подвійним клацанням на самому символі.



Name	Value
Symbol	
Tag	+A01-Q2
Label	Q2
Order	2
Root	Q
Location	A01
Function	
Equipment	GV2DM110Q7
Hyperlink	http://www.schneider-electric.com/products/fr/fr/
Group Name	DIAGRAM

Рис. 7.14. Атрибути символу

Усі атрибути відображаються у вигляді таблиці з полями, які можуть бути редагованими або нередатованими.

Нередатовані поля зазвичай підсвічуються сірим кольором і містять службову інформацію: назву групи, рівень групування, ім'я аркуша, координати вставки (рядок і стовпець).

Редаговані поля дозволяють користувачеві вводити значення вручну або скористатися додатковими кнопками для: покрокового збільшення чи зменшення значення атрибута; вибору значення зі списку; відкриття додаткового діалогового вікна для уточнення параметрів; видалення поточного значення.

До ключових атрибутів, які визначають логіку роботи та ідентифікацію елемента (символа), належать:

- Tag (Тег) – унікальний ідентифікатор елемента у проєкті.
- Label (Позначення) – текстове маркування, яке відображається поруч із символом на схемі.
- Location (Розташування) – визначає фізичне або логічне місце елемента в системі.
- Function (Функція) – описує функціональне призначення компонента.
- Equipment Code (Код обладнання) – використовується для зв'язку між елементом і конкретним пристроєм чи апаратурою.
- Extra Length (Додаткова довжина) – застосовується при розрахунку кабелів або проводів.
- Tag Modification for Masters and Slaves Option – визначає поведінку тегів для головних і підлеглих елементів у складних структурах.
- Hyperlink Visibility – керує відображенням гіперпосилань, якщо вони додані до символу.
- Equipment Tag – дозволяє пов'язати елемент з конкретним обладнанням у базі даних.

- Notes (Примітки) – поле для додаткових коментарів або технічної інформації.

Режим Tagging Mode визначає спосіб введення та призначення CAD-атрибутів символів під час вставки або після розміщення елементів на схемі.



У режимі Blank символ додається без властивостей і тегів, що зручно для попереднього компонування схеми. Режим Next автоматично призначає тег згідно з заданим форматом нумерації, дозволяючи швидко розміщувати елементи без ручного введення даних. У режимі User під час вставки відкривається вікно властивостей, де користувач одразу вводить усі необхідні атрибути, що забезпечує точність і структурованість проекту. Додаткова опція Not assign numbers at pins дозволяє створювати порожні контакти або клеми, які не враховуються під час присвоєння кодів обладнання.

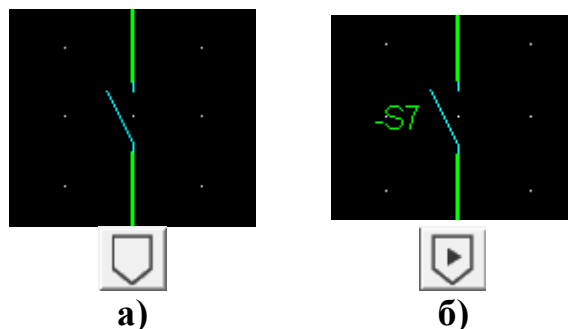
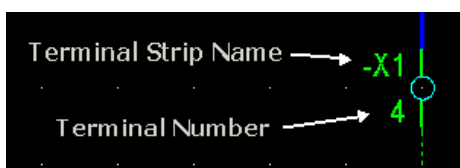


Рис. 7.15. Режим Blank (а) та Next (б) додавання символів

У проєктах типу Wire Connectivity може застосовуватися налаштування Do not tag the cables/wires on connection insertion, яке вимикає автоматичне маркування проводів і кабелів, зберігаючи «порожній» режим лише для цих елементів. Таким чином, система Tagging Mode забезпечує гнучкість і контроль процесу маркування, дозволяючи адаптувати роботу до рівня деталізації та вимог проєктування.

Терміни **Terminals** позначають спеціальні символи, що використовуються для створення аркушів клемних колодок (Terminal Strip Sheets) і організації з'єднань між різними частинами електричної схеми. Кожна клема належить до певної групи або колодки, яку потрібно вказати під час вставки. За замовчуванням група терміналів позначаються літерами, що починаються з X, наприклад X1, X2 тощо.

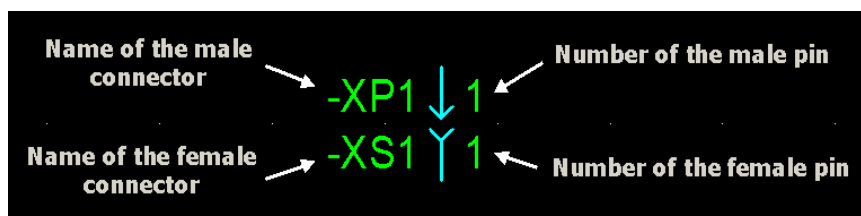


При додаванні терміналу необхідно ввести його CAD-властивості, відповідно до активного Tagging Mode. Якщо використовується режим User, відкривається вікно властивостей, де задаються параметри конкретного елемента. На схемах можна також вставляти клеми типу «For Information», які представляють фізичні клеми, що вже існують або будуть створені в іншому аркуші. Якщо в процесі додавання інформаційної клеми вказано номер, який ще не існує, система автоматично створює відповідну фізичну клему та зв'язує її з «інформаційною» копією.

Вікно властивостей терміналу складається з двох частин, відомостей про клемну колодку та про саму клему, що забезпечує зручність у керуванні їх структурою. Крім стандартних клем, See Electrical дозволяє додавати багаторівневі (multi-level) клеми, клеми без коду обладнання, запасні, заземлювальні, екранувальні та спеціальні клеми, а також розділювачі та перемички.

Роз'єми (Connectors) використовуються для моделювання електричних з'єднань між різними частинами схеми та формування Connector Sheets та аркушів із розташуванням контактів. Роз'єми складаються з контактів двох типів таких, як штекер (male) і гніздо (female), які належать до різних конекторів, що забезпечує коректне логічне з'єднання між ними.

Додавання контактів (pins) здійснюється через Insertion Palette, у вкладці Connector, де користувач обирає сімейство символів (за замовчуванням їхні назви починаються з літери X) і вказує, до якого саме конектора належить контакт. Після вставки курсор показує довідкову інформацію про контакт, що допомагає при візуальній перевірці схеми.



У схемах можуть використовуватись також «Pins For Information», тобто допоміжні контакти, які представляють фізичні контакти, використані в інших аркушах. Такі елементи дозволяють відображати логічні з'єднання між частинами схеми, навіть якщо фізичний контакт ще не створено. Якщо контакт «For Information» вставляється без існуючого фізичного піну, програма автоматично створює відповідний елемент при підтвердженні властивостей.

Система підтримує різні режими вставки роз'ємів: поконтактну вставку (Pin by Pin) або вставку блоку з контактами (Box and Pin). Також можна створювати порожні (spare) контакти, визначати кількість пінів, присвоювати коди обладнання (Equipment Codes) та вставляти піни без номерів.

Black Box у SEE Electrical це універсальний елемент у вигляді прямокутника, який використовується для створення умовних позначень пристроїв, блоків або функціональних частин схеми без необхідності

попереднього створення окремого символу в бібліотеці. Користувач самостійно визначає розмір прямокутника, обираючи дві точки на кресленні, а відстань між контактними точками (за замовчуванням 5 мм) встановлюється відповідно до налаштувань кроку аркуша.

За своїм призначенням чорні блоки поділяються на *зовнішні* (External) та *внутрішні* (Internal).

Зовнішній тип використовується як звичайний символ наприклад, для швидкого створення специфічного графічного елемента без відкриття редактора символів.

Внутрішній тип застосовується для побудови спрощених або функціональних схем, що можуть бути пов'язані з деталізованими схемами на інших аркушах. Такий підхід дозволяє реалізувати структурне проектування, коли загальний блок системи представлений умовним символом, а його повна схема на окремій сторінці.

Перед вставкою користувач задає параметри Black Box – тип, спосіб побудови (фіксований або вільний розмір), інтервал між контактами, стиль лінії, а також поведінку (Master, Unique, Multiple або Slave), що визначає логіку його взаємодії з іншими елементами. У режимі вставки (Insert → Black Box або з панелі інструментів) система автоматично розраховує координати та розташовує контактні точки, які згодом можна під'єднати до проводів чи інших символів.

Користувач також може активувати режим автоматичного під'єднання в цьому випадку з'єднання між контактними точками Black Box та іншими елементами створюються автоматично, якщо вони розміщені на одній лінії.

Equipment це елемент, який використовується для створення кабельних планів і відображення інформації про з'єднання між елементами схеми. Після розроблення електричних діаграм користувач може переходити до формування схем з'єднань, де кожному символу присвоюється код обладнання (Equipment Code). Цей код визначає належність елемента до конкретного пристрою або компонента в системі.

Символ Equipment містить дані про з'єднання: маркування сигналів, кабелів, джерел і точок призначення. Він виступає графічним представленням фізичного пристрою або його частини в кабельному плані. Зв'язок між символом у принциповій схемі та його відповідником у плані обладнання забезпечується за допомогою мітки (Label).

Вставка обладнання здійснюється через команду Insert → Equipment, після чого відкривається вікно вибору, де користувач обирає клас пристрою та відповідний Equipment Code. Кожен клас повинен містити поле з функцією Symbol, у якому визначається, які графічні елементи або блоки пов'язані з цим кодом. При цьому допускається використання кількох варіантів символів, заданих через синтаксис «Symbol name».«Family name»*, або блоків «blockname.blk». Надалі користувач може послідовно переглядати всі доступні графічні форми обладнання, обираючи потрібну за допомогою колеса миші. Якщо активовано режим User Tagging, програма відкриває вікно властивостей, у якому можна ввести CAD-атрибути для нового символу. Після підтвердження

даних система автоматично переходить до наступного елемента, якщо вибраний код обладнання містить кілька пов'язаних символів.

Graphical Locations це спеціальні символи, які позначають прямокутну зону на схемі для групування та логічного розташування елементів. Їх розмір визначається користувачем під час вставки двома точками, що утворюють межі області.

Усі основні (master) символи, розміщені всередині цієї зони, автоматично успадковують ім'я розташування (location name). Це ім'я відображається в нижньому лівому куті прямокутника, що полегшує ідентифікацію елементів на схемі. Графічна локація визначається двома параметрами: назвою та стилем лінії (пунктирна, тонка тощо), якою окреслено межі області. Їх можна створити через меню Insert → Location, після чого активується вкладка Location у палітрі вставки. В цій вкладці користувач може: вибрати або створити нове ім'я локації; задати графічний стиль рамки; визначити шар (layer), у який буде вставлено символи; за потреби може автоматично вставити клеми (terminal strips) у місцях перетину нової локації з наявними з'єднаннями.

Signal Regroupments (Net Groups) це спеціальні символи, призначені для визначення типу сигналу (екранований, скручений, коаксіальний, трьохекранований тощо) та групування кількох сигналів у єдину логічну мережу. Вони містять точки з'єднання, через які можна об'єднувати сигнали в межах функціональної схеми. Такі символи створюються з поведінкою «NetGroup» і однією з функцій: Netgroup Other, Shielded, Shield, Plait/Shield, Coaxial або Triaxial. Вони використовуються для функціональних діаграм установки, де показано електричне обладнання та сигнали без деталізації кабельних з'єднань.

Під час створення символів Net Group визначаються спеціальні користувацькі атрибути, які відповідають системним атрибутам сигналів, заданим у методі функціональної діаграми. Ці атрибути синхронізуються автоматично зміни, внесені у властивості символу, відразу передаються відповідним сигналам у проекті. Інформація про сигнали доступна для редагування у вікні властивостей (Properties) символу NetGroup, а також може бути експортована у формат .xml для подальшої обробки або документування.

Під час вставки символів SEE Electrical виконує контроль узгодженості: кожен сигнал може бути прив'язаний лише до одного типу (наприклад, або екранованого, або скрученого), що гарантує точність і структурованість схеми.

Connections це логічні та графічні елементи, які описують електричні зв'язки між символами на схемі. Існує три основні поняття: Connection (з'єднання) логічний провід, що з'єднує символи або символи з точками розгалуження; Segment (сегмент) мінімальна частина геометрії з'єднання, створюється між двома точками (початком і кінцем), які зазвичай прив'язані до точок з'єднання символів; Signal (сигнал) тип електричного значення, що визначає єдину електричну мережу, утворену групою з'єднань. При наведенні курсора на з'єднання відображаються його основні властивості, тобто тег, тип, напруга, переріз тощо. Ці параметри також доступні у вікні властивостей (Properties), де користувач може переглядати та змінювати CAD-атрибути з'єднання. Типові значення для параметрів, як-от тип, калібр чи ізоляційна

категорія, задаються у Signals Method, що дозволяє стандартизувати характеристики з'єднань у проєкті. При проєктуванні SEE Electrical підтримує автоматичну нумерацію та тегування з'єднань, згідно з установленими форматами нумерації.

Strands or Cables це елементи, що дозволяють відображати електричні з'єднання між пристроями за допомогою проводів або кабелів у схемах (Schematic Sheets) та аркушах клемних з'єднань (Terminal Strip Sheets). Між кабелями, зображеними на цих аркушах, існує взаємна інтерактивна відповідність, за умови, що на схемі попередньо вставлені клемні колодки. Це забезпечує узгодженість між принциповими та монтажними схемами. На одному з'єднанні може бути розміщено кілька маркерів проводів або кабелів, які мають спільні користувацькі атрибути. Ці атрибути автоматично поширюються (propagate) між усіма символами, вставленими на одному проводі чи кабелі. Таким чином, сам провід або кабель не має власних атрибутів, їх властивості визначаються через символи, що його позначають. При повторному вставленні маркерів збережені значення атрибутів автоматично відновлюються. CAD-властивості кабелю та його жил (strands) відображаються у спільному вікні Properties, де для кожної жили створюється окрема секція. Кількість та назви жил залежать від обраного коду обладнання (Equipment Code). Деякі параметри можуть бути визначені цим кодом і не підлягають зміні вручну. Якщо не всі жили потрібні в схемі, користувач може обрати лише необхідні. Під час наведення курсора на кабель відображається коротка інформація, тег, код обладнання, ім'я, режим вставлення, функція або тип поведінки.

Equipment Illustrations це графічні зображення елементів електрообладнання, що використовуються для створення компоновочних схем (Layout sheets). Вони відображають реальні компоненти, такі як шафи (Enclosures), основне та допоміжне обладнання (Main and Additional Equipment), рейки (Rails) та кабельні канали (Wireways). Ці елементи дозволяють будувати точні тривимірні або площинні зображення монтажних схем, на яких візуально показано розташування пристроїв у шафі чи на панелі. Кожен елемент може мати власні CAD-властивості, такі як розміри, позиція, напрямок монтажу та зв'язок із відповідним обладнанням у принциповій схемі.

Drawings використовуються для створення додаткових графічних елементів на робочих аркушах, наприклад таких як прямокутники, кола, стрілки, зображення чи інші геометричні об'єкти. Доступні типи креслень залежать від типу активного аркуша (схема, компоновка, звіт тощо). За допомогою інструментів із вкладки Drawing користувач може додавати допоміжну графіку, створювати розмірні лінії, виконувати штрихування або вставляти об'єкти та зображення для ілюстрації схем. Такі елементи не впливають на логічну структуру електричного проєкту, але забезпечують його наочність, оформлення та відповідність вимогам стандартів.

В програмі SEE Electrical існує можливість роботи з блоками, стандартними схемами та макросами, що значно спрощує створення та повторне використання типових фрагментів схем. Блок це частина електричної схеми (наприклад, пускова схема двигуна або димер), збережена як єдиний об'єкт, який можна

багаторазово вставляти у проекти. Стандартна схема це повністю готовий аркуш із попередньо визначеним вмістом, титульним блоком та атрибутами, який можна вставляти в інші проекти того самого типу. Макрос подібний до блоку, але є більш гнучким бо він може мати кілька варіантів представлення, змінюваний вміст та набір змінних для визначення параметрів. Об'єкти, що входять до складу макросу (символи, з'єднання, кабелі, тексти), залишаються частиною цього макросу, а при його виборі підсвічуються всі елементи, що до нього належать. Керування блоками, стандартними діаграмами та макросами здійснюється через вікно Block Explorer, доступне з меню Environment → Block Explorer або з вкладки Block у палітрі вставки.

В програмі SEE Electrical передбачено велику кількість різноманітних інструментів і налаштувань, які забезпечують гнучкість і точність у створенні електричних схем та креслень. Однак у цьому посібнику розглянуто лише основні можливості, необхідні для базового користування програмою. Інші, більш розширені функції та спеціальні опції будуть докладно пояснені в окремому посібнику та на подальших лекціях, присвячених поглибленому вивченню SEE Electrical.

7.3. Створення нового проекту в SEE Electrical.

Створення нового проекту є початковим етапом роботи, який визначає структуру, тип та параметри майбутньої електричної схеми. Користувач може створити проект у двох режимах: Connection Mode (CM) для схем з'єднань (File → New Project...), або Wire Mode (WM) для проектів із покабельним з'єднанням (File → New Project with type «Wire connectivity»...). Відповідно, файли мають розширення .seeproj та .seewproj.

Новий проект можна створити як у локальній папці, так і в репозиторії (у багатокористувацькому режимі). Під час створення відкривається діалогове вікно, де задаються назва, місце збереження та шаблон проекту (рис. 7.16). Шаблони (.seetpl або .seewtpl) містять попередньо налаштовані параметри та структуру, що відповідає обраному середовищу, наприклад Elec_IЕС.

Залежно від обраного шаблону, система може автоматично визначати правила нумерації та маркування елементів (за всім проектом, по аркушах, групах або місцях розташування). Якщо користувач обирає порожній шаблон, усі параметри проекту задаються вручну через вкладку Properties.

Після підтвердження вибору програма створює новий проект, який відображається у Project Explorer, а при потребі користувач може одразу перейти до редагування методів чи налаштувань.

Подальші налаштування та виконання креслень для розробки проектною документації розглядаються на практичних /лабораторних роботах з дисципліни «Інформаційні комп'ютерні технології для інженерії систем керування».

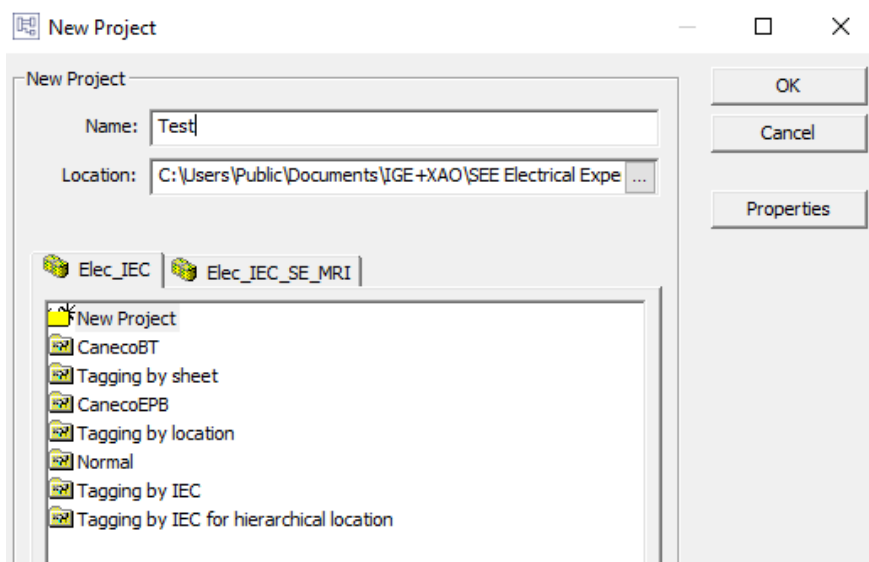


Рис. 7.16. Вікно створення нового проекту

Висновки за розділом.

1. У розділі розглянуто інтерфейс користувача програми SEE Electrical, який забезпечує інтуїтивне керування, зручний доступ до інструментів і швидке виконання базових операцій. Програма дозволяє ефективно створювати, редагувати та документувати електричні схеми, що значно спрощує роботу інженера-розробника.
2. Описано основні етапи створення нового проекту та засоби автоматизації, які допомагають скоротити час розробки та мінімізувати помилки. Програмне середовище SEE Electrical забезпечує точність, стандартизацію та зручну інтеграцію електричних і автоматизаційних систем у єдиний проект.

Питання для самоперевірки.

1. Які основні складові має інтерфейс користувача програми SEE Electrical?
2. Яке призначення головної панелі інструментів та панелі властивостей у середовищі SEE Electrical?
3. Як відкрити або створити новий проект у SEE Electrical?
4. Які типи діаграм можна створювати у середовищі SEE Electrical?
5. Яким чином здійснюється вставка символів електричних компонентів у схему?
6. Які параметри можна налаштувати для символів перед їх вставленням у схему?
7. Як зберігається структура проекту та взаємозв'язки між його сторінками?
8. Які основні інструменти використовуються для редагування елементів електричної схеми?
9. Яке призначення має функція Symbol Properties у програмі SEE Electrical?
10. Як здійснюється перевірка правильності з'єднань та створення звітів у SEE Electrical?

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Antonsen, T. M. Kickstart PLC Programming: Design and Build Scalable Control Systems Using IEC 61131-3. Industrial Press, 2023. 412 p. ISBN 979-8-218-22665-9.
2. Antonsen, T. M. PLC Controls with Ladder Diagram (LD): IEC 61131-3 and Introduction to Ladder Programming. Denmark: MyAutomation, 2021. 252 p. ISBN 978-87-7691-972-6.
3. Antonsen, T. M. PLC Controls with Structured Text (ST): IEC 61131-3 and Best Practice ST Programming. 3rd ed. Denmark: MyAutomation, 2021. 288 p. ISBN 978-87-7691-973-3.
4. Bailey D., Wright E. Practical SCADA for industry. Amsterdam: Elsevier, 2003. 288. ISBN 978-0-7506-5805-8.
5. Buchanan, W. J. Industrial Control Systems Security and Resilience. Springer, 2022. 356 p. ISBN 978-3-030-96056-5.
6. Hanssen D. H. Programmable logic controllers: a practical approach to IEC 61131-3 using CODESYS. Chichester, UK ; Hoboken, New Jersey: John Wiley & Sons, 2015. ISBN 978-1-118-94922-1.
7. Hughes T. A. Measurement and control basics. Research Triangle Park, NC: ISA-The Instrumentation, Systems, and Automation Society, 2002. 371. ISBN 978-1-55617-764-4.
8. Iniewski K. Smart Sensors for Industrial Applications. Boca Raton: CRC Press, 2016. ISBN 978-1-4665-6811-2.
9. Lathi B. P. Linear Systems and Signals, Second Edition. New York: Oxford University Press, 2005. 689.
10. Macdonald D., Reynders D. Practical Modern SCADA Protocols: DNP3, 60870.5 and Related Systems. 2004. 548.
11. Practical Instrumentation for Automation and Process Control for Engineers and Technicians. IDC, 2004. 375.
12. Radvanovsky, R., Brodsky, J. Handbook of SCADA/Control Systems Security. 2nd ed. Boca Raton: CRC Press, 2016. 368 p. ISBN 978-1-4987-2816-5.
13. Sands N. P., Verhappen I. A Guide to the automation body of knowledge. 2018. ISBN 978-1-941546-91-8.
14. Thomas M. S., McDonald J. D. Power system SCADA and smart grids. Boca Raton, Florida: CRC Press, 2015. 336 c. ISBN 978-1-4822-2674-4.
15. Thomas, M. S., McDonald, J. D. Power System SCADA and Smart Grids. 2nd ed. Boca Raton: CRC Press, 2021. 382 p. ISBN 978-0-367-51721-5.
16. Verhappen, I. A., Sands, N. P. A Guide to the Automation Body of Knowledge. 3rd ed. Research Triangle Park, NC: ISA, 2023. 430 p. ISBN 978-1-955705-16-8.
17. Wickert M. Signals & Systems For Dummies. 2013. 387.

Грищенко Володимир Олександрович
Степаненко Сергій Петрович
Грищенко Ірина Юріївна

КОМП'ЮТЕРНО -ІНТЕГРОВАНІ ТЕХНОЛОГІЇ АВТОМАТИЗАЦІЇ

НАВЧАЛЬНИЙ ПОСІБНИК

Підписано до друку 04.11.25 Формат 60x84\16
Ум. друк. арк. 14,0 Наклад 100 прим. Зам. № 250721

Виготовлювач Національний університет біоресурсів
і природокористування України,
вул. Героїв Оборони, 15, м. Київ, 03041.
Свідоцтво суб'єкта видавничої справи
ДК № 4097 від 17.06.2011