

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

О.М. Пупена, І.В. Ельперін,
Н.М. Луцька, А.П. Ладанюк

ПРОМИСЛОВІ МЕРЕЖІ ТА ІНТЕГРАЦІЙНІ ТЕХНОЛОГІЇ В АВТОМАТИЗОВАНИХ СИСТЕМАХ

Навчальний посібник

*Рекомендовано
Міністерством освіти і науки України
Як навчальний посібник для студентів
вищих навчальних закладів*



Київ-2011

Автори:

*О.М. Пупена, І.В. Ельперін,
Н.М. Луцька, А.П. Ладанюк*

*Гриф надано Міністерством освіти і науки України
як навчальний посібник для студентів, які навчаються за напрямом
«Автоматизація та комп'ютерно-інтегровані технології»
(Лист № 1/11-8266 від 30.08.2010 р.)*

П 88 Промислові мережі та інтеграційні технології в автоматизованих системах: Навчальний посібник. — К. : Вид-во «Ліра-К», 2011. — 552 с.
ISBN 978–966–2174–13–7

У навчальному посібнику розглянуті принципи функціонування промислових мереж та інтеграційних технологій, що використовуються в інтегрованих автоматизованих системах.

Посібник складається з чотирьох частин. У *першій частині* висвітлюються загальні питання про місце мережних технологій в інтегрованих автоматизованих системах, фундаментальні принципи функціонування всіх промислових мереж, короткий огляд найбільш популярних в Україні промислових мереж, детально описані особливості реалізації фізичного рівня промислових мереж а також функціонування модемного та бездротового зв'язку. У *другій частині* детально висвітлюються принципи функціонування та особливості проектування промислових мереж AS-I, MODBUS, PROFIBUS, CAN, CANOPEN та мереж на базі Industrial Ethernet, а також описані стандартні технології управління електроприводами через промислові мережі. *Третя частина* присвячена відкритим технологіям міжпрограмної взаємодії DDE/NetDDE, COM/DCOM, WEB, OPC, ODBC, OLEDB, та показано їх використання в інтегрованих автоматизованих системах. *Остання частина* присвячена проектуванню інтегрованих автоматизованих систем, що базуються на використанні промислових мереж та технологій міжпрограмної взаємодії. Весь теоретичний матеріал закріплюється робочими прикладами та питаннями для самостійного контролю.

Навчальний посібник може бути використаний при вивченні курсів циклу професійної і практичної підготовки рівня бакалавра, спеціаліста та при виконанні курсового і дипломного проектування для студентів напрямку «Автоматизація та комп'ютерно-інтегровані технології». Посібник також рекомендується для молодих спеціалістів, що працюють в області промислової автоматизації.

УДК 681.51
ББК 32.973.202

ISBN 978–966–2174–13–7

© Пупена О. М., Ельперін І. В.,
Луцька Н. М., Ладанюк А. П., 2011
© Видавництво «Ліра-К», 2011

Зміст

Вступ	10
1. МІСЦЕ ТА РОЛЬ МЕРЕЖНИХ ТЕХНОЛОГІЙ В ІНТЕГРОВАНІХ АВТОМАТИЗОВАНИХ СИСТЕМАХ УПРАВЛІННЯ	14
1.1. Інтегровані автоматизовані системи управління	14
1.2. Функціональна інтеграція	17
1.3. Інформаційна інтеграція	18
1.4. Технічна та програмна інтеграція	18
1.4.1. Засоби людино-машинного інтерфейсу	19
1.4.2. Контролери	20
1.4.3. Периферійні засоби: перетворювачі та виконавчі механізми	22
1.4.4. Програматори/конфігуратори	23
1.4.5. Програмно-технічні засоби рівня АСУП	24
1.5. Обчислювальні мережі в ієрархії ІАСУ	26
2. ЗАГАЛЬНА ХАРАКТЕРИСТИКА ПРОМИСЛОВИХ МЕРЕЖ	28
2.1. Властивості промислових мереж	28
2.1.1. Визначення промислової мережі та їх градація	28
2.1.2. Функціональне призначення промислових мереж	29
2.1.3. Обмін даними	31
2.1.4. Загальні вимоги до промислових мереж	34
2.2. Історія розвитку та стандартизації промислових мереж	38
2.3. Промислові мережі в контексті моделі ISO OSI	41
2.4. Основні робочі характеристики промислових мереж	46
2.4.1. Прикладні сервіси, інтерфейс прикладного рівня	47
2.4.2. Забезпечення каналом зв'язку між вузлами та методи доступу	53
2.4.3. Формування кадру та контроль за помилками	61
2.4.4. Топологія мережі та сегментація	62
2.4.5. Фізичні інтерфейси передачі даних	64
2.4.6. Маршрутизація	66
2.4.7. Транспортування даних	70
2.4.8. Підсумкова таблиця з основними робочими характеристиками	71

2.5. Короткий огляд промислових мереж	73
2.5.1. Мережі MODBUS RTU/ASCII та MODBUS TCP/IP	73
2.5.2. Мережа World-FIP	75
2.5.3. Мережа Foundation Fieldbus (FF)	79
2.5.4. CAN та CANOpen	87
2.5.5. Мережа LonWorks	90
2.5.6. Мережі PROFIBUS	98
2.5.7. HART-протокол	101
2.5.8. Мережа AS-i	105
2.5.9. Мережа INTERBUS	106
2.5.10. Мережі CIP: DeviceNet, ControlNet, Ethernet/IP та CompoNet	115
2.5.11. Мережі CC-Link	130
2.6. Вибір промислової мережі	136
3. РЕАЛІЗАЦІЯ ФІЗИЧНОГО РІВНЯ	141
3.1. Організація каналу зв'язку	141
3.2. Використання металевих кабелів	144
3.2.1. Фізичні властивості кабельної лінії передачі	145
3.2.2. Електричні характеристики кабельної лінії передачі	149
3.2.3. Відбиття сигналів та способи їх подолання	152
3.2.4. Типи електричних кабелів	153
3.2.5. Електричні шуми, завади та боротьба з ними	153
3.2.6. Заземлення екранів кабелів	156
3.2.7. Основні рекомендації до вибору, прокладки та монтажу електричних кабелів для промислових мереж	159
3.3. Передача напругою та струмом	161
3.3.1. Використання напруги	161
3.3.2. Використання струму	162
3.4. Використання оптоволоконного кабелю	162
3.5. Використання бездротового зв'язку	164
3.6. Кодування інформації	165
3.6.1. Безпосередній спосіб передачі цифрових даних	165
3.6.2. Модуляція несучої (carrier modulation)	167
3.7. Синхронна та асинхронна передача	168
3.7.1. Необхідність синхронізації	168
3.7.2. Символьна передача	169
3.7.3. Використання схем UART	170
3.8. Стандартні послідовні інтерфейси	171
3.9. Інтерфейс RS-232	171
3.9.1. Електричні характеристики	172
3.9.2. Типи роз'ємів	173
3.9.3. Призначення ліній інтерфейсу RS-232 C	174
3.9.4. Управління потоком даних	175
3.10. Інтерфейс RS-422A (EIA-422A)	177

3.11. Інтерфейс RS-485 (EIA/TIA-485)	179
3.11.1. Схеми з'єднання	180
3.11.2. Схеми узгодження	181
3.11.3. Забезпечення захисного зміщення	182
3.12. Інтерфейс «струмова петля» 20 мА (Current Loop, ИПС)	183
3.13. Порівняльна характеристика стандартних послідовних інтерфейсів	186
3.14. Використання адаптерів-перетворювачів інтерфейсів	187
3.14.1. Типи адаптерів-перетворювачів	187
3.14.2. Управління адаптерами RS-232<->RS-485 з боку RS-232	188
3.14.3. Автоматичне управління адаптером RS-232<->RS-485	189
4. ТЕХНОЛОГІЇ МОДЕМНОГО ТА БЕЗДРОТОВОГО ЗВ'ЯЗКУ	192
4.1. Символьний спосіб обміну	192
4.1.1. Способи кодування символів	193
4.1.2. Принципи символьного обміну ASCII	195
4.2. Модемний обмін	197
4.3. Бездротовий спосіб обміну	202
4.3.1. Питання стандартизації	203
4.3.2. Класифікація бездротових способів обміну	204
4.3.3. Стандарти IEEE 802.15	205
4.3.4. Стандарт IEEE 802.11	206
4.3.5. Стандарт IEEE 802.16	208
4.3.6. Технології бездротового зв'язку в промисловості	209
4.4. Особливості бездротових модемів з використанням мобільного зв'язку	215
4.4.1. Коротка характеристика мобільного зв'язку в Україні	215
4.4.2. Основні стандарти мобільного зв'язку	215
5. ІНТЕРФЕЙС AS-i	222
5.1. Загальні відомості	222
5.2. Реалізація фізичного рівня	223
5.3. Принципи функціонування	226
5.4. Етапи роботи шини AS-i	229
5.5. Формат кадру AS-i	230
5.6. Швидкість відновлення даних	233
5.7. Профілювання Ведучих та Ведених	233
5.8. Автоматична адресація	234
5.9. AS-i V2.1 та V3	235
5.10. AS-i профілі	236
6. МЕРЕЖІ MODBUS	238
6.1. Мережі MODBUS у контексті моделі OSI	238
6.2. Реалізація MODBUS на прикладному рівні	239
6.2.1. Формат MODBUS PDU	239
6.2.2. Формат основних функцій	240
6.2.3. Адресна модель MODBUS та доступ до даних	245

6.3. MODBUS Serial	248
6.3.1. Канальний рівень	249
6.3.2. MODBUS RTU	250
6.3.3. MODBUS ASCII	251
6.3.4. Реалізація фізичного рівня для MODBUS Serial	257
6.4. MODBUS TCP/IP	263
6.4.1. Комунікаційна архітектура MODBUS TCP/IP	263
6.4.2. Особливості реалізації протоколу	264
6.5. Рекомендації до проектування мереж MODBUS RTU/ASCII та MODBUS TCP/IP	271
6.5.1. Послідовність розробки мереж MODBUS RTU/ASCII	272
6.5.2. Перенесення клієнтських запитів у мережах MODBUS RTU/ASCII	274
6.5.3. Послідовність розробки мереж MODBUS TCP/IP	276
7. МЕРЕЖІ PROFIBUS	278
7.1. Мережі PROFIBUS у контексті моделі ISO та MEK	278
7.2. Реалізація фізичного рівня PROFIBUS DP	280
7.3. Реалізація фізичного рівня PROFIBUS PA (MBP)	287
7.4. Реалізація канального рівня	290
7.5. Базові функції обміну даними між вузлами в PROFIBUS DP (DP-V0)	291
7.5.1. Функція циклічного обміну даними процесу	292
7.5.2. Функції (сервіси) базових типів вузлів	293
7.5.3. Структура DP-циклу	294
7.5.4. Конфігурування та ініціалізація мережі	295
7.6. Розширені функції обміну даними між вузлами в PROFIBUS DP (DP-V1, DP-V2)	303
7.6.1. Версії DP-V1 та DP-V2	303
7.6.2. Ациклічний обмін даними процесу	304
7.6.3. Широкомовний обмін даними процесу	304
7.6.4. Ізохронний режим	304
7.7. Профілі в PROFIBUS DP	305
7.7.1. Технічна структура PROFIBUS	305
7.7.2. Загальні прикладні профілі	307
7.7.3. Специфічні прикладні профілі	309
7.7.4. Використання GSD для інтеграції пристроїв у мережу	310
8. CAN ПРОТОКОЛ	313
8.1. CAN-стандарти та їх опис у контексті моделі OSI	314
8.2. Організація канального рівня	315
8.2.1. Комунікаційні сервіси	315
8.2.2. Формати кадрів обміну даними	316
8.2.3. Обробка помилок та боротьба з дефектами	318
8.3. Реалізація фізичного рівня	321
8.4. CAN-сумісні стандарти	322
8.4.1. Стандарт ISO 11898	322
8.4.2. Протоколи прикладного рівня для мереж CAN	325
8.5. Реалізація CAN	325

9. CANOpen	329
9.1. CANOpen у контексті моделі OSI	329
9.2. Особливості реалізації фізичного рівня	330
9.3. Базові принципи функціонування	339
9.3.1. Модель обміну	339
9.3.2. Комунікаційні сервіси	340
9.3.3. Основи функціонування сервісу PDO	341
9.3.4. Базові концепції конфігурування та адміністрування мережі (NMT-Сервіси)	346
9.3.5. Схема розподілення Ідентифікаторів за замовченням	354
9.4. Робота зі Словником Об'єктів	362
9.4.1. Структура словника об'єктів	362
9.4.2. Основи функціонування сервісу SDO	364
9.4.3. Типи передачі POO та об'єкт SYNC	365
9.4.4. Об'єкт Time Stamp	368
9.4.5. Об'єкт Emergency	368
9.4.6. Діагностичні сервіси NodeGuard та Heartbeat	369
9.5. Специфікації профілів пристроїв CANOpen	370
9.5.1. Профілі пристроїв у CANOpen	370
9.5.2. Профіль модулів вводу/виводу	371
10. ПРОМИСЛОВИЙ ETHERNET	377
10.1. Мережі Ethernet	378
10.1.1. Загальні поняття	378
10.1.2. Структура кадру	382
10.2. Протоколи рівня IP	384
10.2.1. Протокол IP	384
10.2.2. Протокол ARP	384
10.2.3. Протоколи BOOTP та DHCP	385
10.2.4. Протоколи IGMP та IGMP-snooping	387
10.2.5. Методика NAT-трансляції	388
10.3. Протоколи транспортного рівня	389
10.3.1. Протоколи UDP та TCP	389
10.3.2. Модель сокетів	390
10.4. Промисловий Ethernet	392
10.4.1. Кабельні системи промислового виконання	392
10.4.2. Real Time Ethernet	393
10.4.3. Рішення RTE над TCP/IP	396
10.4.4. Рішення RTE над Ethernet	398
10.4.5. Рішення RTE з модифікацією Ethernet	401
11. КОМУНІКАЦІЙНА АРХІТЕКТУРА ДЛЯ ЕЛЕКТРОПРИВОДІВ	407
11.1. Способи інтеграції пристроїв PDS в автоматизовані системи управління	407
11.1.1. Класичний підхід	407

11.1.2.	Переваги використання промислових мереж	407
11.1.3.	Проблеми використання промислових мереж	408
11.2.	Стандарт на універсальний інтерфейс PDS	408
11.2.1.	Загальні підходи до стандартизації інтерфейсу PDS	408
11.2.2.	Функціональні елементи Логічного Приводу	411
11.2.3.	Прикладні режими PDS	415
11.2.4.	Структура стандартів IEC 61800-7	419
11.3.	Профіль PROFIdrive	420
11.3.1.	Функціональні елементи PROFIdrive	420
11.3.2.	Загальний автомат станів для приводів PROFIdrive	421
11.3.3.	Операційна робота приводів PROFIdrive в режимі управління швидкістю	423
11.4.	Профіль CiA402	429
11.4.1.	Функціональні елементи CiA402	429
11.4.2.	Операційна робота приводів CiA402 в режимі управління швидкістю	433
12.	ВІДКРИТІ ТЕХНОЛОГІЇ ПРОГРАМНОЇ ІНТЕГРАЦІЇ В СЕРЕДОВИЩІ WINDOWS	438
12.1.	Проблеми програмної інтеграції в інтегрованих автоматизованих системах	438
12.2.	Функціонування ОС Windows	439
12.2.1.	Основи функціонування	439
12.2.2.	Архітектура Windows	441
12.3.	Інтерфейс програми користувача до ОС Windows	442
12.3.1.	Win32 API	442
12.3.2.	Проблеми доступу до даних іншого Процесу	445
12.4.	Технології DDE та NetDDE	446
12.4.1.	Локальний обмін через DDE	446
12.4.2.	Обмін через NetDDE в мережі	447
12.5.	Технології COM/DCOM	449
12.5.1.	Доступ до Процесів через COM	449
12.5.2.	Використання OLE та ActiveX	452
12.6.	Web Технології	454
13.	ТЕХНОЛОГІЯ OPC	457
13.1.	Загальні концепції	457
13.1.1.	Передумови виникнення	457
13.1.2.	Стандарти OPC	459
13.1.3.	Функціонування OPC з точки зору інтегратора	461
13.2.	Принципи функціонування OPC DA	466
13.2.1.	OPC модель взаємодії	466
13.2.1.	Механізми читання та запису даних процесу	468
13.2.3.	Ідентифікатори ItemID	470
13.2.4.	Робота OPC Клієнта з віддаленими OPC Серверами	474

13.3. Типи OPC DA інтерфейсів	475
13.3.1. Загальний огляд типів інтерфейсів	475
13.3.2. Об'єктна модель інтерфейсу OPC Automation	476
13.3.3. Синтаксис основних методів, властивостей та подій об'єктів бібліотеки OPCAutomation	478
13.4. Область застосування технології OPC	483
14. СТАНДАРТНІ ТЕХНОЛОГІЇ ДОСТУПУ ДО БАЗ ДАНИХ	486
14.1. Проблеми доступу до баз даних	486
14.2. Мова SQL	487
14.3. ODBC та DAO	488
14.4. OLE DB, ADO та ADO.NET	491
15. ПРОЕКТУВАННЯ ІНТЕГРОВАНИХ АВТОМАТИЗОВАНИХ СИСТЕМ	495
15.1. Загальні підходи до проектування автоматизованих систем	495
15.1.1. ІАС як клас автоматизованих систем	495
15.1.2. Стадії створення автоматизованих систем	495
15.1.3. Розробка документації на стадіях створення автоматизованих систем	500
15.2. Загальні підходи до проектування інтегрованих автоматизованих систем	507
15.2.1. Життєвий цикл ІАС	507
15.2.2. Дослідження об'єкта та формування вимог до ІАС	510
15.2.3. Розробка концепції ІАС	511
15.2.4. Технічне завдання на ІАС та АС	511
15.2.5. Технічний проект ІАС	513
15.2.6. Робоча документація та введення в дію	515
15.3. Розробка структурних схем ІАС	515
15.3.1. Зміст структурних схем	515
15.3.2. Використання графової моделі інформаційних потоків (С2Г)	518
15.3.3. Використання блочної моделі інформаційних потоків (С2Б)	523
15.4. Розробка документів інформаційного забезпечення ІАС	525
15.4.1. Комплектність документів інформаційного забезпечення	525
15.4.2. Використання схеми мережних інформаційних потоків (СІП)	526
15.4.3. Використання схеми інформаційних потоків у мережах MODBUS RTU	532
15.4.4. Використання схеми інформаційних потоків у мережах на базі Ethernet	533
15.4.5. Використання схеми інформаційних потоків при проектуванні мереж PROFIBUS	540
15.4.6. Використання схеми інформаційних потоків при проектуванні мереж CANOpen	542
Література	546

Вступ

Одною з ознак початку нового етапу розвитку систем автоматизації промислових підприємств є пошук шляхів інтеграції двох систем автоматизації — «Автоматизованих систем управління підприємством» (АСУП) і «Автоматизованих систем управління технологічними процесами» (АСУТП), які до останнього часу розвивались відокремлено і незалежно одна від одної. Традиційно системи АСУП орієнтовані на автоматизацію управлінської і фінансово-господарської діяльності, а також на планування ресурсів підприємства. Системи АСУТП вирішують завдання автоматизації технологічних і виробничих процесів. Головною метою інтеграції різних підсистем на виробництві є створення єдиного інформаційного простору підприємства для об'єктивної і оперативної оцінки стану підприємства, оперативного прийняття своєчасних і ефективних управлінських рішень, а також ліквідації інформаційних та організаційних бар'єрів між управлінським та технологічним рівнями.

Економічною основою інтеграції є прагнення керівників підприємств шукати реальні додаткові джерела підвищення економічної ефективності діяльності підприємства. На кожному підприємстві такі джерела є, треба тільки вміти їх знайти, а для цього необхідно забезпечити збір, обробку і аналіз інформації оперативних даних з усіх технологічних і виробничих ділянок виробництва. Саме оперативної, а не в кінці зміни, робочого дня або місяця. Для того, щоб керувати собівартістю продукції треба поряд з інформацією про вартість сировини і робочої сили знати скільки сировини, електроенергії, пари, води і палива пішло на виготовлення кінцевої продукції. Для харчової промисловості це дуже важливо, тому що питома вага вартості сировини в собівартості продукції є досить високою, а основні технологічні процеси вимагають використання великої кількості енергоресурсів. Інтеграція АСУП і АСУТП дає змогу реалізувати поточне та оперативне планування витрат і собівартості, а також забезпечувати їх розрахунок у реальному часі одночасно зі змінами темпу виробництва, миттєво реагуючи на відхилення від заданого рівня. На базі поточної інформації із АСУТП можливо реалізувати цільове керування з таких показників:

- якості продукції і підтримання необхідних споживчих властивостей продукції;
- енергозбереження та економії ресурсів;
- підтримання заданої продуктивності.

Технічною передумовою інтеграції систем управління є широке впровадження на виробництві комп'ютерної і мікропроцесорної техніки, а також сучасних інформаційних технологій. Треба зазначити, що не завжди виробничі процеси реалізовувалися у межах чітко визначеної стратегії розвитку систем управління. Найчастіше ці завдання вирішувалися локально і тому при пошуку шляхів їх інтеграції можуть виникнути певні труднощі. Тому процес інтеграції систем управління може відбуватися або тоді, коли на підприємстві вже функціонують окремі системи автоматизованого управління і необхідно визначити шляхи їх взаємодії або за умов, коли розв'язання цього завдання тільки планується. І в тому і в іншому випадку для практичної реалізації інтегрованої системи управління конкретним виробництвом необхідно чітко визначити мету інтеграції, а також проаналізувати сучасний стан організаційного, функціонального, технічного та програмного забезпечення існуючих на виробництві АСУП і АСУТП. Це дозволить визначити чітку стратегію розвитку систем управління виробництвом і уникнути невиправданих матеріальних втрат від прийняття необґрунтованих і незгоджених рішень.

На рис. 1 наведений приклад узагальненої інтегрованої структури системи управління, із якої видно, що одним із основних елементів цієї схеми є використання різноманітних обчислювальних мереж на різних рівнях управління. Без них неможливо створити єдиний інформаційний простір.

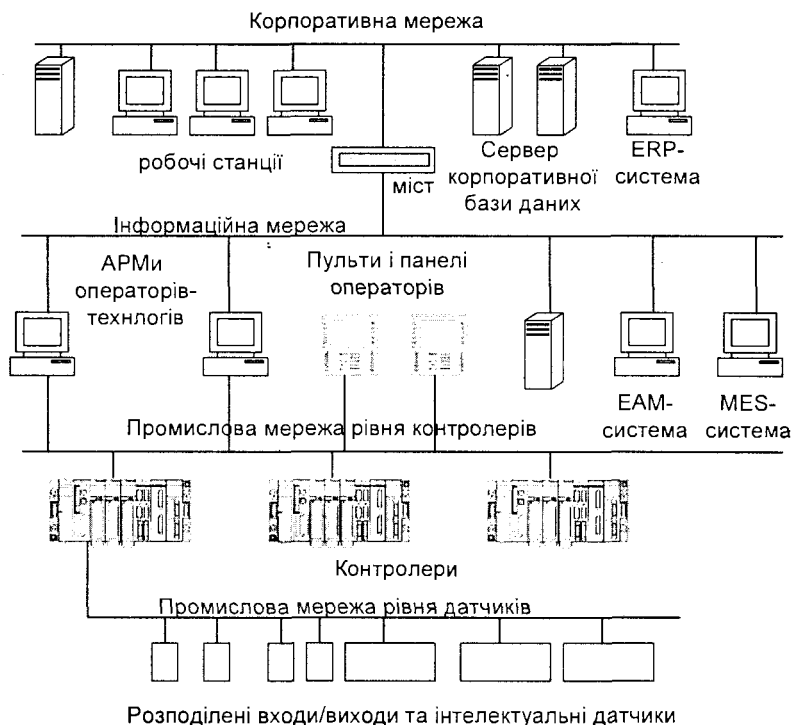


Рис. 1. Приклад узагальненої структури інтегрованої системи управління

Є суттєві відмінності у вимогах до інформаційних мереж (структури, технічного і програмного забезпечення), які використовуються на різних рівнях управління. Якщо обчислювальні мережі, які використовуються на рівні організаційно-економічного управління описані досить широко у спеціальній літературі, то опис мереж та інтеграційних технологій рівня АСУТП потребують узагальнення і систематизації.

У навчальному посібнику розглянуті матеріали, присвячені промисловим мережам та інтеграційним технологіям. Навчальний посібник має умовно чотири частини. У першій частині розглядаються загальні питання, присвячені висвітленню питання та місця мережних технологій в інтегрованих автоматизованих системах (розділ 1), загальним характеристикам та короткому огляду промислових мереж (розділ 2), а також характеристикам каналів зв'язку, тобто реалізації фізичного рівня для цього типу мереж (розділ 3).

У другій частині наводяться дані з принципів функціонування та проектування деяких, найбільш використовуваних в Україні промислових мереж (розділи 4–10). При цьому використаний досвід роботи авторів з цими мережами як у лабораторних, так і промислових умовах. Наведені приклади організації мереж з використанням технічної бази різних виробників. В розділі 11 описані стандартні технології управління електроприводами через промислові мережі.

Третя частина присвячена технологіям програмної інтеграції. Зокрема в розділі 12 наводяться загальні відкриті технології міжпрограмної взаємодії, які використовуються в середовищах Microsoft Windows. Розділ 13 повністю присвячений найбільш популярній у галузі АСУТП технології OPC, зокрема специфікації OPC DA 2.0. Відкриті технології доступу до баз даних розглядаються в розділі 14.

Остання частина (розділ 15) присвячена питанням проектування інтегрованих автоматизованих систем, в яких використовуються промислові мережі та технології міжпрограмної взаємодії.

Навчальний посібник може бути використаний при вивченні курсів циклу професійної і практичної підготовки рівня бакалавра для напряму 6.050202 «Автоматизація та комп'ютерно-інтегровані технології» («Автоматизація технологічних процесів і виробництва», «Інтегровані системи управління», «Контролери та їх програмне забезпечення») при вивченні дисциплін ОКР спеціаліст-магістр зі спеціальностей «Автоматизоване управління технологічними процесами» та «Комп'ютерно-інтегровані технологічні процеси і виробництва» («Проектування комп'ютерно-інтегрованих систем», «Програмно-технічні комплекси та промислові контролери»), а також при виконанні курсового і дипломного проектування.

При вивченні дисципліни «Проектування комп'ютерно-інтегрованих систем управління» рекомендується така послідовність вивчення курсу. Для початку студент ознайомлюється з першим та другим розділами, які є обов'язковими при вивченні дисципліни. За необхідності детального вивчення особливостей реалізації фізичного рівня та стандартних послідовних інтерфейсів, студенту рекомендується прочитати 3-й розділ. Після ознайомлення з основними концепціями можна приступити до розгляду конкретної промислової мережі на вибір (розділи

4–9), які побудовані як самостійні частини. Детальне вивчення розділу 8 (CAN) є хорошою основою для вивчення функціонування мережі CANOpen (розділ 9), однак це не є обов'язковим. Слід наголосити, що розділи 4–9 є незалежними, що дає змогу зосередитись на вивченні конкретної мережі, залежно від наявної технічної бази лабораторій. Якщо вивчення ведеться тільки на теоретичній базі, рекомендується вивчення мереж MODBUS (розділ 6), як найбільш простих для сприйняття.

Після освоєння базових концепцій розділів 1-го, 2-го, бажано 3-го, одного з розділів від 4-го по 9-й, можна приступати до вивчення розділу 15. Така послідовність дає можливість завершити частину, яка присвячена промисловим мережам.

Рекомендується ознайомитися з розділами 10 та 11, що можна зробити вже після вивчення перших двох розділів.

Для ознайомлення з технологіями міжпрограмної інтеграції необхідно послідовно прочитати розділи з 12 по 14. Розділ 15 є логічним завершенням цієї частини.

Отже, мінімальна послідовність для вивчення курсу «Проектування комп'ютерно-інтегрованих систем» така:

1. Розділ 1. Місце та роль мережних технологій в інтегрованих автоматизованих системах управління.
2. Розділ 2. Загальна характеристика промислових мереж.
3. Розділ 3. Реалізація фізичного рівня.
4. Один з розділів (4-10).
5. Розділ 15. Проектування інтегрованих автоматизованих систем управління (в контексті промислових мереж).
6. Розділ 12. Відкриті технології програмної інтеграції в середовищі Windows.
7. Розділ 13. Технологія OPC.
8. Розділ 14. Стандартні технології доступу до баз даних.
9. Розділ 15. Проектування інтегрованих автоматизованих систем управління (в контексті міжпрограмної взаємодії).

З іншими розділами студент може ознайомитись у будь-який час після вивчення основної послідовності.

Посібник також може бути корисний спеціалістам у галузі автоматизації виробництва, які спеціалізуються на проектуванні і розробці комп'ютерно-інтегрованих систем управління виробництвом.

МІСЦЕ ТА РОЛЬ МЕРЕЖНИХ ТЕХНОЛОГІЙ В ІНТЕГРОВАНИХ АВТОМАТИЗОВАНИХ СИСТЕМАХ УПРАВЛІННЯ

1.1. Інтегровані автоматизовані системи управління

Інтеграція — це процес організації окремих компонентів в єдину систему, який забезпечує узгоджену та цілеспрямовану їх взаємодію для досягнення заданої мети. Результатом даного процесу являється інтегрована система.

Залежно від компонентів та кінцевого функціонального призначення, в галузі промислової автоматизації розрізняють декілька видів інтегрованих систем. Інтегрована автоматизована система (*ІАС*) — сукупність двох або більше взаємопов'язаних автоматизованих систем (*АС*), в якій функціонування однієї з них залежить від результатів функціонування іншої (інших) так, що цю сукупність можна розглядати як єдину автоматизовану систему. Інтегрована автоматизована система управління (*ІАСУ*) — автоматизована система, в якій комбінуються функції управління технологічними процесами і організаційного управління виробничим підприємством. Тобто *ІАСУ* — це результат інтеграції *АСУТП* (автоматизована система управління технологічним процесом) та *АСУП* (автоматизована система управління підприємством) або модифікації останніх у вигляді корпоративної системи управління бізнес-процесами (*КСУБП*). Можлива також інтеграція *АСУП* та *АСУЯ* (автоматизовані системи управління якістю), *АСУП* та *САПР* (системи автоматизованого проектування), а також інших систем. Інтегрованими також називають *КСУБП*, маючи на увазі об'єднання матеріальних і фінансових потоків або кількох пакетів програмного забезпечення в один інтегрований пакет прикладних програм для бухгалтерії, кадрових служб, збуту та постачання, а також виробництва з центральним фінансовим компонентом. З появою в системах керування комп'ютерів, як основного технічного засобу, з'являється також термін комп'ютерно-інтегровані системи управління (*КІСУ*). *КІСУ* — ієрархічно розподілена система, що інтегрує функції управління технологічними та організаційно-економічними процесами підприємства і складається

з робочих станцій, об'єднаних у локально-обчислювальні мережі. У зарубіжній літературі аналогічно терміну КІСУ є два терміни **CIM** (Computer Integrated Manufacturing — комп'ютерно-інтегроване виробництво) та **CIP** (Computer Integrated Processing — комп'ютерно-інтегрований процес).

Вибір переважного спрямування інтеграції АСУ потребує оцінки ефективності сумісного та узгодженого функціонування локальних автоматизованих систем, а також витрат на забезпечення їх інтеграції. Розрізняють «горизонтальну» інтеграцію, що потрібна для прийняття рішень на певному рівні управління, і «вертикальну», що пов'язана з інтеграцією функцій управління між різними рівнями ієрархії управління.

Однією з основних функцій *горизонтальної інтеграції* (рис. 1.1.) є забезпечення інформаційної взаємодії між існуючими підсистемами одного рівня. На рівні АСУТП горизонтальна інтеграція передбачає об'єднання між собою АСУ управління технологічних і виробничих процесів, а також адміністративних відділень виробництва в єдину систему. Це забезпечує необхідний обмін даними у реальному масштабі часу між усіма підрозділами основного і додаткового виробництва. З виробничої точки зору, це означає облік кожного кроку виробничого процесу від прибуття сировини до отримання готової продукції. Це дає змогу скоординувати роботу окремих технологічних ділянок з метою поліпшення ритмічності виробництва, зменшення втрат, підвищення якості готової продукції та зменшення собівартості. На рівні АСУП горизонтальна інтеграція призначена для об'єднання робочих станцій організаційно-економічного та планового рівнів управління підприємством. Це дає змогу координувати дії з постачання сировини і необхідних матеріалів основного виробництва, спланувати збут, рівномірно завантажити різні ділянки виробництва, спланувати проведення поточного ремонту і т. ін.



Рис. 1.1. Напрями інтеграції в автоматизованих системах управління

Вертикальна інтеграція призначена для об'єднання систем різного рівня ієрархії управління, наприклад АСУП та АСУТП. Враховуючи, що дані системи працюють для досягнення єдиної мети, наприклад отримання прибутку, їх об'єднання повинно покращити цей процес.

Інтегрована АСУ, як і будь-яка автоматизована система, характеризується такими основними видами забезпечення: організаційне, інформаційне, математичне, програмне, алгоритмічне і технічне. Створення ІАСУ передбачає роботи з об'єднання окремих АС у контексті цих видів забезпечень. Відповідно до видів забезпечення, відносно яких проводиться інтеграція, розрізняють організаційну, функціональну, інформаційну, програмну і технічну інтеграції.

Організаційна інтеграція — раціональне поєднання управлінської діяльності персоналу з усіх рівнів інтегрованої АСУ і в різних локальних її підсистемах, яке визначає узгодженість управлінських рішень.

Функціональна інтеграція забезпечує єдність локальних цілей функціонування, узгодженість функцій та критеріїв ефективності всіх компонентів. Функціональна інтеграція полягає у розробці загальної функціональної структури всієї системи, декомпозиції системи на функціональні підсистеми. При цьому для неї визначаються загальносистемні функції (задачі), проводиться їх декомпозиція на підфункції (підзадачі) у межах кожної функціональної підсистеми, визначається їх функціональний та інформаційний зв'язки. Процес функціональної інтеграції проводиться з використанням як методів декомпозиції, так і агрегування. Для вирішення загальносистемних функцій, підфункцій, що з'являються в результаті декомпозиції, поєднуються між собою інформаційними зв'язками.

Інформаційна інтеграція полягає у створенні умов, за яких можливий доступ до всіх необхідних даних для реалізації функцій системи. Ці дані, як правило, проходять декілька стадій обробки, в яких беруть участь технічні та програмні засоби. Враховуючи можливість багатокомпонентного складу програмного та технічного забезпечення, виникає необхідність в їх інтеграції. Здача інформаційної інтеграції також полягає в узгодженості форматів даних, їх структурування, найменування (ідентифікації) та уніфікації представлення.

Програма інтеграція полягає у забезпеченні сумісного функціонування окремих складових програмного забезпечення з метою їх взаємодії.

Технічна інтеграція передбачає поєднання технічних засобів для забезпечення збору, передачі та циркуляції технологічної, техніко-економічної та командної інформації між необхідними складовими системи. В ІАСУ технічна інтеграція базується на обчислювальних мережах, які об'єднують програмно-технічні засоби в єдину розподілену обчислювальну систему.

Основним завданням ІАСУ є реалізація узгоджених рішень на різних рівнях управління, для чого необхідно визначити комплекс завдань управління, як завдань оптимізації за деяким загальним для системи критерієм ефективності функціонування з використанням економіко-математичних моделей управління. Створення принципів та алгоритмів управління є найбільш складним і клопітким і залежить від типу, специфіки виробництва та підприємства. Тому першим кроком при побудові ІАСУ є процес функціональної інтеграції.

1.2. Функціональна інтеграція

Функціональна інтеграція приводить до створення функціональної структури, визначення загальносистемних функцій (задач), їх декомпозицію на підфункції (підзадачі) у межах функціональних підсистем, синтез існуючих функцій в єдину цільову функцію. **Функція** автоматизованої системи — сукупність дій АС, спрямованих на досягнення певної мети. **Задача** автоматизованої системи — функція чи частина функції АС, що є формалізованою сукупністю автоматичних дій, виконання яких приводить до результату заданого виду.

Результатом функціональної інтеграції мають бути алгоритми оптимального управління підприємством в цілому, однак велика розмірність та складність цієї задачі роблять неможливим її розв'язання без застосування декомпозиційних методів. Водночас, розглядаючи складну задачу як сукупність більш простих декомпозиційних компонентів, в подальшому необхідно інтегрувати ці компоненти в єдиний комплекс. Таким чином, методи декомпозиції та інтеграції в ІАСУ застосовують одночасно.

При створенні ІАСУ розроблюють не тільки моделі і алгоритми вирішення функціональних задач, але і алгоритми зв'язку і узгодження локальних задач. Зв'язки між ними реалізуються за допомогою зовнішніх обмежень, узгодження локальних критеріїв і моделей локальних задач.

У результаті декомпозиції отримують багаторівневу ієрархічну функціональну структуру, в якій кожна задача верхнього рівня має пріоритет відносно до задач нижнього рівня, а завданнями для задач нижнього рівня є результати розв'язання задачі більш високого рівня. Використання принципів багаторівневої ієрархії приводить до виділення в ІАСУ таких функціональних рівнів управління.

1. Рівень управління підприємством (**ERP** — Enterprise Resource Planning) розглядає задачі перспективного та поточного планування, у розв'язанні яких визначаються інтегральні показники оптимальної виробничої програми і розраховуються календарні плани на квартал, місяць, тиждень.

2. Рівень управління виробництвом (**MES** — Manufacturing Execution Systems) відповідає задачам оперативного календарного планування, у розв'язанні яких спочатку коригуються тижневі календарні плани з урахуванням фактичного виконання виробничої програми, а потім визначаються оперативні задачі на поточну добу з урахуванням різних збурень;

3. Рівень управління процесом (рівень **SCADA**) відповідає задачам оперативного керування виробничими комплексами, розв'язання яких забезпечує оптимальну координацію роботи відділень, цехів;

4. Рівень локального управління забезпечує вирішення задач управління технологічним процесом окремої дільниці або відділення, які забезпечують збір і обробку первинної інформації, контроль і регулювання технологічним процесом.

Нижні рівні (локального управління, управління процесом та частково **MES**) можна віднести до систем АСУТП, верхні (частково **MES**, управління підприємством) — до АСУП.

1.3. Інформаційна інтеграція

Виділені в результаті функціональної декомпозиції функції (задачі), поєднуються між собою інформаційними зв'язками. Інформаційна інтеграція має забезпечити доступ до необхідних даних для реалізації цих функцій. Таким чином, можна сказати, що функції (задачі) поєднуються між собою *інформаційними потоками*, які забезпечують передачу даних. В інтегрованих автоматизованих системах управління, дані циркулюють по обчислювальним мережам. Тобто функції, які реалізовані на різних вузлах, об'єднані між собою мережними інформаційними потоками.

В ідеалі доступ до необхідних даних повинен бути прозорим. Дані можуть використовуватись у декількох функціях, що приводить до необхідності в поєднанні їх у бази даних (*БД*), що є сукупністю логічно пов'язаних інформаційних елементів, які зберігаються в запам'ятовуючих пристроях. Ввід даних у БД, їх пошук, оновлення та реорганізацію виконує система управління базою даних (*СУБД*). Водночас інтеграція всіх даних в одну БД, враховуючи їх різний характер, веде до зниження ефективності процедур обробки даних. Тому більш ефективним є створення центральної (для керування бізнес-процесами) та локальних (для керування технологічними процесами) баз даних, при цьому повинна бути забезпечена сумісність цих БД та доступ до них з необхідних вузлів системи. В деяких системах доцільним є використання розподілених баз даних, де дані зберігаються на кількох вузлах, як локальні копії оригінальних даних. Після оновлення даних у джерелі, інформація копіюється на декілька вузлів системи.

Особливі вимоги ставляться до складу і кількості інформації та часу її обробки, які є критичними факторами при створенні системи. З одного боку, тільки своєчасне подання необхідної інформації може забезпечити якісне прийняття рішень, а з іншого — інформаційні потоки мають бути обмеженими за обсягом, щоб запобігти перевантаженню їх другорядними, надто деталізованими даними.

Для розв'язання задач функціональної та інформаційної інтеграції можна використати різноманітні механізми. В 15-му розділі пропонуються два підходи: використання графової та блочної моделі інформаційних потоків, які показують інформаційні взаємозв'язки між функціями, що дає змогу, з одного боку, графічно продемонструвати інформаційну інтеграцію, а з іншого — визначити проблеми інформаційного характеру.

1.4. Технічна та програмна інтеграція

Визначення задач та функцій, які повинна реалізувати автоматизована система, є важливим етапом при її створенні. Процес декомпозиції повинен враховувати особливості реалізації кінцевих функцій, тобто особливості вибраної структури системи. Тому процедура декомпозиції поставлених загально системних задач на окремі підзадачі носить ітераційний характер. Технічна та програмна інтеграції проводяться з урахуванням специфіки використовуваних засобів та технологій, на яких базується їх робота. Програмно-технічні засоби (*ПТЗ*), які використовую-

ються в інтегрованих автоматизованих системах управління можна розділити на два класи: ПТЗ рівня АСУТП та ПТЗ рівня АСУП.

ПТЗ рівня АСУТП можна класифікувати за функціональним призначенням на:

- засоби людино-машинного інтерфейсу;
- контролери;
- периферійні пристрої;
- програматори/конфігуратори.

З технічної точки зору ПТЗ рівня АСУП є комп'ютери (ПК, сервери) з офісним та спеціалізованим програмним забезпеченням

Необхідність їх інтеграції в єдину систему диктується функціональними а відповідно й інформаційними зв'язками. Розглянемо це для кожного конкретного типу засобу.

1.4.1. Засоби людино-машинного інтерфейсу

Основним завданням цих засобів є збирання, зберігання, попередня обробка, передача і відображення технологічної інформації (даних процесу) для забезпечення ефективної взаємодії системи управління і оператора.

Технічно вони можуть бути реалізовані як (рис. 1.2):

- термінали та операторські панелі (*ОП*);
- автоматизовані робочі місця (АРМ) оператора, виконані на базі промислового чи офісного персонального комп'ютера з установленим спеціалізованим програмним забезпеченням — SCADA/HMI.

Термінали та операторські панелі — це мікропроцесорні засоби людино-машинного інтерфейсу (*HMI/MMI* — Human/Man Machine Interface), в яких обов'язково присутній хоча б один інтерфейс підключення до цифрової мережі, за допомогою якого відбувається обмін даними з контролерами. На сьогодні функціонально операторські панелі наближаються до ПК зі SCADA/HMI.

АРМ на базі ПК з виконавчою системою HMI забезпечує відображення та зміну даних, що розміщені на серверах SCADA. Виконавчі системи *SCADA* (Supervisory Control And Data Acquisition — диспетчерське керування й збір даних) відіграють роль програм-серверів, які забезпечують збір, попередню обробку даних, опосередковану їх зміну та контроль значень (тривоги), ведення архіву та доступ до архівних даних. Таким чином HMI системи повинні мати доступ до даних реального часу, які акумулюються в серверах SCADA. Останні, в свою чергу, доступуються до даних контролерів, периферійних засобів, до інших SCADA або до інших програмно-технічних засобів. Практично всі сучасні програмні пакети для побудови систем SCADA призначені також для побудови АРМ з функціями HMI, тому вони мають назву SCADA/HMI. Крім того, виконавчі системи SCADA та HMI в автоматизованих системах середньої складності, як правило, виконуються на одному ПК.

Незалежно від типу наведених вище засобів, їх вхідні дані процесу, як правило, повинні відновлюватись циклічно або періодично, із заданим інтервалом часу. Це обумовлено необхідністю контролю з боку оператора за процесом. У деяких

випадках необхідно налаштувати надходження вхідних даних процесу в SCADA тільки при певній події, яка буде визначатись у контролері. Те саме стосується надходження даних з контролерів або із серверів SCADA в HMI. Більшість засобів SCADA/HMI передбачають можливість опитувати дані процесу з різною частотою. Передача даних у зворотному напрямку (SCADA->контролер, HMI->SCADA, ОП->контролер) відбувається, як правило, при зміні їх значень.

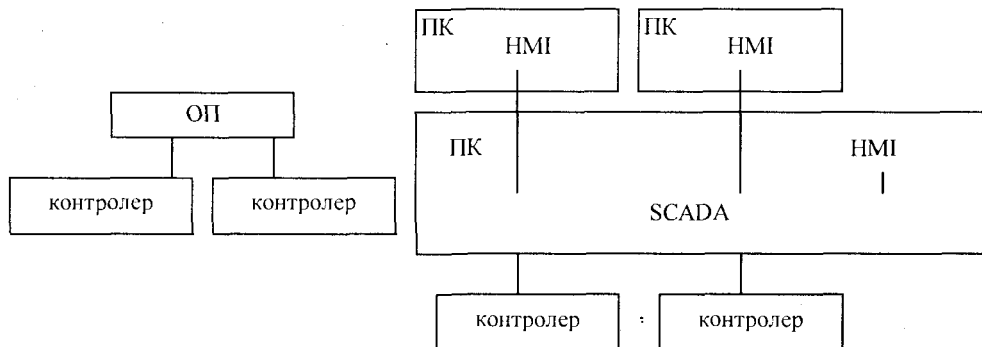


Рис. 1.2. Реалізація засобів людино-машинного інтерфейсу в ІАСУ

Слід зазначити, що в інтегрованих автоматизованих системах управління може виникнути необхідність в обміні даними між ОП, системами SCADA/HMI у різній їх комбінації. Тобто програмна та технічна інтеграції для наведених засобів передбачають об'єднання їх з контролерами або периферійними засобами та між собою. В свою чергу, SCADA/HMI — системи можуть обмінюватися даними з ПТЗ рівня АСУП.

Так як ОП та засоби SCADA/HMI призначені для контролю за даними процесу з боку оператора, це виключає необхідність забезпечення передачі інформації в жорсткому реальному часі. У більшості автоматизованих систем запізнення даних (невчасне їх оновлення) не є аварійнонебезпечним, але сигналізується засобами аварійно-попереджувальної сигналізації (АПС). Таким чином, для засобів людино-машинного інтерфейсу виділимо такі характеристики зв'язку:

- періодичне зчитування даних процесу з контролерів/периферійних засобів;
- запис значень даних процесу при їх зміні;
- м'який реальний час.

1.4.2. Контролери

Пристрої з функціями управління в даному посібнику будемо називати *контролерами*. Саме в них реалізуються алгоритми управління. До цих пристроїв належать:

- **ПЛК** — програмовані логічні контролери (**PLC** — Programming Logical Controller);
- **ІВМ РС-сумісні (PC-base)** контролери;
- **ОПЛК** — ПЛК з убудованою операторською панеллю ОП (**OPLC**);
- **контролери-регулятори** з мережним інтерфейсом.

В якості ПЛК у системах контролю й управління різними технологічними процесами у наш час використовують контролери як вітчизняних, так і закордонних виробників. У наведеному переліку до ПЛК також будемо відносити програмовані реле. До IBM PC-сумісних контролерів належать промислові IBM PC-сумісні ПК, які використовуються для автоматичного управління. ПЛК з убудованими функціями ОП та регулятори з наявним мережним інтерфейсом також будемо відносити до даних засобів.

Особливістю апаратно-програмних засобів «контролерного» рівня управління полягають у пред'явленні жорстких вимог до часу їх реакції на зміну даних процесу, які надходять від датчиків, та видачі управляючих дій на виконавчі механізми. Тобто вони повинні функціонувати в реальному для процесу часі.

У системах управління контролери повинні обмінюватись даними процесу з периферійними пристроями вводу/виводу, іншими контролерами, програматорами, засобами людино-машинного інтерфейсу, а в деяких випадках — і з ПТЗ рівня АСУП.

Вимоги до реального часу обміну даними процесу між контролерами і датчиками та виконавчими механізмами, які мають цифровий інтерфейс підключення, а також віддаленими модулями вводу/виводу (Remote I/O або Distributed I/O) такі самі, як до локальних засобів вводу/виводу. Для цього типу зв'язку характерні такі особливості:

- циклічне (періодичне) відновлення даних на читання та запис;
- забезпечення реального часу;
- можливість прив'язки циклу мережі до циклу контролеру;
- можливість гарячого підключення засобів.

Цикл мережі, тобто повний цикл опитування та відновлення інформації всіх засобів вводу/виводу, може проходити незалежно від циклу контролера, оскільки обмін даними між мережею і програмою контролера, як правило, проводиться через буфер обміну. Такий підхід максимально продуктивний, однак не завжди задовольняє вимоги процесу. Для деяких завдань необхідна чітка синхронізація робочих циклів мережі і контролера. Тобто на початку циклу мережі опитуються всі засоби вводу, а в кінці відновлюються дані всіх засобів виводу.

Для багатьох процесів управління принципово є можливість заміни або підключення нових віддалених засобів вводу/виводу в період функціонування мережі, так звана «гаряча заміна».

При координації роботи декількох контролерів (горизонтальна інтеграція), необхідно побудувати канал зв'язку між ними з такими особливостями комунікаційного обміну:

- наявність загальної бази даних процесу, яка циклічно (періодично) оновлюється;
- обмін даними процесу та повідомленнями за запитом;
- наявність реального часу.

Наявність загальної бази даних процесу або мережних змінних процесу необхідна у випадку, коли одні і ті ж дані використовуються багатьма контролерами. Тобто дані процесу, які є джерелом інформації для одного контролера, потрібні багатьом іншим контролерам цієї системи. У випадку, коли дані необхідно віді-

слати тільки одному вузлу і лише при певних подіях, є доцільним організувати обмін даними за запитом.

В організації обміну інформацією між контролерами окремо слід виділити системи з «гарячим» резервуванням. У цьому випадку окремий фізичний канал мережі може виділятися для обміну контекстом завдання між основним і резервним процесорними модулями. Інший, спільний для обох процесорів, канал використовується для зв'язку з віддаленими І/О.

Загалом, для обміну контролерів з іншими засобами можна виділити два типи даних: дані процесу (інформація про значення та стан змінних процесу) та параметричні дані (конфігураційні параметри).

1.4.3. Периферійні засоби: перетворювачі та виконавчі механізми

До цих засобів можна віднести різного роду інтелектуальні (intelligent) датчики та виконавчі механізми, які мають цифровий інтерфейс зв'язку або засоби розподіленого/віддаленого вводу/виводу (*Distributed I/O, Remote I/O*). Останні забезпечують перетворення сигналів пристроїв зв'язку з об'єктом (ПЗО) в цифровий вигляд і навпаки та обмінюються даними процесу з процесорним вузлом (контролером). У зв'язку з подорожчанням кабельної продукції, здешевленням мікропроцесорної техніки, гнучкістю та зручністю експлуатації, живучості кінцевої системи — пристрої розподіленого вводу/виводу набувають великої популярності. На рис. 1.3 наведені різні методи підключення периферійних пристроїв до контролерів.

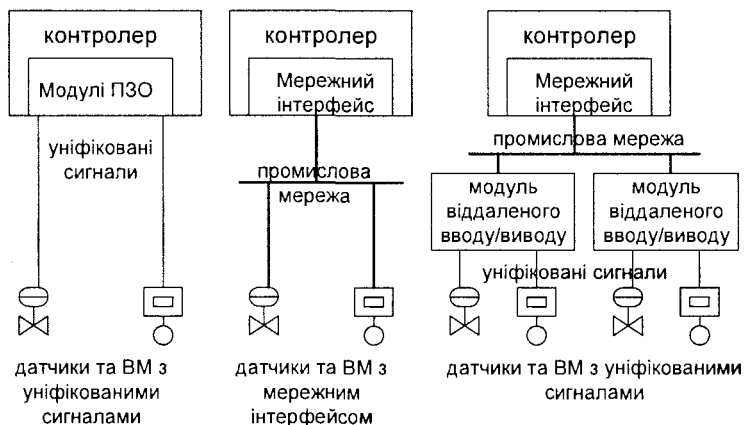


Рис. 1.3. Різні методи підключення периферійних пристроїв до ПЛК

На сьогодні існує велика гама пристроїв віддаленого вводу/виводу. Порівняно з технологією передачі уніфікованого сигналу (0-10В, 4-20 мА) цифровий зв'язок дає такі переваги:

- менш чутливий до зовнішніх завад;
- дає можливість розширеної діагностики пристрою;

– дає можливість змінювати конфігурацію у реальному часі без зупинки роботи управляючої програми;

– дає можливість будувати системи з мережею без виділеного ведучого вузла.

Конфігурування польових периферійних пристроїв може проводитись за допомогою спеціальних засобів — конфігураторів або безпосередньо самим контролером.

Слід зазначити, що мережі з реалізацією обміну між польовими вузлами без виділеного ведучого виключають наявність контролера шини в комунікаційному процесі, а отже — підвищують надійність системи. Однак така можливість доступна тільки для деяких промислових мереж. Слід зазначити, що на відміну від автоматизованих систем управління технологічними процесами (АСУТП) в системах управління інтелектуальними будівлями та будинками (системи **SMART HOUSE** та **BMS** — Building Management Systems) використання інтелектуальних датчиків та приводів є швидше правилом, ніж винятком.

Окремо необхідно виділити засоби управління приводами (POWER DRIVE SYSTEMS — **PDS**). До них належать частотні перетворювачі, сервоприводи та приводи з позиціонуванням. Ці засоби, крім своїх основних функцій — управління двигунами, нерідко включають також функції контролера, людино-машинного інтерфейсу та функції вводу/виводу. Все, що стосується особливостей обміну даними для вищенаведених типів периферійних засобів, може бути актуальними і для PDS. З'єднання за цифровим зв'язком з PDS дає такі переваги:

– отримання всієї інформації про стан привода та двигуна;

– повне управління приводом та двигуном;

– віддалене та швидке конфігурування системи приводу;

– менша витрата інформаційного кабелю порівняно зі зв'язком за уніфікованим сигналом 4-20мА/24В;

– віддалена діагностика привода;

– можливість безпосереднього обміну між приводами.

Особливості обміну даними периферійних засобів з контролерами, засобами людино-машинного інтерфейсу наведені вище, з конфігураторами — розглянуті в пункті 1.4.4. Загалом, дані для обміну з периферійними засобами умовно можна поділити на два типи: дані процесу (інформація про значення та стан каналів вводу/виводу) та параметричні дані (конфігураційні параметри).

1.4.4. Програматори/конфігуратори

В якості програматорів/конфігураторів можуть використовуватись комп'ютери з установленим спеціальним програмним забезпеченням або спеціалізовані пристрої. Незалежно від технічної реалізації засобів цього типу, вони необхідні лише для налаштування роботи інших ПТЗ в системі та їх діагностики. Програматори мають інтерфейс для підключення до мережі, який використовується тільки при необхідності перепрограмування або переконфігурування. Виділимо такі характеристики зв'язку:

– підключення до системи тільки за необхідності;

– спеціально виділений логічний канал зв'язку;

- наявність команд на запис конфігурації, діагностики, зупинки, рестарту та інше;
- низька пріоритетність повідомлень;
- можливість гарячого переконфігурування системи зв'язку.

У зв'язку з особливістю використання даних засобів, для них є доцільним резервувати окремий логічний канал, який за будь-якої діючої конфігурації системи буде вільним. Інакше можливе виникнення ситуації неможливості переконфігурування пристрою, внаслідок задіяння всіх існуючих каналів, що є неприпустимим. Крім того, протокол мережі повинен бути забезпечений функціями запису конфігурації, діагностики, управлінням станом пристрою тощо.

Процес конфігурації одного вузла мережі не повинен заважати її нормальному функціонуванню. Можливість гарячої заміни програми, або її частини, без зупинки роботи системи є одним із основних показників ефективності програмно-технічного комплексу. Низька пріоритетність повідомлень дає можливість не заважати реальному часу обміну між іншими вузлами на тій самій мережі.

Програматори/конфігуратори в основному оперують конфігураційними (параметричними) даними. Винятком є режим відладки, коли наладчику необхідно спостерігати та змінювати дані процесу. Слід зазначити, що параметричні дані на засоби вводу/виводу, як правило, поступають не безпосередньо з конфігураторів, а через ведучий вузол мережі — контролер.

1.4.5. Програмно-технічні засоби рівня АСУП

Як уже зазначалося, програмно-технічними засобами рівня АСУП є комп'ютери. На сьогодні для програмного забезпечення цього рівня в ІАСУ найбільш популярною апаратною платформою є IBM-PC сумісні комп'ютери з ОС WINDOWS.

Програмне забезпечення інтегрованих автоматизованих систем управління на рівні АСУП можна умовно віднести до однієї з таких груп:

- універсальні та спеціалізовані СУБД;
- системи рівня MES
- системи рівня ERP;
- офісне програмне забезпечення.

Під універсальними СУБД будемо розуміти сучасні програмні пакети для роботи з базами даних, призначених для широкого застосування. Спеціалізованими СУБД будемо вважати ті програмні пакети, які призначені для збору та обробки даних процесу в області АСУТП. Спеціалізовані СУБД мають жорсткіші вимоги до реального часу і до надійності.

Усі програмні пакети, які призначені для роботи на організаційно економічному рівні підприємства, за винятком стандартних офісних програм, будемо називати системами рівня ERP.

Обмін даними між засобами АСУП, а також засобів АСУП з АСУТП можна охарактеризувати такими характеристиками:

- відновлення даних за запитом;
- відсутність реального часу;

– необхідність в обміні великими обсягами різнорідної інформації;

– великий обсяг даних для загального доступу;

Для підбиття підсумку з інформаційної взаємодії між вузлами ІАСУ зведемо загальні характеристики такого обміну в таблицю (табл. 1.1). На перетині рядків та колонок вказуються характеристики інформаційного обміну між ПТЗ вказаному в найменуванні рядків та колонок. Для компактності введені скорочення.

Таблиця 1.1

**НАЯВНІСТЬ ТА ХАРАКТЕРИСТИКА ІНФОРМАЦІЙНОГО
ЗВ'ЯЗКУ МІЖ ПТЗ В ІАСУ**

	SCADA/HMI	контролери (PLC)	периферійні засоби (RIO)	програма тори/ конфігуратори (PG)	ПТЗ рівня АСУП (АСУП)
SCADA/HMI	періодичне зчитування даних процесу SCADA/HMI<-SCADA; запис значень процесу SCADA/HMI ->SCADA за запитом; м'який реальний час	періодичне зчитування даних процесу SCADA/HMI<-PLC; запис значень процесу SCADA/HMI -> PLC за запитом; м'який реальний час;	циклічне (періодичне) зчитування даних процесу SCADA/HMI<-RIO; запис значень процесу SCADA/HMI -> RIO по запиту; м'який реальний час	підключення до SCADA тільки за необхідності заливки файлів виконання (параметричних даних); наявність команд діагностики, зупинки, рестарту виконавчої системи;	доступ даних SCADA по запиту; відсутність реального часу; необхідність в агрегуванні та накопиченні інформації SCADA;
PLC		наявність загальної бази даних процесу, яка циклічно (періодично) оновлюється; обмін даними процесу та повідомленнями за запитом; наявність реального часу;	циклічне (періодичне) відновлення даних процесу на читання та запис; забезпечення реального часу; можливість прив'язки циклу мережі до циклу контролеру; можливість гарячого підключення засобів;	підключення до ПЛК тільки за необхідності; наявність спеціально виділеного каналу зв'язку; наявність команд на запис конфігурації (параметричні дані), діагностики, зупинки, рестарту та інше.; низька пріоритетність повідомлень (параметричних даних); можливість гарячого переконфігурування системи зв'язку;	як правило, не використовується
RIO			можливе тільки за наявності інтелекту в RIO з можливістю програмування	аналогічно PG<->PLC	як правило, не використовується
PG				може використовуватись для проектування з декількома розробниками	спеціалізоване ПО для розробки може підключатися до АСУП тільки за необхідності заливки файлів виконання;

Закінчення табл. 1.1

	SCADA/HMI	контролери (PLC)	периферійні засоби (RIO)	програма тори/ конфігуратори (PG)	ПТЗ рівня АСУП (АСУП)
АСУП					відновлення даних за запитом; відсутність реального часу; необхідність в обміні великими обсягами різномірної інформації; великий обсяг даних для загального доступу

1.5. Обчислювальні мережі в ієрархії ІАСУ

Як видно з попередніх підрозділів, необхідність інтеграції різномірного обладнання приводить до великої кількості даних (процесу та параметричних), що циркулюють між засобами, їх багаторазової обробки, перетворення та зберігання для подальшого аналізу. На сьогодні технології інформаційної, технічної та програмної інтеграції базуються на цифрових технологіях з використанням обчислювальних мереж. Тобто інтегроване виробництво нині — це єдина система, що являє собою об'єднання різномісних мікропроцесорних вузлів з використанням обчислювальних мереж. Розглянемо один із варіантів ієрархічної моделі ІАСУ підприємства, який наведений на рис. 1.4.

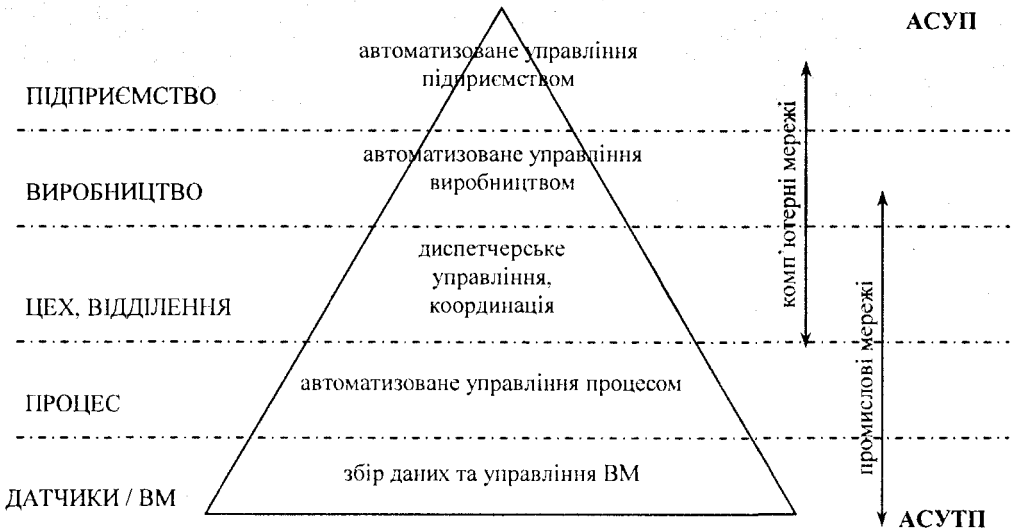


Рис. 1.4. Обчислювальні мережі в інтегрованих автоматизованих системах управління

Для об'єднання вузлів на вищих 3-х рівнях використовуються обчислювальні мережі, які об'єднують ПК та РС-сумісні станції. Такі комунікації базуються на комп'ютерних мережах. На нижніх рівнях управління (цех, відділення, процес, датчики та ВМ) використовуються цифрові комунікації, які об'єднують засоби управління АСУТП. У зв'язку з функціональними та технічними особливостями призначення даних комунікацій, вони базуються на інших типах мереж, які прийнято називати промисловими мережами.

У даному навчальному посібнику розглядаються технології інформаційної, технічної та програмної інтеграції, які використовуються в інтегрованих автоматизованих системах управління: промислові мережі (розділи 2-11) та технології міжпрограмного зв'язку (розділи 12-14).



Контрольні запитання до розділу 1

1. Поясніть необхідність інтеграції автоматизованих систем.
2. Назвіть основні види інтегрованих автоматизованих систем. З яких підсистем вони складаються і для чого проводиться інтеграція?
3. Яке призначення горизонтальної інтеграції на рівнях АСУП та АСУТП?
4. Яке призначення вертикальної інтеграції в ІАСУ?
5. У контексті яких видів забезпечення проводиться інтеграція? У чому полягає їх інтеграція?
6. У чому полягає процес функціональної інтеграції?
7. У чому полягає процес інформаційної інтеграції?
8. У чому полягає процес програмної та технічної інтеграції?
9. Перелічіть основні програмно-технічні засоби ІАСУ та поясніть необхідність в їх інформаційному обміні. Яке призначення та основні відмінності між даними процесу та параметричними даними?
10. Прокоментуйте особливості інформаційного зв'язку між різними ПТЗ в ІАСУ.
11. Яким чином організовується зв'язок між програмно-технічними засобами на різних рівнях управління ІАСУ?

ЗАГАЛЬНА ХАРАКТЕРИСТИКА ПРОМИСЛОВИХ МЕРЕЖ

2.1. Властивості промислових мереж

2.1.1. Визначення промислової мережі та їх градація

2.1.1.1. Визначення «польова шина» згідно з ДСТУ. Згідно з українськими стандартами ДСТУ, *Польова шина* — це локальна мережа з лінійною топологією, призначеною для інформаційного обміну між проблемно і об'єктно орієнтованими блоками (пристроями) та територіально розосередженими джерелами: датчиками, перетворювачами, засобами ручного введення та приймачами (підсилювачами, виконавчими пристроями, засобами віддзеркалення інформації).

2.1.1.2. Визначення «fieldbus» згідно з МЕК. У стандарті МЕК 61158 під терміном *Fieldbus* розуміється цифрова, послідовна, мультиточкова шина з промисловими інструментальними пристроями та пристроями управління такими, як — але не обмежено ними — датчиками, виконавчими механізмами та контролерами.

2.1.1.3. Визначення «промислова мережа» в даному посібнику. В даному посібнику англomовний термін *Fieldbus* (дослівний переклад «польова шина») перекладається як «промислова мережа».

У термінології МЕК є ряд обмежень, визначені словами «шина» («bus»), «мультиточкова» («multipoint»), в які не вписуються ряд промислових мереж. У визначенні ДСТУ фігурує централізований підхід, який характерний тільки для мереж рівня датчиків та виконавчих механізмів. Тому в даному посібнику під *промисловою мережею* розуміється промислова комунікаційна система з послідовною передачею бітів, яка використовує металевий кабель, оптоволокно або радіохвилі для зв'язку між мікропроцесорними засобами автоматизації. Слід зазначити, що в деяких джерелах терміни «польова шина» та «промислова мережа» не є синонімами.

Визначення, які наведені в даному посібнику можуть не співпадати з прийнятими в інших джерелах.

2.1.1.4. Градація промислових мереж за застосуванням. У подальшому при розгляді матеріалу будемо використовувати умовний розподіл промислових мереж залежно від області застосування на два рівні.

Рівень датчиків, завданням мереж якого є безпосереднє або через модулі вводу/виводу опитування датчиків і керування роботою різноманітних виконавчих механізмів. Умовно цей рівень можна розділити на два підрівні: рівень датчиків/виконавчих механізмів (*Sensor/Actuator Level*) та польовий рівня (*Field Level*). Перші призначені для роботи безпосередньо з інтелектуальними датчиками/ВМ, а другі — для зв'язку з польовими засобами (розподіленими засобами вводу/виводу, приводними засобами, операторськими терміналами та панелями). В даному посібнику оба типи мереж належать до рівня датчиків.

Рівень контролерів (*Controller level* або *Cell Level*), промислові мережі якого беруть участь у вирішенні завдань з керування виробництвом в цілому або комплексом технологічних процесів і забезпечують обмін між контролерами, засобами SCADA/HMI та засобами рівня АСУП.

Дане розділення є умовним і може не співпадати з прийнятими в інших джерелах.

2.1.2. Функціональне призначення промислових мереж

Розглянуті у першому розділі типові програмно-технічні засоби та особливості обміну між ними дозволять визначити типові функції промислових мереж. До них можна віднести:

1. Обмін даними процесу в реальному часі.
2. Програмування та конфігурація вузлів.
3. Діагностика вузлів.
4. Управління станом вузла.
5. Функції резервного переключення між вузлами мережі.

2.1.2.1. Обмін даними процесу у реальному часі. Це одна з найважливіших функцій реалізації промислової мережі як рівня датчиків так і рівня контролерів. Дані технологічного процесу постійно змінюються в часі, тому необхідно їх доставити від джерела до споживача в потрібному обсязі, за визначений інтервал часу. Доставка даних процесу в реальному часі є одним із показників ефективності конкретної реалізації мережі. Фізичні рамки реального часу чисельно визначаються для конкретної системи управління. Тому одне рішення може задовольняти вимогам реального часу для однієї системи та зовсім не задовольняти для іншої. Відповідно до стандартів ДСТУ, *режимом реального часу* називається режим оброблення даних, який забезпечує взаємодію обчислювальної системи із зовнішніми, відносно до неї, процесами у темпі, порівнянному зі швидкістю перебігу цих процесів.

2.1.2.2. Програмування та конфігурування вузлів. За допомогою функцій програмування та конфігурування більшість сучасних мікропроцесорних засобів, зокрема контролери та периферійні ПТЗ, можуть програмуватись із використанням спеціалізованого програмного забезпечення, встановленого на ПК або програматорі. На відміну від функцій обміну даними процесу, ці функції не передбачають жорстких вимог до обміну в реальному часі.

2.1.2.3. Функції діагностики. Діагностика роботи вузлів мережі є такою ж важливою, як і діагностика складових будь-якої мікропроцесорної системи. Функції діагностики дозволяють визначити факт (функція виявлення аварії) та причину відмови пристрою (ідентифікація аварії), що дозволяє визначити та замінити дефектний вузол, а в системах з резервуванням (*standby*) — переключитися на резервні підсистеми. Вияв факту несправності вузла повинен бути визначений в режимі реального часу, для можливості зворотної реакції системи на збій, що особливо стосується функціонально небезпечних процесів. Тому ця функція базується на високопріоритетних повідомленнях. Ідентифікація аварії, як правило, потрібна для обслуговуючого персоналу, для можливості усунення проблеми, тому особливих вимог до реального часу не потребує.

2.1.2.4. Функції управління станом вузла. Ці функції необхідні для можливості запуску, перезапуску, зупинки роботи програмного забезпечення та ініціалізації вузла. Враховуючи можливість функціонування декількох вузлів у системі, команди управління станом вузла мають нижчий пріоритет перед обміном даними процесу в реальному часі.

2.1.2.5. Резервування вузлів у системі. За рахунок цих функцій промислова мережа дає зручний і порівняно недорогий спосіб підвищення живучості та надійності системи. Технічні засоби систем з резервуванням діляться на *основні* або *первинні* (Primary) та *резервні* (Standby). В нормальному режимі функціонування у системі працюють основні вузли, за виходу яких з ладу їх функції беруть на себе резервні. В таких системах промислові мережі забезпечують можливість самодіагностики вузлів, тобто вияву дефектних засобів, переключення на резервні вузли та підтримку каналів зв'язку з основними та резервними засобами. Промислова мережа може забезпечити:

- резервування віддалених периферійних засобів;
- резервування контролерів;
- резервування серверів SCADA;
- резервування каналів зв'язку: контролер <-> периферійні засоби, контролер <-> SCADA/HMI, контролер <-> контролер та інші.

Функції мереж в системах з резервуванням розглянемо на прикладі.

Приклад 2.1. Основні концепції. HotStandby.

Завдання. Продемонструвати функціонування системи з резервуванням контролера, на прикладі рішення HotStandby («гарячого резервування») запропонованого Шнейдер Електрик.

Рішення.

На рис. 2.1 показана структура системи, в якій первинний контролер обмінюється процесними даними з віддаленими пристроями вводу/виводу (Remote I/O) через промислову мережу M3. З іншого боку, засоби людино-машинного інтерфейсу обмінюються з контролером через мережу M1.

Для функціонування такої системи резервний контролер працює паралельно з основним контролером, отримуючи від нього контекст задач з промислової мережі M2. Конфігурація та програма обох контролерів ідентичні. Передача контексту забезпечується в реальному часі зі швидкістю, що дозволяє у

будь-який момент часу продовжити виконання програми резервним контролером з того місця, де закінчив первинний.

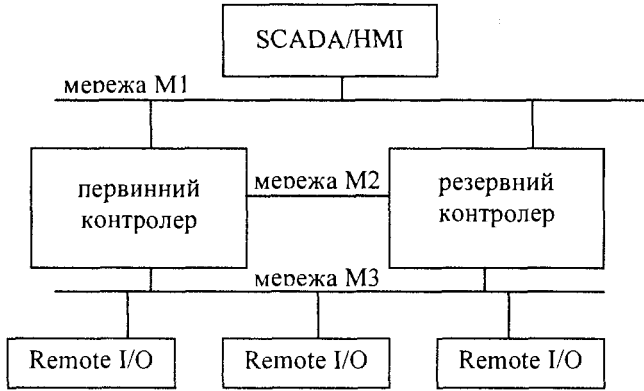


Рис. 2.1. Приклад реалізації системи з резервуванням на основі промислових мереж

У нормальному, тобто штатному режимі роботи системи первинний контролер обмінюється даними з віддаленими входами/виходами та зі SCADA/HMI, а також передає контекст своїх задач резервному контролеру. Резервний контролер в цей час веде себе пасивно щодо всіх перелічених вузлів. За виходу з ладу первинного контролера резервний контролер втратить з ним зв'язок і перейде в режим активної роботи, забезпечуючи обмін процесними даними з входами/виходами по мережі M3 та зі SCADA/HMI по мережі M1. Слід зазначити, що SCADA/HMI, втративши зв'язок з первинним контролером повинна перейти на зв'язок з резервним.

Приклад 2.1

2.1.3. Обмін даними

2.1.3.1. Класифікація за призначенням. Для забезпечення вищевказаних функцій у мережі циркулюють дані та команди. В свою чергу дані можна умовно поділити на два типи: *дані процесу* та *параметричні дані*. Перший тип даних використовується для обміну даними процесу в реальному часі, а другий — для функцій програмування/конфігурування, а також функцій діагностики, що відповідають за ідентифікацію несправності.

2.1.3.2. Класифікація за форматом. Залежно від типу обміну (обмін даними процесу або параметричними даними) дані можуть бути таких форматів:

- числові або аналогові (Integer/Word, Float/Real, Doubleword/Long);
- дискретні або бінарні (Digital/Boolean);
- часові (Time, Date, BCD, BinaryTime)
- масиви та блоки даних (Array, Datablock);
- структурні дані;
- строкові (String).

Перші три групи належать до типів фіксованої довжини. Текстові типи можуть відрізнятися своїм форматом. Передача структурних даних є необов'язковою, однак може значно підвищити функціональність мережі та полегшити роботу розробників системи. До обміну структурними даними будемо також відносити об'єктний підхід, за якого одна прикладна сутність доступається до властивостей та методів об'єкта іншої прикладної сутності. Таким чином, доступ до даних при такому підході — це доступ до методів та властивостей об'єкта.

2.1.3.3. Класифікація за циклічністю відновлення. В одній і тій самій системі можуть бути дані, які необхідно відновлювати постійно (циклічно або періодично) та за необхідності. Можна навести такі схеми відновлення за періодичністю:

– **циклічне відновлення:** відновлення даних проходить регулярно, повторюючись, тобто після чергового відновлення всіх даних — цикл повторюється;

– **періодичне відновлення:** циклічне відновлення з постійною тривалістю циклу; тобто через задані інтервали часу нові дані надходять від джерела до споживача;

– **ациклічне (аперіодичне) при зміні значення** даних або їх стану: дані відправляються від джерела до споживача тільки при зміні їх значення.

– **ациклічне (аперіодичне) за запитом:** дані надходять до споживача після його запиту.

Циклічне відновлення проходить постійно, з мінімально можливими паузами між циклами відновлення даних. Періодичне відновлення теж проходить за циклом, але через рівні інтервали часу, що визначає рамки періоду доставки даних. Обидва способи використовується для обміну даними процесу. Недоліком обох способів є постійне завантаження мережі даними, які передаються. Так, наприклад, у працюючій системі протягом певного часу може не бути ніяких змін у технологічному процесі, але промисловою мережею будуть курсувати одні і ті ж значення даних. Може бути й інша ситуація: в межах циклу опитування даних відбудеться зміна їх значень, які система не встигне передати. Перевагою періодичного способу є можливість визначення мережного трафіку, який витрачається на цей обмін.

Для циклічного та періодичного відновлення характерна циклічність операцій. Тому надалі в порівняльних характеристиках їх будемо називати одним терміном — **циклічно-періодичні**, якщо не буде вказано інше. Циклічно-періодичні операції відновлення можуть проходити шляхом як зчитування даних з джерела, так і шляхом відправки даних самим джерелом.

Альтернативою циклічно-періодичному відновленню для обміну даними процесу може бути відправка джерелом значень ациклічно (в посібнику також використовується термін *аперіодично*, як синонім) тільки у разі їх зміни. Така схема ініціації відправки даних також називається подійно-орієнтована (event-triggered). У цьому випадку значно заощаджуються ресурси промислової мережі, оскільки більшу частину часу дані не змінюються. Недоліком такого способу є незначеність у завантаженні мережі в кожний момент часу.

Ациклічне відновлення за запитом більш підходить до ситуацій, коли у разі виникнення визначеної події споживачу необхідно зчитати певні дані з джерела. Як правило, таким способом користуються при доступі до параметричних даних,

однак він може бути прийнятим і при доступі до даних процесу. При відновленні даних цим способом витрачається додатковий час на формування, відправку та обробку запитів.

2.1.3.4. Класифікація за синхронністю. У деяких системах є необхідність прив'язки ініціації певних дій до конкретних часових або синхронізуючих міток. Це потрібно для *синхронності* (одночасності) виконання цих дій на декількох (або на всіх) вузлах мережі. До синхронних дій можна віднести одночасне відновлення вхідних даних (одночасне зчитування входів вузлів у їх вхідні буфери), одночасне відновлення вихідних даних (одночасний запис значень вихідних буферів вузлів на їх виходи), одночасну передачу даних у мережу.

При передачі даних ідеться про *синхронну* або *асинхронну* передачу. Асинхронна передача не прив'язана до синхронізуючих сигналів і може початися у довільний момент, тоді як синхронна — тільки після появи певного синхросигналу.

Особливий вид синхронних дій — *ізохронна передача*, яка передбачає передачу даних через постійні інтервали часу. Особливістю ізохронної передачі порівняно з періодичною є обов'язкова синхронність роботи внутрішніх тактових годинників, на яких базується ізохронна передача. Інакше кажучи, ізохронна передача більш жорстко визначає рамки періодичності відновлення даних.

Синхронність та ізохронність використовуються, наприклад, за необхідності синхронізувати робочі цикли контролера і мережі. Інший приклад — задача позиціонування з використанням електроприводів PDS, де синхронна та ізохронна передача є принципово необхідною (див. розділ 11).

Приклад 2.2. Основні концепції. Синхронний та періодичний обмін

Завдання. Реалізація промислової мережі дає можливість віддаленому пристрою вводу/виводу (надалі пристрій) відправляти дані у будь-який момент часу (наприклад з використанням мережі CANOpen). Продемонструвати, яким чином буде відбуватися передача значень входів даного пристрою при:

- 1) аперіодичному відновленні, за зміни значення, асинхронно;
- 2) періодичному відновленні (кожні 500 мс), асинхронно;
- 3) аперіодичному відновленні, за зміни значення, синхронно з початком циклу контролера;
- 4) періодичному відновленні (кожні 500 мс), за зміни значення, синхронно з початком циклу контролеру.

Рішення.

Варіант 1. Пристрій опитує входи з максимальною швидкістю. Як тільки значення одного із його входів змінюється, воно відправляється контролеру за наданої можливості передачі.

Варіант 2. Кожні 500 мс пристрій опитує свої входи і за наданої можливості передачі відразу відправляє їх на контролер, незалежно від їх значення.

Варіант 3. Пристрій опитує входи з максимальною швидкістю. Контролер генерує синхросигнал на початку кожного робочого циклу (для прикладу цикл контролеру 100 мс). Після отримання синхроімпульсу, значення входів передається контролеру, якщо їх стан змінився з попередньої передачі.

Варіант 4. Кожні 500 мс пристрій опитує входи. Контролер генерує синхросигнал на початку кожного робочого циклу (для прикладу цикл контролеру 100 мс). Пристрій передає значення входів тільки після отримання синхроімпульсу.

Як бачимо, в третьому випадку відновлення входів пристрою для контролеру буде відбуватися з дискретністю 100 мс, але тільки в момент зміни їх значення. У четвертому випадку значення входів пристрою для контролера буде відновлятися з періодичністю 500 мс. В обох випадках відновлення входів контролеру буде проходити на початку його робочого циклу.

Приклад 2.2

2.1.4. Загальні вимоги до промислових мереж

Промислові мережі споріднені комп'ютерним мережам, однак порівняно з останніми, вимоги до промислових мереж дещо відрізняються. Ідеальний варіант промислової мережі повинен задовольняти таким вимогам.

1. Властивість детермінованості.
2. Завадостійкість та промислові умови експлуатації.
3. Надійність та живучість.
4. Простота, зручність інсталяції та обслуговування.
5. Можливість подачі живлення по кабельній системі мережі.
6. Вільна топологія.

2.1.4.1. Детермінованість. Властивість детермінованості тісно пов'язана з реальним часом, про який вище було неодноразово згадано. Слід звернути увагу, що промислова мережа повинна забезпечити своєчасну доставку та цілісність *даних процесу*. З іншого боку, промислова мережа повинна забезпечити ациклічний обмін *параметричними даними*, які можуть бути передані без чіткого визначення часу їх доставки, однак в межах допустимого. Так, наприклад, одночасно із обміном даними процесу з розподіленими модулями вводу/виводу, необхідно змінювати конфігурацію одного з них.

З вищесказаного можна зробити висновок, що промислова мережа повинна надавати можливість обміну обидвома типами даних, тобто надавати час для *real-time трафіку* (для обміну даними процесу) та *не real-time трафіку* (для обміну параметричними даними). Це можливо за умови визначення пріоритетності повідомлень для обміну даними процесу, або розділення загального мережного часу на циклічно-періодичний обмін даними процесу та ациклічний обмін параметричними даними.

Слід зазначити, що не всі промислові мережі гарантують повну детермінованість. Навіть при чіткому розумінні побудови протоколу, деякі програмно-технічні засоби, наприклад ПК з офісними операційними системами (наприклад Windows) можуть внести певну невизначеність у роботу мережі.

Для комп'ютерних мереж наявність повної детермінованості не є обов'язковою. Для класичного та комутованого Ethernet ця особливість є одним із каменем спотикання визнання її в області промислових мереж. Детальніше про це можна прочитати в розділі 10 даного посібника.

2.1.4.2. Завадостійкість та промислові умови експлуатації. В промислових умовах експлуатації поряд з кабелем мережі може знаходитися силове електро-

обладнання, що може спричинити спотворення корисного сигналу самоіндукуючими паразитними струмами. Для боротьби з цим явищем у промислових мережах використовують спеціальні методи кодування/модуляції бітової послідовності, екранований кабель, захисне зміщення й інше (детальніше див. розділ 3).

Промислові умови експлуатації, які обумовлюються надмірними температурами, вологістю, вібрацією або іншими кліматичними факторами потребують використання кабельної продукції і комутуючих засобів (конектори, роз'єми, коробки й інше) спеціального виконання. В деяких випадках є необхідність в іскробезпечному виконанні мережі, що передбачає додаткові вимоги на середовище та способи передачі даних.

Щодо пило- та вологозахисту технічних засобів у промислових мережах, згідно зі стандартами IEC 529, EN 60529 та ГОСТ 14254-96, всі електричні засоби прийнято класифікувати та маркувати за ступенем захисту *IP-кодом* (Ingress Protection). IP-код — це двохзначне число, перша цифра якого вказує на ступінь захисту електрообладнання від твердих тіл, а друга — від води. Чим більше ці цифри, тим краще оболонка (корпус) засобу захищає його від попадання твердих тіл або рідини.

Перша цифра IP вказує на наступні ступені захисту електрообладнання від твердих тіл: IP 0x — захисту немає; IP 1x — захист від твердих тіл >50 мм (захист від проникнення в корпус руки); IP 2x — захист від твердих тіл >12 мм (захист від проникнення в корпус пальця); IP 3x — захист від твердих тіл >2.5 мм (захист від проникнення в корпус інструменту); IP 4x - захист від твердих тіл >1 мм (захист від проникнення невеликих тіл: гвинти, гайки і т.п.); IP 5x — пилозахисність (пил може попадати всередину в дуже малих кількостях); IP 6x — пилонапроникність (повне виключення попадання пилу);

Друга цифра IP вказує на наступні ступені захисту електрообладнання від води: IP x0 — захисту немає; IP x1 — захист від вертикальних капель (попадання капель зверху на засіб не порушує його роботу); IP x2 — захист від капель, що падають під кутом ≤ 15 град до вертикалі; IP x3 — захист від падаючих бризок під кутом ≤ 60 град до вертикалі (захист від дощу); IP x4 — захист від бризок в будь-якому напрямку; IP x5 — захист від струй води в будь-якому напрямку; IP x6 — захист від морських хвиль (допускається короткочасне погруження в рідину); IP x7 — доступне короткочасне знаходження в рідині на глибині до 1 м; IP x8 — повна водонепроникність (допускається постійне знаходження на глибині 1м);

Якщо в коді замість однієї цифри вказується літера X, то ступінь захисту по цій категорії не обумовлюється.

Згідно з ГОСТ 14254-96, IP код може бути доповнений однією додатковою та однією допоміжною літерами. Перша додаткова літера вказує на захист від доступу до небезпечних частин електрообладнання. Вона вказується тільки у випадках, якщо першій цифри в IP-коді немає або ступінь захисту від доступу до небезпечних частин вища, ніж обумовлена нею. Друга допоміжна літера доповнює інформацію про ступінь захисту, яка вказується в описі самого приладу.

Позначення додатковою літерою ступеня захисту від доступу до небезпечних частин: IP xxAx — захищено від доступу тильною стороною руки; IP xxVx — за-

хищено від доступу пальцем руки; IP ххСх — захищено від доступу інструментом; IP ххDх — захищено від доступу дротом.

Так код IP 23 CS, буде характеризувати таку ступінь захисту: (2) — оболонка захищає людей від доступу до небезпечних частин пальцями рук; захищає обладнання всередині оболонки від попадання зовнішніх твердих предметів діаметром, рівним або більшим 12,5 мм; (3) — оболонка захищає обладнання всередині оболонки від шкідливої дії води в виді дощу; (C) — оболонка захищає людей від доступу до небезпечних частин, якщо вони тримають у руках інструмент діаметром, рівним або більшим 2,5 мм, і довжиною, яка не перевищує 100 мм (інструмент може проникати на всю свою довжину в оболонку); (S) — оболонка піддана випробуванню на відповідність захисту від шкідливих дій внаслідок попадання води, коли всі її частини обладнання знаходяться в стані нерухомості.

Стандарт IEC EN 62262 (попередня версія EN 50102) визначають ступінь захисту електрообладнання від механічної дії (ударостійкість). Цей захист визначається **ІК-кодом**, який вказує на максимальну енергію удару, яку витримує оболонка засобу: ІК00 (0 Дж), ІК01(0,14 Дж), ІК02(0,2 Дж), ІК03(0,35 Дж), ІК04(0,5 Дж), ІК05(0,7 Дж), ІК06(1 Дж), ІК07(2 Дж), ІК08(5 Дж), ІК09(10 Дж) ІК10(20 Дж).

Таким чином промислові умови експлуатації для промислових мереж обумовлюються спеціальними виконаннями технічних засобів та матеріалів, такими як: тип кабелю, ступінь захисту по IP, по ІК, по NEMA, захист від агресивних середовищ, вібростійкість та ін.

2.1.4.3. Надійність та живучість. Надійність системи перш за все залежить від надійності її компонентів та структури мережі. Мережні технології дають можливість будувати системи з резервуванням, що дозволяє збільшити надійність роботи мережі або окремих її компонентів. Резервування окремих, найменш надійних мережних компонентів дозволяє збільшити живучість системи. Однак слід зазначити, що надійність у більшості випадків реалізується на рівні програмно-апаратного забезпечення вузлів, а не мережних компонентів. У свою чергу протоколи промислових мереж повинні забезпечити можливість діагностики мережі, для своєчасного виявлення дефектних вузлів, а самі мережні компоненти повинні мати надійність значно вищу ніж мережні вузли.

У промислових мережах необхідно, щоб при виявленні дефектного вузлу, він автоматично відключався, не порушуючи працездатність усієї мережі. Це здійснюється засобами самодіагностики мережних адаптерів, які можуть перевести вузол у режим аварійної зупинки, відправити сигнал аварії слідкуючому пристрою і відключитись від комунікаційного обміну.

2.1.4.4. Простота, зручність інсталяції та обслуговування. В промислових умовах експлуатації простій обладнання може привести до аварійної зупинки виробничої лінії, що відповідно приводить до втрат різного характеру. Тому обслуговуючий персонал, наприклад служба КВПіА, повинен швидко виявити причину зупинки та замінити дефектну частину системи, при цьому не зупиняючи функціонування мережі. Гаряча заміна вузлів або їх частини — одна з основних вимог до промислових мереж. Процедура заміни або добавлення нового вузла повинна проходити швидко та легко.

Для прикладу, можна привести процедуру додавання нового вузла на шину AS-і, де кабель мережі у необхідному місті просто проколюється самою коробкою-адаптером підключення. При відключенні адаптера гумова ізоляція кабелю повертає його до попереднього стану (див. розділ 5).

Для промислових мереж бажана наявність таких властивостей: мережні компоненти підключаються через конектори, в гіршому випадку через легкодоступні гвинтові або пружинні з'єднувачі, але ніяк не через пайку; мінімальна конфігурація (наприклад, мережна адреса) для вузлів розподіленої периферії виставляється через перемикачі безпосередньо на пристрої; можливість гарячої заміни вузлів та складових мережі.

2.1.4.5. Живлення вузлів по кабелю мережі. Це важлива, але не принципова вимога до промислових мереж. Живлення, як правило, потрібне датчикам та виконавчим механізмам, які підключаються до мережі. Щоб не прокладати живлення окремим кабелем, воно подається тим же самим кабелем, що і цифровий сигнал, а інколи і тією ж самою інформаційною парою проводів, створюючи несучий сигнал для модуляції. Сучасні мережі рівня датчиків, як правило, забезпечують можливість живлення датчиків мережею.

2.1.4.6. Вільна топологія. Потрібна топологія мережі диктується територіальним розміщенням мережних вузлів. Тому в ідеальному варіанті мережа повинна мати вільну топологію, що особливо актуально для мереж рівня датчиків та виконавчих механізмів. Однак більшість промислових мереж мають шинну топологію, рідше — деревовидну або кільцеву. Для деяких мереж топологію можна вибрати, виходячи з вимог та обмежень поставленого завдання.

Для вибору потрібної топології, у деяких мережах використовуються спеціальні адаптери. Інший шлях — побудова інтермереж з використанням міжмережних адаптерів (мостів, шлюзів).

Таким чином промислові мережі виділяються серед комунікаційних систем характеристиками, які зведені в таблицю 2.1.

Таблиця 2.1

ХАРАКТЕРИСТИКИ ПРОМИСЛОВИХ МЕРЕЖ
(Підсумкова таблиця підрозділу 2.1)

Визначення	Промислова мережа — промислова комунікаційна система з послідовною передачею бітів, яка використовує металевий кабель, оптоволокно або радіохвилі для зв'язку міжмікропроцесорними засобами автоматизації
Градація	– рівень датчиків (включає рівень польових засобів); – рівень контролерів
Функціональне призначення	1. Обмін даними процесу в реальному часі (обмін даними процесу). 2. Програмування та конфігурування вузлів (обмін параметричними даними). 3. Діагностика вузлів. 4. Управління станом вузла. 5. Функції резервного переключення між вузлами мережі.
Обмін даними	1. Призначення даних: a. обмін даними процесу; b. обмін параметричними даними.

Обмін даними	<ol style="list-style-type: none"> 2. Формати даних: числові (аналогові); дискретні (бінарні); часові; масиви (блоки даних); структурні дані (+об'єктний підхід);строкові. 3. Спосіб відновлення даних <ul style="list-style-type: none"> – циклічне відновлення(дані процесу); – періодичне відновлення (дані процесу); – ациклічне при зміні значення (дані процесу); – ациклічне по запиту (параметричні дані або дані процесу). 4. Часова синхронізація: <ul style="list-style-type: none"> – синхронна передача; – асинхронна передача.
Вимоги до промислових мереж	<ol style="list-style-type: none"> 1. Властивість детермінованості (для обміну даними процесу). 2. Завадостійкість та промислові умови експлуатації. 3. Надійність та живучість. 4. Простота, зручність інсталяції та обслуговування. 5. Можливість подачі живлення кабельною системою мережі. 6. Вільна топологія.

2.2. Історія розвитку та стандартизації промислових мереж

У своєму розвитку промислові мережі пройшли декілька ключових етапів. В 70-х роках ХХ століття появилися перші промислові мережі (MODBUS-Modicon), а пізніше почали з'являтися нові технології (DataHighWay-AllenBradley, TiWay-Texas Instruments). У середині 80-х років на світовому ринку вже існувала маса різноманітних пропозицій у цій галузі, що заважало нормальній інтеграції обладнання різних виробників. Внаслідок цього для багатьох мереж були видані доступні специфікації, що дало можливість, з одного боку, багатьом виробникам обладнання вибирати найбільш вдалі рішення, а з іншого — зробити ці рішення стандартами де-факто. Таким чином на ринку промислових мереж стали домінувати відкриті системи.

Призначенням перших відкритих стандартів на базі специфікацій була формалізація промислових мереж, забезпечення надійності та стабільності специфікацій, що виключало їх стрімку зміну та закріплювало довіру з боку виробників апаратного та програмного забезпечень. Пізніше багато зі стандартів закріплювались на державному рівні і отримували юридичну силу, що обумовлювало витіснення закритих систем у цих країнах. Однак, розширення ринку збуту приводило до необхідності стандартизації та уніфікації мережних протоколів на міжнародному рівні. Бажання найбільш великих компаній зробити основним саме свій стандарт привело до так званої «війни промислових мереж» («fieldbus war»). Виділимо п'ять етапів розвитку стандартів промислових мереж:

- 1986—1990 рр. — поява відкритих стандартів промислових мереж;
- 1990—1994 рр. — німецько-французька війна стандартів (PROFIBUS, FIP);
- 1995—1998 рр. — стандартизація в тупиковому стані;
- 1999—2000 рр. — знаходження компромісу;
- 2000—2008 рр. — вдосконалення та розширення стандартів МЕК.

У 1984 році комітет TC65C Міжнародної Електротехнічної Комісії **МЕК (IEC)** почав процес розробки єдиного універсального стандарту промислової мережі. Були визначені вимоги для відкритої промислової мережі, пристроїв віддаленого вводу-виводу, контролерів, узгоджувачів пристроїв та ін. Ставилось завдання, щоб така універсальна мережа забезпечувала комунікаційні запити на всіх рівнях багаторівневої системи автоматизації. Найбільшої популярності тоді набули французький стандарт FIP та німецький — PROFIBUS, які були запропоновані в якості основи для універсальної промислової мережі. Однак різні принципи функціонування цих систем зробили їх кращими в конкретних галузях, тому універсальним вони бути не могли. Єдиний стандарт повинен був увібрати у себе найбільш вдалі рішення, тому системи постійно дороблялися, що призвело до появи стандартів WorldFIP та проекту ISP.

Паралельно розробкою базових концепцій стандарту займався комітет SP50 американської організації ISA, що обумовило появу першої версії стандарту IEC 61158-2, основна структура якого до сьогодні залишилась без змін.

В результаті американська організація Fieldbus Foundation в рамках ISA, але без участі МЕК, розробила стандарт промислової мережі під назвою Foundation Fieldbus (FF). Організація Fieldbus Foundation створилась у результаті об'єднання двох груп InterOperable Systems Project (ISP) та WorldFIP North America, тому їх стандарт увібрив в себе елементи мереж WorldFIP та PROFIBUS.

В Європі закріплення єдиного стандарту, який витіснив всі інші промислові мережі, нікому не був вигідний, бо був несумісним з мережами, які вже використовувались. У зв'язку з цим у межах європейської організації **CENELEC** був знайдений компроміс: всі національні стандарти при розгляді компілювались в єдиний стандарт у тій же редакції. Тобто кожна частина багатопрошаркового стандарту була копією відповідного національного стандарту і функціонувала як самостійна частина. Щоб зробити CENELEC більш легким у використанні, різні промислові мережі були пов'язані єдиною областю застосування: EN 50170 — загального призначення, EN 50254 — високошвидкісні системи обміну невеликими об'ємами даних, та EN 50325 — мережі, які базуються на CAN-технологіях. На більш пізніх етапах європейського процесу стандартизації національний комітет запропонував включити в цю групу стандартів американські мережі FF, DeviceNET та ControlNet. Існуючі стандарти EN наведені у табл. 2.2.

Таблиця 2.2

ЄВРОПЕЙСЬКІ СТАНДАРТИ ПРОМИСЛОВИХ МЕРЕЖ

стандарт CENELEC	Мережа
EN 50170-1	P-Net
EN 50170-2	PROFIBUS
EN 50170-3	WorldFIP
EN 50170-A1	Foundation Fieldbus
EN 50170-A2	PROFIBUS-PA
EN 50170-A3	ControlNet
EN 50254-2	INTERBUS
EN 50254-3	PROFIBUS-DP
EN 50254-4	WorldFIP (FIPIO)
EN 50325-2	DeviceNet
EN 50325-3	SDS
EN 50325-4	CANOpen
EN 50295-2	AS-Interface

Організація MEK під впливом Fieldbus Foundation у 1996 р. висунула на головування проект промислової мережі на базі FF та WorldFIP. Однак, країни з домінуючими інсталяціями PROFIBUS та інших «європейських» мереж, проголосували проти цього проекту. Зав'язалися нові баталії з використанням юридичних та політичних прийомів, які закінчилися влітку 1999 року підписанням «Меморандуму про порозуміння» між основним гравцями ринку (Fieldbus Foundation, Fisher Rosemount, ControlNet International, Rockwell Automation, PROFIBUS user organization та Siemens). Внаслідок прийнятого меморандуму був створений всеохоплюючий стандарт IEC 61158, який був прийнятий у 2000 р. та включав найбільш відомі системи промислових мереж.

На відміну від стандартів CENELEC, де специфікації повністю влились у стандарт, MEK зберіг оригінальну структуру вихідного проекту. Архітектура промислових мереж представлена аналогічно OSI-моделі, на яку посилається базова специфікація MEK 61158. На початку у ньому було визначено вісім різних типів мереж, які відображали вісім технологій промислових мереж: FOUNDATION Fieldbus H1, FOUNDATION Fieldbus HSE, ControlNet, PROFIBUS DP, P-Net, Interbus, SwiftNet, WorldFIP.

На сьогодні до стандарту включено ще ряд рішень. Кожна з мереж представлена своїм профілем (CP), який належить до сімейства профілів (CPF). Представлені наступні технології CPF (у дужках вказані назви профілів CP):

- CPF1. Foundation Fieldbus (FF H1, FF HSE, FF H2);
- CPF2. CIP (ControlNet, Ethernet/IP, DeviceNet);
- CPF3. PROFIBUS & PROFINET (PROFIBUS DP, PROFIBUS PA, PROFINET CBA, PROFINET IO CC-A, PROFINET IO CC-B, PROFINET IO CC-C);
- CPF4. P-NET (P-NET RS-485, P-NET RS-232, P-NET on IP);
- CPF5. WorldFIP (WorldFIP, WorldFIP with subMMS, WorldFIP minimal for TCP/IP);
- CPF6. INTERBUS (INTERBUS, INTERBUS TCP/IP, INTERBUS minimum subset)
- CPF7. Видалений в сьогоденній редакції;
- CPF8. CC-Link (CC-Link/V1, CC-Link/V2, CC-Link/LT);
- CPF9. HART;
- CPF10. Vnet/IP;
- CPF11. TCnet (TCnet-star, TCnet-loop);
- CPF12. EtherCAT;
- CPF13. ETHERNET Powerlink (EPL);
- CPF14. EPA (NRT, RT, FRT);
- CPF15. MODBUS-RTPS (MODBUS TCP, RTPS);
- CPF16. SERCOS (SERCOS I, SERCOS II, SERCOS III);
- CPF17. RAPIEnet;
- CPF18. SafetyNet p (RTFL, RTFN)

Тепер кількість стандартів MEK, присвячених промисловим мережам значно збільшилась. Всі вони об'єднані у єдину групу під назвою «Industrial communication networks», після розгляду структури якої можна зробити такі висновки:

1. Стандарти IEC 61158 охоплюють практично всю кількість необхідних вимог для розробки пристроїв, стосовно заявлених у них типів промислових мереж, що вирішує проблему сумісності.

2. Стандарти IEC 61784-1 та IEC 61784-2 систематизують IEC 61158, що полегшує їх використання не тільки розробникам апаратного та програмного забезпечення, а й проєктантам.

3. В стандарті IEC 61784-2 додатково до комунікаційних профілів наведені вимоги до RealTime Ethernet, що обумовлює надійність використання Ethernet у промислових комунікаціях та полегшує його інтегрування в загальну мережу підприємства (див. розділ 10).

4. Для функціонально небезпечних систем існують стандарти IEC 61784-3, в яких описуються вимоги до використання промислових мереж окремих профілів та необхідних для цього рішень.

5. Методика планування, монтажу, перевірки правильної роботи та адміністрування промислових мереж наводяться у стандартах IEC 61918 та IEC 61784-5.

Таким чином, на сьогодні стандарт охоплює всі аспекти розробки, впровадження та експлуатації промислових мереж. Наявність добре продуманої нормативної документації дає можливість уникнути помилок як при розробці апаратного та програмного забезпечення, так і при проєктуванні інтегрованих автоматизованих систем з використанням промислових мереж.

Стрімка поява нових специфікацій показує зацікавленість світової спільноти у цих стандартах, а отже і закріплення його в якості базового на державних рівнях. Очевидно, що у найближчому майбутньому очікується наповнення стандартів новими мережними технологіями.

2.3. Промислові мережі в контексті моделі ISO OSI

Об'єднання в єдину цифрову мережу кількох пристроїв — це тільки початковий крок до ефективної й надійної роботи системи зв'язку між ними. На додаток до апаратних вимог пред'являється також ряд програмних. Там, де системи зв'язку або мережі, гомогенні (однорідні), тобто поєднують пристрої від одного виробника, ці проблеми, як правило, вирішені. Але коли йдеться про побудову мережі із пристроїв різних виробників, ці проблеми носять множинний характер.

Системи, що є унікальними (їх робить і підтримує тільки один виробник) і працюють за унікальними протоколами зв'язку, одержали назву «Закритих систем» (closed/proprietary systems). Більшість таких систем зародилося у часи, коли проблема інтеграції виробів різних виробників не вважалася актуальною. «Відкриті системи» (open systems) приводять у відповідність специфічні вимоги до інтересів усіх. Тільки при використанні принципів відкритих систем інтеграція виробів різних виробників в одну мережу може бути вирішена без особливих проблем.

У 1978 році Міжнародною організацією зі стандартизації (*ISO*) на противагу закритим мережним системам, і з метою вирішення проблеми взаємодії відкритих систем з різними видами обчислювального устаткування які працюють за різними стандартами і протоколами, була запропонована «Описова модель взаємо-

зв'язку відкритих систем» (*OSI-модель*, Open System Interconnection Reference Model). На сьогодні у стандартах МЕК визначена своя модель архітектури промислових мереж. Однак у технічній документації до опису функціонування мереж поки що прийнято давати їх архітектуру в контексті OSI, що допомагає краще зрозуміти принципи її функціонування. З іншого боку, протоколи мереж, які були розроблені задовго до виникнення еталонних моделей відкритих систем доводять до системи OSI (наприклад, MODBUS у системі MODBUS-IDA). Тому ознайомлення з будь-якою мережею через призму OSI часто носить тільки інформативний характер про протоколи та сервіси рівнів.

У моделі OSI для спрощення структури більшість мереж організуються у набори рівнів або шарів (layers), кожен з яких має своє призначення. Кількість рівнів, їх назва та склад відрізняються залежно від мережі. Однак, у всіх мережах цілком кожного рівня є представлення деяких сервісів для вищого рівня і підтримки зв'язку з однойменним рівнем на іншій пристрої. Правила і домовленості, які використовуються у даному спілкуванні, називаються *протоколом* рівня (layer protocol).

На кожному з рівнів обробка даних згідно з протоколом включає в себе перетворення інформації з одного виду в інший, розбивку даних на частини та додавання службової інформації.

При розробці мереж на кожному з рівнів визначають:

- механізм ідентифікації відправників та отримувачів та систему адресації;
- правила переносу даних: напрямки, логічні канали та пріоритети;
- контроль помилок при передачі даних, тип і дії при їх виникненні;
- правило розбивки даних та їх об'єднання;
- управління потоком даних та узгодження швидкодії;
- маршрутизація — вибір шляху передачі даних.

Апаратуру або/і програму, яка займається завдання кожного з рівнів будемо називати *об'єктом рівня* або *сутністю рівня*. Виділяють такі об'єкти рівнів: прикладний об'єкт (прикладна сутність), транспортний об'єкт (транспортна сутність), мережний об'єкт (мережна сутність), каналний об'єкт (канална сутність) та об'єкт фізичного рівня (сутність фізичного рівня або фізичний інтерфейс).

Для забезпечення зв'язку сутність кожного рівня займається обміном даними із сутністю такого ж рівня на іншому вузлі. Однак цей зв'язок проходить не безпосередньо, а за наступною схемою (рис. 2.2): дані, які необхідно передати надходять до самого верхнього рівня, який обробляє їх і передає нижчому рівню і т.д. аж до 1-го рівня, який займається безпосередньою передачею даних фізичним середовищем. На іншому вузлі, дані, які приймаються обробляються кожним рівнем у протилежному напрямку. Обробка інформації на кожному рівні відбувається відповідно до протоколу.

Кожен із рівнів у мережі надає набір сервісів (служб) для реалізації певних завдань. *Сервіс* або служба (Service) — це набір операцій, які нижній рівень представляє верхньому. Сервіс визначає, які саме операції рівень буде виконувати, але не обумовлює, яким чином.

Між кожною парою суміжних рівнів знаходиться *інтерфейс* (interface), який визначає набір примітивних операцій, які надаються верхньому рівню від ниж-

нього. Під інтерфейсом можна розуміти ті програмні чи апаратні функції, які надаються верхньому рівню для доступу до своїх сервісів.

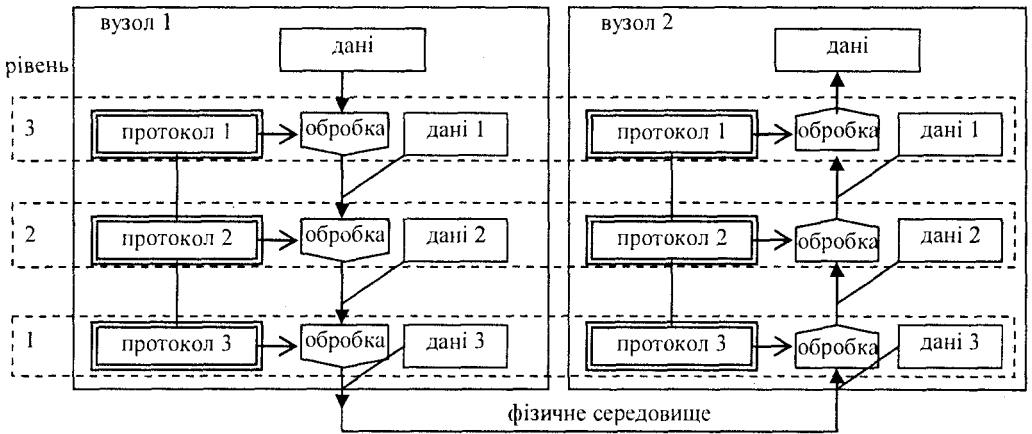


Рис. 2.2. Передача даних від одного вузла до іншого через мережу з багаторівневою архітектурою

Набір рівнів і протоколів називається *архітектурою мережі*. Список протоколів, які використовуються системою (по одному протоколу на рівень), називається *стеком протоколів*.

Ієрархія моделі OSI включає 7 рівнів:

1. Фізичний рівень (Physical Link Layer). Даний рівень описує механічні та електричні характеристики, а також визначає фізичне середовище для передачі даних. Тобто сутність фізичного рівня займається передачею та прийомом бітів по каналу зв'язку. Принциповими питаннями тут є: який тип сигналу (напруга, струм, світло і інші) і якої величини використовується для передачі логічної «1» та «0»; тривалість одного біту; напрямок передачі даних; коли розпочати і закінчити передачу; кількість та призначення фізичних каналів зв'язку, спосіб стикування (типи роз'ємів) і т.ін.

Фізичний рівень повинен забезпечити достовірну доставку бітів від передавача до приймача.

2. Рівень передачі даних (Data Link Layer) або каналний рівень. Його завдання — забезпечити можливість та надійність передачі даних на фізичному рівні. Всі дані, які передаються з верхнього (мережного) рівня, передаються у вигляді кадрів, розмір яких залежить від конкретної мережі. За допомогою спеціальних алгоритмів проводиться контроль за правильною доставкою на фізичному рівні з подальшою реакцією на помилки. Окрім підтримки каналу зв'язку та контролю за помилками, у мережах з загальним доступом до спільно використовуваного фізичного каналу, на цьому рівні регламентується алгоритми та процедури доступу.

Канальний рівень повинен забезпечити достовірну доставку даних від одного вузла до іншого в одній і тій самій мережі.

3. Мережний рівень (Network Layer). Основне із завдань — це визначення маршруту пересилки пакетів від джерела до приймача. Мережа може являти собою об'єднання підмереж, між якими виникає необхідність в обміні даними. Оскільки підмережі можуть бути різнорідними за природою, завданням мережного рівня є забезпечення прозорості між ними. Окрім цих функцій, на даний рівень накладаються також функції визначення часових затримок пакетів у мережах, питання синхронізації, часу передачі і т.ін.

Мережний рівень повинен забезпечити достовірну доставку даних від одного вузла до іншого в різних мережах, об'єднаних в одну інтермережу.

4. Транспортний рівень (Transport Layer). Основна функція — прийняти дані від сеансового рівня (або прикладного, якщо сеансовий відсутній), при необхідності розбити їх на частини, передати їх мережному рівню і гарантувати прибуття їх у правильному порядку до місця призначення. Цей рівень забезпечує передачу даних безпосередньо між прикладними програмами.

Транспортний рівень повинен забезпечити достовірну доставку даних від однієї програми (прикладного Процесу) до іншої, які функціонують на одному вузлі або на різних вузлах у мережі.

5. Сеансовий рівень (Session Layer). Дозволяє організовувати сеанси обміну між прикладними програмами (встановлення та розрив з'єднання тощо).

6. Рівень представлення (Presentation Layer). Служить для перетворення форматів даних із одного в інший.

7. Прикладний рівень (Application Layer). Містить набір протоколів для доступу до даних прикладного Процесу.

Прикладний рівень повинен забезпечити одній прикладній програмі доступ до об'єктів іншої прикладної програми через систему домовленостей.

Більшість промислових мереж підтримують фізичний рівень, рівень передачі даних і прикладний рівень. Як винятки існують протоколи промислових мереж, що реалізують усі сім рівнів OSI-Моделі, наприклад LonWorks.

Функціонування мережі в контексті моделі OSI продемонструємо на прикладі.

Приклад 2.3. Основні концепції. Функціонування мережі в контексті OSI.

Завдання. На вузлі 1 (рис. 2.3) функціонують три прикладні програми (надалі Процеси) під умовними позначками А, В, С, а на вузлі 2 — Процеси D, E, F. *Необхідно:* показати передачу даних від Процесу В до Процесу Е, використовуючи мережу описану на 1-му, 2-му, 3-му, 4-му та 7-му рівні моделі OSI.

Рішення. Процеси С та Е спілкуються за єдиними правилами обміну, які визначені протоколом 7-го рівня. Частково цей протокол визначає семантику спілкування, тобто відповідність символічних позначень до функцій, які необхідно виконати, та дані, які необхідно передати з цими функціями. Таким чином між двома прикладними програмами ведеться обмін кодом та даними, з правил протоколу даного рівня, які називають *APDU* (Application Protocol Data Unit — модуль даних протоколу прикладного рівня). Оскільки Процеси В і Е повинні якимось взаємодіяти між собою, то за передачу даних між цими Процесами відповідає транспортний рівень.

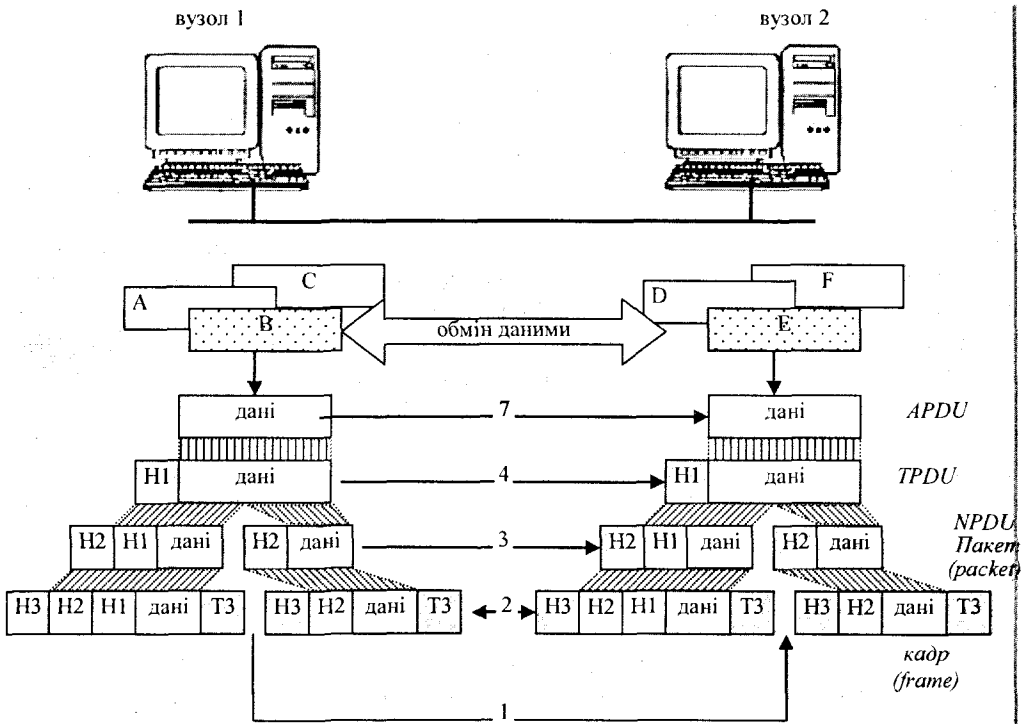


Рис. 2.3. Обробка даних при обміні даними через мережу

Для транспортного рівня код і дані прикладного рівня — це дані, які треба передати від одного Процесу до іншого. Для цього необхідно використати правила адресації цих Процесів, щоб правильно доставити дані та щоб їх отримав Процес E, а не скажімо, Процес F. Ці правила визначаються протоколом транспортного, тобто 4-го рівня. Для того, щоб транспортний об'єкт на вузлі 2 міг ідентифікувати Процес, якому передають дані (у нашому випадку E), транспортний об'єкт на вузлі 1 додає допоміжні дані до тих, які він отримав у прикладного рівня. Дані, які додаються на кожному рівні перед даними з вищого рівня, називаються заголовком (header). Таким чином, для ідентифікації Процесу отримувача транспортний об'єкт відправника додає його адресу (E) у заголовок H1 і передає управління мережному рівню. Заголовок разом із даними на транспортному рівні прийнято називати **TPDU** (Transport Protocol Data Unit — модуль даних транспортного протоколу).

Мережний рівень у свою чергу повинен забезпечити доставку модулів TPDU від вузла до вузла, які можуть міститись у різних за принципом функціонування підмережах. Для цього до TPDU додається заголовок з адресою приймача та відправника (у нашому випадку H2), який необхідний маршрутизаторам (router) для знаходження правильного шляху до визначеної підмережі. Окрім того, на даному рівні дані можуть бути розбиті на частини, для можливості їх передачі на нижніх рівнях (це саме може відбуватися на всіх рівнях). Ці частини разом із заголовками називаються **пакетами** (packet) або **NPDU**.

(Network Protocol Data Unit — модуль даних мережного протоколу). У нашому випадку модуль TPDU на мережному рівні ділиться на два пакети з заголовками H2, які передаються послідовно один за одним канальному рівню.

Пакети, які отримує канальний рівень, вставляються у кадри (frame). Кадри мають заголовок, в якому міститься інформація про адресу вузла отримувача та/або відправника в локальній мережі та/або допоміжні службові коди. Отримувач може бути у свою чергу тільки допоміжним пристроєм для забезпечення передачі даних в іншу підмережу, тому може і не бути кінцевим вузлом призначення. Але для каналного рівня це не важливо, тому що функції маршрутизації належать мережному рівню. Окрім заголовка, в кінці до пакета може добавлятися кінцевик (terminator) з службовою інформацією, наприклад, даними контролю за правильною передачею. Безпосередня передача кадру по бітах забезпечується обладнанням фізичного рівня.

Отримання даних та їх обробка проходить за тією ж схемою з точністю до навпаки. Засоби каналного рівня на вузлі 2 отримують кадр, визначають по заголовку H3, що кадр призначений саме їм, за T3 перевіряють правильність отриманих даних, вилучають пакет і передають мережному рівню. Це саме вони роблять і з другим пакетом. Мережний рівень у свою чергу вилучає модулі TPDU з пакетів, об'єднує їх і передає транспортному рівню, який за заголовком H1 ідентифікує Процес E і передає йому дані.

Це спрощений і далеко неповний перелік дій з даними на кожному рівні, який дає представлення про послідовність і принцип обробки даних при передачі.

Приклад 2.3

2.4. Основні робочі характеристики промислових мереж

При ознайомленні з можливостями промислової мережі загалом та конкретного вузла, звертають увагу на основні її робочі характеристики, за якими можна в цілому оцінити її функціональні можливості, а саме:

- прикладні сервіси, що використовуються, їх тип та модель функціонування;
- використання профілів;
- методи доступу до середовища передачі;
- модель адресації доставки кадрів;
- контроль доставки даних: використання сервісів на різних рівнях мережі — з підтвердженням/без підтвердження;
- топологія мережі;
- максимально можлива кількість вузлів;
- максимальна бітова швидкість;
- середовище передачі;
- сумарна максимальна довжина ліній зв'язку;
- використання стандартних інтерфейсів;
- можливість сегментації;
- можливість об'єднання в інтермережу;
- транспортні сервіси.

Розглянемо більш детально ці характеристики.

2.4.1. Прикладні сервіси, інтерфейс прикладного рівня

На прикладному рівні визначаються правила і семантика обміну між двома прикладними сутностями, які в свою чергу забезпечують обмін між **прикладними Процесами** або просто **Процесами** (коли слово «Процес» буде зустрічатися в тексті з прописної літери, буде вважатися прикладний процес). Для того щоб один прикладний Процес зміг спілкуватися з іншим, необхідно визначити «мову» спілкування і тип діалогу. Сервіси прикладного рівня повинні забезпечити необхідну функціональність мережі (див. табл. 2.1). Можна виділити такі сервіси, які підтримуються прикладним рівнем:

1. обмін даними процесу в реальному часі;
2. програмування/конфігурування вузла;
3. діагностичні сервіси;
4. управління станом вузла;
5. функції резервування.

2.4.1.1. Інтерфейс прикладного рівня. Для користування сервісами прикладного рівня користувачу надається інтерфейс прикладного рівня (рис. 2.4). Інтерфейс прикладного рівня може являти собою комунікаційні функції, доступ до таблиці мережних об'єктів (мережні змінні, словник об'єктів) тощо. Деякі програмно-технічні засоби надають можливість графічного конфігурування обміну даними на прикладному рівні, наприклад через мову функціональних блоків (FF, LONWorks), тобто графічний вигляд інтерфейсу прикладного рівня.

У будь-якому випадку інтерфейс прикладного рівня дозволяє прикладній програмі доступитися до сервісів прикладного рівня. Прикладні Процеси, між якими необхідно встановити зв'язок, можуть знаходитися на різних вузлах або на тому самому вузлі, суть від цього не змінюється, адже обмін проходить саме між ними, а не двома вузлами. Створенням інформаційного каналу між прикладними Процесами займаються нижні рівні. Розглянемо, яким чином реалізуються дані сервіси на прикладному рівні.

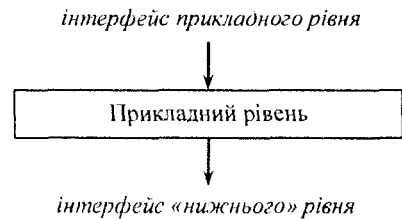


Рис. 2.4. Організація роботи прикладного рівня

2.4.1.2. Взаємодія між прикладними Процесами. Для того, щоб прикладні Процеси на вузлах могли обмінюватися даними, необхідно налаштувати між ними зв'язок. Залежно від реалізації такого зв'язку можна виділити три *моделі взаємодії між прикладними Процесами*:

- модель Клієнт-Сервер (Client-Server);
- модель Видавець-Абонент (Publisher-Subscriber);
- модель Виробник-Споживач (Producer-Consumer);

Модель Клієнт-Сервер передбачає взаємодію тільки двох прикладних Процесів: Процесу-запитувача (Клієнт) та Процесу-відповідача (Сервер). Замовлення послуг проводиться за допомогою *запитів* (request). Сервер, обробивши запит повертає *відповідь* (response). Структура запиту і відповіді залежить від реаліза-

ції протоколу. У виродженому варіанті це можуть бути дані для запису (в запиті) та для читання (у відповіді).

Модель Видавець-Абонент (Publisher-Subscriber) забезпечує зв'язок між декількома прикладними Процесами, один з яких Видавець а інші Абоненти. Процедура передачі даних називається *публікацією* (publication). Цей тип обміну найбільш підходить для передачі даних у багатоадресному режимі, оскільки прикладних Процесів-Абонентів може бути декілька. Таким чином у певний момент часу дані надходять від прикладного Процесу-Видавця всім Процесам-Абонентам. Залежно від того, який прикладний Процес генерує публікацію, виділяють два типи моделі Видавець-Абонент (рис. 2.5):

– *pull model*, коли момент публікації визначає прикладний Процес одного із вузлів, який у необхідний момент відправляє запит на публікацію. У деяких системах це може бути спеціально виділений прикладний Процес, що називається Pull Publishing Manager;

– *push model*, коли момент публікації визначає прикладний Процес-Видавець, наприклад при зміні цих даних або через певні проміжки часу.

Прикладний Процес у системі для одних даних може бути Видавцем, а для інших — Абонентом.

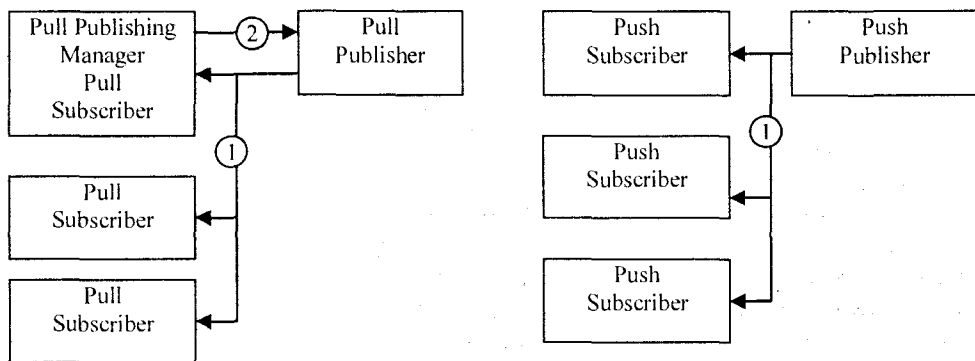


Рис. 2.5. Функціонування моделі Видавець-Абонент:
pull model — ліворуч, push model — праворуч,
1 — публікація, 2 — запит на публікацію

Модель Виробник-Споживач (Producer-Consumer) з точки зору користувача аналогічна моделі Видавець-Абонент, за винятком того, що для адресації отримувачів використовується фільтрація за ідентифікатором повідомлень типу Виробник-Споживач-msg. Як правило фільтрація повідомлень проходить уже на каналному рівні, однак для мереж на базі Ethernet, цей процес може проходити на більш високих рівнях. Прикладний Процес, який у широкомовному режимі видає (виробляє) дані в мережу називається Виробником, а який їх приймає — Споживачем. Усі інші характеристики аналогічні моделі Видавець-Абонент, тому

при розгляді функціонування мереж на прикладному рівні — ці моделі ототожуються.

Слід зазначити, що у стандарті МЕК 61158-1 виділені тільки дві моделі взаємодії між прикладними процесами Клієнт-Сервер та Видавець-Абонент.

2.4.1.3. Ідентифікація даних. Модель обміну між прикладними Процесами вказує на спосіб передачі даних між ними, однак не визначає спосіб ідентифікації цих даних. Додатково необхідно визначити місцезнаходження та формат даних як у прикладному Процесі вузла-джерела, так і вузла-отримувача.

Можна виділити два *способи ідентифікації даних*: указати необхідні дані для обміну на початку функціонування мережі або вказувати місце знаходження та формат даних безпосередньо в момент обміну між прикладними Процесами. Перший спосіб будемо називати *ідентифікованим обміном* (identified data), а другий — *обміном повідомленнями* (messaging).

При ідентифікованому обміні, перед операційним функціонуванням мережі (перед фазою обміну даними між прикладними Процесами) визначають дані на джерелі та відповідні їм дані на приймачі. Іншими словами, дані між прикладними Процесами пов'язують інформаційними зв'язками. Визначені дані в ідентифікованому обміні будемо називати *ідентифікованими даними*. Таким чином, при встановленні зв'язку між прикладними Процесами, дані, які будуть надходити, будуть розміщуватись у наперед сконфігурованих комірках пам'яті, з наперед визначеним форматом.

Для мереж рівня датчиків призначення даних для конкретного пристрою визначаються типом самого пристрою та його специфічними особливостями. Як правило, в таких системах для кожного типу пристроїв створюють свій набір параметрів, де і визначається його поведінка та наповнення даних. Такий набір об'єднують у *Прикладний Профіль* пристрою. Профілювання пристроїв дає можливість легко інтегрувати однотипні пристрої різних виробників, що дає додаткові зручності при конфігуруванні мережі.

При обміні повідомленнями тип необхідної операції кодується у повідомленні, яке може вмішувати також дані та інформацію про їх ідентифікацію. Змістове наповнення повідомлення визначається у момент передачі. Перевага такого способу перед ідентифікованим обміном полягає у визначенні місце знаходження необхідних даних у будь-який момент часу, недоліком — додаткове виділення додаткових часових ресурсів на процес ідентифікації даних.

Для обміну даними процесу, зокрема циклічно-періодичного та ациклічного способу за зміни значення, більше підходить ідентифікований обмін. Для обміну параметричними даними більш підходить обмін повідомленнями. Тому в найкращому випадку промислова мережа повинна надати два типи прикладних сервісів для різних завдань:

- ідентифікований обмін для даних процесу, що циклічно-періодично оновлюються;
- обмін повідомленнями для даних процесу, що потрібні за запитом, або для параметричних даних (конфігурування, програмування, діагностика тощо).

2.4.1.4. Моделі сервісів прикладного рівня. Функціонування сервісів прикладного рівня, що забезпечують обмін даними можна розглядати в контексті

моделей взаємодії між прикладними Процесами в поєднанні зі способом ідентифікації даних. Таким чином, для промислових мереж можна виділити чотири моделі функціонування сервісів прикладного рівня для обміну даними:

1. Клієнт-серверна модель обміну повідомленнями; представники: MODBUS (MBAP), CANOpen (SDO), FF (незаплановані повідомлення), WorldFIP (Aperiodic Traffic).

2. Клієнт-серверна модель ідентифікованого обміну (Polling); представники: Profibus DP (DP-V0/V1), INTERBUS(PDC), AS-I.

3. Модель Видавець-Абонент / Виробник-Споживач ідентифікованого обміну; представники: WorldFIP (Periodic Traffic), FF (заплановані повідомлення), RTPS, Profibus DP (DP-V2), CANOpen (PDO), LON-Works (NVT).

4. Модель Видавець-Абонент / Виробник-Споживач обміну повідомленнями; представники: CIP (Explicit Message Connection), LonWorks (SNVT), FF (VCR розсилка звітів).

Слід зазначити, що наведені моделі не є загальноприйнятими і можуть відрізнятися від наведених в інших джерелах. Наведені моделі використовуються тільки для зручності сприйняття матеріалу в даному посібнику.

Клієнт-серверна модель обміну повідомленнями між прикладними Процесами функціонує таким чином. Прикладний Процес-Клієнт формує повідомлення для Процесу-Серверу, в якому вказує необхідну функцію та дані, що її уточнюють. Він ініціює запит, в якому це повідомлення відправляється Процесу-Серверу. Той робить зворотну операцію декодування повідомлення. Провівши необхідні операції, які вказані в повідомленні, Процес-Сервер генерує повідомлення-відповідь, яке відправляє Процесу-Клієнту. Протокол прикладного рівня визначає семантику формування повідомлення-запиту і повідомлення-відповіді.

Приклад 2.4. Основні концепції. Клієнт-серверна модель обміну повідомленнями.

Завдання. Навести приклад роботи клієнт-серверної моделі обміну повідомленнями.

Рішення. Припустимо, що прикладному Процесу вузла 1 (рис. 2.6) необхідно прочитати значення змінної А з Процесу на вузлі 2 і занести результат у свою змінну В. Для цього користувачу (інженеру) необхідно скористатися відповідним інтерфейсом. Це може бути візуальний інтерфейс або прикладна функція типу:

ЧИТАТИ_МЕРЕЖ_ЗМІННУ_(А, В),

де: А — назва змінної з віддаленого вузла, а В — назва змінної, куди буде записуватись результат.

У цьому випадку Процес вузла 1 виступає як Клієнт, а вузла 2 — як Сервер. Вузол 1 робить запит, який складається з 2-х байт. Перший байт вказує на команду, тобто функцію (в нашому прикладі «R» — читання змінної), другий — являє собою аргумент даної функції. Оскільки використовується функція — це читання, то аргументом буде об'єкт для читання (в нашому випадку змінна А). Єдиний протокол прикладного рівня робить такий формат запиту зрозумілим для Процесу вузла 2. Обробивши запит Клієнта, Процес-Сервер зчитує зна-

чення змінної A (=10) і направляє відповідь із 3 байт: функція — «r» (відповідь на читання), аргумент відповідної функції і безпосередньо результат. На рисунку не показане проходження запиту та відповіді через всі рівні мережі (приклад 2.3), тому вони прямують безпосередньо між Процесами.

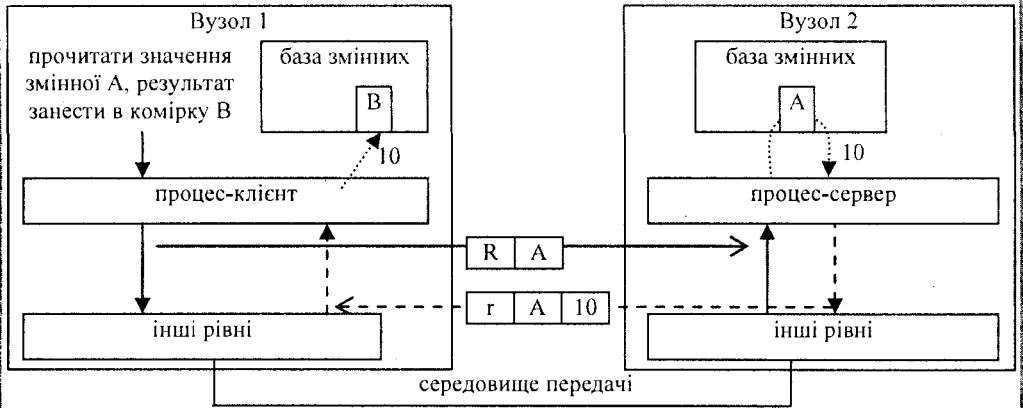


Рис. 2.6. Приклад організації роботи клієнт-серверної моделі обміну повідомленнями

Приклад 2.4

Як правило, для клієнт-серверного обміну повідомленнями, дані при читанні/запису групують. Так, наприклад, при циклічно-періодичному опитуванні системою SCADA/HMI не згрупованих даних з контролера, вона буде звертатися до нього за кожною змінною окремим запитом, що значно зменшить продуктивність обміну. В найгіршому випадку для 100 змінних треба буде генерувати 100 повідомлень-запитів і відповідно отримати 100 повідомлень-відповідей. Крім того, кожний запит обрамляється службовими байтами (символами), які можуть бути більшими за корисну інформацію. Тому на практиці при читанні та при записі користуються повідомленнями-запитами групового пересилання даних з полями «номер початкової змінної» і «кількість змінних». *Для збільшення швидкодії мережі всі змінні, які беруть участь в обміні, бажано групувати разом, щоб вони зчитувались по можливості одним повідомленням-запитом!*

Клієнт-серверна модель обміну повідомленнями насамперед підходить для ациклічних операцій (ациклічний обмін процесу, обмін параметричними даними). У потрібний момент часу Клієнт може відправити повідомлення-запит на виконання будь-якої функції, передбаченої протоколом. У промислових мережах для цього найчастіше використовуються такі формати повідомлень:

- читання/запис значень змінних;
- управління роботою пристроїв (старт, стоп, ініціалізація);
- діагностика пристроїв;
- конфігурування пристроїв та завантаження програми.

У клієнт-серверній моделі обміну повідомленнями можливий також циклічно-періодичний обмін даними процесу, однак при цьому витрачається зна-

чний час на формування та передачу повідомлення для тієї самої операції. Тому для циклічно-періодичних операцій краще підходить ідентифікований обмін.

У *клієнт-серверній моделі ідентифікованого обміну*, яку також називають модель з циклічним полінуванням (*Polling*), Процес-Клієнт ініціює доставку ідентифікованих даних до Процесу-Серверу, а у відповідь отримує ідентифіковані дані призначені йому. Таким чином, на відміну від клієнт-серверної моделі обміну повідомленнями, під час обміну даними немає необхідності в їх ідентифікації, оскільки дані ідентифіковані в передопераційному режимі.

Хоч клієнт-серверна модель ідентифікованого обміну витрачає менше ресурсів порівняно з обміном повідомленнями, для циклічно-періодичного обміну даними процесу вона неідеальна. Враховуючи, що змінні процесу, які зчитуються циклічно-періодично, не змінюють своє значення з часу попереднього читання, значні ресурси мережі будуть витрачатися марно. Дієвою альтернативою клієнт-серверній моделі ідентифікованого обміну для циклічно-періодичного обміну даними є модель функціонування Видавець-Абонент.

Модель Видавець-Абонент (або *Виробник-Споживач*) для ідентифікованого обміну функціонує так. При конфігуруванні мережної системи, як правило, при ідентифікації даних, для кожного вузла вказується належність ідентифікованих даних для видавництва чи для підписки. Якщо дані вузла призначені для Видавництва, то в момент необхідності їх публікації він відправляє ці дані в широкомовному режимі. Вузли, яким потрібне значення цих даних, конфігуруються на них як Абоненти і в момент їх публікації оновлюють їх значення. Аналогічно функціонують сервіси Виробник-Споживач, за винятком того, що функції виробництва та споживання даних можуть бути підтримані вже на каналному рівні через комунікаційні буфери.

Модель Видавець-Абонент (або *Виробник-Споживач*) для обміну повідомленнями підходить, при необхідності частої відправки великої кількості різних за типом повідомлень, без очікування відповіді. Для прикладу, це може бути діагностична інформація, дані тривоги, попереджувальні повідомлення, які передаються через певний виділений логічний канал. На відміну від Клієнт-Серверної моделі обміну повідомленнями, в даній моделі немає необхідності в ініціації сеансу. Повідомлення яке передається від Видавця (Виробника) до Абонентів (Споживачів) не передбачає обов'язкову відповідь, хоча така можлива. Крім того, повідомлення може надсилатись декільком прикладним Процесам одночасно.

Сьогодні багато промислових мереж дають можливість користуватися як мінімум двома типами сервісів прикладного рівня: клієнт-серверна модель обміну повідомленнями для ациклічного трафіку (обмін параметричними даними) та однією з моделей ідентифікованого обміну для циклічного (обмін даними процесу). Підбиваючи підсумки, для кожного сервісу визначимо найкращі моделі функціонування (табл. 2.3).

Таблиця 2.3

НАЙБІЛЬШ ПІДХОДЯЩІ МОДЕЛІ ДЛЯ РЕАЛІЗАЦІЇ СЕРВІСІВ ОБМІНУ ДАНИМИ НА ПРИКЛАДНОМУ РІВНІ

Сервіси прикладного рівня		Краща модель реалізації
обмін даними процесу	циклічно-періодичний	pull Publish-Subscribe (Producer-Consumer) ідентифікованого обміну
	ациклічний за зміною	push Publish-Subscribe (Producer-Consumer) ідентифікованого обміну
	ациклічний за запитом	Client-server обміну повідомленнями
програмування/конфігурування вузла (обмін параметричними даними)		Client-server обміну повідомленнями
управління станом вузла		Client-server обміну повідомленнями
діагностичні сервіси		для ідентифікації помилки — Publish-Subscribe (Producer-Consumer) обміну повідомленнями
функції резервування		для відправки образу процесу — Publish-Subscribe (Producer-Consumer) ідентифікованого обміну

2.4.2. Забезпечення каналом зв'язку між вузлами та методи доступу

Прикладні сутності (прикладні Процеси) виконуються на конкретних апаратних засобах — вузлах мережі. Для того, щоб їх зв'язати, передусім, необхідно забезпечити зв'язок між вузлами. Створенням каналу зв'язку займаються сутності каналного рівня.

Канальний рівень виконує ряд специфічних функцій: забезпечує службовий інтерфейс для верхнього рівня; обробляє помилки передачі даних у повідомленні; управляє потоком даних, що виключає порушення роботи повільних приймачів швидкими передавачами; в мережах із загальним доступом організовує порядок доступу до загального середовища передачі та забезпечує доставку даних до потрібних вузлів.

Канальний рівень оперує *кадрами (Frame)*, що являють собою послідовність інформаційних та службових байтів, які по чергово відправляються сутністю фізичного рівня.

У ширококомовних мережах каналний рівень має забезпечити доставку кадрів конкретному(-ним) адресатові (-ам). Так як одночасно шину прослуховують усі приймачі, необхідно ідентифікувати того, кому призначений даний кадр.

2.4.2.1 Адресація доставки кадрів. У ширококомовних мережах всі вузли прослуховують шину і відповідно отримують однакові дані. Канальний рівень має відфільтрувати кадри, які не призначені для даного вузла (рис. 2.7).

Доставка кадрів може проводитись одним із двох способів:

- доставка кадрів орієнтована на адресу вузлів (модель Source-Destination-msg);
- доставка кадрів орієнтовна на ідентифікатор повідомлення (модель Producer-Consumer-msg).

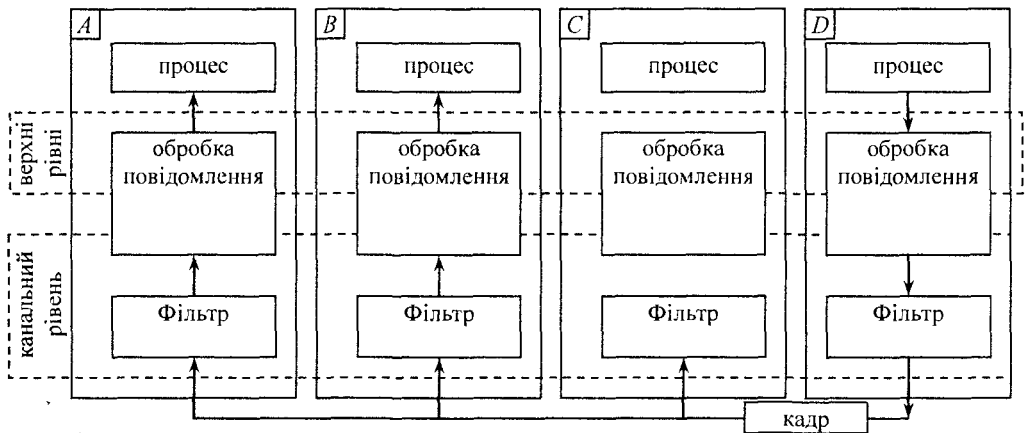


Рис. 2.7. Відфільтрування кадрів на каналному рівні

Доставка кадрів по моделі *Відправник-Адресат-msg (Source-Destination-msg)* орієнтована на адресацію вузлів отримувача (інколи також відправника) і є найбільш популярним способом доставки даних. У таких системах кожен вузол у мережі має свою унікальну адресу. При формуванні кадру, каналний рівень вузла відправника додає до нього заголовок з адресою (-сами) вузлів отримувачів. Усі вузли отримують цей кадр, але тільки вузли з указаною адресою пропустять його для обробки верхнім рівням. Таким чином, фільтр каналного рівня кожного вузла налаштований на свою адресу. Адресу вузла на каналному рівні прийнято називати *MAC-адресою (Media Access Control)*.

У різних протоколах використовуються свої механізми адресації, однак можна дещо їх систематизувати. У мережах з методом доступу Ведучий-Ведений кожний Ведений повинен мати свою адресу, за якою він отримує кадри від Ведучого. Ведучий, у свою чергу, може не мати адреси, якщо всі кадри відправлені Веденими, будуть призначені йому. Для мереж з множинним доступом, відправник у кадрі повинен вказати, окрім адреси отримувача, ще і свою адресу. В іншому випадку не зрозуміти, від кого прийшов кадр.

Більшість мереж дозволяють проводити ширококомовну передачу, тобто передавати дані всім вузлам на шині. Як правило це сервіси, які не передбачають відповіді або підтвердження від вузлів. Для ширококомовної передачі виділяється спеціальна ширококомовна адреса (наприклад 255_{10}), яка резервується для цих цілей і пропускається фільтрами каналного рівня. Крім ширококомовної передачі, деякі мережі підтримують також багатоадресну передачу.

Ряд мереж із загальним доступом використовують інший спосіб фільтрації кадрів — на основі ідентифікаторів повідомлення, який отримав назву *Виробник-Споживач-msg (Producer-Consumer-msg)*. При такому способі адресації доставки кадрів, кожний з них містить ідентифікаційне поле, по якому проводиться фільтрація. Фільтр кожного вузла налаштовується тільки на потрібну множину ідентифікаторів. Інакше кажучи, фільтр буде пропускати через себе тільки ті кадри, ідентифікатори яких відповідають одному із заданих значень.

У системах Виробник-Споживач-msg кожен кадр, що передається мережею, повинен мати унікальний ідентифікатор, на який:

- каналний рівень одного з вузлів буде налаштований як відправник (producer, Виробник);

- каналний рівень вузлів призначення буде налаштований як отримувач (consumer, Споживач), тобто його фільтр буде пропускати цей кадр.

Модель адресації Виробник-Споживач-msg ефективна в мережах з груповою передачею даних, зокрема в системах з ідентифікованим обміном даними Виробник-Споживач. Представниками цієї моделі адресації є мережі CAN та WorldFIP.

Враховуючи однакову назву моделей адресації та функціонування сервісів прикладного рівня, для перших у даному посібнику введений суфікс «msg». Слід зазначити, що наведені моделі не є загальноприйнятими і можуть відрізнятися від наведених в інших джерелах. Дані моделі використовуються тільки для зручності сприйняття матеріалу в даному посібнику.

2.4.2.2. Методи доступу. Для широкомовних мереж (шинна та деревовидна топологія) необхідно визначити порядок доступу до єдиного каналу. Широкомовні мережі, які використовують в якості носія сигналу металевий кабель, мають шинну топологію. Якщо у мережі з шинною топологією не буде чітко визначено, хто в конкретний момент може займати шину для передачі, то може виникнути момент одночасної передачі бітової послідовності двома або кількома передавачами. Така ситуація називається *колізією* (Collision) або конфліктом. У цьому випадку приймачі не зможуть визначити, який біт був переданий, оскільки рівень сигналу буде результатом дій декількох передавачів. Окрім того, у деяких мережах таке загалом неприпустимо, оскільки апаратура передачі пристроїв може вийти з ладу через значні струми на шині. Організація доступу до шини — це одне із завдань каналного рівня.

У промислових мережах найчастіше використовують централізовані методи доступу, тобто коли право видачі дозволу на займання шини має один вузол. У кожний конкретний час саме він визначає, якому вузлу на шині зараз можна передавати дані.

2.4.2.3. Централізований метод доступу Ведучий-Ведений. Найбільш популярні серед мереж з централізованим методом доступу — мережі типу *Ведучий-Ведений (Master-Slave)*, у яких право на управління володінням шиною надається Ведучому, а Ведені займають шину тільки з його дозволу. Останні мають унікальну адресу (адреса Веденого), за допомогою якої Ведучий ідентифікує, кому надсилається повідомлення.

Як правило, у системах Ведучий-Ведені, Ведені є пасивними вузлами, які можуть тільки відповідати на запити Ведучого. Якщо при цьому Процеси взаємодіють за моделлю Клієнт-Сервер, то Процес на вузлі з правами Ведучого завжди буде Клієнтом, а на вузлі Веденого — Сервером, що виключає можливість безпосереднього обміну між Веденими.

Існують рішення, в яких Процес кожного вузла має право на ініціювання обміну. Для клієнт-серверної взаємодії це означає, що клієнтський Процес може розташовуватись на вузлі Веденого. Для реалізації такої схеми Ведучий проводить циклічне опитування (polling) кожного Веденого (рис. 2.8). Якщо клієнтсь-

кий Процес Ведучого хоче звернутися до 1-го Процесу у списку Веденого, система працює як у попередньому випадку. Якщо Процесу Ведучого нічого не потрібно від 1-го Веденого, він надсилає йому запит ідентифікації, на який Ведений відповідає кадром підтвердження (на рис. 2.8 це відповідно запити 1 і 2). Після обслуговування 1-го Веденого, Ведучий опитує 2-го і т.д. Якщо Процесу Веденого необхідно зробити клієнтський запит, замість підтвердження він надсилає кадр з цим запитом (3). У зв'язку з тим, що можливість зайняти шину передавачем Веденого регламентується Ведучим (адже той надав таке право запитом ідентифікації) принцип Ведучий-Ведений залишається. Після отримання кадру запиту Веденого Ведучий йому відповідає звичайним підтвердженням (типу: запит — отримав, коли оброблю — відповім). Така схема дає можливість прикладним Процесам Веденого проявити себе в якості Клієнта, тому умовно будемо називати такі мережі *Ведучий-Ведений з активними Веденими*. Ведучий постійно веде опитування Ведених, дає можливість для виявлення несправних вузлів та включення нових вузлів з правами Веденого у мережу без переконфігурування Ведучого.

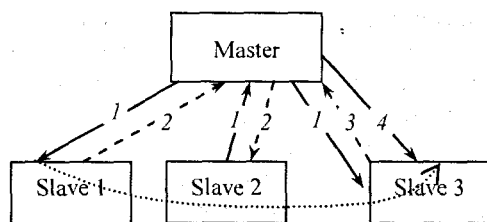


Рис. 2.8. Функціонування мереж з активними Веденими

1 — ідентифікація або клієнтський запит, 2 — підтвердження Веденого, 3 — запит Веденого, 4 — підтвердження Ведучого

Слід зупинитися ще на одній принциповій відмінності між наведеними схемами функціонування Ведучий-Ведений. В обох випадках на каналному рівні використовується сервіс з підтвердженням без установки з'єднання. Тобто після кадру від Ведучого повинен надійти відповідний кадр від Веденого. Щоб уникнути зависання роботи мережі, у випадку виходу з ладу Веденого існують *таймаути* — максимально можливий час очікування кадру від Веденого.

У класичній схемі Ведучий-Ведений після того, як клієнтський прикладний Процес Ведучого відправляє запит Процесу Веденого, Ведучий очікує відповіді на цей запит. Тобто, поки прикладний Процес Веденого не обробить запит від Ведучого, кадр відповіді не надійде, а Ведучий не зможе надіслати кадр іншому Веденому. У системі з активними Веденими, каналний рівень після отримання кадру від Ведучого відразу його підтверджує відповідним кадром. У цей час прикладний Процес Веденого ще не отримує надісланий до нього запит, однак Ведучий уже може сформувати запит до іншого Веденого. Тобто, наприклад, за перший цикл сканування Ведених Процес Ведучого може надіслати запити всім Веденим, а за другий — уже отримати відповіді. Це значить, що продуктивність роботи такої системи з декількома вузлами, як правило, вища, ніж у класичній схемі.

У системах Ведучий-Ведені на прикладному рівні може також функціонувати модель «Видавець-Абонент». У цьому випадку Ведучий повинен регламентувати час відновлення прикладним Процесом-Видавцем своїх змінних.

Серед розглянутих схем функціонування Ведучий-Ведені класична — простіша в реалізації. Однак у системі, що працює за моделлю Клієнт-Сервер, передбачено наявність тільки одного вузла-Клієнта, який є Ведучим. Нерідко у мережах існує необхідність використання декількох Клієнтів (наприклад, два ПК з SCADA або дві панелі оператора і т.д.). Для таких випадків підійде шина з активними Веденими. Недоліком останньої є складність у реалізації та деякі затрати часу на циклічне опитування. Функціонування на прикладному рівні моделі Видавець-Абонент для обміну ідентифікованими даними теж потребує складного протоколу на всіх рівнях моделі OSI. Недоліком усіх моделей є наявність центрального вузла, у випадку збою якого мережа перестає функціонувати. Тому існують мережі з *плаваючим Ведучим*, коли функції Ведучого можуть перейти до іншого вузла у випадку виходу з ладу останнього.

До класичної схеми Ведучий-Ведений можна віднести роботу каналного рівня MODBUS RTU/ASCII, AS-i, HART з активним Веденим — UNI-TELWAY, CC-Link.

2.4.2.4. Централізований метод доступу з Арбітром шини. Цей метод доступу підходить до мереж, де прикладні Процеси вузлів обмінюються на базі pull-моделі Видавець-Абонент (Виробник-Споживач) обміну ідентифікованими даними. Нагадаємо, що в цій моделі ідентифіковані дані передаються Видавцем (Виробником) до Абонентів (Споживачів) в момент надсилання запиту на публікацію (виробництво). На каналному рівні запит на публікацію означає відправку по мережі підготовлених ідентифікованих даних, які містяться у буфері. Для управління доступом до шини всі заплановані передачі відбуваються згідно з розкладом, за яким слідує один вузол, який називають *Арбітром шини*. Арбітр шини має таблицю залежності відновлення мережних змінних та запланованих повідомлень від часу. Для мереж з моделлю адресації Producer-Consumer-msg таблиця містить ідентифікатори змінних/повідомлень, які необхідно відновлювати. Роль Арбітра дуже схожа на роль Ведучого: жоден вузол не почне передавати, поки Арбітр не дасть команду на відновлення мережної змінної згідно з розкладом.

Слід зазначити, що прикладні Процеси на мережі з такою моделлю доступу можуть обмінюватись як за моделлю Видавець-Абонент (Виробник-Споживач), так і за Клієнт-Серверною моделлю обміну повідомленнями. Це може бути реалізовано таким чином. Запит від клієнтського Процесу розміщується у буфері каналного рівня. Як тільки відновлюється якась мережна змінна, для якої Процес даного вузла є Видавцем/Виробником, разом зі змінною він відправляє ідентифікатор повідомлення-запиту, який Арбітр розміщує до списку ациклічних повідомлень. Після відновлення всіх мережних змінних Арбітр відсилає сигнал на дозвіл формування запиту з першим ідентифікатором у списку ациклічних повідомлень, на який зреагує вузол з клієнтським Процесом. Після відправки запиту Арбітр направляє сигнал для відправки наступного ациклічного повідомлення. Повідомлення-відповіді від серверних Процесів відправляються від вузлів аналогічно.

Модель доступу до шини з арбітром функціонує в таких мережах як WorldFip та FF.

2.4.2.5. Децентралізований метод доступу Маркерне кільце. В промислових мережах використовують також відомий у комп'ютерних мережах метод доступу з використанням передачі маркера по кільцю (*Token Ring — маркерне кільце*). У цьому методі кожен вузол займає шину для передачі на час володіння маркером, а після закінчення часу віддає маркер іншому вузлу. Маркерне кільце підходить для мереж з одноранговими вузлами, коли час володіння ресурсами шини (володіння маркером) рівномірно розподіляється між учасниками. Така реалізація дозволяє економити часовий ресурс (якщо вузлу нічого передавати, він може передати маркер далі) та виявити відсутні вузли в мережі. Маркерне кільце використовується в мережах «Транзит», CC-Link IE.

2.4.2.6. Множинний випадковий метод доступу. *Множинний* або *довільний метод доступу* передбачає доступ до шини будь-якого вузла, коли в нього є дані для передачі. При використанні такого способу в один із моментів обов'язково виникне колізія у випадку, коли більше ніж один вузол одночасно захоче передавати дані. Основні відмінності між протоколами довільного доступу визначаються способами вирішення цих колізій. У ряді таких протоколів передавач за необхідності відразу буде намагатися зайняти шину без її прослуховування. Розглянемо це на прикладі.

Приклад 2.5. Основні концепції. Випадковий метод доступу без прослуховування шини.

Завдання. Проілюструвати проблему виникнення колізії в мережі з випадковим методом доступу без попереднього прослуховування шини на прикладі системи з трьома вузлами *A, B, C*.

Рішення. Коли у вузла *A* з'являються дані для передачі вузлу *C*, він займає шину для передачі. Поки передає вузол *A*, у вузла *B* теж з'являються дані для передачі, тому він теж починає їх передавати. Виникає колізія, і вузол *C* не зможе отримати дані ні від вузла *B*, оскільки біти його кадру з самого початку перекриваються з бітами кадру вузла *A*, ні від вузла *A* — через спотворення другої частини кадру. Як бачимо, вузол *B* не тільки не зміг передати свої кадри, а і завдав це зробити вузлу *A*.

Приклад 2.5

Для підвищення ефективності ряд протоколів передбачають прослуховування шини перед тим, як зайняти її для передачі. Такі протоколи називають *протоколами з контролем несучої CSMA* (Carrier-Sense Multiple Access — множинний доступ з контролем несучої). Серед них найбільшу популярність має *CSMA/CD* (CSMA with Collision Detection — множинний доступ з контролем несучої і виявленням конфліктів), який використовується у класичному Ethernet. Розглянемо принцип його функціонування на прикладі.

Приклад 2.6. Основні концепції функціонування CSMA/CD

Завдання. Проілюструвати функціонування протоколу CSMA/CD на прикладі системи з трьома вузлами *A, B, C*.

Рішення. Вузли *A, B* і *C* прослуховують шину. Коли у вузла *A* з'являються дані для передачі і при цьому шина вільна, він займає шину для передачі. Під час передачі кадру вузла *A* у вузлів *B* і *C* з'явилися дані для передачі. Вони виявляють зайнятість шини і чекають її звільнення. Як тільки шина звіль-

льнилася (тобто вузол A повністю передав весь кадр), вузли B і C одночасно намагаються передавати свої кадри, що приводить до колізії. Обидва вузли при передачі своїх бітів прослуховують канал, знову виявляють колізію і припиняють передачу. Кожен включає генератор випадкового часу, через який вони знову зможуть спробувати передати свій кадр. Якщо наприклад у вузла B генератор видав менший час, саме він починає передавати свій кадр.

Приклад 2.6

2.4.2.7. Множинний метод доступу з використанням поля арбітражу. Один із цікавих способів вирішення колізій у мережах типу CSMA є використання у кадрах передачі поля арбітражу. Для реалізації такої схеми необхідно, щоб на шині один логічний сигнал фізично домінував над іншим логічним сигналом. Тобто, скажімо, при одночасній передачі логічної «1» вузлом A і логічного «0» вузлом «В» в результаті на шині була логічна «1». В такому випадку логічну «1» називають **домінантним бітом** (dominant bit), а «0» — **рецесивним** (recessive bit). Усі вузли прослуховують шину, і як тільки вона стає вільною, вони можуть передавати свої кадри. Якщо два вузли одночасно доступилися до каналу, то при передачі бітів наступить момент, коли логічні стани біту одного й іншого вузла не співпадуть і домінантний біт буде визначальним. Вузол, який передавав дані з рецесивним бітом, побачить на шині не свою послідовність і миттєво припинить передачу. Область бітів, в яких виставляється пріоритет називається **полем арбітражу**. Біти у полі арбітражу вузол виставляє, виходячи зі своєї адреси або/і пріоритетності кадру. Розглянемо приклад.

Приклад 2.7. Основні концепції функціонування CSMA/CA

Завдання. Продемонструвати роботу протоколу CSMA/CA з використанням поля арбітражу в системі з двома вузами A і B .

Рішення. Кожний вузол на шині має свою адресу. Нехай до шини одночасно доступилися вузол A , який має адресу $163_{10}(10100011_2)$, і вузол B з адресою $161_{10}(10100001_2)$ (рис. 2.9). Після передачі 7-го біту поля арбітражу вузол B припинить передавати свій кадр, оскільки логічна «1» у кадрі вузла A домінує над логічним «0» кадру вузла B . В такій системі, як бачимо, вузли із старшими адресами будуть мати більший пріоритет.

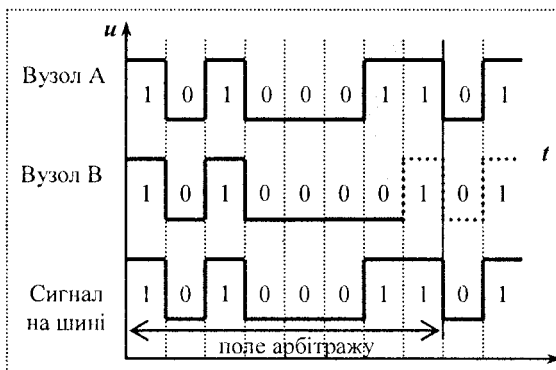


Рис. 2.9. Вирішення колізій за допомогою поля арбітражу

Приклад 2.7

У деяких системах вузли можуть динамічно змінювати поле арбітражу, наприклад, збільшувати пріоритет під час передачі важливих і термінових повідомлень.

Протоколи з використанням поля арбітражу у різних джерелах називають по-різному: CSMA/CA (CSMA Collision Arbitration); CSMA/CD+AMP (CSMA Collision Detection and Arbitration on Message Priority); CSMA/BA (CSMA Bitwise Arbitration); CSMA/MA. Сьогодні в промисловій автоматизації цей протокол використовується в мережах CAN.

2.4.2.8. Передбачуваний псевдопостійний метод доступу CSMA

Цей алгоритм доступу використовується в мережі LonWorks і належить до сімейства CSMA. Він називається псевдопостійним «p-persistent», оскільки час очікування перед відправкою кадру в мережу вибирається випадковим чином. Тобто коли шина вільна, кожен вузол перед відправкою кадру генерує випадковий час. По закінченні цього часу перевіряється вільність шини, і при позитивному результаті відправляється кадр. Такий алгоритм значно зменшує кількість колізій. Цей алгоритм також називається передбачуваним «predective», оскільки алгоритм розраховує трафік каналу. Тобто діапазон, з якого генерується випадкова тривалість паузи перед відправкою пакета, вибирається залежно від кількості вузлів у мережі. Таким чином, у цілому алгоритм доступу до середовища в LonTalk називається *predictive p-persistent CSMA*.

2.4.2.9. Гібридні методи доступу. Гібридні методи доступу використовують комбінацію з методів, розглянутих вище. Найбільш відомим з них є комбінація методу передачі маркеру і Ведучий-Ведений, які використовуються в мережах PROFIBUS (рис. 2.10). Усі вузли діляться на активні учасники — Ведучі і пасивні — Ведені. Між активними учасниками передається маркер (логічне маркерне кільце), на час володіння яким він може спілкуватися з пасивними вузлами.

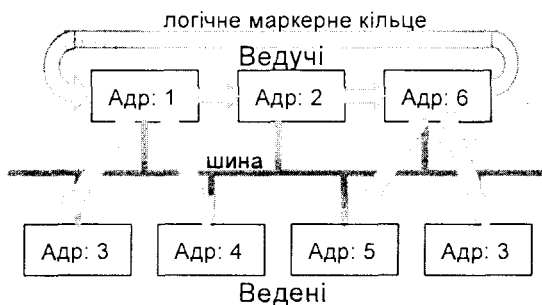


Рис. 2.10. Гібридний метод доступу в Profibus

2.4.2.10. Метод доступу TDMA. В TDMA (Time Division Multiple Access) для доступу до шини кожному вузлу виділяється часовий інтервал (*тайм-слот*), а дані від кожного вузла поєднуються в один *Сумарний Кадр (Summation-Frame)*. Для функціонування такої мережі необхідно чітко визначення часу передачі. Як правило, для цього одному з вузлів надається право Ведучого, який володіє інформацією про кількість вузлів у мережі, та визначає тривалість сумарного кадру як суму всіх тайм-слотів. Можуть бути виділені додаткові тайм-слоти, для пере-

дачі ациклічних повідомлень. Цей метод доступу часто використовують у бездротових мережах та в ряді промислових мереж, наприклад INTERBUS, CIP-CompoNET.

Дещо схожий метод доступу STDMA розглянутий у підрозділі, присвяченому мережі CIP-ControlNET.

Як бачимо, переваги того чи іншого методу доступу до шини відчутні тільки в контексті конкретного застосування та моделі функціонування сервісів на прикладному рівні. Якщо у прикладних сервісів конкретної промислової мережі можливий певний вибір, то метод доступу до шини повністю визначений і незмінний. Методи доступу до шини істотно залежать від реалізації фізичного рівня. Так, разом зі стандартними промисловими інтерфейсами найчастіше використовується метод Ведучий-Ведений. Слід зазначити, що методи доступу до середовища, що використовуються в промислових мережах, не вичерпуються наведеними в даному посібнику.

2.4.3. Формування кадру та контроль за помилками

Для обміну даними між каналними сутностями використовуються кадри (Frame). Кадри вміщують не тільки інформацію про дані процесу, а й додаткову службову інформацію. Канальний рівень користується сервісами фізичного рівня, який приймає потік бітів і передає їх за призначенням, додаючи службові сигнали (наприклад, синхронізуючі). Цей потік не застрахований від помилок. Кількість прийнятих бітів може бути меншою, рівною або більшою кількості переданих бітів. Крім того, значення прийнятих бітів може відрізнитися від переданих. Рівень передачі даних повинен знайти і по можливості виправити ці помилки.

Передача кадрів може вестись як побітно, тобто послідовністю бітів різної довжини, так і посимвольно (побайтно) — послідовністю символів. Символ вміщує не тільки біти даних, а й службові біти (стартовий, стоповий, біти парності тощо). У першому випадку між бітами в кадрі відсутні паузи, в другому — між бітами різних символів йде певна пауза, яка обумовлена стоповими бітами.

Кадр, який був відісланий приймачу, може бути спотворений у результаті дій завод і неправильно прийнятий адресатом. Може виникнути ситуація, що кадр не дійде до отримувача, наприклад, через вихід із ладу останнього. Оскільки однією з вимог промислових мереж є надійність, слід забезпечити контроль за доставкою кадру та його цілісністю. Такий контроль проводиться при використанні надійних сервісів з підтвердженнями.

При отриманні даних отримувач повинен проаналізувати їх правильність і у випадку виявлення помилки (пошкодження даних) може відправити зворотний кадр з інформацією про помилку. У цьому випадку відправник може повторно організувати передачу кадру. Таким чином, при позитивному результаті одержання кадру, отримувач відсилає позитивне підтвердження *ACK* (*ACKnowledgement*), а при отриманні помилкового кадру — негативне підтвердження *NACK* (*Negative Acknowledgement*). Дане завдання вирішується на каналному рівні, а спосіб її вирішення залежить від протоколу. Протокол описує також спосіб визначення помилки в кадрі.

При передачі даних по фізичному каналу завади на лінії можуть спотворити один або, ще гірше, декілька бітів у кадрі. Для боротьби з помилками у мережах існують дві основні стратегії. Кожна з них передбачає додавання надлишкової інформації до даних, що передаються. Тобто до кадру добавляються допоміжні дані, які при його отриманні дозволяють отримувачу визначити помилку. Перша стратегія, яка в основному використовується для дротового зв'язку, базується на тому, що допоміжні дані тільки *виявляють наявність* помилки. При такому підході отримувач за надлишковими даними аналізує правильність отриманих даних, і якщо помилка присутня — робить повторний запит або відправляє негативне підтвердження. Інша стратегія спрямована на передачу надлишкових даних не тільки для ідентифікації помилки, а і для відтворення помилкових даних. Цей підхід використовується у бездротовій передачі.

Надлишкові дані, які добавляються до кадру для контролю за помилками є результатом певної функції над інформаційними даними, які, як правило, називають *контрольною сумою* (Check Sum). Найбільш простий спосіб обрахунку контрольної суми може бути зроблений побітовою логічною операцією з кожним байтом кадру. Результат вставляється в кінець кадру, тобто в кінцевик. Як приклад — використання виключного АБО (XOR) над усіма інформаційними байтами.

Один з поширених алгоритмів розрахунку контрольної суми наведеним вище способом — це алгоритм *LRC* (Longitudinal Redundancy Check). Усі байти повідомлення арифметично сумуються між собою, а результат записується в байт. Під час певного кроку сума може вийти за межі одного байта, але старші біти в сумі не враховуються (відкидаються) і результат завжди буде одно-байтним. В кінці розрахунку міняється знак результату (помножується на «-1»).

Метод *CRC* (Cyclic Redundancy Check) більш складний у використанні, але у випадках, коли можливе спотворення декількох бітів інформації, він більш надійний. Контрольна сума CRC використовується у багатьох мережних протоколах, наприклад у MODBUS RTU (CRC16).

2.4.4. Топологія мережі та сегментація

Канальний рівень оперує кадрами як готовими байтовими структурами. Передачею кадрів по фізичному середовищу займається фізичний рівень. У системах що використовують в якості середовища передачі металевий провід, важливою характеристикою є топологія мережі.

Мережна топологія визначає фізичну структуру мережі. Вона накладає обмеження на розташування ліній зв'язку та впливає на функціонування сервісів канального рівня. Базовими топологіями є: кільце, зірка й шина (рис. 2.11).

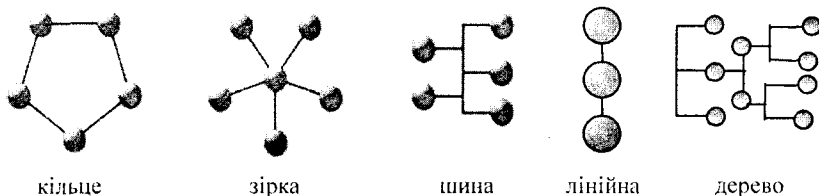


Рис. 2.11. Базові топології мереж

2.4.4.1. Топологія «кільце» (ring). У цій топології вузли попарно з'єднані між собою таким чином, що створюється кільце. Тобто кожен вузол має два канали для зв'язку з двома сусідніми вузлами. Зв'язок з іншими вузлами мережі може проходити тільки через ці два вузли. Враховуючи можливість створення дуплексного каналу між вузлами, можна побудувати високошвидкісну систему, в якій всі вузли рівноцінно будуть доступати до мережних змінних. Разом з кільцевою фізичною топологією на каналному рівні, як правило, використовується логічне маркерне кільце.

Відмова в роботі хоча б одного вузла призводить до порушення роботи кільця, а, отже, і до зупинки всіх передач. Щоб цього уникнути, необхідно включати в мережу автоматичні перемикачі, які замикають мережне кільце в тому місці, де пристрій вийшов з режиму нормальної роботи. Тобто вони дозволяють включати/виключати окремі вузли без переривання нормальної роботи.

Серед промислових мереж, які базуються на цій топології, можна виділити мережі «Транзит» (ПЛК — РЕМІКОНТ Р-130) та INTERBUS. Крім того, промислові та комп'ютерні мережі, що використовують оптоволоконний кабель теж часто базуються на цій топології.

2.4.4.2. Топологія «зірка» (star). У даній топології вся інформація передається через виділений центральний вузол. Кожна периферійна станція підключається до цього вузла окремим фізичним каналом (променем) і може обмінюватись з іншою станцією тільки через центральний вузол. По суті зірка — це з'єднання типу точка-точка багатоканального вузла з іншими вузлами.

На каналному рівні немає необхідності в організації доступу до середовища передачі, оскільки воно використовується для дуплексного обміну тільки між двома вузлами. Центральний вузол повинен бути винятково надійним пристроєм. Крім того, для підключення нових вузлів необхідні канали на центральному вузлі, а також окремі кабельні лінії. Топологія «зірка» дозволяє істотно оптимізувати трафік, передаючи пакети тільки в ті промені, де перебувають їх одержувачі.

Класичний Ethernet на коаксіальному кабелі базується на топології «шина», однак Ethernet на базі витої пари має топологію «зірка». Слід зазначити, що для Ethernet на базі концентраторів фізично всі канали зв'язані між собою через один концентратор, який виконує функції багатоканального репітера і поєднує між собою канали загальною шиною. Внаслідок шинного зв'язку необхідно організувати на каналному рівні порядок доступу до цієї шини. Для комутованого дуплексного Ethernet комутатор забезпечує також маршрутизацію кадрів по каналах, тому метод CSMA/CD для цього рішення неактуальний.

2.4.4.3. Топологія «шина» (bus). Найпоширеніший тип мережної топології. Всі пристрої підключаються до загальної лінії зв'язку, тим самим використовуючи одне й те саме середовище як для передачі, так і для прийому. Основна її перевага полягає в простоті, дешевизні й легкості переконфігурування. Добре підходить для територіально розподілених систем. Має ряд недоліків: присутність у кожному вузлі мережі загального трафіка, небезпека втрати зв'язку при одному обриві каналу зв'язку.

У будь-якій шинній структурі всі пристрої приєднані до загального середовища передачі даних або шини. Таким чином, усі вузли прослуховують шину, що,

з одного боку, дає змогу відправляти широкомовні повідомлення, а з іншого — вимагає визначення методу доступу до середовища задля передачі.

На відміну від топології «кільце», адресат одержує свій інформаційний пакет без посередників. Процес підключення додаткових вузлів до шини не вимагає додаткових каналів з боку вже працюючих вузлів мережі, як у топології «зірка».

Слід відмітити, що в шинній топології не дозволяються великі відгалуження від магістрального кабелю. Максимальна їх довжина регламентується правилами побудови мережі.

2.4.4.4. Лінійна топологія (linear). У лінійній топології вузли з'єднуються один з одним, створюючи лінію.

2.4.4.5. Топологія «дерево» (tree). Це гібридна топологія, яка базується на поєднанні шинної та зіркоподібної топології. Зовні це ієрархічна структура, в якій на верхньому рівні елементи поєднуються у шину, від вузлів якої можуть йти зіркоподібні відгалуження. В таких системах, як правило, використовується середовище із загальним доступом (аналогічно шині).

2.4.4.5. Сегментація

Крім обмеження на територіальне розміщення кабелів зв'язку, топологія також впливає на максимальну швидкість передачі даних, правила термінування, можливість і правила фізичної сегментації. Під *фізичним сегментом* мережі розуміють фізичну частину мережі, відокремлену спеціальним мережним пристроєм:

- на фізичному рівні моделі OSI — *репітером* (повторювачем) або *концентратором* (hub);
- на каналному рівні моделі OSI — *комутатором* (switch) або *мостом* (bridge);
- на мережному рівні моделі OSI — *маршрутизатором* (gateway).

Репітери забезпечують підсилення сигналу, що дозволяє на фізичному рівні збільшити кількість вузлів у мережі, максимальну довжину ліній зв'язку, однак обмежує швидкісні характеристики передачі. Концентратори використовуються для об'єднання декількох пристроїв у мережу, а також можуть виконувати функції репітерів, за рахунок чого територіальна «зірка» реалізовується в фізичну «шину». З точки зору засобів каналного рівня мережа поєднана репітером або концентратором, є одним цілим.

Міст (bridge) забезпечує пересилку кадрів між сегментами за їх MAC-адресами. Тобто міст забезпечує фільтрацію кадрів, що циркулюють між сегментами за їх MAC-адресами, що значно зменшує завантаження цих сегментів. Комутатор — це багатопортовий міст, що забезпечує фільтрацію кадрів між сегментами, підключеними до його портів.

Маршрутизатор забезпечує пересилку даних між сегментами мережі, які архітектурно можуть відрізнятися між собою. Більш детально принципи функціонування маршрутизаторів описані в 2.4.6.

2.4.5. Фізичні інтерфейси передачі даних

Більш детально про фізичні інтерфейси передачі даних та методи передачі бітів Ви можете дізнатися з розділу 3 даного посібника, тут коротко розглянемо основні характеристики. На фізичному рівні для передачі даних може використо-

уватись електричний сигнал (електричні кабелі), оптичний (оптоволокну) або радіохвилі різної частоти (включаючи інфрачервоний спектр). Передачу бітів електричним сигналом можна безпосередньо або модулюючи несучий сигнал. Серед безпосередніх способів найбільш відомі методи NRZ, манчестерське кодування та диференційне манчестерське кодування. При модулюванні сигналу використовують амплітудну, фазову, частотну модуляцію або їх поєднання. Передачу бітів можна реалізувати електричним сигналом за струмом або за напругою. При використанні напруги можуть застосовувати асиметричний або симетричний (диференційний) метод передачі.

Для підключення технічних засобів до мережі та реалізації кабельної інфраструктури використовують різні типи з'єднувачів (конекторів). Для промислових умов експлуатації їх виготовляють герметичними, наприклад, M12.

На фізичному рівні більшість промислових мереж використовують один із стандартних послідовних інтерфейсів: RS-232, RS-422, RS-485 або CurrentLoop («струмова петля»). Використання їх пов'язано з відносною дешевизною організації зв'язку (існують готові мікросхеми з їх реалізацією) та їх популярністю.

2.4.5.1. Інтерфейс RS-232. Цей стандартний дуплексний інтерфейс спочатку був розроблений для передачі даних між ПК (через COM-порт) та модемом. Сьогодні частіше застосовується для зв'язку засобів без використання модему — «нуль-модемне з'єднання», для якого вистачає 3-х проводів (TxD — для передачі, RxD — для прийому, SG — сигнальна земля). Механічно цей стандарт визначає 9- і 25-контактні роз'єми. Бітова швидкість вибирається з діапазону від 50 до 115200 біт/с, максимальна довжина ліній зв'язку — 15 м. Кількість пристроїв у мережі — 2. Особливих вимог до кабелю зв'язку немає.

2.4.5.2. Інтерфейс RS-422. Симетричний дуплексний інтерфейс RS-422 використовує диференціальні сигнальні лінії (перепад напруги на сигнальних лініях). На прийомному кінці використовують дві інформаційні лінії й лінія заземлення. Реалізований принцип передачі робить цей стандарт стійким до зовнішніх завад. Рекомендоване середовище передачі — подвійна вита пара з окремою сигнальною землею (5 — провідне з'єднання). Цей стандарт дозволяє використовувати фізичні лінії передачі даних до 1200 м і швидкість до 10 Мбіт/с. Однак він не розрахований на використання в шинних топологіях. Тим не менше, до одного передавача можна підключити до 10 приймачів: 1 вузол в дуплексному режимі, інші тільки в симплексному — на прослуховування. В кінцях лінії з боку приймача рекомендується ставити термінатори лінії, наприклад резистори, з номіналом, рівним характеристичному опору кабелю.

2.4.5.3. Інтерфейс RS-485. Цей тип інтерфейсу відповідає специфікації симетричної передачі даних, описаної в американському стандарті EIA/TIA-485. Сумісний зі стандартом RS-422, однак дає можливість будувати шинні топології (до 32 вузлів) з напівдуплексним способом обміну, за рахунок використання додаткового інформаційного управляючого сигналу активації/деактивації трансмітера. В якості середовища передачі рекомендується використовувати виту пару. Необхідне термінування ліній при великих бітових швидкостях.

Для порівняння інтерфейсів, розглянутих вище, найбільш важливі характеристики, зведені в таблицю 2.4.

Таблиця 2.4

ПОРІВНЯЛЬНІ ХАРАКТЕРИСТИКИ СТАНДАРТНИХ ПОСЛІДОВНИХ ІНТЕРФЕЙСІВ

Назва, стандарт	EIA RS-232C	EIA RS-422A	EIA/TIA-485, RS-485	20 мА «струмова петля», CL, ІРПС.
Тип сигналу	напруга	напруга, диференційний	напруга, диференційний	струм
режим з'єднання	дуплекс	дуплекс	напівдуплекс, дуплекс	дуплекс
максим. кількість передавачів/приймачів	1/1	1/10	32/32 (при опорі приймачів 12кОм)	1/1, у багатоточкових схемах, залежно від реалізації
максим. відстань	15 м	1200 м	1200 м	декілька км
максим. бітова швидкість	115200 біт/с	10 Мбіт/с	10 Мбіт/с	порядку 1 Мбіт

В інтерфейсі «струмова петля» кількість трансиверів залежить від конкретної реалізації.

Для всіх інтерфейсів (окрім RS-232C) максимальна бітова швидкість залежить від багатьох факторів. Перш за все — це довжина лінії зв'язку, також тип кабелю, рівень завад, способи узгодження ліній і т.д. За необхідності підключення засобів з різними промисловими інтерфейсами необхідно використати адаптери — перетворювачі, наприклад RS-232C->RS-485 або USB->RS-485.

2.4.6. Маршрутизація

Більшість промислових мереж функціонують і описані в контексті фізичного, каналного та прикладного рівнів. Тим не менше, ряд промислових мереж підтримують можливість сегментації на мережному рівні.

Основною функцією мережного рівня є забезпечення передачі кадру з одного фізичного сегмента в інший, навіть якщо ці сегменти відрізняються своєю архітектурою. Вузли, які займаються цією передачею, називають маршрутизаторами. Об'єднання сегментів у цьому випадку будемо називати *інтермережею*, а самі сегменти — *підмережею*. Маршрутизатори одночасно підключаються до декількох мереж, тобто мають вихід на декілька фізичних ліній зв'язку. Вони повинні отримати пакет з однієї мережі, за необхідності розбити на частини і відправити в іншу мережу по іншій лінії відповідно з *маршрутною таблицею*. В маршрутній таблиці міститься інформація про те, яким чином можна потрапити в необхідну мережу (мережі). Слід наголосити, що це дуже спрощена схема функціонування, яка не враховує багатьох особливостей.

Розглянемо функціонування інтермережі на декількох прикладах. На рис. 2.12 зображені 6 мереж, об'єднаних в одну інтермережу. В середині ширококомовних мереж (з #1 по #5) кожен вузол має свою адресу, передбачену протоколом каналного рівня для цієї мережі. Цією адресою користуються інші вузли цієї ж ме-

режі для обміну даними. Але для вузлів інших мереж ця адреса не має ніякого значення. Для адресації вузлів в інтермережі використовується своя система, визначена на мережному рівні. Надалі адресу, визначену на мережному рівні, будемо називати *мережною адресою*, хоч правильніше б було назвати її міжмережною адресою. На рисунку мережі з #1 по #6 можуть бути різні за природою, але такі, що використовують спільний протокол на мережному рівні і єдину міжмережну систему адресації. Таким чином, кожний вузол, крім MAC-адреси на каналному рівні, має ще унікальну в середині інтермережі мережну адресу (на рисунку показані через крапку). Слід нагадати, що мережний рівень оперує пакетами, в яких і розміщується адреса отримувача і відправника.

Приклад 2.8. Основні концепції. Маршрутизація 1.

Завдання. Пояснити, яким чином відправляється запит від Процесу вузла з мережною адресою 1.1 на вузол 1.3 (рис. 2.12), якщо система адресації побудована так, що один байт виділяється під номер мережі, а один байт під номер станції в мережі, тобто використовується двохбайтова система адресації. Це значить, що, якщо вузол має позначення 1.2, то він являє собою станцію 2 в мережі 1 (у першому байті адреси буде значення 1, а в другому — 2).

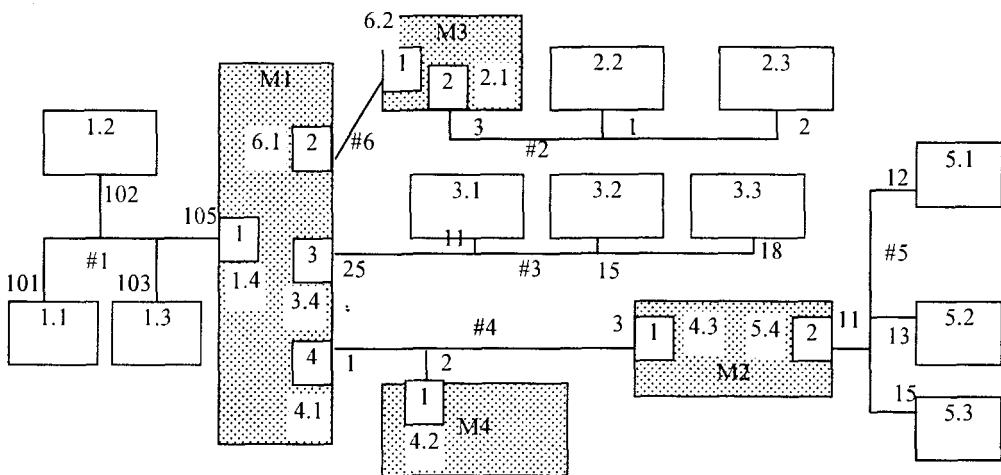


Рис. 2.12. Приклад функціонування інтермережі

Рішення. Як бачимо, ці вузли містяться в одній мережі. Але сутності, вищі за мережний рівень (наприклад, транспортна сутність), у даному випадку не вміють працювати з каналним рівнем, щоб напяму відправити кадр на MAC-адресу 103. Їм байдуже, чи міститься вузол з адресою 1.3 в локальній мережі, чи в іншій. Верхній рівень просто передає дані для відправки і значення адреси отримувача. Розглянемо, як може формуватися кадр на вузлі відправника з адресою 1.1 (рис. 2.13).

Сутність, яка вища мережного рівня (наприклад, транспортна сутність) сформувала дані для передачі на вузол 1.3. Через інтерфейс мережного рівня вона передала «що» відправляти і «куди» відправляти. Сутність мережного рі-

вня «бачить», що номер мережі отримувача співпадає з власним номером мережі (свою адресу вона знає). Отже необхідно знайти MAC-адресу отримувача і відправити пакет безпосередньо йому. В нашій мережі мережна сутність кожного вузла має таблицю відповідності, в якій вона знаходить відповідну MAC-адресу до заданої мережної. Разом з MAC-адресою отримувача вона передає сформований пакет каналному рівню. Той у свою чергу вставляє в заголовок свою адресу та адресу отримувача, а в кінцевик — контрольну суму.

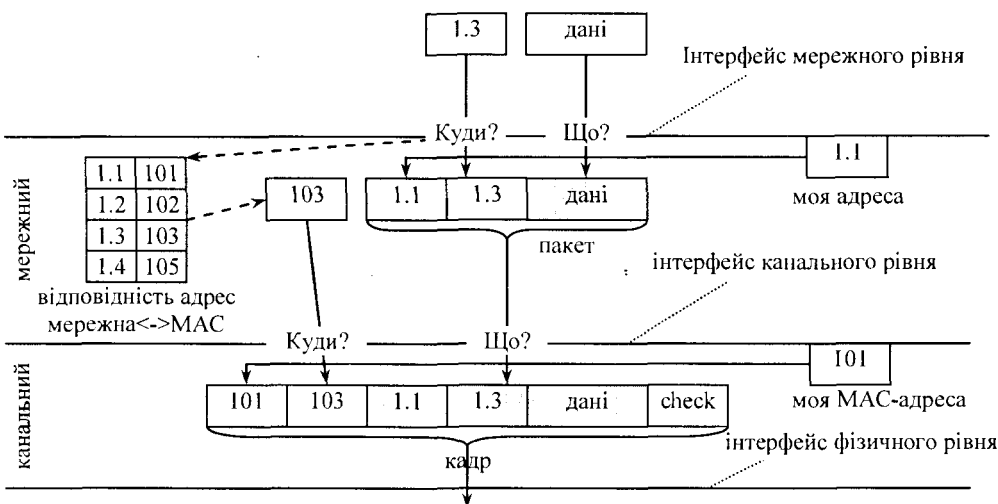


Рис. 2.13. Приклад формування пакету та кадру

Приклад 2.8

Приклад 2.9. Основні концепції. Маршрутизація 2

Завдання. Пояснити, яким чином відправляється запит від Процесу вузла з мережною адресою 1.1 на вузол 2.3 (рис. 2.12).

Рішення. В цьому випадку мережна сутність «бачить», що адреса отримувача не в зоні її досяжності. За такої ситуації вона відправляє кадр маршрутизатору М1, який має MAC-адресу 105. Кадр, який буде направлений маршрутизатору, матиме вигляд такий, як показано на рис. 2.14. Як бачимо, адреса отримувача в пакеті залишилась 2.3, хоча маршрутизатор має мережну адресу 1.4. Все ж, кадр відправляється на MAC-адресу маршрутизатора. Коли маршрутизатор отримав цей кадр, його мережна сутність «побачить», що пакет у кадрі не призначений безпосередньо для неї, отже його треба відправити далі. Оскільки маршрутизатор підключений до декількох мереж, то він перевіряє, чи належить адреса отримувача в кадрі одній із цих мереж. М1 є вузлом у мережах #1, #3, #4 і #6, а мережа отримувача — #2. Якби вони співпали, то ситуація б пішла далі як у попередньому прикладі. Але у цьому випадку маршрутизатор шукає в своїй маршрутній таблиці по мережі #2 наступний маршрутизатор, через який можна вийти на дану мережу. Запис буде знайдено з мережною адресою 6.2 (тобто М3). У своїй таблиці відповідності за

цією адресою M1 знаходить лінію зв'язку, якою буде лінія 2. Далі він вилучає повністю пакет, інкапсулює його в кадр і відправляє по мережі #6. Мережа #6 має інший формат кадру і не потребує адресації на каналному рівні.

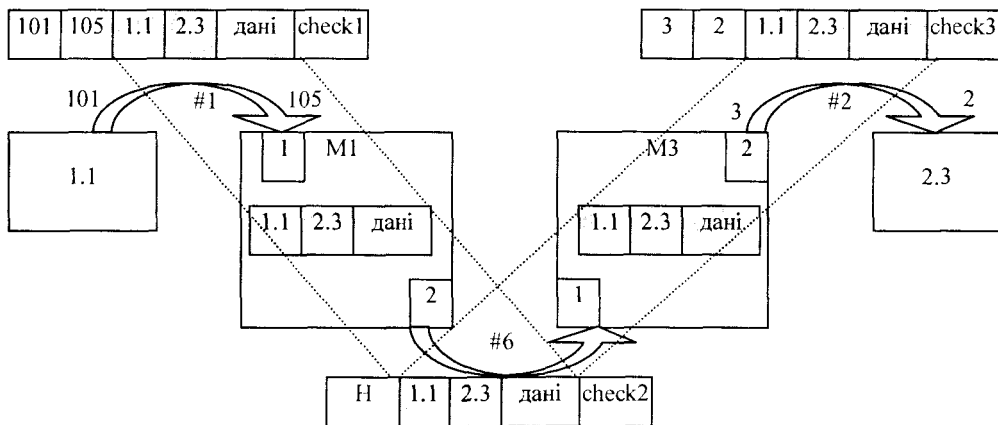


Рис. 2.14. Транспортування пакета за допомогою маршрутизаторів

Коли маршрутизатор M3 отримує кадр, він виявляє, що підключений до тієї мережі, куди повинен потрапити пакет (тобто #2). Знаходить у таблиці відповідності необхідну лінію зв'язку (2) та за мережною адресою необхідну MAC-адресу (теж 2) вилучає пакет та інкапсулює пакет у кадр з цією адресою.

Приклад 2.9

Приклад 2.10. Основні концепції. Маршрутизація 3

Завдання. Пояснити, яким чином відправляється запит від Процесу вузла з мережною адресою 1.1 вузла з адресою 5.2 (рис. 2.12).

Рішення. Все сказане в прикладі 2.9 підходить і для цього випадку, окрім одного кроку. Оскільки мережа #6 являє собою на фізичному рівні з'єднання точка-точка, адресація на каналному рівні тут не потрібна, тому в попередньому прикладі достатньо було знайти лінію зв'язку (канал). Але в нашому випадку на каналі 4 два маршрутизатори. Тому, окрім номера лінії, треба ще вказати MAC-адресу маршрутизатора.

Приклад 2.10

У розглянутих прикладах показаний один із принципів маршрутизації, який навмисно, для його простоти, не враховує багато нюансів. На практиці використовуються дещо схожі алгоритми, наприклад, у протоколі *IP*, на якому базується Internet. У промислових мережах можуть використовуватись зовсім інші підходи, але мета залишається та сама — прозорий обмін пакетами між однорідними або різнорідними мережами.

Разом з маршрутизацією пакетів мережний рівень підтримує ще ряд сервісів:

- маршрутизація пакетів;
- побудова маршрутних таблиць;
- роздача мережних адрес.

2.4.7. Транспортування даних

При доставці пакетів на вузол необхідно, щоб ці дані потрапили до конкретної прикладної сутності (Процесу, прикладної програми). Коли на вузлі виконується тільки одна прикладна програма (наприклад, у контролері), дані передаються їй або системі. Однак сучасні ПТЗ можуть підтримувати одночасне виконання декількох Процесів. Тому необхідно реалізувати доступ до конкретного Процесу. Для цього виділимо такі типи сервісів:

- доступ до операційної системи;
- доступ до прикладного сервісу;
- доступ до програми користувача;
- доступ до окремої програмної складової.

Розглянемо приклад. Припустимо в мережі є два вузли, які представлені двома комп'ютерами (рис. 2.15). Клієнтський Процес *App1* хоче передати певні дані Процесу *App3*, а *App2* — *App4*. Якщо дані будуть вставлятися безпосередньо в пакети/кадри і передаватись з вузла А на вузол Б, то мережна/канальна сутність не зможе визначити, якому з Процесів призначені дані в пакеті/кадрі. Адже на каналному та мережному рівнях вирішується питання доставки даних *вузлу*, а не конкретній прикладній сутності. Це завдання покладається на сутність транспортного рівня. Як і на попередніх двох рівнях вона вирішується за допомогою адресації. В даному випадку адресуються не вузли і не повідомлення, а прикладні сутності, тобто Процеси-отримувачі та Процеси-відправники. Ці адреси (точки доступу TPDU) прийнято називати *TSAP* (Transport Service Access Point — точки доступу транспортного сервісу), а в ряді протоколів *портами* (*port*). Надалі в даному посібнику буде вживатися термін «порт».

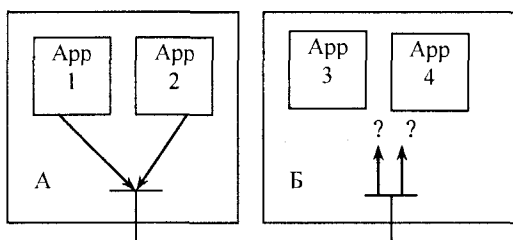


Рис. 2.15. Вибір маршруту для даних

Кожній прикладній сутності призначається свій порт, за допомогою якого до неї можна звернутися (серверний порт). У нашому прикладі *App3* і *App4* можна б було призначити серверні порти, відповідно 3 і 4. Таблиця відповідності портів прикладним сутностям повинна зберігатися як на вузлі А, так і на вузлі Б. Клієнтам теж повинні бути призначені порти. У протоколах UDP і TCP клієнтські порти призначаються динамічно, оскільки сервер буде відповідати тому клієнту, який до нього звернувся.

Транспортні протоколи, як і мережні, можуть надавати декілька типів сервісів: з установкою або без установки з'єднання, з підтвердженням або без підтве-

рджень. Ці типи сервісів будуть прокоментовані при розгляді протоколів TCP та UDP (див. розділ 10).

2.4.8. Підсумкова таблиця з основними робочими характеристиками

Основні характеристики промислових мереж з можливими варіантами їх реалізації в контексті моделі OSI зведемо в одну таблицю. Позначимо їх абревіатурами, які надалі будемо використовувати в посібнику.

Таблиця 2.5

МОЖЛИВІ ВАРІАНТИ РЕАЛІЗАЦІЇ ОСНОВНИХ ХАРАКТЕРИСТИК ПРОМИСЛОВИХ МЕРЕЖ

OSI	характеристика	варіанти реалізації	коментар
	область призначення <i>NetArea</i>	<ul style="list-style-type: none"> – рівень датчиків – рівень контролерів – рівні датчиків/контролерів 	деякі мережі, призначені для кількох рівнів: рівень датчиків включає польовий рівень (field level)
прикладний	наявні сервіси <i>AppService</i>	<ul style="list-style-type: none"> – циклічний обмін даними процесу; – періодичний обмін даними процесу; – ациклічний, за зміною значення обміну даними процесу; – ациклічний, за запитом обміну даними процесу; – обмін параметричними даними (програмування та конфігурування вузла); – управління станом вузлів; – діагностичні сервіси; – функції резервування; 	визначається набором протоколів прикладного та нижніх рівнів; вузол мережі не завжди підтримує всі сервіси;
	модель обміну <i>AppModel</i>	<ul style="list-style-type: none"> – клієнт-серверна модель обміну повідомленнями; – клієнт-серверна модель ідентифікованого обміну (Polling); – Видавець-Абонент/Виробник-Споживач ідентифікованого обміну; – Видавець-Абонент/Виробник-Споживач обміну повідомленнями; 	протокол може підтримувати кілька сервісів з різними варіантами моделей;
	наявність профілів вузлів. <i>AppProfile</i>	<ul style="list-style-type: none"> – профілі не визначені; – перелік підтримуваних профілями типів вузлів; 	вузол мережі, як правило, підтримує тільки один профіль пристрою;
	кількісна характеристика відновлення даних процесу <i>AppProcData</i>	використ. сервіс: максимальна кількість змінних для циклічно-періодичного обміну даними процесу	наводяться заявлені дані в офіційних джерелах; залежить від вузлів мережі; для мереж з відсутнім циклічним трафіком наводиться обмеження ациклічного
	швидкісна характеристика відновлення даних процесу <i>AppResolut</i>	кількість інформації/час оновлення	наводяться заявлені дані в офіційних джерелах

OSI	характеристика	варіанти реалізації	коментар
транспортний	наявні сервіси <i>TpService</i>	<ul style="list-style-type: none"> - доступ до операційної системи; - доступ до прикладного сервісу; - доступ до програми користувача; - доступ до окремої програмної складової; 	визначається набором протоколів транспортного рівня; вузол мережі не завжди підтримує всі сервіси;
	модель обміну <i>TpModel</i>	<ul style="list-style-type: none"> - надійні: з установлення з'єднання, з підтвердженнями (Acknowledge); - ненадійні: без установлення з'єднання, без підтвердження; 	можливі комбінації
мережний	наявні сервіси <i>NtService</i>	<ul style="list-style-type: none"> - маршрутизація пакетів; - побудова маршрутних таблиць; - роздача мережних адрес; 	визначається набором протоколів мережного рівня; вузол мережі не завжди підтримує всі сервіси;
канальний	метод адресації <i>ChAddModel</i>	<ul style="list-style-type: none"> - Відправник-Адресат-msg; - Виробник-Споживач-msg; 	визначається протоколом канального рівня; може включати обидві моделі
	метод доступу <i>ChAccess</i>	<ul style="list-style-type: none"> - Ведучий-Ведений; - Ведучий-Ведений з активними Веденими; - з Арбітром шини; - CSMA/CD, CSMA/CA, predective p-persistent CSMA; - TDMA, CTDMA; - маркерний; - гібридний 	визначається протоколом канального рівня;
	контроль за помилками <i>ChChecksum</i>	<ul style="list-style-type: none"> - LRC; - CRC (різні варіанти); - інший; 	визначається протоколом канального рівня;
	можливість сегментації <i>ChSegment</i>	<ul style="list-style-type: none"> - так: кількість сегментів, спосіб сегментації, обмеження; - ні; 	використовуються мости та комутатори;
фізичний	фізичний інтерфейс <i>PhInterface</i>	<ul style="list-style-type: none"> - стандартні: RS-232, RS-422, RS-485, CL, USB, Bluetooth, IR; - інші: передача напруженого/струмом, метод кодування/модуляції, симетричний/асиметричний; 	для багатьох мереж можливий вибір;
	фізичне середовище передачі <i>PhMedia</i>	<ul style="list-style-type: none"> - звичайний або спец. кабель (кількість жил); - вита пара (кількість витих пар); - екранована вита пара (кількість витих пар); - коаксіальний кабель; - оптоволокно (одно-, багатомодовий); - радіохвилі (тип); - лінійна 	для багатьох мереж можливий вибір;
	топологія <i>PhTopology</i>	<ul style="list-style-type: none"> - лінійна; - шина; - зірка; - кільце; - дерево; - вільна; 	для багатьох мереж можливий вибір; комбінація можлива при сегментації;

Закінчення табл. 2.5

OSI	характеристика	варіанти реалізації	коментар
фізичний	максимальна довжина відгалужень <i>PhLdrop</i>	– не допускається; – довжина в метрах, вплив на загальну довжину лінії, обмеження;	визначається тільки для шини та дерева; залежить від бітової швидкості;
	бітова швидкість передачі <i>PhBaudRate</i>	– вибирається з ряду можливих (вказується мінімакс); – фіксована;	залежить від вибраного інтерфейсу, середовища передачі, довжини та кількості сегментів; обмеження вказуються
	можливість сегментації <i>PhSegment</i>	– так (кількість сегментів); – ні;	залежить від інтерфейсу та середовища; використовуються репітери та концентратори;
	максимальна кількість вузлів на сегмент <i>PhNodes</i>	максимальна кількість вузлів у сегменті, максимальна кількість всього в мережі	залежить від інтерфейсу та середовища;
	максимальна довжина сегмента <i>PhLength</i>	максимальна довжина в метрах сегмента та всієї мережі; правила термінування;	
	живлення по мережі <i>PhSupply</i>	– не дозволяється; – по інформаційним проводам, макс. струм; – по окремим проводам кабелю, макс. струм;	вказується напруга та максимальний струм; залежить від вузла мережі.

2.5. Короткий огляд промислових мереж

У даному огляді наведені характеристики, властивості й галузі застосування ряду відкритих промислових мереж. Мережі AS-і, MODBUS, CAN, CANOpen та PROFIBUS детальніше розглянуті у розділах 5-9. Протоколи мереж на базі Ethernet, а саме Ethernet/IP, P-NET/IP, TC-NET, EtherCAT, EthernetPowerlink, EPA, SEROCS III, Vnet/IP, PROFINET CBA, PROFINET I/O коротко розглянуті в розділі 10.

2.5.1. Мережі MODBUS RTU/ASCII та MODBUS TCP/IP

Більш детально про мережі MODBUS Ви можете почитати в розділі 6.

2.5.1.1. Походження MODBUS. Мережі MODBUS RTU/ASCII розроблені фірмою Gould Inc. (Modicon) для побудови промислових розподілених систем управління. MODBUS приваблює простотою протоколу й використанням стандартних промислових інтерфейсів. Популярність MODBUS RTU та широке впровадження Ethernet і технологій Intranet у промислових мережах призвело до появи мережі MODBUS TCP/IP. Сьогодні просуванням усіх мереж MODBUS займається некомерційна організація *MODBUS-IDA*.

2.5.1.2. Фізичний рівень MODBUS RTU/ASCII. Спеціальний фізичний інтерфейс для *MODBUS RTU/ASCII* не визначений, а вибирається з ряду стандар-

тних: RS-232C, RS-422, RS-485, USB або струмова петля. Однак стандартами MODBUS-IDA визначені правила для використання RS-485. Бітова швидкість вибирається однією з двох 19200 біт/с або 9600 біт/с. Для кодування переданих даних на каналному рівні використовують формати ASCII із 7-бітним символом та RTU з 8-бітним символом.

2.5.1.3. Канальний рівень MODBUS RTU/ASCII. У MODBUS RTU/ASCII використовується класична схема доступу Ведучий-Ведений. Конфігурація на основі цього протоколу припускає наявність одного Ведучого вузла й до 247 Ведених. Тільки Ведучий ініціює цикли обміну даними, тобто тільки його прикладний Процес може бути клієнтським. Кожне повідомлення-запит на каналному рівні Ведучого додатково обрамляється адресою вузла Веденого (адресу 0 використовують для широкомовної передачі) та контрольною сумою (CRC16 для RTU та LRC для ASCII). Запит відправляється Веденому, який його обробляє і повертає повідомлення-відповідь, тим самим підтверджуючи запит.

2.5.1.4. Прикладний рівень MODBUS RTU/ASCII. Протокол прикладного рівня MODBUS (МВАР-протокол) базується на клієнт-серверній моделі обміну повідомленнями. Повідомлення-запит складається з коду команди (читання, запис і т.д.) та додаткових даних. Протокол МВАР забезпечує ациклічний доступ за запитом до змінних іншого вузла (Input Registers, Holding Registers, Discrete Inputs, Discrete Coils), діагностичними сервісами, сервісами запису/зчитування програми. Циклічно-періодичні операції протоколом не визначені і можуть бути реалізовані тільки в прикладній програмі.

2.5.1.5. Походження та структура MODBUS TCP/IP. *MODBUS TCP/IP* (інша назва MODBUS/TCP) на прикладному рівні базується на системі MODBUS повідомлень (МВАР протокол), на транспортному та мережному використовує стек протоколів TCP/IP, а на каналному та фізичному — мережу Ethernet. Таке поєднання дає можливість простої і практично безперешкодної інформаційної інтеграції між вузлами різних рівнів управління. На відміну від MODBUS RTU та ASCII прикладний Процес одночасно може виступати як в якості Клієнта, так і Серверу. Це значить, що кількість MODBUS-Клієнтів та MODBUS-Серверів в єдиній мережі практично не обмежується. Протокол MODBUS TCP/IP внесений до стандартів MEK разом з протоколом реального часу RTPS (див. розділ 10).

Характеристики MODBUS RTU/ASCII та MODBUS TCP/IP зведені в таблицю 2.6.

Таблиця 2.6

ХАРАКТЕРИСТИКИ MODBUS RTU/ASCII ТА TCP/IP

	характеристика	MODBUS RTU/ASCII	MODBUS TCP/IP
	<i>NetArea</i>	– рівні датчиків/контролерів	– рівні датчиків/контролерів
прикладний	<i>AppService</i>	Протокол МВАР (MODBUS Application Protocol)	
		ациклічне за запитом читання/запис даних процесу (16 бітні слова, біти) та параметричних даних; управління станом вузлів; діагностичні сервіси; функції програмування вузла;	
			доступ до змінних вузлів-Ведених MODBUS RTU (шлюз)

Закінчення табл. 2.6

	характеристика	MODBUS RTU/ASCII	MODBUS TCP/IP
	<i>AppModel</i>	клієнт-серверна модель обміну повідомленнями для даних процесу та параметричних даних; Клієнт тільки на вузлі Ведучого	клієнт-серверна модель обміну повідомленнями для даних процесу та параметричних даних;
	<i>AppProcData</i>	циклічний трафік не визначений, для ациклічного — теоретично 65536 об'єктів з кожної області пам'яті на вузол	
	<i>AppResolut</i>	залежить від швидкості та реалізації на вузлах (див. розділ 6).	
транспортний	<i>TpService</i>	не визначений	визначені протоколом TCP
мережний	<i>NtService</i>		визначені протоколом IP
каналний	<i>ChAddModel</i>	Відправник-Адресат-msg, адреси тільки у Ведених 1-247, 0 — ширококомвна;	Відповідно до протоколу EthernetII/IEEE 802.2/3
	<i>ChAccess</i>	Ведучий-Ведений;	
	<i>ChChecksum</i>	LRC для ASCII та CRC16 для RTU	
фізичний	<i>PhInterface</i> *	RS-232, RS-422, RS-485, CL, USB	Відповідно до протоколу EthernetII/IEEE 802.3
	<i>PhMedia</i> *	подвійна екранована вита пара (для RS-485)	
	<i>PhTopology</i> *	шина (для RS-485)	
	<i>PhLdrop</i> *	<=20 м, мультипортові <= 40 м / кількість портів (для RS-485)	
	<i>PhBaudRate</i> *	9600 біт/с або 19200 біт/с (для RS-485)	
	<i>PhSegment</i>	кількість сегментів залежить від реалізації інтерфейсу	
	<i>PhNodes</i> *	32 (для RS-485)	
	<i>PhLength</i> *	<1200 м (для RS-485)	

* — згідно з MODBUS-IDA

2.5.2. Мережа WorldFIP

2.5.2.1. Походження. Протокол **FIP** (The Factory Information Protocol) є результатом колективних зусиль ряду європейських компаній (в основному, Франції, Бельгії й Італії) як альтернативне рішення до запропонованих американським ринком промислових мереж. Цей протокол націлений на високі швидкості передачі у чітко визначені інтервали відновлення даних. У 1986 році стандарт під-

тримується організацією із більш, ніж 100 членів, яка мала назву Club FIP. У 1993 році була створена організація *WorldFIP*, яка об'єднує великих світових виробників контролерів та мережного обладнання. Одноименний протокол внесений до стандартів МЕК.

2.5.2.2. Фізичний рівень. На фізичному рівні для з'єднання використовується вита пара або оптоволокно. Бітова швидкість вибирається з ряду: 31.25 кБіт/сек — мінімальна; 1 Мбіт/сек — стандартна; 2.5 Мбіт/сек — максимальна; 5 Мбіт/сек — при оптоволоконному з'єднанні. Топологія мережі — шина. Використовується синхронний інтерфейс з диференційною передачею за напругою та манчестерським кодуванням.

2.5.2.3. Канальний рівень

Як і в більшості мереж, для обміну даними в мережі WorldFIP, з точки зору користувача використовуються два типи сервісів:

- періодичний та аперіодичний обмін даними процесу;
- аперіодичний обмін параметричними даними.

Істотна частка функціонування цих сервісів покладена на канальний рівень: прикладний рівень працює через мережний буфер, оновленням якого займається сутність каналного рівня. Тому наведені вище сервіси на каналному рівні підтримуються такими сервісами:

- періодичний обмін ідентифікованими змінними (для періодичного обміну даними процесу);
- обмін ідентифікованими змінними за запитом (для аперіодичного обміну даними процесу за запитом);
- обмін повідомленнями за запитом (для обміну параметричними даними).

На каналному рівні WorldFIP послуговується гібридним методом адресації: для обміну ідентифікованими змінними використовується модель Виробник-Споживач-msg, а для обміну повідомленнями — Відправник-Адресат-msg.

Для доступу до шини використовується централізований метод з Арбітром, який слідує за оновленням мережних періодичних та аперіодичних змінних, а також за обміном повідомленнями. Сукупність періодичних змінних у буферах вузлів формують розподілену базу даних. Арбітр шини вміщує таблицю часу відновлення періодичних змінних у вигляді табл. 2.7.

Таблиця 2.7

ПРИКЛАД ТАБЛИЦІ КОНФІГУРАЦІЇ МЕРЕЖНИХ ЗМІННИХ

Змінна	Періодичність відновлення (мс)	Тип	Тривалість відновлення (мкс)
A	5	INT_8	170
B	10	INT_16	178
C	15	OSTR_32	418
D	20	SFPOINT	194
E	20	UNS_32	194
F	30	VSTR_16	290

На кожну змінну в цій таблиці підписуються Виробники (C — Consumer), які відновлюватимуть цю змінну, та Споживачі (P — Producer), які будуть зчитувати її значення. Коли надходить час на відновлення даної змінної Арбітр відправляє широкомовний запит на відновлення цієї змінної: «ID_DAT_A». Після цього каналний рівень Виробника (на рис. 2.16 позначений як P) видасть з буфера значення змінної A в широкомовному режимі «RP_DAT_A». Всі вузли, які підписані на цю змінну як Споживачі (на рис. 2.16 позначені як C), відновлять її у своїх буферах обміну.

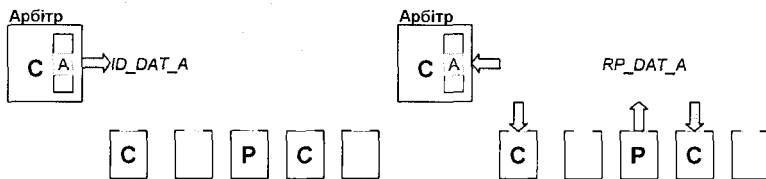


Рис. 2.16. Відновлення змінної A

Мережний буфер каналного рівня кожного вузла складається з двох частин (рис. 2.17):

- змінні, які виробляються: при запиті арбітра на одну з цих змін, станція передає її значення в мережу в широкомовному режимі;

- змінні, які споживаються: відновлюються в буфері, коли виробники цих змінних передають їх значення.

З буфера в прикладну програму змінні зчитуються незалежно від роботи мережі.

Для різного часу відновлення періодичних змінних робота мережі ділиться на **Макроцикли**, які в свою чергу діляться на **Елементарні цикли**. Кожний елементарний цикл триває певний період часу. За цей період повинні відновитися всі періодичні змінні, в яких наступив період відновлення. За весь Макроцикл, будь-яка змінна повинна бути відновлена як мінімум 1 раз (найповільніша).

Наприклад, для змінних з табл. 2.7. Макроцикл буде мати вигляд як на рис. 2.18, тривати 60 мс і складатися з 12 Елементарних циклів. Змінна A буде відновлюватися з кожним Елементарним циклом (кожні 5 мс) а F — з кожним шостим Елементарним циклом.

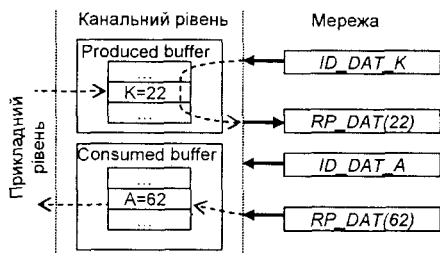


Рис. 2.17. Реалізація мережного буферу

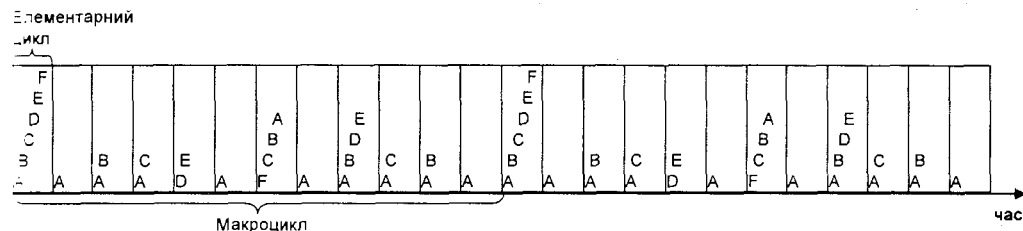


Рис. 2.18. Організація Макроциклу

Час, який залишається до іншого елементарного циклу використовується для аперіодичних повідомлень (рис. 2.19). Відновлення аперіодичних змінних та передача повідомлень забезпечуються таким чином. Коли прикладний рівень замовляє даний сервіс у каналного рівня, при відновленні будь-якої періодичної змінної даним вузлом (як Виробником), в кадрі буде передана мітка про замовлення аперіодичного трафіка, яку Арбітр помістить у чергу замовлень. При закінченні оновлення всіх змінних в Елементарному циклі, Арбітр з черги вибирає замовлення і надає право конкретному вузлу на формування запиту. Нагадаємо, що для обміну повідомленнями використовується модель адресації Відправник-Адресат-msg.

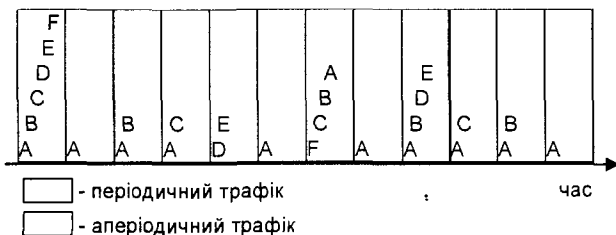


Рис. 2.19. Виділення аперіодичного трафіку

2.5.2.4. Прикладний рівень

Сервіси прикладного рівня діляться на три групи:

- ABAS (прикладні сервіси Арбітра шини);
- MPS (періодичні/аперіодичні сервіси виробництва);
- subMMS (піднабір сервісів повідомлень).

Сервіси ABAS забезпечують зв'язок з відповідними сервісами каналного рівня. MPS забезпечує обмін значеннями періодичних змінних з буфером та активації запиту для оновлення аперіодичної ідентифікованої змінної каналним рівнем за необхідності. Сервіси subMMS забезпечують передачу та активацію аперіодичних повідомлень іншому прикладному Процесу в мережі за адресою його вузла.

Основні характеристики WorldFIP зведені в табл. 2.8.

Таблиця 2.8

ХАРАКТЕРИСТИКИ WORLDVIP

OSI	Характеристика	WorldFIP
	<i>NetArea</i>	рівні датчиків/контролерів
прикладний	<i>AppService</i>	MPS — періодичний та аперіодичний (за запитом) обмін даними процесу через мережні змінні (через буфери пристроїв); subMMS — аперіодичний обмін параметричними даними; управління станом вузлів; діагностичні сервіси; ABAS — управління та резервування функцій Арбітру, функції резервування ліній зв'язку;
	<i>AppModel</i>	– pull-модель ідентифікованого обміну Виробник-Споживач для оновлення мережних змінних (обмін даними процесу); – клієнт-серверна модель обміну повідомленнями для передачі аперіодичних повідомлень (обмін параметричними даними);

Закінчення табл. 2.8

OSI	Характеристика	WorldFIP
прикладний	<i>AppProfile</i>	різні типи пристроїв (Device WorldFIP — DWF), визначені в AFNOR та CENELEC різні профілі мереж (особливості фізичного рівня)
	<i>AppProcData</i>	сервіс MPS — циклічних змінних: залежить від реалізації
	<i>AppResolut</i>	для профілю FIPIO: 10 мс / 100 word***; для FIPWAY 50 мс/128 word****
каналний	<i>ChAddMode</i>	– Виробник-Споживач-msg для мережних змінних – Відправник-Адресат-msg для аперіодичних повідомлень
	<i>ChAccess</i>	централізований з Арбітром шини
	<i>ChChecksum</i>	контрольна сума FCS
фізичний	<i>PhInterface</i>	синхронний інтерфейс з манчестерським кодуванням за напругою (IEC 61158-2)
	<i>PhMedia</i>	екранована вита пара хвил. опір 150 Ом; оптоволокно;
	<i>PhTopology</i>	шина;
	<i>PhLdrop</i>	враховується в $PhLength = 3 * PhLdrop + Ltrunk$; <i>Ltrunk</i> — довжина магістрального кабелю
	<i>PhBaudRate</i>	– 31.25 кБіт/сек; 1 Мбіт/сек (стандартна); 2.5 Мбіт/сек; – 5 Мбіт/сек (оптоволокно);
	<i>PhSegment</i>	максимум 5 сегментів з'єднані репітерами
	<i>PhNodes</i>	32 (96 на 3-х сегментах)
	<i>PhLength</i>	залежить від профілю мережі; <10000м (при 31.25 кБіт/с); загальна — до 50 км; термінатори 150 Ом на кінцях лінії;
	<i>PhSupply</i>	допускається < 35 В по окремій парі проводів

* — умовна модель, у WorldFIP реалізовано через буфер каналного рівня на базі моделі Producer-Consumer-msg

— умовна модель, у документації WorldFIP називається передачею повідомлень

*** — дані орієнтовні, зроблені на базі розрахункових формул Шнейдер Електрик для 12 8-каналних аналогових вхідних модулів Momentum; FIPIO враховує виділення невеликого трафіка на аперіодичні повідомлення

**** — дані взяті з документів Шнейдер Електрик для розподіленої таблиці 128 слів на 32 пристрої; FIPWAY враховує виділення значного трафіка на аперіодичні повідомлення

2.5.3. Мережа Foundation Fieldbus (FF)

2.5.3.1. Походження. В другій половині 90-х років ряд компаній об'єднали свої зусилля для розробки єдиного стандарту промислової мережі, яка б вирішувала більшість типових завдань, покладених на цифрові комунікації в галузі промислової автоматики. Так була створена організація *Fieldbus Foundation*, яка в рамках американської організації зі стандартизації *ISA* розробила стандарт промислової мережі під назвою *Foundation Fieldbus (FF)*. Організацію Fieldbus Foundation створено на базі двох груп InterOperable Systems Project (*ISP*) та WorldFIP North America, тому їх стандарт увібрав в себе елементи мереж WorldFIP та PROFIBUS.

Технологія FF базується на 3-х рівнях моделі OSI: фізичний, каналний та прикладний. Сьогодні *FF* представляється двома типами мереж: *H1* — мережа рівня датчиків на базі синхронного інтерфейсу зі швидкістю 31,25 КБіт/с та *HSE* (High Speed Ethernet) — промислова мережа контролерного рівня на базі Ethernet. Мережі FF входять до стандарту MEK.

2.5.3.2. Фізичний рівень FF H1.

На фізичному рівні FF H1 використовується синхронна передача даних на базі манчестерського коду. Для синхронізації використовується преамбула. Передавачі вузлів подають сигнал ± 10 мА з швидкістю передачі 31,25 КБіт/с на еквівалентне навантаження 50 Ом для створення амплітуди сигналу 1 Вольт, яке модульоване у верхній частині постійної напруги. Постійна напруга може приймати значення в діапазоні від 9 до 32 вольт. Однак, для іскробезпечних застосувань допустима напруга живлення залежить від бар'єра іскрозахисту. Вузли на H1 можуть живитись від загальної шини. Крім того, H1 при цьому може забезпечувати іскробезпечність, якщо в небезпечній зоні встановлюються бар'єри іскрозахисту.

Топологія FF H1 може бути шинною або деревовидною. Сумарна довжина ліній зв'язку визначається швидкістю обміну, типом кабелю, перерізом проводу, енергоспоживанням пристроїв та необхідністю іскрозахисту. В кінцях лінії встановлюються термінатори з комбінації резистора та конденсатора.

2.5.3.3. Канальний рівень FF H1.

Аналогічно більшості промислових мереж FF H1 забезпечує два типи обміну даними — періодичний обмін даними процесу та аперіодичний обмін параметричними даними. На підрівні доступу до середовища періодичне оновлення даних процесу забезпечується за рахунок організації доступу до шини вузлів згідно з часовим графіком, аперіодичний обмін повідомленнями — за рахунок періодичного полігону вузлів. Для цього у FF H1 виділяється вузол з правами Активного Планувальника Зв'язків (*LAS*), який забезпечує:

- доступ вузлів до фізичного середовища на базі запланованих повідомлень (*scheduled communication*);
- циркуляцію маркера за необхідності обміну незапланованими повідомленнями (*unscheduled communication*).

Пристрої, які підтримують тільки базові функції FF H1 (*Basic Device*) не можуть бути Активними Планувальниками Зв'язків. Ці функції доступні тільки для пристроїв типу *Link Master*.

LAS вміщує розклад передач вихідних буферів для всіх вузлів у мережі, місткість яких повинна передаватись періодично. Коли для пристрою настає час передачі вихідного буфера, LAS відсилає в цей пристрій повідомлення, яке називається Примусовою Розсилкою Даних (Compel Data — *CD*). Після отримання цього повідомлення пристрій публікує дані з буфера на шину в ширококомовному режимі. Вузли з підпискою на ці дані відновлюють їх у своєму буфері.

Для виявлення дефектного вузла, або добавлення нового LAS вміщує список активних вузлів на шині і періодично перевіряє їх наявність. Для цього LAS у час відсутності запланованих повідомлень, відправляє на кожний вузол повідомлен-

ня-маркер (Pass Token — *PT*). Якщо вузол відповідає, він залишається в списку активних вузлів (*Live List*). У момент отримання *PT* вузол може скористатися передачею «незапланованого» повідомлення або відповіді на повідомлення від іншого вузла. Для запобігання зависання маркера на вузлі, для нього визначається тайм-аут. *LAS* також бере на себе функції публікування часу для синхронізації годинників вузлів. *FF H1* дозволяє дублювати функцію *LAS* на іншому вузлі, у випадку виходу з ладу основного.

Як видно, функції вузла з правами *LAS* аналогічні функціям Арбітра шини у *WorldFIP*, особливо це характерно для періодичного оновлення даних. Однак, для аперіодичного обміну повідомленнями на відміну від *WorldFIP* у *FF H1* використовується механізм поліну, який називається *Live List*.

Усі вузи в мережі *FF H1* повинні мати унікальну мережну адресу та *тег пристрою*. Адреси та теги кожному вузлузначаються автоматично конфігураційною утилітою. При підключенні нового пристрою він отримує тимчасову адресу (248-251), після ідентифікації йому призначається нова адреса (16-247). Адреси з діапазону 252-255 використовуються для вузлів-програматорів. Крім того, кожному пристрою в момент виготовлення надається унікальний 32-байтний ідентифікатор, який є його апаратною адресою, аналогічною адресації *MAC*, і складається з коду виробника (6 байт), коду типу пристрою (4 байти), серійного номера (22 байти). Цей ідентифікатор є тегом пристрою за замовчання.

2.5.3.4. Прикладний рівень *FF H1*

Прикладний рівень *FF H1* складається з двох підрівнів *FAS* (Fieldbus Access Sublayer) та *FMS* (Fieldbus Message Specification). Сервіси *FAS* (нижній підрівень) забезпечують функціонування сервісів *FMS*. *FAS* визначені зв'язками *VCR* (Virtual Communication Relationship), які визначають тип обміну між прикладними Процесами, адресу даних джерела та даних адресата, спосіб та час їх відновлення. *VCR* налаштовуються в момент конфігурування системи і можуть бути одним із 3-х типів:

1. Клієнт-Сервер (Client-Server), які передаються від одного прикладного Процесу до іншого, як незаплановані повідомлення в черзі їх знаходження у буфері; коли вузол отримує маркер *PT*, він може відправити незаплановане повідомлення-запит або повідомлення-відповідь, яке адресується прикладному Процесу іншого вузла.

2. Розсилка Звітів (Report Distribution), які ініціюються пристроями для передачі ширококомовних повідомлень (наприклад аларми) за групою адресою; ці *VCR* відновлюються як незаплановані повідомлення.

3. Видавець-Абонент (Publisher-Subscriber), для оновлення буферизованих даних на пристроях; обмін цими *VCR* проводиться після отримання повідомлення *CD*, яке може розсилатися вузлом *LAS* або відправитися Абонентами як незаплановане повідомлення.

Сервіси *FMS* дозволяють програмам користувача обмінюватись між собою повідомленнями по промисловим мережам. *FMS* описує комунікаційні сервіси, формати повідомлень та протокол, необхідні для створення повідомлень для прикладного рівня. Дані, обмін якими проводиться через промислову ме-

режу описуються за допомогою *Описувача Об'єктів* (Object Description), які об'єднані в структуру, що називається *Словник Об'єктів* (Object Dictionary). Кожен Описувач Об'єкта ідентифікується унікальним індексом у Словнику. 0-й індекс («Заголовок Словника») вказує на номер першого індексу Описувачів Об'єкта Прикладного Рівня (див. рівень користувача FF H1), який може починатися з 256-го.

Для віддаленого доступу до даних конкретного пристрою, що описуються Описувачем Об'єктів, використовуються Віртуальні Польові Пристрої *VFD* (Virtual Field Device). Кожний фізичний пристрій включає як мінімум два VFD: Network and System Management VFD (дані *NMIB* та *SMIB*) та User Application VFD (дані *FBAP*). Комунікаційні дані NMIB включають VCR, розклад LAS. Системні дані NMIB включають інформацію про тег-пристрій, його адресу, розклад виконання функціональних блоків (див. рівень користувача). FBAP надає доступ до даних функціональних блоків.

Таким чином, маніпуляція даними кожного вузла проходить з використанням сервісів FMS, які підтримуються зв'язками VCR. Ряд зв'язків VCR типу Клієнт-Сервер зарезервовано для доступу з боку конфігуратора. Використовуючи різні сервіси FMS для цих VCR, конфігуратор вступає до даних NMIB, SMIB та FBAP, тим самим конфігуруючи функціонування кожного вузла, включаючи додавання нових VCR. До базових функцій сервісів FMS належать: управління VCR; доступ до словника об'єктів; доступ до змінних; обробка подій; вивантаження/завантаження даних та програм; дистанційне управління роботою програми.

Опис FF H1 не закінчується на сервісах прикладного рівня. Його родзинкою є опис рівня користувача, який уніфікує і значно спрощує інтеграцію різних пристроїв на H1.

2.5.3.5. Рівень користувача FF H1 та FF HSE. В мережній моделі FF на відміну від OSI, над прикладним рівнем існує прикладний рівень користувача (*User Application*), який дозволяє уніфікувати процес інтеграції засобів від різних виробників, використовуючи однаковий інтерфейс для користувача. Даний інтерфейс є графічним і являє собою організацію процесу управління всією системою з використанням так званих блоків. Пристрої конфігуруються за допомогою блоків ресурсів та блоків перетворювача, а стратегія управління формується за допомогою функціональних блоків. Дані про ці блоки розміщуються у FBAP, які разом з описувачами розміщуються в User Application VFD.

Блок ресурсу (Resource Block) вміщує характеристики польового пристрою, такі, як ім'я пристрою, ім'я виробника та серійний номер. У кожному пристрої є тільки один блок ресурсу. *Функціональні блоки* (Function Block) забезпечують поведінку системи управління. Параметри входів та виходів функціональних блоків можуть бути зв'язані у промисловій мережі за допомогою VCR типу Видавець-Абонент. Для кожного функціонального блоку запланований точний час його виконання. В одній прикладній програмі користувача може бути декілька функціональних блоків. Організацією Fieldbus Foundation визначений набір стандартних функціональних блоків. Перелік цих функціональних блоків наведений у табл. 2.9.

Таблиця 2.9

ПЕРЕЛІК ФУНКЦІОНАЛЬНИХ БЛОКІВ

Символ	Ім'я	Призначення	Символ	Ім'я	Призначення
AI	Analog Input	Аналоговий ввід	LL	Lead Lag	Випередження/запізнювання
AO	Analog Output	Аналоговий вивід	DT	Deadtime	Час нечутливості
BG	Bias/Gain	Зміщення/Підсилення	IT	Integrator (Totalizer)	Інтегратор (тоталізатор)
CS	Control Selector	Селектор входів управління	SPG	Setpoint Ramp Generator	Генератор уставок
DI	Discrete Input	Дискретний вхід	IS	Input Selector	Селектор входів
DO	Discrete Output	Дискретних вихід	AR	Arithmetic	Арифметичний
ML	Manual Loader	Операторський ввід	TMR	Timer	Таймер
PD	Proportional/Derivative	ПД-регулятор	AAL	Analog Alarm	Аналоговий сигнал тривоги
PID	Proportional/Integral/Derivative	ПІД-регулятор	MAI	Multiple Analog Input	Багатоканальний аналоговий вхід
RA	Ratio	Відношення	MAO	Multiple Analog Output	Багатоканальний аналоговий вихід
DC	Device Control	Управління пристроєм	MDI	Multiple Discrete Input	Багатоканальний дискретний вхід
OS	Output Splitter	Розгалужувач виходів	MDO	Multiple Discrete Output	Багатоканальний дискретний вихід
SC	Signal Characterizer	Характеризація сигналу	FFB	Flexible Function Block	Гнучкий функціональний блок для написання прикладної програми

Блоки перетворювача. Блоки перетворювача (Transducer Blocks) перетворювачів вміщують таку інформацію, як дата калібровки, тип та параметри датчика. Тобто саме через ці блоки проводиться параметризація, калібровка та діагностика пристрою.

Крім описаних блоків, на рівні користувача визначені такі об'єкти:

- зв'язок (Link), які визначають зв'язки між входами та виходами функціональних блоків всередині пристроїв та через промислову мережу;
- тренд (Trend), які дозволяють локально накопичувати значення параметрів функціональних блоків для передачі їх на верхній рівень управління чи іншим пристроям;
- тривога (Alert), що дозволяють виводити повідомлення про тривоги та події на промисловій мережі;
- контейнер множини змінних (Multi-Variable container, MVC), які дозволяють накопичувати множину параметрів функціональних блоків для оптимізації зв'язку, виконання транзакцій «Видавець-Абонент» та «Розсилання

звітів»; він включає сконфігурований користувачем список для визначення необхідних параметрів, значення яких наведені в списку змінних;

- вид (View), які являють собою попередньо визначені групи наборів параметрів блоків, що можуть використовуватися людино-машинним інтерфейсом; специфікація визначає чотири види кожного типу блока, а саме: оперативні динамічні дані (VIEW 1), оперативні статичні дані (VIEW 2), всі динамічні дані (VIEW 3), інші статичні дані (VIEW 4).

Процес конфігурування системи на рівні користувача зводиться до створення стратегії управління за допомогою мови програмування графічної блок-схеми FOUNDATION. Тобто це визначення функціональних блоків та зв'язків між ними, аналогічно мові FBD з МЕК 61131-3. За допомогою спеціального програмного забезпечення — конфігуратора FF ці функціональні блоки проектуються на конкретні пристрої (рис. 2.20). Так, наприклад, простий датчик температури з підтримкою FF може вмішувати функціональний блок аналогового вводу (AI), а позиціонер регулюючого клапану може вмішувати функціональний блок ПД-регулятора (PID) та блок аналогового виводу (АО). Таким чином, функціонально закінчений контур регулювання може бути побудований з використанням одного датчика та одного клапана. Слід зазначити, що при проектуванні системи необхідно врахувати наявність конкретного функціонального блока в польових пристроях FF. Доступ до ресурсів конкретного вузла проводиться через Словник Об'єктів User Application VFD.

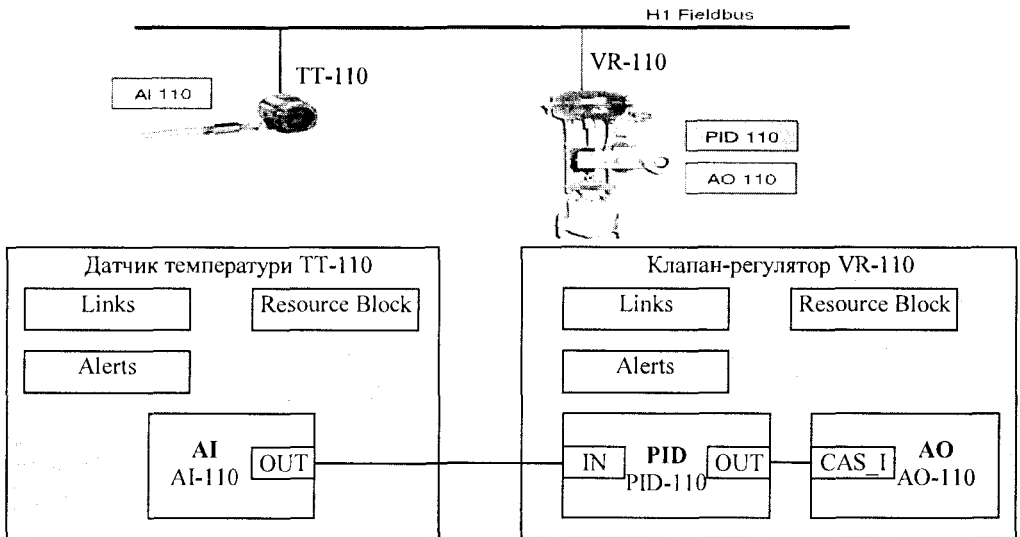


Рис. 2.20. Приклад програми на графічній блок-схемі FOUNDATION

Кожний блок у створеній програмі повинен виконуватися в чітко заданий момент часу. Для цього пристрої в мережі починають обробляти кожний функціональний блок в чітко визначені моменти зсуву відносно часу початку розкладу, який для всіх вузлів однаковий. Для циклічності функціонування програм усі

прикладні Процеси функціонують у контексті так званих **Макроциклів**, які починаються з початком розкладу. В межах Макроциклу необхідно вчасно відновлювати значення для входів функціональних блоків. Якщо входи пов'язані з виходами функціональних блоків інших вузлів, ці значення повинні бути відновлені мережними засобами. Тому всі мережні інформаційні зв'язки функціональних блоків організовані через VCR типу Видавець-Абонент, які відновлюються за розкладом у межах того ж Макроциклу. За обробкою функціональних блоків слідує пристрій, а за відновленням мережних даних, тобто за відновлення VCR слідує LAS.

На рис. 2.21 показано функціонування мережі з рис. 2.20 на базі Макроциклів. У таблиці під рисунком вказані зміщення часу запуску виконання функціональних блоків (DL) відносно абсолютного часу початку розкладу. Управління в датчику ініціює виконання функціонального блока AI-110 з зсувом за часом $DL = 0$. Зі зсувом $DL = 20$ LAS відправляє повідомлення про примусову розсилку даних (CD) блоку AI-110, які публікуються на шину. Ці дані будуть відновлені у входному буфері підписаного на них PID-110. Зі зсувом $DL = 30$ управління системою в клапані ініціює виконання функціонального блока PID-110, після якого зі зсувом $DL = 50$ виконується функціональний блок АО-110. Потім ця послідовність повторюється, забезпечуючи тим самим постійну динаміку контуру регулювання. Для роботи системи за загальним запланованим часом у FF періодично проводиться синхронізація за часом, яка відбувається з використанням публікування часу одним вузлом.

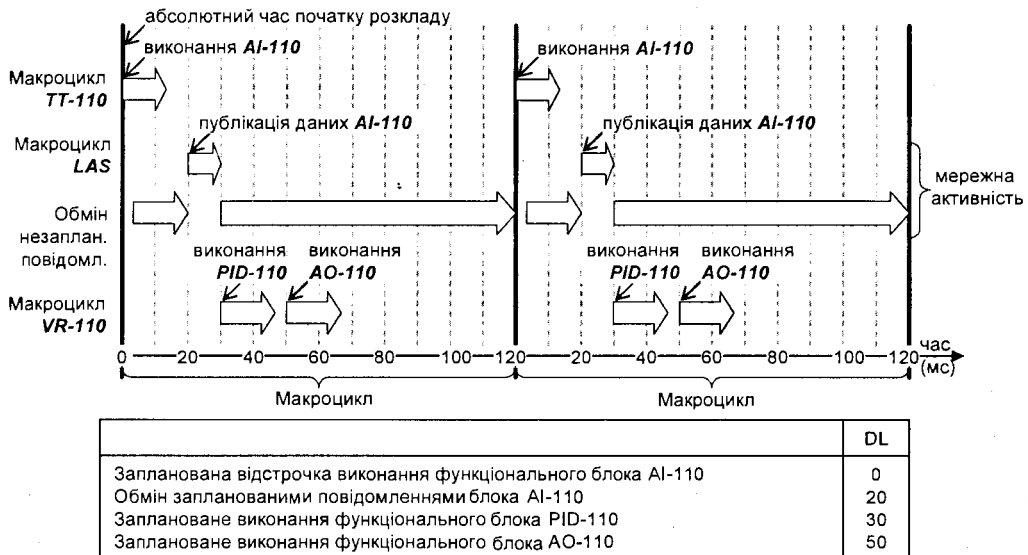


Рис. 2.21. Функціонування FF на базі Макроциклів

Протягом усього часу, окрім інтервалу між зсувом 20 та 30, по мережі може проводитися обмін незапланованими повідомленнями. Слід зазначити, що наяв-

ність вузла з функціями LAS характерна для FF H1, для FF HSE передача по мережі проходить відразу за необхідності, а не за запланованою основою, однак виконання функціональних блоків проходить таким же чином.

2.5.3.6. Функціонування FF HSE

FF HSE (High Speed Ethernet) являє собою промислову мережу, що базується на стеку протоколів Ethernet TCP/IP, яка використовується для високошвидкісних систем управління або для поєднання декількох сегментів FF H1. У HSE використовується такий самий рівень користувача, як і в H1. На прикладному рівні використовується протокол *FDA* (Field Device Access).

У структурі HSE можуть функціонувати чотири категорії пристроїв:

- пристрої зв'язку (LD — Link Device), які служать для з'єднання мереж FF HSE з FF H1; вони передбачають наявність шлюзових служб, які передають повідомлення FDA (SM та FMS) в їх прототипи H1;
- пристрої Ethernet (ED — Ethernet Device), які можуть бути функціональними блоками та мати декілька типових пристроїв вводу/виводу;
- шлюзові пристрої (GD — Gateway Device), які забезпечують інтерфейс FF HSE до інших протоколів мереж (MODBUS, DeviceNet, PROFIBUS);
- головні пристрої (HD — Host Device), до яких належать конфігуратори, операторські робочі станції, які не належать до рівня HSE, але вміють зв'язуватися з пристроями HSE.

Основні характеристики FF H1 та FF HSE зведені в табл. 2.10.

Таблиця 2.10

ХАРАКТЕРИСТИКИ FF H1 ТА FF HSE

OSI	Характеристика	FF H1	FF HSE
	<i>NetArea</i>	рівень датчиків	рівень контролерів
прикладний	<i>AppProfile</i>	визначений через рівень користувача, типові функціональні блоки; функціонування алгоритмів управління процесом через інтелектуальні функції вузлів;	
	<i>AppService</i>	сервіси підрівня FMS: періодичний та аперіодичний (за запитом) обмін даними процесу між функціональними блоками вузлів через Словник Об'єктів: програмування та конфігурування вузла; управління станом вузлів; діагностичні сервіси; функції резервування;	
			доступ до FF H1 через шлюзові пристрої
	<i>AppModel*</i>	реалізовані через різні типи VCR – pull Видавець-Абонент ідентифікованого обміну (VCR Publisher-Subscriber) – push Видавець-Абонент обміну повідомленнями (VCR розсилка звітів) – клієнт-серверний обмін повідомленнями (VCR Client/server)	– push Видавець-Абонент ідентифікованого обміну для періодичного/ аперіодичного обміну даними процесу – клієнт-серверний обмін повідомл для параметричних даних
	<i>AppProcData</i>	VCR Publisher-Subscriber: обмежується кількістю VCR на кожен пристрій у мережі	
	<i>AppResolut</i>	залежить від конфігурації системи та пристроїв; типовий час Макроциклу 250 мс — 2 с	

Закінчення табл. 2.10

OSI	Характеристика	FF H1	FF HSE
	<i>TpService</i>	не визначений	визначені протоколом TCP
	<i>NtService</i>		визначені протоколом IP
канальний	<i>ChAddMode</i> **	Відправник-Адресат-msg (VCR Client/Server); Виробник-Споживач-msg (VCR Publish/Subscribe, VCR Report/Distribution)	Відповідно до протоколу EthernetII/IEEE 802.2/3
	<i>ChAccess</i>	з Арбітром (Активний Планувальник Зв'язків — LAS)	
фізичний	<i>PhInterface</i>	синхронний інтерфейс, модуляція постійної напруги, манчестерське кодування	Відповідно до протоколу EthernetII/IEEE 802.3
	<i>PhMedia</i>	екранована вита пара	
	<i>PhTopology</i>	шина/дерево	
	<i>PhLdrop</i>	<120 м (при $PhNodes < 12$), <1 м (при $PhNodes > 25$), $PhLength = \Sigma PhLdrop + Ltrunk$	
	<i>PhBaudRate</i>	31.25 кБіт/с	
	<i>PhSegment</i>	5 сегментів, термінатори в кінцях лінії (100 Ом + 1мкФ)	
	<i>PhNodes</i>	32 (12 — при подачі живлення, 6 — при подачі живлення для функц. небезпечних застосув.)	
	<i>PhLength</i>	1900 м (кабель типу А), 9600 м сегментована	
	<i>PhSupply</i>	до 30 В по інформаційній парі проводів	

* — умовна модель, у дужках вказана назва відповідно до документів FF

** — умовна модель, у документах FF назви моделей відсутні

*** — враховується час трафіка, час виконання FB, 50% виділяється для незапланованих повідомлень

2.5.4. CAN та CANOpen

Детально про мережу CAN Ви можете прочитати в розділі 8, про CANOpen — у розділі 9.

2.5.4.1. Походження CAN. На початку 80-х років виникла необхідність збору й обробки за короткі проміжки часу даних від багатьох датчиків, установлених в автомобілях. Це завдання можна було вирішити тільки при використанні мережної структури, що поєднує всі компоненти й послуговується для цієї мети недорогою, послідовною мережною структурою. З цієї причини фірма BOSCH (Німеччина) розробила мережу «Control Area Network» (CAN), що пізніше був затверджений як стандарт ISO 11898. Для координації зусиль виробників, розробників і користувачів CAN-систем і технологій створена некомерційна організація CiA (CAN in Automation), яка включає в себе більше 300 компаній.

У контексті моделі OSI, CAN Специфікація 2.0 (також називають **BOSH CAN 2.0 A/B**) описує повністю канальний рівень та частину фізичного. Їх функції реалізуються у вигляді як окремих мікросхем CAN контролера, так і вбудованих в готові мікроконтролери. Всі інші рівні, включаючи вимоги до фізичного середовища, способів передачі бітів, рівнів сигналу та роз'ємів, дана специфікація не описує. Сьогодні на базі CAN успішно функціонують багато промислових мереж, серед них: CANopen, DeviceNet, CAN Kingdom, SDS (Smart Distributed System), які доповнені протоколами верхніх рівнів. У 1993 році CAN закріпився в міжнародному стандарті **ISO 11898** (CAN High Speed Transceiver and Data Link Layer). Цей стандарт на відміну від BOSH CAN 2.0 A/B додатково описує вимоги до високошвидкісних трансиверів та середовища передачі даних. Саме ISO 11898 використовується у багатьох мережах, які базуються на CAN (наприклад, CANopen, DeviceNet).

2.5.4.2. Фізичний рівень CAN. На фізичному рівні BOSH CAN описує вимоги до бітової синхронізації, наявність двох рівнів бітів «домінантного» та «рецесивного». В ISO 11898 додатково визначені шинна топологія мережі, з диференційним методом передачі за напругою, NRZ-кодуванням, бітовими швидкостями від 10 кбіт/с до 1 Мбіт/с залежно від сумарної довжини лінії зв'язку (див. таб. 2.11).

Таблиця 2.11

ХАРАКТЕРИСТИКИ BOSH CAN 2.0 A/B ТА CAN ISO 11898

OSI	Характеристика	BOSH CAN 2.0 A/B	CAN ISO 11898
	<i>NetArea</i>	рівень датчиків	рівень датчиків
канальний	<i>ChAddMode*</i>	Виробник-Споживач-msg, 11-бітний ідентифікатор (A), 29-бітний (B)	Виробник-Споживач-msg 11-бітний ідентифікатор
	<i>ChAccess</i>	CSMA/CA	
	<i>ChChecksum</i>	бітова помилка; помилка начинення; помилка CRC; помилка форми; помилка підтвердження.	
фізичний	<i>PhInterface</i>	домінантний/рецесивний біт	асинхронний NRZ, доміантний/рецесивний біт
	<i>PhMedia</i>	не описаний	диференційний, передача напругою
	<i>PhTopology</i>		вита пара
	<i>PhLdrop</i>		шина
	<i>PhBaudRate</i>		довжина залежить від <i>PhBaudRat</i> 0.3 м (1 Мбіт/с)
	<i>PhSegment</i>		від 10 КБіт/с до 1 Мбіт/с; 20 КБіт/с (обов'язкова)
	<i>PhNodes</i>		5
	<i>PhLength</i>		64, незалежно від кількості сегментів
	<i>PhSupply</i>		1000 м (10 КБіт/с), 30 м (1 Мбіт/с), термінатори 120 Ом на обох кінцях
			допускається по окремим проводам

* — модель умовна, в стандартах назва не заявлена

2.5.4.3. Канальний рівень CAN. CAN — це послідовна шина, механізм доступу до якої описується моделлю CSMA/CA, який розглянутий вище. Можливі колізії, пов'язані з одночасним запитом шини, вирішуються на основі пріоритетності повідомлень, а саме — право на роботу із шиною одержить той вузол, що передає повідомлення з найвищим пріоритетом. Для цього кожен кадр містить 11-бітний ідентифікатор (29-бітний для CAN 2.0 B), що і є пріоритетом даного повідомлення. Призначення пріоритетів може відбуватися в такий спосіб: наприклад, один — для параметра швидкості, інший — для частоти обертання колінчатого вала двигуна й т.п. Кожен вузол-приймач у CAN-шині сам вибирає призначені для нього повідомлення (модель Виробник-Споживач-msg). У кожному повідомленні може бути передане від 0 до 8 байт даних. Більші блоки можна передавати за рахунок використання принципу сегментації повідомлень, яке необхідно вирішувати на верхніх рівнях.

2.5.4.4. Архітектура CANOpen. *CANOpen* — це мережна CAN-сумісна система, яка розроблена і підтримувана некомерційною організацією CiA. На фізичному та каналному рівнях *CANOpen* базується на CAN-контролері та високошвидкісному трансивері, які визначені ISO 11898 у частинах 1 та 2. Додатково до них указані вимоги до бітової синхронізації, характеристики кабелів та роз'ємів (CiA DRP 303). Нижні рівні доповнені протоколом прикладного рівня та комунікаційними профілями (CiA DS-301) які визначають доступ до даних процесу та параметричних даних вузла через його Словник Об'єктів. Цей Словник є своєрідним сховищем та буфером між даними вузла та мережею. Крім того, всі настройки комунікаційного обміну теж зберігаються в Словнику Об'єктів. Обмін даними процесу проходить шляхом періодичного/аперіодичного обміну об'єктами PDO (Process Data Object), які передаються CAN-кадром. При конфігуруванні мережі конкретні об'єкти TPDO на вузлі-джерелі та RPDO на вузлі-приймачі поєднуються логічним зв'язком, який визначає ідентифікатор CAN-кадру (модель ідентифікованого обміну). Відповідність вмісту цих кадрів конкретним даним вузлів визначається шляхом конфігурування наповнення PDO як з боку вузла-джерела, так і з боку вузла-отримувача.

Таблиця 2.12

ХАРАКТЕРИСТИКИ CANOPEN

OSI	Характеристика	CANOpen
	<i>NetArea</i>	рівень датчиків
прикладний	<i>AppService</i>	PDO — періодичний/аперіодичний обмін даними процесу через Словник Об'єктів; SDO — аперіодичний обмін параметричними даними через Словник Об'єктів, конфігурування вузла; NMT — управління станом вузлів, діагностичні сервіси;
	<i>AppModel</i>	– push/pull Виробник/Споживач ідентифікованого обміну для PDO; – клієнт-серверний обмін повідомленнями для SDO;
	<i>AppProfile</i>	великий перелік профілів пристроїв (CiA DS-4xx);
	<i>AppProcData</i>	PDO (8 байт): 512 TPDO/RPDO = 4096 байт
	<i>AppResolut</i>	залежить від реалізації, швидкості, типу PDO; приблизно 500 мс/100 байт (20 кбіт/с); 1 мс/100 байт (1 Мбіт/с)*

OSI	Характеристика	CANOpen
канал		Відповідно до ISO 11898-1 (таб. 2.8)
фізичний	<i>PhInterface</i>	Відповідно до ISO 11898-2 (таб. 2.8)
	<i>PhMedia</i>	
	<i>PhTopology</i>	
	<i>PhLdrop</i>	
	<i>PhBaudRate</i>	
	<i>PhSegment</i>	
	<i>PhNodes</i>	
	<i>PhLength</i>	
роз'єми		Відповідно до CiA DRP-303 (Sub-D, RJ-45, Open-Style, Mini-Style)

* дані орієнтовні, розраховані авторами посібника для циклічних PDO

Конфігурування вузлів, тобто аперіодичний обмін параметричними даними, проходить через об'єкти SDO. Цей обмін функціонує на основі клієнт-серверної моделі обміну повідомленнями. В клієнтському SDO вказуються ідентифікатор Об'єкта в Словнику вузла-Сервера та необхідна дія з цим об'єктом (читання/запис). Управління вузлами та їх діагностика відбувається через сервіси NMT.

Для спрощення механізму обміну з різнорідними пристроями в CANOpen використовується сімейство стандартизованих профілів (DS-4xx), які визначають перелік Об'єктів Словника та їх стандартне налаштування для різних типів пристроїв (засоби віддаленого вводу-виводу, PDS, засоби J1939 тощо).

2.5.5. Мережа LonWorks

2.5.5.1. Походження. Ідея рішення *LonWorks* («Local Operating Network Technology» або просто «LON») полягає в тому, що всі абоненти мережі мають у своєму розпорядженні власні інтелектуальні пристрої управління і можуть приймати рішення за місцем, без використання центрального контролера. Вузлами мережі можуть бути регулюючі пристрої, датчики, комп'ютери, комунікаційні пристрої і т.п. Технологія LonWorks належить американській фірмі «Echelon Corporation», яка була заснована в 1986 році. Сьогодні LON підтримують: у США — асоціація «LONMARK Interoperability Association», в Німеччині — «LON Nutzer Organisation e.V.» (LNO).

2.5.5.2. Структура LON-вузла. В кожен вузол LON-мережі вбудований чіп «Neuron-Chip», який спеціально розроблений для технології LonWorks (рис. 2.22). В цей чіп закладені програмні засоби: програма застосування, що забезпечує функціонування вузла в процесі застосування; операційна система, яка формує та реалізує комунікаційні функції. Для зв'язку між мережними вузлами застосову-

ється протокол **LonTalk**, який теж прописаний у чіпі. Протокол LonTalk описаний на всіх 7-мих рівнях моделі OSI.

Для конфігурування мережних вузлів та їх взаємодії використовуються спеціалізовані програмні засоби, які через LON-мережу доступаються до операційної системи кожного Neuron-Chip.

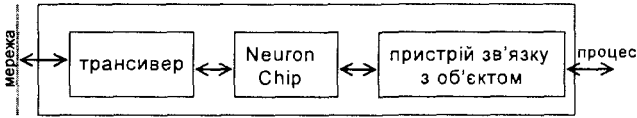


Рис. 2.22. Структура LON-вузла

2.5.5.3. Фізичний рівень. На фізичному рівні протокол LonTalk базується на мережах з сегментами, які підтримують різні середовища передачі. Сегмент, що побудований на одному середовищі в LON, називається **Каналом**. В якості середовища передачі можуть виступати: вита пара, силова проводка, радіочастота, інфрачервоні трансивери, коаксіальний кабель, оптоволокно. Канали в мережі об'єднуються за допомогою маршрутизаторів.

LON підтримує декілька різних типів трансиверів та відповідних методів функціонування Neuron-Chip та кодування даних, що залежать від середовища передачі:

- вбудований у Neuron-Chip трансивер з безпосереднім диференційним режимом функціонування Neuron-Chip (**Differential Direct Mode**) призначений для використання його при підключенні по витій парі без необхідності зовнішнього трансивера; в даному режимі використовується диференційне манчестерське кодування;

- зовнішній неінтелектуальний трансивер з безпосереднім уніполярним режимом функціонування Neuron-Chip (**Single-ended Direct Mode**) використовується при роботі через радіоканал, інфрачервоний порт, з оптоволокном, коаксіальним кабелем або іншими типами кабелів; у даному режимі використовується диференційний манчестерський код;

- зовнішній інтелектуальний трансивер зі спеціальним режимом функціонування Neuron-Chip (**Special Purpose Mode**), призначений для роботи з такими типами середовища передачі, що потребують особливого перетворення або модуляції, наприклад, при підключенні до силової проводки; в даному режимі використовується прямий метод кодування NRZ.

Використання диференційного манчестерського коду разом з витотою парою дає можливість не враховувати полярність при підключенні.

Топологія мережі, швидкість передачі даних, довжина ліній зв'язку, максимальна кількість вузлів, схеми термінування, споживаний мережею струм залежать від трансивера та середовища передачі. При передачі по витій парі ці параметри зведені в табл. 2.13. Шинна топологія потребує двох термінуючих резисторів у кінці, вільна топологія — одного.

2.5.5.4. Канальний рівень. MAC-процесор, що вбудований в Neuron-Chip, забезпечує управління доступом до середовища передачі. В протоколі LonTalk використовується алгоритм доступу до середовища predictive p-persistent CSMA, який розглянутий вище. Доставка даних проходить у широкомовному режимі, визначення вузла призначення відбувається на мережному рівні.

Таблиця 2.13

ПАРАМЕТРИ ПЕРЕДАЧІ

Продукт	Швидкість передачі	Топологія	Вузли	Відстань, м	Тип
TR/XF-78	78 кБіт/с	Шина	64	1400	Ізольований трансивер
TR/XF-1250	1,25 Мбіт/с	Шина	64	130	Ізольований трансивер
FTT-10A	78 кБіт/с	Шина	64	2700	Ізольований трансивер
FTT-10A	78 кБіт/с	Вільна	64	500	Ізольований трансивер
LPT-10	78 кБіт/с	Шина	128	2200	Для силової мережі
LPT-10	78 кБіт/с	Вільна	128	500	Для силової мережі

2.5.5.5. Мережний рівень. Кожен вузол у мережі при конфігурації отримує свій унікальний ідентифікатор — адресу LonTalk, що складається з адреси домену, підмережі та вузла. Домен — це логічний набір вузлів, що може включати декілька каналів (сегментів). Обмін даними між вузлами різних доменів неможливий. Якщо на одному каналі є вузли з різними доменними адресами, вони містяться в різних віртуальних мережах. Адреса домену може складатися з 0,1,3 або 6 байтів. Підмережа — це логічне об'єднання до 127 вузлів і вважається логічним каналом. В одному домені може бути до 255 підмереж. Обмін даними між вузлами підмереж проводиться за допомогою маршрутизаторів LON. Адреса вузла — це унікальний номер вузла в підмережі. Коли вузол логічно присутній в декількох підмережах, він має ту саму адресу в усіх підмережах. Для групової адресації вузли можуть об'єднуватися в логічні групи, незалежно від їх фізичного розташування. Один вузол може бути членом до 15 груп. В одному домені може бути до 256 груп.

Рисунок 2.23 показує приклад логічної адресації та фізичної топології в мережі LON. Як видно, маршрутизатори не є членами підмережі. Мости з'єднують канали та підмережі.

Повна адреса вміщує і адресу відправника, і адресу отримувача в одному з п'яти адресних форматів (табл. 2.14). Він є частиною пакета мережного рівня.

Для адресації вузлів може використовуватися NeuronID — унікальний 48-бітний ідентифікатор, що призначається вузлу на етапі виробництва. Як правило, цією адресою послуговуються при першому використанні даного вузла, наприклад для його конфігурування. Тим не менше, цей ідентифікатор може використовуватись при адресації вузла разом з адресою домену та підмережі.

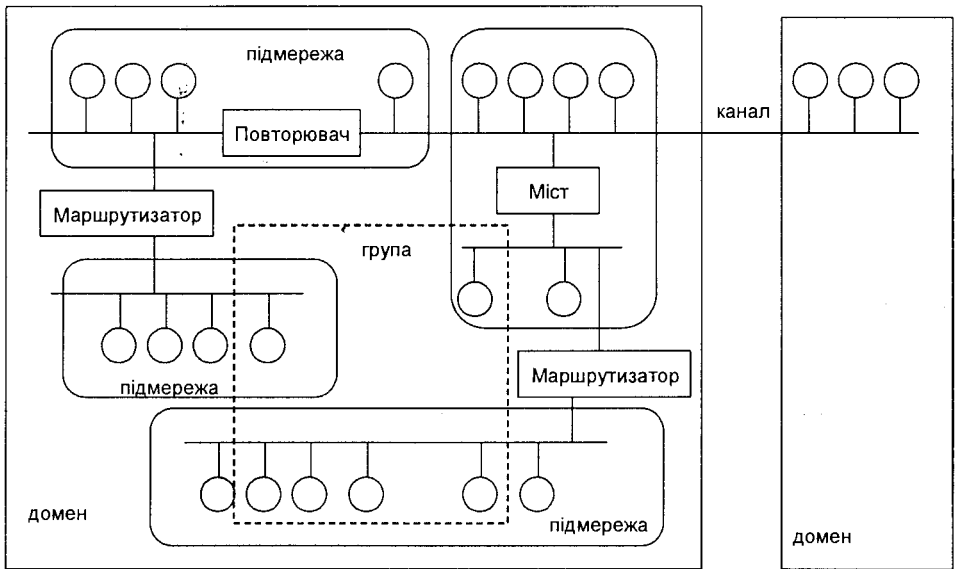


Рис. 2.23. Фізична топологія та логічна адресація в мережі LONWorks

Таблиця 2.14

ФОРМАТ АДРЕСИ

№	Формат адреси	Розмір адреси	Пункт призначення
0	Домен (підмережа=0)	3	Всі вузли в домені
0	Домен — підмережа	3	Всі вузли в підмережі
1	Домен — група	3	Всі вузли в групі
2a	Домен — підмережа — вузол	4	Один вузол у підмережі
2b	Домен — підмережа — вузол — група — член групи	6	Всі вузли групи (повинні відіслати підтвердження)
3	Домен — Neuron-ID	9	Конкретний вузол з Neuron-ID

2.5.5.6. Транспортний рівень. Протокол LonTalk надає 4-ри типи сервісів доставки даних:

1. сервіс з підтвердженням — забезпечує зв'язок між транспортними сутностями двох або більше вузлів, який вимагає підтвердження про отримання даних від кожного вузла-отримувача (надійний, гарантує доставку даних);

2. сервіс без підтвердження — дані відправляються на декілька вузлів один раз, отримання даних транспортною сутністю вузла-отримувача не контролюється (ненадійний, для швидкої передачі великих обсягів даних, не критичних до надійності доставки);

3. сервіс без підтвердження з повтореннями — дані відправляються на декілька вузлів кілька раз, без очікування підтвердження про доставку (ненадійний, для доставки даних великій кількості вузлів);

4. сервіс запит/відповідь — забезпечує взаємодію прикладних Процесів вузлів за моделлю Клієнт-Сервер; дані відправляються на один або декілька вузлів (повідомлення-запит), обробляються там, повертається підтвердження — повідомлення-відповідь, (надійний, гарантує доставку даних та повертає відповідь).

Функціонування перших трьох типів сервісів — завдання транспортного рівня, сервісу «запит-відповідь» — сеансового.

Для слідкування за послідовністю та кількістю TPDU пакетів в їх заголовку вказується ідентифікатор транзакцій, який з кожною відправкою збільшується на одиницю.

2.5.5.7. Сеансовий рівень. На даному рівні визначаються такі завдання: реалізація сервісу «запит-відповідь»; авторизація (захист від несанкціонованого доступу); сервіси мережного менеджменту та мережного інтерфейсу.

2.5.5.8. Рівень представлення даних. У кожному заголовку APDU вміщується інформація про інтерпретацію повідомлень, які надсилаються. Доступні такі типи повідомлень:

- вхідна мережна змінна;
- вихідна мережна змінна;
- явне повідомлення;
- зовнішнє повідомлення (використовується для шлюзів LonWorks);
- повідомлення мережної діагностики;
- повідомлення мережного менеджменту;
- повідомлення для сервісів «запит-відповідь».

2.5.5.9. Прикладний рівень. Разом з рівнем представлення даних такий рівень забезпечує функціонування 5-ти сервісів:

1. Обмін мережними змінними. Функціонування даного сервісу базується на концепції загальних мережних змінних (NV). Кожний вузол може використати вхідні для нього мережні змінні для керування прив'язаним для нього процесом управління. Значення кожної мережної змінної може змінити тільки один вузол, для якого ця мережна змінна є вихідною. Так, наприклад, датчик температури може змінювати значення вихідної для нього мережної змінної, записуючи туди плінне значення температури. Виконавчий механізм, що регулює подачу теплоагенту, приймає цю змінну в якості вхідної, і подає її значення на вимірювальний вхід вбудованого ПІ-регулятора, що забезпечує стабілізацію цієї температури. За допомогою програмного LON-конфігуратора розробник системи забезпечить зв'язок вихідних змінних одних вузлів з вхідними інших.

2. Обмін явними повідомленнями. Цей тип сервісу забезпечує передачу повідомлень, які мають вільний формат, відмінний від формату мережних змінних NV. Він передбачає обробку повідомлень за програмою користувача, що не прописано в протоколі прикладного рівня і може бути використаний для аперіодичного/періодичного обміну даними процесу.

3. Обмін зовнішніми повідомленнями. Використовується для шлюзу LonTalk при передачі даних між двома зовнішніми вузлами.

4. Сервіси мережного менеджменту. Забезпечують інсталяцію, конфігурування вузлів, завантаження прикладних програм, управління станом вузла.

5. Сервіс мережної діагностики. Сервіс використовується для тестування вузлів.

Процес зв'язку мережних змінних забезпечується при конфігуруванні системи. В кожному NeuronChip є **таблиця конфігурацій мережних змінних**, яка може вміщувати до 62 записів. Кожен з них характеризує конкретну мережну змінну вузла: тип, вхідна/вихідна, пріоритет, служба, селектор змінної (умовний ідентифікатор зв'язку) та адреса вузла, з яким зв'язана дана змінна. Таким чином, кожен вузол може мати до 62 мережних змінних.

2.5.5.10. Інтерфейс прикладного рівня. Зв'язок між прикладними Процесами вузлів та їх конфігурування забезпечується через два типи інтерфейсів прикладного рівня: сумісний та несумісний інтерфейс (рис. 2.24).

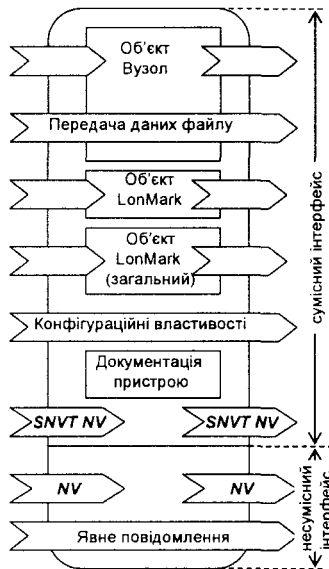


Рис. 2.24. Інтерфейс прикладного рівня LONWorks

Сумісний інтерфейс дещо схожий на інтерфейс прикладного рівня FF. Він базується на стандартизованих **LonMark-об'єктах**, які надають вузлу певну функціональність. З одного боку, сумісний інтерфейс використовується для обміну даними між цими об'єктами на різних вузлах, через стандартні типи мережних змінних (**SNVT**). З іншого боку, цей же інтерфейс надає можливість використання визначеного механізму конфігурування вузла через набір стандартних типів конфігураційних параметрів (**SCPT**). Передача конфігураційних параметрів а також програми проводиться за допомогою механізму передачі файлових даних, що використовує сервіси мережного менеджменту.

Поряд із сумісним інтерфейсом вузол може підтримувати несумісний інтерфейс, який дозволяє визначати специфічну поведінку вузла. Обмін між прикладними Процесами вузлів через несумісний інтерфейс проводиться через явні повідомлення (**Explicit Exchange**) або мережні змінні типу користувача.

У сумісному інтерфейсі всі об'єкти LonMark діляться на: об'єкти загального призначення, що використовуються в широкому спектрі завдань (наприклад,

об'єкт «датчик з відкритим циклом»), та спеціалізовані об'єкти (наприклад, об'єкт «датчик температури», «датчик тиску»). У будь-якому випадку об'єкт LonMark визначається набором мережних змінних SNVT та конфігураційних параметрів SCPT. Кожна мережна змінна SNVT має функціональне текстове ім'я, яке не може перевищувати 11 символів. Перші три символи тобто префікс назви змінної вказують на клас та напрямок мережної змінної: nvi — вхідна мережна змінна, запам'ятовується в RAM; nvo — вихідна мережна змінна, запам'ятовується в RAM; pci — вхідна конфігураційна мережна змінна, запам'ятовується в EEPROM; pvo — вихідна мережна змінна, запам'ятовується в ROM. При конфігуруванні системи, мережні змінні «поєднують», що приводить до появи нових записів у таблиці конфігурацій мережних змінних. Слід зазначити, що мережні змінні об'єктів не обов'язково повинні бути використані.

Розглянемо декілька об'єктів сумісного інтерфейсу. Для управління вузлом в цілому використовується спеціальний тип об'єкта «Вузол» (Node тип#0), наявність якого є обов'язковою для всіх вузлів, що мають сумісний інтерфейс. До складу об'єкта входять дві обов'язкові мережні змінні (nviRequest та nvoStatus) та декілька опціональних (рис. 2.26). Змінна nviRequest дозволяє управляти станом об'єктів вузла та робити запит на отримання статусної інформації про них, яка передається через вихідну змінну nvoStatus.

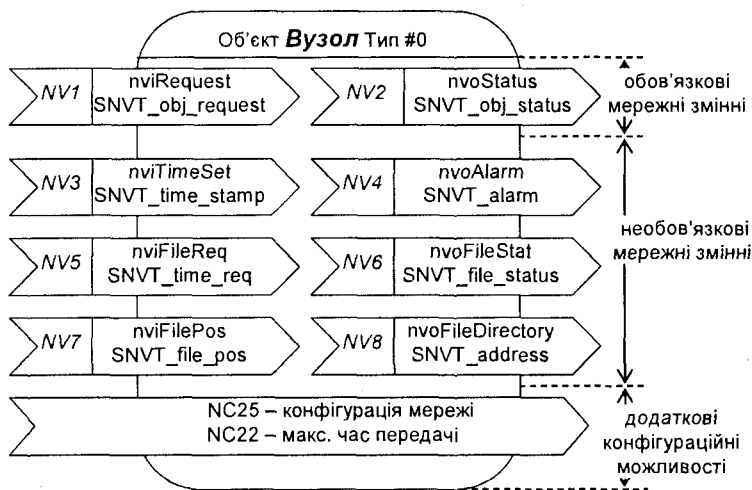


Рис. 2.26. Графічне представлення об'єкта «Node»

Об'єкт «датчик» призначений для отримання інформації з вузлів, які мають у своєму складі аналоговий або дискретний датчик. Через вихідну змінну nvoValue інформація про вимірювальну величину попадає в мережу. Об'єкт «виконавчий механізм» має вхід nviValue, який може управлятися такими об'єктами, як «датчик» або «контролер». Об'єкти «датчик» та «виконавчий механізм» можуть бути двох типів: з відкритим або закритим циклом. «Датчики» із закритим циклом (тип#2) мають у своєму складі входи nviValueFB для підключення зворотного

зв'язку від «виконавчого механізму». «Виконавчі механізми» із закритим циклом (тип#4) мають виходи *nvoValueFb* для зворотного зв'язку з «датчиками». «Датчик» з відкритим циклом (тип#1) не має вхідної мережної змінної для зворотного зв'язку, а «виконавчий механізм» з відкритим циклом (тип#3) — вихідної мережної змінної.

Об'єкт «контролер» (тип#5) дозволяє реалізовувати необхідні алгоритми управління, які вставляються між об'єктами «датчик» та «виконавчий механізм». Тип алгоритму, що прописаний у даному типі об'єкта, визначається його функціональним профілем. Наприклад, у функціональному профілі PID визначений алгоритм ПІД-закону регулювання, який потребує відповідних наборів конфігураційних параметрів. Приклад відкритої системи, побудованої на базі LonMark об'єктів, показаний на рис. 2.27.

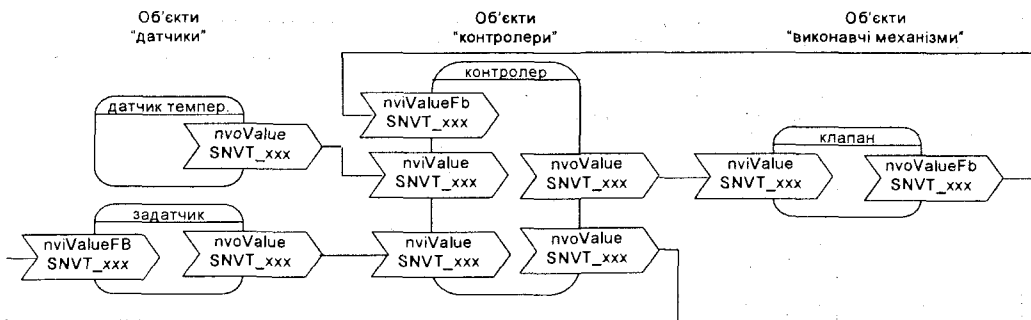


Рис. 2.27. Приклад реалізації системи для управління навколишнім середовищем

Таблиця 2.15

ХАРАКТЕРИСТИКИ LON WORKS

OSI	Характеристика	LON Works
	<i>NetArea</i>	– рівні датчиків/контролерів
Прикладний + + представлення + сеансовий	<i>AppService</i>	NV/SNVT — періодичний/аперіодичний обмін даними процесу між об'єктами вузлів; явні повідомлення — аперіодичний обмін повідомленнями; управління станом вузлів; діагностичні сервіси; SCPT/файлові дані — програмування або конфігурування вузла (параметричні дані); авторизація;
	<i>AppModel*</i>	Видавець-Абонент ідентифікованого обміну для для NV/SNVT/SCPT; Видавець-Абонент обміну повідомленнями для явних повідомлень, зовнішніх повідомлень; клієнт-серверна модель обміну повідомленнями для конфігураційних сервісів (файлові дані), сервісів мережної діагностики;
	<i>AppProfile</i>	використання LonMark об'єктів зі стандартного набору; функціонування алгоритмів управління процесом через інтелектуальні функції вузлів; файл зовнішнього інтерфейсу вузла .XIF

OSI	Характеристика	LON Works
Прикладний + + представлення + сеансовий	<i>AppProcData</i>	NV/SNVT: <63 мережних змінних на Вузол; загальна кількість необмежена
	<i>AppResolut</i>	залежить від конфігурації мережі, точно не визначається (через CSMA)
транспортний	<i>TrpService</i>	доступ до об'єктів; доступ до операційної системи;
	<i>TrpModel</i>	надійні з підтвердженням; надійні «запит-відповідь»; ненадійні без підтвердження; ненадійні без підтвердження з повторенням; слідування за транзакціями
мережний	<i>NtService</i>	5-рівнева адресація вузлів; див. табл. 2.14
каналний	<i>ChAddModel</i>	широкомовна адресація, ідентифікація вузлів через мережний рівень
	<i>ChAccess</i>	predective p-persistent CSMA
	<i>ChChecksum</i>	CRC16
	<i>ChSegment</i>	кількість сегментів необмежена
фізичний	<i>PhInterface</i>	з вбудованим трансивером Differential Direct Mode — диференціальний манчестерський код; зовнішній трансивер Single-ended Direct Mode - RS-485, інші інтерфейси: зовнішній трансивер Special Purpose Mode — використання модемів;
	<i>PhMedia</i>	залежить від <i>PhInterface</i> : вита пара; коаксіальний кабель; оптоволокно; радіохвилі; інфрачервоні хвилі; силова проводка;
	<i>PhTopology</i>	залежить від <i>PhInterface</i> ; для вбудованого трансивера див. табл. 2.13
	<i>PhLdrop</i>	залежить від <i>PhInterface</i> ;
	<i>PhBaudRate</i>	залежить від <i>PhInterface</i> ; для вбудованого трансивера див. табл. 2.13
	<i>PhSegment</i>	залежить від <i>PhInterface</i> ;
	<i>PhNodes</i>	залежить від <i>PhInterface</i> ; для вбудованого трансивера див. табл. 2.13
	<i>PhLength</i>	залежить від <i>PhInterface</i> ; для вбудованого трансивера див. табл. 2.13

* — модель умовна, в LONTalk визначається сервісами, що використовуються на транспортному рівні.

2.5.6. Мережі PROFIBUS

Детальніше про PROFIBUS Ви можете прочитати в розділі 7.

2.5.6.1. Походження. PROFIBUS (Process Field Bus) з'явився завдяки зусиллям групи німецьких компаній: Bosch, Siemens та Klockner-Moller. Сьогодні мережі PROFIBUS закріплені у міжнародних стандартах IEC 61158 та IEC 61784. Прошуванням та розвитком стандарту займаються організація PNO (PROFIBUS Nutzerorganisation) та міжнародна всесвітня організація PI (PROFIBUS International), що гарантує відкритість мережі і зв'язок між пристроями різних виробників.

PROFIBUS — це система, яка включає в себе 3 різні технології:

- **PROFIBUS DP** (Decentralized Periphery);
- **PROFIBUS FMS** (Fieldbus Message Specification);
- **PROFIBUS PA** (for Process Automation).

2.5.6.2. Фізичний рівень. Технології PROFIBUS-DP та PROFIBUS-FMS базуються на однакових реалізаціях фізичного та каналного рівнів, тому можуть функціонувати на одній і тій самій мережі. Фізичний рівень може бути реалізований на RS-485 інтерфейсі або на оптоволоконному з'єднанні з бітовою швидкістю до 12 Мбіт/с та символною передачею даних.

Мережа PROFIBUS-PA використовується в жорстких промислових умовах та функціональнебезпечному виробництві. На фізичному та каналному рівнях PROFIBUS-PA базується на технології передачі **MBP** (IEC 61158-2 Тип 3), яка використовує манчестерське кодування бітів, модулюючи струм живлення пристроїв підключених до шини. Бітова швидкість фіксована і становить 31.25 КБіт/с. Фізично PROFIBUS-PA може підключатися як сегмент до мережі PROFIBUS-DP через спеціальні мости.

2.5.6.3. Канальний рівень. На каналному рівні PROFIBUS-DP та PROFIBUS-FMS використовується протокол FDL, метод доступу до шини — гібридний (між Ведучими — маркерне кільце, з розподіленою периферією — Ведучий-Ведений). Канальний рівень PROFIBUS-PA базується на технології передачі MBP.

2.5.6.4. Прикладний рівень

Протокол PROFIBUS-FMS базується на клієнт-серверному обміні повідомленнями і використовується для читання/запису даних великого обсягу на рівні контролерів.

Таблиця 2.16

ХАРАКТЕРИСТИКИ МЕРЕЖ PROFIBUS

OSI	Характеристика	PROFIBUS FMS	PROFIBUS DP	PROFIBUS PA
	<i>NetArea</i>	контролерний рівень	рівень датчиків	рівень датчиків
прикладний	<i>AppService</i>	FMS — ациклічний доступ до даних процесу іншого вузла	<i>Data Exchange</i> — циклічно-періодичний обмін даними процесу між центральним вузлом та розподіленою периферією; DP-V1 — ациклічний обмін даними процесу між центральним вузлом та розподіленою периферією; DP-V2 — широкомовна розсилка значень даних Ведений-Ведені; управління станом вузлів; діагностичні сервіси; конфігурування вузла; функції резервування;	
	<i>AppModel</i>	клієнт-серверна модель обміну повідомленнями;	клієнт-серверна модель ідентифікованого обміну для DP Data-Exchange (Polling)*; модель Видавець-Абонент ідентифікованого обміну для DP-V2*;	
	<i>AppProfile</i>	профілі для FMS-пристроїв; описується GSD файлом	профілі в галузі застосування; великий перелік профілів пристроїв; описується GSD-файлом	
	<i>AppProcData</i>		DP Data Exchange: (244 Input + 244 Output) байт на Slave	

Закінчення табл. 2.16

OSI	Характеристика	PROFIBUS FMS	PROFIBUS DP	PROFIBUS PA
прикладний	<i>AppResolut</i>		залежить від конфігурації мережі; орієнтовно $3 \text{ мс} / 10 \times (2 \text{ вх.байт} + 2 \text{ вих.байт})$ при 1.5 Мбіт/с** орієнтовно $0.8 \text{ мс} / 10 \times (2 \text{ вх.байт} + 2 \text{ вих.байт})$ при 12 Мбіт/с***	
каналний	<i>ChAddModel</i>	Відправник — Адресат-msg, усі вузли адресуються 1-126, 127 — ширококомвна;		з боку DP видимий як модульний Ведений пристрій
	<i>ChAccess</i>	гібридний: Ведучий-Ведений + маркерний доступ між Ведучими		Ведучий-Ведений;
	<i>ChChecksum</i>	побітове AND		
фізичний	<i>PhInterface</i>	RS-485, IR, оптоволокно, радіохвилі		синхронний інтерфейс, модуляція постійної напруги, манчестерське кодування
	<i>PhMedia</i>	для RS-485 — екранована вита пара, стандарти кабелів А та В; для оптичного зв'язку — пластиківі, скляні, РСF		екранована вита пара типів А-D;
	<i>PhTopology</i>	шина, деревовидна через повторювачі/мости;		шина/дерево
	<i>PhLdrop</i>	для RS-485 допускається тільки з використанням активних з'єднувачів;		$<30 \text{ м}, PhLength = \Sigma PhLdrop + Ltrunk$
	<i>PhBaudRate</i>	від 9600 кБіт/с до 12 Мбіт/с		31.25 КБіт/с
	<i>PhSegment</i>	сегментація з використанням репітерів/мостів, сегментних з'єднувачів		
	<i>PhNodes</i>	32		
	<i>PhLength</i>	до 1000 м (при 187 кБіт/с), термінатор 390 Ом в обох кінцях лінії		1900 м, термінатор (100 Ом + 1 мкФ) в обох кінцях лінії
<i>PhSupply</i>	не передбачено		до 30 В по інформаційній парі проводів	

* — в офіційних документах PROFIBUS-DP прикладні сервіси та їх функціонування визначаються профілем пристрою;

** — модель умовна, в офіційних документах PROFIBUS-DP прикладні сервіси та їх функціонування визначаються профілем пристрою;

*** — дані, взяті з PNO Profibus Description, для 10 пристроїв на 2 вхідні та 2 вихідні байти

PROFIBUS-DP та PROFIBUS-PA призначені для обміну даними процесу в реальному часі центрального вузла з розподіленою периферією. Сервіси прикладного рівня в PROFIBUS-DP/PA реалізовані на рівні профілю. Умовно їх функціонування можна описати клієнт-серверною моделлю ідентифікованого обміну (Polling). Цей обмін реалізується через профілювання пристроїв, в якому визначене призначення, формат та обсяг даних конкретного пристрою, описаного через GSD-файли. При конфігуруванні системи Ведучого вказується обсяг та призначення буфера вводу/виводу для кожного Веденого PROFIBUS-DP/PA. Веду-

чий циклічно або періодично опитує кожний Ведений вузол, передаючи йому дані для виводу та забираючи вхідні дані (сервіс Data-Exchange). Розвиток технології PROFIBUS-DP привів до появи нових можливостей, зокрема ациклічного обміну ідентифікованими даними та ідентифікованого обміну Видавець-Абонент і т.д. Нині існують три версії протоколу PROFIBUS-DP: DP-V0 (циклічний обмін ідентифікованими даними Data-Exchange), DP-V1 (+ациклічний обмін ідентифікованими даними), DP-V2 (+обмін Видавець-Абонент).

2.5.7. HART-протокол

2.5.7.1. Походження. HART-протокол (Highway Addressable Remote Transducer) розроблений фірмою Rosemount Inc. у середині 80-х років, реалізує стандарт BELL 202 FSK (Frequency Shift Keying) для організації цифрової передачі, заснованої на технології 4-20 мА. HART — це відкритий стандарт, що працює з будь-якою системою управління й підтримується всіма провідними виробниками устаткування й програмного забезпечення в автоматизації. HART-протокол, по суті, доповнює технологію 4-20 мА цифровими можливостями, а саме:

- дозволяє передавати одночасно аналоговий і цифровий сигнали однією виною парою, причому зберігається повна сумісність і надійність існуючих аналогових ліній 4-20 мА;

- дає можливість дистанційно здійснювати діагностику й налаштування польових приладів, використовуючи для цього HART-комунікатор або комп'ютер з відповідним ПЗ; це особливо зручно, коли датчики розташовані у важкодоступних місцях, на великих відстанях один від одного, а також в умовах шкідливих або небезпечних виробництв.

- скорочує затрати на кабельну продукцію, установку, налагодження й на поточне технічне обслуговування;

- дає зручність при роботі з багатопараметричними приладами, тому що можна одержувати інформацію про декілька змінних процесу по одній парі проводів;

- прилади, що підтримують HART-протокол, можуть установлюватись у вибухонебезпечних зонах класу 0, класу 1 і класу 2;

- неперервна самодіагностика забезпечує високу надійність устаткування; інформація про стан приладу передається у кожному повідомленні від пристрою;

- користувач має можливість прочитати будь-які параметри датчика: значення змінних, одиниці й діапазон вимірювання, індивідуальні параметри приладу.

Сьогодні доступна також HART-сумісна технологія Wireless HART, яка в декількох словах описана в розділі 4.

2.5.7.2. Фізичний рівень

Існують два режими роботи датчиків, що підтримують HART — протокол:

1. Режим точка-точка, коли передача цифрової інформації проходить *одночасно з аналоговим сигналом*; при цьому можна на відстані (до 3000 м) здійснювати налаштування й конфігурування датчика й операторові немає необхідності обходити всі датчики — їх можна настроїти зі свого робочого місця.

2. У багатоточковому режимі датчик передає й одержує *інформацію тільки в цифровому вигляді*, а аналоговий сигнал датчиків виставляється рівним 4-мА; ін-

формація про змінні процесу зчитуються за HART-протоколом; до однієї пари проводів може бути підключено до 15 датчиків, їх кількість визначається довжиною і якістю лінії, а також потужністю блока живлення датчиків.

У HART-мережі по двом проводам здійснюється живлення датчика, прийом від нього сигналу про значення параметра, а також настроювання датчика за допомогою HART-модему.

В основі HART-протоколу використовується принцип частотної модуляції (ЧМ). Частотно-модульований сигнал є двополярним і при застосуванні відповідної фільтрації не впливає на основний аналоговий сигнал 4–20 мА. Відповідно до HART-протоколу, цифровий сигнал малої амплітуди (± 0.5 мА) накладається на аналоговий сигнал 4–20 мА (рис. 2.27). Відповідно зі стандартом Bell-202 для передачі логічної «1» HART використовує один повний період частоти 1200 Гц, а для передачі логічного «0» — два неповні періоди 2200 Гц. Оскільки такий Чм-сигнал має нульове середнє значення й фаза сигналу безперервна, то на аналоговий сигнал 4–20 мА він ніяк не впливає. Чм-складова накладається на струмову петлю 4–20 мА. Бітова швидкість — 1200 біт/с.

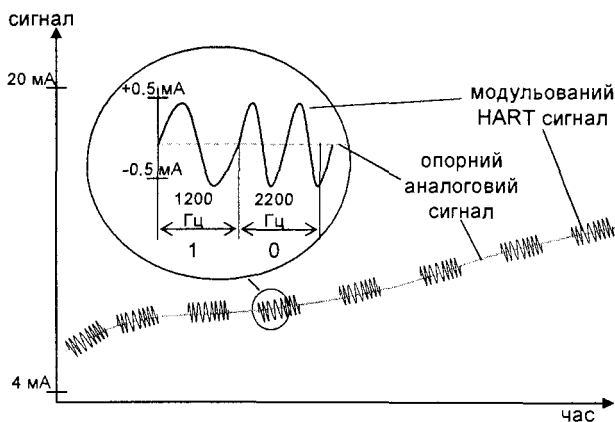


Рис. 2.27. Діаграма накладання цифрового частотно-модульованого сигналу малої амплітуди (± 0.5 мА) на аналоговий сигнал 4–20 мА

Стандартна топологія — зірка (декілька променів до ведучого пристрою), при багатоточковому з'єднанні — шина (15 вузлів підключені до однієї вити пари).

Прилади з HART-інтерфейсом можуть підключатися за допомогою таких способів (рис. 2.28):

- через спеціалізовані модулі HART;
- через HART-Модем, за допомогою якого встановлюється з'єднання «точка-точка» між комп'ютером і HART-пристроєм;
- через HART-мультиплексори.

Найбільш прийнятним засобом для підключення декількох ведених пристроїв в єдину систему є *мультиплексор*, що забезпечує доступ до них через інтерфейс RS-485, RS-232 і Ethernet.

2.5.7.3. Канальний рівень. На каналному рівні використовується метод доступу Ведучий-Ведений з можливістю 2-х Ведучих вузлів. Ведучий формує запити для Веденого, який на них відповідає. При з'єднанні точка-точка Ведучий вузол може замовити так званий пакетний режим передачі, при якому Ведений сам відправляє повідомлення з даними з певною періодичністю.

2.5.7.4. Прикладний рівень

Для передачі даних у шинній топології використовується класична схема Клієнт-Сервер. Ведучий завжди є Клієнтом і генерує запити до Веденого на отримання певних величин, реальних даних та будь-яких інших параметрів, наявних у пристрої. Ведений пристрій інтерпретує ці команди відповідно до HART-протоколу. У відповідь Ведучому пристрою передається інформація про статус і значення його параметрів. За одну послілку один вузол може передати іншому до 4 технологічних змінних, а кожний HART-пристрій може мати до 256 змінних, що описують його стан. Контроль коректності переданих даних заснований на одержанні підтвердження. При пакетному режимі (тільки в режимі точка-точка) після запиту Клієнта Сервер Веденого періодично формує повідомлення.

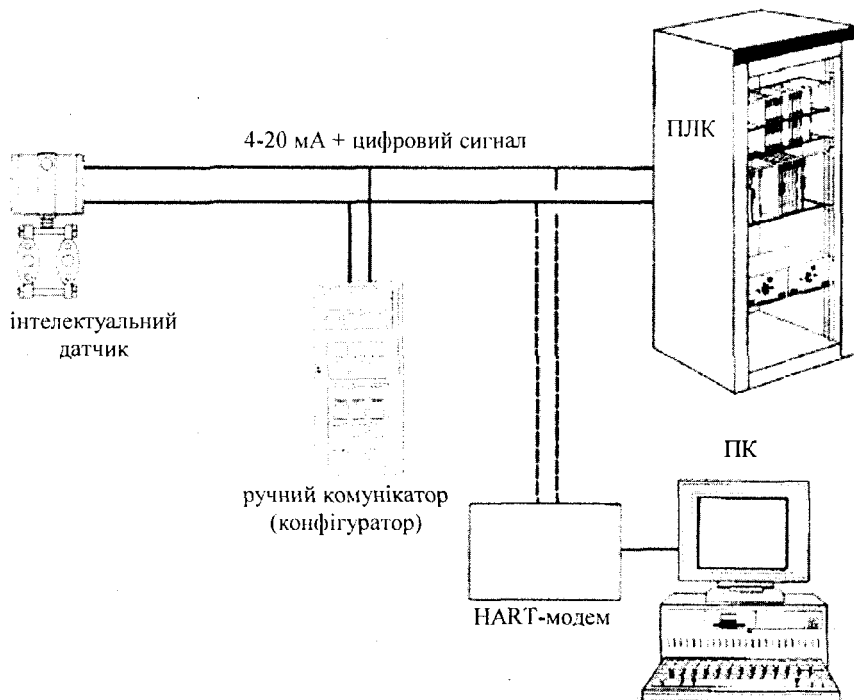


Рис. 2.28. Приклад системи з використанням HART в режимі точка-точка

Весь набір команд, реалізованих в HART-протоколі, умовно можна розділити на три групи.

1. Універсальні — основні команди, які підтримуються всіма типами ведених пристроїв: зчитування інформації про тип засобу, фірми виготовлювача; зчитування вимірювального параметра (PV) та одиниці виміру; зчитування струмового виходу та процента діапазону; зчитування наперед визначених динамічних змінних; зчитування символічних змінних, дати; зчитування/запис 32-символьного повідомлення; зчитування діапазону датчика, одиниць виміру та константи часу демпфування; зчитування серійного номера чутливого елемента та обмеження; зчитування/запис номера останньої зборки; запис адреси для багатоточкового режиму.

2. Стандартні (для груп пристроїв) — команди, які використовуються практично у всіх HART-пристроях.

3. Специфічні (що залежать від пристрою) — команди настроювання специфічних, індивідуальних параметрів якого-небудь пристрою.

Таблиця 2.17

ХАРАКТЕРИСТИКИ HART-ПРОТОКОЛУ

OSI	Характеристика	HART «точка-точка»	HART багатоточкове з'єднання	
	<i>NetArea</i>	рівень датчиків		
прикладний	<i>AppService</i>	періодичний/аперіодичний обмін вимірювальними даними пристрою (дані процесу); управління станом вузла; діагностичні сервіси; конфігурування вузла (параметричні дані);		
	<i>AppModel</i>	клієнт-серверна модель обміну повідомленнями; модель Клієнт — Сервер з пакетним режимом;	клієнт-серверна модель обміну повідомленнями;	
	<i>AppProcData</i>	до 256 змінних на пристрій, до 4 змінних за посилку		
	<i>AppResolut</i>	пакетний режим: 1 с/ 3 транзакції*	1 с / 2 транзакції*	
каналний	<i>ChAddModel</i>	Відправник-Адресат-msg: короткий формат адреси (0–15), де 1-Ведучий — контролер; 0-Ведучий — комунікатор; довгий формат (>=V4) + 4 байти на унікальний ідентифікатор пристрою		
	<i>ChAccess</i>	Ведучий-Редений, дозволяються два Ведучі (контролер+комунікатор)		
	<i>ChChecksum</i>	XOR		
фізичний	<i>PhInterface</i>	BELL 202 FSK (Frequency Shift Keying) модуляція несучої 4-20 мА		
	<i>PhMedia</i>	вита пара		
	<i>PhTopology</i>	точка-точка, зірка	шина	
	<i>PhLdrop</i>	приймаються в розрахунок <i>PhLength</i>		
	<i>PhBaudRate</i>	1200 біт/с		
	<i>PhNodes</i>	3	15 (для V < 5); 38-разр. Адреса (для V >=5)	
	<i>PhLength</i>	1500 м		
	<i>PhSupply</i>	по інформаційній парі проводів		

* — 1 транзакція вміщує до 25 байт, до 4 змінних

2.5.8. Мережа AS-I

Детальніше з AS-i Ви можете познайомитись у розділі 5.

2.5.8.1. Походження. Назва мережі розкриває її призначення: *AS-I* (Actuator Sensor Interface) — інтерфейс із датчиками й виконавчими механізмами. Уперше ASI-мережа вийшла на ринок наприкінці 1989 року, а сьогодні підтримана рядом відомих фірм. Існує й однойменна асоціація з підтримки цієї мережі, ASI.

2.5.8.2. Фізичний рівень. AS-i-інтерфейс, передусім, призначений для підключення дискретних датчиків та виконавчих механізмів до контролера (Ведучого вузла). По інформаційній шині передається живлення, що особливо зручно при підключенні великої кількості фотоелектричних й індуктивних датчиків. Спрощено ASI-мережа може мати вигляд, як на рис. 2.29. В якості середовища передачі використовується пара звичайних провідників або спеціальний ASI-кабель («жовтий кабель»), у якому обидва провідники упаковані в спеціальну м'яку гумову оболонку, що робить цей кабель гнучким і стійким до багаторазових вигинів. Цей кабель використовується для приєднання датчиків, установлюваних на рухливих частинах механізмів. Топологією AS-i-мережі може бути шина, зірка або дерево. Довжина сегмента мережі до 100 м. За рахунок репітерів довжину мережі можна збільшувати до 300 м.

Для кодування даних використовується Манчестерський код. Такий тип кодування знижує вплив на AS-i-кабель зовнішніх збурень. Бітова швидкість фіксована і рівна 167 кБіт/с.

2.5.8.3. Канальний рівень.

Для доступу до шини використовується метод Ведучий-Ведений, де Ведучим виступає контролер, а Веденими можуть бути інтелектуальні AS-I-датчики або виконавчі механізми. До шини також можна підключати звичайні датчики та виконавчі механізми через спеціальні AS-I модулі. Кожен Ведений може обслужити до 4-х дискретних входів та 4-х дискретних виходів. Максимальна кількість Ведених, підключених до одного Ведучого вузла — 31 (для версії 1) або 62 (для версій 2 та 3). Таким чином, до однієї мережі AS-i може бути підключено одночасно до 124 дискретних входів та 124 дискретних виходів (для версії 1) і в два рази більше дискретних датчиків та виконавчих механізмів для версій 2 та 3.

Короткий ASI-цикл забезпечується за рахунок компактного формату телеграми. Цикл опитування Ведучим 31-го Веденого укладається в 5 мс.

У старших версіях протоколу AS-i значно збільшилися його можливості, що дозволяють підключати до мережі аналогові датчики та ВМ. Адреса кожного мережного пристрою записується в його постійній пам'яті. AS-i Ведучий може бути організований як у вигляді шлюзу в іншу промислову мережу, так і у вигляді окремої плати/модуля контролера або комп'ютера.

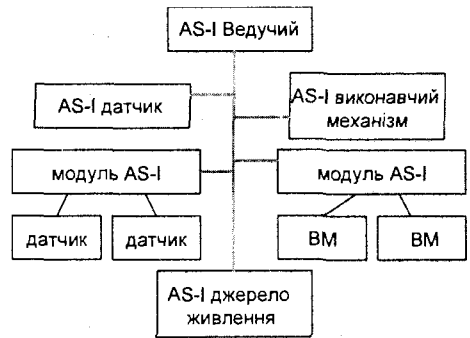


Рис. 2.29. Приклад AS-i-структури

ХАРАКТЕРИСТИКИ МЕРЕЖІ AS-I

OSI	Характеристика	AS-i
	<i>NetArea</i>	рівень датчиків
прикладний*	<i>AppService*</i>	Data Exchange — циклічне зчитування/запис даних процесу (входи/виходи); конфігурування та управління станом вузлів (параметричні дані); діагностичні сервіси;
	<i>AppModel*</i>	клієнт-серверна модель ідентифікованого обміну (Polling) для Data Exchange; клієнт-серверна модель обміну повідомленнями для параметричних даних;
	<i>AppProfile</i>	профілі віддалених дискретних входів/виходів, аналогових засобів вводу/виводу ($\geq V2.0$), специфічні профілі
	<i>AppProcData</i>	124 дискр. вх., до 124 дискр. вих. (248/248 для V2.0, V3.0)
	<i>AppResolut</i>	5 мс / 31 * (4 DI + 4 DO)
каналний	<i>ChAddModel</i>	Відправник-Адресат-msg; адресуються тільки Ведені: 1-31 (V1), 1-62 (V2/3)
	<i>ChAccess</i>	Ведучий-Ведений
	<i>ChChecksum</i>	біт паритету
фізичний	<i>PhInterface</i>	асинхронний інтерфейс, модуляція постійної напруги, манчестерське кодування
	<i>PhMedia</i>	пара проводів, «жовтий кабель»
	<i>PhTopology</i>	шина, зірка, дерево
	<i>PhLdrop</i>	будь-яке відгалуження, $PhLength = \sum PhLdrop + Ltrunk$
	<i>PhBaudRate</i>	167 КБіт/с
	<i>PhSegment</i>	3 сегменти з 2 репітерами, термінування не потрібне
	<i>PhNodes</i>	31 + Ведучий (V1), 62 (V2/3) + Ведучий
	<i>PhLength</i>	100 м, 300 м з репітерами; термінування не потрібне
<i>PhSupply</i>	до 30 В по інформаційній парі проводів (до 2А), можливе живлення по окремій парі проводів	

* — модель умовна, в AS-i відсутній прикладний рівень, прикладні сервіси та їх функціонування визначаються профілем пристрою.

2.5.9. Мережа INTERBUS

2.5.9.1. Походження. Робота над створенням *INTERBUS* почалася ще в 1983 році відомою німецькою організацією Phoenix Contact, а в 1987 результати були представлені на ГанOVERСЬКІЙ ярмарці. Початкова область застосування даної мережі, яка називалась *Interbus-S*, — високошвидкісний детермінований обмін даними з датчиками та виконавчими механізмами. З розвитком PC-сумісних рішень у промисловій автоматизації функціональність мережі доповнювалась, її призначення розширилось на розподілені комунікаційні структури, що базуються

на стандартних PC-технологіях, а сама промислова мережна система стала називатись INTERBUS. Сьогодні рішення INTERBUS підтримують та розвивають організації **INTERBUS Club** (www.interbusclub.com) та **Open Control Foundation**. Мережі закріплені в німецькому стандарті DIN 19258, європейському — EN 50254, та міжнародному — IEC 61158.

INTERBUS є мережею рівня датчиків. Вона має кільцеву топологію, в якій обмін даними організовується шляхом передачі полів кадру від пристрою до пристрою за кільцем. У голові мережної системи міститься Ведучий, який виконує функції контролера або забезпечує зв'язок INTERBUS з верхніми рівнями управління. Всі інші вузли системи є Веденими пристроями, які разом із Ведучим організовують фізичне кільце.

2.5.9.2. Фізична структура. Всі пристрої мережі підключаються один до одного через дуплексний зв'язок, організовуючи тим самим прямий та зворотний канали кільця (рис. 2.30). Тобто кожний вузол, окрім вузла Ведучого, має два дуплексні канали — *Вхідний Інтерфейс (incoming interface)* та *Вихідний Інтерфейс (outgoing interface)* і за відсутності відгалужувачів, кабельний зв'язок утворює видиму лінійну топологію, в якій кінцевий пристрій у мережі всередині замикає кільце.

Фізична структура мережі INTERBUS складається з таких елементів:

1. *Мережний Ведучий (Bus Master)*, який управляє мережею та виконує функції контролера.

2. *Мережні Пристрої (Bus Devices)*, які виконують функції засобів вводу/виводу, і залежно від підтримуваних вхідних/вихідних інтерфейсів можуть бути таких типів:

- a. *Віддалений Пристрій (Remote Device)*;
- b. *Локальний Пристрій (Local Device)*;
- c. *Пристрій Петлі (Loop Device)*.

3. *Мережні Відгалужувачі (Bus Couplers)*, інша назва мережні термінальні модулі (Bus Terminal Module), які забезпечують сегментацію мережі, тобто відгалуження від основної гілки.

4. Мережні кабелі.

Усі елементи поєднуються між собою через Вхідні та Вихідні Інтерфейси, фізична реалізація яких може бути різною: дуплексний варіант RS-485; з'єднання за TTL; модуляція струму живлення 24 В; оптичне кільце. Незалежно від структури мережі, бітова швидкість є постійною 500 кБіт/с, сумарна максимальна довжина лінії зв'язку — 13 км, максимальна кількість пристроїв — 512, максимальна кількість точок вводу/виводу — 4096.

Деякі Пристрої та мережні з'єднувачі, які починаються з Мережного Ведучого або Мережного Відгалужувача та закінчуються наступним Мережним Відгалужувачем або кінцевим пристроєм (без підключеного Вихідного Інтерфейсу), називають *Мережним Сегментом (Bus Segment)*.

У структурі мережі INTERBUS, залежно від типу Інтерфейсів, можуть використовуватись такі основні типи Сегментів:

- *Віддалений Сегмент (Remote bus segment)*, на базі RS-485 або оптоволокні;
- *Локальний Сегмент (Local bus segment)* або *I/O Шина (I/O Bus)* на базі TTL;
- *Сегмент INTERBUS Петлі (INTERBUS Loop segment)*, на базі модуляції струму живлення 24 В.

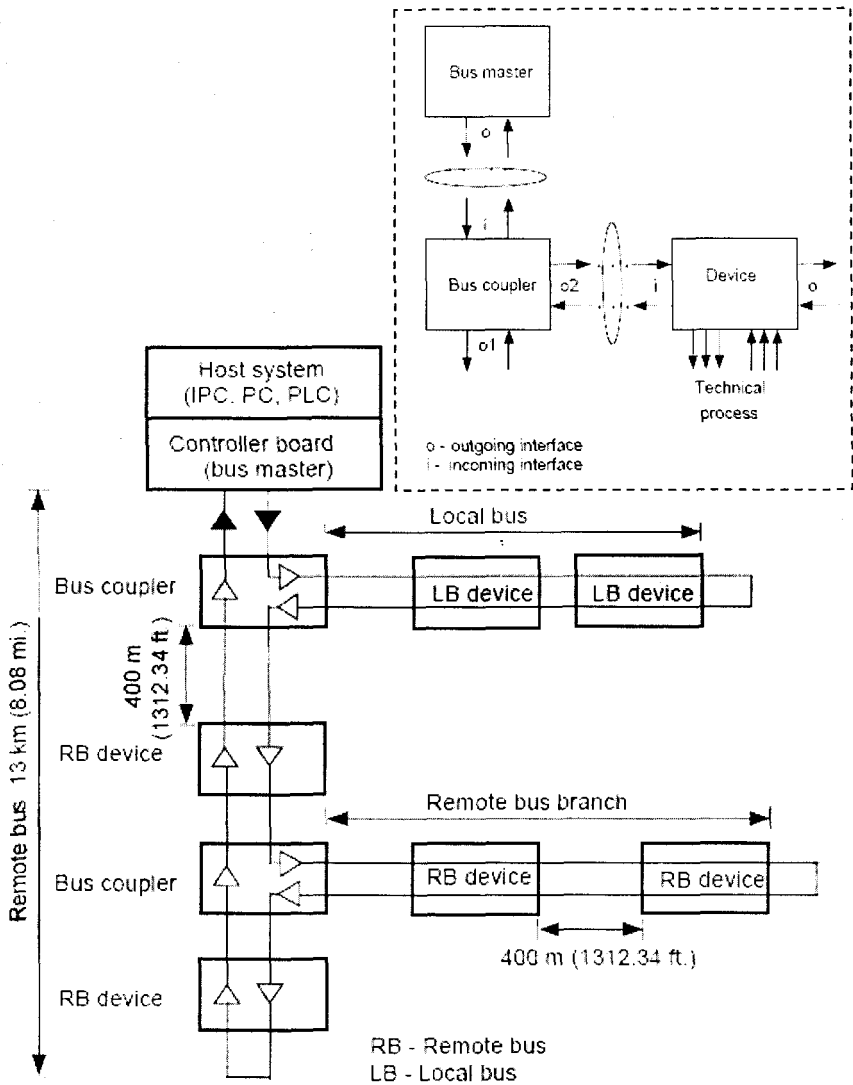


Рис. 2.30. Загальна структура мережі INTERBUS

Структура мережі дозволяє ієрархію Сегментів до 16 рівнів. Тобто Сегмент може включати декілька Відгалужувачів, Сегменти яких, у свою чергу, можуть також включати відгалужувачі. Сегментація INTERBUS дає зручний та прозорий механізм для довільного територіального розміщення пристроїв по всій території ділянки процесу. Крім того, сегментація дозволяє використовувати в одній і тій же мережі різні середовища передачі залежно від умов експлуатації.

2.5.9.3. Віддалений Сегмент. Мережний Ведучий підключається до наступного Віддаленого Пристрою, або Мережного Відгалужувача, через дуплексний

варіант інтерфейсу RS-485. Таким чином основним сегментом INTERBUS є Віддалений Сегмент (рис.2.30). Від нього можуть іти відгалуження на інші сегменти такого ж або іншого типу. Слід зазначити, що відгалуження буде тільки з точки зору кабельних відводів, але топологія залишиться кільцевою. Передача бітів проводиться методом NRZ.

До Віддаленого Сегмента пред'являються такі вимоги (рис. 2.31): максимальна кількість Пристроїв на сегмент — 256, максимальна дистанція між Пристроями — 400 м для RS-485 та 3600 м для оптоволокна, максимальна довжина сегмента — 12,8 км, живлення мережних засобів — локальне, стандартний з'єднувач — 9-піновий SUB-D, ви́лка для Вхідного Інтерфейсу (піни: DO – 1; / DO – 6; DI – 2; / DI – 7; GND – 4) та 9-піновий SUB-D розетка для Вихідного (піни: DO – 1; /DO – 6; DI – 2; /DI – 7; GND – 4, RBST – 9; +5V – 5).

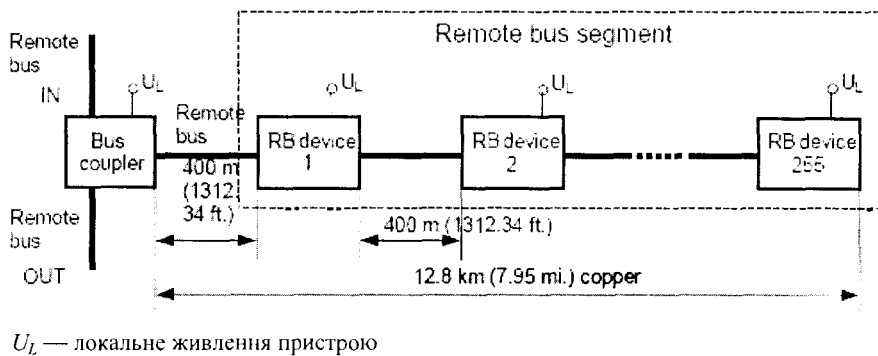


Рис. 2.31. Структура Віддаленого Сегмента.
 U_L — локальне живлення пристрою

Внутрішня структура Віддаленого Пристрою показана на рис. 2.32. Як видно, Вхідний та Вихідний Інтерфейс реалізований за допомогою RS-485 інтерфейсу, в якого вхідні та вихідні сигнали А, В, А', В' позначені як /DI, DI, /DO та DO. Сигнал RBST на Вихідному Інтерфейсі повинен бути підключений до 5В (перемичка між 9-тим та 5-тим піном), що повідомляє даному пристрою, що він не є кінцевим у ланцюгу. Перемичка виставляється безпосередньо на конекторі самого кабелю, щоб при підключенні наступного вузла Пристроїв ставав проміжним у ланцюгу, при відключенні — кінцевим, тобто замикав ланцюг. Обов'язковим є підключення сигнальної землі (пін 4). Таким чином, для з'єднання Віддалених Пристроїв необхідно використати кабель з 3-ма витими парами.

Віддалений Сегмент не живить пристрої по мережному кабелю. Така можливість доступна для рішення з назвою Installation Remote Bus. У цьому випадку використовується 9-жильний кабель (3 виті пари + 3 проводи на живлення), а максимальна довжина сегмента обмежується 50 м.

2.5.9.4. Локальний Сегмент. Локальний Сегмент, який також називається і О Шиною, призначений для об'єднання між собою Локальних Пристроїв на невеликих відстанях (наприклад, модульні острови вводу/виводу). Внаслідок цього

немає необхідності у використанні інтерфейсу RS-485, а використовується TTL — рівні сигналів (5В), які застосовуються в обчислювальній техніці. Локальний Сегмент починається зі спеціального Мережного Відгалужувача, до якого один за одним підключаються Локальні Пристрої. Максимальна відстань між Локальними пристроями — 1.5 м, максимальна довжина сегмента — 10 м. Окрім сигналів DI, /DI, DO та /DO, додатково до Локальних Пристроїв з Мережного Відгалужувача підводиться живлення. Локальні Сегменти доступні в декількох конструктивних варіантах: Локальний Сегмент з ST-шиною; Локальний Сегмент з убудованою станцією (INLINE Station); Локальний Сегмент на оптоволоконному з'єднанні. Для Локального Сегмента максимальна кількість Локальних Пристроїв — 63 (8 для Сегмента з ST-шиною).

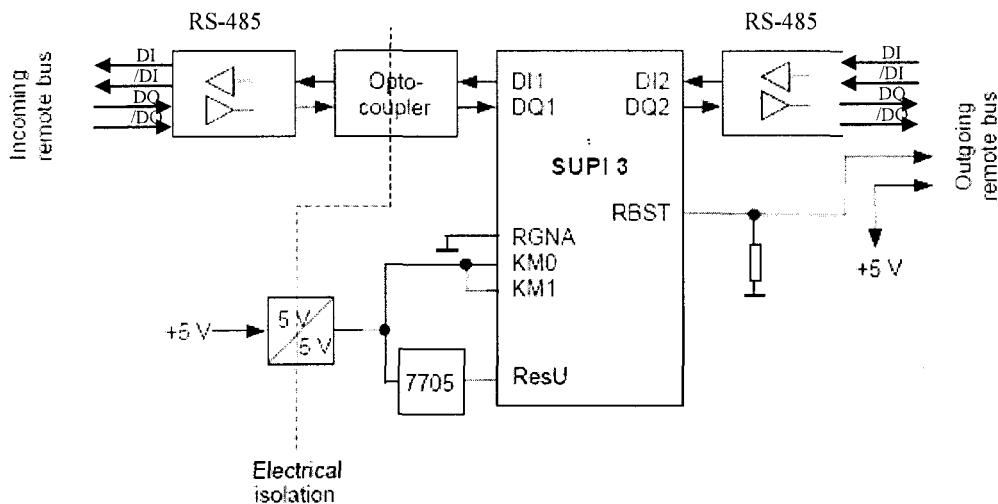


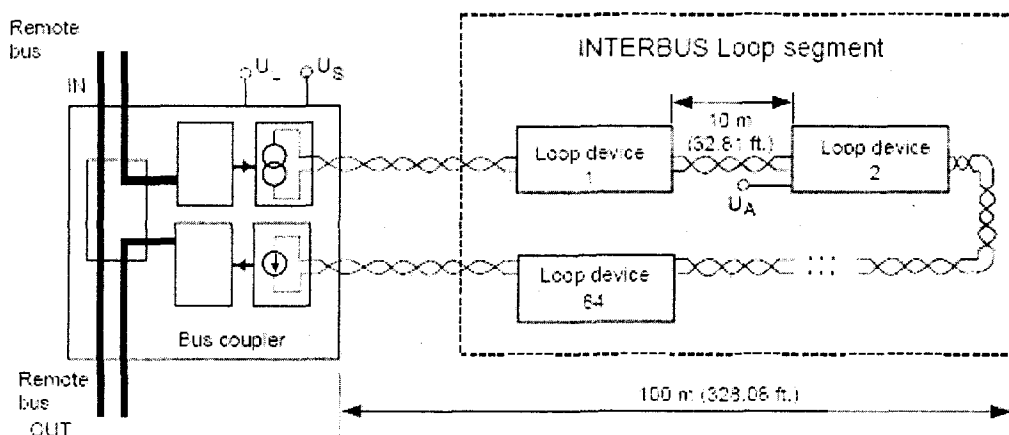
Рис. 2.32. Структура Віддаленого Пристрою

2.5.9.5. Сегмент INTERBUS Петлі. Даний тип сегмента, який також називається Петля Датчиків (Sensor loop, IP65 local bus), використовує 2-провідну лінію передачі, яка в поєднанні з Пристроями Петлі організовує кільце (рис. 2.33). По даній витій парі передається живлення для Пристроїв Петлі напругою 24 В постійного струму, яке модулюється манчестерським кодом, тим самим забезпечуючи передачу цифрових даних. В основному, застосовується Сегмент при *безпосередньому* підключенні датчиків та виконавчих механізмів у промислових умовах експлуатації (виконання IP-65 та IP-54).

Сегмент INTERBUS Петлі має такі характеристики: максимальна довжина Сегмента — 100 м; максимальна відстань між Пристроями — 10 м, максимальна кількість Пристроїв — 32, максимальний струм споживання — 1.5 А; середовище передачі — неекранований двопровідний кабель 2x1,5 мм² (16 AWG).

2.5.9.6. Канальний рівень. На фізичному рівні дані передаються символьним способом — послідовно, за кільцем, починаючи від Мережного Ведучого та ним

же закінчуючи. Виділяються два типи символів: статусна телеграма (5 статусних бітів) та телеграма даних (5 статусних бітів + 8 бітів даних). Символи є складовими кадрами INTERBUS.



U_L — живлення електроніки відгалужувача;
 U_S — живлення INTERBUS Петлі;
 U_A — локальне живлення виконавчих механізмів.

Рис. 2.33. Структура Сегмента INTERBUS Петлі

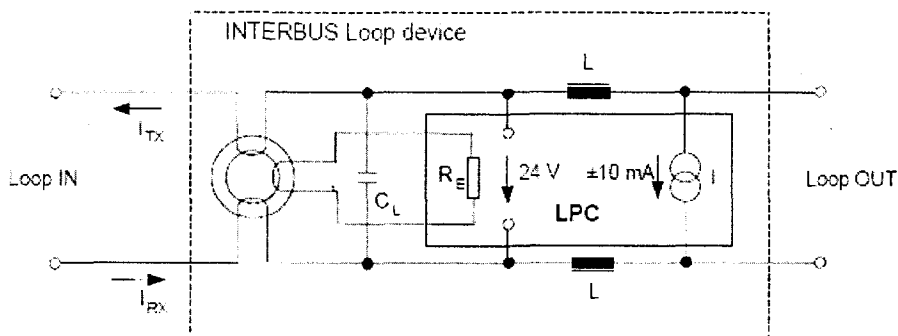


Рис. 2.34. Структура Пристрою INTERBUS Петлі

Ключовою особливістю реалізації мережі INTERBUS на каналному рівні є використання Summation-Frame протоколу з безколізійною процедурою доступу до середовища TDMA. Для обміну даними використовується Сумарний Кадр, який об'єднує входні та вихідні дані для всіх Пристроїв у мережі та відправляється їм однією послілкою. Це значно економить ресурси мережі, оскільки не потрібно витрачати їх додатково на роботу з окремим пристроєм. Тобто передачею одного кадру Ведучий запише дані в усі засоби виводу, а прийомом одного кадру отримає значення з усіх засобів вводу. Для запису входних даних, згідно з TDMA, кожному вузлу виділяється часовий інтервал (тайм-слот). Враховуючи, що Веду-

чий має інформацію про кількість вузлів у мережі, він визначає тривалість усього кадру як суми всіх тайм-слотів. Можуть бути виділені додаткові тайм-слоти, для передачі ациклічних повідомлень.

Враховуючи фізичну та логічну кільцеву топологію INTERBUS, Summation-Frame, процедура в ній реалізована через так звані *зсувні регістри*. Кожний Мережний Пристрій «приєднується» до логічного кільця зсувним регістром, довжина якого (кількість символів) залежить від кількості точок вводу/виводу даного Пристрою (від 4 до 64 бітів). Об'єднуючи всі Пристрої разом в логічне кільце, вони створюють структуру, яка, по суті, є Сумарним Кадром.

Вихідні дані процесу розміщуються у вихідному буфері Ведучого в порядку фізичного розміщення в кільці Пристроїв, для яких вони призначені (див. рис. 2.35, а). Цикл передачі (data cycle) починається з послідовної передачі даних, які починаються службовим 16-бітним *loopback* словом (LBW) та продовжуються вихідними даними буфера (зсувними регістрами). Передаючи цю послідовність, з першого Пристрою «виштовхуються» його вхідні дані, які, в свою чергу, приводять у рух все логічне кільце, що призводить до появи у вхідному буфері Ведучого вхідних даних від останнього Пристрою в кільці. Після «виштовхування» останніх даних з вихідного буфера Ведучого, вони попадають у перший Пристрій в кільці, для якого ці дані і були призначені. Всі інші вихідні дані теж містяться в Пристроях відповідно до їх призначення. В цей момент вхідний буфер Ведучого повністю заповнений вхідними даними Пристроїв (рис. 2.35, б).

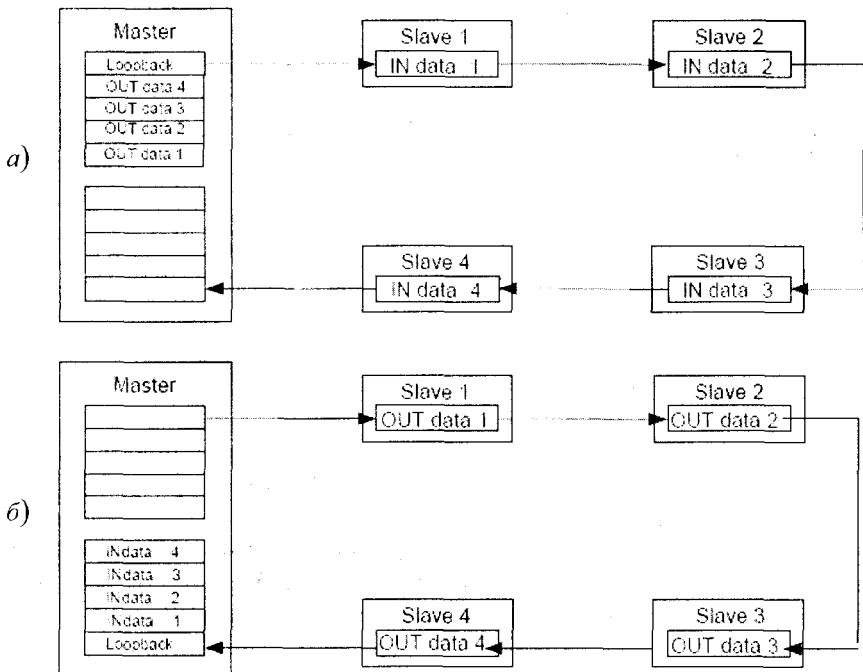


Рис. 2.35. Принцип передачі даних в INTERBUS: розподіл даних а — перед циклом передачі, б — після

На рис. 2.36 показано, яку послідовність зсувних реєстрів передає кожен з вузлів системи, якщо вона складається з 4-рьох пристроїв. Додатково до даних входів/виходів цей Сумарний Кадр вміщує також дані про контрольну суму (CRC). Контроль за правильністю отримання даних здійснюється між суміжними парами вузлів, що приводить до генерації контрольної суми в момент передачі поля CRC кожним вузлом. Врешті-решт про виявлення помилки контрольної суми дізнається Ведучий, отримавши кінцевий CRC (CRC4 на рис. 2.36).

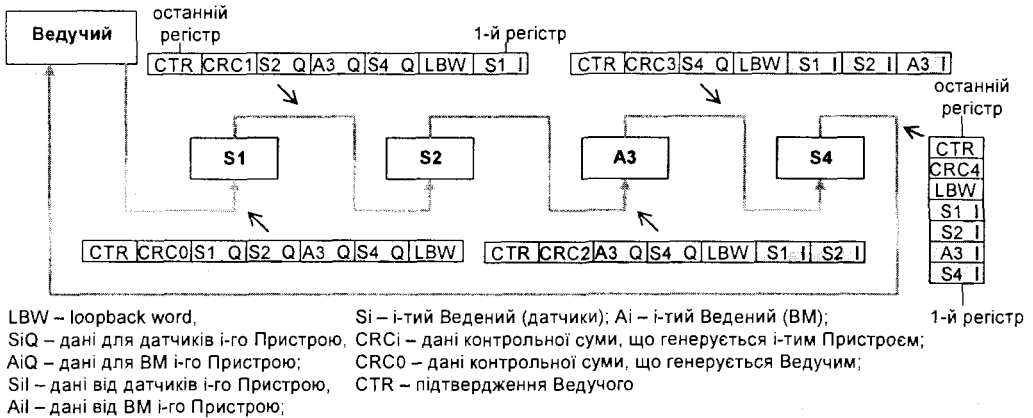


Рис. 2.36. Послідовність передачі даних від Пристрою до Пристрою

Наведена вище схема функціонування INTERBUS не потребує адресації Пристроїв, бо дані для них розміщуються відповідно до фізичного розташування в кільці. Крім того, вона ефективна з точки зору використання мережного трафіка (приблизно 60% корисних даних для 32 Пристроїв на 8 I/O).

2.5.9.6. Прикладний рівень. На прикладному рівні INTERBUS використовуються два різні типи сервісів:

- *process data channel (PDC)* — забезпечує циклічно-періодичним обміном даними процесу з Пристроями вводу/виводу (клієнт-серверна модель ідентифікованого обміну);
- *parameter channel (PC)* — забезпечує обмін параметричними даними двох Пристроїв між собою на основі Клієнт-Серверної моделі (клієнт серверна модель обміну повідомленнями).

Доступ з програми користувача до даних Пристроїв через сервіс PDC відбувається через образ вводу/виводу, який циклічно-періодично оновлюється, як це було описано вище. Послідовність розміщення даних в образі співпадає з фізичною послідовністю Пристроїв у кільці.

Параметричний канал PC використовується для обміну комплексними даними між двома Пристроями, для використання функцій конфігурування, програмування, відображення даних, управління тощо. Для передачі параметричних даних використовується ациклічний обмін повідомленнями на базі моделі Клієнт-Сервер. Для цього для Пристроїв, які беруть участь в обміні додатково виділя-

ються тайм-слоти. Загальна картина обміну параметричними даними показана на рис. 2.37.

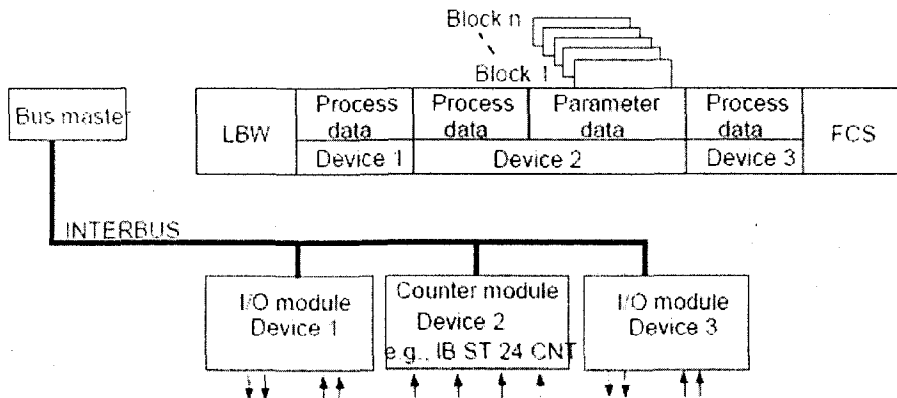


Рис. 2.37. Використання параметричного каналу РС

Таблиця 2.19

ХАРАКТЕРИСТИКИ МЕРЕЖІ

OSI	Характеристика	INTERBUS			
	<i>NetArea</i>	рівень датчиків			
прикладний	<i>AppService</i>	PDC — циклічно-періодичний обмін Ведучого вузла даними вводу/виводу; PC — ациклічний обмін параметричними даними між двома вузлами; діагностичні сервіси;			
	<i>AppModel</i>	клієнт-серверна модель ідентифікованого обміну для PDC;* клієнт-серверна модель обміну повідомленнями для PC;			
	<i>AppProfile</i>	профілі в застосуванні; великий перелік профілів пристроїв: PDS(DRIVECOM), LMI (MMICOM), енкодери, контролери, датчики/BM			
	<i>AppProcData</i>	PDC: всього на мережу до 2048 вх.слів, 2048 вих.слів;			
	<i>AppResolut</i>				
каналний	<i>ChAddModel</i>	Сумарний Кадр + регістровий зсув, орієнтований на фізичне розміщення Пристроїв у кільці;			
	<i>ChAccess</i>	Ведучий-Ведений + TDMA: Ведучий ініціює обмін, Ведені додають свої дані в певний момент часу			
	<i>ChChecksum</i>	CRC, змінюється та контролюється від вузла до вузла			
фізичний	<i>PhBaudRate</i>	500 кБіт/с			
	<i>PhSegmen.</i>	до 16 рівнів сегментації			
	<i>назва сегм</i>	Remote Bus	Installation Remote Bus	Local Bus (I/O Bus)	INTERBUS Loop
	<i>PhInterface</i>	RS-485	RS-485	TTL	модуляція несучої 24 В, манчест. код

OSI	Характеристика	INTERBUS			
фізичний	<i>PhMedia</i>	екранований кабель на 3 виті пари**	екранований кабель на 3 виті пари + 3 проводи живлення	5-провідний плоский кабель або з'єднувач**	неекранований 2-провідний кабель
	<i>PhTopology</i>	фізичне кільце, видима за кабелем — лінійна, організовується дуплексним зв'язком між суміжними засобами, кінцевий Пристрій замикає кільце			кільце
	<i>PhNodes</i>	всього в мережі 512			
		256	обмеження за струмом живлення, Макс. 256	63 (8 для ST)	32
	<i>PhLength</i>	сумарна 13 км			
		12.8 км, між пристроями 400 м	50 м, між пристроями 50 м	10 м, між пристроями 1.5 м	100 м, між пристроями 10 м
<i>PhSupply</i>	ні	24 В по виділенним проводам кабелю	24 В по виділенним проводам	24 В по інформаційним проводам	

* — умовна модель, в офіційних документах INTERBUS модель не має назви

** — можливе оптоволокну

2.5.10. Мережі CIP: *DeviceNet*, *ControlNet*, *Ethernet/IP* та *CompoNet*

2.5.10.1. Походження. В 1994 році компанією Allen-Bradley був розроблений та опублікований CAN-сумісний протокол мережі *DeviceNet*. Пізніше була заснована організація для підтримки та розвитку даного протоколу — **ODVA** (Open DeviceNet Vendor Organization, <http://www.odva.org>). У 1997 році той самий протокол прикладного рівня, що і в *DeviceNet*, був доповнений новими протоколами каналного та фізичного рівнів, які дозволяли обмінюватися на більш високих швидкостях із забезпеченням повної детермінованості. Мережа отримала назву *ControlNet*, яка сьогодні підтримується організацією **CI** (*ControlNet International*, <http://www.controlnet.org>). У 2000 році ODVA та CI адаптувала цей же протокол прикладного рівня до Ethernet TCP/IP під назвою **Ethernet/IP** («IP» — Industrial Protocol).

Таким чином, нині мережі *DeviceNet*, *ControlNet*, *Ethernet/IP* та *CompoNET* (мережа рівня датчиків, розроблена ODVA в 2006 році) використовують спільний протокол верхнього рівня (над мережним), який отримав назву **CIP** (Common Industrial Protocol). Для взаємної сумісності мереж CIP організації ODVA та CI розглядають їх у контексті єдиної структури (рис.2.38). Мережі CIP стандартизовані у стандартах MEK.

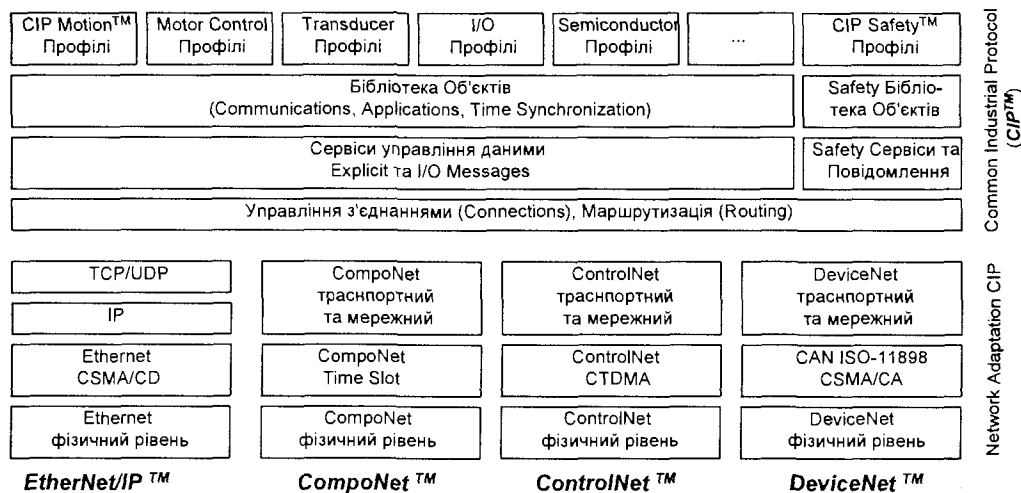


Рис. 2.38. Мережі CIP у контексті моделі OSI

Загальна структура опису CIP-мереж має декілька томів, з яких в 1-му томі описаний протокол CIP, а в інших — адаптація конкретної мережі до даного протоколу.

2.5.10.2. Протокол прикладного рівня CIP

Для опису доступних комунікаційних сервісів, поведінки вузла CIP, взаємодії між вузлами CIP використовується абстрактна об'єктна модель. Модель вузла CIP являє собою колекцію об'єктів, які надають доступ до певних компонентів вузла. Структурування об'єктів проходить через визначення класу, екземпляра та атрибута, які використовуються в класичному об'єктно-орієнтованому підході. На основі класу об'єкта (class), що являє собою опис об'єкта (об'єктний тип), створюються екземпляри об'єктів (instances), які характеризуються набором атрибутів (attribute), що визначені даним класом. Через атрибути екземпляр надає доступ до власних даних, за винятком атрибутів класу, через які надається доступ до даних класу або всіх екземплярів класу.

Для доступу до об'єктів використовується універсальна система адресації, яка включає (див. рис. 2.39):

- **Адресу вузла (Node Address)**: ціле число, призначене для кожного вузла мережі CIP; для DeviceNet, ControlNet та CompoNet називається також MAC ID і є номером пристрою в мережі; для EtherNet/IP це IP адреса.

- **Ідентифікатор класу (Class ID)**: ціле число, яке призначене кожному класу;

- **Ідентифікатор екземпляра (Instance ID)**: ціле число, яким ідентифікується екземпляр серед усіх екземплярів того ж класу;

- **Ідентифікатор атрибута (Attribute ID)**: ціле число, номер атрибута в опису класу;

- **Код сервісу Service Code (для атрибутів класу)**: ціле число, яке визначає номер служби, тобто дію, яку необхідно зробити через явні повідомлення для класу або для екземпляра.

Таким чином, при доступі до об'єкта шляхом обміну повідомленнями, вказується послідовність: «вузол: клас: екземпляр: атрибут: сервіс», в якій тип дії вказується в полі сервіс.

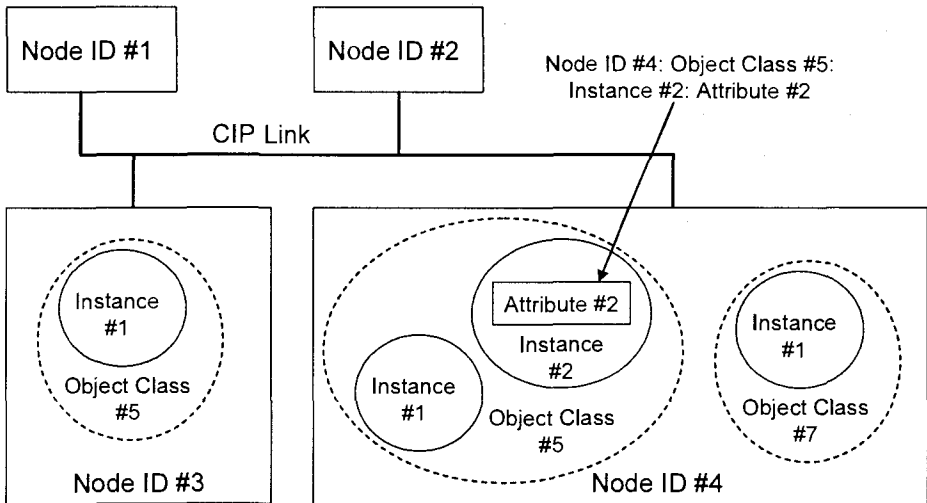


Рис. 2.39. Адресація об'єктів в вузлі CIP

Зв'язок даних між прикладними Процесами функціонує на основі явних (*Explicit Message*) та неявних (*Implicit Message* або *I/O Message*) повідомлень. Ці повідомлення можуть бути реалізовані через з'єднання (connections) між CIP об'єктами (між їх атрибутами). Кожне з'єднання спрямоване, тому для реалізації двохстороннього обміну між об'єктами (наприклад, читання/запис) необхідно використати два з'єднання. З'єднання нумеруються ідентифікатором з'єднання *CID*, формат яких залежить від типу мережі. Для DeviceNet, наприклад, *CID* буде вказувати на ідентифікатор CAN.

Крім функцій, орієнтованих на з'єднання, CIP підтримує функції обміну явними повідомленнями, що не потребують з'єднань, — *Unconnected Explicit Messaging* (менеджера *UCMM* — *Unconnected Message Manager*). Крім того, вони використовуються для побудови самих з'єднань. Зокрема, функція *Forward Open* забезпечує створення CIP з'єднань між об'єктами. При цьому вказується: адреса атрибута вузла, періодичність відновлення даних (періодично, за зміною, за запитом), обмеження на обсяг даних, *CID*, тайм-аут відсутності передачі даних, інформація про відправника.

CIP-мережі на прикладному рівні надають два типи сервісів, що орієнтовані на з'єднання:

- *I/O Connections* — для обміну даними процесу через відправку ідентифікованих даних;
- *Explicit Messaging Connections* — для обміну даними процесу та параметричними даними через відправку повідомлень (тип запит-відповідь).

Ці типи сервісів реалізуються через відповідні Комунікаційні Об'єкти (рис. 2.40). Кожний з них представляє сторону Виробника (Producer) або Споживача (Consumer) чи обидві. Для I/O Connections, ідентифіковані дані передаються від Виробника до Споживача. При двосторонньому з'єднанні даний тип об'єкта представляє обидві сторони. Для Explicit messaging connections (типу запит-відповідь) завжди підтримується двосторонній зв'язок, бо на формування запиту очікується відповідь. Тому даний Комунікаційний Об'єкт повинен підтримувати як бік Виробника, так і бік Споживача. Об'єкт-запитувач (Клієнт) передає дані разом з кодом сервісу, який визначає номер необхідної служби. Відповідно до вказаного номера, ці дані будуть оброблятися об'єктом-відповідачем (Сервером), функції якого накладені на спеціальний об'єкт *Message Router*. Той, у свою чергу, обробивши запит, сформує повідомлення-відповідь аналогічним чином, повторивши код сервісу. Об'єкт Message Router є Сервером для всіх явних повідомлень, тобто і для сервісів менеджера UCMM.

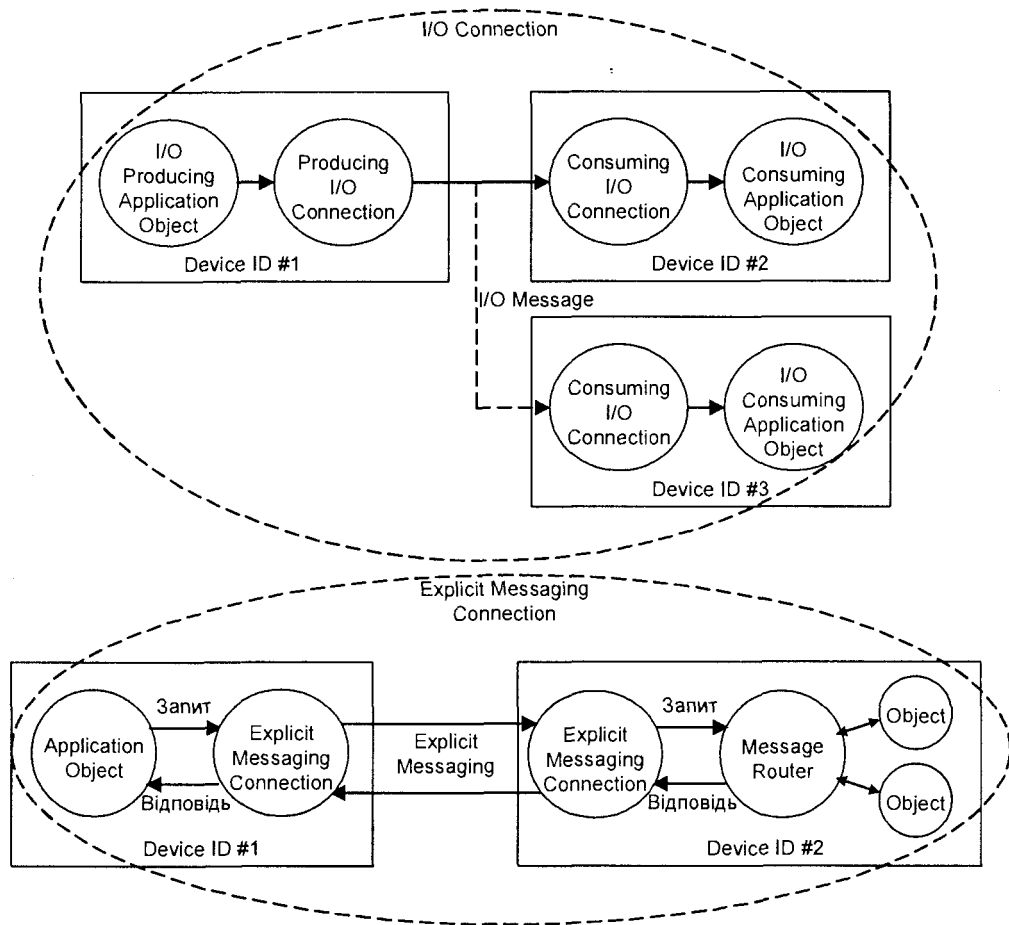


Рис. 2.40. Комунікаційні об'єкти CIP

Слід зазначити, що також доступний обмін даними між вузлами, що містяться в різних мережах CIP, і для сервісів, що потребують з'єднання (I/O Connections та Explicit messaging connections) і для Unconnected Explicit Messaging. Це забезпечується на прикладному, а не на мережному рівні.

2.5.10.3. Бібліотека об'єктів CIP

Об'єкти бібліотеки CIP можна поділити на три типи:

1. General-use (загального призначення), які можуть бути присутні у всіх типах вузлів; це такі об'єкти: Assembly, Acknowledge Handler, Connection, Connection Configuration, Connection Manager, File, Identity, Message Router, Parameter, Parameter Group, Port, Register, Selection.

2. Application-specific (прикладного призначення), їх наявність у вузлах визначається типом вузла; наприклад: AC/DC Drive, Analog Input Group, Analog Output Group, Discrete Input Group, Discrete Output Group, Motor Data та інші.

3. Network-specific (залежні від мережі), їх наявність у вузлах визначається типом мережі CIP, до яких вони підключені; це такі об'єкти: ControlNet, ControlNet Keeper, ControlNet Scheduling, DeviceNet, Ethernet Link, TCP/IP Interface, CompoNet Link.

На рис. 2.41 показана типова об'єктна модель пристрою CIP, який обов'язково повинен включати такі об'єкти:

- будь-який Connection або Connection Manager об'єкт;
 - об'єкт Identity;
 - один або декілька об'єктів Network-specific;
 - об'єкт Message Router;
- та опціонально об'єкти:
- Parameter;
 - Assembly;
 - один або декілька об'єктів типу Application-specific.

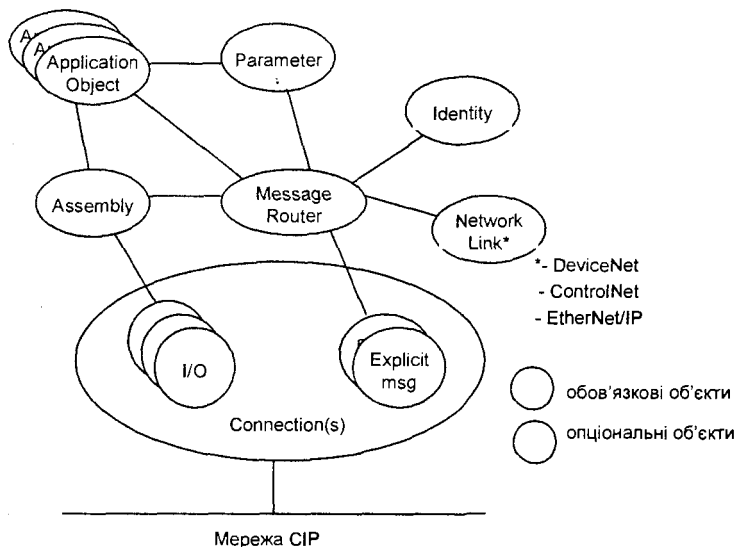


Рис. 2.41. Типова об'єктна модель CIP пристрою

Призначення комунікаційних Об'єктів Connection, а також об'єкта Message Router ми розглянули вище.

Об'єкт класу *Identity* представлений у пристрої тільки одним екземпляром і призначений для ідентифікації пристрою та його стану. Для цього об'єкта доступні такі атрибути: Vendor ID (ідентифікатор виробника), Device Type (номер підтримуваного профілю пристроєм), Product Code (ідентифікатор пристрою), Revision (версія), Status (плинний статус пристрою), Serial Number (серійний номер), Product Name (назва пристрою).

Об'єкт *Assembly* призначений для відображення (mapping) декількох атрибутів від різних екземплярів в одну структуру, що прискорює їх передачу за допомогою I/O Connections. Розрізняють два типи об'єктів Assembly:

- Input Assembly, які збирають дані від атрибутів різних екземплярів та передають їх Producing I/O Connection, для передачі в мережу;
- Output Assembly, які отримують дані від Consuming I/O Connection, та записують їх значення в атрибути екземплярів, відповідно до схеми відображення.

Таким чином, об'єкт Assembly дозволяє передавати дані декількох атрибутів від різних екземплярів, використовуючи мінімум об'єктів I/O Connection, що значно оптимізує мережний трафік та зменшує час транзакцій.

Для полегшення інтеграції вузлів від різних виробників для CIP визначені профілі пристроїв та розроблений єдиний формат їх опису у вигляді EDS-файла.

2.5.10.4. Фізичний рівень DeviceNET

Фізичний та каналний рівень DeviceNET базується на CAN-стандарті ISO11898, з певними доповненнями. На фізичному рівні DeviceNET має такі характеристики: шинна топологія з можливістю відгалужень із 2-ма термінаторами 121 Ом у кінцях магістрального кабелю; 64 вузли на сегменті; використовується спеціальний 5-жильний кабель (2 сигнальні лінії, 2 живлення, 1 екран (заземляється в одному місці)); бітова швидкість 125 кБіт/с (500 м), 250 кБіт/с (250 м) та 500 кБіт/с (100 м); можливість живлення датчиків по мережі та підключення декількох роздільних джерел живлень до однієї мережі (24 В, <16 А); можливість гарячої заміни модулів; підтримка різних типів конекторів; захист від неправильного підключення; вбудований захист від перевантажень.

2.5.10.5. Канальний рівень DeviceNET. Відповідно до специфікацій CIP, для адресації вузлів використовується Node Address, який у DeviceNET також називається *MAC-ID*. Враховуючи, що на каналному рівні використовується CAN-система адресації повідомлень, 6-бітна адреса MAC-ID (0-63) є частиною CAN-ідентифікатора. Інші біти CAN-ID використовуються для кодування ідентифікатора повідомлення. Для з'єднань DeviceNET CAN-ID співпадає з Connection ID (CID). Для виділення пріоритетності повідомлень в єдиній мережі їх згрупували в 4-ри групи (Message Group), правила формування CAN-ID для яких вказані на рис. 2.42. MAC-ID для вузла може виставлятися перемикачами або змінюватись через Об'єкт DeviceNET.

2.5.10.6. Верхні рівні DeviceNET. Для адаптації DeviceNET на прикладному рівні до протоколу CIP доданий новий об'єкт «DeviceNET», а також введені деякі додаткові механізми:

- невеликі повідомлення (I/O Connections) обмежуються 8 байтами (відповідно до обмеження CAN Кадру Даних), а для великих (усіх Explicit Messaging та деяких I/O Connections) використовується механізм фрагментації;
- введений комунікаційний профіль типу Ведучий-Ведений для мінімізації комунікаційних витрат на управління мережею.

Connection ID = Ідентифікатору CAN (біти 10:0)											Використовується для
10	9	8	7	6	5	4	3	2	1	0	
0	Message ID				MAC ID відправника						Message Group 1
1	0	MAC ID					Message ID				Message Group 2
1	1	Message ID			MAC ID відправника						Message Group 3
1	1	1	1	1	Message ID						Message Group 4
1	1	1	1	1	1	1	X	x	x	x	Невірний Ідентифікатор CAN

Рис. 2.42. Формування CAN-ID залежно від Message Group

Ці адаптації дали змогу реалізувати протокол DeviceNET в єдиній недорогій мікросхемі, тобто мінімізувати вартість пристроїв, що дуже принципово для мереж рівня датчиків.

У вузлі для кожного порту DeviceNET виділяється окремий екземпляр Об'єкта «DeviceNET», в якому зберігається бітова швидкість та MAC-ID-адреса.

Нагадуємо, що обмін даними в мережах CIP проходить через з'єднання (див. рис. 2.41), які необхідно встановити. Встановлення з'єднання можна провести через сервіси Unconnected Explicit Messaging (див. 2.5.10.2). Однак ця процедура дуже складна і потребує великих затрат на реалізацію з боку пристрою. Для спрощення створення з'єднань у DeviceNET використовується механізм наперед визначених з'єднань (*Predefined Master-Slave Connection Set*), який базується на таких принципах.

Усі вузли в мережі поділяються на Ведучі (контролери) та Ведені (засоби вводу-виводу). *Слід зазначити, що дана схема не передбачає контроль доступу до шини типу Ведучий-Ведений.* Ведучих у мережі DeviceNET може бути декілька, однак Ведені логічно прив'язані тільки до одного Ведучого. Для цього в кожному Вузлі закладений наперед визначений набір доступних CAN-ID з «Message Group 1» та «Message Group 2» (див. рис. 2.43). Враховуючи, що частина CAN-ID є адресою призначення MAC-ID, яка повинна вказувати на Ведучий вузол, Веденому необхідно цю адресу сповістити. В схемі «Predefined Master-Slave Connection Set» адреса Ведучого повідомляється кожному Веденому через спеціальний зарезервований для цього MAC-ID (порт Group 2 Only Unconnected Request), через який поступають до Об'єкту «DeviceNET». Разом з повідомленням своєї адреси, Ведучий також вказує які з 5-ти наперед визначених з'єднань необхідно активувати. Ці з'єднання забезпечують зв'язок між такими Комунікаційними Об'єктами Ведучого та Веденого:

- 1) Об'єкти Explicit Messaging Connection:
 - Group 2 Explicit Request/Response Message (Екземпляр ID 1)
- 2) Об'єкти I/O Messaging Connections:
 - Polled I/O Connection (Екземпляр ID 2): Ведучий відправляє Веденому вхідні дані (poll Comand), після чого отримує від нього вхідні дані (poll Responce);
 - Bit-Strobe I/O Connection (Екземпляр ID 3): Ведучий відправляє 8-байтну команду (Bit-Strobe command), кожний біт якої вказує на номер Веденого, який відправляє у відповідь вхідні дані (Bit-Strobe Responce);
 - Change of State or Cyclic I/O Connection (Екземпляр ID 4): дані відправляються циклічно (Cyclic) або за зміною (COS) від Ведучого до Веденого або навпаки, залежно від настройки;
 - Multicast Polling I/O Connection (Екземпляр ID 5): подібно до Polled, однак дані адресуються декільком вузлам.

Connection ID = Ідентифікатору CAN (біти 10:0)										Використовується для	
10	9	8	7	6	5	4	3	2	1		0
0	Group 1 Message ID				MAC ID відправника						Group 1 Message
0	1	1	0	0	MAC ID відправника						повідомлення-відповідь Ведених на мультиадресне опитування Poll
0	1	1	0	1	MAC ID відправника						повідомлення Веденого I/O COS або Cyclic
0	1	1	1	0	MAC ID відправника						повідомлення-відповідь Веденого на Bit-Strobe команду
0	1	1	1	1	MAC ID відправника						повідомлення-відповідь Веденого на I/O Poll або підтвердж COS/Cyclic
1	0	MAC ID			Group 2 Message ID						Group 2 Message
1	0	MAC ID відправника			0	0	0				повідомлення-команда Bit-Strobe від Ведучого
1	0	MAC ID відправника			0	0	1				мультиадресне опитування Poll від Ведучого
1	0	MAC ID отримувача			0	1	0				підтвердження Ведучого на COS/Cyclic від Веденого
1	0	MAC ID відправника			0	1	1				Explicit/Unconnected повідомлення-відповідь від Веденого
1	0	MAC ID отримувача			1	0	0				Explicit повідомлення-запит від Ведучого
1	0	MAC ID отримувача			1	0	1				I/O Poll команда або COS/Cyclic повідомлення від Ведучого
1	0	MAC ID отримувача			1	1	0				Group2 Only Unconnected Explicit повідомлення-запит
1	0	MAC ID отримувача			1	1	1				повідомлення Duplicate MAC ID Check

Рис. 2.43. CID для Predefined Master-Slave Connection Set

Ведені вузли, які підтримують тільки механізм Predefined Master-Slave Connection Set, називають **Group 2 Only Servers**, а UCMM-сумісні вузли, які підтримують Unconnected Explicit Messaging, називаються **Group 2 Servers**. Перший тип вузлів може обмінюватися даними тільки з одним Ведучим, тоді як другий тип вузлів підтримує динамічні з'єднання, тобто з'єднується із будь-якими вузлами в інтермережі CAN. Таким чином, у DeviceNET можна виділити три класи пристроїв: Ведучі, Ведені Group 2 Only Servers та Ведені Group 2 Servers.

2.5.10.7. Фізичний рівень ControlNET

На фізичному рівні ControlNET використовується спеціально розроблений для даної мережі інтерфейс, що має такі характеристики: шинна топологія на базі 75-омного коаксіального кабелю (типово RG-6) з двома термінаторами 75 Ом в обох кінцях магістралі; підключення пристроїв через спеціальні коробки з відгалуженнями (1 м), BNC-конектори або TNC-конектори (IP67); максимальна довжина магістрального кабелю 1000 м при 2-х відгалуженнях, 250 — при 48; максимальна кількість вузлів — 99; можлива сегментація через репітери; бітова швидкість — 5 МБіт/с; синхронна передача напругою Манчестерським кодом.

2.5.10.8. Канальний рівень ControlNET. На каналному рівні використовується повністю детермінований метод доступу *CTDMA* (Concurrent Time Domain Multiple Access) з адресацією вузлів через MAC-ID, який базується на таких принципах. Весь мережний час ділиться на однакові проміжки, які називаються *NUT* (Network Update Time). Кожний NUT, у свою чергу, ділиться на Час Запланованих Сервісів (Scheduled Service Time), Час Незапланованих Сервісів (Unscheduled Service Time) та Захисний Інтервал (Guardband Time).

У межах Часу Запланованих Повідомлень кожен вузол у мережі, починаючи від 1-го до SMAX (максимальний номер вузла, що бере участь у передачі Запланованих Повідомлень) по черзі передають свої заплановані кадри (рис. 2.44). Якщо вузлу нічого передавати, він передає короткий кадр, що свідчить про його працездатність. Якщо вузол, що має передавати протягом тайм-ауту, нічого не передає, він вважається дефектним, і за ним починає передавати вузол з наступним номером.

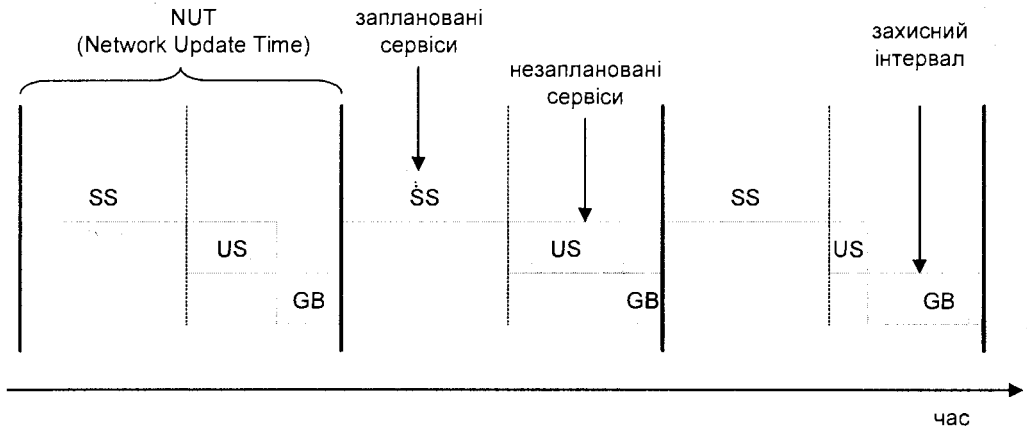


Рис. 2.44. Метод доступу до шини CTDMA в ControlNET

Після передачі запланованих кадрів усіма вузлами, настає Час Незапланованих Повідомлень, який триває протягом NUT-GuardbandTime. В цей час гарантується відправка свого незапланованого кадру тільки одному вузлу, з номером, для якого надійшла черга (наприклад n). Якщо залишається додатковий час, незапланований кадр може відправити вузол з наступною адресою ($n + 1$) і т.д. У будь-якому випадку, на наступний NUT незапланований кадр буде відправляти вузол з номером $n + 1$. Коли черга незапланованих повідомлень дійде до вузла з

номером UMAX (максимальний номер вузла, що бере участь у передачі Незапланованих Повідомлень), на наступний NUT черга передається до 1-го вузла.

Як видно, модель STDMA гарантує рівноцінний розподіл ресурсів між усіма вузлами в мережі, що особливо важливо для мереж рівня контролерів.

Кадри ControlNET (MAC-кадри) включають 8-бітну адресу відправника та декілька *Lpacket* (LinkPacket), які обмежені 510 байтами (рис. 2.45). Кожен Lpacket являє собою індивідуальне СІР-повідомлення (I/O Message або Explicit Message) або спеціальну службову інформацію для управління мережею.

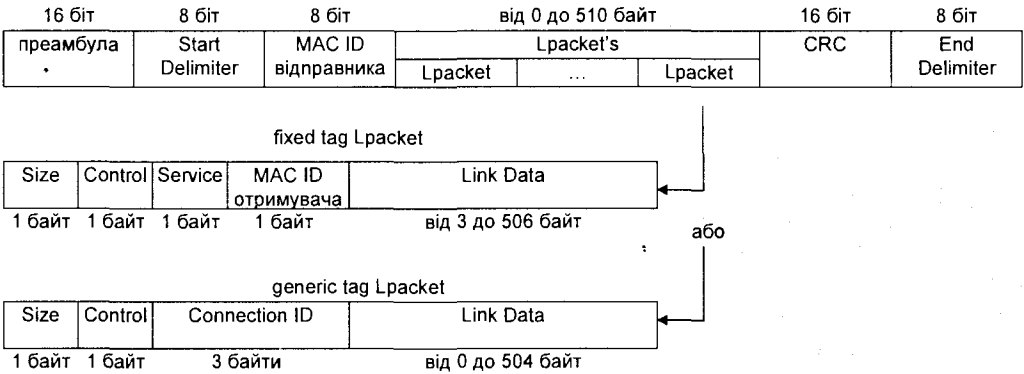


Рис. 2.45. Структура MAC-кадру

Визначені два формати Lpacket (рис. 2.45): *fixed tag Lpacket* — які використовуються для передачі Unconnected Messaging, та *generic tag Lpacket*, — які використовуються для обміну даними через з'єднання.

2.5.10.9. Верхні рівні ControlNET

Для адаптації ControlNET на прикладному рівні до протоколу СІР добавлені такі об'єкти: «ControlNet» Object — на кожний порт свій екземпляр, вміщує MAC-ID, стан пристрою, діагностичний лічильник, часові параметри каналного рівня; Keeper Object — об'єкт який зберігає всю інформацію про заплановані з'єднання, з'єднання створюються вузлами, які називаються Connection Originator, Keeper відправляє всю необхідну інформацію Connection Originator; Scheduling Object — об'єкт, який розміщується на вузлах типу Connection Originator, через нього конфігураційна утиліта може доступитися до всіх запланованих з'єднань.

Обмін повідомленнями (Explicit Message) може проводитись як через Explicit Messaging Connections, так і через Unconnected Messaging (функції UCMM). Перші проводяться, використовуючи Lpacket формату generic tag, другі — fixed tag. Обидва сервіси використовують Час Незапланованих Повідомлень. Функціями UCMM послуговуються для побудови з'єднань вузлами Connection Originator.

Обмін даними через I/O Connections проходить передачею повідомлень MAC-кадром з Lpacket формату generic tag. Один MAC-кадр може вміщувати дані від декількох об'єктів I/O Connections і завжди передається в Час Запланованих Повідомлень. Таким чином, усі вузли на мережі потенційно можуть бути як Connection Consumer.

В ControlNet виділені 4-ри класи пристроїв:

– **Messaging Server**, які підтримують тільки Connected та Unconnected Explicit Messages для завантаження/вивантаження програм, даних, відображення статусу та діагностики і т.д.;

– **I/O Server**, — які підтримують функції Messaging Server та серверні функції I/O Messaging (пристрої розподіленої периферії);

– **Messaging Client**, — які підтримують функції Messaging Server, а також функції Connection Originator (комп'ютер з комунікаційною картою, ЛМІ);

– **I/O Scanner** — підтримує всі функції;

2.5.10.10. Нижні рівні EtherNET/IP.

Фізичний та каналний рівні EtherNET/IP базуються на Ethernet, мережний та транспортний — на стеку TCP/IP. На фізичному та каналному рівнях EtherNET/IP ставляться аналогічні вимоги як і до інших промислових мереж, що базуються на Ethernet (див. розділ 10). Зокрема, згідно з концепцією MICE, для промислових умов експлуатації рекомендується використовувати екрановану виту пару, промислові роз'єми типу «M12-4 з D-кодуванням» або модифікований герметичний RJ-45.

Залежно від області використання EtherNET/IP, до інфраструктури Ethernet пред'являються різні вимоги. Можливі 4-ри різні типи сценаріїв EtherNET/IP:

- 1) EtherNET/IP з одним контролером ізолювана від мережі підприємства;
- 2) EtherNET/IP з декількома контролерами ізолювана від мережі підприємства;
- 3) EtherNET/IP підключена до корпоративної мережі через маршрутизатор (або еквівалентно йому);
- 4) EtherNET/IP інтегрована в корпоративну мережу.

Залежно від цих типів висуваються загальні рекомендації на використання комутаторів та маршрутизаторів з різними функціональними можливостями — для оптимізації мережного трафіку та досягнення необхідних вимог (див. табл. 2.20).

Таблиця 2.20

	Ізолювана з одним контролером	Ізолювана з декількома контролерами	Підключена до корпоративної мережі	Інтегрована в корпоративну мережу
використання комутаторів	обов'язкове	обов'язкове	обов'язкове	обов'язкове
підтримка Auto-MDI/MDX та Auto-negotiation;	рекоменд.	рекоменд.	Рекоменд.	Рекоменд.
промислове виконання комутаторів	рекоменд.	рекоменд.	рекоменд.	рекоменд.
підтримка комутаторами IGMP snooping	—	обов'язкове	обов'язкове	обов'язкове
підтримка комутаторами WEB/SNMP діагностики		рекоменд.	обов'язкове	обов'язкове
підтримка VLAN	—	рекоменд.	рекоменд.	обов'язкове
використання шлюзів	—	—	обов'язкове	—
використання маршрутизаторів та комутаторів 3-го рівня	—	—	Рекоменд.	обов'язкове

Адресація СІР вузлів в EtherNET/IP проводиться через IP-адресацію, яка може встановлюватися різними методами, залежно від пристрою та виробника: DHCP, BOOTP, явне повідомлення Set Attribute, перемикачами і т.д.

Для обміну даними між прикладними Процесами використовуються як TCP, так і UDP-протокол. Усі вузли EtherNET/IP підтримують принаймні два TCP-з'єднання через порт AF12₁₆. Це з'єднання використовується для Explicit Messaging Connections та Unconnected Messaging (функції UCMM). Деякі з команд цих типів сервісів можуть послуговуватись аналогічним UDP-портом. Однак, в основному, UDP використовується для I/O Connections (порт 08AE₁₆), при якому дані передаються в багатоадресному режимі (декілька адресатів IP).

2.5.10.11. Верхні рівні EtherNET/IP

Для адаптації ControlNET на прикладному рівні до протоколу СІР добавлені такі об'єкти: «TCP/IP»-Object, який призначений для конфігурування відповідних сервісів мережного та транспортного рівнів; Ethernet Link Object, який забезпечує доступом до статусної інформації інтерфейсу (кількість екземплярів визначається кількістю портів Ethernet).

Дані, які інкапсулюються в TCP/UDP-TPDU, в EtherNET/IP складаються із заголовка та даних CPF (Common Packet Format). Заголовок необхідний тільки для функціонування сервісів Explicit Messaging Connections та Unconnected Messaging (функції UCMM).

Для I/O Connections використовуються тільки дані CPF. Вони передаються в багатоадресному режимі UDP-TPDU безпосередньо за його заголовком. Нагадаємо, що з'єднання в мережах СІР між двома Комунікаційними Об'єктами визначається спеціальним ідентифікатором ConnectionID. Однак, на відміну від DeviceNET та ControlNET, каналний та мережний рівні не може відфільтрувати I/O Connections повідомлення за їх ідентифікатором. Це значить, що всі UDP-TPDU, які реалізують даний тип з'єднання, повинні розсилатися IP-датаграмою в широкомовному режимі. При великій кількості вузлів та багаторівневій ієрархії мережі це неприпустимо, оскільки дуже завантажує трафік I/O-повідомленнями. Тому в таких випадках рекомендується використовувати групову адресацію за протоколом IGMP і відповідно комутатори, що підтримують функцію IGMP snooping (див. розділ 10). Для розділення мережного трафіка та виділення пріоритетності реал-тайму повідомлень (I/O-повідомлень) від нереал-тайму також рекомендується використовувати VLAN (див. розділ 10). Слід зазначити, що IGMP та VLAN це стандартні механізми, визначені для мереж Ethernet TCP/IP.

В Ethernet/IP виділені 4-ри класи пристроїв аналогічно ControlNet.

2.5.10.12. Фізичний рівень ComproNET

Фізичний рівень ComproNET розроблений шляхом модифікації стандарту RS-485, а саме використанням манчестерського кодування, імпульсних трансформаторів між трансивером та середовищем передачі, фільтрації вхідних сигналів за спеціальною маскою. Це дозволяє збільшити величину сигналу, покращити завадостійкість, що дає можливість використовувати в якості середовища передачі неекрановані кабелі та збільшити максимальну довжину ліній зв'язку.

Використовується шинна топологія на базі звичайного 2-х провідного неекранованого кабелю або 4-х провідного, за необхідності живлення пристроїв. Через спеціальні Т-подібні відгалужувачі допускаються відгалуження. Бітова швидкість вибирається з ряду: 4 Мбіт/с (відгалуження не допускається), 3 Мбіт/с, 1.5 Мбіт/с, 93.75 Кбіт/с. Максимальна кількість пристроїв на сегменті — 32, довжина сегмента — 500 м (при 93.75 Кбіт/с). Ці параметри можуть бути збільшені при використанні репітерів, які також дають можливість будувати деревовидну топологію. Кількість репітерів може досягати 64, однак між центральним Ведучим вузлом і найвіддаленішим Веденим їх має бути не більше 2-х, що обмежує максимальну довжину лінії зв'язку 1500 м.

2.5.10.13. Канальний рівень CompoNET. CompoNET на каналному рівні виділяється серед інших промислових мереж тим, що в ньому відсутні проміжки тиші. Кадри розділяються преамбулою, за рахунок якої досягається синхронізація пристроїв та їх трансиверів. Для доступу до шини використовується метод TDMA. Управління кадрами займається центральний вузол (контролер) з правом Ведучого. Ведучий вміщує конфігурацію мережі, включаючи інформацію про наявні репітери, що дозволяє йому визначити часовий інтервал (тайм-слот) для передачі кожним Веденим своїх вхідних даних та статусної інформації. Слід зазначити, що при визначенні тайм-слоту для Веденого Ведучий враховує кількість репітерів між ним та Веденим та тип пристрою. Початком відліку свого тайм-слоту для отримання вихідних даних Веденого є відправка Ведучим кадру типу OUT. Після закінчення даного кадру починається відлік тайм-слоту для відправки Веденим у мережу своїх вхідних даних.

Виділяються три типи пристроїв: аналоговий вузол (word node), дискретний вузол (bit node) та репітер. Усі вузли мають унікальну адресу MAC-ID (0–127-дискретні, 0–63-аналогові) Пристрої різних типів можуть мати однакові адреси. Ведучий CompoNET періодично отримує статусну інформацію про кожного Веденого і забезпечує гарячу заміну пристроїв.

2.5.10.14. Верхні рівні CompoNET. Ведені на мережі можуть бути в двох станах: учасник (participated state) або неучасник (non-participated state) обміну. Учасник обміну має виділений тайм-слот для відправки вхідних даних та отримання вихідних даних. Неучасником обміну може бути вузол, який тільки-но з'явився в мережі. Транспортний рівень CompoNET забезпечує зв'язок та конфігурування даного вузла за допомогою спеціального типу кадру, використовуючи сервіси UCMM. Після конфігурування вузла (отримавши тайм-слот), він переходить у режим учасника обміну. Обмін конфігураційними даними через обмін Явними Повідомленнями потребує їх фрагментацію та дефрагментацію, що теж забезпечується транспортним рівнем.

Ведучий та Ведений у режимі учасника обміну передає свої вхідні та вихідні дані через проміжний буфер (I/O інтерфейс). Цей буфер автоматично оновлюється засобами каналного рівня і є інтерфейсом для з'єднань типу I/O Connections. З'єднання Explicit Messaging Connections у CompoNET недоступні, тому для обміну повідомленнями використовуються тільки сервіси UCMM.

Таблиця 2.21

ХАРАКТЕРИСТИКИ МЕРЕЖ СІР

OSI	Характеристика	Ethernet/IP	ControlNET	DeviceNET	CompoNET
	<i>NetArea</i>	рівень контролерів, рівень датчиків	рівень контролерів	рівень датчиків	рівень датчиків
прикладний	СІР протокол				
	<i>AppService</i>	орієнтовані на з'єднання: I/O Connections — періодичний/аперіодичний обмін даними; Explicit Messaging Connections — аперіодичний обмін параметричними даними; не орієнтовані на з'єднання: Unconnected Explicit Messaging (UCMM); функції конфігурування та діагностики через явні повідомлення; функції резервування;			
	<i>AppModel</i>	pull/push Виробник-Споживач обміну ідентифікованими даними для I/O Connections; Виробник-Споживач обміну повідомленнями для Explicit Messaging Connections; Клієнт-Сервер обміну повідомленнями для Unconnected Explicit Messaging;			
	<i>AppProfile</i>	доступна велика гама профілів пристроїв, описується EDS-файлом			
	<i>адаптація</i>	об'єкти: Ethernet Link, TCP/IP підтримує функції залежать від класу пристрою	об'єкти: Control-Net, Keeper, Scheduling;	об'єкт «DeviceNET», підтримує функції залежать від комун. профілю Ведучий-Ведений	об'єкт «CompoNet» відсутній сервіс Explicit Messaging Connections
	<i>AppProcData</i>	залежить від реалізації	I/O Connections: 504 вх/вих. байт на вузол	Predefined I/O Connections: (4 * 8 вх. Байт, 3*8 вих. байт) на вузол Веденого*	I/O Connections: всього на мережу до 512 AI + 512 AO та 1024 DI + 1024 DO
	<i>AppResolut</i>	залежить від конфігурації	залежить від конфігурації визначається параметром NUT	залежить від конфігурації	
транспортний		відповідно до TCP/IP	див. адаптація прикладного рівня		
мережний		відповідно до IP, NodeAddress= адреса IP, протокол IGMP	див. адаптація прикладного рівня		
каналний	<i>ChAddModel</i>	Відповідно до протоколу EthernetII/IEEE E 802.2/3, комутований ethernet, VLAN, IGMP snooping	Відправник-Адресат-msg, MAC-ID(0...255)	відповідно до ISO 11898, MAC-ID (0...63) в структурі CAN-ID	Відправник-Адресат-msg за MAC-ID: word node (0...127) bit node (0...63) repeater node (0...63)
	<i>ChAccess</i>		CTDMA	відповідно до ISO 11898	TDMA з Ведучим
	<i>ChChecksum</i>		CRC		CRC

Закінчення табл. 2.21

OSI	Характеристика	Ethernet/IP	ControlNET	DeviceNET	CompoNET
фізичний	<i>PhInterface</i>	Відповідно до протоколу Ethernet II/IEEE 802.3	синхронний за напругою з Манчестерським кодуванням	відповідно до ISO 11898	модиф. RS485 (Манчестерське кодування, імпульс трансформат., фільтрац. за маскою)
	<i>PhMedia</i>		75-омний коаксіальний кабель RG-6	спец. 5-жильний кабель (2 сигн. + 2 живл. + 1 екр.),	2-х провідн., 4-х провідн. (2 сигн. + 2 живл.)
	<i>PhTopology</i>		шина	шина	шина, дерево з викор. репітерів
	<i>PhLdrop</i>		1 м, через спец. коробки з BNC/TNC-конекторами	допуск для всіх <i>PhBaudRate</i> для 125 кБит/с <i>PhLdrop</i> < 6 м, $\Sigma PhLdrop$ < 156 м	4 Мбіт/с — не допуск; 1.5 Мбіт/с — через спец. T-відгалуж (<i>PhLdrop</i> < 6 м $\Sigma PhLdrop$ < 120 м),
	<i>PhBaudRate</i>		5 Мбіт/с	125 кБит/с, 250 кБит/с, 500 кБит/с;	4 Мбіт/с, 3 Мбіт/с, 1.5 Мбіт/с, 93.75 Кбіт/с
	<i>PhSegment</i>		до 20 репітерів	відповідно до ISO 11898	64 репітери, до 2-х між Ведучим та Веденим
	<i>PhNodes</i>		99	64	32; всього в мережі до 128 аналогових пристроїв, до 256 — дискретних, до 64 репітерів
	<i>PhLength</i>		1000 м при 2-х відгалуженнях, 250 — при 48; 20 км сегментована, термінатор 75 Ом на кінцях лінії	500 м для 125 кБит/с, 100 м для 500 кБит/с, термінатори 121 Ом на кінцях лінії	30 м, 90 м з репіт. для 4 Мбіт/с; 500 м, 1500 м сегментована для 93.75 Кбіт/с; термінатор 121 Ом на дальньому кінці лінії
	<i>PhSupply</i>		не підтримується	24 В по окремій парі проводів	24 В по окремій парі проводів

* — розраховані на базі 4-рьох типів I/O Connection: Polled I/O Connection, Bit-Strobe I/O Connection, Change of State or Cyclic I/O Connection, Multicast Polling I/O Connection.

2.5.11. Мережі CC-Link

2.5.11.1. Походження. Промислова мережа CC-Link (Control & Communication Link) розроблена Mitsubishi Electric у 1997 році. Для можливості інтеграції CC-Link з пристроями інших виробників у 2000-му році цей протокол став відкритим промисловим рішенням та підтримується організацією **CLPA** (CC-Link Partner Association). Сьогодні CC-Link представлена в 4-рьох різних варіантах мереж:

- **CC-Link** (Версія 1 та 2) для мереж рівня датчиків та контролерів;
- **CC-Link LT** — CC-Link протокол адаптований для мереж рівня датчиків (живлення по мережі, метод доступу ВІТR, збільшена швидкодія, обмежена функціональність);
- **CC-Link Safety** — комунікаційний та прикладний профіль для використання мережі CC-Link у функціонально небезпечних процесах;
- **CC-Link IE** (Industrial Ethernet), — яка базується на технології Gigabit Ethernet та представлена двома рівнями:
 - **CC-Link IE Control** — мережа рівня контролерів;
 - **CC-Link IE Field** — мережа рівня датчиків.

Усі мережі об'єднують сервіси, які використовуються на прикладному рівні, що дозволяє будувати на базі CC-Link інтермережні структури (рис. 2.46). Після створення CC-Link виникла необхідність у розробці мережі низового рівня датчиків, що привело до появи CC-Link LT. Стрімкий розвиток Ethernet, а особливо поява Гігабітного Ethernet дав можливість розробити високошвидкісну мережу CC-Link IE для рівня контролерів. Необхідність у простій інтеграції польового рівня до інформаційного рівня підприємства привела до створення в Mitsubishi Electric напрямку E-factory, результатом роботи якого стало створення CC-Link IE Field.

CC-Link закріплений у стандарті MEK.

2.5.11.2. Фізичний рівень CC-Link. CC-Link базується на інтерфейсі сумісному з RS-485, та використовує шинну топологію. Метод кодування NRZI, максимальна кількість вузлів — 64, бітова швидкість — від 156 Кбіт/с до 10 Мбіт/с, максим. довжина лінії зв'язку — 1200 м (при 156 кбіт/с), в обох кінцях лінії ставиться термінатор 110 Ом. Середовище передачі — спеціальний 3-х жильний екранований кабель (2-жил., інформ., вита пара + сигн. земля) з характеристичним хвильовим опором 110 Ом. При швидкостях, менших за 625 кбіт/с, допускаються відгалуження від магістрального кабелю через спеціальні Т-подібні відгалужувачі. Максимальна довжина відгалужувачів — 8 м, сумарна 200м (при 156 кбіт/с).

2.5.11.3. Канальний рівень CC-Link. Функціонування канального рівня схоже на функціонування PROFIBUS DP. Використовується метод доступу за моделлю Ведучий-Ведений з активними Веденими, в якому Ведучим пристроєм є контролер, а Веденими — периферійні пристрої на шині. Кожен вузол у мережі має унікальну адресу від 0 до 64, де 0-ва адреса зарезервована для Ведучого. Ведучий вміщує буфер вводу/виводу, який складається з 4-рьох частин: вхідні бітові дані (RX), вихідні бітові дані (RY), вхідні слова (RWr) та вихідні слова (RWw). Як і у PROFIBUS DP, Ведучий у CC-Link зчитує вхідні дані (RX та RWr) з використанням полігунгу, однак на відміну від нього відправляє вихідні дані (RY та RWw) в ширококомовному (Broadcast) режимі одним кадром. Такий механізм у CC-Link називається **Broadcast Polling**.

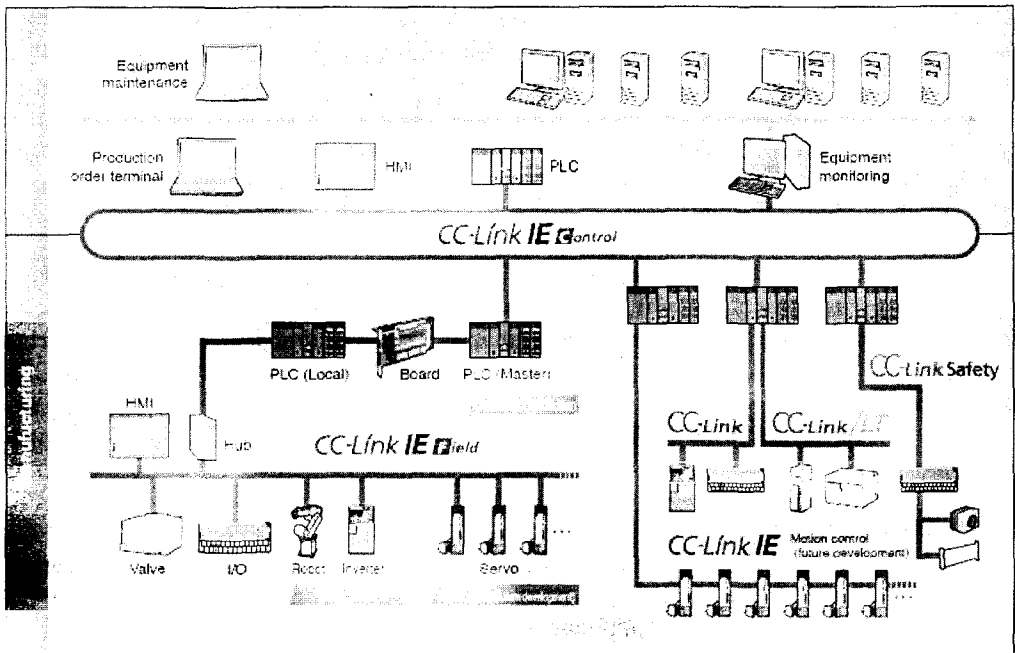


Рис. 2.46. Приклад інтермережі на базі CC-Link

Структура кадру базується на HDLC. Інформаційна частина кадру може включати три частини: бітові дані, слова, ациклічні повідомлення (Transient Message). Коли кадр відправляє Ведучий, бітові дані та слова є вихідними даними, коли ж Ведений — вхідні. Обмеження області кадру для Ведучого: RY-256 байт, для RWw — 512 байт; для Веденого RX-16 байт, RWr — 32 байти.

Для виділення своєї частини вихідних даних із загального кадру кожному вузлу виділяється параметр — кількість заповнених станцій (*Number of Occupied Stations*). Залежно від типу та кількості входів/виходів, які використовуються даним вузлом, йому виділяються від однієї до 4-рх заповнених станцій. Сумарна кількість заповнених станцій у мережі не може перевищувати 64.

2.5.11.4. Прикладний рівень CC-Link

У CC-Link, залежно від особливостей функціонування, всі вузли в мережі такі:

- Ведуча Станція (Master Station);
- Резервна Ведуча Станція (Standby Master Station);
- Ведена Станція (Slave Station):
 - Локальна Станція (Local Station);
 - Інтелектуальна Станція (Intelligent Device Station)
 - Віддалена Станція (Remote Station)
- Віддалений пристрій (Remote Device Station, RX/Ry+RW)
- Віддалена Станція дискретного вводу/виводу (Remote I/O Station, тільки RX/Ry).

Основні Сервіси прикладного рівня забезпечують:

- **Cyclic transmission** — автоматичний циклічний обмін ідентифікованими даними вводу/виводу процесу між Ведучою та Веденими Станціями;
- **Transient Message Transmission / Cyclic Message Transmission** — ациклічний обмін даними між Ведучою та Веденими Станціями, між Локальними станціями;
- **RAS** — функції діагностики, резервування, автоматичне переключення на Резервну Ведучу Станцію у випадку виходу основної.

Циклічний обмін даними процесу забезпечується канальними рівнем, через буфери вводу/виводу. Максимальний обсяг областей кадру визначає обмеження на загальну кількість Вхідів/Виходів у мережі: 256 вхідних реєстрів, 256 вихідних реєстрів, 2048 вхідних бітів та 2048 вихідних бітів. Однак, у 2-й версії CC-Link став можливим обмін у кілька циклів (Extended Cyclic Transmission). Так, дані з пристрою можуть бути фрагментовані і передаватись за 2, 4-ри або 8 циклів. За рахунок такої фрагментації мережа з Ведучим та Веденими Станціями, що підтримують 2-гу версію протоколу, можуть забезпечити до 8192 RX, 8192 RY, 2048 RWw та 2048 RWr.

Крім циклічного обміну ідентифікованими даними, CC-Link ациклічно передає повідомлення. Сервіс Cyclic Message Transmission дає можливість Ведучій Станції замість області даних передавати повідомлення іншій Станції за її адресою. Сервіс Transient Message Transmission для передачі повідомлення використовує відповідну виділену область кадру. Даним сервісом можуть користуватися Локальна або Інтелектуальна Станції для генерації запитів до Ведучої Станції. Тобто при полінгу Станції даного типу в кадрі відповіді, крім даних вводу/виводу Станція може передати запит. Інтелектуальні та Локальні Станції відрізняються від Віддалених станцій Можливістю формування запиту в повідомленні до Ведучої Станції. Локальні Станції можуть формувати запити не тільки Ведучій Станції, а й іншим Станціям.

CC-Link дає можливість резервування Ведучої Станції. Резервна Ведуча Станція в нормальному режимі функціонує як Локальна Станція (має адресу від 1 до 64).

2.5.11.5. Фізичний рівень CC-Link IE Control. Мережа CC-link IE Control використовує технологію Гігабітного Ethernet. Вона базується на стандарті фізичного рівня IEEE 802.3z (1000BASE-SX), де в якості середовища передачі використовується мультимодовий оптичний кабель. Топологія — кільце: всі вузли послідовно з'єднуються дуплексним зв'язком таким чином, щоб замикалось кільце. Така топологія та інтегровані у вузли функції резервування дають можливість при обриві прямого каналу передачі, замкнути кільце на вузлі, де виявлений дефект (рис. 2.47).

Максимальна довжина кільця — 66 км, в якій максимальна відстань між вузлами може досягати 550 м. Максимальна кількість вузлів на мережу — 120.

2.5.11.6. Канальний рівень CC-Link IE Control. Для обміну даними між вузлами використовується механізм Загальної Мережної Пам'яті (Network Shared Memory) обсягом 256 Кбайт та маркерний метод доступу. Серед усіх вузлів виділяється Станція Управління (Control Station), яка слідкує за правильним циркулюванням маркера. В початковому стані Станція Управління відправляє наступ-

ному в кільці вузол кадр з маркером та даними Загальної Мережної Пам'яті, в якій частину даних відновила сама станція (рис. 2.48). Отримавши маркер, вузол змінює свою частину даних і відправляє кадр з маркером наступному вузлу і так далі. Останній вузол передає маркер Станції Управління, тим самим замикаючи логічне кільце. Таким чином, кожен вузол у мережі має доступ для читання до всіх даних Мережної Пам'яті та може змінити свою частину. При виходу з ладу Станції Управління, її функції може взяти на себе інший вузол у мережі.

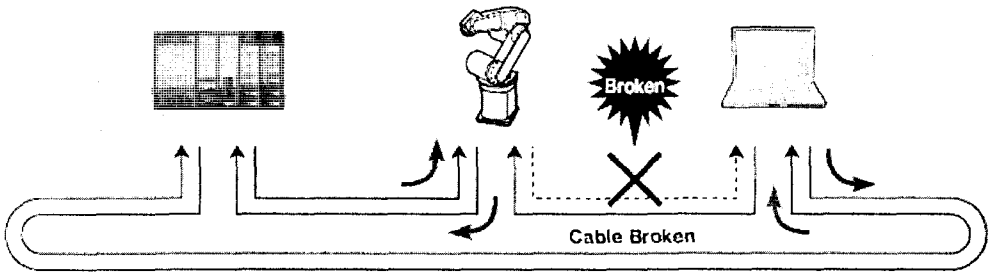


Рис. 2.47. Аварійне перемикання на зворотне кільце в CC-Link IE Control

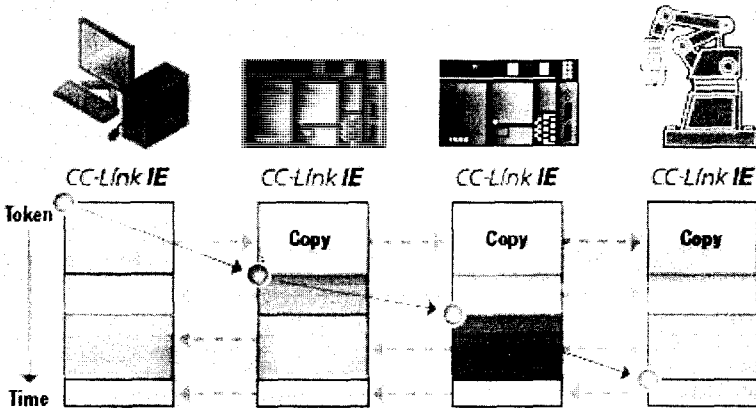


Рис. 2.48. Принципи Обміну Загальною Мережною Пам'ятю в CC-Link IE Control

2.5.11.7. Прикладний рівень CC-Link IE Control. У CC-Link IE Control доступні ті самі сервіси прикладного рівня, що і в CC-Link. Циклічний обмін даними процесу (Cyclic transmission) реалізується через механізм Мережної Пам'яті. Враховуючи, що вихідні дані вузла доступні всім іншим вузлам, можна вважати, що даний сервіс функціонує за моделлю Виробник-Споживач ідентифікованого обміну.

Передача ациклічних повідомлень через Transient Message Transmission можлива за рахунок виділення частини Мережної Пам'яті під ациклічні повідомлен-

ня. Таким чином, отримавши маркер вузол може передати повідомлення іншому вузлу. Формат повідомлення не обговорюється протоколом CC-Link IE, тому може служити в якості транспорту не тільки для прикладних Процесів CC-Link, а й для інших протоколів (наприклад, TCP/IP). Це дозволяє на базі мережі CC-Link IE Control одночасно з реал-тайм — даними процесу функціонувати Intranet-технологіям. Слід зазначити, що ациклічні повідомлення прослуховують усі вузли, що робить цей механізм ще гнучкішим. Функціонування даного сервісу можна описати моделлю Виробник-Споживач обміну повідомленнями.

CC-Link IE Control може функціонувати в одномережній або мультимережній архітектурі. В мультимережній архітектурі (до 239 мереж) кожна мережа та вузол в її межах мають унікальну адресу.

2.5.11.8. Фізичний рівень CC-Link IE Field. На фізичному рівні CC-link IE Field використовується Ethernet з бітовою швидкістю 1Гбіт/с (IEEE 802.3ab (1000BASE-T)) із середовищем передачі вита пара категорії 5е. Можуть використовуватись лінійна, зіркоподібна, кільцева, деревовидна чи змішана топології. В якості з'єднувальних пристроїв допускається використання звичайних (некеро-ваних) концентраторів (але не комутаторів 2-го рівня). Максимальна відстань між станціями — 100 м.

2.5.11.9. Канальний рівень CC-Link IE Field. Так само, як у CC-Link IE Control, у CC-Link IE Field робота канального рівня базується на принципі Загальної Пам'яті та маркерного доступу до середовища. Однак на відміну від першої, в CC-Link IE Field дані Загальної Памяті не передаються від вузла до вузла, а містяться у вхідних буферах усіх вузлів (Розподілена Пам'ять). При отриманні маркера, вузол публікує свої дані (свою частину Пам'яті) в широкомовному режимі, і вони відновлюються у вхідних буферах усіх вузлів.

Максимальна кількість мереж у мультимережній структурі — 239.

Таблиця 2.22

ХАРАКТЕРИСТИКИ МЕРЕЖ CC-LINK

OSI	Характеристика	CC-Link	CC-Link/LT	CC-Link IE Control	CC-Link IE Field
	<i>NetArea</i>	рівень датчиків/ контролерів	рівень датчиків	рівень контролерів	рівень датчиків
прикладний	<i>AppService</i>	– Cyclic transmission — циклічний обмін ідентифікованими даними вводу/виводу процесу між Ведучою та Веденими Станціями; – Transient Message Transmission/ Cyclic Message Transmission — ациклічний обмін даними між Ведучою та Веденими Станціями, між Локальними станціями; – RAS — функції діагностики, резервування, автоматичне переключення на Резервну Ведучу Станцію у випадку виходу основної.			
			RAS та Локальні Станції не підтримуються	Доступний обмін даними між усіма станціями	
	<i>AppProcData</i>	сервіс Cyclic transmission			
		всього на мережу: слова 2048/2048 біти 8192/8192	всього на мережу: біти 2048/2048	всього на мережу: слова 32 кБіт біти 256 кБайт	

Закінчення табл. 2.22

OSI	Характеристика	CC-Link	CC-Link/LT	CC-Link IE Control	CC-Link IE Field	
прикладний	<i>AppResolut</i>	3.9 мс / (256·8 DI + 256·8 DO + 512 AI + 512 AO)	від 0.7 мс до 27.8 мс/(2048 DI + 2048 DO)	до 5 мс/ (32 кБіт + 256 кБайт)		
	<i>AppModel*</i>	Клієнт-Серверна модель обміну ідентифікованими даними для Cyclic transmission; Клієнт-Серверна модель обміну повідомленнями для Transient Message Transmission/ Cyclic Message Transmission		модель Виробник/Споживач обміну ідентифікованими даними для Cyclic transmission; модель Виробник/Споживач обміну повідомленнями для Transient Message Transmission		
	<i>AppProfile</i>	CC-Link Safety				
каналний		базується на HDLC				
	<i>ChAddModel</i>	Відправник-Адресат-msg, вузол може займати 1—4 адрес. 0 — Ведучий		Відправник-Адресат-msg, адреса включає номер мережі та номер станції		
	<i>ChAccess</i>	Ведучий-Ведений з активними Веденими	BITR (Broadcast Polling + Interval Timer Response)	маркерне кільце		
	<i>ChChecksum</i>	CRC			FCS	
	<i>ChSegment</i>	не передбачено			239 сегментів	
фізичний	<i>PhInterface</i>	сумісний з RS-485	спеціальний	IEEE 802.3z (1000BASE-SX)	IEEE 802.3ab (1000BASE-T)	
	<i>PhMedia</i>	скранована вита пара із земл. жилою	4-х жильний кабель	багатоходовий оптичний кабель	вита пара UTP категорії 5e	
	<i>PhTopology</i>	шина, можливі відгалуж	шина з відгалуженнями	дуплексне кільце	зірка, лінійна, кільце, дерево	
	<i>PhLdrop</i>	тільки при $PhBaudRate < 625$ кБит/с, до 8м, $\Sigma PhLdrop < 200$ м), до 6 пристр. на drop	до 60 м, $\Sigma PhLdrop < 200$ м), до 8 пристр на drop	не допускається	відповідно до IEEE 802.3 ab	
	<i>PhBaudRate</i>	156 Кбіт/с до 10 Мбіт/с	156 Кбіт/с до 2.5 Мбіт/с	1 Гбіт/с		
	<i>PhSegment</i>	допускається для відгалужень	не передбачено	не допускається	відповідно до IEEE 802.3ab	
	<i>PhNodes</i>	64		120		
	<i>PhLength</i>	1200 м, 13200м – сегментована	500 м	550 м між вузлами, 66 км загальна		
	<i>PhSupply</i>	не підтримується	по виділенім жилам	не підтримується		

* — моделі умовні

** — модель умовна, в CC-Link називається Broadcast Polling

2.5.11.10. Прикладний рівень CC-Link IE Field. У CC-Link IE Field сервісі циклічного обміну даними процесу (Cyclic transmission) функціонують аналогічно CC-Link IE Control і можуть бути описані моделлю Виробник–Споживач ідентифікованого обміну.

Передача ациклічних повідомлень через Transient Message Transmission можлива за рахунок додаткового часу, виділеного при отриманні маркера для ациклічних повідомлень. У цей час вузол передає повідомлення вузла-адресата, вказуючи його повну адресу: номер мережі та номер станції. Даний обмін можна описати моделлю Клієнт–Сервер обміну повідомленнями.

2.6. Вибір промислової мережі

Надзвичайно велика кількість існуючих промислових мереж не дає можливості зробити навіть поверховий їх аналіз. Вище розглянута тільки певна частина найбільш популярних відкритих промислових мереж. При загальному ознайомленні з ними можна побачити, що кожне з рішень має ряд переваг і недоліків, які роблять їх вигідними при використанні в певній галузі застосування та в певних умовах експлуатації.

Порівнювати переваги того чи іншого рішення можна при умові їх використання в одній і тій же галузі застосування. Однак, сьогодні при виборі промислової мережі необхідно враховувати не тільки функціональні особливості, а і ціновий фактор, вибрану програмну та апаратну платформу, існуючі на підприємстві системи, рівень обслуговуючого персоналу та ін. Спробуємо перелічити критерії вибору промислової мережі в такій формі «вимоги до проектованої системи: функції та можливості промислової мережі».

Залежно від ситуації, ряд перелічених критеріїв стає обмеженнями, а вибір проводиться за іншими критеріями, з урахуванням вагових коефіцієнтів. Загалом, вибір промислової мережі є складним і нетривіальним. За кожною із мереж, наведених у підрозділі 2.5 у підсумковій таблиці характеристик, можна знайти значення для ряду наведених критеріїв. Однак, ряд критеріїв залежать від реалізації вузлів та мережного обладнання, які складають основу мережі; інші можна розрахувати, тільки визначивши всю структуру мережі; а деякі можна визначити тільки з певною долею ймовірності. Так, з наведених підсумкових таблиць по кожній із мереж явно прослідковується відсутність розрахункових даних або формул для визначення орієнтовного часу обміну даними процесу, хоч ця інформація є однією з найважливіших. Тим не менше, на практиці кількість наявних обмежень значно звужує коло можливих рішень, що дозволяє зробити вибір без розв'язання складної багатокритеріальної задачі.

Загальне ознайомлення з принципами функціонування промислових мереж дає можливість зробити вибір мережного рішення. Однак, для проектування мережі та створення апаратного та програмного забезпечення необхідно детально ознайомитися з принципами їх функціонування. В наступних розділах даного посібника більш детально зупинимося на таких промислових мережах: AS-I, MOD-BUS, PROFIBUS, CAN та CANOpen.

Таблиця 2.23

КРИТЕРІЙ ВИБОРУ ПРОМИСЛОВОЇ МЕРЕЖІ

Вимоги до проєктованої системи	Функції та можливості промислової мережі
вимоги до реального часу в системі	мінімально можлива гарантована періодичність відновлення мережних змінних, гарантований мінімальний час доставки повідомлення тощо
вимоги до кількості та типу мережних змінних	обмеження на кількість та тип змінних на один вузол та на мережу в цілому
типи вузлів, які необхідно підключати до мережі	тип мережі — рівень контролерів або датчиків
вимоги до умов експлуатації та середовища передачі системи	вимоги до електромагнітної сумісності (ЕМС), вимоги до ступеня захисту (ІР), вібростійкості, ударостійкості (ІК), пожежобезпеки, вимоги до кабельної продукції і т.д.
вимоги до надійності та живучості системи	можливість резервування вузлів, гарантоване напрацювання на відмову важливих складових, імовірність помилкової доставки даних та час повторення транзакції, час реанімації мережної системи після її збою, можливість заміни складових без зупинки функціонування, наявність та глибина діагностики вузлів і т.д.
вимоги до зручності інсталяції та обслуговування системи	можливість використання комунікаційних коробок, конекторів, добавлення нових вузлів, зручна заміна складових
вимоги до живлення вузлів по мережі	підтримка фізичними інтерфейсами можливості живлення, наявність на комунікаційних портах живлення для засобів мережі
вимоги до розміщення кінцевих точок підключення	топология мережі, максимально можливі довжини відгалужень, максимальна довжина лінії зв'язку, можливість сегментації мережі
вимоги до можливості програмування та управління вузлами через мережу	наявність сервісів конфігурування та програмування в протоколах мережі + підтримка засобами цих сервісів
вимоги до апаратного та програмного забезпечення системи	підтримка даної мережі вказаними апаратними та програмними засобами
вимоги до вартості програмно-апаратного забезпечення	вартість мережних апаратних та програмних засобів, вартість обслуговування
наявність сервісної підтримки та доступної документації для обслуговуючого персоналу	відкритість, популярність мережі, наявність представника в країні та кваліфікація тех. підтримки для даної мережі



Контрольні запитання до розділу 2

1. Дайте визначення промисловій мережі, яке використовується в ДСТУ (під терміном «польова шина»), в МЕК (під терміном «fieldbus») та в даному посібнику. Застосування промислових мереж.

2. Перелічіть та коротко охарактеризуйте типові функції, які виконують промислові мережі.
3. Поясніть, у чому особливості обміну даними процесу в реальному часі. Дайте визначення терміна «реальний час».
4. Поясніть призначення функцій діагностики, конфігурування/програмування, управління станом вузла.
5. Наведіть приклади використання промислових мереж при резервуванні вузлів.
6. Класифікуйте за призначенням дані, які циркулюють у промисловій мережі.
7. Якими форматами даних, як правило, обмінюються в промислових мережах?
8. Класифікуйте тип обміну даними за циклічністю. Наведіть приклади для використання кожного типу.
9. Синхронна, асинхронна та ізохронна відправка даних. Наведіть приклади необхідності синхронної передачі.
10. Перелічіть та коротко охарактеризуйте вимоги, що накладаються на промислові мережі.
11. У чому полягає властивість детермінованості промислових мереж? Доставка яких даних передбачає умови детермінованості?
12. У чому полягають завадостійкість та промислові умови експлуатації? Розкажіть про ступінь захисту IP. На що вказує ІК-код?
13. У чому полягає надійність та живучість промислової мережі?
14. Чим пояснюються вимоги до простоти, зручності інсталяції та обслуговування, живлення вузлів по кабелю мережі, вибір вільної топології?
15. Що привело до появи відкритих стандартів промислових мереж? Що спричинило «війни стандартів» на промислові мережі?
16. Яка нова промислова мережа появилась у результаті розробки єдиного світового стандарту?
17. Які європейські стандарти на промислові мережі були розроблені, які технології в них закріплені?
18. Які стандарти на промислові мережі є сьогодні в межах всесвітньої організації МЕК? Які технології в цих стандартах закріплені?
19. Чим відрізняються «відкриті» мережі від «закритих»? Поясніть призначення моделі OSI.
20. Дайте визначення поняттям: рівень OSI, протокол рівня, сутність рівня, сервіс рівня, інтерфейс рівня, архітектура мережі, стек протоколів.
21. Поясніть призначення фізичного, каналного, мережного, транспортного та прикладного рівнів у контексті моделі OSI.
22. Наведіть перелік робочих характеристик, за допомогою якого можна оцінити промислову мережу.
23. Які сервіси може підтримувати промислова мережа на прикладному рівні? Як може бути реалізований інтерфейс прикладного рівня?
24. Які моделі обміну між прикладними Процесами найчастіше використовуються в промислових мережах? Розкажіть, як вони функціонують.
25. Яким чином визначається місцезнаходження необхідних даних, які беруть участь в обміні? Що таке прикладний профіль пристрою?
26. Які моделі функціонування сервісів прикладного рівня можна виділити в промислових мережах?
27. Як функціонує клієнт-серверна модель обміну повідомленнями? Для яких операцій підходить ця модель? Поясніть необхідність групування даних при такому обміні.
28. Поясніть як функціонує клієнт-серверна модель ідентифікованого обміну (Polling). Для яких операцій вона підходить?

29. Як функціонує модель Видавець-Абонент/Виробник-Споживач ідентифікованого обміну? Що таке процес публікації і хто його ініціює? Чим відрізняється pull-модель від push-моделі ідентифікованого обміну Видавець-Абонент/Виробник-Споживач?
30. Яким чином адресується доставка кадрів вузлам у мережі? В яких системах, як правило, використовується доставка кадрів за адресами вузлів (Відправник-Адресат-msg), а в яких за ідентифікаторами повідомлень (Виробник-Споживач-msg)?
31. Навіщо необхідно регламентувати доступ до середовища передачі? Що таке колізія в мережах з широкоповним доступом? Які варіанти їх запобігання або вирішення?
32. Поясніть як вирішується порядок доступу до шини в системах з методом Ведучий-Ведений? Які недоліки функціонування такої моделі? Поясніть призначення тайм-аутів у системах з централізованим доступом.
33. Які переваги та недоліки функціонування мереж з Активними Веденими порівняно з класичною схемою централізованого доступу?
34. Поясніть, як функціонує обмін на прикладному рівні для систем з централізованим методом доступу. Які моделі обміну при цьому, як правило, використовуються?
35. Як функціонують мережі з використанням Арбітра шини?
36. Розкажіть про функціонування маркерного методу доступу.
37. Які множинні випадкові методи доступу сьогодні використовуються в промислових мережах?
38. Розкажіть про функціонування методу доступу CSMA/CD.
39. Розкажіть про функціонування методу доступу CSMA/CA.
40. Розкажіть про функціонування методу доступу predictive p-persistent CSMA.
41. Розкажіть про функціонування гібридного методу доступу Ведучий-Ведений з передачею маркерного кільця між Ведучими.
42. Розкажіть про функціонування методу доступу TDMA.
43. Як контролюється доставка кадру? Як контролюється правильність отриманих даних у кадрі? Що таке контрольна сума і які алгоритми можуть використовуватися для її розрахунку?
44. Назвіть основні топології мережі, їх переваги та недоліки.
45. Що таке фізичний сегмент мережі? На яких рівнях і з використанням яких програмно-технічних засобів забезпечується сегментація мережі?
46. Які стандартні фізичні інтерфейси використовуються в промислових мережах? Зробіть їх порівняльний аналіз, визначте переваги та недоліки кожного з них.
47. З використанням яких засобів відбувається сегментація на мережному рівні? Дайте визначення поняттям підмережа, інтермережа, маршрутизатор, мережна адреса, маршрутна таблиця, які використовуються в даному посібнику.
48. Поясніть основні принципи маршрутизації, необхідність маршрутних таблиць. Наведіть приклад роботи мережного протоколу.
49. Наведіть приклади типів прикладних Процесів, до яких може відбуватися доступ через промислову мережу. Яким чином адресуються ці Процеси на транспортному рівні?
50. Наведіть та поясніть основні характеристики мереж MODBUS RTU/ASCII та TCP/IP. Як функціонує протокол MBAP?
51. Наведіть та поясніть основні характеристики мережі WorldFIP. Яким чином реалізуються сервіси обміну даними процесу та параметричними даними на всіх рівнях ієрархії?
52. Наведіть та поясніть основні характеристики мереж Foundation Fieldbus: FF H1 та FF HSE. Яким чином реалізуються сервіси обміну даними процесу та параметричними даними на всіх рівнях ієрархії?

- ричними даними на всіх рівнях ієрархії? Яка особливість реалізації прикладного інтерфейсу користувача?
53. Наведіть та поясніть основні характеристики мереж BOSH CAN 2.0 A/B та CAN ISO 11898, чим вони відрізняються.
 54. Розкажіть основні принципи функціонування мережі CANOpen. Яким чином реалізуються сервіси обміну даними процесу та параметричними даними на всіх рівнях ієрархії?
 55. Наведіть та поясніть основні характеристики мереж LON Works. Яким чином реалізуються сервіси обміну даними процесу та параметричними даними на всіх рівнях ієрархії? Яка особливість реалізації прикладного інтерфейсу користувача?
 56. Які мережі PROFIBUS існують сьогодні? Наведіть та поясніть основні характеристики мережі PROFIBUS-DP. Яким чином реалізуються сервіси обміну даними процесу та параметричними даними на всіх рівнях ієрархії PROFIBUS-DP?
 57. Наведіть та поясніть основні характеристики мереж на базі HART-протоколу.
 58. Наведіть та поясніть основні характеристики мережі AS-i.
 59. Наведіть та поясніть основні характеристики мережі INTERBUS. Яким чином реалізуються сервіси обміну даними процесу та параметричними даними на всіх рівнях ієрархії? Поясніть функціонування методу доступу TDMA із Сумарним Кадром в INTERBUS.
 60. Наведіть та поясніть основні характеристики прикладного рівня мереж СІР. Яким чином реалізуються сервіси обміну даними процесу та параметричними даними на всіх рівнях ієрархії?
 61. Наведіть та поясніть основні характеристики мережі DeviceNet. Яким чином вона адаптується до СІР?
 62. Наведіть та поясніть основні характеристики мережі ControlNet. Яким чином вона адаптується до СІР?
 63. Наведіть та поясніть основні характеристики мережі Ethernet/IP. Яким чином вона адаптується до СІР?
 64. Наведіть та поясніть основні характеристики мережі CompoNet. Яким чином вона адаптується до СІР?
 65. Наведіть та поясніть основні характеристики мережі CC-Link. Яким чином реалізуються сервіси обміну даними процесу та параметричними даними на всіх рівнях ієрархії?
 66. Наведіть та поясніть основні характеристики мереж CC-Link IE. Яким чином реалізуються сервіси обміну даними процесу та параметричними даними на всіх рівнях ієрархії? Чим відрізняються CC-Link IE Control та CC-Link IE Field?
 67. Наведіть основні критерії вибору промислової мережі.
 68. В чому полягає складність вибору промислової мережі? Які обмеження крім характеристик промислової мережі, є при виборі конкретного рішення? В чому складність визначення деяких критеріїв при виборі промислової мережі?

РЕАЛІЗАЦІЯ ФІЗИЧНОГО РІВНЯ

3.1. Організація каналу зв'язку

Інформаційний обмін даними між прикладними програмами на всіх рівнях OSI-моделі відбувається у вигляді послідовностей логічних бітів. Канальний рівень передає цю послідовність фізичному рівню, завдання якого — перетворення її в послідовність фізичних сигналів, що відповідають логічній «1» і «0». Тобто фізичний рівень повинен забезпечити фізичну передачу послідовності бітів між канальними рівнями вузлів, пов'язаних між собою.

На фізичному рівні (з точки зору OSI-моделі) описуються механічні, електричні та функціональні характеристики, а також фізичне середовище передачі даних для конкретної мережі.

Фізичним середовищем передачі даних можуть бути: металевий провід (кабель), оптоволоконний кабель, радіохвилі або лазерний промінь, причому деякі мережі можуть базуватися на їх комбінації.

Фізичне середовище, що призначене для переносу інформації між одиницями обладнання, які беруть участь в інформаційному обміні, будемо називати *лініями зв'язку* (Interchange Circuit).

На мережному вузлі безпосередньою передачею даних по лінії зв'язку займається *трансмiтер* або *передавач* (transmitter), а прийомом — *ресивер* або *приймач* (receiver). Трансмiтери та ресивери на принципових схемах, як правило позначають трикутниками з літерами відповідно «G» (або «T») та «R» (рис. 3.1). Пристрій, який об'єднує функції передавача і приймача називають *трансивером* або приймачем-передавачем (transceiver), який, як правило, реалізується у вигляді окремого чіпсету. Трансиверу надходить логічний сигнал «0» або «1» (за рівнем TTL), які він перетворює у певний фізичний сигнал і передає по лінії зв'язку. З іншого боку, інший трансивер робить зворотнє перетворення. Логічний сигнал, який подається на трансмітер для передачі його фізичним каналом, як правило, позначається *TD* (Transmit Data) або *Tx*, а отриманий від ресивера — *RD* (Receive Data) або *Rx*.

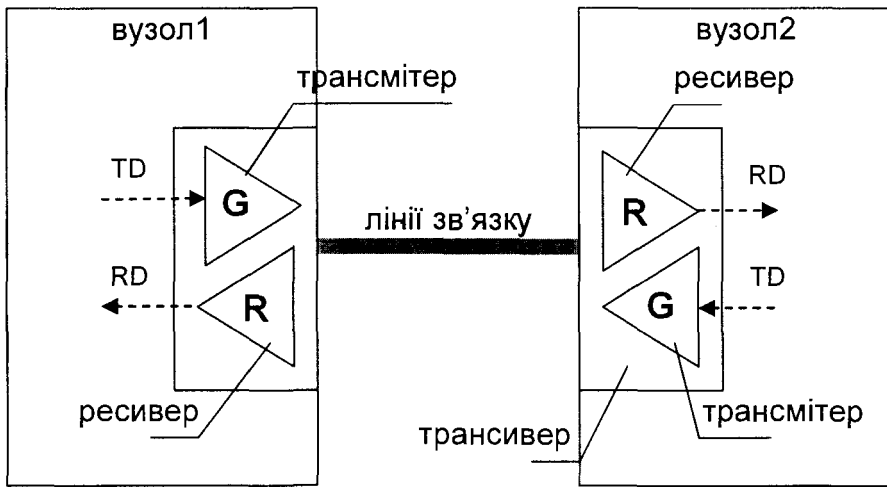


Рис. 3.1. Організація фізичного каналу передачі даних

Основним параметром, який характеризує лінію зв'язку, є **пропускна здатність** (*channel capacity*) — максимальна кількість інформації, яку можна передати за одиницю часу (наприклад, у біт/с). Пропускна здатність фізичної лінії зв'язку (електричного, оптичного чи радіоканалу) тісно пов'язана зі смугою пропускання, потужністю сигналу і рівнем шуму. **Смуга пропускання** (*bandwidth середовища*) визначається як діапазон частот, які лінія може передавати із затуханням менш ніж 3 дБ (що відповідає 50% падінню потужності) і вимірюється в герцах (Гц).

Перенос даних у мережі лініями зв'язку може проходити в трьох режимах: **симплексно** (*simplex*), **дуплексно** (*full duplex*) та — **напівдуплексно** (*half duplex*).

У випадку симплексної передачі (рис. 3.2) дані передаються тільки в одному напрямку. У цьому випадку на одному з вузлів мережі підключений тільки один передавач, а на всіх інших вузлах — тільки приймачі.

При дуплексному з'єднанні одночасно передаються дані в прямому і зворотному напрямках, тобто в мережі трансивери вузлів одночасно працюють як на прийом, так і на передачу. У цьому випадку виникає необхідність у розділенні фізичних ліній зв'язку або їх ресурсів. Для дуплексного з'єднання двох пристроїв можна використати дві різні лінії, одна з яких працює на прийом, а інша — на передачу. Можна також використати різні частоти для передачі за однією і тією ж самою лінією зв'язку. Це дозволяє з'єднувати за дуплексним зв'язком декілька вузлів між собою.

У напівдуплексному з'єднанні дані передаються в прямому і зворотному напрямках, але в різний час. Для цього типу з'єднання характерно розділення однієї лінії зв'язку для передачі від декількох передавачів. У певний момент часу задіяний тільки один передавач, а трансивери всіх інших вузлів можуть працювати в режимі прийому. В інший момент часу активним стає інший передавач. Напівдуплексний спосіб обміну більш економічний порівняно з дуплексним, оскільки використовує тільки одну лінію і не потребує частотного розділення. Однак, він потребує визначення порядку доступу до лінії зв'язку всіх передавачів у мережі.

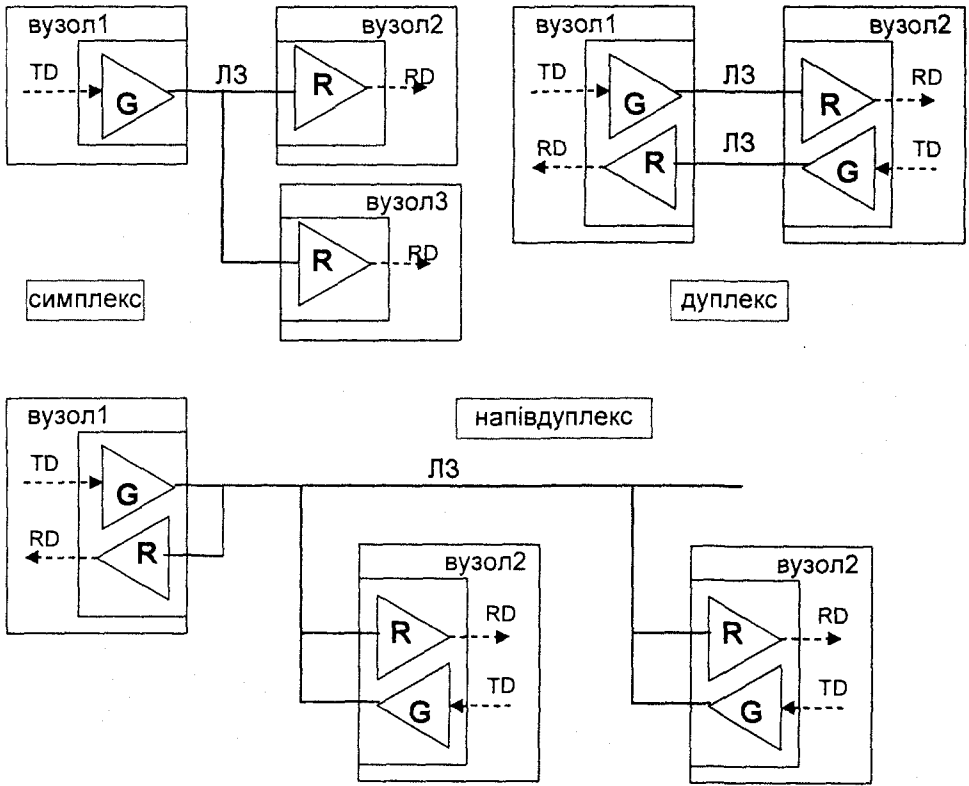


Рис. 3.2. Режими передачі даних

Приклад 3.1. Реалізація фізичного рівня. Напрямки переносу даних

Завдання. Класифікуйте технічні засоби комунікації за типом зв'язку: гучномовець, телебачення, радіо, військова рація (для передачі азбукою Морзе), телефон, міліцейська рація.

Рішення. Симплексний: гучномовець, телебачення, радіо, військова рація (для передачі азбукою Морзе).

Дуплексний: телефон (можна говорити і слухати одночасно).

Напівдуплексний: міліцейська рація (натискаєш — говориш, відпускаєш — слухаєш).

Приклад 3.1

Канал зв'язку може бути реалізований різними середовищами:

- металевим кабелем, за яким передається електричний сигнал;
- оптичним кабелем, за яким передається світловий сигнал;
- по ефіру, з використанням радіохвиль різної частоти.

Трансмітери мають перетворити логічні сигнали «0» та «1» у фізичні електричні, світлові або радіосигнали, а ресивери повинні зробити зворотне перетворення.

рення. Тип каналу зв'язку вибирається залежно від обмежень та вимог, яких слід дотримуватись в промисловій мережі. Нижче розглянуті характеристики кожного із типів каналів та особливості, які необхідно врахувати при їх виборі.

3.2. Використання металевих кабелів

Найбільш дешевий і надійний спосіб передачі — з'єднання передавачів та приймачів металевим кабелем. Носієм сигналу по проводу може бути струм або напруга. Пропускна здатність металевого кабелю залежить від його характеристик, пов'язаних з фізичними особливостями самого кабелю та середовища, в якому він міститься. При виборі металевого кабелю для конкретної промислової мережі враховують такі характеристики:

- тип (вита пара, коаксіальний);
- основні електричні характеристики:
 - хвильовий опір (Ом);
 - погонний опір (Ом/км);
 - погонна ємність (Гн/км);
- додаткові електричні характеристики:
 - коефіцієнт затухання (Дб/км);
 - погонний опір екрана (Ом/км);
- оболонка (матеріал, діаметр, колір);
- діапазон температур (робочий, збереження, транспортування);
- радіус згину (однократний, багатократний);
- наявність галогену в ізоляції;
- пожежобезпечність;
- стійкість до агресивних середовищ;
- інші додаткові характеристики.

Дані характеристики, а також ціна є критеріями вибору для конкретного кабелю. Здебільшого основні характеристики конкретного типу кабелю вказують та рекомендують самі розробники технічних засобів для промислової мережі. Однак, у ряді випадків проєктанту доводиться зустрітись із ситуаціями, що ускладнюють правильний вибір рішення. Наведемо декілька випадків.

1. Виробники технічних засобів для промислових мереж не вказують необхідні характеристики кабельної продукції. У цьому випадку необхідно підібрати кабель за характеристиками сигналів, властивостями трансиверів, зовнішнього середовища, які повинні бути відомі.

2. На підприємстві є в наявності кабель і необхідно прийняти рішення про його використання у даній мережі. Необхідно визначити обмеження на використання даного кабелю для вибраної промислової мережі, за дійсних умов експлуатації.

3. Необхідно зростити декілька різних кабелів. Завдання вирішується аналогічно попередній, з вибором необхідних додаткових матеріалів та способів зрощення.

Способи розв'язання даних проблем виходять за рамки даного посібника. Ми зупинимося тільки на більш загальних характеристиках кабелів та правилах їх монтажу.

3.2.1. Фізичні властивості кабельної лінії передачі

3.2.1.1. Модель кабельної лінії передачі. Канал зв'язку з використанням металевого кабелю може мати вигляд джерела живлення та лампи, які з'єднані між собою двома провідниками (рис. 3.3). Тоді трансмітер буде як джерело живлення з контактом, який управляється логічним станом TD , а ресивер — лампою, світло якої визначає логічний стан RD . Таким чином, при замиканні контакту TD коло замкнеться, і по лінії зв'язку пройде струм, який приведе до загорання лампи, тобто зміни логічного стану RD . Таке уявлення дуже спрощене і не однозначне, однак дає можливість просто перейти до детального розгляду явищ, які відбуваються по всій довжині каналу зв'язку.

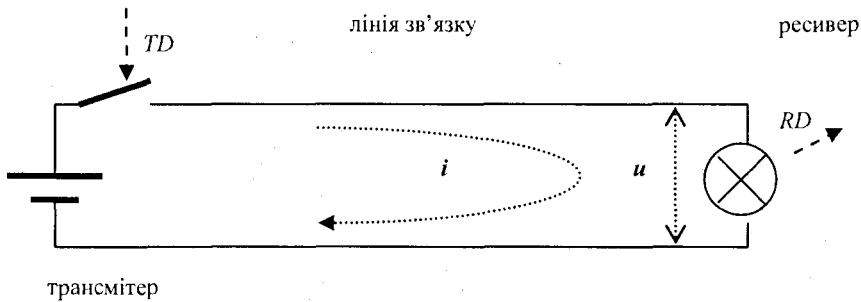


Рис. 3.3. Еквівалентна схема кабельної лінії передачі

На рис. 3.3 трансмітер і ресивер з'єднуються прямим (інформаційним) та зворотним проводом (землю). Таке з'єднання типове для багатьох промислових мереж. Провідники, якими з'єднують трансивери вузлів, об'єднуються в кабелі, конструкція та параметри яких значно впливають на передачу електричних сигналів. Для розуміння фізичних процесів, які відбуваються в кабелі, еквівалентну електричну схему двохпроводного кабелю, показаного на рис. 3.3, уявімо як послідовно з'єднані між собою ланки, кожна з яких є комбінацією двох величин (рис. 3.4.):

- комплексного опору (імпедансу) z , який увімкнено послідовно в ланцюг (Ом/м);
- комплексної провідності (провідності ізоляції), яка з'єднує два проводи між собою (y См/м — Сіменс/метр, 1 Сіменс = 1/Ом).

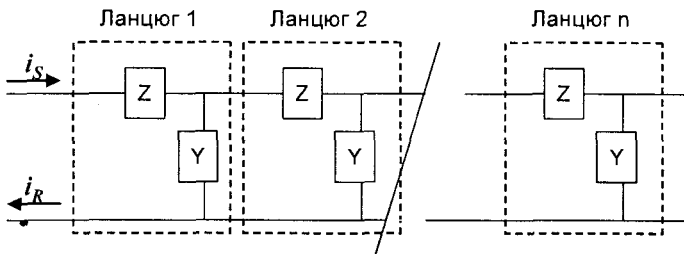


Рис. 3.4. Еквівалентна електрична схема кабельної лінії передачі

3.2.1.2. Комплексний опір. Комплексний опір кабелю визначається послідовним активним опором провідників та реактивною складовою. Активний опір провідників — це опір постійному струму, який залежить від форми, перерізу та матеріалу провідника. В специфікаціях для кабелю вказується його **погонний опір** R_{DC} , тобто опір на одинцю довжини (наприклад, Ом/км). Він є сумою опорів за постійного струму прямого та зворотного провідників.

Природа реактивного опору має індуктивний характер. На рис.3.5. провідник показаний з умовним розрізом, по якому тече струм I .

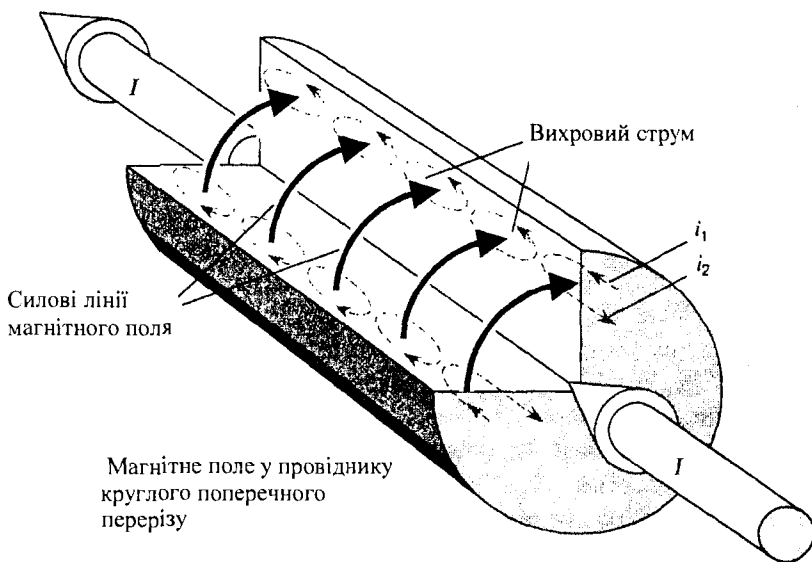


Рис. 3.5. Явище поверхневого ефекту

Об'ємною стрілкою на провіднику показаний напрямок струму. Змінний струм породжує змінне магнітне поле, силові лінії якого проходять через об'єм провідника і створюють у ньому замкнуті вихрові індукційні струми. Збільшення частоти струму приводить до пропорційного збільшення магнітного поля. Як видно з рис. 3.5, напрямок вихрових струмів всередині провідника протилежний напрямку основного струму, а напрямок вихрових струмів у зовнішній частині провідника співпадає з напрямком основного струму. Зі збільшенням частоти струму індуктивні струми будуть зростати, що призведе до зменшення сили струму всередині провідника і збільшення сили струму біля поверхні. Розглянуте явище носить назву **поверхневого ефекту**. На дуже високих частотах струм внаслідок поверхневого ефекту загалом витісниться в тонкий приповерхневий шар провідника.

Інша складова індуктивного опору пов'язана з взаємним впливом провідників з високочастотним струмом, які розташовані один біля одного (рис. 3.6). Змінні магнітні поля цих провідників викликають індукційні струми один в одному. Цей

ефект має назву *ефекту близькості*. При дії ефекту близькості на поверхнях провідників, які повернуті один до одного густина струму збільшується, а на інших сторонах, навпаки, зменшується.

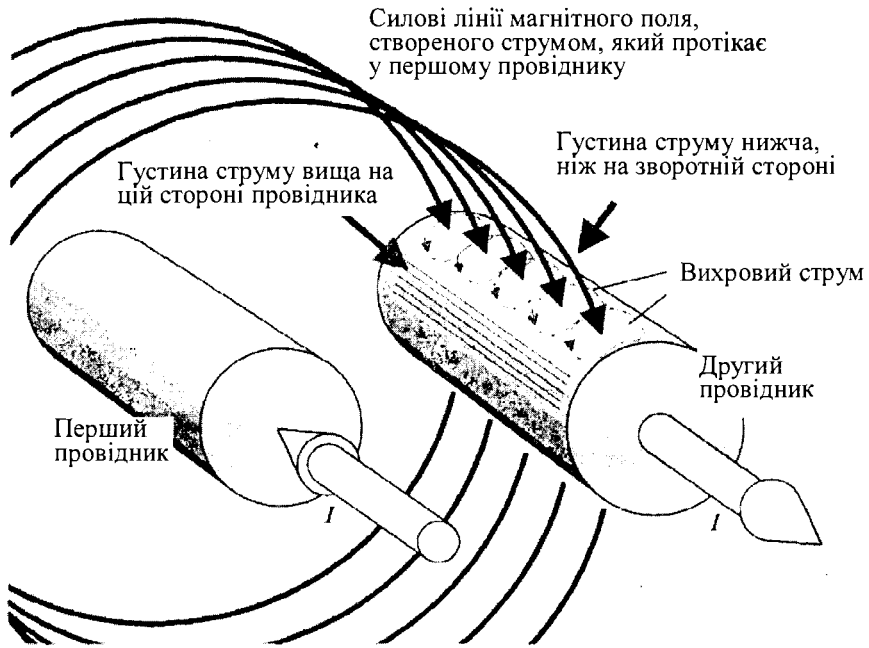


Рис. 3.6. Взаємний вплив індуктивностей провідників

Таким чином, послідовний комплексний опір враховує послідовний опір сигнального проводу, послідовний опір шляху, по якому проходить зворотний струм та індуктивність витка струму, який створений прямим та зворотними струмами.

3.2.1.3. Комплексна провідність. Комплексна провідність теж складається з двох складових: активної провідності, яка обумовлена витком струму через ізоляцію, та реактивної ємності. Ємність виникає внаслідок близького розміщення двох ізованих провідників, по яким тече струм.

Ємність лінії значно впливає на тривалість фронтів сигналів. Якщо уявити ідеалізовану лінію зв'язку з нульовим комплексним опором, яким передається сигнал, то з моменту виникнення переднього фронту біта до моменту його надходження до кінця лінії, розподіл струмів та напруг вздовж лінії буде мати вигляд вказаний на (рис. 3.7). За час t_d хвиля збудження переміститься на відстань $v \cdot t_d$. Біля переднього фронту струм тече через місцеву паразитну ємність лінії, змінюючи її стан, і по зворотному провіднику повертається до джерела імпульсу. В кожен момент часу струм йде на зарядку паразитної ємності тієї ділянки лінії, де розташований передній фронт. Струм тече через ємності тільки в момент появи фронту, оскільки тут змінюється напруга. Тобто з першого моменту виник-

нення напруги на лінії виникає струм, який заряджає першу ємність і повертається до джерела. Потім те саме відбувається з іншими ємностями. Звичайно, вхідні та вихідні струми ємності рівні за величиною. Це значить, що з самого початку струм не тече по сигнальному провіднику до самого кінця лінії, щоб потім вернутись назад. Навпаки, із самого початку вхідний та вихідний струм завжди пов'язані один з одним (нерозривність потоку). Коли фронт імпульсу досягає кінця лінії, прямий та зворотний струми вздовж усієї лінії вже мають усталене значення.

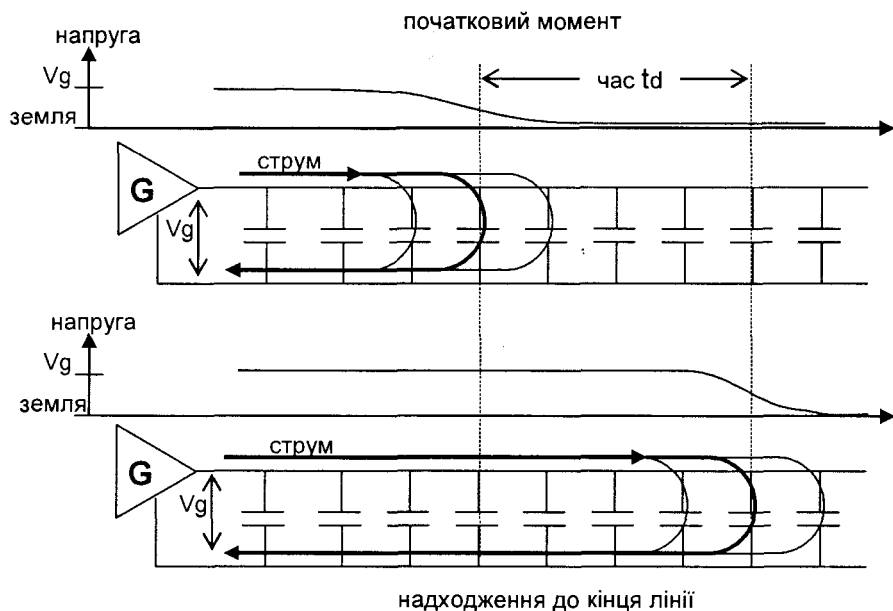


Рис. 3.7. Ємнісний ефект лінії зв'язку

Розглянемо ідеалізовану лінію, якою передається короткий імпульс (рис. 3.8). Проходження переднього фронту сигналу продемонстровано вище. При проходженні заднього фронту, що аналогічно падінню напруги на відповідних ємностях, ємності будуть розряджатись, а їх струми йти на зарядку ємностей по передньому фронту. Таким чином, загальна картина розподілу струмів буде нагадувати гусеницю танка, яка рухається з боку трансмітера до кінця лінії.

3.2.1.4. Розкрита еквівалентна схема кабельної лінії передачі. Кожна ланка являє собою одиничну довжину лінії, яка для кабельних з'єднань може бути в кілометрах або метрах. Тоді для кожного одиничної ділянки кабелю:

- z визначається послідовним з'єднанням активного опору (R) та індуктивності (L);
- y визначається паралельним з'єднанням активної провідності (G) та ємності (C).

Розкрита еквівалентна схема кабельної лінії представлена на рис. 3.9.

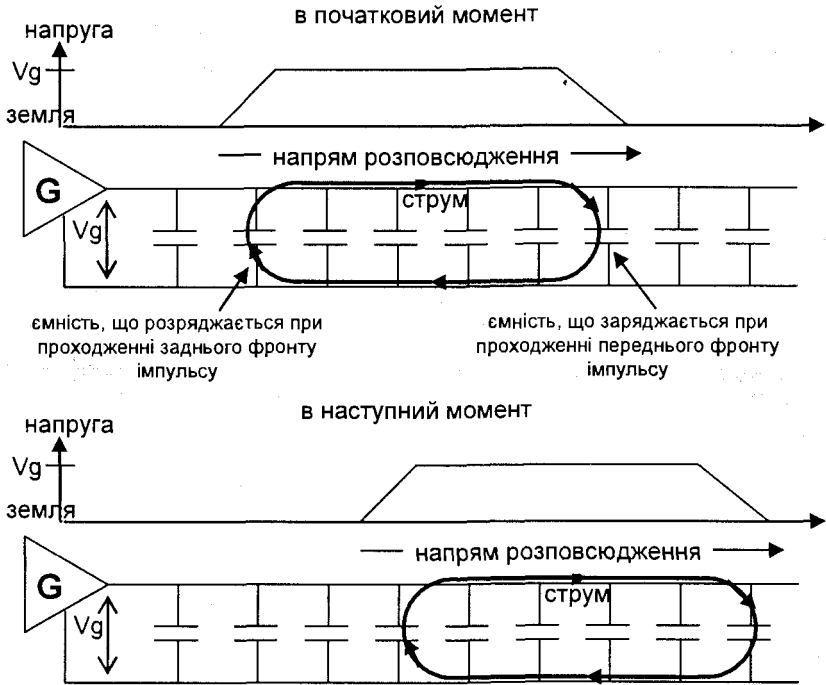


Рис. 3.8. Розподіл струму при короткому імпульсі

При відомих z та у рівняння для еквівалентної схеми дозволяють визначити вхідний імпеданс (опір) та частотну характеристику всієї структури. За цими величинами можна розрахувати реакцію системи на будь-який цифровий вхідний сигнал.

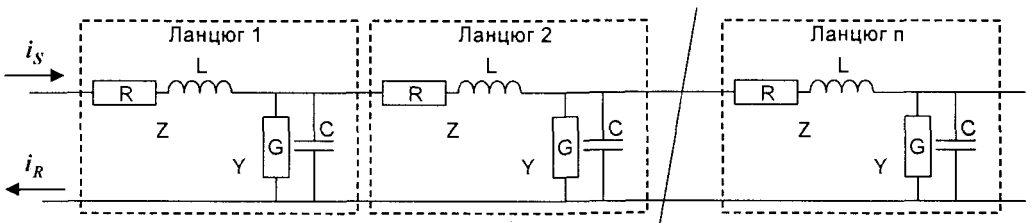


Рис. 3.9. Розкрита еквівалентна схема кабельної лінії передачі

3.2.2. Електричні характеристики кабельної лінії передачі

3.2.2.1. Робочі області кабельної лінії передачі. В різних областях частот ω кабель проявляє різні характеристики. Як видно з рис. 3.10, діапазони областей розміщуються в такому порядку:

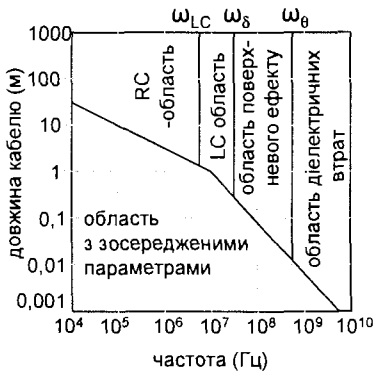


Рис. 3.10. Робочі області залежно від частоти

При наступному збільшенні частоти внутрішня індуктивність провідників стає значною порівняно з їх активним опором. Це викликає зміну розподілу струму за об'ємом провідників, що є ознакою області поверхневого ефекту.

При значній частоті лінія зв'язку переходить в область діелектричних втрат, в якій проявляються значні діелектричні втрати, розмірні з резистивними втратами.

Розподіл частот у робочій області дозволяє для конкретного випадку використовувати конкретну модель розрахунку для кабелю. Промислові мережі в більшості випадків працюють в області частот 10 кбіт/с — 10 Мбіт/с, що при методі передачі NRZ еквівалентна приблизно 5 кГц — 5 МГц. Це значить що промислові мережі при лініях зв'язку порядку 10—1000 м попадають в RC, частково в діапазон переходу з RC в LC-область.

У зв'язку з тим, що значення деяких еквівалентних параметрів для кожного кабелю залежить від частоти, їх можуть не приводити у специфікації. Для більшості кабелів у технічній документації приводять такі характеристики:

- характеристичний хвильовий опір (Ом);
- коефіцієнт затухання (дБ/км);
- погонний опір (Ом/км);
- погонна ємність (Гн/км).

3.2.2.2. Хвильовий опір. *Хвильовий опір* Z_c — це опір лінії зв'язку електромагнітній хвилі за відсутності відбиття від кінців лінії. В кожній точці хвилі, яка поширюється з одного кінця лінії зв'язку в інший, напруга та струм змінюються, однак відношення між ними залишаються постійними. Це відношення напруги до струму і є хвильовим опором. Хвильовий опір визначається для всієї лінії, у будь-якій точці, якщо її параметри незмінні за довжиною лінії.

Доведено, що хвильовий опір можна визначити за таким відношенням:

$$Z_c = \sqrt{\frac{z}{y}} \text{ або } Z_c = \sqrt{\frac{j\omega L + R}{j\omega C + G}}. \quad (3.1)$$

- RC-область, область низьких частот ($0 - \omega_{LC}$);
- LC-область ($\omega_{LC} - \omega_\delta$);
- область поверхневого ефекту ($\omega_\delta - \omega_\theta$);
- область діелектричних втрат (вище ω_θ).

На достатньо низьких частотах, тобто в RC-області, величина індуктивного опору ωL дуже мала порівняно з активним опором за постійним струмом. Тому все, що проходить нижче цієї межі визначається опором лінії та її ємністю.

Збільшення частоти призводить до збільшення індуктивного опору кабелю і зрештою перевищує омичний опір лінії. Лінія переходить у робочу область LC.

З (3.1) та рис. 3.11 видно, що хвильовий опір лінії залежить від усіх її електричних параметрів та частоти.

3.2.2.3. Характеристичний хвильовий опір. На високих частотах величинами R та G можна знехтувати, тоді відношення в (3.1) буде прямувати до відношення індуктивності та ємності, тобто:

$$Z_0 = \lim_{\omega \rightarrow \infty} Z_C(\omega) = \sqrt{\frac{L}{C}}. \quad (3.2)$$

Величину Z_0 називають номінальним або *характеристичним хвильовим опором* (Characteristic impedance). В характеристиках кабелю зв'язку вказують саме це значення хвильового опору.

3.2.2.4. Коефіцієнт затування. По мірі розповсюдження сигналу по лінії він послаблюється в декілька разів на кожній одиниці довжини. При незмінних параметрах кабелю по всій його довжині, *коефіцієнт затування* залишається незмінним на кожній ділянці лінії, однак він залежить від частоти. Коефіцієнт затування приймають на певну одиницю довжини, наприклад (дБ/км). Для сигналу по напрузі коефіцієнт затування розраховується за формулою:

$$K_{\text{зат}} = 20 \cdot \log \left(\frac{u_1}{u_2} \right) \text{ дБ}, \quad (3.3)$$

де u_1 та u_2 — напруга сигналу відповідно на вході та виході кабелю.

У каталогах на кабельну продукцію коефіцієнт затування можуть приводити для різних частот.

Приклад 3.2. Реалізація фізичного рівня. Розрахунок необхідного коефіцієнта затування.

Завдання. Розрахуйте необхідний коефіцієнт затування для кабелю, призначеного для з'єднання трансмітера та ресивера двох вузлів за таких умов:

- швидкість передачі даних — 19 200 біт/с;
- використовується пряме кодування бітів NRZ;
- відстань — 1200 м;
- при мінімальній напрузі на виході трансмітера 1,5 В мінімальна напруга на вході приймача повинна бути не меншою 0,2 В.

Рішення. Розрахунок коефіцієнта затування кабелю:

$$K_{\text{зат}(1200)} = 20 \cdot \log \left(\frac{1,5}{0,2} \right) = 17,5 \text{ дБ на } 1200 \text{ м.}$$

Коефіцієнт затування на 1 км буде:

$$K_{\text{зат}(1000)} = 17,5 \cdot \frac{1000}{1200} = 14,6 \text{ дБ/км.}$$

При прямому кодуванні швидкість 19200 біт/с приводить до максимальної частоти 10 кГц (9600 Гц). Для більшості кабелів типу «вита пара» коефіцієнт затування при такій частоті нижче вказаної, а отже більшість кабелів підходять для даного з'єднання.

3.2.3. Відбиття сигналу та способи їх подолання

На рис. 3.12. показана загальна еквівалентна схема передачі сигналу. Вона складається з вихідного імпедансу джерела Z_S , вхідного імпедансу лінії зв'язку (кабелю) Z_0 та імпедансу навантаження Z_L . Еквівалентна схема кабелю зв'язку показана на рис. 3.9. Як видно з цієї схеми вона являє собою коливальний контур. Отже при подачі одиничного імпульсу по кабелю будуть йти коливання, спричинені наявною ємністю та індуктивністю даного кабелю. Тобто струм, дійшовши до кінця лінії частково розсіється на навантаженні, а інша частина повернеться назад. Відбитий сигнал повертається до джерела, частково розсіюється на ньому, після чого повторно відбивається. Сигнал буде «гуляти» з одного кінця в інший до тих пір, поки повністю не розсіється імпедансами передавача та приймача. Ефект відбиття негативно впливає на передачу, оскільки відбитий сигнал буде спотворювати корисний сигнал.

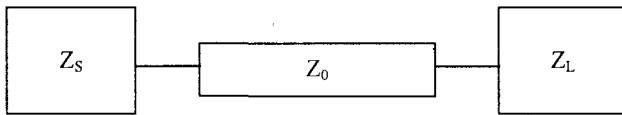


Рис. 3.12. Загальна еквівалентна схема передачі сигналу

Розглянемо, яким чином можна подолати дану проблему. Відносна амплітуда сигналу, яка відбивається в напрямку джерела сигналу, описується такою функцією:

$$R(\omega) = \frac{Z_L(\omega) - Z_0(\omega)}{Z_L(\omega) + Z_0(\omega)} \quad (3.4)$$

Як видно із (3.8), сигнал відбиття можна повністю погасити, зробивши опір приймачів Z_L , рівним хвильовому опору кабелю Z_0 .

У шинних топологіях приймачі розміщені по всій довжині лінії. Тому для усунення ефекту відбиття в кінці лінії ставлять резистори з номіналом, рівним характеристичному хвильовому опору кабелю. Якщо трансмітери містяться в різних ділянках кабелю (не на одному кінці) то необхідно ставити резистори в обох кінцях кабелю. Для боротьби з відбиттям сигналів можуть використовуватися й інші схемні рішення. Загальна назва цих засобів — *термінатори ліній*. Слід зазначити, що хвильовий опір лінії зв'язку залежить від частоти (3.1), тому для роботи в широкому діапазоні частот розрахунок термінатора стає не таким тривіальним.

Для коротких ліній зв'язку термінатори не потрібні, оскільки ефект відбиття не відчутний. Лінія може вважатися короткою, якщо дотримується нерівність:

$$\text{довжина кабелю} < \frac{1}{6} \frac{T_{\text{rise}}}{(LC)^{1/2}}, \quad (3.5)$$

де T_{rise} — довжина фронту сигналу (с), L, C — відповідно погонна індуктивність (Гн/м) та погонна ємність (Ф/м).

3.2.4. Типи електричних кабелів

У промислових мережах як правило використовують два типи електричних кабелів: вита пара та коаксіальний кабель.

Вита пара — пара проводів, які ізольовані один від одного, мають однакову довжину і звиті разом. При використанні напруги в якості передаючого сигналу використання витієї пари, порівняно зі звичайним кабелем, зменшує випромінюючі та індукуючі електромагнітні завади за рахунок скручування (два паралельні проводи створюють просту антену або конденсатор, а скручені — ні).

Виті пари існують в декількох варіантах і поділяються на категорії. Вони відрізняються за перерізом проводів, характеристичним хвильовим опором, кількістю витків на метр. У комп'ютерних мережах набули популярності кабелі **UTP** (unshielded twisted pair — неекранована вита пара). В промислових мережах, як правило, використовують екрановану виту пару для зменшення впливу зовнішніх завад. При частотах у декілька мегагерц та вище кабель з одинарним екраном може збільшити захист в 100 раз при умові, якщо екран заземлений правильно. Залежно від виконання екранів, екрановані виті пари можна поділити на два типи:

- екран з фольги: недорогий, простий в установці, однак крихкий, при деформації помітно зменшує свої захисні властивості;

- екран з оплітки: не втрачає свої властивості при деформації, ізгибах і т.ін.

Коаксіальний кабель складається із покритого ізоляцією твердого мідного проводу, розміщеного в центрі кабелю. Поверх ізоляції натягнутий циліндричний провідник, який виготовлений у вигляді мілкої мідної сітки. Він покритий зовнішнім захисним шаром ізоляції.

Конструкція і спеціальний тип екранування коаксіального кабелю забезпечує високу пропускну здатність і відмінний захист від завад. Максимальна пропусканна здатність залежить від якості, довжини і співвідношення сигналу/шуму лінії. Сучасні кабелі мають смугу пропускання близько 1 ГГц. Для передачі цифрових даних використовують, як правило, 50-омний кабель.

Крім електричних характеристик кабелю, не менш важливими є додаткові характеристики, пов'язані з матеріалом оболонки. Зокрема, це радіус згину, діапазон робочих температур, наявність галогену в ізоляції, пожежобезпечність, стійкість до ультрафіолету та до дії агресивних середовищ. Пожежобезпечність кабелю визначається його ступенем горіння, наявність галогену вказує на отруйний склад диму при горінні, чутливість до ультрафіолету визначає негативний вплив відповідних променів на фізичні властивості оболонки.

3.2.5. Електричні шуми, завади та боротьба з ними

У промислових умовах експлуатації сигнал, що проходить по електричних кабелях, може бути спотворений паразитними струмами, які виникають у результаті електричних наводок різної природи. Типовими джерелами електричних завад є пристрої, що генерують імпульси напруги або струму: великі електродвигуни в момент включення/виключення; флуоресцентні освітлювальні лампи;

високовольтні імпульси напруги, що викликаються несправностями електричного обладнання (пробої); зварювальне обладнання; інший провідник з високочастотним сигналом; інші джерела електромагнітних завад.

Можна виділити чотири типи дії електричних завад на лінії передачі даних:

- імпедансний зв'язок;
- електростатичний зв'язок;
- магнітний або індуктивний зв'язок;
- радіочастотне випромінювання (електростатичний + магнітний зв'язок).

3.2.5.1. Імпедансний зв'язок. У тих випадках, коли ланцюги зв'язку розділяють між собою загальні проводи (загальна земля), струм одного ланцюга проходить по загальному провіднику всіх зв'язаних ланцюгів і приводить до виникнення на цьому загальному провіднику напруги похибки, яка впливає на інші сигнали. Напруга похибки обумовлена ємністю, індуктивністю та опором зворотного проводу. Тобто, в результаті наявності імпедансного зв'язку на лінії зв'язку може впливати інший ланцюг, в якому використовується та сама сигнальна земля. Одне з вирішень цієї проблеми — це використання окремих зворотних провідників для кожного сигналу, а для промислових мереж — симетричного способу передачі сигналу, який розглянутий нижче.

3.2.5.2. Електростатичний зв'язок. Рівень завад при електростатичному (ємнісному) зв'язку пропорційний ємності між джерелом завад та сигнальними проводами. Інтенсивність завади залежить від швидкості зміни напруги завади і від ємності між ланцюгом завади та сигнальним ланцюгом. Величина напруги завад у сигнальних проводах обернено пропорційна: відстані від джерела шумів до сигнальних проводів; довжині (і, відповідно, імпедансу) сигнальних проводів, на які наводяться завади; амплітуді (величині) напруги завад; частоті напруги шумів.

Для зменшення рівня завад, які викликаються електростатичним зв'язком можна виконати такі дії: екранування сигнальних проводів (екрановані кабелі); відокремлення від джерела завад; зменшення амплітуди напруги і частоти завад; скручування сигнальних провідників (кабелі, виті пари).

При екрануванні провідників електростатичний зв'язок замкнеться через екран на землю, з якою він обов'язково повинен бути з'єднаний. Опір екрана повинен бути якнайменший.

Скручування провідників забезпечить взаємне гасіння паразитних струмів, наведених на обох провідниках, оскільки ці струми будуть спрямовані в різні боки.

3.2.5.3. Магнітний або індуктивний зв'язок. Цей тип зв'язку залежить від зміни струму завади і взаємоіндукції між джерелом завад та сигнальними проводами. Рівень завад, що викликаються магнітним зв'язком, обернено пропорційний відстані від джерела завад до сигнальних проводів і залежить від: амплітуди струму завади; частоти струму завади; площі, через яку проходить магнітний потік струму завади з сигнальними проводами; відстані від джерела до сигнальних проводів.

Один з варіантів зменшення рівня завад, викликаних магнітним або індуктивним зв'язком, є скручування сигнальних проводів. Це приводить до зменшення наводок внаслідок зменшення площі кожної петлі, що означає послаблення магнітного потоку, що проходить через петлю, і, відповідно, зменшення напруги, що індукується. Крім того, напруга завад, яка індукується в кожній петлі, має тенде-

нцію компенсувати напругу завад від наступної сусідньої петлі. Таким чином, парна кількість петель буде мати тенденцію взаємної компенсації напруги завад окремими петлями. Іншим способом є використання навколосигнальних проводів екрана з високою магнітною проникністю (наприклад, залізо). Магнітний потік, що генерується струмами завади, індукує в магнітному екрані невеликі вихрові струми. Далі ці струми створюють магнітний потік, обернений до вихідного потоку, що призводить до його зменшення. Слід зазначити, що для боротьби з даним типом завад немає необхідності в заземленні екрана. Прокладка кабелів у металічних каналах також зменшує дію завади.

3.2.5.4. Радіочастотне випромінювання. Напруга завад, що індукується наявністю електростатичного та індуктивного зв'язку, проявляється у вигляді ефекту поля у ближній зоні — чим є електромагнітне випромінювання біля джерела завад. Цього типу завад часто важко позбавитись, і він потребує акуратного заземлення сусідніх електричних ланцюгів. З'єднання із землею часто є єдиним ефективним засобом для цілей, які перебувають у безпосередній близькості до електромагнітного випромінювання. Найчастіше використовують такі способи зменшення електромагнітного випромінювання: відповідне екранування; встановлення конденсаторів, які забезпечують стікання напруги завад на землю.

Однак, будь-який не повністю екранований провідник буде працювати для радіосигналу як приймальна антена і, відповідно, необхідно забезпечити хороше екранування будь-якого відкритого проводу.

3.2.5.5. Рекомендації до зменшення дії завад

Таким чином, зменшення дії всіх типів завад можна такими способами:

1. Використання кабелю типу «вита пара» замість звичайного (не скрученого) кабелю.
2. Використання екранованих кабелів.
3. Зменшення відстані до джерела завад.
4. Використання кабельних каналів та коробів зі спеціального матеріалу.
5. Рознесення кабелів різного рівня на відстані.
6. Встановлення фільтрів.

При прокладці кабелів необхідно враховувати їх взаємне розміщення. Для тих ситуацій, коли є велика кількість кабелів з різними рівнями напруг та струмів, був розроблений стандарт IEEE 518-1982, що є корисним набором таблиць рекомендованого взаємного віддалення для різних типів кабелів. Можна виділити три рівні категорії кабелів:

1. До 1-го рівня (висока сприйнятливність) належать аналогові сигнали нижче 50В та цифрові сигнали з амплітудами менше 15 В. Сюди входять цифрові логічні шини та телефонні лінії а також комунікаційні кабелі для передачі даних.

2. До 2-го рівня (середня сприйнятливність) належать аналогові сигнали з напругою більше 50В та комутаційні ланцюги.

3. До 3-го рівня (низька сприйнятливність) належать комутаційні сигнали та аналогові сигнали з напругою більше 50 В зі струмами менше 20 А.

4. До 4-го рівня (силовий рівень) належать напруги в діапазоні до 100В та струмами в діапазоні від 20 А до 800 А. Це справедливо як для змінного, так і постійного струму.

Сприйнятливостю в даному контексті є те, наскільки добре сигнальний ланцюг може відокремлювати паразитну заваду від корисного сигналу. Із цього випливає, що промислові мережі мають високу сприйнятливостю, а 100-вольтові 200-амперні кабелі змінного струму — низьку.

Необхідна відстань між кабелями 1-го рівня до інших визначається згідно з табл. 3.1.

Таблиця 3.1

ВЗАЄМНЕ РОЗМІЩЕННЯ КАБЕЛІВ РІЗНОГО РІВНЯ КАТЕГОРІЇ

розміщення кабелів	до кабелю рівня 2	до кабелю рівня 3	до кабелю рівня 4
обидва в лотку	30 мм	160 мм	670 мм
один в лотку, а інший — у кабельному каналі	30 мм	110 мм	460 мм
обидва в окремих кабельних каналах	30 мм	80 мм	310 мм

3.2.6. Заземлення екранів кабелів

3.2.6.1. Необхідність заземлення. Екранування використовується для захисту провідників або обладнання від завад, що викликані електромагнітним випромінюванням інших провідників або обладнання. Екранований кабель повинен застосовуватись, якщо потрібно обмежити дію зовнішніх збурень, здатних вплинути на «чутливий» кабель, або, навпаки, — щоб обмежити дію самого кабелю на зовнішнє середовище.

Нагадаємо, що обов'язковою умовою використання екранованого кабелю є його заземлення. Взагалі система заземлення на будь-якому об'єкті повинна: відводити струми замикання на корпус обладнання і струми витоку в землю; відводити синфазні струми зовнішніх кабелів у землю; відводити струми блискавок у землю; задовольняти вимогам забезпечення електробезпеки людей (<25 В змінного струму або <50 В постійного).

З'єднання між собою різних відкритих електропровідних частин обладнання провідниками заземлень (зелено-жовтим або РЕ) створює ланцюг низького опору на низьких частотах. У результаті цього забезпечується вирівнювання потенціалів між двома відкритими електропровідними частинами, які небезпечні для людини. Це заземлення задовольняє вимогам забезпечення електробезпеки людей. Для кожної будівлі такий ланцюг низького опору з'єднується в одному місці із землею і по ній відводяться синфазні струми, що виникають. Однак таке з'єднання різних відкритих провідних частин заземлюючими провідниками не забезпечує еквіпотенціальності на високих частотах, яка необхідна для ефективного захисту від дії електромагнітних завад, оскільки повний опір РЕ провідників дуже великий (1 мкГн/м).

Виходячи з цих обставин, правила заземлення кабелю для конкретної промислової мережі диктуються стандартами для неї, виробниками технічного забезпечення вузлів мережі та конкретними умовами експлуатації обладнання. На жаль, ці правила не завжди виявляються ефективними.

3.2.6.2. Різні підходи до заземлення. Сьогодні існують два кардинально протилежні підходи до заземлення екрана кабелів:

- заземлення екрана з одного кінця кабелю (з боку вводу кабелю в щит);
- заземлення екрана на обох кінцях кабелю.

Заземлення екрана тільки з одного кінця кабелю пояснюється тим, що декілька точок заземлення створюють паразитні земляні контури. Тобто, якщо «землі» в кінцях кабелю будуть мати різні потенціали, то по екрану потече струм, викликаний цією різницею.

Однак, при частотах сигналу більше 1 МГц збільшується індуктивний опір екрана, і струми ємнісної наводки створюють на ньому велике падіння напруги, яке може передаватися на внутрішні жили через ємність між екраном та жилами. Крім того, при дожині кабелю, яка співрозмірна з довжиною хвилі (довжина хвилі на 1 МГц дорівнює 300 м, 10 МГц — 30 м), збільшується опір екрана. Тому на високих частотах екран кабелю необхідно заземляти з обох кінців. При цьому по екрану кабелю буде протікати низькочастотна частина струму, викликана різницями потенціалів земель та ємнісним зв'язком, яка буде наводитиме шум на інформаційних проводах з частотою, що дорівнює частоті живлення, тобто 50 Гц. Однак, для високочастотного цифрового сигналу даний шум не буде спотворювати інформаційні біти. Для виключення можливості протікання низькочастотного струму, викликаного різницею потенціалів земель, на одному з кінців кабелю між екраном і землею можна поставити фільтр, наприклад конденсатор, який для високочастотної складової матиме низький імпеданс.

3.2.6.3. Забезпечення еквіпотенціальності. Для ефективного захисту високочастотних ланцюгів вимагається спеціальне еквіпотенціальне з'єднання. З'єднання між собою всіх металічних частин є для високочастотного сигналу ланцюгом низького опору. Досягнуте в результаті цього вирівнювання потенціалів у мережі електропровідних частин забезпечує ефективний захист від електромагнітних завад завдяки зниженню рівня всіх високочастотних напруг, які інакше могли б існувати між одиницями обладнання. Металічні частини повинні бути з'єднані між собою гвинтами, гайками та болтами або короткими широкими металічними оплітками (перерізом 25 мм² та довжиною менше 30 см) або еквівалентною металічною частиною. Захисні РЕ-провідники не можна застосовувати для еквіпотенціального з'єднання. Втім, система еквіпотенціальних з'єднань повинна бути обов'язково заземлена із загальною землею.

Оскільки відкриті електропровідні частини забезпечують опорний потенціал для електронного обладнання та зворотний контур для синфазних струмів, будь-який струм, що попадає по кабелю в обладнання, ізольоване відносно відкритих провідних частин, виходить через інші кабелі. Тому, якщо якість системи еквіпотенціальних з'єднань незадовільна, то кабель, по якому протікає синфазний струм, створює завади для всіх інших пристроїв. Ефективне еквіпотенціальне з'єднання знижує небезпеку цього явища.

Для забезпечення еквіпотенціальності між електропровідними частинами різних будівель або з віддаленим обладнанням, необхідно їх об'єднати одним із таких типів кабелів (*кабель еквіпотенціального з'єднання*):

- мідний провід з мінімальним перерізом 6 мм^2 ;
- алюмінієвий провід з мінімальним перерізом 16 мм^2 ,
- сталевий провід з мінімальним перерізом 50 мм^2 .

3.2.6.4. Правила монтажу заземлення екрана. Для ефективного функціонування екрана необхідно правильно проводити монтаж його заземлення (рис. 3.13). Найкращим варіантом є надійне з'єднання екрана з кабельним вводом в стінці корпусу за умови, що для отримання хорошого електричного контакту на корпусі в місці з'єднання видаляється фарба.

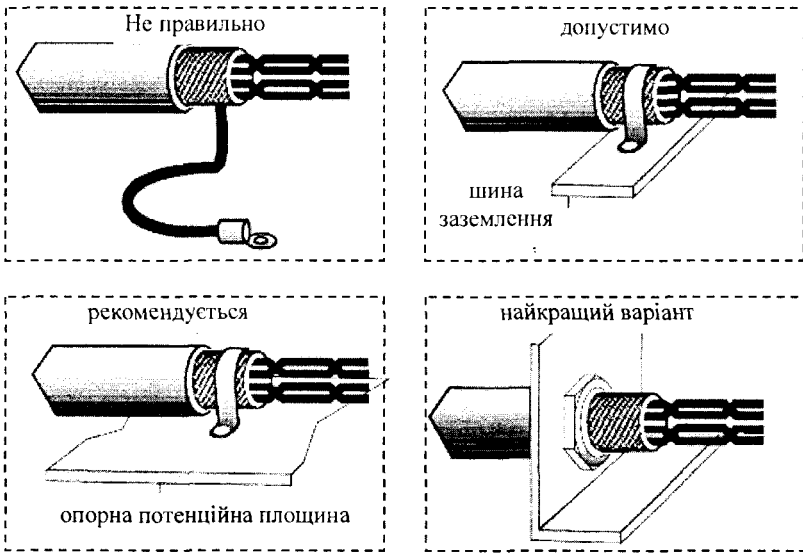


Рис. 3.13. Правила заземлення екрана

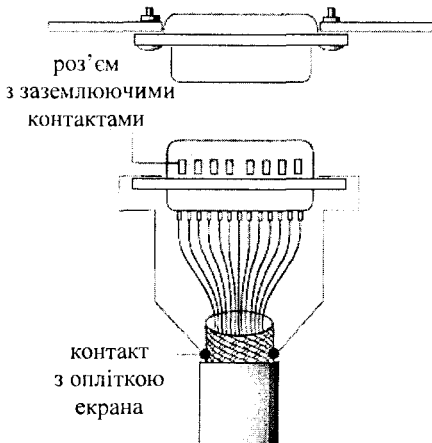


Рис. 3.14. Правила заземлення екрана роз'ємів із заземлюючими контактами

Якщо при підключенні кабелю використовується роз'єм, він повинен забезпечити круговий неперервний електричний контакт між екраном кабелю та корпусом обладнання (рис. 3.14).

Якщо для заземлення екрана при приєднанні до зажиму під гвинт не можна використати U-подібний хомут, провід заземлення повинен бути якомога коротшим (рис. 3.15). Однак такого типу з'єднання краще уникати.

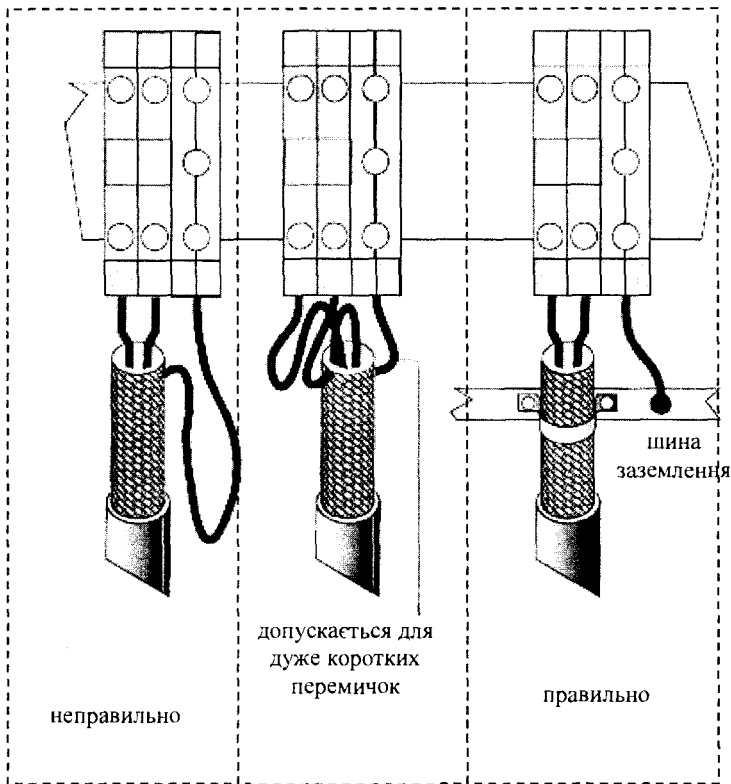


Рис. 3.15. Правила заземлення екрана з вузлами кріплення на DIN-рейку

3.2.7. Основні рекомендації до вибору, прокладки та монтажу електричних кабелів для промислових мереж

При проектуванні промислових мереж необхідно звернути увагу на такі рекомендації.

1. При виборі, прокладці та монтажу кабелів слід дотримуватись правил, визначених виробниками обладнання, стандартів до вибраної мережної технології (особливо апаратного інтерфейсу) і тільки у випадку відсутності такої інформації — рекомендацій даного посібника.

2. В якості електричного середовища передачі фахівці радять використовувати екрановану виту пару або виту пару з подвійним екраном, або коаксіальний кабель, для яких характеристики відповідають заданим параметрам:

- для витих пар — необхідній кількості витих пар (подвійна, три і т.д.);
- для коаксіального кабелю — тип кабелю: вузькополосний чи широкополосний;
- заданому характеристичному хвильовому опорі, якщо він відомий;
- менше заданого коефіцієнта затухання в кабелі, при відомій частоті сигналу, максимально можливого падінню напруги;

- е. менше заданого погонного опору, розрахованого за довжиною необхідної лінії зв'язку;
 - ф. необхідному матеріалу, кольору, діаметра оболонки: за діапазоном робочих температур; необхідній гнучкості — радіус згину; вимогам пожежобезпеки; стійкості до кислоти, лугу, масла, палива і т.д.; отруйності диму при згоранні.
3. При шинній топології в крайніх кінцях кабелю сегмента (з боку ресиверів) виставляти термінатори ліній з опором, рівним хвильовому опору кабелю.
4. Забезпечити еквіпотенціальність ділянок, пов'язаних кабелями промислових мереж;
- а. всі металічні конструкції в межах будівлі необхідно з'єднати між собою металічними балками, гвинтами, болтами, кабельними каналами, зварювальними поверхнями і т.д., з якнайменшим опором та найкоротшими з'єднувальними ділянками;
 - б. лінії функціонального заземлення мережного обладнання необхідно під'єднати до еквіпотенціального контуру за правилами, аналогічними попередній рекомендації;
 - с. для еквіпотенціального з'єднання фарбованих поверхонь необхідно обов'язково видалити фарбу в місцях з'єднання;
 - д. контактуючу поверхню необхідно обробити антикорозійним електропровідним матеріалом (наприклад, оцинковка);
 - е. кінці кабелів, що забезпечують еквіпотенціальність, необхідно залудити або обжати відповідним типом наконечника;
 - ф. частини кабельних каналів, необхідно об'єднати між собою еквіпотенціальною системою зварювання або металічним з'єднувачем з того самого матеріалу;
 - г. при з'єднанні кабелів з іншими будівлями або частинами будівель необхідно паралельно з інформаційним кабелем прокласти кабель еквіпотенціального з'єднання: мідний провід з мінімальним перерізом 6 мм^2 ; алюмінієвий провід з мінімальним перерізом 16 мм^2 , сталевий провід з мінімальним перерізом 50 мм^2 .
5. При використанні невеликих бітових швидкостей (порядку 100 Кбіт/с) екрани необхідно заземляти тільки в місцях підводу кабелю до щита — для запобігання паразитних земляних струмів.
6. При використанні швидкостей більше 100 Кбіт/с екрани необхідно заземляти в обох кінцях кабелю, з використанням конденсатора в одному з кінців.
7. При прокладці кабелю всередині будівлі необхідно користуватися такими правилами:
- а. для забезпечення захисту від високочастотних завад усі кабелі повинні бути прикріплені до конструкцій системи еквіпотенціальних з'єднань; для цього найкраще використати електропровідні кабельні канали, виконані за правилами еквіпотенціального заземлення;
 - б. кабелі необхідно регулярно прикріпляти до металічних конструкцій;
 - с. тільки пари, які передають аналогові, цифрові або телекомунікаційні сигнали, можуть прокладатися близько один до одного в одній зв'язці або в одній групі;
 - д. всі вільні провідники кабелю повинні бути заземлені разом з екраном;
 - е. при використанні кабельних каналів неекрановані кабелі необхідно розмістити в кутках кабельних лотків;

f. при використанні кабельних каналів для недопущення сумісного розміщення несумісних кабелів (силові та інформаційні) необхідно використовувати кабельні перегородки ;

g. при використанні кабельних каналів частина кабельного лотка з інформаційними кабелями (і тільки вона) повинна бути закрита металевою кришкою;

h. при прокладці інформаційних кабелів разом із силовими останні необхідно екранувати;

i. кабелі можуть пересікатися тільки під прямим кутом;

j. кабелі різного рівня категорії повинні бути по можливості в різних лотках або один від одного на відстані, згідно з табл. 3.1;

8. При прокладці мережного кабелю між будівлями необхідно користуватися такими правилами:

a. кабель мережі повинен розташовуватись максимально близько до кабелю еквіпотенціального заземлення;

b. з'єднання зовнішнього кабелю з внутрішнім проводити в спеціально зробленій для цього заземленій металічній коробці з реалізацією захисних схем від високої напруги;

c. зовнішній кабель промислової мережі необхідно прокладати під землею на глибині не менше 50 см;

d. кабель еквіпотенціального заземлення необхідно прокладати вище на 20 см від кабелю мережі;

e. між кабелями різної категорії прокладають металеві перегородки.

3.3. Передача напругою та струмом

3.3.1. Використання напруги

В інтерфейсі фізичного рівня обумовлюють, який сигнал за напругою визначений для кожного логічного стану. Кожний приймач являє собою певне навантаження тобто вхідний імпеданс ресивера. Якщо крізь нього пропустити струм, на ньому пройде певне падіння напруги. Передавач має забезпечити приймачі таким струмом, щоб напруга на них відповідала необхідному рівню, визначеному в інтерфейсі для заданого логічного стану.

Якщо вхідний імпеданс приймача не буде безкінечно великим, то по кабелю буде проходити струм. Якщо збільшувати імпеданс приймачів, то необхідний менший струм для передачі того самого сигналу за напругою, однак це приводить до чутливості їх до паразитних струмів, які спричиняються різними джерелами завад. Тому в мережах опір приймачів вибирають так, щоб, з одного боку, зменшити потужність передавачів, а з іншого — зменшити чутливість до завад.

Напруга може передаватись двома способами:

– **незбалансований (несиметричний, unbalanced)**, коли сигнал передається як різниця потенціалів між сигнальним проводом та сигнальною землею (нулем);

– збалансований (симетричний, диференційний, balanced differential), коли сигнал передається за допомогою різниці потенціалів між двома проводами: по одному проводу йде оригінальний сигнал, а по іншому — його інверсна копія відносно загальної землі (3.16).

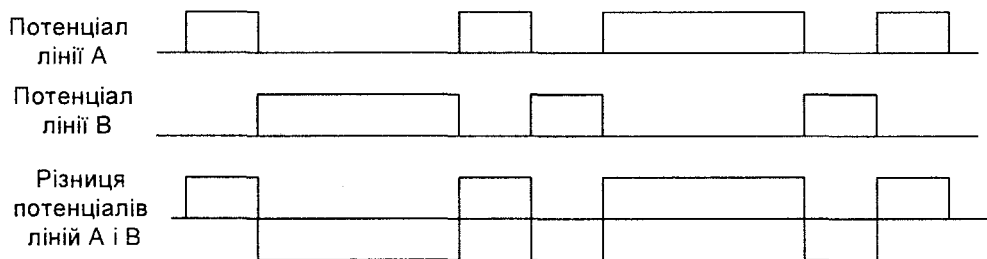


Рис. 3.16. Передача сигналу напругою збалансованим способом

Збалансований спосіб передачі забезпечує високу стійкість до синфазних завад, тому що завада діє практично однаково на обидва проводи, залишаючи різницю потенціалів між сигнальними проводами незмінною, а саме вона вимірюється приймачами.

3.3.2. Використання струму

Для передачі сигналу на великі відстані краще використовувати струм, тому що він залишається постійним за довжиною кабелю. Окрім того використання струму робить мережу менш чутливою до завад ніж при використанні напруги. Для забезпечення надійної роботи каналу необхідно вибрати приймачі з якнайменшим вхідним імпедансом. Струмові сигнали, як правило, використовуються на порівняно низьких швидкостях передачі даних (як правило, до 9600 біт/с). При збільшенні частоти вплив ємнісного ефекту збільшується і частина струму буде втрачатися по довжині кабелю в зворотний провід або в захисний екран.

3.4. Використання оптоволоконного кабелю

При передачі на значні відстані (більше кількох кілометрів) великих обсягів даних використання металевого дроту в якості носія стає проблематичним. Як альтернативу можна використати оптоволоконну систему. Вона складається з оптоволоконного кабелю, генератора світлових сигналів (передавача), детектора (приймача), обладнання для обробки сигналу на обох кінцях кабелю і допоміжних елементів (рис. 3.17).

Ширина смуги пропускання каналу і припустима довжина лінії визначаються затуханням і спектральною дисперсією оптоволоконного кабелю, вихідною потужністю генератора і чутливістю світлового датчика на прийомному кінці. Оптичний кабель описується параметром, який характеризує ширину смуги пропус-

кання і відповідну їй максимальну відстань, які обернено пропорційні один одному. Цей параметр вимірюється в МГц · км і визначає межу працездатності кабелю. Для будь-якого оптичного кабелю — чим більша довжина, тим вужча смуга пропускання. Оскільки шум практично не впливає на оптоволоконні канали, смугу пропускання можна використовувати в повному об'ємі.

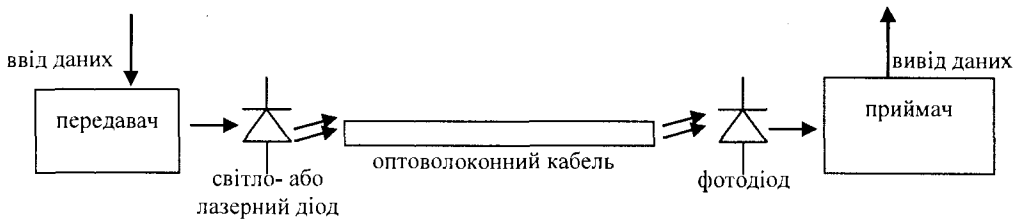


Рис. 3.17. Принцип роботи оптоволоконної системи зв'язку

Оптичні провідники являють собою надзвичайно тонкі (діаметром в долі міліметрів) і легкі волокна, виготовлені із прозорої речовини — кварцу, скла або пластмаси. Волокно складається із серцевини (core) і зовнішньої оболонки (cladding), які захищені зовнішнім пластиком кожухом (sheath). Показник заломлення серцевини більш високий, ніж у зовнішньої оболонки, тому світлові промені відбиваються від оболонки і розповсюджуються по волокну. Оптоволоконні жили зазвичай групуються в пучки, які захищені зовнішньою оболонкою (футляр).

Кожен світловий промінь, який рухається по оптоволоконній жилі, має певну довжину хвилі або, як кажуть, — певну моду (mode). Оптичне волокно, яке має властивість передавати відразу декілька променів (з різною довжиною хвиль), називається *багатомодовим* (multimode fiber). Волокна з товщиною до кількох довжин світла дозволяють передавати промінь практично по прямій лінії без відбиття від стінок волокна. Такі волокна називають *одномодовими* (single mode fiber). Багатомодові волокна, в свою чергу діляться на:

- багатомодові зі стрибком показника заломлення (step index multimode fiber): недорогі волокна зі співвідношенням смуги пропускання на довжину приблизно 35 МГц · км, використовуються на незначних відстанях;

- багатомодові з плавною зміною показника заломлення (graded index multimode fiber): з відповідним співвідношенням 500 МГц · км дозволяють передавати світловий сигнал на більші відстані ніж попередні.

Товщина серцевини одномодового волокна зі стрибком показника заломлення (step index single mode fiber) близько 10 мікрон, що дає змогу пропускати тільки одну довжину хвилі. Таке волокно виробляється з кварцу і має низький коефіцієнт затухання та дуже широку смугу пропускання порядку 10 ГГц · км. Це найдорожче волокно, яке використовується при передачі на значні відстані.

Джерелом світла (передавачем), яке перетворює цифрові електричні сигнали в світлові імпульси, в оптичній системі зв'язку є або світло-випромінюючий, або лазерний діод. Світло-випромінюючий діод (light-emitting diode — LED) має об-

межену вихідну потужність до 0.1 мВт і максимальну бітову швидкість ≈ 8 Мбіт/с. Більш дорогі лазерні діоди (laser diode) мають вихідну потужність до 10 МВт і можуть забезпечувати ШПД більше 10 Гбіт/с. Тільки останні можуть генерувати світло у вузькій смузі довжин (порядку 5 нм) і тому використовуються при одномодовій передачі. Недоліки лазерних діодів — в чутливості до температури, короткому терміні служби та високій ціні.

Детектором (приймачем) є фотодіод, який перетворює світлові імпульси в електричні сигнали. Якщо необхідно забезпечити високу чутливість, використовуються лавинні фотодіоди (photoavalanche diode).

Основні переваги оптичної передачі:

- забезпечення великих швидкостей передач даних;
 - практично не підвладна діям зовнішніх завад;
 - можлива передача на дуже значні відстані;
 - надзвичайно легкі та тонкі кабелі (в одному кабелі може вміститись велика кількість жил);
 - хімічна нейтральність і відсутність можливості корозії.
- Поряд із цим є і недоліки:
- дороговизна;
 - складність у спаї двох волокон;
 - чутливий до механічних дій, крихкий;
 - однонаправлена передача (для двохсторонньої передачі необхідно виділяти окрему жилу або окрему частотну смугу).

3.5. Використання бездротового зв'язку

Не завжди в системах управління в якості фізичного носія можна використати кабель (металевий чи оптоволоконний). Розглянемо декілька прикладів, коли виникає необхідність у бездротовому зв'язку:

- контроль та управління мобільними (пересувними) системами;
- віддалений моніторинг стану системи;
- управління «розумними» будинками;
- відсутність можливості прокладки кабелів.

У першому випадку ситуація очевидна — використання кабелів для зв'язку з мобільною системою (об'єктом управління) обмежує її рух. Крім того, навіть при великій гнучкості кабелів термін їх служби зменшується за рахунок втоми металу або фізичного ушкодження.

Віддалений моніторинг необхідний операторам або технологам для отримання аварійних повідомлень з контролюючої системи (скажімо, SCADA), перебуваючи при цьому у будь-якій точці підприємства. Така система також зручна при аналізі працездатності системи, шляхом перевірки територіально розподілених складових.

Усе більш перспективним напрямком стає автоматизація в багатоповерхових та приватних житлових будинках, що включає в себе управління електро-, тепло- та водопостачанням, пожежну та охоронну сигналізацію і т.д. Альтернативою провідникам для підключення датчиків, виконавчих механізмів, контролерів може

служити радіосигнал, а в якості панелі оператора може бути мобільний телефон, смартфон, комунікатор і т.д.

Звісно, за відсутності можливості прокладки кабелів, або високої вартості його, альтернативи безпроводному зв'язку немає. Безпроводні способи зв'язку дуже чутливі до різного роду завад, тому мережі, побудовані на них, використовують велику ширину смуги пропускання для службової інформації з метою боротьби з помилками. Детальніше технології безпроводного зв'язку розглянуті в розділі 4.

3.6. Кодування інформації

У промислових мережах використовують послідовний спосіб передачі даних. Існує два способи передачі бітової послідовності по фізичному каналу:

- посылка біт у лінію в безпосередньому або закодованому виглядах при збереженні цифрового характеру даних;
- модуляція несучої за амплітудою/частотою/фазою і передача модульованого сигналу.

3.6.1. Безпосередній спосіб передачі цифрових даних

При безпосередньому кодуванні низький рівень сигналу (наприклад, 0 В чи 0 мА) може відповідати логічному «0», тобто *space* (спокій), а високий (наприклад, 10 В чи 20 мА) — логічній «1», тобто *mark* (відправка) (рис. 3.18, а). Такий спосіб називається *прямим кодуванням*.

Часто користуються *зворотним кодуванням*, — коли логічний «0» передається високим рівнем сигналу, а «1» — низьким, а також *полярним кодуванням*, — коли сигнали «0» і «1» протилежні за знаком за напругою відносно загальної точки. Пряме, зворотне і полярне кодування називаються кодуванням без повернення до нуля *NRZ* (Non-Return to Zero), тому що в них відсутній обов'язковий перехід до нульового рівня.

Цей метод простий в реалізації, але чутливий до завад і спотворень. Крім того, приймач не може розрізнити, де закінчується і починається кожний окремий біт, а також налаштуватися на необхідну швидкість. Це значить, що він не зможе відрізнити відсутність сигналу від послідовності бітів (представлених одним рівнем сигналу) та працювати при динамічно змінюваних швидкостях без допоміжної синхронізації.

Вказані вище проблеми вирішуються при кодуванні з поверненням до нуля *RZ* (Return to Zero) (рис. 3.18, б). У даному випадку вихідні дані для передачі комбінуються із сигналом синхронізації. Кожний біт, як і в попередньому випадку, починається з визначеного для нього рівня сигналу, а в середині кожного імпульсу відбувається перехід на нульовий рівень, по фронту якого приймач синхронізується з передавачем. Може також використовуватись *RZ*-кодування із сигналами різної полярності (рис. 3.18, в). Ці переваги над *NRZ*-кодуванням досягаються шляхом виділення в два рази більш широкої смуги пропускання, а також більш складною реалізацією.

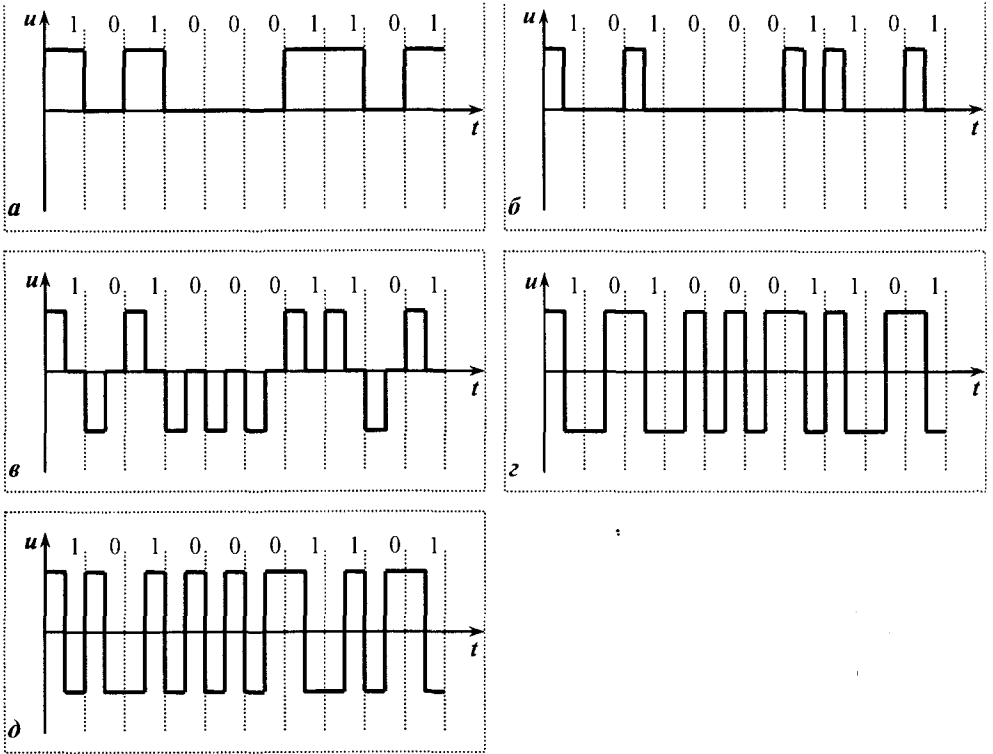


Рис. 3.18. Способи цифрового кодування:

a — пряме двійкове без повернення в нуль (NRZ); *б* — пряме двійкове з поверненням в нуль (RZ); *в* — тривірневе з поверненням в нуль; *г* — манчестерське; *д* — диференційне манчестерське

Манчестерське або двофазне кодування *Bi-phase Level (BiФ-L)* використовує два рівні напруги з переходом в середині кожного імпульсу (біта). При прямому манчестерському кодуванні «0» представлений переходом від низької напруги до високої (передній фронт), а «1» — від високої до низької (задній фронт) (рис. 3.18, *г*). При **диференційному манчестерському** кодуванні, як і у попередньому способі, кожен біт являє собою зміну напруги, але сигнал передається не типом фронту, а наявністю чи відсутністю його зміни: «0» — фронт сигналу не змінюється, «1» — фронт сигналу змінюється з «переднього» на «задній» або навпаки (рис. 3.18, *д*). Можна сказати інакше: якщо рівень сигналу (напруги) не змінюється при зміні періоду передачі одного біта, то наступний біт — логічна «1».

Манчестерський та диференційний манчестерський є кодами з убудованою синхронізацією (дозволяють передавачу налаштуватися на приймач без допоміжних синхросигналів) і мають більш високий захист від шуму, ніж RZ-кодування при використанні такої самої ширини пропускання. Крім того, використовуються два рівні сигналів замість 3-х, що робить реалізацію більш простою, ніж при RZ-кодуванні.

3.6.2. Модуляція несучої (*carrier modulation*)

Модуляція несучої являє собою зміну деяких параметрів (амплітуди, частоти, фази) високочастотної несучої як функції вихідного інформаційного сигналу. В якості несучої використовується високочастотний синусоїдальний або імпульсний сигнали. Різні типи модуляції визначаються типом несучої і процедурою модуляції. Розрізняють:

- амплітудну модуляцію АМ (**AM** — Amplitude Modulation), коли змінюють амплітуду несучої;
- частотну модуляцію ЧМ (**FM** — Frequency Modulation), при якій змінюється частота несучої, як функція від вхідного сигналу;
- фазову модуляцію ФМ (**PM** — Phase Modulation), коли змінюється фазовий зсув несучої.

У випадку модуляції несучої цифровим сигналом розрізняють:

- амплітудну маніпуляцію АМн (**ASK** — Amplitude Shift Keying) (рис. 3.19, б);
- частотну маніпуляцію ЧМн (**FSK** — Frequency Shift Keying) (рис. 3.19, в);
- фазову маніпуляцію ФМн (**PSK** — Phase Shift Keying) (рис. 3.19, г).

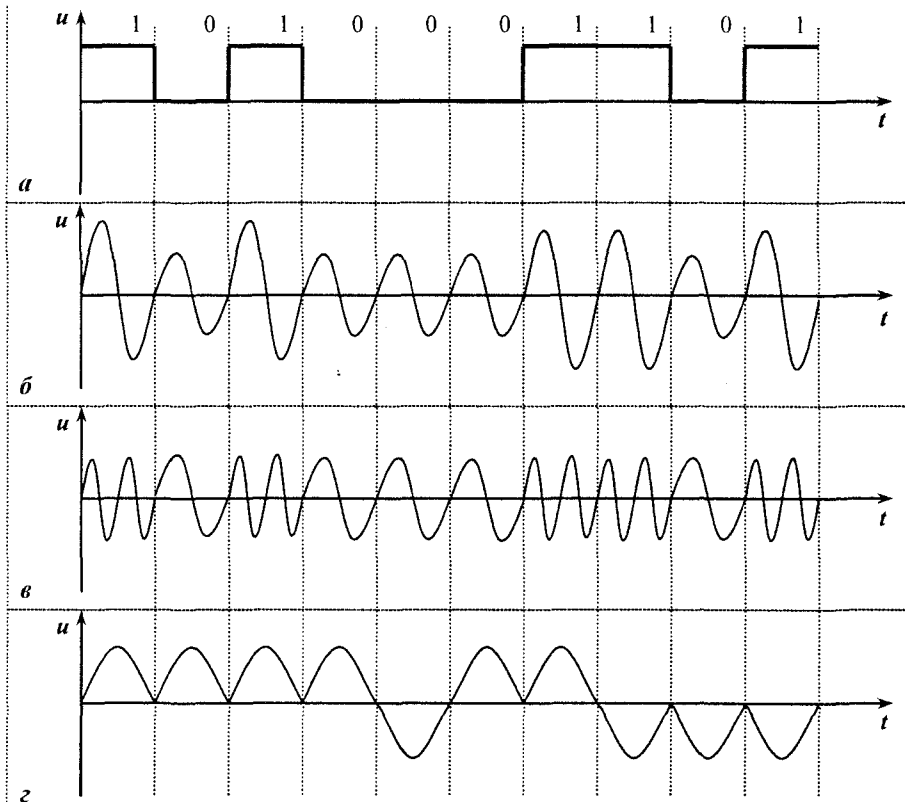


Рис. 3.19. Модуляція несучої:
 а — двійковий сигнал; б — амплітудна модуляція;
 в — частотна модуляція; г — фазова модуляція

Серед трьох названих методів фазова маніпуляція найбільш стійка до завад. Сьогодні, як правило, використовують маніпуляцію несучої не як функцію від двох станів («1» або «0»), а як функцію від комбінації бітів. Наприклад, послідовності 00, 01, 10, 11 можуть кодуватися фазовим зсувом відповідно 0° , 90° , 180° і 270° , при цьому обсяг даних збільшується у два рази. Складнішими маніпуляціями, при яких для передачі декількох біт об'єднуються амплітудна і фазова модуляція, є:

- квадратурно-фазова маніпуляція (**QPSK** — Quadrature Phase Shift Keying);
- квадратурно-амплітудна модуляція (**QUAM-16**, **QUAM-64** — Quadrature Amplitude Modulation).

Число змін одного з параметрів несучої називається *сигнальною швидкістю* або *швидкістю передачі у бодах* (*Baud rate*) і вимірюється у бодах. Кожним бодом передається один символ, який складається з декількох бітів, що визначається методом модуляції. *Битовою швидкістю* називається обсяг бітів, який передається по каналу за 1 сек, включаючи службові біти та біти синхронізації, і вимірюється в біт/с. При модуляції вона дорівнює добутку сигнальної швидкості і кількості бітів на символ.

3.7. Синхронна та асинхронна передача

3.7.1. Необхідність синхронізації

При обміні даними між вузлами необхідно, щоб передавач і приймач використовували одне і те саме джерело часу, тобто, щоб вони були *синхронізовані*. В протилежному випадку, приймач не зможе визначити, коли надходить перший біт, тобто розпізнати початок послілки. Але навіть при початковій синхронізації джерела тактів передавача і приймача, а також домовленості про час початку відправки, при передачі великих послідовностей бітів може виникнути розсинхронізація.

На рис. 3.20. показана послідовність бітів, яку передає трансмітер і розпізнає ресивер. Пунктирними лініями показані такти кожного із пристроїв. Оскільки неможливо фізично організувати абсолютно однакові часові інтервали тактів у пристроях, то в певний інтервал часу передачі, біти будуть зміщені, отже послілка неправильно сприйнята (на рисунку виділено штриховою лінією).

При синхронній передачі приймач і передавач постійно синхронізуються під час передачі. Синхронізація відбувається через певні проміжки часу або даних за допомогою *синхронізуючої послілки*, яка являє собою імпульсний сигнал (набір імпульсів) певної частоти. Цей сигнал генерується або передавачем, або приймачем, або окремим пристроєм. Синхронізуючий сигнал може передаватися по спеціальному проводу (канал) або разом з цифровими даними (наприклад, RZ-кодування, манчестерське кодування), або перед ними (наприклад, у вигляді *пreamбули*). В першому випадку необхідний допоміжний провід (канал), в другому — допоміжна смуга пропускання, в третьому — допоміжні біти.

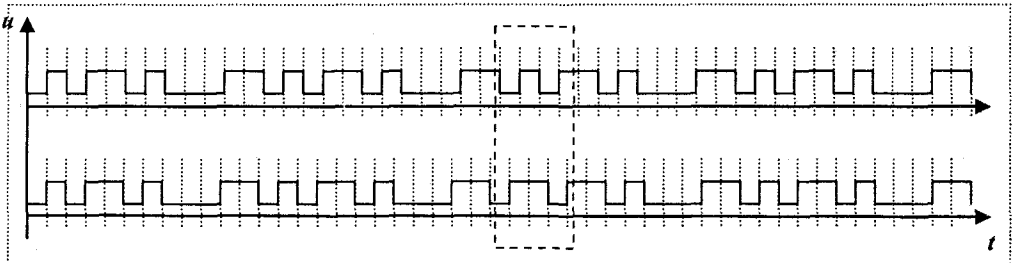


Рис. 3.20. Обмін інформацією між пристроями без синхронізації:
зверху — передавач, знизу — приймач

При *асинхронній* передачі обмін даними може відбуватись і без визначеного джерела часу. Передача починається у будь-який момент часу із сигналу початку передачі — стартового біта. Саме тоді приймач синхронізується з передавачем. Обмін проходить короткими наборами бітів (символами), що дозволяє не робити додаткову синхронізацію під час передачі. При асинхронній передачі приймач повинен наперед знати всі параметри зв'язку, в першу чергу — швидкість, щоб правильно ідентифікувати сигнали, які надходять.

3.7.2. Символьна передача

Один з асинхронних способів передачі є *символьна передача*. При такому способі дані які передаються діляться на *символи* по кілька біт (як правило, по 8 або 7) і обрамляються службовими бітами. На рис. 3.21 графічно представлено передачу двох байт $D8_{16}(11011000_2)$ та $C4_{16}(11000100_2)$ при таких настройках: біт парності — «непарний», 1 — стоповий біт, бітова швидкість 9600 біт/с. За відсутності передачі на лінії утримується логічна «1». При передачі символу передавач скидає сигнал на «0» і тримає його протягом 1-го біта, який називають *стартовим бітом*. За цей час приймач синхронізується з передавачем і готується до прийому *бітів даних*, які передаються відразу після стартового біта. Потім передається *біт парності (біт паритету — parity bit)*, якщо він використовується при обміні. Даний біт призначений для контролю за правильністю передачі даних і вибирається з ряду: парний (even), непарний (odd), відсутній (none). При виборі останнього на фізичному рівні не проводиться контроль помилок. Якщо використовується *парний (even)* біт паритету, то при передачі підраховується кількість одиничних бітів даних, і якщо їхня кількість непарна, то добавляється біт паритету, рівний логічній «1», в протилежному випадку добавляється логічний «0». При *непарному (odd)* паритеті — навпаки, передавач буде добавляти до бітів даних логічну «1» або «0» так, щоб сума одиничних бітів у бітах даних разом з паритетним була непарною. Приймач перевіряє суму прийнятих одиничних бітів даних та паритету і, якщо вона не співпадає з типом наперед визначеного біта паритету, сигналізує про це верхнім рівням, які вирішують можливість повторного запиту.

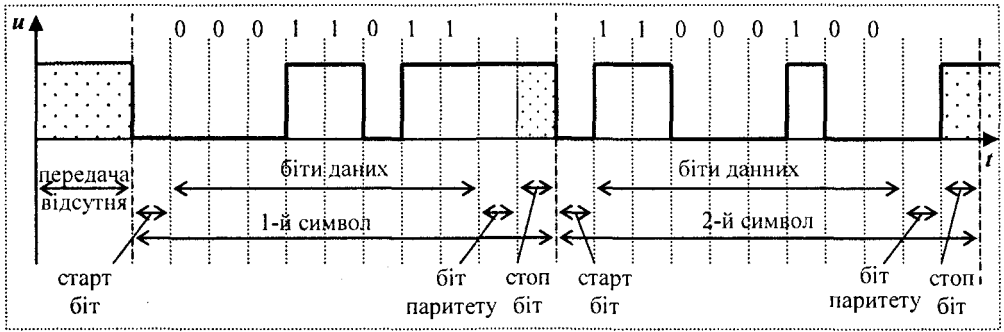


Рис. 3.21. Передача двох символів:

1-й — 11011000; 2-й — 00100011. Біт паритету — непарний; 1 — стоповий біт

За бітом паритету йдуть *стопові біти*, призначення яких витримати мінімальну паузу між символами. Кількість стопових бітів, як правило, вибирається 1, 1,5 або 2. За стоповими бітами може відразу йти наступний символ, тобто стартовий біт наступного символу.

3.7.3. Використання схем UART

Символьну передачу просто і недорого реалізувати завдяки спеціальній мікросхемі — універсальному асинхронному приймачу-передавачу УАПІ (*UART* — Universal Asynchronous Receiver Transmitter), яка застосовується в багатьох пристроях. UART є частиною інтерфейсу між шиною мікропроцесора і трансивером каналу зв'язку. Тобто, вихід UART безпосередньо не підключається до каналу зв'язку, а видає біти на рівні ТТЛ на вхід трансивера. На рис. 3.22 показаний приклад використання UART разом з трансивером RS-232.

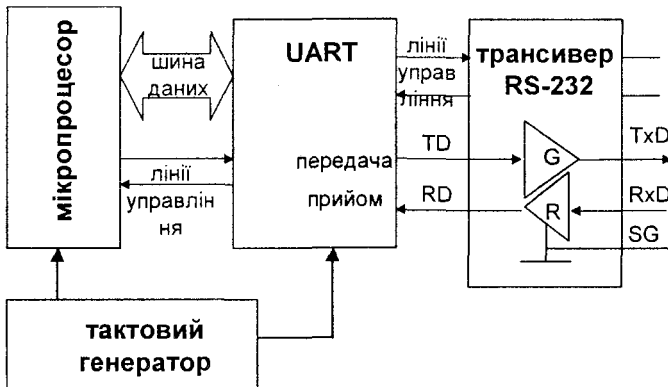


Рис. 3.22. Приклад використання UART з трансивером RS-232

Різні схеми за типом UART використовуються також у синхронній передачі даних і називаються USRT, а їх комбінація — **USART**.

При передачі UART послідовно виконує такі дії: встановлює для трансивера бітову швидкість; приймає від мікропроцесора біти даних символу через паралельну шину; генерує для трансиверу стартовий біт; послідовно передає трансиверу біти даних; визначає та передає трансиверу біт парності; передає трансиверу стопові біти; сигналізує процесору про готовність до наступного символу; за необхідності координує квітування.

При прийомі від трансивера UART послідовно виконує такі дії: встановлює для трансивера бітову швидкість; розпізнає стартовий біт; зчитує біти даних, що надходять від трансивера у вигляді послідовності бітів; зчитує біт парності і робить перевірку на парність; передає символ у паралельному вигляді мікропроцесору; при необхідності координує квітування; перевіряє дані на наявність помилки і при її знаходженні мітить у регістрі станів біт помилки.

Крім сигналів передачі (TD), прийому (RD) та шини даних, у схемі UART також визначені додаткові управляючі сигнали для синхронізації як з мікропроцесором, так і з трансивером.

3.8. Стандартні послідовні інтерфейси

На фізичному рівні більшість промислових мереж використовують один із стандартних послідовних інтерфейсів: RS-232, RS-422, RS-485 або CurrentLoop («струмова петля»). Використання цих інтерфейсів пов'язано з відносною дешевиною організації зв'язку (існують готові мікросхеми з їх реалізацією) та їх популярністю.

Один із найстаріших інтерфейсів — RS-232 — поступово зникає з комп'ютерної техніки, про що свідчить його відсутність на портативних комп'ютерах (COM-порт). На заміну йому прийшов USB, який практично витіснив із ПК не тільки цей інтерфейс, а й інтерфейс паралельного порту. Але в галузі автоматизації інтерфейс RS-232 досі використовується поряд з іншими.

Інтерфейс RS-422A в чистому вигляді мало де застосовується. Замість нього частіше використовують RS-485 у дуплексному режимі, оскільки вони сумісні. RS-485 поряд з RS-232 знайшов найбільше застосування і, як правило, підтримується в мікропроцесорних засобах автоматизації.

Інтерфейс радіальний послідовний — ІРПС, аналог зарубіжного CurrentLoop, був дуже популярним у вітчизняних (радянських) засобах автоматизації, зокрема він реалізований у контролерах ЛОМІКОНТ та РЕМІКОНТ. Сьогодні цей інтерфейс мало де використовується в сучасних засобах автоматизації.

3.9. Інтерфейс RS-232

Стандарт інтерфейсу **RS-232C** розроблений у 1969 році і опублікований асоціацією EIA як варіант «С» рекомендованого стандарту (RS-Recommended Standard) номер 232. Інтерфейс призначений для підключення апаратури, яка пе-

редає або приймає дані АПД (*DTE* — Data Terminal Equipment) до апаратури каналів даних АКД (*DCE* — Data Communication Equipment). Кінцева мета — це з'єднання двох *DTE* (рис. 3.23, *а*). Найпростіший приклад такої системи, коли в ролі *DTE* виступає комп'ютер, в якому RS-232 реалізований в якості COM-портів, а в ролі *DCE* — модем (рис. 3.23, *б*).

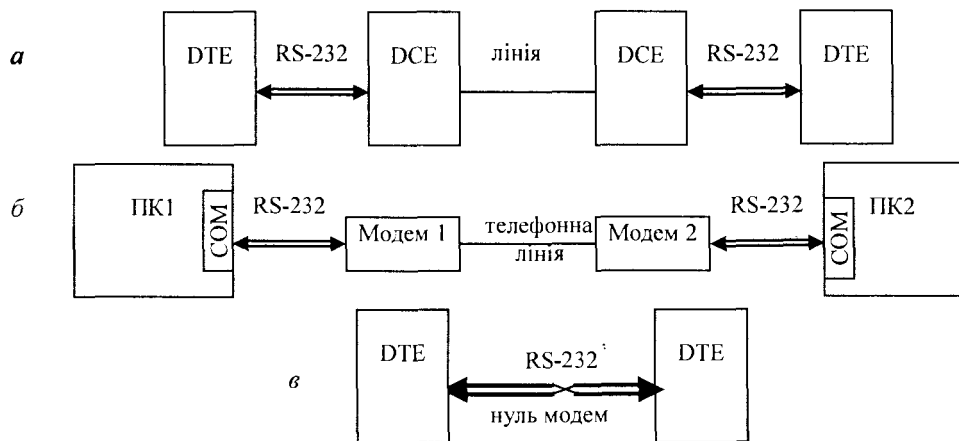


Рис. 3.23. З'єднання за допомогою RS-232:

а — схема з'єднання *DTE* з використанням *DCE*; *б* — приклад з'єднання двох ПК (як *DTE*) через модем (як *DCE*); *в* — схема з'єднання двох *DTE* без *DCE* (нуль-модемний зв'язок)

На практиці великої популярності набув спосіб з'єднання через RS-232 двох *DTE* без *DCE*, який отримав назву *нуль-модемного з'єднання* (*Zero-modem* або *Z-modem*).

Стандарт описує управляючі сигнали інтерфейсу, обмін даними, електричний інтерфейс і типи роз'ємів. Передбачені синхронний та асинхронний режими обміну (в комп'ютері COM-порт підтримує тільки асинхронний режим). Функціонально RS-232C еквівалентний стандарту МККТТ V.24 і V.28.

3.9.1. Електричні характеристики

Стандарт RS-232C описує несиметричні (незбалансовані) передавачі та приймачі, тобто сигнал передається відносно загального проводу — схемної землі за допомогою напруги. Для передачі використовується сигнальна лінія TD — Transmit Data (інколи позначається TxD), а для прийому — RD — Receive Data (інколи позначається RxD), загальний провід SG (Signal Ground). Логічний «1» на вході відповідає діапазон напруги від -12В до -3В ; логічному «0» — від $+3\text{В}$ до $+12\text{В}$. Діапазон від -3В до $+3\text{В}$ — зона нечутливості, яка обумовлює гістерезис приймача: логічний стан лінії поміняється тільки після переходу через поріг (рис. 3.24). Рівні сигналів «1» і «0» на виходах передавачів повинні бути в межах відповідно від -5В до -12В і від $+5\text{В}$ до $+12\text{В}$.

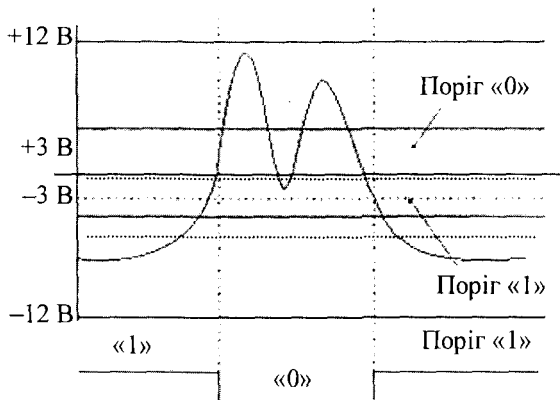


Рис. 3.24. Прийом сигналів в RS-232С

Інтерфейс не забезпечує гальванічної розв'язки пристроїв. Різниця потенціалів між схемними землями SG-пристроїв, що з'єднуються, не повинна бути більше 2В. Підключення і відключення інтерфейсних кабелів пристроїв з автономним живленням повинно проводитись при відключеному живленні. Інакше в момент підключення пристроїв можуть вийти з ладу мікросхеми в результаті дії на них різниці потенціалів.

Стандартні трансивери RS-232 повинні забезпечити передачу та прийом бітів зі швидкістю до 115200 біт/с на відстані до 20 м.

3.9.2. Типи роз'ємів

На обладнанні DTE прийнято встановлювати ви́лки типу DB-9P (9 штиркові) або DB-25P (25 штиркові). Останні, як правило, використовуються для синхронних режимів, оскільки мають допоміжні штирки. На пристроях DCE використовуються розетки DB-9S та DB-25S. Це значить, що при такому з'єднанні пристроїв DTE (наприклад ПК) та DCE (наприклад, модем) можна підключати безпосередньо один до одного або через «прямий» кабель з ви́лкою з одного боку та розеткою з іншого (рис. 3.25).

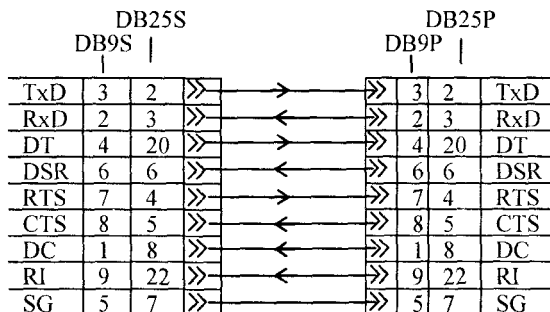


Рис. 3.25. Підключення DTE (зліва) до DCE (справа)

У системах автоматизації частіше використовується нуль-модемне з'єднання, для якого застосовується нуль-модемні кабелі. Якщо при з'єднанні використовуються тільки інформаційні сигнальні лінії (TxD, RxD, SG), то з'єднання проводиться *мінімальним* нуль-модемним кабелем (рис. 3.26, б), у противному разі — *повним* (рис. 3.26, а).

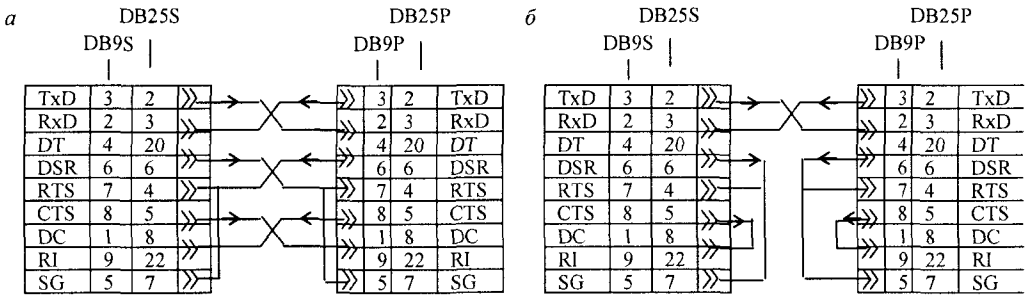


Рис. 3.26. Нуль-модемне підключення за допомогою нуль-модемного кабелю:

а — повного, б — мінімального

3.9.3. Призначення ліній інтерфейсу RS-232C

На практиці інтерфейс RS-232 використовується, як правило, в асинхронному режимі, тому розглянемо сигнальні лінії, призначені тільки для цього режиму. Для кращого розуміння при розгляді ліній інтерфейсу DCE будемо називати модемом, а DTE — комп'ютером.

У табл. 3.2 наведені сигнальні лінії RS-232C та їх призначення. В другій та третій колонках вказані відповідні контакти роз'ємів, а в четвертій — також міститься додаткова інформація, яка вказує напрямок передачі даних по лінії: IN — у комп'ютер, OUT — з комп'ютера.

Таблиця 3.2

СИГНАЛЬНІ ЛІНІЇ ІНТЕРФЕЙСУ RS-232C

Сигнал	DB-25S	DB-9S	Призначення
PG	1	—	Захисна земля (Protected Ground). З'єднується з корпусом пристрою і екраном кабелю.
SG	7	5	Сигнальна земля (Signal Ground), відносно неї діють лінії сигналів.
TD(TxD)	2	3	OUT Вихід передавача (Transmit Data).
RD(RxD)	3	2	IN Вхід приймача (Receive Data).
RTS	4	7	OUT-Запит дозволу на передачу (Request To Send). «Вкл.» означає наявність даних у комп'ютера для передачі. При напівдуплексі — перемикання модема в режим передачі.
CTS	5	8	IN Готовність передачі (Clear To Send). «Відкл.» — модем забороняє комп'ютеру передавати йому дані.

Сигнал	DB-25S	DB-9S	Призначення
DTR	20	4	OUT — Готовність DTE (Data Set Ready). «Вкл.» — комп'ютер готовий до роботи з модемом.
DSR	6	6	IN — Готовність DCE (Data Terminal Ready). «Вкл.» — модем готовий до роботи з комп'ютером.
DCD	8	1	IN — Виявлення несучої (Data Carried Detected). «Відкл.» — модем сигналізує про сигнал поганої якості
RI	22	9	IN — Сигнал виклику (Ring Indicator). «Вкл.» — модем отримав сигнал виклику (дзвінок на телефонній лінії)

3.9.4. Управління потоком даних

При асинхронному режимі у поєднанні з трансивером RS-232 використовуються стандартні схеми UART із символьним способом передачі. Бітова швидкість при цьому вибирається з ряду: 50, 75, 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600 і 115200 біт/с. Кількість біт даних може становити 5, 6, 7 або 8 (5- і 6-бітні формати практично не використовуються). Кількість стопових бітів може бути 1, 1,5 або 2. Для управління потоком даних (Flow Control) можуть використовуватися два варіанти протоколу — апаратний та програмний.

Апаратний протокол управління потоком (hardware flow control) RTS/CTS використовує сигнал RTS, який подається на CTS, що дозволяє зупинити передачу даних, якщо приймач не готовий до прийому (рис. 3.27). Передавач видає черговий байт тільки при увімкненій лінії CTS. Але байт, який уже почав передаватися, затримуватися не буде — це гарантує цілісність посилки. Апаратний протокол забезпечує найшвидшу реакцію передавача на стан приймача. При зв'язку двох пристроїв DTE повним нуль-модемним кабелем (рис. 3.26, а) необхідно перехресно з'єднати RTS передавача та CTS приймача, а при мінімальному (рис. 3.26, б) RTS та CTS перемикаються на тому ж пристрої. У протилежному разі при налаштуванні апаратного управління потоком деякі типи передавачів «мовчатимуть», оскільки на CTS не буде дозволу передачі. Апаратний протокол DTR/DSR аналогічний попередньому, але використовує другу пару сигналів.

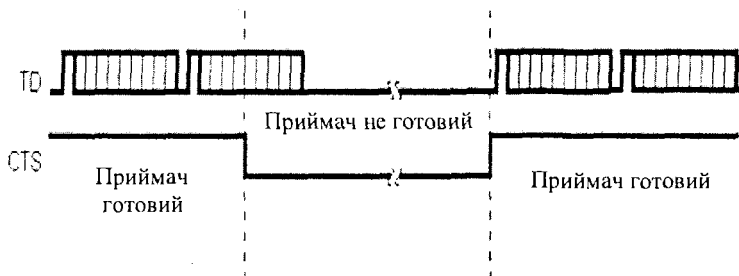


Рис. 3.27. Апаратне управління потоком даних RTS/CTS

Програмний протокол управління потоком (software flow control) XON/XOFF передбачає наявність дуплексного каналу передачі даних (рис. 3.28). Працює він так: якщо пристрій, який приймає дані, з якихось причин не може їх більше приймати, він відсилає передавачу байт-символ XOFF (ASCII код 13₁₆), по отриманню якого останній припиняє передачу. Коли приймаючий пристрій знову стає готовим до приймання даних, він відправляє символ XON (ASCII код 13₁₆), прийнявши який, протилежний пристрій продовжує передавати дані. Час реакції передавача на зміну стану приймача, порівняно з апаратним протоколом, збільшується за рахунок часу передачі символу та часу реакції програми передавача на прийом символу. Перевагою програмного протоколу є відсутність необхідності в передачі управляючих сигналів інтерфейсу, тобто мінімальний кабель для двостороннього обміну може мати тільки 3 проводи (рис. 3.26, б) навіть без перемикання управляючих проводів на пристроях. Недоліком, окрім часових запізнь та наявності допоміжного буфера, є складність реалізації дуплексного режиму обміну, тому що при зворотній передачі даних повинні виділятися і оброблятися управляючі символи, що обмежує набір символів, які можуть передаватися.

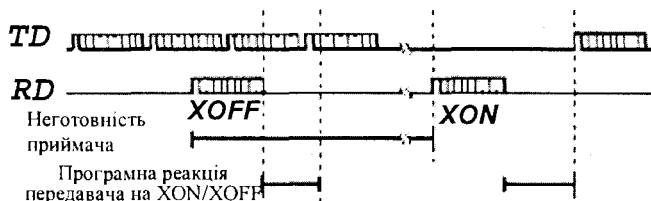


Рис. 3.28. Програмне управління потоком даних XON/XOFF

Функціонально нуль-модемне з'єднання двох трансиверів можна показати у вигляді, представленому на рис. 3.29. На схемі показані тільки сигнальні лінії для передачі даних (без синхронізуючих і управляючих ліній). Дані, які надходять для передачі, позначені як TD, для прийому — як RD, сигнальна земля (загальний) — як SG.

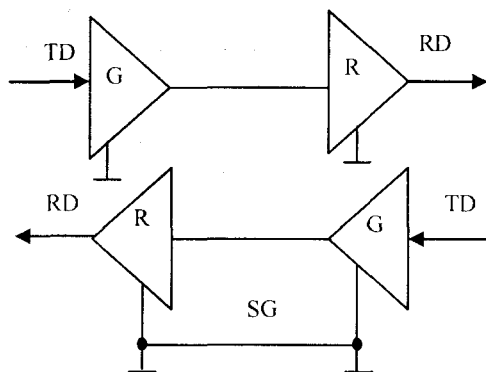


Рис. 3.29. Функціональна схема нуль-модемного з'єднання

Апаратно один передавач даного інтерфейсу може забезпечити необхідним рівнем сигналу лише один приймач. Таким чином, інтерфейс RS-232 дозволяє об'єднати між собою тільки два пристрої з одним передавачем і одним приймачем на максимальній відстані 20 м.

3.10. Інтерфейс RS-422A (EIA-422A)

Інтерфейс RS-232 має ряд недоліків, які обмежують його використання в промислових умовах. Насамперед, це пов'язано з чутливістю до електромагнітних завад, малою довжиною кабелю з'єднання і малою бітовою швидкістю. Організація EIA випустила ряд нових стандартів, один з яких **RS-422A**, під назвою «Electrical Characteristics of Balanced Voltage Digital Interface Circuits» (Електричні характеристики ланцюгів цифрових інтерфейсів із симетричною напругою»).

Даний стандарт описує (рекомендує) тільки електричні характеристики інтерфейсу, тобто вимоги до передавачів (драйверів — drivers) та приймачів і не визначає параметри сигналів, типи роз'ємів, типи і довжину кабелів та ін. Інтерфейс базується на збалансованих (симетричних) лініях передавачів та приймачів, тобто напруга передається по трьох проводах — два сигнальні та сигнальна земля.

На рис. 3.30 показаний принцип функціонування збалансованого передавача (а) та приймача (б) лінії. Збалансований передавач лінії при передачі на TD логічної «1», формує напругу між сигнальним проводом «А» і землею «С» — U_A та протилежну за знаком напругу між проводами «В» і «С» — U_B (на рис. 3.30 показаний знак інверсії). При передачі логічного «0» передавач інвертує знак сигналу. Приймач вимірює напругу між «А» і «В», і, якщо різниця буде більшою, ніж +200 мВ, то приймач буде розпізнавати його як певний логічний стан лінії. Якщо різниця змінить свій знак і стане меншою, ніж -200 мВ, приймач змінить логічний стан на протилежний.

Якщо на проводі «А» передавача в інтерфейсі RS-422A відносно проводу «В» від'ємна напруга, то лінія перебуває в стані логічної «1», якщо ж додатна — логічного «0». Позначення проводу «А», як правило, еквівалентне «-», а позначення «В» — «+» (інколи приймають навпаки). RS-422A витримує синфазну напругу $\pm 7\text{В}$ (середня напруга на виходах А і В відносно землі).

Диференційний спосіб передачі зменшує вплив синфазної завади на лінію передачі, оскільки вимірюється не сигнал між сигнальним проводом і землею, а різниця потенціалів між двома сигнальними лініями. Навіть якщо на обох проводах будуть наведені паразитні струми, диференційна напруга практично не зміниться.

Стандарт також описує електричні вимоги до передавачів та приймачів. Зокрема, один передавач повинен бути розрахований на 10 приймачів. Інтерфейс працює в дуплексному режимі. Тому можливі два варіанти з'єднання: передавачі та приймачі двох вузлів з'єднуються перехресно (точка-точка); передавач одного вузла обслуговує до 10 приймачів інших вузлів.

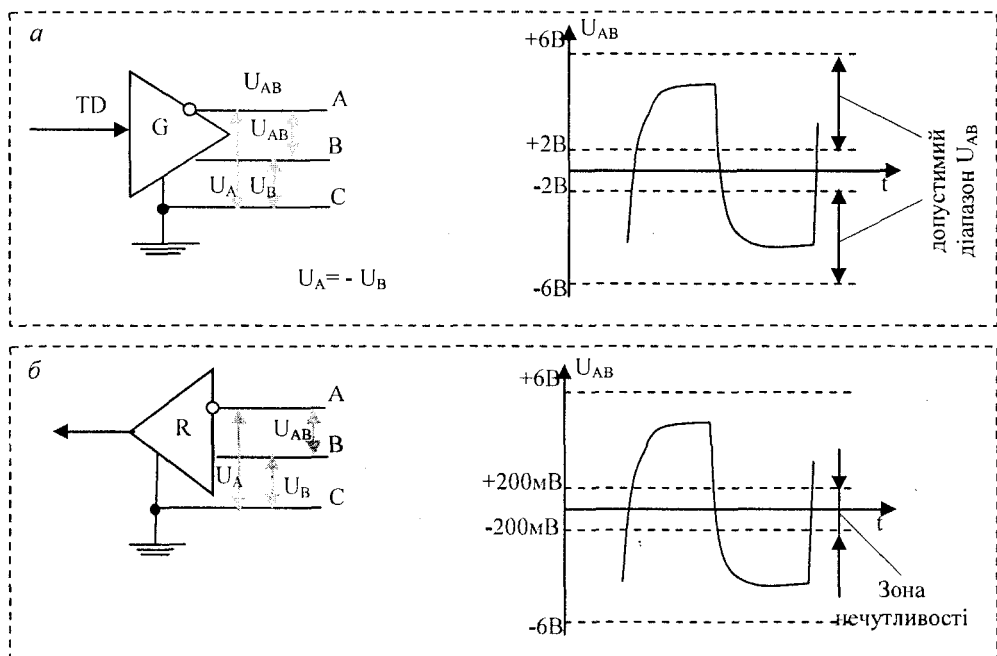


Рис. 3.30. Функціональна схема передавача (а) та приймача (б) в інтерфейсі RS-422A

На основі даного стандарту з'явилися деякі рекомендації щодо його використання. Це схеми підключення, способи заземлення, максимальні відстані, використання резисторів-термінаторів, максимальні бітові швидкості і т.д. Так, установлено, що максимальна бітова швидкість становить близько 10 Мбіт/с, максимальна відстань (при невеликих швидкостях) — близько 1200 м. Ці величини залежать від багатьох параметрів: типу кабелю, рівня завад, типу роз'ємів і т.д. В якості середовища передачі на великі відстані рекомендують використовувати виту пару, бажано екрановану. На рис. 3.31 показаний один із способів реалізації дуплексного з'єднання двох пристроїв за RS-422A.

Інтерфейс RS-422A має один значний недолік — на ньому не можна побудувати мережі із шинною топологією, тому він не знайшов такого широкого застосування як RS-485. Чому ж інтерфейс, в якому передавач може забезпечити струмом 10 приймачів, не може працювати на шині разом з іншими передавачами? Проблема в тому, що передача інформації забезпечується логічними «0» та «1» які мають відповідний рівень фізичного сигналу за напругою. Наприклад, у символному режимі, за відсутності передачі на лінії приймача, повинна бути логічна «1», яку передає трансмітер. Тобто в кожен момент часу трансмітер буде прикладати напругу певного рівня. Якщо декілька передавачів буде підключено до однієї лінії зв'язку, то це може привести до виходу з ладу трансиверів, оскільки кожен з передавачів буде генерувати свої сигнали. Цей недолік відсутній у інтерфейсі RS-485.

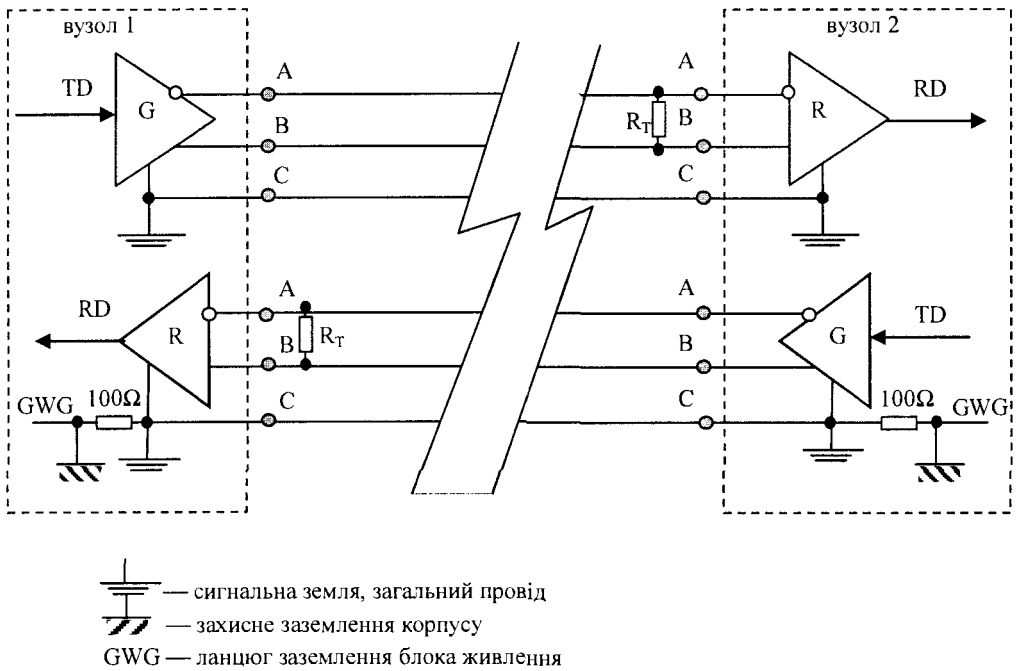


Рис. 3.31. Функціональна схема 5 — провідного дуплексного з'єднання двох вузлів за RS-422A

3.11. Інтерфейс RS-485 (EIA/TIA-485)

Даний стандарт спільно розробили дві асоціації EIA і TIA. Оскільки всі свої стандарти EIA маркувала як RS, то популярна назва стандарту **RS-485**, однак офіційна — **EIA/TIA-485**, що ідентифікує дійсне походження даного стандарту. Нині різноманітні розширення стандарту RS-485 (надалі будемо називати його так) охоплює широку різноманітність доповнень.

Все, що стосується електричних характеристик стандарту RS-422A, справедливо і для RS-485: використовується збалансована система зв'язку з тими ж рівнями сигналу, за винятком синфазної напруги, яку витримує інтерфейс — від +12В до -7В. Тому даний інтерфейс сумісний зі своїм попередником (зворотна сумісність не підтримується).

На відміну від свого попередника RS-422A, інтерфейс RS-485 розрахований на багатоточкове з'єднання, тобто на побудову шинних топологій. Перш за все, на трансиверах, крім інформаційних сигналів RD та TD, з'явився додатковий сигнал управління передачі/прийому (рис. 3.32). Тепер, по закінченні передачі, пристрій може відключити свій передавач (перевести його у високоімпедансний стан) і надати можливість іншим трансмітерам підключатися для передачі.

Іншими словами сигнал дозволу передачі переводить трансмітер з активного стану в пасивний, тобто передавач може генерувати логічну «1», логічний «0» або перебувати в пасивному стані. В який час і в якому порядку передавачі вузлів будуть доступатися до шини стандарт не визначає, дане питання повинне вирішуватись на канальному рівні (організація доступу до шини). Реалізація інтерфейсу повинна забезпечити працездатність обладнання при короткочасних колізіях (коли два передавачі займають одночасно шину), що, до речі, теж не описано в стандарті.

При розробленні даного стандарту розраховували на більшу кількість можливих приймачів. Один трансмітер розрахований на 32 одиниці навантаження, які, в свою чергу, визначаються як вхідний імпеданс (опір) одного стандартного приймача (12 кОм). Таким чином, стандартний передавач може обслуговувати 32 приймачі з опором 12 кОм, які підключені до загальної шини паралельно. Якщо приймачі мають більш високий опір, наприклад, 48 кОм ($4 \cdot 12$ кОм), то до однієї шини можна підключити відразу 128 таких приймачів, при збільшенні опору до 96 кОм — 256 приймачів і т.д. Слід застерегти, що збільшення імпедансу приймача робить систему чутливою до завад.

3.11.1. Схеми з'єднання

На рис. 3.32 показана функціональна схема з'єднання 3-х пристроїв до загальної шини через RS-485 інтерфейс. Напівдуплексна система дозволяє в один момент часу вузлу передавати інформацію, а в інший — приймати. Передавач та приймач кожного вузла підключаються до єдиної шини, тобто контакт «A» кожного передавача та контакт «A'» кожного приймача (штрихом позначені контакти приймачів) підключаються до загальної лінії «A», так само і контакти «B» та «B'» підключаються до лінії «B». Таким чином, коли передавач одного вузла передає послідовність бітів, усі приймачі інших вузлів їх приймають. В один момент часу тільки один передавач може бути активним, його стан управляється з боку самого пристрою дозволом передачі.

RS-485 не визначає вимоги до середовища передачі. Як правило, використовують екрановану виту пару. На рис. 3.32 показаний один з варіантів 3-х провідної схеми, однак на практиці нерідко використовують 2-х провідну (без сигнальної землі). Таке з'єднання дозволяється, якщо різниця потенціалів між «землями» вузлів не перевищує певний діапазон. Для реалізації двохпровідної схеми з'єднання достатньо однієї вити пари, трьохпровідної — подвійної вити пари (пара сигнальних проводів, пара земляних).

Існує декілька схем реалізації з'єднань через RS-485 інтерфейс. Їх розгляд виходить за рамки даного посібника. Бітова швидкість вибирається залежно від сумарної довжини лінії, характеристик кабелю і, як правило, приводиться в документації до обладнання у вигляді номограм.

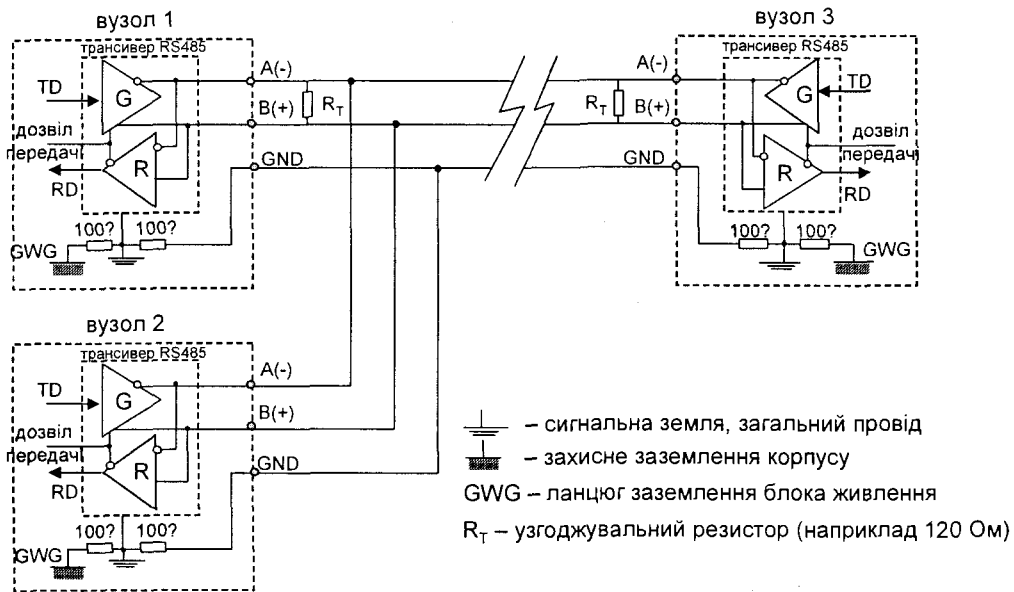


Рис. 3.32. Приклад функціональної схеми трьох провідного напівдуплексного з'єднання вузлів по RS-485

3.11.2. Схеми узгодження

При великих швидкостях і при значних відстанях необхідно вирішити проблему відбиття сигналу на кінцях лінії. Нагадаємо, що основою для мінімізації відбиття є використання узгоджувальних резисторів з номіналами, які відповідають хвильовому опору кабелю. Як правило, в промислових мережах використовують кабель з характеристичним імпедансом 120 Ом, тому на схемі на рис. 3.32 на обох кінцях шини між лініями «A» і «B» підключений узгоджувальний резистор (термінатор) з відповідним опором.

На практиці виникає необхідність паралельного підключення до шини джерела постійного струму (розглянуто нижче). Резистори чинять активний опір, тому розсіюють певну потужність, що вимагає використання більш потужних джерел струму. Це спонукає до пошуку альтернативних схем узгодження.

Додавання конденсатора послідовно з резистором виключає протікання постійного струму і не створює бар'єра для високочастотного сигналу (рис. 3.33, а). Такий спосіб вимагає клопіткого підбору ємності, щоб уникнути ефекту відбиття. Альтернативний варіант зменшення споживаної потужності — використання діодів Шотки.

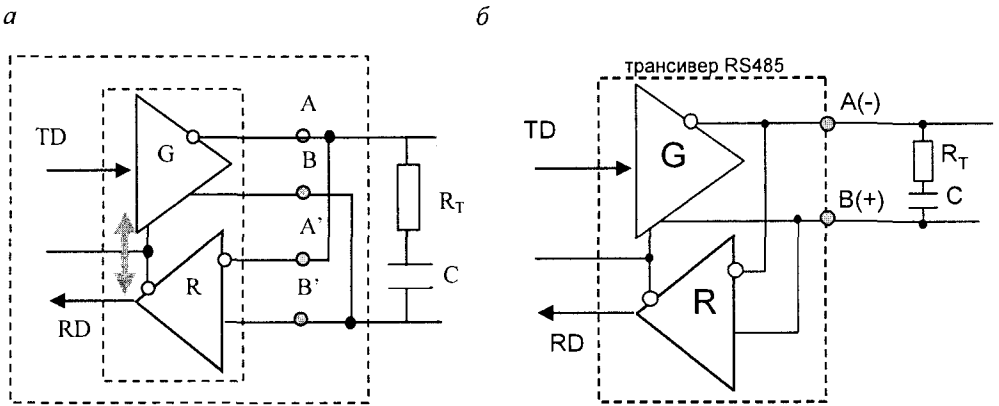


Рис. 3.33. Альтернативна схема RC узгодження

3.11.3. Забезпечення захисного зміщення

Реалізація інтерфейсу передбачає, що за відсутності бітів для передачі трансмітер переходить у високоімпедансний стан. При сигнальній «тиші» всі трансивери на шині будуть працювати в режимі прийому. Оскільки жоден трансмітер не генеруватиме сигнал, на шині буде 0 В. Для більшості мереж, зокрема для тих, які працюють у символьному режимі, відсутність передачі обов'язково позначається певним рівнем сигналу, наприклад, логічною «1». Однак при напрузі на входах приймачів від -200 мВ до 200 мВ вихідний стан залишається невизначеним. У цьому випадку логічна «1» або «0» на виході RD приймачів рівноймовірні (або запам'ятовують попередній стан). Крім того паразитні струми, що можуть наводитись в кабелі, можуть створити достатню напругу на приймачі, щоб сформувати стан логічного сигналу.

Така ситуація неприпустима. В момент інформаційної тиші необхідно тримати сигнал напруги певної величини за допомогою зовнішнього джерела постійного струму, який підключається між сигнальними проводами А та В, тобто необхідно забезпечити зміщення потенціалу відносно нуля.

Всю лінію можна представити як замкнутий ланцюг із опорів ресиверів та двох термінаторів, що підключені паралельно. Захисне зміщення має забезпечити цей ланцюг таким струмом, щоб напруга на лінії була в межах, що відповідає логічній «1», тобто > 200 мВ. Здебільшого підключають джерело напруги 5 В і підбирають такі шунтуючі опори, щоб забезпечити ланцюг необхідним струмом. Ці опори отримали назву *резисторів захисного зміщення (bias resistors)*: резистор завдання початкового високого рівня (*pullup*) — на одній лінії, наприклад «В», і низького рівня (*pulldown*) — на другій лінії, наприклад «А» (рис. 3.34). Значення опору резистора зміщення залежить від виду узгодження (термінування) і кількості вузлів, тобто приймачів, підключених до шини. Номінали опорів зміщення розраховують, виходячи з необ-

хідного мінімального струму зміщення в мережі для забезпечення різниці потенціалів між A і $B > 200$ мВ у такій послідовності:

1. Розраховується загальний опір лінії, який складається з опору всіх приймачів (як правило, 12 кОм кожний) та опору термінаторів (як правило, 120 Ом).

2. Розраховується мінімальний струм для 200 мВ: $I_{зв} = U_{зв} / R_l$ ($U_{зв} = 200$ мВ, R_l — обчислений у п. 1);

3. Розраховується необхідний сумарний опір лінії: $R_{ш} = U_{жс} / I$, де $U_{жс}$ — напруга живлення зміщення (як правило, 5В), $I_{зв}$ — мінімальний струм з п.2;

4. Визначається номінал резисторів зміщення: $R_{зм} = (R_{ш} - R_l) / 2$.

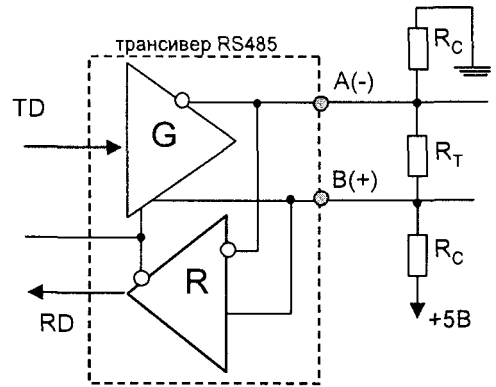


Рис. 3.34. Використання резисторів захисного (R_C) зміщення разом з узгоджувальними резисторами (R_T)

Приклад 3.3. Реалізація фізичного рівня. Розрахунок номіналів резисторів зміщення.

Завдання. Розрахувати номінал резисторів захисного зміщення для мережі на основі RS-485, яка має 10 вузлів на шині (по 12 кОм кожний приймач) і 2 узгоджувальні резистори по 120 Ом.

Рішення.

1. 10 приймачів опором 12 кОм, з'єднаних паралельно та 2 термінатори дають сумарний опір:

загальна провідність = $1/12000 \cdot 10 + 1/120 \cdot 2 = 21/1200$ См;

загальний опір = 57 Ом.

2. Мінімальний необхідний струм:

$I_{зв} = 200\text{мВ} / 57 \text{ Ом} \approx 3.5$ мА.

3. Необхідний опір лінії не повинен перевищувати

$R_{ш} = 5\text{В} \cdot 2 / 3.5 \text{ мА} \approx 1428$ Ом.

4. Віднімемо 57 Ом, які вже в нас є, і отримаємо:

$1371 \text{ Ом} / 2 = 685$ Ом для кожного резистора.

Приклад 3.3

Резистори захисного зміщення можуть бути встановлені на будь-якому вузлі мережі або бути розподіленими між декількома вузлами. Опір усіх резисторів захисного зміщення в системі повинен бути меншим чи дорівнювати розрахованому значенню. Надмірно високий їх опір приводить до недостатнього зміщення, а отже до неправильної роботи системи, малий — до підвищення навантаження на передавачі.

3.12. Інтерфейс «струмова петля» 20 мА (Current Loop, ІРПС)

Вище були розглянуті інтерфейси, які використовували в якості носія інформації напругу. При великих відстанях краще використовувати струм. Коротко

зупинимося ще на одному популярному інтерфейсі, який назвали: 20 mA Current Loop (CL, струмова петля) і ІРПС 20 мА (Інтерфейс Радиальный Последовательный).

Для передачі логічного «0» в інтерфейсі використовують струм 0 мА, логічної «1» — 20 мА, тому в назву інтерфейсу додають «20 мА». Оскільки передача ведеться символами, то, коли передавач мовчить, він виставляє на лінію логічну «1». Таким чином, довготривалий сигнал 0 мА ідентифікується як розрив.

Струмова петля передбачає дуплексне з'єднання. На рис. 3.35 умовно зображена функціональна схема такого з'єднання в тому форматі, в якому були приведені до цих пір всі інтерфейси.

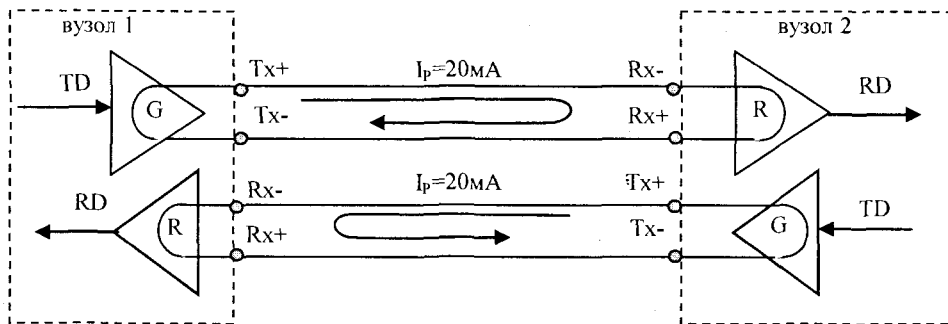


Рис. 3.35. Функціональна схема дуплексного з'єднання двох вузлів за струмовою петлею

Насправді існують схеми використання даного інтерфейсу для побудови шинних топологій. Причому максимальна довжина лінії зв'язку збільшується при збільшенні кількості вузлів у мережі.

Використовують дві схеми з'єднання пристроїв:

- з активним передавачем та пасивним приймачем;
- з активним передавачем та активним приймачем.

Відрізняють ці схеми способом живлення ліній зв'язку. Розглянемо першу схему, зображену на рис. 3.36, де генератор струму міститься в приймачах.

Сигналом TD управляється умовний контакт, який замикає або розмикає лінію. Оскільки струм йде по шляху найменшого опору, то при замиканні лінії він не буде циркулювати через приймач іншого вузла. В протилежному разі, якщо контакт розімкнений, по лінії піде струм 20 мА, який пройде через світлодіод приймача. Останній використаний разом із світлочутливим елементом (скажімо, фотодіодом) в якості гальванічної розв'язки і називається **оптопарою**. Таким чином, якщо контакт вузла 1 розімкнений — у світлочутливому елементі вузла 2 наведеться струм, і приймач видасть на RD логічну «1». Принцип передачі в зворотному напрямку аналогічний.

В іншій схемі з'єднання, яка показана на рис. 3.37, відмінність полягає в реалізації схеми приймача вузла 1 і передавача вузла 2. Як бачимо, вузол 1 має активний передавач і активний приймач, тобто обидва живлять лінії зв'язку.

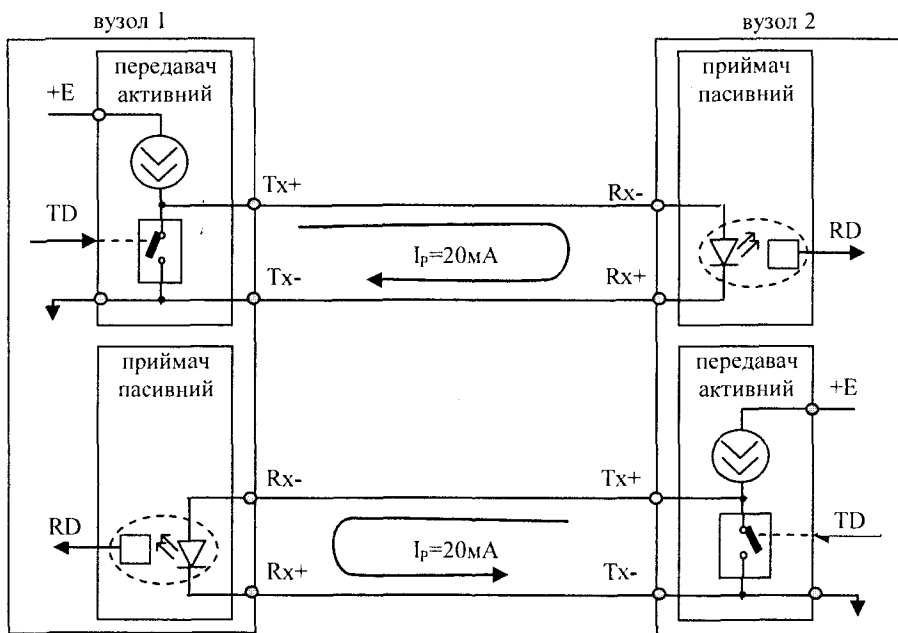


Рис. 3.36. Використання схеми типу активний передавач — пасивний приймач

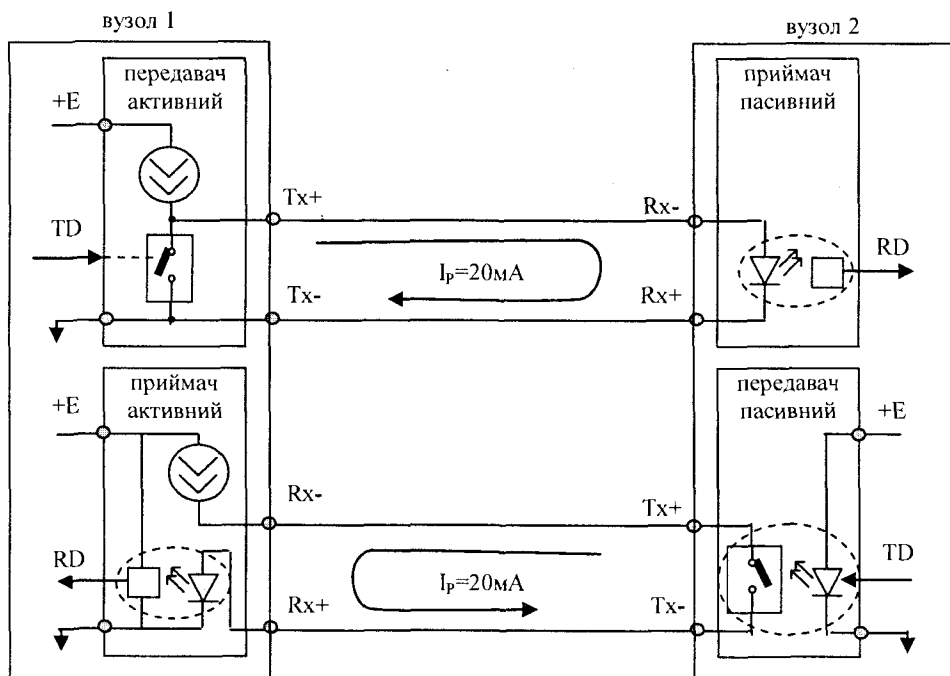


Рис. 3.37. Використання схеми типу активний передавач — активний приймач

У табл. 3.3 наведена залежність максимальної відстані при певній бітовій швидкості для обох згаданих вище схем. Як бачимо, при швидкості 9600 біт/с, яка найчастіше зустрічається в промислових мережах, реалізованих на стандартних послідовних інтерфейсах, можна досягнути відстані декількох кілометрів (правильніше сказати, навпаки: при значній відстані можна досягнути високих швидкостей).

Таблиця 3.3

**ЗАЛЕЖНІСТЬ МАКСИМАЛЬНОЇ ДОВЖИНИ ЛІНІЇ
ВІД ШВИДКОСТІ ПЕРЕДАЧІ**

Швидкість біт/с	Максимальна відстань (м)	
	Активний передавач/ пасивний приймач	Активний передавач/ активний приймач
1 152 000	150	50
230 400	300	150
115 200	600	300
57 600	800	400
38 400	1100	550
19 200	2000	1000
14 400	2600	1300
9 600	3400	1700
300	>3400	>1700

**3.13. Порівняльна характеристика стандартних
послідовних інтерфейсів**

Для порівняння інтерфейсів, розглянутих вище, найбільш важливі характеристики зведені в таблицю 3.4.

Таблиця 3.4

ПОРІВНЯЛЬНІ ХАРАКТЕРИСТИКИ ПОСЛІДОВНИХ ІНТЕРФЕЙСІВ

Назва, стандарт	EIA RS-232C	EIA RS-422A	EIA/TIA-485, RS-485	20 мА «струмова петля», CL, ІРПС
Тип сигналу	Напруга, асиметричний	Напруга, диференційний симетричний	Напруга, диференційний симетричний	струм
Режим з'єднання	дуплекс	дуплекс	напівдуплекс, дуплекс	дуплекс
Максимальна кількість передавачів/приймачів	1/1	1/10	32/32 (при опорі приймачів 12кОм)	1/1 у багатоточкових схемах залежно від реалізації
Максимальна відстань (приблизно)	20 м	1200 м	1200 м	декілька км
Максимальна бітова швидкість (приблизно)	115200 біт/с	10 Мбіт/с	10 Мбіт/с	1 Мбіт/с

Прокоментуємо декілька комірок таблиці. RS-485 може працювати і за 5-провідною дуплексною схемою. Вона передбачає з'єднання одного передавача (з правами Ведучого) з іншими приймачами (на вузлах з правами Ведених) за однією парою проводів. За другою парою передавачі Ведених вузлів з'єднуються з приймачем ведучого. Така схема вимагає більше проводів (відповідно, накладних затрат) і не така універсальна, але виключає колізії між «ведучим» і «веденими» і збільшує швидкість обміну між вузлами за рахунок дуплексного з'єднання. Таке з'єднання прийнятне тільки в мережах типу Ведучий/Ведені і не може бути застосована в широкомовних мережах.

При використанні RS-485 кількість приймачів рівна 32, це при опорі приймачів 12 кОм. При інших опорах, наприклад 48 кОм, максимальна кількість приймачів, відповідно, буде 128.

В інтерфейсі «струмова петля» 20 мА кількість передавачів/приймачів залежить від конкретної реалізації.

Для всіх інтерфейсів (окрім RS-232C) максимальна бітова швидкість залежить від багатьох факторів. Перш за все, — це довжина лінії зв'язку, а також тип кабелю, рівень завад, способи узгодження ліній і т.ін.

Вибір інтерфейсу для реалізації мережі залежить від багатьох факторів. Здебільшого пристрій може підтримувати тільки один визначений інтерфейс. Для зв'язку з іншими пристроями дуже ймовірно використання адаптерів-перетворювачів інтерфейсів, які розглянуті в наступному підрозділі.

При побудові шинних топологій, як правило, користуються RS-485. Інколи використовують також багатоточкове з'єднання за інтерфейсом «струмова петля». Якщо необхідне з'єднання точка-точка, то вибирають дуплексний зв'язок, оскільки за тих самих бітових швидкостей досягається майже в 2 рази більша швидкість передачі даних — за рахунок подвійного каналу. В цьому випадку можна використати RS-422A або RS-485 у дуплексному режимі (якщо обладнання дає можливість організувати таке з'єднання). RS-232 популярний при з'єднанні пристроїв з ПК.

Поряд з розглянутими інтерфейсами існує ряд інших. Деякі з них не знайшли широкого використання в промислових мережах, інші тільки почали активно впроваджуватися. Одним з найбільш перспективних інтерфейсів є **USB**, який практично витіснив COM-порт (RS-232) та паралельний порт з персональних комп'ютерів. У даній книзі стандарт USB не розглядається, однак він добре описаний у літературі з комп'ютерних технологій та комунікацій.

3.14. Використання адаптерів-перетворювачів інтерфейсів

3.14.1. Типи адаптерів-перетворювачів

Однією з проблем у галузі промислових мереж є апаратна несумісність їх складових. Розглянемо це питання більш детально.

Описані вище послідовні інтерфейси складають далеко не повний перелік інтерфейсів фізичного рівня. На фізичному рівні нерідко використовуються уніка-

льні інтерфейси. Однак, навіть для мереж, які базуються на стандартних послідовних інтерфейсах, нерідко виникає необхідність в об'єднанні пристроїв з різними реалізаціями фізичного рівня. Для вирішення такого завдання існують *адаптери-перетворювачі (конвертери) інтерфейсів*. Адаптер можна уявити у вигляді пристрою, що має два різні інтерфейси. Даний пристрій має електрично перетворити вхідні сигнали RD одного інтерфейсу у вихідні TD іншого.

Розглянемо можливі випадки з'єднань пристроїв з різними інтерфейсами:

- RS-422A <-> RS-485 — оскільки інтерфейси сумісні, то RS-485 може працювати в режимі RS-422A, але не навпаки. Для зворотного режиму необхідні допоміжні схемні рішення;

- RS-232 <-> Current Loop 20 mA — необхідний адаптер-перетворювач. Схемні рішення досить прості в реалізації, виконуються у вигляді окремого пристрою або плати в ПК;

- RS-232 <-> RS-485. Оскільки RS-485 не описує спосіб передачі даних, то в даному випадку можливі дві ситуації: перетворювач для побітової та посимвольної передачі. У першому випадку реалізувати схему перетворення доволі складно і дорого. Як правило, в RS-232 передача йде символами, управління нею займається UART, а в побітовій передачі — все залежить від реалізації протоколу фізичного рівня. В посимвольній передачі використовується адаптер-перетворювач RS-232C <-> RS-485.

Зупинимося на принципах роботи адаптерів RS-232C <-> RS-485 більш детально, оскільки вони одні з найбільш поширених у промисловості. Нагадаємо, що передавачі в системі RS-485 повинні підтримувати можливість відключення від лінії зв'язку, якщо трансивер не передає дані. Однак, передавачі RS-232 разом з UART за відсутності передачі видають логічну «1». Тобто, при звичайному схемному рішенні електричного перетворення сигналу з RS-232 в RS-485 передавач RS-485 ніколи не відключиться, що недопустимо в шинних топологіях.

Таким чином, у конвертері повинно бути забезпечене управління станом передавача RS-485. Виробники адаптерів вирішують цю проблему по-різному. Розглянемо декілька способів.

3.14.2. Управління адаптером RS-232<->RS-485 з боку RS-232

Одним із варіантів управління станом трансмітера RS-485 є використання одного із сигналів управління потоком з боку інтерфейсу RS-232C: RTS або DTR.

На рис. 3.38 показана модель перетворювача, який управляється сигналом RTS. На рисунку перетворювач умовно поділений на 2 частини з відповідними інтерфейсами. Коли пристрій із RS-232C має передавати дані, він активує передавач адаптера за допомогою RTS, виставляючи на цій лінії логічну «1». Далі він надсилає послідовність бітів у символьному вигляді по лінії TD. Перетворювач за допомогою схемних рішень відсилає цей сигнал на передавач адаптера з боку RS-485. Оскільки той уже активований, цю ж послідовність він видає на лінію

RS-485 інтерфейсу. Так буде тривати, поки пристрій з боку RS-232C не зніме «1» з RTS, що приведе до переключення передавача у високоімпедансний (неактивний) стан.

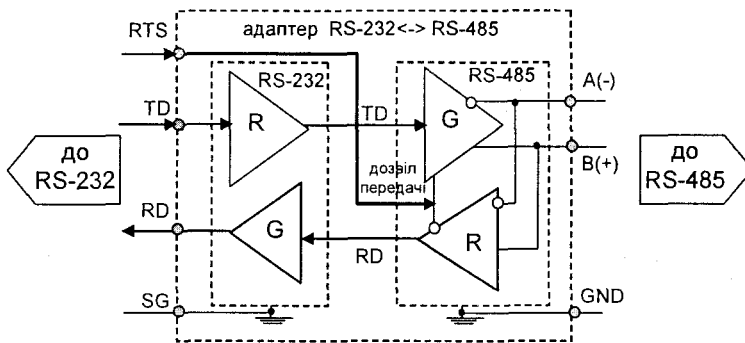


Рис. 3.38. Управління передавачем адаптера RS-232C->RS-485 сигналом RTS

Алгоритм управління сигналом RTS для RS-232C може бути різним. Для комп'ютерів, як правило, налаштовують роботу COM-порту так, щоб сигнал RTS виставлявся автоматично в «1», коли буфер передачі заповнений байтами, і «0», коли він порожній. Для COM-порту це нормальний режим роботи з апаратним управлінням потоком. Рідше використовується сигнал DTR.

При наведеному способі реалізації перетворювача можлива ситуація передчасного відключення сигналу RTS до передачі трансмітером RS-485 останнього біта. Такі проблеми повинні вирішуватись на пристрої з боку інтерфейсу RS-232C. Ще одна проблема полягає в тому, що не всі пристрої вміють управляти сигналом RTS. У цьому випадку існує інший, більш універсальний спосіб управління передавачем адаптера.

3.14.3. Автоматичне управління адаптером RS-232C->RS-485

Цей метод у різних фірм називають по-різному: Automatic Send Data Control (ASDC), Automatic Data Direction Control (ADDC), Automatic RTS Signal Control (ARSC), але всі вони базуються на принципі управління за сигналом даних, які передаються (рис. 3.39). Тобто адаптер сам відслідковує, коли включати і коли відключати передавач за даними, які надходять з боку RS-232. Один з варіантів такої реалізації — відслідковування приймального буфера. Як тільки адаптер з боку RS-232 отримав перший біт символу (стартовий біт) — відразу активує передавач. Оскільки біт у стані логічної «1» не може триматися більше часу одного символу, то після паузи відсутності логічного «0» протягом певного часу (не менше тривалості одного символу) конвертер буде ще тримати передавач в активному стані, після чого переведе його у високоімпедансний стан. Якщо затягнути цю паузу, може виникнути ситуація, коли інший передавач у системі з боку RS-485 захоче передати дані (адже лінія вже звільнилася), тож станеться колізія.

Адаптери з управлінням за сигналом даних за приймальним буфером користуються більшою популярністю, ніж адаптери з управлінням сигналами RTS чи DTR.

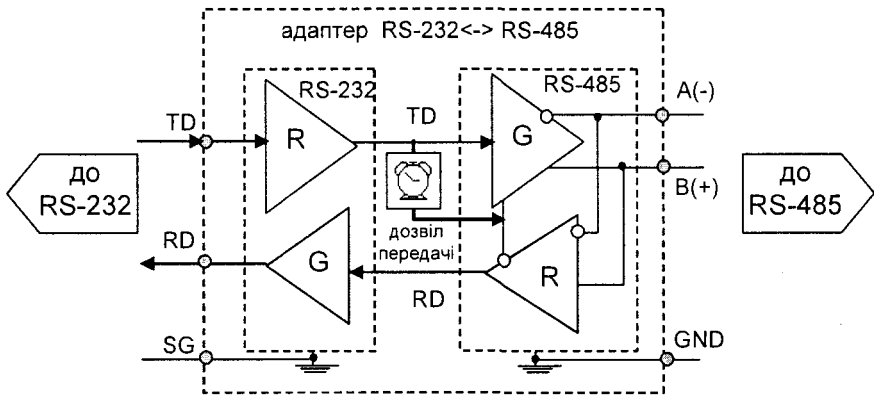


Рис. 3.39. Управління передавачем адаптера за сигналом даних, що передаються

Ще один з методів управління передавачем, який не знайшов широкого використання, за сигналом з боку пристрою з інтерфейсом RS-485. Останній для цього повинен мати спеціальну сигнальну лінію (аналогія з RTS), яка передбачена саме для цього. Коли пристрій з RS-485 планує передавати дані, він переводить перетворювач у стан прийому, тобто вимикає передавач. Специфіка таких перетворювачів полягає в тому, що використати їх можна тільки в режимі точка-точка, що помітно звужує їх використання. Такий перетворювач не ставить ніяких вимог до управління на пристрій із RS-232C, що, звісно, повинно компенсуватись зі сторони RS-485.



Контрольні запитання до розділу 3

1. Які завдання має вирішувати фізичний рівень промислової мережі?
2. Які основні складові фізичного каналу передачі даних?
3. В яких режимах може проходити перенесення даних через лінію зв'язку? Поясніть принципи роботи цих режимів.
4. Які характеристики кабелю враховують при його виборі для промислової мережі?
5. З чого складається еквівалентна схема кабельної лінії передачі? Чим зумовлена наявність комплексного опору та комплексної провідності і від чого вони залежать?
6. Розкажіть про природу явищ погонного опору, поверхневого ефекту, ефекту близькості та реактивної ємності при передачі високочастотного сигналу.
7. В яких робочих областях може міститись режим роботи кабелю при різних частотах? Чим вони характеризуються і від чого залежить перебування в робочій області?
8. Дайте визначення хвильовому опору, характеристичному хвильовому опору, коефіцієнту затухання.

9. Назвіть причини ефекту відбиття сигналу. Як цей ефект впливає на передачу сигналу, які способи боротьби з ним?
10. Які типи електричних кабелів використовуються в промисловій автоматизації?
11. Наведіть приклади типових джерел електромагнітних завад для передачі сигналу по електричним кабелям в промислових умовах експлуатації? Які типи дій їх на лінії передачі?
12. Яким чином можна зменшити дію різних типів електромагнітних завад?
13. Як враховуються типи кабелів при їх прокладці?
14. Розкажіть про необхідність заземлення екрана кабелів? Які схеми заземлення використовують для кабелів промислових мереж? Розкажіть про переваги та недоліки кожного способу.
15. Поясніть необхідність еквіпотенціального з'єднання. Яким чином воно реалізується в межах будівлі та між будівлями?
16. Розкажіть про правила заземлення екрана мережного кабелю при підводі до щита, до клемної колодки та про заземлення екрана роз'ємів.
17. Які способи передачі бітів електричними сигналами Ви знаєте? В чому особливості передачі сигналу за напругою збалансованим та незбалансованим способами?
18. Яким чином передається сигнал по оптоволоконному кабелю? Які типи оптоволоконних кабелів Ви знаєте? Які переваги та недоліки використання оптоволоконного середовища порівняно з електричними кабелями?
19. Коли необхідно використовувати бездротовий зв'язок? В чому його переваги та недоліки?
20. Яке кодування використовують при безпосередньому способі передачі бітів? У чому переваги та недоліки цих методів порівняно один з одним?
21. Які способи модуляції використовуються в цифрових мережах?
22. Поясніть необхідність синхронізації. Що таке синхронна та асинхронна передачі на фізичному рівні?
23. Яке призначення службових бітів при символній передачі?
24. Що таке схеми UART і яке місце вони займають у процесі зв'язку на фізичному рівні?
25. Які способи з'єднання між пристроями доступні при використанні інтерфейсу RS-232? Що таке нуль-модемне з'єднання?
26. Перерахуйте електричні характеристики інтерфейсу RS-232. Які основні недоліки даного інтерфейсу?
27. Які типи роз'ємів визначені інтерфейсом RS-232? Розкажіть про призначення всіх контактів 9-пінового роз'єму.
28. Яким чином організовується управління потоком даних при символному способі передачі по RS-232?
29. Перелічіть електричні характеристики RS-422. Які основні переваги даного інтерфейсу порівняно із RS-232?
30. Перелічіть електричні характеристики RS-485. Які основні переваги даного інтерфейсу порівняно із RS-422?
31. Які схеми з'єднання використовуються з інтерфейсом RS-485?
32. Які схеми термінування використовуються з інтерфейсом RS-485?
33. Розкажіть про призначення захисного зміщення при використанні шинних топологій на базі інтерфейсу RS-485. Як розраховуються резистори захисного зміщення?
34. Перелічіть електричні характеристики інтерфейсу «струмова петля». Які переваги та недоліки даного інтерфейсу порівняно з іншими?
35. В яких випадках необхідно використовувати адаптери-перетворювачі інтерфейсів? Які адаптери найчастіше використовуються?
36. Поясніть необхідність управління передавачем RS-485 в адаптері RS-232<->RS-485. Якими способами вирішується дана проблема?

ТЕХНОЛОГІЇ МОДЕМНОГО ТА БЕЗДРОТОВОГО ЗВ'ЯЗКУ

В даному розділі розглядаються технології символічної передачі даних та модемного зв'язку з використанням різних типів модемів в галузі автоматизації: дротових, бездротових, стандартів стільникового зв'язку.

4.1. Символьний спосіб обміну

Під **символьним способом** обміну розуміють асинхронний обмін між пристроями за допомогою послідовностей старт-стопових символів, протокол якого на каналному та прикладному рівнях не визначений і не реалізований убудованими апаратними та програмними засобами. Простіше кажучи, якщо пристрій має асинхронний канал зв'язку і по ньому дозволяється передавати та приймати символи, визначені користувачем, то можна організувати символічний спосіб обміну.

Символьний спосіб використовують у тих випадках, коли два пристрої не підтримують спільного протоколу мережі, але мають можливість роботи в символічному режимі. Як правило, це такі пристрої, як символічний принтер з послідовним інтерфейсом, операторська панель, відео-термінал і т.п. Застосування такого зв'язку передбачає створення власноруч нового або існуючого протоколу обміну. Тобто, якщо необхідно підключити принтер до контролера, треба визначити набір управляючих символів, послідовність їх посилки і т.д. і, відповідно, програмно реалізувати алгоритм управління друком на принтері. Якщо необхідно організувати таким способом обмін між двома контролерами або контролером і ПК, то треба реалізувати свій протокол на обох пристроях. У будь-якому випадку формуванням кадру і його розшифруванням займається розробник програми користувача, а передачею та прийомом — розробник обладнання.

Символьний спосіб передбачає наявність послідовного інтерфейсу з можливістю асинхронного режиму роботи (наприклад, з мікросхемою UART). Цей інтерфейс налаштовується на необхідні параметри зв'язку: бітова швидкість, кількість бітів даних, паритет, кількість стопових бітів, сигнали управління потоком;

інколи тайм-аути, символи кінця і початку посилки. Як правило, для обміну вибирають 7 або 8 бітів даних. 7-бітний режим обміну ще називають режимом ASCII, оскільки використовується обмін даними в ASCII-кодуванні. Для початку розглянемо способи кодування символів, а потім опишемо символний режим обміну.

4.1.1. Способи кодування символів

Символьний код — це відображення множини чисел на множину символів, тобто числове позначення конкретного символу.

ASCII кодування — Американський стандарт на кодування символів для обміну інформацією (**ASCII-American Standard Code for Information Interchange**). Перша таблиця кодів ASCII складалася із 128 символів, включаючи букви латинського алфавіту, арабські цифри і різноманітні знаки пунктуації (рис. 4.1). Відповідно до цих символів були поставлені перші 128 додатних цілих числа. Таким чином, достатньо було 7 бітів для їх кодування, однак символи кодувалися у вигляді 8-розрядних двійкових чисел. У 1981 році, після випуску першого персонального комп'ютера, IBM розширила початковий код ASCII, доповнивши його ще 128 символами, включаючи деякі символи із алфавітів інших мов, невелику кількість математичних символів і деякі графічні символи для малювання рамок, фону та інших елементів псевдографіки в символному режимі дисплея. Цей код, який складається з 256 символів, відомий як розширений код ASCII (**extended ASCII**).

ASCII знайшов своє застосування в старому ПЗ, зокрема написаному під DOS. Однак, дотепер цей спосіб кодування використовується в мережах з символним способом передачі даних.

В 1987 році Microsoft запропонувала ідею створення **кової сторінки (code page)**, яка є відображенням чисел на символи, залежно від кодової сторінки. Вихідний розширений символний код ASCII, введений IBM, носить назву кодової сторінки 437 або MS-DOS Latin US. Перші 128 символів у всіх кодових сторінках повністю співпадають, але відображення наступних 128 кодів залежить від номера сторінки.

Хоч кількість кодових сторінок швидко збільшувалась, вони не вирішували проблему недостатності 256 кодів (1 байта) для кодування всіх національних символів (літер). В якості прикладу можна навести китайську або японську мови, в кожній з яких більше 20000 символів. Тому в 1988 році був створений проект Unicode, який повинен був вирішити дану проблему. **Unicode** — це двобайтний символний код, який дозволяє представити 65536 різних символів. Версія 2.0 даного стандарту включає 38885 символів. Microsoft використовує термін «широкий» символ (**wide character**) як синонім слова Unicode, хоча інші використовують термін «широкий» для позначення будь-якого двохбайтного символного коду. Перші 256 символів Unicode ті ж, що у розширеного символного коду ASCII, для цього старший байт кожного символу встановлений у нуль. За розвиток Unicode і надання технічної інформації за ним відповідає консорціум Unicode (**Unicode Consortium**), який був створений у 1991 році.

Ctrl	Dec	Hex	Char	Code	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
^⊙	0	00		NUL	32	20	sp	64	40	Q	96	60	·
^A	1	01	␣	SOH	33	21	!	65	41	A	97	61	a
^B	2	02	␣	STX	34	22	"	66	42	B	98	62	b
^C	3	03	♥	ETX	35	23	#	67	43	C	99	63	c
^D	4	04	♦	EOI	36	24	\$	68	44	D	100	64	d
^E	5	05	♣	ENQ	37	25	%	69	45	E	101	65	e
^F	6	06	♠	ACK	38	26	&	70	46	F	102	66	f
^G	7	07	·	BEL	39	27	'	71	47	G	103	67	g
^H	8	08	▣	BS	40	28	(72	48	H	104	68	h
^I	9	09	○	HI	41	29)	73	49	I	105	69	i
^J	10	0A	◻	LF	42	2A	*	74	4A	J	106	6A	j
^K	11	0B	♂	VI	43	2B	+	75	4B	K	107	6B	k
^L	12	0C	♀	FF	44	2C	,	76	4C	L	108	6C	l
^M	13	0D	⌋	CR	45	2D	-	77	4D	M	109	6D	m
^N	14	0E	⌌	SO	46	2E	.	78	4E	N	110	6E	n
^O	15	0F	*	SI	47	2F	/	79	4F	O	111	6F	o
^P	16	10	▾	SLE	48	30	0	80	50	P	112	70	p
^Q	17	11	◀	CS1	49	31	1	81	51	Q	113	71	q
^R	18	12	⬆	DC2	50	32	2	82	52	R	114	72	r
^S	19	13	!!	DC3	51	33	3	83	53	S	115	73	s
^T	20	14	⌈	DC4	52	34	4	84	54	T	116	74	t
^U	21	15	⌍	NAK	53	35	5	85	55	U	117	75	u
^V	22	16	▬	SYN	54	36	6	86	56	V	118	76	v
^W	23	17	≡	EIB	55	37	7	87	57	W	119	77	w
^X	24	18	↑	CAN	56	38	8	88	58	X	120	78	x
^Y	25	19	↓	EM	57	39	9	89	59	Y	121	79	y
^Z	26	1A	→	SE	58	3A	:	90	5A	Z	122	7A	z
^[27	1B	+	ESC	59	3B	;	91	5B	[123	7B	{
^\	28	1C	⌋	FS	60	3C	<	92	5C	\	124	7C	;
]]	29	1D	+	GS	61	3D	=	93	5D]]	125	7D	}
^^	30	1E	▲	RS	62	3E	>	94	5E	^	126	7E	~
_	31	1F	▼	US	63	3F	?	95	5F	_	127	7F	Δ†

Рис. 4.1. Таблица кодів ASCII основних символів (0-127)

Треба відмітити, що в ОС Windows NT 2000 — Unicode основний символний код, хоч підтримується для сумісності ANSI. В Windows XP, Vista, 7 та Windows CE — ANSI не підтримується. В Windows 9x, навпаки, Unicode не використовується за деяким винятком. А всі функції потребують ANSI-формату символних рядків.

4.1.2. Принципи символного обміну ASCII

Як правило, при побудові кадру в символному обміні перші 32 символи ASCII-таблиці використовуються в якості символів керування. Для символів даних використовуються символи, які являють собою латинські букви зі знаками пунктуації та арабські цифри. Таким чином, якщо початок та кінець посилки позначаються службовими символами, то це виключає ситуацію (за винятком спотворення кадру на лінії) співпадання одного із символів даних з позначенням кінця кадру.

Для прикладу розглянемо формування кадру в ASCII-режимі, який показаний на рис. 4.2. Вгорі зображено позначення полів кадру, а внизу символне представлення в 16-ковому форматі. Для позначення початку кадру використовується символ STX з 16-ковим кодом 02. Далі йдуть дані у вигляді рядку буквоцифрових символів, які закінчуються стартовою послідовністю CR LF (0D 0A). Зокрема, CR означає переведення каретки, тобто принтер при зустрічі такого символу почне друкувати з нового абзацу.

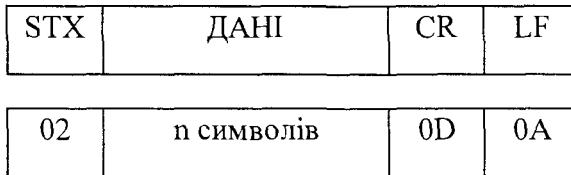


Рис. 4.2. Приклад кадру для ASCII-режиму обміну

Як правило, приймачі можуть бути налаштовані на приймання рядку символів (наприклад, кадру) через буфер прийому, який заповнюється символами доти, поки не спрацює умова заповнення буфера (рис. 4.3). Після заповнення буфера його вміст доступний програмі користувача. Умовою заповнення буфера може бути один із декількох варіантів розмежування кадрів: обмеження за кількістю прийнятих символів, зустріч стопової послідовності символів або тайм-аут після останнього прийнятого символу. В ASCII-режимі дуже зручно користуватися саме другим типом розмежування. В наведеному прикладі приймач налаштований на стопову послідовність CR LF, після якої він припинить приймати символи, поки його не попросить прикладна програма.

Передача текстової послідовності проводиться шляхом кодування літер ASCII-кодами. Для передачі числових даних використовуються принаймні два способи.

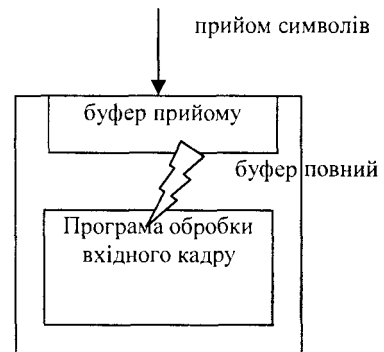


Рис. 4.3. Робота з буфером прийому

При першому способі числові дані при передачі на принтер, дисплей панель оператора чи відео-термінал кодуються за кожною десятковою цифрою, як текст. Тобто число 255_{10} передається трьома символами, як «2», «5» та «5», відповідно в ASCII 32_{16} , 35_{16} та 35_{16} . Такий спосіб зручний, але потребує значної кількості символів при передачі багаторозрядних чисел (з великою кількістю цифр). Окрім того, при реалізації протоколу на обох пристроях програма ускладнюється для передачі дробових чисел (чисел з плаваючою комою).

Другий спосіб — використання шістнадцяткового кодування. Тобто, кожне число передається як набір символів його шістнадцяткового представлення. Наприклад, число 255_{10} , що в 16-ковому представленні відповідає FF_{16} , передається двома символами: «F» (46_{16}) та «F» (46_{16}).

На рисунку 4.4 показано порівняння двох способів кодування чисел в ASCII-режимі при формуванні кадру. Кожний кадр представлений відповідно згоризонтально: у символному, шістнадцятковому і бінарному форматах. Зліва зображений кадр при кодуванні десяткового представлення, справа — шістнадцяткового.

STX	2	5	5	CR	LF
02	32	35	35	0A	0D
0000010	0110010	0110101	0110101	0001101	0001010

STX	F	F	CR	LF
02	46	46	0A	0D
0000010	1000110	1000110	0001101	0001010

Рис. 4.4. Порівняння кодування чисел:
зліва — десяткового представлення, справа — шістнадцяткового

Слід зазначити, що в даному прикладі не враховані деякі особливості, пов'язані з адресацією, контролем за правильністю доставки кадру і т.д. Для всіх цих функцій необхідно вводити допоміжні поля.

ASCII-режим є досить популярним для символного способу зв'язку. Його перевага в простоті реалізації. Одним з основних недоліків такого способу є значні затрати ресурсів мережі. Адже на 7 бітів даних припадає як мінімум 2 службові біти (1 стартовий, 1 стоповий), а числові дані передаються ASCII-кодами їх символного представлення. Для підвищення кількості корисних даних, переданих за одиницю часу, краще використовувати 8 бітів даних. У цьому випадку використання буфера прийому може бути налаштоване на кількість прийнятих байтів, тайм-аути між символами чи на стопову послідовність. Приклад останнього може бути так само представлений, як і в ASCII-режимі (рис. 4.2). При розмежуванні між кадрами службовими символами у сформованому кадрі кодування числових даних може співпасти зі стартовою або стоповою послідовністю. Тобто, якщо в наведеному прикладі (рис. 4.2.) в полі даних будуть передаватися два числа (за байтом кожний) 13_{10} та 10_{10} , то ця послідовність буде прийнята як кінець кадру. Одним із способів вирішення цієї проблеми є додавання спеціального escape-символу (ESC, знак переключення коду) між байтами, які визначають стопову послідовність (Byte-Stuffing — символне начинення). Таким чином, при передачі в полі даних стопової послідовності необхідно між ними вставити escape-символ, який при прийомі попередньо вилучити. Якщо в полі даних зу-

стрінеться байт даних, який рівний ESC, але не є службовим, то перед цим байтом вставляється ще один байт ESC, тобто в полі даних будуть вже два підряд байти з кодом ESC.

4.2. Модемний обмін

Модем (скорочення від модулятор-демодулятор) — пристрій зв'язку, що виконує функції модуляції та демодуляції. Модуляція — це зміна характеристик несучого сигналу відповідно зі зміною вхідного інформаційного сигналу, демодуляція — зворотний процес.

До критеріїв класифікації модемів можна віднести такі: область застосування; функціональне призначення; тип використовуваного каналу; конструктивне виконання; підтримка протоколів модуляції, виправлення помилок і стиснення даних. Можна виділити ще безліч більш детальних технічних ознак, таких, як спосіб модуляції, інтерфейс сполучення з DTE і так далі.

За способом реалізації каналу модеми можна поділити на такі групи:

- для комутованих телефонних каналів;
- для виділених телефонних каналів;
- для фізичних з'єднувальних ліній:
 - модеми низького рівня (лінійні драйвери) або модеми на коротку відстань (short range modems);
 - модеми основної смуги (baseband modems);
- для цифрових систем передачі (CSU/DSU);
- для стільникових систем зв'язку;
- для пакетних радіомереж;
- для локальних радіомереж.

Переважна більшість модемів, що випускаються, призначена для використання на комутованих телефонних каналах. Такі модеми повинні вміти працювати з автоматичними телефонними станціями (АТС), розрізняти їх сигнали і передавати свої сигнали набору номера.

Основна відмінність модемів для фізичних ліній від інших типів модемів полягає в тому, що смуга пропускання фізичних ліній не обмежена значенням 3,1 кГц, характерним для телефонних каналів. Однак, смуга пропускання фізичної лінії також є обмеженою і залежить головним чином від типу фізичного середовища (екранована і неекранована вита пара, коаксіальний кабель тощо) і її довжини. З погляду використовуваних для передачі сигналів модеми для фізичних ліній можуть бути розділені на модеми низького рівня (лінійні драйвери), що використовують цифрові сигнали, і модеми «основної смуги» (baseband), у яких застосовуються методи модуляції, аналогічні вживаним у модемах для телефонних каналів. У модемах першої групи зазвичай використовуються цифрові методи бі-імпульсної передачі, що дозволяють формувати імпульсні сигнали без постійної складової і часто займають більш вузьку смугу частот, ніж вихідна цифрова послідовність. У модемах другої групи часто використовуються різні види квадратурної амплітудної модуляції, що дозволяють радикально скоротити необхідну

для передачі смугу частот. У результаті на однакових фізичних лініях такі модеми можуть досягати швидкості передачі до 100 Кбіт/с, у той час, як модеми низького рівня забезпечують тільки 19,2 Кбіт/с.

Модеми для цифрових систем передачі нагадують модеми низького рівня. Однак, на відміну від них забезпечують підключення до стандартних цифрових каналів, таких, як E1/T1 або ISDN, і підтримують функції відповідних каналних інтерфейсів.

Модеми для стільникових систем зв'язку відрізняються компактністю виконання і підтримкою спеціальних протоколів модуляції і виправлення помилок, що дозволяють ефективно передавати дані в умовах стільникових каналів з високим рівнем перешкод і постійно змінюваними параметрами.

Пакетні радіомодеми призначені для передачі даних по радіоканалу між мобільними користувачами. При цьому декілька радіомодемів використовують один і той же радіоканал у режимі множинного доступу. Радіоканал за своїми характеристиками близький до телефонного і організовується з використанням типових радіостанцій, налаштованих на одну і ту ж частоту. Пакетний радіомодем реалізує методи модуляції і множинного доступу.

Локальні радіомережі є перспективною мережною технологією, що доповнює звичайні локальні мережі. Основним їх елементом є спеціалізовані радіомодеми (адаптери локальних радіомереж). На відміну від пакетних радіомодемів, такі модеми забезпечують передачу даних на невеликі відстані (до 300 м) з високою швидкістю (10 Мбіт/с), зіставною із швидкістю передачі в дротяних локальних мережах. Крім того, радіомодеми локальних радіомереж працюють у певному діапазоні частот із застосуванням сигналів складної форми, таких, як сигнали з псевдовипадковою перебудовою робочої частоти.

У промислових системах управління актуальним є використання останніх трьох груп модемів.

В окремий клас можна виділити промислові модеми, що, крім промислових умов експлуатації, ще можуть підтримувати такі функції:

– багатофункціональний інтерфейс, тобто наявність декількох каналів зв'язку (RS232, RS485, USB);

– можливість подавати на вхід модему через послідовний COM-порт RS232 аналоговий сигнал безпосередньо з виконавчих пристроїв (датчиків, контролерів) завдяки наявності вбудованого 10-розрядного аналого-цифрового перетворювача;

– обробка і керування аналогових сигналів завдяки наявності спеціальних вводів-виводів;

– наявність великого обсягу пам'яті для аварійних повідомлень завдяки наявності FLASH-пам'яті;

– наявність відкритої перепрограмованої платформи, що дозволяє розробникам бездротового обладнання створювати гнучкі рішення і завантажувати ПО користувача прямо в FLASH-пам'ять модема;

– вбудований TCP/IP-стек, що дозволяє не застосовувати додаткове обладнання для передачі файлів і роботи з електронною поштою;

– віддалені настройки, конфігурацію та дистанційний моніторинг модема.

Для управління модемом використовують **АТ-команди** представлені рядками символів, що починаються з літер АТ (Attention — увага) і завершуються символом повернення каретки (<CR>) (іноді у рядку ініціалізації модема кінцевий символ формується автоматично). Символи команд можуть бути прописними або малими, між командами можуть бути присутні або відсутні пробіли. АТ-команди, крім виконання прямих дій з переключення режиму модема, управляють **S-регістрами** модема, в яких міститься службова інформація, потрібна для роботи. Наприклад, у S-регістрах зберігаються константи, що визначають час очікування дзвінка, типи використовуваних протоколів, рівень вихідного і вхідного сигналів, аналогові сигнали і т. п. Кількість S-регістрів визначається розробником модема і може становити декілька сотень. Але прийнято, що за першими 13-ма S-регістрами зберігається призначення, введене фірмою Hayes. Вміст S-регістрів користувач може зчитувати і модифікувати на свій розсуд, але слід пам'ятати, що між значеннями в різних S-регістрах є складний взаємозв'язок. Крім того, довільна зміна будь-якого S-регістра може призвести до того, що неможливо буде використовувати стандартні програми для роботи в Інтернеті.

Основні АТ-команди наведені в табл. 4.1, сесію починають з команди АТ і, якщо є зв'язок модем, відповідає ОК.

Таблиця 4.1

ОСНОВНІ АТ-КОМАНДИ МОДЕМІВ

Команда	Опис
+++	Модем розриває з'єднання і повертається у командний режим
A	Команда відповіді (Answer Command)
Vn	Налаштування зв'язку (Communications Options)
D	Команда набору (Dial Command)
En	Команда вибору ехо-символу (Select Command Character Echo Option)
Hn	Управління Switchhook — емуляція натиснення телефонного важеля (Control The Switchhook)
I0	Ідентифікація коду продукту (Identify The Product Code)
I2	Виконання тесту контрольної суми ROM (Perform ROM Checksum Test)
I7	Номер версії (Version Number)
Ln	Вибір рівня гучності динаміки (Select Speaker Volume Level)
Mn	Функція вибору опцій динаміки (Select Speaker Function Option)
Nn	Вибір опцій для встановлення зв'язку (Select Negotiate Handshake Option)
On	Перехід до онлайн-команд (Go Online Command)
P	Вибір методів імпульсного набору (Select Pulse Dialing Method)
Qn	Вибір опції результуючого коду (Select Result Code Option)
Sn=	Запис у S-регістр (Write To An S-Register)
Sn?	Читання S-регістра (Read An S-Register)
T	Вибір методу тонового набору (Select Tone Dialing Method)

Закінчення табл. 4.1

Команда	Опис
Vn	Вибір опції формату відповіді (Select Response Format Option)
Wn	Вибір розширеного результуючого коду (Select Extended Result Code)
Xn	Вибір опції модемного виклику (Select Call Progress Option)
Yn	Вибір опції бездіяльності та роз'єднання (Select Long Space Disconnect Option)
Zn	Виконання м'якого скидання (Perform Soft Reset)
&An	Вибір ролі автовідповідача (Select Originate/Answer Role For Autoanswer)
&Cn	Вибір опції визначення переданих даних (Select Data Carrier Detect Option)
&Dn	Вибір опції готовності терміналу даних (Select Data Terminal Ready Option)
&F	Завантаження заводських установок (Load Factory Default Profile)
&Gn	Вибір опції захисту тонового набору (Select Guard Tone Option)
&Kn	Вибір опції потоку ConTDol (Select Flow ConTDol Option)
&Pn	Вибір параметрів імпульсного набору (Select Pulse Dialing Parameters)
&Qn	Вибір опції режиму зв'язку (Select Communications Mode Option)
&Rn	Вибір опції RTS/CTS (Select RTS/CTS Option)
&Sn	Вибір опції готовності передачі даних (Select Data Set Ready Option)
&T0	Тест завершення в процесі (Terminate Test In Process)
&T1	Ініціювання локального аналога мережної петлі (Initiate Local Analog Loopback)
&T3	Виконання локальної цифрової мережної петлі (Perform Local Digital Loopback)
&T4	Включення RDL-запитів (Enable Granting Of RDL Requests)
&T5	Заборона надання RDL-запитів (Deny Granting Of RDL Requests)
&T6	Ініціювання віддаленої цифрової мережної петлі (Initiate Remote Digital Loopback)
&T7	Ініціювання внутрішнього тесту RDL (Initiate RDL With Self Test)
&T8	Внутрішній тест локальної мережної петлі (Local Loopback With Self Test)
&T19	Виконання тесту RTS/CTS кабелю (Perform RTS/CTS Cable Test)
&Un	Скасування TDellis-кодування (Disable TDellis Coding)
&V	Перегляд профілів конфігурації (View Configuration Profiles)
&Wn	Збереження активного профілю (Store Active Profile)
&Xn	Вибір джерела синхронізації часу TDansmit (Store Active Profile)
&Yn	Вибір збереження профілю для апаратного перезавантаження (Select Stored Profile For Hard Reset)
&Zn=	Збереження телефонного номера (Store Telephone Number)
,	Пауза (Perform Pause)
=	Запис у S-регістр (Write To An S-Register)
?	Читання S-регістра (Read An S-Register)
P	Вибір імпульсного набору (Select Pulse Dialing)

Приклад 4.1. AT-команди. Обмін SMS-повідомленнями через HyperTerminal.

Завдання. Налаштувати бездротовий GSM-модем фірми Wavescom та ініціалізувати обмін між персональним компютером і стільниковим телефоном за допомогою служби HyperTerminal (рис. 4.4).

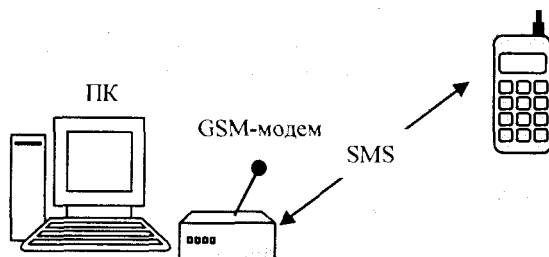


Рис. 4.4. До прикладу 4.1

Рішення. GSM-модем Wavescom оснащений інтерфейсом RS-232, тому з'єднання між ПК та модемом (М) виконується нуль-модемним кабелем. Налаштування та обмін даними за допомогою AT-команд, що вводяться через HyperTerminal, та відповіді модема наведені в табл. 4.2.

Таблиця 4.2

НАБІР АТ-КОМАНД ДО ПРИКЛАДУ 4.2

Команда	Опис
AT	ПК: увага
OK	М(модем): так
AT+CBST?	ПК: запит на вибір типу носія: режими швидкості, стиснення даних, прозорості
+CBST: 0,0,1	М: 0 — швидкість автоматична, 0 — стиснення відсутнє, 1 — непрозорий
OK	М: так
AT+CBST=7,0,1	ПК: запит на підтримання носія
OK	М: так, підтримується
AT+CBST=12,0,1	ПК: встановлення швидкості — 9600 біт/с
OK	М: так
AT+IFC?	ПК: запит команди управління місцевим потоком <DCE_by_DTE> та <DTE_by_DCE>
+IFC: 0,0	М: не підтримується, не підтримується
OK	М: так
AT&W	ПК: запис поточних налаштування в EEPROM
OK	М: так, записано
+CMTI: «SM», 1	М: отримано нове повідомлення, індекс 1

Закінчення табл. 4.2

Команда	Опис
AT + CMGF = 1	ПК: встановлення текстового режиму
OK	М: так, встановлено
AT + CMGR = 1	ПК: читати перше повідомлення
+ CMGR: «REC UNREAD», «+380669572528», «09/05/18.14:09:28 + 12»	М: непрочитане повідомлення з номера телефону +380669572528 отримано 09/05/18 о 14:09:28
Hello	М: текст повідомлення
OK	М: так
AT + CMGL = «REC UNREAD»	ПК: показати непрочитані повідомлення в текстовому режимі
OK	М: так, ця відповідь визначає, що непрочитаних повідомлень не має
AT + CMGR = 2	ПК: читати друге повідомлення
ERROR	М: помилка — другого повідомлення не виявлено
AT + CMGR = 1	ПК: знову читати перше повідомлення
+ CMGR: «REC READ», «+380669572528», «09/05/18.14:09:28 + 12»	М: прочитане повідомлення з номера телефону +380669572528 отримано 09/05/18 о 14:09:28
Hello	М: текст повідомлення
OK	М: так
AT + CMGS = + 380669572528	ПК: відіслати повідомлення в текстовому режимі на номер +380669572528
> HELLOW	ПК: текст повідомлення
+CMGS: 23	М: повідомлення відіслано, номер повідомлення 23 (визначається виробником)
OK	М: так

Продовження 4.1

4.3. Бездротовий спосіб обміну

Не завжди в системах управління в якості фізичного носія можна використати кабель (металевий чи оптоволоконний). Розглянемо декілька прикладів, коли виникає необхідність у бездротовому зв'язку:

- контроль та управління мобільними (пересувними) системами;
- віддалений моніторинг стану системи;
- управління «розумними» будинками;
- відсутність можливості прокладки кабелів.

У першому випадку ситуація очевидна — використання кабелів для зв'язку з мобільною системою (об'єктом управління) обмежує її рух. Крім того, навіть при великій гнучкості кабелів термін їх служби зменшується за рахунок втоми металу або фізичного ушкодження.

Віддалений моніторинг необхідний операторам або технологам для отримання аварійних повідомлень з контролюючої системи (скажімо, SCADA), перебуваючи при цьому у будь-якій точці підприємства. Така система також зручна при аналізі працездатності системи, при перевірці її складових, які територіально розподілені.

Усе більш перспективним напрямком стає автоматизація у багатоповерхових та приватних житлових будинках, що включає в себе управління електро-, тепло- та водопостачанням, пожежну та охоронну сигналізацію і т.д. Альтернативою провідникам для підключення датчиків, виконавчих механізмів, контролерів може служити радіосигнал, а в якості панелі оператора — звичайний мобільний телефон.

Звісно, за неможливості прокладки кабелів або дороговизни такого рішення альтернативи бездротовому зв'язку немає.

4.3.1. Питання стандартизації

Питаннями міжнародної стандартизації промислових мереж займаються МЕК і національні комітети за підтримки незалежних професійних товариств (наприклад, HCF (Hart Communication Foundation), FF (Fieldbus Foundation), PNO (Profibus & Profinet Organization), ODVA (Open Device Vendor Association — Асоціація виробників пристроїв для відкритих систем). Провідна роль при розробці бездротових стандартів для мереж промислової автоматики належить ISA і WCT (Wireless Cooperation Team — спілка промисловців з бездротових технологій). WCT — це зовсім нова спілка з товариств HCF, FF, PNO, ODVA, яка була утворена у вересні 2007 р. з метою створення бездротової інфраструктури на польовому рівні АСУ ТП. У комітеті ISA і WCT ведеться подальша робота зі створення цілого сімейства бездротових стандартів для промислової автоматики, а також єдиної бездротової інфраструктури для підприємств і великих промислових комплексів.

Варто також відмітити відділ ITU-R, який відповідає за питання частотного регулювання та розподілу частот для бездротових мереж у світовому масштабі, що спирається на національні міністерства і відомства. наприклад FCC (Federal Communication Commission — Федеральна комісія США із зв'язку) і CFR47 країн Америки, ФДУП «ГКРЧ» Росії, а також на такі організації і комітети Європи, як ETSI (European Telecommunication Standardization Institute — Європейський Інститут зі стандартизації), комітет ECC (Electronic Communications Committee — Комітет з електронних засобів зв'язку). Так, у розділах регулювання частот ITU-R по регіонах світу була виділена єдина (за деяким винятком) безліцензійна радіочастотна смуга ISM (Industrial, Scientific and Medical) — 6,765 МГц — 246 ГГц.

В Україні виділяються смуги робочих частот відповідно до виду пристроїв, що їх використовують. Використання бездротових промислових мереж належать за *постановою Кабінету Міністрів України від 15 грудня 2005 р. № 1208* до «промислових пристроїв», «наукових пристроїв», «медичних пристроїв», «побутових пристроїв».

При впровадженні бездротових мереж у промисловій автоматичі, крім стандартних вимог, до промислових мереж необхідно враховувати:

- інтенсивність обміну даними;
- можливість використання автономних джерел живлення великої ємності;
- топологію радіомережі, яка повинна передбачити надлишковість зв'язку та самоорганізацію мережі, що дасть змогу підвищити надійність мережі.

4.3.2. Класифікація бездротових способів обміну

Умовно бездротові способи обміну можна поділити так, як на рис. 4.4. Відмітимо, що подана класифікація умовна, тому що нині засоби стільникового зв'язку 3G, 4G підтримують (інтегруються) пакетні та локальні радіомережі стандарту 802.11 та 802.16.

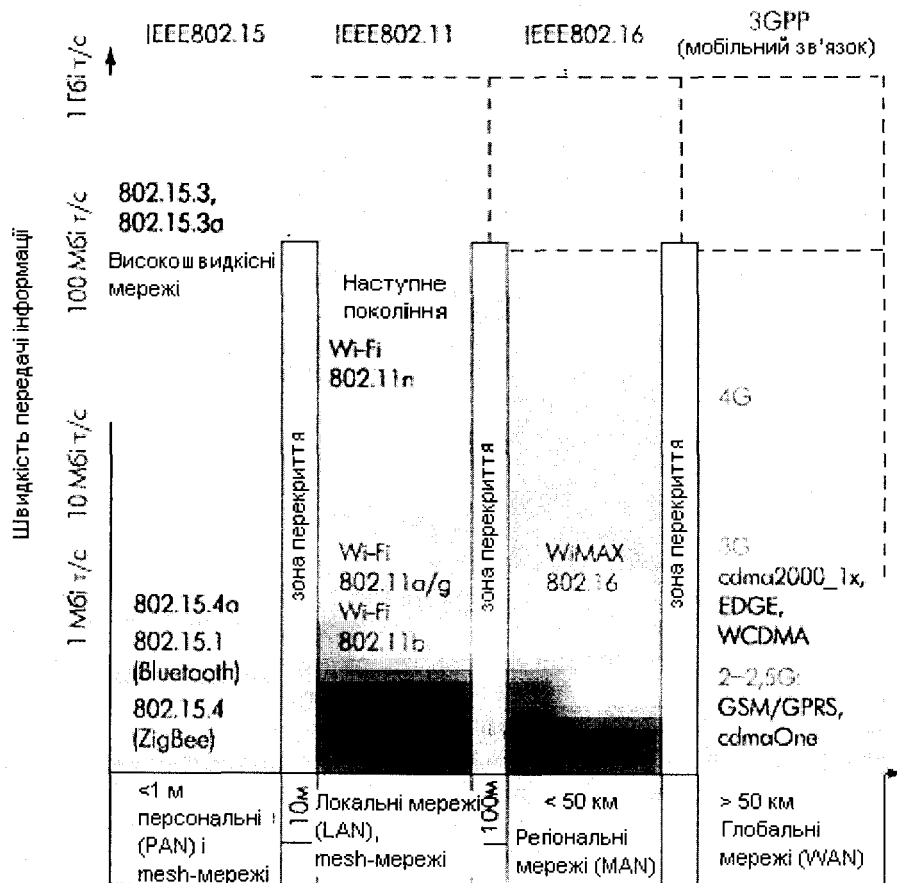


Рис. 4.4. До класифікації бездротових мереж

Наведемо класифікацію бездротових мереж за територіальним розподілом:

– WPAN (wireless personal area network) — персональні мережі, що базуються на стандарті **802.15**. Прикладом можуть бути мережі IrDA, Bluetooth, а також промислові — UWB, Z-Wave, ZigBee, Wireless Hart. Даний стандарт поділяється на такі:

- 802.15.1, що базується на специфікації Bluetooth;
- 802.15.2, що застосовується для взаємодії пристроїв 802.11 и 802.15;
- 802.15.3, що розроблений для високошвидкісної передачі даних;
- 802.15.4, що визначає низькошвидкісну передачу даних.

– WLAN (wireless local area network) — локальні мережі, що базуються на стандарті **802.11**. Прикладом таких мереж є Wi-Fi, HyperLAN.

– WMAN (Wireless Metropolitan Area Networks) — регіональні мережі, що базуються на стандарті **802.16**. Найпопулярніша мережа цього стандарту є мережа WiMAX.

Нагадаємо, що кожен з наведених стандартів, що розроблений комітетом IEEE описує фізичний рівень та підрівень MAC канального рівня.

4.3.3. Стандарт IEEE 802.15

Основним поняттям цього стандарту є пікомережа (piconet), що складається з головного вузла та декількох підпорядкованих у радіусі 10 м. Пікомережі можуть об'єднуватися через спеціальні мости. Головний вузол, крім активних вузлів, може підтримувати пасивні вузли (з пониженим енергоспоживанням). А загалом вузол даної мережі може підтримувати ще два види зниженого енергоспоживання — зупинений та аналізуючий. Завдяки цьому подовжується ресурс їх джерел живлення. Опишемо коротко кожен із сімейства стандартів 802.15.

4.3.3.1. IEEE 802.15.1. Це стандарт для мереж WPAN, який був створений IEEE 802.15 Task Group 1 спільно з Bluetooth SIG (технологія Bluetooth з'явилася раніше 802.15). Базується на специфікації Bluetooth v.1.1, що передбачає радіус дії близько 10 м. Рівень радіозв'язку переносить інформацію від головного вузла до підлеглих і навпаки. Мережа працює в неліцензійному діапазоні 2,4 ГГц, що розділений на 79 каналів по 1 МГц у кожному. Метод модуляції — частотна маніпуляція з 1 бітом на герц, що визначає сумарну швидкість 1 Мбіт/с. Використовується стрибкоподібна перебудова частоти, послідовність якої генерується головним вузлом.

4.3.3.2. IEEE 802.15.3. Група IEEE 802.15 Task Group 3 розробляє цей високошвидкісний стандарт та обіцяє надати недорогу мережу з низьким споживанням енергії, призначену для потреб споживчої електроніки. Стандарт передбачає використання смуги 2,4 GHz, а для модуляції застосовується квадратурна маніпуляція фазовим зрушенням зі зміщенням (Offset Quadrature Phase Shift Keying - OQPSK), за допомогою якої досягається швидкість передачі від 11 до 55 Мбіт/с.

4.3.3.3. IEEE 802.15.4. Специфікація призначається для низькошвидкісних WPAN. Над нею працювала IEEE 802.15 Task Group 4. Очікується, що стандарт забезпечить низьке споживання енергії і простоту розробки пристроїв. Передбачуваним застосуванням є сенсори, інтерактивні іграшки, віддалене управління, домашня автоматика. Стандарт використовує 16 каналів у діапазоні 2,4 ГГц,

10 каналів у діапазоні 915 МГц і один канал у смузі 868 МГц, що прийнято в Європі. Цей стандарт описує фізичний та каналний рівні зі швидкістю обміну даними до 250 Кбіт/с і топології побудови радіомережі «зірка», «точка-точка», mesh. Він є основою побудови більшості бездротових сенсорних мереж із пересилкою даних. І в першу чергу цей стандарт може бути застосований у мережах промислової автоматики, оскільки польові шини АСУ ТП, як правило, є низькошвидкісними.

4.3.4. Стандарт IEEE 802.11

Стандарт IEEE 802.11 — базовий стандарт, він визначає протоколи, необхідні для організації бездротових локальних мереж (WLAN). Основні з них — протокол управління доступом до середовища MAC і протокол PHY передачі сигналів у фізичному середовищі. Як останній, допускається використання радіохвиль та інфрачервоного випромінювання. Стандартом 802.11 визначений єдиний підрівень MAC, взаємодіючий з трьома типами протоколів фізичного рівня, що відповідають різним технологіями передачі сигналів — за радіоканалами у діапазоні 2,4 ГГц із широкосмугового модуляцією з прямим розширенням спектра (DSSS) та з перебудовою частоти (FHSS), а також за допомогою інфрачервоного випромінювання. Специфікаціями стандарту передбачені два значення швидкості передачі даних — 1 і 2 Мбіт/с. Порівняно з дротяними Ethernet можливості підрівня MAC розширені за рахунок включення в нього ряду функцій, що зазвичай виконуються протоколами більш високого рівня, зокрема процедур фрагментації і ретрансляції пакетів. Це викликано прагненням підвищити ефективну пропускну спроможність системи завдяки зниженню накладних витрат на повторну передачу пакетів. Основним методом доступу до середовища стандартом 802.11 визначено механізм CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance — множинний доступ з виявленням несучої та запобіганням колізій).

Для економії енергоресурсів мобільних робочих станцій, що використовуються в мережі 802.11, передбачений механізм перемикання станцій у так званий пасивний режим з мінімальним споживанням потужності.

В основу стандарту 802.11 покладена стільникова архітектура, причому мережа може складатися як з однієї, так і кількох комірок. Кожна комірка управляється базовою станцією (точка доступу — Access Point, AP), яка разом з розташованими в межах радіуса її дії робочими станціями користувачів утворює базову зону обслуговування (Basic Service Set, BSS). Точки доступу багатокімпрчастої мережі взаємодіють між собою через розподільну систему (Distribution System, DS), що являє собою еквівалент магістрального сегмента кабельних локальних мереж. Вся інфраструктура, що включає точки доступу та розподільну систему, утворює розширену зону обслуговування (Extended Service Set).

Стандартом передбачений також однокомірчастий варіант бездротової мережі, який може бути реалізований і без точки доступу, при цьому частину її функцій виконують безпосередньо робочі станції.

Для забезпечення переходу мобільних робочих станцій із зони дії однієї точки доступу до іншої у багатокімпрчастих системах передбачені спеціальні процедури

сканування (активного і пасивного прослуховування ефіру) і приєднання (Association), проте строгих специфікацій з реалізації роумінгу стандарт 802.11 не передбачає.

Для захисту WLAN стандартом IEEE 802.11 слугує цілий комплекс заходів безпеки передачі даних під загальною назвою Wired Equivalent Privacy (WEP). Він включає засоби протидії несанкціонованому доступу до мережі (механізми і процедури аутентифікації), а також запобігання перехопленню інформації (шифрування).

Даний стандарт сумісний зі стандартом Ethernet вище канального рівня, зокрема це стосується можливості посилення IP-пакетів.

Основні різновиди стандарту 802.11 та їх характеристики наведені в табл. 4.5.

Стандарти 802.11a/g використовують метод модуляції OFDM (ортогональне частотне ущільнення), що передбачає 52 різні робочі частоти, 48 з яких використовуються для даних, а 4 — для синхронізації. Розподіл сигналу на багато вузьких діапазонів має переваги перед передачею в одному широкому діапазоні — нижчу чутливість до вузькосмугової інтерференції та можливість використання незалежних діапазонів. Система кодування QAM — це модуляція з фазовим зміщенням.

Стандарт 802.11b використовує метод високошвидкісної передачі широкосмугового сигналу за методом прямої послідовності, що кодує біти зі швидкістю 11 млн. елементарних сигналів за секунду. Для сумісності з 802.11a/g використовує також модуляцію з фазовим зміщенням. Хоча швидкість 802.11b менша за 802.11a/g, діапазон першого майже в 7 разів ширший, що важливо в деяких ситуаціях.

Таблиця 4.5

ОСНОВНІ ВЕРСІЇ СТАНДАРТУ 802.11

Версія 802.11	Дата випуску	Частота (ГГц)	Смуга пропускання (МГц)	ШПД на потік (Мбіт/с)	Модуляція
–	Jun 1997	2.4	20	1, 2	DSSS
a	Sep 1999	5	20	6, 9, 12, 18, 24, 36, 48, 54	OFDM
		3.7			
b	Sep 1999	2.4	20	1, 2, 5.5, 11	DSSS
g	Jun 2003	2.4	20	1, 2, 6, 9, 12, 18, 24, 36, 48, 54	OFDM, DSSS
n	Oct 2009	2.4/5	20	7.2, 14.4, 21.7, 28.9, 43.3, 57.8, 65, 72.2	OFDM
			40	15, 30, 45, 60, 90, 120, 135, 150	

Пристрої стандарту 802.11n можуть працювати у трьох режимах: успадкованому (Legacy), в якому забезпечується підтримка пристроїв 802.11b/g і 802.11a, змішаному (Mixed), в якому підтримуються пристрої 802.11b/g, 802.11a і 802.11n «Чистому» режимі — 802.11n (саме в цьому режимі і можна скористатися перевагами підвищеної швидкості до 600 Мбіт/с і збільшеною дальністю передачі даних, що забезпечує стандарт 802.11n). Ключовий компонент стандарту 802.11n

під назвою МІМО (Multiple Input, Multiple Output — багато входів, багато виходів) передбачає застосування просторового мультиплексування з метою одночасної передачі декількох інформаційних потоків за одним каналом, а також багатопрореневе відображення, яке забезпечує доставку кожного біта інформації відповідного одержувача з невеликою ймовірністю впливу перешкод і втрат даних. Саме можливість одночасної передачі і прийому даних визначає високу пропускну здатність пристроїв 802.11n.

Слід відмітити, що на практиці, навіть при максимально можливому рівні сигналу, продуктивність даних мереж ніколи не досягає зазначеного вище теоретичного максимуму. Наприклад, швидкість мереж, що підтримують стандарт 802.11b, зазвичай, становить не більше 50% їх теоретичного максимуму, тобто приблизно 5.5 Мбіт/сек. Відповідно, швидкість мереж, що підтримують стандарт 802.11a або 802.11g, зазвичай складає не більше 20 Мбіт/сек. Причинами невідповідності теорії і практики є надмірність кодування протоколу, перешкоди у сигналі, а також зміна відстані Хеммінга зі зміною відстані між приймачем і передавачем. Крім того, чим більше пристроїв у мережі одночасно беруть участь в обміні даними, тим нижча пропускну здатність мережі. До того ж, швидкість роботи будь-якої пари пристроїв істотно падає зі зменшенням рівня сигналу, тому часто найбільш ефективним засобом зростання швидкості для віддалених пристроїв є застосування антен з великим коефіцієнтом підсилення.

Стандартом передбачено дві категорії сервісів: сервіси розподілу та станційні сервіси. Перші з них пов'язані з управлінням станціями, що містяться в даній комірці, та взаємодією із зовнішніми станціями. Друга категорія відповідає за управління активністю в середині однієї комірки.

4.3.5. Стандарт IEEE 802.16

Мережі, що засновані на стандарті IEEE 802.16, складаються з таких основних частин — базових і абонентських станцій, а також обладнання, що зв'язує базові станції між собою, з постачальником сервісів і з Інтернетом.

Для з'єднання базової станції з абонентською використовується високочастотний діапазон радіохвиль від 1,5 до 11 ГГц. В ідеальних умовах швидкість обміну даними може досягати 70 Мбіт / с, при цьому не вимагається забезпечення прямої видимості між базовою станцією і приймачем. Між базовими станціями встановлюються з'єднання (прямої видимості), що використовують діапазон частот від 10 до 66 ГГц, швидкість обміну даними може досягати 120 Мбіт / с. При цьому принаймні одна базова станція підключається до мережі провайдера з використанням класичних дротових з'єднань. Однак, чим більшу кількість базових стандартів підключено до мереж провайдера, тим вища швидкість передачі даних і надійність мережі в цілому.

802.16 і 802.11 — технології спрямовані на вирішення абсолютно різних завдань. 802.16 — це система далекої дії, що покриває кілометри простору, яка зазвичай використовує ліцензовані спектри частот (хоча можливо і використання неліцензованих частот) для надання з'єднання із інтернетом типу точка-точка провайдером кінцевому користувачеві. Різні стандарти сімейства 802.16 забезпе-

чують різні види доступу — від мобільного (схожий з передачею даних із мобільних телефонів) до фіксованого (альтернатива провідникового доступу, при якому бездротове обладнання користувача прив'язане до розташування). 802.11 — це система більш короткої дії, що зазвичай покриває сотні метрів, яка використовує нелицензовані діапазони частот для забезпечення доступу до мережі. Зазвичай 802.11 використовується користувачами для доступу до їхньої власної локальної мережі, яка може бути не підключена до Інтернету. 802.16 і 802.11 мають зовсім різний механізм Quality of Service (QoS). 802.16 використовує механізм, заснований на встановленні з'єднання між базовою станцією та пристроєм користувача. Кожне з'єднання базується на спеціальному алгоритмі планування, який може гарантувати параметр QoS для кожного з'єднання. 802.11, у свою чергу, використовує механізм QoS, подібний тому, що в Ethernet, за якого пакети отримують різний пріоритет. Такий підхід не гарантує однаковий QoS для кожного з'єднання.

4.3.6. Технології бездротового зв'язку в промисловості

Найпопулярніші технології бездротового зв'язку наведені в табл. 4.3.

Таблиця 4.3

ПОРІВНЯЛЬНА ТАБЛИЦЯ СТАНДАРТІВ БЕЗДРОВОГО ЗВ'ЯЗКУ

Технологія	Стандарт	Використання	Пропускна здатність	Радіус дії	Частоти
Wi-Fi	802.11a	WLAN	до 54 Мбіт/с	до 100 метрів	5,0 ГГц
Wi-Fi	802.11b	WLAN	до 11 Мбіт/с	до 100 метрів	2,4 ГГц
Wi-Fi	802.11g	WLAN	до 108 Мбіт/с	до 100 метрів	2,4 ГГц
Wi-Fi	802.11n	WLAN	до 300 Мбіт/с (у перспективі до 450, а потім до 600 Мбіт/с)	до 100 метрів	2,4 — 2,5 або 5,0 ГГц
WiMax	802.16d	WMAN	до 75 Мбіт/с	6-10 км	1,5-11 ГГц
WiMax	802.16e	Mobile WMAN	до 30 Мбіт/с	1-5 км	2-6 ГГц
Bluetooth v. 1.1	802.15.1	WPAN	до 1 Мбіт/с	до 10 метрів	2,4 ГГц
Bluetooth v. 1.3	802.15.3	WPAN	від 11 до 55 Мбіт/с	до 100 метрів	2,4 ГГц
UWB	802.15.3a	WPAN	110-480 Мбіт/с	до 10 метрів	7,5 ГГц
ZigBee	802.15.4	WPAN	від 20 до 250 Кбіт/с	1-100 м	2,4 ГГц (16 каналів), 915 МГц (10 каналів), 868 МГц (один канал)
Wireless Hart	802.15.4	WPAN	250 Кбіт/с	до 250 м прямої видимості між пристроями	2,4 ГГц

4.3.6.1. ZigBee. Це назва набору протоколів мережного та вищих рівнів, що використовують малогабаритні, малопотужні радіопередавачі, засновані на стандарті IEEE 802.15.4. ZigBee націлена на додатки, яким потрібний більший час автономної роботи від батарей і велика безпека при менших швидкостях передачі даних. Радіус дії — до 75 м. Мережа здатна підтримувати до 254 клієнтів плюс один повнофункціональний майстер-пристрій (координатор). Основна особливість технології ZigBee полягає в тому, що вона при відносно невисокому енергоспоживанні підтримує не тільки прості топології бездротового зв'язку («точка-точка» і «зірка»), але й складні бездротові мережі з комірчастою топологією з ретрансляцією і маршрутизацією повідомлень (рис. 4.6, а). Области застосування нової технології — це бездротові мережі датчиків, ЖКГ, автоматизація житлових та споруджуваних приміщень, створення індивідуального діагностичного медичного обладнання, промисловий моніторинг і управління, побутова електроніка та периферія персональних комп'ютерів. Але використання технології ZigBee в АСУ ТП обмежена з ряду причин: технологія ZigBee має свій власний стек протоколів верхнього рівня, який суттєво відрізняється від протоколів промислового призначення; ZigBee заснована на базі стандарту IEEE 802.15.4 2003 (метод доступу до середовища CSMA/CA на каналному рівні моделі OSI), який не задовольняє підвищеним вимогам з надійності передачі даних для мереж промислової автоматики. Її можна використовувати для моніторингу та диспетчеризації та не можна використовувати для управління процесами в АСУ ТП.

4.3.6.2. Wireless HART. Цей стандарт створений на базі стандарту 802.15.4 2006. Він має стек протоколів верхнього рівня, який сумісний з промисловими протоколами HART і MODBUSRTU. Завдяки цьому бездротові сенсорні мережі Wireless HART можна підключати до шин HART і MODBUSRTU, а також до Industrial Ethernet. Важливе значення Wireless HART в тому, що хоч він і заснований на стандарті 802.15.4 2006 (у діапазоні 2400-2483,5 МГц), але має ряд особливостей. Так, арбітраж реалізований не за допомогою механізму CSMA/CA, що передбачений в 802.15.4, а завдяки множинному доступу з тимчасовим поділом (TDMA). Крім того, Wireless HART використовує механізм швидкого перемикання між 16 частотними каналами 802.15.4 (аналог технології розширення спектра за допомогою швидкої перебудови частот, FHSS). Це істотно підвищує захищеність і надійність передачі даних. Тому залежно від важливості оброблюваних даних Wireless HART гарантує надійність обміну інформацією, пов'язаної безпосередньо з керуванням технологічним процесом АСУ ТП (рис. 4.7).

4.3.6.3. Bluetooth. Радіозв'язок Bluetooth здійснюється в ISM-діапазоні, який використовується в різних побутових приладах і бездротових мережах (вільний від ліцензування діапазон 2,4-2,4835 ГГц). У Bluetooth застосовується метод розширення спектра зі стрибкоподібною перебудовою частоти (Frequency Hopping Spread Spectrum, FHSS). Метод FHSS простий у реалізації, забезпечує стійкість до ширококутових перешкод, а обладнання коштує недорого. Відповідно до алгоритму FHSS, у Bluetooth несуча частота сигналу стрибкоподібно змінюється 1600 разів за секунду (усього виділяється 79 робочих частот шириною в 1 МГц). Послідовність перемикання між частотами для кожного з'єднання є псевдовипадковою і відома тільки передавачу та приймачу, які кожні 625 мкс (один

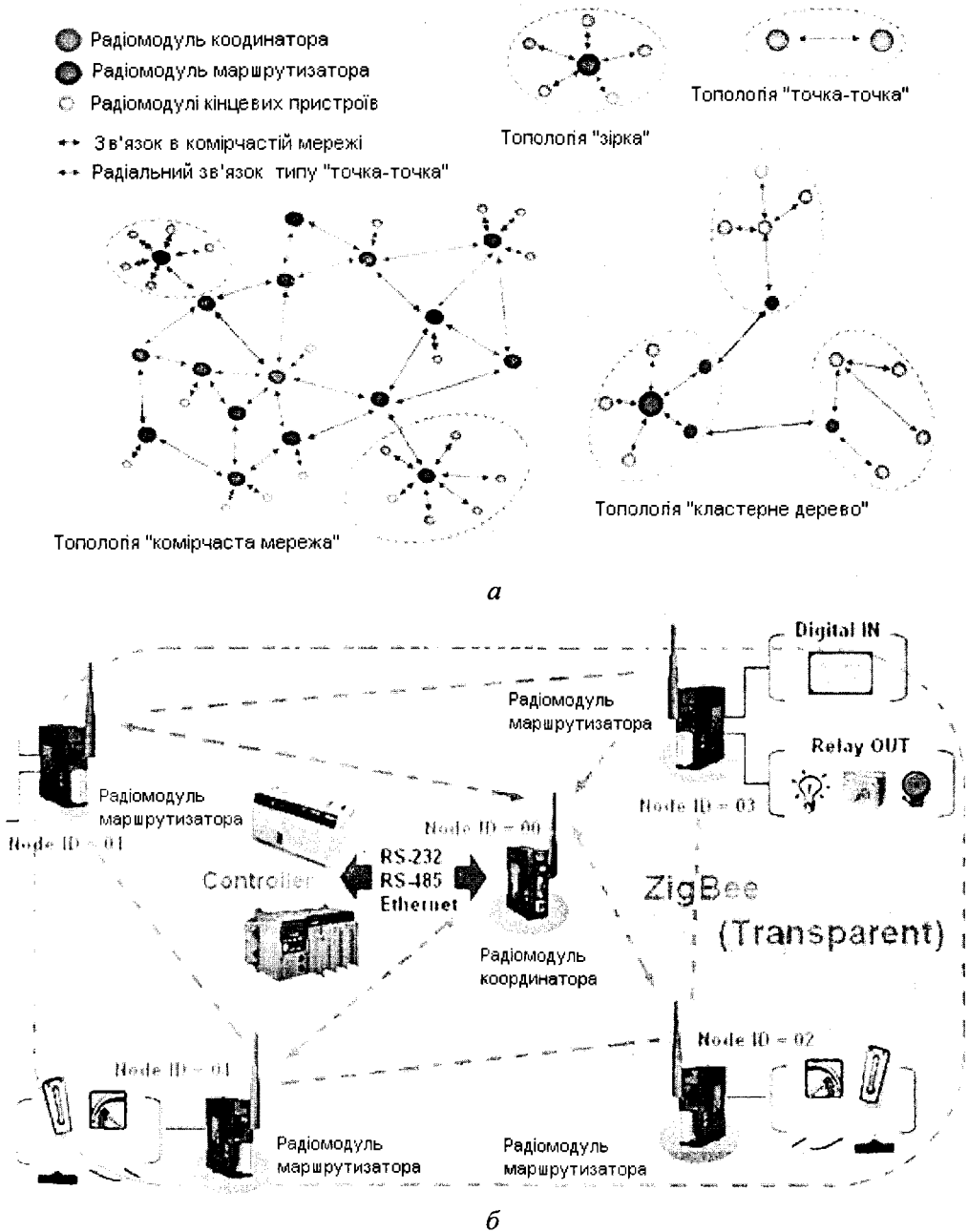
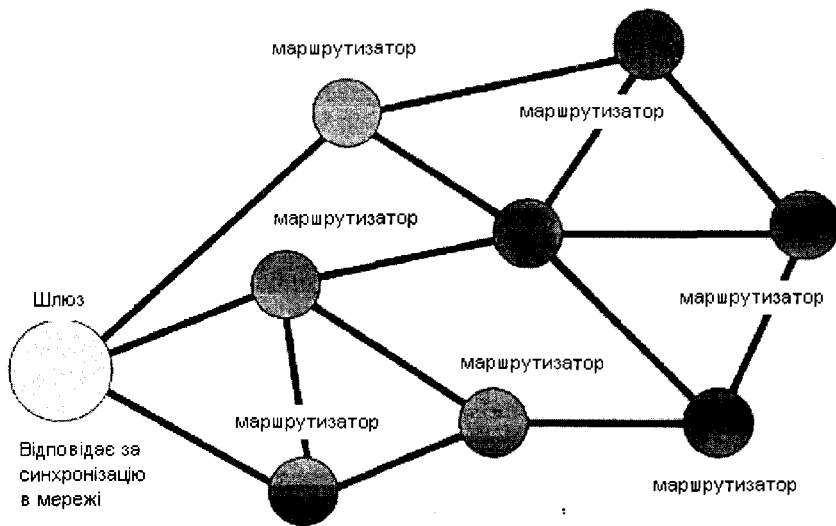
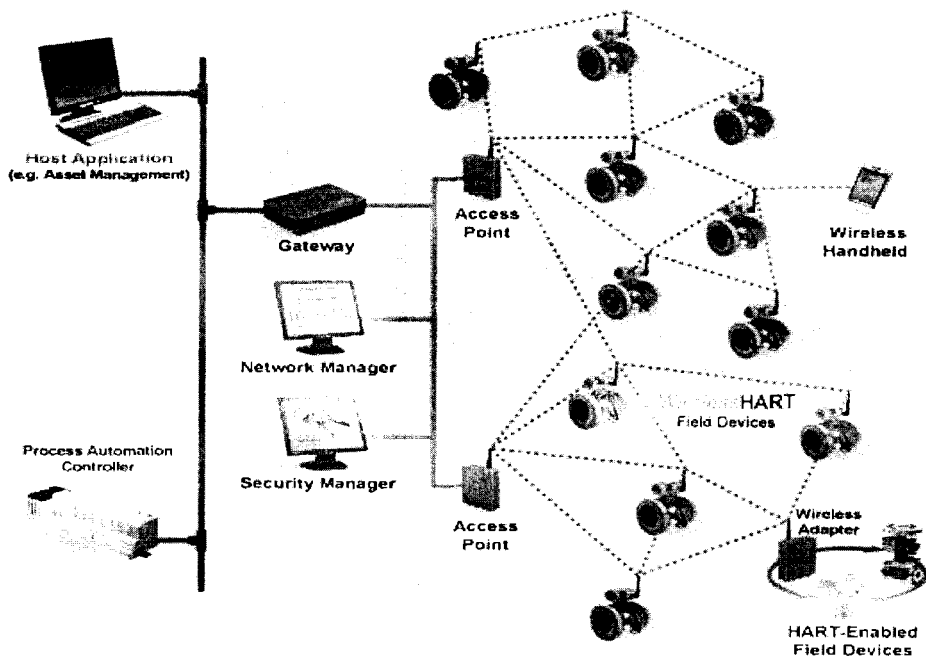


Рис. 4.6. Мережа ZigBee:
а — топологія; б — приклад використання



a



б

Рис. 4.7. Мережа Wireless HART:
a — топологія; б — приклад використання

часовий слот) синхронно перебудовуються з однієї несучої частоти на іншу. Таким чином, якщо поряд працюють кілька пар приймачів-передавачів, то вони не заважають один одному. Цей алгоритм є також складовою частиною системи захисту конфіденційності інформації, що передається: перехід відбувається за псевдовипадковим алгоритмом і визначається окремо для кожного з'єднання. При передачі цифрових даних та аудіосигналу (64 кбіт/с в обох напрямках) використовуються різні схеми кодування: аудіосигнал не повторюється (як правило), а цифрові дані у випадку втрати пакета інформації будуть передані повторно. Без завадостійкого кодування це забезпечує передачу даних зі швидкостями 723,2 кбіт/с зі зворотним каналом 57,6 кбіт/с або 433,9 кбіт/с в обох напрямках.

Залежно від класу обладнання пристрої Bluetooth працюють на відстанях 1, 10, 100 м, а залежно від специфікації — на швидкостях 1, 3, 24 Мбіт/с.

Щоб використовувати технологію бездротового зв'язку Bluetooth, пристрій повинен інтерпретувати певні профілі Bluetooth, які визначають можливості додатків і загальні моделі поведінки Bluetooth пристроїв для зв'язку з іншими пристроями Bluetooth. Є широкий асортимент профілів Bluetooth, які описують різні типи їх застосування, деякі з них, що використовуються в промисловості, наведені в табл. 4.4.

Таблиця 4.4

ДЕЯКІ ПРОФІЛІ BLUETOOTH

Basic Imaging Profile (BIP)	Профіль розроблений для пересилання зображень між пристроями і включає можливість зміни розміру зображення і конвертації в підтримуваний формат приймаючого пристрою.
Common ISDN Access Profile (CIP)	Профіль для доступу пристроїв до ISDN.
Device ID Profile (DID)	Профіль дозволяє ідентифікувати клас пристрою, виробника, версію продукту.
Dial-up Networking Profile (DUN)	Профіль надає стандартний доступ до Інтернету або іншого телефонного сервісу через Bluetooth. Базується на SPP, включає в себе команди PPP і AT, визначені в специфікації ETSI 07.07.
File Transfer Profile (FTP_profile)	Профіль забезпечує доступ до файлової системи пристрою. Включає стандартний набір команд FTP, що дозволяє отримувати список директорій, зміни директорій, отримувати, передавати та видаляти файли. Використовується OBEX, базується на GOEP.
General Audio / Video Distribution Profile (GAVDP)	Профіль є базою для A2DP і VDP.
Generic Access Profile (GAP)	Профіль є базою для всіх інших профілів.
Generic Object Exchange Profile (GOEP)	Профіль є базою для інших профілів передачі даних, базується на OBEX.
Human Interface Device Profile (HID)	Забезпечує підтримку пристроїв з HID (Human Interface Device), таких, як мишки, джойстики, клавіатури та інші. Використовує повільний канал, працює на зниженій потужності
LAN Access Profile (LAP)	LAN Access profile забезпечує можливість доступу Bluetooth-пристроїв до обчислювальних мереж LAN, WAN або Internet за допомогою іншого Bluetooth-пристрою, який має фізичне підключення до цих мереж. Bluetooth-пристрій використовує PPP над RFCOMM для встановлення з'єднання. LAP також допускає створення ad-hoc Bluetooth-мереж.

Object Push Profile (OPP)	Базовий профіль для пересилання «об'єктів», таких, як зображення, віртуальні візитні картки й ін. передачу даних.
Personal Area Networking Profile (PAN)	Профіль дозволяє використовувати протокол Bluetooth Network Encapsulation в якості транспорту через Bluetooth-з'єднання
Serial Port Profile (SPP)	Профіль базується на специфікації ETSI TS07.10 і використовує протокол RFCOMM. Профіль емулює послідовний порт, надаючи можливість заміни стандартного RS-232 бездротовим з'єднанням. Базовий для профілів DUN, FAX, HSP і AVRCP.
Service Discovery Application Profile (SDAP)	Профіль використовується для надання інформації про профілі, які використовує пристрій-сервер.
Wireless Application Protocol Bearer (WAPB)	Протокол для організації P-to-P (Point-to-Point) з'єднання через Bluetooth.

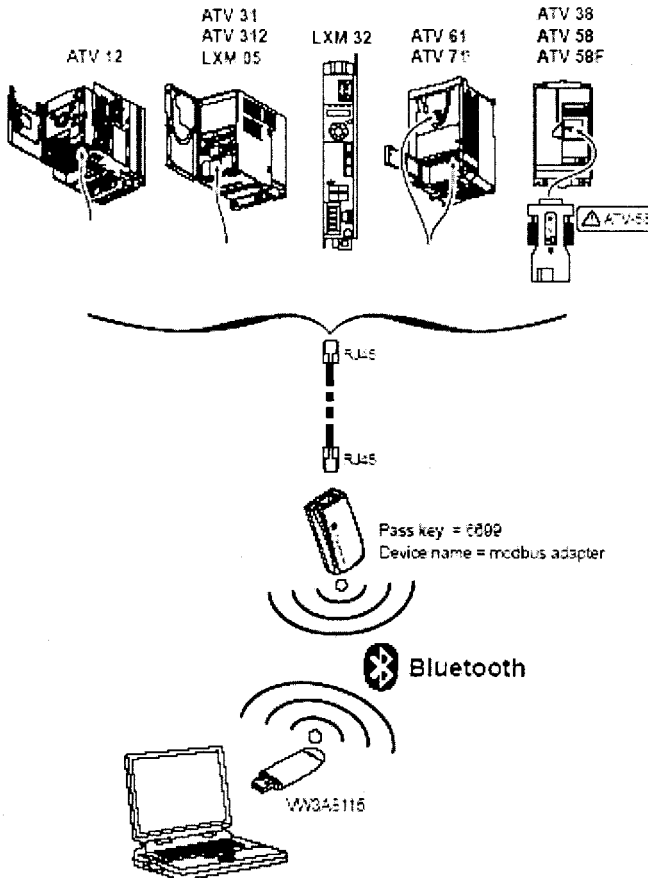


Рис. 4.8. Приклад використання Bluetooth для конфігурації обладнання за допомогою SPP-профілю

4.3.6.4. Wi-fi. Сьогодні, говорячи про промисловий Wi-fi, мають на увазі гібридну архітектуру мережі, що складається з дротових та бездротових сегментів мереж, причому в бездротових сегментах на фізичному та MAC-рівнях використовується стандарт 802.11. На верхніх рівнях використовуються протоколи дротових промислових мереж. Інтеграція двох сегментів відбувається за допомогою спеціалізованого обладнання — репітерів, мостів, шлюзів.

Використання бездротових мереж WLAN у промисловості забезпечує підвищену гнучкість, мобільність, швидке встановлення системи. Велика перевага інсталяції WLAN полягає у безпосередньому доступі до датчиків на рівні IP. До того ж, інтелектуальні регулятори часто пропонують інтегрований сервер Web для визначення параметрів, за допомогою якого стає можливим бездротовий доступ до датчиків за допомогою рішення WLAN. Останні дані про стан і діагностику можна, наприклад, передавати на ноутбук для аналізу, при цьому сам користувач перебуває за межами промислового залу. Також промисловість випускає мости доступу до промислових дротових LAN, що дозволяє об'єднати окремі локальні мережі в єдину інфраструктуру підприємства та забезпечити вихід в Інтернет.

Зазначимо, що, крім підвищених стандартів захисту (Ingress Protection, IP), промисловість пред'являє особливі вимоги до готовності бездротової мережі — як стосовно доступності мережі для клієнтів, так і надійності. І те й інше забезпечується лише шляхом проведення попередніх вимірювань ефіру. Тільки так можна гарантувати, що всі клієнти у заздалегідь визначених областях будуть отримувати безперервний доступ до мережі, а мобільні клієнти прозорим чином стануть переміщатися з однієї зони в сусідню (роумінг). Завдяки плануванню відповідних перекриттів радіокомірок навіть за відмови окремих точок доступу бездротова мережа продовжить своє функціонування. Також необхідна підвищена безпека мережі, що гарантується використанням стандарту WPA2 (802.11i), що використовує більш складне шифрування даних, ніж WPA.

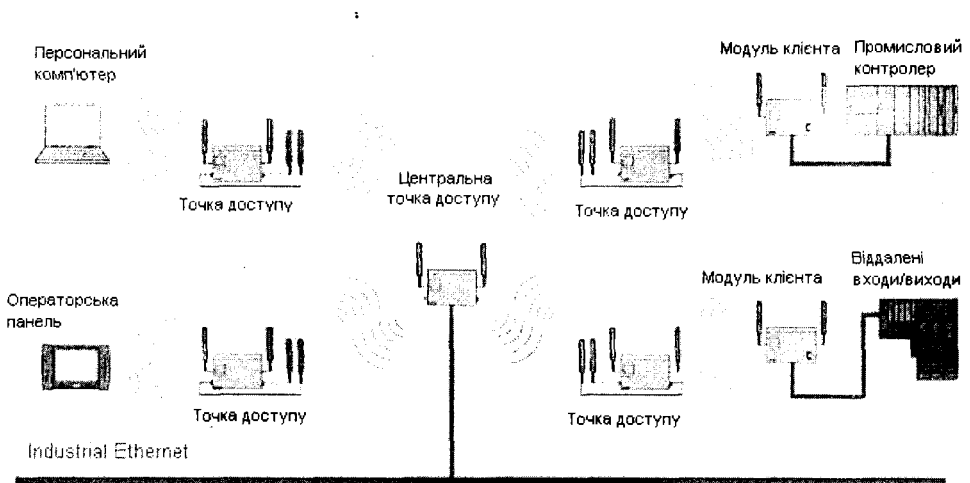


Рис. 4.9. Приклад використання Wi-fi

4.4. Особливості бездротових модемів з використанням мобільного зв'язку

4.4.1. Коротка характеристика мобільного зв'язку в Україні

Останнім часом дуже стрімко розвивається безпроводний спосіб зв'язку з використанням мереж операторів мобільного зв'язку.

Нині стандартів мобільного зв'язку досить багато, тому зупинимось лише на тих, що діють в Україні та підтримують пакетну передачу даних.

Станом на 2010 рік в Україні працюють такі оператори:

- Мобільні оператори GSM — Голден Телеком GSM, life:), Beeline, Київстар, МТС.
- Мобільні оператори CDMA — PEOPLEnet, Інтертелеком, CDMA UKRAINE, Velton telecom.
- Мобільні оператори UMTS — Utel.
- Віртуальні мобільні оператори GSM — Jeans, Djuce, Екотел, Мобілич.

4.4.2. Основні стандарти мобільного зв'язку

4.4.2.1. GSM. GSM (Global System for Mobile Communications — глобальна система мобільного зв'язку) — міжнародний стандарт для мобільного цифрового стільникового зв'язку з розділенням каналу за принципом TDMA та високим рівнем безпеки за рахунок шифрування з відкритим ключем. Стандарт був розроблений під патронатом Європейського інституту стандартизації електрозв'язку (ETSI) наприкінці 80-х років. Більшість мереж GSM працюють у діапазоні 900 МГц або 1800 МГц.

Послуги, що можуть надаватися мережами GSM:

- Передача голосової інформації.
- Послуга передачі даних (синхронний та асинхронний обмін даними, в тому числі пакетна передача даних — GPRS).
- Передача коротких повідомлень (SMS).
- Передача факсів.

Саме друга та третя послуги використовуються в промислових системах зв'язку.

SMS (Short Message Service — служба коротких повідомлень) — послуга обміну (передачі і прийому) короткими текстовими повідомленнями в телекомунікаційних мережах.

GPRS (General Packet Radio Service — пакетний радіозв'язок загального користування) — надбудова над технологією мобільного зв'язку GSM, що здійснює пакетну передачу даних. GPRS дозволяє користувачеві мережі стільниково-

го зв'язку обмінюватися даними з іншими пристроями в мережі GSM і із зовнішніми мережами, в тому числі Інтернет. GPRS передбачає тарифікацію як за обсягом переданої/отриманої інформації, так і за часом, проведеним онлайн. При використанні GPRS інформація збирається в пакети і передається через не використовувані в цей момент голосові канали, така технологія передбачає більш ефективно використання ресурсів мережі GSM. При цьому, що є пріоритетом передачі — голосовий трафік або передача даних — обирається оператором зв'язку. Існують різні класи GPRS, що розрізняються бітровою швидкістю і можливістю суміщення передачі даних з одночасним голосовим викликом. Технологія GPRS використовує GMSK-модуляцію. Залежно від якості радіосигналу, дані, що пересилаються по радіоэфіру, кодуються за однією з 4-х кодових схем (CS1-CS4). Кожна кодова схема характеризується надмірністю кодування і перешкодостійкістю та обирається автоматично, залежно від якості радіосигналу. Протокол GPRS прозорий для TCP/IP, тому інтеграція GPRS з Інтернетом непомітна кінцевому користувачу. Пакети можуть мати формат IP або X.25, при цьому не має значення, які протоколи використовуються поверх IP, тому є можливість використання будь-яких стандартних протоколів транспортного і прикладного рівнів, що застосовуються в Інтернеті (TCP, UDP, HTTP, HTTPS, SSL, POP3, XMPP та ін.) Також при використанні GPRS мобільний телефон виступає, як клієнт зовнішньої мережі і йому присвоюється IP-адреса (постійна або динамічна).

4.4.2.2. CDMA. CDMA (Code Division Multiple Access — множинний доступ із кодовим розподілом каналів) — одна з можливих технологій мультиплексування, тобто одночасної передачі даних у спільному діапазоні (так званому каналі) радіочастот. Для забезпечення множинного доступу до каналу дані кодуються спеціальним кодом, асоційованим із кожним каналом, і використовують властивості конструктивної інтерференції спеціальних кодів, щоб здійснювати мультиплексування.

На основі технології CDMA було розроблено стандарт мобільного зв'язку CDMA2000. Тут історично склалося декілька етапів. При побудові системи мобільного зв'язку на основі технології CDMA2000 1X перша фаза забезпечує передачу даних зі швидкістю до 153 кбіт/с, що дозволяє надавати послуги голосового зв'язку, передачу коротких повідомлень, роботу з електронною поштою, Інтернетом, базами даних, передачу даних та фотографій. Перехід до наступної фази CDMA2000 1xEV-DO відбувається при використанні тієї ж смуги частот 1,23 МГц, бітрової швидкості — до 2,4 Мбіт/с у прямому каналі і до 153 кбіт/с у зворотному, що робить цю систему зв'язку такою, що відповідає вимогам 3G і дає можливість надавати найширший спектр послуг, аж до передачі відео в режимі реального часу. Наступною фазою розвитку стандарту в напрямку збільшення мережної ємності й передачі даних є 1xEV-DO Rev A: передача даних зі швидкістю до 3,1 Мбіт/с у напрямку до абонента і до 1,8 Мбіт/с — від абонента.

4.4.2.3. UMTS. UMTS (Universal Mobile Telecommunications System — універсальна система мобільного зв'язку) — технологія стільникового зв'язку, що

належить до покоління 3G. Як спосіб передачі даних через повітряний простір використовується технологія W-CDMA (*Wideband Code Division Multiple Access* — широкопasmовий множинний доступ із кодовим розподілом каналів, який використовує дві широкі смуги радіочастот по 5 МГц). Ця технологія використовує протоколи HSDPA та HSUPA — протоколи високошвидкісного приймання та передавання пакетних даних, що включають два типи модуляції QPSK та 16QAM, за рахунок чого підвищується швидкість передачі до 7,2 Мбіт/с.

В усіх випадках користування послугами мобільного зв'язку не обходиться без мобільного оператора, а отже передбачає допоміжні фінансові витрати.

Приклад 4.2. Передача SMS-повідомлень.

Завдання. Передати SMS-повідомлення з контролера Unitronics до мобільного телефону при спрацюванні внутрішнього біту.

Рішення. Так як PLC Unitronics має два порти RS-232/RS-485 з роз'ємами RG-11, а перший використовується для програмування, то перемикаємо порт 2 в режим RS-232, а схема модемного кабеля має вигляд як на рис. 4.11.

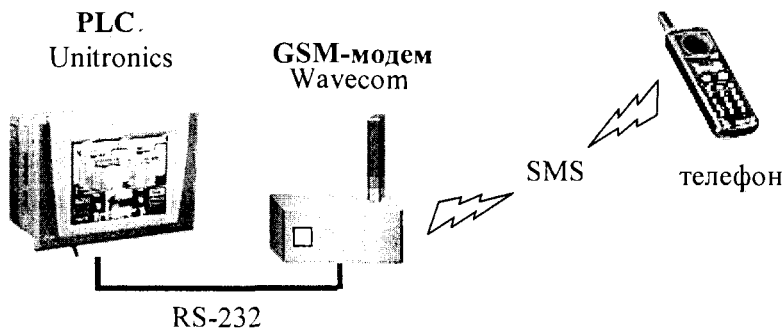


Рис. 4.10. Структурна схема системи до прикладу 4.2

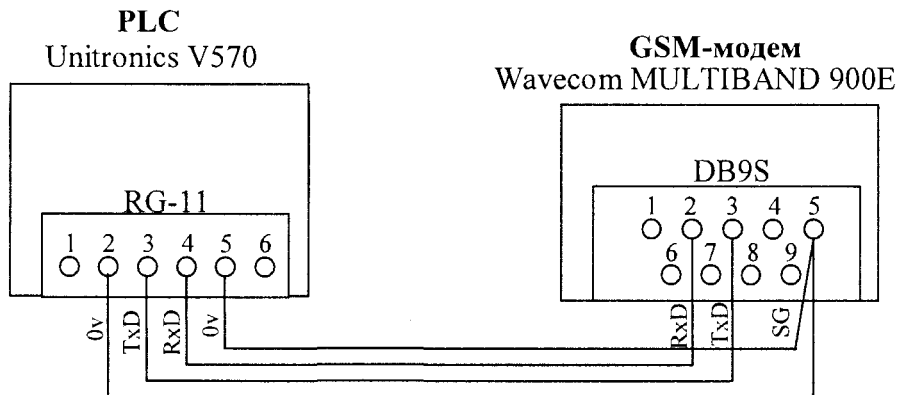


Рис. 4.11. Схема з'єднань до прикладу 4.2

Попередньо перед підключенням необхідно підготувати модем: для цього з'єднують модем з ПК звичайним модемним кабелем та запускають на виконання в програмному забезпеченні VisiLogic Connection/Modem Services, налаштовують відповідні параметри послідовного порту, як показано на рис. 4.12, та натискають Prepare PLC-side modem (список АТ-команд попередньої підготовки модему користувач може задати самостійно). Далі складають програму посилки SMS-повідомлення (див. рис. 4.13).

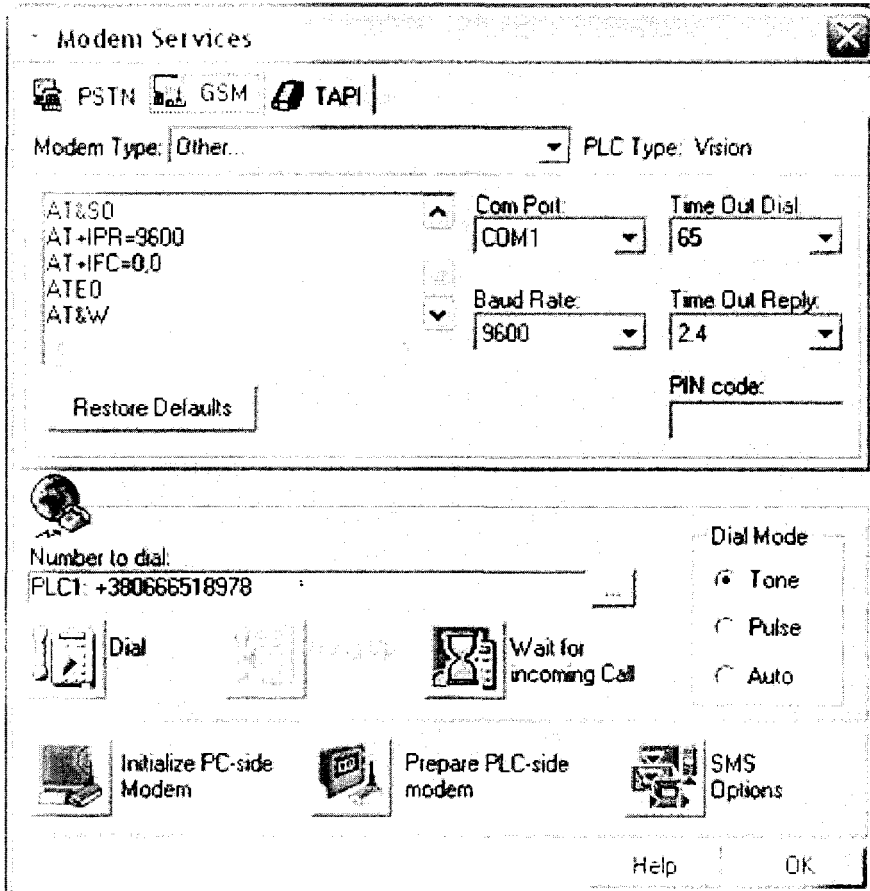


Рис. 4.12. Попередня підготовка модема

The image displays a PLC programming interface with three rungs of a ladder logic program and three associated configuration dialog boxes.

- Rung 5:** Labeled "Ініціалізація port2 PLC". It features a normally open contact labeled "EN ENO COM INIT 2" and a coil labeled "(S)".
- Rung 6:** Labeled "Конфігурація SMS-повідомлень". It contains two normally open contacts: "MB 31 Scan Status" and "EN ENO SMS CONFIG SMS_1", both leading to a coil labeled "(S)".
- Rung 7:** Labeled "Відправка SMS-повідомлення при спрацюванні MB15". It contains two normally open contacts: "MB 33" and "MB 50 Modem Busy (GSM)", and a normally closed contact "MB 33". These contacts lead to a coil labeled "(R)".
- Com Initialize Dialog:** Shows "Com Port" set to "COM2", "Baud Rate" at "3600", "RS232 Time Out" at "2 sec", and "Modem Type" set to "GSM".
- SMS Configuration Dialog:** Shows "Name" as "SMS_1", "Com Port" as "COM2", "Modem Busy (GSM)" set to "MB 50 Modem Busy (GSM)", and "Status Messages" set to "MI 10 Status Message".
- SMS Send Dialog:** Shows "Name" as "SMS_1", "Com Port" as "COM2", "Send Status" set to "MB 16 Send Status", and "Send Fail Bitmap" set to "MI 16 Send: Fail Bitmap".

Рис. 4.13. Фрагмент програми посилки SMS-повідомлення



Контрольні запитання до розділу 4

1. Що таке символний спосіб обміну?
2. Як формується кадр при символній передачі інформації?
3. Розкажіть про призначення та принципи дії модема.
4. Класифікація модемів.
5. AT-команди модемів: принципи використання.
6. Використання бездротового способу обміну в промисловості.
7. Основні концепції при бездротовій передачі інформації.
8. Класифікація бездротових мереж за територіальним розподілом.
9. 802.15: топологія, призначення, класифікація, основні концепції.
10. Розкажіть про функціонування мереж на основі стандарту 802.15 в промисловості.
11. 802.11: топологія, призначення, класифікація, основні концепції.
12. Як використовуються мережі на основі стандарту 802.11 в промисловості.
13. 802.16: топологія, призначення, основні концепції.
14. Назвіть особливості бездротових модемів стільникового зв'язку.
15. Назвіть основні послуги GSM-зв'язку, що використовуються в промисловості.
16. Наведіть приклад використання GSM-зв'язку в промисловості.
17. Порівняйте основні бездротові стандарти, що використовуються в промисловості.

ІНТЕРФЕЙС AS-i

5.1. Загальні відомості

AS-i (Actuator and Sensor interface) — інтерфейс для підключення датчиків та виконавчих механізмів. Є альтернативним рішенням використанню безпосереднього підключення польових пристроїв вимірювання та регулювання з дискретними сигналами. Велика кількість проводів замінюється на два, за якими передається інформація і живлення пристроїв одночасно. Надзвичайно простий та зручний монтаж, а також завадостійкість зробили цю шину дуже популярною у багатьох країнах світу.

Електричні і механічні характеристики даного інтерфейсу були розроблені на початку 90-х років за участі одинадцяти компаній, які спеціалізуються в галузі бінарних датчиків і виконавчих механізмів. AS-i є відкритим гетерогенним стандартом і описаний в EN 50295. За просування і розповсюдження AS-i систем відповідає «Асоціація підтримки інтерфейсів для підключення бінарних виконавчих механізмів і датчиків в шину» або просто Асоціація AS-i (AS-i Association).

Активне просування такої молоді технології пояснюється цілим рядом причин економічного і технічного характеру:

1. Мінімізація витрат на кабельну систему за рахунок: мінімізації або повного виключення (для інтелектуальних датчиків) ліній з нецифровими сигналами; використання дешевого звичайного двохпровідного кабелю; виключення окремих кабелів електроживлення ведених пристроїв (датчиків і виконавчих механізмів).

2. Економія допоміжного обладнання (клем, кабельних вводів і т. інше).

3. Значна економія фінансових і людських ресурсів при проведенні монтажних робіт.

4. Можливість підключення як інтелектуальних (з вбудованим AS-i), так і звичайних датчиків.

5. Надзвичайно малий час опитування (5 мс на всі пристрої), який дозволяє керувати обладнанням у режимі реального часу.

6. Висока ступінь захисту обміну даними між вузлами.
7. Орієнтація на найнижчий рівень автоматизації.
8. Великий вибір ведених пристроїв для встановлення у будь-яких умовах.
9. Наявність широкої номенклатури апаратних засобів для інтеграції мереж AS-і інтерфейсу у існуючі структури АСУТП на будь-якому рівні.

Шину AS-і не описують в контексті OSI. Але умовно її б можна було показати на двох рівнях: фізичному та каналному. В AS-і використовуються профілі для можливості підключення різноманітних пристроїв, які дозволяють правильно ідентифікувати отримувані дані та формувати дані для відправки. За рахунок використання профілів пристроїв, обмін відбувається наперед визначеними даними.

5.2. Реалізація фізичного рівня

Фізичний рівень забезпечує передачу інформації разом з живленням за рахунок спеціального методу кодування бітів, яке використовує комбінацію методу диференційного манчестерського кодування та модуляції струму, яка базується на \sin^2 -хвилях. Такий спосіб кодування дозволяє синхронізувати джерело з приймачем та підвищує надійність передачі.

Для реалізації AS-і шини необхідно використовувати таке обладнання:

- Ведучий мережний пристрій (Master);
- Ведені пристрої (Slave), які залежно від конструкції поділяються на:
 - а) модулі AS-і інтерфейсу, за допомогою яких до шини підключаються звичайні датчики і виконавчі механізми;
 - б) датчики і виконавчі механізми з убудованим AS-і інтерфейсом:
- джерело живлення;
- мережний кабель;
- повторювачі, підсилювачі, шлюзи і інші додаткові пристрої.

Для з'єднання компонентів можна використати будь-який двохжильний кабель, який вибирається залежно від споживаної потужності. Немає необхідності у використанні екранованого кабелю чи витої пари. Однак більш просте і швидке виконання монтажу системи можна досягнути за рахунок використання спеціального плоского кабелю AS-і інтерфейсу, який являє собою трапецієвидний двохжильний кабель $2 \times 1,5 \text{ мм}^2$ з асиметричним профілем («жовтий кабель») (рис. 5.1). Останнє виключає можливість підключення вузлів за неправильною полярністю. Підключення до даного кабелю відбувається методом проколу ізоляції гострими контактами спеціальних контактних пристроїв AS-і (модулі, коробки підключення). Тому немає необхідності різати кабель та зачищати ізоляцію в місці установки нового вузла. Окрім того, за рахунок гумової оболонки структура кабелю має особливість до самовідновлення. Тобто, якщо контактний пристрій вилучається, прорізані отвори затягуються, що зберігає ступінь захисту IP67. Зручність монтажу з таким кабелем і зробила AS-і таким популярним, тому її ще називають «*жовтим кабелем*».

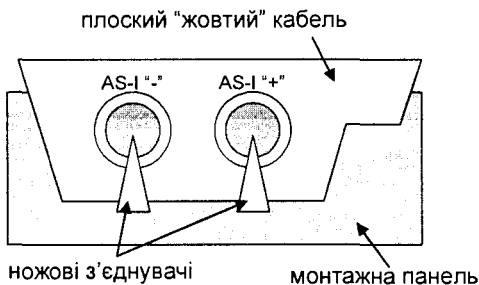


Рис. 5.1. «Жовтий кабель» AS-i шини

AS-i інтерфейс може мати шинну, зіркову або деревовидну топологію. Максимальна довжина всієї мережі — 100 м без повторювача, 300 м — з двома повторювачами. Бітова швидкість фіксована і дорівнює 167 Кбіт/с.

Реалізується AS-i інтерфейс за допомогою мікросхем ASIC (Application Specific Integrated Circuits), які інтегруються у пристрої, що підключаються до шини. Але можливе підключення звичайних датчиків і виконавчих механізмів за допомогою спеціальних модулів, які мають мікросхему ASIC.

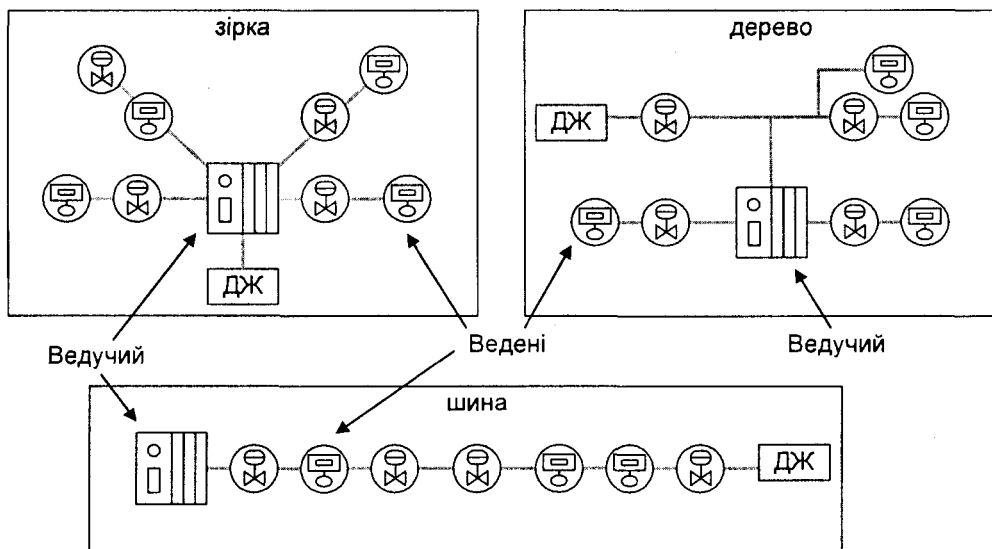


Рис. 5.2. Топологія AS-I мережі (ДЖ- джерело живлення)

У мережах AS-i по одній парі проводів передається інформаційний сигнал і живлення ведених пристроїв. Тому для живлення AS-i шини використовуються спеціальні джерела живлення напругою 30 В різної потужності.

Для збільшення довжини лінії зв'язку можуть використовуватись повторювачі і подовжувачі лінії, які дозволяють продовжити довжину лінії зв'язку ще на 100 м. Можливе сполучення двох повторювачів або одного повторювача і одного

подовжувача (рис. 5.3, рис. 5.4), що забезпечує загальну довжину одного сегмента до 300 м.

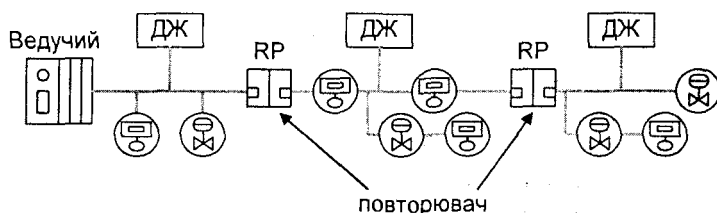


Рис. 5.3. Топологія AS-i інтерфейсу з двома повторювачами

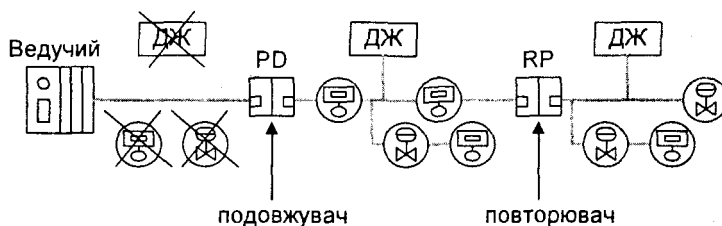


Рис. 5.4. Топологія AS-i інтерфейсу з повторювачем і подовжувачем

На відміну від інших промислових мереж, AS-i інтерфейс не вимагає встановлення термінаторів на кінцях мережі.

Нагадаємо, що до мережі AS-i можуть підключатися як датчики і виконавчі механізми з убудованим AS-i інтерфейсом (інтелектуальні пристрої), так і звичайного виконання. Перші підключаються до мережі безпосередньо, другі — через модулі вводу-виводу (МВВ). Це важливо пам'ятати, особливо на стадії проектування мережі, тому що до одного сегмента мережі AS-i інтерфейсу можна підключати менше інтелектуальних ведених пристроїв, але можливості інформаційного обміну з ними більше.

Різновиди МВВ за типом пристроїв, які підключаються, можуть мати:

- 4 входи;
- 4 виходи;
- 2 входи + 2 виходи;
- 4 входи + 4 виходи (для специфікації 2.0);
- 2 аналогових входи;
- 2 аналогових виходи;
- 4 аналогових входи від термодатчиків Pt 100;
- та інші.

Пристрої випускаються з різними конструкціями монтажу та ступенем захисту.

На рис. 5.5 наведений приклад AS-і мережі.

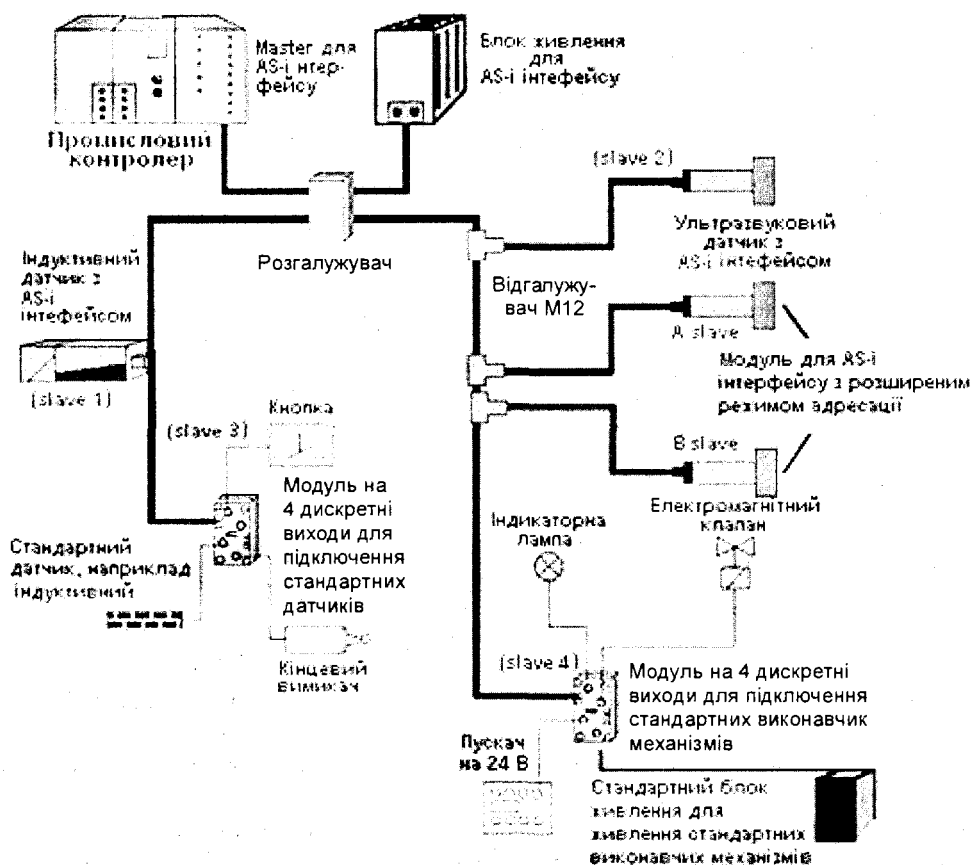


Рис. 5.5. Приклад AS-і мережі

5.3. Принципи функціонування

AS-і інтерфейс — мережа з регламентованим методом доступу Ведучий/Ведений. Ведучий виконує такі функції:

- ініціалізує систему;
- ідентифікує підключені Ведені;
- забезпечує цілісність нормального циклічного обміну даними;
- забезпечує параметризацію Ведених;
- виконує системну діагностику шини;
- забезпечує переконфігурування адрес Ведених за необхідності.

У режимі нормальної роботи системи Ведучий надсилає кадр з командою кожному Веденому в порядку їх адреси, в якому вказує значення 4-х дискретних виходів Веденого. Той відповідає кадром-відповіддю, в якому надсилає інформацію про стан 4 дискретних входів. Знаючи номери (адреси) всіх присутніх Ведених на шині (до 31), Ведучий опитує їх усіх (якщо вони активовані), а потім починає цикл заново. Таким чином, до одної шини AS-i можна підключити до $124(31 \cdot 4)$ дискретних виходів та до $31 \cdot 4(124)$ дискретних входів одночасно. Оскільки бітова швидкість постійна, а відповідь Веденого не залежить від циклу роботи його програми, то час опитування всіх Ведених (час циклу AS-i) точно вираховується і не перевищує 5 мс (для версії V2 – 10 мс).

На рис. 5.6 показана спрощена схема обміну даними між Ведучим та Веденим. Комунікаційний процесор Ведучого шини автоматично (без прив'язки до циклу програми користувача) циклічно опитує всіх Ведених. Комунікаційний процесор Веденого займається відновленням вихідних буферів для ЦДП (цифро-дискретних перетворювачів) та відсилає дані з буферів ДЦП (дискретно-цифрових перетворювачів). Програма користувача (ПРК) Ведучого отримує та відсилає дані через буфер, використовуючи внутрішній інтерфейс. Така конструкція дозволяє організувати обмін на шині без участі роботи центрального процесора (ЦП), тобто розпаралелити роботу програми та шини, а також робить прозорою роботу Ведучого з даними Ведених.

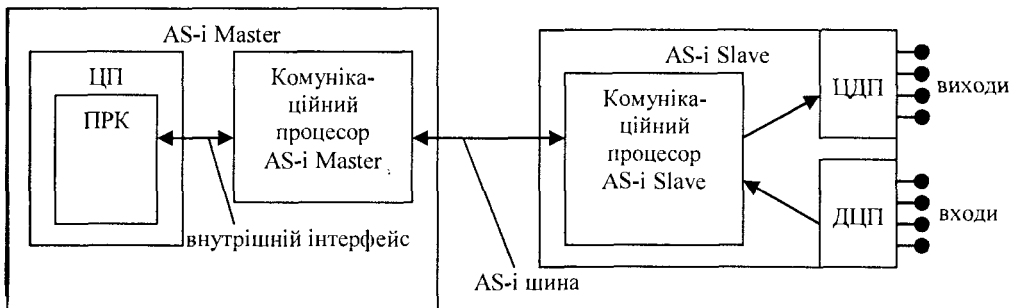


Рис. 5.6. Обмін даними між Ведучим та Веденим на шині AS-i

На рисунку 5.7 більш детально відображена внутрішня структура Ведучого та Веденого, зокрема вказані блоки даних (буфери) та їх типи. У кожного Веденого є буфер для відображення стану входів та виходів (дані процесу), який відображається у відповідному буфері Ведучого. При відправці команди на Ведений Ведучий розміщує його вихідні дані в полі даних кадру. Ведений, отримавши кадр, відновлює вихідні дані свого буфера і відправляє входні дані у кадрі-відповіді. Таким чином, AS-i функціонує за моделлю Клієнт-Сервер обміну ідентифікованими даними (Polling).

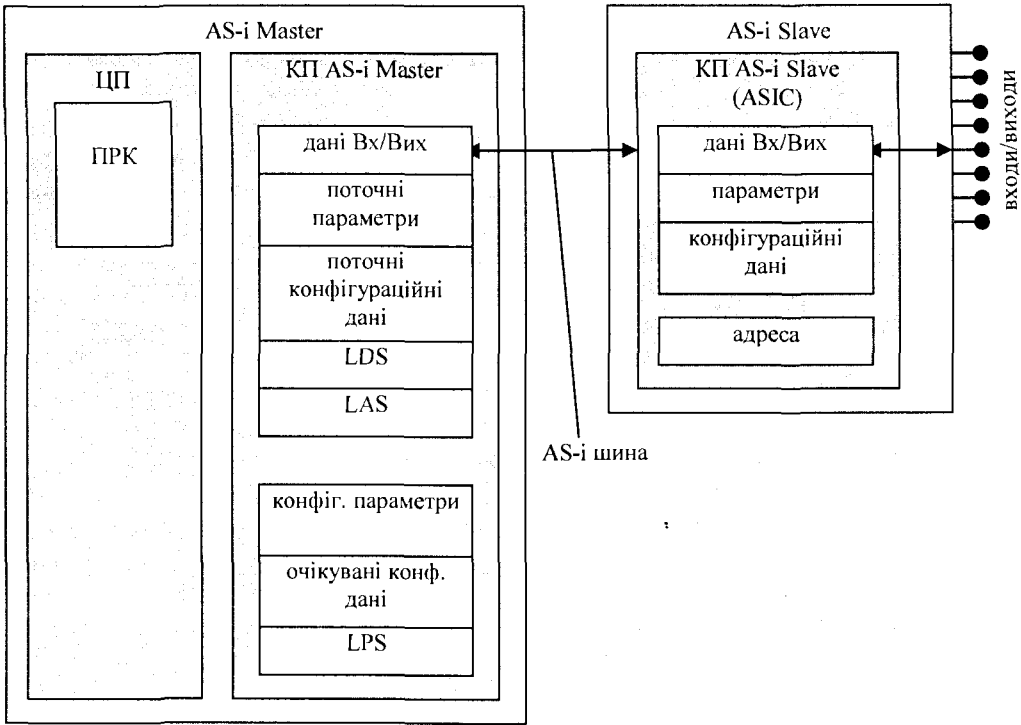


Рис. 5.7. Таблиці даних (буфери) в пристроях AS-i

Окрім буферів, відображення входів/виходів (дані процесу) є також інші буфери, які коротко перерахуємо. Ведучий AS-i має такі блоки даних:

- образи даних, які можуть змінитися під час роботи:
 - дані Вх./Вих. (дані процесу);
 - поточні параметри — містять образ поточних параметрів AS-i Slave (параметричні дані);
 - поточні конфігураційні дані — містять конфігурацію входів/виходів та ідентифікаційні коди всіх підключених Ведених пристроїв, після їх зчитування з цих пристроїв (параметричні дані);
 - LDS (List of Detected Slaves) — список усіх знайдених Ведених пристроїв на шині;
 - LAS (List of Active Slaves) — список активованих Ведених пристроїв; обмін даними проводиться тільки з активованими Веденими;
 - конфігураційні дані (параметричні дані), які вказує користувач при конфігурації Веденого в системі; ці дані необхідні для порівняння їх з образами реальних даних знайдених Ведених та їх конфігурації; вони, як правило, містяться в енергонезалежній пам'яті Ведучого:
 - конфігураційні параметри;
 - очікувані конфігураційні дані;

- LPS (List of Prospective Slaves) — список очікуваних Ведених на шині. В ньому містяться номери (адреси) тих Ведених, які система очікує знайти на шині; Ведений на шині має такі буфери:
 - дані Вх./Вих.(дані процесу);
 - параметри (параметричні дані);
 - конфігураційні дані (параметричні дані) — конфігурація входів/виходів та ідентифікаційні коди даного Веденого («I/O Code» та «ID Code»);
 - адреса — адреса Веденого.

5.4. Етапи роботи шини AS-i

Ведучий на шині AS-i не відразу може приступити до циклічного опитування Ведених. На початку він ініціалізує свої буфери та настроює мережні вузли. На рис. 5.8 показані стадії та етапи роботи шини.

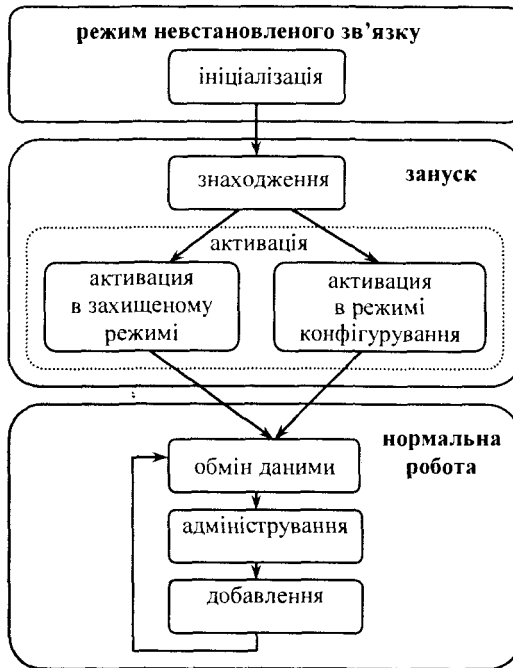


Рис. 5.8. Етапи роботи шини

Стадія ініціалізації, яку також називають режимом невстановленого зв'язку (offline), відбувається після включення живлення Ведучого. Під час ініціалізації всі вхідні та вихідні дані у буфері Вх./Вих. обнулюються, сконфігуровані (очікувані) параметри копіюються в таблицю поточних параметрів. Таблиці LDS та LAS також обнулюються.

Стадія запуску («start-up») відбувається в два етапи: знаходження та активація. Спочатку Ведучий визначає, які Ведені підключені до кабелю AS-і і якого типу. Тип Веденого вказується в конфігураційних даних, які туди записуються при його виготовленні. Це «I/O Code» та «Identification (ID) Code», які задають профіль Веденого пристрою. Таким чином, з 1 по 31 Ведені будуть опитані Ведучим і, якщо вони існують, їх конфігураційні дані («I/O Code» та «ID Code») будуть додані до списку знайдених Ведених пристроїв LDS.

Після того, як Ведені знайдені, вони повинні активізуватися Ведучим. В стадії нормальної роботи тільки активні Ведені беруть участь в обміні даними процесу. При їх активації розрізняють два режими роботи Ведучого пристрою:

- Ведучий у **режимі конфігурування** (configuration mode): активуються всі знайдені Ведені (за винятком тих, що мають адресу 0). В даному режимі можливе читання фактичних значень і збереження їх для конфігурування;

- Ведучий у **захисному режимі** (protected mode): активуються тільки ті Ведені, конфігурація яких («I/O Code» та «ID Code») відповідає очікуваній конфігурації. Тобто Ведучий активує тільки тих Ведених, які містяться в списку очікуваних Ведених (LPS) а їх очікувана конфігурація відповідає фактичній (дійсній) конфігурації.

Адреси всіх активованих Ведених Ведучий заносить до списку LAS. Після активації всіх можливих Ведених Ведучий вступає в **стадію нормальної роботи**, яка складається з трьох етапів: етап обміну даними; етап адміністрування; етап додання.

В стадії нормальної роботи, на етапі обміну даними, Ведучий циклічно передає дані (значення виходів) активним Веденим та отримує від них відповідь (значення входів). У випадку виникнення помилки Ведучий повторює транзакцію.

Після обміну даними з усіма активними Веденими наступає етап адміністрування, під час якого Ведучий може відправити певну команду Веденому, наприклад на запис параметричних даних (калібровка, порогове значення), на зміну адреси Веденого і т.д. Ці команди ациклічні і виконуються по одній на повний цикл.

Після завершення етапу управління настає етап додання, під час якого Ведучий шукає нових Ведених. З кожним новим циклом він буде давати запит на знаходження одного Веденого, якого немає в списках активних. Таким чином, для 100-відсоткового успіху знаходження нового пристрою на AS-і необхідно максимум 31 цикл. У випадку виявлення такого необхідно буде ще 3 цикли для того, щоб: прочитати «I/O Code», прочитати «ID Code», відправити параметричні дані. Завдяки такому механізму пристрої, які були тимчасово недоступні включатимуться знову.

Таким чином, обмін даними процесу проходить циклічно за моделлю Клієнт-Сервер ідентифікованого обміну (полінг), а обмін параметричними даними ациклічно за моделлю Клієнт-Сервер обміну повідомленнями.

5.5. Формат кадру AS-і

Кадри в AS-і від Ведучого до Веденого мають фіксовану довжину незалежно від типу. Так само не змінюється формат відповіді (рис. 5.9).

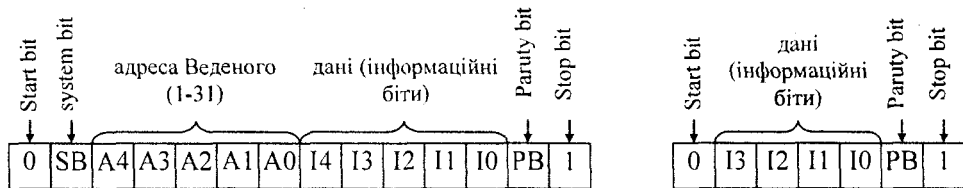


Рис. 5.9. Формат кадру в AS-i:
команда від Ведучого — ліворуч; відповідь Веденого — праворуч

Як видно з рисунка, для адреси виділено 5 бітів, що дозволяє адресувати 32 пристрої. Адреса 0 зарезервована для механізму автоматичної адресації (розглянуто нижче), тому для Ведених залишається 31 адреса. Системний біт використовується для ідентифікації типу кадру. Для передачі даних використовується 5 інформаційних бітів, у відповіді — 4.

Для обміну даними процесу використовується формат команди Data Exchange, який показаний на рис. 5.10.

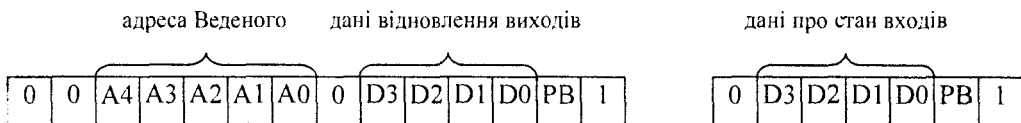


Рис. 5.10. Формат команди «Data Exchange»:
команда від Ведучого — зліва; відповідь Веденого — справа

Як бачимо, 4-ри біти даних призначені для відправки Ведучим вихідних даних, і чотири — для отримання вхідних від Веденого. 5-й інформаційний біт (тобто 4-й, починаючи з 0) при передачі не використовується і завжди дорівнює нулю. Таким чином, Ведучий може передати значення 4-х дискретних виходів, а Ведений відповісти, передаючи 4 значення дискретних входів, тобто одна шина AS-i може мати одночасно до 124 дискретних входів та стільки ж дискретних виходів. Можливе використання 4-х бітів даних для кодування аналогової інформації. Тобто з одним кадром можна передати 4-х розрядне число або 16 (2^4) різних значень, що дозволяє підключати такі засоби ЛМІ, як цифрова клавіатура. Деякі профілі передбачають підключення аналогових датчиків (виконавчих механізмів) з більшою розрядністю, обслуговуючи їх за декілька шинних циклів.

Для більшості контролерів, як Ведучих шини AS-i, читання входів і оновлення виходів Ведених пристроїв виконується автоматично (неявно) на початку і в кінці кожного циклу завдання. Це може проходити через змінні образу процесу, аналогічно як для локальних входів і виходів.

Приклад 5.1. AS-i. Мережа на базі Ведучого TSX Premium.

Завдання. Наведіть приклад адресації Ведучого AS-i шини на базі TSX Premium для входів і виходів Ведених пристроїв, відповідно до схеми мережі, наведеної на рис. 5.11.

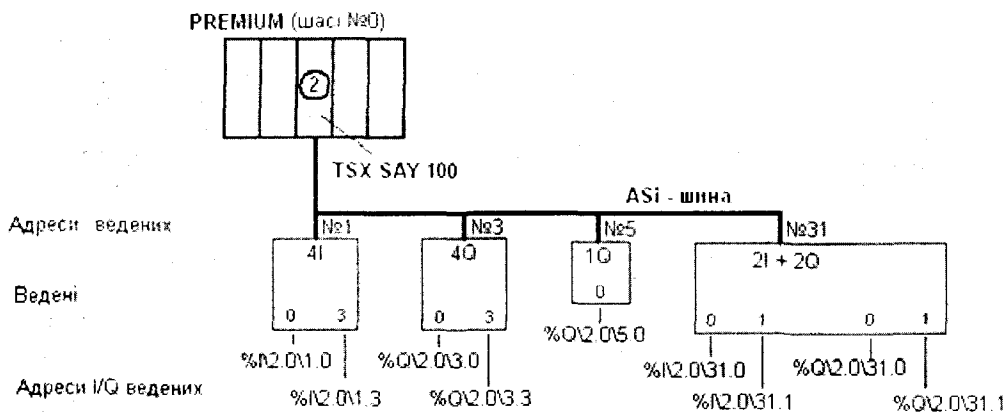


Рис. 5.11. Структура AS-i шини

Рішення. З рис. 5.11 видно, що у контролері TSX Premium, у слоті 2, що міститься на шасі 0, установлений модуль, який виступає у ролі Ведучого на AS-i шині (TSX SAY 100). До цієї шини підключені чотири Ведені, які сконфігуровані таким чином:

- Ведений №1 — для роботи з 4 дискретними входами;
- Ведений №3 — для роботи з 4 дискретними виходами;
- Ведений №5 — для роботи з 1 дискретним виходом;
- Ведений №31 — для роботи з 2 дискретними входами + 2 дискретними виходами.

Відповідно із синтаксисом мови програмування для цього типу контролерів (рис. 5.12), адреси окремих каналів Ведених показані на рис. 5.11.

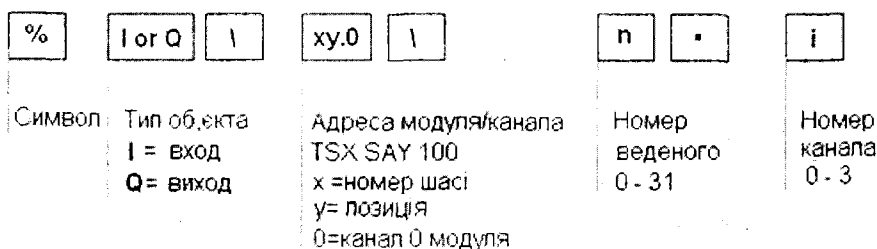


Рис. 5.12. Синтаксис звертання до входів-виходів AS-i шини

Наведена адресація означає, що:

- %I2.0\1.3 — вхід 3 веденого №1, каналу 0 модуля TSX SAY 100, який розміщений у слоті № 2, шасі № 0;
- %Q2.0\31.3 — вихід 0 веденого №31, каналу 0 модуля TSX SAY 100, який розміщений у слоті № 2, шасі № 0;
- і так далі.

5.6. Швидкість відновлення даних

Знаючи бітову швидкість, формат кадру та паузи між кадрами можна вирахувати час, за який Ведучий AS-i опитає всіх Ведених на шині. Транзакцією в даному випадку будемо називати час між опитуванням Ведучим двох Ведених, яких опитують один за одним. На рисунку 5.7 показаний час однієї транзакції. Він складається з:

- часу передачі кадру Ведучого: 14 бітових інтервалів (б.і);
- паузи 1: від 3-х до 10-ти б.і;
- часу передачі кадру Веденого: 7 б.і;
- паузи 2: від 1-го до 2-х б.і.

Перша пауза триває 3-бітові інтервали під час фази нормальної роботи, 5 — під час інших фаз. Максимальною ця пауза може тривати 10 бітових інтервалів при використанні репітерів. Таким чином, у режимі нормальної роботи на шині AS-i без використання репітерів, час однієї транзакції буде $14 + 3 + 7 + 2 = 26$ бітових інтервалів. Оскільки бітова швидкість = 167 кБіт/с, то цей час:

$$T_{\text{транз}} = 26/167\,000 \approx 156 \text{ мкс}$$

Для того, щоб опитати всіх Ведених, необхідно затратити $n \cdot 156$ мкс, де n — кількість Ведених. Але, окрім етапу обміну даними, стадія нормальної роботи складається також з етапів адміністрування та добавлення (див. рис. 5.13), які проводять ще дві транзакції. Втім, якщо кількість Ведених = 31, тобто максимум, етап добавлення (знаходження нового Веденого) не потрібен. Отже загальний час повного циклу шини AS-i буде:

$$T_{\text{циклу}} = 156 \text{ мкс} \cdot (n + 2) \text{ — якщо } n < 31$$

$$T_{\text{циклу}} = 156 \text{ мкс} \cdot (n + 1) \text{ — якщо } n = 31, \text{ тобто } 4,992 \text{ мс.}$$

Отже, за будь-якої кількості Ведених, час одного циклу, тобто час обміну Ведучого з усіма Веденими, не перевищує 5 мс.

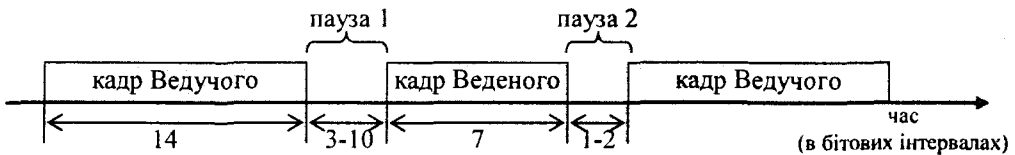


Рис. 5.13. Час однієї транзакції

5.7. Профілювання Ведучих та Ведених

Ведені пристрої, які підключаються до шини AS-i, повинні підтримувати певний профіль Веденого (Slave profile), який визначений виробником і «прошитий» у пам'яті пристрою. Саме за допомогою цього профілю вирізняється спосіб обробки даних, які відсилаються на пристрій і зчитуються з нього. Він робить прозорим і простим користування розподілених засобів Вх./Вих. різного типу та від різних виробників. Профілі дозволяють нарощувати номенклатуру технічних

засобів, не змінюючи при цьому протокол і не використовуючи протокол прикладного рівня.

В AS-і профіль має формат типу X.Y та визначається двома кодами:

- **I/O Code** (Input/Output Code — код Вх./Вих.);
- **ID Code** (Identification Code — ідентифікаційний код).

I/O Code характеризує тип веденого пристрою і показує скільки Вх./Вих. можуть бути підключені до нього. Цей код може приймати значення від 0 до 15₁₀ (0 – F₁₆). Ведучий зчитує його через цикл після того, як знайде даного Веденого.

ID Code призначений для відмінності пристроїв з одним і тим же I/O Code. Тобто це уточнююча інформація до типу пристрою. ID Code теж може приймати значення від 0 до 15₁₀ (0 – F₁₆). Ведучий зчитує його через цикл після того, як зчитає I/O Code даного Веденого.

Таким чином, можна сказати, що в профілі X.Y перше число (X) буде вказувати на I/O Code, тобто на кількість вхідних та вихідних бітів даних, а друге (Y) – на I/D Code, тобто на призначення цих бітів. Всі профілі стандартизовані і визначені в AS-i Association. Це значить, що кожний новий I/O Code повинен бути підтвердженим у даній організації і визначений як стандартний. Деякі профілі наведені в 5.10.

Залежно від можливостей та функцій, які підтримують Ведучі шини, вони належать до одного з трьох профілів Ведучого AS-i, наведених у табл. 5.1. З таблиці видно, що найбільш функціональним буде пристрій (модуль, блок) з профілем Ведучого AS-i M1, найменшим — M0.

Таблиця 5.1

ТИПИ ПРОФІЛІВ

	Тип профілю	Функціональність
Мінімальний профіль	M0	– запис та читання Вх./Вих.
Обмежений профіль	M2	– запис та читання Вх./Вих. – модифікація параметрів Веденого
Повний профіль	M1	– запис та читання Вх./Вих. – модифікація параметрів Веденого – тестування та діагностика системи – перевірка відповідності очікуваної конфігурації реальній

5.8. Автоматична адресація

Автоматична адресація дозволяє підключати Ведені до шини під час її нормальної роботи. Ця функція доступна тільки при роботі в захищеному режимі (Protected Mode). Крім того, Ведучий повинен мати профіль не менше M2. Для функціонування такої системи також необхідно, щоб були правильно сконфігуровані «очікувані» Ведені (тип, адреса та профіль).

Всі Ведені мають заводську настройку адреси, рівну 0. Коли такий пристрій підключається до шини, то протягом максимально 32-х циклів, на етапі добавлення, під цією адресою його знаходить Ведучий. Далі він порівнює LPS та LAS і визначає, з

якими адресами Ведених немає на шині. Порівнявши їх очікувані конфігураційні дані (профілі) він визначає претендента, якщо вони співпадають. Тобто, якщо профіль покищо не існуючого на шині але записаного в LPS Веденого з адресою наприклад 15, співпадає з профілем щойнознайденого, то вважається, що це і є той самий очікуваний Ведений. Протягом декількох циклів Ведучий відішле йому його нову адресу (в нашому випадку 15) і занесе його до списку активних Ведених (LAS).

Функція автоматичної адресації може бути використана для одного пристрою за раз. Тобто, поки не ідентифікується і не активується один, інші не можуть бути підключені. Адже декілька Ведених з однією адресою (0) одночасно на шині бути не можуть.

В іншому випадку (без автоматичної адресації) адресу вказують безпосередньо на Веденому.

5.9. AS-i V2.1 та V3

Поява нових специфікацій засвідчує розвиток даного стандарту. В специфікації AS-interface V2.1 (1998 р.) з'явилася можливість збільшити кількість Ведених пристроїв з 31 до 62. У відповідності до цієї специфікації адресний простір Ведучого, який підтримує дану можливість (профіль *M3*), ділиться на дві під області: А і В. Використання їх стосовно 31-ного Веденого дозволяє підключати до AS-i 62 Ведених пристроїв, які можуть мати до 4-х входів і до 3-х виходів. Тобто, всього до шини можна підключити до 248 дискретних входів та до 186 дискретних виходів. Можливість розширеної адресації з'явилася за рахунок використання найстаршого інформаційного біту в команді Ведучого, як допоміжного біта адреси. Саме цей біт визначає, до пристрою якого типу (А або В) за тією ж адресою звертається Ведучий.

Допоміжною особливістю даної специфікації є можливість організації обміну даними з аналоговими Веденими пристроями профілів 7.3 і 7.4 без використання допоміжних прикладних функцій.

У профілі Ведених для їх ідентифікації, на доповнення до ID Code, можуть використовуватись ID1 Code та ID2 Code. Це можливо за рахунок допоміжного (додаткового) циклу, а отже і фази добавлення та адміністрування. Таким чином, розширений профіль має 4 поля: I/O Code; ID Code; ID1 Code; ID2 Code.

Час циклу з максимальним набором Ведених пристроїв одного з типів (А або В) не перевищує 5 мс, як і в попередній специфікації. Якщо в AS-i версії 2.1 використовується весь адресний простір Ведучого пристрою, то Ведені пристрої А і В обслуговуються за чергою. У першому циклі обслуговуються ведені пристрої під області А, а в другому — під області В, після чого процес повторюється знову. Таким чином, сумарний час обслуговування всіх Ведених може досягати 10 мс.

Важливо, що Ведучі, які підтримують нову специфікацію, можуть обслуговувати Ведених попередніх специфікацій. Крім того, Ведені типу А можуть обслуговувати Ведучі, які не підтримують розширений режим.

У 3-й специфікації AS-i (2004 р.) кількість можливих Ведених, підключених до шини з Ведучим профілю *M4*, збільшилась до 62, а кількість підключених дискретних датчиків та виконавчих механізмів збільшилась до 496. Крім того, в новій версії з'явилася можливість зчитування значень з аналогових датчиків за

один цикл шини. Загальна таблиця доступних специфікаціям профілів пристроїв наведена в таблиці.

Таблиця 5.1

МОЖЛИВОСТІ ВЕРСІЙ AS-1

Версія	V2.0	V2.11	V3.0
Рік випуску версії	1994	1998	2004
Профілі Ведучих	M0, M1, M2	M3	M4
Профілі Ведених	всі інші профілі	S-X.A, S-7.3, S-7.4	S-7.5.5, S-7.A.5, S-B.A.5, S-7.A.7, S-7.A.A, S-7.A.8, S-7.A.9, S-6.0
Нові особливості	31 Ведений	62 Ведені, розширені ID Code, коди помилок	Синхронізація, нові комунікац. типи
Стандарти	EN 50295:1998 IEC 62026-2:2000		IEC 62026-2:200X

5.10. AS-i профілі

В табл. 5.2 наведений діапазон профілів Ведених пристроїв AS-i V2.0.

Таблиця 5.2

ДІАПАЗОН ПРОФІЛІВ ВЕДЕНИХ ПРИСТРОЇВ AS-i

I/O Code	D0	D1	D2	D3	ID Code																					
					0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F						
0	IN	IN	IN	IN	X	0.1																				
1	IN	IN	IN	OUT		1.1																				
2	IN	IN	IN	IN/OUT																						
3	IN	IN	OUT	OUT																						
4	IN	IN	IN/OUT	IN/OUT																						
5	IN	OUT	OUT	OUT																						
6	IN	IN/OUT	IN/OUT	IN/OUT																						
7	IN/OUT	IN/OUT	IN/OUT	IN/OUT		7.1																				
8	OUT	OUT	OUT	OUT																						
9	OUT	OUT	OUT	IN	R																					
A	OUT	OUT	OUT	IN/OUT	X.0																					
B	OUT	OUT	IN	IN	R	B.1																				
C	OUT	OUT	IN/OUT	IN/OUT	X.0																					
D	OUT	IN	IN	IN	R	D.1																				
E	OUT	IN/OUT	IN/OUT	IN/OUT	X.0																					
F	IN/OUT/ NULL	IN/OUT/ NULL	IN/OUT/ NULL	IN/OUT/ NULL	Не використовуються																					

Колонки D0-D3 вказують на використання бітів даних у кадрах типу Data Exchange на етапі обміну даними. Тобто IN вказує на вхідні біти, OUT — на ви-

хідні. Наприклад, для I/O Code 4 використовуються всі 4 біти в кадрі від Веденого до Ведучого, а для кадру Ведучого — тільки 2-ий та 3-ий. Це може бути, скажімо, профіль пристрою з 4-ма дискретними входами, та 2-ма дискретними виходами. Літера R вказує на зарезервованій профіль.

Профілі X.F (вільні профілі) використовуються у випадку пристроїв з дуже спеціалізованими функціями, які не знаходять широкого використання. Такі профілі також у пристроїв, які очікують стандартизації.

Профілі X.0 мають Ведені, що не використовують якихось специфічних, наперед визначених функцій. Тобто, просто віддалені пристрої Вх./Вих., I/O Code, яких визначає кількість дискретних входів та виходів.

Профіль 0.1 використовується для двох вхідних датчиків (2 канали на кожний датчик), 1.1 — для датчика з функцією тестування, В.1 — для управління двома реверсивними двигунами, D.1 — для управління двигуном з функцією тестування.

Профіль 7.1 (аналоговий профіль) застосовується для пристроїв з двонаправленим рухом інформації (4 біти — туди, 4 — в зворотному напрямку) та для аналогових пристроїв. Операцію з аналоговими даними Ведучий проводить за декілька AS-i циклів. Тому для реалізації обміну з даним профілем може знадобитися штучне, тобто програмне оформлення компонування даних, залежно від специфіки роботи Веденого. В специфікації AS-i V2.1 для роботи з аналоговими пристроями з'явилися профілі 7.3 та 7.4, які спрощують роботу з ними.



Контрольні запитання до розділу 5

1. Яке призначення мережі AS-i?
2. Які переваги має AS-i перед іншими промисловими мережами?
3. Які обмеження має AS-i ?
4. Які пристрої можна підключати до AS-i?
5. Яке основне обладнання використовується при побудові мережі AS-i ?
6. Які мережні кабелі можна використовувати при побудові AS-i?
7. Які переваги має спеціальний «жовтий» AS-i — кабель?
8. Яку топологію може мати мережа AS-i?
9. Яка особливість живлення обладнання на AS-i?
10. Яка максимальна довжина доступна для мережі AS-i? Як можна її збільшити? Яка бітова швидкість використовується?
11. Скільки входів/виходів може мати Ведений на AS-i V2.0?
12. Призначення буферів обміну Ведучого і Ведених?
13. Які процедури відбуваються під час конфігурування шини і окремих її складових?
14. В яких режимах може працювати Ведучий після активації Ведених?
15. Поясніть схему обміну даними процесу між Ведучим та Веденими.
16. Поясніть формат кадру команди від Ведучого.
17. Поясніть формат кадру відповіді Веденого.
18. Як можна вирахувати час, за який Ведучий AS-i опитає всіх Ведених на шині?
19. Поясніть поняття профілювання пристроїв в AS-i. Перерахуйте профілі Ведучих та підтримувані ними функції.
20. Які типи профілів використовуються і чим вони відрізняються?
21. Які особливості нових специфікацій стандарту AS-i V2.1 та V3.0?
22. Чи можна підключати до AS-i аналогові сигнали?

МЕРЕЖІ MODBUS

MODBUS розроблений в 1979 р. фірмою Modicon Gould (зараз у складі Schneider Electric). Це один із найстаріших протоколів промислової мережі, який сьогодні є одним з найбільш популярних. Основна причина такої популярності — це простота в реалізації. Даний розділ присвячений мережам, які базуються на протоколі прикладного рівня MODBUS.

Нині MODBUS підтримує і розвиває організація **MODBUS-IDA**, яка являє собою групу незалежних споживачів та постачальників пристроїв автоматизації. Вона забезпечує відкритість даного протоколу та розробляє готові компоненти для спрощення реалізації. Будемо розглядати MODBUS таким, яким він тепер у стандартах MODBUS-IDA. Одна з реалізацій протоколу — MODBUS TCP/IP — увійшла в стандарти МЕК IEC 61158-5-15, IEC 61158-6-15 та IEC 61784-2 як 15-й тип. Враховуючи особливості реалізації мереж на базі MODBUS та опису його в МЕК тільки на прикладному рівні, доцільно розглядати його в контексті моделі OSI а не МЕК.

6.1. Мережі MODBUS у контексті моделі OSI

Згідно зі стандартами MODBUS-IDA — MODBUS є протоколом прикладного рівня для зв'язку типу Клієнт-Сервер між прикладними Процесами пристроїв, які під'єднані до різноманітних типів шин або мереж. У контексті OSI-моделі ці мережі мають архітектуру, наведену на рис. 6.1.

Як видно з рисунка, MODBUS нині представлений 4-ма мережами: MODBUS RTU, MODBUS ASCII, MODBUS Plus і MODBUS TCP/IP. Перші реалізації MODBUS базувалися на послідовних інтерфейсах з двома режимами передачі RTU і ASCII, але з розвитком комп'ютерних мереж і їх інтеграції з промисловими мережами протокол MODBUS адаптували до використання в мережах, що базуються на основі TCP/IP. MODBUS Plus, в основному, використовується в пристроях Schneider Electric, тому розглядати його не будемо.

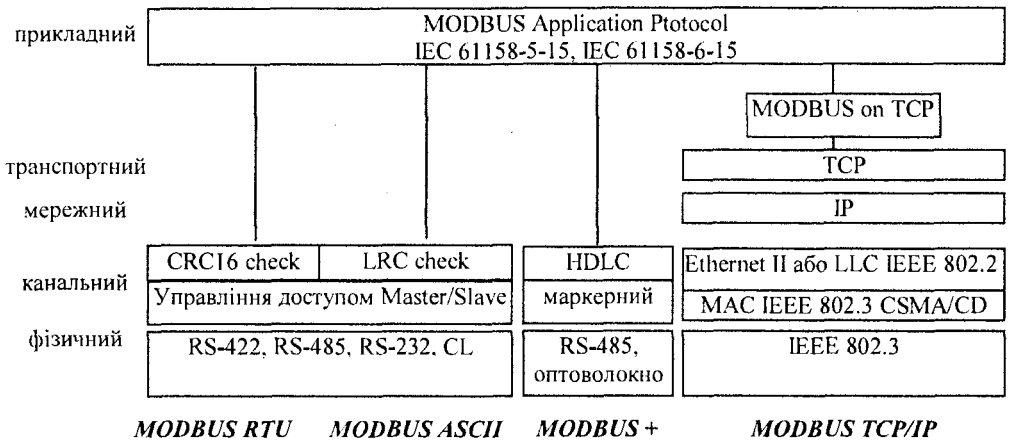


Рис. 6.1. MODBUS у контексті OSI-моделі

6.2. Реалізація MODBUS на прикладному рівні

6.2.1. Формат MODBUS PDU

MODBUS Application Protocol (MBAP) MODBUS протокол прикладного рівня) базується на моделі Клієнт-Серверного обміну повідомленнями і визначає формат повідомлень **MODBUS PDU** (Protocol Data Unit), які мають вигляд, наведений на рис. 6.2.

Клієнтський прикладний Процес робить повідомлення-запит до серверного Процесу, в якому в полі «код функції» вказує йому на дію, яку необхідно провести. Байти даних вміщують інформацію, яка необхідна для виконання даної функції. Серверний прикладний Процес у випадку вдалого виконання цієї функції повторює код функції у відповіді (якщо запит передбачає відповідь). При виникненні помилки код функції у відповіді модифікується (старший біт виставляється в 1), а в байтах даних передається причина помилки. Тобто, якщо при передачі клієнтським прикладним Процесом повідомлення-запиту з функцією 03_{16} (00000011_2) виникла помилка у її виконанні Сервером, той відішле відповідь з полем функції рівним 83_{16} (10000011_2). В доповненні до зміни коду функції, при помилці, Сервер розміщує в поле даних унікальний код, який вказує на тип і причину помилки.

Код функції являє собою поле з одного байта, яке може приймати значення від 1 до 255 (коди 128-255 зарезервовані під коди повідомлень-відповідей при помилкових діях). Всі коди функцій в MBAP діляться на (див. рис. 6.3):

– Public Function Codes — це публічні коди, які описані в стандарті MODBUS-IDA, список їх включає в себе вже призначені та використовувані коди, а також коди для майбутнього використання;

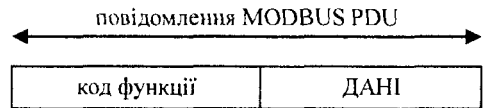


Рис. 6.2. Формат повідомлення MODBUS PDU

- User-Defined Function Codes (65-72, 100-110) — це коди, які можуть використовуватись компаніями для власних функцій і не описані в специфікації;
- Reserved Function Codes (9, 10, 13, 14, 41, 42, 43, 90, 91, 125, 126 і 127) — це зарезервовані коди, які не доступні для загального використання.

Нижче розписані тільки ті функції, які призначені для доступу до даних процесу. Ці дані з точки зору MODBUS функцій діляться на:

- **Discrete Inputs**: дискретні входи, тільки для читання;
- **Coils**: котушки, внутрішні біти або дискретні виходи, читання/запис;
- **Input Registers**: вхідні 16-бітні змінні, тільки читання;
- **Holding Registers**: внутрішні/вихідні 16-бітні змінні, читання/запис.

				Function Codes		
				code	Sub code	(hex)
Data Access	Bit access	Physical Discrete Inputs	Read Discrete Inputs	02		02
		Internal Bits Or Physical coils	Read Coils	01		01
			Write Single Coil	05		05
	Write Multiple Coils ¹		15		0F	
	16 bits access	Physical Input Registers	Read Input Register	04		04
			Read Holding Registers	03		03
		Internal Registers Or Physical Output Registers	Write Single Register	06		06
			Write Multiple Registers	16		10
			Read/Write Multiple Registers	23		17
			Mask Write Register	22		16
			Read FIFO queue	24		18
			Read File record	20	6	14
	File record access	Write File record	21	6	15	
		Read Exception status	07		07	
	Diagnostics	Diagnostic		08	00-18,20	08
		Get Com event counter		11		0B
		Get Com Event Log		12		0C
		Report Slave ID		17		11
Read device Identification		43	14	2B		
Other	Encapsulated Interface Transport		43	13,14	2B	

Рис. 6.3. Функції MODBUS

У повідомленні-запиті за полем коду функції можуть слідувати дані, які уточнюють або доповнюють функцію допоміжними даними. Це можуть бути адреси змінних, їх кількість, лічильник байтів даних та самі дані для запису. Для певних функцій поле даних може бути відсутнім взагалі. Максимальна довжина повідомлення прикладного рівня рівна 253.

6.2.2. Формат основних функцій

Повний список кодів, а також специфікацію протоколу можна знайти на офіційному Веб-сайті MODBUS-IDA — www.MODBUS.org. У посібнику детально

розглянемо тільки найбільш вживані функції MODBUS для обміну даними процесу. Номер функції дається в шістнадцятковому форматі. Скорочення в дужках *Hi* та *Lo* вказують відповідно на старший та молодший байти. Тобто, якщо для вказівки адреси початкової змінної необхідно двобайтове слово, то значення старшого байта буде передаватись у полі з позначенням *Hi*, а молодшого — відповідно *Lo*.

6.2.2.1. Код функції 01₁₆ — читання статусу Coils (дискретних вихідних бітів). Повідомлення-запит вміщує адресу початкового біта і кількість бітів для читання. Біти нумеруються, починаючи з 0. У повідомленні-відповіді кожне значення змінної передається одним бітом, тобто в одному байті пакується статус 8 бітових змінних. Якщо кількість їх некротна восьми, інші біти в байті заповнюються нулями. Лічильник вміщує кількість байтів у полі даних.

Запит:

Код функції	01
Адреса початкового біта (Hi)	0 до FFFF ₁₆
Адреса початкового біта (Lo)	
Кількість бітів (Hi)	1 до 7D0 ₁₆ (2000)
Кількість бітів (Lo)	

Відповідь:

код функції	01
лічильник байтів	N
Значення біта (перші 8 бітів)	0 до FF ₁₆

Значення

ітів (наступні 8 бітів)	0 до FF ₁₆
...	
Значення бітів (N-ні 8 біт)	0 до FF ₁₆

6.2.2.2. Код функції 02₁₆ — читання статусу дискретних входів. Формат даного запиту такий же, як попереднього, за винятком поля функції.

6.2.2.3. Код функції 03₁₆ — читання значення вихідних/внутрішніх регістрів. Повідомлення-запит вміщує адресу початкового вихідного/внутрішнього регістру (двохбайтове слово) і кількість регістрів для читання. Регістри нумеруються, починаючи з 0.

Запит:

код функції	03
Адреса початкового регістру (Hi)	від 0 до FFFF ₁₆
Адреса початкового регістру (Lo)	
Кількість регістрів (Hi)	від 1 до 7D ₁₆ (125)
Кількість регістрів (Lo)	

Відповідь:

код функції	01
лічильник байт	N*2
Значення 1-го регістру (Hi)	0 до FFFF ₁₆
Значення 1-го регістру (Lo)	
...	
Значення N-го регістру (Hi)	0 до FFFF ₁₆
Значення N-го регістру (Lo)	

У відповідному повідомленні в полі даних кожний регістр передається двома байтами.

6.2.2.4. Код функції 04₁₆ — читання значення вхідних регістрів. Формат даного запиту такий же, як попереднього, за винятком поля функції.

6.2.2.5. Код функції 05₁₆ — запис вихідного/внутрішнього біта. В запиті вказується номер бітової змінної та значення: 0 — 0000, а 1 — FF00, всі інші значення не міняють стан змінних. У ширококомвній передачі клієнтський запит виставляє значення даної змінної для всіх серверів.

Функція	05
Адреса біта (Hi)	від 0 до FFFF ₁₆
Адреса біта (Lo)	
Значення біта (Hi)	0000 або FF00 ₁₆
Значення біта (Lo)	

Нормальна відповідь сервера є повторенням запиту до клієнта.

6.2.2.6. Код функції 06₁₆ — запис вихідного/внутрішнього реєстру. Функція аналогічна попередній, але оперує з реєстрами (словами). У запиті вказується номер вихідного/внутрішнього реєстру та його значення. В широкомовній передачі запит виставляє значення даної змінної для всіх серверів.

Функція	06
Адреса реєстру (Hi)	від 0 до FFFF ₁₆
Адреса реєстру (Lo)	
Значення реєстру (Hi)	від 0 до FFFF ₁₆
Значення реєстру (Lo)	

Нормальна відповідь сервера є повторенням запиту клієнту.

6.2.2.7. Код функції 0F₁₆ — запис декількох вихідних/внутрішніх бітів. У запиті вказується початкова адреса біту, кількість бітів для запису, лічильник байтів і безпосередньо значення. В широкомовній передачі біти записуються всім серверам. Розглянемо приклад для встановлення таких бітових вихідних/внутрішніх змінних:

Байт 1								Байт 2							
26	25	24	23	22	21	20	19	—	—	—	—	—	—	28	27
1	1	0	0	1	1	0	1							0	1

У таблиці показана відповідність адреси змінної, починаючи з 19-ї, і значення біта. Для зручності біти розміщені у тому порядку, що і передаються. В другому байті корисні тільки 2 перші біти, значення інших не буде прийнято до уваги, оскільки кількість бітів указані у кадрі. Запит та відповідь матимуть такий вигляд:

Запит:

Функція	0F
Адреса початкового біта (Hi)	00
Адреса початкового біта (Lo)	13
Кількість бітів (Hi)	00
Кількість бітів (Lo)	0A
Лічильник байтів	02
Дані (змінні 19–26)	CD
Дані (змінні 27–28)	01

Відповідь:

Функція	0F
Адреса початкового біта (Hi)	00
Адреса початкового біта Lo	3
Кількість бітів (Hi)	00
Кількість бітів (Lo)	0A

6.2.2.8. Код функції 10₁₆ — запис декількох вихідних/внутрішніх регістрів

Запит:

Функція	10 ₁₆
Адреса початкового регістру (Hi)	0 до FFFF ₁₆
Адреса початкового регістру (Lo)	
Кількість регістрів (Hi)	1 до 007B ₁₆ (123)
Кількість регістрів (Lo)	
Лічильник байтів	2*N
Дані (1-й регістр Hi)	0 до FFFF ₁₆
Дані (1-й регістр Lo)	
...	
Дані (N-й регістр Hi)	0 до FFFF ₁₆
Дані (N-й регістр Lo)	

Відповідь:

Функція	10
Адреса початкового регістру (Hi)	0 до FFFF ₁₆
Адреса початкового регістру Lo	
Кількість регістрів (Hi)	1 до 007B ₁₆ (123)
Кількість регістрів (Lo)	

6.2.2.9. Повідомлення про помилки. Ці повідомлення стосуються всіх типів MODBUS, але першопочатково були визначені для MODBUS Serial (RTU/ASCII).

При запиті Клієнта до Серверу можуть мати місце такі ситуації:

- якщо Сервер прийняв запит без комунікаційних помилок і може нормально розпізнати запит, він повертає нормальну відповідь;
- якщо Сервер не прийняв запит, відповідь не повертається. Клієнт очікує відповіді протягом певного тайм-ауту;
- якщо Ведений (для MODBUS Serial) прийняв кадр, але знайшов комунікаційну помилку (паритет, помилка контрольної суми), то кадр-відповідь не повертається, а Ведучий чекає відповіді на запит протягом певного тайм-ауту;
- якщо Сервер прийняв запит без комунікаційної помилки, але не може виконати замовлену функцію (наприклад, читання неіснуючих виходів або регістрів), Сервер повертає повідомлення про помилку і її причини.

Таблиця 6.1

СПИСОК КОДІВ

Код	Назва	Опис
01	ILLEGAL FUNCTION	Прийнятий код функції не може бути оброблений на Сервері
02	ILLEGAL DATA ADDRESS	Адреса даних, указана в запиті, недоступна даному Серверу
03	ILLEGAL DATA VALUE	Величина, вміщена в полі даних запиту є недопустимою величиною для Серверу
04	SLAVE DEVICE FAILURE	Невиправна помилка мала місце, поки Сервер намагався виконати дію запиту
05	ACKNOWLEDGE	Сервер прийняв запит і обробляє його, але необхідний певний час. Ця відповідь захищає Клієнта від генерації помилки тайм-ауту
06	SLAVE DEVICE BUSY	Сервер зайнятий обробкою команди, Клієнт повинен повторити запит пізніше
07	NEGATIVE ACKNOWLEDGE	Невдалий програмний запит (для функцій 13 і 14)
08	MEMORY PARITY ERROR	Сервер хоче читати розширену пам'ять, але знайшов помилку паритету

Повідомлення про помилку має два поля, які відрізняються від полів нормальної відповіді:

ПОЛЕ КОДУ ФУНКЦІЇ: при нормальній відповіді Сервер повертає в цьому полі той номер функції, який потребував Клієнт. У всіх кодах функції старший біт установлений в 0. При поверненні повідомлень про помилку Сервер встановлює цей біт в 1, після чого Клієнт може ідентифікувати наявність помилки.

ПОЛЕ ДАНИХ: у цьому полі при помилці повертається її код.

Приклад 6.1. MODBUS. Запит на читання статусу вихідних бітів

Завдання. Сформуванати повідомлення-запит та повідомлення-відповідь на читання вихідних/внутрішніх бітів з 20 по 39 при: обробці без помилок; обробці з помилкою ILLEGAL DATA ADDRESS.

Рішення. Формат повідомлень показаний на рис. 6.4. Слід зауважити, що 1-й біт в адресному просторі MODBUS опитується під номером 0. Тому 20-й біт зчитується під номером 19. Деталі читайте в 6.2.3.

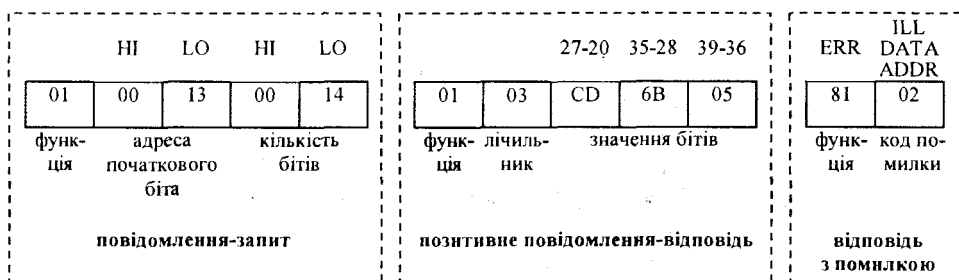


Рис. 6.4. Формат повідомлень для запити читання статусу вихідних бітів

Приклад 6.1

Приклад 6.2. MODBUS. Запит на читання значення вихідних/внутрішніх регістрів.

Завдання. Сформуванати повідомлення-запит та повідомлення-відповідь на читання вихідних/внутрішніх регістрів, починаючи з 108-го по 110-й, при позитивній обробці запити Сервером.

Рішення. Формат повідомлень показаний на рис. 6.5. Як і в попередньому випадку, 108-й регістр у запиті вказується під номером 107 (6В₁₆).

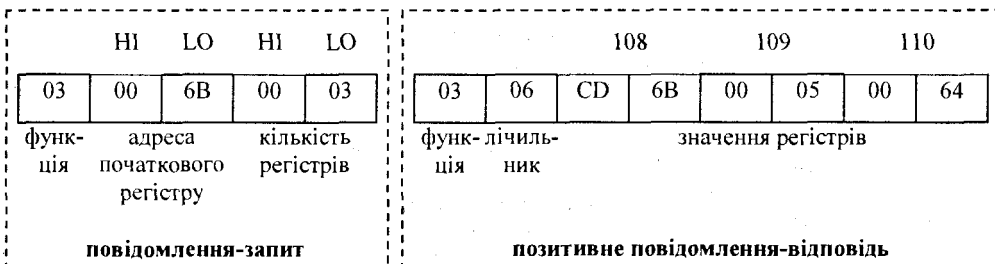


Рис. 6.5. Формат повідомлень для запити читання значень вихідних/внутрішніх регістрів

Приклад 6.1

6.2.3. Адресна модель MODBUS та доступ до даних

Усі змінні, до яких звертається клієнтський прикладний Процес, є частиною області пам'яті пристрою. Однак, фізичні адреси цих змінних можуть не співпадати з тими, до яких він звертається згідно з протоколом. Головне, щоб ці змінні були відображенням (mapping) дійсних даних у пристрої. На рис. 6.6 показаний процес обробки запитів на боці серверу MODBUS. Як видно з рисунка, при зверненні до певної змінної за її номером, наприклад, на читання, по суті, йде звернення до певної області даних у прикладній програмі пристрою. Зв'язок даних MODBUS моделі з фізичними даними називають *відображенням (mapping)*.

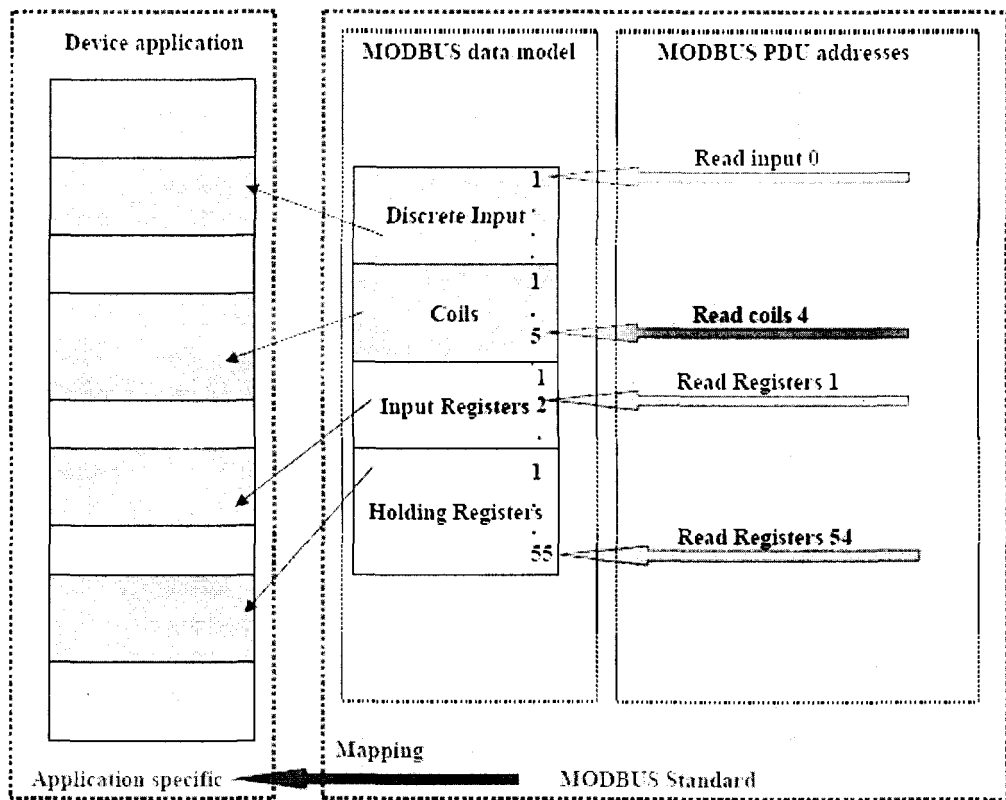


Рис. 6.6. Адресна модель MODBUS

Слід відмітити, що мінімальна адреса елемента даних моделі MODBUS дорівнює 1, а не 0.

Таке розмежування між фізичною структурою даних і MODBUS моделі даних дає можливість адаптувати протокол під структуру різних пристроїв. Для прикладу, в стандарті наводяться два популярні способи реалізації MODBUS моделі даних: розділення даних за блоками (рис. 6.7) та використання доступу до

даних одного і того ж блока (рис. 6.8). Як видно з рисунків, модель даних з одним блоком дає доступ до одних і тих же фізичних даних. Тобто, області Input та Holding реєстрів співпадають, а області Digital Inputs та Coils теж співпадають і містяться в області реєстрів.

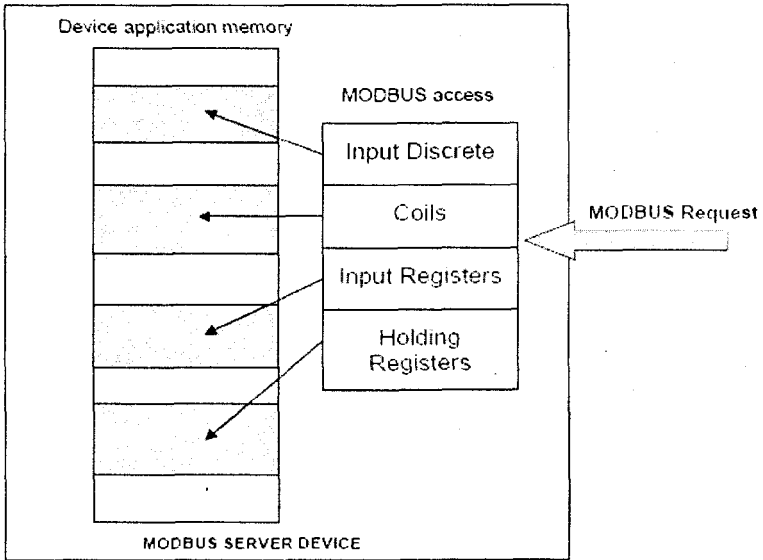


Рис. 6.7. Модель даних MODBUS з розділеними блоками

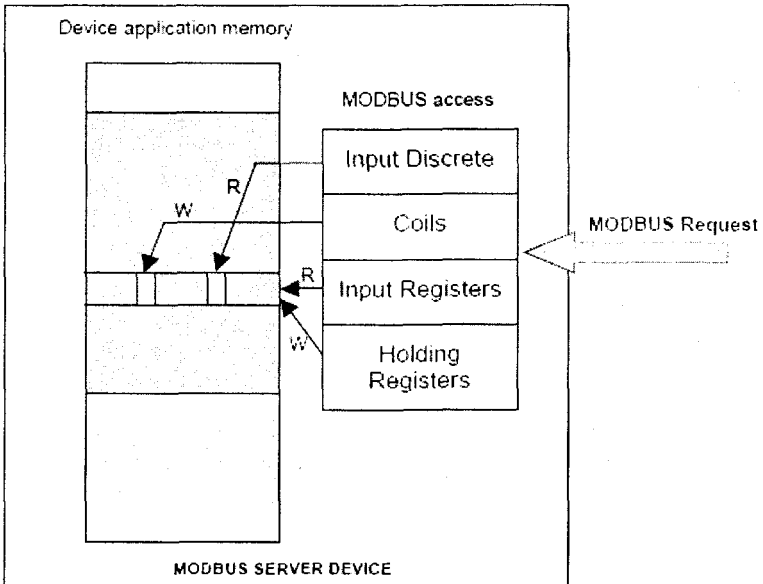


Рис. 6.8. Модель даних MODBUS з одним блоком

Приклад 6.3. MODBUS. Модель даних для різних типів пристроїв.

Завдання. Показати відображення даних моделі MODBUS на адресний простір Momentum/Quantum (Schneider Electric), Micro/Premium/M340/Twido (Schneider Electric), Vipa CPU-21xSER1 (VIPA) Vipa IM-253NET(VIPA).

Рішення. Для контролерів Schneider Electric гілки Modicon тобто Momentum та Quantum, MODBUS — це рідний протокол, тому модель даних MODBUS абсолютно співпадає з моделлю їх адресного простору. Нумерація змінних починається з 1-ї (0-вий номер у запиті звертається до 1-ї змінної)

Для контролерів Micro/Premium, які влились у Schneider Electric під брендом Telemecanique, а також M340 та Twido доступ надається тільки до внутрішніх змінних (M — Memory), тому різниці між вхідними та вихідними бітами та регістрами немає, однак бітова область не співпадає з регістровою. Тобто, Сервер MODBUS цих ПЛК буде однаково реагувати, скажімо, на функції 01 та 02 або 03 та 04. Нумерація змінних починається з 0-ї (0-вий номер у запиті звертається до 0-ї змінної, тобто %M0-біт або %MW0 — регістр). Таким чином загальна таблиця відображення даних для різних ПЛК від Schneider Electric має такий вигляд:

	Momentum/Quantum	Micro/Premium/M340/ Twido
Input Discrete	1XXX	%M
Coils	0XXX	%M
Input Registers	3XXX	%MW
Holding Registers	4XXX	%MW

Для Siemens-подібних контролерів VIPA мережі MODBUS не найбільш жививані. Втім, VIPA випускає номенклатуру модулів з підтримкою даного протоколу. Зокрема, в процесорних модулях типу Vipa CPU-21xSER1 є вбудований послідовний порт з підтримкою MODBUS Serial (RTU/ASCII) як у режимі Ведучого, так і Веденого. Модуль представляє бік Серверу MODBUS у режимі Веденого (пояснення дивись в MODBUS Serial), тому саме в цьому режимі розглядається відображення даних моделі MODBUS. При конфігурації послідовного порту CPU-21xSER1, задається його режим (MODBUS Master). Далі весь обмін проходить через вхідні та вихідні буфери (рис. 6.9), які прикладна програма повинна поновлювати самостійно. Однак, буферів усього два, а отже бітові змінні містяться в пам'яті регістрових, вхідні — в області вхідних регістрів, вихідні — вихідних (див. рис. 6.10).

Слід зазначити, що буфери є внутрішньою пам'яттю комунікаційного порту, а не самого ПЛК. Для відображення буферів на області пам'яті ПЛК, у комунікаційних функціях SEND та RECEIVE вказується необхідна область пам'яті, як правило, це DB-область. Необхідно також пам'ятати, що адресація даних у буферах з боку мережі проходить за словами, а з боку програми до DB ПЛК — за байтами.

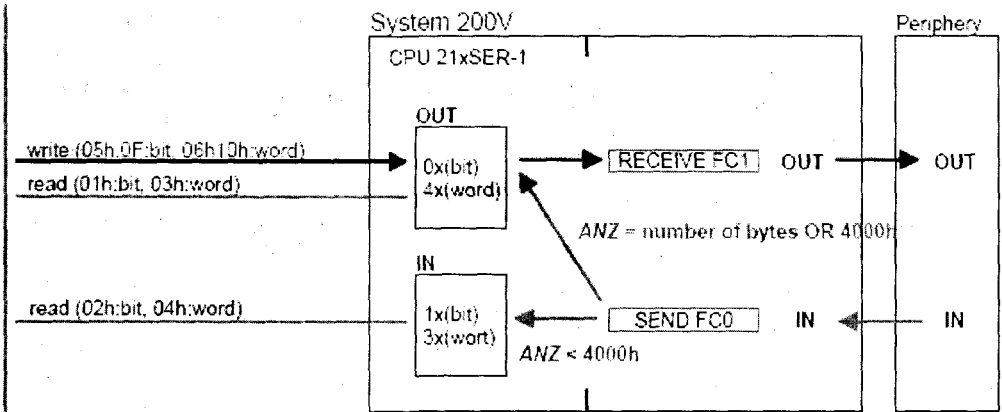


Рис. 6.9. Обмін даними MODBUS у ПЛК VIPA CPU-21xSER1

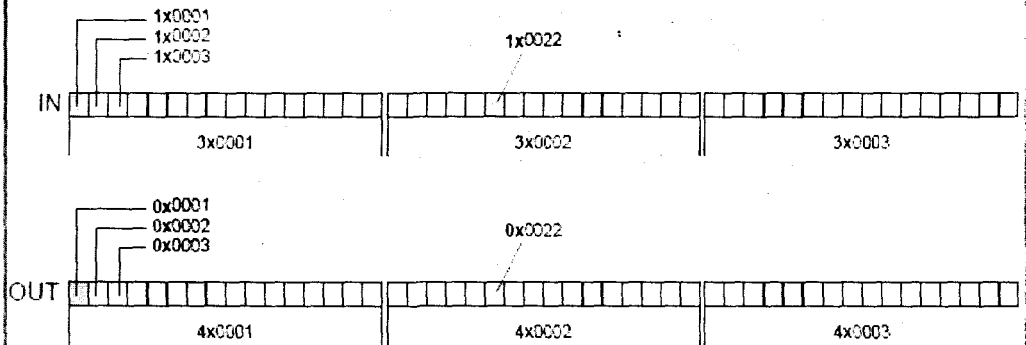


Рис. 6.10. Відображення змінних у вхідному та вихідному буферах для ПЛК VIPA

Інтерфейсний модуль Vipa IM-253NET(VIPA) для побудови розподілених систем вводу/виводу на базі Ethernet підтримує MODBUS TCP/IP, тобто може бути Сервером MODBUS (Клієнтом бути не може). Доступ до входів та виходів системи на базі інтерфейсного модуля відбувається аналогічно, як показано на рис. 6.10. Буфер IN поновлюється модулем автоматично, а виходи ПЛК автоматично відновлюються даними з буфера OUT.

Приклад 6.3

6.3. MODBUS Serial

Перші мережі MODBUS базувалися на асинхронних послідовних лініях зв'язку і отримали назву **MODBUS RTU** та **MODBUS ASCII**. На фізичному рівні вони використовують стандартні послідовні інтерфейси із символьним режимом передачі (див. рис. 6.1).

Сьогодні в MODBUS-IDA ці мережі отримали назву MODBUS over Serial Line і описані у відповідному стандарті. У ньому вказуються правила та рекомендації використання на каналному та фізичному рівнях.

Оскільки мережі MODBUS RTU/ASCII можуть мати шинну топологію, то визначений метод доступу до шини — це модель Ведучий/Ведений. У мережах MODBUS RTU та MODBUS ASCII Процес Ведучого завжди є Клієнтом, а Процеси Ведених — Серверами. Це значить, що Ведучий відсилає запити, а Ведені їх обробляють. Цей запит може бути адресований як індивідуальному вузлу, так і всім Веденим на шині (broadcast).

На каналному рівні MODBUS RTU/ASCII використовується адресація орієнтована на ідентифікатори вузлів. Кожний Ведений повинен мати свою унікальну адресу (1-247), Ведучий не адресується. При індивідуальних запитах Ведучий з клієнтським Процесом формує кадр із повідомленням-запитом і відправляє його за вказаною адресою. Ведений із серверним Процесом отримує цей кадр і обробляє повідомлення. Після його обробки Ведений формує кадр з повідомленням-відповіддю і відправляє його назад Ведучому. Кадр з повідомленням-відповіддю носить також функції кадру підтвердження, який Ведучий буде чекати від Веденого протягом часу, визначеного тайм-аутом.

При широкомовних запитах (broadcast) використовується 0-ва адреса. Широкомовні запити не потребують підтвердження, тому після відправки широкомовного кадру Ведучий не очікує кадр відповіді.

6.3.1. Канальний рівень

На рис. 6.11 показаний загальний вигляд кадру MODBUS Serial. Зверніть увагу, що розмежування між кадрами та тип контрольної суми тут не вказані, оскільки це залежить від режиму передачі ASCII або RTU. В полі адреси пристрою Ведучий при запиті вказує адресу отримувача, а Ведений при відповіді — свою адресу. Поля MODBUS PDU описані вище.

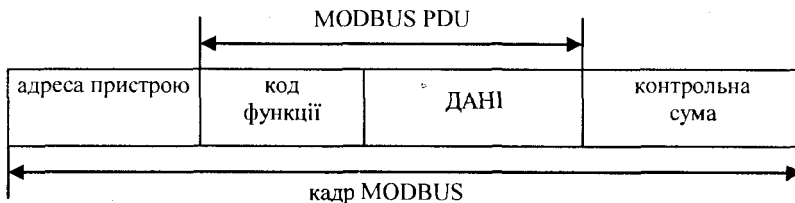


Рис. 6.11. Формування кадру MODBUS Serial

На часовій діаграмі рис. 6.12 показані три типові ситуації роботи моделі Ведучий/Ведений на MODBUS Serial. Перша ситуація — типовий обмін в одноадресному режимі, друга — в широкомовному, третя — реакція Веденого на комунікаційну помилку.

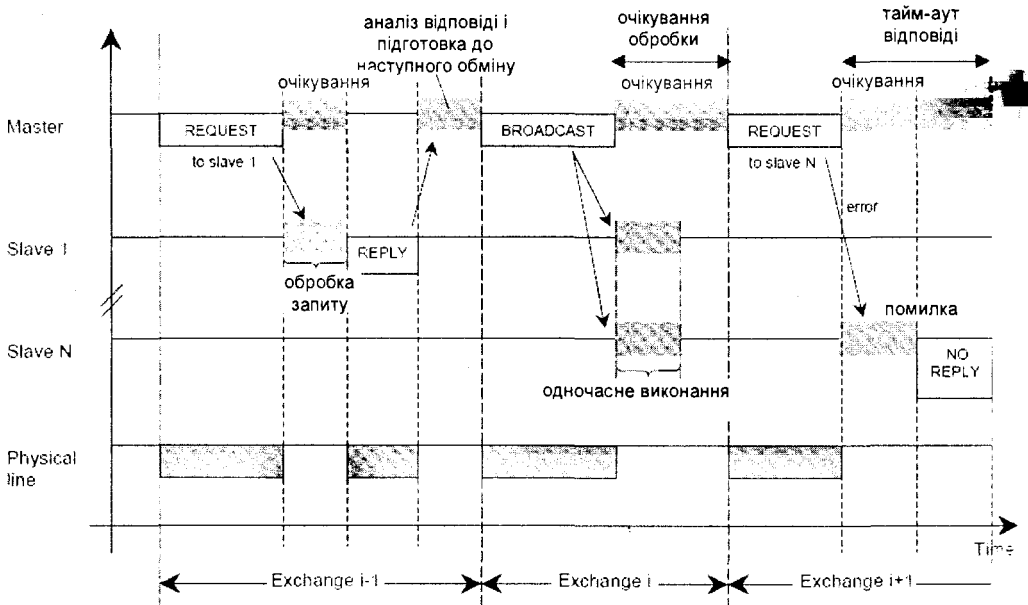


Рис. 6.12. Часова діаграма обміну на MODBUS Serial

6.3.2. MODBUS RTU

Даний режим передбачає використання 8 бітів даних у 11-бітному символі, що дозволяє передавати один байт на символ. Формат символу в RTU-режимі: 1 стартовий біт; 8 бітів даних (молодший біт передається першим); 1 біт паритету + 1 стоповий біт або без паритету + 2 стопових біти.

Формат кадру MODBUS RTU наведений на рисунку 6.13. Розмежування між кадрами проводиться за допомогою пауз між символами. Новий кадр не повинен з'являтися на шині раніше, ніж $3.5 \cdot T_c$ від попереднього, де T_c — час передачі одного символу. Якщо час відсутності сигналу на лінії (інтервал тиші) буде більше ніж $1.5 \cdot T_c$ отримувач ідентифікує закінчення кадру. З іншого боку, появлення нового кадру раніше ніж $3.5 \cdot T_c$ приведе до помилки.

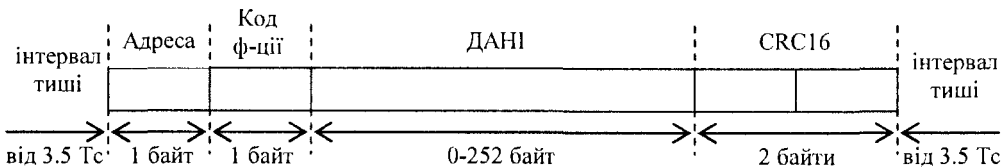


Рис. 6.13. Формат кадру для RTU режиму обміну

Поле адреси і коду функції в RTU режимі займають по одному байту, оскільки байти передаються по символу на кожний. В якості контрольної суми використовується два байти, оброблені за алгоритмом CRC16.

6.3.3. MODBUS ASCII

У даному режимі кожний байт повідомлення передається як два ASCII символи їх шістнадцяткового представлення, тобто значення байта 03_{16} буде передаватися як ASCII-код символів «0» і «3» (0110000 0110011). Отже байти даних, код функції і байт поля перевірки буде передаватися кодами символів 0-9, A-F. Формат символу в ASCII-режимі: 1 стартовий біт; 7 бітів даних, молодший біт передається першим; 1 біт паритету + 1 стоповий біт або без паритету + 2 стопових біта.

Формат кадру наведений на рис. 6.14. Як бачимо, для розмежування між кадрами використовуються стартовий символ «:» та стопова послідовність «CR LF». Отримувачі на шині безперервно відслідковують символ «:» який однозначно вказує на початок кадру. Коли він прийнятий, отримувачі відловлюють поле адреси і т.д. Це дуже простий спосіб синхронізації, який дозволяє не критично відноситись до пауз між символами (до 1 сек.). Адреса Веденого та код функції займають по два символи, відповідно до значення одного байта. Далі йдуть $n \cdot 2$ символів даних, де n кількість байтів даних. В ASCII режимі MODBUS для підрахунку контрольної суми використовується алгоритм LRC. Причому контрольна сума проводиться над усіма байтами кадру, виключаючи стартову та стопову послідовність символів.

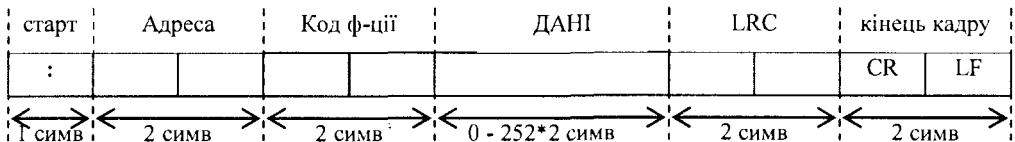


Рис. 6.14. Формат кадру для ASCII режиму обміну

Режим ASCII накладає менші вимоги на обладнання за рахунок використання стартової і стопової послідовності в розмежуванні кадрів і нечутливості до значних пауз між символами. Але переваги є і його недоліками. RTU-режим більш вимогливий до інтервалів між кадрами, але значно продуктивніший, ніж ASCII.

Приклад 6.4. MODBUS. Розрахунок часу опитування Ведених на MODBUS-RTU

Завдання. Побудувати кадри форматів повідомлень запитів та відповідей для MODBUS RTU та розрахувати загальний час опитування 10-ти аналогових 16-бітних змінних для 4-х Ведених (рис. 6.15). Бітова швидкість — 19200 біт/с. Клієнтський Процес Ведучого (TSX Premium) та серверні Процеси Ведених (ПЛК TSX Micro) приймають повідомлення на початку циклу, а відправляють — у кінці циклу. Час циклу Ведучого = 10 мс, Ведених — 5 с.

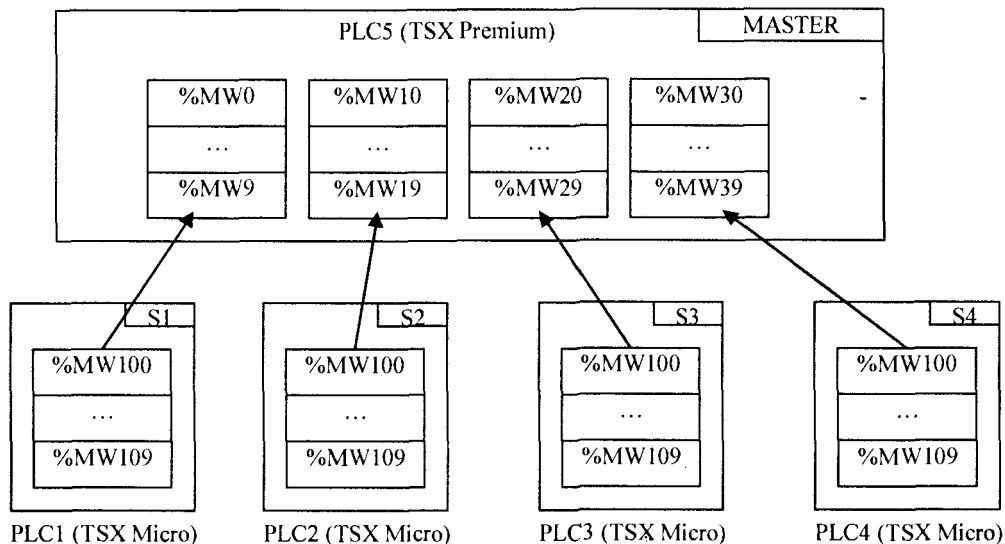


Рис. 6.15. Постановка завдання до прикладу 6.4

Виконання завдання. Доступ до внутрішніх аналогових змінних TSX Micro проводиться через 03 або 04 функцію, тому формат кадрів буде мати вигляд, як на рис. 6.16.

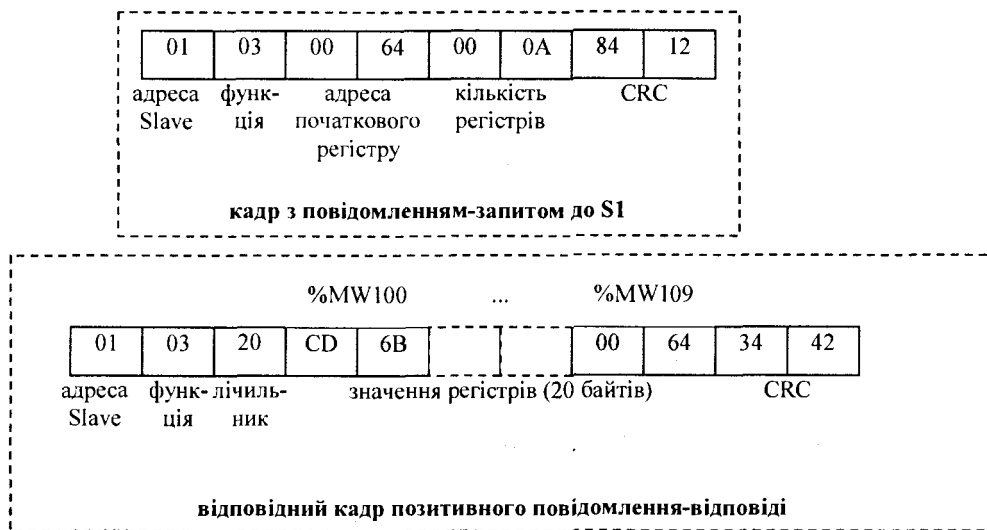


Рис. 6.16. Формат кадрів до S1 та Ведучого

Враховуючи, що структура інших кадрів — аналогічна, наводити їх формат немає сенсу.

Аналогічно рис. 6.12, побудуємо часову діаграму обміну (рис. 6.17). З боку клієнтської програми повідомлення-запит формується за допомогою комунікаційної функції, відправка даних якої через комунікаційний порт проводиться в кінці циклу задачі, а отримування з порту — на початку циклу. Така поведінка клієнтської сторони цілком відповідає багатьом реалізаціям для різних ПЛК.

В TSX Micro MODBUS-Сервер реалізований на рівні операційної системи. Специфіка реалізації полягає в тому, що прийом MODBUS-запитів з комунікаційного порту системою проводиться на початку циклу, а відправка повідомлень-відповідей — в кінці.

Слід зазначити, що реалізація MODBUS-Серверу може бути підтримана на рівні комунікаційного модуля, а обмін даними з пам'яттю самого пристрою проводиться через комунікаційні буфери. В цьому випадку реакція MODBUS-Серверу буде значно швидшою і не залежатиме від циклу програми. Для розрахунку часу транзакції для інших типів систем необхідно ознайомитися з деталями їх реалізації.

На рис. 6.17 показано, що надходження кадру проходить десь в середині циклу. Це значить, що їх обробка та генерація відповіді відбудеться приблизно через 1,5 циклу. Слід розуміти, що це усереднене значення, для найгіршої оцінки краще резервувати 2 цикли (тобто коли кадр прийшов відразу після опитування комунікаційного порту). Таким чином, час транзакції для одного ПЛК, наприклад PLC1 (TT1), буде дорівнювати:

$$TT1 = C5 + T1.req + 2 \cdot C1 + T1.res + C5 \cdot 2 \quad (6.1)$$

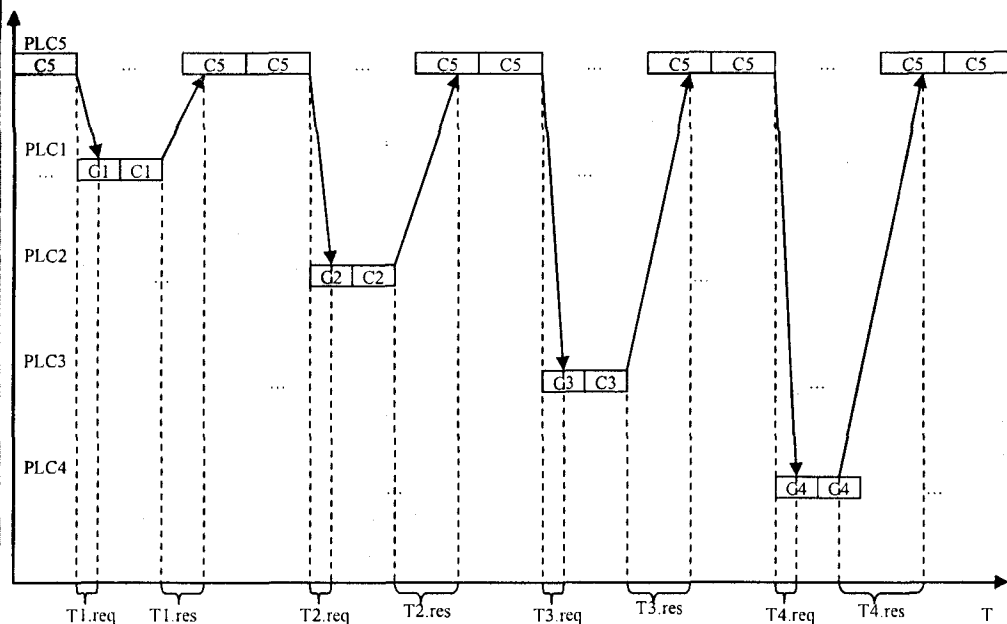


Рис. 6.17. Часова діаграма до прикладу 6.4

ТТ1 має враховувати 2 цикли, затрачених Веденим на генерацію відповіді на повідомлення-запит. Якби транзакція проводилась не періодично, як за умовою завдання, а за виникненням події, то в час транзакції необхідно було б включити також ще один цикл Ведучого. Нескладно вивести час опитування всіх Ведених:

$$TT_{all} = C5 \cdot 9 + C1 \cdot 2 + C2 \cdot 2 + C3 \cdot 2 + C4 \cdot 2 + T1.req + T1.res + T2.req + T2.res + T3.req + T3.res + T4.req + T4.res \quad (6.2)$$

Враховуючи, що цикли Ведених однакові, а кадри запитів і кадри відповідей для всіх Ведених мають однакову структуру, загальна формула буде мати такий вигляд:

$$TT_{all} = C5 \cdot 9 + C1 \cdot 8 + (T1.req + T2.req) \cdot 4 \quad (6.3)$$

Розрахуємо час T1.req та T2.req. Час передачі кадру (Tframe) можна орієнтовно розрахувати за кількістю символів (Nsymb) у кадрі та часу передачі одного символу (Tsymb):

$$Tframe = Nsymb \cdot Tsymb \quad (6.4)$$

Час передачі одного символу розраховується:

$$\begin{aligned} \text{час передачі одного символу} &= \\ &= \text{кількість бітів у символі} / \text{бітова швидкість}; \end{aligned}$$

Час передачі кадрів буде дорівнювати (див.рис.6.16 та рис.6.17):

$$\begin{aligned} T1.req &= 8 \cdot (11/19200) = 4,58 \text{ мс} \\ T1.res &= 25 \cdot (11/19200) = 14,33 \text{ мс} \\ TT_{all} &= 90 + 40 + (4,58 + 14,33) \cdot 4 = 206 \text{ мс}. \end{aligned}$$

Таким чином, для опитування 10-ти змінних з 4-х Ведених зі швидкістю 19200 біт/с необхідно затратити приблизно 206 мс. Якщо необхідне періодичне опитування, то бажано зарезервувати певний час, наприклад, ще додатково 100 мс.

Приклад 6.4

У ряді випадків реалізацію функцій MODBUS-Клієнта покладено на операційну систему, а доступ до них у програмі ПЛК відбувається через інтерфейсні комунікаційні функції. Зокрема, це характерно для більшості ПЛК від Schneider Electric (Momentum, Quantum, TSX Micro, TSX Premium, M340). У ряді інших систем клієнтську сторону на прикладному рівні необхідно повністю прописувати в програмі ПЛК, а інтерфейс надається тільки для обміну з комунікаційним портом. У цьому випадку система надає сервіси відправки та отримання повідомлення, яке формує і аналізує сама програма користувача, і генерації та перевірки контрольної суми. Розглянемо приклад.

Приклад 6.5. MODBUS. Реалізація MODBUS-клієнта на TSX Twido

Завдання. Записати фрагмент програми в TSX Twido для зчитування 3-х регістрів з Веденого з адресою 1 (рис. 6.18).

Рішення. У Twido клієнтську сторону MODBUS необхідно реалізовувати через універсальну функцію EXCHx, яка відправляє та/або отримує дані через

комунікаційний порт з номером x . Параметрами функції є таблиця слів (%MW), в яких розміщуються дані управління функцією, дані для відправки та буфер для приймання. Якщо обмін буде проходити через комунікаційний порт 2, то виклик функції матиме такий формат:

EXCH2 %MWy:n,

де y — номер першої змінної виділеної таблиці,
 n — кількість слів у таблиці.

Формат таблиці, тобто даних, які необхідно заповнити, та область даних для приймання однаковий для всіх типів комунікацій. Для функцій 03/04 (читання N слів) на MODBUS-RTU ця таблиця буде мати вигляд, наведений у табл. 6.2.

Таблиця параметрів складається із 3-х частин-підтаблиць. У таблиці управління функцією задаються параметри для самої функції. Так, у старшому байті 0-го слова вказується, що ця функція працює в обидва боки, тобто після відправки даних необхідно чекати відповіді. Молодший байт цього ж слова вказує на довжину таблиці передачі (в даному випадку 6 байтів) для того, щоб система знала, які байти необхідно передати (з 2-го слова по 4-те) і звідки починається буфер прийому (з 5-го слова). Зміщення в передачі та прийомі необхідно для вирівнювання даних у буферах за словами.

Таблиця передачі вміщує безпосередньо сам запит, тобто це кадр без кода CRC. Таблиця прийому — це буфер, який система заповнить кадром відповіді при позитивному результаті. Таким чином, щоб скористатися цією функцією, спочатку необхідно побудувати кадр запиту та відповіді за виключенням поля CRC (рис. 6.19)

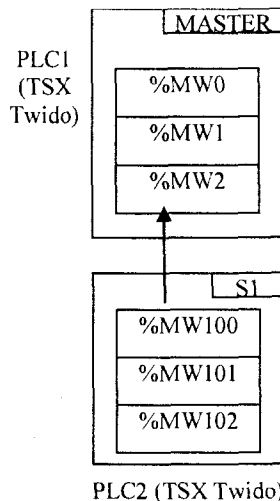


Рис. 6.18. Постановка завдання до прик

ТАБЛИЦЯ ПАРАМЕТРІВ

Таблиця 6.2

	Індекс у таблиці	Старший байт	Молодший байт
Таблиця управління функцією	0	01 (тип ф-ції відправка + прийом)	06 (довжина таблиці передачі)
	1	03 (зміщення в прийомі)	00 (зміщення в передачі)
Таблиця передачі	2	адреса веденого	03 (номер функції)
	3	адреса початкового регістру	
	4	кількість регістрів	
Таблиця прийому (повідомлення-відповідь)	5	адреса веденого	03 (номер функції)
	6	00 (байт для зміщення)	лічильник байтів
	7	перший регістр	
	8	другий регістр	
	
	N+6	N-ний регістр	

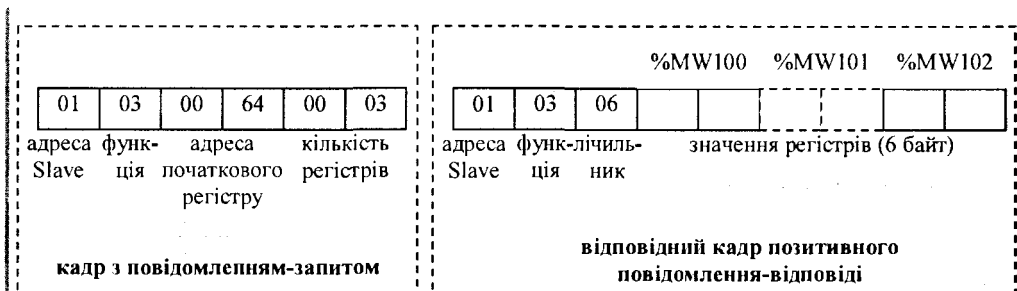


Рис. 6.19. Формат кадрів до запити та відповіді без CRC для прикладу 6.5

Як бачимо, в запиті — 6 байтів. Цю кількість необхідно вписати в молодший байт 0-го слова таблиці. У відповіді очікується 9-байт. Якщо байти кадру відповіді розмістити в послідовності слів (в ПЛК Schneider Electric пам'ять адресується словами), то старший байт першого прийнятого регістру (згідно з умовою %MW100) припаде на молодший байт 2-го слова у буфері, а молодший байт прийнятого регістру — на старший байт 3-го слова в буфері. Таким чином, усі прийняті слова будуть зміщені і прочитати їх буде проблематично. Для усунення цієї проблеми в таблиці параметрів функції є поле зміщення прийому, в якому вказується номер байта у буфері прийому, який зсуватиме всю послідовність.

Заповнивши всі поля, фрагмент програми буде мати вигляд, як на рис. 6.20.

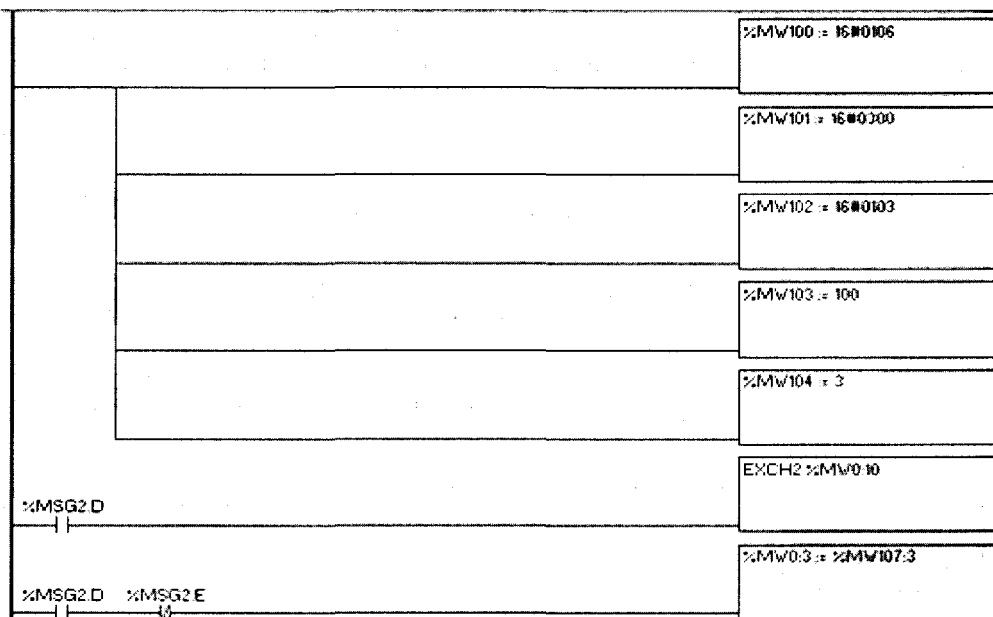


Рис. 6.20. Фрагмент програми для реалізації завдання з прикладу 6.5

Верхній ланцюг LD призначений для заповнення таблиці управління функцією і таблиці передачі.

У другому ланцюзі проводиться безпосередньо виклик функції. Змінна %MSG2.D повертає логічну 1, коли функція EXCH2 оброблена і результат отриманий. Її використання не дає «затопити» мережу надмірною кількістю кадрів, адже поки немає відповіді на попередній запит або не пройшов час тайм-ауту, новий запит відправляти не можна.

Останній ланцюг призначений для запису результату читання в змінні %MW0:3 (таблиця з 3-х слів починаючи з %MW0). Змінна %MSG2.E буде рівною 1-ці тоді, коли є помилка у виклику функції.

Приклад 6.5

6.3.4. Реалізація фізичного рівня для MODBUS Serial

На відміну від початкової специфікації, яка обмежувалась описом кадру, в стандарті MODBUS-IDA описуються також правила для реалізації мережі на фізичному рівні. MODBUS over Serial Line базується на використанні послідовних інтерфейсів RS-485, RS-422 та RS-232.

Для RS-485 визначена топологія — це шина, до якої передбачено три способи підключення пристроїв (рис. 6.21):

- безпосередньо до магістрального (trunk) кабелю, без відгалуження;
- через пасивну коробку підключення та кабель відгалуження (Derivation);
- через активну коробку та специфічний кабель відгалуження.

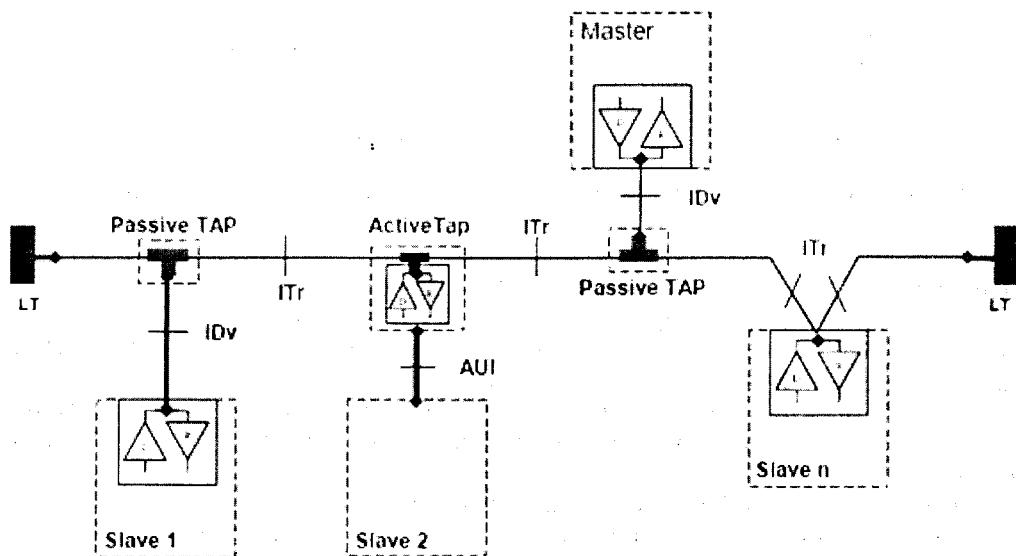


Рис. 6.21. Інфраструктура шини MODBUS для послідовного інтерфейсу RS-485

Інтерфейси між кабелями та елементами мережі мають такі позначення (див. рис. 6.21): ITг — інтерфейс до магістрального кабелю; IDv — інтерфейс між пристроєм та пасивною коробкою; AUI — інтерфейс між пристроєм та активною коробкою; LT — термінатори лінії.

Бітові швидкості визначені рівними 9600 біт/с та 19200 біт/с (по замовченню). Інші швидкості є опціональними. Використовується метод кодування NRZ.

При використанні RS-485 стандарт MODBUS визначає правила підключення пристроїв за 2-х провідною та 4-хпровідною схемами, а також правила сумісності 2-х провідних та 4-х провідних інтерфейсів на єдиній лінії. Нижче розглянуте тільки 2-х провідне підключення, яке є обов'язковим.

По суті, 2-х провідне підключення насправді є 3-х провідним, тому що, крім ліній A-(D0) та B+(D1), використовується також загальна лінія C(Common), яка є обов'язковою (рис. 6.22).

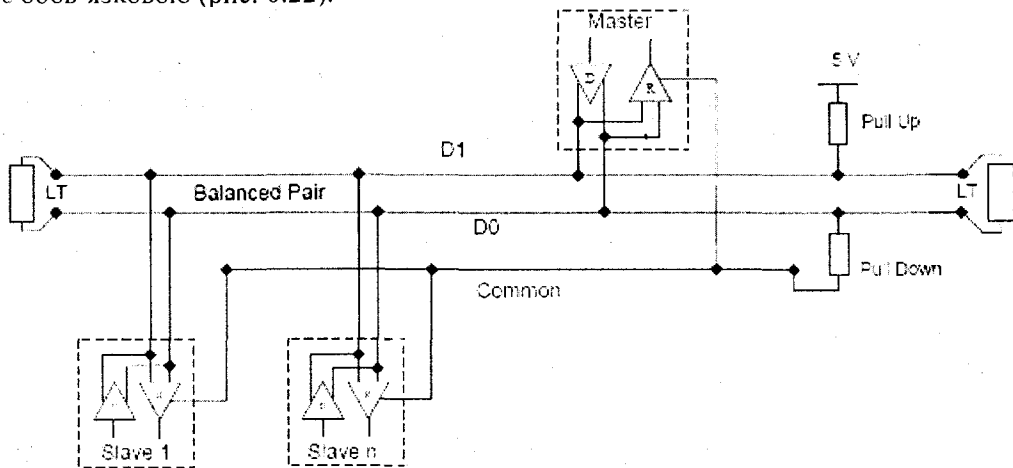


Рис. 6.22. 2-провідне з'єднання пристроїв MODBUS за RS-485

Загальна кількість пристроїв обмежена: 32 пристрої на одному сегменті RS-485, використання репітерів дозволяється. Максимальна довжина кабелю залежить від швидкості, кабелю, кількості навантажень і конфігурації мережі (2-х провідна або 4-х провідна). Для бітової швидкості 9600 і кабелю AWG26 максимальна довжина обмежена 1000 м. Кабель відгалуження повинен бути коротшим за 20 м. Якщо використовується мультипортова коробка з n -портами, то кожен кабель відгалуження обмежений довжиною $40/n$ м.

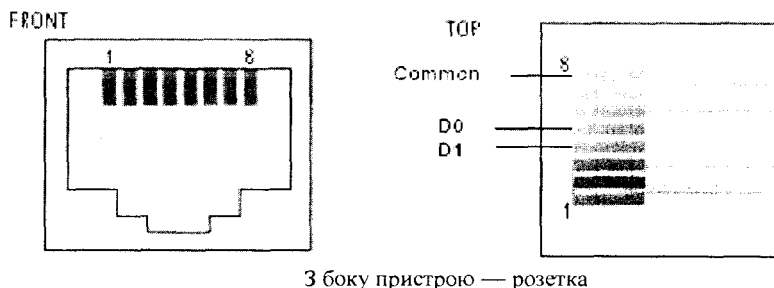
Загальний сигнальний провід (Common) обов'язково з'єднується з екраном в одній точці шини, яка, як правило, міститься на вузлі Ведучого або його коробки відгалуження.

Для погашення відбиття хвиль на кінцях лінії між D1 та D0 виставляються термінатори лінії (LT). Термінатори дозволяється виставляти тільки на магістральному кабелі. В якості термінаторів можна використати:

- резистор номіналом 150 Ом та потужністю 0.5 Вт;
- послідовно з'єднані конденсатор (1 нФ, 10 В мінімум) та резистор номіналом 120 Ом (0.25 Вт) при використанні поляризації лінії.

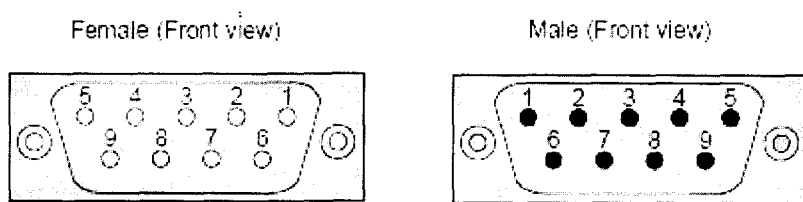
У стандарті MODBUS Serial визначені правила реалізації захисного зміщення (поляризації), які передбачають підключення живлення номіналом 5 В між D1 та D0 через PullUp та PullDown-резистори для утримання логічної «1» на лінії за відсутності передачі. Номінал резисторів вибирається від 450 Ом до 650 Ом залежно від кількості пристроїв (650 Ом за великої кількості). Захисне зміщення проводиться тільки в одній точці лінії, зазвичай, на боці Ведучого. Максимальна кількість пристроїв з реалізованою поляризацією зменшується на 4 порівняно із системою без поляризації. Поляризація необов'язкова. Однак ряд пристроїв критично налаштовані на відсутність логічного сигналу. Якщо це так, то поляризацію необхідно реалізовувати самостійно або використовувати існуючі схеми, якщо такі передбачені пристроями. В іншому випадку пристрої, які потребують поляризації, не будуть нормально працювати в мережі.

Стандарт визначає також механічний інтерфейс, тобто типи роз'ємів, вилок та відповідність сигналів на контактах. В якості механічного терміналу можна використовувати клемну колодку, екранований RJ-45 (рис. 6.23) або екранований SUB-D9 роз'єм (рис. 6.24).



З боку пристрою — розетка

Рис. 6.23. Використання RJ-45 типу з'єднувачів



З боку пристрою — розетка

Рис. 6.24. Використання 9-пінового Sub-D типу з'єднувачів

У таблиці 6.3 вказані призначення контактів для конекторів при 2-х провідному підключенню по RS-485, а в таблиці 6.4 — по RS-232.

В якості кабелів для 2-х провідного типу з'єднання стандарт визначає подвійну екрановану виту пару категорій 4 (до 600 м) або 5 (до 1000 м), де в одній парі йдуть збалансовані сигнали D0 та D1, а в другій — сигнальна земля Common. Рекомендовані кольори кабелів: D1 жовтий; D0 коричневий; Common-сірий.

Таблиця 6.3

ПРИЗНАЧЕННЯ КОНТАКТІВ КОНЕКТОРА ПРИ ПІДКЛЮЧЕННІ ПО RS-485

номери контактів		вимоги до наявності	контур IDv	контур ITr	назва RS-485	коментар (див. розділ 3)
RJ45	SUB-D9					
3	3	опціонально	PMC	—	—	управління режимом ком. порту
4	5	обов'язково	D1	D1	B/B'	напруга V1, V1>V0 для лог. «1»
5	9	обов'язково	D0	D0	A/A'	напруга V0, V0>V1 для лог. «0»
7	2	рекомендовано	VP	—	—	+ живлення 5–24 VDC
8	1	обов'язково	Common	Common	C/C'	– живлення та сигнальна земля

Таблиця 6.4

ПРИЗНАЧЕННЯ КОНТАКТІВ КОНЕКТОРА ПРИ ПІДКЛЮЧЕННІ ПО RS-232

DCE (модем)			контур			DTE		
номери контактів		вимоги до наявності	назва	коментар (див. розділ 3)	джерело RS-232	вимоги до наявності	номери контактів	
RJ45	SUB-D9						RJ45	SUB-D9
1	2	обов'язково	TxD	Transmitted Data	<< DTE	обов'язково	2	3
2	3	обов'язково	RxD	Received Data	DCE >>	обов'язково	1	2
3	7	опціонально	CTS	Clear to Send	DCE >>	опціонально	6	8
6	8	опціонально	RTS	Request to Send	<< DTE	опціонально	3	7
8	5	обов'язково	Common	Signal Common		обов'язково	8	5

Приклад 6.6. MODBUS. Схема мережних з'єднань MODBUS RTU

Завдання. Нарисувати схему мережних з'єднань для 2-х провідної реалізації шини MODBUS RTU з такими вузлами:

PLC1: VIPA CPU 115SER 6BL32 (Ведучий) через убудований послідовний порт процесорного модуля;

PLC2: TSX Twido TWDLMDA40DTK (Ведений) через комунікаційний модуль TWD NOZ 485T;

PLC3: TSX Twido TWDLMDA40DTK (Ведений) через комунікаційний модуль TWD NOZ 485T.

Рішення. На рис. 6.25 показана схема мережних з'єднань до поставленого завдання. Специфікація мережних засобів дана в таб.6.5.

Як видно з рис. 6.25, PLC1 підключається до шини через пасивну коробку, а точніше — через клемну колодку, що в принципі рівнозначно. Це спричинене-

но тим, що на ПЛК підключення відбувається з використанням 9-пінового SUB-D роз'єму, що потребує розробки власного кабелю, схема підключення (спаю) якого до коннектора та до клемної колодки показана нижче основної схеми.

Таким чином, до вилки KK1 проводи кабелю KM2 необхідно припаяти. Призначення пін розетки SER не співпадає зі стандартною. Піни 8 та 3, відповідно A(D0) та B(D1), йдуть в одну пару, яка потім підключається до XT1:1 та XT1:2, а 5 та 6, відповідно M5V(-5В) та P5V(+5В), йдуть в іншу пару кабелю KM2. Живлення 5В необхідне для того, щоб реалізувати захисне зміщення (поляризацію) відповідно до стандарту, крім того -5В є сигнальною землею (Common).

Кабель KM2 підключається до XT1 відповідно до схеми, показаної на рис. 6.25. Екран кабелю з'єднується із сигнальною землею відповідно до вимог стандарту. Слід нагадати, що ПЛК VIPA в цій системі є Ведучим, отже і захисне зміщення і з'єднання екрана із землею необхідно реалізувати саме в цьому місці. Захисне зміщення проводиться за допомогою живлення 5В, яке береться з порту SER та двох резисторів.

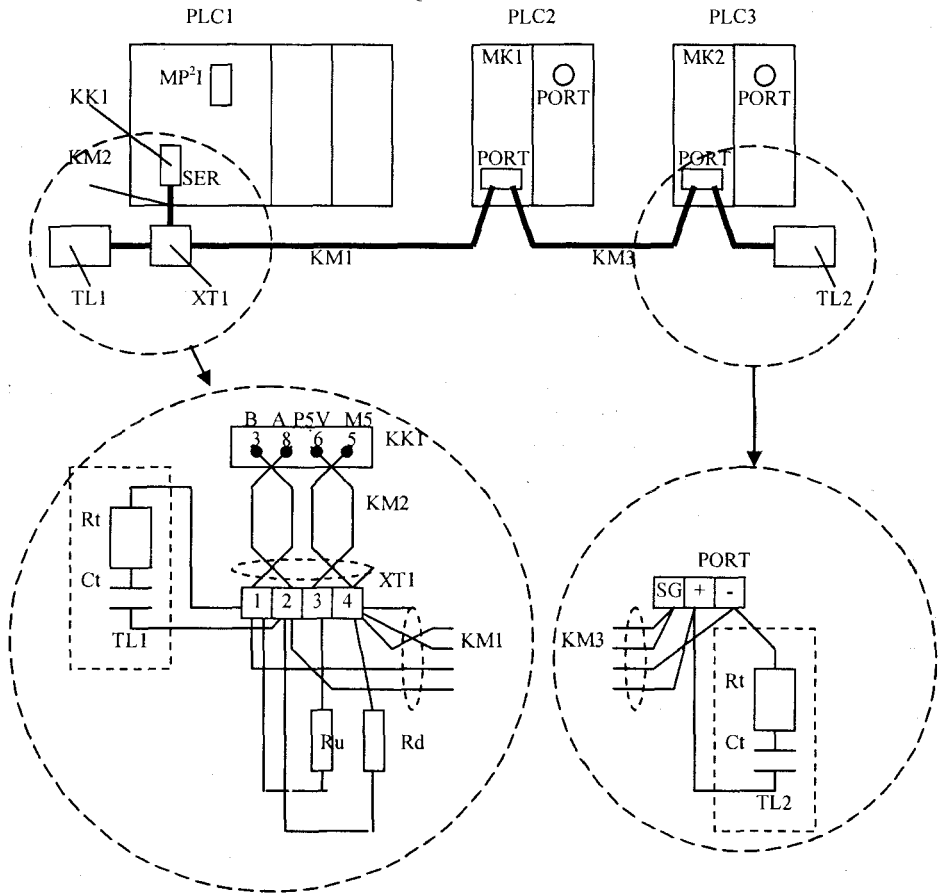


Рис. 6.25. Схема мережних з'єднань до прикладу 6.6

Таблиця 6.5

СПЕЦИФІКАЦІЯ МЕРЕЖНИХ ЗАСОБІВ

№	Позначення	Найменування	Назва	Кількість	Примітка
1	PLC1	ПЛК VIPA 100	VIPA CPU I15SER 6BL32	1 шт.	VIPA
2	PLC2, PLC3	ПЛК Twido	TWDLMDA40 DTK	2 шт.	Schneider Electric
3	МК1, МК2	комунікаційний модуль для реалізації інтерфейсу RS-485, підключення під гвинт	TWD NOZ 485T	2 шт.	Schneider Electric
4	KK1	9-піновий SUB-D конектор типу вилка		1 шт.	
5	XT1	клемна колодка на 4 клеми		1 шт.	
6	TL1, TL2	термінатори лінії		2 шт	виготовляються з поз. 7 та 8
7	Rt	Резистор 120 Ом (0.25 Вт)		2 шт.	у складі поз.6
8	Ct	Конденсатор 1 нФ (>10 В)		2 шт.	у складі поз.6
9	Ru, Rd	Резистор 500 Ом (0.25 Вт)		2 шт	
10	KM1	Кабель, подвійна екранована вита пара 5-ї категорії AWG26		300 м	
11	KM2	Кабель, подвійна екранована вита пара 5-ї категорії AWG26		2 м	
12	KM3	Кабель, подвійна екранована вита пара 5-ї категорії AWG26		300 м	

PLC2 та PLC3 з'єднуються з шиною за допомогою комунікаційного модуля з клемною колодкою. Це дозволяє реалізувати підключення без відгалуження. Однак на колодці не передбачене місце підключення екрана, тому кабель екранується окремо.

Термінатори ліній реалізовані послідовним з'єднанням резистора та конденсатора, оскільки на шині задіяне захисне зміщення.

Приклад 6.6

Нині MODBUS Serial використовується як на рівні контролерів, так і на рівні датчиків (для розподіленої периферії). Його використання проблематичне за наявності на шині декількох диспетчерських пристроїв, які в Клієнт-Серверній архітектурі повинні бути Клієнтами, адже на MODBUS RTU/ASCII тільки Ведучий може бути Клієнтом. Але навіть у такій ситуації нерідко є можливість організувати доставку даних потрібним вузлам.

Виходячи зі сказаного, на шині MODBUS-Serial можна зупинити свій вибір у випадку, якщо:

- всі пристрої-Сервери підтримують MODBUS-RTU/ASCII у режимі Веденого;
- необхідний тільки один пристрій-Клієнт, якому треба ініціювати обміни на шині, який підтримує MODBUS RTU/ASCII як Ведучий;

- швидкість відновлення даних задовольняє умову завдання;
- немає необхідності в пріоритетних повідомленнях (телеграмах);
- не ставляться особливі вимоги до надійності шини.

Альтернативою MODBUS-RTU/ASCII є MODBUS-TCP/IP, який дуже швидко став популярним і не має деяких недоліків з тих, які наведені вище.

6.4. MODBUS TCP/IP

6.4.1. Комунікаційна архітектура MODBUS TCP/IP

Мережа MODBUS TCP/IP базується на стеку протоколів TCP/IP і, перш за все, призначена для роботи на базі мереж Ethernet. MODBUS-TCP/IP описаний у специфікаціях MODBUS-IDA, в яких комунікаційна система MODBUS TCP/IP може включати різні типи пристроїв (рис. 6.26):

- MODBUS TCP/IP Клієнти і Сервери підключені до TCP/IP мережі;
- міжмережні пристрої типу мостів, маршрутизаторів або шлюзів для з'єднання TCP/IP мережі з послідовними лініями підмереж, що дозволяє обмінюватися даними з MODBUS Serial Серверними пристроями.

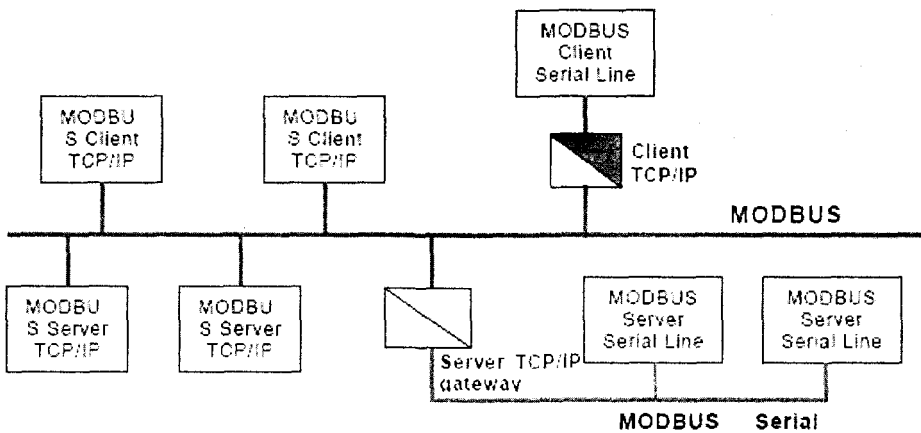


Рис. 6.26. Комунікаційна архітектура MODBUS-TCP/IP

Таким чином комунікаційна система MODBUS-TCP/IP дозволяє обмінюватися пристроям не тільки на мережах зі стеком TCP/IP, а і з пристроями на послідовних лініях зв'язку (MODBUS-RTU/ASCII або MODBUS+). Як приклад, можна навести рис. 6.27, узятий зі специфікації протоколу прикладного рівня. Слід зазначити, що на сьогоднішній день розроблені механізми маршрутизації SDO запитів з мереж MODBUS-TCP/IP на мережі CANOpen.

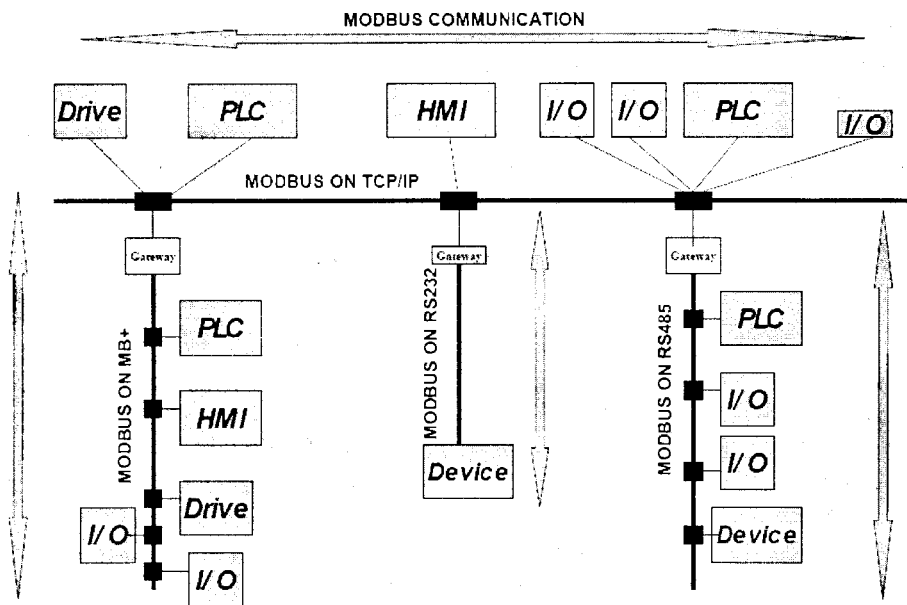


Рис. 6.27. Приклад архітектури мережі MODBUS

6.4.2. Особливості реалізації протоколу

Аналогічно всім мережам MODBUS, дана мережа використовує MODBUS Application Protocol. Нагадаємо, що в MODBUS Serial на каналному рівні до PDU добавляється адреса Веденого і контрольна сума, сам PDU не модифікується. Однак, у MODBUS TCP/IP перед попаданням на транспортний рівень до PDU (код функції та дані) додається додатковий MBAP-заголовок. (рис. 6.28). Заголовок складається з полів, які описані в табл. 6.6. Отриманий модуль передається рівню TCP.

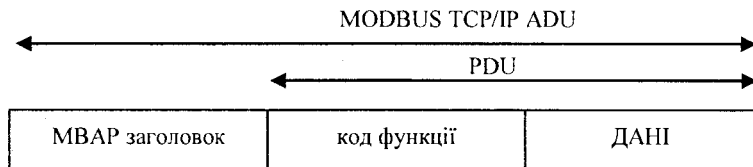


Рис. 6.28. Формат модуля даних прикладного рівня (APDU) MODBUS TCP/IP

Зверніть увагу, що за допомогою поля UnitID можна вказати адресу вузла в MODBUS Serial, наприклад, адресу Веденого в MODBUS RTU. Якщо потрібно адресувати вузол, безпосередньо підключений по TCP/IP, то UnitID = 0.

Поле ProtocolID використовується для міжсистемного мультиплексування. Так, TCP — порт для MODBUS-Серверу має номер 502, однак цей же порт, наприклад, використовує Schneider Electric для UNITE-Серверу. Таким чином, за-

мінивши поле ProtocolID, Ethernet модулі ПЛК TSX Premium одночасно підтримують два протоколи: MODBUS та UNI-TE.

Таблиця 6.6

ПОЛЯ МВАР ЗАГОЛОВКА

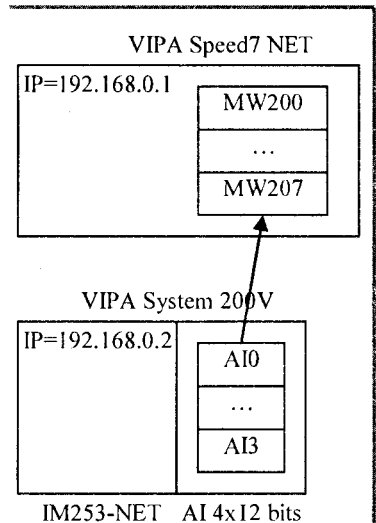
Поле	Довжина (байт)	Пояснення	Клієнт	Сервер
TransactionID	2	ідентифікація транзакцій запитів/ відповідей	ініціалізує Клієнт у запиті	копіює із запиту у повідомлення — відповідь
ProtocolID	2	тип протоколу, 0 = MODBUS протокол	ініціалізує Клієнт у запиті	копіює із запиту у повідомлення — відповідь
Length	2	кількість наступних байтів	ініціалізує Клієнт у запиті	ініціалізує Сервер у відповіді
UnitID	1	адреса Веденого, який підключений до вузла	ініціалізує Клієнт у запиті	копіює із запиту у повідомлення — відповідь

Для ідентифікації пристрою, якому передається запит, вказується його IP-адреса. Для ідентифікації TPDU, які направляються MODBUS-Серверу, як прикладному об'єкту, виділений 502-гий TCP-порт. З деталями функціонування стека TCP/IP, а також Ethernet можна ознайомитися в літературі з комп'ютерних мереж, а також у розділі 10. Крім того, там наведені деякі пояснення з приводу обмежень використання рішень подібних MODBUS TCP/IP, зокрема в системах з жорсткими вимогами до реального часу.

Приклад 6.7. MODBUS. Реалізація MODBUS TCP/IP Клієнтської програми для зв'язку VIPA Speed7 з віддаленими модулями вводу/виводу VIPA IM 253NET

Завдання. Написати прикладну програму в середовищі Step7 для реалізації періодичного (1 раз в 500мс) зчитування значення 4-х аналогових входів у ПЛК Vipa Speed7NET з віддалених модулів вводу/виводу на базі VIPA IM 253NET (рис. 6.29).

Рішення. Даний приклад розглянутий з максимальними спрощеннями і без деталізації, яка не стосується даної теми. Враховуючи, що для програмування VIPA можна скористатися іншим програмним продуктом (WinPLC7), деталі створення конфігурації для Step7 у прикладі упущені. Слід також відмітити, що Siemens пропонує свій комунікаційний функціональний блок як для реалізації Клієнта, так і Серверу MODBUS TCP/IP, однак, в даному прикладі використовуються стандартні функції AG SEND та AG RECV.



IM253-NET AI 4x12 bits

Рис. 6.29. Постановка завдання до прикладу 6.7

З рис. 6.29 видно, що 4-ри 12-бітні входи будуть зчитуватися як 4 слова. Оскільки вони перебувають на першому посадочному місці, змінна AI0 — буде адресуватись як 0-вий вхідний регістр, а AI3 — як 3-тій. Таким чином, завдання можна сформулювати так: написати прикладну програму в ПЛК Vipa Speed7NET (IP = 192.168.0.1) для реалізації зчитування по MODBUS TCP/IP перших 4-х вхідних регістрів із Серверу з адресою IP = 192.168.0.2. При такій постановці вже не важлива конкретика реалізації віддаленого пристрою.

Для повноти картини наведемо схему мережних з'єднань (рис. 6.30). Слід відмітити, що у VIPA S7 NET серед 2-х портів Ethernet X8 та X5 тільки один порт, а саме X8(TP), може використовуватись для різних типів з'єднань, аналогічних CP343 від Siemens. Інший порт (X5) може використовуватись тільки для PG/OP — з'єднань. Таким чином, для з'єднання даного ПЛК за MODBUS TCP/IP необхідно використати роз'єм X8 (TP).

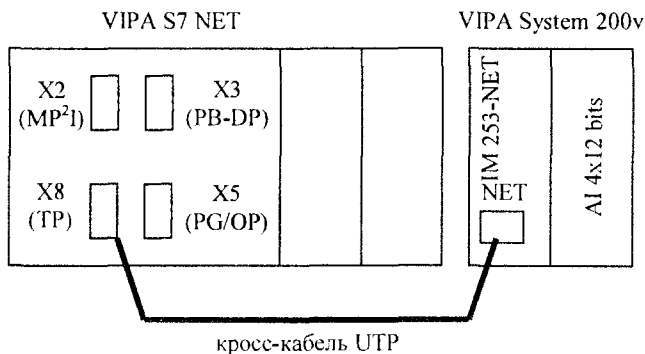


Рис. 6.30. Схема мережних з'єднань до прикладу 6.7

Для реалізації обміну за протоколом TCP у S7-подібних контролерах конфігурують з'єднувальні канали (Connections). Для кожного каналу вказують:

- локальний процесор, для якого конфігурується канал;
- точку виходу в мережу, тобто комунікаційний модуль;
- ідентифікатор з'єднання;
- активність з'єднання, що означає, чи ініціює даний вузол з'єднання;
- адреса комунікаційного партнера, з яким налаштовується з'єднання (IP та TCP port);
- локальний TCP port для з'єднання.

У принципі, використовуючи з'єднувальні канали, можна обмінюватися поверх TCP за будь-яким відкритим протоколом. Тому MODBUS TCP/IP є одним із багатьох варіантів використання даного типу комунікацій Siemens. Так, наприклад, Step5-сумісний зв'язок — теж конфігурується подібним чином, однак порти TCP будуть інші і протокол прикладного рівня теж інший.

Для розв'язання завдання створюється з'єднувальний канал з такими характеристиками:

- канал створюється на боці процесора VIPA S7 NET;

- точка виходу в мережу (маршрутизатор) — вибираємо порт TP (віртуальний модуль CP343), після чого в конфігураторі отримується адреса даного модуля відповідно до його розміщення в корзині ($=140_{16}$);
- тип каналу = TCP;
- ідентифікатор з'єднувального каналу вибираємо довільний, наприклад 3;
- з'єднання активне з цього боку, чим вказуємо, що даний вузол в з'єднанні представляє бік клієнта TCP;
- вказуємо IP-адресу комунікаційного партнера, тобто IP = 192.168.0.2;
- вказуємо TCP port для партнера, TCP = 502 (MODBUS Сервер);
- вказуємо локальний, тобто клієнтський TCP-port, вибираємо довільний у дозволеному діапазоні, наприклад, TCP = 2001.

Після такої конфігурації, навіть без написання додаткової програми, між ПЛК та модулем віддаленого вводу/виводу створиться TCP-з'єднання. Однак, ніякими даними користувача вузли обмінюватися не будуть.

Для передачі даних від ПЛК VIPA S7 NET необхідно скористуватися комунікаційною функцією AG_SEND, яка поставляється з бібліотеками для VIPA. Ця функція відправляє блок даних за створеним з'єднувальним каналом. Одним із аргументів цієї функції є вказівник на блок даних, які необхідно відправити. В нашому випадку це, по суті, MODBUS TCP/IP ADU. Найбільш зручний спосіб описати в Step7 подібну структуру та виділити для неї пам'ять — це використати блоки даних типу DB. Таким чином, формуємо DB10 для створення MODBUS TCP/IP ADU, відповідно до таблиці 6.5 та синтаксису функції читання вхідних регістрів (див. п. 6.2.2). Результат формування блока DB10 в Step7 показаний на рис. 6.31.

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	TransactionID	WORD	W#16#0	ідентифікація транзакцій
+2.0	ProtocolID	WORD	W#16#0	0=Modbus
+4.0	LenByte	WORD	W#16#6	кількість таких байтів=6
+6.0	UnitID	BYTE	B#16#0	адреса Веденого=0 (адресуємо вузол ethernet)
+7.0	Funct	BYTE	B#16#4	номер функції (читання регістрів)
+8.0	StartAddress	WORD	W#16#0	адреса початкового регістру (з 0-по регістру)
+10.0	CountReg	WORD	W#16#4	кількість регістрів
+12.0		END_STRUCT		

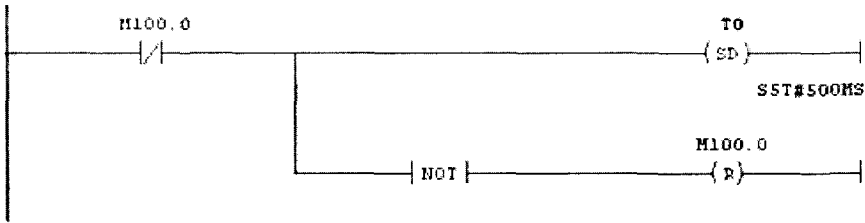
Рис. 6.31. Блок DB10 для формування клієнтського запиту MODBUS до прикладу 6.7

Перші два поля при ініціалізації блока прирівнюються нулю. Поле TransactionID можна використати для відслідковування послідовності відповідей на запиту (який TransactionID в запиті, такий і у відповіді), однак це не обов'язково. ProtocolID=0, бо використовується Сервер MODBUS.

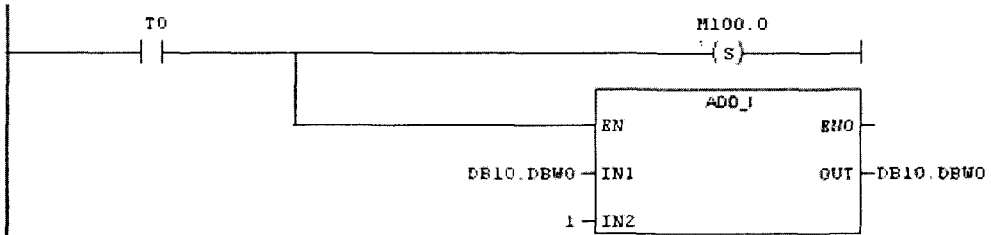
Кількість наступних байтів залежить від функції, в нашому випадку їх 6. Адреса Веденого=0, тому що адресат міститься безпосередньо на Ethernet. Функція=4, оскільки необхідно зчитати вхідні регістри, початкова адреса=0, а кількість=4 (див. рис. 6.28).

Вказівник на сформований блок даних указується в аргументах функції AG_SEND. Частина програми, яка відповідає за відправку запитів, буде мати вигляд, як на рис. 6.32.

Network 1 : старт/рестарт 0-го таймера кожні 500 мс



Network 2 : кожні 500 мс M100.0 виставляється в 1



Network 3 : Відправка повідомлення кожні 500 мс (M100.0)

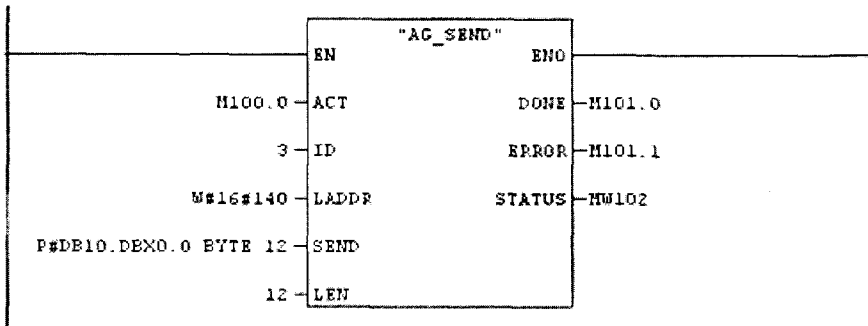


Рис. 6.32. Частина програми (Network1-3), яка відповідає за відправку запитів, (до прикладу 6.7)

Програма написана в організаційному блоці OB1 (циклічний виклик). В першому ланцюжку (Network1) кожні 500 мс перезапускається таймер і генерується одиничний імпульс M100.0 тривалістю в один цикл. Цей імпульс використовується для відправки даних в мережу (параметр ACT) раз в 500 мс. З цією ж періодичністю в Network2 збільшується значення 0-го слова в блоці

даних DB10 (DB10.DBW0), яке вказує на ідентифікатор транзакції. Крім безпосередньо вказівника на блок з даними для відправки довжиною 12 байт (параметр SEND), у функції вказується ідентифікатор з'єднувального каналу (параметр ID=3), адреса комунікаційного модуля (LADDR=140₁₆) та кількість байтів для відправки. Контроль відправки та виклику функцій проводиться за допомогою параметрів DONE (біт виконання операції), ERROR (біт наявності помилки) та STATUS (слово статусу операції), про використання яких можна можна прочитати в довідниковій літературі.

Після запуску програми на виконання процесор ПЛК буде записувати вказані в DB10 дані у вихідний буфер комунікаційного модуля з адресою LADDR=140₁₆, які відправляються цим модулем з'єднувальним каналом TCP з ідентифікатором 3. Інтерфейсний модуль розподіленої системи вводу/виводу VIPA IM 253NET, отримавши за з'єднувальним каналом дані MODBUS TCP/IP ADU, сформує повідомлення-відповідь і відправить його цим же каналом. Це повідомлення попадає у вхідний буфер комунікаційного модуля. Для того, щоб прочитати дані з вхідного буфера, використовується функція AG_RECV.

В якості одного з параметрів функції AG_RECV є вказівник на дані, куди необхідно помістити результат функції, тобто дані вхідного буфера. Аналогічно попередній частині програми, найкраще використати блок даних, наприклад DB11. Значення ініціалізації полів даного блока не відіграє ніякої ролі, оскільки заповнюється значеннями буфера. Однак, бажано задати структуру блоку, яка і показана на рис. 6.33.

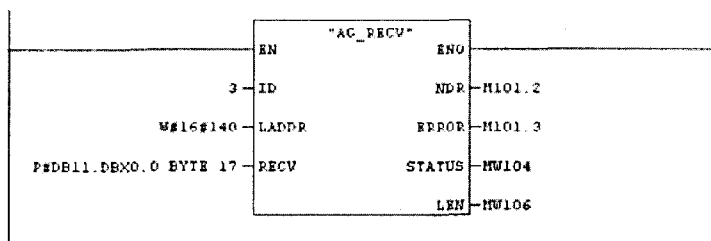
Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	TransactionID	WORD	W#16#0	ідентифікація транзакцій
+2.0	ProtocolID	WORD	W#16#0	0=Modbus
+4.0	LenByte	INT	11	кількість наступних байтів=6
+6.0	UnitID	BYTE	B#16#0	адреса Веденоло=0 (адресуємо вузол ethernet)
+7.0	Funct	BYTE	B#16#4	номер функції (читання регістрів)
+8.0	CountByte	BYTE	B#16#0	кількість байт
+10.0	Register0	INT	4	значення
+12.0	Register1	INT	0	значення
+14.0	Register2	INT	0	значення
+16.0	Register3	INT	0	значення
+18.0		END_STRUCT		

Рис. 6.33. Блок DB11 для отримання відповіді серверу MODBUS
(до прикладу 6.7)

Друга частина програми, в якій реалізоване отримання необхідних значень аналогових входів, показана на рис. 6.34.

У випадку позитивного результату виконання функції та отримання нових даних (NDR=0 та ERROR=0) записуємо значення в змінні MW200-MW206, відповідно до завдання. Слід зазначити, що даний приклад значно спрощений, зокрема в ньому не враховані випадки виникнення комунікаційних помилок.

Однак, він показує, яким чином використовується клієнтський бік MODBUS TCP/IP в Siemens-сумісних ПЛК. Аналогічно можна прописати і серверний бік MODBUS TCP/IP, однак це набагато складніше завдання.



Network 5: якщо отримані нові значення без помилок записати їх в змінні

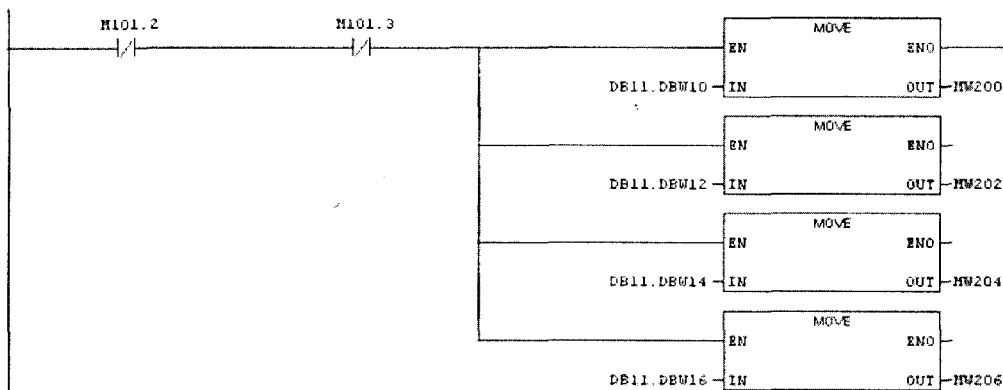


Рис. 6.34. Частина програми (Network 4-5), яка відповідає за отримання відповіді з вхідного буфера комунікаційного модуля (до прикладу 6.7)

Приклад 6.7

Приклад 6.8. MODBUS. Реалізація MODBUS TCP/IP Клієнтської програми для зв'язку прикладних програм з інтегрованим VBA та ПЛК

Завдання. Написати прикладну програму в середовищі VBA (Excel) для реалізації зчитування значення 5-ти внутрішніх регістрів з ПЛК за протоколом MODBUS TCP/IP. Зчитування проводити натисканням кнопки.

Рішення. У VBA прописуємо код, що показаний на рис. 6.35.

Для реалізації даного завдання використаний ActiveX-елемент *Winsock* (має назву Winsock1), який, як правило, вже присутній на комп'ютерах з операційною системою Windows. Цей компонент є інтерфейсом до однойменного сервісу в операційній системі Windows і функціонує приблизно так, як модель сокетів (див.розділ 10). Він може працювати як у режимі TCP, так і UDP. Для останнього з'єднання проводити не потрібно. Достатньо розмістити Winsock-елемент на контейнері (наприклад, форму), і можна користуватися його методами, властивостями та подіями.

```

Dim Reg(1 To 10) As Integer      ' тут зберігаємо значення змінних
Private Sub CloseSocket()
    Winsock1.Close              ' закінчити з'єднання
End Sub

Private Sub ConnectSocket()      ' з'єднуємося з MODBUS-сервером
    Winsock1.Protocol = sckTCPProtocol ' вибираємо протокол TCP
    Winsock1.Connect "192.168.9.17", 502 ' вказуємо IP та 502 порт
End Sub

Private Sub CommandButton3_Click()
    ReadRegisters 0, 5 ' читаємо 5 регістрів починаючи з 0
End Sub

Sub ReadRegisters(StartAddr As Integer, CountAddr As Integer)
    Dim a(1 To 12) As Byte ' послідовність байт
    If Winsock1.State = sckConnected Then ' якщо під'єднанні
        a(6) = 6 ' кількість байтів
        a(8) = 3 ' функція
        a(9) = StartAddr \ 256 ' початковий № - старший байт
        a(10) = StartAddr Mod 256 ' початковий № - молодший байт
        a(11) = CountAddr \ 256 ' кількість - старший молодший байт
        Winsock1.SendData a ' відправити масив байтів
    End If
End Sub

Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long) ' отримали дані
    Dim a As Variant, LoInd As Integer, i As Integer, j As Integer
    Winsock1.GetData a, , bytesTotal ' вийняти дані з буфера
    LoInd = LBound(a) ' визначити початковий індекс масиву
    If a(7 + LoInd) = 3 Then ' якщо це функція читання і вона оброблена без помилок
        For i = 1 To a(8 + LoInd) \ 2 ' перебираємо кожний регістр
            j = 9 + (i - 1) * 2 ' номер байта в масиві для регістру i
            If (a(j) And &H80) > 0 Then ' якщо число від'ємне (старший біт в 1)
                a(j) = a(j) And &H7F ' обнулити старший біт
                Reg(i) = a(j) * 256 + a(j + 1) - 32768 ' рахувати з урахуванням знака
            Else
                Reg(i) = a(j) * 256 + a(j + 1) ' рахуємо без знаку
            End If
        Next
    End If
End Sub

```

Рис. 6.35. Фрагмент програми до прикладу 6.8

Приклад 6.8

6.5. Рекомендації до проектування мереж MODBUS RTU/ASCII та MODBUS TCP

Основними перевагами мереж MODBUS перед іншими мережами є їх простота та бюджетність рішення. Як було зазначено в 2-му розділі, нерідко вибір конкретного варіанта диктується наявними обмеженнями з боку вибраної платформи. Якщо вибрані засоби мають інтегровані канали з підтримкою MODBUS, то за необхідності побудови мережних структур є сенс в їх використанні. Тільки якщо дане рішення принципово не підходить з ряду критеріїв, необхідно зміню-

вати конфігурацію обладнання, а інколи і самі програмно-технічні засоби. В даному підрозділі розглянемо загальні підходи до проектування мереж MODBUS у типових ситуаціях, загальна ж послідовність розробки інтегрованих автоматизованих систем наведена в розділі 15.

6.5.1. Послідовність розробки мереж MODBUS RTU/ASCII

Загалом проектування мереж MODBUS RTU можна проводити за алгоритмом, наведеному на рис. 6.36 (для MODBUS ASCII аналогічно). Враховуючи функціонування протоколу прикладного рівня за моделлю Клієнт/Сервер обміну повідомленнями, в блоці 1 необхідно визначити вузли, які повинні ініціювати обмін. Серед типів програмно-технічних засобів, що наведені в розділі 1 даного посібника, типовими ініціаторами обміну можуть бути:

- засоби людино-машинного інтерфейсу (ЛМІ), до яких належать ОП та комп'ютери зі SCADA/HMI;
- контролери, яким необхідно доступатись до інших контролерів або до периферійних засобів;
- програматори або конфігуратори.

Нагадаємо, що засоби ЛМІ обмінюються даними з контролерами, периферійними засобами або іншими засобами ЛМІ. Для відображення плинних даних процесу, засоби ЛМІ опитують їх значення з певною періодичністю тільки тоді, коли сторінка відображення є активною. Така поведінка логічна, оскільки економляться часові ресурси мережі. Однак, деякі засоби ЛМІ дають можливість постійного опитування даних. Зокрема це характерно для операцій ведення архівів, а також для задач тривоги та аварій. Запис даних процесу з ЛМІ проводиться, як правило, тільки за їх зміни. Тому логічним є те, що засоби ЛМІ в мережі повинні бути Клієнтами. Виключенням можуть бути ситуації, коли в мережі повинно бути декілька засобів ЛМІ, або коли необхідно передати термінові повідомлення на термінал. Ці ситуації розглянуті нижче.

За необхідності обміну даними контролера з іншим контролером або з периферійними засобами, він повинен ініціювати обмін, тобто бути Клієнтом.

Програматори та конфігуратори теж повинні бути Клієнтами. У зв'язку з тим, що підключення їх до мережі проводиться тільки за необхідності, та за відсутності вимог до одночасного функціонування мережі разом зі зміною конфігурації/програми мережних вузлів, можна вважати, що програматор не входить до структури мережі. В іншому випадку, необхідно виділяти окремий канал, тобто виділяти окрему мережу для пристроїв, які необхідно програмувати, якщо таке можливе (блок 14).

У будь-якому випадку, кількість клієнтських вузлів не може бути більше одного, оскільки прикладний процес Клієнта в одній мережі MODBUS RTU може міститись тільки на Ведучому. Якщо кількість Клієнтів (блок 1) все ж таки більша одного, можна спробувати перенести клієнтські запити вузлів на один вузол з правами Ведучого (блок 4). Нижче розглянуті декілька прикладів такого переносу. Якщо перенос не вдається з певних причин, необхідно на вузлах виділити окремі канали, тобто замість однієї мережі створити кілька мереж, якщо таке можливо (блоки 14 та 16), або змінити забезпечення вузлів (блок 15).

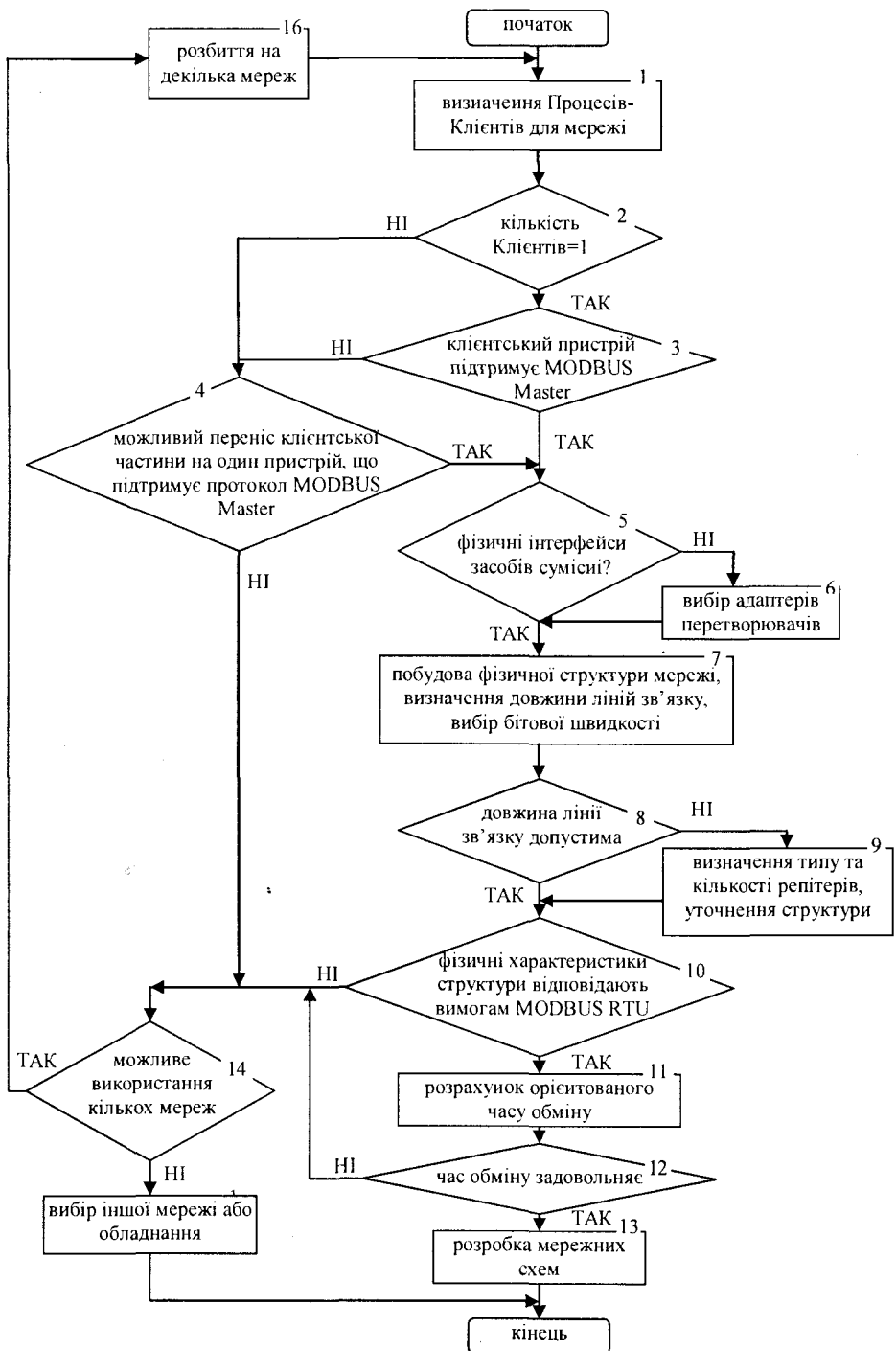


Рис. 6.36. Алгоритм проектування мереж MODBUS RTU

Якщо Клієнт тільки один, його прикладний Процес буде виконуватися на вузлі з правами Ведучого шини (блок 3). Після всіх проведених вище перевірок, необхідно узгодити фізичні інтерфейси (блок 5) та за необхідності використати адаптери перетворювачі. При великих довжинах ліній необхідно підібрати репітер, після чого визначитися з кінцевою структурою, бітовими швидкостями та перевірити відповідність фізичних характеристик вимогам MODBUS RTU.

За наявності часових комунікаційних характеристик для портів усіх засобів, які беруть участь в обміні, бажано визначити орієнтований час обміну, розрахунок якого показаний у прикладі 6.4. Якщо час обміну задовольняє, то мережа підходить для вирішення поставлених завдань і можна переходити до розробки документів робочого проекту.

6.5.2. Перенесення клієнтських запитів у мережах MODBUS RTU/ASCII

Якщо при проектуванні мережі MODBUS RTU виявилось декілька прикладних Процесів Клієнтів, то можна спробувати їх перенести на один вузол.

На рис. 6.37 показаний приклад вирішення завдання, в якому необхідно поєднати в мережу чотири ПЛК (PLC1-PLC4) для обміну даними з певною періодичністю. Розглянемо, який із PLC повинен бути Клієнтом, а отже і Ведучим MODBUS RTU.

Враховуючи, що обмін даними повинен бути періодичним, то кожен з ПЛК для інформаційних потоків може бути як ініціатором обміну, тобто Клієнтом, так і Сервером. Чи можна виділити єдиний Клієнтський вузол? PLC1 не може бути єдиним клієнтським вузлом, бо присутній інформаційний зв'язок між PLC3 та PLC4. PLC2 теж не може через зв'язки PLC3-PLC1 та PLC3-PLC4. Для PLC3 так, як і для PLC4, залишається недосяжним зв'язок PLC1-PLC2. Для PLC4, крім того, не може бути Клієнтом через зв'язок PLC3-PLC1. Таким чином, явно виділяються як мінімум два Клієнти: PLC1 та PLC3.

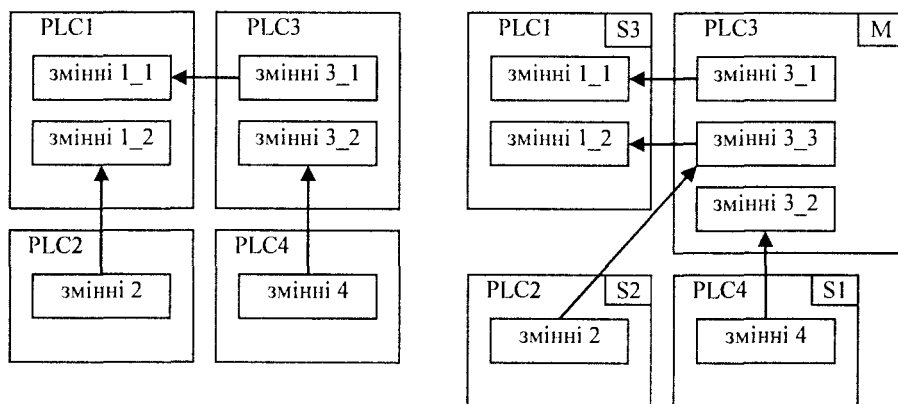


Рис. 6.37. Перенесення клієнтських запитів на один ПЛК:
зліва — постановка завдання, справа — варіант вирішення завдання

Розв'язанням даного завдання може бути перенесення клієнтських запитів на один із вузлів, наприклад PLC3. Прямі клієнтські запити залишаються: PLC3 записує значення змінної 3_1 в змінній 1_1, зчитує значення зі змінних 4 в змінні 3_2. Для реалізації зв'язку між змінною 2 та змінною 1_2 необхідно, щоб PLC3 зчитував ці значення з PLC2, після чого записував їх у PLC1. Як правило, для цього виділяється додатковий буфер, у прикладі це змінні 3_3.

Вибір вузла з клієнтським прикладним Процесом, а отже Ведучого, може проводитись за різними критеріями: здатність комунікаційних каналів працювати в режимі MODBUS RTU Ведучого або Веденого, потужність комунікаційних каналів (продуктивність запитів, що може обробити ПЛК), потужність процесора, зручність програмування або конфігурування, кількість прямих клієнтських запитів, зручність налагоджування тощо. В нашому випадку ми зупинилися на PLC3, що обумовлено зменшенням кількості непрямих клієнтських запитів.

Розглянемо інший випадок. За необхідності підключення до єдиної мережі MODBUS RTU декількох засобів людино-машинного інтерфейсу, які, як ми зазначили, повинні формувати клієнтські запити, виникає необхідність у декількох Ведучих. На рис. 6.38 продемонстрований приклад, в якому необхідно підключити дві операторські панелі до одного ПЛК як для відображення, так і для управління. Жодна з операторських панелей не може взяти на себе ексклюзивні права Клієнта, оскільки інша операторська панель не зможе отримувати дані. Якщо операторські панелі підтримують MODBUS RTU в режимі Веденого, а ПЛК — у режимі Ведучого, то розв'язання цього завдання може бути як на рис. 6.38 (справа). Для цього в операторських панелях повинна бути виділена область пам'яті (на рис. 6.38 змінні 2 та змінні 3), з якою буде обмінюватись ПЛК, генеруючи запити на читання та запис регістрів/бітів. Відображення та зміна значень змінної 1 з операторських панелей буде проходити через ці виділені області.

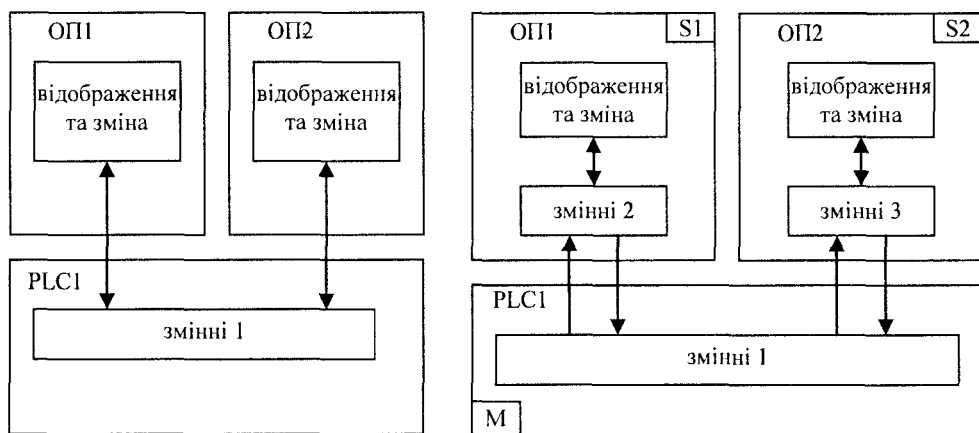


Рис. 6.38. Перенесення клієнтських запитів з декількох ОП на один ПЛК:

зліва — постановка задачі, справа — варіант вирішення завдання

Хоч в розглянутому прикладі використовуються операторські панелі, аналогічно вирішується ситуація, коли в якості засобів людино-машинного інтерфейсу виступають ПК зі SCADA, або комбінація ОП та SCADA. Для SCADA в цьому випадку необхідна наявність драйверів MODBUS RTU в режимі Веденого.

В даній задачі можливе також перенесення клієнтських запитів на один із засобів людино-машинного інтерфейсу. Однак, це далеко не завжди вдається, оскільки в цьому випадку необхідно реалізувати переприсвоєння змінних з різними джерелами даних. Крім того, засоби ІМІ здебільшого не можуть забезпечити детермінований час доставки даних.

6.5.3. Послідовність розробки мереж MODBUS TCP/IP

На відміну від MODBUS RTU/ASCII, в MODBUS TCP/IP, що, як правило, базується на Ethernet, немає проблем з наявністю клієнтських запитів тільки на одному вузлі. Прикладний Процес на вузлі може одночасно бути як Клієнтом, так і Сервером. При розробці мережі враховують фізичні особливості засобів Ethernet (див. розділ 10). Таким чином, у загальній послідовності проекту мережі можна виділити такі кроки:

1. Визначення клієнтських вузлів.
2. Визначення підтримки потенційними клієнтськими вузлами режиму MODBUS Клієнта, а серверних — режиму MODBUS Серверу.
3. Перенесення клієнтських або серверних запитів на інші вузли за необхідності.
4. Розрахунок орієнтовного часу транзакції між вузлами Клієнта та Серверу.
5. Розробка мережних схем на базі Ethernet.

Розрахунок орієнтованого часу транзакції для MODBUS TCP/IP базується на тих саме принципах що і MODBUS RTU/ASCII. Однак замість часу, який витрачається на передачу кадру, необхідно врахувати час реакції комунікаційних засобів Ethernet (концентраторів, комутаторів, мостів, шлюзів і т.д.).



Контрольні запитання до розділу 6

1. Якими мережами сьогодні представлений MODBUS? Охарактеризуйте їх у контексті моделі OSI.
2. Розкажіть про основи функціонування MODBUS Application Protocol. Який формат повідомлення MODBUS PDU? Як поділяються коди функцій з точки зору MODBUS-IDA?
3. Як MODBUS-серверний Процес дізнається про помилковий результат обробки повідомлення-запиту?
4. Які функції використовуються для доступу до даних процесу?
5. Як формуються повідомлення-запити та повідомлення-відповіді для читання та запису діапазону вхідних та вихідних регістрів?
6. Які ситуації можливі при обробці запиту MODBUS Клієнта? Наведіть приклади відповідей про помилку.

7. Що таке MODBUS відображення даних? Які моделі відображення даних MODBUS використовуються в сучасних засобах автоматизації?
8. Прокоментуйте основи функціонування MODBUS RTU/ASCII в контексті моделі OSI. Як пов'язана модель функціонування обміну на прикладному рівні з функціонуванням на каналному?
9. Яка модель адресації та який метод доступу до шини використовується в MODBUS RTU/ASCII на каналному рівні? Як на цьому рівні контролюється правильність доставки бітової послідовності?
10. В чому відмінність роботи MODBUS RTU та MODBUS ASCII на каналному та фізичному рівнях?
11. Розкажіть про принципи розрахунку часу обміну між вузлами з використанням часової діаграми.
12. Розкажіть про принципи побудови кадрів для MODBUS RTU та MODBUS ASCII.
13. Які інтерфейси, бітова швидкість та топологія використовуються для MODBUS RTU/ASCII на фізичному рівні? Яким чином вузли можуть підключатися до загальної шини?
14. Як реалізується двохпроводне з'єднання вузлів по MODBUS RTU/ASCII? В якому місті і як реалізується захисне зміщення?
15. Які типи роз'ємів визначені в стандартах MODBUS-IDA для підключення пристроїв до шини по RS-485?
16. Які основні недоліки Ви можете назвати в принципах функціонування мереж MODBUS RTU/ASCII?
17. Прокоментуйте основи функціонування MODBUS TCP/IP у контексті моделі OSI. Які типи пристроїв передбачає комунікаційна архітектура MODBUS TCP/IP?
18. Які поля включає формат модуля даних прикладного рівня MODBUS TCP/IP? Поясніть призначення полів заголовка MBAP?
19. Яка адреса порту TCP використовується для MODBUS Серверу?
20. Яким чином забезпечується доступ до необхідного вузла MODBUS RTU/ASCII, підключених через шлюз до MODBUS TCP/IP у заголовку MBAP?
21. Які вузли, як правило, є ініціаторами обміну в MODBUS RTU/ASCII? Поясніть чому?
22. Навіщо необхідно визначати вузли з клієнтськими запитами? Чому вузол з клієнтським Процесом у MODBUS RTU/ASCII може бути тільки один?
23. Наведіть приклади переносу клієнтських запитів на один вузол. Які обмеження при цьому є?
24. В чому відмінність проектування MODBUS TCP/IP порівняно з MODBUS RTU/ASCII?

МЕРЕЖИ PROFIBUS

PROFIBUS (PROcess Field Bus) — це відкрита промислова комунікаційна система, яка призначена для використання в системах автоматизації технологічних процесів та виробництв і розроблена спеціально для швидкодіючих та складних задач зв'язку.

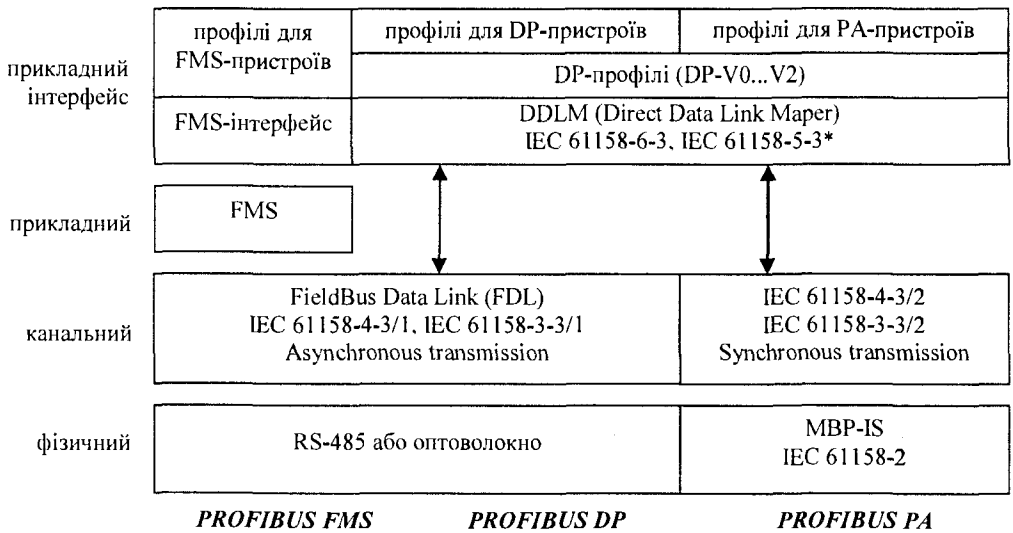
Робота над PROFIBUS була започаткована в 1987 році рядом німецьких компаній та інститутів. Метою було створення високошвидкісної промислової мережі для німецької промисловості. Першою була розроблена специфікація PROFIBUS FMS, а в 1993 році з'явилася специфікація PROFIBUS DP. Ці специфікації закріпилися в національному німецькому стандарті DIN 19245 відповідно в 1991 та 1993 р. В 1996 році вони увійшли до загальноєвропейського стандарту EN 50170. Нині мережі PROFIBUS також закріплені у міжнародних стандартах IEC 61158 та IEC 61784, а аспекти їх застосування і технічні характеристики визначені в загальнодоступних документах некомерційної німецької організації **PNO** (PROFIBUS Nutzerorganisation, заснованої у 1989 р). Крім того, PROFIBUS підтримується міжнародною всесвітньою організацією **PI** (PROFIBUS International, 1995 р.), що гарантує відкритість мережі і зв'язок між пристроями різних виробників.

7.1. Мережі PROFIBUS у контексті моделі ISO та MEK

Сьогодні PROFIBUS — це система, яка включає в себе різноманітні технології (рис. 7.1):

- **PROFIBUS DP** (Decentralized Periphery);
- **PROFIBUS FMS** (Fieldbus Message Specification);
- **PROFIBUS PA** (for Process Automation).

Мережа PROFIBUS DP використовується в системах автоматизації процесу для зв'язку контролерів з розподіленою периферією, а PROFIBUS PA — для зв'язку контролерів з датчиками в промисловій зоні.



* в IEC описані на прикладному рівні

Рис. 7.1. PROFIBUS у контексті OSI-моделі

PROFIBUS DP — це символноорієнтований асинхронний тип мережі, який на фізичному рівні базується на інтерфейсі RS-485 або на різних типах оптоволоконних з'єднань. Відсутність протоколу прикладного рівня (згідно з описом EN) дозволяє безпосередньо обмінюватися ідентифікованими даними між вузлами, що значно прискорює процес обміну. Кількість та призначення даних визначається профілем та конфігурацією мережі. Слід зазначити, що в трьохрівневій моделі MEK, PROFIBUS DP на прикладному рівні описаний: протоколом — IEC 61158-5-3 та інтерфейсом рівня — IEC 61158-6-3. У розділі 7.7 наведена інша модель PROFIBUS DP, яка викладена в специфікації від PNO.

PROFIBUS PA — перша із PROFIBUS комунікацій, яка з'явилася в стандарті IEC 61158. Мережа реалізує синхронний тип обміну, що більш характерний для обміну даними вводу/виводу в жорсткому реальному часі. На фізичному рівні використовується інтерфейс MBP, який дозволяє жити по інформаційній парі проводів підключені датчики та виконавчі механізми. Це спеціальна пропозиція для вибухонебезпечних зон виробництва. Сегмент PROFIBUS PA може підключатися до сегмента PROFIBUS DP і таким чином включатися у загальну PROFIBUS-систему.

PROFIBUS FMS — це, по суті, сервіс прикладного рівня, який забезпечує ациклічний обмін повідомленнями, що дозволяє передавати дані великого обсягу. Враховуючи, що він базується на тих же сервісах канального рівня, що і PROFIBUS DP, вони можуть працювати в одній і тій же мережі.

У мережах PROFIBUS, незалежно від їх типу, використовується гібридний метод доступу до шини (рис. 7.2), що дозволяє обмінюватися даними між рівноправними вузлами (активні учасники — Ведучі), а також забезпечувати їх

обмін з розподіленими периферійними засобами на шині (пасивні учасники — Ведені).

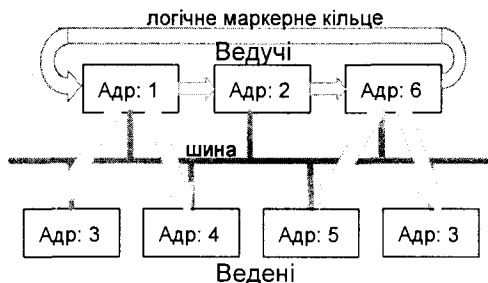


Рис. 7.2. Метод доступу в мережах Profibus

7.2. Реалізація фізичного рівня PROFIBUS DP

PROFIBUS DP на фізичному рівні базується на інтерфейсі RS-485 або на різних типах оптоволоконних з'єднань з бітовою швидкістю до 12 МБіт/с. Дані передаються 11-бітними символами (1 старт, 8 даних, 1 паритет, 1 стоп) методом NRZ. Максимальна кількість вузлів на сегмент — 32, максимальна довжина лінії залежить від бітової швидкості:

- від 9600біт/с до 187500біт/с — до 1000 м;
- 500 кбіт/с — до 500 м;
- 1500 кбіт/с — до 200 м;
- 12 Мбіт/с — до 100 м.

Використовується шинна топологія (рис. 7.3). З'єднання вузлів проводиться за 2-провідною схемою (сигнал А та В) без сигнальної землі. На кінцях кожного сегмента виставляються термінатори лінії опором 220 Ом.

У стані спокою, коли всі передавачі мовчать, на лінії обов'язково повинна бути логічна «1». Тобто, в мережі необхідно реалізувати поляризацію лінії (захисне зміщення). Для цього на кінцях лінії, крім термінаторів лінії, до шини підключається живлення 5 В через PullUp та PullDown — резистори номіналом 390 Ом. Захисне зміщення з обох кінців забезпечується, як правило, безпосередньо кінцевими вузлами та з'єднувальними аксесуарами. Використання зміщення з обох кінців більш надійне порівняно з централізованою схемою, як, наприклад, у Modbus RTU, оскільки при відключенні одного з кінцевих вузлів мережа залишається робочою.

Підключення пристроїв до загальної шини проводиться без відгалужень. Ці вимоги пов'язані з ефектом відбиття хвиль на коротких ділянках відгалуження. На рис. 7.4 видно, що кабель шини повинен «зайти» до кожного і «вийти» з кожного пристрою. На цьому ж рисунку показаний спосіб розділення загальної шини на сегменти, які об'єднуються за допомогою репітерів. Таким чином, можна досягнути іншої топології і збільшити кількість пристроїв та загальну довжину мережі. Необхідно пам'ятати, що наявність репітерів треба враховувати при виборі швидкості передачі даних.

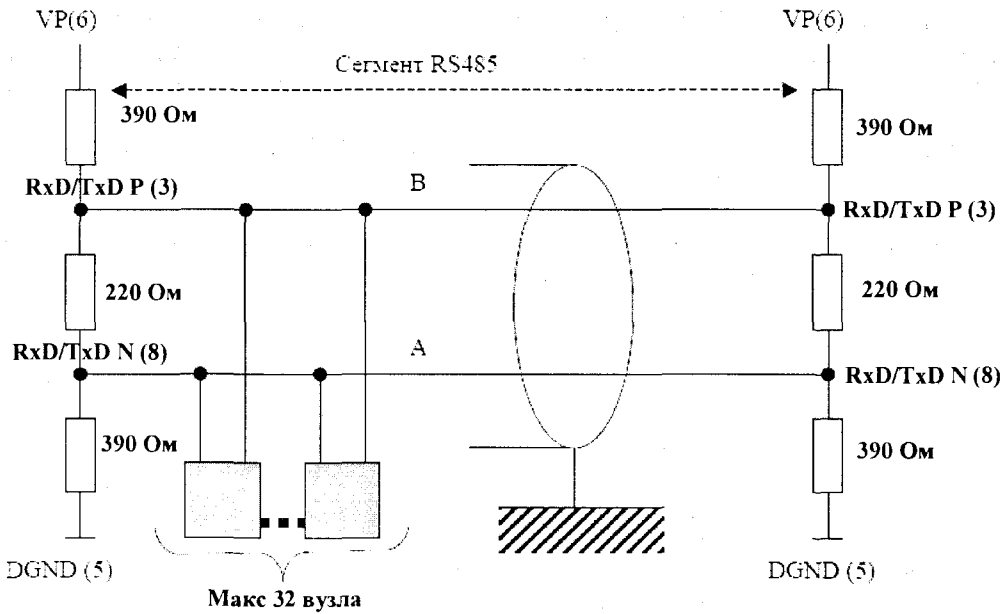


Рис. 7.3. Структура шинного сегмента RS-485

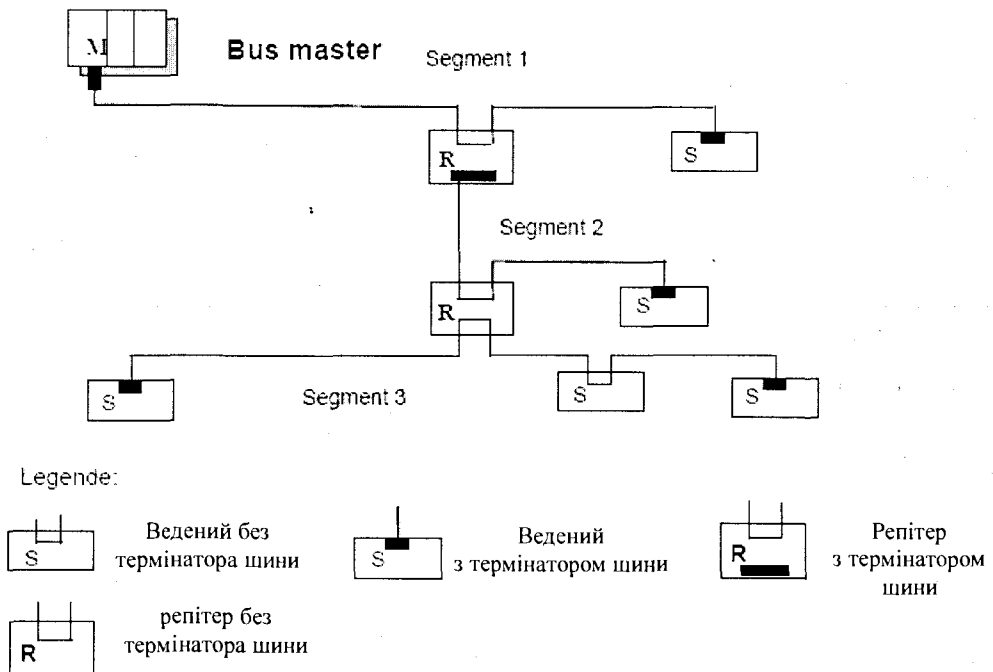
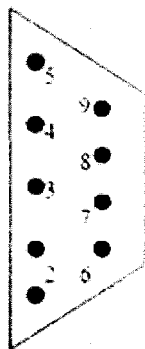


Рис. 7.4. Сегментація та підключення пристроїв до PROFIBUS-шини

Деякі пристрої необхідно підключати тільки в певні періоди часу, наприклад, програматори. В цьому випадку дозволяється підключення до шини через активні відгалужувачі (наприклад, шинні термінали, репітери, активні кабельні з'єднувачі).

В якості з'єднувача в стандартах IEC 61158-2 і EN 50170-2 рекомендується використовувати 9-піновий SUB-D конектор, призначення контактів якого наведено на рис. 7.5. Наявність наведених функцій для контактів 3,5,6 та 8 є обов'язковою. Стандарти дозволяють використовувати інший тип з'єднувачів або клемну колодку з аналогічними контактами.



№ контакта	Позначення	Сигнал
1	—	не використовується
2	M24	— 24 В
3	RxD/TxD-P	Прийом/передача, В+
4	CNTR-P	Сигнал для управління передачею, +
5	DGND	Сигнальна земля, —
6	VP	Напруга живлення 5В, +
7	P24	+ 24 В
8	RxD/TxD-N	Прийом/передача, А—
9	CNTR-N	Сигнал для управління передачею, —

Рис. 7.5. Призначення контактів 9-пінового SUB-D конектора

Стандарт IEC 61158-2 визначає два типи кабелів рекомендованих для підключення PROFIBUS шини: виті пари типу А та В. Характеристики цих кабелів і припустимі швидкості передачі даних для них, залежно від довжини лінії зв'язку, наведені у табл. 7.1 і 7.2.

Таблиця 7.1

ХАРАКТЕРИСТИКИ КАБЕЛІВ

Параметр кабелю	Тип А	Тип В
Характеристичний імпеданс	135–165 Ом (при частотах від 3 до 20 МГц)	100–130 Ом (при частотах більше 100 кГц)
Ємність	<30 пФ.м	<60 пФ.м
Погонний опір	<110 Ом/км	не визначено
Переріз проводу	$\geq 0.34 \text{ мм}^2$	$\geq 0.22 \text{ мм}^2$
Колір ізоляції для не IS застосування	фіолетовий	не визначено
Колір ізоляції для IS застосування	синій	не визначено
Колір проводу А (RxD/TxD-N)	зелений	не визначено
Колір проводу В (RxD/TxD-P)	червоний	не визначено

Для шинної топології на базі RS-485 інтерфейсу для достатнього погашення відбиття сигналів рекомендують мінімізувати довжину відгалужень від магістральної шини. Загальна ємність усіх відгалужень на шині не повинна перевищувати: 0,05 nF для 3 Мбіт/с, 6 Мбіт/с та 12 Мбіт/с; 0,2 nF для 1,5 Мбіт/с; 0,6 nF для 500 кбіт/с; 1,5 nF для 187,5 кбіт/с; 3,0 nF для 93,75 кбіт/с; 15 nF для 9,6 та 19,2 кбіт/с.

Таблиця 7.2

ЗАЛЕЖНІСТЬ МАКСИМАЛЬНОЇ БІТОВОЇ ШВИДКОСТІ ВІД ДОВЖИНИ ЛІНІЇ ЗВ'ЯЗКУ

Бітова швидкість (кбіт/с)	9,6	19,2	93,75	187,5	500	1500	3000	6000	12000
Макс. довжина кабелю типу А (м)	1200	1200	1200	1000	400	200	100	100	100
Макс. довжина кабелю типу В (м)	1200	1200	1200	600	200	70	не допустима		

Таким чином, для бітової швидкості порядку 1500 Кбіт/с сумарна довжина всіх відгалужень на кабелі типу А не повинна перевищувати 6,6 м, а для швидкості 3/6/12 Мбіт/с — 1,6 м. При високих швидкостях (3—12 Мбіт/с) необхідно включити додатковий імпеданс на лінії зв'язку відгалужувачів або в шинні з'єднувачі. Стандарт рекомендує використати для цього додаткові індуктивності (рис. 7.6).

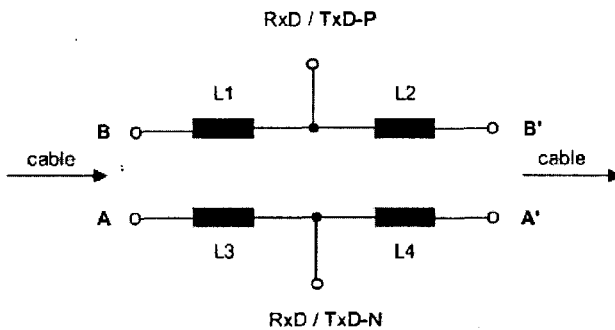


Рис. 7.6. Використання індуктивностей в якості додаткових імпедансів

Для кабелю типу А індуктивності L1-L4 мають мати значення від 110 ± 22 нГн. Значення індуктивностей розраховуються відповідно до імпеданса між B та B', A та A', ємнісних характеристик з'єднувача, трансивера та інших деталей. Тобто, при відключенні пристрою від шини (з'єднувача від пристрою), в даній точці може відбутись відбиття сигналу, що спотворює корисний сигнал на загальній шині. На рис. 7.7 показана конструкція мережного з'єднувача при підключенні без відгалужень.

Слід зазначити, що при використанні кабелю типу В термінатор лінії повинен мати опір 150 Ом замість 220 Ом (для кабелю типу А).

При прокладці лінії передачі у місцях з потужними електромагнітними завадами необхідно використати екранований кабель та правильно виконати заземлення екрана. Екран кабелю повинен бути з'єднаний з корпусом штекерів (рис. 7.7).

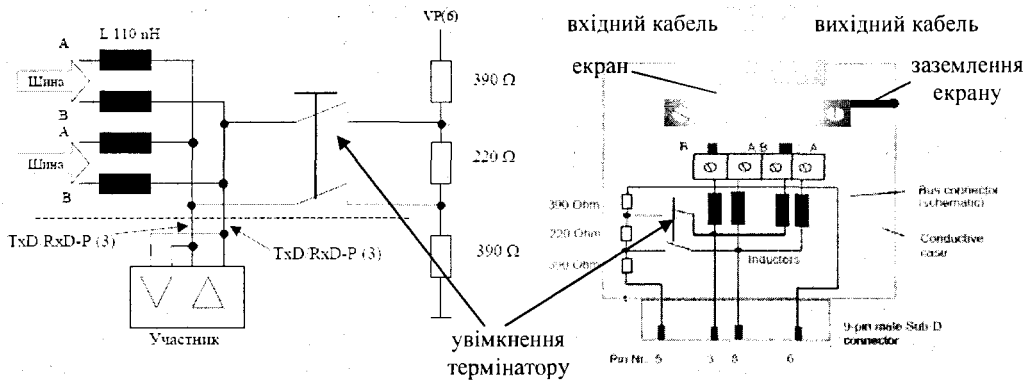


Рис. 7.7. Схема мережного з'єднувача для мереж з бітовою швидкістю >1500 біт/с

Згідно зі стандартами PROFIBUS заземлення екрана необхідно проводити в усіх точках підключення кабелю через відповідне виконання роз'ємів (рис. 7.7).

Приклад 7.1. PROFIBUS. Побудова схеми мережних з'єднань для PROFIBUS DP на базі RS-485

Завдання. Підібрати необхідні технічні засоби та розробити схему мережних з'єднань для PROFIBUS DP відповідно до наведеної структури рис. 7.8.

PLC1 (Master)

VIPA SPEED7



PLC2

VIPA 200



PDS1

LENZE 8200



RIO1

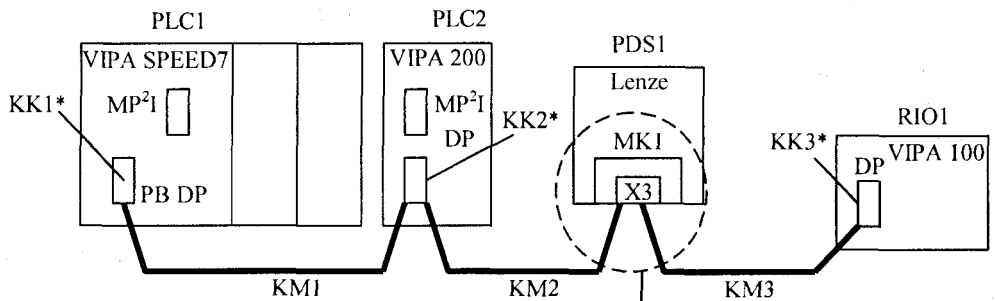
VIPA 100



PROFIBUS DP RS-485

Рис. 7.8. Структурна схема системи до прикладу 7.1

Рішення. PLC1 — Speed7 має вбудований PROFIBUS DP інтерфейс, який можна використати в якості Ведучого. PLC2 вибираємо з таким розрахунком, щоб у процесорному модулі теж був інтегрований інтерфейс ROFIBUS-DP (VIPA 215-2BP02), що значно дешевше, ніж купувати окремий модуль. Модулі розподіленого вводу виводу RIO1 повинні бути PROFIBUS DP Веденими (VIPA 153-4PH00). Для підключення до мережі частотного перетворювача Lenze необхідний відповідний комунікаційний модуль E82ZAFPC001. Таким чином, схема мережних з'єднань буде мати вигляд, показаний на рис. 7.9. Специфікація мережних засобів та вузлів дана в таблиці 7.3. На рис. 7.10 наводиться схема підключень кабелів KM1-KM3 до мережних з'єднувачів KK1-KK3.



* — схема підключення дана в текстовій частині

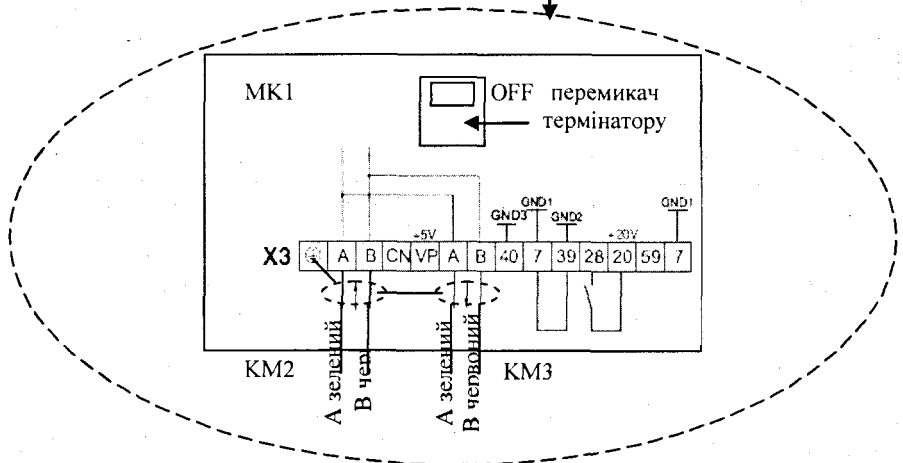


Рис. 7.9. Схема мережних з'єднань до прикладу 7.1

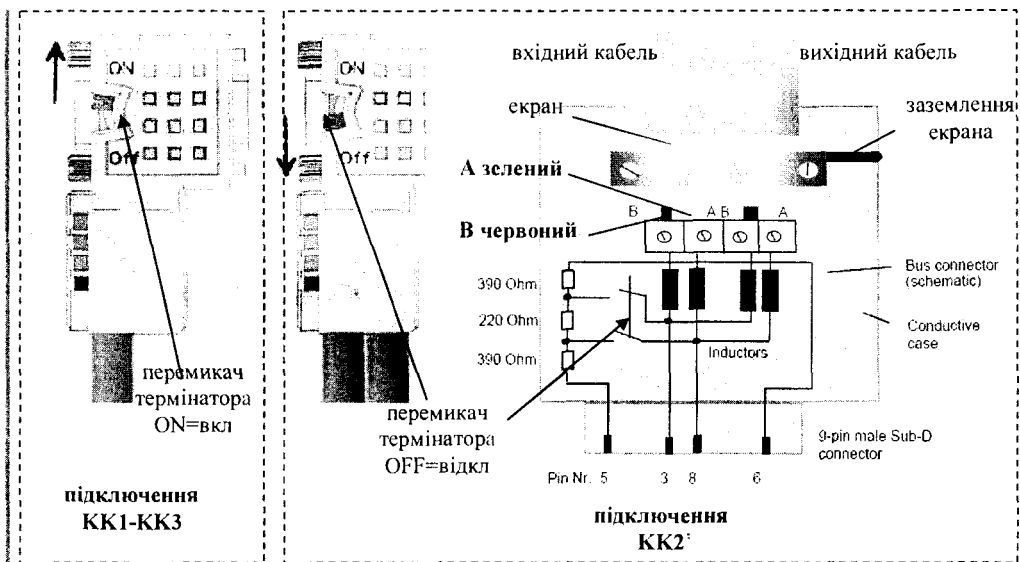


Рис. 7.10. Схема підключення кабелів до мережних з'єднувачів (до прикладу 7.1)

Таблиця 7.3

СПЕЦИФІКАЦІЯ МЕРЕЖНИХ ЗАСОБІВ ТА ВУЗЛІВ

№	Позначення	Назва	Тип	Кількість	Фірма
1	PLC1	ПЛК VIPA Speed7	VIPA 314-6CF02	1 шт.	VIPA
2	PLC2	ПЛК VIPA 200	VIPA 215-2BP02	1 шт.	VIPA
3	RIO1	Модуль віддаленого вводу/виводу VIPA 100	VIPA 153-4PH00	1 шт.	VIPA
4	PDS1	Частотний перетворювач Lenze 8200 Vector для управління асинхронним двигуном 0,37 кВт	E82EV371K2C	1 шт.	LENZE
5	MK1	комунікаційний модуль для роботи в мережі PROFIBUS DP частотних перетворювачів Lenze 8200 з убудованим термінатором	E82ZAFPC001	1 шт.	LENZE
6	КК1-КК3	мережний з'єднувач Easy Conn PB 90° для мережі PROFIBUS DP з убудованим термінатором	VIPA 972-0DP10	3 шт.	VIPA
7	КМ1	кабель екранована вита пара для PROFIBUS DP (тип «Стандарт»)	VIPA 830-0LE00	30 м	VIPA
8	КМ2	кабель екранована вита пара для PROFIBUS DP (тип «Стандарт»)	VIPA 830-0LE00	40 м	VIPA
9	КМ3	кабель екранована вита пара для PROFIBUS DP (тип «Стандарт»)	VIPA 830-0LE00	15 м	VIPA

Приклад 7.1

7.3. Реалізація фізичного рівня PROFIBUS PA (MBP)

PROFIBUS PA на фізичному та каналному рівнях базується на технології передачі **MBP** (Manchester Coding (M) and Bus Powered (BP)), яка застосовується у функціонально небезпечних галузях автоматизації, що дозволяє використовувати цю шину в потенційно вибухонебезпечних процесах. Як видно за самою назвою, для передачі використовується манчестерське кодування з можливістю живлення пристроїв по одній 2-х провідній шині. В стандарті IEC 61158-2 PROFIBUS PA визначена як Тип 3 в синхронному режимі з бітовою швидкістю 31,25 Кбіт/с.

Безпечна технологія передачі MBP, як правило, обмежується певними сегментами заводу (польові пристрої в небезпечних Ex-зонах), які з'єднуються із сегментами RS-485 (безпечна зона, система управління та інженерні пристрої в пункті управління) за допомогою сегментних відгалужувачів (segment coupler) та сегментних з'єднувачів (segment link). З'єднання сегментів PROFIBUS DP та PROFIBUS PA показані на рис. 7.11.

Сегментні відгалужувачі (segment coupler) — це перетворювачі сигналів, які модулюють сигнал з RS-485 в MBP та навпаки. Пристрої на шині PROFIBUS PA прозорі з точки зору загальної шини, тобто Ведучі обмінюються з ними як з окремими Веденими. Швидкість на сегменті PROFIBUS DP в цьому випадку повинна бути 45.45 Кбіт/с.

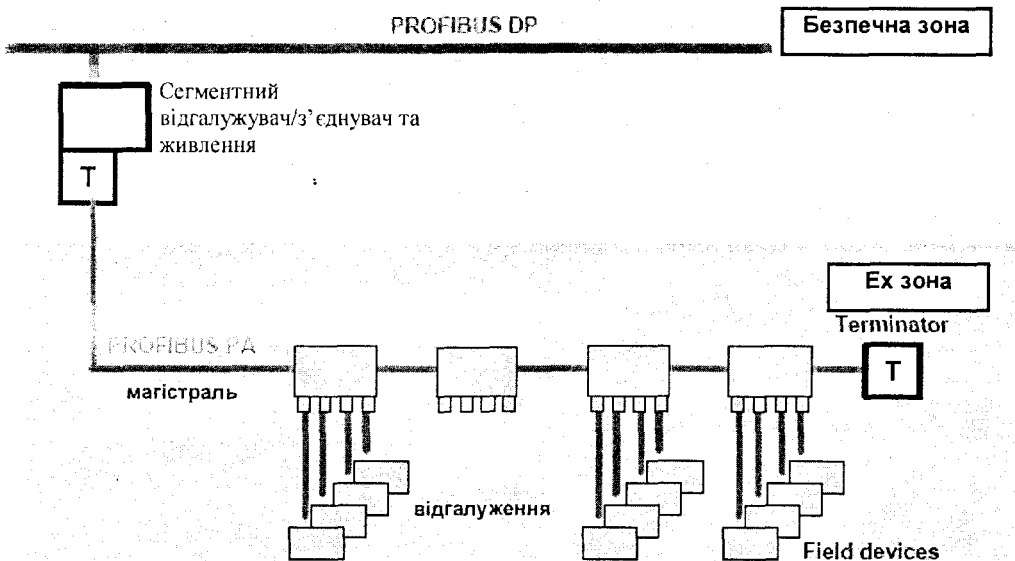


Рис. 7.11. З'єднання сегментів PROFIBUS DP та PROFIBUS PA

Сегментні з'єднувачі (segment link) — це інтелектуальні пристрої, які є Ведучими пристроями на сегменті PROFIBUS PA і обмінюються даними з Веденими сегмента. З боку PROFIBUS DP сегментні з'єднувачі є Веденими, які мають одну адресу, а вузли на PROFIBUS PA видимі як модулі вводу/виводу. При використанні сегментних з'єднувачів обмеження на швидкість з боку PROFIBUS DP відсутня.

PROFIBUS PA підтримує шинну та деревовидну топології. В якості магістральних кабелів та відгалужень рекомендується використовувати кабелі типу *A*, *B*, *C* та *D* (див. табл. 7.4).

Таблиця 7.4

ХАРАКТЕРИСТИКА КАБЕЛІВ

	Тип А (основний)	Тип В	Тип С	Тип D
Структура кабелю	Витий, двожильний, екранований	Одна або декілька витих пар, екрановані	Декілька витих пар, неекрановані	Декілька невитих пар, неекрановані
Площа перерізу (номінальна)	0,8 мм ² (AWG18)	0,32 мм ² (AWG22)	0,13 мм ² (AWG26)	1,26 мм ² (AWG16)
Погонний опір (пост. струм)	44 Ω/км	112 Ω/км	264 Ω/км	40 Ω/км
Хвильовий опір при 31,25 кГц	100 Ω±20%	100 Ω±30%	***	***
Затухання при 39 кГц	3 dB/km	5 dB/km	8 dB/km	8 dB/km
Смнісна неузгодженість	2 nF/km	2 nF/km	***	***
Груповий час запізнювання (7,9...39 кГц)	1,7 μs/km	***	***	***
Ступінь екранування	90 %	***	—	—
Рекомендована довжина мережі включно з ¼-хвильові узгоджуючі шлейфи	1900 m	1200 m	400 m	200 m

Максимальна кількість пристроїв у сегменті обмежена 32-ма вузлами. Загальна довжина лінії залежить від топології, довжини відгалужень, кабелю. Наприклад, при кабелі типу *A* — не більше 1900 м. Довжина відгалужень не повинна перевищувати 30 м. Сегмент з обох кінців обмежений термінатором, один з яких убудований у сегментний відгалужувач/з'єднувач, а інший підключається у кінці сегмента (рис. 7.12).

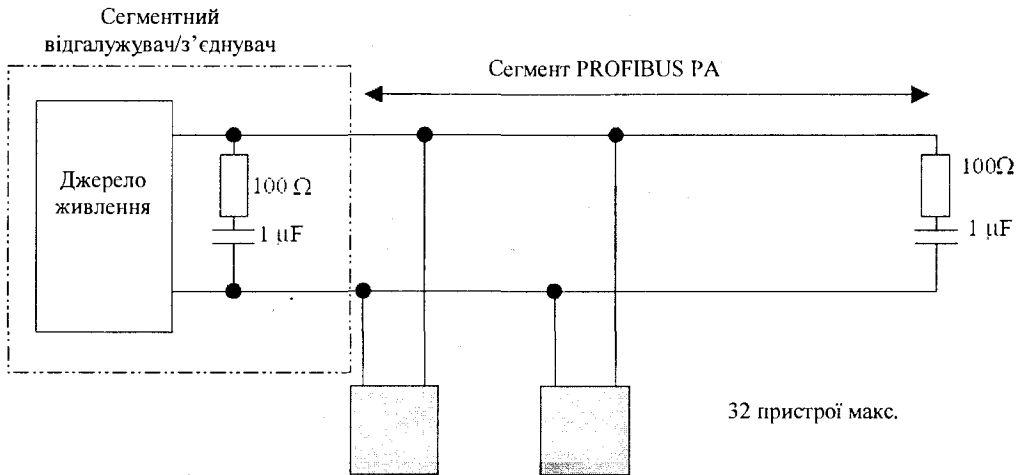


Рис. 7.12. Принципова схема сегмента PROFIBUS PA

Фізичне підключення кабелів відгалуження проводиться з використанням спеціальних з'єднувачів, які мають високий рівень захисту за IP (рис. 7.13).

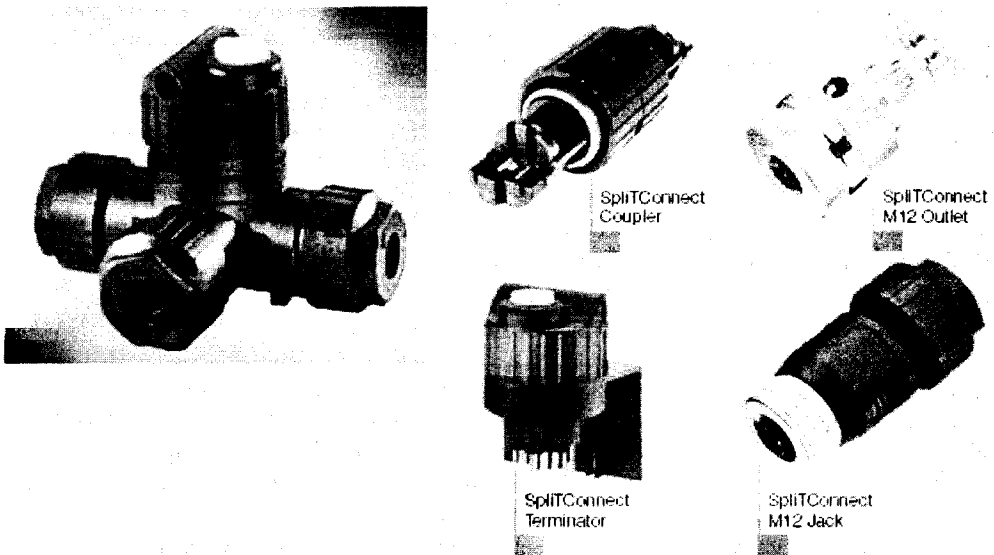


Рис. 7.13. Відгалужувачі, з'єднувачі, термінатори SplitConnect (Siemens) для PROFIBUS PA

У PROFIBUS PA для передачі даних використовується біт-синхронізований з манчестерським кодом протокол передачі (позначається також H1). Дані передаються за допомогою модуляції $\pm 9\text{mA}$ несучого струму шини I_b (рис. 7.14).

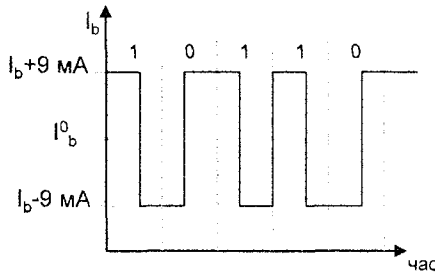


Рис. 7.14. Передача даних у PROFIBUS PA за допомогою модуляції струму

7.4. Реалізація каналного рівня

У мережах PROFIBUS використовується гібридний метод доступу (див. рис. 7.2). Кожний вузол (Ведучий або Ведений) має унікальну адресу від 0 до 126. Вузол може передати кадри одному вузлу (точка-точка), використовуючи його адресу, або всім вузлам (широкомовний режим) використовуючи широкомовну адресу = 127.

Кадри PROFIBUS DP передаються 11-бітними символами, а в PROFIBUS PA — побітово синхронним способом. Однак, структура кадрів має однаковий формат для обох типів мережі. На каналному рівні відомі такі типи сервісів: SRD — дані відправляються і приймаються з підтвердженням; SDN — дані відправляються без підтвердження; SDA (тільки для FMS) — відправка даних з підтвердженням; CSRD (тільки для FMS) — циклічна посилка та прийом з підтвердженням. Ці сервіси використовують різні формати кадрів як з постійною, так зі змінною довжиною (рис. 7.15).

Всі кадри, окрім кадру короткого підтвердження, починаються з стартового байта, який вказує на тип формату кадру. Далі вказується адреса отримувача/чів (0-247) та адреса відправника (0-246). Контрольний байт у кадрі уточнює призначення кадру. Контрольна сума вираховується логічним об'єднанням значення всіх байтів по AND без переповнення. Закінчуються кадри кінцевим байтом (16₁₆).

За допомогою кадру-маркера активний вузол (Ведучий) передає маркер наступному Ведучому вузлу (рис. 7.2). Отримавши маркер, Ведучий обмінюється даними з Веденими, після чого передає маркер далі. Так відбувається до тих пір, поки маркер не надходить до активного вузла з найбільшою адресою (HSA—Highest Station Address). Цей вузол передає маркер активному вузлу з найменшою адресою, тим самим замикаючи маркерне кільце.

Управління доступом до шини активних вузлів відбувається як на фазі ініціалізації, так і на фазі функціонування маркерного кільця. При цьому встановлюється адреса всіх активних вузлів, які присутні на шині і заносяться до списку активних станцій *LAS* (List of Active Station). Для управління маркером при цьому особливо важливі адреси попередньої станції, від якої маркер отримується, і наступної станції, якій маркер передається. Крім того, *LAS* потрібен для виключення з кільця дефектних активних вузлів та включення в кільце вузлів, які знову з'явилися, без переривання функціонування мережі.

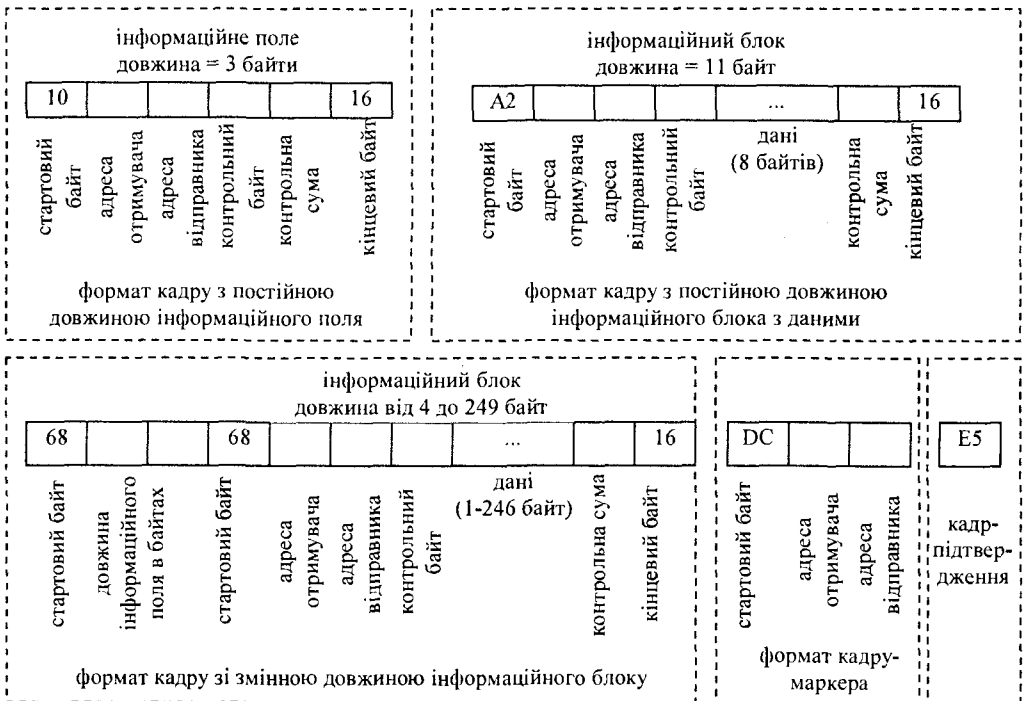


Рис. 7.15. Формат кадрів для PROFIBUS

7.5. Базові функції обміну даними між вузлами в PROFIBUS DP (DP-V0)

У PROFIBUS DP доступні різні типи сервісів: циклічний/ациклічний обмін даними процесу, ациклічний обмін параметричними даними, функції діагностування та управління вузлом.

Для швидкого циклічного обміну даними процесу в PROFIBUS DP використовується постійне опитування центральним вузлом (контролером) розподіленої периферії. Цей механізм може бути описаний клієнт-серверною моделлю ідентифікованого обміну (модель Polling), а в PROFIBUS DP він називається **циклічним обміном** (Cyclic Data Exchange), тому що обмін Ведучого з Веденими відбувається за циклом. Функції циклічного обміну забезпечується сервісами Data_Exchange (див. 7.5.2). Вони з'явилися першими і входять до складу базових функцій PROFIBUS DP. До базових функцій також належать усі інші функції, перелічені вище, крім ациклічного обміну даними процесу, яка та ряд інших з'явилися в старших версіях протоколу PROFIBUS. Нині існує три версії PROFIBUS DP:

- PROFIBUS **DP-V0** з базовими функціями;
- PROFIBUS **DP-V1** та PROFIBUS **DP-V2** з додатковими функціями.

Оскільки функції молодших версій реалізовані в протоколах старшої версії також, зберігається їх сумісність. У даному підрозділі розглянуті базові функції

PROFIBUS DP, тобто функції PROFIBUS DP-V0, а в наступному — додаткові функції PROFIBUS DP-V1 та PROFIBUS DP-V2.

7.5.1. Функція циклічного обміну даними процесу

7.5.1.1. Циклічний обмін даними процесу. Циклічний обмін даними процесу продемонструємо на типовій конфігурації мережі PROFIBUS DP, тобто структурі з одним Ведучим (рис. 7.16).

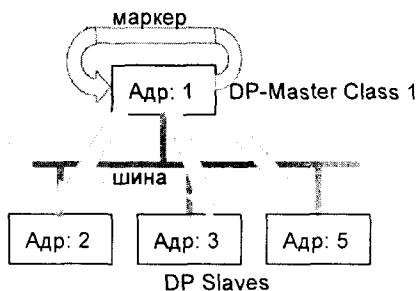


Рис. 7.16. Структура з одним Ведучим

Кожний Ведений може зайняти шину для передачі кадру тільки з дозволу Ведучого. Перелік адрес Ведених зберігається в списку опитування Ведучого (Polling-Liste). **DP-циклом** називається процес опитування Ведучим усіх своїх Ведених за час від отримання маркера до передачі його наступному Ведучому. Структура DP-циклу складається з циклічної частини, за якої Ведучий обмінюється з веденими даними процесу, та ациклічної, за якої Ведучий може обмінюватися параметричними даними (див. 7.5.3).

Обмін даними процесу між Ведучим і Веденим проходить без аналізу змісту цих даних з використанням сервісу SRD. Ведучий відправляє запит типу **Data_Exchange** в якому передає Веденому значення вихідних даних процесу і отримує від нього кадр-відповідь (підтвердження) з вхідними даними процесу (рис. 7.17). Це ідентифікований тип обміну, тому кількість і призначення цих даних визначається в момент конфігурації мережі, завдяки чому забезпечується максимальна швидкодія.

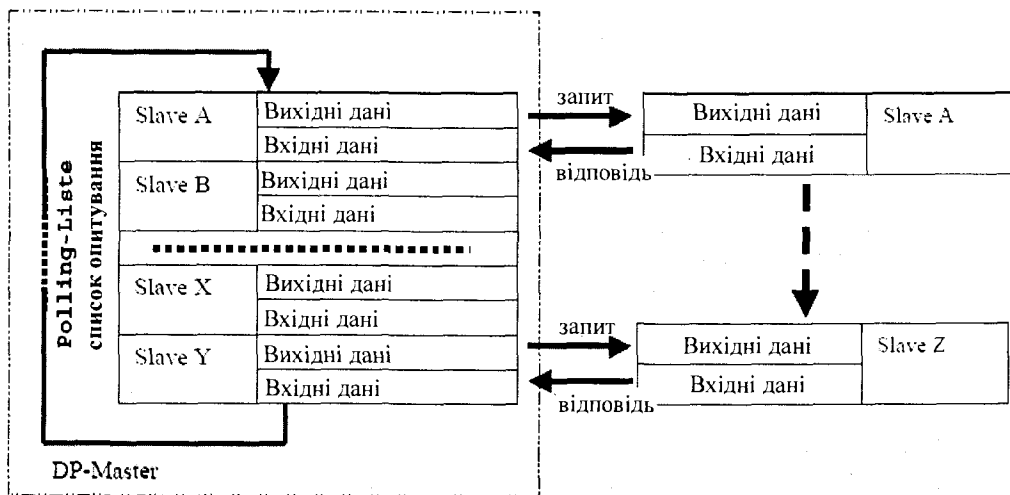


Рис. 7.17. Опитування Ведених за списком опитування

При циклічному обміні даними процесу, Ведений реагує тільки на ті запити `Data_Exchange`, які надходять від Ведучого, який його конфігурував та параметризував (див 7.5.3).

Повідомлення, які надсилає Ведучий, мають «високий» пріоритет, а які видає Ведений — «низький». Однак Ведений може під час відповіді змінити клас пріоритету повідомлення на «високий», якщо необхідно повідомити про наявність діагностичної події. Після отримання такої відповіді Ведучий додатково опитує веденого на предмет уточнення діагностичної інформації повідомленням типу `SlaveDiag`. Процес обміну діагностичною інформацією проходить ациклічним способом у період ациклічної частини ROFIBUS DP циклу.

7.5.1.2. Команди SYNC та FREEZE. Крім циклічного обміну даними процесу з Веденим, Ведучий може також відправляти команди в широкомовному режимі. Насамперед, це команди **SYNC** та **FREEZE**, які дозволяють синхронізувати процес зчитування та запису даних.

Так, при циклічному опитуванні різниця в часі опитування впливає на «свіжість» оновлення вхідних даних процесу. Тобто, при опитуванні Веденого 5 його дані будуть новішими, ніж Веденого 2, оскільки за час, який минув від останнього опитування Веденого 2, той отримав нові вхідні дані. Якщо необхідно, щоб вхідні дані Ведених були з однаковою часовою міткою, необхідно спочатку їх заморозити, а потім зчитати. Таким чином, на початку циклу Ведучий у широкомовному режимі відправляє команду «freeze», після якої Ведені заморожують плинне значення входів до наступної команди «freeze», а Ведучий зчитує ці значення. Якщо режим FREEZE необхідно відмінити, то Ведучий відправляє команду «unfreeze», після якої Ведені відновлюють вхідні дані процесу в довільний час.

Подібним чином працює режим SYNC. Ведучий відправляє команду «sync», тим самим зберігаючи виходи Ведених у попередньому значенні. Потім кожному з Ведених він записує значення вихідних змінних, однак Ведені не оновлюють виходи новими значеннями, поки не надійде наступна команда «sync». Таким чином виходи Ведених одночасно оновлюються в момент отримання широкомовної синхронізуючої команди.

7.5.2. Функції (сервіси) базових типів вузлів

У ROFIBUS DP визначено два класи Ведучих, набір сервісів яких відрізняється: Ведучий класу 1 (DPM1) та Ведучий класу 2 (DPM2).

DP Master Class 1 (**DPM1**) — це центральний контролер, який циклічно обмінюється даними користувача з Веденими. Типовим прикладом такого вузла є ПЛК. Йому доступні такі базові функції протоколу:

1. `Data_Exchange` — циклічний обмін даними процесу з призначеними йому Веденими;
2. `Set_Prm` и `Chk_Cfg` — відправка параметричних (конфігураційних) даних Веденому в момент його конфігурації та перевірка цих даних;
3. `Slave_Diag` — читання у Веденого діагностичної інформації;
4. `Global_Control` — управління станом Ведених.

DP Master Class 2 (**DPM2**) — це Ведучі, призначені для конфігурування та діагностики мережі (програмувальники). Пристрої DPM2 не повинні постійно функціонувати в мережі. DP Ведучим цього класу доступні всі описані вище функції та ряд додаткових:

- RD_Inp и RD_Outp — читання входів та виходів з потрібного Веденого;
- Get_Cfg — зчитування дійсних даних конфігурації Веденого;
- Set_Slave_Add — призначення нової адреси Веденому.

Ведені, тобто **DP-Slave**, обмінюються даними тільки з Ведучим DPM1, який попередньо його сконфігурував, або з Ведучим DPM2.

Таким чином, у ROFIBUS DP можливі три випадки з'єднання:

- між DPM1 (ініціатор обміну) та DP-Slave;
- між DPM2 (ініціатор обміну) та DP-Slave;
- між DPM1 та DPM2 (ініціатор обміну).

Обмін між Ведучими одного класу не підтримується базовими функціями ROFIBUS DP. Однак, можлива комбінація функцій в одному засобі: DPM1 та DPM2 чи DPM1 та DP-Slave.

7.5.3. Структура DP-циклу

На рис.7.16 та 7.17 показана структура циклічного обміну даними прикладно Процесу Ведучого зі своїми Веденими по шині. Крім циклічного обміну даними процесу (Data_Exchange), необхідне виділення часу для інших сервісів, які не потребують постійного використання ресурсів мережі, однак час від часу повинні функціонувати (обмін параметричними даними). Це такі функції, як параметризація, конфігурування та діагностика Ведених. Для функціонування таких сервісів у DP-V0 в DP-циклі виділена ациклічна частина (рис. 7.18).

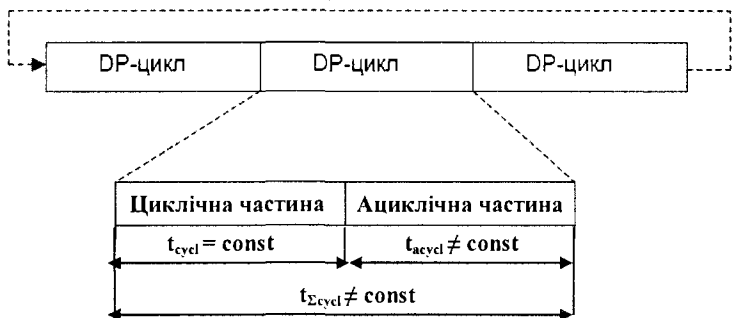


Рис. 7.18. Структура циклу ROFIBUS DP у системі з одним Ведучим

У циклічній частині DP-циклу проходить процес циклічного обміну даними процесу (Data_Exchange), які розглянуті вище. До змінної за часом ациклічної частини належать такі функції:

- обмін даними під час фази ініціалізації Веденого;
- діагностичні функції для Веденого;

- комунікації з Ведучим DPM2;
- повторення спотворених кадрів.

У протоколі DP-V0 обмін даними процесу з Веденим в ациклічній частині не підтримується. Однак, у циклічній частині при передачі повідомлення-відповіді Ведений може повідомити Ведучому про наявність *діагностичного переривання* (Diagnostic Interrupt) шляхом зміни пріоритету повідомлення на високий. Ведучий, отримавши таке повідомлення, в ациклічній частині дає повідомлення-запит типу SlaveDiag, на який Ведений передає відповідь з діагностичними даними.

Ациклічні функції мають низький пріоритет і можуть виконуватись протягом декількох циклів. Залежно від кількості цих дій час виконання конкретного DP-циклу може змінюватись.

У деяких системах необхідно, щоб час шинного циклу DP був постійним, тобто обмін даними процесу проходив строго періодично. Це, наприклад, актуально для систем управління електроприводами (PDS) для самосинхронізації декількох приводів. На відміну від нормального циклу DP, Ведучим передбачається (резервується) певна частина часу для ациклічного обміну (рис. 7.19). Ведучий слідкує, щоб зарезервована змінна частина не перевищила виділений час, обмежуючи кількість ациклічних повідомлень у DP-циклі. З іншого боку, якщо зарезервований час не використовується, Ведучий витримує необхідну паузу. Завдяки цьому гарантується, що заданий постійний час буде витриманий з точністю до мікросекунди.

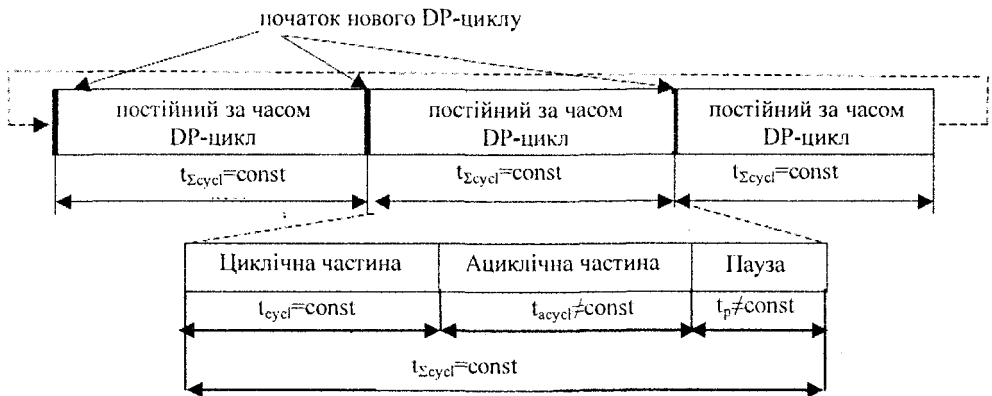


Рис. 7.19. Структура постійного за часом циклу ROFIBUS DP у системі з одним Ведучим

7.5.4. Конфігурування та ініціалізація мережі

7.5.4.1. Ініціалізація Веденого. Слід ще раз зазначити, що циклічний обмін у PROFIBUS DP — це ідентифікований обмін. Тобто Ведучий віддає Веденому вихідні дані та забирає вхідні, а кількість та призначення цих даних визначається при ініціалізації Веденого. По суті, процес налаштування такого зв'язку — це

домовленість між комунікаційними партнерами перед операційним функціонуванням циклічного обміну. Ця домовленість проходить з використанням службових сервісів управління мережею. Процес налаштування Веденого перед циклічним опитуванням називається *ініціалізацією*. Процес ініціалізації конкретного Веденого відбувається в ациклічній частині DP-циклу.

У PROFIBUS DP ініціалізує (конфігурує та параметризує) Веденого той Ведучий, з яким буде налаштований ідентифікований обмін. На рис.7.20 показані стадії процесу ініціалізації. Спочатку за допомогою діагностичних даних перевіряється готовність Веденого. Якщо DP-Slave готовий для параметризації, Ведучий відсилає параметричні та конфігураційні дані. Саме ця інформація потрібна Веденому, щоб визначити, якими саме даними процесу він буде обмінюватись із Ведучим у циклічному обміні. Позитивний результат перевірки готовності Веденого до роботи (діагностична інформація) завершує процес його ініціалізації. Коли Ведений проініціалізований, він переходить в операційний режим, тобто в режим циклічного обміну даними.

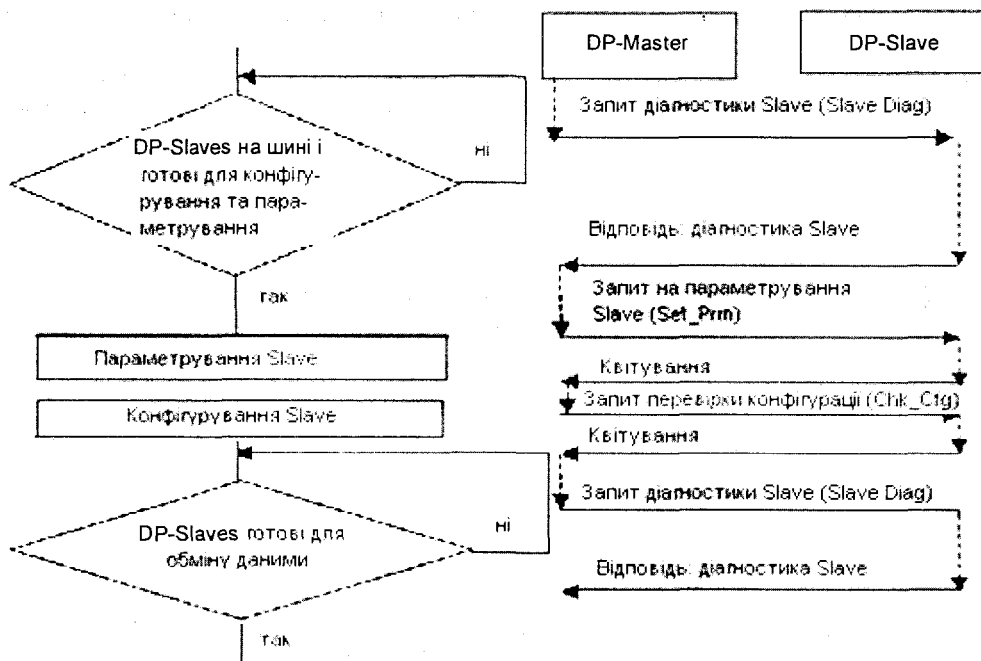


Рис. 7.20. Стадії ініціалізації Веденого

7.5.4.2. Режими роботи мережі. Поведінка Ведених на шині PROFIBUS DP залежить від режиму функціонування Ведучого DPМІ, який його ініціалізував. Ці режими стандартизовані і можуть приймати значення:

Offline — відсутній зв'язок Ведучого з Веденим;

Stop — циклічний обмін даними з Веденими відсутній;

Clear — Ведучий зчитує вхідну інформацію з Ведених, однак виходи виставляє в аварійному значенні (fail-safe state);

Operate — нормальний операційний режим, в якому доступні операції циклічного обміну.

DPM1 циклічно відправляє стан режиму всім призначеним веденим через визначені інтервали в ширококомовному режимі. Крім того, контролюється працездатність кожного Веденого, для чого Ведучий DPM1 виділяє для них окремі діагностичні таймери (Data_Control_Timer). Якщо протягом дозволеного часу Ведений не зреагував на запити Ведучого, той сигналізує про це програмі користувача. Крім того, якщо конфігураційний параметр Auto_Clear виставлений у стан 1, система переходить у стан Clear, про що повідомляється всім Веденим.

Ведені, в свою чергу, використовують сторожовий таймер (WatchDog), який відслідковує активність звернення до них Ведучого. Якщо таймер спрацював — вихідні дані, які оновлює даний Ведучий, виставляються в аварійне значення.

Слід зазначити, що в мультимастерній структурі тільки один Ведучий DPM1 може ініціалізувати конкретного Веденого, а також оновити його виходи. Інші Ведучі можуть тільки зчитувати з цього Веденого вхідні дані.

Прокоментуємо функції, які використовуються в процесі ініціалізації.

7.5.4.3. Дані параметризації (Set_Prm). За допомогою даних параметризації Веденому повідомляються необхідні локальні та глобальні параметри, властивості та функції. Зміст даних параметризації встановлюється при проектуванні Ведучого.

Проектування Ведучого включає конфігурацію мережі та настройки Ведених. Враховуючи велику різноманітність типів засобів, які можуть функціонувати на PROFIBUS DP в якості Веденого, необхідний універсальний механізм переліку та опису параметрів для конкретного засобу. В PROFIBUS DP для цього використовуються **GSD**-файли (Geräte Stamm Daten (нім.) — дані про походження приладу), які описані в 7.7.4. Підключивши до інструменту конфігурації мережі GSD-файл для конкретного засобу, можна додати в конфігурацію цей засіб в якості Веденого та налаштувати для нього необхідні параметри.

Таким чином, для кожного Веденого в пам'яті Ведучого містяться значення всіх параметрів, які необхідно змінити. Саме вони передаються на стадії параметризації. Зокрема, це такі параметри:

- Station-Status — специфічні настройки для поведінки Веденого;
- Watchdog — контроль часу спрацювання;
- Ident-Number — ідентифікаційний номер типу засобу, який видається PNO; цей ID зберігається в GSD-файлі і при параметризації відправляється Веденому; Ведений приймає дані параметризації тільки якщо отриманий ідентифікаційний номер у запиті співпадає з його власним, завдяки чому запобігається помилкова параметризація;

– Group-Ident — груповий ідентифікатор, за допомогою якого Ведені можна об'єднувати в групи для групових операцій типу «Sync» та «Freeze».

– User-Prm-Data — специфічні, характерні даному засобу дані для Веденого.

7.5.4.4. Дані конфігурування (Chk_Cfg). За допомогою конфігураційного кадру Ведучий повідомляє Веденому формат розпізнавання, обсяг та структуру

вхідних/вихідних даних процесу, які беруть участь в обміні. Ця область, яка називається також модулем, узгоджується за формою між Ведучим та Веденим (формат розпізнавання): буде це байтова структура чи структура зі слів. Через формат розпізнавання на модуль можна встановлювати вхідні/вихідні області розміром максимум у 16 байтів/слів.

Області вхідних/вихідних даних процесу, які вміщують зв'язну інформацію і тому мають передаватися тільки разом (одним масивом), повинні оброблятися як **консистентні дані**. До них належать, наприклад, області параметрів для регуляторів чи набори параметрів для приводу. За допомогою спеціальних форматів розпізнавання (які залежать від Веденого та виробника) можна встановлювати консистентність для області входів та виходів (модулів) з максимальною довжиною 64 байти або слова. Формат області входів/виходів процесу для Веденого зберігаються в GSD-файлах і при проектуванні пропонуються відповідним інструментом проектування.

7.5.4.5. Діагностичні дані (Slave_Diag). За допомогою запиту діагностичних даних Ведучий перевіряє на фазі запуску, чи присутній Ведений і чи готовий він для параметризації. Діагностичні дані від Веденого складаються із діагностичної частини і необов'язкової, специфічної для Веденого, діагностичної інформації. Через діагностичні дані Ведений повідомляє Ведучому свій робочий стан і у випадку діагностики — причину діагностичного переривання.

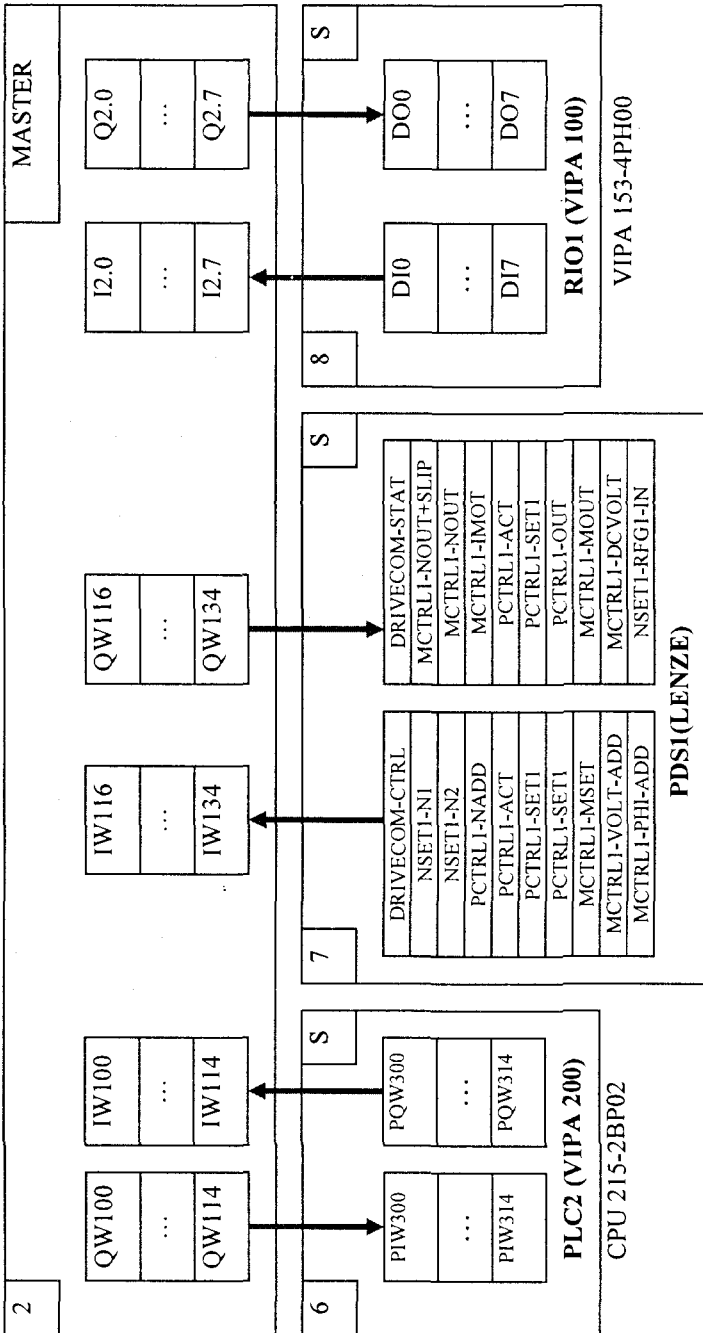
Приклад 7.2. Розробка проекту для мережі PROFIBUS DP

Завдання. Сконфігурувати всі вузли мережі PROFIBUS DP для реалізації поставленого на рис. 7.21 завдання, схема мережних з'єднань якої дана в прикладі 7.1.

Рішення. Щоб краще зрозуміти завдання, зробимо деякі пояснення. PLC1 (VIPA SPEED7) є центральним вузлом мережі, який повинен обмінюватися даними процесу з частотним перетворювачем (PDS1) та віддаленим модулем вводу/виводу (RIO1). Це значить, що він повинен бути активним вузлом, тобто Ведучим на шині PROFIBUS DP. Частотний перетворювач повинен обмінюватися даними з PLC1 статусною інформацією та значеннями різних поточних та заданих параметрів, призначення яких не принципово для розв'язання даного завдання. Дискретні входи та виходи модуля віддаленого вводу/виводу повинні відображатися в змінних PLC1. По суті, для програми користувача в PLC1 значення входів та параметрів на периферійних засобах — це лише вхідні/вихідні змінні, які у ПЛК VIPA та SIEMENS позначаються як I (входи) та Q (виходи).

PLC1 повинен координувати роботу PLC2 (VIPA 200), обмінюючись в обох напрямках 8-мома 16-бітними змінними. Для цього PLC2 на цій шині виступає в якості Веденого вузла (інтерфейс DP, який інтегрований у CPU 215-2BP02, може використовуватись як у режимі Ведучого, так і Веденого). Такий обмін можливий з використанням периферійних вхідних та вихідних змінних PLC2, які повинні бути виділені для цього. Слід зазначити, що з точки зору PLC2 змінні PIW (периферійні вхідні слова) є входами, однак в адресному просторі Ведучого, тобто PLC1, вони будуть відображатися як виходи. Аналогічна ситуація і для PQW.

PLC1 (VIPA SPEED7)



E82EV371K2C

Рис. 7.21. Постановка завдання до прикладу 7.2

Для функціонування такої системи необхідно налаштувати та сконфігурувати кожний вузол. Прокоментуємо, яким чином необхідно налаштувати кожного з Ведених.

PLC2 — це інтелектуальний Ведений. Ведучий не ініціалізує його, оскільки для нього проект створюється за допомогою спеціальних інструментів. У даному випадку можна використати Step7. Для роботи в мережі відповідно до завдання необхідно налаштувати DP-канал, тобто:

- вказати режим роботи = Slave;
- вказати адресу = 6 (див. рис. 7.21);
- вказати бітову швидкість = 12 Мбіт/с;
- при необхідності налаштувати додаткові параметри.

Необхідно вказати діапазон периферійної пам'яті, виділеної для обміну з Ведучим: (input bytes 300-315, output bytes 300-315).

Частотний перетворювач PDS1 (LENZE) можна налаштувати за допомогою панелі або спеціального ПЗ. Необхідно вказати адресу (C1509 = 7), вказати джерелом завдання функціональний модуль (C0005 = 200). Бітова швидкість визначається автоматично.

Модуль віддаленого вводу/виводу RIO1(VIPA 100) має перемикач адреси, яким необхідно виставити адресу 8. Усі інші настройки визначаються Ведучим у процесі ініціалізації.

Налаштування Ведучого шини PROFIBUS DP, тобто PLC1, проводиться в проекті для даного контролера. В якості інструменту проектування будемо використовувати Step7, який дасть можливість більш наочно продемонструвати шляхи створення конфігурації мережі.

Конфігурацію мережі можна сформувати в розділі HWConfig (Hardware Configuration), де створюється апаратна конфігурація для ПЛК. Мережа конфігурується в настройках комунікаційного модуля, де вказуються:

- режим роботи = Master;
- адреса = 2 (див. рис. 7.21);
- бітова швидкість = 12 Мбіт/с;
- за необхідності додаткові параметри.

Після конфігурації самої мережі необхідно вказати Ведені, які «належать» даному ПЛК (адже при мультимастерній архітектурі на мережі можуть бути Ведені для інших Ведучих). Для включення та налаштування Ведених, відповідно до їх профілю, в Step7 необхідно підключити всі необхідні GSD-файли:

- для VIPA CPU2xxDP;
- для VIPA DP100V;
- для модуля Lenze E82ZAFPC001.

Після підключення цих файлів профілів у каталозі профілів з'являться нові розділи та комунікаційні об'єкти. На рис. 7.22 показані три частини каталогу, які належать до Ведених у даному завданні. Прокоментуємо можливі комунікаційні об'єкти, які доступні для обміну з цими пристроями.

Для VIPA CPU2xxDP можна сконфігурувати обмін даними різної довжини (до 8 слів) з можливістю консистентної передачі. В поставленому завданні необхідно передавати і отримувати 8 слів.

Для VIPA DP100V вибирається конкретна модель модуля, в нашому випадку це 153-4PH00 (DIO8), що передбачає обмін одним байтом в обидва боки.

VIPA_System_200V

- VIPA_253-1DP01 (DPV0)
- VIPA_253-1DP01 (DPV1)
- VIPA_253-1DP11 (DPV0)
- VIPA_253-1DP11 (DPV1)
- VIPA_253-1DP30 (DPV0)
- VIPA_253-1DP30 (DPV1)
- VIPA_253_2DP20
- VIPA_CPU21x
- VIPA_CPU2xDP

Universal module

- 1 Byte Input
- 2 Byte Input
- 4 Byte Input
- 8 Byte Input
- 1 Byte Output
- 2 Byte Output
- 4 Byte Output
- 8 Byte Output
- 1 Word Input, Consistency
- 2 Word Input, Consistency
- 4 Word Input, Consistency
- 1 Word Output, Consistency
- 2 Word Output, Consistency
- 4 Word Output, Consistency
- 1 Word In/Out, Consistency
- 2 Word In/Out, Consistency
- 4 Word In/Out, Consistency
- 8 Word In/Out, Consistency

VIPA_System_200V

- VIPA_253-1DP01 (DPV0)
- VIPA_253-1DP01 (DPV1)
- VIPA_CPU11x
- VIPA_CPU11xDP

VIPA_SPEEDbus

- Universal module
- 121-4BH00 DI16xDC24V
- 121-6BH00 DI16xDC24V
- 121-6BL00 DI32xDC24V
- 122-4BH00 DO16xDC24V
- 122-6BH00 DO16xDC24V
- 122-6BH80 DO16xDC24V
- 122-6BL00 DO32xDC24V
- 122-6H000 DO16xRelay
- 123-4BF00 DI08xDC24V
- 123-4BH00 DI08xDC24V
- 123-6BL00 DI016xDC24V
- 123-6BL10 DI032xDC24V
- 123-6BL80 DI016xDC24V
- 151-4FH00 DI16xDC24V
- 151-6FL00 DI32xDC24V
- 152-4FH00 DO16xDC24V
- 152-6FH00 DO16xDC24V
- 152-6FH50 DO16xRelay
- 152-6FH80 DO16xDC24V
- 152-6FL00 DO32xDC24V
- 153-4FF00 DI08xDC24V
- 153-4FH00 DI08xDC24V
- 153-6FH00 DI08xDC24V
- 153-6FL00 DI016xDC24V
- 153-6FL80 DI016xDC24V

VIPA_System_200V

- VIPA_253-1DP01 (DPV0)

PROFIBUS DP

- Additional Field Devices
- Drives
- Lenze
- 820Inu-Profinbus-DP
- Universal module
- PAR(Cons.)+PZD(1 Word)
- PAR(Cons.)+PZD(2 Words)
- PAR(Cons.)+PZD(3 Words)
- PAR(Cons.)+PZD(4 Words)
- PAR(Cons.)+PZD(5 Words)
- PAR(Cons.)+PZD(6 Words)
- PAR(Cons.)+PZD(7 Words)
- PAR(Cons.)+PZD(8 Words)
- PAR(Cons.)+PZD(9 Words)
- PAR(Cons.)+PZD(10 Words)
- PAR(Cons.)+PZD(11 Words)
- PAR(Cons.)+PZD(12 Words)
- PAR(Cons.)+PZD(13 Words)
- PAR(Cons.)+PZD(14 Words)
- PAR(Cons.)+PZD(15 Words)
- PAR(Cons.)+PZD(16 Words)
- PAR(Cons.)+PZD(17 Words)
- PAR(Cons.)+PZD(18 Words)
- PAR(Cons.)+PZD(19 Words)
- PAR(Cons.)+PZD(20 Words)
- PAR(Cons.)+PZD(21 Words)
- PAR(Cons.)+PZD(22 Words)
- PAR(Cons.)+PZD(23 Words)
- PAR(Cons.)+PZD(24 Words)
- PAR(Cons.)+PZD(25 Words)
- PAR(Cons.)+PZD(26 Words)
- PAR(Cons.)+PZD(27 Words)
- PAR(Cons.)+PZD(28 Words)
- PAR(Cons.)+PZD(29 Words)
- PAR(Cons.)+PZD(30 Words)
- PAR(Cons.)+PZD(31 Words)
- PAR(Cons.)+PZD(32 Words)
- PAR(Cons.)+PZD(33 Words)
- PAR(Cons.)+PZD(34 Words)
- PAR(Cons.)+PZD(35 Words)
- PAR(Cons.)+PZD(36 Words)
- PAR(Cons.)+PZD(37 Words)
- PAR(Cons.)+PZD(38 Words)
- PAR(Cons.)+PZD(39 Words)
- PAR(Cons.)+PZD(40 Words)
- PAR(Cons.)+PZD(41 Words)
- PAR(Cons.)+PZD(42 Words)
- PAR(Cons.)+PZD(43 Words)
- PAR(Cons.)+PZD(44 Words)
- PAR(Cons.)+PZD(45 Words)
- PAR(Cons.)+PZD(46 Words)
- PAR(Cons.)+PZD(47 Words)
- PAR(Cons.)+PZD(48 Words)
- PAR(Cons.)+PZD(49 Words)
- PAR(Cons.)+PZD(50 Words)
- PAR(Cons.)+PZD(51 Words)
- PAR(Cons.)+PZD(52 Words)
- PAR(Cons.)+PZD(53 Words)
- PAR(Cons.)+PZD(54 Words)
- PAR(Cons.)+PZD(55 Words)
- PAR(Cons.)+PZD(56 Words)
- PAR(Cons.)+PZD(57 Words)
- PAR(Cons.)+PZD(58 Words)
- PAR(Cons.)+PZD(59 Words)
- PAR(Cons.)+PZD(60 Words)
- PAR(Cons.)+PZD(61 Words)
- PAR(Cons.)+PZD(62 Words)
- PAR(Cons.)+PZD(63 Words)
- PAR(Cons.)+PZD(64 Words)
- PAR(Cons.)+PZD(65 Words)
- PAR(Cons.)+PZD(66 Words)
- PAR(Cons.)+PZD(67 Words)
- PAR(Cons.)+PZD(68 Words)
- PAR(Cons.)+PZD(69 Words)
- PAR(Cons.)+PZD(70 Words)
- PAR(Cons.)+PZD(71 Words)
- PAR(Cons.)+PZD(72 Words)
- PAR(Cons.)+PZD(73 Words)
- PAR(Cons.)+PZD(74 Words)
- PAR(Cons.)+PZD(75 Words)
- PAR(Cons.)+PZD(76 Words)
- PAR(Cons.)+PZD(77 Words)
- PAR(Cons.)+PZD(78 Words)
- PAR(Cons.)+PZD(79 Words)
- PAR(Cons.)+PZD(80 Words)
- PAR(Cons.)+PZD(81 Words)
- PAR(Cons.)+PZD(82 Words)
- PAR(Cons.)+PZD(83 Words)
- PAR(Cons.)+PZD(84 Words)
- PAR(Cons.)+PZD(85 Words)
- PAR(Cons.)+PZD(86 Words)
- PAR(Cons.)+PZD(87 Words)
- PAR(Cons.)+PZD(88 Words)
- PAR(Cons.)+PZD(89 Words)
- PAR(Cons.)+PZD(90 Words)
- PAR(Cons.)+PZD(91 Words)
- PAR(Cons.)+PZD(92 Words)
- PAR(Cons.)+PZD(93 Words)
- PAR(Cons.)+PZD(94 Words)
- PAR(Cons.)+PZD(95 Words)
- PAR(Cons.)+PZD(96 Words)
- PAR(Cons.)+PZD(97 Words)
- PAR(Cons.)+PZD(98 Words)
- PAR(Cons.)+PZD(99 Words)
- PAR(Cons.)+PZD(100 Words)
- PAR(Cons.)+PZD(101 Words)
- PAR(Cons.)+PZD(102 Words)
- PAR(Cons.)+PZD(103 Words)
- PAR(Cons.)+PZD(104 Words)
- PAR(Cons.)+PZD(105 Words)
- PAR(Cons.)+PZD(106 Words)
- PAR(Cons.)+PZD(107 Words)
- PAR(Cons.)+PZD(108 Words)
- PAR(Cons.)+PZD(109 Words)
- PAR(Cons.)+PZD(110 Words)
- PZD(1 Word)
- PZD(2 Words)
- PZD(3 Words)
- PZD(4 Words)

Рис. 7.22. Доступні об'єкти в каталозі профілів (до прикладу 7.2)

Об'єкти Lenze теж можна вибрати відповідно до завдання. При цьому виділяються два типи об'єктів: PZD (змінні процесу) та PAR (параметри). За допомогою об'єкта PAR, який займає 4 вхідні + 4 вихідні слова, можна доступитись за індексом до будь-якого параметра частотного перетворювача як для читання, так і для запису. Об'єкти PZD передають значення змінних процесу та задані значення. Відповідно до поставленого завдання, об'єкти PAR не використовуються, а використовуються 10 вхідних та вихідних об'єктів PZD.

Використовуючи каталог профілів, вибирається профіль засобу, наприклад VIPA CPU2xxDP, для якого налаштовуються адреса та загальні параметри. Загальний вигляд конфігурації для поставленого завдання показаний на рис. 7.23.

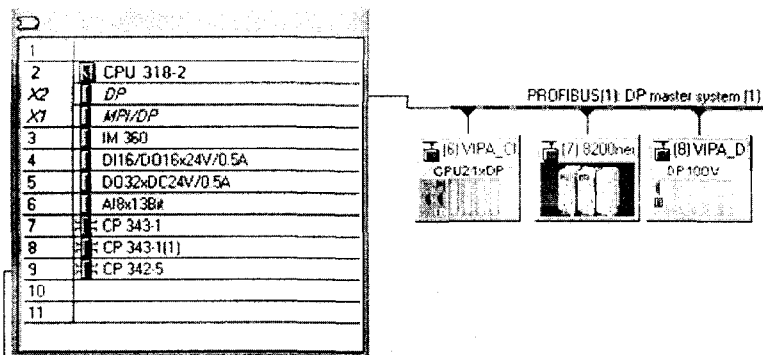


Рис. 7.23. Конфігурація PROFIBUS DP в проєкті Ведучого в Step7 (до прикладу 7.2)

Для кожного Веденого необхідно вказати комунікаційні об'єкти, які вибираються в каталозі профілів. Автоматично для кожного з цих об'єктів в області вводу/виводу процесу Ведучого виділяється пам'ять (налаштовується їх відображення на образ процесу Ведучого). Розподіл пам'яті можна змінити відповідно до завдання рис. 7.24.

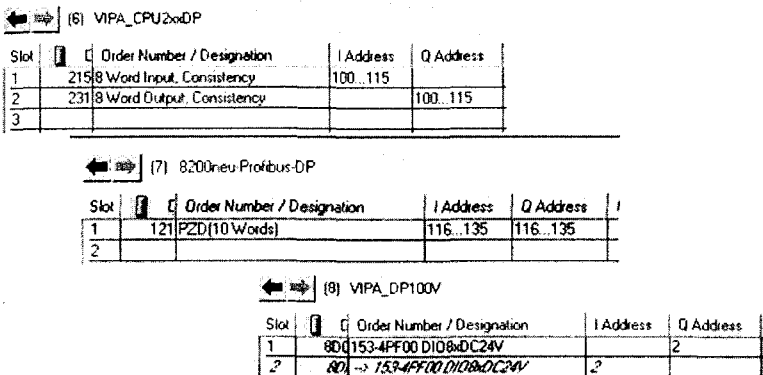


Рис. 7.24. Налаштування комунікаційних об'єктів Ведених та їх відображення в адресному просторі Ведучого (до прикладу 7.2)

Якщо немає необхідності в додаткових настройках, процес конфігурування можна вважати завершеним.

Приклад 7.2

7.6. Розширені функції обміну даними між вузлами в PROFIBUS DP (DP-V1, DP-V2)

7.6.1. Версії DP-V1 та DP-V2

З розвитком систем автоматизації та підвищенням вимог до комунікаційного зв'язку PROFIBUS DP удосконалювався та доповнювався новими функціями. Так з'явилися нові версії протоколів PROFIBUS DP-V1 та PROFIBUS DP-V2, що доповнилися розширеними функціями, які є необов'язковими для використання. На рис. 7.25 показано добавлення нових функцій при переході до нових версій.

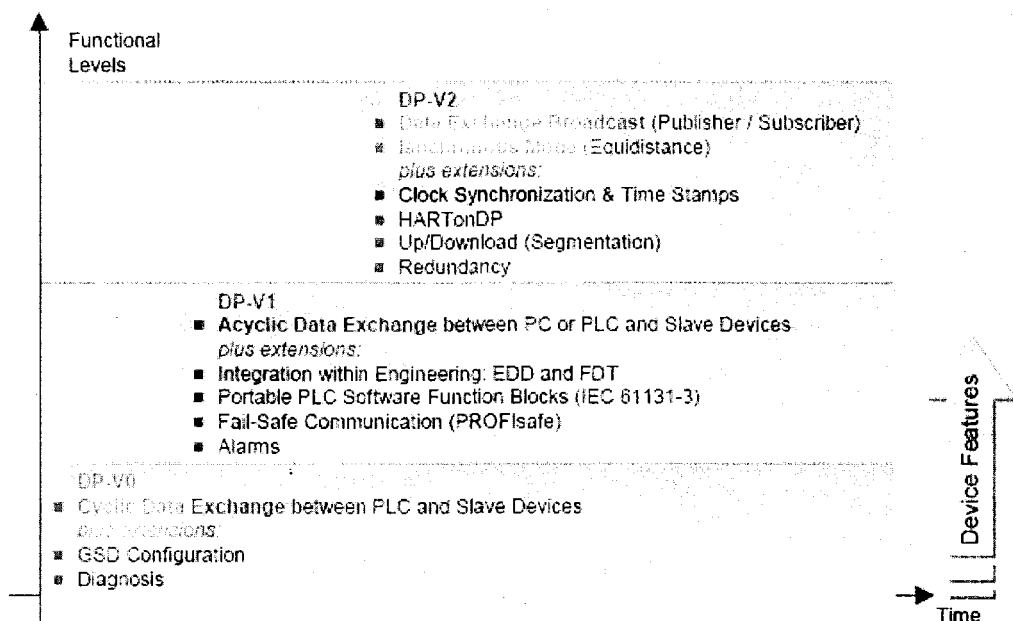


Рис. 7.25. Функціональність PROFIBUS DP від версії до версії

Основні функції DP-V0 перераховані в попередньому підрозділі. Структура GSD-файлів, які підтримувалися ще в базовій версії разом з доповненнями, приводиться в наступному підрозділі. Ведучі та Ведені, які підтримують розширення, називаються, відповідно, Master DP-V1/Master DP-V2 та Slave DP-V1/Slave DP-V2.

Основною відмінністю DP-V1 від попередньої версії є поява функції *ациклического обміну даними процесу (Acyclic Data Exchange)* паралельно з функціонуванням циклічного. Такий зв'язок характерний для генерування подій (Events), тривог (Alarm), обміну з ЛІМІ, які не потребують постійного трафіка. Пізніше PROFIBUS DP був удосконалений до DP-V2, в основному за рахунок додаткових

функцій для роботи з приводами (PDS): ізохронним способом обміну даними (*Isochronous Mode*) та широкомовним обміном даними між Процесами вузлів по типу Видавець/Абонент (*Data Exchange Broadcast-DXB*).

7.6.2. Ациклічний обмін даними процесу

Для функціонування ациклічного обміну даними процесу використовується механізм переривань (Interrupt), аналогічний функціонуванню діагностичного переривання, яке ми розглянули в попередньому підрозділі. Тобто при циклічному обміні даними процесу Ведений передає повідомлення-відповідь з високим пріоритетом, яке вимагає від Ведучого додаткового обміну в ациклічному трафіку. А в ациклічній частині DP-циклу Ведений уже передає ациклічні дані процесу.

7.6.3. Широкомовний обмін даними процесу

Data Exchange Broadcast забезпечує прямим зв'язком між Веденими за принципом pull моделі Видавець/Абонент (рис. 7.26). При запиті від Ведучого прикладний процес Веденого в широкомовному режимі відправляє кадр з вхідними даними, на які можуть підписатись прикладні Процеси інших Ведених. Це розширення дає можливість значно зменшити навантаження на шину, оскільки в задачах, де доводиться організовувати обмін між Веденими, в DP-V0/V1 цим займається Ведучий.

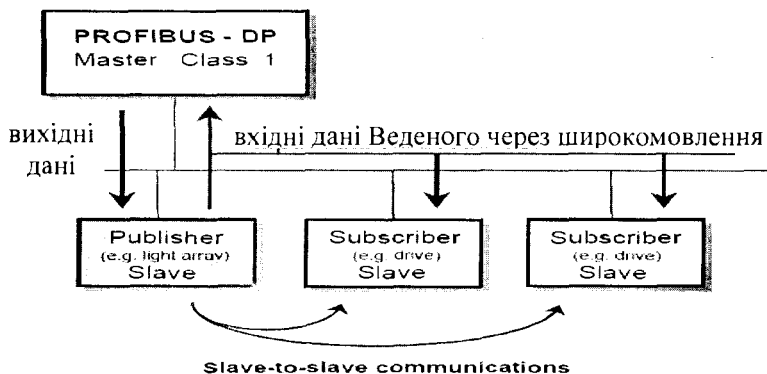


Рис. 7.26. Широкомовний обмін даними процесу

7.6.4. Ізохронний режим

Режим Isochronous Mode дозволяє реалізувати часово-синхронну передачу даних незалежно від навантаження шини. Функція дає змогу реалізувати високоточні процеси позиціонування з часовим відхиленням менш ніж одна мікросекунда. Для цього всі цикли пристроїв синхронізовані до циклу Ведучого через широкомовне повідомлення «global control». Спеціальний сигнал (послідовний номер циклу) дозволяє контролювати цикли синхронізації. На рис. 7.27 показані: доступний час для обміну даними процесу (DX), час доступу Ведучих DPM2 (MSG) та час паузи (Reserve). Синхронізація годинників вузлів дає змогу чітко визначити ін-

тервали опитування входів, обробки процесу та запису виходів. Для зменшення часу реакції системи (оптимізації) можна вказати значення часових зсувів: опитування входів Ведених відносно кінця DP-циклу (T_i), запису виходів відносно початку DP-циклу (T_o), запуску завдання відносно початку DP-циклу (T_m).

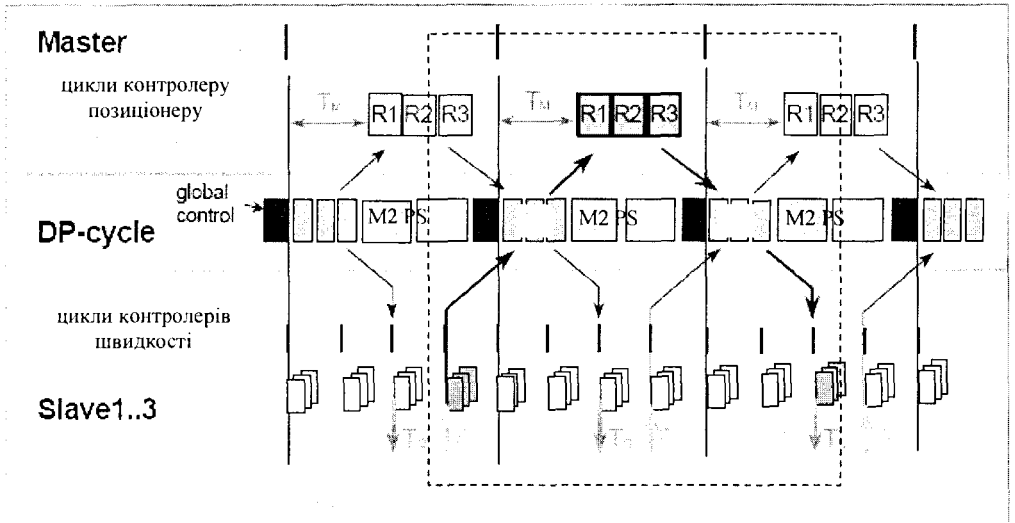


Рис. 7.27. Обмін в ізохронному режимі

7.7. Профілі в PROFIBUS DP

7.7.1. Технічна структура PROFIBUS

На рис. 7.1 показана структура PROFIBUS у контексті моделі OSI. Однак, у специфікаціях PROFIBUS стандартів DIN та EN для PROFIBUS DP прикладний рівень не описаний, а в IEC 61158 до цього рівня увійшли функції інтерфейсу DDLM та профілі. А у загальнодоступному документі від PNO наводиться така модель (технічна структура системи) PROFIBUS (рис. 7.28).

На фізичному рівні описані різні типи реалізації фізичних інтерфейсів, тобто різні технології передачі (Transmission Technologies), загальний опис яких наведений у табл. 7.4. На рівні комунікаційних технологій (Communication Technologies), який також називається протокольним рівнем, визначені два типи протоколів FMS та DP (DP-V0, DP-V1, DP-V2). Прикладний рівень визначається системою прикладних профілів та технологій: Application Profiles I (загальні прикладні профілі); Application Profiles II (специфічні прикладні профілі для пристроїв); Integration Technologies — інтеграційні технології (методи та інструменти), які забезпечують опис та швидке впровадження різномірних пристроїв в системі PROFIBUS; System Profiles — системні профілі, тобто діапазон стандартів (профілі Ведучих, інтерфейси), які забезпечують реалізацію однорідних, стандартизованих систем.

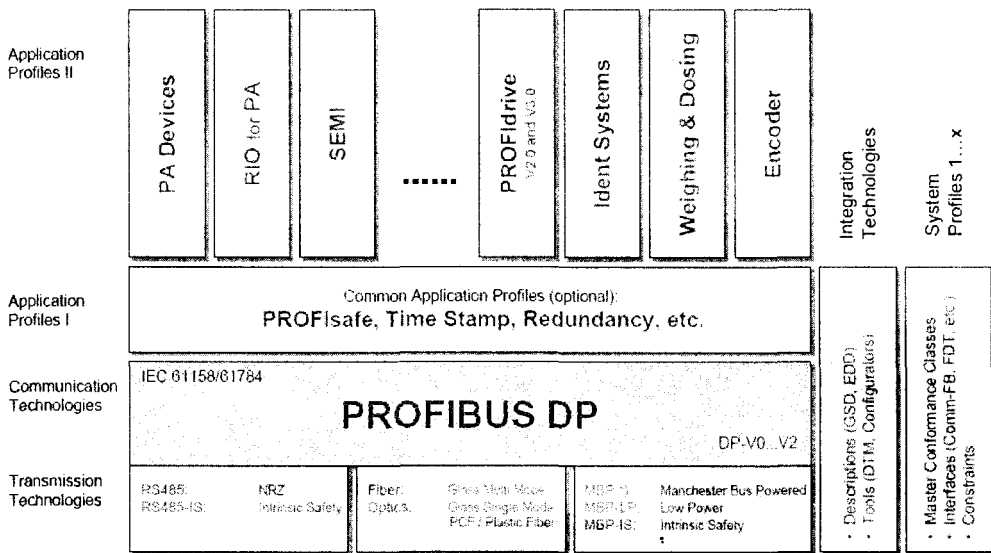


Рис. 7.28. Технічна структура системи PROFIBUS

Таблиця 7.4

**ХАРАКТЕРИСТИКИ МЕРЕЖ PROFIBUS НА ФІЗИЧНОМУ РІВНІ
З РІЗНИМИ ВАРІАНТАМИ РЕАЛІЗАЦІЇ**

Характеристика	RS485	RS-485-IS	MBP	Fiber Optic
<i>PhInterface</i>	RS-485, NRZ		синхронний інтерфейс, модуляція постійної напруги, манчестерське кодування	NRZ
<i>PhMedia</i>	екранована вита пара, кабелі А та В;	подвійна скранована вита пара кабелі А та В;	екранована вита пара типів А–D;	оптоволоконні мультимодові кабелі; пластикові, скляні, PCF
<i>PhTopology</i>	шина		шина/дерево	зірка/кільце/шина
<i>PhLdrop</i>	допускається тільки з використанням активних з'єднувачів;		<30 м, $PhLength = \sum PhLdrop + Ltrunk$	допускається
<i>PhBaudRate</i>	9600 кбіт/с – 12 Мбіт/с	9600 кбіт/с – 1.5 Мбіт/с	31.25 Кбіт/с	9600 кбіт/с – 12 Мбіт/с
<i>PhSegment</i>	сегментація з використанням релітерів/мостів, сегментних з'єднувачів			
	9	9	4	не обмежена
<i>PhNodes</i>	32 (126 на всю мережу)			126 на мережу
<i>PhLength</i>	до 1000 м (при 187 кбіт/с), термінатор 390 Ом в обох кінцях лінії		1900 м, термінатор (100 Ом+1мкФ) в обох кінцях лінії	кілька кілометрів
<i>PhSupply</i>	можливе до 30 В по додатковій парі проводів		до 30 В по інформаційній парі проводів	можливе до 30 В по додатковій парі проводів
тип захисту	немає	Intrinsic Safety (EEx ib)	Intrinsic Safety (EEx ia/ib)	немає

Профілі PROFIBUS — це специфікації, визначені виробником і користувачами відносно специфічних властивостей, особливостей виконання та поведінки пристроїв і систем. **Специфікації профілю** визначають параметри та поведінку пристроїв та систем, які належать сімейству профілю. **Сімейства профілів** розроблені для полегшення здатності до взаємодії з пристроєм, а в деяких випадках для полегшення взаємозаміни пристроїв одного сімейства на інші. У профілях враховуються особливості застосування і визначені типом спеціальні особливості польових пристроїв, управління і методів інтеграції (інжинірингу). Розрізняють:

- **general application profiles** (загальні прикладні профілі, Application Profiles I) — це профілі, які описують специфіку пристроїв для різних варіантів виконання; наприклад, профілі PROFIsafe для систем функціональної безпеки, Redundancy — для систем з резервуванням);
- **specific application profiles** (специфічні прикладні профілі, Application Profiles II) — це профілі, які розвинуті для пристроїв певного класу задач; наприклад, PROFIdrive — для електроприводів PDS, PanelDevices — для засобів людино-машинних інтерфейсів;
- **system and master profiles** (системні профілі) — це профілі, які описують функціональні можливості систем, доступних пристроям: DPM1, DPM2, DP Slave, DP-V0/V1/V2.

7.7.2. Загальні прикладні профілі

Загальні прикладні профілі (Application Profiles I) описують функції та характеристики, які стосуються області застосування. Вони можуть також використовуватися в поєднанні зі специфічними прикладними профілями.

7.7.2.1. PROFIsafe. Донедавна для функціонально небезпечних процесів та виробництв використання промислових мереж було проблематичним. З появою профілю **PROFIsafe** PROFIBUS створив всебічне, відкрите рішення для функціонально безпечного (safety-base) застосування, яке задовольняє найбільш безпечним критеріям. PROFIsafe визначає, яким чином запобіжні засоби (кнопка аварійної зупинки, лампи) можуть зв'язатися за PROFIBUS із запобіжними контролерами (контролери ПАЗ) настільки безпечно, щоб можна було їх використовувати для зв'язаних з функціональною безпекою завдань до категорій KAT4, сумісною з EN954, AK6 або **SIL3** (Safety Integrity Level). Це реалізовує безпечні з'єднання за профілем, тобто за спеціальним форматом даних користувача та спеціального протоколу. Специфікація була складена суспільно виробниками, користувачами, комітетами стандартизації та інспектування (TÜV, VIA). Вона базується на сумісних стандартах, перш за все, IEC 61508, які заодно стосуються розробників програмного забезпечення (рис. 7.29).

PROFIsafe приймає до уваги кількість імовірних помилок, які можуть відбутися в шинах, типу затримки, втрати або повторення даних, неправильних послідовностей, спотворення даних. Є декілька способів корекції, частина яких були відібрані для PROFIsafe, а саме: послідовна нумерація телеграм; тайм-аут для

вхідних кадрів-повідомлень та їх підтвердження; ідентифікатор між відправником та отримувачем («пароль»); додатковий захист даних (CRC).

Комбінуючи ці заходи корекції разом із запатентованим «SIL monitor» (контроль частоти помилкових повідомлень), PROFIsafe досягає класів безпеки до SIL3 та вище.

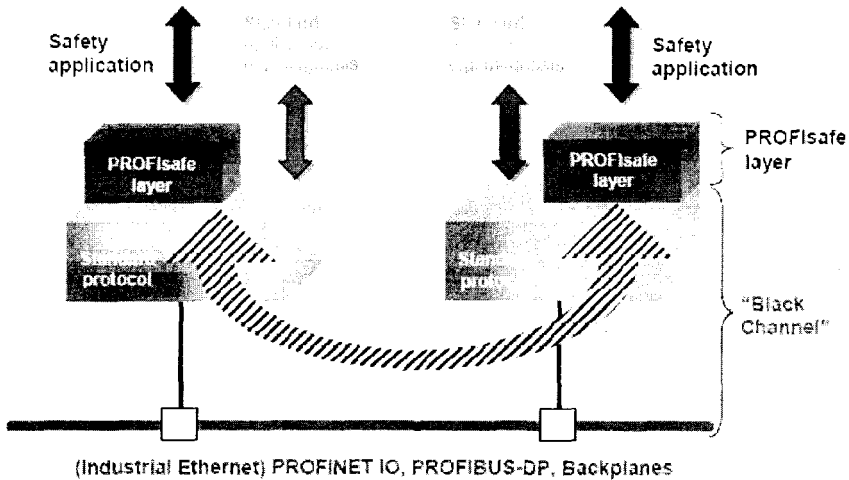


Рис. 7.29. Реалізація PROFIsafe

PROFIsafe — це програмне рішення, яке впроваджене в пристроях як додатковий рівень над прикладним рівнем, при цьому стандартні компоненти типу ліній зв'язку, ASICs або протоколи залишаються незмінними. Пристрої з профілем PROFIsafe можуть використовуватись у сумісності зі стандартними пристроями на тому самому кабелі без якихось обмежень. PROFIsafe використовує ациклічний обмін даними процесу і може використовуватись на технологіях передачі RS485, оптоволокні або MBP.

7.7.2.2. Time Stamp. При реєстрації часових подій у мережах, особливо для діагностики і визначення помилки, корисно забезпечити точний час виникнення події. Для цього PROFIBUS пропонує профіль *Time Stamp* (часовий відбиток). При виникненні повідомлення шляхом ациклічного обміну даними Ведучий зчитує не тільки саме повідомлення, а і його часовий відбиток.

7.7.2.3. Slave Redundancy. У багатьох випадках необхідна реалізація системи з дублюванням зв'язку. З цієї причини для PROFIBUS створили специфікацію механізму дублювання ведених *Slave Redundancy* (резервування Веденого), який описує такі характеристики пристрою (рис. 7.30):

– ведені пристрої повинні мати два різні інтерфейси PROFIBUS, які називаються первинним (primary) та резервним (backup); вони можуть бути на одному пристрої (резервування шляхів) або розподілені між декількома пристроями (резервування пристроїв);

– пристрої обладнані двома незалежними стеками протоколів зі спеціальним розширенням для резервування;

– резервний зв'язок redundancy communication (RedCom) відбувається між стеками протоколів, тобто в межах пристрою (при резервуванні шляхів) або між двома пристроями (при резервуванні пристроїв).

У нормальному режимі з'єднання відбувається винятково з первинним Веденим, і тільки він конфігурується. За резервним зв'язком Ведучий відсилає діагностичні дані та дані процесу резервному Веденому. Коли первинний Ведений виходить з ладу, резервний Ведений приймає на себе його функції або в результаті безпосереднього виявлення відмови, або за проською Ведучого. Резервний Ведений пристрій може використовуватись на одній PROFIBUS лінії, або, у випадку додаткового резервування лінії, — на двох лініях зв'язку.

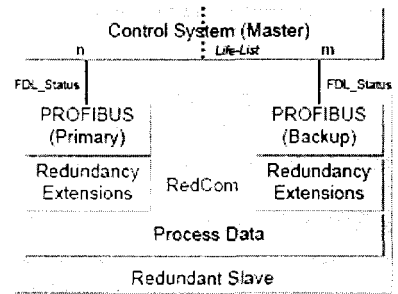


Рис. 7.30. Резервування Ведених

7.7.3. Специфічні прикладні профілі

PROFIBUS виділяється серед інших промислових мереж великою гамою підтримуваних пристроїв для різних прикладних задач. Для цього розроблено ряд специфічних прикладних профілів (Application Profiles II) для конкретних типів пристроїв, які наведені в табл. 7.5.

Таблиця 7.5

СПЕЦИФІЧНІ ПРИКЛАДНІ ПРОФІЛІ

Профіль	Застосування профілю
PROFIdrive	визначає поведінку пристроїв та процедури доступу до даних для електроприводів (PDS)
PA devices	визначає характеристики PA-пристроїв
Robots/NC	описує, яким чином управляються роботи, які призначені для обробки та збирання
Panel devices	описує інтерфейс для простих засобів людино-машинного інтерфейсу до високорівневих компонентів автоматизації
Encoders	описує інтерфейс до поворотних, кутових та лінійних енкодерів
Fluid power	описує, яким чином управляються гідравлічні приводи
SEMI	описує характеристики пристроїв для виробництва напівпровідників за стандартом SEMI
Low-voltage switchgear	описує обмін даними для різних пристроїв вимикання/перемикання (вимикачі, магнітні пускачі)
Dosign/weighing	описує використання систем зважування та дозування

Профіль	Застосування профілю
Ident systems	описує комунікації між пристроями ідентифікації (сканери штрих кодів, транспондери)
Liquid pumps	визначає використання рідинних насосів
Remote I/O for PA devices	для пристроїв вводу/виводу для PROFIBUS PA

Використання прикладних профілів дає можливість спростити процедуру розробки системи з однотипних пристроїв від різних виробників, оскільки їх поведінка та структура даних визначаються однаковими специфікаціями профілю. Крім того, така уніфікація зменшує кількість помилок, оскільки профілі розробляються з урахуванням багатолітнього досвіду експлуатації подібних пристроїв. У даному посібнику відсутній опис наведених вище профілів, за винятком PROFIdrive, який наведений в розділі 11.

7.7.4. Використання GSD для інтеграції пристроїв у мережу

Сучасні польові засоби забезпечують широкий діапазон інформації та функцій, які перед цим виконувались у контролерах та в системах управління. Для цього необхідні інструменти для вводу в дію, обслуговування, розробки та параметризації таких пристроїв, які потребують точного та повного опису даних для пристрою та його функцій, таких, як: тип прикладної функції; конфігураційні параметри; діапазон значень; одиниці вимірювання; значення за замовчуванням; значення граничних меж; ідентифікація і таке інше.

Те ж саме стосується контролера/системи управління, специфічні параметри яких та формат даних повинні також бути повідомлені (інтегровані) для гарантії безпомилкового обміну даними з польовими пристроями.

У PROFIBUS окреслилося багато методів та інструментів («інтеграційних технологій») для опису пристроїв. В автоматизованих системах управління процесом переважно використовують технологію GSD-опису, в якому описуються особливості зв'язку між пристроями PROFIBUS. GSD зручний для простих застосувань і розробляється та поставляється виробником обладнання. GSD-опис оформлений у вигляді текстового GSD-файлу формату ASCII, який вміщує основну та специфічну інформацію для зв'язку з конкретним типом пристрою. Для конфігурації цього пристрою за допомогою ключових слів інструмент конфігурації зчитує з GSD ідентифікатор пристрою, настроювальні параметри, відповідний тип даних та межі їх значень і т.д. Деякі із ключових слів обов'язкові, наприклад Vendor_Name, а інші є опціональними, наприклад, Sync_Mode_supported. GSD-файли поділені на три секції.

1.General Specifications. Ця секція включає інформацію про виробника та назву пристрою, апаратну та програмну версії, підтримувані швидкості, часові інтервали для контролю та призначення сигналів на з'єднувачах.

2. Master Specifications. Ця секція включає параметри Ведучого, такі як максимальна кількість можливих Ведених, завантажувани/вивантажувани опції. Для пристроїв, які можуть бути тільки Веденими, — ця секція відсутня.

3. Slave Specifications. Секція включає всю інформацію, специфічну для даного пристрою, як Веденого. Це може бути кількість та тип вхідних/вихідних каналів, специфікація тексту діагностичного повідомлення та інформація про доступні модулі у випадку модульного пристрою.

GSD-формат розроблений для максимальної зручності у використанні (наявні списки, опції і т.д.) та підтримує інтеграцією растрових малюнків для символічного зображення пристроїв.

Кожний Ведений та Ведучий DPM1 повинні мати ідентифікатор виробника (Manufacturer ID). Це потрібно для того, щоб Ведучий міг ідентифікувати тип підключених Ведених без додаткових протокольних затрат. Ведучий порівнює ідентифікатор підключеного Веденого з указаним ідентифікатором у даних конфігурації. Тільки при їх рівності почнеться обмін даними в операційному режимі. Це гарантує помилку від хибної конфігурації.

Спеціальні ідентифікатори зарезервовані для профілів пристроїв (Profile ID):

- 9700₁₆ — 977F₁₆ — PA пристрої;
- 3A00₁₆ — 3AFF₁₆ — пристрої профілю PROFIdrive.



Контрольні запитання до розділу 7

1. Перерахуйте мережі PROFIBUS, які використовуються сьогодні? Яке призначення кожної з них? Охарактеризуйте їх у контексті моделі OSI.
2. На якому інтерфейсі, яке середовище передачі, яка бітова швидкість, топологія та який метод кодування використовуються для електричного з'єднання для PROFIBUS DP?
3. Як забезпечуються термінування, захисне зміщення, сегментація в PROFIBUS DP, побудованого на базі RS-485?
4. Які правила підключення вузлів до загальної шини PROFIBUS DP, побудованого на базі RS-485?
5. Який тип з'єднувача рекомендується використовувати для PROFIBUS DP, побудованого на базі RS-485? Прокоментуйте призначення кожного піну.
6. Які типи кабелів використовуються для PROFIBUS DP, побудованого на базі RS-485? Які вимоги до бітової швидкості, загальної довжини лінії зв'язку та довжини відгалужень при використанні різних кабелів?
7. Які додаткові засоби необхідно використати при підключенні пристроїв до шини, на бітових швидкостях вище 3Мбіт/с? Поясніть конструкцію з'єднувача, який використовується при таких швидкостях.
8. Як правильно виконати заземлення екранів кабелів?
9. На якій технології передачі на фізичному рівні базується мережа PROFIBUS PA? Розкажіть про принципи її функціонування.
10. Яке призначення сегментних відгалужувачів та сегментних з'єднувачів? Чим вони відрізняються?

11. Яка топологія, бітова швидкість та кабелі використовуються в PROFIBUS PA? Прокоментуйте принципову схему сегмента PROFIBUS PA.
12. Який метод доступу використовується в мережах PROFIBUS? Як адресуються вузли в мережі?
13. Які способи обміну використовуються в PROFIBUS DP? Які функції належать до базових PROFIBUS DP-V0?
14. Що таке циклічний обмін даними процесу? Які він функціонує?
15. Розкажіть про призначення та принципи функціонування команд SYNC/FREEZE.
16. Які Ви знаєте класи Ведучих вузлів PROFIBUS DP-V0? Які функції цим вузлам доступні?
17. Які складові циклу PROFIBUS DP у системі з одним Ведучим? Які функції виконуються в ациклічній частині? Навіщо потрібні DP-цикли з постійним часом та які їх складові?
18. Як функціонують діагностичні переривання Веденого в контексті функціонування DP-циклу?
19. Поясніть процедуру ініціалізації Ведучим свого Веденого.
20. Що таке консистентність даних процесу в PROFIBUS DP? Навіщо її визначати? Які обмеження на обсяг консистентних даних?
21. Які розширені функції обміну доступні в PROFIBUS DP-V1 та DP-V2?
22. Як функціонують сервіси ациклічного обміну даними процесу?
23. Як функціонують сервіси широкомовного обміну даними процесу?
24. Як функціонують сервіси ізохронного режиму обміну?
25. Прокоментуйте технічну структуру системи PROFIBUS. Що таке загальні прикладні профілі, специфічні прикладні профілі та системні профілі і як вони пов'язані?
26. Охарактеризуйте мережі PROFIBUS на фізичному рівні з різними варіантами реалізації.
27. Яке призначення і принципи функціонування профілю PROFIsafe?
28. Як функціонують системи на базі профілю Slave Redundancy?
29. Наведіть приклади специфічних профілів PROFIBUS.
30. Розкажіть про призначення та принципи побудови GSD-файлу.

CAN ПРОТОКОЛ

Протокол *CAN* (Controller Area Network) — одне із найбільш перспективних і популярних рішень у галузі промислових мереж. Поява його пов'язана з бурхливим розвитком рівня електроніки в автомобілях. Все більша кількість різнорідних датчиків та блоків управління привела до загрозливого наростання кількості проводок, що ускладнювало монтаж і надійність бортових систем управління. У другій половині 80-х років минулого століття фірма Robert Bosch GmbH (більш відома як BOSCH) запропонувала розв'язок цієї проблеми заміною всіх проводок двохпровідною цифровою послідовною шиною, до якої б підключалися різнорідні пристрої за допомогою спеціалізованого чіп-контролера. Результатом такої роботи стала специфікація на шину CAN-версії 1.1. Пізніше деякі недоліки, зокрема з питань синхронізації, були виправлені і з урахуванням цих виправлень вийшла *CAN Специфікація 1.2*.

Протокол CAN описаний у контексті канального та частини фізичного рівнів моделі OSI. Їх функції реалізуються у вигляді як окремих мікросхем CAN-контролера, так і вбудованих в готові мікроконтролери CAN-шини. Сьогодні на базі CAN успішно функціонують багато промислових мереж, які доповнюють CAN додатковими протоколами верхніх рівнів. Серед найбільш популярних мереж, які базуються на CAN, є мережі CANopen, DeviceNet, CAN Kingdom, SDS (Smart Distributed System). Мікроконтролери CAN в поєднанні з різними трансиверами знайшли своє використання і без наявних описаних функцій прикладного рівня. Мережні рішення на базі CAN прийнято називати CAN-шинами. Для координації зусиль виробників, розробників і користувачів CAN-систем і технологій створена некомерційна організація *CiA* (CAN in Automation), яка налічує більше 300 компаній.

Згідно зі специфікацією версії 1.2, канальний рівень CAN базується на широкомовній передачі, що використовує метод адресації, орієнтований на ідентифікатор повідомлення (Виробник-Споживач-msg). Тобто, кожне повідомлення в мережі має унікальний 11-бітний *Ідентифікатор* (IDENTIFIER), який засто-

совують для фільтрації прийнятих повідомлень кожним вузлом. CAN послуговується множинним методом доступу CSMA/CA з використанням поля арбітражу. Для реалізації такої схеми необхідно, щоб на шині один логічний сигнал фізично домінував над іншим, тому вимогою CAN до фізичного рівня є наявність домінантного та рецесивного логічних рівнів сигналів. Послідовність домінантних та рецесивних бітів визначає пріоритет повідомлення, який задається відправником у полі арбітражу кадру, а саме в Ідентифікаторі повідомлення. Детальніше про модель адресації Виробник-Споживач-msg та метод доступу CSMA/CA можна ознайомитись у 2-му розділі.

8.1. CAN-стандарти та їх опис у контексті моделі OSI

Протокол CAN набув визнання вже після випуску першої специфікації. Однак, у зв'язку з усе більшим ростом потреб, виникла необхідність у розширенні одного з полів кадру (11-бітного Ідентифікатора повідомлення). Для сумісності версій нова *CAN Специфікація 2.0* дозволяє використовувати як старий формат кадру з 11-бітним Ідентифікатором, який отримав назву «стандартний формат», так і новий — «розширений формат» з 29-бітним Ідентифікатором повідомлення. Нова специфікація складається з 2-х частин:

- частина «А», яка описує CAN-формат повідомлення такий самий, як у CAN Специфікації 1.2;

- частина «В», яка описує як стандартний, так і розширений формат.

Для сумісності реалізації CAN з новою специфікацією достатньо, щоб вона була сумісною з Частиною «А» або Частиною «В». Крім того, вузли, реалізовані на основі специфікації CAN 1.2 або CAN 2.0 Частина «А», можуть зв'язуватися з вузлами, реалізованими на основі CAN 2.0 Частина «В», використовуючи стандартний формат повідомлення. Можливе одночасне функціонування обох форматів повідомлення на одній мережі, за рахунок наявності резервного біта в стандартному форматі кадру (розглянуто нижче).

CAN Специфікацію 2.0 також називають *BOSH CAN 2.0 A/B*. В контексті моделі OSI вона описує повністю каналний рівень та частину фізичного (рис.8.1). На фізичному рівні CAN 2.0 описує тільки загальні правила синхронізації, кодування бітів та вимоги до швидкодії. Однак, фізичний спосіб передачі бітів, середовище передачі, роз'єми та характеристики трансиверів залишаються поза увагою специфікації. Таким чином, CAN визначений повністю на підрівні MAC, частково на LLC та фізичному рівні, а також він пред'являє певні вимоги щодо реалізації інших рівнів (фізичного та прикладного).

У 1993 році CAN закріпився у міжнародному стандарті *ISO 11898* (CAN High Speed Transceiver and Data Link Layer). Цей стандарт, на відміну від BOSH CAN 2.0 A/B, додатково описує вимоги до високошвидкісних трансиверів та середовища передачі даних (рис.8.1.). Саме ISO 11898 використовується у багатьох мережах, які базуються на CAN (наприклад, CANOpen, DeviceNet). Як правило, під поняттям «CAN-шина» мають на увазі мережне рішення на базі ISO 11898.

каналний	MAC	фільтрація повідомлень перевантаження повідомлень управління відтворенням	
	LLC	інкапсуляція/декапсуляція даних кодування кадру управління доступом знаходження/сигналізація помилок підтвердження отримання даних	
фізичний		кодування/декодування бітів синхронізація управління тривалістю біта	
	не описаний	характеристики приймачів/передавачів фізичний рівень сигналів кабелі, роз'єми	
		BOSH CAN 2.0 A/B	ISO 11898

Рис. 8.1. Стандарти BOSH CAN та ISO 11898 у контексті OSI-моделі

8.2. Організація каналного рівня

8.2.1. Комунікаційні сервіси

Протокол CAN забезпечує два комунікаційні сервіси: це сервіс *Запису Об'єкта* (Write Object) та сервіс *Читання Об'єкта* (Read Object). Для обміну даними визначені два типи кадрів — *Кадр Даних* (DATA FRAME) та *Дистанційний Кадр* (REMOTE FRAME), які можуть вміщувати до 8 байтів даних повідомлення.

При записі об'єкта від одного вузла (Виробника) передається Кадр Даних до одного або більше приймаючих вузлів (Споживачів) (рис. 8.2).

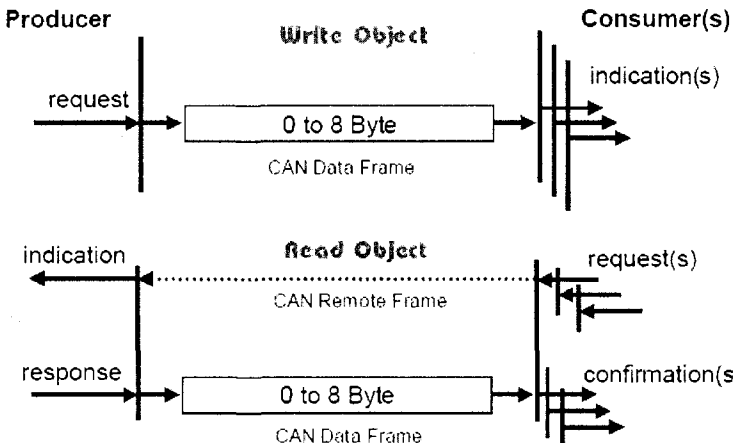


Рис. 8.2. Обмін даними між вузлами мережі CAN

Сервіс Читання Об'єкта ініціюється одним або декількома Споживачами. Для цього один з цих вузлів передає Дистанційний Кадр, за яким вузол Виробник відповідає Кадром Даних. У цьому випадку потрібний Кадр Даних та Дистанційний Кадр будуть мати ідентичні Ідентифікатори.

Широкомовність і фільтрація дає змогу прийняти Кадр Даних на всіх необхідних вузлах. За одночасної передачі по шині Кадру Даних та Дистанційного Кадру з тим же Ідентифікатором, пріоритет має (виграє арбітраж) Кадр Даних.

8.2.2. Формати кадрів обміну даними

8.2.2.1. Типи кадрів

Повідомлення виявляються і управляються чотирма різними типами кадрів:

- Кадр Даних (DATA FRAME) — несе дані від передавача до приймача;
- Дистанційний Кадр (REMOTE FRAME) — проводить запит на передачу Кадру Даних з тим же Ідентифікатором;
- Кадр Помилки (ERROR FRAME) — передається будь-яким пристроєм при виявленні помилки;
- Кадр Перевантаження (OVERLOAD FRAME) — використовується для попередження про необхідність затримки до передачі наступного кадру.

Ми вже зазначали, що перші два типи кадрів служать для обміну даними і можуть бути використані як у стандартному, так і в розширеному форматі. Інші два — призначені для управління роботою шини в нештатних ситуаціях. Мінімальна пауза між кадрами називається Міжкадровим Простором (INTERFRAME SPACE).

Кожний кадр складається з бітових полів. Розглянемо формати кадрів для обміну даними, починаючи з Кадру Даних (рис. 8.3).

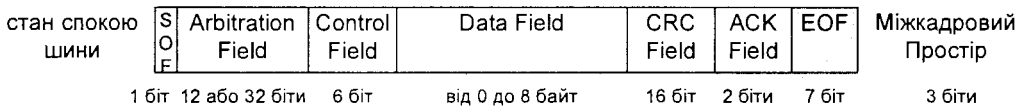


Рис. 8.3. Формат Кадру Даних

8.2.2.2. Формат Кадру Даних.

Початок кадру (SOF-Start Of Frame) складається з одного домінуючого біта, який призначений для синхронізації приймачів з передавачем. Будь-якій станції дозволяється займати шину для передачі, якщо вона в стані очікування, тобто в період Міжкадрового Простору. Всі вузли синхронізуються за переднім фронтом сигналу біта SOF. Функція даного біта аналогічна стартовому біту при символній передачі.

Поле Арбітражу (ARBITRATION FIELD) призначене для визначення пріоритету повідомлення/кадру. Склад даного поля різний для стандартного та розширеного форматів (рис. 8.4). В стандартному форматі поле має 11-бітний Ідентифікатор та RTR-біт. У розширеному форматі поле має 29-бітний Ідентифікатор, SRR-біт, IDE-біт і RTR-біт. Біти Ідентифікатору позначаються ID-28...ID-0. У попередніх специфікаціях (1.0-1.2) біт IDE стандартного формату позначався як r1.

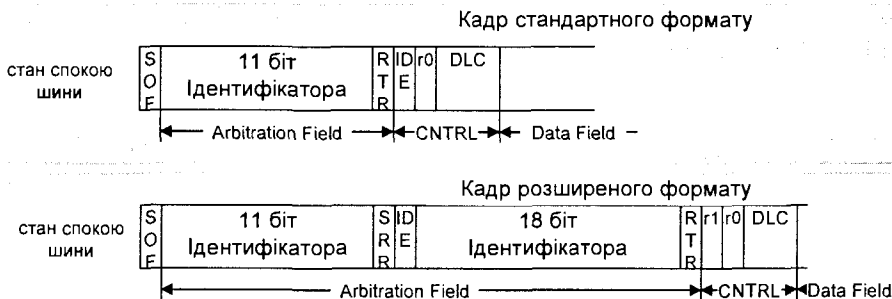


Рис. 8.4. Кадр стандартного та розширеного формату

Для стандартного формату біти Ідентифікатора нумеруються за порядком від ID-28(найбільш значимий) до ID-18(найменш значимий), з яких 7 перших (ID-28–ID-22) не можуть бути рецесивними. В розширеному форматі Ідентифікатор має дві секції: 11-бітну BaseID та 18-бітну ExtendedID. BaseID (Базовий ID) — це еквівалент Ідентифікатора в стандартному форматі, він визначає базовий пріоритет у розширеному форматі. ExtendedID (Розширений ID) складається з 18 бітів, які позначаються, починаючи з ID-17, закінчуючи ID-0.

Біт RTR (Remote Transmission Request BIT) вказує на тип кадру. Для Кадру Даних даний біт буде домінантним, для Дистанційного Кадру — рецесивним. Таким чином, досягається вищий пріоритет кадру з даними над кадром із запитом на передачу цих даних при тому самому Ідентифікаторі. Однак, слід відмітити, що RTR у стандартному форматі йде раніше, ніж при передачі розширеного формату кадру, хоч в обох випадках — після бітів Ідентифікатора.

Біт SRR (Substitute Remote Request BIT) існує тільки в розширеному форматі і завжди передається в рецесивному стані. Оскільки його позиція співпадає з позицією біта RTR у стандартному форматі, то за одночасної передачі кадрів стандартного та розширеного форматів з Ідентифікатором стандартного кадру, рівним BaseID розширеного, стандартний кадр буде вигравати арбітраж.

Біт IDE (Identifier Extension Bit) — для стандартного формату передається домінуючим, для розширеного — рецесивним.

Поле Управління (CONTROL FIELD) складається із 6 бітів (рис. 8.5). Чотири з них (DLC3–DLC0) називаються Кодом Довжини Даних (DATA LENGTH CODE) і вказують на кількість байтів у Полі Даних. Інші 2 біти r1 і r0 зарезервовані і відправляються домінантними. Однак, для стандартного формату кадру на місці біта r1 стоїть біт IDE, призначення якого описано вище.

Поле Даних (DATA FIELD) — складається з байтів від 0 до 8, які передаються в Кадрі Даних.

Поле CRC — використовується для перевірки контрольної суми. Воно складається з 15-бітової CRC послідовності, після якої йде рецесивний біт під назвою

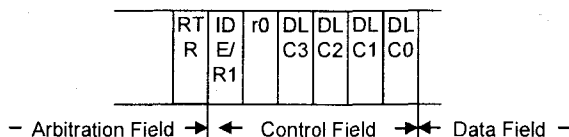


Рис. 8.5. Формат ПОЛЯ УПРАВЛІННЯ

Розмежувач CRC (CRC DELIMITER), який служить для розмежування між послідовністю CRC та полем ACK.

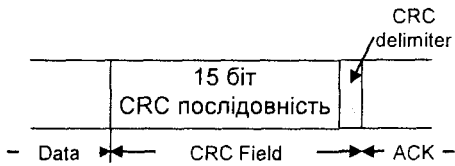


Рис. 8.6. Формат ПОЛЯ CRC

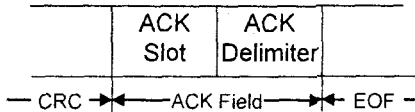


Рис. 8.7. Формат Поля ACK

Поле ACK складається з двох бітів: Слота ACK (ACK SLOT) та Розмежувача ACK (ACK DELIMITER). Обидва біти передаються рецесивними, а отримувачі, які одержали повідомлення правильно (перевіривши CRC), замінюють Слот ACK на домінуючий біт. Таким чином, відправник контролює правильну доставку повідомлення.

Кінець кадру (EOF — End Of Frame) — це послідовність із семи рецесивних бітів, які вказують на кінець Кадру Даних або Дистанційного Кадру.

8.2.2.3. Формат Дистанційного Кадру. Дистанційний Кадр сформований із шести полів: Початку Кадру, Поля Арбітражу, Поля Управління, Поля CRC, Поля ACK, Кінця Кадру (рис. 8.8).

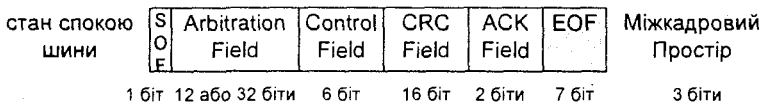


Рис. 8.8. Формат Дистанційного Кадру

На відміну від Кадру Даних біт RTR Дистанційного Кадру є рецесивним, а отже при одночасній передачі Кадру Даних та Дистанційного з тим самим Ідентифікатором арбітраж виграє Кадр Даних.

Слід зазначити, що в Дистанційному Кадрі немає Поля Даних, оскільки цей тип кадру є тільки ініціатором передачі даних. Для полів Початок Кадру, Поля Арбітражу, Поля Управління, Поля Даних, Послідовності CRC справедливе правило бітового начинення. Всякий раз, коли передавач виявляє у бітовому потоці п'ять послідовних бітів однакової величини, які повинні бути передані, він автоматично включає допоміжний біт у фактичний переданий бітовий потік, який обернений до них за рівнем сигналу. Для інших полів, а також для Кадрів Помилки та Кадрів Перевантаження це правило не діє.

8.2.3. Обробка помилок та боротьба з дефектами

8.2.3.1. Типи помилок. У CAN виділяються 5 різних типів помилок, які не виключають одна одну:

- бітова помилка;
- помилка начинення;
- помилка CRC;

- помилка форми;
- помилка підтвердження.

Бітова помилка (bit error) виникає, коли вузол-відправник, прослуховуючи шину при передачі, виявить інший рівень біта, ніж він передавав. Винятком є Поле Арбітражу, Поле АСК та Пасивний Прапор Помилки, в яких вияв домінуючого біта замість рецесивного не є помилкою.

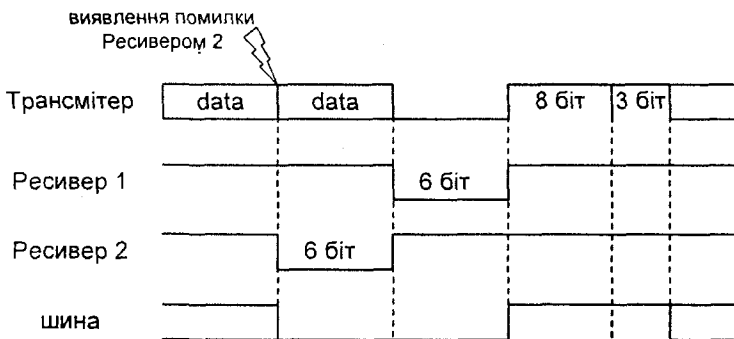
Помилка начинення (stuff error) є результатом вияву послідовності з 6 бітів однакового логічного рівня в тих полях, де діє правило бітової начинки.

Помилка CRC (CRC error), як і в інших протоколах з перевіркою контрольної суми, виникає при відмінності розрахованої CRC з отриманим значенням у CRC Послідовності.

Помилка форми (form error) виникає, коли фіксовані бітові поля мають один або декілька незаконних бітів (домінантний біт у ділянках Розмежувач CRC, Розмежувач АСК або EOF).

Помилка підтвердження (acknowledgment error) завжди виникає, коли передавач не виявляє в Слоті АСК домінантного біта, що говорить про відсутність підтвердження з боку отримувача (-ів).

8.2.3.2. Механізм глобалізації помилки. При виявленні одним із вузлів помилки передача повинна припинитися, навіть якщо всі інші вузли помилок не виявили. Це досягається за рахунок *глобалізації помилки*, тобто штучного спотворення бітів, які передаються, задля виявлення помилок усіма вузлами. Тобто, якщо помилку виявив один із вузлів (локальна помилка), він повинен примусити всі інші вузли виявити цю помилку. Для цього він передає послідовність із 6 домінантних бітів (Прапор Помилки), що приводить до вияву всіма іншими вузлами помилки начинення. Ті, в свою чергу, виявивши локальну помилку, теж починають передавати послідовність із 6 бітів (рис. 8.9).



Після виявлення помилки Ресивер 2 починає передавати 6 домінантних бітів

Рис. 8.9. Механізм глобалізації локальної помилки

Таким чином, при виникненні помилки на шині з'явиться від 6 до 12 домінантних бітів. Ця послідовність разом з допоміжними бітовими полями формує Кадр Помилки (рис. 8.10).

Після того, як вузол передає останній біт Прапора Помилки, він посилає рецесивний біт, з контролем його стану на шині. Як тільки цей стан з'явиться на шині (тобто всі вузли передали свій Прапор Помилки), кожний з них передає ще 7 рецесивних бітів. Ця послідовність із 8-ми рецесивних бітів називається Розмежувачем Помилки (ERROR DELIMITER). Після Кадру Помилки, дочекавшись паузи із 3-х бітів (Міжкадрового Простору) передавач перерваного кадру повторює його передачу знову.

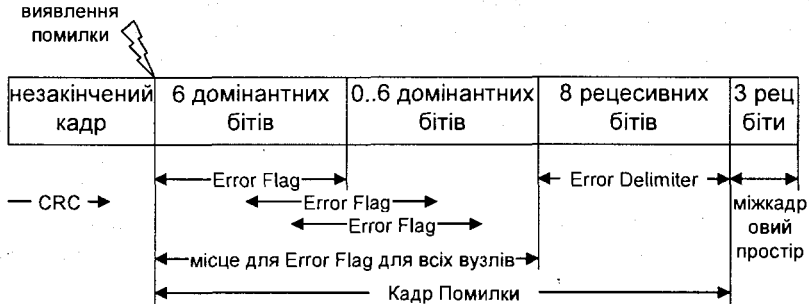


Рис. 8.10. Формат Кадру Помилки

8.2.3.3. Механізми самодіагностики. Глобалізація помилки могла б привести до абсолютно зворотного ефекту у випадку виходу з ладу або збоїв у роботі одного з вузлів. Наприклад, якщо вузол починає кожен раз посилати Прапор Помилки, у зв'язку з певними дефектами приймача, шина зациклиться на передачі одного й того самого інформаційного кадру. Однак, у CAN визначені механізми для локалізації дефектних вузлів, які не допускають «зависання» мережі. Кожен контролер CAN оснащений засобами самоконтролю, які переводять його в один із 3-х станів:

- активним до помилок (Active Error);
- пасивним до помилок (Passive Error);
- неактивний на шині (Bus Off).

У нормальному стані вузол перебуває в активному до помилок стані і при виникненні помилки поводить себе так, як описано вище.

Якщо вузол з певних причин (описано нижче) перейшов у пасивний до помилок стан, то при передачі Прапора Помилки (див. рис. 8.11) замість домінуючих бітів він посилає рецесивні. Це значить, що у випадку виникнення локальної помилки при прийомі кадру на даному вузлі він не зможе провести її глобалізацію, а отже, всі інші вузли на шині помилку не виявлять.

Втім, при передачі кадру пасивний до помилок вузол при виявленні помилки сам перериває свій власний кадр послідовністю із 6 рецесивних бітів. Це викличе помилку бітового начиння в інших вузлах, що призведе до глобалізації помилки. З цієї причини даний вузол знову повторить передачу кадру. Щоб не зациклити шину даним кадром (адже дефектний вузол може сам спотворювати власну передачу), введена допоміжна пауза (Призупинка Передачі), під час якої інші вузли зможуть почати передачу свого кадру. Неактивний на шині вузол взагалі не бере участі в передачі.

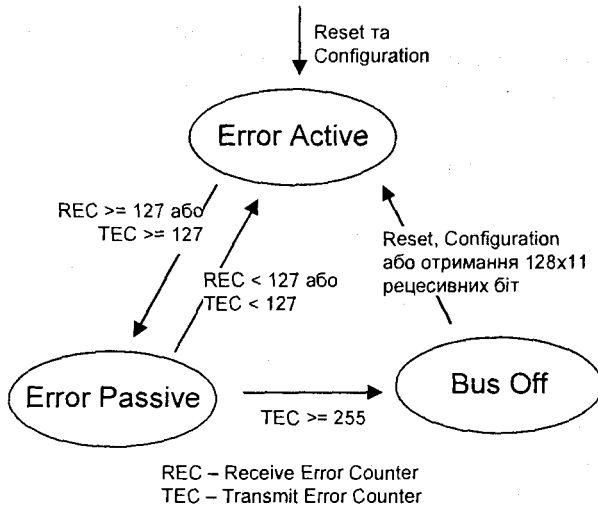


Рис. 8.11. Управління станом вузла CAN

Щоб управляти своїм станом кожен вузол має лічильники для підрахунку помилок приймання REC (Receive Error Counter) та відправки TEC (Transmit Error Counter). На діаграмі на рис. 8.11 видно, як вузол переходить зі стану в стан, залежно від значення цих лічильників. У специфікаціях CAN визначені правила збільшення/зменшення значень REC і TEC.

8.3. Реалізація фізичного рівня

Реалізація фізичного рівня для CAN-основаних мереж поділена на 3 підрівні, які описують такі характеристики:

- PLS (Physical Signaling): кодування/декодування бітів; управління тривалістю біта; синхронізація.

- MDA (Physical Medium Attachment): характеристики трансмітерів;

- PMI (Medium Dependent Interface): характеристики кабелів та роз'ємів.

Рівень PLS, описаний у специфікації CAN 2.0, реалізується готовими мікросхемами контролера CAN. Рівні MDA і PMI є предметом інших промислових міжнародних чи національних стандартів або розробляється самостійно за власними специфікаціями. Найбільш загальним є стандарт ISO 11898, який визначає високошвидкісний трансивер для мереж CAN. Спочатку розглянемо механізми фізичного рівня, які стосуються підрівня PLS і описані в специфікації CAN, нижче розглянемо найбільш популярні способи реалізації інших підрівнів.

Кодування бітів проводиться методом NRZ. Для синхронізації вузлів необхідні фронти, за якими генератор приймача підлаштується під генератор передавача. Для цього CAN-кадр починається зі спеціального стартового біта (поле SOF), який переводить шину з рецесивного стану в домінуючий. Такий механізм у CAN називається жорсткою синхронізацією (*hard synchronization*). Далі, під

час передачі інших бітів з різною полярністю, приймач також підлаштовується під передавач при зміні сигналу з рецесивного на домінуючий стан, використовуючи механізм ресинхронізації (*resynchronization*).

8.4. CAN-сумісні стандарти

CAN Специфікація 2.0 є основою для багатьох популярних мереж, тому не описує рівні, які містяться вище канального, та частину фізичного. Так, наприклад, є декілька офіційних і промислових стандартів, які визначають підрівень MDA:

- CAN High-Speed (ISO 11898-2) — високошвидкісний;
- CAN Low-Speed (ISO 11519-1) — низькошвидкісний;
- Fault-Tolerant Transceivers (ISO 11898-3) — стійкий до помилок, використовують в автомобільній бортовій електроніці;
- Truck/Trailer Transceiver (ISO 11992) — використовується німецьким стандартом LBS(DIN 9684) для сільськогосподарських машин;
- Single-Wire (SAE 2411) — однопровідний зв'язок;
- Fiber Optical Transmission — волоконно-оптична передача;
- Wire-Less Transmission — безпроводна передача;
- Power-Supply Transmission — живлення пристроїв по шині.

Протоколи верхніх рівнів описані в стандартах мереж CANopen, CAN Kingdom, DeviceNet, SDS (Smart Distributed System) та інших. Багато з них базуються на стандарті ISO 11898, тому його розглянемо детальніше.

8.4.1. Стандарт ISO 11898

8.4.1.1. Загальний огляд ISO 11898. Стандарт *ISO/IS 11898* (Road vehicles. Interchange of digital information. Controller area network (CAN)) складається з п'яти частин:

- ISO 11898-1: описує канальний рівень та частину фізичного аналогічно як в CAN 2.0 частини А;
- ISO 11898-2: стандарт високошвидкісного доступу до шини, описує трансивери, параметри передачі даних та фізичне середовище для реалізації на фізичному рівні;
- ISO 11898-3: стандарт низькошвидкісного, стійкого до помилок доступу до шини, описує трансивери, параметри передачі даних та фізичне середовище для реалізації на фізичному рівні;
- ISO 11898-4: описує обмін даними за часовим тригером.

На відміну від CAN 2.0 частини А, в ISO 11898-1 значення поля DLC може бути більш ніж 8 (до 15). Однак, у специфікації від BOSCH не визначена жодна з помилок у такій ситуації, що значить про сумісність у реалізації обох стандартів. Додатково визначений новий необов'язковий режим, в якому може перебувати вузол: моніторинг шини (Bus monitoring mode). В цьому режимі вузол тільки слідкує за передачею, однак не передає дані сам.

Для систем з регламентованим часом відновлення даних може знадобитися допоміжна функція, яка описана в ISO 11898-4 (TTC — Time Triggered Communication). Ця функція визначає умови для генерації повідомлення кожним вузлом через певні інтервали часу, що виключає одночасний доступ до шини декількох пристроїв і дозволяє проводити так зване «планування кадрів». Для реалізації такого функціонування необхідне однакове «проходження часу» на кожному пристрої, що досягається за рахунок синхронізації внутрішніх лічильників за бітом SOF або за точкою синхронізації останнього біта поля EOF. На основі TTC полегшується створення глобальної системи часу на реалізації верхніх рівнів OSI для мереж, оснований на CAN. Однак, використання режиму TTC заперечує послуговування Кадрами Перевантаження та Кадрами Помилки і повторення передачі кадру у випадку помилки. Реалізовується функція TTC як підрівень між LLC та MAC.

8.4.1.2. ISO 11898-2. Стандарт ISO 11898-2 описує високошвидкісні трансивери CAN та вимоги до фізичного середовища. Кожен вузол підключається до шини, яка складається з двох ліній, які називаються CAN_H і CAN_L. Забезпечують мінімізацію ефекту відбиття сигналу в кінцях ліній з використанням узгоджувальних резисторів номіналом 120 Ом (рис. 8.12).

Використовується диференційний спосіб передачі за напругою: вузлами ідентифікуються рецесивний біт, якщо різниця потенціалів між CAN_H і CAN_L не перевищує 0.5 В, домінантний — якщо більше ніж 0.9 В. Номінальна напруга в домінантному стані — 3.5 В для лінії CAN_H і 1.5 В для CAN_L (рис. 8.13).

ISO 11898-2 сумісні вузли являють собою мікроконтролер (mC) та CAN-контролер, які підключені до трансивера, що живиться 5 В (рис. 8.14).

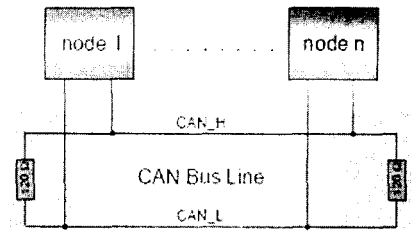


Рис. 8.12. Підключення вузлів до CAN мережі по ISO 11898

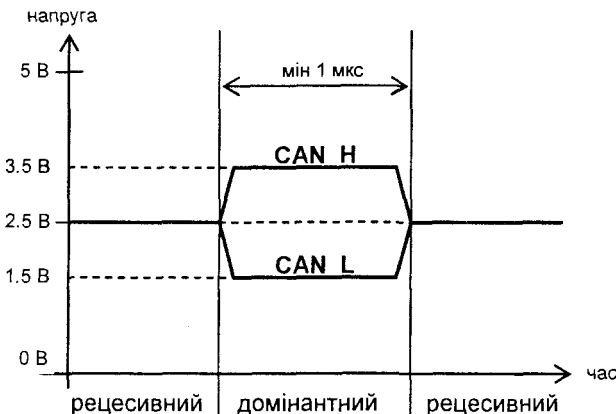


Рис. 8.13. Домінантний та рецесивний біти в ISO 11898-2 Перевантаження

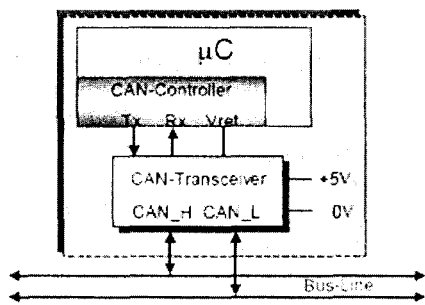


Рис. 8.14. Конструкція CAN-вузлів ISO 11898

Максимально припустима довжина лінії зв'язку визначається різними фізичними обмеженнями, пов'язаними з характеристиками вузлів та кабелю. Використовуючи трансивер згідно ISO 11898-2 і високошвидкісну опторозв'язку, можна досягнути максимальної довжини 9 метрів при бітій швидкості 1 Мбіт/с. При менших швидкостях цю довжину можна істотно збільшити. В таблиці 8.1 показана довжина лінії, яка може бути досягнута при використанні ISO 11898 сумісного трансивера зі стандартним шинним кабелем, без урахування опторозв'язки.

Таблиця 8.1

ПРИПУСТИМА ДОВЖИНА ЛІНІЇ ЗВ'ЯЗКУ

Бітова швидкість	Довжина лінії зв'язку
1 Мбіт/с	30 м
800 кбіт/с	50 м
500 кбіт/с	100 м
250 кбіт/с	250 м
125 кбіт/с	500 м
62,5 кбіт/с	1000 м
20 кбіт/с	2500 м
10 кбіт/с	5000 м

Враховуючи велику популярність ISO 11898-2 серед CAN-сумісних промислових мереж, визначимо його загальні характеристики.

- сумісний з BOSH CAN 2.0 A;
- використовується симетрична передача за напругою між CAN_H та CAN_L;
- середовище передачі — вита пара з характеристичним хвильовим імпедансом 120 Ом;
- бітова швидкість вибирається з ряду від 10 кБіт/с до 1 Мбіт/с (таб.8.1); підтримка 20 Кбіт/с — обов'язкова;
- топологія — шина, з короткими відгалуженнями (до 0,3 м при 1Мбіт/с);
- максимальна довжина лінії зв'язку — до 1000 м;
- два обов'язкові термінальні резистори 120 Ом (108 Ом — 132 Ом) на кінцях лінії;
- максимум 64 пристрої на один сегмент.

8.4.2. Протоколи прикладного рівня для мереж CAN

Необхідність у використанні функцій прикладного рівня для мереж CAN обумовила ряд мережних рішень, деякі з яких дуже схожі між собою. Серед найбільш популярних протоколів прикладного рівня, які базуються на CAN, є CANopen, DeviceNet, CAN Kingdom, SDS (Smart Distributed System). Сьогодні вони стандартизовані в міжнародних стандартах. Зокрема, в EN 50325 описані такі промислові мережі:

- EN 50325-1 2002 Industrial communications subsystems based on ISO 11898 (CAN) for controller device interfaces — part 1: general requirements;
- EN 50325-2 2000 Part 2: DeviceNet;
- EN 50325-3 2000 Part 3: Smart Distributed System;
- EN 50325-4 2002 Part 4: CANopen.

Крім того, в серії стандартів MEK Industrial Communication Networks (IEC 61158, IEC 61784, IEC 61918) теж стандартизовані деякі рішення, зокрема DeviceNet (у складі сімейства CIP). Загальна картина прикладних рішень на базі CAN показана на рис. 8.15. Більшість з них базується на ISO 11898.

Application Profile		CiA Application Profiles		SAE J1939-based Application Profiles
Device Profile		ODVA Device Profiles	CiA Device Profiles	
Application Layer		SDS EN 50325-3	DeviceNet EN 50325-2	CANopen EN 50325-4
Data Link Layer	CAN 2.0A	ISO 11898-1 (11-bit ID)	ISO 11898-1 (11-bit and 29-bit ID)	ISO 11898-1 (29-bit ID)
Physical Layer	RS-485	ISO 11898-2		ISO 11898-2

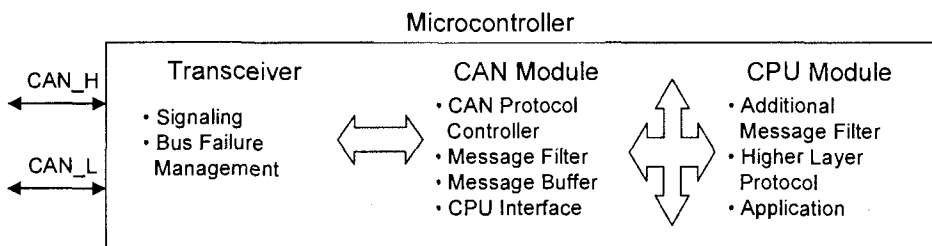
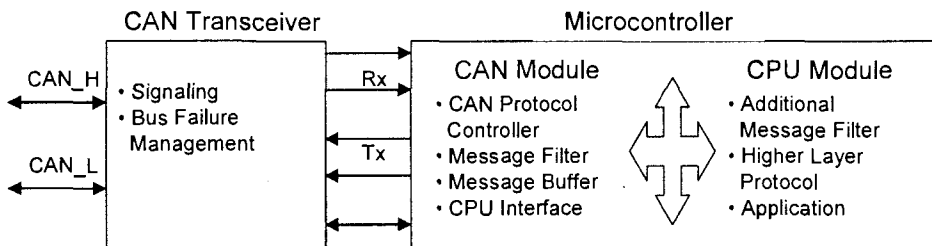
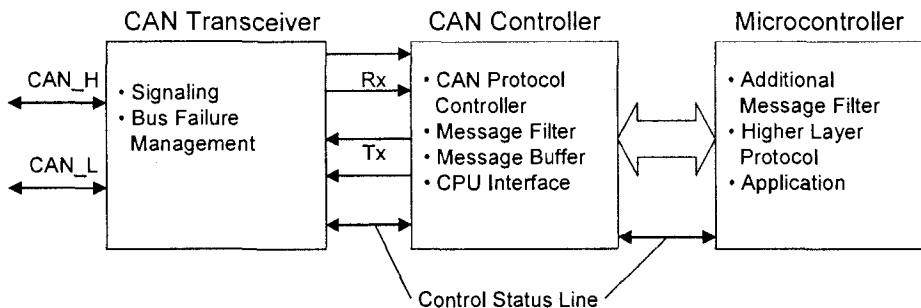
Рис. 8.15. Мережі на базі технології CAN

8.5. Реалізація CAN

Декілька компаній реалізують протоколи CAN у вигляді готових мікросхем, які мають забезпечити основні послуги зв'язку: передачу даних, запит на дані, слідування за помилками та їх ідентифікацію. Всі існуючі реалізації використовують контролери CAN, функції та інтерфейс яких сумісний з моделлю реалізації від BOSCH. Відмінності полягають в ацептованій фільтрації, можливості буферизації кадрів та використання деяких додаткових функцій.

Є декілька варіантів реалізації всієї системи обміну з використанням протоколу CAN (рис. 8.16):

- з окремо виділеним CAN-контролером;
- з інтегрованим CAN-контролером у мікроконтролер;
- з використанням єдиного чіпсету разом з трансивером.



Архітектура CAN Controller

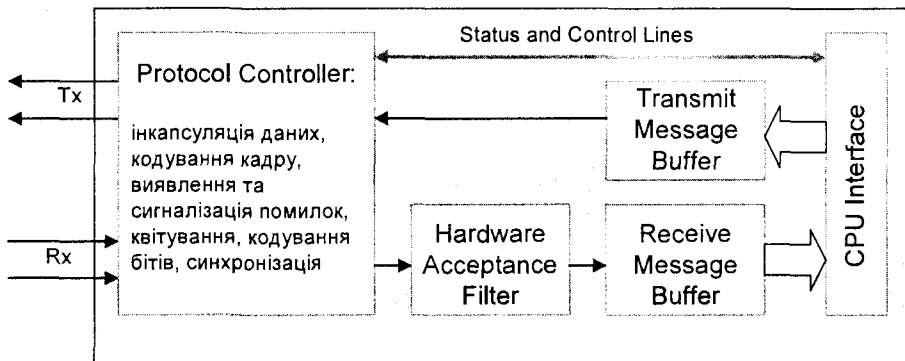


Рис. 8.16. Різні варіанти реалізації CAN-вузлів

Перевага першого способу реалізації полягає у багатофункціональності та гнучкості. Такий підхід має місце у випадках використання протоколу верхніх рівнів, які базуються на різних реалізаціях каналного та фізичного рівнів, або, навпаки, коли один і той самий контролер CAN використовується для різних мереж (мультипротокольна реалізація, шлюзи і т.д.).

Інтеграція CAN контролера разом з мікроконтролером збільшує швидкість взаємодії між складовими та зменшує навантаження на процесори. Ціна такого рішення, порівняно з попереднім способом, як правило, нижча, а надійність — вища.

Для систем розподіленого вводу/виводу більш доцільним є використання єдиного мікроконтролера з реалізацією протоколів усіх рівнів. Недоліком таких систем є те, що неможлива комбінація різноманітних технологій тобто гнучкості.

Окрім контролера протоколу, кожен CAN-чип має вбудований ацептований фільтр для фільтрації повідомлень, буфери для кадрів даних, що передаються та відправляються. Реалізація цих функцій, а також інтерфейсу з CPU можуть відрізнятися залежно від реалізації CAN-чипа.

CAN — дуже цікаве рішення в галузі промислових мереж. З одного боку, він надзвичайно складний у функціонуванні (і в розумінні), з іншого — надійний і ефективний. Окрім того, наявність великої кількості компонентів CAN (від фірм Intel, Philips, Siemens, Motorola) робить систему недорогою.



Контрольні запитання до розділу 8

1. На яких рівнях моделі OSI описаний протокол CAN? Якими засобами необхідно доповнити мікроконтролер CAN для можливої побудови мережі?
2. Який метод доступу до шини та модель адресації використовується в CAN? Розкажіть про їх принципи функціонування.
3. Що таке Ідентифікатор повідомлення і для чого він використовується в CAN-шинах?
4. Які CAN-стандарти Ви знаєте і чим вони відрізняються?
5. Які типи комунікаційних сервісів використовуються в протоколі CAN? Якими типами кадрів забезпечується передача даних?
6. Як реалізуються функції Запису Об'єкта та Читання Об'єкта?
7. Якими типами кадрів забезпечується ідентифікація помилки та затримка передачі наступного кадру?
8. За допомогою якого поля кадру даних забезпечується синхронізація приймача з передавачем?
9. Розкажіть про призначення Поля Арбітражу в кадрі даних. З яких частин складається це поле? Які типи Ідентифікаторів Ви знаєте? Як визначається пріоритетність Кадру Даних над Дистанційним Кадром?
10. Який кадр буде вигравати арбітраж — стандартний чи розширений? Поясніть чому?
11. Як контролюється правильність доставки даних отримувачем? Як контролюється правильність доставки даних відправником?
12. Поясніть, чому Дистанційний Кадр не має Поля Даних?
13. Які типи помилок виявляються в CAN?

14. Що таке глобалізація помилки і навіщо вона потрібна? Розкажіть про механізм функціонування глобалізації помилки.
15. Яким чином у CAN реалізовується захист шини від роботи несправних вузлів? В яких станах може перебувати вузол CAN з точки зору реакцій на помилки?
16. Які функції описує CAN на фізичному рівні? Які функції необхідно додатково реалізувати на фізичному рівні для мереж, які базуються на протоколі CAN?
17. Які механізми використовуються в CAN для синхронізації трансмітерів між собою?
18. Які стандарти, що базуються на CAN та описують характеристики трансмітерів Ви знаєте?
19. Що описує стандарт ISO 11898?
20. Розкажіть про основні характеристики CAN-шини на базі ISO 11898-2: спосіб передачі бітів, топологія, правила термінування, максимальна довжина ліній зв'язку та відгалужень, бітова швидкість, максимальна кількість пристроїв.
21. Які протоколи верхніх рівнів базуються на CAN-протоколі?
22. Які варіанти реалізації вузлів CAN Ви можете назвати?

CANOpen

CANOpen — це мережна система, яка базується на послідовній шині CAN. Сама назва «CANOpen» засвідчує про відкритий стандарт із самого початку його створення, який розроблений і підтримується некомерційною організацією *CiA*. Потужність рішень протоколу CAN задіяна в зручних для використання сервісах прикладного рівня CANOpen. CANOpen підтримує передачу даних процесу в реальному часі, циклічний та ациклічний зв'язок, забезпечує можливість синхронної передачі та відмінний захист від помилок.

9.1. CANOpen у контексті моделі OSI

CANOpen є продуктом організації *CiA* і описується серією стандартів та специфікацій, які розвиваються і доповнюються. Розглянемо, що являє собою дана мережа в контексті моделі OSI (рис. 9.1).

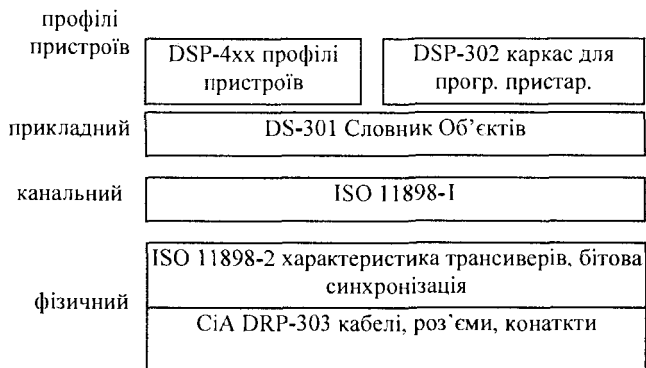


Рис. 9.1. CANOpen у контексті OSI-моделі

На фізичному та канальному рівнях CANOpen базується на CAN-контролері та високошвидкісного трансиверу, які визначені ISO 11898 у частинах 1 та 2. Додатково до них вказані вимоги до бітової синхронізації, характеристики кабелів та роз'ємів. Нижні рівні доповнені протоколом прикладного рівня, комунікаційними профілями та Каркасами (Frameworks) для програмованих пристроїв. Для спрощення механізму обміну з різнорідними пристроями в CANOpen використовується сімейство різнорідних стандартизованих профілів. Таким чином, мережа CANOpen описується системою стандартів:

- ISO 11898-1 та ISO 11898-2 фізичний та канальний рівень;
- CiA DRP-303-1 — опис рекомендованих характеристик кабелів і рознімів;
- CiA DRP-303-2 — визначення одиниць CI та префіксів;
- CiA DS-301 — прикладний рівень та комунікаційний профіль;
- CiA DSP-302 — каркас для Програмованих пристроїв;
- CiA DS-4XX — множина профілів пристроїв.

9.2. Особливості реалізації фізичного рівня

Фізичний рівень, як видно з рис. 9.1, базується на ISO 11898-2 з додатковими вимогами, які стосуються формування біта, а також рекомендацій з використання кабелів, роз'ємів та їх контактів.

Згідно з ISO 11898-2, у мережі використовується диференційна передача напруги за двома проводами CAN_H та CAN_L відносно загальної землі CAN_GND (див. розділ 8). Для надійної роботи на шині з довжиною більш ніж 200 метрів рекомендують використовувати опторозв'язок, а більш ніж 1 км — мости та повторювачі.

Крім правил формування бітів у CANOpen, додатково визначені рекомендовані кабелі, роз'єми та їх контакти. Стандарт CiA DRP-303-1(CiA Draft Recommendation) рекомендує використовувати кабель із хвильовим опором 120 Ом та з характеристиками, наведеними у табл. 9.1, для шини, яка має менш ніж 64 вузли.

Таблиця 9.1

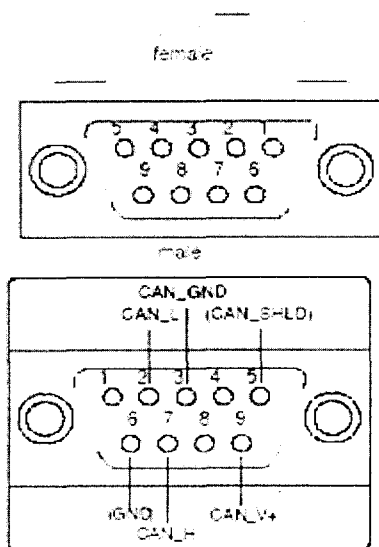
ХАРАКТЕРИСТИКИ ШИНИ

Довжина шини (м)	Кабель		термінуючі резистори (Ом)	Бітова швидкість (Кбіт/с)
	поздовжній опір (МОм/м)	поперечний переріз (мм ²)		
0...40	70	0,25...0,34	124	1000 на 40м
40...300	<60	0,34...0,6	150...300	>500 на 100м
300...600	<40	0,5...0,6	150...300	>100 на 500м
600...1000	<26	0,75...0,8	150...300	>50 на 1000м

Кабелі відгалуження краще підбирати з поперечним розрізом 0.25—0.34 мм². Крім опору кабелю, необхідно враховувати опір роз'ємів, який повинен бути в

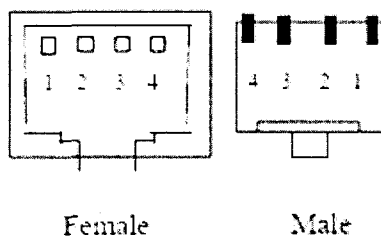
межах 2,5—10 МОм. У даній рекомендації вказуються також випадки, в яких можна не використовувати терміатори лінії.

Тут же визначені декілька типів роз'ємів та їх контактів (рис. 9.2 — 9.5). Найбільш популярний з них 9-піновий sub-D конектор (DIN 41652 або відповідний до нього міжнародний стандарт), який сумісний зі стандартом CiA DS-102 (CAN physical layer for industrial applications). Контакти CAN_H і CAN_L є основними і використання їх обов'язкове. Хоч контакт CAN_GND не вказаний як опціональний, при повністю гальванічно-розв'язаних мережах CAN його можна не використовувати. На кожному із вузлів контакти GND і CAN_GND повинні бути з'єднані. CAN_V+ використовується для підключення зовнішнього живлення 24В за необхідності живлення трансиверів і опторозв'язки для гальванічно розв'язаних вузлів на шині.



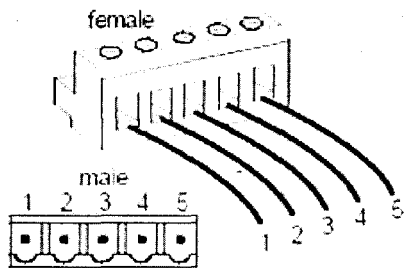
Контакт	Сигнал	Призначення
1	-	зарезервований
2	CAN_L	CAN_L лінія
3	CAN_GND	CAN сигнальна земля
4	-	зарезервований
5	(CAN_SHLD)	опціональний, CAN екран
6	(GND)	опціональний, земля
7	CAN_H	CAN_H лінія
8	-	зарезервований
9	(CAN_V+)	опціональний, плюс зовнішнього живлення

Рис. 9.2. Призначення контактів 9-пінових. SUB-D конекторів



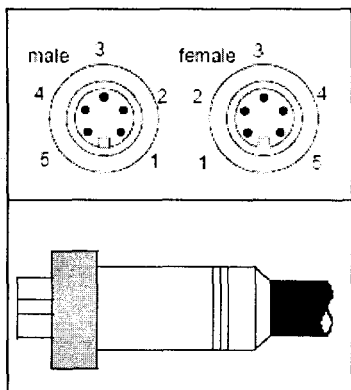
Контакт	Сигнал	Призначення
1	(CAN_V+)	опціональний, плюс зовнішнього живлення
2	CAN_H	CAN_H лінія
3	CAN_L	CAN_L лінія
4	CAN_GND	CAN сигнальна земля

Рис. 9.3. Призначення контактів RJ-45 конекторів



Контакт	Сигнал	Призначення
1	CAN_GND	сигнальна земля
2	CAN_L	CAN_L лінія
3	(CAN_SHLD)	опціональний, CAN екран
4	CAN_H	CAN_H лінія
5	(CAN_V+)	опціональний, плюс зовнішнього живлення

Рис. 9.4. Призначення контактів конекторів типу Open Style



Контакт	Сигнал	Призначення
1	(CAN_SHLD)	опціональний, CAN екран
2	(CAN_V+)	опціональний, плюс зовнішнього живлення
3	CAN_GND	сигнальна земля
4	CAN_H	CAN_H лінія
5	CAN_L	CAN_L лінія

Приклад 9.1. CANOpen. Побудова схеми мережних з'єднань для CANOpen на базі засобів Schneider Electric

Завдання. Необхідно підібрати технічні засоби та розробити схему мережних з'єднань для CANOpen, відповідно до наведеної на рис. 9.6 структури до якої входять: PLC1 — TSX Premium (модель у даному випадку не має значення); RIO1 — розподілена система вводу виводу Advantys STB

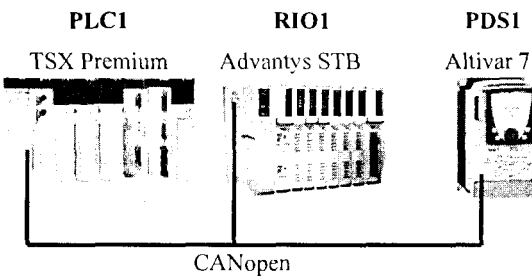


Рис. 9.6. Структурна схема системи до прикладу 9.1

модулями вводу/виводу DDI3610 (6 входів) та DDO3610 (6 виходів); PDS1 — частотний перетворювач Altivar 71.

Рішення. Для засобів Шнейдер Електрик рішення на базі CANOpen сьогодні є одним із базових у системах розподіленого управління, аналогічно, як Profibus DP для Siemens. Тому всі необхідні аксесуари (кабелі відгалужувачі, коробки підключення), які необхідні для побудови мережі CANOpen,

є готовими до використання. Винятком хіба що є сам магістральний кабель, який повинен підключатися до коробок відгалуження.

Для підключення ПЛК TSX Premium та TSX Micro до мережі CANOpen використовується комунікаційна карта TSX CPP110 формату PCMCIA. Ця карта поставляється разом з кабелем відгалуження, який у кінці має коробку підключення з убудованим 9-піновим SUB-D роз'ємом (рис. 9.7). Таке виконання дає змогу уніфікувати кабелі відгалуження.

Для підключення модульного острова вводу/виводу Advantys STB необхідно вибрати базовий комунікаційний модуль. Звісно, що це буде модуль для CANOpen STB NCO 2212 (рис. 9.8). Підключення до модуля виконується через 9-піновий SUB-D конектор, який розміщений безпосередньо на модулі.

Для підключення TSX Premium та Advantys STB на великі відстані можна використати коробки відгалуження, які поєднані магістральним кабелем. У нашому прикладі ми використаємо 4-х портову коробку відгалуження (рис. 9.9) з убудованим термінатором лінії.

Частотні перетворювачі Altivar 71 мають убудований порт Modbus/CANOpen з типом роз'єму RJ-45. Підключення до магістральної шини рекомендується проводити з використанням спеціальної коробки відгалуження VW3 CAN TAP 2 (рис. 9.10). Ця коробка має 2 порти RJ-45 для підключення частотних перетворювача (ATV1 та ATV2) за допомогою кабелю VW3 CA RR 1 та порт для підключення ПК (PowerSuite). Останній дозволяє в режимі роботи системи по Modbus RTU налаштовувати частотні перетворювачі за допомогою спеціального конфігураційного забезпечення PowerSuite. Крім того, дана коробка має вбудований термінатор лінії.

Схема мережних з'єднань показана на рис. 9.11. Специфікація мережних засобів та вузлів дана в таблиці 9.6.

Як видно зі схеми, в обох кінцях лінії, на коробках підключення необхідно перемикачем виставити термінатори лінії.

Слід зауважити, що всі елементи підключення від Шнейдер Електрик виготовлені згідно із CiA DS-102 (для прикладу порівняйте підключення кабелю KM2 до клем S4 з рис. 9.4). Тобто поз. 5-9 специфікації з таблиці 9.6 можна замінити на аксесуари власного виготовлення.

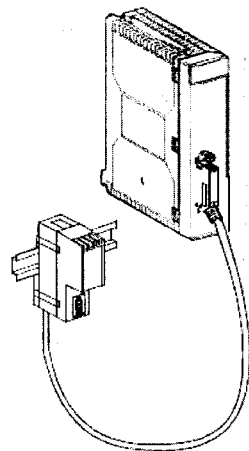


Рис. 9.7. Зовнішній вигляд карти TSX CPP110 з відгалуженням

Приклад 9.1

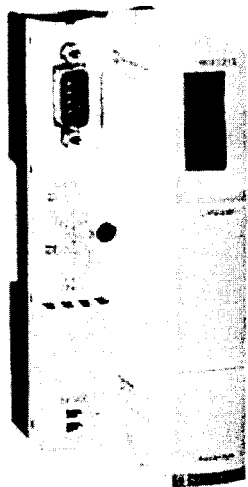


Рис. 9.8. Комунікаційний модуль CANOpen STB NCO 2212

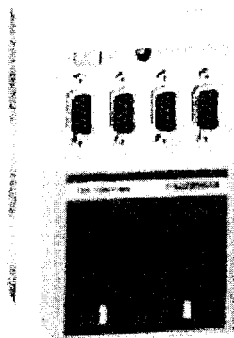


Рис. 9.9. Коробка відгалуження TSX CAN TDM4 на 4 порти типу Sub-D 9-pin

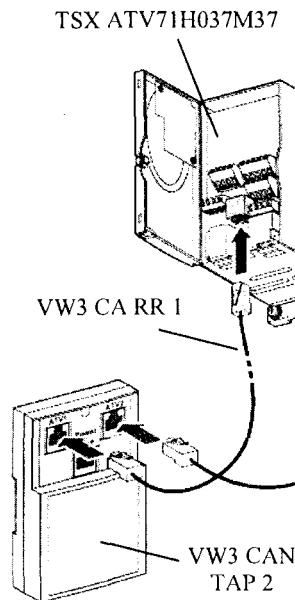
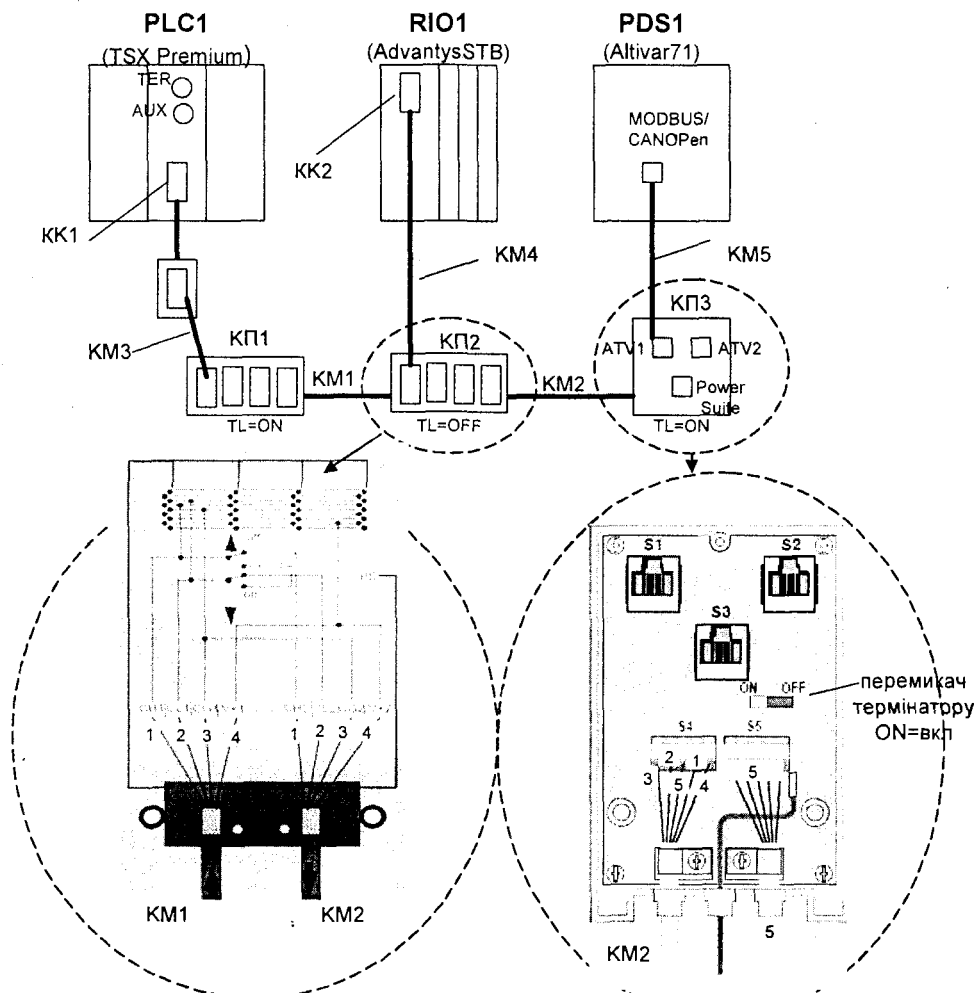


Рис. 9.10. Підключення ATV71 до коробки відгалуження VW3 CAN TAP 2

Таблиця 9.6

СПЕЦИФІКАЦІЯ МЕРЕЖНИХ ЗАСОБІВ

№	Позначення	Назва	Тип	Кількість	Фірма
1	PLC1	ПЛК TSX Premium	TSX 57204M	1 шт.	SE
2	RIO1 (KK2)	Острів віддаленого вводу/виводу Advantys STB, з комунікаційним модулем CANOpen Advantys STB	STB NCO 2212	1 шт.	---
3	PDS1	Частотний перетворювач Altivar71 для управління асинхронним двигуном	ATV71H037M3	1 шт.	---
4	KK1	Комунікаційна PCMCIA карта CANOpen для TSX Micro/Premium	TSX CPP110	1 шт.	---
5	KM1, KM2	Магістральний кабель (подвійна екранована вита пара для CANOpen)	TSX CAN CA100	40 м	---
6	KM3, KM4	Кабель для підключення пристроїв за CANOpen, з обох боків 9-пінова SUB-D розетка, 1 м.	TSX CAN CADD1	1 шт.	---
7	KM5	Кабель для підключення пристроїв за CANOpen, з обох боків RJ-45 вилка, 1 м	TSX CAN CARR1	1 шт.	---
8	KП1, КП2	Коробка підключення пристроїв до магістральної шини CANOpen з 4-ма портами типу 9-пінова SUB-D вилка	TSX CAN TDM4	1 шт.	---
9	KП3	Коробка підключення частотних перетворювачів Altivar до магістральної шини CANOpen, 3 порти RJ-45 розетка (2 – CAN, 1 – Modbus RTU)	VW3 CAN TAP 2	1 шт.	---



*Кольори проводів позначені цифрами: 1 (CAN_H) – білий; 2 (CAN_L) – синій; 3 (GND) – чорний; 4 (V+) – червоний; 5 – не підключений

Рис. 9.11. Схема мережних з'єднань до прикладу 9.1

Приклад 9.1

Приклад 9.2. CANOpen. Побудова схеми мережних з'єднань для CANOpen на базі засобів VIPA та LENZE

Завдання. Необхідно підібрати технічні засоби та розробити схему мережних з'єднань для CANOpen відповідно до наведеної структури на рис. 9.12, до якої входять: PLC1 — VIPA 200V з процесором, що має вбудований CANOpen інтерфейс (CPU 214CAN); RIO1 — розподілена система вводу/виводу VIPA 200V; PDS1 — частотний перетворювач Lenze 8200 Servo.

Рішення. Для підключення розподіленої системи вводу/виводу VIPA 200 до CANOpen використовується інтерфейсний модуль VIPA IM253CAN. Цей

модуль, а також процесорний модуль CPU 214CAN мають SUB-D 9-піновий роз'єм типу «вилка», який відповідає стандарту CiA DS-102. Всі інші аксесуари для підключення (кабелі, конектори, коробки відгалуження) підбираються відповідно до вимог цього ж стандарту.

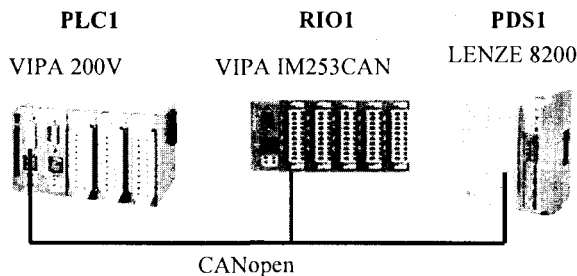


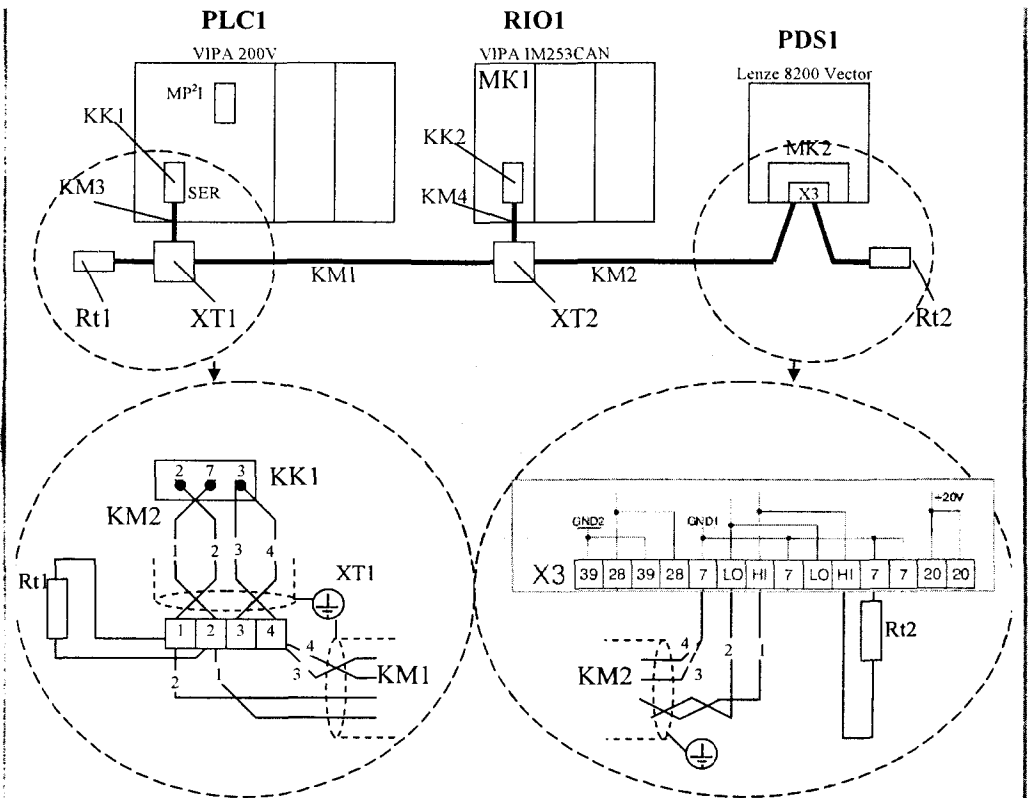
Рис. 9.12. Структурна схема системи до прикладу 9.2

У Lenze є декілька спеціальних модулів для підключення частотного перетворювача 8200 Servo до CANOpen. Серед них — функціональний модуль E82ZAFCC — CAN System Bus. У цього модуля є свої особливості щодо використання в CANOpen. Основні з них — можливість роботи частотних перетворювачів без NMT-Ведучого та відсутність профілю CANOpen для частотного перетворювача. Нижче (в іншому прикладі) ми розглянемо ці особливості більш детально. Слід зазначити, що в Lenze існують модулі, які підтримують CANOpen на всіх рівнях профілів. Враховуючи відсутність готових аксесуарів, схема мережних з'єднань наведена на рис. 9.13. Специфікація мережних засобів та вузлів дана у таблиці 9.7.

Таблиця 9.7

СПЕЦИФІКАЦІЯ МЕРЕЖНИХ ЗАСОБІВ

№	Позначення	Назва	Тип	Кількість	Фірма
1	PLC1	ПЛК VIPA 214 CAN з убудованим інтерфейсом CANOpen	VIPA 214-2CM02	1 шт.	VIPA
2	RIO1 (MK1)	Система віддаленого вводу/виводу VIPA 200V на базі IM253 CAN	VIPA 253-1CA01	1 шт.	VIPA
3	PDS1	Частотний перетворювач Lenze 8200 Vector для управління асинхронним двигуном 0,37 кВт	E82EV371K2C	1 шт.	Lenze
4	MK2	Функціональний модуль CAN System Bus для Lenze 8200	E82ZAFCC	1 шт.	Lenze
5	KK1, KK2	9-піновий SUB-D конектор типу розетка		2 шт.	
6	KM1, KM2	Магістральний кабель, подвійна екранована вита пара для CANOpen	TSX CAN CA100	40 м 40 м	SE
7	KM3, KM4	Кабель відгалуження, подвійна екранована вита пара для CANOpen	TSX CAN CA100	1 м 1 м	SE
8	XT1, XT2	Клемна колодка з 4 клем	TSX CAN CADD1	2 шт.	
9	Rt1, Rt2	Резистор 120 Ом (0.25 Вт)		2 шт.	



* Цифрами позначені кольори проводів:

- 1 (CAN_H) — білий
- 2 (CAN_L) — синій
- 3 (GND) — чорний
- 4 — червоний

Рис. 9.13. Схема мережних з'єднань до прикладу 9.2

На схемі мережних з'єднань показано підключення PLC1 до клемної колодки XT1 кабелем KM3 та розпайку конектора KK1. Аналогічно проводиться підключення KK2 до XT2. Для клемних колодок, конекторів та резисторів виробник не вказаний, оскільки це не має принципового значення. В якості кабелів вибраний стандартний кабель, який пропонується від Шнейдер Електрик. Однак можна вибрати кабель іншого виробника, який задовольняє стандарти CiA DRP-303-1.

Приклад 9.2

Приклад 9.3. CANOpen. Побудова схеми мережних з'єднань для CANOpen на базі засобів UNITRONICS та LENZE.

Завдання. Необхідно підібрати технічні засоби та розробити схему мережних з'єднань для CANOpen відповідно до наведеної на рис. 9.14 структури, до якої входять: PLC1 — Unitronics має вбудований CAN інтерфейс, який можна використати в режимі CAN-bus або CANOpen; DOS1-LENZE 8200.

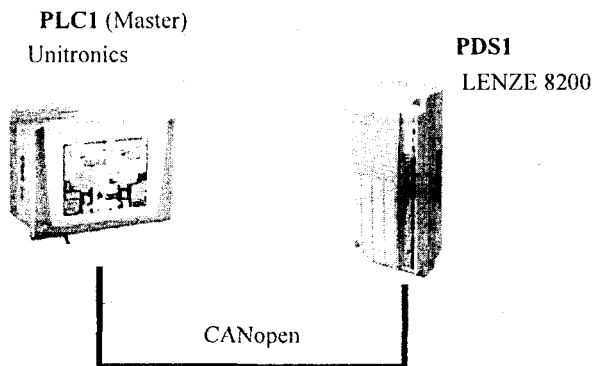


Рис. 9.14. Структурна схема системи до прикладу 9.3

Рішення. Для підключення до CAN частотного перетворювача Lenze необхідний відповідний комунікаційний модуль E82ZAFCC001, який сумісний з CANopen. Схема мережних з'єднань наведена на рис. 9.15.

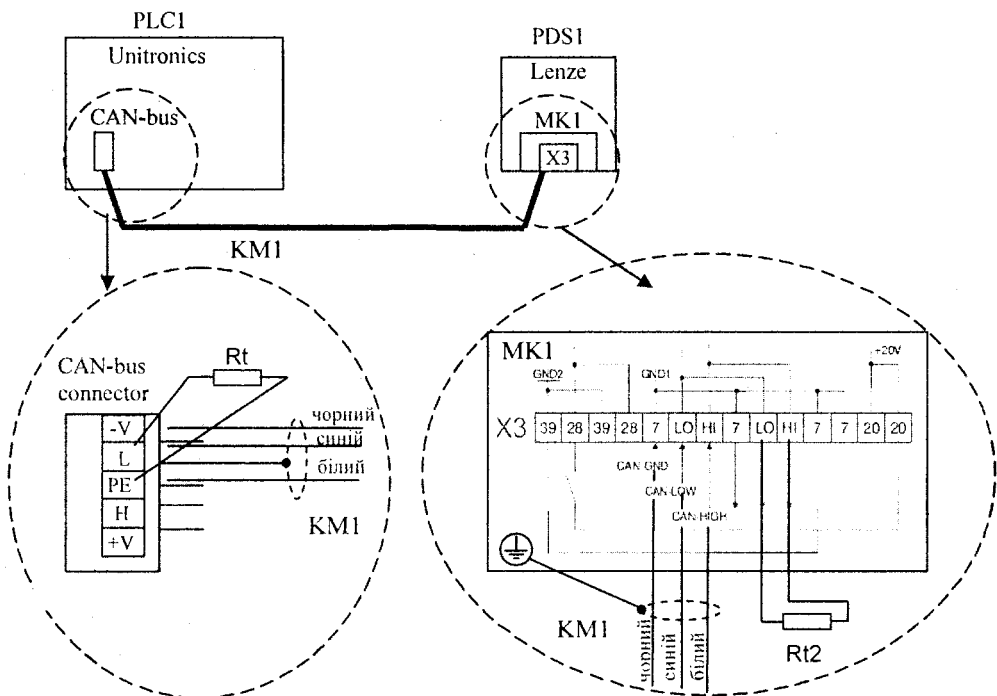


Рис. 9.15. Схема мережних з'єднань до прикладу 9.3

Таблиця 9.8

СПЕЦИФІКАЦІЯ МЕРЕЖНИХ ЗАСОБІВ

№	Позначення	Назва	Тип	Кількість	Фірма
1	PLC1	ПЛК Unitronics Visio570	Visio570	1 шт.	Unitronics
2	PDS1	Частотний перетворювач Lenze 8200 Vector для управління асинхронним двигуном 0,37 кВт	E82EV371K2C	1 шт.	LENZE
3	MK1	комунікаційний модуль для роботи в мережі CAN частотних перетворювачів Lenze 8200	E82ZAFCC001	1 шт.	LENZE
4	KM1	Кабель екранована вита пара	TSX CAN CA100	100 м	
5	Rt1, Rt2	Резистор 120 Ом (0,25 Вт)		2 шт.	

Приклад 9.3

9.3. Базові принципи функціонування

9.3.1. Модель обміну

9.3.1.1. Прикладні Об'єкти та Словник Об'єктів. Для обміну даними між прикладними Процесами вузлів використовуються *Прикладні Об'єкти* (Application Objects), які зберігають ці дані в структурованому вигляді. Базовим поняттям прикладного рівня є *Словник Об'єктів* (Object Dictionary), який вміщує Прикладні Об'єкти, що доступні для обміну, їх типи та опис. Іншими словами, дві прикладні сутності вузлів обмінюються даними через доступні в Словнику Прикладні Об'єкти.

9.3.1.2. Адресація Об'єктів у Словнику. Кожний Прикладний Об'єкт у словнику має унікальну адресу (індекс), яка складається з 16-бітового *Індексу* (Index) та 8-бітового *Під-індексу* (Subindex). Саме за ними проводиться доступ до необхідного об'єкта в Словнику (рис. 9.16).

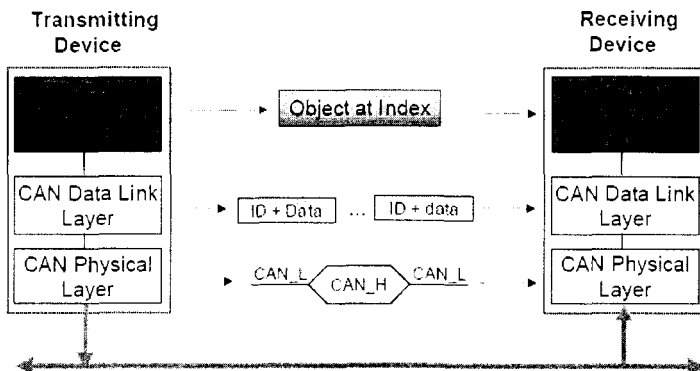


Рис. 9.16. Принципи обміну даними між пристроями в CANOpen

9.3.1.3. Комунікаційні Об'єкти канального рівня. На канальному рівні обмін відбувається за допомогою Комунікаційних Об'єктів **COB** (Communication Object) різних форматів, які передаються CAN-кадрами. Поле Ідентифікатора CAN-кадру відповідає Ідентифікатору COB (**COB-ID**). На фізичному рівні високошвидкісний трансивер передає задані послідовності бітів відповідно до стандарту ISO 11898-2 (рис. 9.16).

Параметри Комунікаційних Об'єктів також зберігаються у Словнику Об'єктів та мають свій Індекс. Крім того, у Словнику зберігаються типи даних, які використовуються для опису Прикладних та Комунікаційних Об'єктів. Таким чином, Словник Об'єктів — це центральна частина моделі вузла CANOpen.

9.3.1.4. Модель пристрою CANOpen. З точки зору моделі CANOpen кожний пристрій складається з трьох частин (рис. 9.17). У центрі моделі міститься Словник Об'єктів — набір типів та структурованих даних, які беруть участь в мережному обміні: дані процесу (внутрішні змінні, входи/виходи), їх типи, інформація про способи їх оновлення, конфігураційні параметри пристрою, комунікаційні параметри каналу і таке інше. Всі об'єкти згруповані в Словнику за їх Індексами. Так, наприклад, у діапазоні 1 по $9F_{16}$ містяться типи даних, з 1000_{16} по $1FFF_{16}$ — дані комунікаційного профілю і т.д.

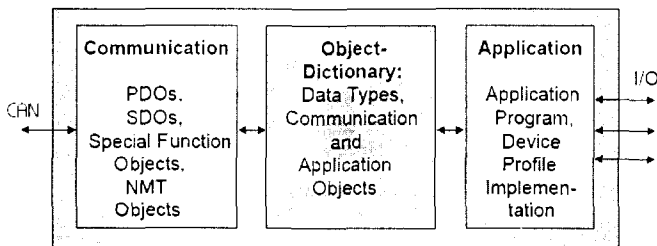


Рис. 9.17. Модель пристрою CANOpen

Словник є проміжним буфером між прикладною програмою пристрою та шиною. Тобто прикладна програма не займається безпосереднім інтерфейсом з шиною, а забезпечує опитування та оновлення значень Прикладних Об'єктів у Словнику. Комунікаційний інтерфейс забезпечує обміном об'єктами Словника з мережею за допомогою спеціальних сервісів. Така структура забезпечує максимальну швидкість та гнучкість.

9.3.2. Комунікаційні сервіси

Канальний рівень CANOpen повинен забезпечити передачу даних між вузлами. Він являє собою реалізацію CAN (ISO 11898), яка оперує стандартними кадрами з 11-бітним ідентифікатором або розширеними 29-бітними, залежно від реалізації CAN-контролеру. Контролер CAN не зможе безпосередньо реалізувати обмін Прикладними Об'єктами між пристроями на шині. Очевидно, що додатково повинен існувати механізм за допомогою якого всі операції з Прикладними

Об'єктами переводяться всього в два типи CAN кадрів: Кадр Даних та Дистанційний Кадр. Крім того, обсяги даних, які ці кадри повинні «переносити», часто перевищують 8 байтів, що потребує передачі декількох CAN-кадрів для реалізації передачі.

У CANOpen є декілька типів сервісів для реалізації функцій переносу Прикладних Об'єктів. Усі вони використовують спеціальні Комунікаційні Об'єкти. Залежно від того, який сервіс використовується, Комунікаційні Об'єкти діляться на 4-ри групи:

- Process Data Object (PDO) — для ідентифікованого обміну даними процесу в реальному часі;

- Service Data Object (SDO) — для читання чи запису об'єктів за їх адресою у Словнику Об'єктів (обмін параметричними даними);

- Special Function Objects (SFO) — реалізують додаткові функції: Синхронізаційний Об'єкт (SYNC) — для синхронізації повідомлень; Аварійний Об'єкт (EMCY) — для аварійних повідомлень; Об'єкт Часовий Відбиток (Time Stamp Object) — для чіткого визначення часу події;

- Network Management (NMT) Objects — для управління станом мережі та пристроїв (ініціалізація, слідкування за помилками, управління станом пристрою): NMT повідомлення; Об'єкт Завантаження (Boot-Up Object); Об'єкт Контролю Помилки (Error Control Object).

Відповідно до цих Комунікаційних Об'єктів, формується CAN-кадр з певним Ідентифікатором та заповненим полем даних. Серед наведених типів комунікаційних об'єктів тільки PDO та SDO займаються переносом значень Прикладних Об'єктів, а інші два сервіси є службовими.

Об'єкти PDO використовуються для реал-тайму обміну ідентифікованими даними процесу (до 8 байтів на PDO) в синхронному або асинхронному, циклічному/періодичному чи ациклічному режимах. Обмін PDO налаштовується при конфігурації мережі.

Об'єкти SDO призначені для обміну параметричними даними та даними процесу великого обсягу, шляхом доступу з мережі до Об'єктів Словника за їх індексом (Індекс + Під-індекс). Це аперіодичний клієнт-серверний тип обміну повідомленнями. Обсяг даних, які можна передати за рахунок SDO, теоретично необмежений.

9.3.3. Основи функціонування сервісу PDO

9.3.3.1. Призначення та типи об'єктів PDO. Сервіс *PDO* (Process Data Object) забезпечує функціонування ідентифікованого обміну Комунікаційними Об'єктами PDO. Під сервісом PDO розуміється весь набір функцій, які забезпечують таке функціонування. Комунікаційний Об'єкт PDO — це, по суті, структура, яка відображає дані процесу (Прикладні Об'єкти). Кожен PDO Об'єкт є унікальним у мережі і має свій унікальний COB-ID ідентифікатор, який співпадає з Ідентифікатором кадру, яким він передається. Параметри Об'єктів PDO розміщуються в Словнику Об'єктів вузлів.

Об'єкти PDO вміщують дані, які повинні передаватися/прийматися в реальному часі. PDO, який містить дані для відправки по мережі, називають Transmit-PDO (**T-PDO**), а для приймання даних — Receive-PDO (**R-PDO**). CANOpen забезпечує відправку даних від T-PDO до R-PDO, які мають однакові COB-ID.

Для відправки одного об'єкта T-PDO використовується один Кадр Даних CAN, який, як відомо, може вміщувати не більше 8 байтів (див. розділ 8). З цього випливає, що один PDO-об'єкт вміщує не більше 8 байтів даних. Передача PDO відбувається за моделлю Виробник-Споживач. Сервіс не потребує підтвердження, що значно зменшує мережний трафік і робить можливим передачу даних процесу в реальному часі. Додатково про модель Виробник-Споживач можна ознайомитися у розділі 2.

9.3.3.2. Реалізація зв'язку T-PDO з R-PDO. Функціонування сервісу PDO проілюстровано на рис. 9.18. PDO Producer вміщує T-PDO з певним COB-ID (позначений ID). Він відправляє Кадр Даних з ідентифікатором ID, у ділянці даних якого розміщується наповнення T-PDO. Вузли, в яких існують R-PDO з COB-ID, рівним ID (позначені як PDO Consumer), переписують значення даних у наповнення R-PDO.

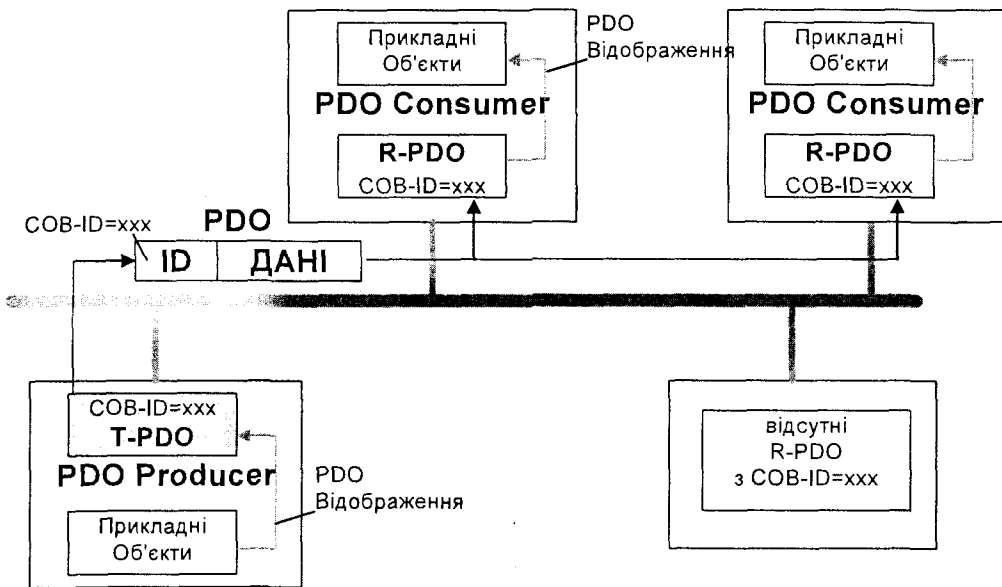


Рис. 9.18. Обмін даними за допомогою PDO

Ідентифікатори PDO, тобто їх COB-ID, зв'язують T-PDO та R-PDO логічним каналом, який так і називається **Link** (Зв'язок). Інформація про ідентифікатори COB-ID, налаштування обміну та наповнення PDO так само містяться в Словнику Об'єктів, як і сам PDO. Таким чином, процес зв'язку та наповнення об'єктів PDO проводиться шляхом зміни інших Об'єктів із Словника. Це робиться за допомогою службових сервісів та спеціальних конфігураторів, які розглянуті нижче.

9.3.3.3. Зв'язок даних PDO з Об'єктами Словника. Джерелом даних для T-PDO є значення певних Об'єктів Словника. Так само отримувачами даних R-PDO є Об'єкти Словника. Зв'язок Об'єктів Словника з наповненням PDO називається **PDO-Відображенням (PDO-Mapping)**. На рис. 9.19 показано, яким чином відбувається Відображення. В настройках PDO послідовно вказуються індекси Об'єктів (Індекс + Під-індекс), які необхідно відобразити на цей PDO. У вказаному на рисунку прикладі це об'єкти *A* (тип Unsigned16), *B* (тип Unsigned8) та *C* (тип Unsigned8) з відповідними умовними індексами xxxx.xx, yyyu.yu та zzzz.zz. При передачі T-PDO з таким відображенням на каналному рівні сформується CAN Кадр Даних, у полі даних якого послідовно передадуться значення об'єкта *B* (1 байт), *A* (2 байти), *C* (1 байт).

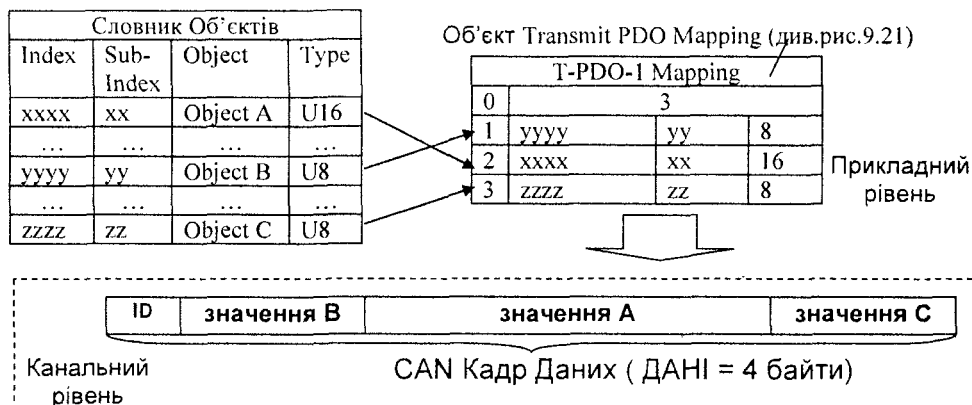


Рис. 9.19. Відображення Прикладних Об'єктів на PDO (PDO-Відображення)

Враховуючи обмеження на обсяг даних у PDO розміром 8 байтів, максимальна кількість Прикладних Об'єктів, які можуть відобразитися у ньому дорівнює 64-м (64 дискретні значення = 8×8 бітів).

Конфігураційні параметри Відображення T-PDO та R-PDO містяться у Словнику з такими Індексами (рис. 9.20):

- Transmit PDO Mapping Parameter (1A00-1BFF₁₆);
- Receive PDO Mapping Parameter (1600-17FF₁₆).

Конфігураційні параметри відображення являють собою Об'єкти типу PDO Mapping Parameter. Сам тип теж міститься в Словнику Об'єктів під індексом 21₁₆ (рис. 9.20). Тип визначає масив структурних даних, в якому нульовий запис вказує на кількість елементів у масиві, а наступні є безпосередньо елементами, 32-бітне значення яких вказує на розміщення та розмір відображених Об'єктів. Зокрема, перші 8 бітів вказують на розмір відображеного Об'єкта, наступні 8 на Під-індекс а останні 16 — на Індекс.

Index 16річ	SubIndex 16річ	Description	Data Type
0021	0	кількість відобр. об'єктів	Unsigned8
	1	1-й об'єкт	Unsigned32
	2	2-й об'єкт	Unsigned32
...
	40	64-й об'єкт	Unsigned32

Тип PDO Mapping Parameter (21₁₆)

Index 16річ	Object	Name	Type	Access	M/O
1600	ARRAY	R-PDO-1 Mapping	PDO Mapping Parameter	rw	M/O
1601	ARRAY	R-PDO-2 Mapping	PDO Mapping Parameter	rw	M/O
...
17FF	ARRAY	R-PDO-512 Mapping	PDO Mapping Parameter	rw	M/O
...
1A00	ARRAY	T-PDO-1 Mapping	PDO Mapping Parameter	rw	M/O
1A01	ARRAY	T-PDO-2 Mapping	PDO Mapping Parameter	rw	M/O
...
1BFF	ARRAY	T-PDO-512 Mapping	PDO Mapping Parameter	rw	M/O

конфігураційні параметри
Відображення

Index 16річ	SubIndex 16річ	Description	Value
1A01	0	кількість відобр. об'єктів	біти 15..8 N
1A01	1	ім'я 1-го об'єкта	Index Length
1A01	2	ім'я 2-го об'єкта	Index Length
...
	N ₁₆	ім'я N-го об'єкта	Index Length

Значення
конфігураційних
параметрів

Рис. 9.20. Параметри відображення PDO

9.3.3.4. Комунікаційні параметри обміну PDO. Крім PDO-Відображення, необхідно також налаштувати такі комунікаційні параметри обміну PDO, як порядок ініціації передачі T-PDO, часові затримки тощо. Для цього в Словнику Об'єктів є спеціальні записи з типом Communication Parameter (20₁₆) (рис. 9.21):

- Receive PDO Communication Parameter (1400-15FF₁₆);
- Transmit PDO Communication Parameter (1800-19FF₁₆);

Призначення полів структури Communication Parameter ми розглянемо в 9.4.3.

Index 16річ	SubIndex 16річ	Description	Data Type
0020	0	кількість входжень	Unsigned8
	1	COB-ID	Unsigned32
	2	transmission type	Unsigned8
	3	inhibit time	Unsigned16
	4	reserved	Unsigned8
	5	event timer	Unsigned16

Тип PDO
Communication
Parameter (21₁₆)

Index 16річ	Object	Name	Type	Access	M/O
1400	RECORD	R-PDO-1 Parameter	PDO Comm Parameter	rw	M/O
1401	RECORD	R-PDO-2 Parameter	PDO Comm Parameter	rw	M/O
...
15FF	RECORD	R-PDO-512 Parameter	PDO Comm Parameter	rw	M/O
...
1800	RECORD	T-PDO-1 Parameter	PDO Comm Parameter	rw	M/O
1801	RECORD	T-PDO-2 Parameter	PDO Comm Parameter	rw	M/O
...
19FF	RECORD	T-PDO-512 Parameter	PDO Comm Parameter	rw	M/O

комунікаційні
параметри PDO

Рис. 9.21. Комунікаційні параметри обміну PDO

9.3.3.5. Основні властивості сервісу PDO. Підбиваючи підсумки, виділимо основні властивості сервісу PDO:

- сервіс PDO забезпечує обмін даними процесу в реальному часі;
- Комунікаційний Об'єкт PDO — це структурні дані, які відображають значення Прикладних Об'єктів зі Словника Об'єктів; цей процес відображення називається PDO-Відображеннями;
- в кожному вузлі розміщується два типи PDO: T-PDO, які призначені для передачі, та R-PDO, які призначені для отримання даних;
- T-PDO передається одним CAN Кадром Даних у широкомовному режимі з Ідентифікатором, рівним COB-ID цього T-PDO; R-PDO вузлів з таким саме COB-ID приймають ці дані (модель Виробник-Споживач);
- T-PDO та R-PDO з однаковими COB-ID називаються зв'язаними, а процес присвоєння їм рівних ідентифікаторів — Зв'язуванням;
- параметри настройки обміну об'єктами PDO, їх типи та об'єкти відображення зберігаються в спеціальних Об'єктах Словника;
- максимальна кількість PDO об'єктів у мережі обмежена і дорівнює 512;

Зміна Об'єктів Словника, які відповідають за настройку PDO-обміну, може проводитись шляхом використання сервісів SDO та NMT, які розглянуті нижче.

9.3.4. Базові концепції конфігурування та адміністрування мережі (NMT-Сервіси)

9.3.4.1. Загальні принципи функціонування NMT. Враховуючи, що область застосування CANOpen — це системи розподілених польових засобів управління рівня датчиків, які необхідно конфігурувати та діагностувати, в протоколі визначений механізм адміністрування мережі. Сервіси, які призначені для адміністрування CANOpen називаються *NMT-Сервісами* (Network Management), і забезпечують управління та діагностування роботи пристроїв на мережі. Функції адміністрування мережі виконує один із вузлів, який зветься *NMT-Ведучий* (NMT-Master). Усі інші вузли називаються *NMT-Веденими* (NMT-Slave). Кожний вузол у CANOpen має свою унікальну адресу NODE-ID (від 1 до 127).

Не слід плутати поняття «NMT-Ведучий» з «Ведучим шини» або «NMT-Ведені» з «Веденими шини». NMT-Ведучий — аналог адміністратора мережі. Він конфігурує всі інші вузли, керує їх операційним станом, діагностує їх роботу, однак не регламентує порядок їх доступу до шини, оскільки в CANOpen (так, як і в CAN) використовується випадковий метод доступу з арбітражем (див. розділ 8). А в мережах з централізованим методом доступу типу Ведучий-Ведений — Ведучий шини регламентує доступ до шини для передачі.

NMT-Ведучий бере на себе такі функції:

- ініціалізація вузлів NMT-Ведених;
- спостереження за станом мережі та її вузлів;
- завантаження/вивантаження конфігураційних даних вузлів.

9.3.4.2. Конфігурування NMT-Ведених. Однією з функцій NMT-Ведучого є конфігурування NMT-Ведених вузлів. Процес конфігурування — це зміна значень потрібних об'єктів у Словнику NMT-Веденого вузла. Запис та читання цих значень проводиться за допомогою SDO-об'єктів за їх Індексом та Під-індексом.

З точки зору розробника мережі, налаштування NMT-Ведучого для конфігурування NMT-Ведених може проводитись різними шляхами:

- 1) за допомогою явної організації в прикладній програмі обміну Об'єктами SDO;
- 2) за допомогою спеціалізованої програми (конфігуратора мережі) та файлів EDS.

Перший спосіб використовується, в основному, в тих випадках, коли недоступний другий спосіб, наприклад для контролерів з підтримкою безпосереднього обміну кадрами CAN, що надають прямий інтерфейс до каналного рівня (наприклад, ОПЛК Unitronics). Однак для інших систем за наявності доступу з прикладної програми до сервісів SDO можна також організувати перенесення конфігураційних Об'єктів Словника.

Другий спосіб автоматизує процес конфігурації NMT-Ведених. Він базується на використанні вбудованих або зовнішніх конфігураторів мереж. Процес конфігурування проходить з використанням файлів *EDS* (Electronic Data Sheet), в яких у визначеному стандарті форматі зберігається інформація про пристрої. Більша частина цієї інформації являє собою опис Об'єктів Словника. Таким чином, використовуючи конфігуратор з підключенням EDS-файлів для необхідних типів пристроїв, процес створення конфігурації зводиться до зміни конфігураційних параметрів вузлів системи. Завантаживши конфігураційні дані у вузол NMT-Ведучого, той сам організовує передачу необхідних конфігураційних SDO в NMT-Ведений, а також його ініціалізацію та запуск. Продемонструємо це на прикладах.

Приклад 9.4. CANOpen. Створення базової конфігурації CANOpen для ПЛК Premium.

Завдання. Створити конфігурацію мережі CANOpen для NMT-Ведучого типу ПЛК Premium та налаштувати всі вузли для роботи системи, показаної в завданні 9.1, якщо:

PLC1 — є NMT-Ведучим з адресою NODE-ID = 127;

RIO1 — NMT-Ведений з адресою NODE-ID = 1;

PDS1 — NMT-Ведений з адресою NODE-ID = 2;

Бітова швидкість для мережі дорівнює 1 Мбіт/с.

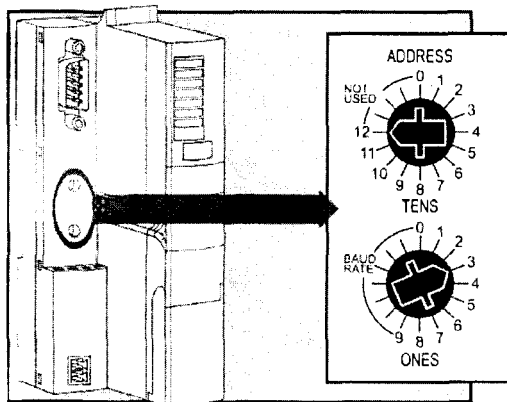


Рис. 9.22. Налаштування адреси та БІТОВА ШВИДКІСТЬ для Advantys STB

Рішення. Налаштування швидкості та адреси для RIO1 (Advantys STB модуль NCO 2212) проводиться за допомогою вбудованих перемикачів (рис. 9.22). Для налаштування біткової швидкості перед увімкненням живлення засобу, нижній перемикач необхідно переключити в позицію BAUD RATE а верхній (вибір біткової швидкості) — в позицію 7 (1 Мбіт/с). Після включення живлення, Advantys запам'ятає вказану швидкість. Адреса пристрою задається комбінацією позицій обох перемикачів: верхній — десятки, нижній — одиниці. Для нашої задачі виставляємо: верхній = 0, нижній = 1.

Налаштування частотного перетворювача PDS1 проводиться за допомогою зовнішньої панелі настройки (можливий варіант — використання спеціалізованого ПЗ — PowerSuite). Виставляємо такі настройки: CANOpen address = 2, CANOpen bit rate = 1 Mbps.

Для частотного перетворювача необхідно виставити такі додаткові настройки:

Drive Menu-> Command->Ref.1.Channel	CANOpen	1 Канал заданої частоти двигуна — CANOpen
Drive Menu-> Command->Ref.2.Channel	HMI	2 Канал заданої частоти двигуна — панель
Drive Menu-> Command->Cmd. channel 1	CANOpen	1 Канал управління двигуном — CANOpen
Drive Menu-> Command->Cmd. channel 2	HMI	2 Канал управління двигуном — панель
Drive Menu-> Command->Profile	I/O Profile	вибраний профіль
Drive Menu-> Command->Cmd Switching	Ch1 Active	активний канал для управління — 1
Drive Menu-> Command->Copy Channel 1<->2	Cmd+ref	копіювання між каналами: команди та частота
Drive Menu-> Command->F1 key assignment	T/K	Призначення кнопки F1 — переключення каналів
Drive Menu-> Command->HMI Cmd	STOP	дозволити зупинку двигуна з панелі

Для створення конфігурації мережі для NMT-Ведучого, тобто ПЛК TSX Premium, окрім програм PL7 PRO або Unity PRO, необхідне додаткове спеціалізоване ПЗ — SyCon. Порядок створення конфігурації наступний:

1. Створюється проект конфігурації мережі на SyCon та записується на диск (рис. 9.23).

2. Створюється конфігурація ПЛК NMT-Ведучого, тобто TSX Premium, в якій замовляється комунікаційна карта TSX CPP110.

3. Для комунікаційної карти вказується файл проекту мережі, створений за допомогою SyCon (рис. 9.24).

Для роботи в SyCon необхідні EDS файли для STB NCO 2212, ATV71H037M3 та TSX CPP110.

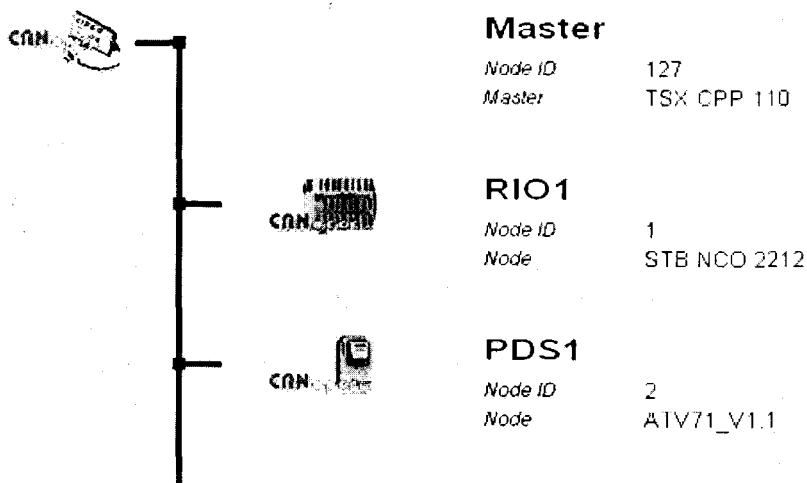


Рис. 9.23. Вигляд конфігурація мережі в SyCon

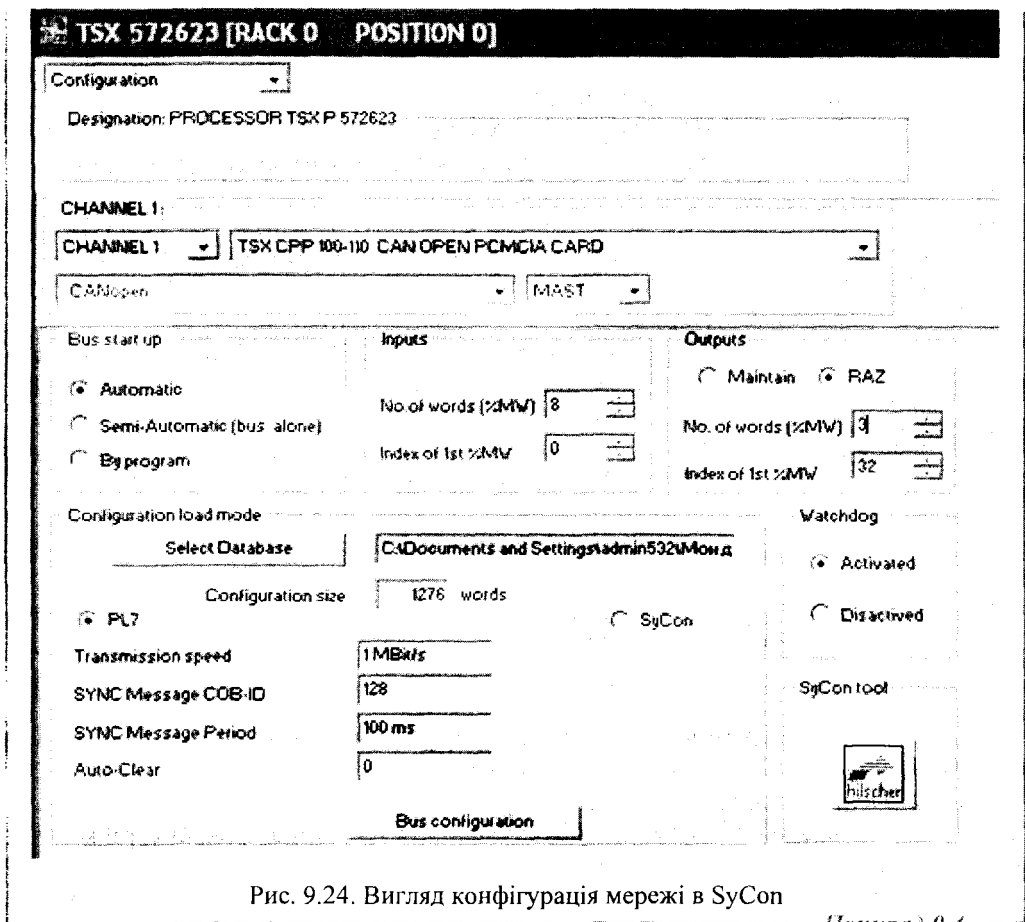


Рис. 9.24. Вигляд конфігурація мережі в SyCon

Приклад 9.4

Приклад 9.5. CANOpen. Створення базової конфігурації CANOpen для ПЛК VIPA.

Завдання. Створити конфігурацію мережі CANOpen для NMT-Ведучого типу ПЛК VIPA та налаштувати всі вузли для роботи системи, показаної в прикладі 9.2, якщо:

PLC1 — є NMT-Ведучим з адресою NODE-ID = 127;

RIO1 — NMT-Ведений з адресою NODE-ID = 1;

PDS1 — NMT-Ведений з адресою NODE-ID = 2;

Бітова швидкість для мережі дорівнює 1 Мбіт/с.

Рішення. Бітова швидкість та адреса модуля IM253CAN налаштовуються за допомогою селектора ADR (рис. 9.25). Перед його включенням селектор установлюють в позицію 00. Включивши пристрій, протягом 5 с, за допомогою селектора вибирається потрібна швидкість (00 — 1 Мбіт/с), після чого вибрана швидкість запам'ятовується в енергонезалежну пам'ять. Далі вводять потрібну адресу (01 — NODE-ID = 1).

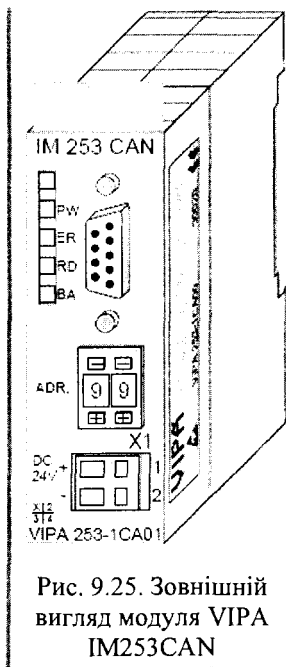


Рис. 9.25. Зовнішній вигляд модуля VIPA IM253CAN

Частотний перетворювач Lenze налаштовується з пульта або за допомогою спеціалізованого ПЗ PowerDrive. Виставляються такі параметри: C0350 = 2 (номер NODE-ID), C0351 = 4 (1 Мбіт/с), C0352 = 0 (Lenze являється NMT Slave). Налаштування активуються після перезапуску частотного перетворювача.

Для створення конфігурації мережі для NMT-Ведучого, тобто ПЛК VIPA, окрім програм WIN PLC7 або STEP 7, необхідне додаткове спеціалізоване ПЗ WinCoST. Порядок створення конфігурації такий:

1. Створюється проект конфігурації мережі на WinCoST та експортується в файл з розширенням wld. (рис. 9.26).

2. Створюється конфігурація ПЛК NMT-Ведучого, тобто VIPA200 за допомогою програм WIN PLC7 або STEP 7.

3. Імпортується wld файл у проект ПЛК, як DB2000 (9.27).

Для роботи в WinCoST необхідні EDS файли для всіх вузлів.

Приклад 9.5

9.3.4.3. Діаграма станів NMT-Веденого. Процес функціонування NMT-Сервісів для управління вузлами прокоментуємо по діаграмі станів NMT-Веденого (рис. 9.28). Після включення живлення пристрою він проходить стадію внутрішньої ініціалізації (**Initialization**), в якій проводить тестування та налаштовує параметри комунікаційного зв'язку (виставляє бітову швидкість, призначає адресу вузла, призначає COB-ID за замовченням і т.д.). Після цього, за відсутності внутрішніх помилок NMT-Ведений переходить у Передопераційний режим (**Pre-Operational**). У цьому режимі NMT-Ведучий може обмінюватися даними з NMT-Веденим, використовуючи тільки сервіси SDO, обмін об'єктами PDO в цьому режимі недоступний. У Передопераційному режимі NMT-Ведучий, використовуючи сервіси SDO, може конфігурувати NMT-Ведені вузли, зокрема налаштувати параметри обміну PDO.

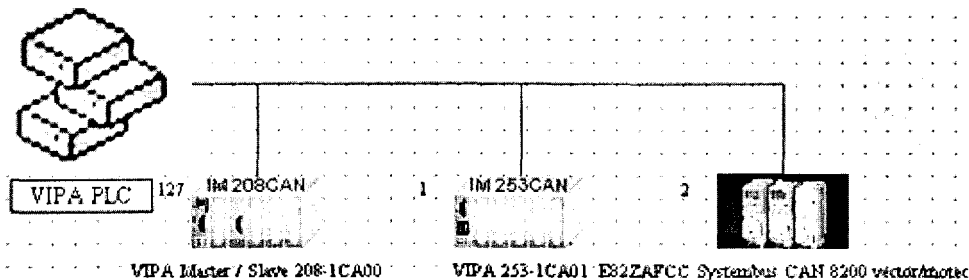


Рис. 9.26. Вигляд конфігурація мережі в WinCoST

- старт Операційного режиму (CS = 1);
- стоп (CS = 2);
- переведення у Передопераційний режим (CS = 80₁₆);
- перезапустити вузол (CS = 81₁₆);
- перезапустити комунікацію (CS = 82₁₆).

Таким чином, використовуючи NMT-Об'єкт, NMT-Ведучий переводить NMT-Ведених у потрібний режим, а використовуючи SDO-Об'єкти — виконує їх конфігурацію. Нагадаємо, що в CANOpen усі Комунікаційні Об'єкти мають унікальний COB-ID, який передається в Ідентифікаторі CAN Кадру Даних. Це значить, що для прийому Об'єктів з відповідним COB-ID NMT-Ведений повинен налаштувати на них вхідний фільтр (див. розділ 8). NMT-Об'єкт має 0-вий Ідентифікатор і приймається всіма вузлами, а його зміст визначає, кому призначена команда. Для обміну з NMT-Веденими іншими Комунікаційними Об'єктами в CANOpen визначена примусова схема розподілу ідентифікаторів COB-ID за замовчуванням (див. 9.3.5).

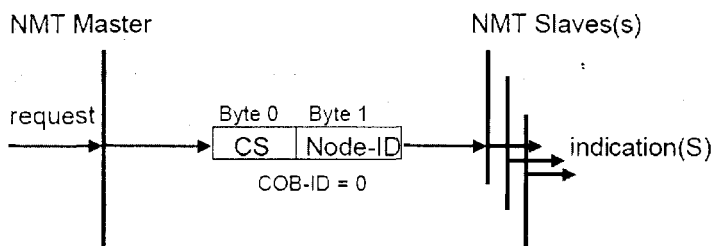


Рис. 9.29. Структура NMT-Об'єкта

Приклад 9.6. CANOpen. Конфігурування мережі CANOpen з використанням ПЛК Unitronics та частотних перетворювачів Lenze

Завдання. Налаштувати всі вузли для роботи системи, показаної в прикладі 9.3, та написати програму в ПЛК для управління станом LENZE, якщо:

- PLC1 — є NMT-Ведучим з адресою NODE-ID = 1;
- PDS1 — NMT-Ведений з адресою NODE-ID = 2;
- БІТОВА ШВИДКІСТЬ для мережі дорівнює 125 Кбіт/с.

Рішення. Частотний перетворювач Lenze налаштовується з пульта або за допомогою спеціалізованого ПЗ PowerDrive. Виставляються такі параметри: C0350 = 2 (номер NODE-ID), C0351 = 2 (125 Кбіт/с), C0352 = 0 (Lenze є NMT Slave). Налаштування активуються після перезапуску частотного перетворювача.

З боку контролера фрагмент програми, що відповідає за обмін у мережі CAN та CANOpen, має вигляд, наведений на рис. 9.30. У блоці CAN_Layer_2 при першому старті контролера налаштовується тип протоколу та бітова швидкість на комунікаційному порті.

Для відправлення кадру використовується блок CAN_L2_Send, де настраюється:

- номер порту, A = 1 (порт 1);
- кількість бітів в ідентифікаторі, B = 0 (11 бітний);

- в якій змінній будуть перебувати дані ідентифікатора COB-ID, C = MI1 (внутрішнє слово);
 - кількість байтів даних для відправки, D = 8 (8 байтів, CAN поле Кадру Даних);
 - номер першої змінної буфера даних для відправки, E = MI2.
- Таким чином, необхідний COB-ID вказується в змінній MI1, а 8 байтів даних містяться в змінних MI2-MI5.

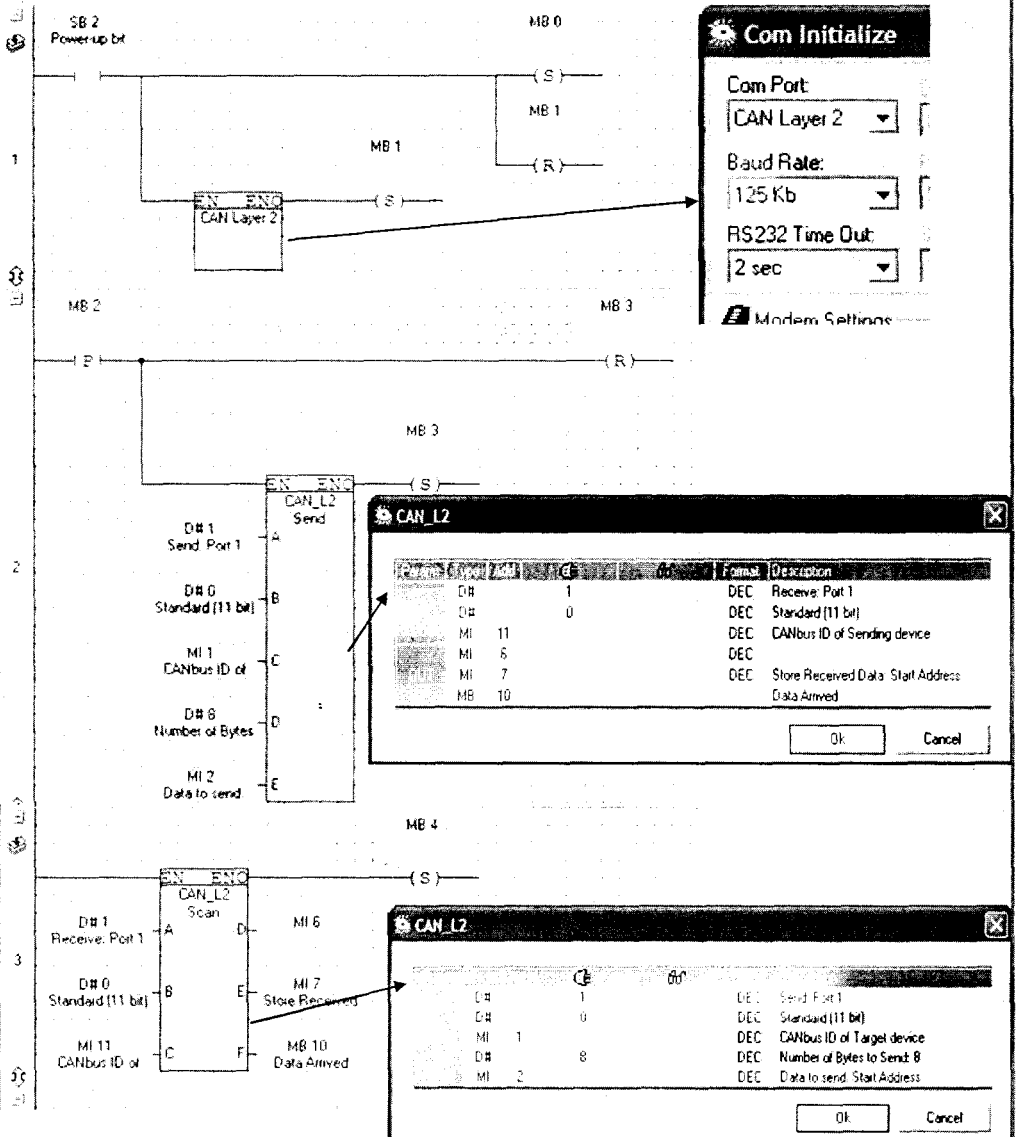


Рис. 9.30. Програма для ПЛК UNITRONICS до прикладу 9.6

Для переведення частотного перетворювача LENZE в різні режими функціонування необхідно з ПЛК відіслати ширококомовний NMT-Об'єкт (COB-ID = 0), з відповідною командою CS, для NODE-ID=2. Тобто, телеграми відправки будуть мати вигляд, як на рис. 9.31. Отже, MI1=0 (COB-ID), CS міститься в молодшому байті змінної MI2, а NODE-ID — у старшому.

Для отримання відповіді використовується блок CAN_L2_Scan, де встановлюються такі настройки: вхідні — номер порту (A); кількість бітів в ідентифікаторі (B), в якій змінні будуть міститися дані ідентифікатора (C); вихідні — змінна, в якій міститься значення кількості отриманих байтів; номер першої змінної, в якій містяться отримані дані «user data» (E); біт активності отриманого кадру (E).

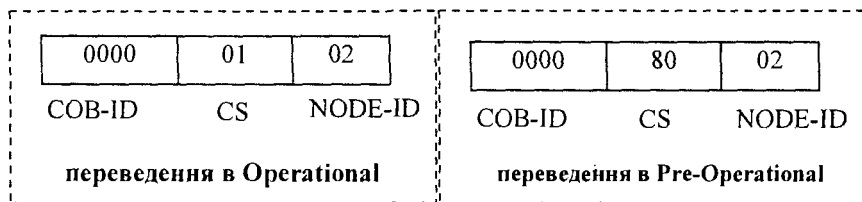


Рис. 9.31. Телеграми NMT-Об'єктів для переведу в різні режими вузла з NODE-ID = 2

Приклад 9.6

9.3.5. Схема розподілення Ідентифікаторів за замовченням

9.3.5.1. Загальна схема розподілу COB-ID. NMT-Ведучий використовує NMT-Об'єкт для переведу вузлів у різні стани. Оскільки COB-ID (Ідентифікатор CAN-кадру) NMT-Об'єкта дорівнює 0, вузли всіх фільтрів пропускають цей кадр. Для обміну іншими Комунікаційними Об'єктами (SDO, PDO, SFO), необхідно щоб вони мали Ідентифікатори відмінні від нуля. Призначення COB-ID можна проводити через сервіс SDO, записавши їх в конкретні записи Словнику. Однак для цього необхідно, щоб хоча б один серверний SDO на кожному вузлі вже мав унікальний COB-ID.

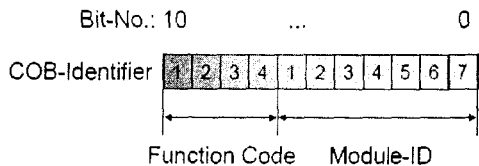


Рис. 9.32. Схема присвоєння Ідентифікаторів

Тому в CANOpen визначена обов'язкова схема розподілення ідентифікаторів COB-ID по замовченню. Розподіл наперед визначених COB-ID проходить в момент ініціалізації NMT-Веденого і базується на використанні ідентифікаторів модулів вузла (**Module-ID**), які в свою чергу поєднані з NODE-ID. Module-ID може виставлятися апаратно (перемикачами) або через інший інтерфейс пристрою.

Схема розподілення Ідентифікаторів по замовченню складається з функціональної частини (**Function Code**), яка визначає призначення об'єкту та Module-ID (рис. 9.32).

Схема розподілу визначає COB-ID для одного Об'єкту Emergency, одного SDO, 4-рьох T-PDO та 4-рьох R-PDO, одного NodeGuard.

Більшість ідентифікаторів COB-ID можна змінити за допомогою SDO, однак частина з них доступна в Словнику тільки для читання. Це ідентифікатори для (див. табл. 9.9):

- NMT-Об'єкту (0);
- SDO-Об'єкту по замовченню (1409-1535 і 1537-1663);
- NMT Error Control (1793-1919);
- NMT, LMT та DBT сервісів (2015-2031);
- Safety-relevant Data Objects (SRDO) (257-384)

Таблиця 9.9

**СХЕМА РОЗПОДІЛУ ФУНКЦІОНАЛЬНИХ КОДІВ
ДЛЯ ІДЕНТИФІКАТОРІВ NMT-ВЕДЕНИХ**

Об'єкт	Function Code (2-кова)	Результуючий COB-ID	Комунікаційні Параметри по Індексу
NMT	0000 ₂	0	—
SYNC	0001 ₂	128 (80 ₁₆)	1005 ₁₆ , 1006 ₁₆ , 1007 ₁₆
TIME STAMP	0010 ₂	256 (100 ₁₆)	1012 ₁₆ , 1013 ₁₆
EMEGRENCY	0001 ₂	129 (81 ₁₆) — 255 (FF ₁₆)	1014 ₁₆ , 1015 ₁₆
T-PDO-1	0011 ₂	385 (181 ₁₆) — 511 (1FF ₁₆)	1800
R-PDO-1	0100 ₂	513 (201 ₁₆) — 639 (27F ₁₆)	1400
T-PDO-2	0101 ₂	641 (281 ₁₆) — 767 (2FF ₁₆)	1801
R-PDO-2	0110 ₂	769 (301 ₁₆) — 895 (37F ₁₆)	1401
T-PDO-3	0111 ₂	897 (381 ₁₆) — 1023 (3FF ₁₆)	1802
R-PDO-3	1000 ₂	1025 (401 ₁₆) — 1151 (47F ₁₆)	1402
T-PDO-4	1001 ₂	1153 (481 ₁₆) — 1279 (4FF ₁₆)	1803
R-PDO-4	1010 ₂	1281 (501 ₁₆) — 1407 (57F ₁₆)	1403
Server T-SDO-1	1011 ₂	1409 (581 ₁₆) — 1535 (5FF ₁₆)	1200
Server R-SDO-1	1100 ₂	1537 (601 ₁₆) — 1663 (67F ₁₆)	1200
NMT Error Control	1110 ₂	1793 (701 ₁₆) — 1919 (77F ₁₆)	1016 ₁₆ , 1017 ₁₆

9.3.5.2. Схема Ідентифікації SDO. Функціонування сервісу SDO описане в 9.4.2. Обмін між Клієнтським та Серверним SDO проводиться через два Комунікаційні Об'єкти з кожного боку: Transmit-SDO (T-SDO) та Receive-SDO(R-SDO). Механізм розподілу COB-ID T-SDO/R-SDO по замовченню для Серверних SDO дає можливість в Передопераційному режимі обмінюватися даними NMT-Ведучому з потрібними NMT-Веденими.

Кожний NMT-Ведений містить принаймні один Серверний SDO. Для цього Об'єкту функціональна частина T-SDO в двійковому вигляді має по замовченню значення FunctionCode = 1011₂ а R-SDO — FunctionCode = 1100₂. Наприклад для 1-го вузла, ідентифікатори Серверних SDO-об'єктів будуть рівними:

T-SDO-1 = 10110000001₂ = 581₁₆

R-SDO-1 = 11000000001₂ = 601₁₆.

9.3.5.3. Схема Ідентифікації PDO

Крім Серверних SDO, при ініціалізації NMT-Веденого надаються Ідентифікатори чотирьом його R-PDO та чотирьом T-PDO. Схема розподілу Function Code для наперед визначених за замовченням Ідентифікаторів дана в таблиці 9.9.

9.3.5.4. Статичне та динамічне зв'язування. Розподіл Ідентифікаторів для PDO за замовченням дає жорстку схему Зв'язування, яка безумовно визначається Module-ID (NODE-ID). Це наперед визначене статичне PDO-Зв'язування NMT-Ведучого — з NMT-Веденими, яке передбачає Зв'язок всіх PDO в NMT-Ведених з відповідними PDO NMT-Ведучого за схемою, показаною на рис. 9.33(а). Такий тип Зв'язування повинні підтримувати всі вузли CANOpen.

Деякі вузли можуть забезпечувати динамічне Зв'язування PDO (PDO-Linking), яке показано на рис. 9.33 (б). Якщо підключені пристрої підтримують PDO-Linking і змінну схему ідентифікації, то розробник може реалізувати зв'язки між Комунікаційними Об'єктами, призначивши їм довільний COB-ID.

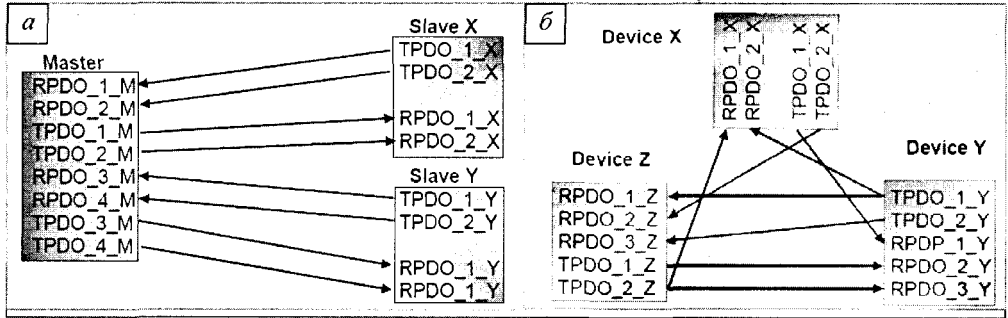


Рис. 9.33. Типи PDO Зв'язування:
 а — наперед визначене статичне; б — динамічне

Приклад 9.7. CANOpen. Зв'язування та Відображення PDO для системи CANOpen з TSX Premium

Завдання. Забезпечити Зв'язування та Відображення PDO вузлів CANOpen для розв'язання завдання, представленого на рис. 9.34. Воно стосується декількох завдань даного розділу, тому на рис. 9.34 слід звертати увагу тільки на змінні вузлів та напрямки їх передачі.

Рішення. Реалізація схеми мережних з'єднань для даного завдання подане в прикладі 9.1. Конфігурація вузлів проводиться згідно з прикладом 9.4. Завдання передбачає, що дані повинні передаватися в реальному часі, тобто необхідно використати сервіс PDO. Всі змінні в частотному перетворювачі PDS1 — 16-бітні аналогові цілі, в RIO1 — дискретні.

При створенні конфігурації в CANOpen в SyCon за замовченням для вузлів виділяється така конфігурація:

- RIO1: 4 об'єкти RPDO + 4 об'єкти TPDO
- PDS1: 1 об'єкти RPDO + 1 об'єкти TPDO.

PLC1 (TSX Premium)

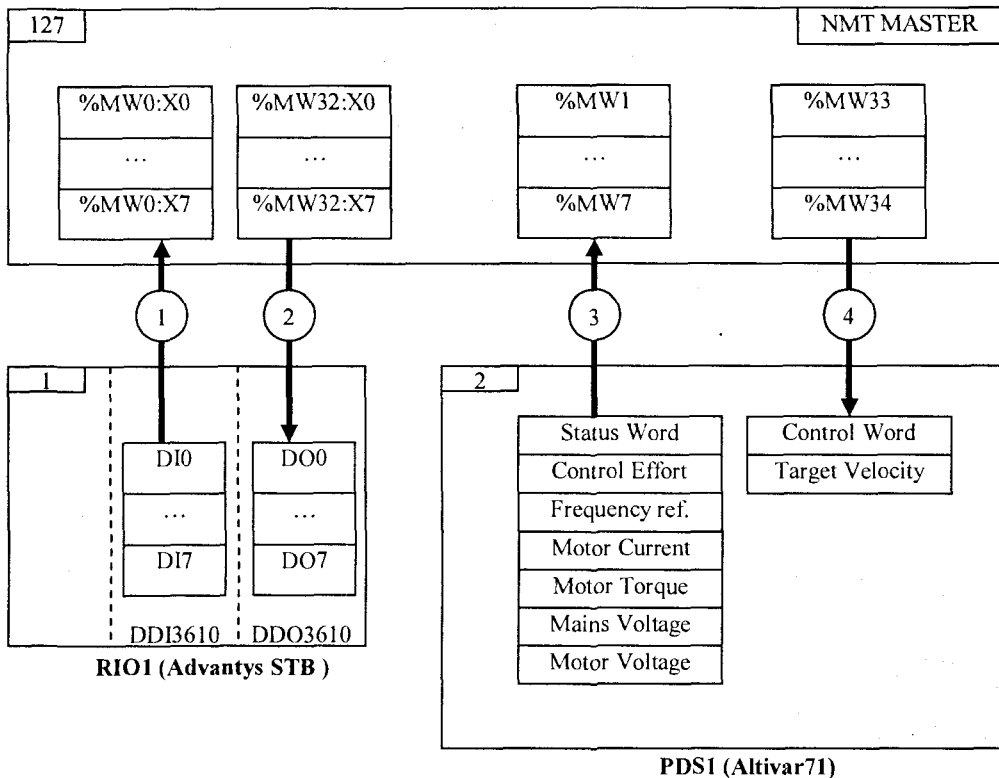


Рис. 9.34. Постановка завдання до прикладу 9.7

PDO Відображення за замовченням визначається профілем пристрою і не співпадає з поставленим в умові прикладу завданням.

У конфігураторі SyCon для RIO1 залишаємо тільки 1 PDO та 1 SDO, що цілком достатньо для передачі 8 бітів в одному та іншому напрямку.

Від PDS1 треба отримати сім 16-бітних слів, що займає 14 байтів, і передати 2 слова — відповідно 4 байти. Враховуючи обмеження PDO 8-ма байтами, необхідно в PDS1 виділити 2 TPDO та 1 RPDO.

Замовлення потрібних PDO та PDO-Відображення в SyCon проводиться доволі просто, без необхідності в налаштуванні параметрів Об'єктів Словника, які відповідають за PDO-Сервіси. Кінцева картина наявних PDO для RIO1 та PDS1 показана на рис. 9.35. Стрілками показані Об'єкти, доступні через PDO, які сконфігуровані в процесі PDO-Відображення (кнопка PDO Contents Mapping).

Після створення (зміни) конфігурації в SyCon в проекті PL7 для комунікаційної карти необхідно виділити 8 вхідних слів (1 байт RIO1 + 7 слів PDS1) та 3 вихідних (1 байт RIO1 + 2 слова PDS1), так, як показано на рис. 9.24.

Node Configuration

Node: STB NCO 2212 Node ID (address): 1

Description: |R101 Configuration Error Control Protocol

File name: STBNC02212.EDS

Activate node in actual configuration Emergency COB-ID: |129

Automatic COB-ID allocation in accordance with Profile 201 Nodeguard COB-ID: |1793

Device Profile: 401 Device type: Analog Output, Analog Input, Digital Output, Digital Input

OK Cancel Node BootUp Object Configuration

Mapped Object dictionary

Obj.Idx.	Sub.Idx.	Parameter	Symbolic name
6200	1	8bit Output Block No. 1	Object6200Idx1

Predelinead Process Data Objects (PDOs) from

Obj.Idx.	PDO name	COB-ID	Type	Len.	IO Type	IO Addr.	IO Len.
1400	RxPDO1 Communication Parameter	513	OB	0	0	1	
1401	RxPDO2 Communication Parameter						
1402	RxPDO3 Communication Parameter						
1403	RxPDO 4 Communication Parameter						
1800	TxPDO1 Communication Parameter	385	IB	0	1		
1801	TxPDO2 Communication Parameter						

PDO mapping method: DS301 V4

Add to configured PDOs

Configured PDOs

PDO name	Symbolic Name	COB-ID	Type	IO Addr.	IO Len.	IO Type	IO Addr.	IO Len.
RxPDO1	PDO_1400	513	OB	0	1			
TxPDO1	PDO_1800	385	IB	0	1			

PDO Contents Mapping... PDD Characteristics... Define new Receive PDD...

Node Configuration

Node: ATV71_V1.1 Node ID (address): 2

Description: |PDS1 Configuration Error Control Protocol

File name: TEATV7111E.EDS

Activate node in actual configuration Emergency COB-ID: |130

Automatic COB-ID allocation in accordance with Profile 301 Nodeguard COB-ID: |1794

Device Profile: 402 Device type: Frequency Converter

OK Cancel Node BootUp Object Configuration

Predelinead Process Data Objects (PDOs) from EDS file

Obj.Idx.	PDO name	COB-ID	Type	Len.	IO Type	IO Addr.	IO Len.
1400	Receive PDO1 parameter	514	OB	1	4		
1401	Receive PDO2 parameter						
1402	Receive PDO3 parameter						
1800	Transmit PDO1 parameter	386	IB	1	8		
1801	Transmit PDO2 parameter	642	IB	5	6		
1802	Transmit PDO3 parameter						

Actual node: 2 | ATV71_V1.1

Mapped Object dictionary

Obj.Idx.	Sub.Idx.	Parameter	Symbolic name
6040	0	Controlword	Object6040Idx0
6042	0	Target Velocity	Object6042Idx0

Add to configured PDOs

Configured PDOs

PDO name	Symbolic Name	COB-ID	Type	IO Addr.	IO Len.	IO Type	IO Addr.	IO Len.
Receive PDO1	PDO_1400	514	OB	1	4			
Transmit PDO1	PDO_1800	386	IB	1	8			
Transmit PDO2	PDO_1801	642	IB	5	6			

PDO Contents Mapping... PDD Characteristics...

Mapped Object dictionary

Obj.Idx.	Sub.Idx.	Parameter	Symbolic name
6041	0	Statusword	Object6041Idx0
6044	0	Control Error	Object6044Idx0
2002	4	Frequency ref.	Object2002Idx4
2002	5	Motor current	Object2002Idx5

Mapped Object dictionary

Obj.Idx.	Sub.Idx.	Parameter	Symbolic name
2002	6	Motor torque	Object2002Idx6
2002	8	Mains voltage	Object2002Idx8
2002	9	Motor voltage	Object2002Idx9

Рис. 9.35. Конфігурування PDO до прикладу 9.7

Приклад 9.7

Приклад 9.8. CANOpen. Зв'язування та Відображення PDO для системи CANOpen з VIPA

Завдання. Забезпечити Зв'язування та Відображення PDO вузлів CANOpen для розв'язання завдання, представленого на рис. 9.36. Воно стосується декількох завдань даного розділу, тому на рис. 9.36 слід звертати увагу тільки на змінні вузлів та напрямки їх передачі.

Рішення. Реалізація схеми мережних з'єднань для даного завдання подана в прикладі 9.2. Конфігурація вузлів проводиться згідно із завданням 9.5. Передбачено, що дані повинні передаватися в реальному часі, тобто необхідно використати сервіс PDO. Всі змінні в частотному перетворювачі PDS1 — 16-бітні аналогові цілі, в RIO1 — дискретні.

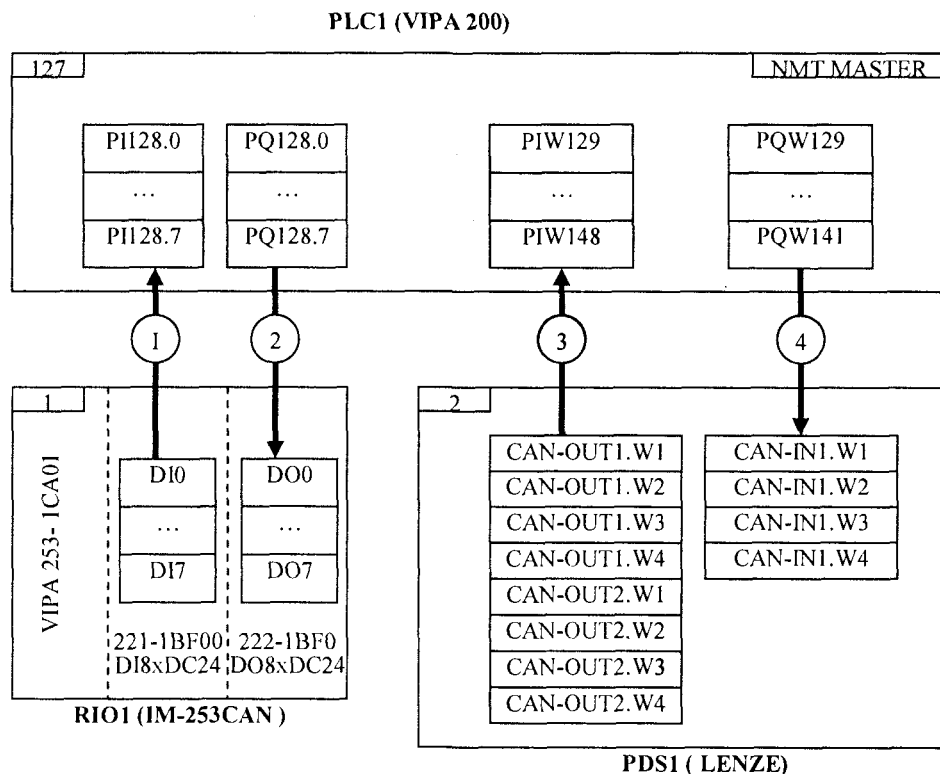


Рис. 9.36. Постановка завдання до прикладу 9.7

Першопочатково в конфігураторі мережі WinCoСТ для RIO1 замовляються дискретні модулі (див. рис. 9.37). Ця процедура автоматично додає необхідні Об'єкти до Словника Об'єктів.

PDO-Зв'язування та PDO-Відображення в WinCoСТ проводиться в одному вікні (рис. 9.39). Спочатку вибирається два PDO, які необхідно зв'язати (поз. 3 на рис. 9.39), а потім для кожного PDO вказуються Об'єкти Словника. Так,

наприклад, TPDO1 з Індексом 1800_{16} в RIO1 (поз.1 на рис. 9.39) зв'язується з RPDO1 з Індексом 1400_{16} в PLC1 (поз. 2). В свою чергу, на TPDO1 в RIO1 відображається Об'єкт з Індексом 6000_{16} (поз. 4.) і Під-індексом 1, які відповідають за дискретні входи на модулі 221-1BF00 (див. рис. 9.36), що з'явився у Словнику після додавання модулів (рис. 9.37). Так само на RPDO1 в PLC1 відображається Об'єкт з Індексом $A4C0_{16}$ та Під-індексом 1, який відповідає за внутрішній буфер обміну 8-бітного розміру.

Device modules for 1 VIPA 253-1CA01



The left list below shows all available modules and the right list below shows the modules, linked to Device. Press '>' button to add and '<' button to remove module.
Note: To add/remove something, select it first.

<ul style="list-style-type: none"> 221-1xxxx DI8 221-1xxxx DI16 221-2xxxx DI32 222-1xxxx DO8 222-1xxxx DO16 222-2xxxx DO32 223-1xxxx DIO8 223-2xxxx DIO16 231-1RD30 AI4x12R 	<p>< ></p>	<ul style="list-style-type: none"> 221-1xxxx DI8 222-1xxxx DO8
---	------------------	--

Description:

222-1xxxx DO8 (8 bit Digital Out)
222-1BF00 DO8xDC24V 1A
222-1BF10 DO8xDC24V 2A

- 0x6000 - Dig8_In
 - Object type: (Var)
 - Data type: (UInt8)
 - Access: (ReadOnly)
 - Low limit
 - High limit
 - Default: (0)
 - Value: (0)
 - 0x0000 - NrOfObjects
 - 0x0001 - 221-1xxxx DI8_1_1**
 - 0x6002 - Polarity Dig8_In
 - Object type: (Var)
 - Data type: (UInt8)
 - Access: (ReadOnly)
 - Low limit
 - High limit
 - Default: (0)
 - Value: (0)
 - 0x0000 - NrOfObjects
 - 0x0001 - 221-1xxxx DI8_1_1

Рис. 9.37. Замовлення дискретних модулів (зліва) та відповідні Об'єкти в Словнику (справа) для RIO1

- 0x181
 - 0x1400 0x181 (127 VIPA Master / Slave 208-1CA00)
 - 0x1800 0x181 (1 VIPA 253-1CA01)
 - 0x182
 - 0x1401 0x182 (127 VIPA Master / Slave 208-1CA00)
 - 0x1800 0x182 (2 E82ZAFCC Systembus CAN 8200 vectorfr)
 - 0x201
 - 0x1400 0x201 (1 VIPA 253-1CA01)
 - 0x1800 0x201 (127 VIPA Master / Slave 208-1CA00)
 - 0x202
 - 0x1400 0x202 (2 E82ZAFCC Systembus CAN 8200 vectorfr)
 - 0x1801 0x202 (127 VIPA Master / Slave 208-1CA00)
 - 0x282
 - 0x1402 0x282 (127 VIPA Master / Slave 208-1CA00)
 - 0x1801 0x282 (2 E82ZAFCC Systembus CAN 8200 vectorfr)

Рис. 9.38. Список Зв'язаних PDO та їх COB-ID

Загальний список Зв'язаних у мережі PDO згрупований за їх COB-ID, наведений на рис. 9.38.

Змінні, які відповідають в ПЛК1 за вхідні та вихідні дані, вказуються в самому конфігураторі WinCoST. Кількість байтів, які передаються в ПЛК = 17, з ПЛК = 9.

		1 VIPA 253-1CA01								- E82ZAFCC Systembus CA								27 VIPA Master / Slave 208-1CA00												
		0x1400	0x1401	0x1402	0x1403	0x1404	0x1405	0x1406	0x1407	0x1408	0x1409	0x1400	0x1401	0x1402	0x1403	0x1404	0x1400	0x1401	0x1402	0x1403	0x1404									
1	4	0x6200.01										0x5CA7.01	0x5CA7.02	0x5CA7.03	0x5CA7.04	0x5CA8.01	0x5CA8.02	0x5CA8.03	0x5CA8.04	0x5CA9.01	0x5CA9.02	0x580.03	0x580.04	0x580.05	0x580.06	0x580.07	0x580.08	0x580.09		
1	VIPA 253-1CA01	0x1800	0x6000.01																											
	0x1801																													
	0x1802																													
	0x1803																													
	0x1804																													
	0x1805																													
	0x1806																													
	0x1807																													
	0x1808																													
	0x1809																													
	82ZAFCC Systembus CA											0x5CAD.01	0x5CAD.02	0x5CAD.03	0x5CAD.04	0x5CAE.01	0x5CAE.02	0x5CAE.03	0x5CAE.04											
	0x1800																													
	0x1801											0x5CAE.01	0x5CAE.02	0x5CAE.03	0x5CAE.04															
	0x1800											0xA100.01	0xA100.02	0xA100.03	0xA100.04	0xA100.05														
	0x1801																													

Connections		Device: 2E82ZAFCC System	Device: 127 VIPA Master / Slave
COB ID:		Tx PDO:	Rx PDO 1:
Transmission Type:		0x5CAE 0x01 16 bit	0x580 0x06 16 bit
Inhibit Timer:		0x5CAE 0x02 16 bit	0x580 0x07 16 bit
Event Timer:		0x5CAE 0x03 16 bit	0x580 0x08 16 bit
PDO Check:		0x5CAE 0x04 16 bit	0x580 0x09 16 bit
• Match	Size
Force 29 bit PDO	<input type="checkbox"/>
Connected:	<input checked="" type="checkbox"/>

Рис. 9.39. PDO-Зв'язування та PDO-Відображення

При конфігурації ПЛК1 у WinCoST пам'ять виділяється блоками по 4 байти. Таким чином, для ділянки входів необхідно виділити 5 блоків, а для виходів — 3 блоки (див. рис. 9.40).

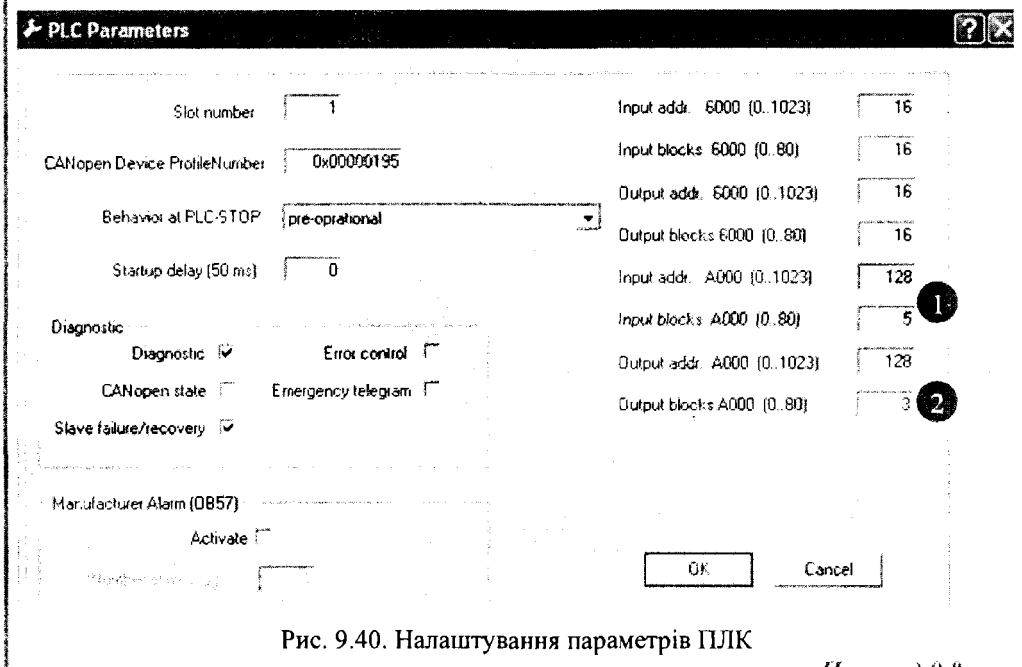


Рис. 9.40. Налаштування параметрів ПЛК

Приклад 9.8

9.4. Робота зі Словником Об'єктів

9.4.1. Структура Словника Об'єктів

Загальна структура Словника Об'єктів усіх пристроїв однакова, оскільки Об'єкти в Словнику розміщуються згідно з їх функціональним призначенням. Наявність певних Об'єктів є обов'язковою, а їх розміщення визначено стандартом. Інші Об'єкти можуть міститися тільки в спеціально виділених областях Індексів. Наявність і властивості Об'єктів Словника доступні розробнику мережі через EDS-файли, структура яких стандартизована, що дозволяє переглядати їх за допомогою різних конфігураторів (наприклад, SyCon та WinCoST).

Загальний розподіл Об'єктів у Словнику показаний в таблиці 9.10. Область з Індexами 1-9F₁₆ виділяється під типи даних. Об'єкти комунікаційного профілю відповідають за комунікаційний обмін між пристроями. Частина з цих об'єктів, які відповідають за обмін PDO, розглянута вище, частина наведена в таблиці 9.11. Специфічні для пристрою дані, які визначені його Профілем, розміщуються в Об'єктах з Індexами 2000-5FFF₁₆. Для CANopen визначено ряд профілів пристроїв, Прикладні Об'єкти яких завжди розміщуються в області 6000-9FFF₁₆.

Таблиця 9.10

ЗАГАЛЬНИЙ РОЗПОДІЛ ОБ'ЄКТІВ У СЛОВНИКУ

Index (16-кова)	Об'єкти (англ)	Об'єкти (укр)
0000	Reserved	зарезервовані
0001-001F	Static Data Types	Статичні Типи Даних
0020-003F	Complex Data Types	Структурні Типи Даних
0040-005F	Manufactured Specific Data Types	Типи Даних, визначені виробником
0060-007F	Device Profile Specific Static Data Types	Статичні Типи Даних, визначені Профілем пристрою
0080-009F	Device Profile Specific Complex Data Types	Структурні Типи Даних, визначені Профілем пристрою
00A0-0FFF	Reserved for further use	зарезервовані на майбутнє використання
1000-1FFF	Communication Profile Area	Область Об'єктів, визначених Комунікаційним Профілем
2000-5FFF	Manufactured Specific Profile Area	Область Об'єктів, визначених виробником
6000-9FFF	Standardized Device Profile Area	Область Об'єктів, визначених Стандартними Профілями Пристроїв
A000-FFFF	Reserved for further use	зарезервовані на майбутнє використання

Таблиця 9.11

ОБ'ЄКТИ ОБЛАСТІ КОМУНІКАЦІЙНОГО ПРОФІЛЮ

Index (16-кова)	Object	Name	Type	Access	M/O
1000	VAR	device type	Unsigned32	ro	M
1001	VAR	error register	Unsigned8	ro	M
1002	VAR	manufacturer status register	Unsigned32	ro	O
1003	ARRAY	pre-defined error field	Unsigned32	ro	O
1004	зарезервований для майбутнього використання				
1005	VAR	COB-ID SYNC-message	Unsigned32	rw	O
1006	VAR	communication cycle period	Unsigned32	rw	O
1007	VAR	synchronous window length	Unsigned32	rw	O
1008	VAR	manufacturer device name	VIS-String	c	O
1009	VAR	manufacturer hardware version	VIS-String	c	O
100A	VAR	manufacturer software version	VIS-String	c	O
100B	зарезервований для майбутнього використання				
100C	VAR	guard time	Unsigned32	rw	O
100D	VAR	life time factor	Unsigned32	rw	O
100E	зарезервований для майбутнього використання				

Index (16-кова)	Object	Name	Type	Access	M/O
100F	зарезервований для майбутнього використання				
1010	VAR	store parameters	Unsigned32	rw	O
1011	VAR	restore default parameters	Unsigned32	rw	O
1012	VAR	COB-ID time stamp	Unsigned32	rw	O
1013	VAR	high resolution time stamp	Unsigned32	rw	O
1014	VAR	COB-ID Emergency	Unsigned32	rw	O
1015	VAR	Inhibit Time Emergency	Unsigned16	rw	O
1016	ARRAY	Consumer Heartbeat Time	Unsigned32	rw	O
1017	VAR	Producer Heartbeat Time	Unsigned16	ro	O
1018	RECORD	identity object	Identity		M

9.4.2. Основи функціонування сервісу SDO

SDO (Service Data Object) — це Об'єкти Сервісних Даних, які призначені для доступу до даних Словника Об'єктів за Індексом та Під-Індексом. Використовується Клієнт-Серверна модель обміну між прикладними Процесами. Словник Об'єктів кожного вузла включає мінімум один Серверний Об'єкт SDO, через який можна проводити обмін конфігураційними даними. Клієнтський процес повинен включати Клієнтський SDO, за допомогою якого він формує запит на читання чи запис Об'єкту Словника, вказуючи його Індекс та Під-Індекс (рис. 9.41).

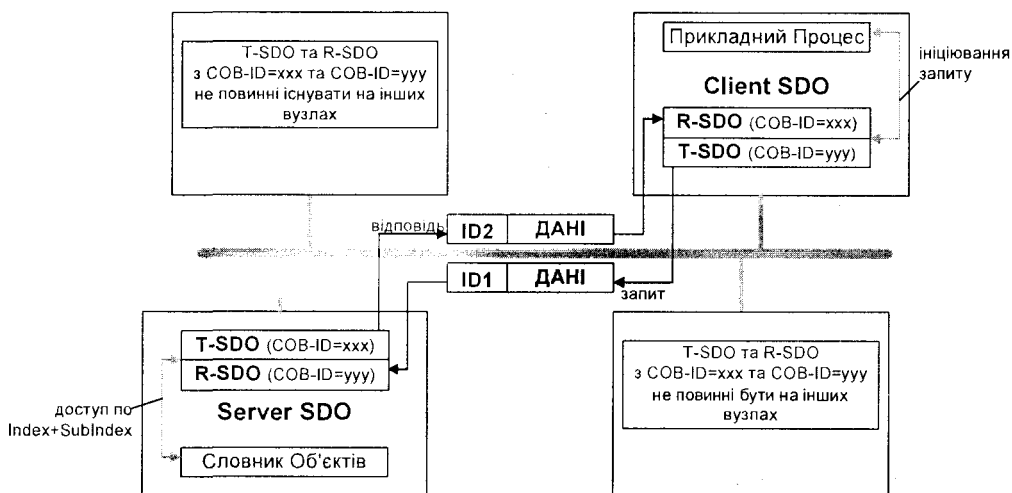


Рис. 9.41. Клієнт-серверна архітектура SDO

Клієнтські та Серверні SDO створюють канал з двома COB-ID з кожного боку для організації двосторонньої передачі: клієнтський Transmit-SDO (T-SDO) «поєднується» Ідентифікатором серверного Receive-SDO (R-SDO) та навпаки. В одній мережі CANOpen можуть функціонувати до 256 каналів SDO.

За допомогою сервісу SDO можна зчитувати або записувати дані обсягом більшим, ніж 8 байтів. Це досягається шляхом сегментування цих даних. Параметри SDO можна налаштувати через Словник Об'єктів (рис. 9.42).

Index 16 річ	SubIndex 16річ	Description	DataType		
0022	0	кількість входжень	Unsigned8		
	1	COB-ID Client->Server	Unsigned32		
	2	COB-ID Server->Client	Unsigned32		
	3	Node ID	Unsigned8		

Index 16 річ	Object	Name	Type	Access	M/O
1200	RECORD	Server SDO-1 Parameter	SDO Parameter	rw	M/O
1201	RECORD	Server SDO-2 Parameter	SDO Parameter	rw	M/O
...
127F	RECORD	Server SDO-128 Parameter	SDO Parameter	rw	M/O
...
1280	RECORD	Client SDO-1 Parameter	SDO Parameter	rw	M/O
1281	RECORD	Client SDO-2 Parameter	SDO Parameter	rw	M/O
...
12FF	RECORD	Client SDO-128 Parameter	SDO Parameter	rw	M/O

Рис. 9.42. SDO в словнику об'єктів

9.4.3. Типи передачі PDO та об'єкт SYNC

9.4.3.1. Типи передачі PDO. З точки зору ініціації обміну, відправка PDO може проходити в таких режимах:

- ациклічний, при виникненні специфічної події на пристрої;
- ациклічний, за запитом;
- циклічний, з прив'язкою до циклічного об'єкта SYNC.

З точки зору синхронності в CANOpen виділяють такі режими передачі:

- синхронна передача, з прив'язкою до об'єкта SYNC;
- асинхронна передача, без прив'язки до об'єкта SYNC.

Комбінації цих режимів дають *Тип Передачі PDO (Transmission Type)*, який є одним із полів структури PDO Communication Parameter (див. 9.3.3). Тип Передачі визначає подію, при якій відбувається T-PDO. Список усіх доступних значень цих типів залежно від режимів показаний у таблиці 9.12.

ЗНАЧЕННЯ ПОЛЯ TRANSMISSION TYPE PDO

Номер типу	циклічно	ациклічно	синхронно	асинхронно	тільки за RTR
0		X	X		
1-240	X		X		
241-251			зарезервовано		
252			X		X
253				X	X
254				X	
255				X	

9.4.3.2. Об'єкт SYNC. Функціонування синхронного обміну забезпечується шляхом прив'язки ініціації передачі T-PDO до Комунікаційного Об'єкта *SYNC* (належить до SFO). Цей Об'єкт періодично передається одним із вузлів (*SYNC-Producer*), яким, як правило, є NMT-Ведучий. Прив'язавшись до цього Об'єкта, інші вузли (*SYNC-Consumer*) можуть організувати циклічну передачу. Враховуючи, що публікація Об'єкта SYNC періодична, циклічний тип T-PDO буде передаватись теж періодично. З іншого боку, Об'єкт SYNC може служити і для ациклічних передач в якості синхросигналу. Крім того, з отриманням Синхронного Об'єкта пристрої можуть зберегти значення своїх входів для ефекту квазіодночасності. У будь-якому випадку, якщо передача PDO залежить від появи на шині Об'єкта SYNC, цей PDO є синхронним, якщо ні — асинхронним (рис. 9.43).

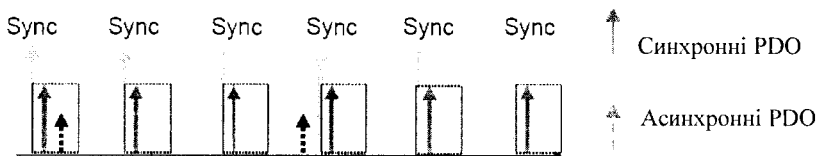


Рис. 9.43. Синхронні та асинхронні PDO

Об'єкт SYNC у Словнику Об'єктів міститься під Індексом 1005₁₆. Там вказується його Ідентифікатор, який за замовченням дорівнює 128. Повідомлення передається без додаткової передачі даних (рис. 9.44).

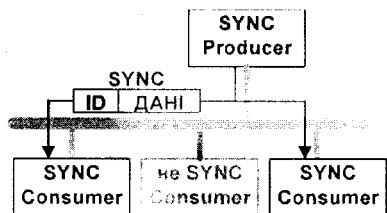


Рис. 9.44. Відновлення Об'єкта SYNC

Синхронні PDO передаються тільки під час синхронного вікна. Тобто, довжина синхронного вікна — це інтервал часу між появою Об'єкта SYNC та дозволеною передачею синхронних PDO (рис. 9.45). Ця величина може конфігуруватися окремо для всіх вузлів через Об'єкт Synchronous window length (1007₁₆, величина вказується в мікросекундах). Для оптимальної роботи ме-

режі кожний SYNC-Consumer у момент конфігурування також отримує значення періоду появи Об'єкта SYNC, який записується в Communication cycle period (1006_{16} , величина вказується в мікросекундах).

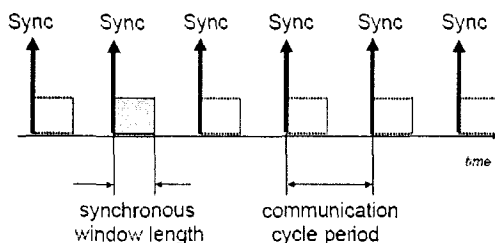


Рис. 9.45. Довжина синхронного вікна

9.4.3.3. Ациклічні синхронні PDO (Тип Передачі 0). Цей тип PDO буде передаватись ациклічно при специфічній для пристрою події (наприклад при зміні значення змінної), але тільки в момент отримання Об'єкта SYNC (9.46).

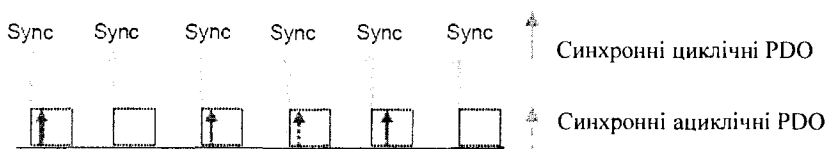


Рис. 9.46. Синхронні циклічні та ациклічні PDO

9.4.3.4. Циклічні синхронні PDO (Тип Передачі 1-240). Ці PDO передаються синхронно з періодом, який визначається кількістю Об'єктів SYNC між передачами. На цю кількість вказує значення типу PDO (від 1 до 240). Так тип=1 вказує, що PDO буде передаватися з кожною появою Об'єкта SYNC, а тип=2 — з появою кожного другого SYNC.

9.4.3.5. Синхронні за запитом PDO (Тип Передачі 252). Цей тип PDO вказує на те, що дані, які містяться в ньому, повинні бути оновлені в момент отримання Об'єкта SYNC. Для віддаленого вхідного модуля, наприклад, це значить, що в момент отримання Об'єкта SYNC вхідні дані будуть оцифровані і збережені в T-PDO. Однак, відправлений T-PDO буде тільки в момент отримання запиту Дистанційного Кадру (RTR). Запит на зчитування T-PDO генерує NMT-Ведучий, для якого в конфігураторі можна вказати додаткові настройки (наприклад, кількість циклів контролеру між опитуваннями).

9.4.3.6. Асинхронні за запитом PDO (Тип Передачі 253). Дані для цього типу PDO відновляються відразу при отриманні Дистанційного Кадру, після чого T-PDO відправляється в мережу.

9.4.3.7. Асинхронні ациклічні PDO (Тип Передачі 254 та 255). Відправка в мережу цих типів PDO не залежить від появи Об'єкта SYNC. Момент відправки визначається профілем пристрою (Тип Передачі 255) або виробником пристрою

(Тип Передачі 254). Для таких типів PDO можна визначити інтервал часу між відправками T-PDO. Цей параметр доступний не для всіх типів пристроїв і на-строюється в Комунікаційних Параметрах PDO в полі *Event Timer* (див. рис. 9.21).

Враховуючи, що Відображені Об'єкти на PDO типів 254 та 255 можуть оновлюватися дуже часто, передача PDO з високим пріоритетом може витіснити інші PDO. Для запобігання цього ефекту в Комунікаційних Параметрах PDO визначене поле Час Заборони (*Inhibit Time*), яке вказує на мінімальний час між двома передачами даного PDO (див. рис. 9.21).

9.4.4. Об'єкт Time Stamp

Для синхронізації процесів за часом, у CANOpen використовується Об'єкт *Time Stamp*. Даний Об'єкт є 6-байтною послідовністю, яка показує абсолютний час у мілісекундах після опівночі та кількість днів після 1-го січня 1984 року. Для більш критичних за часом синхронізацій (порядку мікросекунд) використовується високоякісний time-stamp, який кодується 32-бітною послідовністю, що показує кількість мікросекунд. Лічильник мікросекунд таким чином повинен рестартувати кожні 72 хвилини. Даний сервіс функціонує за моделлю Producer-Consumer.

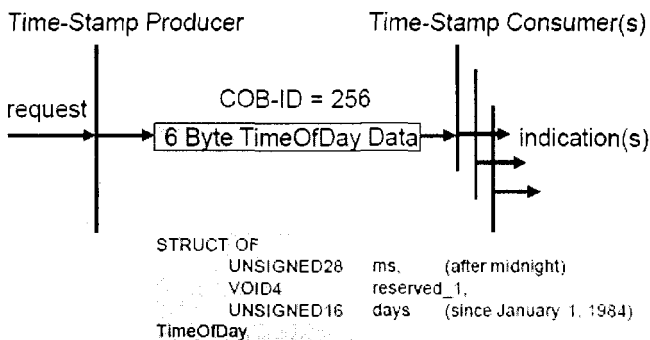


Рис. 9.47. Об'єкт Time Stamp

9.4.5. Об'єкт Emergency

Повідомлення *Emergency* передається при виникненні фатальної помилки на пристрої і має високий пріоритет. Воно відправляється кожен раз при виникненні нової помилки. Коди помилок визначаються за допомогою комунікаційного профілю пристрою. Emergency Object є опціональним. Якщо він підтримується, то повинні підтримуватись, як мінімум, два коди помилок 00xx (помилки немає) і 11xx (генерація помилки). Вміщує 8 байтів даних і підтвердження передачі. Даний сервіс функціонує за моделлю Producer-Consumer.

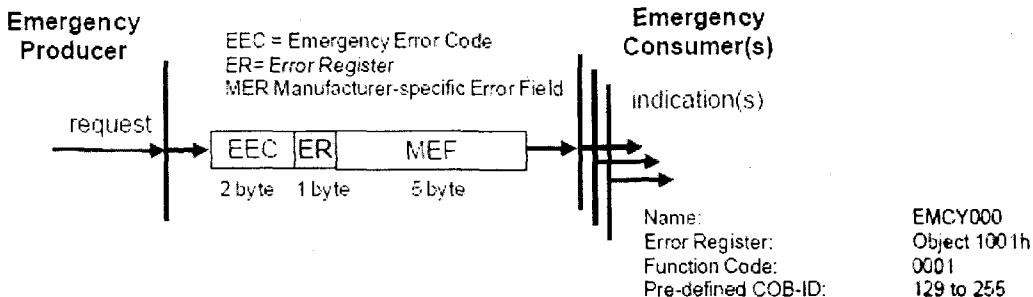


Рис. 9.48. Об'єкт Emergency

9.4.6. Діагностичні сервіси NodeGuard та Heartbeat

9.4.6.1. Node Guarding. Для слідкування за працездатністю вузлів у мережі, тобто за регулярною передачею ними PDO, NMT Master може вести базу даних стану NMT-Ведених. Цей процес називається *Node Guarding* (Охорона Вузлів). Для цього NMT-Ведучий регулярно політнгує NMT-Ведених (рис. 9.49). Для виявлення відсутності NMT-Ведучого NMT-Ведені в свою чергу перевіряють, чи проводився на певному інтервалі політн (Life Guarding). Node Guarding активний також і в режимі стоп.

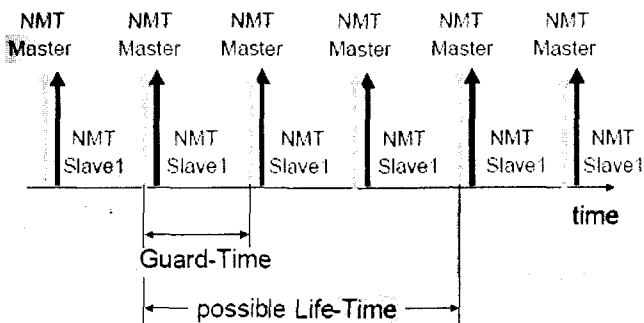


Рис. 9.49. Функціонування Node Guard

Регулярно NMT Master відправляє Дистанційні Кадри і перевіряє дані у відповіді із записом у базі даних. Якщо вони відрізняються, це відображається в прикладній програмі NMT-Ведучого через Network Event Service (рис. 9.50).

9.4.6.2. Heartbeat. Протокол *Heartbeat* (серцебиття) є опціональним і є рекомендованою альтернативою Node Guarding. З певною циклічністю Heartbeat Producer відправляє Heartbeat-повідомлення з частотою, визначеною Heartbeat Producer Time Object (рис. 9.51). Його отримати можуть один або більше Heartbeat Consumer. Якщо Heartbeat Consumer не отримає за певний час цей об'єкт, то буде згенероване повідомлення Heartbeat Event. Даний сервіс функціонує за моделлю Producer-Consumer.

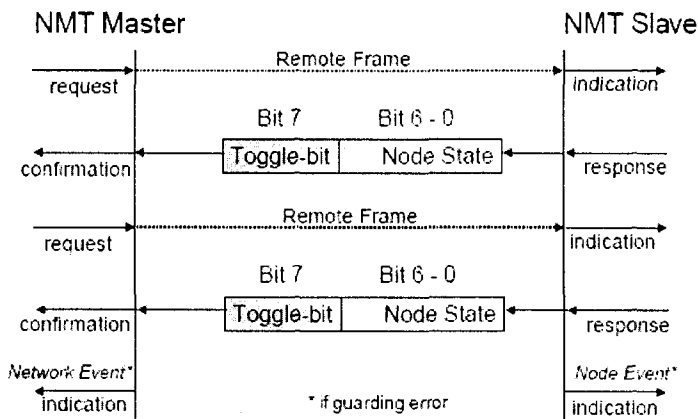


Рис. 9.50. Виявлення дефектного вузла в Node Guard

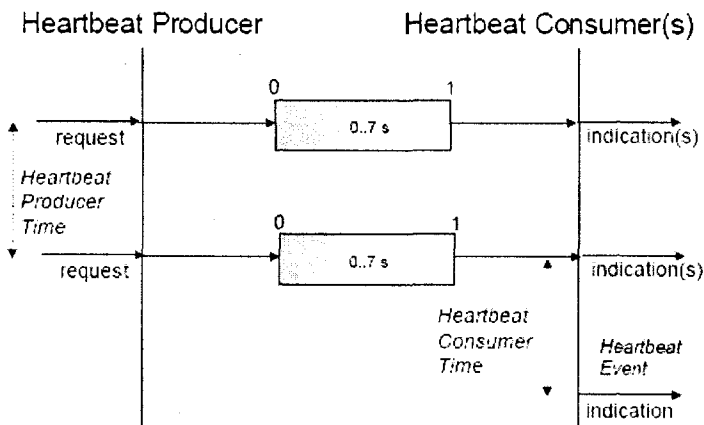


Рис. 9.51. Функціонування протоколу Hertzbeat

9.5. Специфікації профілів пристроїв CANOpen

9.5.1. Профілі пристроїв у CANOpen

CANOpen використовує профілі пристроїв для забезпечення легкої системної інтеграції пристроїв різних виробників (рис. 9.1). Реалізація обов'язкових характеристик пристрою відповідно з CANOpen гарантує коректне функціонування базової мережі. Водночас CANOpen допускає визначення опціональних і спеціально визначених виробником функціональних можливостей приладу. Нині профілі пристроїв CANOpen визначені для широкого спектра різних типів приладів:

- CiA DSP-401: I/O Modules;
- CiA DSP-402: Drives and Motion Control;

- CiA DSP-403: Human Machine Interface;
- CiA WD-404: Measuring Devices and Closed-Loop Controllers;
- CiA DSP-406: Encoders;
- CiA WD-408: Proportional Hydraulic Valves;
- CiA WD-409: Door Control (Railways);
- CiA WDP-4XX: Brake Control (Railways);
- CiA WDP-4XX: Train Bus Gateways.

Слід зазначити, що основна відмінність прикладних профілів від комунікаційних полягає в тому, що комунікаційний профіль визначає порядок доступу прикладної програми одного вузла до об'єктів іншого, а прикладний профіль визначає призначення цих об'єктів та правила їх використання.

Профілі пристроїв в CANOpen визначають:

- підтримувані Прикладні Об'єкти;
- значення комунікаційних параметрів PDO;
- значення PDO-Відображення за замовченням;
- додаткові коди помилок;
- додаткові типи даних.

Кожен пристрій має 32-бітний ідентифікатор, який записаний в Об'єкті **Device Type** — 1000_{16} (таб. 9.11). Молодше слово ідентифікатора вказує на номер профілю пристрою, старше — вміщує додаткову інформацію про пристрій. Для багатомодульних пристроїв, кожний модуль якого підтримує різні профілі, додаткова інформація в Об'єкті Device Type має значення $FFFF_{16}$, а номер профілю вказує на профіль пристрою першого модуля. Профілі всіх інших модулів пристрою записуються в об'єктах $67FF_{16} + x \cdot 800_{16}$ (де x = внутрішній номер модуля пристрою).

Стандартизовані CiA профілі пристроїв використовують Об'єкти Словника з адресами від 6000_{16} по $9FFF_{16}$. Нижче розглянемо один з найбільш популярних профілів пристроїв CANOpen — профілю вводу/виводу, а в розділі 11-профіль для пристроїв PDS (DSP-402).

9.5.2. Профіль модулів вводу/виводу

Профіль I/O Modules описаний у специфікації CiA DSP-401 і призначений для роботи з віддаленими модулями вводу/виводу по мережі CANOpen. Значення молодшого слова вказує на номер профілю (401_{10}), старшого — на тип модуля вводу/виводу:

- 1-й біт — модуль дискретних входів;
- 2-й біт — модуль дискретних виходів;
- 3-й біт — модуль аналогових входів;
- 4-й біт — модуль аналогових виходів.

За замовченням, для даного типу пристрою сконфігуровані два Об'єкти T-PDO та два R-PDO (рис. 9.52), які відповідають за виходи та входи пристрою. В першому об'єкті відображаються 64 дискретні сигнали, в другому — 4 аналогові. За замовченням усі PDO передаються асинхронно з нульовою затримкою (Inhibit Time=0). Залежно від можливості пристрою кількість та конфігурація PDO може змінюватись NMT-Ведучим.

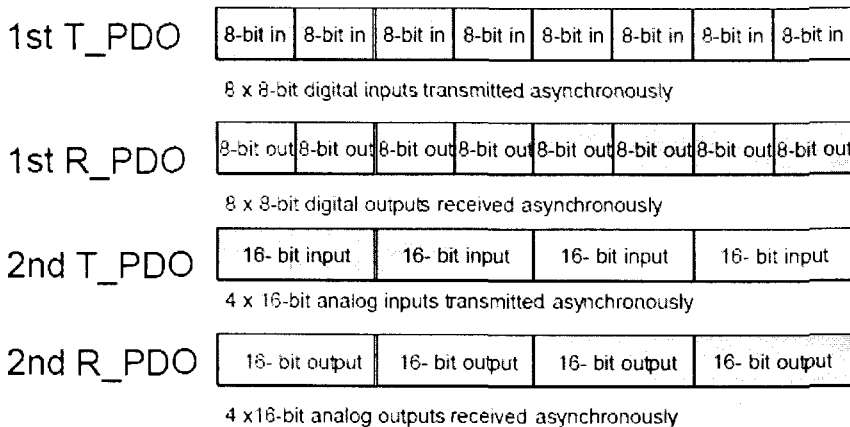


Рис. 9.52. PDO-Відображення за замовченням для модулів вводу/виводу

Окрім визначеного за замовченням профілем доступу до каналів вводу/виводу, CANOpen опціонально підтримує інші методи доступу через Об'єкти Словника.

1. Для дискретних входів:
 - доступ до конкретного біта (6020_{16} - 6027_{16});
 - 2-байтовий доступ (6100_{16});
 - 4-байтовий доступ (6120_{16}).
2. Для дискретних виходів:
 - доступ до конкретного біту (6220_{16} - 6227_{16});
 - 2-байтовий доступ (6300_{16});
 - 4-байтовий доступ (6320_{16}).
3. Для аналогових входів:
 - 1-байтовий доступ (6400_{16});
 - 4-байтовий доступ (6402_{16});
 - специфічний (6404_{16}).
4. Для аналогових виходів:
 - 1-байтовий доступ (6410_{16});
 - 4-байтовий доступ (6412_{16});
 - специфічний (6414_{16}).

Крім того, опціонально підтримуються додаткові функції перетворення. Для дискретних входів можливі такі додаткові функції (рис. 9.53):

- зміна полярності логіки (6002_{16});
- відправка T-PDO за переднім фронтом сигналу входу (6008_{16});
- відправка T-PDO за заднім фронтом сигналу входу (6007_{16});
- відправка T-PDO за будь-якою зміною (6006_{16});
- активація/деактивація відправки за зміною значення (6005_{16}).

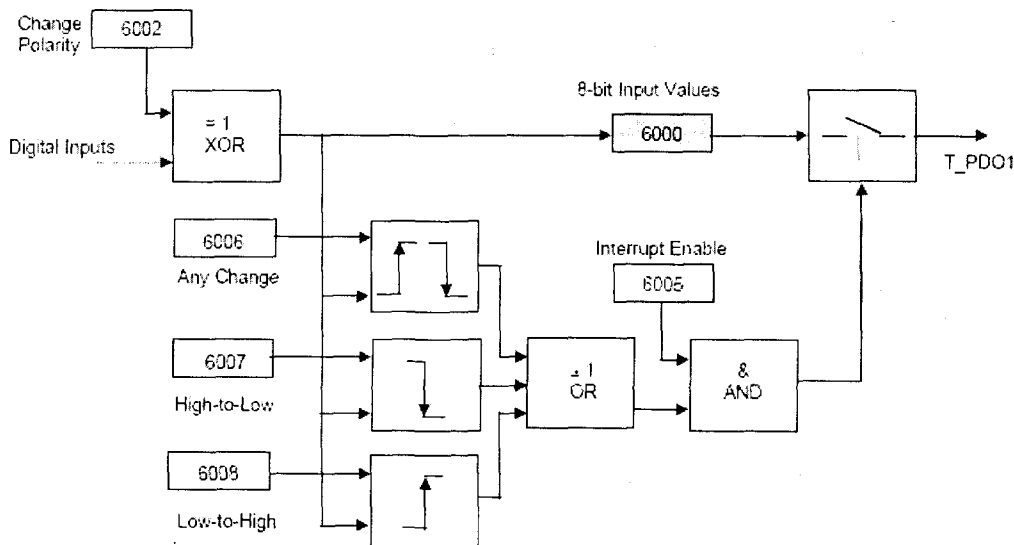


Рис. 9.53. Схема перетворення для перших 8-ми дискретних входів (6000₁₆)

Для дискретних виходів визначені такі можливості (рис. 9.54):

- зміна полярності логіки (6202₁₆);
- настройка фільтра (6208₁₆);
- визначення режиму стану виходів при аварійній ситуації (6206₁₆);
- визначення значення виходів при аварійній ситуації (6207₁₆).

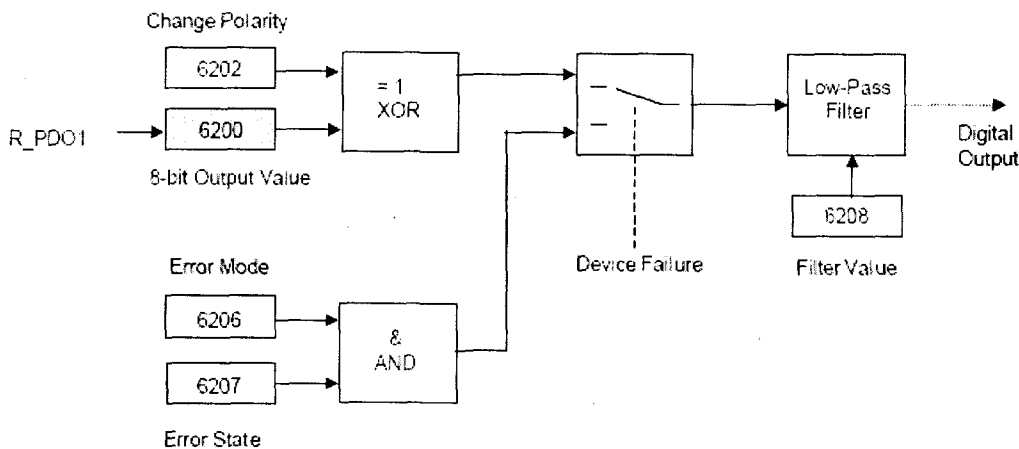


Рис. 9.54. Схема перетворення для перших 8-ми дискретних виходів (6200₁₆)

Для аналогових входів визначені такі можливості (рис. 9.55):

- вхідний діапазон для масштабування (6420₁₆);
- зміщення для масштабування (642E₁₆);
- коефіцієнт масштабування (642F₁₆);
- настройки події активації передачі PDO:
- верхня межа (6424₁₆);
- нижня межа (6425₁₆);
- зона нечутливості (6426₁₆, 6427₁₆, 6428₁₆);
- активація/деактивація відправки за зміною значення (6423₁₆).

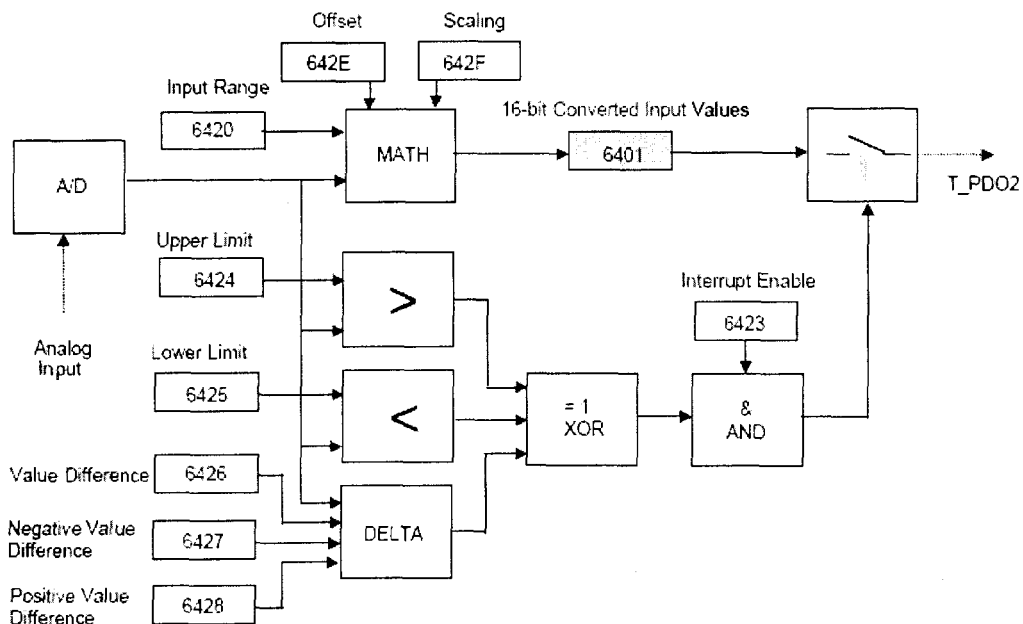


Рис. 9.55. Схема перетворення для першого аналогового входу (6401₁₆)

Для аналогових виходів визначені такі можливості (рис. 9.56):

- вхідний діапазон для масштабування (6440₁₆);
- зміщення для масштабування (6441₁₆);
- коефіцієнт масштабування (6442₁₆);
- визначення режиму стану виходів при аварійній ситуації (6443₁₆);
- визначення значення виходів при аварійній ситуації (6444₁₆).

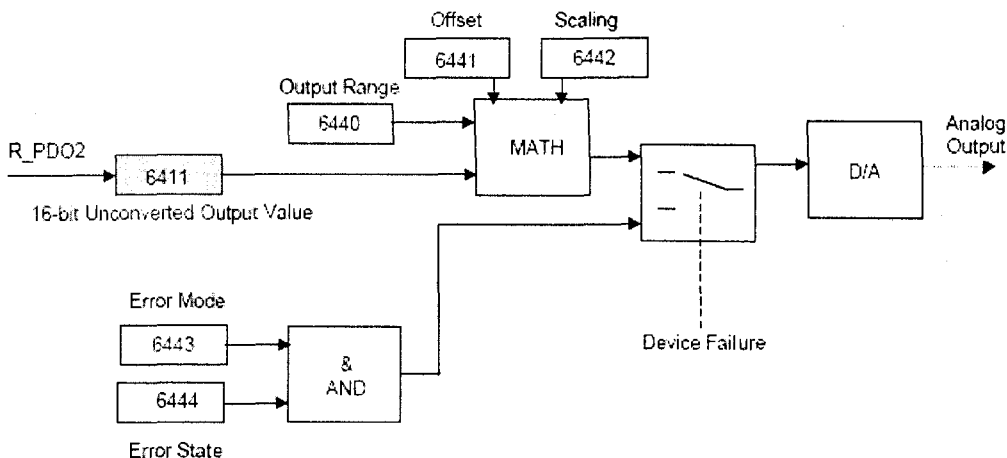


Рис. 9.56. Схема перетворення для першого аналогового виходу (6411_{16})



Контрольні запитання до розділу 9

1. Поясніть відмінність між CAN та CANOpen.
2. Поясніть функціонування CANOpen у контексті моделі OSI. Які стандарти використовує мережа CANOpen на різних рівнях?
3. На яких стандартах на фізичному рівні базується CANOpen і які додаткові вимоги ставляться на цьому рівні? Наведіть основні характеристики CANOpen на фізичному рівні: спосіб передачі бітів, топологія, правила термінування, максимальна довжина ліній зв'язку та відгалужень, бітова швидкість, максимальна кількість пристроїв.
4. Які кабелі, роз'єми та призначення контактів рекомендуються для використання в CANOpen? Який стандарт це визначає?
5. Розкажіть про призначення Словника Об'єктів. Яким чином визначається місце розміщення Об'єктів у Словнику?
6. Яким чином Прикладні Об'єкти передаються на каналному рівні? Де містяться настройки комунікаційного обміну?
7. Поясніть модель функціонування пристрою-вузла CANOpen.
8. Перерахуйте комунікаційні сервіси, які використовуються в CANOpen. Поясніть їх призначення.
9. Які Комунікаційні Об'єкти використовуються для реал-тайму обміну невеликими обсягами даних процесу? Які Комунікаційні Об'єкти використовуються для ациклического обміну великими обсягами даних?
10. Що забезпечують функції сервісу PDO? Які комунікаційні Об'єкти використовуються для функціонування даного сервісу?
11. Поясніть призначення T-PDO та R-PDO. Як забезпечується їх взаємний зв'язок та зберігається інформація про нього? Як передається T-PDO на каналному рівні?
12. Як забезпечується зв'язок даних T-PDO/R-PDO з Об'єктами Словника? Як називається цей механізм? Де зберігаються настройки таких зв'язків?

13. Як забезпечується зв'язок даних T-PDO з даними R-PDO? Де зберігаються настройки параметрів комунікаційного обміну?
14. Яке призначення NMT-сервісів? Яке призначення NMT-Ведучого в CANOpen та які функції він виконує? Поясніть різницю між NMT-Ведучим та Ведучим шини.
15. Які способи конфігурування можуть бути доступні розробнику для конфігурування NMT-Ведених?
16. Прокоментуйте роботу діаграми станів NMT-Веденого. Які сервіси доступні в кожному із станів? За допомогою яких Об'єктів налаштовується конфігурація вузла NMT-Веденого перед переводом його в операційний режим?
17. Яким чином відбувається перехід NMT-Веденого з одного стану в інший? Які об'єкти для цього використовуються? Як ідентифікується конкретний вузол у CANOpen і для чого це потрібно?
18. Яким чином забезпечується зв'язок між Комунікаційними Об'єктами вузлів до їх функціонування в операційному режимі? Поясніть механізм розподілу Ідентифікаторів за замовченням.
19. Яке правило наперед визначеного статичного зв'язування PDO в мережі? Що таке динамічне зв'язування PDO?
20. В чому необхідність знання структури Словника Об'єктів конкретного засобу? Який ресурс необхідно використати для її визначення? Прокоментуйте загальноприйнятий розподіл Індексів різних типів Об'єктів у Словнику.
21. Що забезпечують функції сервісу SDO? Які комунікаційні Об'єкти використовуються для функціонування даного сервісу? На якій моделі прикладного обміну базуються сервіси SDO?
22. Яким чином забезпечується обмін даними за допомогою об'єктів SDO? Як дані SDO-об'єктів зв'язуються з даними Прикладних Об'єктів Словника? Як забезпечується передача даних великого обсягу?
23. Які режими ініціації обміну об'єктів PDO є в CANOpen? Поясніть їх функціонування? До яких моделей обміну на прикладному рівні вони належать?
24. Як забезпечується синхронізація передачі PDO в чітко визначені моменти часу? Що таке синхронне вікно? Який вузол генерує синхронні об'єкти?
25. Які комунікаційні режими передачі (Transmission Type) визначені в CANOpen? Прокоментуйте призначення кожного режиму. Де зберігається налаштування режиму для кожного PDO?
26. Яким чином у CANOpen забезпечується синхронізація годинників реального часу для різних вузлів?
27. Яким чином у CANOpen вузол повідомляє про наявність помилки? Які сервіси використовуються для визначення станів NMT-Ведених та NMT-Ведучих вузлів на шині?
28. Які переваги дає профілювання пристроїв для використання їх у CANOpen? Які типи профілів пристроїв для CANOpen Ви можете назвати? Яким чином в Об'єкті Словника вказується його приналежність до конкретного профілю?
29. Які профілі для модулів вводу/виводу CANOpen Ви можете назвати? Як за замовченням налаштовується для них PDO-Відображення?
30. Які Прикладні Об'єкти Словника використовуються за замовченням для налаштування перетворення дискретних входів, дискретних виходів, аналогових входів, аналогових виходів у пристроях відповідних профілів?

ПРОМИСЛОВИЙ ETHERNET

У 2-му розділі показано, як необхідність у формуванні єдиного у світі стандарту на промисловій мережі привела до так званої fieldbus-війни. Це сприяло розробці нових промислових мереж (наприклад, FF) та стандартів (наприклад EN 50170) і закінчилось випуском у 2000 р. універсального модульного стандарту IEC 61158. У цей час в офісних системах найпопулярнішими стандартами де-факто стали мережі, які базуються на Ethernet та TCP/IP і які практично витіснили всі інші аналогічні рішення. Затрати на побудову інфраструктури цих мереж перманентно зменшувалися у зв'язку з великою кількістю комунікаційного обладнання, доступного на ринку. Сьогодні практично все офісне обладнання можна об'єднати за допомогою технологій Internet/Intranet та мереж Ethernet. Серед переваг цих технологій можна виділити їх відкритість, популярність, доступність, швидкодію та постійний розвиток. Усе це спонукало до пошуку рішень для їх використання при інтеграції організаційно-економічного рівня з рівнем АСУТП а також на самому рівні АСУТП, в якості промислової мережі.

Слід зазначити, що технології на базі TCP/IP (Intranet технології), які містяться на мережному та транспортному рівнях, не передбачають обов'язкову реалізацію нижніх рівнів на базі Ethernet. Для офісних систем використовуються й інші мережі. Однак промислові мережі не підтримували TCP/IP, тому для прозорої взаємодії АСУТП та верхніх рівнів управління використання Ethernet було найбільш перспективним. Це стало ще одним аргументом на користь Ethernet на рівні промислових мереж.

Використання Ethernet на рівні АСУТП пов'язано з багатьма труднощами. Навіть сьогодні частина кваліфікованих фахівців у галузі автоматизації скептично ставляться до будь-якої можливості використання Ethernet на рівні промислових мереж. Це, перш за все, пов'язано з особливостями самої технології Ethernet. Утім, на рівні промислових мереж використовувати Ethernet почали одночасно багато світових виробників. Всі вони мали вирішувати проблеми, які пов'язані із зовсім не промисловим виконанням стандартних технологій Ethernet, зокрема:

- забезпечення детермінованого часу обміну;
- забезпечення синхронізованої у часі передачі між польовими засобами (наприклад, пристроями PDS);

- забезпечення ефективного і частого обміну даними процесу в реальному часі;
- промислові умови експлуатації.

Поряд з цим одночасно повинні були функціонувати стандартні технології Intranet.

Даний розділ присвячений особливостям реалізації ряду рішень для Ethernet, які увійшли в стандарти МЕК. На початку нагадаємо принципи реалізації базових Ethernet технологій та протоколів IP та TCP

10.1. Мережі Ethernet

10.1.1. Загальні поняття

10.1.1.1. Походження. Технологія і перші стандарти *Ethernet* були розроблені компанією Xerox Corporation. Пізніше група компаній DEC, Intel і Xerox спільно розробили стандарт під назвою DIX, який став другою версією свого попередника (*Ethernet II*). Ще через деякий час подібний стандарт був розроблений інститутом IEEE (Інститут інженерів з електротехніки і радіоелектроніки), який є законодавцем у галузі стандартів на локальні комп'ютерні мережі. Цей стандарт називається IEEE 802.3 і має деякі відмінності у формуванні кадру порівняно з форматом Ethernet II. Сьогодні мережні карти Ethernet, як правило, підтримують обидва стандарти. В якості доступу до шини використовується метод CSMA/CD, який розглянутий у другому розділі. Для адресації вузлів у мережі послуговуються 48-бітним (6 байтів) ідентифікатором, який називається *MAC-адресою*.

IEEE використовує відмінну від ISO OSI ієрархічну мережну модель. Тому каналний рівень OSI представлений підрівнями LLC та MAC. MAC-рівень описаний в *IEEE 802.3*, а LLC — в *IEEE 802.2*. Архітектура Ethernet показана на рис. 10.1.

10.1.1.2. Фізичний рівень. На фізичному рівні Ethernet дозволяє використовувати різноманітні середовища передачі даних, для кожної з яких є стандартна назва виду XBaseY, де *X* — бітова швидкість у Мбіт/с (10, 100, 1000 ...), Base — ключове слово (позначає немодульовану передачу), *Y* — умовне позначення середовища передачі і дальності зв'язку. Раніше, як правило, використовувався коаксіальний кабель та шинна топологія, однак, у сучасних реалізаціях Ethernet частіше зустрічається кабель типу «вита пара» або оптоволоконний і топологія типу «зірка», інколи використовується з'єднання типу точка-точка. У центрі зірки може бути *концентратор* (хаб — *hub*) або *комутатор* (*switch*). У табл. 10.1 показані стандарти мереж Ethernet на фізичному рівні.

канальний	Ethernet II	LLC IEEE 802.2
		MAC IEEE 802.3
фізичний		IEEE 802.3

Рис. 10.1. Стандарти нижніх рівнів мереж Ethernet

Таблиця 10.1

СТАНДАРТИ МЕРЕЖІ ETHERNET

Стандарт	Тип стандарту	Тип кабелю	Топологія	Довжина зв'язку/променя	Інтерфейс конекторів, коментар
10Base5	Ethernet 10Mbit/c IEEE 802.3	товстий коаксіальний	Ш	500м	AUI, підключення до трансивера на кабелі через «зуб вампіра»
10Base2		тонкий коаксіальний	Ш	185м	BNC
10BaseT		UTP категорія 3 та вище (2 пари проводів)	3	100м	RJ-45
10BaseF		оптоволокно	3	1–10 км	AUI
100BaseTX	Fast Ethernet 100Mbit/c IEEE 802.3u	UTP кат. 5 та вище (2 пари проводів)	3	100 м	RJ-45
100BaseT4		UTP кат. 3 та вище (4 пари проводів)	3	100м	RJ-45
100BaseFX		оптоволокно одномодове	3	10 км	ST, SC, MT-RJ...
100BaseSX		оптоволокно багатомодове	3	300 м	сумісний з 10BaseF, автоузгодженість
1000BaseCX	Gigabit Ethernet 802.3z	STP (2 пари проводів)	3	25 м	RJ-45
1000BaseT		UTP категорія 5 та вище (4 пари проводів)	3	100 м	RJ-45
1000BaseSX		оптоволокно багатомодове	3	200–500 м	ST, SC, MT-RJ...
1000BaseLX		оптоволокно одномодове	3	10 км	ST, SC, MT-RJ...

10.1.1.3. Використання витої пари. При використанні витої пари пристрої підключають через роз'єми типу RJ-45. На пристроях використовують розетки, а на кабелях — вилки. Призначення контактів роз'єму мережного адаптера (порт типу *MDI*) показано в табл. 10.2. Під'єднання в мережу вузлів проводиться за допомогою спеціальних пристроїв — концентраторів та комутаторів.

Порти концентраторів мають тип *MDIX*, в яких сигнали Rx та Tx поміняні місцями. Таким чином, при підключенні пристрою з MDI (мережна карта) до MDIX (концентратор/комутатор) використовується «прямий кабель» (рис. 10.3, а), а при безпосередньому підключенні двох MDI — «перехресний кабель» (рис. 10.3, б).

Схема «прямого» та «перехресного» кабелю показана на рис. 10.4 (а — пряме, б — перехресне).

У телекомунікаційних пристроях (комутаторах та концентраторах) один із портів може використовуватися з перемикачем MDI-MDIX або може існувати додатковий роз'єм для підключення його до концентратора/комутатора верхнього рівня (up-link). Сьогодні комутатори, як правило, підтримують автоматичне

визначення типу підключеного кабелю (*Auto-MDI/MDI-X*) і тому не потребують додаткового порту up-link.



Рис. 10.2. Призначення контактів роз'єму RJ-45

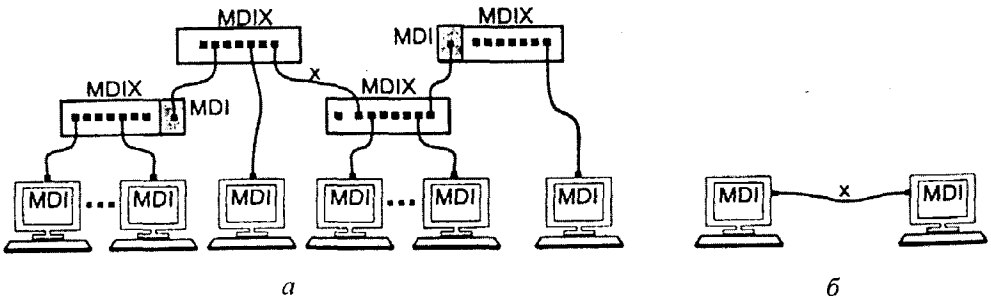


Рис. 10.3. Схеми підключень пристроїв мережі Ethernet

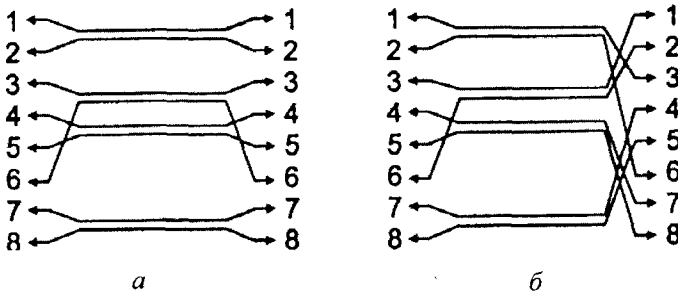


Рис. 10.4. Схема з'єднань: а — MDI до MDIX; б — MDI до MDI

10.1.1.4. Концентратори (hub). У найпростішій схемі мережі 10BaseT центральний елемент зірки являє собою активний елемент, який називається хабом (hub) або концентратором. Кожна станція з'єднується з ним двома витими пара-

ми — для прийому і для передачі. Він також має функції повторювача. Хоч фізично така схема являє собою зірку, логічно вона є шиною, адже передача кожної станції приймається всіма іншими станціями, що може призводити до колізій при одночасній передачі. Концентратори можуть об'єднуватись в ієрархічну структуру із головного та проміжних хабів. Концентратори в комп'ютерних мережах на базі Ethernet практично вже не використовуються, однак у промислових мережах вони знайшли своє застосування (наприклад, CC-Link IE Field).

10.1.1.5. Комутатори 2-го рівня (switch L2). Комутатори 2-го рівня (layer 2 switch) працюють за іншим принципом. Кадр, який надходить на комутатор за одним портом, направляється тільки в той порт, до якого підключений вузол-приймач. Тобто, комутатори 2-го рівня аналізують кадр, та за MAC-адресою приймача визначають, куди необхідно його направити. Відбувається комутація між портами, тобто створюється канал, який не заважає створювати такі самі канали між іншими портами. Це в свою чергу виключає можливість колізії, тому немає необхідності у використанні алгоритму CSMA/CD. Поява таких комутаторів дала можливість працювати в дуплексному режимі, що значно збільшило ефективність роботи Ethernet. У цьому режимі використовується такий же формат кадру MAC-стандарту.

За способом передачі кадрів комутатори діляться на:

- *store-and-forward* (з проміжним збереженням) — комутатор отримує з вхідного каналу весь кадр, аналізує його на наявність помилок і у випадку їх відсутності відправляє кадр на вихідний канал отримувача;
- *cut-through* (наскрізний) — комутатор починає відправляти дані у вихідний порт призначення відразу, як тільки визначить MAC-адресу отримувача, тобто без очікування кінця передачі та аналізу цілісності отриманого кадру.

Вибір того чи іншого комутатора залежить від протоколів верхніх рівнів, що базуються на Ethernet. Перший тип комутаторів працює більш повільно, однак відкидає пошкоджені кадри вже в момент їх отримання.

Сучасні комутатори 2-го рівня можуть надавати ряд додаткових сервісів. Наприклад, це комутатори з підтримкою технології VLAN, виділення пріоритетних портів, додаткова MAC-фільтрація, IGMP snooping (потребує додаткової обробки на мережному рівні). Такий тип комутаторів потребує конфігурування, тому отримали назву *керовані комутатори (managed switches)*. Для конфігурування та керування керованими комутаторами можуть надаватися різні типи сервісів, наприклад WEB, SNMP.

10.1.1.6. Auto-negotiation. Нині більшість телекомунікаційних пристроїв підтримують технологію *Auto-negotiation (авто-переговор)*, що дозволяє автоматично налаштувати оптимальний зв'язок між пристроями. В момент ініціалізації зв'язку (при включенні або підключенні пристрою в мережі) за спеціальним протоколом пристрої домовляються про швидкість передачі та тип з'єднання (дуплекс/напівдуплекс). Автопереговори дають можливість відмовитись від ручного налаштування мережних карт. Однак слід зазначити, що для використання технології автопереговорів, усі вузли, які зв'язуються, повинні її підтримувати, інакше можлива втрата кадрів або взагалі відсутність обміну.

10.1.1.7. Структуровані кабельні системи (СКС). Повсякденне використання Ethernet в офісних будівлях призвело до необхідності організації робочих місць розетками для підключення комп'ютерів до мережі. Таким чином, при проектуванні виникала потреба в правильному плануванні кабельної системи будівлі, що включала всі телекомунікаційні ланцюги (комп'ютерна мережа, телефон, аудіо, відео тощо). В результаті спільної роботи комітетів ISO та IEC виник стандарт ISO/IEC 11801 на організацію універсальних структурованих кабельних системи (СКС) в офісних приміщеннях.

10.1.2. Структура кадрів

10.1.2.1. DIX Ethernet II та IEEE 802.3. Нагадаємо, що нині існує два формати кадрів Ethernet. Це пов'язано з тим, що в той час, коли організація IEEE розвивала стандарти для локальних мереж, DIX Ethernet уже широко використовувався. Мета IEEE полягала в тому, щоб розвинути стандарти та правила, які були б базовими для всіх типів обчислювальних мереж так, щоб між ними був неможливий обмін даними (Ethernet та Token Ring). Оскільки це було потенційно небезпечним для існуючих впроваджень Ethernet, правила обробки для «нового» і «старого» формату були різними. З тих пір існує відмінність між кадрами DIX Ethernet та IEEE 802.3, хоч обидва називаються Ethernet.

Кадри DIX та IEEE 802.3 ідентичні за кількістю та довжиною полів (рис. 10.5). Єдина відмінність полягає в змісті полів та їх інтерпретації станціями, які відправляють та отримують кадри. Враховуючи це, будь-яка станція може відправити будь-який з цих кадрів.

Проаналізуємо структуру кадрів. З рис. 10.5 видно, що відрізняється структура преамбули, однак наповнення та інтерпретація цієї частини буде однаковою. Поля адреси призначення та адреси відправника однакові і призначені для адресації за допомогою MAC-адреси.

Призначення поля — довжина/тип (які також називаються EtherType) дещо відрізняються в стандартах. Для DIX це поле описує тип PDU вищого рівня (мережного), який переноситься даним кадром. Наприклад 800_{16} вказує, що кадр використовується для пакета IP. У кадрі IEEE 802.3 дане поле використовується як для вказівки типу PDU, так і для опису довжини кадру. Якщо значення цього поля $< 600_{16}$, воно вказує на довжину поля даних, а в іншому випадку — на тип протоколу даних.

Оскільки каналний рівень у стандартах IEEE включає підрівень LLC, поле даних в IEEE 802.3 включає частину, яка належить до IEEE 802.2.

10.1.2.2. IEEE 802.1Q та VLAN. Для підтримки керованих мостів та комутаторів IEEE випустила додатковий стандарт 802.1Q. Стандарт IEEE 802.1Q забезпечує незалежний від виробника спосіб реалізації віртуальних мереж VLAN (Virtual bridged Local Area Network) за рахунок виділення частини портів комутатора в незалежні групи перемикання. Це дозволяє менеджеру мережі призначити окремі групи портів незалежним віртуальним мережам. Розподіл між віртуальними мережами проходить за виділеними для цього полями кадру (рис. 10.6).

На відміну від стандартного кадру IEEE 802.3, після адреси відправника вставляються додаткові 4 байти: 2 байти TPID та два байти TCI.

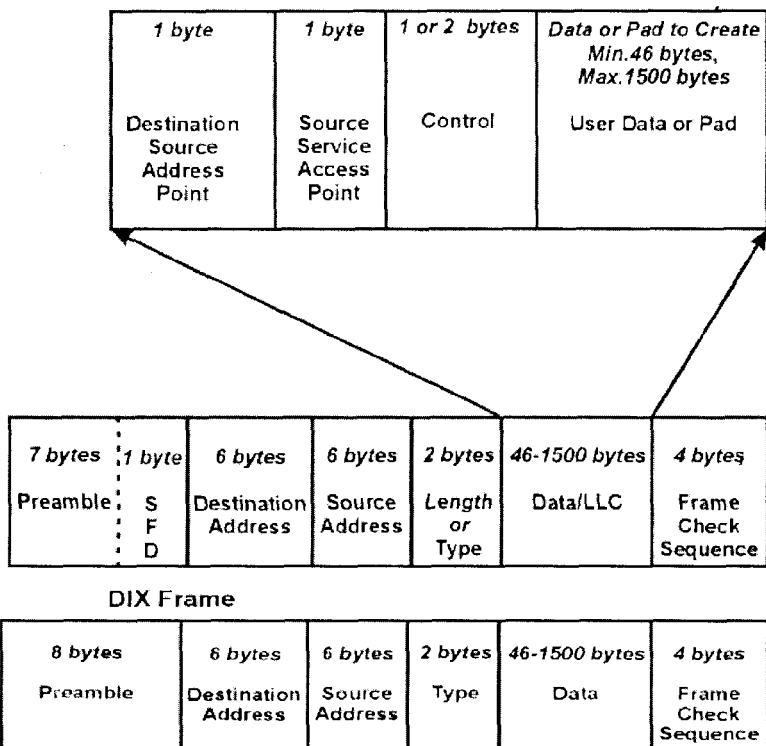


Рис. 10.5. Структура кадрів IEEE 802.3 та DIX

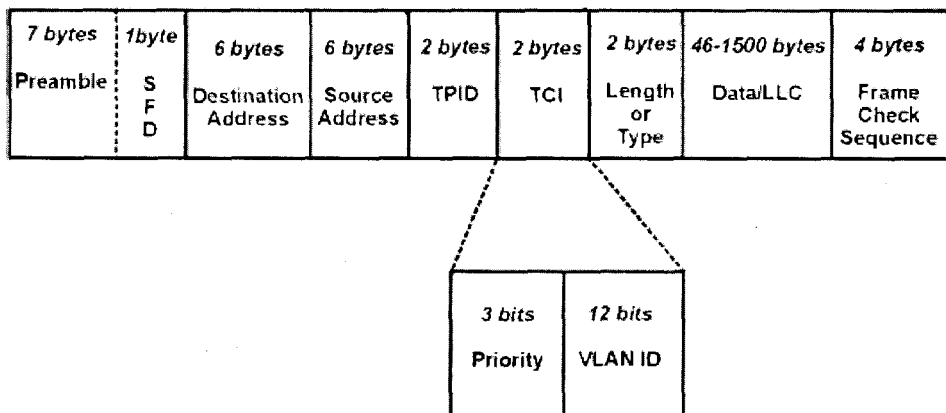


Рис. 10.6. Структура кадрів IEEE 802.3 з розширенням IEEE 802.1Q

Поле TPID — це ідентифікатор протоколу. Призначення цього поля полягає в ідентифікації даного кадру, як особливого (тегового). Зустрічаючи такий кадр, комутатори або мости, які підтримують додаткові можливості 802.1Q почнуть обробляти його по особливому алгоритму. Для Ethernet значення цього поля 8100_{16} . Поле TCI вміщує інформацію для управління теговим кадром. Перші три біти дають інформацію про пріоритетність кадру, яка базується на 8-ми рівнях *CoS* (Class of Service, 0-7, 7-й — найбільш пріоритетний), описаних в IEEE 802.1p. Останні 12 бітів ідентифікують віртуальну мережу VLAN (0-4095), для якої призначений даний кадр.

Кадри VLAN можуть одночасно функціонувати зі стандартними кадрами IEEE 802.3 та DIX. Використання VLAN потребує керованих елементів мережної інфраструктури — керовані мости, керовані комутатори тощо.

10.2. Протоколи рівня IP

10.2.1. Протокол IP

IP (Internet Protocol) — протокол мережного рівня, який в основному служить для маршрутизації пакетів між мережами. Принципи маршрутизації наведені в другому розділі. Коротко розглянемо його функціонування та супутні протоколи.

Сьогодні існують дві версії протоколу IP: *IPv4* та *IPv6*. Хоч остання версія більш перспективна, поки що, як правило, користуються *IPv4*, тому надалі буде розглядатися тільки цей протокол. Для адресації пристроїв відправників та отримувачів використовуються унікальні 32-бітні (4-байтові) *IP-адреси*. Сьогодні використовують безкласову адресацію: для адреси використовується пара — IP-адреса + 32-бітна маска. За маскою сутність рівня IP виділяє підмережі. Наприклад: 192.168.1.2 — адреса вузла, а 255.255.255.0 — маска. У цьому випадку 192.168.1 — підмережа, а 2 — адреса вузла в цій підмережі.

Однак IP-дейтаграма упаковується в кадр канального рівня, тому канальний рівень на вузлі отримувача не зможе визначити, що кадр призначений саме йому. Якщо, скажімо, на нижніх рівнях використовується Ethernet, а на мережному IP, то мережна карта в кадрі буде розпізнавати MAC-адресу, а не IP-адресу отримувача, адже остання призначена для адресації в інтермережах. Тому для ідентифікації вузла-отримувача MAC-адресою за його IP-адресою використовуються спеціальні алгоритми, наприклад ARP.

Слід зазначити, що IP-заголовок, який добавляється на мережному рівні, крім адрес відправника та отримувача, вміщує додаткову інформацію. Так, для управління пріоритетами IP-пакетів служить поле *ToS* (Type of service), яке може бути використане в промислових мережах.

10.2.2. Протокол ARP

Коли IP-сутність мережного рівня отримує завдання від верхніх рівнів на пересилку певних даних, то вона аналізує задану адресу IP вузла отримувача. Якщо

вузол отримувача міститься в тій самій підмережі, що і даний вузол, необхідно визначити Ethernet MAC-адресу отримувача за його IP-адресою. Одне з рішень — зберігати в системі файл з таблицею відповідності. Однак, при збільшенні кількості вузлів у системі такі таблиці збільшуються, через що необхідно додати записи у ці таблиці. Крім того, зміна мережної карти або IP-адреси вузла теж призводить до зміни таблиці.

Для автоматичного визначення MAC-адреси за IP-адресою існує **ARP**-протокол (Address Resolution Protocol). Його функціонування розглянемо на прикладі. Коли вузлу-відправнику (вузол1) необхідно реалізувати поставлене вище завдання, він у ширококомовному режимі відправляє пакет із запитом ідентифікації MAC-адреси вузла за його IP. Цей пакет буде отриманий усіма вузлами, а вузол із запитуваним IP (вузол 2) підтвердить запит зі своєю MAC-адресою. Знаючи MAC-адресу, вузол1 вказує отриману адресу в кадрі Ethernet як адресу отримувача.

Для зменшення навантаження на шину вузол може зберігати адреси у вигляді таблиці відповідності в ARP-кеші, записи якого старіють через декілька хвилин після відновлення і вилючаються. Крім того, при завантаженні вузол відправляє свій IP+MAC іншим машинам, а ті записують його в свій ARP-кеш. Для перегляду власної ARP таблиці можна в командному рядку Windows вказати: >> *ARP* – *a*.

10.2.3. Протоколи BOOTP та DHCP

Інколи, при завантаженні, вузлу необхідно автоматично отримати свою IP-адресу і конфігураційні настройки з іншого вузла-серверу, який зберігає ці настройки (автоматичне налаштування IP). Особливо це актуально для систем розподіленого вводу/виводу з використанням Ethernet TCP/IP. Сьогодні для цього використовуються технології BOOTP та DHCP.

У протоколі **BOOTP** (Bootstrap Protocol) клієнт (якому необхідний IP) відсилає ширококомовне UDP-повідомлення із запитом на отримання IP-адреси за своєю MAC-адресою. На BOOTP-сервері повинна бути налаштована таблиця відповідності MAC — IP. Отримавши такий пакет, сервер відішле даному вузлу-клієнту пакет, в якому вкаже його IP, маску підмережі та IP-адресу вузла, де міститься образ пам'яті вузла-клієнта.

Недоліком протоколу BOOTP є необхідність обов'язково конфігурування на BOOTP-сервері таблиці відповідності. Тому альтернативою до нього став протокол **DHCP** (Dynamic Host Configuration Protocol — протокол динамічної настройки хостів), який дозволяє настроювати таблиці адрес як вручну, так і автоматично. Принципи функціонування подібні до BOOTP-протоколу, однак DHCP-сервер може і не знати про існування вузла з конкретною MAC-адресою. У цьому випадку, при запиті DHCP-клієнта на отримання IP адреси, DHCP-сервер виділяє цю адресу з *пула адрес* (виділеного діапазону) і надає клієнту на певний час (час оренди). Через певні проміжки часу (приблизно половина часу оренди), клієнт надсилає запит на продовження використання виділеного IP, тим самим продовжуючи час оренди.

Приклад 10.1. Промисловий Ethernet. Налаштування Ethernet у системах Schneider Electric

Завдання. Створити конфігурацію для комунікацій Ethernet-пристроїв, показаних на схемі рис. 10.7, якщо:

- PLC1 має статичну IP адресу;
- PLC2 (MAC=00.80. F4.01.7B.D4) отримує IP-адресу з PLC1 за протоколом BOOTP;
- RIO1 (MAC=00.00.54.11.EC.0A) отримує IP-адресу з PLC1 за протоколом BOOTP.

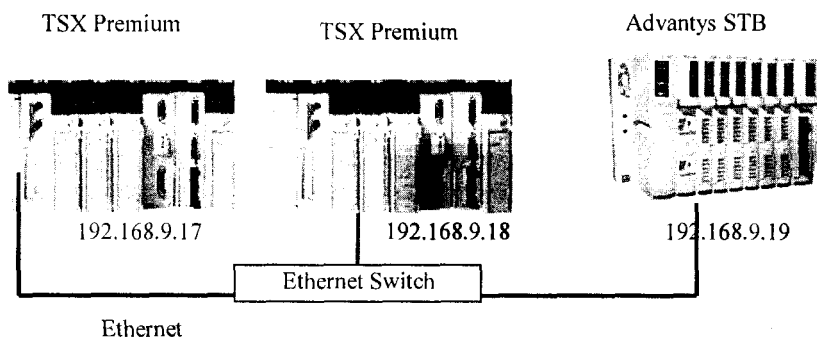


Рис. 10.7. Структурна схема системи до прикладу 10.1

Рішення. Інтегрований порт PLC1 налаштовується в процесі конфігурування ПЛК у програмному середовищі PL7 або Unity (рис. 10.8). Оскільки адреса IP є статичною, вказуємо її у відповідному полі вкладки Messaging. Маску підмережі вказуємо як 255.255.255.0. Шлюз вказувати не треба, так як використовується тільки локальне з'єднання. Тип Ethernet (DIX Ethernet II або IEEE 802.3) вибирається залежно від комунікаційного обладнання, наприклад комутатора. Конфігурація BOOTP-серверу виконується шляхом надання таблиці відповідності MAC-адреси та IP-адреси згідно з поставленим завданням (рис. 10.8).

Для ПЛК2 на вкладці Messaging виставляється опція Client/Server configuration, що вказує на необхідність для ПЛК2 отримання IP-адреси з BOOTP-серверу. Після переконфігурування ПЛК2 він отримує свою IP-адресу відповідно до своєї MAC.

Для комунікаційного модуля RIO1 — STB NIP 2212 необхідність в отриманні IP-адреси за MAC виставляється поворотним перемикачем на модулі (позиція BOOTP).

TSX ETY PORT [RACK 0 POSITION 1]

Designation: TCP/IP 10/100 MODULE

Module IP address
 IP address: 192 168 9 17 Subnetwork mask: 255 255 255 0 Gateway address: 0 0 0 0

Module utilities
 IO Scanning Global data
 Address server Bandwidth

Messaging | IO Scanning | **Address server** | SNMP | Global Data | Bandwidth | Bridge

XWAY address
 Network [1] Station [17]

IP address configuration
 Configured
 IP address: 192 | 168 | 9 | 17
 Subnetwork mask: 255 | 255 | 255 | 0
 Gateway address: 0 | 0 | 0 | 0

Client/Server configuration

Ethernet configuration
 Ethernet II 802.3

Access control

Xway Addr.	IP address	Protocol	Access	Mode
1		UNITE	<input checked="" type="checkbox"/>	MULTI
2		UNITE	<input checked="" type="checkbox"/>	MULTI
3		UNITE	<input checked="" type="checkbox"/>	MULTI
4		UNITE	<input checked="" type="checkbox"/>	MULTI
5		UNITE	<input checked="" type="checkbox"/>	MULTI
6		UNITE	<input checked="" type="checkbox"/>	MULTI
7		UNITE	<input checked="" type="checkbox"/>	MULTI
8		UNITE	<input checked="" type="checkbox"/>	MULTI
9		UNITE	<input checked="" type="checkbox"/>	MULTI
10		UNITE	<input checked="" type="checkbox"/>	MULTI
11		UNITE	<input checked="" type="checkbox"/>	MULTI
12		UNITE	<input checked="" type="checkbox"/>	MULTI

Messaging | IO Scanning | **Address server** | SNMP | Global Data | Bandwidth | Bridge

Utility
 BOOTP DHCP (FDR)

HTTP modification
 Locked in operation

Client/server address table

	MAC address	Name	IP address	Netmask	Gateway
1	00:80:F4:017B:D4		192.168.9.18		
2	00:00:54:11EC:04		192.168.9.13		

Рис. 10.8. Налаштування порту Ethernet в ПЛК Premium

Приклад 10.1

10.2.4. Протоколи IGMP та IGMP-snooping

IGMP — це протокол управління груповою передачею даних у мережах, що базуються на протоколі IPv4. Він використовується маршрутизаторами та IP-вузлами для об'єднання мережних пристроїв у групи. Для цього для багатоадресної розсилки зарезервовані IP-адреси і призначаються з діапазону адресів з 224.0.0.0 по 239.255.255.255. Кожній групі багатоадресної розсилки відповідає одна IP-адреса з цього діапазону. Тобто вузли, що будуть входити до цієї групи, будуть отримувати дані за єдиною груповою адресою. Для функціонування IGMP необхідно, щоб маршрутизатор та вузли підтримували цей протокол. Про членство вузлів у групах повідомляють самі вузли, а стан членства періодично перевіряється маршрутизаторами, які підтримують багатоадресну розсилку.

Для запобігання ширококомунікаційній ретрансляції комутаторами 2-го рівня багатоадресного трафіка, створеного сервісами IGMP, ряд з них підтримують функцію **IGMP snooping**. Ця функція полягає в тому, що, отримавши запит про підключення

чення/відключення вузла до IGMP-групи, комутатор включає/виключає його порт у/з список членів цієї групи. Як бачимо, комутатори з такою функціональністю забезпечують маршрутизацію не тільки на каналному рівні, а і на мережному.

10.2.5. Методика NAT-трансляції

Коли корпоративний клієнт хоче підключитися до Internet, то кожному вузлу в даній корпорації необхідно виділити унікальну в мережі Internet адресу. Однак, IP-адреса — це дефіцитний ресурс, тому виділяти великий діапазон адрес доволі проблематично. Для вирішення цієї проблеми використовують метод трансляції мережної адреси *NAT* (Network Address Translation). Сервіси NAT доступні вузлу або програмі, яку називають *NAT-проксі* (може бути суміщений з іншими типами проксі-серверів). Всередині корпорації кожному вузлу виділяється довільна IP-адреса із діапазону так званих приватних або локальних IP-адрес. Наведемо три діапазони зарезервованих IP-адрес:

10.0.0.0 — 10.255.255.255;
 172.16.0.0 — 172.31.255.255;
 192.168.0.0 — 192.168.255.255.

Коли пакет з вузла-відправника надходить в Інтернет через NAT-проксі, останній замість його IP-адреси вставляє єдину IP-адресу для всієї корпорації. Таким чином, коли вузол-отримувач в Інтернеті відправить відповідь, його пакет надійде саме до NAT-проксі. Тобто всі вузли в межах NAT-проксі з точки зору Інтернет будуть мати одну адресу — адресу NAT-проксі. Для того, щоб той зміг перенаправити пакети потрібному вузлу в корпорації, він веде таблицю, в якій зберігає допоміжну інформацію про відправлені пакети. Дана методика базується на використанні портів TCP та UDP, які будуть описані нижче.

Наведемо спрощену схему функціонування NAT.

Наприклад, вузол відправник (вузол1) у межах корпорації має IP 192.168.2.4. Прикладна програма Internet Explorer надсилає пакет на WEB сервер вузла в Інтернеті (вузол 2) з IP-адресою 194.24.24.24. При цьому Internet Explorer на рівні TCP формує клієнтський порт TSAP = 2265, звертається відповідно до 80 порту (WEB сервер). Пакет буде з параметрами:

IP_{відправника} = 192.168.2.4, IP_{отримувача} = 192.24.24.24, TSAP_{відправника} = 2265,
 TSAP_{отримувача} = 80.

NAT-блок (IP=194.24.1.1) створює запис із номером 4567 (вибраний довільно), в який записує:

Запис=4567, IP_{відправника} = 192.168.1.1, TSAP_{відправника} = 2265.

Після цього в пакеті замість номера порту відправника записується номер запису в таблиці TSAP=4567, а IP відправника змінюється на IP NAT-проксі IP=194.24.1.1. Таким чином, в Інтернет відправиться пакет з параметрами:

IP_{відправника} = 192.24.1.1, IP_{отримувача} = 192.24.24.24, TSAP_{відправника} = 4567,
 TSAP_{отримувача} = 80.

Вузол2 (IP = 192.24.24.24) сформує відповідь у вигляді:

$IP_{\text{відправника}} = 192.24.24.24$, $IP_{\text{отримувача}} = 192.24.1.1$, $TSAP_{\text{відправника}} = 80$,
 $TSAP_{\text{отримувача}} = 4567$.

Отримавши такий пакет, NAT-блок знаходить у своїй таблиці запис з номером, який дорівнює $TSAP_{\text{отримувача}} = 4567$, вилучає всі поля і формує запит до вузла 1:
 $IP_{\text{відправника}} = 192.24.24.24$, $IP_{\text{отримувача}} = 192.168.2.4$, $TSAP_{\text{відправника}} = 80$,
 $TSAP_{\text{отримувача}} = 2265$.

Для вузла1 і вузла2 ця операція проходить прозоро. Однак, такий метод економії ресурсів IP адрес має свої недоліки, зокрема:

- така схема працює тільки для протоколів TCP та UDP, для інших необхідно пропонувати інші рішення;
- деякі прикладні програми вставляють свої IP в поле даних, що приводить до їх неправильної роботи (FTP, H.323 — IP-телефонія).

10.3. Протоколи транспортного рівня

10.3.1. Протоколи UDP та TCP

10.3.1.1. UDP. Протокол *UDP* (User Datagram Protocol — дейтаграмний протокол користувача) — один із поширених транспортних протоколів, який використовується в Інтернеті. Він дозволяє відправляти інкапсульовані IP-дейтаграми без встановлення з'єднання. В заголовку для ідентифікації кінцевих точок (програм) вказується порт джерела та порт призначення на мережному вузлі відправника та отримувача. Принцип роботи портів висвітлений у другому розділі, тут тільки коротко нагадаємо, що коли прибуває пакет UDP, інкапсульовані корисні дані передаються тій прикладній програмі, з якою пов'язаний порт призначення. Якщо на цю посылку треба відправити відповідь, то програма копіює поле порту джерела в поле порт призначення і таким чином направляє відповідь до програми клієнта.

Отже, єдина функція UDP — ідентифікація програм (процесів) відправника та отримувача, тобто адресація на транспортному рівні. Контролем потоку, помилок, повторною передачею після прийому пошкодженого сегмента даний протокол не займається.

10.3.1.2. TCP. Протокол *TCP* (Transmission Control Protocol — контроль управління передачею) спеціально розроблений для забезпечення надійного потоку даних за ненадійною мережею. Транспортна сутність на мережному вузлі відправника розбиває корисні дані на декілька частин і відправляє окремими IP-дейтаграмами. На вузлі отримувача TCP сутність забезпечує «складання» цих частин. Оскільки IP-рівень не гарантує збереження послідовності доставки даних та взагалі їх доставку, то цією процедурою займається TCP. За відсутності деяких дейтаграм TCP забезпечує їх повторну відправку і таким чином гарантує надійність передачі. Сутність TCP орієнтована на з'єднання. Тобто спочатку необхідно провести з'єднання двох транспортних сутностей (TCP клієнта і TCP-серверу), через яке два процеси зможуть передавати потоки байтів. Адресація процесів на вузлах проходить аналогічно, як в UDP. Для розуміння роботи TCP розглянемо функціонування транспортних сутностей у контексті моделі примітивів-сокетів.

10.3.2. Модель сокетів

10.3.2.1. TCP у контексті моделі сокетів. У табл. 10.2 наведені основні примітиви роботи *сокетів*. Сокети можна уявляти як віртуальне гніздо зв'язку до якого підключається віртуальний канал. Дві прикладні програми створюють сокети за допомогою примітивів SOCKET і «з'єднують» їх між собою. Той, хто очікує з'єднання, є Сервером, а той, хто активно з'єднується, називається Клієнтом. При з'єднанні сокетів створюються канали.

Тепер детальніше. Сервер виконує примітив BIND для того, щоб зв'язати даний сокет з IP-адресою та потрібним номером TCP-порту. Після цього він виконує процедуру LISTEN, якою вказує системі, що очікує з'єднання за цим сокетом.

Для того щоб підключитися до сокета Серверу, Клієнту необхідно скористатися примітивом CONNECT, вказавши IP адресу та порт отримувача. Коли запит на з'єднання надходить на Сервер, той підтверджує з'єднання примітивом ACCCEPT. Слід зазначити, що Клієнт не повинен прив'язуватись до конкретної локальної адреси порту, тому що клієнтські адреси формуються динамічно.

Таблиця 10.2

ОСНОВНІ ПРИМІТИВИ СОКЕТІВ

Примітив	Призначення
SOCKET (СОКЕТ)	Створити новий сокет (гніздо зв'язку)
BIND (ЗВ'ЯЗАТИ)	Зв'язати локальну адресу із сокетом
LISTEN (ОЧІКУВАТИ)	Очікувати з'єднання з боку клієнта
ACCCEPT (ПРИЙНЯТИ)	Підтвердити з'єднання
CONNECT (З'ЄДНАТИ)	Активно пробувати з'єднатись із сервером
SEND (ВІДПРАВИТИ)	Відправити дані за встановленим каналом
RECEIVE (ОТРИМАТИ)	Отримати дані із з'єданого каналу
CLOSE (ЗАКРИТИ)	Розірвати з'єднання

Після установки з'єднання Клієнт і Сервер можуть обмінюватись даними за дуплексним TCP-каналом за допомогою примітивів SEND та RECEIVE.

Для роз'єднання каналу викликається примітив CLOSE.

Усі TCP-з'єднання є повнодуплексними та двохточковими, тобто трафік може йти в обидва боки одночасно між двома точками. Широкомовна та багатоадресна пересилка не підтримуються. Кожний сокет має адресу, яка складається з IP-адреси та 16-бітного номера порту.

Порти використовуються як у TCP, так і в UDP-протоколах. Номери портів із значеннями нижче 1024, які називаються популярними портами, зарезервовані стандартними сервісами. Наприклад, будь-який процес може з'єднатися з портом 21 необхідного вузла для того, щоб доступитися до його FTP-серверу. Тобто порт 21 зарезервований для FTP-серверів. Список таких портів наведений на сайті www.iana.org. У табл. 10.3 представлені деякі з них.

Таблиця 10.3

ПРИЗНАЧЕННЯ ПОРТІВ TCP ТА UDP-ПРОТОКОЛІВ

Порт	Протокол	Використання
21	FTP	Пересилка файлів
23	Telnet	Дистанційний вхід у систему
25	SMTP	Електронна пошта
69	TFTP	Простий протокол передачі файлів
79	Finger	Пошук інформації про користувача
80	HTTP	WEB-сервер
110	POP-3	Віддалений доступ до електронної пошти
502	Modbus TCP	Modbus TCP-сервер

10.3.2.2. Використання Winsock з TCP/IP. Припустимо, що є дві SCADA-програми, в яких є вбудована мова VBA. Є необхідність у передачі даних між вузлами з використанням TCP або UDP поверх протоколу IP. Для цього можна скористатися ActiveX-елементом *Winsock*, який, як правило, вже присутній на комп'ютерах з операційною системою Windows. Цей компонент є інтерфейсом до однойменного сервісу в операційній системі Windows і функціонує приблизно так, як на наведеній вище моделі сокетів. Однак, він може працювати як у режимі TCP, так і UDP. Для останнього — з'єднання проводити не потрібно. Достатньо розмістити Winsock-елемент на контейнері (наприклад, формі) і можна користуватися його методами, властивостями та подіями.

Приклад 10.2. Промисловий Ethernet. Обмін даними за допомогою компонентів Winsock.

Завдання. Передати одну змінну типу Integer між прикладними програмами VBA за допомогою компонентів Winsock.

Рішення. Елемент Winsock на сервері має назву tcpServer, а на клієнтському комп'ютері — tcpClient. На боці сервера прописуємо наступний код:

```
Private RegInput As Integer, RegOutput As Integer
Private Sub Server_Listen() 'очікувати з'єднання
    tcpServer.LocalPort = 1001 'Задається номер порту для з'єднання
    tcpServer.Listen 'чекаємо на з'єднання
End Sub
Private Sub tcpServer_ConnectionRequest (ByVal requestID As Long)
    ' якщо попереднє з'єднання не закінчилось закрити його примусово
    If tcpServer.State <> sckClosed Then tcpServer.Close
    ' Прийняти запит з параметром requestID на установку з'єднання
    tcpServer.Accept requestID
End Sub
Private Sub Send_Reg()
    tcpServer.SendData RegOutput 'Відправити значення RegOutput
End Sub
Private Sub tcpServer_DataArrival (ByVal bytesTotal As Long)
tcpServer.GetData RegInput 'прийняти з буфера прийому RegInput
End Sub
```

Програма на боці клієнта:

```

Private RegInput As Integer, RegOutput As Integer
Private Sub Client_Connect ()
tcpClient.RemoteHost = 192.168.1.1
tcpClient.RemotePort = 1001
tcpClient.Connect
End Sub
Private Sub Send_Reg ()
tcpClient.SendData RegOutput
End Sub
Private Sub tcpClient_DataArrival ByVal bytesTotal As Long)
tcpClient.GetData RegInput
End Sub

```

Приклад 10.2

10.4. Промисловий Ethernet

10.4.1. Кабельні системи промислового виконання

10.4.1.1. Необхідність промислового виконання. Використання Ethernet у промисловій автоматизації зробило необхідним прокладку кабелів та розміщення мережного обладнання (концентратори, комутатори, мости, тощо) в промислових умовах експлуатації. Різні організації пропонували свої рішення в цьому напрямку під загальною назвою *Industrial Ethernet*. Ці рішення базуються на використанні особливих матеріалів та конструкції мережних елементів, а також кабельної структури.

10.4.1.2. З'єднувачі промислового виконання. Серед найбільш вразливих місць кабельної структури звичайного Ethernet у промислових умовах експлуатації є використання з'єднання на базі RJ-45, що не підходить для промислового виконання, оскільки не задовольняє вимоги до умов експлуатації. Деякими організаціями в якості варіанта для кабельних систем промислового Ethernet пропонується використання модифікованого RJ-45, де з'єднання захищається герметичними ізолюючим корпусом (рис. 10.9). Ще одним рішенням є заміна RJ-45 іншими типами з'єднувачів. В промисловості найкраще себе зарекомендували з'єднувачі типу M-12, які мають ступінь захисту IP-67, а також стійкі до вібрацій та можуть працювати в широких температурних діапазонах. Для промислового Ethernet використовується 4-піновий варіант з'єднувача M-12 з D-кодуванням (рис. 10.10). Слід зазначити, що використовується тільки 2 вити пари замість 4-х.

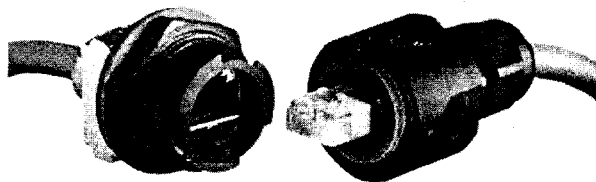


Рис. 10.9. Модифіковані RJ-45 з'єднувачі

10.4.1.3. Стандарти побудови СКС промислового виконання. Для інтеграції систем управління виробництвом у загальну систему управління підприємством необхідно, щоб кабельні системи промислового Ethernet входили у структуровані кабельні системи (СКС) підприємства. Загальні підходи до побудови СКС промислових будівель залишаються такими ж, однак відмінність умов експлуатації та специфіки побудови промислових систем автоматизації привела до появи стандарту ISO/IEC 24702. Стандарт розроблений з урахуванням таких особливостей:

- великі розміри території обслуговування поряд з дуже низькою (за офісними мірками) густиною розміщення розеток для підключення кінцевого активного обладнання;
- складна електромагнітна обстановка;
- жорсткі умови навколишнього середовища (кліматичні показники, механічні і хімічні впливи);
- немає технічної необхідності у масовому застосуванні мережних пристроїв зі швидкостями передачі інформації вище 100 Мбіт/с;
- потреба в підключенні до інформаційної системи контролерів, комутаторів, перетворювачів частоти та інших активних пристроїв, розміщених на різних рухомих механізмах;
- висока ймовірність застосування активного обладнання для побудови різних промислових мереж.

Таким чином, стандарти ISO/IEC 11801 та ISO/IEC 24702 описують можливі структури та мінімальну конфігурацію для прокладки СКС, вимоги до реалізації, продуктивності кабельних ліній, відповідності та процедур верифікації. Вимоги до планування кабельних ліній та їх монтаж визначаються стандартом ISO/IEC TR 14763-2 (Створення та експлуатація кабельних систем приміщень замовників. Частина 2. Планування та монтаж).

Згідно з ISO/IEC 11801 та ISO/IEC 24702, для СКС промислового виконання дозволяється використовувати конектори IEC 60603-7-х, IEC 61076-3-106 (варіанти 1 та 6), 61076-3-117 (варіант 14) та IEC 61076-2-101 (4-х-піновий M-12 з D-кодуванням). В якості середовища передачі замість звичайних UTP-кабелів рекомендують використовувати екрановані.

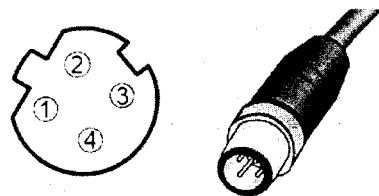


Рис. 10.10. 4-піновий M-12 з'єднувач D-кодуванням

10.4.2. Real Time Ethernet

10.4.2.1. Умови для роботи Ethernet на рівні промислових мереж. Для функціонування Ethernet як промислової мережі недостатнє тільки промислове виконання мережних компонентів. До промислового Ethernet рівнів датчиків та контролерів висуваються такі самі вимоги та функції, як для інших промислових мереж (див. розділ 2), а саме:

Функції:

1. Обмін даними процесу в реальному часі.
2. Програмування та конфігурація вузлів.
3. Діагностика вузлів.
4. Управління станом вузла.
5. Функції резервного переключення між вузлами мережі.

Вимоги:

1. Властивість детермінованості.
2. Завадостійкість та промислові умови експлуатації.
3. Надійність та живучість.
4. Простота, зручність інсталяції та обслуговування.
5. Можливість подачі живлення за кабельною системою мережі.
6. Вільна топологія.

10.4.2.2. Забезпечення реального часу — рішення RTE в стандартах МЕК.

Серед найбільш суттєвих недоліків класичного Ethernet є використання методу CSMA/CD, який виключає можливість повної детермінованості. З появою комутованого Ethernet методом CSMA/CD вже не послугуються для мереж, інфраструктура яких включає комутатори. Однак звичайні комутатори не забезпечують виділення реал-тайм-повідомлень як високопріоритетних і не можуть гарантувати своєчасну доставку даних. Крім того, при великому обсязі даних для передачі вхідні буфери комутаторів можуть переповнитися, що може привести до втрати даних, що неприпустимо.

Сьогоднішні є ряд рішень використання Ethernet в якості промислової мережі, в яких наведені вище проблеми та ряд інших розв'язують різними шляхами. Ці рішення доповнилися і модифікувалися організацією МЕК та увійшли до стандарту IEC 61784-2 під загальною назвою Ethernet реального часу — *Real Time Ethernet (RTE)*. При розробці стандарту враховували, щоб нові варіанти, крім вимог до реального часу, одночасно підтримували найбільш популярні протоколи, такі, як TCP/IP, HTTP та FTP для легкої інтеграції польових засобів в єдину систему управління підприємством. Іншими словами, мережі RTE, які визначені IEC 61784-2, повністю сумісні з існуючими системами на базі Ethernet TCP/IP і можуть функціонувати на одній і тій самій кабельній структурі.

10.4.2.3. Показники робочих характеристик RTE. Кожне з RTE-рішень, що увійшло в IEC 61784-2, має переваги перед іншими в тій чи іншій галузі використання. Для можливості вибору кінцевим користувачем тієї чи іншої технології кожна з них характеризується фіксованим набором показників робочих характеристик (Performance Indicators RTE). Цими показниками характеризуються як можливості самої мережі, так і вузлів та прикладних Процесів на RTE. Перші визначені стандартом, а другі — виробниками обладнання. Слід зазначити, що більшість із показників робочих характеристик є взаємозалежними. Перерахуємо основні характеристики.

1. Час доставки (Delivery time) — максимальний час, який необхідний для доставки даних від одного прикладного Процесу до іншого, включаючи всі затримки в мережі. Вказуються дві величини: максимальний час доставки за відсутності помилок та при одній помилці з повторенням передачі.

2. Кількість RTE-вузлів (Number of RTE end-stations) — максимальна кількість вузлів на мережі.

3. Базова мережна топологія (Basic network topology) — одна або комбінація із топологій: ієрархічна зірка, кільце, лінійна.

4. Кількість комутаторів між кінцевими станціями (Number of switches between RTE end-stations) — максимально можлива кількість комутаторів між кінцевими станціями.

5. Пропускна здатність для RTE (Throughput RTE) — загальна сума APDU даних в октетах/секунду (байт/секунду), що може бути передана на один зв'язок.

6. Ширина не-RTE смуги (Non-RTE bandwidth) — ширина смуги пропускання, яка може бути виділена під не-RTE обмін (для обміну типами сервісів, які не належать до даного рішення RTE, наприклад, WEB, FTP тощо.) Додатково визначається загальна ширина смуги пропускання. Показники Throughput RTE та Non-RTE bandwidth взаємозалежні.

7. Точність часової синхронізації (Time synchronization accuracy) — максимальне відхилення між годинниками двох будь-яких вузлів.

8. Точність не часово-базової синхронізації (Non-time-based synchronization accuracy) — максимальний час тимчасового відхилення в мережі періодичного синхроімпульсу, на якому базуються періодичні операції.

9. Час резервного відновлення (Redundancy recovery time) — максимальний час, який потрібний для повного відновлення операційного режиму після одиночного збою системи.

10.4.2.4. Класи RTE мереж. Для полегшення вибору RTE-системи всі рішення залежно від вимог до реального часу умовно поділяються на три класи:

1. Низький клас швидкості для систем людино-машинного інтерфейсу. Характеризується порівняно низькою періодичністю обміну даними (порядку 100 мс), що задовольняє вимогами процесів відображення та супервізорного управління. Така швидкість може бути досягнута, якщо використовувати стандартні протоколи TCP/IP.

2. Середній клас швидкості для систем управління технологічним процесом (рівень ПЛК). Детермінований час обміну порядку 10 мс. Таких вимог можна дотриматись шляхом використання спеціальних протоколів та високошвидкісного надійного обладнання.

3. Високий клас швидкості для управління рухом (системи позиціонування): для синхронізації декількох осей за мережею з часовим циклом менше 1 мс необхідно, щоб джиттер не перевищував 1 мкс. Це може бути досягнуто тільки на мережах з бітовою швидкістю >100 Мбіт/с, модифікацією мережного обладнання та протоколом доступу до середовища.

На рис. 10.11 показані структури стеків протоколів для різних рішень. Загальне для всіх — використання спільної кабельної інфраструктури. Для побудови реал-тайм рішень використовуються три різні підходи. У першому підході (над TCP/IP) послугуються прикладним протоколом реального часу, який використовує протоколи TCP/UDP/IP. Другий підхід (над Ethernet) передбачає наявність поряд з протоколами TCP/IP протоколів реального часу, які доступуються безпосередньо до Ethernet. Для третього підходу (модифікація Ethernet) необхідно за-

мінити комунікаційне обладнання Ethernet для забезпечення взаємодії деяких вузлів у мережі між собою в реальному часі. Слід нагадати, що всі підходи передбачають нормальне функціонування всіх інших (не реал-тайм) вузлів системи та їх протоколів на тій самій кабельній інфраструктурі.

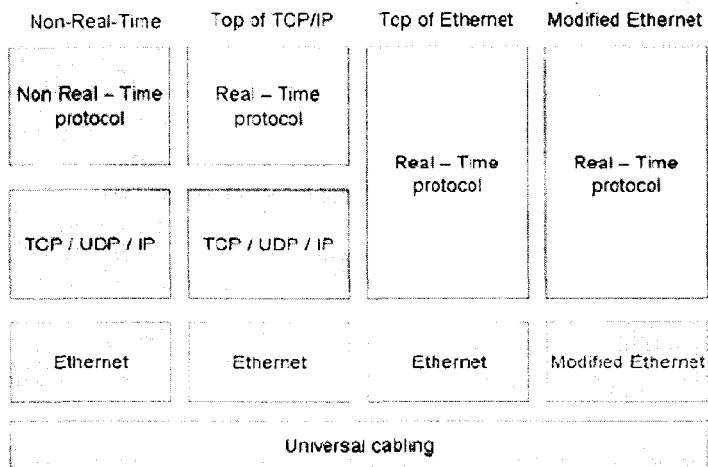


Рис. 10.11. Стеки протоколів різних рішень Industrial Ethernet

10.4.3. Рішення RTE над TCP/IP

10.4.3.1. MODBUS RTPS. Одне з реал-таймів рішень поверх TCP/IP — це вдосконалений MODBUS TCP/IP. Протокол прикладного рівня MODBUS, а також його реалізація з використанням TCP/IP описана в розділі 6. Його можна віднести до найнижчого за швидкістю класу, який, як правило, задовольняє вимоги функціональності для ЛІМІ, а також низькошвидкісний клас завдань обміну між ПЛК та польовими засобами рівня датчиків. У стандартах IEC 61784-2 та IEC 61158 додатково до даного протоколу добавлені нові реал-тайми розширення, які описані в протоколі Real-Time Publisher Subscriber (**RTPS**). RTPS-протокол забезпечує дві основні комунікаційні моделі: модель Видавець-Абонент обміну ідентифікованими даними; Composite State Transfer (CST) protocol — передача інформації про стан змінної від записувача до зчитувача.

Нагадаємо, що передача даних процесу від Видавця до одного або декількох Абонентів називається публікацією. Властивості публікованих об'єктів описують зміст, тип і якість (наприклад, часовий інтервал) виданого потоку в мережу. Абонент визначає мінімальний час між послідовністю видань. Це впливає на максимальну швидкість, при якій Абонент готовий отримувати видання. Протокол RTPS розроблений для функціонування над ненадійним транспортним протоколом UDP, для якого корисне повідомлення вміщується в межах однієї дейтаграми UDP/IP. На відміну від стандартного протоколу MODBUS, протокол RTPS не знайшов широкого застосування. Навіть серед засобів Schneider Electric, для яких

MODBUS є рідним протоколом, RTPS розширення використовується в TSX Premium, Quantum та M340 для обміну між ПЛК (Сервіс Global Data). Слід зазначити, що розширення RTPS дають можливість виконання роботи середнього класу швидкості.

10.4.3.2. ETHERNET/IP. EtherNet/IP, який визначений Rockwell і підтримується Open DeviceNet Vendor Association (ODVA — www.odva.org) та ControlNet International (www.controlnet.org), використовує Common Interface Protocol (CIP), який є загальним для мереж EtherNet/IP, ControlNet, DeviceNet та CompoNET. Комунікаційна технологія EtherNet/IP забезпечує реал-тайм з'єднання на основі ISO/IEC 8802-3 та IEEE 802.1Q. У дуплексному Ethernet (на базі комутаторів) немає затримок через колізії. Однак, у пристрої комутації різні Ethernet кадри можуть бути затримані, якщо вихідний порт уже зайнятий для передачі іншого кадру. Це може привести до недетермінованих затримок, які неприпустимі в реал-таймі Ethernet.

Для розділення реал-тайму від не реал-тайму з'єднань у EtherNet/IP застосовується концепція віртуальних мереж VLAN, яка визначена стандартом IEEE 802.1Q. Для реал-таймів кадрів у полі Priority (рис. 10.6) виставляється найвищий пріоритет, що дозволяє керованим комутаторам передати їх раніше реал-тайм повідомлень. Детальніше з EtherNet/IP та іншими CIP мережами можна ознайомитись у розділі 2.

10.4.3.3. P-NET on IP. Ця специфікація запропонована датським національним комітетом та призначена для використання в IP-орієнтованих мережах. Вона дозволяє послуговуватися реал-таймзв'язком P-NET, інкапсульованим у UDP/IP-пакети. Ідея функціонування даної мережі полягає в прозорості маршрутизації пакетів у будь-якому напрямку як через IP мережі, так і через інші типи P-NET мереж. Кожний P-NET кадр має два P-NET-route елементи, побудовані як таблиця із адреси відправника та отримувача. Загалом для мережі — це адреси вузлів. Для передачі по IP ці таблиці розширені для включення IP-адреси. Для вузлів P-NET IP-адреси — звичайні адреси, тільки в іншому форматі. Це означає, що будь-який P-NET клієнт може досягнути до серверу на IP мережі, не інтерпретуючи формат IP-адреси. Функціонування самої мережі Ethernet залишається без змін, оскільки P-NET взаємодіє через протокол UDP.

10.4.3.4. VNET/IP. Vnet/IP розроблений Yokogawa для систем DCS середніх та великих масштабів з можливістю передачі великих обсягів даних з періодичністю порядку 100 мс за умови використання гігабітного Ethernet. Дане рішення дозволяє одночасно функціонувати на Ethernet реал-таймі та нереал-таймі трафіка за рахунок застосування 3-х прийомів: планування передачі, управління пріоритетами та високошвидкісної відповіді.

Планування передачі запобігає одночасній відправці даних декількома станціям одночасно, що привело б до переповнення буферів комутаторів. Для цього весь час ділиться на макроцикли тривалістю 100 мс, у межах яких всі станції гарантовано відправляють свої дані з дискретністю виділеного для них тайм-слота (1 мс). Для управління пріоритетністю повідомлень даних процесу перед нереал-тайм даними в IP-пакеті використовується поле ToS з 4-ма рівнями привілеїв (розглянуте вище), яке підтримується буферами станцій на мережі, а також може

бути підтримане комутаторами. Для передачі даних процесу використовується стек протоколів UDP/IP, що дозволяє значно зменшити час передачі порівняно з TCP. Обмін даними процесу функціонують на базі моделі Видавець-Абонент.

10.4.4. Рішення RTE над Ethernet

10.4.4.1. Загальні концепції функціонування RTE поверх Ethernet. Ідея використання реалізацій даного типу полягає у безпосередньому використанні Ethernet та визначенні в кадрі Ethernet спеціального типу протоколу (поле EtherType). Нагадаємо, що стандартний тип протоколу для IP — це Ethertype = 800_{16} . Дані RTE-протоколи поряд зі стеком протоколів TCP/IP використовують свої стеки, ідентифіковані з їх власним типом протоколу. Таблиця 10.3 вказує перелік значень Ethertype для різних протоколів.

Таблиця 10.3

ЗНАЧЕННЯ ETHERTYPE ДЛЯ РІЗНИХ ПРОТОКОЛІВ

Профіль IEC 61784	Ім'я бранда	Ethertype
CPF-2	CIP (Ethernet/IP)	800_{16} (IP)
CPF-3	PROFIBUS & PROFINET	8892_{16}
CPF-4	P-Net (P-Net on IP)	800_{16} (IP)
CPF-10	Vnet/IP	800_{16} (IP)
CPF-11	TCnet	$888B_{16}$
CPF-12	EtherCAT	$88A4_{16}$
CPF-13	Ethernet POWERLINK (EPL)	$88AB_{16}$
CPF-14	EPA	$88BC_{16}$
CPF-15	MODBUS-RTPS	800_{16} (IP)
CPF-16	SERCOS (SERCOS III)	$88CD_{16}$

10.4.4.2. ETHERNET POWERLINK. Ethernet Powerlink (EPL) був визначений Bernecker+Rainer (B&R), і тепер підтримується Ethernet Powerlink Standardisation Group (EPSG — www.ethernetpowerlink.org).

Мережа базується на використанні системи планування на загальному сегменті Ethernet, яка називається Slot Communication Network Management (SCNM). Одному з вузлів мережі дають права Керуючого Вузла — Managing Node (MN), який слідкує за розподілом часу між вузлами. Він гарантує доступ у режимі реал-тайм до циклічних даних і резервує часові слоти для нереал-таймів кадрів (TCP/IP). Всім іншим вузлам, які називаються Керовані Вузли — Controlled Nodes (CN), відправляти і отримувати запити дозволяється тільки з дозволу Керуючого.

Загальний час обміну складається з ізохронних комунікаційних циклів, які поділені на періоди. На початку циклу Керуючий Вузол у ширококомовному

режимі відправляє групові Start-of-Cycle (SoC) кадри (рис. 10.12). Час відправки і отримання цього кадру є базовим для загальної синхронізації всіх вузлів. Далі, у межах цього часу циклу, функціонують періоди: Стартовий, Ізохронний (Isochronous — через однакові інтервали часу), Асинхронний (Asynchronous — не прив'язаний до синхросигналу) та додатковий період простою (Idle). Довжина індивідуальних періодів може змінюватися в межах заданого періоду циклу EPL.

В Ізохронний період циклу, в режимі точка-точка, кожному сконфігурованому і активному вузлу відправляється спеціальний кадр-запит Poll-Request (PReq). Вузол, до якого звертаються, відповідає широкомовним кадром — відповіддю Poll-Response (PRes). За рахунок чітко визначених інтервалів часу запиту в мережі забезпечується ізохронний режим відновлення необхідних даних.

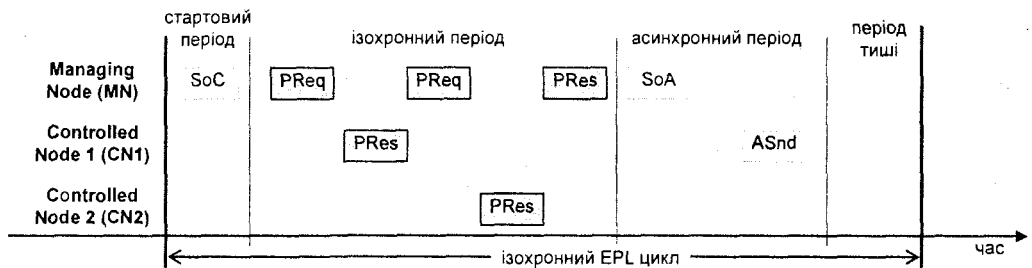


Рис. 10.12. Організація циклу EPL

В Асинхронний період циклу, доступ до EPL-сегмента мережі може бути наданий одному CN або MN для передачі єдиного асинхронного повідомлення. Привілейованими для асинхронних повідомлень є протоколи UDP/IP. Кадр типу Start-of-Asynchronous (SoA) — перший кадр в асинхронний період. Цей кадр є сигналом для всіх CNs про-те, що всі ізохронні дані були передані.

Так побудована організація обміну на базі мережі Ethernet виключає можливість одночасної передачі ізохронних та асинхронних повідомлень, що регламентує повну детермінованість. Для правильного функціонування Ethernet Powerlink необхідно виділити окремий «захищений» сегмент, в якому будуть присутні тільки EPL вузли. Інакше сторонній вузол зруйнує SCNM-механізм доступу. Для зв'язку з «незахищеними» сегментами в Керуючі Вузли включають функції маршрутизації.

Прикладний рівень EPL базується на стандартах CANopen. Інтеграція EPL з CANopen дозволяє використовувати профілі, забезпечити високошвидкісний обмін даними та відкритий і прозорий зв'язок з TCP/UDP/IP-протоколами. Профілі пристроїв визначають об'єкти PDO та SDO, які передаються за ізохронним EPL зв'язку, а SDO — за UDP/IP-протоколом. Таким чином, на EPL мережі використовуються базові CANopen профілі пристроїв, номенклатура яких сьогодні дуже велика.

10.4.4.3. TCnet. TCnet (Time-critical Control Network) — пропозиція від Toshiba. Подібно Ethernet Powerlink, мережний час, який називається High-speed-

Transmission період, складається з реал-тайму (Timecritical) циклічного сервісу/обслуговування даних і спонтанного (Sporadic) сервісу відправки повідомлення. Реал-тайм циклічний сервіс функціонує на одному сегменті мережі і може бути описаний моделлю обміну ідентифікованими даними Видавець-Абонент.

На початку періоду High-speed-Transmission усім RTE-TCnet вузлам у широкомовному режимі відправляється спеціальне синхронізуюче повідомлення SYN. Після отримання SYN-кадру вузол з номером 1 починає відправляти кадр даних, який запланований у процесі конфігурування системи. Після завершення передачі цих кадрів вузол 1 у широкомовному режимі відправляє кадри типу Completed Message (дивись CMP1 на рис 10.13). Після цього свої кадри передає вузол 2 і т. д. Кожний вузол може тримати право передачі протягом заданого часу і повинен віддати це право наступному вузлу в межах цього часу. Таким чином, у межах періоду High-speed-Transmission відбувається маркерний метод доступу до шини Ethernet.

Вузол, що отримав маркер, може відіслати як циклічні, так ациклічні повідомлення. Циклічна передача даних розділена на High, Medium та Low-Speed циклічну передачу. При отриманні права передачі кожний вузол встигає принаймні передати Highspeed циклічні дані. Якщо термін дозволеного часу не вичерпався, відправляються Medium та Low-Speed-повідомлення. Таким чином, час відновлення для High-speed повідомлень рівний часу SYN, а Medium-speed або Low-speed-повідомлення передаються через декілька SYN-кадрів.

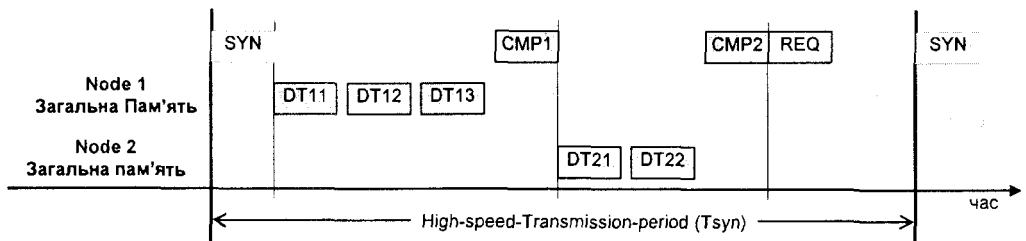


Рис. 10.13. Організація циклу Tsyn

Сервіси прикладного рівня RTE-TCnet базуються на моделі Видавець-Абонент із загальною пам'яттю (Common Memory). Загальна мережна пам'яття є віртуальною загальнодоступною пам'яттю, розподіленою між прикладними Процесами вузлів TCnet окремими блоками різної величини. Для кожного блока даних один з вузлів є Видавцем, який публікує їх у період циклічного обміну. Кожен вузол отримує блок даних як Абонент і відновлює свою локальну копію загальної пам'яті. Це значить, що кожний контролер може швидко отримати доступ до даних один одного, звертаючись до своєї локальної копії.

10.4.4.4. ЕРА. ЕРА-протокол (Ethernet for Plant Automation) є пропозицією китайських виробників і належить торговій марці ZHEJIANG SUPCON CO. LTD.

В ЕРА загальний мережний час складається з макроциклів, які позначені літерою Т (рис. 10.14). Кожний макроцикл розділений на дві частини: фаза передачі періодичного повідомлення (Tp) і фаза передачі неперіодичних повідомлень

(T_n). Остання частина періодичного повідомлення кожного пристрою може вміщувати заявку на передачу аперіодичного повідомлення, яка реалізується на фазі передачі аперіодичних повідомлень.

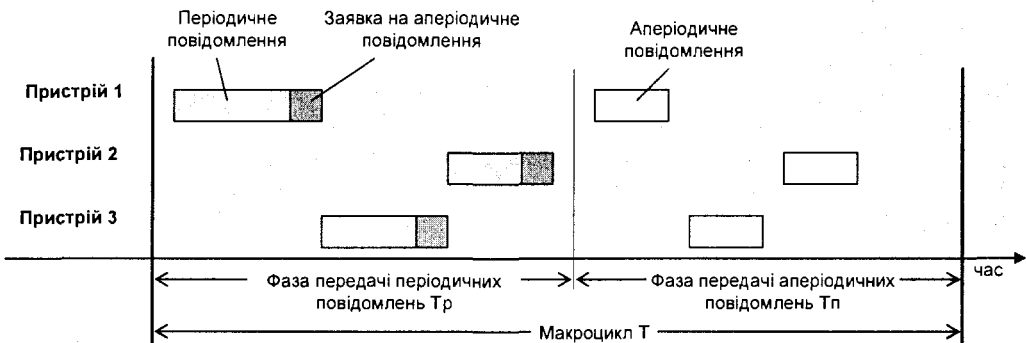


Рис. 10.14. Організація макроциклу EPA

В EPA-системах є два види прикладних Процесів, які можуть виконуватися паралельно в одній системі: реал-тайм EPA прикладні функціональні блоки і нереал-тайми TCP/IP прикладні Процеси. Обмін даними між функціональними блоками проводиться шляхом поєднання їх вхідних/вихідних параметрів, використовуючи EPA прикладні сервіси.

10.4.4.5. PROFINET CBA. PROFINET визначений декількома виробниками (включаючи Siemens) та підтримується міжнародною організацією PNO. Сьогодні визначені два типи RTE рішень для PROFINET: PROFINET CBA для мереж рівня контролерів та PROFINET IO для мереж рівня датчиків. Тут розглянемо функціонування CBA.

Для PROFINET CBA доступні два типи сервісів: передача нереал-тайм (NRT) та реал-тайм (RT) повідомлень. NRT-повідомлення призначені для обміну параметричними та іншими даними прикладних Процесів, що базуються на TCP/IP. Час передачі NRT-повідомлень порядку 100 мс.

Для передачі реал-тайм даних процесу використовуються сервіси RT. RT-протокол базується на спеціальному типі Ethernet і використовує пріоритетні повідомлення VLAN аналогічно EtherNet/IP. При обміні даними в одній мережі APDU безпосередньо передаються кадром Ethernet без використання транспортного та мережного протоколів, що значно зменшує витрати на передачу (менше 10 мс). При обміні даними між вузлами різних мереж та при мультиадресній передачі додатково використовується стек UDP/IP.

10.4.5. Рішення RTE з модифікацією Ethernet

10.4.5.1. Загальні концепції функціонування RTE з модифікацією Ethernet. Типова кабельна топологія Ethernet — це зірка, за якої всі пристрої підключаються до центрального комутуючого пристрою. Для промислових мереж най-

більш популярними стала шинна топологія, що зменшує вартість кабельних ліній. Таким чином, Real-Time Ethernet рішення повинні врахувати шинну або кільцеву топологію зі зменшенням витрат прокладки кабелю.

У промислових мережах набуло популярності з'єднання типу daisy-chain — «ланцюжок», коли всі пристрої поєднуються в шину або кільце шляхом послідовного підключення мережних пристроїв (видима лінійна топологія або кільце). Щоб організувати таке з'єднання з комутованим (дуплексним) варіантом Ethernet, необхідно вбудувати комутатори в усі пристрої. Більшість рішень, які забезпечують жорсткі реал-таймсервіси базуються на певних модифікаціях в апаратних засобах мережної інфраструктури (комутатори, концентратори або мости). Модифікації обов'язкові для всіх пристроїв у межах сегмента RTE, однак дозволяють нереал-тайм кадрам передаватись без модифікацій.

10.4.5.2. SERCOS. Промислова мережа SERCOS (SEriell Real time Communication System Interface, www.sercos.org) описана в стандарті IEC 61491 і більш відома як інтерфейс CNC (Computer(ized) Numerical(ly) Control(led)). Пізніше цей стандарт був приєднаний до стандарту IEC 61158/IEC 61784 (SERCOS I, SERCOS II), а також до рішень на базі Ethernet, як SERCOS III.

У системі SERCOS III виділяється одна станція з правами Ведучого та до 254 станцій з правами Ведених. Кожен вузол має по два порти Ethernet, за допомогою яких організовується лінійна топологія або кільце. Жодний вузол у фізичній структурі, окрім останнього, не може бути зв'язаний із загальними комутаторами. Зв'язок між сусідніми станціями може проходити тільки безпосередньо. Останній вузол у ланцюжку може бути під'єднаний до комутатора для зв'язку з пристроями через TCP/IP або UDP/IP.

У SERCOS III комунікаційний обмін складається з логічних каналів зв'язку: канал реал-тайму (RT) та нереал-тайму (IP). Цикл зв'язку ініціюється Ведучим і вміщує до 4-х Master Data Telegrams (MDT) та 4-х Device Telegrams (AT acknowledge) як у каналі RT, так і в IP.

Телеграми MDT передаються Ведучим і отримуються кожним Веденим (рис. 10.15). Вони містять інформацію про синхронізацію, записи даних про інформацію управління, сервісні дані каналу та значення команди, які відсилаються кожному Веденому.

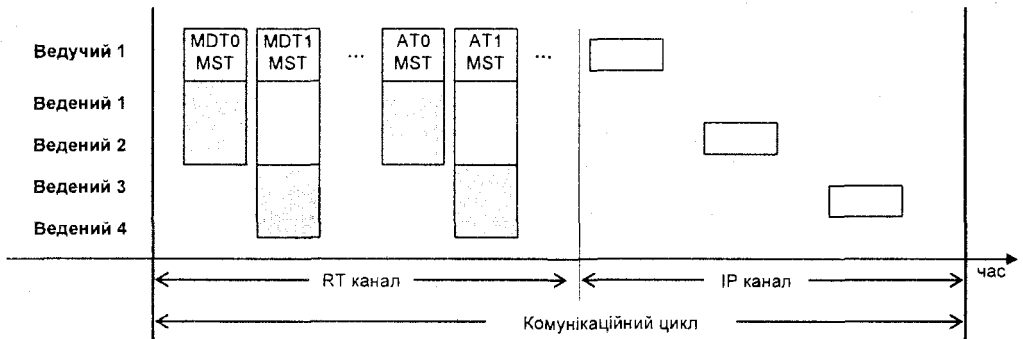


Рис. 10.15. Організація циклу в SERCOS

Телеграми АТ передаються Ведучим як порожній кадр з наперед визначеними полями, але без інформації. Кожний Ведений вставляє свої дані в асигновані для нього поля цього кадру. В межах своїх полів Ведені передають інформацію статусу, сервісні дані, фактичні значення каналу як Ведучому, так і іншим Веденим. Кількість та довжина телеграм даних MDT та АТ встановлені відповідно до конфігурації, яка також визначена протягом ініціалізації.

IP телеграми є стандартними нереал-тайм повідомленнями, які можуть використовуватися для будь-якої цілі і навіть можуть бути опущені. Тривалість IP каналу має встановлене значення і передбачає максимальну кількість IP телеграм, які можна відправити протягом цього часу.

Ця послідовність передачі RT та IP телеграм повторюються на кожному комунікаційному циклі. Тривалість кожного циклу вибирається з ряду: 31,25; 62,5; 125; 250; $n \cdot 250$ (до 65000) мкс. Часові інтервали для каналів RT, IP та часу передачі АТ передані протягом ініціалізації і тому відомі кожному Веденому. У кожному пристрої та спеціалізованому ПЗ на мережі необхідне розділення каналів RT від IP.

Система SERCOS складається з контролера та одного або декількох приводів (PDS), які циклічно обмінюються даними. Цей обмін включає передачу статусу та фактичні значення від приводу до контролера та передачу команди і заданого значення від контролера до приводу. Функціональні можливості PDS визначаються способом зміни різноманітних параметрів у моделі.

10.4.5.3. ETHERCAT. EtherCAT визначений Beckhoff і підтримується EtherCAT Technology Group (ETG see also www.ethercat.org). Дана технологія використовує Ethernet-кадри та відправляє їх за спеціальною кільцевою топологією.

Для управління доступом до середовища передачі використовується технологія Ведучий/Ведений, де Ведучий вузол (наприклад, контролер) відправляє кадри Ethernet Веденим вузлам, які вилучають дані з кадрів та вставляють свої дані в ці кадри.

В EtherCAT усі вузли об'єднуються в кільце (рис. 10.16). Ведучий зв'язаний з одним із відкритих кінців або безпосередньо з пристроєм, або через комутатор Ethernet. Тобто, кадри передаються від Ведучого до першого Веденого, потім до наступного і так далі, поки кадр не дійде до останнього Веденого. Той повертає кадр зворотним шляхом.

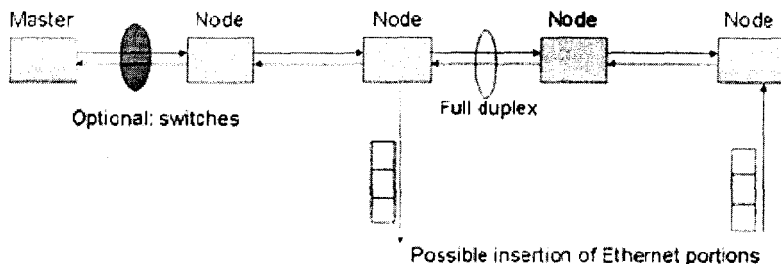


Рис. 10.16. Організація циклу ETHERCAT

3 точки зору Ethernet, сегмент EtherCAT — єдиний пристрій Ethernet, який приймає та відправляє стандартні ISO/IEC 8802-3 Ethernet-кадри. Однак, це може бути і багато Ведених пристроїв, які обробляють кадри, що надходять до них з боку вхідних каналів, вилучаючи призначені їм дані та вставляючи в кадр свої дані. Таким чином, останній пристрій у ланцюзі сегмента відправляє назад повністю оброблений кадр. Цей кадр, досягнувши першого Веденого, відправиться Ведучому як кадр відповіді.

Щоб досягати максимальної продуктивності, кадри Ethernet повинні бути оброблені «на льоту». Тобто вузол обробляє та ретранслює повідомлення наступному вузлу на лінії, як тільки повідомлення починає прийматися. Це значить, що вузли ретранслюють кадри ще до їх повного отримання.

Для реалізації таких вузлів необхідний спеціальний чіп ASIC для доступу до середовища, який інтегрує двопортовий комутатор у фактичний пристрій.

Вузли мають адресовану пам'ять, яка може бути доступна для послуг читання чи запису для кожного вузла послідовно або до декількох вузлів одночасно. Декілька телеграм EtherCAT можуть бути вкладені в межах одного кадру Ethernet. Кожна телеграма звертається до секції даних як до набору змінних пам'яті (наприклад входи чи виходи).

Телеграми EtherCAT транспортуються безпосередньо в ділянці даних кадру Ethernet або в межах секції блока даних протоколу UDP, який транспортується через IP. Перший варіант обмежений одною підмережею Ethernet, тому що зв'язані кадри не передаються маршрутизаторами. Для управління вузлами це, як правило, не є обмеженням. Декілька сегментів EtherCAT можуть бути зв'язані з одним або кількома комутаторами. Для передачі даних між іншими сегментами використовуються MAC-адреси першого Веденого в сегменті EtherCAT. Другий варіант призначений для менш критичних за часом застосувань, в якому кадри можуть бути маршрутизовані.

Якщо використовується тільки реал-таймтрафік, та час циклу може тривати 30 мкс. Максимальна одиниця передачі Maximum Transmission Unit (MTU) в Ethernet кадрі з 1514 байтами на бітовій швидкості 100Mbit/s в нереал-тайм часі міг би збільшити цикл до 200-250 мкс. Однак, в EtherCAT телеграми Ethernet поділені на частини і повторно збираються у вузлі призначення перш ніж ретранслюватися на пристрій, пов'язаний з вузлом. Ця процедура не обмежує досяжний час циклу, бо розмір фрагментів може бути оптимізований згідно з доступною шириною смуги. Таким чином, EtherCAT при передачі нереал-тайм повідомлення використовує свій спосіб фрагментації, який відмінний від фрагментації IP. Цей метод дозволяє будь-якому пристрою EtherCAT брати участь у нормальному трафіку Ethernet та все ще мати час циклу для RTE менший, ніж 100 мкс.

Подібно до EthernetPowerLink, EtherCAT на прикладному рівні використовує модель CANopen. PDO відображаються на вхідних та вихідних буферах пристроїв. Однак, для обміну SDO, EtherCAT використовує механізми поштового ящика (в EPL використовується IP).

10.4.5.4. PROFINET IO. Мережа PROFINET IO призначена для рівня датчиків та виконавчих механізмів. Обмін даними між пристроями може проходити в

різних класах сервісів зв'язку, таких, як Ізохронний реал-тайм (IRT), реал-тайм (RT) та нереал-тайм (NRT).

Зв'язок по Ethernet поділений на часові цикли відправки (send clock cycles), кожний з яких має різні часові фази, як це представлено на рис. 10.17. У першій часовій фазі, яка називається Ізохронна Фаза, передаються всі IRT-кадри. Ці кадри проходять через спеціалізований комутатор без будь-якої адресної інформації в кадрі Ethernet. Спеціалізовані комутатори встановлені згідно з наперед визначеним та сформованим розкладом: на кожному часовому зміщенні (offset time) запланований кадр відправляють від одного порту до іншого без інтерпретації адреси.

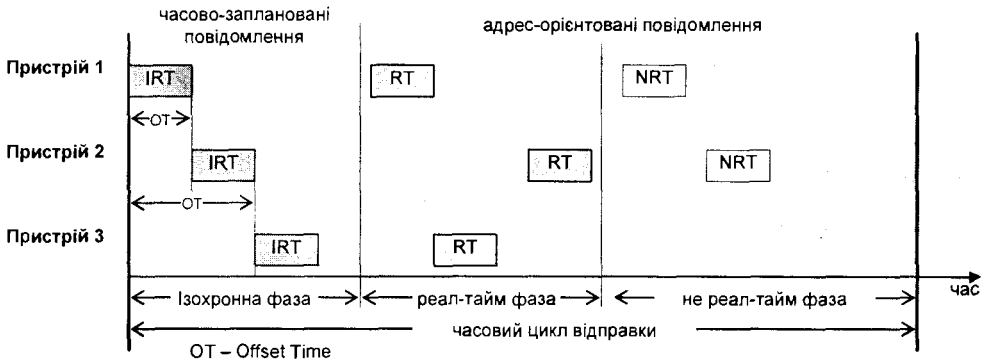


Рис. 10.17. Організація циклу EPL

У наступній часовій фазі, яка називається Реал-тайм Фаза, спеціалізовані комутатори перестроюються на зв'язок, який базується на адресі, та поводяться як стандартні комутатори Ethernet. У цій стадії передаються RT та NRT-кадри, які описані при розгляді PROFINET CBA. Всі PROFINET-комутатори синхронізовані і використовують механізми IEEE 1588 з штампкою «на льоту» для надання їм розкладу циклів IRT з похибкою в одну мікросекунду.

PROFINET CBA та IO не потребують ніякого спеціального обладнання для RT-зв'язку. Для гарантування надійної роботи PROFINET IO потребує 100 Мбіт/с та комутуючий повнодуплексний Ethernet. Для IRT необхідний спеціальний PROFINET-Ethernet комутатор, який може бути інтегрований у мережні пристрої для побудови всіх можливих топологій.



Контрольні запитання до розділу 10

1. Що стало передумовою та з чим пов'язані труднощі використання TCP/IP та Ethernet на рівні промислових мереж?
2. Які стандарти кадрів Ethernet сьогодні підтримують виробники мережних карт? Як вони описані в контексті моделі OSI?

3. Перерахуйте основні характеристики Ethernet з різними фізичними середовищами: тип фізичного середовища, топологія, метод доступу до шини, бітова швидкість, інтерфейси конекторів, використання комутаторів/концентраторів.
4. Поясніть призначення контактів RJ-45 для Ethernet. Поясніть призначення прямого та перехресних кабелів. Що визначає властивість Auto-MDI/MDI-X для комутаторів/концентраторів?
5. У чому відмінність роботи концентраторів та комутаторів 2-го рівня? Як відрізняються комутатори за способом передачі кадрів? Чим відрізняються керовані від звичайних комутаторів?
6. Що визначає властивість комунікаційного обладнання Auto-negotiation і які правила її використання?
7. Поясніть, що таке СКС.
8. Поясніть основні відмінності структури кадрів Ethernet II DIX та IEEE 802.3.
9. Як функціонує методика фільтрації кадрів, виділення віртуальних мереж та визначення пріоритетності кадрів на рівні керованих мостів, що працюють на базі стандарту 802.1Q?
10. Поясніть основні принципи IP адресації. Які версії протоколу IP використовуються сьогодні?
11. Яким чином функціонує IP-адресація вузлів на мережах Ethernet, як відбувається прив'язка IP-адрес до MAC-адрес? Що таке ARP-кеш?
12. Які Ви знаєте протоколи для автоматичного розподілу IP-адрес між мережними вузлами? Поясніть, що таке пул адрес та час оренди в контексті DHCP.
13. Призначення протоколу IGMP. Що таке IGMP-snooping, в яких мережних засобах і в якому випадку його необхідно використовувати?
14. Розкажіть про призначення та функціонування методики NAT-трансляції.
15. Які основні призначення транспортного протоколу? Як функціонує протокол UDP?
16. Поясніть на базі примітивів сокетів, як функціонує протокол TCP? Порівняйте його з протоколом UDP.
17. Яке призначення TCP/UDP портів? Які зарезервовані номери портів Ви знаєте? Які номери назначаються клієнтським портом?
18. Наведіть основні відмінності «промислового» Ethernet від «офісного» Ethernet.
19. Які особливості враховують стандарти на побудову СКС у промислових будівлях? Які кабелі та роз'єми рекомендуються для промислового Ethernet?
20. Які основні недоліки використання звичайного та комутованого Ethernet в якості реал-тайм рішення?
21. Що таке RTE? Якими показниками характеризують мережі RTE в IEC 61784-2?
22. На які класи можна поділити RTE мережі, залежно від вимог до реального часу? Яке призначення кожного класу та які їх основні характеристики?
23. Основи функціонування MODBUS RTPS, EtherNet/IP, P-NET on IP та Vnet/IP.
24. Основи функціонування Ethernet Powerlink, TCnet, EPA.
25. Основи функціонування SERCOS та EtherCAT.
26. Основи функціонування та відмінності PROFINET CBA і PROFINET IO.

КОМУНІКАЦІЙНА АРХІТЕКТУРА ДЛЯ ЕЛЕКТРОПРИВОДІВ

11.1. Способи інтеграції пристроїв PDS в автоматизовані системи управління

11.1.1. Класичний підхід

З розвитком систем управління електроприводами *PDS* (Power Drive Systems) виникла необхідність в їх інтеграції в єдину автоматизовану систему для дистанційного управління та контролю за двигунами. Класичним підходом до підключення таких систем вважається використання уніфікованих аналогових, дискретних та імпульсних (частотних) сигналів. Наприклад, для управління частотним перетворювачем виділяється один аналоговий вихід контролера, який підключається до аналогового входу частотного перетворювача, що відповідає за задане значення частоти (швидкості) обертів двигуна. Дискретні входи перетворювача використовуються для зупинки та запуску двигуна, переключення частоти, а виходи — для контролю за станом двигуна, аваріями. Аналогові виходи частотного перетворювача служать для контролю за поточною швидкістю, напругою, струмом і т.д.

11.1.2. Переваги використання промислових мереж

Зі збільшенням інформаційного обміну між контролерами та частотними перетворювачами значно зростає кількість необхідних каналів, що призводить до додаткових витрат як на засоби автоматизації, так і на інсталяцію та експлуатацію систем. Завдяки промисловим мережам ефективно використовуються всі ресурси частотних перетворювачів незалежно від кількості задіяних функцій, покращується заводостійкість та збільшується дальність зв'язку. Крім того, підключення за цифровими інтерфейсами дає додаткові можливості та зручності при конфігуруванні частотних перетворювачів, зокрема віддалене налагодження, збереження плинної конфігурації, її перенесення на інші приводи і т.д.

11.1.3. Проблеми використання промислових мереж

З появою промислових мереж у системах PDS конфігурація та дія систем управління стали помітно залежати від архітектурних рішень, вибраних для інтеграції їх у загальний процес управління. Ці рішення суттєво відрізнялися одне від одного, однак вирішували однакові за типом завдання. І для виробників автоматизованих систем управління, і для виробників обладнання виникла необхідність у стандартизації підходів.

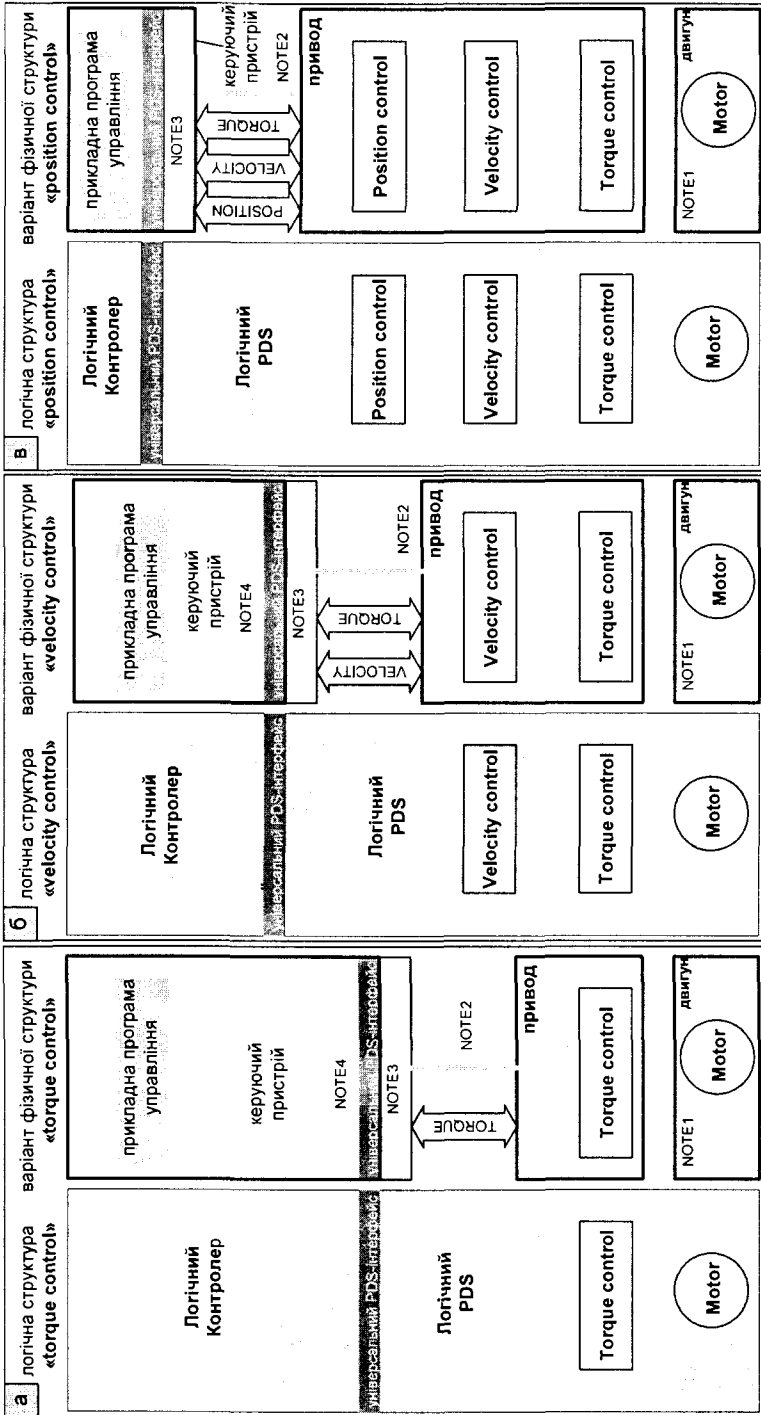
11.2. Стандарт на універсальний інтерфейс PDS

11.2.1. Загальні підходи до стандартизації інтерфейсу PDS

11.2.1.1. Ідея створення та призначення стандарту. Ідеєю створення стандарту для цифрових комунікацій з електроприводами було об'єднання декількох промислових рішень шляхом виділення загальних елементів, структури для створення універсального інтерфейсу та профілів для PDS. Враховуючи, що для PDS-систем уже функціонувало декілька стандартів у загальній серії IEC 61800 «Adjustable speed electrical power drive systems», а також відсутність на той момент функціонуючих стандартів МЕК на промислові мережі, нові стандарти на інтерфейс електроприводів увійшли до групи 61800, як частина 7. Таким чином, стандарти **61800-7** описують універсальний інтерфейс між системою автоматизації та PDS.

11.2.1.2. Місце Універсального Інтерфейсу в логічній структурі системи управління PDS. Згідно з концепцією 61800-7, модель системи PDS являє собою поєднання приводу, тобто безпосередньо модуля PDS, двигуна та комунікаційного інтерфейсу приводу. Таку модель будемо називати Логічним PDS (Logical Power Drive System) або просто **Логічним Приводом**. Функції управління Логічним Приводом реалізуються в одному або більше **Логічних Контролерах** та виконуються відповідно з прикладною програмою управління. Таким чином, структура автоматизованої системи являє собою поєднання Логічного Приводу разом з Логічним Контролером. Перший стандарт 61800-7-1 є базовим і визначає **Універсальний Інтерфейс** (Generic Interface) доступу до Логічного Приводу з боку Логічного Контролера.

11.2.1.3. Функціональні можливості Логічного Приводу. Функціональні можливості Логічного Приводу можуть бути розділені на: управління позиціонуванням (*position-control*), управління швидкістю (*velocity-control*) та управління моментом (*torque-control*). Залежно від реалізованих функцій Логічного Приводу, їх фізична структура, згідно зі стандартом 61800-7-1, буде відрізнятися. На рис. 11.1(а) показана структура автоматизованої системи управління, в якій Логічний Привод повинен управляти моментом двигуна. В цьому випадку прикладна програма через промислові мережі видає значення заданого моменту. Універсальний PDS-інтерфейс забезпечує однаковий, з точки зору прикладної програми, спосіб зміни заданого моменту, незалежно від виробника приводу. Аналогічно на рис. 11.1 (б) та рис. 11.1 (в) показані структури систем, що відповідають управлінням швидкістю та управління позиціонуванням, де прикладна програма керуючого пристрою задає необхідні швидкість та позицію. Функції, які відсутні в структурі пристрою PDS, можуть бути реалізовані в керуючому пристрої (контролері).

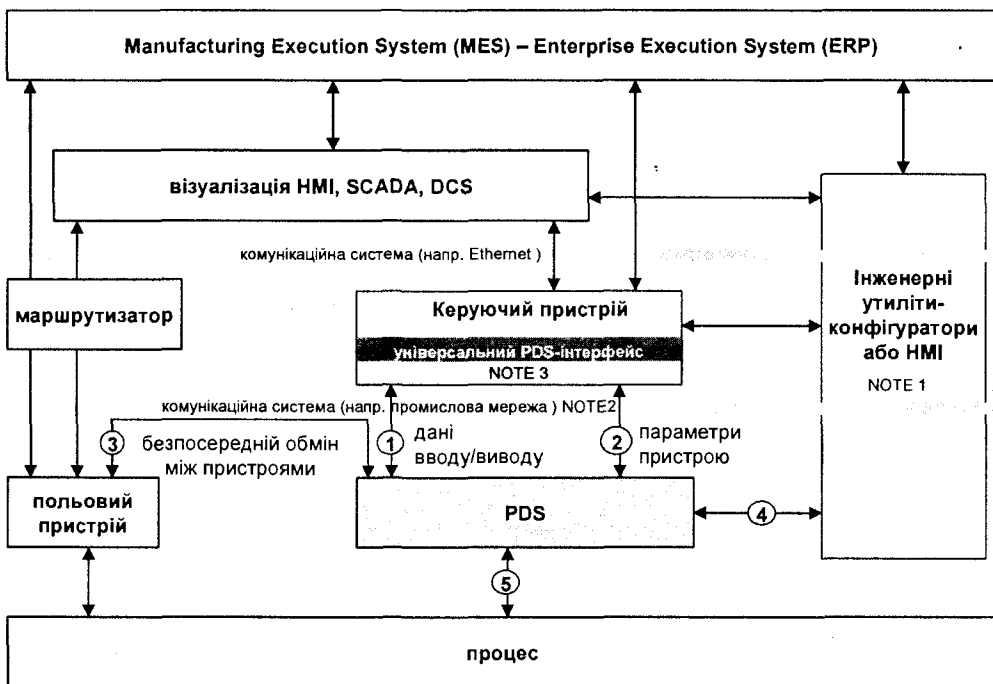


NOTE1 – в загальному, PDS модель включає двигун
 NOTE2 – промислова мережа для передачі даних таких як швидкість, момент, позиція
 NOTE3 – відображення та адаптація відповідно до конкретної промислової мережі
 NOTE4 – прикладна програма може включати також управління позиціонуванням, швидкістю(для варіанту а)

Рис. 11.1. Фізична структура різних типів систем PDS

Слід звернути увагу на розміщення універсального інтерфейсу PDS. Це логічний, а не фізичний інтерфейс. Якщо стандарти промислових мереж визначають правила поєднання між пристроями, то стандарти IEC 61800-7 визначають правила роботи з об'єктами Логічного Приводу, доступ до яких буде проводитись через конкретні промислові мережі. Це значить, що одні й ті ж профілі IEC 61800-7 можуть функціонувати на різних типах промислових мереж.

11.2.1.4. Місце PDS у типовій структурі автоматизованої системи. З точки зору типової структури управління, PDS є польовим пристроєм. На рис. 11.2 показано місце пристрою PDS у типовій структурі автоматизованої системи та його зв'язки з іншими засобами.



NOTE 1 – інженерна утиліта може бути єдиною для різних мереж, або різні утиліти в залежності від типу необхідної мережі

NOTE 2 – комунікаційна система при необхідності може бути резервована, це ніяк не впливає на універсальний PDS-інтерфейс

NOTE 3 – відображення або адаптація відповідно до конкретної промислової мережі

Рис. 11.2. Місце PDS у типовій структурі автоматизованої системи

Фізично модуль PDS може з'єднуватись для передачі різних типів даних:

1. Дані вводу/виводу (IO Data). Цей інтерфейс є частиною інтерфейсу PDS, який забезпечує управління та контроль приводу.

2. Параметри пристрою (Device Parameters). Цей інтерфейс забезпечує доступ до параметрів пристрою для віддаленого конфігурування, моніторингу, діагностики. Параметри пристрою доступні через універсальний PDS-інтерфейс.

3. З'єднання між польовими пристроями (peer to peer device). Цей інтерфейс забезпечує безпосереднє з'єднання між польовими пристроями. Він залежить від особливостей промислової мережі і не входить до універсального інтерфейсу PDS.

4. Параметри пристрою (інший або місцевий інтерфейс). Забезпечує доступ інженерних утиліт до параметрів пристрою через унікальний інтерфейс, який не входить до універсального інтерфейсу PDS.

5. Інтерфейс до процесу. Зв'язок з технологічним процесом (датчики, виконавчі механізми) не охоплюється універсальним інтерфейсом PDS.

11.2.2. Функціональні елементи Логічного Приводу

11.2.2.1. Призначення Функціональних Елементів у Логічному PDS. Логічний Привод складається з **Функціональних Елементів (FE)**, які включають параметри та алгоритми для опису його поведінки. Для Логічного PDS доступні такі FE (рис. 11.3):

1. Функціональні Елементи ідентифікації пристрою (Device identification FE), які вміщують параметри для ідентифікації фізичних пристроїв:

- Profile ID (профіль);
- Manufacturer ID (виробник);
- Product ID (тип та модель пристрою);
- Serial number (серійний номер);
- Hardware revision (версія апаратного забезпечення);
- Software revision (версія ПЗ);
- Tag (поле користувача);
- Location (мітка користувача);
- Profile defined (додаткові параметри профілю).

2. Функціональні Елементи управління пристроєм (Device control FE), які включають автомат станів для управління станом приводу залежно від тривоги та попереджень.

3. Комунікаційний Функціональний Елемент Логічного Приводу (Communication FE), який складається з визначених комунікаційних параметрів мережі між керуючим пристроєм та приводу; а також включає автомат станів підключення пристроїв.

4. Базові Функціональні Елементи приводу (Basic drive FE), які включають базові функції управління позиціонуванням, швидкістю та моментом.

5. Опціональні прикладні Функціональні Елементи (Optional application FE), які включають додаткові опціональні функції управління операцій з використанням убудованих входів/виходів (енкодери, кінцеві вимикачі); не описуються стандартом;

6. Функціональні Елементи іншого або місцевого інтерфейсу (Other interface FE). Комунікаційні можливості для Логічних Приводів, які не входять до стандарту IEC 61800-7.

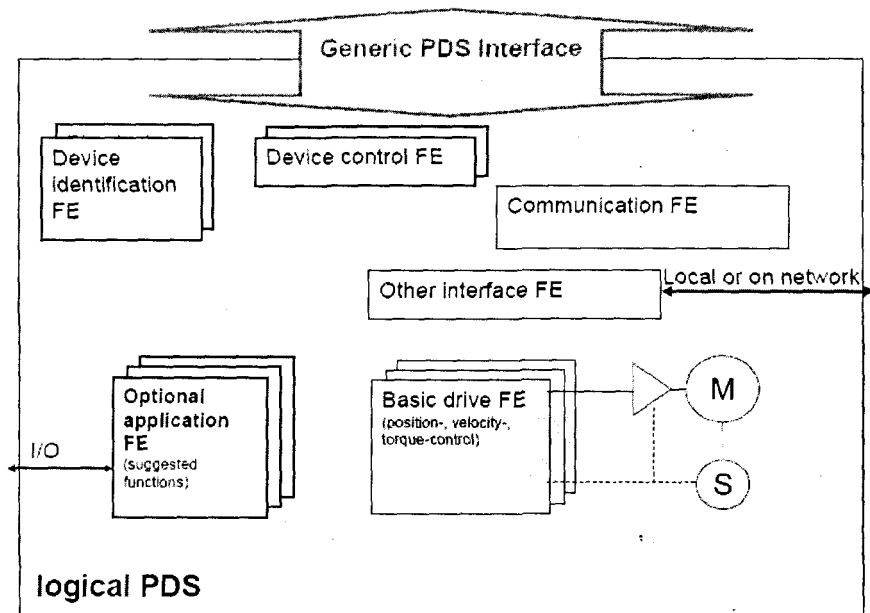


Рис. 11.3. Функціональні Елементи в Логічному PDS

11.2.2.2. Основи використання FE для управління PDS. Нагадаємо, що універсальний PDS інтерфейс використовується для двох різних призначень: для налаштування роботи та обслуговування приводу за допомогою інженерних утиліт і для управління приводом з прикладної програми (див. рис.11.2). У першому випадку використовуються такі типи операцій, як ідентифікація, конфігурування, запис та зчитування параметричних даних, а управління Логічним Приводом з боку Логічного Контролера (обмін даними процесу) в основному зводиться до управління та контролю трьох функціональних елементів: Device Control, Communication та Basic drive, а саме (див. рис. 11.4):

- відправка команди від прикладної програми до PDS (COMMAND);
- відправка заданих значень від прикладної програми до PDS (Set Point);
- відправка статусу від PDS до прикладної програми (STATUS);
- відправка дійсних значень від PDS до прикладної програми (Actual Values).

11.2.2.3. Контроль та управління автоматом станів FE (COMMAND та STATUS). Кожен з наведених вище Функціональних Елементів управляється автоматом станів. Тобто, в кожний момент часу Функціональний Елемент перебуває у певному стані, який визначає його поведінку в даній ситуації. Визначити поточний стан Функціональних Елементів можна за допомогою змінної статусу (STATUS), а управляти станом — за допомогою команди (COMMAND). Перехід від стану до стану Функціональних Елементів проходить у момент отримання конкретної команди або виникнення певної події у приводі.

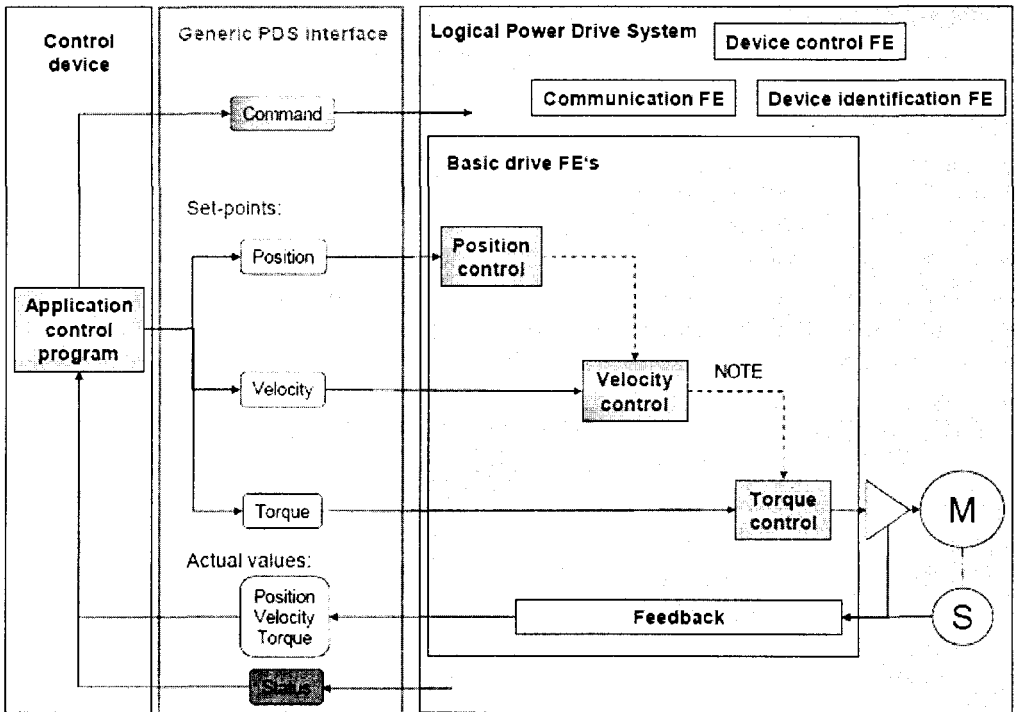


Рис. 11.4. Використання універсального інтерфейсу для управління PDS

11.2.2.4. Device Control FE. В універсальному інтерфейсі PDS закладений єдиний автомат станів для управління обробками помилок та загальним станом PDS. Вона закладена в функціональному елементі Device Control. Якщо PDS не може забезпечити правильну роботу, то він переходить у стан помилки (*Faulted State*). Для переходу його у нормальний стан (*No Faulted*) необхідна команда скидання помилки (*Reset Fault*) з керуючого пристрою або локально, за місцем.

Перед виникненням помилки, PDS може подати сигнал попередження (*Warning*). Враховуючи відсутність необхідності підтвердження попереджувального сигналу, він не входить до автомату станів.

Таким чином, автомат станів даного Функціонального Елемента включає всього два рівня: помилка і нормальний стан. Однак, залежно від профілю пристрою, кількість станів може бути розширена. Стани Device Control відображаються одним бітом змінної статусу, а для управління машиною станів використовується біт-команда на скидання помилки (*Reset Fault*). Залежно від профілю PDS додатково можуть бути використані й інші команди.

Через параметри Функціонального Елемента можна доступитись до буфера останніх помилок та попереджень, дізнатись про їх кількість, час виникнення тощо.

11.2.2.5. Communication FE. Комунікаційний Функціональний Елемент вміщує всі автомати станів та параметри для доступних промислових мереж PDS. Якщо профіль підтримує декілька мереж, то для PDS доступні кілька екземплярів комунікаційного FE.

Функціональний Елемент може бути в двох станах: обмеженому комунікаційному (*Limited communication*) та нормальному комунікаційному (*Normal communication*). В обмеженому комунікаційному стані PDS може тільки контролюватися, але не управлятися з боку Універсального Інтерфейсу. Тобто управління доступне тільки через Other interface FE. В нормальному режимі для Універсального Інтерфейсу доступні функції управління. Доступ до певних параметрів приводу у нормальному режимі може бути обмежений через додаткові настройки та підстани, визначені профілем.

Комунікаційний стан відображається в змінній статусу у вигляді біту Normal Communication. При включенні живлення PDS він переводиться в обмежений комунікаційний режим. Для переводу в нормальний режим використовується біт-команда *Run communication*. Для профілів PDS, які підтримують синхронний режим, додатково визначені біт статусу *Synchronized* та біт-команда *Synchronize/Do not Synchronize*.

Параметри Комунікаційного Функціонального Елемента визначають тип і настройки промислової мережі. Ці параметри залежать від профілю.

11.2.2.6. Basic Drive FE. Ці Функціональні Елементи включають автомати станів та параметри для управління базовими функціями PDS. В Універсальному Інтерфейсі PDS для автомата станів визначені такі стани:

– *operating* — PDS через локальний чи віддалений інтерфейс реагує на команди та задані значення, а також видає значення статусу та необхідні дійсні значення;

– *not operating* — PDS через локальний чи віддалений інтерфейс не реагує на зміну заданого значення, однак приймає команди та реагує на них.

Опціонально можуть бути визначені такі стани:

– *local control* — PDS приймає команди та задані значення через локальний інтерфейс;

– *remote control* — PDS приймає команди та задані значення через віддалений інтерфейс.

Залежно від режиму роботи, профілем можуть бути визначені додаткові стани.

Значення режимів залишається слово «контролюється» відповідними бітами статусу Operating та Remote Control.

При подачі живлення на PDS він стартує у режимі «not operating» та «local control». Команда *Operate* переключає автомат станів між режимами «operating» та «not operating». Біт-команди *Local/Remote* переводять привод у відповідний режим.

У параметрах Basic Drive FE визначаються:

- режим роботи PDS (application mode);
- уставки даних вводу/виводу;
- поведінка функціонального елемента PDS залежно від режиму його роботи.

11.2.3. Прикладні режими PDS

11.2.3.1. Загальні положення. Залежно від доступних функцій апаратного та програмного забезпечення PDS, Логічний Привод може бути у декількох прикладних режимах:

- управління позиціонуванням (*position-control*),
- управління швидкістю (*velocity-control*);
- управління моментом (*torque-control*).

У таб. 11.1 наведене призначення обов'язкової уставки залежно від вибраного режиму роботи. Можуть бути доступні також опціональні уставки.

Таблиця 11.1

ПРИЗНАЧЕННЯ УСТАВОК ЗАЛЕЖНО ВІД РЕЖИМУ РОБОТИ

Прикладний Режим (Application Mode)	Уставка Set-Point	Функції в PDS	Рис.
Torque Preset	індекс моменту	розв'язане управління моментом	11.5 (а)
Torque Control	значення моменту	розв'язане управління моментом	11.5 (а)
Torque Control with velocity feedback	значення моменту	управління моментом зі зворотним зв'язком по швидкості	11.5 (б)
Velocity Preset	індекс швидкості	переключення на фіксовані швидкості з контролем моменту та швидкості	11.6 (а)
Velocity Control	значення швидкості	розв'язане управління моментом та швидкістю	11.6 (б)
Velocity Control with position feedback	значення швидкості	управління моментом та швидкістю зі зворотним зв'язком по позиції	11.6 (в)
Position Preset	індекс позиції	перехід приводу на фіксовані позиції за зворотним зв'язком	11.7 (а)
Position Control	значення позиції	управління позицією, швидкістю або моментом	11.7 (б)

11.2.3.2. Режим управління моментом. У режимі управління моментом (див. рис. 11.5, (а)) прикладна програма керуючого пристрою задає необхідний момент, який регулюється пристроєм PDS з використанням зворотного зв'язку (канал вимірювання) або з використанням розрахованого значення. Може бути використаний зворотний зв'язок за швидкістю (див. рис. 11.5, (б)).

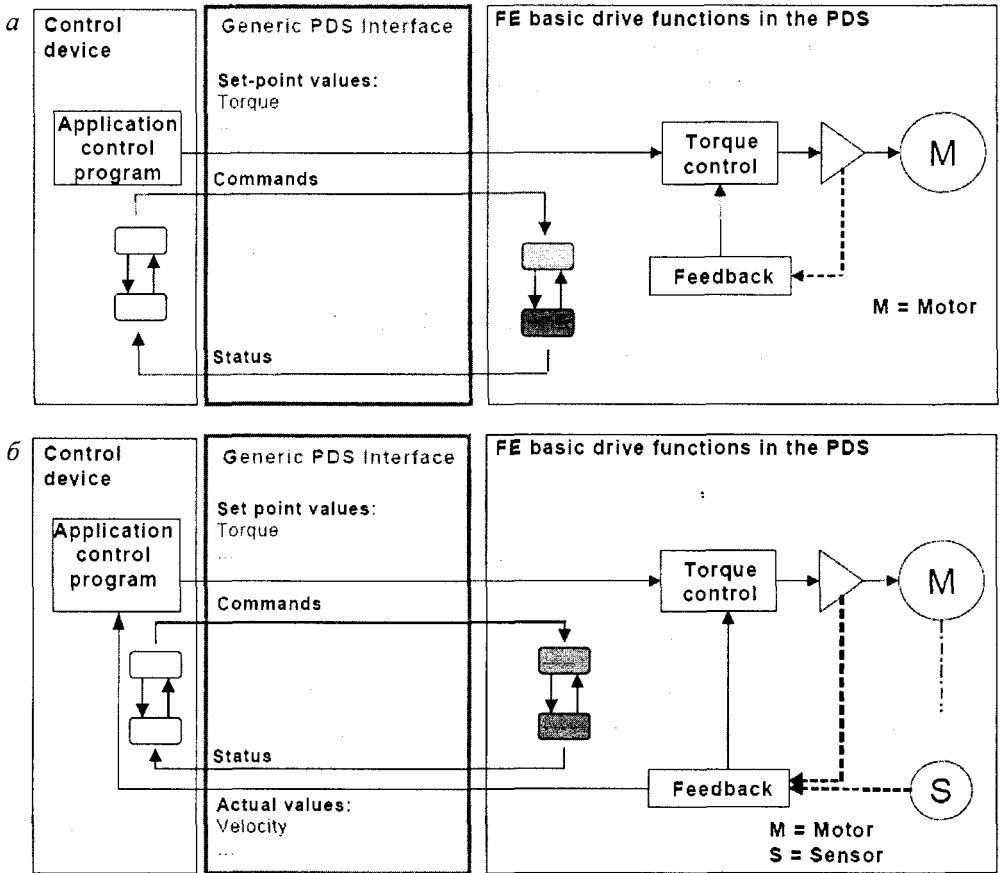


Рис. 11.5. Режим управління моментом без (а) та зі (б) зворотним зв'язком за швидкістю

11.2.3.3. Режим управління швидкістю. Можливі три випадки режимів управління швидкістю:

1. Режим з наперед установленими швидкостями (рис. 11.6, (а), коли задані значення швидкостей виставляються в момент конфігурування PDS, а в момент операційного функціонування проводиться їх вибір за індексом.

2. Режим регулювання швидкості (рис. 11.6, (б), де задане значення швидкості задається в уставці; опціонально можна також задавати момент; опціонально підтримується передача дійсного значення моменту та швидкості.

3. Режим регулювання швидкості зі зворотним зв'язком за позицією, який аналогічний попередньому, однак зі зворотним зв'язком за датчиком положення.

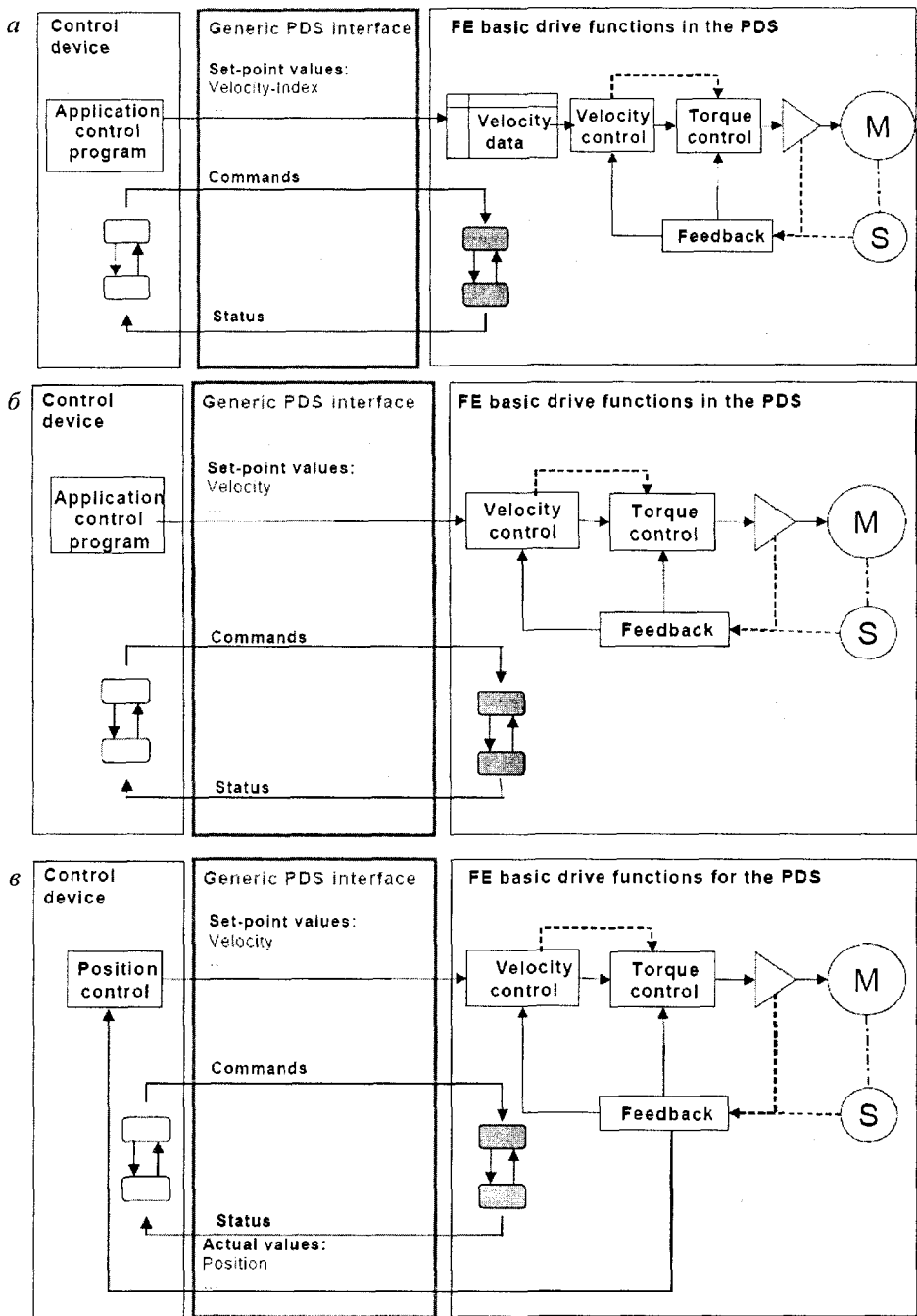


Рис. 11.6. Режими управління швидкістю:
a — з наперед установленими швидкостями; *б* — з регулюванням швидкості;
в — зі зворотним зв'язком за позицією

11.2.3.4. Режим управління позиціонуванням. У режимі з наперед установленими позиціями (рис. 11.7, (а)) у момент операційної роботи уставкою вибирається необхідна точка контролю позиції. Опціонально можливо вказати траєкторію між двома сусідніми точками.

У режимі управління позиціонуванням (рис. 11.7, (б)) в уставці задається задана позиція. Опціонально можна задати необхідну швидкість та момент. Досягнення необхідної позиції сигналізується змінною стану. PDS може передавати прикладній програмі дійсні значення позиції, швидкості та моменту.

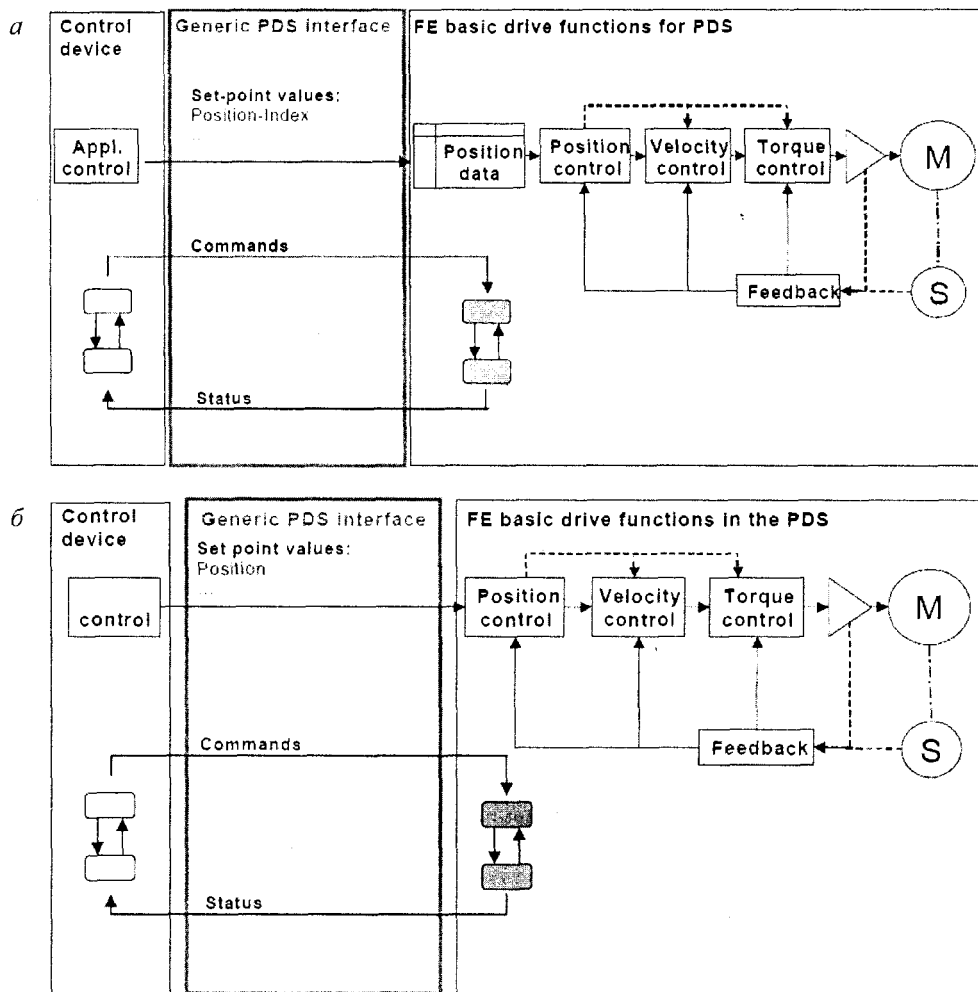


Рис. 11.7. Режими управління позиціонуванням:

а — з наперед установленими позиціями; б — з регулюванням позиції

11.2.4. Структура стандартів IEC 61800-7

11.2.4.1. Типи профілів PDS. Крім базового стандарту, на універсальний інтерфейс IEC 61800-7-1, в якому даються основні і загальні для всіх профілів концепції зв'язку та управління PDS, група стандартів також включає специфікації на профілі IEC 61800-7-200 (201-204) та їх відображення на промислові мережі IEC 61800-7-300 (301-304). На рис. 11.8 показана структура всієї серії стандартів IEC 61800-7. Як видно з рисунка, у стандартах визначені 4-ри типи профілів PDS:

1. CiA 402 (CAN in Automation);
2. CIP Motion (ODVA);
3. PROFIDrive (PROFIBUS International);
4. SERCOS (SERCOS International).

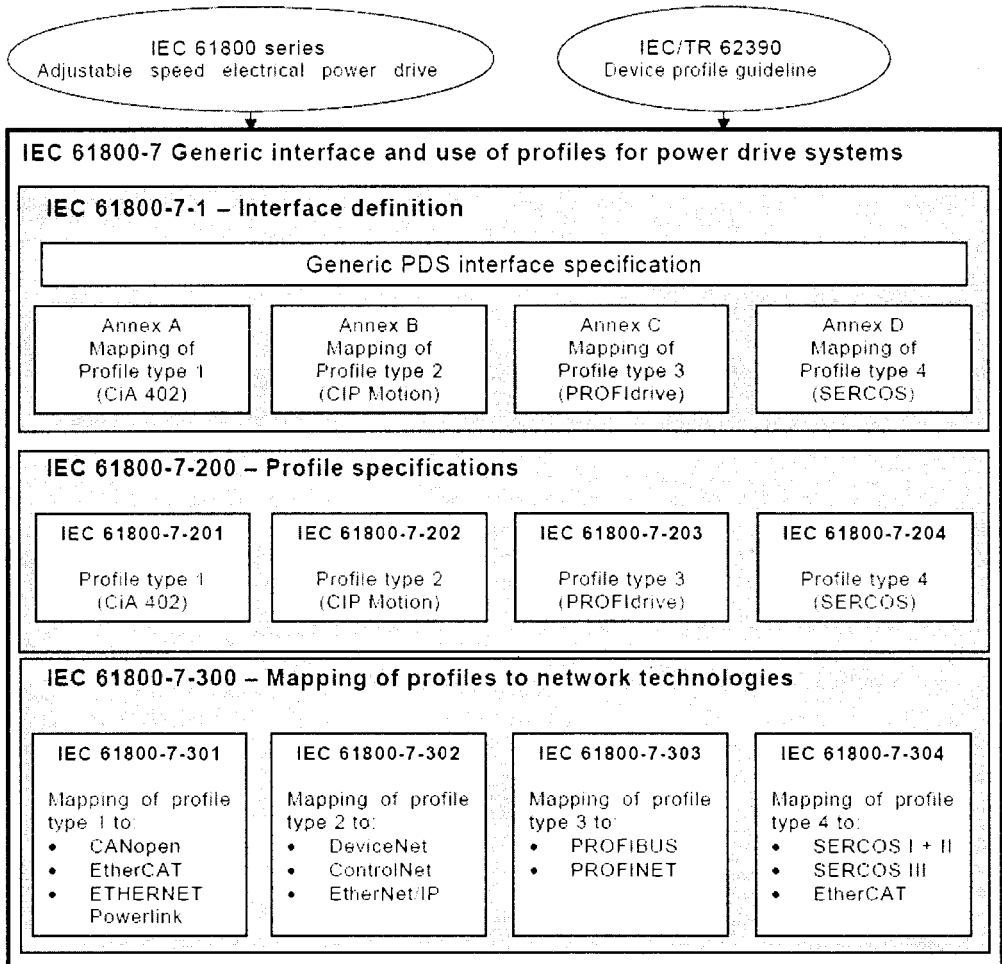


Рис. 11.8. Структура стандартів IEC 61800-7

11.2.4.2. Відображення профілів PDS на мережні технології. Ці профілі конкретизують і доповнюють універсальний інтерфейс PDS та потрібні для правильного написання прикладної програми, яка управляє Логічним Приводом. Однак, профіль Приводу не визначає механізм взаємодії через мережні технології. Припустимо, відомо, яка команда повинна бути надіслана для зупинки двигуна, однак невідомо, в якій мережній змінній вона міститься. Для визначення конкретної реалізації профілю PDS він відображається на конкретну промислову мережу. В групі стандартів IEC 61800-7-300 профілі приводів описуються в контексті конкретної мережної технології.

У перших робочих редакціях стандартів IEC 61800-7 було визначено 5 профілів для PDS. До діючої офіційної редакції не увійшов один із популярних нині профілів DRIVECOM (INTERBUS), який сьогодні використовуються у багатьох частотних перетворювачах. У наступних главах будуть розглянуті 2 профілі приводів та їх відображення на мережі: PROFIDRIVE та CiA402.

11.3. Профіль PROFIDrive

PROFIDrive (Profile Drive Technology) — стандарт профілю для PDS, який запропонований організацією PROFIBUS & PROFINET International для використання його спільно з мережами PROFIBUS та PROFINET. У даній главі PROFIDrive розглядається тільки в контексті PROFIBUS DP. Слід зазначити, що використання частотних перетворювачів в мережі PROFIBUS не обов'язково передбачає послугоування профілю PROFIDrive.

Щоб налаштувати роботу приводу потрібні ациклічні сервіси. В режимі функціонування використовуються циклічні сервіси.

11.3.1. Функціональні елементи PROFIDrive

11.3.1.1. Слова управління та статусу (STW та ZSW). Автомати станів функціональних елементів управляються за допомогою керуючих слів Control (Control Words — **STW**), через які передається команда, а контролюються через слова статусу Status (Status Words — **ZSW**).

Дані входів/виходів включають один або декілька сигналів уставок та дійсних значень для управління та контролю за приводом. Можливі комбінації сигналів складені в наперед визначених стандартних телеграмах.

11.3.1.2. Device Control FE. Автомат станів може перебувати в режимі *No Fault* ($ZSW1.3=0$) або *Fault Present* ($ZSW1.3=1$). Додатково визначені стани *Warning Present* ($ZSW1.7=1$) та *No Warning* ($ZSW1.7=0$). Підтвердження аварії проводиться командою *Fault Acknowledge* ($STW1.7=1$). Поведінка приводу в кожному стані визначається у самому пристрої, наприклад при конфігуруванні.

11.3.1.3. Communication FE. Для базової моделі PROFIDrive визначені такі стани, які належать до Комунікаційного Функціонального Елемента (рис. 11.9):

1. **Offline:** відсутній зв'язок по мережі.

2. **Preparation (Limited Communication):** у цьому режимі працюють сервіси Acyclic Data Exchange та Alarm Mechanism; обмін даними I/O не підтримується; можливі конфігурування та діагностика пристрою.

3. **Synchronisation** (Not Synchronized): у цей час активуються сервіси циклічного обміну та відбувається синхронізація.

4. **Operation** (Normal Operation, Synchronized): усі сервіси доступні.

Перехід від стану до стану проводиться комунікаційними сервісами PROFIBUS або самим приводом внаслідок виникнення помилки.

11.3.1.4. Basic Drive FE. В стандарті 61800-7-1 автомат станів складається з двох станів: *operating* та *not operating*. У технології PROFIdrive автомат станів більш комплексний і охоплює додаткові стани та команди. Розглянемо більш детально загальний автомат станів для приводів PROFIdrive, а також для прикладних режимів управління моментом та швидкістю.

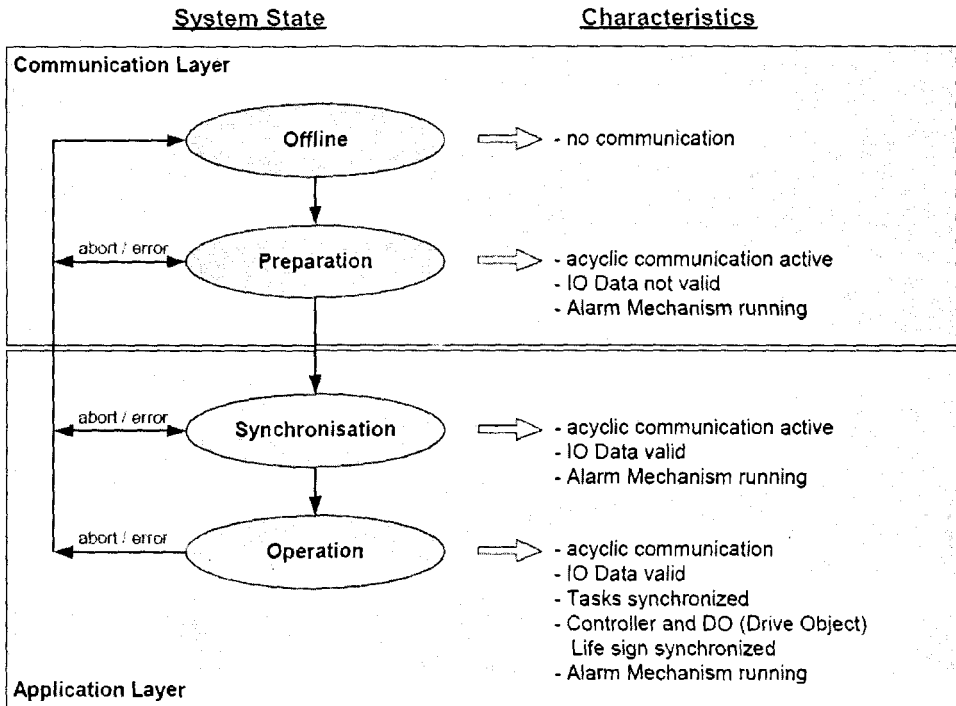


Рис. 11.9. Відображення станів комунікаційного функціонального елемента

11.3.2. Загальний автомат станів для приводів PROFIdrive

Загальний автомат станів PROFIdrive показаний на рис.11.10. Кожен з них нумерується S1-S5. Стан **Operation** (S4) співпадає з визначеним у стандарті і відповідає за операційний режим роботи приводу. Даний стан сигналізується 2-им бітом у першому слові статусу (**Operation Enabled** — ZSW1.2=1). Стандартний стан Not Operating в PROFIdrive ділиться на декілька підрежимів: **Switching On Inhibited**, **Ready for Switching On**, **Switched On**, а також **Switching OFF**.

Логічний контролер може управляти автоматом станів тільки у випадку якщо:

- комунікаційний функціональний елемент перебуває в стані Operation;
- виставлений режим віддаленого управління *Control Requested* (ZSW1.9=1);
- контролер відправляє команду *Control By PLC* (STW1.10=1), повідомляючи, що він готовий до управління і його значення даних є правильними;
- контролер відправляє команду *ON* (STW1.0=1), включаючи PDS;
- PDS сигналізує про готовність до операційного режиму сигналом *Ready To Operate* (ZSW1.1=1).

На рис. 11.10 у прямокутниках показані стани загального автомату та біти статусу, якими вони сигналізуються. Стрілками показані події та команди, які приводять до переведення автомату з одного стану в інший. Подія *Standstill detected* сигналізує про виявлення зупинки двигуна. Можливе аварійне переведення приводу в стан S1 (на діаграмі позначено індексом *b*).

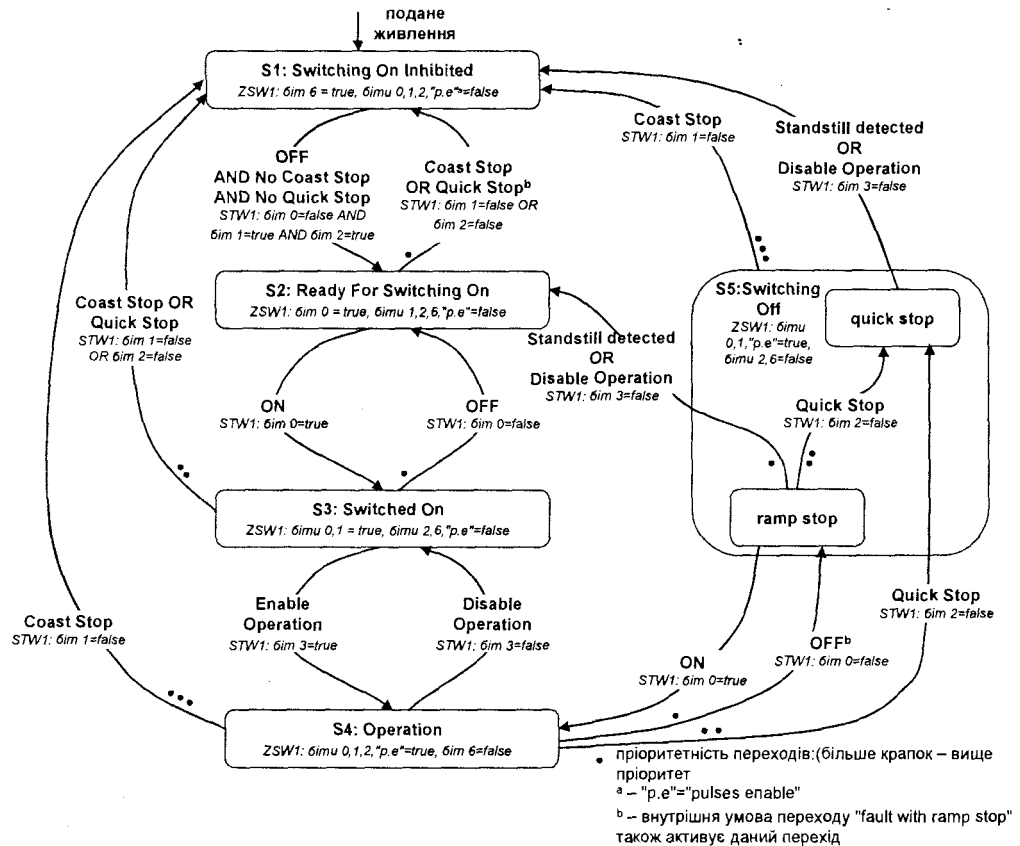


Рис. 11.10. Загальний автомат станів PROFIDrive для базового функціонального елемента

11.3.3. Операційна робота приводів PROFIDrive в режимі управління швидкістю

У PROFIDrive прикладні режими роботи приводів виділяються у шість прикладних класів (Application Class, AC). Всі режими управління моментом та розімкнене управління швидкістю належать до AC1 «Standard Drive». На рис. 11.11 показана функціональна діаграма операційного режиму приводу StandardDrive.

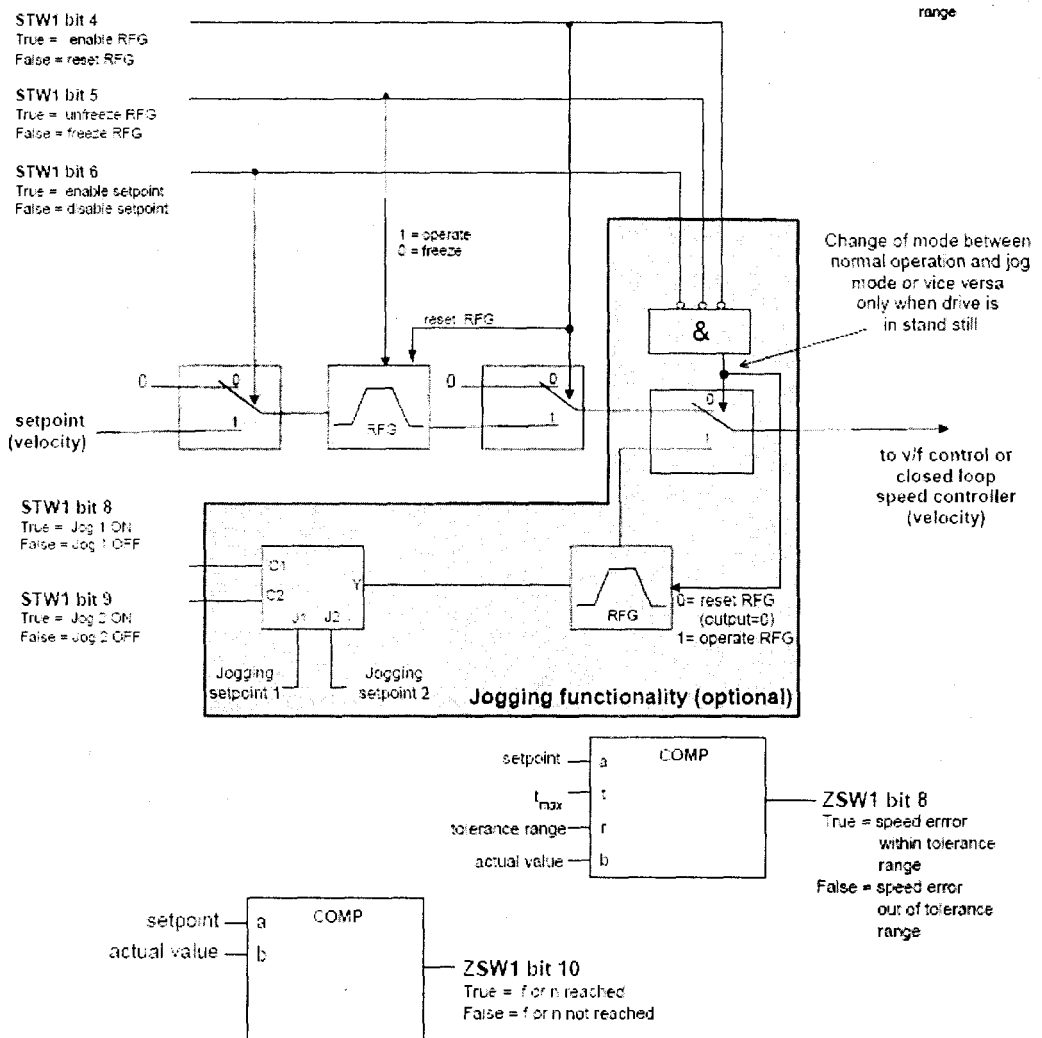


Рис. 11.11. Загальний автомат станів PROFIDrive для приводу StandardDrive

У таблиці 11.2 зведені всі команди PROFIdrive, в яких біти 4, 5, 6, 8, 9 тільки для режиму управління SatndardDrive.

Таблиця 11.2

КОМАНДИ PROFIDRIVE

Назва Command IEC 61800-7	Назва PROFIdrive	Біт STW1	Призначення команди
	1 — ON 0 — OFF (OFF1)	0	Команда на включення/виключення основного вимикача живлення. При подачі команди OFF1 двигун буде плавно зупинитись до повної зупинки, після чого вимикач вимкнеться.
	1 — No Coast Stop (No OFF2) 0 — Coast Stop (OFF2)	1	Команда OFF2 негайно знімає напругу з двигуна основним вимикачем живлення. Двигун зупиняється за рахунок інерції.
	1 — No Quick Stop (No OFF3) 0 — Quick Stop (OFF3)	2	Команда на швидку зупинку двигуна.
OPERATE	1 — Enable Operation 0 — Disable Operation	3	Команда на включення/виключення операційного режиму. В операційному режимі всі уставки функціонують.
	1 — Enable Ramp Generator 0 — Reset Ramp Generator	4	Активация нарощування уставки
	1 — Unfreeze Ramp Generator 0 — Freeze Ramp Generator	5	Фіксація заданої швидкості
	1 — Enable Setpoint 0 — Disable Setpoint	6	Активация уставки
RESET FAULT	1 — Fault Acknowledge 0 — No Significance	7	Підтвердження помилки. Реакція приводу залежить від типу помилки.
	1 — JOG1 ON 0 — JOG1 OFF	8	Активация фіксованої частоти 1
	1 — JOG2 ON 0 — JOG2 OFF	9	Активация фіксованої частоти 2
REMOTE LOCAL	1 — Control By PLC 0 — No Control By PLC	10	Команда на управління приводом з контролера.

В таблиці 11.3 наведені зведені біти статусу PROFIdrive, в яких біти 4, 5, 6, 8, 10 тільки для режиму управління SatndardDrive.

Приклад 11.1. PROFIDrive. Написання програми для організації взаємодії ПЛК S300 з ЧПД Danfoss

Завдання. Створити конфігурацію та програму користувача в пакеті STEP7 для управління (задана частота) та контролю (плинна частота, струм двигуна, напруга двигуна, аварії, попередження) частотними перетворювачами Danfoss типу FC100/200/300 з ПЛК Siemens типу S300 по мережі PROFIBUS DP з використанням профілю PROFIDrive.

Рішення. Частотні перетворювачі Danfoss FC100/200/300 можуть управлятися із використанням профілів FC або PROFIDrive. Тип профілю, а також інші комунікаційні настройки вибираються при кофігуруванні частотно-го перетворювача.

Таблиця 11.3

ЗВЕДЕНІ БІТИ СТАТУСУ PROFIDrive

Назва STATUS IEC 61800-7	Назва PROFIDrive	Біт ZSW1	Призначення
	1 — Ready To Switch On 0 — Not Ready To Switch On	0	Джерело живлення увімкнено, електроніка ініціалізована
	1 — Ready To Operate 0 — Not Ready To Operate	1	Повторює STW1.0
Operating	1 — Operation Enabled 0 — Operation Disabled	2	Привод в операційному/не операційному режимі
Faulted	1 — Fault Present 0 — No Fault	3	В буфері є непідтвердженні помилки. Підтвердження тільки неактивної помилки може зняти даний біт
	1 — Coast Stop Not Activated (No OFF 2) 0 — Coast Stop Activated (OFF 2)	4	Зупинка по відключенню напруги з двигуна активована
	1 — Quick Stop Not Activated (No OFF 3) 0 — Quick Stop Activated (OFF 3)	5	Швидка зупинка активована
	1 — Switching On Inhibited 0 — Switching On Not Inhibit	6	Привод увімкнений в режимі заборони функціонування
Warning	1 — Warning Present 0 — No Warning	7	В буфері є попередження.
	1 — Speed Error Within Tolerance Range 0 — Speed Error Out Of Tolerance Range	8	Якщо =1 — розузгодження по швидкості (задана-вимірjana) в межах норми

Назва STATUS IEC 61800-7	Назва PROFIdrive	Біт ZSW1	Призначення
Remote control	1 — Control Requested 0 — No Control Requested	9	Управління з контролера можливе/неможливе. У випадку готовності до управління з контролера привод виставляє даний біт в 1
	1 — f Or n Reached Or Exceeded 0 — f Or n Not Reached	10	Якщо =1 — частота/швидкість досягнула уставки

Для ПЛК S300 в середовищі STEP7 необхідно створити проект, налаштувати ПЛК на роботу з PROFIBUS DP в якості Ведучого. В якості Веденого добавляється пристрій типу FC100/200/300, як на рис. 11.12 (попередньо необхідно встановити потрібні файли .gsd).

Для даного пристрою вибирається кількість змінних обміну та їх консистентність. Для Danfoss 6 вхідних та вихідних слів даних процесу забезпечує 4-тий тип PPO, а модульна консистентність — нерозривність даних для всіх 12 байтів (6 слів). На вкладці Parameter Assignment властивостей вибраного ЧІР вибираються необхідні змінні для його вхідних/вихідних параметрів. Параметри Fieldbus CTW(слово COMMAND), Fieldbus REF (завдання) та Status Word (слово STAUS) — є наперед визначеними і не змінюються. Інші параметри контролю вибираємо відповідно: Frequency (дійсна частота), MotorCurrent (струм двигуна), Motor Voltage(напруга двигуна), VLT Alarm Word (слово аварій), VLT Warning Word(слово попереджень).

Відповідно до виділеної пам'яті образу процесу (рис. 11.12), параметри будуть розміщуватися в наступних змінних: FieldbusCTW-QW0; FieldbusREF-QW2; StatusWord-IW0; Frequency-IW2; MotorCurrent-IW4; MotorVoltage-IW6; VLTAlarmWord-IW8; VLTWarningWord-IW10.

Отже, задану частоту або швидкість (залежить від налаштування частотного перетворювача) необхідно записувати в QW2, управляти станом PDS через зміну QW0, контролювати стан — IW0.

Для управління машиною станів створимо функцію FC200 на мові SCL з назвою PDS_CTRL (рис. 11.14). Дану функцію можна викликати в блоці OB1 (рис. 11.13).

У функції PDS_CTRL слід звернути увагу на те, що байти в 16-бітному слові в ПЛК типу S7 розміщуються таким чином: молодший байт слова — зі старшим номером, старший байт — з молодшим номером. Тобто, для IW0 старший байт буде в IB0, молодший — в IB1. Тому при використанні змінних статусу та команди, як набору бітів tmpSTATUS_AR та tmpCOMMAND_AR, необхідно врахувати іншу нумерацію бітів: біти 0–7 містяться в масиві з індексами 8–15, 8–15 — з індексами 0–7.

Properties - DP slave

General Parameter Assignment

Parameters

- Auto Configuration
- PNU in P915/0
- PNU in P915/1
- PNU in P915/2
- PNU in P915/3
- PNU in P915/4
- PNU in P915/5

PROFIBUS DP

Additional Field Devices

Drives

- Danfoss
 - FC100/200/300
 - FC100/200/300
 - Universal module
 - Prodrive standard telegram 1
 - PPD Type 1 Module consistent PCD
 - PPD Type 1 Word consistent PCD
 - PPD Type 2 Module consistent PCD
 - PPD Type 2 Word consistent PCD
 - PPD Type 3 Module consistent PCD
 - PPD Type 3 Word consistent PCD
 - PPD Type 4 Module consistent PCD
 - PPD Type 4 Word consistent PCD
 - PPD Type 5 Module consistent PCD
 - PPD Type 5 Word consistent PCD
 - PPD Type 6 Module consistent PCD
 - PPD Type 6 Word consistent PCD
 - PPD Type 7 Module consistent PCD

Value

- Enable Autoconfig
- Fieldbus CTW
- Fieldbus REF
- None
- None
- None
- None
- None
- None
- None
- None
- Status Word
- Frequency
- Motor Current
- Motor Voltage
- WLT Alarm Word
- WLT Warning Word
- None

Cancel Help

Slot	DP ID	Order Number / Designation	I Address	Q Address
0	117	PPD Type 4 Word consistent PCD	0..11	0..11
1				

Рис. 11.12. Створення конфігурації для приводу FC 100/200/300

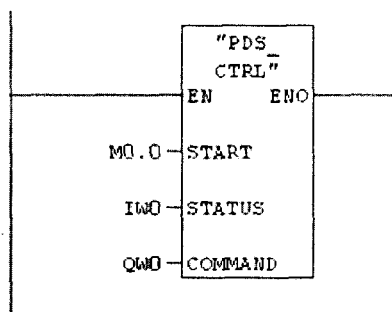


Рис. 11.13. Лістинг програми з викликом функції PDS_CTRL

```

FUNCTION FC200: VOID

VAR_TEMP
// тимчасові змінні
//для оголошення слова статусу як набору бітів
tmpSTATUS:WORD;
// статус як набір бітів
tmpSTAUS_AR AT tmpSTATUS:ARRAY[0..15] OF BOOL;
//для оголошення слова команди як набору бітів
tmpCOMMAND:WORD;
// слово команди як набір бітів
tmpCOMMAND_AR AT tmpCOMMAND:ARRAY[0..15] OF BOOL;
END_VAR
VAR_IN_OUT
COMMAND:WORD; //слово команди
END_VAR
VAR_INPUT
START : BOOL:=false; //запуск/зупинка двигуна
STATUS:WORD; //слово статусу
END_VAR;
tmpSTATUS:=STATUS;
//значення ключових бітів слова команди залишаємо,
//та виставляємо біти для режиму operation
//4-Enable Ramp Generator;5-Unfreeze Ramp Generator;
//6-Enable Setpoint;10-Control BY PLC
tmpCOMMAND:=2#0000_0100_0111_0000 OR COMMAND;
IF START THEN//якщо ПУСК
//якщо стан Switching On Inhibited
IF tmpSTAUS_AR[14]//6 bit - Switching On Inhibited
AND NOT tmpSTAUS_AR[8]//0-Not Ready To Switch On
AND NOT tmpSTAUS_AR[9] //1-Not Ready To Operate
AND NOT tmpSTAUS_AR[10] //2-Operation Disabled
THEN
tmpCOMMAND_AR[8]:=false;//0-OFF1
tmpCOMMAND_AR[9]:=true;//1-No Coast Stop
tmpCOMMAND_AR[10]:=true;//2-No Quick Stop
END_IF;
//якщо стан Ready for Switching On
IF tmpSTAUS_AR[8]//0 - Ready To Switch On
AND NOT tmpSTAUS_AR[9]//1-Not Ready To Operate
AND NOT tmpSTAUS_AR[10]//2-Operation Disabled
AND NOT tmpSTAUS_AR[14]//6-Switching On Not Inhibit
THEN
tmpCOMMAND_AR[8]:=true;//0-ON
END_IF;
//якщо стан Switched On
IF tmpSTAUS_AR[8]//0-Ready To Switch On
AND tmpSTAUS_AR[9] //1-Ready To Operate
AND NOT tmpSTAUS_AR[10]//2-Operation Disabled
AND NOT tmpSTAUS_AR[14]//6-Switching On Not Inhibit
THEN
tmpCOMMAND_AR[11]:=true;//3-Enable Operation
END_IF;
ELSE //якщо STOP
tmpCOMMAND_AR[8]:=false;//0-OFF
tmpCOMMAND_AR[9]:=false;//1-Coast Stop
tmpCOMMAND_AR[10]:=false;//2-Quick Stop
END_IF;
COMMAND:=tmpCOMMAND;
END_FUNCTION

```

Рис. 11.14. Лістинг функції PDS_CTRL

11.4. Профіль CiA402

Профіль *CiA402* розроблений організацією CAN in Automation як профіль пристрою *DS-402*. Він використовується як базовий прикладний профіль управління PDS у мережах CANOpen, ETHERNET Powerlink та ETHERCat. Нижче розглянемо функціонування CiA402 тільки стосовно мережі CANOpen. Слід зазначити, що використання частотних перетворювачів у мережі CANOpen не обов'язково передбачає послугоування профілем CiA402 (DS-402).

11.4.1. Функціональні елементи CiA402

11.4.1.1. Слова управління та статусу (Controlword та Statusword). Автомати станів функціональних елементів управляються за допомогою керуючих слів Command (Controlword Індекс у Словнику 6040₁₆), через які передається команда, а контролюються через слова статусу Status (Statusword — Індекс в Словнику 6041₁₆).

Дані входів/виходів включають один або декілька сигналів уставок та дійсних значень для управління і контролю за приводом. Ці дані та параметри прикладних профілів зберігаються у конкретних Об'єктах Словника. Так, наприклад, для режиму управління швидкістю без зворотного зв'язку (див. рис. 11.6, (б)) визначені такі Об'єкти Словника:

- Command — *Controlword* (ID 6040₁₆);
- Status — *Statusword* (ID 6041₁₆);
- Velocity set point — *Target velocity* (ID 6042₁₆);
- Actual velocity — *velocity actual value* (ID 6044₁₆).

11.4.1.2. Device Control FE. Автомат станів може перебувати в режимі *No Fault* (3-й біт Statusword = 1) або *No Fault* (3-й біт Statusword = 0). Скидання аварії проводять командою *Fault Reset* (7-й біт Controlword = 1). Поведінка приводу в кожному стані визначається в самому пристрої, наприклад при конфігуруванні. Після скидання аварії привод переводиться в стан *Ready To Switch On*.

11.4.1.3. Communication FE. Для CiA402 визначені такі комунікаційні стани: *Operational* та *Pre-operational* або *Stopped*. Робота автомата станів Комунікаційного Функціонального Елемента залежить від промислової мережі, на якій працює привод з даним профілем. Для CANOpen діаграма станів NMT-веденого наведена в 9-му розділі.

11.4.1.4. Basic Drive FE. Автомат станів профілю CiA402 наведений на рис. 11.15. Стан not operating, який визначений стандартом IEC 61800-7-1, у профілі CiA402 складається з декількох підстанів.

Автомату доступні такі стани:

– *Not ready to Switch On* — стадія ініціалізації приводу, комунікації недоступні;

– **Switch On Disabled** — завершення стадії ініціалізації, силова частина приводу неактивна, подача живлення на неї неможлива, на двигун не подається живлення, доступна параметризація через комунікації;

– **Ready to Switch ON** — очікування подачі живлення на силову частину приводу, на двигун не подається живлення, доступна параметризація через комунікації;

– **Switched On** — подане живлення на силову секцію приводу, на двигун не подається живлення, доступна параметризація через комунікації;

– **Operation Enabled** — нормальний операційний режим роботи, помилок не виявлено, на двигун подається живлення, доступна параметризація через комунікації за винятком конфігураційних даних силової частини;

– **Fault Reaction Active** — виявлення помилки на приводі та її обробка; функції управління двигуном доступні;

– **Fault** — стан помилки, привод заблокований, на двигун не подається живлення;

– **Quick Stop Active** — стан швидкої зупинки двигуна.

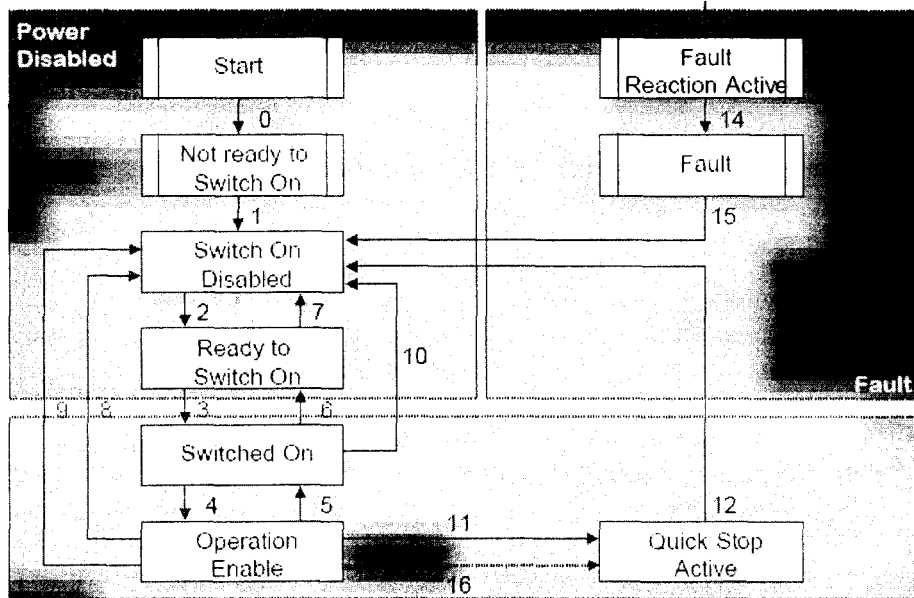


Рис. 11.15. Автомат станів для профілю CiA-402

Стан приводу визначений Statusword, а переходи між станами можуть відбуватися за внутрішніми подіями або за командою, визначеною в Controlword. У таблиці 11.4 наведені значення бітів слова статусу у різних станах.

Таблиця 11.4

ЗНАЧЕННЯ БІТІВ СЛОВА СТАТУСУ

Стан	Біти Statusword (6041 ₁₆)							
	Switch On Disabled	Quick Stop Active	Voltage Enabled	Fault	Operation Enabled	Switched On	Ready to Switch ON	Значення по масці 006F*
	6	5	4	3	2	1	0	
Not Ready To Switch ON	0	x	x	0	0	0	0	—
Switch On Disabled	1	x	x	0	0	0	0	0040
Ready to Switch ON	0	1	x	0	0	0	1	0021
Switched On	0	1	1	0	0	1	1	0023
Operation Enabled	0	1	1	0	1	1	1	0027
Quick Stop Active	0	0	1	0	1	1	1	0007
Fault Reaction Active	0	x	x	1	1	1	1	—
Fault	0 0	0 1	x x	1 1	0 0	0 0	0 0	0008 0028

x — невизначений стан

* — значення в шістнадцятковому форматі, наведене по масці 006F

У таблиці 11.5 наведені команди для переходу між станами та біти слова Controlword, які їм відповідають.

Таблиця 11.5

КОМАНДИ ДЛЯ ПЕРЕХОДУ МІЖ СТАНАМИ

Команда	Біти слова Controlword (6040 ₁₆)					
	Fault Reset	Enable Operation	Quick Stop	Enable Voltage	Switch On	знач в 16-ковому форматі
	7	3	2	1	0	
Shutdown	x	x	1	1	0	0006
Switch On	x	x	1	1	1	0007
Enable Operation	x	1	1	1	1	000F
Disable Operation	x	0	1	1	1	0007
Disable Voltage	x	x	x	0	x	0000
Quick Stop	x	x	0	1	x	0002
Fault Reset	0 → 1*	x	x	x	x	0080

* — за переднім фронтом сигналу

У таблиці 11.6 наведені події та команди, за якими відбувається перехід між станами. Номер переходу показаний на рис. 11.5.

Таблиця 11.6

**ПОДІЇ ТА КОМАНДИ, ЗА ЯКИМИ ВІДБУВАЄТЬСЯ
ПЕРЕХІД МІЖ СТАНАМИ**

№ пере- ходу	Зі стану	На стан	Подія	Реакція на подію
0	Start	Not Ready To Switch ON	Запуск приводу	Привод проходить самодіагностику та стартову ініціалізацію
1	Not Ready To Switch On	Switch On Disabled	позитивний результат ініціалізації	Активізація комунікації та відображення процесних даних
2	Switch On Disabled	Ready to Switch ON	команда Shutdown	перехід у наступний стан
3	Ready to Switch On	Switched On	команда Switch On	силова секція приводу включається
4	Switched On	Operation Enabled	команда Enable Operation	перехід в операційний режим: доступні функції управління двигуном
5	Operation Enabled	Switched On	команда Disable Operation	операційний режим відключається
6	Switched On	Ready to Switch ON	команда Shutdown	силова секція приводу відключається
7	Ready to Switch ON	Switch On Disabled	Команда Quick Stop	перехід в наступний стан
8	Operation Enabled	Ready to Switch ON	команда Shutdown	негайне відключення силової секції та напруги з двигуна
9	Operation Enabled	Switch On Disabled	команда Disable Voltage	негайне відключення силової секції та напруги з двигуна
10	Switched On	Switch On Disabled	команда Disable Voltage або Quick Stop	негайне відключення силової секції та напруги з двигуна
11	Operation Enabled	Quick Stop Active	команда Quick Stop	виконується швидка зупинка двигуна
12	Quick Stop Active	Switch On Disabled	швидка зупинка виконана або команда Disable Voltage	силова секція приводу відключається
13	Всі стани	Fault Reaction Active	помилка на приводі	виконання дії відповідно до типу помилки
14	Fault Reaction Active	Fault	обробка помилки закінчена	функції управління двигуном недоступні, силова секція може відключитися
15	Fault	Switch On Disabled	команда Fault Reset	зняття помилкового стану у випадку відсутності помилок, біт команди Fault Reset повинен обнулитись з контролера
16	Quick Stop Active	Operation Enabled	команда Enable Operation	активується функції управління двигуном

11.4.2. Операційна робота приводів CiA-402 в режимі управління швидкістю

У CiA-402 доступні всі прикладні режими, описані в IEC 61800-7-1. Режим управління швидкістю без зворотного зв'язку має назву **Profile Velocity mode**. Для цього режиму в операційному режимі запуск двигуна відбувається за допомогою команди **No Halt** (8-й біт Controlword = 0), зупинка — **Halt** (8-й біт Controlword = 1). Задана швидкість обертання задається об'єктом **Target velocity** (6042_{16}), а контролюється об'єктом — **Velocity actual value** (6044_{16}). Контроль за роботою двигуна в операційному режимі проводиться також додатковими бітами статусу, які є опціональними:

– **Target Reached** (біт 10) — при Halt = 1, 0 означає, що двигун зупиняється, 1 — двигун зупинився; при Halt = 0, 0 означає, що швидкість не досягнула уставки, 1 — швидкість досягнула уставки;

– **Speed** (біт 12) — якщо 0 — швидкість не дорівнює нулю, 1 — швидкість дорівнює нулю.

Загальний список бітів слова команди для режиму управління швидкістю наведений у таблиці 11.7

Таблиця 11.7

БІТИ СЛОВА КОМАНДИ ДЛЯ РЕЖИМУ УПРАВЛІННЯ ШВИДКІСТЮ

Назва Command IEC 61800-7	Назва CiA-402	Біт Controlword	Пояснення
	Switch ON	0	=1, включення силової секції приводу
	Enable Voltage	1	=0, негайне відключення силової секції та напруги з двигуна
	Quick Stop	2	=1, швидка зупинка двигуна
OPERATE	Enable Operation	3	=1, переведення в операційний режим
		4	зарезервовані
		5	зарезервовані
		6	зарезервовані
RESET FAULT	Fault Reset	7	скидання помилок
	Halt	8	зупинка двигуна
		11-15	призначення залежить від виробника

Загальний список бітів слова статусу для режиму управління швидкістю наведені в таблиці 11.8.

Приклад 11.2. CiA 402. Написання програми для роботи ПЛК TSX Premium з ЧПР ATV71

Завдання. Розробити конфігурацію та програму користувача в пакеті PL7 для управління (задана частота) та контролю (плинна частота, струм двигуна, напруга двигуна, аварії, попередження) частотним перетворювачем ATV 71 відповідно до завдання з прикладу 9.7.

БІТИ СЛОВА СТАТУСУ ДЛЯ РЕЖИМУ УПРАВЛІННЯ ШВИДКІСТЮ

Назва STATUS IEC 61800-7	Назва CiA-402	Bit Statusword	Пояснення
	Ready to Switch ON	0	=1, готовий до подачі живлення на силову частину приводу
	Switched On	1	=1, силова частина приводу підключена до живлення
Operating	Operation Enabled	2	=1, перебуває в операційному режимі
Faulted	Fault	3	=1, є активна помилка
	Voltage Enabled	4	=1, активність обробки команди Disable Voltage
	Quick Stop Active	5	=0, відбувається швидка зупинка двигуна
	Switch On Disabled	6	=1, не готовий до подачі живлення на силову частину приводу
Warning	Warning	7	=1, є попереджувальні повідомлення
		8	призначення залежить від виробника
Remote control	Remote	9	=1, доступне управління з CANOpen, =0, управління з мережі недоступне
	Target Reached	10	залежить від HALT: двигун зупиняється/зупинився або швидкість рівна/нерівна уставки
	Speed	12	=1, швидкість дорівнює нулю =0, швидкість не дорівнює нулю

Рішення. Частотні перетворювачі Altivar (Schneider Electric) підтримують профіль CiA-402. Для управління ЧПР створимо функціональний блок на мові ST з назвою PDS_CTRL (рис. 11.18). Вхідними параметрами блока є Status1 — куди передається слово статусу, Start1 — команда на запуск двигуна, Command1 — слово команди (вхід/вихід).

На рис. 11.17 показаний автомат станів для даних частотних перетворювачів Altivar, який не відрізняється від стандартного, що показаний на рис. 11.15. Відмінність полягає в позначенні Statusword як ETA, а Commandword як CMD.

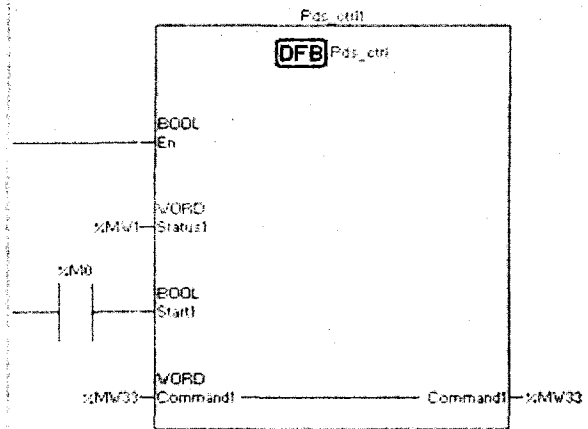


Рис. 11.16. Використання функціонального

автомата станів для даних частотних перетворювачів Altivar, який не відрізняється від стандартного, що показаний на рис. 11.15. Відмінність полягає в позначенні Statusword як ETA, а Commandword як CMD.

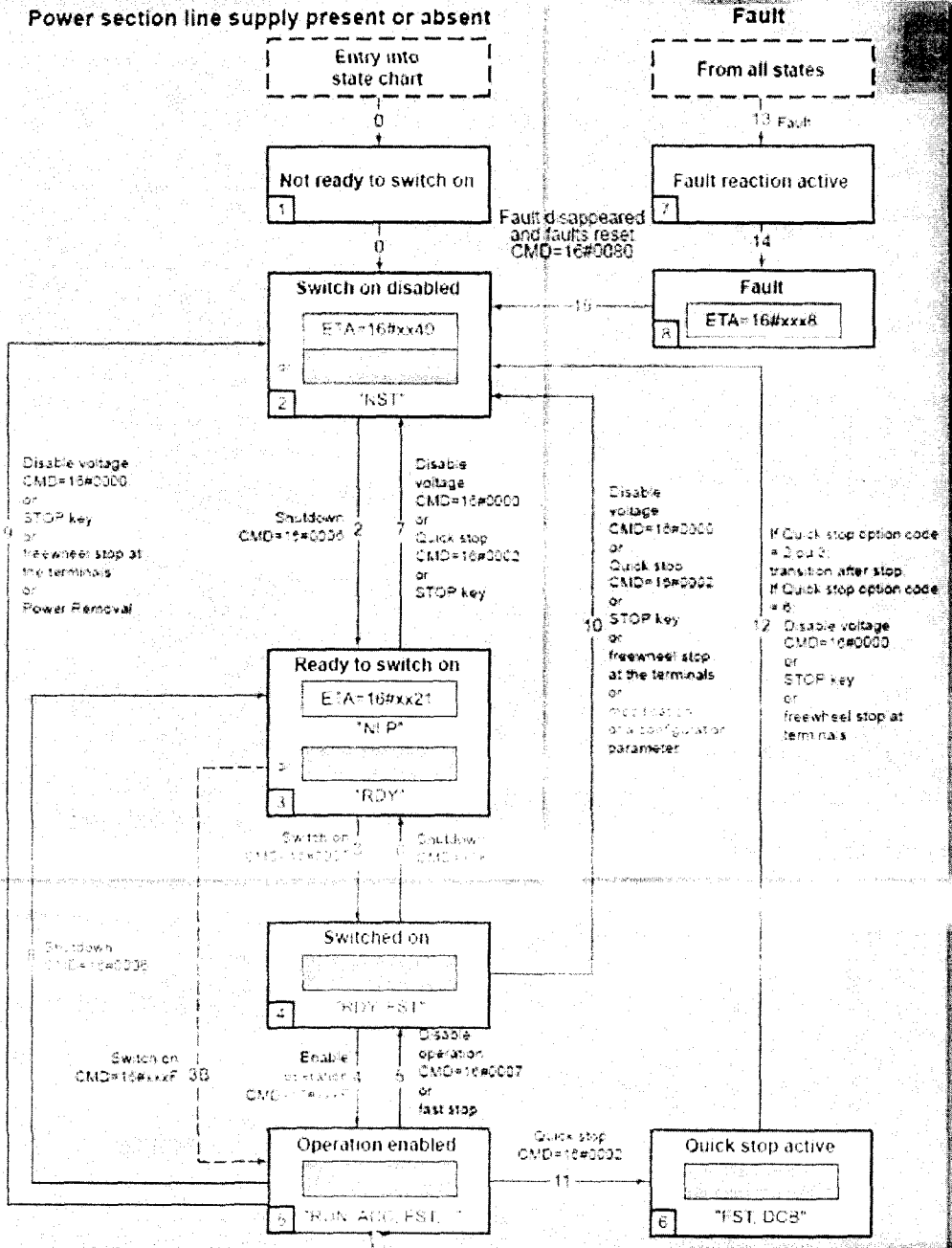


Рис. 11.17. Автомат станів для профілю CiA-402 в ЧПР Altivar

Програма управління частотним перетворювачем з використанням Status1 зображена на рис. 11.16. Згідно з прикладом 9.7, ControlWord міститься в %MW33, StatusWord — в %MW1, задана швидкість обертання TargetVelocity — в %MW34, дійсна ControlEffort — в %MW2. Відповідно інші параметри для контролю містяться в змінних %MW3-%MW7. Змінна %M0 буде запускати/зупиняти двигун.

```

❑ ST : DFB - Pds_ctrl
! Eta_wrd:=Status1 AND 16#00FF;
  Cmd_wrd:= Command1;

  (*switch on disabled*)
  IF (Eta_wrd=16#40 OR Eta_wrd=16#50) AND Start1 THEN
    Cmd_wrd:=16#0006; (*Shutdown*)
  END_IF;

  (*ready to switch on*)
  IF (Eta_wrd=16#21 OR Eta_wrd=16#31) AND Start1 THEN
    Cmd_wrd:=16#0007; (*switch on*)
  END_IF;

  (*switched on*)
  IF Eta_wrd=16#33 AND Start1 THEN
    Cmd_wrd:=Cmd_wrd OR 16#000F; (*enable operation*)
  END_IF;

  (*operation enabled*)
  IF Eta_wrd=16#37 THEN
    IF Start1 THEN
      Cmd_wrd:=Cmd_wrd AND 16#FEFF; (*no_halt*)
    ELSE
      Cmd_wrd:=16#0006; (*Shutdown*)
    END_IF;
  END_IF;
Command1:=Cmd_wrd;

```

Рис. 11.18. Лістинг програми функціонального блока PDS_CTRL у середовищі PL7

Приклад 11.2



Контрольні запитання до розділу 11

1. Навіщо пристрої PDS необхідно підключати до контролерів? Які типи сигналів використовуються при підключенні?
2. В чому переваги використання мережного зв'язку перед аналоговими, дискретними та імпульсними підключеннями контролера до PDS?
3. Які проблеми виникають при використанні мережного зв'язку з PDS різних виробників? Як ці проблеми вирішуються на рівні стандартизації зв'язку з PDS?

4. Які основні складові логічної моделі системи з PDS? Які функції виконують ці складові? Що визначає Універсальний Інтерфейс у контексті стандарту MEK 61800-7?
5. Які функціональні можливості може підтримувати Логічний Привід? Як відрізняється фізична структура реалізації Логічного Приводу, залежно від наявних функцій в PDS? В якій складовій системі можуть бути реалізовані функції, відсутні в PDS?
6. Поясніть відмінність між Універсальним Інтерфейсом MEK 61800-7 та промисловою мережею.
7. Які інтерфейси можуть бути доступні для з'єднання з PDS? Поясніть їх призначення. Які з них підтримуються функціями Універсального Інтерфейсу MEK 61800-7?
8. В яких логічних складових закладені параметри та алгоритми поведінки Логічного Приводу?
9. Перелічіть Функціональні Елементи Логічного Приводу. Які функції в них закладені?
10. Які Функціональні Елементи використовуються при конфігуруванні PDS з боку програматорів, а які для управління Логічним Контролером?
11. Розкажіть про принципи функціонування процесу управління Логічним Приводом. За допомогою яких змінних відбувається процес управління та контролю за Логічним Приводом?
12. Які стани визначені для Функціонального Елемента Device Control та як відбувається перехід між ними? Як контролюється та управляється автомат станів Device Control FE?
13. Які стани визначені для Communication FE та як відбувається перехід між ними? Як контролюється та управляється автомат станів Communication FE?
14. Які стани визначені для Basic Drive FE та як відбувається перехід між ними? Як контролюється та управляється автомат станів?
15. В яких прикладних режимах може функціонувати Логічний Привід?
16. Прокоментуйте схему функціонування режимів управління моментом приводу.
17. Прокоментуйте схему функціонування режимів управління швидкістю приводу.
18. Прокоментуйте схему функціонування режимів управління позиціонуванням приводу.
19. Розкажіть про наявні профілі пристроїв PDS та їх відображення на промисловій мережі в стандартах MEK 61800-7. Який популярний профіль PDS не увійшов до переліку стандартних?
20. Поясніть відмінність поняття профілю PDS від промислової мережі, що підтримуються даним пристроєм. Чи визначає факт використання конкретної промислової мережі наявність конкретного профілю PDS та навпаки?
21. Прокоментуйте функціонування профілю PROFIDRIVE в контексті Функціональних Елементів. Які основні відмінності даної реалізації від стандартної MEK 61800-7-1?
22. Як реалізуються в PROFIDRIVE змінні COMMAND та STATUS, визначені в MEK 61800-7-1? З якими мережами використовується даний профіль?
23. Прокоментуйте схему функціонування автомату станів базових функціональних елементів для профілю PROFIDRIVE.
24. Прокоментуйте схему функціонування операційної роботи приводу в режимі управління швидкістю для профілю PROFIDRIVE.
25. Прокоментуйте функціонування профілю CiA402 в контексті функціональних елементів. Які основні відмінності даної реалізації від стандартної MEK 61800-7-1? Яка назва профілю CiA402 в CANOpen?
26. Як реалізуються в CiA402 змінні COMMAND та STATUS, визначені в MEK 61800-7-1? З якими мережами використовується даний профіль?
27. Прокоментуйте схему функціонування автомату станів базових функціональних елементів для профілю CiA402.
28. Прокоментуйте схему функціонування операційної роботи приводу в режимі управління швидкістю для профілю CiA402.

ВІДКРИТІ ТЕХНОЛОГІЇ ПРОГРАМНОЇ ІНТЕГРАЦІЇ В СЕРЕДОВИЩІ WINDOWS

12.1. Проблеми програмної інтеграції в інтегрованих автоматизованих системах

У першому розділі показана необхідність інтеграції між різними програмними засобами рівнів АСУТП та АСУП. Враховуючи визначені стандарти промислових мереж та незалежність сучасних програмних продуктів від апаратної платформи комп'ютерів, проблема апаратної сумісності між рівнями АСУТП та АСУП, як правило, не розглядається.

Однак, зовсім інша ситуація склалася на ринку програмних засобів. Нагадаємо, програмне забезпечення інтегрованих автоматизованих систем управління на рівні АСУП можна умовно віднести до однієї з таких груп:

- універсальні та спеціалізовані СУБД;
- системи рівня MES;
- системи рівня ERP;
- офісне програмне забезпечення.

До програмних засобів на рівні АСУТП, що потребують інтеграції з іншими ПЗ, можна віднести SCADA/HMI та спеціалізовані СУБД реального часу. Програмна інтеграція має забезпечувати взаємозв'язок між програмами для їх узгодженої роботи, що призводить до необхідності створення інформаційного каналу між даними в цих програмах.

На ринку програмних продуктів для автоматизованих систем не має єдиного стандарту для вирішення даної задачі. Великі компанії-розробники спеціалізованого ПЗ для промислової автоматизації впроваджують програмні пакети для різного рівня автоматизації, які порівняно легко інтегруються між собою. Крім високої вартості даних рішень, вони теж не розв'язують усі необхідні задачі. Через це необхідна інтеграція різних програмних продуктів, розробники яких можуть навіть не знати про існування один одного. Таким чином, крім проблем інформаційної та функціональної інтеграції виникає проблема програмної сумісності.

У розділах 12–14 розглядаються сучасні відкриті технології інтеграції між програмними засобами, які забезпечують доступ до даних іншої програми.

12.2. Функціонування ОС Windows

12.2.1. Основи функціонування

12.2.1.1. Багатозадачний режим. Для розуміння технологій взаємодії між прикладними програмами, необхідні базові поняття про їх функціонування в контексті конкретної операційної системи. Сьогодні найбільш популярними є операційні системи Windows, тому коротко розглянемо принципи роботи програм на цих платформах.

Операційні системи Windows функціонують у багатозадачному режимі, тобто в їх середовищі одночасно виконуються декілька прикладних програм. Однак, це не значить, що на одному процесорі паралельно виконуються машинні команди цих програм. Операційна система створює такі умови для кожної програми, щоб надати їй усі ресурси комп'ютера у виділений момент часу.

12.2.1.2. Потіки. Програмний код, якому необхідно «виділити процесор» в операційній системі Windows, називається *Потоком (Thread)*. Щоб одночасно функціонувало декілька Потоків, операційна система виділяє кожному із них певний час, який називається *квантом*. Тривалість кванта залежить від конкретної реалізації. Таким чином, уся система працює за принципом каруселі: операційна система почергово, на один квант часу відправляє процесору кожний Потік для його виконання. Після обробки останнього Потіку система знову запускає перший (рис. 12.1).

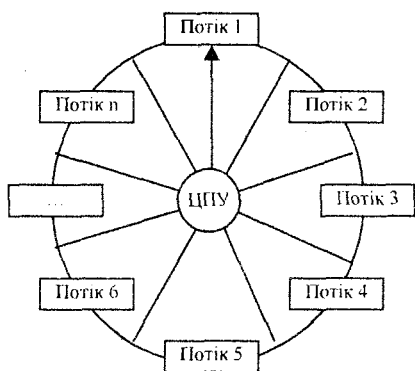


Рис. 12.1. Принцип роботи процесора в ОС Windows

Перед передачею процесору іншого Потіку система зберігає для попереднього Потіку стан усіх регістрів процесора. Ця інформація називається *контекстом Потіку*. Повертаючись до цього Потіку знову, система завантажує його контекст у процесор, для того щоб той продовжив виконувати перерваний минулого разу Потік.

Описана вище схема функціонування працює для Потоків з однаковими *пріоритетами*. Кожний Потік має свій пріоритет, який надається йому системою. Чим він вищий тим важливіший Потік, тому операційна система буде обробляти всі Потіки з найвищим пріоритетом до тих пір, поки всі вони не зупиняться (перейдуть у режим очікування).

Коли не буде жодного робочого Потіку з найвищим пріоритетом — система перейде на Потіки з нижчим пріоритетом і так далі. Цей принцип розподілу часу між різнопріоритетними Потоками називається *багатозадачністю з витісненням*.

Прикладна програма часто складається з одного Потіку, однак деякі з них запускають декілька Потоків. Зі сказаного вище може скластися враження, що у випадку існування в системі Потіку з найвищим пріоритетом він не дасть жод-

ного шансу для інших Потоків. Але це не так. Як правило, всі Потоки простоюють, тобто перебувають у режимі очікування якоїсь події (переривання), яка буде їм адресована. Подією може бути клік або рух миші, сигнал від системного таймера й інше.

12.2.1.3. Віртуальна та фізична пам'ять. Кожний Потік потребує ресурсів пам'яті. Для того, щоб прикладні програми (їх Потоки) не ділили єдину оперативну пам'ять між собою, кожному Потoku виділяється спеціально реалізований операційною системою ресурс, який називається *віртуальною пам'яттю*. Віртуальна пам'ять розміщує дані не тільки в оперативній пам'яті, а і на жорсткому диску. Коли необхідно скористатися ресурсами пам'яті, диспетчер пам'яті системи виділяє частину оперативної пам'яті. Якщо потрібна частина пам'яті була виділена під інший Потік (точніше Процес), її попередньо вивантажують та зберігають на диску, щоб за активації останнього відновити його на тому ж місці.

12.2.1.4. Процеси Windows. Усі потоки та їх дані розміщуються в конкретному *Процесі* (Process). З точки зору прикладної програми, Процес — це адресний простір, в якому розміщуються код та дані всіх EXE та DLL модулів. Саме в ньому містяться області пам'яті, які динамічно розподіляються для стеків потоку та інших потреб. При запуску на виконання EXE-програми система створює для неї Процес, у контексті якого запускає первинний Потік. Він може запустити на виконання ще дочірні Потоки, які будуть виконуватися паралельно. Коли всі Потоки знищуються, Процес теж знищується. Процес може запустити на виконання ще одну програму, тобто створити ще один Процес (до речі, так робить Провідник Windows).

12.2.1.5. Об'єкти ядра Windows. Процеси та Потоки є ресурсами операційної системи, які також є її *об'єктами ядра*, якими можна оперувати. Кожний об'єкт ядра має свій унікальний ідентифікатор у системі. Так, об'єкти ядра Процес мають ProcessID (*PID*), який можна побачити під час виклику «Диспетчера задач» (Ctrl+Alt+Del). Однак, у контексті кожного Процесу об'єктами ядра користуються через *дескриптори (handle)*. Таким чином, у кожному Процесі один і той самий об'єкт ядра буде мати однаковий ідентифікатор, але різні дескриптори.

12.2.1.6. Служби Windows. Є спеціальний вид Процесів, які створюють допоміжну функціональність у системі і не залежить від інтерактивних дій користувача. Тобто, є прикладні програми, які запускаються до того, як користувач реєструється в системі. Такі програми називаються сервісами або *Службами (Services)*. Для користувача Windows настройку Служб можна провести за допомогою програми «services», яку необхідно попередньо завантажити з консолі управління Microsoft (mmc.exe) або з панелі управління адміністратора («Адміністрування->Службы»).

12.2.1.7. Вікна та повідомлення Windows. Операційна система Windows реалізує інтерфейс з користувачем через вікна. Прикладна програма може створити для себе вікна відповідно до їх класу. Для ідентифікації вікон, при їх створенні система повертає Потoku *дескриптор вікна* (handle window або *hwnd*). Вона слідкує за подіями, які пов'язані з клавіатурою, мишею, сенсорною панеллю і відправляє *повідомлення (message)* конкретному вікну. Кожне вікно виконується в контексті того Потoku, який його створив. Для того, щоб прийняти повідомлення, які направлені

вікнам Поточку, у нього є *черга повідомлень*. Потік витягує повідомлення з черги і направляє його певній віконній процедурі, яка пов'язана з цим вікном. Наприклад, при натисканні кнопки миші система відправить повідомлення «лівий клік миші» Поточку, який створив це вікно (кнопка — це теж вікно). Система, знайшовши повідомлення в черзі повідомлень, викликає відповідну віконну процедуру.

12.2.2. Архітектура Windows

12.2.2.1. Рівні привілеїв процесора. Розглянемо, які складові операційних систем на базі Windows NT (рис. 12.2) з одноядерним мікропроцесором. Мікропроцесори типу Pentium мають чотири *рівні привілеїв* (privilege levels), відомих також як кільця. Ці привілеї визначають, наприклад, можливість доступу до пам'яті, використання деяких критичних команд процесора (такі команди, пов'язані із захистом і ін.). Кожний потік виконується на одному з цих рівнів привілеїв. Кільце 0 —

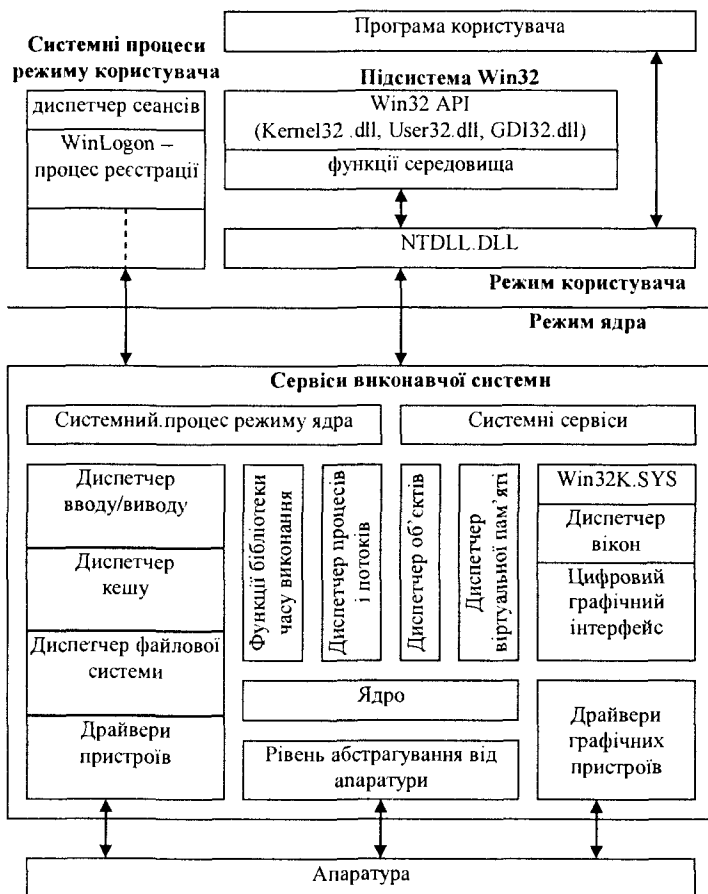


Рис. 12.2. Функціональна структура Windows NT

найбільш привілейований рівень, з повним доступом до всіх команд процесора та всієї його пам'яті. Кільце 3 — найменш привілейований рівень. Для забезпечення сумісності із системами на базі процесорів, відмінних від тих, що випускає Intel, Windows підтримує тільки два рівні привілеїв — кільця 0 та 3. Якщо потік виконується в кільці 0, кажуть, що він виконується в *режимі ядра (kernel mode)*, якщо в 3-му кільці — то в *режимі користувача (user mode)*.

12.2.2.2. Обмеження доступу програми користувача до ресурсів. Код операційної системи на низькому рівні працює в режимі ядра, а програми користувача — як правило, в режимі користувача. Для виконання деяких завдань необхідно перемикання програми користувача в режим ядра. В цьому випадку вона викликає *системну функцію (system function)* використовуючи інтерфейс Windows API (розглянутий нижче). Таким чином код програми користувача безпосередньо не може працювати в режимі ядра, що захищає систему від збоїв. У режимі ядра також працюють драйвери пристроїв.

12.2.2.3. Системні процеси. Процеси, які обслуговують операційну систему, називаються *системними Процесами (system process)*:

- Процес idle: складається з одного Поточку, який управляє часом простою процесора;
- Процес system: спеціальний Процес, який виконується в режимі ядра (всі інші перераховані виконуються в режимі користувача); його Поточки зветься системними Поточками (system threads);
- Процес Session Manager (SMSS.EXE) — диспетчер сеансів;
- підсистема Win32 (CSRSS.EXE);
- Процес реєстрації в системі Win Logon (WINLOGON.EXE).

12.3. Інтерфейс програми користувача до ОС Windows

12.3.1. Win32 API

Для того, щоб програми користувача могли використовувати ресурси операційної системи, вона надає їм певний інтерфейс. Інтерфейс доступу прикладних програм до об'єктів якої-небудь програмної системи називають *API (Application Programming Interface)*. Інтерфейси прикладних програм являють собою набір функцій, які забезпечують сервіси даної програми. У 32-х розрядних операційних системах Windows (x86) програмний інтерфейс має назву *Win32 API*. Хоча функції для всіх Win32-систем в основній масі перекриваються (однакові), реалізація їх різна, оскільки ядра в операційних системах відрізняються. Тому деякі прикладні програми добре функціонують у Windows системах, а деякі ведуть себе специфічно або взагалі не працюють на одній із них.

API-функції Windows входять до складу бібліотек з динамічним підключенням типу *DLL (Dynamic Link Library)*, які є підсистемою середовища Win32. Бібліотеки DLL — це виконавчий файл, який вміщує декілька експортних функцій (exportable functions), тобто функцій, до яких можуть звертатися інші виконавчі прикладні програми (EXE або DLL). Такі бібліотеки можна викликати за необ-

хідності, тобто при виконанні експортованих функцій, які там містяться. В цьому випадку бібліотека проектується на адресний простір Процесу того Поточку, який викликав її функцію. Будь-яка прикладна програма може скористатися експортними функціями dll-бібліотеки, якщо в ній правильно оголошений виклик і правильно передані всі фактичні параметри. Це значить, що при описі функцій бібліотеки він може ними скористатися, просто викликавши їх.

API-функції імпортуються прикладною програмою з декількох бібліотек DLL:

- KERNEL32.DLL — вміщує близько 700 функцій, які призначені для управління пам'яттю, Процесами та Поточками;
- USER32.DLL — надає близько 600 функцій для управління інтерфейсом користувача, наприклад, створенням вікон та передачею повідомлень;
- GDI.DLL — екпортує близько 400 функцій для малювання графічних образів, відображення тексту та роботи зі шрифтами;
- COMDLG32.DLL — відкриває доступ до майже 20 функцій управління стандартними діалоговими вікнами Windows;
- LZ32.DLL — зберігає близько 10 функцій архівування і розархівування файлів;
- ADVAPI32.DLL — екпортує близько 400 функцій, пов'язаних із захистом об'єктів та роботи з реєстром;
- WINM.DLL — вміщує близько 200 функцій, які належать до мультимедіа.

Приклад 12.1. Технології програмної інтеграції. Функція для створення Процесу з відповідним об'єктом ядра «Процес».

Завдання. Створити функцію для створення Процесу з використанням VB/VBA.

Рішення. Використаємо API-функцію CreateProcess, більшу частину аргументів яких треба просто проініціалізувати за замовчуванням.

```
Public Declare Function CreateProcess Lib "kernel32" Alias
"CreateProcessA" _
    (ByVal lpApplicationName As String, _
    ByVal lpCommandLine As String, _
    lpProcessAttributes As SECURITY_ATTRIBUTES, _
    lpThreadAttributes As SECURITY_ATTRIBUTES, _
    ByVal bInheritHandles As Long, _
    ByVal dwCreationFlags As Long, _
    lpEnvironment As Any, _
    ByVal lpCurrentDirectory As String, _
    lpStartupInfo As STARTUPINFO, _
    lpProcessInformation As PROCESS_INFORMATION) _
    As Long
```

Де: lpApplicationName — ім'я файла. Якщо файл перебуває в поточній директорії краще задавати ім'я в наступному полі, а в цьому вказувати vbNullString; lpCommandLine — ім'я файла в форматі повного командного рядка. Якщо рядок не вказаний, то він шукає в локальному, системному, основ-

ному каталогах Windows. Використовується, якщо lpApplicationName= vbNullString; lpProcessAttributes — атрибути захисту для дочірнього процесу. Як правило, треба вказати змінну, яку попередньо оголосити з типом SECURITY_ATTRIBUTES; lpThreadAttributes — атрибути захисту для дочірнього потоку. Як правило, треба вказати змінну, яку попередньо об'явити з типом SECURITY_ATTRIBUTES; bInheritHandles — наслідувати чи ні дескриптори. Вкажіть FALSE; dwCreationFlags — визначає прапорці, які впливають на те, як саме створюється новий процес. lpEnvironment — область пам'яті з окружними змінними (Передавайте 0); lpCurrentDirectory — поточний диск і каталог створеному процесу. Як правило, передається 0; lpStartupInfo — параметри, які передаються створюваному процесу. Навіть якщо треба створити процес із параметрами за замовчуванням, необхідно проініціалізувати цю структуру і в ній вказати розмір (cb); lpProcessInformation — через цю структурну змінну функція повертає значення дескрипторів та ідентифікаторів процесу, що створюється, та його первинного потоку.

Декілька прапорців об'єднуються через OR. Серед корисних можна виділити ті, які задають базовий пріоритет процесу: IDLE_PRIORITY_CLASS (&H40); NORMAL_PRIORITY_CLASS (&H20); HIGH_PRIORITY_CLASS (&H80); REALTIME_PRIORITY_CLASS (&H100).

Отже, приклад на VB буде мати такий вигляд.

Деклараційна частина:

```
Public Type SECURITY_ATTRIBUTES ' атрибути захисту об'єктів ядра
    nLength As Long
    lpSecurityDescriptor As Long
    bInheritHandle As Long
End Type
Public Type PROCESS_INFORMATION
    hProcess As Long 'дескриптор процесу
    hThread As Long 'дескриптор первинного потоку
    dwProcessId As Long 'ідентифікатор процесу
    dwThreadId As Long 'ідентифікатор потоку
End Type
Public Type STARTUPINFO '
    cb As Long 'розмір даної структури (для контролю версій)
    lpReserved As String
    lpDesktop As String
    lpTitle As String
    dwX As Long
    dwY As Long
    dwXSize As Long
    dwYSize As Long
    dwXCountChars As Long
    dwYCountChars As Long
    dwFillAttribute As Long
    dwFlags As Long
    wShowWindow As Integer
    cbReserved2 As Integer
```

```

lpReserved2 As Long
hStdInput As Long
hStdOutput As Long
hStdError As Long
End Type
Public Declare Function CreateProcess Lib "kernel32" Alias
"CreateProcessA" _
    (ByVal lpApplicationName As String, _
    ByVal lpCommandLine As String, _
    lpProcessAttributes As SECURITY_ATTRIBUTES, _
    lpThreadAttributes As SECURITY_ATTRIBUTES, _
    ByVal bInheritHandles As Long, _
    ByVal dwCreationFlags As Long, _
    lpEnvironment As Any, _
    ByVal lpCurrentDirectory As String, _
    lpStartupInfo As STARTUPINFO, _
    lpProcessInformation As PROCESS_INFORMATION) _
    As Long

```

Виконавча частина:

```

Sub CreateProc(ByVal file1 As String)
    Dim StartInfo As STARTUPINFO, Pi As PROCESS_INFORMATION
    Dim thrSA As SECURITY_ATTRIBUTES
    Dim procSA As SECURITY_ATTRIBUTES
    With StartInfo
        .cb = LenB(StartInfo)
    End With
    CreateProcess (vbNullString, file1, procSA, thrSA, False, _
    0, 0, vbNullString, StartInfo, Pi)
End Sub

```

Приклад 12.1

12.3.2. Проблеми доступу до даних іншого Процесу

Кожний із Процесів в операційній системі має свій адресний простір, тому передача даних між двома різними Процесами — завдання не тривіальне і вирішується декількома шляхами. Спеціалістам з автоматизації більш цікаві стандартні підходи, які не потребують написання програм (принаймні, складних), а тільки конфігурування для кожного з Процесів. Наприклад, для зв'язку таблиці Excel з потрібними даними в базі даних Access, з подальшим автоматичним відновленням інформації, можна скористатися ресурсами самого Excel. Для того, щоб гнучко використовувати такі вбудовані в програмні засоби можливості, необхідне розуміння внутрішньої сутності функціонування. У зв'язку з цим коротко розглянемо базові поняття функціонування відкритих технологій між програмної (міжпроцесної) взаємодії, а саме DDE/NetDDE, COM/DCOM та WEB-технології.

12.4. Технології DDE та NetDDE

12.4.1. Локальний обмін через DDE

12.4.1.1. Походження. *DDE* (Dynamic Data Exchange — динамічний обмін даними) — технологія, яка базується на зв'язку між прикладними програмами з використанням передачі віконних повідомлень (WM - Windows Message).

Перші прикладні програми, які користувалися цією технологією, безпосередньо передавали, отримували і обробляли віконні повідомлення спеціального типу, виділені для DDE. Пізніше Microsoft розробила для цього спеціальний набір функцій бібліотеки динамічного обміну даними (*DDEML* – DDE Management Library), які підтримували сумісність із старими прикладними програмами.

12.4.1.2. Взаємодія між Процесами. Коли дві прикладні програми обмінюються даними за допомогою DDE, то кажуть, що вони зайняті в *DDE сеансі зв'язку* (DDE conversation). Ту програму, яка починає сеанс зв'язку, називають *Клієнт DDE* (DDE client application), яка відповідає — *Сервер DDE* (DDE server application).

Сеанс зв'язку проходить між двома вікнами, по одному для кожного з програм-учасників. Вікно може бути сховане, наприклад у випадку, коли воно створюється тільки для обміну по DDE. Прикладна програма може приймати участь у декількох сеансах зв'язку у той самий час як в якості Клієнта, так і в якості Серверу. Однак, для кожного обміну використовується тільки одне вікно як для Клієнта, так і для Серверу (наприклад, 10 сеансів — це 10 клієнтських вікон + 10 серверних вікон на дві прикладні програми).

12.4.1.3. Модель ідентифікації даних. Для ідентифікації модуля даних, до яких доступається Клієнт на Сервері, використовується символічне ім'я, яке складається з трьох частин:

- назва прикладної програми (*Application*);
- назва документа або розділу (*Topic*);
- назва елемента даних (*Item*).

На початку сеансу зв'язку DDE Клієнт і Сервер визначають назву прикладної програми і документа. Наприклад, якщо в якості програми-Серверу виступає Microsoft Excel, то назва прикладної програми буде «Excel». Документ в такому випадку може складатися з назви файла та листа, а якщо файл Excel уже завантажений, то просто з назви листа, наприклад «Лист2». Слід зауважити, що за допомогою цих двох частин назви встановлюється сеанс зв'язку, тобто визначаються вікна Клієнта та Серверу. Це значить, що протягом цього сеансу наведені назви змінитися не можуть.

Елемент даних DDE — це інформація, яка пов'язана з документом і бере участь в обміні. Назва цього елемента теж визначається залежно від прикладної програми. Так, наприклад, в Excel елемент даних — це комірка на листі, яка адресується номером рядка та колонки (наприклад «R1C1»). Значення елемента даних може передаватись як від Клієнта до Серверу, так і навпаки. Формат даних може бути будь-яким, який підтримується буфером обміну.

12.4.1.4. Системний розділ (System). Будь-яка прикладна програма-Сервер DDE повинна підтримати розділ (Topic) з назвою «System». Цей розділ забезпечує контекстною інформацією, яка може являти інтерес для програми-Клієнта. У табл. 12.1 перераховані деякі назви елементів даних, які підтримуються даним розділом.

Таблиця 12.1

ПРИЗНАЧЕННЯ ЕЛЕМЕНТІВ ДАНИХ

Item	Призначення
Formats	Список підтримуваних форматів буферу обміну даного DDE-серверу
Help	Коротка допомога з використанням DDE
Status	Інформація про стан прикладної програми
SysItems	Список елементів даних, які підтримуються прикладною програмою для розділу System
Topics	Список розділів (документів), які доступні для даної програми в даний момент часу. Він від часу до часу може змінюватись.

Приклад 12.2. Технології програмної інтеграції. Доступ до комірки Excel з використанням DDE

Завдання. Вказати адресу даних у комірці Excel для доступу за DDE.

Рішення. Щоб доступитися до комірки даних Microsoft Excel у 2 колонці та у 3 рядку на листі «Лист 2», необхідно вказати:

- Application — Excel;
- Topic — Лист2;
- Item — R3C2.

Якщо необхідно вказати ще назву файлу, то це Topic — [Книга1.XLS]Лист2. Часто формат запису поєднує назви програми та документа:

Excel\Лист2 або *Excel\[Книга1.XLS]Лист2*

Іноколи необхідний модуль даних вказується повністю. Наприклад, Excel може бути Клієнтом DDE. Тому якщо на комп'ютері завантажені дві програми Excel, можна обмінюватися даними між їх комірками, вказавши в комірці однієї з них:

= *Excel\Лист2!'R3C2'*

Зверніть увагу, що в Excel деякі назви беруться в апострофи.

Приклад 12.2

12.4.2. Обмін через NetDDE в мережі

12.4.2.1. Основи функціонування. *Network DDE*, або просто *NetDDE*, використовується для обслуговування мережних зв'язків, які потрібні для сеансів зв'язку DDE між прикладними програмами, що виконуються на різних комп'ютерах у мережі.

Для запуску обміну через NetDDE необхідно, щоб у системі комп'ютера був завантажений *агент NetDDE* (NETDDE.EXE). Зверніть увагу, що програма-агент

не запускається автоматично за активації NetDDE, тому її треба запустити як на комп'ютері Клієнта, так і на комп'ютері Серверу DDE. В операційній системі Windows вона запускається у вигляді двох служб: «Служба сетевого DDE» та «Диспетчер сетевого DDE», тому їх треба активувати та запустити. Агент за необхідності запускає прикладну програму, на якій розмішені дані. Весь обмін проходить саме через вікно програми-агента, тому він є проміжним Сервером (проху-server), який замінює сторону дійсного Серверу, перенаправляючи всі повідомлення його вікну.

12.4.2.2. Виділення загальних ресурсів. DDE Shares — загальні ресурси DDE, якими можуть користуватися програми-Клієнти DDE на даному комп'ютері. Тобто, якщо програма-Клієнт хоче доступитися до даних програми-Серверу на іншому комп'ютері, необхідно на останньому прописати ресурси, з якими можна зв'язатися та їх загальну (shared) назву, яка завжди закінчується символом «\$». На комп'ютері програми-Серверу DDE, в базі даних DSDM, ресурси зберігаються у вигляді записів типу «назва загального ресурсу» — «розміщення ресурсу». Таким чином, програма-Клієнт буде проводити сеанс з'єднання через ці ресурси.

Для налаштування загальних ресурсів є спеціальна програма DDESHARE.EXE (вона запускається з командного вікна):

- додається новий ресурс, вказується ім'я із «\$» в кінці (наприклад «SHARA\$»);
- вказується назва програми та документ (розділ) модуля даних DDE;
- настроюються опції: автоматичний пуск програми та ін.

Слід врахувати особливості захисту на рівні операційної системи. Конфігурування загальних ресурсів можна проводити віддалено, вказавши необхідний комп'ютер у конфігураторі.

12.4.2.3. Типи загальних ресурсів. Є три типи загальних ресурсів: за старим стилем, за новим стилем та статичний тип. Вони відрізняються за правилами ідентифікації ресурсів. Як правило, всі прикладні програми використовують статичний тип. Старий стиль призначений для програм, які реалізують DDE сеанс зв'язку за допомогою віконних повідомлень, а новий стиль — на основі бібліотеки DDEML.

З боку Клієнта, при ідентифікації ресурсу DDE вказується:

- Application — \\ім'я_комп'ютера\NDDE\$ (наприклад, \\COMP1\NDDE\$);
- Topic — ім'я_загального_ресурсу (наприклад, SHARA\$);
- Item — назва_елементу_даних (наприклад, R3C2).

12.4.2.4. Область використання DDE/NetDDE. DDE та NetDDE використовується не тільки для обміну між комірками Excel, а й у програмному забезпеченні для АСУТП та АСУП. Зокрема, більшість SCADA-програм надають доступ до своїх даних через DDE та, навпаки, можуть бути Клієнтами DDE. Сьогодні технологія DDE все більше витісняється новими технологіями, такими, як COM/DCOM, .NET та іншими.

Приклад 12.3. Технології програмної інтеграції. Доступ до загального ресурсу через NetDDE

Завдання. Забезпечити створення загального ресурсу на комп'ютері «COMP1» для доступу до комірки Excel, аналогічно прикладу 12.2.

Рішення. Створюємо на ПК Сервері загальнодоступні ресурси, для чого у DDESHARE.EXE виконаємо таку послідовність:

- додамемо ресурс з іменем «SHARA» (назви писати без лапок);
 - назва прикладної програми: статичний зв'язок = Excel, всі інші порожні;
 - назва документа: статичний зв'язок = Лист2 або статичний зв'язок = [Книга1.XLS]Лист2, всі інші порожні;
 - виставити опцію: з дозволу запускати прикладну програму.
- З боку Клієнта, при ідентифікації ресурсу DDE вказується:
- Application — \\COMP1\NDDE\$;
 - Topic — SHARA\$;
 - Item — R3C2.

Таким чином, для зв'язку двох комірок Excel на різних комп'ютерах на першому (Серверний бік) налаштовуємо загальний ресурс, а на другому в ко-мірці-приймачі вказуємо рядок:

```
= '\\COMP1\NDDE$'! SHARA$!'R3C2'
```

Перед цим слід на обох комп'ютерах запустити необхідні служби (NETDDE.EXE).

Приклад 12.3

12.5. Технології COM/DCOM

12.5.1. Доступ до Процесів через COM

12.5.1.1. Загальні поняття. COM (Component Object Model — компонентна модель об'єктів) — це об'єктно-орієнтована технологія, яка дозволяє одній прикладній програмі користуватися об'єктами іншої прикладної програми або бібліотеки. **DCOM** (Distributed Object Model — розподілена COM) дала можливість прозорого використання цих об'єктів на віддалених вузлах. У чистому вигляді технологія COM має практичне використання тільки при програмуванні. Втім, розуміння фундаментальних засад дозволяє уникнути проблем при використанні похідних від неї технологій.

12.5.1.2. Технології на базі COM. Сама по собі модель COM реалізована у вигляді бібліотек, які завантажуються з Процесом. Компонент COM — це програмний код, який міститься в бібліотеці DLL (OCX) або у виконавчій програмі (типу EXE).

Спочатку технологія розроблялася для підтримки складних документів, наприклад, для можливості редагування таблиці Excel у документі Word. Це призвело до появи технології **OLE (Object Linking and Embedding** — зв'язування та вміщення об'єктів), перша версія якого базувалася на DDE. Оскільки DDE не забезпечувала необхідну гнучкість, наступна версія вже базувалася на COM.

Окрім термінів COM та OLE, існують також COM+, OLE Automation та ActiveX. **COM+** можна вважати як розширений варіант моделі COM, який вміщує різноманітні служби, що раніше були корисними доповненнями до COM.

OLE Automation — реалізація COM, що вимагає наявності спеціального типу DISP-інтерфейсу. DISP-інтерфейс дозволяє реалізувати механізм пізнього зв'язування з об'єктами COM, що дає можливість використовувати їх у VB, VBA, VB Script, JAVA script та інших системах програмування.

Терміном **ActiveX** називають усе, що належить до OLE Automation, плюс деякі допоміжні можливості, зокрема підтримку сценаріїв та використання *елементів управління ActiveX*. Останні, до речі, дуже широко використовуються у різноманітних програмах із області промислової автоматизації (SCADA-, MES-системи і т.п.). Хоч наведені терміни широко вживаються, сьогодні немає чіткого визначення терміна ActiveX, тому часто з цим словом пов'язують усе, що стосується OLE Automation та ActiveX.

12.5.1.3. COM-інтерфейси. Доступ до методів об'єкта COM проводиться через *COM-інтерфейси*. Під COM-інтерфейсом (надалі інтерфейсом) можна розуміти умовний логічний канал доступу до методів об'єкта. Програма, яка користується цим об'єктом, не викликає його методи (методи класу), а викликає методи *інтерфейсу*.

Кожний інтерфейс має унікальний 128-бітний ідентифікатор, який зветься *ідентифікатором інтерфейсу (IID* — Interface Identifier). Якщо два об'єкти підтримують інтерфейси з однаковими ідентифікаторами, вони повинні мати одні і ті ж методи інтерфейсів, які призначені для однієї й тієї ж самої дії. Це значить, що внутрішня реалізація об'єктів може бути різною, а інтерфейси та поведінка — співпадати.

12.5.1.4. COM-класи. Об'єкти (екземпляри) в COM створюються на основі *класу об'єкта*. Для людей, знайомих з об'єктно-орієнтованим програмуванням, зрозуміло, що об'єкт — це екземпляр класу. COM-об'єкти, які реалізують один і той самий набір COM-інтерфейсів належать до одного класу. Кожний клас має унікальний *ідентифікатор класу (CLSID* — Class Identifier).

12.5.1.5. Взаємодія між Процесами через COM. Взаємодія між прикладними Процесами функціонує на базі моделі Клієнт-Сервер. Програми, в яких містяться та створюються COM-об'єкти називаються *COM-Серверами*, а програми, які користуються цими об'єктами — *COM-Клієнтами*. Залежно від того, де перебуває програма COM-Серверу відносно COM-Клієнта, виділяють такі типи Серверів:

- *Внутрішній Сервер (In-Process Server)* — Сервер, який міститься в середині того самого Процесу, що і Клієнт (наприклад, реалізований як бібліотека DLL);
- *Локальний Сервер (Local Server)* — Сервер, який запускається як окремий Процес (EXE модуль);
- *Віддалений Сервер (Remote Server)* — Сервер, який запускається на віддаленому вузлі в мережі (EXE-модуль або бібліотека DLL у сурогатному Процесі).

Всі класи та інтерфейси реєструються в системному реєстрі Windows. Там, крім їх ідентифікатора, зберігається додаткова інформація, зокрема про імена файлів, в яких реалізований даний клас. Для доступу до них та їх активації використовується спеціальний процес SCM (диспетчер управління службами).

12.5.1.6. Контроль доступу до об'єктів COM. Для захисту доступу до об'єктів класів використовується так званий декларативний захист. Механізм активації і контролю доступу базуються на механізмі ролей. Для кожного зареєстрованого класу записи в реєстрі визначають, які користувачі Windows або групи користува-

чів мають право створювати екземпляри даного класу, тобто об'єкти. Користувачі визначаються при аутентифікації, тобто при їх перевірці, наприклад вводу пароля при вході у Windows. DCOM підтримує декілька рівнів аутентифікації:

- без аутентифікації;
- при першому з'єднанні із сервером;
- при кожному зверненні;
- всіх пакетів даних;
- пакетів даних з перевіркою їх цілісності;
- пакетів даних з перевіркою їх цілісності і шифрування пакетів даних.

Для настройки захисту у Windows є спеціальна утиліта *Dcomcnfg.exe*, яка запускається з командного рядка.

Приклад 12.4. Технології програмної інтеграції. Налаштування DCOM конфігуратору

Завдання. Забезпечити можливість доступу до COM-серверу Schneider-Aut OPC Factory Server з боку інших комп'ютерів у мережі.

Рішення.

1. Для роботи в мережі без виникнення проблем, необхідною умовою є відключення програм, які можуть блокувати мережний трафік. Відключення Брандмауерів можна зробити в розділі Адміністрування->Служби:

- брандмауэр Windows/Общий доступ к Интернету (ICS) = отключено;
- центр обеспечения безопасности = отключено.

2. Для роботи сервісів DCOM необхідно запустити та перевести в автоматичний запуск такі служби:

- запуск серверных процессов DCOM;
- локатор удаленного вызова процедур (RPC);
- удаленный вызов процедур (RPC).

3. Необхідно правильно задати конфігурацію DCOM. З панелі запуску викликається *dcomcnfg*, де налаштовуються властивості за замовченням: Служби компонентів -> Комп'ютери -> Мой компьютер -> контекстне меню Свойства.

Вкладка «Свойства по умолчанию»:

- разрешить использования DCOM = так;
- уровень проверки подлинности по умолчанию = подключение;
- уровень олицетворения по умолчанию = идентификация.

Вкладка «Безопасность COM».

Для «Права доступа» та «Разрешение на запуск и активацию» додати користувача типу «Все» і відкрити доступ до всього (виставити всі опції).

Вкладка «Протоколы по умолчанию».

Вибираємо «TCP/IP с ориентацией на подключение «->«Свойства», додаємо діапазон портів «0-65535» для призначення «Диапазон интрасети». В DCOM конфігураторі вибираємо «Настройка DCOM» і для кожного із серверів (Schneider-Aut OPC Factory Server) виставляємо всі настройки рівнями «по умолчанию». Для інших настройок:

- Расположение = запустить приложение на данном компьютере/запустить приложение на компьютере, где находятся данные;
- Удостоверение = текущий пользователь.

На вузлі клієнта і серверу повинен бути зареєстрований користувач з однаковими параметрами. Необхідно відмітити, що у даному прикладі відключаються більшість захисних механізмів, які надає Windows XP.

Приклад 12.4

12.5.2. Використання OLE та ActiveX

Від користувачів технології COM часто приховані механізми його функціонування. Так, при створенні мнемосхеми на SCADA можна скористатися елементами ActiveX, розмістивши їх на формі (сторінці, мнемосхемі ...) та прив'язавши їх властивості до певної змінної. Звісно, цей механізм повинен підтримуватися даною SCADA-програмою, а потрібні елементи ActiveX — присутністю на комп'ютері. За допомогою OLE можна помістити на мнемосхему таблицю Excel, документ типу PDF або інші OLE-об'єкти.

Якщо у складі програмного пакета (SCADA, офісна програма, CAD) є вбудована VBA, то програма написана цією мовою може працювати з об'єктами основного програмного пакета теж за допомогою ActiveX. Крім того, межі використання програмного пакета можна значно розширити, за рахунок використання додаткових елементів ActiveX.

Розглядаючи ActiveX, треба зазначити, що його поява пов'язана з бурхливим розвитком Інтернет-технологій, де елементи ActiveX можуть використовуватись для пошкваллення Інтернет-сторінок. Однак, у цій технології є ряд недоліків. Так, наприклад, заради безпеки вузлів у мережі брандмауери закривають доступ DCOM пакетам. Це обмежує його використання в мережі Інтернет. Знову ж таки з причин надійності кращою альтернативою ActiveX елементам у WEB-сторінках можуть бути JAVA-аплети. Адже код, який поставляється в ActiveX навмисно або ненавмисно може пошкодити дані на комп'ютері. Microsoft ввели додатковий механізм захисту під назвою «Цифровий підпис»; однак часто навіть «підписані» програми можуть призвести до збоїв у роботі системи. JAVA-аплети ж виконуються на віртуальній машині, тому більш надійні порівняно з ActiveX.

Приклад 12.5. Технології програмної інтеграції. Використання ActiveX-компонентів

Завдання. В SCADA Citect забезпечити можливість вибору оператором необхідної дати в зручному графічному інтерфейсі для подальшого її використання в перегляді історичних трендів. Для цього використати ActiveX елемент «Microsoft Date AND Time Picker Control».

Рішення. Спочатку в Citect Editor визначимо три внутрішні змінні типу INT: SEL_DAY, SEL_MONTH, SEL_YEAR. Ці змінні плануються для використання в мові Cicode для визначення початкової дати перегляду трендів. У Graphics Builder на панелі Tools вибираємо елемент ActiveX, після чого з'явиться список установлених на ПК елементів ActiveX. Необхідно вибрати елемент «Microsoft Date AND Time Picker Control», і вибравши вкладку «Appearance», закладку «TagAssociation», у вікні «Properties» настроїти такі властивості:

–властивість Day, вибрати тер SEL_DAY, у полі «Update association on» вибрати «Change» (рис. 12.3);

–властивість Month, вибрати тег SEL_MONTH, у полі «Update association on» вибрати «Change»;

–властивість Date, вибрати тег SEL_DATE, в полі «Update association on» вибрати «Change».

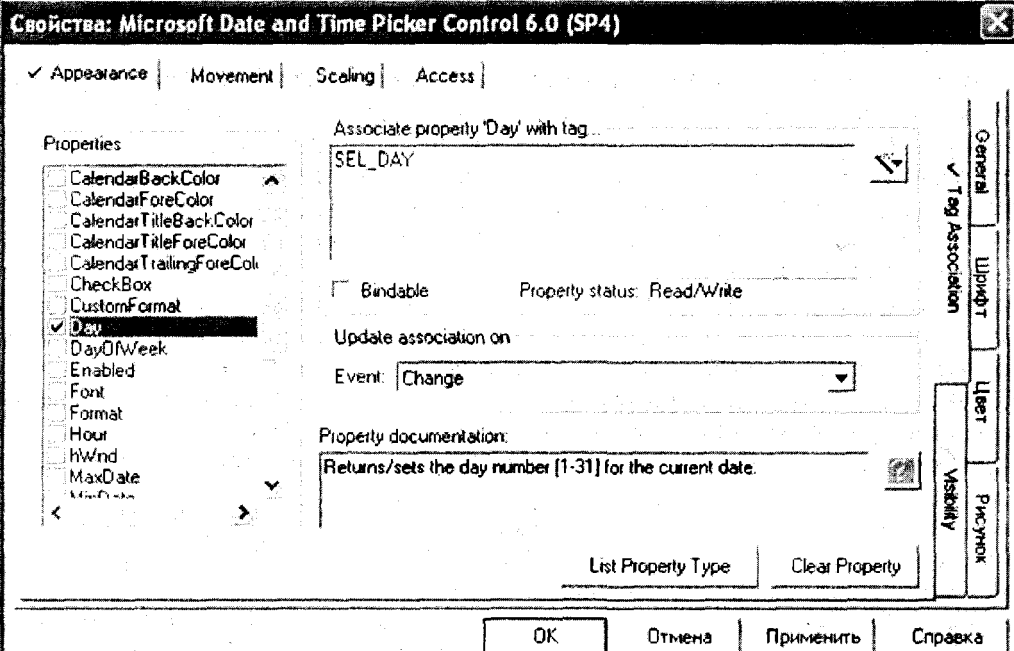


Рис. 12.3. Зовнішній вигляд вікна настройки елементів ActiveX в SCADA Citect

Зробимо деякі пояснення. На вкладці «Tag-Association» налаштується зв'язок властивостей ActiveX із значеннями змінних. Запис значень властивостей у змінні проводиться в момент виникнення події, яка задається в полі «Update association on».

Для відображення внутрішніх змінних на мнемосхемі виведемо 3 елементи Number і в закладці «Display Value» прив'яжемо їх значення до змінних SEL_DAY, SEL_MONTH, SEL_YEAR

Результат роботи RunTime системи з елементом ActiveX «Microsoft Date AND Time Picker Control» показаний на рис. 12.4.

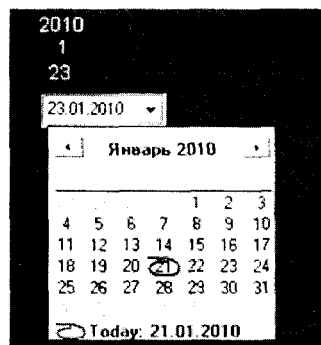


Рис. 12.4. Фрагмент мнемосхеми RunTime системи SCADA Citect з демонстрацією роботи ActiveX елемента

Приклад 12.5

12.6. WEB-технології

У системах автоматизації все більше впроваджуються Інтернет-технології. Так, наприклад, у контролерах або модулях розподілених входів/виходів, які підтримують Ethernet TCP/IP, часто вбудовані примітивні функції Web-серверу, з якого можна отримати необхідну початкову інформацію про його функціонування.

World Wide Web (*WWW*) — це гігантська розподілена система для доступу до зв'язаних документів, яка складається з великої кількості (мільйони) клієнтів та серверів. *Web-сервер* приймає запити на видачу документа і передає їх клієнту. Документ може зберігатися у файлі або генеруватися «на льоту» на сервері. Клієнт взаємодіє із web-серверами за допомогою *браузера* (browser). Браузер відповідає за правильну обробку документа, обробляє операції, зроблені користувачем, викликає за необхідності зв'язані документи.

Для посилання на конкретний ресурс (документ) використовуються уніфікований показник ресурсу *URL* (Uniform Resource Locator), який вміщує інформацію про те, де і як отримати доступ до документа. URL може складатися з трьох або чотирьох частин: ім'я протоколу (*http://*); DNS або адреса IP-вузла, де розміщена сторінка (*www.nuft.edu.ua* або *130.42.23.10*); номери TCP або UDP порту, за відсутності використовується стандартний (*:80*), імені файлу сторінки (*index.html*). Наприклад:

- *http:// www.nuft.edu.ua /home/index.html*;
- *http:// www.nuft.edu.ua :80 /home/index.html*;
- *http:// 130.42.23.10 :80 /home/index.html*.

Звісно, що необхідний вузол у локальній мережі або в Інтернеті міститься за його IP адресою. Для пошуку вузла за DNS-іменем, його необхідно зіставити з потрібною IP-адресою. *DNS* (Domain Name System — служба імен доменів) — це система, яка служить для ідентифікації вузлів за іменами, які мають ієрархічну структуру. Для перетворення DNS-імені в IP-адресу прикладна програма (наприклад, браузер) звертається за допомогою UDP-пакета до локального *DNS-серверу*, який шукає у своїй базі даних відповідну до імені IP-адресу і повертає її програмі. Це спрощена модель функціонування. Для того, щоб програма знала знаходження DNS-серверу, його IP-адресу необхідно їй вказати. В MS Windows це робиться в настройках TCP та IP: «Свойства подключения локальной сети» -> «Протокол TCP/IP» -> «Свойства».

Крім протоколу *http* в URL, можна використати також: *ftp*, *file*, *data*, *telnet*, *tel*, *modem*.

Більшість документів у Web описуються за допомогою мови *HTML* (Hyper Text Markup Language — мова розмітки гіпертексту). Ця мова призначена для розбивки тексту документа на окремі розділи, списки, таблиці і форми для його наочного представлення. Також у певні місця документа можна вставити зображення та анімацію. Для гнучкого представлення даних у HTML код можна вставити різноманітні сценарії, написані, наприклад, мовою Java-Script. Для посилання на інші зв'язані документи або навігації по самій сторінці використовуються гіперпосилання *hyperlinks*.

Крім HTML, використовується також мова **XML** (eXtensible Markup Language — розширювана мова розмітки). XML забезпечує тільки структурування документа. Вона не забезпечує ключових слів для форматування документа, наприклад, вирівнювання тексту за центром та виділення його курсивом, однак її можна використовувати для визначення похідних структур. Тобто, XML надає засоби для визначення різних типів документів. Все частіше ця мова використовується в якості зв'язку між прикладними програмами. Так, у технології **SOAP** (Simple Object Access Protocol — простий протокол доступу до об'єктів) XML надає можливість виконання віддаленого виклику процедур між прикладними програмами способом, незалежним від мови системи. Клієнт формує запит у вигляді повідомлення XML і відсилає його на сервер за протоколом HTTP. Сервер відсилає назад відповідь, яка являє собою форматоване XML-повідомлення. Таким чином можуть діяти прикладні програми, які працюють на різномірних платформах. У галузі автоматизації XML усе більше використовується як в якості стандартного формату збереження структурованих даних та для міграції програм користувача, так і для обміну даними між прикладними програмами

Для динаміки Web-сторінок у HTML код, крім сценаріїв, можуть вставлятися ActiveX-елементи та скомпільовані програми у вигляді аплетів. **Аплет** (applet) являє собою невелику автономну прикладну програму, яку можна надіслати клієнту і виконати її в просторі адрес браузера. Найчастіше використовуються аплети у вигляді програм мовою **Java**, скомпільованих в інтерпретований байт-код Java. Тобто, на боці клієнта повинна існувати наперед інстальована Java-машина (інтерпретатор), яка перетворює байт-коди в коди процесора. Існують серверні варіанти аплетів, які називаються **сервлетами** (servlets). Вони виконуються на боці серверу.

Будь-яка взаємодія між клієнтом і сервером Web відбувається за протоколом передачі гіпертексту (**HTTP** — Hyper text Transfer Protocol). Це протокол прикладного рівня моделі OSI, який базується на клієнт-серверній архітектурі. На транспортному рівні HTTP використовує протокол TCP та адресується 80-тим портом. Зв'язок здійснюється таким чином: клієнт (браузер) відправляє запити на сервер і очікує відповіді. Той, обробивши запит, відсилає потрібну сторінку.



Контрольні запитання до розділу 12

1. Наведіть приклади програмного забезпечення, яке використовується в інтегрованих автоматизованих системах управління. В чому полягає їх програмна інтеграція? Які проблеми виникають при програмній інтеграції продуктів від різних виробників?
2. Поясніть принципи функціонування багатозадачного режиму Windows.
3. Як розподіляються часові ресурси процесора між Потоками? Яким чином операційна система розділяє регістри процесора між Потоками за допомогою контексту Потoku?
4. Розкажіть, у чому полягає багатозадачність з витісненням?
5. Що таке віртуальна пам'ять?

6. Поясніть призначення Процесів в ОС Windows.
7. Що належить до об'єктів ядра Windows? Як ідентифікуються об'єкти ядра?
8. Які прикладні програми належать до Служб? Як запускаються Служби?
9. Як ідентифікуються вікна Windows? Яким чином вікна реагують на події?
10. Яке призначення рівнів привілеїв для процесорів Pentium? Які рівні привілеїв використовуються у Windows і для яких програм вони використовуються.
11. Прокоментуйте функціональну структуру Windows NT сумісних операційних систем. Що таке системні Процеси? Які системні Процеси Ви можете назвати?
12. Яке призначення інтерфейсу Win32 API? Які бібліотеки входять до складу Win32 API?
13. У чому полягає проблема обміну даними між двома різними Процесами?
14. Розкажіть про основні принципи функціонування технології DDE. Наведіть приклади використання DDE/NetDDE в інтегрованих системах управління.
15. Як ідентифікуються дані на DDE-Сервері? Розкажіть, які функції надає розділ System у даних DDE-Серверу?
16. Якими програмами забезпечується функціонування сервісів NetDDE? Які дії користувача NetDDE для роботи цієї технології? Яким чином виділяються загальні ресурси DDE?
17. Розкажіть про призначення технологій COM/DCOM. Які технології базуються на COM/DCOM?
18. Яким чином забезпечується доступ до методів об'єктів COM? Як забезпечується унікальність ідентифікаторів Інтерфейсів та Класів об'єктів COM?
19. Як відрізняються COM-Сервери за їх розміщення відносно COM-Клієнта?
20. Як забезпечується захист доступу до об'єктів класів? Де налаштовуються параметри захисту?
21. Наведіть приклади використання технології OLE, компонентів ActiveX.
22. Розкажіть про основні принципи обміну даними між браузером та WEB-сервером. На основі яких мережних, транспортних та прикладних протоколів функціонує такий обмін? Яким чином відбувається ідентифікація вузлів в Інтернет за символьним доменним іменем?
23. Розкажіть про функціональне призначення HTML, XML та SOAP.
24. Наведіть приклади використання технологій WEB в інтегрованих автоматизованих системах управління.
25. Розкажіть про використання технологій JAVA та ActiveX у WEB-технологіях.

ТЕХНОЛОГІЯ OPC

13.1. Загальні концепції

13.1.1. Передумови виникнення

13.1.1.1. Вибір SCADA/НМІ за набором підтримуваних комунікацій. Один з критеріїв вибору SCADA-програми — перелік підтримуваних комунікацій. Тобто, SCADA, з одного боку, і технічний засіб (надалі контролер), з іншого, повинен підтримувати однаковий протокол (або протоколи) промислової мережі. Найчастіше вибирають ту мережу, яка вже інтегрована в контролер. У цьому випадку при виборі SCADA враховують наявність даних протоколів у переліку комунікаційних драйверів.

13.1.1.2. Проблеми сумісності SCADA/НМІ з контролерами. При інтеграції продуктів одного виробника наявність у SCADA-програмі драйверів зв'язку з необхідними контролерами є очевидною. Найскладнішим є випадок, коли необхідно інтегрувати засоби від декількох виробників, ряд яких підтримують закриті протоколи. В цій ситуації дуже важко підібрати таку SCADA-програму, яка б підтримувала всі необхідні протоколи промислових мереж. Розглянемо, які можливі варіанти реалізації подібної системи.

1. Вибір іншої промислової мережі, яка б підтримувалась з боку SCADA та контролера. Цей варіант не завжди можливо реалізувати.

2. Написання спеціального драйвера, якого не існує в SCADA, для забезпечення зв'язку з контролером. Цей варіант потребує залучення досить кваліфікованого програміста, наявності відкритого програмного інтерфейсу з боку SCADA-програми та відкритого протоколу обміну з контролером.

3. Заміна частини одних контролерів іншими, для яких є драйвери зв'язку. Цей варіант потребує значних капітальних затрат і може бути використаний як винятковий захід.

4. Використання шлюзів для промислових мереж. Варіант також потребує значних капітальних затрат і не завжди реалізується.

Як бачимо, одні способи помітно обмежують реалізацію системи, а інші потребують значних капітальних затрат. Розглянемо приклад.

Приклад 13.1. OPC. Вибір SCADA програми

Завдання. Система управління технологічним процесом включає три підсистеми, в кожній із яких використовуються різні за типом і виробником контролера: TSX Premium (Schneider Electric), S7-300 (Siemens), Ломіконт Л-110 (Чебоксари) (рис. 13.1). Необхідно вибрати SCADA-програму серед трьох варіантів (Citect 7, Zenon 6, Trace Mode 6), яка б забезпечувала зв'язок з усіма підсистемами, враховуючи використання вбудованих комунікацій у ПЛК, та без придбання додаткових конвертуючих програмних засобів.

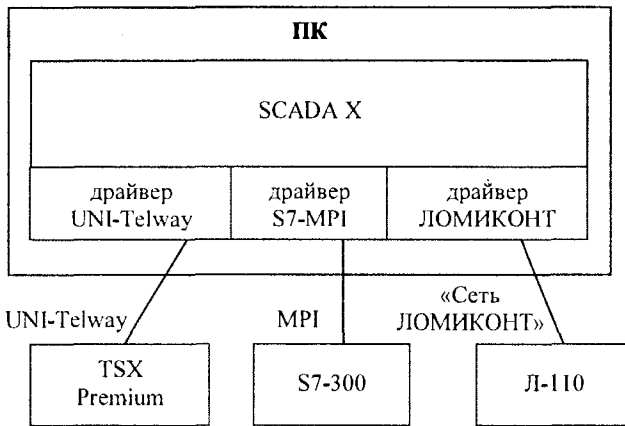


Рис. 13.1. Підключення декількох контролерів до SCADA-програми

Рішення. При виборі SCADA-програми необхідно перевірити перелік комунікаційних драйверів зв'язку для підтримки вказаних контролерів. Необхідні три типи драйверів: UNI-Telway (для TSX Premium), S7-MPI (для S7-300) та драйвер Сеть Ломіконт (для Л-110).

Протоколи для підключення засобів до SCADA-програми реалізовані у вигляді драйверів зв'язку. Зрозуміло, що інтерфейс драйверів з боку SCADA для кожної реалізації різний, тому драйвери для однієї SCADA-програми не підійдуть для іншої. Тож необхідно вибрати ту, яка має всі три драйвери. Citect 7 та Zenon 6 підтримують усі комунікації, крім Ломіконта. ТМ6 — усі комунікації, однак S7-MPI потребує драйверів SIMATIC NET і має певні обмеження у використанні. Таким чином, жодна з перелічених SCADA-програм не задовольняє вказані в умові задачі критерії вибору. Необхідне використання додаткових програм-конверторів, розглянути можливість вибору іншої SCADA, мережі або написання драйвера зв'язку.

Приклад 13.1

Наведений вище приклад — це один із багатьох, який обумовлює вирішення проблеми вибору SCADA. Для конкурентоздатності найбільш відомі компанії, які займаються розробкою програмного забезпечення верхнього рівня для систем

автоматизованого управління, забезпечили їх великою кількістю драйверів зв'язку для пристроїв найбільш відомих брендів. Однак, таке завдання не завжди може бути вирішене шляхом вибору певної SCADA. Адже пристрій управління може бути розроблений невеликою компанією, яка не є всесвітньо відомим брендом і для якої не розроблявся відповідний драйвер.

13.1.1.3. OPC — як універсальний драйвер зв'язку. Для подолання цієї проблеми групою великих компаній було вирішено створити стандартний інтерфейс доступу до даних «драйвера» з боку програмного забезпечення верхнього рівня. Таким чином, будь-який драйвер зі стандартним інтерфейсом може бути використаний будь-якою SCADA-програмою, яка цей інтерфейс підтримує. Технологія одержала назву OPC.

13.1.2. Стандарти OPC

13.1.2.1. Походження. Перша версія стандарту OPC (OPC DA 1.0) розроблена групою компаній, які в 1996 році організували некомерційну організацію **OPC Foundation** (www.opcfoundation.org), що займається розвитком та просуванням даної технології на ринку.

13.1.2.2. Доступні специфікації. Сьогодні стандарт існує у наборі специфікацій, основні з яких:

- 1) **OPC DA** (Data Access) — специфікація доступу до даних реального часу;
- 2) **OPC AE** (Alarms & Events) — для реалізації завдань попереджувально-аварійних сигналізацій;
- 3) **OPC HDA** (Historical Data Access) — для реалізації завдань ведення архіву та доступу до архівних даних;
- 4) **OPC DX** (Data eXchange) — для безпосереднього обміну між OPC-серверами;
- 5) **OPC XML** — для обміну даними через інтермережі за допомогою структур XML на базі WEB-сервісів та SOAP;
- 6) **OPC Batch** — для реалізації управління рецептурними завданнями;
- 7) **OPC UA (United Architecture)** — найновіший платформи незалежний стандарт, який об'єднує функції всіх наведених вище специфікацій, але функціонує не на базі COM, а WEB-сервісах.

Серед наведених вище стандартів найбільшу популярність тепер має OPC DA 2.0, усі інші зустрічаються набагато рідше. Стандарт OPC UA є найновішим стандартом, однак поки що не знайшов широкого вжитку. Планується, що OPC UA в найближчому майбутньому витіснить OPC DA 2.0 та всі супутні йому стандарти.

13.1.2.3. Визначення OPC. Першопочатково технологія **OPC** розшифровувалась як *OLE for Process Control* та була промисловим стандартом взаємодії між програмними засобами в галузі промислової автоматизації, який базується на об'єктній моделі COM/DCOM (OLE). Однак, при появі нових специфікацій, зокрема XML та UA, які не базуються на COM, слово «OLE» в аббревіатурі перестало відповідати дійсному функціональному змісту технології. Тому й немає офіційної розшифровки терміна OPC.

Тож будемо користуватися таким поняттям OPC: це відкрита технологія зв'язку (open connectivity) в галузі промислової автоматизації та управління виробництвом.

13.1.2.4. Загальна модель. Загалом технологія OPC забезпечує одній програмі (OPC-Клієнту) доступ до даних процесу іншої програми (OPC-Серверу) через стандартний набір інтерфейсів. Розглянемо набір інтерфейсів, які базуються на COM-технології, через призму їх використання в системах АСУТП (рис. 13.2). Це інтерфейси, які описуються специфікаціями OPC DA, OPC A&E та OPC HDA.

13.1.2.5. OPC DA (Data Access). COM-інтерфейси OPCDA стандартизують доступ OPCDA-Клієнта до даних процесу OPCDA-Серверу. В свою чергу, програма OPC-Сервер, як правило, здійснює обмін даними з контролерами або розподіленою периферією через специфічний, відмінний від OPC, інтерфейс. У цьому випадку, OPCDA-Сервер надає доступ OPCDA-Клієнту до даних процесу пристроїв, з якими він обмінюється, тому виступає в якості програми-шлюза.

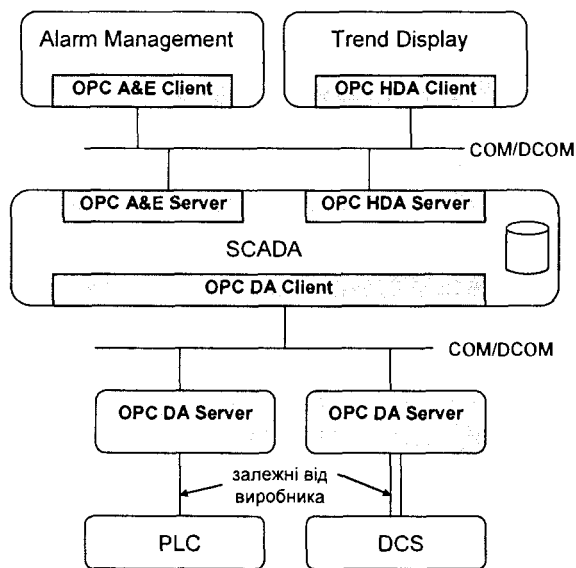


Рис. 13.2. Типове використання технології OPC

Всі перераховані в завданні 13.1 SCADA-програми можуть бути OPCDA-Клієнтами, оскільки мають відповідний драйвер. Тому, придбавши OPCDA-Сервер для Ломіконта, можна зв'язати Zenon та Citect з Л-110, а придбавши OPCDA-Сервер з підтримкою S7-MPI, можна повноцінно поєднати Trace Mode з S7-300. Таким чином, у даному випадку OPCDA-Сервер можна назвати універсальним (з боку SCADA) драйвером зв'язку. Слід зазначити, що ряд SCADA-програм повністю базуються на OPC (Genesis, Master SCADA).

13.1.2.6. OPC AE (Alarms & Events). OPCAE-Клієнт використовує OPCAE-Сервер для контролю за процесом, тобто за виникненням певних подій. Ці події налаштовуються в межах Серверу. OPCAE-Клієнт з'єднується з OPCAE-Сервером і підписується на отримання повідомлень про виникнення цих подій. При підписці OPCAE-Клієнт вказує додаткові критерії фільтрації повідомлень. Специфікацією підтримується можливість квітування повідомлень OPCAE-Клієнтом.

13.1.2.7. OPC HDA (Historical Data Access). OPCHDA-Сервер надає доступ OPCHDA-Клієнту для отримання збережених даних. Даний сервер підтримує два COM-Об'єкти: OPCHDA Server, який надає доступ до архівних даних, та OPCHDA Browser, який надає доступ до переліку змінних, які зберігаються в архіві.

Читання архівних даних проводять з використанням 3-х різних механізмів. Перший механізм передбачає зчитування архівних даних у певному часовому діапазоні для однієї або декількох змінних, які визначені іменами. Кількість зчитаних значень обмежується Клієнтом. Другий механізм передбачає зчитування архівних даних за часом їх відновлення (TimeStamp). Третій механізм дозволяє отримувати статистичну інформацію за збереженими даними. В OPC HDA інтерфейсі також передбачена можливість вставки, заміни або знищення архівних даних.

У наступних підрозділах буде розглядатися тільки OPC DA технологія.

13.1.3. Функціонування OPC з точки зору інтегратора

Найчастіше OPC-технологія використовується в якості універсального інтерфейсу до драйверів контролерів та периферійних пристроїв. Тобто, разом з контролером може поставлятися спеціальна програма — OPC-Сервер, який надає доступ до змінних цього типу контролеру. Тобто, OPC-Сервер, з одного боку, має драйвери для зв'язку з контролерами по конкретним протоколами промислових мереж, а з іншого, надає універсальний OPC-інтерфейс для зв'язку із сервером SCADA-програми. В такій системі SCADA буде OPC-Клієнтом.

На рис. 13.3 показана спрощена схема функціонування роботи OPC-технології в контексті описаної системи. База даних реального часу SCADA-програми (з умовною назвою «SamplSCADA») збирає дані з чотирьох джерел: ПЛК1, ПЛК2, ПЛК3 та ПЛК4. Для перших двох контролерів для збору даних використовуються драйвери зв'язку для цих ПЛК, точніше для протоколів промислових мереж, за якими вони з'єднуються. Дані зчитуються (або записуються) з ПЛК у БДРЧ. Зв'язок з ПЛК3 та ПЛК4 здійснюється через OPC-сервери з умовними назвами відповідно «Sampl.OPC» та «Exmpl.OPC» з використанням драйвера OPC-Клієнт. Тобто, OPC-Сервери через убудовані драйвери зчитують дані з ПЛК та зберігають їх у своїй базі даних реального часу. SCADA-програма, в свою чергу, зчитує дані з OPC-серверів. Запис даних відбувається аналогічно.

Для реалізації такого зв'язку користувач повинен:

1. Налаштувати OPC-Сервер за допомогою спеціалізованої програми-конфігуратора, що поставляється разом з ним: створити всі необхідні змінні серверу, тобто дати їм ім'я (ItemID) та вказати джерела даних у ПЛК, на які вони посилаються.

2. В SCADA-програмі вказати:

– назву OPC-Серверу, з яким необхідно зв'язатися (ProgID). У нашому прикладі це будуть два сервери «Sampl.OPC» та «Exmpl.OPC». Інколи SCADA надає можливість вибору ProgID зі списку зареєстрованих OPC-Серверів;

– для вибраної змінної в якості джерела даних вказати ім'я на OPC-Сервері, тобто ItemID, що був створений на 1-му кроці. Як правило, ItemID вибирається зі списку, який надає Browser на боці OPC-Клієнта.

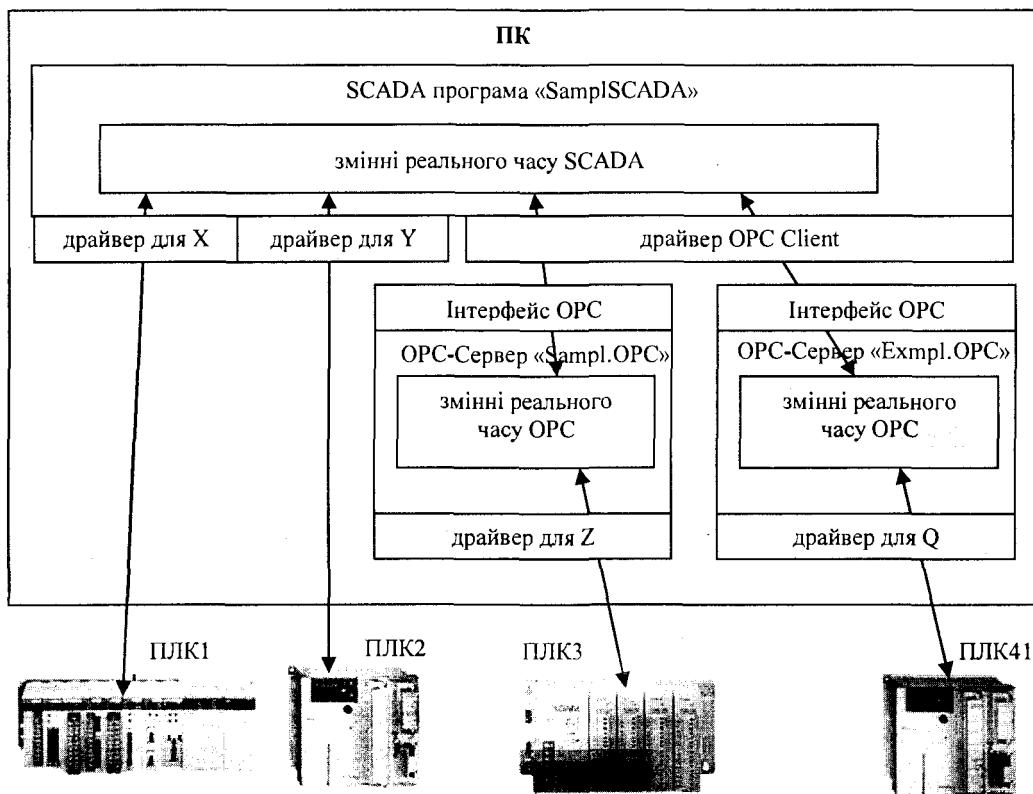


Рис. 13.3. Функціонування OPC з точки зору інтегратора

Приклад 13.2. OPC. Конфігурація OPC-Серверів та OPC-Клієнта (SCADA)

Завдання. Налаштувати OPC-сервери (Schneider-Aut.OFS, VIPA.OPC-Server) та SCADA (Citect) для зчитування таких змінних (рис. 13.4):

- MW100, що відповідає за температуру з PLC1(VIPA200) за протоколом MPI;
- %MW100, що відповідає за тиск з PLC2 (TSX Micro) за протоколом Modbus RTU.

Для зв'язку з контролерами використовуються OPC-сервери:

- OFS-Servrer від Шнейдер Електрик (ProgID=Schneider-Aut.OFS), що підтримує ряд протоколів, зокрема Modbus RTU;
- VIPA OPC-Server від фірми VIPA (ProgID=VIPA.OPCServer), що підтримує протокол MPI.

Рішення.

1. Конфігурування OFS. Відповідно до рис. 13.4, на даному OPC-сервері необхідно створити змінну з назвою «Pressure», джерелом даних для якої буде змінна %MW100 на PLC2. Для конфігурування OFS-серверу використовується утиліта OFS Configuration Tool.

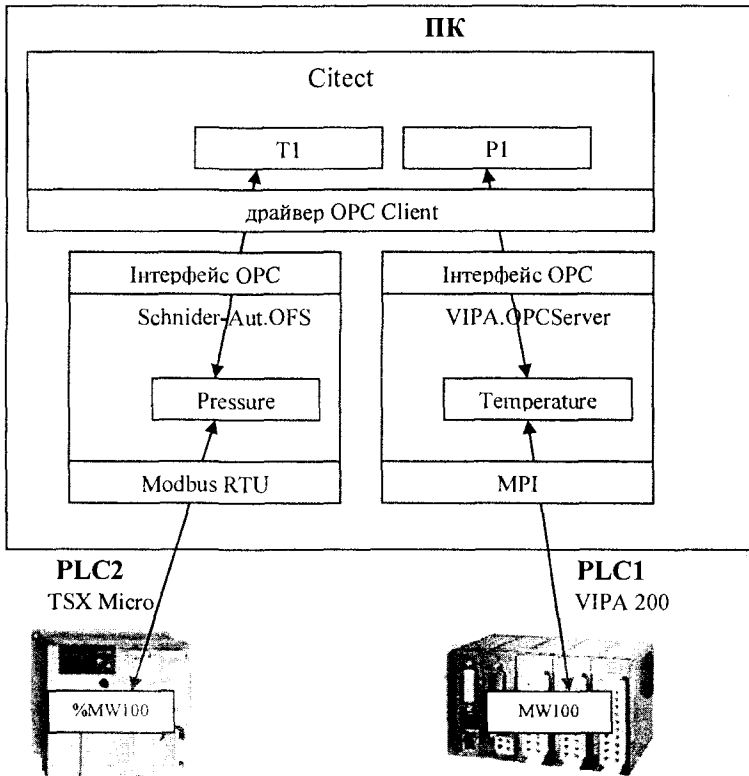


Рис. 13.4. Схема обміну даними SCADA Citect з використанням OPC

Дані визначаються таким чином (рис. 13.5):

- створюється псевдонім, який буде вказувати на адресу конкретного пристрою, з яким може обмінюватись OPC-сервер. У нашому випадку псевдонім пристрою має ім'я PLC2);

- для створеного псевдоніма вказується драйвер зв'язку, адреса пристрою та додаткові параметри, що уточнюють місцезнаходження його в мережі; в нашому випадку в результаті конфігурування створиться адреса: MODBUS01:1/T;

- для створеного псевдоніма пристрою вказати файл, в якому будуть міститись символічні імена та відповідні їм змінні контролера. У нашому випадку вибраний файл PLC2.CSV, в якому сформований запис:

%MW100 PRESSURE.

Відповідно до правил іменування змінних в OFS, ідентифікатор потрібної змінної буде формуватися таким чином:

ItemID = Ім'я_псевдоніму_пристрою!Ім'я_змінної

В нашому випадку ідентифікатор змінної буде — PLC2!PRESSURE.

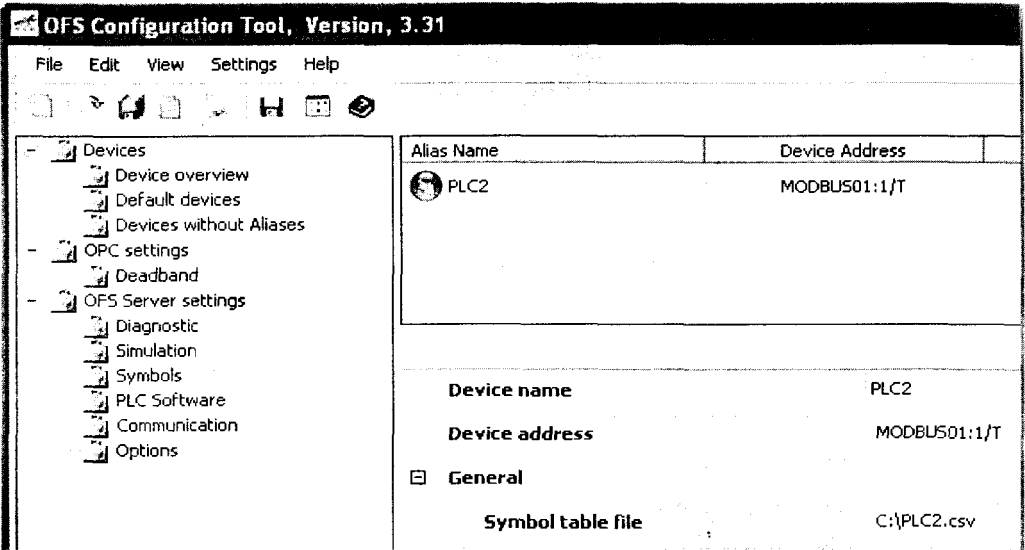


Рис. 13.5. Конфігурування OFS

2. Конфігурування VIPA-OPC. Конфігурування для VIPA OPC-Server виконується з використанням утиліти OPC Editor. Аналогічно OFS, на сервері створюються пристрої, в межах яких визначаються змінні, однак їх порядок і форма дещо відрізняються (рис. 13.6).

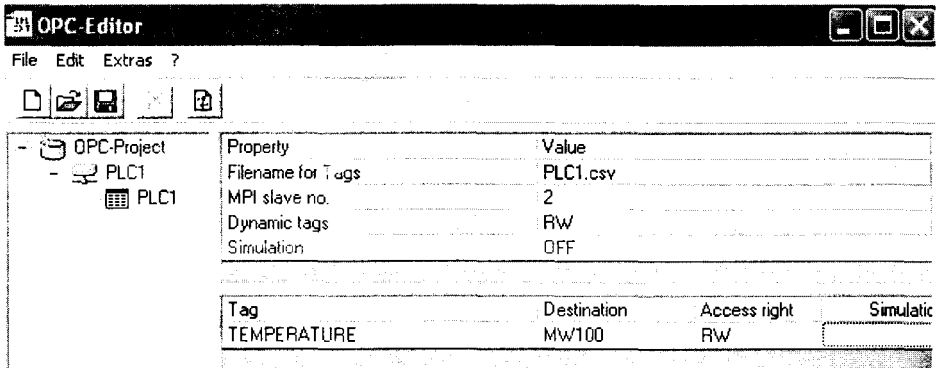


Рис. 13.6. Конфігурування VIPA-OPC

Спочатку вибирається драйвер мережі (в нашому випадку MPI). В межах мережі створюється PLC (у нашому випадку PLC1), а в межах пристрою вказується назва змінної (Tag) та її адреса в ПЛК (TEMPERATURE — MW100). Відповідно до правил іменування змінних у VIPA OPC, ідентифікатор змінної буде формуватися таким чином:

ItemID = Ім'я_PLC/Ім'я_змінної.

В нашому випадку ідентифікатор змінної буде: PLC1/TEMPERATURE.

3. Створення проекту для Citect. Усі змінні в Citect (VariableTag) формуються в межах віртуальних пристроїв (IODevices). Тому необхідно створити два IODevices, які будуть відповідати за OPC Сервери. Для створення IODevices скористаємось Express Communication Wizard (рис. 13.7). В обох випадках вибирається драйвер OPC DA Client. У полях Address вказується ім'я OPC-Серверів, тобто їх ProgID.

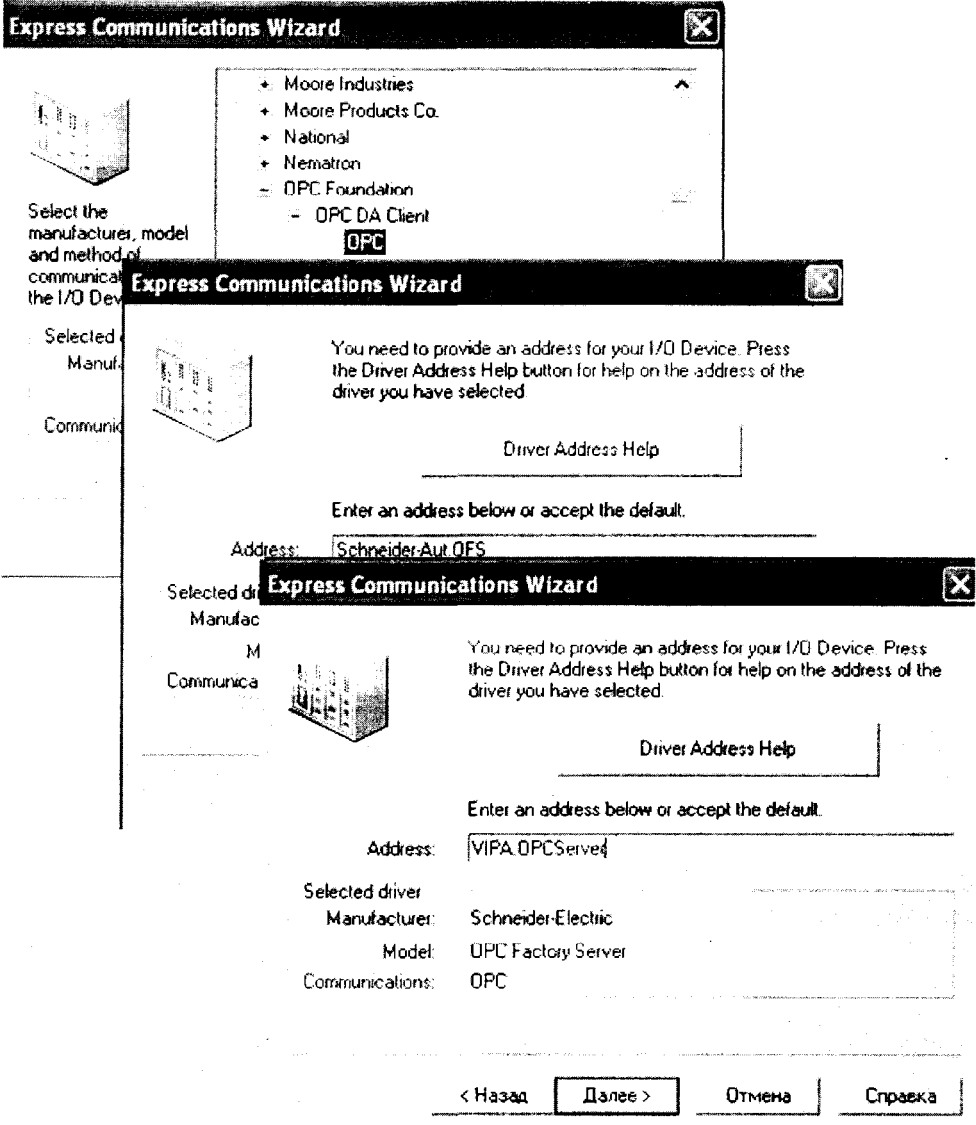


Рис. 13.7. Створення в Citect IODevices, прив'язаних до OPC-серверів

Далі необхідно отримати змінні (рис. 13.8) з прив'язкою до створених IODevices та в полі адреси вказати відповідні ItemID.

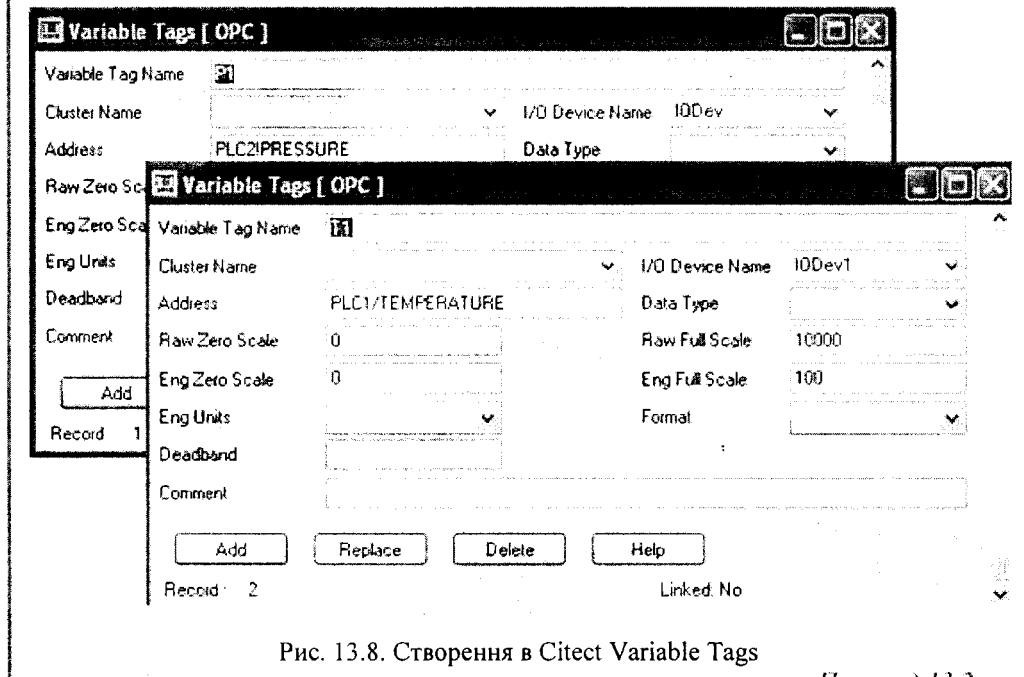


Рис. 13.8. Створення в Citect Variable Tags

Приклад 13.2

13.2. Принципи функціонування OPC DA

13.2.1. OPC модель взаємодії

13.2.1.1. Клієнт-Серверна модель. OPC DA технологія базується на Клієнт-Серверній архітектурі. OPC-Клієнт користується послугами OPC-Серверу, використовуючи COM-інтерфейси його об'єктів. У наведеному на рис. 13.9 прикладі, *OPC-Клієнтом* є SCADA-програма, завданням якої є відображення чотирьох змінних (%MW100-%MW103), які містяться на ПЛК. OPC-Сервер отримує необхідні дані через драйвери зв'язку і зберігає їх у своїй базі даних реального часу. Для того щоб доступитися до даних OPC-Серверу, OPC-Клієнт створює для себе OPC-Group (Group1, Group2), в яких створює OPC Item (Item1, Item2), що посилаються на ці дані.

OPC-Клієнт (OPC Client) — прикладна програма, яка вміє користуватися об'єктами OPC-Серверу за допомогою OPC-інтерфейсів (підмножина COM-інтерфейсів).

OPC-Сервер (OPC Server) — прикладна програма, яка надає доступ до визначених у специфікації OPC COM-об'єктів за допомогою OPC-інтерфейсів.

З одним OPC-Сервером можуть з'єднатися декілька OPC-Клієнтів. З іншого боку, одна і та сама програма OPC-Клієнт може одночасно користуватися послугами декількох OPC-Серверів. Тобто, OPC-технологія є мультиклієнтною і мультисерверною.

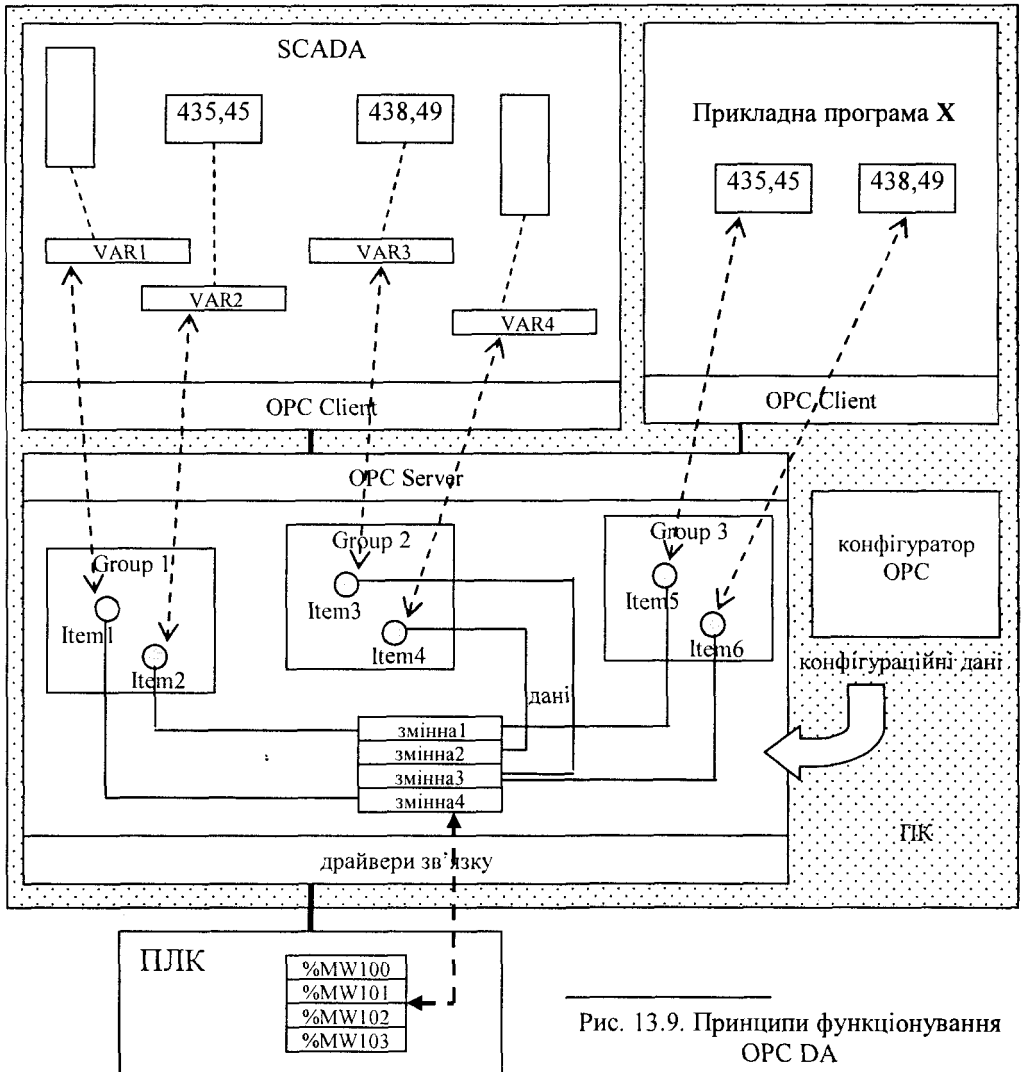


Рис. 13.9. Принципи функціонування OPC DA

13.2.1.2. Ідентифікація OPC-Серверу. Так як OPC-Сервер — це COM-Сервер, він реєструється на комп'ютері унікальним числовим ідентифікатором (GUID) та має унікальний строковий програмний ідентифікатор (*ProgID*). Тобто, для того щоб для OPC-Клієнта визначити, з яким OPC-Сервером на тому самому ПК йому необхідно з'єднатися, достатньо вказати його ProgID.

13.2.1.3. Об'єкти OPC-Item та ідентифікація даних. Об'єкт *OPC-Item* надає доступ до джерела даних (надалі *mez*) у межах OPC-Серверу, яке ідентифікується унікальним у межах сервера ідентифікатором *ItemID*. Тому при створенні OPC-Item'a вказується ItemID необхідного тега. Правила ідентифікації даних залежать від реалізації OPC-Серверу, а механізм визначення їх джерел (наприклад, адреси пристрою та змінної в ПЛК), як правило, реалізується в конфігураторі цього серверу. В прикладі 13.2 ми вже розглянули два варіанти формування ItemID, нижче більш детально розглянуті способи ідентифікації тегів. Тут тільки зазначимо, що весь список ItemID може мати деревовидну ієрархічну структуру, що дозволяє зручніше використовувати цей механізм у проектах з великою кількістю даних. Для навігації за списком/деревом ідентифікаторів OPC-Сервер, як правило, має об'єкт *OPC Browser*.

OPC-Item належить Клієнту, який його створив, і тому його не можуть використовувати декілька Клієнтів. Утім, є можливість посилатися на одні і ті ж дані. На рис. 13.9 два Клієнти одночасно використовують дані з %MW100 та %MW102, однак створюють для цього різні OPC-Item. Слід відмітити, що джерелом даних не обов'язково є змінна на зовнішньому пристрої, це можуть бути внутрішні дані самого Серверу.

З кожним OPC-Item'ом асоціюються плинне значення (*Value*), відмітка часу (*Time Stamp*) та якість (*Quality*).

13.2.1.4. Групування OPC-Item в OPC-Group. *OPC-Group* — об'єкт OPC-Серверу, який призначений для виконання групових операцій над OPC-Item'ами. Так як OPC-Item не може існувати без цього об'єкта, спочатку OPC-Клієнт створює OPC-Group, а потім у його межах створює OPC-Item'и. В інтерфейсі OPC DA 2.0 кожний OPC-Group, як і все його наповнення, належить окремому Клієнту. Механізм групування дозволяє розділяти дані за принципом читання/запису, періодичністю операцій та активувати/деактивувати відновлення змінних.

13.2.2. Механізми читання та запису даних процесу

13.2.2.1. Загальні підходи. Технологія OPC надає двохсторонній доступ до даних, тобто, як для читання, так і для запису. Механізми реалізації цих сервісів практично однакові за принципом, однак мають свої особливості у різних версіях специфікації OPC DA. Ми розглянемо їх у контексті 2-ї версії цієї специфікації, оскільки сьогодні вона є найбільш популярною.

Читання зводиться до вирішення таких питань:

– коли на OPC-Сервері повинні відновлюватися дані з пристроїв для кожного з OPC-Item'ів;

– яким чином про відновлення даних дізнається OPC-Клієнт і як він їх отримає.

Операції читання та запису проводиться одночасно для всіх Item'ів у межах OPC-Group.

13.2.2.2. Синхронне читання (Sync Read). Ініціація процесу відновлення змінних на OPC-Сервері може проводитись самим OPC-Клієнтом. Тобто, за необхідності OPC-Клієнт робить запит на відновлення певної OPC-Group. У такому

випадку Клієнт може заморозити виконання своєї програми (поток), поки не дочекається результату читання від OPC-Серверу. Такий спосіб називається **Синхронним Читанням (Sync Read)**. На рис. 13.10 графічно зображений процес обміну між OPC-Клієнтом та OPC-Сервером. За необхідності Клієнт робить запит за допомогою виклику методу SyncRead для OPC-Group «myGroup» та чекає, поки той не поверне відповідь.

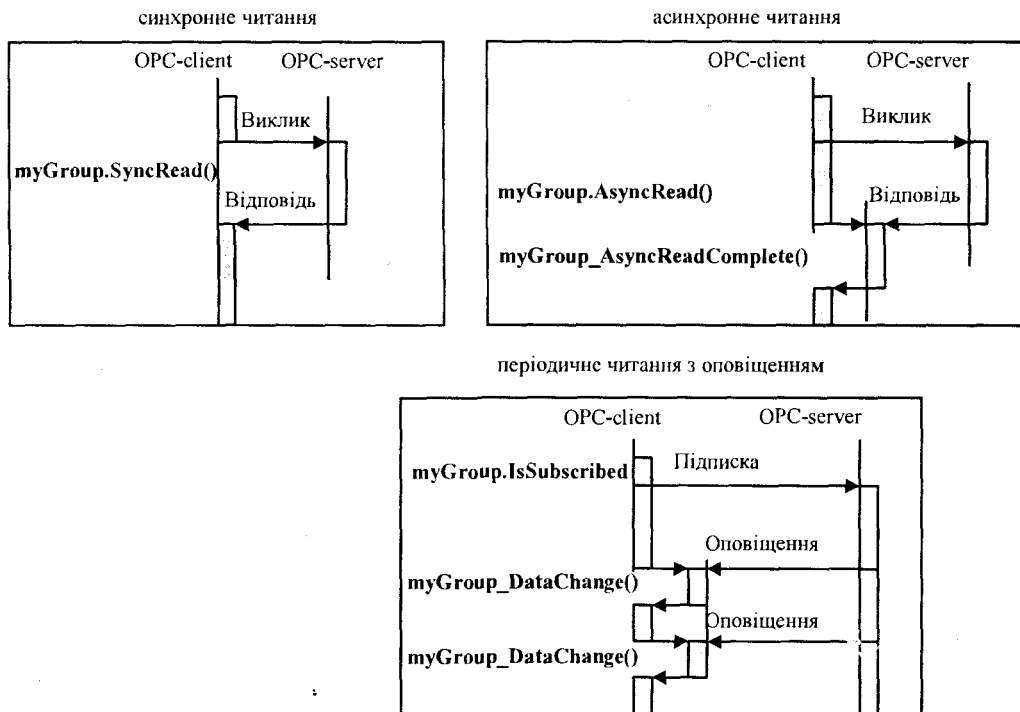


Рис. 13.10. Способи читання OPC-Item в OPC-Group

13.2.2.3. Асинхронне читання (Async Read). Механізм синхронного читання гальмує роботу програми (Потоку) Клієнта, тому доречний для читання невеликих обсягів даних. Альтернативою йому може бути використання **Асинхронного Читання (Async Read)**, при якому OPC-Клієнт теж ініціює обмін, однак не чекає результату обробки. Замість цього при закінченні процесу читання OPC-Сервер викликає функцію зворотного виклику OPC-Клієнта (обробник події *AsyncReadComplete*), в яку передає результат читання. Для реалізації такого механізму необхідно, щоб в об'єкті OPC-Group був активований механізм **Підписки (Subscribe)**.

13.2.2.4. Періодичне Читання з Оповіщенням (Periodical Read with Notify). За необхідності відновлення даних обидва наведених вище способи потребують від OPC-Клієнта кожний раз проводити запит до OPC-Серверу. Однак, як правило, дані необхідно читати періодично через певні інтервали часу. Для цього в специфікаціях OPC DA є механізм **Періодичного Читання з Оповіщенням**

(*Periodical Read with Notify*). При створенні OPC-Group Клієнт замовляє частоту відновлення Item'ів у межах цієї групи. Через вказані проміжки часу OPC-Сервер відновлюватиме ці дані, а результат буде зберігати в *Кеші (Cache)*. Якщо дані (Value або Quality) хоча б для одного OPC-Item'а в OPC-Group змінилися, викликається зворотна функція *Оповіщення (Notify)*, тобто обробник події *DataChange*, в параметрах виклику якого будуть передані нові значення. Для ефективного використання цього механізму можна скористатися зоною нечутливості (*Deadband*). Необхідно зазначити, що в об'єкті OPC-Group повинен бути активований механізм Підписки та прапорець *Активності (ACTIVE FLAG)*. Крім того, періодично відновлюватимуться тільки Активні OPC-Item.

13.2.2.5. Синхронний та асинхронний запис. Операції запису можуть проводитись двома способами: *Синхронний Запис (Sync Write)* та *Асинхронний Запис (Async Write)*. Функціонування повністю аналогічне як і в операціях читання (рис. 13.11).

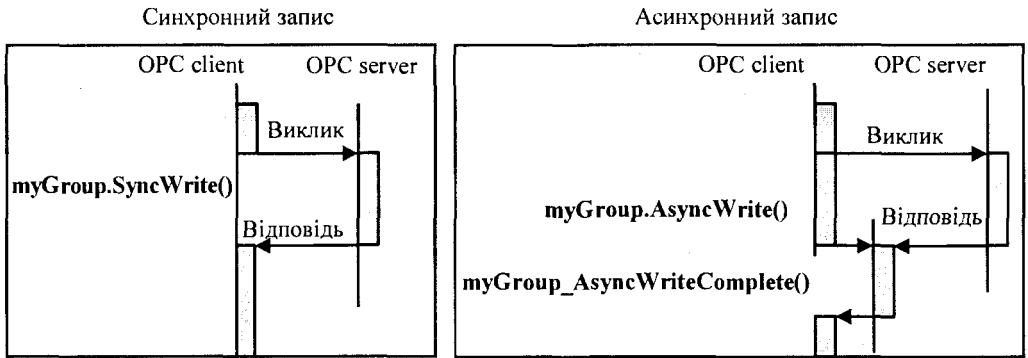


Рис. 13.11. Способи запису OPC Group

13.2.3. Ідентифікатори ItemID

13.2.3.1. Способи ідентифікації даних. ItemID — це унікальний, у межах OPC-Серверу, символічний ідентифікатор, який однозначно ідентифікує дані (теги) на цьому Сервері. Тобто, він не повинен вказувати, з якого пристрою беруться дані, а тільки те, де вони розміщуються на Сервері.

Найчастіше ItemID створюються за допомогою конфігуратора OPC Серверу. Саме там і конфігурується розміщення джерела даних. У прикладі 13.2. таким способом ідентифікуються теги на VIPA.OPCServer — «Temperature».

В деяких реалізаціях ItemID може створюватись автоматично. В цьому випадку розміщення джерела даних, зона нечутливості, мінімум та максимум та інше вказується в самому символічному рядку ідентифікатора. Наприклад, символічний рядок ItemID «MODBUS01:5!%MW100» в OFS Server (OPC Server від Schneider Electric) означає, що джерело даних розміщується на шині Modbus у Веденого з адресою 5, у змінній %MW100.

13.2.3.2. Доступ до списку ItemID (Об'єкт OPCBrowser). Для зручності ідентифікації джерела даних OPC-Сервер опціонально може підтримувати об'єкт-навігатор OPCBrowser. У версіях OPC DA 1.0/2.0 та OPC DA 3.0 реалізація механізмів навігації відрізняється, однак і в першому, і в другому випадку весь перелік-ItemID може формувати плаский список (*flat*) або ієрархічне дерево (*hierarchical*). Ієрархічна структура формується у вигляді дерева, приклад якого наведений на рис. 13.12.

OPCBrowser, як правило, потрібен тільки для Клієнтів, які необхідно конфігурувати, наприклад SCADA. За його відсутності, користувачу треба добре знати правило формування символічного рядка ItemID для конкретного OPC-Серверу, адже це не обумовлено в стандарті. Так, наприклад, в рядку ItemID з рис. 13.12 замість відокремлюючих крапок можуть використовуватися інші символи, наприклад «!».

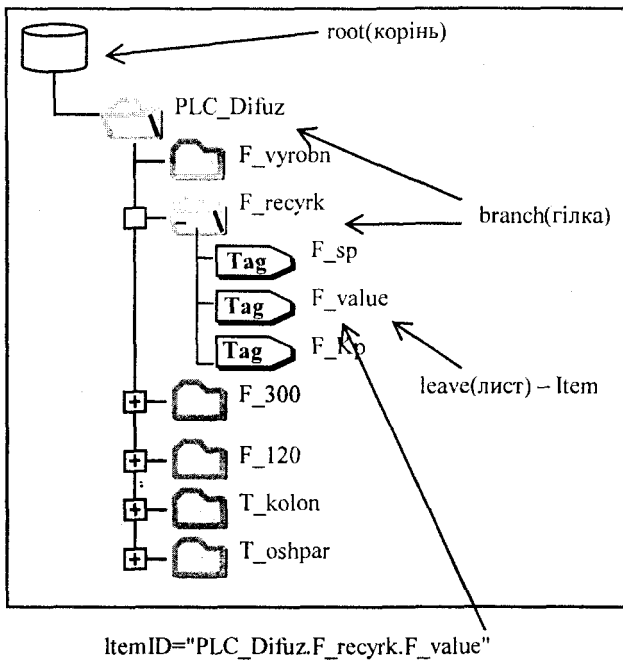


Рис. 13.12. Приклад ієрархічного дерева ItemID

Приклад 13.3. OPC. Створення OPCGroup, OPCItem та налаштування відновлення з використанням тестової програми OPC-Client

Завдання. Необхідно за допомогою тестової програми OFS-Client створити 2 групи в межах OFS-Серверу, сконфігурованого за прикладом 13.2 з такими параметрами:

- перша група з періодичним читанням та оповіщенням (період=1с);
- друга група тільки для синхронних операцій.

Продемонструвати роботу об'єкта OPC Browser для OFS Server та VIPA-OPC.

Рішення. OFS-Client — це тестова програма OPC-Клієнта, яка поставляється разом із Сервером OFS від Шнейдер Електрик. За її допомогою можна перевірити роботу будь-якого OPC-Серверу. При запуску OPC-Клієнта він пропонує вибрати один з OPC-Серверів, зареєстрованих на ПК (рис. 13.13). У нашому прикладі ми вибираємо Schneider-Aut.OFS.

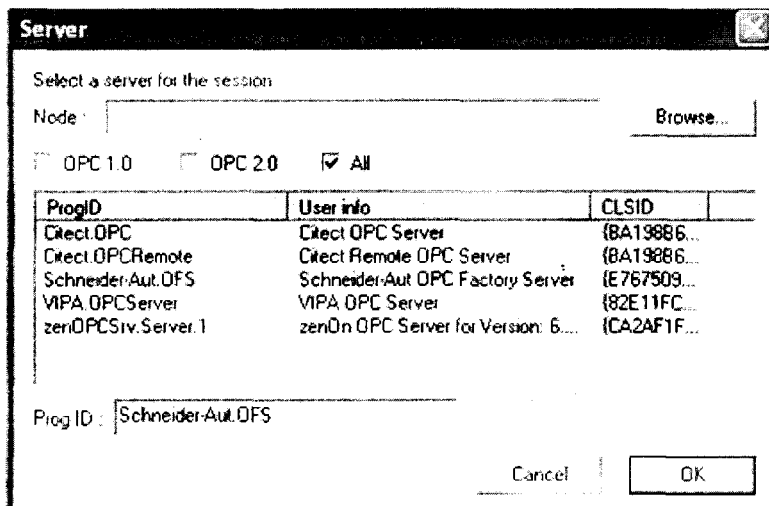


Рис. 13.13. Вікно вибору OPC Серверу в OFS Client

При підключенні до Серверу створюємо OPC-Group (рис. 13.14). Для першої визначаємо швидкість відновлення UpdateRate=1000 мс, виставляємо опцію Active — періодичне зчитування, виставляємо опцію Notification — активація оповіщення. На рис. 13.14 видно, що можна також налаштувати зону нечутливості та способи читання (Cash чи Device). Для другої групи опції Notification та Active повинні бути відключені. В обох випадках можна проводити операції синхронного читання або запису (Read Group/Write Group). Асинхронні операції можна проводити тільки для першої групи, оскільки лише для неї активована функція оповіщення (Notification). Крім того, для першої групи всі Item будуть автоматично відновлятися на сервері 1 раз/с.

Вікна браузера OPC-ItemID при підключенні до OFS Server та VIPA-OPC зображені відповідно на рис. 13.15 та 13.16. З них видно, що перелік ItemID для обох серверів має ієрархічний вигляд. Виставивши опцію Browse Flat можна перейти до пласкої форми. Крім того, браузер автоматично формує ItemID при його виборі.

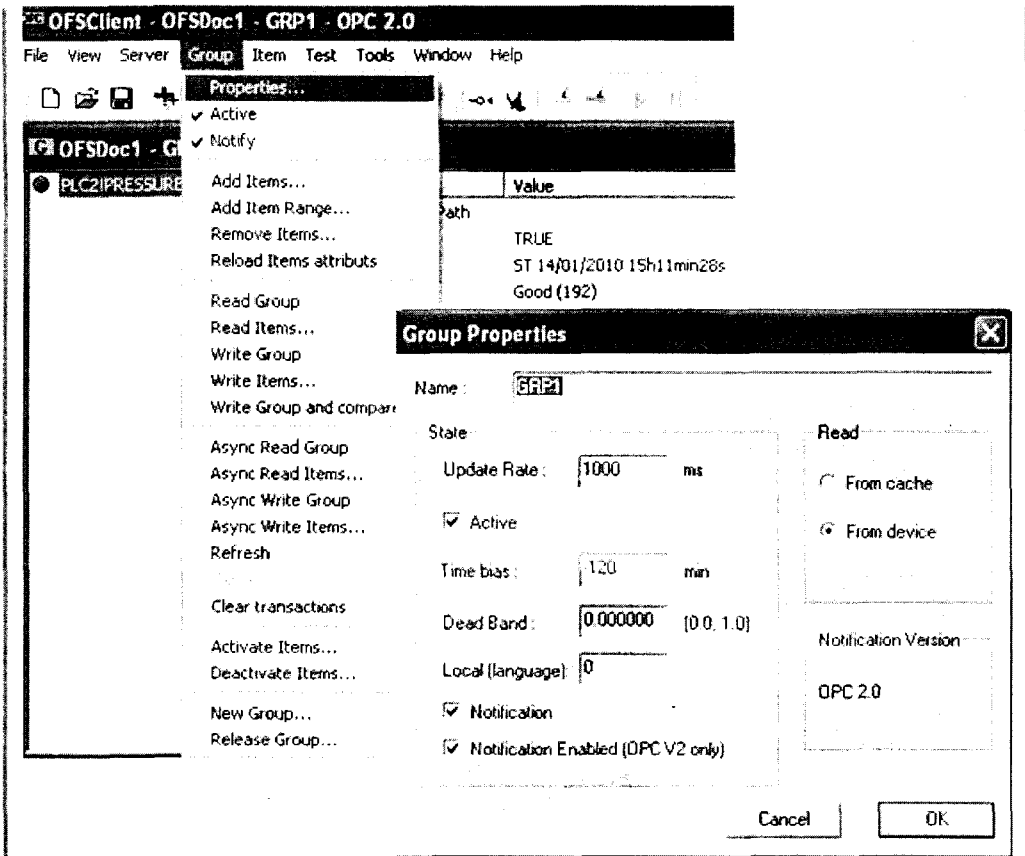


Рис. 13.14. Робота з OPC Group

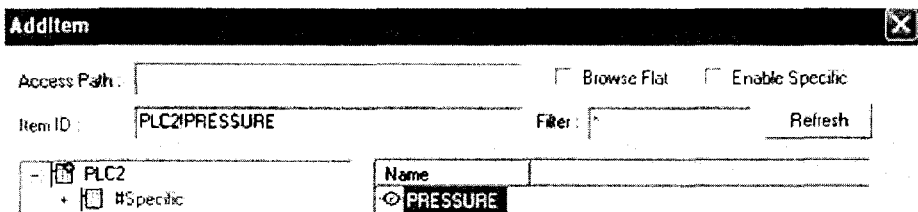


Рис. 13.15. Вікно браузера OPC-ItemID для OFS-Серверу

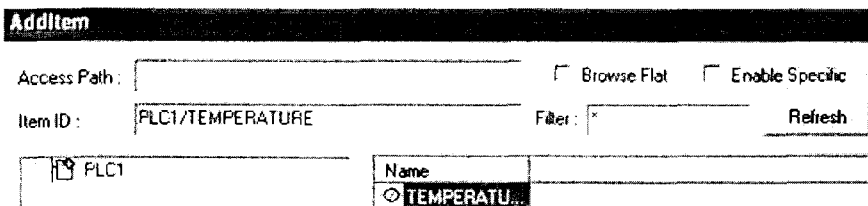


Рис. 13.16. Вікно браузера OPC-ItemID для Серверу VIPA-OPC

Приклад 12.5

13.2.4. Робота OPC-Клієнта з віддаленими OPC-Серверами

OPC-Клієнт та OPC-Сервер на одному і тому самому ПК запускаються як окремі Процеси. Обмін даними між цими Процесами відбувається за правилами СОМ-технології. Інколи виникає необхідність у з'єднанні OPC-Клієнта з віддаленим OPC-Сервером, який розміщений у мережі на іншому ПК. Для такого з'єднання використовуються сервіси DCOM.

На рис. 13.17 наведений приклад, у якому OPC-Клієнт (SCADA) на ПК1 з'єднується з локальним OPC-Сервером (OPCServer1) та двома віддаленими (OPCServer2 на ПК2 та OPCServer3 на ПК3). Для реалізації такого з'єднання для OPC-Клієнта, окрім ProgID, необхідно вказати розміщення ПК з OPC-Сервером, а також правильно налаштувати DCOM-конфігуратор. Таким чином необхідно виконати таку послідовність:

1. Налаштувати DCOM Конфігуратор на вузлі Серверу та Клієнта (див. розділ 12).
2. Вказати *Server Node* (Ім'я вузла OPC Серверу) або його IP.
3. Вказати ProgID Серверу.

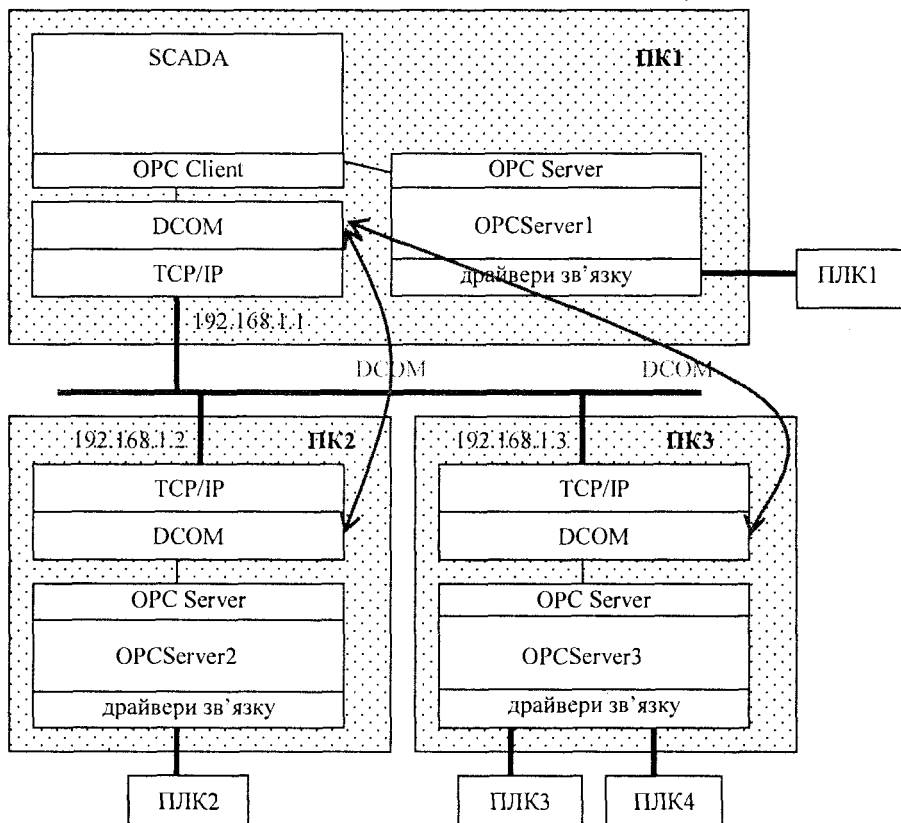


Рис. 13.17. Приклад з'єднання трьох пристроїв через DCOM

Однак, зв'язатися з віддаленим OPC за допомогою OPC DA можливо тільки у випадку коли вузли перебувають у межах одного домену або робочої групи Windows та не розмежовуються брандмауерами. Останні можуть не пропустити пакети COM (порти RPC, як правило, закриті для доступу), тому для з'єднання через Інтернет необхідно вдаватися до неабияких хитрощів. Щоб вирішити цю проблему, OPC Foundation пропонує технології OPC XML та OPC UA.

13.3. Типи OPC DA інтерфейсів

13.3.1. Загальний огляд типів інтерфейсів

OPC-Клієнти можуть зв'язуватись з OPC-Серверами через два набори COM-інтерфейсів: *OPC Custom Interfaces* і *OPC Automation Interfaces* (рис. 13.18). Набір Custom інтерфейсів є обов'язковим набором COM-інтерфейсів, який використовують SCADA-програми та прикладні програми написані, наприклад, на C++ чи Delphi.

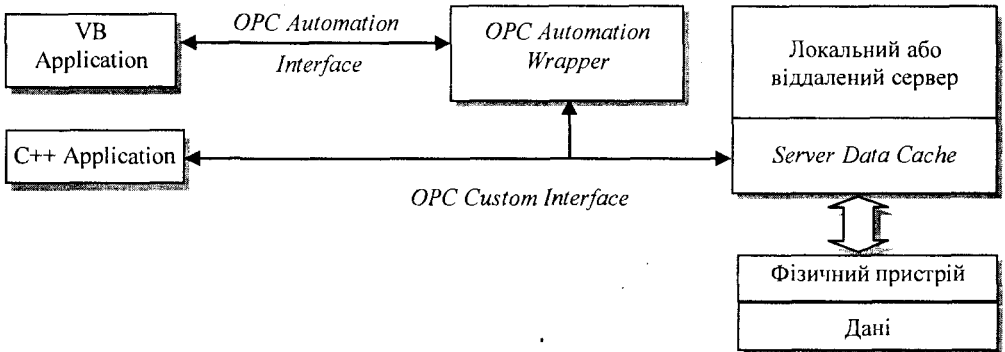


Рис. 13.18. Типи OPC-інтерфейсів

Набір Automation інтерфейсів був розроблений для використання їх у програмах, побудованих на MS Visual Basic ($\geq V5.0$) та VB for Application з використанням технології OLE Automation. Цей набір інтерфейсів є необов'язковим, але, як правило, присутній на OPC-Серверах основних виробників контролерів. Доступ до Серверу через такі види інтерфейсів проходить через так звану «Wrapper» DLL-бібліотеку, яка, в свою чергу, використовує той самий OPC Custom Interface.

OPC Custom інтерфейси більш гнучкі і використовуються для написання OPC-Клієнтів професійними спеціалістами, наприклад для SCADA-програм. Однак, для спеціалістів, які займаються розробкою та впровадженням систем автоматизації, може знадобитися знання Automation інтерфейсу, зокрема для написання невеликих програм на VBA, який убудований в офісний пакет MS Office. Цей інтерфейс виділяється простотою у використанні, оскільки програміст, не знаючи про механізми роботи COM/DCOM, може користуватися методами та властивостями об'єктів Серверу, як звичайного елемента управління. Для цього необхідно підключити бібліотеку

ку типів — Wrapper (як правило, «OPC Automation 2.0» (OPCDAuto.dll), яку перед цим необхідно зареєструвати в операційній системі, виконавши операції:

1. Переписати OPCDAuto.dll в папку system32 системної папки.
2. Запустити команду реєстрації бібліотеки: regsvr32 opcdaauto.dll.

13.3.2. Об'єктна модель інтерфейсу OPC Automation

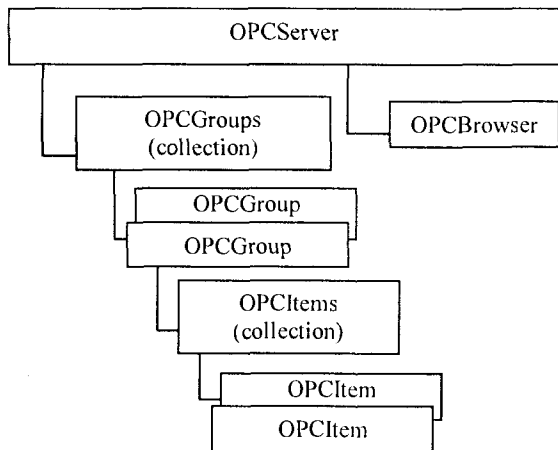


Рис. 13.19. Архітектура Automation-інтерфейсу

13.3.2.1. Основні об'єкти OPC Automation. Нижче перераховані основні об'єкти бібліотеки OPC Automation 2.0, а також приклад їх застосування (табл. 13.1). На рис. 13.19 показана об'єктна архітектура Automation-інтерфейсу.

Спочатку створюється об'єкт OPCServer, за допомогою якого можна продивитись список зареєстрованих Серверів та з'єднатися з необхідним. Об'єкт OPCBrowser призначений для доступу до списку ItemID. Створення та використання об'єктів OPCGroup та OPCItem можливе через об'єкти-колекції, в які вони об'єднані.

Таблиця 13.1

ОСНОВНІ ОБ'ЄКТИ OPC AUTOMATION 2.0 ТА ЇХ ПРИЗНАЧЕННЯ

Об'єкт	Призначення
OPCServer	Створюється для доступу до інших об'єктів OPC-серверу: OPCGroups і OPCBrowser.
OPCGroups	Колекція об'єктів OPCGroup для даного OPC-клієнта
OPCGroup	Даний об'єкт реалізує механізм створення та знищення об'єктів OPCItem та збору даних для цих об'єктів
OPCItems	Колекція об'єктів OPCItem, створених у межах OPCGroup
OPCItem	Об'єкт, який зберігає поточне значення даних, на які посилається, інформацію про їх стан та час останньої модифікації
OPCBrowser	Об'єкт, призначений для перегляду списку ідентифікаторів даних OPC-серверу

13.3.2.2. Послідовність створення об'єктів OPC Automation у програмі VBA. Застосування OPC Automation інтерфейсу не потребує високої кваліфікації програміста. Тому найбільш імовірні випадки його використання — це невеликі за обсягом програми, написані VB-подібною мовою. Це може бути макрос у MS Excel чи програма для візуалізації, написана в VisualBasic. Багато сучасних SCADA-програм також мають убудовані VBA чи VB-скрипти, в яких можна ко-

ристуватися об'єктами OLE Automation та ActiveX. За допомогою цього інтерфейсу відкриваються додаткові можливості при вертикальній інтеграції в комп'ютерно-інтегрованих системах.

Пропонується такий порядок створення об'єктів для доступу до даних на OPC-Сервері.

1. Створюється об'єкт OPCServer:
 - оголошується об'єкт типу OPCServer з ключовим словом NEW;
 - викликається його метод Connect.
2. Створюється об'єкт OPCGroup:
 - оголошується об'єкт типу OPCGroup (для роботи з подіями об'єкта необхідно оголосити його з ключовим словом WithEvents на рівні модуля);
 - в колекції OPCGroups викликається метод Add.
3. Створюються об'єкти OPCItem шляхом виклику методу AddItems або AddItem.
4. Для закінчення роботи з OPC-Сервером необхідно викликати метод Disconnect відповідного об'єкта OPCServer.

13.3.2.3. Користування сервісами читання та запису об'єктів OPC Automation у програмі VBA

1. Періодичне читання групи (OPCGroup) з повідомленням:
 - відповідний об'єкт OPCGroup необхідно оголосити з ключовим словом WithEvents на рівні модуля;
 - властивостям IsActive та IsSubscribe відповідного об'єкта OPCGroup необхідно присвоїти TRUE;
 - властивості IsActive кожного OPCItem, який треба відновлювати, необхідно присвоїти TRUE;
 - якщо хоча б одне значення OPCItem змінилося, буде викликаний обробник подій DataChange відповідного об'єкта OPCGroup.
2. Синхронне читання:
 - для читання декількох OPCItem у групі необхідно викликати метод SyncRead відповідного об'єкта OPCGroup;
 - для читання OPCItem необхідно викликати його метод Read.
3. Синхронний запис:
 - для запису декількох OPCItem у групі необхідно викликати метод SyncWrite відповідного об'єкта OPCGroup;
 - для запису OPCItem необхідно викликати його метод Write.
4. Асинхронне читання:
 - відповідний об'єкт OPCGroup необхідно оголосити з ключовим словом WithEvents на рівні модуля;
 - викликати метод AsyncRead відповідного об'єкта OPCGroup;
 - після обробки транзакції буде викликаний обробник подій AsyncRead-Complete відповідного об'єкта OPCGroup.
5. Асинхронний запис:
 - відповідний об'єкт OPCGroup необхідно оголосити з ключовим словом WithEvents на рівні модуля;
 - викликати метод AsyncWrite відповідного об'єкта OPCGroup;
 - після обробки транзакції буде викликаний обробник подій AsyncWrite-Complete відповідного об'єкта OPCGroup.

13.3.3. Синтаксис основних методів, властивостей та подій об'єктів бібліотеки OPCAutomation

13.3.3.1. Загальні формальні параметри. Перед тим, як розглянути основні методи та властивості об'єктів бібліотеки OPCAutomation, розглянемо формальні параметри, які найчастіше використовуються.

Для об'єктів типу OPCGroup та OPCItem:

– ClientHandle — це клієнтський дескриптор, за допомогою якого Сервер зможе вказати Клієнту на об'єкт (для ідентифікації);

– ServerHandle — це серверний дескриптор, за допомогою якого Клієнт зможе вказати на Сервері об'єкт.

У деяких методах та подіях повертаються такі параметри:

– Values — це масив прочитаних (записаних) значень OPCItem;

– Errors — це масив, у якому розміщені коди помилок, якщо операція пройшла невдало;

– NumItems — кількість OPCItem, які беруть участь у методі або повертаються з подією;

– Qualities — це масив, в якому розміщені значення параметрів якості даних

– TimeStamps — список значень UTC TimeStamps (час відновлення) для кожного із даних.

Параметр Source вказує на Джерело даних OPC_DS_CACHE (кеш) або OPC_DS_DEVICE (пристрій).

13.3.3.2. OPCServer. Повертає вказівку на колекцію об'єктів OPCGroup. Властивість за замовченням.

Таблиця 13.2

Властивість/метод	Опис
OPCGroups	Повертає вказівку на колекцію об'єктів OPCGroup. Властивість за замовченням.
GetOPCServers (Optional Node As Variant) As Variant	Повертає масив імен (ProgID's) зареєстрованих OPC Серверів на локальному або віддаленому комп'ютері. Повернений ProgID можна використати в методі Connect. Node — ім'я віддаленого вузла (комп'ютера), на якому потрібно переглянути список зареєстрованих OPC-серверів через DCOM. Наприклад: UNC ім'я («Comp1»), або DNS ім'я («server.com», «www.vendor.com», або «180.151.19.75»).
Connect (ProgID As String, Optional Node As Variant)	Використовується для з'єднання з OPC Data Access Server (через custom interface). При повторному використанні методу об'єкта без явного його від'єднання від Сервера «automation wrapper» буде автоматично відключати існуюче з'єднання. ProgID — Унікальне ім'я (ProgID), зареєстрованого OPC Data Access Server Node — Може використовуватись для підключення до іншого комп'ютера через DCOM.
Disconnect()	Від'єднання від OPC-Серверу. Після від'єднання всі посилання об'єктів OPCServer, OPCGroup(s) і OPCItem(s) будуть знищені.
CreateBrowser() As OPCBrowser	Створює об'єкт OPCBrowser та повертає посилання на нього. Метод буде виконаний тільки в тому випадку, якщо OPC Custom інтерфейс його підтримує (він є опціональний).

13.3.3.3. OPCGroups. Об'єкт OPCGroups є колекцією об'єктів OPCGroup, за допомогою якого можна створювати, знищувати та управляти ними. Через нього можна також настроїти властивості за замовчуванням для груп, які будуть створюватись. Але для вже існуючих груп ці властивості залишаються такими, як були

Таблиця 13.3

Властивість/метод	Опис
Item (ItemSpecifier As Variant) As OPCGroup	Повертає посилання на об'єкт OPCGroup за його ItemSpecifier (індекс, починаючи з 1, або ім'я). Метод за замовчуванням
Add (Optional Name As Variant) As OPCGroup	Створює новий об'єкт OPCGroup і додає його в колекцію. Група створюється з властивостями за замовчуванням. Після створення властивості групи також можуть змінюватись. Name — Ім'я групи. Повинно бути унікальним серед усіх груп даного клієнта, інакше буде згенерована помилка. Якщо ім'я групи не вказується, сервер автоматично присвоює їй унікальну назву.
GetOPCGroup (ItemSpecifier As Variant) As OPCGroup	Повертає посилання на об'єкт OPCGroup за його ItemSpecifier (OPCGroup's ServerHandle або ім'я).
Remove (ItemSpecifier As Variant)	Знищує вказану групу з колекції. ItemSpecifier — OPCGroup's ServerHandle або ім'я OPCGroup, яка буде знищена.

13.3.3.4. OPCGroup. OPC Group призначені для організації даних для клієнтів. Дані можуть як зчитуватись, так і записуватись. OPC Клієнт може сконфігурувати час відновлення, через який OPC Сервер буде повідомляти про зміни значень OPCItem у групі.

Таблиця 13.4

Властивість/метод	Опис
Name	Унікальне серед усіх груп даного Клієнта символічне ім'я групи. Якщо Клієнт не дає ім'я групі при створенні, то Сервер генерує його автоматично.
IsActive	Стан групи (активна/неактивна). Якщо група неактивна, то вона не буде періодично зчитувати дані.
IsSubscribed	Ця властивість вказує на можливість повідомлення Клієнту результатів роботи асинхронних операцій.
ClientHandle	Значення клієнтського дескриптора, який пов'язаний з групою. Його можна використовувати для ідентифікації групи при повідомленнях.
ServerHandle	Значення унікального серед усіх груп у Сервері серверного дескриптора, який пов'язаний з групою. Цей дескриптор використовується в деяких методах, наприклад OPCGroups Remove.
UpdateRate	Найменше значення часу в мілісекундах відновлення всіх змінних у групі. Сервер може не підтримувати таку швидкість відновлення, тоді він буде зчитувати дані з найбільшою швидкістю, на яку здатен.
OPCItems	Колекція об'єктів OPCItem. Властивість за замовчуванням

Властивість/метод	Опис
<p>SyncRead (Source As Integer, NumItems As Long, ServerHandles() As Long, ByRef Values() As Variant, ByRef Errors() As Long, Optional ByRef Qualities As Variant, Optional ByRef TimeStamps As Variant)</p>	<p>Операція синхронного читання, повертає значення(value), якість (quality) та час відновлення(timestamp) items у групі. Функція буде чекати повернення результату. Дані можуть бути прочитані із кеша у випадку, якщо вони будуть задовільняти 'UpdateRate' і 'deadband' для групи, в іншому випадку вони будуть прочитані з пристрою. Якщо стан групи або елемента буде неактивний, то при читанні з кеша Quality буде рівним OPC_QUALITY_OUT_OF_SERVICE.</p>
<p>SyncWrite (NumItems As Long, ServerHandles() As Long, Values() As Variant, ByRef Errors() As Long)</p>	<p>Операція синхронного запису значення одного або кількох елементів у групі. Функція повертає управління тільки у випадку, коли значення буде записано на пристрій(DEVICE).</p>
<p>AsyncRead (NumItems As Long, ServerHandles() As Long, ByRef Errors() As Long, TransactionID As Long, ByRef CancelID As Long)</p>	<p>Операція синхронного читання повертає значення (value), якість (quality) та час відновлення (timestamp) items у групі, результат повертається при повідомленні AsyncReadComplete для даного об'єкта OPCGroup. Асинхронне читання проводиться тільки з пристрою ('DEVICE') при будь-яких значеннях стану ACTIVE групи або елемента. TransactionID — ID транзакції, за якою Клієнт при отриманні повідомлення про обробку зможе ідентифікувати номер запиту на читання. CancelID — ID, згенерований сервером, для можливості відмови поточної транзакції.</p>
<p>AsyncWrite (NumItems As Long, ServerHandles() As Long, Values() As Variant, ByRef Errors() As Long, TransactionID As Long, ByRef CancelID As Long)</p>	<p>Операція асинхронного запису значення одного або кількох елементів у групі. Результат операції повертається з повідомленням AsyncWriteComplete поточного об'єкта OPCGroup. TransactionID — ID транзакції, за якою Клієнт при отриманні повідомлення про обробку зможе ідентифікувати номер запиту на запис. CancelID — ID, згенерований Сервером, для можливості відмови поточної транзакції.</p>
<p>DataChange (TransactionID As Long, NumItems As Long, ClientHandles() As Long, ItemValues() As Variant, Qualities() As Long, TimeStamps() As Date)</p>	<p>Подія виникає в тому випадку, коли значення або якість елемента в групі змінилося і якщо даний елемент і група є активними. TransactionID — ID (ідентифікатор) клієнтського запиту. Повертає ненульове значення ID запиту функції AsyncRefresh, 0 — якщо подія виникла в результаті відновлення за звичайною підпискою на відновлення.</p>
<p>AsyncReadComplete (TransactionID As Long, NumItems As Long, ClientHandles() As Long, ItemValues() As Variant, Qualities() As Long, TimeStamps() As Date, Errors() As Long)</p>	<p>Подія виникає після закінчення обробки методу AsyncRead</p>

Закінчення табл. 13.4

Властивість/метод	Опис
AsyncWriteComplete (TransactionID As Long, NumItems As Long, ClientHandles() As Long, Errors() As Long)	Подія виникає після закінчення обробки методу AsyncWrite

13.3.3.5. OPCItems. Це колекція елементів OPCItem, яка також визначає настрійку об'єктів OPCItem за замовчуванням. При добавленні об'єкта OPCItem у колекцію властивості DefaultXXXX будуть ініціалізуватися за замовчуванням.

Таблиця 13.5

Властивість/метод	Опис
DefaultIsActive	Статус активності для новостворюваних елементів. За замовчуванням True.
Count	Кількість елементів в колекції.
Item (ItemSpecifier As Variant) As OPCItem	Повертає посилання на елемент у колекції. ItemSpecifier — Номер (починаючи з 1) елемента в колекції.
GetOPCItem (ServerHandle As Long) As OPCItem	Повертає посилання на OPCItem за його ServerHandle. ServerHandle — це серверний дескриптор OPCItem.
AddItem (ItemID As String, ClientHandle As Long)	Створює новий об'єкт OPCItem і додає його в колекцію. ItemID — Повний ідентифікатор елемента. ClientHandle — Клієнтський дескриптор, який буде присвоєний об'єктові.
AddItems (Count As Long, ItemIDs() As String, ClientHandles() As Long, ByRef Server- Handles() As Long, ByRef Errors() As Long, Optional RequestedDataTypes As Variant, Optional AccessPaths As Variant)	Створює декілька об'єктів OPCItem і додає їх у колекцію. Count — Кількість елементів для добавлення. ItemIDs — Масив із повних ідентифікаторів ItemID. ClientHandles — Масив клієнтських дескрипторів елементів, які добавляються. ServerHandles — Масив серверних дескрипторів елементів, які добавляються. Errors — Масив помилок, який індикує результат створення елементів. RequestedDataTypes — Опціональний Список типів даних за кожним з елементів. AccessPaths — Опціональний Список типів даних за кожним елементом, в якому вказаний шлях доступу Access Path's.
Remove (Count As Long, ServerHandles() As Long, ByRef Errors() As Long)	Знищує об'єкти OPCItem і видаляє з колекції
SetActive (Count As Long, ServerHandles() As Long, ActiveState As Boolean, ByRef Errors() As Long)	Дозволяє активувати або деактивувати OPCItem у колекції ActiveState — TRUE — для активації. FALSE — для деактивації.

13.3.3.6. OPCItem. Являє собою об'єкт, який підключається до джерела даних у межах Серверу. Він зв'язаний з Value, Quality та Time Stamp кожного з елементів.

Таблиця 13.6

Властивість/метод	Опис
ClientHandle	Клієнтський дескриптор, асоційований з OPCItem. За цим дескриптором Клієнт може визначити джерело даних при виклику події в групі.
ServerHandle	Унікальний серверний дескриптор для елемента. Його необхідно вказувати при виклику деяких методів (наприклад OPCItems.Remove).
IsActive	Стан елемента (Активний = TRUE / неактивний = FALSE).
Value	Повертає останнє значення елемента. Властивість за замовчуванням.
Quality	Повертає останню величину якості елемента.
TimeStamp	Повертає останню величину часового відбитку елемента.
Read (Source As Integer, Optional ByRef Value As Variant, Optional ByRef Quality As Variant, Optional ByRef TimeStamp As Variant)	Зчитує дані для елемента з Сервера. Дані можуть бути зчитані як з кеша так і з пристрою (OPCCache або OPCDevice) для відновлення інформації про значення, якості та часового відбитку елемента.
Write (Value As Variant)	Записує значення елемента в Сервері.

Приклад 13.4. OPC. Робота з OPC за допомогою VBA.

Завдання. Написати програму на VB/VBA для періодичного зчитування 2-х змінних з ПЛК та 1 внутрішньої змінної OFS Серверу.

Рішення. В VB створюємо форму, в якій розміщуємо 3 елементи: Labels(1), Labels(2), Labels(3). У цих елементах будуть відображатися значення змінних. Також створимо 3 кнопки, які будуть викликати відповідні процедури: SerConnect, CreateGroups та CreateItems.

У розділі оголошення змінних (рис. 13.20) вказуємо змінні для об'єктів класів OPCServer та OPCGroup. Для зберігання клієнтських та серверних індексів Item оголошуємо два масиви.

При створенні групи в процедурі CreateGroups вказується ім'я групи, періодичність відновлення та активність. При створенні Item у методі AddItems передається масив клієнтських дескрипторів (ClientHandleItems). При відновленні даних буде викликаний обробник події DataChange об'єкта MyOPCGroup, в який у вигляді масивів будуть передані значення ItemValues(), TimeStamps(), Qualities() а також масив клієнтських дескрипторів тих змінних, які змінили своє значення. За клієнтськими дескрипторами змінні будуть відображені у кожному текстовому полі Label.

```

Private MyOPCServer As New OPCServer ' оголошений в ключовим словом New
Private WithEvents MyOPCGroup As OPCGroup
Private C1HandleItems() As Long, SerHandlesItems() As Long
Private Values(1 To 3) As MSForms.Label
'----- процедура для з'єднання з сервером
Sub SerConnect()
Dim MyProgID As String, NodeComp As String
MyProgID = "Schneider-Aut.OFS" ' ProgID сервера
NodeComp = "" ' на локальному комп'ютері
MyOPCServer.Connect MyProgID, NodeComp ' виклик методу
MsgBox "Сервер підключений!"
End Sub
'----- процедура для закінчення сеансу
Sub SerDisconnect()
MyOPCServer.disconnect
End Sub
Sub CreateGroups()
MyOPCServer.OPCGroups.DefaultGroupIsActive = True
Set MyOPCGroup = MyOPCServer.OPCGroups.Add("Group1")
MyOPCGroup.IsSubscribed = True
MyOPCGroup.UpdateRate = 1000
MsgBox "Група створена!"
End Sub
Sub CreateItems()
Dim i As Integer, MyItemIDs(1 To 3) As String
Dim Errs() As Long
ReDim C1HandleItems(1 To 3) As Long
For i = 1 To 3
C1HandleItems(i) = i
Next i
MyItemIDs(1) = "UNTLW01:0.254.0!%NW100"
MyItemIDs(2) = "UNTLW01:0.254.0!%NW102"
MyItemIDs(3) = "LOCAL:!Zminna"
MyOPCGroup.OPCItems.DefaultIsActive = True
MyOPCGroup.OPCItems.AddItem 3, MyItemIDs(), C1HandleItems(), SerHandlesItems(), Errs()
MsgBox "OPCItem добавлені!"
End Sub
Private Sub MyOPCGroup_DataChange (ByVal TransactionID As Long, _
ByVal NumItems As Long, _
ClientHandles() As Long, _
ItemValues() As Variant, _
Qualities() As Long, _
TimeStamps() As Date)
Dim i As Integer
For i = 1 To NumItems
Values(ClientHandles(i)).Caption = ItemValues(i)
Next
End Sub

```

Рис. 13.20. Лістинг програми на VBA для роботи з OPC Сервером

Приклад 13.4

13.4. Область застосування технології OPC

У прикладах, наведених вище, розглянута технологія OPC-DA в контексті вирішення проблеми доступу до даних ПЛК зі SCADA. Тобто, OPC-Сервер розглядався у якості стандартного драйвера зв'язку. Однак, застосування OPC цим не обмежується.

На рис. 13.21 показаний приклад використання інтерфейсів OPC в якості «мосту» між двома прикладними програмами на різних ПК. При горизонтальній ін-

теграції може знадобитися об'єднання в єдиний інформаційний простір SCADA програм. Популярність OPC-технології призвела до появи в останніх не тільки клієнтської сторони OPC, а і серверної. Тобто, SCADA може виступати як OPC-Клієнтом, так і OPC-Сервером.

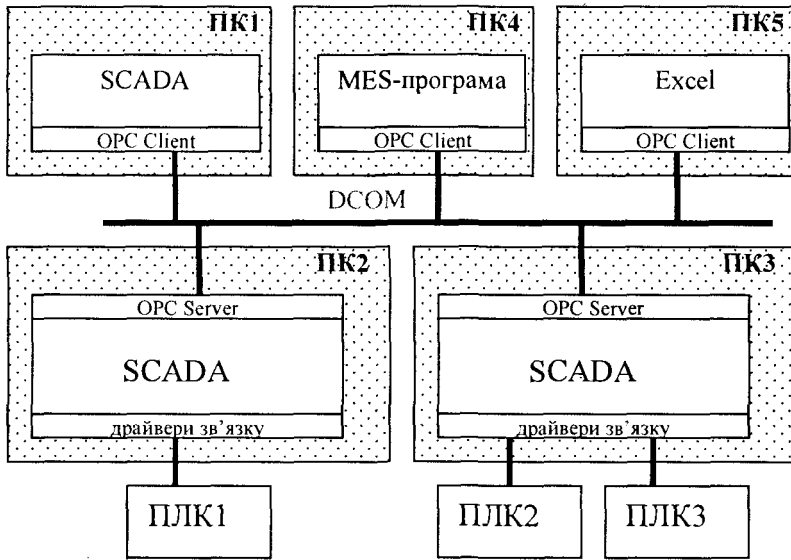


Рис. 13.21. Приклад використання OPC в якості «мосту»

Існування в SCADA Серверного інтерфейсу дає можливість доступитись до її даних з боку прикладних програм рівня MES чи ERP. Офісні програми завдяки наявності VBA та ActiveX теж надають таку можливість.



Контрольні запитання до розділу 13

1. Розкажіть про передумови виникнення технології OPC. Наведіть декілька способів подолання проблеми сумісності зв'язку SCADA-програм з контролерами, периферійними засобами тощо.
2. Поясніть основні принципи функціонування технології OPC. На якій технології міжпрограмного обміну вона базується? Назвіть специфікації стандарту OPC, які функціонують сьогодні, та їх призначення.
3. Якої послідовності необхідно дотримуватись, щоб забезпечити доступ у програмі OPC-Клієнта до змінної в контролері через OPC-Сервер?
4. Як визначаються дані в межах OPC-Серверу та їх джерело? Як ідентифікуються ці дані OPC-Клієнтом?
5. Як OPC-Клієнт ідентифікує необхідний OPC-Сервер?

6. Яка програма може називатись OPC-Клієнтом, а яка — OPC-Сервером? Чи може одна і та сама програма бути одночасно і OPC-Клієнтом, і OPC-Сервером? Поясніть мультисерверність та мультиклієнтність OPC.
7. Розкажіть, як відбувається доступ до даних через об'єкти OPC-Item. Чи може надаватися доступ OPC-Клієнтам до одних і тих самих даних OPC-Серверу? Чи можуть декілька OPC-Клієнтів користуватися одним і тим самим OPC-Item?
8. Яка інформація про дані надається через об'єкт OPC-Item?
9. Розкажіть про призначення об'єктів OPC-Group. Які групові операції для об'єктів OPC-Item проводяться через OPC-Group.
10. Які способи читання з джерела даних доступні в OPC-DA 2.0? Як вони функціонують? Яке з них, на Вашу думку, найбільш підходить до опитування даних процесу в засобах SCADA/HMI?
11. Які способи запису даних доступні в OPC-DA 2.0? Як вони функціонують? Яке з них, на Вашу думку, найбільше підходить до супервізорної зміни даних у засобах SCADA/HMI?
12. Як ідентифікуються дані в OPC-Сервері? Які правила створення імен ідентифікаторів, визначені стандартами OPC?
13. Розкажіть про необхідність використання об'єкта OPCBrowser? Які два види структури імен ItemID можуть бути доступні в OPC-Сервері?
14. Які вимоги до мережної системи для можливості зв'язку OPC-Клієнта з віддаленим OPC-Сервером? Що додатково необхідно вказати в OPC-Клієнті для ідентифікації OPC-Серверу, з яким необхідно з'єднатись?
15. Які типи OPC-Інтерфейсів визначені в стандартах OPC DA? Розкажіть, в яких випадках використовується кожний із типів. Навіщо потрібна бібліотека OPC Wrapper і як її використовують?
16. Перерахуйте основні об'єкти OPC DA Automation інтерфейсів та їх призначення. Яка послідовність дій у програмі необхідна для організації доступу до даних для читання/запису?
17. Наведіть приклади використання OPC-технології.

СТАНДАРТНІ ТЕХНОЛОГІЇ ДОСТУПУ ДО БАЗ ДАНИХ

14.1. Проблеми доступу до баз даних

В інтегрованих автоматизованих системах управління бази даних використовуються, як правило, для ведення історії подій та збереження значень даних для трендів. У цьому контексті можна виділити два способи доступу до даних: для запису та для читання. В ІАС, як правило, доступуються до архівних баз даних такі прикладні програми:

- SCADA-програми — для запису поточних даних та читання історичних;
- програми MES-систем — для читання історичних даних рівня АСУТП та запису агрегованих показників;
- програми ERP-систем — для читання та запису організаційно-економічних даних;
- службові програми та СУБД.

Дані в архів зберігаються, як правило, в тому форматі, який пропонує прикладна програма (наприклад, SCADA). Тобто тип архіву та формат записуваних даних диктуються вимогами цього програмного засобу або вибираються з ряду стандартних. Рідше є можливість в корегуванні розміщення, типу та формату бази даних. Тому можна умовно допустити, що при записі даних проблеми з сумісністю не виникають.

Інша ситуація виникає при доступу до даних для читання. Якщо програмні засоби, які пишуть дані і які їх читають, підтримують різні формати, то виникає проблема, пов'язана із сумісністю форматів, а, отже, і з доступом до даних для читання. Очевидно, що одне з рішень, яке може бути запропоновано, — це вибір програмних засобів, які підтримують єдиний формат доступу до баз даних. Однак, як правило, вибір програмних засобів часто виступає в якості обмеження при побудові систем, особливо, коли останні впроваджуються поетапно. Крім того, такий підхід не дає гнучкості при реалізації системи: адже обмеження диктуються вибором засобів одного виробника. Інше рішення — це використання додаткового спеціалізованого програмного забезпечення, яке перетворює дані з

одного формату в інший, або написання додаткових бібліотек. Такий підхід дорогий і потребує тривалої апробації.

Найбільш простим рішенням, яке, як правило, доступне в сучасних програмних засобах ІАСУ, є використання стандартних технологій доступу до баз даних. Серед них можна виділити використання СУБД з підтримкою стандартної мови запитів SQL, а також стандартних інтерфейсів доступу до баз даних ODBC та OLEDB. Крім наведених технологій є й інші відомі і можливо більш прогресивні рішення, однак сьогодні в системах ІАСУ вони поки що не знайшли великої популярності.

14.2. Мова SQL

SQL (Structured Query Language — мова структурованих запитів) — це універсальна мова для створення, модифікації і управління даними в реляційних базах даних. Дана мова описана в стандарті ANSI 1992 року як **SQL2 (SQL-92)**. Сьогодні практично всі відомі СУБД підтримують даний стандарт, з деяким розширенням до нього для адаптації під свої формати даних та функціональні можливості серверів. Такі мови називають діалектами SQL. Так, наприклад, діалект СУБД MS SQL Server має назву Transact-SQL.

SQL використовується для створення структури бази даних, маніпуляції з даними, тобто їх вибірки і модифікації, та для їх адміністрування. Будь-яка операція з вибірки, модифікації, визначення або адміністрування виконується за допомогою *оператора (statement)* або *команди (command)* SQL.

Є дві можливості операцій з маніпуляції даними — *вибірка даних (data retrieval)* і *модифікація даних (data modification)*. Вибірка — це пошук необхідних даних, а модифікація означає добавлення, знищення або заміну даних. Операції з вибірки називають **SQL запитами (SQL queries)**. Вони проводять пошук у базі даних, найбільш ефективно вибирають необхідну інформацію і відображають її. У всіх запитах SQL використовується ключове слово SELECT. Операції з модифікації виконуються відповідно з використанням ключових слів INSERT, DELETE та UPDATE.

На практиці найчастіше доводиться робити вибірку даних, тому коротко розглянемо оператор SELECT. Спрощена його конструкція має вигляд:

SELECT — список_стовпчиків;

FROM таблиця[-ці];

[WHERE умови],

де слова в квадратних дужках [] — не обов'язкові.

У списку стовпчиків (полів записів) вказуються ті поля таблиці, які повертаються після обробки запиту. Список таблиць визначає, з яких таблиць необхідно проводити вибірку, а в умовах вказують умови для вибірки рядків.

Приклад 14.1. Бази даних. Формування SQL-запитів 1

Завдання. Записати запит для вибірки даних з поля (колонки) Value таблиці Group_1_1 бази даних Difuzija_1, які були записані після 19 вересня 2006 року. Дата запису міститься в полі Group_1_1.TDate.

Рішення.

```
SELECT Value,
FROM Difuzija_1.Group_1_1,
WHERE Group_1_1.TDate>'2006-09-19 00:00:00'.
```

Приклад 14.1

Приклад 14.2. Бази даних. Формування SQL-запитів 2

Завдання. Записати запит для вибірки даних з усіх полів таблиці Group_1_1 бази даних Difuzija_1, які були записані після 19 вересня 2006 року. Дата запису міститься в полі Group_1_1.TDate.

```
SELECT *,
FROM Difuzija_1.Group_1_1,
WHERE Group_1_1.TDate>'2006-09-19 00:00:00'.
```

Приклад 14.2

Повний синтаксис оператора SELECT має такий вигляд:

```
SELECT [ALL|DISTINCT] — список_стовпчиків,
FROM {таблиця | курсор}[,{таблиця | курсор}...],
[WHERE умови],
[GROUP BY стовпчик [, стовпчик...]],
[HAVING умови],
[ORDER BY {ім'я_стовпчика | список_вибору}[ASC|DESC] ... ]
```

У списку SELECT вказуються ті поля (стовпчики), які необхідно повернути запитом. Є можливість зробити деякі операції над полями перед відображенням: +, -, *, /.

Умови у WHERE задаються операторами порівняння (=, <, >, >=, <=, != або <>), логічними (AND, OR, NOT), визначення діапазону (BETWEEN і NOT BETWEEN) та ін.

Ключове слово ORDER BY дозволяє впорядковувати знайдені записи за вказаним стовпчиком за зростанням ASC або за спаданням DESC.

14.3. ODBC та DAO

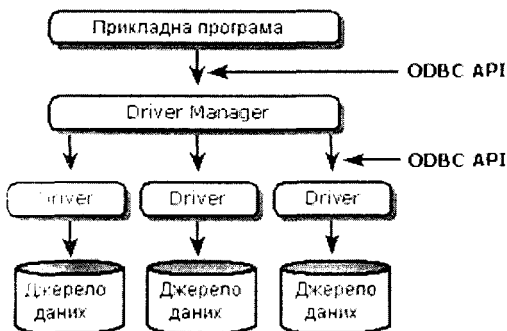


Рис. 14.1. Архітектура ODBC

ODBC (Open Database Connectivity) — один із стандартних інтерфейсів доступу до реляційних баз даних, які засновані на мові SQL. Архітектура ODBC показана на рис. 14.1.

Зв'язок з різними джерелами даних проводиться за допомогою Драйверів ODBC, які оформлені у вигляді динамічних бібліотек (DLL) і підтримують єдиний інтерфейс ODBC API, що базується на виконанні SQL-запитів. Тобто, незалежно від типу джерела даних, прикладна програма генерує SQL-запити, які відправляються пот-

рібному драйверу, що працює з даним джерелом. Драйвер забезпечує потрібне перетворення цих запитів у мову джерела або сам виконує необхідні операції, що вказані в запиті.

Для користувача даної технології необхідно зробити наступну послідовність операцій. В *Адміністраторі ODBC*, який міститься в панелі управління адміністратора Windows, створюється *DSN* (Data Source Name) для необхідного джерела даних:

- вибирається тип DSN: System DSN (доступний усім користувачам), User DSN (доступний тільки певному користувачу), File DSN (настройки зберігаються в окремому файлі);

- вказується ім'я DSN;

- вибирається драйвер ODBC;

- налаштовується драйвер для конкретного джерела даних.

У клієнтській прикладній програмі в якості джерела даних вказується ім'я DSN. З'єднання прикладної програми з необхідним джерелом даних забезпечиться Диспетчером Драйверів (Driver Manager), який реалізований у вигляді бібліотеки ODBC.DLL (рис. 14.1).

При написанні прикладних програм з використання ODBC більш зручний доступ до ODBC через RDO та DAO (рис. 14.2). *RDO* (Remote Data Object) — є об'єктним інтерфейсом доступу до ODBC джерел даних.

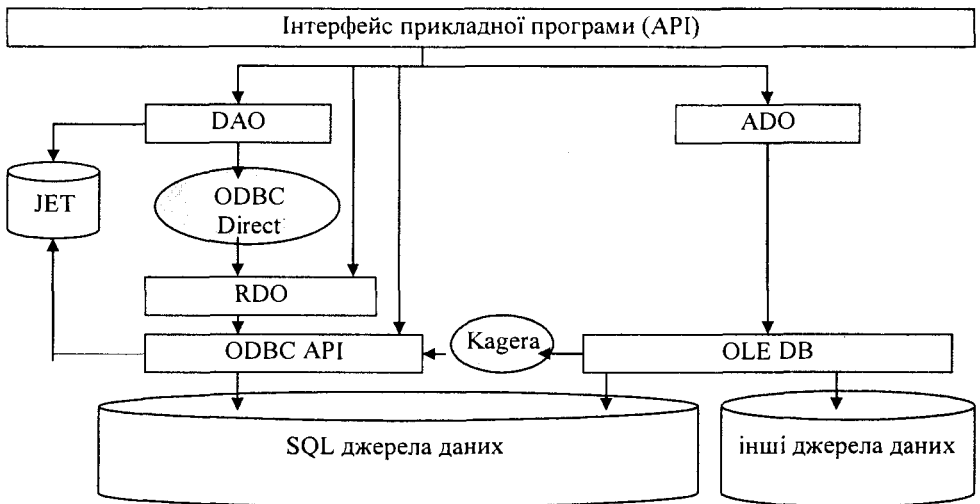


Рис. 14.2. Принцип доступу до джерел даних через стандартні інтерфейси

DAO (Data Access Object) є об'єктним COM-інтерфейсом до процесора баз даних *Jet* (Joint Engine Technology database engine), а також до надбудови над RDO-інтерфейсом.

Режим ODBCDirect перетворює всі об'єкти та методи DAO в еквівалент RDO.

Приклад 14.3. Бази даних. Доступ до змінних Citect через ODBC інтерфейс

Завдання. Використовуючи технологію ODBC, забезпечити періодичне відновлення даних у комірках листа Microsoft Excel, що посилаються на змінні з прикладу 13.3.

Рішення. SCADA Citect надає можливість доступу до бази даних реально-го часу через інтерфейс ODBC. При інсталяції SCADA на комп'ютер у списку драйверів ODBC з'явиться драйвер CitectDriver.

Для створення DSN викликаємо Адміністратор джерел даних: «Пуск»->«Настройка»->«Панель управління»->«Администрирование»->«Источники данных ODBC». На вкладці «Системные DSN» створюємо DSN типу CitectDriver.

Настроюємо DSN як на рис. 14.3. Якщо необхідно з'єднатися з Citect на іншому ПК, замість LOCALHOST вказується ім'я даного ПК.

У Microsoft Excel створити зв'язок із зовнішніми даними, використовуючи майстра зі створення SQL запитів Microsoft Query:

1. «Данные»->«Импорт внешних данных»-> «Создать запрос».

2. Вибрати джерело даних з іменем Citect.

3. Вибрати стовбці NAME, VALUE.

4. На відповідній вкладці вибрати «Просмотр или изменение в Microsoft Query», а потім натиснути кнопку SQL, для перегляду створеного майстром запиту (рис. 14.4).

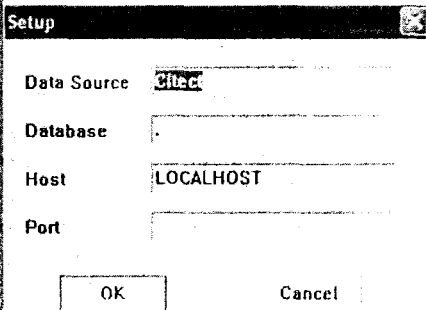


Рис. 14.3. Зовнішній вигляд вікна настройки DSN для джерела даних CitectDriver

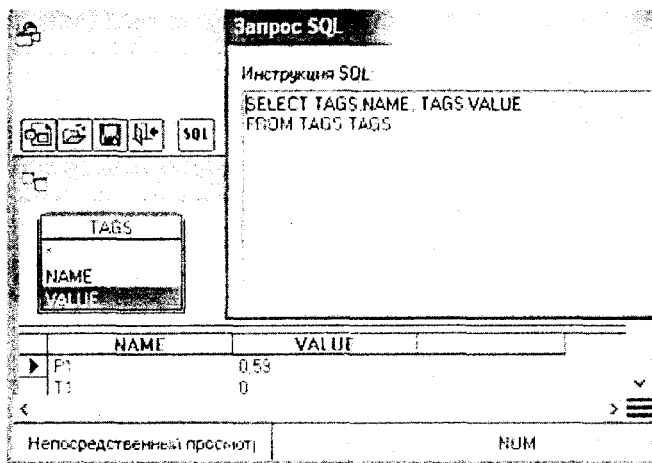


Рис. 14.4. Зовнішній вигляд вікна настройки запитів у Microsoft Query

5. Після закриття Microsoft Query, вказати ячейки, де будуть розмішуватися дані.

Для автоматичного періодичного відновлення даних у контекстному меню комірок з імпортованими даними вибираємо «Свойства диапазона данных»->«Обновлять каждые»->«1 мин».

Приклад 14.3

14.4. OLE DB, ADO та ADO.NET

Альтернативою ODBC є **OLE DB** — це об'єктно орієнтована технологія доступу до даних, яка базується на COM. Спеціалістами Microsoft запропонована стратегія універсального доступу до будь-яких джерел даних **UDA** (Universal Data Access), де драйвери OLE DB займають нижній щабель ієрархії (рис. 14.5). У цій архітектурі виділяються **OLE DB Consumers** (**OLE DB Споживачі**) — це будь-яка частина прикладної програми, яка *користується* OLE DB-інтерфейсами та **OLE DB Providers** (**OLE DB Провайдери**) — це частина прикладної програми, яка *надає* свої послуги через OLE DB-інтерфейси.

Розрізняють два види OLE DB Провайдерів: **OLE DB Провайдер Даних** (**OLE DB Data Provider**), **OLE DB Провайдер Сервісів** (**OLE DB Service Provider**). OLE DB Провайдер Даних надає послуги доступу до власних баз даних у вигляді таблиць. Провайдер Сервісів не має власних даних, однак, як Провайдер, він надає набір послуг (сервісів). Споживачі користуються цими сервісами, частина з яких використовує Провайдерів Даних для доступу до даних. Тобто Провайдер Сервісів щодо Провайдерів Даних виступає як Споживач.

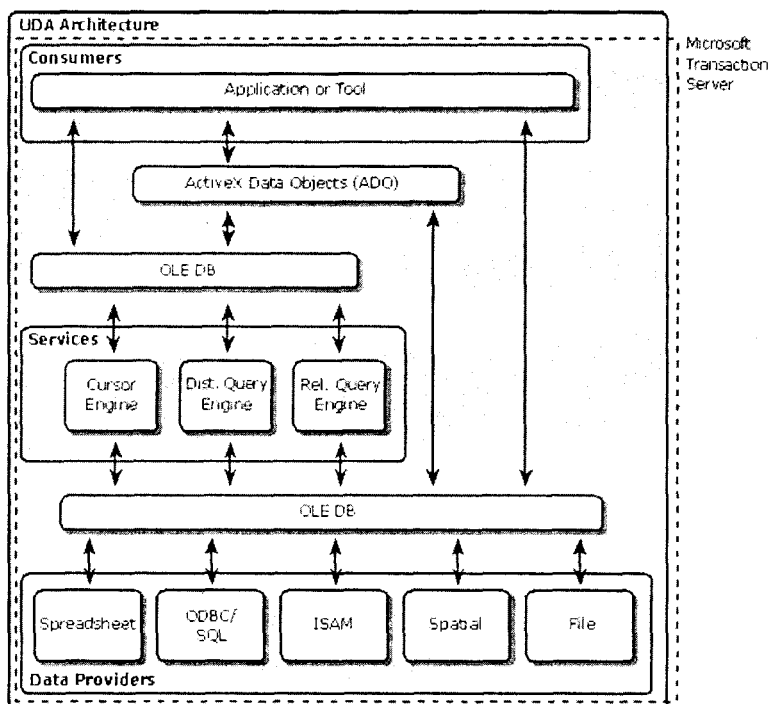


Рис. 14.5. UDA Архітектура

На відміну від ODBC, який надає доступ тільки до реляційних даних, OLE DB може надати доступ до будь-якого джерела даних, оскільки базується на об'єктній моделі, яка спроектована для представлення даних довільного формату. Для сумісності з розробленими ODBC драйверами OLE DB має інтерфейс доступу до них через Провайдера ODBC. Інтерфейс OLE DB складний у реалізації і призначений для розробки драйверів для нового типу джерела даних. Для кінцевого користувача необхідно мати зручний і легко зрозумілий прикладний інтерфейс (API), яким служить технологія ADO.

ADO (ActiveX Data Object) визначає модель програмування — послідовність дій, які необхідні для отримання доступу та модифікації джерела даних. Це об'єктна модель, що являє собою набір об'єктів зі своїми методами, властивостями та подіями.

Модель базується на таких ключових поняттях.

1. **Connection (Підключення)** — це джерело даних, з яким необхідно з'єднатися, може бути представлений у вигляді символічного **Рядка Підключення (Connection String)** або **Uniform Resource Locator (URL)**. У рядку вказується вся послідовність (транзакція) підключення, частини (кроки) якої розділені спеціальними роздільниками. В Рядку Підключення спочатку вказується Провайдер OLE DB, а потім усі інші параметри, які відрізняються залежно від Провайдера. Слід зазначити, що транзакція виконується повністю або не виконується взагалі, тобто виконується підключення до частини, вказаної в Рядку Підключення неможливе.

2. **Command (Команда)** — вказує операцію, яку необхідно виконати з джерелом даних, визначених Підключенням (добавити, знищити, модифікувати, знайти дані за заданими параметрами). Розрізняють декілька типів команд:

- за замовчуванням: тип команди визначається самим Провайдером;
- текст (SQL) : команда записується у вигляді SQL-виразу;
- таблиця: команда являє собою назву таблиці, всі колонки якої повертаються із запитом;

– збережена процедура (stored procedure) — назва процедури, яка повинна бути викликана;

– файл: вказується ім'я файла, з яким необхідно з'єднатися.

3. **Parameters (Параметри)** — це параметри команди, які можуть змінюватися. Команда може виступати як функція, тобто частина команди незмінна, а інша частина виступає в якості параметрів (параметричні запити).

4. **Recordset** — це об'єкт, за допомогою якого можна доступитись до рядків таблиці, які повертає Команда-запит.

5. **Field (Поле)** — об'єкт, який надає доступ до полів об'єкта Recordset.

6. **Error (Помилка)** — об'єкт, який містить інформацію про помилку. Ці об'єкти асоціюються з кожним Підключенням.

7. **Property (Властивість)** — кожний ADO-об'єкт має набір динамічних та статичних властивостей.

8. **Record (Запис)** — це об'єкт, який надає доступ до даних у вигляді **контейнера (container)** та **місткості (content)**. Для прикладу, у файльовій системі каталоги — це контейнери, які можуть вміщувати інші каталоги (контейнери)

або файли (місткість). Цей об'єкт дає можливість працювати з нереляційними даними.

9. **Stream (Потік)** — це об'єкт, за допомогою якого можна працювати з потоками байтів файлів чи буферів пам'яті (дані content).

10. **Collection (Колекція)** — це об'єкт, який вміщує декілька об'єктів одного типу, до яких можна доступитися за іменем чи індексом.

11. **Event (Подія).**

Таким чином, для доступу до даних за допомогою ADO та їх модифікації необхідно провести таку послідовність:

1. Указати джерело даних для Підключення, тобто його розміщення в Рядку Підключення або в URL. Виконати з'єднання з указаним джерелом.

2. Визначити Команду для доступу до джерела даних, її тип та за необхідності параметри. Виконати команду.

3. Для реляційних баз даних результати виконання команди у вигляді таблиці записуються в кеш. За необхідності їх можна модифікувати за допомогою об'єкта Recordset.

Технологія ADO в якості API інтерфейсу OLE DB надає стандартні сервіси та методи роботи з даними незалежно від Провайдера даних. Однак, у зв'язку зі специфікою останніх, можливі деякі особливості при їх використанні. Відмінності, як правило, стосуються Рядка Підключення (Connection String), використання Команди та об'єкта Recordset. Зупинимося на основних правилах формування Рядка Підключення та Команди.

Рядок Підключення (Connection String) формується з послідовності виразів типу *аргумент = значення*, розділених крапкою з комою. Типові аргументи наведені в табл. 14.1.

Таблиця 14.1

ТИПОВІ АРГУМЕНТИ ADO

Аргумент	Пояснення
<i>Provider=</i>	Назва Провайдера даних
<i>File Name=</i>	Назва файла, де зберігаються наперед визначені настройки Підключення
<i>Remote Provider=</i>	Назва віддаленого Провайдера (тільки для Remote Data Service)
<i>Remote Server=</i>	Шлях до віддаленого серверу (тільки для Remote Data Service)
<i>URL=</i>	URL-шлях до файла чи папки

Команда (Command) вказується за допомогою текстового рядка. Правила його формування залежать від типу команди. Зазвичай — це команда SQL, діалект якої залежить від Провайдера даних, однак це може бути інший текст.

Сьогоднішні найбільш використовувані Провайдери Даних: Microsoft OLE DB Provider for ODBC, OLE DB Provider for Microsoft Jet, Microsoft OLE DB Provider for SQL Server. Для більш детальної інформації з викладеного матеріалу та використання описаних технологій в інструментах програмування можна звернутися до довідкової інформації в MSDN.

Все більшого використання набуває технологія .NET, яка стала основою стандартного механізму доступу до даних під назвою *ADO.NET*. Аналогічно як ADO базується на OLE DB, в основі ADO.NET лежить .NET Framework. Принципи застосування залишилися такі самі. На момент написання книги відомі такі провайдери від Microsoft:

- Провайдер даних .NET Framework для SQL-серверу;
- Провайдер даних .NET Framework для OLE DB;
- Провайдер даних .NET Framework для ODBC;
- Провайдер даних .NET Framework для Oracle.

Як видно зі списку, технологія ADO.NET підтримує сумісність із OLE DB та ODBC.



Контрольні запитання до розділу 14

1. Які програмні засоби і навіщо доступуються до архівних даних? Які проблеми сумісності можуть при цьому виникати? Які варіанти розв'язання цих проблем?
2. Як вирішується питання стандартизації інтерфейсу до СУБД через використання мови SQL? Що таке діалект SQL?
3. Поясніть синтаксис оператора вибірки SELECT?
4. На яких технологіях базується ODBC? Які складові архітектури ODBC? Поясніть призначення кожного компонента цієї структури.
5. Як ідентифікується джерело даних в ODBC?
6. Наведіть послідовність операцій, яку необхідно виконати користувачеві, щоб вказати джерела даних програмі-Клієнту ODBC.
7. Якими програмними інтерфейсами користуються для доступу до даних через ODBC з прикладної програми?
8. На основі якої технології базується OLE DB? Які недоліки в технології ODBC порівняно з OLE DB?
9. Поясніть призначення компонентів UDA-архітектури.
10. Яке призначення OLE DB Провайдерів Даних та Провайдерів Сервісів?
11. Яким прикладним інтерфейсом, зазвичай користуються при OLE DB?
12. Чи можна доступитись до джерел даних ODBC через інтерфейс OLE DB? Чи можна через інтерфейс ADO? Обґрунтуйте свою відповідь.
13. Поясніть призначення об'єктів Connection (Connection String), Command та Recordset у технології ADO? Яким чином формується Рядок Підключення?
14. Яка послідовність дій необхідна при налаштуванні OLE DB Споживача для доступу до даних за допомогою ADO?
15. На основі якої технології базується ADO.NET? Чи можна доступитись до джерел даних ODBC через інтерфейс ADO.NET? Чи можна доступитись до джерел даних ADO через інтерфейс ADO.NET? Обґрунтуйте свою відповідь.

ПРОЕКТУВАННЯ ІНТЕГРОВАНИХ АВТОМАТИЗОВАНИХ СИСТЕМ

15.1. Загальні підходи до проектування автоматизованих систем

15.1.1. ІАС як клас автоматизованих систем

Інтегровані автоматизовані системи належать до класу автоматизованих систем, на правила проектування яких поширюються вітчизняні (ДСТУ), міждержавні (ГОСТ) та міжнародні (ISO, IEC) стандарти. В даному розділі пропонується підхід до ІАС з урахуванням етапів розробки згідно з діючими стандартами. В першому підрозділі наведений життєвий цикл та відповідно вимоги до комплектності та змісту документації автоматизованих систем, згідно з діючими стандартами. На базі цих вимог у наступних підрозділах пропонуються деякі підходи та рекомендації щодо життєвого циклу та документації інтегрованих автоматизованих систем управління як підкласу автоматизованих систем.

15.1.2. Стадії створення автоматизованих систем

15.1.2.1. Стандарт на стадії створення АС. Життєвий цикл інтегрованої автоматизованої системи, як і будь-якої автоматизованої системи (надалі АС), включає сукупність взаємозв'язаних процесів створення та послідовної зміни станів системи від формування вихідних вимог до неї, введення в дію до кінця експлуатації та утилізації комплексу засобів автоматизації. При створенні нової або вдосконаленні існуючої АС проектні та інжинірингові організації керуються рядом нормативних документів (стандартами, робочими документами, методичними вказівками).

Процес створення автоматизованих систем регламентується стандартом *ГОСТ 34.601-90* «Автоматизированные системы. Стадии создания». Згідно з цим стандартом процес створення АС є сукупністю впорядкованих у часі взаємозалежних, об'єднаних у *стадії, етапів* робіт, виконання яких необхідне і достатне

для створення АС, яка відповідає заданим вимогам. Стадії та етапи створення автоматизованих систем виділяються як частини процесу створення для раціонального планування та організації робіт, які закінчуються заданим результатом.

15.1.2.2. Структура стадій створення АС. Склад та правила виконання робіт на стадіях та етапах визначають у відповідній документації організації, які беруть участь у створенні конкретних видів автоматизованих систем. У роботах зі створення автоматизованих систем можуть брати участь такі організації: замовник; розробник; постачальник; ген-проектувальник об'єкта автоматизації; проектувальники різних частин об'єкта автоматизації; будівельні, монтажні та налагоджувальні організації.

Згідно з цим же стандартом створення автоматизованих систем складається з таких стадій:

1. Формування вимог до АС.
2. Розробка концепції АС.
3. Технічне завдання.
4. Ескізний проект.
5. Технічний проект.
6. Робоча документація.
7. Введення в дію.
8. Супровід.

Перші три стадії належать до *передпроектних* робіт, стадії 4—6, а також частково 7 — до *проектних* робіт. Кожна стадія складається з етапів. Стадії та етапи, які виконують організації-учасники робіт зі створення АС, установлюються в договорах і технічному завданні на основі даного стандарту. Допускається виключати стадію «Ескізний проект» та окремі етапи робіт на всіх стадіях, об'єднувати стадії «Технічний проект» та «Робоча документація» в єдину стадію «Техно-робочий проект». Залежно від специфіки виготовлюваних АС та умов їх створення допускається виконувати окремі етапи робіт до завершення попередніх стадій, паралельне в часі виконання етапів робіт, включення нових їх етапів.

Нижче розглянемо етапи робіт на кожній стадії створення АС, згідно зі стандартом ГОСТ 34.601-90.

15.1.2.3. Стадія формування вимог до АС (ТР). Ця стадія складається з таких етапів:

1. Дослідження об'єкта та обумовлення необхідності створення АС.
2. Формування вимог до АС.
3. Оформлення звіту про виконану роботу та заявки на розробку АС (тактико-технічного завдання).

При дослідженні об'єкта збирають дані про об'єкт автоматизації, оцінюють якості функціонування об'єкта, виявляють проблеми, вирішення яких можливе засобами автоматизації. В кінці дослідження роблять техніко-економічну оцінку доцільності створення АС. При позитивному результаті проводять підготовку вихідних даних для формування вимог АС: характеристику об'єкта автоматизації, опис вимог до системи, обмеження допустимих затрат на розробку, введення в дію та експлуатацію, очікуваний ефект, умови створення і функціонування сис-

теми. Про виконанні роботи повідомляють у звіті. В кінці стадії формують заявку на розробку АС (тактико-технічне завдання).

15.1.2.4. Стадія розробки концепції АС (ПК). На даній стадії виконують такі етапи робіт:

1. Вивчення об'єкта.
2. Проведення науково-дослідних робіт.
3. Розробка варіантів концепції АС, яка задовольняє вимогам користувача.
4. Оформлення звіту про виконану роботу.

На даних етапах організація-розробник проводить детальне вивчення об'єкта автоматизації та необхідні науково-дослідні роботи (НДР), які пов'язані з пошуком шляхів та оцінкою можливості реалізації вимог. Після цього проводиться розробка альтернативних варіантів концепції створюваної АС та планів їх реалізації, оцінку необхідних ресурсів для їх реалізації та забезпечення функціонування, оцінку переваг та недоліків кожного варіанта, визначення порядку оцінки якості та умов прийомки системи, оцінку очікуваних ефектів. Усі роботи подаються в звіті про виконану роботу.

15.1.2.5. Технічне завдання (ТЗ). На даній стадії проводиться розробка, оформлення, узгодження та затвердження технічного завдання на АС у цілому та за необхідності — на його окремі частини.

15.1.2.6. Ескізний проект (ЕП). На даній стадії визначаються: функції АС: функції підсистем, їх цілі та ефекти; склад комплексів задач та окремих задач; концепція інформаційної бази, її укрупнена структура; функції системи управління базою даних; склад обчислювальної системи; функції та параметри основних програмних засобів. Після цього розробляють документацію в обсязі, необхідному для опису повної сукупності прийнятих проектних рішень та достатньому для подальшого виконання робіт зі створення АС. Види документів розробляються згідно з ГОСТ 34.201-89.

Згідно з прийнятими проектними рішеннями розробляють документацію в обсязі, яка достатня для виконання

15.1.2.6. Технічний проект (ТП). Дана стадія складається з таких етапів:

1. Розробка проектних рішень із системи та її частин.
2. Розробка документації на АС та її частини.
3. Розробка та оформлення документації на поставку виробів для комплектування АС та технічних вимог (технічних завдань) на їх розробку.
4. Розробка завдань на проектування в суміжних частинах проекту об'єкта автоматизації.

На першому етапі даної стадії розробляють загальні рішення із системи та її частин, функціонально-алгоритмічної структури системи, з функцій персоналу та організаційної структури, зі структури технічних засобів, з алгоритмів рішень задач та використовуваних мов, з організації та ведення інформаційної бази, системи класифікації та кодування інформації, з програмного забезпечення. Результати висвітлюють у документації згідно з ГОСТ 34.201-89.

На 3-му етапі проводять підготовку та оформлення документації на постановку вимог для комплектування АС, визначення технічних вимог та складання ТЗ на ті вироби, які не виготовляються серійно.

На 4-му етапі складають завдання на проектування в суміжних частинах проекту об'єкта автоматизації для проведення будівельних, електротехнічних, санітарно-технічних та інших підготовчих робіт, пов'язаних зі створенням АС.

15.1.2.7. Робоча документація (РД). Стадія складається з двох етапів:

1. Розробка робочої документації на систему та її частини.
2. Розробка або адаптація програмного забезпечення.

На першому етапі оформляють робочу документацію згідно із ГОСТ 34.201-89, узгоджують та затверджують. Робоча документація повинна вміщувати всі необхідні та достатні відомості для забезпечення виконання робіт із введення АС у дію та її експлуатацію, а також для підтримки рівня експлуатаційних характеристик (якості) системи відповідно з прийнятими проектними рішеннями.

На другому етапі розробляють програми та програмні засоби системи, вибір, адаптацію та (або) прив'язку програмних засобів, що мають придбати, виготовляють програмну документацію згідно в ГОСТ 19.101.

15.1.2.8. Введення в дію (ВД). Стадію виконують у 8 етапів:

1. Підготовка об'єкта автоматизації до вводу АС в дію.
2. Підготовка персоналу.
3. Комплектація АС виробами що поставляються.
4. Будівельно-монтажні роботи.
5. Пуско-налагоджувальні роботи.
6. Проведення попередніх випробувань.
7. Проведення дослідної експлуатації.
- 7.8. Проведення приймальних випробувань.

На першому етапі проводять роботи з організаційної підготовки об'єкта автоматизації до вводу АС у дію, в тому числі:

- реалізацію проектних рішень з організаційної структури АС;
- забезпечення підрозділів об'єкта управління інструктивно-методичними матеріалами;

– впровадження класифікаторів інформації.

На другому етапі після підготовки персоналу перевіряють також його здатність забезпечити функціонування АС.

На 3-му етапі забезпечують отримання комплектуючих засобів серійного та одиничного виробництва, матеріалів та монтажних виробів (програмними та технічними засобами, програмно-технічними комплексами, інформаційними виробами), проводять вхідний контроль їх якості.

На етапі будівельно-монтажних робіт проводять:

– виконання робіт з будівництва спеціалізованих будівель (приміщень) для розміщення технічних засобів та персоналу АС;

– створення кабельних каналів;

– виконання робіт з монтажу технічних засобів та ліній зв'язку;

– випробування змонтованих технічних засобів;

– передачу технічних засобів для проведення пуско-налагоджувальних робіт;

На етапі пуско-налагоджувальних робіт проводять:

– автономну наладку технічних та програмних засобів;

– завантаження інформації у базу даних та перевірку системи її введення;

– комплексну наладку всіх засобів системи.

На етапі проведення попередніх випробувань проводять:

- випробування АС на працездатність та відповідність технічному завданню згідно з програмою та методикою попередніх випробувань;
- усунення несправностей та внесення змін у документацію на АС, в тому числі експлуатаційну відповідно з протоколом випробувань;
- оформлення акту про приймання АС у дослідну експлуатацію.

На етапі проведення дослідної експлуатації проводять:

- попередню експлуатацію АС;
- аналіз результатів досвідної експлуатації АС;
- доробку (за необхідності) технічних засобів АС;
- оформлення акта про завершення досвідної експлуатації.

На етапі проведення приймальних випробувань проводять:

- випробування на відповідність технічному завданню згідно з програмою та методикою приймальних випробувань;
- аналіз результатів випробувань АС та усунення недоліків, виявлених при випробуваннях;
- оформлення акта про приймання АС у постійну експлуатацію.

15.1.2.9. Супровід (СП). Стадія проходить у два етапи:

1. Виконання робіт відповідно з гарантійними обов'язками.
2. Післягарантійне обслуговування.

На першому етапі усуваються недоліки, виявлені при експлуатації АС протягом установлених гарантійних термінів, вносяться необхідні зміни в документацію про АС.

На другому етапі проводять роботи з:

- аналізу функціонування системи;
- виявлення відхилень фактичних експлуатаційних характеристик АС від проектних значень;
- установлення причин відхилень;
- виявлення недоліків із забезпечення стабільності експлуатаційних характеристик АС;
- внесення необхідних змін у документацію про АС.

15.1.2.10. Допустимі варіанти стадійності. Відповідно до ГОСТ 34.601-90 п.2.2, допускається:

- виключати стадію «Ескізний проект»;
- виключати окремі етапи робіт на всіх стадіях;
- для невеликих об'єктів об'єднувати стадії «Технічний проект» та «Робоча документація» в одну стадію — «Техно-робочий проект».

Крім того, залежності від специфіки створюваних АС та умов їх створення допускається:

- виконувати окремі етапи робіт до завершення попередніх стадій;
- паралельне в часі виконання етапів робіт;
- включати нові етапи робіт.

15.1.3. Розробка документації на стадіях створення автоматизованих систем

15.1.3.1. Стандарт на види, тип та комплектність документів на АС. На стадіях життєвого циклу автоматизованої системи, зокрема при її створенні, розробляється ряд документів, які зводяться в певний вид документації. Правила створення документації регламентуються міждержавним стандартом *ГОСТ 34.201-89* «Виды, комплектность и обозначение документов при создании автоматизированных систем». Згідно з даним стандартом, *документація на автоматизовану систему* — комплекс взаємопов'язаних документів, в якому повністю описані всі рішення зі створення та функціонування системи, а також документи, які підтверджують відповідність системи вимогам технічного завдання та готовність її до експлуатації (функціонування). *Проектно-кошторисна документація на АС* — частина документації на АС, яка розробляється для виконання будівельних та монтажних робіт, які пов'язані зі створенням АС. *Робоча документація на АС* — частина документації на АС, яка необхідна для виготовлення, будівництва, монтажу та наладки автоматизованої системи в цілому, а також програмно-технічних, програмно-методичних комплексів та компонентів технічного, програмного та інформаційного забезпечення, що входять у систему.

15.1.3.2. Документи на види забезпечення. Створюють АС об'єкта управління за документами таких видів забезпечення:

- технічне забезпечення (ТО);
- інформаційне забезпечення (ІО);
- організаційне забезпечення (ОО);
- математичне забезпечення (МО);
- програмне забезпечення (ПО);
- алгоритмічне забезпечення (АО).

Технічне забезпечення АС (*ТО*) — сукупність технічних засобів системи та експлуатаційної документації, здатних забезпечити функціонування системи в повному обсязі.

Інформаційне забезпечення АС (*ІО*) — набір документів та даних, які вміщують:

- перелік і характеристики змінних, які відображають фактичний стан об'єкта управління та автоматизованої системи;
- опис правил класифікації та кодування інформації та її груп;
- опис масивів вхідної та вихідної інформації;
- форми документів, відеокадрів, які використовуються в системі.

Математичне забезпечення (*МО*) — сукупність методів, моделей та алгоритмів, які використовуються в системі. МО реалізується у вигляді програм спеціального ПО.

Програмне забезпечення АС (*ПО*) — сукупність програм та експлуатаційної програмної документації, необхідних для реалізації АС при використанні технічного забезпечення системи. *Загальне програмне забезпечення* — частина програмного забезпечення, яке:

- поставляється в комплексі із засобами обчислювальної техніки;
- можна придбати на ринку програмних продуктів.

Спеціальне програмне забезпечення — частина програмного забезпечення, яке розробляється спеціально при створенні даної системи для реалізації основних та додаткових функцій АС. Це програмне забезпечення створюється на базі загального ПО.

Організаційне забезпечення (ОО) — опис структур АС, інструкції оперативному персоналу, на основі яких персонал повинен функціонувати в складі автоматизованого технологічного комплексу.

Перелік найменувань документів, що розробляються, та їх комплектність на систему та її частини повинен бути визначений у технічному завданні на створення автоматизованої системи (підсистеми). Комплектність проектно-кошторисної документації визначають згідно з правилами, які встановлені системою проектної документації для будівництва (СПДС). На кожний комплект повинна бути складена відомість документів. Комплектність документації, яка забезпечує виготовлення, приймання та монтаж технічних засобів визначається *ГОСТ 2.102*, експлуатаційної документації на ці засоби — *ГОСТ 2.601*. Комплектність документації на програмні засоби обчислювальної техніки визначається *ГОСТ 19.101*.

15.1.3.3. Комплектність та види документів за стадіями створення АС. Види документів за стадіями створення АС, відповідно до типу забезпечення, зведені в таб. 15.1.

Таблиця 15.1

**ВИДИ ДОКУМЕНТІВ, ЩО СТВОРЮЮТЬСЯ
ПО СТАДІЯМ СТВОРЕННЯ**

Стадія	Найменування документа	Код	Частина проекту	Документація	
				Проектно-кошторисна	Експлуатаційна
	Обґрунтування інвестицій	ОИ			
1.ТР	Вимоги до АС, формування тактико-технічного завдання				
2.РК	Звіти про виконані роботи				
3.ТЗ	Технічне завдання	ТЗ			
4.ЕП	Відомість ескізного проекту	ЭП	ОР*	—	—
	Пояснювальна записка до ЕП	П1	ОР	—	—
4.ЕП.	Схема організаційної структури	СО	ОР	—	—
	Схема структурна комплексу технічних засобів	С1	ТО	+	—
	Схема функціональної структури	С2	ОР	—	—
4.ЕП.	Перелік завдань на розробку спеціалізованих технічних засобів	В9	ТО	+	—
5.ПІ	Схема автоматизації	С3	ТО	+	—
	Технічне завдання на розробку спеціалізованих технічних засобів	—	ТО	—	—
	Завдання на розробку будівельних, електротехнічних, санітарно-технічних та інших розділів проекту, пов'язаних зі створенням системи	—	ТО	—	—

Продовження табл. 15.1

Стадія	Найменування документа	Код	Частина проекту	Документація	
				Проектно-кошторисна	Експлуатаційна
5.ТІ	Відомість технічного проекту	ТП	ОР	—	—
	Відомість покупних виробів	ВП	ОР	—	—
	Перелік вхідних сигналів та даних	В1	ИО	—	—
	Перелік вихідних сигналів (документів)	В2	ИО	—	—
	Перелік завдань на розробку будівельних, електротехнічних, санітарно-технічних та інших розділів проекту, пов'язаних зі створенням системи	В3	ТО	+	—
	Пояснювальна записка до технічного проекту	П2	ОР	—	—
	Опис функцій, що автоматизуються	П3	ОР	—	—
	Опис постановки задач (комплексу задач)	П4	ОР	—	—
	Опис інформаційного забезпечення системи	П5	ИО	—	—
	Опис організації інформаційної бази	П6	ИО	—	—
	Опис систем класифікації та кодування	П7	ИО	—	—
	Опис масиву інформації	П8	ИО	—	—
	Опис комплексу технічних засобів	П9	ТО	—	—
	Опис програмного забезпечення	ПА	ПО	—	—
	Опис алгоритму (проектної процедури)	ПБ	МО	—	—
	Опис організаційної структури	ПВ	ОО	—	—
	План розміщення	С8	ТО	+	—
	Відомість обладнання та матеріалів	—	ТО	+	—
Локальний кошторисний розрахунок	Б2	ОР	+	—	
Проектна оцінка надійності системи	Б1	ОР	—	—	
5.ТІП	Креслення форми документа (відеокадру)	С9	ИО	—	+
6.РД	Відомість держателів оригіналу	ДП	ОР	—	—
6.РД	Відомість експлуатаційних документів	ЭД	ОР	—	+
	Специфікація обладнання	В4	ТО	+	—
	Відомість потреби в матеріалах	В5	ТО	+	—
	Відомість машинних носіїв інформації	ВМ	ИО	—	+
	Масив вхідних даних	В6	ИО	—	+
	Каталог бази даних	В7	ИО	—	+
	Склад вихідних даних (повідомлень)	В8	ИО	—	+
	Локальний кошторис	Б3	ОР	+	—
Методика (технологія) автоматизованого проектування	И1	ОО	—	+	

Закінчення табл. 15.1

Стадія	Найменування документа	Код	Частина проекту	Документація	
				Проектно-кошторисна	Експлуатаційна
6.РД	Технологічна інструкція	И2	ОО	—	+
	Інструкція користувача	И3	ОО	—	+
	Інструкція з формування та ведення бази даних	И4	ИО	—	+
	Інструкція з експлуатації	ИЭ	ТО	—	+
	Схема з'єднань зовнішніх проводок	С4	ТО	+	—
	Схема підключення зовнішніх проводок	С5	ТО	+	—
	Таблиця з'єднань та підключень	С6	ТО	+	—
	Схема поділу системи (структурна)	Е1	ТО	—	—
	Креслення загального вигляду	В0	ТО	+	—
	Креслення установки технічних засобів	СА	ТО	+	—
	Схема принципова	СБ	ТО	+	—
	Схема структурна комплексу технічних засобів	С1	ТО	+	—
	План розміщення обладнання та проводок	С7	ТО	+	—
	Опис технологічного процесу обробки даних (включаючи телеобробку)	ПГ	ОО	—	+
	Загальний опис системи	ПД	ОР	—	+
	Програма та методика випробувань (компонентів, комплексів засобів автоматизації, підсистем, систем)	ПМ	ОР	—	—
Формуляр	ФО	ОР	—	+	
Паспорт	ПС	ОР	—	+	
7.ВД	<ul style="list-style-type: none"> – акт закінчення робіт; – акт приймання в дослідну експлуатацію; – акт приймання в промислову експлуатацію; – план-графік робіт; – наказ про склад приймальної комісії; – наказ про проведення робіт; – програма робіт; – протокол досліджень; – протокол узгодження 				

*ОР — загальносистемні рішення

Залежно від методів проектування та специфіки АС, що створюються, дозволяється: розробляти групові та базові документи відповідно з розділами 1, 3, 4, 6 ГОСТ 2.113; випускати документи окремими самостійними частинами, які відповідають розділам основного документа; розширяти номенклатуру документів, які встановлені стандартом ГОСТ 34.201-89.

Для невеликих систем використовують одностадійне проектування, коли стадія «Ескізний проект» виключається, а стадія «Технічний проект» та «Робоча документація» об'єднуються в одну стадію — «Техноробочий проект». Згідно з таким підходом, «Техноробочий проект» включає частину, яка затверджується (замість «Технічного проекту») і безпосередньо «Робочий проект».

Вимоги до складу та змісту документації на АС наведені в методичних вказівках *РД 50-34.698-90*. В табл. 15.2-15.6 наведений склад документів техноробочого проекту.

15.1.3.4. Загальносистемні рішення (ОР).

Таблиця 15.2

СКЛАД ДОКУМЕНТІВ ІЗ ЗАГАЛЬНОСИСТЕМНИХ РІШЕНЬ

ЗАГАЛЬНОСИСТЕМНІ РІШЕННЯ	
Відомість (склад) проекту (П1).	
Пояснювальна записка до проекту (П2).	Загальні положення. Опис процесу діяльності. Основні технічні рішення. Заходи щодо підготовки та вводу системи в дію.
Схема функціональної структури (С2).	
Відомість покупних виробів (ВП).	
Опис функцій, що автоматизуються (ПЗ).	Вихідні дані. Цілі АС та автоматизовані функції. Характеристика функціональної структури. Типові рішення.
Опис постановки задачі (комплексу задач) (П4).	Характеристики комплексу задач. Вихідна інформація. Вхідна інформація.
Локальний кошторис (Б3) та локальний кошторисний розрахунок (Б2).	
Паспорт (ПС).	Загальні відомості про АС. Основні характеристики АС. Комплектність. Свідчення про приймання. Гарантії виробника. Відомості про рекламу.
Формуляр (ФО).	Загальні відомості. Основні характеристики. Комплектність. Свідчення про приймання. Гарантійні обов'язки. Відомості про приймання. Відомості про стан АС. Відомості про рекламу.
Проектна оцінка надійності системи (Б1).	Введення. Вихідні дані. Методика розрахунку. Розрахунок показників надійності. Аналіз результатів розрахунку.
Загальний опис системи (ПД).	Призначення системи. Опис системи. Опис взаємозв'язків АС з іншими системами. Опис підсистем.
Відомість тримачів оригіналів (ДП).	
Відомість експлуатаційних документів (ЕД).	
Програма та методика випробувань (ПМ).	
Схема організаційної структури (СО).	

15.1.3.5. Організаційне забезпеченню (ОО)

Таблиця 15.3

СКЛАД ДОКУМЕНТІВ З ОРГАНІЗАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ

ОРГАНІЗАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ.

Опис організаційної структури (ПВ).	Зміна в організаційній структурі управління об'єктом. Організація підрозділів. Організація наявних підрозділів управління.
Методика (технологія) автоматизованого проектування (И1).	Загальні положення. Постановка завдання. Методика проектування. Вихідні дані. Проектні процедури. Оцінка результатів.
Технологічна інструкція (И2).	
Інструкція користувача (И3).	Введення. Призначення та умови застосування. Підготовка до роботи. Опис операцій. Аварійні ситуації. Рекомендації до освоєння.
Опис технологічного процесу обробки даних (ПГ).	Технологічний процес збору та обробки даних на периферійних пристроях при децентралізованій обробці даних. Технологічний процес обробки даних на обчислювальному центрі.

15.1.3.6. Технічне забезпечення (ТО)

Таблиця 15.4

СКЛАД ДОКУМЕНТІВ З ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ

ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

Схема автоматизації (С3).	
Опис комплексу технічних засобів (П9).	Загальні положення. Структура комплексу технічних засобів. Засоби обчислювальної техніки. Апаратура передачі даних.
План розміщення обладнання та проводок на об'єкті (С7).	
План розміщення обладнання та проводок (С8).	
Технічне завдання на розробку спеціалізованих технічних засобів.	
Завдання на розробку будівельних, електротехнічних, санітарно-технічних та інших розділів проекту, пов'язаних зі створенням системи.	
Перелік завдань на розробку спеціалізованих технічних засобів.	
Перелік завдань на розробку будівельних, електротехнічних, санітарно-технічних та інших розділів проекту, пов'язаних зі створенням системи (В3).	
Схема структурна комплексу технічних засобів (С1).	
Схема з'єднань зовнішніх проводок (С4).	
Схема підключень зовнішніх проводок (С5).	

Закінчення таблиці 15.4

ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ	
Таблиця з'єднань та підключень (С6)	
Схема поділу системи (структурна) (Е1).	
Креслення загального вигляду щита (пульта) (В0).	
Креслення установки технічних засобів (СА).	
Схеми принципів (СБ).	
Специфікація обладнання (В4).	
Відомість потреб у матеріалі (В5).	
Інструкція з експлуатації комплексу технічних засобів (ІЭ).	Загальні вказівки. Заходи безпеки. Порядок роботи. Перевірка правильності функціонування. Вказівки про діяльність у реальних режимах.
Відомість обладнання та матеріалів.	

15.1.3.7. Інформаційне забезпечення (ІО).

Таблиця 15.5

СКЛАД ДОКУМЕНТІВ З ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ

ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ	
Перелік вхідних сигналів та даних (В1).	Перелік вхідних сигналів. Перелік вхідних даних.
Перелік вихідних сигналів (документів) (В2).	Перелік вихідних сигналів. Перелік вихідних документів.
Опис інформаційного забезпечення системи (П5).	Склад інформаційного забезпечення. Організація інформаційного забезпечення. Організація збору та передачі інформації. Побудова системи класифікації та кодування. Організація внутрішньомашинної інформаційної бази. Організація позамашиної інформаційної бази.
Відомість машинних носіїв інформації (ВМ).	
Опис організації інформаційної бази (П6).	Опис внутрішньомашинної інформаційної бази. Опис позамашиної інформаційної бази.
Опис систем класифікації та кодування (П7).	
Опис масиву інформації (П8).	
Креслення форми документа (відеокадру) (С9).	
Масив вхідних даних (В6).	
Каталог бази даних (В7).	
Склад вихідних даних (повідомлень) (В8).	
Інструкція з формування та ведення бази даних (набору даних)(І4):	Правила підготовки даних. Порядок та засоби заповнення бази даних. Процедури зміни та контролю бази даних. Порядок та засоби відновлення бази даних.

15.1.3.8. Програмне забезпечення (ПО)

Таблиця 15.6

СКЛАД ДОКУМЕНТІВ З ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ	
Опис програмного забезпечення (ПА).	Структура програмного забезпечення. Функції частин програмного забезпечення. Методи та засоби розробки програмного забезпечення. Операційна система. Засоби, які розширюють можливості операційної системи.

15.1.3.9. Математичне забезпечення (МО).

Таблиця 15.6

СКЛАД ДОКУМЕНТІВ З МАТЕМАТИЧНОГО ЗАБЕЗПЕЧЕННЯ

МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ.	
Опис алгоритму (проектної процедури) (ПБ).	Призначення та характеристика. Інформація, що використовується. Результати рішення. Математичний опис. Алгоритм рішення.

15.1.3.10. Допустимі варіанти комплектності. Перелік (комплектність) документів та їх зміст згідно зі стандартом визначається за узгодженістю сторін договору. Тобто, вітчизняні нормативні документи дають достатньо ступенів вибору для можливості добавлення та модифікації тих документів, які вважає за необхідне проектувальник, за умови погодження цього з замовником.

Існуючі стандарти не визначають механізми розробки проекту. Методика проектування залежить від багатьох факторів:

1. типу та масштабу об'єкта;
2. комплексу програмних засобів, що використовуються при проектуванні (САІР);
3. досвіду та вміння проектувальників;
4. кількості задіяних організацій та осіб.

Для ІАС, як підкласу АС, загальна послідовність проектних робіт співпадає з етапами згідно з ГОСТ 34.601-90, а комплектність документів — згідно з ГОСТ 34.201-89 та РД 50-34.698-90. Нижче запропоновані деякі механізми, які можна використати при проектуванні ІАС, зокрема при вирішенні специфічних для цих систем завдань.

15.2. Загальні підходи до проектування інтегрованих автоматизованих систем

15.2.1. Життєвий цикл ІАС

15.2.1.1. ІАС — система взаємопов'язаних АС різного типу. Інтегрована автоматизована система може бути складною ієрархічною системою і включати в себе декілька підсистем автоматизованого управління різними класами об'єктів.

Хоч ці підсистеми належать до класу автоматизованих систем, послідовність розробки, механізми та комплектність документації для них можуть значно відрізнятися між собою. Так, скажімо, розробленням підсистем АСУП та АСУТП для інтегрованих автоматизованих систем управління (ІАСУ), займаються люди різної кваліфікації, що значно ускладнює її створення однією організацією-підрядником. Необхідно також урахувати масштабність та складність таких робіт: адже ІАС може створюватися для всього підприємства. З наведеного можна зробити висновок, що життєвий цикл створення ІАС необхідно поділити на декілька частин.

15.2.1.2. Загальний підхід до розробки ІАС, запропонований у даному посібнику. По суті, створення ІАС — це створення автоматизованої системи шляхом інтеграції вже існуючих та/або розроблювальних підсистем. Враховуючи цю особливість, проектування ІАС пропонується проводити в двох площинах: на рівнях підсистем та системи в цілому. Тобто, пропонується відокремити процеси проектування автоматизованих систем, як самостійних частин, однак узгоджених на загальносистемному рівні. Такий підхід дає певні можливості:

- інтегрувати існуючі АС в ІАС, не змінюючи комплектність та склад їх документації;
- відокремити життєві цикли підсистем як незалежних між собою АС, що дає можливість проводити поетапне створення та введення в дію ІАС;
- залучати до робіт декілька організацій-субпідрядників, з визначеними межами та обов'язками їх робіт.

Відповідно до запропонованого підходу, життєвий цикл інтегрованої автоматизованої системи управління включає в себе життєві цикли всіх АС, що є її підсистемами, та процеси їх інтеграції в єдине ціле. Надалі *під аббревіатурою АС будемо розуміти ті АС, які планується інтегрувати як підсистему в складі єдиної ІАС.*

У даному посібнику під **процесом створення інтегрованої автоматизованої системи будемо розуміти сукупність взаємопов'язаних процесів функціональної, інформаційної, технічної та програмної інтеграції наявних або/та розроблювальних АС для досягнення заданої мети.** Результатом такого процесу є інтегрована автоматизована система (ІАС), яка, крім АС, включає додаткові засоби, а інколи підсистеми, які служать для досягнення загальносистемних цілей. ІАСУ можна розглядати як частковий випадок ІАС, що включає вертикальну інтеграцію рівнів АСУТП та АСУП та появу в результаті цього підсистем рівня MES.

При запропонованому підході створенням інтегрованої автоматизованої системи може займатись окрема організація, а сама система супроводжуватися окремим комплектом документації.

Сьогодні до проектних робіт організації залучаються на конкурсній (тендерній) основі. Враховуючи це, передпроектні роботи повинні фінансуватися замовником, за окремим договором. У будь-якому випадку, кожна стадія, а навіть ряд її етапів з розробки окремої системи може проводитись різними організаціями за окремими договорами. Організації, що запрошені до конкурсу на розробку АС/ІАС, будемо називати потенційними розробниками.

15.2.1.3. Послідовність розробки ІАС. Розглянемо, як створення ІАС взаємопов'язане з життєвими циклами всіх АС, що входять до неї (рис. 15.1). Передпроектні роботи проводяться паралельно організаціями-розробниками АС та ІАС.

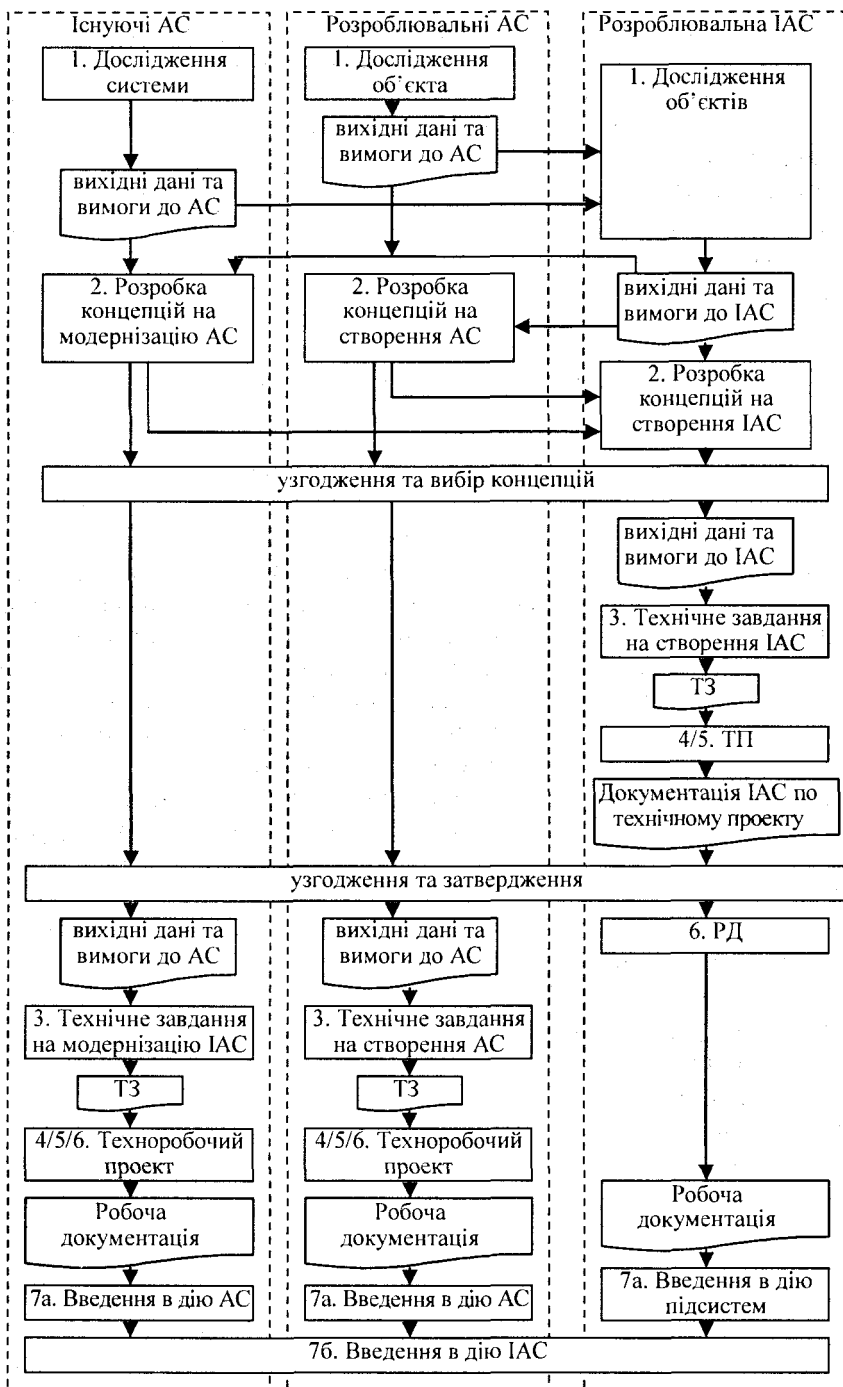


Рис. 15.1. Життєвий цикл ІАС

Серед АС окремо виділяються існуючі системи, які необхідно інтегрувати в єдину АС. Для неї пункти 3-7а є опціональними, оскільки їх інтеграція в єдину систему можлива без модернізації. Це рекомендована блок-схема життєвого циклу розробки ІАС. Схема на рис. 15.1 спрощена, оскільки на ній не показані ітерації. При розробці АС та ІАС різними організаціями ми рекомендуємо стадію технічне завдання на АС виконувати після узгодження технічного проекту ІАС. В іншому випадку — можлива неузгодженість рішень (навіть на рівні ТЗ) між АС та ІАС, що призведе до проблем в реалізації. Нижче більш детально розглянуті основні стадії розробки ІАС та їх взаємозв'язок зі стадіями АС.

15.2.2. Дослідження об'єкта та формування вимог до ІАС

На першій стадії замовник проводить дослідження об'єктів автоматизації та формує вимоги до них: характеристику об'єктів автоматизації, опис вимог до АС, обмеження допустимих затрат на розробку, введення в дію та експлуатацію, очікуваний ефект, умови створення і функціонування системи. Дослідження об'єкта може проводитись з участю інших компетентних організацій, наприклад потенційних розробників АС/ІАС, як правило, при фінансуванні замовника. Слід зазначити, що виділення АС у складі ІАС повинно бути вже на даному етапі.

У процесі дослідження об'єктів формуються вимоги до АС. Паралельно ведеться дослідження об'єктів на загальносистемному рівні. Потенційні розробники ІАС оцінюють результати досліджень АС та формують вимоги до ІАС. Загальний перелік функцій та вимог, поставлених на загальносистемному рівні, та вся необхідна інформація входять до складу вихідних даних на ІАС. Сформовані вихідні дані та вимоги оформлюють у довідній формі у вигляді тактико-технічного завдання на кожен АС та ІАС в цілому, яке вміщує:

- пропозицію організації-замовника до потенційної організації-розробника на проведення робіт зі створення АС/ІАС;
- вимоги замовника до АС/ІАС;
- умови та ресурси на створення АС/ІАС.

При дослідженні існуючих АС, що повинні входити до складу розроблювальної ІАС, необхідно звернути увагу на такі особливості:

- наявність сервісної підтримки даної АС та програмно-технічного комплексу, на базі якого вона розроблена;
- працездатність та надійність існуючої АС у цілому та окремих її складових;
- використання відкритих стандартів технологій зв'язку для можливого доступу до її інформаційних ресурсів;
- можливість нарощування або заміни апаратних та програмних складових.

Процес формування вихідних даних та вимог для АС/ІАС носить ітераційний характер. Після їх отримання потенційні організації-розробники перевіряють односторонність, узгодженість та цілісність вихідних матеріалів, що потребує уточнення вихідних даних замовником.

15.2.3. Розробка концепції ІАС

На даних етапах потенційні організації-розробники АС/ІАС за участі організації-замовника за необхідності проводять:

- детальне вивчення об'єкта автоматизації та необхідні науково-дослідні роботи (НДР);

- оцінку можливості реалізації вимог замовника.

Кожна з потенційних організацій-розробників АС/ІАС розробляє альтернативні варіанти концепції створюваної системи та планів їх реалізації, оцінює необхідні ресурси для їх реалізації та забезпечення функціонування, переваги та недоліки кожного варіанта, визначає порядок оцінки якості та умов приймання системи, оцінює очікувані ефекти, орієнтовну вартість системи.

Розмежування між областями проектування АС та ІАС проводиться вже на етапі формування вимог, однак потенційний розробник ІАС може взяти участь і в конкурсі на розробку АС. При аналізі концепцій кожної з організацій-конкурсантів враховується сумісність рішень з концепціями з розробки ІАС.

Вибір альтернативних програмно-технічних засобів для АС та ІАС можна проводити за наступними критеріями:

- взаємна апаратна та програмна сумісність засобів АС;
- технічний рівень програмного/технічного засобу;
- рівень забезпечення необхідної надійності;
- рівень повноти програмних засобів та простота конфігурування;
- ступінь захисту від проникнення в систему;
- досвід використання даного обладнання на аналогічних об'єктах;
- рівень довіри до постачальника обладнання та програмного забезпечення;
- здатність постачальника обладнання взяти на себе роль розробника;
- адекватність ціни та запропонованих засобів або виконуваних послуг.

15.2.4. Технічне завдання на ІАС та АС

15.2.4.1. Порядок розробки ТЗ. Результатом виконання попередніх стадій є розробка та оформлення технічного завдання (ТЗ) на ІАС в цілому. Як видно з рис. 15.1, створюють ТЗ для АС тільки після узгодження технічного проекту на ІАС із замовником системи. Така послідовність диктується необхідністю узгодженості АС та ІАС на загальносистемному рівні: технічне завдання на АС у багатьох розділах містить інформацію, яка стосується стику АС з іншими системами, зокрема іншими АС. При виконанні всіх АС у складі ІАС однією організацією на стадії ТЗ може бути розроблений та затверджений єдиний документ технічного завдання.

Згідно з ГОСТ 34.602-89, проект ТЗ на АС подає розробник з участю замовника на основі технічних вимог. При конкурсній організації робіт варіанти ТЗ розглядаються замовником, який або вибирає кращий варіант, або на основі порівняльного аналізу готує із участю потенційного розробника АС кінцевий варі-

ант ТЗ. Будемо вважати, що на даному етапі вже визначені результати конкурсного відбору розробника ІАС, тому технічне завдання створюється саме ним.

При використанні запропонованого в даному посібнику підході, в першу чергу, розробляється проект технічного завдання на ІАС. Зміст більшості пунктів ТЗ залежить від особливостей об'єкта управління. Загальні рекомендації до ТЗ за кожним пунктом наводяться в ГОСТ 34.602-89.

15.2.4.2. Зміст технічного завдання. Згідно з *ГОСТ 34.602-89*, технічне завдання повинно складатися з таких розділів:

1. Загальні відомості.
 - 1.1. Повне найменування системи та її умовне позначення.
 - 1.2. Шифр теми або шифр (номер) договору.
 - 1.3. Найменування підприємств (об'єднань) розробника та замовника (користувача) системи та їх реквізити.
 - 1.4. Перелік документів, на основі яких створюється система, зокрема ким і коли затвердженні ці документи.
 - 1.5. Планові терміни початку та закінчення роботи зі створення системи.
 - 1.6. Відомості про джерела та порядок фінансування робіт.
 - 1.7. Порядок оформлення та пред'явлення замовнику результатів робіт зі створення системи (її частин), з виготовлення та наладки окремих засобів (технічних, програмних, інформаційних) та програмно-технічних комплексів системи.
2. Призначення та цілі створення (розвитку) системи.
 - 2.1. Призначення системи.
 - 2.2. Цілі створення системи.
3. Характеристика об'єкта автоматизації.
4. Вимоги до системи.
 - 4.1. Вимоги до системи в цілому:
 - 4.1.1. вимоги до структури та функціонування системи;
 - 4.1.2. вимоги до чисельності та кваліфікації персоналу системи та режиму його роботи;
 - 4.1.3. показники призначення;
 - 4.1.4. вимоги до надійності;
 - 4.1.5. вимоги безпеки;
 - 4.1.6. вимоги до ергономіки та технічної естетики;
 - 4.1.7. вимоги до транспортабельності для пересувних АС;
 - 4.1.8. вимоги до експлуатації, технічного обслуговування, ремонту та збереження компонентів системи;
 - 4.1.9. вимоги до захисту інформації від несанкціонованого доступу;
 - 4.1.10. вимоги до збереження інформації при аваріях;
 - 4.1.11. вимоги до захисту від впливу зовнішніх дій;
 - 4.1.12. вимоги до патентної чистоти;
 - 4.1.13. вимоги до стандартизації та уніфікації;
 - 4.1.14. додаткові вимоги.
 - 4.2. Вимоги до функцій (задач), що виконуються системою:
 - 4.2.1. перелік функцій, задач або їх комплексів (у тому числі що забезпечують взаємодію частин системи); функціональних підсистем;

- 4.2.2. часовий регламент реалізації кожних функції, задач (або комплексу задач);
 - 4.2.3. вимоги до якості реалізації кожної функції (задачі або комплексу задач), до форми представлення вихідної інформації, характеристики необхідної точності та часу виконання, вимоги одночасності виконання групи функцій, достовірності видачі результатів;
 - 4.2.4. перелік та критерії відмов для кожної функції, за якою задаються вимоги надійності.
- 4.3. Вимоги до видів забезпечення:
- 4.3.1. вимоги до математичного забезпечення;
 - 4.3.2. вимоги до інформаційного забезпечення;
 - 4.3.3. вимоги до лінгвістичного забезпечення;
 - 4.3.4. вимоги до програмного забезпечення;
 - 4.3.5. вимоги до технічного забезпечення;
 - 4.3.6. вимоги до метрологічного забезпечення;
 - 4.3.7. вимоги до організаційного забезпечення;
 - 4.3.8. вимоги до методичного забезпечення;
 - 4.3.9. вимоги до інших видів забезпечення.
5. Склад та зміст робіт зі створення (розвитку) системи.
 6. Порядок контролю та приймання системи.
 7. Вимоги до складу та змісту робіт з підготовки об'єкта автоматизації до вводу системи в дію.
 8. Вимоги до документування.
 9. Джерела розробки.
 10. Додатки (за необхідності).

15.2.4.3. Зв'язок ТЗ на АС з технічним проектом ІАС. Технічне завдання на АС пов'язане з технічним проектом ІАС в основному пунктами 4.1–4.3. Зокрема, це стосується таких підпунктів:

- 4.1.1 вимоги до структури та функціонування системи, зокрема: до переліку підсистем, їх призначення та основних характеристик; до кількості рівнів в ієрархії та ступеня централізації системи; до способів та засобів зв'язку для інформаційного обміну між компонентами системи; до характеристики взаємозв'язків АС із суміжними системами (ІАС), до її сумісності, в тому числі вказуються способи обміну інформацією (автоматично, пересилкою документів, по телефону і т.д.)
- 4.2.2 вимоги до функцій (задач), що виконуються системою;
- 4.3 вимоги до видів забезпечення;

Ці вимоги впливають з ряду документів, розроблених у технічному проекті ІАС, які є предметом наступних підрозділів.

15.2.5. Технічний проект ІАС

Рекомендована комплектність документації для технічного проекту співпадає із загальними вимогами до комплектності для автоматизованих систем, наведеної в таб. 1 та відповідно до неї таб. 2-табл. 6. У будь-якому випадку форма та зміст

цих документів визначається в документі ТЗ — у вимогах до документування. В посібнику пропонується послідовність розробки технічного проекту та, відповідно, документації, яка зображена у вигляді блок-схеми на рис. 15.2.

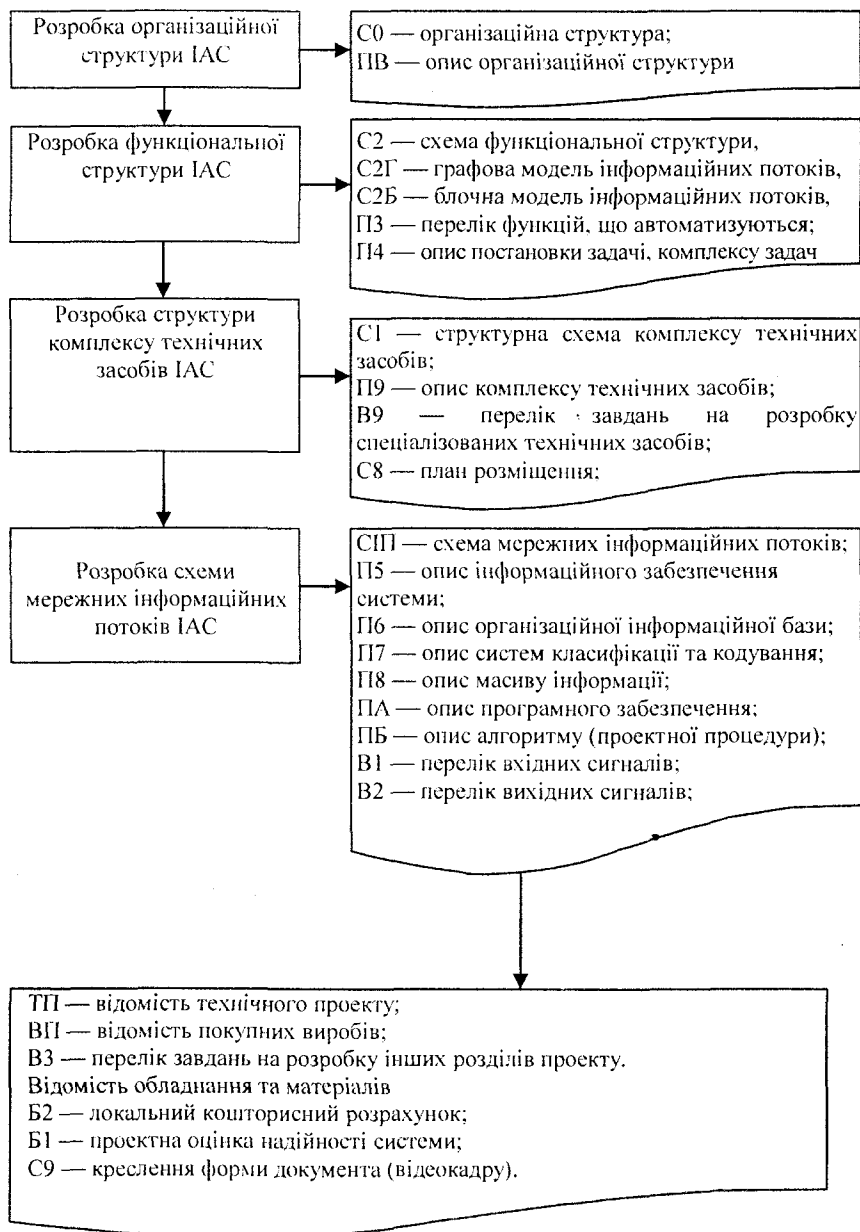


Рис. 15.2. Послідовність розробки та комплектність документації ІАС на стадії «Технічний проект»

15.2.6. Робоча документація та введення в дію

Після затвердження технічного проекту розробники ІАС працюють над робочою документацією (рис.15.1). Порядок та правила створення документів робочого проекту виходять за рамки даного посібника. Комплектність робочої документації на ІАС відповідає комплектності на АС, що наведена в табл. 15.1 і виконується аналогічно як для автоматизованих систем, зокрема АСУТП.

Нагадаємо, що згідно з підходом, запропонованим у даному посібнику, до проекту ІАС не включаються роботи над проектами АС (підсистемами) а тільки роботи з їх інтеграції. Таким чином, до проектування ІАС на стадії «робоча документація» належать роботи з проектування всієї мережної інфраструктури (програмне та технічне забезпечення серверів, робочих станцій тощо), що служать для вирішення завдань інтеграції АС в єдину систему. При створенні ІАСУ, для якого інтеграція АСУТП та АСУП приводить до появи рівня MES, на стадії «робоча документація» працюють над проектуванням саме цього рівня.

Введення ІАС в дію пропонується проводити двома етапами: введення в дію окремо всіх підсистем, та введення в дію всієї системи. Поетапний підхід дасть можливість зробити наладку систем організаціям-розробникам на локальному рівні АС. Практично, це означає, що кожна підсистема може бути запущена на дослідну експлуатацію незалежно одна від одної. Загальносистемні функції можна вводити в дію теж поетапно, залежно від введення в дію тих АС, які беруть участь у вирішенні цих функцій.

У наступному підрозділі розглянемо рекомендації зі створення структурних схем, які використовуються на стадії «Технічний проект» життєвого циклу ІАС.

15.3. Розробка структурних схем ІАС

15.3.1. Зміст структурних схем

15.3.1.1. Місце структурних схем у проектній документації. Після дослідження об'єкта на стадії «Технічний проект» необхідно визначити: функції ІАС; функції підсистем, їх цілі та ефекти; склад комплексів задач та окремих задач; концепцію інформаційної бази, її укрупненої структури; функції систем управління базою даних; склад обчислювальної системи; функції та параметри основних програмних засобів.

Методика виділення функцій систем та підсистем із загальних поставлених завдань залежить від об'єкта і виходить за рамки даного посібника. Результати даної стадії формалізуються у вигляді організаційної структури (C0), схеми функціональної структури (C2), структурної схеми комплексу технічних засобів (C1).

15.3.1.2. Схема організаційної структури (C0). Дана схема належить до загальносистемних рішень, однак опис її входить до складу організаційного забезпечення. Згідно з РД 50-34.698.90, схема та опис включають: склад підрозділів (посадових осіб) організації, які забезпечують функціонування автоматизованої системи або використовують при прийнятті рішень інформацію, яка отримана від

автоматизованої системи; основні функції та зв'язки між підрозділами та окремими посадовими особами, які вказані на схемі, та їх підлеглість.

Схема організаційної структури в початковому вигляді визначається ще на етапі формування тактико-технічного завдання. Надалі вона може бути, за необхідності, модифікована.

15.3.1.3. Схема функціональної структури (С2). Схема належить до загальносистемних рішень. Згідно з РД 50-34.698.90, вона включає:

– елементи функціональної структури ІАС (підсистем АС); автоматизовані функції та (або) задачі (комплекс задач); сукупність дій (операцій), які виконуються при реалізації автоматизованих функцій тільки технічними засобами (автоматично) або тільки людиною;

– інформаційні зв'язки між елементами та із зовнішнім середовищем і короткою вказівкою змісту повідомлень та (або) сигналів, які передаються за зв'язками і, за необхідності, зв'язки інших типів;

– деталізовані схеми частин функціональної структури (за необхідності).

15.3.1.4. Опис автоматизованих функцій (ПЗ). Функціональна структура повинна супроводжуватись описом автоматизованих функцій (ПЗ), який, згідно зі стандартом вміщує розділи:

1. Вихідні дані: перелік вихідних матеріалів та документів, які використовувались при розробці функціональної частини проекту системи; особливості об'єкта управління, які впливають на проектні рішення за автоматизованими функціями; дані про системи управління, які взаємопов'язані з розроблювальною ІАС і відомості про інформацію, якою вона повинна обмінюватися з абонентами та іншими системами; опис інформаційної моделі об'єкта разом з його системою управління.

2. Цілі ІАС та функції що автоматизуються: опис функцій, що автоматизуються, спрямованих на досягнення встановлених цілей.

3. Характеристику функціональної структури: перелік підсистем АС з указівкою функцій та (або) задач, які реалізуються в кожній підсистемі; опис процесу виконання функцій (за необхідності); необхідні пояснення до розділення автоматизованих функцій на дії (операції), які виконуються технічними засобами та людиною; вимоги до часового регламенту та характеристикам процесу реалізації автоматизованих функцій (точності, надійності і т.д.) та розв'язання задач.

4. Типові рішення: перелік типових рішень з указівкою функцій, завдань, комплексів задач, для виконання яких вони застосовуються.

15.3.1.5. Опис постановки задач (П4). Крім ПЗ, у комплектності документів загальносистемного забезпечення наводиться опис постановки задач (П4), який вміщує розділи:

1. Характеристика комплексу задач: призначення комплексу задач; перелік об'єктів, при управлінні якими вирішують комплекс задач; періодичність і тривалість рішення; умови, при яких закінчується розв'язання комплексу задач автоматизованим способом (за необхідності); зв'язки даного комплексу задач з іншими комплексами (задачами) ІАС; посади осіб та (або) найменувань підрозділів, які визначають умови та часові характеристики конкретного розв'язання задачі (якщо вони не визначені загальним алгоритмом функціонування системи);

розподіл дій між персоналом та технічними засобами при різних ситуаціях розв'язання комплексу задач;

2. Вихідна інформація: перелік вихідних повідомлень; перелік та опис структурних одиниць інформації вихідних повідомлень (показників, реквізитів та їх сукупності, сигналів управління), які мають самостійне смислове призначення, або посилання на документи з цими даними.

3. Вхідна інформація (аналогічно вихідній).

В описі кожного повідомлення або структурної одиниці інформації необхідно вказати: ідентифікатор повідомлення, найменування структурної одиниці, форму представлення повідомлення (документ, відеокадр, сигнал управління) та вимоги до неї; періодичність видачі (вихідна); терміни видачі та допустимі затримки рішення (вихідна); отримувачів та призначення вихідної інформації, джерела вхідної інформації; вимоги до точності та надійності розрахунку.

В описі постановки задач (П4) допускається давати у вигляді додатків ілюстраційний матеріал, таблиці або текст допоміжного характеру, а також документи, які мають самостійне позначення (креслення форм документів, опис масивів інформації, схеми і т.д.)

15.3.1.5. Структурна схема комплексу технічних засобів (СІ). Враховуючи необхідність інтеграції існуючих підсистем в єдину систему, функціональну структуру та завдання необхідно розробляти в комплексі з технічною структурою інтегрованої системи. Структурна схема комплексу технічних засобів (структурна схема КТС) є частиною ТО технічного проекту та робочої документації і вміщує склад комплексу і зв'язки між технічними засобами або групами технічних засобів, які об'єднані за певними логічними ознаками (наприклад, сумісне виконання окремих або декількох функцій, однакове призначення і т.д.).

При виконанні цих схем допускається: вказувати основні характеристики технічних засобів; представляти структуру комплексу ІАС (за необхідності) декількома схемами, першою з яких є укрупнена схема в цілому.

Для розробки схеми інформаційних потоків (СІП) структурна схема комплексу технічних засобів повинна містити інформацію про способи об'єднання вузлів між собою. Тобто, на кресленні необхідно показати інформацію про засоби та порти, через які кожен вузол підключається до загальної мережі. Таким чином будемо виділяти два рівні структурних схем КТС:

- узагальнена структурна схема КТС;
- деталізована структурна схема КТС.

Узагальнена структурна схема КТС являє собою укрупнену схему ІАС в цілому, яку треба скласти вже на стадії розробки концепції. Вона є частиною вихідних даних та вимог до ІАС, що входять до тактико-технічного завдання на ІАС. *Деталізована структурна схема КТС (СІ)* розробляється на стадії «Технічний проект» з урахуванням інформаційної структури. Вона може бути виконана для частини ІАС і містити інформацію про засоби та порти підключення вузлів до мережі.

15.3.1.6. Моделі інформаційних потоків як особливий вид функціональної структури. При розробці інтегрованих автоматизованих систем вже на початковій стадії технічного проекту необхідно визначитися з організаційною та фу-

нкціональною структурою, функціями та комплексом задач, а також узагальноною технічною структурою системи в цілому, в контексті засобів якої виконуються задачі. Враховуючи складність декомпозиції поставлених перед системою задач на функції та підзадачі з урахуванням інформаційних взаємозв'язків, пропонується в якості доповнення або альтернативи до функціональної структури використовувати такі графічні матеріали: графову модель інформаційних потоків (С2Г) або блочну модель інформаційних потоків (С2Б). В основі цих моделей — функціональна структура, інформаційні зв'язки якої є інформаційними потоками.

15.3.2. Використання графової моделі інформаційних потоків (С2Г)

Графова модель інформаційних потоків виступає у вигляді потокового орієнтованого графа. Вершинами її є реалізація функцій (процес розв'язання та отримані дані), а дугами — передача інформації (аргументів функцій — необхідні дані). На дугах указуються умовні номери інформаційних потоків, на вершинах — назву функцій, або умовне позначення (наприклад, числове). Напрямок дуг указує на напрямок передачі даних. Двосторонні дуги — об'єднання інформаційних потоків у протилежному напрямку, але з однаковими даними.

Для розшифровки позначень дуг та вершин до графічного представлення моделі додається текстова частина з поясненням до них. У поясненні вказуються дані, які передаються, параметри потоків (періодичність, особливості тощо). Форма та зміст текстової частини відповідає вимогам до ПЗ та П4.

Для забезпечення виділених при декомпозиції функцій необхідними даними можуть з'явитися додаткові функції, які теж необхідно показати. Найбільш примітивні функції, які є кінцевими вершинами, є функції вводу/виводу (І/О), тобто: опитування датчиків, управління виконавчими механізмами, передача/прийом даних по мережі. Ці функції умовно можна назвати інтерфейсними. Далі від цих функцій інформаційні потоки приймають інший (не цифровий вигляд), тому не показуються на графі. Перетворення цифрової інформації з одного формату в інший теж є додатковою функцією, тому виділяється як окрема вершина.

Показуються тільки ті функції та інформаційні потоки, які стосуються мережного зв'язку. Ці функції на графі часто об'єднуються в єдину вершину, а дуги вказують на об'єдані інформаційні потоки в один. Така операція можлива у випадку, коли інформаційні потоки, які об'єднуються, будуть мати однакові характеристики: напрямок, швидкість відновлення, спосіб можливої реалізації тощо.

При проектуванні бажано розпочати з детального графового представлення інформаційних потоків, що дозволить наочно представити поставлене завдання та допомогти в організації його розв'язання.

Приклад 15.1. Проектування. Використання графової моделі інформаційних потоків

Завдання. Аналіз наявної системи на цукровому заводі показав, що всі виробничі дільниці автоматизовані з використанням сучасної мікропроцесорної техніки. Узгодженість роботи дільниць досягається за рахунок управлінських

дій начальників змін та головного технолога. Реалізовані контури управління кожного підрозділу за якісними критеріями управління задовольняють поставлені вимоги. Організаційно-економічні відділи підприємства автоматизовані з використанням пакета ІС «Предприятие». Попри високий рівень автоматизації, є деякі проблеми:

1. За нештатних ситуацій на одній із дільниць виникають проблеми з управлінням потоком та підтримкою стабільної якості на суміжних дільницях, зокрема:

- при значному збільшенні відкачки соку з дифузійної дільниці помітна інерційність роботи фільтрів соку І сатурації;
- при значному зменшенні подачі соку на випарну станцію фільтри соку І сатурації та дифузійне відділення продовжують працювати з тією ж потужністю.

2. Існують джерела наднормативних витрат пари, які практично важко визначити.

3. Складно проаналізувати роботу всього підприємства в цілому через відсутність централізованого доступу до необхідних плинних та історичних даних.

4. Параметри управління не підстроюються під плинні якісні параметри сировини.

5. Відсутня інформація про середні, максимальні, мінімальні та загальні витрати основного потоку по всьому підприємству, що утрудняє визначення «вузьких» місць та причин неритмічності роботи заводу.

Для того, щоб розв'язати названі проблеми, вирішено на першому етапі реалізувати такі проектні завдання:

1. Забезпечити централізований збір інформації по всьому виробничому потоку з можливістю кореляції завдань з основних витратних параметрів;

2. Забезпечити координацію сокоочисного, дифузійного відділення та фільтрів соку І-ї сатурації.

3. Забезпечити агрегованою інформацією робочі станції організаційно-економічного рівня.

4. Забезпечити централізований доступ до архівних даних.

Для розв'язання цих завдань в першому варіанті запропонована узагальнена структура комплексу технічних засобів ІАСУ виробництвом цукру, яка зображена на рис. 15.3. Перелік технічних засобів наведений у табл. 15.7. Необхідно провести декомпозицію загальних задач та показати інформаційну взаємодію отриманих підфункцій у вигляді графової моделі, відповідно з поставленими вимогами.

Таблиця 15.7

СПЕЦИФІКАЦІЯ ТЕХНІЧНИХ ЗАСОБІВ

Позначення	Найменування	Характеристика
М1, М ТЕЦ, М БУХ	Локальні обчислювальні мережа відповідно виробництва, ТЕЦ та бухгалтерії	Ethernet 100 Base-TX
М2-М6	Локальні мережі управління технологічного відділення очистки	Modbus RTU на RS-485
М7-М9	Локальні мережі управління технологічних відділень відповідно щодо газової печі, дифузії та випарки	UniTelway на RS-485
М10	Локальна мережа управління тракту подачі буряків	Ethernet 100 Base-TX

Позначення	Найменування	Характеристика
M11	Локальна мережа управління тракту подачі буряків	Profibus DP
M12-M20	Локальні мережі управління продуктового відділення	UniTelway на RS-485
ТС ТЕЦ, ТС ВЦ	Технологічні сервери відповідно ТЕЦ та виробництва цукру	Intel Pentium 4 GHz, RAM 512Mb
ПК ГТЕХ, ПК ДКС, ПК ЛАБ, ПК НЗ	Персональний комп'ютер (ПК) головного технолога, диспетчерсько-координуючої станції, лабораторії та начальника зміни відповідно	Celeron 1,7 GHz, RAM 256Mb
ПК ГП, ПК ДИФ, ПК ВИП	ПК відділень газової печі, дифузії та випарної станції відповідно	Celeron 1,7 GHz, RAM 256Mb
ПК ЦФ, ПК ВА	ПК центрифуг та вакуум-апаратів продуктової дільниці відповідно	Celeron 1,7 GHz, RAM 256Mb
ПК ТПБ	ПК тракту подачі буряків	Celeron 1,7 GHz, RAM 256Mb
ПК САТ	ПК дільниці очистки	Celeron 1,7 GHz, RAM 256Mb
ПЛК ТПБ	Програмований логічний контролер тракту подачі буряків	Vipa 317SN
ПЛК ГП, ПЛК ДИФ, ПЛК ВИП	Програмований логічний контролер відділень газової печі, дифузії та випарки відповідно	TSX 57303
ПЛК ЦФ1-4, ПЛК ВА1-3	Програмований логічний контролер центрифуг та вакуум-апаратів продуктового відділення відповідно	TSX 3722
ПЛК САТ	Програмований логічний контролер відділення очистки	TSX 57303
ПЛК ФЛП-3	Програмований логічний контролер фільтрації відділення очистки	TWD LCAA 40DRF
ОП ФЛП-3	Операторська панель фільтрації відділення очистки	XBTN400
ОП ВА1-3, ОП ЦФ1-4	Операторська панель вакуум-апаратів та центрифуг продуктового відділення	XBTN400
ЧПР САТ	Частотний перетворювач відділення сатурації	ATV58
І/О ТБП1,2	Віддалені входи/виходи відповідних відділень	SM 153-4PF00

Рішення. Один із варіантів декомпозиції поставлених завдань та їх інформаційні зв'язки показані на рис. 15.4, де вершинами графа є:

1, 5 — опитування, збереження локальних даних, реалізація управляючих дій; 2 — координація роботи фільтрів; 3 — управління відкачкою соку з холодного дефекатора; 4 — управління станом комунікаційного обміну; 6 — аналіз стану та управління приводом; 7 — контроль та супервізорне управління технологічного процесу фільтрації; 8 — управління станом операторської панелі; 9 — контроль та супервізорне управління технологічних процесів дефекосатурації та фільтрації; 10 — контроль роботи приводу насоса соку на гарячий дефекатор і сатурації; 11, 13 — збереження даних; 12 — анімація технологічного процесу дифузії; 14 — збереження даних; 15 — ведення архіву; 16 — управління основним потоком; 17 — управління станом операторської панелі; 18 — відображення архівних даних.

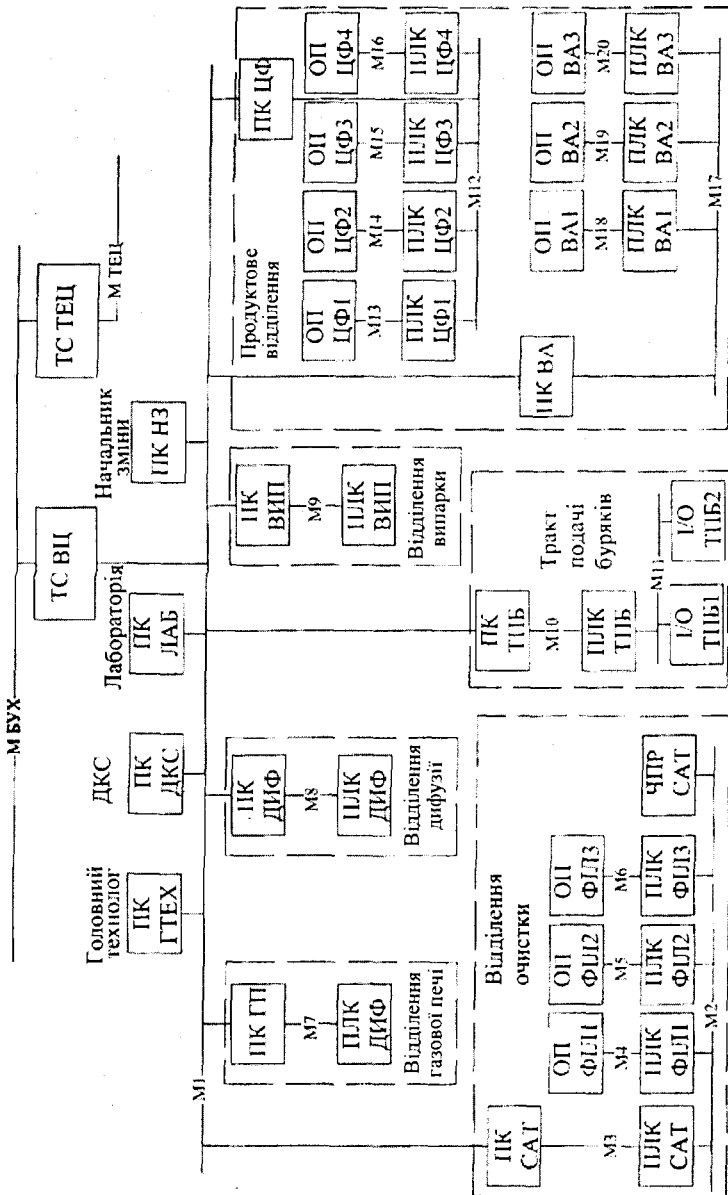


Рис. 15.3. Загальна структура комплексу технічних засобів ІАСУ виробництва цукру

Дуги графів представляють інформаційні потоки, зміст та вимоги до яких: 1 — параметри для відображення стану фільтрації: (йде перерахунок параметрів) — 1 раз/с; 2 — параметри, які відображають стан приводу насоса подачі соку на гарячий дефекатор: (йде перерахунок параметрів) — 1 раз/с; 3 — параметри для відображення стану, диспетчерського управління та зміни завдання дефекосатурації: (йде перерахунок параметрів) — 1 раз/с, за зміною значення; 4 — задана частота обертання, параметри управління приводом, задане значення ПІ-регулятора — 1 раз/с; 5 — команда на включення роботи приводу з комунікацією — при старті ПЛК; 6 — рівень у збірнику фільтрованого соку, рівень у збірнику нефільтрованого соку — якнайчастіше; 7 — F соку на дефекосатурацію, F соку на гар. дефекатор 1-ї сатурації, F соку на 2-гу сатурацію — за запитом; 8 — сторінки для відображення, стан кнопок, аварійна таблиця — 400 мс; 9 — параметри для відображення стану фільтрації, диспетчерського управління та зміни завдання: (йде перерахунок параметрів) — за запитом; 10, 12 — параметри для збереження: (йде перерахунок параметрів) — 1/1 хв. на весь сезон; 11 — параметри для читання (всі які пишуться) — за запитом; 13 — параметри для відображення стану і для диспетчерського управління та зміни завдання на відділення дифузії: (йде перерахунок параметрів, що відображають витратні характеристики та управляючі дії) 1/5 с; 14 — параметри для відображення стану і для диспетчерського управління та зміни завдання на відділення дефекосатурації: (йде перерахунок параметрів, що відображають витратні характеристики та управляючі дії) — 1/5 с.

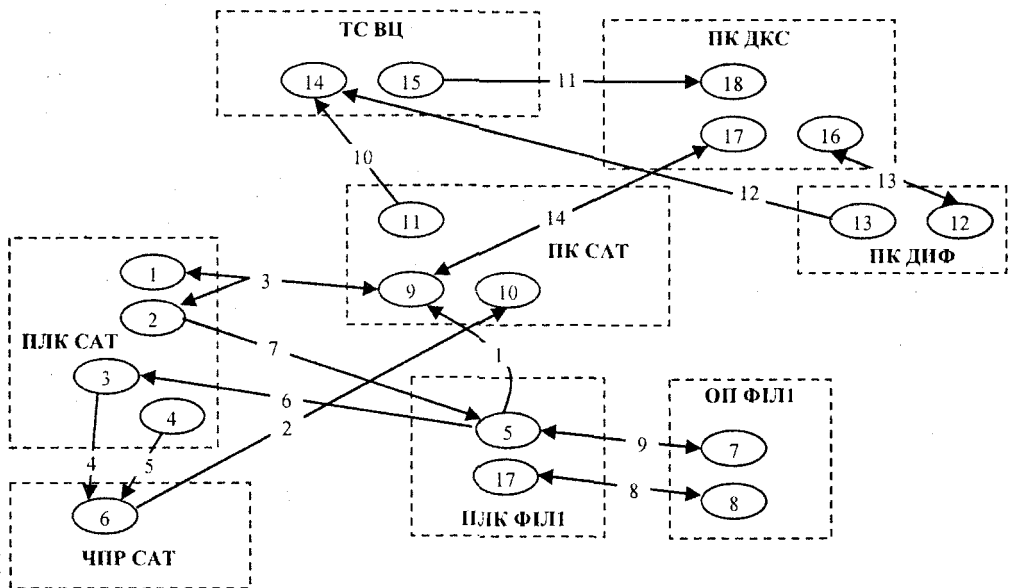


Рис. 15.4. Фрагмент графової моделі інформаційних потоків КІСУ

15.3.3. Використання блочної моделі інформаційних потоків (С2Б)

У блочній моделі інформаційних потоків кожна функція графічно відображається як функціональний блок, який «наповнений» даними — аргументами та результатами функцій. Зверху над функціональним блоком вказується назва функції. Дані функціонального блока можуть являти собою входи, виходи, входи/виходи та внутрішні (рис. 15.5). Вхідні дані — це аргументи функції, які беруться з інших функціональних блоків або з локальних джерел вводу даних (датчики, засоби ЛМІ, тощо). Дані, які є результатом даної функції, є вихідними даними. Це можуть бути входи функціональних блоків (функцій) інших вузлів або виконавчі механізми локального вузла. Деякі дані можуть модифікувати вхідні дані, в такому випадку дані є і вхідними, і вихідними. Інколи дані є проміжними результатами інших функцій даного вузла, які не треба показувати окремо на моделі. Якщо дані є локальними, то до них підключаються локальні прилади, які позначаються так само, як на схемах автоматизації.

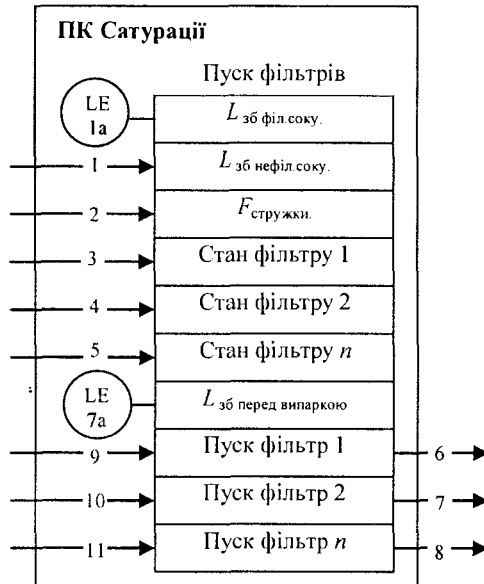


Рис. 15.5. Фрагмент блочної моделі інформаційних потоків

Додаткову інформацію, яку неможливо показати на схемі, приводять у текстовій частині у вигляді документів ПЗ та П4.

Приклад 15.2. Проектування. Використання блочної моделі інформаційних потоків

Завдання. Для розв'язання завдання 1.1 скористуватися блочною моделлю інформаційних потоків.

Рішення. Варіант представлений на рис. 15.6.

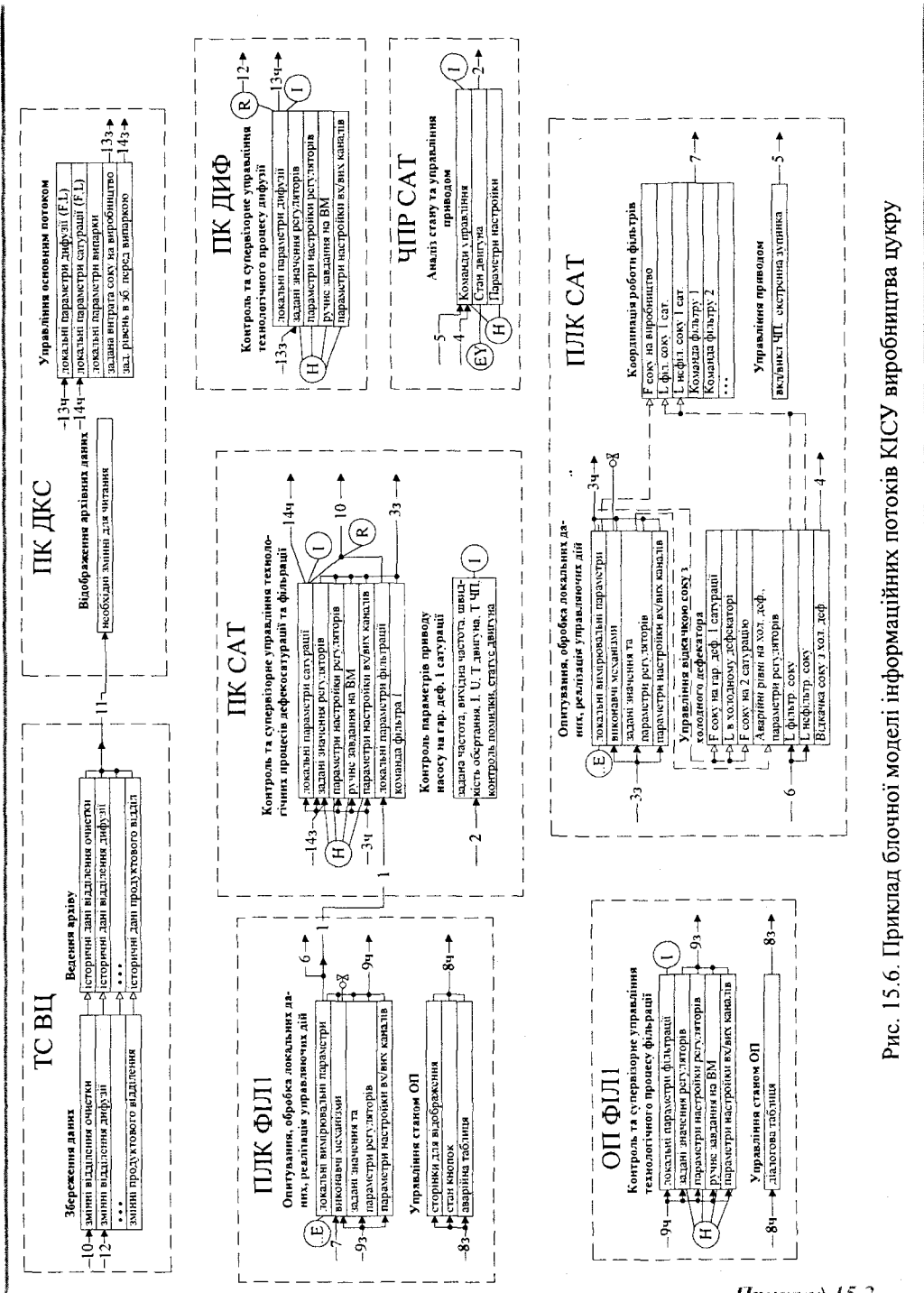


Рис. 15.6. Приклад блочної моделі інформаційних потоків КІСУ виробництва цукру

15.4. Розробка документів інформаційного забезпечення ІАС

15.4.1. Комплектність документів інформаційного забезпечення

Для організації інформаційного забезпечення на стадії «Технічний проект» необхідно розробити узгоджуючі документи комплекту інформаційного забезпечення на рівні всієї системи. Згідно зі стандартом, комплектність цих документів повинна вміщувати такі важливі розділи.

1. Опис інформаційного забезпечення системи. Він має розділи:

1.1. Склад інформаційного забезпечення: найменування та призначення всіх баз даних та наборів даних.

1.2. Організація інформаційного забезпечення: принципи організації інформаційного забезпечення системи; обґрунтування вибору носіїв даних та принципи розподілу інформації за типами носіїв; опис прийнятих видів та методів контролю в маршрутах обробки даних при створенні та функціонуванні позамашиної та внутрішньомашинної інформаційних баз з указівкою вимог, на відповідність яким проводять контроль; опис рішень, які забезпечують інформаційну сумісність АС з іншими системами управління за джерелами, споживачами інформації, за спряженням використовуваних класифікаторів (за необхідності), за використанням в АС уніфікованих систем документації.

1.3. Організація збору та передачі інформації: перелік джерел та носіїв інформації з указівкою оцінки інтенсивності та обсягу потоків інформації; опис загальних вимог до організації збору, передачі, контролю та корегування інформації.

1.4. Побудова системи класифікації та кодування: опис прийнятих для застосування в ІАС класифікації об'єктів у заново розроблених класифікаторах та в тих діючих класифікаторах, з яких використовується частина коду; методи кодування об'єктів класифікації в заново розроблених класифікаторах.

1.5. Організація внутрішньої інформаційної бази: опис принципів побудови внутрішньої інформаційної бази, характеристика її складу та обсягу; опис структури внутрішньої інформаційної бази на рівні баз даних з описом характеру взаємозв'язків баз даних та вказівкою функцій ІАС, при реалізації яких використовують кожен базу даних, характеристики даних, які вміщуються в кожній базі даних.

1.6. Організація позамашиної інформаційної бази: характеристики складу та обсягу позамашиної інформаційної бази, принципи її побудови, в тому числі основні положення з організації та обслуговування фонду нормативно-довідкової інформації у взаємозв'язку з автоматизованими функціями.

1.7. Опис інформаційного забезпечення системи: довідникові й інші додаткові матеріали та відомості (систематизований перелік найменувань структурних одиниць інформації з присвоєними їм позначеннями та описами їх сутності)

2. Опис організації інформаційної бази.

2.1. Опис внутрішньої інформаційної бази: опис логічної структури (складу даних, їх формату та взаємозв'язків між даними); опис фізичної структури (вибраний варіант та розміщення даних на конкретних машинних носіях).

2.2. Опис позамашиної інформаційної бази.

3. Опис масиву інформації.

3.1. Найменування масиву.

3.2. Позначення масиву.

3.3. Найменування носіїв інформації.

3.4. Перелік реквізитів у порядку їх слідування в записах масиву з указівкою за кожним реквізитом, позначення алфавіту, довжини в знаках та діапазону змін (за необхідності), логічних та семантичних зв'язків з іншими реквізитами даного запису та іншими записами масиву.

3.5. Оцінка об'єму масиву.

3.6. Інші характеристики масиву (за необхідності).

Для наочності представлення даної інформації на загальносистемному рівні пропонується використовувати схеми мережних інформаційних потоків, які графічно є масивом даних у базах даних вузлів системи, та їх взаємозв'язок інформаційними потоками.

15.4.2. Використання схеми мережних інформаційних потоків (СІП)

15.4.2.1. Призначення схеми мережних інформаційних потоків. Схему мережних інформаційних потоків (СІП) розробляють на основі деталізованої структури КТС ІАС (С1) та моделі мережних інформаційних потоків (С2Г або С2Б). Основне призначення схеми — показати реалізацію інформаційних потоків з точки зору інформаційного забезпечення (ІО), оскільки апаратна частина показується та описується на структурних схемах КТС.

Схема мережних інформаційних потоків повинна:

- дати уявлення про обмін даними в мережі між різними типами баз даних;
- служити інструментом для вияву конфліктних ситуацій, вирішення оптимальної стратегії зв'язку, зменшення надлишкових потоків і т.д.;
- допомогти в формуванні технічного завдання для підрозділів або підрядників, які відповідають за певну частину (підсистему) проекту ІАС, який розробляється.

Наочність дає змогу краще розуміти процеси обміну, які діють у системі, тому бажано особливо не нагнітати її надлишковою інформацією. Саме з цієї схеми можна почати розподіл адрес між пристроями, виділення ресурсів (змінних), визначення боку клієнта та серверу тощо.

Для грамотної побудови такої схеми проєктант повинен розуміти основні принципи роботи мережного обладнання, протоколів обміну, програмного забезпечення вузлів і т.д.

15.4.2.2. Елементи схеми інформаційних потоків. Основними елементами схеми є:

- 1) масиви даних процесу в межах баз даних, які беруть участь у загальносистемному, на певному рівні, обміні, тобто ті, які циркулюють по обчислювальним мережам;

2) інформаційні потоки, які забезпечують доставку даних з бази даних джерела в базу даних отримувачів; вони показані на моделі інформаційних потоків з відповідними номерами, що можуть співпадати, об'єднуватись, добавлятися; загальна концепція позначень інформаційних потоків на схемі повинна зберегтися з позначеннями на моделі (С2Г або С2Б);

3) мережні сервіси або протоколи, за допомогою яких реалізовані інформаційні потоки;

4) комунікаційні логічні канали пристроїв та програмного забезпечення, через які проходять інформаційні потоки: драйвер, логічний канал, адреса тощо;

5) уточнюючі характеристики інформаційних потоків: у графічній частині бажано показати, хто ініціював запит (Клієнт, Видавець або Виробник) та напрямок передачі інформації; в текстовій — всю іншу інформацію, відповідно до вимог до ІО.

15.4.2.3. Масиви даних. З точки зору моделі інформаційних потоків *масиви даних* або *блок даних* — це відображення аргументів функцій у конкретній базі даних, конкретного вузла системи. Кожний інформаційний потік починається та закінчується масивом даних або операціями. Масивами даних можуть бути змінні бази даних контролера, змінні бази даних реального часу SCADA, записи в базах даних і т.д., які, як правило, об'єднані (згруповані) між собою за принципом: один масив даних спільний для одного інформаційного потоку. Таке об'єднання дозволяє графічно наочно представити весь процес обміну, а інколи допомагає згрупувати змінні для зменшення навантаження на мережу.

Масиви даних розміщуються у базах даних програмно-технічного засобу. Під базою даних розуміється виділена структурна одиниця даних. Фізично масиви даних можуть являти собою пам'ять контролера або його частини, виділені для обміну змінної програми або її частини, дисковий простір тощо. Для ідентифікації розміщення даних необхідно показати їх у складі засобу в порядку їх вкладення:

засіб -> програма -> частина програми (база даних) -> масив даних.

На рис. 15.7 на комп'ютері ПК ДИФ розміщуються 4-ри масиви даних. Перші три масиви розміщені в SCADA-системі, однак потрібно ідентифікувати де саме вони містяться. Блоки 1 та 2 — це змінні (теги) бази даних реального часу (БДРЧ), а блок 3 — це таблиця, або її частина, реляційної бази даних (РБД) під назвою «Trends». Звісно, що деталізація розміщення залежить від конкретної ситуації. Так, наприклад, якщо всі дані

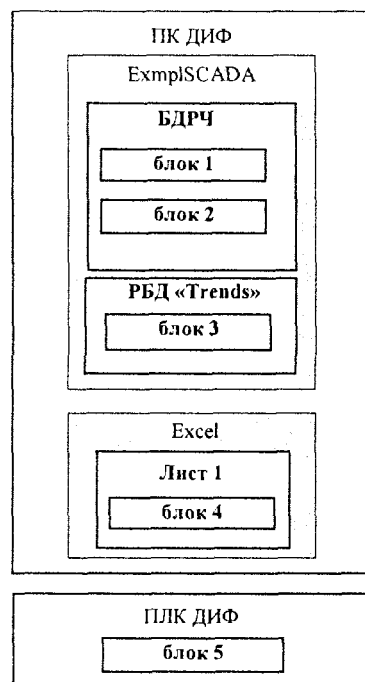


Рис. 15.7. Фрагмент схеми мережних інформаційних потоків

на схемі інформаційних потоків є тільки складовими бази даних реального часу, то для спрощення можна БДРЧ не вказувати. Блок 4 розміщений на тому самому комп'ютері, однак в іншій прикладній програмі — Excel. Причому ці дані розміщені на листі «Лист 1». Якщо якийсь масив даних стосується суто змінних ВВА в Excel, то необхідно вказати разом з листом ще і ВВА-модуль.

На контролері, як правило, виконується тільки одна прикладна програма (принаймні з точки зору розробника програми користувача). Тому блок 5 розміщується безпосередньо в ПЛК ДИФ.

Для ідентифікації складових масивів даних, у них вказуються назви змінних, або групи змінних, таблиці змінних, таблиці, масиви, записи тощо. Є декілька способів показати склад масивів. Перший спосіб — відобразити масив у вигляді списку, тобто як повний перелік одиничних даних однакового формату (рис. 15.8. — верхні блоки). Цей універсальний спосіб дає змогу побачити всі елементи з даного масиву без додаткової уточнюючої інформації. Такий спосіб втрачає свої переваги при великій кількості одиничних даних.

Більш економічний спосіб, з точки зору місця на кресленні, — *монолітний*, тобто використання унітарного позначення всього масиву (рис. 15.8 — середні блоки). Тоді весь масив позначається одним унікальним для цього креслення ідентифікатором, який розшифровується (розписується) в текстовій частині. Якщо врахувати, що перелік мережних змінних (вхідних та вихідних змінних) повинен перераховуватися в текстовій частині ІО, цей спосіб має суттєві переваги перед попереднім. Однак, у деяких випадках переваги перетворюються в недоліки. Так, наприклад, за присутності одних і тих самих змінних у декількох інформаційних потоках, вони повторюються в різних масивах, що не видно на основному кресленні. В цьому випадку перехресні (стосовно потоків) дані можна виділити в окремі масиви як у монолітному вигляді, так і у вигляді списку.

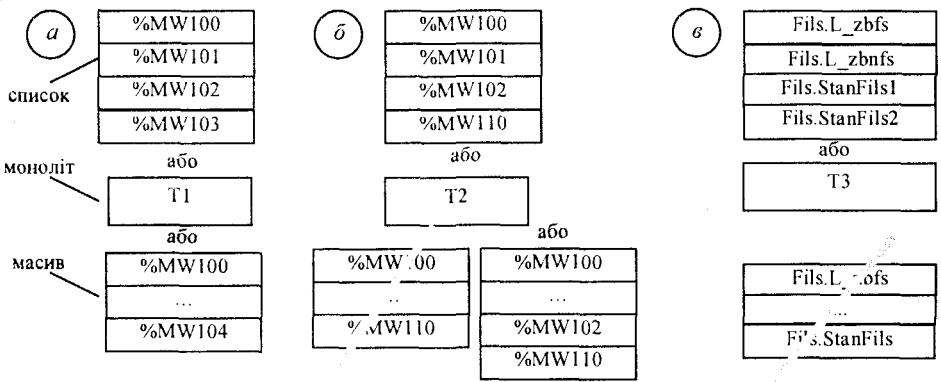


Рис. 15.8. Приклади зображення масивів даних: а, б — в ПЛК; в — в SCADA

Деякі дані можна представити у вигляді розірваного масиву з першим та кінцевим елементами. Таке зображення можливе у випадку, коли термінальні елементи (перший та останній у масиві) повністю визначають весь склад масиву да-

них. Так, на рис. 15.8 показані рекомендовані зображення масивів. Для ситуації (а) можливі всі три варіанти, оскільки змінні нумеруються і розміщені без розриву за номером. Однак, якщо деякі з них присутні в різних інформаційних потоках, використання даних у вигляді моноліту та розірваного масиву себе не виправдує. Крім того, якщо існує розрив у номерах, то слід указати ключові номери елементів, як, наприклад, на рис. 15.8, (б).

Усі ці способи можна комбінувати. Так, для змінних контролера найбільш підходить вигляд розірваного масиву, для тегів SCADA — список або моноліт. Для останніх вигляд розірваного масиву не підходить, оскільки дані між термінальними елементами не можна однозначно ідентифікувати (рис. 15.8, (в)). З приводу зображення даних слід мати на увазі, що все наведене вище є рекомендаціями. Головне правило — дані повинні бути однозначно ідентифіковані і прив'язані до інформаційних потоків.

15.4.2.4. Інформаційні потоки. Інформаційні потоки зображуються у вигляді лінії, на якій вказується його номер. Для уточнення інформації про інформаційний потік номер може бути вписаний у певну геометричну фігуру. Фігура може вказувати на тип сервісу. Наприклад, для Клієнт-Серверних моделей сервісів з боку Клієнта (той, хто ініціює запит) можна вказати фігурну стрілку, яка вказує на напрямок передачі даних, а з боку Серверу — коло зі стрілками (рис. 15.9). У всіх інших випадках можна вказувати круг. Для pull моделі Видавець-Абонент (Виробник-Споживач) фігурну стрілку можна вказати з боку Pull Publishing Manager. За двостороннього обміну (читання/запис однокорих змінних), при використанні одного і того ж сервісу, два і більше потоків можна умовно об'єднувати в один. При цьому з боку Клієнта вказується двостороння фігурна стрілка, а в текстовій частині інформація уточнюється.

Номери потоків у схемі інформаційних потоків повинні перекликатися з номерами в моделі інформаційних потоків. Однак, за необхідності, — коли потік реалізовується як декілька окремих потоків або, навпаки, коли декілька потоків об'єднуються в один, можна змінити позначення.

Так, при розбитті потоку 1 на три потоки можна їх позначити відповідно 1–1, 1–2, 1–3, а при об'єднанні потоків 4, 8, 9 можна позначити потік 4.8.9. Головний критерій формування таких номерів — максимальна простота і зрозумілість. У будь-якому випадку на кресленні в таблиці умовних позначень необхідно вказати призначення кожного неоднозначного елемента схеми.

15.4.2.5. Комунікаційні сервіси. На кресленні вказуються *комунікаційні сервіси*, якими «користуються» інформаційні потоки. Якщо сервіси однозначно визначаються протоколом, то вказується протокол обміну. Як правило, необхідно вказувати прикладні сервіси і, відповідно, прикладні протоколи. Для спрощення в контексті правил побудови буде використовуватись слово «сервіс», а не «протокол».

Кожний потік починається з масиву даних. Початком потоку умовно можна вважати бік його ініціювання. Далі в порядку слідування потоку вказуються ко-

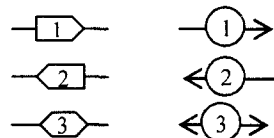


Рис. 15.9. Приклади зображення інформаційних потоків

мунікаційні засоби програмного забезпечення, що відповідає за обмін, тобто його інтерфейсна частина (драйвери пристроїв, бібліотеки обміну і т.д.). Якщо ці складові приховані від розробника системи і вважаються внутрішньою закритою складовою програмно-технічного засобу, то її вказувати не потрібно. Вказуються тільки ті частини програмного забезпечення, які неоднозначні і можуть бути замінені на альтернативні. Вказувати частини програмного засобу, що відповідають за обмін, надзвичайно важливо, оскільки це дає уявлення про його реалізацію.

На рис. 15.10 показані три типові випадки (не повний перелік) слідування інформаційних потоків від масиву даних до зовнішнього порту комп'ютера. Всі вони користуються умовними інтерфейсними каналами прикладної програми, які будемо називати *драйверами*. Для кожного драйвера можуть існувати настройки, які інколи необхідно уточнювати або на схемі, або в текстовій частині. Як мінімум, на схемі потрібно показувати справжню або умовну назву цього каналу.

Перший інформаційний потік починається з масиву даних T1. Для передачі даних по мережі M1 використовується драйвер X, який є інтерфейсною частиною прикладної програми ExmplSCADA. В даному випадку достатньо вказати тільки цей інтерфейс, можливо з деякими уточненнями. Якщо всі дані передаються через єдиний інтерфейс, який неможливо змінити, то драйвер X не вказується, оскільки інших варіацій бути не може.

Інколи драйвер прикладної програми використовує проміжну бібліотеку (програму, драйвер). Причому ім'я бібліотеки або її тип можуть відрізнитися. Для прикладу, якщо на рис. 15.10 драйвер Y — це реалізація OPC-клієнтського інтерфейсу, то необхідно вказати також OPC-Сервер для реалізації потрібного протоколу мережі. В цьому випадку бібліотеку просто показують на шляху слідування потоку без його «розриву». Уточнення до бібліотеки можна вказати разом з її назвою.

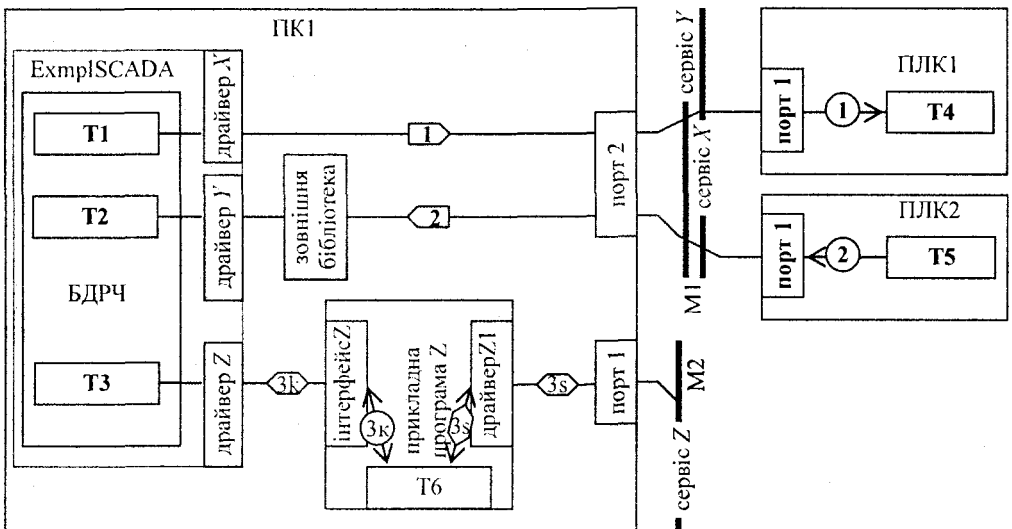


Рис. 15.10. Приклад проходження інформаційних потоків

За необхідності в проміжній бібліотеці (програмі, драйвері) можна деталізувати внутрішню структуру. Так, наприклад, при використанні OPC з декількома реалізованими протоколами обміну необхідно вказати, який з них використовується даним інформаційним потоком (йде через нього), та настройки цього драйвера. В деяких випадках необхідно показати внутрішні дані цієї програми, через який проходить обмін. На рис. 15.10 інформаційний потік 3 проходить через таку програму. Вона «розриває» потік, тому він реалізовується у вигляді двох потоків 3k та 3s. Всередині показаний допоміжний масив даних T6.

15.4.2.6. Логічні канали. Після програмного забезпечення, драйверів, бібліотек на шляху проходження потоку, за необхідності, вказуються адреси станцій (вузлів) на мережі та логічні канали або порти, які є інтерфейсними для програмно-технічного засобу. *Логічний канал* є відображенням фізичного каналу, а його маркування залежить від реалізації та типу ПЗ. Якщо ніякого представлення каналу в останнього немає, загалом логічний канал може мати вигляд:

тип_комунікаційного_обладнання : номер_плати . номер_каналу.

Такий вигляд необхідний для зрозумілої ідентифікації каналу. Логічні канали на одному пристрої не можуть мати один і той же ідентифікатор, однак можуть співпадати з логічними каналами інших пристроїв.

Адреса(-си) можуть прив'язуватися до логічних каналів. У випадках, коли на одному логічному каналі міститься декілька адрес, пристрій через нього може спілкуватися від різних адрес, а отже необхідно їх указати на шляху інформаційних каналів. Коли канал використовується декількома драйверами (якщо таке можливо), адреси прив'язуються не до каналу, а до драйвера.

15.4.2.7. Інформаційні шини. Після каналу всі потоки об'єднуються в одну *інформаційну шину*, що являє собою використання єдиного комунікаційного сервісу, яким вони користуються. Окрім інформації про спосіб реалізації потоку, таке представлення може допомогти оцінити інформаційне навантаження на мережу. Одна мережа (фізична) та протокол можуть підтримувати декілька сервісів (протоколів). Інформаційні шини не обов'язково повинні відповідати сервісам протоколу одного рівня (наприклад, прикладного), але повинні однозначно вказати, чим «користуються» інформаційні потоки.

Інформаційний потік закінчується аналогічно, як і починався. Тому «початок» потоку є умовним і не обов'язково починається з боку Клієнта. Оскільки в ПЛК1 та ПЛК2 (рис. 15.11) немає явно виділених прикладних програм та драйверів, масив даних інформаційним потоком зв'язується безпосередньо з логічним каналом. Однак, до цього каналу можна прив'язати мережну адресу або логічний порт і т.д., що розглянуто нижче.

15.4.2.8. Текстова частина. Текстова частина повинна описати кожний інформаційний потік з точки зору його реалізації. Це параметри інформаційних потоків за номерами: періодичність обміну, уточнення до способу реалізації. Крім того, текстова частина повинна вмещувати необхідну інформацію за масивими даних, що наведені в переліку комплектності інформаційного забезпечення технічного та робочого проектів (П5-П8).

15.4.2.9. Рекомендації до виконання. Схему рекомендується приводити окремо для кожної мережі (рівня мереж), якщо дані з пристрою одної мережі (одна схема) безпосередньо не передаються на пристрої іншої мережі (друга схема). Тобто інформаційний потік графічно бажано не «розривати». Крім того, краще не нагнітати схему маловажливою інформацією, яку можна вказати в текстовій частині, оскільки втрачається її наочність.

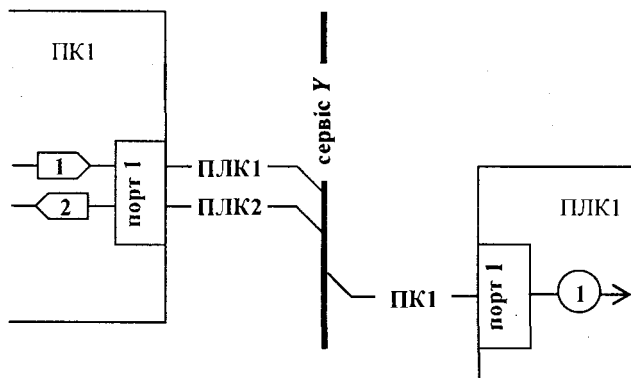


Рис. 15.11. Приклад зображення інформаційного потоку з інформаційною шиною

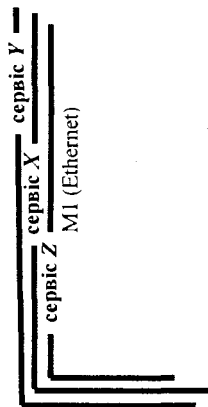


Рис. 15.12. Приклад зображення інформаційних шин

Для швидшого пошуку продовження інформаційного потоку після інформаційної шини можна «загинати» кінці потоку при попаданні на шину в той бік, куди направляється потік (рис. 15.11). Крім цього, можна вказувати назву пристрою — адресату.

У мережі можуть використовуватись декілька сервісів, які показуються інформаційними шинами. Назву кожної інформаційної шини бажано показувати в її розриві, а самі шини розміщувати якнайближче одна до одної в складі єдиної фізичної мережі. Графічне групування шин можна проводити зміною кольору заднього фону (рис. 15.12), або об'єднанням в єдиний контур.

15.4.3. Використання схеми інформаційних потоків при проектуванні мереж MODBUS RTU

Виконання схеми інформаційних потоків розглянемо на прикладі MODBUS RTU (рис. 15.13). Комп'ютер під'єднується до шини MODBUS RTU через COM-порти. Для операційної системи ПК не важливо, чи це реалізація інтерфейсу

RS-232, чи RS-485 у послідовному режимі. Для неї це завжди буде COM-порт з певним номером. Оскільки при настройці драйвера необхідно вказати номер порту, його треба показати на інформаційному потоці. Як відомо, SCADA/НМІ-пристрої на MODBUS RTU, як правило, є Клієнтами, а отже Ведучими, однак це теж треба показати, адже можуть бути винятки (деякі SCADA/НМІ підтримують драйвер MODBUS RTU Веденого).

Хоч на MODBUS RTU використовуються тільки Modbus-запити, все одно їх бажано показати для уніфікації підходу. Один із варіантів — показувати їх у вигляді зноски, як це продемонстровано на рис. 15.13.

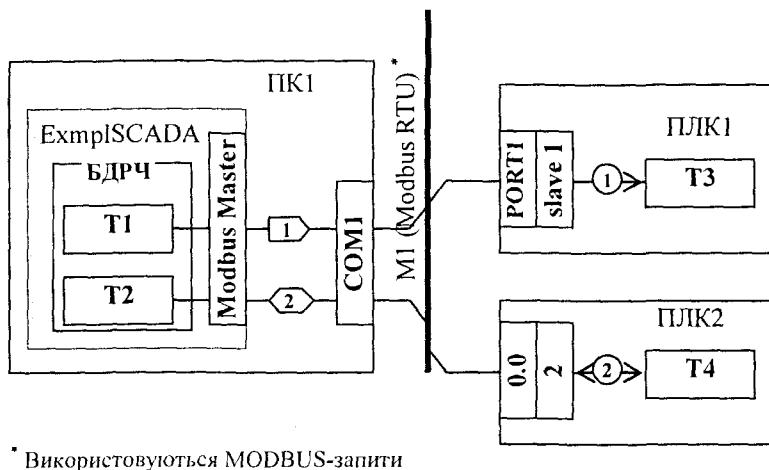


Рис. 15.13. Приклад схеми мережних інформаційних потоків системи, реалізованої з використанням MODBUS RTU

ПЛК на шині може бути як Ведучим, так і Веденим. Оскільки в нашому прикладі Ведучий вже є, ПЛК1 та ПЛК2 є Веденими. Адреса Веденого прив'язується до логічного каналу контролера. В ПЛК1 логічний канал адресується як PORT1 (наприклад, контролер TWIDO). В ПЛК2 адресація йде за принципом номер_модуля.номер_каналу (наприклад, TSX Premium), тобто з'єднання з MODBUS RTU проводиться через 0-й канал 0-го модуля (термінальний порт). Адреса Ведених для двох ПЛК для прикладу показана по-різному: слово «slave» може вказуватись, а може бути присутній тільки номер Веденого.

15.4.4. Використання схеми інформаційних потоків при проектуванні мереж на базі Ethernet

На рис. 15.14 показано функціонування системи, яка складається з трьох ПК та двох ПЛК, які підключені до Ethernet. Комп'ютери підключаються за допомогою вбудованих портів на материнській платі, ПЛК1 та ПЛК2 — через інтегровані канали в процесорному модулі. Всі сервіси без винятку в даному випадку

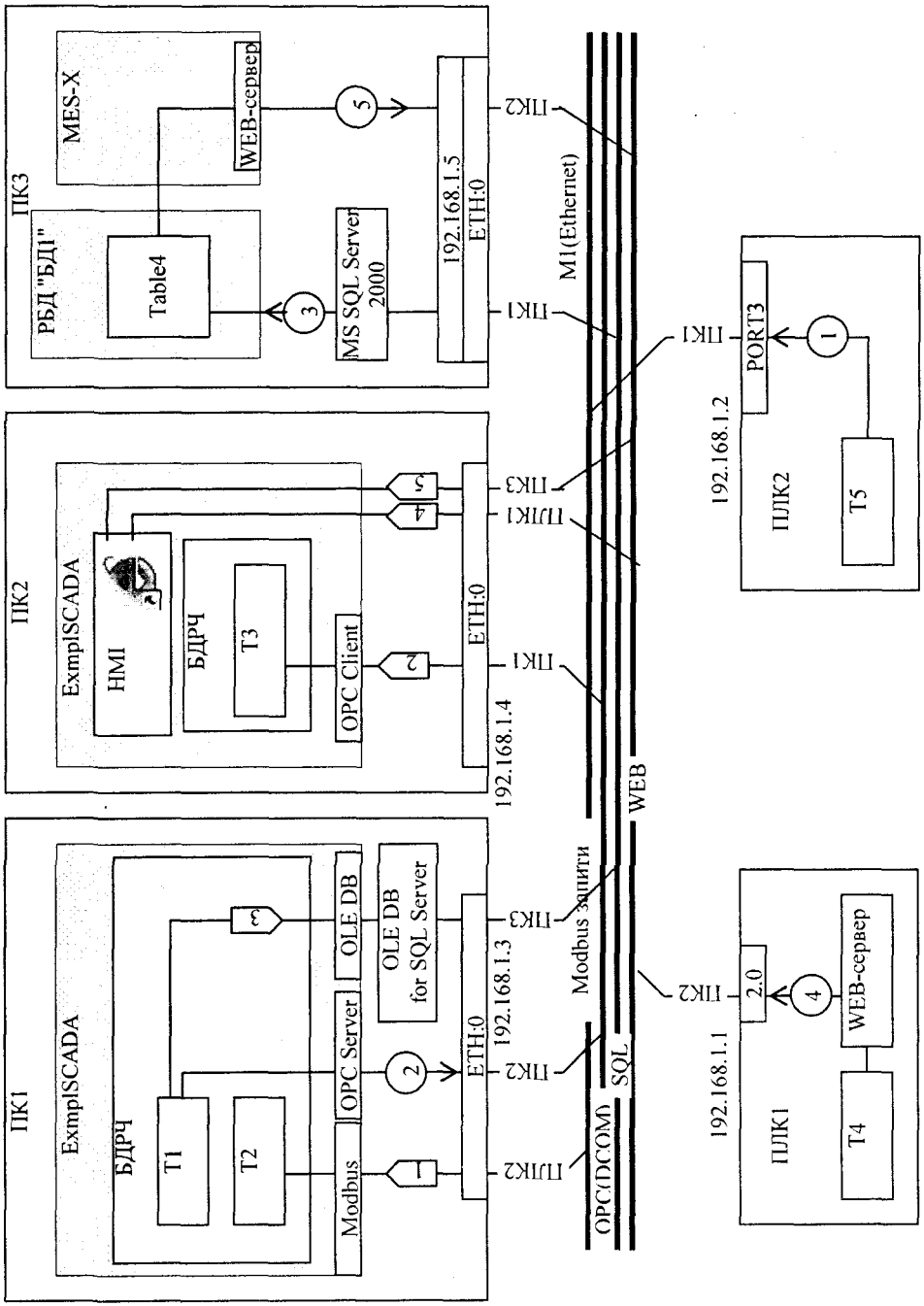


Рис. 15.14. Приклад схеми мережних інформаційних потоків мережі реалізованій на Ethernet

використовують протоколи TCP/IP, тому до портів необхідно прив'язати адресу IP (у даній ситуації MAC-адреса мало цікавить інтегратора). Графічно адреса може бути розташована над логічним каналом, як показано в ПК3, або біля нього — як в усіх інших випадках.

Прокоментуємо потоки.

1-й потік. SCADA в ПК1 читає змінні з ПЛК2, послуговуючись MODBUS TCP/IP. Для цього в SCADA використовується відповідний драйвер, а ПЛК, звісно, підтримує цей протокол (наявність Ethernet у ПЛК не визначає присутність протоколу MODBUS TCP/IP).

2-й потік. SCADA на ПК2 читає дані зі SCADA на ПК1, використовуючи технологію OPC, яка, в свою чергу, базується на сервісах DCOM. Для цього SCADA на ПК2 має OPC-клієнтський інтерфейс, а на ПК1 — OPC-серверний.

3-й потік. SCADA на ПК1 записує дані до архіву реляційної бази даних під назвою «БД1» в таблицю «Table4». Формат БД — SQL Server, управляється СУБД MS SQL Server 2000, обмін проводиться за допомогою технології OLE DB через відповідний провайдер даних.

4-й потік. У SCADA на ПК2 вбудований стандартний елемент ActiveX — Internet Explorer, за допомогою якого користувач може доступатися до WEB-сторінок. У даному випадку він може заглянути на WEB-сторінку ПЛК1, на якій зображена мнемосхема з даними Т4.

5-й потік аналогічний 4-му. В цьому випадку користувач доступується через WEB-сервер програми MES-системи під назвою MES-X до архівних даних з БД1.

У наведеному нижче прикладі названі всі деталі побудови графічної та текстової частини схеми інформаційних потоків.

Приклад 15.3. Проектування. Виконання деталізованої структурної схеми КТС.

Завдання. Розробити деталізовану структурну схему КТС, відповідно до завдання з прикладу 15.1.

Рішення. Деталізована структурна схема КТС показана на рис. 15.15. Специфікація мережних засобів показана в табл. 15.7.

Приклад 15.3

Приклад 15.4. Проектування. Виконання схеми мережних інформаційних потоків.

Завдання. Розробити схему мережних інформаційних потоків до завдання з прикладу 15.1.

Рішення. На основі структурної схеми КТС та моделі інформаційних потоків розробляється схема мережних інформаційних потоків — рис. 15.16. До неї додається таблиця мережних змінних, (табл. 15.8).

Таблиця 15.8

**ФРАГМЕНТ ТАБЛИЦІ МЕРЕЖНИХ ЗМІННИХ ІАСУ
ВИРОБНИЦТВОМ ЦУКРУ**

Призначення	ПК САТ (SCADA)		ПЛК САТ	ПЛК ФІЛІ	ЧРП САТ
F соку на виробництво	Fsok_vir	T2	%MW100		
F соку на гар. деф. 1 сатурації	Fsok_gar_def	T2	%MW101		
F соку повернення	Fsok_pover	T2	%MW102		
F соку на 2 сатурацію	Fsok_2sat	T2	%MW103		
F молока на 1 сатурації	Fmol_1sat	T2	%MW104		
F молока на 2 сатурації	Fmol_2sat	T2	%MW105		
F CO ₂ на сатуратор 1А	Fgaz_1Asat	T2	%MW106		
L у холодному дефекаторі	L_hol_def	T2	%MW107		
L у збірнику фільтрованого соку 1 сатурації	Lzb_fil_1sat	T2	%MW108		
L у збірнику нефільтрованого соку 1 сатурації	Lzb_nefil_1sat	T2	%MW109		
L у збірнику фільтрованого соку 2 сатурації	Lzb_fil_2sat	T2	%MW110		
L у збірнику нефільтрованого соку 2 сатурації	Lzb_nefil_2sat	T2	%MW111		
L перед випаркою	L_na_vip	T2	%MW112		
pH у переддефекаторі	pH_preddef	T2	%MW113		
pH у сатураторі 1Б	pH_1Bsat	T2	%MW114		
pH у сатураторі 2	pH_2sat	T2	%MW115		
D густина молока	Dmol	T2	%MW116		
T 1-го ступеня підігрівників перед гарячим дефекатором	Tsok_pid1st	T2	%MW117		
T 2-го ступеня підігрівників перед гарячим дефекатором	Tsok_pid2st	T2	%MW118		
T 3-го ступеня підігрівників перед гарячим дефекатором	Tsok_pid3st	T2	%MW119		
T підігрівача нефільтрованого соку 1 сатурації	Tsok_nefil	T2	%MW120		
P газу в колекторі CO ₂	Pgaz	T2	%MW121		

Закінчення табл. 15.8

Призначення	ПК CAT (SCADA)		ПЛК CAT	ПЛК ФІЛІ	ЧРП CAT
Подача молока в переддефекатор	Kl_mol_preddef	T2	%MW122		
Подача молока в гарячий дефекатор 1 сатурації	Kl_mol_gardef_1sat	T2	%MW123		
Подача молока на 2 сатурацію	Kl_mol_2sat	T2	%MW124		
Подача газу на сатуратор 1А	Kl_gaz_1Asat	T2	%MW125		
Подача газу на сатуратор 1Б	Kl_gaz_1Bsat	T2	%MW126		
Витрата CO ₂ в атмосферу	Kl_gaz_atm	T2	%MW127		
Подача газу на 2 сатурацію	Kl_gaz_2sat	T2	%MW128		
Подача пари в підігрівник нефільтрованого соку 1 сатурації	Kl_par_nefil_1sat	T2	%MW129		
Подача пари в підігрівник нефіл. перед гар. деф. 1 сат.	Kl_par_nefil_gardef_1sat	T2	%MW130		
Подача пари в підігрівник соку на 2 сатурацію	Kl_par_nefil_2sat	T2	%MW131		
Відкачка соку зі збірника філ.соку 1 сатурації	Kl_sok_fil_1sat	T2	%MW132		
Відкачка соку зі збірника філ.соку 2 сатурації	Kl_sok_fil_2sat	T2	%MW133		
Команда на вкл/викл ФІЛІ			%MW181	%MW200	
ATV необх. частота двигуна	Atv_FreqRef	T3	%MW220		%MW450
ATV вихідна частота двигуна	Atv_FreqOut	T3	%MW221		%MW451
ATV швидкість обертання	Atv_MotSpeed	T3	%MW222		%MW452
ATV струм у двигуні	Atv_MotCur	T3	%MW223		%MW453
ATV напруга живлення	Atv_LineVolt	T3	%MW224		%MW454
ATV температура двигуна	Atv_MotTemp	T3	%MW225		%MW455
ATV температура перетвор.	Atv_ATVTemp	T3	%MW226		%MW456
ATV код помилки	Atv_LastFault	T3	%MW227		%MW457
ATV DRIVECOM статус	Atv_EtaStatus	T3	%MW228		%MW458
ATV DRIVECOM команда			%MW205		%MW400
ATV задана частота			%MW200		%MW401

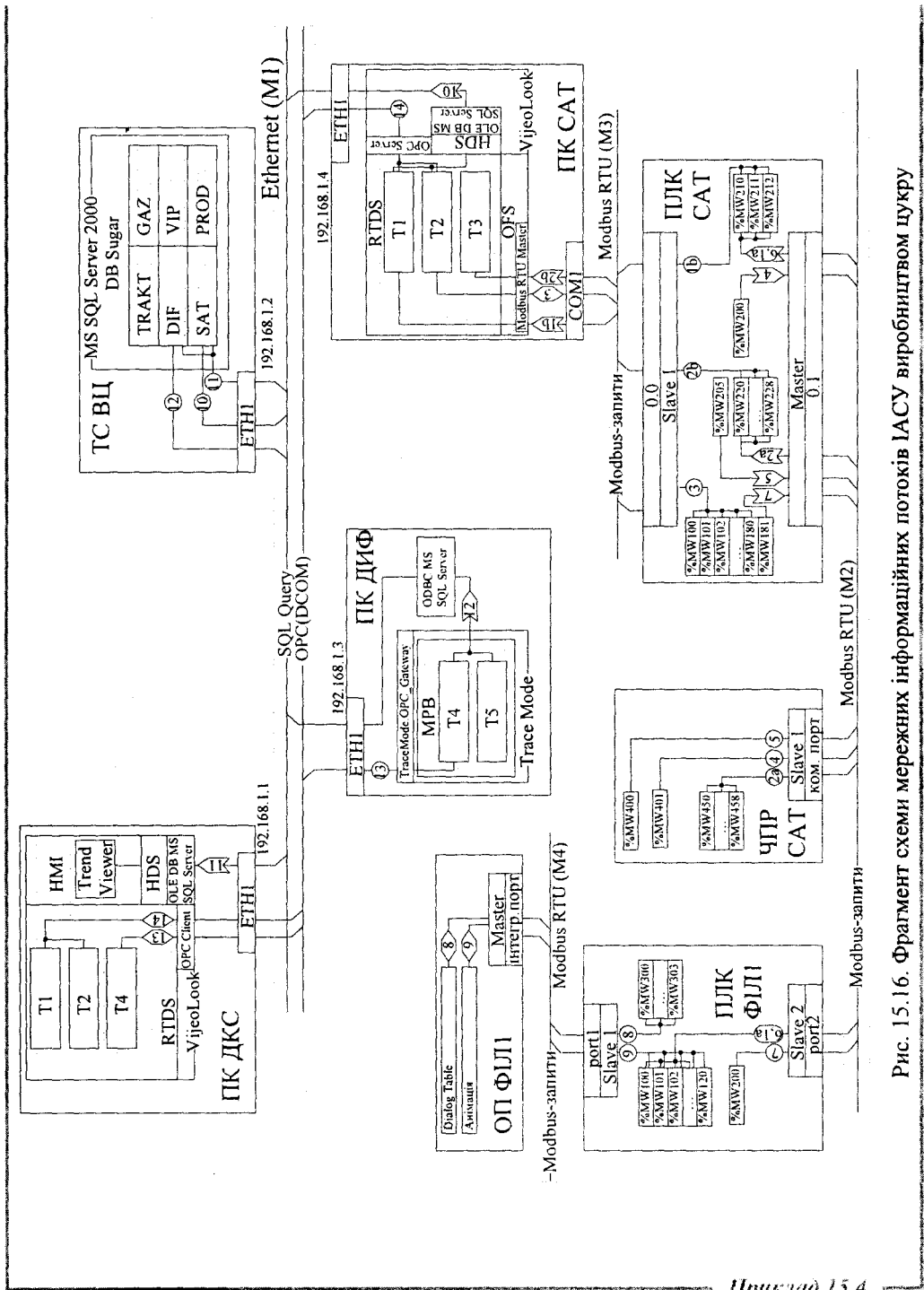


Рис. 15.16. Фрагмент схеми мережних інформаційних потоків ІАСУ виробництва цукру

15.4.5. Використання схеми інформаційних потоків при проектуванні мережі PROFIBUS

Загальні рекомендації для побудови схеми мережних інформаційних потоків для мережі PROFIBUS залишаються без змін. На прикладному рівні або на «рівні профілів» можливе послуговування різними типами сервісів, що необхідно показати на схемі інформаційних потоків. Для обміну процесними даними можуть використовуватись циклічний (Data-Exchange) та ациклічний DP-обмін даними, обмін типу Видавець/Абонент, обмін FMS-повідомленнями. Крім того, на цій же мережі можуть підтримуватись інші сервіси, які не визначені в стандартах PROFIBUS. Так, у програмно-технічних засобах Siemens та VIPA підтримуються Siemens-сумісні S7-функції обміну (до них також належать PG/OP-сервіси) та S5-функції (SEND/RECEIVE). Таким чином, на одній мережі можуть бути зображені декілька інформаційних шин.

Вхідні та вихідні дані, що беруть участь в обміні показуються, відповідно, як окремі масиви даних. У наповненні масиву даних вказується джерело даних для даного вузла (змінні, номери входів або виходів, назва параметра тощо).

Для клієнт-серверного типу обміну циклічними та ациклічними даними процесу позначення клієнтської та серверної сторони співпадає із загальноприйнятими. Для обміну Видавець-Абонент номери потоків у фігурних стрілках можна вказувати з боку Видавця, а можна з боку Ведучого. У будь-якому випадку позначення необхідно описати.

За необхідності, на схемі інформаційних потоків можна окремо виділяти групи SYNC/FREEZE, консистентність даних тощо.

Приклад 15.5. Проектування. Схема мережних інформаційних потоків PROFIBUS

Завдання. Розробити схему мережних інформаційних потоків до завдання 7.2 за умови добавлення до мережі операторської панелі Siemens TP 177B, яка обмінюється даними з PLC1 (з DB10.DBW0 по DB10.DBW98) за S7 сумісним протоколом зв'язку.

Рішення. На рис. 15.17 наведена схема мережних інформаційних потоків для даного завдання. Всі пристрої, крім OP1 (операторська панель Siemens TP 177B), обмінюються даними з Ведучим з використанням циклічного DP-обміну Data-Exchange. На графічному зображенні даний сервіс покажемо, як DP. Враховуючи, що в даному обміні процес Ведучого є Клієнтом, саме від нього показаний бік ініціювання обміну.

Вузол OP1 є DP-Веденим на шині (операторські панелі можуть бути і Ведучими шини). Дана панель використовує DP-сервіси тільки для відправки Ведучому (PLC1) номера нажатой клавіші на сенсорному екрані панелі (режим Direct Keys). Для цього в області входів PLC1 виділяється 4-ри байти (IB3-IB6), де буде зберігатися стан 32 клавішів панелі. Зчитування та запис значень для змінних Tag1-Tag50 операторська панель буде проводити з використанням S7 функцій зв'язку, зокрема сервісів PG/OP.

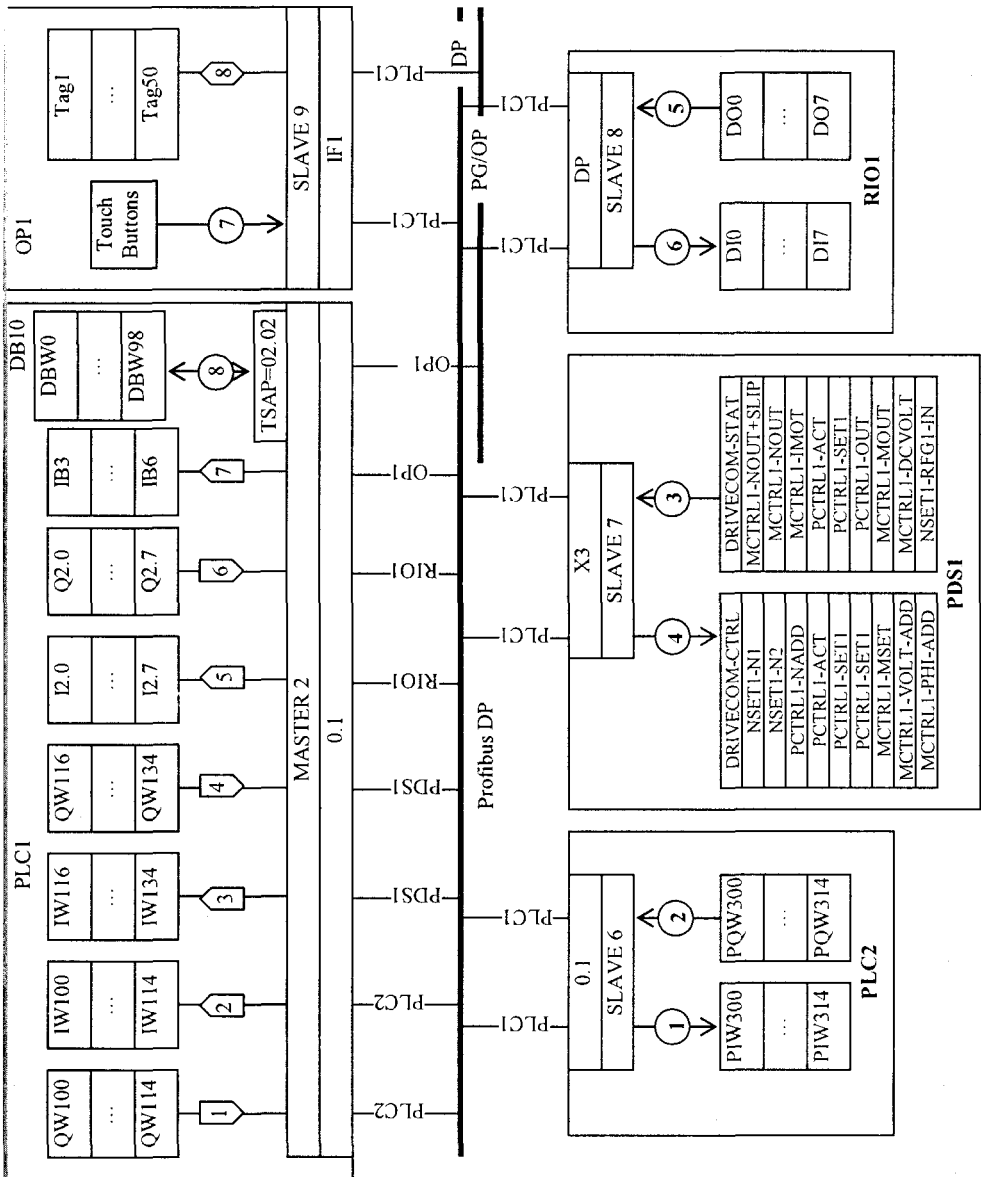


Рис. 15.17. Схема мережних інформаційних потоків до прикладу 15.5

S7-комунікації є стандартним типом зв'язку між ПЛК Siemens S200, S300/400 та засобами ЛМІ від Siemens. Вони базуються на клієнт-серверному обміні зі встановленням з'єднання. З'єднання можуть бути наперед конфігурованими або без попередньої конфігурації. У будь-якому випадку Клієнт та Сервер організують з'єднання через транспортні адреси TSAP. Адреса TSAP складається з 2-х байтів: 1-й байт — ресурс з'єднання (Connection Resource), 2-й байт — шасі та слот модуля з'єднання (Rack/Slot). Для PG та OP з'єднань зарезервовані спеціальні ресурси з'єднання: PG = 1 та OP = 2, тому при використанні цих серверних ресурсів, попередньо конфігурувати з'єднання не потрібно. Виходячи з вищесказаного, для налаштування S7 зв'язку при конфігурації панелі необхідно, крім адреси DP-вузла, вказати шасі (=0) та слот (=2), де міститься центральний процесор PLC1, який є його партнером по зв'язку. Враховуючи, що панель використовує OP-ресурс, серверний TSAP для зв'язку OP1-PLC1 буде дорівнювати TSAP=02.02. Клієнтський TSAP у даному випадку виділяється в OP1 самостійно, тому його вказувати при конфігуруванні не потрібно.

Слід звернути увагу, що на схемі мережних інформаційних потоків 8-й потік, що реалізований з використанням PG/OP сервісів, ініціюється з боку операторської панелі OP1, оскільки вона є Клієнтом.

Приклад 15.5

15.4.6. Використання схеми інформаційних потоків при проектуванні мереж CANOpen

На прикладному рівні CANOpen для обміну даними процесу можуть використовуватися два типи сервісів: PDO та SDO. Тому на схемі мережних інформаційних потоків в якості інформаційної шини необхідно показати, який саме тип сервісу використовується.

Масиви даних процесу, які беруть участь в обміні, показуються окремо для кожного PDO/SDO. Наповнення масиву даних для PDO співпадає з PDO-Відображенням (PDO Mapping), а номер TPDO чи RPDO вказується як назва цього масиву. Бажано також вказати COB-ID для кожної пари зв'язаних PDO. Якщо в якості NMT-Ведучого є ПЛК, то для масиву даних вказується діапазон змінних, який пов'язаний з TPDO або RPDO.

Для обміну великим обсягом даних процесу можуть використовуватись SDO. В цьому випадку у вигляді масиву даних вказуються об'єкти SDO (номери та COB-ID), а в якості його наповнення — перелік назви об'єктів та їх Index/ SubIndex, які переносяться цим SDO. Слід зауважити, що SDO використовується для конфігурації вузлів, однак на схемі мережних інформаційних потоків немає сенсу це вказувати.

Приклад 15.6. Проектування. Схема мережних інформаційних потоків CANOpen

Завдання. Розробити схему мережних інформаційних потоків до завдання 9.7 за умови, що 1-й, 2-й та 4-й потоки реалізують запис значення за зміною, 3-й — періодичне читання.

Рішення. Схема мережних інформаційних потоків зображена на рис. 15.18. Для реалізації всіх потоків використовуються сервіси PDO, що зображено відповідною інформаційною шиною. Значення змінних відповідно до розв'язання завдання 9.7 передаються та приймаються TPDO з номерами, що вказані на рис. 15.18 біля кожного масиву даних. Оскільки змінні з PDS1 передаються двома PDO (TPDO1 та TPDO2), інформаційний потік поділений на дві частини: 3 та 3a. Кожний PDO в мережі має унікальний COB-ID, який вказаний біля цих PDO.

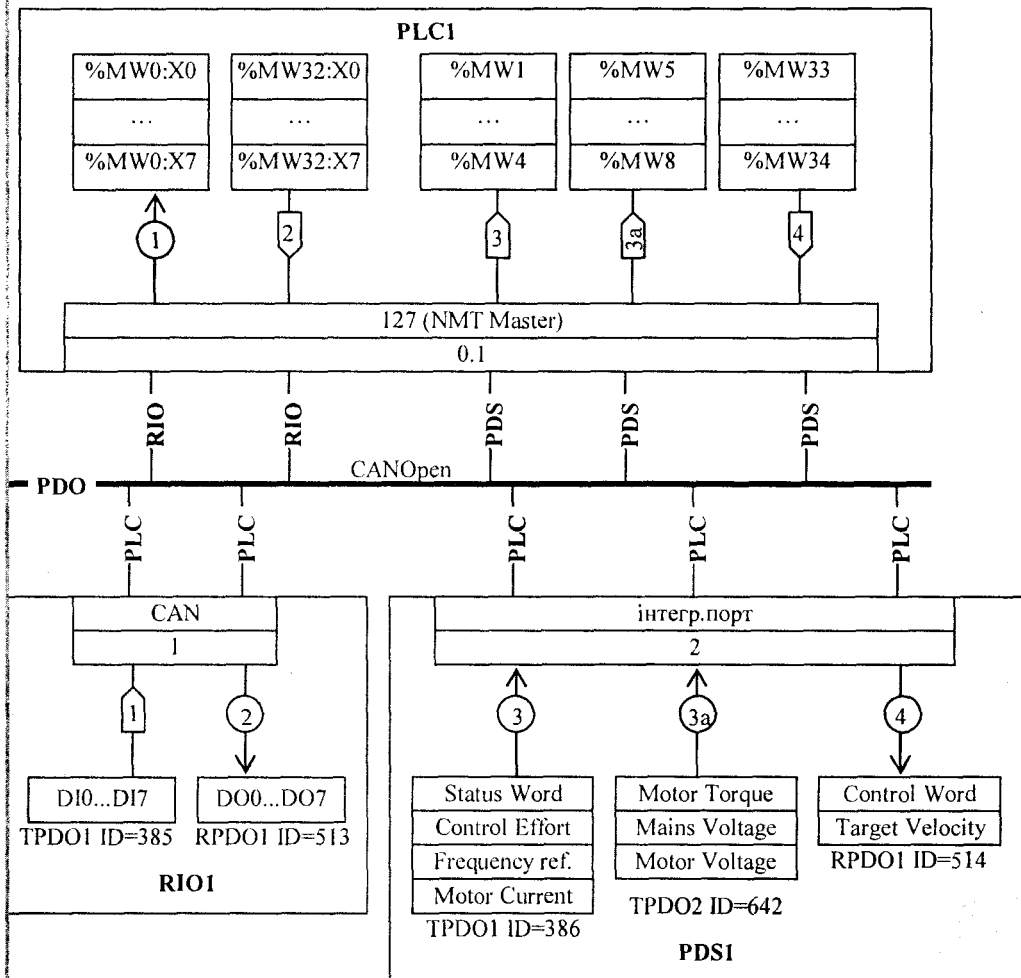


Рис. 15.18. Схема мережних інформаційних потоків до прикладу 15.6

PDO, відповідно до завдання мають такі комунікаційні режими (додатково див. 9.4.3):

– RIO1: TPDO1 — (передача значення в PLC1 за зміною в RIO1), RPDO1 — 255 (передача значення в RIO1 за зміною в PLC1);

– PDS1: TPDO1 — 253 (передача значення за запитом RTR), TPDO2 — 253 (передача значення за запитом RTR), RPDO1 — 255 (передача значення в PDS1 за зміною в PLC1)

Таким чином, 2, 3, 3а та 4 потік будуть ініціюватися PLC1, а саме: 2-й та 4-й — це запис при зміні значення, 3 та 3а — читання шляхом передачі запиту RTR (періодичність виконання потоків 3 та 3а налаштовується в конфігураторі SyCon). Перший потік ініціюється RIO1, оскільки передача значення TPDO починається зі зміни його значення в RIO1. Тому ці потоки мають відповідне зображення на схемі.

Враховуючи відсутність можливості явного конфігурування PDO в NMT-Ведучому та відсутність динамічного зв'язування PDO, на схемі (рис. 15.18) біля масивів змінних відсутня ідентифікація PDO в PLC1.

Приклад 15.4



Контрольні запитання до розділу 15

1. Які стандарти проектування поширюються на інтегровані автоматизовані системи?
2. З яких стадій складається життєвий цикл автоматизованих систем згідно зі стандартом ГОСТ 34.601-90? Які з них належать до передпроектних робіт, які до проектних, а які до узгоджувальної частини проекту? Які стадії можна виключати, об'єднувати між собою і в яких випадках?
3. З яких основних частин складається документація на АС згідно з ГОСТ 34.201-89?
4. За документами яких видів забезпечення створюється документація на АС згідно з ГОСТ 34.201-89? Які об'єкти належать до кожного з видів забезпечення?
5. З яких етапів складається стадія формування вимог до АС? Якими документами закріплюється дана стадія?
6. Що передбачає виконання стадії розробки концепції АС? Які документи є результатом даної стадії?
7. Поясніть необхідність стадії «Технічне завдання».
8. Які роботи проводять на стадії «Технічний проект»? Які основні документи для АС розробляються на даній стадії?
9. Які роботи проводять на стадії «Робоча документація»? Навіщо потрібна робоча документація? Які креслення розробляють на даній стадії? Як документи з технічного проекту включаються в робочу документацію?
10. Які роботи проводять на стадії «Введення в дію»? Якими документами закріплюється дана стадія?
11. Які роботи проводять на стадії «Супроводження»?
12. Що розуміти під процесом створення інтегрованих автоматизованих систем, згідно із запропонованим у посібнику підходом? Що є результатом створення ІАС?
13. Яким чином можна розділити життєвий цикл створення ІАС від АС? Прокоментуйте життєвий цикл ІАС. Поясніть необхідність формування технічного завдання та виконання технічного проекту для ІАС для виконання відповідних стадій для підсистем, що входять до неї (АС).
14. Які основні роботи проводять на стадії дослідження об'єкта та формування вимог для ІАС? Чим відрізняються дослідницькі роботи над об'єктами АС, що розробляються, від робіт над наявними АС.

15. За якими критеріями рекомендується проводити вибір програмно-технічних засобів при аналізі концепцій для ІАС?
16. З яких розділів повинно складатися технічне завдання згідно з ГОСТ 34.602-89?
17. Які пункти технічного завдання АС найбільше залежать від технічного проекту на ІАС?
18. Яка послідовність розробки та комплектність документації використовуються при розробці технічного проекту для ІАС у запропонованому посібником підході проектування?
19. Які основні підходи пропонуються при розробці проекту на ІАС?
20. Які основні питання необхідно вирішити на стадії технічного проекту для ІАС?
21. Які основні структурні схеми використовуються при розробці технічного проекту? Розкажіть про їх призначення. До яких видів забезпечення вони належать?
22. Поясніть призначення та необхідність розробки схеми функціональної структури. Якими додатковими документами вона супроводжується згідно з РД 50-34.698.90?
23. Поясніть призначення та необхідність розробки технічної структури. На яких стадіях вона розробляється? Чим відрізняється узагальнена структура від деталізованої структури КТС?
24. Поясніть місце моделей інформаційних потоків у життєвому циклі розробки ІАС.
25. Поясніть принципи побудови та використання графових моделей інформаційних потоків.
26. Поясніть принципи побудови та використання блочних моделей інформаційних потоків.
27. Які розділи повинна вмещувати комплектність документів на інформаційне забезпечення?
28. Поясніть місце схеми мережних інформаційних потоків у життєвому циклі розробки ІАС. На основі яких документів розробляють схему мережних інформаційних потоків? Яке призначення схеми інформаційних потоків?
29. Які основні елементи показуються в схемі інформаційних потоків?
30. Що розуміти під масивом/блоком даних у схемах інформаційних потоків? Які рівні вкладеності необхідно показати на схемі для достатньої ідентифікації даних?
31. Які є способи відображення даних у масивах? Які правила використання цих способів?
32. Яким чином зображуються інформаційні потоки? Як інформаційні потоки на схемі мережних інформаційних потоків перегукуються з відповідними на моделі?
33. Яким чином показуються напрямки інформаційних потоків? Які об'єкти з'єднують потоки? Яким чином показується ініціатор обміну в потоці?
34. Що розуміти під комунікаційними сервісами, драйверами, логічним каналом та інформаційними шинами в схемах мережних інформаційних потоків? Розкажіть про основні правила побудови проходження інформаційних потоків.
35. Яким чином можна зобразити використання декількох сервісів на одній і тій самій мережі? Яким чином можна ідентифікувати вузли, до яких ідуть потоки у великих та складних схемах?
36. Що необхідно показати на схемі мережних інформаційних потоків при проектуванні мереж MODBUS?
37. Які особливості побудови схеми мережних інформаційних потоків при проектуванні мереж на базі Ethernet?
38. Які особливості побудови схеми мережних інформаційних потоків при проектуванні мереж на базі PROFIBUS?
39. Які особливості побудови схеми мережних інформаційних потоків при проектуванні мереж на базі CANOpen?

Список використаної літератури

1. *Бойерле, Ханс-Петер*. Коммуникация в технике автоматизации / Ханс-Петер Бойерле и Гюнтер Бах-Беценар. — Берлин; Мюнхен: АО Siemens, 1991, 155 с. [ISBN 3-8009-1563-4]
2. *Гук М.* Аппаратные средства локальных сетей. Энциклопедия / Гук М. — СПб: Издательство «Питер», 2000. — 576 с.: ил.
3. *Дансмор, Брэд*. Справочник по телекоммуникационным технологиям / Дансмор, Брэд, Скандьер, Тоби.; пер. с англ. — М.: Издательский дом «Вильямс», 2004. — 640 с.: ил. [ISBN 5-8459-0562-1 (рус.)]
4. *Джонсон Говард В.* Высокоскоростная передача цифровых данных: высший курс черной магии.: пер. с англ. Парал. Тит. — М.: Издательский дом «Вильямс», 2005. — 1024 с.: ил. [ISBN 5-8459-0824-8 (рус.)]
5. *Джонсон Говард В.* Конструирование высокоскоростных цифровых устройств: начальный курс черной магии. / Джонсон, Говард В., Грэхем, Мартин.: пер. с англ. Парал. Тит. — М.: Издательский дом «Вильямс», 2006. — 624 с.: ил. [ISBN 5-8459-0807-8 (рус.)]
6. Электронный форум АСУТП, режим доступа <http://asutpforum.spb.ru>.
7. Электронный форум АСУТП, режим доступа <http://iprogram.pp.ru/forum>.
8. Электронный форум від фірми Siemens, режим доступа <http://www.automation-drives.ru/forum>.
9. Электронный форум з питань PLC, режим доступа <http://plcforum.uz.ua>.
10. Электронный форум специалистов АВОК, режим доступа <http://forum.abok.ru>.
11. *Елизаров И.А.* Технические средства автоматизации. Программно-технические комплексы и контроллеры: учебное пособие / Елизаров И.А., Мартемьянов Ю.Ф., Схиртладзе А.Г., Фролов С.В. — М.: «Издательство Машиностроение-1», 2004. 180 с. [ISBN 5-94275-104-8]
12. *Ельперін І.В.* Промислові контролери [Текст]: навч. посібник для студентів вищ. навч. закладів / І.В. Ельперін ; Нац. ун-т харч. технологій. — К.: НУХТ, 2003. — 319 с. [ISBN 966-612-024-0]
13. *Лапин А.А.* Интерфейсы. Выбор и реализация / Лапин А.А. — М.: Техносфера, 2005. — 168 с.
14. *Левин Л.С.* Цифровые системы передачи информации / Левин Л.С., Плоткин М.А. — М.: Радио и связь, 1982. — 216 с., ил.

15. Локотков А. Интерфейсы последовательной передачи данных: стандарты. — IIA RS_422A / RS_485 / Локотков Александр // Современные технологии автоматизации (СТА). — №3. — 1997. — С.110 — 119.

16. Луцька Н.М. Використання моделі та схеми інформаційних потоків при розробці автоматизованих систем управління технологічними процесами на прикладі мереж Modbus / Луцька Н.М., Пупена О.М., Івашук О.М. // Восточно-Европейский журнал передовых технологий. — №5/3. — 2008 р. — С. 16 — 22.

17. Нестеров А. Л. Проектирование АСУТП. Книга 1 / Нестеров А. Л.: — СПб. Издательство: ДЕАН. 2006. — 552 с. [ISBN: 5-93630-530-9]

18. Нестеров А. Л. Проектирование АСУТП. Книга 2. / Нестеров А. Л.: — СПб. Издательство: ДЕАН. 2009. — 552 с. [ISBN: 978-5-93630-654-9]

19. Парк Дж. Передача данных в системах контроля и управления: практическое руководство / Парк Дж., Маккей С., Райт Э.; перевод с англ. В.В. Савельева. — М.: ООО «Группа ИДТ», 2007. — 480 с.: ил., табл. [ISBN 978-5-94833-023-5]

20. Прокис Джон. Цифровая связь.: Пер. с англ. / Под ред. Д.Д. Кловского. — М.: Радио и связь. 2000. — 800 с.: ил. [ISBN 5-256-01-434-X]

21. Пономарев О.П. Наладка и эксплуатация средств автоматизации. SCADA-системы. Промышленные шины и интерфейсы. Общие сведения о программируемых логических контроллерах и одноплатных компьютерах: Учебное пособие / О.П. Пономарев; Ин-т «КВШУ». — Калининград: Изд-во Ин-та «КВШУ», 2006. — 80 с.

22. Проектирование и расчет структурированных кабельных систем и их компонентов / Семенов А.Б. — М.: ДМК Пресс; М.: Компания АйТи, 2003. — 416 + 16 с.: ил. [ISBN 5-94074-210-6]

23. Пупена О.М. Використання концепції OPC в сучасних системах автоматизації / Пупена О.М., Ельперін І.В., Ладанюк А.П. // Автоматизація виробничих процесів. — 2003. — №1. — С. 65 — 70.

24. Пупена О.М. Інтеграція систем управління / Пупена О.М., Ельперін І.В. // Харчова і переробна промисловість. — 2005. — №1. — С. 9 — 11.

25. Пупена А.Н. Планирование и монтаж кабельной системы для промышленных сетей / Пупена А.Н., Ельперин И.В. // Автоматизация в промышленности. — 2009. — №12. — С. 26 — 29.

26. Пупена О.М. Проектування комп'ютерно-інтегрованих систем: курс лекцій для студ. спец. 7.092502 «Комп'ютерно-інтегровані технологічні процеси та виробництва» напряму підготовки 0925 «Автоматизація та комп'ютерно-інтегровані технології» ден. та заоч. форм навч. / Пупена О.М., Луцька Н.М., Ельферін І.В. — К.: НУХТ, 2007. — 142 с.

27. Пупена А.Н. Развитие и становление международной стандартизации промышленных сетей / Пупена А.Н., Ельперин И.В. // Автоматизация в промышленности. — 2009. — №3. — С. 19 — 23.

28. Полевая шина FOUNDATION Fieldbus, Технический обзор FD-043, версия 3.0 [Электронный ресурс, режим доступа <http://www.fieldbus.org>]

29. Рихтер Дж. Windows для профессионалов: создание эффективных Win32 приложений с учетом специфики 64-разрядной версии Windows / Рихтер Дж.; пер, англ. — 4-е изд. — СПб; Питер. — М.: Издательско-торговый дом «Русская Редакция», 2001. — 752 с.; ил. [ISBN 5-272-00384-5]

30. Семенов А. Международный стандарт на СКС промышленного назначения / Семенов А. // LAN. Журнал сетевых решений. — 2007. — №10.
31. Скляр, Бернард. Цифровая связь. Теоретические основы и практическое применение / Скляр, Бернард. Изд. 2-е, испр.: Пер. с англ. — М.: Издательский дом «Вильямс», 2003. — 1104 с.: ил. [ISBN 5-8459-0497-8 (рус)]
32. Таненбаум Эндрю. Компьютерные сети. 3-е изд / Таненбаум Эндрю. — СПб.: Питер, 2002. — 848 стр. [ISBN 5-318-00300-1]
33. Таненбаум Эндрю. Распределенные системы Принципы и парадигмы / Эндрю Таненбаум, Маартен ван Стеен. — СПб.: Питер, 2003. — 880 стр. [ISBN 5-272-00053-6]
34. Теркель Д. OLE for Process Control — свобода выбора / Теркель Д. // Современные технологии автоматизации (СТА). — №3. — 1999. — С. 28 — 32.
35. Техническая коллекция Schneider Electric. [Электронный ресурс] // Руководство по организации сети Modbus. — 2007. — № 8. — Режим доступа: <http://www.schneider-electric.com>.
36. Турш Ф. Введение в технологию LonWorks: пер. с англ. — М.: Энергоатомиздат, 2001. — 144 с., ил. [ISBN 5-283-0317-5 (рус)]
37. Федоров Ю.Н. Справочник инженера по АСУТП: Проектирование и разработка. Учебно-практическое пособие / Федоров Ю.Н. — М.: Инфра-Инженерия, 2008. — 928 стр., 12 ил. [ISBN 978-5-9729-0019-0]
38. Финк Л.М. Сигналы, помехи, ошибки. Заметки о некоторых неожиданно-парадоксах и заблуждениях в теории связи / Финк Л.М. — 2-е изд., перераб. и доп. — М.: Радио и связь, 1984. — 256 с., ил.
39. Хорстман К.С. Библиотека профессионала. Java 2. Основы.: Пер. с англ. / Хорстман К.С., Корнелл Г. — М.: Издательский дом «Вильямс», 2003. — 848 с.: ил. — парал. Тит. англ. [ISBN 5-8459-0385-8 (рус)]
40. Шишов О.В. Современные технологии промышленной автоматизации: учебник / О. В. Шишов. — Саранск: Изд-во Мордов. ун-та, 2007. — 250 с. [ISBN 5-7103-1123-5]
41. Benzi Francesco, Buja Giuseppe S., Felser Max. Communication Architectures for Electrical Drives // IEEE Transaction on industrial informatics, VOL.1, No.1, February 2005, pp. 47–53.
42. CANopen Communication Profile for Industrial Systems based on CAL, CiA DS 301, Ver. 3.93, 1998.
43. David Bailey. Practical SCADA for Industry / David Bailey, Edwin Wright.: IDC Technologies, 2003, — 298 с. [ISBN 07506 58053]
44. Industrial Data Communication. Theoretical and General Applications. Westermo Handbook 5.0: Westermo, Sweden, 2005. — 158 p.
45. IAONA Industrial Ethernet — Planning and Installation Guide Version 4.0: IAONA 2003. — 63p.
46. IEC 61158. Industrial communication networks. Fieldbus specifications. [Электронный доступ до стандарту: <http://webstore.iec.ch/>]
47. IEC 61784. Industrial communication networks. Profiles Industrial communication networks. [Электронный доступ до стандарту: <http://webstore.iec.ch/>]

48. IEC 61800-7-1: Adjustable speed electrical power drive systems — Part 7-1: Generic interface and use of profiles for power drive systems — Interface definition. [Електронний доступ до стандарту: <http://webstore.iec.ch/>]
49. IEC 61918 — Industrial communication networks. Installation of communication networks in industrial premises. [Електронний доступ до стандарту: <http://webstore.iec.ch/>]
50. IMS CANopen Implementation DS-301 and DSP-402: Intelligent Motion Systems, 2008. [Електронний ресурс, режим доступу www.imshome.com]
51. ISO/IS 11898: Road Vehicles. Interchange of Digital Information . Controller Area Network (CAN) for High Speed Communication, 1993.
52. *Felser M.* The Fieldbus Standard: History and Structure. Technology Leadership Day 2002. Organised by MICROSWISS Network. HTA Luzern. №10. Oktober. 2002.
53. *Jonas Berge.* Introduction to Fieldbuses for Process Control: ISA — The Instrumentation, Systems, and Automation Society, 2002. — 40 p.
54. *Josef Weigmann, Gerhard Killian.* Decentralization with PROFIBUS DP/DPV1. Second Edition: Publicis Kommunikations Agentur Gmbh, GWA Erlagen, 2003. — 252 p. [ISBN 3-89578-218-1]
55. J. DE AZEVEDO. N. CRAVOISY. The WorldFIP protocol, 1998.
56. HART Протокол первичной связи. Технический обзор. [Електронний ресурс] / Fisher Rosemount. — Режим доступу: www2.emersonprocess.com.
57. Holger Zeltwanger. Generic Power Drive Systems // PRAXIS Profiline — Factory Automation — April 2007. pp. 2—3.
58. *Hugh Jack.* Automating Manufacturing Systems with PLCs Version 5.0 [Електронний ресурс] / Hugh Jack, 2007, — 839 p. — Режим доступу: jackh@gvsu.edu.
59. *Hugh Jack.* Integration and Automation of Manufacturing Systems [Електронний ресурс] / Hugh Jack. 2007, — 593 p. — Режим доступу: jackh@gvsu.edu.
60. *Max Felser.* Real-Time Ethernet — Industry Prospective // PROCEEDINGS OF THE IEEE. — VOL. 93. — NO.6. — JUNE 2005. — pp.1118-1128.
61. MODBUS APPLICATION PROTOCOL SPECIFICATION, V1.1a — Modbus-IDA, 2004. [Електронний доступ до стандарту: <http://www.Modbus-IDA.org>]
62. MODBUS MESSAGING ON TCP/IP IMPLEMENTATION GUIDE V1.0a. — Modbus-IDA, 2004. [Електронний доступ до стандарту: <http://www.Modbus-IDA.org>]
63. MODBUS over Serial Line Specification and Implementation Guide V1.02. - Modbus-IDA, 2006, [Електронний доступ до стандарту: <http://www.Modbus-IDA.org>]
64. ODVA. The Common Industrial Protocol (CIP™) and the Family of CIP Networks. Publication Number: PUB00123R0.
65. OLE For Process Control (OPC) Data Access Custom Interface: Standard Version 2.0: OPC Foundation, 1998 [Електронний ресурс, режим доступу <http://www.opcfoundation.org/>]
66. Practical Fieldbus, DeviceNet and Ethernet for Industry: IDC Technologies. 2005. — 447 p.
67. Premium and Atrium using Unity Pro AS-i Bus User manual. — Schneider Electric: September, 2004. [Електронний доступ: <http://www.schneider-electric.com>]

68. PROFIBUS Installation Guideline for Cabling and Assembly: PROFIBUS Nutzerorganisation e.V., 2006. [Електронний ресурс, режим доступу <http://www.profibus.com>]

69. PROFIBUS Technology and Application. System Description.: PROFIBUS Trade Organization, 2002. [Електронний ресурс, режим доступу <http://www.profibus.com>]

70. PROFIDRIVE technology and application. System Description: PROFIBUS Nutzer Organization, 2007. [Електронний ресурс, режим доступу <http://www.profibus.com>]

71. Real-Time Publish-Subscribe (RTPS) Wire Protocol Specification: Real-Time Innovations, 2001. [Електронний ресурс, режим доступу <http://www.rti.com>]

72. *Reinhard Langmann*. INTERBUS BASICS / Reinhard Langmann. [Електронний ресурс, режим доступу <http://www.teleautomation.de>]

73. Robert Bosch GmbH: CAN Specification Ver.2.0, 1991.

74. RS-422 and RS-485 Application Note. B&B Electronics, 1997. [Електронний ресурс, режим доступу <http://www.bb.elec.com>]

75. SIMATIC NET AS-Interface — Introduction and Basics. Release 04. — Siemens AG, 2006. — 62 p. [Електронний доступ: <http://www.automation.siemens.com>]

76. Todd A. Snide, CIP-Modbus Integration. — Hannover Messe, 2008. [Електронний доступ до стандарту: www.odva.org, www.modbus.org]

77. Transparent Ready User Guide. — Schneider Electric, 2007. [Електронний ресурс, режим доступу <http://www.schneider-electric.com>]

78. *Wolfgang Mahnke*. OPC Unified Architecture / Wolfgang Mahnke, Stefan-Helmut Leitner, Matthias Damm.: Springer-Verlag Berlin Heidelberg, 2009. — 351 p. [ISBN 978-3-540-68898-3]

79. *Zurawski, Richard*. Integration technologies for industrial automated systems / Richard Zurawski. p. cm. — (Industrial information technology series): CRC Press, Taylor & Francis Group. 2007. — 539 p. [ISBN 0-8493-9262-4]

80. *Zurawski, Richard*. The industrial communication technology handbook / Richard Zurawski, editor. p. cm. — (The industrial information technology series): CRC Press, Taylor & Francis Group. 2005. — 879 p. [ISBN 0-8493-3077-7.]

**О.М. Пупена, І.В. Ельперін,
Н.М. Луцька, А.П. Ладанюк**

ПРОМИСЛОВІ МЕРЕЖІ ТА ІНТЕГРАЦІЙНІ ТЕХНОЛОГІЇ В АВТОМАТИЗОВАНИХ СИСТЕМАХ

Навчальний посібник

Керівник видавничого проекту *Зарицкій В. І.*
Дизайн обкладинки *Сєдих О. О.*
Комп'ютерна верстка *Іваненко О. М.*

Підписано до друку 22.11.2010. Формат 70×100 1/16.
Папір офсетний. Друк офсетний. Гарнітура Times New Roman.
Умовн. друк. аркушів — 29,76. Обл.-вид. аркушів — 34,82. Тираж
Зам. № 3/11-2010.

Видавництво «Ліра-К»
Свідоцтво ДК № 1156 від 17.12.2002 р.
03067, м. Київ, вул. Прилужна 14, оф. 42
тел./факс (044) 247-93-37; 450-91-96

Друк ТзОВ “Видавнича фірма «Афіша»”

**Замовлення на книги видавництва «Ліра-К»,
а також на книги інших видавництв можна зробити
наступним чином:**

по факсу: (044) 450-91-96;
електронною поштою: lira-k@ukr.net;
або по телефону: (044) 247-93-37; 228-81-12.

Наша адреса:

03179, м. Київ, вул. Прилужна, 14, к. 42 ТОВ «Ліра-К».

Банківські реквізити:

р/р 2600600640118 в АКБ «ПРАВЕКС-БАНК»

Білицьке відділення м. Київ

МФО 321983, ЄДРПОУ 31720302

ІПН 317203026059, номер свідоцтва 36055184.

Свідоцтво про внесення суб'єкта видавничої справи до державного реєстру видавців, виготівників і розповсюджувачів видавничої продукції № 1156 серія ДК.

**Видавництво запрошує до співпраці авторів
на взаємовигідних умовах**