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

С. М. Лісовець, В. С. Антонюк, С. В. Барилко

SCADA-СИСТЕМИ

НАВЧАЛЬНИЙ ПОСІБНИК

Рекомендовано Методичною радою КПІ імені Ігоря Сікорського
як навчальний посібник для здобувачів ступеня бакалавра
за спеціальністю G7 Автоматизація, комп'ютерно-інтегровані технології та робототехніка
(174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка)

Електронне мережне навчальне видання

Київ
КПІ ім. Ігоря Сікорського
2026

УДК 004.5
Л63

Автори: *Лісовець Сергій Миколайович*, кандидат технічних наук, доцент
Антонюк Віктор Степанович, доктор технічних наук, професор
Барилко Сергій Віталійович, доктор технічних наук, доцент

Рецензенти: *Бондаренко М. О.*, завідувач кафедри приладобудування, мехатроніки та комп'ютеризованих технологій Черкаського державного технологічного університету, доктор технічних наук, професор
Киричук Ю. В., завідувач кафедри автоматизації та систем неруйнівного контролю КПІ ім. Ігоря Сікорського, доктор технічних наук, доцент

Відповідальний редактор: *Стельмах Н. В.*, кандидат технічних наук, доцент

Гриф надано Методичною радою КПІ імені Ігоря Сікорського
(протокол № 7 від 08.05.2026 р.)
за поданням Вченої ради факультету робототехніки та приладобудування
(протокол № 4/26 від 27.04.2026 р.)

Лісовець С. М., Антонюк В. С., Барилко С. В.

Л63 SCADA-системи. Навчальний посібник [Електронний ресурс] : навчальний посібник для здобувачів ступеня бакалавра за спеціальністю G7 Автоматизація, комп'ютерно-інтегровані технології та робототехніка (174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка) / КПІ ім. Ігоря Сікорського ; авт.: С. М. Лісовець, В. С. Антонюк, С. В. Барилко. – Електронні текстові дані (1 файл: 5,51 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2026. – 179 с. – Назва з екрана.

В навчальному посібнику розглядаються питання, присвячені: необхідності і доцільності використання SCADA-систем, основним середовищам обміну даними, інтерфейсам і протоколам, які використовуються в промисловій автоматизації, програмним засобам розширення можливостей SCADA-систем, обладнанню для SCADA-систем, SCADA-системам на основі панелі оператора ИП320 і SIMATIC WinCC, прикладам реалізації SCADA-систем в промисловій автоматизації. Навчальний посібник призначений для здобувачів ступеня бакалавра за спеціальністю G7 Автоматизація, комп'ютерно-інтегровані технології та робототехніка (174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка), а також всім, хто цікавиться SCADA-системами.

УДК 004.5

Реєстр. № НП 25/26-387. Обсяг 5,98 авт. арк.

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

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

© С. М. Лісовець, В. С. Антонюк, С. В. Барилко, 2026
© КПІ імені Ігоря Сікорського, 2026

ЗМІСТ

ЗМІСТ 3
ПЕРЕДМОВА 8
ВСТУП 9
1. SCADA-СИСТЕМИ: НЕОБХІДНІСТЬ І ДОЦІЛЬНІСТЬ ВИКОРИСТАННЯ 10
1.1. Основні відомості про SCADA-системи 10
1.2. Людино-машинний інтерфейс 12
1.3. Місце SCADA-систем в промисловій автоматизації 15
1.3.1. Структура промислової автоматизації 15
1.3.2. Основні підходи до оперативного збору даних 17
1.3.3. Основні підходи до оперативного керування 19
1.4. OPC-стандарти 20
1.5. Модель OSI 22
1.6. Інтеграція SCADA-систем з операційними системами реального часу 23
1.6.1. Загальні відомості про операційні системи реального часу 23
1.6.2. Операційна система реального часу QNX Neutrino RTOS: огляд основних можливостей 24
1.7. Питання для самоперевірки 25
1.8. Рекомендована література 26
2. ОСНОВНІ СЕРЕДОВИЩА ОБМІНУ ДАНИМИ, ЯКІ ВИКОРИСТОВУЮТЬСЯ В ПРОМИСЛОВІЙ АВТОМАТИЗАЦІЇ 27
2.1. Загальні відомості про середовища обміну даними, які використовуються в промисловій автоматизації 27
2.2. Кабель на основі мідних/алюмінієвих провідників 28
2.2.1. Кабель на основі паралельних провідників 28
2.2.2. Кабель на основі кручених провідників 28
2.2.3. Кабель на основі коаксіальних провідників 29
2.2.4. Еквівалентна електрична схема кабелів 30
2.2.5. Передавачі і приймачі електричних сигналів 31
2.2.6. Боротьба із завадами 36
2.2.7. Загальні рекомендації щодо обміну аналоговими і дискретними сигналами 42
2.3. Кабель на основі оптичного волокна 43
2.4. Електромагнітні хвилі 44
2.5. Структуровані кабельні системи 45
2.6. Питання для самоперевірки 46
2.7. Рекомендована література 47

3. ОСНОВНІ ІНТЕРФЕЙСИ, ЯКІ ВИКОРИСТОВУЮТЬСЯ В ПРОМИСЛОВІЙ АВТОМАТИЗАЦІЇ 48
3.1. Загальні відомості про інтерфейси, які використовуються в промисловій автоматизації 48
3.2. Інтерфейс у вигляді уніфікованих електричних неперервних сигналів напруги і струму 51
3.3. Інтерфейси RS-232 і RS-485 53
3.3.1. Інтерфейс RS-232 53
3.3.2. Інтерфейс RS-485 55
3.3.3. Кадр інтерфейсів RS-232 і RS-485 56
3.3.4. Деякі особливості обміну даними по інтерфейсам RS-232 і RS-485 57
3.4. Інтерфейс Ethernet 58
3.4.1. Специфікації інтерфейсу Ethernet 58
3.4.2. Деякі особливості обміну даними по інтерфейсу Ethernet 60
3.5. Питання для самоперевірки 61
3.6. Рекомендована література 62
4. ОСНОВНІ ПРОТОКОЛИ, ЯКІ ВИКОРИСТОВУЮТЬСЯ В ПРОМИСЛОВІЙ АВТОМАТИЗАЦІЇ 63
4.1. Загальні відомості про протоколи, які використовуються в промисловій автоматизації 63
4.2. Сімейство протоколів Modbus 63
4.2.1. Протокол Modbus RTU 63
4.2.2. Протокол Modbus ASCII 65
4.2.3. Протокол Modbus TCP/IP 66
4.2.4. Основні функції обробки даних сімейства протоколів Modbus 67
4.2.5. Обробка помилок, які виникли при обробці даних в сімействі протоколів Modbus 75
4.3. Протокол DCON 76
4.4. Протокол ОВЕН 77
4.5. Протокол HART 80
4.6. Питання для самоперевірки 82
4.7. Рекомендована література 82
5. ПРОГРАМНІ ЗАСОБИ РОЗШИРЕННЯ МОЖЛИВОСТЕЙ SCADA-СИСТЕМ 83
5.1. Використання типових алгоритмів обробки даних 83
5.1.1. Необхідність і доцільність додаткової обробки даних 83
5.1.2. Знаходження середнього арифметичного при прийманні даних від об’єкта автоматизації 83
5.1.3. Знаходження похідної при прийманні даних від об’єкта

автоматизації ... 83
5.1.4. Знаходження інтеграла при прийманні даних від об'єкта автоматизації ... 84
5.1.5. Коригування даних при їх прийманні від об'єкта автоматизації ... 84
5.1.6. Перетворення даних при їх прийманні від об'єкта автоматизації ... 85
5.1.7. Обмеження даних їх мінімальним і максимальним значеннями при їх передачі до об'єкта автоматизації ... 85
5.1.8. Обмеження швидкості зміни даних їх мінімальним і максимальним значеннями при їх передачі до об'єкта автоматизації ... 86
5.2. Використання мов програмування промислових контролерів ... 86
5.2.1. Необхідність і доцільність використання мов програмування промислових контролерів ... 86
5.2.2. Мова програмування IL (Instruction List) ... 87
5.2.3. Мова програмування ST (Structured Text) ... 87
5.2.4. Мова програмування LD (Ladder Diagram) ... 88
5.2.5. Мова програмування FBD (Function Block Diagram) ... 89
5.2.6. Мова програмування SFC (Sequential Function Chart) ... 89
5.2.7. Мова програмування CFC (Continuous Function Chart) ... 90
5.3. Використання мов структурованих запитів ... 91
5.3.1. Необхідність і доцільність використання мов структурованих запитів ... 91
5.3.2. Мова структурованих запитів SQL ... 92
5.4. Використання технологій World Wide Web ... 93
5.4.1. Необхідність і доцільність використання технологій World Wide Web ... 93
5.4.2. Основні можливості убудованих Web-серверів ... 94
5.5. Використання хмарних технологій ... 95
5.5.1. Необхідність і доцільність використання хмарних технологій ... 95
5.5.2. Основні можливості хмарних сервісів ... 95
5.6. Кібербезпека SCADA-систем ... 96
5.6.1. Необхідність і доцільність використання кібербезпеки SCADA-систем ... 96
5.6.2. Основні можливості кібербезпеки SCADA-систем ... 96
5.7. Питання для самоперевірки ... 97
5.8. Рекомендована література ... 98
6. ОБЛАДНАННЯ ДЛЯ SCADA-СИСТЕМ ... 99
6.1. Основні принципи вибору обладнання ... 99
6.2. Вибір апаратного забезпечення: загальні рекомендації ... 100
6.2.1. Модулі аналогового уведення ... 100
6.2.2. Модулі аналогового виведення ... 101
6.2.3. Модулі дискретного уведення ... 102

6.2.4. Модулі дискретного виведення 103
6.2.5. Модулі для здійснення показання 104
6.2.6. Модулі для здійснення реєстрації 105
6.2.7. Модулі для здійснення регулювання 106
6.2.8. Модулі для здійснення сигналізації 107
6.2.9. Програмовані промислові реле 108
6.2.10. Моноблокові програмовані промислові контролери (без операційної системи) 109
6.2.11. Моноблокові програмовані промислові контролери (з операційною системою) 110
6.2.12. Сенсорні панельні програмовані промислові контролери 111
6.2.13. Убудовані промислові комп'ютери 113
6.2.14. Сенсорні панельні промислові комп'ютери 115
6.2.15. Високопродуктивні промислові комп'ютери 116
6.2.16. Промислові монітори 117
6.3. Вибір програмного забезпечення: загальні рекомендації 118
6.4. Питання для самоперевірки 119
6.5. Рекомендована література 120
7. SCADA-СИСТЕМА НА ОСНОВІ ПАНЕЛІ ОПЕРАТОРА ИП320 121
7.1. Основні технічні характеристики панелі оператора ИП320 121
7.2. Програмне забезпечення панелі оператора ИП320 123
7.2.1. Загальні відомості про програмне забезпечення панелі оператора ИП320 123
7.2.2. Шаблон елементів "екранів" Текст 124
7.2.3. Шаблон елементів "екранів" Графічний текст 124
7.2.4. Шаблон елементів "екранів" Динамічний текст 125
7.2.5. Шаблон елементів "екранів" Динамічний графічний текст 126
7.2.6. Шаблон елементів "екранів" Індикатор 127
7.2.7. Шаблон елементів "екранів" Малюнок 127
7.2.8. Шаблон елементів "екранів" Регістр 128
7.2.9. Шаблон елементів "екранів" Лінійка 129
7.2.10. Шаблон елементів "екранів" Функціональна кнопка 130
7.2.11. Шаблон елементів "екранів" Графік 131
7.2.12. Шаблон елементів "екранів" Регістри ASCII-тексту 132
7.3. Приклад реалізації SCADA-системи 132
7.4. Питання для самоперевірки 135
7.5. Рекомендована література 135
8. SCADA-СИСТЕМА SIMATIC WINCC: ОГЛЯД ОСНОВНИХ МОЖЛИВОСТЕЙ 136
8.1. Знайомство з SIMATIC WinCC 136
8.2. Основні складові SIMATIC WinCC 136

8.2.1. Комп'ютер (Computer) 138
8.2.2. Керування ознаками (Tag Management) 138
8.2.3. Дизайнер графіки (Graphics Designer) 142
8.2.4. Меню і панелі інструментів (Menus and Toolbars) 154
8.2.5. Текстові і графічні списки (Text and Graphics Lists) 154
8.2.6. Реєстрація тривог (Alarm Logging) 155
8.2.7. Реєстрація ознак (Tag Logging) 155
8.2.8. Дизайнер звітів (Report Designer) 155
8.2.9. Глобальний сценарій (Global Script) 156
8.2.10. Бібліотека текстів (Text Library) 156
8.2.11. Розподілювач тексту (Text Distributor) 156
8.2.12. Адміністратор користувача (User Administrator) 157
8.2.13. Перехресне посилання (Cross Reference) 157
8.2.14. Завантаження Online-змін (Load Online Changes) 157
8.2.15. Надлишковість (Redundancy) 157
8.2.16. Архів користувача (User Archive) 158
8.2.17. Синхронізація часу (Time Synchronization) 158
8.2.18. Звукова сигналізація (Horn) 158
8.2.19. Дерево зображень (Picture Tree) 158
8.2.20. Контроль працеспроможності (Lifebeat Monitoring) 159
8.2.21. Редактор проєктів OS (OS Project Editor) 159
8.2.22. Web-навігатор (Web Navigator) 159
8.3. Використання SIMATIC WinCC в промисловій автоматизації 160
8.4. Питання для самоперевірки 161
8.5. Рекомендована література 162
9. ПРИКЛАДИ РЕАЛІЗАЦІЇ SCADA-СИСТЕМ В ПРОМИСЛОВІЙ АВТОМАТИЗАЦІЇ 163
9.1. SCADA-система для системи керування вентиляцією 163
9.2. SCADA-система для системи керування сушильною камерою 166
9.3. SCADA-система для системи керування дозуванням 168
9.4. SCADA-система для системи керування сховищем бітуму 171
9.5. SCADA-система для системи керування дуговою піччю 174
9.6. Питання для самоперевірки 176
9.7. Рекомендована література 177
СПИСОК ЛІТЕРАТУРИ 178

ПЕРЕДМОВА

Сучасні виробничі процеси (а також технологічні процеси як складова виробничих процесів) передбачають максимально можливу автоматизацію і одночасно максимально можливий контроль зі сторони людини. Це дозволяє виявляти порушення на будь-яких операціях, ділянках або лініях таких процесів, за рахунок чого з'являється можливість підвищення безпеки праці і якості продукції, яка випускається.

Для того, щоб забезпечити максимальну керуємість виробничим процесом, необхідно оперативно збирати дані про виробничі параметри і одночасно з цим (за необхідності) створювати керувальні впливи на ці параметри. Для розв'язання таких задач призначені SCADA-системи.

SCADA-системи (абр. від англ. Supervisory Control and Data Acquisition, Оперативне керування і збір даних) – це пакети програм, які призначені для розробки і/або забезпечення роботи систем збору, обробки, відображення і архівування даних про об'єкт автоматизації. Звичайно вони є частиною систем промислової автоматизації і можуть використовуватися практично в усіх галузях промисловості. Основні особливості SCADA-систем – взаємодія з об'єктом автоматизації в реальному часі і забезпечення людино-машинного інтерфейсу. Вони можуть бути як централізованими, так і розподіленими.

Основним людино-машинним інтерфейсом SCADA-систем є мнемосхеми – наглядні графічні зображення структурної або функціональної схеми об'єкта автоматизації, які розробляються і виконуються на засобах обчислювальної техніки (персональних/промислових комп'ютерах, серверах, робочих станціях і так далі). Крім того, до складу мнемосхем SCADA-систем часто входять убудовані екрани контролерів, панелі оператора, сенсорні панельні контролери, промислові комп'ютери при наявності монітора і так далі.

Разом зі SCADA-системою звичайно використовуються один або кілька OPC-серверів (OPC DA, OPC HDA, OPC AE, OPC UA і так далі). Вони є своєрідним "прошарком" між технічними засобами автоматизації (з одного боку) і SCADA-системою (з іншого боку), здійснюючи доступ до таких засобів за допомогою відповідним інтерфейсів, протоколів і каналів обміну даними. Іноді OPC-сервери є невід'ємною складовою частиною SCADA-систем.

Таким чином, знання основних принципів побудови SCADA-систем дає можливість суттєво підвищити якість виробничої діяльності людини.

ВСТУП

Навчальний посібник з дисципліни “SCADA-системи” передбачає розгляд основних питань, які відносяться до розробки і/або забезпечення роботи систем збору, обробки, відображення і архівування даних про об’єкт автоматизації (тобто SCADA-систем).

Він складається з дев’яти лекційних занять. Кожне лекційне заняття утримує лекційний матеріал, питання для самоперевірки і рекомендовану літературу.

Зокрема, 1-ше лекційне заняття присвячено SCADA-системам: необхідності і доцільності використання; 2-ге лекційне заняття присвячено основним середовищам обміну даними, які використовуються в промисловій автоматизації; 3-тє лекційне заняття присвячено основним інтерфейсам, які використовуються в промисловій автоматизації; 4-те лекційне заняття присвячено основним протоколам, які використовуються в промисловій автоматизації; 5-те лекційне заняття присвячено програмним засобам розширення можливостей SCADA-систем; 6-те лекційне заняття присвячено обладнанню для SCADA-систем; 7-ме лекційне заняття присвячено SCADA-системі на основі панелі оператора ИП320; 8-ме лекційне заняття присвячено SCADA-системі SIMATIC WinCC: огляду основних можливостей; 9-те лекційне заняття присвячено прикладам реалізації SCADA-систем в промисловій автоматизації.

1. SCADA-СИСТЕМИ: НЕОБХІДНІСТЬ І ДОЦІЛЬНІСТЬ ВИКОРИСТАННЯ

1.1. Основні відомості про SCADA-системи

SCADA-системи (абр. від англ. Supervisory control and data acquisition, Оперативне керування і збір даних) – це пакети програм, які призначені для приймання даних від об’єкта автоматизації (здійснюється вимірювання) і/або передачі даних до об’єкта автоматизації (здійснюється керування). При цьому типовими функціональними ознаками SCADA-систем є показання, реєстрація, регулювання і/або сигналізація. SCADA-системи є частиною промислової автоматизації, вони можуть використовуватися практично в усіх галузях промисловості. SCADA-системи створюються як іноземними, так і вітчизняними розробниками. SCADA-системи можуть бути як самостійними системами, так і входити до складу інших систем (автоматизованих систем керування виробничими (технологічними) процесами, автоматизованих систем керування підприємствами, систем керування “розумними будівлями”, систем моніторингу екологічного стану довкілля і так далі). Основна особливість переважної більшості SCADA-систем – це взаємодія з об’єктом автоматизації в реальному часі. SCADA-системи звичайно можуть працювати в двох режимах: в режимі розробки (в такому режимі здійснюється, наприклад, розробка мнемосхем) і в режимі виконання (в цьому такому здійснюється обмін даними з об’єктом автоматизації). SCADA-системи переважно є не централізованими, а розподіленими системами. Одне з їх призначань – забезпечення людино-машинного інтерфейсу. Використання SCADA-систем дозволяє підвищити якість і продуктивність виробництва, оптимізувати витрати на виробництво, підвищити безпеку і керованість виробництва. Одним з “недоліків” SCADA-систем є їхня надлишковість – для взаємодії з об’єктом автоматизації часто не потрібні всі можливості, які такі системи мають. Відповідно, рекомендується використовувати наступні підходи при виборі SCADA-систем: при великій складності об’єкта автоматизації – обирати програмне забезпечення відомих розробників, при невеликій складності об’єкта автоматизації – створювати SCADA-системи самостійно шляхом використання відповідних середовищ розробки програмного забезпечення.

SCADA-системи, як складові частини промислової автоматизації, можуть виконувати наступні типові функції.

1. Оперативний збір даних про виробничі (технологічні), налагоджувальні і інші параметри об’єкта автоматизації. Такий збір даних може бути здійсненим від вимикачів, кнопок і тумблерів, кінцевих вимикачів, вимірювальних перетворювачів, датчиків і сенсорів, аналого-цифрових і цифро-аналогових перетворювачів, програмованих контролерів і реле, модулів уведення/виведення, регуляторів і так далі.

2. Оперативне керування виробничими (технологічними), налагоджувальними і іншими параметрами об’єкта автоматизації. Таке керування може бути здійсненне до підсилювачів потужності, ШІМ- і частотних перетворювачів, електромеханічних, електропневматичних і електрогідравлічних приводів, електромеханічних і твердотільних реле, пристроїв плавного пуску і захисту електричних двигунів і так далі.

3. Збереження зібраних даних, а також даних інших типів в архівах.

В якості архівів можуть виступати автономні реєстратори, модулі збору даних, локальні і розподілені бази даних, SSD- і HDD-накопичувачі, хмарні сервіси і так далі.

4. Додаткова обробка зібраних даних, а також даних інших типів.

До такої обробки відноситься: придушення “деренчання” контактів, отримання мінімальних і максимальних значень сигналу, диференціювання і інтегрування сигналу, затримка сигналу на заданий інтервал часу, спрацьовування по передньому і задньому фронтах сигналу, обмеження швидкості зміни сигналу, визначення “якості” прийнятого сигналу (наприклад, наявності короткого замикання або обриву) і так далі.

5. Формування звітних документів.

Передбачає створення електронних версій документів з подальшим друком їх на принтері, відправкою у вигляді SMS-повідомлень і повідомлень електронної пошти, збереженням в базах даних і так далі.

6. Обмін даними з іншими системами промислової автоматизації.

При цьому з боку SCADA-систем необхідна підтримка тих форматів обміну даними (зокрема, інтерфейсів і протоколів), які такі системи промислової автоматизації мають.

7. Забезпечення людино-машинного інтерфейсу.

Реалізує “інформаційну модель” – певне відображення об’єкта автоматизації, через яке людина сприймає його стан (так як безпосередньо сприйняти стан об’єкта автоматизації, тобто сукупність значень його параметрів (температур, тисків, витрат, рівнів і так далі), людина не може).

1.2. Людино-машинний інтерфейс

Спрощена схема людино-машинного інтерфейсу у вигляді “людина–мнемосхема–машина” наведена на рис. 1.1.

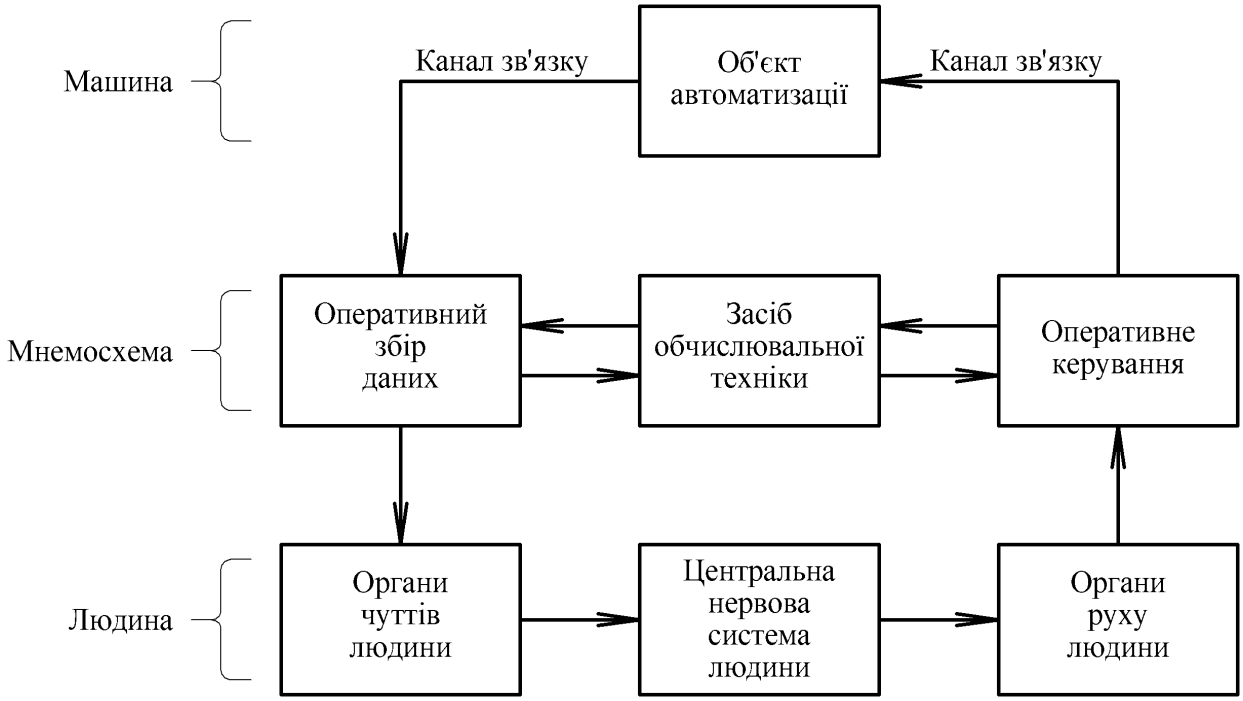


Рис. 1.1. Спрощена схема людино-машинного інтерфейсу

Забезпечення людино-машинного інтерфейсу звичайно здійснюється шляхом використання однієї або кількох мнемосхем (наглядних графічних зображень людино-машинного інтерфейсу). Мнемосхеми можуть бути представлені або повністю на екрані монітора комп’ютера (екранах моніторів комп’ютерів), або на екрані монітора (екранах моніторів) комп’ютерів і одночасно з цим у вигляді зображень на панелях шаф, щитів і пультів керування, екранах програмованих контролерів і реле, панелях операторів і так далі. Типовими елементами мнемосхем, які дозволяють забезпечувати людино-машинний інтерфейс, є такі елементи, як графічні примітиви (наприклад, прямокутники, кола, лінії і дуги), елементи діалогу (наприклад, рядки для уведення тексту, кнопки і списки) і елементи мультимедіа (наприклад, діаграми і рисунки).

Застосування мнемосхем є ефективним, коли структура об’єкта автоматизації є достатньо складною і/або коли структура об’єкта автоматизації в процесі його роботи може змінюватися. Співвідношення кількості “пасивних” і “активних” елементів мнемосхеми має назву коефіцієнта інформативності

мнемосхеми, а відношення кількості “пасивних” елементів мнемосхеми до загальної кількості елементів мнемосхеми – коефіцієнта заповнюваності мнемосхеми. Крім мнемосхем, для забезпечення людино-машинного інтерфейсу також можуть використовуватися такі засоби як, наприклад, клавішні, кнопкові і поворотні вимикачі і перемикачі, тумблери, важелі керування, маховики керування, штурвали і рульові колеса.

Вимоги до людино-машинного інтерфейсу наведені в різних документах – зокрема, одним з таких документів є стандарт ANSI/ISA-101.01-2015 Human Machine Interfaces for Process Automation Systems.

При створені мнемосхем звичайно притримуються наступних п’яти принципів.

Принцип функціональності – передбачає групування елементів мнемосхем по їх функціям і/або по об’єктам автоматизації, до яких вони відносяться.

Принцип значимості – передбачає групування елементів мнемосхем по їх важливості при виконання тієї або іншої дії.

Принцип розташування – передбачає групування елементів мнемосхем по їх конструктивним особливостям (наприклад, по розміру шкал, по точності вимірювання і по швидкодії).

Принцип послідовності – передбачає групування елементів мнемосхем по послідовності рухів при виконанні тієї або іншої дії.

Принцип частоти використання – передбачає групування елементів мнемосхем по частоті їх використання.

При створені мнемосхем дуже важливим є розуміння принципів (логіки) роботи об’єкта автоматизації, в якому мнемосхеми її повинні допомагати. Також при створені мнемосхем звичайно розробляється кілька їх варіантів, з яких в подальшому обирається (наприклад, шляхом побудови моделі певної діяльності людини і подальшої її симуляції) найбільш оптимальний. Розробка мнемосхем може бути віднесена не тільки до суто технічної задачі, але й до задачі художнього конструювання, при розв’язанні якої необхідно враховувати специфічні вимоги технічної естетики і інженерної психології.

При розробці мнемосхем необхідно притримуватися наступних простих правил (які витікають з принципів функціональності, значимості і так далі):

- на мнемосхемі не повинні відображатися другорядні елементи виробничого (технологічного) процесу, які можуть відволікати увагу людини і утруднювати пошук потрібних їй елементів мнемосхеми;

- мнемосхема повинна давати людині можливість швидко оцінити стан виробничого (технологічного) процесу – на ній повинні бути відображені всі основні контури показання, сигналізації, реєстрації і керування такого процесу, але ступінь їх деталізації не повинна бути надлишковою;

- зображення елементів виробничого (технологічного) процесу на мнемосхемі повинні бути не тільки красивими, але й достатньо компактними і доступними для огляду;

- щільність розміщення елементів мнемосхеми не повинна бути надмірною (необхідно мати можливість в подальшому вносити в мнемосхему необхідні зміни якщо, наприклад, буде змінено виробничий (технологічний) процес);

- напрям ліній виробничих (технологічних) потоків на мнемосхемах звичайно приймається зліва направо;

- символи виробничого (технологічного) обладнання, які відносяться до однієї і тієї ж виробничої (технологічної) операції, ділянки, лінії або процесу, повинні відображатися на мнемосхемі рівномірно і недалеко один від одного як одна відокремлена група;

- довжина і кількість перетинань існуючих на мнемосхемі ліній виробничих (технологічних) потоків повинна бути мінімальною;

- лінії виробничих (технологічних) потоків на мнемосхемі повинні мати стрілки або написи, як вказують напрям таких потоків.

Приклад нескладних мнемосхем, призначених для контролю температури в 2-ох точках і виконаних в SCADA-системах Smart ReView version 1.0 і Visual Intellect version 2.0 розробки ТОВ “МІКРОЛ” (Україна, м. Івано-Франківськ), наведено відповідно на рис. 1.2 і рис. 1.3.

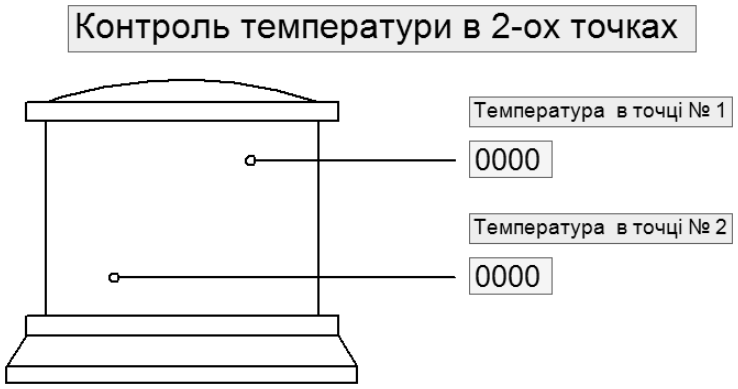


Рис. 1.2. Мнемосхема SCADA-системи SCADA-система Smart ReView version 1.0

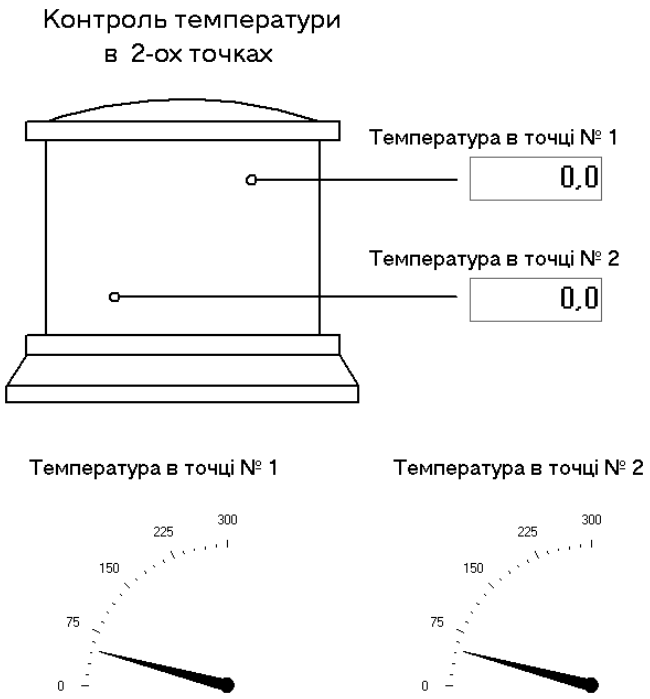


Рис. 1.3. Мнемосхема SCADA-системи Visual Intellect version 2.0

1.3. Місце SCADA-систем в промисловій автоматизації

1.3.1. Структура промислової автоматизації

Промислову автоматизації часто представляють у вигляді чотирирівневої структури, кожна з складових частин якої (також і SCADA-системи) мають своє призначення (див. рис. 1.4).

На 1-му рівні знаходяться засоби автоматизації, які безпосередньо контактують з об’єктом автоматизації. До засобів, які можуть забезпечити оперативний збір даних, відносяться, зокрема, вимірювальні перетворювачі, датчики, сенсори, аналого-цифрові перетворювачі, модулі уведення і так далі – з їх допомогою можна визначати стан об’єкта автоматизації. А до засобів, які можуть забезпечити оперативне керування, відносяться, зокрема, модулі виведення, регулятори і так далі – з їх допомогою можна впливати на стан об’єкта автоматизації.

На 2-му рівні знаходиться безпосередньо SCADA-система (часто виробництво може обійтись взагалі без SCADA-систем, але наявність SCADA-системи і людино-машинного інтерфейсу, який вона утворює, суттєво покращує інформованість людини про стан такого виробництва, внаслідок чого можна підвищити продуктивність і безпеку праці, зменшити відсоток браку і навантаження на навколишнє середовище).

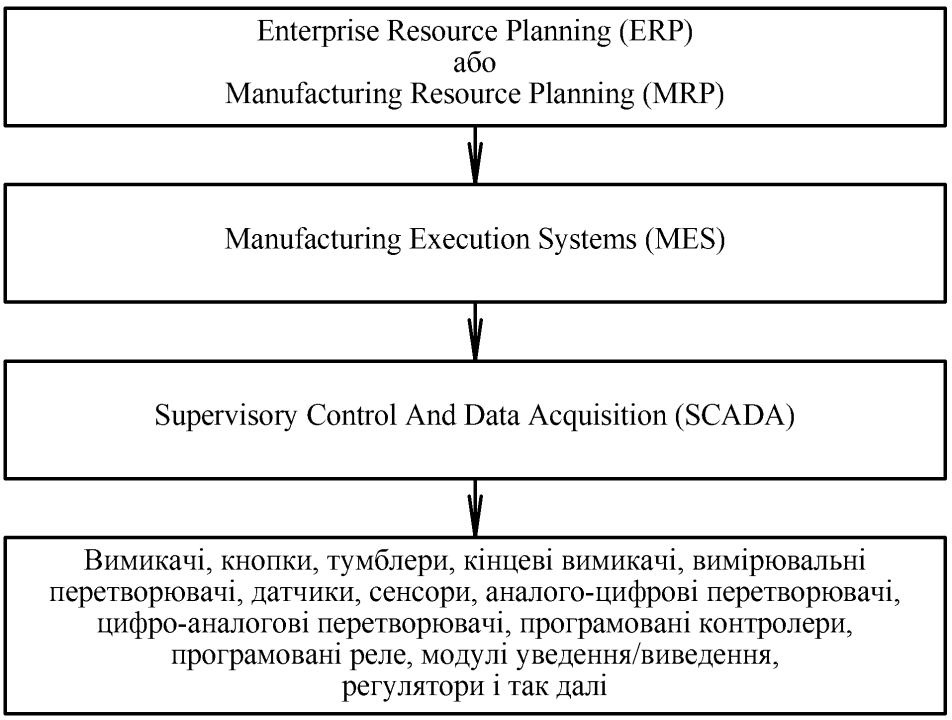


Рис. 1.4. Місце SCADA-систем в промисловій автоматизації

На 3-му рівні знаходиться Manufacturing Execution Systems (MES) – система організації виробництва. Така система звичайно знаходиться на рівні цеху, виробничої (технологічної) лінії або процесу. Іноді без неї на виробництві можна обійтися.

На 4-му рівні знаходиться Enterprise Resource Planning (ERP) – система планування ресурсів підприємства або Manufacturing Resource Planning (MRP) – система планування ресурсів виробництва. Така система звичайно знаходиться на рівні підприємства. Іноді без неї на виробництві також можна обійтися.

Типовий підхід до реалізації 1-го і 2-го рівнів систем промислової автоматизації наведено на рис. 1.5. Особливістю такої реалізації є те, що вона дозволяє здійснювати оперативне керування і збір даних в межах однієї виробничої (технологічної) операції або ділянки, використовуючи в середньому від кількох одиниць до кількох десятків засобів автоматизації. При переході до оперативного керування і збору даних в межах однієї виробничої (технологічної) лінії або процесу, які звичайно складаються з кількох виробничих (технологічних) операцій або ділянок, оперативне керування і збір даних на інших операціях або ділянках буде здійснюватися аналогічним чином.

Звичайно SCADA-система, як пакет програм, розміщується на одному промисловому комп’ютері (іноді на кількох – наприклад, з метою підвищення надійності своєї роботи). Крім цього, на виробництві можуть, незалежно одна

від одної, використовуватися кілька SCADA-систем. Все наявне обладнанням, яке “охоплюється” SCADA-системою, з’єднується за допомогою тих або інших інтерфейсів.

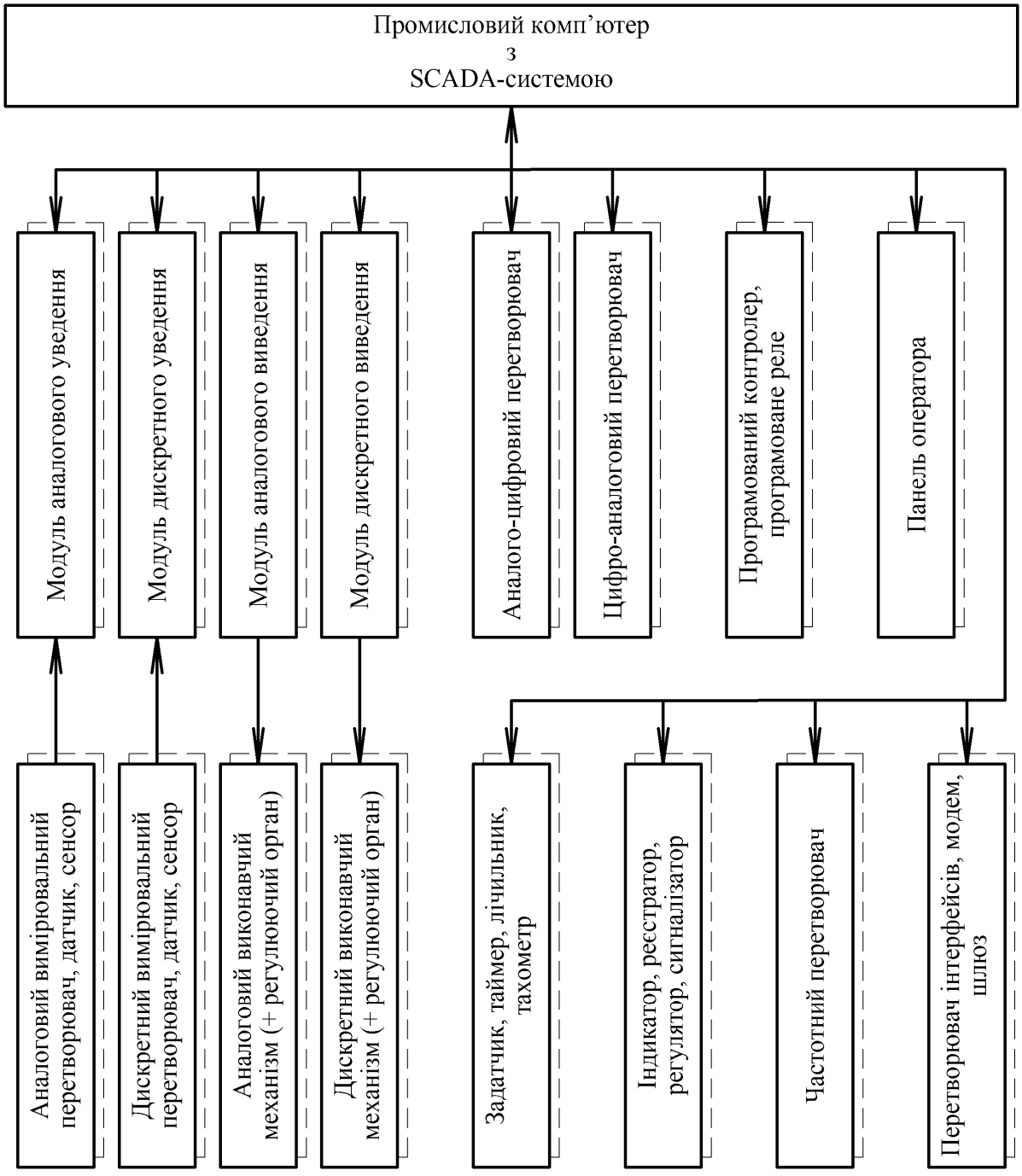


Рис. 1.5. Реалізація 1-го і 2-го рівнів систем промислової автоматизації

1.3.2. Основні підходи до оперативного збору даних

Приклади реалізації контролю температури, при якому здійснюється вимірювання температури, наведено на рис. 1.6.

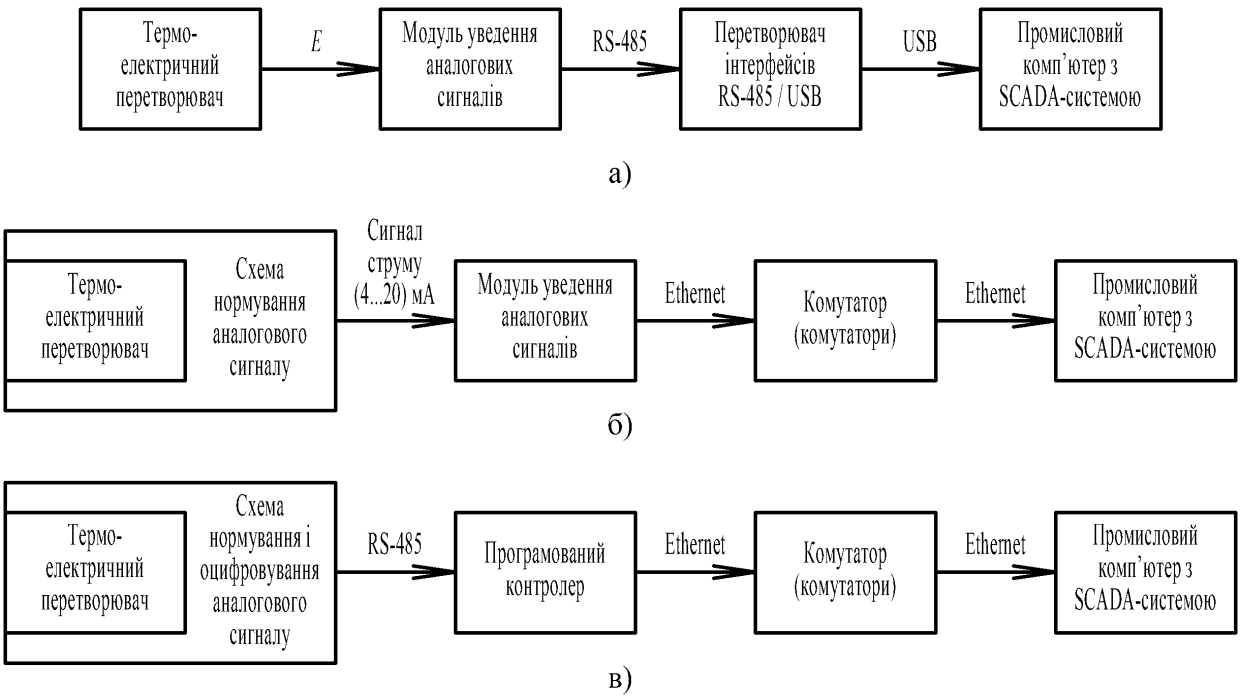


Рис. 1.6. Реалізація контролю температури (здійснюється вимірювання температури): а) перший випадок; б) другий випадок; в) третій випадок

В першому випадку (див. рис. 1.6, а) нелінійний аналоговий сигнал даних від термоелектричного перетворювача у вигляді термоелектрорушійної сили E надходить до модуля уведення аналогових сигналів, де нормується і перетворюється в цифрову форму. Після цього такий сигнал через інтерфейс RS-485 надходить до перетворювача інтерфейсів RS-485/USB, а далі через інтерфейс USB надходить до промислового комп'ютера з SCADA-системою.

В другому випадку (див. рис. 1.6, б) термоелектричний перетворювач обладнаний схемою нормування аналогового сигналу даних, яка дозволяє отримати з нелінійного аналогового сигналу у вигляді термоелектрорушійної сили E відповідний йому лінійний аналоговий сигнал у вигляді постійного струму. Після цього такий сигнал у вигляді сигналу струму $(4 \ldots 20)$ мА надходить до модуля уведення аналогових сигналів, де перетворюється в цифрову форму. Далі він через інтерфейс Ethernet надходить до комутатора (комутаторів), а ще далі через інтерфейс Ethernet надходить до промислового комп'ютера з SCADA-системою.

В третьому випадку (див. рис. 1.6, в) термоелектричний перетворювач обладнаний схемою нормування і перетворення в цифрову форму аналогового сигналу даних, яка дозволяє отримати з нелінійного аналогового сигналу у вигляді термоелектрорушійної сили E відповідний йому лінійний цифровий сигнал в заданому цифровому форматі. Після цього такий сигнал через

інтерфейс RS-485 надходить до програмованого контролера, де додатково обробляється (наприклад, послідовно отримується кілька десятків значень сигналу і розраховується середнє арифметичне цих значень), а ще далі через інтерфейс Ethernet надходить до промислового комп'ютера з SCADA-системою.

Таким чином, вимірювання температури можна здійснювати, використовуючи той же самий термоелектричний перетворювач, але різні схеми нормування і/або оцифровування, різні способи додаткової обробки, різні інтерфейси, різне додаткове обладнання і так далі.

1.3.3. Основні підходи до оперативного керування

Приклади реалізації контролю температури, при якому здійснюється керування температурою, наведено на рис. 1.7.

В першому випадку (див. рис. 1.7, а) цифровий сигнал керування від промислового комп'ютера з SCADA-системою через інтерфейс USB надходить до перетворювача інтерфейсів RS-485/USB, а далі через інтерфейс RS-485 надходить до модуля виведення аналогових сигналів, де перетворюється в аналоговий сигнал напруги $(0 \ldots 10)$ В. Після цього такий сигнал надходить до електричного нагрівача, який обладнаний схемою керування тиристорами і/або симісторами, що призначена для керування подачею на такий нагрівач електричної енергії.

В другому випадку (див. рис. 1.7, б) цифровий сигнал керування від промислового комп'ютера з SCADA-системою через інтерфейс Ethernet надходить до комутатора (комутаторів), а ще далі через інтерфейс Ethernet надходить до модуля виведення аналогових сигналів, де перетворюється в аналоговий сигнал струму $(4 \ldots 20)$ мА. Після цього такий сигнал надходить до електричного нагрівача, який обладнаний схемою керування тиристорами і/або симісторами, що призначена для керування подачею на такий нагрівач електричної енергії.

В третьому випадку (див. рис. 1.7, в) цифровий сигнал керування від промислового комп'ютера з SCADA-системою через інтерфейс Ethernet надходить до комутатора (комутаторів), а ще далі через інтерфейс Ethernet надходить до програмованого контролера, де додатково обробляється (наприклад, обмежується швидкість зміни сигналу). Після цього такий сигнал через інтерфейс RS-485 надходить до електричного нагрівача, який обладнаний схемою керування тиристорами і/або симісторами, що призначена для

керування подачею на такий нагрівач електричної енергії.

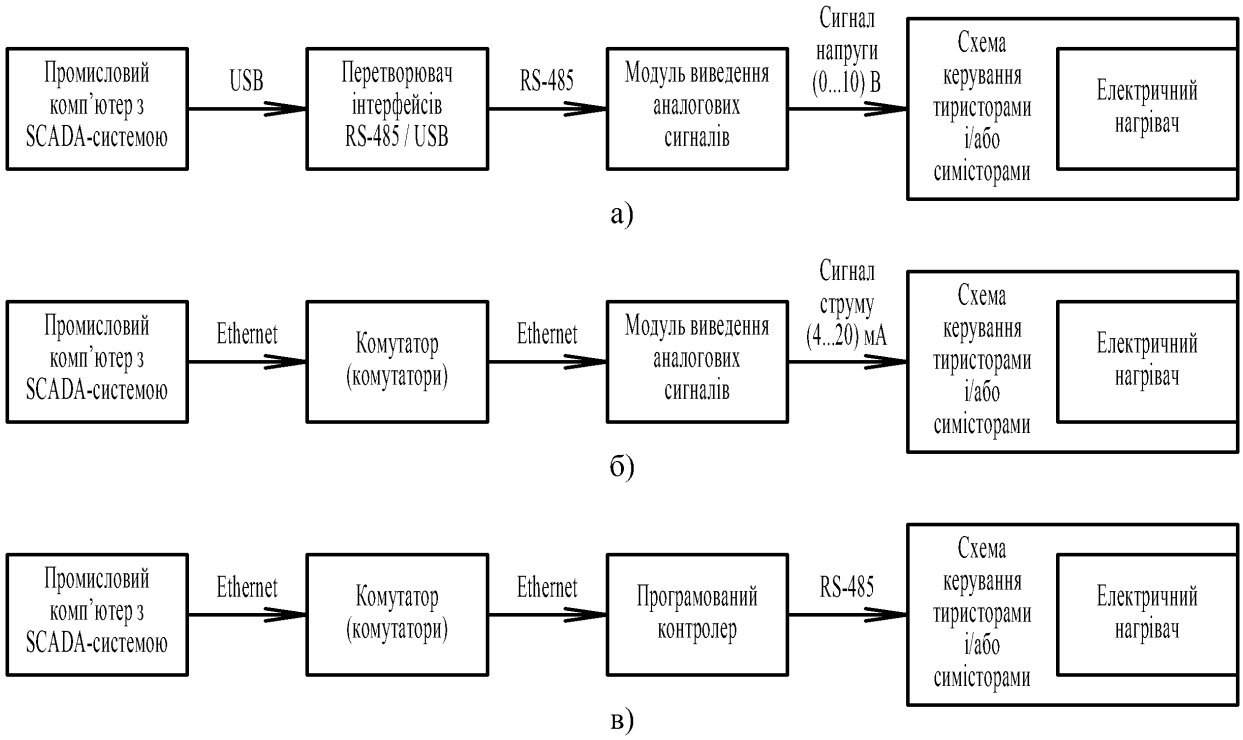


Рис. 1.7. Реалізація контролю температури (здійснюється керування температурою): а) перший випадок; б) другий випадок; в) третій випадок

1.4. OPC-стандарти

OPC-стандарти (абр. від англ. Open platform communications, Відкрита платформа для комунікацій) – це описи взаємодії між об'єктами автоматизації і SCADA-системами. Застаріла абревіатура OPC – Object linking and embedding for Process Control. Стандартизацією OPC-стандартів займається некомерційна організація OPC Foundation (opcfoundation.org), яка була створена в 1994 р. Девізом організації OPC Foundation є девіз "Відкриті комунікації по відкритих протоколах".

OPC-сервери/клієнти – це пакети програм, які призначені для реалізації відповідних OPC-стандартів (одного або кількох одночасно). Звичайно одна SCADA-система (як OPC-клієнт) може взаємодіяти з кількома різними OPC-серверами (за умови, що SCADA-система і OPC-сервери – це незалежне одне від іншого програмне забезпечення).

На теперішній час існують і використовуються наступні OPC-стандарти.

1. OPC DA (абр. від англ. Data access, Доступ до даних) – OPC-стандарт, який використовується найбільш часто для опису обміну даними об'єктами автоматизації в реальному часі. Стандарт OPC DA підтримує більше 90 %

сучасних OPC-серверів/клієнтів.

2. OPC HDA (абр. від англ. Historical data access, Доступ до накопичених даних) – OPC-стандарт, який також достатньо часто використовується для опису обміну архівованими даними з об’єктами автоматизації. Стандарт OPC HDA підтримує також більше 90 % сучасних OPC-серверів/клієнтів.

3. OPC AE (абр. від англ. Alarms and events, Сигналізація і події) – OPC-стандарт, який використовується для опису тривог (незапланованих змін в умовах протікання виробничого (технологічного) процесу) і подій (запланованих змін в умовах протікання виробничого (технологічного) процесу). Часто як тривоги, так і події вимагають з боку людини підтвердження.

4. OPC Batch – OPC-стандарт, який використовується для “рецептурного” керування виробничим (технологічним) процесом (згідно з стандартами ANSI/ISA-88). “Рецептурне” керування полягає у визначенні послідовності дій над первинним матеріалом і/або сировиною для отримання кінцевого продукту (в основному при масовому виробництві в перероблювальних галузях різних промисловостей).

5. OPC DX (абр. від англ. Data exchange, Обмін даними) – OPC-стандарт, який визначає обмін даними між OPC-серверами/клієнтами через різні мережі (зокрема, через Ethernet). Основне призначення стандарту OPC DX – створення шлюзів для обміну даними між обладнанням і програмним забезпеченням різних виробників.

6. OPC Security – OPC-стандарт, який визначає права доступу людини до даних через OPC-сервер/клієнт.

7. OPC XML DA (абр. від англ. Extensible markup language data access, Доступ до даних, які представлені за допомогою розширюваної мови розмітки) – OPC-стандарт, який визначає формат обміну даними через SOAP (Simple Object Access Protocol) і HTTP (HyperText Transfer Protocol).

8. OPC UA (абр. від англ. Unified architecture, Уніфікована архітектура) – найбільш сучасний і перспективний OPC-стандарт, який є заміною OPC DA, OPC HDA і OPC AE та має багато нових можливостей.

Основною більшості OPC-стандартів (за виключенням стандарту OPC UA) є такі технології Microsoft Corp., як DDE (Dynamic Data Exchange), OLE (Object Linking and Embedding), COM (Component Object Model) і/або DCOM (Distributed Component Object Model).

На відміну від цього, основною стандарту OPC UA є технологія SOA

(Service-Oriented Architecture), яка може бути реалізована на різних комп'ютерних "платформах".

1.5. Модель OSI

Модель OSI (абр. від англ. Open systems interconnection, Взаємодія відкритих систем) – модель, призначена для створення єдиного середовища взаємодії між різними мережевим пристроями (див. рис. 1.8). Вона була розроблена в кінці 70-х і прийнята на початку 80-х років XX століття ISO (абр. від англ. International organization for standardization, Міжнародна організація по стандартизації).

Модель OSI визначна в кількох стандартах ISO 7498 і є концептуальною моделлю (достатньо складною, опис якої займає більше 1000 сторінок тексту). Сучасні промислові мережі використовують переважно 2 або 3 рівня такої моделі (але, наприклад, промислова мережа Ethernet звичайно використовує 1-ий, 2-ий, 3-ій, 4-ий і 7-ий рівні – тобто загалом 5 рівнів).

Кожний з рівнів моделі OSI має своє призначення:

- фізичний рівень (Physical Layer) – забезпечує обмін даними через середовище обміну даних (наприклад, через коаксіальний кабель);
- канальний рівень (Data Link Layer) – забезпечує доступність середовища обміну даними шляхом поділу даних на кадри та виявлення і коригування помилок обміну даними;
- мережевий рівень (Network Layer) – забезпечує обмін даними між різними мережами;
- транспортний рівень (Transport Layer) – забезпечує обмін даними із заданим рівнем якості;
- сеансовий рівень (Session Layer) – забезпечує керування сеансом обміну даними;
- представницький рівень (Presentation Layer) – забезпечує представлення даних, не змінюючи їх умісту, в різних формах;
- прикладний рівень (Application Layer) – забезпечує надання доступу до мережі (на цьому рівні працює прикладне програмне забезпечення).

Середовище обміну даними може бути або проводовим (використовуються звичайно мідних і алюмінієві провідники та оптичні волокна), або безпроводовим (використовуються звичайно електромагнітні хвилі різних частотних діапазонів – зокрема, діапазон радіохвиль, інфрачервоний діапазон, видимий діапазон).

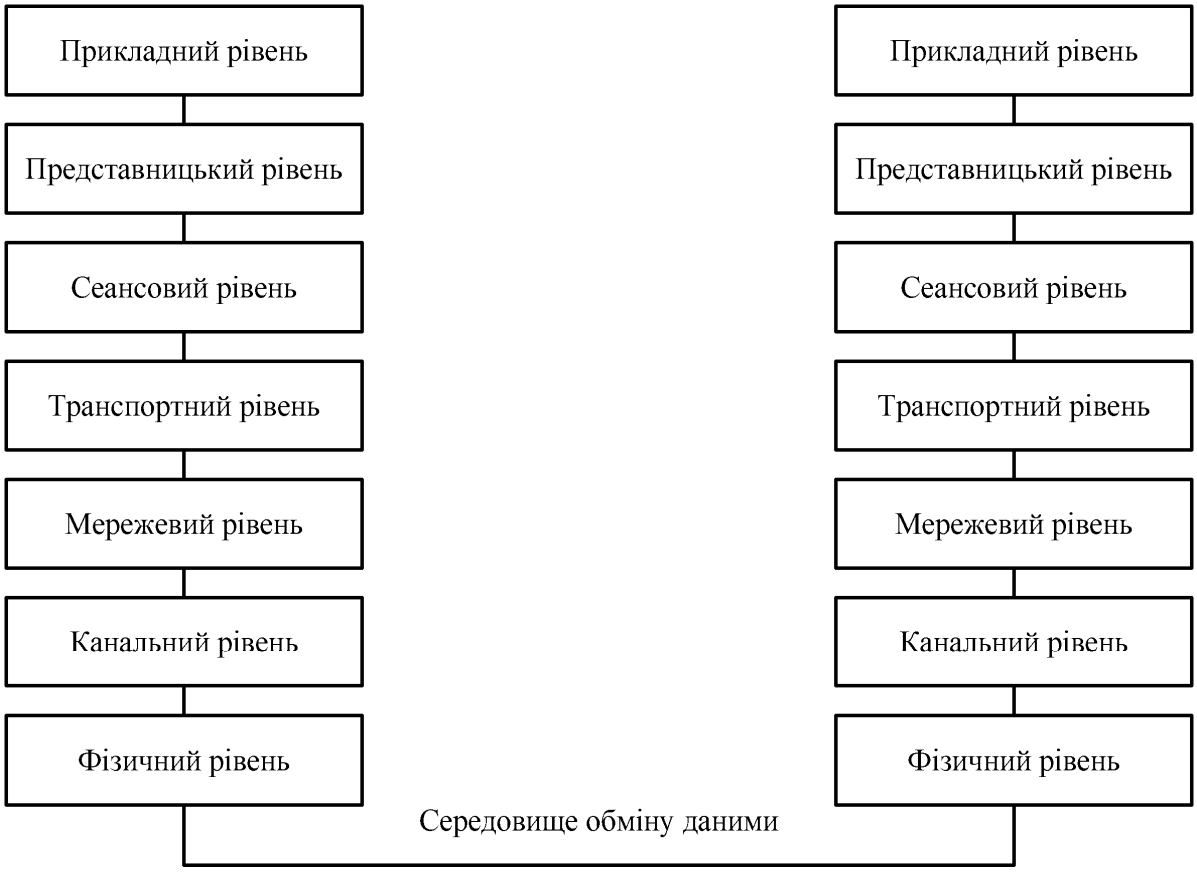


Рис. 1.8. Модель OSI

1.6. Інтеграція SCADA-систем з операційними системами реального часу

1.6.1. Загальні відомості про операційні системи реального часу

RTOS (абр. від англ. Real-time operating system, Операційна система реального часу) – операційна система, яка може надавати повністю або частково свої обчислювальні можливості, необхідні для роботи певного програмного забезпечення, в “реальному часі”. В промисловій автоматизації таким програмним забезпечення є, зокрема, SCADA-системи.

Реальний час – поняття, яке показує спроможність операційної системи забезпечити потрібний рівень сервісу в задані (часто заздалегідь) інтервали часу. В ідеалі операційна система реального часу повинна забезпечувати повністю передбачувану поведінку при будь-яких допустимих обчислювальних навантаженнях.

Необхідність використання SCADA-систем з операційними системами саме реального часу пояснюється тим, що найголовнішою умовою виконання оперативного збору даних від об’єкта автоматизації і/або оперативного

керування об’єктом автоматизації є їх часова детермінованість.

В операційних системах “жорсткого” реального часу будь-яке порушення часової детермінованості (поява додаткових часових затримок) є “фатальною” помилкою, яка може привести до катастрофічних наслідків і в загальному випадку є неприпустимим. До таких операційних систем можна віднести, наприклад, QNX, FreeRTOS і деякі версії Linux.

В операційних системах “м’якого” реального часу таке порушення є “відновлюваною” помилкою, яке може привести до зниження якості виробництва, збільшення фінансових витрат, зниження продуктивності праці, але не є катастрофічним. До таких операційних систем можна віднести наприклад, сімейство Windows IoT.

1.6.2. Операційна система реального часу QNX Neutrino RTOS: огляд основних можливостей

QNX Neutrino RTOS (абр. від англ. Real-Time Operating System, Операційна система реального часу) (далі – QNX Neutrino) – операційна система, яка є однією з найкращих реалізацій концепції мікроядерних операційних систем.

QNX Neutrino відповідає багатьом промисловим стандартам (також і стандартам POSIX).

Архітектура QNX Neutrino кардинально відрізняється як від архітектури операційних систем Windows, так і від архітектури операційних систем Unix.

QNX Neutrino складається з мікроядра Neutrino і різних процесів, кожний з яких (також і драйвери пристроїв) виконується в своєму ізольованому просторі. Між собою такі процеси можуть обмінюватися даними за допомогою механізмів міжпроцесної взаємодії IPC (абр. від англ. Inter-Process Communication), таких як іменовані і неіменовані програмні канали, черги повідомлень, іменовані і неіменовані семафори і деякі інші.

Основна область використання QNX Neutrino – убудовані промислові комп’ютери і інше аналогічне обладнання. Завдяки використанню мікроядерної архітектури і можливості використовувати саме ті процеси, які потрібно, QNX Neutrino можна “заточувати” саме під потрібні задачі.

QNX Neutrino можна використовувати на обладнанні, яке має як “дуже малі”, так і “дуже великі” обчислювальні ресурси.

Разом з QNX Neutrino звичайно поставляється середовище розробки QNX Momentics – дуже потужний програмний інструментарій для створення

програмного забезпечення. Він утримує, зокрема, наступні складові:

- DDK (абр. від англ. Driver Development Kit) – програмні інструменти для розробки драйверів пристроїв;
- TDK (абр. від англ. Technology Development Kit) – програмні інструменти для підтримки серверів DHCP, DNS і так далі, протоколів IP Security, IP version 6 і так далі;
- BSP (абр. від англ. Board Support Packages) – програмні інструменти підтримки роботи QNX Neutrino на різних “апаратних платформах”;
- SAT (абр. від англ. System Analysis Toolkit) – програмні інструменти для профілювання;
- Embedding Tools – програмні інструменти для підтримки “убудовування”.

Для створення графічного інтерфейсу в QNX Neutrino використовується PhAB (Photon Application Builder).

1.7. Питання для самоперевірки

1. Що таке SCADA-система?
2. Які типові функції може виконувати SCADA-система?
3. Що таке людино-машинний інтерфейс?
4. Що таке мнемосхема?
5. Що таке інформаційна модель об’єкта автоматизації?
6. Що таке коефіцієнти інформативності і заповнюваності мнемосхеми?
7. В чому при створені мнемосхем полягає принцип функціональності?
8. В чому при створені мнемосхем полягає принцип значимості?
9. В чому при створені мнемосхем полягає принцип розташування?
10. В чому при створені мнемосхем полягає принцип послідовності?
11. В чому при створені мнемосхем полягає принцип частоти використання?
12. Що таке OPC-стандарти?
13. Що таке OPC-сервери і OPC-клієнти?
14. Яке основне призначення OPC-стандарту OPC DA (Data Access)?
15. Яке основне призначення OPC-стандарту OPC HDA (Historical Data Access)?
16. Яке основне призначення OPC-стандартів OPC AE (Alarms and Events), OPC Batch, OPC DX (Data eXchange), OPC Security і OPC XML-DA (XML-Data Access)?

17. Яке основне призначення OPC-стандарту OPC UA (Unified Architecture)?

1.8. Рекомендована література

Базова література: [1–6].

Додаткова література: [1–6].

2. ОСНОВНІ СЕРЕДОВИЩА ОБМІНУ ДАНИМИ, ЯКІ ВИКОРИСТОВУЮТЬСЯ В ПРОМИСЛОВІЙ АВТОМАТИЗАЦІЇ

2.1. Загальні відомості про середовища обміну даними, які використовуються в промисловій автоматизації

Серед обладнання промислової автоматизації (також і вітчизняних розробників) найчастіше використовуються наступні середовища обміну даними: кабелі на основі мідних/алюмінієвих провідників, кабелі на основі оптичного волокна і електромагнітні хвилі.

При виборі середовища обміну даними для SCADA-систем одним з найважливіших чинників є пропускна спроможність такого середовища (тобто якою максимальною кількістю даних можна обмінюватися через таке середовище за одиницю часу) – при цьому дуже бажано, щоб при обміні даних вони передаватися і прийматися з мінімальним рівнем помилок, а ті помилки, які були виявлені, виправлялися.

Залежність, яка пов’язує пропускну спроможність середовища обміну даними C (в біт/с) з його шириною смуги пропускання F (в Гц), була запропонована Клодом Елвудом Шенноном (англ. Claude Elwood Shannon):

$$C = F \log_2 \left(1 + \frac{P_\text{С}}{P_\text{Ш}}\right),$$

де $P_\text{С}$ – потужність сигналу; $P_\text{Ш}$ – потужність шуму.

Так як в будь-якому обладнанні ширина смуги пропускання F завжди є обмеженою, потужність сигналу $P_\text{С}$ не можна збільшити вище певної межі, а потужність шуму $P_\text{Ш}$ відповідно зменшити, то і пропускна спроможність середовища обміну даними C завжди є обмеженою.

Аналогічна залежність для пропускної спроможності середовища обміну даними C (в біт/с) з урахуванням ширини смуги пропускання F (в Гц) такого середовища, але без безпосереднього урахування безпосередньо рівня шуму в ньому, була запропонована Гаррі Найквістом (англ. Harry Nyquist):

$$C = F \log_2(M),$$

де M – кількість можливих інформаційних станів середовища обміну даними які можна розрізнити (ця кількість фактично і визначається рівнем шуму в такому середовищі).

2.2. Кабель на основі мідних/алюмінієвих провідників

2.2.1. Кабель на основі паралельних провідників

Одну з можливих внутрішніх будов кабелю на основі паралельних провідників наведено на рис. 2.1.

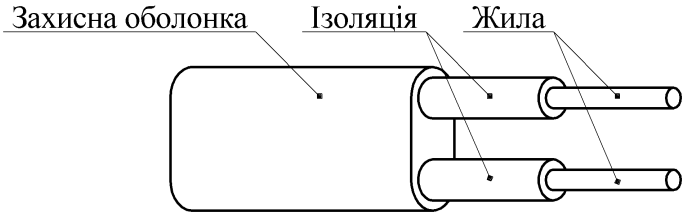


Рис. 2.1. Будова кабелю на основі паралельних провідників

У випадку такої будови кабель має дві жили, які рознесені (знаходяться на заданій відстані) одна від іншої і знаходяться в захисній оболонці, причому кожна з жил практично завжли має свою власну ізоляцію. Будова деяких кабелів на основі паралельних провідників передбачає використання більшої кількості жил (наприклад, 4, 6, 8, 10, 12, 14 і так далі), які розташовуються впритул одна до одної.

Наприклад, кабель Belden MachFlex 7101W має наступні параметри:

- кількість жил – 3 (одна з них жила заземлення) з неізольованої міді;
- калібр кожної з жил – AWG 20 (площа перерізу 0,518 мм^2);
- ізоляція кожної з жил – PVC (абр. від англ. Polyvinyl Chloride, Полівінілхлорид);
- товщина ізоляції кожної з жил – 0,015 inch (0,381 мм);
- опір кожної з жил – 8,8 Ом/1000 ft (28,87 Ом/км).

2.2.2. Кабель на основі кручених провідників

Одну з можливих внутрішніх будов кабелю на основі кручених провідників наведено на рис. 2.2.

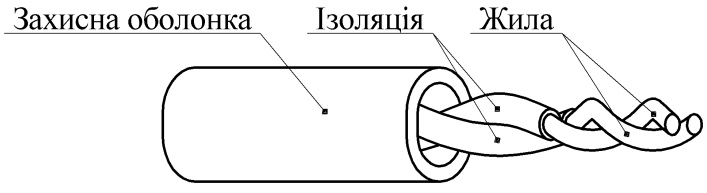


Рис. 2.2. Будова кабелю на основі кручених провідників

У випадку такої будови кабель має дві жили, які скручені одна з іншою і знаходяться в захисній оболонці, причому кожна з жил також має свою власну ізоляцію. Будова деяких кабелів на основі кручених провідників передбачає використання більшої кількості жил (наприклад, 4 або 8), які попарно скручені одна з іншою.

Наприклад, кабель Belden 973105Z має наступні параметри:

- кількість жил – 2 (скручені в 1 пару) з лудженої міді;
- калібр кожної з жил – AWG 22 (площа перерізу 0,326 мм^2);
- ізоляція кожної з жил – PE (Foam) (абр. від англ. Polyethylene (Foam), Поліетилен (спінений));
- товщина ізоляції кожної з жил – 0,029 inch (0,737 мм);
- екранування – стрічка (алюміній і полімер) і обплетення (луджена мідь);
- армування – обплетення (бронза);
- хвильовий опір – 120 Ом;
- номінальна швидкість розповсюдження сигналу – 78 % від швидкості розповсюдження електромагнітних хвиль у вакуумі.

2.2.3. Кабель на основі коаксіальних провідників

Одну з можливих внутрішніх будов кабелю на основі коаксіальних провідників наведено на рис. 2.3.

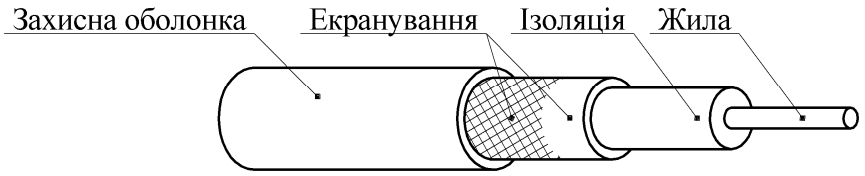


Рис. 2.3. Будова кабелю на основі коаксіальних провідників

У випадку такої будови кабель має одну жилу, розташовану по центру, і екранування, причому між жилою і екрануванням розташовується ізоляція, а навколо екранування розташовується захисна оболонка.

Наприклад, кабель Belden 1671A має наступні параметри:

- кількість жил – 1 центральна з посрібленої обмідненої сталі;
- калібр центральної жили – AWG 24 (площа перерізу 0,205 мм^2);
- ізоляція центральної жили – PTFE (абр. від англ. Polytetrafluoroethylene, Політетрафторетилен);
- діаметр ізоляції центральної жили – 0,062 inch (1,575 мм);

- екранування – стрічка (неізольована мідь) і обплетення (луджена мідь, заповнена оловом);

- опір центральної жили – 64,2 Ом/1000 ft (210,63 Ом/км);

- опір екранування – 10,2 Ом/1000 ft (33,46 Ом/км);

- ємність “центральна жила – екранування” – 29,5 пФ/ft (96,78 пФ/м);

- хвильовий опір – 50 Ом;

- номінальна швидкість розповсюдження сигналу – 70 % від швидкості розповсюдження електромагнітних хвиль у вакуумі.

Загасання електричних коливань в кабелі Belden 1671A наступне:

- на частоті 0,5 ГГц – не більше 15,0 дБ/100 ft (49,2 дБ/ 100 м);

- на частоті 1 ГГц – не більше 22,5 дБ/100 ft (73,8 дБ/ 100 м);

- на частоті 2 ГГц – не більше 32,8 дБ/100 ft (107,6 дБ/ 100 м);

- на частоті 3 ГГц – не більше 41,2 дБ/100 ft (135,2 дБ/ 100 м).

2.2.4. Еквівалентна електрична схема кабелів

При досліджені умов обміну даними в промисловій автоматизації, під час якого використовується кабель на основі мідних/алюмінієвих провідників (за умови, що такий кабель має досить велику довжину (десятки метрів і більше) і/або обмін даними відбувається на досить високій частоті (одиниці мегагерц і більше)), часто виникає необхідність представляти таке середовище обміну даними у вигляді еквівалентної електричної схеми (схеми заміщення).

Така схема є послідовним з’єднанням великої кількості однакових комірок, кожна з яких характеризується наступними параметрами: розподіленим опором R, розподіленою індуктивністю L, розподіленою провідністю G і розподіленою ємністю C (див. рис. 2.4). Значення таких параметрів звичайно наводяться на одиницю довжини кабелю.

Рис. 2.4. Еквівалентна електрична схема (схема заміщення) кабелю

Розповсюдження сигналу в такому кабелі може бути достатньо точно описане за допомогою системи з двох телеграфних рівнянь (такі рівняння

можуть бути отримані з системи рівнянь Максвела):

$$\begin{cases} \dfrac{\partial U(x,t)}{\partial x} = -L\dfrac{\partial I(x,t)}{\partial t} - RI(x,t)\,; \\ \dfrac{\partial I(x,t)}{\partial x} == -C\dfrac{\partial U(x,t)}{\partial t} - GU(x,t)\,, \end{cases}$$

де $U(x,t)$ і $I(x,t)$ – відповідно електричні напруга і струм в кабелі, які є функціями координати x і часу t.

Якщо значеннями R і G можна знехтувати (кабель без втрат), то швидкість розповсюдження сигналу в кабелі становитиме приблизно $1/\sqrt{LC}$, а хвильовий опір такого кабелю становитиме приблизно $\sqrt{C/L}$.

2.2.5. Передавачі і приймачі електричних сигналів

При досліджені умов обміну даними в промисловій автоматизації, під час якого використовується кабель на основі мідних/алюмінієвих провідників, також виникає необхідність представляти передавачі і приймачі електричних сигналів, які використовуються відповідно для передачі і приймання таких сигналів, у вигляді еквівалентних електричних схем (схем заміщення).

Передавачі напруги електричного сигналу $U_{\text{сигн.}}(t)$ можуть бути, зокрема, заземленими, незаземленими і балансними.

Еквівалентна електрична схема заземленого передавача напруги електричного сигналу $U_{\text{сигн.}}(t)$ наведена на рис. 2.5.

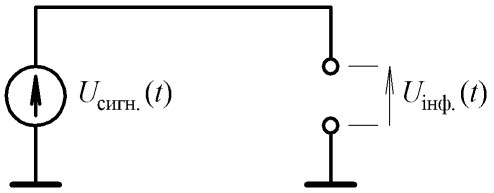


Рис. 2.5. Електрична схема заземленого передавача напруги

В такому випадку інформаційний сигнал $U_{\text{інф.}}(t)$, який приймається, вимірюється відносно “землі”. Прикладом такого передавача є лінії TxD (Transmit Data) і SG (Signal Ground) інтерфейсу RS-232 та лінії уніфікованого електричного неперервного сигналу напруги.

Еквівалентна електрична схема незаземленого передавача напруги електричного сигналу $U_{\text{сигн.}}(t)$ наведена на рис. 2.6.

В такому випадку інформаційний сигнал $U_{\text{інф.}}(t)$, який приймається,

вимірюється відносно напруги $U_{\text{пл.}}(t)$, яке є “плаваючою” (тобто такою, значення якої залежить від вимірювальної схеми і режимів її роботи та з плином часу може змінюватися). Прикладом такого передавача є термоелектричні перетворювачі, фотоелектричні елементи, ізолюючі операційні підсилювачі, розв’язувальні трансформатори і так далі.

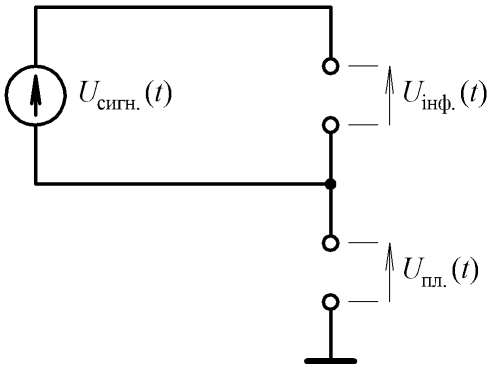
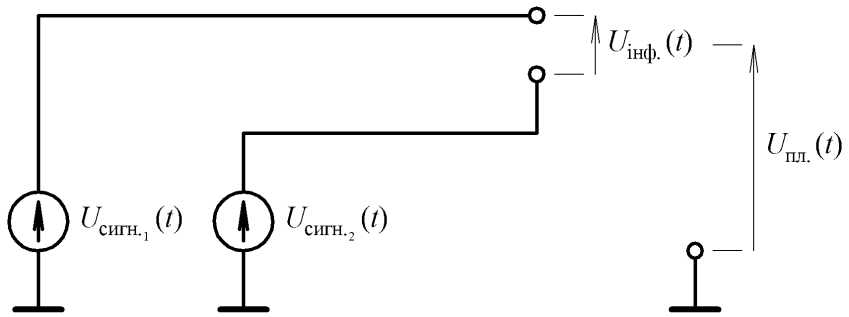
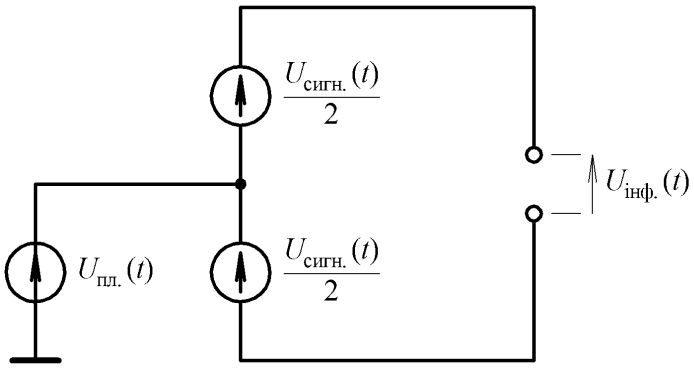


Рис. 2.6. Електрична схема незаземленого передавача напруги

Еквівалентні електричні схеми балансних передавачів напруги електричного сигналу $U_{\text{сигн.}}(t)$ наведені на рис. 2.7.



а)



б)

Рис. 2.7. Електричні схеми балансних передавачів напруги

В такому випадку інформаційний сигнал $U_{\text{інф.}}(t)$, який приймається, вимірюється відносно напруги $U_{\text{пл.}}(t)$, але при цьому він відносно цієї напруги є “збалансованим” (“симетричним”). Прикладом такого передавача є лінії A (Data+) і B (Data–) інтерфейсу RS-485.

Передавачі струму електричного сигналу можуть бути, зокрема, заземленими, незаземленими і балансними.

Еквівалентна електрична схема заземленого передавача струму електричного сигналу наведена на рис. 2.8.

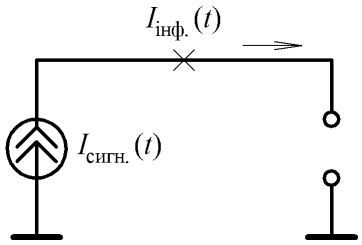


Рис. 2.8. Електрична схема заземленого передавача струму

В такому випадку інформаційний сигнал $I_{\text{інф.}}(t)$, який приймається, вимірюється відносно “землі”. Прикладом такого передавача є лінії уніфікованого електричного неперервного сигналу струму.

Еквівалентна електрична схема незаземленого передавача струму електричного сигналу наведена на рис. 2.9.

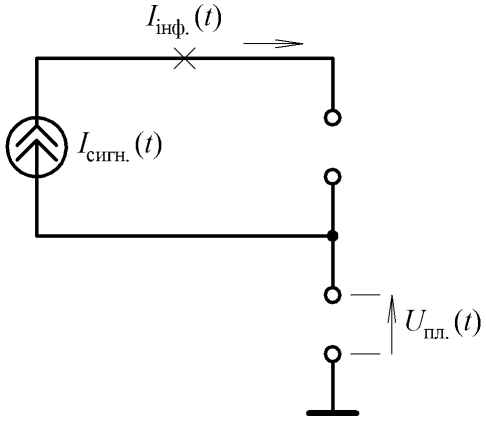


Рис. 2.9. Електрична схема незаземленого передавача струму

В такому випадку інформаційний сигнал $I_{\text{інф.}}(t)$, який приймається, вимірюється відносно напруги $U_{\text{пл.}}(t)$, яке є “плаваючою”.

Еквівалентна електрична схема балансного передавача струму електричного сигналу наведена на рис. 2.10.

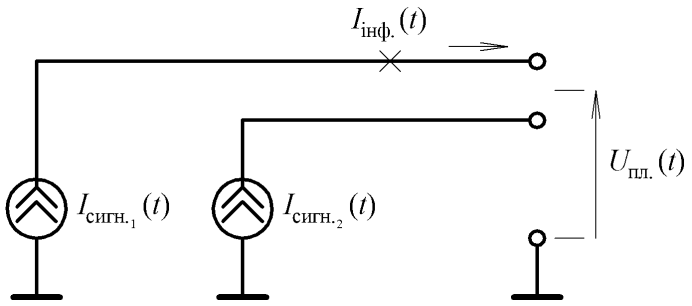


Рис. 2.10. Електрична схема балансного передавача струму

В такому випадку інформаційний сигнал $I_{\text{інф.}}(t)$, який приймається, вимірюється відносно напруги $U_{\text{пл.}}(t)$, але при цьому він відносно цієї напруги є “збалансованим” (“симетричним”).

Приймачі напруги інформаційного сигналу $U_{\text{інф.}}(t)$ можуть бути, зокрема, із заземленим живленням, диференціальними і з плаваючим живленням.

Еквівалентна електрична схема приймача напруги із заземленим живленням інформаційного сигналу $U_{\text{інф.}}(t)$ наведена на рис. 2.11.

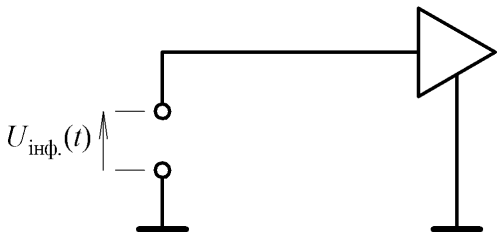


Рис. 2.11. Електрична схема приймача напруги із заземленим живленням

Відповідно інформаційний сигнал $U_{\text{інф.}}(t)$, який приймається, вимірюється відносно “землі”.

Еквівалентна електрична схема диференціального приймача напруги інформаційного сигналу $U_{\text{інф.}}(t)$ наведена на рис. 2.12.

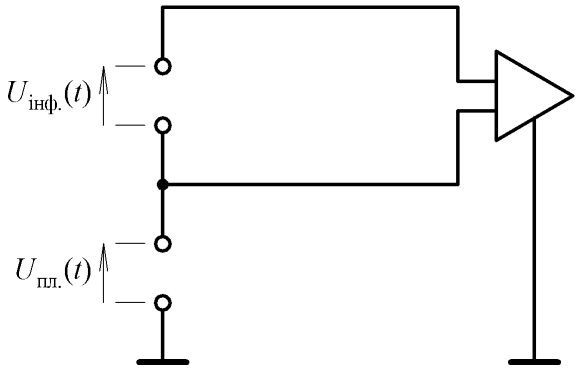


Рис. 2.12. Електрична схема диференціального приймача напруги

Відповідно інформаційний сигнал $U_{\text{інф.}}(t)$, який приймається, вимірюється відносно напруги $U_{\text{пл.}}(t)$, яке є “плаваючою”.

Еквівалентна електрична схема приймача напруги з плаваючим живленням інформаційного сигналу $U_{\text{інф.}}(t)$ наведена на рис. 2.13.

Відповідно інформаційний сигнал $U_{\text{інф.}}(t)$, який приймається, вимірюється відносно “локальної землі”, яка відрізняється від “глобальної землі” на “плаваючу” напругу $U_{\text{пл.}}(t)$.

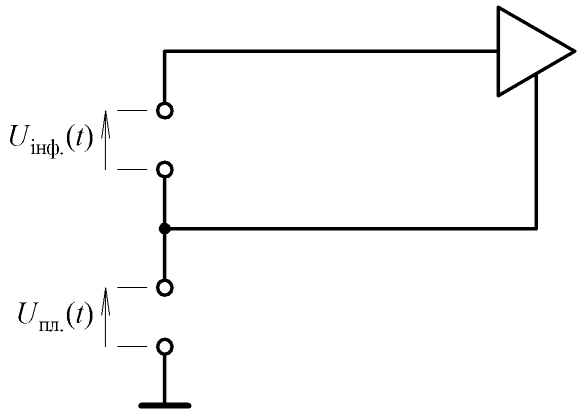


Рис. 2.13. Електрична схема приймача напруги з плаваючим живленням

Приймачі струму інформаційного сигналу $I_{\text{інф.}}(t)$ можуть бути, зокрема, із заземленим живленням, диференціальними і з плаваючим живленням.

Еквівалентна електрична схема приймача струму із заземленим живленням інформаційного сигналу $I_{\text{інф.}}(t)$ наведена на рис. 2.14.

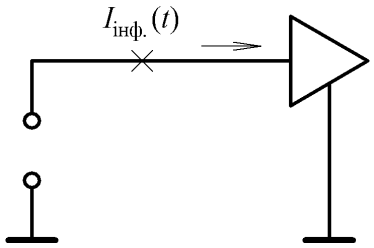


Рис. 2.14. Електрична схема приймача струму із заземленим живленням

Відповідно інформаційний сигнал $I_{\text{інф.}}(t)$, який приймається, вимірюється відносно “землі”.

Еквівалентна електрична схема диференціального приймача струму інформаційного сигналу $I_{\text{інф.}}(t)$ наведена на рис. 2.15.

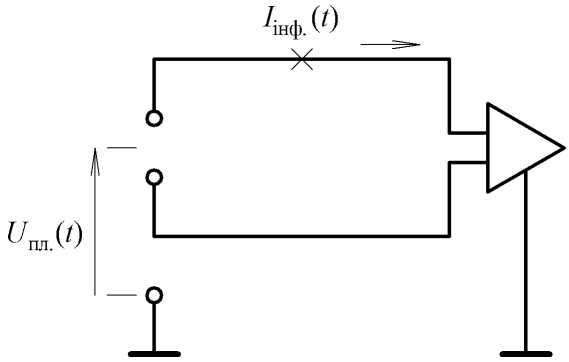


Рис. 2.15. Електрична схема диференціального приймача струму

Відповідно інформаційний сигнал $I_{\text{інф.}}(t)$, який приймається, вимірюється відносно напруги $U_{\text{пл.}}(t)$, яке є “плаваючою”.

Еквівалентна електрична схема приймача струму з плаваючим живленням інформаційного сигналу $I_{\text{інф.}}(t)$ наведена на рис. 2.16.

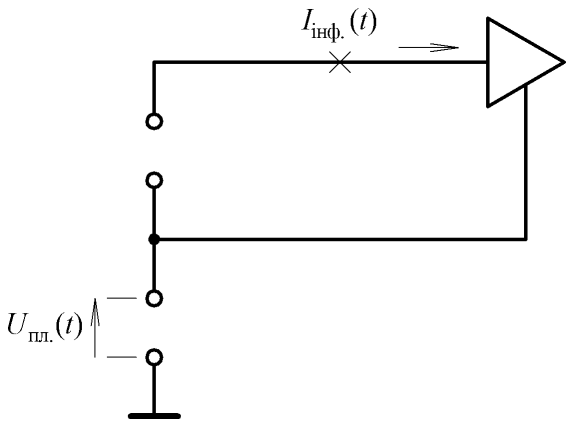


Рис. 2.16. Електрична схема приймача струму з плаваючим живленням

Відповідно інформаційний сигнал $I_{\text{інф.}}(t)$, який приймається, вимірюється відносно “локальної землі”, яка відрізняється від “глобальної землі” на “плаваючу” напругу $U_{\text{пл.}}(t)$.

2.2.6. Боротьба із завадами

Якщо монтаж кабелів виконаний без урахування боротьби із завадами, то середовище обміну даними може виявитися непрацездатним. Зокрема, до 40 % випадків простою обладнання промислової автоматизації повязані з неправильним виконанням заземлення – розповсюджуючись у вигляді електричного струму по заземлюючим провідникам, завади створюють на кінцях таких провідників паразитну електричну напругу (різницю потенціалів).

Основними джерелами завад в промисловій автоматизації є наступні

джерела: блискавки, статична електрика, електромагнітне випромінення, мережа електричного живлення частотою 50 або 60 Гц, перехідні процесі внаслідок підключення до мережі електричного живлення і відключення від неї потужних навантажень, трибоелектрика, гальванічні пари, термоелектричний ефект, електролітичні процеси, рух провідників в електромагнітному полі, обладнання, яке створює електромагнітний “шум” і так далі. Причому мережа електричного живлення частотою 50 або 60 Гц, наприклад, є найбільшим джерелом наступних завад: “фоновий” шум з частотою 50 або 60 Гц, викиди електричної напруги, короткочасні загасаючі коливання (зокрема, внаслідок підключення і відключення індуктивних навантажень), високочастотний шум (зокрема, накладений на електричну напругу частотою 50 або 60 Гц), інфранизькочастотний шум (зокрема, внаслідок повільної зміни дійсного (ефективного) значення електричної напруги частотою 50 або 60 Гц), довготривалі спотворення форми електричної напруги частотою 50 або 60 Гц (зокрема, через насичення осердь трансформаторів, дроселів і так далі).

Основні шляхи впливу завад на обладнання промислової автоматизації (зокрема, на процес обміну даними) можна класифікувати наступним чином: вплив через кондуктивні, індуктивні і ємнісні зв’язки, вплив через нееквіпотенціальність “земель”, зовнішні електромагнітні наведення.

Кондуктивні завади – це завади, які передаються між сусідніми електричними ланцюгами через різні провідники (доріжки на друкованій платі, проводи живлення, “загальні” проводи, проводи заземлення, екранування і так далі), які є загальними для таких ланцюгів (див. рис. 2.17).

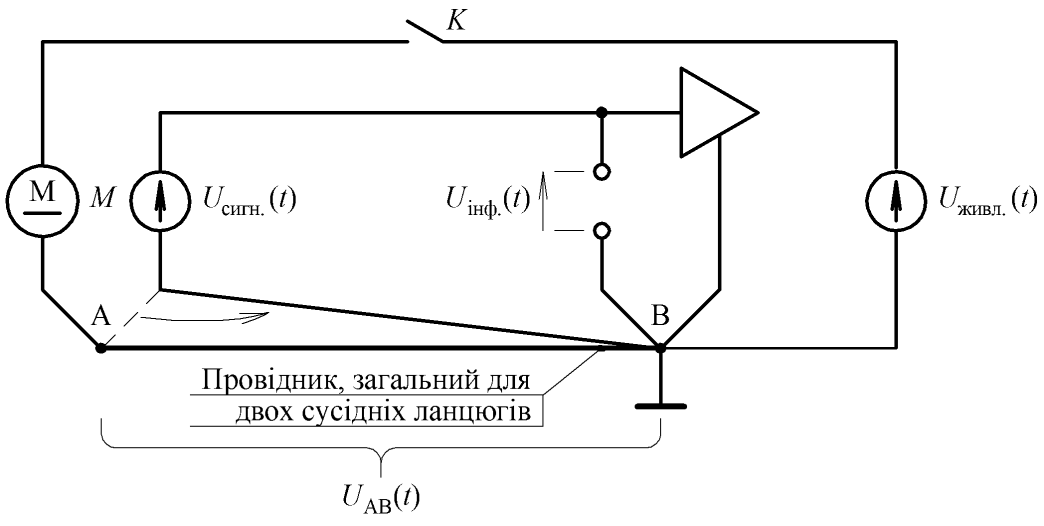


Рис. 2.17. Проходження кондуктивної завади між сусідніми електричними ланцюгами

При розімкненому ключі K струм від напруги живлення $U_{\text{живл.}}(t)$ через електричний двигун M не протікає. Так як вхідний опір приймача напруги звичайно є великим, то електричний сигнал $U_{\text{сигн.}}(t)$ створює через провідник, загальний для двох сусудніх лагцюгів (один з яких утримує електричний сигнал $U_{\text{сигн.}}(t)$, а іншій – електричний двигун M), невеликий струм. Відповідно, $U_{\text{AB}}(t) \approx 0$.

В такому випадку, так як інформаційний сигнал $U_{\text{інф.}}(t) = U_{\text{сигн.}}(t) + U_{\text{AB}}(t)$, то звідси $U_{\text{інф.}}(t) \approx U_{\text{сигн.}}(t)$. Тобто інформаційний сигнал $U_{\text{інф.}}(t)$ практично співпадає з електричним сигналом $U_{\text{сигн.}}(t)$.

При замкненому ключі K від напруги живлення $U_{\text{живл.}}(t)$ через електричний двинун M протікає достатньо великий струм. Причому він протікає і через провідник, загальний для двох сусідніх ланцюгів. Відповідно, $U_{\text{AB}}(t) \neq 0$, і інформаційний сигнал $U_{\text{інф.}}(t)$ відрізняється від електричного сигналу $U_{\text{сигн.}}(t)$ дуже сильно.

Зменшити вплив таких завад на обладнання промислової автоматизації можна шляхом використання “силових земель”, “аналогових земель”, “цифрових земель” (можливо кількох з них одночасно), які повинні з’єднуватися тільки в одній точці (в крайньому випадку через провідники з великим перерізом і малою довжиною, які мають відповідно малий опір) (див. рис. 2.18).

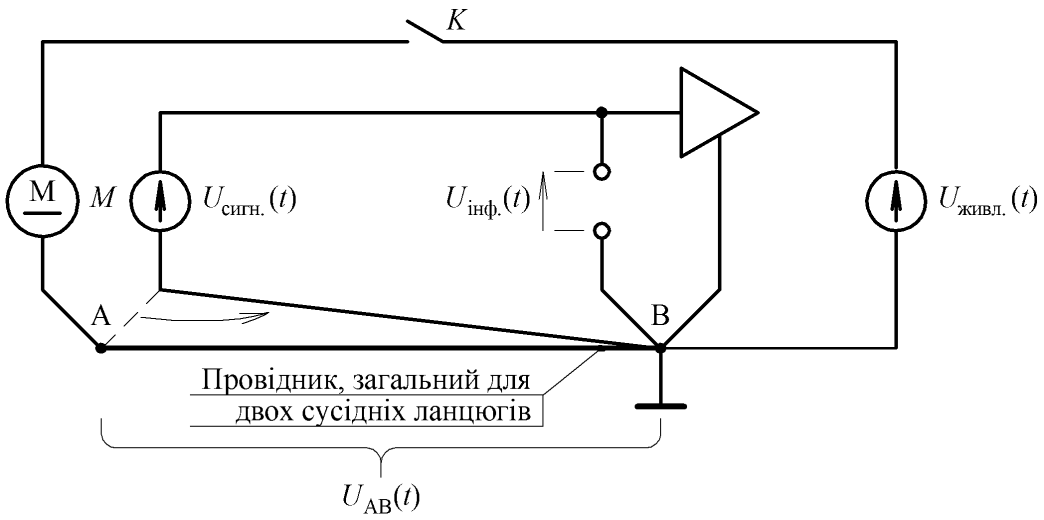


Рис. 2.18. Зменшення впливу кондуктивної завади між сусідніми електричними ланцюгами

Індуктивні завади – це завади, які передаються між сусідніми електричними ланцюгами через електромагнітне поле (див. рис. 2.19).

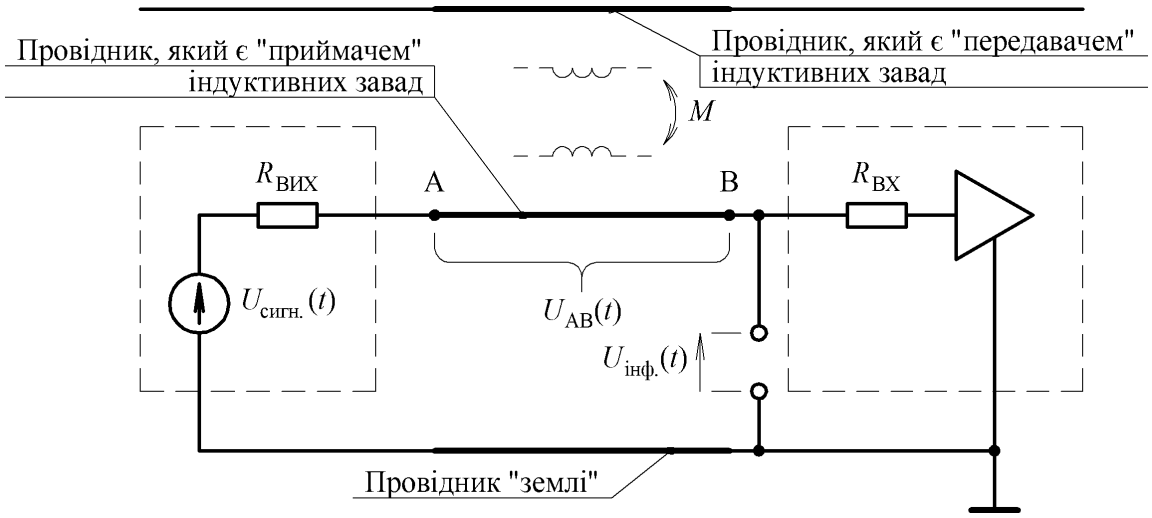


Рис. 2.19. Проходження індуктивної завади між сусідніми електричними ланцюгами

Зміна струму через провідник, який є “передавачем” індуктивних завад, приводить до зміни магнітного поля через контур, який утворений вихідним опором $R_{ВИХ}$, провідником, який є “приймачем” індуктивних завад, вхідним опором $R_{ВХ}$ і провідником “землі” (внаслідок чого в ньому виникає струм).

Боротися з індуктивною завадою можна, зокрема, зменшуючи площу такого контуру (наприклад, скручуючи між собою провідники, які є “приймачем” індуктивних завад і “землею”) – тобто зменшуючи коефіцієнт взаємної індукції M (див. рис. 2.20).

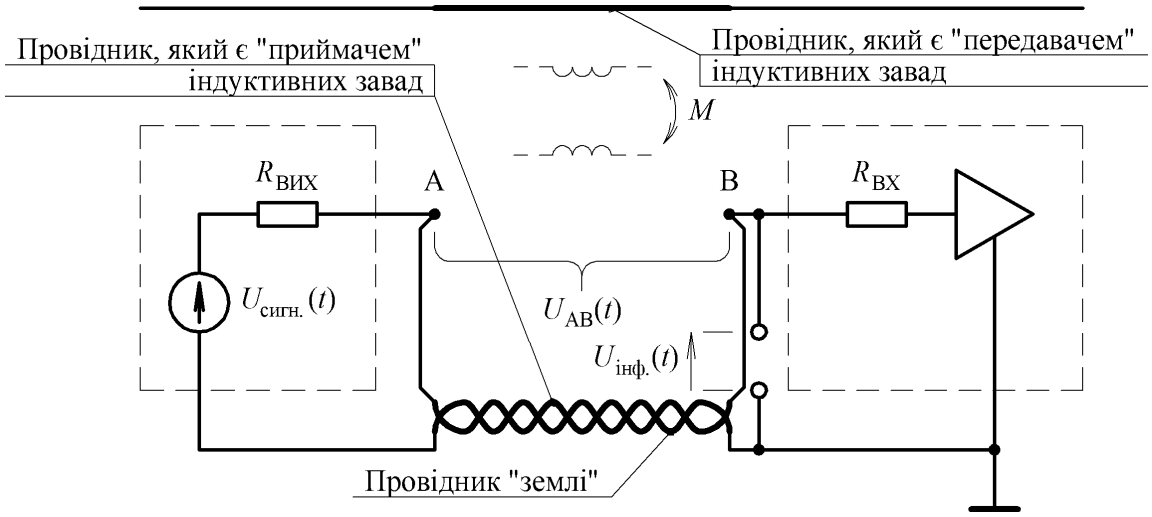


Рис. 2.20. Зменшення впливу індуктивної завади між сусідніми електричними ланцюгами

Загалом для зменшення індуктивних завад електричний сигнал краще передавати за допомогою ідеального джерела струму (коли $R_{ВИХ} \to \infty$).

Ємнісні завади – це завади, які також передаються між сусідніми електричними ланцюгами через електромагнітне поле (див. рис. 2.21).

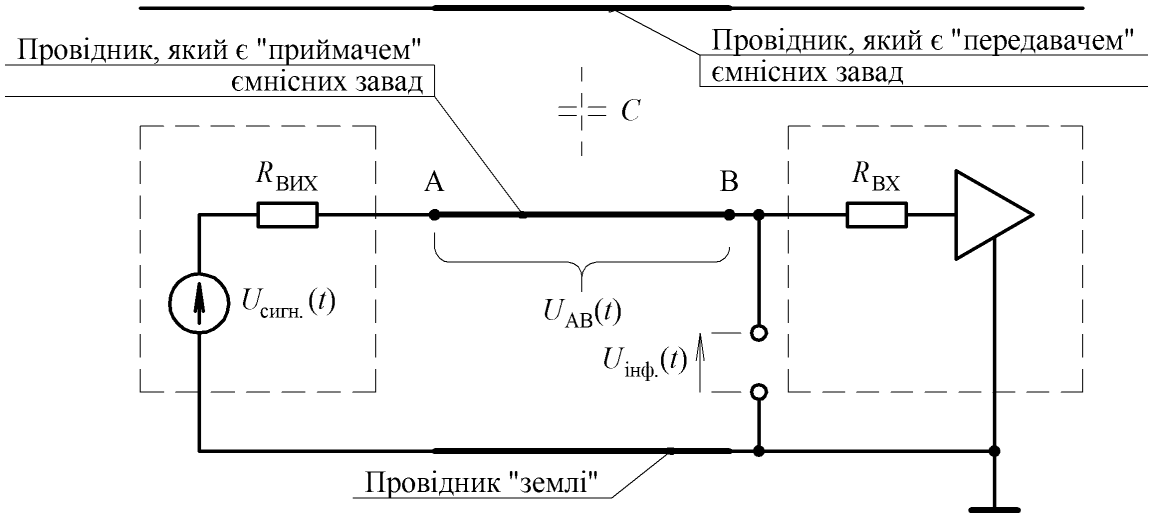


Рис. 2.21. Проходження ємнісної завади між сусідніми електричними ланцюгами

Зміна напруги між провідником, який є “передавачем” ємнісних завад, і провідником, який є “приймачем” ємнісних завад, приводить до зміни електричного поля між такими провідниками (внаслідок чого в провіднику, який є “приймачем” ємнісних завад, виникає струм).

Боротися з ємнісною завадою можна, зокрема, збільшуючи відстань між провідниками, які є “передавачем” і “приймачем” ємнісних завад – тобто зменшуючи ємність C (див. рис. 2.22).

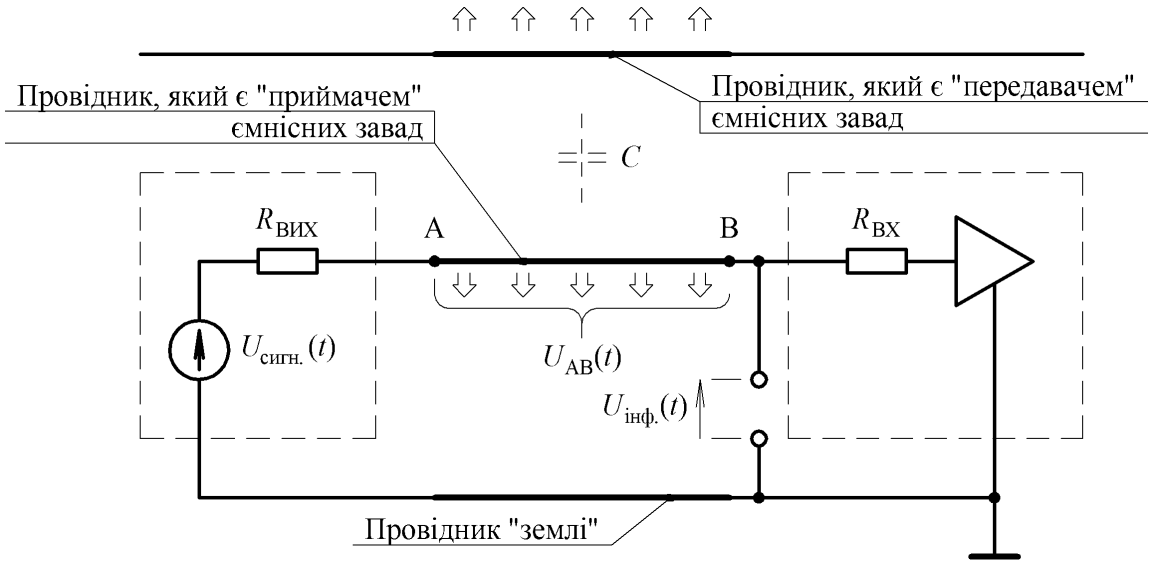


Рис. 2.22. Зменшення впливу ємнісної завади між сусідніми електричними ланцюгами

Загалом для зменшення ємнісних завад електричний сигнал краще передавати за допомогою ідеального джерела напруги (коли $R_{\text{ВИХ}} \to 0$).

Завади, які впливають на обладнання промислової автоматизації, мають діапазон частот від нуля до одиниць гігагерц. На аналогові частини такого обладнання звичайно впливають завади, які мають максимальні частоти до десятків кілогерц, на цифрові частини – які мають максимальні частоти до сотень мегагерц.

Завади, які мають максимальні частоти до одиниць гігагерц, безпосередньо на обладнання промислової автоматизації практично не впливають, але внаслідок нелінійних і багатьох інших ефектів вони часто є джерелом інших (більш низькочастотних) завад.

Основною характеристикою завад є залежність “спектральна густина потужності – частота”. Середньоквадратичне значення електричної напруги E завади, якщо відомі її спектральна густина потужності $e^2(f)$ та нижня межа $f_{\text{Н}}$ і верхня межа $f_{\text{В}}$ спектра, визначається наступним чином:

$$E = \sqrt{\int_{f_{\text{Н}}}^{f_{\text{В}}} e^2(f)df}\,.$$

Середньоквадратичне значення електричного струму I завади, якщо відомі її спектральна густина потужності $i^2(f)$ та нижня межа $f_{\text{Н}}$ і верхня межа $f_{\text{В}}$ спектра, визначається аналогічним чином:

$$I = \sqrt{\int_{f_{\text{Н}}}^{f_{\text{В}}} i^2(f)df}\,.$$

Одним з шляхів зменшення E при незмінному значенні $e^2(f)$ (або I при незмінному значенні $i^2(f)$)) є зменшення різниці $f_{\text{В}} - f_{\text{Н}}$.

Зменшити вплив завад на обладнання промислової автоматизації можна як шляхом заземлення такого обладнання (з’єднання з ґрунтом землі або “загальним” провідником електричної системи, відносно якої вимірюється електрична напруга (різниця потенціалів)), так і шляхом екранування такого обладнання (зменшення впливу на нього паразитного електромагнітного поля в певному обмеженому просторі за рахунок використання “бар’єрів” з провідних і/або магнітних матеріалів).

2.2.7. Загальні рекомендації щодо обміну аналоговими і дискретними сигналами

При обміні аналоговими і дискретними сигналами для того, щоб такий обмін був надійним, неохідно дотримуватися наступних загальних рекомендацій.

1. Кабелі, які використовуються для обміну аналоговими і дискретними сигналами, повинні мати мінімально можливу довжину. При обміні аналоговими сигналами дуже бажано використовувати диференціальне підключення.

2. Кабелі, через які проходить невеликий струм, повинні розташовуватися окремо (на відстані, не меншої від заданої) від кабелів, через які проходить великий струм.

3. При великих відстанях і/або на відкритій місцевості замість кабелів на основі мідних/алюмінієвих провідників дуже бажано використовувати кабелі на основі оптичного волокна.

4. Аналого-цифрові і цифро-аналогові перетворювачі, модулі уведення/виведення і інше аналогічне обладнання промислової автоматизації повинні мати гальванічну розв’язку, при цьому вони повинні розташовуватися максимально близько до відповідних їм вимірювальних перетворювачів, датчиків, сенсорів і так далі. Для здійснення обміну даними перевага повинна надаватися цифровим інтерфейсам.

5. Якщо необхідно виконати заземлення кабелів, через які здійснюється обмін аналоговими і дискретними сигналами, а також кабелів, через які здійснюється живлення, то його необхідно виконати тільки в одній точці (якщо це неможливо, для такого заземлення потрібно використати мідну шину з великим перерізом і малою довжиною). При виконанні заземлення його довжина повинна бути мінімальною, при цьому не повинні утворюватися замкнені контури.

6. Смуги пропускання вхідних і вихідних ланцюгів обладнання промислової автоматизації, яке здійснює обмін даними, повинні обмежуватися значеннями, які необхідні для повноцінної роботи такого обладнання, і не бути більш широкими.

7. Для обладнання промислової автоматизації, яке є дуже чутливим до завад, необхідно використовувати живлення від батарей і/або акумуляторів.

8. При обміні аналоговими і дискретними сигналами заземлення не

повинно використовуватися в якості нульового рівня відліку напруги (різниці потенціалів).

9. Для обладнання промислової автоматизації, яке гальванічно розв'язане, необхідно уникнення накопичення статичних розрядів шляхом здійснення його окремого заземлення.

10. Обладнання промислової автоматизації, яке потребує заземлення, може бути розділене на кілька незалежних груп, що мають приблизно однакову потужність, і кожна з груп може бути заземлена незалежно від іншої групи.

2.3. Кабель на основі оптичного волокна

Одну з можливих внутрішніх будов кабелю на основі оптичного волокна наведено на рис. 2.23.

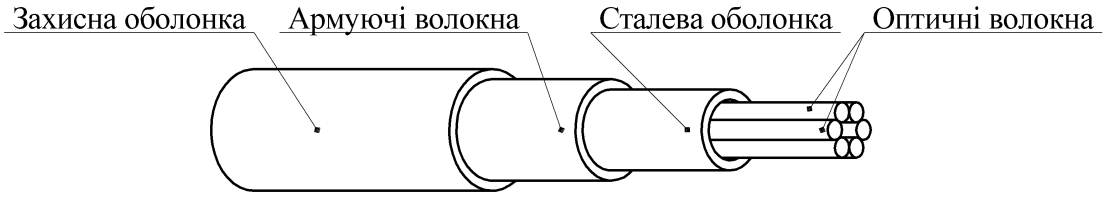


Рис. 2.23. Будова кабелю на основі оптичного волокна

У випадку такої будови кабель звичайно має кілька оптичних волокон (іноді більше сотні), які розташовуються впритул одне до одного, і кілька послідовно розташованих навколо таких волокон оболонок, кожна з яких має своє призначення (наприклад, сталева оболонка, армуючі волокна і так далі).

Наприклад, кабель Belden FD2D024R9 має наступні параметри:

- кількість оптичних волокон – 24;
- тип кожного оптичного волокна – OM2 (багатомодове для внутрішнього і зовнішнього застосування);
- діаметр внутрішнього осердя кожного оптичного волокна – 50 мкм;
- діаметр зовнішнього осердя кожного оптичного волокна – 125 мкм;
- діаметр захисної оболонки кожного оптичного волокна – 900 мкм;

Загасання електричних коливань в кабелі FD2D024R9 наступне:

- на довжині хвилі 850 нм – не більше 3,5 дБ/км;
- на довжині хвилі 1300 нм – не більше 1,25 дБ/км.

Широкосмуговість кабелю FD2D024R9 по методу RML (абр. від англ. Restricted Modal Launch):

- на довжині хвилі 850 нм – не менше 950 МГц · км;

- на довжині хвилі 1300 нм – не менше 500 МГц · км.

Широкосмуговість кабелю FD2D024R9 по методу OFL (абр. від англ. Overfilled Launch)

- на довжині хвилі 850 нм – не менше 700 МГц · км;
- на довжині хвилі 1300 нм – не менше 500 МГц · км.

2.4. Електромагнітні хвилі

В інфрачервоному діапазоні електромагнітних хвиль може використовуватися технологія IrDA (абр. від англ. Infrared Data Association).

Така технологія має наступні типові технічні характеристики:

- відстань при обміні даними – до 1 м;
- швидкість обміну даними – до 1 Гбіт/с;
- відхилення між осями передавача і приймача електромагнітних хвиль – до ± 15 °;
- довжина електромагнітних хвиль – (850 ... 900) нм;
- наявність інфрачервоних фільтрів.

Електромагнітні хвилі від передавача до приймача можуть проходити наступним чином:

- шляхом проходження по прямій лінії;
- шляхом відбиття по ламаній лінії від певних перешкод (наприклад, від стелі);
- шляхом розсіювання від певних перешкод (наприклад, від стін).

В UHF (абр. від англ. Ultra High Frequency, Ультрависокі частоти) і SHF (Super High Frequency, Надвисокі частоти) діапазонах електромагнітних хвиль може використовуватися технологія Wi-Fi (абр. від англ. Wireless Fidelity).

Така технологія має наступні типові технічні характеристики:

- відстань при обміні даними – до (10 ... 30) м;
- швидкість обміну даними – до 1 Гбіт/с;
- діапазон електромагнітних хвиль – 2,4 ГГц, 5 ГГц і 6 ГГц.

В UHF діапазоні електромагнітних хвиль також може використовуватися технологія Bluetooth.

Така технологія має наступні типові технічні характеристики:

- відстань при обміні даними – до (10 ... 30) м;
- швидкість обміну даними – до 2 Мбіт/с;
- діапазон електромагнітних хвиль – 2,4 ГГц.

2.5. Структуровані кабельні системи

СКС (Структурована кабельна система) – це сукупність елементів комутації (кабелів, рознімачів, кріплень, кабельних каналів, рукавів і лотків, розподільних шаф, розеток і так далі), які призначені для забезпечення обміну даними між різним обладнанням промислової автоматизації. Звичайно СКС використовуються для об'єднання між собою кількох виробничих і/або невиробничих будівель, в межах кожної будівлі – кількох поверхів, в межах кожного поверху – кількох приміщень (див. рис. 2.24).

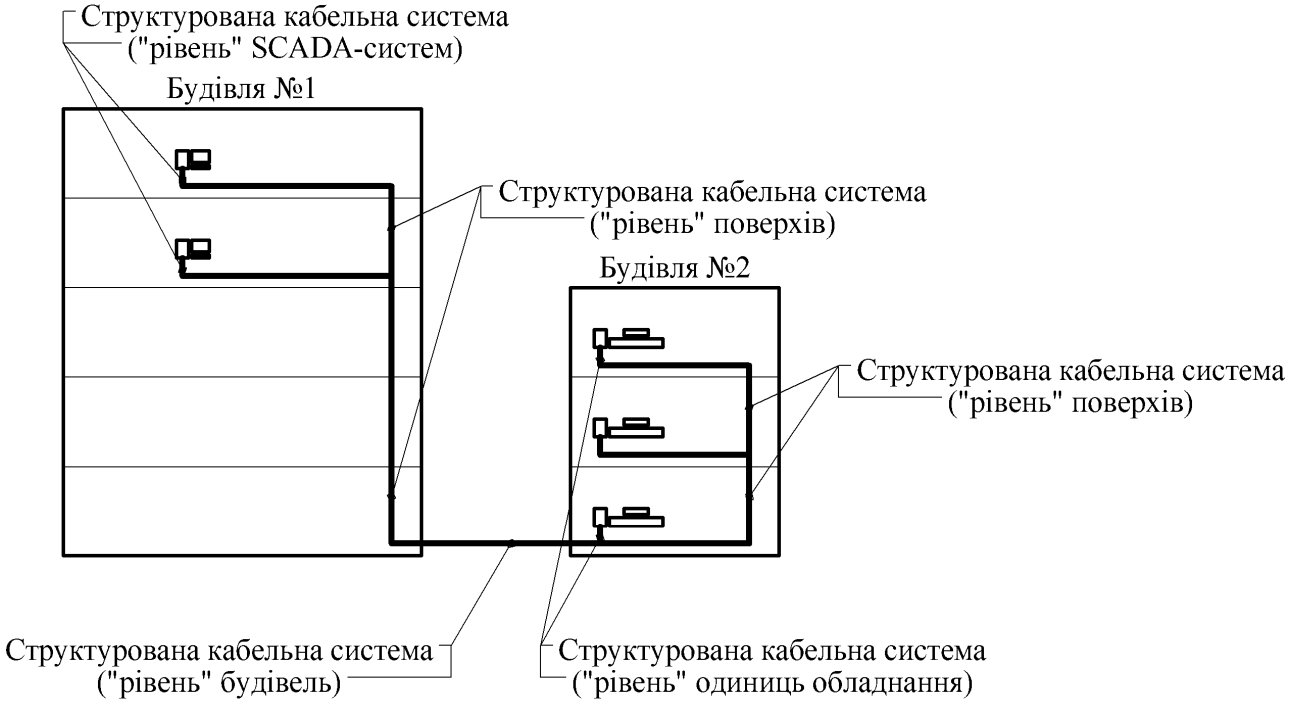


Рис. 2.24. Приклад СКС на виробництві

СКС використовуються також і для під'єднання до обладнання промислової автоматизації, яке знаходиться зовні будівель. До СКС часто відносять таке необхідне для їх роботи обладнання, як повторювачі, перетворювачі інтерфейсів, комутатори, маршрутизатори, шлюзи і так далі.

СКС проєктуються і експлуатуються не тільки для потреб SCADA-систем, але і для, зокрема, наступних потреб:

- систем електричного живлення змінного і постійного струмів;
- систем штучного освітлення;
- систем пожежної сигналізації;
- систем охоронної сигналізації;
- систем контролю доступу;

- систем відеоспостереження;

- систем телефонії.

Проєктування СКС звичайно передбачає створення наступних основних документів:

- планів трас прокладання кабелів на міських і заміських ділянках;

- планів профілів кабельних переходів;

- планів розташування площадок регенераційних (підсилювальних) пунктів;

- схем прокладання кабелів через мережу кабельних каналізацій і колекторів;

- схем магістральних і локальних кабельних мереж;

- будову кабельних уводів у виробничі і/або невиробничі будівлі.

2.6. Питання для самоперевірки

1. Що таке середовище обміну даними?

2. Що таке пропускна спроможність середовища обміну даними?

3. Що таке смуга пропускання середовища обміну даними?

4. Яку будову має кабель на основі паралельних провідників?

5. Яку будову має кабель на основі кручених провідників?

6. Яку будову має кабель на основі коаксіальних провідників?

7. Що таке еквівалентна електрична схема кабелів?

8. Що таке еквівалентні електричні схеми передавачів електричних сигналів?

9. Що таке еквівалентні електричні схеми приймачів електричних сигналів?

10. Як здійснюється боротьба із завадами при виконанні монтажу кабелів?

11. Які джерела є основними джерелами завад в промисловій автоматизації?

12. Що таке кондуктивні завади, як можна зменшити їх вплив на обладнання промислової автоматизації?

13. Що таке індуктивні завади, як можна зменшити їх вплив на обладнання промислової автоматизації?

14. Що таке ємнісні завади, як можна зменшити їх вплив на обладнання промислової автоматизації?

15. Яких загальних рекомендацій необхідно дотримуватися для того, щоб обмін аналоговими і дискретними сигналами був надійним?

16. Яку будову має кабель на основі оптичного волокна?

17. Які особливості має обмін даними за допомогою електромагнітних хвиль?

18. Що таке структурована кабельна система?

2.7. Рекомендована література

Базова література: [1, 4, 6].

Додаткова література: [1, 5, 6].

3. ОСНОВНІ ІНТЕРФЕЙСИ, ЯКІ ВИКОРИСТОВУЮТЬСЯ В ПРОМИСЛОВІЙ АВТОМАТИЗАЦІЇ

3.1. Загальні відомості про інтерфейси, які використовуються в промисловій автоматизації

Серед обладнання промислової автоматизації (також і вітчизняних розробників) найчастіше використовуються наступні інтерфейси: аналогові інтерфейси (уніфіковані електричні неперервні сигнали напруги і струму) і цифрові інтерфейси (RS-232, RS-485 і Ethernet). Інші цифрові інтерфейси, такі, наприклад, як Universal Serial Bus (USB) і Controller Area Network (CAN), звичайно використовуються не так часто. Потреба в швидкостях обміну даними, які можуть забезпечити цифрові інтерфейси, в промисловій автоматизації рідко перевищує 100 Мбіт/с, а основними вимогами до таких інтерфейсів є надійна роботи в складних умовах експлуатації.

Цифрові інтерфейси звичайно відповідають одному (іноді кільком) рівням моделі OSI. Відповідно, для більшості з них визначаються такі поняття, як типи використовуваних рознімачів, рівні напруг, способи кодування, рекомендовані типи кабелів і так далі. Деякі з інтерфейсів можуть утворювати з'єднання тільки у вигляді "точка-точка" (наприклад, RS-232), деякі – у вигляді як "точка-точка", так і "мережа" (наприклад, RS-485).

При передачі цифровими інтерфейсами кадрів (які є основними "носіями інформації" при здійсненні обміну даними) використовуються різні способи кодування таких кадрів, за рахунок чого можна розв'язати деякі з наступних задач:

- досягти мінімальної ширини смуги пропускання кодованого сигналу;
- досягти мінімальної потужності передавача кодованого сигналу;
- забезпечити надійну синхронізацію між передавачем і приймачем кодованого сигналу;
- забезпечити максимальну стійкість до завад;
- забезпечити виявлення і по можливості виправлення максимальної кількості "неправильних" бітів.

Найпростіший код NRZ (абр. від англ. Non return to zero, Без повернення до нуля), наприклад, використовується в інтерфейсі RS-232 (див. рис. 3.1). При використанні коду NRZ лог. 1 і лог. 0 кодуються відповідно високим і низьким рівнями електричного сигналу. Основними недоліками коду NRZ є наявність в

електричному сигналі постійної складової, а також погана синхронізація передавача з приймачем.

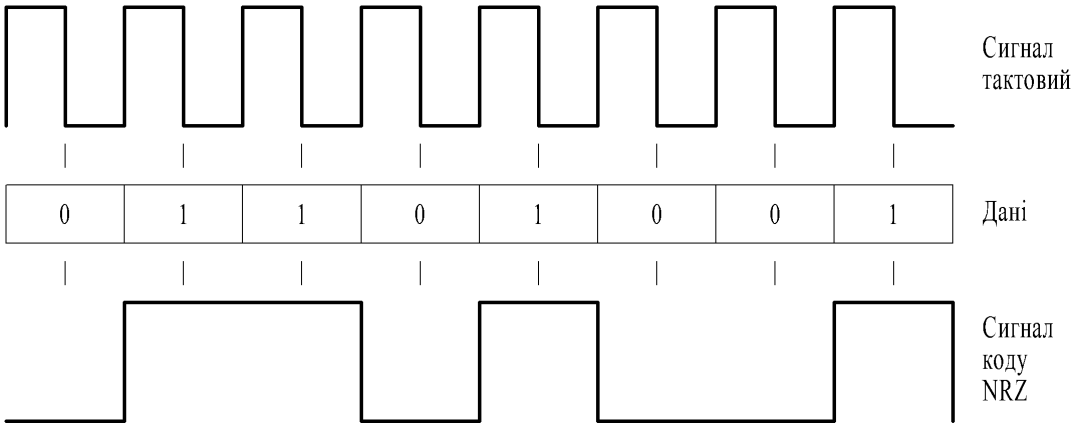


Рис. 3.1. Кодування при використанні коду NRZ

Манчестерський код (англ. Manchester code) використовується в деяких інтерфейсах Ethernet, які працюють на швидкості 10 Мбіт/с (див. рис. 3.2). При використанні манчестерського коду лог. 1 кодується переходом електричного сигналу з низького рівня на високий рівень, а лог. 0 – переходом з високого рівня на низький. Одним з недоліків манчестерського коду є широкий частотний спектр, джерелом якого є необхідність постійного виконання таких переходів.

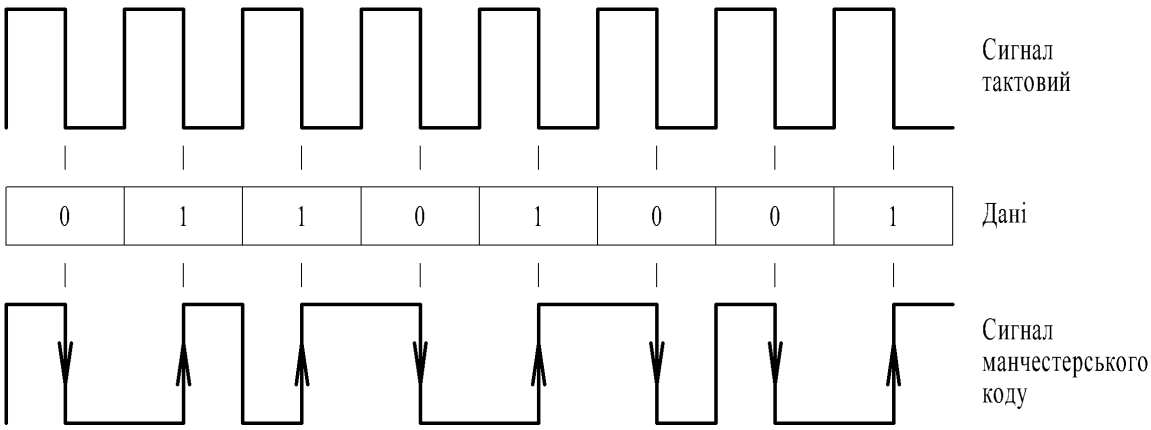


Рис. 3.2. Кодування при використанні манчестерського коду

Код NRZI (абр. від англ. Non return to zero invertive, Без повернення до нуля) також використовується в деяких інтерфейсах Ethernet, які працюють на швидкості 100 Мбіт/с (див. рис. 3.3). При використанні коду NRZI звичайно лог. 1 кодується переходом електричного сигналу з поточного рівня на протилежний (тобто з низького рівня на високий рівень або навпаки), а лог. 0 –

відсутністю такого переходу (див. рис. 3.3, а). В деяких випадках, навпаки, лог. 0 кодується переходом електричного сигналу з поточного рівня на протилежний (тобто з низького рівня на високий рівень або навпаки), а лог. 1 – відсутністю такого переходу (див. рис. 3.3, б). Основні недоліки коду NRZI такі ж самі, як і коду NRZ – наявність в електричному сигналі постійної складової і погана синхронізація передавача з приймачем.

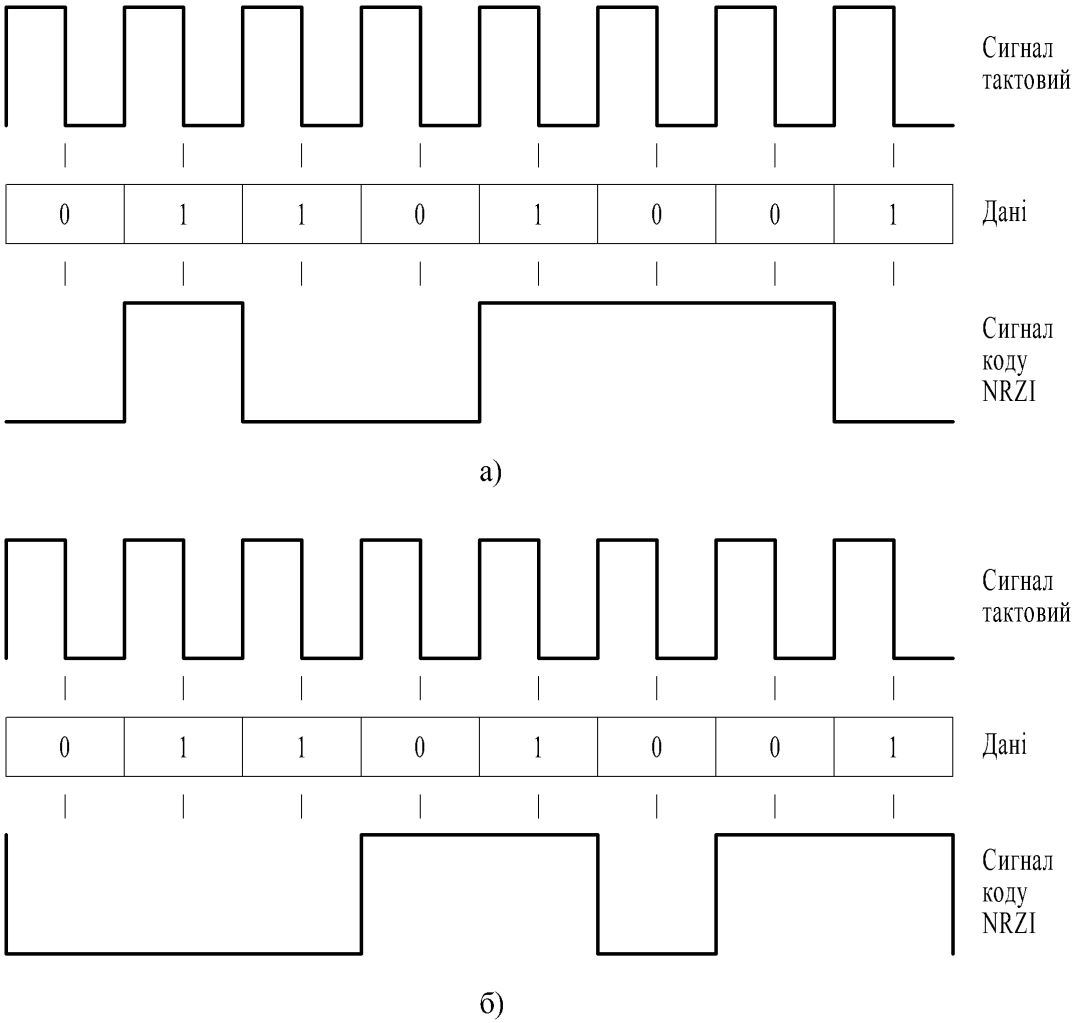


Рис. 3.3. Кодування при використанні коду NRZI

Погана синхронізація передавача з приймачем, які використовують код NRZI, особливо проявляється тоді, коли передається багато лог. 0 (і, відповідно, переходи електричного сигналу з поточного рівня на протилежний не відбуваються). В такому випадку може додатково використовуватися код 4B/5B, який полягає в додаванні до кожної тетради даних, які передаються (4-ри біта), 5-го біта. З отриманих 32-ох кодових комбінацій виключають ті, які мають підряд 3-ри або більшу кількість лог. 0, а іншим ставлять у відповідність цифри від 0 до F, отримуючи таким чином потрібну таблицю кодування.

3.2. Інтерфейс у вигляді уніфікованих електричних неперервних сигналів напруги і струму

Уніфіковані сигнали – сигнали, які використовуються для створення взаємодії між різним обладнанням промислової автоматизації “за єдиними правилами”.

Уніфіковані електричні неперервні сигнали напруги – сигнали напруги з межами допустимих змін $(0 \dots 0{,}01)$ В, $(0 \dots 0{,}05)$ В, $(0 \dots 0{,}1)$ В, $(0 \dots 1)$ В, $(-1 \dots 1)$ В, $(0 \dots 5)$ В, $(1 \dots 5)$ В, $(-5 \dots 5)$ В, $(0 \dots 10)$ В, $(-10 \dots 10)$ В, $(2{,}4 \dots 12{,}6)$ В і $(-2{,}4 \dots -12{,}6)$ В. В якості середовища обміну даними при використані інтерфейсів у вигляді уніфікованих електричних неперервних сигналів напруги звичайно використовуються кабелі на основі паралельних провідників. Такі інтерфейси є односпрямованими, тобто сигнал напруги може передаватися тільки в одному напрямі (від передавача до приймача) і звичайно використовуються тоді, коли відстань між обладнанням промислової автоматизації невелика (так як в такому випадку опір жил кабелів повинен бути врахованим) (див. рис. 3.4).

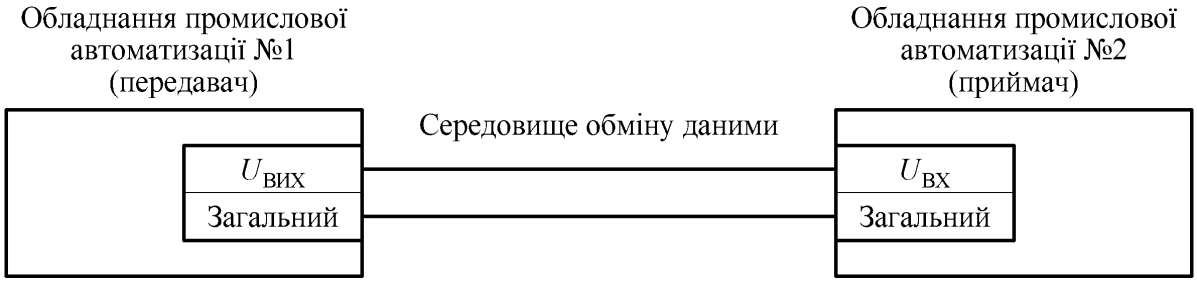


Рис. 3.4. Обмін даними за допомогою уніфікованих електричних неперервних сигналів напруги

Уніфіковані електричні неперервні сигнали струму – сигнали струму з межами допустимих змін $(0 \dots 5)$ мА, $(-5 \dots 5)$ мА, $(0 \dots 20)$ мА, $(4 \dots 20)$ мА, $(-20 \dots 20)$ мА і $(-100 \dots 100)$ мА. В якості середовища обміну даними при використані інтерфейсів у вигляді уніфікованих електричних неперервних сигналів струму також звичайно використовуються кабелі на основі паралельних провідників. Такі інтерфейси також є односпрямованими, тобто сигнал напруги може передаватися тільки в одному напрямі (від передавача до приймача) і звичайно використовуються тоді, коли відстань між обладнанням промислової автоматизації велика (так як в такому випадку опір жил кабелів може бути не врахованим) (див. рис. 3.5).

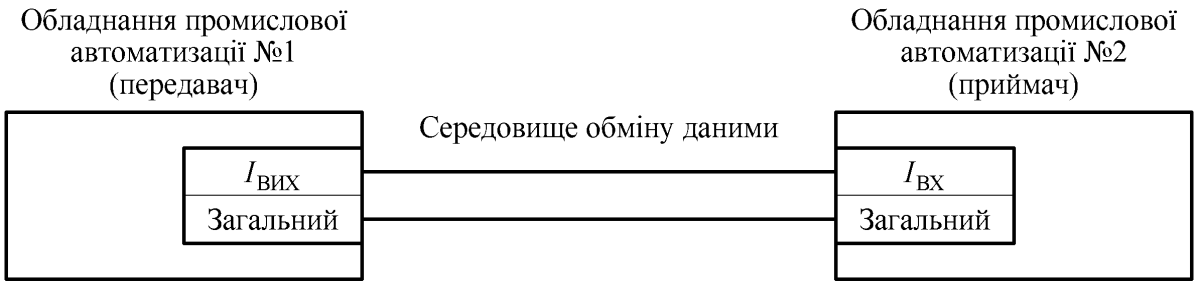


Рис. 3.5. Обмін даними за допомогою уніфікованих електричних неперервних сигналів струму

“Струмова петля” – інша назва обміну даними за допомогою уніфікованих електричних неперервних сигналів струму. Серед таких сигналів найчастіше використовуються сигнали, які мають межі допустимих змін (4 ... 20) мА. Використання саме таких сигналів дає наступні переваги.

1. “Струмова петля” (4 ... 20) мА є простим і надійним інтерфейсом, його вартість при виробництві є невеликою. Такого інтерфейсу достатньо для розв’язання більшості задач промислової автоматизації.

2. Дві жили кабелю, які використовуються “струмовою петлею” для обміну даними між обладнанням промислової автоматизації, можуть також використовуватися для живлення такого обладнання.

3. “Струмова петля” (4 ... 20) мА має так званий “живий нуль” – в такому випадку будь-якому значенню виробничого (технологічного) параметра буде відповідати ненульове значення струму в “струмовій петлі”.

4. “Струмова петля” (4 ... 20) мА може одночасно з передачею аналогових сигналів передавати дискретні сигнали (згідно з протоколом HART), які під час передачі будуть накладатись на такий аналоговий сигнал.

5. “Струмова петля” (4 ... 20) мА має високий рівень іскробезпеки, внаслідок чого може використовуватися у пожежо- і вибухонебезпечних місцях. Крім цього, пошук несправностей в “струмовій петлі” може здійснюватися без її відключення (тобто без розриву ланцюга з провідників, які утворюють такий інтерфейс).

6. “Струмова петля” (4 ... 20) мА може, за необхідності, формувати струми менше 4 мА або більше 20 мА – такі значення струмів можуть бути використані, наприклад, для повідомлення про відмови і/або помилки обладнання промислової автоматизації.

7. “Струмова петля” (4 ... 20) мА накладає менше обмежень на максимальну довжину і мінімальний переріз провідників, які утворюють такий інтерфейс (зокрема, її довжина може становити більше 1 км).

8. “Струмова петля” (4 ... 20) мА щодо завад є відносно малочутливою (шуми і наведення слабо впливають на її роботу).

9. “Струмова петля” (4 ... 20) мА може використовувати практично будь-який кабель на основі мідних/алюмінієвих провідників (звичайно це кабель на основі паралельних провідників).

10. “Струмова петля” (4 ... 20) мА дозволяє розривати ланцюг з провідників, які утворюють такий інтерфейс, на кілька різних частин, при цьому з’єднуючи такі частини за допомогою клем і/або рознімачів чи послідовно охоплюючи такими частинами обладнання промислової автоматизації.

11. “Струмова петля” (4 ... 20) мА для людини є повністю безпечною – при торканні людиною провідників, які утворюють такий інтерфейс, ураження її електричним струмом виключено.

12. “Струмова петля” (4 ... 20) мА для свого обслуговування (наприклад, для пошуку несправностей) не вимагає обслуговуючого персоналу з високою кваліфікацією.

3.3. Інтерфейси RS-232 і RS-485

3.3.1. Інтерфейс RS-232

Стандарт RS-232 – стандарт, розроблений в 1969 р. і призначений для забезпечення асинхронного і/або синхронного послідовного обміну даними між двома пристроями. Відповідно, в ньому описаний і безпосередньо інтерфейс RS-232 (фізичний рівень стандарту RS-232). Основне призначення інтерфейсу RS-232 в промисловості – забезпечення обміну даними між двома незалежними одиницями обладнання промислової автоматизації (тобто реалізація з’єднання у вигляді “точка-точка”).

DTE (абр. від англ. Data terminal equipment, Кінцеве обладнання, яке може здійснювати обмін даними) – наприклад, це може бути програмований контролер або лабораторний блок живлення.

DCE (абр. від англ. Data circuit-terminating equipment (Data communication equipment, Data carrier equipment)), Обладнання, яке може здійснювати пряме і обернене перетворення даних, що формуються кінцевим обладнанням) – наприклад, це може бути перетворювач інтерфейсів або GSM/GPRS-модем.

Загальний принцип обміну даними по інтерфейсу RS-232 наведено на рис. 3.6.

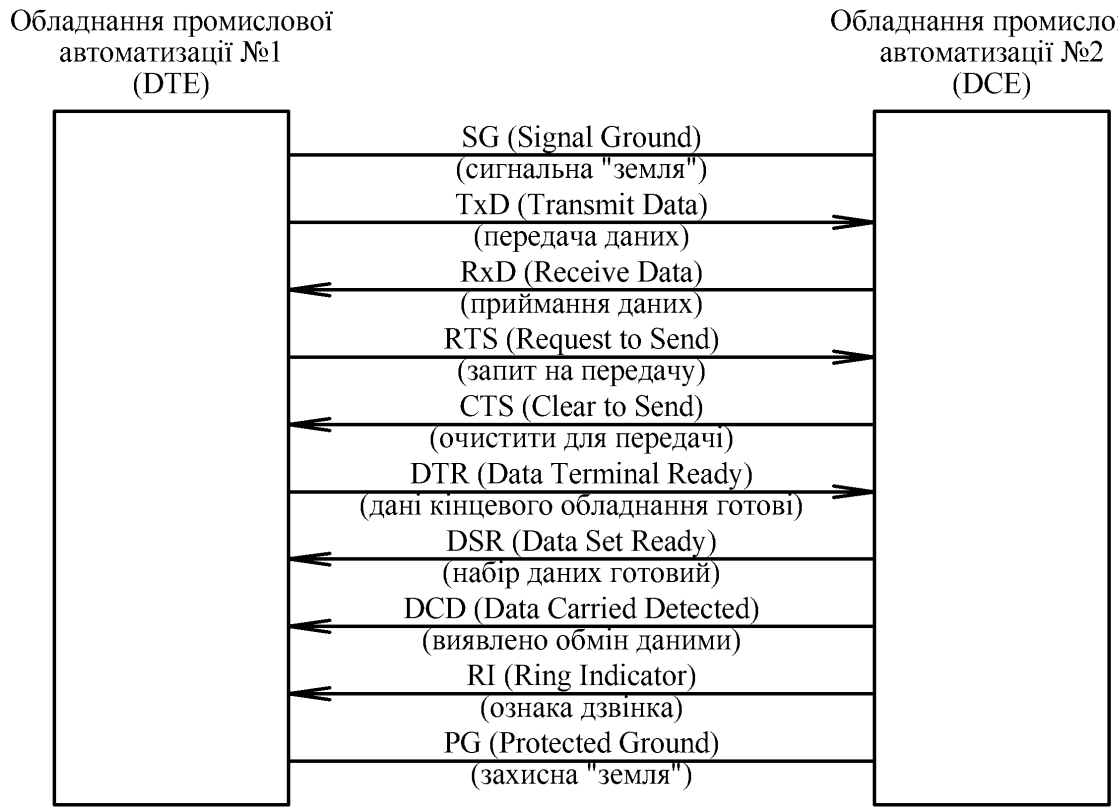


Рис. 3.6. Загальний принцип обміну даними по інтерфейсу RS-232

Інтерфейс RS-232 має наступні типові механічні характеристики:

- на стороні обладнання промислової автоматизації типу DTE звичайно використовуються рознімачі типу DB9-M або DB25-M;

- на стороні обладнання промислової автоматизації типу DCE звичайно використовуються рознімачі типу DB9-F або DB25-F;

- максимальна допустима довжина кабелю – приблизно 15 м.

Інтерфейс RS-232 має наступні типові електричні характеристики:

- при передачі даних лог. 1 відповідає діапазон напруг відносно сигнальної “землі” від -5 до -25 В, лог. 0 – діапазон напруг відносно сигнальної “землі” від 5 до 25 В, діапазон напруг відносно сигнальної “землі” від -5 до 5 В є невизначеним;

- при прийманні даних лог. 1 відповідає діапазон напруг відносно сигнальної “землі” від -3 до -25 В, лог. 0 – діапазон напруг відносно сигнальної “землі” від 3 до 25 В, діапазон напруг відносно сигнальної “землі” від -3 до 3 В є невизначеним.

Інтерфейс RS-232 має наступні типові часові характеристики:

- швидкість обміну даними – до 115200 біт/с.

3.3.2. Інтерфейс RS-485

Стандарт RS-485 – стандарт, розроблений в 1983 р. і призначений для забезпечення послідовного обміну даними між кількома пристроями. Відповідно, в ньому описаний і безпосередньо інтерфейс RS-485 (фізичний рівень стандарту RS-485). Основне призначення інтерфейсу RS-485 в промисловості – забезпечення обміну даними між кількома незалежними одиницями обладнання промислової автоматизації (тобто реалізація з'єднання у вигляді "мережа").

Загальний принцип обміну даними по інтерфейсу RS-485 наведено на рис. 3.7.

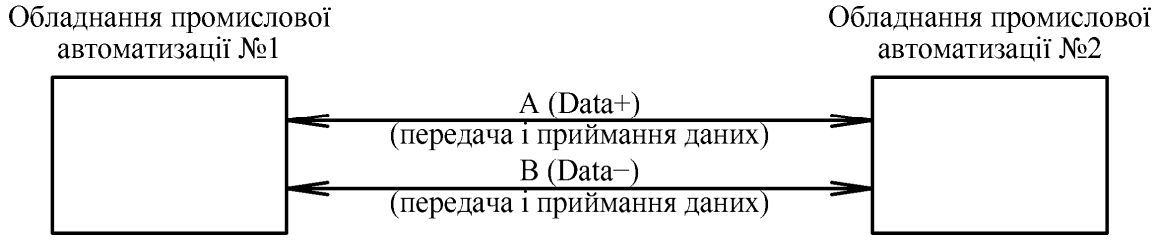


Рис. 3.7. Загальний принцип обміну даними по інтерфейсу RS-485

Інтерфейс RS-485 має наступні типові механічні характеристики:

- рознімачі не визначені:

- максимальна допустима довжина кабелю – приблизно 1200 м.

Інтерфейс RS-485 має наступні типові електричні характеристики:

- при передачі даних лог. 1 відповідає різниця напруг між A (Data+) і B (Data–) від $-1{,}5$ до -6 В (струм протікає від B (Data–) до A (Data+)), лог. 0 – різниця напруг між A (Data+) і B (Data–) від 1,5 до 6 В (струм протікає від A (Data+) до B (Data–)), різниця напруг між A (Data+) і B (Data–) від $-1{,}5$ до 1,5 В є невизначеною;

- при прийманні даних лог. 1 відповідає різниця напруг між A (Data+) і B (Data–) менше $-0{,}2$ В, лог. 0 – різниця напруг між A (Data+) і B (Data–) більше 0,2 В, діапазон напруг від $-0{,}2$ В до 0,2 В є невизначеним;

- напруги A (Data+) і B (Data–) як на виходах передавачів, так і на входах приймачів повинні знаходитися в діапазоні від -7 до 12 В відносно сигнальної "землі";

- максимальна допустима кількість передавачів – 32;

- мінімальний опір навантаження передавача – 54 Ом;

- максимальна допустима кількість приймачів – 32;

- мінімальний вхідний опір приймача – 12 кОм.

Інтерфейс RS-485 має наступні типові часові характеристики:

- швидкість обміну даними – до 10 Мбіт/с.

3.3.3. Кадр інтерфейсів RS-232 і RS-485

Обмін даними за допомогою інтерфейсів RS-232 і RS-485 здійснюється послідовно у вигляді кадрів, формат яких наведено на рис. 3.8.

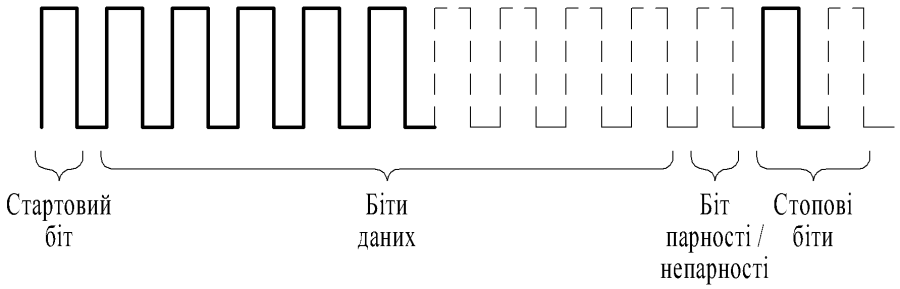


Рис. 3.8. Формат кадрів інтерфейсів RS-232 і RS-485

Послідовність бітів в кожному кадрі наступна.

1. Обов’язковий стартовий біт (завжди утримує лог. 0).

2. Обов’язкові біти даних в кількості від 5 до 9.

3. Необов’язковий біт парності / непарності.

4. Обов’язкові стопові біти в кількості 1, 1,5 (це один стоповий біт, який має збільшену тривалість) або 2 (завжди утримують лог. 1).

Спрощено біти кадрів розпізнаються так, як наведено на рис. 3.9.

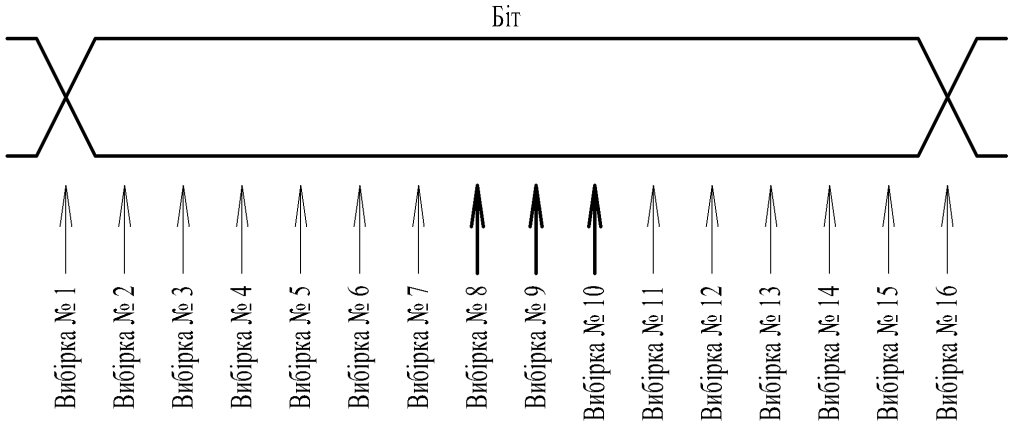


Рис. 3.9. Розпізнавання бітів кадру інтерфейсів RS-232 і RS-485

Під час розпізнавання одного біта виконується задана послідовність вибірок (наприклад, 16). Далі з неї виділяється кілька вибірок (наприклад, 3),

які знаходяться приблизно посередині такої послідовності (наприклад, вибірки з порядковими номерами 8, 9 і 10).

Якщо всі вони відповідають лог. 0, то результатом розпізнавання буде те, що біт відповідає лог. 0. Якщо всі вони відповідають лог. 1, то результатом розпізнавання буде те, що біт відповідає лог. 1. Якщо ж кілька вибірок відповідає лог. 0, а кілька – лог. 1, то результат розпізнавання буде залежати від обраного алгоритму розпізнавання (наприклад, можна буде вважати, що такий біт є помилковим).

3.3.4. Деякі особливості обміну даними по інтерфейсам RS-232 і RS-485

При реалізації людино-машинного інтерфейсу SCADA-системи можуть здійснювати обмін даними з обладнанням промислової автоматизації за допомогою інтерфейсів RS-232 і/або RS-485, при якому необхідно мати відповіді на більшість з наведених нижче питань.

1. Інтерфейс доступний або недоступний (наприклад, доступний)?
2. Який використовується протокол (наприклад, Modbus RTU)?
3. Який використовується апаратний порт (наприклад, COM1)?
4. Яка використовується швидкість обміну даними (наприклад, 9600 біт/с)?
5. Яка використовується кількість бітів даних в одному кадрі (наприклад, 8)?
6. Як використовується біт парності/непарності (наприклад, не використовується)?
7. Яка використовується кількість стопових бітів (наприклад, 1)?
8. Яка використовується адреса одиниці обладнання промислової автоматизації (наприклад, 16)?
9. Який максимальний час очікування відгуку на запит від одиниці обладнання промислової автоматизації, впродовж якого така одиниця вважається справною (наприклад, 500 мс)?
10. Яка максимальна кількість запитів до одиниці обладнання промислової автоматизації, на які не було отримано відгуків, впродовж яких така одиниця вважається справною (наприклад, 3)?
11. Який час між запитами до одиниці обладнання промислової автоматизації (наприклад, 50 мс)?

12. Яка періодичність запитів до одиниці обладнання промислової автоматизації (наприклад, 10 с)?

13. Яка послідовність опитування різних одиниць обладнання промислової автоматизації (наприклад, спочатку одиниця обладнання промислової автоматизації № 3, потім – № 9, потім – № 5 і так далі)?

3.4. Інтерфейс Ethernet

3.4.1. Специфікації інтерфейсу Ethernet

Ethernet (абр. від англ. Ether + network, Ефір + мережа) – технології, які почали з'являтися починаючи приблизно з 1973 р. Всі основні характеристики технологій Ethernet описані переважно в стандартах IEEE (абр. від. англ. Institute of electrical and electronics engineers, Інститут інженерів з електротехніки і електроніки) сімейства 802.3. Відповідно, в них описані і безпосередньо інтерфейси Ethernet (фізичні рівні Ethernet) (наприклад, стандарт 802.3u утримує опис інтерфейсів Ethernet специфікацій 100Base-T, 100Base-T4, 100Base-FX і 100Base-TX). Основне призначення інтерфейсу Ethernet в промисловості – забезпечення обміну даними між кількома (двома і більше) незалежними одиницями обладнання промислової автоматизації (тобто реалізація з'єднання у вигляді "мережа").

Найчастіше в промисловості використовуються інтерфейси Ethernet наступних специфікацій:

- 10BASE-5 – реалізація інтерфейсу Ethernet, яка використовує в якості середовища обміну даними кабелі на основі "товстих" (діаметром приблизно (9 ... 10) мм) коаксіальних провідників (на даний момент є трохи застарілою);

- 10BASE-2 – реалізація інтерфейсу Ethernet, яка використовує в якості середовища обміну даними кабелі на основі "тонких" (діаметром приблизно (5 ... 6) мм) коаксіальних провідників (на даний момент є трохи застарілою);

- 10BASE-F – реалізація інтерфейсу Ethernet, яка використовує в якості середовища обміну даними кабелі на основі двох оптичних волокон;

- 10BASE-T – реалізація інтерфейсу Ethernet, яка використовує в якості середовища обміну даними кабелі на основі двох кручених провідників категорії 3 або вищої;

- 100BASE-FX – реалізація інтерфейсу Ethernet, яка використовує в якості середовища обміну даними кабелі на основі двох оптичних волокон;

- 100BASE-TX – реалізація інтерфейсу Ethernet, яка використовує в якості

середовища обміну даними кабелі на основі двох кручених провідників категорії 5 або вищої.

Загальний принцип обміну даними, наприклад, по інтерфейсу Ethernet специфікації 100BASE-TX наведено на рис. 3.10.

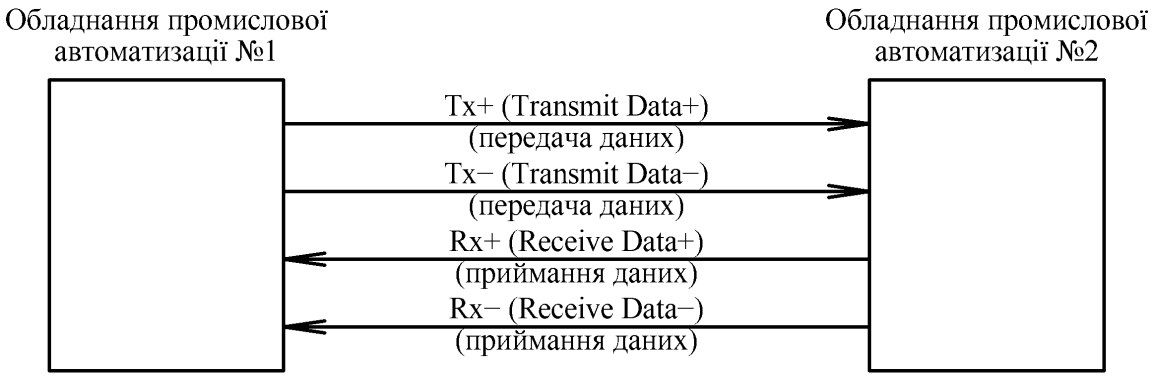


Рис. 3.10. Загальний принцип обміну даними по інтерфейсу Ethernet специфікації 100BASE-TX

При обміні даними в Ethernet використовується метод доступу CSMA/CD (абр. від англ. Carrier sense multiple access with collision detection, Множинний доступ з розпізнаванням несучої і виявленням колізій).

Спрощено алгоритм обміну даними в Ethernet полягає в наступному.

1. Для того, щоб передати в середовище обміну даними кадр, інтерфейс Ethernet обладнання промислової автоматизації повинен “переконатися”, що розділюване (тобто таке, доступ до якого одночасно можуть отримувати кілька незалежних одиниць такого обладнання) середовище обміну даними є вільним. Для цього такий інтерфейс “прослуховує” несучу (основну гармоніку) сигналу, за допомогою якого здійснюється обмін даними.

Якщо розпізнавання несучої “показує” її відсутність, то інтерфейс Ethernet має право почати передачу кадру.

Початок передачі кадру кожного з інтерфейсів Ethernet кожного разу визначається випадково, тому:

- якщо свій кадр одночасно передає тільки один інтерфейс Ethernet, то він буде успішно переданим;

- якщо свої кадри одночасно передають кілька інтерфейсів Ethernet, то виникне колізія, такі кадри змішаються і будуть втрачені, і їх потрібно буде передавати повторно.

Так як передача кадрів від кожного з інтерфейсів Ethernet кожного разу визначається випадково, то завжди буде ситуація, коли до розділюваного

середовища обміну даними буде підключений один інтерфейсі Ethernet і кадр від такого інтерфейсу буде переданий успішно.

2. Усі інтерфейси Ethernet, які приєднані до розділюваного середовища обміну даними, мають можливість негайно (але з урахуванням затримки розповсюдження сигналу в такому середовищі) отримувати будь-які пакети, які передаються в таке середовище.

Пакет, який передається, завжди утримує адресу того інтерфейсу Ethernet (тобто тієї одиниці обладнання промислової автоматизації), до якого він надсилається (якщо такий пакет є широкомовним, то до кількох відповідних інтерфейсів Ethernet обладнання промислової автоматизації). Але при цьому використання комутації, наприклад, дозволяє отримувати кадри саме тим інтерфейсам Ethernet, яким такі кадри були призначені.

3. Після успішної передачі кадру в розділюване середовище обміну даними всі інтерфейси Ethernet повинні витримати паузу, яка дорівнює міжкадровому інтервалу. Така пауза потрібно для того, щоб привести всі інтерфейси Ethernet у вихідний стан і для того, щоб запобігти монопольному захопленню розділюваного середовища обміну даними якимось одним інтерфейсом Ethernet.

4. Після витримки паузи для передачі наступного кадру в розділюване середовище обміну даними необхідно повторити попередні п. 1 – п. 3.

3.4.2. Деякі особливості обміну даними по інтерфейсу Ethernet

При реалізації людино-машинного інтерфейсу SCADA-системи можуть здійснювати обмін даними з обладнанням промислової автоматизації також за допомогою інтерфейсів Ethernet, при якому також необхідно мати відповіді на більшість з наведених нижче питань.

1. Інтерфейс доступний або недоступний (наприклад, доступний)?

2. Який використовується протокол (наприклад, Modbus TCP/IP)?

3. Який використовується апаратний порт (наприклад, Ethernet1)?

4. Яка використовується швидкість обміну даними (наприклад, 100 Мбіт/с)?

5. Яка використовується IP-адреса (наприклад, 192.168.1.50)?

6. Який використовується програмний порт (наприклад, 502).

7. Яка використовується адреса одиниці обладнання промислової автоматизації (наприклад, 24)?

8. Який максимальний час очікування відгуку на запит від одиниці

обладнання промислової автоматизації, впродовж якого така одиниця вважається справною (наприклад, 400 мс)?

9. Яка максимальна кількість запитів до одиниці обладнання промислової автоматизації, на які не було отримано відгуків, впродовж яких така одиниця вважається справною (наприклад, 4)?

10. Який час між запитами до одиниці обладнання промислової автоматизації (наприклад, 25 мс)?

11. Яка періодичність запитів до одиниці обладнання промислової автоматизації (наприклад, 20 с)?

12. Яка послідовність опитування різних одиниць обладнання промислової автоматизації (наприклад, спочатку одиниця обладнання промислової автоматизації № 10, потім – № 1, потім – № 6, потім – № 4 і так далі)?

3.5. Питання для самоперевірки

1. Що таке аналоговий інтерфейс?
2. Що таке цифровий інтерфейс?
3. Що таке з’єднання у вигляді “точка-точка?
4. Що таке з’єднання у вигляді “мережа”?
5. Що таке кадр?
6. Які способи кодування кадрів в промисловій автоматизацій використовуються найчастіше?
7. Що таке уніфіковані сигнали?
8. Які межі допустимих змін мають уніфіковані електричні неперервні сигнали напруги?
9. Які межі допустимих змін мають уніфіковані електричні неперервні сигнали струму?
10. Що таке “струмова петля” (4 ... 20) мА, які її переваги у порівнянні з іншими інтерфейсами?
11. Що таке інтерфейс RS-232, які він має типові механічні, електричні і часові характеристики?
12. Що таке інтерфейс RS-485, які він має типові механічні, електричні і часові характеристики?
13. Який формат мають кадри інтерфейсів RS-232 і RS-485?
14. Які особливості має обмін даними по інтерфейсам RS-232 і RS-485?
15. Що таке інтерфейс Ethernet?

16. Який метод доступу використовується при обміні даними в Ethernet?
17. В чому полягає алгоритм обміну даними в Ethernet?
18. Які особливості має обмін даними по інтерфейсу Ethernet?

3.6. Рекомендована література

Базова література: [1, 4–6].
Додаткова література: [1, 5, 6].

4. ОСНОВНІ ПРОТОКОЛИ, ЯКІ ВИКОРИСТОВУЮТЬСЯ В ПРОМИСЛОВІЙ АВТОМАТИЗАЦІЇ

4.1. Загальні відомості про протоколи, які використовуються в промисловій автоматизації

Серед обладнання промислової автоматизації (в тому числі і вітчизняних розробників) найчастіше використовуються наступні протоколи: сімейство протоколів Modbus (Modbus RTU, Modbus ASCII і Modbus TCP/IP), DCON, ОВЕН і HART. Інші протоколи, такі, наприклад, як Message Queuing Telemetry Transport (MQTT), Simple Network Management Protocol (SNMP) і Secure Shell (SSH), звичайно використовуються не так часто. Основними вимогами до таких протоколів є надійна роботи в складних умовах експлуатації.

Протоколи звичайно відповідають одному (іноді кільком) рівням моделі OSI – найчастіше це прикладний рівень. Відповідно, для більшості з них визначаються такі поняття, як адреси пристроїв, префікси, суфікси, коди функцій, ідентифікатори обміну, ідентифікатори протоколу, контрольні суми і так далі.

4.2. Сімейство протоколів Modbus

4.2.1. Протокол Modbus RTU

Протокол Modbus RTU – протокол для обміну даними, в основі якого лежить архітектура “ведучий-ведений” (“master-slave”). Він входить до складу протоколів сімейства Modbus разом з протоколами Modbus ASCII і Modbus TCP/IP.

Він є одним з найбільш розповсюджених протоколів, які використовуються в промисловості.

Основні інтерфейси, які використовують протокол Modbus RTU: RS-232 і RS-485.

Загальний принцип роботи протоколу Modbus RTU полягає в наступному:

- ведучий (master) формує пакет даних і передає його в промислову мережу веденому (slave);

- ведений (slave) приймає адресований йому пакет даних, обробляє його, формує вже свій пакет даних і передає його у відповідь в промислову мережу ведучому (master);

- ведучий (master) приймає адресований йому пакет даних і обробляє його.

ADU (абр. від англ. Application data unit, Блок даних програми) – це пакет, який становить основу протоколу Modbus RTU (див. рис. 4.1). Максимальний розмір пакета **ADU** – 256 байтів.

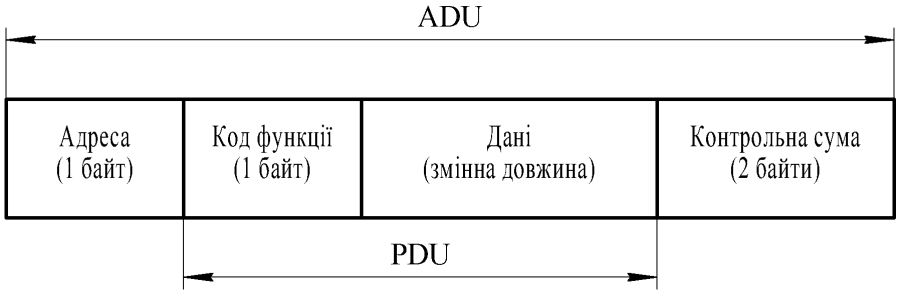


Рис. 4.1. Формат пакета **ADU** протоколу Modbus RTU

Адреса – це поле, яке утримує адресу веденого (slave) (навіть у відповіді на пакет, який надіслав ведучий (master)). Завдяки цьому ведучий (master) знає, якому веденому (slave) було надіслано пакет і від якого веденого (slave) прийшла відповідь. Адреса 0 використовується для широкомовної передачі (вона розпізнається кожним веденим (slave)), адреси з 1 по 247 можуть використовуватися, адреси з 248 по 255 є зарезервованими для майбутнього використання.

Код функції – це поле, яке утримує код тієї дії, яку повинен виконати ведений (slave) (наприклад, код 06_{16} показує, що ведений (slave) повинен змінити уміст одного з своїх регістрів (записати в нього інше значення)).

Дані – це поле, яке утримує дані певної одиниці обладнання промислової автоматизації (такі дані визначаються виробником такого обладнання і загалом можуть бути довільними).

Контрольна сума – це сума, яка обчислюється згідно з алгоритмом CRC16 (абр. від англ. Cyclic redundancy check, Циклічна надлишкова перевірка).

PDU (абр. від англ. Protocol data unit, Блок даних протоколу) – це поля **Код функції** і **Дані** разом.

Протокол Modbus RTU передбачає, що між кадрами пакетів **ADU** повинні бути інтервали часу не більші, ніж 1,5 від тривалості таких кадрів. При цьому між самими пакетами **ADU** повинні бути інтервали часу не менші, ніж 3,5 від тривалості таких пакетів.

4.2.2. Протокол Modbus ASCII

Протокол Modbus ASCII – протокол для обміну даними, в основі якого також лежить архітектура “ведучий-ведений” (“master-slave”). Входить до складу протоколів сімейства Modbus разом з протоколами Modbus RTU і Modbus TCP/IP.

Він є одним з достатньо розповсюджених протоколів, які використовуються в промисловості.

Основні інтерфейси, які використовують протокол Modbus ASCII: RS-232 і RS-485.

Загальний принцип роботи протоколу Modbus ASCII співпадає із загальним принципом роботи протоколу Modbus RTU.

ADU – це пакет, який становить основу протоколу Modbus ASCII (див. рис. 4.2). Максимальний розмір пакета **ADU** – 513 байтів.

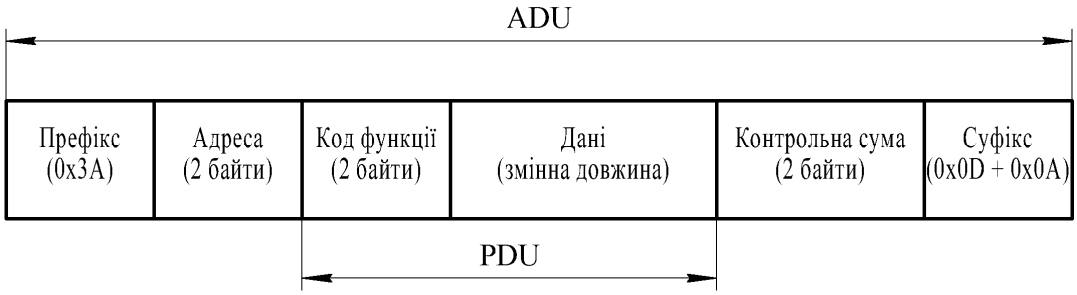


Рис. 4.2. Формат пакета **ADU** протоколу Modbus ASCII

Префікс – це поле, яке показує початок пакета **ADU**.

Адреса – це поле, яке утримує адресу веденого (slave) (кодується по принципу “ASCII-символ – дві тетради”).

Код функції –це поле, яке утримує код тієї дії, яку повинен виконати ведений (slave) (кодується по принципу “ASCII-символ – дві тетради”).

Дані – це поле, яке утримує дані певної одиниці обладнання промислової автоматизації (кодується по принципу “ASCII-символ – дві тетради”).

Контрольна сума – це сума, яка обчислюється згідно з алгоритмом LRC8 (абр. від англ. Longitudinal Redundancy Check, Поздовжня надлишкова перевірка) (кодується по принципу “ASCII-символ – дві тетради”).

Суфікс – це поле, яке показує кінець пакета **ADU**.

PDU – це поля **Код функції** і **Дані** разом.

Протокол Modbus ASCII передбачає, що між пакетами **ADU** інтервали часу можуть бути відсутніми.

4.2.3. Протокол Modbus TCP/IP

Протокол Modbus TCP/IP – протокол для обміну даними, в основі якого також лежить архітектура “ведучий-ведений” (“master-slave”). Входить до складу протоколів сімейства Modbus разом з протоколами Modbus RTU і Modbus ASCII.

Він є одним з найбільш розповсюджених протоколів, які використовуються в промисловості.

Основні інтерфейси, які використовують протокол Modbus TCP/IP: Ethermet.

Загальний принцип роботи протоколу Modbus TCP/IP співпадає із загальним принципом роботи протоколу Modbus RTU.

ADU – це пакет, який становить основу протоколу Modbus TCP/IP (див. рис. 4.3). Максимальний розмір пакета **ADU** – 260 байтів.

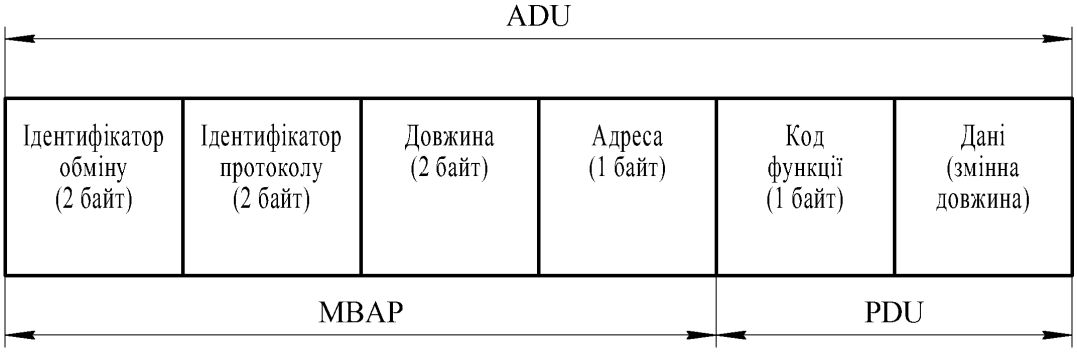


Рис. 4.3. Формат пакета **ADU** протоколу Modbus TCP/IP

Ідентифікатор обміну – це поле, яке звичайно утримує порядковий номер пакета **ADU**.

Ідентифікатор протоколу – це поле, яке утримує ідентифікатор протоколу Modbus TCP/IP (який дорівнює 0).

Довжина – це поле, яке утримує кількість байтів в полях **Адреса**, **Код функції** і **Дані** (тобто від поля **Довжина** і до кінця пакета **ADU**).

Адреса – це поле, яке утримує адресу веденого (slave).

Код функції – це поле, яке утримує код тієї дії, яку повинен виконати ведений (slave).

Дані – це поле, яке утримує дані певної одиниці обладнання промислової автоматизації.

PDU – це поля **Код функції** і **Дані** разом.

4.2.4. Основні функції обробки даних сімейства протоколів Modbus

4.2.4.1. Функція 01_{16} (читання стану кількох “котушок”)

Функція 01_{16} (читання стану кількох “котушок”– англ. Read Coils) читає стан кількох “котушок” одиниці обладнання промислової автоматизації.

Для виклику такої функції необхідно послідовно вказати:

- адресу одиниці обладнання промислової автоматизації (в даному випадку 10_{16});
- код функції (в даному випадку 01_{16});
- адресу “котушки”, з якої повинно початися читання стану “котушок” (в даному випадку 0050_{16});
- кількість “котушок” (в даному випадку 0012_{16}).

“Котушки”адресуються, починаючи з 0. Їх стан читається у вигляді “одна “котушка” – один біт”.

Якщо кількість “котушок”, стан яких читається, не кратна 8, то біти, які осталися в останньому байті, будуть встановлені в лог. 0. Приклад використання функції 01_{16} наведено нижче.

При передачі пакета по протоколу Modbus RTU:

Поле пакета ADU (без контрольної суми)	Значення
Адреса одиниці обладн. промис. автоматиз.	10_{16}
Код функції	01_{16}
Адреса “котушки” (Hi) Адреса “котушки” (Lo)	00_{16} 50_{16}
Кількість “котушок” (Hi) Кількість “котушок” (Lo)	00_{16} 12_{16}

При прийманні відгуку на такий пакет (якщо помилки відсутні):

Поле пакета ADU (без контрольної суми)	Значення
Адреса одиниці обладн. промис. автоматиз.	10_{16}
Код функції	01_{16}
Кількість байтів	03_{16}
Стан котушок	$9D_{16}$
Стан котушок	$5E_{16}$
Стан котушок	02_{16}

4.2.4.2. Функція 02_{16} (читання стану кількох дискретних входів)

Функція 02_{16} (читання стану кількох дискретних входів – англ. Read Discrete Inputs) читає стан кількох дискретних входів одиниці обладнання промислової автоматизації.

Для виклику такої функції необхідно послідовно вказати:

- адресу одиниці обладнання промислової автоматизації (в даному випадку 10_{16});
- код функції (в даному випадку 02_{16});
- адресу дискретного входу, з якої повинно початися читання стану дискретних входів (в даному випадку $00A0_{16}$);
- кількість дискретних входів (в даному випадку 0016_{16}).

Дискретні входи адресуються, починаючи з 0. Їх стан читається у вигляді “один дискретний вхід – один біт”.

Якщо кількість дискретних входів, стан яких читається, не кратна 8, то біти, які остались в останньому байті, будуть встановлені в лог. 0. Приклад використання функції 02_{16} наведено нижче.

При передачі пакета по протоколу Modbus RTU:

Поле пакета ADU (без контрольної суми)	**Значення**
Адреса одиниці обладн. промис. автоматиз.	10_{16}
Код функції	02_{16}
Адреса дискретного входу (Hi) Адреса дискретного входу (Lo)	00_{16} $A0_{16}$
Кількість дискретних входів (Hi) Кількість дискретних входів (Lo)	00_{16} 16_{16}

При прийманні відгуку на такий пакет (якщо помилки відсутні):

Поле пакета ADU (без контрольної суми)	**Значення**
Адреса одиниці обладн. промис. автоматиз.	10_{16}
Код функції	02_{16}
Кількість байтів	03_{16}
Стан дискретних входів	78_{16}
Стан дискретних входів	$E6_{16}$
Стан дискретних входів	$2C_{16}$

4.2.4.3. Функція 03_{16} (читання стану кількох регістрів зберігання)

Функція 03_{16} (читання стану кількох регістрів зберігання – англ. Read Holding Registers) читає стан кількох регістрів зберігання одиниці обладнання промислової автоматизації.

Для виклику такої функції необхідно послідовно вказати:

- адресу одиниці обладнання промислової автоматизації (в даному випадку 10_{16});
- код функції (в даному випадку 03_{16});
- адресу регістра зберігання, з якої повинно початися читання стану регістрів зберігання (в даному випадку $00F0_{16}$);
- кількість регістрів зберігання (в даному випадку 0003_{16}).

Регістри зберігання адресуються, починаючи з 0. Їх стан читається у вигляді “один регістр зберігання – два байта”. Приклад використання функції 03_{16} наведено нижче.

При передачі пакета по протоколу Modbus RTU:

Поле пакета ADU (без контрольної суми)	**Значення**
Адреса одиниці обладн. промис. автоматиз.	10_{16}
Код функції	03_{16}
Адреса регістра зберігання (Hi) Адреса регістра зберігання (Lo)	00_{16} $F0_{16}$
Кількість регістрів зберігання (Hi) Кількість регістрів зберігання (Lo)	00_{16} 03_{16}

При прийманні відгуку на такий пакет (якщо помилки відсутні):

Поле пакета ADU (без контрольної суми)	**Значення**
Адреса одиниці обладн. промис. автоматиз.	10_{16}
Код функції	03_{16}
Кількість байтів	06_{16}
Стан регістра зберігання (Hi) Стан регістра зберігання (Lo)	81_{16} BF_{16}
Стан регістра зберігання (Hi) Стан регістра зберігання (Lo)	$0D_{16}$ 39_{16}
Стан регістра зберігання (Hi) Стан регістра зберігання (Lo)	93_{16} 05_{16}

4.2.4.4. Функція 04_{16} (читання стану кількох вхідних регістрів)

Функція 04_{16} (читання стану кількох вхідних регістрів – англ. Read Input Registers) читає стан кількох вхідних регістрів одиниці обладнання промислової автоматизації.

Для виклику такої функції необхідно послідовно вказати:

- адресу одиниці обладнання промислової автоматизації (в даному випадку 10_{16});

- код функції (в даному випадку 04_{16});

- адресу вхідного регістра, з якої повинно початися читання стану вхідних регістрів (в даному випадку 0140_{16});

- кількість вхідних регістрів (в даному випадку 0003_{16}).

Вхідні регістри адресуються, починаючи з 0. Їх стан читається у вигляді "один вхідний регістр – два байта". Приклад використання функції 04_{16} наведено нижче.

При передачі пакета по протоколу Modbus RTU:

Поле пакета ADU (без контрольної суми)	**Значення**
Адреса одиниці обладн. промис. автоматиз.	10_{16}
Код функції	04_{16}
Адреса вхідного регістра (Hi) Адреса вхідного регістра (Lo)	01_{16} 40_{16}
Кількість вхідних регістрів (Hi) Кількість вхідних регістрів (Lo)	00_{16} 03_{16}

При прийманні відгуку на такий пакет (якщо помилки відсутні):

Поле пакета ADU (без контрольної суми)	**Значення**
Адреса одиниці обладн. промис. автоматиз.	10_{16}
Код функції	04_{16}
Кількість байтів	06_{16}
Стан вхідного регістра (Hi) Стан регістра зберігання (Lo)	$8D_{16}$ $1B_{16}$
Стан вхідного регістра (Hi) Стан регістра зберігання (Lo)	$3A_{16}$ $B8_{16}$
Стан вхідного регістра (Hi) Стан регістра зберігання (Lo)	99_{16} 24_{16}

4.2.4.5. Функція 05_{16} (запис стану однієї “котушки”)

Функція 05_{16} (запис стану однієї “котушки” – англ. Write Single Coil) записує стан однієї “котушки” одиниці обладнання промислової автоматизації.

Для виклику такої функції необхідно послідовно вказати:

- адресу одиниці обладнання промислової автоматизації (в даному випадку 10_{16});

- код функції (в даному випадку 05_{16});

- адресу “котушки”, в яку повинен початися запис стану “котушки” (в даному випадку 0190_{16});

- стан “котушки” (значення 0000_{16} або $FF00_{16}$ (всі інші значення є невірними)) (в даному випадку $FF00_{16}$).

“Котушки” адресуються, починаючи з 0. Їх стан записується у вигляді “одна “котушка” – два байта” (лог. 0 відповідає значення 0000_{16}, лог. 1 – значення $FF00_{16}$). В деяких випадках запис стану “котушки” є неможливим через її апаратну і/або програмну недоступність. Приклад використання функції 05_{16} наведено нижче.

При передачі пакета по протоколу Modbus RTU:

Поле пакета ADU (без контрольної суми)	**Значення**
Адреса одиниці обладн. промис. автоматиз.	10_{16}
Код функції	05_{16}
Адреса “котушки” (Hi) Адреса “котушки” (Lo)	01_{16} 90_{16}
Стан “котушки” (Hi) Стан “котушки” (Lo)	FF_{16} 00_{16}

При прийманні відгуку на такий пакет (якщо помилки відсутні):

Поле пакета ADU (без контрольної суми)	**Значення**
Адреса одиниці обладн. промис. автоматиз.	10_{16}
Код функції	05_{16}
Адреса “котушки” (Hi) Адреса “котушки” (Lo)	01_{16} 90_{16}
Стан “котушки” (Hi) Стан “котушки” (Lo)	FF_{16} 00_{16}

4.2.4.6. Функція 06_{16} (запис стану одного регістра)

Функція 06_{16} (запис стану одного регістра – англ. Write Single Register) записує стан одного регістра одиниці обладнання промислової автоматизації.

Для виклику такої функції необхідно послідовно вказати:

- адресу одиниці обладнання промислової автоматизації (в даному випадку 10_{16});
- код функції (в даному випадку 06_{16});
- адресу регістра, в яку повинен початися запис стану регістра (в даному випадку $01E0_{16}$);
- стан регістра (значення від 0000_{16} до $FFFF_{16}$) (в даному випадку $E94C_{16}$).

Регістри адресуються, починаючи з 0. Їх стан записується у вигляді “один регістр – два байта”. В деяких випадках запис стану регістра є неможливим через його апаратну і/або програмну недоступність. Приклад використання функції 06_{16} наведено нижче.

При передачі пакета по протоколу Modbus RTU:

Поле пакета ADU (без контрольної суми)	**Значення**
Адреса одиниці обладн. промис. автоматиз.	10_{16}
Код функції	06_{16}
Адреса регістра (Hi) Адреса регістра (Lo)	01_{16} $E0_{16}$
Стан регістра (Hi) Стан регістра (Lo)	$E9_{16}$ $4C_{16}$

При прийманні відгуку на такий пакет (якщо помилки відсутні):

Поле пакета ADU (без контрольної суми)	**Значення**
Адреса одиниці обладн. промис. автоматиз.	10_{16}
Код функції	06_{16}
Адреса регістра (Hi) Адреса регістра (Lo)	01_{16} $E0_{16}$
Стан регістра (Hi) Стан регістра (Lo)	$E9_{16}$ $4C_{16}$

4.2.4.7. Функція $0F_{16}$ (запис стану кількох “котушок”)

Функція $0F_{16}$ (запис стану кількох “котушок” – англ. Write Multiple Coils) записує стан кількох “котушок” одиниці обладнання промислової автоматизації.

Для виклику такої функції необхідно послідовно вказати:

- адресу одиниці обладнання промислової автоматизації (в даному випадку 10_{16});
- код функції (в даному випадку $0F_{16}$);
- адресу “котушки”, з якої повинен початися запис стану “котушок” (в даному випадку 0230_{16});
- кількість “котушок” (в даному випадку $000C_{16}$);
- кількість байтів, які будуть утримувати стан “котушок” (в даному випадку 02_{16});
- байти, які будуть утримувати стан “котушок” (в даному випадку 84_{16} і $0E_{16}$).

“Котушки” адресуються, починаючи з 0. Їх стан записується у вигляді “одна “котушка” – один біт”. В деяких випадках запис стану “котушок” є неможливим через її апаратну і/або програмну недоступність.

Якщо кількість “котушок, стан яких записується, не кратна 8, то біти, які осталися в останньому байті, повинні бути встановлені в лог. 0. Приклад використання функції $0F_{16}$ наведено нижче.

При передачі пакета по протоколу Modbus RTU:

Поле пакета ADU (без контрольної суми)	**Значення**
Адреса одиниці обладн. промис. автоматиз.	10_{16}
Код функції	$0F_{16}$
Адреса “котушки” (Hi) Адреса “котушки” (Lo)	02_{16} 30_{16}
Кількість “котушок” (Hi) Кількість “котушок” (Lo)	00_{16} $0C_{16}$
Кількість байтів	02_{16}
Стан “котушки”	84_{16}
Стан “котушки”	$0E_{16}$

При прийманні відгуку на такий пакет (якщо помилки відсутні):

Поле пакета ADU (без контрольної суми)	Значення
Адреса одиниці обладн. промис. автоматиз.	10_{16}
Код функції	$0F_{16}$
Адреса “котушки” (Hi) Адреса “котушки” (Lo)	02_{16} 30_{16}
Кількість “котушок” (Hi) Кількість “котушок” (Lo)	00_{16} $0C_{16}$

4.2.4.8. Функція 10_{16} (запис стану кількох регістрів)

Функція 10_{16} (запис стану кількох регістрів – англ. Write Multiple Registers) записує стан кількох регістрів одиниці обладнання промислової автоматизації.

Для виклику такої функції необхідно послідовно вказати:

- адресу одиниці обладнання промислової автоматизації (в даному випадку 10_{16});

- код функції (в даному випадку 10_{16});

- адресу регістра, з якої повинен початися запис стану регістрів (в даному випадку 0280_{16});

- кількість регістрів (в даному випадку 0002_{16});

- кількість байтів, які будуть утримувати стан регістрів (в даному випадку 04_{16});

- байти, які будуть утримувати стан регістрів (в даному випадку $3D_{16}$, 90_{16}, AC_{16} і 13_{16}).

Регістри адресуються, починаючи з 0. Їх стан зарисується у вигляді “один регістр – два байта”. Приклад використання функції 10_{16} наведено нижче.

При передачі пакета по протоколу Modbus RTU:

Поле пакета ADU (без контрольної суми)	Значення
Адреса одиниці обладн. промис. автоматиз.	10_{16}
Код функції	10_{16}
Адреса регістра (Hi) Адреса регістра (Lo)	02_{16} 80_{16}
Кількість регістрів (Hi) Кількість регістрів (Lo)	00_{16} 02_{16}
Кількість байтів	04_{16}
Стан регістра (Hi) Стан регістра (Lo)	$3D_{16}$ 90_{16}

Поле пакета ADU (без контрольної суми)	Значення
Стан регістра (Hi)	AC_{16}
Стан регістра (Lo)	13_{16}

При прийманні відгуку на такий пакет (якщо помилки відсутні):

Поле пакета ADU (без контрольної суми)	Значення
Адреса одиниці обладн. промис. автоматиз.	10_{16}
Код функції	10_{16}
Адреса регістра (Hi)	02_{16}
Адреса регістра (Lo)	80_{16}
Кількість регістрів (Hi)	00_{16}
Кількість регістрів (Lo)	02_{16}

4.2.5. Обробка помилок, які виникли при обробці даних в сімействі протоколів Modbus

При обробці помилок, які виникли при обробці даних в сімействі протоколів Modbus, поля **Код функції** і **Дані** (поле **PDU**) пакета **ADU**, який передається ведучому (master) від веденого (slave), утримують дані про поточну помилку.

Поле **Код функції** утримує код функції, який було передано від ведучого (master) до веденого (slave), з додаванням до нього 80_{16}.

Поле **Дані** утримує код помилки, який може мати значення 01_{16}, 02_{16}, 03_{16} або 04_{16}.

Спрощено алгоритм формування кодів помилок полягає в наступному.

1. Чи підтримує ведений (slave) код функції, переданий ведучим (master) в пакеті **ADU**? Якщо ні, то формується код помилки 01_{16}.

2. Чи підтримує ведений (slave) кількість “котушок” (дискретних входів, регістрів зберігання і так далі), передану ведучим (master) в пакеті **ADU**? Якщо ні, то формується код помилки 02_{16}.

3. Чи підтримує ведений (slave) діапазон адрес “котушок” (дискретних входів, регістрів зберігання і так далі), переданий ведучим (master) в пакеті **ADU**? Якщо ні, то формується код помилки 03_{16}.

4. Чи була успішно виконана веденим (slave) операція читання стану кількох “котушок” (читання стану кількох дискретних входів і так далі)? Якщо ні, то формується код помилки 04_{16}.

4.3. Протокол DCON

Протокол DCON – протокол для обміну даними, в основі якого також лежить архітектура “ведучий-ведений” (“master-slave”).

Він є одним з достатньо розповсюджених нескладних протоколів, які використовуються в промисловості (має кілька різних реалізацій).

Основні інтерфейси, які використовують протокол DCON: RS-232 і RS-485.

Загальний принцип роботи протоколу DCON співпадає із загальним принципом роботи протоколу Modbus RTU.

Формат пакета протоколу DCON (в загальному вигляді) наведено на рис. 4.4. Такий пакет повністю складається з ASCII-символів.

Роздільник (1 байт)	[Адреса] (1 байт)	[Додаткова команда] (від 1 до 5 байт)	[Блок даних] (від 1 до 256 байт)	[Контрольна сума] (1 байт)	Прапор “CR” (1 байт)

Рис. 4.4. Формат пакета протоколу DCON (в загальному вигляді)

Роздільник – це поле, яке утримує ASCII-символи “#”, “@”, “$” і деякі інші (при передачі пакета від ведучого (master) до веденого (slave)) та “>”, “!”, “?” і деякі інші (при передачі пакета від веденого (slave) до ведучого (master)).

Адреса – це поле, яке утримує адресу веденого (slave). Адреса 0 використовується для широкомовної передачі (вона розпізнається кожним веденим (slave)), адреси з 1 по 255 можуть використовуватися.

Додаткова команда – це поле, яке утримує уточнення до тих дій, які визначаються роздільником.

Блок даних – це поле, яке утримує дані певної одиниці обладнання промислової автоматизації.

Контрольна сума – це сума, яка обчислюється як сума кодів всіх ASCII-символів (при їх наявності) від початку пакета до поля **Блок даних** (якщо така сума перевищує FF_{16}, то використовується тільки її молодший байт).

Прапор “CR” – це ASCII-символ, який має код $0D_{16}$ і зустрічається в пакеті тільки один раз в його кінці.

Якщо при передачі пакета від ведучого (master) до веденого (slave) виникла помилка, то відповідь від веденого (slave) до ведучого (master) може взагалі не передаватися.

4.4. Протокол ОВЕН

Протокол ОВЕН – протокол для обміну даними, в основі якого також лежить архітектура “ведучий-ведений” (“master-slave”).

Він є одним з достатньо розповсюджених протоколів, які використовуються в промисловості.

Основні інтерфейси, які використовують протокол ОВЕН: RS-232 і RS-485.

Загальний принцип роботи протоколу ОВЕН співпадає із загальним принципом роботи протоколу Modbus RTU.

Маркера початку пакета “#” – це ASCII-символ, який має код 23_{16} і зустрічається в пакеті тільки один раз на його початку (див. рис. 4.5).

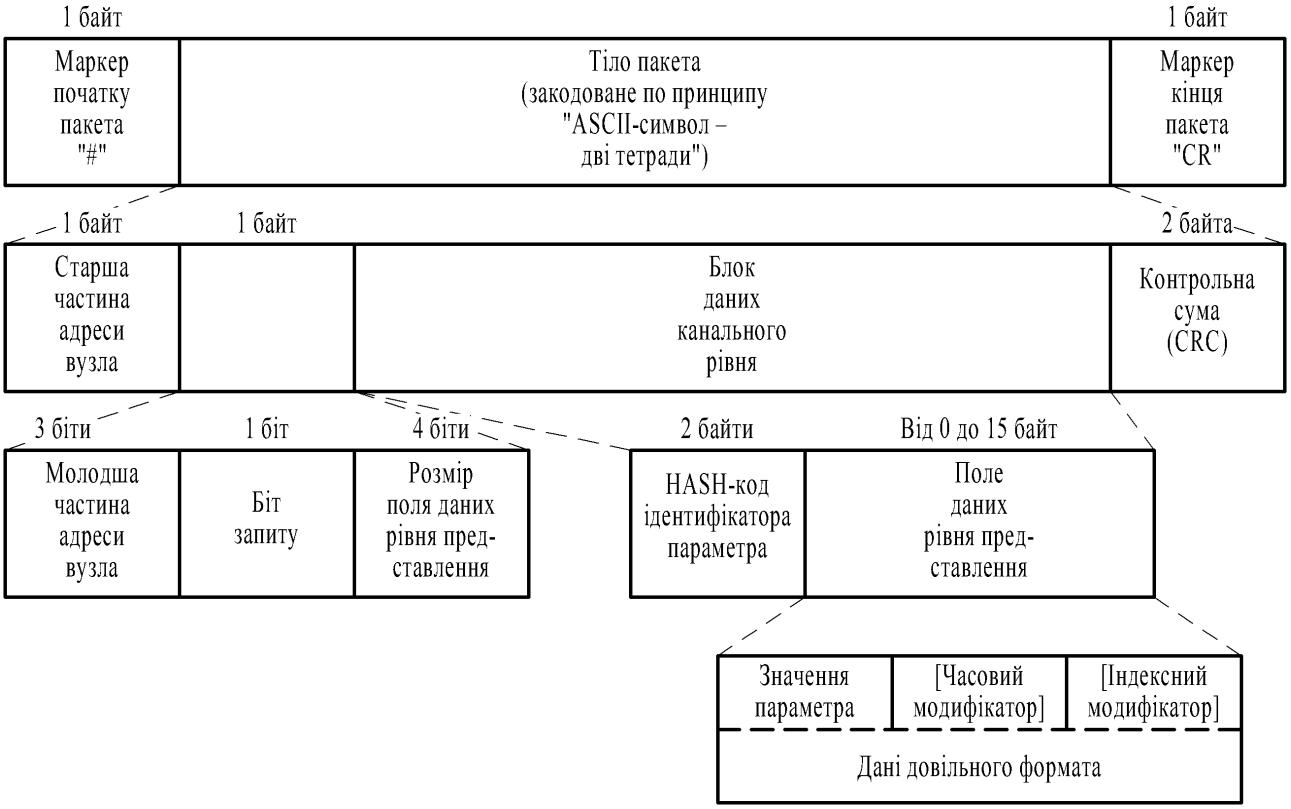


Рис. 4.5. Формат пакета протоколу ОВЕН

Маркера кінця пакета “CR” – це ASCII-символ, який має код $0D_{16}$ і зустрічається в пакеті тільки один раз в його кінці. Все тіло пакета (за виключенням ASCII-символів “#” і “CR”) кодується по принципу “ASCII-символ – дві тетради”. Так як тетрада складається з 4 бітів, то 1 байт тіла пакета можна передати за допомогою 2-ох тетрад. Всі тетради від 0000_2 до 1111_2 перетворюються відповідно в ASCII-символи від “G” до “V” (які мають коди

від 47_{16} до 56_{16}). Припустимо, що в тілі пакета необхідно передати байт $9B_{16}$ (10011011_2). Він складається з 2-ох тетрад: 1001_2 і 1011_2. Тетраді 1001_2 відповідає ASCII-символ “P” (50_{16}), а тетраді 1011_2 – ASCII-символ “R” (52_{16}). Таким чином, замість байта $9B_{16}$ необхідно передати два байта 50_{16} і 52_{16}, з яких після їх приймання відновити вихідний байт $9B_{16}$. Так як в тілі пакета можна передавати тільки символи від “G” до “V” (тобто байти від 47_{16} до 56_{16}), то будь-які інші значення байтів (менші від 47_{16} або більші від 56_{16}) при обміні даними трактуються як помилки.

Адреса вузла – це адреса одиниці обладнання промислової автоматизації, яка може бути задана одним з двох варіантів. Один з варіантів передбачає використання 8-бітової адресації, при цьому повну адресу вузла (8 біт) утримує старша частина адреси вузла (8 біт), а молодша частина адреси вузла (3 біта) утримує нулі. Таким чином, загалом можна адресувати до $2^8 = 256$ вузлів. Інший варіант передбачає використання 11-бітової адресації, при цьому повну адресу вузла (11 біт) утримують одночасно як старша частина адреси вузла (8 біт), так і молодша частина адреси вузла (3 біта). Таким чином, загалом можна адресувати до $2^{11} = 2048$ вузлів.

Біт запиту – це біт, який визначає, буде певний параметр одиниці обладнання промислової автоматизації читатися чи такий параметр буде записуватися. При читанні біт запиту (1 біт) повинен утримувати лог. 1, а при записі – лог. 0.

Розмір поля даних рівня представлення – це поле, яке утримує розмір поля даних рівня представлення в байтах. Таким чином, поле даних рівня представлення може бути розміром від 0 до $2^4 - 1 = 15$ байтів.

HASH-код ідентифікатора параметра – це ідентифікатор параметра одиниці обладнання промислової автоматизації, який закодований згідно з HASH-алгоритмом. Всі ідентифікатори параметрів в протоколі ОВЕН складаються з 4 символів, які можуть бути тільки наступними (загальна кількість символів становить 40):

- 26 букв англійського алфавіту (від “A” до “Z” або від “a” до “z”, причому різниці між великими і малими буквами немає);

- 10 арабських цифр від “0” до “9”;

- спеціальні символи “–”, “_”, “/” і “пробіл”.

Крім того, кожний з символів може бути доповнений точкою. Таким чином, загальна кількість символів подвоюється і становить 80. Наприклад,

ідентифікатору параметра A.Len буде відповідати HASH-код $1ED2_{16}$.

Значення параметра – це дані, які мають один з наступних форматів:

- число з плаваючою точкою;

- число із знаком з односторонньою десятковою точкою (в двійковому вигляді);

- число із знаком з односторонньою десятковою точкою (в двійково-десятковому вигляді);

- дата/час (в двійковому вигляді);

- дата/час (в двійково-десятковому вигляді);

- число ціле нетипізоване (в двійковому вигляді);

- число ціле нетипізоване (в двійково-десятковому вигляді);

- рядок ASCII-символів.

Часовий модифікатор – це поле, яке утримує інформацію про дату/час (наприклад, дату/час читання значення коефіцієнта підсилення вимірювального каналу багатоканального регулятора).

Індексний модифікатор – це поле, яке утримує інформацію про порядковий номер параметра в одиниці обладнання промислової автоматизації (наприклад, значення 4 в такому полі може свідчити про те, що необхідно записати в багатоканальний регулятор значення коефіцієнта підсилення саме 4-го вимірювального каналу (а не 3-го або 5-го)).

Маючи **Значення параметра** в одному з 8 різних форматів, наявність або відсутність **Часового модифікатора** і наявність або відсутність **Індексного модифікатора**, загалом можна представляти дані одним з 32 різних варіантів.

При обробці даних в протоколі ОВЕН можуть виникнути наступні помилки:

- мережеві помилки – в такому випадку ведений (slave) передає в пакеті ведучому (master) в якості **HASH-коду ідентифікатора параметра** HASH-код ідентифікатора параметра n.Err (який має значення 0233_{16}), а в **Полі даних рівня представлення** – код мережевої помилки і HASH-код того ідентифікатора параметра, який ведучий (master) в пакеті перед цим передав веденому (slave);

- виключні ситуації – в такому випадку ведений (slave) передає в пакеті ведучому (master) той сам пакет, який ведучий (master) йому перед цим передав, але в **Поле даних рівня представлення** додається інформація про виключну ситуацію.

4.5. Протокол HART

Протокол HART (абр. від англ. Highway Addressable Remote Transducer, Віддалений перетворювач, який адресується через основний канал передачі даних) – протокол для обміну даними, в основі якого також лежить архітектура “ведучий-ведений” (“master-slave”).

Він є одним з достатньо розповсюджених протоколів, які використовуються в промисловості.

Основні інтерфейси, які використовують протокол HART: “струмова петля” (4 ... 20) мА.

Загальний принцип роботи протоколу ОВЕН співпадає із загальним принципом роботи протоколу Modbus RTU.

Протокол HART працює “поверх” “струмової петлі” (4 ... 20) мА: при використанні частотної модуляції лог. 1 відповідає одному повному періоду гармонійних коливань частотою 1200 Гц, а лог. 0 – двом неповним гармонійним коливанням частотою 2200 Гц (з амплітудами приблизно 0,5 мА). При цьому спектр “струмової петлі” (4 ... 20) мА не повинен бути більше 25 Гц, а спектр протоколу HART – менше 500 Гц.

Двопровідна “струмова петля” (4 ... 20) мА може одночасно забезпечити:

- інтерфейс у вигляді уніфікованого електричного неперервного сигналу струму (4 ... 20) мА;
- обмін даними по протоколу HART;
- живлення одиниці (кількох одиниць) обладнання промислової автоматизації (принаймні струмом не менше 4 мА).

PREAMBULE – це преамбула (яка утримує не менше 3 байтів FF_{16}), використання якої дозволяє синхронізувати початок обміну даними між ведучим (master) і веденим (slave) (див. рис. 4.6).

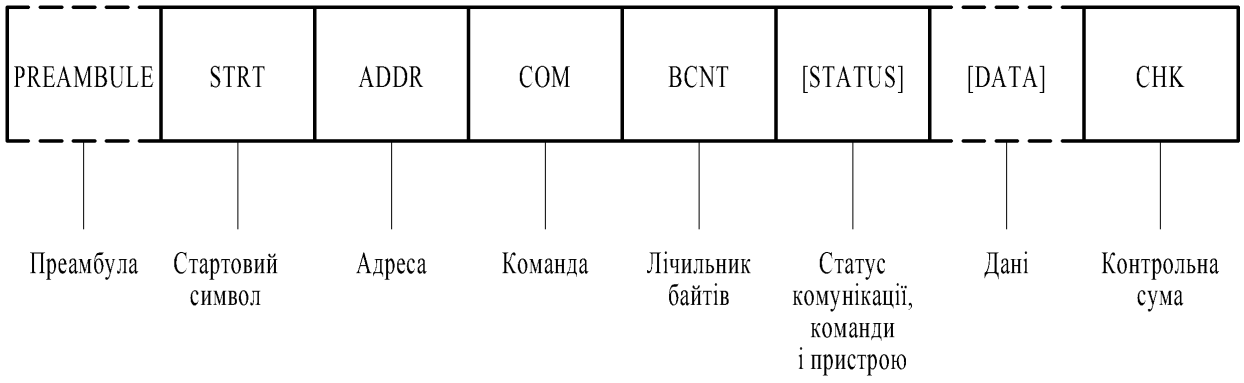


Рис. 4.6. Формат довгого пакета протоколу HART

STRT – це стартовий символ, значення якого визначається наступним:

- пакет буде передаватися від ведучого (master) до веденого (slave), від веденого (slave) до ведучого (master) або від веденого (slave) до ведучого (master) в “пакетному” режимі;

- буде використовуватися короткий формат пакета (більш старий) або довгий формат пакета (більш новий).

ADDR – це адреса, яка утримує як адресу ведучого (master), так і адресу веденого (slave) (протокол HART передбачає наявність одночасно двох ведучих (master)).

COM – це команда, яка може бути наступною:

- універсальною – повинна підтримуватися всіма одиницями обладнання промислової автоматизації, які використовують протокол HART (наприклад, команда 0 показує, що необхідно прочитати унікальний ідентифікатор одиниці обладнання промислової автоматизації);

- поширеною – повинна підтримуватися тільки тими одиницями обладнання промислової автоматизації, де така команда необхідна (наприклад, команда 45 показує, що необхідно налагодити “нуль” цифро-аналогового перетворювача – але, якщо в одиниці обладнання промислової автоматизації цифро-аналоговий перетворювач відсутній, то команда 45 нею підтримуватися не повинна);

- специфічною – є унікальною для одиниці обладнання промислової автоматизації певного виробника (в обладнанні інших виробників може бути відсутня або призначена для зовсім інших дій).

BCNT – це лічильник байтів, який утримує кількість байт в **[STATUS]** і **[DATA]**.

[STATUS] – це статус комунікації, команди і пристрою, який передається тільки від веденого (slave) до ведучого (master) і який утримує:

- інформацію про наявність або відсутність помилок при обміні даними;

- інформацію про прийняту команду;

- інформацію про стан одиниці обладнання промислової автоматизації.

[DATA] – це дані у форматі чисел цілих беззнакових, чисел з плаваючою точкою і/або рядків ASCII-символів (від 0 до 25 байтів).

CHK – це контрольна сума, яка обчислюється як результат побітової операції “Виключне АБО” над усіма байтами повідомлення, крім **PREAMBULE** (тобто над **STRT**, **ADDR**, **COM**, **BCNT**, **[STATUS]** і **[DATA]**).

4.6. Питання для самоперевірки

1. Що таке сімейство протоколів Modbus?
2. Які основні характеристики має протокол Modbus RTU?
3. Який формат має пакет **ADU** протоколу Modbus RTU?
4. Які основні характеристики має протокол Modbus ASCII?
5. Який формат має пакет **ADU** протоколу Modbus ASCII?
6. Які основні характеристики має протокол Modbus TCP/IP?
7. Який формат має пакет **ADU** протоколу Modbus TCP/IP?
8. Яке призначення функції 01_{16} сімейства протоколів Modbus?
9. Яке призначення функції 02_{16} сімейства протоколів Modbus?
10. Яке призначення функції 03_{16} сімейства протоколів Modbus?
11. Яке призначення функції 04_{16} сімейства протоколів Modbus?
12. Яке призначення функції 05_{16} сімейства протоколів Modbus?
13. Яке призначення функції 06_{16} сімейства протоколів Modbus?
14. Яке призначення функції $0F_{16}$ сімейства протоколів Modbus?
15. Яке призначення функції 10_{16} сімейства протоколів Modbus?
16. Які основні характеристики має протокол DCON?
17. Який формат має пакет протоколу DCON (в загальному вигляді)?
18. Які основні характеристики має протокол ОВЕН?
19. Який формат має пакет протоколу ОВЕН?
20. Які основні характеристики має протокол HART?
21. Який формат має довгий пакет протоколу HART?

4.7. Рекомендована література

Базова література: [1, 4–6].

Додаткова література: [1, 3, 5, 6].

5. ПРОГРАМНІ ЗАСОБИ РОЗШИРЕННЯ МОЖЛИВОСТЕЙ SCADA-СИСТЕМ

5.1. Використання типових алгоритмів обробки даних

5.1.1. Необхідність і доцільність додаткової обробки даних

Виконуючи своє основне призначення, яке полягає в забезпеченні людино-машинного інтерфейсу, SCADA-системи часто здійснюють обмін даними без їх попередньої обробки (наприклад, безпосереднє відображення на мнемосхемі результату вимірювання фізичної величини). Іноді цього недостатньо, так як часто виникає потреба в обмеженні, фільтрації, масштабуванні, калібруванні, лінеаризації, корекції і інших аналогічних операціях над даними. Враховуючи те, що SCADA-системи (як пакети програм, які виконуються на засобах обчислювальної техніки) оперують дискретними даними (тобто такими, які отримані або передані в певні моменти часу), такі дані можуть додатково оброблятися згідно з відомими алгоритмами обробки дискретних даних (наприклад, чисельного диференціювання або інтегрування).

5.1.2. Знаходження середнього арифметичного при прийманні даних від об’єкта автоматизації

Середнє арифметичне $In_{\text{сер.ариф.}}$ (маючи n значень In_{i-n+1}, In_{i-n+2}, …, In_{i-1} і In_i даних, прийнятих останніми від об’єкта автоматизації) може бути знайдено наступним чином:

$$In_{\text{сер.ариф.}} = \frac{In_{i-n+1} + In_{i-n+2} + \ldots + In_{i-1} + In_i}{n}.$$

Для кожного наступного i-го приймання даних процес обчислення повинен повторюватися. Наприклад, знаходження середнього арифметичного значень концентрації діоксиду вуглецю CO_2 на об’єкті автоматизації дозволить підвищити достовірність такого вимірювання.

5.1.3. Знаходження похідної при прийманні даних від об’єкта автоматизації

Похідна $In_{\text{пох.}}$ (маючи три значення In_{i-2}, In_{i-1} і In_i даних) може бути знайдена наступним чином:

$$In_{\text{пох.}} = \frac{-3In_i - 4In_{i-1} + In_{i-2}}{2\Delta t},$$

де Δt – інтервал часу між вимірюваннями In_{i-2} і In_{i-1} та In_{i-1} і In_i.

Для находження похідної можуть бути використані і інші формули диференціювання, зокрема, основані на першій формулі Ньютона або багаточлені Лагранжа. Отримання такої похідної може бути використане, наприклад, при формуванні диференціюючої складової ПІД-регулятора.

5.1.4. Знаходження інтеграла при прийманні даних від об'єкта автоматизації

Інтеграл $In_{\text{інт.}}$ (маючи n значень In_{i-n+1}, In_{i-n+2}, …, In_{i-1} і In_i даних) може бути знайдений наступним чином (використовується формула трапецій):

$$In_{\text{інт.}} = \frac{(In_{i-n+1} + 2In_{i-n+2} + \ldots + 2In_{i-1} + In_i)\Delta t}{2},$$

де Δt – інтервал часу між вимірюваннями In_{i-n+1} і In_{i-n+2}, In_{i-n+2} і In_{i-n+3} і так далі.

Для находження інтеграла можуть бути використані і інші формули інтегрування, зокрема, основані на формулі Сімпсона або формулах Ньютона-Котекса. Отримання такої похідної може бути використане, наприклад, при формуванні інтегруючої складової ПІД-регулятора.

5.1.5. Коригування даних при їх прийманні від об'єкта автоматизації

Скориговане значення даних $In_{\text{кор.}_0}$ (маючи останнє значення In_i даних і використовуючи корекцію 0-го порядку) може бути знайдено наступним чином:

$$In_{\text{кор.}_0} = In_i + k_0,$$

де k_0 – постійний коефіцієнт.

Скориговане значення даних $In_{\text{кор.}_1}$ (маючи те саме значення даних і використовуючи корекцію 1-го порядку) може бути знайдено наступним чином:

$$In_{\text{кор.}_0} = k_1 In_i + k_0,$$

де k_1 – лінійний коефіцієнт; k_0 – постійний коефіцієнт.

Скориговане значення даних $In_{\text{кор.}_2}$ (маючи те саме значення даних і використовуючи корекцію 2-го порядку) може бути знайдено наступним чином:

$$In_{\text{кор.}_0} = k_2 In_i^2 + k_1 In_i + k_0 ,$$

де k_2 – квадратичний коефіцієнт; k_1 – лінійний коефіцієнт; k_0 – постійний коефіцієнт.

Найчастіше використовується коригування 0-го і 1-го порядків, коригування 2-го порядку використовується рідко, коригування більш високих порядків практично не використовується.

5.1.6. Перетворення даних при їх прийманні від об’єкта автоматизації

Перетворення даних $In_{\text{пер.}}$ (маючи останнє значення In_i даних) може бути здійснено наступним чином:

$$In_{\text{пер.}} = k_1 In_i + k_0 ,$$

де k_1 – лінійний коефіцієнт; k_0 – постійний коефіцієнт.

Найчастіше перетворення даних використовується при перетворенні уніфікованих електричних неперервних сигналів в значення певної фізичної величини (в припущенні, що залежність між такими сигналом і фізичною величиною є лінійною). Наприклад, діапазону зміни сигналу від 4 до 20 мА може відповідати діапазон зміни температури від 0 до 300 °С, а залежність $In_{\text{пер.}}(In_i)$ буде мати наступний вигляд:

$$In_{\text{пер.}}(In_i) = 18{,}75 In_i - 75 .$$

5.1.7. Обмеження даних їх мінімальним і максимальним значеннями при їх передачі до об’єкта автоматизації

Знаходження мінімального значення $Out_{\text{мін.}}$ і максимального значення $Out_{\text{макс.}}$ даних (маючи останнє Out_i значення даних, які повинні бути передані до об’єкта автоматизації) може бути здійснено наступним чином:

$$Out_{\text{мін.}} = \min\{Out_i\} ,$$

$$Out_{\text{макс.}} = \max\{Out_i\} .$$

Отримання мінімального і максимального значення даних, яке повинно

бути передано, і обмеження його, якщо воно буде завеликим або замалим, наприклад, ступеня відкриття дроселя для подачі стисненого повітря до пневматичної системи об’єкта автоматизації, дозволить зменшити “пневматичний удар” на такий об’єкт.

5.1.8. Обмеження швидкості зміни даних їх мінімальним і максимальним значеннями при їх передачі до об’єкта автоматизації

Знаходження швидкості зміни даних $Out_{\text{шв.}}$ (маючи два останніх значення Out_{i-1} і Out_i даних) може бути здійснено наступним чином:

$$In_{\text{пох.}} = \frac{-Out_{i-1} + Out_i}{\Delta t},$$

де Δt – інтервал часу між значеннями даних Out_{i-1} і Out_i.

Отримання похідної з двох значень даних, які повинні бути передані, і обмеження останнього з них, якщо похідна буде завеликою або замалою, наприклад, швидкості зміни кута повороту вентиля для подачі насиченої пари до теплової системи об’єкта автоматизації, дозволить зменшити “тепловий удар” на такий об’єкт.

5.2. Використання мов програмування промислових контролерів

5.2.1. Необхідність і доцільність використання мов програмування промислових контролерів

Незважаючи на достатню потужність сучасних промислових комп’ютерів, на яких виконуються SCADA-системи, і достатню пропускну спроможність середовищ обміну даними попередня обробка даних при взаємодії з об’єктами автоматизації звичайно здійснюється за допомогою програмованих контролерів (або, якщо алгоритм попередньої обробки даних нескладний, за допомогою програмованих реле). Таким чином, SCADA-системи мають можливість здійснювати збір вже підготовлених даних, що підвищує достовірність таких даних і одночасно пришвидшує роботу SCADA-систем.

Звичайно для програмування промислових контролерів використовується одна або кілька мов програмування, визначених в стандарті IEC 61131-3 (3-я редакція вийшла в 2013 р.), причому в деяких випадках кілька мов можуть використовуватися одночасно. Ефективність програмного коду, незалежно від мови програмування, яка була використана, є приблизно однаковою.

5.2.2. Мова програмування IL (Instruction List)

Мова програмування IL, яка має текстовий “інтерфейс”, є основною мовою програмування стандарту IEC 61131-3, хоча вважається застарілою. По суті вона представляє собою “список інструкцій”, близький за синтаксисом до мови програмування Assembler. Її основне призначення полягає в створенні невеликого за обсягом програмного коду, структура якого є повністю зрозумілою. Приклад додавання числа 20 і змінної **x** типу **INT** за допомогою оператора **ADD** на мові програмування IL в середовищі програмування CoDeSys version 2.3, результат якого буде збережений в змінній **z** типу **INT**, наведено нижче.

```
LD   20
ADD  x
ST   z
```

Приклад вибору з числа 150 і змінної **x** типу **DINT** за допомогою оператора **SEL**, результат якого буде збережений в змінній **z** типу **DINT**, також наведено нижче.

```
LD   TRUE
SEL  150, x
ST   z
```

5.2.3. Мова програмування ST (Structured Text)

Мова програмування ST, яка має текстовий “інтерфейс”, є основною мовою програмування стандарту IEC 61131-3. По суті вона представляє собою “список інструкцій”, близький за синтаксисом до мови програмування Pascal. Її основне призначення полягає в створенні невеликого або середнього за обсягом програмного коду, який матиме гарну “читабельність”. Приклад побітової логічної операції AND над числом 10010001_2 і змінною **x** типу **BYTE** за допомогою оператора **AND** на мові програмування ST в середовищі програмування CoDeSys version 2.3, результат якого буде збережений в змінній **z** типу **BYTE**, наведено нижче.

```
z := 2#1001_0001 AND x ;
```

Приклад порівняння числа 400 і змінної **x** типу **WORD**, результат якого

буде збережений в змінній **z** типу **BOOL**, також наведено нижче.

```
z := 400 < x ;
```

5.2.4. Мова програмування LD (Ladder Diagram)

Мова програмування LD, яка має графічний “інтерфейс”, є основною мовою програмування стандарту IEC 61131-3. Її основне призначення полягає в створенні програмного коду, який реалізує релейно-контактну логіку. Приклад логічної операції (x1 OR x2) AND (x3 OR x4) над чотирма змінними **x1**, **x2**, **x3** і **x4** типу **BOOL** за допомогою операторів **AND** і **OR** на мові програмування LD в середовищі програмування CoDeSys version 3.5, результат якого буде збережений в змінній **z** типу **BOOL**, наведено нижче на рис. 5.1.

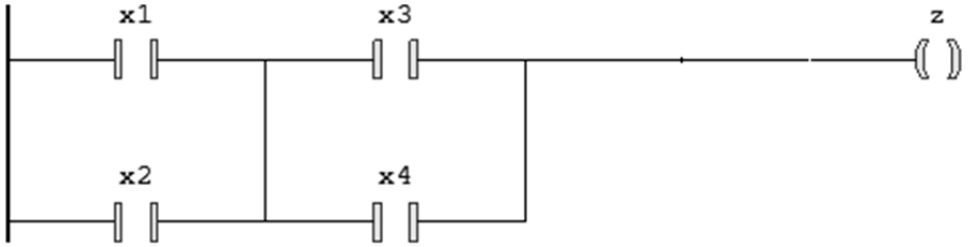


Рис. 5.1. Логічна операція (x1 OR x2) AND (x3 OR x4)

Приклад реалізації RS-тригера за допомогою бістабільного функціонального блока типу **RS** з вхідними змінними **x1** і **x2** типу **BOOL** та вихідною змінною **z** типу **BOOL** також наведено нижче на рис. 5.2.

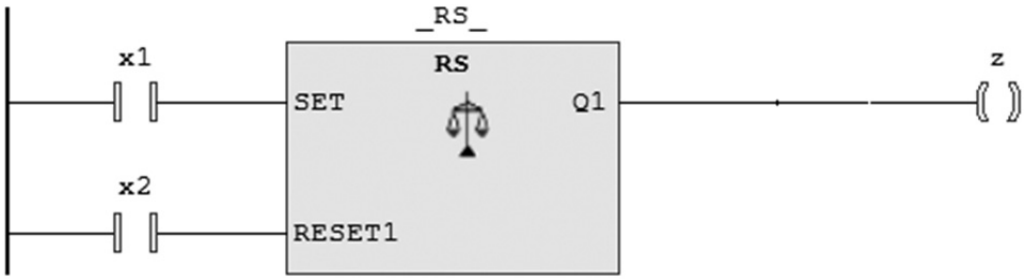


Рис. 5.2. Бістабільний функціональний блок типу RS

Змінні **x1** і **x2** також можуть відповідати, наприклад, дискретним входам програмованого контролера, а змінна **z** – дискретному виходу такого контролера.

5.2.5. Мова програмування FBD (Function Block Diagram)

Мова програмування FBD, яка має графічний “інтерфейс”, є основною мовою програмування стандарту IEC 61131-3. Її основне призначення полягає в створенні нескладного для розуміння і добре структурованого програмного коду, який може створюватися навіть користувачами, які не є спеціалістами з програмування. Приклад піднесення змінної **x1** типу **REAL** в ступінь, який визначається змінною **x2** типу **REAL**, за допомогою оператора типу **EXPT** з подальшим множенням на змінну **x3** типу **REAL** за допомогою оператора типу **MUL** на мові програмування FBD в середовищі програмування CoDeSys version 3.5, результат якого буде збережений в змінній **z** типу **REAL**, наведено нижче на рис. 5.3.

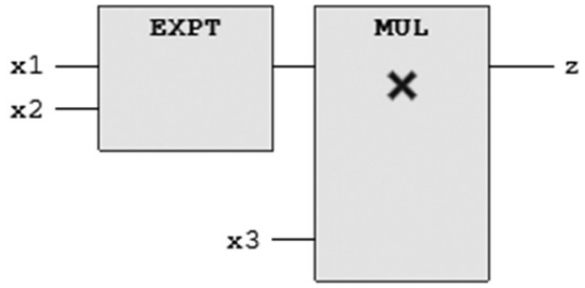


Рис. 5.3. Оператори типу EXPT і MUL

Приклад віднімання від змінної **x1** типу **REAL** змінної **x2** типу **REAL** за допомогою оператора типу **SUB**, результат якого буде збережений в змінній **z** типу **REAL**, також наведено нижче на рис. 5.4.

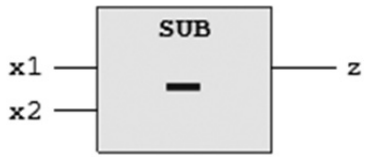


Рис. 5.4. Оператор типу SUB

Змінні **x1** і **x2** можуть відповідати, наприклад, аналоговим входам програмованого контролера, а змінна **z** – аналоговому виходу такого контролера.

5.2.6. Мова програмування SFC (Sequential Function Chart)

Мова програмування SFC, яка має графічний “інтерфейс”, є додатковою мовою програмування стандарту IEC 61131-3. Її основне призначення полягає в створенні середнього або великого за обсягом програмного коду, причому на

ній звичайно створюється тільки “каркас” такого програмного коду, а наповнення такого “каркаса” може бути здійснене на інших мовах програмування стандарту IEC 61131-3. Приклад виконання кроків **Init**, **_Step_1_** і **_Step_2_** за умови доступності відповідних їм переходів **TRUE**, **x1<10** і **x2>20** на мові програмування SFC в середовищі програмування CoDeSys version 3.5 наведено на рис. 5.5.

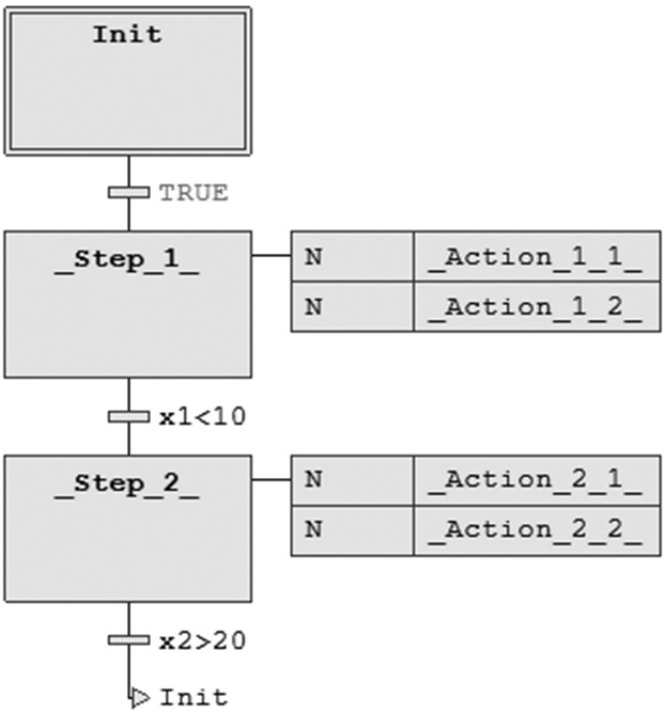


Рис. 5.5. Кроки Init, _Step_1_ і _Step_2_ та переходи TRUE, x1<10 і x2>20

Перший з кроків (крок з умовою **TRUE**) доступний завжди, другий – за умови, коли змінна **x1** типу **WORD** менше 10, третій – коли змінна **x2** типу **WORD** більше 20. При цьому програмний код кроків **Init**, **_Step_1_** і **_Step_2_** спочатку може бути повністю відсутнім. Змінні **x1** і **x2** можуть відповідати, наприклад, внутрішнім змінним програмованого контролера, отриманим в результаті певних розрахунків.

5.2.7. Мова програмування CFC (Continuous Function Chart)

Мова програмування CFC, яка має графічний “інтерфейс”, в стандарті IEC 61131-3 не наведена. Вона є розвитком мови програмування FBD і в основному призначена для створення середнього або великого за обсягом програмного коду, дозволяючи розміщувати елементи свого графічного “інтерфейсу” довільним чином.

Приклад обмеження змінної **x1** типу **REAL** мінімальним значенням 40 і максимальним значенням 160 за допомогою оператора типу **LIMIT** з подальшим діленням на змінну **x2** типу **REAL** за допомогою оператора типу **DIV** на мові програмування CFC в середовищі програмування CoDeSys version 3.5, результат якого буде збережений в змінній **z** типу **REAL**, наведено нижче на рис. 5.6.

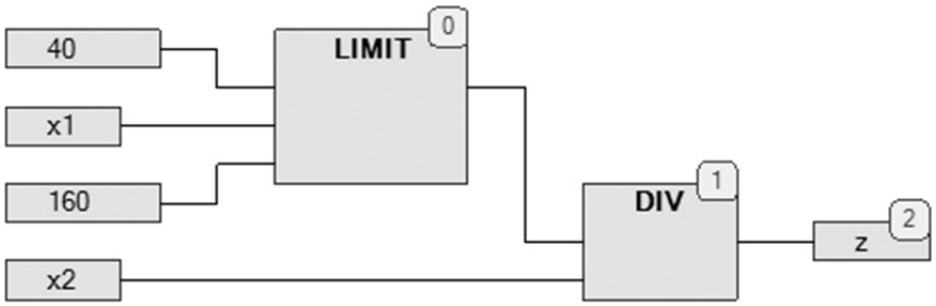


Рис. 5.6. Оператори типу LIMIT і DIV

Змінні **x1** може відповідати, наприклад, аналоговому входу програмованого контролера, змінна **x2** – внутрішній змінній програмованого контролера, а змінна **z** – аналоговому виходу такого контролера.

5.3. Використання мов структурованих запитів

5.3.1. Необхідність і доцільність використання мов структурованих запитів

Промислова автоматизація, частиною якої є SCADA-системи, звичайно “генерує” велику кількість даних, таких як значення виміряних фізичних величин, виниклі тривоги і події, зміни в умовах протікання виробничих (технологічних) процесів, звітності і так далі. Звичайно для забезпечення постійного доступу до таких даних використовуються модулі збирання даних, але великий обсяг таких даних, їх різнорідність, необхідність постійного оновлення і обробки, обмеження доступу до них і так далі приводять до того, що використання модулів збирання даних стає недоцільним.

Заміною таким модулям є використання SQL-баз даних і відповідних систем керування ними на основі мов структурованих запитів SQL (наприклад, однією з таких систем керування є Microsoft SQL Server розробки Microsoft Corporation, в основі якої лежить мова структурованих запитів Transact SQL (стандартизована мова SQL з певними змінами)). Звичайно SQL-бази даних

розміщуються у виглядів одного або кількох файлів на одному або кількох накопичувачах, які, в свою чергу, знаходяться або “локально” на одному персональному/промисловому комп’ютері, сервері, робочій станції і так далі, або “розподілено” в різних місцях промислової мережі.

5.3.2. Мова структурованих запитів SQL

Особливістю зберігання даних в SQL-базах даних є те, що доступ до них є можливим тільки через систему керування такими базами, а безпосередній доступ (наприклад, шляхом безпосереднього відкриття, обробки і закриття файлів) заборонений. Перевагою такого зберігання є можливість зберігання дуже великих обсягів різнорідних даних, велика швидкодія і надійність (наприклад, можна одночасно зберігати значення виміряних фізичних величин від кількох сотень вимірювальних перетворювачів, причому такі дані можуть постійно оновлюватися).

Мову SQL можна умовно розділити на 4 складові:

- Data Control Language (DCL) – забезпечує контроль доступу до даних;
- Data Definition Language (DDL) – забезпечує опис даних;
- Data Manipulation Language (DML) – забезпечує маніпулювання даними;
- Transaction Control Language (TCL) – забезпечує контроль виконання транзакцій.

Мова SQL дозволяє використовувати наведені нижче SQL-команди, що дає можливість керування даними в SCADA-системах.

1. **CREATE DATABASE** і **DROP DATABASE** – дозволяють відповідно створювати і видаляти бази даних.
2. **CREATE TABLE** і **DROP TABLE** – дозволяють відповідно створювати і видаляти таблиці баз даних.
3. **INSERT** і **DELETE** – дозволяють відповідно створювати і видаляти записи таблиць баз даних.
4. **UPDATE** – дозволяє оновлювати записи таблиць баз даних.
5. **SELECT** – дозволяє формувати результуючий набір даних з таблиць баз даних.
6. **CREATE FUNCTION** і **DROP FUNCTION** – дозволяють відповідно створювати і видаляти функцій баз даних, які визначаються користувачами.
7. **CREATE PROCEDURE** і **DROP PROCEDURE** – дозволяють відповідно створювати і видаляти збережувані процедури баз даних, які визначаються користувачами.

8. **CREATE TRIGGER** і **DROP TRIGGER** – дозволяють відповідно створювати і видаляти тригери баз даних, які визначаються користувачами.

9. **CREATE VIEW** і **DROP VIEW** – дозволяють відповідно створювати і видаляти представлення баз даних.

10. **DECLARE CURSOR**, **OPEN**, **CLOSE** і **DEALLOCATE** – дозволяють відповідно об'являти, відкривати, закривати і звільнювати курсори баз даних.

Крім того, існує багато інших SQL-команд, які розширюють таку можливість шляхом синхронізації даних, обробки помилок при обміні даними, обробки транзакцій, явного перетворення даних, які мають різні типи і так далі.

Приблизна послідовність дій для зберігання даних в SQL-базах даних має бути наступною.

1. Створення однієї або кількох SQL-бази даних (використовуючи **CREATE DATABASE**).

2. Створення в кожній SQL-базі даних кількох таблиць згідно з типом даних, які повинні зберігатися (використовуючи **CREATE TABLE**).

3. Створення в кожній SQL-базі даних кількох функцій, збережуваних процедур, тригерів, представлень, курсорів і так далі, призначених для обробки даних.

4. Створення в кожній SQL-базі даних потрібних даних (використовуючи **INSERT**), за необхідності оновлюючі їх (використовуючи **UPDATE**) або видаляючи (використовуючи **DELETE**). Під час знаходження SCADA-системи в режимі виконання такий процес може відбуватися постійно.

5. Формування (одночасно з створенням, оновленням або видаленням даних) результуючих наборів даних (використовуючи **SELECT**). Під час знаходження SCADA-системи в режимі виконання такий процес також може відбуватися постійно.

6. Видалення непотрібних складових SQL-баз даних (функцій, збережуваних процедур і так далі) або самих SQL-баз даних.

5.4. Використання технологій World Wide Web

5.4.1. Необхідність і доцільність використання технологій World Wide Web

Людино-машинний інтерфейс SCADA-систем переважно реалізується за допомогою мнемосхем, які розміщуються на персональних/промислових комп'ютерах, серверах, робочих станціях і так далі. Відповідно, доступ до

таких мнемосхем можна отримати тільки з цих засобів обчислювальної техніки. Але часто є необхідність отримати доступ до такого інтерфейсу через локальну або глобальну мережу (зокрема, через Internet), не використовуючи при цьому пакети програм SCADA-систем – тобто як до звичайних ресурсів, розміщених в World Wide Web.

Звичайно така необхідність виникає, коли в якості обладнання промислової автоматизації використовуються програмовані контролери, програмовані реле, панелі оператора і так далі. Найпростіше це зробити, використовуючи Web-візуалізацію – але для цього потрібно, щоб таке обладнання мало убудований Web-сервер (крім того, можливості Web-візуалізації можуть бути набагато більшими, якщо таке обладнання буде мати убудовану операційну систему). Доступ до Web-візуалізації може здійснюватися за допомогою звичайного Web-браузера.

5.4.2. Основні можливості убудованих Web-серверів

Звичайно убудовані Web-сервери можуть забезпечити наступні можливості.

1. Зберігання ресурсів убудованого Web-сайту (HTML-документів, CSS-стилів, JavaScript-файлів і так далі).
2. Перевірка умов доступу до Web-сайту користувачів, які можуть мати різні привілеї.
3. Ведення журналу, в якому утримуються зміни в режимі роботи Web-сайту.
4. Підтримка як статичних HTML-документів, так і динамічних HTML-документів.
5. Підтримка протоколу обміну даними HTTP (абр. від англ. Hypertext Transfer Protocol, Протокол передачі гіпертексту).
6. Підтримка протоколу обміну даними HTTPS (абр. від англ. Hypertext Transfer Protocol Secure, Безпечний протокол передачі гіпертексту).
7. Підтримка протоколу обміну даними (абр. від англ. File Transfer Protocol, Протокол передачі файлів).
8. Підтримка криптографічного протоколу SSL (абр. від англ. Secure Sockets Layer, Рівень безпечних “сокетів“).
9. Підтримка криптографічного протоколу TLS (абр. від англ. Transport Layer Security, Безпечний транспортний рівень).

5.5. Використання хмарних технологій

5.5.1. Необхідність і доцільність використання хмарних технологій

Системи промислової автоматизації, які використовуються в різних галузях промисловості, звичайно мають суттєві відмінності в режимах своєї роботи. Відповідно, робота кожної з SCADA-систем повинна бути “прив’язана” до роботи “своєї“ системи промислової автоматизації. Наприклад, може знадобитися, щоб SCADA-система працювала цілодобово або забезпечувала доступ до даних кількох об’єктів автоматизації, які розміщені на великих відстанях один від одного.

Забезпечити розв’язання деяких з таких задач силами тільки однієї SCADA-системи часто є недоцільним по технічним, економічним або іншим причинам. Виходом з цього є використання розміщених віддалено сервісів (хмарних сервісів), які мають можливість зберігати дані, виконувати над даними обчислення, повідомляти про зміни даних і так далі.

5.5.2. Основні можливості хмарних сервісів

Звичайно хмарні сервіси можуть забезпечити наступні можливості як доповнення (а іноді і як заміна) можливостей SCADA-систем.

1. Оперативний обмін даними з об’єктом автоматизації в заданий момент часу (за умови підключення його до глобальної мережі (зокрема, до Internet)).
2. Збереження даних об’єкта автоматизації впродовж заданого інтервалу часу.
3. Відображення даних об’єкта автоматизації безпосередньо на хмарних сервісах.
4. Повідомлення про тривоги і події за допомогою електронної пошти і інших аналогічних проводових і безпроводових засобів.
5. Виконання перетворень даних, розрахунків, симуляцій і інших обчислень.
6. Можливість інтеграції з MES (системою організації виробництва), а також з ERP (системою планування ресурсів підприємства) і MRP (системою планування ресурсів виробництва).
7. Різноманітні API (Application Programming Interface) і інші аналогічні засоби для віддаленого доступу до хмарного сервісу.

5.6. Кібербезпека SCADA-систем

5.6.1. Необхідність і доцільність використання кібербезпеки SCADA-систем

Серйозні питання кібербезпеки SCADA-систем за умови, коли обмін даними здійснюється в локальній мережі підприємства, звичайно не виникають. Ситуація принципово змінюється, коли такій обмін здійснюється через глобальну мережу (зокрема, через Internet) – в такому випадку для кібербезпеки SCADA-систем потрібно використовувати комплексний підхід, який полягає в одночасному проведенні низки безпекових заходів. Необхідно пам’ятати, що SCADA-системи надають доступ до промислової автоматизації, і результати несанкціонованого порушення існуючих станів виробничих (технологічних) процесів можуть бути катастрофічними.

5.6.2. Основні можливості кібербезпеки SCADA-систем

Загальні рекомендації, яким чином можна якщо не виключити повністю, то звести до мінімуму вплив на SCADA-систему внутрішніх і зовнішніх спроб несанкціонованого доступу, полягають в наступному.

1. Відключення від глобальної мережі тих складових SCADA-системи, для яких такий доступ не є життєво важливим. Особливо, коли в якості середовища обміну даними використовуються електромагнітні хвилі.

2. Виявлення і зведення до мінімуму всіх можливих підключень до SCADA-системи – як фізичних (наприклад, інтерфейсів USB), так і програмних (наприклад, клієнтських і/або серверних “сокетів”).

3. Проводити на постійній основі перевірки умов доступу до SCADA-системи (наприклад, коли деякі з користувачів мають підвищені привілеї). При цьому важливо мати можливість в будь-який час такий доступ надійно заблокувати.

4. Постійне виявлення спроб несанкціонованого доступу як шляхом використання відповідного апаратного і/або програмного забезпечення (менш витратний варіант), так і шляхом використання обслуговуючого персоналу, який спеціалізується на кібербезпеці (більш витратний варіант).

5. Бажано мати план швидкого реагування якщо не на всі типи несанкціонованого доступу, то хоча б на найбільш небезпечні з них.

Такий план повинен “відповідати” на наступні основні питання:

- “Що можна і потрібно роботи до, під час і після несанкціонованого доступу?”;

- “Наскільки серйозним є “витік” даних, чи можна буде їх відновити?”;

- “Чи позначився “витік” даних на роботі SCADA-системи і як саме?”;

- “Чи потрібно обмежити або повністю відключити обмін даними через глобальну мережу (зокрема, через Internet)?”;

- “Як в подальшому можна підготуватися до аналогічних спроб несанкціонованого доступу?”.

6. Обслуговуючий персонал, який спеціалізується на кібербезпеці, повинен проходити (наскільки це можливо) постійні тренування і підвищення кваліфікації (це можна пояснити тим, що типи несанкціонованого доступу, які використовують кіберзлочинці, теж постійно вдосконалюються).

5.7. Питання для самоперевірки

1. Що таке типові алгоритми обробки даних?
2. В чому полягає необхідність і доцільність додаткової обробки даних?
3. Як можна знайти середнє арифметичне при прийманні даних?
4. Як можна знайти похідну при прийманні даних?
5. Як можна знайти інтеграл при прийманні даних?
6. Як можна здійснити коригування даних при їх прийманні?
7. Як можна здійснити перетворення даних при їх прийманні?
8. Як можна здійснити обмеження даних їх мінімальним і максимальним значеннями?
9. Як можна здійснити обмеження швидкості зміни даних їх мінімальним і максимальним значеннями?
10. В чому полягає необхідність і доцільність використання мов програмування промислових контролерів?
11. Що таке мова програмування IL?
12. Що таке мова програмування ST?
13. Що таке мова програмування LD?
14. Що таке мова програмування FBD?
15. Що таке мова програмування SFC?
16. Що таке мова програмування CFC?
17. В чому полягає необхідність і доцільність використання мов структурованих запитів?
18. Що таке мова структурованих запитів SQL?

19. В чому полягає необхідність і доцільність використання технологій World Wide Web?

20. Які основні можливості мають убудовані Web-сервери?

21. В чому полягає необхідність і доцільність використання хмарних технологій?

22. Які основні можливості мають хмарні сервіси?

23. В чому полягає необхідність і доцільність використання кібербезпеки SCADA-систем?

24. Які основні можливості має кібербезпека SCADA-систем?

5.8. Рекомендована література

Базова література: [4–6].

Додаткова література: [1, 2, 3, 5, 6].

6. ОБЛАДНАННЯ ДЛЯ SCADA-СИСТЕМ

6.1. Основні принципи вибору обладнання

В залежності від того, на якому етапі життєвого циклу SCADA-систем обирається для них обладнання, можна запропонувати ті або інші підходи до такого обрання. Виходячи з того, що основне призначення SCADA-систем – робота в складі промислової автоматизації, вони повинні (в ідеалі) працювати виключно на обладнанні, які призначено для роботи у виробничих умовах.

Враховуючи необхідність в захисті такого обладнання від небажаного впливу навколишнього середовища, вартість обладнання для SCADA-систем звичайно в $2 \ldots 3$ рази вище обладнання з схожими технічними характеристиками (продуктивність процесора, об'єм пам'яті і так далі), але призначеного для використання в побутових умовах. Основна вимога до обладнання для SCADA-систем (а загалом і до іншого обладнання промислової автоматизації) – не швидка, а надійна робота в складних умовах експлуатації протягом достатньо довгого інтервалу часу. Причому бажано, щоб втручання в роботу обладнання SCADA-систем (наприклад, з метою оновлення програмного забезпечення) було мінімальним. Таке обладнання іноді повинно безперервно працювати протягом достатньо довгого часу – відповідно, життєвий цикл такого обладнання може становити до 10, а іноді і до 15 років. Необхідно пам'ятати, що сукупна вартість всіх складових об'єкта автоматизації і/або вартість ліквідації аварії на такому об'єкті іноді може в десятки, сотні, тисячі раз перевищувати вартість обладнання для SCADA-систем.

Більшість обладнання для SCADA-систем повинно працювати в широкому діапазоні змін температур, відносних вологостей і тисків навколишнього середовища (іноді достатньо агресивного), мати стійкість до вібрації і ударів, бути водо- і пилонепроникним, не реагувати на зовнішні електричні і магнітні поля. Відновлення роботи такого обладнання повинно займати мінімум часу, так як вартість навіть одного простою виробництва може в багато раз перевищувати вартість всього обладнання для SCADA-систем.

Для випуску обладнання для SCADA-систем звичайно використовують цілий "спектр" різних технічних рішень – наприклад, корпуси можуть мати додаткові ребра жорсткості, ізоляція проводів і кабелів та покриття контактів можуть мати збільшену товщину і так далі. Одними з виробників такого обладнання є ТОВ "АКУТЕК" (Україна), Cincoze Co., Ltd. (Тайвань), Vecow Co., Ltd. (Тайвань), Advantech Co., Ltd. (Тайвань).

6.2. Вибір апаратного забезпечення: загальні рекомендації

6.2.1. Модулі аналогового уведення

Типовим “представником” обладнання для SCADA-систем, яке дозволяє здійснювати приймання аналогових даних від об’єкта автоматизації, є модуль аналогового уведення МВ110-2А (далі – МВ110-2А) (див. рис. 6.1).

Рис. 6.1. Зовнішній вигляд МВ110-2А

МВ110-2А призначений для уведення і перетворення в цифрову форму температури (при використанні в якості вимірювальних перетворювачів термоелектричних перетворювачів і термометрів опору) і опору (при використанні в якості вимірювальних перетворювачів резисторів і потенціометрів), інших фізичних величин (тиску, вологості, витрати, рівня і так далі), значення яких за допомогою вимірювальних перетворювачів може бути перетворено в уніфіковані сигнали напруги або струму в одиницях вимірювання фізичної величини або у відсотках від діапазону можливої зміни фізичної величини, а також стану “сухих контактів” з подальшим обміном даними, отриманими в результаті здійснення вимірювання, за допомогою інтерфейсу RS-485. Основні технічні характеристики МВ110-2А наведено в табл. 6.1.

Табл. 6.1. Основні технічні характеристики МВ110-2А

№ з/п	Параметр	Значення
1	**2**	**3**
1	Кількість аналогових входів	2
2	Межа основної приведеної похибки вимірювання	$\pm 0{,}25$ % (для термоелектричних перетворювачів $\pm 0{,}5$ %)

Продовження табл. 6.1.

1	2		3
3	Розрядність убудованого аналого-цифрового перетворювача		16 біт
4	Час оновлення даних одного аналогового входу		(0,4 ... 0,8) с
5	Підтримувані інтерфейси		RS-485
6	Підтримувані протоколи		Modbus RTU, Modbus ASCII, DCON
7	Напруга живлення	змінного струму	(90 ... 264) В частотою (47 ... 63) Гц
		постійного струму	(18 ... 30) В

6.2.2. Модулі аналогового виведення

Типовим “представником” обладнання для SCADA-систем, яке дозволяє здійснювати передачу аналогових даних до об’єкта автоматизації, є модуль аналогового виведення МВ110-6У (далі – МВ110-6У) (див. рис. 6.2).

Рис. 6.2. Зовнішній вигляд МВ110-6У

МВ110-6У призначений для обміну даними, отриманими в результаті здійснення керування, за допомогою інтерфейсу RS-485 з подальшим перетворенням їх в аналогову форму і виведенням у вигляді уніфікованих сигналів напруги з межами допустимих змін (0 ... 10) В. Основні технічні характеристики МВ110-6У наведено в табл. 6.2.

Табл. 6.2. Основні технічні характеристики МВ110-6У

№ з/п	Параметр	Значення
1	2	3
1	Кількість аналогових виходів	6

Продовження табл. 6.2.

1	2		3
2	Межа основної приведеної похибки керування		±0,5 %
3	Розрядність убудованого цифро-аналогового перетворювача		10 біт
4	Час оновлення даних одного аналогового виходу		< 0,1 с
5	Підтримувані інтерфейси		RS-485
6	Підтримувані протоколи		Modbus RTU, Modbus ASCII, DCON
7	Напруга живлення	змінного струму	(90 ... 264) В частотою (47 ... 63) Гц
		постійного струму	(18 ... 30) В

6.2.3. Модулі дискретного уведення

Типовим “представником” обладнання для SCADA-систем, яке дозволяє здійснювати приймання дискретних даних від об’єкта автоматизації, є модуль дискретного уведення МВ110-16Д (далі – МВ110-16Д) (див. рис. 6.3).

Рис. 6.3. Зовнішній вигляд МВ110-16Д

МВ110-16Д призначений для уведення і перетворення в цифрову форму стану “сухих контактів” і ключів типу n-p-n з подальшим обміном даними, отриманими в результаті здійснення вимірювання, за допомогою інтерфейсу RS-485. Основні технічні характеристики МВ110-16Д наведено в табл. 6.3.

Табл. 6.3. Основні технічні характеристики МВ110-16Д

№ з/п	Параметр	Значення
1	2	3
1	Кількість дискретних входів	16

Продовження табл. 6.3.

1	2		3
2	Максимальна частота сигналу на дискретному вході		1 кГц
3	Мінімальна тривалість сигналу на дискретному вході		0,5 мс
4	Підтримувані інтерфейси		RS-485
5	Підтримувані протоколи		Modbus RTU, Modbus ASCII, DCON
6	Напруга живлення	змінного струму	(90 ... 264) В частотою (47 ... 63) Гц
		постійного струму	(18 ... 30) В

6.2.4. Модулі дискретного виведення

Типовим “представником” обладнання для SCADA-систем, яке дозволяє здійснювати передачу дискретних даних до об’єкта автоматизації, є модуль дискретного виведення МУ110-8Р (далі – МУ110-8Р) (див. рис. 6.4).

Рис. 6.4. Зовнішній вигляд МУ110-8Р

МУ110-8Р призначений для обміну даними, отриманими в результаті здійснення керування, за допомогою інтерфейсу RS-485 з подальшим перетворенням їх в дискретну форму і виведенням у вигляді замкнених або розімкнених контактів електромагнітних реле. Основні технічні характеристики МУ110-8Р наведено в табл. 6.4.

Табл. 6.4. Основні технічні характеристики МУ110-8Р

№ з/п	Параметр	Значення
1	2	3
1	Кількість дискретних виходів	8

Продовження табл. 6.4.

1	2		3
2	Вихідний елемент		Реле електромагнітне
3	Підтримувані інтерфейси		RS-485
4	Підтримувані протоколи		Modbus RTU, Modbus ASCII, DCON
5	Напруга живлення	змінного струму	(90 ... 264) В частотою (47 ... 63) Гц
		постійного струму	(18 ... 30) В

6.2.5. Модулі для здійснення показання

Типовим "представником" обладнання для SCADA-систем, яке дозволяє здійснювати показання, є двоканальний вимірювач ТРМ200 (далі – ТРМ200) (див. рис. 6.5).

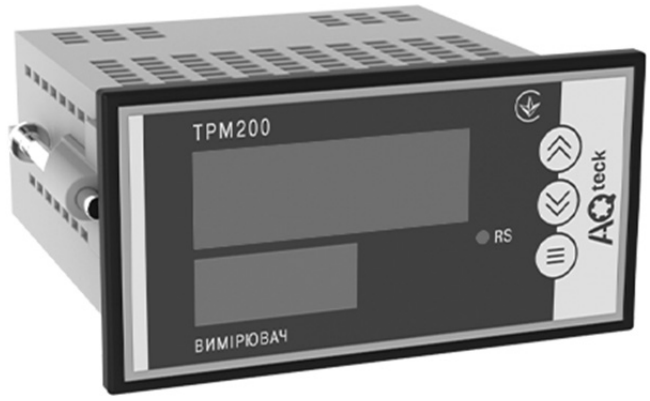


Рис. 6.5. Зовнішній вигляд ТРМ200

ТРМ200 призначений для уведення і перетворення в цифрову форму температури і інших фізичних величин (тиску, вологості, витрати, рівня і так далі), значення яких за допомогою вимірювальних перетворювачів може бути перетворено в уніфіковані сигнали напруги або струму, а також для показу температури і інших фізичних величин з подальшим обміном даними, отриманими в результаті здійснення вимірювання, за допомогою інтерфейсу RS-485. Основні технічні характеристики ТРМ200 наведено в табл. 6.5.

Табл. 6.5. Основні технічні характеристики ТРМ200

№ з/п	Параметр	Значення
1	2	3
1	Кількість аналогових входів	2
2	Межа основної приведеної похибки вимірювання	±0,5 % (для термометрів опору ±0,25 %)

Продовження табл. 6.1.

1	2	3
3	Час оновлення даних одного аналогового входу	< 1 с
4	Підтримувані інтерфейси	RS-485
5	Підтримувані протоколи	Modbus RTU, Modbus ASCII, ОВЕН
6	Напруга живлення змінного струму	(90 ... 245) В частотою (47 ... 63) Гц

6.2.6. Модулі для здійснення реєстрації

Типовим "представником" обладнання для SCADA-систем, яке дозволяє здійснювати реєстрацію, є модуль збору даних МСД-200 (далі – МСД-200) (див. рис. 6.6).

Рис. 6.6. Зовнішній вигляд МСД-200

МСД-200 призначений для збору і зберігання даних з подальшим їх обміном за допомогою інтерфейсів RS-485 і USB. Основні технічні характеристики МСД-200 наведено в табл. 6.6.

Табл. 6.6. Основні технічні характеристики МСД-200

№ з/п	Параметр	Значення
1	2	3
1	Кількість аналогових входів	4
2	Час оновлення даних одного аналогового входу	0,1 с
3	Кількість інтерфейсів RS-485	2

Продовження табл. 6.6.

1	2	3
4	Режим роботи по інтерфейсу RS-485 “ПК”	Slave
5	Протокол обміну даними по інтерфейсу RS-485 “ПК”	Modbus RTU
6	Режим роботи по інтерфейсу RS-485 “Прилади”	Master, slave, slave ext, spy
7	Протокол обміну даними по інтерфейсу RS-485 “Прилади”	Modbus RTU, Modbus ASCII, ОВЕН
8	Кількість інтерфейсів USB	1
9	Режим роботи по інтерфейсу USB	Slave
10	Напруга живлення постійного струму	(20 ... 33) В

6.2.7. Модулі для здійснення регулювання

Типовим “представником” обладнання для SCADA-систем, яке дозволяє здійснювати регулювання, є одноканальний програмний ПІД-регулятор ТРМ251 (далі – ТРМ251) (див. рис. 6.7).

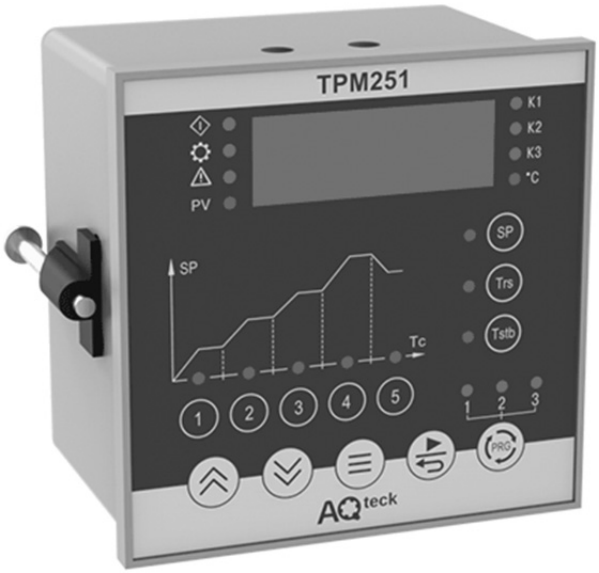


Рис. 6.7. Зовнішній вигляд ТРМ251

ТРМ251 призначений для уведення і перетворення в цифрову форму температури і інших фізичних величин (тиску, вологості, витрати, рівня і так далі), значення яких за допомогою вимірювальних перетворювачів може бути перетворено в уніфіковані сигнали напруги або струму, а також для регулювання температури і інших фізичних величин з подальшим обміном даними, отриманими в результаті здійснення вимірювання, за допомогою

інтерфейсу RS-485. Основні технічні характеристики ТРМ251 наведено в табл. 6.7.

Табл. 6.7. Основні технічні характеристики ТРМ251

№ з/п	Параметр	Значення
1	**2**	**3**
1	Кількість аналогових входів	2 (основний і резервний)
2	Межа основної приведеної похибки вимірювання	$\pm 0{,}25$ % (для термоелектричних перетворювачів $\pm 0{,}5$ %)
3	Час оновлення даних одного аналогового входу	$< 0{,}3$ с
4	Підтримувані інтерфейси	RS-485
5	Підтримувані протоколи	Modbus RTU, Modbus ASCII, ОВЕН
6	Напруга живлення змінного струму	(90 ... 245) В частотою (47 ... 63) Гц

6.2.8. Модулі для здійснення сигналізації

Типовим “представником” обладнання для SCADA-систем, яке дозволяє здійснювати сигналізацію, є індикатор СМИ2 (далі – СМИ2) (див. рис. 6.8).

Рис. 6.8. Зовнішній вигляд СМИ2

СМИ2 призначений для обміну даними за допомогою інтерфейсу RS-485. Основні технічні характеристики СМИ2 наведено в табл. 6.8.

Табл. 6.8. Основні технічні характеристики СМИ2

№ з/п	Параметр	Значення
1	**2**	**3**
1	Кількість індикаторів	1 (чотирьохрозрядний семисегментний)

Продовження табл. 6.8.

1	2	3
2	Висота розряду	14 мм
3	Підтримувані інтерфейси	RS-485
4	Підтримувані протоколи	Modbus RTU, Modbus ASCII, ОВЕН
5	Напруга живлення постійного струму	(10,5 ... 30) В

6.2.9. Програмовані промислові реле

Типовим "представником" обладнання для SCADA-систем, яке дозволяє здійснювати програмну обробку даних, є програмоване реле ПР102 (далі – ПР102) (див. рис. 6.9).

Рис. 6.9. Зовнішній вигляд ПР102

ПР102 призначене для використання в складі автоматизованих систем керування, а також інших аналогічних за призначенням систем "малого" рівня складності.

Для програмування ПР102 використовується середовище програмування AQLogic 2.5. Підтримуються (згідно з стандартом IEC 61131-3) наступні мови програмування: ST (Structured Text) і FBD (Function Block Diagram).

ПР102 має аналогові входи і виходи, дискретні входи і виходи та інтерфейси, які дозволяється за допомогою такого контролера здійснювати ефективне керування виробничими процесами. Основні технічні характеристики ПР102 наведено в табл. 6.9.

Табл. 6.9. Основні технічні характеристики ПР102

№ з/п	Параметр		Значення
1	**2**		**3**
1	Кількість аналогових входів		До 8
2	Кількість аналогових виходів		До 2
3	Кількість дискретних входів		До 24
4	Кількість дискретних виходів		До 16
5	Підтримувані інтерфейси		RS-485, USB
6	Підтримувані протоколи		Modbus RTU, Modbus ASCII
7	Напруга живлення	змінного струму	(90 ... 264) В частотою (47 ... 63) Гц
		постійного струму	(9 ... 30) В

6.2.10. Моноблокові програмовані промислові контролери (без операційної системи)

Типовим “представником” обладнання для SCADA-систем, яке дозволяє здійснювати програмну обробку даних, є програмований контролер ПЛК160 (далі – ПЛК160) (див. рис. 6.10).

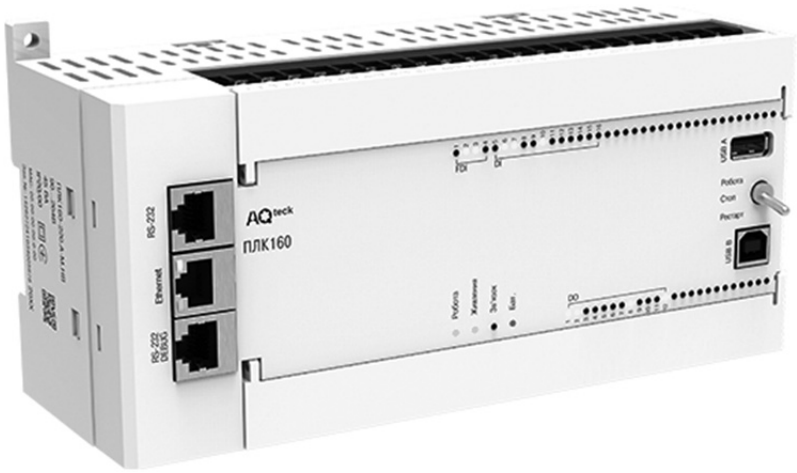


Рис. 6.10. Зовнішній вигляд ПЛК160

ПЛК160 призначений для використання в складі автоматизованих систем керування, а також інших аналогічних за призначенням систем “малого” і “середнього” рівнів складності.

Для програмування ПЛК160 використовується середовище програмування CoDeSys version 2.3. Підтримуються (згідно з стандартом IEC 61131-3) наступні мови програмування: IL (Instruction List), ST (Structured Text),

LD (Ladder Diagram), FBD (Function Block Diagram), SFC (Sequential Function Chart) і CFC (Continuous Function Chart).

ПЛК160 має аналогові входи і виходи, дискретні входи і виходи та інтерфейси, які дозволяється за допомогою такого контролера здійснювати ефективне керування виробничими процесами. Основні технічні характеристики ПЛК160 наведено в табл. 6.10.

Табл. 6.10. Основні технічні характеристики ПЛК160

№ з/п	Параметр		Значення
1	**2**		**3**
1	Кількість аналогових входів		8
2	Кількість аналогових виходів		4
3	Кількість дискретних входів		16
4	Кількість дискретних виходів		12
5	Підтримувані інтерфейси		RS-232, RS-485, USB, Ethernet
6	Підтримувані протоколи		Modbus RTU, Modbus ASCII, Modbus TCP/IP, Modbus UDP/IP, DCON
7	Напруга живлення	змінного струму	(90 ... 264) В частотою (47 ... 63) Гц
		постійного струму	(9 ... 26) В

6.2.11. Моноблокові програмовані промислові контролери (з операційною системою)

Типовим “представником” обладнання для SCADA-систем, яке дозволяє здійснювати програмну обробку даних, є програмований контролер ПЛК200 (див. рис. 6.11).

ПЛК200 призначений для використання в складі автоматизованих систем керування, а також інших аналогічних за призначенням систем “малого” і “середнього” рівнів складності.

Для програмування ПЛК200 використовується середовище програмування CoDeSys version 3.5. Підтримуються (згідно з стандартом IEC 61131-3) наступні мови програмування: ST (Structured Text), LD (Ladder Diagram), FBD (Function Block Diagram), SFC (Sequential Function Chart) і CFC (Continuous Function Chart).

Рис. 6.11. Зовнішній вигляд ПЛК200

ПЛК200 має аналогові входи, дискретні входи і виходи, швидкі дискретні входи і виходи та інтерфейси, які дозволяється за допомогою такого контролера здійснювати ефективне керування виробничими процесами. Основні технічні характеристики ПЛК200 наведено в табл. 6.11.

Табл. 6.11. Основні технічні характеристики ПЛК200

№ з/п	Параметр	Значення
1	**2**	**3**
1	Кількість аналогових входів	До 4
2	Кількість дискретних входів	До 12
3	Кількість дискретних виходів	До 14
4	Підтримувані інтерфейси	RS-485, USB, Ethernet
5	Підтримувані протоколи	Modbus RTU, Modbus ASCII, ОВЕН, тепло- і електролічильників, NTP, FTP, SSH, HTTP, HTTPS, Modbus TCP/IP, OPC UA, MQTT, SNMP
6	Напруга живлення постійного струму	(10 ... 48) В

6.2.12. Сенсорні панельні програмовані промислові контролери

Типовим “представником” обладнання для SCADA-систем, яке дозволяє здійснювати програмну обробку даних, є панель оператора програмована СПК110 (далі – СПК110) (див. рис. 6.12).

Рис. 6.12. Зовнішній вигляд СПК110

СПК110 призначена для використання в складі автоматизованих систем керування, а також інших аналогічних за призначенням систем “малого” і “середнього” рівнів складності.

Для програмування СПК110 використовується середовище програмування CoDeSys version 3.5. Підтримуються (згідно з стандартом IEC 61131-3) наступні мови програмування: ST (Structured Text), LD (Ladder Diagram), FBD (Function Block Diagram), SFC (Sequential Function Chart) і CFC (Continuous Function Chart).

СПК110 має інтерфейси, які дозволяється за допомогою такої панелі здійснювати ефективне керування виробничими процесами. Основні технічні характеристики СПК110 наведено в табл. 6.12.

Табл. 6.12. Основні технічні характеристики СПК110

№ з/п	Параметр	Значення
1	**2**	**3**
1	Тип дисплея	TFT LCD з діагоналлю 10,2"
2	Розміри дисплея	222,0 мм × 132,5 мм
3	Роздільна здатність дисплея	800 пікселів × 480 пікселів
4	Підтримувані інтерфейси	RS-232, RS-485, USB, Ethernet
5	Підтримувані протоколи	Modbus RTU, Modbus ASCII, ОВЕН, Modbus TCP/IP
6	Напруга живлення постійного струму	(12 ... 28) В

6.2.13. Убудовані промислові комп’ютери

Типовим “представником” обладнання для SCADA-систем, яке дозволяє здійснювати програмну обробку даних, є модульний захищений убудований промисловий комп’ютер Cincoze DS-1301 (далі – Cincoze DS-1301) (див. рис. 6.13).

Рис. 6.13. Зовнішній вигляд Cincoze DS-1301

Cincoze DS-1301 призначений для використання в складі автоматизованих систем керування, а також інших аналогічних за призначенням систем “середнього” і “великого” рівнів складності. Основні технічні характеристики Cincoze DS-1301 наведено в табл. 6.13.

Табл. 6.13. Основні технічні характеристики Cincoze DS-1301

№ з/п	Параметр	Значення
1	**2**	**3**
1	Процесор	Intel Celeron/Pentium/Core/Xeon
2	Пам’ять	До 64 Гбайт DDR4 SO-DIMM
3	Накопичувачі	SSD M.2, SSD/HDD SATA, mSATA, RAID 0/1/5/10
4	Підтримувані графічні підключення	HDMI, DisplayPort, VGA (одночасно до 3 промислових моніторів)
5	Основні підтримувані інтерфейси	RS-232/422/485, USB, Ethernet
6	Додаткові підтримувані інтерфейси	SIM Socket, Mini PCI Express (MEC-модулі), Combined Multiple I/O (CMI-модулі), Control Function Module (CFM-модулі)
7	Операційна система	Windows 10
8	Напруга живлення постійного струму	(9 ... 48) В

За рахунок наявності у Cincoze DS-1301 кількох додаткових інтерфейсів Mini PCI Express, Combined Multiple I/O і Control Function Module його можливості з обміну можуть бути суттєво розширені. Застосування, наприклад, MEC-модуля MEC-LAN-M102 дозволяє додати до Cincoze DS-1301 2-ва додаткові інтерфейси Ethernet з швидкістю обміну даними 10/100/1000 Мбіт/с (див. рис. 6.14).

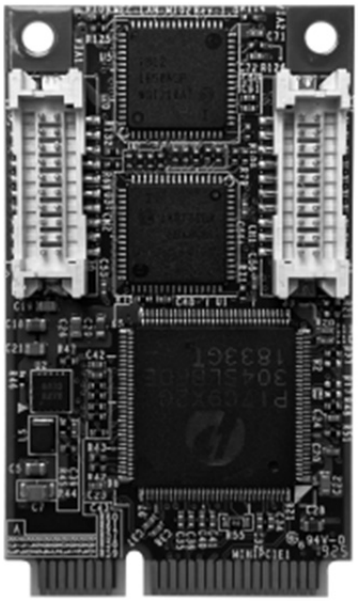


Рис. 6.14. Зовнішній вигляд MEC-LAN-M102

А застосування, наприклад, CMI-модуля CMI-COM02 (див. рис. 6.15) дозволяє додати до Cincoze DS-1301 4-ри додаткові інтерфейси RS-232/422/485.

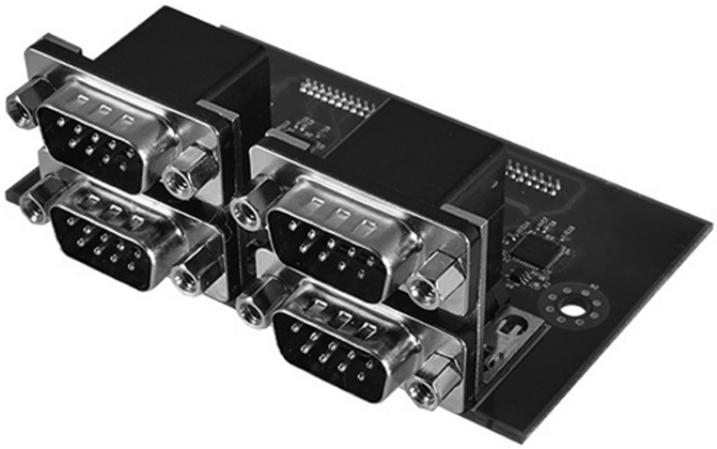


Рис. 6.15. Зовнішній вигляд CMI-COM02

Крім того, застосування, наприклад, CFM-модуля CFM-PoE03 (див. рис. 6.16) дозволяє додати до Cincoze DS-1301 можливість живлення через Ethernet

віддалених пристроїв (вимірювальних перетворювачів, датчиків і сенсорів, точок доступу до мережі Wi-Fi і так далі).

Рис. 6.16. Зовнішній вигляд CFM-PoE03

6.2.14. Сенсорні панельні промислові комп’ютери

Типовим “представником” обладнання для SCADA-систем, яке дозволяє здійснювати програмну обробку даних, є безвентиляторний мультисенсорний панельний промисловий комп’ютер Vecow MTC-7021W (далі – Vecow MTC-7021W) (див. рис. 6.17).

Рис. 6.17. Зовнішній вигляд Vecow MTC-7021W

Vecow MTC-7021W призначений для використання в складі автоматизованих систем керування, а також інших аналогічних за призначенням систем “середнього” і “великого” рівнів складності. Основні технічні характеристики Vecow MTC-7021W наведено в табл. 6.14.

Табл. 6.14. Основні технічні характеристики Vecow MTC-7021W

№ з/п	Параметр	Значення
1	**2**	**3**
1	Процесор	Intel Core
2	Пам’ять	До 32 Гбайт DDR4 SO-DIMM
3	Накопичувачі	SSD/HDD SATA, mSATA
4	Підтримувані графічні підключення	DVI-D, DisplayPort
5	Основні підтримувані інтерфейси	RS-232/422/485, USB, Ethernet
6	Додаткові підтримувані інтерфейси	Mini PCI Express (MEC-модулі)
7	Операційна система	Windows 10, Linux
8	Напруга живлення постійного струму	(9 ... 48) В

6.2.15. Високопродуктивні промислові комп’ютери

Типовим “представником” обладнання для SCADA-систем, яке дозволяє здійснювати програмну обробку даних, є компактний безвентиляторний високопродуктивний промисловий комп’ютер Advantech MIC-770 V3 (далі – MIC-770 V3) (див. рис. 6.18).

Рис. 6.18. Зовнішній вигляд Advantech MIC-770 V3

Advantech MIC-770 V3 призначений для використання в складі автоматизованих систем керування, а також інших аналогічних за призначенням систем “великого” рівнів складності. Основні технічні характеристики Advantech MIC-770 V3 наведено в табл. 6.15.

Табл. 6.15. Основні технічні характеристики Advantech MIC-770 V3

№ з/п	Параметр	Значення
1	**2**	**3**
1	Процесор	Intel Core
2	Пам’ять	До 64 Гбайт DDR5 SO-DIMM ECC
3	Накопичувачі	SSD M.2, SSD/HDD SATA, mSATA, RAID 0/1/5/10
4	Підтримувані графічні підключення	HDMI, VGA (одночасно до 4 промислових моніторів)
5	Основні підтримувані інтерфейси	RS-232/422/485, USB, Ethernet
6	Додаткові підтримувані інтерфейси	Mini PCI Express (MEC-модулі)
7	Операційна система	Windows 10, Linux
8	Напруга живлення постійного струму	(9 ... 36) В

6.2.16. Промислові монітори

Типовим “представником” обладнання для SCADA-систем, яке дозволяє здійснювати показ даних, є мультисенсорний промисловий монітор Vecow MTD-6017 (далі – Vecow MTD-6017) (див. рис. 6.19).

Рис. 6.19. Зовнішній вигляд Vecow MTD-6017

Vecow MTD-6017 призначений для використання в складі автоматизованих систем керування, а також інших аналогічних за

призначенням систем “малого”, “середнього” і “великого” рівнів складності. Основні технічні характеристики Vecow MTD-6017 наведено в табл. 6.16.

Табл. 6.16. Основні технічні характеристики Vecow MTD-6017

№ з/п	Параметр	Значення
1	**2**	**3**
1	Тип дисплея	TFT LCD з діагоналлю 17,0"
2	Співвідношення сторін дисплея	4: 3
3	Роздільна здатність дисплея	1280 пікселів × 1024 пікселів
4	Яскравість, не гірше	250 кд/м^2
5	Контраст, не гірше	1000: 1
6	Підтримувані графічні підключення	DVI-I, VGA
7	Напруга живлення постійного струму	12 В

6.3. Вибір програмного забезпечення: загальні рекомендації

Переважна більшість SCADA-систем може працювати з операційним системами загального призначення (не реального часу) – але в такому випадку абсолютно не гарантується взаємодія з об’єктом автоматизації в реальному часі. Тобто оперативний збір даних про виробничі (технологічні), налагоджувальні і інші параметри об’єкта автоматизації, а також оперативне керування ними в будь-який момент часу може бути перервано.

Найкращий підхід полягає у використанні операційних систем “жорсткого” реального часу – наприклад, таких як QNX Neutrino. Але для них потрібно, відповідно, і “власне” програмне забезпечення, якого існує не так багато.

Інший підхід полягає у використанні операційних систем “м’якого” реального часу – наприклад, переважна більшість SCADA-систем може також працювати з операційними системами сімейства Windows IoT. З одного боку, забезпечується “м’якість” роботи SCADA-систем, а з іншого боку – підтримуються практично всі можливості SCADA-систем.

Також забезпечити роботу SCADA-систем в “жорсткому” реальному часі (за виробничої необхідності і в “м’якому” реальному часі) можна, використовуючи програмовані промислові контролери з операційною системою реального часу (наприклад, такі, як ПЛК200 (див. рис. 6.11) – вони мають убудовану операційну систему Linux з RT-патчем).

Загалом вибір програмного забезпечення SCADA-систем є непростою

“задачею”, і в залежності від складності об’єкта автоматизації рекомендується використовувати наступні підходи:

- при невеликій складності об’єкта автоматизації – створювати SCADA-системи самостійно шляхом використання відповідних середовищ розробки програмного забезпечення або використовувати такі SCADA-системи, як, наприклад, Smart ReView version 1.0 і Visual Intellect version 2.0 розробки ТОВ “МІКРОЛ” (Україна, м. Івано-Франківськ) або візуалізації CoDeSys version 2.3 і CoDeSys version 3.5;

- при великій складності об’єкта автоматизації – обирати такі SCADA-системи, як, наприклад, SIMATIC WinCC version 7.4 (і більш нових версій).

6.4. Питання для самоперевірки

1. В чому полягають основні принципи вибору обладнання для SCADA-систем?
2. Яке апаратне забезпечення може бути використане для приймання аналогових даних в SCADA-системах?
3. Яке апаратне забезпечення може бути використане передачі аналогових даних в SCADA-системах?
4. Яке апаратне забезпечення може бути використане для приймання дискретних даних в SCADA-системах?
5. Яке апаратне забезпечення може бути використане для передачі дискретних даних в SCADA-системах?
6. Яке апаратне забезпечення може бути використане для здійснення показання в SCADA-системах?
7. Яке апаратне забезпечення може бути використане для здійснення реєстрації в SCADA-системах?
8. Яке апаратне забезпечення може бути використане для здійснення регулювання в SCADA-системах?
9. Яке апаратне забезпечення може бути використане для здійснення сигналізації в SCADA-системах?
10. Що таке програмовані промислові реле, як вони можуть використовуватися в SCADA-системах?
11. Що таке програмовані промислові контролери (без операційної системи), як вони можуть використовуватися в SCADA-системах?
12. Що таке програмовані промислові контролери (з операційною системою), як вони можуть використовуватися в SCADA-системах?

13. Що таке убудовані промислові комп’ютери, як вони можуть використовуватися в SCADA-системах?

14. Що таке сенсорні панельні промислові комп’ютери, як вони можуть використовуватися в SCADA-системах?

15. Що таке високопродуктивні промислові комп’ютери, як вони можуть використовуватися в SCADA-системах?

16. Що таке промислові монітори, як вони можуть використовуватися в SCADA-системах?

17. Яке програмне забезпечення може бути використане в SCADA-системах?

6.5. Рекомендована література

Базова література: [4–6].

Додаткова література: [2, 3, 5].

7. SCADA-СИСТЕМА НА ОСНОВІ ПАНЕЛІ ОПЕРАТОРА ИП320

7.1. Основні технічні характеристики панелі оператора ИП320

Панель оператора ИП320 (далі – ИП320) призначена для обміну даними з програмованими контролерами і реле, модулями уведення/виведення, регуляторами і так далі. Використовуючи інтерфейс RS-485 і протокол Modbus RTU в режимах “ведучого” (master) або “веденого” (slave), ИП320 забезпечує людино-машинний інтерфейс шляхом використання кількох елементів мнемосхем, які є “убудованими” в ИП320 (безпосередньо програмування ИП320 здійснюється з використанням інтерфейсу RS-232).

Основні електричні характеристики ИП320 наведено в табл. 7.1.

Табл. 7.1. Основні електричні характеристики ИП320

№ з/п	Характеристика	Значення
1	**2**	**3**
1	Напруга живлення постійного струму	Від 20 до 28 В (номінальне значення 24 В)
2	Споживана потужність	Не більше 4 Вт
3	Інтерфейси, які підтримуються	RS-232 і RS-485
4	Швидкості роботи інтерфейсів, які підтримуються	2400, 4800, 9600, 19200, 38400 і 115200 біт/с
5	Протоколи, які підтримуються	Modbus RTU

Основні механічні характеристики ИП320 наведено в табл. 7.2.

Табл. 7.2. Основні механічні характеристики ИП320

№ з/п	Характеристика	Значення
1	**2**	**3**
1	Конструктивне виконання корпусу	Кріплення на щиті
2	Ступінь захисту корпусу зі сторони лицевої панелі	IP54
3	Тип дисплея	Графічний монохромний рідиннокристалічний з підсвічуванням з діагоналлю 3,7"

Продовження 7.2.

1	2	3
4	Розміри дисплея	100 мм × 35 мм
5	Роздільна здатність дисплея	192 пікселя × 64 пікселя
6	Кількість кнопок	20
7	Розміри	(172 ± 1) мм × (94 ± 1) мм × (30 ± 1) мм
8	Маса	Не більше 0,5 кг

Схема електрична підключення до ИП320 живлення та інтерфейсів RS-232 і RS-485 наведена на рис. 7.1.

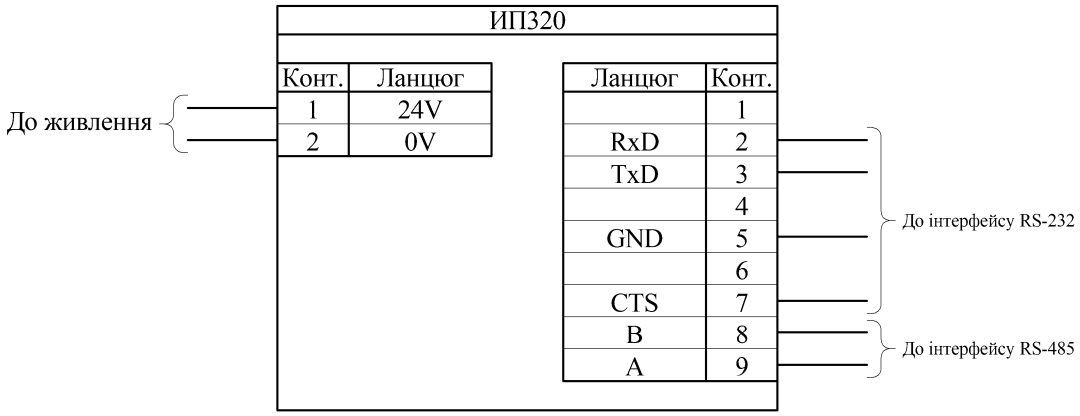


Рис. 7.1. Схема електрична підключення до ИП320 живлення та інтерфейсів RS-232 і RS-485

Інтерфейс RS-232 призначений для програмування ИП320 та для обміну даними з технічними засобами автоматизації, які підтримують такий самий інтерфейс, а також протокол Modbus RTU. В свою чергу, інтерфейс RS-485 призначений для обміну даними з технічними засобами автоматизації, які підтримують такий самий інтерфейс, а також протокол Modbus RTU.

Зовнішній вигляд передньої панелі ИП320 наведено на рис. 7.2. На такій пане розташовані дисплей, 8 керуючих кнопок і 12 цифрових/функціональних кнопок. Особливістю ИП320 є використання годинника реального часу, який живиться від убудованої літієвої батарейки типу CR2032, що забезпечує неперервну роботу такого годинника приблизно від двох до трьох років.

ИП320 є одним з технічних засобів автоматизації, які часто використовуються як “доповнення” можливостей SCADA-систем, що представляють собою розміщені на персональних/промислових комп’ютерах, серверах, робочих станціях і так далі пакети програм.

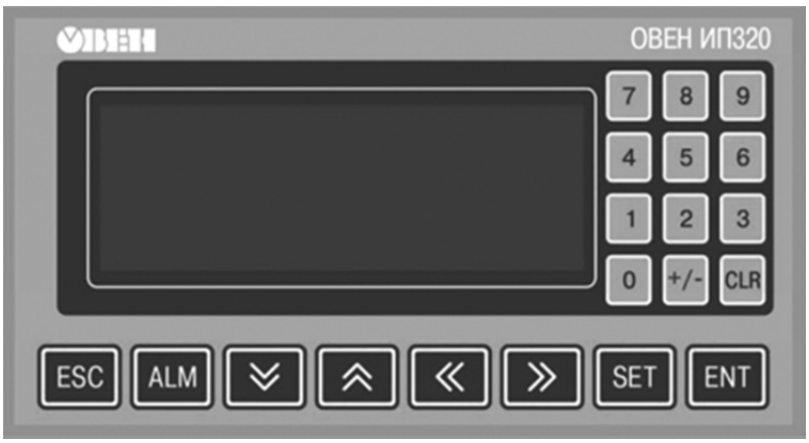


Рис. 7.2. Зовнішній вигляд передньої панелі ИП320

7.2. Програмне забезпечення панелі оператора ИП320

7.2.1. Загальні відомості про програмне забезпечення панелі оператора ИП320

Програмне забезпечення для програмування ИП320 представлене програмою “Конфігуратор ИП320”. Інтерфейс такої програми утримує наступні складові:

- меню;
- панель інструментів;
- рядок “стану”;
- область у вигляді таблиці з двома кнопками, за допомогою яких можна створювати і видаляти “екрани”;
- область, де розміщуються шаблони елементів “екранів”;
- область, де розміщується поточний “екран”;
- область, де розміщуються атрибути елементів поточного “екрану”.

До шаблонів елементів “екранів” відносяться наступні шаблони:

- **Текст** – дозволяє виводити статичні дані;
- **Графічний текст** – дозволяє виводити статичні дані;
- **Динамічний текст** – дозволяє виводити динамічні дані;
- **Динамічний графічний текст** – дозволяє виводити динамічні дані;
- **Індикатор** – дозволяє виводити динамічні дані;
- **Малюнок** – дозволяє виводити статичні дані;
- **Регістр** – дозволяє як уводити динамічні дані, так і виводити динамічні дані;
- **Лінійка** – дозволяє виводити динамічні дані;

- **Функціональна кнопка** – дозволяє як уводити динамічні дані, так і здійснювати певні дії;

- **Графік** – дозволяє виводити динамічні дані;

- **Регістри ASCII-тексту** – дозволяє виводити динамічні дані.

7.2.2. Шаблон елементів “екранів” Текст

Шаблон елементів “екранів” **Текст** наведено на рис. 7.3. Такий шаблон призначений для відображення найменувань виробничих (технологічних) операцій, ділянок, ліній або процесів та будь-якої інформації, яка є необхідною.

Рис. 7.3. Шаблон елементів “екранів” **Текст**

Особливістю шаблона елементів “екранів” **Текст** є те, що властивості більшості з його атрибутів (шрифту, написання, розміру і деяких інших), якими будуть виводитися відповідні найменування, є незмінними. Якщо більшість таких найменувань будуть виглядати в одному “стилі”, то інформація з “екранів” ИП320 буде легше сприйматися (див. рис. 7.4).

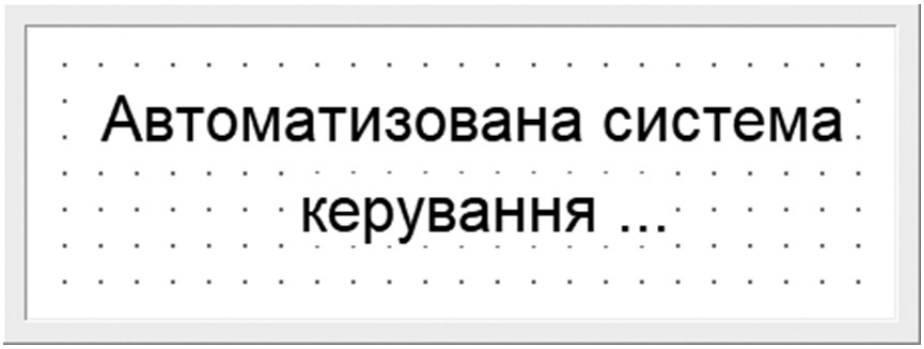


Рис. 7.4. Приклад використання шаблона елементів “екранів” **Текст**

7.2.3. Шаблон елементів “екранів” Графічний текст

Шаблон елементів “екранів” **Графічний текст** наведено на рис. 7.5. Такий шаблон також призначений для відображення найменувань виробничих (технологічних) операцій, ділянок, ліній або процесів та будь-якої інформації, яка є необхідною.

Рис. 7.5. Шаблон елементів “екранів” **Графічний текст**

Особливістю шаблона елементів “екранів” **Графічний текст** є те, що властивості більшості з його атрибутів, якими будуть виводитися відповідні найменування, можна змінювати – зокрема, можна поміняти шрифт, написання, розмір, видозміну і набір символів. Це, в свою чергу, дає можливість виділити на “екранах” ИП320 певну інформацію (див. рис. 7.6).

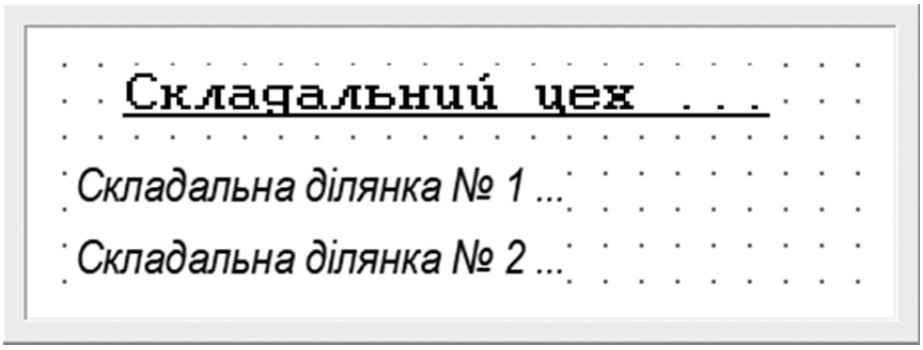


Рис. 7.6. Приклад використання шаблона елементів “екранів” **Графічний текст**

7.2.4. Шаблон елементів “екранів” Динамічний текст

Шаблон елементів “екранів” **Динамічний текст** наведено на рис. 7.7. Він має таке ж саме призначення, як і шаблон елементів “екранів” **Текст**.

Рис. 7.7. Шаблон елементів “екранів” **Динамічний текст**

Але в процесі роботи ИП320 його уміст можна змінювати, обираючи з кількох наперед заданих варіантів. Це дає можливість відображувати відповідні найменування саме в той момент часу, коли потрібно. В той момент часу, коли конвеєр працює, може бути відображене найменування “Конвеєр № ... працює” (див. рис. 7.8), а коли не працює –найменування “Конвеєр № ... не працює”.

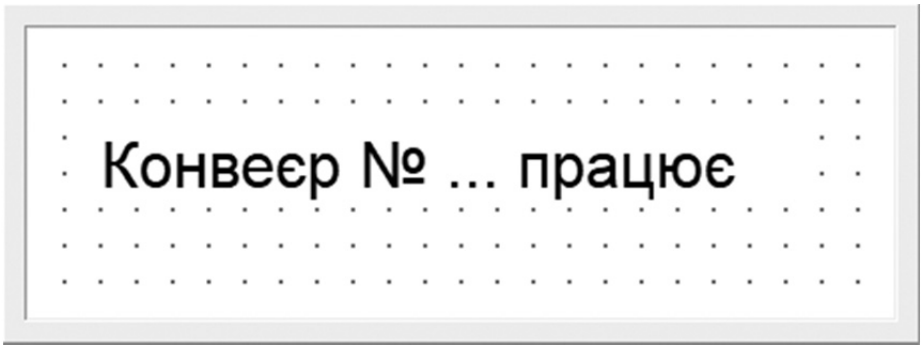


Рис. 7.8. Приклад використання шаблона елементів “екранів” **Динамічний текст**

7.2.5. Шаблон елементів “екранів” Динамічний графічний текст

Шаблон елементів “екранів” **Динамічний графічний текст** наведено на рис. 7.9. Він має таке ж саме призначення, як і шаблон елементів “екранів” **Графічний текст**.

Рис. 7.9. Шаблон елементів “екранів” **Динамічний графічний текст**

В процесі роботи ИП320 його уміст також можна змінювати, обираючи з кількох наперед заданих варіантів, і відображувати відповідні найменування саме в той момент часу, коли потрібно. В той момент часу, коли вентиль відкритий, може бути відображене найменування “Вентиль № ... відкритий” (див. рис. 7.10), а коли закритий – найменування “Вентиль № ... закритий”.

Рис. 7.10. Приклад використання шаблона елементів “екранів” **Динамічний графічний текст**

7.2.6. Шаблон елементів “екранів” Індикатор

Шаблон елементів “екранів” **Індикатор** наведено на рис. 7.11. Такий шаблон призначений для відображення одного з двох можливих станів виробничих (технологічних) операцій, ділянок, ліній або процесів.

Рис. 7.11. Шаблон елементів “екранів” **Індикатор**

Стан непрацюючих печей може відображатися так, як наведено на рис. 7.12. З плином часу кожна з чотирьох печей може як переходити з непрацюючого в працюючий стан, так і з працюючого в непрацюючий – відповідно до зміну стану таких печей буде змінюватися і стан чотирьох елементів “екранів” **Індикатор**.

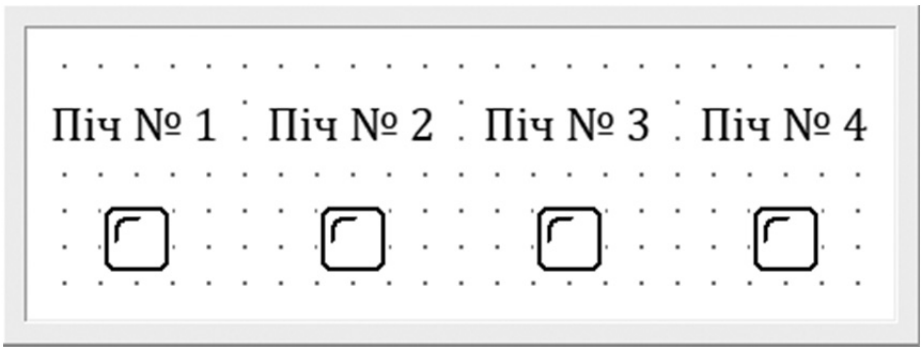


Рис. 7.12. Приклад використання шаблона елементів “екранів” **Індикатор**

7.2.7. Шаблон елементів “екранів” Малюнок

Шаблон елементів “екранів” **Малюнок** наведено на рис. 7.13. Такий шаблон призначений для відображення структури автоматизованих систем керування, частин схем і креслень, умовних графічних позначень і так далі.

Рис. 7.13. Шаблон елементів “екранів” **Малюнок**

Відображення структури контуру керування, як наведено на рис. 7.14, може полегшити розуміння принципів його роботи і спростити налагодження його параметрів.

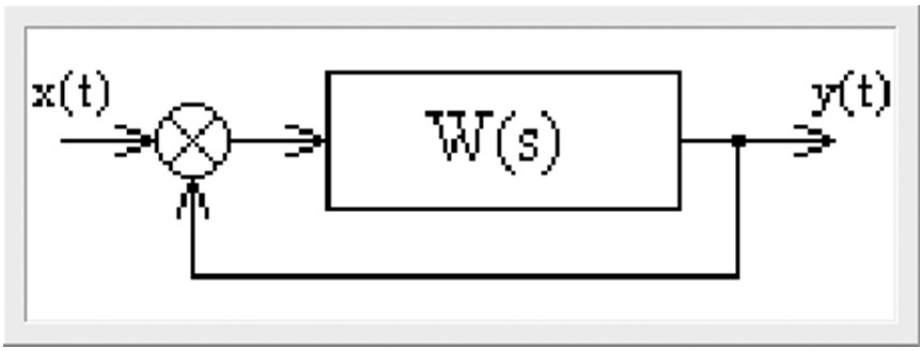


Рис. 7.14. Приклад використання шаблона елементів “екранів” **Малюнок**

7.2.8. Шаблон елементів “екранів” Регістр

Шаблон елементів “екранів” **Регістр** наведено на рис. 7.15. Такий шаблон є одним з найважливіших шаблонів елементів “екранів” ИП320, тому що він призначений як для уведення умісту регістрів одиниць обладнання промислової автоматизації, так і для їх виведення.

Рис. 7.15. Шаблон елементів “екранів” **Регістр**

Якщо прапор **Настр.** скинутий, то шаблон елементів “екранів” **Регістр** дозволяє виводити уміст регістрів одиниці обладнання промислової автоматизації (тобто дані передаються від такого обладнання до ИП320).

Якщо при скинутому прапорі **Настр.** встановити прапор **Конв.**, то можна виконати перетворення даних з одного діапазону в інший діапазон. Наприклад, початковому діапазону зміни даних (струму I) від 4 до 20 мА може відповідати перетворений діапазон зміни даних (тиску p) від 0 до 25 МПа. Для збільшення роздільної здатності (тобто числових значень, які будуть безпосередньо передаватися через відповідні регістри одиниці обладнання промислової автоматизації), зручно представити, що початковому діапазону зміни даних (струму I) від 4000 до 20000 мкА може відповідати перетворений діапазон

зміни даних (тиску p) від 0 до 25000 кПа. Тоді залежність $p(I)$ буде мати наступний вигляд:

$$p(I) = 1{,}5625I - 6250\,.$$

Отже, наприклад, струму $I = 10000$ мкА буде відповідати тиск $p = 9375$ кПа. Задавши, крім цього, для лічильника **Розряди** (загальна кількість знакових розрядів) значення 5, а для лічильника **Дробові** (кількість знакових розрядів після десяткової точки) значення 3, тиск p можна буде представити у форматі 12.345, де до десяткової точки є 2 розряди і після неї є 3 розряди, а замість цифр 1, 2, 3, 4 і 5 буде підставлятися поточне округлене значення тиску (див. рис. 7.16). Причому, так як десяткова точка в значенні тиску зміщена на 3 розряди вліво, то таке значення можна зразу відображати не в кПа, а в МПа (виходячи з того, що 1000 кПа = 1 МПа).

Рис. 7.16. Приклад використання шаблона елементів “екранів” **Регістр**

Якщо прапор **Настр.** встановлений, то шаблон елементів “екранів” **Регістр** дозволяє уводити уміст регістрів одиниці обладнання промислової автоматизації (тобто дані передаються від ИП320 до такого обладнання). Якщо при встановленому прапорі **Настр.** встановити прапор **Межа**, то можна обмежити уміст регістрів одиниці обладнання промислової автоматизації як мінімальним, так і максимальним значеннями. А якщо при встановленому прапорі **Настр.** встановити прапор **Пароль**, то для доступу до умісту таких регістрів треба буде мати пароль.

7.2.9. Шаблон елементів “екранів” Лінійка

Шаблон елементів “екранів” **Лінійка** наведено на рис. 7.17. Такий шаблон призначений для приблизного (якісного, але не кількісного) відображення умісту регістрів одиниць обладнання промислової автоматизації.

Рис. 7.17. Шаблон елементів “екранів” **Лінійка**

Рівень рідини в баку може відображатися так, як наведено на рис. 7.18. З плином часу такий рівень може змінюватися – відповідно до цього, буде змінюватися і ступінь “зачорнення” елементу “екранів” **Лінійка**.

Рис. 7.18. Приклад використання шаблона елементів “екранів” **Лінійка**

7.2.10. Шаблон елементів “екранів” Функціональна кнопка

Шаблон елементів “екранів” **Функціональна кнопка** наведено на рис. 7.19. Такий шаблон є важливим шаблоном елементів “екранів” ИП320, тому що він дозволяє здійснювати переходи між “екранами” (“екраном” для роботи з паролем, “екраном” для роботи з тривогами, “екраном” для роботи з датою/часом і “екранами”, які визначаються користувачами) та уводити уміст “котушок” (дані передаються по 1 біту) і регістрів (дані передаються по 16 бітів) одиниці обладнання промислової автоматизації (тобто дані передаються від ИП320 до такого обладнання).

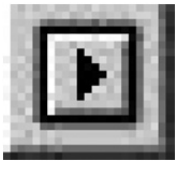


Рис. 7.19. Шаблон елементів “екранів” **Функціональна кнопка**

Пуск лінії можна здійснювати, натискаючи керуючу кнопку **SET**, а зупинку – натискаючи функціональну кнопку **CLR** (див. рис. 7.20).

Рис. 7.20. Приклад використання шаблона елементів “екранів” **Функціональна кнопка**

7.2.11. Шаблон елементів “екранів” Графік

Шаблон елементів “екранів” **Графік** наведено на рис. 7.21. Такий шаблон призначений для достатньо точного (кількісного, але не якісного) відображення умісту регістрів одиниць обладнання промислової автоматизації (на одному елементі “екранів” **Графік** відображатися може тільки один графік).

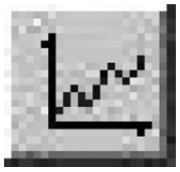


Рис. 7.21. Шаблон елементів “екранів” **Графік**

Температура в приміщенні може відображатися так, як наведено на рис. 7.22. Змінюючи за допомогою лічильника **Число точок** число точок на графіку, а за допомогою лічильника **Інтервал** інтервал часу між точками на графіку, можна змінювати відповідно ступінь деталізації такого графіка і його часовий масштаб.

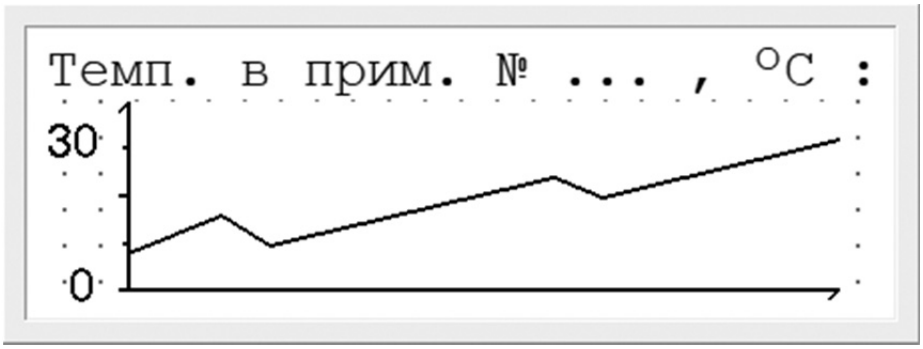


Рис. 7.22. Приклад використання шаблона елементів “екранів” **Графік**

7.2.12. Шаблон елементів “екранів” Регістри ASCII-тексту

Шаблон елементів “екранів” **Регістри ASCII-тексту** наведено на рис. 7.23. Такий шаблон призначений для відображення умісту регістрів одиниць обладнання промислової автоматизації у вигляді символьних рядків (такі рядки можуть утримувати до 16 символів, кожен з яких є буквою англійського алфавіту, арабською цифрою або спеціальним символом.

Рис. 7.23. Шаблон елементів “екранів” **Регістри ASCII-тексту**

Зарезервувавши за допомогою лічильника **Символів** 15 “знакомісць” (див. рис. 7.24), можна буде на “екрані” ИП320 відобразити символьний рядок “Open the valve!” (він утримує 15 символів).

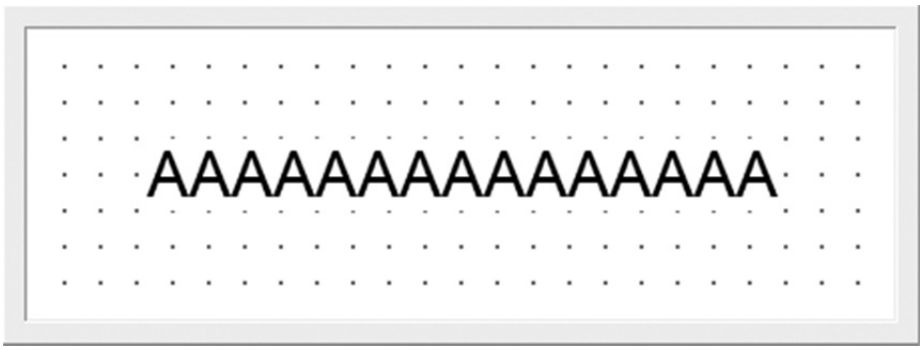


Рис. 7.24. Приклад використання шаблона елементів “екранів” **Регістри ASCII-тексту**

7.3. Приклад реалізації SCADA-системи

Схема автоматизації системи керування штучним освітленням наведена на рис. 7.25. Основною складовою такої системи є програмоване реле.

В якості об’єкта керування системи керування штучним освітленням виступає виробниче приміщення, в якому необхідно або умикати і вимикати штучне освітлення вручну (“ручний” режим роботи системи керування), або умикати штучне освітлення тоді, коли освітленість зовні виробничого приміщення стає менше заданого значення і навпаки (“автоматичний” режим роботи системи керування).

За допомогою датчика освітленості 1.1 виконується вимірювання

освітленості зовні виробничого приміщення, за допомогою нормуючого перетворювача 1.2 сигнал від чутливого елемента такого датчика перетворюється в уніфікований електричний неперервний сигнал, за допомогою регулятора 1.3 виконується регулювання освітленості шляхом використання лампи 1.4. Загальне керування штучним освітленням виконується за допомогою пульта 2 (тобто штучним освітленням можна керувати як “зі сторони” такого пульта, так і “зі сторони” SCADA-системи на основі ИП320 – це залежить від поточної виробничої необхідності).

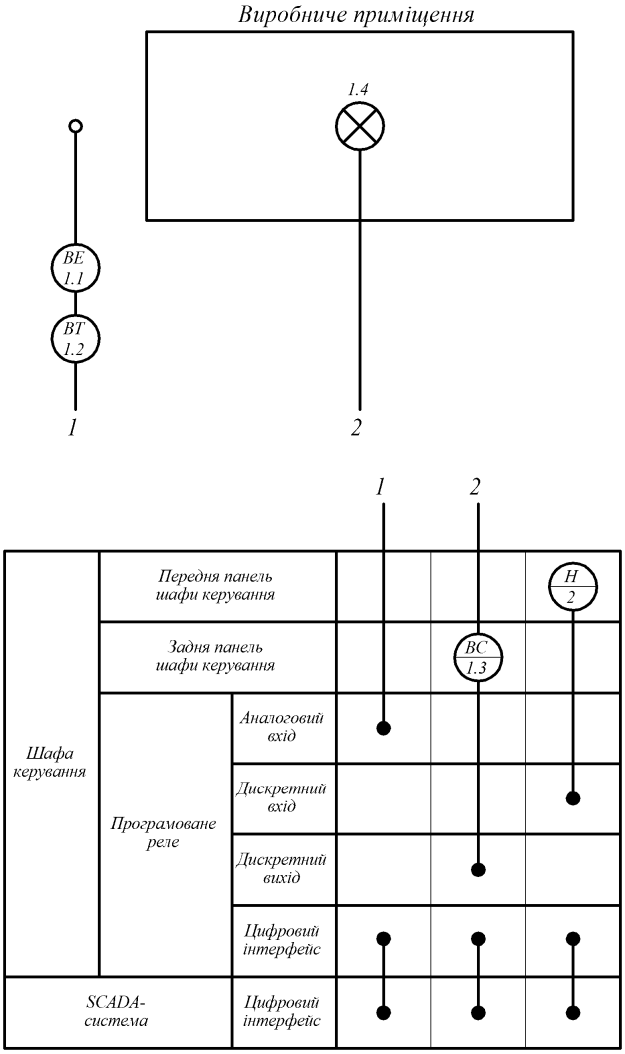


Рис. 7.25. Схема автоматизації системи керування штучним освітленням

Мнемосхема SCADA-системи для системи керування штучним освітленням може складатися з 4-х “екранів”:

- “Екран” № 1 – може утримувати елементи “екранів”, які відображають найменування SCADA-системи (див. рис. 7.26);

- “Екран” № 2 – може утримувати елементи “екранів”, які дозволяють

обирати “ручний” або “автоматичний” режими роботи системи керування штучним освітленням та обирати умикання або вимикання штучного освітлення вручну (див. рис. 7.27);

- “Екран” № 3 – може утримувати елементи “екранів”, які дозволяють обирати задане значення освітленості зовні виробничого приміщення та відображають поточне значення освітленості зовні виробничого приміщення (див. рис. 7.28);

- “Екран” № 4 – може утримувати елементи “екранів”, які відображають поточний стан лампи (лампа увімкнена або вимкнена) (див. рис. 7.29).

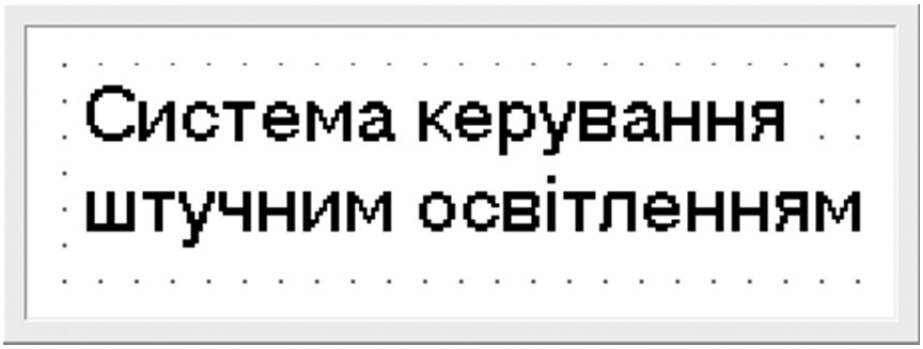


Рис. 7.26. “Екран” № 1

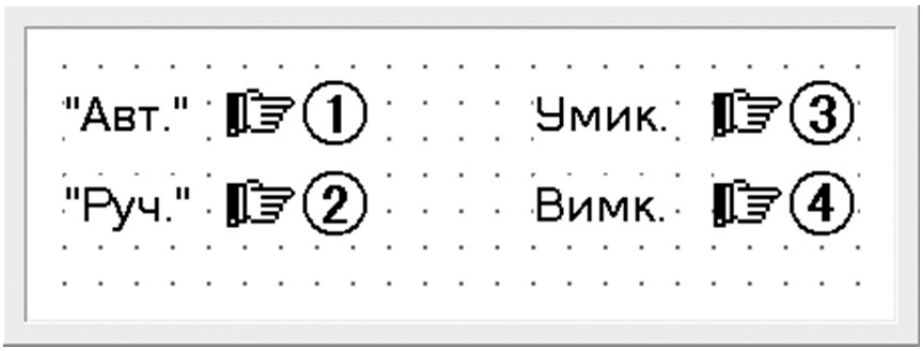


Рис. 7.27. “Екран” № 2

Рис. 7.28. “Екран” № 3

Рис. 7.29. “Екран” № 4

7.4. Питання для самоперевірки

1. Що таке панель оператора ИП320?
2. Як підключити до панелі оператора ИП320 живлення та та інтерфейси RS-232 і RS-485?
3. Що таке шаблони елементів “екранів”?
4. Що таке шаблон елементів “екранів” **Текст**?
5. Що таке шаблон елементів “екранів” **Графічний текст**?
6. Що таке шаблон елементів “екранів” **Динамічний текст**?
7. Що таке шаблон елементів “екранів” **Динамічний графічний текст**?
8. Що таке шаблон елементів “екранів” **Індикатор**?
9. Що таке шаблон елементів “екранів” **Малюнок**?
10. Що таке шаблон елементів “екранів” **Регістр**?
11. Що таке шаблон елементів “екранів” **Лінійка**?
12. Що таке шаблон елементів “екранів” **Функціональна кнопка**?
13. Що таке шаблон елементів “екранів” **Графік**?
14. Що таке шаблон елементів “екранів” **Регістри ASCII-тексту**?

7.5. Рекомендована література

Базова література: [4–6].

Додаткова література: [1, 5, 6].

8. SCADA-СИСТЕМА SIMATIC WINCC: ОГЛЯД ОСНОВНИХ МОЖЛИВОСТЕЙ

8.1. Знайомство з SIMATIC WinCC

SIMATIC WinCC (далі – WinCC) – це потужна SCADA-система, яка може забезпечити досконалий людино-машинний інтерфейс у вигляді “людина – мнемосхема – машина” в межах концепції TIA (абр. від англ. Totally Integrated Automation, Повністю інтегрована автоматизація). WinCC – це абр. від англ. Windows Control Center. Далі нижче розглядається версія WinCC version 7.4.

Для симуляції роботи WinCC без підключення до автоматизованої системи може використовуватися **Симулятор WinCC** (WinCC Simulator).

8.2. Основні складові SIMATIC WinCC

WinCC є модульною SCADA-системою, яка утримує наступні дві основні складові:

- CS (абр. від англ. Configuration Software, Програмне забезпечення Системи розробки);
- RT (абр. від англ. Runtime Software, Програмне забезпечення Системи виконання).

“Ядром” CS є **Провідник WinCC** (WinCC Explorer).

В свою чергу, CS утримує наступні основні складові (в тому числі **Редактори системи розробки** (Configuration Editors)):

- **Комп’ютер** (Computer);
- **Керування ознаками** (Tag Management);
- **Дизайнер графіки** (Graphics Designer);
- **Меню і панелі інструментів** (Menus and Toolbars);
- **Текстові і графічні списки** (Text and graphics lists);
- **Реєстрація тривог** (Alarm Logging);
- **Реєстрація ознак** (Tag Logging);
- **Дизайнер звітів** (Report Designer);
- **Глобальний сценарій** (Global Script);
- **Бібліотека текстів** (Text Library);
- **Розподілювач тексту** (Text Distributor);
- **Адміністратор користувача** (User Administrator);
- **Перехресне посилання** (Cross Reference);

- **Завантаження Online-змін** (Load Online Changes);
- **Надлишковість** (Redundancy);
- **Архів користувача** (User Archive);
- **Синхронізація часу** (Time synchronization);
- **Звукова сигналізація** (Horn);
- **Дерево зображень** (Picture Tree);
- **Контроль працеспроможності** (Lifebeat Monitoring);
- **Редактор проєктів OS** (OS Project Editor);
- **Web-навігатор** (Web Navigator).

Взаємодія між складовими WinCC наведена на рис. 8.1.

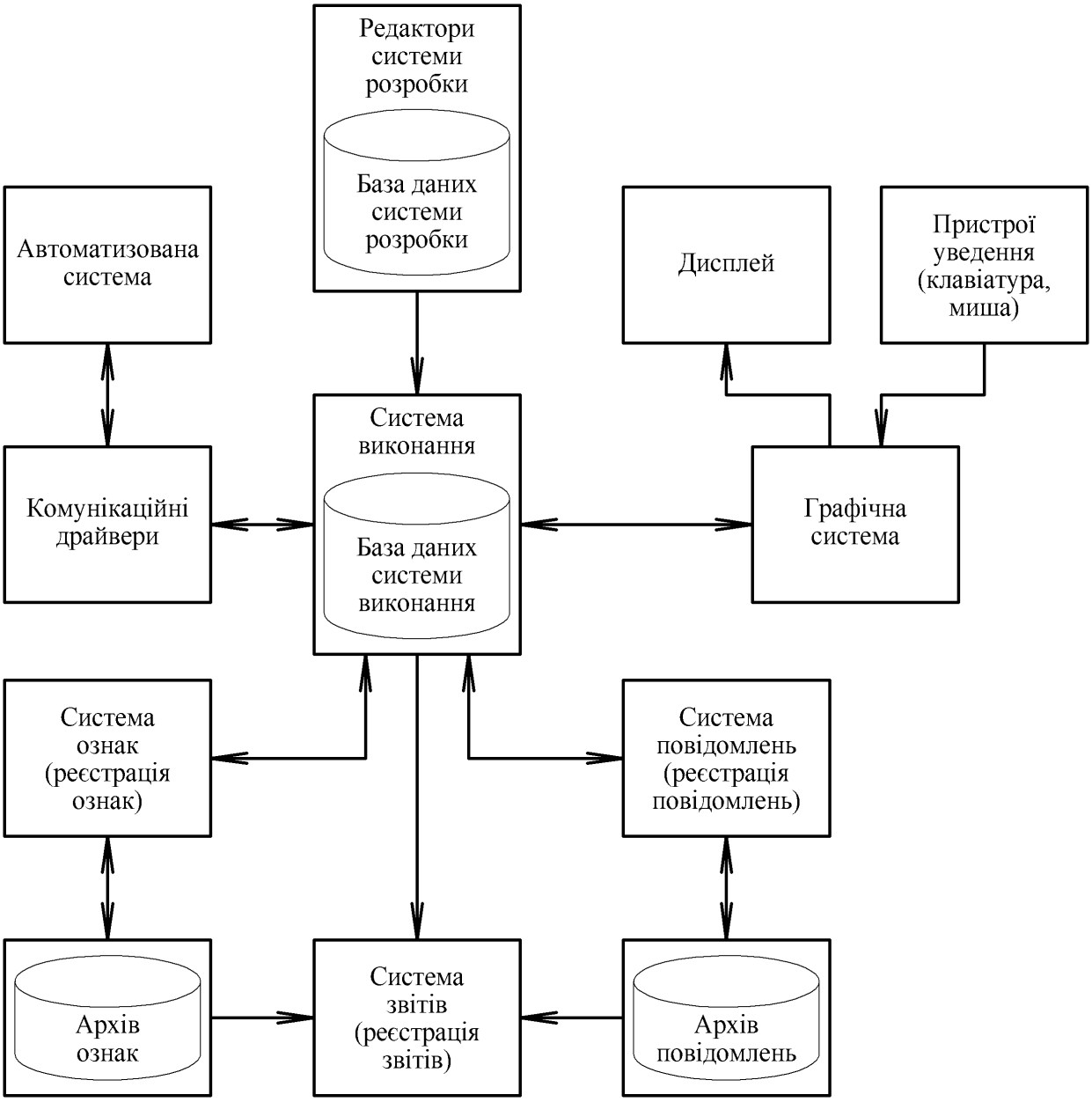


Рис. 8.1. Взаємодія між складовими WinCC

8.2.1. Комп’ютер (Computer)

Надаються відомості про:

- найменування комп’ютера;

- тип проєкту.

Проєкти можуть мати один з трьох типів:

- однокористувацький проєкт (використовується один комп’ютер, який одночасно буде і сервером, і клієнтом – але також можливо використання ще одного комп’ютера, який буде резервним або архівним сервером);

- багатористувацький проєкт (використовуються кілька комп’ютерів, які будуть кількома серверами і одним клієнтом або кількома серверами і кількома клієнтами);

- клієнтський проєкт (використовується один комп’ютер, який буде клієнтом, що може звертатись до одного сервера, або клієнтом, що може звертатись до кількох серверів).

Діалогове вікно **Властивості комп’ютера** (Computer properties) складової **Комп’ютер** (Computer) дозволяє налагодити наступні її властивості:

- на вкладці **Загальне** (General) – найменування комп’ютера і деякі інші;

- на вкладці **Автозавантаження** (Startup) – ті складові, які будуть доступні в **Системі виконання** (Runtime);

- на вкладці **Параметри** (Parameters) – мову **Системи виконання** (Runtime) і деякі інші;

- на вкладці **Система виконання графіки** (Graphics Runtime) – початкову мнемосхему і деякі інші;

- на вкладці **Система виконання** (Runtime) – режим налагодження під час виконання “скриптів” на мові програмування Visual Basic, використання оформлення в стилі “WinCC Classic”, використання віртуальної клавіатури на екрані монітора і деякі інші.

8.2.2. Керування ознаками (Tag Management)

8.2.2.1. Загальні відомості про Керування ознаками (Tag Management)

Керування ознаками (Tag Management) підтримує роботу з наступними **Ознаками** (Tag):

- внутрішніми (**Ознаки** (Tag), які не підтримують обмін даними із

системою автоматизації);

- зовнішніми (**Ознаки** (Tag), які підтримують обмін даними із системою автоматизації).

8.2.2.2. Внутрішні Ознаки (Tag) Керування ознаками (Tag Management)

Внутрішні **Ознаки** (Tag) підтримують обмін даними всередині **Системи виконання** (Runtime) (наприклад, при обміні даними з архівами).

Внутрішні **Ознаки** (Tag) мають наступні властивості (вибірково):

- **Найменування** (Name);
- **Значення** (Value);
- **Часова мітка** (Time stamp);
- **Код якості обміну даними** (Quality code);
- **Тип даних** (Data type);
- **Довжина** (Length);
- **З'єднання** (Connection);
- **Група** (Group).

Властивість **Тип даних** (Data type) може бути наступною:

- **Двійкове значення** (Binary Tag);
- **8-бітове значення із знаком** (Signed 8-bit value);
- **8-бітове значення без знаку** (Unsigned 8-bit value);
- **16-бітове значення із знаком** (Signed 16-bit value);
- **16-бітове значення без знаку** (Unsigned 16-bit value);
- **32-бітове значення із знаком** (Signed 32-bit value);
- **32-бітове значення без знаку** (Unsigned 32-bit value);
- **32-бітове число з плаваючою точкою** (Floating-point number 32-bit IEEE 754);
- **64-бітове число з плаваючою точкою** (Floating-point number 64-bit IEEE 754);
- **Текстове значення (8-бітове кодування символів)** (Text tag 8-bit character set);
- **Текстове значення (16-бітове кодування символів)** (Text tag 16-bit character set);
- **Необроблені дані** (Raw Data Type);
- **Посилання на текст** (Text reference);
- **Дата/Час** (Data/Time).

8.2.2.3. Зовнішні Ознаки (Tag) Керування ознаками (Tag Management)

Зовнішні **Ознаки** (Tag) підтримують обмін даними зовні **Системи виконання** (Runtime) (наприклад, при обміні даними з модулями аналогового уведення або дискретного виведення).

Порядок обміну даними між зовнішньою **Ознакою** (Tag) і системою автоматизації може бути наступним:

- з використанням груп **Ознак** (Tag);
- з використанням структурних **Ознак** (Tag).

Порядок обміну даними між зовнішньою **Ознакою** (Tag) і системою автоматизації (з використанням груп **Ознак** (Tag)) наведено на рис. 8.2.

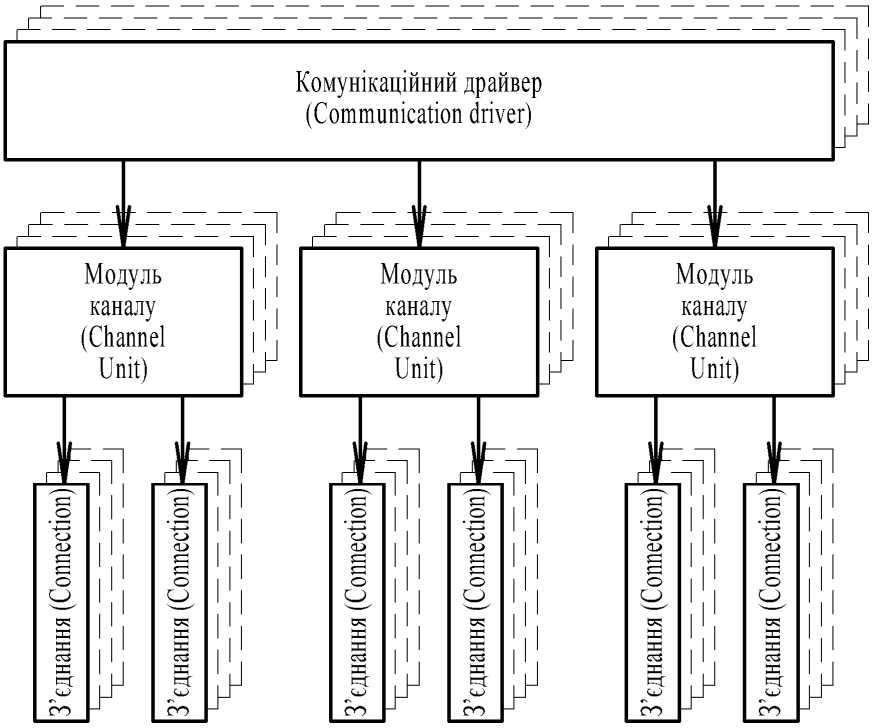


Рис. 8.2. Порядок обміну даними між зовнішньою **Ознакою** (Tag) і системою автоматизації (з використанням груп **Ознак** (Tag))

Комунікаційні драйвери (Communication drivers) можуть бути наступними (вибірково): SIMATIC S7-1200 Channel, SIMATIC S7-1500 Channel, ModBus TCPIP, OPC, OPC UA WinCC Chanel. Якщо було обрано, наприклад, OPC UA WinCC Chanel, то також буде автоматично створено **Модуль каналу** (Channel Unit) OPC UA Connections. Далі в такому модулі можна буде створювати кілька **З'єднань** (Connection), а в кожному з них створити (не обов'язково) кілька груп **Ознак** (Tag). В кожній з груп **Ознак** (Tag) (або

безпосередньо в кожному із **З'єднань** (Connection), якщо групи **Ознак** (Tag) не використовуються) можна створити кілька зовнішніх **Ознак** (Tag).

Порядок обміну даними між зовнішньою **Ознакою** (Tag) і системою автоматизації (з використанням структурних ознак) наведено на рис. 8.3.

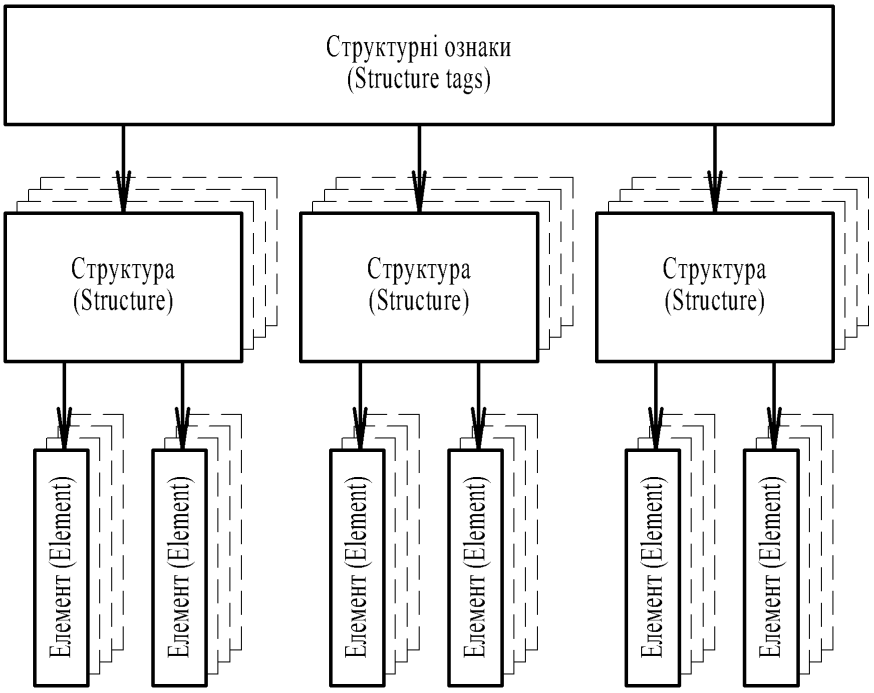


Рис. 8.3. Порядок обміну даними між зовнішньою **Ознакою** (Tag) і системою автоматизації (з використанням структурних **Ознак** (Tag))

Зовнішні **Ознаки** (Tag) мають наступні властивості (фактично ті ж самі, що і внутрішні **Ознаки** (Tag)) (вибірково):

- **Найменування** (Name);
- **Значення** (Value);
- **Часова мітка** (Time stamp);
- **Код якості обміну даними** (Quality code);
- **Тип даних** (Data type);
- **Довжина** (Length);
- **З'єднання** (Connection);
- **Група** (Group).

Властивість **Тип даних** (Data type) може бути наступною (за невеликою відмінністю тією ж самою, що і для внутрішньої **Ознаки** (Tag)):

- **Двійкове значення** (Binary Tag);
- **8-бітове значення із знаком** (Signed 8-bit value);

- **8-бітове значення без знаку** (Unsigned 8-bit value);

- **16-бітове значення із знаком** (Signed 16-bit value);

- **16-бітове значення без знаку** (Unsigned 16-bit value);

- **32-бітове значення із знаком** (Signed 32-bit value);

- **32-бітове значення без знаку** (Unsigned 32-bit value);

- **32-бітове число з плаваючою точкою** (Floating-point number 32-bit IEEE 754);

- **64-бітове число з плаваючою точкою** (Floating-point number 64-bit IEEE 754);

- **Текстове значення (8-бітове кодування символів)** (Text tag 8-bit character set);

- **Текстове значення (16-бітове кодування символів)** (Text tag 16-bit character set);

- **Необроблені дані** (Raw Data Type);

- **Дата/Час** (Data/Time).

Ознаки (Tag), які починаються з символу "@", звичайно створюються WinCC автоматично, і зміна їх властивостей є небажаною.

Якщо властивість **Тип даних** (Data type) певної **Ознаки** (Tag) і відповідна їй властивість системи автоматизації відрізняються, необхідно використання "перетворення типів". Наприклад, **Ознакам** (Tag), властивості **Тип даних** (Data type) яких без перетворення типів відповідає **Необроблені дані** (Raw Data Type), при використанні перетворення типів буде відповідати **32-бітове значення із знаком** (Signed 32-bit value), а **Ознакам** (Tag), властивості **Тип даних** (Data type) яких без перетворення типів відповідає **Двійкове значення** (Binary Tag), при використанні перетворення типів буде відповідати **8-бітове значення без знаку** (Unsigned 8-bit value),

8.2.3. Дизайнер графіки (Graphics Designer)

8.2.3.1. Загальні відомості про Дизайнер графіки (Graphics Designer)

Дизайнер графіки (Graphics Designer) призначений для:

- відображення об'єктів мнемосхеми (наприклад, **Лінія** (Line)), які мають статичні і динамічні властивості;

- забезпечення реакції об'єктів мнемосхеми на зовнішні впливи.

Мнемосхеми зберігаються в файлах, які мають формат .pdl (абр. від англ.

Process Display, Відображення процесу).

8.2.3.2. Стандартні об’єкти (Standard Objects)

1. Об’єкт **Лінія** (Line) (розміри) наведено на рис. 8.4.

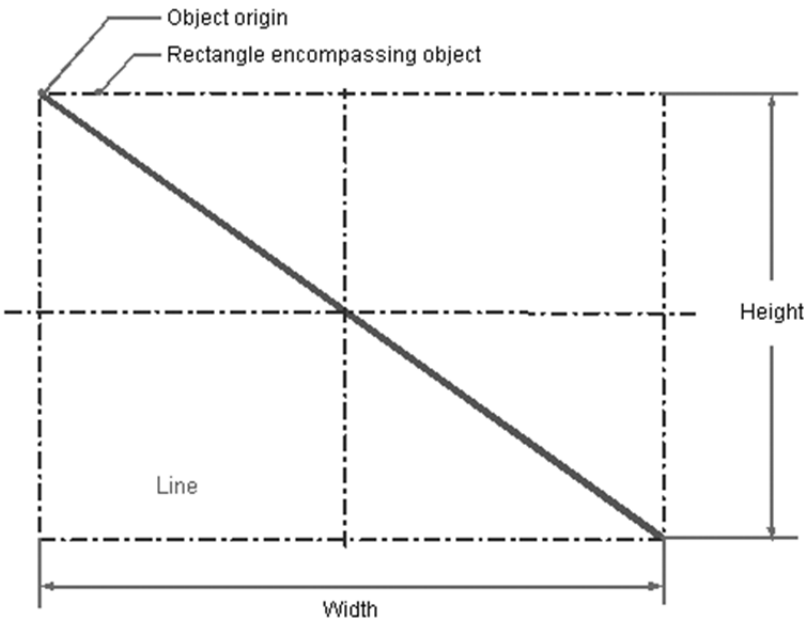


Рис. 8.4. Об’єкт **Лінія** (Line) (розміри)

2. Об’єкт **Багатокутник** (Polygon) (розміри) наведено на рис. 8.5.

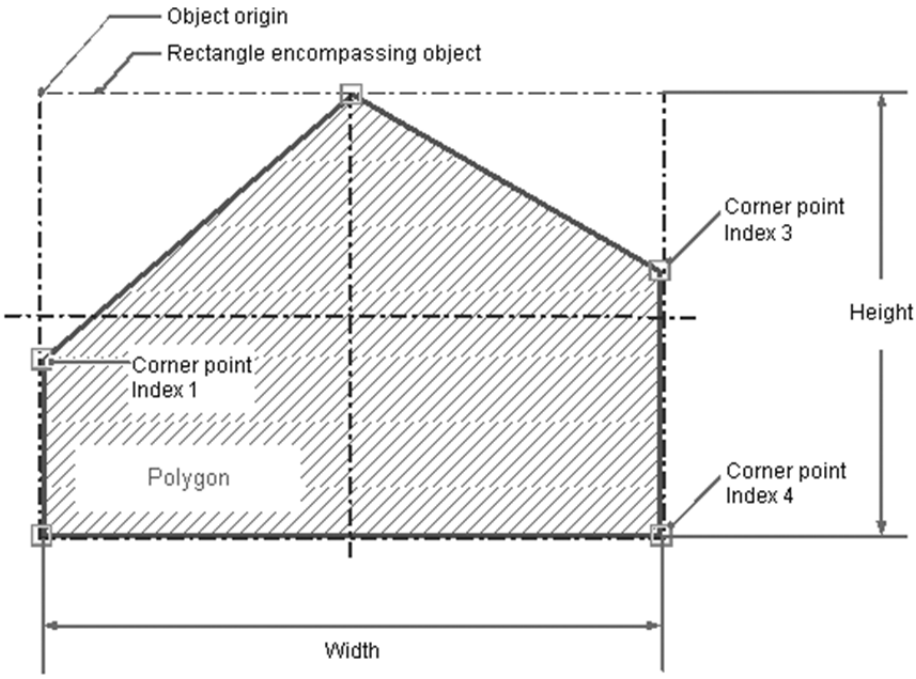


Рис. 8.5. Об’єкт **Багатокутник** (Polygon) (розміри)

3. Об’єкт **Ламана** (Polyline) (розміри) наведено на рис. 8.6.

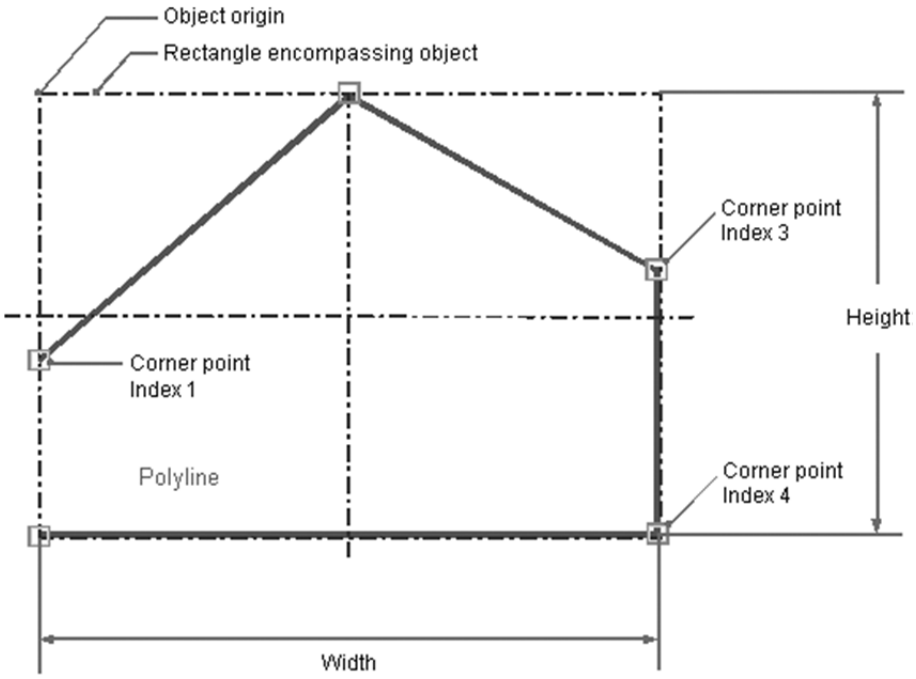


Рис. 8.6. Об’єкт **Ламана** (Polyline) (розміри)

4. Об’єкт **Еліпс** (Ellipse) (розміри) наведено на рис. 8.7.

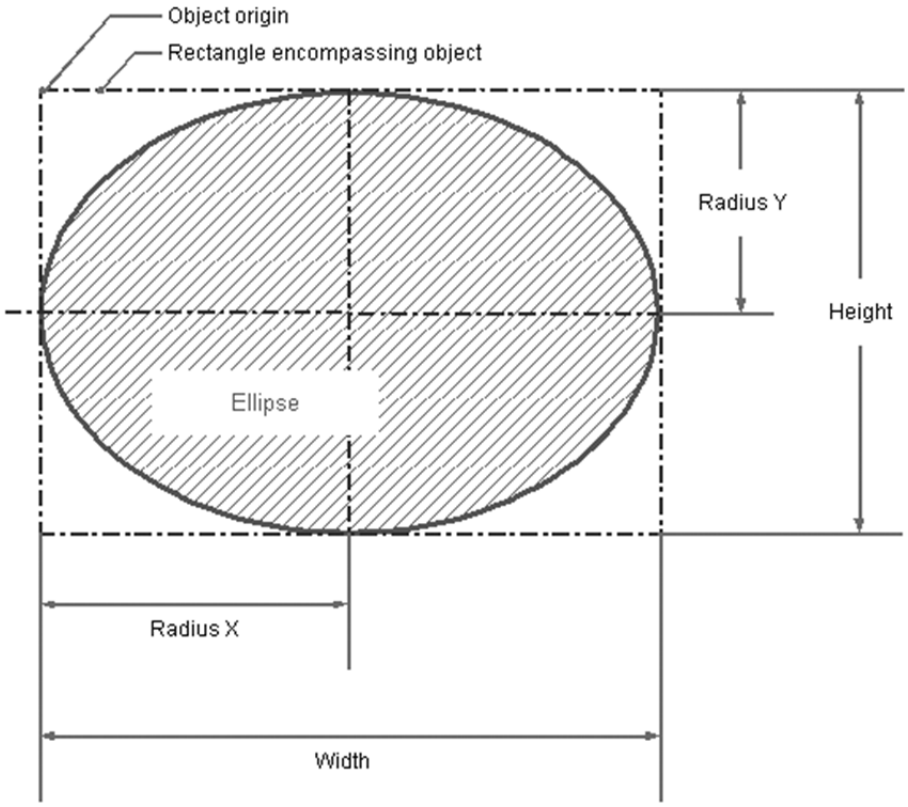


Рис. 8.7. Об’єкт **Еліпс** (Ellipse) (розміри)

5. Об’єкт **Коло** (Circle) (розміри) наведено на рис. 8.8.

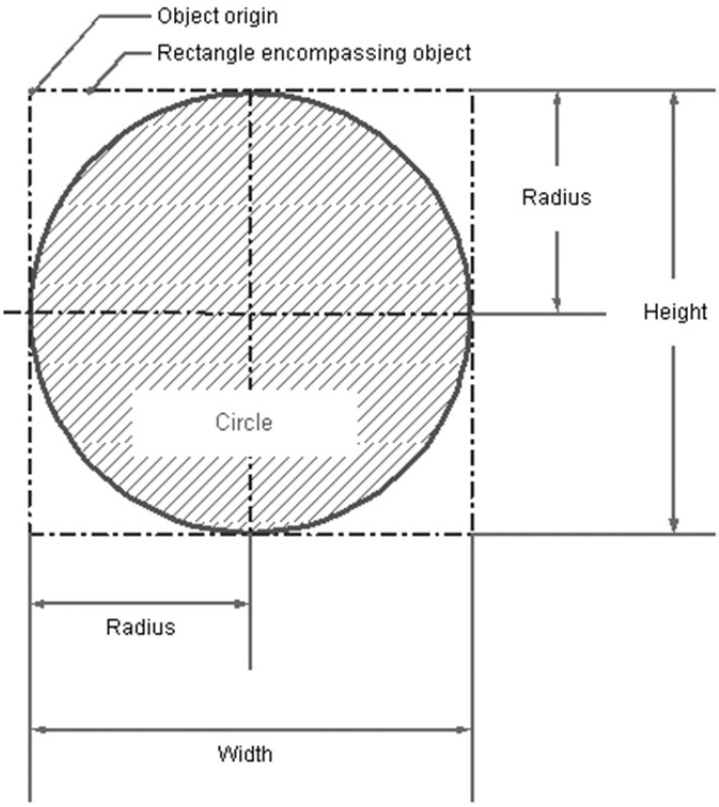


Рис. 8.8. Об’єкт **Коло** (Circle) (розміри)

6. Об’єкт **Сектор еліпса** (Ellipse Segment) (розміри) наведено на рис. 8.9.

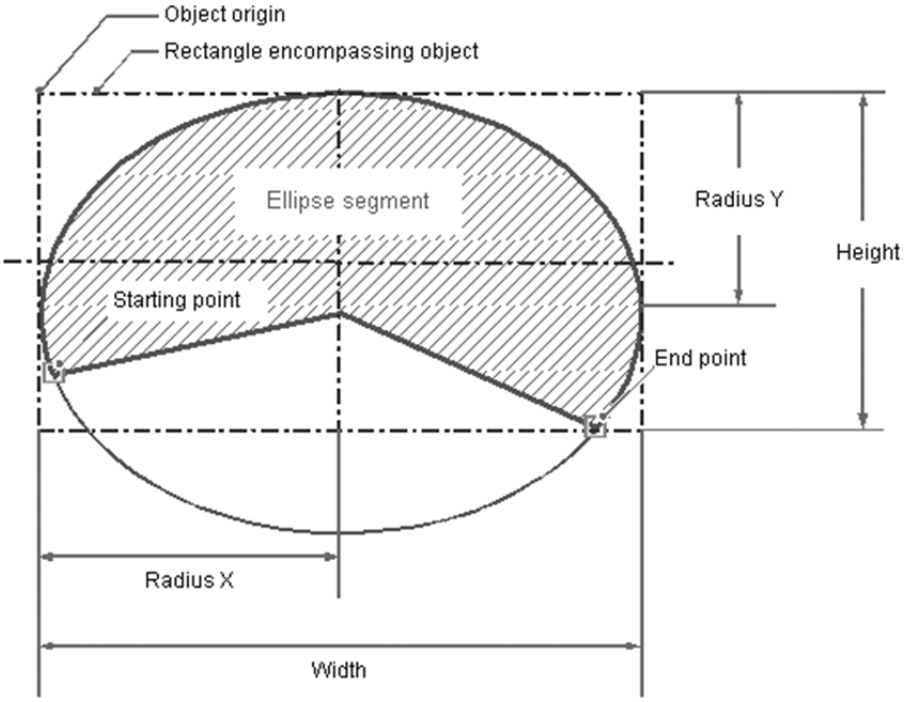


Рис. 8.9. Об’єкт **Сектор еліпса** (Ellipse Segment) (розміри)

7. Об’єкт **Сектор кола** (Pie Segment) (розміри) наведено на рис. 8.10.

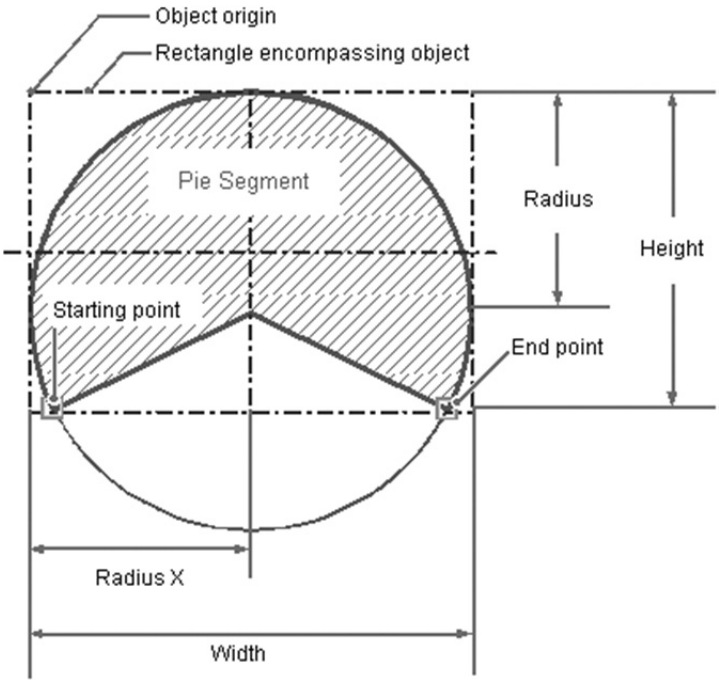


Рис. 8.10. Об’єкт **Сектор кола** (Pie Segment) (розміри)

8. Об’єкт **Дуга еліпса** (Ellipse Arc) (розміри) наведено на рис. 8.11.

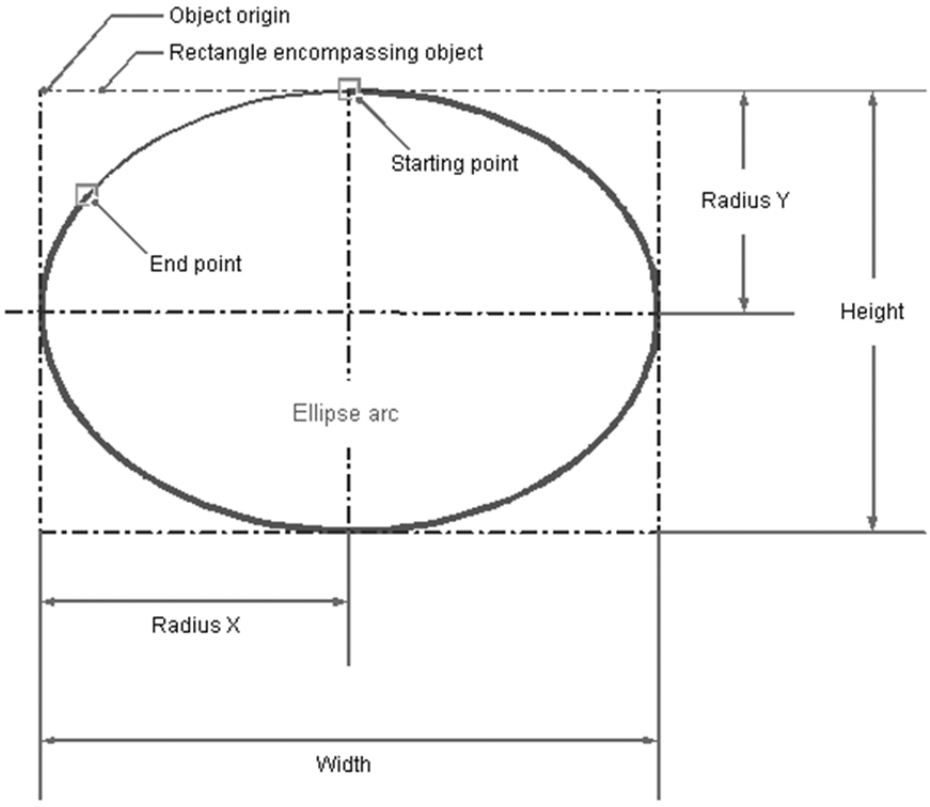


Рис. 8.11. Об’єкт **Дуга еліпса** (Ellipse Arc) (розміри)

9. Об’єкт **Дуга кола** (Circular Arc) (розміри) наведено на рис. 8.12.

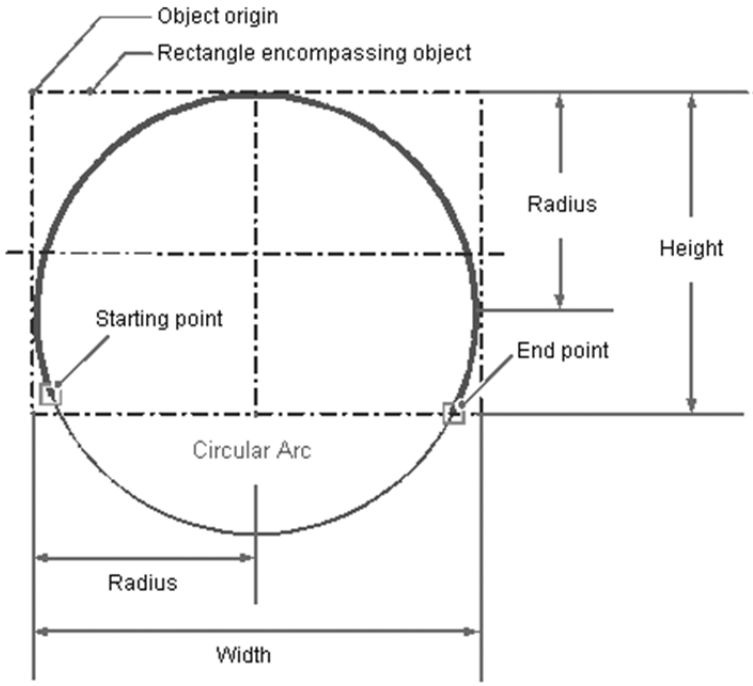


Рис. 8.12. Об’єкт **Дуга кола** (Circular Arc) (розміри)

10. Об’єкт **Прямокутник** (Rectangle) (розміри) наведено на рис. 8.13.

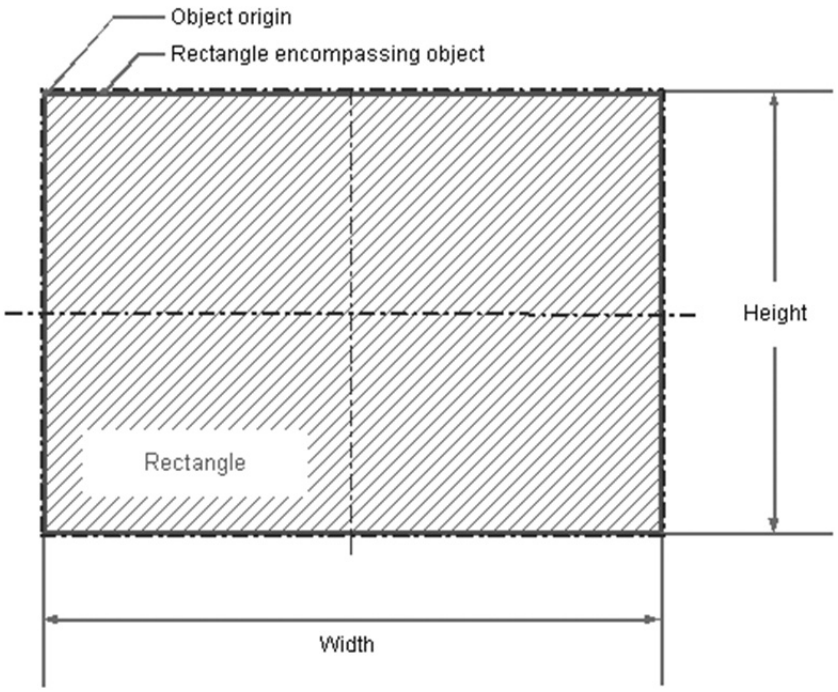


Рис. 8.13. Об’єкт **Прямокутник** (Rectangle) (розміри)

11. Об’єкт **Закруглений прямокутник** (Rounded Rectangle) (розміри) наведено на рис. 8.14.

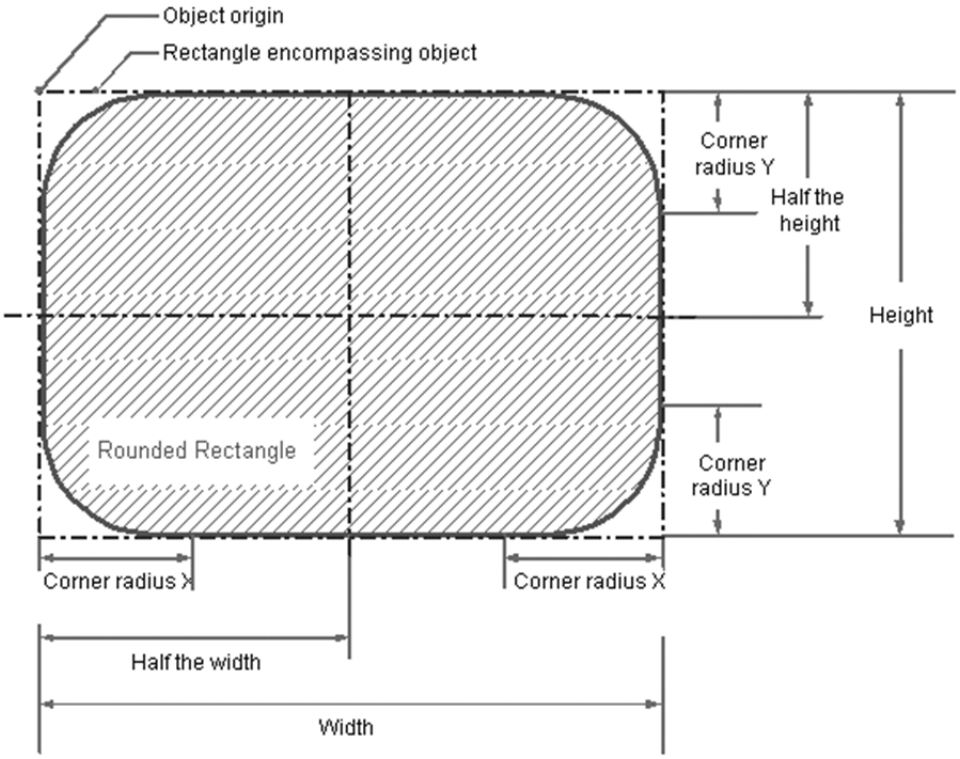


Рис. 8.14. Об’єкт **Закруглений прямокутник** (Rounded Rectangle) (розміри)

12. Об’єкт **Статичний текст** (Static Text) (розміри) наведено на рис. 8.15.

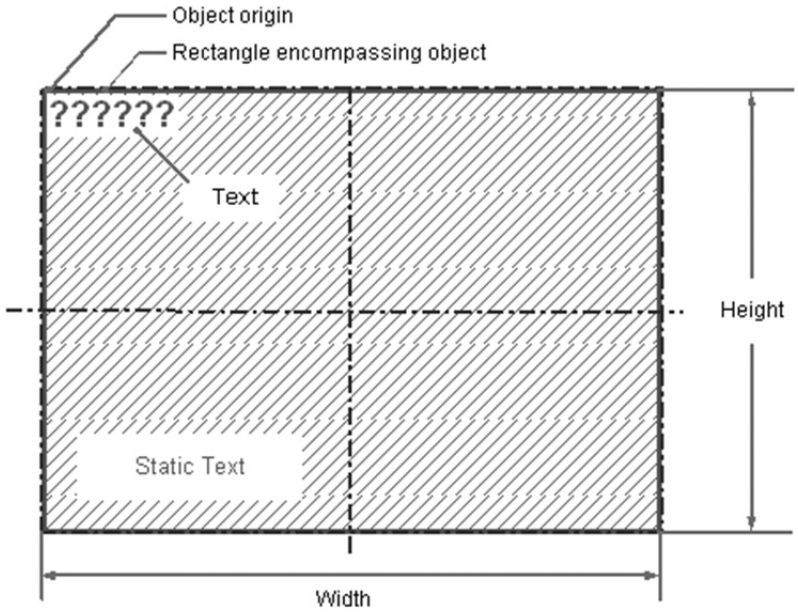


Рис. 8.15. Об’єкт **Статичний текст** (Static Text) (розміри)

8.2.3.3. Інтелектуальні об’єкти (Smart Objects) (вибірково)

1. Об’єкт **Вікно програми** (Application Window) (загальний вигляд) наведено на рис. 8.16.

Рис. 8.16. Об’єкт **Вікно програми** (Application Window) (загальний вигляд)

2. Об’єкт **Вікно зображення** (Picture Window) (загальний вигляд) наведено на рис. 8.17.

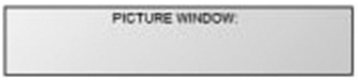


Рис. 8.17. Об’єкт **Вікно зображення** (Picture Window) (загальний вигляд)

3. Об’єкт **Керування** – **Контроль тренду функції** (Control – WinCC Function Trend Control) (загальний вигляд) наведено на рис. 8.18.

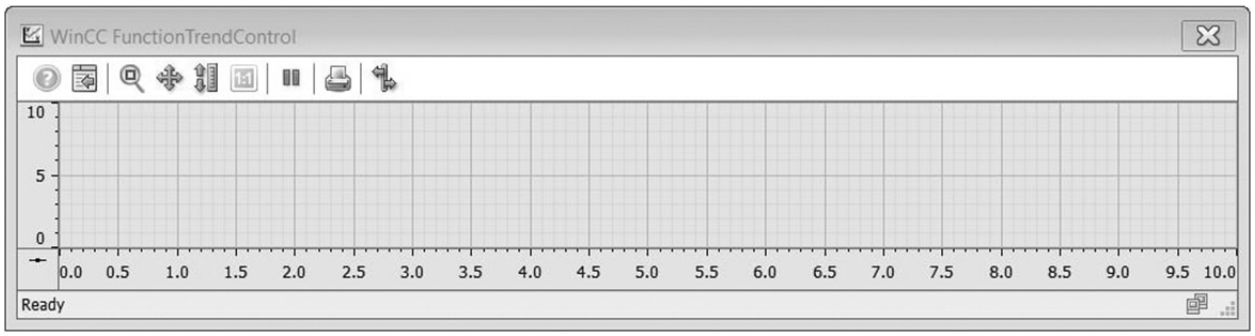


Рис. 8.18. Об’єкт **Керування** – **Контроль тренду функції WinCC** (Control – WinCC Function Trend Control) (загальний вигляд)

4. Об’єкт **Об’єкт OLE** – **Зображення PaintBrush** (OLE Object – Зображення PaintBrush) (загальний вигляд) наведено на рис. 8.19.

Рис. 8.19. Об’єкт **Об’єкт OLE** – **Зображення PaintBrush** (OLE Object – Зображення PaintBrush) (загальний вигляд)

5. Об’єкт **Поле уведення/виведення** (I/O Field) (загальний вигляд) наведено на рис. 8.20.

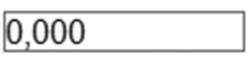


Рис. 8.20. Об’єкт **Поле уведення/виведення** (I/O Field) (загальний вигляд)

6. Об’єкт **Рівень** (Bar) (загальний вигляд) наведено на рис. 8.21.

Рис. 8.21. Об’єкт **Рівень** (Bar) (загальний вигляд)

7. Об’єкт **Графічний об’єкт** – **Замок** (Graphic Object – Lock) (загальний вигляд) наведено на рис. 8.22.

Рис. 8.22. Об’єкт **Графічний об’єкт** – **Замок** (Graphic Object – Lock) (загальний вигляд)

8. Об’єкт **Відображення стану** – **Користувач** (Status Display – User) (загальний вигляд) наведено на рис. 8.23.

Рис. 8.23. Об’єкт **Відображення стану** – **Користувач** (Status Display – User) (загальний вигляд)

9. Об’єкт **Текстовий список** (Text List) (загальний вигляд) наведено на рис. 8.24.

Рис. 8.24. Об’єкт **Текстовий список** (Text List) (загальний вигляд)

10. Об’єкт **Багаторядковий текст** (Multiple Row Text) (загальний вигляд) наведено на рис. 8.25.

Рис. 8.25. Об’єкт **Багаторядковий текст** (Multiple Row Text) (загальний вигляд)

11. Об’єкт **Комбіноване вікно** (Combo Box) (загальний вигляд) наведено на рис. 8.26.

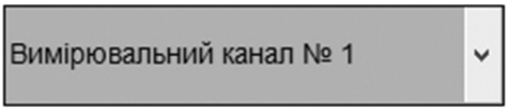


Рис. 8.26. Об’єкт **Комбіноване вікно** (Combo Box) (загальний вигляд)

12. Об’єкт **Спискове вікно** (List Box) (загальний вигляд) наведено на рис. 8.27.

Рис. 8.27. Об’єкт **Спискове вікно** (List Box) (загальний вигляд)

13. Об’єкт **Екземпляр лицьової панелі** (Faceplate Instance) (загальний вигляд) наведено на рис. 8.28.

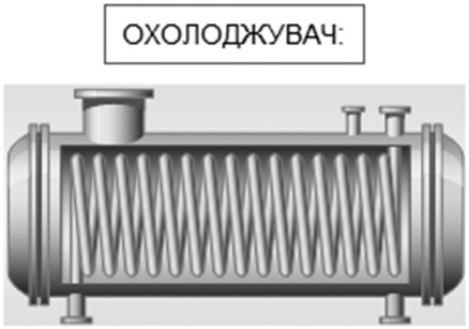


Рис. 8.28. Об’єкт **Екземпляр лицьової панелі** (Faceplate Instance) (загальний вигляд)

14. Об’єкт **3D-рівень** (3D-bar) (загальний вигляд) наведено на рис. 8.29.

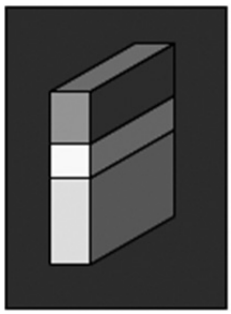


Рис. 8.29. Об’єкт **3D-рівень** (3D-bar) (загальний вигляд)

15. Об’єкт **Відображення групи** (Group Display) (загальний вигляд) наведено на рис. 8.30.

Рис. 8.30. Об’єкт **Відображення групи** (Group Display) (загальний вигляд)

16. Об’єкт **Набір даних** (DataSet) (загальний вигляд) наведено на рис. 8.31.

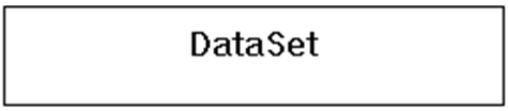


Рис. 8.31. Об’єкт **Набір даних** (DataSet) (загальний вигляд)

17. Об’єкт **Об’єкт SVG** (SVG Object) (загальний вигляд) наведено на рис. 8.32.

Рис. 8.32. Об’єкт **Об’єкт SVG** (SVG Object) (загальний вигляд)

8.2.3.4. Об’єкти Windows (Windows Objects)

1. Об’єкт **Кнопка** (Button) (загальний вигляд) наведено на рис. 8.33.

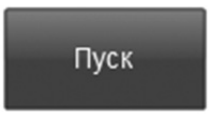


Рис. 8.33. Об’єкт **Кнопка** (Button) (загальний вигляд)

2. Об’єкт **Вікно з кнопками незалежного вибору** (Check Box) (загальний вигляд) наведено на рис. 8.34.

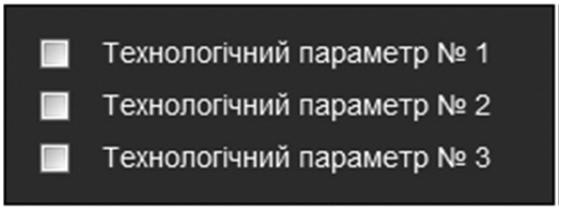


Рис. 8.34. Об’єкт **Вікно з кнопками незалежного вибору** (Check Box) (загальний вигляд)

3. Об’єкт **Група з кнопками залежного вибору** (Option Group) (загальний вигляд) наведено на рис. 8.35.

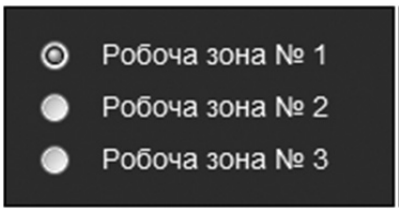


Рис. 8.35. Об’єкт **Група з кнопками залежного вибору** (Option Group) (загальний вигляд)

4. Об’єкт **Кнопка кругла** (Round Button) (загальний вигляд) наведено на рис. 8.36.

Рис. 8.36. Об’єкт **Кнопка кругла** (Round Button) (загальний вигляд)

5. Об’єкт **Об’єкт Повзунок** (Slider Object) (загальний вигляд) наведено на рис. 8.37.

Рис. 8.37. Об’єкт **Об’єкт Повзунок** (Slider Object) (загальний вигляд)

8.2.4. Меню і панелі інструментів (Menus and Toolbars)

Меню і панелі інструментів (Menus and Toolbars) дозволяє:

- створювати **Меню** (Menus);
- створювати **Панелі інструментів** (Toolbars).

Кожен з **Пунктів меню** (Menu item) має наступні властивості:

- **Найменування** (Name);
- **Текст** (Text);
- **Активний** (Active);
- **Видимий** (Visible);
- **Роздільник** (Separator);
- **Авторизація** (Authorization);
- **Дані користувача** (User Data);
- **Сценарій** (Script);
- **Зображення** (Picture).

Кожен з **Пунктів панелі інструментів** (Toolbar item) має наступні властивості:

- **Найменування** (Name);
- **Текст** (Text);
- **Контекстний текст** (Tooltip Text);
- **Активний** (Active);
- **Видимий** (Visible);
- **Роздільник** (Separator);
- **Сценарій** (Script);
- **Авторизація** (Authorization);
- **Зображення** (Picture).
- **Дані користувача** (User Data).

8.2.5. Текстові і графічні списки (Text and Graphics Lists)

Текстові і графічні списки (Text and Graphics Lists) надає можливість:

- створювати **Текстові списки** (Text Lists);
- створювати **Графічні списки** (Graphics Lists).

Складові кожного з **Текстових списків** (Text Lists) і **Графічних списків** (Graphics Lists) можуть нумеруватися:

- задаючи для кожної з них номера 0 або 1 (в такому випадку в списку може бути максимум 2 складові);

- задаючи для кожної з них номера від 0 до 31 (в такому випадку в списку може бути максимум 32 складові);

- задаючи для кожної з них діапазон значень номерів (наприклад, від −65 до −35 або від 100 до 120).

8.2.6. Реєстрація тривог (Alarm Logging)

Реєстрація тривог (Alarm Logging) дозволяє реєструвати **Тривоги** (Alarm).

За замовчуванням доступні наступні **Класи повідомлень** (Message Classes):

- **Помилка** (Error);
- **Системні, які вимагають підтвердження** (System, requires acknowledgment);
- **Системні, які не вимагають підтвердження** (System, without acknowledgment).

8.2.7. Реєстрація ознак (Tag Logging)

Реєстрація ознак (Tag Logging) дозволяє реєструвати **Ознаки** (Tag) з подальшим збереженням їх в архівах:

- на накопичувачах (SSD і/або HDD);
- в оперативній пам’яті.

Архіви можуть бути:

- **Архіви значень процесу** (Process Value Archives);
- **Стиснуті архіви** (Compressed Archives).

8.2.8. Дизайнер звітів (Report Designer)

Дизайнер звітів (Report Designer) дозволяє як створювати, так і виводити звіти:

- за допомогою **Редактора макета сторінки** (Page Layout Editor);
- за допомогою **Редактора макета рядка** (Line Layout Editor).

Звіти можна виводити:

- на екрани моніторів комп’ютерів (можна тільки спостерігати);
- у файли (зберігаються на накопичувачах (SSD і/або HDD));
- на принтери (відбувається друк).

8.2.9. Глобальний сценарій (Global Script)

8.2.9.1. Загальні відомості про Глобальний сценарій (Global Script)

Для додаткової обробки даних WinCC **Глобальний сценарій** (Global Script) надає можливість:

- використовувати **Редактор С** (C-Editor), який дозволяє створювати програмний код на мові програмування С;
- використовувати **Редактор VBS** (VBS-Editor), який дозволяє створювати програмний код на мові програмування Visual Basic.

8.2.9.2. Редактор С (C-Editor)

Редактор С (C-Editor) дозволяє на мові програмування С створювати:

- **Дії** (Action);
- **Функції проєкту** (Project Function);
- **Стандартні функції** (Standard Function);
- **Файли заголовків** (Header File).

8.2.9.3. Редактор VBS (VBS-Editor)

Редактор VBS (VBS-Editor) дозволяє на мові програмування Visual Basic створювати:

- **Дії** (Action);
- **Модулі проєкту** (Project Module);
- **Стандартні модулі** (Standard Module).

8.2.10. Бібліотека текстів (Text Library)

Бібліотека текстів (Text Library) надає можливість зберігати тексти одночасно на кількох різних мовах (наприклад, на українській і англійській), причому кожному з текстів призначається унікальний номер. Тексти можна експортувати, редагувати (наприклад, з метою перевірки перекладу) і знову імпортувати.

8.2.11. Розподілювач тексту (Text Distributor)

Розподілювач тексту (Text Distributor) дозволяє здійснювати експорт і імпорт текстів, пов’язаних с кількома різними мовами (наприклад, можна

створити мнемосхему з текстами на українській мові, експортувати їх, поміняти уміст текстів з української мови на англійську мову і знову імпортувати ці тексти – відповідно, в наявності будуть як англійська, так і українська “версії” тієї ж самої мнемосхеми).

8.2.12. Адміністратор користувача (User Administrator)

Адміністратор користувача (User Administrator) – це важливий “інструмент”, який дозволяє запобігти несанкціонованому втручанню в роботу автоматизованої системи. Він дозволяє створювати облікові записи **Користувач** (Users) з унікальними **Логон** (Logon) і **Пароль** (Password), які розподіляються по **Групи користувачів** (User Groups) і можуть мати різні повноваження: **Адміністратор користувача** (User Administration), **Авторизація для області** (Authorization for area), **Зміна системи** (System change), **Моніторинг** (Monitoring) і так далі.

8.2.13. Перехресне посилання (Cross Reference)

Перехресне посилання (Cross Reference) утримує всі складові проєкту, на які в такому проєкті є посилання – до них відносяться **Ознака** (Tag), **Зображення (PDL)** (Picture (PDL)), **Об’єкт OLE** (OLE object), **Стандартна функція C** (C-standard function), **Функція проєкту VBS** (VBS project function) і так далі.

8.2.14. Завантаження Online-змін (Load Online Changes)

Завантаження Online-змін (Load Online Changes) надає можливість редагувати проєкт на одному комп’ютері в той час, коли він виконується на іншому комп’ютері (одночасно на таке редагування накладаються певні обмеження).

8.2.15. Надлишковість (Redundancy)

Надлишковість (Redundancy) дозволяє одночасно використовувати два комп’ютера, які є серверами, причому один з них повинен бути “основним”, а інший – “резервним”. Таке використання забезпечує “підвищену доступність” WinCC. В якості середовища обміну даними між комп’ютерами рекомендується використовувати кабелі саме на основі мідних/алюмінієвих провідників або оптичного волокна (проводовий зв’язок), а не електромагнітні хвилі

(безпроводовий зв’язок).

8.2.16. Архів користувача (User Archive)

Архів користувача (User Archive) надає можливість утримувати дані (наприклад, “рецепти” для кожного типу продукції) і здійснювати обмін ними з різними складовими автоматизованої системи (серверами, клієнтами, промисловими контролерами і так далі) як в реальному часі, так і при виникненні такої потреби (за запитом).

8.2.17. Синхронізація часу (Time Synchronization)

Синхронізація часу (Time Synchronization) дозволяє забезпечити синхронізацію часу між різними складовими автоматизованої системи (серверами, клієнтами, промисловими контролерами і так далі):

- використовуючи **Термінальну шину** (Terminal Bus) в режимі “веденого” (slave);
- використовуючи **Системну шину** (System Bus) в режимах “ведучого” (master) і/або “веденого” (slave).

8.2.18. Звукова сигналізація (Horn)

Звукова сигналізація (Horn) дозволяє, в залежності від отриманого повідомлення, здійснювати:

- звукову сигналізацію (використовуючи звукові карти комп’ютерів);
- звукову і/або світлову сигналізацію (використовуючи спеціально розроблені “модулі сигналів”).

Типовий “модуль сигналів” має:

- 3-и незалежні канали керування звукову і/або світлову сигналізацію;
- сторожовий блок;
- 3-и незалежні двійкові входи;
- можливість підтвердження того, що звукове і/або світлове повідомлення отримано.

8.2.19. Дерево зображень (Picture Tree)

Дерево зображень (Picture Tree) надає можливість керувати зображеннями мнемосхем, які використовуються WinCC (такі зображення можуть утримувати такі об’єкти мнемосхем, як **Лінія** (Line), **Багатокутник**

(Polygon), **Ламана** (Polyline), **Еліпс** (Ellipse) і так далі).

8.2.20. Контроль працеспроможності (Lifebeat Monitoring)

WinCC як SCADA-система є, згідно з концепцією TIA, частиною SIMATIC PCS 7 (абр. від англ. Process Control System, Система керування процесами) – потужної системи організації виробництва, яку умовно можна віднести до 3-го рівня промислової автоматизації MES (див. рис. 1.4). До складу SIMATIC PCS 7 входять, крім WinCC, засоби програмування промислових контролерів, промислові мережі, засоби зв'язку з 4-м рівнем промислової автоматизації ERP і MRP (див. рис. 1.4) і так далі.

Контроль працеспроможності (Lifebeat Monitoring) надає можливість циклічно перевіряти в однойменному вікні доступність різних складових SIMATIC PCS 7, таких як OS (абр. від англ. Operator Station, Станція оператора) (наприклад, комп'ютери, які є серверами) і AS (абр. від англ. Automation Systems, Автоматизована система) (наприклад, промислові контролери).

8.2.21. Редактор проєктів OS (OS Project Editor)

Редактор проєктів OS (OS Project Editor) надає можливість налагодити такі три області OS, як **Область перегляду** (Overview area), **Робоча область** (Working area) і **Панель з кнопками** (Button bar). Такі налагодження можна здійснити в однойменному вікні:

- на вкладці **Макет** (Layout);
- на вкладці **Конфігурація повідомлень** (Message Configuration);
- на вкладці **Відображення повідомлень** (Message Display);
- на вкладці **Область** (Area);
- на вкладці **Вікно системи виконання** (Runtime Window);
- на вкладці **Основні дані** (Basic Data);
- на вкладці **Загальне** (General);

8.2.22. Web-навігатор (Web Navigator)

Web-навігатор (Web Navigator) надає можливість отримати доступ до автоматизованої системи керування через локальну або глобальну мережу (зокрема, через Internet).

Він утримує наступні 3-и основні складові:

- **Web-навігатор сервер** (WebNavigator server) – розміщується на

комп’ютері, який є сервером, і дозволяє отримати доступ до автоматизованої системи керування безпосередньо (на такому комп’ютері повинні розміщуватися також проєкт і WinCC, а сам комп’ютер повинен мати всі інтерфейси, необхідні для підключення його до автоматизованої системи керування);

- **Web-навігатор клієнт** (WebNavigator client) – розміщується на комп’ютері, який є клієнтом, і дозволяє отримати доступ до автоматизованої системи керування за допомогою браузера операційної системи Windows (наприклад, Edge) через локальну або глобальну мережу і далі через комп’ютер, який є сервером, із розміщеним на ньому WebNavigator server (допускається використання не тільки одного, але й кількох таких складових одночасно);

- **Web-навігатор діагностичний клієнт** (WebNavigator diagnostics client) – може виконувати те ж саме, що і WebNavigator client, але додатково дозволяє здійснювати в будь-який момент часу “гарантований” доступ до комп’ютера, який є сервером (крім того, він може використовуватися для докладної діагностики роботи **Web-навігатор сервер** (WebNavigator server)).

Програмне забезпечення **Видавець Web-представлень** (Web View Publisher), яке є складовою **Web-навігатор клієнт** (WebNavigator client), дозволяє розміщувати на Web-сайті мнемосхеми SCADA-системи.

8.3. Використання SIMATIC WinCC в промисловій автоматизації

SIMATIC WinCC може мати дуже широке використання, але найбільш ефективним є її використання в наступних областях діяльності людини:

- аеро- і космічне виробництво (в тому числі оборонне);
- виробництво транспортних засобів (зокрема, автомобілів);
- виробництво засобів автономного живлення (в тому числі акумуляторних батарей);
- виробництво цементу (а також сумішей на основі цементу);
- виробництво побутових і промислових хімічних речовин;
- операції з промисловою нерухомістю;
- фасування товарів;
- операції в центрах обробки даних;
- виробництво електроніки;
- виробництво скла;
- охорона здоров’я;

- виробництво важкої техніки;
- проведення лабораторних досліджень;
- виробництво певних груп побутових товарів;
- засоби для “покращення якості життя”;
- логістика;
- машинобудування;
- морські перевезення;
- металургія і обробка металів;
- видобування корисних копалин (зокрема, вугілля, нафти, газу, сланців, торфу);
- сонячна і вітряна енергетика.

8.4. Питання для самоперевірки

1. Що таке SCADA-система SIMATIC WinCC?
2. Які основні складові має SCADA-система SIMATIC WinCC?
3. Для чого призначена складова SCADA-система SIMATIC WinCC **Комп’ютер** (Computer)?
4. Для чого призначена складова SCADA-система SIMATIC WinCC **Керування ознаками** (Tag Management)?
5. Для чого призначена складова SCADA-система SIMATIC WinCC **Дизайнер графіки** (Graphics Designer)?
6. Для чого призначена складова SCADA-система SIMATIC WinCC **Меню і панелі інструментів** (Menus and Toolbars)?
7. Для чого призначена складова SCADA-система SIMATIC WinCC **Текстові і графічні списки** (Text and graphics lists)?
8. Для чого призначена складова SCADA-система SIMATIC WinCC **Реєстрація тривог** (Alarm Logging)?
9. Для чого призначена складова SCADA-система SIMATIC WinCC **Реєстрація ознак** (Tag Logging)?
10. Для чого призначена складова SCADA-система SIMATIC WinCC **Дизайнер звітів** (Report Designer)?
11. Для чого призначена складова SCADA-система SIMATIC WinCC **Глобальний сценарій** (Global Script)?
12. Для чого призначена складова SCADA-система SIMATIC WinCC **Бібліотека текстів** (Text Library)?
13. Для чого призначена складова SCADA-система SIMATIC WinCC

Розподілювач тексту (Text Distributor)?

14. Для чого призначена складова SCADA-система SIMATIC WinCC **Адміністратор користувача** (User Administrator)?

15. Для чого призначена складова SCADA-система SIMATIC WinCC **Перехресне посилання** (Cross Reference)?

16. Для чого призначена складова SCADA-система SIMATIC WinCC **Завантаження Online-змін** (Load Online Changes)?

17. Для чого призначена складова SCADA-система SIMATIC WinCC **Надлишковість** (Redundancy)?

18. Для чого призначена складова SCADA-система SIMATIC WinCC **Архів користувача** (User Archive)?

19. Для чого призначена складова SCADA-система SIMATIC WinCC **Синхронізація часу** (Time Synchronization)?

20. Для чого призначена складова SCADA-система SIMATIC WinCC **Звукова сигналізація** (Horn)?

21. Для чого призначена складова SCADA-система SIMATIC WinCC **Дерево зображень** (Picture Tree)?

22. Для чого призначена складова SCADA-система SIMATIC WinCC **Контроль працеспроможності** (Lifebeat Monitoring)?

23. Для чого призначена складова SCADA-система SIMATIC WinCC **Редактор проєктів OS** (OS Project Editor)?

24. Для чого призначена складова SCADA-система SIMATIC WinCC **Web-навігатор** (Web Navigator)?

8.5. Рекомендована література

Базова література: [4–6].

Додаткова література: [1, 3, 6].

9. ПРИКЛАДИ РЕАЛІЗАЦІЇ SCADA-СИСТЕМ В ПРОМИСЛОВІЙ АВТОМАТИЗАЦІЇ

9.1. SCADA-система для системи керування вентиляцією

Схема автоматизації системи керування вентиляцією наведена на рис. 9.1. Основною складовою такої системи є програмований контролер.

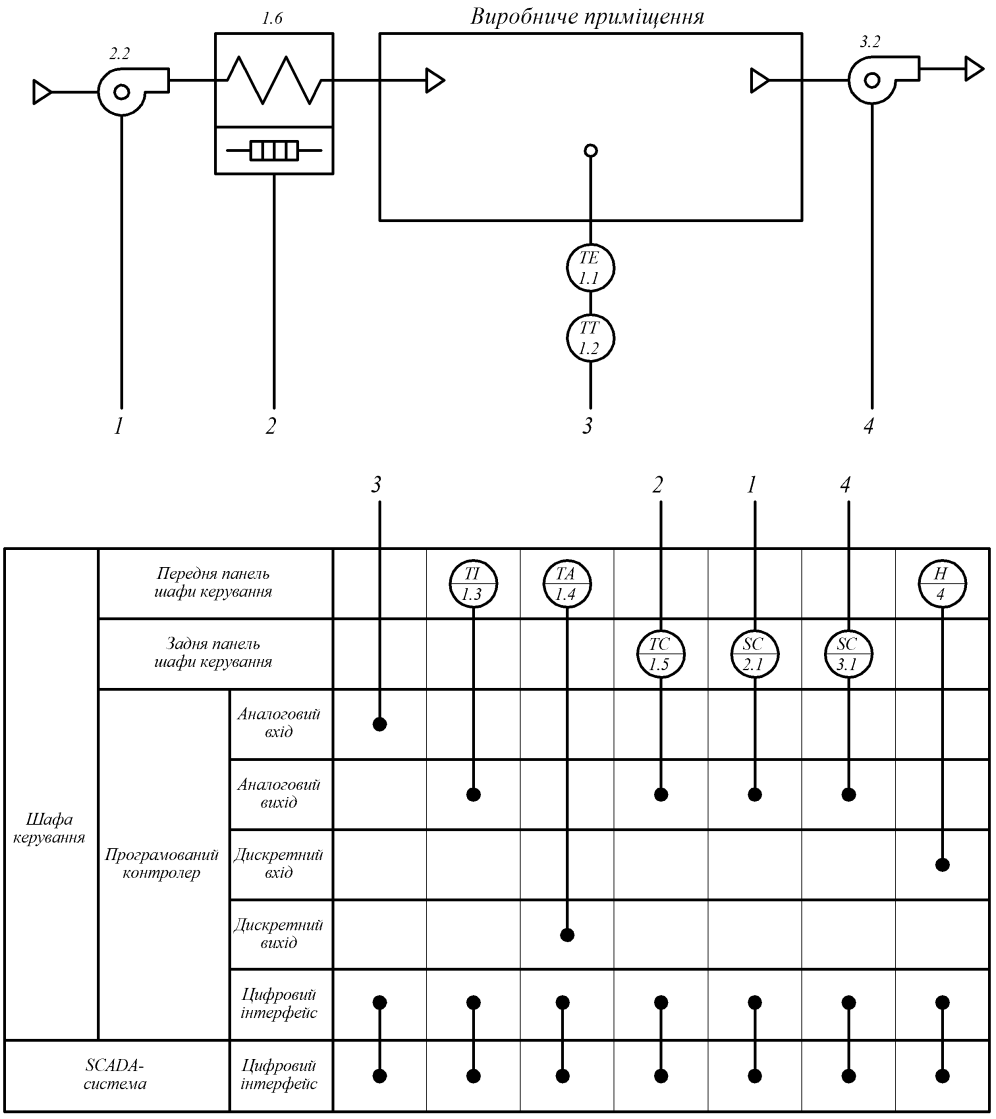


Рис. 9.1. Схема автоматизації системи керування вентиляцією

В якості об’єкта керування системи керування вентиляцією виступає виробниче приміщення, яке необхідно провітрювати і в якому необхідно підтримувати задану температуру повітря.

За допомогою датчика температури 1.1 у виробничому приміщенні виконується вимірювання температури повітря, за допомогою нормуючого

перетворювача 1.2 сигнал від чутливого елемента такого датчика перетворюється в уніфікований електричний неперервний сигнал, за допомогою індикатора 1.3 виконується показ значення температури, за допомогою сигналізатора 1.4 – сигналізація про відхилення температури від заданого значення, за допомогою регулятора 1.5 – регулювання температури шляхом використання електричного теплообмінника 1.6. За допомогою регулятора 2.1 виконується регулювання обертів припливного вентилятора 2.2, за допомогою регулятора 3.1 – витяжного вентилятора 3.2. Загальне керування вентиляцією виконується за допомогою пульта 4.

Температура повітря у виробничому приміщенні може підтримуватися шляхом;

- зміни електричної потужності, яка підводиться до електричного теплообмінника 1.6;
- зміни продуктивності припливного вентилятора 2.2, який впливає на кількість теплого повітря, що надходить у виробниче приміщення;
- зміни продуктивності витяжного вентилятора 3.2, який впливає на кількість теплого повітря, що видаляється з виробничого приміщення.

Мнемосхема SCADA-системи для системи керування вентиляцією наведена на рис. 9.2.

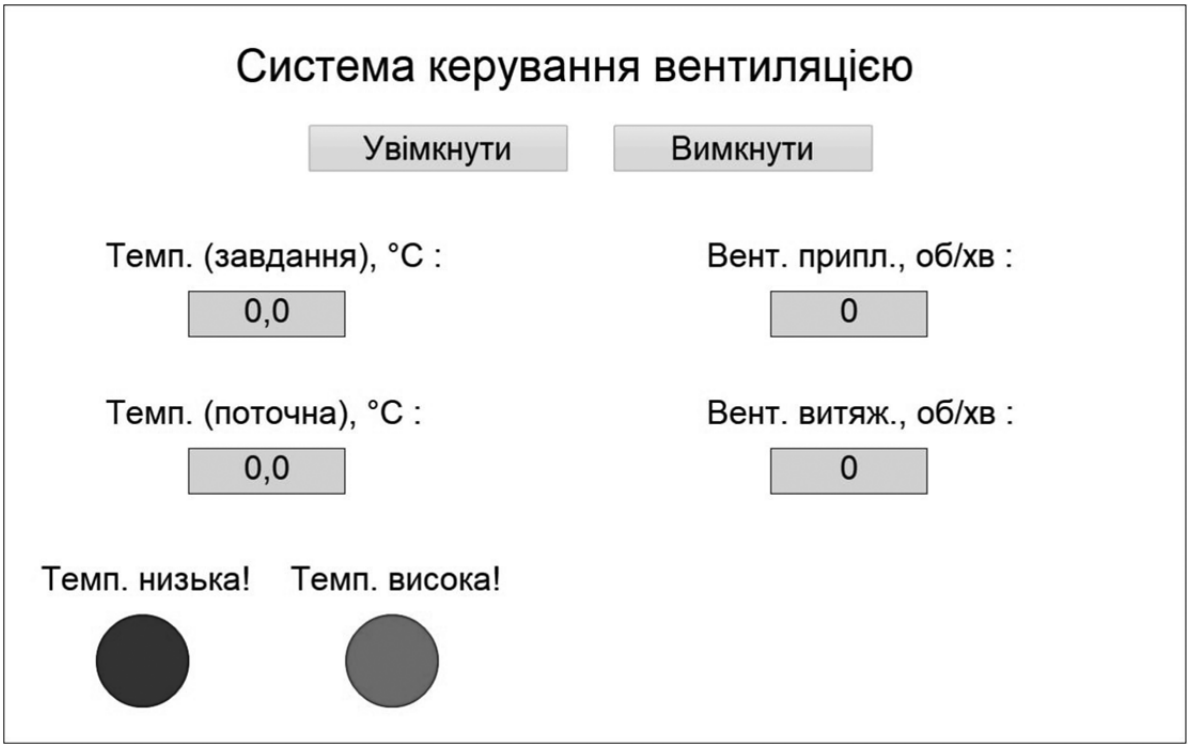


Рис. 9.2. Мнемосхема SCADA-системи для системи керування вентиляцією

Статичний текст “Система керування вентиляцією” системи керування вентиляцією задається за допомогою об’єкта мнемосхеми (далі – об. мн.) **Статичний текст** (Static Text).

Кнопки “Увімкнути” і “Вимкнути”, які дозволяють відповідно умикати і вимикати систему керування вентиляцією, відповідають об. мн. **Кнопка** (Button).

Статичний текст “Темп. (завдання), °C :” (об. мн. **Статичний текст** (Static Text)) і поле уведення “0,0” (об. мн. **Поле уведення/виведення** (I/O Field)) дозволяють задавати значення температури, яке необхідно підтримувати у виробничому приміщенні (наприклад, 22,5 °C).

Статичний текст “Темп. (поточна), °C :” (об. мн. **Статичний текст** (Static Text)) і поле виведення “0,0” (об. мн. **Поле уведення/виведення** (I/O Field)) дозволяють постійно відслідковувати значення температури у виробничому приміщенні і показувати його (наприклад, 18,0 °C).

Статичний текст “Темп. низька!” (об. мн. **Статичний текст** (Static Text)) і коло синього кольору (об. мн. **Коло** (Circle)) дозволяють в процесі роботи системи керування вентиляцією постійно відслідковувати значення температури у виробничому приміщенні і сигналізувати у випадку, якщо воно буде менше заданого значення (наприклад, коли воно менше заданого значення температури у виробничому приміщенні на > 3 °C, тобто менше $22{,}5\ °C - 3\ °C = 19{,}5\ °C$).

Статичний текст “Темп. висока!” (об. мн. **Статичний текст** (Static Text)) і коло червоного кольору (об. мн. **Коло** (Circle)) також дозволяють в процесі роботи системи керування вентиляцією постійно відслідковувати значення температури у виробничому приміщенні і сигналізувати у випадку, якщо воно буде більше заданого значення (наприклад, коли воно більше заданого значення температури у виробничому приміщенні на > 2 °C, тобто більше $22{,}5\ °C + 2\ °C = 24{,}5\ °C$).

Статичний текст “Вент. припл., об/хв :” (об. мн. **Статичний текст** (Static Text)) і поле виведення “0” (об. мн. **Поле уведення/виведення** (I/O Field)) дозволяють постійно відслідковувати значення обертів припливного вентилятора і показувати його (наприклад, 1350 об/хв).

Статичний текст “Вент. витяж., об/хв :” (об. мн. **Статичний текст** (Static Text)) і поле виведення “0” (об. мн. **Поле уведення/виведення** (I/O Field)) дозволяють постійно відслідковувати значення обертів витяжного вентилятора і показувати його (наприклад, 1200 об/хв).

9.2. SCADA-система для системи керування сушильною камерою

Схема автоматизації системи керування сушильною камерою наведена на рис. 9.3. Основною складовою такої системи є програмований контролер.

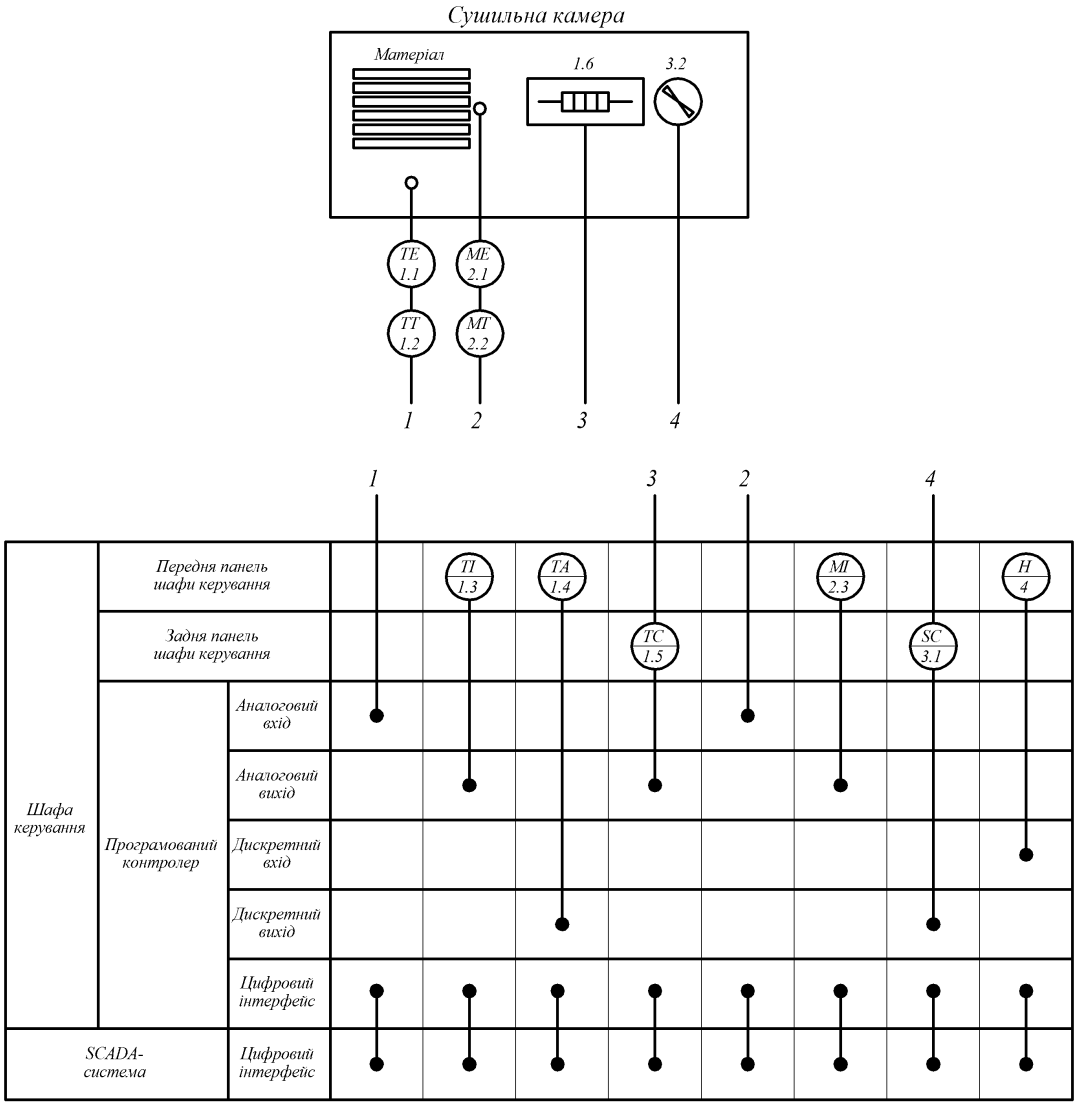


Рис. 9.3. Схема автоматизації системи керування сушильною камерою

В якості об’єкта керування системи керування сушильною камерою виступає сушильна камера, в якій необхідно підтримувати задану температуру повітря і забезпечувати переміщення повітря в залежності від відносної вологості матеріалу, який підсушується.

За допомогою датчиків температури 1.1 і відносної вологості 2.1 в сушильній камері виконується вимірювання відповідно температури повітря і відносної вологості матеріалу, за допомогою нормуючих перетворювачів відповідно 1.2 і 2.2 сигнали від чутливих елементів таких датчиків перетворюються в уніфіковані електричні неперервні сигнали, за допомогою

індикаторів 1.3 і 2.3 виконується показ значень температури і відносної вологості, за допомогою сигналізатора 1.4 – сигналізація про відхилення температури від заданого значення, за допомогою регулятора 1.5 – регулювання температури шляхом використання електричного теплообмінника 1.6. За допомогою регулятора 3.1 виконується регулювання обертів вентилятора 3.2. Загальне керування сушильною камерою виконується за допомогою пульта 4.

Температура повітря в сушильній камері може підтримуватися шляхом;

- зміни електричної потужності, яка підводиться до електричного теплообмінника 1.6;

- зміни продуктивності вентилятора 3.2, який впливає на кількість теплого повітря, що надходить в сушильну камеру.

Мнемосхема SCADA-системи для системи керування сушильною камерою наведена на рис. 9.4.

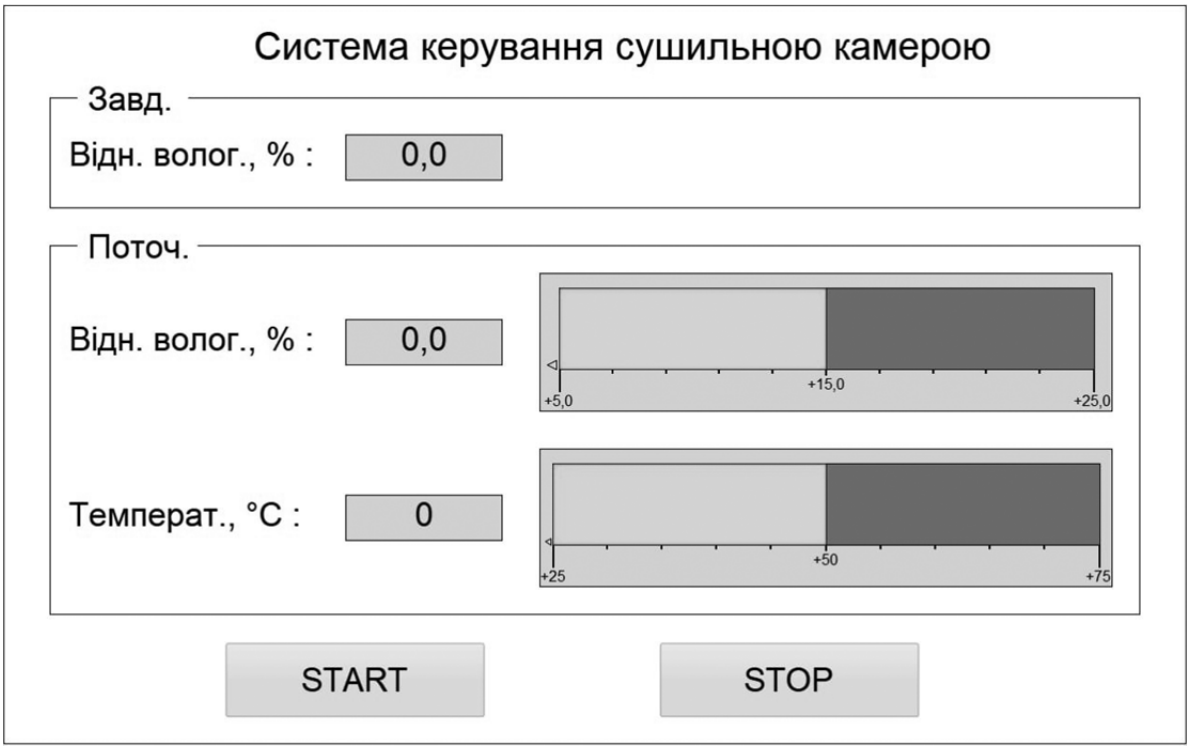


Рис. 9.4. Мнемосхема SCADA-системи для системи керування сушильною камерою

Статичний текст “Система керування сушильною камерою” системи керування сушильною камерою задається за допомогою об. мн. **Статичний текст** (Static Text).

Кнопки “START” і “STOP”, які дозволяють відповідно умикати і

вимикати систему керування сушильною камерою, відповідають об. мн. **Кнопка** (Button).

Статичний текст “Відн. волог., % :” (об. мн. **Статичний текст** (Static Text)) і поле уведення “0,0” (об. мн. **Поле уведення/виведення** (I/O Field)), які мають навколо себе обмежуючу рамку “Завд.”, дозволяють задавати значення відносної вологості, до якої необхідно підсушувати матеріал сушильної камери (наприклад, деревина для виготовлення меблів повинна мати відносну вологість (6 ... 10) %).

Статичний текст “Відн. волог., % :” (об. мн. **Статичний текст** (Static Text)), поле виведення “0,0” (об. мн. **Поле уведення/виведення** (I/O Field)) і індикатор рівня (об. мн. **Рівень** (Bar)), які мають навколо себе обмежуючу рамку “Поточ.”, дозволяють постійно відслідковувати значення відносної вологості матеріалу, який підсушується в сушильній камері, і показувати його (наприклад, 9,5 %). Мінімальне значення, яке може показувати індикатор рівня, становить 5,0 %, а максимальне 25,0 % – таким чином, показ значення відносної вологості забезпечується саме в “потрібному” діапазоні.

Статичний текст “Температ., °C :” (об. мн. **Статичний текст** (Static Text)), поле виведення “0” (об. мн. **Поле уведення/виведення** (I/O Field)) і індикатор рівня (об. мн. **Рівень** (Bar)), які мають навколо себе обмежуючу рамку “Поточ.”, дозволяють постійно відслідковувати значення температури в сушильній камері, і показувати його (наприклад, 65 °C). Мінімальне значення, яке може показувати індикатор рівня, становить 25 °C, а максимальне 75 °C – таким чином, показ значення відносної вологості забезпечується саме в “потрібному” діапазоні.

Припускається, що регулювання обертів вентилятора, який обдуває електричний теплообмінник, може виконуватися повністю без втручання оператора SCADA-системи – таким чином, об. мн. для керування таким вентилятором на мнемосхемі SCADA-системи відсутні.

Також припускається, що немає необхідності в постійному відслідковуванні значення температури в сушильній камері і сигналізувати у випадку, якщо воно буде менше або більше заданого значення, за допомогою об. мн. SCADA-системи – відповідно, на мнемосхемі SCADA-системи відсутні об. мн., які відповідають за таку сигналізацію.

9.3. SCADA-система для системи керування дозуванням

Схема автоматизації системи керування дозуванням наведена на рис. 9.5.

Основною складовою такої системи є програмований контролер.

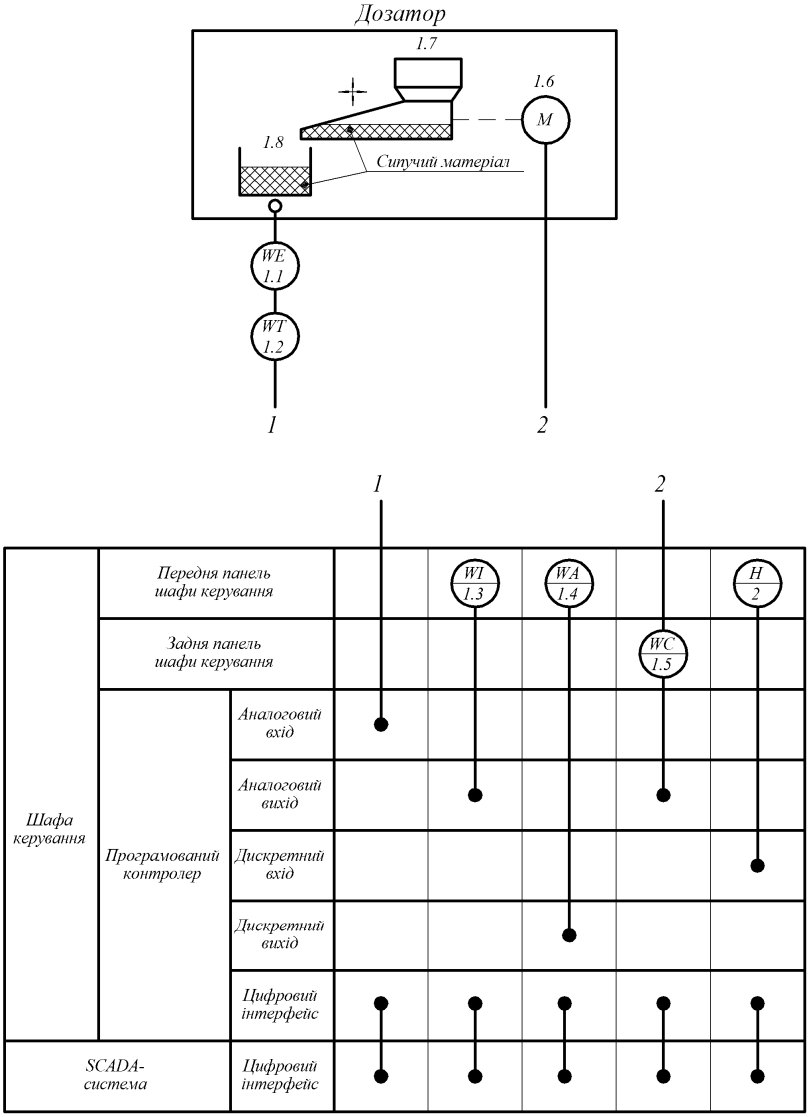


Рис. 9.5. Схема автоматизації системи керування дозуванням

В якості об’єкта керування системи керування дозуванням виступає дозатор, який повинен насипати з дозуючого бункера в накопичуючий бункер порції сипучого матеріалу заданої ваги.

За допомогою датчика ваги 1.1 в дозаторі виконується вимірювання ваги сипучого матеріалу, а за допомогою нормуючого перетворювача 1.2 сигнал від чутливого елемента такого датчика перетворюється в уніфікований електричний неперервний сигнал, за допомогою індикатора 1.3 виконується показ значення ваги, за допомогою сигналізатора 1.4 – сигналізація про відхилення ваги від заданого значення, за допомогою регулятора 1.5 – регулювання ваги шляхом використання віброприводу 1.6. Вібропривід 1.6 може впливати на інтенсивність вібрації дозуючого бункера 1.7, внаслідок чого

швидкість переміщення сипучого матеріалу під дією власної ваги з дозуючого бункера 1.7 в накопичуючий бункер 1.8 може змінюватися – при відсутності вібрації сипучий матеріал може взагалі не рухатися, а при максимальній вібрації швидкість переміщення сипучого матеріалу буде максимальною. Звичайно дозатор працює “порційному” режимі – при досягненні масою сипучого матеріалу в накопичуючому бункері 1.8 заданого значення такий бункер спустошується, і процес його наповнення циклічно повторюється. Загальне керування дозуванням виконується за допомогою пульта 2.

Швидкість дозування сипучого матеріалу в дозаторі може підтримуватися шляхом зміни амплітуди вібрації, яка підводиться до дозуючого бункера 1.7.

Мнемосхема SCADA-системи для системи керування дозуванням наведена на рис. 9.6.

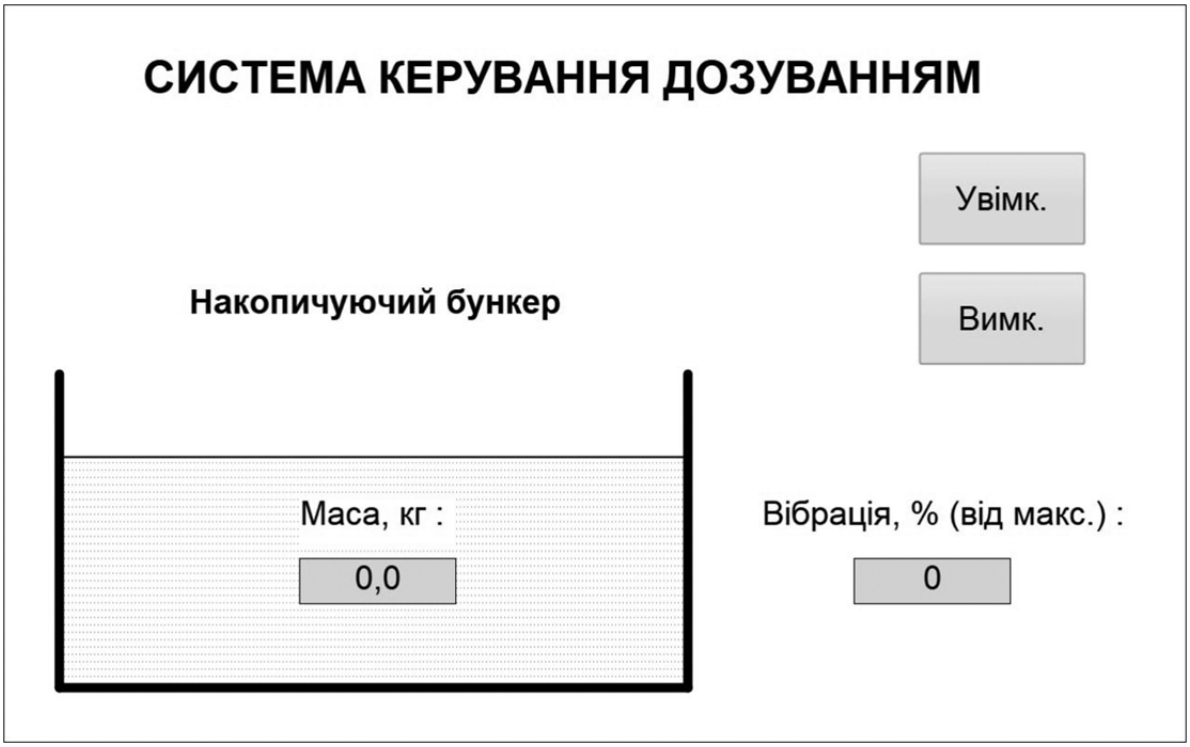


Рис. 9.6. Мнемосхема SCADA-системи для системи керування дозуванням

Статичний текст “Система керування дозуванням” системи керування дозуванням задається за допомогою об. мн. **Статичний** текст (Static Text).

Кнопки “Увімк.” і “Вимк.”, які дозволяють відповідно умикати і вимикати систему керування дозуванням, відповідають об. мн. **Кнопка** (Button).

Статичний текст “Накопичуючий бункер” задається за допомогою об. мн. **Статичний текст** (Static Text).

Лінії (об. мн. **Лінія** (Line)) і прямокутник (об. мн. **Прямокутник**

(Rectangle)) утворюють контури накопичуючого бункера.

Статичний текст “Маса, кг :” (об. мн. **Статичний текст** (Static Text)) і поле виведення “0,0” (об. мн. **Поле уведення/виведення** (I/O Field)) дозволяють постійно відслідковувати значення маси сипучого матеріалу, який дозується, в накопичуючому бункері і показувати його (наприклад, 18,0 кг).

Статичний текст “Вібрація, % (від макс.) :” (об. мн. **Статичний текст** (Static Text)) і поле виведення “0” (об. мн. **Поле уведення/виведення** (I/O Field)) дозволяють постійно відслідковувати значення інтенсивності вібрації дозуючого бункера відносно максимального значення такої інтенсивності, яке може бути досягнуто віброприводом, і показувати його (наприклад, 65 %).

Припускається, що регулювання інтенсивності вібрації дозуючого бункера, яке здійснюється віброприводом, може виконуватися повністю без втручання оператора SCADA-системи – таким чином, об. мн. для керування таким віброприводом на мнемосхемі SCADA-системи відсутні.

Також припускається, що немає необхідності в постійному відслідковуванні значення інтенсивності вібрації дозуючого бункера і сигналізувати у випадку, якщо воно буде менше або більше заданого значення, за допомогою об. мн. SCADA-системи – відповідно, на мнемосхемі SCADA-системи відсутні об. мн., які відповідають за таку сигналізацію.

9.4. SCADA-система для системи керування сховищем бітуму

Схема автоматизації системи керування сховищем бітуму наведена на рис. 9.7. Основною складовою такої системи є програмований контролер.

В якості об’єкта керування системи керування сховищем бітуму виступає сховище бітуму, в якому необхідно підтримувати задану температуру бітуму.

За допомогою датчика температури 1.1 в сховищі бітуму виконується вимірювання температури бітуму, за допомогою нормуючого перетворювача 1.2 сигнал від чутливого елемента такого датчика перетворюється в уніфікований електричний неперервний сигнал, за допомогою індикатора 1.3 виконується показ значення температури, за допомогою сигналізатора 1.4 – сигналізація про відхилення температури від заданого значення, за допомогою регулятора 1.5 – регулювання температури шляхом використання теплообмінника 1.8. Через теплообмінник 1.8 подається теплоносій, який має достатньо високу температуру (наприклад, пара або мастило). Зміна подачі теплоносія через теплообмінник 1.8 виконується за допомогою вентиля 1.7 (регулюючого органа), зміна положення якого виконується за допомогою

сервопривода 1.6 (виконавчого механізму), причому регулювальна характеристика вентиля 1.7 (регулюючого органа) загалом може бути не тільки лінійною, але й логарифмічною (“рівновідсотковою”), квадратичною (“швидкого відкриття”) або параболічною. Загальне керування сховищем бітуму виконується за допомогою пульта 2.

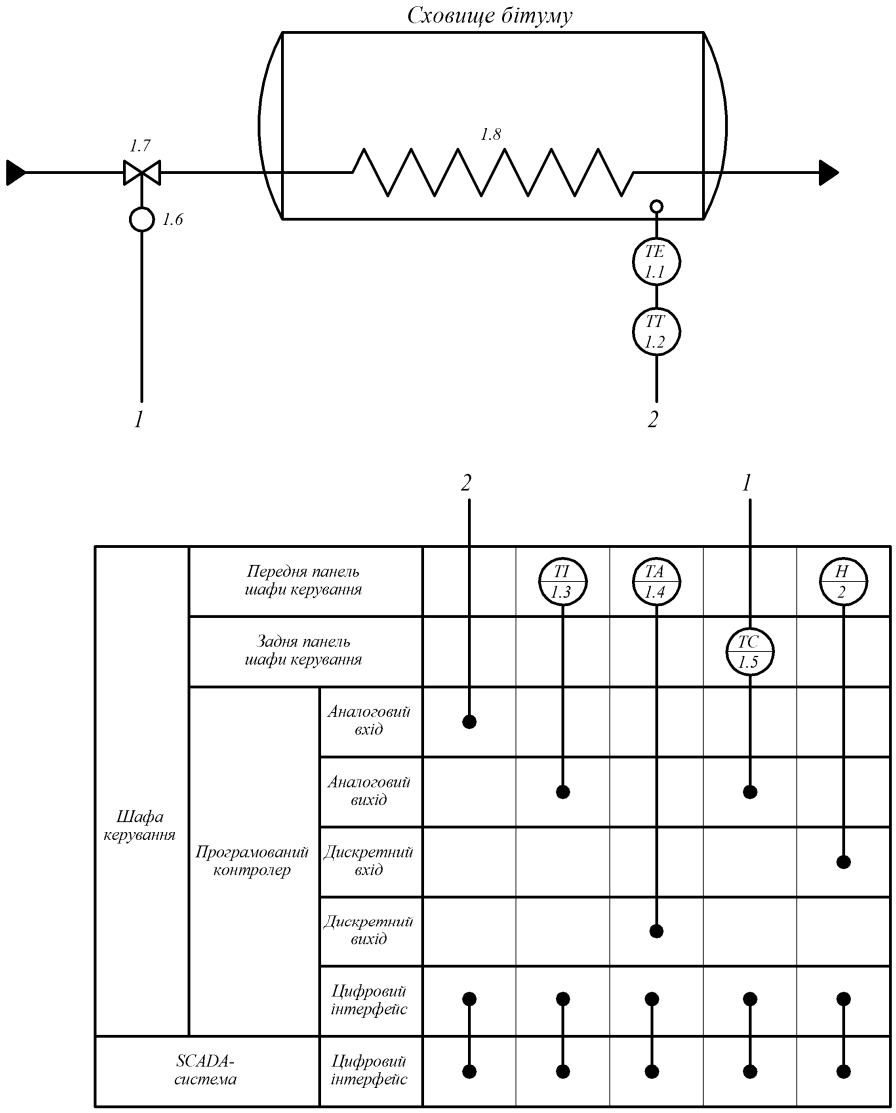


Рис. 9.7. Схема автоматизації системи керування сховищем бітуму

Температура бітуму в сховищі бітуму підтримується шляхом зміни витрати теплоносія, який проходить через теплообмінник 1.8.

Мнемосхема SCADA-системи для системи керування сховищем бітуму наведена на рис. 9.8.

Статичний текст “Система керування сховищем бітуму” системи керування сховищем бітуму задається за допомогою об. мн. **Статичний текст** (Static Text).

Рис. 9.8. Мнемосхема SCADA-системи для системи керування сховищем бітуму

Кнопки “Прац.” і “Не прац.”, які дозволяють відповідно умикати і вимикати систему керування сховищем бітуму, відповідають об. мн. **Кнопка** (Button).

Статичний текст “Сховище бітуму” задається за допомогою об. мн. **Статичний текст** (Static Text).

Статичний текст “Темп. (завд.), °C :” (об. мн. **Статичний текст** (Static Text)) і поле уведення “0” (об. мн. **Поле уведення/виведення** (I/O Field)) дозволяють задавати значення температури, яке необхідно підтримувати в сховищі бітуму (наприклад, 135 °C).

Статичний текст “Темп. (поточ.), °C :” (об. мн. **Статичний текст** (Static Text)) і поле виведення “0” (об. мн. **Поле уведення/виведення** (I/O Field)) дозволяють постійно відслідковувати значення температури в сховищі бітуму і показувати його (наприклад, 148 °C).

Статичний текст “Відкр./закр., % (від макс.) :” (об. мн. **Статичний текст** (Static Text)) і поле виведення “0” (об. мн. **Поле уведення/виведення** (I/O Field)) дозволяють постійно відслідковувати значення ступеня відриття/закриття вентиля (регулюючого органа) відносно максимального значення такого ступеня, яке може бути досягнуто сервоприводом (виконавчим механізмом), і показувати його (наприклад, 72 %).

Також припускається, що немає необхідності в постійному відслідковуванні значення температури в сховищі бітуму і сигналізувати у випадку, якщо воно буде менше або більше заданого значення, за допомогою об. мн. SCADA-системи – відповідно, на мнемосхемі SCADA-системи відсутні об. мн., які відповідають за таку сигналізацію.

9.5. SCADA-система для системи керування дуговою піччю

Схема автоматизації системи керування дуговою піччю наведена на рис. 9.9. Основною складовою такої системи є програмований контролер.

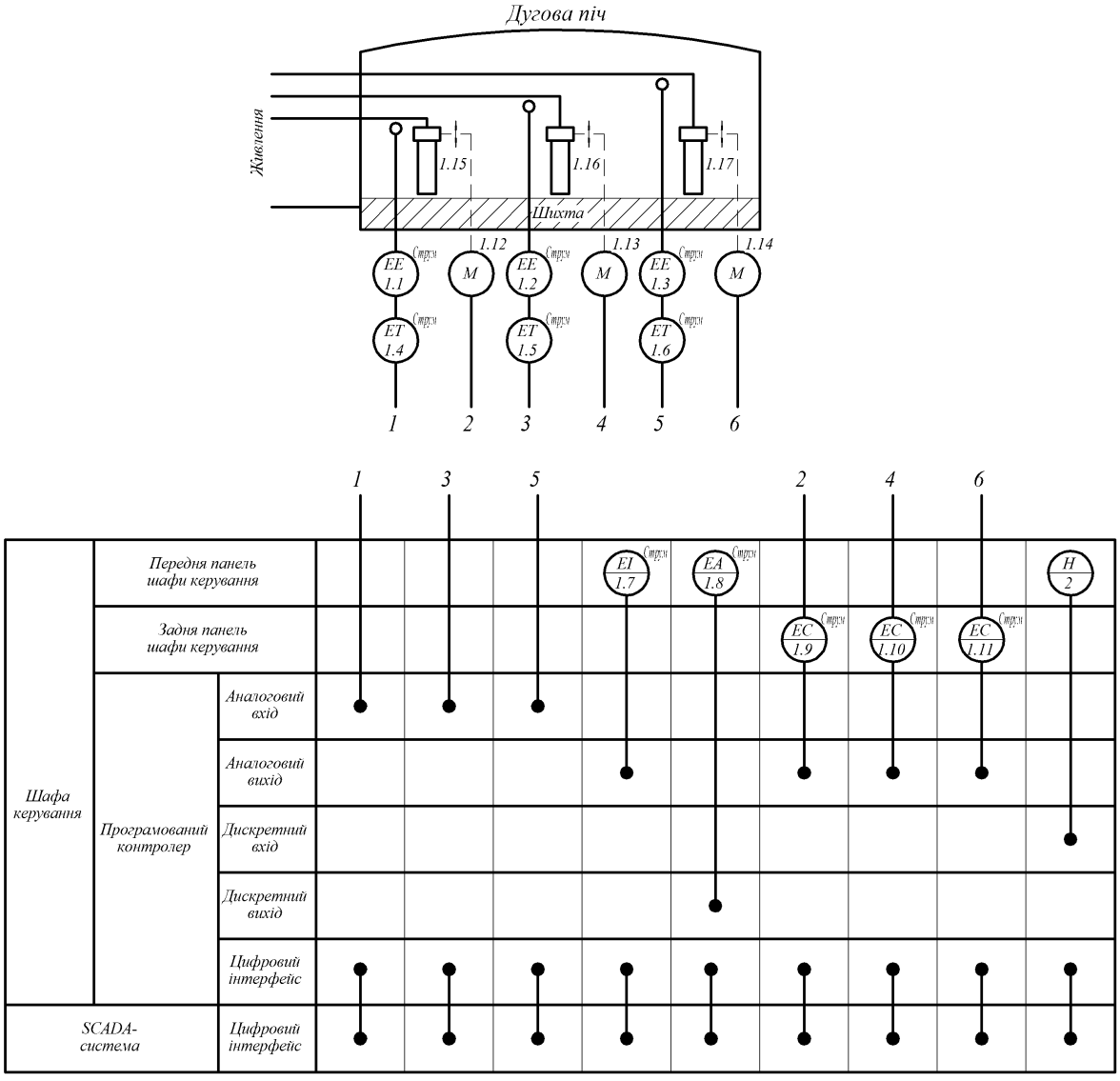


Рис. 9.9. Схема автоматизації системи керування дуговою піччю

В якості об’єкта керування системи керування дуговою піччю виступає дугова піч, в якій необхідно підтримувати задані струми через кілька

електродів.

За допомогою датчиків струму 1.1, 1.2 і 1.3 в дуговій печі виконується вимірювання струму через відповідно електроди 1.15, 1.16 і 1.17, за допомогою відповідно нормуючих перетворювачів 1.4, 1.5 і 1.6 сигнали від чутливих елементів таких датчиків перетворюються в уніфіковані електричний неперервні сигнали, за допомогою індикатора 1.7 виконується показ значення струму, за допомогою сигналізатора 1.8 – сигналізація про відхилення струму від заданого значення, за допомогою відповідно регуляторів 1.9, 1.10 і 1.11 – регулювання струму шляхом використання відповідно електроприводів 1.12, 1.13 і 1.14, які можуть переміщувати відповідно електроди 1.15, 1.16 і 1.17, Використання саме трьох електродів пов'язано з тим, шо для електричного живлення дугової печі використовується саме трьохфазна електрична мережа. Загальне керування дуговою піччю виконується за допомогою пульта 2.

Струм через електроди в дуговій печі підтримується шляхом:

- переміщення електрода 1.15 відносно шихти за допомогою електроприводу 1.12;
- переміщення електрода 1.16 відносно шихти за допомогою електроприводу 1.13;
- переміщення електрода 1.17 відносно шихти за допомогою електроприводу 1.14.

Мнемосхема SCADA-системи для системи керування дуговою піччю наведена на рис. 9.10.

Статичний текст "Система керування дуговою піччю" системи керування дуговою піччю задається за допомогою об. мн. **Статичний текст** (Static Text).

Кнопки "On" і "Off", які дозволяють відповідно умикати і вимикати систему керування дуговою піччю, відповідають об. мн. **Кнопка** (Button).

Статичний текст "Дугова піч" задається за допомогою об. мн. **Статичний текст** (Static Text).

Статичні тексти "Струм, кА (електр. № 1) :", "Струм, кА (електр. № 2) :" і "Струм, кА (електр. № 3) :" (об. мн. **Статичний текст** (Static Text)) та відповідні їм поля уведення "0,0", "0,0" і "0,0" (об. мн. **Поле уведення/виведення** (I/O Field)), які мають навколо себе обмежуючу рамку "Завд.", дозволяють задавати значення струмів через відповідні їм електроди в дуговій печі (наприклад, 7,5 кА, 7,5 кА і 7,5 кА).

Статичні тексти "Струм, кА (електр. № 1) :", "Струм, кА (електр. № 2) :" і "Струм, кА (електр. № 3) :" (об. мн. **Статичний текст** (Static Text)) та

відповідні їм поля виведення “0,0”, “0,0” і “0,0” (об. мн. **Поле уведення/виведення** (I/O Field)), які мають навколо себе обмежуючу рамку “Поточ.”, дозволяють постійно відслідковувати значення струмів через відповідні їм електроди в дуговій печі і показувати його (наприклад, 7,7 кА, 7,8 кА і 7,6 кА).

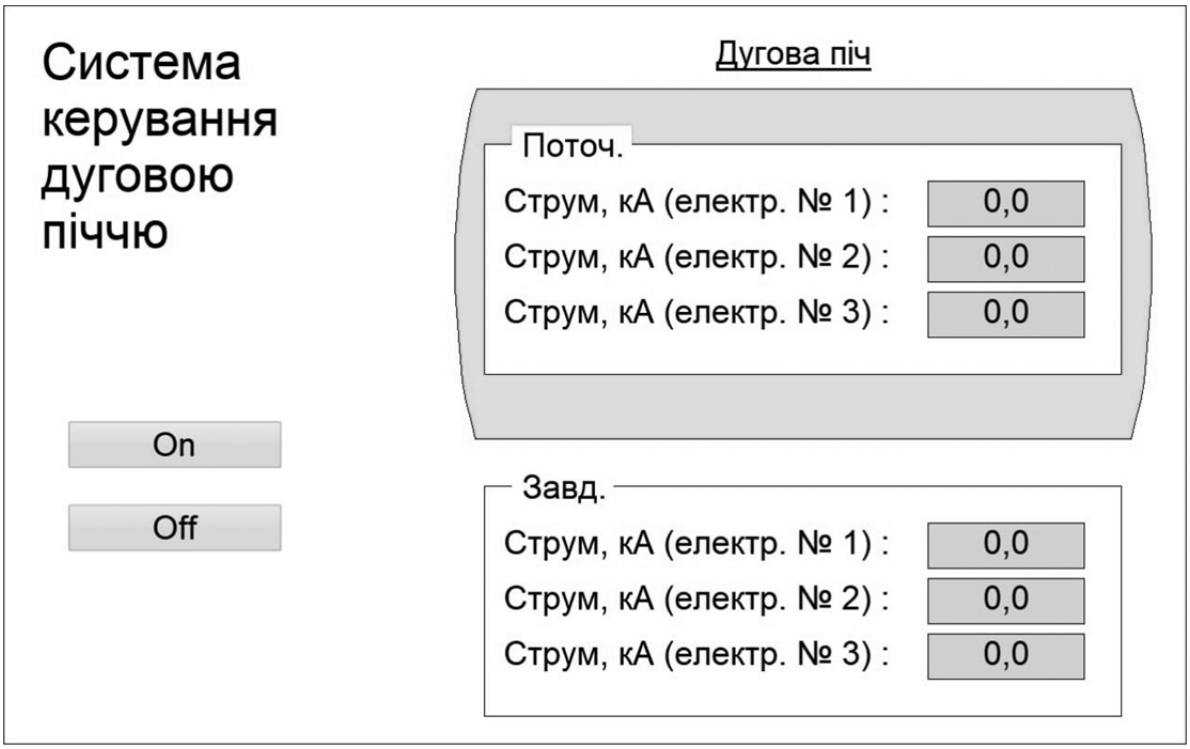


Рис. 9.10. Мнемосхема SCADA-системи для системи керування дуговою піччю

Також припускається, що немає необхідності в постійному відслідковуванні значень струмів через електроди в дуговій печі і сигналізувати у випадку, якщо вони будуть менше або більше задантх значень, за допомогою об. мн. SCADA-системи – відповідно, на мнемосхемі SCADA-системи відсутні об. мн., які відповідають за таку сигналізацію.

9.6. Питання для самоперевірки

1. Як при розробці мнемосхем SCADA-систем систем керування можна збільшити їх коефіцієнти інформативності і заповнюваності?

2. За яким принципом при розробці мнемосхем SCADA-систем систем керування передбачається групування елементів мнемосхем по їх функціям і/або по об’єктам автоматизації, до яких вони відносяться?

3. За яким принципом при розробці мнемосхем SCADA-систем систем

керування передбачається групування елементів мнемосхем по їх важливості при виконання тієї або іншої дії?

4. За яким принципом при розробці мнемосхем SCADA-систем систем керування передбачається групування елементів мнемосхем по їх конструктивним особливостям (наприклад, по розміру шкал, по точності вимірювання і по швидкодії)?

5. За яким принципом при розробці мнемосхем SCADA-систем систем керування передбачається групування елементів мнемосхем по послідовності рухів при виконанні тієї або іншої дії?

6. За яким принципом при розробці мнемосхем SCADA-систем систем керування передбачається групування елементів мнемосхем по частоті їх використання?

7. За допомогою яких об’єктів **Дизайнера графіки** (Graphics Designer) можна на мнемосхемах SCADA-систем систем керування відображати незмінні дані (наприклад, назву технологічної операції)?

8. За допомогою яких об’єктів **Дизайнера графіки** (Graphics Designer) на мнемосхемах SCADA-систем систем керування можна “ініціювати” виконання певних дій (наприклад, увімкнення електричного двигуна)?

9. За допомогою яких об’єктів **Дизайнера графіки** (Graphics Designer) на мнемосхемах SCADA-систем систем керування можна уводити дані (наприклад, значення тиску, який необхідно підтримувати)?

10. За допомогою яких об’єктів **Дизайнера графіки** (Graphics Designer) на мнемосхемах SCADA-систем систем керування можна виводити дані (наприклад, значення напруги живлення)?

11. За допомогою яких об’єктів **Дизайнера графіки** (Graphics Designer) на мнемосхемах SCADA-систем систем керування можна відображати часові залежності даних (наприклад, значення вологості у вигляді рівня, який з плином часу змінюється)?

9.7. Рекомендована література

Базова література: [1–6].

Додаткова література: [1–6].

СПИСОК ЛІТЕРАТУРИ

Базова література

1. Бочков В.М. Обладнання автоматизованого виробництва / В.М. Бочков, Р.І. Сілін. – Львів: Львівська політехніка, 2015. – 404 с. – ISBN 978–617–607–780–0.
2. Васильківський І.С. Виконавчі пристрої систем автоматизації. Навчальний посібник / І.С. Васильківський, В.О. Фединець, Я.П. Юсик. – Львів: Львівська політехніка, 2020. – 220 с. – ISBN 978–966–941–543–1.
3. Ладанюк А.П. Автоматизація технологічних процесів і виробництв харчової промисловості: Підручник / А.П. Ладанюк, В.Г. Трегуб, І.В. Ельперін, В.Д. Цюцюра. – К.: Аграрна освіта, 2001. – 224 с. – ISBN 966–95661–2–6.
4. Пушкар М.С. Проектування систем автоматизації: Навч. посібник / М.С. Пушкар, С.М. Проценко. – Д.: Національний гірничий університет, 2013. – 268 с. – ISBN 978–966–350–423–0.
5. Пупена О.М. Розроблення людино-машинних інтерфейсів та систем збирання даних з використанням програмних засобів SCADA/HMI: Навч. посіб. – Київ: Видавництво Ліра-К, 2020. – 594 с. – ISBN 978–617–7910–07–6.
6. Трегуб В.Г. Проектування систем автоматизації: Навч. посібник. – К.: Видавництво Ліра-К, 2017. – 344 с. – ISBN 978–966–2609–58–5.

Додаткова література

1. Грицунов О.В. Інформаційні системи та технології. Навчальний посібник. – Х.: ХНАМГ, 2010. – 222 с. – ISBN 978–966–695–195–6.
2. Ельперін І.В. Промислові контролери: Навчальний посібник / І.В. Ельперін – К.: НУХТ, 2003. – 320 с. – ISBN 966–612–024–0.
3. Лисаченко І.Г. Програмне забезпечення комп’ютерно-інтегрованих систем управління хіміко-технологічними процесами: Навчально-методичний посібник / І.Г. Лисаченко. – Х.: НТУ “ХПІ”, 2012. – 112 с. – ISBN 000-000-000-000-0.
4. Павлюк О.М. Основи теорії надійності технічних систем: Навчальний посібник / О.М. Павлюк, М.О. Медиковський, Н.К. Лиса, І.В. Ізонін. – Львів: Видавництво Львівської політехніки, 2021. – 208 с. – ISBN 978–966–941–659–9.

5. Савицький В. Технічні засоби автоматизації / В. Савицький, Р. Федоришин. – Львів: Львівська політехніка, 2018. – 292 с. – ISBN 978–966–941–182–2.
6. Синєглазов В.М. Автоматизація технологічних процесів / В.М. Синєглазов, І.Ю. Сергеєв. – К.: НАУ, 2015. – 444 с. – ISBN 978–966–598–928–8.